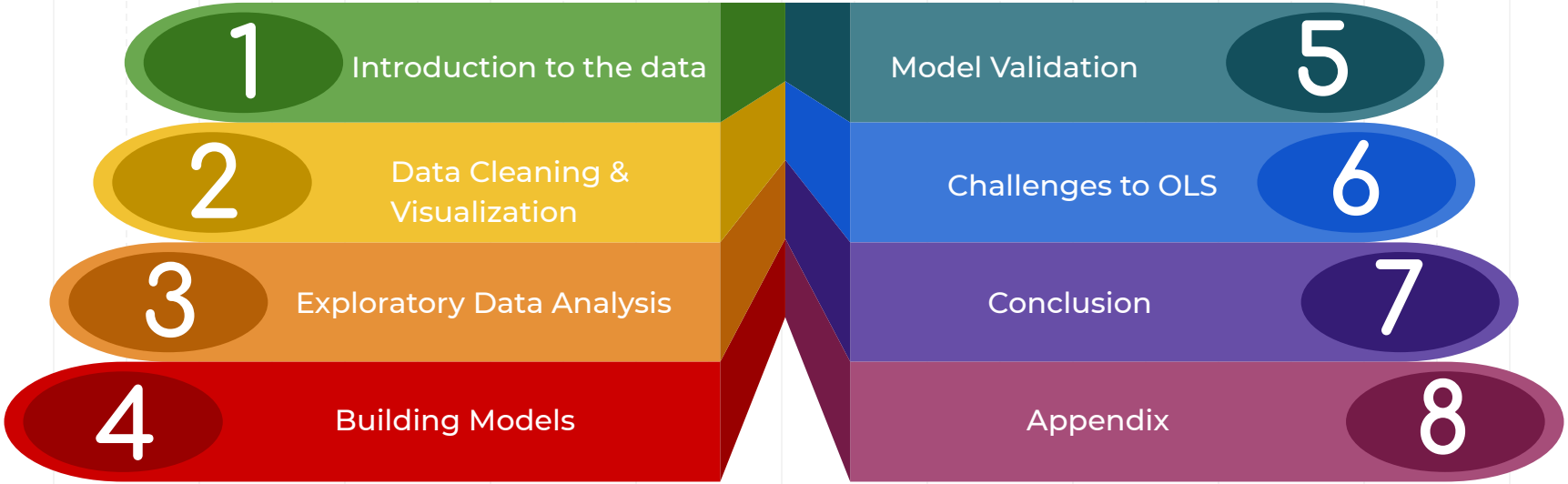
Weekly Sales

Team 3

Raymond Chen, Danielle Rowan,
Scott Su, Shaili Tanna

AGENDA


1
Introduction to the data
2
Data Cleaning & Visualization
3
Exploratory Data Analysis
4
Building Models
5
Model Validation
6
Challenges to OLS
7
Conclusion
8
Appendix

1 Introduction

TO THE DATA

ORIGINAL DATASET

- **Sourced from a Walmart recruiting Kaggle competition**
  - Providing historical weekly sales data for 45 Walmart store locations in different regions for 1.75 years between 2010 and 2012
- **The dataset was provided across 5 files**
  - We merged the features, stores, and train files to create a panel dataset with **421,570** observations and **16** variables
    - Store
    - Type
    - Date
    - IsHoliday
    - Temperature
    - Fuel_Price
    - Dept
    - Weekly_Sales
    - MarkDown 1
    - MarkDown 2
    - MarkDown 3
    - MarkDown 4
    - MarkDown 5
    - CPI
    - Unemployment
    - Size

summary(data)

The summary of this original dataset is available in the appendix as A.1

Data Cleaning & Visualization

2

CLEANING

- **We started with evaluating the structure of the data and adjusting data types**
  - **CPI:** Consumer Price Index by region
    - *From character to numeric*
  - **Unemployment:** the unemployment rate for that region
    - *From character to numeric*
  - **Type:** the type of store, there are 3 ‘A’, ‘B’, ‘C’
    - *From character to factor*
  - **Dept:** there are 99 possible departments for one store to have
    - *From numeric to factor*
  - **Store:** there are 45 stores in the dataset, each with their own unique store ID
    - *From numeric to factor*
  - **MarkDown 1 - MarkDown 5:** Each MarkDown variable represents as promotion associated with specific holidays in terms of sales revenue generated through the promotion
    - *From character to numeric*

str(data)

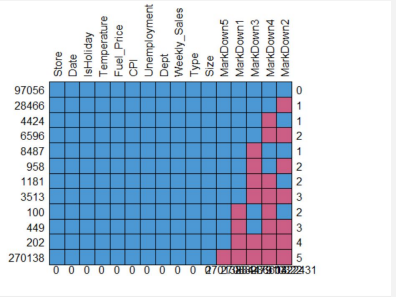
```
data$CPI  <- as.numeric(data$CPI)
data$Unemployment  <- as.numeric(data$Unemployment)
data$Type  <- as.factor(data$Type)
data$Dept  <- as.factor(data$Dept)
data$Store  <- as.factor(data$Store)
data$MarkDown1 <- as.numeric(data$MarkDown1)
data$MarkDown2 <- as.numeric(data$MarkDown2)
data$MarkDown3<- as.numeric(data$MarkDown3)
data$MarkDown4 <- as.numeric(data$MarkDown4)
data$MarkDown5 <- as.numeric(data$MarkDown5)
str(data)
```

The structure of the data before and after is available in the appendix as A.2

MISSING DATA

- **We then investigated any missings values, in the conversion of data types we the introduction of NAs through coercion in the MarkDown variables**
  - All other variables are fully observed
  - From the mice package we see that there are only **97056 (23%)** of records completely observed for all variables
    - MarkDown 1: 270,570 missing values
    - MarkDown2: 310.322 missing values
    - MarkDown 3: 284,479 missing values
    - MarkDown 4: 286,603 missing values
    - MarkDown 5: 270,138 missing values

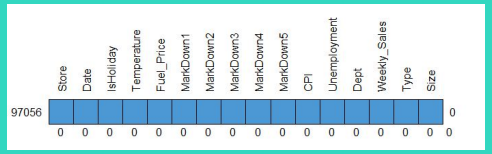
```
library(mice)
md.pattern(data,
rotate.names = TRUE)
```



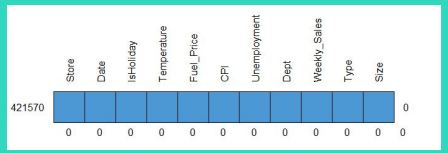
The details of missing values before and after are available in the appendix as A3.

MISSING DATA

- **Option 1: Drop all observations with missing values**
  - **data <- na.exclude(data)**
  - The resulting dataset was only 23% of out original data, determine not to be the best option to address NAs



- **Option 2: Drop all the MarkDown variables from the dataset**
  - **data <- data[-c(6:10)]**
  - The resulting dataset retain 100% of the observations, we determined that the other variables we were left with served our needs for this project and did not need the MarkDown variables




```
library(mice)
md.pattern(data,
rotate.names = TRUE)
```


```
No need for mice. This data set is completely observed.
```

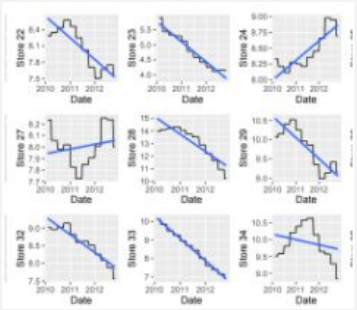
The details of missing values before and after are available in the appendix as A3.

DETERMINING WHICH STORE TO FOCUS ON

- **From the grids in the appendix A.5 we evaluated how each store varied over time for each variables:**
  - **CPI:** All stores displayed the same pattern with different ranges
  - **Temperature :** All stores displayed the same pattern with different ranges
  - **Unemployment:** the majority of stores show a decrease in unemployment over time, but at varying slopes and there are stores that display the region where these stores are located actually saw an increase in unemployment
  - **Fuel Prices:** All stores increase over time with varying slopes unrelated to the time of the year
  - **Weekly Sales:** The majority of stores displayed the same seasonal peaks during the winter holidays with minimal growth year over year although a few stodd which varied greatly from this pattern
    - *Store 44 displays a faster rate of weekly sales YOY*
    - *Store 38 displays a drop in weekly sales YOY*
    - *Stores 38, 33, and 42 do not display peaks for winter holiday sales*
  - We were able to rule out these stores from our selection as they would not represent the whole set accurately

```
store1 <- data[which(data['Store']==1),]
```

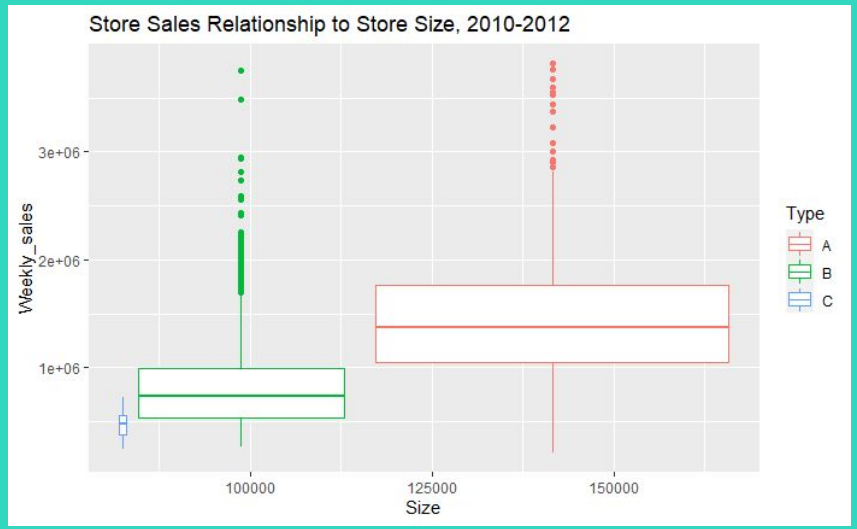
The rest of this process is available in the appendix as A.4

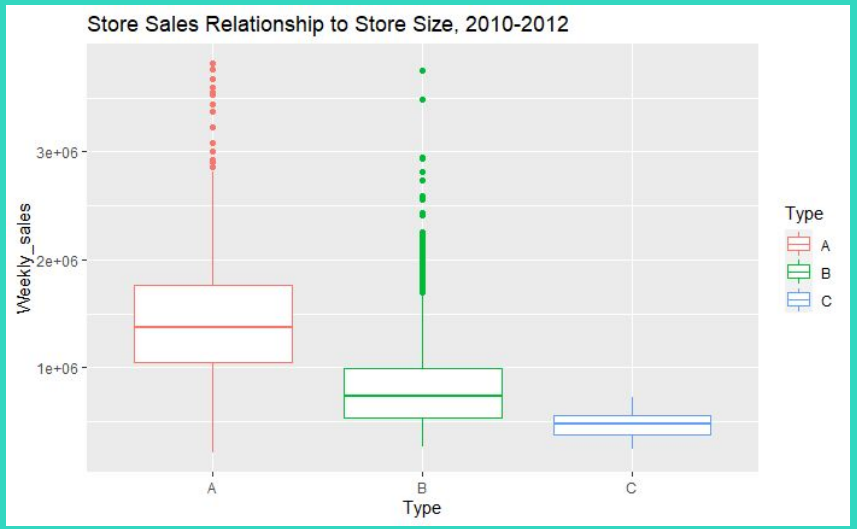


The complete visual analysis of all stores data is available in the appendix as A.5

STORE TYPE

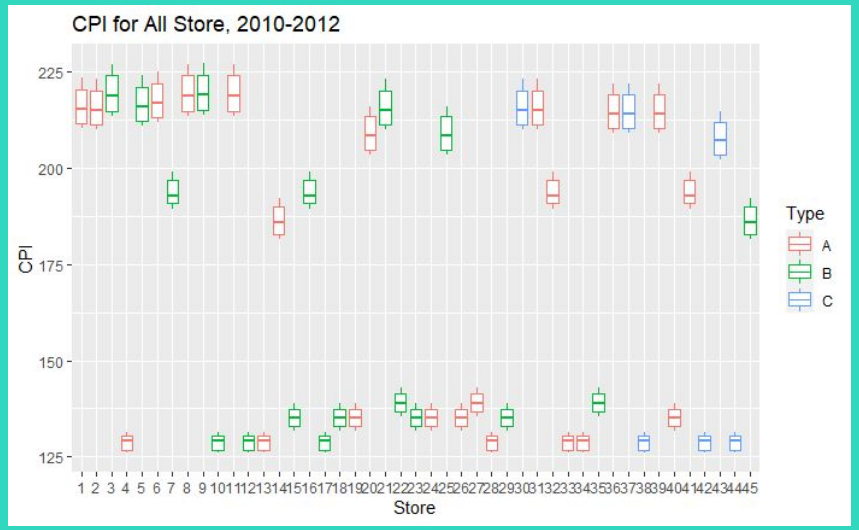
- **Here we evaluate store type in order to determine which type of store we want to focus on**
  - We already knew there are three types of stores ‘A’, ‘B’, and ‘C’
  - We learn from these visualizations that
    - Store type A is the category of large stores with the highest weekly sales
    - Store type B is the category of medium stores with medium weekly sales
    - Store type C is the category of small stores with the lowest weekly sales
  - **From this we determined the ideal store would be type A**





