Gurjus Singh

November 8th, 2020

MSDS 460 – DECISION ANALYTICS

HW #4

1.

A.

To find the best fit one needs to know the form of a line which is y=ax+b. For this problem we are trying to minimize the Sum-Squared Error. The decisions variables are a and b while the optimal function is the SSE function. There are no constraints. The Formula and Excel setups are below:

The y = ax+b formula is represented here:

1	Hours of Operation	Average Revenue ($)			
2	40	5958		=B15*A2:A11 +B16	
3	44	6662		0	44382244
4	48	6004		0	36048016
5	48	6011		0	36132121
6	60	7250		0	52562500
7	70	8632		0	74511424
8	72	6964		0	48497296
9	90	11097		0	1.23E+08
10	100	9107		0	82937449
11	168	11498		0	1.32E+08
12					
13	Linear Form				
14	y = ax + b				
15	a coefficient	0			
16	b intercept	0			
17					
18	Cost	665916227			
19					
20					
21					
22					
23					
24					

Then the Sum Squared Errors is represented below:

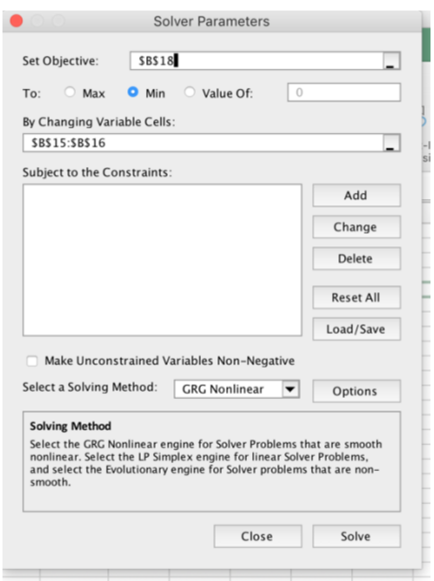
	A	B	C	D	E	F	G
1	Hours of Operation	Average Revenue ($)		Predicted			Predicte
2	40	5958		0	=(D2#-B2:B11)^2		6044.(
3	44	6662		0	44382244		6328.(
4	48	6004		0	36048016		6598.!
5	48	6011		0	36132121		6598.!
6	60	7250		0	52562500		7347
7	70	8632		0	74511424		7914.:
8	72	6964		0	48497296		8022..
9	90	11097		0	1.23E+08		8932.!
10	100	9107		0	82937449		9398.(
11	168	11498		0	1.32E+08		12066
12							
13	Linear Form						
14	y = ax + b						
15	a coefficient	0					
16	b intercept	0					
17							
18	Cost	665916227					
19							
20							
21							

	A	B	C	D	E	F
1	Hours of Operation	Average Revenue ($)		Predicted	Error	
2	40	5958		0	35497764	
3	44	6662		0	44382244	
4	48	6004		0	36048016	
5	48	6011		0	36132121	
6	60	7250		0	52562500	
7	70	8632		0	74511424	
8	72	6964		0	48497296	
9	90	11097		0	1.23E+08	
10	100	9107		0	82937449	
11	168	11498		0	1.32E+08	
12						
13	Linear Form					
14	y = ax + b					
15	a coefficient	0				
16	b intercept	0				
17						
18	Cost	=SUM(E2#)				
19						
20						
21						
22						

Here our all my numbers in each cell before solving:

	A	B	C
1	Hours of Operation	Average Revenue ($)	
2	40	5958	
3	44	6662	
4	48	6004	
5	48	6011	
6	60	7250	
7	70	8632	
8	72	6964	
9	90	11097	
10	100	9107	
11	168	11498	
12			
13	Linear Form		
14	y = ax + b		
15	a coefficient	0	
16	b intercept	0	
17			
18	Cost	665916227	
19			
20			

Then we input the constraints and optimization function into solver here:



And our final solution is when a = 47.07 and b = 4435.08

Hours of Operation	Average Revenue ($)
40	5958
44	6662
48	6004
48	6011
60	7250
70	8632
72	6964
90	11097
100	9107
168	11498
Linear Form	
y = ax + b	
a coefficient	47.07050415
b intercept	4435.082571
Cost	9249747.558

The model the linear model predicts for 120 hours is

Found by substituting into the linear model created which is y=47.07x+4435.08 where the revenue is 10,083.48.

B. To find our nonlinear model, you use the same type of formula but the formula is in exponential form such as Ax^b. Our decision variables are again both A and B. No need to have constraints.

