รายงานผลการวิจัย

เรื่อง

การเปรียบเทียบวิธีการสร้างตัวเลขสุ่มเมื่อนำไปประมาณค่าพารามิเตอร์ของการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่อง

The Comparative of Random number Generation when Estimate Parameter of Continuous Random Variable Distribution and Discrete Random Variable Distribution

โดย

นภาภรณ์ จันทรศัพท์

รายงานการวิจัยนี้ได้รับทุนอุดหนุนจาก มหาวิทยาลัยธุรกิจบัณฑิตย์
พ.ศ. 2551

ชื่อเรื่อง	การเปรียบเทียบวิธีการสร้างตัวเลขสุ่มเมื่อนำไปประมาณค่าพารามิเตอร์ของการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่อง
ผู้วิจัย	นภาภรณ์ จันทรศัพท์ สถาบัน มหาวิทยาลัยธุรกิจบัณฑิตย์
ปีที่พิมพ์	2551 สถานที่พิมพ์ มหาวิทยาลัยธุรกิจบัณฑิตย์
แหล่งที่เก็บรายงานการวิจัยฉบับสมบูรณ์ : มหาวิทยาลัยธุรกิจบัณฑิตย์	จำนวนหน้างานวิจัย 217 หน้า
คำสำคัญ -	ลิขสิทธิ์ มหาวิทยาลัยธุรกิจบัณฑิตย์

บทคัดย่อ

การศึกษาเรื่อง การเปรียบเทียบวิธีการสร้างตัวเลขสุ่มเมื่อนำไปประมาณค่าพารามิเตอร์ของการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่อง มีวัตถุประสงค์ในการวิจัย คือ 1.เพื่อเปรียบเทียบวิธีการสร้างตัวเลขสุ่ม 5 วิธีสำหรับนำไปประมาณค่าพารามิเตอร์ของการแจกแจงตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงตัวแปรสุ่มแบบไม่ต่อเนื่อง และ 2. เพื่อศึกษาว่าวิธีการสร้างตัวเลขสุ่มวิธีใดที่ดีที่สุดสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่อง โดยการแจกแจงตัวแปรสุ่มแบบต่อเนื่องในที่นี้ศึกษา 2 การแจกแจง คือ การแจกแจงแบบเอ็กซ์โปเนนเชียล (Exponential Distribution) และการแจกแจงแบบปกติ (Normal Distribution) ส่วนการแจกแจงตัวแปรสุ่มแบบไม่ต่อเนื่องศึกษา 2 การแจกแจง คือ การแจกแจงแบบพัวซอง (Poisson Distribution) และ การแจกแจงแบบทวินาม (Binomial Distribution) สำหรับการแจกแจงแบบเอ็กซ์โปเนนเชียล (Exponential Distribution) กำหนดค่าพารามิเตอร์ (β) ที่ใช้ในการทดสอบ เท่ากับ 1.3 , 1.8 , 2.0 , 2.4 และ 2.9 การแจกแจงแบบปกติ (Normal Distribution) กำหนดค่าพารามิเตอร์ (μ ,σ^2) ที่ใช้ในการทดสอบเท่ากับ (0,1),(1,3),(2,5),(3,6) และ (4,5) ส่วนค่าพารามิเตอร์ของการแจกแจงแบบไม่ต่อเนื่อง สำหรับการแจกแจงแบบพัวซอง (Poisson Distribution) กำหนดค่าพารามิเตอร์ (λ) สำหรับการทดสอบเท่ากับ 1.3 , 1.8 , 2.0 , 2.4 และ 2.9 ส่วนการแจกแจงแบบทวินาม (Binomial Distribution) กำหนดค่าพารามิเตอร์ (n,p) สำหรับการทดสอบเท่ากับ (20,0.1),(30,0.3),(40,0.5),(50,0.6) และ (70,0.8) ในทุกการแจกแจงทดสอบที่ขนาดตัวอย่าง เท่ากับ 20 หน่วยตัวอย่าง 40 หน่วยตัวอย่าง 60 หน่วยตัวอย่าง 80 หน่วยตัวอย่าง และ 100 หน่วยตัวอย่าง 500 หน่วยตัวอย่าง 1000 หน่วยตัวอย่าง 1500 หน่วยตัวอย่าง และ 2000 หน่วยตัวอย่าง ที่จำนวนรอบของการหาค่าเฉลี่ยของความคลาดเคลื่อน (MSE : Mean Square Error) 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

ผลการวิจัยพบว่า วิธีการสร้างตัวเลขสุ่มทั้ง 5 วิธี คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method เป็นวิธีที่เมื่อนำไปใช้ในการประมาณค่าพารามิเตอร์ของการแจกแจงตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงตัวแปรสุ่มแบบไม่ต่อเนื่องสามารถใช้ได้ดีทุกวิธี โดยไม่มีวิธีการสร้างตัวเลขสุ่มวิธีใดที่ดีที่สุดทั้งสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องโดยในการศึกษาครั้งนี้ทดสอบกับ 2 การแจกแจง คือ การแจกแจงแบบเอ๊กซ์โปเนนเชียล (Exponential Distribution) และการแจกแจงแบบปกติ (Normal Distribution) และการประมาณค่าพารามิเตอร์ของการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่องสำหรับการศึกษาในครั้งนี้ทดสอบกับ 2 การแจกแจง คือ การแจกแจงแบบพัวซอง (Poisson Distribution) และการแจกแจงแบบทวินาม (Binomial Distribution)

Title : The Comparative of Random number Generation when Estimate Parameter of Continuous Random Variable Distribution and Discrete Random Variable Distribution
Researcher: Napaporn Juntarasab
Institution : Dhurakij Pundit University
Year of Publication : 2008
Publisher : Dhurakij Pundit University
Sources : Dhurakij Pundit University
No. of page : 217 pages
Keyword : -

Abstract

The study of the comparative of random number generation when estimate parameter of continuous random variable distribution and discrete random variable distribution have two objective the first is for comparative random number generation when estimate parameter of continuous random variable distribution and discrete random variable distribution the second is for study that which the best random number generation for estimate parameter , for this study estimate exponential distribution with parameter 1.3,1.8,2.0,2.4 and 2.9 and normal distribution with parameter (0,1),(1,3),(2,5),(3,6) and (4,5) of continuous random variable distribution , poisson distribution with parameter 1.3,1.8,2.0,2.4 and 2.9 and binomial distribution with parameter (20,0.1),(30,0.3),(40,0.5),(50,0.6) and (70,0.8) of discrete random variable distribution for all distributions test at sample size 20, 40, 60, 80 ,100 ,500,1000,1500 and 2000 at 500 , 1000, 2000 and 3000 loop test for Root of Mean Square Error : RMSE

The results indicate that all 5 methods of random number generation which the first method is Midsquare Method the second method is Midproduct Technique the third method is Constant Multiplier Technique the fourth method is Additive Congruential Method and the last method is Linear Congruential Method , all 5 methods were efficient for estimation parameter of continuous random variable distribution (exponential distribution and normal distribution) and discrete random variable (poisson distribution and binomial distribution).

กิตติกรรมประกาศ

งานวิจัยเรื่อง การเปรียบเทียบวิธีการสร้างตัวเลขสุ่มเมื่อนำไปประมาณค่าพารามิเตอร์ของการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่อง ได้รับทุนสนับสนุนการวิจัยจากมหาวิทยาลัยธุรกิจบัณฑิตย์ และขอขอบคุณรองศาสตราจารย์ ดร.สรชัย พิศาลบุตร ที่ให้คำปรึกษา แนะนำ รวมทั้งกรรมการทุกท่านที่พิจารณาแก้ไข ท้วงติง และแนะนำเพื่อความถูกต้องเหมาะสม ขอขอบคุณรองศาสตราจารย์อุปถัมภ์ สายแสงจันทร์ ที่ให้คำแนะนำในการเริ่มต้นการทำงานวิจัย รวมทั้งหน่วยงานที่เกี่ยวข้องที่อำนวยความสะดวกในการดำเนินการ ขอขอบคุณทุกท่านอีกครั้งทั้งที่ระบุและไม่ระบุนามไว้ ณ โอกาสนี้

ผู้วิจัย

สารบัญ

ข. ประวัติผู้วิจัย

สารบัญตารางและกราฟ

สารบัญตารางกราฟ (ต่อ)

ตารางที่ **หน้า**

สารบัญตารางและกราฟ (ต่อ)

สารบัญตารางและกราฟ (ต่อ)

สารบัญตารางและกราฟ (ต่อ)

สารบัญตารางและกราฟ (ต่อ)

ตารางที่ **หน้า**

สารบัญตารางและกราฟ (ต่อ)

ตารางที่		หน้า
50	แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุดสำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ	112
51	แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุดสำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ	114
52	แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุดสำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ	116
53	แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุดสำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ	118
54	แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุดสำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ	120
55	แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุดสำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ	122
56	แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุดสำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ	124
57	แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุดสำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ	126
58	แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุดสำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 500 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ	128
59	แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุดสำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 500 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ	130

สารบัญตารางและกราฟ (ต่อ)

ตารางที่ **หน้า**

บทที่ 1
บทนำ

ที่มาและความสำคัญของปัญหา

เทคนิคการจำลองเป็นวิธีการหนึ่งที่ใช้ในการแก้ปัญหาหรือศึกษาถึงระบบงานในด้านต่าง ๆ เป็นกระบวนการออกแบบตัวแบบจำลอง(Model) ของระบบงานจริง (Real System) แล้วดำเนินการใช้ตัวแบบจำลองนั้นเพื่อเรียนรู้พฤติกรรมของระบบ หรือประเมินผลการดำเนินงานการใช้แผนงานต่าง ๆตัวอย่างเช่น การจำลองระบบงานด้านอุตสาหกรรม เช่น ระบบสินค้าคงคลัง ระบบแถวคอย ระบบการสื่อสาร การจำลองระบบงานด้านบริหารธุรกิจและเศรษฐศาสตร์ เช่น การศึกษาภาวะการตลาด ภาวะเงินเฟ้อ พฤติกรรมผู้บริโภค การจำลองด้านธุรกิจต่าง ๆเมื่อผู้บริหารมีแผนการแบบต่าง ๆมาทดลองใช้ เป็นต้น

ตัวเลขสุ่มมีความจำเป็นอย่างมากในการจำลองปัญหาเกือบทั้งหมดของระบบไม่ต่อเนื่อง ในภาษาคอมพิวเตอร์เกือบทุกภาษามักจะมีโปรแกรมย่อยหรือฟังก์ชั่นในการสร้างตัวเลขสุ่ม โดยตัวเลขสุ่มที่สร้างขึ้นนี้จะนำมาใช้ในการหาเวลาของการเกิดเหตุการณ์ต่าง ๆหรือแม้กระทั่งการนำไปสร้างค่าตัวแปรสุ่มแบบต่าง ๆที่มีอยู่ก็สามารถนำไปประยุกต์ใช้กับงานจริง ๆได้ ขณะเดียวกันการสร้างตัวเลขสุ่มแบบต่าง ๆที่มีผู้คิดขึ้นมาก็สามารถนำไปใช้งานได้ โดยในที่นี้จะศึกษาถึงวิธีการสร้างตัวเลขสุ่มที่มีผู้คิดขึ้น 5 วิธี คือ

1. วิธีตัดกลางกำลังสอง (Midsquare Method)
2. วิธีตัดกลางของผลคูณ (Midproduct Technique)
3. วิธีตัวคูณคงที่ (Constant Multiplier Technique)
4. วิธี Additive Congruential Method
5. วิธี Linear Congruential Method

โดยจะศึกษาว่าแต่ละวิธีเมื่อสร้างตัวเลขสุ่มได้แล้ว ตัวเลขสุ่มดังกล่าวเมื่อนำไปสร้างข้อมูลของการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องโดยในที่นี้ใช้การแจกแจงแบบเอ็กซ์โพเนนเชียล (Exponential Distribution) และการแจกแจงแบบปกติ (Normal Distribution) ส่วนการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่องในที่นี้ใช้การแจกแจงแบบพัวซอง (Poisson Distribution) และการแจกแจงแบบทวินาม (Binomial Distribution) โดยเมื่อนำไปประมาณค่าพารามิเตอร์ของแต่ละการแจกแจงแล้ว วิธีการสร้างตัวเลขสุ่มวิธีใดที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุดของแต่ละการแจกแจง วิธีการสร้างตัวเลขสุ่มวิธีนั้นจะเป็นวิธีการสร้างตัวเลขสุ่มที่ดีที่สุดสำหรับใช้ในการประมาณค่าพารามิเตอร์

วัตถุประสงค์

1. เพื่อเปรียบเทียบวิธีการสร้างตัวเลขสุ่ม 5 วิธีสำหรับนำไปประมาณค่าพารามิเตอร์ของการแจกแจงตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงตัวแปรสุ่มแบบไม่ต่อเนื่อง
2. เพื่อศึกษาว่าวิธีการสร้างตัวเลขสุ่มวิธีใดที่ดีที่สุดสำหรับการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่อง

ขอบเขตการวิจัย

1. ในการศึกษานี้จะศึกษาวิธีการสร้างตัวเลขสุ่ม 5 วิธี ที่มีผู้คิดค้นขึ้น คือ
    1. วิธีตัดกลางกำลังสอง (Midsquare Method)
    2. วิธีตัดกลางของผลคูณ (Midproduct Technique)
    3. วิธีตัวคูณคงที่ (Constant Multiplier Technique)
    4. วิธี Additive Congruential Method
    5. วิธี Linear Congruential Method
2. วิธีการประมาณค่าในที่นี้ใช้การประมาณค่าแบบสภาวะสูงสุด (Maximum Likelihood Estimation) และ วิธีกำลังสองน้อยที่สุด (Method of Least Square)
3. การแจกแจงของตัวแปรสุ่มแบบต่อเนื่องในที่นี้ศึกษาการแจกแจงแบบเอ๊กซ์โพเนนเชียล (Exponential Distribution) และการแจกแจงแบบปกติ (Normal Distribution) ส่วนการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่องศึกษาการแจกแจงแบบพัวซอง (Poisson Distribution) และการแจกแจงแบบทวินาม (Binomial Distribution)

ประโยชน์ที่คาดว่าจะได้รับ

1 ได้ทราบว่าวิธีการสร้างตัวเลขสุ่มแต่ละวิธีเมื่อนำไปใช้ โดยในที่นี้ใช้ในการประมาณค่าพารามิเตอร์ของการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่อง ค่าประมาณที่ได้จากวิธีการสร้างตัวเลขสุ่มใดที่ดีที่สุดสำหรับแต่ละการแจกแจงโดยพิจารณาจากค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error)

2 เป็นแนวทางในการศึกษาวิจัยต่อไป

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการเปรียบเทียบวิธีการสร้างตัวเลขสุ่มเมื่อนำไปประมาณค่าพารามิเตอร์สำหรับการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่อง ในที่นี้ได้เรียบเรียงข้อมูลที่เกี่ยวข้องกับงานวิจัยตามลำดับ ดังนี้

1. วิธีการสร้างตัวเลขสุ่ม 5 วิธี
2. ตัวแปรสุ่มแบบต่อเนื่องและตัวแปรสุ่มแบบไม่ต่อเนื่อง
3. ค่าคาดหวังและความแปรปรวนของตัวแปรสุ่มแบบต่อเนื่องและตัวแปรสุ่มแบบไม่ต่อเนื่อง
4. วิธีการสร้างค่าตัวแปรสุ่ม
5. ฟังก์ชันการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องและฟังก์ชันการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่องและการสร้างค่าตัวแปรสุ่มของแต่ละการแจกแจง
6. วิธีการประมาณค่า 2 วิธี คือ

6.1 การประมาณค่าแบบสภาวะสูงสุด (Maximum Likelihood Estimation)

6.2 วิธีกำลังสองน้อยที่สุด (Method of Least Square)

1. วิธีการสร้างตัวเลขสุ่ม 5 วิธี

ในที่นี้ศึกษาวิธีการสร้างตัวเลขสุ่มที่มีผู้คิดค้นขึ้น 5 วิธี คือ

1. วิธีตัดกลางกำลังสอง (Midsquare Method)
2. วิธีตัดกลางของผลคูณ (Midproduct Technique)
3. วิธีตัวคูณคงที่ (Constant Multiplier Technique)
4. วิธี Additive Congruential (Additive Congruential Method)
5. วิธี Linear Congruential (Linear Congruential Method)

1. วิธีตัดกลางกำลังสอง (Midsquare Method)

โดย Von-Neuman & Metropolis วิธีนี้อาศัยตัวเลขเริ่มต้น (Seed) มายกกำลังสองและตัวตัวเลขส่วนหัวและส่วนท้ายออกด้วยจำนวนหลักเท่า ๆกัน ตัวเลขที่อยู่ตรงกลางจะนำมาใช้เป็นตัวเลขสุ่มโดยทำให้ตัวเลขนั้นเป็นเลขทศนิยม ทำเช่นนี้ต่อเนื่องไปเรื่อย ๆจนได้ตัวเลขสุ่มตามจำนวนที่ต้องการ

ตัวอย่าง ให้ X0 เป็นตัวเลขเริ่มต้น เท่ากับ 3579

$X_0^2 = (3579)^2 = 12809241 = X_1 = 8092$

$R_1 = 0.8092$

$X_1^2 = (8092)^2 = 65480464 = X_2 = 4804$

$R_2 = 0.4804$

.

.

2. วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีการนี้คล้ายกับวิธีตัดกลางกำลังสอง แต่ตัวเลขเริ่มต้นมี 2 ตัว คือ X_0' เป็นตัวเลขเริ่มต้นตัวแรก X_0 เป็นตัวเลขเริ่มต้นตัวที่สอง โดย X_0' , X_0 เป็นเลขขนาด d หลัก นำ X_0' และ X_0 คูณกันแล้วนำตัวเลขตรงกลางขนาด d หลักมาเป็น X_1 ทำเป็นเลขจุดทศนิยมก็จะกลายเป็น R_1 นำ X_0 คูณกับ X_1 แล้วนำตัวเลขตรงกลาง d หลักมาเป็น X_2 แล้วสร้าง R_2 ต่อไปทำเช่นนี้ไปเรื่อย ๆจนได้ตัวเลขสุ่มตามจำนวนที่ต้องการ

ตัวอย่าง $X_0' = 3579$ $X_0 = 9753$

$U_1 = X_0' X_0 = (3579)(9753) = 34905987 = X_1 = 9059$

$R_1 = 0.9059$

$U_2 = X_0 X_1 = (9753)(9059) = 88352427 = X_2 = 3524$

$R_2 = 0.3524$

.

.

.

3. วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีตัวคูณคงที่ ต่างจากวิธีตัวกลางของผลคูณ โดยวิธีการนี้มีค่าคงที่ 1 ตัวคือ k นำมาคูณกับ ตัวเลขเริ่มต้น (X_0) ซึ่ง X_0 มีขนาด d หลัก แล้วนำเอาตัวเลขตรงกลางขนาด d หลักมาเป็น X_1 แล้วนำมาสร้าง R_1 จากนั้นนำ k มาคูณกับ X_1 ซึ่งมีขนาด d หลัก แล้วนำเอาตัวเลขตรงกลางขนาด d หลักมาเป็น X_2 แล้วนำมาสร้าง R_2 และทำเช่นนี้ไปเรื่อย ๆ จนได้ตัวเลขสุ่มตามจำนวนที่ต้องการ

ตัวอย่าง $k = 1357$, $X_0 = 3579$

$V_1 = k X_0$

$V_1 = (1357)(3579) = 04856703 = X_1 = 8567$

$R_1 = 0.8567$

$V_2 = k X_1$

$V_2 = (1357)(8567) = 11625419 = X_2 = 6254$

$R_2 = 0.6254$

.

.

4. วิธี Additive Congruential (Additive Congruential Method)

วิธีนี้กำหนดตัวเลขจำนวนเต็ม $X_1, X_2, \ldots, X_n$ ตามลำดับ นำตัวเลขเหล่านี้มาสร้าง $X_{n+1}, X_{n+2}, \ldots$ จากนั้นนำตัวเลขที่สร้างมานี้มาสร้าง $R_1, R_2, \ldots$ ต่อไป โดยสูตรการสร้างซึ่งมีการทำ Modulo m ซึ่งจะทำให้ค่าที่ได้เป็นเลขเศษที่มีค่าน้อยกว่า m ซึ่งการคำนวณค่าตามสูตรต่อไปนี้

$$X_i = (X_{i-1} + X_{i-n}) \bmod m$$

$$R_{i-n} = \frac{X_i}{m}$$

ตัวอย่าง $X_1 = 65$, $X_2 = 24$, $X_3 = 39$, $X_4 = 88$, $X_5 = 75$,
$n = 5$, $m = 100$

$$X_i = (X_{i-1} + X_{i-n}) \bmod m$$

$$X_6 = (X_5 + X_1) \bmod 100 = (75 + 65) \bmod 100$$

$$= 140 \bmod 100 = 40$$

$$R_{6-5} = R_1 = 40/100 = 0.40$$

$$X_7 = (X_6 + X_2) \bmod 100 = (40 + 24) \bmod 100$$

$$= 64 \bmod 100 = 64$$

$$R_{7-5} = R_2 = 0.64$$

.

.

.

5. วิธี Linear Congruential (Linear Congruential Method)

วิธีนี้ได้พัฒนา โดย Lehmer ในปี ค.ศ.1951 และเป็นวิธีที่นิยมใช้กันในปัจจุบัน วิธีนี้มีสูตรการคำนวณหาดังนี้

$$X_{i+1} = (a X_i + c) \bmod m \qquad ; i = 0,1,2,...$$

$X_1, X_2, X_3, ..., X_i, X_{i+1}$ เป็นเลขจำนวนเต็มอยู่ระหว่าง 0 กับ m -1

เมื่อ

X_0 เป็นตัวเลขเริ่มต้น

a เป็นค่าคงที่ที่ใช้ในการคูณ

c เป็นค่าที่เพิ่มขึ้น

m เป็นตัว Modulus

ดังนั้นจะได้ตัวเลขสุ่ม

$$R_{i+1} = \frac{X_{i+1}}{m}$$

ตัวอย่าง a = 65539 c = 435 m = 10000 และ X_0 = 3579

$X_{i+1} = (a X_i + c) \bmod m$

$X_1 = (65539*3579 + 435) \bmod 10000$

$X_1 = 234564516 \bmod 10000 = 4516$

$R_1 = 4516/10000 = 0.4516$

$X_2 = (65539*4516 + 435) \bmod 10000$

$X_2 = 4559$

$R_2 = 4559/10000 = 0.4559$

.

.

.

โดยตัวเลขสุ่มที่ได้ต้องมีคุณสมบัติ 2 ข้อ ดังนี้

1. มีความสม่ำเสมอ (Uniformity) ทดสอบด้วยการทดสอบการแจกแจง (Distribution Test) ด้วยไคสแควร์ (Chi-Square Test) เปรียบเทียบการแจกแจงของตัวเลขที่ผลิตขึ้นกับการแจกแจงสม่ำเสมอ U(0,1)

2. มีความเป็นอิสระต่อกัน (Independence) ทดสอบด้วยการทดสอบรัน (Run Test)

2. ตัวแปรสุ่มแบบต่อเนื่องและตัวแปรสุ่มแบบไม่ต่อเนื่อง

ตัวแปรสุ่ม(Random Variable)

ตัวแปรสุ่มจำแนกได้ 2 ประเภท คือ

2.1 ตัวแปรสุ่มแบบต่อเนื่อง (Continuous Random Variable)

2.2 ตัวแปรสุ่มแบบไม่ต่อเนื่อง (Discrete Random Variable)

2.1 ตัวแปรสุ่มแบบต่อเนื่อง (Continuous Random Variable)

<u>นิยาม</u>

ตัวแปรสุ่มแบบต่อเนื่อง (Continuous Random Variable)

ให้ตัวแปรสุ่ม X เป็นตัวแปรสุ่มเราไม่อาจจะนับจำนวนค่าที่เป็นไปได้ของตัวแปรนั้น ค่าของตัวแปรสุ่มประเภทนี้จะกำหนดเป็นช่วง ๆ เช่น

$$X = \{x \mid 0 < x < 10\}$$

<u>นิยาม</u>

ตัวแปรสุ่ม X เป็นตัวแปรสุ่มแบบต่อเนื่องที่มีค่าอยู่ในช่วง (a,b) โดย $a < b$ $f(x)$ เรียกว่าฟังก์ชันความน่าจะเป็น ถ้า $f(x)$ มีคุณสมบัติดังนี้

1. $f(x) \geq 0$
2. $\int_a^b f(x)dx = 1$
3. ถ้า c และ d เป็นค่าในช่วง (a,b) และ $c < d$

$$P(c < X < d) = \int_c^d f(x)dx$$

<u>นิยาม</u>

ถ้าตัวแปรสุ่ม X เป็นตัวแปรสุ่มแบบต่อเนื่องที่มีค่าอยู่ในช่วง (a,b)

$F(x)$ จะเรียกว่าฟังก์ชันการแจกแจงสะสม ถ้า $F(x)$ กำหนดไว้ดังนี้

$$F(x) = P(X \leq x) = \int_a^x f(x)dx$$

คุณสมบัติของฟังก์ชันการแจกแจงสะสม

1. $0 \leq F(x) \leq 1$
2. ถ้า $x_1 < x_2$ แล้ว $F(x_1) \leq F(x_2)$
3. $f(x) = \frac{dF(x)}{dx}$
4. $P(a < X \leq b)$

   $P(a \leq X \leq b)$

$P(a \le X < b) \quad = F(b) - F(a)$
$P(a < X < b)$

5. $P(X=a)=0$
6. ฟังก์ชันการแจกแจงสะสมเป็นฟังก์ชันที่ขึ้นทางขวาอย่างเดียว

2.2 ตัวแปรสุ่มแบบไม่ต่อเนื่อง (Discrete Random Variable)

นิยาม

ตัวแปรสุ่มแบบไม่ต่อเนื่อง (Discrete Random Variable)

ให้ตัวแปรสุ่ม X เป็นตัวแปรสุ่ม ถ้าค่าที่เป็นไปได้ของ X มีจำนวนจำกัด (Finite)

นับจำนวนได้เรียก X ว่า ตัวแปรสุ่มไม่ต่อเนื่อง (Discrete Random Variable) กล่าวคือ ตัวแปรสุ่ม $X = \{x/x=x_1, x_2, x_3,...\}$

นิยาม

ตัวแปรสุ่ม X เป็นตัวแปรสุ่มแบบไม่ต่อเนื่องที่มีค่าเป็นจำนวนนับได้และมีค่าเป็น $x_1, x_2, x_3,...$ ฟังก์ชัน p เรียกว่า ฟังก์ชันความน่าจะเป็นของ X ถ้า $p(x_i)$ กำหนดไว้ดังนี้

$$p(x_i) = P(X=x_i) \; ; \; i = 1,2,3,...$$

โดยที่ $p(x_i)$ ต้องมีคุณสมบัติ ดังนี้

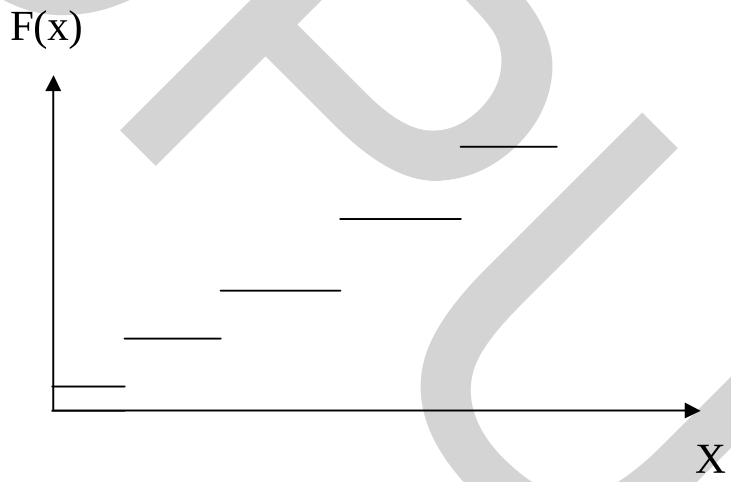
1. $0 \le p(x_i) \le 1$
2. $\sum_{x_i} p(x_i) = 1$

<u>นิยาม</u> ให้ X เป็นตัวแปรสุ่มแบบไม่ต่อเนื่อง ฟังก์ชัน F(x) เรียกว่าฟังก์ชันการแจกแจงสะสมของ X ถ้า F(x) กำหนดไว้ดังนี้

$$F(x) = P(X \le x) = \sum_{x_i \le x} p(x_i) \quad \text{ซึ่ง } x_i \le x$$

คุณสมบัติของฟังก์ชันการแจกแจงสะสม

1. $0 \le F(x) \le 1$
2. ถ้า $x_1 < x_2$ แล้ว $F(x_1) \le F(x_2)$
3. ถ้า $x_1 < x_2 < \ldots < x_{i-1} < x_i$ แล้ว $P(X=x_i)=F(x_i) - F(x_{i-1})$
4. $P(c < X \le d)=F(d)-F(c)$
   $P(c \le X \le d)=F(d)-F(c)+P(X=c)$
   $P(c \le x < d) = F(d)-F(c)+P(X=c)-P(X=d)$
   $P(c < x < d) = F(d)-F(c)-P(X=d)$
5. ฟังก์ชันการแจกแจงสะสมเป็นฟังก์ชันที่ขึ้นทางขวาแบบขั้นบันได



3. ค่าคาดหวังและความแปรปรวน (Expected Value and Variance)

<u>นิยาม</u> ให้ X เป็นตัวแปรสุ่ม E(X) เรียกว่า ค่าคาดหวังของ X ถ้า

$$E(X) = \sum_{\forall x} x f(x) = \mu$$ X เป็นตัวแปรสุ่มแบบไม่ต่อเนื่อง

$$E(X) = \int_a^b x f(x)\,dx$$ X เป็นตัวแปรสุ่มแบบต่อเนื่อง

มีค่า $a < x < b$

<u>นิยาม</u> ให้ X เป็นตัวแปรสุ่ม E(X) = μ เป็นค่าคาดหวังของ X เราเรียก V(X)=σ^2 ว่า

ความแปรปรวนของ X ถ้า

$$V(X) = E(X^2) - [E(X)]^2$$

$E(X^2) = \sum_{\forall x} x^2 f(x)$ X เป็นตัวแปรสุ่มแบบไม่ต่อเนื่อง

$E(X^2) = \int_a^b x^2 f(x)dx$ X เป็นตัวแปรสุ่มแบบต่อเนื่อง

มีค่า $a < x < b$

$S(X) = \sqrt{V(X)}$ S(X) เป็นส่วนเบี่ยงเบนมาตรฐานของ X

4. วิธีการสร้างค่าตัวแปรสุ่ม

ในการศึกษานี้ใช้ 2 วิธีของการสร้างค่าตัวแปรสุ่มตามการแจกแจงที่ต้องการทดสอบ คือ

1. วิธีการแปลงผกผัน (The Inverse Transform Method) ใช้สำหรับการสร้างค่าตัวแปรสุ่มของการแจกแจงแบบเอ๊กซ์โปเนนเชียล(Exponential Distribution) การแจกแจงแบบพัวซอง(Poisson Distribution) และการแจกแจงแบบทวินาม (Binomial Distribution)

2. วิธีการปฏิเสธ (The Rejection Method) ใช้สำหรับสร้างค่าตัวแปรสุ่มของการแจกแจงแบบปกติ(Normal Distribution)

1. วิธีการแปลงผกผัน (The Inverse Transform Method)

วิธีนี้สะดวกเนื่องจากถ้าเราทราบรูปแบบฟังก์ชันการแจกแจงสะสม(Cumulative Distribution Function) ของตัวแปรสุ่มนั้นแล้วนำฟังก์ชันการแจกแจงสะสมนี้มาเท่ากับตัวเลขสุ่มที่มีการแจกแจงแบบสม่ำเสมอ (0,1) นั้นคือ ฟังก์ชันการแจกแจงสะสมต้องมีการแจกแจงแบบสม่ำเสมอ (0,1) เช่นกัน

<u>**ทฤษฎี**</u> ฟังก์ชันการแจกแจงสะสม(Cumulative Distribution Function) ของตัวแปรสุ่ม X คือ F(x) จะมีการแจกแจงแบบสม่ำเสมอ (0,1)

เมื่อ F(X) มีการแจกแจงแบบสม่ำเสมอ และ R (Random Number) ก็มีการแจกแจงแบบสม่ำเสมอ จึงสามารถใช้วิธีการแปลงผกผันหาค่าตัวแปรสุ่ม ได้ดังนี้

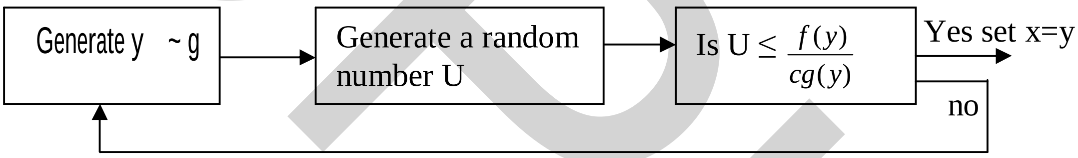
$$F(X) = R$$
$$X = F^{-1}(R)$$

2. วิธีการปฏิเสธ (The Rejection Method)

วิธีการในการสร้างค่าตัวแปรสุ่มที่มีฟังก์ชันความหนาแน่น $g(x)$ เราสามารถใช้ในการสร้างจากฟังก์ชั่น $f(x)$ โดยการสร้าง y จากฟังก์ชัน g หลังจากนั้นสร้างค่าความน่าจะเป็นด้วยสัดส่วน $\frac{f(y)}{g(y)}$ โดยให้ c เป็นค่าคงที่

$$\frac{f(y)}{g(y)} \leq c \quad \text{สำหรับทุกค่าของ } y$$

ดังแผนภาพการสร้างค่าตัวแปรสุ่ม x ที่มีฟังก์ชันหนาแน่น f ด้วยวิธีการปฏิเสธ ดังนี้



มีขั้นตอนการทำงาน ดังนี้

ขั้นตอนที่ 1 สร้าง y ที่มีความหนาแน่น g

ขั้นตอนที่ 2 สร้างตัวเลขสุ่ม U

ขั้นตอนที่ 3 ถ้า $U \leq \frac{f(y)}{cg(y)}$, ให้ $x = y$ ถ้าไม่ใช่กลับไปที่ขั้นตอนที่ 1

5. ฟังก์ชันการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่องและการสร้างค่าตัวแปรสุ่มของแต่ละการแจกแจง

ในการศึกษานี้ศึกษาถึงการแจกแจงของตัวแปรสุ่มแบบต่อเนื่อง 2 ประเภท คือ

1. การแจกแจงแบบเอ๊กซ์โปเนนเชียล(Exponential Distribution)
2. การแจกแจงแบบปกติ(Normal Distribution)

ส่วนการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่องศึกษาจาก 2 ประเภทของการแจกแจง คือ

1. การแจกแจงแบบพัวซอง(Poisson Distribution)
2. การแจกแจงแบบทวินาม(Binomial Distribution)

สำหรับการสร้างค่าตัวแปรสุ่มในการศึกษาครั้งนี้ใช้วิธีการแปลงผกผัน(The Inverse Transform Method) สำหรับการแจกแจงแบบเอ๊กซ์โปเนนเชียล การแจกแจงแบบพัวซอง และการแจกแจงแบบทวินาม ส่วนการแจกแจงแบบปกติใช้วิธีการปฏิเสธ(The Rejection Method) ในการสร้างค่าตัวแปรสุ่ม

การแจกแจงแบบเอ๊กซ์โปเนนเชียล(Exponential Distribution)

มีฟังก์ชันหนาแน่นความน่าจะเป็น(Probability density function) ดังนี้

$$f(x) = \begin{cases} \frac{1}{\beta} \cdot e^{-\frac{x}{\beta}} & \text{ถ้า } x \geq 0 \\ 0 & \text{อื่น ๆ} \end{cases}$$

$E(X) = \beta$ $Var(X) = \beta^2$

เมื่อใช้วิธีการแปลงผกผันในการสร้างค่าตัวแปรสุ่มหาค่าตัวแปรสุ่มดังนี้

$$F(x) = R$$
$$x = F^{-1}(R)$$

$$F(x) = P(X \leq x)$$
$$= \int_0^x f(x)dx = \int_0^x \frac{1}{\beta} e^{-\frac{x}{\beta}} dx$$
$$F(x) = 1- e^{-\frac{x}{\beta}}$$
$$F(x) = R$$
$$1- e^{-\frac{x}{\beta}} = R$$
$$e^{-\frac{x}{\beta}} = R$$
$$x = -\beta * \ln (R)$$

และมีขั้นตอนการทำงาน (Algorithm) ดังนี้

ขั้นตอนที่ 1 สร้างตัวเลขสุ่ม U

ขั้นตอนที่ 2 $x = -\beta * \ln (R)$

ขั้นตอนที่ 3 Return

การแจกแจงแบบปกติ (Normal Distribution)

