Chapter 6

Modeling Random Data & Noise

6.1 Problems

Problem 6.1 (Sample mean, variance, and SD) *Let the RV X produce the following $n_x = 10$ realizations:*

$$1.7, -0.8, -0.6, 0.5, 0.3, -0.5, 0.8, -0.1, 0.9, -0.7$$

Compute the sample mean, variance, and SD.

(ans:
Sample mean:

$$Ave(X_i) = \frac{1.7 - 0.8 - 0.6 + 0.5 + 0.3 - 0.5 + 0.8 - 0.1 + 0.9 - 0.7}{10} = 0.15$$

Sample variance: $Var(X_i) = Ave(X_i^2) - [Ave(X_i)]^2$

$$Ave(X_i^2) = \frac{2.89 + 0.64 + 0.36 + 0.25 + 0.9 + 0.25 + 0.64 + 0.01 + 0.81 + 0.49}{10} = 0.724$$

$$Var(X_i) = 0.724 - 0.0225 = 0.7015$$

Sample SD:

$$SD(X_i) = \sqrt{Var(X_i)} = 0.838$$

)

Problem 6.2 (Probability that Y lies in [0.25,0.75)) *What is the probability that the RV Y that is uniformly distributed over [0,1), will lie in the range [0.25, 0.75)?*

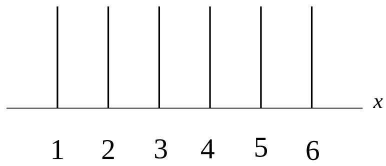
(ans: The area from $x = 0.25$ to $x = 0.75$ = 0.5.
)

Problem 6.3 (Probability that Y lies in [0.25,2)) *What is the probability that the RV Y that is uniformly distributed over [0,1), will lie in the range [0.25,2)?*

(ans: The area from $x = 0.25$ to $x = 2$ under the PDF that extends over [0,1] is the same as the area from $x = 0.25$ to $x = 1$ = 0.75 .
)

Problem 6.4 (Simulating random die toss) *Starting with realizations of Y, how would you form T, the RV that simulates the result of a fair die toss, which equals the number of dots showing on the top face? Sketch the PMF of T.*

(ans:
Y extends over [0,1).
$6Y$ extends over [0,6).
$T =$ Ceil$(6Y)$ is the integer set [1,6].
The PMF of T:



)

Problem 6.5 (Adding 1 to Y) *Starting with RV Y, we add 1 to each value we observe to form*

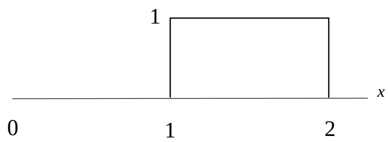
$$X = 1 + Y$$

Compute μ_X, σ_X^2 and sketch $p_X(x)$.

(ans:

$$\mu_X = \mu_{Y+1} = \overbrace{\mu_Y}^{=0.5} + 1 = 1.5$$
$$\sigma_X^2 = \sigma_{Y+1}^2 = \sigma_Y^2 = \frac{1}{12}$$

$p_X(x)$:



)

Problem 6.6 (Adding 1 to G) *Starting with standardized RV G, add 1 to each value to form*

$$X = 1 + G$$

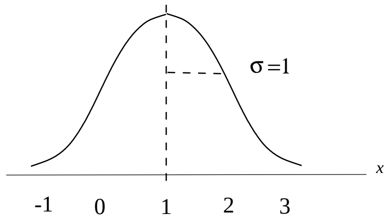
Compute μ_X, σ_X^2 and sketch $p_X(x)$.

(ans:

$$\mu_X = \mu_{G+1} = \overbrace{\mu_G}^{=0} + 1 = 1$$

$$\sigma_X^2 = \sigma_{G+1}^2 = \sigma_G^2 = 1$$

$p_X(x)$:



)

Problem 6.7 (Multiplying Y by 2) *Starting with RV Y that is uniformly distributed over [0,1), multiply each value to form*

$$Z = 2Y$$

Compute μ_Z, σ_Z^2 and sketch $p_Z(x)$.

