

# Monte Carlo method applied to Samsung prices to determine monthly profit

Marco Antonio de la Cruz

December 11th, 2021

## 1 Important

This document, as well as the information, calculations and procedures are confidential. If you are not authorized to read or possess this document, please notify your supervisor, Supply Chain Mexico management or Disposition regional management immediately for instructions on how to proceed with this document.

Please do not share this document without the consent of Supply Chain Mexico management, Disposition regional management, or the creator of this document.

## 2 Considerations

Due to the behavior of received Devices from the trade in program, it is believed that we will receive five hundred devices per month, however, in order for the simulation to consider scenarios where sales are slightly lower than planned, it is proposed two hundred simulations where a random number of pieces of the beta function  $X \sim Beta(\alpha = 4, \beta = 5)$  with location 400 and scale of 200 (figure 1). This distribution has the characteristic of having a mean ( $\bar{X}$ ) of 488.03, a median ( $\tilde{X}$ ) of 488.89 and a standard variation ( $\sigma$ ) of 31.43.

Figure 1: Beta Function for received devices.

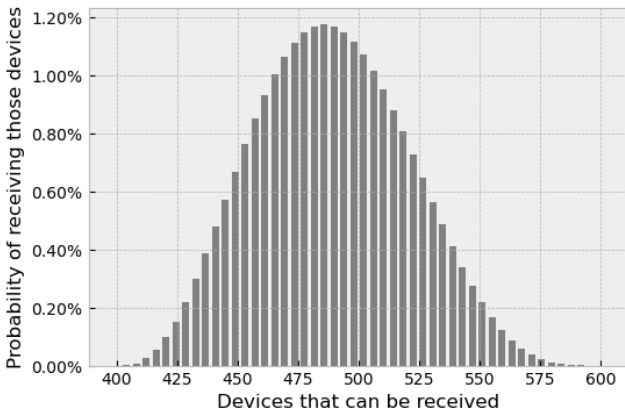


Table 1: Probability of receiving different grade devices.

Grading	Prob. of being received
Grade 1	80.00%
Grade 2	15.24%
Grade 3	4.76%

Analytics has determined that the equipment reception will have a distribution as indicated in table 1. For simulation issues, each received equipment will generate a semi-random number between 0 and 1 to simulate the grade of the device received, only in the event that a unit does not have a price available for grade 3, the price of grade 2 will be taken.

Due to the above, a file containing the probabilities of each device of being acquired by the trade-in program has been restructured, separated into three main columns: Model, grading and probability. The probability has been calculated as follows.

$$p_{model-grade} = p_{model} \cdot p_{grading}$$

An example of the above is the **Galaxy Note 10** model, which in the model mix has a probability of receiving it of 0.6%, this amount would be multiplied by each of the probabilities of each grade of being received (table 1).

$$p_{Note10-Grade1} = 0.6\% \cdot 80.00\% = 0.4800\%$$

$$p_{Note10-Grade2} = 0.6\% \cdot 15.24\% = 0.0914\%$$

$$p_{Note10-Grade3} = 0.6\% \cdot 4.76\% = 0.0286\%$$

With the above we can affirm the following, where:

- $p_{i\alpha}$  is the probability of each model of grade 1 to being traded.
- $p_{i\beta}$  is the probability of each model of grade 2 to being traded.

Table 2: Description in Probabilities.xlsx.

Column name	Description
Device name	Model name.
Model Mix	BetProbability of receiving this model in the trade-in program.
Grade	Model grade.
Probability	Probability of receiving this model with the corresponding grading in the trade-in program.

Table 3: Description in Buying price.xlsx.

Column name	Description
Device name	Model name.
Grading	Model grade.
Buying price	Acquisition price of the model and corresponding grading.

- $p_{i\gamma}$  is the probability of each model of grade 3 to being traded.
- $n$  is the quantity of models on the list.

$$\sum_{i=1}^n (p_{i\alpha} + p_{i\beta} + p_{i\gamma}) = 1.00$$

Since three files have been manipulated, these will be in the annexes, only details of what is contained in each column will be given.

