Market models
Assignment 2

rijksuniversiteit
groningen

Table of Contents

1. Introduction

In the first assignment, we proposed **an original sales forecasting model (*)** for lemonade sales of Raak on the right. Considering the enthusiastic feedback from the members of the board who have all read the report on the first assignment, we proceed with the initial model for further estimation and validation with some new variables.

This report will demonstrate insights of how well the selected model is able to predict future observations since it is an important sales forecasting tool for future logistic and operational decisions. In the first section, a deeper look is taken at the variables within a simple multiplicative model, after which we dive into the estimation methods, validation and a final functional form.

2. Variable Description

In this section, there are seven additional independent variables based on the feedback besides the original independent variables which are *base price, feature, display* and *feature plus display*. Then, we briefly explain how these new variables are constructed in order to improve the model performance and the intuition behind it.
These newly added variables are ***temperature variable (T), PrivateLabel (PPL)***, ***Slimpie (PS)***, ***Sales Lag (SL)***, smoothened ***BasePrice (BP), Price Index (PI)*** and ***New COVID cases (C)*** which are included in the following sections regarding the corresponding factors we considered.

1. Seasonality and Temperature
Since the result of our ANOVA test (see Table A1) on quarters is significant ($p = 0.001$), we decided to include seasonality. Besides, lemonade is a specialty product with seasonal effects (Hoos, 1956). So we included ***a temperature variable (T)*** to ensure our model captures this effect. By doing a simple linear regression we again found that temperature is highly correlated with Sales for all brands and chains.

2. Competitors' Prices
Since the price of a competitor will also influence the Sales of the Raak brand (Farm, 2016), we included ***the price of both PrivateLabel (PPL)*** and ***Slimpie (PS)***, as they are the closest competitors in terms of sales and price.

3. Dynamic effects
The current effects model would not be the most ideal one as promotional activities might have pre- and post-sales effects. Furthermore, in-store promotional activities are associated with creating physical marketing elements and setting them up. This may create delayed-response effects, such as execution and noting delays (Leeflang et al., 2016). In order to measure such dynamic effects, we included a new variable ***Sales Lag (SL)***.

4. Price index and base price
Since the size of the promotion also has a large impact on Sales, we also added a price index derived from pricePU / BasepricePU. Moreover, BasePrice also has an effect on the Sales so we include both ***BasePrice (BP)*** and ***Price Index (PI)*** in the model.

5. COVID data

As COVID had a tremendous impact on Sales in the supermarkets, especially in 2020, we also thought including the COVID data would be of severe importance. We did so based on the GitHub data frame. We took weekly averages and only focussed on ***the New COVID cases (C)*** instead of the total because the number of new cases was the most important indicator for the Dutch government in terms of regulations.

6. Smoothened base price

For the ***BasePrice (BP)*** we used the smooth version for the cases in which the actual price was higher than the base price. We did so because we are looking into promotional effects and we consider an increase in the actual price as an increase in the base price instead of a promotion where the actual price is higher than the base price.

3. Estimation

Despite the log transformation being needed before estimating with the multiplicative model, the research of Dai et al. (2022) has demonstrated that interaction effects exist between display-related promotions and price discounts when modelling sales, suggesting that a multiplicative model may be more appropriate if we want to measure interaction effects and elasticity. Besides, from our correlation analysis among independent variables, they seem to have interesting interactions across the years for the Raak data. Our functional form decision for a multiplicative model is well-represented in and supported by past literature as described in the meta-analysis of Tellis (1988). So we decided to build **a simple multiplicative model** (non-linear in parameters but linearized) in the following sections.

Method 1: not pooled (unit-by-unit model)

Step 1: The above adjustments resulted in changes to our initial model. In the end, this has led to the model on the right.

$$S_{it} = \alpha_i \, BP_{it}^{\beta 1i} \, PI_{it}^{\beta 2i} PPL_{it}^{\beta 3i} PS_{it}^{\beta 4i} T_t^{\beta 5i} SL_{it}^{\beta 6i} \beta_{7i}^{C_t} \beta_{8i}^{F_{it}} \beta_{9i}^{D_{it}} \beta_{10i}^{FD_{it}} e^{\varepsilon_{it}}$$

with $i = 1, \ldots, 6$ brands, $t = 1, \ldots, 208$ weeks

where

S_{it} = sales of Raak at Chain i, in week t

α_i = Constant for Sales of Raak at Chain i

BP_{it} = Base Price of Raak at chain i, in week t

PI = Price Index Raak at chain i, in week t

PPL = Price Private Label at chain i, in week t

PS = Price Slimpie at chain i, in week t

T = Temperature (not per chain)

SL = Sales Lag of the Raak at chain i, in week t

C = New covid cases (not per chain) in week t

F = Feature only of Raak at chain i, in week t

D = Display only of Raak at chain i, in week t

FD = Feature and Display of Raak at chain i, in week t

Step 2: For a certain chain, after natural log transformation on the first equation (*), we are able to run a linear regression in the following equation (^):

$$\ln S_{it} = \ln(\alpha_i) + \beta_{1i} \ln BP_{it} + \beta_{2i} \ln PI_{it} + \beta_{3i} \ln PPL_{it} + \beta_{4i} \ln PS_{it} + \beta_{5i} \ln T_t + \beta_{6i} \ln SL_{it} + C_t \ln \beta_{7i} + F_{it} \ln \beta_{8i} + D_{it} \ln \beta_{9i} + FD_{it} \ln \beta_{10i} + \ln \varepsilon_{it}$$ *, which is also written as*

$$S_{it}^* = (\alpha_i)^* + \beta_{1i} BP_{it}^* + \beta_{2i} PI_{it}^* + \beta_{3i} PPL_{it}^* + \beta_{4i} PS_{it}^* + \beta_{5i} T_t^* + \beta_{6i} SL_{it}^* + C_t \beta_{7i}^* + F_{it} \beta_{8i}^* + D_{it} \beta_{9i}^* + FD_{it} \beta_{10i}^* + e\hat{}\varepsilon_{it}^*$$

Step 3: With antilog transformation, we can get the original parameters in equation *. For example, we can get $\alpha\hat{}$ by applying $\hat{\alpha} = \exp(\ln(\hat{\alpha})) \exp(-\frac{1}{2} var(\ln(\hat{\alpha})))$, with $\ln(\alpha\hat{}) = \alpha\hat{}^*$. In a

similar fashion, we can get β_2, β_3, etc. Specifically, $\beta_1 \beta_8$ are elasticity estimates that indicate the percentage change in that variable according to the percentage change in price.

Method 2 & Method 3: pooled model and partially pooled model

To make sure our model has the best fit with reality, we performed the chow test to test whether pooling or partially pooling is allowed. The table on the right shows the degrees of freedom per model.

df_pooled	(7*207) - 11 = 1438
df_partially	7*207 - 17 = 1432
df_unpooled	196(AH)+196(Coop)+198(Deen)+196(Hoogvliet)+198(Jumbo)+196(Plus)+198(Online) = 1378

Unpooled vs. Fully pooled: We performed the chow-test by computing the F-statistic with the Residuals' Sum of Squares for the unpooled model vs. the fully pooled model. We found an F-statistic of 65.895, which indicates a p-value of 0.00. This indicates that a fully pooled model is not allowed.

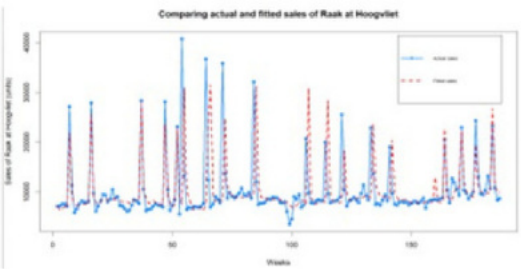
Unpooled vs. Partially pooled: We performed the chow-test by computing the F-statistic with the Residuals' Sum of Squares for the unpooled model vs. the partially pooled model. We found an F-statistic of 14.993, which also indicates a p-value of 0.00. This indicates that a partially pooled model is also not allowed.

Consequently, we decided not to pool our data and to **continue with an unpooled (unit-by-unit) model**. We decided to use the Hoogvliet chain as our selected chain for further analysis. Then we first ran a linear regression analysis on the model and before we continue with testing all the assumptions, we first discuss the initial significance of the model as it currently exists.

4. Validation

Model fit

The graph on the right shows a comparison of the actual sales of Raak at Hoogvliet (blue line) and the fitted sales (red line), according to our current model. The graph clearly shows that the fit between the model and the actual sales is already quite good. We are going to test this more formally after testing for the assumptions.



Significance of initial unit-by-unit model

To be able to test for validation, we split our dataset into two parts. We used the first 80% of our data for estimation and the last 20% (from 01-08-2022) for validation.

Overall our model (see Table A2) was statistically significant ($(F(10, 175) = 78.4, p < .001)$) with a high model fit ($R^2 = 0.82$; Adj. $R^2 = 0.81$). When examining the individual parameters, found the following variables to be statistically significant: Price per Unit for Private Label ($p = 0.017$),

Feature Only ($p < .001$), Feature and Display ($p < .001$), Lagged Unit Sales ($p < .001$), Temperature ($p < .001$), Weekly Avg. COVID Cases ($p = 0.003$). This thus indicates that some of the variables which we would expect to be significant, based on the literature, are not that important. We see that surprisingly the Price Index is only slightly significant ($p = 0.083$) and for the two competitor variables we have included, only the Price of Hoogvliet's private label is significant ($p = 0.017$). However, since we haven't tested and solved the issues regarding all the assumptions, we are not yet able to draw conclusions from these coefficients.

Multicollinearity

After deciding to use the unit-by-unit model for the Hoogvliet chain we checked for multicollinearity. By looking at the VIF values, we found high multicollinearity for one variable, namely the log(PriceIndex), with a VIF of around 5.504. We found relatively high correlations between the Price Index and the promotional variables. To fix the multicollinearity issue we decided to split the PriceIndex variable into four different variables:

1. **pf1:** Price Promotion combined with Feature advertising support
2. **pd1:** Price Promotion combined with Display advertising support
3. **pfd1:** Price Promotion combined with Feature & Display advertising support
4. **pwo1:** Price Promotion with no advertising support

And we adjusted the three promotional variables, so they only take into account the observations in which there is no price cut:

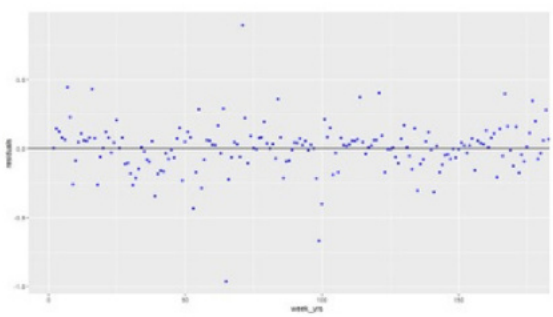
5. **fwo1:** Feature advertising with no Price Promotion support
6. **dwo1:** Display advertising with no Price Promotion support
7. **fdwo1:** Feature and Display advertising with no Price Promotion support.

By looking at histograms and other plots we decided to use 17 as the cut-off point for the promotion variables. This means that in at least 17% of the stores there needs to be a specific promotion to categorize it as "support". For the PriceIndex we took a cut-off point of 0.76. This means that the actual price must be at least 24% lower than the base price to categorize it as a promotion.

After doing so, we found that the VIF values for all the variables in the new model were < 5.0, indicating no multicollinearity. For the PriceIndex variable with DisplayOnly support (pd1), we had fewer than 5 observations. Therefore, we eliminated this variable from our model. After this, all VIF values remained < 5.0. These modifications resulted in a statistically significant model ($F(11, 174) = 54.6$, $p < .001$) with a slightly lower but still excellent model fit ($R^2 = 0.78$; Adj. $R^2 = 0.76$), visible in Table A4. Furthermore, we had the following significant variables: Price per Unit for Private Label ($p = 0.010$), Price Promotion with Feature advertising ($p < .001$), Price Promotion with Feature & Display advertising ($p < .001$), Feature and Display advertising with no Price Promotion support ($p < .001$), Lagged Unit Sales ($p < .001$), Temperature ($p < .001$) and Weekly Avg. COVID Cases ($p = 0.003$).

Heteroscedasticity

To make sure the residual variance of the error term does not differ over time, we checked for heteroscedasticity. The plot on the right shows the



residuals over time and this does not indicate heteroscedasticity. However, we need to check this more formally. We checked the influence of 3 variables on the variance of the Unit Sales. We did so by performing the Goldfield-Quandt test.

Effect of promotions
We decided to define promotion as a situation in which the Price Index dropped below 0.95. This resulted in an F-statistic of 5.1982 and a p-value of 0.999. This indicates that there is no heteroscedasticity, meaning that the residuals of the Unit Sales do not differ during promotional weeks.

Effect of warm days
We decided to define a warm day as a situation in which the temperature rises above 18 degrees Celsius. This resulted in an F-statistic of 5.6843 and a p-value of 0.999. This indicates that there is no heteroscedasticity, meaning that the residuals of the Unit Sales do not differ during warm days.

Effect of COVID
To compare the effect of COVID on the variance, we compared the period before the first COVID case with the period after the first registered COVID case. This resulted in an F-statistic of 0.09382 and a p-value of 0.000674. This indicates that there is heteroscedasticity, meaning that the residuals of the Unit Sales do differ after COVID started. We attempted to treat this with a GLS model (see Table A3), however, it performed significantly worse in terms of predictive validity, which is why we use the non-GLS as our main model. We will discuss this further in the conclusion, as there are implications to consider for both models.

Non-normality

Then, we detected there is non-normality in the distribution of residuals under the Lilliefors (Kolmogorov-Sirnov) normality test with a significantly small p-value of 4.963e-08. Further, the Shapiro-Wilk normality test and the Jarque-Bera normality test also showed significantly small p-values indicating that the normality assumption is violated.
By applying a remedy of a bootstrapped version of our model, we can see the majority of the significant explanatory variables still remain significant as they are in the OLS model (see Table A5). However, without the assumption of normality under the bootstrapped version, we can get the true level of significance from the corresponding p values in the empirical distribution of residuals.

Autocorrelation

Another OLS assumption is autocorrelation, which we tested in three ways. Firstly, we looked at the Durbin-Watson test, which was significant ($p = 0.008$) with a DW value of 1.687. After checking the Durbin-Watson table we found the lower and upper limits ($dL = 1.561$; $dU = 1.791$) based on $k' = 11$ and $n = 186$. This classifies our DW value in the "grey zone", making the Durbin-Watson test not definitive and presenting a need for further testing. Next, we opted to run a visual test, namely an ACF plot (see Figure A1). After the lag-0 correlation, the subsequent

correlations drop quickly to zero and stay between the limits of the significance level indicated with blue dashed lines (Coding Prof, 2022). Therefore, we can conclude that the residuals of this model meet the assumption of non-autocorrelation. Finally, we conducted a Breusch-Godfrey test, which checks for autocorrelation among residuals of the first-order, second-order, third-order, etc. Our results were insignificant ($p = 0.081$) indicating we cannot reject the null hypothesis and conclude that there is no significant autocorrelation in our model.

Unusual data points

First, we applied the student residuals to detect outliers, measured the multivariate distance between the point and its average to detect leverage points and Cook's D to detect influential points (see Figure A2).

Then, a table containing all the above statistics was constructed (see Table A7) where we conclude that observation 4528 is a high-leverage point (but no outlier) and has the highest influence of all on the regression and observations 4440, 4483, 4434 are high-leverage outlier points with moderately high influence on the regression.

Finally, we did the robust estimation in R in order to weigh down these unusual data points according to their detrimental influence on the estimation. And we can also see it is significantly different from the null model due to the ANOVA test.

Significance of our final model & parameters

The statistical results of our final model based on the Hoogvliet chain can be found in the table on the right. When examining the F-statistic, $F(11,174) = 54.63$ ($p < .001$), we can see that our final model is highly significant. Furthermore, the variables in our model explain 76.1% of the variability in Unit Sales ($R^2 = 0.78$; Adj. $R^2 = 0.76$), indicating a good model fit (see Table A6).

Next, we examine the statistically significant predictors individually. The p-values of our parameters are based on the bootstrapped version of our model treated for nonnormality.

- **Price per Unit (Private Label):** The price/unit index of Hoogvliet's private label has a significant positive effect on Unit Sales ($p = .008$), indicating that an x unit increase in the price of private label lemonade leads to the unit sales for Raak lemonade to be multiplied by $x^{0.63}$. In practice, this means that as Hoogvliet's private label increases its price, Raak's unit sales also increase. Inversely, if the private label lowers its price, Raak might lose sales.
- **Price Promotion (with Feature advertising):** Price promotion supported by feature advertising has a significant negative effect on Unit Sales ($p = .011$), indicating that an x unit increase in the usage of price promotion combined with feature advertising leads to multiplying unit sales by $x^{-1.90}$. As this variable is related to the price index when it increases it implies a smaller discount. In practice, this means that when the price goes up, even with feature advertising support, the sales would decrease. Vice versa, when the discount grows supported by feature advertising the sales exponentially increase.
- **Price Promotion (with Feature & Display advertising):** Price promotion supported by feature and display advertising has a significant negative effect on Unit Sales ($p < .001$),

indicating that an x unit increase in the usage of these marketing actions leads to Raak's unit sales being multiplied by $x^{-1.40}$. This implies the same effect as the one proposed by the results of the previous variable. In essence, Raak's sales are significantly augmented by higher price discounts when those are supported by Feature & Display advertising.

