Homework 4

Problem 1

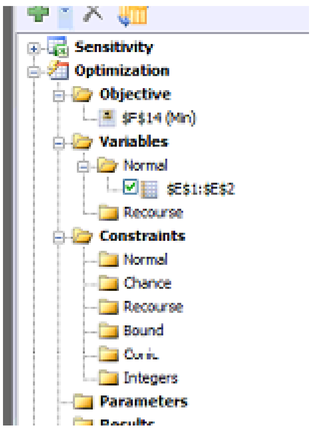
a)

The linear model is Y = a + b X

Where Y is the revenue, X is the hours of operation, *a* and *b* are parameters – the intercept and the slope respectively.

Set up

	A	B	C	D	E	F
1		Y=a+bX		a		
2				b		
3		Hours of operation	Average Revenue	a+bX (Predicted Revenue)	Error	Sq. error
4		40	5958	0	5958	35497764
5		44	6662	0	6662	44382244
6		48	6004	0	6004	36048016
7		48	6011	0	6011	36132121
8		60	7250	0	7250	52562500
9		70	8632	0	8632	74511424
10		72	6964	0	6964	48497296
11		90	11097	0	11097	123143409
12		100	9107	0	9107	82937449
13		168	11498	0	11498	132204004
14					Total	665916227



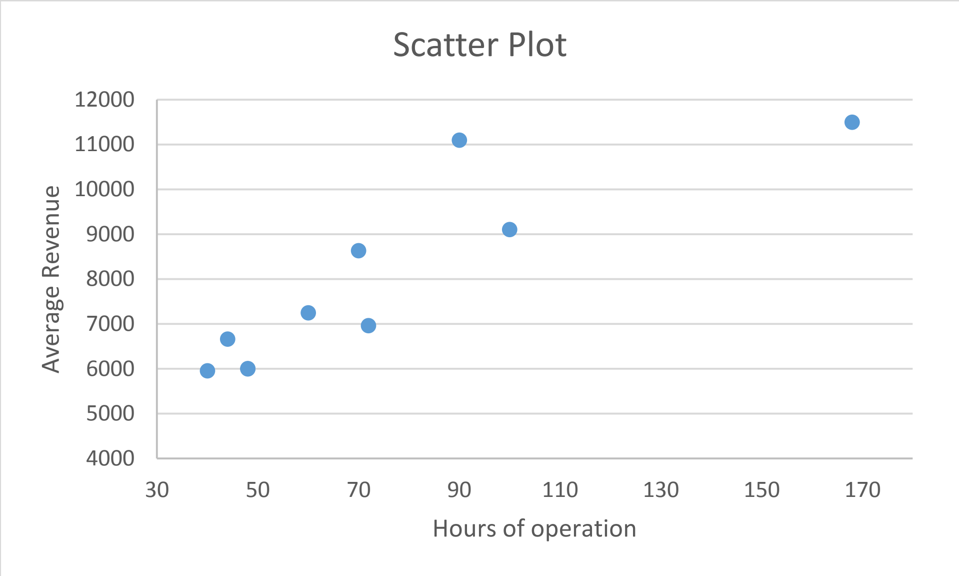
Solution

	A	B	C	D	E	F
1		Y=a+bX		a	4435.084	
2				b	47.07049	
3		Hours of operation	Average Revenue	a+bX (Predicted Revenue)	Error	Sq. error
4		40	5958	6317.903345	-359.903	129530.42
5		44	6662	6506.185305	155.8147	24278.219
6		48	6004	6694.467264	-690.467	476745.04
7		48	6011	6694.467264	-683.467	467127.5
8		60	7250	7259.313142	-9.31314	86.734617
9		70	8632	7730.018041	901.982	813571.46
10		72	6964	7824.15902	-860.159	739873.54
11		90	11097	8671.427838	2425.572	5883400.3
12		100	9107	9142.132736	-35.1327	1234.3091
13		168	11498	12342.92605	-844.926	713900.02
14					Total	9249747.6

The linear model is Predicted Revenue = 4435.08 + 47.07 * Hours of operation

When hours of operation = 120, then

Predicted Revenue = 4435.08 + 47.07 * 120 = **$10,083.54**



Decision variable is intercept and slope. The sum of squares of errors is the objective function which is to be minimized. There are no additional constraints.