(ans:

$$\mu_Z = \mu_{2Y} = 2\overbrace{\mu_Y}^{=0.5} = 1$$

$$\sigma_Z^2 = \sigma_{2Y}^2 = 4\sigma_Y^2 = \frac{4}{12} = \frac{1}{3}$$

$p_Z(x)$:

)

Problem 6.8 (Multiplying G by 2) *Starting with standardized Gaussian RV G, multiply each value by 2 to form*

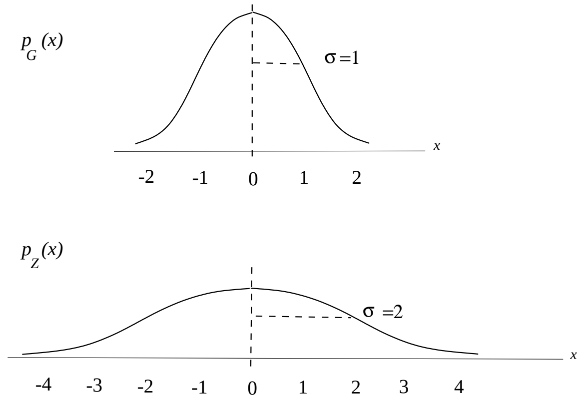
$$Z = 2G$$

Compute μ_Z, σ_Z^2 and sketch $p_Z(x)$.

(ans:

$$\mu_Z = \mu_{2G} = 2\overbrace{\mu_G}^{=0} = 0$$

$$\sigma_Z^2 = \sigma_{2G}^2 = 4\sigma_G^2 = 4$$



)

Problem 6.9 (Summing two independent standardized Gaussian RVs) *Starting with independent standardized Gaussian RVs G_1 and G_2, form*

$$V = 2(G_1 + 1) - (G_2 - 1)$$

Compute μ_V, σ_V^2 and $p_V(x)$.

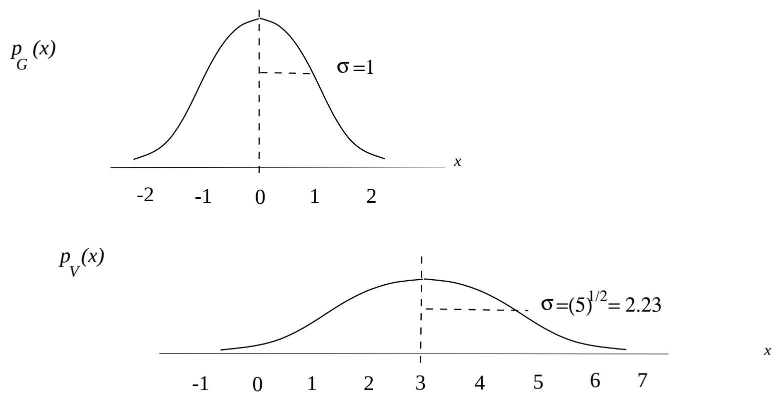
(ans:

$$V = 2(G_1 + 1) - (G_2 - 1) = 2G_1 + 2 - G_2 + 1 = 2G_1 - G_2 + 3$$

$$\mu_V = \mu_{2G_1 - G_2 + 3} = \mu_V = 2\overbrace{\mu_{G_1}}^{=0} - \overbrace{\mu_{G_2}}^{=0} + 3 = 3$$

$$\sigma_V^2 = \sigma_{2G_1 - G_2 + 3}^2 = \sigma_{2G_1 - G_2}^2 = 4\overbrace{\sigma_{G_1}^2}^{=1} + (-1)^2\overbrace{\sigma_{G_2}^2}^{=1} = 5$$

When adding Gaussian RV, the resulting PDF remains Gaussian and is completely specified by the mean and standard deviation.



)

6.2 Excel Projects

Project 6.1 (Random numbers using RAND()) *Extend Example 13.28 to generate* $n_Y = 25$ *random numbers* Y_i *for* $0 \leq i \leq 24$*, and compute the sample averages. Observe how the sample averages change as a new set of random numbers is generated.*

(ans:

	A	B	C	D
1	i	Y_i		
2	0	0.683		
3	1	0.875		Sample Ave=
4	2	0.279		0.487
5	3	0.380		
6	4	0.606		Sample SD=
7	5	0.347		0.306
8	6	0.716		
9	7	0.072		Sample Var=
10	8	0.397		0.094
11	9	0.876		
12	10	0.216		
13	11	0.785		
14	12	0.985		
15	13	0.470		
16	14	0.360		
17	15	0.245		
18	16	0.029		
19	17	0.772		
20	18	0.014		
21	19	0.232		
22	20	0.833		
23	21	0.593		
24	22	0.099		
25	23	0.955		
26	24	0.357		