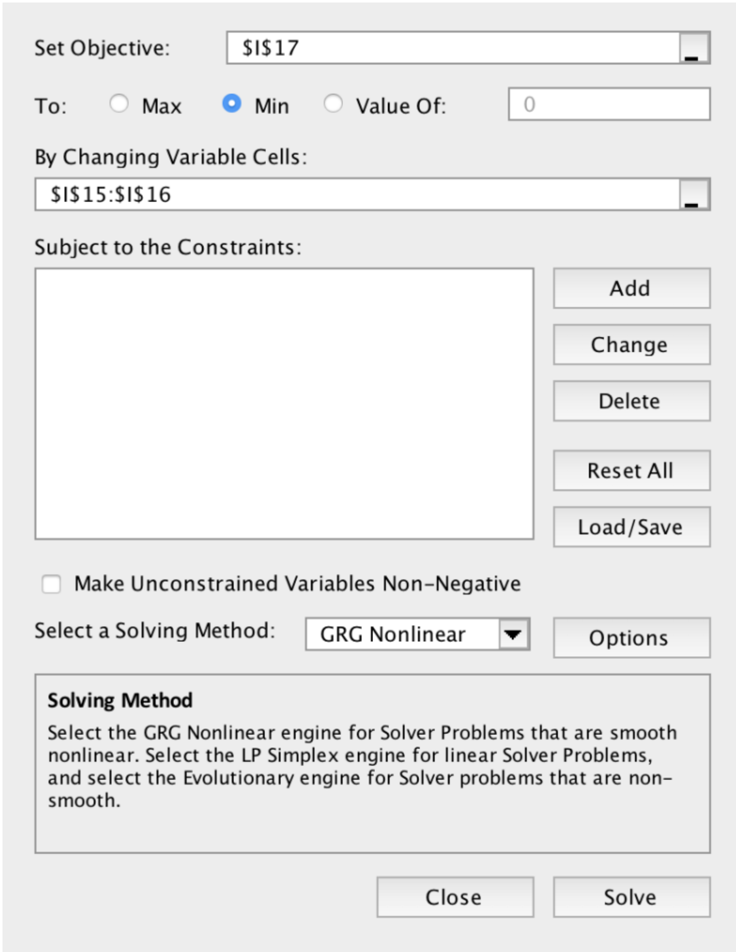
Excel set up is below:

=I15*(A2:A11)^(I16)			
6328.083	111500.303		
6598.978	353998.62		
6598.978	345717.93		
7347.89	9582.46285		
7914.327	515054.907		
8022.466	1120349.78		
8932.928	4683205.93		
9398.041	84704.7451		
12066.44	323122.174		
		NonLinear	Form
		ax^b	
	a	1022.224	
	b	0.481746	
	cost	7554650	

Predicted	Error		
6044.097	7412.74293		
6328.083	111500.303		
6598.978	353998.62		
6598.978	345717.93		
7347.89	9582.46285		
7914.327	515054.907		
8022.466	1120349.78		
8932.928	4683205.93		
9398.041	84704.7451		
12066.44	323122.174		
		NonLinear	Form
		ax^b	
	a	1022.224	
	b		
	cost	=SUM(H2#)	

Predicted	Error		
0	35497764		
0	44382244		
0	36048016		
0	36132121		
0	52562500		
0	74511424		
0	48497296		
0	123143409		
0	82937449		
0	132204004		
		NonLinear	Form
		ax^b	
	a	0	
	b	0	
	cost	6.66E+08	

In solver here is the setup:



The below is our optimal solution where a = 1022.224 and b = 0.4817

Predicted	Error		
6044.097	7412.74293		
6328.083	111500.303		
6598.978	353998.62		
6598.978	345717.93		
7347.89	9582.46285		
7914.327	515054.907		
8022.466	1120349.78		
8932.928	4683205.93		
9398.041	84704.7451		
12066.44	323122.174		
		NonLinear	Form
		ax^b	
	a	1022.224	
	b	0.481746	
	cost	7554650	

With this information we can create our formula which is 1022.224x^0.482 and we can predict for 120 hours our revenue is 85.625.

By looking at the above models I can predict that the best model is the nonlinear model for this particular dataset.

2.

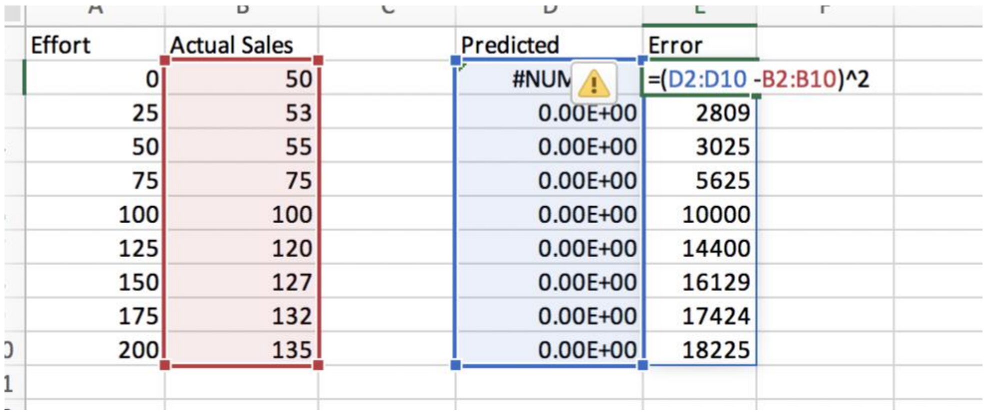
This problem is the same as number 1 but involves min and max constraints for the decision vairables.

The formula setup is below:

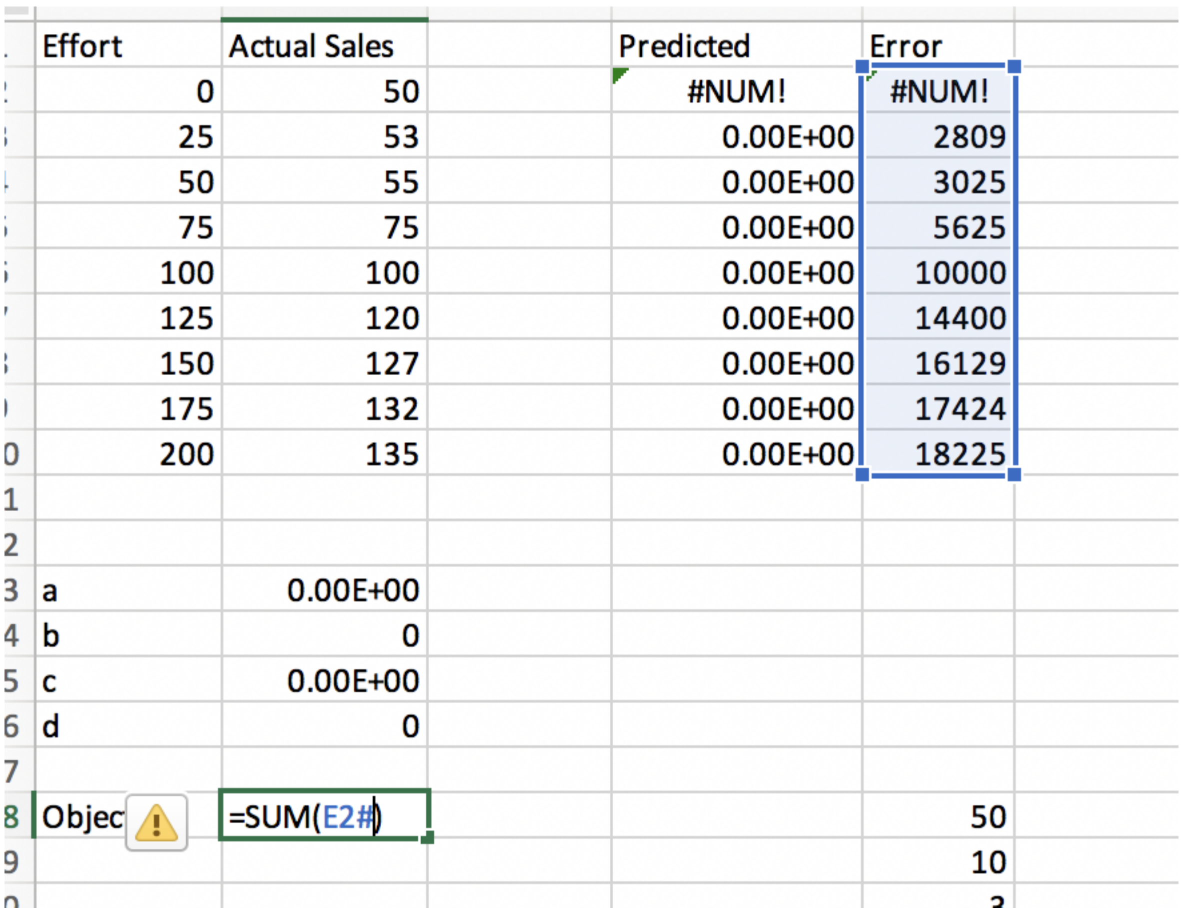
Below is our function formula the Sales function

Effort	Actual Sales		Predicted	Error
0	50		=$B$13 + ((($B$14-$B$13)*(A2^$B$15)) /($B$16 +(A2^$B$15)))	
25	53		0.00E+00	2809
50	55		0.00E+00	3025
75	75		0.00E+00	5625
100	100		0.00E+00	10000
125	120		0.00E+00	14400
150	127		0.00E+00	16129
175	132		0.00E+00	17424
200	135		0.00E+00	18225
a	0.00E+00			
b	0			
c	0.00E+00			
d	0			

Again I use SSE for optimal function below:



Effort	Actual Sales		Predicted	Error
0	50		#NUM	=(D2:D10 -B2:B10)^2
25	53		0.00E+00	2809
50	55		0.00E+00	3025
75	75		0.00E+00	5625
100	100		0.00E+00	10000
125	120		0.00E+00	14400
150	127		0.00E+00	16129
175	132		0.00E+00	17424
200	135		0.00E+00	18225



Effort	Actual Sales	Predicted	Error
0	50	#NUM!	#NUM!
25	53	0.00E+00	2809
50	55	0.00E+00	3025
75	75	0.00E+00	5625
100	100	0.00E+00	10000
125	120	0.00E+00	14400
150	127	0.00E+00	16129
175	132	0.00E+00	17424
200	135	0.00E+00	18225
a	0.00E+00		
b	0		
c	0.00E+00		
d	0		
Objec	=SUM(E2#)		50
			10

Here our upper and lower constraints for a, b, c, d, respectively where d is the in the millions.

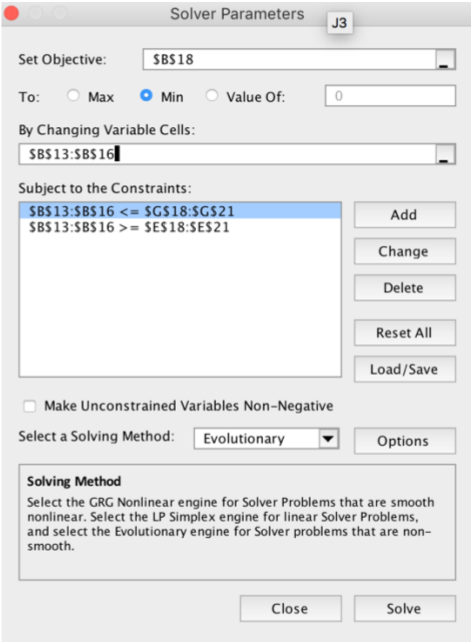
By first loosening constraints, and then tightening I got an optimal solution with bounds below.

