Final Project Report

Delaney Norman

2023-12-05

Importing the Dataset

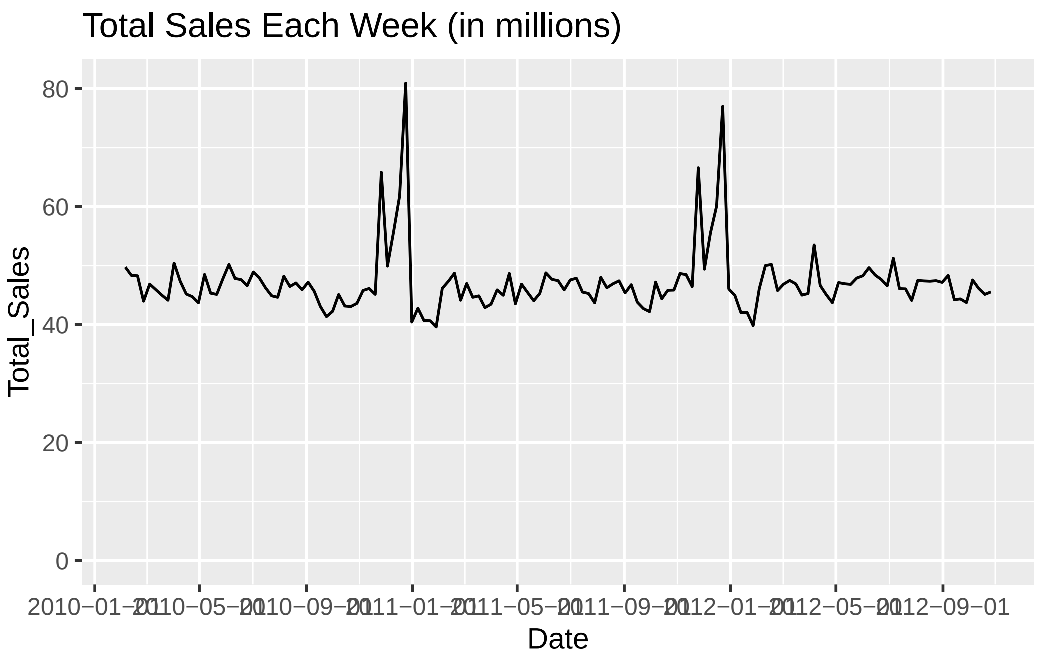
```r
# Original URL: https://www.kaggle.com/datasets/yasserh/walmart-dataset/data

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)
  ) %>%
  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
1        8.106 2010     2   5    6
2        8.324 2010     2   5    6
3        7.368 2010     2   5    6
4        8.623 2010     2   5    6
```

```
5        6.566 2010     2   5    6
6        7.259 2010     2   5    6
```

Total Sales Data

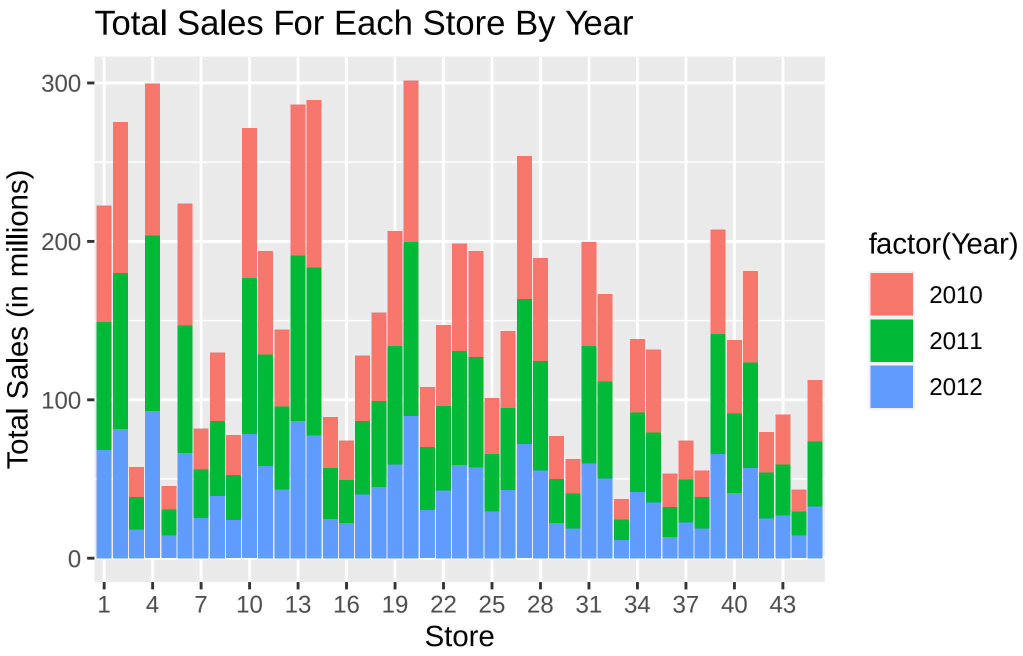
```
store_data_sales = store_data %>%
  group_by(Date) %>%
  summarize(Total_Sales=sum(Weekly_Sales))

store_data_sales %>%
  ggplot() +
    geom_line(aes(x=Date,y=Total_Sales)) +
    labs(title="Total Sales Each Week (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)
    )
```



