

**Applied Statistical Analysis**

**Predictive Modeling of TV Sales**

**Suliman S. Olayan School of Business - AUB**

**Fall Term 2023/2024**

**MSBA 310**

**Supervised by:**

**Dr. Imad Bou Hamad**

**Prepared by:**

**Christ Naim**

**Sasha Nasser**

**Romanos Rizk**

**Abstract:**

This report analyzes TV sales forecasting for wholesalers, using a dataset of 8,323 transactions from Linkers Group S.A.L. in Lebanon. It applies regression models and regression trees to identify factors influencing sales, such as unit price, size, and distribution channels. Our findings reveal that key factors such as TV specifications, unit price, and distribution channels (Casa) significantly influence consumer purchasing decisions, aiding in inventory management. This work highlights the importance of data-driven decision-making in the dynamic electronics market, offering strategic insights for wholesalers.

**Introduction & Literature Review:**

In this report, we delve into the complex and dynamic world of sales forecasting, with a specific focus on predicting TV sales in the wholesale market. In an industry characterized by rapid technological advancements and shifting consumer preferences, the ability to accurately forecast sales is not just beneficial but essential for effective inventory management, strategic planning, and maintaining competitive edge. Through a combination of theory and practical application, we aim to develop forecasting models that capture and adapt to the unpredictable nature of the electronics market. The goal is to provide wholesalers with a tool that enhances decision-making processes, optimizes inventory levels, and anticipates market trends, thereby contributing to operational efficiency and profitability.

The necessity for precise sales forecasting in the retail sector, particularly for products with brief lifecycles and limited historical data, is highlighted in four pivotal studies. These studies utilize advanced machine learning techniques, offering insights applicable to various domains, including TV sales forecasting.

In sales forecasting, the most recent study in 2023 by Seongbeom Hwang, Goonhu Yoon, Eunjung Baek, and Byoung-Ki Jeon, published in Electronics, delved into the challenges of forecasting sales for short-term mobile phone products. The research stood out for its evaluation of 12 machine learning models, notably Random Forest, XGBoost, and LSTM, with Random Forest emerging as the most effective. It demonstrated a MAPE of 42.6258, an RMSE of 8443.3328, and a correlation coefficient of 0.8629, underscoring its superior ability to capture sales patterns in a rapidly evolving market (Hwang et al., 2023).

In 2022, Rita Sleiman, Ahmad Mazyad, Moez Hamad, Kim-Phuc Tran, and Sébastien Thomassey's study in the International Journal of Computational Intelligence Systems introduced a novel forecasting framework for the fashion industry during the COVID-19 pandemic. Incorporating K-means clustering, Random Forest classification, and Polynomial Regression, the study adeptly predicted and adjusted sales patterns in response to the pandemic, demonstrating adaptability to changing consumer behaviors, especially during the lockdown period, by learning from online sales data (Sleiman et al., 2022).

In 2021, Daniel Büttner and Markus Rabe, in their contribution to the 9th International Conference on Traffic and Logistic Engineering, compared Time Series Analysis and Machine Learning methods in the electrical industry. They focused particularly on ARIMA and Random Forest models, highlighting Machine Learning's advantages for handling complex datasets and Time Series Analysis's suitability for simpler datasets. Their findings emphasized the importance of data quality and complexity in choosing the appropriate forecasting method (Büttner & Rabe, 2021).

Lastly, Qiuping Xu and Vikas Sharma, in a Springer International Publishing release in 2017, explored CPU sales forecasting at Intel Corporation. They employed a blend of multiple regressions, time series analysis, random forest, and boosting trees to create an ensemble model approach. This approach was noted for its effectiveness in capturing distinct sales characteristics across various business segments and was validated using weekly sales data, adapting to current market conditions with a weekly forecasting schema (Xu & Sharma, 2017).

Collectively, these studies highlight the necessity for dynamic and adaptive forecasting models in sectors with rapid product turnover and unpredictable market conditions. The integration of machine learning techniques and the focus on product-specific characteristics and market conditions are essential for developing sophisticated models in fields like TV sales forecasting. Understanding consumer behavior and sales trends in the dynamic consumer electronics market is crucial, as evidenced by these methodologies.

**Problem Description:**

Wholesale appliance companies are facing a major challenge in accurately forecasting the inventory required to meet the market demands. The root of the problem is the lack of direct access to customer data, which is only available to retailers. This data includes crucial insights into the market trends and the evolving television specifications that play a significant role in the purchasing decisions of the customers. Wholesale companies can only utilize order transactions from retailers, which provide limited details like pricing, quantity, and TV specifications. Due to the lack of direct access to granular customer data, companies are struggling to make accurate predictions, which can result in inventory imbalances such as shortages and overstocking of products. Additionally, the existing literature does not cover the business-to-business aspect of the supply chain and only focuses on the business-to-customer. This leaves a gap and a lack of guidance for wholesalers seeking effective strategies to navigate these challenges. It is crucial for the wholesale company to bridge this information gap and develop informed and responsive inventory management practices aligned with dynamic market trends and customer preferences.

**Data Description:**

The data used for the analysis consists of 8323 order transactions of televisions made by appliance retailers in Lebanon. The data was obtained from Linkers Group S.A.L., a highly recognized wholesales company in the household appliances and consumer electronics fields. Each order transaction in the data set details:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Total Quantity** per order transaction. | | | | | |
| **Size of the TV** (in inches):25,29,32,  40,41,43,45,50,  52,55, 58,65,75 | **Casa** order was delivered to:Beirut, Bekaa, Mount Lebanon, North, South | **Unit Price** of each TV (in $) | **Response Time**(time for pixel to change ):4,10 ms. | **Energy Efficiency** Level: A,B,C,D,E | **Month** the order was placed in |
| **Sound Quality**: Basic, Enhanced, Premium | **Refresh Rate** (Hz): 60,120. | **Pixel Density** (PPI) | **Connectivity Options** :2, 3, 4 ports | **Brand** of the TV:LG, Panasonic, Samsung, Sony, TCL | **Resolution**: HD, FHD, UHD |

**Methodology:**

Based on the problem and data descriptions, in order to properly forecast the demand for TVs based on customer preferences and changing market trends, certain statistical approaches need to be employed. We first started by setting the Total Quantity as our target variable. Based on that, we focused on analyzing the trends and patterns between the Quantity and the different factors. Next, we decided to run Regression Models to study the relationship between the different predictors and the Quantity target variable. Finally, Classification and Regression Trees (CART) were built in order to predict the outcomes of demand forecasts based on the predictors.

**Results and Discussion:**

1. **Exploratory Analysis:**

For this step, we used a scatter plot in order to visualize the trends for the Unit Price with the Total Quantity. For the rest of the predictors, we used side-by-side boxplots.

From comparing the boxplots, we can see that Casa, Size (Inches), Connectivity, Efficiency display a clear shift in the boxplots. This means that there is a relationship between these factors and the Total Quantity.

From comparing the boxplots, we can see that Sound Quality, Month, Response Time and Refresh Rate display a slight shift in the boxplots. This means that there might be a relationship between these factors and the Total Quantity.

From comparing the boxplots, we can see that Brand and Resolution do not display a clear shift in the boxplots. This means that no relationship exists between the factors and the Total Quantity.

**Total Quantity vs Unit Price (in $):** We notice that there is a negative linear relationship between both factors. The higher the price, the lower the quantity being ordered.

To sum up, these plots ultimately reveal that there is a clear association between the unit price of a TV and its ordered quantity. In Parallel, we can also see a pattern between the specification of a TV and its unit price as TVs with premium specifications end up being more expensive. Thus, this goes hand in hand with the observed graphs with premium TV ordered at a lesser rate than TVs with regular features and lower unit prices.

1. **Regression Models:-**

Before starting to build our regression models, we split our data into a Training dataset (70%) and a Validation dataset (30%) after setting a seed of 100 for replicability.

**Model 1:**

For model 1, all of the specifications that were detailed in the Data Description were used as predictors for the Total Quantity target variable. After running the regression on the training dataset, we obtained an R^2 of 75.01%. This means that 75% of the variations in the Total Quantity of TVs ordered is explained by the predictors of the model.

To evaluate the significance of the model:

H0: Bi = 0

H1: at least one Bi ≠ 0

p-value < 2.2e-16

F-statistic:442 on 40 and 5838 DF

=> Based on the F statistic: Reject H0, at least 1 predictor in Model 1 is statistically significant, Model 1 is statistically significant.

After that, we obtained the mean of the Total Quantity from the Validation data set yielding a value of 84.6248, which equates to around 85 TVs. We then calculated the error of the model and found an RMSE of 11.02. This means that the actual Total Quantities deviate from the estimated Total Quantities by around 11 TVs. Compared to the mean, the RMSE is acceptable enough to say that the model has an acceptable prediction accuracy.

From Model 1, based on a t test (0.688 and 1.180 respectively) and an alpha of 5%, we can see that Pixel Density and the Refresh Rate were non statistically significant predictors yielding a p-value higher than 0.05 (0.491663 and 0.238207 respectively). This implies that for further improvements in the model, these 2 predictors can be removed.

We can also notice that the Response Time yielded NA for the output of the regression. Upon further analysis, it is noticed that this was due to a perfect multicollinearity with another predictor. In this case, it would be the Refresh Rate as TVs with 120 Hz Refresh Rate all have a Response Time of 4ms and TVs with 60 Hz displays all have a Response Time of 10 ms.

**Model 2:**

For model 2, the same predictors as model 1 were included except for Pixel Density and the Refresh Rate . After running the regression on the training dataset, we obtained an R^2 of 75.01% and an RMSE of 11.02 which are similar to the values obtained in Model 1, indicating that the second model also has an acceptable prediction accuracy.

To evaluate the significance of the model:

H0: Bi = 0

H1: at least one Bi ≠ 0

p-value < 2.2e-16

F-statistic: 453.4 on 39 and 5839 DF

=> Based on the F Statistic: Reject H0, at least 1 predictor in Model 2 is statistically significant, Model 2 is statistically significant.

After further inspection of the predictors, we can notice that upon removing the Refresh Rate from the model, the Response Time has yielded results. However, based on the t-test (a value of -1.272) and an alpha of 5%, we can see that Response Time is non statistically significant with a high p-value of 0.203533 and can be removed from the model for improvements.

We then decided to test for multicollinearity in model 2. All of the factors have squared GVIF values that fall below 5, the acceptable range of threshold, which means that there are no instances of multicollinearity.

**Model 3:**

For model 3, the same predictors as model 2 were included except for Response Time . After running the regression on the training dataset, we obtained an R^2 of 75.01% and an RMSE of 11.03 which are similar to the values obtained in Model 1 and 2, indicating that the third model also has an acceptable prediction accuracy.

To evaluate the significance of the model:

H0: Bi = 0

H1: at least one Bi ≠ 0

p-value < 2.2e-16

F-statistic: 465.2 on 38 and 5840 DF

=> Based on the F Statistic: Reject H0, at least 1 predictor in Model 3 is statistically significant, Model 3 is statistically significant.

Upon further inspection of the predictors in model 3, we can see that all of them are statistically significant.

We then decided to test for multicollinearity in model 3. All of the factors have squared GVIF values that fall below 5, the acceptable range of threshold, which means that there are no instances of multicollinearity.

**Model 4:**

For model 4, we decided to include only the most significant predictors in the model. These are the Unit Price, the Size, and the Casa, which were identified by Trial and Error. After running the regression on the training dataset, we obtained an R^2 of 73.16% and an RMSE of 11.35 which are similar to the values obtained in Model 1,2, and 3, indicating that the fourth model also has an acceptable prediction accuracy.

To evaluate the significance of the model:

H0: Bi = 0

H1: at least one Bi ≠ 0

p-value < 2.2e-16

F-statistic: 1234 on 13 and 5865 DF

=> Based on the F Statistic: Reject H0, at least 1 predictor in Model 4 is statistically significant, Model 4 is statistically significant.

Upon further inspection of the predictors in model 4, we can see that the three predictors are all statistically significant.

We then decided to test for multicollinearity in model 4. All of the factors have squared GVIF values that fall below 5, the acceptable range of threshold, which means that there are no instances of multicollinearity.

