Case: Price Optimization

Team L: Sherine George George, Sherry Li, Wendy Kuo, Yuchen Zhu, Enrique Garcia

Date: February 5th, 2024

Part 1:

Q1: **A.** Using a log-log model, we estimated the price elasticities of Tropicana Premium, Tropicana, Minute Maid, and Dominick's are seen in **Table 1**. The interpretation of these values is that for every 1% increase in the price of each product, the quantity demanded is estimated to decrease by these percentages. For example, a 1% increase in Minute Maid is expected to decrease demand for that product by -3.378%. Of note, it makes sense that Tropicana Premium has the smallest elasticity because premium brand consumers are usually less sensitive to price changes.

B & C. The optimal prices of these products, based on the wholesale costs for store #2 during week #93, are also seen in the below table. When comparing these optimal prices to the actual prices charged per 64 oz unit, we can see that the first three products were overpriced, while Dominick's was underpriced.

Product	Part A: Price Elasticity	Part B: Optimal Price	Part C: Actual Price/64 oz Case
Tropicana Premium (1)	-2.687 %	$3.27	$3.59 [$0.32]
Tropicana (4)	-3.895 %	$2.06	$2.49 [$0.43]
Minute Maid (5)	-3.378 %	$2.21	$2.39 [$0.18]
Dominick's (11)	-3.375 %	$1.45	$1.29 [- $0.16]

Table 1: Price Elasticity and Optimal Price using log-log model

Part 2:

Q2: When constructing the three models, we decided to incorporate a greater number of variables and utilize regularization techniques. For each model, we included all 12 product prices into the regression as they hold both statistically and theoretically significant relationships with each other (being substitutes), as well as the in-store and out-of-store promotion index for that particular product. We also conducted regularization using the Lasso method to ensure that our models do not encounter the issue of overfitting, discouraging our model from being too complex.

In general, most models exhibit a mean error around zero (Exhibit 1). However, through the evaluation of root mean squared error and mean absolute deviation, we observed that the pooled model and decision tree have the highest root mean squared error, suggesting that the model may not be well-suited for both training and validation datasets. In comparison, in-sample and out-of-sample performance is consistent, except for the random tree model, which exhibits increased mean absolute deviation in out-of-sample data (from 0.1 to 0.24), suggesting a potential overfitting issue in the random forest model. The specificities of error values can be observed in **Table 2.**

In contrast, the store and mixed models show similar results between in-sample and out-of-sample data, and their root mean-squared errors fall within a reasonable range; it suggests they have less concern for overfitting or underfitting.

Training Data	pooled model	store model	mixed model	decision tree	random forest
mean-error	0.00	0.00	0.01	0.00	0.00
root mean-squared error	0.59	0.38	0.36	0.52	0.23
mean-absolute deviation	0.46	0.25	0.25	0.40	0.10

Validation Data	pooled model	store model	mixed model	decision tree	random forest
mean-error	0.00	0.00	-0.03	0.01	0.00
root mean-squared error	0.59	0.38	0.36	0.52	0.23
mean-absolute deviation	0.46	0.33	0.28	0.41	0.24

Table 2: Values of Errors in Training and Validation Data

<u>Q3:</u> The analysis of residuals across diverse models reveals distinct performances in capturing the dynamic patterns within sales data. A focused examination on the spread of the residuals and their dispersion across time provides detailed insights for each model. In the Pooled Model, there are notable underpredictions accompanied by positive skewness in residuals, as visually represented in Exhibit 2.1. On the other hand, the Store Model showcases balanced residuals with occasional underpredictions, highlighting a nuanced comprehension of sales variability across stores, as visually depicted in Exhibit 2.2. The Mixed Model indicates overpredictions with left-skewed residuals in Exhibit 2.3. The Decision Tree model presents balanced residuals with a slight right-skew, implying occasional underpredictions, as demonstrated in Exhibit 2.4. Lastly, Random Forest stands out with excellent predictive accuracy, featuring tightly clustered residuals, as observable in Exhibit 2.5. These nuanced evaluations underscore the models' diverse capabilities in capturing and interpreting the intricacies within the sales data.