50		100
10		200
3		100
8		800,000,000

Our whole sheet below:

	A	B	C	D	E	F	G	H
1	Effort	Actual Sales		Predicted	Error			
2	0	50		#NUM!	#NUM!			
3	25	53		0.00E+00	2809			
4	50	55		0.00E+00	3025			
5	75	75		0.00E+00	5625			
6	100	100		0.00E+00	10000			
7	125	120		0.00E+00	14400			
8	150	127		0.00E+00	16129			
9	175	132		0.00E+00	17424			
10	200	135		0.00E+00	18225			
11								
12								
13	a	0.00E+00						
14	b	0						
15	c	0.00E+00						
16	d	0						
17								
18	Objective	#NUM!			50		100	
19					10		200	
20					3		100	
21					8		1,000,000,000,000	
22								
23								
24								

Inputting into solver gets us the following:



A. Best values for a b c d are below:

Effort	Actual Sales		Predicted	Error		
0	50		5.11E+01	1.16E+00		
25	53		5.13E+01	2.87776		
50	55		5.59E+01	0.86708		
75	75		7.44E+01	0.33587		
100	100		1.01E+02	0.48918		
125	120		1.19E+02	1.1998		
150	127		1.28E+02	1.03639		
175	132		1.32E+02	0.09195		
200	135		1.34E+02	0.36806		
a	5.11E+01					
b	137.070194					
c	4.51E+00					
d	773855894					
Objective	8.43E+00			50		100
				10		200
				3		100
				8		800,000,000

Where A = 51.1, B =137.07, C = 4.51 and D = 773,855,894

The optimal solution is 8.43.

B.

With this information can say 115% effort predicts 133.82 in sales

With the formula 51.1 + (((137.07 - 51.1)*E^4.51)/(77385894 + E^4.51))

3.

For #3 we use the MCC problem from the book From Figure 12.23

We need to get the Maximum Average Profit and for this we use PSIMEAN(). Since we have referenced uncertain decision variables in our profit function we need to use PSIMEAN to find the mean for this function:

My formula sheet is set up below where the cells have the same formulas from example 12-23 :

Millennium Computer Corp.

