**ADM3305 – Assignment 1**

**Rylie Austin**

**8210704**

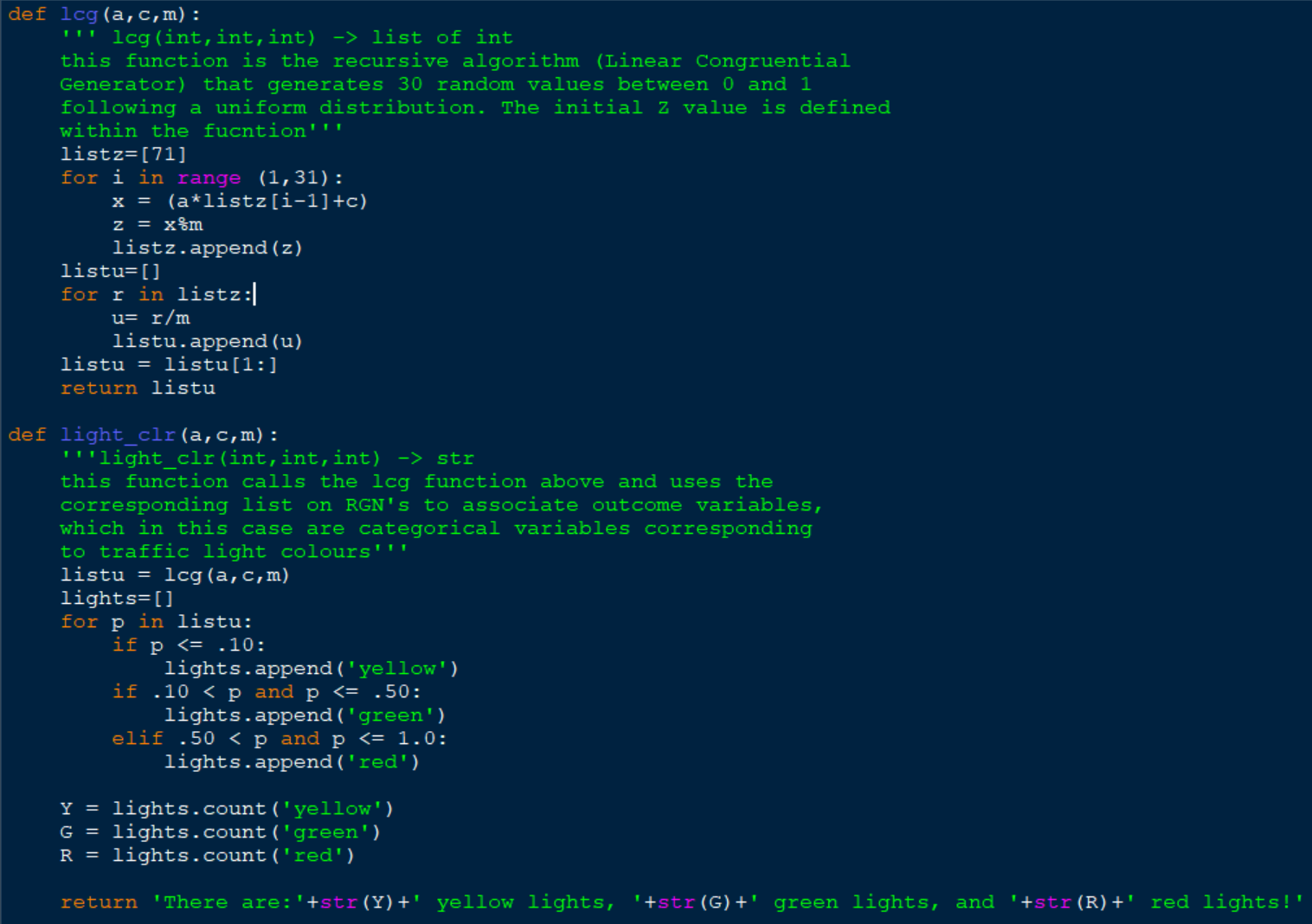
**Due: 2018-09-21**

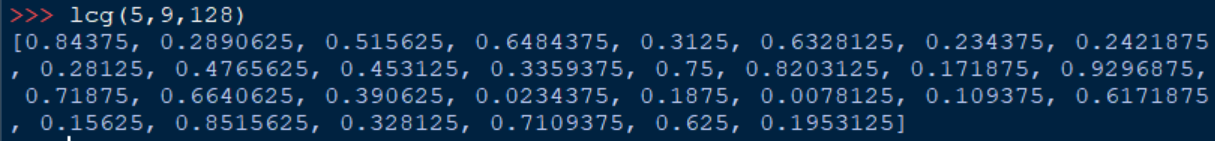
**Professor: Antoine Sauré**

**Part 1**

**a)**

The following Python code is a linear congruential generator with parameters a = 5, c = 9, m = 128 and Z­­0 = 71 used to generate a series of 30 numbers uniformly distributed in the interval [0,1].



a) Below is the python output of the 30 RN’s generated by the LCG function shown above:

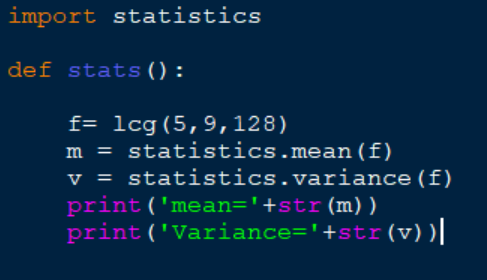
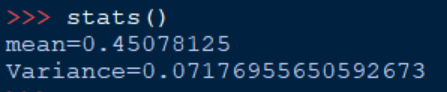
