Merged Final Econ Project

2023-12-11

Load libraries

```
library(readr)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(Metrics)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Importing the dataset

```
df_store <- read_csv("~/University/E401 Machine Learning For Economic Data/Final Project/Walmart Sales I
```

```
## Rows: 45 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Type
## dbl (2): Store, Size
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df_features = read_csv("~/University/E401 Machine Learning For Economic Data/Final Project/Walmart Sales
```

```
## Rows: 8190 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (10): Store, Temperature, Fuel_Price, MarkDown1, MarkDown2, MarkDown3, ...
## lgl   (1): IsHoliday
## date  (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df_train = read_csv("~/University/E401 Machine Learning For Economic Data/Final Project/Walmart Sales Fo
```

```
## Rows: 421570 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (3): Store, Dept, Weekly_Sales
## lgl  (1): IsHoliday
## date (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Preprocessing of the dataset

```
# Display the first three rows of the data frame
head(df_store, 3)
```

```
## # A tibble: 3 x 3
##   Store Type    Size
##   <dbl> <chr>  <dbl>
## 1     1 A     151315
## 2     2 A     202307
## 3     3 B      37392
```

```
head(df_features, 3)
```

```
## # A tibble: 3 x 12
##   Store Date       Temperature Fuel_Price MarkDown1 MarkDown2 MarkDown3
##   <dbl> <date>           <dbl>      <dbl>     <dbl>     <dbl>     <dbl>
## 1     1 2010-02-05        42.3       2.57        NA        NA        NA
## 2     1 2010-02-12        38.5       2.55        NA        NA        NA
## 3     1 2010-02-19        39.9       2.51        NA        NA        NA
## # i 5 more variables: MarkDown4 <dbl>, MarkDown5 <dbl>, CPI <dbl>,
## #   Unemployment <dbl>, IsHoliday <lgl>
```

```r
#merging df_train and df_features dataset and df_store
df <- inner_join(inner_join(df_train, df_features, by = c("Store", "Date")), df_store, by = "Store")

# Print the first 5 rows of the resulting data frame
head(df, 5)
```

```
## # A tibble: 5 x 17
##   Store  Dept Date       Weekly_Sales IsHoliday.x Temperature Fuel_Price
##   <dbl> <dbl> <date>            <dbl> <lgl>             <dbl>      <dbl>
## 1     1     1 2010-02-05       24924. FALSE              42.3       2.57
## 2     1     1 2010-02-12       46039. TRUE               38.5       2.55
## 3     1     1 2010-02-19       41596. FALSE              39.9       2.51
## 4     1     1 2010-02-26       19404. FALSE              46.6       2.56
## 5     1     1 2010-03-05       21828. FALSE              46.5       2.62
## # i 10 more variables: MarkDown1 <dbl>, MarkDown2 <dbl>, MarkDown3 <dbl>,
## #   MarkDown4 <dbl>, MarkDown5 <dbl>, CPI <dbl>, Unemployment <dbl>,
## #   IsHoliday.y <lgl>, Type <chr>, Size <dbl>
```

```r
# removing duplicate column since IsHoliday_y and IsHoliday_x are the same
df$IsHoliday.y <- NULL
#renaming Isholiday.x as isHoliday
names(df)[names(df) == "IsHoliday.x"] <- "IsHoliday"
# Print the first 5 rows of the resulting data frame
head(df, 5)
```

```
## # A tibble: 5 x 16
##   Store  Dept Date       Weekly_Sales IsHoliday Temperature Fuel_Price MarkDown1
##   <dbl> <dbl> <date>            <dbl> <lgl>           <dbl>      <dbl>     <dbl>
## 1     1     1 2010-02-05       24924. FALSE            42.3       2.57        NA
## 2     1     1 2010-02-12       46039. TRUE             38.5       2.55        NA
## 3     1     1 2010-02-19       41596. FALSE            39.9       2.51        NA
## 4     1     1 2010-02-26       19404. FALSE            46.6       2.56        NA
## 5     1     1 2010-03-05       21828. FALSE            46.5       2.62        NA
## # i 8 more variables: MarkDown2 <dbl>, MarkDown3 <dbl>, MarkDown4 <dbl>,
## #   MarkDown5 <dbl>, CPI <dbl>, Unemployment <dbl>, Type <chr>, Size <dbl>
```

```r
# check for non-zero and zero values for weekly sales
filtered_df <- subset(df, Weekly_Sales <= 0)
filtered_df
```

```
## # A tibble: 1,358 x 16
##    Store  Dept Date       Weekly_Sales IsHoliday Temperature Fuel_Price
##    <dbl> <dbl> <date>            <dbl> <lgl>           <dbl>      <dbl>
##  1     1     6 2012-08-10      -140.   FALSE            85.0       3.49
##  2     1    18 2012-05-04        -1.27 FALSE            75.6       3.75
##  3     1    47 2010-02-19      -863    FALSE            39.9       2.51
##  4     1    47 2010-03-12      -698    FALSE            57.8       2.67
##  5     1    47 2010-10-08       -58    FALSE            63.9       2.63
##  6     1    47 2011-03-11         0    FALSE            53.6       3.46
##  7     1    47 2011-04-08      -298    FALSE            67.8       3.62
##  8     1    47 2011-07-08      -198    FALSE            85.8       3.48
##  9     1    47 2011-08-12         0    FALSE            90.8       3.64
## 10     1    47 2011-08-19         0    FALSE            89.9       3.55
## # i 1,348 more rows
## # i 9 more variables: MarkDown1 <dbl>, MarkDown2 <dbl>, MarkDown3 <dbl>,
## #   MarkDown4 <dbl>, MarkDown5 <dbl>, CPI <dbl>, Unemployment <dbl>,
```

```
## #   Type <chr>, Size <dbl>
```

```
# total rows in dataframe = 421570
# total rows with missing or zero or negative values = 1348
# perventage of rows with missing values= 1348/ 421570= 0.31%

# therefore, removing them
df <- subset(df, Weekly_Sales > 0)
df
```

```
## # A tibble: 420,212 x 16
##    Store  Dept Date       Weekly_Sales IsHoliday Temperature Fuel_Price
##    <dbl> <dbl> <date>            <dbl> <lgl>           <dbl>      <dbl>
##  1     1     1 2010-02-05       24924. FALSE            42.3       2.57
##  2     1     1 2010-02-12       46039. TRUE             38.5       2.55
##  3     1     1 2010-02-19       41596. FALSE            39.9       2.51
##  4     1     1 2010-02-26       19404. FALSE            46.6       2.56
##  5     1     1 2010-03-05       21828. FALSE            46.5       2.62
##  6     1     1 2010-03-12       21043. FALSE            57.8       2.67
##  7     1     1 2010-03-19       22137. FALSE            54.6       2.72
##  8     1     1 2010-03-26       26229. FALSE            51.4       2.73
##  9     1     1 2010-04-02       57258. FALSE            62.3       2.72
## 10     1     1 2010-04-09       42961. FALSE            65.9       2.77
## # i 420,202 more rows
## # i 9 more variables: MarkDown1 <dbl>, MarkDown2 <dbl>, MarkDown3 <dbl>,
## #   MarkDown4 <dbl>, MarkDown5 <dbl>, CPI <dbl>, Unemployment <dbl>,
## #   Type <chr>, Size <dbl>
```

```
colnames(df)[colnames(df) == "IsHoliday_x"] <- "IsHoliday"
df
```

```
## # A tibble: 420,212 x 16
##    Store  Dept Date       Weekly_Sales IsHoliday Temperature Fuel_Price
##    <dbl> <dbl> <date>            <dbl> <lgl>           <dbl>      <dbl>
##  1     1     1 2010-02-05       24924. FALSE            42.3       2.57
##  2     1     1 2010-02-12       46039. TRUE             38.5       2.55
##  3     1     1 2010-02-19       41596. FALSE            39.9       2.51
##  4     1     1 2010-02-26       19404. FALSE            46.6       2.56
##  5     1     1 2010-03-05       21828. FALSE            46.5       2.62
##  6     1     1 2010-03-12       21043. FALSE            57.8       2.67
##  7     1     1 2010-03-19       22137. FALSE            54.6       2.72
##  8     1     1 2010-03-26       26229. FALSE            51.4       2.73
##  9     1     1 2010-04-02       57258. FALSE            62.3       2.72
## 10     1     1 2010-04-09       42961. FALSE            65.9       2.77
## # i 420,202 more rows
## # i 9 more variables: MarkDown1 <dbl>, MarkDown2 <dbl>, MarkDown3 <dbl>,
## #   MarkDown4 <dbl>, MarkDown5 <dbl>, CPI <dbl>, Unemployment <dbl>,
## #   Type <chr>, Size <dbl>
```

