Final Project Report

Delaney Norman

2023-12-06

Importing the Dataset

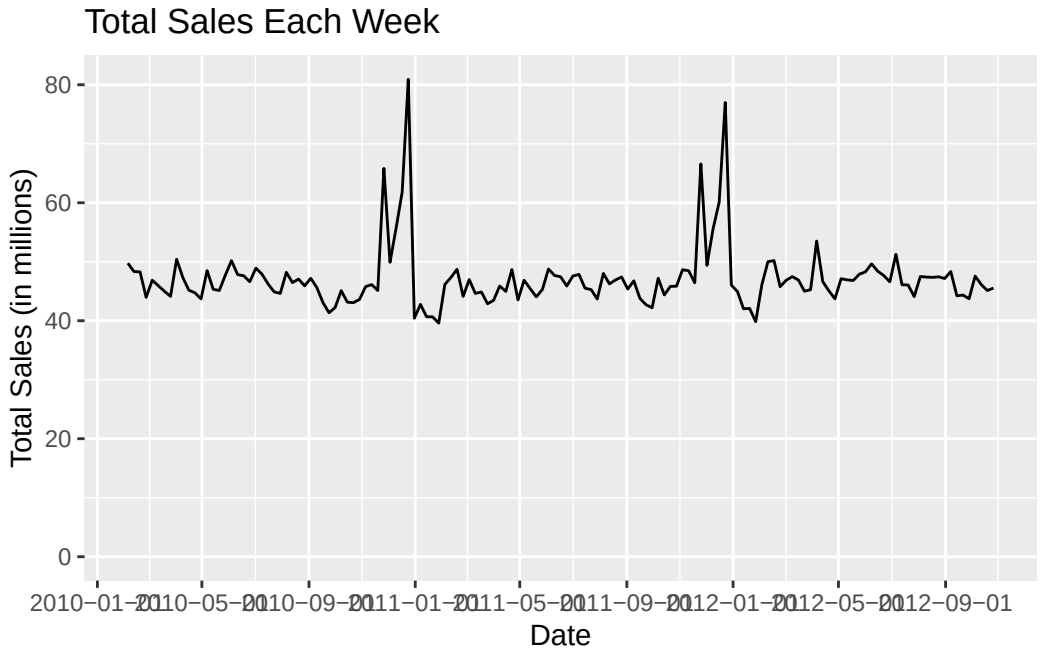
```
# Original URL: https://www.kaggle.com/datasets/yasserh/walmart-dataset/data

store_data = read.csv(
  "/Users/delracman/Documents/STAT 7500/Project/Walmart.csv"
) %>%
  mutate(
    Date=as.Date(Date,format="%d-%m-%Y"),
    Year=year(Date),
    Month=month(Date),
    Day=day(Date),
    Week=week(Date),
    Plot_Date=as.Date(paste(2013,Month,Day,sep="-"),format="%Y-%m-%d")
  ) %>%
  arrange(Date)
head(store_data)
```

```
  Store       Date Weekly_Sales Holiday_Flag Temperature Fuel_Price      CPI
1     1 2010-02-05    1643690.9            0       42.31      2.572 211.0964
2     2 2010-02-05    2136989.5            0       40.19      2.572 210.7526
3     3 2010-02-05     461622.2            0       45.71      2.572 214.4249
4     4 2010-02-05    2135143.9            0       43.76      2.598 126.4421
5     5 2010-02-05     317173.1            0       39.70      2.572 211.6540
6     6 2010-02-05    1652635.1            0       40.43      2.572 212.6224
  Unemployment Year Month Day Week  Plot_Date
1        8.106 2010     2   5    6 2013-02-05
```

```
2         8.324 2010     2   5    6 2013-02-05
3         7.368 2010     2   5    6 2013-02-05
4         8.623 2010     2   5    6 2013-02-05
5         6.566 2010     2   5    6 2013-02-05
6         7.259 2010     2   5    6 2013-02-05
```