**Probabilities.xlsx** (table 2) indicates the probability that a device will be acquired in the trade-in program by model and grading.

**Buying price.xlsx** (table 3) indicates the purchase prices by model and grading.

**Aggregators prices.xlsx** (table 4) indicates the prices at which the equipment will be sold, in this specific case it will be considered that 60% of the collected equipment will be sold to the best option, 30% to the second highest bidder and 10% to the highest bidder.

### 3 Explanation how Monte Carlo method works

This is a non-deterministic method and it is used so that through probabilistic values the unpredictable behavior of reality is simulated with the generation of semi-random numbers.

Table 4: Description in Buying price.xlsx.

Column name	Description
Device name	Model name.
Grade	Model grade.
Selling price 1	Maximum sale price to aggregators for the model and grading indicated.
Selling price 2	Second highest selling price to aggregators for the model and grading indicated.
Selling price 3	Third highest selling price to aggregators for the designated model and grading.

Although it is true that a greater number of simulations should be more similar to the calculations obtained by deterministic methods, it is precisely with this behavior that extra data such as variability can be obtained and behavior ranges obtained.

For this specific exercise, three simulations will be used in parallel in two hundred different tests. The first will indicate how many pieces will be received, as previously mentioned a beta function will be used, this first number will indicate how many times the second simulation will be executed. The second simulation will determine which device has been received considering a continuous semi-random number between 0 and 1, and with respect to the accumulated function of the different models it will indicate which model and grading it corresponds to.

The third simulation determines to whom the equipment will be sold, the option that pays the best for the device will have a probability of 60%, the second best option will have a probability of 30% and the third option a 10%.

An arithmetic operation will be made of the sale price of each equipment minus the acquisition price and the total will be recorded in a table.

Finally, all the results of the table will be taken and we will be able to determine the descriptive statistics numbers and know the scope of the program under said quantities.

In order to be able to replicate the analysis on any computer with the same software, seeds will be used in the methods and functions that require random numbers.

## 4 Analysis using Python

The exercises in this document have been done in Python 3, with the following packages:

- Pandas, package for data manipulation and analysis.
- Numpy, package for multi-dimensional arrays and high-level mathematical functions.
- Matplotlib, package for embedding plots.
- Scipy, package for linear algebra, integrations and interpolation.
- Random, package for generate random numbers.

### 4.1 Step 1

Import all the packages that are required for the process, and import the CSV files.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import ticker
import scipy.stats as st

df_prob = pd.read_excel('
    Probabilities.xlsx', sheet_name='
    Hoja1')
df_buy = pd.read_excel('Buying price.
    xlsx', sheet_name='Hoja1')
df_sell = pd.read_excel('Aggregators
    prices.xlsx', sheet_name='Hoja1')
```

### 4.2 Step 2

Determinate the quantities of simulations and the probability of each device to be sold to the option one, two and three.

```
simulations = 200
sell_1 = 0.60
sell_2 = 0.30
sell_3 = 0.10
```

### 4.3 Step 3

Create two empty list, the first one will contain the result of each simulation and the second one will contain the results for each piece per simulation.

```
subtotal= pd.DataFrame({
    'Pieces': np.zeros(simulations),
    'Buying': np.zeros(simulations),
    'Selling': np.zeros(simulations),
    'Margin': np.zeros(simulations)
})
subsubtotal = pd.DataFrame({
    'Model': np.zeros(700),
    'Buying': np.zeros(700),
    'Selling': np.zeros(700),
    'Margin': np.zeros(700)
})
```

### 4.4 Step 4

Join "df\_buy" and "df\_sell" in a dataset, so it's necessary both datasets have a similar column.

