## Walmart Sales Analysis

```
#Name: Team A: Megan, Nate Zane
#Course number IST 687
#Final Project
#Due 9/19/22, Submitted 9/19/22

# 2. We did this project with help from the book and the professor
# and these Internet sources: dplr.tidyverse.org,
#https://stackoverflow.com/questions/65596464/parsing-dates-with-different-formats-using-lubridate,
#https://shiny.rstudio.com/articles/layout-guide.html
#https://datascienceplus.com/building-a-simple-sales-revenue-dashboard-with-r-shiny-shinydashboard/
#https://www.programmingr.com/tutorial/left-join-in-r/#:~:text=A%20left%20join%20in%20R%20is%20a%20merge,do%20not%20already%20exist%20in%20the%20first%20table

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(tidyverse)

## ── Attaching packages
## ─────────────────────────────────────
## tidyverse 1.3.2 ──

## ✓ ggplot2 3.3.6     ✓ purrr   0.3.4
## ✓ tibble  3.1.8     ✓ stringr 1.4.1
## ✓ tidyr   1.2.0     ✓ forcats 0.5.2
## ✓ readr   2.1.2
## ── Conflicts ──────────────────────────────────────── tidyverse_conflict
s() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()

library(tidyr)
library(ggplot2)
library(ggpubr)
library(lubridate)
```

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

```
library(readxl)
```

```
#Read in files of data

#Read in files for Section 2
salesdataset <- read_csv("salesdataset.csv")
```

```
## Rows: 421570 Columns: 5
## ── Column specification ────────────────────────────────────────────────────
──────
## Delimiter: ","
## chr (1): Date
## dbl (3): Store, Dept, Weekly_Sales
## lgl (1): IsHoliday
##
## 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 m
essage.
```

```
head(salesdataset,1)
```

```
## # A tibble: 1 × 5
##   Store  Dept Date     Weekly_Sales IsHoliday
##   <dbl> <dbl> <chr>           <dbl> <lgl>
## 1     1     1 5/2/2010       24924. FALSE
```

```
#Read in additional file for Section 3 to combine with salesdataset
features <- read_csv("features.csv")
```

```
## Rows: 8190 Columns: 12
## ── Column specification ────────────────────────────────────────────────────
──────
## Delimiter: ","
## chr  (1): Date
## dbl (10): Store, Temperature, Fuel_Price, MarkDown1, MarkDown2, MarkDown3,
M...
## lgl  (1): IsHoliday
##
## 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 m
essage.
```

```
#head(features,1)
#dfSales <- read_csv("Walmart.csv")  Data set used before join was working.
```

```r
#Read in file for Section 1
#attach(stores_averages)
stores_averages <- read_excel("stores_averages.xlsx")
stores_averages <- data.frame(stores_averages)

#Fix the Date columns to consistent format
fix_Dates <- function(x, format = c("ymd", "dmy", "mdy")){
  fmt <- lubridate::guess_formats(x, format)
  fmt <- unique(fmt)
  y <- as.Date(x, format = fmt[1])
  for(i in seq_along(fmt)[-1]){
    na <- is.na(y)
    if(!any(na)) break
    y[na] <- as.Date(x[na], format = fmt[i])
  }
  y
}

#dfSales$Date <- fix_Dates(dfSales$Date)

salesdataset$Date <- fix_Dates(salesdataset$Date)
features$Date <- fix_Dates(features$Date)

#Combine the salesdataset and features dataframes for Section 3
salesdataset <- data.frame(salesdataset)
features <- data.frame(features)

#Check out the data and replace the NAs
combinedSalesdf <- right_join(salesdataset, features, by=c('Store','Date'))
#sum(!is.na(combinedSalesdf$Weekly_Sales))
combinedSalesdf <- replace(combinedSalesdf,is.na(combinedSalesdf), 0)
glimpse(combinedSalesdf)
```

```
## Rows: 286,854
## Columns: 15
## $ Store        <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, …
## $ Dept         <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, …
## $ Date         <date> 2010-02-19, 2010-02-26, 2010-03-12, 2010-03-19, 2010
-03-…
## $ Weekly_Sales <dbl> 41595.55, 19403.54, 21043.39, 22136.64, 26229.21, 175
96.9…
## $ IsHoliday.x  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, …
## $ Temperature  <dbl> 39.93, 46.63, 49.27, 54.58, 51.45, 66.32, 64.84, 67.4
1, 7…
## $ Fuel_Price   <dbl> 2.514, 2.561, 2.708, 2.720, 2.732, 2.808, 2.795, 2.78
0, 2…
```

```
## $ MarkDown1    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, …
## $ MarkDown2    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, …
## $ MarkDown3    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, …
## $ MarkDown4    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, …
## $ MarkDown5    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, …
## $ CPI          <dbl> 211.2891, 211.3196, 211.6072, 211.2156, 211.0180, 210
.488…
## $ Unemployment <dbl> 8.106, 8.106, 7.838, 8.106, 8.106, 7.808, 7.808, 7.80
8, 7…
## $ IsHoliday.y  <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALS
E, F…
```

```
#head(combinedSalesdf, 1000)

#Total Weekly_Sales is low; I suspect rows fell out in the join because of
#imperfect formatting of the dates in the date fix function
sum(!is.na(combinedSalesdf$Weekly_Sales))
```

```
## [1] 286854
```

```
#Save DF for Section 3
dfSales <- combinedSalesdf
```

##########SECTION 1###############

#How does weather affect the sales of 45 different stores. Summary data - top 5 max hot and cold, top 5 min hot and cold, and Average weekly sales data per #store; Time series scatter plot or line graph #Conclusion business question - which stores should be more active with sales plans to be profitable in different temperatures?

```
#Averages for the graphs
 stores_averages <- data.frame(stores_averages)
 stores_averages_hot <- stores_averages[order(stores_averages$ave_temp, decre
asing = TRUE), ]
 stores_averages_hot
```

```
##    store average_sales_dept average_sales_store ave_temp
## 33    33            5728.41            567113.0       75
## 10    10           26332.30           2606898.1       71
## 11    11           19276.76           1908399.5       71
## 42    42           11443.37           1132893.6       71
## 3      3            6373.03            630930.4       70
## 36    36            8584.41            849856.8       70
## 37    37           10297.36           1019438.2       70
## 39    39           21000.76           2079075.6       70
## 6      6           21913.24           2169411.1       69
```

```
## 12    12           14867.31           1471863.6       69
## 28    28           18714.45           1852730.4       69
## 38    38            7492.48            741755.4       69
## 5      5            5053.42            500288.2       68
## 43    43           13415.11           1328096.3       68
## 1      1           21710.54           2149343.8       67
## 2      2           26898.07           2662908.9       67
## 21    21           11283.44           1117060.1       67
## 30    30            8764.24            867659.5       67
## 31    31           19681.91           1948508.8       67
## 9      9            8772.89            868516.2       66
## 4      4           29161.21           2886959.8       61
## 8      8           13133.01           1300168.5       61
## 34    34           13522.08           1338686.1       58
## 14    14           28784.85           2849700.3       56
## 27    27           24826.98           2457871.5       56
## 35    35           13803.60           1366556.1       56
## 45    45           11662.90           1154626.8       56
## 20    20           29508.30           2921321.9       54
## 22    22           15181.22           1502940.7       53
## 24    24           18969.11           1877941.5       53
## 29    29            8158.81            807722.2       53
## 13    13           27355.14           2708158.5       52
## 18    18           15733.31           1557598.0       52
## 44    44            6038.93            597854.1       52
## 19    19           20362.13           2015850.6       51
## 25    25           10308.16           1020507.6       51
## 32    32           16351.62           1618810.6       51
## 15    15            9002.49            891246.8       50
## 23    23           19776.18           1957841.9       47
## 41    41           17976.00           1779624.5       47
## 40    40           13763.63           1362599.6       46
## 17    17           12954.39           1282485.0       45
## 16    16            7863.22            778459.2       43
## 26    26           14554.13           1440858.8       42
## 7      7            8358.77            827517.8       38
```

```
 head(stores_averages_hot)
```

```
##    store average_sales_dept average_sales_store ave_temp
## 33    33            5728.41            567113.0       75
## 10    10           26332.30           2606898.1       71
## 11    11           19276.76           1908399.5       71
## 42    42           11443.37           1132893.6       71
## 3      3            6373.03            630930.4       70
## 36    36            8584.41            849856.8       70
```

```
 tail(stores_averages_hot)
```

```
##    store average_sales_dept average_sales_store ave_temp
## 41    41           17976.00           1779624.5       47
```

```
## 40    40           13763.63           1362599.6       46
## 17    17           12954.39           1282485.0       45
## 16    16            7863.22            778459.2       43
## 26    26           14554.13           1440858.8       42
## 7      7            8358.77            827517.8       38
```

```
 stores_averages_cold <- stores_averages[order(stores_averages$ave_temp, decr
easing = FALSE), ]
 head(stores_averages_cold)
```

```
##    store average_sales_dept average_sales_store ave_temp
## 7      7            8358.77            827517.8       38
## 26    26           14554.13           1440858.8       42
## 16    16            7863.22            778459.2       43
## 17    17           12954.39           1282485.0       45
## 40    40           13763.63           1362599.6       46
## 23    23           19776.18           1957841.9       47
```

```
 head(stores_averages_cold, 5)
```