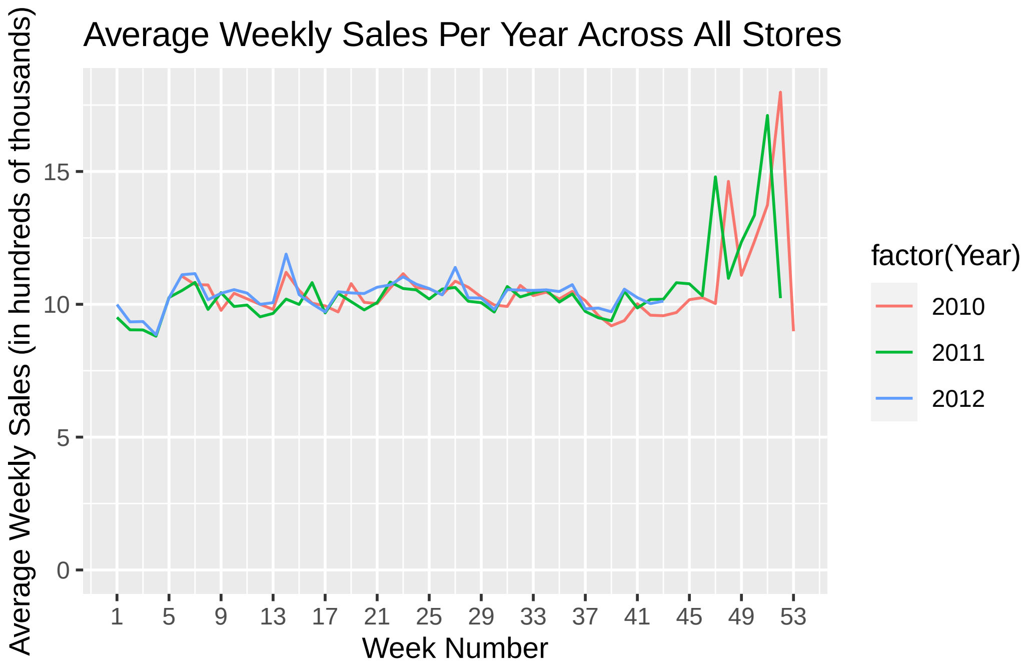
```
total_sales_by_store = store_data %>%
  group_by(Store, Year) %>%
  summarize(Total_Store_Sales=sum(Weekly_Sales)) %>%
  arrange(desc(Total_Store_Sales))

total_sales_by_store
```

```
# A tibble: 135 x 3
# Groups:   Store [45]
   Store  Year Total_Store_Sales
   <int> <dbl>             <dbl>
 1     4  2011        111092293.
 2    20  2011        109837002.
 3    14  2011        106096271.
 4    14  2010        105462242.
 5    13  2011        104537513.
 6    20  2010        101733081.
 7    10  2011         98916895.
 8     2  2011         98607881.
 9     4  2010         95680471.
10     2  2010         95277864.
# i 125 more rows
```