Total Sales Data

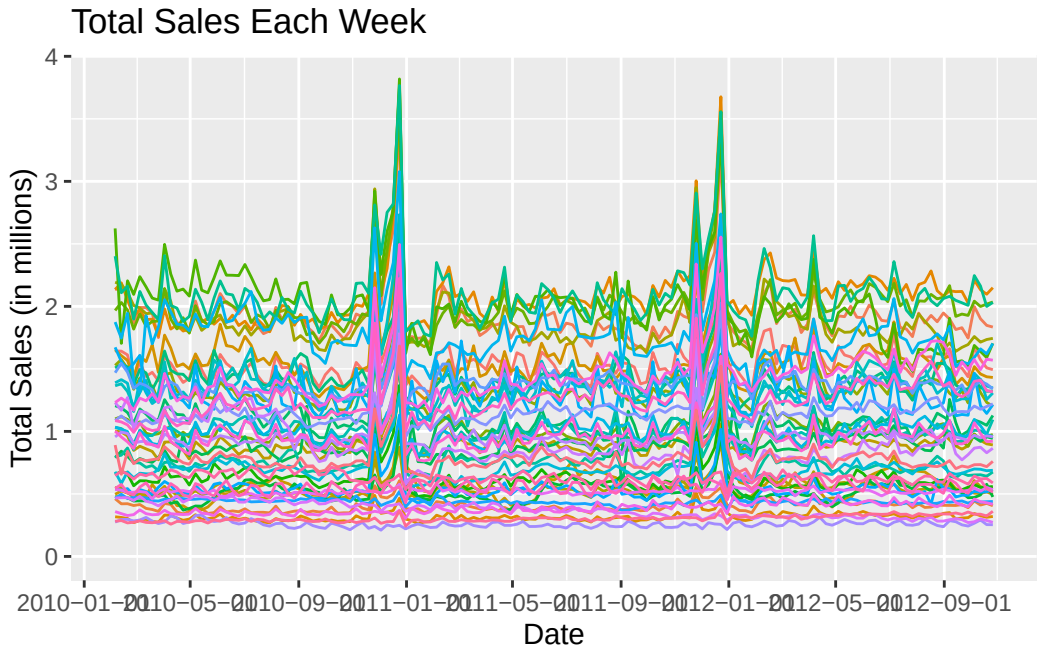
```
store_data_sales = store_data %>%
  group_by(Date) %>%
  summarize(Total_Sales=sum(Weekly_Sales))

store_data_sales %>%
  ggplot() +
    geom_line(aes(x=Date,y=Total_Sales)) +
    labs(
      title="Total Sales Each Week",
      y="Total Sales (in millions)"
    ) +
    scale_x_date(breaks = function(x) seq.Date(from = as.Date("2010-01-01"),
                                               to = as.Date("2012-12-31"),
                                               by = "4 months")
                                             ) +
    scale_y_continuous(
      labels = scales::number_format(scale = 1/1000000),
      limits=c(0,NA)
    )
```




```
store_data %>%
  group_by(Store) %>%
  summarize(Total_Store_Sales=sum(Weekly_Sales)) %>%
  arrange(desc(Total_Store_Sales))
```

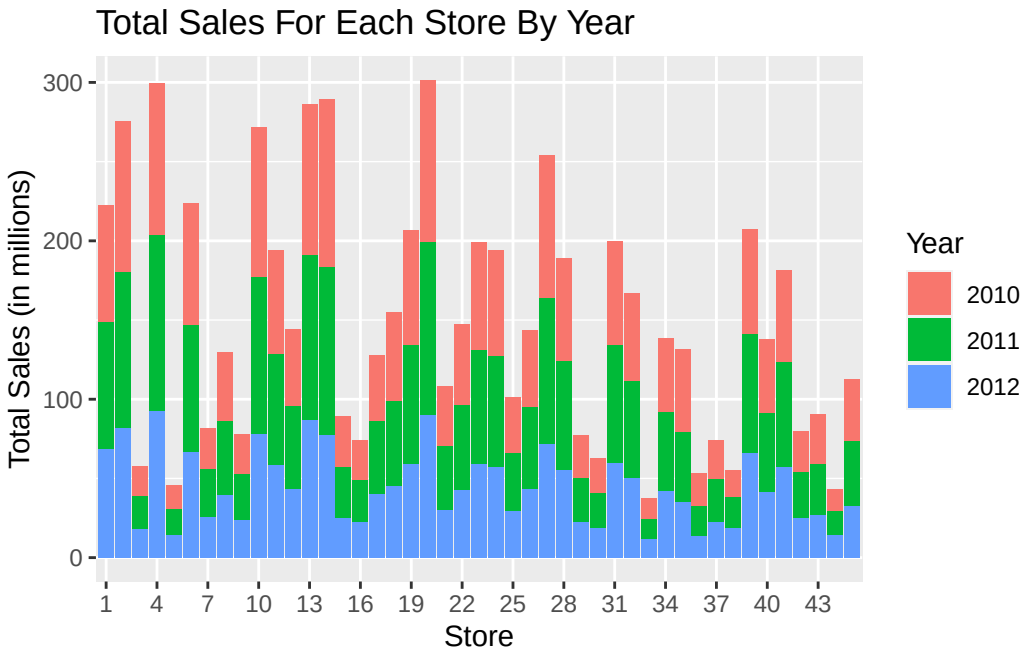
```
# A tibble: 45 x 2
   Store Total_Store_Sales
   <int>             <dbl>
 1    20        301397792.
 2     4        299543953.
 3    14        288999911.
 4    13        286517704.
 5     2        275382441.
 6    10        271617714.
 7    27        253855917.
 8     6        223756131.
 9     1        222402809.
10    39        207445542.
# i 35 more rows
```

```r
store_data %>%
  ggplot() +
    geom_line(aes(x=Date,y=Weekly_Sales, color=factor(Store))) +
    labs(
      title="Total Sales Each Week",
      y="Total Sales (in millions)"
    ) +
    scale_x_date(breaks = function(x) seq.Date(from = as.Date("2010-01-01"),
                                               to = as.Date("2012-12-31"),
                                               by = "4 months")
                                             ) +
    scale_y_continuous(
      labels = scales::number_format(scale = 1/1000000),
      limits=c(0,NA)
    ) +
    theme(legend.position = "none")
```