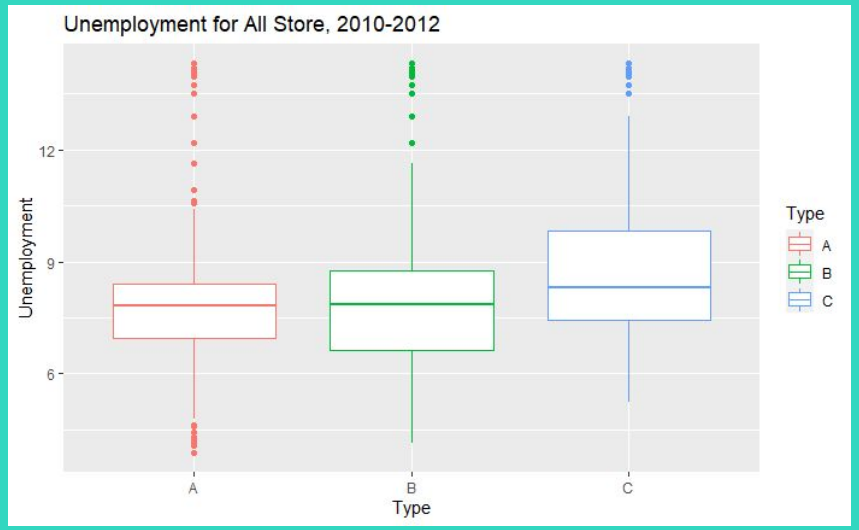
CPI

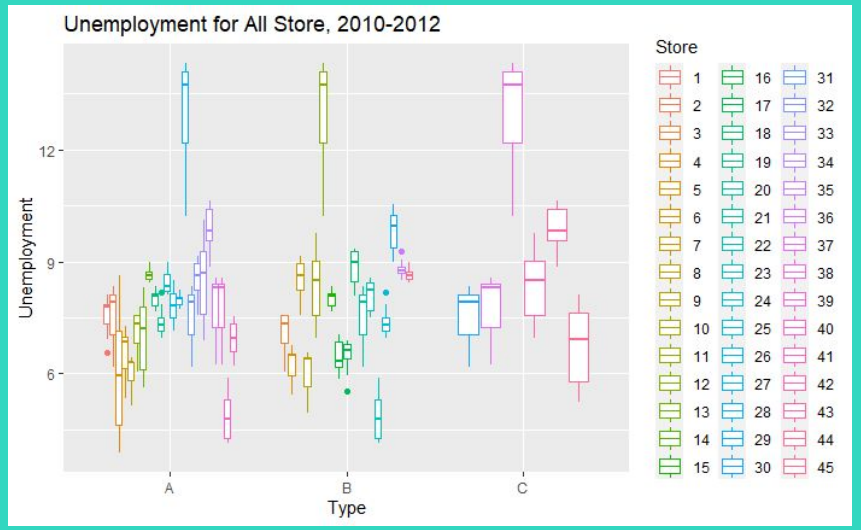
- **Here we evaluate CPI in more detail than the grids**
  - We already know that overall all stores follow the same trend for CPI, increasing overtime with varying ranges
  - Here we see that are clusters that are not dependant on the store type as each type is in each cluster
    - We also see that the median of each CPI by store type differs, type A has the highest median, B the lowest, and C is in between
  - **From this we determined the ideal store would be type C or A**



Unemployment

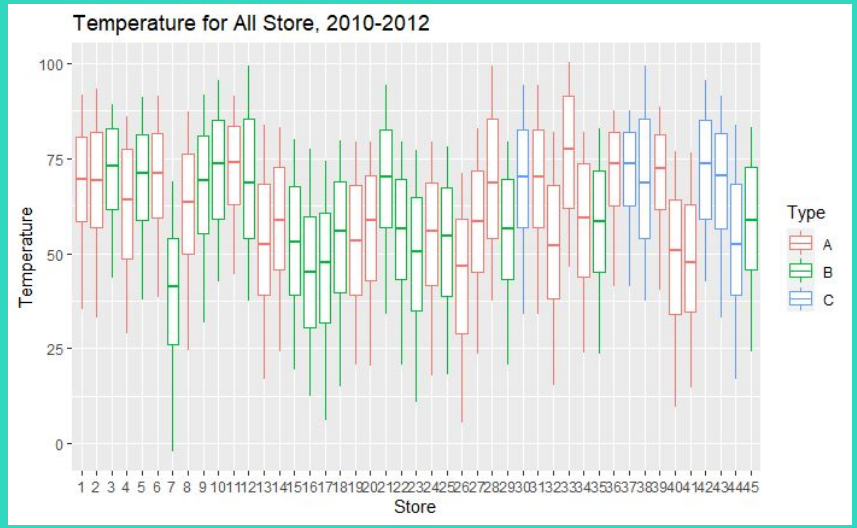
- **Here we evaluate Unemployment in more detail than the grids**
  - We already know that overall unemployment has decreased over time for the stores observed with exceptions
  - Here we see that between store types the unemployment rate has very similar distributions and that there are three stores, 1 in each type, which have higher unemployment rates indicating these stores may be in the same region,
  - **From this we determined the ideal store would be type A or B**





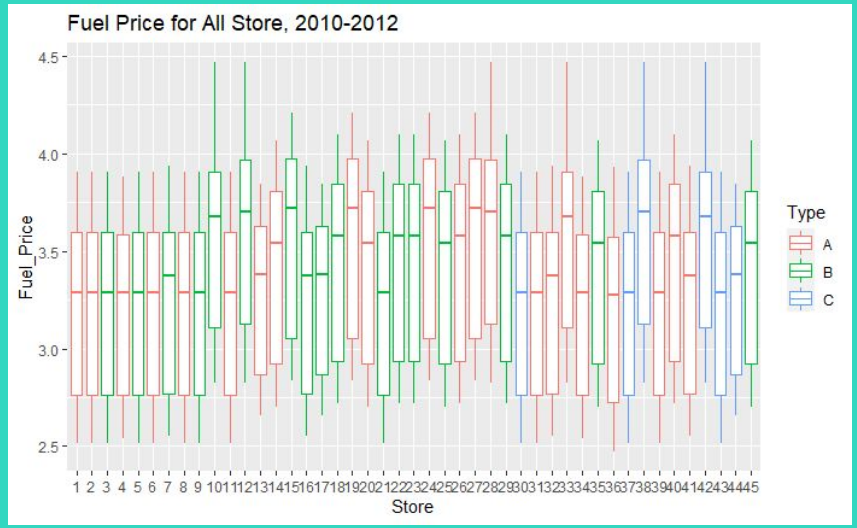
Temperature

- **Here we evaluate temperature in more detail than the grids**
  - We already know that stores follow the same seasons across the year with different ranges
  - Here we can see that store type doesn't seem to strongly correlate to the temperature range of that location
  - **From this we determined the ideal store would be type A or B**



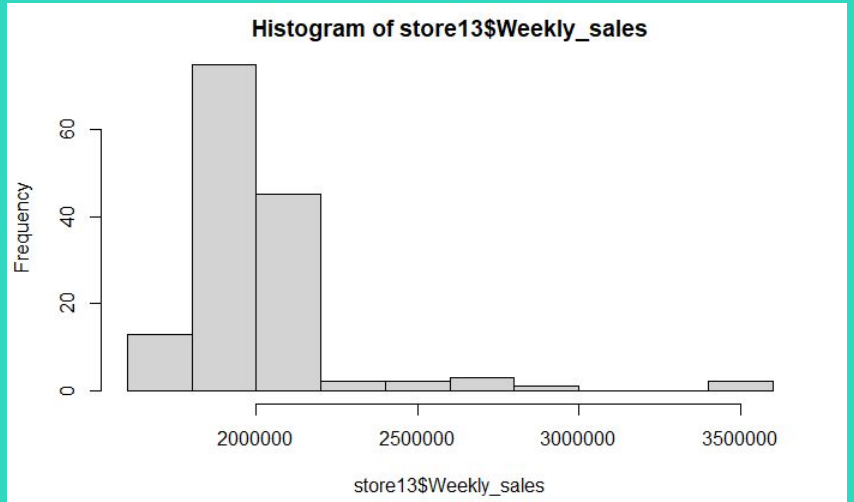
Fuel Price

- **Here we evaluate temperature in more detail than the grids**
  - We already know that fuel prices increases across all stores with varying slopes
  - Here we can see that store type doesn't seem to strongly correlate the fuel price of that location
  - **From this we determined that there was no ideal store type**

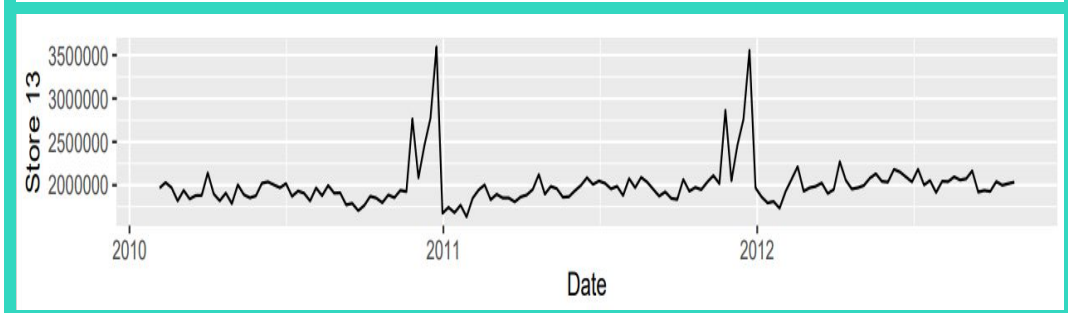


THE IDEAL STORE: STORE 13

- **We determined that store 13 would best represent the set of stores**
  - Store 13 is type A: which the most variables indicated as the ideal store type
    - *This store also follows the trends expressed over time that match the majority of stores seen on the next slide*
    - ***The rest of the variable visual analysis is in the appendix at A.6***



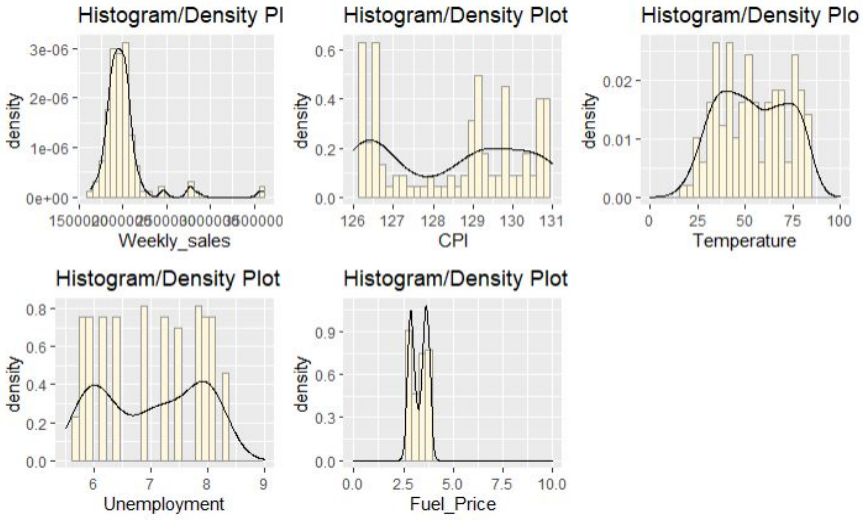

```
     Date                Weekly_sales        Store          CPI         Temperature     Unemployment
Min.   :2010-02-05   Min.   :1633663   13     :143   Min.   :126.1   Min.   :16.94   Min.   :5.621
1st Qu.:2010-10-11   1st Qu.:1877476   1      :  0   1st Qu.:126.6   1st Qu.:38.93   1st Qu.:6.104
Median :2011-06-17   Median :1958824   2      :  0   Median :129.1   Median :52.50   Median :7.193
Mean   :2011-06-17   Mean   :2003620   3      :  0   Mean   :128.7   Mean   :53.70   Mean   :7.002
3rd Qu.:2012-02-20   3rd Qu.:2041469   4      :  0   3rd Qu.:130.5   3rd Qu.:68.34   3rd Qu.:7.795
Max.   :2012-10-26   Max.   :3595903   5      :  0   Max.   :131.2   Max.   :83.68   Max.   :8.316
                                       (Other):  0

  Fuel_Price    IsHoliday          Size          Type
Min.   :2.654   Mode :logical   Min.   :219622   A:143
1st Qu.:2.869   FALSE:133       1st Qu.:219622   B:  0
Median :3.380   TRUE :10        Median :219622   C:  0
Mean   :3.286                   Mean   :219622
3rd Qu.:3.627                   3rd Qu.:219622
Max.   :3.845                   Max.   :219622
```

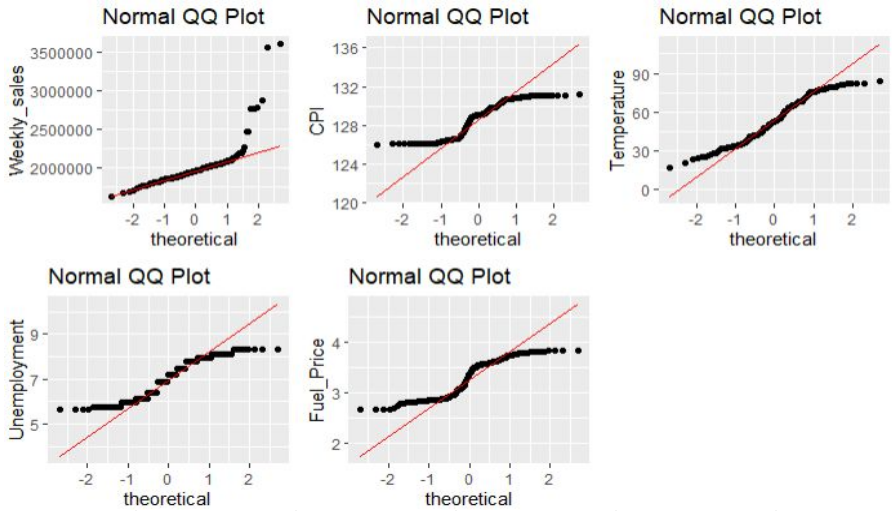


Exploratory Data Analysis

3

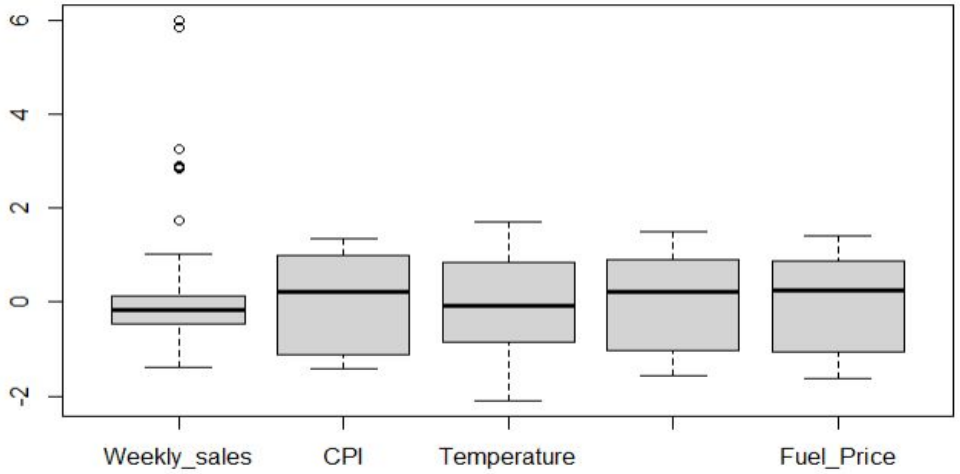
Exploring Distribution of Numeric Variables