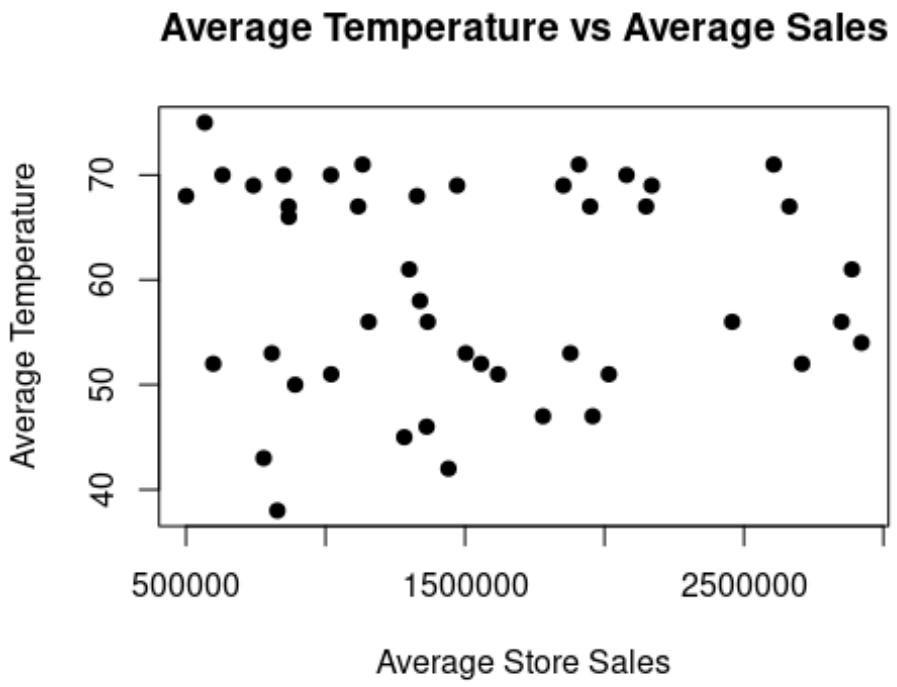
```
##    store average_sales_dept average_sales_store ave_temp
## 7      7            8358.77            827517.8       38
## 26    26           14554.13           1440858.8       42
## 16    16            7863.22            778459.2       43
## 17    17           12954.39           1282485.0       45
## 40    40           13763.63           1362599.6       46
```

```
 stores_averages_cold <- head(stores_averages_cold, 5)
 stores_averages_hot <- head(stores_averages_hot, 5)
 #head(stores_averages$ave_temp, stores_averages$average_sales_store,5)
 head(stores_averages$ave_temp)
```

```
## [1] 67 67 70 61 68 69
```

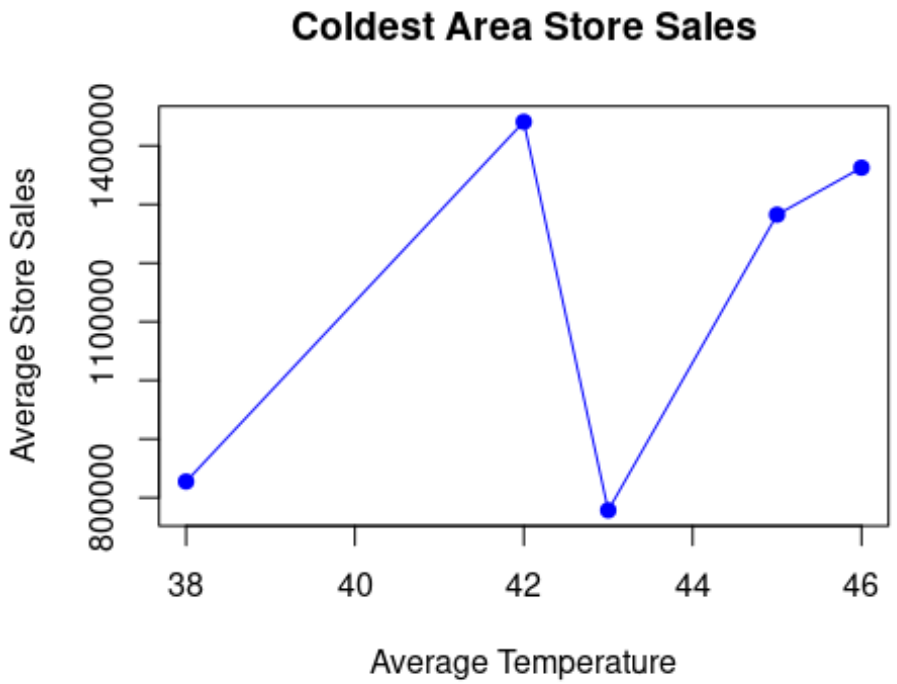
```
 head_hot <-head(stores_averages$ave_temp)
 head_hot <-head(stores_averages_hot$ave_temp,5)
 head_cold <- head(stores_averages_cold$ave_temp, 5)
 hot_sales <- head(stores_averages_hot$average_sales_store,5)
 cold_sales <- head(stores_averages_cold$average_sales_store, 5)
```

```
#Scatterplots for top hot and top cold sales
#plot(average_sales_store, ave_temp, main="Average Temperature vs Average Sal
es", xlab="Average Store Sales", ylab="Average Temperature", pch=19)
#glimpse(stores_averages)
plot(stores_averages$average_sales_store,stores_averages$ave_temp, main="Aver
age Temperature vs Average Sales", xlab="Average Store Sales", ylab="Average
Temperature", pch=19)
```




```
plot(head_hot, hot_sales, type = "o", col = "red", main="Hottest Area Store S
ales", xlab="Average Temperature", ylab="Average Store Sales", pch=19)
```

```
plot(head_cold, cold_sales, type = "o", col = "blue", main="Coldest Area Stor
e Sales", xlab="Average Temperature", ylab="Average Store Sales", pch=19)
```




##########SECTION 2############## #

#How does weather affect the 99 departments and when which department becomes more profitable.Summary information - average sales per department #Create a line graph showing the sales trends per department based on temperature.