ตัวแปรสุ่ม X มีการแจกแจงแบบปกติ มีค่าเฉลี่ย (Mean) เท่ากับ μ และความแปรปรวน (Variance) เท่ากับ σ^2 โดยมีฟังก์ชันหนาแน่นความน่าจะเป็น(Probability density function) ดังนี้

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}}.e^{-(x-\mu)^2/2\sigma^2} \quad , -\infty < x < \infty$$

$$E(X) = \mu \qquad Var(X) = \sigma^2$$

สำหรับการแจกแจงแบบปกติในการศึกษานี้ใช้วิธีการปฏิเสธ(The Rejection Method)ในการสร้างค่าตัวแปรสุ่ม ดังนี้

การสร้างค่าตัวแปรสุ่มที่มีการแจกแจงแบบปกติ (Generating a Normal Random Variable)

การสร้างตัวแปรสุ่มที่มีการแจกแจงแบบปกติมาตรฐาน Z~N(0,1) นั้น เมื่อค่าสัมบูรณ์ของ Z มีฟังก์ชันหนาแน่นความน่าจะเป็น (Probability density function)

$$f(x)= \frac{2}{\sqrt{2\pi}}.e^{-\frac{x^2}{2}} \qquad ; \quad 0 < x < \infty$$

เมื่อให้ g คือ ฟังก์ชันหนาแน่นน่าจะเป็นแบบเอ๊กซ์โปเนนเชียลมีค่าเฉลี่ยเท่ากับ 1 นั่นคือ

$$g(x) = e^{-x} \qquad ; \quad 0 < x < \infty$$

ดังนั้น

$$\frac{f(x)}{g(x)} = \sqrt{2/\pi e^{x-\frac{x^2}{2}}}$$

ค่าสูงสุดของ $\frac{f(y)}{g(y)}$ เกิดขึ้นเมื่อ x ทำให้ x-$\frac{x^2}{2}$ มีค่าสูงสุด เมื่อ x=1

ดังนั้น

$$c = Max\ \frac{f(y)}{g(y)} = \frac{f(1)}{g(1)} = \sqrt{2e/\pi}$$

เพราะว่า $\frac{f(y)}{cg(y)} = \exp\{x-\frac{x^2}{2}-\frac{1}{2}\}$

$$= \exp\{-\frac{(x-1)^2}{2}\}$$

ดังนั้นสามารถสร้างค่าตัวแปรสุ่มที่มีการแจกแจงแบบปกติมาตรฐาน Normal(0,1) ตามขั้นตอน ดังนี้

ขั้นตอนที่ 1 สร้าง y ที่มีการแจกแจงแบบเอ๊กซ์โปเนนเชียลมีค่าเฉลี่ยเท่ากับ 1

ขั้นตอนที่ 2 สร้างตัวเลขสุ่ม U

ขั้นตอนที่ 3 ถ้า $U \le \exp\{-{(y-1)^2}/{2}\}$,ให้ x=y ถ้าไม่ใช่กลับไปที่ขั้นตอนที่ 1

ดังนั้นเราสร้างตัวแปรสุ่ม x ที่มีฟังก์ชันหนาแน่นดังสมการ

$$f(x)= \frac{2}{\sqrt{2\pi}}.e^{-\frac{x^2}{2}} \quad ; \quad 0 < x < \infty$$

ที่มีการแจกแจงแบบปกติมาตรฐาน ดังนั้นจะได้ z~N(0,1) โดยให้ z = x หรือ –x และในขั้นตอนที่ 3 ค่าของ x=y ถ้า $U \le \exp\{-{(y-1)^2}/{2}\}$ ซึ่งมีค่าเท่ากับ log U $\ge {(y-1)^2}/{2}$ ดังนั้นสามารถเขียนขั้นตอนการทำงานใหม่ได้ดังนี้

ขั้นตอนที่ 1 สร้างตัวแปรสุ่มที่มีการแจกแจงแบบเอ๊กซ์โปเนนเชียลที่มีค่าเฉลี่ยเท่ากับ 1 y_1 ,y_2

ขั้นตอนที่ 2 ถ้า $y_2 \ge {(y-1)^2}/{2}$, ให้ $x = y_1$ ถ้าไม่ใช่กลับไปที่ขั้นตอนที่ 1

ถ้าขณะนี้ได้ $x= y_1$ และเรารู้ว่า $y_2 \ge {(y-1)^2}/{2}$ ซึ่งมากกว่าอยู่เท่าใดนั้น คำตอบก็คือให้ y_2 มีการแจกแจงแบบเอ็กซ์โปเนนเชียลที่มีค่าเฉลี่ยเท่ากับ 1 และให้มีค่ามากกว่าค่าใด ๆ จำนวนที่ y_2 มากกว่า ${(y-1)^2}/{2}$ ก็จะมีการแจกแจงแบบเอ็กซ์โปเนนเชียลที่มีค่าเฉลี่ยเท่ากับ 1 นั่นคือ เมื่อเราผ่านในขั้นตอนที่ 2 ไม่เพียงแต่เราจะได้ x (ค่าสัมบูรณ์ของการแจกแจงแบบปกติมาตรฐาน) แต่จาก

$y_2 - {(y-1)^2}/{2}$ เราสามารถสร้างค่าตัวแปรสุ่มแบบเอ็กซ์โปเนนเชียลที่มีค่าเฉลี่ยเท่ากับ 1 ได้

ดังนั้นโดยสรุปขั้นตอนการสร้างค่าตัวแปรที่มีการแจกแจงแบบเอ็กซ์โปเนนเชียลที่มีค่าเฉลี่ยเท่ากับ 1 และการสร้างค่าตัวแปรสุ่มที่มีการแจกแจงแบบปกติมาตรฐาน ดังนี้

ขั้นตอนที่ 1 สร้าง y_1 ที่มีการแจกแจงแบบเอ็กซ์โปเนนเชียลที่มีค่าเฉลี่ยเท่ากับ 1

ขั้นตอนที่ 2 สร้าง y_2 ที่มีการแจกแจงแบบเอ็กซ์โปเนนเชียลที่มีค่าเฉลี่ยเท่ากับ 1

ขั้นตอนที่ 3 ถ้า $y_2 - {(y-1)^2}/{2} > 0$, ให้ $y = y_2 - {(y-1)^2}/{2}$ และไปที่ขั้นตอนที่ 4 ถ้าไม่ใช่กลับไปที่ขั้นตอนที่ 1

ขั้นตอนที่ 4 สร้างตัวเลขสุ่ม U และกำหนด

$$Z = \begin{cases} \dfrac{y_1 - \mu}{\sigma} & ; \quad U \le \dfrac{1}{2} \\[2ex] \dfrac{-y_1 - \mu}{\sigma} & ; \quad U > \dfrac{1}{2} \end{cases}$$

การแจกแจงแบบพัวซอง(Poisson Distribution)

มีฟังก์ชันหนาแน่นความน่าจะเป็น(Probability density function) ดังนี้

$$p_i = P\{X=i\} = e^{-\lambda}\frac{\lambda^i}{i!} \qquad ; i = 0,1,...$$

$$E(X) = Var(X) = \lambda$$

เมื่อใช้วิธีการแปลงผกผัน (Inverse Transform) ในการสร้างค่าตัวแปรสุ่มแบบพัวซองที่มีค่าเฉลี่ยเท่ากับ λ โดยใช้สมการดังนี้

$$p_{i+1} = \frac{\lambda}{i+1}p_i \quad , \quad i \ge 0$$

โดยมีขั้นตอนการทำงาน (Algorithm) ดังนี้

ขั้นตอนที่ 1 สร้างตัวเลขสุ่ม U

ขั้นตอนที่ 2 ให้ i = 0 , $p = e^{-\lambda}$, F = p

ขั้นตอนที่ 3 ถ้า U < F ให้ X = i และหยุด

ขั้นตอนที่ 4 $p = \frac{\lambda}{i+1}p_i$, F = F + p , i = i + 1

ขั้นตอนที่ 5 กลับไปที่ขั้นตอนที่ 3

จาก 5 ขั้นตอนในการสร้างค่าตัวแปรสุ่มที่มีการแจกแจงแบบพัวซองนั้น $p = p_i$ เป็นค่าความน่าจะเป็นที่ X = i และ F = F(i) เป็นความน่าจะเป็นที่ X น้อยกว่าหรือเท่ากับ i!

การแจกแจงแบบทวินาม(Binomial Distribution)

เมื่อต้องการสร้างค่าตัวแปรสุ่มที่มีการแจกแจงแบบทวินามที่มีพารามิเตอร์ n,p โดยที่มีฟังก์ชันหนาแน่นความน่าจะเป็น(Probability density function) ดังนี้

$$P\{X=i\} = \frac{n!}{i!(n-i)!}p^i(1-p)^{n-i} \ , \ i = 0,1,....,n$$

เมื่อใช้วิธีการแปลงผกผันโดยการกระทำซ้ำ(Loop) เมื่อ pr=P{X=i} เป็นค่าความน่าจะเป็นที่ X เท่ากับ i และ F = F(i) เป็นค่าความน่าจะเป็นที่ X น้อยกว่าหรือเท่ากับ i โดยการทำซ้ำ(Loop) ตามสมการและขั้นตอนการทำงาน (Algorithm) ดังนี้

$$P\{X=i+1\}=\frac{n-i}{i+1}.\frac{p}{1-p}.P\{X=i\}$$

ขั้นตอนที่ 1 สร้างตัวเลขสุ่ม U

ขั้นตอนที่ 2 $c = \frac{p}{(1-p)}$, i=0 , pr = $(1-p)^n$, F = pr

ขั้นตอนที่ 3 ถ้า U < F , ให้ X = i และหยุด

ขั้นตอนที่ 4 pr = [c(n-i)/(i+1)]pr , F = F + pr, i = i+1

ขั้นตอนที่ 5 กลับไปที่ขั้นตอนที่ 3

6. วิธีการประมาณค่า

ในการศึกษาวิจัยนี้ใช้วิธีการประมาณค่า 2 วิธี คือ การประมาณค่าแบบสภาวะสูงสุด (Maximum Likelihood Estimation) และการประมาณค่าด้วยวิธีกำลังสองน้อยที่สุด(Method of Least Square) ซึ่งในแต่ละวิธีมีทฤษฎีและวิธีการดังนี้

6.1 การประมาณค่าแบบสภาวะสูงสุด(Maximum Likelihood Estimation)

การประมาณค่าพารามิเตอร์ด้วยวิธีภาวะน่าจะเป็นสูงสุด (เป็นวิธีที่ใช้กันแพร่หลายมากที่สุด มีแนวความคิดมานานตั้งแต่คริสต์ศตวรรษที่ 18 คาร์ล ฟรีดริค เกาส์ (Karl Friedrich Gauss,1777-1855) และแดเนียล เบอร์นูลลี่ (Daniel Bernoulli) ได้ใช้วิธีการนี้มาแล้วต่อมาในต้นศตวรรษที่ 20 โรนัลด์ ไอล์เมอร์ ฟิเชอร์ (Ronald Aylmer Fisher 1890-1962) ได้ทำการศึกษาคุณสมบัติของวิธีการนี้ ทำให้มีผู้ใช้กันกว้างขวางขึ้น และถือว่าวิธีการนี้เป็นผลงานของฟิเชอร์ โดยเขาได้นำเสนอผลงานเกี่ยวกับวิธีการนี้ในปี ค.ศ.1912 พร้อมทั้งมีการปรับปรุงแก้ไขส่วนที่เกี่ยวข้องให้เหมาะสมขึ้นอีกด้วย นักสถิติคนอื่น ๆก็มีส่วนทำให้วิธีการนี้เป็นที่นิยมแพร่หลายยิ่งขึ้นด้วย

นิยาม ให้ $x_1,\ldots,x_n$ เป็นตัวอย่างสุ่มจากประชากรที่มีฟังก์ชันความหนาแน่น $f(x;\theta)$, $\theta \in \Omega$ ฟังก์ชันภาวะน่าจะเป็น(Likelihood Function) ของตัวอย่างสุ่ม คือ ฟังก์ชันความหนาแน่นร่วม $L = L(\theta;x_1,x_2,\ldots,x_n)=L(\theta)$ ของตัวอย่างสุ่มนั้นที่ถือว่าเป็นฟังก์ชันของพารามิเตอร์ θ นั่นคือ

$$L(\theta;x_1,x_2,\ldots,x_n) = \prod_{i=1}^{n} f(x_i;\theta)$$

นิยาม ค่าของพารามิเตอร์ θ ในเทอมของค่าสังเกตของตัวอย่างสุ่ม $x_1,x_2,...,x_n$ ที่ทำให้ฟังก์ชันภาวะน่าจะเป็นมีค่าสูงสุด เรียกว่า ตัวประมาณแบบภาวะน่าจะเป็นสูงสุด (Maximum Likelihood Estimator-MLE) ของ θ นั่นคือค่าของ θ คือ $\hat{\theta}=\hat{\theta}(x_1,...,x_n)$ เป็น MLE ของ θ ก็ต่อเมื่อ $L(\hat{\theta})=L(\hat{\theta}(x_1,...,x_n))$ มีค่าสูงสุด

วิธีการหาตัวประมาณแบบภาวะน่าจะเป็นสูงสุด

เป็นวิธีการหาค่าของพารามิเตอร์ θ ที่ทำให้ฟังก์ชัน $L(\theta)$ สูงสุด ในการนี้มีข้อควรสังเกตดังต่อไปนี้

1. เป้าหมายในการหา MLE ของ θ คือ การหาค่า θ เรียกว่า $\hat{\theta}=\hat{\theta}(x_1,...,x_n)$ ที่ทำให้
   $L(\hat{\theta};\ x_1,...,x_n) \geq L(\theta;x_1,x_2,\ldots,x_n)$
   เมื่อ $\theta \in \Omega$ และ $X_1=x_1,...,X_n=x_n$
2. ถ้าฟังก์ชันภาวะน่าจะเป็น $L(\theta)$ เป็นฟังก์ชันที่หาอนุพันธ์ได้(Differentiable function) เมื่อเทียบกับ θ อาจใช้อนุพันธ์หา MLE ของ θ ได้ เมื่อเรนจ์ของ $f(x;\ \theta)$ ไม่ขึ้นอยู่กับ θ และ θ อยู่ในช่วงจำนวนจริงช่วงหนึ่งในกรณีดังกล่าว $\hat{\theta}$ คือ รากของสมการ

   $$\frac{\partial L}{\partial \theta} = 0$$

   เงื่อนไขพอเพียง (Sufficient Condition) ที่ $\hat{\theta}$ ทำให้ $L(\hat{\theta}) \geq L(\theta), \theta \in \Omega$ คือ

   $$\frac{\partial^2 L}{\partial \theta^2} < 0$$

   เมื่อ $\theta = \hat{\theta}$
3. การใช้อนุพันธ์หา MLE ในหลายกรณีใช้ $\ln L$ จะสะดวกกว่าที่จะใช้ L
   ควรสังเกตว่า

   $$\frac{\partial \ln L}{\partial \theta} = \frac{1}{L}.\frac{\partial L}{\partial \theta}$$

   และ $L > 0$ ดังนั้นเมื่อให้

   $$\frac{\partial \ln L}{\partial \theta} = 0$$

   เราจะได้ $\frac{\partial L}{\partial \theta} = 0$ ดังนั้น นอกจากนั้น เมื่อ

   $\frac{\partial^2 \ln L}{\partial \theta^2} < 0$ ก็จะทำให้ $\frac{\partial^2 L}{\partial \theta^2} < 0$ ด้วย

นิยาม สมการที่ใช้หา MLE คือ $\frac{\partial L}{\partial \theta} = 0$ หรือ $\frac{\partial \ln L}{\partial \theta} = 0$ เราเรียกว่า สมการภาวะน่าจะเป็น (Likelihood equation)

การหาตัวประมาณค่าพารามิเตอร์ของการแจกแจงแบบทวินาม (Binomial Distribution)

ให้ X_i ~ Bernulli เมื่อ i = 1,...,n

L(p ; X) = $p^{x_1}(1-p)^{1-x_2}.p^{x_2}(1-p)^{1-x_2}.....p^{x_n}(1-p)^{1-x_n}$

L(p ; x_i) = $p^{\sum x_i}(1-p)^{n-\sum x_i}$

ln L = $\sum X_i \ln p + (n - \sum X_i).\ln(1-p)$

$$\frac{\partial \ln L}{\partial p} = \frac{\sum X_i}{p} + \frac{n - \sum X_i}{1-p}(-1) = 0$$

$$\frac{\sum X_i}{p} - \frac{n - \sum X_i}{1-p} = 0$$

$$\frac{\sum X_i}{p} = \frac{n - \sum X_i}{1-p}$$

$$\sum X_i - p\sum X_i = np - p\sum X_i$$

$$\overline{X} = \frac{\sum X_i}{n} = p$$

ดังนั้น ตัวประมาณค่าพารามิเตอร์ p ของการแจกแจงแบบทวินาม คือ $\overline{X}$

การหาตัวประมาณค่าพารามิเตอร์ของการแจกแจงแบบพัวซอง (Poisson Distribution)

ให้ X_i ~ Poisson เมื่อ i = 1,...,n

โดย X_i มีฟังก์ชันการแจกแจง ดังนี้

f($x_i, \lambda) = \frac{e^{-\lambda}\lambda^{x}}{x!}; x = 0,1,2,\ldots,$

$$L = f(x_1,\lambda).f(x_2,\lambda).f(x_3,\lambda)\ldots f(x_n,\lambda)$$

$$= \frac{e^{-\lambda}.\lambda^{x_1}}{x_1!} \times \frac{e^{-\lambda}.\lambda^{x_2}}{x_2!} \times \ldots \times \frac{e^{-\lambda}.\lambda^{x_n}}{x_n!}$$

$$= \frac{e^{-n\lambda}.\lambda^{\sum x_i}}{\coprod x_i!}$$

$$\ln L = -n\lambda + \sum x_i \ln \lambda + \ln \prod x_i!$$

$$\frac{\partial \ln L}{\partial \lambda} = -n + \frac{\sum x_i}{\lambda} = 0$$

$$\frac{\sum x_i}{\lambda} = n$$

$$\lambda = \frac{\sum x_i}{n}$$

ดังนั้นตัวประมาณค่าพารามิเตอร์ λ ของการแจกแจงแบบพัวซอง คือ $\overline{X}$

การหาตัวประมาณค่าพารามิเตอร์ของการแจกแจงแบบเอ๊กซ์โปเนนเชียล (Exponential Distribution)

ให้ X_i ~ Exponential เมื่อ i = 1,...,n

โดย X_i มีฟังก์ชันการแจกแจง ดังนี้

$$f(x_i,\beta) = \frac{1}{\beta}.e^{-\frac{x}{\beta}}; x_i > 0,$$

$$L = f(x_1,\beta).f(x_2,\beta).f(x_3,\beta)...f(x_n,\beta)$$

$$= \frac{1}{\beta}.e^{-\frac{x_1}{\beta}} \times \frac{1}{\beta}.e^{-\frac{x_2}{\beta}} \times \frac{1}{\beta}.e^{-\frac{x_3}{\beta}} \times ... \times \frac{1}{\beta}.e^{-\frac{x_n}{\beta}}$$

$$= \frac{1}{\beta^n}.e^{-\frac{1}{\beta}\sum x_i}$$

$$\ln L = -n\ln\beta - \frac{1}{\beta}\sum x_i$$

$$\frac{\partial \ln L}{\partial \beta} = -\frac{n}{\beta} + \frac{\sum x_i}{\beta^2} = 0$$

$$\frac{\sum x_i}{\beta^2} = \frac{n}{\beta}$$

$$\frac{\sum x_i}{n} = \frac{\beta^2}{\beta}$$

$$\overline{X} = \frac{\sum x_i}{n} = \beta$$

ดังนั้นตัวประมาณค่าพารามิเตอร์ β ของการแจกแจงแบบเอ๊กซ์โปเนนเชียล คือ $\overline{X}$

การหาตัวประมาณค่าพารามิเตอร์ของการแจกแจงแบบปกติ (Normal Distribution)

ให้ X_i ~ Normal(μ,σ^2) เมื่อ i = 1,...,n

โดย X_i มีฟังก์ชันการแจกแจง ดังนี้

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}}.e^{-(x-\mu)^2/2\sigma^2} \quad , -\infty < x < \infty$$

$$L(\mu,\sigma^2;x_1,x_2,...,x_n)=\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}}.e^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2}$$

$$\ln L(\mu,\sigma^2)=-\frac{n}{2}\ln 2\pi-\frac{n}{2}\ln\sigma^2-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

ดังนั้น $\frac{\partial \ln L(\mu,\sigma^2)}{\partial\mu}=\frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i-\mu)=0$

และ $\frac{\partial \ln L(\mu,\sigma^2)}{\partial\sigma^2}=-\frac{n}{2\sigma^2}+\frac{1}{2\sigma^4}\sum_{i=1}^{n}(x_i-\mu)^2=0$

จะได้ $\mu=\frac{1}{n}\sum_{i=1}^{n}x_i=\overline{X}$ และ $\sigma^2=\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2$

นั่นคือ MLE ของ μ คือ $\hat{\mu}=\frac{1}{n}\sum_{i=1}^{n}x_i=\overline{X}$

และ MLE ของ σ^2 คือ $\hat{\sigma}^2=\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2=S^2$

ในที่นี้ $\hat{\sigma}^2$ ขึ้นอยู่กับ μ จะต้องประมาณ μ ก่อนจึงจะประมาณ σ^2 ได้ ในกรณีเช่นนี้มักเรียก μ ว่า พารามิเตอร์รบกวน (Nuisance parameter)

6.2 การประมาณค่าพารามิเตอร์ด้วยวิธีกำลังสองน้อยที่สุด (Method of Least Square)

แนวความคิดและวิธีการหาตัวประมาณแบบกำลังสองน้อยที่สุด วิธีนี้เป็นวิธีการสำคัญมีรากฐานมาจากทฤษฎีการประมาณเชิงเส้น(Theory of Linear estimation) โดยไม่จำเป็นต้องทราบรูปการแจกแจงความน่าจะเป็นแต่อาศัยผลต่างระหว่างค่าสังเกตและค่าคาดหมายเป็นสำคัญ วิธีการนี้คิดขึ้นโดยคาร์ล ฟรีดริค เกาส์ (Karl Friedrich Gauss, 1777-1855) และอังเดร อังเดรเยวิช มาร์โคฟ (Andrei Andreevich Markov, 1856-1922)

นิยาม ให้ $X_1,\ldots,X_n$ เป็นตัวอย่างสุ่มจากประชากรหนึ่งที่มีค่าคาดหมาย E(X) ค่าของพารามิเตอร์ในเทอมของค่าสังเกตที่ทำให้ผลบวกของกำลังที่สองของผลต่างระหว่างค่าสังเกตกับค่าคาดหมายต่ำสุดจะเรียกว่า ตัวประมาณแบบกำลังสองน้อยที่สุด (Least Squares estimator) ของพารามิเตอร์ที่เกี่ยวข้อง

นั่นคือ ตัวประมาณแบบกำลังสองน้อยที่สุดของพารามิเตอร์ θ (อาจเป็นเวคเตอร์)ที่ปรากฏอยู่ในค่าคาดหมาย E(X) ได้แก่ ค่าของ θ ในเทอมของค่าสังเกตที่ทำให้

$$Q=\sum_{i=1}^{n}(x_i-E(x_i))^2$$ ต่ำที่สุด

มักหาตัวประมาณดังกล่าว โดยการใช้อนุพันธ์กล่าวคือ ใช้

$$\frac{\partial Q}{\partial \theta} = 0$$

และเรียกสมการนี้ว่า สมการปกติ (Normal equation(s))

คุณสมบัติของตัวประมาณแบบกำลังสองน้อยที่สุด

ตัวประมาณแบบกำลังสองน้อยที่สุดมีคุณสมบัติที่สำคัญหลายประการ โดยเฉพาะในตัวแบบเชิงเส้น ดังต่อไปนี้

1. เมื่อ $\underline{Y} = X\underline{\theta} + \in$ ที่ $E(\in) = \underline{0}$ ตัวประมาณแบบกำลังสองน้อยที่สุดของ $\underline{\theta}$ เป็นตัวประมาณไม่เอนเอียง

   พิสูจน์ จาก $\underline{\hat{\theta}} = (X'X)^{-1}X'\underline{Y} = (X'X)^{-1}X'(X\underline{\theta} + \in)$

   $= \underline{\theta} + (X'X)^{-1}X'\in$

   ดังนั้น $E(\hat{\theta}) = \underline{\theta}$ เนื่องจาก $E(\in) = \underline{0}$

2. เมื่อ $\underline{Y} = X\underline{\theta} + \in, E(\in) = \underline{0}, V(\in) = \sigma^2 I$ เมตริกซ์ของความแปรปรวนและความแปรปรวนร่วมของตัวประมาณแบบกำลังสองน้อยที่สุดของ $\underline{\theta}$ คือ $(X'X)^{-1}.\sigma^2$

   พิสูจน์ $V(\underline{\hat{\theta}}) = E[(\underline{\hat{\theta}} - \underline{\theta})(\underline{\hat{\theta}} - \underline{\theta})']$

   $= E[((X'X)^{-1}X'\in)(\in'X(X'X)^{-1})]$

   $= (X'X)^{-1}X'(\sigma^2 I)X(X'X)^{-1}$

3. ในตัวแบบเชิงเส้น $\underline{Y} = X\underline{\theta} + \in$ ที่ $E(\in) = \underline{0}, E(\in\in') = \sigma^2 I$
   ตัวประมาณแบบกำลังสองน้อยที่สุด $\underline{\hat{\theta}}$ ของ $\underline{\theta}$ เป็นตัวประมาณที่มีความแปรปรวนต่ำสุดในบรรดาตัวประมาณที่ไม่เอนเอียงของ $\underline{\theta}$
   คุณสมบัตินี้ คือ ทฤษฎีของเกาส์และมาร์โคฟ

ทฤษฎีบทที่ 1 (เกาส์และมาร์โคฟ) เมื่อ

$$\underline{Y} = X\underline{\theta} + \in \ , E(\in) = 0 \ , E(\in\in') = \sigma^2 I \ , E(\underline{Y}) = X\underline{\theta}$$

ตัวประมาณแบบกำลังสองน้อยที่สุด $\underline{\hat{\theta}} = (X'X)^{-1}X'\underline{Y}$ เป็นตัวประมาณที่เป็นฟังก์ชันเชิงเส้นไม่เอนเอียงและดีที่สุดของ $\underline{\theta}$

พิสูจน์ ให้ $\underline{\hat{\theta}^*} = [(X'X)^{-1}X' + A]\underline{Y}$ เป็นฟังก์ชันเชิงเส้นของ $\underline{Y}$ ที่ใช้ประมาณ $\underline{\theta}$ จะหาเมตริกซ์ A ที่ทำให้ $\underline{\hat{\theta}^*}$ เป็นตัวประมาณไม่เอนเอียงของ $\underline{\theta}$ ที่มีส่วนประกอบแต่ละตัวมีความแปรปรวนต่ำที่สุดในบรรดาตัวประมาณไม่เอนเอียงของส่วนประกอบของ $\underline{\theta}$

จาก $E(\underline{\hat{\theta}^*}) = \underline{\theta}$ จะได้

$\underline{\theta} = [(X'X)'X' + A]E(\underline{Y}) = [(X'X)^{-1}X' + A]\,X\underline{\theta} = \underline{\theta} + AX\underline{\theta}$

ดังนั้น $AX\underline{\theta} = \underline{\theta}$ หรือ $AX = 0$

$$V(\underline{\hat{\theta}}^*) = E[(\underline{\hat{\theta}}^* - \underline{\theta})(\underline{\hat{\theta}}^* - \underline{\theta})']$$
$$= E[((X'X)'X' + A)\underline{Y} - \underline{\theta}][((X'X)^{-1}X' + A)Y - \underline{\theta}]'$$

$=E[(X'X)^{-1}X'\underline{\in\in}'X(X'X)^{-1} + A\underline{\in\in}'A' + (X'X)^{-1}X'\underline{\in\in}'A' + A\underline{\in\in}X(X'X)^{-1}]$

$=[(X'X)^{-1} + AA']\sigma^2$

ให้ $AA' = B = (b_{ij})$ เมตริกซ์บวกกำหนดแน่(Positive definite matrix) จึงมี $b_{ij} \geq 0$ ทุกค่า i=1,...,p

ต้องการหา B ที่มีสมาชิกบนเส้นทแยงมุมหลักมีค่าน้อยที่สุด เพื่อทำให้ $V(\hat{\theta}^*_i)$ มีสมาชิกบนเส้นทแยงมุมหลักต่ำที่สุด เมื่อ $b_{ii} = \sum_{j=1}^{n} a^2_{ij} = 0$ ทุกค่า i =1,...,p

แต่เมื่อ $A = (a_{ij})$ จะได้ $b_{ii} = \sum_{j=1}^{n} a^2_{ij} = 0$

ดังนั้น $a_{ij} = 0$ ทุกค่า i และ j , i=1,2,...,p ; j = 1,2,...,n

นั่นคือ A = 0 และ $\underline{\hat{\theta}}^* = (X'X)^{-1}X'Y = \underline{\hat{\theta}}$ ซึ่งเป็นตัวประมาณแบบกำลังสองต่ำสุดของ $\underline{\theta}$

4. ตัวประมาณไม่เอนเอียงและมีความแปรปรวนต่ำสุดของฟังก์ชันเชิงเส้นใด ๆของพารามิเตอร์ $\theta_1,...,\theta_p$ คือ ฟังก์ชันเดียวกันของตัวประมาณแบบกำลังสองน้อยที่สุดของ $\theta_1,...,\theta_p$

ทฤษฎีบท ในตัวแบบเชิงเส้น $\underline{Y} = X\underline{\theta} + \underline{\in}$ ที่ $E(\underline{\in}) = \underline{0}$

$E(\underline{\in\in}') = \sigma^2 I$, $E(\underline{Y}) = X\underline{\theta}$ ให้ t เป็นเวคเตอร์ขนาด p x 1 ที่มีส่วนประกอบคงที่ตัวประมาณเชิงเส้นไม่เอนเอียง และมีความแปรปรวนต่ำที่สุดของ $\underline{t}'\underline{\hat{\theta}}$ เมื่อ $\underline{\hat{\theta}} = (X'X)^{-1}X'\underline{Y}$

พิสูจน์ ให้ $T\underline{Y}$ เป็นฟังก์ชันเชิงเส้นของ $\underline{Y}$ ที่ใช้ประมาณ $\underline{t}'\underline{\hat{\theta}}$ จะหา T ที่ทำให้ $T\underline{Y}$ เป็นตัวประมาณไม่เอนเอียงและมีความแปรปรวนต่ำที่สุดของ $\underline{t}'\underline{\hat{\theta}}$

จาก $E(T\underline{Y}) = \underline{t}'\underline{\hat{\theta}}$ จะได้ $TX\underline{\theta} = \underline{t}'\underline{\hat{\theta}}$ ดังนั้น $TX = \underline{t}'$

$$V(T\underline{Y}) = E[(T\underline{Y} - \underline{t}'\underline{\theta})(T\underline{Y} - \underline{t}'\underline{\theta})']$$

$$T\underline{Y} - \underline{t}'\underline{\theta} = T(X\underline{\theta} + \underline{\in}) - \underline{t}'\underline{\theta} = TX\underline{\theta} + T\in - \underline{t}'\underline{\theta} = T\in$$

จึงได้

$$V(T\underline{Y}) = E[T\underline{\in}\,\underline{\in}'T'] = TT'\sigma^2$$

$$TT' = [TX(X'X)^{-1}X'][TX(X'X)^{-1}X']' + [T - TX(X'X)^{-1}X'][T - TX(X'X)^{-1}X']'$$

$$= [\underline{t}'(X'X)^{-1}X'][\underline{t}'(X'X)^{-1}X']' + [T - \underline{t}'(X'X)^{-1}X'][T - \underline{t}'(X'X)^{-1}X']'$$

ขวามือของสมการนี้เป็นผลบวกของเมตริกซ์ในรูป AA' ซึ่งแต่ละเทอมมีสมาชิกบนเส้นทแยงมุมหลักไม่ติดลบ เฉพาะเทอมหลังเท่านั้นที่เป็นฟังก์ชันของ T
ดังนั้น TT' มีสมาชิกบนเส้นทแยงมุมหลักน้อยที่สุด เมื่อสมาชิกบนเส้นทแยงมุมหลักของเทอมหลังเป็น 0 หมด ซึ่งจะเป็นเช่นนั้นได้เมื่อ $T = \underline{t}'(X'X)^{-1}X'$
ดังนั้น ตัวประมาณไม่เอนเอียงและมีความแปรปรวนต่ำสุดของ $\underline{t}'\underline{\theta}$ ได้แก่ $T\underline{Y} = \underline{t}'(X'X)^{-1}X'\underline{Y} = \underline{t}'\hat{\underline{\theta}}$

การประมาณความแปรปรวนในตัวแบบเชิงเส้น

ทฤษฎี ในตัวแบบเชิงเส้น $\underline{Y} = X\underline{\theta} + \underline{\in}$ ที่ $E(\underline{\in}) = \underline{0}$ $E(\underline{\in}\underline{\in}') = \sigma^2 I$ ตัวประมาณที่ไม่เอนเอียงของ σ^2 คือ $\frac{1}{n-p}(\underline{Y} - X\hat{\underline{\theta}})'(\underline{Y} - X\hat{\underline{\theta}})$ เมื่อ $\underline{Y}$ เป็นเวคเตอร์ขนาด n x 1 และ x เป็นเมตริกซ์ขนาด p x 1 , $\underline{\in}$ เป็นเวคเตอร์ขนาด n x 1 และ $\hat{\underline{\theta}} = (X'X)^{-1}X'\underline{Y}$

พิสูจน์

$\underline{Y} - X\hat{\underline{\theta}} = (X\underline{\theta} + \underline{\in})$

$I - X(X'X)^{-1}X'$ เป็นเมตริกซ์สมมาตรและไอเดมโพเทนต์ (Symmetric idempotent matrix)
ดังนั้น

$$(\underline{Y} - X\hat{\underline{\theta}})'(\underline{Y} - X\hat{\underline{\theta}}) = \underline{\in}'[I - X(X'X)^{-1}X']\underline{\in}$$

$$E[(\underline{Y} - X\hat{\underline{\theta}})'(\underline{Y} - X\hat{\underline{\theta}})] = \sigma^2 tr[I - X(X'X)^{-1}X']$$

$$= \sigma^2[trI - tr(X(X'X)^{-1}X')]$$

$$= \sigma^2[n - tr.Ip] = (n-p)\sigma^2$$

ดังนั้น $\frac{1}{n-p}(\underline{Y} - X\underline{\theta})'(\underline{Y} - X\underline{\theta})$ เป็นตัวประมาณไม่เอนเอียงของ σ^2

ตัวประมาณแบบภาวะน่าจะเป็นสูงสุด (Maximum Likelihood Estimation) และ
วิธีกำลังสองน้อยที่สุด (Method of Least Square) ของแต่ละการแจกแจงที่ใช้สำหรับการศึกษา ดังนี้

การแจกแจง	พารามิเตอร์	ตัวประมาณค่า
การแจกแจงแบบพัวซอง	λ	$\hat{\lambda} = \overline{X}$

(Poisson Distribution)		
การแจกแจงแบบทวินาม (Binomial Distribution)	p	$\hat{p} = \overline{X}$
การแจกแจงแบบเอ๊กซ์โปเนนเชียล (Exponential Distribution)	β	$\hat{\beta} = \overline{X}$
การแจกแจงแบบปกติ (Normal Distribution)	μ และ σ^2	$\hat{\mu} = \overline{X}, \hat{\sigma}^2 = S^2$

บทที่ 3
วิธีดำเนินการวิจัย

ในการศึกษานี้มีขั้นตอนในการศึกษา 6 ขั้นตอน โดยทั้ง 6 ขั้นตอนดำเนินการโดยใช้การเขียนคำสั่งด้วยโปรแกรมVisual Basic 6.0 มีขั้นตอนในการศึกษา ดังนี้

1. กำหนดค่าพารามิเตอร์ โดยที่

1.1 ค่าพารามิเตอร์ของการแจกแจงแบบต่อเนื่อง โดยในที่นี้

1.1.1 สำหรับการแจกแจงแบบเอ๊กซ์โพเนนเชียล(β)

กำหนดค่าพารามิเตอร์เท่ากับ 1.3 , 1.8 , 2.0 , 2.4 และ 2.9

1.1.2 สำหรับการแจกแจงแบบปกติ (μ ,σ^2)

กำหนดค่าพารามิเตอร์เท่ากับ (0,1),(1,3),(2,5),(3,6) และ (4,5)

1.2 ค่าพารามิเตอร์ของการแจกแจงแบบไม่ต่อเนื่อง โดยในที่นี้

1.2.1 สำหรับการแจกแจงแบบพัวซอง (λ)

กำหนดค่าพารามิเตอร์เท่ากับ 1.3 , 1.8 , 2.0 , 2.4 และ 2.9

1.2.2 สำหรับการแจกแจงแบบทวินาม (n,p)

