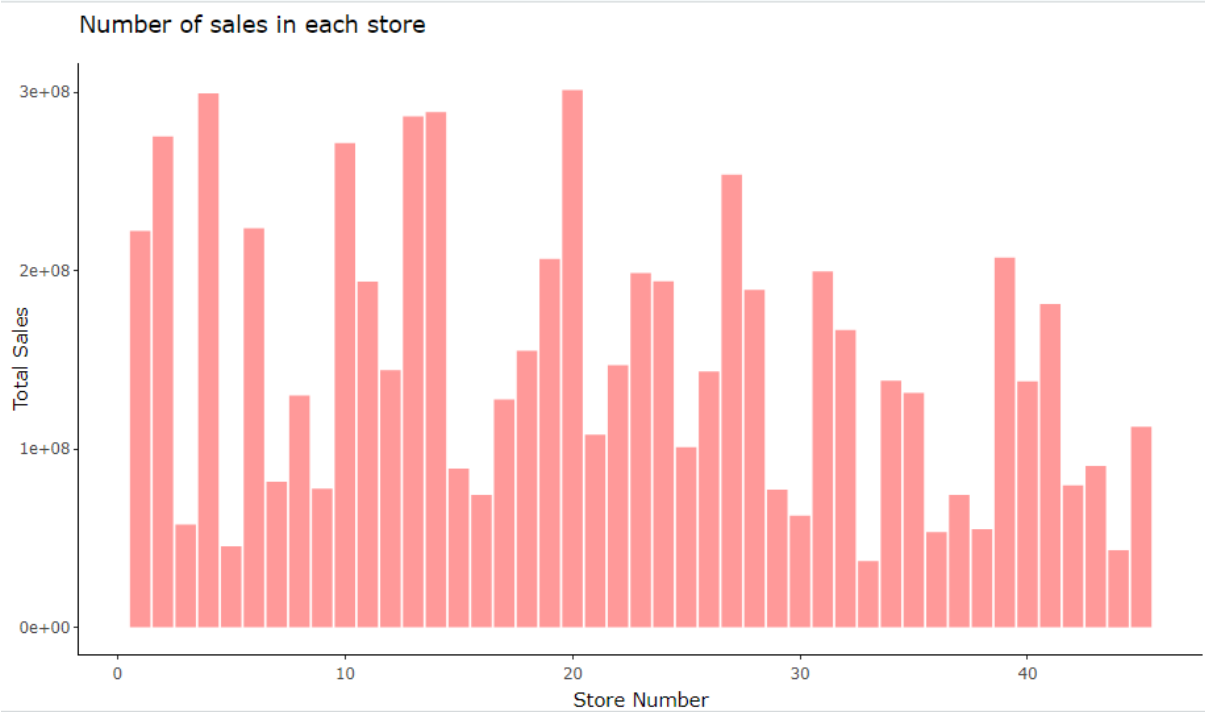
Analysis Tasks

Basic Statistics tasks

- Which store has maximum sales

Solution

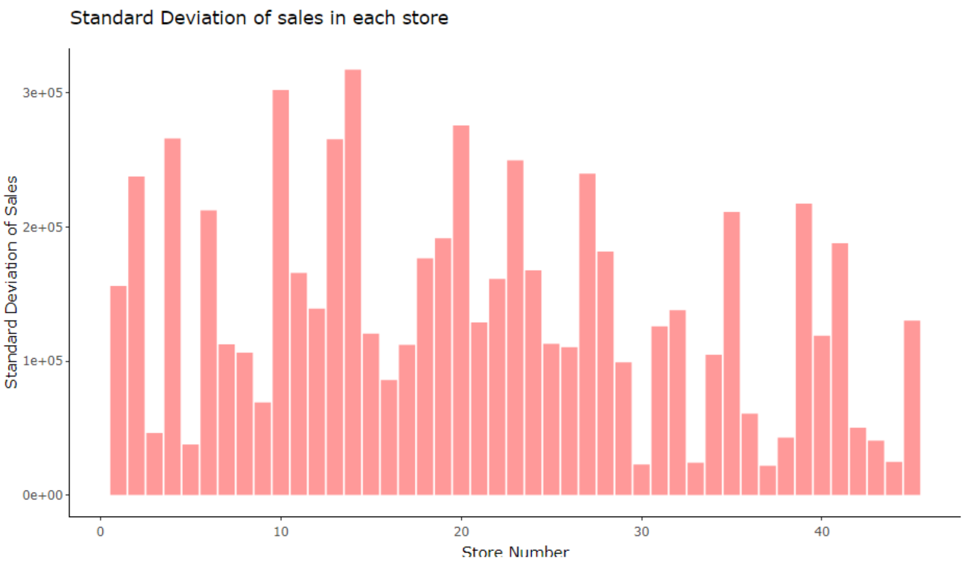
```r
library(ggplot2)
library(plotly)
library(data.table)
library(stats)
library(usdm)
library(caTools)
walmart<-read.csv('D:\\Data_Analysis_Simplilearn_materials\\Data_Science_with_R\\Walmart_Store_sales.csv')
#View(walmart)

# Analysis Task

# Basic Statistics tasks

# 1. Which store has maximum sales

store_sales<-summarise(group_by(walmart,Store),sum(Weekly_Sales),mean(Weekly_Sales),sd(Weekly_Sales))
names(store_sales)<-c('Store','Total_sales','avg_weekly_sales','sales_sd')

ggplot(store_sales,aes(Store,Total_sales))+geom_col(fill='red',alpha=0.4)+
  labs(x='Store Number',y='Total Sales',title='Number of sales in each store')+theme_classic()
ggplotly()
cat("Store number ",head(arrange(store_sales,-Total_sales),1)$Store,"has maximum sales =",
    head(arrange(store_sales,-Total_sales),1)$Total_sales)
```



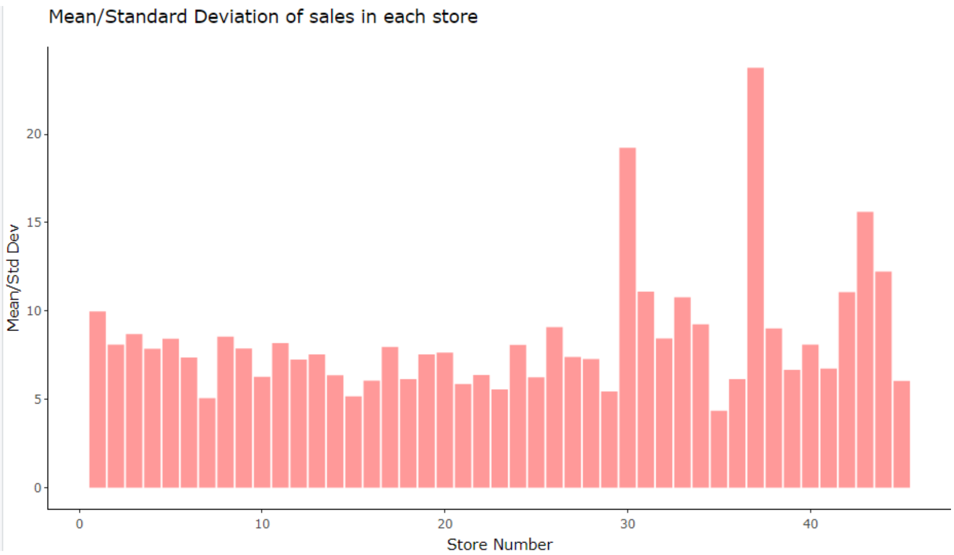

```
> cat("Store number ",head(arrange(store_sales,-Total_sales),1)$Store,"has maximum sales =",
+     head(arrange(store_sales,-Total_sales),1)$Total_sales)
