# ECSA Final Report 2025

## Quality Assurance 344

Dirk-Louw van der Westhuizen
25929372

# Introduction

The purpose of this report is to demonstrate proficiency in data analysis, statistical process control (SPC), and applied problem-solving in an industrial engineering setting. It is a component of the QA344 ECSA GA4 assessment. The project uses R programming to assess process stability, capability, and overall system performance by integrating multiple datasets, mainly product, sales, and service time data. Trends, distributions, and relationships among important variables like sales volume, markup, and delivery times were revealed by the preliminary descriptive data analysis. The creation of $\bar{X}$ and S control charts to track process variation and spot possible out-of-control situations came next, guaranteeing compliance with practical process monitoring guidelines. While Type I and Type II errors were examined in later analyses to gauge the dependability of control choices, process capability indices (Cp and Cpk) were computed to determine whether product delivery times satisfied customer requirements. Lastly, in order to demonstrate how data-driven decision-making can improve operational performance, optimization models were used to balance service efficiency and profitability in simulated retail environments. All things considered, this report shows how statistical tools and analytical reasoning can be applied thoroughly to support engineering reliability and continuous process improvement.

# Part 1.1

## 1. Data Loading and Inspection

The report begins by using the dim, str, head, and colnames tools to visualize an integrated txhousing dataset from the ggplot2 package. In this step, the variables (city, year, month, sales, volume, median, listings, inventory, etc.) and the amount of rows/columns will be guaranteed. In order to organize the rest of the analysis, including which variables are numbers and which are categories, as well as which time fields you wish to plot, it is more useful to have types and sample values accessible at the start of the analysis.

## 2. Summary Statistics

The algorithm then produces two types of descriptive statistics. As medians, means, quartiles, and ranges, the underlying summary() function provides a brief overview of the central tendency and dispersion of the different variables. The deeper descriptors (such as skewness, kurtosis, and standard deviation) are then appended to it using psych:describe. Last but not least, a custom table summary TXH is created using dplyr::summarise() to estrange the mean and standard deviation of the main numerical variables (sales, volume, median, listings, inventory) using na.rm = TRUE in order to prevent missing values from distorting the data. This combination results in both breadth and precision.

## 3. Handling Missing Values

ColSums(is.na(txhousing)) is used to count the number of missing values in each column. The location of NAs (which could be in sales or volume) and the seriousness of the issue will be disclosed by this diagnostic. to complete all additional plots and numerical summaries on NA.To ensure that calculations and visuals don't use a missing entry and fail or skew the results, rm = TRUE is employed. It would also inform about the necessity of imputation or focused filtering if NAs were prevalent in one of the fields.

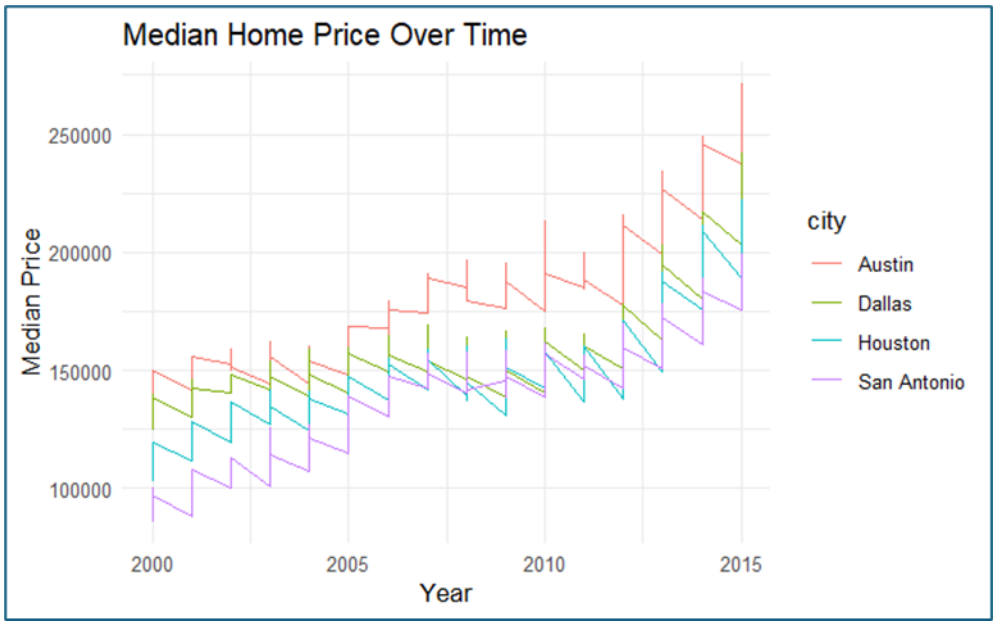
## 4. Data Filtering and Subsetting

Due to the large number of cities that were included in the dataset in Texas, the analysis was limited to four of the biggest housing markets in Texas, which include Austin, Dallas, Houston, and San Antonio. This was subsetting to facilitate making the comparisons more visible and understanding, because at once inclusion of all the cities would have congested the graphs. The

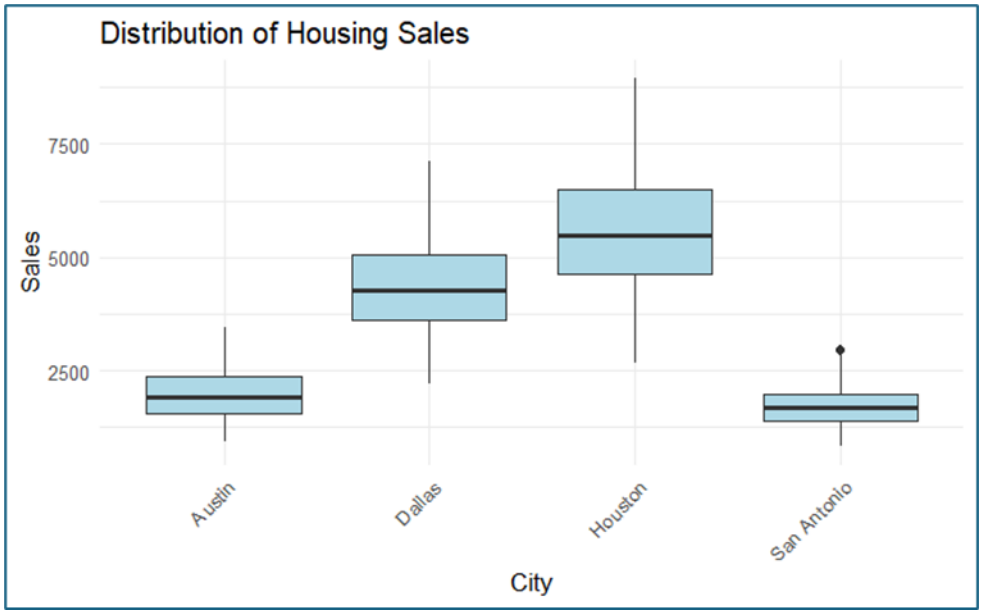
smaller subset was more focused to analyze, whereas the overall averages and the yearly trends were calculated with the full dataset.

## 5. Data visualization

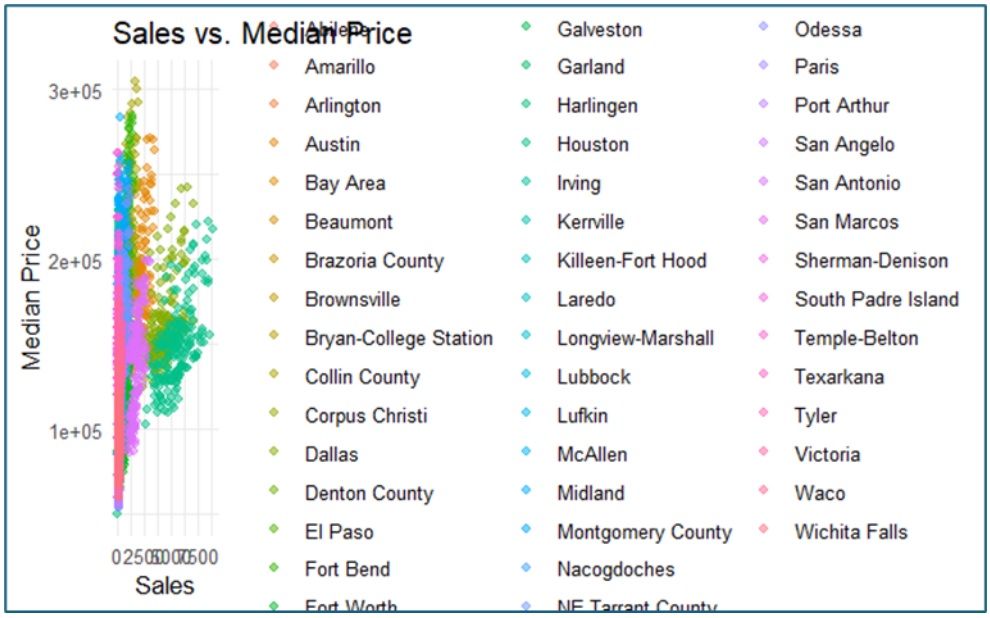
Median Home Price Over Time



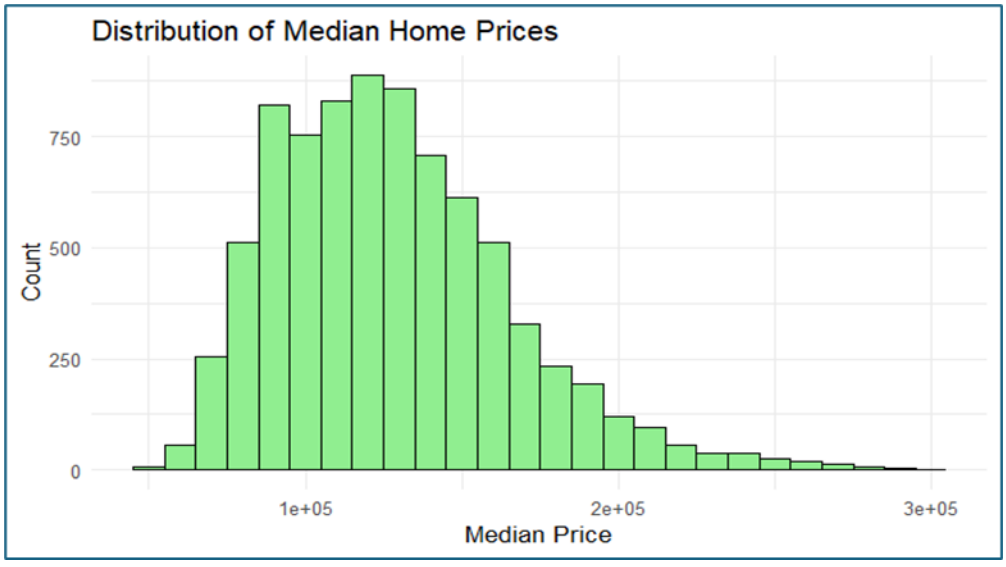

Distribution of housing sales



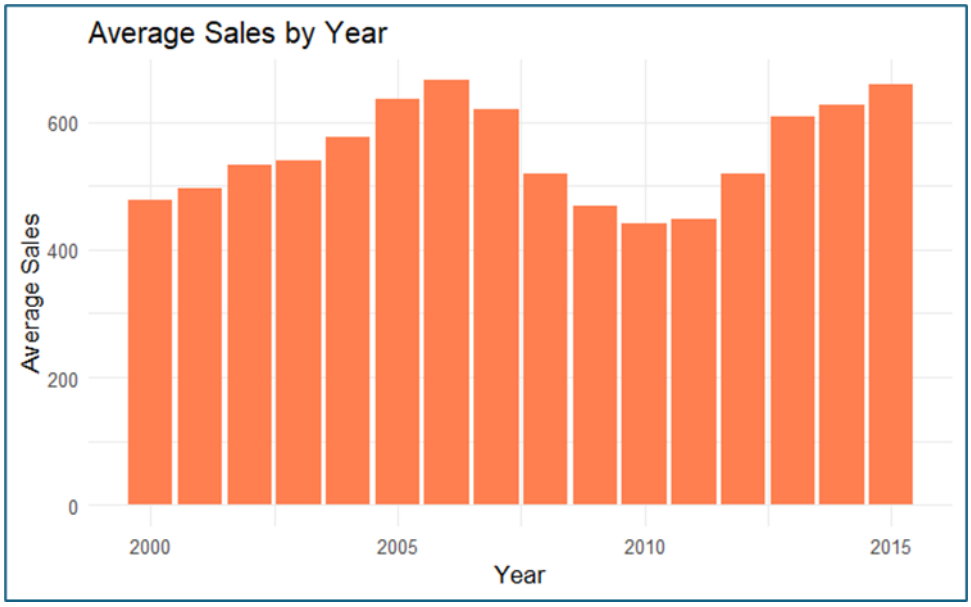
Sales vs Median Price



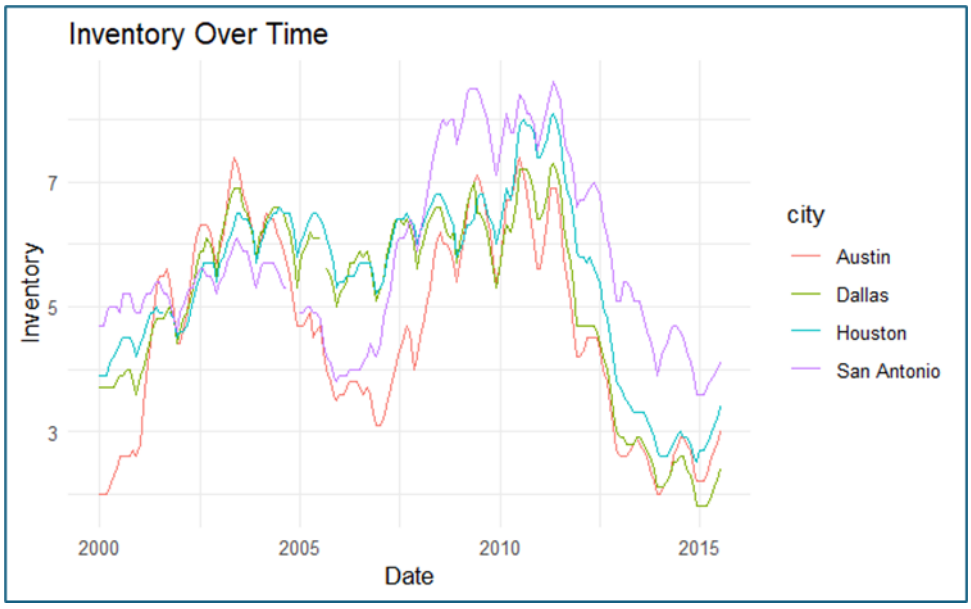

Histogram of Median Prices



Average sales by year



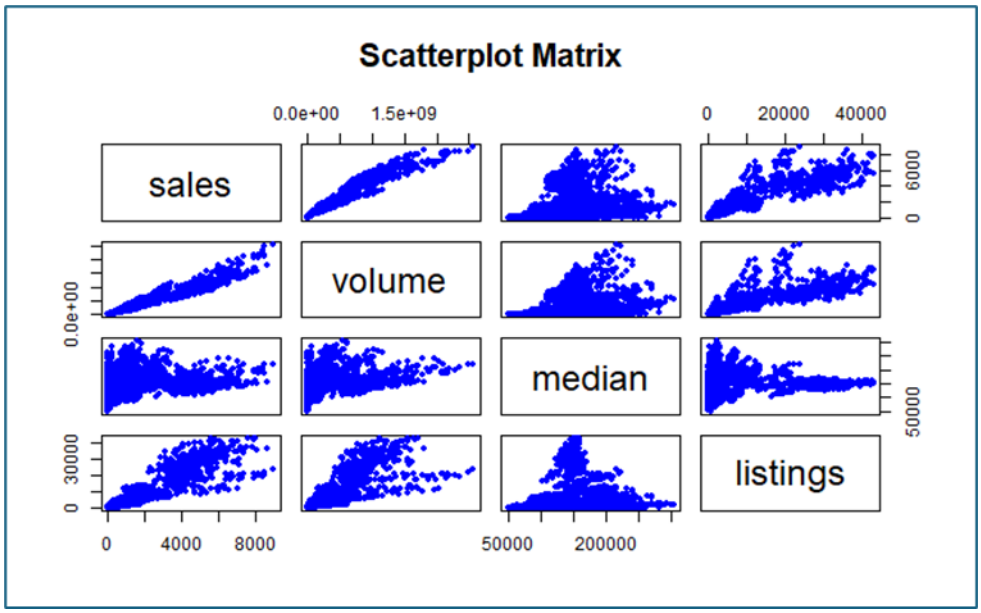

Inventory Over Time by City




A number of visualizations were created to help better understand the housing tendencies. The city of Austin has consistent growth when compared to other cities, according to a linear graph of median prices throughout time. Houston's sales were substantially higher than those of other cities, according to a boxplot analysis of city sales. According to a scatter plot of sales versus median price, cities with higher sales generally had lower median pricing, meaning that big volumes could be afforded. According to a median price

histogram, the majority of median values fell between $100,000 and $200,000, with relatively few exceeding $300,000.

## 6. Exploring Relationships




The association between sales, volume, median pricing, and listing was examined using a scatterplot. The results showed that while there was a weak correlation between sales and median price, there was a large positive correlation between sales and volume. Two conclusions were drawn from the overall analysis. Houston has the largest sales and inventories, making it appealing in terms of market size. Austin, on the other hand, has seen substantial and steady price increases, which suggests strong demand and future possibilities. Therefore, Austin would be the most suitable alternative for the long-term growth prospect, whereas Houston would be the greatest option if the target objective is a big transaction volume in the short term.

# Part 1.2

## 1. Data Loading and Inspection

Using R code to analyze customer, product, and sales data is like removing layers to reveal the meaning behind the numbers. Using read.csv, the code loads the three datasets—customers, products, and sales—smoothly. This gives us a sneak peek using str() to see the structure (such as customer ages as integers or product prices as numerics), dim() to count rows and columns (about 5000 customers, 60 products, and a substantial sales log), and head() to identify the opening entries (such as a 16-year-old male from New York or an expensive laptop). Everything is arranged neatly in the colnames() call, which displays fields like "Gender" and "SellingPrice." Sales data connects the dots with order details like quantities up to 50 and processing times that occasionally exceed 40 hours. This step reveals a diverse customer base with ages ranging from teens to over 100—some of those high ages raise an eyebrow, hinting at possible data entry quirks—and products ranging from inexpensive software to laptops hitting nearly $20,000.

## 2. Summary Statistics

The code then moves on to the next layer, which gives us a good overview with summary() providing min, max, mean, and quartiles (e.g., customer ages averaging 45-50, incomes hovering near $70,000), and describe() from the psychology package adding depth with skewness and standard deviations (incomes likely lean right with a wide spread). Cities like New York and Los Angeles appear frequently, product categories are evenly distributed, and there is a balanced gender split with a dash of "Other," according to the table() function that counts frequencies. This is further refined by custom dplyr summaries, which compute means and standard deviations (e.g., product markups averaging 20% with a range of 10-30%). It is evident that the customers' ages and incomes vary greatly, that the products are diverse in terms of category, with laptops having the highest prices, that sales quantities are generally low but fluctuate, and that processing times indicate sporadic delays.

## 3. Handling Missing Values

For keeping things tidy, the code checks for gaps with colSums(is.na()), which seems to show no major issues based on the clean snippets provided, and na.omit() preps the data by dropping any stray NAs. This keeps the analysis on track without skew from missing data—though if any hid in the truncated parts, they're quietly removed.If any data are hidden in the truncated

sections, they are discreetly removed to prevent the analysis from being skewed by missing data.