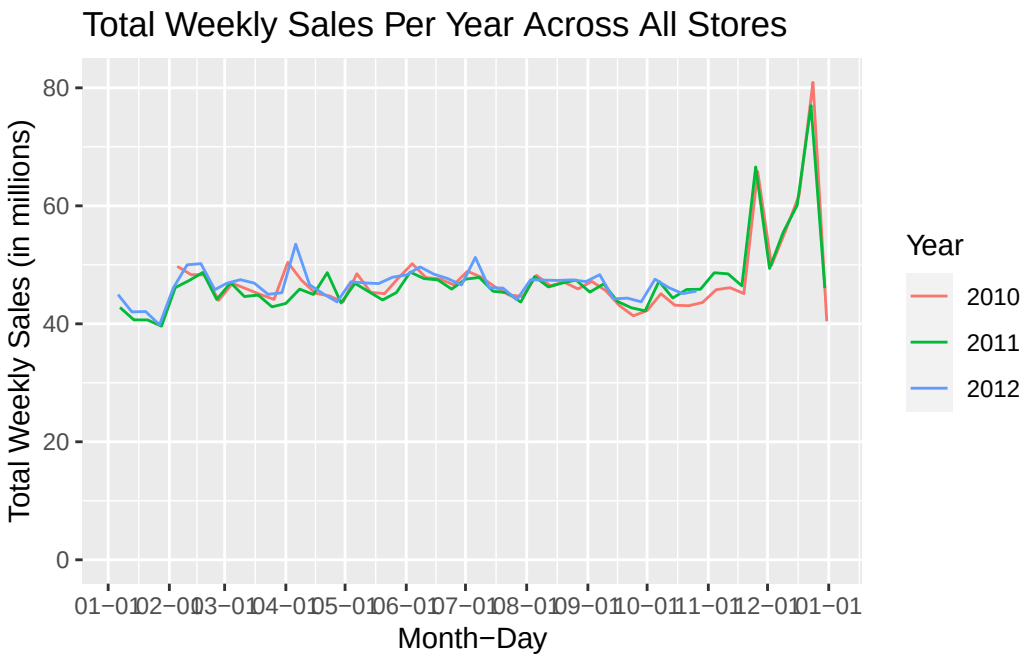

```r
store_data %>%
  group_by(Store, Year) %>%
  summarize(Total_Store_Sales=sum(Weekly_Sales)) %>%
  arrange(desc(Total_Store_Sales)) %>%
```

```
  ggplot() +
    geom_bar(
      aes(x=factor(Store), y=Total_Store_Sales, fill=factor(Year)),
      stat = "identity"
    ) +
    scale_x_discrete(breaks=seq(1,45,by=3)) +
    scale_y_continuous(
      labels = scales::number_format(scale = 1/1000000),
      limits=c(0,NA)
    ) +
    labs(
      title="Total Sales For Each Store By Year",
      x="Store",
      y="Total Sales (in millions)",
      fill="Year"
    )
```