Parameters are a and b which is the intercept and the slope respectively

b)

The non-linear model chosen is:

Y = a + b * sqrt(X)

Set up would be similar. The solution is:

16						
17	Y=a+b*sqrt(X)		a	174.467		
18			b	923.4198		
19	Hours of operation	Average Revenue	a+b*sqrt(X)	Error	Sq. error	
20	40	5958	6014.686613	-56.6866	3213.3721	
21	44	6662	6299.741007	362.259	131231.58	
22	48	6004	6572.107049	-568.107	322745.62	
23	48	6011	6572.107049	-561.107	314841.12	
24	60	7250	7327.246025	-77.246	5966.9485	
25	70	8632	7900.351358	731.6486	535309.73	
26	72	6964	8009.943845	-1045.94	1093998.5	
27	90	11097	8934.796434	2162.204	4675124.3	
28	100	9107	9408.665023	-301.665	91001.786	
29	168	11498	12143.3556	-645.356	416483.85	
30				Total	7589916.8	

The non-linear model is

Predicted Revenue = 174.47 + 923.42 * sqrt (Hours of operation)

When hours of operation = 120, then

Predicted Revenue = 174.47 + 923.42 * sqrt (120) = **$10,290.02**

I would prefer model (b) because the sum of squares of error is less in non-linear model than in the linear model. Hence non-linear model described in (b) gives a better fit than the linear model.

Note that on can develop other non-linear models like Y = a + b X + c X^2

Problem 2

The sales response curve to be used is:

$$S = a + \frac{(b-a)E^c}{(d+E^c)}$$

<u>Set up</u>

The excel setup is shown below.

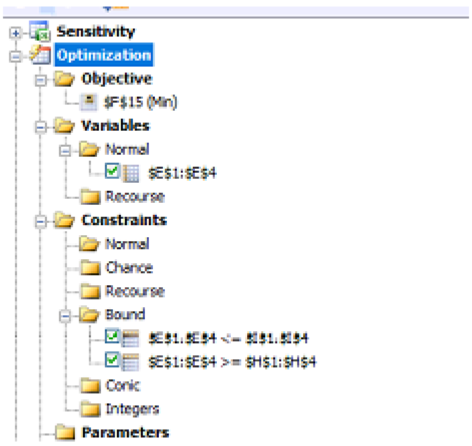
The objective function is the sum of the squares of the errors (highlighted in yellow) which is to be minimized.

The decision variables are the four parameters in the model, namely a, b, c and d

Effort is the independent variable and Sales is the dependent variable.

Standard Evolutionary Engine selected and lower and upper bounds provided for the decision variables.

	A	B	C	D	E	F	G	H	I
1				a				0	9.13517E+17
2				b				0	9.13517E+17
3				c				1	9.13517E+17
4				d				1000	9.13517E+17
5		Effort (X)	Sales (Y)	Predicted Sales	Error	Sq. error		Lower	Upper
6		0	50	#NUM!	#NUM!	#NUM!			
7		25	53	0	53	2809			
8		50	55	0	55	3025			
9		75	75	0	75	5625			
10		100	100	0	100	10000			
11		125	120	0	120	14400			
12		150	127	0	127	16129			
13		175	132	0	132	17424			
14		200	135	0	135	18225			
15					Minimize	#NUM!	Sum of squares of errors		
16									
17		Present Level							
18		500	200						