```
df_buy['union']= df_buy['Device name'
    ] + " " + df_buy['Grading']
df_sell['union']= df_sell['Model'] +
    " " + df_sell['Grade']
df = pd.merge(df_buy, df_sell, how='
    left', on='union')
```

### 4.5 Step 5

Define the first simulation: how many devices we will receive using a Beta distribution  $X \sim B(\alpha = 4, \beta = 5)$

```
for i in range(0, len(subtotal)):
    np.random.seed(df_prob.index[i
    ]*100)
    subtotal['Pieces'].loc[i] = np.
    round(st.beta(4, 5, loc=400, scale
    =200).rvs(),

        decimals = 0)
```

### 4.6 Step 6

Simulate models and grades we will receive in two hundred simulations. To determinate the model and grading we use the dataframe created on step 4.

This step is the most important, we will create two loops, and it is necessary to reset the information between two simulations.

```
for i in range(0, len(subtotal)):
    print(i / len(subtotal)*100)
```

```

subsubtotal['Model'] = np.zeros
(700) #Reset info
subsubtotal['Buying'] = np.zeros
(700)
subsubtotal['Selling'] = np.zeros
(700)
subsubtotal['Margin'] = np.zeros
(700)
for j in range(0, subtotal['
Pieces'].loc[i].astype(int)):
    np.random.seed(j*20)
    modelindex = np.random.choice
(df_prob.index, 1, p=df_prob['
Probability']) #Indica el index
del modelo
    subsubtotal['Model'].loc[j] =
    str(df_prob['Device name'].loc[
modelindex[0]] +

    " " + df_prob['Grade'].loc[
modelindex[0]]) #Indicate model
and grading with index
    subsubtotal['Buying'].loc[j]
= float(df[df['union'] ==
subsubtotal['Model'].loc[j]]['
Buying price']) #Usa filtro para
buscar el precio de compra
    np.random.seed(j*20)
    to_who = np.random.choice([1,
2, 3], 1, p=[sell_1, sell_2,
sell_3])
    if to_who == 1:
        subsubtotal['Selling'].
loc[j] = float(df[df['union'] ==
subsubtotal['Model'].loc[j]]['
Selling price 1'])
    elif to_who == 2:
        subsubtotal['Selling'].
loc[j] = float(df[df['union'] ==
subsubtotal['Model'].loc[j]]['
Selling price 2'])
    else:
        subsubtotal['Selling'].
loc[j] = float(df[df['union'] ==
subsubtotal['Model'].loc[j]]['
Selling price 2'])
    subsubtotal['Margin'] =
subsubtotal['Selling'] -
subsubtotal['Buying'] #Venta menos
compra
    subtotal['Buying'].loc[i] = sum(
subsubtotal['Buying'])
    subtotal['Selling'].loc[i] = sum(
subsubtotal['Selling'])
    subtotal['Margin'].loc[i] = sum(
subsubtotal['Margin'])
    decimals = 0)

```

Table 5: Description of results.

Measure	Pieces	Buying	Selling	Margin
Count	200	200	200	200
Mean	488.16	2 764 982	3 813 408	1 048 426
SD	28.59	195 267	239 947	88 971
Min	417	1 578 175	3 120 001	802 386
Perc. 25	466	2 655 347	3 653 859	986 775
Perc. 50	487	2 771 968	3 804 898	1 045 143
Perc. 75	506	2 874 776	3 959 612	1 100 156
Max	564	3 267 891	4 356 019	1 602 506

#### 4.7 Step 7

Get the statistical result of the simulation within a single code line.