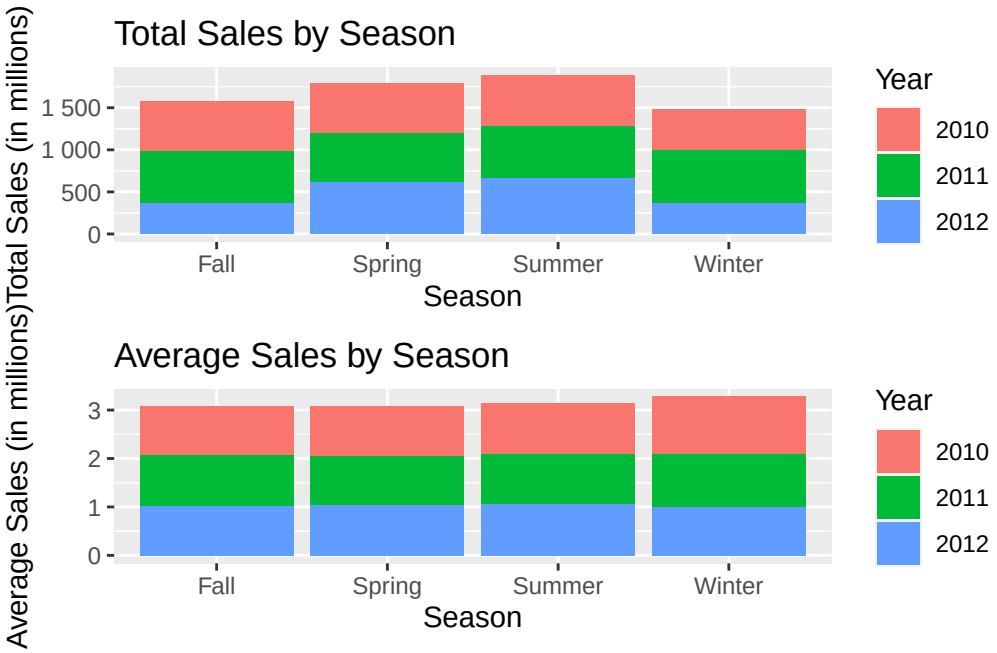



```
sales = store_data %>%
  group_by(Year,Week) %>%
  mutate(
    Average_Sales=mean(Weekly_Sales),
    Total_Sales=sum(Weekly_Sales)
```

```
  )

sales %>%
  ggplot() +
    geom_line(aes(x=Plot_Date,y=Total_Sales,color=factor(Year))) +
    labs(
        x="Month-Day",
        y="Total Weekly Sales (in millions)",
        title="Total Weekly Sales Per Year Across All Stores",
        color="Year"
      ) +
    scale_x_date(
      breaks = function(x) seq.Date(from = as.Date("2013-01-01"),
                                                    to = as.Date("2014-01-01"),
                                                    by = "1 month"),
      date_labels="%m-%d"
    ) +
    scale_y_continuous(
      labels = scales::number_format(scale = 1/1000000),
      limits=c(0,NA)
    )
```



```r
season_data = store_data %>%
  mutate(
    Season=case_when(
      Month %in% c(12,1,2) ~ "Winter",
      Month %in% c(3:5) ~ "Spring",
      Month %in% c(6:8) ~ "Summer",
      .default = "Fall"
    )
  ) %>%
  group_by(Season, Year) %>%
  summarize(Total = sum(Weekly_Sales), Average=mean(Weekly_Sales))

season_plots = list()

season_plots[["total"]] = season_data %>%
  ggplot() +
    geom_bar(aes(x=Season, y=Total, fill=factor(Year)), stat = "identity") +
    labs(title="Total Sales by Season", y="Total Sales (in millions)", fill="Year") +
    scale_y_continuous(
      labels = scales::number_format(scale = 1/1000000),
      limits=c(0,NA)
    )
season_plots[["average"]] = season_data %>%
  ggplot() +
    geom_bar(aes(x=Season, y=Average, fill=factor(Year)), stat = "identity") +
    labs(title="Average Sales by Season", y="Average Sales (in millions)", fill="Year") +
    scale_y_continuous(
      labels = scales::number_format(scale = 1/1000000),
      limits=c(0,NA)
    )

season_plots[["total"]] + season_plots[["average"]] + plot_layout(ncol = 1, nrow = 2)
```