	A	B	C	D
1	i	Y_i		
2	0	=RAND()		
3	=A2+1	=RAND()		Sample Ave=
4	=A3+1	=RAND()		=AVERAGE(B2:B26)
5	=A4+1	=RAND()		
6	=A5+1	=RAND()		Sample SD=
7	=A6+1	=RAND()		=STDEV.S(B2:B26)
8	=A7+1	=RAND()		
9	=A8+1	=RAND()		Sample Var=
10	=A9+1	=RAND()		=VAR(B2:B26)

)

Project 6.2 (Two Random Dice) *Extend Example 13.30 to simulate the toss of two dice, $D1_i$ and $D2_i$ for $0 \leq i \leq 24$, and add the observed values together. Observe how the sums change as another set of random numbers is generated.*

(ans:

	A	B	C	D	E	F
1	i	$Y1_i$	$Y2_i$	$D1_i$	$D2_i$	Sum
2	0	0.424	0.787	3	5	8
3	1	0.809	0.045	5	1	6
4	2	0.324	0.692	2	5	7
5	3	0.801	0.422	5	3	8
6	4	0.011	0.703	1	5	6
7	5	0.349	0.376	3	3	6
8	6	0.073	0.294	1	2	3
9	7	0.965	0.536	6	4	10
10	8	0.573	0.028	4	1	5
11	9	0.606	0.720	4	5	9
12	10	0.812	0.900	5	6	11
13	11	0.719	0.280	5	2	7
14	12	0.554	0.141	4	1	5
15	13	0.210	0.955	2	6	8
16	14	0.655	0.293	4	2	6
17	15	0.121	0.438	1	3	4
18	16	0.791	0.885	5	6	11
19	17	0.701	0.089	5	1	6
20	18	0.865	0.534	6	4	10
21	19	0.593	0.245	4	2	6
22	20	0.757	0.224	5	2	7
23	21	0.484	0.564	3	4	7
24	22	0.796	0.686	5	5	10
25	23	0.083	0.004	1	1	2
26	24	0.216	0.561	2	4	6

	A	B	C	D	E	F
1	i	$Y1_i$	$Y2_i$	$D1_i$	$D2_i$	Sum
2	0	=RAND()	=RAND()	=CEILING(6*B2,1)	=CEILING(6*C2,1)	=D2+E2
3	1	=RAND()	=RAND()	=CEILING(6*B3,1)	=CEILING(6*C3,1)	=D3+E3
4	2	=RAND()	=RAND()	=CEILING(6*B4,1)	=CEILING(6*C4,1)	=D4+E4
5	3	=RAND()	=RAND()	=CEILING(6*B5,1)	=CEILING(6*C5,1)	=D5+E5
6	4	=RAND()	=RAND()	=CEILING(6*B6,1)	=CEILING(6*C6,1)	=D6+E6
7	5	=RAND()	=RAND()	=CEILING(6*B7,1)	=CEILING(6*C7,1)	=D7+E7
8	6	=RAND()	=RAND()	=CEILING(6*B8,1)	=CEILING(6*C8,1)	=D8+E8
9	7	=RAND()	=RAND()	=CEILING(6*B9,1)	=CEILING(6*C9,1)	=D9+E9
10	8	=RAND()	=RAND()	=CEILING(6*B10,1)	=CEILING(6*C10,1)	=D10+E10

)

Project 6.3 (Histogram of uniformly-distributed random numbers) *Using Example 13.32 as a guide generate a histogram of ten thousand random numbers produced by RAND().*

(ans: The problem involves setting up the proper bin intervals in `C` *and computing the bin number, corresponding to the row number in* `D`*, from the random number generated in* `A2`*. The example shows dividing the interval [0,1) into 10 bins, and including the adjacent bins to illustrate their zero counts. Note that* `FLOOR` *must be used, as* `ROUND` *produces half-counts in the two limit bins.*

	A	B	C	D
1	Y_i=		bin	count
2	0.53448142		-0.1	0
3			0	929
4	i=		0.1	1058
5	10000		0.2	1025
6			0.3	972
7	n_x=		0.4	944
8	10000		0.5	1028
9			0.6	1011
10	bin#=		0.7	963
11	8		0.8	1032
12			0.9	1038
13		Hist	1	0
14			1.1	0