<u>Q4:</u> In the given context, the Random Forest model emerges as a well-suited approach for capturing intricate nonlinear relationships and interactions among variables without the need for explicitly specifying these connections, as opposed to traditional linear models. This characteristic proves advantageous when dealing with real-world sales data, which often exhibits complexities that may not be easily characterized by linear relationships. Furthermore, the lowest In terms of validating economic theories, the model provides insights into Price Elasticity. The substantial feature importance assigned to lpr1 (0.48414274) as shown in Exhibit 3 aligns with the foundational economic principle that the price of a product (own-price elasticity) significantly influences its demand. A high importance score for the product's own price variable supports the notion that an increase in the product's price is likely to result in a decrease in its demand or quantity sold. Regarding Substitution Effects, where changes in the price of one product affect the demand for another, the model's findings suggest a nuanced interpretation. While lpr1 (the price of the product itself) demonstrates high importance, the importance scores for prices of other products (lpr2 to lpr12) are comparatively lower. This doesn't entirely dismiss substitution effects but indicates that the model may not capture these effects as strongly as the direct influence of the product's own price. However, the inclusion and relative importance of multiple price variables suggest some level of substitution effect within the modeled relationships.

Part 3:

Q5: Utilizing Excel Solver and R, we determine the optimal price for Tropicana Premium 64 oz (product #1) across various models, as illustrated in **Table 3.** The Pooled and Store models detailed in Exhibit 4.1 and 4.2, the mixed model presented in Exhibit 4.3, and the random forest model showcased in Exhibit 4.4 all contribute to identifying the most advantageous pricing strategies. In general, the optimal price per ounce for product #1 falls around $0.05 to $ 0.06, while the optimal price per unit varies between $3.00 and $4.00 across different models and diverse stores. In our analysis, we highlighted 5 different stores across 5 zones to consider the effects of diverse geographical locations. We were able to observe significant differences across optimal pricing across zones, highlighting importance in locational factors.

	Optimal Price/oz	Optimal Price/unit	Profit
Pooled Model	$ 0.05	$ 3.29	$ 34.94
Store Model #2	$ 0.06	$ 3.91	$ 173.37
Store Model #5	$ 0.05	$ 3.01	$ 117.33
Store Model #85	$ 0.05	$ 3.09	$ 102.86
Store Model #112	$ 0.05	$ 3.28	$ 298.77
Store Model #115	$ 0.05	$ 3.13	$ 154.82
Mixed Model #2	$ 0.05	$ 3.27	$ 105.59
Mixed Model #5	$ 0.05	$ 3.20	$ 128.08
Mixed Moel #83	$ 0.05	$ 3.22	$ 103.91
Mixed Model #112	$ 0.05	$ 3.20	$ 155.03
Mixed Model #115	$ 0.05	$ 3.21	$ 120.57
Random Forest #2	$ 0.06	$ 3.87	$ 223.00
Random Forest #5	$ 0.06	$ 3.66	$ 197.21
Random Forest #83	$ 0.05	$ 3.19	$ 139.48
Random Forest #112	$ 0.05	$ 3.19	$ 139.48
Random Forest #115	$ 0.06	$ 3.66	$ 197.21

Table 3: Optimal Price of All Models Used

These findings demonstrated the nuanced range of optimal pricing outcomes influenced by distinct modeling approaches and store-specific dynamics.

Q6: Based on our analysis, we assert that the random forest model stands out as the most effective among the four models we created, surpassing the simplicity of the cost-plus rule utilized in Part 1. While the cost-plus model offers simplicity in calculating optimal prices using price elasticity and wholesale cost, its significant drawbacks lie in neglecting the impact of competitor products and customer behavioral considerations. In contrast, the random forest model has incorporated critical factors such as substitution effects, localization, and dynamic pricing. By making competitor's price the dependent variable in the regression, it captures the inverse relationship between quantity and price, enabling the calculation of optimal quantities for product #1. Simultaneously, the model considers the geographical context by using the store as a variable in the regression, reflecting how neighborhoods react to price changes.This dual approach empowers the manager with a dynamic pricing strategy that not only considers competitor actions but also adapts swiftly to changes.

Part 4 (Optional):

Q7: For Product 4, 5, and 11, we can see the optimal price as produced by the random forest model in **Table 4** below. We can see that product 1 is a more premium product with consistently higher prices across all stores, while product 11 being the least premium product with consistently lower prices. Store 2 is at a location where one can afford higher prices for all products, compared to all other locations.