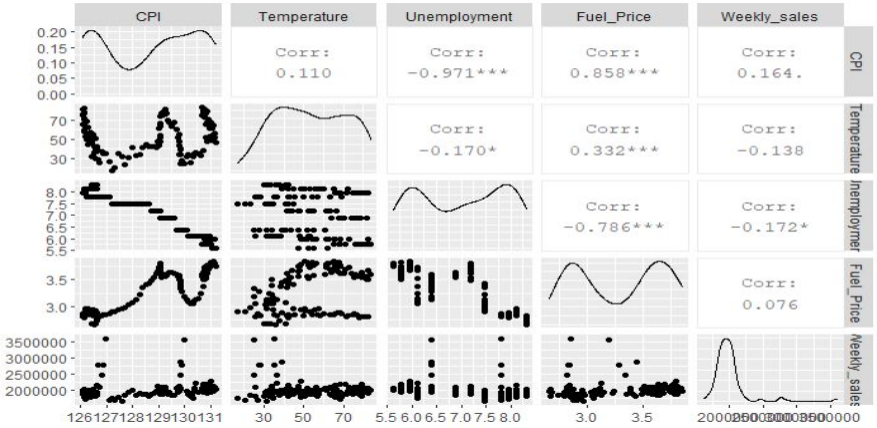
- Histogram/Density Plot and QQ Plot **show distribution is not normal**
- Weekly Sales is skewed right perhaps due to outliers; otherwise normally distributed
- All other variables seem to be bimodal

Further Exploring Distribution Using BoxPlots



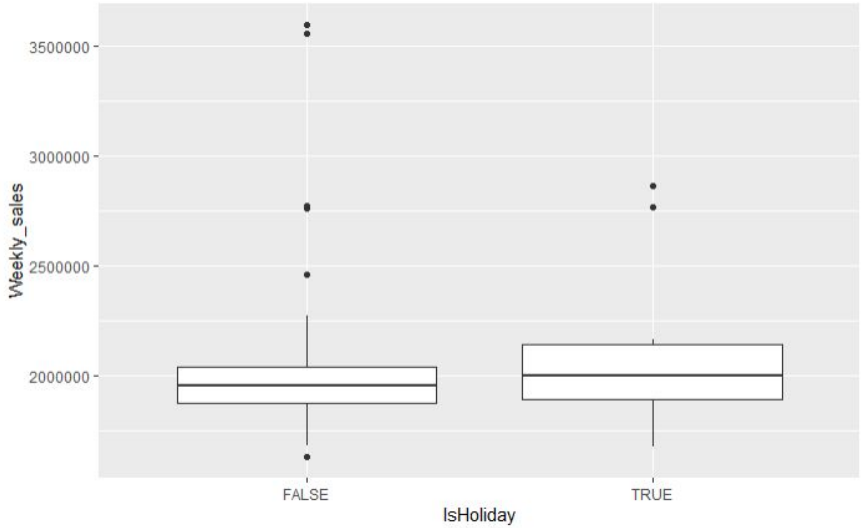
- CPI, Temperature, Unemployment, Fuel Price seem dispersed; Weekly Sales Not Dispersed
- Only **Weekly Sales have outliers**, some seem to be extreme.

Correlations of Numeric Variables

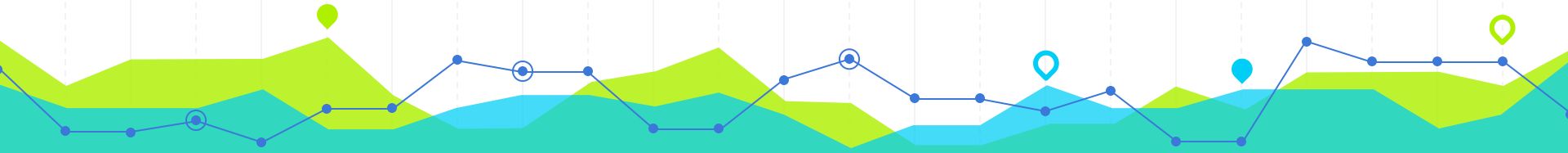


- Predictors for Weekly Sales seems to have weak correlations
- CPI and Fuel Price are positively correlated
- Temperature and Unemployment are negatively correlated

Categorical Data as Predictor



- Being a holiday does have impact on weekly sales
- Extreme outliers for “Not a holiday”



Building Models 4

Linear Model

```
Call:
lm(formula = Weekly_Sales ~ CPI + Temperature + Unemployment +
    Fuel_Price + IsHoliday, data = Store13)

Residuals:
    Min      1Q  Median      3Q     Max
-434625 -114544  -36322   50454 1597757

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    7364492   11067475   0.665    0.507
CPI             -34053      81554  -0.418    0.677
Temperature      -2622       1690  -1.551    0.123
Unemployment   -126573     134738  -0.939    0.349
Fuel_Price       12644     153587   0.082    0.935
IsHolidayTRUE    93772      87268   1.075    0.284

Residual standard error: 260900 on 137 degrees of freedom
Multiple R-squared:  0.06865,   Adjusted R-squared:  0.03466
F-statistic:  2.02 on 5 and 137 DF,  p-value: 0.07961
```

- For our linear model, we included all the main feature variables (excluding markdown).
- The coefficients for the variables show the effects each has on the weekly sales forecast for Store 13.

```
Start:  AIC=3572.79
Weekly_sales ~ CPI + Temperature + Unemployment + Fuel_Price +
    IsHoliday

               Df  Sum of Sq        RSS    AIC
- Fuel_Price    1 4.6119e+08 9.3234e+12 3570.8
- CPI           1 1.1865e+10 9.3348e+12 3571.0
- Unemployment  1 6.0053e+10 9.3830e+12 3571.7
- IsHoliday     1 7.8572e+10 9.4015e+12 3572.0
<none>                      9.3229e+12 3572.8
- Temperature   1 1.6375e+11 9.4867e+12 3573.3

Step:  AIC=3570.8
Weekly_sales ~ CPI + Temperature + Unemployment + IsHoliday

               Df  Sum of Sq        RSS    AIC
- CPI           1 2.1835e+10 9.3452e+12 3569.1
- IsHoliday     1 7.8127e+10 9.4015e+12 3570.0
- Unemployment  1 8.3787e+10 9.4072e+12 3570.1
<none>                      9.3234e+12 3570.8
- Temperature   1 2.5176e+11 9.5751e+12 3572.6

Step:  AIC=3569.13
Weekly_sales ~ Temperature + Unemployment + IsHoliday

               Df  Sum of Sq        RSS    AIC
- IsHoliday     1 8.3141e+10 9.4283e+12 3568.4
<none>                      9.3452e+12 3569.1
- Temperature   1 2.3082e+11 9.5760e+12 3570.6
- Unemployment  1 3.9102e+11 9.7362e+12 3573.0

Step:  AIC=3568.4
Weekly_sales ~ Temperature + Unemployment

               Df  Sum of Sq        RSS    AIC
<none>                      9.4283e+12 3568.4
- Temperature   1 2.8692e+11 9.7153e+12 3570.7
- Unemployment  1 3.9186e+11 9.8202e+12 3572.2

Call:
lm(formula = Weekly_sales ~ Temperature + Unemployment, data = Store13)

Coefficients:
 (Intercept)   Temperature  Unemployment
     2566245         -2593        -60469
```

Stepwise Algorithm

- Based on the stepwise algorithm, we should eliminate CPI, Fuel Price, and Holiday as predictor variables.
- Therefore, an optimal model will simply use temperature and unemployment data to forecast weekly sales.

Comparison of Small and Big Models

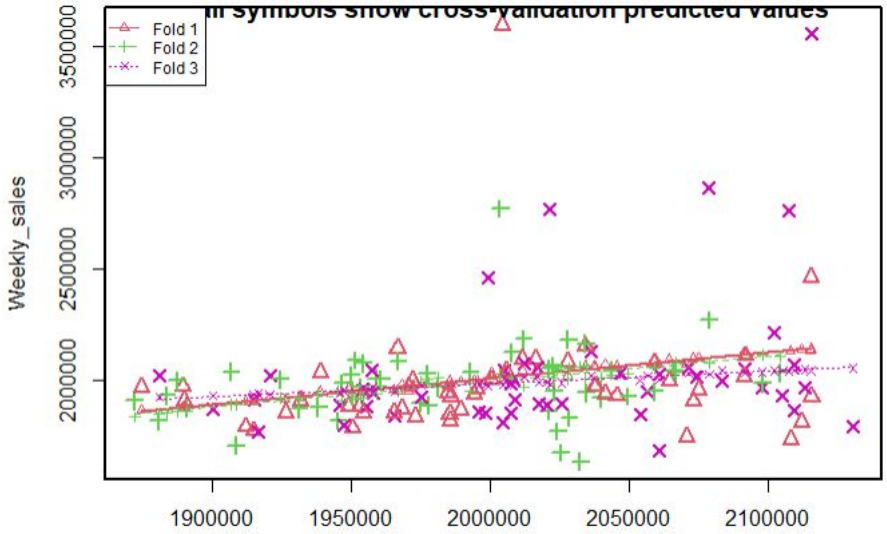
```
Analysis of Variance Table

Model 1: Weekly_sales ~ Temperature + Unemployment
Model 2: Weekly_sales ~ CPI + Temperature + Unemployment + Fuel_Price +
    IsHoliday
  Res.Df        RSS Df  Sum of Sq      F Pr(>F)
1    140 9.4283e+12
2    137 9.3229e+12  3 1.0544e+11 0.5165 0.6716
```

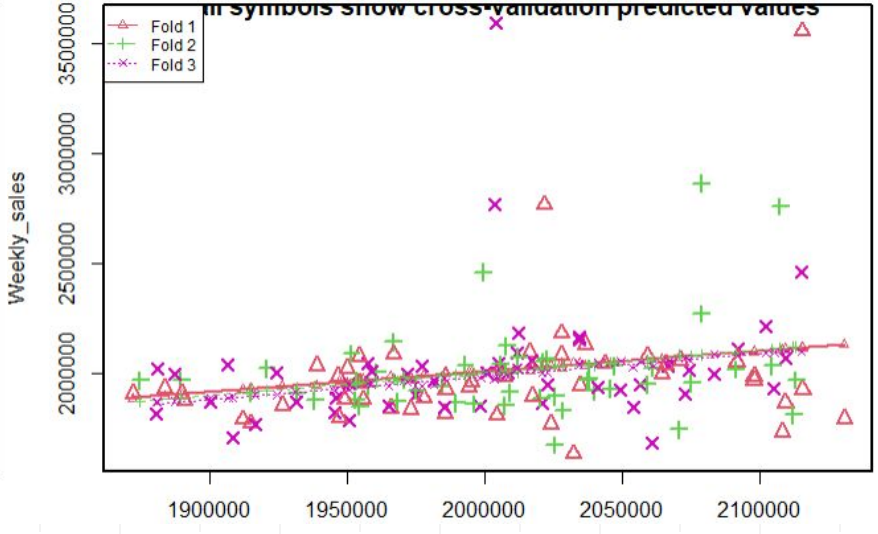
Based on our p-value, which is 0.67, we accept the null hypothesis (small model).

- For the anova table we compared our larger model (with all the variables) to our smaller model (with just temperature and unemployment as predictors).
- Null Hypothesis, H0: b1 = b4 = b5 = 0 (CPI, Fuel Price, Holiday do not predict weekly sales)
- Alternate Hypothesis, H1: b1 or b4 or b5 does not equal 0 (CPI, Fuel Price, Holiday DO predict weekly sales)
- Decision Rule: if p-value < alpha, you reject the null hypothesis, H0. If p=value > alpha, accept the null hypothesis, H0

Model Validation 5

Stability of Model

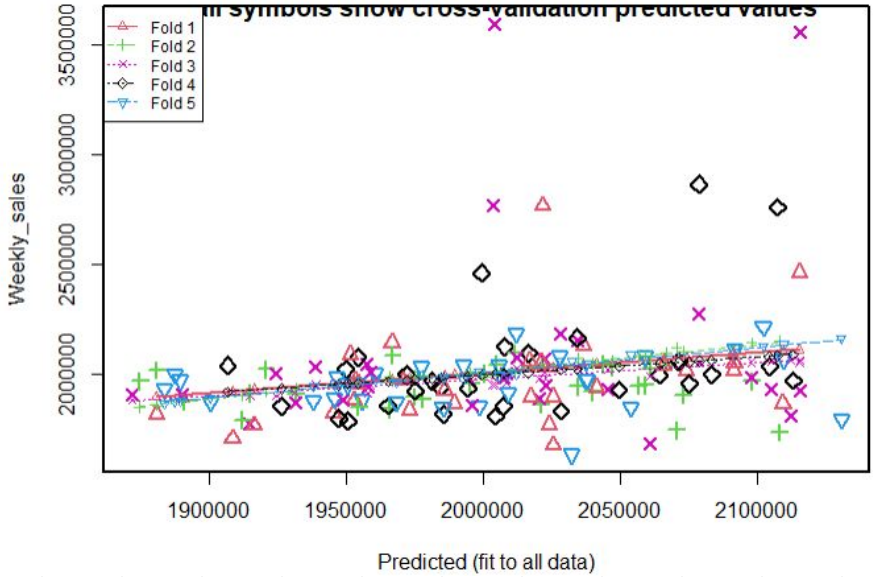




Small model: Weekly_sales~Temperature+Unemployment
Shows instability. But that can just be a result of bad set of folds.

Running seed for random folds show more stability, therefore, we can try with different number of folds.

Optimal Number of Folds



At 5 folds we start to see the lines converge and therefore, we will use this number of folds when comparing mse.

Which Model is Better?

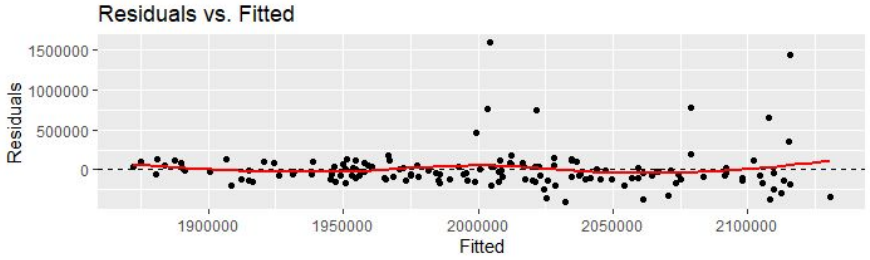
mse.g1 <dbl>	mse.g2 <dbl>
74778332391	69091630717

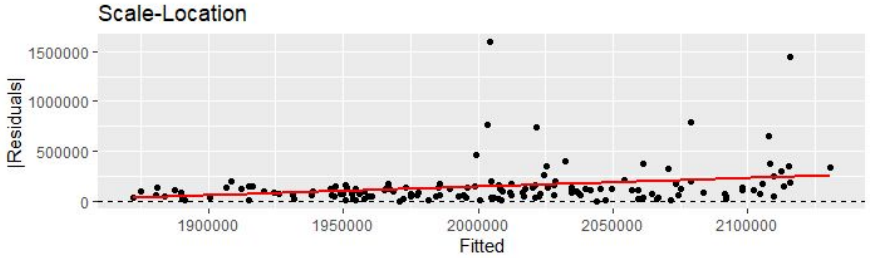
Big Small: Weekly_sales~CPI+Temperature+Unemployment+Fuel_Price+IsHoliday is g1
Small model: Weekly_sales~Temperature+Unemployment is g2
We compared the MSE for g1 and g2 using 5 folds. MSE for g2 is lower showing that it's the better model. Consistent with Anova .