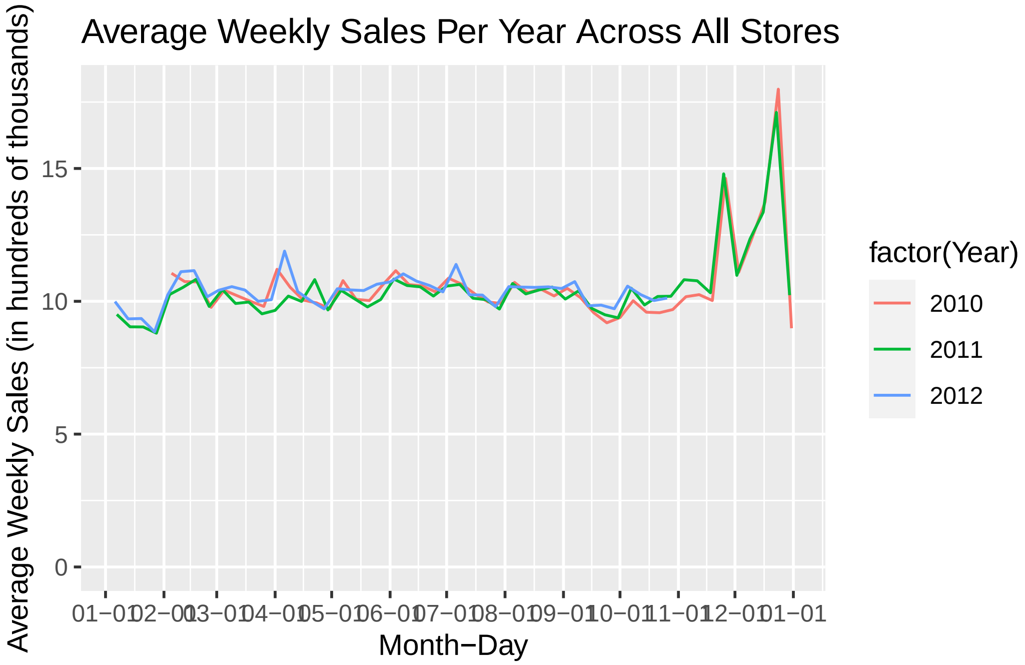
```
total_sales_by_store %>%
  ggplot() +
    geom_bar(aes(x=factor(Store), y=Total_Store_Sales, fill=factor(Year)), stat = "identit
    scale_x_discrete(breaks=seq(1,45,by=3)) +
    scale_y_continuous(
      labels = scales::number_format(scale = 1/1000000),
      limits=c(0,NA)
    ) +
    labs(title="Total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



```
avgsales = store_data %>%
  group_by(Year,Week) %>%
  mutate(
    Average_Sales=mean(Weekly_Sales),
    Plot_Date=as.Date(paste(2013,Month,Day,sep="-"),format="%Y-%m-%d")
  )

avgsales %>%
  ggplot() +
    geom_line(aes(x=Week,y=Average_Sales,color=factor(Year))) +
    labs(
      x="Week Number",
      y="Average Weekly Sales (in hundreds of thousands)",
      title="Average Weekly Sales Per Year Across All Stores"
    ) +
    scale_x_continuous(breaks = seq(1,53,by=4)) +
    scale_y_continuous(
      labels = scales::number_format(scale = 1/100000),
      limits=c(0,NA)
    )
```