To test for interaction, we ran a regression model with the predictors being the product of UnitPrice\*Size\*Casa. We notice that for some predictors interacting, their impact on total quantity variates. As an example, the interaction between unit price, inches (“75”), and Casa Mount Lebanon resulted in a positive and statistically significant coefficient. (p value of 2.47e-05 less than 0.05). Meaning that the effect of Unit Price on the dependent variable Total Quantity increases if the TV has a size of 17 inches and if the Casa is “Mount Lebanon”, or vice versa.

**Model Comparison:**

MAPE MAE RMSE numb\_predictors

perform1 10.52526 6.528354 11.02643 12

perform2 10.52570 6.529110 11.02625 10

perform3 10.52222 6.529426 11.03070 9

perform4 10.59465 6.566468 11.35494 3

Selecting the best model requires identifying the one with the least tradeoff between performance and complexity. As seen from the table comparing the accuracy of each model and their complexity levels, we notice that all the models have an approximately similar RMSE. However, model 4, with 3 predictors, is much less complex. Thus, we choose model 4 as our predictive model for total quantity.

1. **Regression Trees:**

Our second approach for predicting quantity was data driven, specifically using regression trees with total quantity as a target variable. Four regression trees were tested and validated on the same training and validation datasets used for the linear regression models.

**Tree 1:**

For our first regression tree: we used all the predictors and specified a low complexity parameter (cp) of 0.0001, a minimum bucket size of 50, a minimum split of 100 and a maximum depth of 10. The tree was then pruned based on the lowest cross sectional error.

Tree 1 resulted in a root node error of 488.38. The pruned tree was validated on the validation dataset and resulted in an RMSE of 10.12551 which is acceptable compared to the mean quantity of 84.6248 in the validation dataset. The tree had 63 nodes and 32 leafs.

**Tree 1.1:**

In an attempt to decrease the complexity of the tree, we pruned the tree based on a best pruning approach by checking if the addition of the cross sectional of the last node with its standard deviation is higher than the cross sectional error of the node above. This was possible because the xerror (0.21858) corresponding to the best cp (best cp = 0.0005137) plus its standard deviation (0.011727) was less than the xerror (0.230307) of the cp above (0.00052630) : 0.21858 + 0.011727=0.230307 > 0.21799 - We can consider cp=0.00052630.

Tree 1.1 resulted in an RMSE of 10.574883, 61 nodes, and 31 leafs.

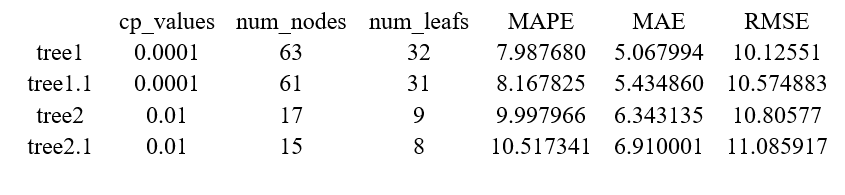
**Tree 2:**

To further decrease complexity, we decreased the cp from 0.0001 to 0.01. This will make it harder for the algorithm to add branches as additional splits would only be made if the improvement in fit (measured by the reduction in the sum of squared errors or another criterion) is above the cp threshold. The tree was then pruned based on the lowest cross sectional error and had an RMSE of 10.80577, 17 nodes, and 9 leafs.

**Tree 2.1:**

In an attempt to decrease the complexity even more, we applied the best pruning technique on tree 2.0 as explained in tree 1.1. This resulted in a tree that is less complex than tree 2.0 and with an RMSE of 11.085917, 15 nodes, and 8 leafs

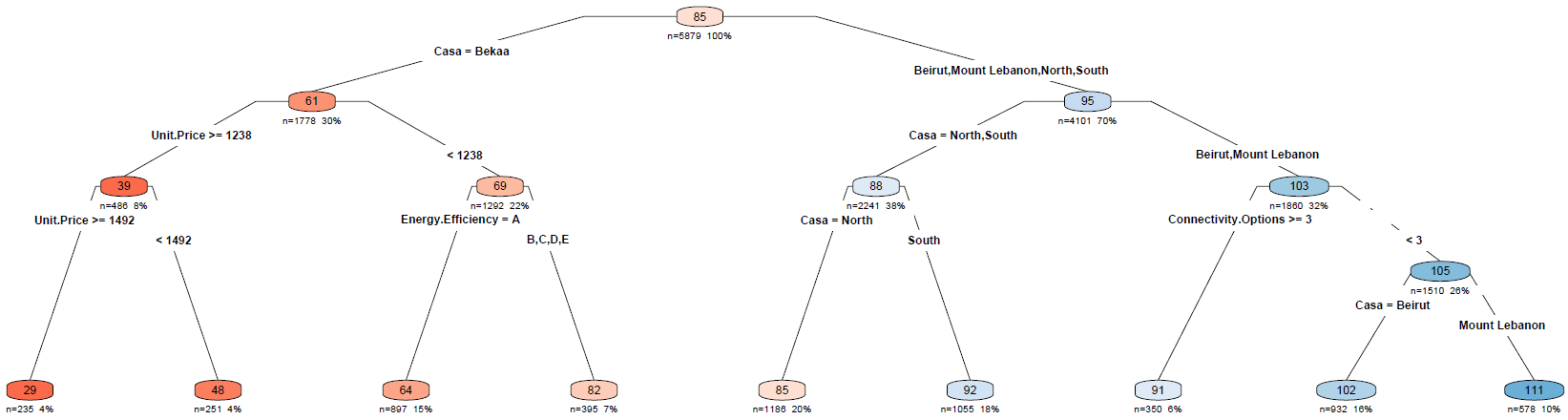
**Comparison Table for Trees:**



**Analysis:**

* Tree 2's cp is 0.01, higher than Tree 1 (0.0001), reducing the risk of overfitting and making it less complex. It has 17 nodes and 9 leaves, significantly fewer than Tree 1 (63 nodes, 32 leaves) and Tree 1.1 (61 nodes, 31 leaves), simplifying interpretation and analysis.
* The RMSE of 10.80577 is slightly higher than Tree 1 (10.12551) and Tree 1.1 (10.574883), but still indicates good predictive accuracy.
* Tree 2.1 has a higher RMSE (11.085917) and marginally fewer nodes and leaves, making Tree 2 preferable due to its better accuracy.

In conclusion, Tree 2 strikes an optimal balance, capturing essential data patterns without unnecessary complexity.



**Conclusion and Recommendations:**

This report, through its detailed analysis and application of statistical models, offers significant insights into the complexities of TV sales forecasting for wholesale appliance companies. Our comprehensive study, grounded in a robust dataset from Linkers Group S.A.L., underscores the pivotal role of accurate forecasting in aligning inventory management with the dynamic demands of the consumer electronics market.

Our findings reveal that key factors such as TV specifications, unit price, and distribution channels (Casa) significantly influence consumer purchasing decisions. The negative linear relationship between unit price and total quantity ordered particularly highlights the price sensitivity within the TV market. Moreover, the preference for TVs with specific features and specifications, as indicated by our analysis, provides a clear direction for wholesalers in stocking products that align with current market trends.

The regression models we developed demonstrate that a balanced approach between model complexity and predictive accuracy is crucial. Model 4, with its focus on unit price, size, and Casa, emerges as the optimal tool for forecasting, offering a streamlined yet effective solution for predicting demand. This model’s strength lies in its ability to capture essential market dynamics while maintaining operational simplicity.

In addition, the regression trees approach, particularly Tree 2, further reinforces the importance of a nuanced understanding of market trends. By capturing essential data patterns without overcomplicating the model, Tree 2 offers an alternative perspective, enriching our overall forecasting strategy.

As wholesalers in the rapidly evolving electronics market, embracing these analytical insights is key to developing more informed and responsive inventory management practices. This study provides a framework for wholesalers to adapt to shifting market conditions, optimize inventory levels, and ultimately enhance profitability.

In conclusion, our report not only addresses the immediate forecasting challenges faced by TV wholesalers but also lays the groundwork for a more data-driven, responsive approach to inventory management in the face of ever-changing consumer preferences and market conditions.

**Recommendations:**

1.The negative coefficient for Unit.Price indicates that price is a sensitive factor in sales quantity. It is recommended to review the pricing strategy to find the optimal price points that can maximize sales without sacrificing profit margins too much.

2. The negative coefficients for larger sizes (65 and 75 inches) that are statistically significant imply that these models may be priced too high or not in high demand. Consider either adjusting the price or focusing inventory on sizes that are more positively correlated with sales quantity.

3. Significant positive coefficients for regions like Beirut and Mount Lebanon indicate higher sales quantities compared to the reference region (Bekaa). This suggests that these regions may have higher demand or more effective distribution strategies. It would be beneficial to concentrate marketing efforts and distribution resources in these areas or replicate the successful strategies from these regions in other areas.

4. The tree shows a split at a unit price of 1238, indicating that price is a significant factor in sales volume. The wholesaler should consider segmenting their inventory to ensure they have a variety of options above and below this price point to cater to different market segments.

5. The wholesaler should adjust their supply strategy based on regional preferences, as indicated by the splits on 'Casa'. Certain regions like 'Bekaa' show different sales patterns compared to 'Beirut' and 'Mount Lebanon'. Inventory should be allocated and marketed differently in these regions to optimize sales.

6. There is a split for energy efficiency, with category 'A' being a decision node. The wholesaler should stock a higher proportion of 'A' energy efficiency-rated TVs, as there seems to be a distinct market for these models.

7. A split occurs where the number of connectivity options is greater than 3, suggesting that TVs with more connectivity features are preferred. The wholesaler should ensure that their inventory is stocked with TVs that offer a higher number of connectivity options to meet this demand.

**References:**

Büttner, D., & Rabe, M. (2021). Sales Forecasting in the Electrical Industry – An Illustrative Comparison of Time Series and Machine Learning Approaches. 2021 9th International Conference on Traffic and Logistic Engineering. <https://ieeexplore.ieee.org/document/9525747>

Hwang, S., Yoon, G., Baek, E., & Jeon, B.-K. (2023). A Sales Forecasting Model for New-Released and Short-Term Product: A Case Study of Mobile Phones. Electronics, 12(3256). <https://www.mdpi.com/2079-9292/12/15/3256>

Sleiman, R., Mazyad, A., Hamad, M., Tran, K.-P., & Thomassey, S. (2022). Forecasting Sales Profiles of Products in an Exceptional Context: COVID-19 Pandemic. International Journal of Computational Intelligence Systems, 15(99). <https://link.springer.com/article/10.1007/s44196-022-00161-x>

Xu, Q., & Sharma, V. (2017). Ensemble Sales Forecasting Study in Semiconductor Industry. Springer International Publishing. <https://link.springer.com/content/pdf/10.1007/978-3-319-62701-4_3.pdf>

**Appendix:**

**Exploratory Analysis:**

**#Read the data:**

df\_tv=read.csv("C:\\Users\\Lenovo\\OneDrive\\01\_Education\\02\_AUB\\01\_Fall2023\\MSBA310-Applied Statistical Analysis\\MainProject\\TV Dataset 3-Manual.csv")

attach(df\_tv)

**#Adjust data types:**

df\_tv$Inches <- as.factor(df\_tv$Inches)

df\_tv$Resolution <- as.factor(df\_tv$Resolution)

df\_tv$Smart <- as.factor(df\_tv$Smart)

df\_tv$Brand <- as.factor(df\_tv$Brand)

df\_tv$Revenue <- as.numeric(df\_tv$Revenue)

df\_tv$Refresh.Rate <- as.factor(df\_tv$Refresh.Rate)

df\_tv$Sound.Quality <- as.factor(df\_tv$Sound.Quality)

df\_tv$Connectivity.Options <- as.factor(df\_tv$Connectivity.Options)

df\_tv$Energy.Efficiency <- as.factor(df\_tv$Energy.Efficiency)

df\_tv$Response.Time..ms. <- as.factor(df\_tv$Response.Time..ms.)

df\_tv$Month <- as.factor(df\_tv$Month)

df\_tv$Casa <- as.factor(df\_tv$Casa)

library(ggplot2)