Challenges to OLS

6

Checking for Nonconstant Error Variance

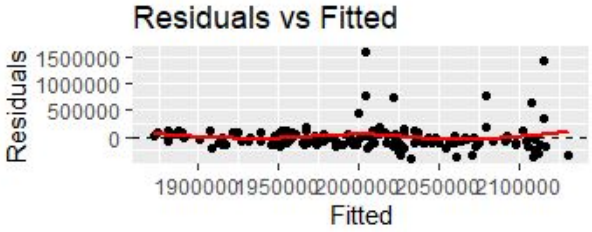


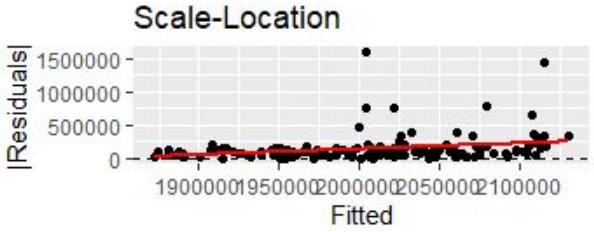


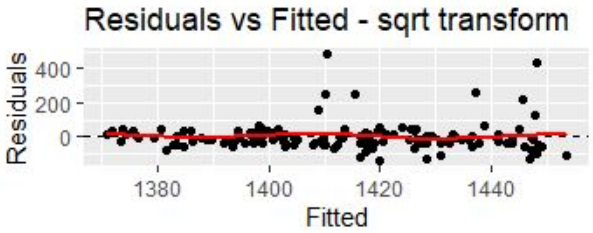

```
Call:
lm(formula = abs(residuals(g.sm)) ~ fitted(g.sm))

Residuals:
    Min      1Q  Median      3Q     Max
-201442  -91038  -33727   20295 1446347

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.587e+06  5.425e+05  -2.926  0.00401 **
fitted(g.sm)  8.643e-01  2.706e-01   3.193  0.00173 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 206400 on 141 degrees of freedom
Multiple R-squared:  0.06745,   Adjusted R-squared:  0.06084
F-statistic:  10.2 on 1 and 141 DF,  p-value: 0.001734
```

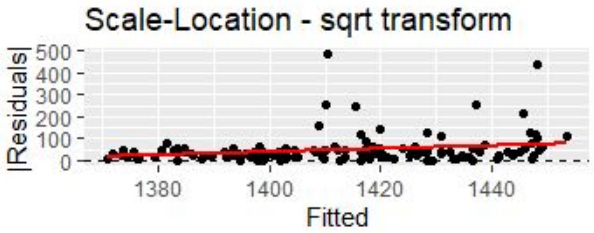
- Residuals scattered more closely around 1900000 than around 2100000 - visual evidence of heteroskedasticity
- Null hypothesis: Slope is 0
- Alternate hypothesis: Slope is not 0
- P-value < alpha = 0.01 (1%) → Reject null hypothesis that slope is 0.
- **Model violates first assumption for OLS**

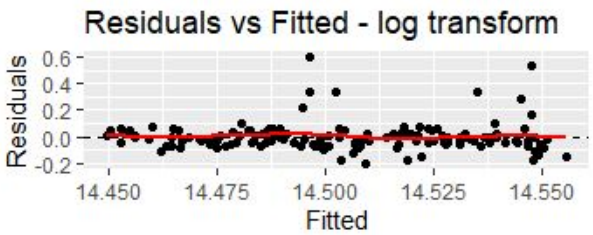
Applying Transforms to Model

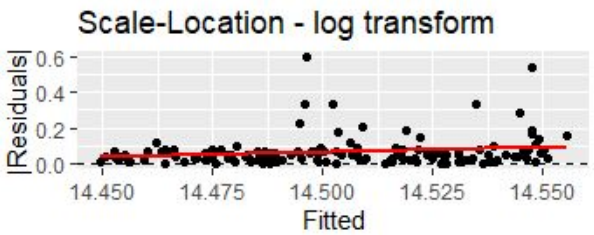












- Applied a square root and log transform to our small model
- Visually, it appears that we still see some heteroskedasticity even in the transformed models

Applying Transforms to Model

Square Root Transformation

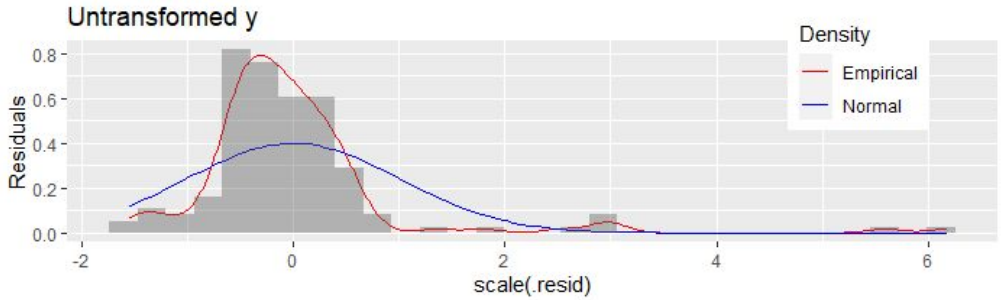
```
Call:
lm(formula = abs(residuals(gs.sm)) ~ fitted(gs.sm))

Residuals:
   Min     1Q Median     3Q    Max
-63.85 -30.60 -12.73   7.00 439.58

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -957.4444   369.6813  -2.590  0.01061 *
fitted(gs.sm)    0.7117     0.2616   2.721  0.00734 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 65.89 on 141 degrees of freedom
Multiple R-squared:  0.04987,   Adjusted R-squared:  0.04313
F-statistic: 7.401 on 1 and 141 DF,  p-value: 0.007339
```

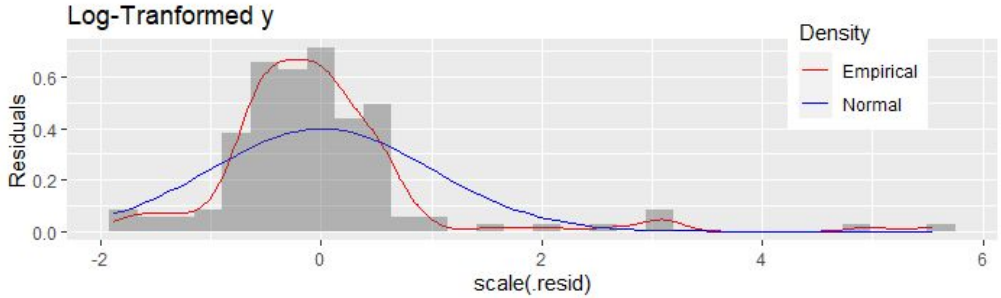
Log Transformation

```
Call:
lm(formula = abs(residuals(glog.sm)) ~ fitted(glog.sm))

Residuals:
     Min       1Q   Median       3Q      Max
-0.08377 -0.04280 -0.01839  0.00864  0.53792

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     -7.8708     3.6491  -2.157   0.0327 *
fitted(glog.sm)  0.5472     0.2516   2.175   0.0313 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.08513 on 141 degrees of freedom
Multiple R-squared:  0.03245,   Adjusted R-squared:  0.02559
F-statistic: 4.729 on 1 and 141 DF,  p-value: 0.03132
```

- At the 1% significance level, the **log transformation model eliminates non-constant variance**! We can use this moving forward.

Checking for non-normal errors



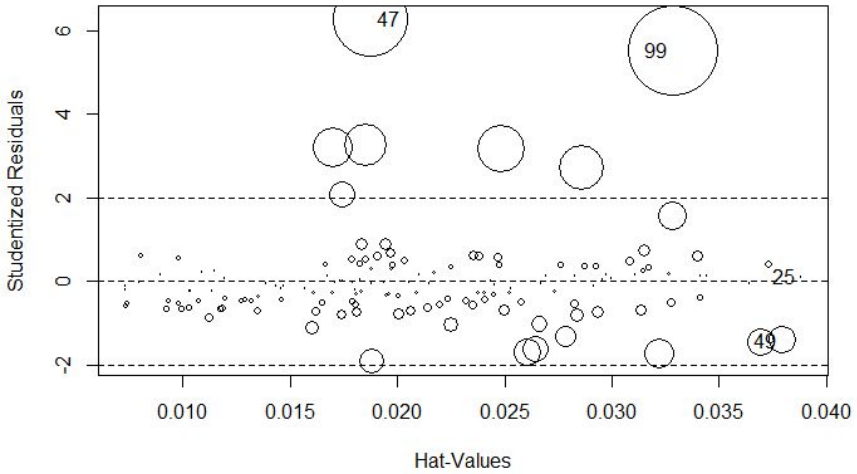


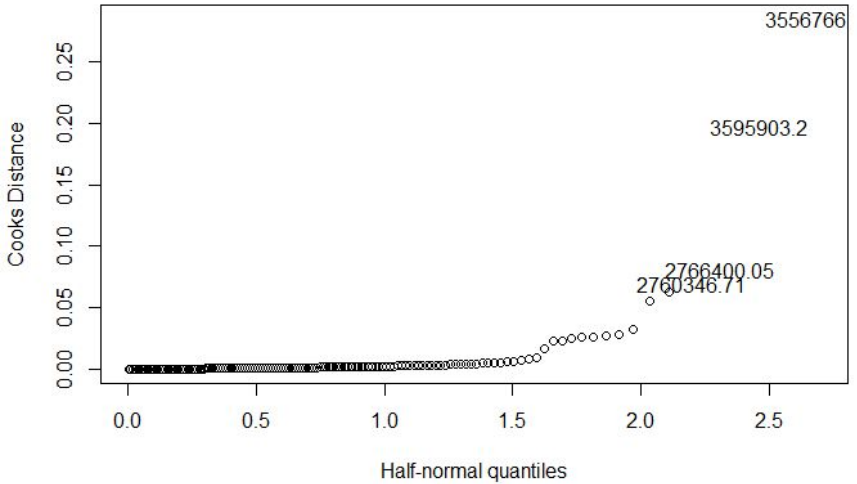

```
        Shapiro-Wilk normality test

data:  residuals(g.sm)
W = 0.67096, p-value < 2.2e-16
```

```
        Shapiro-Wilk normality test

data:  residuals(glog.sm)
W = 0.7537, p-value = 3.303e-14
```

- Transformed model looks better than untransformed model
- Null hypothesis: Model is normal
- Alternate hypothesis: Model is not normal
- Shapiro-Wilk Test: p-value < alpha (1%) → must reject normality for both models
- Box Cox transform yielded better results, but still not normal → may require WLS or other method due to non-normality

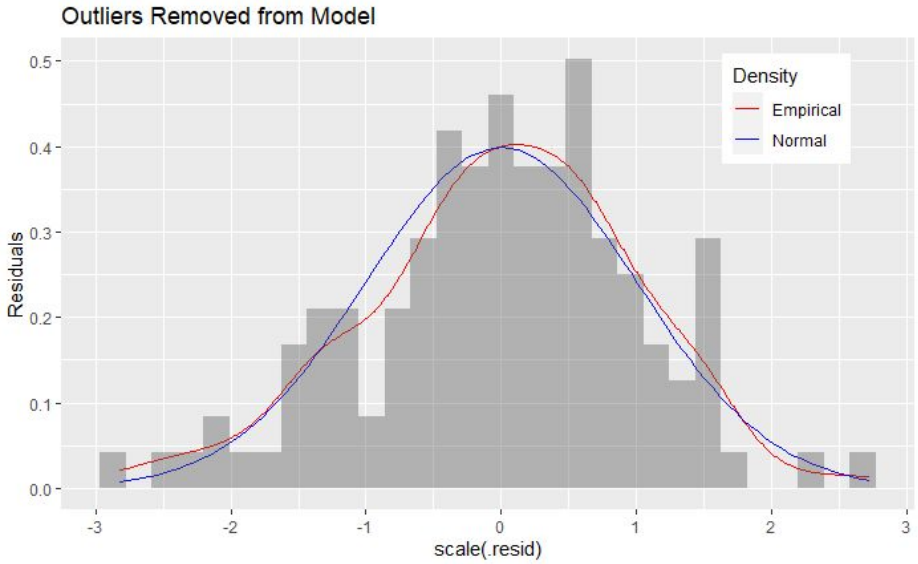
Checking for influential outliers





- We can see there are several influential outliers in the data
- Datapoints 25, 47, 49, and 99 are particularly large in size (refer to the weekly sales points in the Cooks Distance chart)
- These reflect outliers in the “Weekly Sales” that are influencing the model - leaving these data points out would change the values of the estimates in the model