Product	Store	Optimal Unit Price	Profit
1	2	$3.87	$222.44
1	5	$3.66	$196.71
1	83	$3.19	$139.13
1	112	$3.19	$139.13
1	115	$3.66	$196.71
4	2	$3.06	$876.90
4	5	$2.89	$779.46
4	83	$2.59	$607.50
4	112	$2.70	$673.10
4	115	$2.89	$779.46
5	2	$3.17	$107.72
5	5	$2.99	$95.72
5	83	$2.62	$71.06
5	112	$2.62	$71.06
5	115	$2.99	$95.72
11	2	$3.08	$77.22
11	5	$2.91	$70.53
11	83	$2.53	$55.59
11	112	$2.53	$55.59
11	115	$2.91	$70.53

-

Table 4: Optimal Unit Price for Product 1, 4 ,5, and 11

Q8: When optimizing total profits by varying all prices, we achieve a different group of results when using the random forest algorithm. This can be seen below in **Table 5**, where we similarly extract stores 2, 5, 83, 112, and 115 for our analysis. It is important to note that we have trained our algorithm using a subset of past prices in the historical dataset, and hence it is unable to predict combination prices which are not available before, or out of this price range. In this case, it might be more advantageous to use a conventional linear regression if a sufficiently linear relationship exists.

Store	Total Profit	Pr1	Pr2	Pr3	Pr4	Pr5	Pr6	Pr7	Pr8	Pr9	Pr10	Pr11	Pr12
2	$2,138.43	$3.59	$3.33	$2.87	$1.99	$1.99	$2.64	$2.39	$2.53	$2.19	$1.99	$1.39	$1.65
5	$1,946.23	$2.99	$3.39	$2.99	$2.29	$2.99	$2.86	$2.46	$2.66	$1.99	$2.79	$2.49	$2.35
83	$2,077.75	$2.59	$2.54	$2.59	$1.99	$2.25	$2.46	$1.99	$2.22	$1.82	$1.29	$1.58	$1.50
112	$2,395.37	$2.99	$3.35	$2.99	$2.29	$2.62	$2.83	$2.46	$2.66	$1.99	$2.53	$2.29	$2.30
115	$2,010.03	$2.99	$3.39	$2.99	$2.29	$2.99	$2.86	$2.46	$2.66	$1.99	$2.79	$2.49	$2.35

Table 5: Optimal Pricing over Total Profits (Varying Prices)

Q9: If I am Dominick's (product 11) pricing manager, I will adopt the random forest algorithm to determine the prices of my product. However, I will use the model in Q7 rather than Q8, because we cannot assume that all other products will follow the same pricing method. This means that we can only assume that the past actions of consumers stay the same as before, and that they won't optimize their decisions. This is a more reasonable assumption to make, as we cannot assume rationality of other firms that will result in a Nash Equilibrium outcome. Another insight we draw from this is that we can price discriminate across different stores, and hence charge different prices to different consumers to maximize profit.

—---------------- —---------------- —---------------- —---------------- —---------------- —----------------

Appendix

Exhibit 1: Descriptive Statistics Table

*How to read table: We adapt Lasso estimator to filter out variables that will be impactable. After that, we get the following tables for statistical analysis.

- Group 1 : Using training data to fit the model
- Group 2 : Using validation data to fit the model
- err : mean-error
- sqerr : square of the error = error*error
- rmse : root mean-squared error
- abserr : mean-absolute deviation = the absolute error = abs (error)

1.1: Pooled Model

Descriptive statistics by group

INDICES: 1

	vars	n	mean	sd	min	max	range	se
err1	1	5824	0.00	0.59	-3.21	3.12	6.33	0.01
sqerr1	2	5824	0.35	0.57	0.00	10.31	10.31	0.01
rmse1	3	5824	0.59	0.00	0.59	0.59	0.00	0.00
abserr1	4	5824	0.46	0.37	0.00	3.21	3.21	0.00

INDICES: 2

	vars	n	mean	sd	min	max	range	se
err1	1	3873	0.00	0.59	-2.19	2.15	4.33	0.01
sqerr1	2	3873	0.35	0.53	0.00	4.79	4.79	0.01
rmse1	3	3873	0.59	0.00	0.59	0.59	0.00	0.00
abserr1	4	3873	0.46	0.37	0.00	2.19	2.19	0.01