Here are the RN’s in tabular format:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| .844 | .289 | .516 | .648 | .313 |
| .633 | .234 | .242 | .281 | .477 |
| .453 | .336 | .75 | .82 | .172 |
| .93 | .719 | .664 | .391 | .03 |
| .188 | .008 | .109 | .617 | .156 |
| .852 | .328 | .711 | .625 | .195 |

Theoretically, the expected mean and variance of a continuous U(0,1) is:

|  |  |
| --- | --- |
| mean = 0.5 | Var = (1/12) = .0833 |

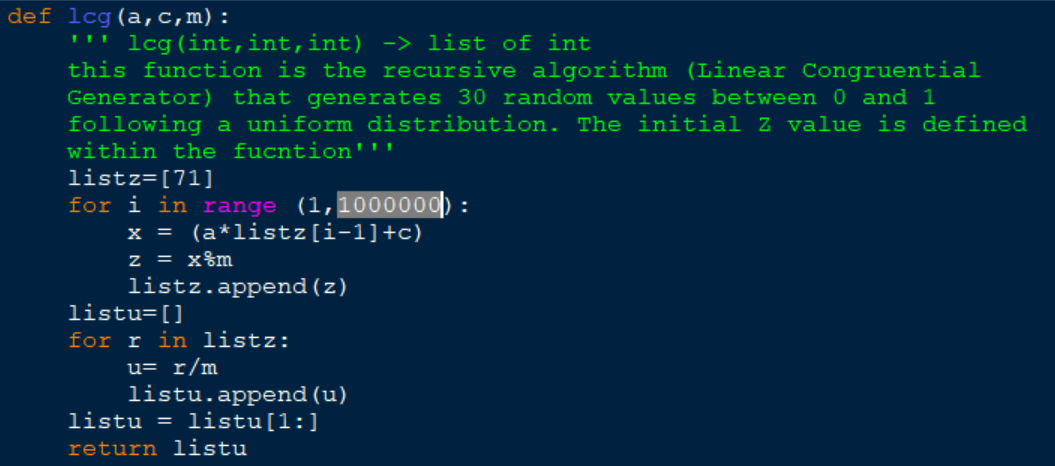
For the series numbers generated in this question the mean and variance are calculated as follows:



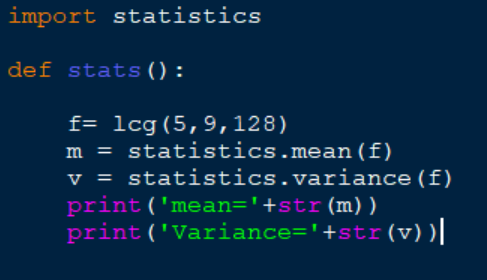
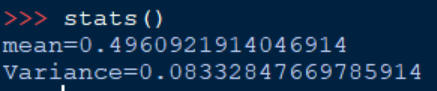
This results in:

|  |  |
| --- | --- |
| mean = 0.4508 | Var = 0.0718 |

The difference in these two sets of parameters can be easily explained by the law of large numbers that states that the actual mean and variance will approach the theoretical mean and variance as the set of RN’s grows larger. I will demonstrate this by changing the original lcg function in the Python code to generate 1,000,000 RN’s instead of thirty, and then re-compute the mean and variance.



Now, using the stats function we will recompute the mean and variance:



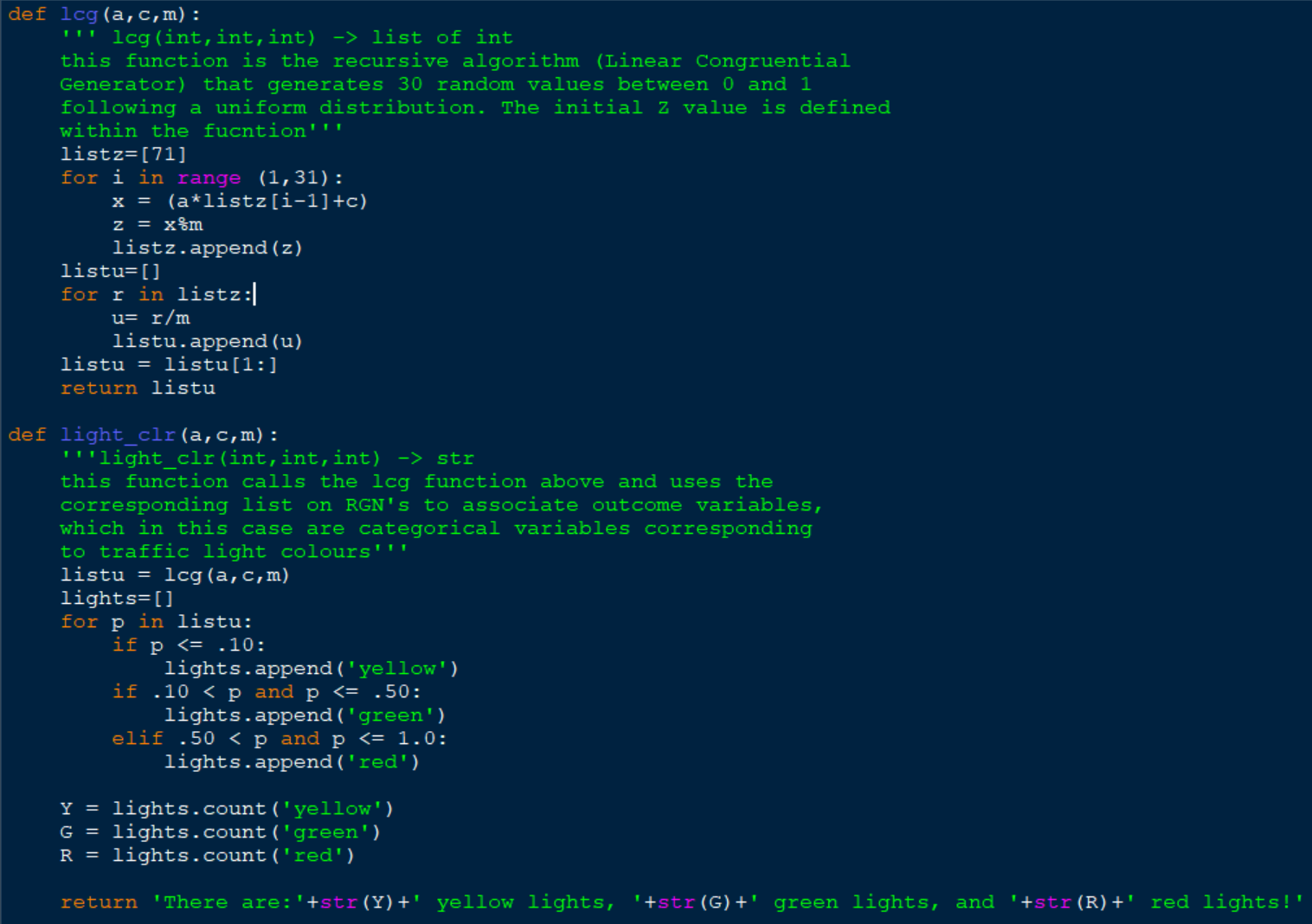
This results in:

|  |  |
| --- | --- |
| mean = 0.496 | Var = 0.0833 |

Now we can see that with 1,000,000 RN’s generated, the mean and variance are very close to their theoretical counterparts.

**b)**

The following Python code calls on the lcg function from part a) and uses the discrete cumulative probability distribution of traffic light colours to assign each RN to a variable (see docstring)



The program output is as follows:



Here is a tabular representation of the assignment of variables:

|  |  |  |
| --- | --- | --- |
| Item Number | Random Number U(0,1) | Corresponding Variable (G=green R=red Y=yellow) |
| 1 | .844 | R |
| 2 | .289 | G |
| 3 | .516 | R |
| 4 | .648 | R |
| 5 | .313 | G |
| 6 | .633 | R |
| 7 | .234 | G |
| 8 | .242 | G |
| 9 | .281 | G |
| 10 | .477 | G |
| 11 | .453 | G |
| 12 | .336 | G |
| 13 | .75 | R |
| 14 | .82 | R |
| 15 | .172 | G |
| 16 | .93 | R |
| 17 | .719 | R |
| 18 | .664 | R |
| 19 | .391 | G |
| 20 | .03 | Y |
| 21 | .188 | G |
| 22 | .008 | Y |
| 23 | .109 | G |
| 24 | .617 | R |
| 25 | .156 | G |
| 26 | .852 | R |
| 27 | .328 | G |
| 28 | .711 | R |
| 29 | .625 | R |
| 30 | .195 | G |