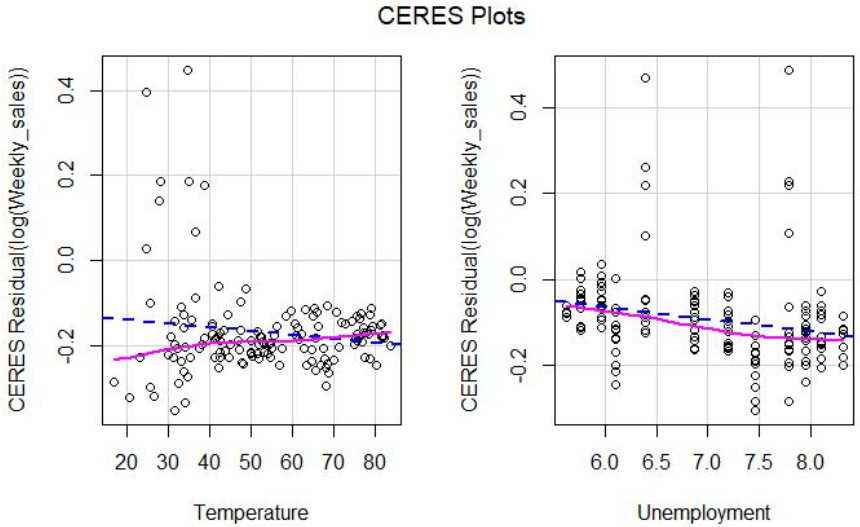
Remove the Outliers



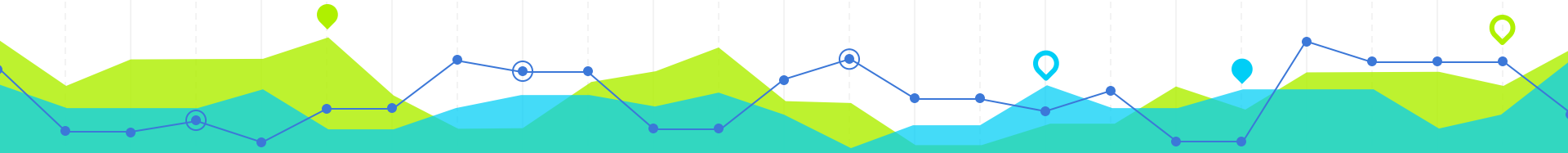

```
        Shapiro-Wilk normality test

data:  residuals(g.sm.clean)
W = 0.99217, p-value = 0.7118
```

- Removed Weekly Sales > 2,100,000 from dataset
- Null hypothesis: Model is normal
- Alternate hypothesis: Model is not normal
- Shapiro-Wilk Test: p-value > alpha (1%) → **accept normality for the small model with outliers removed from the dataset**

Checking for Correct Model Specification

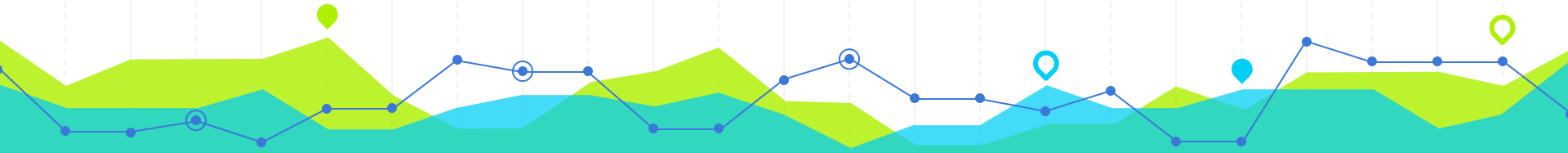


- The pink line is close to the straight line for both predictor variables → no evidence for transformation or higher order terms in the model



Conclusion 7

Weekly Sales ~ Temperature + Unemployment (small model) is the best model for this dataset, so long as the outliers are removed and the log transform is applied to eliminate non-constant error variance



Appendix 8

A.1 SUMMARY OF ORIGINAL DATASET

```r
summary(data)
```

```
     Store           Type                Size             Date                        IsHoliday
 Min.   : 1.0   Length:421570      Min.   :  34875   Min.   :2010-02-05 00:00:00   Mode :logical
 1st Qu.:11.0   Class :character   1st Qu.:  93638   1st Qu.:2010-10-08 00:00:00   FALSE:391909
 Median :22.0   Mode  :character   Median :140167    Median :2011-06-17 00:00:00   TRUE :29661
 Mean   :22.2                      Mean   :136728    Mean   :2011-06-18 08:30:31
 3rd Qu.:33.0                      3rd Qu.:202505    3rd Qu.:2012-02-24 00:00:00
 Max.   :45.0                      Max.   :219622    Max.   :2012-10-26 00:00:00
  Temperature      Fuel_Price      MarkDown1           MarkDown2           MarkDown3
 Min.   : -2.06   Min.   :2.472   Length:421570      Length:421570      Length:421570
 1st Qu.: 46.68   1st Qu.:2.933   Class :character   Class :character   Class :character
 Median : 62.09   Median :3.452   Mode  :character   Mode  :character   Mode  :character
 Mean   : 60.09   Mean   :3.361
 3rd Qu.: 74.28   3rd Qu.:3.738
 Max.   :100.14   Max.   :4.468
  MarkDown4          MarkDown5             CPI             Unemployment          Dept
 Length:421570      Length:421570      Length:421570      Length:421570      Min.   : 1.00
 Class :character   Class :character   Class :character   Class :character   store1total45
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Median :37.00
                                                                             Mean   :44.26
                                                                             3rd Qu.:74.00
                                                                             Max.   :99.00

  Weekly_Sales
 Min.   : -4989
 1st Qu.:  2080
 Median :  7612
 Mean   : 15981
 3rd Qu.: 20206
 Max.   :693099
```

A.2 DATA TYPES

```
str(data)
```

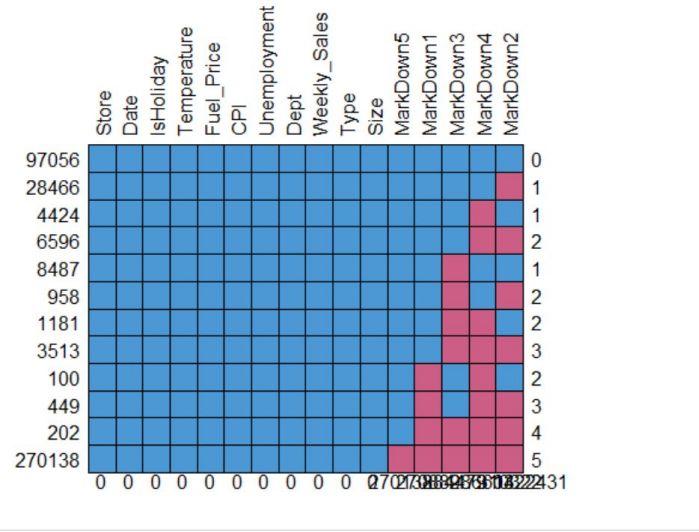
BEFORE

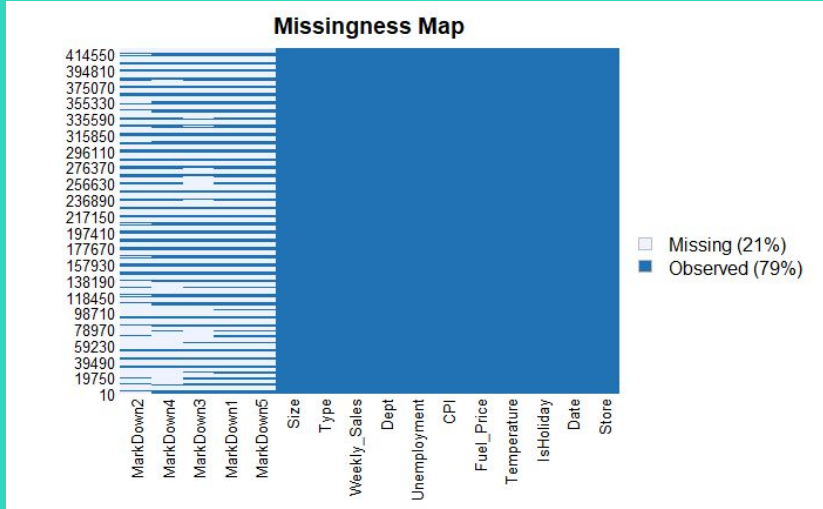
```
'data.frame':   421570 obs. of  16 variables:
 $ Store       : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Type        : chr  "A" "A" "A" "A" ...
 $ Size        : num  151315 151315 151315 151315 151315 ...
 $ Date        : POSIXct, format: "2010-02-05" "2010-02-05" "2010
 $ IsHoliday   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ Temperature : num  42.3 42.3 42.3 42.3 42.3 ...
 $ Fuel_Price  : num  2.57 2.57 2.57 2.57 2.57 ...
 $ MarkDown1   : chr  "NA" "NA" "NA" "NA" ...
 $ MarkDown2   : chr  "NA" "NA" "NA" "NA" ...
 $ MarkDown3   : chr  "NA" "NA" "NA" "NA" ...
 $ MarkDown4   : chr  "NA" "NA" "NA" "NA" ...
 $ MarkDown5   : chr  "NA" "NA" "NA" "NA" ...
 $ CPI         : chr  "211.0963582" "211.0963582" "211.0963582" "
 $ Unemployment: chr  "8.1059999999999999" "8.1059999999999999" "
...
 $ Dept        : num  1 26 17 45 28 79 55 5 58 7 ...
 $ Weekly_Sales: num  24924.5 11737.1 13223.8 37.4 1085.3 ...
```

AFTER

```
'data.frame':   421570 obs. of  16 variables:
 $ Store       : Factor w/ 45 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1
 $ Type        : Factor w/ 3 levels "A","B","C": 1 1 1 1 1 1 1 1 1 ...
 $ Size        : num  151315 151315 151315 151315 151315 ...
 $ Date        : POSIXct, format: "2010-02-05" "2010-02-05" "2010-02-05"
 $ IsHoliday   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ Temperature : num  42.3 42.3 42.3 42.3 42.3 ...
 $ Fuel_Price  : num  2.57 2.57 2.57 2.57 2.57 ...
 $ MarkDown1   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ MarkDown2   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ MarkDown3   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ MarkDown4   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ MarkDown5   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ CPI         : num  211 211 211 211 211 ...
 $ Unemployment: num  8.11 8.11 8.11 8.11 8.11 ...
 $ Dept        : Factor w/ 81 levels "1","2","3","4",..: 1 25 16 44 27 65
 $ Weekly_Sales: num  24924.5 11737.1 13223.8 37.4 1085.3 ...
```

A.3 MISSING DATA BEFORE






```r
install.packages("mice", repos='http://cran.rstudio.com')
library(mice)
md.pattern(data, rotate.names = TRUE)
```

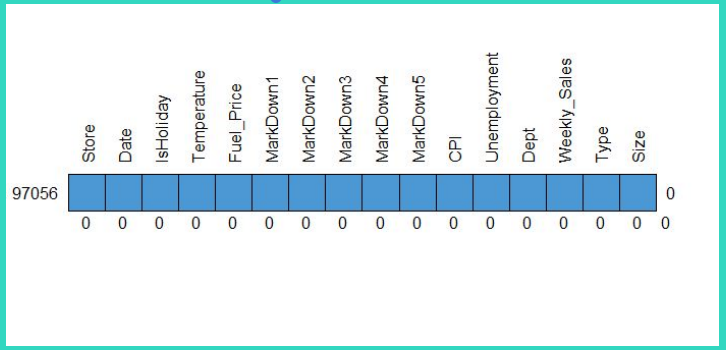
```r
library(Amelia)
missmap(data)
```

```
      Store        Date   IsHoliday Temperature  Fuel_Price   MarkDown1   MarkDown2   MarkDown3   MarkDown4
          0           0           0           0           0      270889      310322      284479      286603
 MarkDown5         CPI Unemployment        Dept Weekly_Sales        Type        Size
     270138           0           0           0           0           0           0
```

AFTER

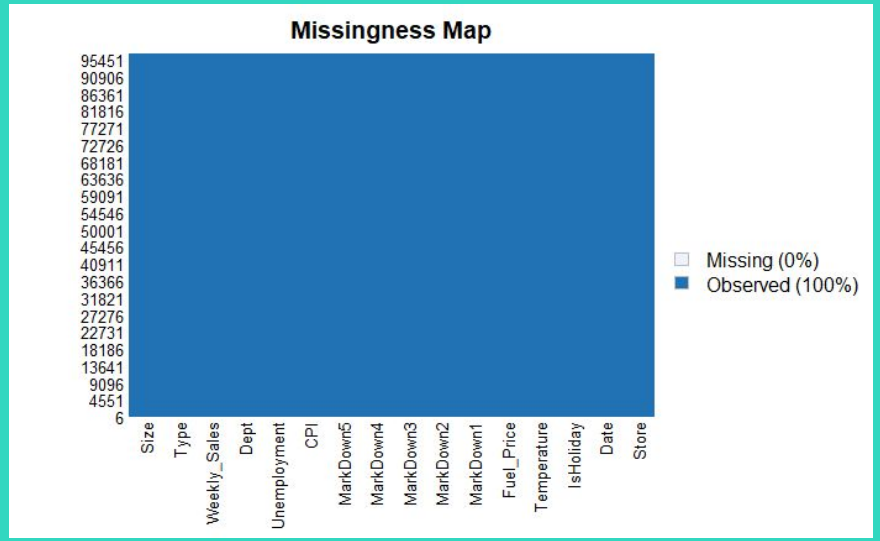
A.3 MISSING DATA

Dropping records with missing values




```r
install.packages("mice", repos='http://cran.rstudio.com')
library(mice)
md.pattern(data, rotate.names = TRUE)
```

```
 /\     /\
{  `---'  }
{  O   O  }
==>  V <==  No need for mice. This data set is completely observed.
 \  \|/  /
  `-----'

      Store Date IsHoliday Temperature Fuel_Price CPI Unemployment Dept Weekly_Sales
Type Size
421570     1    1         1           1          1   1            1    1            1
1    1 0
           0    0         0           0          0   0            0    0            0
0    0 0
```

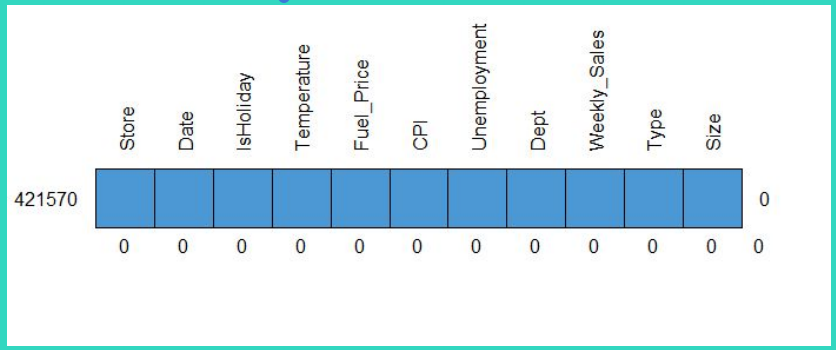



```r
library(Amelia)
missmap(data)
```