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#Base R:**

**#boxplot(Tot.Qty~Casa,col="blue")**

**#Casa and Q, without legend:**

ggplot(df\_tv, aes(x = Casa, y = Tot.Qty, fill = Casa)) +

geom\_boxplot() +

scale\_fill\_manual(values = rep("blue", nlevels(df\_tv$Casa))) +

labs(title = "Total Quantity of TVs Ordered by Casa", y = "Total Quantity") +

theme\_light() +

theme(

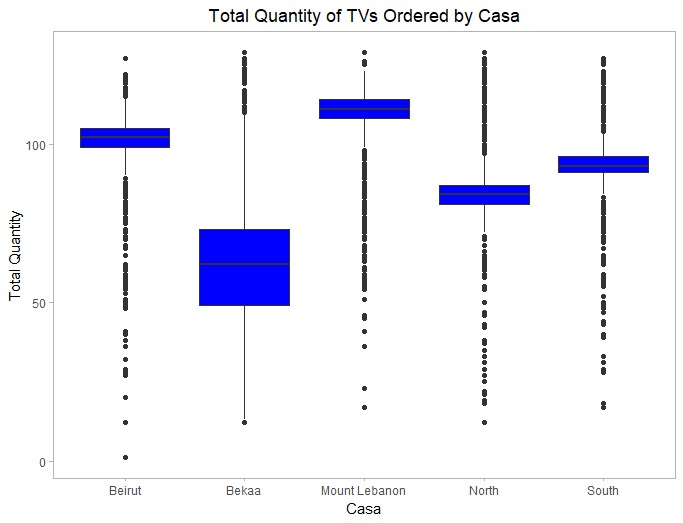
panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

plot.title = element\_text(hjust = 0.5),

legend.position = "none" # Removes the legend

)



~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#Brand and Q:**

**#boxplot(Tot.Qty~Brand,col="blue")**

**#Brand and Q:**

ggplot(df\_tv, aes(x = Brand, y = Tot.Qty, fill = Brand)) +

geom\_boxplot() +

scale\_fill\_manual(values = rep("blue", nlevels(df\_tv$Brand))) +

labs(title = "Total Quantity of TVs Ordered by Brand", y = "Total Quantity") +

theme\_light() +

xlab("Brand") +

theme(

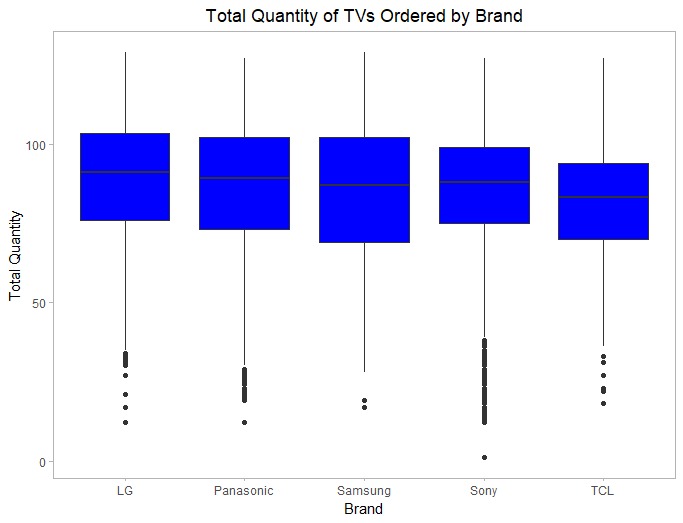
panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

plot.title = element\_text(hjust = 0.5),

legend.position = "none" # Removes the legend

)



~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#Inches and Q:**

**#boxplot(Tot.Qty~Inches,col="blue")**

**#Inches by Q**

ggplot(df\_tv, aes(x = Inches, y = Tot.Qty, fill = Inches)) +

geom\_boxplot() +

scale\_fill\_manual(values = rep("blue", nlevels(df\_tv$Inches))) +

labs(title = "Total Quantity of TVs Ordered by Size", y = "Total Quantity") +

theme\_light() +

xlab("Inches") +

theme(

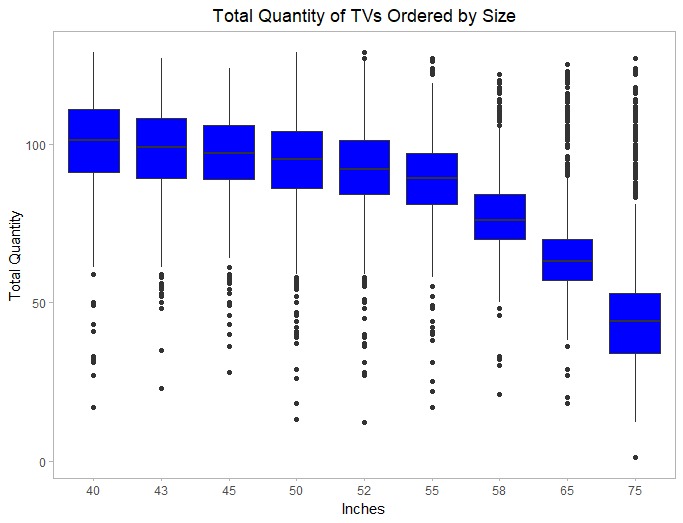
panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

plot.title = element\_text(hjust = 0.5),

legend.position = "none" # Removes the legend

)



~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#Resolution and Q:**

**#boxplot(Tot.Qty~Resolution,col="blue")**

**#Resolution and Q:**

ggplot(df\_tv, aes(x = Resolution, y = Tot.Qty, fill = Resolution)) +

geom\_boxplot() +

scale\_fill\_manual(values = rep("blue", nlevels(df\_tv$Resolution))) +

labs(title = "Total Quantity of TVs Ordered by Resolution", y = "Total Quantity") +

theme\_light() +

xlab("Resolution") +

theme(

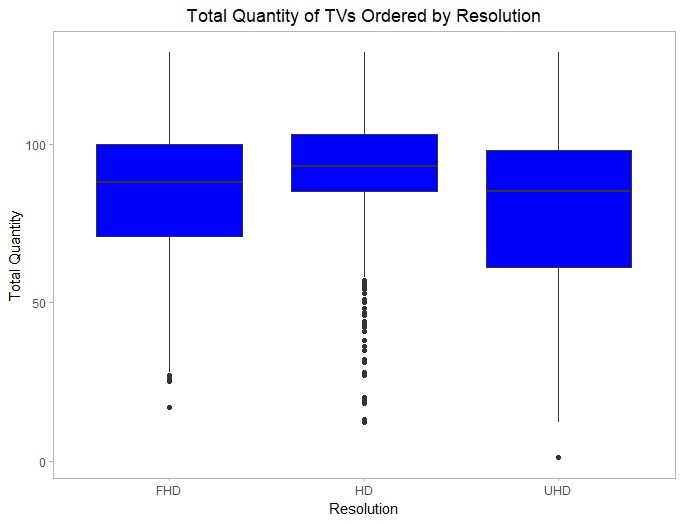
panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

plot.title = element\_text(hjust = 0.5),

legend.position = "none" # Removes the legend

)



~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#Sound Quality by Q:**

**#boxplot(Tot.Qty~Sound.Quality,col="blue")**

**#Sound Quality by Q:**

ggplot(df\_tv, aes(x = Sound.Quality, y = Tot.Qty, fill = Sound.Quality)) +

geom\_boxplot() +

scale\_fill\_manual(values = rep("blue", nlevels(df\_tv$Sound.Quality))) +

labs(title = "Total Quantity of TVs Ordered by Sound Quality", y = "Total Quantity") +

theme\_light() +

xlab("Sound Quality") +

theme(

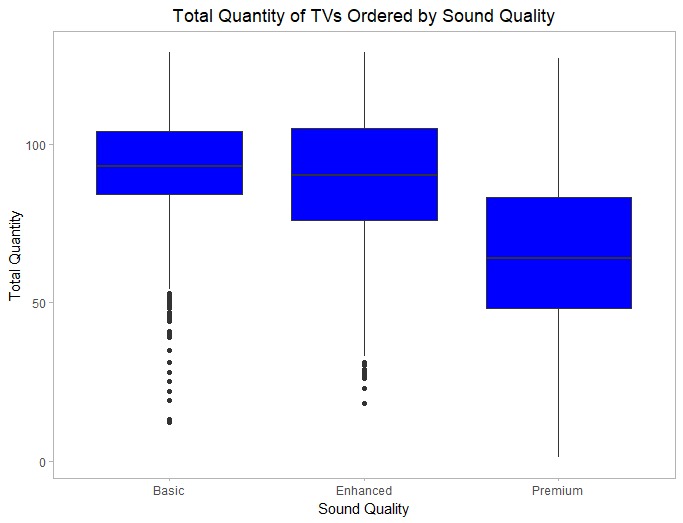
panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

plot.title = element\_text(hjust = 0.5),

legend.position = "none" # Removes the legend

)



~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#Connectivity by Q:**

**#boxplot(Tot.Qty~Connectivity.Options,col="blue")**

**#Connectivity by Q:**

ggplot(df\_tv, aes(x = Connectivity.Options, y = Tot.Qty, fill = Connectivity.Options)) +

geom\_boxplot() +

scale\_fill\_manual(values = rep("blue", nlevels(df\_tv$Connectivity.Options))) +

labs(title = "Total Quantity of TVs Ordered by Number of Ports", y = "Total Quantity") +

theme\_light() +

xlab("Connectivity Options") +

theme(

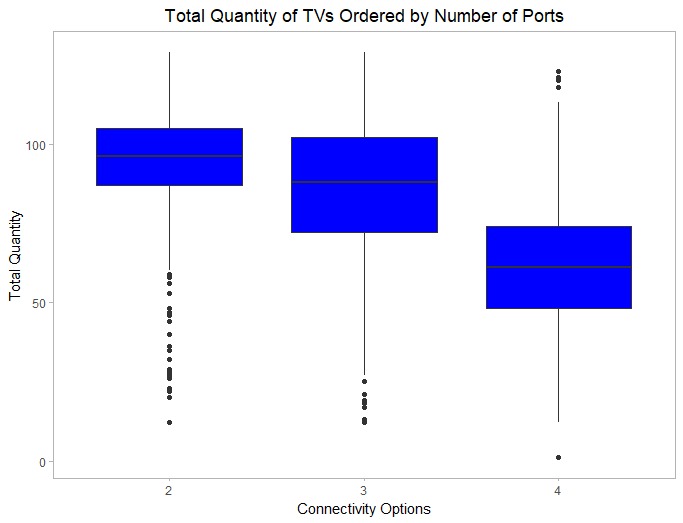
panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

plot.title = element\_text(hjust = 0.5),

legend.position = "none" # Removes the legend

)



~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#Efficiency by Q:**

**#boxplot(Tot.Qty~Energy.Efficiency,col="blue")**

**#Efficiency by Q:**

ggplot(df\_tv, aes(x = Energy.Efficiency, y = Tot.Qty, fill = Energy.Efficiency)) +

geom\_boxplot() +

scale\_fill\_manual(values = rep("blue", nlevels(df\_tv$Energy.Efficiency))) +

labs(title = "Total Quantity of TVs Ordered by Energy Efficiency", y = "Total Quantity") +

xlab("Energy Efficiency") +

theme\_light() +

theme(

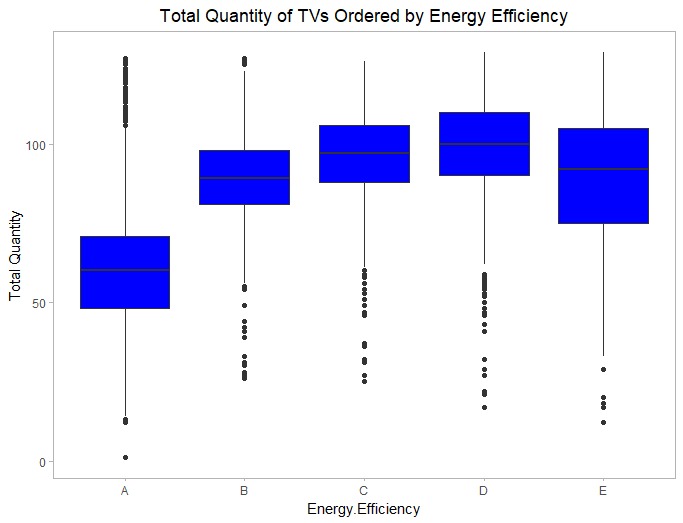
panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

plot.title = element\_text(hjust = 0.5),

legend.position = "none" # Removes the legend

)