	A	B	C	D
1	Y_i=		bin	count
2	= RAND()		-0.1	0
3			=C2+0.1	1000
4	i=		=C3+0.1	1006
5	10000		=C4+0.1	1000
6			=C5+0.1	1010
7	n_x=		=C6+0.1	952
8	10000		=C7+0.1	989
9			=C8+0.1	975
10	bin#=		=C9+0.1	1009
11	=3+FLOOR(10*A2,1)		=C10+0.1	1046
12			=C11+0.1	1013
13		Hist	=C12+0.1	0
14			=C13+0.1	0

```
Sub hist()
Dim Val As Integer           ' define an integer variable
Range("D2:D14").Value = 0    ' reset bin counts
Range("A5").Value = 0        ' reset counter i
Do While Range("A5").Value < Range("A8").Value  ' loop for nX times
    Val = Range("A11").Value                    ' get bin value
    If Val > 1 Then                             '  if valid bin value
        Range("D" & Val).Value = Range("D" & Val).Value + 1    ' incr bin count
    End If
    Range("A5").Value = Range("A5").Value + 1 ' increment counter i
Loop
End Sub
```

)

Project 6.4 (Random Gaussian noise) *Extend Example 13.33 to generate $n_N = 25$ random numbers N_i for $0 \le i \le 24$ and compute the sample averages. Combine columns B and C to form N_i with a single formula using the PRNG. Observe how the sample averages change as another set of random numbers is generated.*

(ans:

	A	B	C	D
1	σ_N=	4		
2				
3	i	N_i		
4	0	-4.401		
5	1	-0.931		Sample Ave=
6	2	5.052		0.509
7	3	4.673		
8	4	3.157		Sample SD=
9	5	-4.978		3.959
10	6	-1.091		
11	7	4.841		Sample Var=
12	8	4.215		15.673
13	9	3.565		
14	10	2.109		
15	11	-1.378		
16	12	-2.788		
17	13	5.537		
18	14	4.461		
19	15	-0.966		
20	16	4.473		
21	17	1.705		
22	18	-5.163		
23	19	-1.696		
24	20	1.914		
25	21	0.483		
26	22	-8.931		
27	23	-3.951		
28	24	2.801		

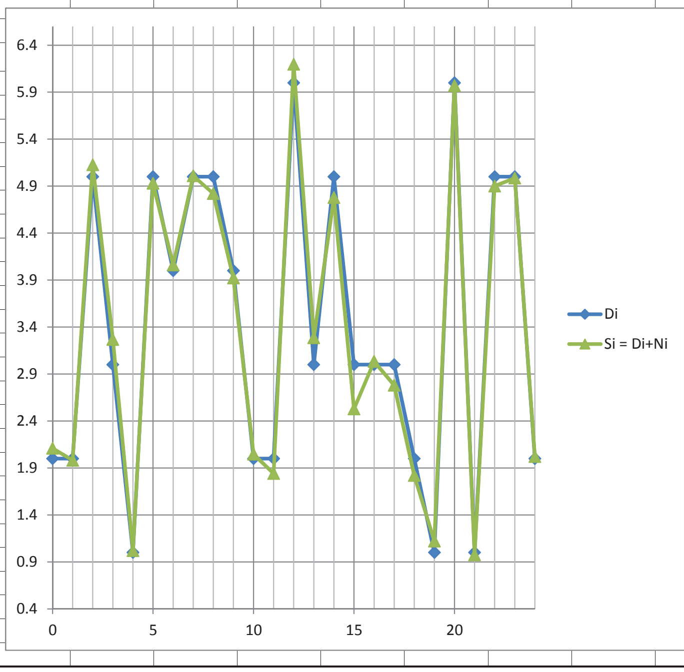
	A	B	C	D
1	σ_N=	4		
2				
3	i	N_i		
4	0	=$B$1*NORM.S.INV(RAND())		
5	=A4+1	=$B$1*NORM.S.INV(RAND())		Sample Ave=
6	=A5+1	=$B$1*NORM.S.INV(RAND())		=AVERAGE(B4:B28)
7	=A6+1	=$B$1*NORM.S.INV(RAND())		
8	=A7+1	=$B$1*NORM.S.INV(RAND())		Sample SD=
9	=A8+1	=$B$1*NORM.S.INV(RAND())		=STDEV.S(B4:B28)
10	=A9+1	=$B$1*NORM.S.INV(RAND())		
11	=A10+1	=$B$1*NORM.S.INV(RAND())		Sample Var=
12	=A11+1	=$B$1*NORM.S.INV(RAND())		=VAR(B4:B28)

)

Project 6.5 (Noisy digital die-toss signals) *Using Example 13.34 as a guide, compose a worksheet that generates 25 samples of random data corresponding to a die toss that is corrupted by additive random Gaussian noise with* $\sigma_N = 0.2$

(ans: The issues here involve formatting the amplitude in the plot. Because almost all Gaussian values fall within [-3σ, 3σ] of the mean, for $\sigma = 0.2$ the y scale range should be [(1-0.6)=0.4, (6+0.6)=6.6].