- **Lagged Unit Sales:** The lagged unit sales have a significant positive effect on Unit Sales ($p = .012$), indicating that an x unit increase in sales in previous periods leads to the unit sales being multiplied by $x^{0.16}$. In practice, this implies that the extent to which we did well with our sales in the past will contribute to doing well in our current and future sales.
- **Temperature:** Our regression results show the temperature to have a significant positive effect on Unit Sales ($p < .001$), indicating that an x unit increase in temperature will lead to unit sales being multiplied by $x^{3.23}$. In practice this showcases that as temperatures rise and it becomes warmer, people will exponentially purchase more lemonade, suggesting a seasonality effect. Inversely, it also means that people purchase a lot less lemonade when temperatures fall. For example, if it is 32 degrees the sales equation is multiplied by $(273+20)^{3.23} = 1.06e8$. Meanwhile, if the temperatures are negative (-15 degrees), as in winter, the multiplier is two times less (0.62e8).
- **COVID Cases (Avg. per Week):** The average weekly COVID cases have a significant positive effect on Unit Sales ($p = 0.006$), indicating that x unit increase in COVID cases leads to unit sales being multiplied by 1.00027^x. Considering the coefficient value being close to 1, it shows that COVID has nearly no influence up to a certain number of weekly cases, but once it surpasses a certain value the influence grows extremely large. To illustrate, if there are 15,000 cases the unit sales are multiplied by 57.37. Now, if the cases in the next week double to 30,000; the number with which unit sales are multiplied grows to 3290.87, a 57 times larger influence.

After examining the statistical validity of our final model and the individual parameters, it is important to consider the face validity of our results, whether they are consistent with previous research and a logical expectation of reality.

Face Validity

Firstly, we consider our result that an increase in the prices of Hoogvliet's private label leads to an increase in Raak's sales. This is a well-documented effect in retail sales, where when competitors increase their prices, demand for their own products goes up. This happens because price-sensitive consumers who have no brand loyalty migrate to the less expensive option (Pesendorfer, 2002). Furthermore, Van Heerde et al. (2003) found that approximately 33% of unit sales increase due to promotion cannibalized from competitor brands, confirming the relationship between the private label's price and Raak's unit sales.

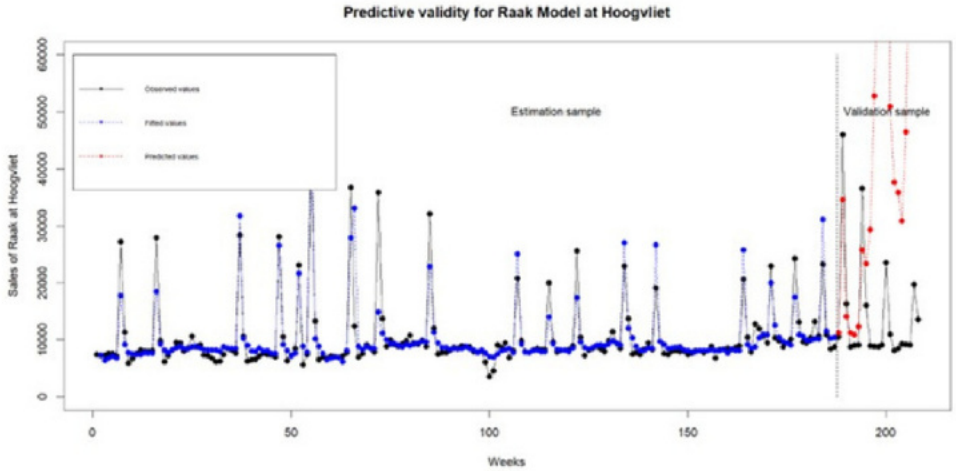
Next, according to our model, price promotion combined with feature advertising and price promotion combined with feature and display advertising affect unit sales positively. Previous research, such as this by Woodside & Waddle (1975) has found that price discounts combined

with point-of-sale advertising increased purchases even more than only advertising or price promotion alone, which is also what our model suggests. We also consider the findings of Sethuraman & Tellis (2002), who found that price promotions combined with advertising can have both a positive and negative effect based on the role that advertising plays. For example, advertising has a positive effect only when it is informative enough to increase consumer response to promotions. Therefore, our findings might suggest that Raak's advertising at Hoogvliet is informative enough to consumers and therefore has a positive effect. Furthermore, it is important to consider that the Raak does not do many discounts, in fact approximately 70% of observations at Hoogvliet indicate no discount (see Figure A3). This aspect of our brand may also influence the results of our model.

Next, we consider that our model indicated the significant positive effect of unit sales of previous periods on current and future unit sales. According to Parsons et al. (1976) lagged sales are related positively to current-period sales due to the effect of consumer inertia or loyalty. In contrast, a meta-analysis of econometric models of sales by Tellis (1988), which combined 424 models from 42 studies yielded a negative effect of lagged sales on current sales, however, the result was not significant ($p = 0.12$), making the face validity of our result likely.

Finally, we consider Temperature and COVID. Firstly, when it comes to lemonade and temperature there is a long-standing literature showcasing the seasonal nature of lemonade sales (Hoos, 1956). Our model also showcases this, indicating that Raak's lemonade sales at Hoogvliet increase, when temperatures rise. Next, our model showed a significant positive effect of COVID Average cases per week on unit sales. We attribute this to the "panic buying" effect discovered by the research of Eriksson & Stenius (2020) during the first months of the pandemic when people were afraid of supplies not lasting and stockpiling lasting grocery items. Lemonade may fall under this category as it does not spoil for a long time, making the panic buying effect likely for lemonade, thereby resulting in a positive coefficient in our model.

Predictive validity



Prediction Criterion	Value
APE	-64213.43
ASPE	14243994
RASPE	119348.2
MAPE	4.857552
RAE	7.828423
TheilsU	8.692934

Judging by the plot above, we can see that our model fits well in the estimation sample, as well as for the first 8 weeks in the validation sample, this partially proves that our model is robust in the short term, but unfortunately not effective when given a longer period of time. Some possible explanations can be derived from the following results of prediction error tests. APE has a high negative value, which means after the first 8 weeks, the predictions go much higher than the y value of data points in validation samples; ASPE, a measurement having large errors weighted, reveals that there are quite a lot of data points in predictions after the first 8 weeks going beyond the upper limit. Also, RAE and TheilsU results show that the predictive ability of our model still yet needs to be improved. However, one possible reason why the prediction goes higher than normally expected in the later stage might be due to the sample bias in COVID data, since our model is mostly trained on weeks without COVID (before March 2020), therefore it is hard for the model to adequately capture the effect in predicting future values. Especially since our model was trained on the estimate sample that only includes COVID data with a maximum number of weekly cases of 1,086 and where most values are even lower than 200. The validation dataset could contain COVID data with more than 10,000 weekly cases, which can have a large impact on the predicted values, while the actual values do not increase as much.

(Note: we did antilog transformation for both **UnitSales** fitted values and predictions, which is also one of the reasons why APE, ASPE, and RASPE values are high.)

5. Conclusion & limitations

In the end, we decided on the final functional form after estimation and validation as follows.

$$S_{it} = \alpha_i \, PS_{it}^{\beta 1i} \, PPL_{it}^{\beta 2i} \, pf1_{it}^{\beta 3i} \, pfd1_{it}^{\beta 4i} \, pwo1_{it}^{\beta 5i} \, \beta_{6i}^{fwo1_{it}} \, \beta_{7i}^{dwo1_{it}} \, \beta_{8i}^{fdwo1_{it}} \, SL_{it}^{\beta 9i} \, (T_t + 273)^{\beta 10i} \, \beta_{11i}^{C_t} \, e^{\varepsilon_{it}}$$

- *with $i = 1, \ldots, 7$ chains, $t = 1, \ldots, 208$ weeks*
- *where*

  S_{it} = sales of Raak at Chain i, in week t

  α_i = constant for Sales of Raak at Chain i

  PS = price Slimpie at chain i, in week t

  PPL = price Private Label at chain i, in week t

  pf1 = price index if there is feature-only support at chain i, in week t

  pfd1 = price index if there is feature and display support at chain i, in week t

  pwo1 = price index if there is no support at chain i, in week t

  fwo1 = feature support but no price cut at chain i, in week t

  dwo1 = display support but no price cut at chain i, in week t

  fdwo1 = feature and display support but no price cut at chain i, in week t

  SL = Sales Lag of the Raak at chain i, in week t

  T = Temperature (not per chain)

  C = New covid cases (not per chain) in week t

And the Hoogvliet chain is selected to examine the prediction of Raak sales as an example:

$$S_t = 6.1e\text{-}9\ PS_t^{\ 0.03}\ PPL_t^{0.63} pf1_t^{-1.9} pfd1_t^{-1.4} pwo1_t^{1.51} 1^{fwo1_t} 0.99^{dwo1_t} 1.01^{fdwo1_t} SL_t^{0.16} (T_t + 273)^{3.23} 1^{C_t} e^{\varepsilon_t}$$

In conclusion, we are convinced that this model can be used for the prediction of future sales, since it is highly significant. As explained above, we tested all the required assumptions and made the required adjustments to encounter the problems we faced. However, we do need to mention that COVID has a large negative effect on the quality of our model. We found heteroscedasticity issues for this variable. We have tried a lot of ways to solve this issue, one of which is a GLS (see Table A3). Unfortunately, this had a severe negative impact on the overall quality of our model so much so that we decided not to continue with the GLS version. On top of that, we found unreasonable predictions in the later weeks from our dataset (see Figure A4). The most reasonable explanation for these errors are the huge differences in terms of weekly COVID cases between the estimation part and validation part of our dataset. Therefore, we assume this model should be applied only in situations where the amount of COVID cases is low or zero. To be able to come up with a solution for this COVID issue, further research is required with a dataset that contains a more balanced dataset in terms of observations with vs. without COVID cases.

Nevertheless, our model still provides interesting practical insights for managers. Firstly, when it comes to competitor prices, Raak needs to keep in mind the cannibalization effect (Van Heerde et al., 2003) resulting from competitors lowering their prices. We recommend Raak track prices and adjust their price accordingly to minimize this effect and avoid losing sales to other brands in their segment. Furthermore, Raak can use these insights to strategically lower their prices relative to competitors in order to acquire additional customers. Next, when it comes to price promotions and advertising, our model shows the importance of combining these. According to our results even with advertising support a price that is not significantly discounted will not contribute to sales. We recommend Raak plan their marketing campaigns during times when they also plan to offer a discount, in order to maximize the positive effect on sales. Furthermore, combining our results with the findings of Sethuraman & Tellis (2002), we recommend Raak focus on more informative marketing campaigns. Next, considering crisis times such as COVID, our model and previous research (Eriksson & Stenius, 2020) showcases the “panic buying” effect at the beginning of crisis times. Therefore, if an event of this nature is anticipated or happens abruptly, Raak may consider temporarily ramping up production to cover the initial peak in demand. Finally, when it comes to temperature, both our model and well-established previous research have shown the seasonal nature of lemonade. Therefore, Raak should plan their production, as well as marketing efforts, in a way to accommodate the increased demand during warmer months.

References

Coding Prof. (2022, May 16). 3 Easy Ways to Test for Autocorrelation in R. CodingProf.com. https://www.codingprof.com/3-easy-ways-to-test-for-autocorrelation-in-r-examples/

Dai, H., Ge, L., Li, C., & Wen, Y. (2022). The interaction of discount promotion and display-related promotion on on-demand platforms. Information Systems and e-Business Management, 20(2), 285-302.

Eriksson, N., & Stenius, M. (2020). Changing Behavioral Patterns in Grocery Shopping in the Initial Phase of the Covid-19 Crisis—A Qualitative Study of News Articles. Open Journal of Business and Management, 8, 1946-1961

Farm, A. (2016). Pricing and price competition in consumer markets. Journal of Economics, 120(2), 119-133.

Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables. R package version 5.2.3. https://CRAN.R-project.org/package=stargazer

Hoos, S. (1956). Lemon industry in California: Long-term projection of market potential for lemon juice products based on variable determinants of summer demand. California Agriculture, 10(10), 2-15.

Lal, R., & Matutes, C. (1994). Retail pricing and advertising strategies. Journal of business, 345-370.

Leeflang, P. S. H., Wieringa, J. E., Bijmolt, T. H. A., & Pauwels, K. H. (2016). Modeling Markets. Springer-Verlag New York.

Parsons, L. J., Hall, R. A., & Schultz, R. L. (1976). Marketing models and econometric research. North-Holland.

Pesendorfer, M. (2002). Retail sales: A study of pricing behavior in supermarkets. The Journal of Business, 75(1), 33-66.

Sethuraman, R., & Tellis, G. (2002). Does manufacturer advertising suppress or stimulate retail price promotions? Analytical model and empirical analysis. Journal of Retailing, 78(4), 253-263.

Tellis, G. J. (1988). The price elasticity of selective demand: A meta-analysis of econometric models of sales. Journal of marketing research, 25(4), 331-341.

Van Heerde, H. J., Gupta, S., & Wittink, D. R. (2003). Is 75% of the sales promotion bump due to brand switching? No, only 33% is.

Woodside, A. G., & Waddle, G. L. (1975). Sales effects of in-store advertising. Journal of Advertising Research, 15(3), 29-33.

Appendix A

Tables & Figures

Table A1: ANOVA Seasonality

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Quarter	3	2.735e+10	9.116e+09	5.055	0.00169 **
Residuals	8604	1.551e+13	1.803e+09		

Table A2: Initial unit-by-unit model

	Dependent variable:
	Unit Sales
Smooth Base Price	−0.060
	(0.422)
Price Index	−0.595
	(0.342)
Price per Unit (Slimpie)	0.020
	(0.147)
Price per Unit (Private Label)	0.560*
	(0.232)
Feature Only	0.005***
	(0.001)
Display Only	−0.005
	(0.005)
Feature and Display	0.013***
	(0.001)
Lagged Unit Sales	0.180***
	(0.034)
Temperature	3.050***
	(0.659)
COVID Cases (Avg. per week)	0.0002**
	(0.0001)
Intercept	−10.093**
	(3.674)
Observations	186
R^2	0.818
Adjusted R^2	0.807
Residual Std. Error	0.171 (df = 175)
F Statistic	78.397*** (df = 10; 175)
Note:	*p<0.05; **p<0.01; ***p<0.001

Table A3: Heteroscedasticity-treated Model (GLS Model)

	Dependent variable:
	Unit Sales
Price per Unit (Slimpie)	0.064
	(0.154)
Price per Unit (Private Label)	0.757**
	(0.247)
Price Promotion (with Feature advertising)	−2.498***
	(0.274)
Price Promotion (with Feature and Display advertising)	−0.789***
	(0.318)
Price Promotion (No advertising)	2.022***
	(1.753)
Feature advertising (No Price Promotion)	−0.002
	(0.002)
Display advertising (No Price Promotion)	0.001
	(0.007)
Feature and Display advertising (No Price Promotion)	0.007***
	(0.002)
Lagged Unit Sales	0.160***
	(0.038)
Temperature	2.991***
	(0.716)
COVID Cases (Avg. per week)	0.0002*
	(0.0001)
Intercept†	−9.674***
	(4.013)
Observations	186
R^2	1.000††
Adjusted R^2	1.000††
Residual Std. Error	1.158 (df = 174)
F Statistic	39,156.020*** (df = 12; 174)
Note:	*p<0.05; **p<0.01; ***p<0.001

† The GLS model's original intercept is replace by this variable.

†† Due to the lack of an original intercept the R-Square values cannot be interpreted.

Table A4: Multicollinearity-treated Model

	Dependent variable:
	Unit Sales
Price per Unit (Slimpie)	0.033
	(0.157)
Price per Unit (Private Label)	0.634**
	(0.242)
Price Promotion (with Feature advertising)	−1.898***
	(0.376)
Price Promotion (with Feature and Display advertising)	−1.395***
	(0.376)
Price Promotion (No advertising)	1.512
	(1.632)
Feature advertising (No Price Promotion)	−0.0003
	(0.002)
Display advertising (No Price Promotion)	−0.005
	(0.007)
Feature and Display advertising (No Price Promotion)	0.008***
	(0.002)
Lagged Unit Sales	0.162***
	(0.038)
Temperature	3.231***
	(0.708)
COVID Cases (Avg. per week)	0.0003**
	(0.0001)
Intercept	−10.994**
	(3.978)
Observations	186
R^2	0.775
Adjusted R^2	0.761
Residual Std. Error	0.190 (df = 174)
F Statistic	54.633*** (df = 11; 174)
Note:	*p<0.05; **p<0.01; ***p<0.001

Table A5: Bootstrapped & Original p-values

Variable	Bootstrapped p-value	Original p-value
Price per Unit (Slimpie)	0.387	0.835
Price per Unit (Private Label)	0.008 **	0.010 **
Price Promotion (with Feature advertising)	0.011 *	< .001 ***
Price Promotion (with Feature and Display advertising)	NA†	< .001 ***
Price Promotion (No advertising)	0.187	0.356
Feature advertising (No Price Promotion)	0.378	0.846
Display advertising (No Price Promotion)	NA†	0.469
Feature and Display advertising (No Price Promotion)	0.257	< .001 ***
Lagged Unit Sales	0.013 **	< .001 ***
Temperature	< .001 ***	< .001 ***
COVID Cases (Avg. per week)	0.006	0.003
Intercept	0.009 **	0.006 **

Significance codes: * 0.001 ** 0.01 * 0.05**

† Too few observations

Table A6: Final model for Raak at Hoogvliet

	Dependent variable:
	Unit Sales
Price per Unit (Slimpie)	0.03 (0.157)
Price per Unit (Private Label)	0.63** (0.242)
Price Promotion (with Feature advertising)	−1.90* (0.376)
Price Promotion (with Feature and Display advertising)	−1.40*** (0.376)
Price Promotion (No advertising)	1.51 (1.632)
Feature advertising (No Price Promotion)	1.00(0.002)
Display advertising (No Price Promotion)	0.99(0.007)
Feature and Display advertising (No Price Promotion)	1.01 (0.002)
Lagged Unit Sales	0.16* (0.038)
Temperature	3.23*** (0.708)
COVID Cases (Avg. per week)	1.00027** (0.0001)
Intercept	6.15e−9** (3.978)
Observations	186
R^2	0.775
Adjusted R^2	0.761
Residual Std. Error	0.190 (df = 174)
F Statistic	54.633*** (df = 11; 174)
Note:	*p<0.05; **p<0.01; ***p<0.001