## 4. Data Filtering & Subsetting

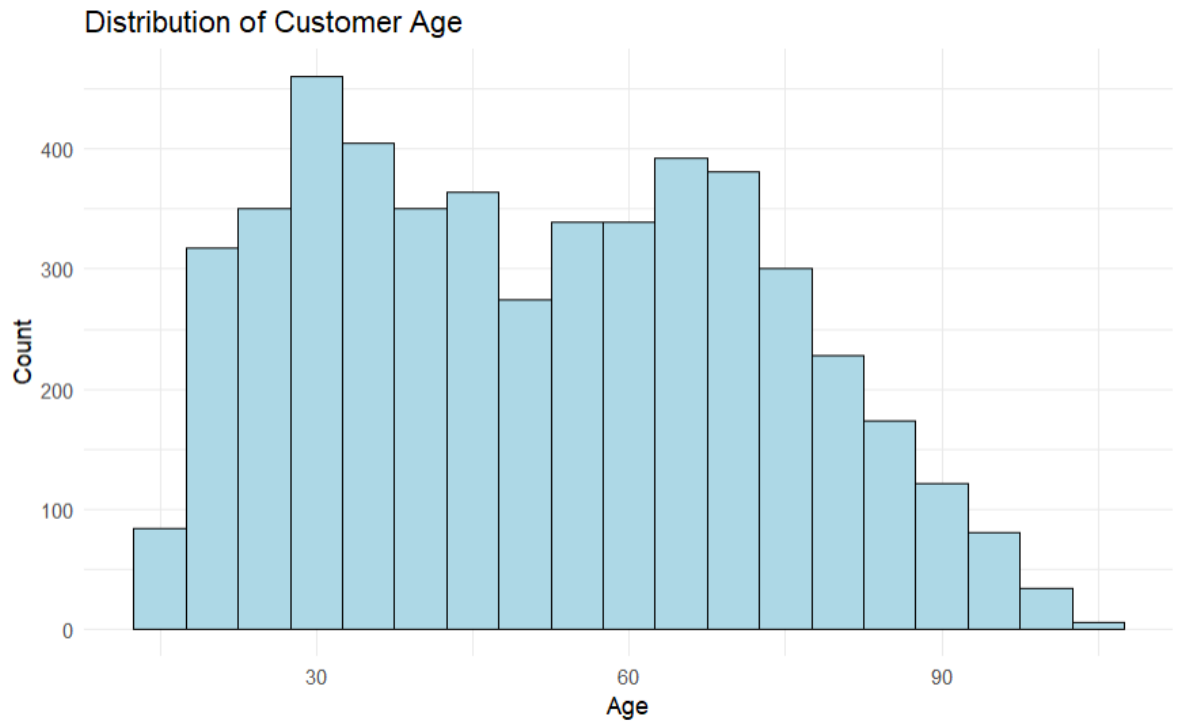
Here, the code excels with a useful twist: filtering sales for 2023 with dplyr::filter(orderYear == 2023) allows us to see trends (e.g., a 2023 order for LAP026). This makes it possible to compare years and suggests possible changes in purchasing habits without becoming overwhelmed by the entire dataset.

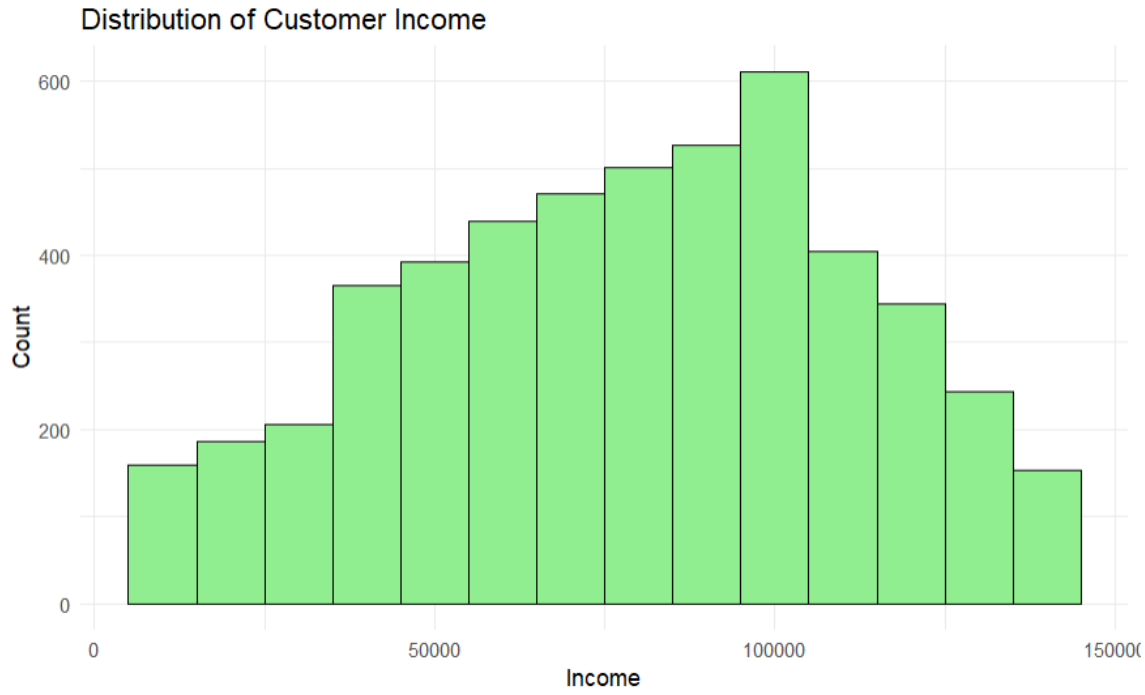
## 5. Data Visualization

The code does its magic when it comes to making the numbers come to life. Distributions are mapped out by histograms: sales quantities taper off at higher values, product prices cluster by category with laptops spiking, incomes skew right with $10,000 bins, and customer ages may exhibit a bimodal curve (teens and seniors). Boxplots highlight the spreads: selling prices by category display laptops with the greatest range, age by city varies (Seattle may lean younger), and income by gender appears similar but may have male outliers. Age versus income in a scatter plot indicates a weak but interesting positive trend. Combining the data, a line plot of quantity by month suggests seasonal bumps, perhaps near year-end, while a bar chart of revenue by category crowns laptops as the revenue kings. These images depict a range of income levels, expensive anomalies, and varying demand.

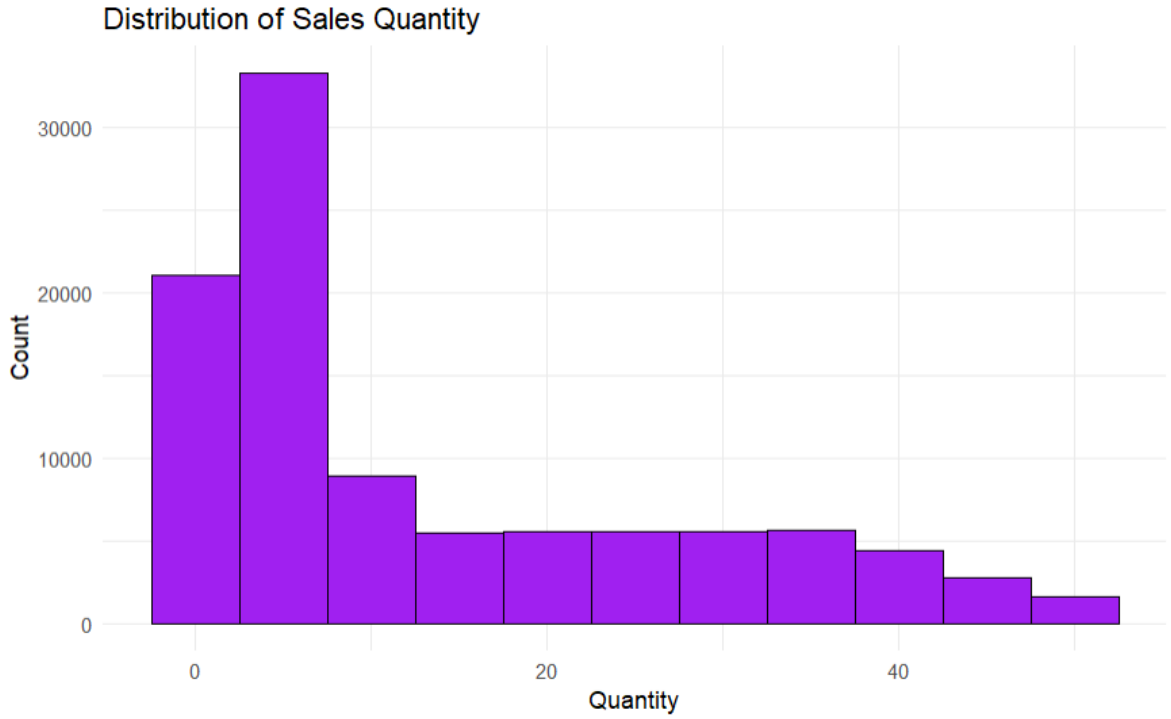
## 6. Exploring Relationships

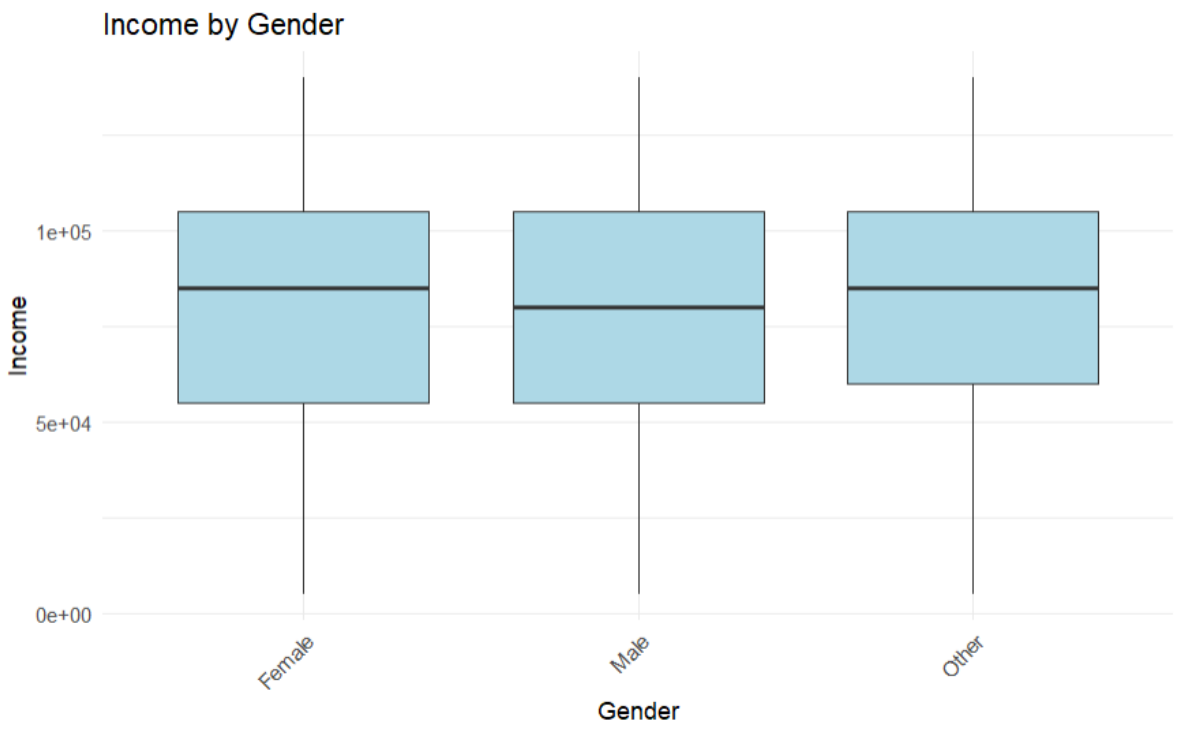
The code then examines relationships using a pairs() scatterplot matrix for age, income, quantity, selling price, and revenue. It finds that there is a clear tie between quantity, price, and revenue, but overall there are weak correlations (for example, age and income don't lock step). With a human touch, the comments summarize the following: monthly sales may follow seasonal patterns, laptops drive cash flow, the customer base is a mix of young and old with a range of budgets, and although the data appears to be accurate, those extreme ages warrant a second look. It's an exploration of the data that leaves us informed and intrigued.

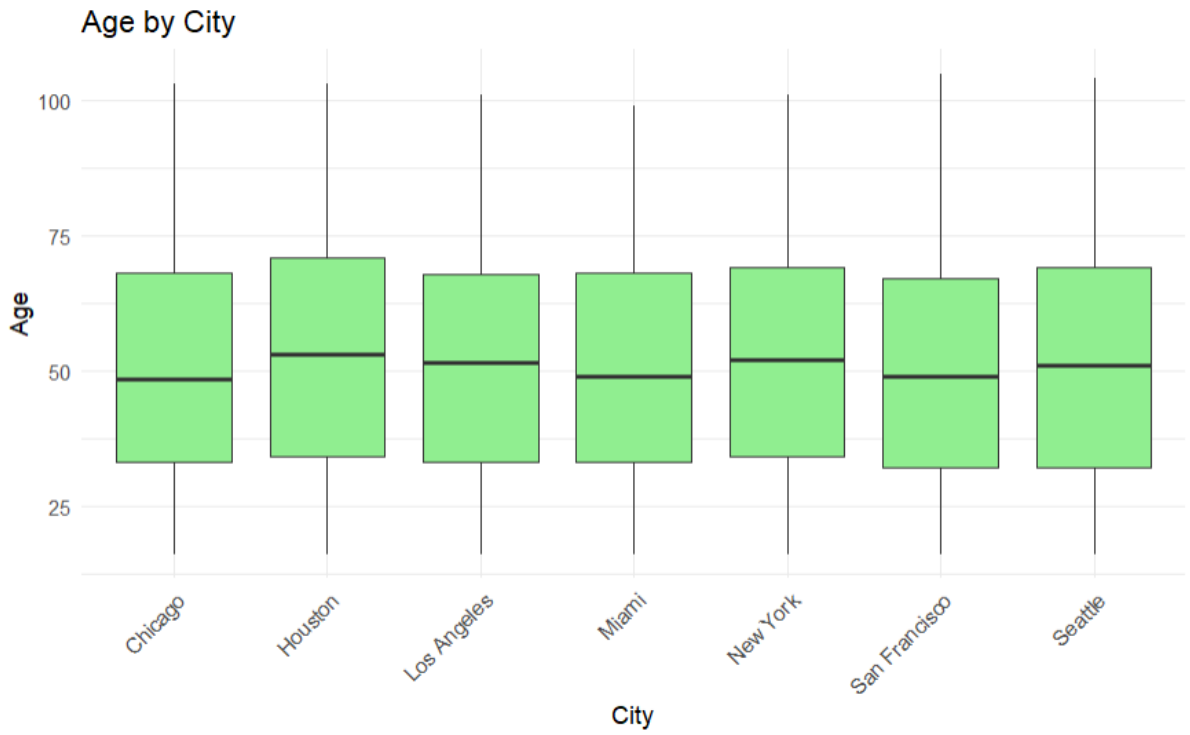

Distribution of Customer Age
Count
400
300
200
100
0
30
60
90
Age

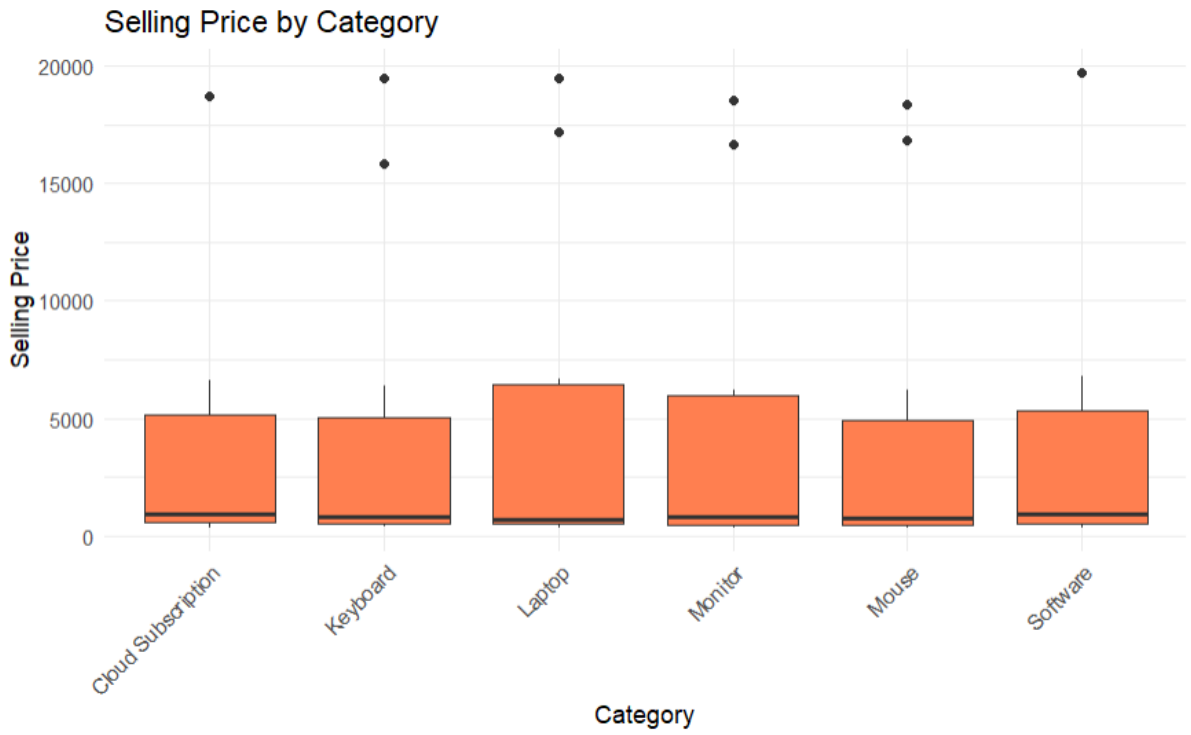

Distribution of Customer Income
Count
600
400
200
0
0
50000
100000
Income


Distribution of Product Selling Price
Count
20
10
0
0
5000
10000
15000
20000
Selling Price