****

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#Month by Q:**

**#boxplot(Tot.Qty~Month,col="blue")**

**#Month by Q:**

ggplot(df\_tv, aes(x = as.factor(Month), y = Tot.Qty, fill = Month)) +

geom\_boxplot() +

scale\_fill\_manual(values = rep("blue", nlevels(df\_tv$Month))) +

labs(title = "Total Quantity of TVs Ordered by Month", y = "Total Quantity") +

theme\_light() +

xlab("Month") +

theme(

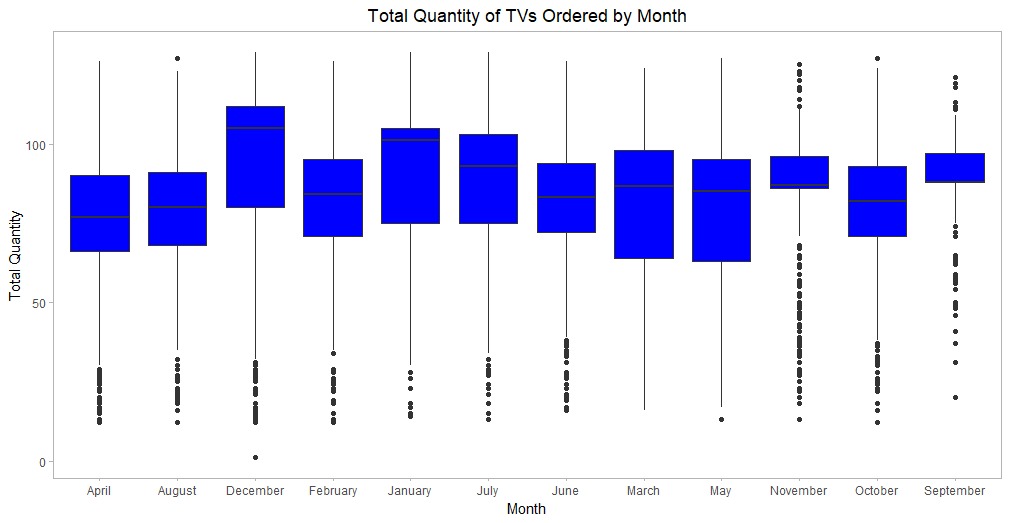
panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

plot.title = element\_text(hjust = 0.5),

legend.position = "none" # Removes the legend

)

****

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#Response Time by Q:**

**#boxplot(Tot.Qty~Response.Time..ms.,col="blue")**

**#Response Time by Q:**

ggplot(df\_tv, aes(x = as.factor(Response.Time..ms.), y = Tot.Qty, fill = Response.Time..ms.)) +

geom\_boxplot() +

scale\_fill\_manual(values = rep("blue", nlevels(df\_tv$Response.Time..ms.))) +

labs(title = "Total Quantity of TVs Ordered by Response Time", y = "Total Quantity") +

theme\_light() +

xlab("Response Time") +

theme(

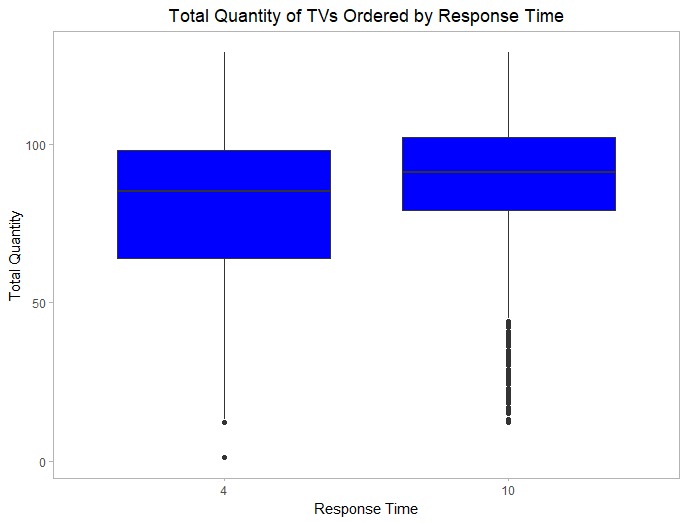
panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

plot.title = element\_text(hjust = 0.5),

legend.position = "none" # Removes the legend

)



~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#Refresh Rate by Q:**

**#boxplot(Tot.Qty~Refresh.Rate,col="blue")**

**#Refresh Rate by Q:**

ggplot(df\_tv, aes(x = as.factor(Refresh.Rate), y = Tot.Qty, fill = Refresh.Rate)) +

geom\_boxplot() +

scale\_fill\_manual(values = rep("blue", nlevels(df\_tv$Refresh.Rate))) +

labs(title = "Total Quantity of TVs Ordered by Refresh Rate", y = "Total Quantity") +

theme\_light() +

xlab("Refresh Rate") +

theme(

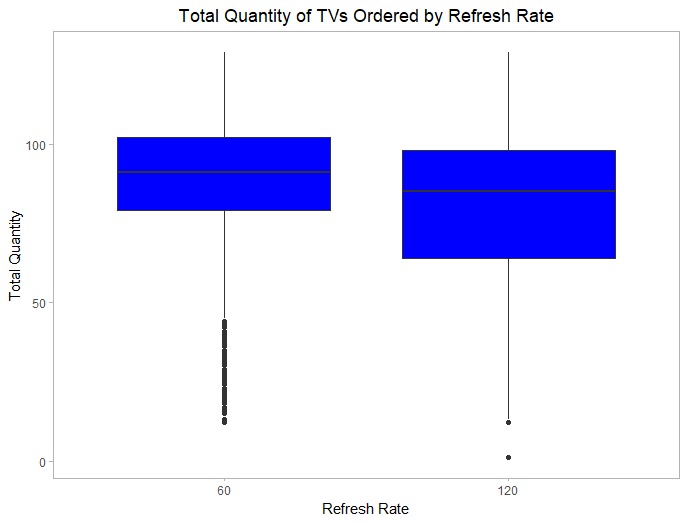
panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

plot.title = element\_text(hjust = 0.5),

legend.position = "none" # Removes the legend

)

****

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#Unit Price and Q:**

**#plot(Tot.Qty~Unit.Price)**

**#Unit Price and Q:**

ggplot(df\_tv, aes(x = Unit.Price, y = Tot.Qty)) +

geom\_point(color = "blue", alpha = 0.7) +

labs(title = "Total Quantity and Unit Price", x = "Unit Price", y = "Total Quantity") +

theme\_minimal() +

theme(

plot.title = element\_text(hjust = 0.5) # Center the title

)



~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**Regression Models:**

**#Adjust data types:**

df\_tv$Inches <- as.factor(df\_tv$Inches)

df\_tv$Resolution <- as.factor(df\_tv$Resolution)

df\_tv$Smart <- as.factor(df\_tv$Smart)

df\_tv$Brand <- as.factor(df\_tv$Brand)

df\_tv$Revenue <- as.numeric(df\_tv$Revenue)

df\_tv$Refresh.Rate <- as.factor(df\_tv$Refresh.Rate)

df\_tv$Sound.Quality <- as.factor(df\_tv$Sound.Quality)

df\_tv$Connectivity.Options <- as.factor(df\_tv$Connectivity.Options)

df\_tv$Energy.Efficiency <- as.factor(df\_tv$Energy.Efficiency)

df\_tv$Response.Time..ms. <- as.factor(df\_tv$Response.Time..ms.)

df\_tv$Month <- as.factor(df\_tv$Month)

df\_tv$Casa <- as.factor(df\_tv$Casa)

**#Splitting the data for training and validation:**

set.seed(100)

split=sample(1:2, nrow(df\_tv), replace = TRUE, prob=c(0.7, 0.3))

train=df\_tv[split==1, ]

val=df\_tv[split==2, ]

**#Inserting the pred function for calculating RMSE:**

pred\_error<- function(actual,pred){

mape <- mean(abs((actual - pred)/actual))\*100

mae=mean(abs(actual-pred))

RMSE= sqrt(mean((actual-pred)^2))

vec=c(mape,mae, RMSE)

names(vec)= c("MAPE", "MAE", "RMSE")

return(vec)

}

**#Model 1: All Predictors**

model1=lm(Tot.Qty~ Pixel.Density..PPI.+Unit.Price +relevel(as.factor(Inches),ref="52")+ relevel(as.factor(Resolution),ref="UHD")+relevel(as.factor(Brand),ref="LG")+relevel(as.factor(Casa),ref="Bekaa")+Refresh.Rate+relevel(as.factor(Sound.Quality),ref="Basic")+relevel(as.factor(Connectivity.Options),ref = '2')+relevel(as.factor(Energy.Efficiency),ref = "A")+Response.Time..ms.+relevel(as.factor(Month),ref = "June"),data=train)

summary(model1)

Residuals:

Min 1Q Median 3Q Max

-74.152 -3.679 0.156 4.006 58.737

Coefficients: (1 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) 8.259e+01 1.235e+00 66.861 < 2e-16 \*\*\*

Pixel.Density..PPI. 3.482e-03 5.063e-03 0.688 0.491663

Unit.Price -1.130e-02 6.981e-04 -16.184 < 2e-16 \*\*\*

relevel(as.factor(Inches), ref = "52")40 -1.003e-01 8.894e-01 -0.113 0.910209

relevel(as.factor(Inches), ref = "52")43 -6.706e-01 8.056e-01 -0.832 0.405253

relevel(as.factor(Inches), ref = "52")45 -9.043e-01 8.086e-01 -1.118 0.263476

relevel(as.factor(Inches), ref = "52")50 1.855e-01 7.356e-01 0.252 0.800962

relevel(as.factor(Inches), ref = "52")55 -8.288e-02 6.345e-01 -0.131 0.896075

relevel(as.factor(Inches), ref = "52")58 1.865e+00 7.740e-01 2.409 0.016037 \*

relevel(as.factor(Inches), ref = "52")65 -1.750e+00 8.868e-01 -1.973 0.048551 \*

relevel(as.factor(Inches), ref = "52")75 -1.399e+01 9.541e-01 -14.665 < 2e-16 \*\*\*

relevel(as.factor(Resolution), ref = "UHD")FHD 7.169e-01 3.419e-01 2.097 0.036036 \*

relevel(as.factor(Resolution), ref = "UHD")HD 5.974e-01 3.838e-01 1.557 0.119642

relevel(as.factor(Brand), ref = "LG")Panasonic -1.702e+00 4.360e-01 -3.904 9.55e-05 \*\*\*

relevel(as.factor(Brand), ref = "LG")Samsung 9.395e-02 5.055e-01 0.186 0.852567

relevel(as.factor(Brand), ref = "LG")Sony -2.242e+00 4.600e-01 -4.874 1.12e-06 \*\*\*

relevel(as.factor(Brand), ref = "LG")TCL 1.194e+00 6.048e-01 1.974 0.048481 \*

relevel(as.factor(Casa), ref = "Bekaa")Beirut 1.736e+01 6.222e-01 27.902 < 2e-16 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 2.388e+01 6.965e-01 34.286 < 2e-16 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")North 4.654e+00 5.890e-01 7.901 3.28e-15 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")South 1.115e+01 6.157e-01 18.101 < 2e-16 \*\*\*

Refresh.Rate120 3.756e-01 3.184e-01 1.180 0.238207

relevel(as.factor(Sound.Quality), ref = "Basic")Enhanced -2.739e+00 6.989e-01 -3.919 9.00e-05 \*\*\*

relevel(as.factor(Sound.Quality), ref = "Basic")Premium -4.275e+00 4.721e-01 -9.056 < 2e-16 \*\*\*

relevel(as.factor(Connectivity.Options), ref = "2")3 1.932e+00 5.522e-01 3.499 0.000471 \*\*\*

relevel(as.factor(Connectivity.Options), ref = "2")4 -6.028e+00 7.086e-01 -8.507 < 2e-16 \*\*\*

relevel(as.factor(Energy.Efficiency), ref = "A")B 4.971e+00 6.845e-01 7.262 4.30e-13 \*\*\*

relevel(as.factor(Energy.Efficiency), ref = "A")C 5.273e+00 7.861e-01 6.709 2.15e-11 \*\*\*

relevel(as.factor(Energy.Efficiency), ref = "A")D 4.434e+00 8.533e-01 5.196 2.11e-07 \*\*\*

relevel(as.factor(Energy.Efficiency), ref = "A")E -7.087e-01 9.802e-01 -0.723 0.469701