Store number  20 has maximum sales = 301397792
```

- Which store has maximum standard deviation i.e., the sales vary a lot. Also, find out the coefficient of mean to standard deviation

```r
# 2. Which store has maximum standard deviation i.e., the sales vary a lot.
# Also, find out the coefficient of mean to standard deviation

ggplot(store_sales,aes(Store,sales_sd))+geom_col(fill='red',alpha=0.4)+
  labs(x='Store Number',y='Standard Deviation of Sales',title='Standard Deviation of sales in each store')+
  theme_classic()
ggplotly()
cat("Store number ",head(arrange(store_sales,-sales_sd),1)$Store,"has maximum standard deviation of sales =",
    head(arrange(store_sales,-sales_sd),1)$sales_sd)

store_sales<-mutate(store_sales,sales_mean_to_sd=avg_weekly_sales/sales_sd)
ggplot(store_sales,aes(Store,sales_mean_to_sd))+geom_col(fill='red',alpha=0.4)+
  labs(x='Store Number',y='Mean/Std Dev',title='Mean/Standard Deviation of sales in each store')+
  theme_classic()
ggplotly()
cat("Store number ",head(arrange(store_sales,-sales_mean_to_sd),1)$Store,
    "has maximum mean to standard deviation ratio of sales =",
    head(arrange(store_sales,-sales_mean_to_sd),1)$sales_mean_to_sd)
```