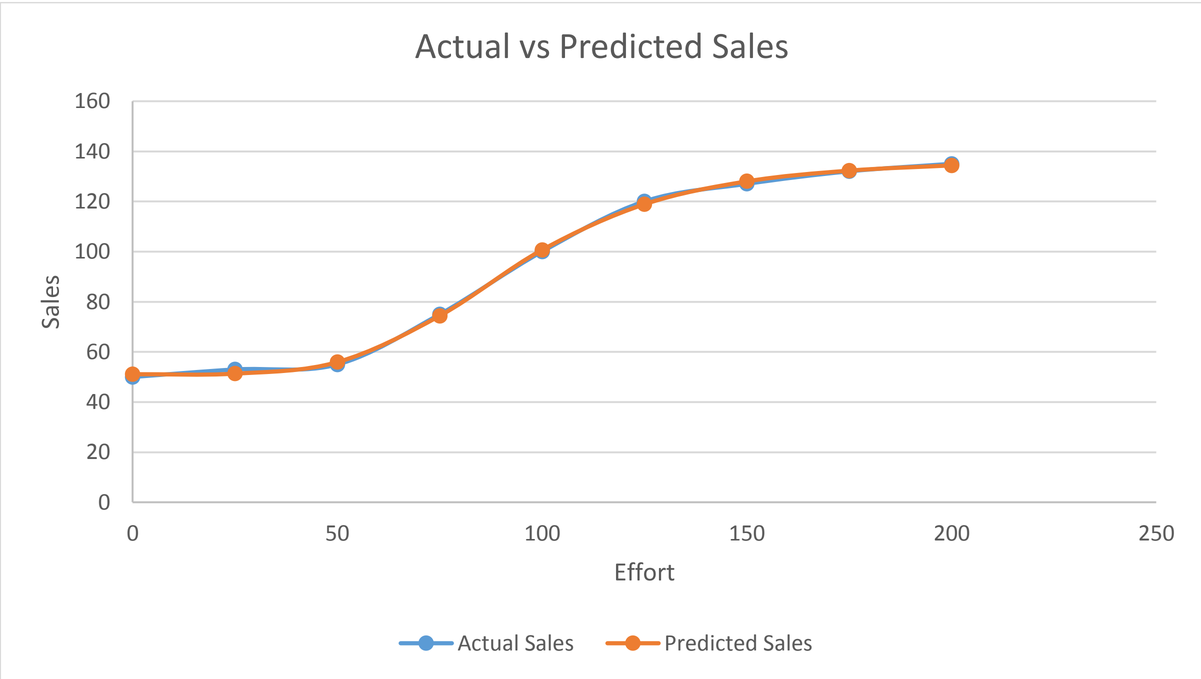
Solution

	A	B	C	D	E	F	G	H	I
1				a	51.09016			0	9.13517E+17
2				b	137.0583			0	9.13517E+17
3				c	4.515328			1	9.13517E+17
4				d	7.87E+08			1000	9.13517E+17
5		Effort (X)	Sales (Y)	Predicted Sales	Error	Sq. error		Lower	Upper
6		0	50	51.09015511	-1.09016	1.188438			
7		25	53	51.3136492	1.686351	2.843779			
8		50	55	55.92631036	-0.92631	0.858051			
9		75	75	74.39440676	0.605593	0.366743			
10		100	100	100.6803162	-0.68032	0.46283			
11		125	120	118.8973359	1.102664	1.215868			
12		150	127	128.0136064	-1.01361	1.027398			
13		175	132	132.2978534	-0.29785	0.088717			
14		200	135	134.3863611	0.613639	0.376553			
15					Minimize	8.428377	Sum of squares of errors		
16									

Based on the solution, the predicted sales equation is:

$$S = 51.1 + \frac{(137.1 - 51.1)E^{4.5}}{(7.87 * 10^8 + E^{4.5})}$$

The diagram below shows the actual versus predicted squares. The fit is quite good. The sum of squares is also very low.



The best values of the parameters that I obtained are:

a = 51.1, b = 137.1, c = 4.5 and d = $7.87 * 10^8$

b)

For Effort = 115%

$$S = 51.1 + \frac{(137.1 - 51.1)(115)^{4.5}}{(7.87 * 10^8 + 115^{4.5})} = 112.93$$

For effort of 115%, the predicted sales from the model is 112.93 (% of current).

Problem 3

The setup of the problem is done in Excel ASPE and is similar to the textbook.

Revenue = Total demand satisfied in 25 days * $45

Holding Cost = Total beginning inventory * $0.30

Ordering Cost = Total of column H (which is order) * $20

Opportunity Cost = (Total quantity demanded – Total demand satisfied) * $65

Profit = Revenue – Holding Cost – Ordering Cost – Opportunity Cost

Columns D (quantity demanded) and I (lead time) have simulation variable/function

We maximize the expected profit. The constraint is that reorder point and order quantity are non-negative and integer values.