Holidays

Adding additional holidays

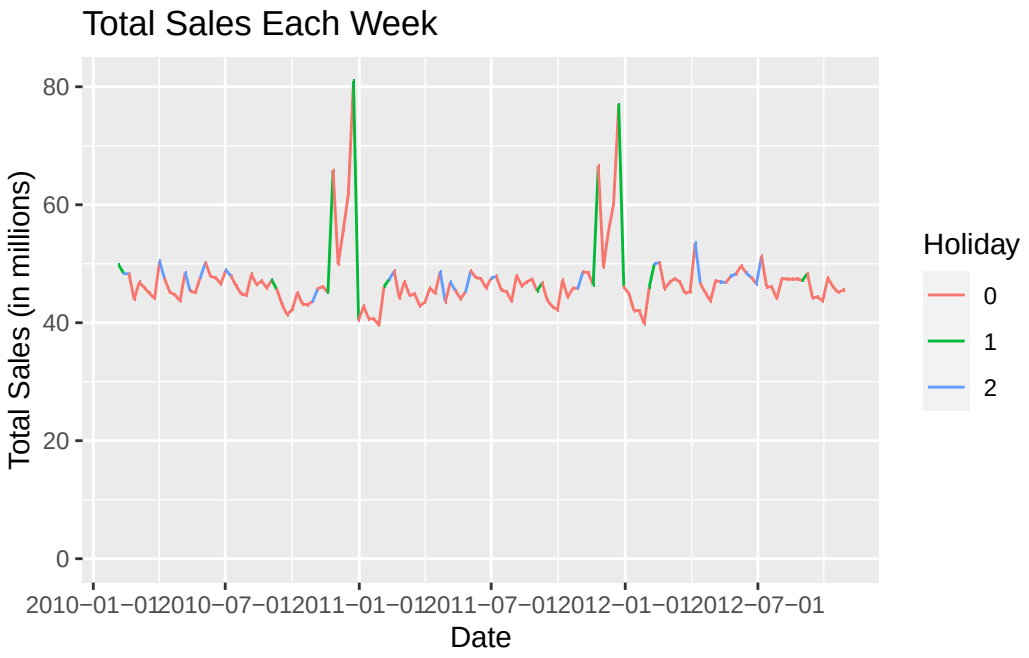
```
# Add other holidays to the store_data.
# Flag with "2" to differentiate from pre-existing holidays in the store_dataset
# Observation: dates are for end of week, not beginning of week
# (so take date and check week previous, not week after)
# Also vectorized to be able to use with ifelse
holidays = c(
  "2010-05-09", "2011-05-08", "2012-05-13", # Mother's Day
  "2012-06-20", "2012-06-19", "2012-06-17", # Father's Day
  "2010-05-31", "2011-05-30", "2012-05-28", # Memorial Day
  "2010-04-04", "2011-04-24", "2012-04-08", # Easter
  "2010-10-31", "2011-10-31", "2012-10-31", # Halloween
  "2010-07-04", "2011-07-04", "2012-07-04", # 4th of July
  "2010-02-14", "2011-02-14", "2012-02-14"  # Valentine's Day
)
determine_holiday = function(day) {
  for (holiday in holidays) {
    if(between(as.numeric(as.Date(holiday,format="%Y-%m-%d")-day), -6, 0)) {
```

```
      return(TRUE)
    }
  }
  return(FALSE)
}

# Allows determine_holiday to take in one date at a time in the ifelse
determine_holiday_vectorized <- Vectorize(determine_holiday)
store_data_with_holidays = store_data %>%
  mutate(
    Holiday_Flag = ifelse(
      Holiday_Flag==0,
      ifelse(
        determine_holiday_vectorized(Date),
        2,
        Holiday_Flag
      ),
      Holiday_Flag
    )
  )
```