1.2: Store Model

Descriptive statistics by group

INDICES: 1

	vars	n	mean	sd	min	max	range	se
err2	1	5824	0.00	0.34	-2.61	2.34	4.95	0
sqerr2	2	5824	0.12	0.26	0.00	6.82	6.82	0
rmse	3	5824	0.38	0.00	0.38	0.38	0.00	0
abserr2	4	5824	0.25	0.23	0.00	2.61	2.61	0

INDICES: 2

	vars	n	mean	sd	min	max	range	se
err2	1	3873	0.00	0.44	-2.21	2.21	4.42	0.01
sqerr2	2	3873	0.19	0.38	0.00	4.90	4.90	0.01
rmse	3	3873	0.38	0.00	0.38	0.38	0.00	0.00
abserr2	4	3873	0.33	0.30	0.00	2.21	2.21	0.00

1.3: Mixed Model

```
 Descriptive statistics by group 
INDICES: 1
        vars    n mean   sd   min  max range se
err3       1 5824 0.00 0.34 -2.61 2.63  5.25  0
sqerr3     2 5824 0.12 0.29  0.00 6.94  6.94  0
rmse3      3 5824 0.36 0.00  0.36 0.36  0.00  0
abserr3    4 5824 0.25 0.24  0.00 2.63  2.63  0
--------------------------------------------------
INDICES: 2
        vars    n mean   sd   min  max range   se
err3       1 3261 0.00 0.38 -2.19 2.07  4.25 0.01
sqerr3     2 3261 0.15 0.31  0.00 4.79  4.79 0.01
rmse3      3 3873 0.36 0.00  0.36 0.36  0.00 0.00
abserr3    4 3261 0.28 0.26  0.00 2.19  2.19 0.00
```

1.4: Decision Tree

```
 Descriptive statistics by group 
group: 1
        vars    n mean   sd   min   max range   se
err5       1 5824 0.00 0.52 -3.56  2.75  6.31 0.01
sqerr5     2 5824 0.27 0.49  0.00 12.70 12.70 0.01
rmse5      3 5824 0.52 0.00  0.52  0.52  0.00 0.00
abserr5    4 5824 0.40 0.33  0.00  3.56  3.56 0.00
--------------------------------------------------
group: 2
        vars    n mean   sd   min  max range   se
err5       1 3873 0.01 0.52 -1.89 2.20  4.09 0.01
sqerr5     2 3873 0.27 0.44  0.00 4.85  4.85 0.01
rmse5      3 3873 0.52 0.00  0.52 0.52  0.00 0.00
abserr5    4 3873 0.41 0.33  0.00 2.20  2.20 0.01
```

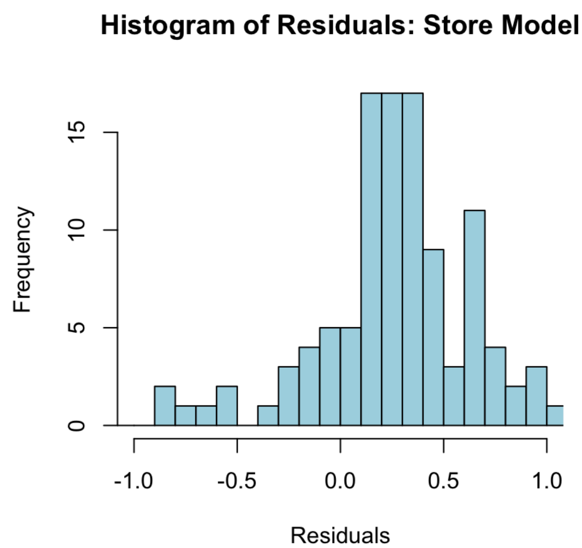
1.5: Random Forest

```
 Descriptive statistics by group
group: 1
          vars    n mean   sd   min  max range se
err_rf       1 5824 0.00 0.13 -1.25 1.09  2.34  0
sqerr_rf     2 5824 0.02 0.05  0.00 1.57  1.57  0
rmse_rf      3 5824 0.23 0.00  0.23 0.23  0.00  0
abserr_rf    4 5824 0.10 0.09  0.00 1.25  1.25  0
-----------------------------------------------------
group: 2
          vars    n mean   sd   min  max range   se
err_rf       1 3873 0.00 0.32 -1.68 1.96  3.64 0.01
sqerr_rf     2 3873 0.10 0.21  0.00 3.84  3.84 0.00
rmse_rf      3 3873 0.23 0.00  0.23 0.23  0.00 0.00
abserr_rf    4 3873 0.24 0.21  0.00 1.96  1.96 0.00
```