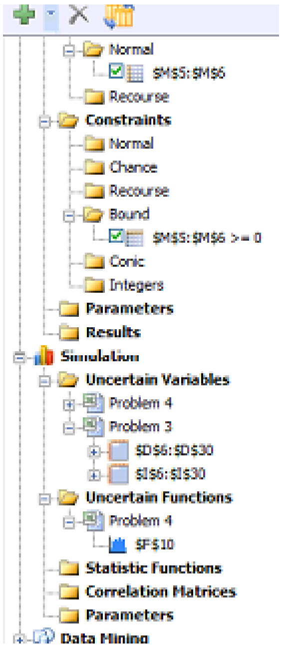
<u>Setup</u>

Day	Beginning Inventory	Units Received	Quantity Demanded	Demand satisfied	Ending Inventory	Inventory position	Order (0=No, 1=Yes)	Lead Time	Order arrives on day
1	50	0	7	7	43	43	0	0	0
2	43	0	3	3	40	40	0	0	0
3	40	0	7	7	33	33	0	0	0
4	33	0	7	7	26	26	0	0	0
5	26	0	4	4	22	22	0	0	0
6	22	0	3	3	19	19	0	0	0
7	19	0	6	6	13	13	0	0	0
8	13	0	6	6	7	7	0	0	0
9	7	0	6	6	1	1	0	0	0
10	1	0	10	1	0	0	0	0	0
11	0	0	9	0	0	0	0	0	0
12	0	0	1	0	0	0	0	0	0
13	0	0	4	0	0	0	0	0	0
14	0	0	5	0	0	0	0	0	0
15	0	0	5	0	0	0	0	0	0
16	0	0	7	0	0	0	0	0	0
17	0	0	5	0	0	0	0	0	0
18	0	0	9	0	0	0	0	0	0
19	0	0	6	0	0	0	0	0	0
20	0	0	6	0	0	0	0	0	0
21	0	0	5	0	0	0	0	0	0
22	0	0	7	0	0	0	0	0	0
23	0	0	5	0	0	0	0	0	0
24	0	0	8	0	0	0	0	0	0
25	0	0	7	0	0	0	0	0	0

Decision Variables	
Reorder Point	
Order Quantity	
Performance Measures	
Service Level	0.337838
Avg. Inventory	10.16
Revenue	2250
Holding Cost	76.2
Ordering Cost	0
Opportunity Cost	6370
Profit	-4196.2

Shipping Time	
Days	Probability
3	0.2
4	0.6
5	0.2
Total	1

Quantity Demanded	
Units	Probability
0	0.01
1	0.02
2	0.04
3	0.06
4	0.09
5	0.14
6	0.18
7	0.22
8	0.16
9	0.06
10	0.02
Total	1



<u>Solution</u>

The following solution was obtained in Excel

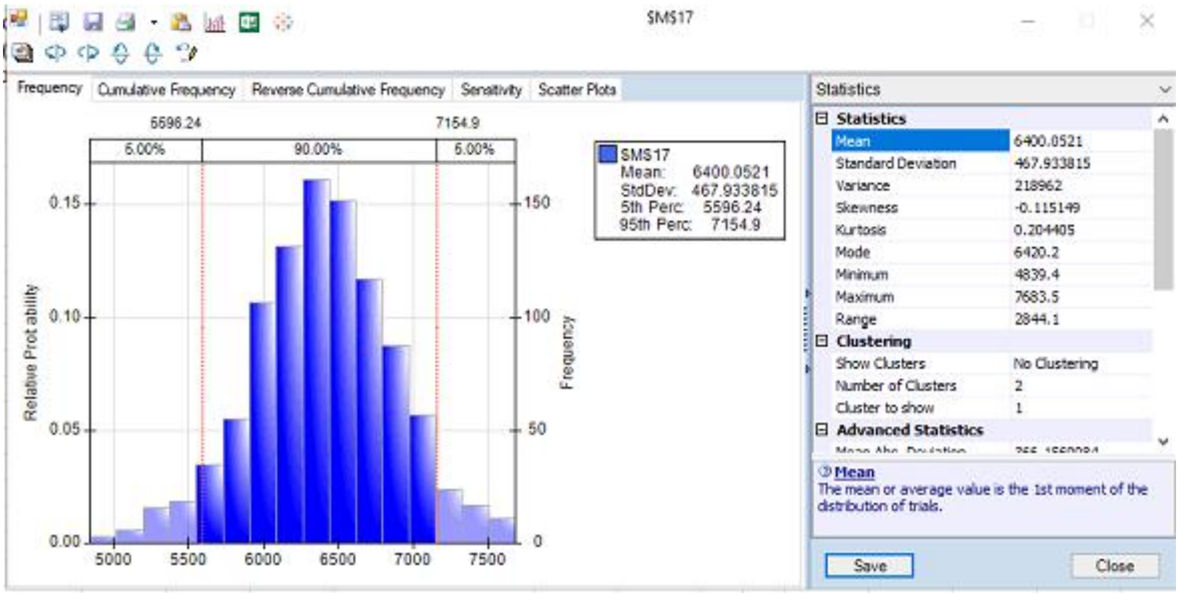
Day	Beginning Inventory	Units Received	Quantity Demanded	Demand satisfied	Ending Inventory	Inventory position	Order (0=No, 1=Yes)	Lead Time	Order arrives on day
1	50	0	5	5	45	45	0	0	0
2	45	0	6	6	39	39	1	4	7
3	39	0	7	7	32	63	0	0	0
4	32	0	7	7	25	56	0	0	0
5	25	0	4	4	21	52	0	0	0
6	21	0	6	6	15	46	0	0	0
7	15	31	9	9	37	37	1	4	12
8	37	0	5	5	32	63	0	0	0
9	32	0	8	8	24	55	0	0	0
10	24	0	4	4	20	51	0	0	0
11	20	0	10	10	10	41	0	0	0
12	10	31	9	9	32	32	1	4	17
13	32	0	7	7	25	56	0	0	0
14	25	0	5	5	20	51	0	0	0
15	20	0	5	5	15	46	0	0	0
16	15	0	1	1	14	45	0	0	0
17	14	31	8	8	37	37	1	3	21
18	37	0	5	5	32	63	0	0	0
19	32	0	7	7	25	56	0	0	0
20	25	0	6	6	19	50	0	0	0
21	19	31	6	6	44	44	0	0	0
22	44	0	6	6	38	38	1	4	27
23	38	0	3	3	35	66	0	0	0
24	35	0	9	9	26	57	0	0	0
25	26	0	7	7	19	50	0	0	0