#Histogram of sales (spread) #Top 5 performers in the hot and the cold - bar charts #Worst 5 Performers in the hot and the cold #Conclusion business question - which departments should have deals or be moved to different locations to #improve overall store sales weekly?

```
#I reviewed the information within the salesdataset
head(salesdataset)
## # A tibble: 6 × 5
##   Store  Dept Date       Weekly_Sales IsHoliday
##   <dbl> <dbl> <chr>             <dbl> <lgl>
## 1     1     1 05/02/2010       24924. FALSE
## 2     1     1 12/02/2010       46039. TRUE
## 3     1     1 19/02/2010       41596. FALSE
## 4     1     1 26/02/2010       19404. FALSE
## 5     1     1 05/03/2010       21828. FALSE
## 6     1     1 12/03/2010       21043. FALSE
#the next step was to create a function that would calculate the totat sales
per department based off of the department number.
```

```
DepartmentSales <- aggregate(x=salesdataset$Weekly_Sales, by=
list(salesdataset$Dept),FUN=sum)
#I created a new data frame with only the date.

Temperature<- data.frame(salesdataset$Date)
#I renamed the column to Temp.
colnames(Temperature)[1]<-"Temp"
#review the information to see the data in the table.
head(Temperature)
##         Temp
## 1 05/02/2010
## 2 12/02/2010
## 3 19/02/2010
## 4 26/02/2010
## 5 05/03/2010
## 6 12/03/2010
#I then reviewed the strings to see what type the column was this is
important for the transfer function later written.
str(Temperature)
## 'data.frame':    420251 obs. of  1 variable:
##  $ Temp: chr  "05/02/2010" "12/02/2010" "19/02/2010" "26/02/2010" ...
#I reviewed the data table with the Temperature.
#this is where I would review and connect the temperature to the date to make
sure that the right temperature was at the correct place.
head(dfSales)
## # A tibble: 6 × 8
##   Store Date       Weekly_Sales Holiday_Flag Temperature Fuel_Pr…¹   CPI
Unemp…²
##   <dbl> <chr>             <dbl>        <dbl>       <dbl>     <dbl>
<dbl>   <dbl>
## 1     1
5/2/2010       1643691.            0        42.3      2.57  211.    8.11
## 2     1
12/2/2010      1641957.            1        38.5      2.55  211.    8.11
## 3     1 19-02-
2010     1611968.            0        39.9      2.51  211.    8.11
## 4     1 26-02-
2010     1409728.            0        46.6      2.56  211.    8.11
## 5     1
5/3/2010       1554807.            0        46.5      2.62  211.    8.11
## 6     1
12/3/2010      1439542.            0        57.8      2.67  211.    8.11
## # … with abbreviated variable names ¹Fuel_Price, ²Unemployment
str(dfSales)
## spec_tbl_df [6,435 × 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Store       : num [1:6435] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date        : chr [1:6435] "5/2/2010" "12/2/2010" "19-02-2010" "26-02-
2010" ...
##  $ Weekly_Sales: num [1:6435] 1643691 1641957 1611968 1409728 1554807 ...
##  $ Holiday_Flag: num [1:6435] 0 1 0 0 0 0 0 0 0 0 ...
```

```
##  $ Temperature : num [1:6435] 42.3 38.5 39.9 46.6 46.5 ...
##  $ Fuel_Price  : num [1:6435] 2.57 2.55 2.51 2.56 2.62 ...
##  $ CPI         : num [1:6435] 211 211 211 211 211 ...
##  $ Unemployment: num [1:6435] 8.11 8.11 8.11 8.11 8.11 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Store = col_double(),
##   ..   Date = col_character(),
##   ..   Weekly_Sales = col_double(),
##   ..   Holiday_Flag = col_double(),
##   ..   Temperature = col_double(),
##   ..   Fuel_Price = col_double(),
##   ..   CPI = col_double(),
##   ..   Unemployment = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
#Then I wrote codes to connect temperature to a specific date.
#alot of scrubbing.

Temperature$Temp[Temperature$Temp=='05/02/2010'] <- '42.31'
Temperature$Temp[Temperature$Temp=='12/02/2010'] <- '38.51'
Temperature$Temp[Temperature$Temp=='19/02/2010'] <- '39.93'
Temperature$Temp[Temperature$Temp=='26/02/2010'] <- '46.63'
Temperature$Temp[Temperature$Temp=='05/03/2010'] <- '46.50'
Temperature$Temp[Temperature$Temp=='12/03/2010'] <- '57.79'
Temperature$Temp[Temperature$Temp=='19/03/2010'] <- '54.58'
Temperature$Temp[Temperature$Temp=='26/03/2010'] <- '51.45'
Temperature$Temp[Temperature$Temp=='02/04/2010'] <- '62.27'
Temperature$Temp[Temperature$Temp=='09/04/2010'] <- '65.86'
Temperature$Temp[Temperature$Temp=='16/04/2010'] <- '66.32'
Temperature$Temp[Temperature$Temp=='23/04/2010'] <- '64.84'
Temperature$Temp[Temperature$Temp=='30/04/2010'] <- '67.41'
Temperature$Temp[Temperature$Temp=='07/05/2010'] <- '72.55'
Temperature$Temp[Temperature$Temp=='14/05/2010'] <- '74.78'
Temperature$Temp[Temperature$Temp=='21/05/2010'] <- '76.44'
Temperature$Temp[Temperature$Temp=='28/05/2010'] <- '80.44'
Temperature$Temp[Temperature$Temp=='04/06/2010'] <- '80.69'
Temperature$Temp[Temperature$Temp=='11/06/2010'] <- '80.43'
Temperature$Temp[Temperature$Temp=='18/06/2010'] <- '84.11'
Temperature$Temp[Temperature$Temp=='25/06/2010'] <- '84.34'
Temperature$Temp[Temperature$Temp=='02/07/2010'] <- '80.91'
Temperature$Temp[Temperature$Temp=='09/07/2010'] <- '80.48'
Temperature$Temp[Temperature$Temp=='16/07/2010'] <- '83.15'
Temperature$Temp[Temperature$Temp=='23/07/2010'] <- '83.36'
Temperature$Temp[Temperature$Temp=='30/07/2010'] <- '81.84'
Temperature$Temp[Temperature$Temp=='06/08/2010'] <- '87.16'
Temperature$Temp[Temperature$Temp=='13/08/2010'] <- '87'
Temperature$Temp[Temperature$Temp=='20/08/2010'] <- '86.65'
Temperature$Temp[Temperature$Temp=='27/08/2010'] <- '85.22'
Temperature$Temp[Temperature$Temp=='03/09/2010'] <- '81.21'
```

```
Temperature$Temp[Temperature$Temp=='10/09/2010'] <- '78.69'
Temperature$Temp[Temperature$Temp=='17/09/2010'] <- '82.11'
Temperature$Temp[Temperature$Temp=='24/09/2010'] <- '80.94'
Temperature$Temp[Temperature$Temp=='01/10/2010'] <- '71.89'
Temperature$Temp[Temperature$Temp=='08/10/2010'] <- '63.93'
Temperature$Temp[Temperature$Temp=='15/10/2010'] <- '67.18'
Temperature$Temp[Temperature$Temp=='22/10/2010'] <- '69.86'
Temperature$Temp[Temperature$Temp=='29/10/2010'] <- '69.64'
Temperature$Temp[Temperature$Temp=='05/11/2010'] <- '58.74'
Temperature$Temp[Temperature$Temp=='12/11/2010'] <- '59.61'
Temperature$Temp[Temperature$Temp=='19/11/2010'] <- '51.41'
Temperature$Temp[Temperature$Temp=='26/11/2010'] <- '64.52'
Temperature$Temp[Temperature$Temp=='03/12/2010'] <- '49.27'
Temperature$Temp[Temperature$Temp=='10/12/2010'] <- '46.33'
Temperature$Temp[Temperature$Temp=='17/12/2010'] <- '49.84'
Temperature$Temp[Temperature$Temp=='24/12/2010'] <- '52.33'
Temperature$Temp[Temperature$Temp=='31/12/2010'] <- '48.43'
Temperature$Temp[Temperature$Temp=='07/01/2011'] <- '48.27'
Temperature$Temp[Temperature$Temp=='14/01/2011'] <- '35.4'
Temperature$Temp[Temperature$Temp=='21/01/2011'] <- '44.04'
Temperature$Temp[Temperature$Temp=='28/01/2011'] <- '43.83'
Temperature$Temp[Temperature$Temp=='04/02/2011'] <- '42.27'
Temperature$Temp[Temperature$Temp=='11/02/2011'] <- '36.39'
Temperature$Temp[Temperature$Temp=='18/02/2011'] <- '57.36'
Temperature$Temp[Temperature$Temp=='25/02/2011'] <- '62.9'
Temperature$Temp[Temperature$Temp=='04/03/2011'] <- '59.58'
Temperature$Temp[Temperature$Temp=='11/03/2011'] <- '53.56'
Temperature$Temp[Temperature$Temp=='18/03/2011'] <- '62.76'
Temperature$Temp[Temperature$Temp=='25/03/2011'] <- '69.97'
Temperature$Temp[Temperature$Temp=='01/04/2011'] <- '59.17'
Temperature$Temp[Temperature$Temp=='08/04/2011'] <- '67.84'
Temperature$Temp[Temperature$Temp=='15/04/2011'] <- '71.27'
Temperature$Temp[Temperature$Temp=='22/04/2011'] <- '72.99'
Temperature$Temp[Temperature$Temp=='29/04/2011'] <- '72.03'
Temperature$Temp[Temperature$Temp=='06/05/2011'] <- '64.61'
Temperature$Temp[Temperature$Temp=='13/05/2011'] <- '75.64'
Temperature$Temp[Temperature$Temp=='20/05/2011'] <- '67.63'
Temperature$Temp[Temperature$Temp=='27/05/2011'] <- '77.72'
Temperature$Temp[Temperature$Temp=='03/06/2011'] <- '83'
Temperature$Temp[Temperature$Temp=='10/06/2011'] <- '83.13'
Temperature$Temp[Temperature$Temp=='17/06/2011'] <- '86.41'
Temperature$Temp[Temperature$Temp=='24/06/2011'] <- '83.58'
Temperature$Temp[Temperature$Temp=='01/07/2011'] <- '85.55'
Temperature$Temp[Temperature$Temp=='08/07/2011'] <- '85.83'
Temperature$Temp[Temperature$Temp=='15/07/2011'] <- '88.54'
Temperature$Temp[Temperature$Temp=='22/07/2011'] <- '85.77'
Temperature$Temp[Temperature$Temp=='29/07/2011'] <- '86.83'