Table A7: Unusual Data Points

Unusual data points	Studentized residuals	Hat	Cook's D
4434	-6.290424	0.2061735	0.7010263
4440	5.847877	0.2232314	0.6877700
4483	3.039328	0.5631039	0.9473182
4528	1.938606	0.8691999	2.0487040

Figure A1: ACF Plot

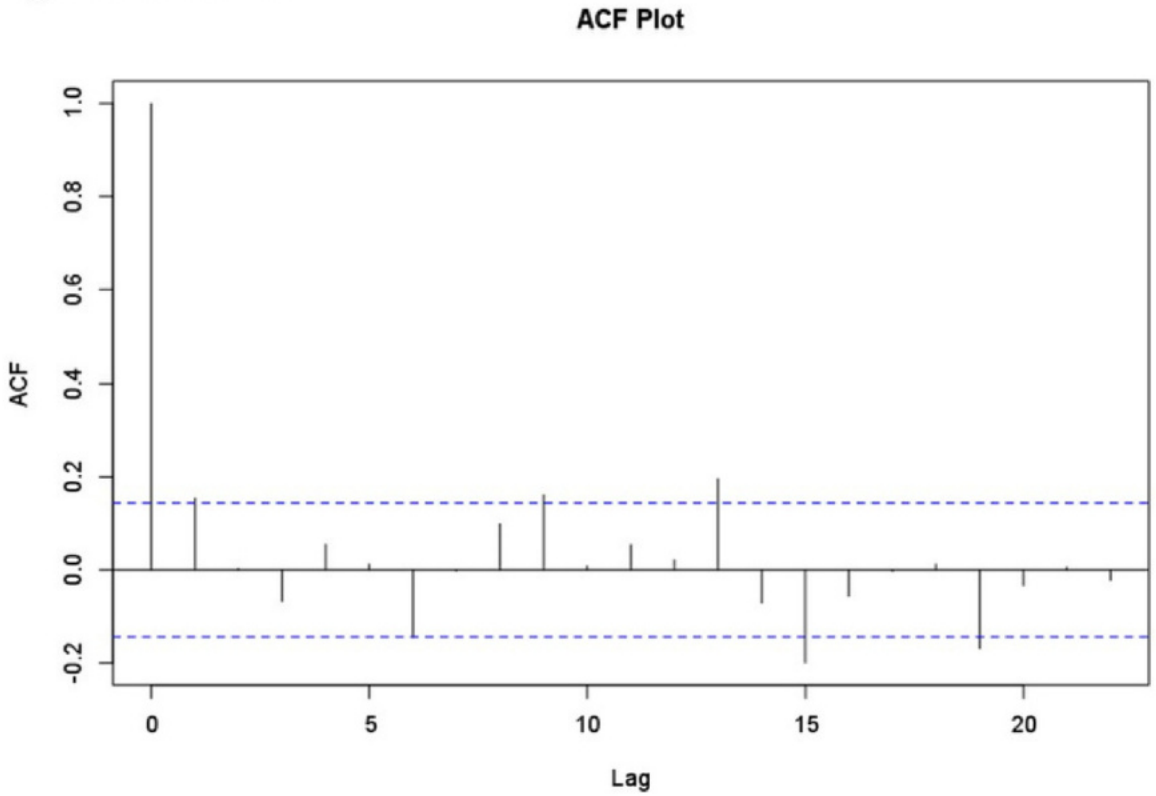


Figure A2: Influence Plot

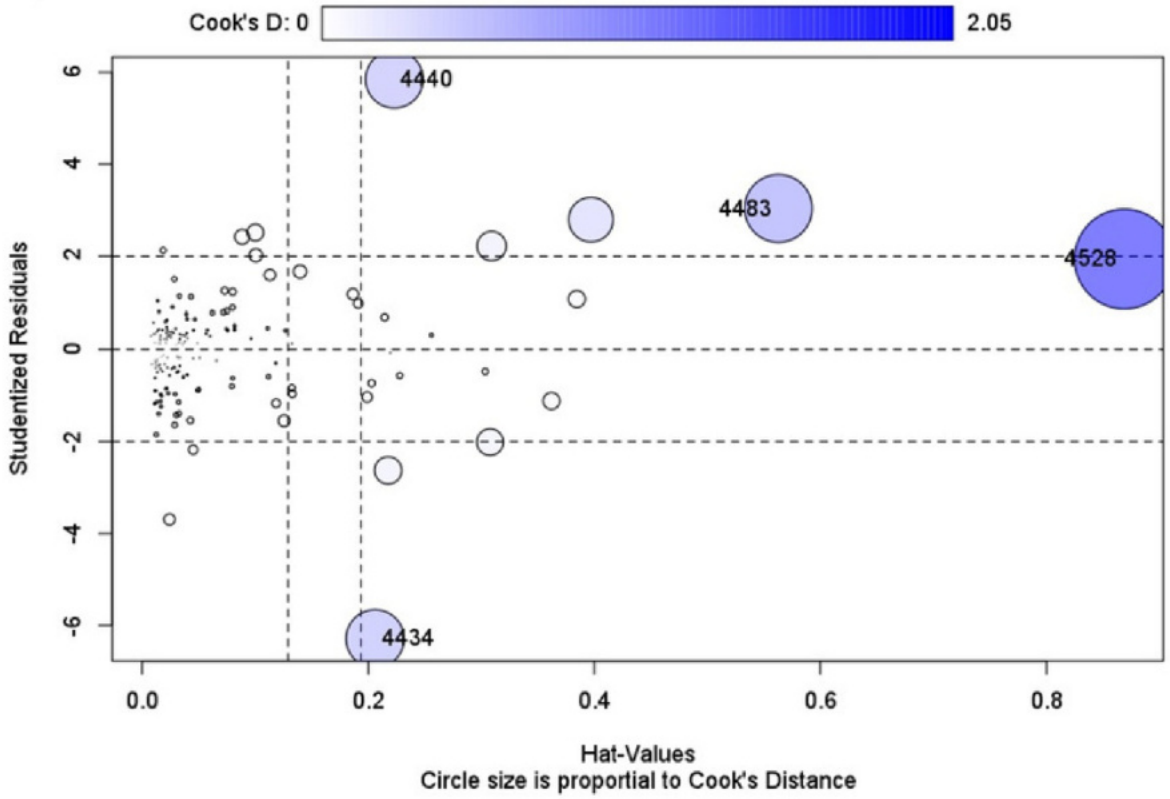


Figure A3: Histogram of Discounts of Raak at Hoogvliet

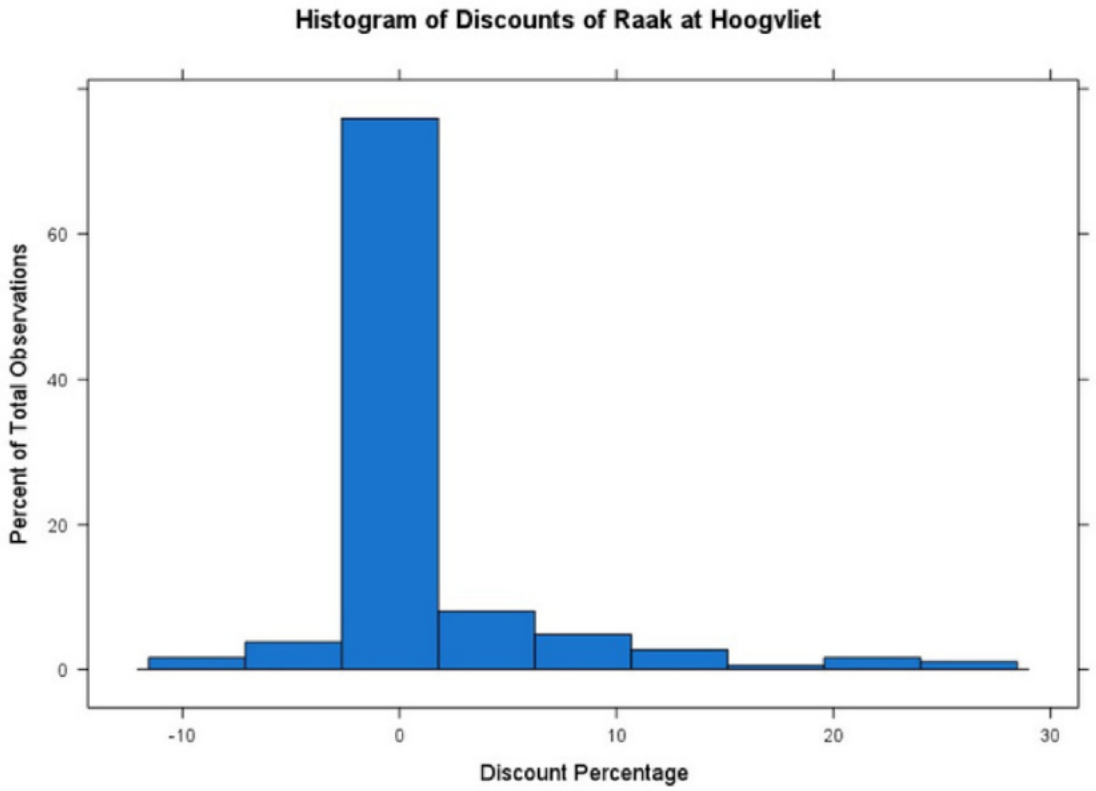
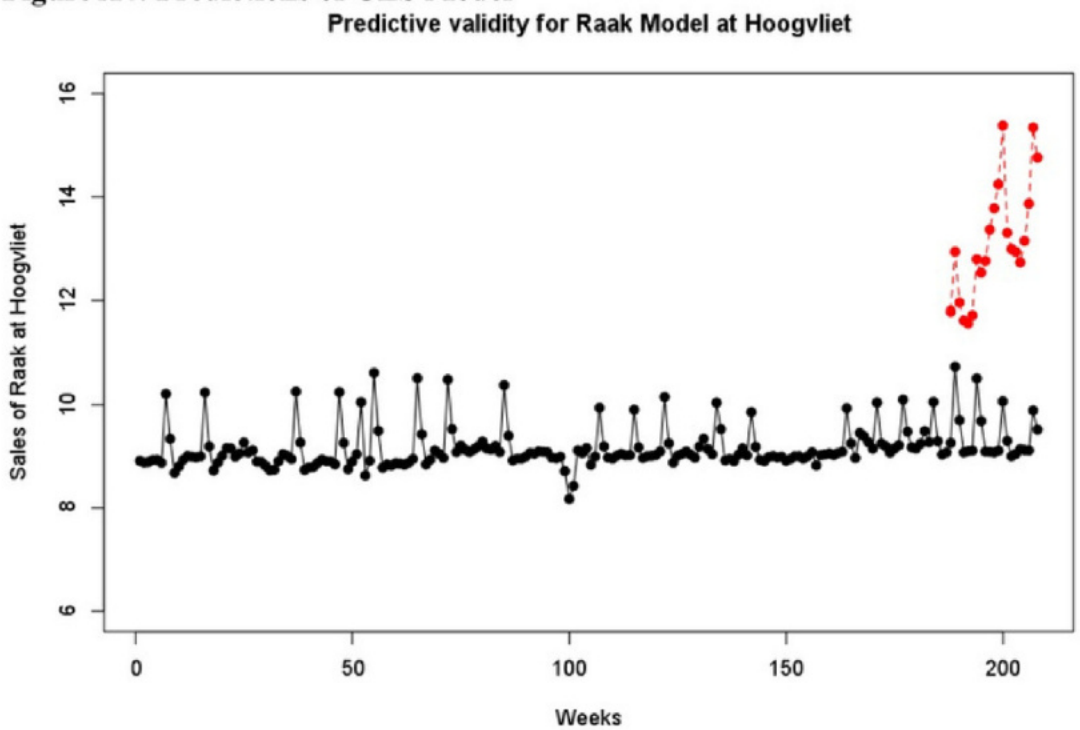


Figure A4: Predictions of GLS Model



Appendix B

R Script

```r
#############################################################
#############
##                            Preparation
                             ##
#############################################################
#############

#clear workspace
rm(list = ls())

##::::::::::::::::::##
##  Installations  ##
##::::::::::::::::::##
# INSTALLATIONS ----

# Handy package that Installs, Updates and Loads
# packages from CRAN, Github & Bioconductor
# install.packages("librarian")
library("librarian")

#install & load packages with shelf function
librarian::shelf(RColorBrewer, lmtest, aTSA, ggplot2
, ggcorrplot, dplyr, fastDummies,
                 car,corrplot,dplyr, vars, urca,
mclust, tidyr, reshape2, ISOweek, lubridate, mice,
tidyverse, VIM, quiet = TRUE)

##::::::::::
##  Data  ::
##::::::::::
# DATA ----
setwd("C:/Users/CHEN XINGMING/Desktop/Market Model")
#set working directory#
lemonade <- read.csv("Lemonade Tidy format 2022.csv"
, header = TRUE)
lemonade$Chain <- as.factor(lemonade$Chain)
lemonade$Brand <- as.factor(lemonade$Brand)
lemonade$Date <- ISOweek2date(sub("(wk) (\\d{2}) (\\d
{2})","20\\3-W\\2-1", lemonade$Week))
lemonade$Quarter <- as.factor(quarter(lemonade$Date))

```

```
33 str(lemonade) #types of variables examined: correct.
34 summary(lemonade) #unitsales 741k!;128 NA's;
35
36 #missing values
37 sum(is.na(lemonade))
38 #check missed data in a specific attribute
39 summary.na<-lemonade %>% summarise_all(funs(sum(is.na
   (.))))
40 View(summary.na)
41 dim(summary.na)
42 #UnitSales, PricePU, PricePL, BasePricePU,
   BasePricePU - Each of the 4 variables have NAs.
43
44 #Visualize missing data
45 #percentage + histogram + pattern
46 aggr_plot <- aggr(lemonade, col=c('navyblue','red'),
47                   numbers=TRUE,
48                   sortVars=TRUE,
49                   labels=names(data),
50                   cex.axis=.7,
51                   gap=3,
52                   ylab=c("Histogram of Missing data",
   "Pattern"))
53
54 # 1.make graphs of variables over time ----
55 # extract week num from time data
56 lemonade <- lemonade %>%
57   separate(Week, c('removed', 'week_num', 'year'),
   ' ') %>%
58   mutate(cleaned_wk = trimws((week_num),which='left',
   whitespace='0'))
59 lemonade$cleaned_wk <- as.numeric(unlist(
   lemonade$cleaned_wk))
60
61 # in order to generate graphs over time, reassign
   week num according to year no. with if else
   statements
62 attach(lemonade)
63 lemonade$week_yrs <- ifelse(year=='17',seq(1,52),
64                             ifelse(year=='18',seq(53,
   104),
```

```
65                                          ifelse(year=='19'
   ,seq(105,156),
66                                                 ifelse(
   year=='20',seq(157,208),NA))))
67
68
69 # create variable 'sales'
70 lemonade$sales <- UnitSales*PricePU
71
72 # create variables 'discountPU_raw' (raw price
   difference)
73 # and 'discountPU_perc' (percent discount)
74 lemonade$discount_raw <- BasePricePU - PricePU
75 lemonade$discount_perc <- ((BasePricePU - PricePU
   ) / BasePricePU) * 100
76
77 #create ADJUSTED PROMOTION VARIABLES
78 str(which(lemonade$FeatDisp+lemonade$FeatOnly+
   lemonade$DispOnly > 100)) # just see how many rows
   there are (not used in function below)
79
80 #Create dummy which says 1 if the sum is more than
   100 (used in the variable creation ifstatement)
81 lemonade$sum100 <- ifelse(lemonade$FeatDisp+
   lemonade$FeatOnly+lemonade$DispOnly > 100, 1, 0)
82
83 #Creating the adjusted promotion colums
84 lemonade$FeatOnly_Adjusted <- ifelse(lemonade$sum100
    =='1',
85                                      ((
   lemonade$FeatOnly / (lemonade$FeatDisp+
   lemonade$FeatOnly+lemonade$DispOnly))*100),
   lemonade$FeatOnly)
86
87 lemonade$DispOnly_Adjusted <- ifelse(lemonade$sum100
    =='1',
88                                      ((
   lemonade$DispOnly / (lemonade$FeatDisp+
   lemonade$FeatOnly+lemonade$DispOnly))*100),
   lemonade$DispOnly)
89
```

```
lemonade$FeatDisp_Adjusted <- ifelse(lemonade$sum100
 =='1',
                                      ((
lemonade$FeatDisp / (lemonade$FeatDisp+
lemonade$FeatOnly+lemonade$DispOnly))*100),
lemonade$FeatDisp)


# do some cleaning up
lemonade <- lemonade[,-c(3,4)]
lemonade <- relocate(lemonade, week_yrs, .after =
Brand)
lemonade <- relocate(lemonade, sales, .after =
UnitSales)
lemonade <- relocate(lemonade, cleaned_wk, .after =
Brand)
lemonade <- relocate(lemonade, discount_raw, .after
 = BasePricePL)
lemonade <- relocate(lemonade, discount_perc, .after
 = discount_raw)
lemonade <- relocate(lemonade, sum100, .after =
FeatDisp_Adjusted)
lemonade <- relocate(lemonade, FeatOnly_Adjusted, .
after = FeatDisp)
lemonade <- relocate(lemonade, DispOnly_Adjusted, .
after = FeatOnly_Adjusted)
lemonade <- relocate(lemonade, FeatDisp_Adjusted, .
after = DispOnly_Adjusted)

#Gaining some general insights
#Total unit sales
plot(lemonade$UnitSales, xlab = "Observation", ylab
 = "Unit Sales", main = "Unit Sales",type = "o", pch
 = 20,col="dodgerblue")

#Identifying median and outliers
boxplot(lemonade$BasePricePU[lemonade$BasePricePU>0
]~lemonade$Brand[lemonade$BasePricePU>0],col="
firebrick1",ylab = "Prices (\u20AC)", main = "Prices
 per brand", xlab = NULL)