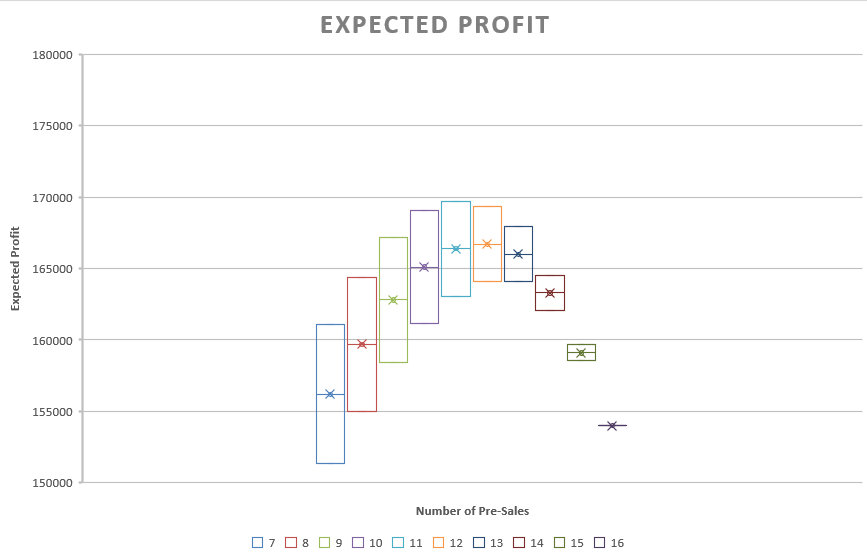
**Part 2**

**a)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Event # | Last Minute Sales | # of occurrences | Mass Probability | Cumulative Probability |
| 7 | 9 | 5 | 0.05 | 0.05 |
| 30 | 9 |
| 38 | 9 |
| 39 | 9 |
| 62 | 9 |
| 8 | 10 | 10 | 0.1 | 0.15  **\*\*Methodology\*\***  1- sort list of sales  2- # of occurrences of each event/ 100 (Mass Pr)  3- accumulate each probability (Cumulative Pr)  4- RN 🡪 RVG (see part b)) |
| 11 | 10 |
| 23 | 10 |
| 25 | 10 |
| 54 | 10 |
| 68 | 10 |
| 72 | 10 |
| 89 | 10 |
| 92 | 10 |
| 93 | 10 |
| 1 | 11 | 15 | 0.15 | 0.3 |
| 9 | 11 |
| 10 | 11 |
| 16 | 11 |
| 24 | 11 |
| 26 | 11 |
| 51 | 11 |
| 53 | 11 |
| 64 | 11 |
| 76 | 11 |
| 79 | 11 |
| 82 | 11 |
| 90 | 11 |
| 94 | 11 |
| 97 | 11 |
| 2 | 12 | 20 | 0.2 | 0.5 |
| 6 | 12 |
| 18 | 12 |
| 19 | 12 |
| 22 | 12 |
| 32 | 12 |
| 44 | 12 |
| 48 | 12 |
| 52 | 12 |
| 58 | 12 |
| 59 | 12 |
| 60 | 12 |
| 65 | 12 |
| 70 | 12 |
| 78 | 12 |
| 80 | 12 |
| 83 | 12 |
| 85 | 12 |
| 95 | 12 |
| 96 | 12 |
| 13 | 13 | 10 | 0.1 | 0.6 |
| 29 | 13 |
| 40 | 13 |
| 42 | 13 |
| 50 | 13 |
| 56 | 13 |
| 57 | 13 |
| 69 | 13 |
| 71 | 13 |
| 99 | 13 |
| 3 | 14 | 10 | 0.1 | 0.7 |
| 4 | 14 |
| 20 | 14 |
| 46 | 14 |
| 47 | 14 |
| 66 | 14 |
| 67 | 14 |
| 74 | 14 |
| 87 | 14 |
| 91 | 14 |
| 5 | 15 | 10 | 0.1 | 0.8 |
| 14 | 15 |
| 31 | 15 |
| 35 | 15 |
| 37 | 15 |
| 45 | 15 |
| 49 | 15 |
| 55 | 15 |
| 98 | 15 |
| 100 | 15 |
| 12 | 16 | 10 | 0.1 | 0.9 |
| 15 | 16 |
| 21 | 16 |
| 28 | 16 |
| 33 | 16 |
| 36 | 16 |
| 75 | 16 |
| 77 | 16 |
| 86 | 16 |
| 88 | 16 |
| 27 | 17 | 5 | 0.05 | 0.95 |
| 41 | 17 |
| 43 | 17 |
| 63 | 17 |
| 81 | 17 |
| 17 | 18 | 5 | 0.05 | 1.00 |
| 34 | 18 |
| 61 | 18 |
| 73 | 18 |
| 84 | 18 |
|  |  |  | 1.00 |  |

**b)**

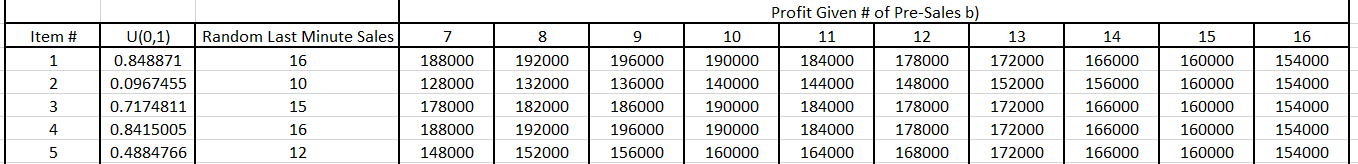
The maximum average expected profit derived from 100 iterations of Monte Carlo is determined to be obtained by pre-selling 12 advertising slots and leaving the rest for last-minute sales.