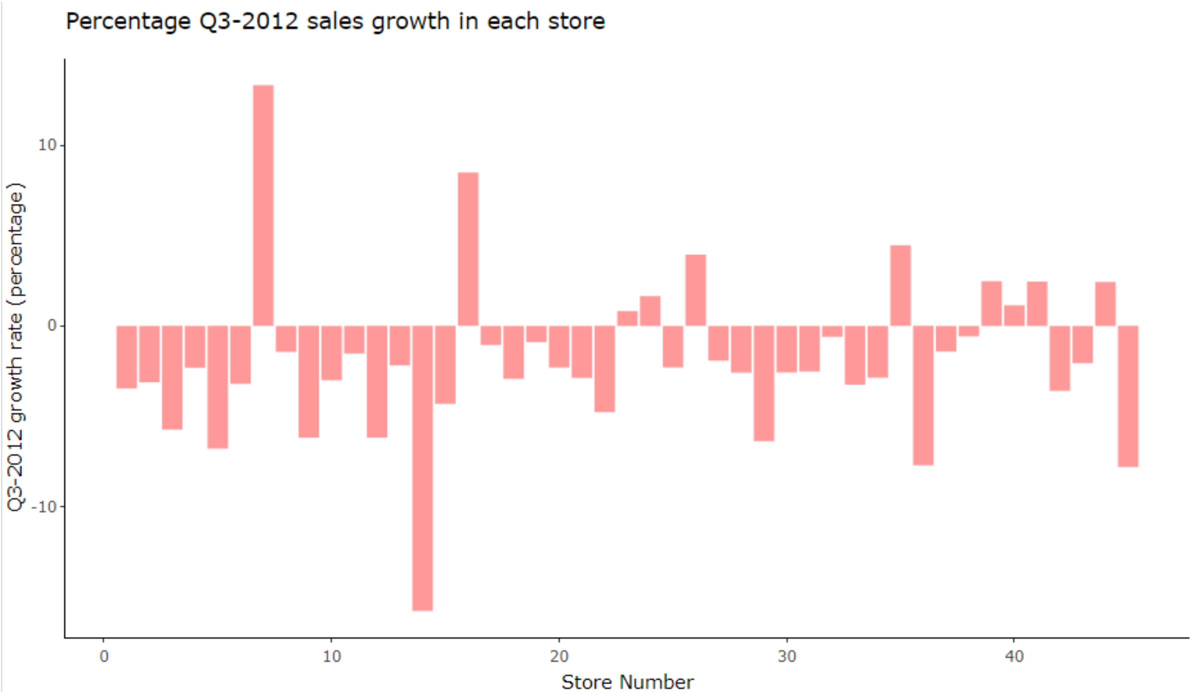

```
> cat("Store number ",head(arrange(store_sales,-sales_sd),1)$Store,"has maximum standard deviation of sales =",
+     head(arrange(store_sales,-sales_sd),1)$sales_sd)
Store number  14 has maximum standard deviation of sales = 317569.9
> 
```




```
> cat("Store number ",head(arrange(store_sales,-sales_mean_to_sd),1)$Store,
+      "has maximum mean to standard deviation ratio of sales =",
+      head(arrange(store_sales,-sales_mean_to_sd),1)$sales_mean_to_sd)
Store number  37 has maximum mean to standard deviation ratio of sales = 23.76193
```