```
avgsales %>%
  ggplot() +
    geom_line(aes(x=Plot_Date,y=Average_Sales,color=factor(Year))) +
    labs(
        x="Month-Day",
        y="Average Weekly Sales (in hundreds of thousands)",
        title="Average Weekly Sales Per Year Across All Stores"
      ) +
    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 = 1/100000),
      limits=c(0,NA)
    )
```



Holidays

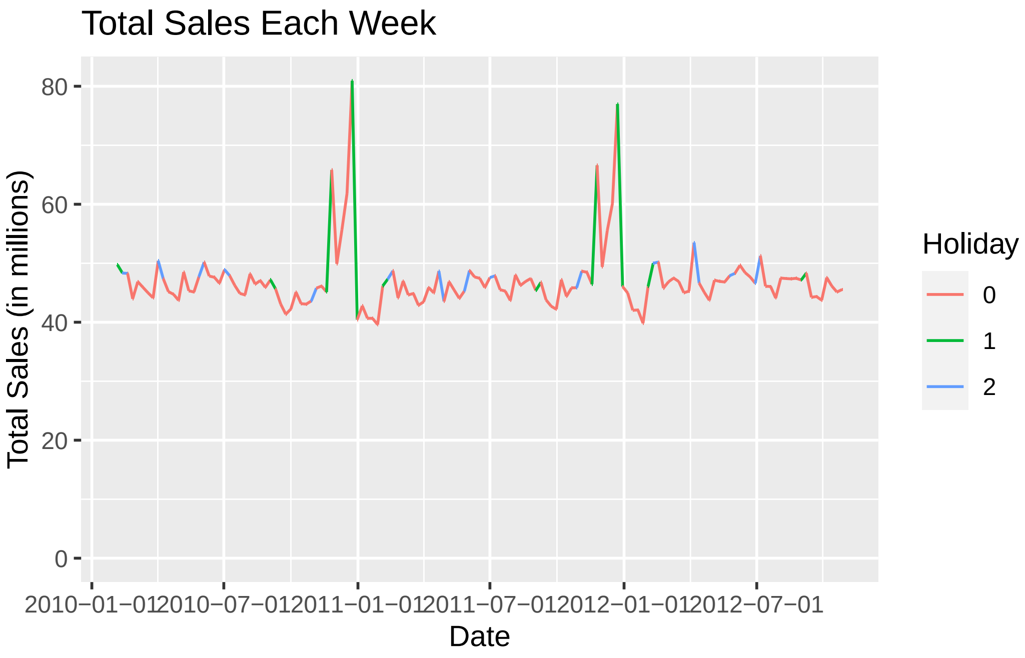
Adding additional holidays

```
# Add other holidays to the store_data. Flag with "2" to differentiate from pre-existing h
# Observation: dates are for end of week, not beginning of week (so take date and check we
# Also vectorized to be able to use with ifelse
holidays = c(
  "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)
```