กำหนดค่าพารามิเตอร์เท่ากับ 20,0.1),(30,0.3),(40,0.5),(50,0.6) และ

(70,0.8)

2. กำหนดขนาดตัวอย่าง (n) เท่ากับ 20 , 40 , 60 , 80 , 100 , 500 ,1000 ,1500 และ 2000 ของทั้งสองการแจกแจง

3. สร้างตัวเลขสุ่มด้วยวิธีการสร้างตัวเลขสุ่ม 5 วิธี คือ

1. วิธีตัดกลางกำลังสอง (Midsquare Method)
2. วิธีตัดกลางของผลคูณ (Midproduct Technique)
3. วิธีตัวคูณคงที่ (Constant Multiplier Technique)
4. วิธี Additive Congruential Method
5. วิธี Linear Congruential Method

4. นำตัวเลขสุ่มที่ได้มาสร้างข้อมูลตามการแจกแจงที่ใช้ในการศึกษาในที่นี้ คือ การแจกแจงแบบเอ๊กซ์โพเนนเชียล การแจกแจงแบบพัวซอง และการแจกแจงแบบทวินาม ใช้วิธีการแปลงผกผัน(The Inverse Transform Algorithm) ส่วนการแจกแจงแบบปกติใช้วิธีการปฏิเสธ(The Rejection Method)

5. นำข้อมูลที่ได้มาประมาณค่าพารามิเตอร์ 2 วิธี คือ

1. การประมาณค่าแบบสภาวะสูงสุด (Maximum Likelihood Estimation)

2. วิธีกำลังสองน้อยที่สุด (Method of Least Square)

6. คำนวณค่าความคลาดเคลื่อนของค่าประมาณเทียบกับค่าพารามิเตอร์ที่แท้จริง โดยในที่นี้คำนวณซ้ำ 500 รอบ , 1000 รอบ , 2000 รอบ และ 3000 รอบ คือ

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(\hat{\theta}_i - \theta_i)^2 \quad \text{และ} \quad \text{RMSE} = \sqrt{MSE}$$

เมื่อ

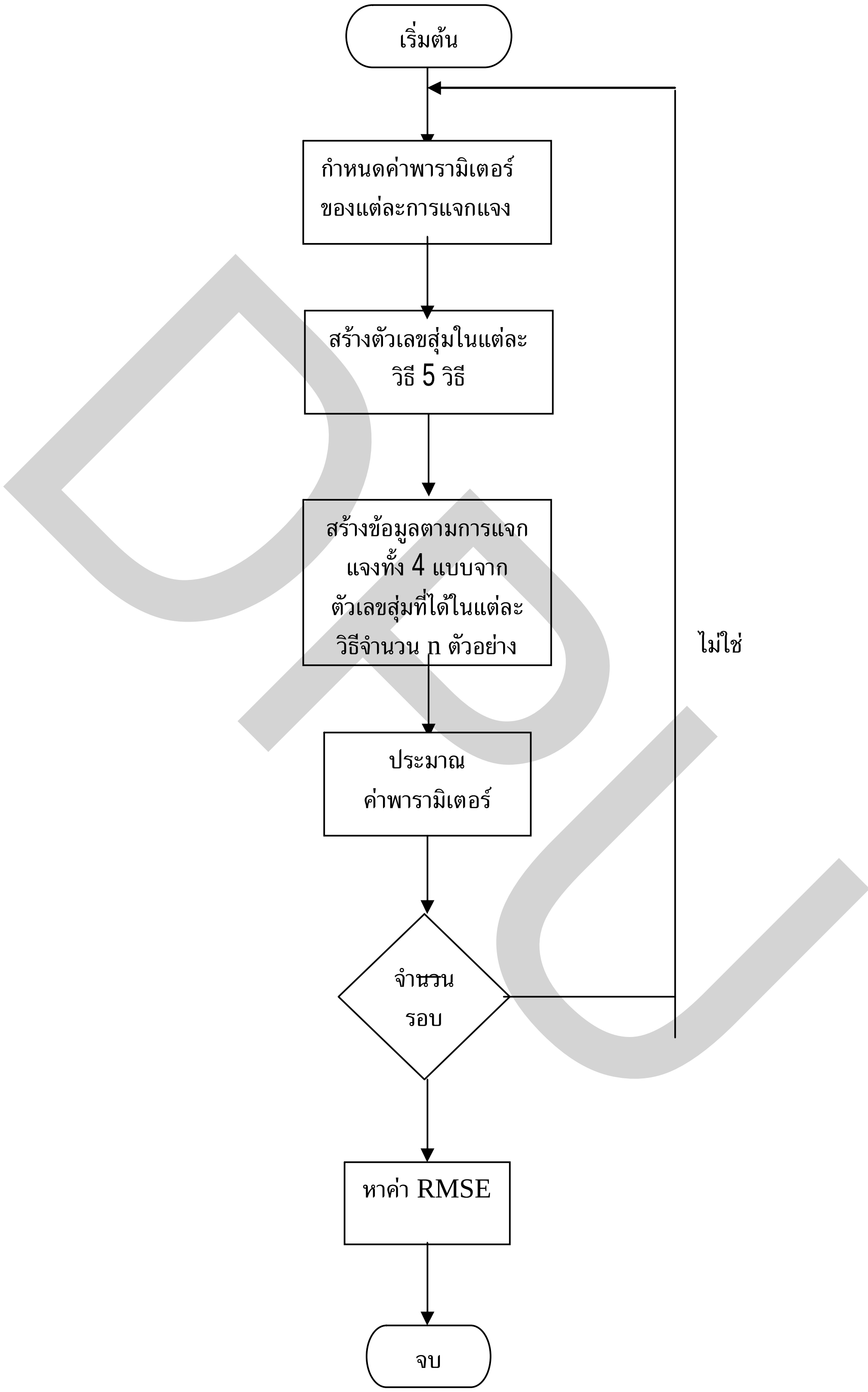
m คือ จำนวนรอบของการทำซ้ำ m = 1,2, ... , m

MSE คือ ค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง

RMSE คือ ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง

โดยนำค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองของทั้ง 2 วิธี มาเปรียบเทียบกันว่าวิธีการสร้างตัวเลขสุ่มวิธีใดที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองของการประมาณค่าน้อยที่สุด แสดงว่าวิธีการสร้างตัวเลขสุ่มวิธีนั้นเป็นวิธีการสร้างค่าตัวเลขสุ่มที่ดีที่สุดเมื่อนำไปใช้ในการประมาณค่าสำหรับการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่อง

แผนภาพแสดงขั้นตอนการสร้างตัวเลขสุ่มและการประมาณค่าพารามิเตอร์



บทที่ 4
ผลการวิจัย

จากการทดสอบการประมาณค่าพารามิเตอร์ของแต่ละการแจกแจงโดยใช้ข้อมูลที่สร้างด้วยวิธีการสร้างตัวเลขสุ่มทั้ง 5 วิธี เพื่อเปรียบเทียบว่าวิธีการสร้างตัวเลขสุ่มวิธีใดที่ดีที่สุด คือ ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองน้อยที่สุด (RMSE:Root of Mean Square Error) นำเสนอผลด้วยตารางและกราฟโดยแยกตามแต่ละการแจกแจงคือ

1. การแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่อง(Discrete Random Variable Distribution) มี 2 การแจกแจง คือ

    1.1 การแจกแจงแบบพัวซอง(Poisson Distribution)

    1.2 การแจกแจงแบบทวินาม(Binomial Distribution)

2. การแจกแจงของตัวแปรสุ่มแบบต่อเนื่อง(Continuous Random Variable Distribution) มี 2 การแจกแจง คือ

    2.1 การแจกแจงแบบเอ๊กซ์โปเนนเชียล(Exponential Distribution)

    2.2 การแจกแจงแบบปกติ(Normal Distribution)

โดย

<u>ตารางและกราฟที่ 1 ถึง ตารางและกราฟที่ 20</u>

แสดงค่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองที่จำนวนรอบของการหาค่า RMSE (RMSE:Root of Mean Square Error) 500 รอบ

<u>ตารางที่ 21 ถึง ตารางที่ 65</u>

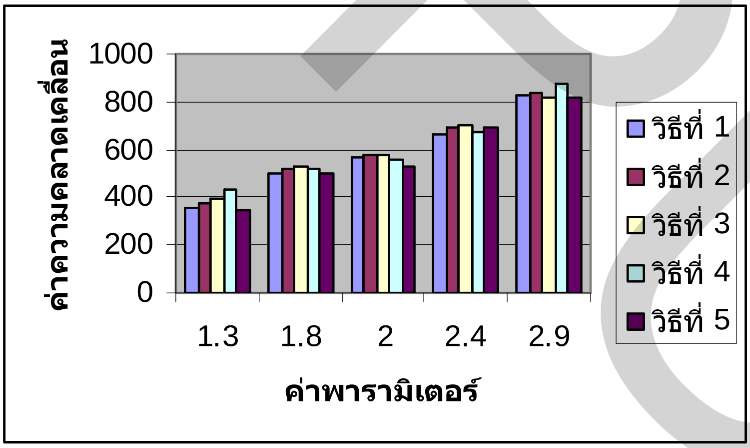
แสดงค่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองที่จำนวนรอบของการหาค่า RMSE (RMSE:Root of Mean Square Error) 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

ตารางและกราฟที่ 1

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบพัวซองเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(λ)				
	1.3	1.8	2.0	2.4	2.9
1.วิธีตัดกลางกำลังสอง	352.3986	501.3252	566.8929	662.6857*	827.8048
2.วิธีตัดกลางของผลคูณ	374.5934	519.5138	579.0952	695.1685	838.1759
3.วิธีตัวคูณคงที่	393.1642	526.0857	573.777	702.0669	818.2703*
4.วิธี Additive Congruential	432.5363	518.8358	562.0288	677.3256	879.2682
5.วิธี Linear Congruential	345.8039*	504.3804*	533.118*	691.781	821.4467

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์



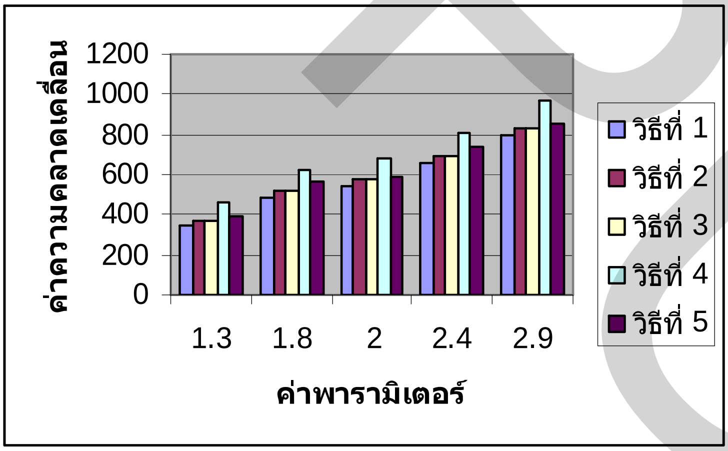
จากตารางและกราฟที่ 1 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือวิธี Linear Congruential จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ 3 ค่า คือ 1.3 1.8 และ 2.0 ส่วนที่ค่าพารามิเตอร์เท่ากับ 2.4 วิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือวิธีตัดกลางกำลังสองให้ค่าต่ำที่สุดและที่ค่าพารามิเตอร์ 2.9 วิธีที่ 3 คือวิธีตัวคูณคงที่เป็นวิธีที่ให้ค่าต่ำที่สุด

ตารางและกราฟที่ 2

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณ ค่าพารามิเตอร์ของการแจกแจงแบบพัวซองเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(λ)				
	1.3	1.8	2.0	2.4	2.9
1.วิธีตัดกลางกำลังสอง	351.1234*	485.4271*	544.2781*	658.9221*	797.1471*
2.วิธีตัดกลางของผลคูณ	371.7341	515.0064	572.9364	687.3352	832.064
3.วิธีตัวคูณคงที่	369.3355	521.2914	575.8822	690.5272	836.0744
4.วิธี Additive Congruential	461.447	627.2512	684.8996	807.4241	966.0005
5.วิธี Linear Congruential	396.3978	562.2019	590.9396	735.1472	857.5851

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์



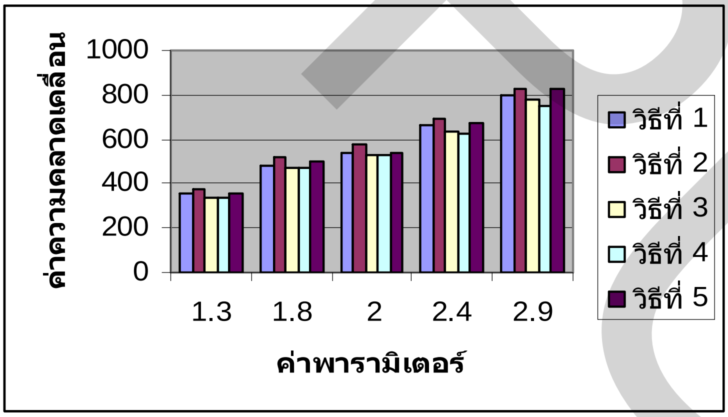
จากตารางและกราฟที่ 2 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือวิธีตัดกลางกำลังสองจะให้ค่าต่ำที่สุดสำหรับทุกค่าพารามิเตอร์ที่ทดสอบ

ตารางและกราฟที่ 3

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบพัวซองเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(λ)				
	1.3	1.8	2.0	2.4	2.9
1.วิธีตัดกลางกำลังสอง	353.0505	483.6199	539.0826	658.8693	793.5549
2.วิธีตัดกลางของผลคูณ	371.4976	515.5009	572.8264	688.1683	831.3711
3.วิธีตัวคูณคงที่	332.1099*	470.8432	524.7002*	636.4749	774.4823
4.วิธี Additive Congruential	340.9855	470.6511*	533.118	624.3226*	753.9882*
5.วิธี Linear Congruential	355.4120	504.3515	542.7982	677.2391	830.9682

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์

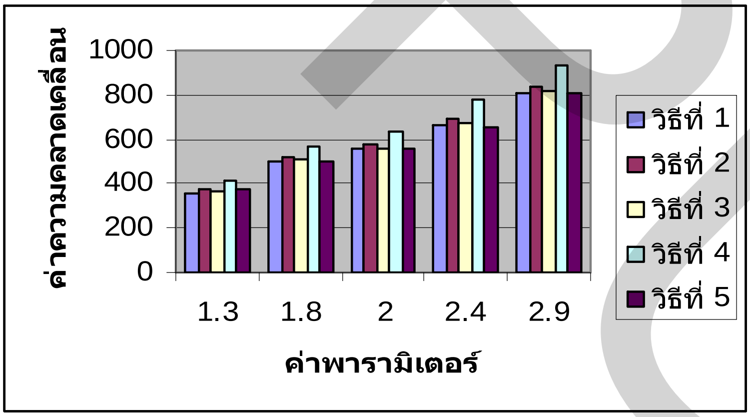


จากตารางและกราฟที่ 3 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 3 คือวิธีตัวคูณคงที่จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ 2 ค่า คือ 1.3 และ 2.0 ส่วนที่ค่าพารามิเตอร์เท่ากับ 1.8 2.4 และ 2.9 วิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือ วิธี Additive Congruential ให้ค่าต่ำที่สุด

ตารางและกราฟที่ 4
ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบพัวซองเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(λ)				
	1.3	1.8	2.0	2.4	2.9
1.วิธีตัดกลางกำลังสอง	356.7877*	497.4041	557.0393*	665.7603	810.6838
2.วิธีตัดกลางของผลคูณ	374.0820	518.6493	575.3405	691.2070	835.6414
3.วิธีตัวคูณคงที่	365.2792	505.3864	559.7135	677.0825	820.7578
4.วิธี Additive Congruential	414.4670	565.8158	637.9196	778.5133	929.8621
5.วิธี Linear Congruential	371.1009	497.1527*	558.4149	655.6425*	806.9913*

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์



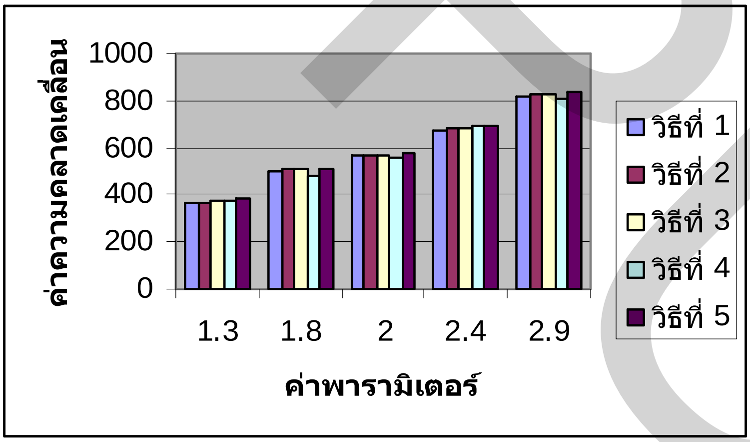
จากตารางและกราฟที่ 4 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือวิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ 2 ค่า คือ 1.3 และ 2.0 ส่วนที่ค่าพารามิเตอร์เท่ากับ 1.8 2.4 และ 2.9 วิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือวิธี Linear Congruential ให้ค่าต่ำที่สุด

ตารางและกราฟที่ 5

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบพัวซองเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(λ)				
	1.3	1.8	2.0	2.4	2.9
1.วิธีตัดกลางกำลังสอง	364.4925*	502.5175	563.4576	670.9149*	814.8109
2.วิธีตัดกลางของผลคูณ	369.3501	513.7038	569.9957	684.8393	827.8396
3.วิธีตัวคูณคงที่	372.0187	514.2333	570.7717	685.2785	827.0681
4.วิธี Additive Congruential	374.8272	484.4112*	559.3714*	689.0631	810.1853*
5.วิธี Linear Congruential	383.3879	513.0536	579.3752	691.7810	833.0110

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์



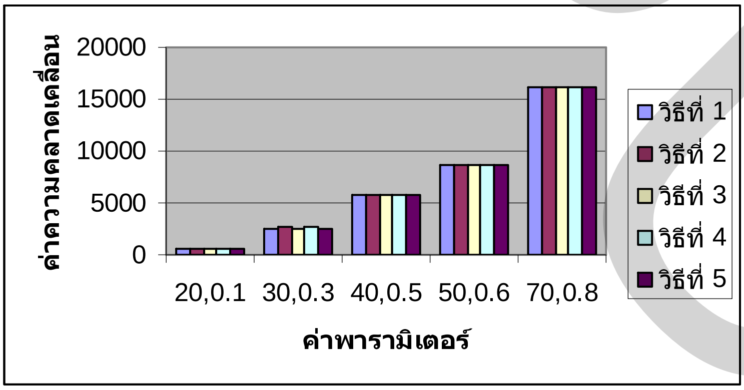
จากตารางและกราฟที่ 5 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือวิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ 2 ค่า คือ 1.3 และ 2.4 ส่วนที่ค่าพารามิเตอร์เท่ากับ 1.8 2.0 และ 2.9 วิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือวิธี Additive Congruential ให้ค่าต่ำที่สุด

ตารางและกราฟที่ 6

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบทวินามเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(n,p)				
	20,0.1	30,0.3	40,0.5	50,0.6	70,0.8
1.วิธีตัดกลางกำลังสอง	565.3776	2,558.9833	5,738.4331	8,616.0916	16,113.8661
2.วิธีตัดกลางของผลคูณ	578.5937	2,599.5633	5,772.4477	8,656.5505	16,150.1721
3.วิธีตัวคูณคงที่	575.7505	2,568.5034	5,713.6525*	8,578.9107*	16,078.7228*
4.วิธี Additive Congruential	562.0288	2,608.6342	5,808.2079	8,676.1733	16,184.9230
5.วิธี Linear Congruential	533.1180*	2,536.3572*	5,721.4756	8,618.3518	16,112.6460

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์



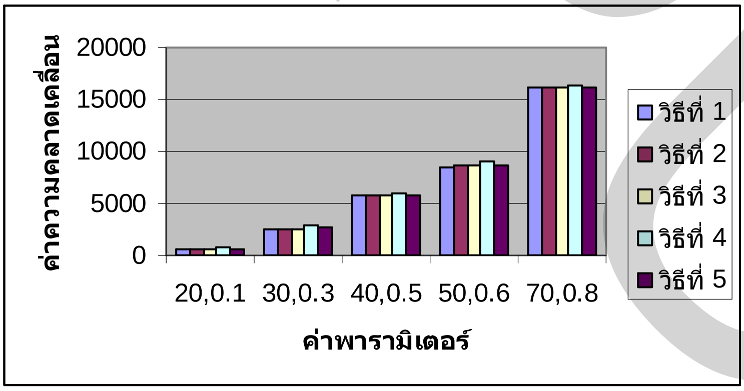
จากตารางและกราฟที่ 6 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 3 คือวิธีตัวคูณคงที่ จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ 3 ค่า คือ (40,0.5),(50,0.6)และ(70,0.8) ส่วนที่ค่าพารามิเตอร์เท่ากับ (20,0.1)และ(30,0.3) วิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือวิธี Linear Congruential ให้ค่าต่ำที่สุด

ตารางและกราฟที่ 7

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบทวินามเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(n,p)				
	20,0.1	30,0.3	40,0.5	50,0.6	70,0.8
1.วิธีตัดกลางกำลังสอง	543.8226*	2536.2134*	5687.4398*	8555.5333*	16060.1153*
2.วิธีตัดกลางของผลคูณ	573.5242	2588.0184	5758.2370	8638.7097	16134.7800
3.วิธีตัวคูณคงที่	574.7049	2593.9629	5771.1316	8652.2217	16146.5339
4.วิธี Additive Congruential	684.8996	2803.7819	6025.0386	8943.5980	16437.8922
5.วิธี Linear Congruential	605.3949	2637.5449	5815.4356	8697.8564	16184.9230

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์



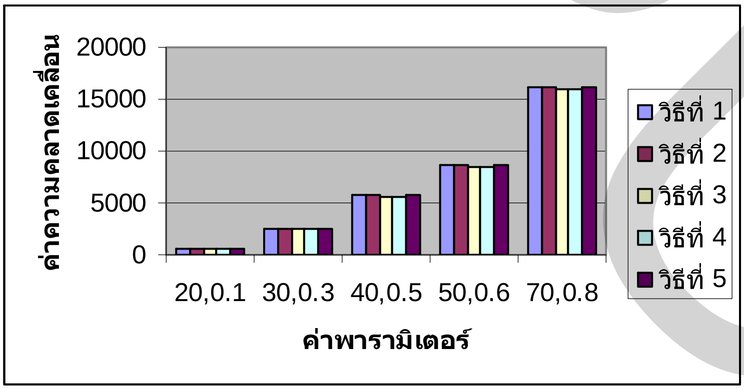
จากตารางและกราฟที่ 7 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือวิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุดสำหรับทุกค่าพารามิเตอร์ที่ทดสอบ

ตารางและกราฟที่ 8

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบทวินามเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(n,p)				
	20,0.1	30,0.3	40,0.5	50,0.6	70,0.8
1.วิธีตัดกลางกำลังสอง	543.4807	2539.8181	5690.5832	8562.4763	16065.1522
2.วิธีตัดกลางของผลคูณ	572.5114	2588.5129	5758.9100	8640.3078	16135.7393
3.วิธีตัวคูณคงที่	527.1895*	2510.6418	5660.1262	8535.2359	16035.5534
4.วิธี Additive Congruential	537.9365	2483.3542*	5605.8325*	8483.4349*	15997.0030*
5.วิธี Linear Congruential	547.6167	2593.9624	5745.3226	8632.5042	16131.6458

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์

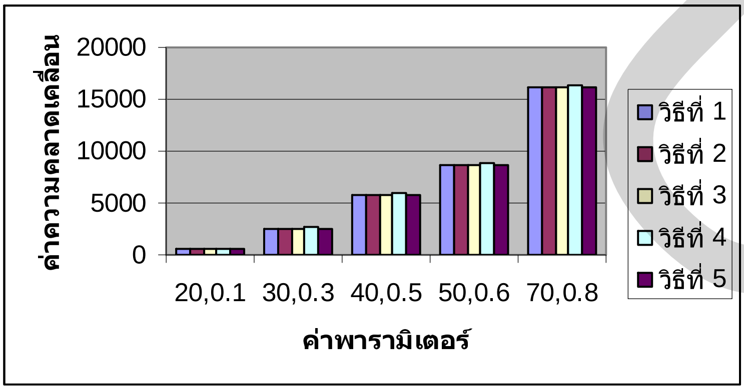


จากตารางและกราฟที่ 8 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 3 คือวิธีตัวคูณคงที่ จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ (20,0.1) ส่วนที่ค่าพารามิเตอร์เท่ากับ (30,0.3),(40,0.5),(50,0.6)และ(70,0.8) วิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือวิธี Additive Congruential ให้ค่าต่ำที่สุด

ตารางและกราฟที่ 9
ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบทวินามเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(n,p)				
	20,0.1	30,0.3	40,0.5	50,0.6	70,0.8
1.วิธีตัดกลางกำลังสอง	556.2060*	2,554.6311	5,713.3522*	8,589.7292*	16,088.0125 *
2.วิธีตัดกลางของผลคูณ	575.3546	2,593.6468	5,763.5021	8,647.0314	16,140.9830
3.วิธีตัวคูณคงที่	562.4003	2,570.1484	5,731.7519	8,612.5066	16,108.5531
4.วิธี Additive Congruen Tial	634.3057	2,720.6634	5,923.8509	8,813.4995	16,304.1799
5.วิธี Linear Congruen Tial	569.2565	2,547.1988*	5,717.8617	8,600.2826	16,094.5768

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์

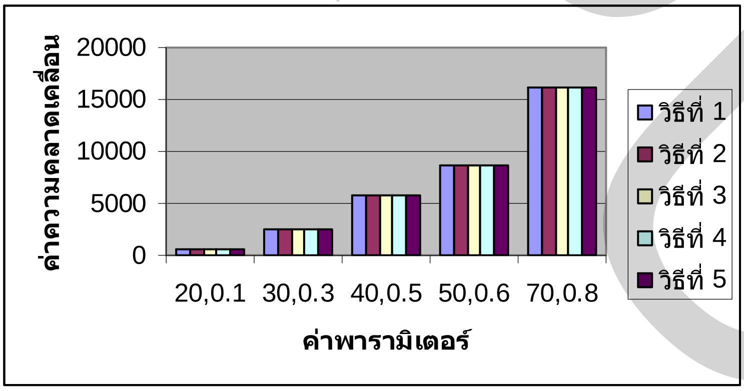


จากตารางและกราฟที่ 9 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือวิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ (20,0.1) , (40,0.5),(50,0.6)และ(70,0.8) ส่วนที่ค่าพารามิเตอร์เท่ากับ (30,0.3) วิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือวิธี Linear Congruential ให้ค่าต่ำที่สุด

ตารางและกราฟที่ 10
ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบทวินามเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(n,p)				
	20,0.1	30,0.3	40,0.5	50,0.6	70,0.8
1.วิธีตัดกลางกำลังสอง	562.4620	2,559.5929*	5,715.1132*	8,589.0047*	16,083.8728*
2.วิธีตัดกลางของผลคูณ	570.4815	2,583.2978	5,750.7233	8,631.3170	16,126.3987
3.วิธีตัวคูณคงที่	570.8381	2,582.3939	5,748.1330	8,627.3958	16,121.1204
4.วิธี Additive Congruential	556.4457*	2,585.8518	5,739.2721	8,636.1657	16,121.7867
5.วิธี Linear Congruential	588.0485	2,591.2877	5,779.2971	8,655.9358	16,153.1211

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์

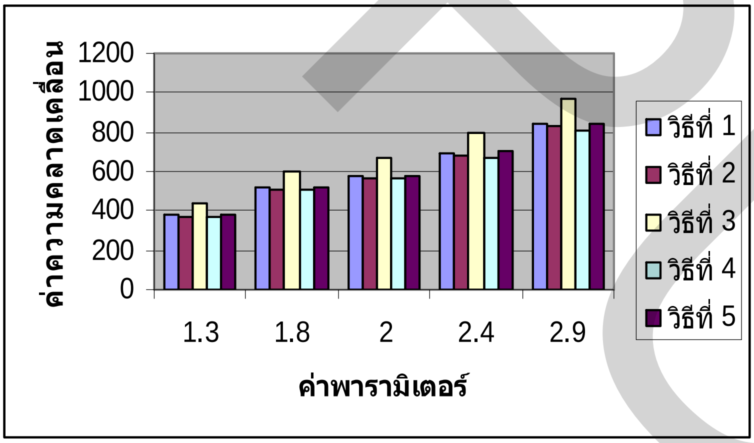


จากตารางและกราฟที่ 10 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือวิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ (30,0.3) , (40,0.5),(50,0.6)และ(70,0.8) ส่วนที่ค่าพารามิเตอร์เท่ากับ (20,0.1) วิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือวิธี Additive Congruential ให้ค่าต่ำที่สุด

ตารางและกราฟที่ 11
ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบเอ๊กซ์โปเนนเชียลเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(β)				
	1.3	1.8	2.0	2.4	2.9
1.วิธีตัดกลางกำลังสอง	375.2358	519.5572	577.2858	692.7429	837.0644
2.วิธีตัดกลางของผลคูณ	370.4358	512.9114	569.9015	683.8818	826.3570
3.วิธีตัวคูณคงที่	433.1638	599.7646	666.4037	799.6836	966.2859
4.วิธี Additive Congruential	364.0987*	504.1375*	560.1510*	672.1795*	812.2198*
5.วิธี Linear Congruential	378.5108	524.0917	582.3241	698.7846	844.3657

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์



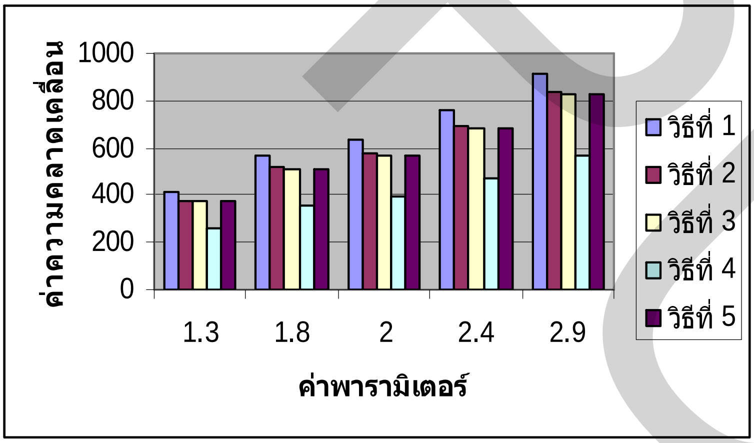
จากตารางและกราฟที่ 11 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือวิธี Additive Congruential จะให้ค่าต่ำที่สุดสำหรับทุกค่าพารามิเตอร์ที่ทดสอบ

ตารางและกราฟที่ 12

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบเอ็กซ์โปเนนเชียลเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(β)				
	1.3	1.8	2.0	2.4	2.9
1.วิธีตัดกลางกำลังสอง	410.9818	569.0518	632.2798	758.7357	916.8057
2.วิธีตัดกลางของผลคูณ	376.1587	520.8355	578.7058	694.4471	839.1237
3.วิธีตัวคูณคงที่	371.2691	514.0643	571.1823	685.4191	828.2149
4.วิธี Additive Congruential	256.0571*	354.5425*	393.9352*	472.7228*	571.2075*
5.วิธี Linear Congruential	371.4247	514.2794	571.4248	685.7076	828.5652

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์



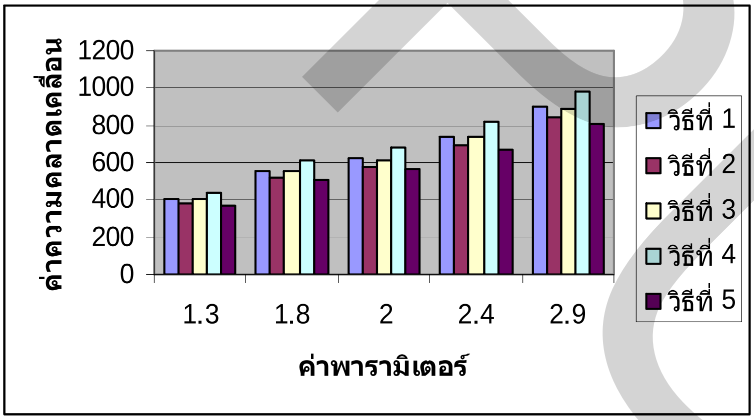
จากตารางและกราฟที่ 12 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือวิธี Additive Congruential จะให้ค่าต่ำที่สุดสำหรับทุกค่าพารามิเตอร์ที่ทดสอบ

ตารางและกราฟที่ 13

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบเอ็กซ์โปเนนเชียลเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(β)				
	1.3	1.8	2.0	2.4	2.9
1.วิธีตัดกลางกำลังสอง	401.7027	556.2038	618.0042	741.6050	896.1061
2.วิธีตัดกลางของผลคูณ	375.8203	520.3668	578.1853	693.8224	838.3685
3.วิธีตัวคูณคงที่	399.2001	552.7388	614.1540	736.9849	890.5233
4.วิธี Additive Congruential	442.1597	612.2248	680.2483	816.2968	986.3619
5.วิธี Linear Congruential	364.5352*	504.7395*	560.8215*	672.9855*	813.1903*

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์



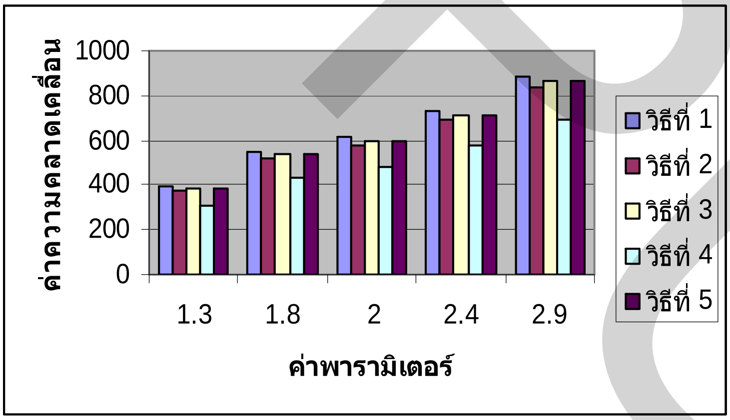
จากตารางและกราฟที่ 13 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือวิธี Linear Congruential จะให้ค่าต่ำที่สุดสำหรับทุกค่าพารามิเตอร์ที่ทดสอบ

ตารางและกราฟที่ 14

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบเอ็กซ์โปเนนเชียลเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(β)				
	1.3	1.8	2.0	2.4	2.9
1.วิธีตัดกลางกำลังสอง	397.1076	549.8413	610.9348	733.1217	885.8554
2.วิธีตัดกลางของผลคูณ	374.5733	518.6403	576.2668	691.5202	835.5867
3.วิธีตัวคูณคงที่	387.5838	536.6552	596.2831	715.5398	864.6106
4.วิธี Additive Congruential	310.9260*	430.5145*	478.3506*	574.0188*	693.6040*
5.วิธี Linear Congruential	386.7936	535.5596	595.0666	714.0810	862.8491

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์



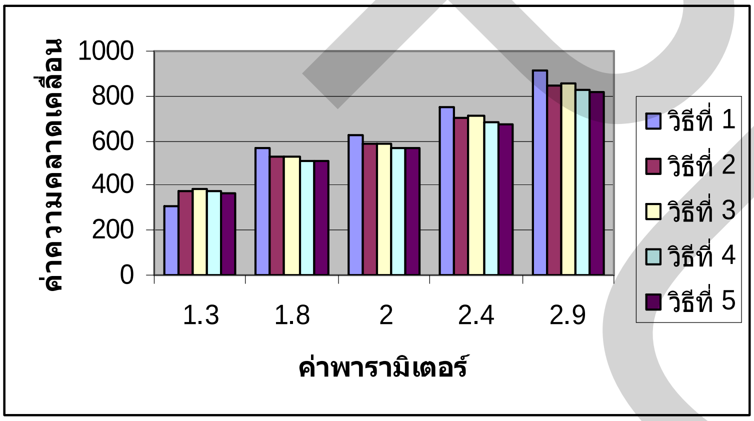
จากตารางและกราฟที่ 14 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือวิธี Additive Congruential จะให้ค่าต่ำที่สุดสำหรับทุกค่าพารามิเตอร์ที่ทดสอบ

<u>ตารางและกราฟที่ 15</u>

<u>ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ ของการแจกแจงแบบเอ็กซ์โปเนนเชียลเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 จำนวนรอบ ของการหาค่า RMSE 500 รอบ</u>

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(β)				
	1.3	1.8	2.0	2.4	2.9
1.วิธีตัดกลางกำลังสอง	304.4829*	564.2071	626.8967	752.2761	909.0003
2.วิธีตัดกลางของผลคูณ	379.6462	525.6644	584.0713	700.8857	846.9034
3.วิธีตัวคูณคงที่	383.8045	531.4218	590.2420	708.5625	856.1791
4.วิธี Additive Congruential	370.6562	513.2168	570.2420	684.2893	826.8500
5.วิธี Linear Congruential	365.9322	506.6739*	562.9727*	675.5684*	816.3101*