Temperature$Temp[Temperature$Temp=='05/08/2011'] <- '91.65'
Temperature$Temp[Temperature$Temp=='12/08/2011'] <- '90.76'
Temperature$Temp[Temperature$Temp=='19/08/2011'] <- '89.94'
```

```
Temperature$Temp[Temperature$Temp=='26/08/2011'] <- '87.96'
Temperature$Temp[Temperature$Temp=='02/09/2011'] <- '87.83'
Temperature$Temp[Temperature$Temp=='09/09/2011'] <- '76'
Temperature$Temp[Temperature$Temp=='16/09/2011'] <- '79.94'
Temperature$Temp[Temperature$Temp=='23/09/2011'] <- '75.8'
Temperature$Temp[Temperature$Temp=='30/09/2011'] <- '79.69'
Temperature$Temp[Temperature$Temp=='07/10/2011'] <- '69.31'
Temperature$Temp[Temperature$Temp=='14/10/2011'] <- '71.74'
Temperature$Temp[Temperature$Temp=='21/10/2011'] <- '63.71'
Temperature$Temp[Temperature$Temp=='28/10/2011'] <- '66.57'
Temperature$Temp[Temperature$Temp=='04/11/2011'] <- '54.98'
Temperature$Temp[Temperature$Temp=='11/11/2011'] <- '59.11'
Temperature$Temp[Temperature$Temp=='18/11/2011'] <- '62.25'
Temperature$Temp[Temperature$Temp=='25/11/2011'] <- '60.14'
Temperature$Temp[Temperature$Temp=='02/12/2011'] <- '48.91'
Temperature$Temp[Temperature$Temp=='09/12/2011'] <- '43.93'
Temperature$Temp[Temperature$Temp=='16/12/2011'] <- '51.63'
Temperature$Temp[Temperature$Temp=='23/12/2011'] <- '47.96'
Temperature$Temp[Temperature$Temp=='30/12/2011'] <- '44.55'
Temperature$Temp[Temperature$Temp=='06/01/2012'] <- '49.01'
Temperature$Temp[Temperature$Temp=='13/01/2012'] <- '48.53'
Temperature$Temp[Temperature$Temp=='20/01/2012'] <- '54.11'
Temperature$Temp[Temperature$Temp=='27/01/2012'] <- '54.26'
Temperature$Temp[Temperature$Temp=='03/02/2012'] <- '56.55'
Temperature$Temp[Temperature$Temp=='10/02/2012'] <- '48.02'
Temperature$Temp[Temperature$Temp=='17/02/2012'] <- '45.32'
Temperature$Temp[Temperature$Temp=='24/02/2012'] <- '57.25'
Temperature$Temp[Temperature$Temp=='02/03/2012'] <- '60.96'
Temperature$Temp[Temperature$Temp=='09/03/2012'] <- '58.76'
Temperature$Temp[Temperature$Temp=='16/03/2012'] <- '64.74'
Temperature$Temp[Temperature$Temp=='23/03/2012'] <- '65.93'
Temperature$Temp[Temperature$Temp=='30/03/2012'] <- '67.61'
Temperature$Temp[Temperature$Temp=='06/04/2012'] <- '70.43'
Temperature$Temp[Temperature$Temp=='13/04/2012'] <- '69.07'
Temperature$Temp[Temperature$Temp=='20/04/2012'] <- '66.76'
Temperature$Temp[Temperature$Temp=='27/04/2012'] <- '67.23'
Temperature$Temp[Temperature$Temp=='04/05/2012'] <- '75.55'
Temperature$Temp[Temperature$Temp=='11/05/2012'] <- '73.77'
Temperature$Temp[Temperature$Temp=='18/05/2012'] <- '70.33'
Temperature$Temp[Temperature$Temp=='25/05/2012'] <- '77.22'
Temperature$Temp[Temperature$Temp=='01/06/2012'] <- '77.95'
Temperature$Temp[Temperature$Temp=='08/06/2012'] <- '78.3'
Temperature$Temp[Temperature$Temp=='15/06/2012'] <- '79.35'
Temperature$Temp[Temperature$Temp=='22/06/2012'] <- '78.39'
Temperature$Temp[Temperature$Temp=='29/06/2012'] <- '84.88'
Temperature$Temp[Temperature$Temp=='29/06/2012'] <- '81.57'
Temperature$Temp[Temperature$Temp=='13/07/2012'] <- '77.12'
Temperature$Temp[Temperature$Temp=='20/07/2012'] <- '42.31'
Temperature$Temp[Temperature$Temp=='27/07/2012'] <- '42.31'
Temperature$Temp[Temperature$Temp=='03/08/2012'] <- '80.42'
```

```
Temperature$Temp[Temperature$Temp=='10/08/2012'] <- '82.66'
Temperature$Temp[Temperature$Temp=='17/08/2012'] <- '86.11'
Temperature$Temp[Temperature$Temp=='24/08/2012'] <- '85.05'
Temperature$Temp[Temperature$Temp=='31/08/2012'] <- '84.85'
Temperature$Temp[Temperature$Temp=='07/09/2012'] <- '77.66'
Temperature$Temp[Temperature$Temp=='14/09/2012'] <- '80.49'
Temperature$Temp[Temperature$Temp=='21/09/2012'] <- '83.96'
Temperature$Temp[Temperature$Temp=='28/09/2012'] <- '74.97'
Temperature$Temp[Temperature$Temp=='05/10/2012'] <- '69.87'
Temperature$Temp[Temperature$Temp=='12/10/2012'] <- '76.08'
Temperature$Temp[Temperature$Temp=='19/10/2012'] <- '68.55'
Temperature$Temp[Temperature$Temp=='26/10/2012'] <- '62.99'
Temperature$Temp[Temperature$Temp=='06/07/2012'] <- '62.99'
#I reviewed to make sure the temperature correctly changed and that all dates
were replaced with a Temperature.
head(Temperature)
##    Temp
## 1 42.31
## 2 38.51
## 3 39.93
## 4 46.63
## 5 46.50
## 6 57.79
# I changed the character string back to a numerical value.
Temperature$Temp <- as.numeric(Temperature$Temp)
# I pulled the Temperature data frame and the Department sales into the same
dataframe.
DeptSalesWeather<-data.frame(salesdataset,Temperature)
# I reviewed to make sure All information moved over and had the correct
character strings.
head(DeptSalesWeather)
##   Store Dept       Date Weekly_Sales IsHoliday  Temp
## 1     1    1 05/02/2010     24924.50     FALSE 42.31
## 2     1    1 12/02/2010     46039.49      TRUE 38.51
## 3     1    1 19/02/2010     41595.55     FALSE 39.93
## 4     1    1 26/02/2010     19403.54     FALSE 46.63
## 5     1    1 05/03/2010     21827.90     FALSE 46.50
## 6     1    1 12/03/2010     21043.39     FALSE 57.79
str(DeptSalesWeather)
## 'data.frame':    420251 obs. of  6 variables:
##  $ Store       : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Dept        : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Date        : chr  "05/02/2010" "12/02/2010" "19/02/2010" "26/02/2010"
...
##  $ Weekly_Sales: num  24924 46039 41596 19404 21828 ...
##  $ IsHoliday   : logi  FALSE TRUE FALSE FALSE FALSE FALSE ...
##  $ Temp        : num  42.3 38.5 39.9 46.6 46.5 ...
# I ran two filters to cut the departments sales in half by temperature over
and below 60 degress.
SalesVsTempAdv60 <- filter(DeptSalesWeather,Temp > 60)
```

```r
SalesVsTempBlw60 <- filter(DeptSalesWeather,Temp < 60)

#Then I ran codes to breakdown the most sales per department in both cold and
warm weather.
BestSalesAbv60 <- aggregate(x=SalesVsTempAdv60$Weekly_Sales, by=
list(SalesVsTempAdv60$Dept),FUN=sum)

BestSalesAbv60
##    Group.1            x
## 1        1  79953870.38
## 2        2 195030853.05
## 3        3  55627308.79
## 4        4 116243929.82
## 5        5  81972591.65
## 6        6  19032285.54
## 7        7  97754749.84
## 8        8 133251779.70
## 9        9  98390070.41
## 10      10  82754499.65
## 11      11  65383327.77
## 12      12  19898471.99
## 13      13 138041338.31
## 14      14  62092373.18
## 15      16  79678173.39
## 16      17  43103482.08
## 17      18  14526946.12
## 18      19   4601752.87
## 19      20  23735306.55
## 20      21  21681483.13
## 21      22  39321954.77
## 22      23  94134551.40
## 23      24  23620204.78
## 24      25  40214385.71
## 25      26  29741670.18
## 26      27   5339758.42
## 27      28   2327769.85
## 28      29  19518533.45
## 29      30  15636076.65
## 30      31   9700116.45
## 31      32  25557955.65
## 32      33  23880125.73
## 33      34  59390907.92
## 34      35  11077207.68
## 35      36   7914638.29
## 36      37   5475163.90
## 37      38 268410946.64
## 38      39       165.08
## 39      40 197262558.11
## 40      41   7275884.94
```

```
## 41      42  23610634.08
## 42      43         8.50
## 43      44  16996708.64
## 44      45     30194.78
## 45      46  87958596.72
## 46      47     -7885.27
## 47      48   1570012.12
## 48      49  25016135.42
## 49      50   2864371.19
## 50      51     20727.78
## 51      52   8219679.01
## 52      54    339853.18
## 53      55  34338743.45
## 54      56  19605296.00
## 55      58  11368725.84
## 56      59   1543228.18
## 57      60   1315319.78
## 58      65   4619558.65
## 59      67  31899644.60
## 60      71  20203885.29
## 61      72 194933288.35
## 62      74  61696809.57
## 63      77     30966.36
## 64      78      1059.88
## 65      79  97207288.98
## 66      80  49724328.50
## 67      81  69554416.85
## 68      82  64694104.46
## 69      83  13379685.15
## 70      85   9085838.92
## 71      87  58287597.18
## 72      90 196782774.98
## 73      91 147268162.49
## 74      92 324088079.75
## 75      93 110698160.31
## 76      94 134539218.43
## 77      95 314178662.68
## 78      96  53005656.45
## 79      97  62082245.68
## 80      98  27024507.88
## 81      99    138929.09
BestSalesBlw60 <- aggregate(x=SalesVsTempBlw60$Weekly_Sales, by=
list(SalesVsTempBlw60$Dept),FUN=sum)

BestSalesBlw60
##    Group.1           x
## 1        1  43684906.16
## 2        2  85580321.38
## 3        3  20265141.16
## 4        4  50902815.76
```