AFTER

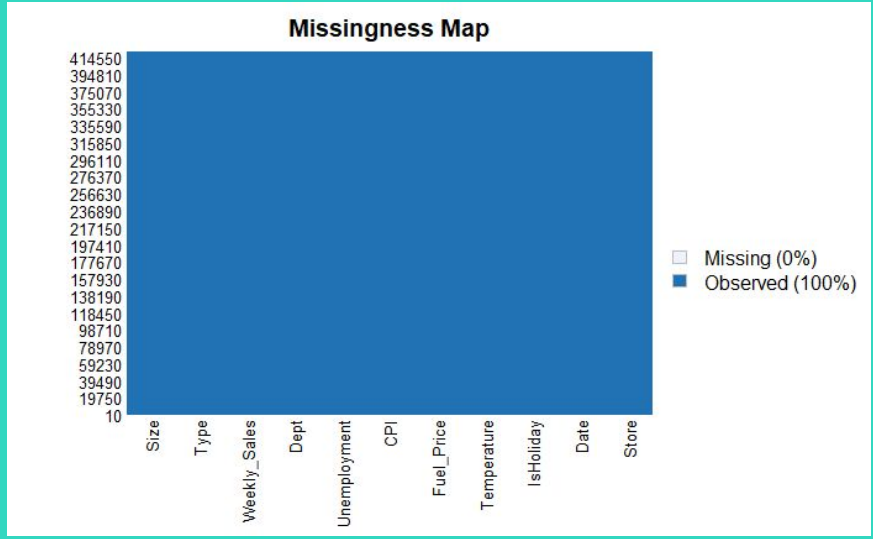
Dropping MarkDown Variables

A.3 MISSING DATA




```{r}
install.packages("mice", repos='http://cran.rstudio.com')
library(mice)
md.pattern(data, rotate.names = TRUE)
```

```
 /\     /\
{  `---'  }
{  O   O  }
==>  V <==  No need for mice. This data set is completely observed.
 \  \|/  /
  `-----'

      Store Date IsHoliday Temperature Fuel_Price CPI Unemployment Dept Weekly_Sales
Type Size
421570     1    1         1           1          1   1            1    1            1
1    1 0
           0    0         0           0          0   0            0    0            0
0    0 0
```




```{r}
library(Amelia)
missmap(data)
```

A.4 CREATING SUBSETS FOR EACH STORE

STEP 1:
Create subsets for each store by

```r
store1 <- data[which(data[,'Store']==1),]
store2 <- data[which(data[,'Store']==2),]
store3 <- data[which(data[,'Store']==3),]
store4 <- data[which(data[,'Store']==4),]
store5 <- data[which(data[,'Store']==5),]
store6 <- data[which(data[,'Store']==6),]
store7 <- data[which(data[,'Store']==7),]
store8 <- data[which(data[,'Store']==8),]
store9 <- data[which(data[,'Store']==9),]
store10 <- data[which(data[,'Store']==10),]

store11 <- data[which(data[,'Store']==11),]
store12 <- data[which(data[,'Store']==12),]
store13 <- data[which(data[,'Store']==13),]
store14 <- data[which(data[,'Store']==14),]
store15 <- data[which(data[,'Store']==15),]
store16 <- data[which(data[,'Store']==16),]
store17 <- data[which(data[,'Store']==17),]
store18 <- data[which(data[,'Store']==18),]
store19 <- data[which(data[,'Store']==19),]
store20 <- data[which(data[,'Store']==20),]

store21 <- data[which(data[,'Store']==21),]
store22 <- data[which(data[,'Store']==22),]
store23 <- data[which(data[,'Store']==23),]
store24 <- data[which(data[,'Store']==24),]
store25 <- data[which(data[,'Store']==25),]
store26 <- data[which(data[,'Store']==26),]
store27 <- data[which(data[,'Store']==27),]
store28 <- data[which(data[,'Store']==28),]
store29 <- data[which(data[,'Store']==29),]

store30 <- data[which(data[,'Store']==30),]
store31 <- data[which(data[,'Store']==31),]
store32 <- data[which(data[,'Store']==32),]
store33 <- data[which(data[,'Store']==33),]
store34 <- data[which(data[,'Store']==34),]
store35 <- data[which(data[,'Store']==35),]
store36 <- data[which(data[,'Store']==36),]
store37 <- data[which(data[,'Store']==37),]
store38 <- data[which(data[,'Store']==38),]
store39 <- data[which(data[,'Store']==39),]

store40 <- data[which(data[,'Store']==40),]
store41 <- data[which(data[,'Store']==41),]
store42 <- data[which(data[,'Store']==42),]
store43 <- data[which(data[,'Store']==43),]
store44 <- data[which(data[,'Store']==44),]
store45 <- data[which(data[,'Store']==45),]
```

STEP 2:
Create subsets for each store which calculated the total weekly sales for 1 observation per week.

The original data had up to 99 observations per store per week because there was one observation per department. Although we dropped the dept variable the multiple observations were still present.

```r
store1total <- aggregate(store1$Weekly_Sales, by=list(store1$Date), sum)
store2total <- aggregate(store2$Weekly_Sales, by=list(store2$Date), sum)
store3total <- aggregate(store3$Weekly_Sales, by=list(store3$Date), sum)
store4total <- aggregate(store4$Weekly_Sales, by=list(store4$Date), sum)
store5total <- aggregate(store5$Weekly_Sales, by=list(store5$Date), sum)
store6total <- aggregate(store6$Weekly_Sales, by=list(store6$Date), sum)
store7total <- aggregate(store7$Weekly_Sales, by=list(store7$Date), sum)
store8total <- aggregate(store8$Weekly_Sales, by=list(store8$Date), sum)
store9total <- aggregate(store9$Weekly_Sales, by=list(store9$Date), sum)


store10total <- aggregate(store10$Weekly_Sales, by=list(store10$Date), sum)
store11total <- aggregate(store11$Weekly_Sales, by=list(store11$Date), sum)
store12total <- aggregate(store12$Weekly_Sales, by=list(store12$Date), sum)
store13total <- aggregate(store13$Weekly_Sales, by=list(store13$Date), sum)
store14total <- aggregate(store14$Weekly_Sales, by=list(store14$Date), sum)
store15total <- aggregate(store15$Weekly_Sales, by=list(store15$Date), sum)
store16total <- aggregate(store16$Weekly_Sales, by=list(store16$Date), sum)
store17total <- aggregate(store17$Weekly_Sales, by=list(store17$Date), sum)
store18total <- aggregate(store18$Weekly_Sales, by=list(store18$Date), sum)
store19total <- aggregate(store19$Weekly_Sales, by=list(store19$Date), sum)

store20total <- aggregate(store20$Weekly_Sales, by=list(store20$Date), sum)
store21total <- aggregate(store21$Weekly_Sales, by=list(store21$Date), sum)
store22total <- aggregate(store22$Weekly_Sales, by=list(store22$Date), sum)
store23total <- aggregate(store23$Weekly_Sales, by=list(store23$Date), sum)
store24total <- aggregate(store24$Weekly_Sales, by=list(store24$Date), sum)
store25total <- aggregate(store25$Weekly_Sales, by=list(store25$Date), sum)
store26total <- aggregate(store26$Weekly_Sales, by=list(store26$Date), sum)
store27total <- aggregate(store27$Weekly_Sales, by=list(store27$Date), sum)
store28total <- aggregate(store28$Weekly_Sales, by=list(store28$Date), sum)
store29total <- aggregate(store29$Weekly_Sales, by=list(store29$Date), sum)

store30total <- aggregate(store30$Weekly_Sales, by=list(store30$Date), sum)
store31total <- aggregate(store31$Weekly_Sales, by=list(store31$Date), sum)
store32total <- aggregate(store32$Weekly_Sales, by=list(store32$Date), sum)
store33total <- aggregate(store33$Weekly_Sales, by=list(store33$Date), sum)
store34total <- aggregate(store34$Weekly_Sales, by=list(store34$Date), sum)
store35total <- aggregate(store35$Weekly_Sales, by=list(store35$Date), sum)
store36total <- aggregate(store36$Weekly_Sales, by=list(store36$Date), sum)
store37total <- aggregate(store37$Weekly_Sales, by=list(store37$Date), sum)
store38total <- aggregate(store38$Weekly_Sales, by=list(store38$Date), sum)
store39total <- aggregate(store39$Weekly_Sales, by=list(store39$Date), sum)

store40total <- aggregate(store40$Weekly_Sales, by=list(store40$Date), sum)
store41total <- aggregate(store41$Weekly_Sales, by=list(store41$Date), sum)
store42total <- aggregate(store42$Weekly_Sales, by=list(store42$Date), sum)
store43total <- aggregate(store43$Weekly_Sales, by=list(store43$Date), sum)
store44total <- aggregate(store44$Weekly_Sales, by=list(store44$Date), sum)
store45total <- aggregate(store45$Weekly_Sales, by=list(store45$Date), sum)
```

A.4 CREATING SUBSETS FOR EACH STORE

STEP 3:
In step 2 we only addressed reducing one variable from multiple weekly observations per store to 1. Here is step 3 we condense all other variables to match. We did so by taking the average of all of these variables.

Each of these variables were the exact same information in each observation for the specific store on that specific date. By taking the average of this data we created a subset which had only one observation per week with the same data that was originally captured.

```
store1total1 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday +Size + Type, data=store1,
mean)
store1total2 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store2,
mean)
store1total3 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store3,
mean)
store1total4 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store4,
mean)
store1total5 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store5,
mean)
store1total6 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store6,
mean)
store1total7 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store7,
mean)
store1total8 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store8,
mean)
store1total9 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store9,
mean)

store1total10 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store10,
mean)
store1total11 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store11,
mean)
store1total12 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store12,
mean)
store1total13 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store13,
mean)
store1total14 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store14,
mean)
store1total15 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store15,
mean)
store1total16 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store16,
mean)
store1total17 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store17,
mean)
store1total18 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store18,
mean)
store1total19 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store19,
mean)

store1total20 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store20,
mean)
store1total21 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store21,
mean)
store1total22 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store22,
mean)
store1total23 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store23,
mean)
store1total24 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store24,
mean)
store1total25 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store25,
mean)
store1total26 <- aggregate(. ~ Date + Store+ CPI + Temperature + Unemployment + Fuel_Price +IsHoliday+Size + Type, data=store26,
```

A.4 CREATING SUBSETS FOR EACH STORE

STEP 4:
Here we create another subset which drops the original observations for Dept and Weekly Sale.

We do this in order to prevent duplicate variables when merge the data to create the finished dataset.

```
store1total1 <- store1total1[ -c(10:11) ]
store1total2 <- store1total2[ -c(10:11) ]
store1total3 <- store1total3[  -c(10:11) ]
store1total4 <- store1total4[ -c(10:11) ]
store1total5 <- store1total5[ -c(10:11) ]
store1total6 <- store1total6[ -c(10:11) ]
store1total7 <- store1total7[ -c(10:11) ]
store1total8 <- store1total8[ -c(10:11) ]
store1total9 <- store1total9[  -c(10:11) ]

store1total10 <- store1total10[  -c(10:11) ]
store1total11 <- store1total11[  -c(10:11) ]
store1total12 <- store1total12[  -c(10:11) ]
store1total13 <- store1total13[  -c(10:11) ]
store1total14 <- store1total14[ -c(10:11) ]
store1total15 <- store1total15[  -c(10:11) ]
store1total16 <- store1total16[ -c(10:11) ]
store1total17 <- store1total17[  -c(10:11) ]
store1total18 <- store1total18[  -c(10:11) ]
store1total19 <- store1total19[ -c(10:11) ]

store1total20 <- store1total20[  -c(10:11) ]
store1total21 <- store1total21[  -c(10:11) ]
store1total22 <- store1total22[  -c(10:11) ]
store1total23 <- store1total23[ -c(10:11) ]
store1total24 <- store1total24[  -c(10:11) ]
store1total25 <- store1total25[  -c(10:11) ]
store1total26 <- store1total26[  -c(10:11) ]
store1total27 <- store1total27[  -c(10:11) ]
store1total28 <- store1total28[ -c(10:11) ]
store1total29 <- store1total29[  -c(10:11) ]

store1total30 <- store1total30[  -c(10:11) ]
store1total31 <- store1total31[ -c(10:11) ]
store1total32 <- store1total32[  -c(10:11) ]
store1total33 <- store1total33[  -c(10:11) ]
store1total34 <- store1total34[  -c(10:11) ]
store1total35 <- store1total35[  -c(10:11) ]
store1total36 <- store1total36[  -c(10:11) ]
store1total37 <- store1total37[  -c(10:11) ]
store1total38 <- store1total38[  -c(10:11) ]
store1total39 <- store1total39[  -c(10:11) ]

store1total40 <- store1total40[  -c(10:11) ]
store1total41 <- store1total41[  -c(10:11) ]
store1total42 <- store1total42[ -c(10:11) ]
store1total43 <- store1total43[ -c(10:11) ]
store1total44 <- store1total44[ -c(10:11) ]
```

STEP 5:
Here we merge two of the subsets we created. We merge by the date because each of the subsets is only for one store we do not need to worry about merging by store as we would have if we were merging the entire original dataset.

```
store1 <- merge(store1total, store1total1, by.x='Group.1', by.y='Date')
store2 <- merge(store2total, store1total2, by.x='Group.1', by.y='Date')
store3 <- merge(store3total, store1total3, by.x='Group.1', by.y='Date')
store4 <- merge(store4total, store1total4, by.x='Group.1', by.y='Date')
store5 <- merge(store5total, store1total5, by.x='Group.1', by.y='Date')
store6 <- merge(store6total, store1total6, by.x='Group.1', by.y='Date')
store7 <- merge(store7total, store1total7, by.x='Group.1', by.y='Date')
store8 <- merge(store8total, store1total8, by.x='Group.1', by.y='Date')
store9 <- merge(store9total, store1total9, by.x='Group.1', by.y='Date')


store10 <- merge(store10total, store1total10, by.x='Group.1', by.y='Date')
store11 <- merge(store11total, store1total11, by.x='Group.1', by.y='Date')
store12 <- merge(store12total, store1total12, by.x='Group.1', by.y='Date')
store13 <- merge(store13total, store1total13, by.x='Group.1', by.y='Date')
store14 <- merge(store14total, store1total14, by.x='Group.1', by.y='Date')
store15 <- merge(store15total, store1total15, by.x='Group.1', by.y='Date')
store16 <- merge(store16total, store1total16, by.x='Group.1', by.y='Date')
store17 <- merge(store17total, store1total17, by.x='Group.1', by.y='Date')
store18 <- merge(store18total, store1total18, by.x='Group.1', by.y='Date')
store19 <- merge(store19total, store1total19, by.x='Group.1', by.y='Date')

store20 <- merge(store20total, store1total20, by.x='Group.1', by.y='Date')
store21 <- merge(store21total, store1total21, by.x='Group.1', by.y='Date')
store22 <- merge(store22total, store1total22, by.x='Group.1', by.y='Date')
store23 <- merge(store23total, store1total23, by.x='Group.1', by.y='Date')
store24 <- merge(store24total, store1total24, by.x='Group.1', by.y='Date')
store25 <- merge(store25total, store1total25, by.x='Group.1', by.y='Date')
store26 <- merge(store26total, store1total26, by.x='Group.1', by.y='Date')
store27 <- merge(store27total, store1total27, by.x='Group.1', by.y='Date')
store28 <- merge(store28total, store1total28, by.x='Group.1', by.y='Date')
store29 <- merge(store29total, store1total29, by.x='Group.1', by.y='Date')

store30 <- merge(store30total, store1total30, by.x='Group.1', by.y='Date')
store31 <- merge(store31total, store1total31, by.x='Group.1', by.y='Date')
store32 <- merge(store32total, store1total32, by.x='Group.1', by.y='Date')
store33 <- merge(store33total, store1total33, by.x='Group.1', by.y='Date')
store34 <- merge(store34total, store1total34, by.x='Group.1', by.y='Date')
store35 <- merge(store35total, store1total35, by.x='Group.1', by.y='Date')
store36 <- merge(store36total, store1total36, by.x='Group.1', by.y='Date')
store37 <- merge(store37total, store1total37, by.x='Group.1', by.y='Date')
store38 <- merge(store38total, store1total38, by.x='Group.1', by.y='Date')
store39 <- merge(store39total, store1total39, by.x='Group.1', by.y='Date')

store40 <- merge(store40total, store1total40, by.x='Group.1', by.y='Date')
store41 <- merge(store41total, store1total41, by.x='Group.1', by.y='Date')
store42 <- merge(store42total, store1total42, by.x='Group.1', by.y='Date')
store43 <- merge(store43total, store1total43, by.x='Group.1', by.y='Date')
store44 <- merge(store44total, store1total44, by.x='Group.1', by.y='Date')
store45 <- merge(store45total, store1total45, by.x='Group.1', by.y='Date')
```

A.4 CREATING SUBSETS FOR EACH STORE

STEP 6:
Here we rename the newly created variable which was called ‘x’.