<u>กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าพารามิเตอร์</u>



จากตารางและกราฟที่ 15 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือวิธี Linear Congruential จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ 1.8 2.0 2.4 และ 2.9 ส่วนวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือวิธีตัดกลางกำลังสองจะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ 1.3

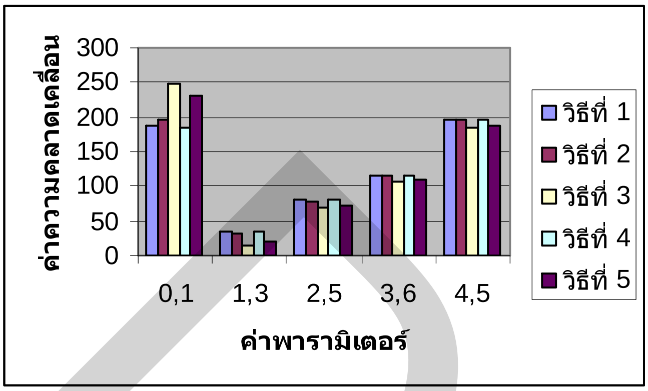
ตารางและกราฟที่ 16

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบปกติ

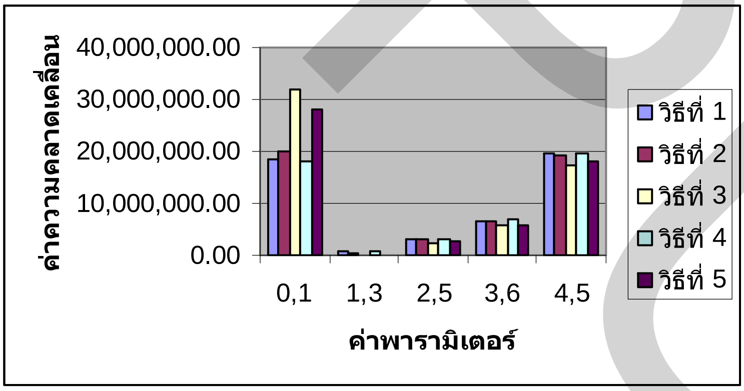
เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(μ, σ^2)					
		(0,1)	(1,3)	(2,5)	(3,6)	(4,5)
1.วิธีตัดกลางกำลังสอง	μ	188.4658	34.4298	79.6939	115.7558	197.073
	σ^2	18,581,821.2373	589,265.3118	3,179,247.9265	6,697,648.092	19,608,238.4899
2.วิธีตัดกลางของผลคูณ	μ	194.9922	32.2489	78.3855	114.6653	195.7652
	σ^2	19,890,711.9341	515,397.8118	3,073,707.6987	6,569,531.3172	19,344,802.4064
3.วิธีตัวคูณคงที่	μ	246.6582	15.0346*	68.0527*	106.0546*	185.4321*
	σ^2	31,833,031.5523	105,100.4932*	2,300,600.8446*	5,598,346.1272*	17,320,895.9134*
4.วิธี Additive Congruential	μ	185.8632*	35.2849	80.2097	116.1859	197.5898
	σ^2	18,073,688.5794*	619,719.418	3,221,500.1894	6,748,718.2638	19,713,068.3219
5.วิธี Linear Congruential	μ	231.8242	19.9673	71.0183	108.5264	188.3981
	σ^2	28,116,893.9094	190,811.144	2,510,958.0189	5,869,081.2066	17,890,255.7677

<u>กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าเฉลี่ยของการแจกแจงแบบปกติเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 ที่จำนวนรอบของการหาค่า RMSE 500 รอบ</u>



<u>กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าความแปรปรวนของการแจกแจงแบบปกติเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20</u>



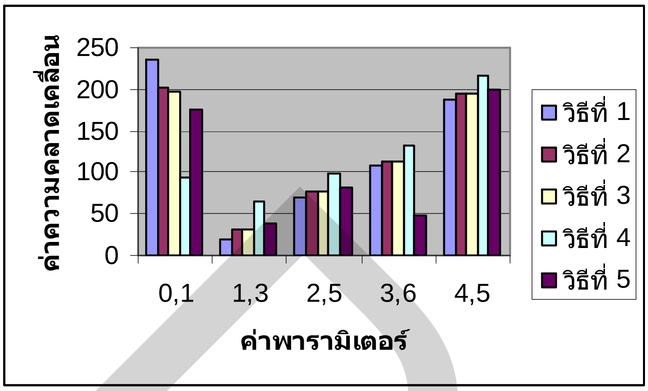
จากตารางและกราฟที่ 16 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 3 คือวิธีตัวคูณคงที่จะให้ค่าต่ำที่สุด สำหรับค่าพารามิเตอร์ (1,3),(2,5),(3,6) และ (4,5) ส่วนวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือ Additive Congruential จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ (0,1)

ตารางและกราฟที่ 17

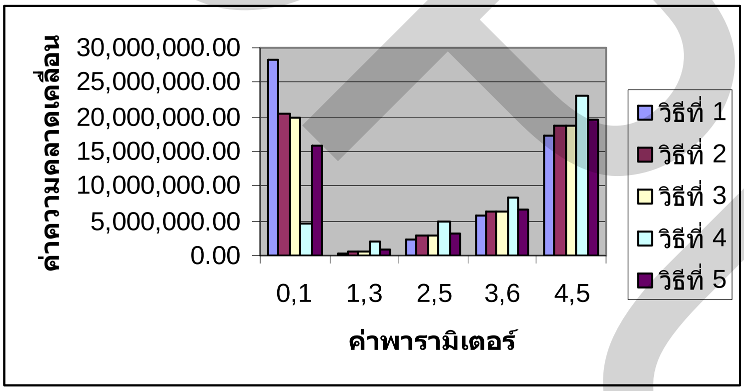
ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบปกติ เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(μ, σ^2)					
		(0,1)	(1,3)	(2,5)	(3,6)	(4,5)
1.วิธีตัดกลางกำลังสอง	μ	235.5928	18.718*	70.2662*	107.8994*	187.6455*
	σ^2	28,294,338.5435	162,378.787*	2,393,690.1451*	5,651,026.4278*	17,289,827.0931*
2.วิธีตัดกลางของผลคูณ	μ	200.8044	30.3098	77.2234	113.6971	194.603
	σ^2	20,552,430.6477	441,955.1422	2,904,661.4426	6,290,814.6946	18,621,446.0039
3.วิธีตัวคูณคงที่	μ	197.9117	31.2807	77.8037	114.1805	195.1829
	σ^2	19,965,517.638	471,570.4672	2,949,460.997	6,345,632.4666	18,734,521.6331
4.วิธี Additive Congruential	μ	94.0014*	65.9039	98.5809	131.4951	215.9613
	σ^2	4,504,701.0824*	2,156,103.7816	4,780,757.6456	8,467,966.7055	23,015,547.5925
5.วิธี Linear Congruential	μ	175.747	38.6567	82.2328	47.8718	199.6129
	σ^2	15,745,801.1741	727,969.8823	3,302,857.2345	6,772,360.7204	19,610,102.7666

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าเฉลี่ยของการแจกแจงแบบปกติเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40
ที่จำนวนรอบของการหาค่า RMSE 500 รอบ



กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าความแปรปรวนของการแจกแจงแบบปกติเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40



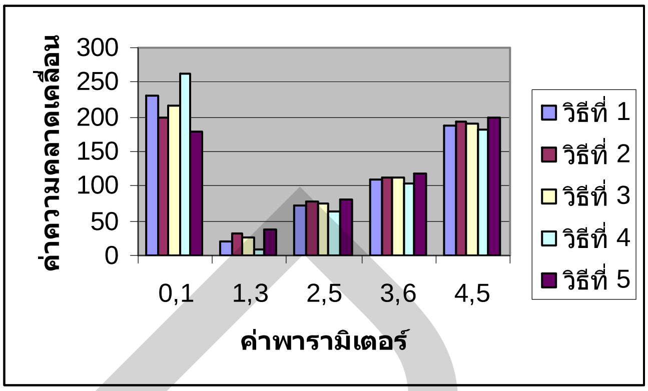
จากตารางและกราฟที่ 17 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือวิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุด สำหรับค่าพารามิเตอร์ (1,3),(2,5),(3,6) และ (4,5) ส่วนวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือ Additive Congruential จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ (0,1)

ตารางและกราฟที่ 18

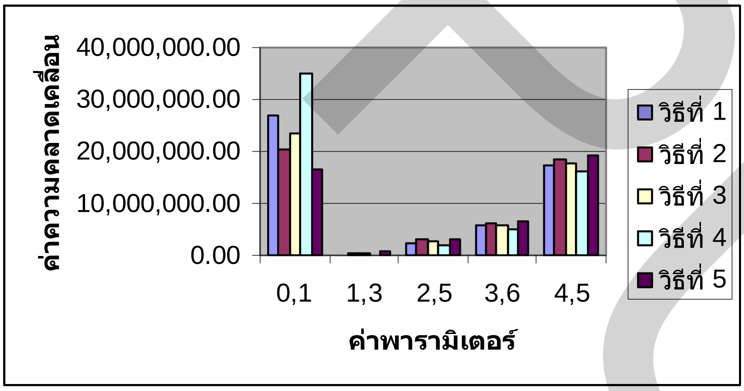
ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบปกติ เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(μ, σ^2)					
		(0,1)	(1,3)	(2,5)	(3,6)	(4,5)
1.วิธีตัดกลางกำลังสอง	μ	230.6159	20.3694	71.2593	108.7271	188.6392
	σ^2	26,883,024.9688	192,257.7779	2,442,769.518	5,691,662.717	17,328,894.1836
2.วิธีตัดกลางของผลคูณ	μ	200.3183	30.4785	77.3231	113.7802	194.7022
	σ^2	20,280,054.1991	443,136.5971	2,887,521.0985	6,246,653.5408	18,482,516.1756
3.วิธีตัวคูณคงที่	μ	216.2409	25.161	74.1348	111.1234	191.5146
	σ^2	23,634,270.2628	298,305.2957	2,649,073.919	5,951,684.7279	17,871,453.174
4.วิธี Additive Congruential	μ	263.3205	9.4755*	64.7198*	103.2775*	182.0992*
	σ^2	35,046,464.1642	37,444.1253*	2,004,766.1202*	5,122,028.0778*	16,126,349.8204*
5.วิธี Linear Congruential	μ	179.8709*	37.2817	81.4078	117.1842	198.7879
	σ^2	16,353,310.9587*	670,202.7474	3,208,078.4564	6,635,105.5558	19,280,700.6209

กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าเฉลี่ยของการแจกแจงแบบปกติเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 ที่จำนวนรอบของการหาค่า RMSE 500 รอบ



กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าความแปรปรวนของการแจกแจงแบบปกติเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60



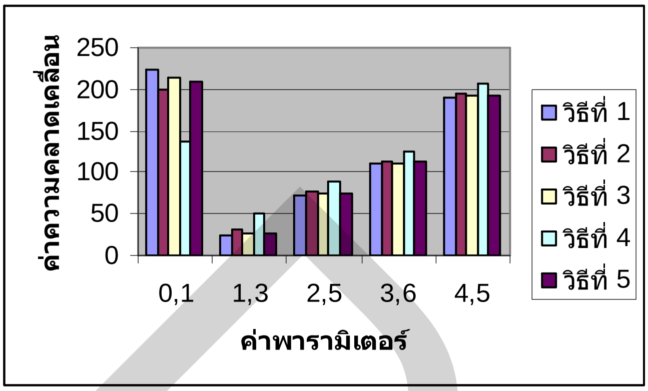
จากตารางและกราฟที่ 18 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือวิธี Additive Congruential จะให้ค่าต่ำที่สุด สำหรับค่าพารามิเตอร์ (1,3),(2,5),(3,6) และ (4,5) ส่วนวิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือ วิธี Linear Congruential จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ (0,1)

ตารางและกราฟที่ 19

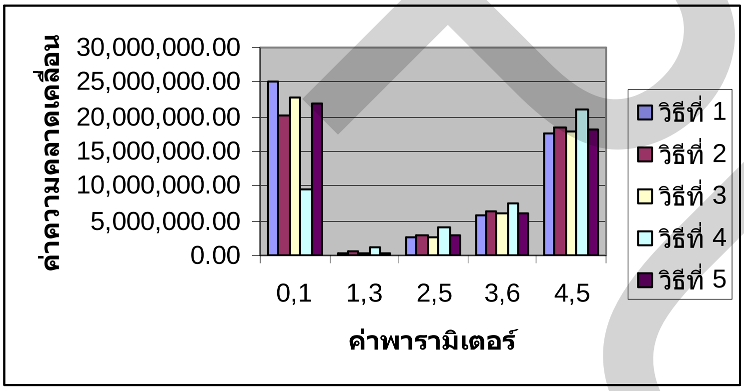
ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบปกติ เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(μ, σ^2)					
		(0,1)	(1,3)	(2,5)	(3,6)	(4,5)
1.วิธีตัดกลางกำลังสอง	μ	222.9091	22.9378*	72.8005*	110.0114*	190.1805*
	σ^2	25010655.2256	245,180.8314*	2,541,508.7045*	5,805,666.7949*	17544226.5397*
2.วิธีตัดกลางของผลคูณ	μ	200.2378	30.4938	77.3348	113.7901	194.7148
	σ^2	20180727.8637	441,773.0833	2,876,252.2204	6,221,440.2817	18406910.6659
3.วิธีตัวคูณคงที่	μ	213.4194	26.0987	74.6973	111.5921	192.0776
	σ^2	22926625.1876	320,504.6105	2,679,103.027	5,977,939.0758	17902776.3862
4.วิธี Additive Congruential	μ	137.1968*	51.506	89.9423	124.2963	207.3226
	σ^2	9474120.8001*	1,290,575.2111	3,915,654.0349	7,452,853.3811	20913978.5776
5.วิธี Linear Congruential	μ	209.0396	27.5605	75.5748	112.3235	192.9547
	σ^2	21993765.4278	358,598.0332	2,743,844.6818	6,058,334.6851	18069574.0861

<u>กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าเฉลี่ยของการแจกแจงแบบปกติเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80</u>
<u>ที่จำนวนรอบของการหาค่า RMSE 500 รอบ</u>



<u>กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าความแปรปรวนของการแจกแจงแบบปกติเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80</u>



จากตารางและกราฟที่ 19 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือวิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุด สำหรับค่าพารามิเตอร์ (1,3),(2,5),(3,6) และ (4,5) ส่วนวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือ Additive Congruential จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ (0,1)

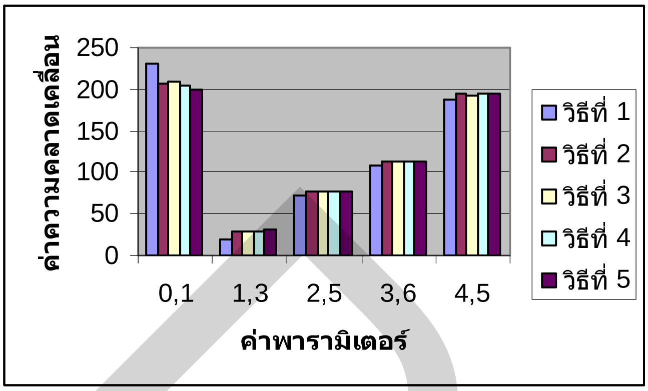
ตารางและกราฟที่ 20

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบปกติ

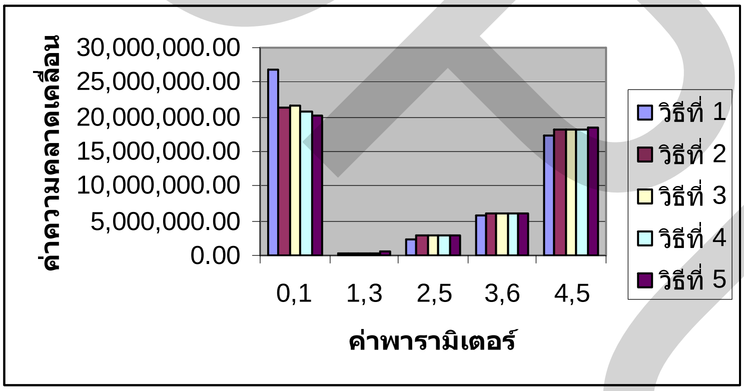
เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 จำนวนรอบของการหาค่า RMSE 500 รอบ

วิธีการสร้างตัวเลขสุ่ม	ค่าพารามิเตอร์(μ, σ^2)					
		(0,1)	(1,3)	(2,5)	(3,6)	(4,5)
1.วิธีตัดกลางกำลังสอง	μ	230.6833	20.3528*	71.2479*	108.7176*	188.6273*
	σ^2	26,717,190.8756	190,532.9893*	2,425,402.3282*	5,652,168.1605*	17,209,758.3449*
2.วิธีตัดกลางของผลคูณ	μ	205.8058	28.6364	76.2205	112.8614	193.6006
	σ^2	21,265,659.3448	387,160.4073	2,785,035.3304	6,102,501.5868	18,147,085.6106
3.วิธีตัวคูณคงที่	μ	208.1898	27.8432	75.744	112.4643	193.1241
	σ^2	21,762,296.4447	365,320.1505	2,749,444.8615	6,058,530.4442	18,056,039.1888
4.วิธี Additive Congruential	μ	203.4448	29.4256	76.6939	113.256	194.0738
	σ^2	20,779,418.619	409,398.9433	2,820,496.7053	6,146,193.8625	18,237,457.9896
5.วิธี Linear Congruential	μ	200.5127*	30.4024	77.2801	113.7445	194.66
	σ^2	20,184,942.6529*	437,905.8124	2,864,800.6897	6,200,600.4568	18,349,844.9969

<u>กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าเฉลี่ยของการแจกแจงแบบปกติเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100</u>
<u>ที่จำนวนรอบของการหาค่า RMSE 500 รอบ</u>



<u>กราฟแสดงการเปรียบเทียบค่าความคลาดเคลื่อนของการประมาณค่าความแปรปรวนของการแจกแจงแบบปกติเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100</u>



จากตารางและกราฟที่ 20 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือวิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุด สำหรับค่าพารามิเตอร์ (1,3),(2,5),(3,6) และ (4,5) ส่วนวิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือ วิธี Linear Congruential จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ (0,1)

แต่เนื่องจากเมื่อทดลองประมวลผลซ้ำหลาย ๆครั้ง ที่จำนวนรอบของการหาค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (MSE: Mean Square Error) จำนวน 500 รอบ พบว่า วิธีการสร้างตัวเลขสุ่มที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุดในแต่ละการแจกแจงทั้งแบบต่อเนื่องและการแจกแจงแบบไม่ต่อเนื่อง แตกต่างกันบ้าง เหมือนกันบ้างจึงได้ทดลองประมวลผลในหลาย ๆลักษณะของการทดสอบ คือ จำนวนรอบของการหาค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (MSE: Mean Square Error) 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ และทดสอบโดยเพิ่มจำนวนตัวอย่างมากขึ้น คือ จำนวนตัวอย่างเท่ากับ 500 ตัวอย่าง 1000 ตัวอย่าง 1500 ตัวอย่าง และ 2000 ตัวอย่าง ในแต่ละการแจกแจงทั้งการแจกแจงแบบต่อเนื่องและการแจกแจงแบบไม่ต่อเนื่อง โดยทดสอบ 5 ครั้งในแต่ละรอบของการทดสอบและแต่ละจำนวนตัวอย่างที่กำหนด พบว่าวิธีการสร้างตัวเลขสุ่มที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด ปรากฏผลโดยระบุจำนวนครั้งเป็นรอยขีดไว้ ดังตารางต่อไปนี้

ตารางที่ 21

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Poisson เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(λ) 1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1	//	/	/	//	//
	วิธีที่ 2					
	วิธีที่ 3	/				
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5		//	//	/	/
1000	วิธีที่ 1	//	//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3		/	/	/	/
	วิธีที่ 4	/				
	วิธีที่ 5	//	//	//	//	//
2000	วิธีที่ 1	//	//	//	/	//
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	/	/	/	//	/
	วิธีที่ 5	//	//	//	//	//
3000	วิธีที่ 1	////	////	///	////	////
	วิธีที่ 2					
	วิธีที่ 3		/	//	/	/
	วิธีที่ 4	/				
	วิธีที่ 5					

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 21 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error)ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) ที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error)ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 4 วิธี Additive Congruential Method

ตารางที่ 22

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Poisson เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	Parameter(λ)					
	วิธีการสร้างตัวเลขสุ่ม	1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	///	//	//	//	//
	วิธีที่ 4	//	///	///	///	///
	วิธีที่ 5					
1000	วิธีที่ 1	//	///	//	//	//
	วิธีที่ 2	/	/			
	วิธีที่ 3	/		/	/	/
	วิธีที่ 4			/	/	/
	วิธีที่ 5	/	/	/	/	/
2000	วิธีที่ 1	//	//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	/	/	/	/	/
3000	วิธีที่ 1	/	/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	//	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 22 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และ วิธีที่ 4 วิธี Additive Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 23

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Poisson เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	Parameter(λ)					
	วิธีการสร้างตัวเลขสุ่ม	1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1		/			
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	////	///	////	///	///
	วิธีที่ 5	/	/	/	/	/
1000	วิธีที่ 1	/	/	//	/	/
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	/	//	/	/	//
	วิธีที่ 5	///	///	//	///	//
2000	วิธีที่ 1	///	//	//	//	///
	วิธีที่ 2					
	วิธีที่ 3		/	/	/	
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	/	/	/	/	/
3000	วิธีที่ 1	/	//	//	/	//
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	//
	วิธีที่ 4	///	/	//	//	
	วิธีที่ 5		/		/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 23 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 24

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Poisson เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	Parameter(λ)					
	วิธีการสร้างตัวเลขสุ่ม	1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1		/	/		/
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	///	//	//	///	//
	วิธีที่ 5	//	//	//	//	//
1000	วิธีที่ 1	///	///	///	///	//
	วิธีที่ 2	/	/	/		
	วิธีที่ 3					
	วิธีที่ 4					
	วิธีที่ 5	/	/	/	//	///
2000	วิธีที่ 1	////	////	////	////	////
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4					
	วิธีที่ 5					
3000	วิธีที่ 1	///	///	/	//	///
	วิธีที่ 2					
	วิธีที่ 3	/	/	//		
	วิธีที่ 4			/	//	/
	วิธีที่ 5	/	/	/	/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 24 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) และวิธีที่ 3 คือ วิธีตัวคูณคงที่ (Constant Multiplier Technique)

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่า RMSE ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 25

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Poisson เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ		Parameter(λ)				
	วิธีการสร้างตัวเลขสุ่ม	1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1	/	/	/	/	/
	วิธีที่ 2		/			
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4					
	วิธีที่ 5	///	//	///	///	///
1000	วิธีที่ 1		/	/	/	
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	///	//	//	//	///
2000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	///	//	//	///	///
	วิธีที่ 5	//	///	///	//	//
3000	วิธีที่ 1	/	//	/	//	/
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	//	//	//	/	//
	วิธีที่ 5	/		/	/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 25 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่า RMSE ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 26

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Poisson เมื่อทดสอบที่ขนาดเท่ากับ 500 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	Parameter(λ)					
	วิธีการสร้างตัวเลขสุ่ม	1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1	////	////	////	////	////
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4		/	/		/
	วิธีที่ 5	/			/	
1000	วิธีที่ 1	/		/	//	/
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	///	//		//
	วิธีที่ 5	//	//	//	///	//
2000	วิธีที่ 1		/		/	
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	/	//	/	//
	วิธีที่ 5	///	///	///	///	///
3000	วิธีที่ 1	//	//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	/				
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	/	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 26 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ ทุกวิธีให้ค่า RMSE ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 27

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Poisson เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 1000 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	Parameter(λ)					
	วิธีการสร้างตัวเลขสุ่ม	1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1				///	
	วิธีที่ 2	/	//	/		//
	วิธีที่ 3	/	/	/		/
	วิธีที่ 4					
	วิธีที่ 5	///	//	///	//	/
1000	วิธีที่ 1				/	//
	วิธีที่ 2	/	/	///		/
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4					
	วิธีที่ 5	///	///	/	///	/
2000	วิธีที่ 1				/	
	วิธีที่ 2	/	/	/		/
	วิธีที่ 3					
	วิธีที่ 4					
	วิธีที่ 5	////	////	////	////	////
3000	วิธีที่ 1				/	/
	วิธีที่ 2	//	/	//		
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	/	/		/	/
	วิธีที่ 5	/	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 27 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ตารางที่ 28

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Poisson เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 1500 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	Parameter(λ)					
	วิธีการสร้างตัวเลขสุ่ม	1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1				/	
	วิธีที่ 2	/	/	/		/
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4					
	วิธีที่ 5	///	///	///	///	///
1000	วิธีที่ 1					
	วิธีที่ 2	/			/	
	วิธีที่ 3	/	/	/	/	//
	วิธีที่ 4	//	////	///	//	//
	วิธีที่ 5	/		/	/	/
2000	วิธีที่ 1				/	
	วิธีที่ 2	//	/	/	/	//
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4		/	/	/	
	วิธีที่ 5	//	//	//	/	//
3000	วิธีที่ 1					
	วิธีที่ 2	/				/
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4					
	วิธีที่ 5	//	///	///	///	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 28 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ ทุกวิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 29

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Poisson เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 2000 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ		Parameter(λ)				
	วิธีการสร้างตัวเลขสุ่ม	1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1				/	
	วิธีที่ 2	//	/	/		/
	วิธีที่ 3					
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//	///	///	///	///
1000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	/	//	/	/
	วิธีที่ 5	///	////	///	////	////
2000	วิธีที่ 1					
	วิธีที่ 2	///	/	///	//	//
	วิธีที่ 3					
	วิธีที่ 4		/			/
	วิธีที่ 5	//	///	//	///	//
3000	วิธีที่ 1					
	วิธีที่ 2	//		//	///	/
	วิธีที่ 3	/	//	/		//
	วิธีที่ 4					
	วิธีที่ 5	//	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 29 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 30

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Binomial เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	Parameter(n,p)					
	วิธีการสร้างตัวเลขสุ่ม	(20,0.1)	(30,0.3)	(40,0.5)	(50,0.6)	(70,0.8)
500	วิธีที่ 1		/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	///	//	//	//	//
1000	วิธีที่ 1	//	//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4					
	วิธีที่ 5	//	//	//	//	//
2000	วิธีที่ 1		///	/	/	//
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	/		/	/	/
	วิธีที่ 5	////	//	///	///	//
3000	วิธีที่ 1	////	///	///	///	///
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4					
	วิธีที่ 5		/	/	/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 30 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 31

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Binomial เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(n,p)				
		(20,0.1)	(30,0.3)	(40,0.5)	(50,0.6)	(70,0.8)
500	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4	///	///	///	///	///
	วิธีที่ 5					
1000	วิธีที่ 1	//	//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	/	/	/	/	/
2000	วิธีที่ 1	//	//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	/	/	/	/	/
3000	วิธีที่ 1	/	/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	///	///	///
	วิธีที่ 5	//	//	/	/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 31 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และ วิธีที่ 4 วิธี Additive Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 32

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Binomial เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(n,p)				
		(20,0.1)	(30,0.3)	(40,0.5)	(50,0.6)	(70,0.8)
500	วิธีที่ 1	/				
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	///	////	////	////	////
	วิธีที่ 5	/	/	/	/	/
1000	วิธีที่ 1	//	/	//	///	///
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//	///	//	/	/
2000	วิธีที่ 1	///	//	///	//	///
	วิธีที่ 2					
	วิธีที่ 3		/		/	
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	/	/	/	/	/
3000	วิธีที่ 1	///	//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4		/	/	/	/
	วิธีที่ 5	/	/	/	/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 32 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และ วิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 33

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Binomial เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	Parameter(n,p)					
	วิธีการสร้างตัวเลขสุ่ม	(20,0.1)	(30,0.3)	(40,0.5)	(50,0.6)	(70,0.8)
500	วิธีที่ 1	/	/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	//	//	//	//	//
1000	วิธีที่ 1	//	///	///	///	///
	วิธีที่ 2	/				
	วิธีที่ 3					
	วิธีที่ 4	/				
	วิธีที่ 5	/	//	//	//	//
2000	วิธีที่ 1	////	////	////	////	////
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4					
	วิธีที่ 5					
3000	วิธีที่ 1	/	//	//	/	/
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	//	/	/	/	/
	วิธีที่ 5	//	//	//	///	///

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 33 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และ วิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) และวิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 34

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Binomial เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	Parameter(n,p)					
	วิธีการสร้างตัวเลขสุ่ม	(20,0.1)	(30,0.3)	(40,0.5)	(50,0.6)	(70,0.8)
500	วิธีที่ 1	/	/	/	//	/
	วิธีที่ 2	/	/			
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4					
	วิธีที่ 5	//	//	///	//	///
1000	วิธีที่ 1	/	/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	///	//	//	//	//
	วิธีที่ 5	/	//	//	//	//
2000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	///	///	///
	วิธีที่ 5	///	///	//	//	//
3000	วิธีที่ 1	/	/	/	//	//
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	//	//	//	/	/
	วิธีที่ 5	/	/	/	/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 34 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และ วิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 35

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Binomial เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 500 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(n,p)				
		(20,0.1)	(30,0.3)	(40,0.5)	(50,0.6)	(70,0.8)
500	วิธีที่ 1	////	////	////	////	////
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5					
1000	วิธีที่ 1	/		//	//	//
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	///	/	/	/
	วิธีที่ 5	//	//	//	//	//
2000	วิธีที่ 1					
	วิธีที่ 2	/				
	วิธีที่ 3					/
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	///	////	////	////	///
3000	วิธีที่ 1	//	//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 35 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือวิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) และวิธีที่ 4 วิธี Additive Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 36

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ค่าต่ำที่สุด สำหรับการแจกแจงแบบ Binomial เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 1000 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(n,p) (20,0.1)	(30,0.3)	(40,0.5)	(50,0.6)	(70,0.8)
500	วิธีที่ 1				/	
	วิธีที่ 2	//	/	/		/
	วิธีที่ 3	/	//	//	//	//
	วิธีที่ 4					
	วิธีที่ 5	//	//	//	//	//
1000	วิธีที่ 1			/	///	//
	วิธีที่ 2	/	//	/		
	วิธีที่ 3	//	//	//	/	//
	วิธีที่ 4					
	วิธีที่ 5	//	/	/	/	/
2000	วิธีที่ 1				/	/
	วิธีที่ 2	/	/	/		
	วิธีที่ 3					
	วิธีที่ 4					
	วิธีที่ 5	////	////	////	////	////
3000	วิธีที่ 1			/	/	/
	วิธีที่ 2	//	/			
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	/	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 36 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ตารางที่ 37

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ค่าต่ำที่สุด สำหรับการแจกแจงแบบ Binomial เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 1500 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	Parameter(n,p)					
	วิธีการสร้างตัวเลขสุ่ม	(20,0.1)	(30,0.3)	(40,0.5)	(50,0.6)	(70,0.8)
500	วิธีที่ 1					
	วิธีที่ 2	/		/	/	
	วิธีที่ 3	/	/		/	
	วิธีที่ 4					
	วิธีที่ 5	///	////	////	///	/////
1000	วิธีที่ 1					
	วิธีที่ 2			/	/	/
	วิธีที่ 3	/	/	/	//	/
	วิธีที่ 4	///	////	//	//	///
	วิธีที่ 5	/		/		
2000	วิธีที่ 1					
	วิธีที่ 2	/	//	//	/	//
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	/	/		//	/
	วิธีที่ 5	//	/	//	/	/
3000	วิธีที่ 1					
	วิธีที่ 2	/	/	/	/	/
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4					
	วิธีที่ 5	//	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 37 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 38

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Binomial เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 2000 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ		Parameter(n,p)				
	วิธีการสร้างตัวเลขสุ่ม	(20,0.1)	(30,0.3)	(40,0.5)	(50,0.6)	(70,0.8)
500	วิธีที่ 1					
	วิธีที่ 2	//	/	/	/	/
	วิธีที่ 3					
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//	///	///	///	///
1000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	////	////	////	////	////
2000	วิธีที่ 1					
	วิธีที่ 2	//	/	/	/	/
	วิธีที่ 3					
	วิธีที่ 4		/	/	/	/
	วิธีที่ 5	///	///	///	///	///
3000	วิธีที่ 1					
	วิธีที่ 2	/	//	//	//	//
	วิธีที่ 3	//				
	วิธีที่ 4		/	/	/	/
	วิธีที่ 5	//	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 38 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 39

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ค่าต่ำที่สุด สำหรับการแจกแจงแบบ Exponential เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	Parameter(β)					
	วิธีการสร้างตัวเลขสุ่ม	1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1					
	วิธีที่ 2	/	/	/	/	/
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//	//	//	//	//
1000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	/////	/////	/////	/////	/////
	วิธีที่ 5					
2000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4	///	///	///	///	///
	วิธีที่ 5					
3000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	///	///	///	///	///
	วิธีที่ 5	/	/	/	/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 39 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 4 วิธี Additive Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 4 วิธี Additive Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 40

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Exponential เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	Parameter(β)					
	วิธีการสร้างตัวเลขสุ่ม	1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1					
	วิธีที่ 2	//	//	//	//	//
	วิธีที่ 3					
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//	//	//	//	//
1000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	//	//	//	//	//
2000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	///	///	///	///	///
3000	วิธีที่ 1					
	วิธีที่ 2	/	/	/	/	/
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4					
	วิธีที่ 5	//	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 40 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 41

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Exponential เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(β) 1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1					
	วิธีที่ 2	/	/	/	/	/
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//	//	//	//	//
1000	วิธีที่ 1					
	วิธีที่ 2	/	/	/	/	/
	วิธีที่ 3					
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	//	//	//	//	//
2000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	//	//	//	//	//
3000	วิธีที่ 1	/	/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	//	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 41 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 42

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Exponential เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	Parameter(β)					
	วิธีการสร้างตัวเลขสุ่ม	1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	///	///	///	///	///
1000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	///	///	///	///	///
2000	วิธีที่ 1					
	วิธีที่ 2	/	/	/	/	/
	วิธีที่ 3					
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	///	///	///	///	///
3000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	///	///	///	///	///
	วิธีที่ 5	/	/	/	/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 42 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 43

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Exponential เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(β)				
		1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	///	///	///	///	///
	วิธีที่ 5	/	/	/	/	/
1000	วิธีที่ 1					
	วิธีที่ 2	/	/	/	/	/
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4					
	วิธีที่ 5	//	//	//	//	//
2000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	////	////	////	////	////
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5					
3000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	///	//	//	//	//
	วิธีที่ 5	//	///	///	///	///

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 43 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 4 วิธี Additive Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 44

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Exponential เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 500 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(β) 1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	///	///	///	///	///
1000	วิธีที่ 1					
	วิธีที่ 2	/	/	/	/	/
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4					
	วิธีที่ 5	//	//	//	//	//
2000	วิธีที่ 1					
	วิธีที่ 2	/	/	/	/	/
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5					
3000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4	///	///	///	///	///
	วิธีที่ 5					

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)
วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)
วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)
วิธีที่ 4 วิธี Additive Congruential Method
วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 44 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 4 วิธี Additive Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 4 วิธี Additive Congruential Method

ตารางที่ 45

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Exponential เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 1000 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(β) 1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1					
	วิธีที่ 2	/	/	/	/	/
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5					
1000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	///	///	///	///	///
	วิธีที่ 4					
	วิธีที่ 5	//	//	//	//	//
2000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	///	///	///	///	///
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5					
3000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)
วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)
วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)
วิธีที่ 4 วิธี Additive Congruential Method
วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 45 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 4 วิธี Additive Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 4 วิธี Additive Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method) และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 46

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Exponential เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 1500 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(β) 1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1	///	///	///	///	///
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4					
	วิธีที่ 5	/	/	/	/	/
1000	วิธีที่ 1	/	/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4					
	วิธีที่ 5	////	////	////	////	////
2000	วิธีที่ 1	/	/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	/	/	/	/	/
3000	วิธีที่ 1	/	/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	//	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 46 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 47

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Exponential เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 2000 จำนวนรอบของการหาค่า RMSE 500 รอบ 1000 รอบ 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(β) 1.3	1.8	2.0	2.4	2.9
500	วิธีที่ 1	//	//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4					
	วิธีที่ 5	/	/	/	/	/
1000	วิธีที่ 1	//	//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5					
2000	วิธีที่ 1	//	//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	///	///	///	///	///
	วิธีที่ 4					
	วิธีที่ 5					
3000	วิธีที่ 1					
	วิธีที่ 2					
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4					
	วิธีที่ 5	///	///	///	///	///

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 47 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 4 วิธี Additive Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) และวิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 48