```

```
113 boxplot(lemonade$BasePricePU[lemonade$BasePricePU>0
    ]~lemonade$Chain[lemonade$BasePricePU>0],col=c("
    dodgerblue", "orange", "firebrick2", "dodgerblue", "
    gold","yellowgreen", "grey") ,ylab = "Prices (\u20AC
    )", xlab = NULL, main = "Prices per chain" )
114
115 BrandNames <- levels(unique(lemonade$Brand))
116 ChainNames <- levels(unique(lemonade$Chain))
117 ChainColors <- c("dodgerblue", "orange", "firebrick2
    ", "dodgerblue","gold","yellowgreen","grey")
118 names(ChainColors) <- c("Albert Heijn", "Coop", "
    Deen", "Hoogvliet", "Jumbo", "Plus", "
    TotalOnlineSales")
119
120 BoxPlotChainColors <- NULL
121 for (i in 1:length(ChainNames)) {
122   BoxPlotChainColors <- c(BoxPlotChainColors, rep(
    ChainColors[ChainNames[i]],length(BrandNames)))
123 }
124
125 op <- par(mar = c(8,4,4,2) + 0.1) ## default is c(5,
    4,4,2) + 0.1 Temporarily increase X-margin to allow
    for long brand names
126
127 boxplot(lemonade$BasePricePU[lemonade$BasePricePU>0
    ]~lemonade$Brand[lemonade$BasePricePU>0] +
    lemonade$Chain[lemonade$BasePricePU>0],las=2,col=
    BoxPlotChainColors, main="Prices per chain per brand
    ",ylab="Price (\u20AC)",names=rep(BrandNames,7),xlab
     = NULL)
128
129 StartText <- 4
130
131 for (i in 1:length(ChainNames)) {
132   text(StartText+(i-1)*6,3.2,ChainNames[i],col=
    ChainColors[ChainNames[i]])
133 }
134
135 op <- par(mar = c(5,4,4,2) + 0.1) ## set margins
    back to default, which is c(5,4,4,2) + 0.1
136
```

```
137 # make graphs of variables over time
138
139 # sales by channel; legend brand
140 lemonade %>%
141   filter(Chain=='Albert Heijn') %>%
142   ggplot(aes(x=Date, y=UnitSales)) +
143   geom_line(aes(colour=Brand)) +
144   ggtitle('Unit Sales per Brand Over Time\nChannel:
    Albert Heijn') +
145   xlab("Week") + ylab("Unit Sales") +
146   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
147
148 lemonade %>%
149   filter(Chain=='Coop') %>%
150   ggplot(aes(x=Date, y=UnitSales)) +
151   geom_line(aes(colour=Brand)) +
152   ggtitle('Unit Sales per Brand Over Time\nChannel:
    Coop') +
153   xlab("Week") + ylab("Unit Sales") +
154   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
155
156 lemonade %>%
157   filter(Chain=='Deen') %>%
158   ggplot(aes(x=Date, y=UnitSales)) +
159   geom_line(aes(colour=Brand)) +
160   ggtitle('Unit Sales per Brand Over Time\nChannel:
    Deen') +
161   xlab("Week") + ylab("Unit Sales") +
162   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
163
164 lemonade %>%
165   filter(Chain=='Hoogvliet') %>%
166   ggplot(aes(x=Date, y=UnitSales)) +
167   geom_line(aes(colour=Brand)) +
168   ggtitle('Unit Sales per Brand Over Time\nChannel:
    Hoogvliet') +
169   xlab("Week") + ylab("Unit Sales (\u20ac)") +
170   theme(plot.title = element_text(color="black",
```

```
size=14, face="bold", hjust = 0.5))

lemonade %>%
  filter(Chain=='Jumbo') %>%
  ggplot(aes(x=Date, y=UnitSales)) +
  geom_line(aes(colour=Brand)) +
  ggtitle('Unit Sales per Brand Over Time\nChannel: Jumbo') +
  xlab("Week") + ylab("Unit Sales") +
  theme(plot.title = element_text(color="black", size=14, face="bold", hjust = 0.5))

lemonade %>%
  filter(Chain=='Plus') %>%
  ggplot(aes(x=Date, y=UnitSales)) +
  geom_line(aes(colour=Brand)) +
  ggtitle('Unit Sales per Brand Over Time\nChannel: Plus') +
  xlab("Week") + ylab("Unit Sales") +
  theme(plot.title = element_text(color="black", size=14, face="bold", hjust = 0.5))

lemonade %>%
  filter(Chain=='TotalOnlineSales') %>%
  ggplot(aes(x=Date, y=UnitSales)) +
  geom_line(aes(colour=Brand)) +
  ggtitle('Unit Sales per Brand Over Time\nChannel: Online (total)') +
  xlab("Week") + ylab("Unit Sales") +
  theme(plot.title = element_text(color="black", size=14, face="bold", hjust = 0.5))

# sales by brand; legend channel
lemonade %>%
  filter(Brand=='EuroShopper') %>%
  ggplot(aes(x=Date, y=UnitSales)) +
  geom_line(aes(colour=Chain)) +
  ggtitle('Unit Sales per Channel Over Time\nBrand: EuroShopper') +
  xlab("Week") + ylab("Unit Sales") +
  theme(plot.title = element_text(color="black",
```

```
203 size=14, face="bold", hjust = 0.5))
204
205 lemonade %>%
206   filter(Brand=='KarvanCevitam') %>%
207   ggplot(aes(x=Date, y=UnitSales)) +
208   geom_line(aes(colour=Chain)) +
209   ggtitle('Sales per Channel Over Time\nBrand:
    Karavan Cevitam') +
210   xlab("Week") + ylab("Unit Sales") +
211   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
212
213 lemonade %>%
214   filter(Brand=='PrivateLabel') %>%
215   ggplot(aes(x=Date, y=UnitSales)) +
216   geom_line(aes(colour=Chain)) +
217   ggtitle('Unit Sales per Channel Over Time\nBrand:
    Private Labels') +
218   xlab("Week") + ylab("Unit Sales") +
219   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
220
221 lemonade %>%
222   filter(Brand=='Raak') %>%
223   ggplot(aes(x=Date, y=UnitSales)) +
224   geom_line(aes(colour=Chain)) +
225   ggtitle('Unit Sales per Channel Over Time\nBrand:
    Raak') +
226   xlab("Week") + ylab("Unit Sales") +
227   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
228
229 lemonade %>%
230   filter(Brand=='Slimpie') %>%
231   ggplot(aes(x=Date, y=UnitSales)) +
232   geom_line(aes(colour=Chain)) +
233   ggtitle('Unit Sales per Channel Over Time\nBrand:
    Slimpie') +
234   xlab("Week") + ylab("Unit Sales") +
235   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
```

```
236
237 lemonade %>%
238   filter(Brand=='Teisseire') %>%
239   ggplot(aes(x=Date, y=UnitSales)) +
240   geom_line(aes(colour=Chain)) +
241   ggtitle('Unit Sales per Channel Over Time\nBrand:
    Teisseire') +
242   xlab("Week") + ylab("Unit Sales") +
243   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
244
245
246 # discount (%) by channel; legend brand
247 lemonade %>%
248   filter(Chain=='Albert Heijn') %>%
249   ggplot(aes(x=Date, y=discount_perc)) +
250   geom_line(aes(colour=Brand)) +
251   ggtitle('Discount (%) per Brand Over Time\nChannel
    : Albert Heijn') +
252   xlab("Week") + ylab("Discount (%)") +
253   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
254
255 lemonade %>%
256   filter(Chain=='Coop') %>%
257   ggplot(aes(x=Date, y=discount_perc)) +
258   geom_line(aes(colour=Brand)) +
259   ggtitle('Discount (%) per Brand Over Time\nChannel
    : Coop') +
260   xlab("Week") + ylab("Discount (%)") +
261   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
262
263 lemonade %>%
264   filter(Chain=='Deen') %>%
265   ggplot(aes(x=Date, y=discount_perc)) +
266   geom_line(aes(colour=Brand)) +
267   ggtitle('Discount (%) per Brand Over Time\nChannel
    : Deen') +
268   xlab("Week") + ylab("Discount (%)") +
269   theme(plot.title = element_text(color="black",
```

```
size=14, face="bold", hjust = 0.5))

lemonade %>%
  filter(Chain=='Hoogvliet') %>%
  ggplot(aes(x=Date, y=discount_perc)) +
  geom_line(aes(colour=Brand)) +
  ggtitle('Discount (%) per Brand Over Time\nChannel
  : Hoogvliet') +
  xlab("Week") + ylab("Discount (%)") +
  theme(plot.title = element_text(color="black",
  size=14, face="bold", hjust = 0.5))

lemonade %>%
  filter(Chain=='Jumbo') %>%
  ggplot(aes(x=Date, y=discount_perc)) +
  geom_line(aes(colour=Brand)) +
  ggtitle('Discount (%) per Brand Over Time\nChannel
  : Jumbo') +
  xlab("Week") + ylab("Discount (%)") +
  theme(plot.title = element_text(color="black",
  size=14, face="bold", hjust = 0.5))

lemonade %>%
  filter(Chain=='Plus') %>%
  ggplot(aes(x=Date, y=discount_perc)) +
  geom_line(aes(colour=Brand)) +
  ggtitle('Discount (%) per Brand Over Time\nChannel
  : Plus') +
  xlab("Week") + ylab("Discount (%)") +
  theme(plot.title = element_text(color="black",
  size=14, face="bold", hjust = 0.5))

lemonade %>%
  filter(Chain=='TotalOnlineSales') %>%
  ggplot(aes(x=Date, y=discount_perc)) +
  geom_line(aes(colour=Brand)) +
  ggtitle('Discount (%) per Brand Over Time\nChannel
  : TotalOnlineSales') +
  xlab("Week") + ylab("Discount (%)") +
  theme(plot.title = element_text(color="black",
  size=14, face="bold", hjust = 0.5))
```

```
302
303
304
305 # 2.1 building brand position map----
306 seg.summ <- function(data , groups) {aggregate(data
     , list(groups), function(x) mean(as.numeric(x), na.
    rm = TRUE))}
307 lemonade_bp <- seg.summ(lemonade[, c("UnitSales", "
    PricePU")], lemonade$Brand)
308
309 ggplot(data=lemonade_bp,
310        aes(x = UnitSales, y = PricePU, label = Group
    .1)) +
311   geom_hline(yintercept = 0, colour = "gray70") +
312   geom_vline(xintercept = 0, colour = "gray70") +
313   geom_text(aes(colour=Group.1),size = 6) +
314   ggtitle("Brand Maps - UnitSales X Price") +
315   theme(legend.position = "none") +
316   theme(axis.title = element_text(size=20))+
317   theme(plot.title = element_text(size=20))+
318   xlim(-50000,100000) #including negative value only
     to show the full label name in the chart for
    aesthetics purpose!
319
320 # 2.2 investigate frequency of promotion across
    different brands and supermarket formulas?----
321 # create dummies for each kind of promotion
322
323 lemonade$FeatOnlyD <- as.integer(ifelse(
    lemonade$FeatOnly>0,1,0))
324 lemonade$DispOnlyD <- as.integer(ifelse(
    lemonade$DispOnly>0,1,0))
325 lemonade$FeatDispD <- as.integer(ifelse(
    lemonade$FeatDisp>0,1,0))
326
327 # frequency of promotion across different brands
328 seg.summ <- function(data , groups) {aggregate(data
     , list(groups), function(x) sum(data.frame(x)))}
329 lemonade_promotion_fq <- seg.summ(lemonade[, c("
    FeatOnlyD", "DispOnlyD", "FeatDispD")],
    lemonade$Brand)
```

```
colnames(lemonade_promotion_fq)[1] <- 'Brand'
lemonade_promotion_fq <- melt(lemonade_promotion_fq
, id = c("Brand"))
ggplot(data=lemonade_promotion_fq,
       aes(x=reorder(lemonade_promotion_fq$variable,
lemonade_promotion_fq$value,decreasing = TRUE),
           y=lemonade_promotion_fq$value, fill=
lemonade_promotion_fq$Brand)) +
  ggtitle('Promotion Frequency per Brand') +
  xlab("Promotion Type") + ylab("Total Promotions"
) + labs(fill = "Brand") +
  geom_col(position = position_dodge())

# frequency of promotion across different
supermarket formulas
seg.summ <- function(data , groups) {aggregate(data
 , list(groups), function(x) sum(data.frame(x)))}
lemonade_promotion_fq <- seg.summ(lemonade[, c("
FeatOnlyD", "DispOnlyD", "FeatDispD")],
lemonade$Chain)
colnames(lemonade_promotion_fq)[1] <- 'Chain'
lemonade_promotion_fq <- melt(lemonade_promotion_fq
, id = c("Chain"))
ggplot(data=lemonade_promotion_fq,
       aes(x=reorder(lemonade_promotion_fq$variable,
lemonade_promotion_fq$value,decreasing = TRUE),
           y=lemonade_promotion_fq$value, fill=
lemonade_promotion_fq$Chain)) +
  ggtitle('Promotion Frequency per Channel') +
  xlab("Promotion Type") + ylab("Total Promotions"
) + labs(fill = "Channel") +
  geom_col(position = position_dodge())

# depth of promotion across different brands
seg.summ <- function(data , groups) {aggregate(data
 , list(groups), function(x) mean(x,na.rm=TRUE))}
lemonade_promotion_depth <- seg.summ(lemonade[, c("
discount_perc")], lemonade$Brand)
colnames(lemonade_promotion_depth)[1] <- 'Brand'
ggplot(data=lemonade_promotion_depth,
       aes(x=reorder(lemonade_promotion_depth$Brand,
```

```
356 lemonade_promotion_depth$x,decreasing = TRUE),