	A	B	C	D
1	σ_N=	0.2		
2				
3	i	D_i	N_i	$S_i = D_i+N_i$
4	0	2	0.109	2.109
5	1	2	-0.018	1.982
6	2	5	0.126	5.126
7	3	3	0.268	3.268
8	4	1	0.021	1.021
9	5	5	-0.070	4.930
10	6	4	0.061	4.061
11	7	5	0.010	5.010
12	8	5	-0.179	4.821
13	9	4	-0.078	3.922
14	10	2	0.047	2.047
15	11	2	-0.158	1.842
16	12	6	0.197	6.197
17	13	3	0.287	3.287
18	14	5	-0.223	4.777
19	15	3	-0.472	2.528
20	16	3	0.034	3.034
21	17	3	-0.220	2.780
22	18	2	-0.180	1.820
23	19	1	0.122	1.122
24	20	6	-0.031	5.969
25	21	1	-0.026	0.974
26	22	5	-0.099	4.901
27	23	5	-0.014	4.986
28	24	2	0.022	2.022



	A	B	C	D
3	i	D_i	N_i	$S_i = D_i+N_i$
4	0	=CEILING(6*RAND(),1)	=$B$1*NORM.S.INV(RAND())	=B4+C4
5	=A4+1	=CEILING(6*RAND(),1)	=$B$1*NORM.S.INV(RAND())	=B5+C5
6	=A5+1	=CEILING(6*RAND(),1)	=$B$1*NORM.S.INV(RAND())	=B6+C6
7	=A6+1	=CEILING(6*RAND(),1)	=$B$1*NORM.S.INV(RAND())	=B7+C7
8	=A7+1	=CEILING(6*RAND(),1)	=$B$1*NORM.S.INV(RAND())	=B8+C8
9	=A8+1	=CEILING(6*RAND(),1)	=$B$1*NORM.S.INV(RAND())	=B9+C9
10	=A9+1	=CEILING(6*RAND(),1)	=$B$1*NORM.S.INV(RAND())	=B10+C10
11	=A10+1	=CEILING(6*RAND(),1)	=$B$1*NORM.S.INV(RAND())	=B11+C11
12	=A11+1	=CEILING(6*RAND(),1)	=$B$1*NORM.S.INV(RAND())	=B12+C12

)

Project 6.6 (Thresholding noisy binary signals) *Extend Example 13.35 to form 20 samples of random binary data that is corrupted by additive random Gaussian noise with specified σ_N. Attempt to determine the data from S_i by applying a threshold $\tau = 0.5$ to estimate the data value. Repeatedly generate new data. Determine the σ_N value that typically produces 1 error is the 20 values of S_i.*

(ans:

	A	B	C	D	E	F
1	σ_N=	0.3				
2						
3	i	B_i	N_i	$S_i = B_i+N_i$	$\sim B_i$	Error
4	0	1	-0.316	0.684	1	0
5	1	1	-0.489	0.511	1	0
6	2	1	-0.429	0.571	1	0
7	3	1	0.028	1.028	1	0
8	4	0	0.212	0.212	0	0
9	5	0	0.122	0.122	0	0
10	6	0	0.416	0.416	0	0
11	7	1	-0.184	0.816	1	0
12	8	0	0.394	0.394	0	0
13	9	0	-0.160	-0.160	0	0
14	10	1	-0.239	0.761	1	0
15	11	1	-0.022	0.978	1	0
16	12	1	-0.890	0.110	0	1
17	13	1	-0.180	0.820	1	0
18	14	1	0.548	1.548	1	0
19	15	1	0.484	1.484	1	0
20	16	1	-0.090	0.910	1	0
21	17	1	-0.189	0.811	1	0
22	18	0	0.088	0.088	0	0
23	19	1	0.148	1.148	1	0