```r
}

# 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,Hol
  )
```

Looking at holiday impact

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

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

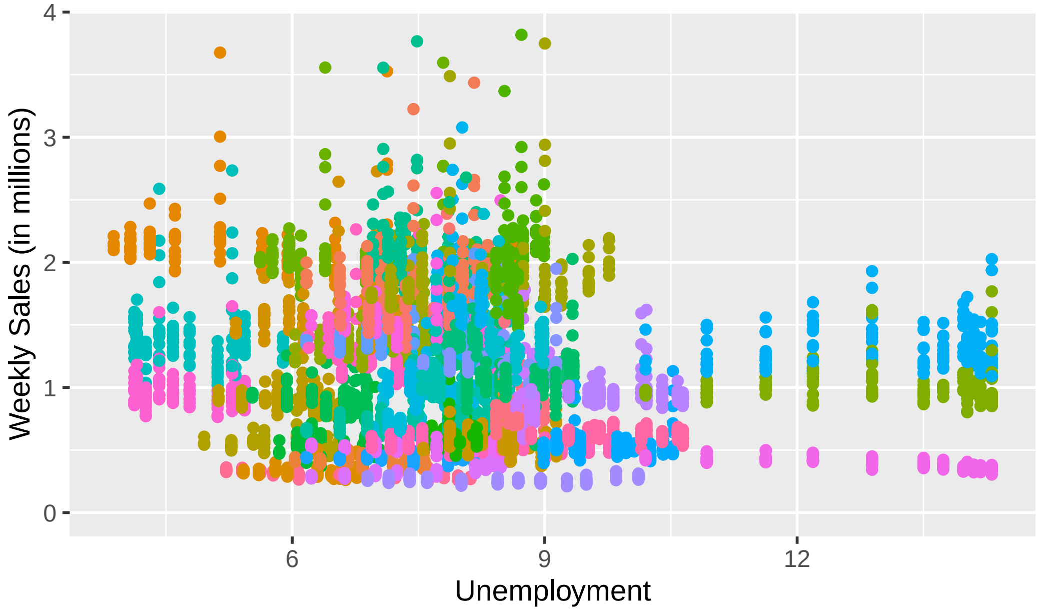


Looking at other factors

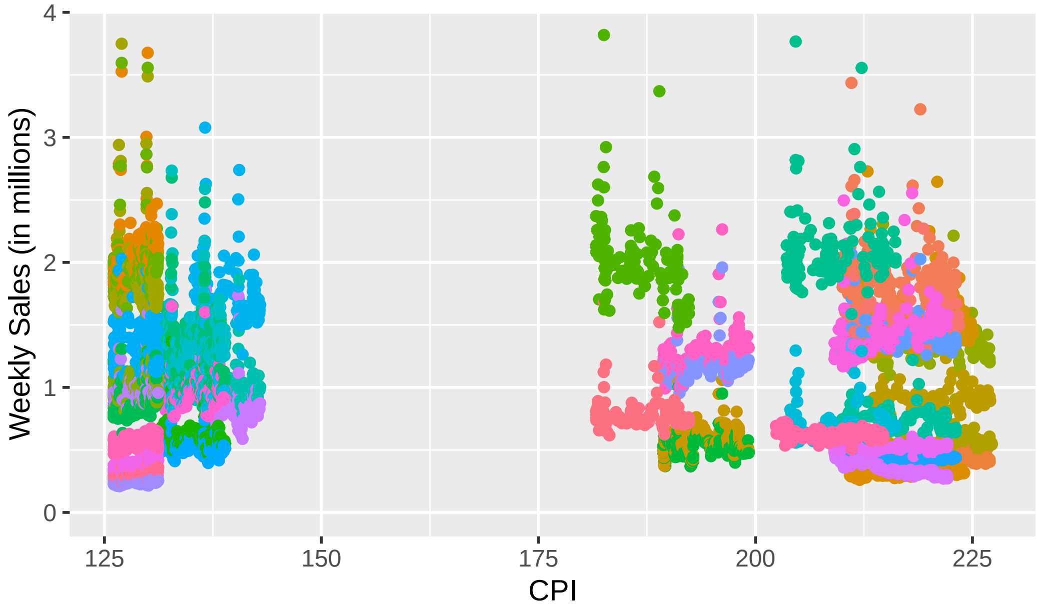
```
for (col in c("Unemployment", "CPI", "Fuel_Price", "Temperature")) {
  label = str_replace_all(col, "_", " ")
  print(
    store_data %>%
      ggplot() +
        geom_point(aes_string(x=col,y="Weekly_Sales",color="factor(Store)")) +
        labs(title=paste0(label," vs Weekly Sales for Each Store"), x=label, y="Weekly Sal
        theme(legend.position = "none") +