- Which store/s has good quarterly growth rate in Q3'2012

```r

# 3. which store/s has good quarterly growth rate in Q3'2012

walmart$Date<-as.Date(walmart$Date,format='%d-%m-%Y')
storewise_sales_Q3_2012<-walmart%>%filter(Date>='2012-07-01',Date<='2012-09-30')%>%
  group_by(Store)%>%summarise(sum(Weekly_Sales))
names(storewise_sales_Q3_2012)<-c('Store','sales_q3_2012')
storewise_sales_Q2_2012<-walmart%>%filter(Date>='2012-04-01',Date<='2012-06-30')%>%
  group_by(Store)%>%summarise(sum(Weekly_Sales))
names(storewise_sales_Q2_2012)<-c('Store','sales_q2_2012')
storewise_sales_Q2_Q3_2012<-merge(storewise_sales_Q2_2012,storewise_sales_Q3_2012)
storewise_sales_Q2_Q3_2012$q3_growth_rate<-
  ((storewise_sales_Q2_Q3_2012$sales_q3_2012/storewise_sales_Q2_Q3_2012$sales_q2_2012)-1)*100;

ggplot(storewise_sales_Q2_Q3_2012,aes(Store,q3_growth_rate))+
  geom_col(fill='red',alpha=0.4)+labs(x='Store Number',y='Q3-2012 growth rate (percentage)',
                                       title='Percentage Q3-2012 sales growth in each store')+theme_classic()
ggplotly()
print('Top 4 stores with good Q3 growth rate')
print(head(arrange(storewise_sales_Q2_Q3_2012,-q3_growth_rate),4))

```




```
> ggplotly()
> print('Top 4 stores with good Q3 growth rate')
[1] "Top 4 stores with good Q3 growth rate"
> print(head(arrange(storewise_sales_Q2_Q3_2012,-q3_growth_rate),4))
  Store sales_q2_2012 sales_q3_2012 q3_growth_rate
1     7       7290859       8262787      13.330776
2    16       6564336       7121542       8.488378
3    35      10838313      11322421       4.466637
4    26      13155336      13675692       3.955478
> 
```

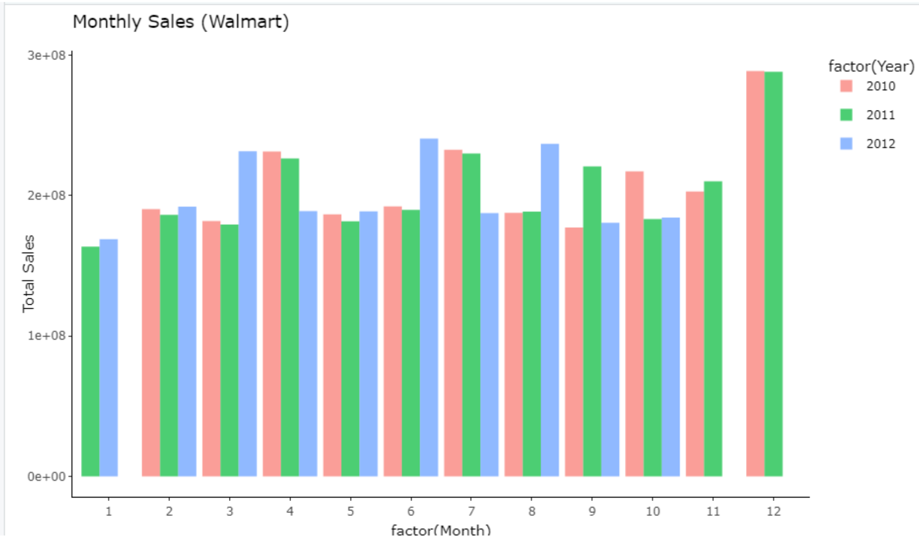
- Some holidays have a negative impact on sales. Find out holidays which have higher sales than the mean sales in non-holiday season for all stores together