Response.Time..ms.10 NA NA NA NA

relevel(as.factor(Month), ref = "June")April 1.514e-01 7.420e-01 0.204 0.838363

relevel(as.factor(Month), ref = "June")August -1.396e+00 7.976e-01 -1.750 0.080121 .

relevel(as.factor(Month), ref = "June")December 2.870e+00 6.975e-01 4.115 3.93e-05 \*\*\*

relevel(as.factor(Month), ref = "June")February 1.004e+00 7.986e-01 1.257 0.208674

relevel(as.factor(Month), ref = "June")January 2.899e+00 7.172e-01 4.042 5.37e-05 \*\*\*

relevel(as.factor(Month), ref = "June")July 1.283e+00 7.653e-01 1.677 0.093680 .

relevel(as.factor(Month), ref = "June")March -3.634e-01 7.413e-01 -0.490 0.624008

relevel(as.factor(Month), ref = "June")May -5.917e-01 8.048e-01 -0.735 0.462230

relevel(as.factor(Month), ref = "June")November 7.645e-01 8.302e-01 0.921 0.357165

relevel(as.factor(Month), ref = "June")October 6.754e-01 7.919e-01 0.853 0.393767

relevel(as.factor(Month), ref = "June")September 1.755e+00 8.419e-01 2.085 0.037109 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.05 on 5838 degrees of freedom

Multiple R-squared: 0.7518, Adjusted R-squared: 0.7501

F-statistic: 442 on 40 and 5838 DF, p-value: < 2.2e-16

**#Model 1 RMSE:**

pred1=predict(model1,val)

perform1=pred\_error(val$Tot.Qty, pred1)

perform1

MAPE MAE RMSE

10.525263 6.528354 11.026433

**#Mean**

mean(val$Tot.Qty)

84.6248

**#Model 1 Complexity:**

12

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#Model 2: Remove: PPI and Refresh Rate**

model2=lm(Tot.Qty~Unit.Price +relevel(as.factor(Inches),ref="52")+ relevel(as.factor(Resolution),ref="UHD")+relevel(as.factor(Brand),ref="LG")+relevel(as.factor(Casa),ref="Bekaa")+relevel(as.factor(Sound.Quality),ref="Basic")+relevel(as.factor(Connectivity.Options),ref = '2')+relevel(as.factor(Energy.Efficiency),ref = "A")+Response.Time..ms.+relevel(as.factor(Month),ref = "June"),data=train)

summary(model2)

Residuals:

Min 1Q Median 3Q Max

-74.060 -3.672 0.148 4.008 58.652

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 8.318e+01 1.220e+00 68.157 < 2e-16 \*\*\*

Unit.Price -1.130e-02 6.981e-04 -16.185 < 2e-16 \*\*\*

relevel(as.factor(Inches), ref = "52")40 -9.122e-02 8.892e-01 -0.103 0.918302

relevel(as.factor(Inches), ref = "52")43 -6.737e-01 8.056e-01 -0.836 0.403009

relevel(as.factor(Inches), ref = "52")45 -9.111e-01 8.085e-01 -1.127 0.259809

relevel(as.factor(Inches), ref = "52")50 1.867e-01 7.355e-01 0.254 0.799591

relevel(as.factor(Inches), ref = "52")55 -6.768e-02 6.341e-01 -0.107 0.915005

relevel(as.factor(Inches), ref = "52")58 1.874e+00 7.739e-01 2.421 0.015509 \*

relevel(as.factor(Inches), ref = "52")65 -1.742e+00 8.867e-01 -1.964 0.049540 \*

relevel(as.factor(Inches), ref = "52")75 -1.399e+01 9.541e-01 -14.665 < 2e-16 \*\*\*

relevel(as.factor(Resolution), ref = "UHD")FHD 7.192e-01 3.418e-01 2.104 0.035436 \*

relevel(as.factor(Resolution), ref = "UHD")HD 5.986e-01 3.838e-01 1.560 0.118905

relevel(as.factor(Brand), ref = "LG")Panasonic -1.705e+00 4.360e-01 -3.910 9.34e-05 \*\*\*

relevel(as.factor(Brand), ref = "LG")Samsung 9.416e-02 5.055e-01 0.186 0.852235

relevel(as.factor(Brand), ref = "LG")Sony -2.249e+00 4.598e-01 -4.892 1.03e-06 \*\*\*

relevel(as.factor(Brand), ref = "LG")TCL 1.204e+00 6.046e-01 1.991 0.046537 \*

relevel(as.factor(Casa), ref = "Bekaa")Beirut 1.736e+01 6.222e-01 27.907 < 2e-16 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 2.389e+01 6.965e-01 34.298 < 2e-16 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")North 4.658e+00 5.890e-01 7.908 3.10e-15 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")South 1.115e+01 6.157e-01 18.105 < 2e-16 \*\*\*

relevel(as.factor(Sound.Quality), ref = "Basic")Enhanced -2.741e+00 6.989e-01 -3.922 8.89e-05 \*\*\*

relevel(as.factor(Sound.Quality), ref = "Basic")Premium -4.267e+00 4.719e-01 -9.042 < 2e-16 \*\*\*

relevel(as.factor(Connectivity.Options), ref = "2")3 1.955e+00 5.512e-01 3.546 0.000394 \*\*\*

relevel(as.factor(Connectivity.Options), ref = "2")4 -6.015e+00 7.083e-01 -8.492 < 2e-16 \*\*\*

relevel(as.factor(Energy.Efficiency), ref = "A")B 4.961e+00 6.843e-01 7.250 4.72e-13 \*\*\*

relevel(as.factor(Energy.Efficiency), ref = "A")C 5.265e+00 7.860e-01 6.699 2.29e-11 \*\*\*

relevel(as.factor(Energy.Efficiency), ref = "A")D 4.415e+00 8.528e-01 5.176 2.34e-07 \*\*\*

relevel(as.factor(Energy.Efficiency), ref = "A")E -7.112e-01 9.801e-01 -0.726 0.468079

Response.Time..ms.10 -4.020e-01 3.161e-01 -1.272 0.203533

relevel(as.factor(Month), ref = "June")April 1.445e-01 7.419e-01 0.195 0.845579

relevel(as.factor(Month), ref = "June")August -1.402e+00 7.975e-01 -1.758 0.078733 .

relevel(as.factor(Month), ref = "June")December 2.864e+00 6.974e-01 4.106 4.08e-05 \*\*\*

relevel(as.factor(Month), ref = "June")February 1.002e+00 7.985e-01 1.254 0.209776

relevel(as.factor(Month), ref = "June")January 2.891e+00 7.170e-01 4.032 5.59e-05 \*\*\*

relevel(as.factor(Month), ref = "June")July 1.288e+00 7.652e-01 1.684 0.092301 .

relevel(as.factor(Month), ref = "June")March -3.646e-01 7.413e-01 -0.492 0.622828

relevel(as.factor(Month), ref = "June")May -5.972e-01 8.047e-01 -0.742 0.458004

relevel(as.factor(Month), ref = "June")November 7.546e-01 8.300e-01 0.909 0.363305

relevel(as.factor(Month), ref = "June")October 6.726e-01 7.918e-01 0.849 0.395687

relevel(as.factor(Month), ref = "June")September 1.751e+00 8.419e-01 2.080 0.037562 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.05 on 5839 degrees of freedom

Multiple R-squared: 0.7518, Adjusted R-squared: 0.7501

F-statistic: 453.4 on 39 and 5839 DF, p-value: < 2.2e-16

**#Model 2 RMSE:**

pred2=predict(model2,val)

perform2=pred\_error(val$Tot.Qty, pred2)

perform2

MAPE MAE RMSE

10.52570 6.52911 11.02625

**#Model 2 Complexity:**

10

**#Multicollinearity:**

library(car)

vif(model2)

vif2Squarred <- vif(model2)

vif2Squarred[, ncol(vif2Squarred)] <- vif2Squarred[, ncol(vif2Squarred)]^2

print(vif2Squarred)

GVIF Df GVIF^(1/(2\*Df))^2

Unit.Price 3.627455 1 3.627455

relevel(as.factor(Inches), ref = "52") 38.306183 8 1.577279

relevel(as.factor(Resolution), ref = "UHD") 1.153200 2 1.073872

relevel(as.factor(Brand), ref = "LG") 2.211849 4 1.219520

relevel(as.factor(Casa), ref = "Bekaa") 5.063627 4 1.500083

relevel(as.factor(Sound.Quality), ref = "Basic") 2.730571 2 1.652444

relevel(as.factor(Connectivity.Options), ref = "2") 6.505314 2 2.550552

relevel(as.factor(Energy.Efficiency), ref = "A") 26.482235 4 2.268499

Response.Time..ms. 1.144442 1 1.144442

relevel(as.factor(Month), ref = "June") 1.969433 11 1.063551

**#Model 3: Remove response time:**

model3=lm(Tot.Qty~Unit.Price +relevel(as.factor(Inches),ref="52")+ relevel(as.factor(Resolution),ref="UHD")+relevel(as.factor(Brand),ref="LG")+relevel(as.factor(Casa),ref="Bekaa")+relevel(as.factor(Sound.Quality),ref="Basic")+relevel(as.factor(Connectivity.Options),ref = '2')+relevel(as.factor(Energy.Efficiency),ref = "A")+relevel(as.factor(Month),ref = "June"),data=train)

summary(model3)

Residuals:

Min 1Q Median 3Q Max

-73.705 -3.645 0.108 4.035 58.804

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 8.291e+01 1.201e+00 69.030 < 2e-16 \*\*\*

Unit.Price -1.130e-02 6.981e-04 -16.181 < 2e-16 \*\*\*

relevel(as.factor(Inches), ref = "52")40 -1.048e-01 8.892e-01 -0.118 0.906192

relevel(as.factor(Inches), ref = "52")43 -6.750e-01 8.056e-01 -0.838 0.402172

relevel(as.factor(Inches), ref = "52")45 -9.232e-01 8.085e-01 -1.142 0.253549

relevel(as.factor(Inches), ref = "52")50 1.880e-01 7.356e-01 0.256 0.798285

relevel(as.factor(Inches), ref = "52")55 -5.742e-02 6.341e-01 -0.091 0.927842

relevel(as.factor(Inches), ref = "52")58 1.901e+00 7.736e-01 2.458 0.014017 \*

relevel(as.factor(Inches), ref = "52")65 -1.714e+00 8.864e-01 -1.934 0.053168 .

relevel(as.factor(Inches), ref = "52")75 -1.397e+01 9.539e-01 -14.640 < 2e-16 \*\*\*

relevel(as.factor(Resolution), ref = "UHD")FHD 7.120e-01 3.418e-01 2.083 0.037299 \*

relevel(as.factor(Resolution), ref = "UHD")HD 6.061e-01 3.838e-01 1.579 0.114349

relevel(as.factor(Brand), ref = "LG")Panasonic -1.709e+00 4.360e-01 -3.919 9.00e-05 \*\*\*

relevel(as.factor(Brand), ref = "LG")Samsung 9.722e-02 5.055e-01 0.192 0.847493

relevel(as.factor(Brand), ref = "LG")Sony -2.271e+00 4.596e-01 -4.941 7.97e-07 \*\*\*

relevel(as.factor(Brand), ref = "LG")TCL 1.215e+00 6.045e-01 2.010 0.044482 \*

relevel(as.factor(Casa), ref = "Bekaa")Beirut 1.735e+01 6.222e-01 27.889 < 2e-16 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 2.390e+01 6.964e-01 34.319 < 2e-16 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")North 4.633e+00 5.887e-01 7.870 4.18e-15 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")South 1.111e+01 6.150e-01 18.064 < 2e-16 \*\*\*

relevel(as.factor(Sound.Quality), ref = "Basic")Enhanced -2.705e+00 6.984e-01 -3.874 0.000108 \*\*\*

relevel(as.factor(Sound.Quality), ref = "Basic")Premium -4.267e+00 4.719e-01 -9.043 < 2e-16 \*\*\*

relevel(as.factor(Connectivity.Options), ref = "2")3 2.018e+00 5.490e-01 3.675 0.000240 \*\*\*

relevel(as.factor(Connectivity.Options), ref = "2")4 -5.968e+00 7.074e-01 -8.437 < 2e-16 \*\*\*

relevel(as.factor(Energy.Efficiency), ref = "A")B 4.979e+00 6.842e-01 7.276 3.88e-13 \*\*\*

relevel(as.factor(Energy.Efficiency), ref = "A")C 5.274e+00 7.860e-01 6.711 2.12e-11 \*\*\*

relevel(as.factor(Energy.Efficiency), ref = "A")D 4.431e+00 8.528e-01 5.196 2.10e-07 \*\*\*

relevel(as.factor(Energy.Efficiency), ref = "A")E -6.912e-01 9.800e-01 -0.705 0.480666

relevel(as.factor(Month), ref = "June")April 1.438e-01 7.419e-01 0.194 0.846334

relevel(as.factor(Month), ref = "June")August -1.399e+00 7.975e-01 -1.754 0.079463 .