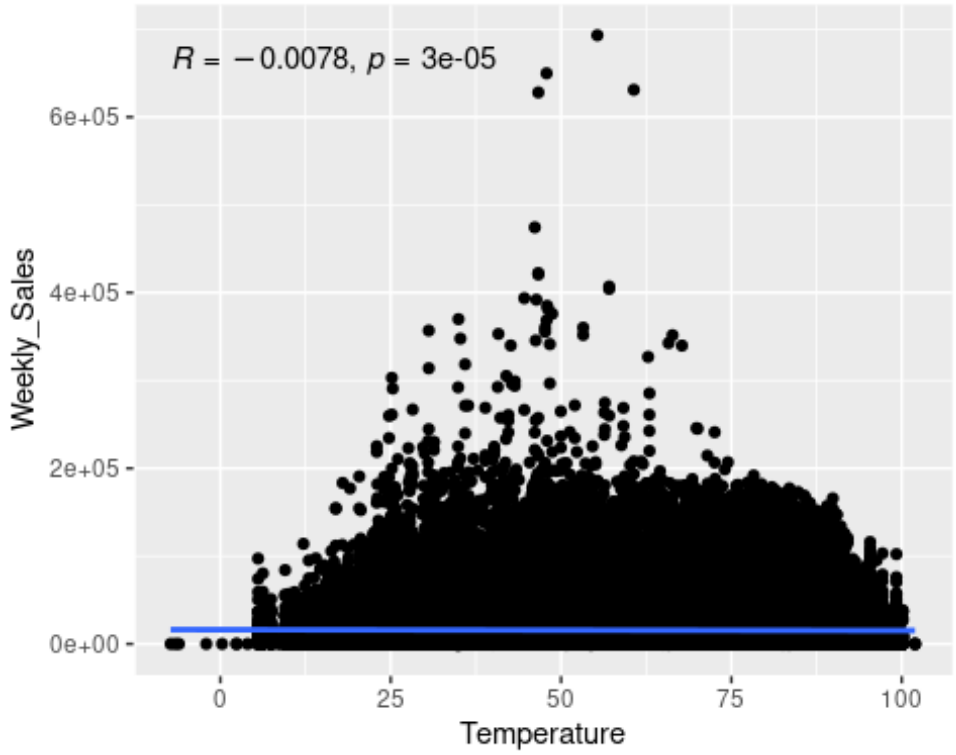
```
## 5        5  53634766.92
## 6        6   9388381.60
## 7        7  57722812.91
## 8        8  61029001.03
## 9        9  30003186.24
## 10      10  35142887.93
## 11      11  27945948.61
## 12      12   6861647.52
## 13      13  59280231.64
## 14      14  33602293.27
## 15      16  11992508.88
## 16      17  22216334.55
## 17      18  22370542.94
## 18      19   2214430.24
## 19      20   9614338.56
## 20      21  10383161.95
## 21      22  20401152.59
## 22      23  44045779.38
## 23      24  10727381.48
## 24      25  16896832.91
## 25      26  14158339.42
## 26      27   3557578.17
## 27      28   1457383.40
## 28      29  10491068.97
## 29      30   6153304.78
## 30      31   4493267.77
## 31      32  15141005.16
## 32      33  11003844.20
## 33      34  23832717.51
## 34      35   4395567.40
## 35      36   2794875.48
## 36      37   2542079.45
## 37      38 124707190.28
## 38      39        12.90
## 39      40  91673463.94
## 40      41   3324380.13
## 41      42   9608657.49
## 42      43         5.82
## 43      44   8452904.32
## 44      45     14742.85
## 45      46  40385814.72
## 46      47      2922.34
## 47      48    772792.49
## 48      49  11556766.29
## 49      50   1288825.94
## 50      51      9845.05
## 51      52   3867257.98
## 52      54    176441.45
## 53      55  25631711.90
## 54      56   2906226.87
```

```
## 55      58   4887037.73
## 56      59   2697166.34
## 57      60    689701.18
## 58      65   1878605.34
## 59      67  15981133.08
## 60      71  10179137.60
## 61      72 110791863.86
## 62      74  27085176.00
## 63      77     18377.91
## 64      78       654.83
## 65      79  44189255.16
## 66      80  22049731.70
## 67      81  29997612.46
## 68      82  34852975.87
## 69      83   6609145.69
## 70      85   4533123.37
## 71      87  26570566.58
## 72      90  90900301.66
## 73      91  67133747.63
## 74      92 152973258.74
## 75      93  48610309.20
## 76      94  54878735.44
## 77      95 127577348.01
## 78      96  20828253.77
## 79      97  26489490.67
## 80      98  12728644.22
## 81      99    219220.76
```