We changed it to Weekly_sales.

```
library(plyr)
library(dplyr)
store1 <- store1 %>% rename('Weekly_sales'='x')
store1 <- store1 %>% rename('Date'='Group.1')
store2 <- store2 %>% rename('Weekly_sales'='x')
store2 <- store2 %>% rename('Date'='Group.1')
store3 <- store3 %>% rename('Weekly_sales'='x')
store3 <- store3 %>% rename('Date'='Group.1')
store4 <- store4 %>% rename('Weekly_sales'='x')
store4 <- store4 %>% rename('Date'='Group.1')
store5 <- store5 %>% rename('Weekly_sales'='x')
store5 <- store5 %>% rename('Date'='Group.1')
store6 <- store6 %>% rename('Weekly_sales'='x')
store6 <- store6 %>% rename('Date'='Group.1')
store7 <- store7 %>% rename('Weekly_sales'='x')
store7 <- store7 %>% rename('Date'='Group.1')
store8 <- store8 %>% rename('Weekly_sales'='x')
store8 <- store8 %>% rename('Date'='Group.1')
store9 <- store9 %>% rename('Weekly_sales'='x')
store9 <- store9 %>% rename('Date'='Group.1')

store10 <- store10 %>% rename('Weekly_sales'='x')
store10 <- store10 %>% rename('Date'='Group.1')
store11 <- store11 %>% rename('Weekly_sales'='x')
store11 <- store11 %>% rename('Date'='Group.1')
store12 <- store12 %>% rename('Weekly_sales'='x')
store12 <- store12 %>% rename('Date'='Group.1')
store13 <- store13 %>% rename('Weekly_sales'='x')
store13 <- store13 %>% rename('Date'='Group.1')
store14 <- store14 %>% rename('Weekly_sales'='x')
store14 <- store14 %>% rename('Date'='Group.1')
store15 <- store15 %>% rename('Weekly_sales'='x')
store15 <- store15 %>% rename('Date'='Group.1')
store16 <- store16 %>% rename('Weekly_sales'='x')
store16 <- store16 %>% rename('Date'='Group.1')
store17 <- store17 %>% rename('Weekly_sales'='x')
store17 <- store17 %>% rename('Date'='Group.1')
store18 <- store18 %>% rename('Weekly_sales'='x')
store18 <- store18 %>% rename('Date'='Group.1')
store19 <- store19 %>% rename('Weekly_sales'='x')
store19 <- store19 %>% rename('Date'='Group.1')

store20 <- store20 %>% rename('Weekly_sales'='x')
store20 <- store20 %>% rename('Date'='Group.1')
store21 <- store21 %>% rename('Weekly_sales'='x')
store21 <- store21 %>% rename('Date'='Group.1')
store22 <- store22 %>% rename('Weekly_sales'='x')
store22 <- store22 %>% rename('Date'='Group.1')
store23 <- store23 %>% rename('Weekly_sales'='x')
store23 <- store23 %>% rename('Date'='Group.1')
store24 <- store24 %>% rename('Weekly_sales'='x')
store24 <- store24 %>% rename('Date'='Group.1')
store25 <- store25 %>% rename('Weekly_sales'='x')
store25 <- store25 %>% rename('Date'='Group.1')
store26 <- store26 %>% rename('Weekly_sales'='x')
```

STEP 7:
Here we write files for each store.

This was done in case we decided we wanted to explore multiple stores and compare them. Ultimately, for this project, we decided to stick to one of these stores.

```
install.packages("xlsx")
write.csv(store1, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store1.csv', row.names=FALSE)
write.csv(store2, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store2.csv', row.names=FALSE)
write.csv(store3, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store3.csv', row.names=FALSE)
write.csv(store4, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store4.csv', row.names=FALSE)
write.csv(store5, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store5.csv', row.names=FALSE)
write.csv(store6, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store6.csv', row.names=FALSE)
write.csv(store7, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store7.csv', row.names=FALSE)
write.csv(store8, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store8.csv', row.names=FALSE)
write.csv(store9, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store9.csv', row.names=FALSE)

write.csv(store10, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store10.csv', row.names=FALSE)
write.csv(store11, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store11.csv', row.names=FALSE)
write.csv(store12, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store12.csv', row.names=FALSE)
write.csv(store13, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store13.csv', row.names=FALSE)
write.csv(store14, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store14.csv', row.names=FALSE)
write.csv(store15, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store15.csv', row.names=FALSE)
write.csv(store16, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store16.csv', row.names=FALSE)
write.csv(store17, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store17.csv', row.names=FALSE)
write.csv(store18, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store18.csv', row.names=FALSE)
write.csv(store19, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store19.csv', row.names=FALSE)
write.csv(store20, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store20.csv', row.names=FALSE)
write.csv(store21, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store21.csv', row.names=FALSE)
write.csv(store22, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store22.csv', row.names=FALSE)
write.csv(store23, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store23.csv', row.names=FALSE)
write.csv(store24, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store24.csv', row.names=FALSE)
write.csv(store25, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store25.csv', row.names=FALSE)
write.csv(store26, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store26.csv', row.names=FALSE)
write.csv(store27, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM
project/store_data/data_cleaned_store27.csv', row.names=FALSE)
```

A.4 CREATING SUBSETS FOR EACH STORE

STEP 8:
Here we use rbind to bind all of the individual stores together by row.

```
all_stores_data <- rbind(store1,
                         store2,
                         store3,
                         store4,
                         store5,
                         store6,
                         store7,
                         store8,
                         store9,
                         store10,
                         store11,
                         store12,
                         store13,
                         store14,
                         store15,
                         store16,
                         store17,
                         store18,
                         store19,
                         store20,
                         store21,
                         store22,
                         store23,
                         store24,
                         store25,
                         store26,
                         store27,
                         store28,
                         store29,
                         store30,
                         store31,
                         store32,
                         store33,
                         store34,
                         store35,
                         store36,
                         store37,
                         store38,
                         store39,
                         store40,
                         store41,
                         store42,
                         store43,
                         store44,
                         store45)
```

STEP 9: Here we write the allstores data to a csv file

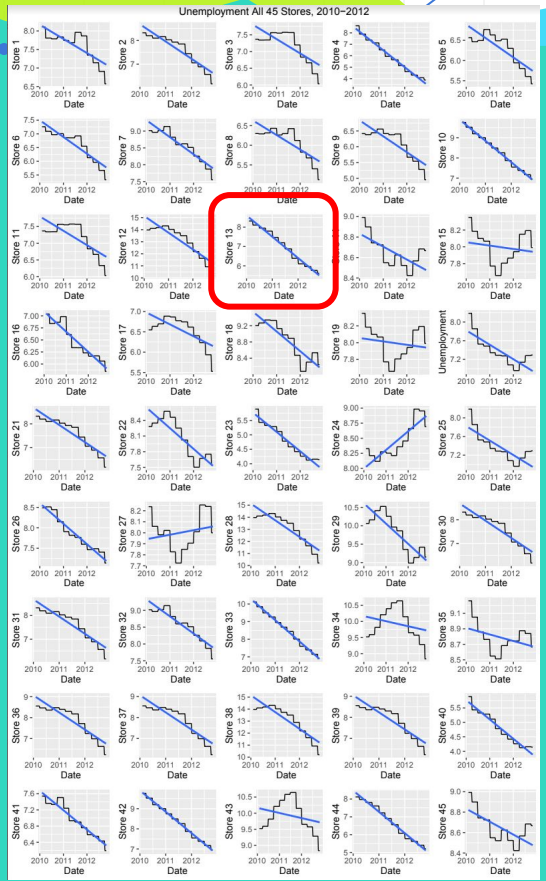
```
write.csv(data, 'C:/Users/danie/OneDrive/Desktop/Data Analysis & Decision Making/project/DADM project/store_data/all_stores_data.csv', row.names=FALSE)
```

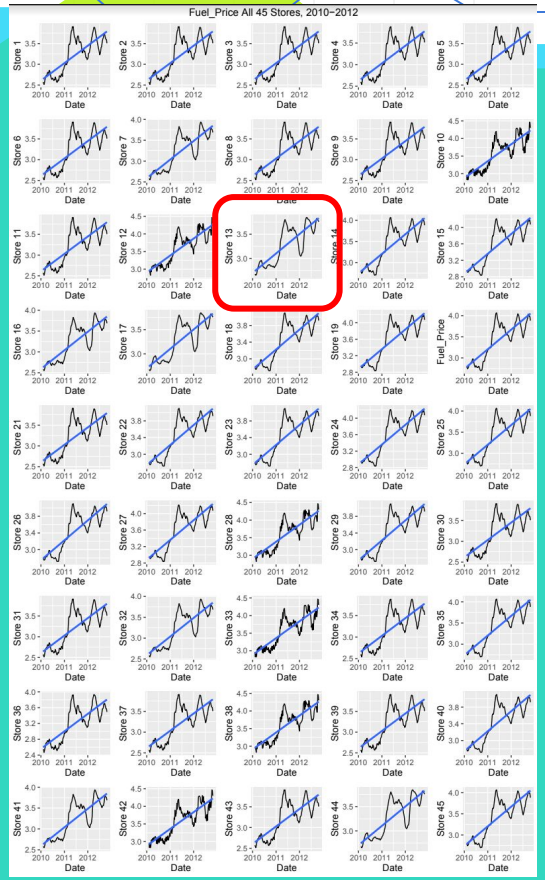
STEP 10: rename the allstores data to data to be used to analyze which store we wanted to focus in. This data was used into to create the visualization analysis of all stores.