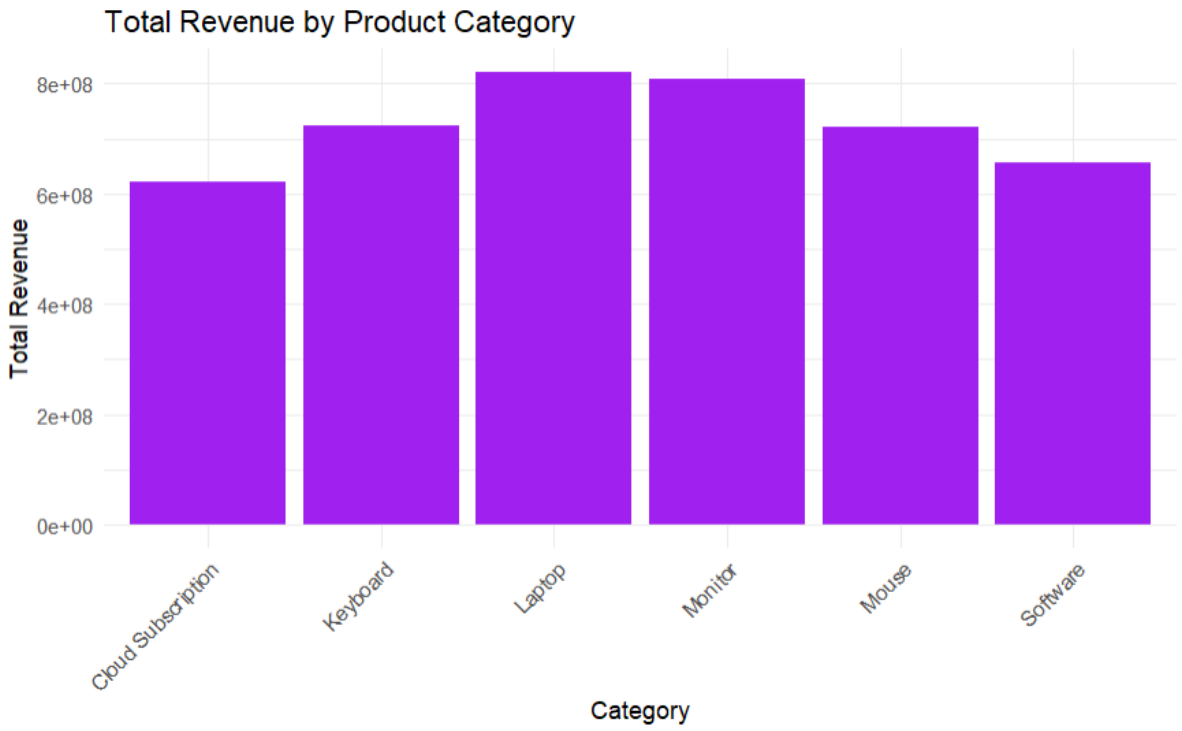

Distribution of Sales Quantity
Count
30000
20000
10000
0
0
20
40
Quantity

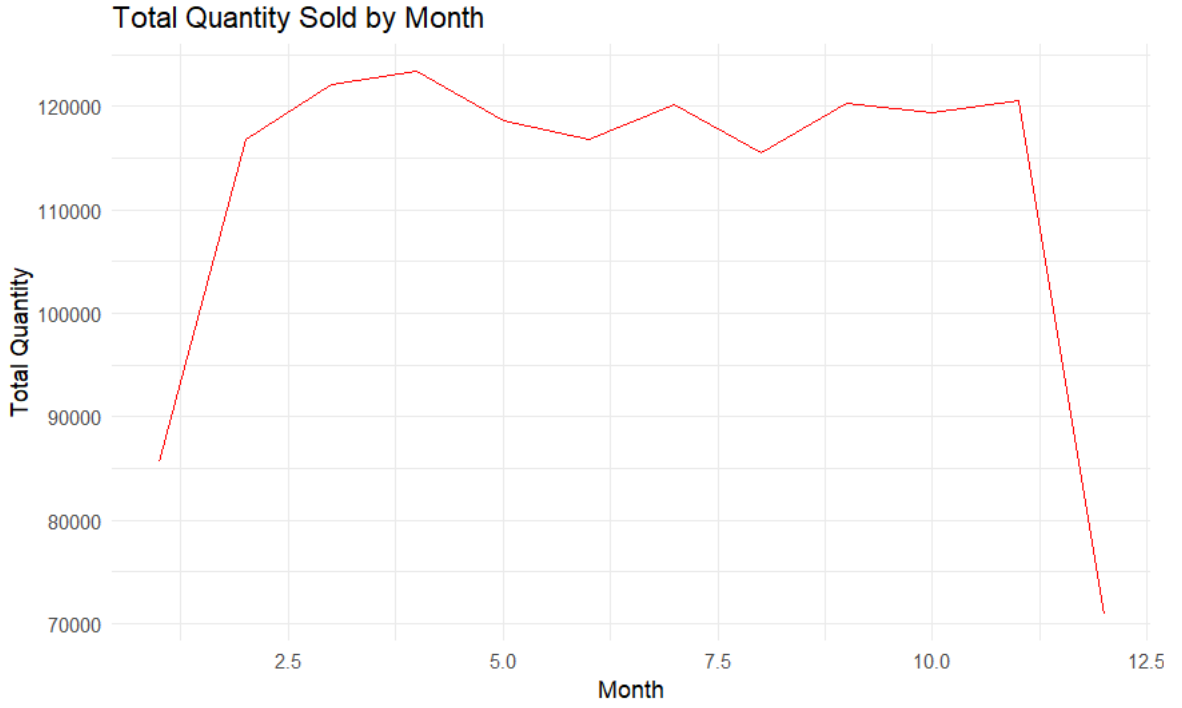

Income by Gender
Income
1e+05
5e+04
0e+00
Female
Male
Other
Gender


Age by City
Age
100
75
50
25
Chicago
Houston
Los Angeles
Miami
New York
San Francisco
Seattle
City


Selling Price by Category
Selling Price
20000
15000
10000
5000
0
Cloud Subscription
Keyboard
Laptop
Monitor
Mouse
Software
Category


Age vs. Income
Income
1e+05
5e+04
0e+00
25
50
75
100
Age

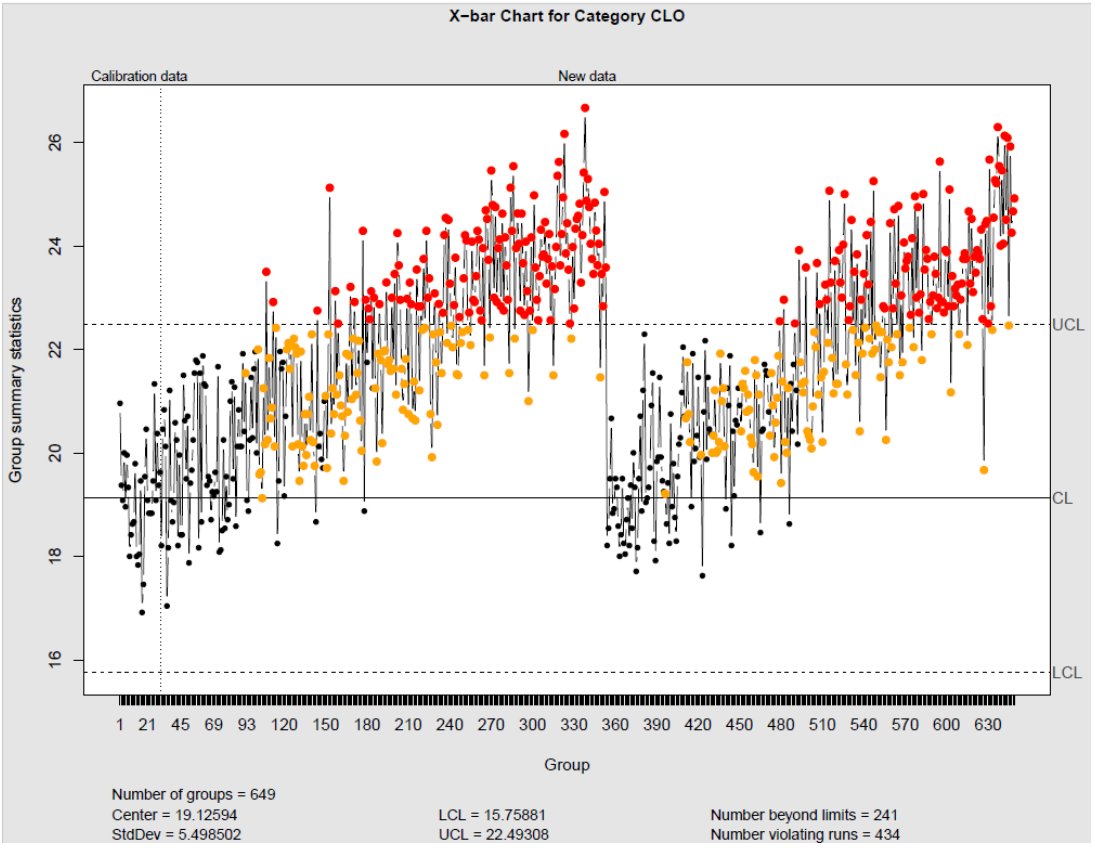

Total Revenue by Product Category
8e+08
6e+08
4e+08
2e+08
0e+00
Total Revenue
Cloud Subscription
Keyboard
Laptop
Monitor
Mouse
Software
Category

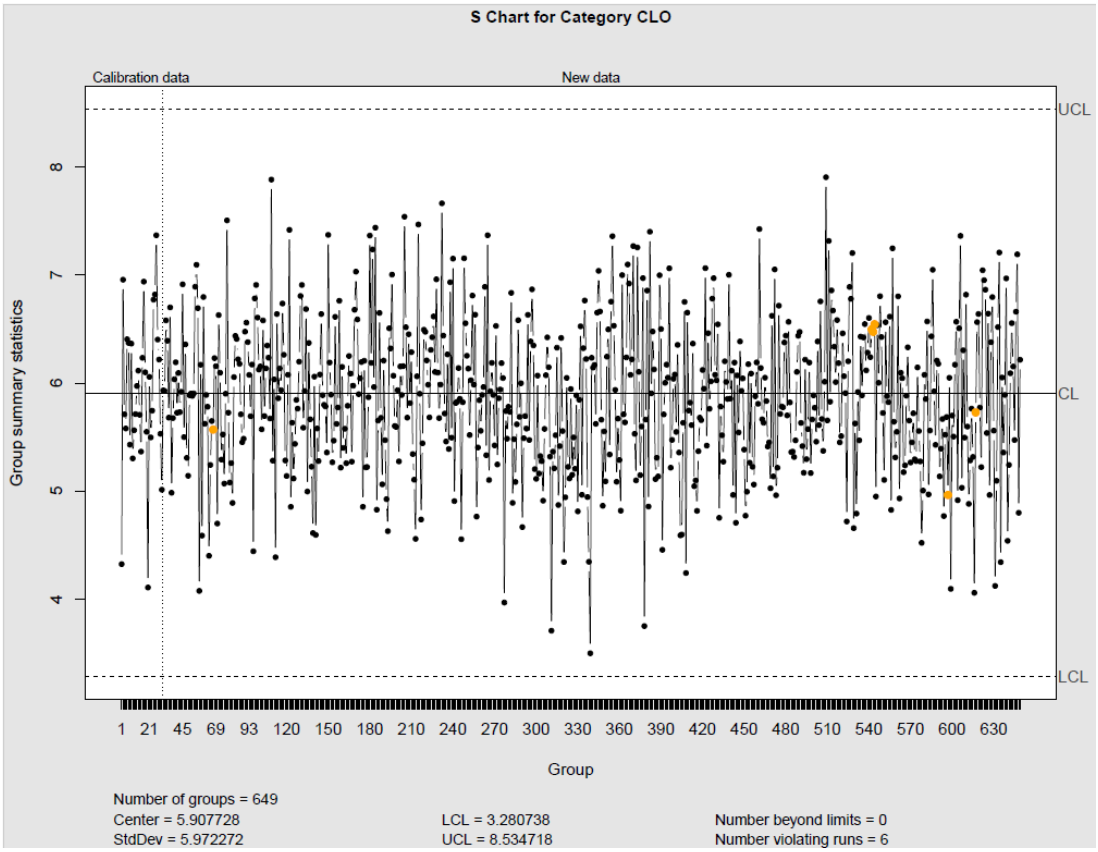

Total Quantity Sold by Month
120000
110000
100000
90000
80000
70000
Total Quantity
2.5
5.0
7.5
10.0
12.5
Month

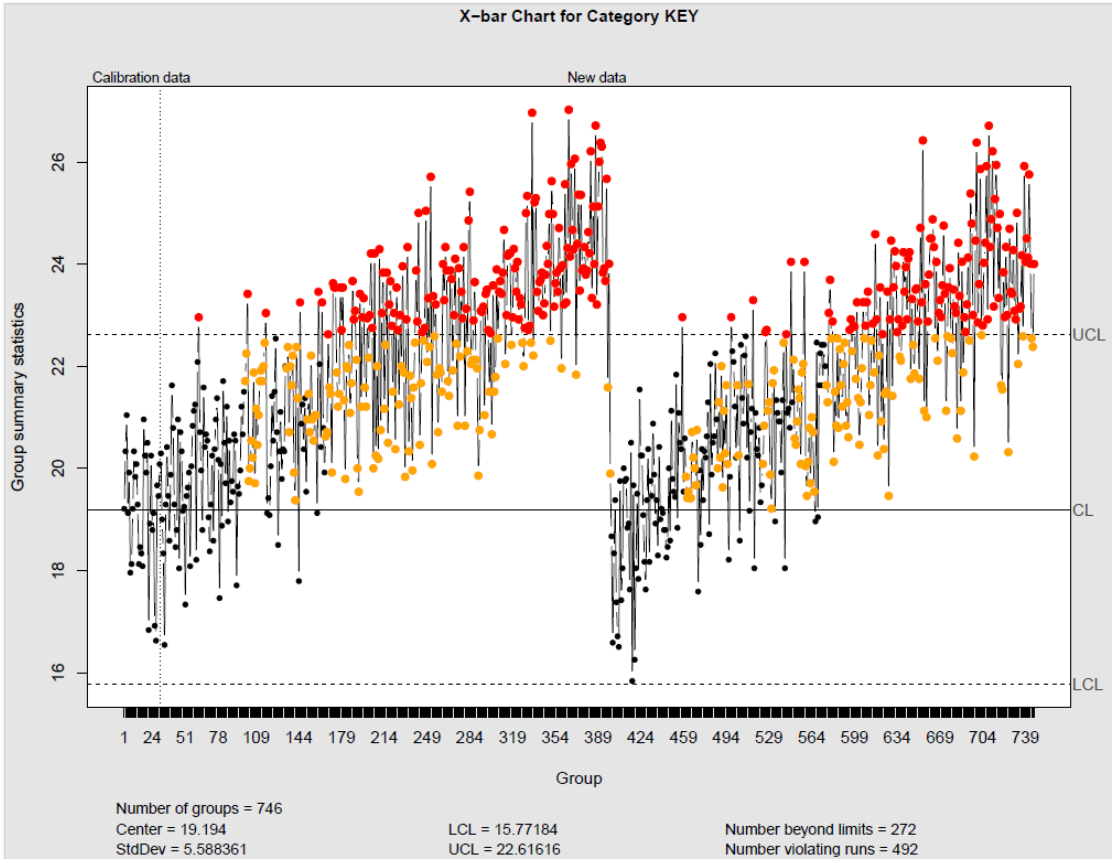
## Part 3

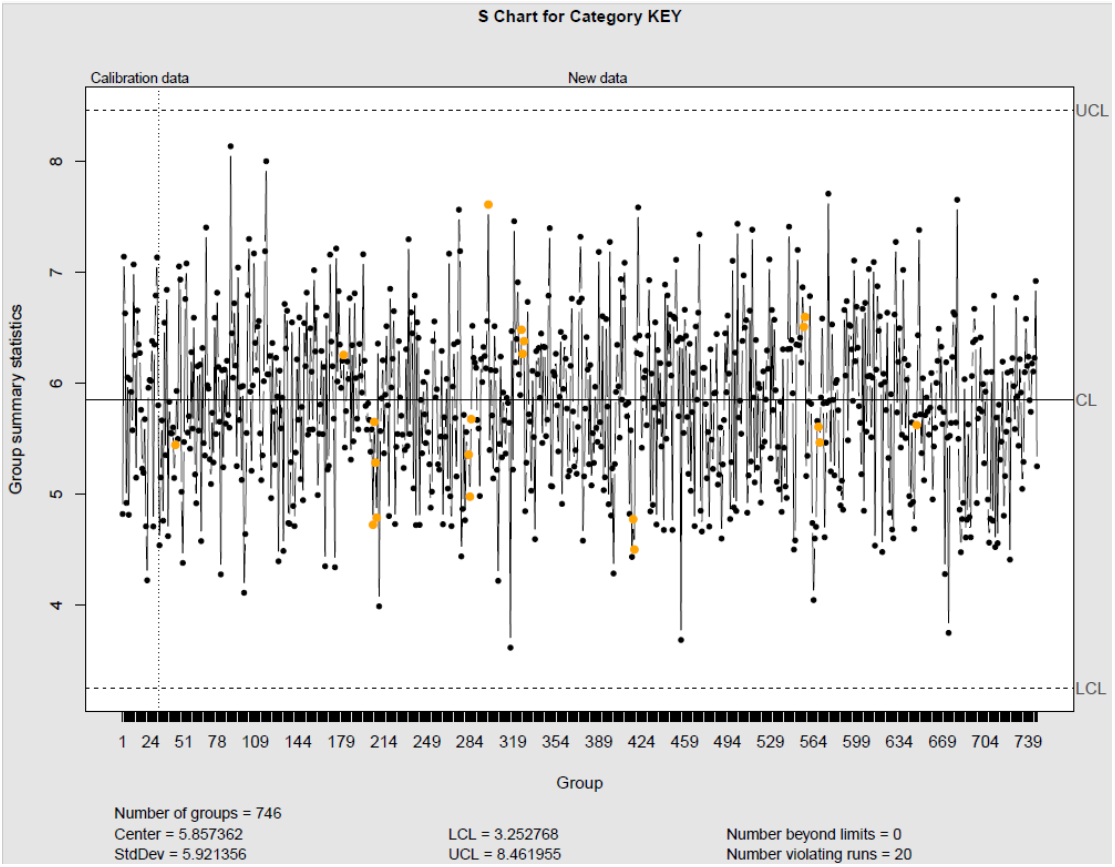
Delivery times for the following product categories are examined in the SPC report: CLO, KEY, LAP, MON, MOU, and SOF. For every category, there are S charts that show the standard deviation within the samples and X-bar charts that show the average delivery times of the grouped samples. The charts highlight points outside of control limits and violating runs that suggest possible process instability, making a distinction between calibration data (the original samples used to establish control limits) and new data. Using indices such as Cp, Cpl, Cpu, and Cpk, the report's conclusion assesses how well each category's delivery process satisfies specifications (LSL=0, USL=32 hours). Only SOF exhibits capability (Cpk > 1) because of its low mean and variability, while the others fall short, indicating areas for process improvement to lessen variability and better center the means within the specification limits.