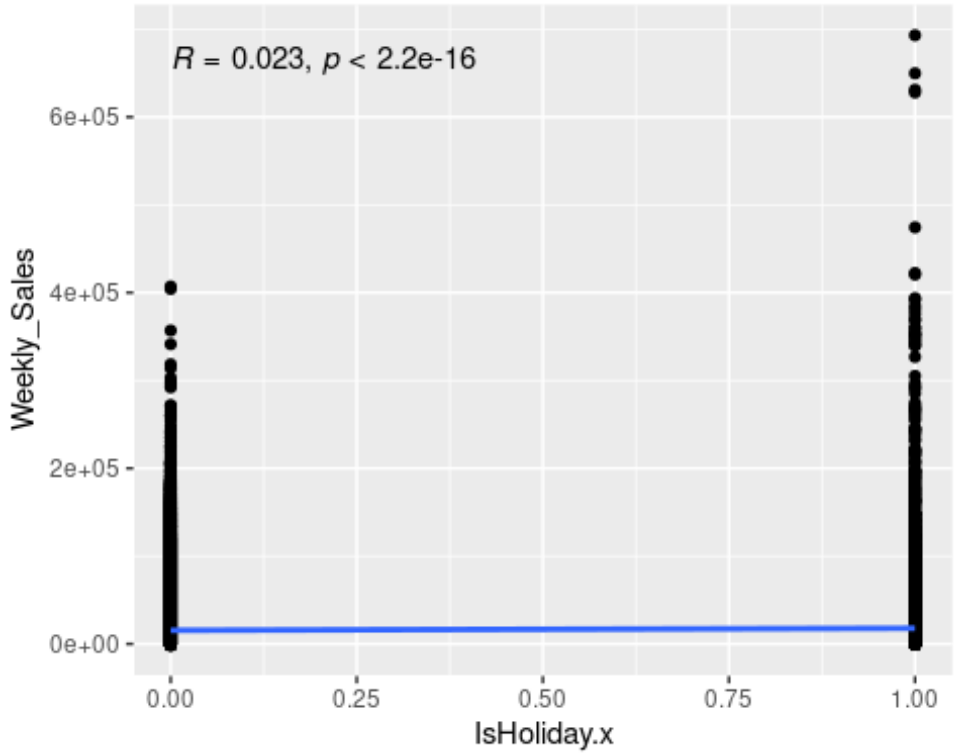
##########SECTION 3###############

#How do various inputs affect weekly sales overall. Compare the weekly sales to CPI, unemployment, and Fuel_price to see if the temperature is a good predictor and if there might be another metric that #should be considered instead of temperature.

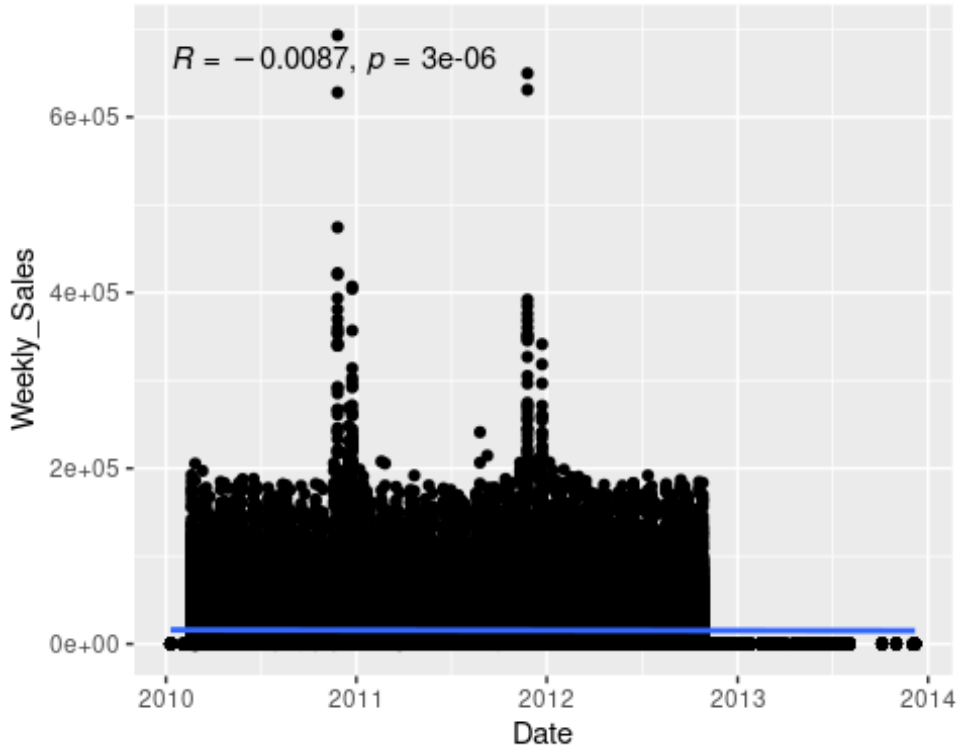
```
#glimpse(dfSales)

#Correlation of Temperature and Weekly_Sales
#Temp and Weekly Sales Bivariate Scatterplot
ggplot(dfSales, aes(x=Temperature, y=Weekly_Sales)) + geom_point()+ geom_smoo
th(method="lm") + stat_cor(method = "pearson")

## `geom_smooth()` using formula 'y ~ x'
```




```
#Holiday Flag and Weekly Sales Bivariate Scatterplot
ggplot(dfSales, aes(x=IsHoliday.x, y=Weekly_Sales)) + geom_point()+ geom_smoo
th(method="lm") + stat_cor(method = "pearson")

## `geom_smooth()` using formula 'y ~ x'
```




```
#Date and Weekly Sales Bivariate Scatterplot
ggplot(dfSales, aes(x=Date, y=Weekly_Sales)) + geom_point()+ geom_smooth(meth
od="lm") + stat_cor(method = "pearson")

## `geom_smooth()` using formula 'y ~ x'
```