relevel(as.factor(Month), ref = "June")December 2.798e+00 6.955e-01 4.022 5.84e-05 \*\*\*

relevel(as.factor(Month), ref = "June")February 1.009e+00 7.985e-01 1.264 0.206275

relevel(as.factor(Month), ref = "June")January 2.843e+00 7.161e-01 3.970 7.28e-05 \*\*\*

relevel(as.factor(Month), ref = "June")July 1.383e+00 7.616e-01 1.816 0.069454 .

relevel(as.factor(Month), ref = "June")March -2.715e-01 7.377e-01 -0.368 0.712830

relevel(as.factor(Month), ref = "June")May -5.806e-01 8.046e-01 -0.722 0.470589

relevel(as.factor(Month), ref = "June")November 8.043e-01 8.291e-01 0.970 0.332067

relevel(as.factor(Month), ref = "June")October 6.680e-01 7.919e-01 0.844 0.398934

relevel(as.factor(Month), ref = "June")September 1.779e+00 8.416e-01 2.114 0.034584 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.05 on 5840 degrees of freedom

Multiple R-squared: 0.7517, Adjusted R-squared: 0.7501

F-statistic: 465.2 on 38 and 5840 DF, p-value: < 2.2e-16

**#Model 3 RMSE:**

pred3=predict(model3,val)

perform3=pred\_error(val$Tot.Qty, pred3)

perform3

MAPE MAE RMSE

10.522219 6.529426 11.030702

**#Model 3 Complexity:**

9

**#Multicolinearity:**

library(car)

vif(model3)

vif3Squarred <- vif(model3)

vif3Squarred[, ncol(vif3Squarred)] <- vif3Squarred[, ncol(vif3Squarred)]^2

print(vif3Squarred)

GVIF Df GVIF^(1/(2\*Df))^2

Unit.Price 3.627433 1 3.627433

relevel(as.factor(Inches), ref = "52") 38.225773 8 1.576865

relevel(as.factor(Resolution), ref = "UHD") 1.152156 2 1.073385

relevel(as.factor(Brand), ref = "LG") 2.206714 4 1.218811

relevel(as.factor(Casa), ref = "Bekaa") 5.036456 4 1.498067

relevel(as.factor(Sound.Quality), ref = "Basic") 2.725500 2 1.650909

relevel(as.factor(Connectivity.Options), ref = "2") 6.452358 2 2.540149

relevel(as.factor(Energy.Efficiency), ref = "A") 26.467203 4 2.268177

relevel(as.factor(Month), ref = "June") 1.829066 11 1.056426

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#Model 4: Remove all except Unit Price, Inches, Casa:**

model4=lm(Tot.Qty~Unit.Price +relevel(as.factor(Inches),ref="52")+relevel(as.factor(Casa),ref="Bekaa"),data=train)

summary(model4)

Residuals:

Min 1Q Median 3Q Max

-77.480 -3.596 0.005 3.775 75.666

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 8.694e+01 7.816e-01 111.242 <2e-16 \*\*\*

Unit.Price -1.619e-02 6.514e-04 -24.856 <2e-16 \*\*\*

relevel(as.factor(Inches), ref = "52")40 2.700e-01 6.522e-01 0.414 0.6789

relevel(as.factor(Inches), ref = "52")43 -1.243e-01 6.467e-01 -0.192 0.8476

relevel(as.factor(Inches), ref = "52")45 -4.807e-01 6.349e-01 -0.757 0.4490

relevel(as.factor(Inches), ref = "52")50 2.246e-01 6.352e-01 0.354 0.7237

relevel(as.factor(Inches), ref = "52")55 -6.580e-01 6.315e-01 -1.042 0.2975

relevel(as.factor(Inches), ref = "52")58 -1.744e+00 6.837e-01 -2.551 0.0108 \*

relevel(as.factor(Inches), ref = "52")65 -7.385e+00 7.636e-01 -9.671 <2e-16 \*\*\*

relevel(as.factor(Inches), ref = "52")75 -2.000e+01 8.474e-01 -23.600 <2e-16 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")Beirut 2.064e+01 5.931e-01 34.789 <2e-16 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 2.746e+01 6.582e-01 41.719 <2e-16 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")North 6.875e+00 5.606e-01 12.265 <2e-16 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")South 1.357e+01 5.898e-01 23.006 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.45 on 5865 degrees of freedom

Multiple R-squared: 0.7322, Adjusted R-squared: 0.7316

F-statistic: 1234 on 13 and 5865 DF, p-value: < 2.2e-16

**#Model 4 RMSE:**

pred4=predict(model4,val)

perform4=pred\_error(val$Tot.Qty, pred4)

perform4

MAPE MAE RMSE

10.594648 6.566468 11.354944

**#Model 4 complexity:**

3

**#Multicolinearity:**

library(car)

vif(model4)

vif4Squarred <- vif(model4)

vif4Squarred[, ncol(vif4Squarred)] <- vif4Squarred[, ncol(vif4Squarred)]^2

print(vif4Squarred)

GVIF Df GVIF^(1/(2\*Df))^2

Unit.Price 2.941096 1 2.941096

relevel(as.factor(Inches), ref = "52") 3.812829 8 1.182105

relevel(as.factor(Casa), ref = "Bekaa") 2.657660 4 1.276806

**#Model 4: Interaction:**

model4\_interaction=lm(Tot.Qty~Unit.Price\*relevel(as.factor(Inches),ref="52")\*relevel(as.factor(Casa),ref="Bekaa"),data=train)

summary(model4\_interaction)

Residuals:

Min 1Q Median 3Q Max

-75.675 -2.742 0.353 3.487 97.621

Coefficients:

Estimate

(Intercept) 9.210e+01

Unit.Price -4.979e-03

relevel(as.factor(Inches), ref = "52")40 -1.588e+00

relevel(as.factor(Inches), ref = "52")43 -3.334e+00

relevel(as.factor(Inches), ref = "52")45 -8.493e+00

relevel(as.factor(Inches), ref = "52")50 5.711e+00

relevel(as.factor(Inches), ref = "52")55 2.905e+00

relevel(as.factor(Inches), ref = "52")58 -9.691e-01

relevel(as.factor(Inches), ref = "52")65 -2.031e+00

relevel(as.factor(Inches), ref = "52")75 1.592e+01

relevel(as.factor(Casa), ref = "Bekaa")Beirut 1.155e+01

relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 7.060e+00

relevel(as.factor(Casa), ref = "Bekaa")North -7.732e-01

relevel(as.factor(Casa), ref = "Bekaa")South -4.161e+00

Unit.Price:relevel(as.factor(Inches), ref = "52")40 4.817e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")43 -2.167e-04

Unit.Price:relevel(as.factor(Inches), ref = "52")45 9.582e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")50 -8.696e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")55 -2.732e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")58 -1.977e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")65 -2.187e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")75 -4.157e-02

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")Beirut 5.933e-04

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 1.086e-03

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")North -1.057e-02

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")South 1.307e-02

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Beirut 1.935e+00

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Beirut 1.561e+00

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Beirut 4.319e+00

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Beirut -6.164e+00

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Beirut -2.804e+00

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Beirut -2.167e+01

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Beirut -9.665e+00

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Beirut -1.846e+01

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 2.039e+01

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 2.693e+01

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 2.155e+01

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 1.003e+01

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon -9.086e+00

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 1.191e+00

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 8.126e+00

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon -1.557e+01

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")North -5.051e+00

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")North -2.983e+00

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")North 1.094e+00

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")North -8.884e+00

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")North -1.238e+01

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")North 1.711e+01

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")North 9.766e+00

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")North -2.745e+01

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")South 1.395e+01

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")South 6.778e+00

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")South 1.119e+01

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")South -6.189e+00

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")South -9.238e+00

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")South -2.871e+00

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")South -1.920e+00

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")South -1.465e+01

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Beirut -1.180e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Beirut 2.740e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Beirut -3.113e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Beirut 8.080e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Beirut 2.398e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Beirut 3.178e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Beirut 2.320e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Beirut 2.809e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon -2.172e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon -3.266e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon -1.625e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon -2.688e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 3.026e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 1.130e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 1.276e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 3.798e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")North 1.983e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")North 1.944e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")North 1.082e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")North 1.659e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")North 4.646e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")North 3.127e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")North 2.304e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")North 6.364e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")South -3.606e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")South -1.237e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")South -1.261e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")South 1.009e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")South 3.818e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")South 2.383e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")South 1.580e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")South 3.186e-02

Std. Error

(Intercept) 2.966e+00

Unit.Price 4.213e-03

relevel(as.factor(Inches), ref = "52")40 4.222e+00

relevel(as.factor(Inches), ref = "52")43 4.470e+00

relevel(as.factor(Inches), ref = "52")45 4.462e+00

relevel(as.factor(Inches), ref = "52")50 4.363e+00

relevel(as.factor(Inches), ref = "52")55 4.372e+00

relevel(as.factor(Inches), ref = "52")58 4.186e+00

relevel(as.factor(Inches), ref = "52")65 4.788e+00

relevel(as.factor(Inches), ref = "52")75 3.620e+00

relevel(as.factor(Casa), ref = "Bekaa")Beirut 3.868e+00

relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 4.512e+00

relevel(as.factor(Casa), ref = "Bekaa")North 4.283e+00

relevel(as.factor(Casa), ref = "Bekaa")South 3.584e+00

Unit.Price:relevel(as.factor(Inches), ref = "52")40 5.775e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")43 8.100e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")45 6.534e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")50 6.053e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")55 7.103e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")58 5.587e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")65 5.532e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")75 4.447e-03

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")Beirut 6.375e-03

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 6.867e-03

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")North 7.687e-03

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")South 5.692e-03

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Beirut 5.062e+00

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Beirut 5.650e+00

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Beirut 5.348e+00

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Beirut 5.440e+00

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Beirut 5.474e+00

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Beirut 5.627e+00

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Beirut 6.087e+00

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Beirut 5.633e+00

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 5.607e+00

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 5.988e+00

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 6.088e+00

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 5.894e+00

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 6.432e+00

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 8.372e+00

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 7.140e+00

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 6.464e+00

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")North 5.676e+00

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")North 5.756e+00

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")North 5.940e+00

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")North 5.671e+00

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")North 5.859e+00

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")North 7.172e+00

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")North 6.929e+00

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")North 5.944e+00

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")South 5.190e+00

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")South 6.036e+00

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")South 5.390e+00

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")South 5.186e+00

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")South 5.361e+00

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")South 5.556e+00

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")South 6.640e+00

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")South 5.513e+00

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Beirut 8.056e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Beirut 1.146e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Beirut 8.778e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Beirut 8.874e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Beirut 9.369e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Beirut 8.376e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Beirut 8.357e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Beirut 7.762e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 8.925e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 1.096e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 1.004e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 8.884e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 1.084e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 1.211e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 9.533e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 8.997e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")North 1.051e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")North 1.113e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")North 1.076e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")North 9.635e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")North 1.041e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")North 1.067e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")North 9.751e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")North 9.064e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")South 9.634e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")South 1.316e-02

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")South 9.305e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")South 8.077e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")South 8.999e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")South 7.952e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")South 8.385e-03