357             y=lemonade_promotion_depth$x, fill=
    lemonade_promotion_depth$Brand)) +
358   ggtitle('Promotion Depth per Brand') +
359   xlab("Brand") + ylab("Aggregated mean value of
    discount in %") + labs(fill = "Brand") +
360   geom_col(position = position_dodge())
361
362 # depth of promotion across different supermarket
    formulas
363 seg.summ <- function(data , groups) {aggregate(data
     , list(groups), function(x) mean(x,na.rm=TRUE))}
364 lemonade_promotion_depth <- seg.summ(lemonade[, c("
    discount_perc")], lemonade$Chain)
365 colnames(lemonade_promotion_depth)[1] <- 'Chain'
366 ggplot(data=lemonade_promotion_depth,
367        aes(x=reorder(lemonade_promotion_depth$Chain,
    lemonade_promotion_depth$x,decreasing = TRUE),
368             y=lemonade_promotion_depth$x, fill=
    lemonade_promotion_depth$Chain)) +
369   ggtitle('Promotion Depth per Chain') +
370   xlab("Brand") + ylab("Aggregated mean value of
    discount in %") + labs(fill = "Chain") +
371   geom_col(position = position_dodge())
372
373 # 3.(seasonal influences) ----
374 anova_quarter <- aov(UnitSales ~ Quarter, data =
    lemonade)
375 summary(anova_quarter)
376
377 # visualization for the seasonal influence
378 lemonadeRaak <- lemonade[lemonade$Brand == "Raak",]
    # make dataframe with only Raak at all chains
379 boxplot((lemonadeRaak$UnitSales[
    lemonadeRaak$UnitSales>0])/1000 ~
    lemonadeRaak$Quarter[lemonadeRaak$UnitSales>0] +
    lemonadeRaak$Chain[lemonadeRaak$UnitSales>0],las=2,
    col="#0DBDC2", main="Raak UnitSales per chain per
    quarter",ylab="Unit sales (x 1000)",xlab = NULL)
380
381 # 4.simple linear regression to see the effect of
```

```
price on sales----

ModelAllBrandsAllChains <- lm(UnitSales ~ PricePU,
data = lemonade)
summary(ModelAllBrandsAllChains)
by(lemonade,lemonade[,c("Chain", "Brand")],function(
x) summary(lm(UnitSales~PricePU,data = x)))

# Effect of Price (per brand)
by(lemonade,lemonade[,c("Brand")],function(x)
summary(lm(UnitSales~PricePU,data = x)))
# Effect of Discount (per brand)
by(lemonade,lemonade[,c("Brand")],function(x)
summary(lm(UnitSales~discount_perc,data = x)))
# Effect of Feature Promotion (per brand)
by(lemonade,lemonade[,c("Brand")],function(x)
summary(lm(UnitSales~FeatOnly_Adjusted,data = x)))
# Effect of Display Promotion (per brand)
by(lemonade,lemonade[,c("Brand")],function(x)
summary(lm(UnitSales~DispOnly_Adjusted,data = x)))
# Effect of F&D Promotion (per brand)
by(lemonade,lemonade[,c("Brand")],function(x)
summary(lm(UnitSales~FeatDisp_Adjusted,data = x)))

# 5.more graph analysis on the selected brand Raak
----
# help me to inspect data more easily...
# write.csv(lemonade,file="lemonade.csv")

lemonade %>%
  filter(Brand=='Raak') %>%
  ggplot(aes(x=Date, y=UnitSales/1000)) +
  geom_line(aes(colour=Chain)) +
  ggtitle('Unit Sales per Channel Over Time\nBrand:
Raak') +
  xlab("Week") + ylab("Unit Sales x 1000") +
  theme(plot.title = element_text(color="black",
size=14, face="bold", hjust = 0.5))

lemonade %>%
  filter(Brand=='Raak') %>%
```

```
412   ggplot(aes(x=Date, y=sales/1000)) +
413   geom_line(aes(colour=Chain)) +
414   ggtitle('Sales per Channel Over Time\nBrand: Raak'
    ) +
415   xlab("Week") + ylab("Sales x 1000") +
416   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
417
418 lemonade %>%
419   filter(Brand=='Raak') %>%
420   ggplot(aes(x=Date, y=PricePU)) +
421   geom_line(aes(colour=Chain)) +
422   ggtitle('PricePU per Channel Over Time\nBrand:
    Raak') +
423   xlab("Week") + ylab("PricePU") +
424   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
425
426 lemonade %>%
427   filter(Brand=='Raak') %>%
428   ggplot(aes(x=Date, y=BasePricePU)) +
429   geom_line(aes(colour=Chain)) +
430   ggtitle('BasePricePU per Channel Over Time\nBrand
    : Raak') +
431   xlab("Week") + ylab("BasePricePU") +
432   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
433
434 lemonade %>%
435   filter(Brand=='Raak'& Chain=="Albert Heijn") %>%
436   ggplot(aes(x=Date, y=discount_perc)) +
437   geom_line(aes(colour=Chain)) +
438   ggtitle('Discount Percentage per Channel Over Time
    \nBrand: Raak') +
439   xlab("Week") + ylab("discount_perc") +
440   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
441
442 # Relational Plots
443 glimpse(lemonade[,c("UnitSales","PricePU","
    FeatOnly_Adjusted","DispOnly_Adjusted","
```

```
443 FeatDisp_Adjusted","Quarter","discount_perc")])
444 # Correlation Matrix between key variables
445 lemonade.cor <- cor(lemonade[lemonade$Brand == "Raak
    ",][,c("sales","PricePU","FeatOnly_Adjusted","
    DispOnly_Adjusted","FeatDisp_Adjusted","
    discount_perc")])
446 ggcorrplot(lemonade.cor,
447            hc.order = TRUE,
448            type = "lower",
449            lab = TRUE,
450            colors =  brewer.pal(3,"RdBu"))
451
452 # Regression plots
453 #Feature
454 ggplot(subset(lemonade, Brand == "Raak"), aes(x =
    FeatOnly_Adjusted, y= sales ))+
455   geom_point()+
456   stat_smooth(method=lm) +
457   ggtitle('Feature Promotion effect on Sales \n
    Brand: Raak') +
458   xlab("Feature Promotion") + ylab("Sales") +
459   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
460 #Display
461 ggplot(subset(lemonade, Brand == "Raak"), aes(x =
    DispOnly_Adjusted, y= sales ))+
462   geom_point()+
463   stat_smooth(method=lm) +
464   ggtitle('Display Promotion effect on Sales \n
    Brand: Raak') +
465   xlab("Display Promotion") + ylab("Sales") +
466   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
467 #Feature & Display
468 ggplot(subset(lemonade, Brand == "Raak"), aes(x =
    FeatDisp_Adjusted, y= sales ))+
469   geom_point()+
470   stat_smooth(method=lm) +
471   ggtitle('Feature & Display Promotion effect on
    Sales \n Brand: Raak') +
472   xlab("F&D Promotion") + ylab("Sales") +
```

```
473   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
474 #Discount
475 ggplot(subset(lemonade, Brand == "Raak"), aes(x =
    discount_perc, y= sales ))+
476   geom_point()+
477   stat_smooth(method=lm) +
478   ggtitle('Discount effect on Sales \n Brand: Raak'
    ) +
479   xlab("Discount") + ylab("Sales") +
480   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
481 #Price (per unit)
482 ggplot(subset(lemonade, Brand == "Raak"), aes(x =
    PricePU, y= sales ))+
483   geom_point()+
484   stat_smooth(method=lm) +
485   ggtitle('Price (per unit) effect on Sales \n Brand
    : Raak') +
486   xlab("Price") + ylab("Sales") +
487   theme(plot.title = element_text(color="black",
    size=14, face="bold", hjust = 0.5))
488
489 # Dominant promotion influence
490 # Create Promotion dominance groups
491 # : 1-Feature dominant, 2-Display dominant, 3-
    Combination dominant (if there is no clear dominance
     then NA, so that it's excluded from the analysis)
492 lemonade$PromoDominance <- ifelse(
    lemonade$FeatOnly_Adjusted >
    lemonade$DispOnly_Adjusted &
    lemonade$FeatOnly_Adjusted >
    lemonade$FeatDisp_Adjusted,1,
493                                    ifelse(
    lemonade$DispOnly_Adjusted >
    lemonade$FeatOnly_Adjusted &
    lemonade$DispOnly_Adjusted >
    lemonade$FeatDisp_Adjusted,2,
494                                           ifelse(
    lemonade$FeatDisp_Adjusted >
    lemonade$FeatOnly_Adjusted &
```

```
494 lemonade$FeatDisp_Adjusted >
    lemonade$DispOnly_Adjusted,3,NA)))
495 anova_promo <- aov(UnitSales ~ as.factor(
    PromoDominance), data = lemonade)
496 summary(anova_promo)
497 TukeyHSD(anova_promo)
498
499
500 # model corrected - by Elias Date: 01/06/2022----
501 Lemonade <- lemonade
502
503 # remove unnecessary stuff from the environment
504 rm(aggr_plot,anova_promo,anova_quarter,le_endo,
    le_exo,lemonade_bp,lemonade_est,
    lemonade_promotion_depth,lemonade_promotion_fq,
    lemonade.cor,lemonadeRaak,ModelAllBrandsAllChains,op
    ,summary.na)
505 rm(BoxPlotChainColors,ChainColors,i,StartText,seg.
    summ)
506 rm(lemonade)
507
508 # create smooth version of BasePricePU
509 #Check for inconsistiencies:
510
511 Lemonade$Promotion <- Lemonade$BasePricePU -
    Lemonade$PricePU
512 percentagefalse <- (sum(Lemonade$Promotion < 0, na.
    rm=TRUE)/nrow(Lemonade))*100
513
514 Lemonade$BasePricePU_Smooth <- rep(0,nrow(Lemonade
    )) #create a new variable in the data frame - a
    smoothed version of BasePricePU - initially filled
    up with zeros
515
516 #Loop over chains and brands: within this loop,
    loess regression is used to create a smoothed
    version of BasePricePU
517
518 for(ch in ChainNames){
519   for(br in BrandNames){
520     loess_temp <- loess(Lemonade$BasePricePU[
```

```
520 Lemonade$Chain==ch & Lemonade$Brand==br]~c(1:length(
    Lemonade$BasePricePU[Lemonade$Chain==ch &
    Lemonade$Brand==br])),span=0.10)
521     if (length(Lemonade$BasePricePU_Smooth[
    Lemonade$Chain==ch & Lemonade$Brand==br]) != length(
    loess_temp$fitted)) {
522       fill_up <- c(loess_temp$fitted,rep(NA,length(
    Lemonade$BasePricePU_Smooth[Lemonade$Chain==ch &
    Lemonade$Brand==br])-length(loess_temp$fitted)))
523       Lemonade$BasePricePU_Smooth[Lemonade$Chain==ch
     & Lemonade$Brand==br] <- fill_up
524     } else {
525       Lemonade$BasePricePU_Smooth[Lemonade$Chain==ch
     & Lemonade$Brand==br] <- loess_temp$fitted
526     }
527   }
528 }
529
530 #Now that smoothing is done, replace any remaining
    smoothed BasePricePU values by PricePU values if
    they are lower than PricePU
531 Lemonade$BasePricePU_Smooth <- ifelse(
    Lemonade$BasePricePU_Smooth > Lemonade$PricePU,
    Lemonade$BasePricePU_Smooth, Lemonade$PricePU)
532
533 #Now check again if there are inconsistencies
534 Lemonade$Promotion <- Lemonade$BasePricePU_Smooth -
    Lemonade$PricePU
535 percentagefalse <- (sum(Lemonade$Promotion < 0, na.
    rm=TRUE)/nrow(Lemonade))*100 # now no
    inconsistencies anymore
536
537 #Just a visual check how it looks like for one brand
    -chain combination:
538 plot(Lemonade$PricePU[Lemonade$Chain=="Jumbo" &
    Lemonade$Brand=="Raak"],type="l")
539 lines(Lemonade$BasePricePU_Smooth[Lemonade$Chain=="
    Jumbo" & Lemonade$Brand=="Raak"],type="l",col="red")
540
541 # combine covid data
542 # Download COVID data from the web ----
```

```r
Covid_cases <- read.csv("https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv")

# Data cleaning ----

## Only focus on data from the Netherlands ----
Covid_cases_Reduced <- Covid_cases[Covid_cases$Country.Region == "Netherlands",]

## Only focus on data from the main country ----
Covid_cases_Reduced_Further <- Covid_cases_Reduced[Covid_cases_Reduced$Province.State == "",]

## Transpose the dataframe and exclude the first four entries ----
Covid_cases_df <- data.frame(t(Covid_cases_Reduced_Further[,-c(1:4)]))

## Extract a date variable from the row names ----
Covid_cases_df$Date <- as.Date(row.names(Covid_cases_df),"X%m.%d.%y")

## Rename focal variable ----
names(Covid_cases_df)[names(Covid_cases_df)=="X201"] <- "CovidCases"

## Make a plot of the focal variable ----
plot(Covid_cases_df$Date,Covid_cases_df$CovidCases,type="l")

## Create a new variable, consisting of new cases ----
Covid_cases_df$NewCovidCases <- c(NA,diff(Covid_cases_df$CovidCases,1))

## Make a plot of the new variable ----
plot(Covid_cases_df$Date,Covid_cases_df$NewCovidCases,type="l")