	A	B	C	D	E	F
3	i	B_i	N_i	$S_i = B_i+N_i$	$\sim B_i$	Error
4	0	=IF(RAND()<0.5,0,1)	= $B$1*NORM.S.INV(RAND())	=B4+C4	=IF(D4<0.5,0,1)	=IF(B4<>E4,1,0)
5	=A4+1	=IF(RAND()<0.5,0,1)	= $B$1*NORM.S.INV(RAND())	=B5+C5	=IF(D5<0.5,0,1)	=IF(B5<>E5,1,0)
6	=A5+1	=IF(RAND()<0.5,0,1)	= $B$1*NORM.S.INV(RAND())	=B6+C6	=IF(D6<0.5,0,1)	=IF(B6<>E6,1,0)
7	=A6+1	=IF(RAND()<0.5,0,1)	= $B$1*NORM.S.INV(RAND())	=B7+C7	=IF(D7<0.5,0,1)	=IF(B7<>E7,1,0)
8	=A7+1	=IF(RAND()<0.5,0,1)	= $B$1*NORM.S.INV(RAND())	=B8+C8	=IF(D8<0.5,0,1)	=IF(B8<>E8,1,0)
9	=A8+1	=IF(RAND()<0.5,0,1)	= $B$1*NORM.S.INV(RAND())	=B9+C9	=IF(D9<0.5,0,1)	=IF(B9<>E9,1,0)
10	=A9+1	=IF(RAND()<0.5,0,1)	= $B$1*NORM.S.INV(RAND())	=B10+C10	=IF(D10<0.5,0,1)	=IF(B10<>E10,1,0)

)

Project 6.7 (Histogram of noisy binary signals) *Using Example 13.32 as a guide generate a histogram of ten thousand noisy binary signals.*

(ans: The random binary data takes on values of +1 and -1 computed in `B1`*. The corresponding signal is computed in* `D2`*. The noise standard deviation is specified in* `B3`*. The Sd is convenient for setting the histogram limits. The Gaussian noise sample is computed in* `D3`*, and the signal plus noise is formed in* `D4`*.*

The histogram extends from hist min *to* hist max *(=* $-$ hist min*) and its parameters are computed in the worksheet. The* hist min *value in* `A9` *is computed from* A *and* σ_N *as* hist min $= -A - 4\sigma_N$*)) to effectively guarantee that it will be less than the minimum* X_i *value encountered. The number of bins is specified by the user in* `B9`*. The bin width is computed in* `C9` *as* $-2($ *hist min)/(# bins-1). The bin number of an observed value is the row number is the histogram display and is computed as rounded((*X_i *value minus* hist min*)/bin width) +2 in* `C6` *as*

```
=ROUND((D4-A9)/C9,0)+2
```

This allows the verification of the bin number for each manual observation (produce by `F9`*).*

	A	B	C	D	E	F	G
1	D=	-1				bin	count
2	A=	10	S_i=	-10.00		-18	0
3	σ_N=	2	N_i=	0.42		-16	8
4			X_i= S_i+N_i	-9.58		-14	40
5	n_T=	i=	bin#=			-12	112
6	1000	1000	6			-10	200
7						-8	113
8	hist min =	# bins=	width=			-6	26
9	-18	19	2			-4	3
10						-2	0
11						0	0
12						2	0
13	Hist					4	3
14						6	32
15						8	114
16						10	215
17						12	101
18						14	33
19						16	0
20						18	0

	A	B	C	D
1	D=	=2*ROUND(RAND(),0)-1		
2	A=	10	S_i=	=B1*B2
3	σ_N=	2	N_i=	=B3*NORM.S.INV(RAND())
4			X_i= S_i+N_i	=D2+D3
5	n_T=	i=	bin#=	
6	1000	1000	=ROUND((D4-A9)/C9,0)+2	
7				
8	hist min =	# bins=	width=	
9	= -ROUND(B2+4*B3,0)	19	=-ROUND(2*A9/(B9-1),0)	

Performing most of the computations in the worksheet allows the following VBA code to be relatively simple.

```
Sub hist()
Dim Val As Integer
Range("B6").Value = 0                ' Reset counter i
For Val = 1 To Range("B9").Value     ' Set Hist Bin Vals
    Range("F" & 1 + Val).Value = Range("A9").Value + (Val - 1) * Range("C9").Value
    Range("G" & 1 + Val).Value = 0  ' Initialize counts
Next
Do While Range("B6").Value < Range("A6").Value  ' loop for nT times
    Val = Round((Range("D4").Value - Range("A9").Value) / Range("C9").Value) + 2  'bin value
    If Val > 1 Then                            '  if valid bin value
        Range("G" & Val).Value = Range("G" & Val).Value + 1    ' incr bin count
    End If
    Range("B6").Value = Range("B6").Value + 1   ' increment counter i
Loop
End Sub
```

)