Data Description

```
summary(df)
```

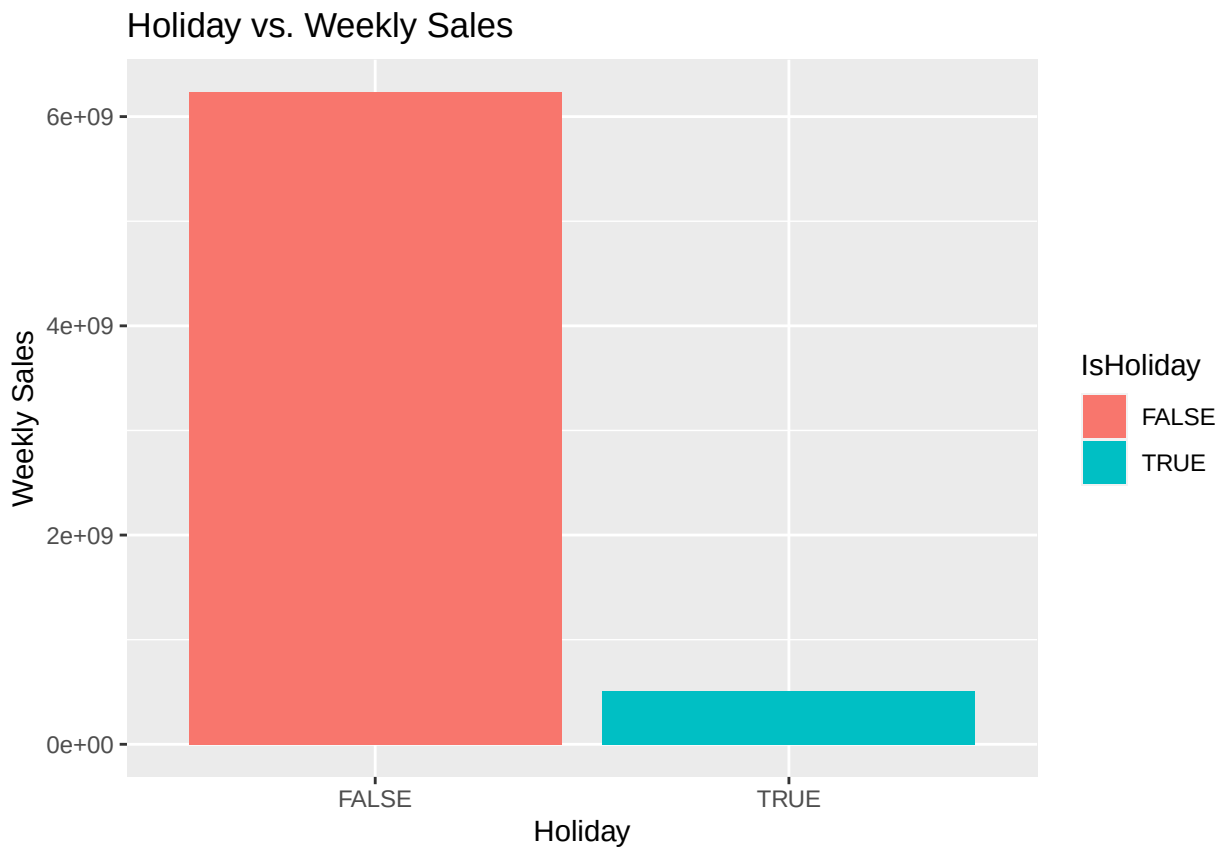
```
##      Store            Dept            Date              Weekly_Sales
##  Min.   : 1.0   Min.   : 1.00   Min.   :2010-02-05   Min.   :     0
##  1st Qu.:11.0   1st Qu.:18.00   1st Qu.:2010-10-08   1st Qu.:  2120
##  Median :22.0   Median :37.00   Median :2011-06-17   Median :  7662
```

```
##  Mean   :22.2   Mean   :44.24   Mean   :2011-06-18   Mean   : 16033
##  3rd Qu.:33.0   3rd Qu.:74.00   3rd Qu.:2012-02-24   3rd Qu.: 20271
##  Max.   :45.0   Max.   :99.00   Max.   :2012-10-26   Max.   :693099
##
##  IsHoliday        Temperature       Fuel_Price      MarkDown1
##  Mode :logical   Min.   : -2.06   Min.   :2.472   Min.   :    0.27
##  FALSE:390652    1st Qu.: 46.68   1st Qu.:2.933   1st Qu.: 2240.27
##  TRUE :29560     Median : 62.09   Median :3.452   Median : 5347.45
##                  Mean   : 60.09   Mean   :3.361   Mean   : 7247.82
##                  3rd Qu.: 74.28   3rd Qu.:3.738   3rd Qu.: 9210.90
##                  Max.   :100.14   Max.   :4.468   Max.   :88646.76
##                                                   NA's   :270031
##    MarkDown2          MarkDown3          MarkDown4          MarkDown5
##  Min.   :  -265.8   Min.   :   -29.1   Min.   :    0.22   Min.   :   135.2
##  1st Qu.:    41.6   1st Qu.:     5.1   1st Qu.:  504.22   1st Qu.:  1878.4
##  Median :   192.0   Median :    24.6   Median : 1481.31   Median :  3359.4
##  Mean   :  3330.2   Mean   :  1441.7   Mean   : 3384.78   Mean   :  4629.5
##  3rd Qu.:  1926.9   3rd Qu.:   104.0   3rd Qu.: 3595.04   3rd Qu.:  5563.8
##  Max.   :104519.5   Max.   :141630.6   Max.   :67474.85   Max.   :108519.3
##  NA's   :309308     NA's   :283561     NA's   :285694     NA's   :269283
##       CPI         Unemployment        Type                Size
##  Min.   :126.1   Min.   : 3.879   Length:420212      Min.   : 34875
##  1st Qu.:132.0   1st Qu.: 6.891   Class :character   1st Qu.: 93638
##  Median :182.4   Median : 7.866   Mode  :character   Median :140167
##  Mean   :171.2   Mean   : 7.960                      Mean   :136750
##  3rd Qu.:212.4   3rd Qu.: 8.567                      3rd Qu.:202505
##  Max.   :227.2   Max.   :14.313                      Max.   :219622
##
```

EXPLORATORY DATA ANALYSIS

Holiday vs. Weekly Sales

```r
# Create a bar plot
ggplot(df, aes(x = IsHoliday, y = Weekly_Sales, fill = IsHoliday)) +
  geom_bar(stat = "identity") +
  labs(x = "Holiday", y = "Weekly Sales") +
  ggtitle("Holiday vs. Weekly Sales")
```



Print the dates of holidays

```
# Create a subset for rows where 'IsHoliday' is true
df_holiday <- subset(df, IsHoliday == TRUE)

# Get unique dates from the subset
unique_dates <- unique(df_holiday$Date)

# Print unique dates
print(unique_dates)
```

```
##  [1] "2010-02-12" "2010-09-10" "2010-11-26" "2010-12-31" "2011-02-11"
##  [6] "2011-09-09" "2011-11-25" "2011-12-30" "2012-02-10" "2012-09-07"
```

Super Bowl, Labor Day, Thanksgiving, Christmas are the holidays tht are present in the dataset.