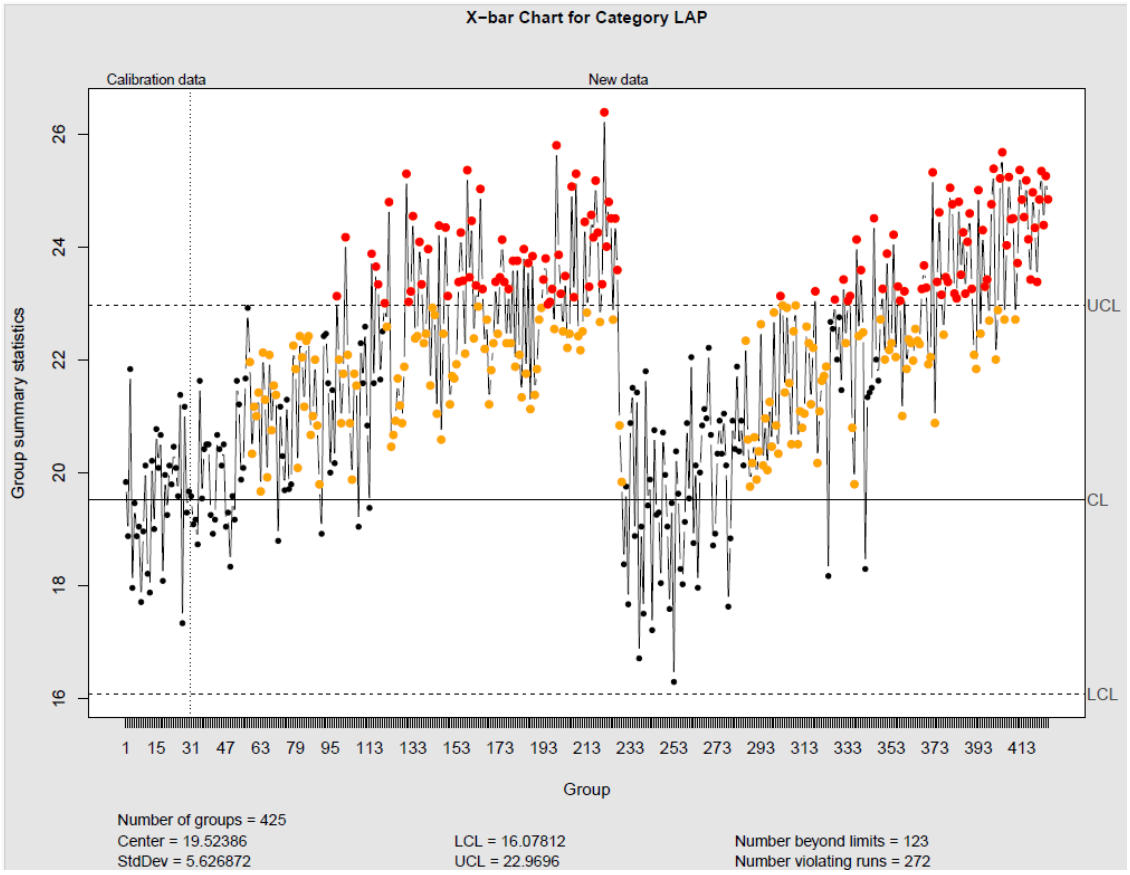
| Category | Cp | Cpl | Cpu | Cpk | Mean | Std |
|---|---|---|---|---|---|---|
| **CLO** | 0.90 | 1.08 | 0.72 | 0.72 | 19.23 | 5.94 |
| **KEY** | 0.92 | 1.10 | 0.73 | 0.73 | 19.28 | 5.82 |
| **LAP** | 0.90 | 1.10 | 0.70 | 0.70 | 19.61 | 5.93 |
| **MON** | 0.89 | 1.08 | 0.70 | 0.70 | 19.41 | 6.00 |
| **MOU** | 0.92 | 1.10 | 0.73 | 0.73 | 19.30 | 5.83 |
| **SOF** | 18.14 | 1.08 | 35.19 | 1.08 | 0.96 | 0.29 |

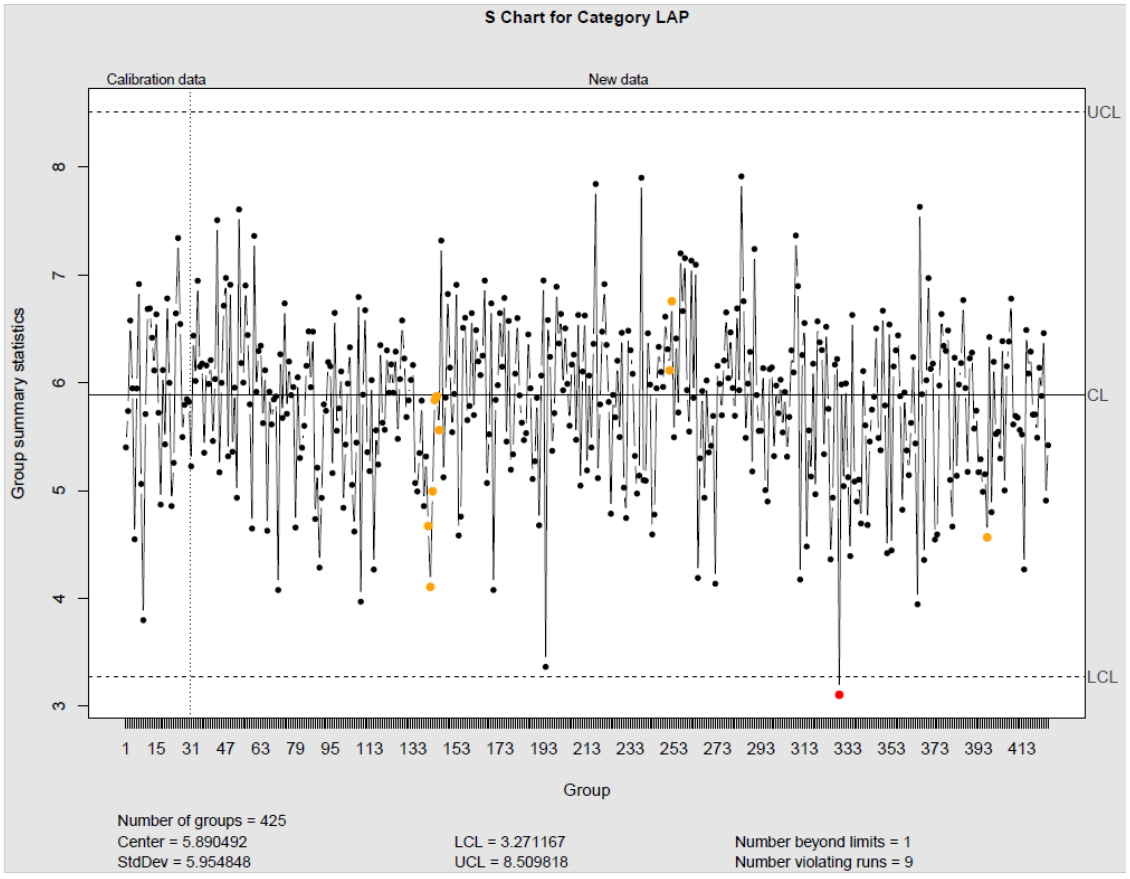

X−bar Chart for Category CLO
Calibration data
New data
Group summary statistics
16
18
20
22
24
26
UCL
CL
LCL
1 21 45 69 93 120 150 180 210 240 270 300 330 360 390 420 450 480 510 540 570 600 630
Group
Number of groups = 649
Center = 19.12594
StdDev = 5.498502
LCL = 15.75881
UCL = 22.49308
Number beyond limits = 241
Number violating runs = 434

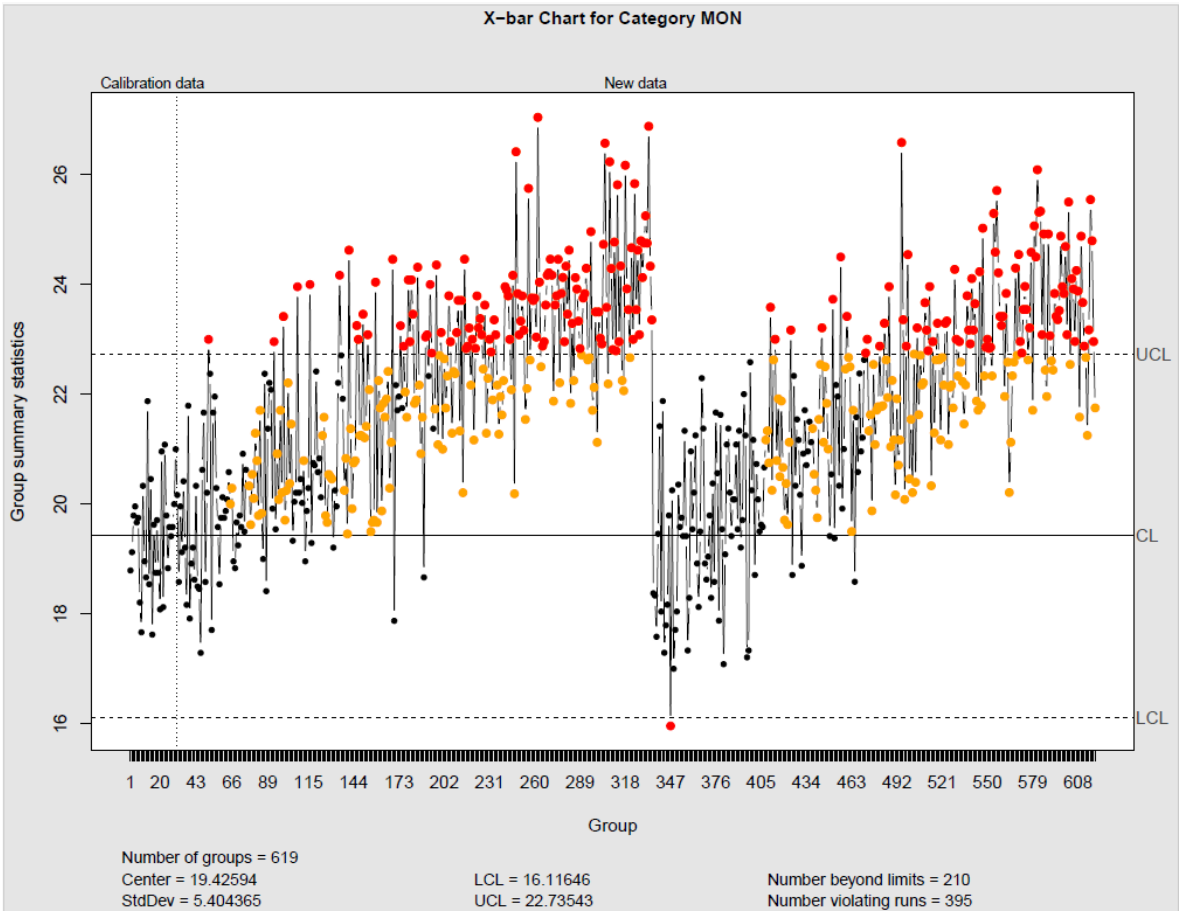

S Chart for Category CLO
Calibration data
New data
Group summary statistics
4
5
6
7
8
UCL
CL
LCL
1 21 45 69 93 120 150 180 210 240 270 300 330 360 390 420 450 480 510 540 570 600 630
Group
Number of groups = 649
Center = 5.907728
StdDev = 5.972272
LCL = 3.280738
UCL = 8.534718
Number beyond limits = 0
Number violating runs = 6

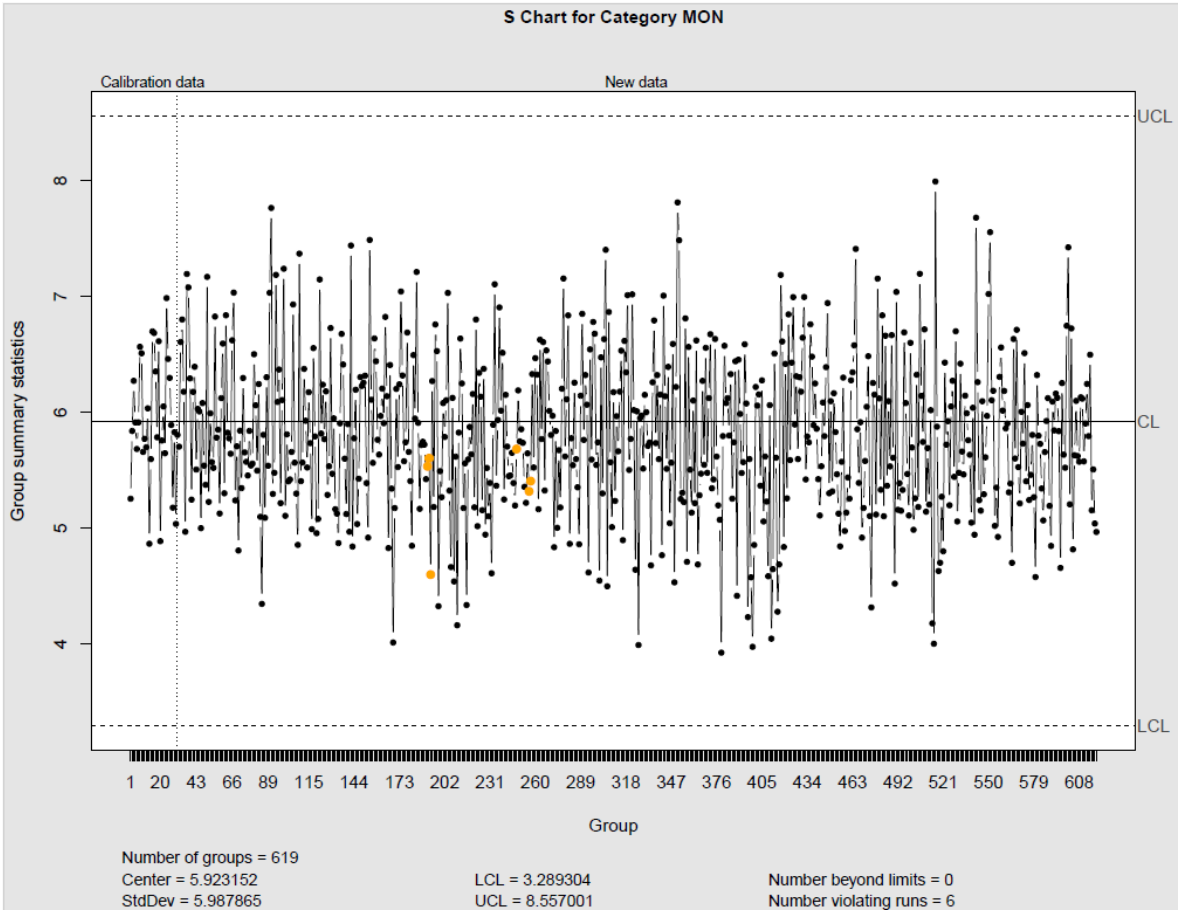

X-bar Chart for Category KEY
Calibration data
New data
Group summary statistics
16
18
20
22
24
26
UCL
CL
LCL
1 24 51 78 109 144 179 214 249 284 319 354 389 424 459 494 529 564 599 634 669 704 739
Group
Number of groups = 746
Center = 19.194
StdDev = 5.588361
LCL = 15.77184
UCL = 22.61616
Number beyond limits = 272
Number violating runs = 492

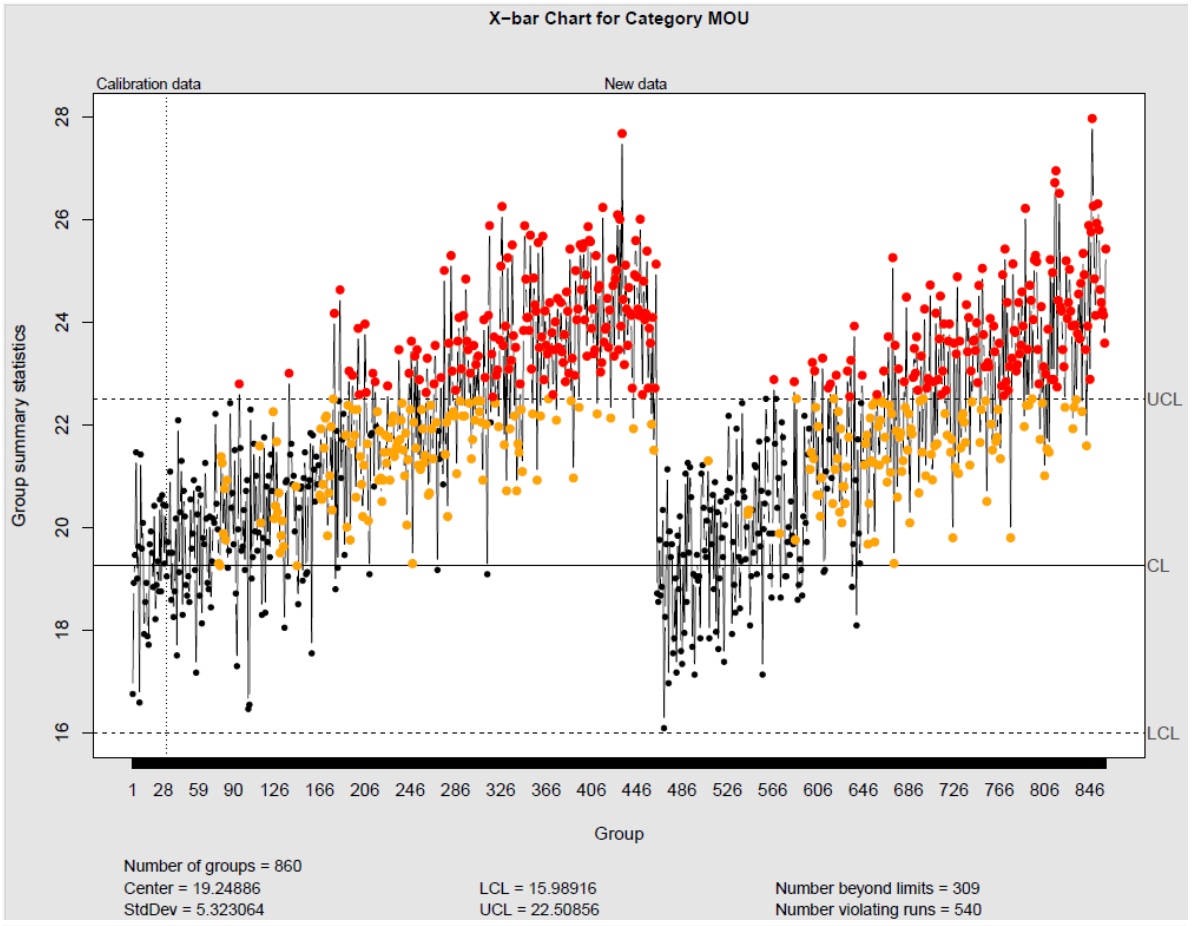

S Chart for Category KEY
Calibration data
New data
Group summary statistics
4
5
6
7
8
UCL
CL
LCL
1 24 51 78 109 144 179 214 249 284 319 354 389 424 459 494 529 564 599 634 669 704 739
Group
Number of groups = 746
Center = 5.857362
StdDev = 5.921356
LCL = 3.252768
UCL = 8.461955
Number beyond limits = 0
Number violating runs = 20

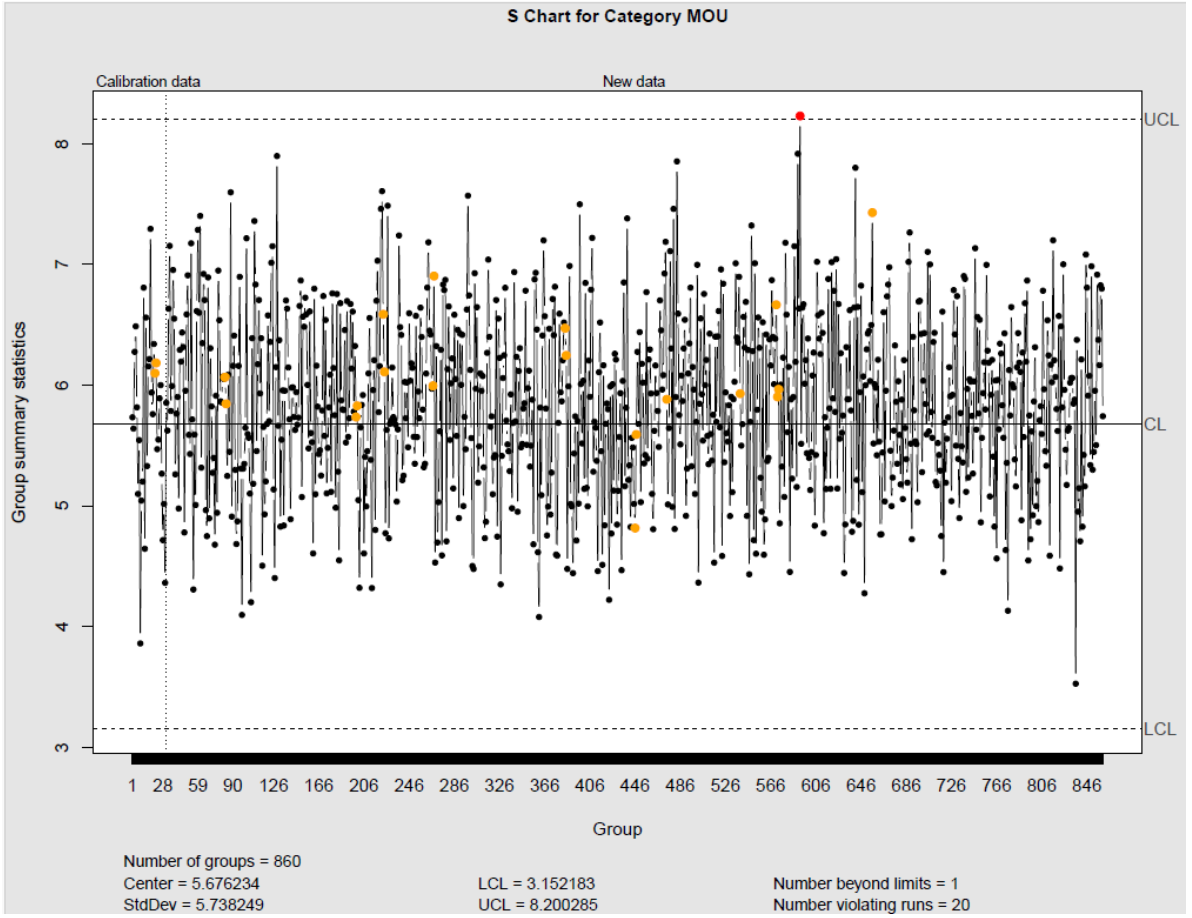

X-bar Chart for Category LAP
Calibration data
New data
Group summary statistics
UCL
CL
LCL
Group
Number of groups = 425
Center = 19.52386
StdDev = 5.626872
LCL = 16.07812
UCL = 22.9696
Number beyond limits = 123
Number violating runs = 272