        scale_y_continuous(
          labels = scales::number_format(scale = 1/1000000),
          limits=c(0,NA)
      )
  )
}
```

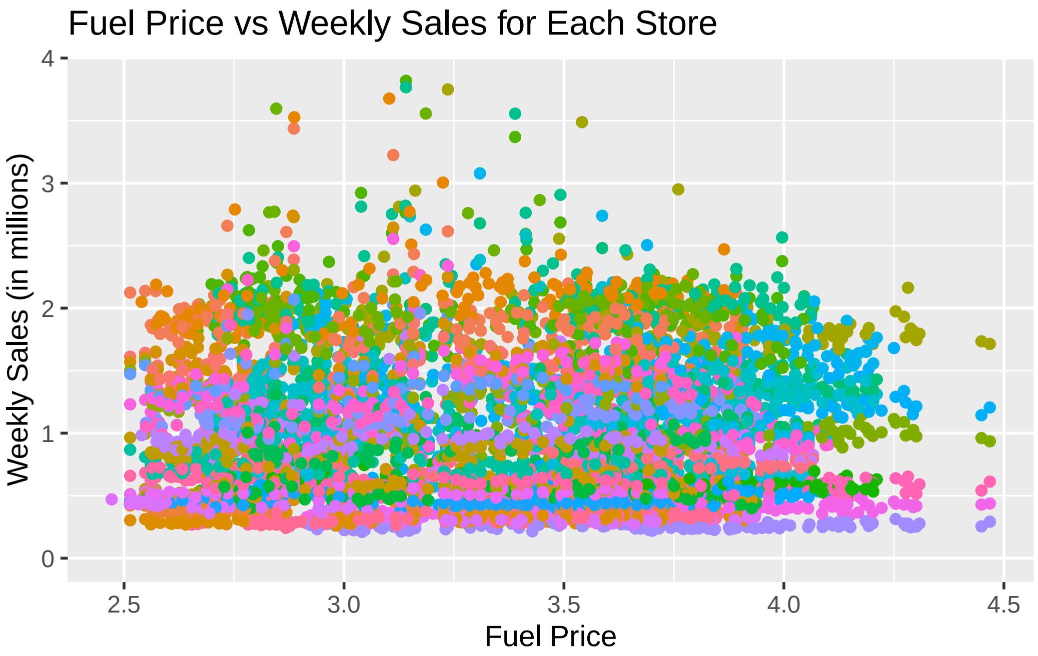
```
Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
i Please use tidy evaluation idioms with `aes()`.
i See also `vignette("ggplot2-in-packages")` for more information.
```

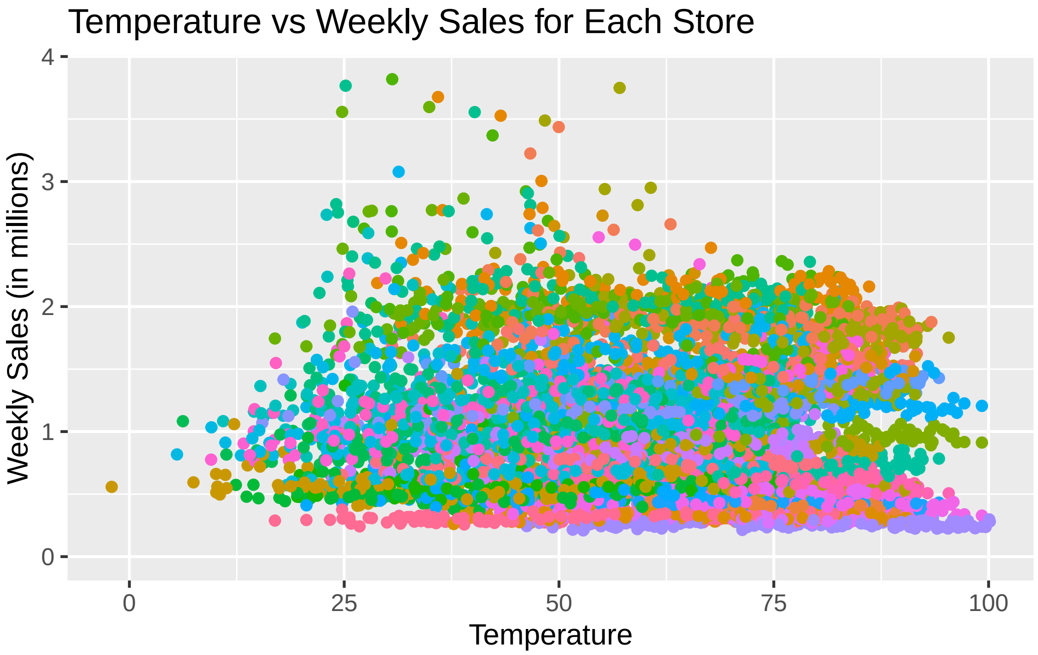
Unemployment vs Weekly Sales for Each Store



CPI vs Weekly Sales for Each Store




Fuel Price vs Weekly Sales for Each Store
Weekly Sales (in millions)
0
1
2
3
4
2.5
3.0
3.5
4.0
4.5
Fuel Price


Temperature vs Weekly Sales for Each Store
Weekly Sales (in millions)
0
1
2
3
4
0
25
50
75
100
Temperature

Conclusion