Looking at holiday impact

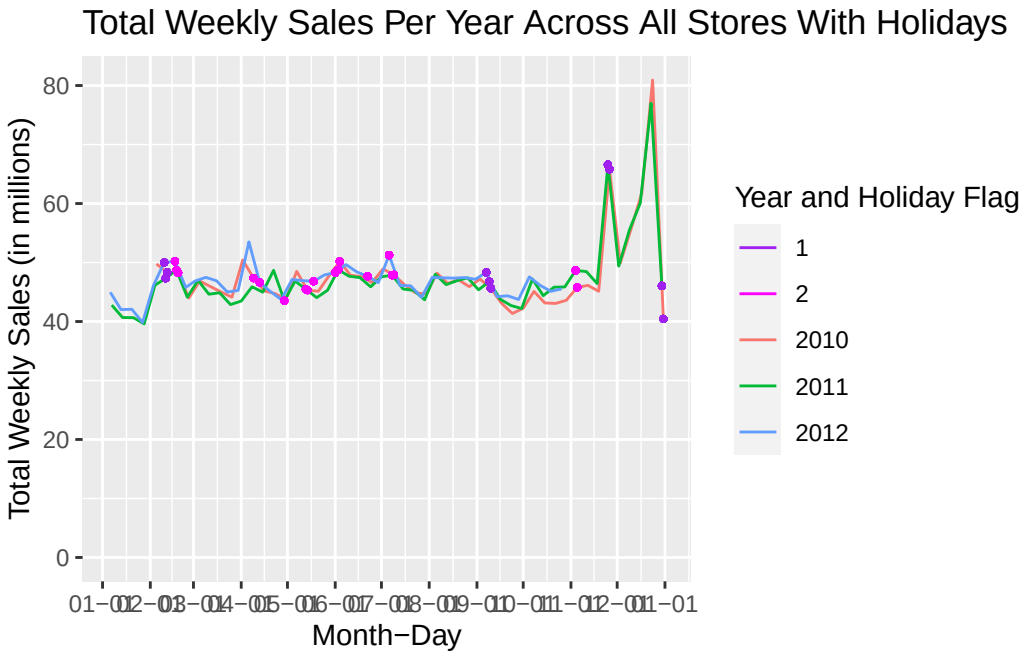
```
holidays = store_data_with_holidays %>%
  select(Date, Holiday_Flag) %>%
  distinct() %>%
  mutate(
    Next_Week_Holiday = lead(
      Holiday_Flag, default = 0
    )
  )

store_data_with_holidays = store_data_with_holidays %>%
  group_by(Date) %>%
  mutate(
    Total_Sales=sum(Weekly_Sales),
  ) %>%
  left_join(
    holidays %>%
      select(-Holiday_Flag),
```

```
    by="Date"
  )

store_data_with_holidays %>%
  ggplot() +
    geom_line(
      aes(
        x=Date,
        y=Total_Sales,
        color=factor(Next_Week_Holiday),
        group=1
      )
    ) +
    labs(
      title="Total Sales Each Week",
      color="Holiday",
      y="Total Sales (in millions)"
    ) +
    scale_x_date(breaks = function(x) seq.Date(from = as.Date("2010-01-01"),
                                               to = as.Date("2012-12-31"),
                                               by = "6 months")) +
    scale_y_continuous(
      labels = scales::number_format(scale = 1/1000000),
      limits=c(0,NA)
    )
```




```
store_data_with_holidays %>%
  ggplot() +
    geom_line(aes(x=Plot_Date,y=Total_Sales,color=factor(Year))) +
    geom_point(aes(
      x=Plot_Date,
      y=Total_Sales,
      alpha=factor(Holiday_Flag),
      color=factor(Holiday_Flag)
    ),size=0.75, show.legend = FALSE) +
    scale_alpha_manual(values = c("0" = 0, "1" = 1, "2" = 1)) +
    scale_color_manual(values = c(
      "2010" = "#F8766D",
      "2011" = "#00BA38",
      "2012" = "#619CFF",
      "1"="purple",
      "2"="magenta"
    )) +
    labs(
        x="Month-Day",
        y="Total Weekly Sales (in millions)",
        title="Total Weekly Sales Per Year Across All Stores With Holidays",
        color="Year and Holiday Flag"
```

```
  ) +
scale_x_date(
  breaks = function(x) seq.Date(from = as.Date("2013-01-01"),
                                to = as.Date("2014-01-01"),
                                by = "1 month"),
  date_labels="%m-%d"
) +
scale_y_continuous(
  labels = scales::number_format(scale = 1/1000000),
  limits=c(0,NA)
)
```



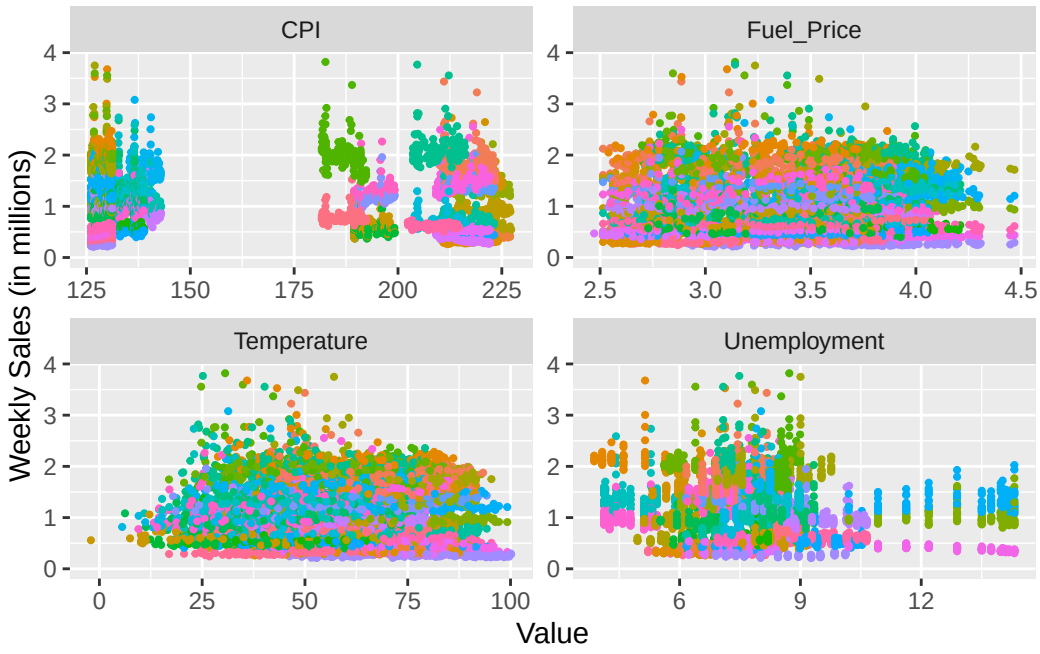
Looking at other factors

```
longer_store_data = store_data %>%
  pivot_longer(
    cols=c("Unemployment", "CPI", "Fuel_Price", "Temperature"),
    names_to="Category",
    values_to="Value"
  )
```

```r
head(longer_store_data)
```