**x̅ = $168200.00**

**95% Profit C.I. = (170479.95,165920.05)**

Excel Analysis (Monte-Carlo Model):





=IF($I3+J$2<=25,J$2\*4000+$I3\*10000,IF(AND($I3+J$2>25,J$2>$I3),J$2\*4000+(25-J$2)\*10000,IF(AND($I3+J$2>25,J$2<=$I3),J$2\*4000+($I3-(($I3+J$2)-25))\*10000,0)))

This function calculates the profit derived from the pre-determined level of pre-sales and randomly generated number of last-minute sales

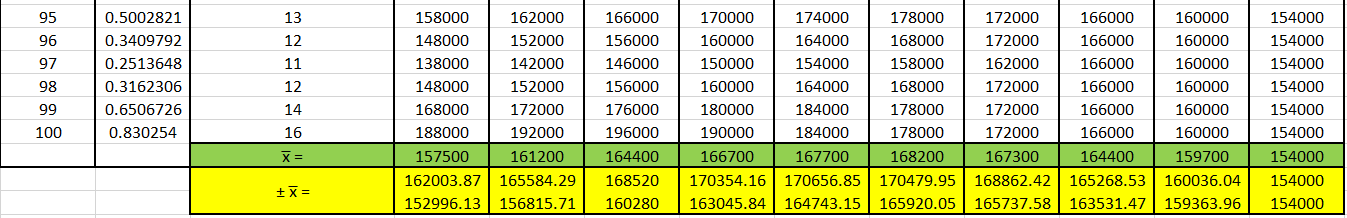
Rand()

Generates U(0,1) RN’s

=IF(H3<=$E$2,9,IF(AND(H3>$E$2,H3<=$E$7),10,IF(AND(H3>$E$7,H3<=$E$17),11,IF(AND(H3>$E$17,H3<=$E$32),12,IF(AND(H3>$E$32,H3<=$E$52),13,IF(AND(H3>$E$52,H3<=$E$62),14,IF(AND(H3>$E$62,H3<=$E$72),15,IF(AND(H3>$E$72,H3<=$E$82),16,IF(AND(H3>$E$82,H3<=$E$92),17,IF(AND(H3>$E$92,H3<=$E$97),18,0))))))))))

This function depicts nested IF functions that guide the program to associate U(0,1) RN’s corresponding with Variables based on the cumulative probability dist. calculated in part a).





=J$103+1.96\*((STDEV.S(J$3:J$102))/(SQRT(100)))

=J$103-1.96\*((STDEV.S(J$3:J$102))/(SQRT(100)))

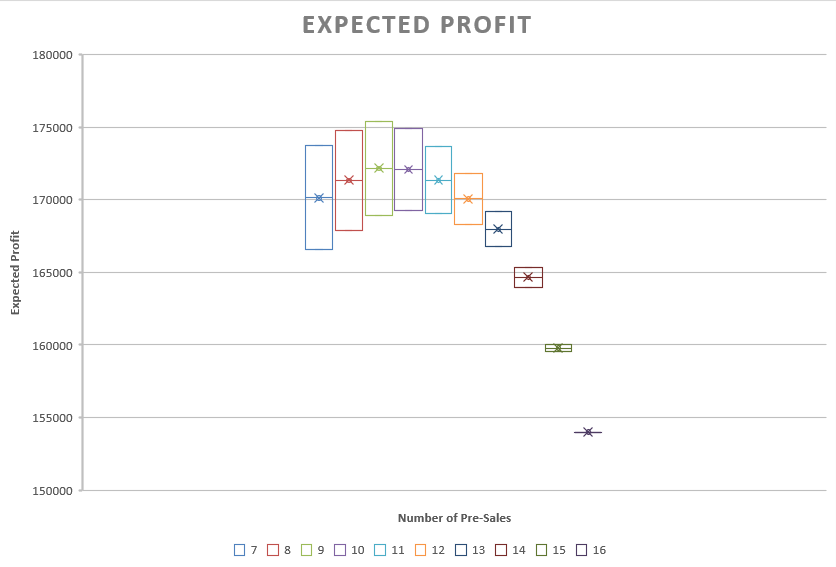
Upper and lower bound of 95% C.I. of point estimate (x-bar)