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุดสำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
500	วิธีที่ 1		/	/	/	/
	วิธีที่ 2	/				
(μ)	วิธีที่ 3	/	/			
	วิธีที่ 4	/	//	//	//	//
	วิธีที่ 5	//	/	//	//	//
(σ^2)	วิธีที่ 1		/	/	/	/
	วิธีที่ 2	/				
	วิธีที่ 3	/	/			
	วิธีที่ 4	/	//	//	//	//
	วิธีที่ 5	//	/	//	//	//
1000	วิธีที่ 1		//	//	//	//
(μ)	วิธีที่ 2					
	วิธีที่ 3		/	/	/	/
	วิธีที่ 4	/////				
	วิธีที่ 5		//	//	//	//
(σ^2)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3		/	/	/	/
	วิธีที่ 4	/////				
	วิธีที่ 5		//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)
วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)
วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)
วิธีที่ 4 วิธี Additive Congruential Method
วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 48 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 49

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
2000 (μ)	วิธีที่ 1		///	/	/	/
	วิธีที่ 2					
	วิธีที่ 3	//				
	วิธีที่ 4	///	/	/	/	/
	วิธีที่ 5		/	///	///	///
(σ^2)	วิธีที่ 1		///	/	/	/
	วิธีที่ 2					
	วิธีที่ 3	//				
	วิธีที่ 4	///	/	/	/	/
	วิธีที่ 5		/	///	///	///
3000 (μ)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	///				
	วิธีที่ 5	/	//	//	//	//
(σ^2)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	///				
	วิธีที่ 5	/	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 49 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 50

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
500	วิธีที่ 1	/	/			
	วิธีที่ 2	/				
(μ)	วิธีที่ 3		/	//	//	//
	วิธีที่ 4		/	/	/	/
	วิธีที่ 5	///	//	//	//	//
(σ^2)	วิธีที่ 1	/	/			
	วิธีที่ 2	/				
	วิธีที่ 3		/	//	//	//
	วิธีที่ 4		/	/	/	/
	วิธีที่ 5	///	//	//	//	//
1000	วิธีที่ 1		//	//	//	//
(μ)	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	//	/	/	/	/
	วิธีที่ 5	//	/	/	/	/
(σ^2)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	//	/	/	/	/
	วิธีที่ 5	//	/	/	/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 50 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 51

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
2000 (μ)	วิธีที่ 1		///	//	//	//
	วิธีที่ 2	/				
	วิธีที่ 3					
	วิธีที่ 4	/	//	//	//	//
	วิธีที่ 5	///		/	/	/
(σ^2)	วิธีที่ 1		///	//	//	//
	วิธีที่ 2	/				
	วิธีที่ 3					
	วิธีที่ 4	/	//	//	//	//
	วิธีที่ 5	///		/	/	/
3000 (μ)	วิธีที่ 1					
	วิธีที่ 2	///				
	วิธีที่ 3	/				
	วิธีที่ 4		///	///	///	///
	วิธีที่ 5	//	//	//	//	//
(σ^2)	วิธีที่ 1					
	วิธีที่ 2	///				
	วิธีที่ 3	/				
	วิธีที่ 4		///	///	///	///
	วิธีที่ 5	//	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 51 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 52

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ, σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
500 (μ)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2	/				
	วิธีที่ 3	//				
	วิธีที่ 4	/	//	//	//	//
	วิธีที่ 5	/	/	/	/	/
(σ^2)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2	/				
	วิธีที่ 3	//				
	วิธีที่ 4	/	//	//	//	//
	วิธีที่ 5	/	/	/	/	/
1000 (μ)	วิธีที่ 1		///	///	///	///
	วิธีที่ 2	/				
	วิธีที่ 3					
	วิธีที่ 4	//				
	วิธีที่ 5	//	//	//	//	//
(σ^2)	วิธีที่ 1		///	///	///	///
	วิธีที่ 2	/				
	วิธีที่ 3					
	วิธีที่ 4	//				
	วิธีที่ 5	//	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 52 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 53

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter (μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
2000	วิธีที่ 1		///	///	///	///
	วิธีที่ 2					
(μ)	วิธีที่ 3					
	วิธีที่ 4	///	/	/	/	/
	วิธีที่ 5	//	/	/	/	/
(σ^2)	วิธีที่ 1		///	///	///	///
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	///	/	/	/	/
	วิธีที่ 5	//	/	/	/	/
3000	วิธีที่ 1	/	//	//	//	//
(μ)	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	/	/	/	/
	วิธีที่ 5	//	//	//	//	//
(σ^2)	วิธีที่ 1	/	//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	/	/	/	/
	วิธีที่ 5	//	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)
วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)
วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)
วิธีที่ 4 วิธี Additive Congruential Method
วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 53 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 54

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
500 (μ)	วิธีที่ 1		/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	///	//	//	//	//
(σ^2)	วิธีที่ 1		/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3					
	วิธีที่ 4	//	//	//	//	//
	วิธีที่ 5	///	//	//	//	//
1000 (μ)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	/				
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	///	//	//	//	//
(σ^2)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	/				
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	///	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 54 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 55

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter (μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
2000 (μ)	วิธีที่ 1		////	////	////	////
	วิธีที่ 2					
	วิธีที่ 3		/	/	/	/
	วิธีที่ 4	//				
	วิธีที่ 5	///				
(σ^2)	วิธีที่ 1		////	////	////	////
	วิธีที่ 2					
	วิธีที่ 3		/	/	/	/
	วิธีที่ 4	//				
	วิธีที่ 5	///				
3000 (μ)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	/				
	วิธีที่ 4	///	/	/	/	/
	วิธีที่ 5	/	//	//	//	//
(σ^2)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	/				
	วิธีที่ 4	///	/	/	/	/
	วิธีที่ 5	/	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 55 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 56

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
500	วิธีที่ 1		//	//	//	//
	วิธีที่ 2					
(μ)	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	////				
	วิธีที่ 5		//	//	//	//
(σ^2)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4	////				
	วิธีที่ 5		//	//	//	//
1000	วิธีที่ 1		/	/	/	/
(μ)	วิธีที่ 2	/				
	วิธีที่ 3	//				
	วิธีที่ 4		///	////	////	////
	วิธีที่ 5	//	/			
(σ^2)	วิธีที่ 1		/	/	/	/
	วิธีที่ 2	/				
	วิธีที่ 3	//				
	วิธีที่ 4		///	////	////	////
	วิธีที่ 5	//	/			

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 56 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ตารางที่ 57

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
2000 (μ)	วิธีที่ 1		/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3	////				
	วิธีที่ 4	/	//	//	//	//
	วิธีที่ 5		//	//	//	//
(σ^2)	วิธีที่ 1		/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3	////				
	วิธีที่ 4	/	//	//	//	//
	วิธีที่ 5		//	//	//	//
3000 (μ)	วิธีที่ 1		///	///	///	///
	วิธีที่ 2					
	วิธีที่ 3		/	/	/	/
	วิธีที่ 4	///				
	วิธีที่ 5	//	/	/	/	/
(σ^2)	วิธีที่ 1		///	///	///	///
	วิธีที่ 2					
	วิธีที่ 3		/	/	/	/
	วิธีที่ 4	///				
	วิธีที่ 5	//	/	/	/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)
วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)
วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)
วิธีที่ 4 วิธี Additive Congruential Method
วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 57 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 58

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 500 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
500 (μ)	วิธีที่ 1		////	////	////	////
	วิธีที่ 2					
	วิธีที่ 3	//				
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//				
(σ^2)	วิธีที่ 1		////	////	////	////
	วิธีที่ 2					
	วิธีที่ 3	//				
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//				
1000 (μ)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2	/				
	วิธีที่ 3	//				
	วิธีที่ 4					
	วิธีที่ 5	//	///	///	///	///
(σ^2)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2	/				
	วิธีที่ 3	//				
	วิธีที่ 4					
	วิธีที่ 5	//	///	///	///	///

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)
วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)
วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)
วิธีที่ 4 วิธี Additive Congruential Method
วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 58 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และวิธีที่ 5 วิธี Linear Congruential Method

<u>ตารางที่ 59</u>

<u>แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 500 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ</u>

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
2000 (μ)	วิธีที่ 1		/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3	//	/	/	/	/
	วิธีที่ 4	//				
	วิธีที่ 5		//	//	//	//
(σ^2)	วิธีที่ 1		/	/	/	/
	วิธีที่ 2					
	วิธีที่ 3	//	/	/	/	/
	วิธีที่ 4	//				
	วิธีที่ 5		//	//	//	//
3000 (μ)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	//				
	วิธีที่ 4	//	/	/	/	/
	วิธีที่ 5	/	//	//	//	//
(σ^2)	วิธีที่ 1		//	//	//	//
	วิธีที่ 2					
	วิธีที่ 3	//				
	วิธีที่ 4	//	/	/	/	/
	วิธีที่ 5	/	//	//	//	//

<u>หมายเหตุ</u>

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 59 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ตารางที่ 60

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 1000 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
	วิธีที่ 1					
500	วิธีที่ 2		//	/	/	/
(μ)	วิธีที่ 3	///	/	/	/	/
	วิธีที่ 4	//				
	วิธีที่ 5		//	///	///	///
(σ^2)	วิธีที่ 1					
	วิธีที่ 2		//	/	/	/
	วิธีที่ 3	///	/	/	/	/
	วิธีที่ 4	//				
	วิธีที่ 5		//	///	///	///
1000	วิธีที่ 1		/	/	/	/
(μ)	วิธีที่ 2		/	/	/	/
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4	/	/			
	วิธีที่ 5	//		/	/	/
(σ^2)	วิธีที่ 1		/	/	/	/
	วิธีที่ 2		/	/	/	/
	วิธีที่ 3	//	//	//	//	//
	วิธีที่ 4	/	/			
	วิธีที่ 5	//		/	/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 60 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ตารางที่ 61

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 1000 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
2000	วิธีที่ 1	/				
	วิธีที่ 2		/	/	/	/
	วิธีที่ 3	///				
(μ)	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5		///	///	///	///
(σ^2)	วิธีที่ 1	/				
	วิธีที่ 2		/	/	/	/
	วิธีที่ 3	///				
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5		///	///	///	///
3000	วิธีที่ 1		/	/	/	/
(μ)	วิธีที่ 2		/	/	/	/
	วิธีที่ 3	//	/	/	/	/
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//	/	/	/	/
(σ^2)	วิธีที่ 1		/	/	/	/
	วิธีที่ 2		/	/	/	/
	วิธีที่ 3	//	/	/	/	/
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//	/	/	/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)
วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)
วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)
วิธีที่ 4 วิธี Additive Congruential Method
วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 61 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ตารางที่ 62

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 1500 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
500	วิธีที่ 1	//				
	วิธีที่ 2		/	/	/	/
(μ)	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4		/			
	วิธีที่ 5	//	//	///	///	///
(σ^2)	วิธีที่ 1	//				
	วิธีที่ 2		/	/	/	/
	วิธีที่ 3	/	/	/	/	/
	วิธีที่ 4		/			
	วิธีที่ 5	//	//	///	///	///
1000	วิธีที่ 1	//				
(μ)	วิธีที่ 2		/	/	/	/
	วิธีที่ 3		//	//	//	//
	วิธีที่ 4		/	/	/	/
	วิธีที่ 5	///	/	/	/	/
(σ^2)	วิธีที่ 1	//				
	วิธีที่ 2		/	/	/	/
	วิธีที่ 3		//	//	//	//
	วิธีที่ 4		/	/	/	/
	วิธีที่ 5	///	/	/	/	/

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 62 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ตารางที่ 63

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 1500 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
2000 (μ)	วิธีที่ 1					
	วิธีที่ 2		//	//	//	//
	วิธีที่ 3	//				
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//	//	//	//	//
(σ^2)	วิธีที่ 1					
	วิธีที่ 2		//	//	//	//
	วิธีที่ 3	//				
	วิธีที่ 4	/	/	/	/	/
	วิธีที่ 5	//	//	//	//	//
3000 (μ)	วิธีที่ 1	/				
	วิธีที่ 2		//	/	/	/
	วิธีที่ 3		/	//	//	//
	วิธีที่ 4	//				
	วิธีที่ 5	//	//	//	//	//
(σ^2)	วิธีที่ 1	/				
	วิธีที่ 2		//	/	/	/
	วิธีที่ 3		/	//	//	//
	วิธีที่ 4	//				
	วิธีที่ 5	//	//	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)
วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)
วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)
วิธีที่ 4 วิธี Additive Congruential Method
วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 63 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ตารางที่ 64

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 2000 จำนวนรอบของการหาค่า RMSE 500 รอบ และ 1000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
500 (μ)	วิธีที่ 1	//				
	วิธีที่ 2		/	/	/	/
	วิธีที่ 3	//				
	วิธีที่ 4		//	//	//	//
	วิธีที่ 5	/	//	//	//	//
(σ^2)	วิธีที่ 1	//				
	วิธีที่ 2		/	/	/	/
	วิธีที่ 3	//				
	วิธีที่ 4		//	//	//	//
	วิธีที่ 5	/	//	//	//	//
1000 (μ)	วิธีที่ 1	///				
	วิธีที่ 2		/	/	/	/
	วิธีที่ 3	//				
	วิธีที่ 4		/	/	/	/
	วิธีที่ 5		///	///	///	///
(σ^2)	วิธีที่ 1	///				
	วิธีที่ 2		/	/	/	/
	วิธีที่ 3	//				
	วิธีที่ 4		/	/	/	/
	วิธีที่ 5		///	///	///	///

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 64 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 500 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 1000 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ตารางที่ 65

แสดงวิธีการสร้างตัวเลขสุ่มที่ให้ค่า RMSE ต่ำที่สุด สำหรับการแจกแจงแบบ Normal เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 2000 จำนวนรอบของการหาค่า RMSE 2000 รอบ และ 3000 รอบ

จำนวนรอบของการทดสอบ	วิธีการสร้างตัวเลขสุ่ม	Parameter(μ,σ^2) (0,1)	(1,3)	(2,5)	(3,6)	(4,5)
2000	วิธีที่ 1	//				
	วิธีที่ 2		///	/	/	/
(μ)	วิธีที่ 3	///				
	วิธีที่ 4		/	/	/	/
	วิธีที่ 5		/	///	///	///
(σ^2)	วิธีที่ 1	//				
	วิธีที่ 2		///	/	/	/
	วิธีที่ 3	///				
	วิธีที่ 4		/	/	/	/
	วิธีที่ 5		/	///	///	///
3000	วิธีที่ 1					
(μ)	วิธีที่ 2		/	/	/	/
	วิธีที่ 3	//				
	วิธีที่ 4		///	//	//	//
	วิธีที่ 5	///	/	//	//	//
(σ^2)	วิธีที่ 1					
	วิธีที่ 2		/	/	/	/
	วิธีที่ 3	//				
	วิธีที่ 4		///	//	//	//
	วิธีที่ 5	///	/	//	//	//

หมายเหตุ

วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method)

วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique)

วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique)

วิธีที่ 4 วิธี Additive Congruential Method

วิธีที่ 5 วิธี Linear Congruential Method

จากตารางที่ 65 พบว่า ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 2000 รอบ ทุกวิธีให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ที่จำนวนรอบของการทดสอบค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) 3000 รอบ วิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด คือ วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method

บทที่ 5
สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

จากการศึกษาเรื่อง การเปรียบเทียบวิธีการสร้างตัวเลขสุ่มเมื่อนำไปประมาณค่าพารามิเตอร์ของการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่อง โดยมีวัตถุประสงค์ในการวิจัย ดังนี้

1. เพื่อเปรียบเทียบวิธีการสร้างตัวเลขสุ่ม 5 วิธีสำหรับนำไปประมาณค่าพารามิเตอร์ของการแจกแจงตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงตัวแปรสุ่มแบบไม่ต่อเนื่อง

2. เพื่อศึกษาว่าวิธีการสร้างตัวเลขสุ่มวิธีใดที่ดีที่สุดสำหรับการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่อง

<u>สรุปผลการวิจัยและอภิปรายผล</u>

จากการเปรียบเทียบวิธีการสร้างตัวเลขสุ่ม 5 วิธี คือ วิธีที่ 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method และวิธีที่ 5 วิธี Linear Congruential Method โดยนำตัวเลขสุ่มที่ได้จากแต่ละวิธีไปประมาณค่าพารามิเตอร์ของการแจกแจงแบบต่อเนื่อง คือ การแจกแจงแบบเอ็กซ์โปเนนเชียล (Exponential Distribution) และการแจกแจงแบบปกติ (Normal Distribution) และประมาณค่าพารามิเตอร์ของการแจกแจงแบบไม่ต่อเนื่อง คือ การแจกแจงแบบพัวซอง (Poisson Distribution) และการแจกแจงแบบทวินาม (Binomial Distribution)

พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) สำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบพัวซองเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 วิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือวิธี Linear Congruential จะให้ค่าต่ำที่สุดสำหรับการทดสอบที่ค่าพารามิเตอร์ 3 ค่า คือ 1.3 1.8 และ 2.0 ส่วนการทดสอบที่ค่าพารามิเตอร์เท่ากับ 2.4 วิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือ วิธีตัดกลางกำลังสองให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด และการทดสอบที่ค่าพารามิเตอร์ 2.9 วิธีที่ 3 คือ วิธีตัวคูณคงที่เป็นวิธีที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือ วิธีตัดกลางกำลังสองจะให้ค่าต่ำที่สุดสำหรับทุกค่าพารามิเตอร์ที่ทดสอบ เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square

Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 3 คือ วิธีตัวคูณคงที่จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ 2 ค่า คือ 1.3 และ 2.0 ส่วนการทดสอบที่ค่าพารามิเตอร์เท่ากับ 1.8 2.4 และ 2.9 วิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือ วิธี Additive Congruential ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง(RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือ วิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุดสำหรับการทดสอบที่ค่าพารามิเตอร์ 2 ค่า คือ 1.3 และ 2.0 ส่วนการทดสอบที่ค่าพารามิเตอร์เท่ากับ 1.8 2.4 และ 2.9 วิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือ วิธี Linear Congruential ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด และเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือ วิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ 2 ค่า คือ 1.3 และ 2.4 ส่วนการทดสอบที่ค่าพารามิเตอร์เท่ากับ 1.8 2.0 และ 2.9 วิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือ วิธี Additive Congruential ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง(RMSE:Root of Mean Square Error)ต่ำที่สุด

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) สำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบทวินามเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 3 คือ วิธีตัวคูณคงที่ จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ 3 ค่า คือ (40,0.5),(50,0.6)และ(70,0.8) ส่วนการทดสอบที่ค่าพารามิเตอร์เท่ากับ (20,0.1)และ(30,0.3) วิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือ วิธี Linear Congruential ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง(RMSE:Root of Mean Square Error) ต่ำที่สุด เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือ วิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุดสำหรับทุกค่าพารามิเตอร์ที่ทดสอบ เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 3 คือ วิธีตัวคูณคงที่จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ (20,0.1) ส่วนการทดสอบที่ค่าพารามิเตอร์เท่ากับ (30,0.3),(40,0.5),(50,0.6)และ(70,0.8) วิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือ วิธี Additive Congruential ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง(RMSE:Root of Mean Square Error)ต่ำที่สุด เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือ วิธีตัดกลางกำลังสอง จะให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด สำหรับค่าพารามิเตอร์ (20,0.1) , (40,0.5),(50,0.6)และ(70,0.8) ส่วนการทดสอบที่ค่าพารามิเตอร์เท่ากับ (30,0.3) วิธีการ

สร้างตัวเลขสุ่มวิธีที่ 5 คือ วิธี Linear Congruential ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด และเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือวิธีตัดกลางกำลังสองจะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ (30,0.3) , (40,0.5),(50,0.6)และ(70,0.8) ส่วนการทดสอบที่ค่าพารามิเตอร์เท่ากับ (20,0.1) วิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือ วิธี Additive Congruential ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) สำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบเอ็กซ์โปเนนเชียลเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error)ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือ วิธี Additive Congruential จะให้ค่าต่ำที่สุดสำหรับทุกค่าพารามิเตอร์ที่ทดสอบ เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error)ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือวิธี Additive Congruential จะให้ค่าต่ำที่สุดสำหรับทุกค่าพารามิเตอร์ที่ทดสอบ เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือ วิธี Linear Congruential จะให้ค่าต่ำที่สุดสำหรับทุกค่าพารามิเตอร์ที่ทดสอบ เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือ วิธี Additive Congruential จะให้ค่าต่ำที่สุดสำหรับทุกค่าพารามิเตอร์ที่ทดสอบ และเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือวิธี Linear Congruential จะให้ค่าต่ำที่สุดสำหรับค่าพารามิเตอร์ 1.8 2.0 2.4 และ 2.9 ส่วนวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือ วิธีตัดกลางกำลังสองจะให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุดสำหรับค่าพารามิเตอร์ 1.3

ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) สำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบปกติเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 20 พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสองในวิธีการสร้างตัวเลขสุ่มวิธีที่ 3 คือ วิธีตัวคูณคงที่ จะให้ค่าต่ำที่สุด สำหรับการทดสอบที่ค่าพารามิเตอร์ (1,3),(2,5),(3,6) และ (4,5) ส่วนวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือ Additive Congruential จะให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุดสำหรับการทดสอบที่ค่าพารามิเตอร์ (0,1) เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 40 พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square

Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือ วิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุด สำหรับการทดสอบที่ค่าพารามิเตอร์ (1,3),(2,5),(3,6) และ (4,5) ส่วนวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือ Additive Congruential จะให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุดสำหรับการทดสอบที่ค่าพารามิเตอร์ (0,1) เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 60 พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือวิธี Additive Congruential จะให้ค่าต่ำที่สุด สำหรับการทดสอบที่ค่าพารามิเตอร์ (1,3),(2,5),(3,6) และ (4,5) ส่วนวิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือวิธี Linear Congruential จะให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุดสำหรับการทดสอบที่ค่าพารามิเตอร์ (0,1) เมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 80 พบว่า ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือ วิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุด สำหรับการทดสอบที่ค่าพารามิเตอร์ (1,3),(2,5),(3,6) และ (4,5) ส่วนวิธีการสร้างตัวเลขสุ่มวิธีที่ 4 คือ Additive Congruential จะให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง(RMSE:Root of Mean Square Error) ต่ำที่สุดสำหรับการทดสอบที่ค่าพารามิเตอร์ (0,1) และเมื่อทดสอบที่ขนาดตัวอย่างเท่ากับ 100 พบว่าค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในวิธีการสร้างตัวเลขสุ่มวิธีที่ 1 คือวิธีตัดกลางกำลังสอง จะให้ค่าต่ำที่สุด สำหรับการทดสอบที่ค่าพารามิเตอร์ (1,3),(2,5),(3,6) และ (4,5) ส่วนวิธีการสร้างตัวเลขสุ่มวิธีที่ 5 คือ วิธี Linear Congruential จะให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุดสำหรับการทดสอบที่ค่าพารามิเตอร์ (0,1)

และเมื่อทดลองประมวลผลซ้ำหลาย ๆครั้งที่จำนวนรอบของการคำนวณหาค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (MSE: Mean Square Error) 500 รอบ พบว่า วิธีการสร้างตัวเลขสุ่มที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ในทุกการแจกแจง ทุกค่าพารามิเตอร์และในทุกขนาดตัวอย่างที่ทดสอบนั้นวิธีการสร้างตัวเลขสุ่มที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุดแตกต่างกัน ตัวอย่างเช่น ในการนำตัวเลขสุ่มที่ได้จากทั้ง 5 วิธีไปใช้ประมาณค่าพารามิเตอร์สำหรับการแจกแจงแบบเอ๊กซ์โปเนนเชียล ที่ขนาดตัวอย่างเท่ากับ 20 โดยทดลองประมวลผล 5 ครั้ง พบว่า วิธีการสร้างตัวเลขสุ่มที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุด เป็นวิธีที่ 2 คือ วิธีตัดกลางของผลคูณ (Midproduct Technique) จำนวน 1 ครั้ง วิธีที่ 3 คือ วิธี ตัวคูณคงที่ (Constant Multiplier Technique) จำนวน 1 ครั้ง วิธีที่ 4 คือ วิธี Additive Congruential Method จำนวน 1 ครั้ง และวิธีที่ 5 คือ วิธี Linear Congruential Method จำนวน 2 ครั้ง จึงได้ทดลองทดสอบโดยเพิ่มขนาดตัวอย่างให้มากขึ้นเป็น 500 ตัวอย่าง 1000 ตัวอย่าง 1500 ตัวอย่าง และ 2000 ตัวอย่าง และเพิ่มจำนวนรอบของการคำนวณค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง

(MSE: Mean Square Error) เป็น 1000 รอบ 2000 รอบ และ 3000 รอบ ของทุกการแจกแจง พบว่า วิธีการสร้างตัวเลขสุ่มที่ให้ค่ารากที่สองของค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุดนั้นแตกต่างกัน นั่นคือ ในแต่ละครั้งที่ทดสอบ วิธีการสร้างตัวเลขสุ่มที่ให้ค่าเฉลี่ยของความคลาดเคลื่อนกำลังสอง (RMSE:Root of Mean Square Error) ต่ำที่สุดอาจเป็นวิธีที่ 1 คือ วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) วิธีที่ 4 วิธี Additive Congruential Method หรือ วิธีที่ 5 วิธี Linear Congruential Method ดังที่แสดงผลไว้ในบทที่ 4 แสดงว่าสำหรับในการศึกษาเพื่อเปรียบเทียบวิธีการสร้างตัวเลขสุ่ม 5 วิธีสำหรับนำไปประมาณค่าพารามิเตอร์ของการแจกแจงตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงตัวแปรสุ่มแบบไม่ต่อเนื่อง เมื่อพิจารณาจากผลการทดสอบที่ได้แสดงว่าวิธีการสร้างตัวเลขสุ่มทั้ง 5 วิธีที่มีผู้คิดค้นขึ้นนี้ โดยที่ 4 วิธีแรก ซึ่งเป็นวิธีที่คิดค้นมาในอดีต คือ วิธี 1 วิธีตัดกลางกำลังสอง (Midsquare Method) วิธีที่ 2 วิธีตัดกลางของผลคูณ (Midproduct Technique) วิธีที่ 3 วิธีตัวคูณคงที่ (Constant Multiplier Technique) และ วิธีที่ 4 วิธี Additive Congruential Method ส่วน วิธีที่ 5 คือ วิธี Linear Congruential Method ซึ่งเป็นวิธีที่ใช้ในปัจจุบัน ทั้ง 5 วิธีเป็นวิธีที่เมื่อนำไปใช้ในการประมาณค่าพารามิเตอร์ของการแจกแจงตัวแปรสุ่มแบบต่อเนื่องและการแจกแจงตัวแปรสุ่มแบบไม่ต่อเนื่องสามารถใช้ได้ดีทุกวิธีสำหรับในการศึกษาครั้งนี้ โดยไม่มีวิธีการสร้างตัวเลขสุ่มวิธีใดที่ดีที่สุดทั้งสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงของตัวแปรสุ่มแบบต่อเนื่องโดยในการศึกษาครั้งนี้ทดสอบกับ 2 การแจกแจง คือ การแจกแจงแบบเอ็กซ์โปเนนเชียล (Exponential Distribution) และการแจกแจงแบบปกติ (Normal Distribution) และการประมาณค่าพารามิเตอร์ของการแจกแจงของตัวแปรสุ่มแบบไม่ต่อเนื่องสำหรับการศึกษาในครั้งนี้ทดสอบกับ 2 การแจกแจง คือ การแจกแจงแบบพัวซอง (Poisson Distribution) และการแจกแจงแบบทวินาม (Binomial Distribution)

<u>ข้อเสนอแนะ</u>

1. สำหรับในการศึกษาครั้งนี้นำตัวเลขสุ่มที่ได้จากวิธีการสร้างตัวเลขสุ่มทั้ง 5 วิธีไปประมาณค่าพารามิเตอร์ของการแจกแจงแบบต่อเนื่องและการแจกแจงแบบไม่ต่อเนื่องประเภทละ 2 การแจกแจง คือ สำหรับการแจกแจงแบบต่อเนื่อง ประมาณค่าพารามิเตอร์ของการแจกแจงแบบเอ็กซ์โปเนนเชียล (Exponential Distribution) และ การแจกแจงแบบปกติ (Normal Distribution) ส่วนการแจกแจงแบบไม่ต่อเนื่อง ประมาณค่าพารามิเตอร์ของการแจกแจงแบบพัวซอง (Poisson Distribution) และ การแจกแจงแบบทวินาม (Binomial Distribution) อาจศึกษาโดยนำตัวเลขสุ่มที่ได้จากวิธีการสร้างตัวเลขสุ่มทั้ง 5 วิธีไปทดสอบสำหรับการประมาณค่าพารามิเตอร์ของการแจกแจงแบบอื่น ๆ เพื่อขยายผลต่อไป

2. ในการศึกษาต่อไปอาจศึกษาวิธีการสร้างตัวเลขสุ่มวิธีอื่น ๆที่มีผู้คิดค้นขึ้น หรือ ศึกษาจากตัวเลขสุ่มที่ได้จากโปรแกรมคอมพิวเตอร์แต่ละโปรแกรมที่สร้างขึ้นว่าสามารถนำไปใช้ได้อย่างมีประสิทธิภาพมากน้อยเพียงใดสำหรับกรณีที่นำไปประมาณค่าพารามิเตอร์ของการแจกแจงแบบต่าง ๆทั้งแบบต่อเนื่องและไม่ต่อเนื่อง

บรรณานุกรม

ธัญลักษณ์ คันธะวงศ์. การเปรียบเทียบวิธีการประมาณค่าพารามิเตอร์ของการแจกแจงแบบไวบูลย์และการแจกแจงแบบเอ๊กซ์โพเนนเชียลสองพารามิเตอร์. บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์, 2547.

วิชัย สุรเชิดเกียรติ. การจำลองเชิงคอมพิวเตอร์. กรุงเทพฯ : สกายบุ๊กส์, 2544.

Averill M, Law and W.David Kelton, Simulation Modeling & Analysis, McGraw-Hill : Singapore,1991.

Ross, Sheldon. Simulation 3rd Edition, Academic Press, 2002.

Ross, Sheldon. A First Course in Probability, Macmillan : New York, 1976.

ภาคผนวก

ภาษาคอมพิวเตอร์ (Visual Basic 6.0) ที่ใช้ในการสร้างค่าตัวแปรสุ่ม 5 วิธีและนำไปสร้างข้อมูลที่มีการแจกแจง 4 แบบ คือ การแจกแจงแบบพัวซอง การแจกแจงแบบทวินาม การแจกแจงแบบเอ็กซ์โปเนนเชียล และการแจกแจงแบบปกติ

```
Option Explicit
Public a, b, z, answer, answer5, i, j, k, x1, beta, x_expo, s, parameter, m_error,
loop_test As Variant
Public s_m, est_para, mse_error As Variant
Public x1_3, x1_8, x2_0, x2_4, x2_9 As Variant
Public s1_3, s1_8, s2_0, s2_4, s2_9 As Variant
Public r, p, F, x_poisson As Variant
Public x, a5, w, m, sample, g As Variant

Public m11_3, m11_8, m12_0, m12_4, m12_9, m21_3, m21_8, m22_0, m22_4,
m22_9 As Variant
Public m31_3, m31_8, m32_0, m32_4, m32_9, m41_3, m41_8, m42_0, m42_4,
m42_9 As Variant
Public m51_3, m51_8, m52_0, m52_4, m52_9 As Variant

Public m1_expo1, m1_expo2, m1_expo3, m1_expo4, m1_expo5 As Variant
Public m2_expo1, m2_expo2, m2_expo3, m2_expo4, m2_expo5 As Variant
Public m3_expo1, m3_expo2, m3_expo3, m3_expo4, m3_expo5 As Variant
Public m4_expo1, m4_expo2, m4_expo3, m4_expo4, m4_expo5 As Variant
Public m5_expo1, m5_expo2, m5_expo3, m5_expo4, m5_expo5 As Variant

Public m1s1_3, m1s1_8, m1s2_0, m1s2_4, m1s2_9 As Variant
Public m2s1_3, m2s1_8, m2s2_0, m2s2_4, m2s2_9 As Variant
Public m3s1_3, m3s1_8, m3s2_0, m3s2_4, m3s2_9 As Variant
Public m4s1_3, m4s1_8, m4s2_0, m4s2_4, m4s2_9 As Variant
Public m5s1_3, m5s1_8, m5s2_0, m5s2_4, m5s2_9 As Variant

Public ep11_3, ep11_8, ep12_0, ep12_4, ep12_9 As Variant
Public ep21_3, ep21_8, ep22_0, ep22_4, ep22_9 As Variant
Public ep31_3, ep31_8, ep32_0, ep32_4, ep32_9 As Variant
Public ep41_3, ep41_8, ep42_0, ep42_4, ep42_9 As Variant
Public ep51_3, ep51_8, ep52_0, ep52_4, ep52_9 As Variant

Public s_m11_3, s_m11_8, s_m12_0, s_m12_4, s_m12_9 As Variant
Public s_m21_3, s_m21_8, s_m22_0, s_m22_4, s_m22_9 As Variant
Public s_m31_3, s_m31_8, s_m32_0, s_m32_4, s_m32_9 As Variant
Public s_m41_3, s_m41_8, s_m42_0, s_m42_4, s_m42_9 As Variant
Public s_m51_3, s_m51_8, s_m52_0, s_m52_4, s_m52_9 As Variant

Public mlem11_3, mlem11_8, mlem12_0, mlem12_4, mlem12_9 As Variant
Public mlem21_3, mlem21_8, mlem22_0, mlem22_4, mlem22_9 As Variant
Public mlem31_3, mlem31_8, mlem32_0, mlem32_4, mlem32_9 As Variant
Public mlem41_3, mlem41_8, mlem42_0, mlem42_4, mlem42_9 As Variant
Public mlem51_3, mlem51_8, mlem52_0, mlem52_4, mlem52_9 As Variant
```

```
Public rmlem11_3, rmlem11_8, rmlem12_0, rmlem12_4, rmlem12_9 As Variant
Public rmlem21_3, rmlem21_8, rmlem22_0, rmlem22_4, rmlem22_9 As Variant
Public rmlem31_3, rmlem31_8, rmlem32_0, rmlem32_4, rmlem32_9 As Variant
Public rmlem41_3, rmlem41_8, rmlem42_0, rmlem42_4, rmlem42_9 As Variant
Public rmlem51_3, rmlem51_8, rmlem52_0, rmlem52_4, rmlem52_9 As Variant
Dim m1(0 To 5000), m2(0 To 5000), m3(0 To 5000), m4(0 To 5000), m5(0 To 5000) As Variant
Dim normal1_01(0 To 5000), normal1_13(0 To 5000), normal1_25(0 To 5000), normal1_36(0 To 5000), normal1_45(0 To 5000) As Variant
Dim normal2_01(0 To 5000), normal2_13(0 To 5000), normal2_25(0 To 5000), normal2_36(0 To 5000), normal2_45(0 To 5000) As Variant
Dim normal3_01(0 To 5000), normal3_13(0 To 5000), normal3_25(0 To 5000), normal3_36(0 To 5000), normal3_45(0 To 5000) As Variant
Dim normal4_01(0 To 5000), normal4_13(0 To 5000), normal4_25(0 To 5000), normal4_36(0 To 5000), normal4_45(0 To 5000) As Variant
Dim normal5_01(0 To 5000), normal5_13(0 To 5000), normal5_25(0 To 5000), normal5_36(0 To 5000), normal5_45(0 To 5000) As Variant

Public r11_3, r11_8, r12_0, r12_4, r12_9 As Variant
Public r21_3, r21_8, r22_0, r22_4, r22_9 As Variant
Public r31_3, r31_8, r32_0, r32_4, r32_9 As Variant
Public r41_3, r41_8, r42_0, r42_4, r42_9 As Variant
Public r51_3, r51_8, r52_0, r52_4, r52_9 As Variant

Public p11_3, p11_8, p12_0, p12_4, p12_9 As Variant
Public p21_3, p21_8, p22_0, p22_4, p22_9 As Variant
Public p31_3, p31_8, p32_0, p32_4, p32_9 As Variant
Public p41_3, p41_8, p42_0, p42_4, p42_9 As Variant
Public p51_3, p51_8, p52_0, p52_4, p52_9 As Variant

Public F11_3, F11_8, F12_0, F12_4, F12_9 As Variant
Public F21_3, F21_8, F22_0, F22_4, F22_9 As Variant
Public F31_3, F31_8, F32_0, F32_4, F32_9 As Variant
Public F41_3, F41_8, F42_0, F42_4, F42_9 As Variant
Public F51_3, F51_8, F52_0, F52_4, F52_9 As Variant

Public x11_3, x11_8, x12_0, x12_4, x12_9 As Variant
Public x21_3, x21_8, x22_0, x22_4, x22_9 As Variant
Public x31_3, x31_8, x32_0, x32_4, x32_9 As Variant
Public x41_3, x41_8, x42_0, x42_4, x42_9 As Variant
Public x51_3, x51_8, x52_0, x52_4, x52_9 As Variant

Public s11_3, s11_8, s12_0, s12_4, s12_9 As Variant
Public s21_3, s21_8, s22_0, s22_4, s22_9 As Variant
Public s31_3, s31_8, s32_0, s32_4, s32_9 As Variant
Public s41_3, s41_8, s42_0, s42_4, s42_9 As Variant
Public s51_3, s51_8, s52_0, s52_4, s52_9 As Variant

Public epp11_3, epp11_8, epp12_0, epp12_4, epp12_9 As Variant
```

```
Public epp21_3, epp21_8, epp22_0, epp22_4, epp22_9 As Variant
Public epp31_3, epp31_8, epp32_0, epp32_4, epp32_9 As Variant
Public epp41_3, epp41_8, epp42_0, epp42_4, epp42_9 As Variant
Public epp51_3, epp51_8, epp52_0, epp52_4, epp52_9 As Variant

Public m1p1_3, m1p1_8, m1p2_0, m1p2_4, m1p2_9 As Variant
Public m2p1_3, m2p1_8, m2p2_0, m2p2_4, m2p2_9 As Variant
Public m3p1_3, m3p1_8, m3p2_0, m3p2_4, m3p2_9 As Variant
Public m4p1_3, m4p1_8, m4p2_0, m4p2_4, m4p2_9 As Variant
Public m5p1_3, m5p1_8, m5p2_0, m5p2_4, m5p2_9 As Variant

Public sp_m11_3, sp_m11_8, sp_m12_0, sp_m12_4, sp_m12_9 As Variant
Public sp_m21_3, sp_m21_8, sp_m22_0, sp_m22_4, sp_m22_9 As Variant
Public sp_m31_3, sp_m31_8, sp_m32_0, sp_m32_4, sp_m32_9 As Variant
Public sp_m41_3, sp_m41_8, sp_m42_0, sp_m42_4, sp_m42_9 As Variant
Public sp_m51_3, sp_m51_8, sp_m52_0, sp_m52_4, sp_m52_9 As Variant

Public p1mle1_3, p1mle1_8, p1mle2_0, p1mle2_4, p1mle2_9 As Variant
Public p2mle1_3, p2mle1_8, p2mle2_0, p2mle2_4, p2mle2_9 As Variant
Public p3mle1_3, p3mle1_8, p3mle2_0, p3mle2_4, p3mle2_9 As Variant
Public p4mle1_3, p4mle1_8, p4mle2_0, p4mle2_4, p4mle2_9 As Variant
Public p5mle1_3, p5mle1_8, p5mle2_0, p5mle2_4, p5mle2_9 As Variant

Public rp1mle1_3, rp1mle1_8, rp1mle2_0, rp1mle2_4, rp1mle2_9 As Variant
Public rp2mle1_3, rp2mle1_8, rp2mle2_0, rp2mle2_4, rp2mle2_9 As Variant
Public rp3mle1_3, rp3mle1_8, rp3mle2_0, rp3mle2_4, rp3mle2_9 As Variant
Public rp4mle1_3, rp4mle1_8, rp4mle2_0, rp4mle2_4, rp4mle2_9 As Variant
Public rp5mle1_3, rp5mle1_8, rp5mle2_0, rp5mle2_4, rp5mle2_9 As Variant

Public M1c20_01, s_M1x20_01, M1i20_01, M1pr20_01, M1F20_01, M1x20_01 As Variant
Public M1c30_03, s_M1x30_03, M1i30_03, M1pr30_03, M1F30_03, M1x30_03 As Variant
Public M1c40_05, s_M1x40_05, M1i40_05, M1pr40_05, M1F40_05, M1x40_05 As Variant
Public M1c50_06, s_M1x50_06, M1i50_06, M1pr50_06, M1F50_06, M1x50_06 As Variant
Public M1c70_08, s_M1x70_08, M1i70_08, M1pr70_08, M1F70_08, M1x70_08 As Variant

Public M2c20_01, s_M2x20_01, M2i20_01, M2pr20_01, M2F20_01, M2x20_01 As Variant
Public M2c30_03, s_M2x30_03, M2i30_03, M2pr30_03, M2F30_03, M2x30_03 As Variant
Public M2c40_05, s_M2x40_05, M2i40_05, M2pr40_05, M2F40_05, M2x40_05 As Variant
Public M2c50_06, s_M2x50_06, M2i50_06, M2pr50_06, M2F50_06, M2x50_06 As Variant
Public M2c70_08, s_M2x70_08, M2i70_08, M2pr70_08, M2F70_08, M2x70_08 As Variant
```