Decision Variables	
Reorder Point	41
Order Quantity	31
Performance Measures	
Service Level	1
Avg. Inventory	28.48
Revenue	6975
Holding Cost	213.6
Ordering Cost	100
Opportunity Cost	0
Profit	6661.4

Shipping Time	
Days	Probability
3	0.2
4	0.6
5	0.2
Total	1
Quantity Demanded	
Units	Probability
0	0.01
1	0.02
2	0.04
3	0.06
4	0.09
5	0.14
6	0.18
7	0.22
8	0.16
9	0.06
10	0.02
Total	1

The reorder point and order quantity that maximize the average monthly profit associated with this monitor based on 1000 runs is 41 and 31 respectively.

Below shows the distribution of the average monthly profit associated with this monitor. The mean is $6400.



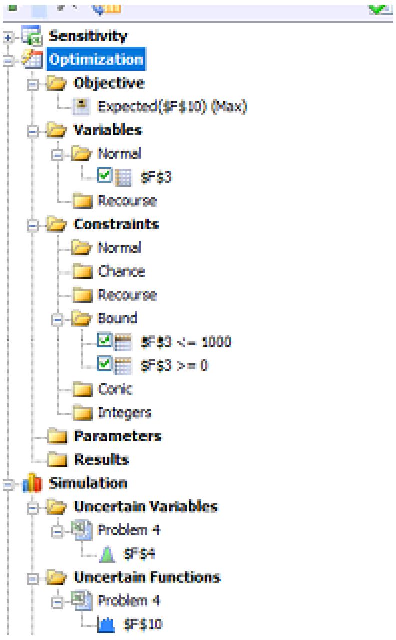
Problem 4

We first use simulation to solve the problem. The problem is set up in Excel as shown below. The formula for each cell is written next to it for illustration purpose.

Average daily net profit is to be maximized which is profit from on boarding less the compensation for refusals.

<u>Set up</u>

	A	B	C	D	E	F	G	H
1								
2								
3		Number of reservations				150		
4		Customers arrived				139	=psibinomial(F3,0.95)	
5		Number boarded (max 100)				100	=min(F4,100)	
6		Number of reservations refused				39	=max(0,F4-100)	
7		Profit from boarding				12000	=F5*(150-20)	
8		Compensation/loss for refusals				7800	=F6*100	
9								
10		Net profit (to be maximized)				4200	=F7-F8+psioutput()	
11								