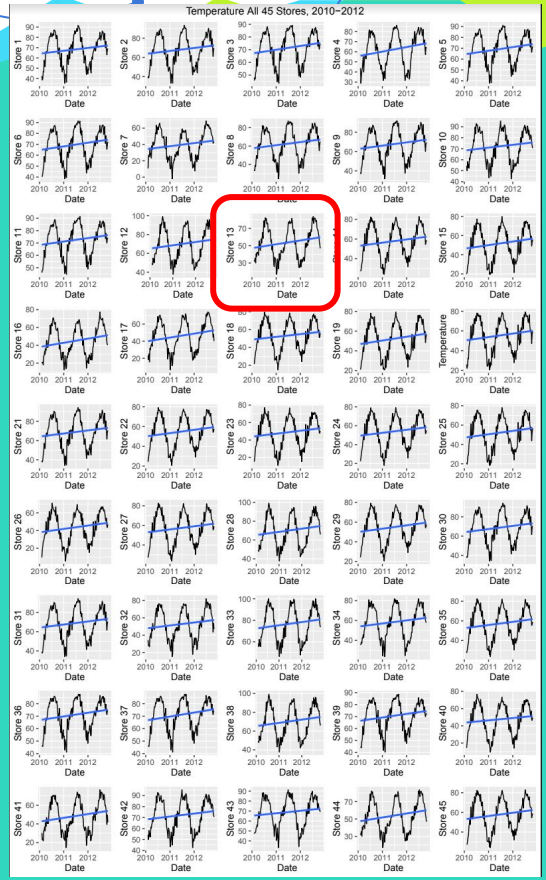
```
data <- all_stores_data
```

A.5 VISUAL ANALYSIS OF ALL STORES

TO DETERMINE WHICH STORE TO FOCUS ON

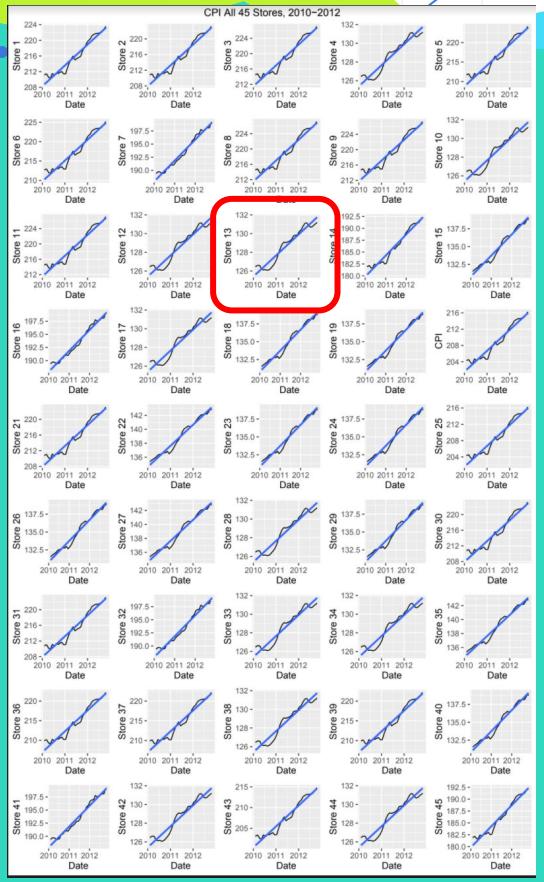


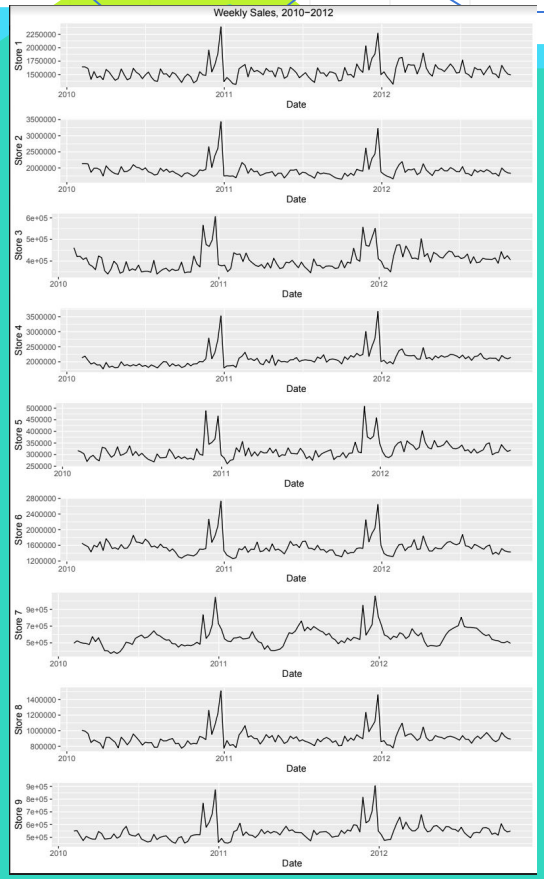


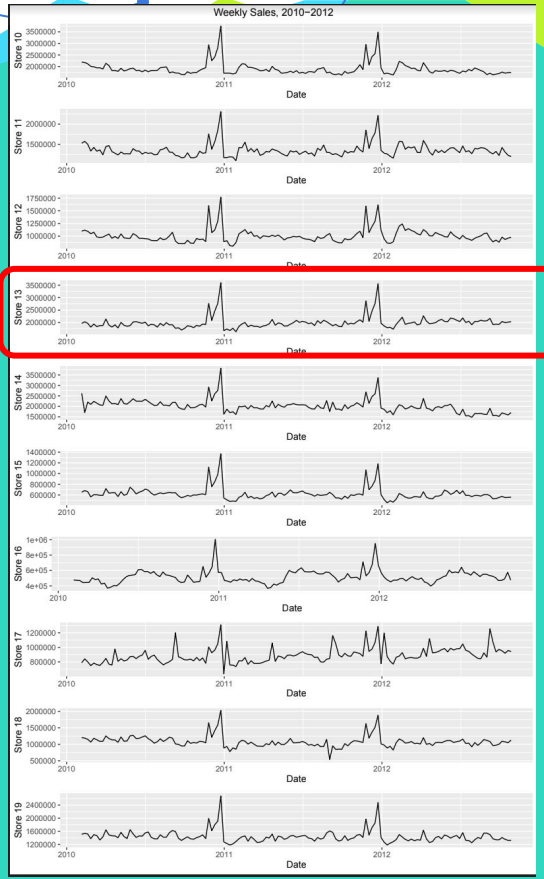


A.5 VISUAL ANALYSIS OF ALL STORES

TO DETERMINE WHICH STORE TO FOCUS ON

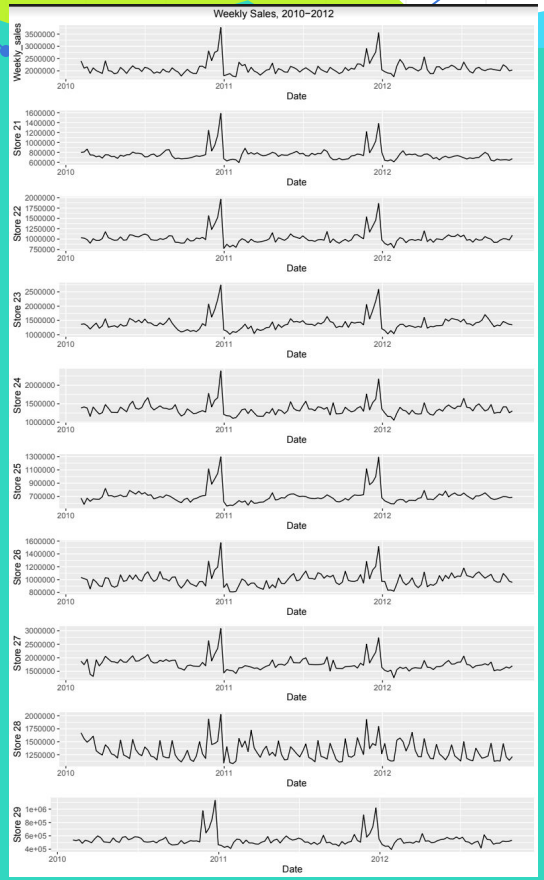


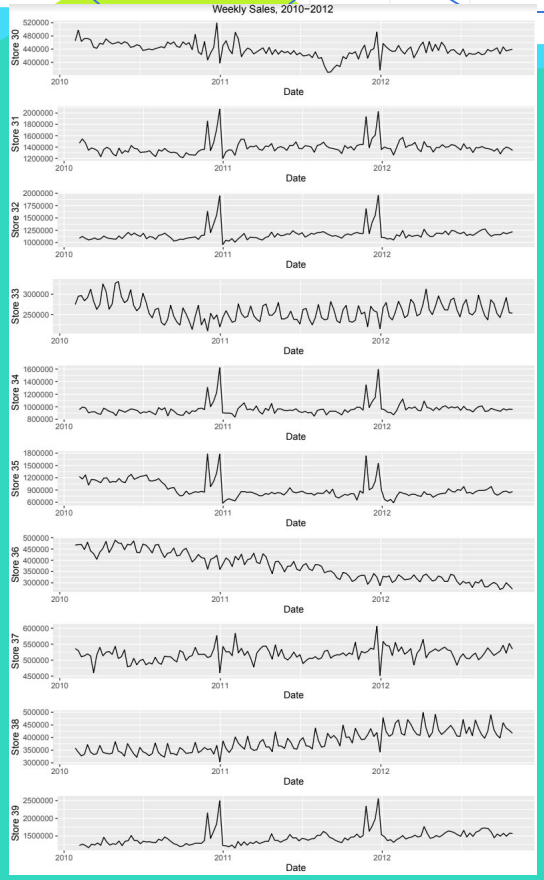


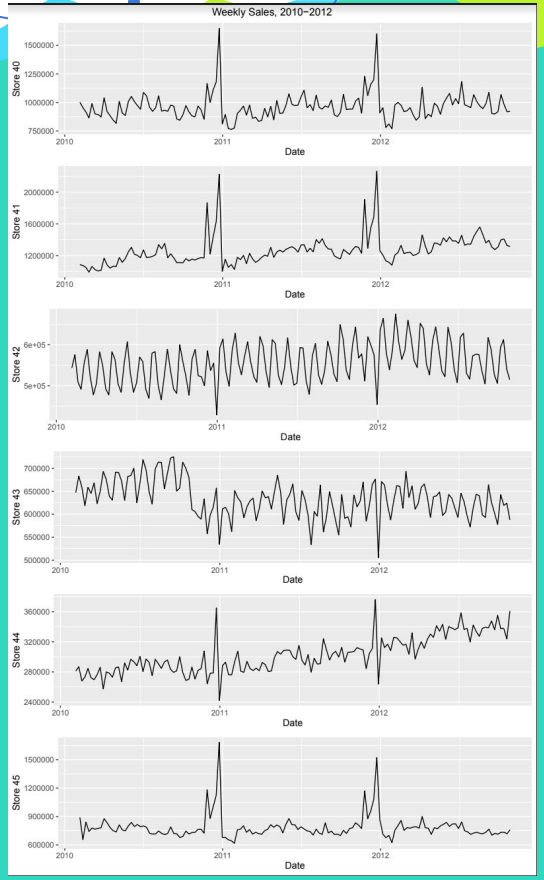


A.5 VISUAL ANALYSIS OF ALL STORES

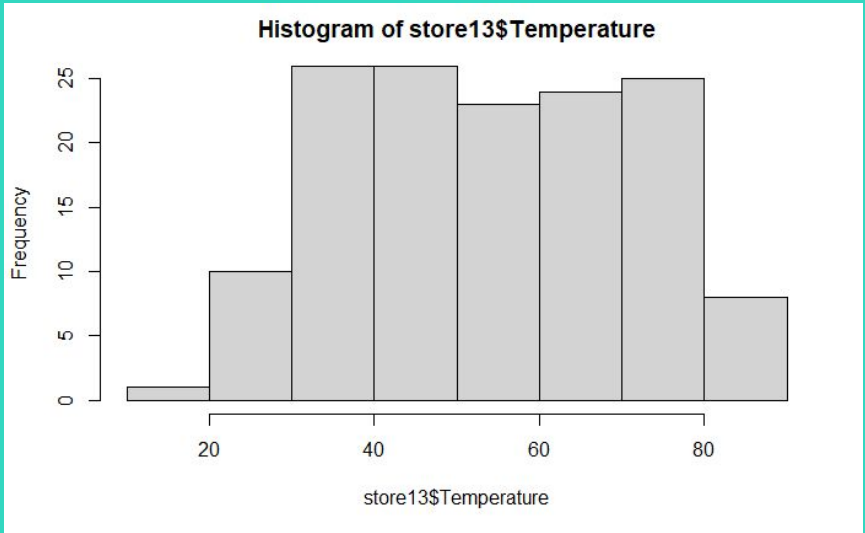
TO DETERMINE WHICH STORE TO FOCUS ON



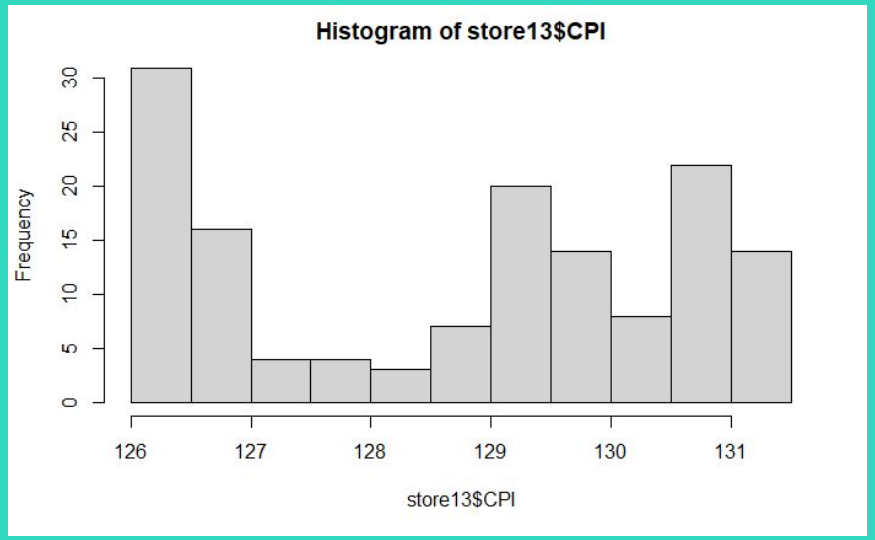


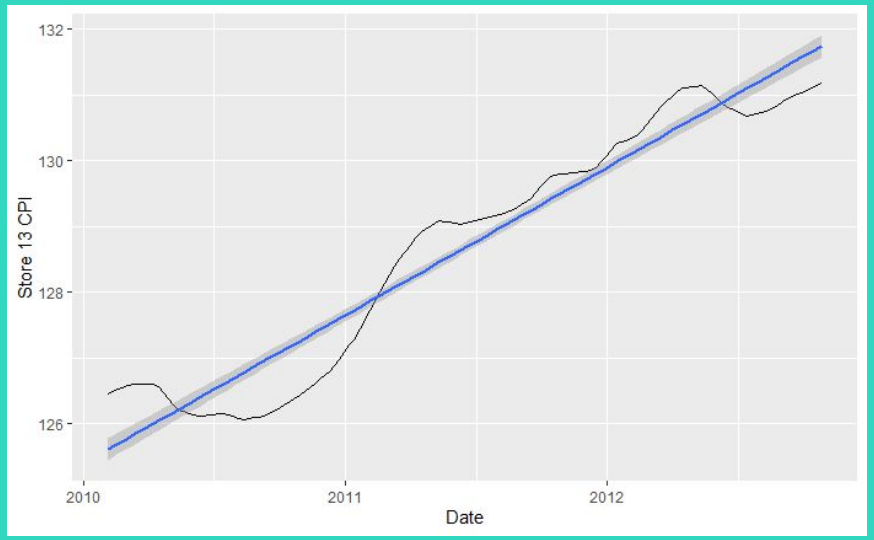


A.6 THE IDEAL STORE: STORE 13

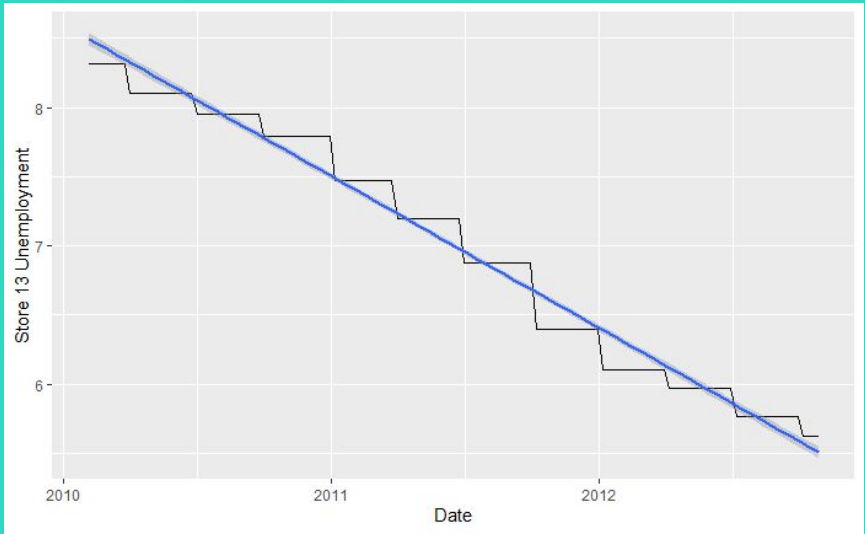


A.6 THE IDEAL STORE: STORE 13





A.6 THE IDEAL STORE: STORE 13



A.6 THE IDEAL STORE: STORE 13