```

```
571 ## Replace one very extreme outlier ----
572 Covid_cases_df$NewCovidCases[
    Covid_cases_df$NewCovidCases > 150000] <- max(
    Covid_cases_df$NewCovidCases[
    Covid_cases_df$NewCovidCases < 150000],na.rm = TRUE)
573
574 ## Make a plot of the new variable again - does this
     look better? ----
575 plot(Covid_cases_df$Date,
    Covid_cases_df$NewCovidCases,type="l")
576
577 ## Add a week variable ----
578 library(lubridate)
579 Covid_cases_df$Week <- week(Covid_cases_df$Date)
580 Covid_cases_df$Year <- year(Covid_cases_df$Date)
581 ## Calculate weekly averages ----
582 Years <- unique(Covid_cases_df$Year)
583 Weeks <- unique(Covid_cases_df$Week)
584 for (iYear in Years) {
585   for (iWeek in Weeks) {
586     Covid_cases_df$WeekAverage[Covid_cases_df$Year
     == iYear & Covid_cases_df$Week == iWeek] <- mean(
    Covid_cases_df$NewCovidCases[Covid_cases_df$Year ==
    iYear & Covid_cases_df$Week == iWeek],na.rm = TRUE)
587   }
588 }
589
590 ## Make a plot to check whether we created the
    weekly average correctly
591 plot(Covid_cases_df$Date,
    Covid_cases_df$NewCovidCases,type="l")
592 lines(Covid_cases_df$Date,Covid_cases_df$WeekAverage
    ,type="l", col="Red")
593
594 # Clean up after getting external data ----
595 rm(Covid_cases,Covid_cases_Reduced,
    Covid_cases_Reduced_Further,iWeek,Weeks)
596
597 # Combine lemonade data and covid data ----
598 Covid_cases_df <- subset(Covid_cases_df,select = c("
    Week","Year","WeekAverage"))
```

```
599 Covid_cases_df <- Covid_cases_df %>% distinct(Year,
    Week, .keep_all = TRUE)
600 Lemonade$year <- year(Lemonade$Date)
601 Lemonade_extended <- data.frame()
602 for(ch in ChainNames){
603   for(br in BrandNames){
604     Lemonade_ch_br <- merge(subset(Lemonade[
    Lemonade$Chain==ch & Lemonade$Brand==br, ]),
    Covid_cases_df,by.x = c("year","cleaned_wk"),by.y =
    c("Year","Week"),all.x = TRUE)
605     Lemonade_extended <- rbind(Lemonade_extended,
    Lemonade_ch_br)
606   }
607 }
608
609 # combine extended data and weather data----
610 WeatherDF <- read.csv("etmgeg_280.csv", header =
    TRUE)
611 WeatherDF$Date <- as.Date(as.character(
    WeatherDF$YYYYMMDD),"%Y%m%d")
612 WeatherDF$Week <- ISOweek(WeatherDF$Date)
613 TempAvg <- aggregate(TG/10~Week, FUN=mean, data=
    WeatherDF, na.rm=TRUE)
614
615 UniqueDates <- unique(Lemonade_extended$Date)
616
617 Lemonade_extended$Temp <- rep(0,nrow(
    Lemonade_extended))
618
619 for (i in 1:length(UniqueDates)) {
620   Lemonade_extended$Temp[Lemonade_extended$Date==
    UniqueDates[i]] <- TempAvg[ISOweek2date(paste(
    TempAvg$Week,"1",sep="-")) == UniqueDates[i],"TG/10"
    ]
621 }
622
623 rm(WeatherDF)
624
625 # include dynamic effect - partial adjustment
626 # first, select subset of Raak
627 LemonadeRaak <- subset(Lemonade_extended[
```

```
Lemonade_extended$Brand == "Raak", ])
# create variable lag of unitsales
LemonadeRaak$UnitSalesLag <- c(NA,
LemonadeRaak$UnitSales[1:nrow(LemonadeRaak)-1])

LemonadeRaak$UnitSalesLag[LemonadeRaak$year == "2017
" & LemonadeRaak$cleaned_wk == "1"] <- NA

# add competitors price

LemonadeRaak$PricePUPL <- rep(0,nrow(LemonadeRaak))
LemonadeRaak$PricePUSlimpie <- rep(0,nrow(
LemonadeRaak))

Dates <- unique(LemonadeRaak$Date)

Chains <- unique(LemonadeRaak$Chain)

for (i in Dates) {
  for (j in Chains) {
    LemonadeRaak$PricePUPL[LemonadeRaak$Chain == j
 & LemonadeRaak$Date == i] <- Lemonade$PricePU[
Lemonade$Brand == "PrivateLabel" & Lemonade$Chain
 == j & Lemonade$Date == i]
    LemonadeRaak$PricePUSlimpie[LemonadeRaak$Chain
 == j & LemonadeRaak$Date == i] <- Lemonade$PricePU[
Lemonade$Brand == "Slimpie" & Lemonade$Chain == j &
Lemonade$Date == i]
  }
}

# create price index = price per unit/base price
LemonadeRaak$PriceIndex <- LemonadeRaak$PricePU/
LemonadeRaak$BasePricePU_Smooth

#Building model----
#replace na with 0 in week average covid cases to
keep degree of freedom
LemonadeRaak$WeekAverage[is.na(
LemonadeRaak$WeekAverage)] <- 0
#unit by unit; unpooled
```

```
656 R2s <- data.frame(Chains) #Make the dataframe
657
658 for (i in Chains) { #loop over chains
659   MultiplicativeTemp <- lm(log(UnitSales) ~ log(
    BasePricePU_Smooth)+log(PriceIndex)+log(
    PricePUSlimpie)+log(PricePUPL)
660                            + FeatOnly_Adjusted +
    DispOnly_Adjusted + FeatDisp_Adjusted + log(
    UnitSalesLag) + log(Temp+273) + WeekAverage,
661                            data = LemonadeRaak[
    LemonadeRaak$Chain==i,])
662   #This object changes for every value of i, and is
    not accessible outside of the loop
663   message("Output for ",i,":",sep="") #Writes this
    sentence in red to the console window
664   print(summary(MultiplicativeTemp)) #Prints the
    output to the screen
665   R2s$R2[which(Chains==i)] <- summary(
    MultiplicativeTemp)$r.squared
666   print(summary(aov(MultiplicativeTemp)))
667   #Stores the R2-value in the right row of the R2
    variable in the data frame R2s
668 }
669 #total residual=4.374+4.665+9.1+6.193+1.842+9.051+8.
    729 !!!
670
671 print(R2s) #the R2 values are now accessible outside
     of the loop
672
673 #Pooled version of the model:
674 PooledModel <- lm(log(UnitSales) ~ log(
    BasePricePU_Smooth)+log(PriceIndex)+log(
    PricePUSlimpie)+log(PricePUPL)
675                    + FeatOnly_Adjusted +
    DispOnly_Adjusted + FeatDisp_Adjusted + log(
    UnitSalesLag) + log(Temp+273) + WeekAverage,
676                    data = LemonadeRaak)
677 summary(PooledModel)
678 summary(aov(PooledModel))
679
680 #Partially pooled version of the model:
```

```
#First create dummies for the chains:
LemonadeRaak$D_AH    <- rep(0,nrow(LemonadeRaak))
LemonadeRaak$D_Jumbo <- rep(0,nrow(LemonadeRaak))
LemonadeRaak$D_Plus  <- rep(0,nrow(LemonadeRaak))
LemonadeRaak$D_Coop  <- rep(0,nrow(LemonadeRaak))
LemonadeRaak$D_Deen  <- rep(0,nrow(LemonadeRaak))
LemonadeRaak$D_HV    <- rep(0,nrow(LemonadeRaak))
LemonadeRaak$D_TOS   <- rep(0,nrow(LemonadeRaak))

LemonadeRaak$D_AH[LemonadeRaak$Chain=="Albert Heijn"
]      <- 1
LemonadeRaak$D_Jumbo[LemonadeRaak$Chain=="Jumbo"
]          <- 1
LemonadeRaak$D_Plus[LemonadeRaak$Chain=="Plus"
]            <- 1
LemonadeRaak$D_Coop[LemonadeRaak$Chain=="Coop"
]            <- 1
LemonadeRaak$D_Deen[LemonadeRaak$Chain=="Deen"
]            <- 1
LemonadeRaak$D_HV[LemonadeRaak$Chain=="Hoogvliet"
]         <- 1
LemonadeRaak$D_TOS[LemonadeRaak$Chain=="
TotalOnlineSales"] <- 1

#Option 1: estimate model without intercept
PartiallyPooledModelTemp1 <- lm(log(UnitSales) ~ -1
 + D_AH + D_Jumbo + D_Plus + D_Coop + D_Deen + D_HV
 + D_TOS +
                                    log(
BasePricePU_Smooth)+log(PriceIndex)+log(
PricePUSlimpie)+log(PricePUPL)
                                  + FeatOnly_Adjusted
 + DispOnly_Adjusted + FeatDisp_Adjusted + log(
UnitSalesLag) + log(Temp+273) + WeekAverage,
                                  data = LemonadeRaak)
summary(PartiallyPooledModelTemp1)

#Option 2: estimate model with intercept
PartiallyPooledModelTemp2 <- lm(log(UnitSales) ~
D_AH + D_Jumbo + D_Plus + D_Coop + D_Deen + D_HV +
D_TOS +
```

```
707                                          log(
    BasePricePU_Smooth)+log(PriceIndex)+log(
    PricePUSlimpie)+log(PricePUPL)
708                                       + FeatOnly_Adjusted
     + DispOnly_Adjusted + FeatDisp_Adjusted + log(
    UnitSalesLag) + log(Temp+273) + WeekAverage,
709                                       data = LemonadeRaak)
710 summary(PartiallyPooledModelTemp2)
711 summary(aov(PartiallyPooledModelTemp2))
712
713 partially_pooled_sum_sq <- summary(aov(
    PartiallyPooledModelTemp2))[1][[1]][[2]][[17]] #69.8
714 pooled_sum_sq <- summary(aov(PooledModel))[1][[1]][[
    2]][[11]] #170.1
715 unit_by_unit_sum_sq <- 4.374+4.665+9.1+6.193+1.842+9
    .051+8.729 #43.954
716
717 # Please use a part of the data for estimation, and
    save a part for validation
718
719 LemonadeRaak_Calibrate <- LemonadeRaak[
    LemonadeRaak$Date < "2020-08-01",]
720 LemonadeRaak_Validate <- LemonadeRaak[
    LemonadeRaak$Date >= "2020-08-01",]
721
722 length(unique(LemonadeRaak_Calibrate$Date))
723 length(unique(LemonadeRaak_Validate$Date))
724
725 #performing the chow test----
726 #degree of freedom
727 #df_pooled = 7*207-11=1438
728 #df_unpooled = 7*(207-11)=1372 <---------- added up
    instead: 1378 = 196(AH)+196(Coop)+198(Deen)+196(
    Hoogvliet)+198(Jumbo)+196(Plus)+198(Online)
729 #df_partially_pooled = 7*207-17=1432
730 F_UnitbyUnit_P<- (
731   (pooled_sum_sq-unit_by_unit_sum_sq)/(1438-1372))/(
732     unit_by_unit_sum_sq/1372)
733 F_UnitbyUnit_P
734 #when F(66,1372), p-value = 0, fully pooled is not
    allowed
```

```
F_partiallyP_P <- (
  (partially_pooled_sum_sq-unit_by_unit_sum_sq)/(1432-1372))/(
    unit_by_unit_sum_sq/1372)
F_partiallyP_P
#when F(60,1372), p-value = 0 partially pooled is not allowed
#Therefore, we choose not to pool!

#performing the chow test (UPDATED - pls check it out just in case and delete this text) ----
#degree of freedom
#df_pooled = 7*207-11=1438
#df_unpooled = 7*(207-11)=1372 <---------- added up instead: 1378 = 196(AH)+196(Coop)+198(Deen)+196(Hoogvliet)+198(Jumbo)+196(Plus)+198(Online)
#df_partially_pooled = 7*207-17=1432
F_UnitbyUnit_P_update<- (
  (pooled_sum_sq-unit_by_unit_sum_sq)/(1438-1378))/(
    unit_by_unit_sum_sq/1378)
F_UnitbyUnit_P_update # = 65.89476
#when F(60,1378), p-value = 0, fully pooled is not allowed
F_partiallyP_P_update <- (
  (partially_pooled_sum_sq-unit_by_unit_sum_sq)/(1432-1378))/(
    unit_by_unit_sum_sq/1378)
F_partiallyP_P_update # = 14.99258
#when F(54,1378), p-value = 0 partially pooled is not allowed
#Therefore, we choose not to pool!

# vif test----
MultiplicativeAH <- lm(log(UnitSales) ~ log(BasePricePU_Smooth)+log(PriceIndex)+log(PricePUSlimpie)+log(PricePUPL)
                       + FeatOnly_Adjusted + DispOnly_Adjusted + FeatDisp_Adjusted + log(UnitSalesLag) + log(Temp+273) + WeekAverage,
                       data = LemonadeRaak_Calibrate[LemonadeRaak_Calibrate$Chain=="Albert Heijn",])
```

```
763 vif(MultiplicativeAH)
764 summary(MultiplicativeAH)
765 # log(BasePricePU_Smooth)          log(PriceIndex
    )     log(PricePUSlimpie)
766 #                1.215898                 5.087799
                   1.130562
767 #           log(PricePUPL)       FeatOnly_Adjusted
          DispOnly_Adjusted
768 #                1.116053                17.354348
                   1.235906
769 #        FeatDisp_Adjusted       log(UnitSalesLag
    )         log(Temp + 273)
770 #                11.547643                1.071903
                   1.095113
771 #              WeekAverage
772 #                1.154179
773
774 MultiplicativePlus <- lm(log(UnitSales) ~ log(
    BasePricePU_Smooth)+log(PriceIndex)+log(
    PricePUSlimpie)+log(PricePUPL)
775                             + FeatOnly_Adjusted +
    DispOnly_Adjusted + FeatDisp_Adjusted + log(
    UnitSalesLag) + log(Temp+273) + WeekAverage,
776                             data =
    LemonadeRaak_Calibrate[LemonadeRaak_Calibrate$Chain
    =="Plus",])
777 vif(MultiplicativePlus)
778 # log(BasePricePU_Smooth)          log(PriceIndex
    )     log(PricePUSlimpie)
779 #                1.341361                 4.061166
                   1.065802
780 #           log(PricePUPL)       FeatOnly_Adjusted
          DispOnly_Adjusted
781 #                1.285438                 7.901394
                   1.671563
782 #        FeatDisp_Adjusted       log(UnitSalesLag
    )         log(Temp + 273)
783 #                9.516049                 1.691277
                   1.153694
784 #              WeekAverage
785 #                1.192421
```

```
786
787 MultiplicativeCoop <- lm(log(UnitSales) ~ log(
    BasePricePU_Smooth)+log(PriceIndex)+log(
    PricePUSlimpie)+log(PricePUPL)
788                           + FeatOnly_Adjusted +
    DispOnly_Adjusted + FeatDisp_Adjusted + log(
    UnitSalesLag) + log(Temp+273) + WeekAverage,
789                           data =
    LemonadeRaak_Calibrate[LemonadeRaak_Calibrate$Chain
    =="Coop",])
790 vif(MultiplicativeCoop)
791 # log(BasePricePU_Smooth)         log(PriceIndex
    )     log(PricePUSlimpie)
792 #                1.261248                10.474991
                   1.024517
793 #          log(PricePUPL)       FeatOnly_Adjusted
          DispOnly_Adjusted
794 #                1.342303                 8.520145
                   1.207626
795 #       FeatDisp_Adjusted       log(UnitSalesLag
    )         log(Temp + 273)
796 #                1.873290                 1.409149
                   1.466902
797 #             WeekAverage
798 #                1.220419
799
800 MultiplicativeHoogvliet <- lm(log(UnitSales) ~ log(
    BasePricePU_Smooth)+log(PriceIndex)+log(
    PricePUSlimpie)+log(PricePUPL)
801                                 + FeatOnly_Adjusted +
    DispOnly_Adjusted + FeatDisp_Adjusted + log(
    UnitSalesLag) + log(Temp+273) + WeekAverage,
802                                 data =
    LemonadeRaak_Calibrate[LemonadeRaak_Calibrate$Chain
    =="Hoogvliet",])
803 vif(MultiplicativeHoogvliet)
804 summary(MultiplicativeHoogvliet)
805 # log(BasePricePU_Smooth)         log(PriceIndex
    )     log(PricePUSlimpie)
806 #                1.691111                 5.503598
                   1.293660
```

```
807 #          log(PricePUPL)       FeatOnly_Adjusted
        DispOnly_Adjusted
808 #                1.402924                1.992625
                1.173843
809 #       FeatDisp_Adjusted       log(UnitSalesLag
    )         log(Temp + 273)
810 #                4.401795                1.089811
                1.164649
811 #             WeekAverage
812 #                1.201136
813
814
815
816 # other 3 chains (excluded Jumbo, Deen and TOS,
    since they are missing data wrt F, D and FD) got
    multi-collinearity issue as well...
817
818 # CONCLUSION: We will solve the multicollinearity
    for the Hoogvliet chain and continue with it because
     it has the best VIF values overall.
819
820 # solve multi-collinearity: Hoogvliet ----
821
822 # We recode the Feat/Disp/FeatDisp variables and
    Price Index into new combined variables
823 # which signify the presence of both price and
    advertising promotion based on index
824
825 LemonadeRaak_Calibrate$PriceIndex # we already have
    our price index variable
826
827 #CUT-OFF LOGIC: to determine cut off value we look
    at boxplots and histograms
828 # BOXPLOT (F, D and F&D)
829 # Conclusion: if we look at where the majority of
    outliers start:
830 # for F it is around 8, for D around 2, for F&D
    around 41 => the cut off value we will use is 17
831 boxplot(LemonadeRaak_Calibrate[c("FeatOnly_Adjusted"
    , "DispOnly_Adjusted", "FeatDisp_Adjusted")],
832         main = "Boxplots of 3 Promotion types",
```

```
833         at = c(1,2,3),
834         names = c("F Only", "D Only", "F&D"),
835         las = 1,
836         col = c("red","blue", "green"),
837         border = c("red","blue", "green"),
838         horizontal = FALSE,
839         notch = TRUE
840 )
841
842 # BOXPLOT (PriceIndex)
843 # Conclusion: we choose 0.766
844 hist(LemonadeRaak_Calibrate[c("PriceIndex")],
845      main="Feature Only Promotion Histogram",
846      col="darkmagenta",
847      freq=TRUE
848 )
849
850 #create cutoff variables for easy change if needed
851 promocutoff1 <- 17
852 pricecutoff1 <- 0.766
853
854 #recode price and promotion variables
855 # variable meaning:
856 # pf1: price index if there is feature only support
     (>17), otherwise 1
857 # pd1: price index if there is display only support
     (>17), otherwise 1
858 # pfd1: price index if there is feature & display
    support (>17), otherwise 1
859 # pwo1: price index if display only, feature only
    AND f&d support is (<=17), otherwise 1
860 # fwo1: feature only support, but no price cut (> 0.
    761), otherwise 0
861 # dwo1: display only support, but no price cut (> 0.
    761), otherwise 0
862 # fdwo1: f&d support, but no price cut (> 0.761),
    otherwise 0
863
864 #1: create and populate variables (which we replace
    in step 2 based on the cutoff)
865 LemonadeRaak_Calibrate$pf1 <- rep(1,nrow(
```

```
865 LemonadeRaak_Calibrate))
866 LemonadeRaak_Calibrate$pd1 <- rep(1,nrow(
    LemonadeRaak_Calibrate))
867 LemonadeRaak_Calibrate$pfd1 <- rep(1,nrow(
    LemonadeRaak_Calibrate))
868 LemonadeRaak_Calibrate$pwo1 <- rep(1,nrow(
    LemonadeRaak_Calibrate))
869 LemonadeRaak_Calibrate$fwo1 <- rep(0,nrow(
    LemonadeRaak_Calibrate))
870 LemonadeRaak_Calibrate$dwo1 <- rep(0,nrow(
    LemonadeRaak_Calibrate))