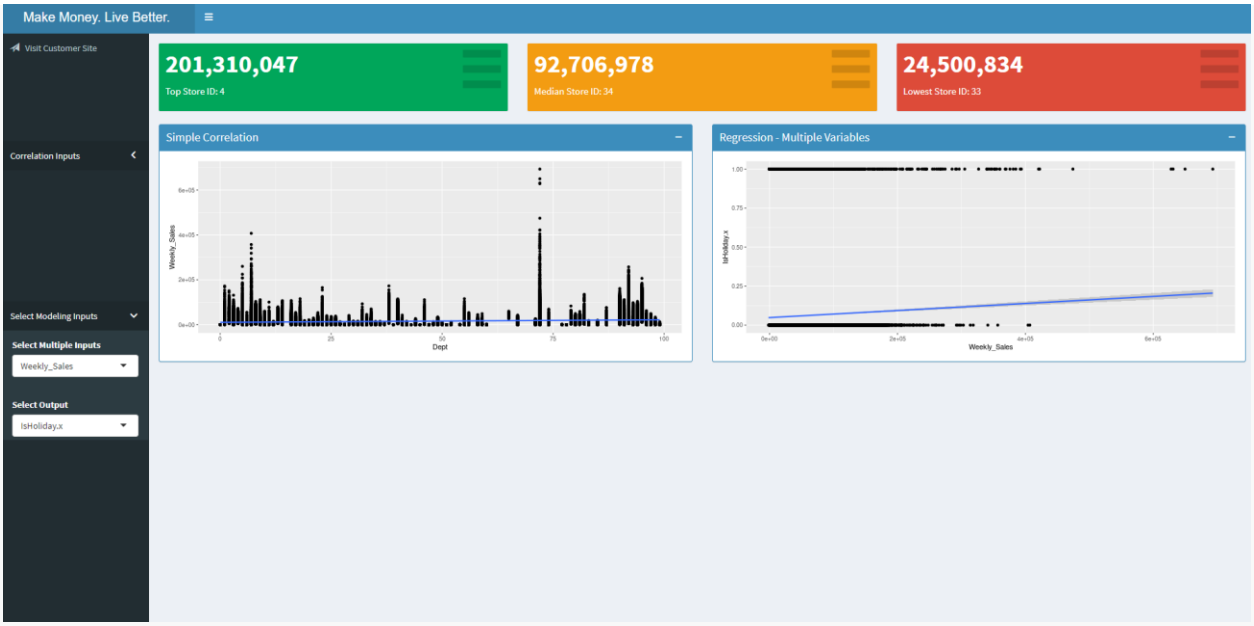



```
#Multi-variable regression - includes CPI, unemployment, Fuel_price, Temp, an
d IsHoliday boolean
#ml1 <- lm(formula=Weekly_Sales~Temperature+Store, data=dfSales)
#ml2 <- lm(formula=Weekly_Sales~CPI+Unemployment+Fuel_Price+IsHoliday.x+Tempe
rature+Store+Date, data=dfSales)
#ml3 <- lm(formula=Weekly_Sales~CPI+Unemployment+Fuel_Price+IsHoliday.x+Store
+Date, data=dfSales)
#ml4 <- lm(formula=Weekly_Sales~CPI+Unemployment+Fuel_Price+Store+Date, data=
dfSales)
#ml5 <- lm(formula=Weekly_Sales~IsHoliday.x+Store+Date, data=dfSales)
ml6 <- lm(formula=Weekly_Sales~CPI+Unemployment+IsHoliday.x+Store+Date, data=
dfSales)

#Output the multivariate regression findings to console
#Temp affects Sales
#summary(ml1)
#CPI, Unempl, Fuel, Holiday & Temp affects Sales
#summary(ml2)
#CPI, Unempl, Fuel, Holiday affects Sales
#summary(ml3)
#CPI+Unemployment+Fuel_Price affects Sales
#summary(ml4)
#Holiday_Flag, Date, Store affects Sales
#summary(ml5)
#CPI+Unemployment+Holiday+Store+Date affects Sales
summary(ml6)
```

```
##
## Call:
## lm(formula = Weekly_Sales ~ CPI + Unemployment + IsHoliday.x +
##     Store + Date, data = dfSales)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -23130 -13068  -8214   4286 672393
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35604.1993  2363.5949  15.064  < 2e-16 ***
## CPI            -20.1407     1.1029 -18.262  < 2e-16 ***
## Unemployment  -163.9431    24.0751  -6.810 9.80e-12 ***
## IsHoliday.x   2373.2212   189.7894  12.504  < 2e-16 ***
## Store         -157.5762     3.4118 -46.186  < 2e-16 ***
## Date            -0.7811     0.1516  -5.152 2.57e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22530 on 286848 degrees of freedom
## Multiple R-squared:  0.008932,   Adjusted R-squared:  0.008915
## F-statistic: 517.1 on 5 and 286848 DF,  p-value: < 2.2e-16

#Conclusion: The models were all determined to be valid based on the p value
or the F Stat; however, the correlation and Adjusted R Squares were all very
low. The Fuel Price and Temperature variables were not significant, so we wou
ld move forward with the 6th model as the best option with all variables sign
ificant and 2nd highest Adjusted R Squared. Due to this initial finding that
we need to find additional impacts on Walmart Sales, a suggestion is to pull
in additional data, perhaps geographic coordinates, more specifics on discoun
ts and online sales, and to create a dynamic dashboard that enables the clien
t to model using all the variables and with expanded data set over time. Sugg
est reading in URL with live as to limited and static data points in excel to
monitor analysis over time.
```