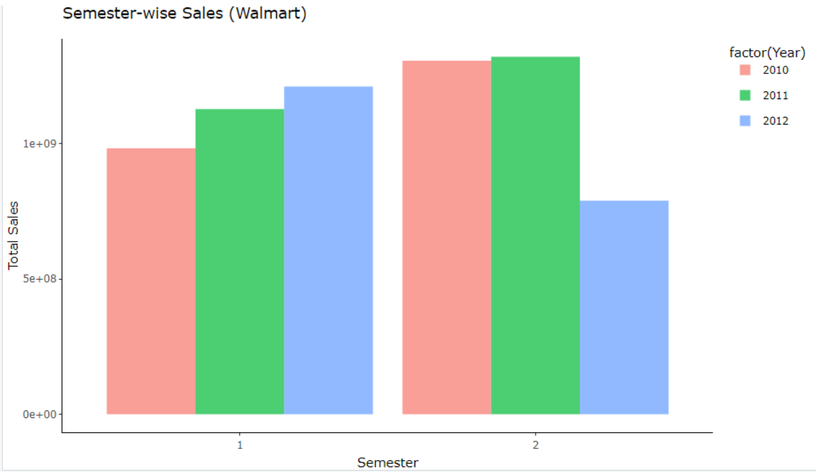
```r

# 4. Some holidays have a negative impact on sales. Find out holidays which have higher
# sales than the mean sales in non-holiday season for all stores together
HolidayType<-function(x){
  if(x=='February'){return('Super Bowl')}
  else if(x=='September'){return('Labour Day')}
  else if(x=='November'){return('Thanksgiving')}
  else if(x=='December'){return('Christmas')}
  else{return('Not Holiday Month')}};


daywise_sales<-summarise(group_by(walmart,Date,Holiday_Flag),sum(Weekly_Sales));
names(daywise_sales)<-c("Date","Holiday_Flag",'Total_Weekly_Sales');
daywise_sales<-as.data.frame(daywise_sales)
working_week_sales<-summarise(filter(daywise_sales,Holiday_Flag==0),mean(Total_Weekly_Sales))[[1]]
print("Holiday weeks when sales are more than mean sales")
print(mutate(filter(daywise_sales,Holiday_Flag==1,Total_Weekly_Sales>working_week_sales),
             festival_season=sapply(months(Date),HolidayType)))

holiday_sales<-summarise(group_by(filter(daywise_sales,Holiday_Flag==1),months(Date)),
                         mean(Total_Weekly_Sales))
names(holiday_sales)<-c('HolidayMonth','MeanWeeklySales')
holiday_sales<-as.data.frame(holiday_sales)
holiday_sales<-mutate(holiday_sales,Holiday_Festival=sapply(HolidayMonth,HolidayType))
print("Festive Seasons when mean sales is higher than mean sales in non-holiday season")
print(filter(holiday_sales,MeanWeeklySales>working_week_sales))
```

```
> print("Holiday weeks when sales are more than mean sales")
[1] "Holiday weeks when sales are more than mean sales"
> print(mutate(filter(daywise_sales,Holiday_Flag==1,Total_Weekly_Sales>working_week_sales),
+              festival_season=sapply(months(Date),HolidayType)))
        Date Holiday_Flag Total_Weekly_Sales festival_season
1 2010-02-12            1           48336678      Super Bowl
2 2010-11-26            1           65821003    Thanksgiving
3 2011-02-11            1           47336193      Super Bowl
4 2011-11-25            1           66593605    Thanksgiving
5 2012-02-10            1           50009408      Super Bowl
6 2012-09-07            1           48330059      Labour Day
> holiday_sales<-summarise(group_by(filter(daywise_sales,Holiday_Flag==1),months(Date)),
+                          mean(Total_Weekly_Sales))
> names(holiday_sales)<-c('HolidayMonth','MeanWeeklySales')
> holiday_sales<-as.data.frame(holiday_sales)
> holiday_sales<-mutate(holiday_sales,Holiday_Festival=sapply(HolidayMonth,HolidayType))
> print("Festive Seasons when mean sales is higher than mean sales in non-holiday season")
[1] "Festive Seasons when mean sales is higher than mean sales in non-holiday season"
> print(filter(holiday_sales,MeanWeeklySales>working_week_sales))
  HolidayMonth MeanWeeklySales Holiday_Festival
1     February        48560759       Super Bowl
2     November        66207304     Thanksgiving
3    September        46909228       Labour Day
> 
```

- Provide a monthly and semester view of sales in units and give insights

```r
# 5. Provide a monthly and semester view of sales in units and give insights
monthly_sales<-summarise(group_by(walmart,month(as.IDate(Date)),year(as.IDate(Date))),sum(Weekly_Sales))
names(monthly_sales)<-c("Month",'Year','TotalSales')
arrange(monthly_sales,Month,Year)

View(monthly_sales)

ggplot(monthly_sales,aes(x=factor(Month),y=TotalSales))+
  geom_col(aes(fill=factor(Year)),alpha=0.7,position = position_dodge(preserve = "single"))+
  labs(y='Total Sales',title='Monthly Sales (Walmart)')+theme_classic()+
  scale_x_discrete(name='Month',labels=month_name)
ggplotly()

monthly_sales=as.data.frame(monthly_sales)
monthly_sales$Semester=ifelse(monthly_sales$Month<=6,1,2)
View(summarise(group_by(monthly_sales,Semester,Year),TotalSales=sum(TotalSales)))
ggplot(summarise(group_by(monthly_sales,Semester,Year),TotalSales=sum(TotalSales)),
       aes(x=factor(Semester),y=TotalSales))+
  geom_col(aes(fill=factor(Year)),alpha=0.7,position = position_dodge(preserve = "single"))+
  labs(x='Semester',y='Total Sales',title='Semester-wise Sales (Walmart)')+theme_classic()
ggplotly()
```