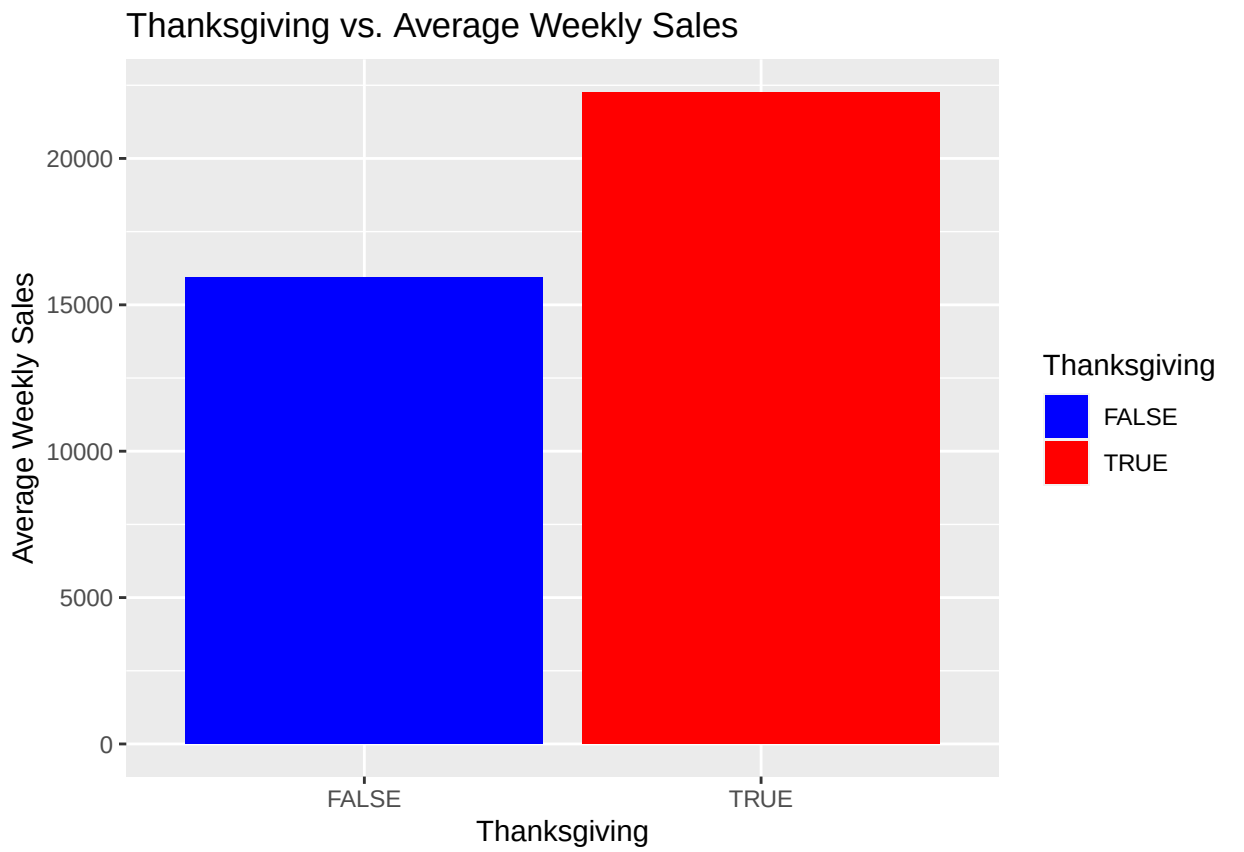
Create different rows in the dataset for the unique holidays

```
#create different rows in the dataset for the unique holidays
df$Super_Bowl <- ifelse(df$Date %in% as.Date(c('2010-02-12', '2011-02-11', '2012-02-10')), TRUE, FALSE)
df$Thanksgiving <- ifelse(df$Date %in% as.Date(c('2010-11-26', '2011-11-25')), TRUE, FALSE)
df$Labor_Day <- ifelse(df$Date %in% as.Date(c('2010-09-10', '2011-09-09', '2012-09-07')), TRUE, FALSE)
df$Christmas <- ifelse(df$Date %in% as.Date(c('2010-12-31', '2011-12-30')), TRUE, FALSE)
```

THANKSGIVING AVERAGE SALES VS AVERAGE WEEKLY SALES

```
# Calculate the average weekly sales for both TRUE and FALSE values
average_sales <- tapply(df$Weekly_Sales, df$Thanksgiving, mean)
```

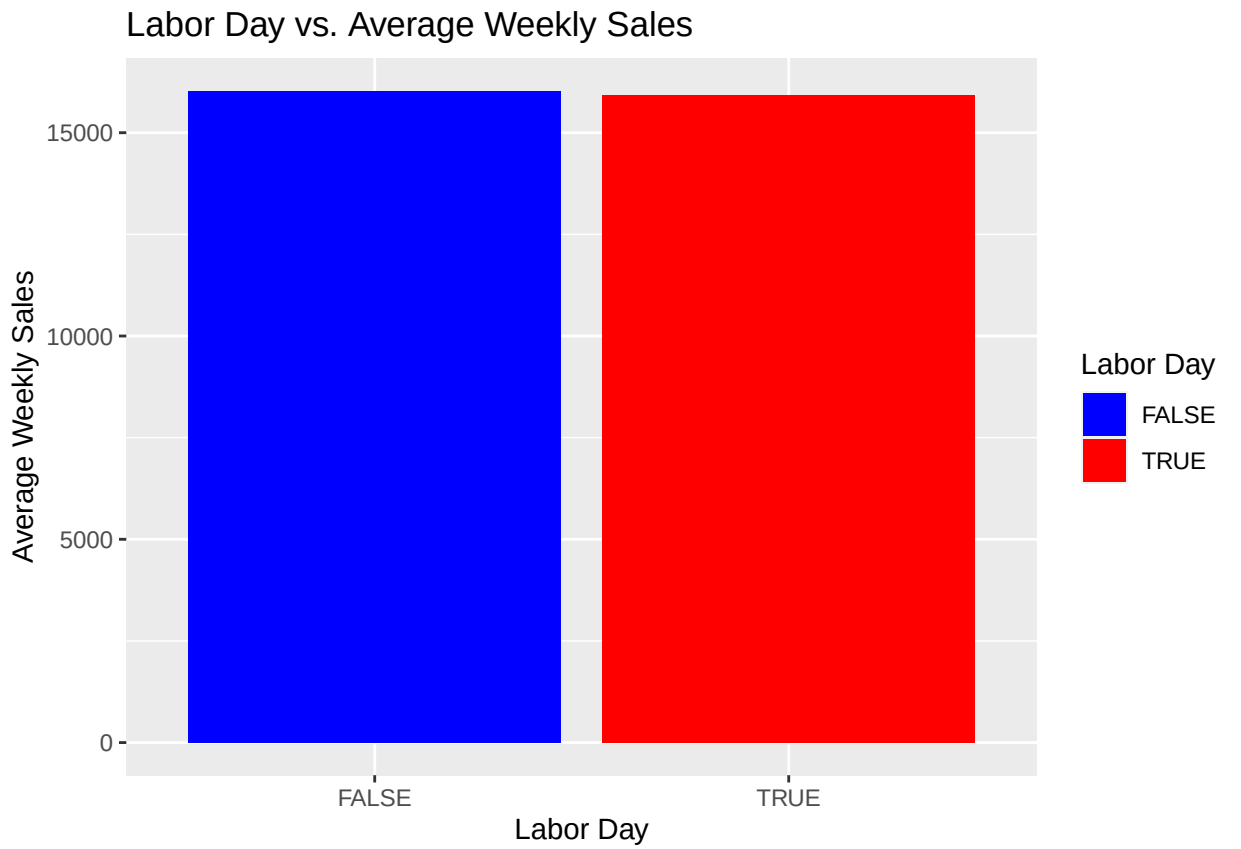
```r
# Create a data frame for plotting
plot_data <- data.frame(
  Thanksgiving = factor(names(average_sales)),
  Average_Weekly_Sales = average_sales
)

# Create a bar plot
ggplot(plot_data, aes(x = Thanksgiving, y = Average_Weekly_Sales, fill = Thanksgiving)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Thanksgiving", y = "Average Weekly Sales") +
  scale_fill_manual(values = c("FALSE" = "blue", "TRUE" = "red"), name = "Thanksgiving") +
  ggtitle("Thanksgiving vs. Average Weekly Sales")
```



LABOUR DAY AVERAGE SALES VS AVERAGE WEEKLY SALES

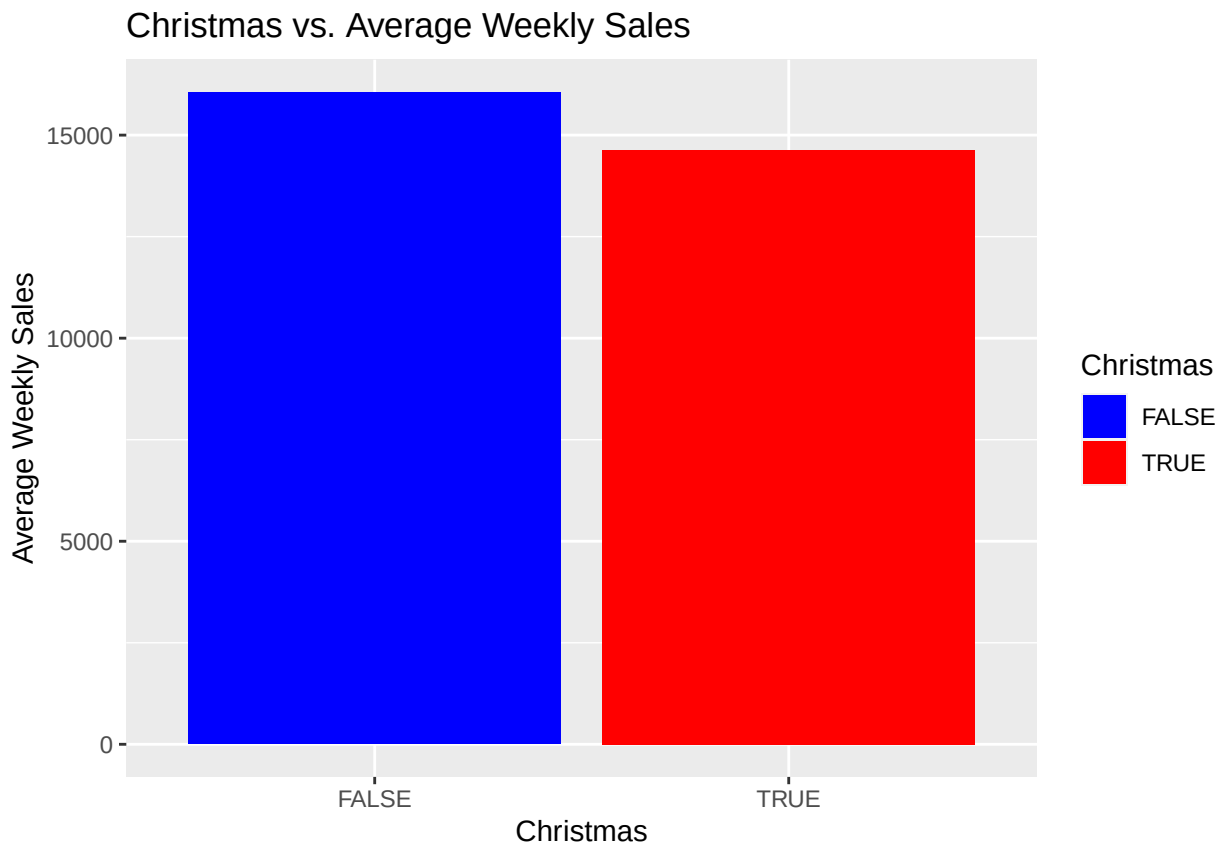
```r
# Calculate the average weekly sales for both TRUE and FALSE values for Labor Day
average_sales_labor_day <- tapply(df$Weekly_Sales, df$Labor_Day, mean)

# Create a data frame for plotting for Labor Day
plot_data_labor_day <- data.frame(
  Holiday = factor(names(average_sales_labor_day)),
  Average_Weekly_Sales = average_sales_labor_day
)

# Create a bar plot for Labor Day
ggplot(plot_data_labor_day, aes(x = Holiday, y = Average_Weekly_Sales, fill = Holiday)) +
  geom_bar(stat = "identity", position = "dodge") +
```

```
  labs(x = "Labor Day", y = "Average Weekly Sales") +
  scale_fill_manual(values = c("FALSE" = "blue", "TRUE" = "red"), name = "Labor Day") +
  ggtitle("Labor Day vs. Average Weekly Sales")
```