Day	Beginning Inventory	Units Received	Quantity Demanded	Demand Satisfied	Ending Inventory	Inventory Position	Order? (0=n,1=y)	Lead Time	Order Arrives On Day
1	50	0	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D6,B6+C6)	=B6+C6-E6	=F6	=IF(G6<$M$5,1,0)	=IF(H6=0,0,Psi	=IF(I6=0,0,A6+1+I6)
2	=F6	=COUNTIF($J$6:J6	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D7,B7+C7)	=B7+C7-E7	=G6-E7+IF(H6=1,$M$6	=IF(G7<$M$5,1,0)	=IF(H7=0,0,Psi	=IF(I7=0,0,A7+1+I7)
3	=F7	=COUNTIF($J$6:J7	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D8,B8+C8)	=B8+C8-E8	=G7-E8+IF(H7=1,$M$6	=IF(G8<$M$5,1,0)	=IF(H8=0,0,Psi	=IF(I8=0,0,A8+1+I8)
4	=F8	=COUNTIF($J$6:J8	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D9,B9+C9)	=B9+C9-E9	=G8-E9+IF(H8=1,$M$6	=IF(G9<$M$5,1,0)	=IF(H9=0,0,Psi	=IF(I9=0,0,A9+1+I9)
5	=F9	=COUNTIF($J$6:J9	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D10,B10+C	=B10+C10-E10	=G9-E10+IF(H9=1,M	=IF(G10<$M$5,1,0)	=IF(H10=0,0,Ps	=IF(I10=0,0,A10+1+I10)
6	=F10	=COUNTIF($J$6:J1	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D11,B11+C11	=B11+C11-E11	=G10-E11+IF(H10=1,$M	=IF(G11<$M$5,1,0)	=IF(H11=0,0,Ps	=IF(I11=0,0,A11+1+I11)
7	=F11	=COUNTIF($J$6:J1	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D12,B12+C	=B12+C12-E12	=G11-E12+IF(H11=1,$M	=IF(G12<$M$5,1,0)	=IF(H12=0,0,Ps	=IF(I12=0,0,A12+1+I12)
8	=F12	=COUNTIF($J$6:J1	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D13,B13+C	=B13+C13-E13	=G12-E13+IF(H12=1,$	=IF(G13<$M$5,1,0)	=IF(H13=0,0,Ps	=IF(I13=0,0,A13+1+I13)
9	=F13	=COUNTIF($J$6:J1	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D14,B14+C	=B14+C14-E14	=G13-E14+IF(H13=1,$	=IF(G14<$M$5,1,0)	=IF(H14=0,0,Ps	=IF(I14=0,0,A14+1+I14)
10	=F14	=COUNTIF($J$6:J1	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D15,B15+C1	=B15+C15-E15	=G14-E15+IF(H14=1,$	=IF(G15<$M$5,1,0)	=IF(H15=0,0,Ps	=IF(I15=0,0,A15+1+I15)
11	=F15	=COUNTIF($J$6:J1	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D16,B16+C	=B16+C16-E16	=G15-E16+IF(H15=1,$	=IF(G16<$M$5,1,0)	=IF(H16=0,0,Ps	=IF(I16=0,0,A16+1+I16)
12	=F16	=COUNTIF($J$6:J1	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D17,B17+C1	=B17+C17-E17	=G16-E17+IF(H16=1,$	=IF(G17<$M$5,1,0)	=IF(H17=0,0,Ps	=IF(I17=0,0,A17+1+I17)
13	=F17	=COUNTIF($J$6:J1	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D18,B18+C	=B18+C18-E18	=G17-E18+IF(H17=1,$	=IF(G18<$M$5,1,0)	=IF(H18=0,0,Ps	=IF(I18=0,0,A18+1+I18)
14	=F18	=COUNTIF($J$6:J1	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D19,B19+C	=B19+C19-E19	=G18-E19+IF(H18=1,$	=IF(G19<$M$5,1,0)	=IF(H19=0,0,Ps	=IF(I19=0,0,A19+1+I19)
15	=F19	=COUNTIF($J$6:J1	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D20,B20+C	=B20+C20-E20	=G19-E20+IF(H19=1,$	=IF(G20<$M$5,1,0)	=IF(H20=0,0,P	=IF(I20=0,0,A20+1+I20)
16	=F20	=COUNTIF($J$6:J2	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D21,B21+C	=B21+C21-E21	=G20-E21+IF(H20=1,$	=IF(G21<$M$5,1,0)	=IF(H21=0,0,Ps	=IF(I21=0,0,A21+1+I21)
17	=F21	=COUNTIF($J$6:J2	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D22,B22+C	=B22+C22-E22	=G21-E22+IF(H21=1,$	=IF(G22<$M$5,1,0)	=IF(H22=0,0,P	=IF(I22=0,0,A22+1+I22)
18	=F22	=COUNTIF($J$6:J2	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D23,B23+C	=B23+C23-E23	=G22-E23+IF(H22=1,$	=IF(G23<$M$5,1,0)	=IF(H23=0,0,P	=IF(I23=0,0,A23+1+I23)
19	=F23	=COUNTIF($J$6:J2	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D24,B24+C	=B24+C24-E24	=G23-E24+IF(H23=1,$	=IF(G24<$M$5,1,0)	=IF(H24=0,0,P	=IF(I24=0,0,A24+1+I24)
20	=F24	=COUNTIF($J$6:J2	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D25,B25+C	=B25+C25-E25	=G24-E25+IF(H24=1,$	=IF(G25<$M$5,1,0)	=IF(H25=0,0,P	=IF(I25=0,0,A25+1+I25)
21	=F25	=COUNTIF($J$6:J2	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D26,B26+C	=B26+C26-E26	=G25-E26+IF(H25=1,$	=IF(G26<$M$5,1,0)	=IF(H26=0,0,P	=IF(I26=0,0,A26+1+I26)
22	=F26	=COUNTIF($J$6:J2	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D27,B27+C	=B27+C27-E27	=G26-E27+IF(H26=1,$	=IF(G27<$M$5,1,0)	=IF(H27=0,0,P	=IF(I27=0,0,A27+1+I27)
23	=F27	=COUNTIF($J$6:J2	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D28,B28+C	=B28+C28-E28	=G27-E28+IF(H27=1,$	=IF(G28<$M$5,1,0)	=IF(H28=0,0,P	=IF(I28=0,0,A28+1+I28)
24	=F28	=COUNTIF($J$6:J2	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D29,B29+C	=B29+C29-E29	=G28-E29+IF(H28=1,$	=IF(G29<$M$5,1,0)	=IF(H29=0,0,P	=IF(I29=0,0,A29+1+I29)
25	=F29	=COUNTIF($J$6:J2	=PsiDiscrete(Data!$F$7:$F$17,Data!$G$7:$G$17)	=MIN(D30,B30+C	=B30+C30-E30	=G29-E30+IF(H29=1,$	=IF(G30<$M$5,1,0)	=IF(H30=0,0,P	=IF(I30=0,0,A30+1+I30)
26	=F30	=COUNTIF($J$6:J3		=MIN(D31,B31+C	=B31+C31-E31	=G30-E31+IF(H30=1,$	=IF(G31<$M$5,1,0)		=IF(I31=0,0,A31+1+I31)
27	=F31	=COUNTIF($J$6:J3		=MIN(D32,B32+C	=B32+C32-E32	=G31-E32+IF(H31=1,$	=IF(G32<$M$5,1,0)		=IF(I32=0,0,A32+1+I32)
28	=F32	=COUNTIF($J$6:J3		=MIN(D33,B33+C	=B33+C33-E33	=G32-E33+IF(H32=1,$	=IF(G33<$M$5,1,0)		=IF(I33=0,0,A33+1+I33)
29	=F33	=COUNTIF($J$6:J3		=MIN(D34,B34+C	=B34+C34-E34	=G33-E34+IF(H33=1,$	=IF(G34<$M$5,1,0)		=IF(I34=0,0,A34+1+I34)
30	=F34	=COUNTIF($J$6:J3		=MIN(D35,B35+C	=B35+C35-E35	=G34-E35+IF(H34=1,$	=IF(G35<$M$5,1,0)		=IF(I35=0,0,A35+1+I35)

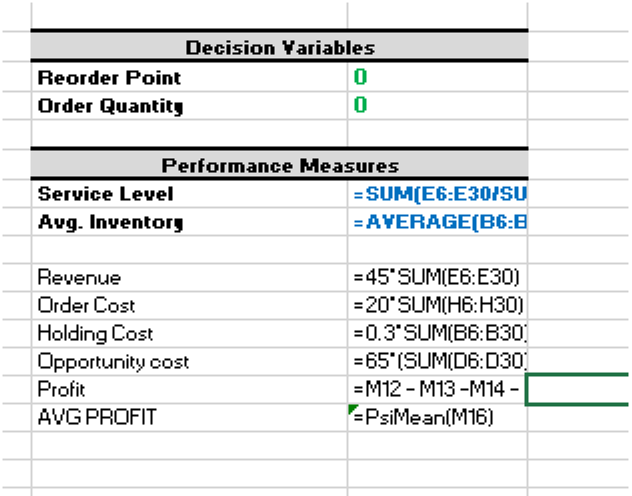
The formulas for revenue reference row cells 6-30 since we only need to calculate 25 days, we know we will get our revenue from demand satisfied * 45,