Statistical Model

For Store 1 – Build prediction models to forecast demand

- Linear Regression – Utilize variables like date and restructure dates as 1 for 5 Feb 2010 (starting from the earliest date in order). Hypothesize if CPI, unemployment, and fuel price have any impact on sales.
- Change dates into days by creating new variable.

```
# Statistical Model

# For Store 1 - Build  prediction models to forecast demand

# 1. Linear Regression - Utilize variables like date and restructure dates as 1 for 5 Feb 2010
# (starting from the earliest date in order). Hypothesize if CPI, unemployment,
# and fuel price have any impact on sales.

# 2. Change dates into days by creating new variable.

View(walmart)
walmart_store1<-filter(walmart,Store==1);

View(walmart_store1)




walmart_store1$Day_number<-as.numeric(walmart_store1$Date-as.Date('2010-02-04'))
walmart_store1$month=month(walmart_store1$Date);
walmart_store1$year=year(walmart_store1$Date);
walmart_store1$Super_Bowl=as.numeric(walmart_store1$Holiday_Flag & walmart_store1$month==2)
walmart_store1$Labour_Day=as.numeric(walmart_store1$Holiday_Flag & walmart_store1$month==9)
walmart_store1$Thanksgiving=as.numeric(walmart_store1$Holiday_Flag & walmart_store1$month==11)
walmart_store1$Christmas=as.numeric(walmart_store1$Holiday_Flag & walmart_store1$month==12)
View(walmart_store1)
```

For training data

```
set.seed(101);
train_sample<-sample.split(walmart_store1$Weekly_Sales,SplitRatio = 0.8)
train_data<-subset(walmart_store1,train_sample==T);
test_data<-subset(walmart_store1,train_sample==F);

# Impact of CPI on sales
ggplot(train_data,aes(CPI,Weekly_Sales))+geom_point()
cor.test(train_data$CPI,train_data$Weekly_Sales,method='spearman')
```

```
> cor.test(train_data$CPI,train_data$Weekly_Sales,method='spearman')

        Spearman's rank correlation rho

data:  train_data$CPI and train_data$Weekly_Sales
S = 174142, p-value = 0.001521
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.2947004
```

```r
# Impact of Unemployment on sales
ggplot(train_data,aes(Unemployment,Weekly_Sales))+geom_point()
cor.test(train_data$Unemployment,train_data$Weekly_Sales,method='spearman')
```

```
> cor.test(train_data$Unemployment,train_data$Weekly_Sales,method='spearman')

	Spearman's rank correlation rho

data:  train_data$Unemployment and train_data$Weekly_Sales
S = 292195, p-value = 0.05076
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho 
-0.1834317 
```