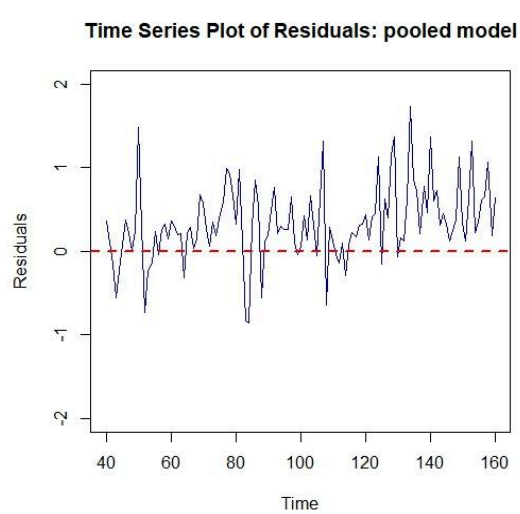
Exhibit 2: Histograms and Time Series Plots of Residuals

2.1: Pooled Model

A: Histogram of Residuals

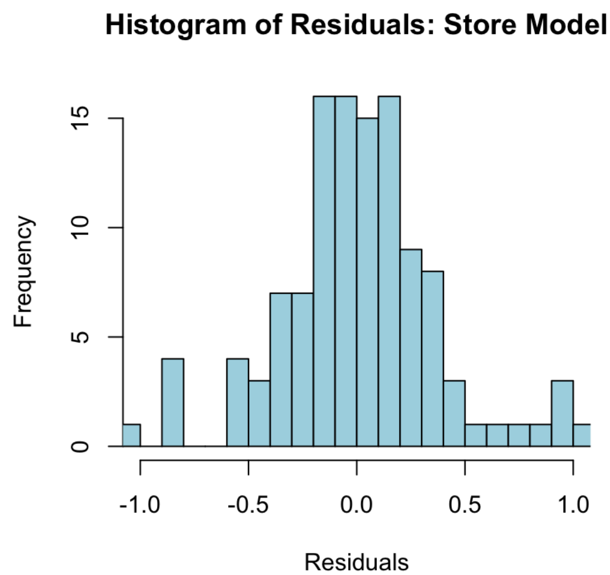


B: Time Series Plot of Residual

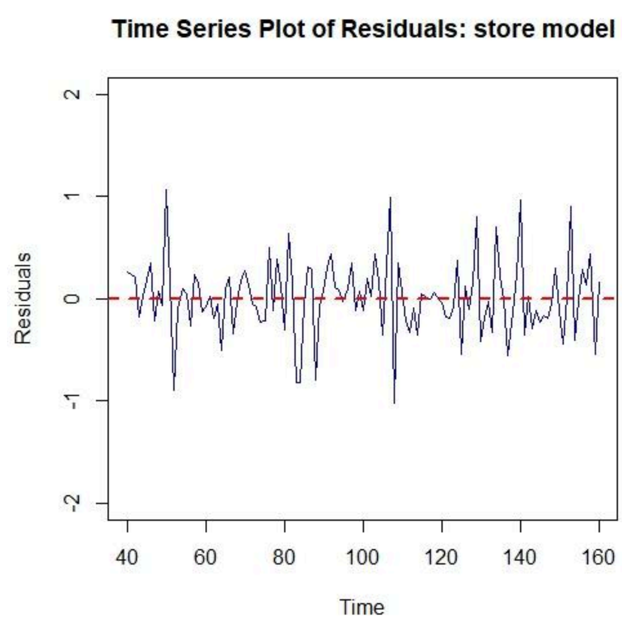


2.2: Store Model (Store 5)