871 LemonadeRaak_Calibrate$fdwo1 <- rep(0,nrow(
    LemonadeRaak_Calibrate))
872 #2: replace based on the logic of the cutoff value
873 LemonadeRaak_Calibrate$pf1[
    LemonadeRaak_Calibrate$FeatOnly_Adjusted >
    promocutoff1] <- LemonadeRaak_Calibrate$PriceIndex[
    LemonadeRaak_Calibrate$FeatOnly_Adjusted >
    promocutoff1]
874 LemonadeRaak_Calibrate$pd1[
    LemonadeRaak_Calibrate$DispOnly_Adjusted >
    promocutoff1] <- LemonadeRaak_Calibrate$PriceIndex[
    LemonadeRaak_Calibrate$DispOnly_Adjusted >
    promocutoff1]
875 LemonadeRaak_Calibrate$pfd1[
    LemonadeRaak_Calibrate$FeatDisp_Adjusted >
    promocutoff1] <- LemonadeRaak_Calibrate$PriceIndex[
    LemonadeRaak_Calibrate$FeatDisp_Adjusted >
    promocutoff1]
876 LemonadeRaak_Calibrate$pwo1[
    LemonadeRaak_Calibrate$FeatOnly_Adjusted <=
    promocutoff1 &
    LemonadeRaak_Calibrate$DispOnly_Adjusted <=
    promocutoff1 &
    LemonadeRaak_Calibrate$FeatDisp_Adjusted <=
    promocutoff1] <- LemonadeRaak_Calibrate$PriceIndex[
    LemonadeRaak_Calibrate$FeatOnly_Adjusted <=
    promocutoff1 &
    LemonadeRaak_Calibrate$DispOnly_Adjusted <=
    promocutoff1 &
    LemonadeRaak_Calibrate$FeatDisp_Adjusted <=
```

```
876 promocutoff1]
877
878 LemonadeRaak_Calibrate$fwo1[
    LemonadeRaak_Calibrate$PriceIndex > pricecutoff1
    ] <- LemonadeRaak_Calibrate$FeatOnly_Adjusted[
    LemonadeRaak_Calibrate$PriceIndex > pricecutoff1]
879 LemonadeRaak_Calibrate$dwo1[
    LemonadeRaak_Calibrate$PriceIndex > pricecutoff1
    ] <- LemonadeRaak_Calibrate$DispOnly_Adjusted[
    LemonadeRaak_Calibrate$PriceIndex > pricecutoff1]
880 LemonadeRaak_Calibrate$fdwo1[
    LemonadeRaak_Calibrate$PriceIndex > pricecutoff1
    ] <- LemonadeRaak_Calibrate$FeatDisp_Adjusted[
    LemonadeRaak_Calibrate$PriceIndex > pricecutoff1]
881
882
883 summary(LemonadeRaak_Calibrate[c("pf1","pd1","pfd1",
    "pwo1","fwo1","dwo1","fdwo1")])
884 #     pf1               pd1               pfd1
                pwo1
885 #  Min.   :0.5169   Min.   :0.6906   Min.   :0.5176
       Min.   :0.6696
886 #  1st Qu.:1.0000   1st Qu.:1.0000   1st Qu.:1.0000
       1st Qu.:0.9933
887 #  Median :1.0000   Median :1.0000   Median :1.0000
       Median :0.9995
888 #  Mean   :0.9856   Mean   :0.9990   Mean   :0.9914
       Mean   :0.9901
889 #  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
       3rd Qu.:1.0000
890 #  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
       Max.   :1.0000
891 #       fwo1              dwo1              fdwo1
892 #  Min.   :  0.000   Min.   : 0.0000   Min.   : 0.
    0000
893 #  1st Qu.:  0.000   1st Qu.: 0.0000   1st Qu.: 0.
    0000
894 #  Median :  0.000   Median : 0.0000   Median : 0.
    0000
895 #  Mean   :  1.274   Mean   : 0.7898   Mean   : 0.
    9627
```

```
896 #  3rd Qu.:  0.000   3rd Qu.: 0.0000   3rd Qu.: 0.
    0000
897 #  Max.   :100.000   Max.   :89.0000   Max.   :73.
    9837
898
899 library(Hmisc)
900 rcorr(as.matrix(LemonadeRaak_Calibrate[c("pf1","pd1"
    ,"pfd1","pwo1","fwo1","dwo1","fdwo1")]))
901 #       pf1    pd1    pfd1   pwo1   fwo1   dwo1
    fdwo1
902 # pf1          0.0000 0.0000 0.0039 0.0000 0.4351 0.
    0000
903 # pd1   0.0000        0.0000 0.3996 0.3521 0.0000 0.
    2673
904 # pfd1  0.0000 0.0000        0.0236 0.0000 0.9514 0.
    0000
905 # pwo1  0.0039 0.3996 0.0236        0.2744 0.0010 0.
    1018
906 # fwo1  0.0000 0.3521 0.0000 0.2744        0.6104 0.
    0000
907 # dwo1  0.4351 0.0000 0.9514 0.0010 0.6104        0.
    1320
908 # fdwo1 0.0000 0.2673 0.0000 0.1018 0.0000 0.1320
909
910
911 # HOOGVLIET
912 # Recoded model: Hoogvliet (removed old promotion
    and pricing variables + added new variables)
913 MultiplicativeHoogvliet_Recoded <- lm(log(UnitSales
    ) ~ log(PricePUSlimpie)+log(PricePUPL)
914                                         +log(pf1)+log(
    pd1)+log(pfd1)+log(pwo1)+fwo1+dwo1+fdwo1
915                                         + log(
    UnitSalesLag) + log(Temp+273) + WeekAverage,
916                                         data =
    LemonadeRaak_Calibrate[LemonadeRaak_Calibrate$Chain
     == "Hoogvliet", ])
917
918 vif(MultiplicativeHoogvliet_Recoded) # All VIF's are
     less than 5!
919 # log(PricePUSlimpie)      log(PricePUPL
```

```
)             log(pf1)             log(pd1)
#             1.185285             1.245755
             4.963173             1.195184
#           log(pfd1)            log(pwo1
)                fwo1                 dwo1
#            4.959370             1.094106
             2.250392             1.173885
#               fdwo1   log(UnitSalesLag)     log(
Temp + 273)         WeekAverage
#            2.573918             1.115350
             1.104925             1.159633

# Next we will build a recoded REDUCED model based
on
# which variables may not have enough observations (
5 observation rule of thumb)
sum(LemonadeRaak_Calibrate[
LemonadeRaak_Calibrate$Chain == "Hoogvliet",c("pf1"
)] < 1) #17 observations
sum(LemonadeRaak_Calibrate[
LemonadeRaak_Calibrate$Chain == "Hoogvliet",c("pd1"
)] < 1) #2 observations <- too few observations
sum(LemonadeRaak_Calibrate[
LemonadeRaak_Calibrate$Chain == "Hoogvliet",c("pfd1"
)] < 1)  #16 observations
sum(LemonadeRaak_Calibrate[
LemonadeRaak_Calibrate$Chain == "Hoogvliet",c("pwo1"
)] < 1)  #53  observations
sum(LemonadeRaak_Calibrate[
LemonadeRaak_Calibrate$Chain == "Hoogvliet",c("fwo1"
)] > 0)  #11 observations
sum(LemonadeRaak_Calibrate[
LemonadeRaak_Calibrate$Chain == "Hoogvliet",c("dwo1"
)] > 0) #5 observations
sum(LemonadeRaak_Calibrate[
LemonadeRaak_Calibrate$Chain == "Hoogvliet",c("fdwo1
")] > 0) #10 observations

# Recoded REDUCED model: Hoogvliet (removed new
variables that didn't have enough observations for
the chain)
```

```r
MultiplicativeHoogvliet_Recoded.REDUCED <- lm(log(UnitSales) ~ log(PricePUSlimpie)+log(PricePUPL)+log(pf1)+log(pfd1)+log(pwo1)+fwo1+dwo1+fdwo1+log(UnitSalesLag) + log(Temp+273) + WeekAverage, data = LemonadeRaak_Calibrate[LemonadeRaak_Calibrate$Chain == "Hoogvliet", ])


vif(MultiplicativeHoogvliet_Recoded.REDUCED)
# log(PricePUSlimpie)      log(PricePUPL)            log(pf1)            log(pfd1)
#            1.183987            1.241649            4.932409            4.917970
#           log(pwo1)                fwo1                dwo1               fdwo1
#            1.092554            2.249794            1.113537            2.537682
#   log(UnitSalesLag)     log(Temp + 273)         WeekAverage
#            1.110788            1.085707            1.158645

# Conclusion and 2 reasons
# All VIFs are within reasonable values for marketing variables (1)
# furthermore, as we are building a predictive model it is also reasonable to not exclude pf1 and pfd1 (2)
summary(MultiplicativeHoogvliet_Recoded.REDUCED)
#                       Estimate Std. Error t value Pr(>|t|)
# (Intercept)         -1.099e+01  3.978e+00  -2.764 0.006331 **
# log(PricePUSlimpie)  3.279e-02  1.569e-01   0.209 0.834717
# log(PricePUPL)       6.345e-01  2.423e-01   2.618 0.009617 **
# log(pf1)            -1.898e+00  3.760e-01  -5.048
```

```
959 1.12e-06 ***
960 # log(pfd1)            -1.395e+00  3.757e-01  -3.714
    0.000275 ***
961 # log(pwo1)             1.512e+00  1.632e+00   0.926
    0.355600
962 # fwo1                 -3.469e-04  1.788e-03  -0.194
    0.846348
963 # dwo1                 -5.102e-03  7.034e-03  -0.725
    0.469231
964 # fdwo1                 7.545e-03  1.715e-03   4.399
    1.89e-05 ***
965 # log(UnitSalesLag)     1.621e-01  3.783e-02   4.285
    3.02e-05 ***
966 # log(Temp + 273)       3.231e+00  7.082e-01   4.561
    9.56e-06 ***
967 # WeekAverage           2.712e-04  9.023e-05   3.006
    0.003042 **
968 # ---
969 # Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
     '.' 0.1 ' ' 1
970 #
971 # Residual standard error: 0.1899 on 174 degrees of
    freedom
972 #   (1 observation deleted due to missingness)
973 # Multiple R-squared:  0.7755,  Adjusted R-squared
    :  0.7613
974 # F-statistic: 54.63 on 11 and 174 DF,  p-value: < 2
    .2e-16
975
976
977 # Heteroscedasticity: Hoogvliet ----
978 # VISUAL TEST: seems there are no Heteroscedasticity
     issues
979 df.heteroscedasticity.visual <-
    LemonadeRaak_Calibrate[LemonadeRaak_Calibrate$Chain
     == "Hoogvliet", ]
980 is.na(df.heteroscedasticity.visual$UnitSalesLag) #NA
     in row 1
981 df.heteroscedasticity.visual <- df.
    heteroscedasticity.visual[-c(1),] #removed the NA in
     UnitSalesLag
```

```
df.heteroscedasticity.visual$residuals <-
MultiplicativeHoogvliet_Recoded.REDUCED$residuals
ggplot(data = df.heteroscedasticity.visual, aes(y
 = residuals, x = week_yrs)) + geom_point(col = '
blue') + geom_abline(slope = 0)

# FORMAL TEST - Goldfield-Quandt: insignificant =>
Heteroscedasticity is not present
#https://www.statology.org/goldfeld-quandt-test-in-
r/
# model: The linear regression model created by the
 lm() command.
# order.by: The predictor variable(s) in the model.
# data: The name of the dataset.
# fraction*: The number of central observations to
remove from the dataset. Typically we choose to
remove around 20% of the total observations.
gqtest(MultiplicativeHoogvliet_Recoded.REDUCED,
order.by = ~log(PricePUSlimpie)+log(PricePUPL)+log(
pf1)+log(pfd1)+log(pwo1)+fwo1+dwo1+fdwo1+log(
UnitSalesLag)+log(Temp+273)+WeekAverage, data =
LemonadeRaak_Calibrate[LemonadeRaak_Calibrate$Chain
 == "Hoogvliet", ], fraction = 36)
# GQ = 0.38192, df1 = 63, df2 = 63, p-value = 0.
9999
# alternative hypothesis: variance increases from
segment 1 to 2


# Autocorrelation: Hoogvliet ----
dwtest(MultiplicativeHoogvliet_Recoded.REDUCED)
# DW = 1.6873, p-value = 0.007732
# alternative hypothesis: true autocorrelation is
greater than 0
# FINDINGS: We can reject the null hypothesis,
therefore there is autoccrelation
# HOWEVER: the DW is 1.68 which may signify lack of
 autocorrelation since it's close to 2
# CONCLUSION: Further testing is needed

# We use the DW statistic (1.6873); Number of
```

```
1004 observations: n = 186; Number of Independent Vars:
     k' = 11
1005 # to look at the Durbin Watson table (https://www3.
     nd.edu/~wevans1/econ30331/Durbin_Watson_tables.pdf)
1006 # => dL = 1.561   dU = 1.791 => This puts 1.6873 in
      the grey zone. The autocorrelation is not
     significant.
1007
1008 # To further test this we employ ACF
1009 #(https://www.codingprof.com/3-easy-ways-to-test-
     for-autocorrelation-in-r-examples/)
1010 library(stats)
1011 acf(MultiplicativeHoogvliet_Recoded.
     REDUCED$residuals, type='correlation')
1012 # The plot also shows no autocorrelation
1013
1014 # Finally, we tested with Breusch-Godfrey
1015 bgtest(MultiplicativeHoogvliet_Recoded.REDUCED,
     order = 3)
1016 #   Breusch-Godfrey test for serial correlation of
     order up to 3
1017 #
1018 # data:  MultiplicativeHoogvliet_Recoded.REDUCED
1019 # LM test = 6.7218, df = 3, p-value = 0.08131
1020 # Conclusion: We cannot reject the null hypothesis
      => there is not autocorrelation
1021
1022
1023 # Nonnormality: Hoogvliet ----
1024
1025 # (1) VISUAL TESTING:
1026
1027 # HISTOGRAM+curve: we look at the distribution,
     which looks normal
1028 hist(MultiplicativeHoogvliet_Recoded.
     REDUCED$residuals,probability = TRUE)
1029 curve(dnorm(x, mean=mean(
     MultiplicativeHoogvliet_Recoded.REDUCED$residuals
     ), sd=sd(MultiplicativeHoogvliet_Recoded.
     REDUCED$residuals)), add=TRUE, col="red")
1030
```

```
1031 # QQ-Plot: we are now looking if the tails, they
     seem to deviate on both ends
1032 # => this may indicate non-normality, therefore we
     will perform formal tests.
1033 res_std <- rstandard(
     MultiplicativeHoogvliet_Recoded.REDUCED)
1034 qqnorm(res_std,ylab="Standardized Residuals",xlab="
     Normal Scores")
1035 qqline(res_std,col="red")
1036
1037 # (2) FORMAL TESTING:
1038 # each test differs in how it checks to see if your
      distribution matches the normal. For example, the
     KS test looks at the quantile where your empirical
     cumulative distribution function differs maximally
     from the normal's theoretical cumulative
     distribution function. This is often somewhere in
     the middle of the distribution, which isn't where
     we typically care about mismatches. The SW test
     focuses on the tails, which is where we typically
     do care if the distributions are similar. As a
     result, the SW is usually preferred.
1039 # so, there's no normality issue in our case, as
     Shapiro-Wilk normality test gives a p-value of 0.
     02329.
1040
1041 #SHAPIRO-WILK test: significant => nonnormality
1042 shapiro.test(MultiplicativeHoogvliet_Recoded.
     REDUCED$residuals)
1043 # W = 0.9015, p-value = 8.925e-10
1044
1045 #LILLIE/KOLMOGOROV-SMIRNOV test: significant =>
     nonnormality
1046 library(nortest)
1047 lillie.test(MultiplicativeHoogvliet_Recoded.
     REDUCED$residuals)
1048 # D = 0.12929, p-value = 4.963e-08
1049
1050 #JARQUE-BERA test: significant => nonnormality
1051 library(fBasics)
1052 jarqueberaTest(MultiplicativeHoogvliet_Recoded.
```

```
REDUCED$residuals)
#   STATISTIC:
#     X-squared: 360.6834
#   P VALUE:
#     Asymptotic p Value: < 2.2e-16

# CONCLUSION: We can conclude there is nonnormality in the distribution of the residuals.


# SOLUTION of Nonnormality: Bootstrapping
# Note: this bootstrapping generates slightly different p-values each time
set.seed(444) #does not affect bootstrapping p-values
library(boot)
lmbootstrap <- function(formula,data){
  boot.run <- function(data, indices){
    data <- data[indices,] # select obs. in bootstrap sample
    mod <- lm(formula, data=data)
    coefficients(mod) # return coefficient vector
  }
  boot_aux <- boot(data,boot.run, 1999)
  Coefficient <- names(boot_aux$t0)
  boot_out <- data.frame(Coefficient)
  for (i in 1:length(boot_aux$t0)) {
    boot_out$Estimate[i] <- boot_aux$t0[i]
    boot_out$Std.Error[i] <- sd(boot_aux$t[,i])
    boot_out$Bias[i] <- mean(boot_aux$t[,i])-boot_aux$t0[i]
    boot_out$p_bootstrap[i] <- dt(mean(boot_aux$t[,i])/sd(boot_aux$t[,i]),boot_aux$R-dim(boot_aux$t)[2])
  }
  return(boot_out)
}

MultiplicativeHoogvliet_Recoded.REDUCED_BOOT <- lmbootstrap(log(UnitSales) ~ log(PricePUSlimpie)+ log(PricePUPL)
```

```
1084
              +log(pf1)+log(pfd1)+log(pwo1)+fwo1+dwo1+
    fdwo1
1085
              + log(UnitSalesLag) + log(Temp+273) +
    WeekAverage,
1086
              data = LemonadeRaak_Calibrate[
    LemonadeRaak_Calibrate$Chain == "Hoogvliet", ])
1087
1088 #            Coefficient      Estimate    Std.Error
              Bias    p_bootstrap       original p-
    value
1089 # 1          (Intercept) -1.099416e+01 3.811649e+00
      6.284694e-01   9.933567e-03 **       0.006331 **
1090 # 2  log(PricePUSlimpie)  3.278785e-02 1.090610e-01
     -5.612982e-03   3.866936e-01          0.834717
1091 # 3       log(PricePUPL)  6.344801e-01 2.185924e-01
     -2.578733e-02   8.308538e-03 **       0.009617 **
1092 # 4             log(pf1) -1.897935e+00 8.007942e-01
     -2.465592e-01   1.110742e-02 *        1.12e-06 ***
1093 # 5            log(pfd1) -1.395303e+00           NA
                NA              NA          0.000275
     ***  <- ? question for teacher below
1094 # 6            log(pwo1)  1.511665e+00 9.921345e-01
     -2.933107e-01   1.876519e-01          0.355600
1095 # 7                fwo1 -3.469044e-04 1.269894e-02
     -3.816791e-03   3.780073e-01          0.846348
1096 # 8                dwo1 -5.101938e-03           NA
                NA              NA          0.469231
          <- ? question for teacher below
1097 # 9               fdwo1  7.545001e-03 1.064314e-02
      2.451891e-03   2.565808e-01          1.89e-05 ***
1098 # 10   log(UnitSalesLag)  1.620716e-01 6.730316e-02
      1.459285e-02   1.277964e-02 *        3.02e-05 ***
1099 # 11     log(Temp + 273)  3.230611e+00 7.059730e-01
     -1.329554e-01   2.743028e-05 ***      9.56e-06 ***
1100 # 12         WeekAverage  2.711903e-04 8.834144e-05
     -1.487158e-05   5.967026e-03 **       0.003042 **
1101 # ---
1102 # Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
```

```
1102  '.' 0.1 ' ' 1
1103
1104 # ORIGINAL MODEL (BEFORE BOOTSTRAPPTING)
1105 #                       Estimate Std. Error t value
     Pr(>|t|)
1106 # (Intercept)         -1.099e+01  3.978e+00  -2.764
      0.006331 **
1107 # log(PricePUSlimpie)  3.279e-02  1.569e-01   0.209
      0.834717
1108 # log(PricePUPL)       6.345e-01  2.423e-01   2.618
      0.009617 **
1109 # log(pf1)            -1.898e+00  3.760e-01  -5.048
      1.12e-06 ***
1110 # log(pfd1)           -1.395e+00  3.757e-01  -3.714
      0.000275 ***