Public M3c20_01, s_M3x20_01, M3i20_01, M3pr20_01, M3F20_01, M3x20_01 As Variant
Public M3c30_03, s_M3x30_03, M3i30_03, M3pr30_03, M3F30_03, M3x30_03 As Variant
Public M3c40_05, s_M3x40_05, M3i40_05, M3pr40_05, M3F40_05, M3x40_05 As Variant
Public M3c50_06, s_M3x50_06, M3i50_06, M3pr50_06, M3F50_06, M3x50_06 As Variant
Public M3c70_08, s_M3x70_08, M3i70_08, M3pr70_08, M3F70_08, M3x70_08 As Variant

Public M4c20_01, s_M4x20_01, M4i20_01, M4pr20_01, M4F20_01, M4x20_01 As Variant
Public M4c30_03, s_M4x30_03, M4i30_03, M4pr30_03, M4F30_03, M4x30_03 As Variant
Public M4c40_05, s_M4x40_05, M4i40_05, M4pr40_05, M4F40_05, M4x40_05 As Variant
Public M4c50_06, s_M4x50_06, M4i50_06, M4pr50_06, M4F50_06, M4x50_06 As Variant
Public M4c70_08, s_M4x70_08, M4i70_08, M4pr70_08, M4F70_08, M4x70_08 As Variant

Public M5c20_01, s_M5x20_01, M5i20_01, M5pr20_01, M5F20_01, M5x20_01 As Variant
Public M5c30_03, s_M5x30_03, M5i30_03, M5pr30_03, M5F30_03, M5x30_03 As Variant
Public M5c40_05, s_M5x40_05, M5i40_05, M5pr40_05, M5F40_05, M5x40_05 As Variant
Public M5c50_06, s_M5x50_06, M5i50_06, M5pr50_06, M5F50_06, M5x50_06 As Variant
Public M5c70_08, s_M5x70_08, M5i70_08, M5pr70_08, M5F70_08, M5x70_08 As Variant

Public eBM120_01, M120_01, s_M120_01 As Variant
Public eBM130_03, M130_03, s_M130_03 As Variant
Public eBM140_05, M140_05, s_M140_05 As Variant
Public eBM150_06, M150_06, s_M150_06 As Variant
Public eBM170_08, M170_08, s_M170_08 As Variant

Public eBM220_01, M220_01, s_M220_01 As Variant
Public eBM230_03, M230_03, s_M230_03 As Variant
Public eBM240_05, M240_05, s_M240_05 As Variant
Public eBM250_06, M250_06, s_M250_06 As Variant
Public eBM270_08, M270_08, s_M270_08 As Variant