A: Histogram of Residuals

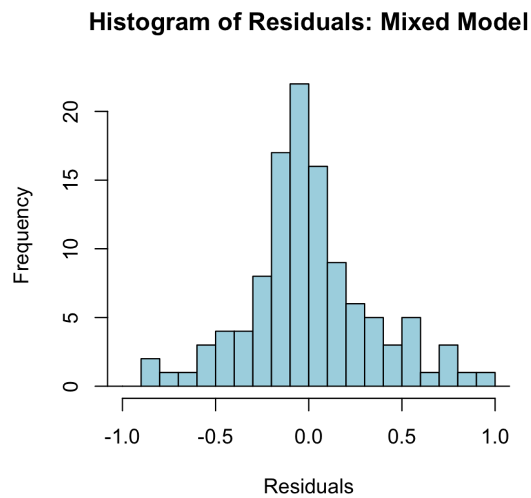


B: Time Series Plot of Residuals

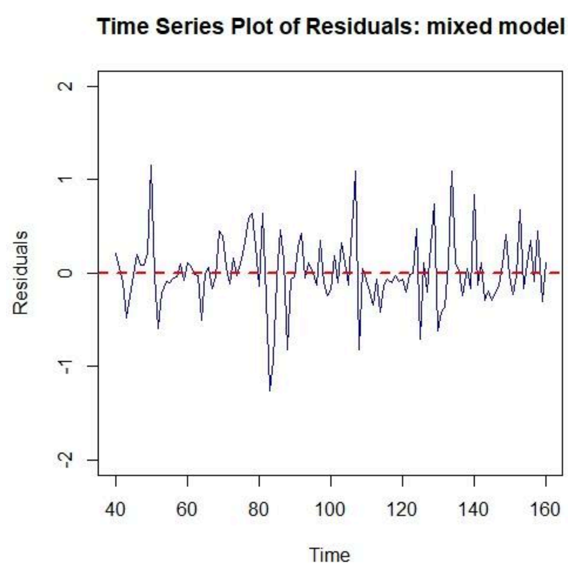


2.3: Mixed Model (Store 5)