```
subtotal.describe()
```

## 5 Results

Step 7 has shared with us the statistical resume, in this case will be four different information:

- Simulated received pieces
- Simulated buying amount.
- Simulated selling amount.
- Difference in selling and buying amount per simulation.

The result of the simulation is found in table 5, in which we can see the behavior of the four previous points.

The resulting quantity of each simulation is seen in the figure 2.

With the information that has been obtained, different estimates can be made, the first of which is to view the results in histograms (figure 3) that allow us to see where most of the results obtained are concentrated.

For this exercise we will use the statistical method to estimate the overall shape of a random variable distribution with the Kernel Density Estimation function, which will allow us to analyse it as a Gaussian distribution and determine confidence intervals.

The mathematical formula to find this series is:

$$f(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

However, the Scipy library is advised to save time and improve accuracy.

To determine the confidence intervals with a 95 % probability that the mean is in that interval, you find the formula.

$$CI = \bar{x} \pm Z\left(\frac{s}{\sqrt{n}}\right)$$

The data will be taken from the sampling (simulation). In figure 4 we can see the KDE function (smoothed distribution of the histogram), in addition a normal distribution can be observed considering the mean ( $\bar{x}$ ) of each data and its standard deviation ( $s$ ). Shaded in gray the confidence interval where the real mean is found.