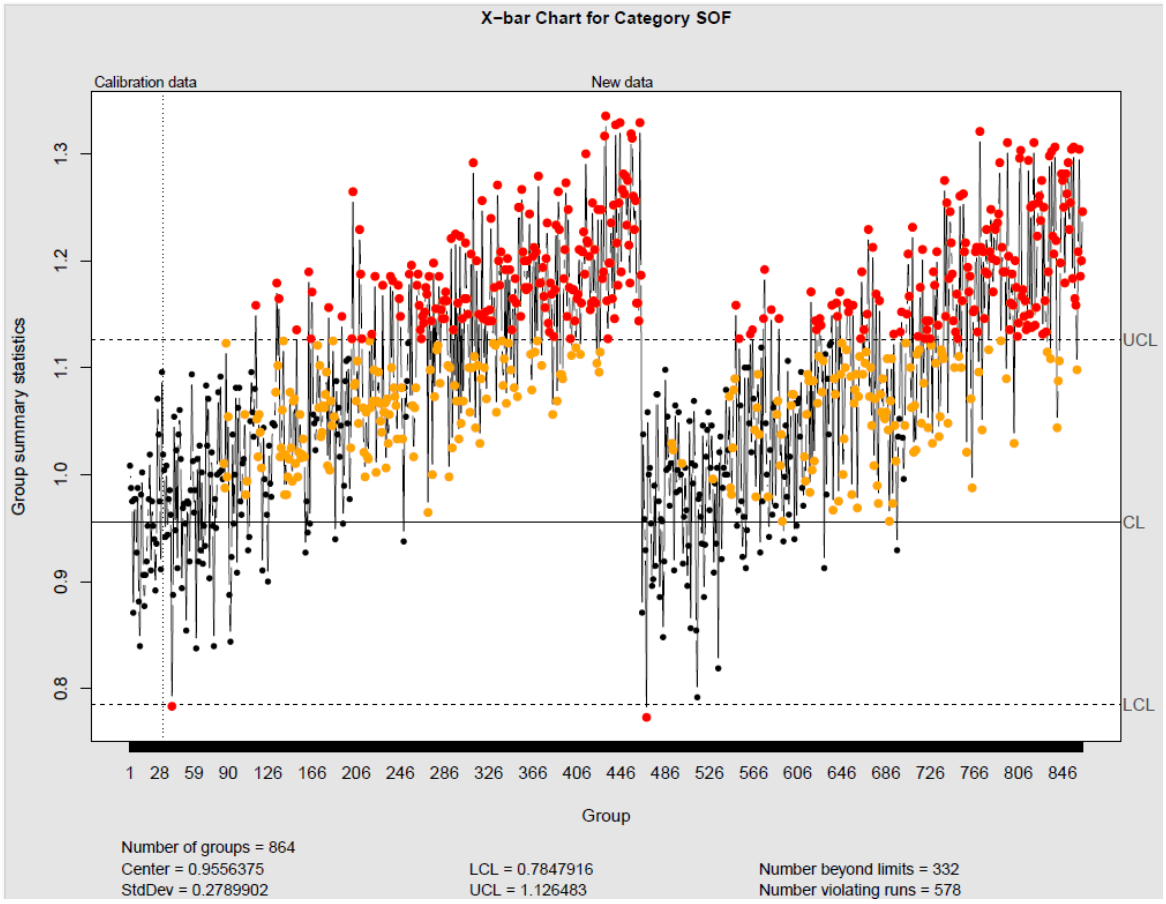

S Chart for Category LAP
Calibration data
New data
Group summary statistics
UCL
CL
LCL
Group
Number of groups = 425
Center = 5.890492
StdDev = 5.954848
LCL = 3.271167
UCL = 8.509818
Number beyond limits = 1
Number violating runs = 9

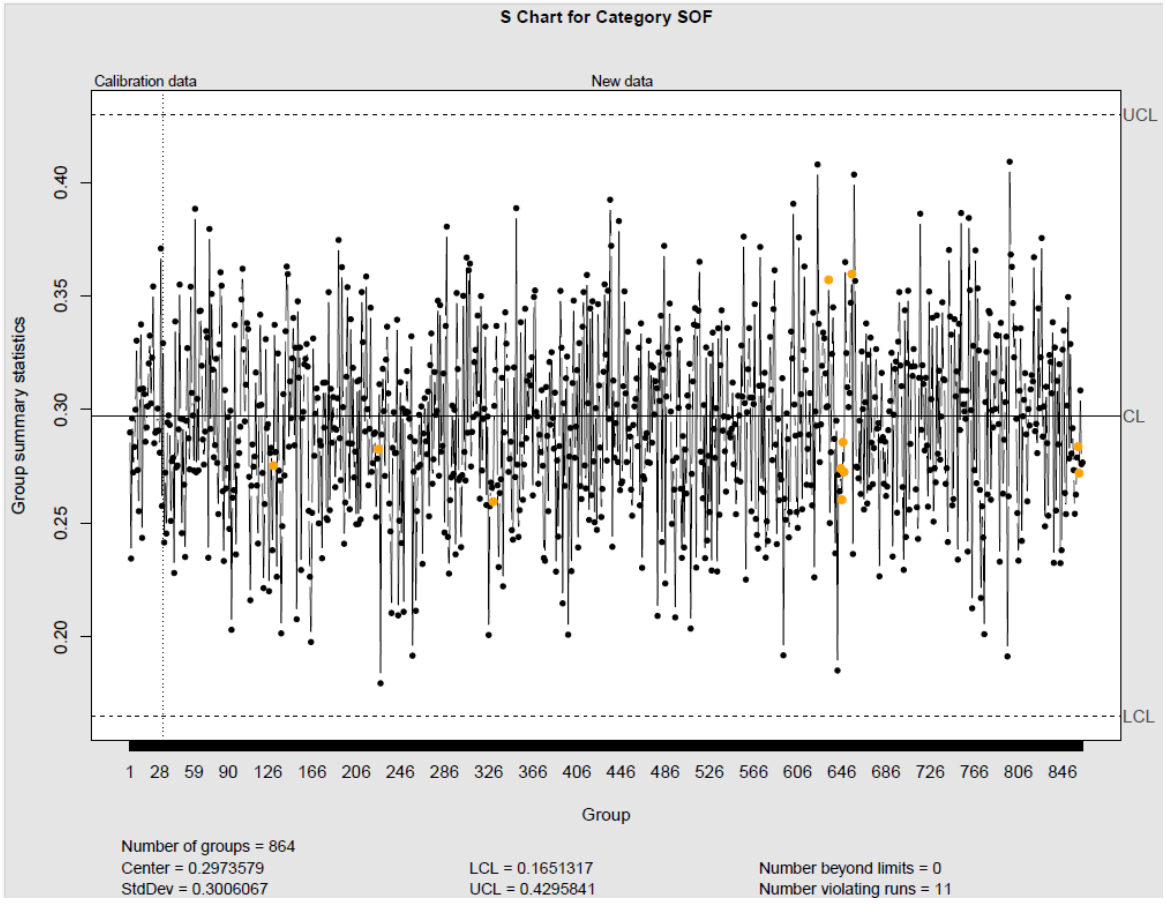

X-bar Chart for Category MON
Calibration data
New data
Group summary statistics
16
18
20
22
24
26
UCL
CL
LCL
1 20 43 66 89 115 144 173 202 231 260 289 318 347 376 405 434 463 492 521 550 579 608
Group
Number of groups = 619
Center = 19.42594
StdDev = 5.404365
LCL = 16.11646
UCL = 22.73543
Number beyond limits = 210
Number violating runs = 395


S Chart for Category MON
Calibration data
New data
Group summary statistics
4
5
6
7
8
UCL
CL
LCL
1 20 43 66 89 115 144 173 202 231 260 289 318 347 376 405 434 463 492 521 550 579 608
Group
Number of groups = 619
Center = 5.923152
StdDev = 5.987865
LCL = 3.289304
UCL = 8.557001
Number beyond limits = 0
Number violating runs = 6


X−bar Chart for Category MOU
Calibration data
New data
Group summary statistics
16
18
20
22
24
26
28
UCL
CL
LCL
1 28 59 90 126 166 206 246 286 326 366 406 446 486 526 566 606 646 686 726 766 806 846
Group
Number of groups = 860
Center = 19.24886
StdDev = 5.323064
LCL = 15.98916
UCL = 22.50856
Number beyond limits = 309
Number violating runs = 540


S Chart for Category MOU
Calibration data
New data
Group summary statistics
3
4
5
6
7
8
UCL
CL
LCL
1 28 59 90 126 166 206 246 286 326 366 406 446 486 526 566 606 646 686 726 766 806 846
Group
Number of groups = 860
Center = 5.676234
StdDev = 5.738249
LCL = 3.152183
UCL = 8.200285
Number beyond limits = 1
Number violating runs = 20


X-bar Chart for Category SOF
Calibration data
New data
Group summary statistics
0.8
0.9
1.0
1.1
1.2
1.3
UCL
CL
LCL
1 28 59 90 126 166 206 246 286 326 366 406 446 486 526 566 606 646 686 726 766 806 846
Group
Number of groups = 864
Center = 0.9556375
StdDev = 0.2789902
LCL = 0.7847916
UCL = 1.126483
Number beyond limits = 332
Number violating runs = 578


S Chart for Category SOF
Calibration data
New data
Group summary statistics
0.20
0.25
0.30
0.35
0.40
UCL
CL
LCL
1 28 59 90 126 166 206 246 286 326 366 406 446 486 526 566 606 646 686 726 766 806 846
Group
Number of groups = 864
Center = 0.2973579
StdDev = 0.3006067
LCL = 0.1651317
UCL = 0.4295841
Number beyond limits = 0
Number violating runs = 11

# Part 4

## 4.1 Type 1 (Manufacturer's Error)

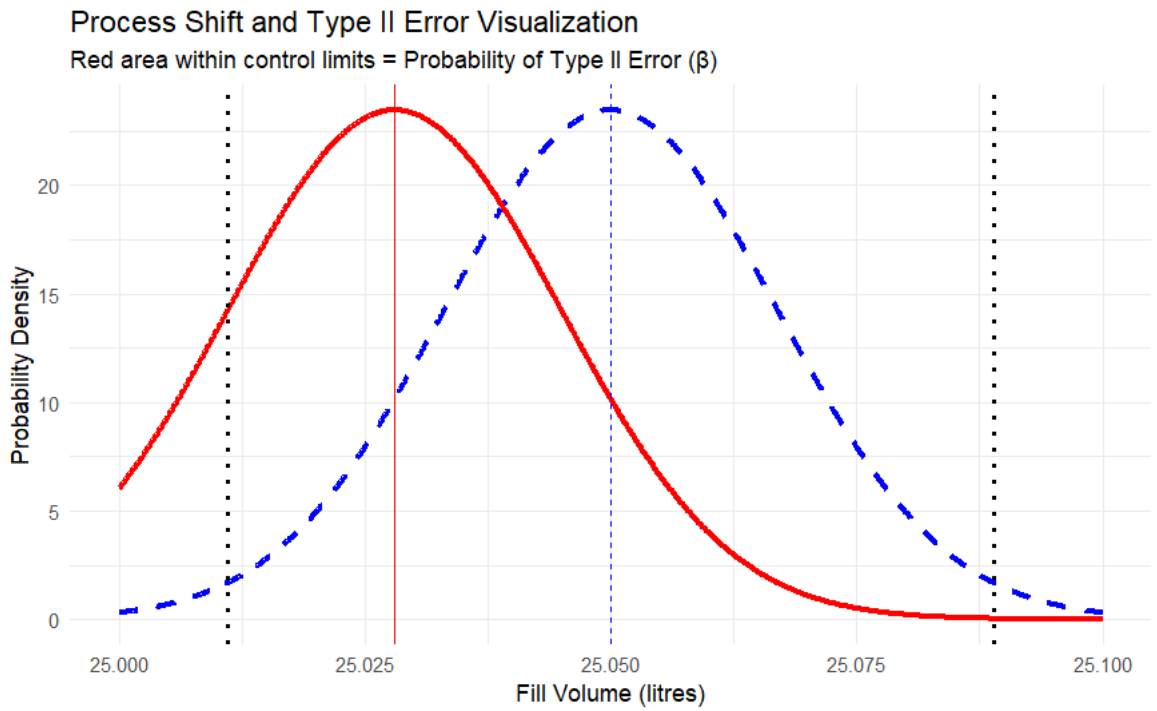
When a control chart indicates that a process is out of control when it is actually stable, this is known as a Type I (Manufacturer's) error. Based on the normal distribution of process variation, this error is theoretical. For instance, there is a 0.27% chance that one sample will fall outside the ±3σ limits, and a 0.0078 (approximately 8 in 1000) chance that seven consecutive samples will appear above the centerline. These tiny odds demonstrate that even in a process that is perfectly controlled, false alarms can happen, albeit infrequently.
From R studio:

```
Explanation:
Type I Error represents the chance of falsely signalling that a process is out of
control
when it is actually stable. For example, the probability of getting 7 consecutive
samples
above the centreline is 0.5^7 = 0.0078, or about 8 in 1000.
```

## 4.2 Type 2 (Consumer's Error)

When the process mean has changed but the control chart is unable to identify it because the sample averages are still within the control limits, this is known as a Type II (Consumer's) error. In this instance, the variation slightly increased and the process average changed from 25.05 L to 25.028 L. There is a high likelihood that the problem would go unnoticed because the calculated probability of failing to detect this shift (β) is roughly 0.73, or 73%. This demonstrates that if the control limits are large in comparison to the process variation, even slight changes in the mean may go unnoticed.
From R Studio:

```
Probability of Type II Error (β): 0.8412 (84.12%)
Power of the Test (1 - β): 0.1588 (15.88%)

Interpretation:
β represents the chance of failing to detect that the filling process mean has shifted.
A high β means the consumer might unknowingly receive underfilled or inconsistent
bottles.
```



## 4.3 Fixing head office and product data errors

### 1. Data Loading and Inspection

Examining the data from Question 4.3 in comparison to Question 1.2 for the first time is like cleaning up a cluttered desk! Products_data2025.csv and products_Headoffice2025.csv have been updated to include categories that match their prefixes, which weren't sorted in 1.2, as well as corrected ProductIDs (such as changing "NA011" to "SOF011). The fundamentals remain the same—60 products, 5000 customers—but the clearer labels help us to believe what we see from the outset.

### 2. Summary Statistics

Examining the data, 4.3's statistics are nicely polished. Markups settle into a tighter 23-25% range, unlike the wilder spread in 1.2, and the mean SellingPrice may dip slightly (e.g., Software from ~399 to ~396) since prices and markups for items 11-60 are now taken from the first 10 of each type in products_data.csv. It's similar to removing wrinkles to reveal the essence of the data.

### 3. Handling Missing Values

This step remains quiet and steady because there aren't many missing values in either 4.3 or 1.2. It's similar to checking the weather and discovering that it's still sunny, which allows us to concentrate on the enjoyable activities without any interruptions.

### 4. Data Filtering and Subsetting

Both analyses still make it easy to filter 2023 sales, but in 4.3, those updated prices infiltrate the mix, paving the way for later, more precise insights. It's a minor adjustment that could have a significant impact later on.

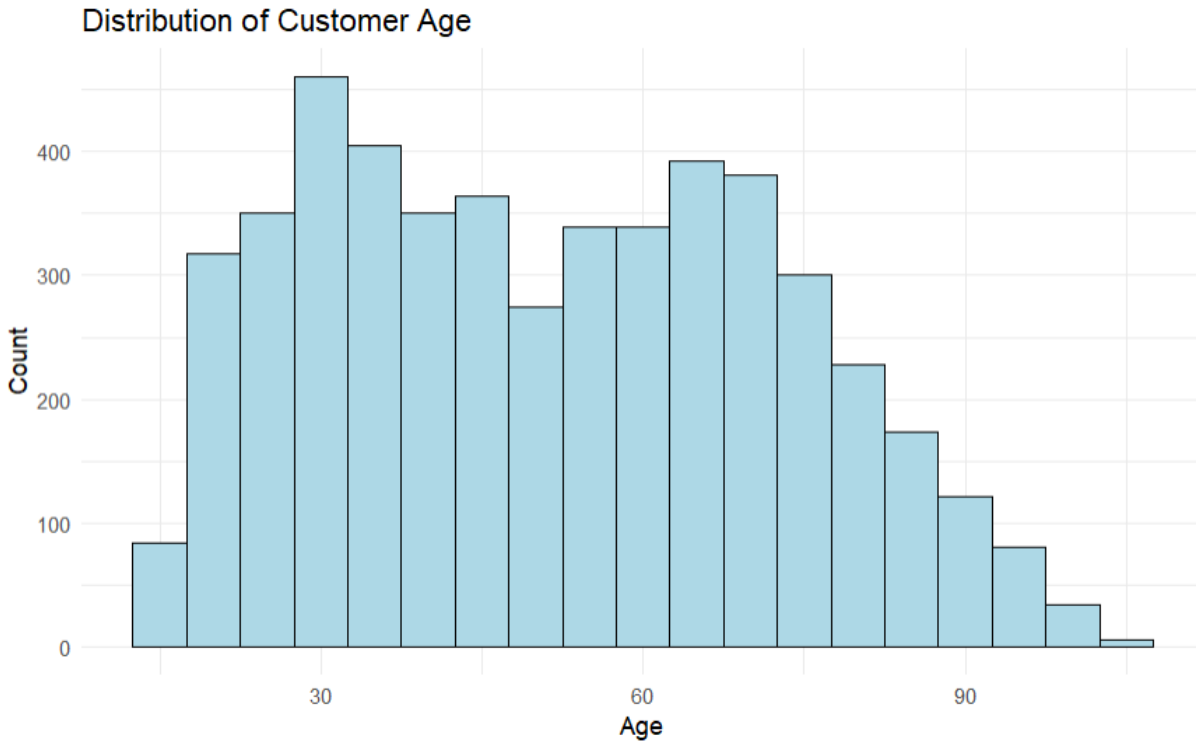
### 5. Data Visualization

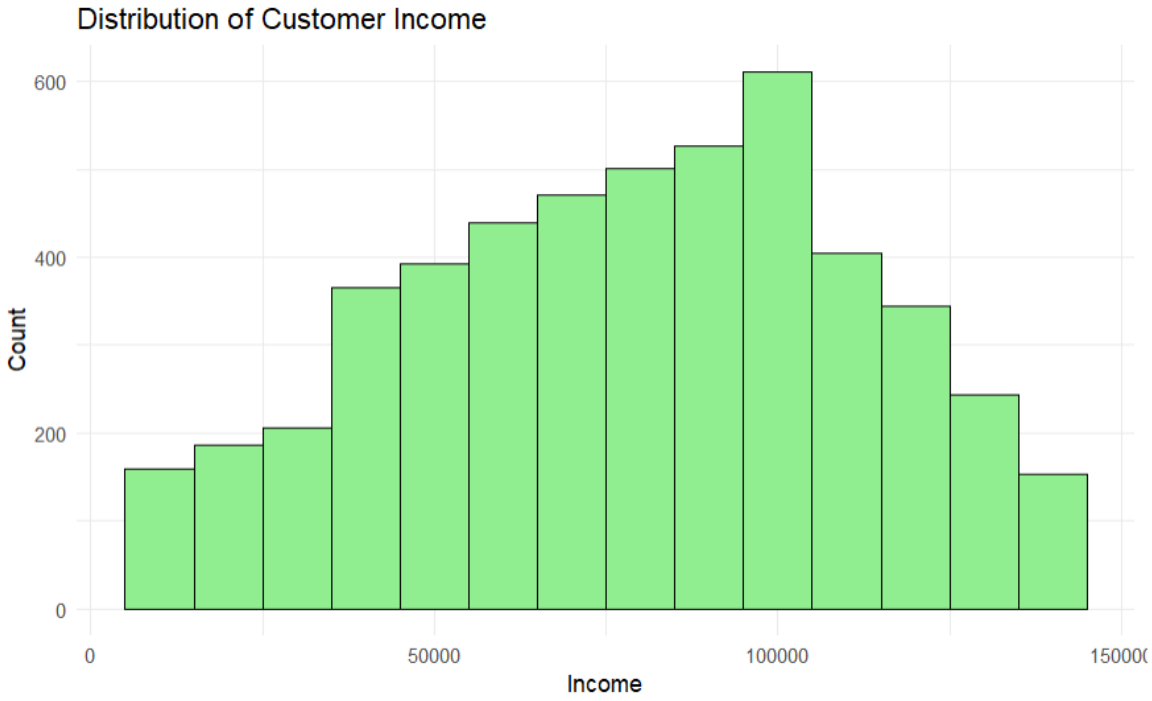
Boxplots display updated price ranges, SellingPrice histograms now hug closer together by category (e.g., laptops peak more neatly), and the revenue bar chart probably shows higher totals for expensive items like laptops, correcting the underestimates from 1.2. It's similar to improving a grainy photo of the company to a clear one.