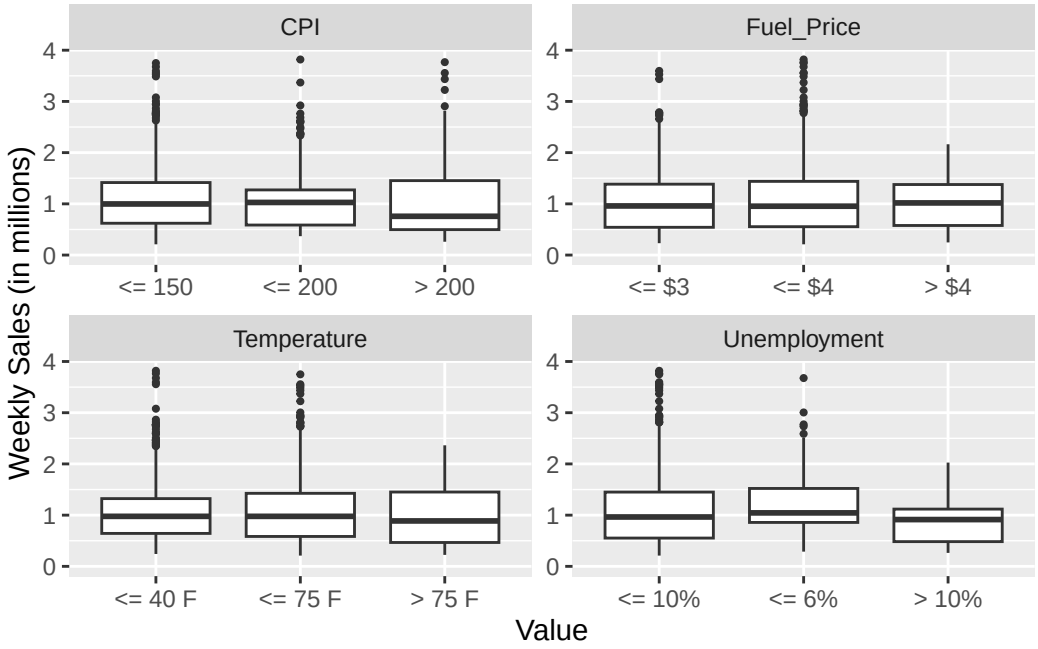
```
# A tibble: 6 x 11
  Store Date       Weekly_Sales Holiday_Flag  Year Month   Day  Week Plot_Date
  <int> <date>            <dbl>        <int> <dbl> <dbl> <int> <dbl> <date>
1     1 2010-02-05     1643691.            0  2010     2     5     6 2013-02-05
2     1 2010-02-05     1643691.            0  2010     2     5     6 2013-02-05
3     1 2010-02-05     1643691.            0  2010     2     5     6 2013-02-05
4     1 2010-02-05     1643691.            0  2010     2     5     6 2013-02-05
5     2 2010-02-05     2136989.            0  2010     2     5     6 2013-02-05
6     2 2010-02-05     2136989.            0  2010     2     5     6 2013-02-05
# i 2 more variables: Category <chr>, Value <dbl>
```

```r
longer_store_data %>%
  ggplot() +
    geom_point(aes(x=Value,y=Weekly_Sales,color=factor(Store)), size=0.75) +
    facet_wrap(~Category, scales="free") +
    theme(legend.position = "none") +
    scale_y_continuous(
      labels = scales::number_format(scale = 1/1000000),
      limits=c(0,NA)
    ) +
    labs(y = "Weekly Sales (in millions)")
```




```
store_data %>%
  mutate(
    Unemployment = case_when(
      Unemployment <= 6~"<= 6%",
      Unemployment <= 10~"<= 10%",
      .default="> 10%"
    ),
    CPI = case_when(
      CPI <= 150 ~"<= 150",
      CPI <= 200 ~"<= 200",
      .default="> 200"
    ),
    Fuel_Price = case_when(
      Fuel_Price <= 3 ~"<= $3",
      Fuel_Price <= 4 ~"<= $4",
      .default="> $4"
    ),
    Temperature = case_when(
      Temperature <= 40 ~"<= 40 F",
      Temperature <= 75 ~"<= 75 F",
      .default="> 75 F"
    )
```

```
  ) %>%
  select(Date, Weekly_Sales, Unemployment, CPI, Fuel_Price, Temperature) %>%
  pivot_longer(
    cols = c("Unemployment", "CPI", "Fuel_Price", "Temperature"),
    names_to = "Category",
    values_to = "Value"
  ) %>%
  ggplot() +
    geom_boxplot(aes(x=Value,y=Weekly_Sales),outlier.size=0.75) +
    facet_wrap(~Category, scales = "free", nrow = 2, ncol = 2) +
    scale_y_continuous(
      labels = scales::number_format(scale = 1/1000000),
      limits=c(0,NA)
    ) +
    labs(y = "Weekly Sales (in millions)")
```