## 6 Conclusions.

With the information described above we can ensure that under the conditions of pieces received from a beta function  $X \sim Beta(\alpha = 4, \beta = 5)$ , with the mix shared by Analytics team and with 60% of devices sold to the best buyer, 30% to the second best buyer and 10% to the third best buyer, under a semi-random scheme the number of received devices has an average with 95% confidence of being between 484.17 and 492.14. In the case of the buying amount of the devices, it is between MXN 2 737 754.58 and MXN 2 792 209.97. The sale amount is between MXN 3 779 950.38 and MXN 3 846 866.00. Finally, the average margin for the month will be between MXN 1 036 020.00 and 1 060 831.83.

If the nature of Samsung's trade-in program is similar to the information described in the input parameters, I can confirm that the results in terms of earnings are positive.

There is the possibility that one or more input parameters are different in reality because it is a new program and there is not enough experience to determine the input values, as well as an atypical behavior of the market, however, they have been taken several interconnected factors and considered a tiny downward trend of devices to receive to study the effects of sales prices in a case of a weak launch and check if this agrees with Assurant's numbers. resulting in positive numbers, even if said launch does not have the expected effect on the first approaches with the client.

Figure 2: Simulation results

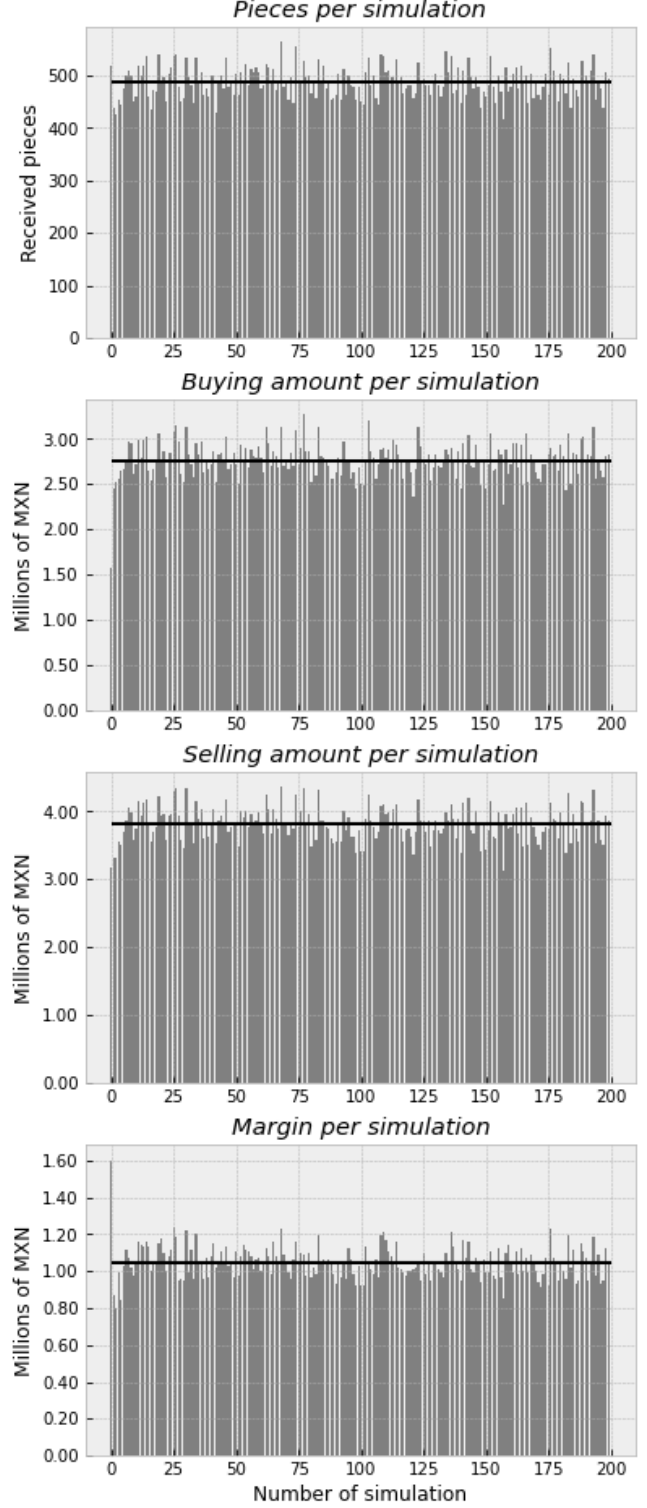


Figure 3: Histogram of the simulation results.

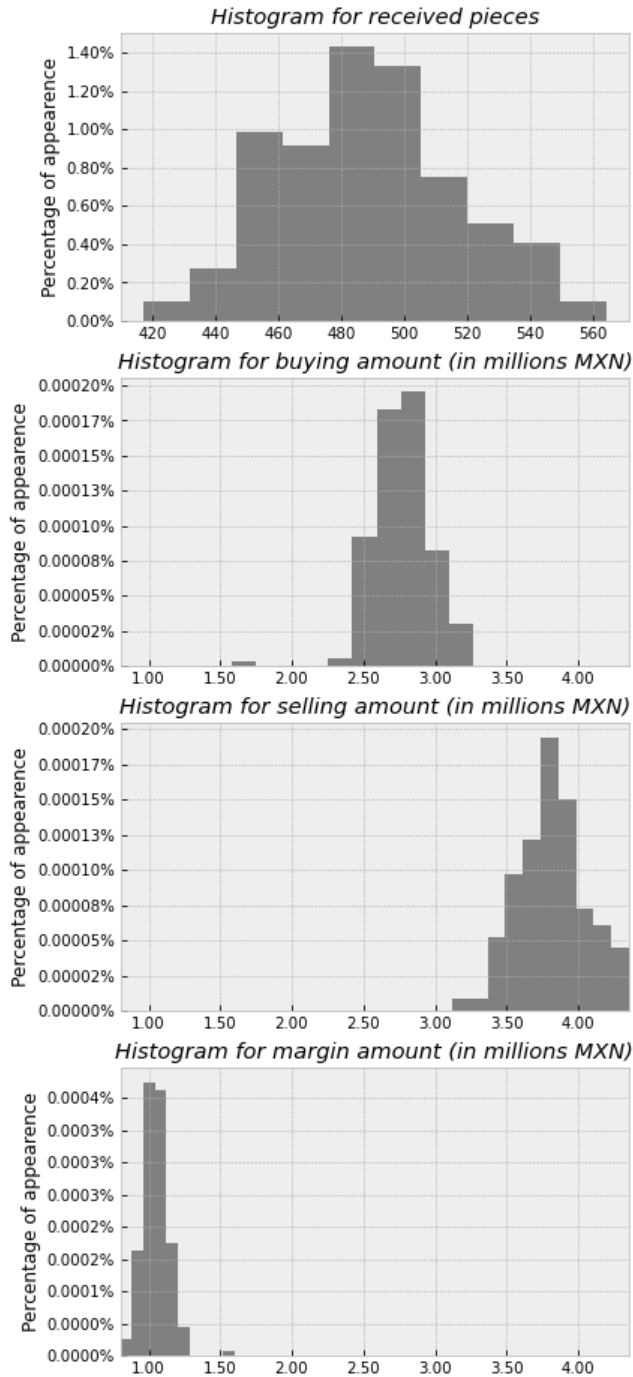


Figure 4: KDE Density of the simulation results.