<u>Solution</u>

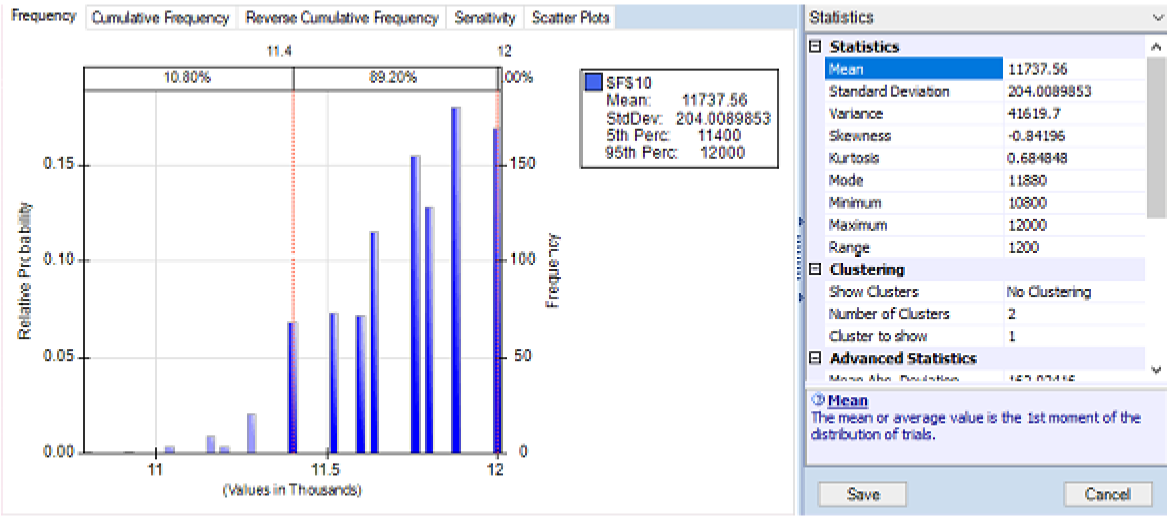
After the problem is solved in Excel, the result obtained is displayed below. Two results have been displayed.

	A	B	C	D	E	F	G	H
1								
2								
3		Number of reservations				105.6663		
4		Customers arrived				101	=psibinomial(F3,0.95)	
5		Number boarded (max 100)				100	=min(F4,100)	
6		Number of reservations refused				1	=max(0,F4-100)	
7		Profit from boarding				12000	=F5*(150-20)	
8		Compensation/loss for refusals				200	=F6*100	
9								
10		Net profit (to be maximized)				11800	=F7-F8+psioutput()	
11								

	A	B	C	D	E	F	G	H
1								
2								
3		Number of reservations				104.5428		
4		Customers arrived				98	=psibinomial(F3,0.95)	
5		Number boarded (max 100)				98	=min(F4,100)	
6		Number of reservations refused				0	=max(0,F4-100)	
7		Profit from boarding				11760	=F5*(150-20)	
8		Compensation/loss for refusals				0	=F6*100	
9								
10		Net profit (to be maximized)				11760	=F7-F8+psioutput()	

Note that every time we run the optimization, the outputs would be slightly different because of the uncertain variable and the uncertain function involved. However, the number of reservations that the hotel should accept to maximize the average daily profit would be around **105** reservations.

Mathematically calculating, since there is 5% chance of the guests not turning up, the number of reservations to be accepted should be 100/(1-0.05) = 105 (after rounding off). This aligns with the value obtained from the simulation exercise.



The average daily profit obtained from 1000 simulation runs is $11737.56. Note that the maximum daily profit possible is $12000 when exactly 100 guests arrive and there is no need to make any compensation for refusals.

Problem 5

The setup of the problem is Excel is as follows:

	A	B	C	D	E	F
1						
2			Max	Expected	Min	RNGS
3		R&D (in millions)	6	4	3	5.24
4		Market life	8		3	8
5		Units sold per year	350	250	50	149
6		Unit manufacturing cost	18000	14000	12000	16667
7						
8		Unit selling price	$ 23,000			
9		Cost of Capital	15%			
10						
11		PV of Future Profits	4,241,534			
12		R&D costs	(5,237,375)			
13		NPV	(995,841)			
14		Expected NPV	1,980,280			
15		P(NPV>0)	0.753			
16						

The RNGs used are as follows:

- For R&D, units sold per year and unit manufacturing cost, =psitriangular(min,expected, max) is used.
- For Market life = psiuniform is used.