Public eBM320_01, M320_01, s_M320_01 As Variant
Public eBM330_03, M330_03, s_M330_03 As Variant
Public eBM340_05, M340_05, s_M340_05 As Variant
Public eBM350_06, M350_06, s_M350_06 As Variant

```
Public eBM370_08, M370_08, s_M370_08 As Variant

Public eBM420_01, M420_01, s_M420_01 As Variant
Public eBM430_03, M430_03, s_M430_03 As Variant
Public eBM440_05, M440_05, s_M440_05 As Variant
Public eBM450_06, M450_06, s_M450_06 As Variant
Public eBM470_08, M470_08, s_M470_08 As Variant

Public eBM520_01, M520_01, s_M520_01 As Variant
Public eBM530_03, M530_03, s_M530_03 As Variant
Public eBM540_05, M540_05, s_M540_05 As Variant
Public eBM550_06, M550_06, s_M550_06 As Variant
Public eBM570_08, M570_08, s_M570_08 As Variant

Public mlem120_01, rmlem120_01 As Variant
Public mlem130_03, rmlem130_03 As Variant
Public mlem140_05, rmlem140_05 As Variant
Public mlem150_06, rmlem150_06 As Variant
Public mlem170_08, rmlem170_08 As Variant

Public mlem220_01, rmlem220_01 As Variant
Public mlem230_03, rmlem230_03 As Variant
Public mlem240_05, rmlem240_05 As Variant
Public mlem250_06, rmlem250_06 As Variant
Public mlem270_08, rmlem270_08 As Variant

Public mlem320_01, rmlem320_01 As Variant
Public mlem330_03, rmlem330_03 As Variant
Public mlem340_05, rmlem340_05 As Variant
Public mlem350_06, rmlem350_06 As Variant
Public mlem370_08, rmlem370_08 As Variant

Public mlem420_01, rmlem420_01 As Variant
Public mlem430_03, rmlem430_03 As Variant
Public mlem440_05, rmlem440_05 As Variant
Public mlem450_06, rmlem450_06 As Variant
Public mlem470_08, rmlem470_08 As Variant

Public mlem520_01, rmlem520_01 As Variant
Public mlem530_03, rmlem530_03 As Variant
Public mlem540_05, rmlem540_05 As Variant
Public mlem550_06, rmlem550_06 As Variant
Public mlem570_08, rmlem570_08 As Variant

Public M1y0_1, M1y10_1, M1y20_1, M1z0_1, s_M1z0_1 As Variant
Public M1y1_3, M1y11_3, M1y21_3, M1z1_3, s_M1z1_3 As Variant
Public M1y2_5, M1y12_5, M1y22_5, M1z2_5, s_M1z2_5 As Variant
Public M1y3_6, M1y13_6, M1y23_6, M1z3_6, s_M1z3_6 As Variant
Public M1y4_5, M1y14_5, M1y24_5, M1z4_5, s_M1z4_5 As Variant
```

```
Public M2y0_1, M2y10_1, M2y20_1, M2z0_1, s_M2z0_1 As Variant
Public M2y1_3, M2y11_3, M2y21_3, M2z1_3, s_M2z1_3 As Variant
Public M2y2_5, M2y12_5, M2y22_5, M2z2_5, s_M2z2_5 As Variant
Public M2y3_6, M2y13_6, M2y23_6, M2z3_6, s_M2z3_6 As Variant
Public M2y4_5, M2y14_5, M2y24_5, M2z4_5, s_M2z4_5 As Variant

Public M3y0_1, M3y10_1, M3y20_1, M3z0_1, s_M3z0_1 As Variant
Public M3y1_3, M3y11_3, M3y21_3, M3z1_3, s_M3z1_3 As Variant
Public M3y2_5, M3y12_5, M3y22_5, M3z2_5, s_M3z2_5 As Variant
Public M3y3_6, M3y13_6, M3y23_6, M3z3_6, s_M3z3_6 As Variant
Public M3y4_5, M3y14_5, M3y24_5, M3z4_5, s_M3z4_5 As Variant

Public M4y0_1, M4y10_1, M4y20_1, M4z0_1, s_M4z0_1 As Variant
Public M4y1_3, M4y11_3, M4y21_3, M4z1_3, s_M4z1_3 As Variant
Public M4y2_5, M4y12_5, M4y22_5, M4z2_5, s_M4z2_5 As Variant
Public M4y3_6, M4y13_6, M4y23_6, M4z3_6, s_M4z3_6 As Variant
Public M4y4_5, M4y14_5, M4y24_5, M4z4_5, s_M4z4_5 As Variant

Public M5y0_1, M5y10_1, M5y20_1, M5z0_1, s_M5z0_1 As Variant
Public M5y1_3, M5y11_3, M5y21_3, M5z1_3, s_M5z1_3 As Variant
Public M5y2_5, M5y12_5, M5y22_5, M5z2_5, s_M5z2_5 As Variant
Public M5y3_6, M5y13_6, M5y23_6, M5z3_6, s_M5z3_6 As Variant
Public M5y4_5, M5y14_5, M5y24_5, M5z4_5, s_M5z4_5 As Variant

Public eN10_1, eN11_3, eN12_5, eN13_6, eN14_5 As Variant
Public eN20_1, eN21_3, eN22_5, eN23_6, eN24_5 As Variant
Public eN30_1, eN31_3, eN32_5, eN33_6, eN34_5 As Variant
Public eN40_1, eN41_3, eN42_5, eN43_6, eN44_5 As Variant
Public eN50_1, eN51_3, eN52_5, eN53_6, eN54_5 As Variant

Public m10_1, m1_1_3, m12_5, m13_6, m14_5 As Variant
Public m20_1, m2_1_3, m22_5, m23_6, m24_5 As Variant
Public m30_1, m3_1_3, m32_5, m33_6, m34_5 As Variant
Public m40_1, m4_1_3, m42_5, m43_6, m44_5 As Variant
Public m50_1, m5_1_3, m52_5, m53_6, m54_5 As Variant

Public s_m10_1, s_m1_1_3, s_m12_5, s_m13_6, s_m14_5 As Variant
Public s_m20_1, s_m2_1_3, s_m22_5, s_m23_6, s_m24_5 As Variant
Public s_m30_1, s_m3_1_3, s_m32_5, s_m33_6, s_m34_5 As Variant
Public s_m40_1, s_m4_1_3, s_m42_5, s_m43_6, s_m44_5 As Variant
Public s_m50_1, s_m5_1_3, s_m52_5, s_m53_6, s_m54_5 As Variant

Public mlem10_1, mlem1_1_3, mlem12_5, mlem13_6, mlem14_5 As Variant
Public mlem20_1, mlem2_1_3, mlem22_5, mlem23_6, mlem24_5 As Variant
Public mlem30_1, mlem3_1_3, mlem32_5, mlem33_6, mlem34_5 As Variant
Public mlem40_1, mlem4_1_3, mlem42_5, mlem43_6, mlem44_5 As Variant
Public mlem50_1, mlem5_1_3, mlem52_5, mlem53_6, mlem54_5 As Variant

Public rmlem10_1, rmlem1_1_3, rmlem12_5, rmlem13_6, rmlem14_5 As Variant
Public rmlem20_1, rmlem2_1_3, rmlem22_5, rmlem23_6, rmlem24_5 As Variant
```

```
Public rmlem30_1, rmlem3_1_3, rmlem32_5, rmlem33_6, rmlem34_5 As Variant
Public rmlem40_1, rmlem4_1_3, rmlem42_5, rmlem43_6, rmlem44_5 As Variant
Public rmlem50_1, rmlem5_1_3, rmlem52_5, rmlem53_6, rmlem54_5 As Variant

Public v10_1, s_v10_1, var1_01, mle_v10_1, rmle_v10_1 As Variant
Public v11_3, s_v11_3, var1_13, mle_v11_3, rmle_v11_3 As Variant
Public v12_5, s_v12_5, var1_25, mle_v12_5, rmle_v12_5 As Variant
Public v13_6, s_v13_6, var1_36, mle_v13_6, rmle_v13_6 As Variant
Public v14_5, s_v14_5, var1_45, mle_v14_5, rmle_v14_5 As Variant

Public v20_1, s_v20_1, var2_01, mle_v20_1, rmle_v20_1 As Variant
Public v21_3, s_v21_3, var2_13, mle_v21_3, rmle_v21_3 As Variant
Public v22_5, s_v22_5, var2_25, mle_v22_5, rmle_v22_5 As Variant
Public v23_6, s_v23_6, var2_36, mle_v23_6, rmle_v23_6 As Variant
Public v24_5, s_v24_5, var2_45, mle_v24_5, rmle_v24_5 As Variant

Public v30_1, s_v30_1, var3_01, mle_v30_1, rmle_v30_1 As Variant
Public v31_3, s_v31_3, var3_13, mle_v31_3, rmle_v31_3 As Variant
Public v32_5, s_v32_5, var3_25, mle_v32_5, rmle_v32_5 As Variant
Public v33_6, s_v33_6, var3_36, mle_v33_6, rmle_v33_6 As Variant
Public v34_5, s_v34_5, var3_45, mle_v34_5, rmle_v34_5 As Variant

Public v40_1, s_v40_1, var4_01, mle_v40_1, rmle_v40_1 As Variant
Public v41_3, s_v41_3, var4_13, mle_v41_3, rmle_v41_3 As Variant
Public v42_5, s_v42_5, var4_25, mle_v42_5, rmle_v42_5 As Variant
Public v43_6, s_v43_6, var4_36, mle_v43_6, rmle_v43_6 As Variant
Public v44_5, s_v44_5, var4_45, mle_v44_5, rmle_v44_5 As Variant

Public v50_1, s_v50_1, var5_01, mle_v50_1, rmle_v50_1 As Variant
Public v51_3, s_v51_3, var5_13, mle_v51_3, rmle_v51_3 As Variant
Public v52_5, s_v52_5, var5_25, mle_v52_5, rmle_v52_5 As Variant
Public v53_6, s_v53_6, var5_36, mle_v53_6, rmle_v53_6 As Variant
Public v54_5, s_v54_5, var5_45, mle_v54_5, rmle_v54_5 As Variant

Private Sub Form_Activate()

    s = 0
    s1_3 = 0
    s1_8 = 0
    s2_0 = 0
    s2_4 = 0
    s2_9 = 0
    x_expo = 0
    s_m = 0
    m_error = 0
    mse_error = 0
    r = 0

    Randomize Timer
    m1(0) = Round(Rnd, 4) * 10000
```

```
  m2(0) = Round(Rnd, 4) * 10000
  m2(1) = Round(Rnd, 4) * 10000

  m3(0) = Round(Rnd, 4) * 10000
  k = Round(Rnd, 4) * 10000

  m4(0) = Rnd * 10000
  m4(1) = Rnd * 10000
  m4(2) = Rnd * 10000
  m4(3) = Rnd * 10000
  m4(4) = Rnd * 10000
  m = Rnd * 10000

  If m4(0) Or m4(1) Or m4(2) Or m4(3) Or m4(4) Or m = 0 Then
      Do Until m4(0) And m4(1) And m4(2) And m4(3) And m4(4) And m <> 0

            Randomize Timer
            m4(0) = Rnd * 10000
            m4(1) = Rnd * 10000
            m4(2) = Rnd * 10000
            m4(3) = Rnd * 10000
            m4(4) = Rnd * 10000
            m = Rnd * 10000
      Loop

  End If

  m5(0) = Rnd * 10000
  a5 = Rnd * 10000
  k = Rnd * 10000
  w = Rnd * 10000

  If m5(0) Or a5 Or k Or w = 0 Then
    Do Until m5(0) And a5 And k And w <> 0
            Randomize Timer
            m5(0) = Rnd * 10000
            a5 = Rnd * 10000
            k = Rnd * 10000
            w = Rnd * 10000

    Loop
  End If

sample = InputBox("SAMPLE TEST  ")
loop_test = InputBox("LOOP TEST ")
```

```
For j = 1 To loop_test

    For i = 1 To sample

        Call crn_m1
            Call expo1
            Call poisson1
            Call binomial1
            Call normal1

        Call crn_m2
            Call expo2
            Call poisson2
            Call binomial2
            Call normal2


        Call crn_m3
            Call expo3
            Call poisson3
            Call binomial3
            Call normal3


        Call crn_m4
             Call expo4
             Call poisson4
             Call binomial4
             Call normal4

        Call crn_m5
             Call expo5
             Call poisson5
             Call binomial5
             Call normal5

    Next i

    Call cal_est1
    Call cal_est2
    Call cal_est3
    Call cal_est4
    Call cal_est5

    Call cal_est_poi1
    Call cal_est_poi2
    Call cal_est_poi3
    Call cal_est_poi4
    Call cal_est_poi5
```

```
    Call cal_est_binomial1
    Call cal_est_binomial2
    Call cal_est_binomial3
    Call cal_est_binomial4
    Call cal_est_binomial5

    Call cal_est_normal1
    Call cal_est_normal2
    Call cal_est_normal3
    Call cal_est_normal4
    Call cal_est_normal5

    Call est_std
    Call cal_variance

Next j

   Call cal_mse_err1
   Call cal_mse_err2
   Call cal_mse_err3
   Call cal_mse_err4
   Call cal_mse_err5

   Call mse_poisson1
   Call mse_poisson2
   Call mse_poisson3
   Call mse_poisson4
   Call mse_poisson5

   Call mse_binomial1
   Call mse_binomial2
   Call mse_binomial3
   Call mse_binomial4
   Call mse_binomial5

   Call mse_normal1
   Call mse_normal2
   Call mse_normal3
   Call mse_normal4
   Call mse_normal5

   Call mse_variance

End Sub

Function crn_m3()


        z = k * m3(i - 1)
```

```
        If z < 10000000 Then
          Randomize Timer
          m3(i - 1) = Round(Rnd, 4) * 10000
          z = k * m3(i - 1)
          Call Cal
          If b = 0 Then
             Do Until b <> 0
                  Randomize Timer
                  m3(i - 1) = Round(Rnd, 4) * 10000
                  z = k * m3(i - 1)
                  Call Cal
             Loop
           End If

        Else
           Call Cal
            If b = 0 Then
               Do Until b <> 0
                  Randomize Timer
                  m3(i - 1) = Round(Rnd, 4) * 10000
                  z = k * m3(i - 1)
                  Call Cal
               Loop

           End If

        End If
        m3(i) = b

End Function

Function crn_m1()

        z = m1(i - 1) * m1(i - 1)
        If z < 10000000 Then
          Randomize Timer
          m1(i - 1) = Round(Rnd, 4) * 10000
          z = m1(i - 1) * m1(i - 1)
          Call Cal
          If b = 0 Then
             Do Until b <> 0
                  Randomize Timer
                  m1(i - 1) = Round(Rnd, 4) * 10000
                  z = m1(i - 1) * m1(i - 1)
                  Call Cal
             Loop
          Else
            Call Cal
          End If
        Else
```

```
            Call Cal
             If b = 0 Then
                Do Until b <> 0
                    Randomize Timer
                    m1(i - 1) = Round(Rnd, 4) * 10000
                    z = m1(i - 1) * m1(i - 1)
                    Call Cal
                Loop
             Else
                    Call Cal
             End If
        End If
        m1(i) = b

End Function

Function crn_m2()

       z = m2(i - 1) * m2(i)
       If z < 10000000 Then
          Randomize Timer
          m2(i - 1) = Round(Rnd, 4) * 10000
          m2(i) = Round(Rnd, 4) * 10000
          z = m2(i - 1) * m2(i)
          Call Cal
          If b = 0 Then
            Do Until b <> 0
              Randomize Timer
              m2(i - 1) = Round(Rnd, 4) * 10000
              m2(i) = Round(Rnd, 4) * 10000
              z = m2(i - 1) * m2(i)
              Call Cal
            Loop
         Else
              Call Cal
         End If
       Else
          Call Cal
          If b = 0 Then
             Do Until b <> 0
                 Randomize Timer
                 m2(i - 1) = Round(Rnd, 4) * 10000
                 m2(i) = Round(Rnd, 4) * 10000
                 z = m2(i - 1) * m2(i)
                 Call Cal
             Loop
          Else
             Call Cal

          End If
```

```
        End If
        m2(i) = b
End Function

Function crn_m4()

            m4(i + 4) = (m4(i + 3) + m4(i - 1)) Mod m
            answer = Round(m4(i + 4) / m, 4)
            If answer = 0 Then
                Do Until answer <> 0
                        Randomize Timer
                        m4(0) = Rnd * 10000
                        m4(1) = Rnd * 10000
                        m4(2) = Rnd * 10000
                        m4(3) = Rnd * 10000
                        m4(4) = Rnd * 10000
                        m = Rnd * 10000
                        m4(i + 4) = (m4(i + 3) + m4(i - 1)) Mod m
                        answer = m4(i + 4) / m
                Loop
            End If

End Function

Function crn_m5()

          m5(i) = ((a5 * m5(i - 1)) + k) Mod w
            answer = m5(i) / w
            If answer = 0 Then
                 Do Until answer <> 0
                     Randomize Timer
                     m5(0) = Rnd * 10000
                     a5 = Rnd * 10000
                     k = Rnd * 10000
                     w = Rnd * 10000
                    m5(i) = ((a5 * m5(i - 1)) + k) Mod w
                    answer = m5(i) / w
                 Loop
             End If

End Function

Function cal_mse_err1()

           mlem11_3 = (s_m11_3 / loop_test)
           rmlem11_3 = Round(Sqr(mlem11_3), 4)


           mlem11_8 = (s_m11_8 / loop_test)
           rmlem11_8 = Round(Sqr(mlem11_8), 4)
```

```
            mlem12_0 = (s_m12_0 / loop_test)
            rmlem12_0 = Round(Sqr(mlem12_0), 4)


            mlem12_4 = (s_m12_4 / loop_test)
            rmlem12_4 = Round(Sqr(mlem12_4), 4)


            mlem12_9 = (s_m12_9 / loop_test)
            rmlem12_9 = Round(Sqr(mlem12_9), 4)
 Print "n=" & sample, "loop test = " & loop_test
 Print "M1: RMSE EXPO 1.3=" & rmlem11_3, "1.8=" & rmlem11_8, "2.0=" &
rmlem12_0, "2.4=" & rmlem12_4, "2.9=" & rmlem12_9


End Function

Function cal_mse_err2()

            mlem21_3 = (s_m21_3 / loop_test)
            rmlem21_3 = Round(Sqr(mlem21_3), 4)

            mlem21_8 = (s_m21_8 / loop_test)
            rmlem21_8 = Round(Sqr(mlem21_8), 4)


            mlem22_0 = (s_m22_0 / loop_test)
            rmlem22_0 = Round(Sqr(mlem22_0), 4)


            mlem22_4 = (s_m22_4 / loop_test)
            rmlem22_4 = Round(Sqr(mlem22_4), 4)


            mlem22_9 = (s_m22_9 / loop_test)
            rmlem22_9 = Round(Sqr(mlem22_9), 4)

   Print "M2: RMSE EXPO 1.3=" & rmlem21_3, "1.8=" & rmlem21_8, "2.0=" &
rmlem22_0, "2.4=" & rmlem22_4, "2.9=" & rmlem22_9


End Function

Function cal_mse_err3()

            mlem31_3 = (s_m31_3 / loop_test)
            rmlem31_3 = Round(Sqr(mlem31_3), 4)
```

```
        mlem31_8 = (s_m31_8 / loop_test)
        rmlem31_8 = Round(Sqr(mlem31_8), 4)

        mlem32_0 = (s_m32_0 / loop_test)
        rmlem32_0 = Round(Sqr(mlem32_0), 4)

        mlem32_4 = (s_m32_4 / loop_test)
        rmlem32_4 = Round(Sqr(mlem32_4), 4)

        mlem32_9 = (s_m32_9 / loop_test)
        rmlem32_9 = Round(Sqr(mlem32_9), 4)

Print "M3: RMSE EXPO 1.3=" & rmlem31_3, "1.8 =" & rmlem31_8, "2.0 =" &
rmlem32_0, " 2.4 =" & rmlem32_4, "2.9 =" & rmlem32_9


End Function

Function cal_mse_err4()

        mlem41_3 = (s_m41_3 / loop_test)
        rmlem41_3 = Round(Sqr(mlem41_3), 4)

        mlem41_8 = (s_m41_8 / loop_test)
        rmlem41_8 = Round(Sqr(mlem41_8), 4)

        mlem42_0 = (s_m42_0 / loop_test)
        rmlem42_0 = Round(Sqr(mlem42_0), 4)

        mlem42_4 = (s_m42_4 / loop_test)
        rmlem42_4 = Round(Sqr(mlem42_4), 4)

        mlem42_9 = (s_m42_9 / loop_test)
        rmlem42_9 = Round(Sqr(mlem42_9), 4)

Print "M4: RMSE EXPO 1.3 =" & rmlem41_3, "1.8 =" & rmlem41_8, "2.0 =" &
rmlem42_0, "2.4=" & rmlem42_4, "2.9 =" & rmlem42_9

End Function

Function cal_mse_err5()

        mlem51_3 = (s_m51_3 / loop_test)
        rmlem51_3 = Round(Sqr(mlem51_3), 4)

        mlem51_8 = (s_m51_8 / loop_test)
        rmlem51_8 = Round(Sqr(mlem51_8), 4)

        mlem52_0 = (s_m52_0 / loop_test)
        rmlem52_0 = Round(Sqr(mlem52_0), 4)
```

```
        mlem52_4 = (s_m52_4 / loop_test)
        rmlem52_4 = Round(Sqr(mlem52_4), 4)

        mlem52_9 = (s_m52_9 / loop_test)
        rmlem52_9 = Round(Sqr(mlem52_9), 4)

Print "M5: RMSE EXPO 1.3=" & rmlem51_3, "1.8=" & rmlem51_8, "2.0=" &
rmlem52_0, "2.4=" & rmlem52_4, "2.9=" & rmlem52_9

End Function

Function Cal()

  a = z Mod 1000000
  b = Int(a / 100)
  answer = b / 10000

End Function

Function expo1()

      m11_3 = -(1.3) * Log(answer)
      m1_expo1 = m11_3
      m1s1_3 = m1s1_3 + m1_expo1


      m11_8 = -(1.8) * Log(answer)
      m1_expo2 = m11_8
      m1s1_8 = m1s1_8 + m1_expo2


      m12_0 = -(2) * Log(answer)
      m1_expo3 = m12_0
      m1s2_0 = m1s2_0 + m1_expo3


      m12_4 = -(2.4) * Log(answer)
      m1_expo4 = m12_4
      m1s2_4 = m1s2_4 + m1_expo4


      m12_9 = -(2.9) * Log(answer)
      m1_expo5 = m12_9
      m1s2_9 = m1s2_9 + m1_expo5


End Function
```

```
Function expo2()

    m21_3 = -(1.3) * Log(answer)
    m2_expo1 = Round(m21_3, 4)
    m2s1_3 = m2s1_3 + m2_expo1


    m21_8 = -(1.8) * Log(answer)
    m2_expo2 = Round(m21_8, 4)
    m2s1_8 = m2s1_8 + m2_expo2


    m22_0 = -(2) * Log(answer)
    m2_expo3 = Round(m22_0, 4)
    m2s2_0 = m2s2_0 + m2_expo3


    m22_4 = -(2.4) * Log(answer)
    m2_expo4 = Round(m22_4, 4)
    m2s2_4 = m2s2_4 + m2_expo4


    m22_9 = -(2.9) * Log(answer)
    m2_expo5 = Round(m22_9, 4)
    m2s2_9 = m2s2_9 + m2_expo5


End Function

Function expo3()

    m31_3 = -(1.3) * Log(answer)
    m3_expo1 = Round(m31_3, 4)
    m3s1_3 = m3s1_3 + m3_expo1


    m31_8 = -(1.8) * Log(answer)
    m3_expo2 = Round(m31_8, 4)
    m3s1_8 = m3s1_8 + m3_expo2


    m32_0 = -(2) * Log(answer)
    m3_expo3 = Round(m32_0, 4)
    m3s2_0 = m3s2_0 + m3_expo3


    m32_4 = -(2.4) * Log(answer)
    m3_expo4 = Round(m32_4, 4)
    m3s2_4 = m3s2_4 + m3_expo4
```

```
    m32_9 = -(2.9) * Log(answer)
    m3_expo5 = Round(m32_9, 4)
    m3s2_9 = m3s2_9 + m3_expo5


End Function

Function expo4()

    m41_3 = -(1.3) * Log(answer)
    m4_expo1 = Round(m41_3, 4)
    m4s1_3 = m4s1_3 + m4_expo1


    m41_8 = -(1.8) * Log(answer)
    m4_expo2 = Round(m41_8, 4)
    m4s1_8 = m4s1_8 + m4_expo2


    m42_0 = -(2) * Log(answer)
    m4_expo3 = Round(m42_0, 4)
    m4s2_0 = m4s2_0 + m4_expo3


    m42_4 = -(2.4) * Log(answer)
    m4_expo4 = Round(m42_4, 4)
    m4s2_4 = m4s2_4 + m4_expo4


    m42_9 = -(2.9) * Log(answer)
    m4_expo5 = Round(m42_9, 4)
    m4s2_9 = m4s2_9 + m4_expo5


End Function

Function expo5()

    m51_3 = -(1.3) * Log(answer)
    m5_expo1 = Round(m51_3, 4)
    m5s1_3 = m5s1_3 + m5_expo1


    m51_8 = -(1.8) * Log(answer)
    m5_expo2 = Round(m51_8, 4)
    m5s1_8 = m5s1_8 + m5_expo2


    m52_0 = -(2) * Log(answer)
```

```
        m5_expo3 = Round(m52_0, 4)
        m5s2_0 = m5s2_0 + m5_expo3


        m52_4 = -(2.4) * Log(answer)
        m5_expo4 = Round(m52_4, 4)
        m5s2_4 = m5s2_4 + m5_expo4


        m52_9 = -(2.9) * Log(answer)
        m5_expo5 = Round(m52_9, 4)
        m5s2_9 = m5s2_9 + m5_expo5

End Function

Function cal_est1()

        ep11_3 = m1s1_3 / sample
        m11_3 = (ep11_3 - 1.3) ^ 2
        s_m11_3 = s_m11_3 + m11_3

        ep11_8 = m1s1_8 / sample
        m11_8 = (ep11_8 - 1.8) ^ 2
        s_m11_8 = s_m11_8 + m11_8

        ep12_0 = m1s2_0 / sample
        m12_0 = (ep12_0 - 2) ^ 2
        s_m12_0 = s_m12_0 + m12_0

        ep12_4 = m1s2_4 / sample
        m12_4 = (ep12_4 - 2.4) ^ 2
        s_m12_4 = s_m12_4 + m12_4

        ep12_9 = m1s2_9 / sample
        m12_9 = (ep12_9 - 2.9) ^ 2
        s_m12_9 = s_m12_9 + m12_9


End Function


Function cal_est2()

        ep21_3 = m2s1_3 / sample
        m21_3 = (ep21_3 - 1.3) ^ 2
        s_m21_3 = s_m21_3 + m21_3

        ep21_8 = m2s1_8 / sample
        m21_8 = (ep21_8 - 1.8) ^ 2
        s_m21_8 = s_m21_8 + m21_8
```

```
    ep22_0 = m2s2_0 / sample
    m22_0 = (ep22_0 - 2) ^ 2
    s_m22_0 = s_m22_0 + m22_0

    ep22_4 = m2s2_4 / sample
    m22_4 = (ep22_4 - 2.4) ^ 2
    s_m22_4 = s_m22_4 + m22_4

    ep22_9 = m2s2_9 / sample
    m22_9 = (ep22_9 - 2.9) ^ 2
    s_m22_9 = s_m22_9 + m22_9


End Function

Function cal_est3()

    ep31_3 = m3s1_3 / sample
    m31_3 = (ep31_3 - 1.3) ^ 2
    s_m31_3 = s_m31_3 + m31_3

    ep31_8 = m3s1_8 / sample
    m31_8 = (ep31_8 - 1.8) ^ 2
    s_m31_8 = s_m31_8 + m31_8

    ep32_0 = m3s2_0 / sample
    m32_0 = (ep32_0 - 2) ^ 2
    s_m32_0 = s_m32_0 + m32_0

    ep32_4 = m3s2_4 / sample
    m32_4 = (ep32_4 - 2.4) ^ 2
    s_m32_4 = s_m32_4 + m32_4

    ep32_9 = m3s2_9 / sample
    m32_9 = (ep32_9 - 2.9) ^ 2
    s_m32_9 = s_m32_9 + m32_9


End Function


Function cal_est4()

    ep41_3 = m4s1_3 / sample
    m41_3 = (ep41_3 - 1.3) ^ 2
    s_m41_3 = s_m41_3 + m41_3

    ep41_8 = m4s1_8 / sample
    m41_8 = (ep41_8 - 1.8) ^ 2
```

```
    s_m41_8 = s_m41_8 + m41_8

    ep42_0 = m4s2_0 / sample
    m42_0 = (ep42_0 - 2) ^ 2
    s_m42_0 = s_m42_0 + m42_0

    ep42_4 = m4s2_4 / sample
    m42_4 = (ep42_4 - 2.4) ^ 2
    s_m42_4 = s_m42_4 + m42_4

    ep42_9 = m4s2_9 / sample
    m42_9 = (ep42_9 - 2.9) ^ 2
    s_m42_9 = s_m42_9 + m42_9


End Function

Function cal_est5()

    ep51_3 = m5s1_3 / sample
    m51_3 = (ep51_3 - 1.3) ^ 2
    s_m51_3 = s_m51_3 + m51_3

    ep51_8 = m5s1_8 / sample
    m51_8 = (ep51_8 - 1.8) ^ 2
    s_m51_8 = s_m51_8 + m51_8

    ep52_0 = m5s2_0 / sample
    m52_0 = (ep52_0 - 2) ^ 2
    s_m52_0 = s_m52_0 + m52_0

    ep52_4 = m5s2_4 / sample
    m52_4 = (ep52_4 - 2.4) ^ 2
    s_m52_4 = s_m52_4 + m52_4

    ep52_9 = m5s2_9 / sample
    m52_9 = (ep52_9 - 2.9) ^ 2
    s_m52_9 = s_m52_9 + m52_9


End Function


Function poisson1()

    r11_3 = 0
    p11_3 = Exp(-1.3)
    F11_3 = p11_3
    If answer < F11_3 Then
        x11_3 = r11_3
```

```
Else
    Do Until answer < F11_3
        p11_3 = (1.3 * p11_3) / (r11_3 + 1)
        F11_3 = F11_3 + p11_3
        r11_3 = r11_3 + 1
        x11_3 = r11_3
    Loop
End If
s11_3 = s11_3 + x11_3


r11_8 = 0
p11_8 = Exp(-1.8)
F11_8 = p11_8
If answer < F11_8 Then
    x11_8 = r11_8
Else
    Do Until answer < F11_8
        p11_8 = (1.8 * p11_8) / (r11_8 + 1)
        F11_8 = F11_8 + p11_8
        r11_8 = r11_8 + 1
        x11_8 = r11_8
    Loop
End If
s11_8 = s11_8 + x11_8


r12_0 = 0
p12_0 = Exp(-2)
F12_0 = p12_0
If answer < F12_0 Then
    x12_0 = r12_0
Else
    Do Until answer < F12_0
        p12_0 = (2 * p12_0) / (r12_0 + 1)
        F12_0 = F12_0 + p12_0
        r12_0 = r12_0 + 1
        x12_0 = r12_0
    Loop
End If
s12_0 = s12_0 + x12_0


r12_4 = 0
p12_4 = Exp(-2.4)
F12_4 = p12_4
If answer < F12_4 Then
    x12_4 = r12_4
Else
    Do Until answer < F12_4
```

```
            p12_4 = (2.4 * p12_4) / (r12_4 + 1)
            F12_4 = F12_4 + p12_4
            r12_4 = r12_4 + 1
            x12_4 = r12_4
        Loop
    End If
    s12_4 = s12_4 + x12_4

    r12_9 = 0
    p12_9 = Exp(-2.9)
    F12_9 = p12_9
    If answer < F12_9 Then
        x12_9 = r12_9
    Else
        Do Until answer < F12_9
            p12_9 = (2.9 * p12_9) / (r12_9 + 1)
            F12_9 = F12_9 + p12_9
            r12_9 = r12_9 + 1
            x12_9 = r12_9
        Loop
    End If
    s12_9 = s12_9 + x12_9


End Function


Function poisson2()

    r21_3 = 0
    p21_3 = Exp(-1.3)
    F21_3 = p21_3
    If answer < F21_3 Then
        x21_3 = r21_3
    Else
        Do Until answer < F21_3
            p21_3 = (1.3 * p21_3) / (r21_3 + 1)
            F21_3 = F21_3 + p21_3
            r21_3 = r21_3 + 1
            x21_3 = r21_3
        Loop
    End If
    s21_3 = s21_3 + x21_3


    r21_8 = 0
    p21_8 = Exp(-1.8)
    F21_8 = p21_8
    If answer < F21_8 Then
        x21_8 = r21_8
```

```
Else
    Do Until answer < F21_8
        p21_8 = (1.8 * p21_8) / (r21_8 + 1)
        F21_8 = F21_8 + p21_8
        r21_8 = r21_8 + 1
        x21_8 = r21_8
    Loop
End If
s21_8 = s21_8 + x21_8


r22_0 = 0
p22_0 = Exp(-2)
F22_0 = p22_0
If answer < F22_0 Then
    x22_0 = r22_0
Else
    Do Until answer < F22_0
        p22_0 = (2 * p22_0) / (r22_0 + 1)
        F22_0 = F22_0 + p22_0
        r22_0 = r22_0 + 1
        x22_0 = r22_0
    Loop
End If
s22_0 = s22_0 + x22_0


r22_4 = 0
p22_4 = Exp(-2.4)
F22_4 = p22_4
If answer < F22_4 Then
    x22_4 = r22_4
Else
    Do Until answer < F22_4
        p22_4 = (2.4 * p22_4) / (r22_4 + 1)
        F22_4 = F22_4 + p22_4
        r22_4 = r22_4 + 1
        x22_4 = r22_4
    Loop
End If
s22_4 = s22_4 + x22_4

r22_9 = 0
p22_9 = Exp(-2.9)
F22_9 = p22_9
If answer < F22_9 Then
    x22_9 = r22_9
Else
    Do Until answer < F22_9
        p22_9 = (2.9 * p22_9) / (r22_9 + 1)
```

```
            F22_9 = F22_9 + p22_9
            r22_9 = r22_9 + 1
            x22_9 = r22_9
        Loop
    End If
    s22_9 = s22_9 + x22_9


End Function

Function poisson3()

    r31_3 = 0
    p31_3 = Exp(-1.3)
    F31_3 = p31_3
    If answer < F31_3 Then
        x31_3 = r31_3
    Else
        Do Until answer < F31_3
            p31_3 = (1.3 * p31_3) / (r31_3 + 1)
            F31_3 = F31_3 + p31_3
            r31_3 = r31_3 + 1
            x31_3 = r31_3
        Loop
    End If
    s31_3 = s31_3 + x31_3


    r31_8 = 0
    p31_8 = Exp(-1.8)
    F31_8 = p31_8
    If answer < F31_8 Then
        x31_8 = r31_8
    Else
        Do Until answer < F31_8
            p31_8 = (1.8 * p31_8) / (r31_8 + 1)
            F31_8 = F31_8 + p31_8
            r31_8 = r31_8 + 1
            x31_8 = r31_8
        Loop
    End If
    s31_8 = s31_8 + x31_8


    r32_0 = 0
    p32_0 = Exp(-2)
    F32_0 = p32_0
    If answer < F32_0 Then
        x32_0 = r32_0
    Else
```

```
        Do Until answer < F32_0
            p32_0 = (2 * p32_0) / (r32_0 + 1)
            F32_0 = F32_0 + p32_0
            r32_0 = r32_0 + 1
            x32_0 = r32_0
        Loop
    End If
    s32_0 = s32_0 + x32_0


    r32_4 = 0
    p32_4 = Exp(-2.4)
    F32_4 = p32_4
    If answer < F32_4 Then
        x32_4 = r32_4
    Else
        Do Until answer < F32_4
            p32_4 = (2.4 * p32_4) / (r32_4 + 1)
            F32_4 = F32_4 + p32_4
            r32_4 = r32_4 + 1
            x32_4 = r32_4
        Loop
    End If
    s32_4 = s32_4 + x32_4

    r32_9 = 0
    p32_9 = Exp(-2.9)
    F32_9 = p32_9
    If answer < F32_9 Then
        x32_9 = r32_9
    Else
        Do Until answer < F32_9
            p32_9 = (2.9 * p32_9) / (r32_9 + 1)
            F32_9 = F32_9 + p32_9
            r32_9 = r32_9 + 1
            x32_9 = r32_9
        Loop
    End If
    s32_9 = s32_9 + x32_9


End Function


Function poisson4()

    r41_3 = 0
    p41_3 = Exp(-1.3)
    F41_3 = p41_3
```

```
If answer < F41_3 Then
    x41_3 = r41_3
Else
    Do Until answer < F41_3
         p41_3 = (1.3 * p41_3) / (r41_3 + 1)
         F41_3 = F41_3 + p41_3
         r41_3 = r41_3 + 1
         x41_3 = r41_3
    Loop
End If
s41_3 = s41_3 + x41_3


r41_8 = 0
p41_8 = Exp(-1.8)
F41_8 = p41_8
If answer < F41_8 Then
    x41_8 = r41_8
Else
    Do Until answer < F41_8
         p41_8 = (1.8 * p41_8) / (r41_8 + 1)
         F41_8 = F41_8 + p41_8
         r41_8 = r41_8 + 1
         x41_8 = r41_8
    Loop
End If
s41_8 = s41_8 + x41_8


r42_0 = 0
p42_0 = Exp(-2)
F42_0 = p42_0
If answer < F42_0 Then
    x42_0 = r42_0
Else
    Do Until answer < F42_0
         p42_0 = (2 * p42_0) / (r42_0 + 1)
         F42_0 = F42_0 + p42_0
         r42_0 = r42_0 + 1
         x42_0 = r42_0
    Loop
End If
s42_0 = s42_0 + x42_0


r42_4 = 0
p42_4 = Exp(-2.4)
F42_4 = p42_4
If answer < F42_4 Then
    x42_4 = r42_4
```

```
    Else
        Do Until answer < F42_4
            p42_4 = (2.4 * p42_4) / (r42_4 + 1)
            F42_4 = F42_4 + p42_4
            r42_4 = r42_4 + 1
            x42_4 = r42_4
        Loop
    End If
    s42_4 = s42_4 + x42_4

    r42_9 = 0
    p42_9 = Exp(-2.9)
    F42_9 = p42_9
    If answer < F42_9 Then
        x42_9 = r42_9
    Else
        Do Until answer < F42_9
            p42_9 = (2.9 * p42_9) / (r42_9 + 1)
            F42_9 = F42_9 + p42_9
            r42_9 = r42_9 + 1
            x42_9 = r42_9
        Loop
    End If
    s42_9 = s42_9 + x42_9


End Function


Function poisson5()

    r51_3 = 0
    p51_3 = Exp(-1.3)
    F51_3 = p51_3
    If answer < F51_3 Then
        x51_3 = r51_3
    Else
        Do Until answer < F51_3
            p51_3 = (1.3 * p51_3) / (r51_3 + 1)
            F51_3 = F51_3 + p51_3
            r51_3 = r51_3 + 1
            x51_3 = r51_3
        Loop
    End If
    s51_3 = s51_3 + x51_3


    r51_8 = 0
    p51_8 = Exp(-1.8)
    F51_8 = p51_8
```

```
If answer < F51_8 Then
    x51_8 = r51_8
Else
    Do Until answer < F51_8
        p51_8 = (1.8 * p51_8) / (r51_8 + 1)
        F51_8 = F51_8 + p51_8
        r51_8 = r51_8 + 1
        x51_8 = r51_8
    Loop
End If
s51_8 = s51_8 + x51_8


r52_0 = 0
p52_0 = Exp(-2)
F52_0 = p52_0
If answer < F52_0 Then
    x52_0 = r52_0
Else
    Do Until answer < F52_0
        p52_0 = (2 * p52_0) / (r52_0 + 1)
        F52_0 = F52_0 + p52_0
        r52_0 = r52_0 + 1
        x52_0 = r52_0
    Loop
End If
s52_0 = s52_0 + x52_0


r52_4 = 0
p52_4 = Exp(-2.4)
F52_4 = p52_4
If answer < F52_4 Then
    x52_4 = r52_4
Else
    Do Until answer < F52_4
        p52_4 = (2.4 * p52_4) / (r52_4 + 1)
        F52_4 = F52_4 + p52_4
        r52_4 = r52_4 + 1
        x52_4 = r52_4
    Loop
End If
s52_4 = s52_4 + x52_4

r52_9 = 0
p52_9 = Exp(-2.9)
F52_9 = p52_9
If answer < F52_9 Then
    x52_9 = r52_9
Else
```

```
        Do Until answer < F52_9
            p52_9 = (2.9 * p52_9) / (r52_9 + 1)
            F52_9 = F52_9 + p52_9
            r52_9 = r52_9 + 1
            x52_9 = r52_9
        Loop
    End If
    s52_9 = s52_9 + x52_9


End Function


Function cal_est_poi1()

    epp11_3 = s11_3 / sample
    m1p1_3 = (epp11_3 - 1.3) ^ 2
    sp_m11_3 = sp_m11_3 + m1p1_3

    epp11_8 = s11_8 / sample
    m1p1_8 = (epp11_8 - 1.8) ^ 2
    sp_m11_8 = sp_m11_8 + m1p1_8

    epp12_0 = s12_0 / sample
    m1p2_0 = (epp12_0 - 2) ^ 2
    sp_m12_0 = sp_m12_0 + m1p2_0

    epp12_4 = s12_4 / sample
    m1p2_4 = (epp12_4 - 2.4) ^ 2
    sp_m12_4 = sp_m12_4 + m1p2_4

    epp12_9 = s12_9 / sample
    m1p2_9 = (epp12_9 - 2.9) ^ 2
    sp_m12_9 = sp_m12_9 + m1p2_9

End Function


Function cal_est_poi2()

    epp21_3 = s21_3 / sample
    m2p1_3 = (epp21_3 - 1.3) ^ 2
    sp_m21_3 = sp_m21_3 + m2p1_3

    epp21_8 = s21_8 / sample
    m2p1_8 = (epp21_8 - 1.8) ^ 2
    sp_m21_8 = sp_m21_8 + m2p1_8

    epp22_0 = s22_0 / sample
    m2p2_0 = (epp22_0 - 2) ^ 2
```

```
    sp_m22_0 = sp_m22_0 + m2p2_0

    epp22_4 = s22_4 / sample
    m2p2_4 = (epp22_4 - 2.4) ^ 2
    sp_m22_4 = sp_m22_4 + m2p2_4

    epp22_9 = s22_9 / sample
    m2p2_9 = (epp22_9 - 2.9) ^ 2
    sp_m22_9 = sp_m22_9 + m2p2_9


End Function


Function cal_est_poi3()

    epp31_3 = s31_3 / sample
    m3p1_3 = (epp31_3 - 1.3) ^ 2
    sp_m31_3 = sp_m31_3 + m3p1_3

    epp31_8 = s31_8 / sample
    m3p1_8 = (epp31_8 - 1.8) ^ 2
    sp_m31_8 = sp_m31_8 + m3p1_8

    epp32_0 = s32_0 / sample
    m3p2_0 = (epp32_0 - 2) ^ 2
    sp_m32_0 = sp_m32_0 + m3p2_0

    epp32_4 = s32_4 / sample
    m3p2_4 = (epp32_4 - 2.4) ^ 2
    sp_m32_4 = sp_m32_4 + m3p2_4

    epp32_9 = s32_9 / sample
    m3p2_9 = (epp32_9 - 2.9) ^ 2
    sp_m32_9 = sp_m32_9 + m3p2_9


End Function

Function cal_est_poi4()

    epp41_3 = s41_3 / sample
    m4p1_3 = (epp41_3 - 1.3) ^ 2
    sp_m41_3 = sp_m41_3 + m4p1_3

    epp41_8 = s41_8 / sample
    m4p1_8 = (epp41_8 - 1.8) ^ 2
    sp_m41_8 = sp_m41_8 + m4p1_8

    epp42_0 = s42_0 / sample
```

```
    m4p2_0 = (epp42_0 - 2) ^ 2
    sp_m42_0 = sp_m42_0 + m4p2_0

    epp42_4 = s42_4 / sample
    m4p2_4 = (epp42_4 - 2.4) ^ 2
    sp_m42_4 = sp_m42_4 + m4p2_4

    epp42_9 = s42_9 / sample
    m4p2_9 = (epp42_9 - 2.9) ^ 2
    sp_m42_9 = sp_m42_9 + m4p2_9


End Function

Function cal_est_poi5()

    epp51_3 = s51_3 / sample
    m5p1_3 = (epp51_3 - 1.3) ^ 2
    sp_m51_3 = sp_m51_3 + m5p1_3

    epp51_8 = s51_8 / sample
    m5p1_8 = (epp51_8 - 1.8) ^ 2
    sp_m51_8 = sp_m51_8 + m5p1_8

    epp52_0 = s52_0 / sample
    m5p2_0 = (epp52_0 - 2) ^ 2
    sp_m52_0 = sp_m52_0 + m5p2_0

    epp52_4 = s52_4 / sample
    m5p2_4 = (epp52_4 - 2.4) ^ 2
    sp_m52_4 = sp_m52_4 + m5p2_4

    epp52_9 = s52_9 / sample
    m5p2_9 = (epp52_9 - 2.9) ^ 2
    sp_m52_9 = sp_m52_9 + m5p2_9


End Function

Function mse_poisson1()

    p1mle1_3 = (sp_m11_3 / loop_test)
    rp1mle1_3 = Round(Sqr(p1mle1_3), 4)

    p1mle1_8 = (sp_m11_8 / loop_test)
    rp1mle1_8 = Round(Sqr(p1mle1_8), 4)

    p1mle2_0 = (sp_m12_0 / loop_test)
    rp1mle2_0 = Round(Sqr(p1mle2_0), 4)
```

```
        p1mle2_4 = (sp_m12_4 / loop_test)
        rp1mle2_4 = Round(Sqr(p1mle2_4), 4)

        p1mle2_9 = (sp_m12_9 / loop_test)
        rp1mle2_9 = Round(Sqr(p1mle2_9), 4)

Print "                              "
Print "M1: RMSE POI 1.3 =  " & rp1mle1_3, "1.8 = " & rp1mle1_8, "2.0= " &
rp1mle2_0, "2.4 = " & rp1mle2_4, "2.9 = " & rp1mle2_9


End Function


Function mse_poisson2()

        p2mle1_3 = (sp_m21_3 / loop_test)
        rp2mle1_3 = Round(Sqr(p2mle1_3), 4)

        p2mle1_8 = (sp_m21_8 / loop_test)
        rp2mle1_8 = Round(Sqr(p2mle1_8), 4)

        p2mle2_0 = (sp_m22_0 / loop_test)
        rp2mle2_0 = Round(Sqr(p2mle2_0), 4)

        p2mle2_4 = (sp_m22_4 / loop_test)
        rp2mle2_4 = Round(Sqr(p2mle2_4), 4)

        p2mle2_9 = (sp_m22_9 / loop_test)
        rp2mle2_9 = Round(Sqr(p2mle2_9), 4)

 Print "M2 : RMSE POI 1.3 =  " & rp2mle1_3, "1.8 = " & rp2mle1_8, "2.0= " &
rp2mle2_0, "2.4 = " & rp2mle2_4, "2.9 = " & rp2mle2_9


End Function


Function mse_poisson3()

        p3mle1_3 = (sp_m31_3 / loop_test)
        rp3mle1_3 = Round(Sqr(p3mle1_3), 4)

        p3mle1_8 = (sp_m31_8 / loop_test)
        rp3mle1_8 = Round(Sqr(p3mle1_8), 4)

        p3mle2_0 = (sp_m32_0 / loop_test)
        rp3mle2_0 = Round(Sqr(p3mle2_0), 4)

        p3mle2_4 = (sp_m32_4 / loop_test)
```

```
        rp3mle2_4 = Round(Sqr(p3mle2_4), 4)
        p3mle2_9 = (sp_m32_9 / loop_test)
        rp3mle2_9 = Round(Sqr(p3mle2_9), 4)

Print "M3 : RMSE POI 1.3 =  " & rp3mle1_3, "1.8 = " & rp3mle1_8, "2.0= " &
rp3mle2_0, "2.4 = " & rp3mle2_4, "2.9 = " & rp3mle2_9

End Function


Function mse_poisson4()

        p4mle1_3 = (sp_m41_3 / loop_test)
        rp4mle1_3 = Round(Sqr(p4mle1_3), 4)

        p4mle1_8 = (sp_m41_8 / loop_test)
        rp4mle1_8 = Round(Sqr(p4mle1_8), 4)

        p4mle2_0 = (sp_m42_0 / loop_test)
        rp4mle2_0 = Round(Sqr(p4mle2_0), 4)

        p4mle2_4 = (sp_m42_4 / loop_test)
        rp4mle2_4 = Round(Sqr(p4mle2_4), 4)

        p4mle2_9 = (sp_m42_9 / loop_test)
        rp4mle2_9 = Round(Sqr(p4mle2_9), 4)

Print "M4 : RMSE POI 1.3 = " & rp4mle1_3, "1.8 = " & rp4mle1_8, "2.0= " &
rp4mle2_0, "2.4 =" & rp4mle2_4, "2.9 =" & rp4mle2_9

End Function

Function mse_poisson5()

        p5mle1_3 = (sp_m51_3 / loop_test)
        rp5mle1_3 = Round(Sqr(p5mle1_3), 4)

        p5mle1_8 = (sp_m51_8 / loop_test)
        rp5mle1_8 = Round(Sqr(p5mle1_8), 4)

        p5mle2_0 = (sp_m52_0 / loop_test)
        rp5mle2_0 = Round(Sqr(p5mle2_0), 4)

        p5mle2_4 = (sp_m52_4 / loop_test)
        rp5mle2_4 = Round(Sqr(p5mle2_4), 4)

        p5mle2_9 = (sp_m52_9 / loop_test)
        rp5mle2_9 = Round(Sqr(p5mle2_9), 4)
```

```
Print "M5 : RMSE POI 1.3 = " & rp5mle1_3, "1.8 =" & rp5mle1_8, " 2.0=" &
rp5mle2_0, "2.4 =" & rp5mle2_4, "2.9 =" & rp5mle2_9

End Function


Function binomial1()

        M1i20_01 = 0
        M1c20_01 = 0.1 / (1 - 0.1)
        M1pr20_01 = (1 - 0.1) ^ 20
        M1F20_01 = M1pr20_01
        If answer < M1F20_01 Then
                M1x20_01 = M1i20_01
        Else
            Do Until answer < M1F20_01
                M1pr20_01 = (M1c20_01 * (20 - M1i20_01) / (M1i20_01 + 1)) *
M1pr20_01
                M1F20_01 = M1F20_01 + M1pr20_01
                M1i20_01 = M1i20_01 + 1
                M1x20_01 = M1i20_01
            Loop
        End If
        s_M1x20_01 = s_M1x20_01 + M1x20_01


        M1i30_03 = 0
        M1c30_03 = 0.3 / (1 - 0.3)
        M1pr30_03 = (1 - 0.3) ^ 30
        M1F30_03 = M1pr30_03
        If answer < M1F30_03 Then
                M1x30_03 = M1i30_03
        Else
            Do Until answer < M1F30_03
                M1pr30_03 = (M1c30_03 * (30 - M1i30_03) / (M1i30_03 + 1)) *
M1pr30_03
                M1F30_03 = M1F30_03 + M1pr30_03
                M1i30_03 = M1i30_03 + 1
                M1x30_03 = M1i30_03
            Loop
        End If
        s_M1x30_03 = s_M1x30_03 + M1x30_03


        M1i40_05 = 0
        M1c40_05 = 0.5 / (1 - 0.5)
        M1pr40_05 = (1 - 0.5) ^ 40
        M1F40_05 = M1pr40_05
        If answer < M1F40_05 Then
                M1x40_05 = M1i40_05
```

```
        Else
            Do Until answer < M1F40_05
                M1pr40_05 = (M1c40_05 * (40 - M1i40_05) / (M1i40_05 + 1)) *
M1pr40_05
                M1F40_05 = M1F40_05 + M1pr40_05