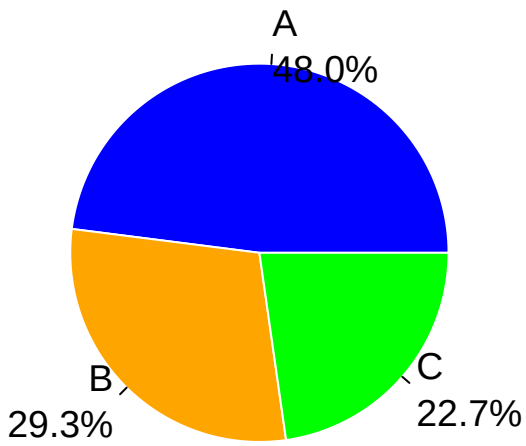
Christmas vs. Average Weekly Sales

```
# Calculate the average weekly sales for both TRUE and FALSE values for Christmas
average_sales_christmas <- tapply(df$Weekly_Sales, df$Christmas, mean)

# Create a data frame for plotting for Christmas
plot_data_christmas <- data.frame(
  Holiday = factor(names(average_sales_christmas)),
  Average_Weekly_Sales = average_sales_christmas
)

# Create a bar plot for Christmas
ggplot(plot_data_christmas, aes(x = Holiday, y = Average_Weekly_Sales, fill = Holiday)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Christmas", y = "Average Weekly Sales") +
  scale_fill_manual(values = c("FALSE" = "blue", "TRUE" = "red"), name = "Christmas") +
  ggtitle("Christmas vs. Average Weekly Sales")
```



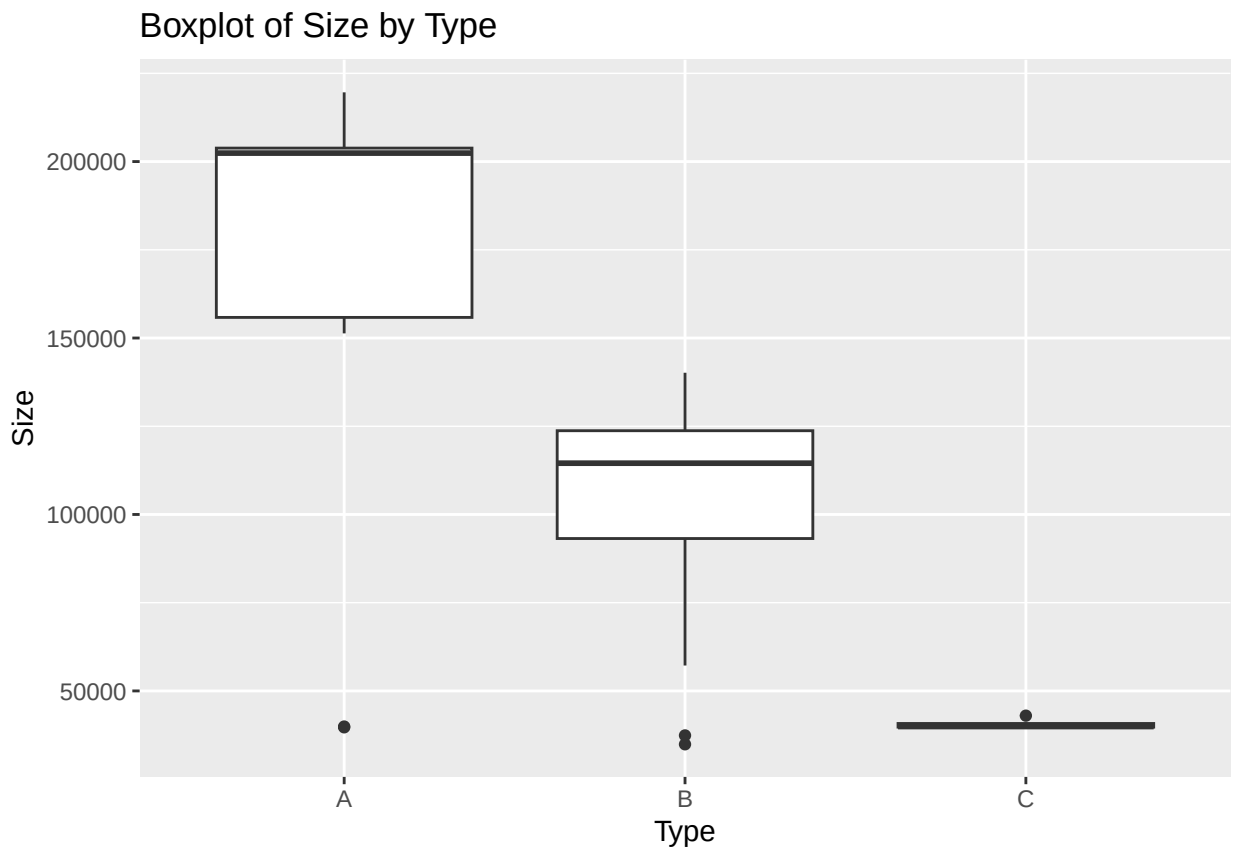
Pie chart withthe average weekly sales by store type

```
# Calculate the average weekly sales by store type
average_sales_by_store <- aggregate(Weekly_Sales ~ Type, data = df, FUN = mean)

# Store types
store_types <- average_sales_by_store$Type

# Average weekly sales data
average_sales_data <- average_sales_by_store$Weekly_Sales

# Plotting a pie chart with percentage labels
pie(average_sales_data, labels = paste0(store_types, "\n", percent(average_sales_data / sum(average_sal
    col = c("blue", "orange", "green"), cex = 1.2, border = "white",
    main = "Average Weekly Sales by Store Type", cex.lab = 1.5)
```



Boxplot of Size of Store by Type of Store

```
# Create a boxplot
ggplot(df_store, aes(x = Type, y = Size)) +
  geom_boxplot() +
  labs(title = "Boxplot of Size by Type", x = "Type", y = "Size")
```