Order Costs is from the Order Binary column, using Sum Function, multiplied by 20

We also know holding cost is Beginning inventory*0.30

We know opportunity cost is 65*demand missed which is demanded-satisfied demand

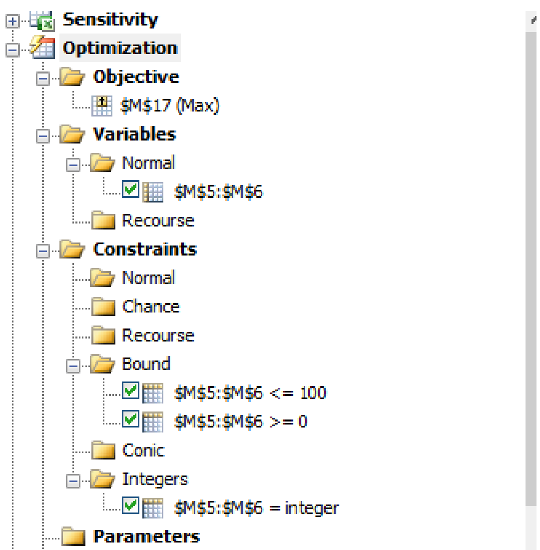
Our profit is the Revenue-all the total costs

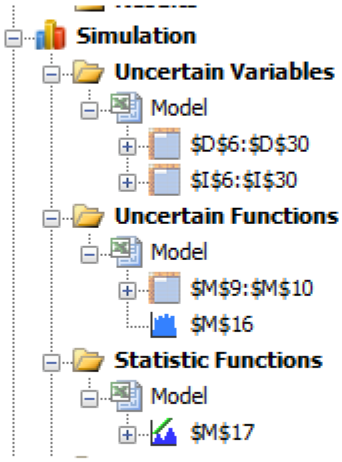


Decision Variables	
Reorder Point	0
Order Quantity	0
Performance Measures	
Service Level	=SUM(E6:E30/SU
Avg. Inventory	=AVERAGE(B6:B
Revenue	=45*SUM(E6:E30)
Order Cost	=20*SUM(H6:H30)
Holding Cost	=0.3*SUM(B6:B30)
Opportunity cost	=65*(SUM(D6:D30)
Profit	=M12 - M13 -M14 -
AVG PROFIT	=PsiMean(M16)

Once we know all the formulas, we can input into solver as shown below:

From our solver our constraints are that Reorder Point and Order Quantity, which are decision variables as mentioned in the question, will be greater than 0 and less than 100 by trial and error. We will maximize our average profit as mentioned in the question. We should also make our decision variables integers as they should not be fractional instances.





From our answer our average profit is $6,429.8 and our decision variables reorder point = 40, and order quantity = 37.

Decision Variables	
Reorder Point	40
Order Quantity	37
Performance Measures	
Service Level	5.1%
Avg. Inventory	35.80
Revenue	5310
Order Cost	60
Holding Cost	268.5
Opportunity cost	0
Profit	4981.5
AVG PROFIT	6429.8

4.

For the Hotel Problem my excel sheet is below:

Room Capacity		100	rooms		150	cost/night
overcapcacity per person		200		Cost/room	30	
Rooms Reserved	=PsiBinomial(B5, 1-C6)					
Number of reservation	107					
Guest Cancels		0.05				
Variable costs		=F3*C1				
Cost for Overcapacity	=200*MAX(0, B4-C1)					
Profit	=150*B5-C8-B9					
Average Profit	=PsiMean(B10)					

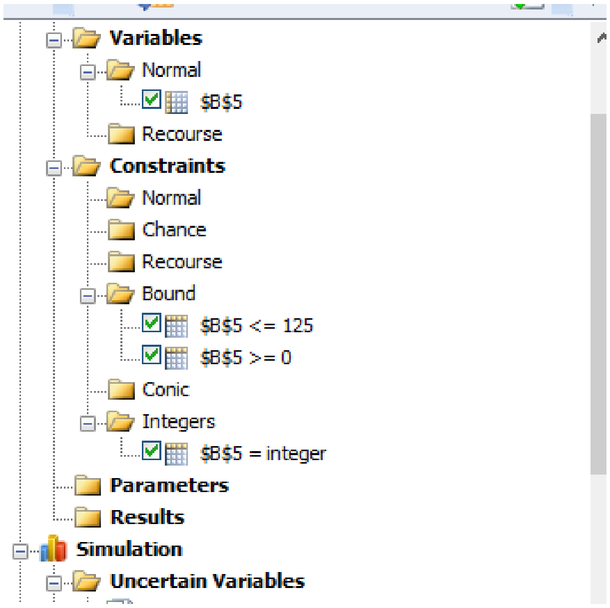
From above we use PsiBinomial a Binomial Distribution that gives us the distribution probability of x successes in n trials.

We know that 95% of the people do not cancel and 5% do. With this information our PsiBinomial function has references of (number of reservations, 1-0.05)

We also know the cost of overcapacity is 200*max(0, the difference between rooms reserved and number of rooms that are actually available). This will give us either 0, or the discrepancy.

We also know the hotel make revenue of 150 per night so we multiply this by number of reservations and subtract costs of overcapacity, variable costs which is 30 dollars per room This will get our profit and our goal is to maximize average profit.