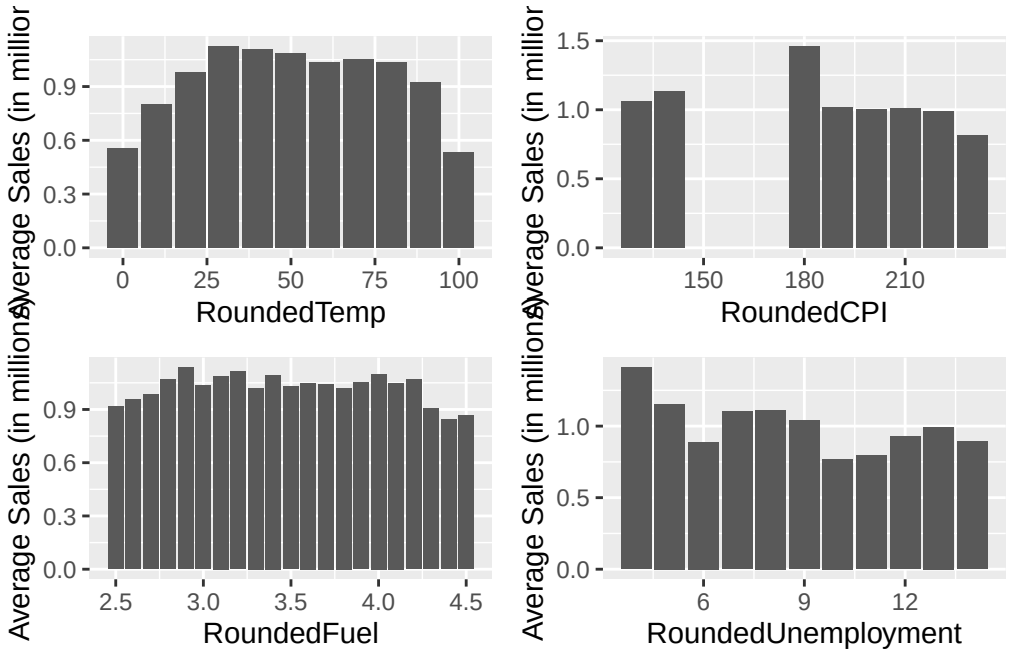



```
rounded_data = store_data %>%
  mutate(
    RoundedTemp=round(Temperature,digits=-1),
    RoundedCPI=round(CPI,digits=-1),
    RoundedFuel=round(Fuel_Price,digits=1),
```

```
    RoundedUnemployment=round(Unemployment)
  )

plots = list()
for (col in c("RoundedTemp", "RoundedCPI", "RoundedFuel", "RoundedUnemployment")) {
  plot = rounded_data %>%
    group_by(!!sym(col)) %>%
    summarize(AvgSales=mean(Weekly_Sales)) %>%
    ggplot(aes_string(x=col,y="AvgSales")) +
      geom_bar(stat = "identity") +
      scale_y_continuous(
        labels = scales::number_format(scale = 1/1000000),
        limits=c(0,NA)
      ) +
      labs(y="Average Sales (in millions)")
  plots[[col]] <- plot
}
```

```
Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
i Please use tidy evaluation idioms with `aes()`.
i See also `vignette("ggplot2-in-packages")` for more information.
```

```
plots[["RoundedTemp"]] +
  plots[["RoundedCPI"]] +
  plots[["RoundedFuel"]] +
  plots[["RoundedUnemployment"]] +
  plot_layout(ncol = 2, nrow = 2)
```


Average Sales (in millior
0.0
0.3
0.6
0.9
0
25
50
75
100
RoundedTemp
Average Sales (in millior
0.0
0.5
1.0
1.5
150
180
210
RoundedCPI
Average Sales (in millions)
0.0
0.3
0.6
0.9
2.5
3.0
3.5
4.0
4.5
RoundedFuel
Average Sales (in millions)
0.0
0.5
1.0
6
9
12
RoundedUnemployment

Conclusion