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")South 7.628e-03

t value

(Intercept) 31.056

Unit.Price -1.182

relevel(as.factor(Inches), ref = "52")40 -0.376

relevel(as.factor(Inches), ref = "52")43 -0.746

relevel(as.factor(Inches), ref = "52")45 -1.904

relevel(as.factor(Inches), ref = "52")50 1.309

relevel(as.factor(Inches), ref = "52")55 0.664

relevel(as.factor(Inches), ref = "52")58 -0.231

relevel(as.factor(Inches), ref = "52")65 -0.424

relevel(as.factor(Inches), ref = "52")75 4.398

relevel(as.factor(Casa), ref = "Bekaa")Beirut 2.986

relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 1.565

relevel(as.factor(Casa), ref = "Bekaa")North -0.181

relevel(as.factor(Casa), ref = "Bekaa")South -1.161

Unit.Price:relevel(as.factor(Inches), ref = "52")40 0.834

Unit.Price:relevel(as.factor(Inches), ref = "52")43 -0.027

Unit.Price:relevel(as.factor(Inches), ref = "52")45 1.466

Unit.Price:relevel(as.factor(Inches), ref = "52")50 -1.437

Unit.Price:relevel(as.factor(Inches), ref = "52")55 -3.846

Unit.Price:relevel(as.factor(Inches), ref = "52")58 -3.538

Unit.Price:relevel(as.factor(Inches), ref = "52")65 -3.953

Unit.Price:relevel(as.factor(Inches), ref = "52")75 -9.349

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.093

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.158

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")North -1.375

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")South 2.296

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.382

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.276

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.808

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Beirut -1.133

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Beirut -0.512

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Beirut -3.850

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Beirut -1.588

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Beirut -3.277

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 3.637

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 4.498

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 3.540

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 1.702

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon -1.413

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.142

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 1.138

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon -2.409

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")North -0.890

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")North -0.518

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")North 0.184

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")North -1.567

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")North -2.113

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")North 2.386

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")North 1.409

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")North -4.619

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")South 2.688

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")South 1.123

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")South 2.076

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")South -1.193

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")South -1.723

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")South -0.517

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")South -0.289

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")South -2.658

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Beirut -1.465

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.239

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Beirut -0.355

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.911

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Beirut 2.559

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Beirut 3.795

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Beirut 2.776

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Beirut 3.619

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon -2.433

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon -2.981

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon -1.618

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon -0.303

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 2.792

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.934

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 1.339

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 4.221

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")North 1.887

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")North 1.747

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")North 1.005

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")North 1.722

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")North 4.463

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")North 0.293

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")North 2.363

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")North 7.021

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")South -3.743

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")South -0.094

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")South -1.355

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")South 1.250

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")South 4.242

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")South 2.996

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")South 1.885

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")South 4.177

Pr(>|t|)

(Intercept) < 2e-16 \*\*\*

Unit.Price 0.237309

relevel(as.factor(Inches), ref = "52")40 0.706844

relevel(as.factor(Inches), ref = "52")43 0.455864

relevel(as.factor(Inches), ref = "52")45 0.057023 .

relevel(as.factor(Inches), ref = "52")50 0.190580

relevel(as.factor(Inches), ref = "52")55 0.506410

relevel(as.factor(Inches), ref = "52")58 0.816939

relevel(as.factor(Inches), ref = "52")65 0.671377

relevel(as.factor(Inches), ref = "52")75 1.11e-05 \*\*\*

relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.002840 \*\*

relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.117686

relevel(as.factor(Casa), ref = "Bekaa")North 0.856726

relevel(as.factor(Casa), ref = "Bekaa")South 0.245681

Unit.Price:relevel(as.factor(Inches), ref = "52")40 0.404253

Unit.Price:relevel(as.factor(Inches), ref = "52")43 0.978659

Unit.Price:relevel(as.factor(Inches), ref = "52")45 0.142572

Unit.Price:relevel(as.factor(Inches), ref = "52")50 0.150855

Unit.Price:relevel(as.factor(Inches), ref = "52")55 0.000122 \*\*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")58 0.000407 \*\*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")65 7.81e-05 \*\*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")75 < 2e-16 \*\*\*

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.925850

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.874386

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")North 0.169087

Unit.Price:relevel(as.factor(Casa), ref = "Bekaa")South 0.021685 \*

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.702204

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.782418

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.419363

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.257158

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.608414

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.000119 \*\*\*

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.112390

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.001056 \*\*

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.000278 \*\*\*

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 7.00e-06 \*\*\*

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.000403 \*\*\*

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.088802 .

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.157788

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.886866

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.255160

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.016046 \*

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")North 0.373567

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")North 0.604358

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")North 0.853939

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")North 0.117265

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")North 0.034626 \*

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")North 0.017068 \*

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")North 0.158793

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")North 3.94e-06 \*\*\*

relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")South 0.007211 \*\*

relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")South 0.261476

relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")South 0.037956 \*

relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")South 0.232795

relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")South 0.084892 .

relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")South 0.605272

relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")South 0.772449

relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")South 0.007875 \*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.143057

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.810971

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.722879

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.362583

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.010518 \*

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.000149 \*\*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.005525 \*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Beirut 0.000298 \*\*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.014987 \*

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.002881 \*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.105715

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.762195

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.005256 \*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.350600

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 0.180781

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")Mount Lebanon 2.47e-05 \*\*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")North 0.059240 .

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")North 0.080759 .

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")North 0.314777

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")North 0.085154 .

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")North 8.23e-06 \*\*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")North 0.769356

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")North 0.018168 \*

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")North 2.46e-12 \*\*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")40:relevel(as.factor(Casa), ref = "Bekaa")South 0.000184 \*\*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")43:relevel(as.factor(Casa), ref = "Bekaa")South 0.925102

Unit.Price:relevel(as.factor(Inches), ref = "52")45:relevel(as.factor(Casa), ref = "Bekaa")South 0.175342

Unit.Price:relevel(as.factor(Inches), ref = "52")50:relevel(as.factor(Casa), ref = "Bekaa")South 0.211475

Unit.Price:relevel(as.factor(Inches), ref = "52")55:relevel(as.factor(Casa), ref = "Bekaa")South 2.25e-05 \*\*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")58:relevel(as.factor(Casa), ref = "Bekaa")South 0.002745 \*\*

Unit.Price:relevel(as.factor(Inches), ref = "52")65:relevel(as.factor(Casa), ref = "Bekaa")South 0.059495 .

Unit.Price:relevel(as.factor(Inches), ref = "52")75:relevel(as.factor(Casa), ref = "Bekaa")South 3.00e-05 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.1 on 5789 degrees of freedom

Multiple R-squared: 0.7943, Adjusted R-squared: 0.7912

F-statistic: 251.2 on 89 and 5789 DF, p-value: < 2.2e-16

**#Comparative table:**

perf\_comp=rbind(perform1,perform2,perform3,perform4)

numb\_predictors=c(num\_predictors1,num\_predictors2,num\_predictors3,num\_predictors4)

cbind(perf\_comp,numb\_predictors)

MAPE MAE RMSE numb\_predictors

perform1 10.52526 6.528354 11.02643 12

perform2 10.52570 6.529110 11.02625 10

perform3 10.52222 6.529426 11.03070 9

perform4 10.59465 6.566468 11.35494 3

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**Classification and Regression Trees (CART):**

**#Libraries**

library(rpart)

library(rpart.plot)

**#CART 1**

tree1 <-rpart(Tot.Qty ~ Casa+Brand+Smart+Inches+Resolution+Unit.Price+Refresh.Rate+Sound.Quality+Connectivity.Options+Energy.Efficiency+Pixel.Density..PPI.+Response.Time..ms.+Month, data = train, control = rpart.control(minbucket=50,minsplit=100,maxdepth=10,cp = 0.0001))

printcp(tree1)

Regression tree1:

rpart(formula = Tot.Qty ~ Casa + Brand + Smart + Inches + Resolution +

Unit.Price + Refresh.Rate + Sound.Quality + Connectivity.Options +

Energy.Efficiency + Pixel.Density..PPI. + Response.Time..ms. +

Month, data = train, control = rpart.control(minbucket = 50,

minsplit = 100, maxdepth = 10, cp = 1e-04))

Variables actually used in tree1 construction:

[1] Brand Casa Connectivity.Options Energy.Efficiency

[5] Inches Month Resolution Sound.Quality

[9] Unit.Price

Root node error: 2871206/5879 = 488.38

n= 5879

CP nsplit rel error xerror xstd

1 0.49042189 0 1.00000 1.00029 0.019985

2 0.11345630 1 0.50958 0.50991 0.013891

3 0.07193941 2 0.39612 0.40270 0.012798

4 0.03295180 3 0.32418 0.33086 0.013216

5 0.01934161 4 0.29123 0.29939 0.012779

6 0.01499582 5 0.27189 0.27647 0.011786

7 0.01079935 6 0.25689 0.26472 0.011530

8 0.01009662 7 0.24609 0.25695 0.011599

9 0.00798409 8 0.23600 0.24274 0.011582

10 0.00681719 9 0.22801 0.23522 0.011536

11 0.00273221 10 0.22120 0.23100 0.011634

12 0.00260368 11 0.21846 0.22782 0.011696

13 0.00208981 12 0.21586 0.22475 0.011646

14 0.00188825 13 0.21377 0.22422 0.011621

15 0.00188523 14 0.21188 0.22352 0.011600

16 0.00150348 15 0.21000 0.22199 0.011505

17 0.00136250 16 0.20849 0.22240 0.011554

18 0.00127438 17 0.20713 0.22206 0.011623

19 0.00124666 19 0.20458 0.22220 0.011651

20 0.00116778 20 0.20334 0.22147 0.011656

21 0.00108145 21 0.20217 0.22037 0.011665

22 0.00103420 22 0.20109 0.22079 0.011734

23 0.00102993 23 0.20005 0.22059 0.011724

24 0.00088413 25 0.19799 0.21989 0.011734

25 0.00071960 26 0.19711 0.21854 0.011710

26 0.00065935 27 0.19639 0.21749 0.011702

27 0.00061619 28 0.19573 0.21781 0.011724

28 0.00057304 29 0.19511 0.21787 0.011699

29 0.00052630 30 0.19454 0.21799 0.011704

30 0.00051373 31 0.19401 0.21858 0.011727

31 0.00050417 32 0.19350 0.21834 0.011688

32 0.00042460 33 0.19300 0.21770 0.011665

33 0.00039291 34 0.19257 0.21791 0.011659

34 0.00037921 35 0.19218 0.21792 0.011653

35 0.00032716 37 0.19142 0.21790 0.011639

36 0.00032066 38 0.19109 0.21838 0.011662

37 0.00030566 39 0.19077 0.21841 0.011669

38 0.00030107 40 0.19047 0.21827 0.011659

39 0.00026977 41 0.19017 0.21802 0.011637

40 0.00026437 42 0.18990 0.21785 0.011608

41 0.00023751 43 0.18963 0.21763 0.011575

42 0.00022154 44 0.18939 0.21753 0.011574

43 0.00020851 45 0.18917 0.21768 0.011582

44 0.00018499 46 0.18896 0.21761 0.011578

45 0.00015949 47 0.18878 0.21747 0.011549

46 0.00014305 48 0.18862 0.21711 0.011538

47 0.00014133 49 0.18848 0.21717 0.011540

48 0.00011511 50 0.18833 0.21715 0.011525

49 0.00010433 52 0.18810 0.21720 0.011525

50 0.00010409 53 0.18800 0.21716 0.011523

51 0.00010000 54 0.18790 0.21711 0.011524

bestcp=tree1$cptable[which.min(tree1$cptable[,"xerror"]),"CP"]

tree1.pruned=prune(tree1, cp = bestcp)

rpart.plot(tree1.pruned,

type = 4,

extra = 101,

under = TRUE,

faclen = 0,

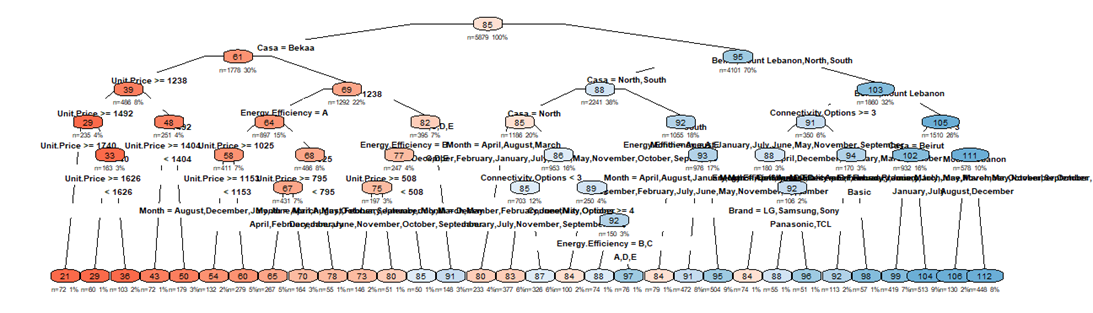
cex = 0.6,

tweak = 1.0,

box.palette = "RdBu",

shadow.col = NA,

branch = 0.3 )