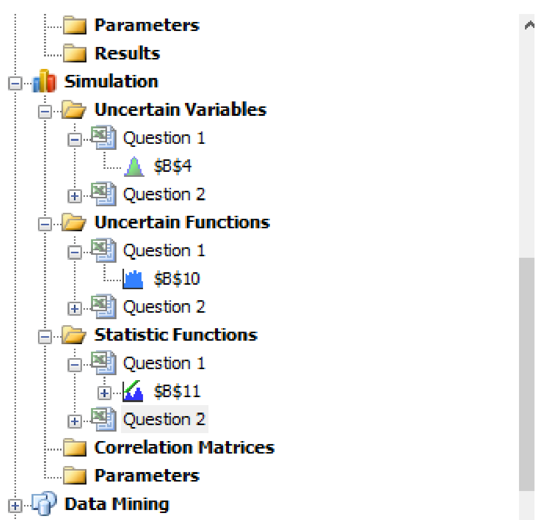
This is what I put into analytical solver:



Our reservations cell is the decision variable which will be greater than 0 and less than 125.

We also need to make it an integer.

Since Rooms has a binomial distribution of random variables it is an uncertain variable, while Average Profits is a statistical function where we need to use PSIMEAN as it references an uncertain function and variables in the profits cell.



After inputting in our solver, our max average profit is 12,654.40 and 107 reservations, but we have and we have overbooked by 3 rooms. We also have profit of 12,450.

Room Capacity		100	rooms		150	cost/night
overcapacity per person		200		Cost/room	30	
Rooms Reserved	103					
Number of reservation	107					
Guest Cancels		0.05				
Variable costs		3000				
Cost for Overcapacity	600					
Profit	12450					
Average Profit	12654.4					

5.

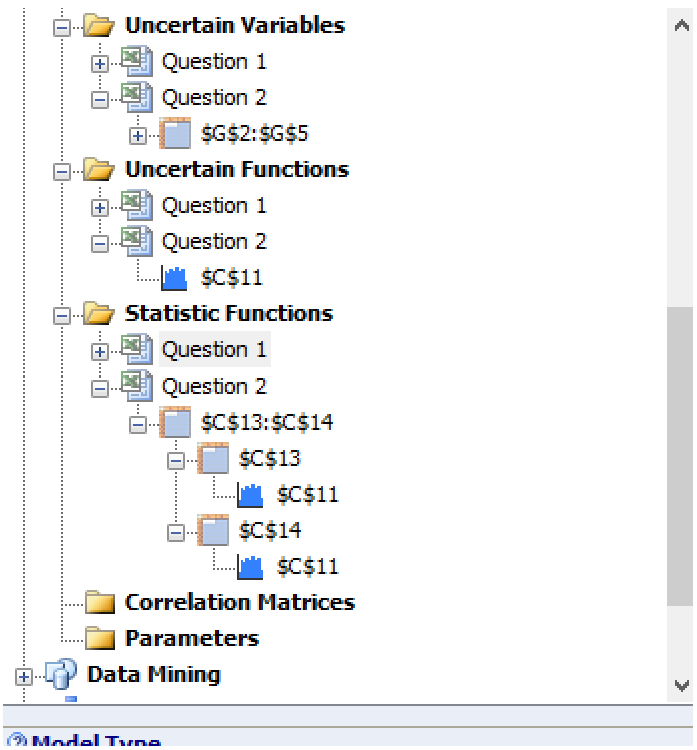
The Excel setup is below:

	A	B	C	D	E	F	G	H
1				Min	Expected	Max		
2			RD COSTS	3	4	6	=PsiTriangular(D2,E2,F2)	
3			Units Sold	50	250	350	=INT(PsiTriangular(D3,E3,F3))	
4			Market Life	3	8		=INT(PsiUniform(D4,E4))	
5			cost manuf	12000	14000	18000	=PsiTriangular(D5,E5,F5)	
6								
7		Sell price	23000					
8		Cost of Cap	0.15					
9		PV Future	=PV(C8,G4, -(C7-G5)*G3)					
10		RD COSTS	=G2*1000000					
11		NPV	=C9-C10 + PsiOutput()					
12								
13		Expected NPV	=PsiMean(C11)					
14		Positive Prob	=1-PsiTarget(C11,0)					
15								
16								
17								
18								
19								

As one can see we need PsiTriangular functions for RD Costs, Units Sold, Manufacturing Costs, as they have expected, max and min values. We use PsiUniform as For Market Life as it is a range with equal distribution, and INT is used to make sure random variables for Units and Market Life are whole numbers.

We have a special function for PV(), NPV is our net Present Value and is computed by Present Value Future Cell, and Research and Development Cost. Our Expected Present Value is found by the PSIMean and once can figure out about getting the probability of getting positive NPV by using PSITarget which gives probability of Expected Value being less than 0, so to figure out the Expected Value more than zero you have to subtract value from 1.

I have put the above in analytical solver:



Our uncertain variables are the cells with PSITriangular, and PSIUniform, and Uncertain function is NPV function. Our Stats functions are ones containing PSITarget and PSIMean().

	A	B	C	D	E	F	G	H	I	J	K	L	M
1				Min	Expected	Max							
2			RD COSTS	3	4	6	3.986537485						
3			Units Sold	50	250	350	151						
4			Market Life	3	8		6						
5			cost manuf	12000	14000	18000	13797.6458						
6													
7		Sell price	23,000										
8		Cost of Cap	0.15										
9		PV Future	$5,258,748.68										
10		RD COSTS	3986537.485										
11		NPV	$1,272,211.20										
12													
13		Expected NPV	1600919.546										
14		Positive Prob	0.72										
15													

After running our model, we get NPV of 1,272,211.20 and Expected of 1,600,919.546 and Probability greater than 0 of 0.72.

Extra Credit

A.