### 6. Exploring Relationships

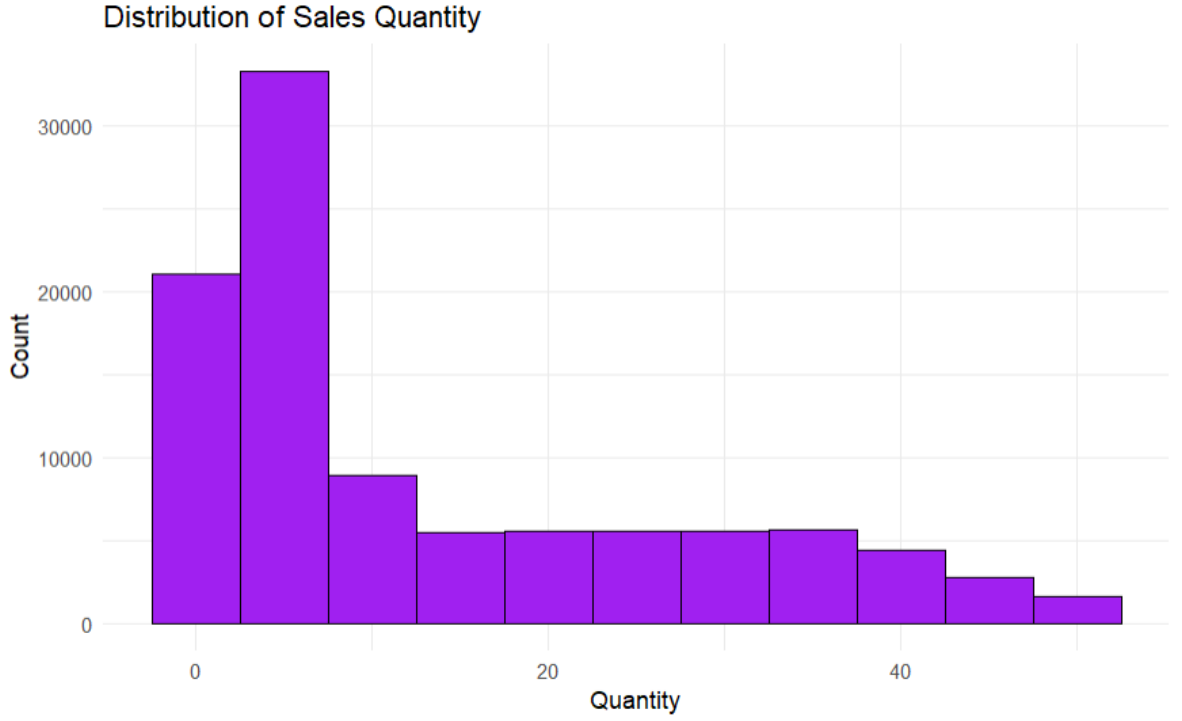
Peeking at relationships, 4.3's scatterplot matrix shows a stronger link between SellingPrice and Revenue thanks to the price fixes, while Age and Income still

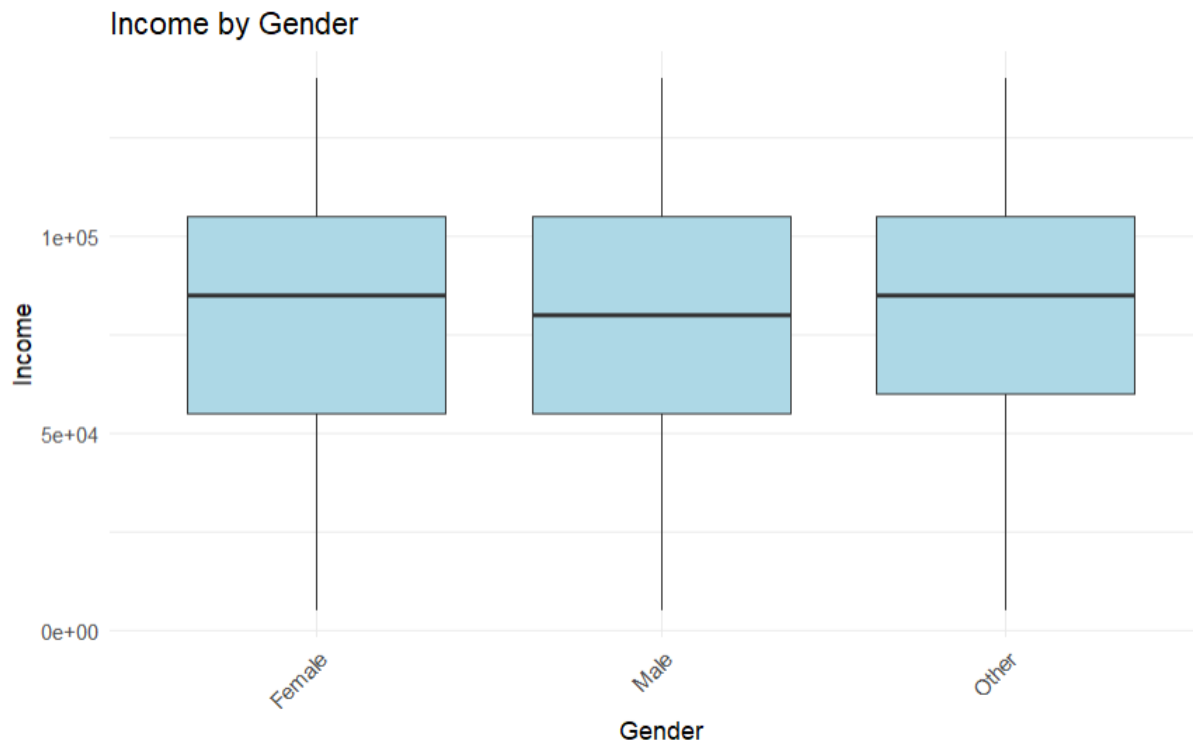
dance around without much connection—just like before. The updated data gives us a sharper lens, highlighting where the real money moves are, making the journey through the numbers even more rewarding.

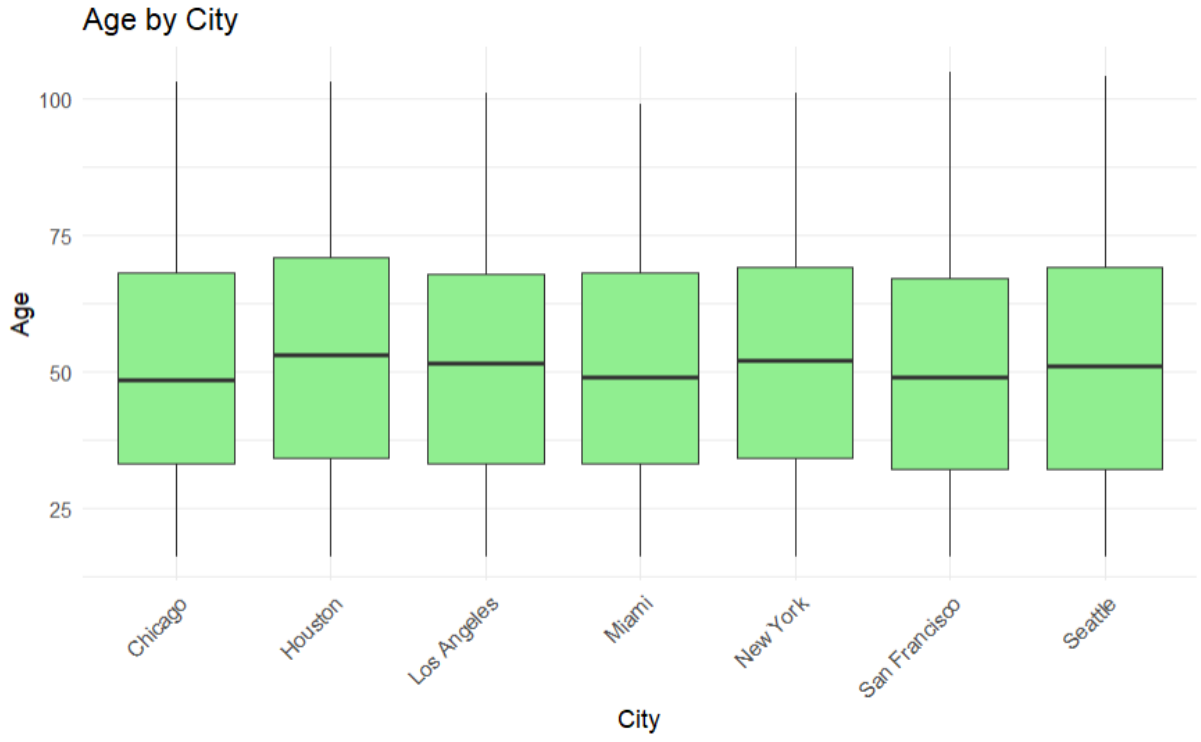


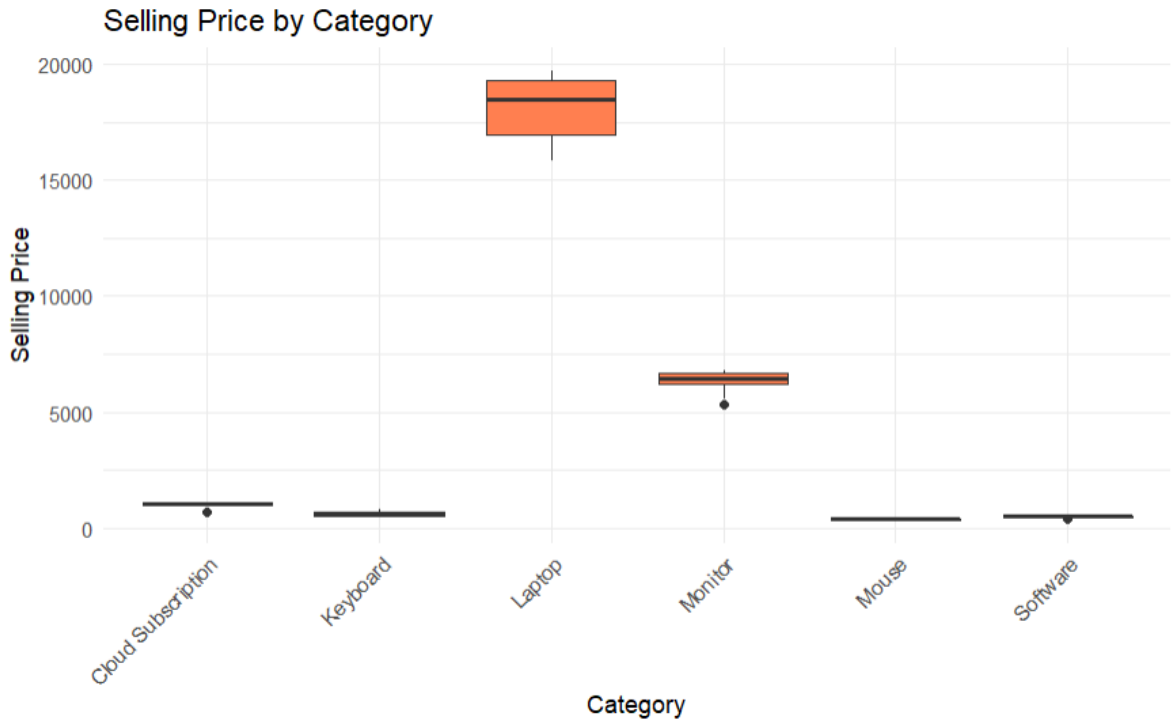



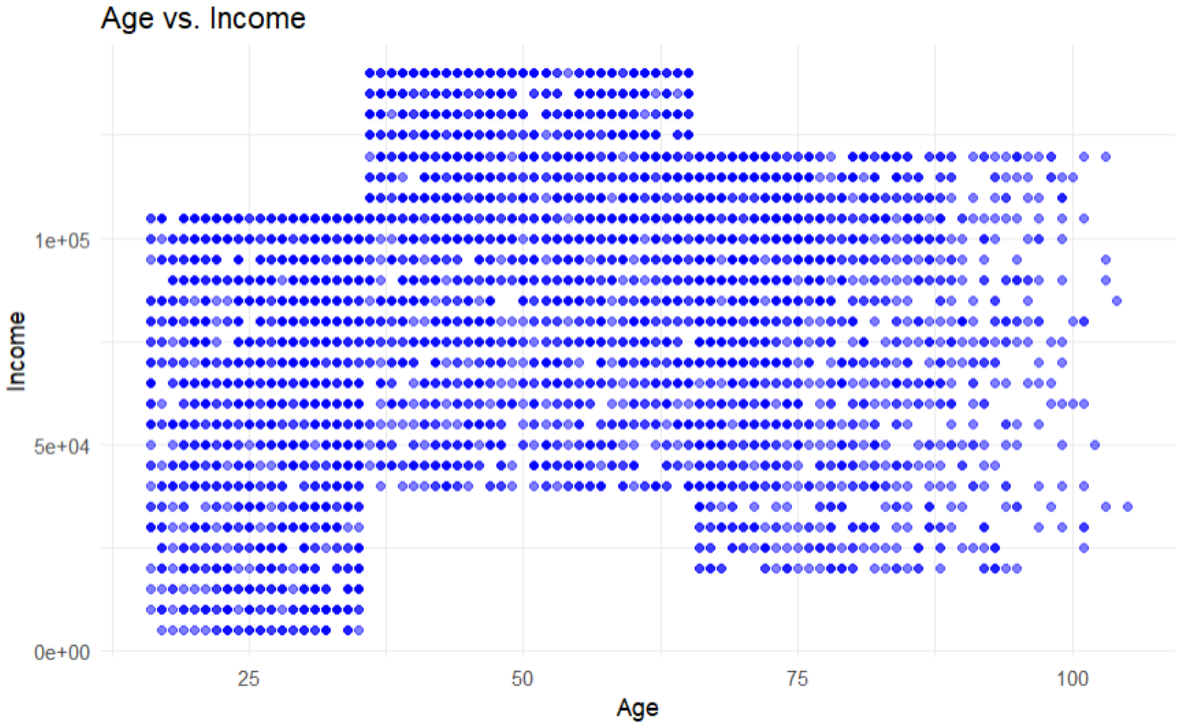

Distribution of Product Selling Price
Count
20
10
0
0
5000
10000
15000
20000
Selling Price