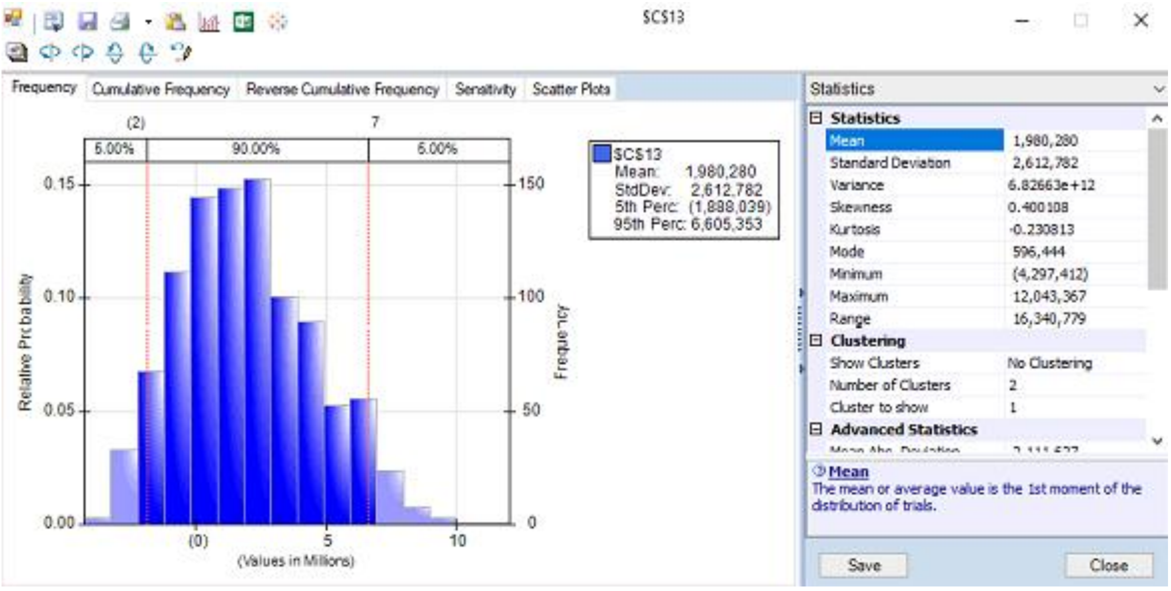
PV of future profits: = PV(15%,8,(23000-RNG of unit manufacturing cost)*RNG of units sold per year)

R&D cost is RNG of R&D multiplied by 1 million

NPV = PV of future profits – R&D costs

Expected NPV = psimean(NPV)

P(NPV>0): The formula used is =1-PsiTaget(NPV,0)



In this case the NPV of the project was -$995,841

However, based on the diagram above, NPV ranges from -$4,297,412 to $12,043,367.

The expected NPV of this project is $1,980,280

The probability of this project generating a positive NPV for the company is 0.753.

Extra Credit Problem

a)

```
insidecircle <- function(x,y){
  ifelse(x^2 + y^2 < 1,return(1),return(0))
}
```

Test:

```
> insidecircle(.1,.2)
[1] 1
```

b)

```
estimatepi <- function(N){ #Single input N
  # N pairs of uniform random numbers
  x <- runif(N)
  y <- runif(N)
  #Putting in data frame
  df <- data.frame(x=x,y=y)
  #using insidecircle function created in part (a)
  df$inside <- apply(df,1,function(x) insidecircle(x[1],x[2]))

  #estimation of pi
  pi_est <- 4*sum(df$inside)/length(df$inside)
  #Standard error
  se <- sd(df$inside)
  pi_se <- 4*se
  #95% confidence interval
  ci95 <- (1.96*pi_se)/sqrt(length(df$inside))

  return(list(pi=pi_est,standard.error=pi_se,ci95=ci95))
}
```

Test:

```
> estimatepi(1000)
$`pi`
[1] 3.064

$standard.error
[1] 1.694336

$ci95
[1] 0.105016
```

c)

```
results <- data.frame(n=c(),estimate=c(),se=c(),upper=c(),lower=c(),interval=c())
for(n in seq(1000,10000,by=500)){
  pi <- estimatepi(n)
  results <- rbind(results,c(n,pi$pi,pi$standard.error,
                             pi$pi+pi$ci95,pi$pi-pi$ci95,pi$ci95*2))
}
colnames(results) <- c("N","estimate", "se", "upper", "lower",
                       "interval")
results
```

Results table shown below:

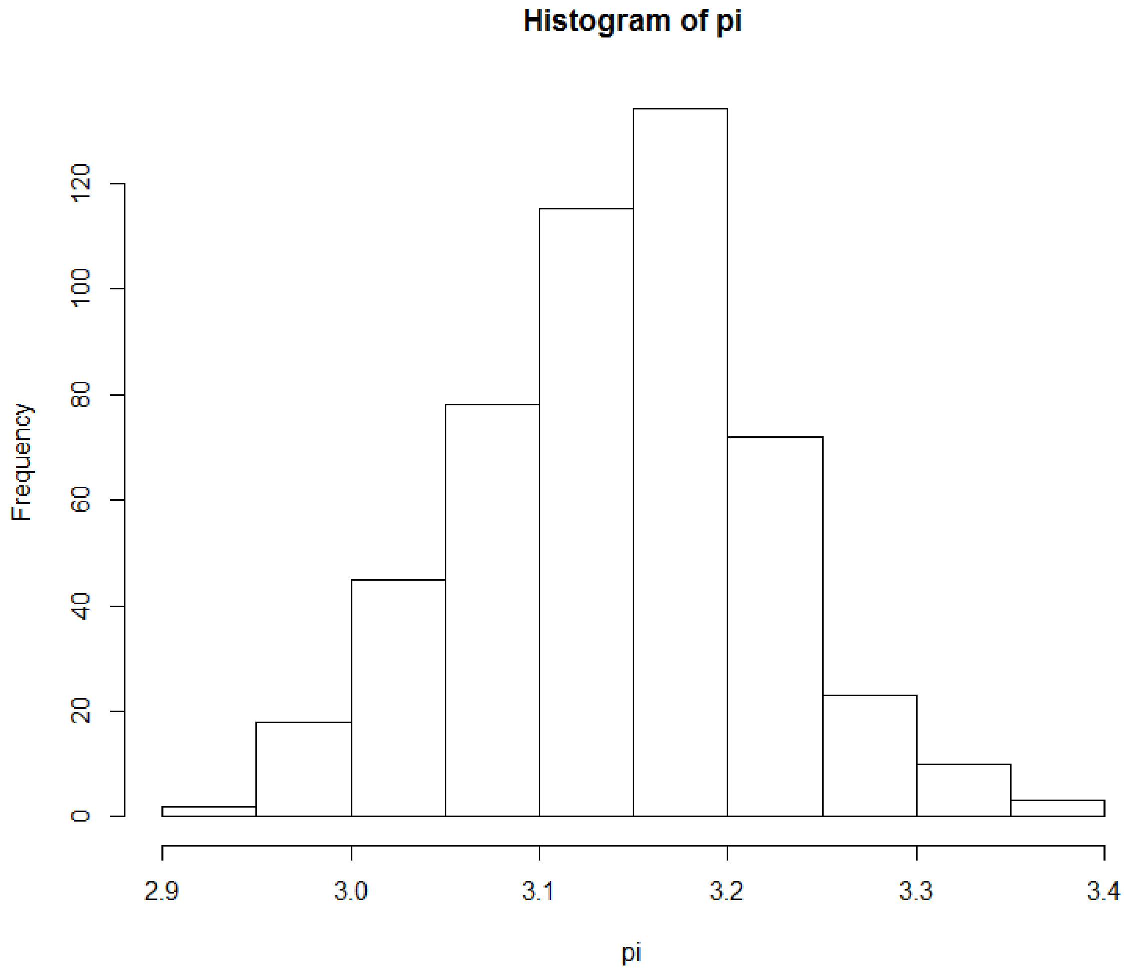
```
> results
       N estimate       se    upper    lower   interval
1   1000 3.156000 1.632890 3.257208 3.054792 0.20241516
2   1500 3.173333 1.620197 3.255327 3.091340 0.16398658
3   2000 3.114000 1.661440 3.186816 3.041184 0.14563167
4   2500 3.137600 1.645281 3.202095 3.073105 0.12899002
5   3000 3.140000 1.643563 3.198814 3.081186 0.11762832
6   3500 3.173714 1.619612 3.227372 3.120056 0.10731567
7   4000 3.140000 1.643495 3.190932 3.089068 0.10186487
8   4500 3.146667 1.638827 3.194550 3.098784 0.09576632
9   5000 3.132800 1.648426 3.178492 3.087108 0.09138406
10  5500 3.133818 1.647711 3.177365 3.090271 0.08709348
11  6000 3.163333 1.626990 3.204502 3.122165 0.08233706
12  6500 3.112615 1.662081 3.153022 3.072209 0.08081305
13  7000 3.136571 1.645780 3.175126 3.098017 0.07710967
14  7500 3.123200 1.654928 3.160655 3.085745 0.07490910
15  8000 3.152500 1.634649 3.188321 3.116679 0.07164166
16  8500 3.139765 1.643549 3.174705 3.104824 0.06988104
17  9000 3.128889 1.651034 3.163000 3.094778 0.06822143
18  9500 3.153684 1.633797 3.186539 3.120830 0.06570864
19 10000 3.140000 1.643372 3.172210 3.107790 0.06442016
```

N should be 4500 in order to ensure that the estimate of π is within 0.1 of the true value.

d)

Use n = 4500 (For n = 4000, the interval is more than 0.1)

```
pi <- c()
for(i in 1:500){
  pi <- c(pi,estimatepi(500)$pi)
}
hist(pi)
```



The histogram looks approximately normally distributed.

Standard deviation is 0.07699847

```
> sd(pi)
[1] 0.07699847
```

It is smaller than what we had before.