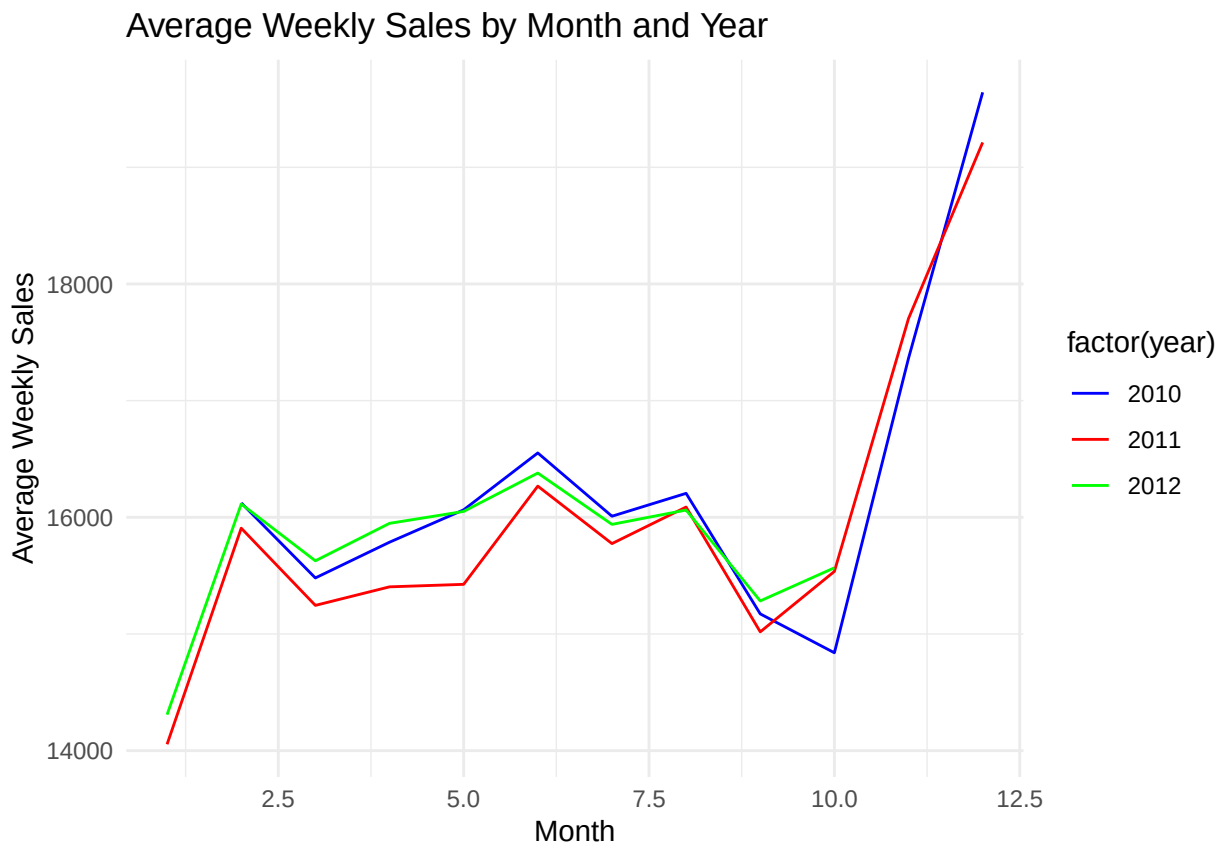


Scatter plot for 'month' and 'year' against average weekly sales

```
df$Date <- as.Date(df$Date)

# Create 'week', 'month', and 'year' columns
df$week <- week(df$Date)
df$month <- month(df$Date)
df$year <- year(df$Date)


# Calculate average weekly sales for each combination of 'month' and 'year'
df_avg <- df %>%
  group_by(month, year) %>%
  summarise(Avg_Weekly_Sales = mean(Weekly_Sales, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'month'. You can override using the
## `.groups` argument.
```

```
# Create a scatter plot for 'month' and 'year' against average weekly sales
ggplot(df_avg, aes(x = month, y = Avg_Weekly_Sales, color = factor(year))) +
  geom_line() +
  labs(title = "Average Weekly Sales by Month and Year", x = "Month", y = "Average Weekly Sales") +
  scale_color_manual(values = c("blue", "red", "green")) +  # Set your desired color for each year
  theme_minimal()
```

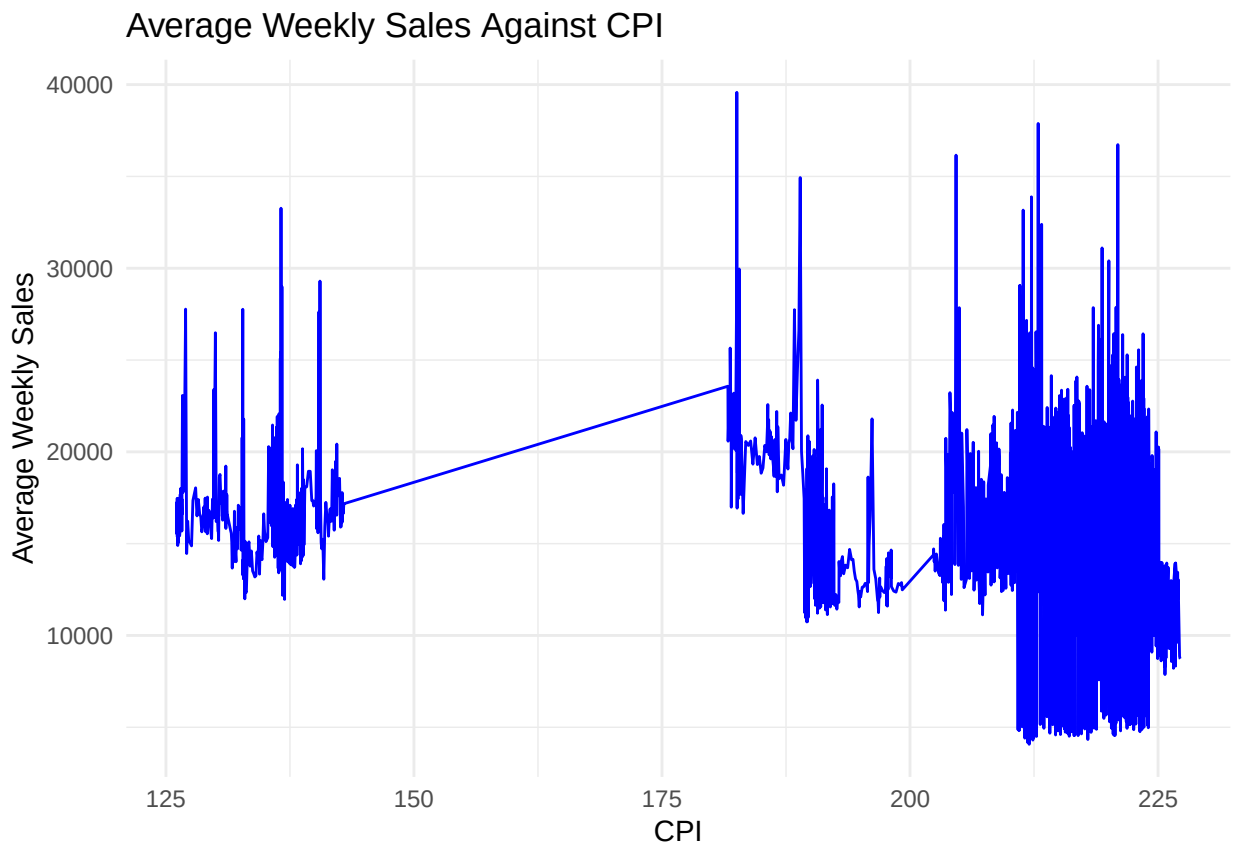


Average Weekly Sales Against Unemployment

```
average_sales <- df %>%
  group_by(Unemployment) %>%
  summarise(Avg_Weekly_Sales = mean(Weekly_Sales, na.rm = TRUE))

# Create a line plot with 'Unemployment' on the x-axis and average 'Weekly_Sales' on the y-axis
ggplot(average_sales, aes(x = Unemployment, y = Avg_Weekly_Sales)) +
  geom_line(color = "blue") +
  labs(title = "Average Weekly Sales Against Unemployment", x = "Unemployment", y = "Average Weekly Sal
  theme_minimal()
```