##########SHINY APP DASHBOARD###############

#A Shiny App dashboard is proposed for Phase 2 to allow dynamic selection of inputs, as they are added to the model and to enable the client to monitor trends over time using real time data. The sample dashboard and code and data sources align with Section 3. The dashboard is partially functional; however, it is for illustrative purposes only and should not be used for business decisions until fully developed and tested.




```
#Citations: https://shiny.rstudio.com/articles/layout-guide.html,

#https://datascienceplus.com/building-a-simple-sales-revenue-dashboard-with-r-shiny-shinydashboard/


#Load the required packages

library(shiny)

require(shinydashboard)

library(ggplot2)

library(dplyr)

library(tidyverse)

library(tidyr)

library(lubridate)

#library(flexdashboard)


#Read in the CSV file

walmartsales <- read_csv("salesdataset.csv")
```

```
features <- read_csv("features.csv")

#Fix the Date columns to consistent format
fix_Dates <- function(x, format = c("ymd", "dmy", "mdy")){
  fmt <- lubridate::guess_formats(x, format)
  fmt <- unique(fmt)
  y <- as.Date(x, format = fmt[1])
  for(i in seq_along(fmt)[-1]){
    na <- is.na(y)
    if(!any(na)) break
    y[na] <- as.Date(x[na], format = fmt[i])
  }
  y
}

walmartsales$Date <- fix_Dates(walmartsales$Date)
features$Date <- fix_Dates(features$Date)

#Combine the salesdataset and features dataframes
walmartsales <- data.frame(walmartsales)
features <- data.frame(features)

#Check out the data and replace the NAs
combinedSalesdf <- right_join(walmartsales, features, by=c('Store','Date'))
combinedSalesdf <- replace(combinedSalesdf,is.na(combinedSalesdf), 0)
#glimpse(combinedSalesdf)

#Total Weekly_Sales is low; I suspect rows fell out in the join because of
```

```
#imperfect formatting of the dates in the date fix function

sum(!is.na(combinedSalesdf$Weekly_Sales))


#Save DF for Section 3

dfSales <- combinedSalesdf


################ Code for Phase 2 modeling ######################

#Multi-variable regression - in Phase 2, create string from input drop down

#ml <- lm(formula=Weekly_Sales~CPI+Unemployment+Fuel_Price+IsHoliday.x+Temper
ature+Store+Date, data=dfSales)

#summary(ml)


################ Code to generate the Shiny App ######################

#Linebreak function to help space things in the Shiny app

linebreaks <- function(n){HTML(strrep(br(), n))}


ui <- dashboardPage(

  dashboardHeader(),

  dashboardSidebar(),

  dashboardBody()

)


server <- function(input, output) { }

shinyApp(ui, server)


#Dashboard header carrying the title of the dashboard

header <- dashboardHeader(titleWidth = 300, title= "Make Money. Live Better."
)

#header <- dashboardHeader(titleWidth = 300, logo_path="walmartlogo.jpg")
```

```
#Sidebar content of the dashboard
sidebar <- dashboardSidebar(
  sidebarMenu(
    #menuItem("Dashboard", tabName = "dashboard", icon = icon("dashboard")),
    menuItem("Visit Customer Site", icon = icon("send",lib='glyphicon'),
             href = "https://www.walmart.com"),
    #menuItem("           "),
    #menuItem("           "),
    #menuItem("           "),
    #menuItem("           "),
    linebreaks(6),
    menuItem("Correlation Inputs",
             selectInput("attribute1","Select Input",
                         colnames(dfSales[,2:5])),
             selectInput("attribute2","Select Output",
                         colnames(dfSales[,1:5]))),
    #    menuItem("           "),
    #    menuItem("           "),
    #    menuItem("           "),
    #    menuItem("           "),
    #    menuItem("           "),
    #    menuItem("           "),
    #    menuItem("           "),
    #    menuItem("           "),
    #    menuItem("           "),
    #    menuItem("          "),
    linebreaks(10),
    menuItem("Select Modeling Inputs",
```

```
             selectInput("attribute3","Select Multiple Inputs",
                         #                         colnames(walmartsales[,-3]
), multiple=TRUE),
                         colnames(dfSales[,-3])),
             selectInput("attribute4","Select Output",
                         colnames(dfSales[,c(3,5)]))
    )


  )
)



frow1 <- fluidRow(
  valueBoxOutput("value1")
  ,valueBoxOutput("value2")
  ,valueBoxOutput("value3")
)
frow2 <- fluidRow(
  box(
    title = "Simple Correlation"
    ,status = "primary"
    ,solidHeader = TRUE
    ,collapsible = TRUE
    ,plotOutput("sales.model", height = "300px")
  )
  ,box(
    title = "Regression - Multiple Variables"
    ,status = "primary"
    ,solidHeader = TRUE
```

```
    ,collapsible = TRUE
    ,plotOutput("sales.model2", height = "300px")
  )
)
###Comment this out since value box outputs aren't working with three fluid r
ows
frow3 <- fluidRow(
  box(
    title = "Store Profitability"
    ,status = "primary"
    ,solidHeader = TRUE
    ,collapsible = TRUE
    ,plotOutput("sales.model", height = "300px")
  )
  ,box(
    title = "Store Level Regression - Multiple Variables"
    ,status = "primary"
    ,solidHeader = TRUE
    ,collapsible = TRUE
    ,plotOutput("sales.model2", height = "300px")
  ),
  #  )
  linebreaks(1)
  ,box(
    title = "Store Profitability"
    ,status = "primary"
    ,solidHeader = TRUE
    ,collapsible = TRUE
    ,plotOutput("sales.model", height = "300px")
```

```
  )
  ,box(
    title = "Store Level Regression - Multiple Variables"
    ,status = "primary"
    ,solidHeader = TRUE
    ,collapsible = TRUE
    ,plotOutput("sales.model2", height = "300px")
  )
)



# combine the two fluid rows to make the body
body <- dashboardBody(frow1, frow2)
#Three fluid rows placeholder for phase 2 < had issue creating 3 rows
#body <- dashboardBody(frow1, frow2, frow3)


#completing the ui part with dashboardPage
ui <- dashboardPage(title = 'Walmart Sales Analysis', header, sidebar, body,
skin='blue')


# create the server functions for the dashboard
server <- function(input, output) {
  ####Data manipulation for the three Revenue boxes at the top
  top.store <-  dfSales %>% group_by(Store) %>% summarise(value = sum(Weekly_
Sales)) %>% filter(value==max(value))
  median.store <- dfSales %>% group_by(Store) %>% summarise(value = sum(Weekl
y_Sales)) %>% filter(value==median(value))
  low.store <- dfSales %>% group_by(Store) %>% summarise(value = sum(Weekly_S
ales)) %>% filter(value==min(value))
  show.model <- dfSales %>% summary(ml)
```

```
  ####Data manipulation for the two graphs in the middle

  #revenue.by.week <- walmartsales %>% group_by(Date) %>% summarise(value = s
um(Weekly_Sales))

  # sales.model <- walmartsales %>% group_by(Date) %>%


  #Create dynamic model function based on inputs

  #Placeholder for phase 2


  #Other functions I didn't end up using

  #total.revenue <- sum(recommendation$Revenue)

  #total.revenue <- sum(walmartsales$Weekly_Sales)

  #sales.account <- walmartsales %>% group_by(Store) %>% summarise(value = su
m(Weekly_Sales)) %>% filter(value==max(value))

  #sale.Temperature <- walmartsales %>% group_by(Temperature) %>% summarise(v
alue = sum(Weekly_Sales))


  ####Data manipulation for the customer sentiment word cloud for top store a
t the bottom

  #Placeholder for phase 2


  ####Creating the three Revenue boxes at the top

  output$value1 <- renderValueBox({

    valueBox(

      formatC(top.store$value, format="d", big.mark=',')

      ,paste('Top Store ID:',top.store$Store)

      ,icon = icon("menu-hamburger",lib='glyphicon')

      ,color = "green")

  })
```

```
  output$value2 <- renderValueBox({
    valueBox(
      formatC(median.store$value, format="d", big.mark=',')
      ,paste('Median Store ID:',median.store$Store)
      ,icon = icon("menu-hamburger",lib='glyphicon')
      ,color = "yellow")
  })
  output$value3 <- renderValueBox({
    valueBox(
      formatC(low.store$value, format="d", big.mark=',')
      ,paste('Lowest Store ID:',low.store$Store)
      ,icon = icon("menu-hamburger",lib='glyphicon')
      ,color = "red")
  })



  ####Create the simple correlation model
  output$sales.model <- renderPlot({
    colIndex <- which(input$attribute1==colnames(dfSales))
    x <- dfSales[, colIndex]
    colIndex2 <- which(input$attribute2==colnames(dfSales))
    y <- dfSales[, colIndex2]


    #Draw the scatterplot with the input
    ggplot(data=dfSales,aes(x,y))+geom_point()+labs(x=input$attribute1, y=inp
ut$attribute2)+ geom_smooth(method="lm")
  })
  ####Create the multiple regression model
  output$sales.model2 <- renderPlot({
```

```
    #ggplot(data=walmartsales, aes(x=Temperature, y=Weekly_Sales)) + geom_poi
nt()+ geom_smooth(method="lm")

    # generate x and y coordinates for the dataset from ui.R

    colIndex <- which(input$attribute3==colnames(dfSales))

    x2 <- dfSales[, colIndex]

    colIndex2 <- which(input$attribute4==colnames(dfSales))

    y2 <- dfSales[, colIndex2]



    #Draw the scatterplot with the input

    ggplot(data=dfSales,aes(x2,y2))+geom_point()+labs(x=input$attribute3, y=i
nput$attribute4)+ geom_smooth(method="lm")

  })

}



#run/call the shiny app

shinyApp(ui, server)
```