A: Histogram of Residuals

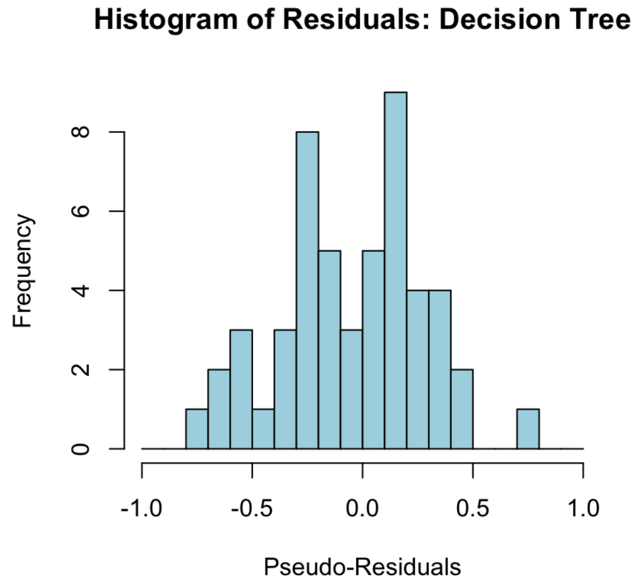


B: Time Series Plot of Residuals

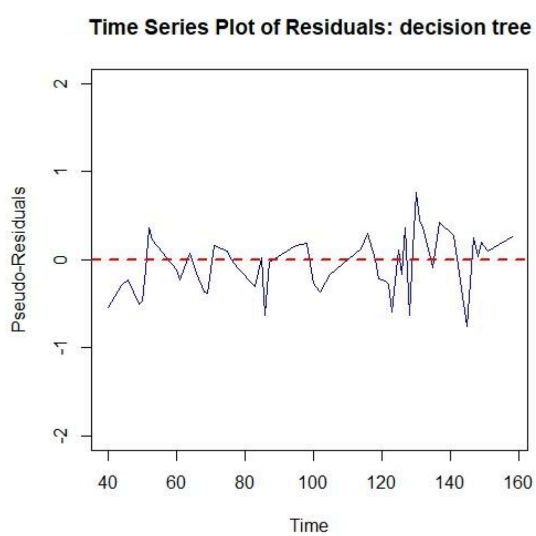


2.4: Decision Tree

A: Histogram of Residuals

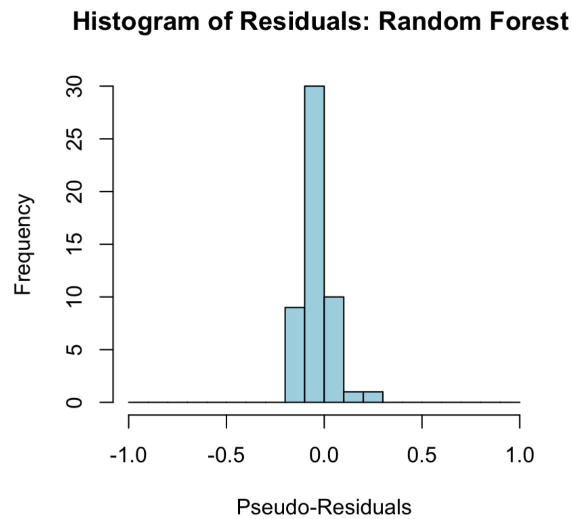


B: Time Series Plot of Residuals



2.5: Random Forest

A: Histogram of Residuals



B: Time Series Plot of Residuals

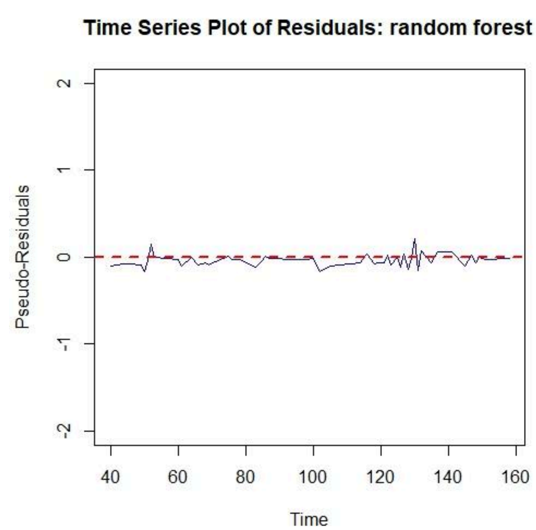


Exhibit 3: Features Importance for Random Forest Model

In [11]:
```
feature_importances = reg_rf.feature_importances_

# Print or use the feature importances as needed
for feature, importance in zip(X_train.columns, feature_importances):
    print(f"{feature}: {importance}")
```

```
store: 0.21683740195635748
lpr1: 0.4841427405406477
lpr2: 0.018089157941861375
lpr3: 0.024644636982054696
lpr4: 0.04397440848950512
lpr5: 0.018013384956192948
lpr6: 0.017490292339126405
lpr7: 0.024127343174447268
lpr8: 0.012942395731374191
lpr9: 0.016702395377248172
lpr10: 0.021363740581475263
lpr11: 0.017000129246021407
lpr12: 0.013589322219215988
afeat1: 0.055632815476982074
adisp1: 0.01544983498748991
```

Exhibit 4: Price Optimization

4.1: Pooled Model

Pooled Model

Brand	Description	Constant	lpr1	lpr2	lpr3	lpr4	lpr5	lpr6	lpr7	lpr8	lpr9	lpr10	lpr11	lpr12	afeat1	adisp1
1	TropPrem64	6.43	-2.67	0.17877	0.23021	0.16091	0.08865	0.35714	0.35621	-0.69847	0.35172	0.23093	0.04869	0.39032	0.59282	-0.02677
		Average		-3.028705	-3.114345	-3.353994	-3.368282	-3.180379	-3.328603	-3.205597	-3.391577	-3.452942	-3.632069	-3.567928	0.1596575	0.4273482
		Average (per oz)		-3.028705	-3.028705	-3.028705	-3.028705	-3.028705	-3.028705	-3.028705	-3.028705	-3.028705	-3.028705	-3.028705	-3.028705	-3.028705

Decision Variables

Brand	Description	Price/oz	Price/Unit	E[Move]	E[units]	Cost/oz	Cost/Unit	Profits
1	TropPrem64	$0.051	$3.29	1,815	28	0.0321	2.0545	$34.945