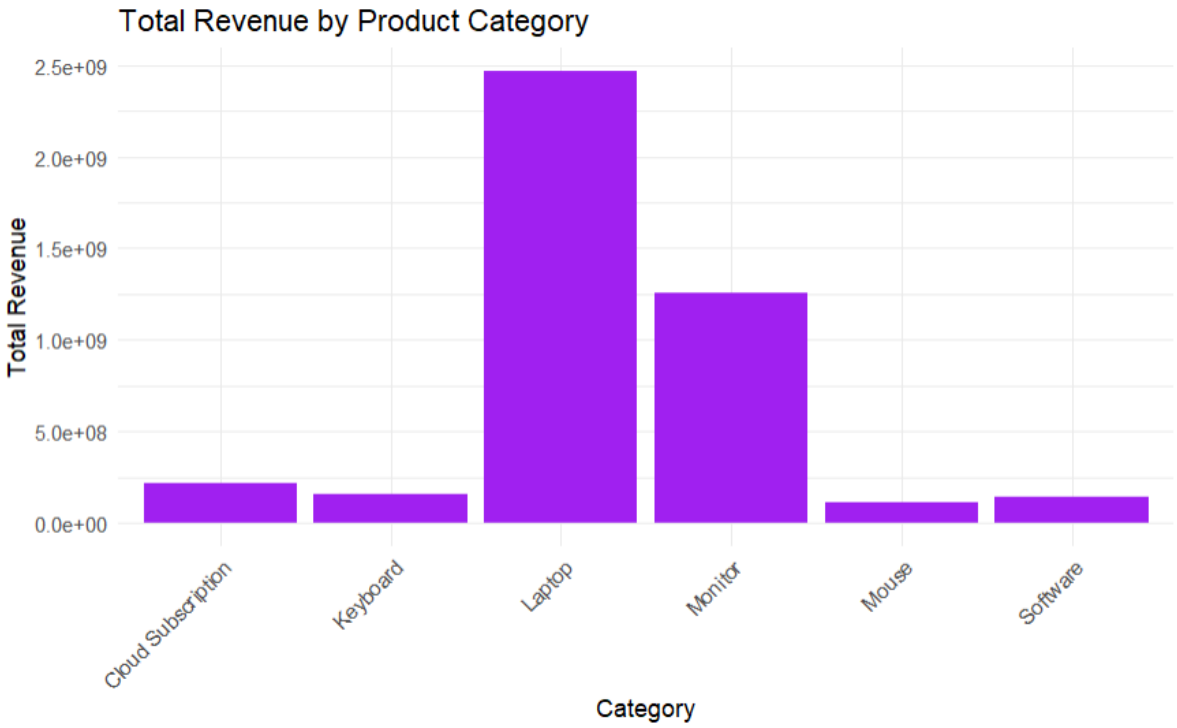

Distribution of Sales Quantity
Count
30000
20000
10000
0
0
20
40
Quantity

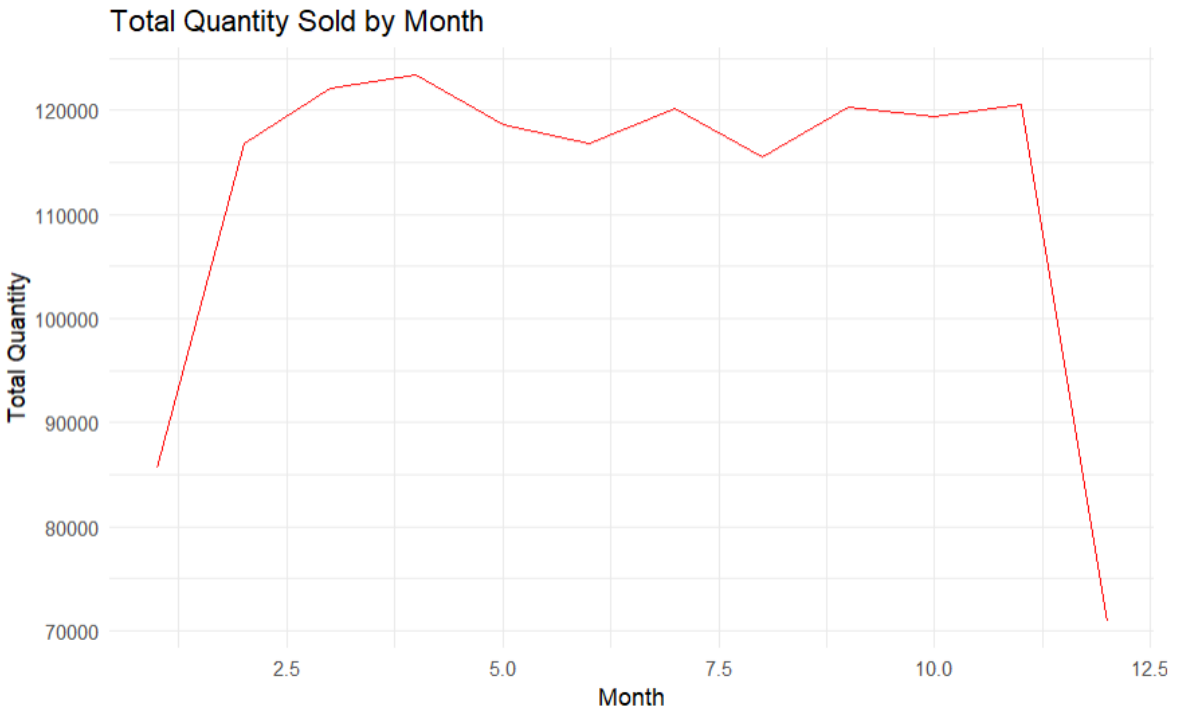

Income by Gender
Income
1e+05
5e+04
0e+00
Female
Male
Other
Gender


Age by City
Age
100
75
50
25
Chicago
Houston
Los Angeles
Miami
New York
San Francisco
Seattle
City


Selling Price by Category
20000
15000
10000
5000
0
Selling Price
Cloud Subscription
Keyboard
Laptop
Monitor
Mouse
Software
Category


Age vs. Income
1e+05
5e+04
0e+00
Income
25
50
75
100
Age

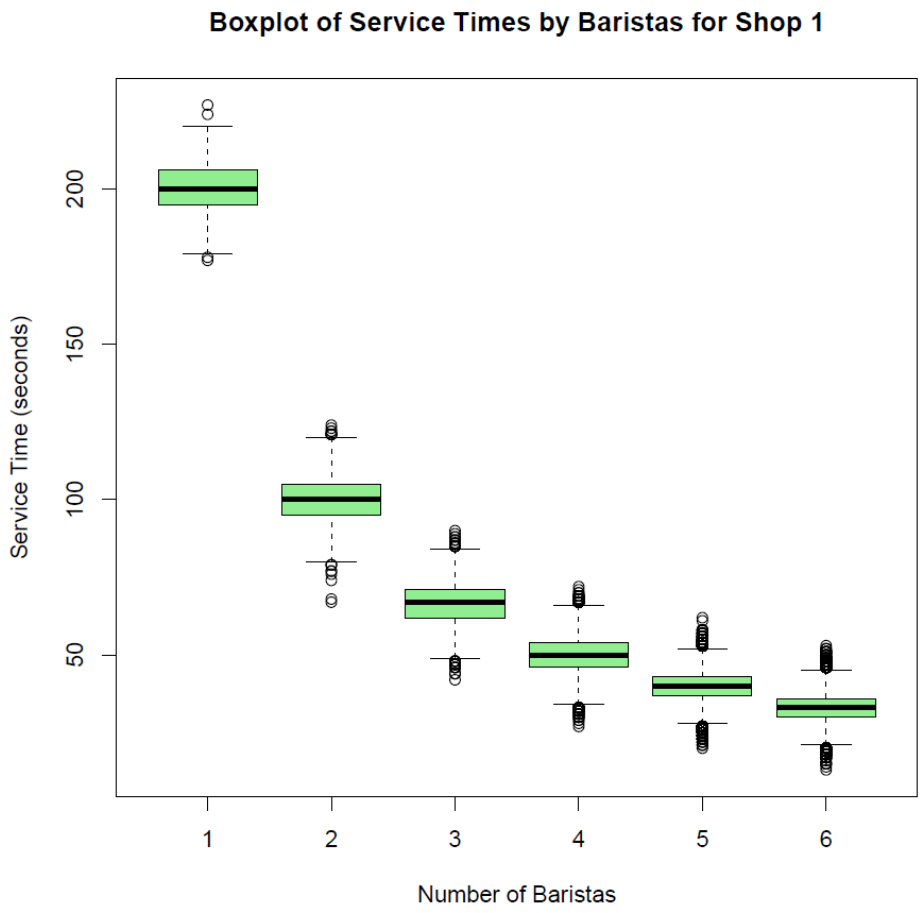


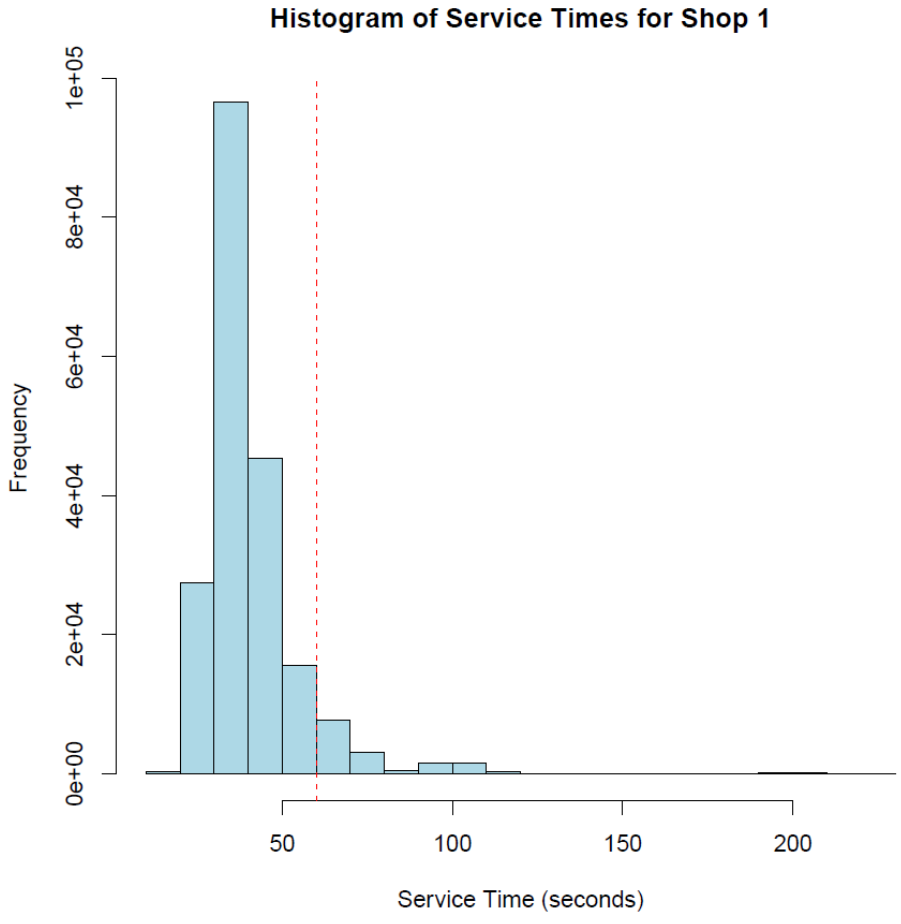



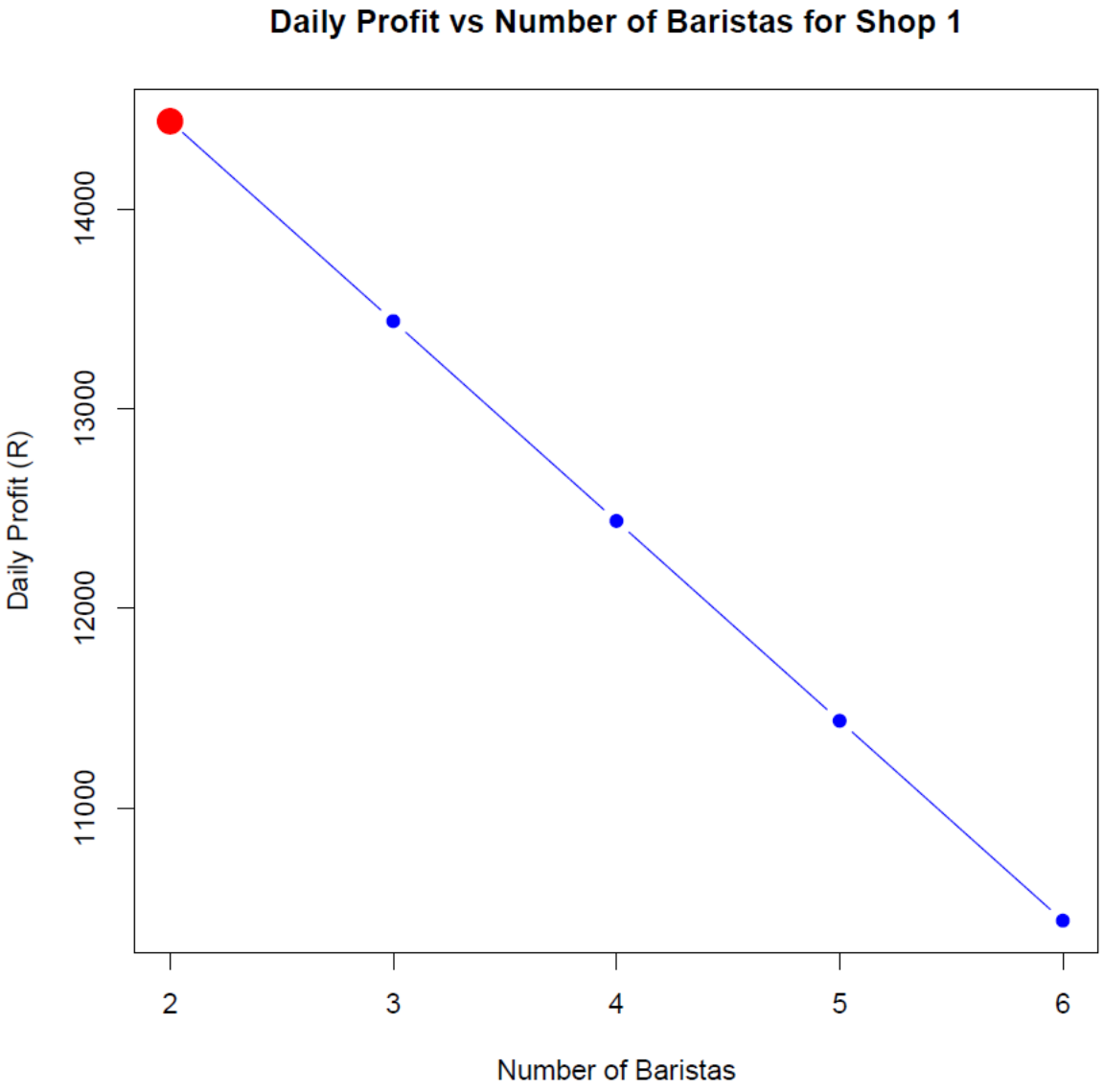

| Category | Total Sales value |
|---|---|
| Cloud Subscription | 98715482 |
| Keyboard | 73499067 |
| Laptop | 1163889479 |
| Monitor | 578385570 |
| Mouse | 51219577 |
| Software | 66468485 |

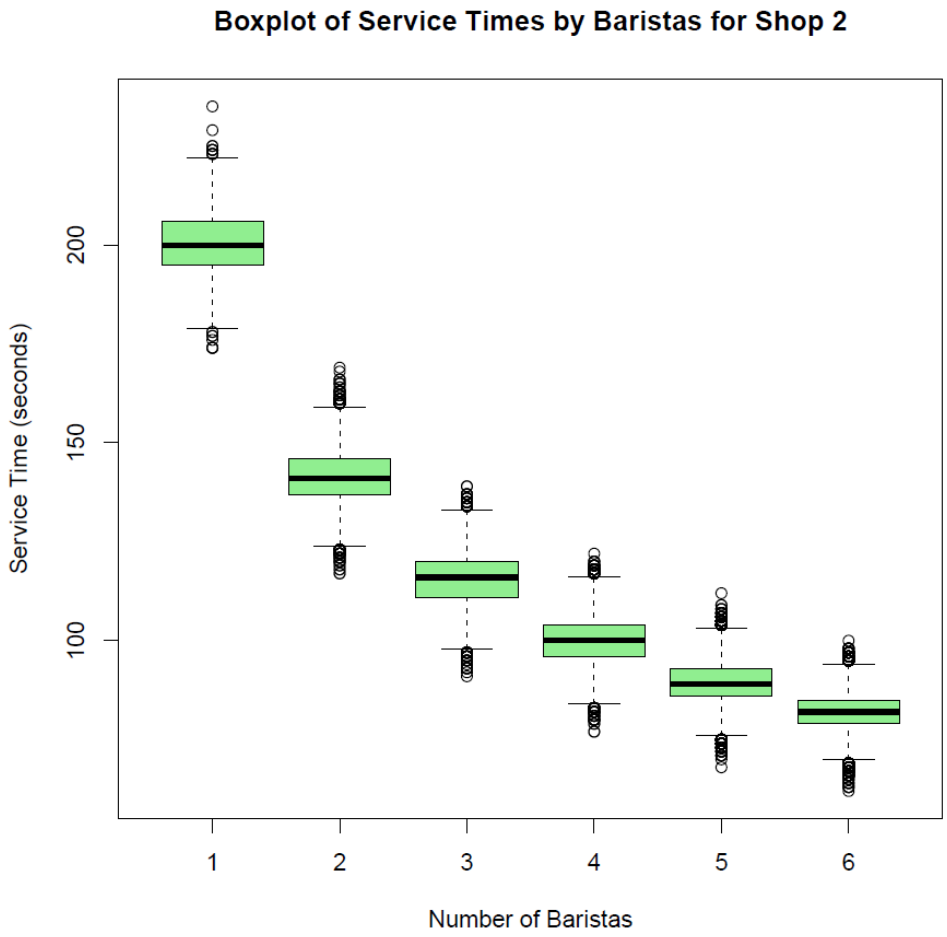
## Part 5

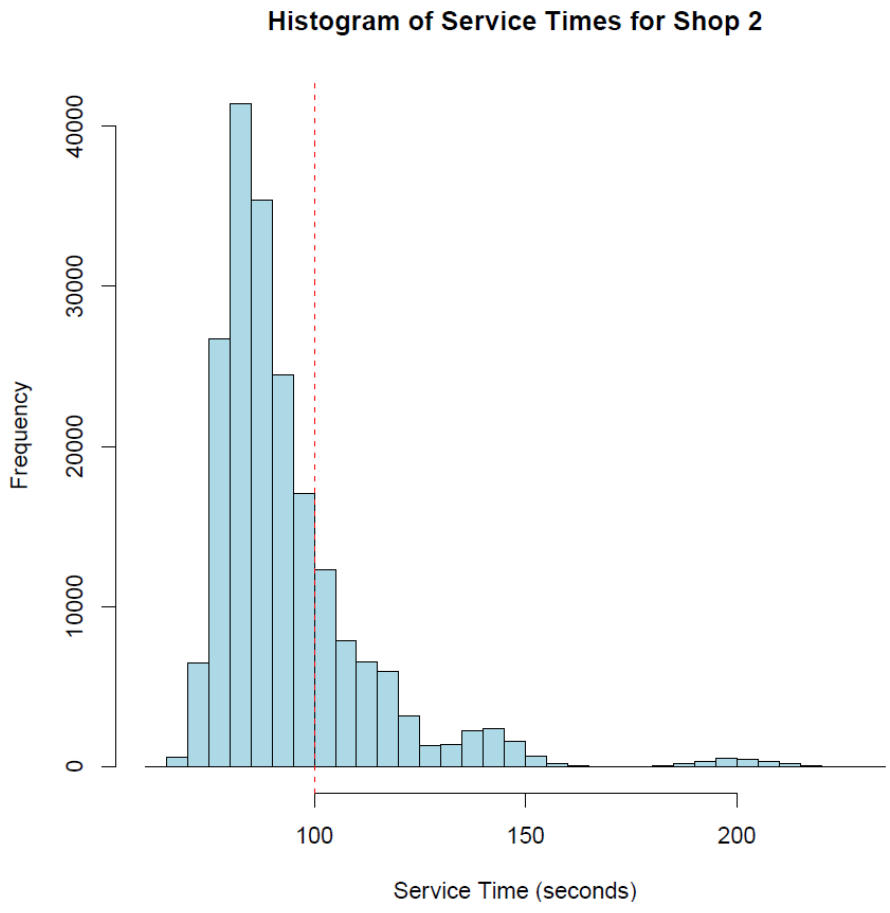
Based on the supplied excel spreadsheets (timeToServe and timeToServe2) the R script I used supplied me with a thorough visual analysis of service times and profit optimization for the two coffee shops. Even though the tail stretches to more than 200 seconds, Shop 1's service time histogram displays a right-skewed distribution, with the most times concentrated between 20 and 50 seconds and peaks around 30-40 seconds. The red dashed line at 60 seconds shows the reliable service threshold. As the number of baristas increases, the median service times for Shop 1 show a clear trend of decreasing from about 200 seconds with one barista (due to high variability and outliers) to about 30 seconds with six baristas (due to much tighter interquartile ranges). This shows that the more staff you have greatly lowers wait times and variability. Given the service capacity and costs, the daily profit plot for Shop 1 shows a linear decline from roughly R14,000 at two baristas to R11,000 at six baristas, with the optimal point indicated at two baristas. This shows that the less staff you have the more profit you will make. The red dashed line at 100 seconds shows a reliable threshold for this shop's longer average times. Shop 2's histogram has a similar right-skewed pattern, but is is shifted a bit higher, with the most of the service times falling between 70 and 100 seconds and a peak at 80-90 seconds that extends to over 200 seconds. There might be some variation across all the levels, but the boxplot for shop 2 shows a trend of declining medians with more baristas, beginning at about 200 seconds for 1 barista and falling to around 80 seconds for 6 baristas. Finally, Shop 2's profit plot shows a slight peak at 3 baristas, around about R13 250, and then a decline to roughly R10 500 at 6 baristas. The optimal staffing level is indicated at 3, showing a balance where moderate staffing balances capacity best with demand while keeping personnel costs under control. Shop 1 prefers fewer baristas for the highest profit whereas Shop 2 benefits from a slightly higher number of baristas. These visuals show the trade-offs between service efficiency, staffing levels, and profitability overall.