=SUM(J3:J102)/100

Calculates x-bar of each pre-sale value

**c)**

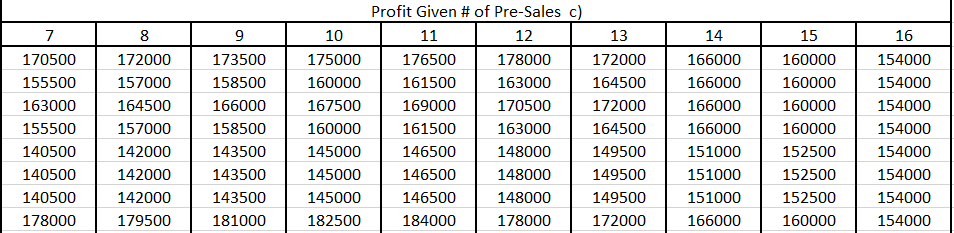
The maximum average expected profit derived from 100 iterations of Monte Carlo is determined to be obtained by pre-selling 10 advertising slots and leaving the rest for last-minute sales.



**x̅ = $171325.00**

**95% Profit C.I. = (173639.35,169010.65)**

Excel Analysis (Monte-Carlo Model):

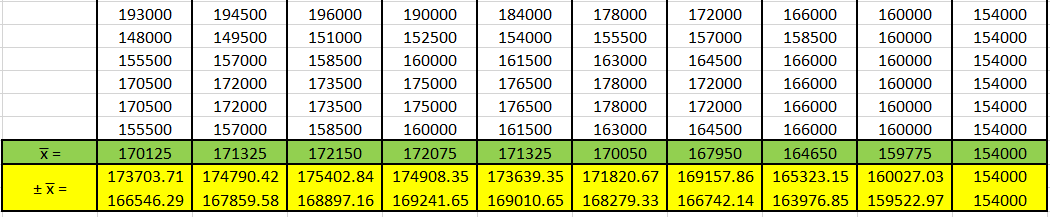




=IF($I3+W$2<=25,W$2\*4000+$I3\*10000+(25-(W$2+$I3))\*2500,IF(AND($I3+W$2>25,W$2>$I3),W$2\*4000+(25-W$2)\*10000,IF(AND($I3+W$2>25,W$2<=$I3),W$2\*4000+($I3-(($I3+W$2)-25))\*10000,0)))

This function calculates the profit derived from the pre-determined level of pre-sales and randomly generated number of last-minute sales, as well as, the number of salvage slots. (Using same Random sales as shown in part b))





=v$103+1.96\*((STDEV.S(v$3:v$102))/(SQRT(100)))

=v$103-1.96\*((STDEV.S(v$3:v$102))/(SQRT(100)))

Upper and lower bound of 95% C.I. of point estimate (x-bar)

=SUM(v3:v102)/100

Calculates x-bar of each pre-sale vlaue

**Personal Ethics Statement**

**Individual Assignment:**

By signing this Statement, I am attesting to the fact that I have reviewed the entirety of my attached work and that I have applied all the appropriate rules of quotation and referencing in use at the Telfer School of Management at the University of Ottawa, as well as adhered to the fraud policies outlined in the Academic Regulations in the University’s Undergraduate Studies Calendar. Academic Fraud Webpage

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018-10-05\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature Date

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 8210704\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Last Name (print), First Name (print) Student Number

**Managerial Recommendations:**

If there is no salvage value for unused advertising slots; it is recommended that 12 slots are presold, and the rest are left for last-minute sales. This will return the highest average profit and variability doesn’t need to be considered.

If there is a salvage value of 2500 dollars for unused advertising slots; it is recommended that 10 slots are presold, and the rest are left for last-minute sales and salvage. This will return the highest average profit and variability doesn’t need to be considered.