                M1i40_05 = M1i40_05 + 1
                M1x40_05 = M1i40_05
            Loop
        End If
        s_M1x40_05 = s_M1x40_05 + M1x40_05


        M1i50_06 = 0
        M1c50_06 = 0.6 / (1 - 0.6)
        M1pr50_06 = (1 - 0.6) ^ 50
        M1F50_06 = M1pr50_06
        If answer < M1F50_06 Then
                M1x50_06 = M1i50_06
        Else
            Do Until answer < M1F50_06
                M1pr50_06 = (M1c50_06 * (50 - M1i50_06) / (M1i50_06 + 1)) *
M1pr50_06
                M1F50_06 = M1F50_06 + M1pr50_06
                M1i50_06 = M1i50_06 + 1
                M1x50_06 = M1i50_06
            Loop
        End If
        s_M1x50_06 = s_M1x50_06 + M1x50_06


         M1i70_08 = 0
        M1c70_08 = 0.8 / (1 - 0.8)
        M1pr70_08 = (1 - 0.8) ^ 70
        M1F70_08 = M1pr70_08
        If answer < M1F70_08 Then
                M1x70_08 = M1i70_08
        Else
            Do Until answer < M1F70_08
                M1pr70_08 = (M1c70_08 * (70 - M1i70_08) / (M1i70_08 + 1)) *
M1pr70_08
                M1F70_08 = M1F70_08 + M1pr70_08
                M1i70_08 = M1i70_08 + 1
                M1x70_08 = M1i70_08
            Loop
        End If
        s_M1x70_08 = s_M1x70_08 + M1x70_08


End Function
```

```
Function binomial2()

        M2i20_01 = 0
        M2c20_01 = 0.1 / (1 - 0.1)
        M2pr20_01 = (1 - 0.1) ^ 20
        M2F20_01 = M2pr20_01
        If answer < M2F20_01 Then
                M2x20_01 = M2i20_01
        Else
            Do Until answer < M2F20_01
                M2pr20_01 = (M2c20_01 * (20 - M2i20_01) / (M2i20_01 + 1)) *
M2pr20_01
                M2F20_01 = M2F20_01 + M2pr20_01
                M2i20_01 = M2i20_01 + 1
                M2x20_01 = M2i20_01
            Loop
        End If
        s_M2x20_01 = s_M2x20_01 + M2x20_01


        M2i30_03 = 0
        M2c30_03 = 0.3 / (1 - 0.3)
        M2pr30_03 = (1 - 0.3) ^ 30
        M2F30_03 = M2pr30_03
        If answer < M2F30_03 Then
                M2x30_03 = M2i30_03
        Else
            Do Until answer < M2F30_03
                M2pr30_03 = (M2c30_03 * (30 - M2i30_03) / (M2i30_03 + 1)) *
M2pr30_03
                M2F30_03 = M2F30_03 + M2pr30_03
                M2i30_03 = M2i30_03 + 1
                M2x30_03 = M2i30_03
            Loop
        End If
        s_M2x30_03 = s_M2x30_03 + M2x30_03


        M2i40_05 = 0
        M2c40_05 = 0.5 / (1 - 0.5)
        M2pr40_05 = (1 - 0.5) ^ 40
        M2F40_05 = M2pr40_05
        If answer < M2F40_05 Then
                M2x40_05 = M2i40_05
        Else
            Do Until answer < M2F40_05
                M2pr40_05 = (M2c40_05 * (40 - M2i40_05) / (M2i40_05 + 1)) *
M2pr40_05
                M2F40_05 = M2F40_05 + M2pr40_05
```

```
                M2i40_05 = M2i40_05 + 1
                M2x40_05 = M2i40_05
            Loop
        End If
        s_M2x40_05 = s_M2x40_05 + M2x40_05


        M2i50_06 = 0
        M2c50_06 = 0.6 / (1 - 0.6)
        M2pr50_06 = (1 - 0.6) ^ 50
        M2F50_06 = M2pr50_06
        If answer < M2F50_06 Then
                M2x50_06 = M2i50_06
        Else
            Do Until answer < M2F50_06
                M2pr50_06 = (M2c50_06 * (50 - M2i50_06) / (M2i50_06 + 1)) *
M2pr50_06
                M2F50_06 = M2F50_06 + M2pr50_06
                M2i50_06 = M2i50_06 + 1
                M2x50_06 = M2i50_06
            Loop
        End If
        s_M2x50_06 = s_M2x50_06 + M2x50_06


        M2i70_08 = 0
        M2c70_08 = 0.8 / (1 - 0.8)
        M2pr70_08 = (1 - 0.8) ^ 70
        M2F70_08 = M2pr70_08
        If answer < M2F70_08 Then
                M2x70_08 = M2i70_08
        Else
            Do Until answer < M2F70_08
                M2pr70_08 = (M2c70_08 * (70 - M2i70_08) / (M2i70_08 + 1)) *
M2pr70_08
                M2F70_08 = M2F70_08 + M2pr70_08
                M2i70_08 = M2i70_08 + 1
                M2x70_08 = M2i70_08
            Loop
        End If
        s_M2x70_08 = s_M2x70_08 + M2x70_08

End Function

Function binomial3()

        M3i20_01 = 0
        M3c20_01 = 0.1 / (1 - 0.1)
        M3pr20_01 = (1 - 0.1) ^ 20
        M3F20_01 = M3pr20_01
```

```
        If answer < M3F20_01 Then
                M3x20_01 = M3i20_01
        Else
            Do Until answer < M3F20_01
                M3pr20_01 = (M3c20_01 * (20 - M3i20_01) / (M3i20_01 + 1)) *
M3pr20_01
                M3F20_01 = M3F20_01 + M3pr20_01
                M3i20_01 = M3i20_01 + 1
                M3x20_01 = M3i20_01
            Loop
        End If
        s_M3x20_01 = s_M3x20_01 + M3x20_01


        M3i30_03 = 0
        M3c30_03 = 0.3 / (1 - 0.3)
        M3pr30_03 = (1 - 0.3) ^ 30
        M3F30_03 = M3pr30_03
        If answer < M3F30_03 Then
                M3x30_03 = M3i30_03
        Else
            Do Until answer < M3F30_03
                M3pr30_03 = (M3c30_03 * (30 - M3i30_03) / (M3i30_03 + 1)) *
M3pr30_03
                M3F30_03 = M3F30_03 + M3pr30_03
                M3i30_03 = M3i30_03 + 1
                M3x30_03 = M3i30_03
            Loop
        End If
        s_M3x30_03 = s_M3x30_03 + M3x30_03


        M3i40_05 = 0
        M3c40_05 = 0.5 / (1 - 0.5)
        M3pr40_05 = (1 - 0.5) ^ 40
        M3F40_05 = M3pr40_05
        If answer < M3F40_05 Then
                M3x40_05 = M3i40_05
        Else
            Do Until answer < M3F40_05
                M3pr40_05 = (M3c40_05 * (40 - M3i40_05) / (M3i40_05 + 1)) *
M3pr40_05
                M3F40_05 = M3F40_05 + M3pr40_05
                M3i40_05 = M3i40_05 + 1
                M3x40_05 = M3i40_05
            Loop
        End If
        s_M3x40_05 = s_M3x40_05 + M3x40_05
```

```
        M3i50_06 = 0
        M3c50_06 = 0.6 / (1 - 0.6)
        M3pr50_06 = (1 - 0.6) ^ 50
        M3F50_06 = M3pr50_06
        If answer < M3F50_06 Then
                M3x50_06 = M3i50_06
        Else
            Do Until answer < M3F50_06
                M3pr50_06 = (M3c50_06 * (50 - M3i50_06) / (M3i50_06 + 1)) *
M3pr50_06
                M3F50_06 = M3F50_06 + M3pr50_06
                M3i50_06 = M3i50_06 + 1
                M3x50_06 = M3i50_06
            Loop
        End If
        s_M3x50_06 = s_M3x50_06 + M3x50_06


        M3i70_08 = 0
        M3c70_08 = 0.8 / (1 - 0.8)
        M3pr70_08 = (1 - 0.8) ^ 70
        M3F70_08 = M3pr70_08
        If answer < M3F70_08 Then
                M3x70_08 = M2i70_08
        Else
            Do Until answer < M3F70_08
                M3pr70_08 = (M3c70_08 * (70 - M3i70_08) / (M3i70_08 + 1)) *
M3pr70_08
                M3F70_08 = M3F70_08 + M3pr70_08
                M3i70_08 = M3i70_08 + 1
                M3x70_08 = M3i70_08
            Loop
        End If
        s_M3x70_08 = s_M3x70_08 + M3x70_08


End Function


Function binomial4()

        M4i20_01 = 0
        M4c20_01 = 0.1 / (1 - 0.1)
        M4pr20_01 = (1 - 0.1) ^ 20
        M4F20_01 = M4pr20_01
        If answer < M4F20_01 Then
                M4x20_01 = M4i20_01
        Else
            Do Until answer < M4F20_01
```

```
                M4pr20_01 = (M4c20_01 * (20 - M4i20_01) / (M4i20_01 + 1)) *
M4pr20_01
                M4F20_01 = M4F20_01 + M4pr20_01
                M4i20_01 = M4i20_01 + 1
                M4x20_01 = M4i20_01
            Loop
        End If
        s_M4x20_01 = s_M4x20_01 + M4x20_01


        M4i30_03 = 0
        M4c30_03 = 0.3 / (1 - 0.3)
        M4pr30_03 = (1 - 0.3) ^ 30
        M4F30_03 = M4pr30_03
        If answer < M4F30_03 Then
                M4x30_03 = M4i30_03
        Else
            Do Until answer < M4F30_03
                M4pr30_03 = (M4c30_03 * (30 - M4i30_03) / (M4i30_03 + 1)) *
M4pr30_03
                M4F30_03 = M4F30_03 + M4pr30_03
                M4i30_03 = M4i30_03 + 1
                M4x30_03 = M4i30_03
            Loop
        End If
        s_M4x30_03 = s_M4x30_03 + M4x30_03


        M4i40_05 = 0
        M4c40_05 = 0.5 / (1 - 0.5)
        M4pr40_05 = (1 - 0.5) ^ 40
        M4F40_05 = M4pr40_05
        If answer < M4F40_05 Then
                M4x40_05 = M4i40_05
        Else
            Do Until answer < M4F40_05
                M4pr40_05 = (M4c40_05 * (40 - M4i40_05) / (M4i40_05 + 1)) *
M4pr40_05
                M4F40_05 = M4F40_05 + M4pr40_05
                M4i40_05 = M4i40_05 + 1
                M4x40_05 = M4i40_05
            Loop
        End If
        s_M4x40_05 = s_M4x40_05 + M4x40_05


        M4i50_06 = 0
        M4c50_06 = 0.6 / (1 - 0.6)
        M4pr50_06 = (1 - 0.6) ^ 50
        M4F50_06 = M4pr50_06
```

```
        If answer < M4F50_06 Then
                M4x50_06 = M4i50_06
        Else
            Do Until answer < M4F50_06
                M4pr50_06 = (M4c50_06 * (50 - M4i50_06) / (M4i50_06 + 1)) *
M4pr50_06
                M4F50_06 = M4F50_06 + M4pr50_06
                M4i50_06 = M4i50_06 + 1
                M4x50_06 = M4i50_06
            Loop
        End If
        s_M4x50_06 = s_M4x50_06 + M4x50_06


        M4i70_08 = 0
        M4c70_08 = 0.8 / (1 - 0.8)
        M4pr70_08 = (1 - 0.8) ^ 70
        M4F70_08 = M4pr70_08
        If answer < M4F70_08 Then
                M4x70_08 = M4i70_08
        Else
            Do Until answer < M4F70_08
                M4pr70_08 = (M4c70_08 * (70 - M4i70_08) / (M4i70_08 + 1)) *
M4pr70_08
                M4F70_08 = M4F70_08 + M4pr70_08
                M4i70_08 = M4i70_08 + 1
                M4x70_08 = M4i70_08
            Loop
        End If
        s_M4x70_08 = s_M4x70_08 + M4x70_08


End Function


Function binomial5()

        M5i20_01 = 0
        M5c20_01 = 0.1 / (1 - 0.1)
        M5pr20_01 = (1 - 0.1) ^ 20
        M5F20_01 = M5pr20_01
        If answer < M5F20_01 Then
                M5x20_01 = M5i20_01
        Else
            Do Until answer < M5F20_01
                M5pr20_01 = (M5c20_01 * (20 - M5i20_01) / (M5i20_01 + 1)) *
M5pr20_01
                M5F20_01 = M5F20_01 + M5pr20_01
                M5i20_01 = M5i20_01 + 1
                M5x20_01 = M5i20_01
```

```
        Loop
    End If
    s_M5x20_01 = s_M5x20_01 + M5x20_01


    M5i30_03 = 0
    M5c30_03 = 0.3 / (1 - 0.3)
    M5pr30_03 = (1 - 0.3) ^ 30
    M5F30_03 = M5pr30_03
    If answer < M5F30_03 Then
            M5x30_03 = M5i30_03
    Else
        Do Until answer < M5F30_03
            M5pr30_03 = (M5c30_03 * (30 - M5i30_03) / (M5i30_03 + 1)) *
M5pr30_03
            M5F30_03 = M5F30_03 + M5pr30_03
            M5i30_03 = M5i30_03 + 1
            M5x30_03 = M5i30_03
        Loop
    End If
    s_M5x30_03 = s_M5x30_03 + M5x30_03


    M5i40_05 = 0
    M5c40_05 = 0.5 / (1 - 0.5)
    M5pr40_05 = (1 - 0.5) ^ 40
    M5F40_05 = M5pr40_05
    If answer < M5F40_05 Then
            M5x40_05 = M5i40_05
    Else
        Do Until answer < M5F40_05
            M5pr40_05 = (M5c40_05 * (40 - M5i40_05) / (M5i40_05 + 1)) *
M5pr40_05
            M5F40_05 = M5F40_05 + M5pr40_05
            M5i40_05 = M5i40_05 + 1
            M5x40_05 = M5i40_05
        Loop
    End If
    s_M5x40_05 = s_M5x40_05 + M5x40_05


    M5i50_06 = 0
    M5c50_06 = 0.6 / (1 - 0.6)
    M5pr50_06 = (1 - 0.6) ^ 50
    M5F50_06 = M5pr50_06
    If answer < M5F50_06 Then
            M5x50_06 = M5i50_06
    Else
        Do Until answer < M5F50_06
```

```
                M5pr50_06 = (M5c50_06 * (50 - M5i50_06) / (M5i50_06 + 1)) *
M5pr50_06
                M5F50_06 = M5F50_06 + M5pr50_06
                M5i50_06 = M5i50_06 + 1
                M5x50_06 = M5i50_06
            Loop
        End If
        s_M5x50_06 = s_M5x50_06 + M5x50_06


        M5i70_08 = 0
        M5c70_08 = 0.8 / (1 - 0.8)
        M5pr70_08 = (1 - 0.8) ^ 70
        M5F70_08 = M5pr70_08
        If answer < M5F70_08 Then
                M5x70_08 = M5i70_08
        Else
            Do Until answer < M5F70_08
                M5pr70_08 = (M5c70_08 * (70 - M5i70_08) / (M5i70_08 + 1)) *
M5pr70_08
                M5F70_08 = M5F70_08 + M5pr70_08
                M5i70_08 = M5i70_08 + 1
                M5x70_08 = M5i70_08
            Loop
        End If
        s_M5x70_08 = s_M5x70_08 + M5x70_08


End Function


Function cal_est_binomial1()

        eBM120_01 = s_M1x20_01 / sample
        M120_01 = (eBM120_01 - (20 * 0.1)) ^ 2
        s_M120_01 = s_M120_01 + M120_01

        eBM130_03 = s_M1x30_03 / sample
        M130_03 = (eBM130_03 - (30 * 0.3)) ^ 2
        s_M130_03 = s_M130_03 + M130_03

        eBM140_05 = s_M1x40_05 / sample
        M140_05 = (eBM140_05 - (40 * 0.5)) ^ 2
        s_M140_05 = s_M140_05 + M140_05

        eBM150_06 = s_M1x50_06 / sample
        M150_06 = (eBM150_06 - (50 * 0.6)) ^ 2
        s_M150_06 = s_M150_06 + M150_06

        eBM170_08 = s_M1x70_08 / sample
```

```
    M170_08 = (eBM170_08 - (70 * 0.8)) ^ 2
    s_M170_08 = s_M170_08 + M170_08

End Function

Function cal_est_binomial2()

    eBM220_01 = s_M2x20_01 / sample
    M220_01 = (eBM220_01 - (20 * 0.1)) ^ 2
    s_M220_01 = s_M220_01 + M220_01

    eBM230_03 = s_M2x30_03 / sample
    M230_03 = (eBM230_03 - (30 * 0.3)) ^ 2
    s_M230_03 = s_M230_03 + M230_03

    eBM240_05 = s_M2x40_05 / sample
    M240_05 = (eBM240_05 - (40 * 0.5)) ^ 2
    s_M240_05 = s_M240_05 + M240_05

    eBM250_06 = s_M2x50_06 / sample
    M250_06 = (eBM250_06 - (50 * 0.6)) ^ 2
    s_M250_06 = s_M250_06 + M250_06

    eBM270_08 = s_M2x70_08 / sample
    M270_08 = (eBM270_08 - (70 * 0.8)) ^ 2
    s_M270_08 = s_M270_08 + M270_08

End Function

Function cal_est_binomial3()

    eBM320_01 = s_M3x20_01 / sample
    M320_01 = (eBM320_01 - (20 * 0.1)) ^ 2
    s_M320_01 = s_M320_01 + M320_01

    eBM330_03 = s_M3x30_03 / sample
    M330_03 = (eBM330_03 - (30 * 0.3)) ^ 2
    s_M330_03 = s_M330_03 + M330_03

    eBM340_05 = s_M3x40_05 / sample
    M340_05 = (eBM340_05 - (40 * 0.5)) ^ 2
    s_M340_05 = s_M340_05 + M340_05

    eBM350_06 = s_M3x50_06 / sample
    M350_06 = (eBM350_06 - (50 * 0.6)) ^ 2
    s_M350_06 = s_M350_06 + M350_06

    eBM370_08 = s_M3x70_08 / sample
    M370_08 = (eBM370_08 - (70 * 0.8)) ^ 2
    s_M370_08 = s_M370_08 + M370_08
```

```
End Function

Function cal_est_binomial4()

        eBM420_01 = s_M4x20_01 / sample
        M420_01 = (eBM420_01 - (20 * 0.1)) ^ 2
        s_M420_01 = s_M420_01 + M420_01

        eBM430_03 = s_M4x30_03 / sample
        M430_03 = (eBM430_03 - (30 * 0.3)) ^ 2
        s_M430_03 = s_M430_03 + M430_03

        eBM440_05 = s_M4x40_05 / sample
        M440_05 = (eBM440_05 - (40 * 0.5)) ^ 2
        s_M440_05 = s_M440_05 + M440_05

        eBM450_06 = s_M4x50_06 / sample
        M450_06 = (eBM450_06 - (50 * 0.6)) ^ 2
        s_M450_06 = s_M450_06 + M450_06

        eBM470_08 = s_M4x70_08 / sample
        M470_08 = (eBM470_08 - (70 * 0.8)) ^ 2
        s_M470_08 = s_M470_08 + M470_08

End Function


Function cal_est_binomial5()

        eBM520_01 = s_M5x20_01 / sample
        M520_01 = (eBM520_01 - (20 * 0.1)) ^ 2
        s_M520_01 = s_M520_01 + M520_01

        eBM530_03 = s_M5x30_03 / sample
        M530_03 = (eBM530_03 - (30 * 0.3)) ^ 2
        s_M530_03 = s_M530_03 + M530_03

        eBM540_05 = s_M5x40_05 / sample
        M540_05 = (eBM540_05 - (40 * 0.5)) ^ 2
        s_M540_05 = s_M540_05 + M540_05

        eBM550_06 = s_M5x50_06 / sample
        M550_06 = (eBM550_06 - (50 * 0.6)) ^ 2
        s_M550_06 = s_M550_06 + M550_06

        eBM570_08 = s_M5x70_08 / sample
        M570_08 = (eBM570_08 - (70 * 0.8)) ^ 2
        s_M570_08 = s_M570_08 + M570_08
```

```
End Function

Function mse_binomial1()

        mlem120_01 = (s_M120_01 / loop_test)
        rmlem120_01 = Round(Sqr(mlem120_01), 4)

         mlem130_03 = (s_M130_03 / loop_test)
         rmlem130_03 = Round(Sqr(mlem130_03), 4)

         mlem140_05 = (s_M140_05 / loop_test)
         rmlem140_05 = Round(Sqr(mlem140_05), 4)

         mlem150_06 = (s_M150_06 / loop_test)
         rmlem150_06 = Round(Sqr(mlem150_06), 4)

         mlem170_08 = (s_M170_08 / loop_test)
         rmlem170_08 = Round(Sqr(mlem170_08), 4)


Print "                             "
Print "M1: RMSE BI 20,0.1=" & rmlem120_01, "30,0.3=" & rmlem130_03, "40,0.5="
& rmlem140_05, "50,0.6=" & rmlem150_06, "70,0.8=" & rmlem170_08


End Function


Function mse_binomial2()

        mlem220_01 = (s_M220_01 / loop_test)
        rmlem220_01 = Round(Sqr(mlem220_01), 4)

         mlem230_03 = (s_M230_03 / loop_test)
         rmlem230_03 = Round(Sqr(mlem230_03), 4)

         mlem240_05 = (s_M240_05 / loop_test)
         rmlem240_05 = Round(Sqr(mlem240_05), 4)

         mlem250_06 = (s_M250_06 / loop_test)
         rmlem250_06 = Round(Sqr(mlem250_06), 4)

         mlem270_08 = (s_M270_08 / loop_test)
         rmlem270_08 = Round(Sqr(mlem270_08), 4)


Print "M2: RMSE BI 20,0.1=" & rmlem220_01, "30,0.3=" & rmlem230_03, "40,0.5="
& rmlem240_05, "50,0.6=" & rmlem250_06, "70,0.8=" & rmlem270_08
```

```
End Function

Function mse_binomial3()

        mlem320_01 = (s_M320_01 / loop_test)
        rmlem320_01 = Round(Sqr(mlem320_01), 4)

         mlem330_03 = (s_M330_03 / loop_test)
         rmlem330_03 = Round(Sqr(mlem330_03), 4)

         mlem340_05 = (s_M340_05 / loop_test)
         rmlem340_05 = Round(Sqr(mlem340_05), 4)

         mlem350_06 = (s_M350_06 / loop_test)
         rmlem350_06 = Round(Sqr(mlem350_06), 4)

         mlem370_08 = (s_M370_08 / loop_test)
         rmlem370_08 = Round(Sqr(mlem370_08), 4)


Print "M3: RMSE BI 20,0.1=" & rmlem320_01, "30,0.3=" & rmlem330_03, "40,0.5="
& rmlem340_05, "50,0.6=" & rmlem350_06, "70,0.8=" & rmlem370_08

End Function

Function mse_binomial4()

        mlem420_01 = (s_M420_01 / loop_test)
        rmlem420_01 = Round(Sqr(mlem420_01), 4)

         mlem430_03 = (s_M430_03 / loop_test)
         rmlem430_03 = Round(Sqr(mlem430_03), 4)

         mlem440_05 = (s_M440_05 / loop_test)
         rmlem440_05 = Round(Sqr(mlem440_05), 4)

         mlem450_06 = (s_M450_06 / loop_test)
         rmlem450_06 = Round(Sqr(mlem450_06), 4)

         mlem470_08 = (s_M470_08 / loop_test)
         rmlem470_08 = Round(Sqr(mlem470_08), 4)


Print "M4: RMSE BI 20,0.1=" & rmlem420_01, "30,0.3=" & rmlem430_03, "40,0.5="
& rmlem440_05, "50,0.6=" & rmlem450_06, "70,0.8=" & rmlem470_08

End Function
```

```
Function mse_binomial5()

        mlem520_01 = (s_M520_01 / loop_test)
        rmlem520_01 = Round(Sqr(mlem520_01), 4)

         mlem530_03 = (s_M530_03 / loop_test)
         rmlem530_03 = Round(Sqr(mlem530_03), 4)

         mlem540_05 = (s_M540_05 / loop_test)
         rmlem540_05 = Round(Sqr(mlem540_05), 4)

         mlem550_06 = (s_M550_06 / loop_test)
         rmlem550_06 = Round(Sqr(mlem550_06), 4)

         mlem570_08 = (s_M570_08 / loop_test)
         rmlem570_08 = Round(Sqr(mlem570_08), 4)


Print "M5: RMSE BI 20,0.1=" & rmlem520_01, "30,0.3=" & rmlem530_03, "40,0.5="
& rmlem540_05, "50,0.6=" & rmlem550_06, "70,0.8=" & rmlem570_08


End Function

Function normal1()

            M1y10_1 = -1 * Log(answer)

            If answer <= 1 / 2 Then

                        M1z0_1 = (M1y10_1 - 0) / 1

            Else

                        M1z0_1 = (-M1y10_1 - 0) / 1

            End If
            s_M1z0_1 = s_M1z0_1 + M1z0_1
            normal1_01(i) = M1z0_1


            M1y11_3 = -1 * Log(answer)
            If answer <= 1 / 2 Then

                        M1z1_3 = (M1y11_3 - 1) / 3

            Else

                        M1z1_3 = (-M1y11_3 - 1) / 3
```

```
        End If
        s_M1z1_3 = s_M1z1_3 + M1z1_3
        normal1_13(i) = M1z1_3

        M1y12_5 = -1 * Log(answer)
        If answer <= 1 / 2 Then

                M1z2_5 = (M1y12_5 - 2) / 5

        Else

                M1z2_5 = (-M1y12_5 - 2) / 5

        End If
        s_M1z2_5 = s_M1z2_5 + M1z2_5
        normal1_25(i) = M1z2_5

        M1y13_6 = -1 * Log(answer)
        If answer <= 1 / 2 Then

                M1z3_6 = (M1y13_6 - 3) / 6

        Else

                M1z3_6 = (-M1y13_6 - 3) / 6

        End If
        s_M1z3_6 = s_M1z3_6 + M1z3_6
        normal1_36(i) = M1z3_6

        M1y14_5 = -1 * Log(answer)
        If answer <= 1 / 2 Then

                M1z4_5 = (M1y14_5 - 4) / 5

        Else

                M1z4_5 = (-M1y14_5 - 4) / 5

        End If
        s_M1z4_5 = s_M1z4_5 + M1z4_5
        normal1_45(i) = M1z4_5

End Function

Function normal2()

        M2y10_1 = -1 * Log(answer)
        If answer <= 1 / 2 Then
```

```
        M2z0_1 = (M2y10_1 - 0) / 1

Else

        M2z0_1 = (-M2y10_1 - 0) / 1

End If
s_M2z0_1 = s_M2z0_1 + M2z0_1
normal2_01(i) = M2z0_1


M2y11_3 = -1 * Log(answer)
If answer <= 1 / 2 Then

        M2z1_3 = (M2y11_3 - 1) / 3

Else

        M2z1_3 = (-M2y11_3 - 1) / 3

End If
s_M2z1_3 = s_M2z1_3 + M2z1_3
normal2_13(i) = M2z1_3

M2y12_5 = -1 * Log(answer)
If answer <= 1 / 2 Then

        M2z2_5 = (M2y12_5 - 2) / 5

Else

        M2z2_5 = (-M2y12_5 - 2) / 5

End If
s_M2z2_5 = s_M2z2_5 + M2z2_5
normal2_25(i) = M2z2_5

M2y13_6 = -1 * Log(answer)
If answer <= 1 / 2 Then

        M2z3_6 = (M2y13_6 - 3) / 6

Else

        M2z3_6 = (-M2y13_6 - 3) / 6

End If
s_M2z3_6 = s_M2z3_6 + M2z3_6
normal2_36(i) = M2z3_6
```

```
        M2y14_5 = -1 * Log(answer)
        If answer <= 1 / 2 Then

                M2z4_5 = (M2y14_5 - 4) / 5

        Else

                M2z4_5 = (-M2y14_5 - 4) / 5

        End If
        s_M2z4_5 = s_M2z4_5 + M2z4_5
        normal2_45(i) = M2z4_5

End Function


Function normal3()

        M3y10_1 = -1 * Log(answer)
        If answer <= 1 / 2 Then

                M3z0_1 = (M3y10_1 - 0) / 1

        Else

                M3z0_1 = (-M3y10_1 - 0) / 1

        End If
        s_M3z0_1 = s_M3z0_1 + M3z0_1
        normal3_01(i) = M3z0_1

        M3y11_3 = -1 * Log(answer)
        If answer <= 1 / 2 Then

                M3z1_3 = (M3y11_3 - 1) / 3

        Else

                M3z1_3 = (-M3y11_3 - 1) / 3

        End If
        s_M3z1_3 = s_M3z1_3 + M3z1_3
        normal3_13(i) = M3z1_3

        M3y12_5 = -1 * Log(answer)
        If answer <= 1 / 2 Then

                M3z2_5 = (M3y12_5 - 2) / 5

        Else
```

```
            M3z2_5 = (-M3y12_5 - 2) / 5

        End If
        s_M3z2_5 = s_M3z2_5 + M3z2_5
        normal3_25(i) = M3z2_5

        M3y13_6 = -1 * Log(answer)
        If answer <= 1 / 2 Then

            M3z3_6 = (M3y13_6 - 3) / 6

        Else

            M3z3_6 = (-M3y13_6 - 3) / 6

        End If
        s_M3z3_6 = s_M3z3_6 + M3z3_6
        normal3_36(i) = M3z3_6

        M3y14_5 = -1 * Log(answer)
        If answer <= 1 / 2 Then

            M3z4_5 = (M3y14_5 - 4) / 5

        Else

            M3z4_5 = (-M3y14_5 - 4) / 5

        End If
        s_M3z4_5 = s_M3z4_5 + M3z4_5
        normal3_45(i) = M3z4_5

End Function

Function normal4()

        M4y10_1 = -1 * Log(answer)
        If answer <= 1 / 2 Then

            M4z0_1 = (M4y10_1 - 0) / 1

        Else

            M4z0_1 = (-M4y10_1 - 0) / 1

        End If
        s_M4z0_1 = s_M4z0_1 + M4z0_1
        normal4_01(i) = M4z0_1
```

```
M4y11_3 = -1 * Log(answer)
If answer <= 1 / 2 Then

        M4z1_3 = (M4y11_3 - 1) / 3

Else

        M4z1_3 = (-M4y11_3 - 1) / 3

End If
s_M4z1_3 = s_M4z1_3 + M4z1_3
normal4_13(i) = M4z1_3

M4y12_5 = -1 * Log(answer)
If answer <= 1 / 2 Then

        M4z2_5 = (M4y12_5 - 2) / 5

Else

        M4z2_5 = (-M4y12_5 - 2) / 5

End If
s_M4z2_5 = s_M4z2_5 + M4z2_5
normal4_25(i) = M4z2_5

M4y13_6 = -1 * Log(answer)
If answer <= 1 / 2 Then

        M4z3_6 = (M4y13_6 - 3) / 6

Else

        M4z3_6 = (-M4y13_6 - 3) / 6

End If
s_M4z3_6 = s_M4z3_6 + M4z3_6
normal4_36(i) = M4z3_6

M4y14_5 = -1 * Log(answer)
If answer <= 1 / 2 Then

        M4z4_5 = (M4y14_5 - 4) / 5

Else

        M4z4_5 = (-M4y14_5 - 4) / 5

End If
```

```
        s_M4z4_5 = s_M4z4_5 + M4z4_5
        normal4_45(i) = M4z4_5

End Function


Function normal5()

        M5y10_1 = -1 * Log(answer)
        If answer <= 1 / 2 Then

                M5z0_1 = (M5y10_1 - 0) / 1

        Else

                M5z0_1 = (-M5y10_1 - 0) / 1

        End If
        s_M5z0_1 = s_M5z0_1 + M5z0_1
        normal5_01(i) = M5z0_1


        M5y11_3 = -1 * Log(answer)
        If answer <= 1 / 2 Then

                M5z1_3 = (M5y11_3 - 1) / 3

        Else

                M5z1_3 = (-M5y11_3 - 1) / 3

        End If
        s_M5z1_3 = s_M5z1_3 + M5z1_3
        normal5_13(i) = M5z1_3

        M5y12_5 = -1 * Log(answer)
        If answer <= 1 / 2 Then

                M5z2_5 = (M5y12_5 - 2) / 5

        Else

                M5z2_5 = (-M5y12_5 - 2) / 5

        End If
        s_M5z2_5 = s_M5z2_5 + M5z2_5
        normal5_25(i) = M5z2_5

        M5y13_6 = -1 * Log(answer)
        If answer <= 1 / 2 Then
```

```
                M5z3_6 = (M5y13_6 - 3) / 6

        Else

                M5z3_6 = (-M5y13_6 - 3) / 6

        End If
        s_M5z3_6 = s_M5z3_6 + M5z3_6
        normal5_36(i) = M5z3_6

        M5y14_5 = -1 * Log(answer)
        If answer <= 1 / 2 Then

                M5z4_5 = (M5y14_5 - 4) / 5

        Else

                M5z4_5 = (-M5y14_5 - 4) / 5

        End If
        s_M5z4_5 = s_M5z4_5 + M5z4_5
        normal5_45(i) = M5z4_5

End Function

Function cal_est_normal1()

        eN10_1 = s_M1z0_1 / sample
        m10_1 = (eN10_1 - 0) ^ 2
        s_m10_1 = s_m10_1 + m10_1

        eN11_3 = s_M1z1_3 / sample
        m1_1_3 = (eN11_3 - 1) ^ 2
        s_m1_1_3 = s_m1_1_3 + m1_1_3

        eN12_5 = s_M1z2_5 / sample
        m12_5 = (eN12_5 - 2) ^ 2
        s_m12_5 = s_m12_5 + m12_5

        eN13_6 = s_M1z3_6 / sample
        m13_6 = (eN13_6 - 3) ^ 2
        s_m13_6 = s_m13_6 + m13_6

        eN14_5 = s_M1z4_5 / sample
        m14_5 = (eN14_5 - 4) ^ 2
        s_m14_5 = s_m14_5 + m14_5

End Function
```

```
Function cal_est_normal2()

        eN20_1 = s_M2z0_1 / sample
        m20_1 = (eN20_1 - 0) ^ 2
        s_m20_1 = s_m20_1 + m20_1

        eN21_3 = s_M2z1_3 / sample
        m2_1_3 = (eN21_3 - 1) ^ 2
        s_m2_1_3 = s_m2_1_3 + m2_1_3

        eN22_5 = s_M2z2_5 / sample
        m22_5 = (eN22_5 - 2) ^ 2
        s_m22_5 = s_m22_5 + m22_5

        eN23_6 = s_M2z3_6 / sample
        m23_6 = (eN23_6 - 3) ^ 2
        s_m23_6 = s_m23_6 + m23_6

        eN24_5 = s_M2z4_5 / sample
        m24_5 = (eN24_5 - 4) ^ 2
        s_m24_5 = s_m24_5 + m24_5

End Function

Function cal_est_normal3()

        eN30_1 = s_M3z0_1 / sample
        m30_1 = (eN30_1 - 0) ^ 2
        s_m30_1 = s_m30_1 + m30_1

        eN31_3 = s_M3z1_3 / sample
        m3_1_3 = (eN31_3 - 1) ^ 2
        s_m3_1_3 = s_m3_1_3 + m3_1_3

        eN32_5 = s_M3z2_5 / sample
        m32_5 = (eN32_5 - 2) ^ 2
        s_m32_5 = s_m32_5 + m32_5

        eN33_6 = s_M3z3_6 / sample
        m33_6 = (eN33_6 - 3) ^ 2
        s_m33_6 = s_m33_6 + m33_6

        eN34_5 = s_M3z4_5 / sample
        m34_5 = (eN34_5 - 4) ^ 2
        s_m34_5 = s_m34_5 + m34_5

End Function
```

```
Function cal_est_normal4()

        eN40_1 = s_M4z0_1 / sample
        m40_1 = (eN40_1 - 0) ^ 2
        s_m40_1 = s_m40_1 + m40_1

        eN41_3 = s_M4z1_3 / sample
        m4_1_3 = (eN41_3 - 1) ^ 2
        s_m4_1_3 = s_m4_1_3 + m4_1_3

        eN42_5 = s_M4z2_5 / sample
        m42_5 = (eN42_5 - 2) ^ 2
        s_m42_5 = s_m42_5 + m42_5

        eN43_6 = s_M4z3_6 / sample
        m43_6 = (eN43_6 - 3) ^ 2
        s_m43_6 = s_m43_6 + m43_6

        eN44_5 = s_M4z4_5 / sample
        m44_5 = (eN44_5 - 4) ^ 2
        s_m44_5 = s_m44_5 + m44_5

End Function


Function cal_est_normal5()

        eN50_1 = s_M5z0_1 / sample
        m50_1 = (eN50_1 - 0) ^ 2
        s_m50_1 = s_m50_1 + m50_1

        eN51_3 = s_M5z1_3 / sample
        m5_1_3 = (eN51_3 - 1) ^ 2
        s_m5_1_3 = s_m5_1_3 + m5_1_3

        eN52_5 = s_M5z2_5 / sample
        m52_5 = (eN52_5 - 2) ^ 2
        s_m52_5 = s_m52_5 + m52_5

        eN53_6 = s_M5z3_6 / sample
        m53_6 = (eN53_6 - 3) ^ 2
        s_m53_6 = s_m53_6 + m53_6

        eN54_5 = s_M5z4_5 / sample
        m54_5 = (eN54_5 - 4) ^ 2
        s_m54_5 = s_m54_5 + m54_5

End Function
```

```
Function mse_normal1()

    mlem10_1 = (s_m10_1 / loop_test)
    rmlem10_1 = Round(Sqr(mlem10_1), 4)

    mlem1_1_3 = (s_m1_1_3 / loop_test)
    rmlem1_1_3 = Round(Sqr(mlem1_1_3), 4)

    mlem12_5 = (s_m12_5 / loop_test)
    rmlem12_5 = Round(Sqr(mlem12_5), 4)

    mlem13_6 = (s_m13_6 / loop_test)
    rmlem13_6 = Round(Sqr(mlem13_6), 4)

    mlem14_5 = (s_m14_5 / loop_test)
    rmlem14_5 = Round(Sqr(mlem14_5), 4)


Print "                              "
Print "M1: RMSE NM 0,1=" & rmlem10_1, "1,3=" & rmlem1_1_3, "2,5=" &
rmlem12_5, "3,6=" & rmlem13_6, "4,5=" & rmlem14_5


End Function


Function mse_normal2()

    mlem20_1 = (s_m20_1 / loop_test)
    rmlem20_1 = Round(Sqr(mlem20_1), 4)

    mlem2_1_3 = (s_m2_1_3 / loop_test)
    rmlem2_1_3 = Round(Sqr(mlem2_1_3), 4)

    mlem22_5 = (s_m22_5 / loop_test)
    rmlem22_5 = Round(Sqr(mlem22_5), 4)

    mlem23_6 = (s_m23_6 / loop_test)
    rmlem23_6 = Round(Sqr(mlem23_6), 4)

    mlem24_5 = (s_m24_5 / loop_test)
    rmlem24_5 = Round(Sqr(mlem24_5), 4)


Print "M2: RMSE NM 0,1=" & rmlem20_1, "1,3=" & rmlem2_1_3, "2,5=" &
rmlem22_5, "3,6=" & rmlem23_6, "4,5=" & rmlem24_5


End Function
```

```
Function mse_normal3()

        mlem30_1 = (s_m30_1 / loop_test)
        rmlem30_1 = Round(Sqr(mlem30_1), 4)

        mlem3_1_3 = (s_m3_1_3 / loop_test)
        rmlem3_1_3 = Round(Sqr(mlem3_1_3), 4)

        mlem32_5 = (s_m32_5 / loop_test)
        rmlem32_5 = Round(Sqr(mlem32_5), 4)

        mlem33_6 = (s_m33_6 / loop_test)
        rmlem33_6 = Round(Sqr(mlem33_6), 4)

        mlem34_5 = (s_m34_5 / loop_test)
        rmlem34_5 = Round(Sqr(mlem34_5), 4)


Print "M3: RMSE NM 0,1=" & rmlem30_1, "1,3=" & rmlem3_1_3, "2,5=" &
rmlem32_5, "3,6=" & rmlem33_6, "4,5=" & rmlem34_5


End Function


Function mse_normal4()

        mlem40_1 = (s_m40_1 / loop_test)
        rmlem40_1 = Round(Sqr(mlem40_1), 4)

        mlem4_1_3 = (s_m4_1_3 / loop_test)
        rmlem4_1_3 = Round(Sqr(mlem4_1_3), 4)

        mlem42_5 = (s_m42_5 / loop_test)
        rmlem42_5 = Round(Sqr(mlem42_5), 4)

        mlem43_6 = (s_m43_6 / loop_test)
        rmlem43_6 = Round(Sqr(mlem43_6), 4)

        mlem44_5 = (s_m44_5 / loop_test)
        rmlem44_5 = Round(Sqr(mlem44_5), 4)


Print "M4: RMSE NM 0,1=" & rmlem40_1, "1,3=" & rmlem4_1_3, "2,5=" &
rmlem42_5, "3,6=" & rmlem43_6, "4,5=" & rmlem44_5


End Function
```

```
Function mse_normal5()

    mlem50_1 = (s_m50_1 / loop_test)
    rmlem50_1 = Round(Sqr(mlem50_1), 4)

    mlem5_1_3 = (s_m5_1_3 / loop_test)
    rmlem5_1_3 = Round(Sqr(mlem5_1_3), 4)

    mlem52_5 = (s_m52_5 / loop_test)
    rmlem52_5 = Round(Sqr(mlem52_5), 4)

    mlem53_6 = (s_m53_6 / loop_test)
    rmlem53_6 = Round(Sqr(mlem53_6), 4)

    mlem54_5 = (s_m54_5 / loop_test)
    rmlem54_5 = Round(Sqr(mlem54_5), 4)


Print "M5: RMSE NM 0,1=" & rmlem50_1, "1,3=" & rmlem5_1_3, "2,5=" &
rmlem52_5, "3,6=" & rmlem53_6, "4,5=" & rmlem54_5


End Function

Function est_std()

      For i = 1 To sample

            v10_1 = (normal1_01(i) - eN10_1) ^ 2
            s_v10_1 = s_v10_1 + v10_1

            v11_3 = (normal1_13(i) - eN11_3) ^ 2
            s_v11_3 = s_v11_3 + v11_3

            v12_5 = (normal1_25(i) - eN12_5) ^ 2
            s_v12_5 = s_v12_5 + v12_5

            v13_6 = (normal1_36(i) - eN13_6) ^ 2
            s_v13_6 = s_v13_6 + v13_6

            v14_5 = (normal1_45(i) - eN14_5) ^ 2
            s_v14_5 = s_v14_5 + v14_5


            v20_1 = (normal2_01(i) - eN20_1) ^ 2
            s_v20_1 = s_v20_1 + v20_1

            v21_3 = (normal2_13(i) - eN21_3) ^ 2
            s_v21_3 = s_v21_3 + v21_3
```

```
v22_5 = (normal2_25(i) - eN22_5) ^ 2
s_v22_5 = s_v22_5 + v22_5

v23_6 = (normal2_36(i) - eN23_6) ^ 2
s_v23_6 = s_v23_6 + v23_6

v24_5 = (normal2_45(i) - eN24_5) ^ 2
s_v24_5 = s_v24_5 + v24_5


v30_1 = (normal3_01(i) - eN30_1) ^ 2
s_v30_1 = s_v30_1 + v30_1

v31_3 = (normal3_13(i) - eN31_3) ^ 2
s_v31_3 = s_v31_3 + v31_3

v32_5 = (normal3_25(i) - eN32_5) ^ 2
s_v32_5 = s_v32_5 + v32_5

v33_6 = (normal3_36(i) - eN33_6) ^ 2
s_v33_6 = s_v33_6 + v33_6

v34_5 = (normal3_45(i) - eN34_5) ^ 2
s_v34_5 = s_v34_5 + v34_5


v40_1 = (normal4_01(i) - eN40_1) ^ 2
s_v40_1 = s_v40_1 + v40_1

v41_3 = (normal4_13(i) - eN41_3) ^ 2
s_v41_3 = s_v41_3 + v41_3

v42_5 = (normal4_25(i) - eN42_5) ^ 2
s_v42_5 = s_v42_5 + v42_5

v43_6 = (normal4_36(i) - eN43_6) ^ 2
s_v43_6 = s_v43_6 + v43_6

v44_5 = (normal4_45(i) - eN44_5) ^ 2
s_v44_5 = s_v44_5 + v44_5


v50_1 = (normal5_01(i) - eN50_1) ^ 2
s_v50_1 = s_v50_1 + v50_1

v51_3 = (normal5_13(i) - eN51_3) ^ 2
s_v51_3 = s_v51_3 + v51_3

v52_5 = (normal5_25(i) - eN52_5) ^ 2
```

```
        s_v52_5 = s_v52_5 + v52_5

        v53_6 = (normal5_36(i) - eN53_6) ^ 2
        s_v53_6 = s_v53_6 + v53_6

        v54_5 = (normal5_45(i) - eN54_5) ^ 2
        s_v54_5 = s_v54_5 + v54_5

    Next i

End Function


Function cal_variance()

        var1_01 = s_v10_1 / (sample - 1)
        var1_13 = s_v11_3 / (sample - 1)
        var1_25 = s_v12_5 / (sample - 1)
        var1_36 = s_v13_6 / (sample - 1)
        var1_45 = s_v14_5 / (sample - 1)

        var2_01 = s_v20_1 / (sample - 1)
        var2_13 = s_v21_3 / (sample - 1)
        var2_25 = s_v22_5 / (sample - 1)
        var2_36 = s_v23_6 / (sample - 1)
        var2_45 = s_v24_5 / (sample - 1)

        var3_01 = s_v30_1 / (sample - 1)
        var3_13 = s_v31_3 / (sample - 1)
        var3_25 = s_v32_5 / (sample - 1)
        var3_36 = s_v33_6 / (sample - 1)
        var3_45 = s_v34_5 / (sample - 1)

        var4_01 = s_v40_1 / (sample - 1)
        var4_13 = s_v41_3 / (sample - 1)
        var4_25 = s_v42_5 / (sample - 1)
        var4_36 = s_v43_6 / (sample - 1)
        var4_45 = s_v44_5 / (sample - 1)

        var5_01 = s_v50_1 / (sample - 1)
        var5_13 = s_v51_3 / (sample - 1)
        var5_25 = s_v52_5 / (sample - 1)
        var5_36 = s_v53_6 / (sample - 1)
        var5_45 = s_v54_5 / (sample - 1)

End Function
```

```
Function mse_variance()

        mle_v10_1 = (var1_01 - 1) ^ 2
        rmle_v10_1 = Round(Sqr(mle_v10_1), 4)

        mle_v11_3 = (var1_13 - 3) ^ 2
        rmle_v11_3 = Round(Sqr(mle_v11_3), 4)

        mle_v12_5 = (var1_25 - 5) ^ 2
        rmle_v12_5 = Round(Sqr(mle_v12_5), 4)

        mle_v13_6 = (var1_36 - 6) ^ 2
        rmle_v13_6 = Round(Sqr(mle_v13_6), 4)

        mle_v14_5 = (var1_45 - 5) ^ 2
        rmle_v14_5 = Round(Sqr(mle_v14_5), 4)

        mle_v20_1 = (var2_01 - 1) ^ 2
        rmle_v20_1 = Round(Sqr(mle_v20_1), 4)

        mle_v21_3 = (var2_13 - 3) ^ 2
        rmle_v21_3 = Round(Sqr(mle_v21_3), 4)

        mle_v22_5 = (var2_25 - 5) ^ 2
        rmle_v22_5 = Round(Sqr(mle_v22_5), 4)

        mle_v23_6 = (var2_36 - 6) ^ 2
        rmle_v23_6 = Round(Sqr(mle_v23_6), 4)

        mle_v24_5 = (var2_45 - 5) ^ 2
        rmle_v24_5 = Round(Sqr(mle_v24_5), 4)

        mle_v30_1 = (var3_01 - 1) ^ 2
        rmle_v30_1 = Round(Sqr(mle_v30_1), 4)

        mle_v31_3 = (var3_13 - 3) ^ 2
        rmle_v31_3 = Round(Sqr(mle_v31_3), 4)

        mle_v32_5 = (var3_25 - 5) ^ 2
        rmle_v32_5 = Round(Sqr(mle_v32_5), 4)

        mle_v33_6 = (var3_36 - 6) ^ 2
        rmle_v33_6 = Round(Sqr(mle_v33_6), 4)

        mle_v34_5 = (var3_45 - 5) ^ 2
        rmle_v34_5 = Round(Sqr(mle_v34_5), 4)


        mle_v40_1 = (var4_01 - 1) ^ 2
        rmle_v40_1 = Round(Sqr(mle_v40_1), 4)
```

```
        mle_v41_3 = (var4_13 - 3) ^ 2
        rmle_v41_3 = Round(Sqr(mle_v41_3), 4)

        mle_v42_5 = (var4_25 - 5) ^ 2
        rmle_v42_5 = Round(Sqr(mle_v42_5), 4)

        mle_v43_6 = (var4_36 - 6) ^ 2
        rmle_v43_6 = Round(Sqr(mle_v43_6), 4)

        mle_v44_5 = (var4_45 - 5) ^ 2
        rmle_v44_5 = Round(Sqr(mle_v44_5), 4)

        mle_v50_1 = (var5_01 - 1) ^ 2
        rmle_v50_1 = Round(Sqr(mle_v50_1), 4)

        mle_v51_3 = (var5_13 - 3) ^ 2
        rmle_v51_3 = Round(Sqr(mle_v51_3), 4)

        mle_v52_5 = (var5_25 - 5) ^ 2
        rmle_v52_5 = Round(Sqr(mle_v52_5), 4)

        mle_v53_6 = (var5_36 - 6) ^ 2
        rmle_v53_6 = Round(Sqr(mle_v53_6), 4)

        mle_v54_5 = (var5_45 - 5) ^ 2
        rmle_v54_5 = Round(Sqr(mle_v54_5), 4)

Print "                    "
Print "M1: RMSE VAR 0,1=" & rmle_v10_1, "1,3=" & rmle_v11_3, "2,5=" &
rmle_v12_5
Print "3,6=" & rmle_v13_6, "4,5=" & rmle_v14_5
Print "                    "
Print "M2: RMSE VAR 0,1=" & rmle_v20_1, "1,3=" & rmle_v21_3, "2,5=" &
rmle_v22_5
Print "3,6=" & rmle_v23_6, "4,5=" & rmle_v24_5
Print "                    "
Print "M3: RMSE VAR 0,1=" & rmle_v30_1, "1,3=" & rmle_v31_3, "2,5=" &
rmle_v32_5
Print "3,6=" & rmle_v33_6, "4,5=" & rmle_v34_5
Print "                    "
Print "M4: RMSE VAR 0,1=" & rmle_v40_1, "1,3=" & rmle_v41_3, "2,5=" &
rmle_v42_5
Print "3,6=" & rmle_v43_6, "4,5=" & rmle_v44_5
Print "                    "
Print "M5: RMSE VAR 0,1=" & rmle_v50_1, "1,3=" & rmle_v51_3, "2,5=" &
rmle_v52_5
Print "3,6=" & rmle_v53_6, "4,5=" & rmle_v54_5

End Function
```

ประวัติผู้วิจัย

นางสาว นภาภรณ์ จันทรศัพท์

วุฒิการศึกษา

วท.บ.(สถิติประยุกต์) สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
พบ.ม.(วิทยาการประกันภัย) สถาบันบัณฑิตพัฒนบริหารศาสตร์

ประสบการณ์การทำงาน

อาจารย์ประจำมหาวิทยาลัยธุรกิจบัณฑิตย์