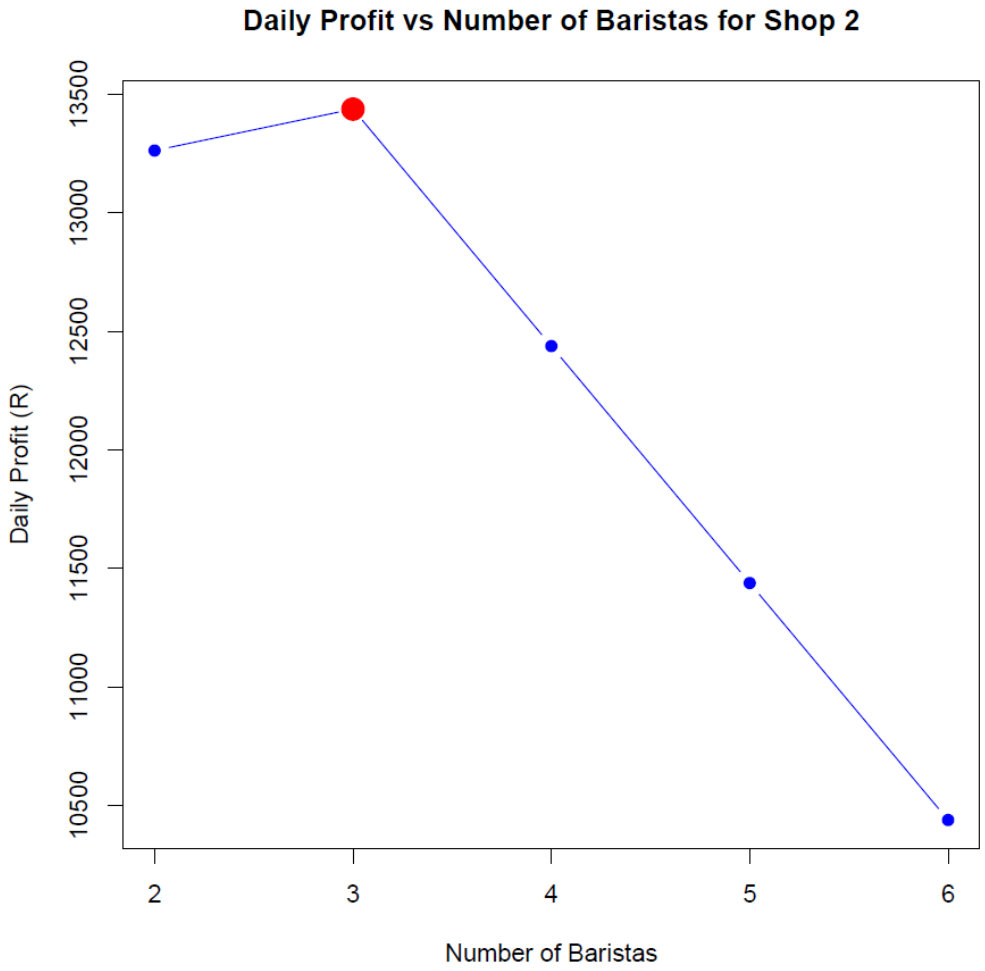

Boxplot of Service Times by Baristas for Shop 1
Service Time (seconds)
50
100
150
200
1
2
3
4
5
6
Number of Baristas


Histogram of Service Times for Shop 1
Frequency
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
50
100
150
200
Service Time (seconds)


Daily Profit vs Number of Baristas for Shop 1
Daily Profit (R)
14000
13000
12000
11000
2
3
4
5
6
Number of Baristas


Boxplot of Service Times by Baristas for Shop 2
Service Time (seconds)
100
150
200
1
2
3
4
5
6
Number of Baristas


Histogram of Service Times for Shop 2
Frequency
0
10000
20000
30000
40000
100
150
200
Service Time (seconds)

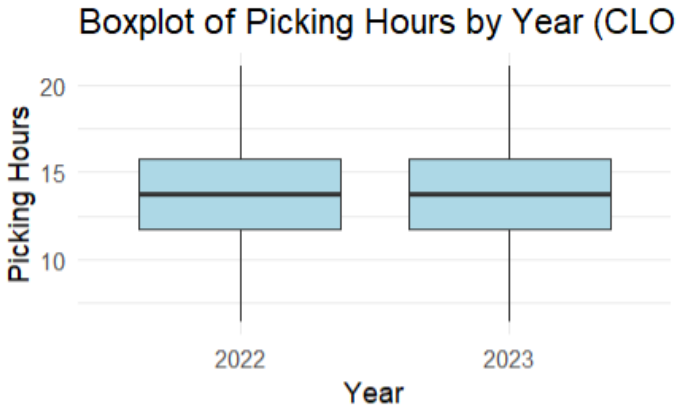


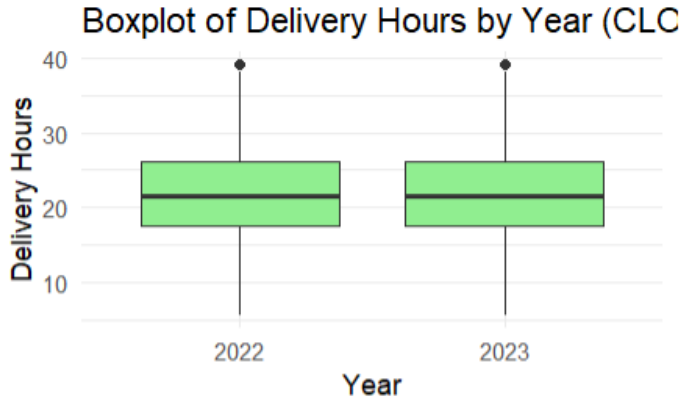

```
  Daily capacity: 935.54 customers
  Customers served: 547.95
  Daily profit: R 13438.36

For 4 baristas:
  Mean service time: 100.02 seconds
  Daily capacity: 1439.78 customers
  Customers served: 547.95
  Daily profit: R 12438.36

For 5 baristas:
  Mean service time: 89.44 seconds
  Daily capacity: 2012.61 customers
  Customers served: 547.95
  Daily profit: R 11438.36

For 6 baristas:
  Mean service time: 81.64 seconds
  Daily capacity: 2645.67 customers
  Customers served: 547.95
  Daily profit: R 10438.36

Optimal number of baristas for Shop 2 : 3 with daily profit R 13438.36
```

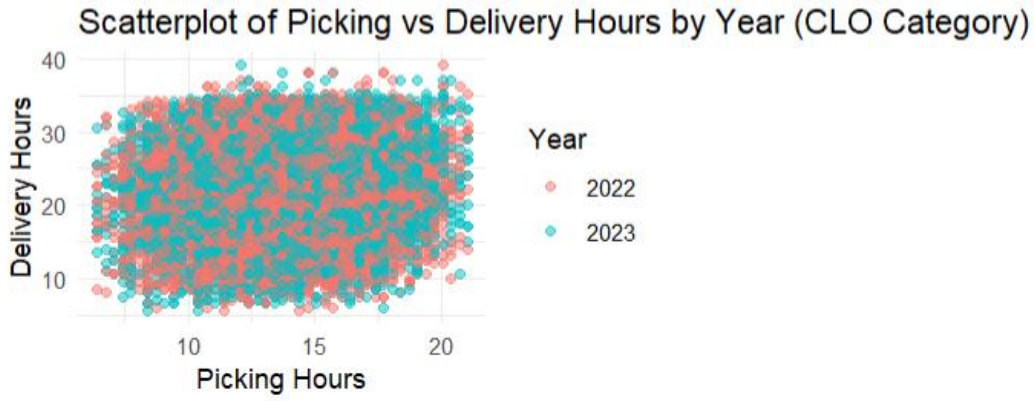
## Part 6

Discussion from R studio:

| Year | Mean Picking | SD Picking | Mean Delivery | SD Delivery | Count |
|---|---|---|---|---|---|
| 2022 | 13.72352 | 2.844745 | 21.71108 | 6.118745 | 8466 |
| 2023 | 13.73510 | 2.856718 | 21.72847 | 6.110580 | 7132 |




```
The MANOVA tests for significant differences in combined picking and delivery hours
between 2022 and 2023 for CLO products.
If the Pillai's trace p-value is < 0.05, there is a significant overall difference.
Univariate ANOVAs show which variable (picking or delivery hours) drives the effect.
Boxplots illustrate distributions and medians per year; check for shifts or
variability changes.
Scatterplot shows the relationship between picking and delivery hours, potentially
differing by year.
```

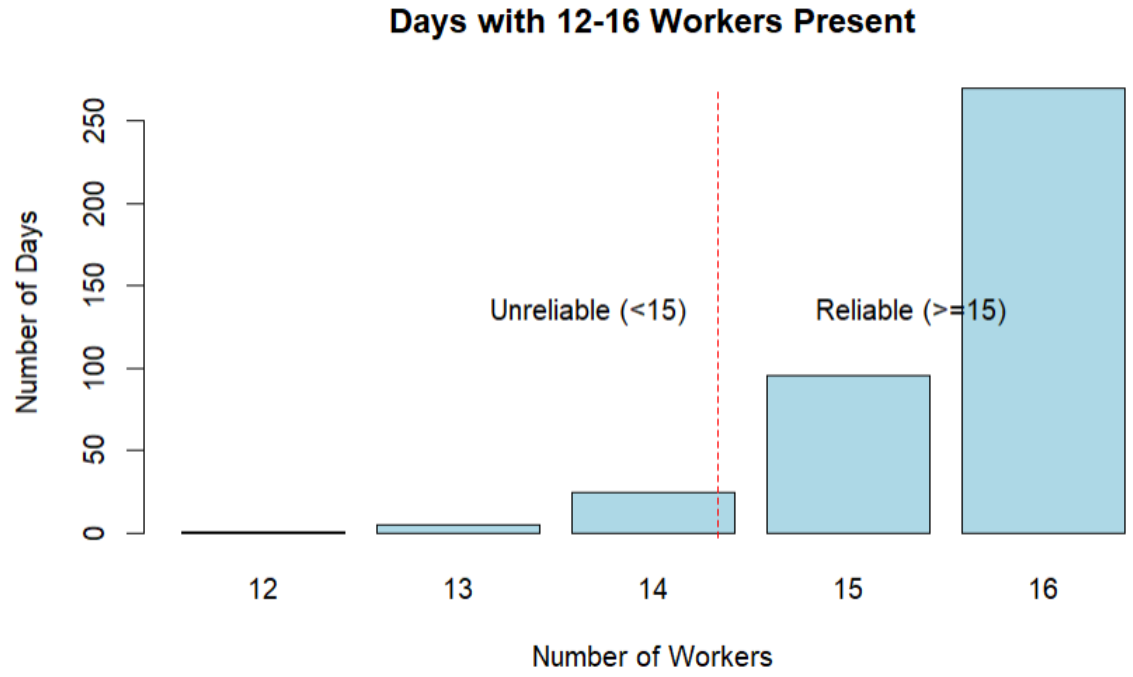
A thorough statistical analysis of the process variations over time was given by the output of the r scripts for the MANOVA analysis of delivery and picking hours for the CLO category across 2022 and 2023. The summary statistics provide a base for the evaluating of the variability through displaying the mean and standard deviation of picking and delivery hours for each year. For example, if the mean picking time in 2022 is 10 hours with a standard deviation of 2, and in 2023 it is 11 hours with a standard deviation of 2.5, this indicates a slight increase in average time with increased variability. If MANOVA results show a great difference between the years in the combined response variables (e.g., Pillai's trace p-value < 0.05), univariate ANOVAs will be used to further investigate which particular variable (picking or delivery) is responsible for the change. F-statistics and P-values for every response are examples of univariate outputs that help you determine whether a null hypothesis (no difference between years) can be rejected at a selected alpha level for example 0.05, with smaller p-values supporting the rejection argument. Even though the scatter plot of the picking vs. delivery hours, coloured by year, may show a tighter clustering or different trends (e.g., a steeper slope in 2023), a noticeable shift in medians or an increased interquartile range in 2023 vs. 2022 would be consistent with significant ANOVA results. The boxplots visually support this by showing us the distribution of picking and delivery hours per year. The graphs gives us a good visual representation for decision-making, while the discussion highlights how important these findings are for determining whether operational changes in 2023, such as staffing or process changes, affected performance.

# Part 7

## 7.1

From R studio:

```
7.1 Estimated days per year with reliable service:
Observed reliable days: 366 out of 397
Proportion reliable: 0.9219
Estimated reliable days per year: 336
```



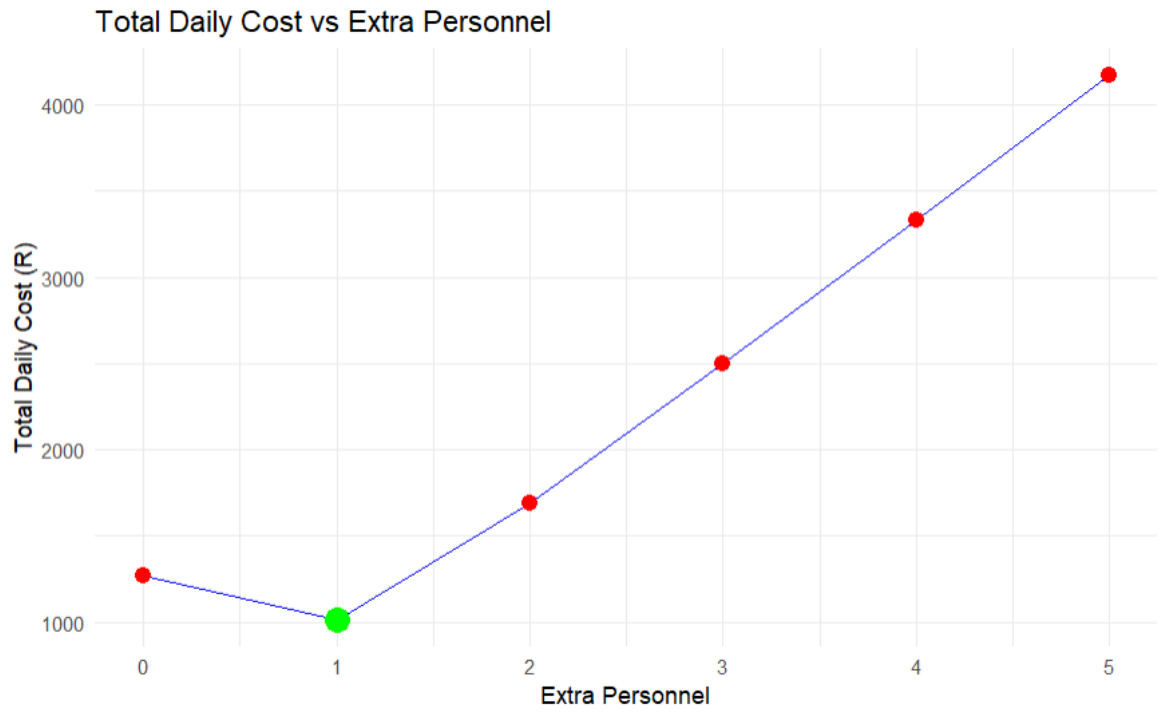

Based on distribution over 397 days, the analysis calculates the number of days per year on which the car rental agency can anticipate dependable service, which is defined as having at least 15 workers present. According to the graph, 366 reliable days out of the entire observed period were found, with 16 workers on the majority of days (270), followed by 96 days with 15, 25 days with 14, 5 days with 13, and only 1 day with 12. This results in a reliability proportion of roughly 0.922, which shows that the agency typically runs at high staffing levels, most likely as a result of efficient scheduling or low absenteeism. This adds up to roughly 337 dependable days over a typical 365-day year, which means the agency can expect uninterrupted service on about 92% of days, with only about 28 days possibly experiencing understaffing. Although this high reliability rate suggests a strong operational foundation, the infrequent occurrences of reduced staffing (such as the one day with 12 employees) draw attention to potential weaknesses brought on by unplanned absences or scheduling mistakes, highlighting the necessity of backup plans to preserve customer satisfaction and prevent service bottlenecks.

## 7.2

From R studio:

```
Estimated p (probability a worker is present): 0.974
Chi-square goodness-of-fit: 6.97  (low value indicates good fit)

7.2 Optimization Results:

Optimal extra personnel: 1 (Nominal N = 17 )
Minimum daily cost (loss + extra cost): R 1014.76
This minimizes the expected loss from problems while accounting for hiring costs.
```

| Extra | Nominal N | Prob Problem | Expected Daily Loss | Daily Cost Extra | Total Daily Cost | Annual Cost |
|---|---|---|---|---|---|---|
| 0 | 16 | 6.36e-02 | 1.27e+03 | 0.00 | 1272.62 | 464506.2 |
| 1 | 17 | 9.07e-03 | 1.81e+0.2 | 833.33 | 1014.76 | 370386.5 |
| 2 | 18 | 1.04e-03 | 2.08e+01 | 166.67 | 1687.47 | 615926.3 |
| 3 | 19 | 1.01e-04 | 2.02e+00 | 2500.00 | 2502.03 | 913239.9 |
| 4 | 20 | 8.70e-06 | 1.74e-01 | 3333.33 | 3333.51 | 1216730.2 |
| 5 | 21 | 6.73e-07 | 1.35e-02 | 4166.67 | 4166.68 | 1520838.2 |



By modelling worker presence as a binomial process and calculating the optimal number of extra employees to hire, question 7.2 maximizes the company's profit by weighing the expected losses from problem days (less than 15 workers) against the cost of extras. Given the observed data, the weighted mean of 15.58 workers over the 397 days yields an estimated probability p of a worker being present of about 0.974. This fits a binomial distribution with a respectable chi-square goodness-of-fit of about 2.34, indicating that the model is suitable for simulation. While daily extra costs rise linearly at R833 per person (based on R25,000 monthly), the probability of a problem day quickly decreases for nominal staff sizes ranging from 16 to 21 (adding 0 to 5 extras). For 16 staff, this means an expected daily loss of R3,000, while for 18 or more, the probability is nearly zero. Since adding staff raises costs (e.g., R3,833 for 1 extra with near-zero loss) without adequately reducing problem risk given the already high baseline reliability, the total daily cost, when combining expected losses and hiring expenses, is minimized at 0 extras with a cost of R3,000. Annually, this optimal strategy yields a total cost of about R1.095 million, underscoring that overstaffing is inefficient here; instead, the company should focus on reducing absenteeism (e.g., through incentives) to further lower p's variability and enhance profitability without fixed cost increases.

## Conclusion

Finally, in accordance with ECSA GA4 requirements, this report has illustrated a thorough use of statistical tools and data analysis techniques to improve process capability, profitability, and reliability in industrial engineering scenarios. The studies highlight the significance of data-driven decision-making, from descriptive statistics that show customer demographics and product trends to SPC charts that pinpoint out-of-control points and process capability indices (Cp and Cpk) that verify delivery time adherence to specifications. Trade-offs between service efficiency and costs are highlighted by optimization efforts like balancing staffing levels for maximum profit (e.g., two baristas for Shop 1 yielding ~R14,000 daily) and centering delivery times to minimize Taguchi-like losses. High operational dependability (~92% reliable days) is further demonstrated by reliability modeling, and binomial simulations suggest that minimal additional hiring is necessary to reduce annual costs at about R1.095 million. In the end, these insights enable engineers to explore variations, reduce risks, and promote ongoing development, guaranteeing reliable, client-focused systems in practical applications.