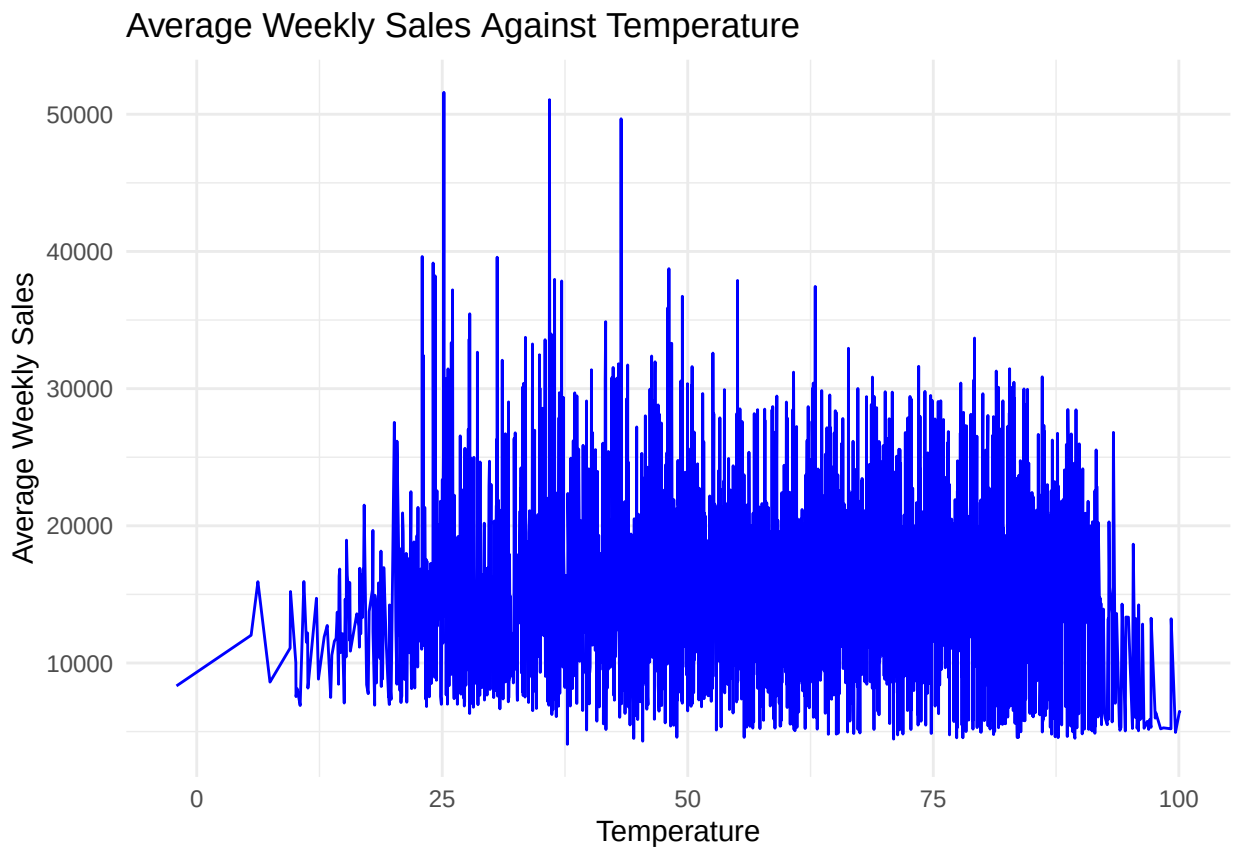
Average Weekly Sales Against Unemployment

```
average_sales <- df %>%
  group_by(CPI) %>%
  summarise(Avg_Weekly_Sales = mean(Weekly_Sales, na.rm = TRUE))

ggplot(average_sales, aes(x = CPI, y = Avg_Weekly_Sales)) +
  geom_line(color = "blue") +
  labs(title = "Average Weekly Sales Against CPI", x = "CPI", y = "Average Weekly Sales") +
  theme_minimal()
```

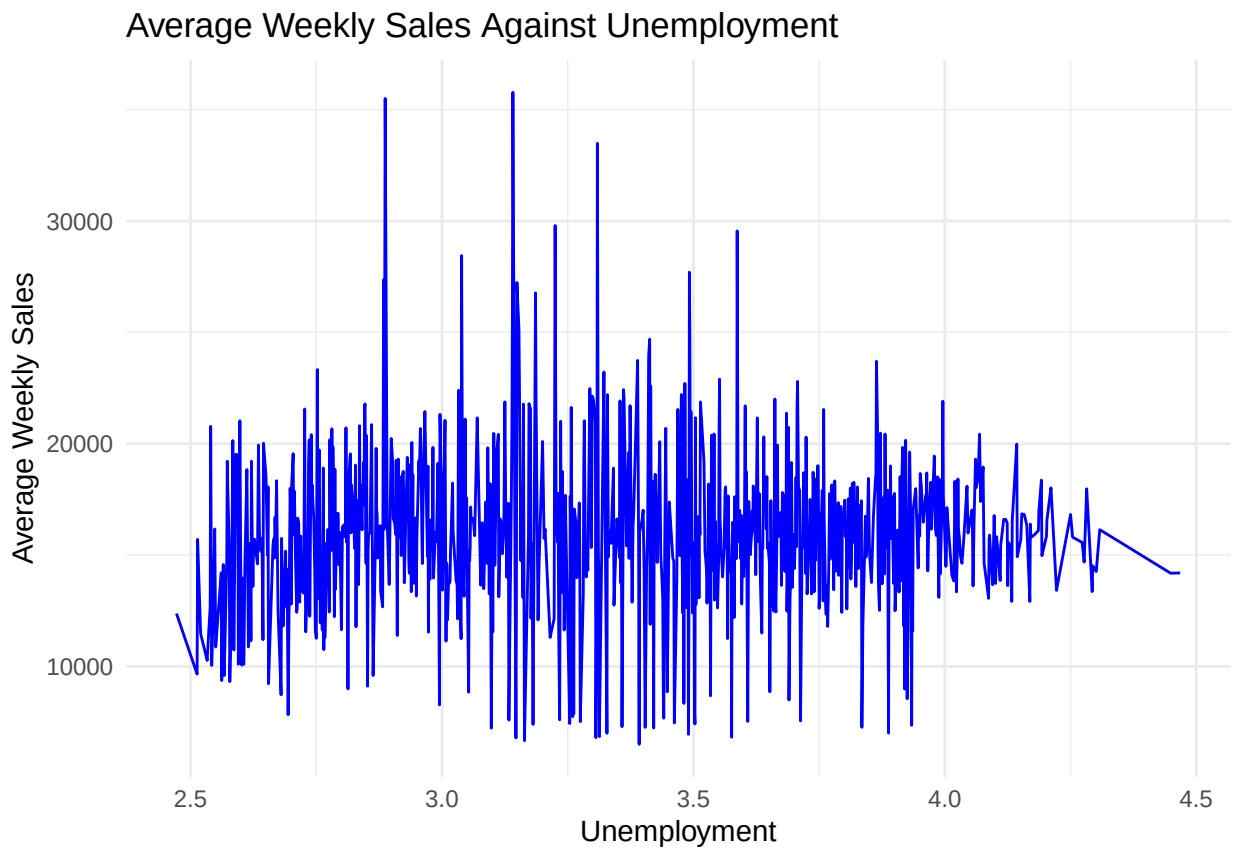


Average Weekly Sales Against Temperature

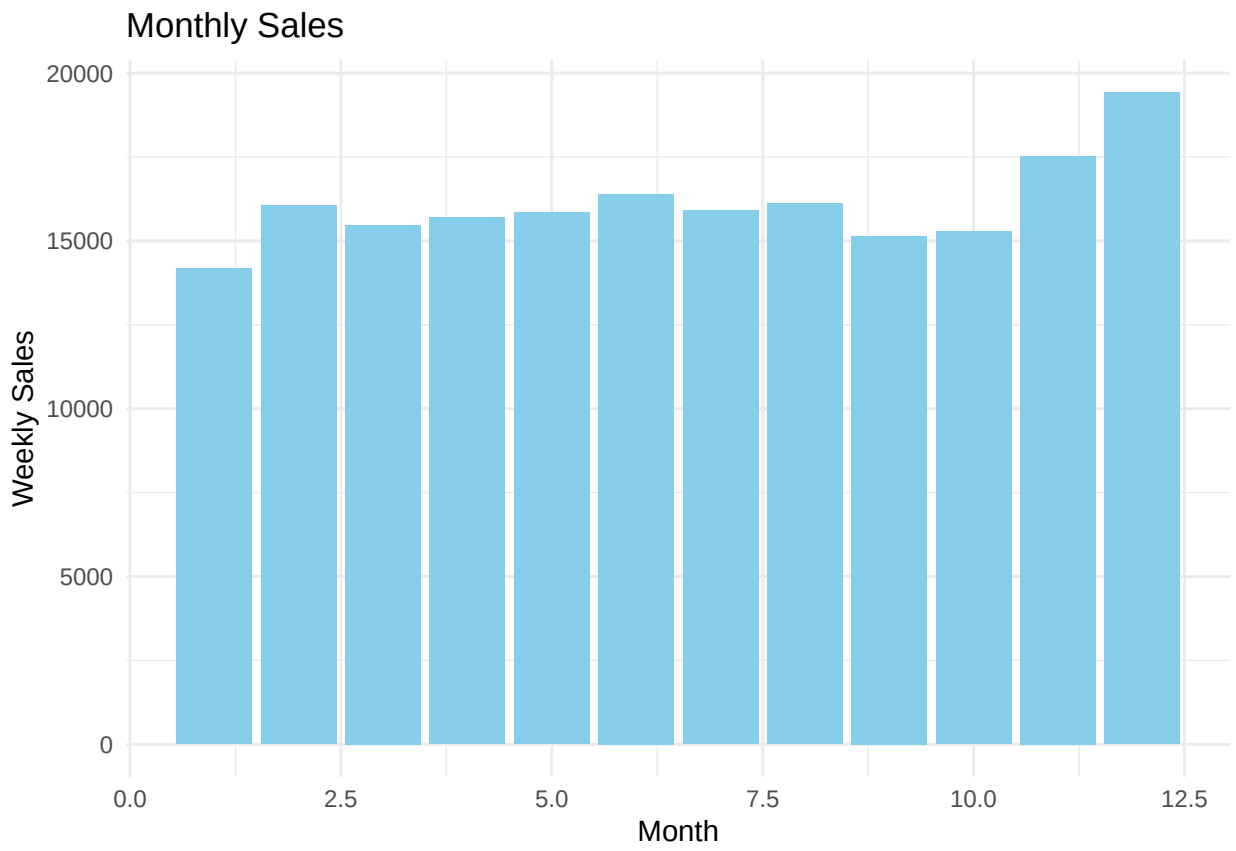
```
average_sales <- df %>%
  group_by(Temperature) %>%
  summarise(Avg_Weekly_Sales = mean(Weekly_Sales, na.rm = TRUE))

ggplot(average_sales, aes(x = Temperature, y = Avg_Weekly_Sales)) +
  geom_line(color = "blue") +
  labs(title = "Average Weekly Sales Against Temperature", x = "Temperature", y = "Average Weekly Sales
  theme_minimal()
```




```
average_sales <- df %>%
  group_by(Fuel_Price) %>%
  summarise(Avg_Weekly_Sales = mean(Weekly_Sales, na.rm = TRUE))

ggplot(average_sales, aes(x = Fuel_Price, y = Avg_Weekly_Sales)) +
  geom_line(color = "blue") +
  labs(title = "Average Weekly Sales Against Unemployment", x = "Unemployment", y = "Average Weekly Sal
  theme_minimal()
```




```r
average_sales <- df %>%
  group_by(month) %>%
  summarise(Avg_Weekly_Sales = mean(Weekly_Sales, na.rm = TRUE))
# Create a bar plot with 'month' on the x-axis and 'Weekly_Sales' on the y-axis
ggplot(average_sales, aes(x = month, y = Avg_Weekly_Sales)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Monthly Sales", x = "Month", y = "Weekly Sales") +
  theme_minimal()
```