1111 # log(pwo1)            1.512e+00  1.632e+00   0.926
      0.355600
1112 # fwo1                -3.469e-04  1.788e-03  -0.194
      0.846348
1113 # dwo1                -5.102e-03  7.034e-03  -0.725
      0.469231
1114 # fdwo1                7.545e-03  1.715e-03   4.399
      1.89e-05 ***
1115 # log(UnitSalesLag)    1.621e-01  3.783e-02   4.285
      3.02e-05 ***
1116 # log(Temp + 273)      3.231e+00  7.082e-01   4.561
      9.56e-06 ***
1117 # WeekAverage          2.712e-04  9.023e-05   3.006
      0.003042 **
1118
1119
1120
1121
1122 # <--- QUESTIONS FOR TUTORIAL --->
1123
1124 # (0) Is it ok to leave the variables in the
     REDUCED model of Hoogvliet that have VIF ~4.9? Is
     the reasoning ok?
1125
1126 # (1) Is it REALLY nonnormal? I read these tests
     are way too strict and the first plot seems fine.
```

```
1127 # However, the QQ does show the two tails like in
     the summary (page 111)
1128
1129 # (2) Is it okay that bootstrapping generates
     slightly different p-values each time?
1130
1131 # (3) Why does our bootstrapped model have NAs
     while the non-bootstrapped doesn't?
1132
1133 # Statistical Validity Testing ----
1134 ## Information criteria ----
1135 #Calculate AIC
1136
1137 extractAIC(MultiplicativeHoogvliet_Recoded.REDUCED)
1138 # [1]   12.0000 -606.3786
1139
1140 #Check calculations
1141 T <- length(MultiplicativeHoogvliet_Recoded.
     REDUCED$residuals)            #Number of
     observations
1142 K <- length(MultiplicativeHoogvliet_Recoded.
     REDUCED$coefficients)-1       #Number of regressors

1143 RSS_MultiplicativeHoogvliet_Recoded.REDUCED <-
     anova(MultiplicativeHoogvliet_Recoded.REDUCED)$`Sum
      Sq`[K+1]
1144
1145 #Calculate AIC
1146 AIC_check <- T*log(
     RSS_MultiplicativeHoogvliet_Recoded.REDUCED/T)+2*(K
     +1) #Correct!!!
1147 AIC_check # AIC: -606.3786
1148 #Calculate BIC
1149 BIC <- T*log(RSS_MultiplicativeHoogvliet_Recoded.
     REDUCED/T)+log(T)*(K+1)
1150 BIC # BIC:  -567.6696
1151
1152 # Robust estimation:
1153 plot(MultiplicativeHoogvliet_Recoded.
     REDUCED$residuals)
1154 plot(rstudent(MultiplicativeHoogvliet_Recoded.
```

```
1154 REDUCED))
1155 # There is no discernible pattern! GOOD.
1156
1157 # Assessing Outliers
1158 outlierTest(MultiplicativeHoogvliet_Recoded.REDUCED
     ) # Bonferonni p-value for most extreme obs
1159 # rstudent unadjusted p-value Bonferroni p
1160 # 4434 -6.290424         2.5077e-09   4.6643e-07
1161 # 4440  5.847877         2.4289e-08   4.5178e-06
1162 # Points 4440 and 4434 are outliers
1163 qqPlot(MultiplicativeHoogvliet_Recoded.REDUCED,
     main="QQ Plot") #qq plot for studentized resid
1164 # 4434 4440
1165 # 66   72
1166
1167 library(sur)
1168
1169 h <- leverage(MultiplicativeHoogvliet_Recoded.
     REDUCED) # leverage values
1170
1171 leveragePlots(MultiplicativeHoogvliet_Recoded.
     REDUCED) # leverage plots
1172
1173 library(car)
1174
1175 # Influential Observations
1176 # added variable plots
1177 avPlots(MultiplicativeHoogvliet_Recoded.REDUCED)
1178 # Cook's D plot
1179 # identify D values > 4/(T-K-1)
1180 cutoff <- 4/((nrow(LemonadeRaak_Calibrate[
     LemonadeRaak_Calibrate$Chain == "Hoogvliet", ])-
     length(MultiplicativeHoogvliet_Recoded.
     REDUCED$coefficients)-2))
1181 plot(MultiplicativeHoogvliet_Recoded.REDUCED, which
     =4, cook.levels=cutoff)
1182 ## point 4528 has a Cook's above 1, this is a high
     influence point
1183
1184 # Influence Plot
1185 influencePlot(MultiplicativeHoogvliet_Recoded.
```

```
1185 REDUCED, id.method="identify", main="Influence Plot
     ", sub="Circle size is proportial to Cook's
     Distance" )
1186 ## point 4440, 4434, 4483, 4528 are high leverage
     outliers
1187 #        StudRes       Hat     CookD
1188 # 4434 -6.290424 0.2061735 0.7010263
1189 # 4440  5.847877 0.2232314 0.6877700
1190 # 4483  3.039328 0.5631039 0.9473182
1191 # 4528  1.938606 0.8691999 2.0487040
1192
1193 # We can see the following:
1194 # Observation 4528 is a high-leverage point (but no
      outlier) and has the highest influence of all on
     the regression.
1195 # Observations 4440, 4483, 4434 are a high-leverage
      OUTLIER point with moderately high influence on
     the regression
1196
1197 cooksd <- cooks.distance(
     MultiplicativeHoogvliet_Recoded.REDUCED)
1198
1199 library(robustbase)
1200
1201 MultiplicativeHoogvliet_Recoded.REDUCEDRobust <-
     lmrob(log(UnitSales) ~ log(PricePUSlimpie)+log(
     PricePUPL)
1202                                                 +log(
     pf1)+log(pfd1)+log(pwo1)+fwo1+dwo1+fdwo1
1203                                                 + log(
     UnitSalesLag) + log(Temp+273) + WeekAverage,
1204                                                 data =
     LemonadeRaak_Calibrate[LemonadeRaak_Calibrate$Chain
      == "Hoogvliet", ])
1205 summary(MultiplicativeHoogvliet_Recoded.
     REDUCEDRobust)
1206 # Coefficients:
1207 #                          Estimate   Std. Error t
     value Pr(>|t|)
1208 #   (Intercept)         -4.395e+00  2.257e+00  -1.
     948  0.05307 .
```

```
#   log(PricePUSlimpie) -4.601e-02  7.975e-02  -0.577  0.56476
#   log(PricePUPL)       3.762e-01  1.247e-01   3.016  0.00294 **
#   log(pf1)            -3.217e+00  1.415e-01 -22.740  < 2e-16 ***
#   log(pfd1)            2.947e-01  1.546e-01   1.906  0.05831 .
#   log(pwo1)            5.783e-01  7.383e-01   0.783  0.43448
#   fwo1                 5.198e-04  5.790e-04   0.898  0.37054
#   dwo1                 1.435e-02  4.743e-03   3.026  0.00286 **
#   fdwo1                8.310e-03  1.564e-03   5.314 3.25e-07 ***
#   log(UnitSalesLag)    2.497e-01  2.380e-02  10.492  < 2e-16 ***
#   log(Temp + 273)      1.948e+00  4.030e-01   4.833 2.94e-06 ***
#   WeekAverage          1.220e-04  4.969e-05   2.456  0.01502 *
#   ---
#   Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

#   Robust residual standard error: 0.09744
#   Multiple R-squared:  0.9028,    Adjusted R-squar:  0.8967
#   Convergence in 22 IRWLS iterations

NullModel <-  lmrob(log(UnitSales)~1,data=LemonadeRaak_Calibrate[LemonadeRaak_Calibrate$Chain == "Hoogvliet", ])

anova(NullModel,MultiplicativeHoogvliet_Recoded.REDUCEDRobust,test="Deviance")

# Predictive Validity Testing ----

#1: create and populate variables to the validation
```

```
1233  sample (which we replace in step 2 based on the
     cutoff)
1234 LemonadeRaak$pf1 <- rep(1,nrow(LemonadeRaak))
1235 LemonadeRaak$pd1 <- rep(1,nrow(LemonadeRaak))
1236 LemonadeRaak$pfd1 <- rep(1,nrow(LemonadeRaak))
1237 LemonadeRaak$pwo1 <- rep(1,nrow(LemonadeRaak))
1238 LemonadeRaak$fwo1 <- rep(0,nrow(LemonadeRaak))
1239 LemonadeRaak$dwo1 <- rep(0,nrow(LemonadeRaak))
1240 LemonadeRaak$fdwo1 <- rep(0,nrow(LemonadeRaak))
1241 #2: replace based on the logic of the cutoff value
1242 LemonadeRaak$pf1[LemonadeRaak$FeatOnly_Adjusted >
     promocutoff1] <- LemonadeRaak$PriceIndex[
     LemonadeRaak$FeatOnly_Adjusted > promocutoff1]
1243 LemonadeRaak$pd1[LemonadeRaak$DispOnly_Adjusted >
     promocutoff1] <- LemonadeRaak$PriceIndex[
     LemonadeRaak$DispOnly_Adjusted > promocutoff1]
1244 LemonadeRaak$pfd1[LemonadeRaak$FeatDisp_Adjusted >
     promocutoff1] <- LemonadeRaak$PriceIndex[
     LemonadeRaak$FeatDisp_Adjusted > promocutoff1]
1245 LemonadeRaak$pwo1[LemonadeRaak$FeatOnly_Adjusted
      <= promocutoff1 & LemonadeRaak$DispOnly_Adjusted
      <= promocutoff1 & LemonadeRaak$FeatDisp_Adjusted
      <= promocutoff1] <- LemonadeRaak$PriceIndex[
     LemonadeRaak$FeatOnly_Adjusted <= promocutoff1 &
     LemonadeRaak$DispOnly_Adjusted <= promocutoff1 &
     LemonadeRaak$FeatDisp_Adjusted <= promocutoff1]
1246
1247 LemonadeRaak$fwo1[LemonadeRaak$PriceIndex >
     pricecutoff1] <- LemonadeRaak$FeatOnly_Adjusted[
     LemonadeRaak$PriceIndex > pricecutoff1]
1248 LemonadeRaak$dwo1[LemonadeRaak$PriceIndex >
     pricecutoff1] <- LemonadeRaak$DispOnly_Adjusted[
     LemonadeRaak$PriceIndex > pricecutoff1]
1249 LemonadeRaak$fdwo1[LemonadeRaak$PriceIndex >
     pricecutoff1] <- LemonadeRaak$FeatDisp_Adjusted[
     LemonadeRaak$PriceIndex > pricecutoff1]
1250 #3:
1251 LemonadeRaak_Validate <- LemonadeRaak[
     LemonadeRaak$Date >= "2020-08-01",]
1252 MultiplicativeHoogvliet_Recoded.REDUCEDPredictions
      <- predict(MultiplicativeHoogvliet_Recoded.REDUCED
```

```
1252 ,newdata = LemonadeRaak_Validate[
     LemonadeRaak_Validate$Chain == "Hoogvliet", ])
1253
1254 # quick look at RMSE (predictive errors) by using
     packages
1255 library(caret)
1256 library(glmnet)
1257
1258 real_value<- LemonadeRaak_Validate[
     LemonadeRaak_Validate$Chain == "Hoogvliet", ]
     $UnitSales
1259 MultiplicativeHoogvliet_Recoded.REDUCEDPredictions
      <- exp(1)^MultiplicativeHoogvliet_Recoded.
     REDUCEDPredictions
1260 rmse <- RMSE(MultiplicativeHoogvliet_Recoded.
     REDUCEDPredictions, real_value)
1261 # RMSE: 119348.2
1262
1263 # predictive graphs
1264
1265 plot(1:208,LemonadeRaak[LemonadeRaak$Chain == "
     Hoogvliet",]$UnitSales,col="black",bg="black",pch=
     21,xlab = "Weeks",ylab="Sales of Raak at Hoogvliet"
     ,ylim = c(0,60000),main="Predictive validity for
     Raak Model at Hoogvliet")
1266 lines(1:208,LemonadeRaak[LemonadeRaak$Chain == "
     Hoogvliet",]$UnitSales)
1267 # MultiplicativeHoogvliet_Recoded.
     REDUCEDPredictions <- exp(1)^
     MultiplicativeHoogvliet_Recoded.REDUCEDPredictions
1268
1269 lines(188:208,MultiplicativeHoogvliet_Recoded.
     REDUCEDPredictions,pch=21,col="red",bg="red",type="
     o",lty=2)
1270 sd_residuals <- summary(
     MultiplicativeHoogvliet_Recoded.REDUCED)$sigma
1271 RaakHoogvlietSalesFit <- exp(
     MultiplicativeHoogvliet_Recoded.REDUCED$fitted.
     values)*exp(1/2*sd_residuals^2)
1272
1273 lines(2:187,RaakHoogvlietSalesFit,pch=21,col="blue"
```

```
1273 ,bg="blue",type="o",lty=2)
1274 lines(c(187.5,187.5),c(0,60000),col="black",lty=2)
1275 text(130,50000,pos=2,"Estimation sample",cex = .9)
1276 text(187.5,50000,pos=4,"Validation sample",cex = .9
     )
1277
1278 legend(-5, 60000, c("Observed values","Fitted
     values","Predicted values"), col=c("black","blue","
     red"), pt.bg=c("black","blue","red"), pch=c(21,21,
     21), lty=c(1,2,2),cex = 0.6)
1279
1280 APE <- sum(real_value-
     MultiplicativeHoogvliet_Recoded.REDUCEDPredictions
     )/(21)
1281 # -64213.43 the predictions are larger than the
     actual values.
1282 ASPE <- sum((real_value-
     MultiplicativeHoogvliet_Recoded.REDUCEDPredictions
     )^2)/(21)
1283 # 14243994247 after weighted larger errors.
1284 RASPE <- sqrt(ASPE)
1285 # 119348.2 also known as RMSE, the lower the better
     , it seems good.
1286 MAPE <- (1/21)*sum(abs( (real_value-
     MultiplicativeHoogvliet_Recoded.REDUCEDPredictions
     ) /
1287                            (real_value) ) )
1288 # 4.857552 MAPE values range from 0 to infinity,
     where the lower the value the more accurate the
     predictions are.
1289 RAE <- sum(abs(real_value-
     MultiplicativeHoogvliet_Recoded.REDUCEDPredictions
     )) /
1290   sum(abs(real_value-LemonadeRaak[
     LemonadeRaak$Chain == "Hoogvliet",]$UnitSales[187:
     207]))
1291 # 7.828423 for the validation sample, model didnt
     outperform naive model.
1292 TheilsU <- sqrt( sum((real_value-
     MultiplicativeHoogvliet_Recoded.REDUCEDPredictions
     )^2) /
```

```
1293                       sum((real_value-LemonadeRaak[
    LemonadeRaak$Chain == "Hoogvliet",]$UnitSales[187:
    207])^2))
1294 # 8.692934 for the validation sample, model didnt
    outperform naive model.
1295
1296 # anti log of model's coefficients----
1297
1298 MultiplicativeHoogvliet_Recoded.REDUCED <- lm(log(
    UnitSales) ~ log(PricePUSlimpie)+log(PricePUPL)
1299                                                 +log(
    pf1)+log(pfd1)+log(pwo1)+fwo1+dwo1+fdwo1
1300                                                 + log
    (UnitSalesLag) + log(Temp+273) + WeekAverage,
1301                                                 data
     = LemonadeRaak_Calibrate[
    LemonadeRaak_Calibrate$Chain == "Hoogvliet", ])
1302 summary(MultiplicativeHoogvliet_Recoded.REDUCED)
1303
1304 ## Apply the anti-log transformation to
    alpha_hat_star ----
1305 alpha_hat_star <- summary(
    MultiplicativeHoogvliet_Recoded.REDUCED)
    $coefficients[1,1]
1306 sd_alpha_hat_star <- summary(
    MultiplicativeHoogvliet_Recoded.REDUCED)
    $coefficients[1,2]
1307 alpha_hat <- exp(alpha_hat_star) * exp(-0.5*(
    sd_alpha_hat_star^2))
1308 sprintf("alpha_hat = %.20f", alpha_hat )
1309
1310 ## For beta_log(PricePUSlimpie), the anti-log
    transformation is not needed! ----
1311 sprintf("beta_log(PricePUSlimpie) = %.2f",
    MultiplicativeHoogvliet_Recoded.
    REDUCED$coefficients[2])
1312
1313 ## For beta_log(PricePUPL), the anti-log
    transformation is not needed! ----
1314 sprintf("beta_log(PricePUPL) = %.2f",
    MultiplicativeHoogvliet_Recoded.
```

```
1314 REDUCED$coefficients[3])
1315
1316 ## For beta_log(pf1) , the anti-log transformation
     is not needed! ----
1317 sprintf("beta_log(pf1)  = %.2f",
     MultiplicativeHoogvliet_Recoded.
     REDUCED$coefficients[4])
1318
1319 ## For beta_log(pfd1), the anti-log transformation
     is not needed! ----
1320 sprintf("beta_log(pfd1) = %.2f",
     MultiplicativeHoogvliet_Recoded.
     REDUCED$coefficients[5])
1321
1322 ## For beta_log(pwo1), the anti-log transformation
     is not needed! ----
1323 sprintf("beta_log(pwo1) = %.2f",
     MultiplicativeHoogvliet_Recoded.
     REDUCED$coefficients[6])
1324
1325 ## Apply the anti-log transformation to beta_fwo1
      ----
1326 beta2_hat_star <- MultiplicativeHoogvliet_Recoded.
     REDUCED$coefficients[7]
1327 sd_beta2_hat_star <- summary(
     MultiplicativeHoogvliet_Recoded.REDUCED)
     $coefficients[7,2]
1328 beta2_hat <- exp(beta2_hat_star) * exp(-0.5*(
     sd_beta2_hat_star^2))
1329 sprintf("beta_fwo1 = %.2f", beta2_hat )
1330
1331 ## Apply the anti-log transformation to beta_dwo1
      ----
1332 beta3_hat_star <- MultiplicativeHoogvliet_Recoded.
     REDUCED$coefficients[8]
1333 sd_beta3_hat_star <- summary(
     MultiplicativeHoogvliet_Recoded.REDUCED)
     $coefficients[8,2]
1334 beta3_hat <- exp(beta3_hat_star) * exp(-0.5*(
     sd_beta3_hat_star^2))
1335 sprintf("beta_dwo1 = %.2f", beta3_hat )
```

```
1336
1337 ## Apply the anti-log transformation to beta_fdwo1
      ----
1338 beta4_hat_star <- MultiplicativeHoogvliet_Recoded.
     REDUCED$coefficients[9]
1339 sd_beta4_hat_star <- summary(
     MultiplicativeHoogvliet_Recoded.REDUCED)
     $coefficients[9,2]
1340 beta4_hat <- exp(beta4_hat_star) * exp(-0.5*(
     sd_beta4_hat_star^2))
1341 sprintf("beta_fdwo1 = %.2f", beta4_hat )
1342
1343 ## For beta_log(UnitSalesLag) , the anti-log
     transformation is not needed! ----
1344 sprintf("beta_log(UnitSalesLag)  = %.2f",
     MultiplicativeHoogvliet_Recoded.
     REDUCED$coefficients[10])
1345
1346 ## For beta_log(Temp + 273), the anti-log
     transformation is not needed! ----
1347 sprintf("beta_log(Temp + 273) = %.2f",
     MultiplicativeHoogvliet_Recoded.
     REDUCED$coefficients[11])
1348
1349 ## Apply the anti-log transformation to
     beta_WeekAverage ----
1350 beta5_hat_star <- MultiplicativeHoogvliet_Recoded.
     REDUCED$coefficients[12]
1351 sd_beta5_hat_star <- summary(
     MultiplicativeHoogvliet_Recoded.REDUCED)
     $coefficients[12,2]
1352 beta5_hat <- exp(beta5_hat_star) * exp(-0.5*(
     sd_beta5_hat_star^2))
1353 sprintf("beta_WeekAverage = %.20f", beta5_hat )
1354
1355 ## Obtain the standard deviation of the residuals
      ----
1356 sd_residuals <- summary(
     MultiplicativeHoogvliet_Recoded.REDUCED)$sigma
1357
1358 ## Calculate the fitted values ----
```

```
1359 RaakHoogvlietSalesFit <- exp(
     MultiplicativeHoogvliet_Recoded.REDUCED$fitted.
     values)*exp(1/2*sd_residuals^2)
1360 new_data <- LemonadeRaak_Calibrate[
     LemonadeRaak_Calibrate$Chain == "Hoogvliet", ]
1361 new_data <- new_data[order(new_data$Date),]
1362 view(new_data)
1363
1364 ## Make the comparison plot ----
1365 plot(new_data$Date[2:187],new_data$UnitSales[2:187
     ],type="p",pch=21,bg="dodgerblue",col="dodgerblue",
     xlab = "Weeks",ylab="Sales of Raak at Hoogvliet (
     units)",main = "Comparing actual and fitted sales
     of Raak at Hoogvliet")
1366 lines(new_data$Date[2:187],new_data$UnitSales[2:187
     ],lwd=2,col="dodgerblue")
1367 lines(new_data$Date[2:187],RaakHoogvlietSalesFit,
     col="red",lty=2,lwd=2)
1368 legend("topleft",inset = c(.75,0.02), c("Actual
     sales","Fitted sales"), cex=0.8, col=c("dodgerblue"
     ,"red"), pch=c(21,NA),lty = 1:2,lwd=2)
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
```