4.2: Store Model

Store Model

Brand	Description	Constant	store	lpr1	lpr2	lpr3	lpr4	lpr5	lpr6	lpr7	lpr8	lpr9	lpr10	lpr11	lpr12	afeat1	adisp1
1	TropPrem64	6.85		-2.106828337	0.426044279	0.021380607	0.108363747	0.117703578	0.655381308	0.161296638	-1.57366	0.74216742	0.08588657	0.38042646	0.121062157	0.410169723	0.05446403
		Average			-3.028705	-3.114345	-3.353994	-3.368282	-3.180379	-3.328603	-3.205597	-3.391577	-3.452942	-3.632069	-3.567928	0.410169723	1.05446403
		Average(per oz)			-3.028705	-3.114345	-3.353994	-3.368282	-3.180379	-3.328603	-3.205597	-3.391577	-3.452942	-3.632069	-3.567928	0.410169723	2.05446403
		Store 2(zone 1)															
		Store 5(zone 2)	-1.774463803	-1.037564807	-0.033332	0.37370587	0.08659366	0.11266936	0.932232975	0.13511843	-0.4417685	-0.152855	-0.2193935	-0.2350499	-0.0841403	0.04708519	-0.1912552
		Store 83(zone 6)	-0.800065102	-0.868844543	0.4053497	0.57374434	0.15503073	0.25220279	-0.691495607	0.44008	0.34357443	-0.7816512	0.16324035	0.17512734	-0.3435891	0.04444729	-0.1033615
		Store 112(zone 14)	3.091784189	-0.573131033	0.63891668	0.10026097	0.20134354	-0.3903344	-0.940840833	0.84018037	1.07381342	-0.7034512	0.31616983	-0.0871348	0.30021767	-0.0322996	0.05926478
		Store 115(zone 12)	-3.980124567	-0.810414549	-0.1124293	-0.674438	0.05004256	0.11378072	-0.202504602	0.26868373	0.11462489	-0.811044	0.04514103	-0.2128336	0.87996769	-0.0388593	0.00653376

Decision Variables

Brand	Description	Store	Price/oz	Price/Unit	E[Move]	E[units]	Cost/oz	Cost/Unit	Profits
1	TropPrem64	#2	$0.0611047020	$3.9107009290	5,978	93	0.0321	2.0545	$173.371
1	TropPrem64	#5	$0.0470757537	$3.0128482387	7,837	122	0.0321	2.0547	$117.331
1	TropPrem64	#83	$0.0483470877	$3.0942136123	6,331	99	0.0321	2.0544	$102.865
1	TropPrem64	#112	$0.0512100468	$3.2774429929	15,622	244	0.0321	2.0534	$298.770
1	TropPrem64	#115	$0.0488443847	$3.1260406222	9,247	144	0.0321	2.0545	$154.823

4.3: Mixed Model

Mixed Model

			Random Effect	Fixed Effect													
Brand	Description	Constant	Store	lpr1	lpr2	lpr3	lpr4	lpr5	lpr6	lpr7	lpr8	lpr9	lpr10	lpr11	lpr12	afeat1	adisp1
1	TropPrem64	4.55		-2.7669328	0.10138021	0.22191446	0.14458121	0.12899947	0.3119466	0.24665882	-0.9586358	0.3055439	0.05447124	0.56035534	0.07398005	0.54763663	-0.0489202
		Average			-3.028705	-3.114345	-3.353994	-3.368282	-3.180379	-3.328603	-3.205597	-3.391577	-3.452942	-3.632069	-3.567928	0.1596575	0.4273482
		Average(per oz)			-3.028705	-3.114345	-3.353994	-3.368282	-3.180379	-3.328603	-3.205597	-3.391577	-3.452942	-3.632069	-3.567928	0.1596575	0.4273482
		Store 2(zone 1)	0.168769292	0.070566141													
		Store 5(zone 2)	0.063173298	-0.029495683													
		Store 83(zone 6)	-0.064593411	-0.002249586													
		Store 112(zone 14)	0.2918201	-0.016555768													
		Store 115(zone 12)	0.073275515	-0.005906142													

Decision Variables

Brand	Description	Store	Price/oz	Price/Unit	E[Move]	E[units]	Cost/oz	Cost/Unit	Profits
1	TropPrem64	#2	$0.0510252295	$3.2656146850	5,580	87	0.0321	2.0545	$105.593
1	TropPrem64	#5	$0.0499756714	$3.1984429726	7,167	112	0.0321	2.0547	$128.076
1	TropPrem64	#83	$0.0502434119	$3.2155783644	5,727	89	0.0321	2.0544	$103.914
1	TropPrem64	#112	$0.0500746570	$3.2047780489	8,618	135	0.0321	2.0534	$155.028
1	TropPrem64	#115	$0.0502081891	$3.2133241052	6,659	104	0.0321	2.0545	$120.577

4.4: Random Forest Model

```
   Store  Calculated_Price  Highest_Profit
0      2              3.87      223.002933
1      5              3.66      197.208078
2     83              3.19      139.476737
3    112              3.19      139.476737
4    115              3.66      197.208078
```