Let's look at the correlation plot

```
data <- read_csv("~/University/E401 Machine Learning For Economic Data/Final Project/Walmart Sales Fore
```

```
## New names:
## Rows: 126 Columns: 28
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (1): Events dbl (27): ...1, Weekly_Sales, Holiday_Flag, Temperature,
## Fuel_Price, CPI, Un...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
corr = cor(data[, c(1:9)])
corrplot(corr, method = "color", cl.pos = 'n', rect.col = "black",  tl.col = "black", addCoef.col = "bla
```

Econometric Methods

Read the dataset

```
train <- read_csv("~/University/E401 Machine Learning For Economic Data/Final Project/Walmart Sales Fore
```

```
## New names:
## Rows: 81 Columns: 18
## -- Column specification
## -------------------------------------------------------- Delimiter: "," dbl
## (18): ...1, Weekly_Sales, Fuel_Price, Unemployment, Week_Number, EventsL...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
test <- read_csv("~/University/E401 Machine Learning For Economic Data/Final Project/Walmart Sales Forec
```

```
## New names:
## Rows: 45 Columns: 18
## -- Column specification
## -------------------------------------------------------- Delimiter: "," dbl
## (18): ...1, Weekly_Sales, Fuel_Price, Unemployment, Week_Number, EventsL...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

Prepare the train and test dataset

```r
y_train <- train$Weekly_Sales
X_train <- subset(train, select = -Weekly_Sales)
X_train <- as.matrix(X_train)
y_train <- as.matrix(y_train)
y_test <- test$Weekly_Sales
X_test <- subset(test, select = -Weekly_Sales)
```

Perform Linear Regression

```r
fit_lm <- lm(Weekly_Sales ~ ., data = train)
summary(fit_lm)
```

```
##
## Call:
## lm(formula = Weekly_Sales ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -123134  -46522    1841   44510  114359
##
## Coefficients: (1 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1637841.6   345979.4   4.734 1.26e-05 ***
## ...1                -146.0      757.6  -0.193 0.847794
## Fuel_Price         49812.9    47710.2   1.044 0.300380
## Unemployment      -50517.0    50231.0  -1.006 0.318350
## Week_Number             NA         NA      NA       NA
## EventsLabour.Day   79280.1    86347.1   0.918 0.361984
## EventsNo_Holiday  -25662.8    72364.6  -0.355 0.724031
## month2            260225.4    42181.7   6.169 5.19e-08 ***
## month3            189405.7    39016.8   4.854 8.11e-06 ***
## month4            137171.4    44248.7   3.100 0.002876 **
## month5            173484.8    43940.5   3.948 0.000199 ***
## month6            206825.1    38475.7   5.375 1.15e-06 ***
## month7             95704.1    37717.1   2.537 0.013614 *
## month8            172879.3    37006.9   4.672 1.58e-05 ***
## month9             83160.4    40649.5   2.046 0.044891 *
## month10           102923.0    40265.4   2.556 0.012969 *
## month11           171773.5    46293.8   3.711 0.000435 ***
## month12           115723.3   107500.7   1.076 0.285750
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66040 on 64 degrees of freedom
## Multiple R-squared:  0.5813, Adjusted R-squared:  0.4766
## F-statistic: 5.554 on 16 and 64 DF,  p-value: 3.221e-07
```

Check performance against test set

```r
prediction = predict(fit_lm, newdata = X_test)
```

```
## Warning in predict.lm(fit_lm, newdata = X_test): prediction from a
## rank-deficient fit may be misleading
```

```r
residuals <- y_test - prediction
squared_residuals <- residuals^2
```

```r
mse <- mean(squared_residuals)
rmse_lm <- sqrt(mse)
rmse_lm
```

```
## [1] 83085.05
```

```r
mae_lm <- mae(y_test, prediction)
mae_lm
```

```
## [1] 66191.32
```

Let's perform the same but with LASSO

```r
lasso_model <- cv.glmnet(X_train, y_train, alpha = 1)
coef(lasso_model)
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                            s1
## (Intercept)      1582568.16
## ...1                   .
## Fuel_Price         43058.39
## Unemployment      -28056.54
## Week_Number            .
## EventsLabour.Day   35377.77
## EventsNo_Holiday  -24103.64
## month2            136694.91
## month3             71550.01
## month4             21476.53
## month5             57557.03
## month6             89435.03
## month7                 .
## month8             56409.55
## month9                 .
## month10                .
## month11            40328.77
## month12                .
```

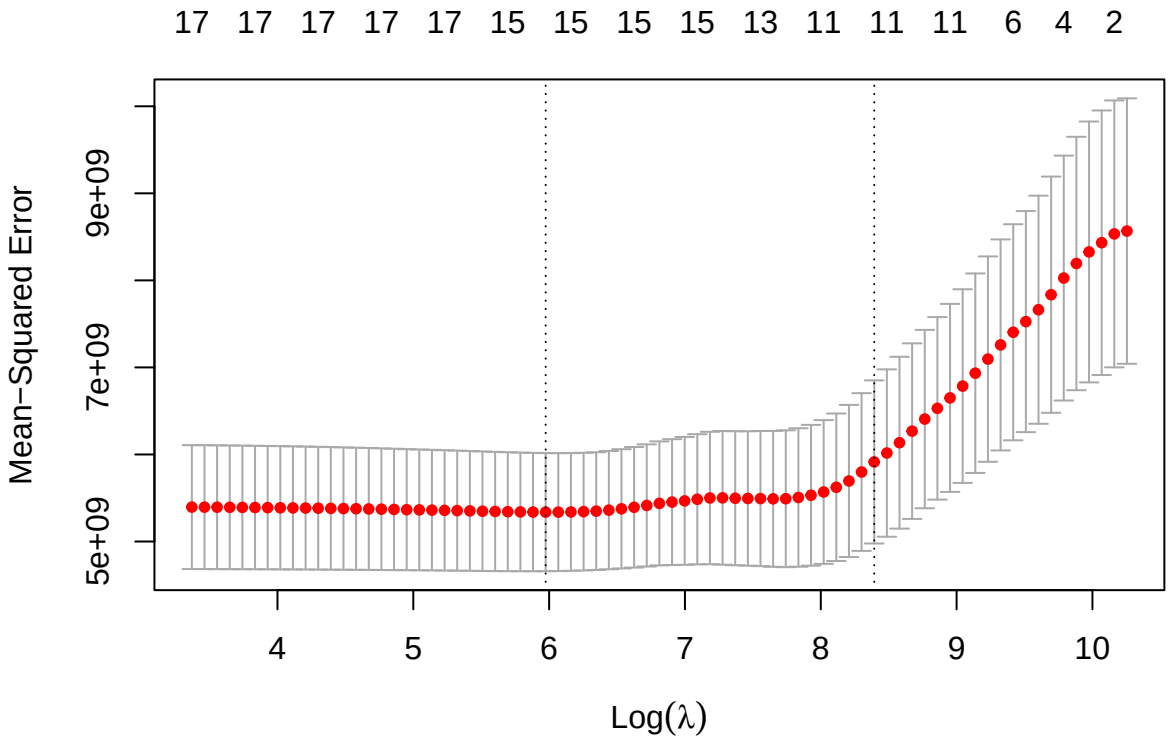
Prepare the test dataset

```r
X_test <- as.matrix(X_test)
y_test <- as.matrix(y_test)
```

```r
prediction <- predict(lasso_model, newx = X_test)
mae_lasso <- mae(y_test, prediction)
mae_lasso
```

```
## [1] 64508.42
```

```r
plot(lasso_model)
```



One interesting analysis we can perform is see if removing the features that we thought were promising from EDA will impact the performance of our model. To begin with, we can try removing the holidays information from our training dataset