**#CART 1: RMSE**

val$pred1 <- predict(tree1.pruned, newdata = val, type = "vector")

errors1 <- pred\_error(actual = val$Tot.Qty, pred = val$pred1)

print(errors1)

MAPE MAE RMSE

7.987680 5.067994 10.12551

**#Mean**

mean(val$Tot.Qty)

84.6248

n1=length(tree1.pruned$frame$var)

109

t1=sum(tree1.pruned$frame$var == "<leaf>")

55

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#CART 1.1**

n= 5879

CP nsplit rel error xerror xstd

1 0.49042189 0 1.00000 1.00029 0.019985

2 0.11345630 1 0.50958 0.50991 0.013891

3 0.07193941 2 0.39612 0.40270 0.012798

4 0.03295180 3 0.32418 0.33086 0.013216

5 0.01934161 4 0.29123 0.29939 0.012779

6 0.01499582 5 0.27189 0.27647 0.011786

7 0.01079935 6 0.25689 0.26472 0.011530

8 0.01009662 7 0.24609 0.25695 0.011599

9 0.00798409 8 0.23600 0.24274 0.011582

10 0.00681719 9 0.22801 0.23522 0.011536

11 0.00273221 10 0.22120 0.23100 0.011634

12 0.00260368 11 0.21846 0.22782 0.011696

13 0.00208981 12 0.21586 0.22475 0.011646

14 0.00188825 13 0.21377 0.22422 0.011621

15 0.00188523 14 0.21188 0.22352 0.011600

16 0.00150348 15 0.21000 0.22199 0.011505

17 0.00136250 16 0.20849 0.22240 0.011554

18 0.00127438 17 0.20713 0.22206 0.011623

19 0.00124666 19 0.20458 0.22220 0.011651

20 0.00116778 20 0.20334 0.22147 0.011656

21 0.00108145 21 0.20217 0.22037 0.011665

22 0.00103420 22 0.20109 0.22079 0.011734

23 0.00102993 23 0.20005 0.22059 0.011724

24 0.00088413 25 0.19799 0.21989 0.011734

25 0.00071960 26 0.19711 0.21854 0.011710

26 0.00065935 27 0.19639 0.21749 0.011702

27 0.00061619 28 0.19573 0.21781 0.011724

28 0.00057304 29 0.19511 0.21787 0.011699

29 0.00052630 30 0.19454 0.21799 0.011704

30 0.00051373 31 0.19401 0.21858 0.011727

31 0.00050417 32 0.19350 0.21834 0.011688

32 0.00042460 33 0.19300 0.21770 0.011665

33 0.00039291 34 0.19257 0.21791 0.011659

34 0.00037921 35 0.19218 0.21792 0.011653

35 0.00032716 37 0.19142 0.21790 0.011639

36 0.00032066 38 0.19109 0.21838 0.011662

37 0.00030566 39 0.19077 0.21841 0.011669

38 0.00030107 40 0.19047 0.21827 0.011659

39 0.00026977 41 0.19017 0.21802 0.011637

40 0.00026437 42 0.18990 0.21785 0.011608

41 0.00023751 43 0.18963 0.21763 0.011575

42 0.00022154 44 0.18939 0.21753 0.011574

43 0.00020851 45 0.18917 0.21768 0.011582

44 0.00018499 46 0.18896 0.21761 0.011578

45 0.00015949 47 0.18878 0.21747 0.011549

46 0.00014305 48 0.18862 0.21711 0.011538

47 0.00014133 49 0.18848 0.21717 0.011540

48 0.00011511 50 0.18833 0.21715 0.011525

49 0.00010433 52 0.18810 0.21720 0.011525

50 0.00010409 53 0.18800 0.21716 0.011523

51 0.00010000 54 0.18790 0.21711 0.011524

0.21858 + 0.011727=0.230307 > 0.21799

We can consider cp=0.00052630

tree1.1.pruned=prune(tree1, cp = 0.00052630 )

rpart.plot(tree1.1.pruned,

type = 4,

extra = 101,

under = TRUE,

faclen = 0,

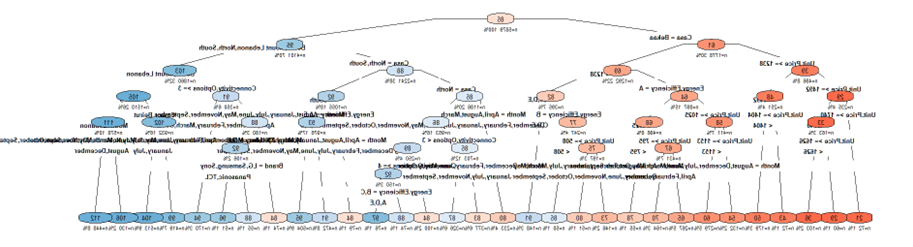
cex = 0.6,

tweak = 1.0,

box.palette = "RdBu",

shadow.col = NA,

branch = 0.3 )



val$pred1.1 <- predict(tree1.1.pruned, newdata = val, type = "vector")

errors1.1 <- pred\_error(actual = val$Tot.Qty, pred = val$pred.1.1)

print(errors1.1)

MAPE MAE RMSE

8.167825 5.43486 10.574883

n1.1=length(tree1.1.pruned$frame$var)

[1] 61

t1.1=sum(tree1.1.pruned$frame$var == "<leaf>")

[1] 31

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#CART 2**

tree2 <-rpart(Tot.Qty ~ Casa+Brand+Smart+Inches+Resolution+Unit.Price+Refresh.Rate+Sound.Quality+Connectivity.Options+Energy.Efficiency+Pixel.Density..PPI.+Response.Time..ms.+Month, data = train, control = rpart.control(minbucket=50,minsplit=100,maxdepth=10))

printcp(tree2)

Regression tree:

rpart(formula = Tot.Qty ~ Casa + Brand + Smart + Inches + Resolution +

Unit.Price + Refresh.Rate + Sound.Quality + Connectivity.Options +

Energy.Efficiency + Pixel.Density..PPI. + Response.Time..ms. +

Month, data = train, control = rpart.control(minbucket = 50,

minsplit = 100, maxdepth = 10))

Variables actually used in tree construction:

[1] Casa Connectivity.Options Energy.Efficiency

[4] Unit.Price

Root node error: 2871206/5879 = 488.38

n= 5879

CP nsplit rel error xerror xstd

1 0.490422 0 1.00000 1.00016 0.019982

2 0.113456 1 0.50958 0.50988 0.013895

3 0.071939 2 0.39612 0.39867 0.012596

4 0.032952 3 0.32418 0.32681 0.013012

5 0.019342 4 0.29123 0.29621 0.012623

6 0.014996 5 0.27189 0.27716 0.011766

7 0.010799 6 0.25689 0.26447 0.011554

8 0.010097 7 0.24609 0.25126 0.011591

9 0.010000 8 0.23600 0.24671 0.011631

bestcp=tree2$cptable[which.min(tree2$cptable[,"xerror"]),"CP"]

tree2.pruned=prune(tree2, cp = bestcp)

rpart.plot(tree2.pruned,

type = 4,

extra = 101,

under = TRUE,

faclen = 0,

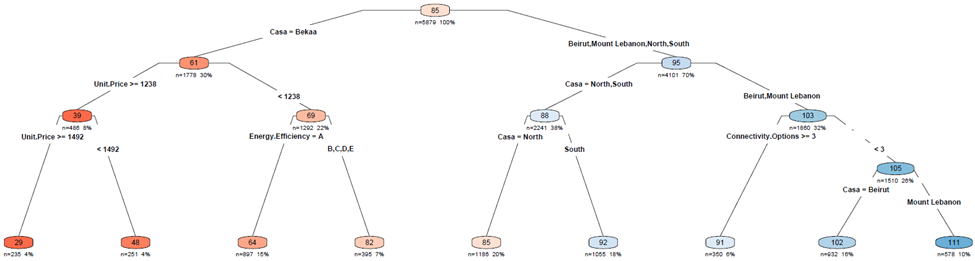
cex = 0.6,

tweak = 1.0,

box.palette = "RdBu",

shadow.col = NA,

branch = 0.3 )



~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

**#CART 2.1**

Regression tree:

Variables actually used in tree construction:

[1] Casa Connectivity.Options Energy.Efficiency

[4] Unit.Price

Root node error: 2871206/5879 = 488.38

n= 5879

CP nsplit rel error xerror xstd

1 0.490422 0 1.00000 1.00016 0.019982

2 0.113456 1 0.50958 0.50988 0.013895

3 0.071939 2 0.39612 0.39867 0.012596

4 0.032952 3 0.32418 0.32681 0.013012

5 0.019342 4 0.29123 0.29621 0.012623

6 0.014996 5 0.27189 0.27716 0.011766

7 0.010799 6 0.25689 0.26447 0.011554

8 0.010097 7 0.24609 0.25126 0.011591

9 0.010000 8 0.23600 0.24671 0.011631

0.24671+0.011631 =0.258341>0.25126

We can consider cp=0.010097

tree2.1.pruned=prune(tree2, cp = 0.010097)

rpart.plot(tree2.1.pruned,

type = 4,

extra = 101,

under = TRUE,

faclen = 0,

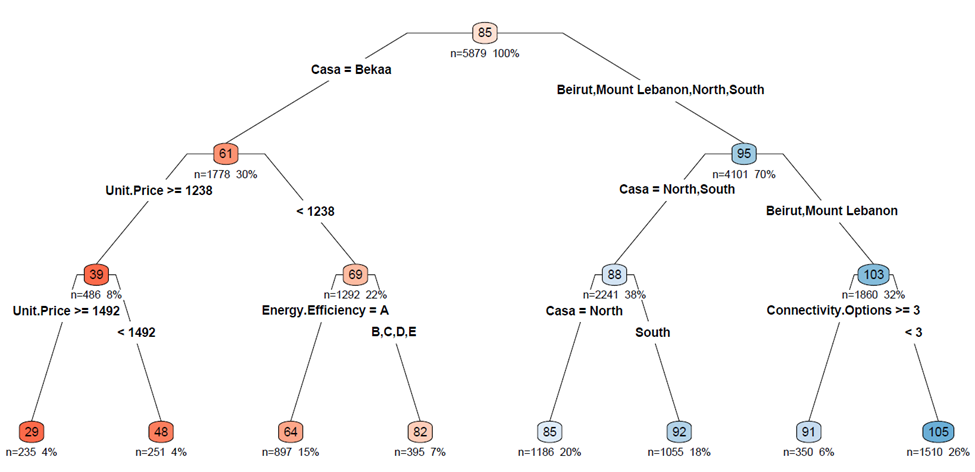
cex = 0.6,

tweak = 1.0,

box.palette = "RdBu",

shadow.col = NA,

branch = 0.3 )



> val$pred2.1 <- predict(tree2.1.pruned, newdata = val, type = "vector")

> errors2.1 <- pred\_error(actual = val$Tot.Qty, pred = val$pred.2.1)

> print(errors2.1)

MAPE MAE RMSE

10.517341 6.910001 11.085917

Comparative Table:

> results <- data.frame(

cp\_values = c(0.0001, 0.0001, 0.01, 0.01),

+ numb\_nodes = numb\_nodes,

+ numb\_leafs = numb\_leafs,

+ MAPE = errors\_Acomp[, "MAPE"],

+ MAE = errors\_comp[, "MAE"],

+ RMSE = errors\_comp[, "RMSE"],

+ row.names = c("tree1", "tree1.1", "tree2", "tree2.1")

)