Here is my function for inside circle:

```
def inside_circle(x, y):
    c = x**2 + y**2
    if c < 1:
        return 1

    return 0
```

I create a function for the circle which is x^2 +y^2 and we want to know if there points in the circle that is why I create an if statement so it’s < 1.

B.

For B I create this function:

```
In [125]: import numpy as np
          import scipy.stats as st
          def estimatepi(n):
              pointlst = []
              for i in range(1, n + 1):
                  x = np.random.uniform(0,1)
                  y = np.random.uniform(0,1)
                  pointlst.insert(0, 4*inside_circle(x, y))
              estimatepinum = np.mean(pointlst)
              #standard error
              ste= st.sem(pointlst)
              #95% CI
              CI95 = st.t.interval(0.95, len(pointlst)-1, loc=np.mean(pointlst), scale=ste)
              print("estimate of pi: " + str(estimatepinum))
              print("standard error: " + str(ste))
              print("95% confience interval: " + str(CI95))
              Print
              print(" ")
```

I utilize numpy and scipy.stats package for standard error, mean and confidence interval.

C.

I then create this code to find the stats for N = 1,000 to 10,000 below:

```
]: for i in range(1000, 10001, 500):
       print("N = " + str(i))
       estimatepi(i)
```

The following is the output for the above code:

```
N = 1000
estimate of pi: 3.172
standard error: 0.051274214231376035
95% confience interval: (3.0713824835267323, 3.272617516473268)
Confidence range: 0.20123503294653577

N = 1500
estimate of pi: 3.1413333333333333
standard error: 0.042419788312081146
95% confience interval: (3.0581248904429366, 3.22454177622373)
Confidence range: 0.1664168857807935

N = 2000
estimate of pi: 3.104
standard error: 0.037299968415829936
95% confience interval: (3.030849113968978, 3.177150886031022)
Confidence range: 0.14630177206204387

N = 2500
```

```
N = 2500
estimate of pi: 3.1856
standard error: 0.03222043324549517
95% confience interval: (3.1224185102275688, 3.248781489772431)
Confidence range: 0.12636297954486242

N = 3000
estimate of pi: 3.1733333333333333
standard error: 0.029575686016795694
95% confience interval: (3.115342649656755, 3.2313240170099116)
Confidence range: 0.11598136735315645

N = 3500
estimate of pi: 3.1474285714285712
standard error: 0.027693086194893923
95% confience interval: (3.0931323379691857, 3.2017248048879567)
Confidence range: 0.108592466918771
```

```
N = 4000
estimate of pi: 3.158
standard error: 0.025786144172943483
95% confience interval: (3.1074447848125533, 3.2085552151874466)
Confidence range: 0.10111043037489331

N = 4500
estimate of pi: 3.144888888888889
standard error: 0.024448708993274535
95% confience interval: (3.0969574048590904, 3.1928203729186877)
Confidence range: 0.09586296805959726

N = 5000
estimate of pi: 3.1192
standard error: 0.023443297328616712
95% confience interval: (3.0732408539118032, 3.165159146088197)
Confidence range: 0.09191829217639391
```

```
N = 5500
estimate of pi: 3.1614545454545455
standard error: 0.021956591185748442
95% confience interval: (3.1184109433735103, 3.2044981475355807)
Confidence range: 0.08608720416207039

N = 6000
estimate of pi: 3.1166666666666667
standard error: 0.02142239667044229
95% confience interval: (3.074671067681902, 3.1586622656514316)
Confidence range: 0.08399119796952981

N = 6500
estimate of pi: 3.1587692307692308
standard error: 0.020220563565962746
95% confience interval: (3.119130272151849, 3.1984081893866128)
Confidence range: 0.07927791723476396
```

```
N = 7000
estimate of pi: 3.126857142857143
standard error: 0.019750526441036358
95% confience interval: (3.0881401268902935, 3.165574158823992)
Confidence range: 0.07743403193369858

N = 7500
estimate of pi: 3.1136
standard error: 0.019184240320557138
95% confience interval: (3.07599351009255, 3.151206489900745)
Confidence range: 0.07521297980149022

N = 8000
estimate of pi: 3.117
standard error: 0.018549444337652123
95% confience interval: (3.080638255119819, 3.153361744880181)
Confidence range: 0.07272348976036191
```

```
N = 8500
estimate of pi: 3.147294117647059
standard error: 0.017769875288649l8
95% confience interval: (3.11246084138606, 3.1821273939080577)
Confidence range: 0.06966655252199772

N = 9000
estimate of pi: 3.1244444444444444
standard error: 0.017435368534593893
95% confience interval: (3.090267153227338, 3.158621735661551)
Confidence range: 0.06835458243421311

N = 9500
estimate of pi: 3.126736842105263
standard error: 0.016954277634964304
95% confience interval: (3.0935028338812662, 3.15997085032926)
Confidence range: 0.0664680164479936
```

```
N = 10000
estimate of pi: 3.132
standard error: 0.016488929270236072
95% confience interval: (3.09967838000872, 3.1643216199912803)
Confidence range: 0.06464323998256027
```

When N = 4,500 one gets within 0.1 of the true estimate.

D. After running pi estimate 500 times with N = 4,500, I see a normal distribution of the Histogram. The code is below where I utilize matplotlib to plot histogram, first creating list which I pass into plthist() function:

```
import matplotlib.pyplot as plt
piest500 = []
for i in range(500):
    piest500.append(estimatepi(4500))
```

In [140]:
```
plt.hist(piest500)
plt.xlabel('Pi estimates')
plt.ylabel('frequency')
plt.show()
```