```
X_train_rem_hol <- subset(X_train, select = -EventsNo_Holiday)
X_train_rem_hol <- as.matrix(X_train_rem_hol)
fit2_rem_hol <- lm(y_train ~ X_train_rem_hol)
summary(fit2_rem_hol)
```

```
##
## Call:
## lm(formula = y_train ~ X_train_rem_hol)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -127513  -46484    3252   44450  114445
##
## Coefficients: (1 not defined because of singularities)
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    1611391.9   335565.0   4.802 9.61e-06 ***
## X_train_rem_hol...1               -142.8      752.4  -0.190 0.850079
## X_train_rem_holFuel_Price        49730.0    47387.7   1.049 0.297868
## X_train_rem_holUnemployment      -50409.6    49891.1  -1.010 0.316054
## X_train_rem_holWeek_Number            NA         NA      NA       NA
## X_train_rem_holEventsLabour.Day  104943.3    46788.5   2.243 0.028317 *
## X_train_rem_holmonth2            264549.9    40108.1   6.596 8.92e-09 ***
## X_train_rem_holmonth3            189427.4    38753.4   4.888 6.99e-06 ***
```

```
## X_train_rem_holmonth4            137252.0    43949.6   3.123 0.002673 **
## X_train_rem_holmonth5            173545.3    43643.6   3.976 0.000178 ***
## X_train_rem_holmonth6            206829.1    38216.1   5.412 9.63e-07 ***
## X_train_rem_holmonth7             95709.2    37462.6   2.555 0.012976 *
## X_train_rem_holmonth8            172859.4    36757.2   4.703 1.38e-05 ***
## X_train_rem_holmonth9             83138.2    40375.2   2.059 0.043492 *
## X_train_rem_holmonth10           102888.9    39993.6   2.573 0.012385 *
## X_train_rem_holmonth11           171751.3    45981.4   3.735 0.000397 ***
## X_train_rem_holmonth12           141266.7    79263.7   1.782 0.079381 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65590 on 65 degrees of freedom
## Multiple R-squared:  0.5805, Adjusted R-squared:  0.4837
## F-statistic: 5.996 on 15 and 65 DF,  p-value: 1.34e-07
```

Looks like that one did not make too much of a difference

We remember seeing that the weekly sales depended on which month it was too

```
X_train_rem_months <- X_train[, !(colnames(X_train) %in% c("month2", "month3", "month4", "month5", "mon
X_train_rem_months <- as.matrix(X_train_rem_months)
fit3 <- lm(y_train ~ X_train_rem_months)
summary(fit3)
```

```
##
## Call:
## lm(formula = y_train ~ X_train_rem_months)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -201454  -50895    5791   50895  177016
##
## Coefficients: (1 not defined because of singularities)
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        2073420     364789   5.684 2.39e-07 ***
## X_train_rem_months...1               -1088        588  -1.849  0.06833 .
## X_train_rem_monthsFuel_Price        108014      38830   2.782  0.00684 **
## X_train_rem_monthsUnemployment      -95891      46082  -2.081  0.04086 *
## X_train_rem_monthsWeek_Number           NA         NA      NA       NA
## X_train_rem_monthsEventsLabour.Day  -42284      79735  -0.530  0.59747
## X_train_rem_monthsEventsNo_Holiday  -90290      64011  -1.411  0.16251
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 86220 on 75 degrees of freedom
## Multiple R-squared:  0.1636, Adjusted R-squared:  0.1078
## F-statistic: 2.933 on 5 and 75 DF,  p-value: 0.01793
```

We see that the adjusted R-squared value has dropped significantly. So our EDA has proved beneficial in keeping the month variable even though correlation did not show the same.

```
X_train_rem_fuel <- X_train[, !(colnames(X_train) %in% c("Fuel_Price"))]
X_train_rem_fuel <- as.matrix(X_train_rem_fuel)
fit4 <- lm(y_train ~ X_train_rem_fuel)
summary(fit4)
```

```
##
## Call:
## lm(formula = y_train ~ X_train_rem_fuel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -121140  -43029     580   49711  113047
##
## Coefficients: (1 not defined because of singularities)
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1463639.8   303299.6   4.826 8.80e-06 ***
## X_train_rem_fuel...1                568.7      324.8   1.751 0.084724 .
## X_train_rem_fuelUnemployment      -13948.7    36030.8  -0.387 0.699922
## X_train_rem_fuelWeek_Number            NA         NA      NA       NA
## X_train_rem_fuelEventsLabour.Day   84076.5    86284.5   0.974 0.333465
## X_train_rem_fuelEventsNo_Holiday  -25292.4    72413.9  -0.349 0.728012
## X_train_rem_fuelmonth2            263256.9    42110.8   6.252 3.56e-08 ***
## X_train_rem_fuelmonth3            202684.8    36911.1   5.491 7.10e-07 ***
## X_train_rem_fuelmonth4            160584.6    38171.8   4.207 8.10e-05 ***
## X_train_rem_fuelmonth5            196902.2    37811.2   5.208 2.11e-06 ***
## X_train_rem_fuelmonth6            214970.0    37702.7   5.702 3.13e-07 ***
## X_train_rem_fuelmonth7             97707.8    37694.3   2.592 0.011769 *
## X_train_rem_fuelmonth8            174773.2    36988.0   4.725 1.27e-05 ***
## X_train_rem_fuelmonth9             79758.8    40546.8   1.967 0.053445 .
## X_train_rem_fuelmonth10            99520.7    40161.1   2.478 0.015818 *
## X_train_rem_fuelmonth11           160962.9    45152.3   3.565 0.000689 ***
## X_train_rem_fuelmonth12            86674.4   103909.8   0.834 0.407262
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66080 on 65 degrees of freedom
## Multiple R-squared:  0.5742, Adjusted R-squared:  0.4759
## F-statistic: 5.843 on 15 and 65 DF,  p-value: 2.035e-07
```

```r
X_train_rem_emp <- X_train[, !(colnames(X_train) %in% c("Unemployment"))]
X_train_rem_emp <- as.matrix(X_train_rem_emp)
fit5 <- lm(y_train ~ X_train_rem_emp)
summary(fit5)
```

```
##
## Call:
## lm(formula = y_train ~ X_train_rem_emp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -122148  -44495    1932   48911  120700
##
## Coefficients: (1 not defined because of singularities)
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1310011.9   115954.4  11.298  < 2e-16 ***
## X_train_rem_emp...1                526.4      356.3   1.478 0.144353
## X_train_rem_empFuel_Price        16356.5    34202.0   0.478 0.634090
## X_train_rem_empWeek_Number            NA         NA      NA       NA
## X_train_rem_empEventsLabour.Day   85808.4    86110.3   0.996 0.322706
## X_train_rem_empEventsNo_Holiday  -25223.9    72369.6  -0.349 0.728560
```

```
## X_train_rem_empmonth2            261747.2    42158.2   6.209 4.22e-08 ***
## X_train_rem_empmonth3            196923.2    38297.4   5.142 2.70e-06 ***
## X_train_rem_empmonth4            156309.5    39951.2   3.913 0.000221 ***
## X_train_rem_empmonth5            191038.3    40328.7   4.737 1.22e-05 ***
## X_train_rem_empmonth6            212959.6    37992.5   5.605 4.56e-07 ***
## X_train_rem_empmonth7             96171.7    37717.5   2.550 0.013146 *
## X_train_rem_empmonth8            175167.3    36940.2   4.742 1.20e-05 ***
## X_train_rem_empmonth9             76205.6    40060.5   1.902 0.061569 .
## X_train_rem_empmonth10           101346.1    40238.4   2.519 0.014251 *
## X_train_rem_empmonth11           158355.6    44333.5   3.572 0.000673 ***
## X_train_rem_empmonth12            83629.5   102663.9   0.815 0.418279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66040 on 65 degrees of freedom
## Multiple R-squared:  0.5747, Adjusted R-squared:  0.4766
## F-statistic: 5.856 on 15 and 65 DF,  p-value: 1.967e-07
```

Random Forests

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
# Fit random forest model using the training data
rf_model <- randomForest(X_train, y_train, ntree = 500, mtry = 4)
```

```
## Warning in rfout$mse/(var(y) * (n - 1)/n): Recycling array of length 1 in vector-array arithmetic is
##   Use c() or as.vector() instead.
```

```
# View model summary
print(rf_model)
```

```
##
## Call:
##  randomForest(x = X_train, y = y_train, ntree = 500, mtry = 4)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 6861064662
##                     % Var explained: 16.63
```

RMSE

```
# Predict using the trained model on the test data
predictions <- predict(rf_model, X_test)
```

```r
# Calculate RMSE (if needed)
# Assuming 'y_test' contains the actual test set values
rmse_rf <- sqrt(mean((predictions - y_test)^2))
rmse_rf
```

```
## [1] 81681.82
```

```r
plot(rf_model)
```