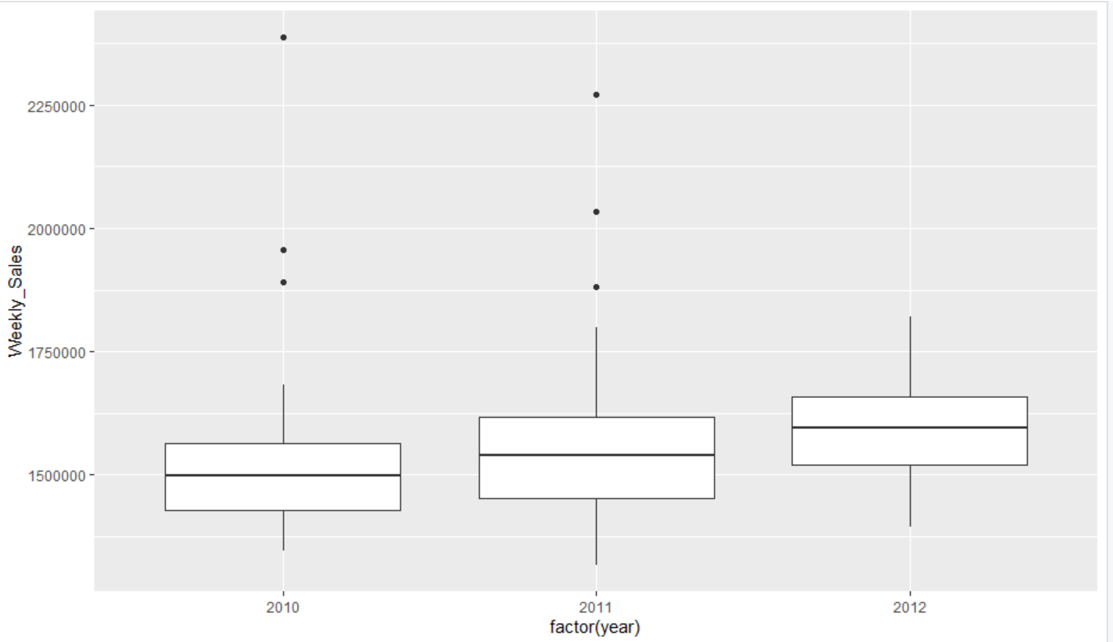
```r
# Impact of Fuel Price on sales
ggplot(train_data,aes(Fuel_Price,Weekly_Sales))+geom_point()
cor.test(train_data$Fuel_Price,train_data$Weekly_Sales,method='spearman')
```

```
> cor.test(train_data$Fuel_Price,train_data$Weekly_Sales,method='spearman')

	Spearman's rank correlation rho

data:  train_data$Fuel_Price and train_data$Weekly_Sales
S = 190905, p-value = 0.01524
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho 
0.2268093 
```

```r
# Impact of Temperature on sales
ggplot(train_data,aes(Temperature,Weekly_Sales))+geom_point()
cor.test(train_data$Temperature,train_data$Weekly_Sales,method='spearman')
```

```
> cor.test(train_data$Temperature,train_data$Weekly_Sales,method='spearman')

	Spearman's rank correlation rho

data:  train_data$Temperature and train_data$Weekly_Sales
S = 293284, p-value = 0.04548
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho 
-0.1878415 
```

```r
# Impact of year on sales
ggplot(train_data,aes(factor(year),Weekly_Sales))+geom_boxplot()
cor.test(train_data$year,train_data$Weekly_Sales,method='spearman')
```

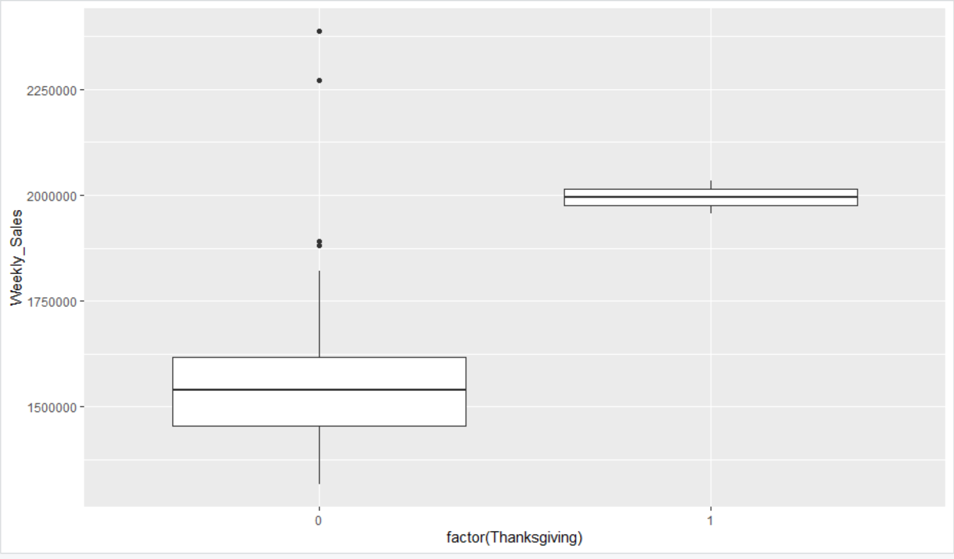



```
> cor.test(train_data$year,train_data$Weekly_Sales,method='spearman')

        Spearman's rank correlation rho

data:  train_data$year and train_data$Weekly_Sales
S = 177268, p-value = 0.002365
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.2820402
```

```r
# Impact of day number on sales
ggplot(train_data,aes(Day_number,Weekly_Sales))+geom_point()
cor.test(train_data$Day_number,train_data$Weekly_Sales,method='spearman')

```

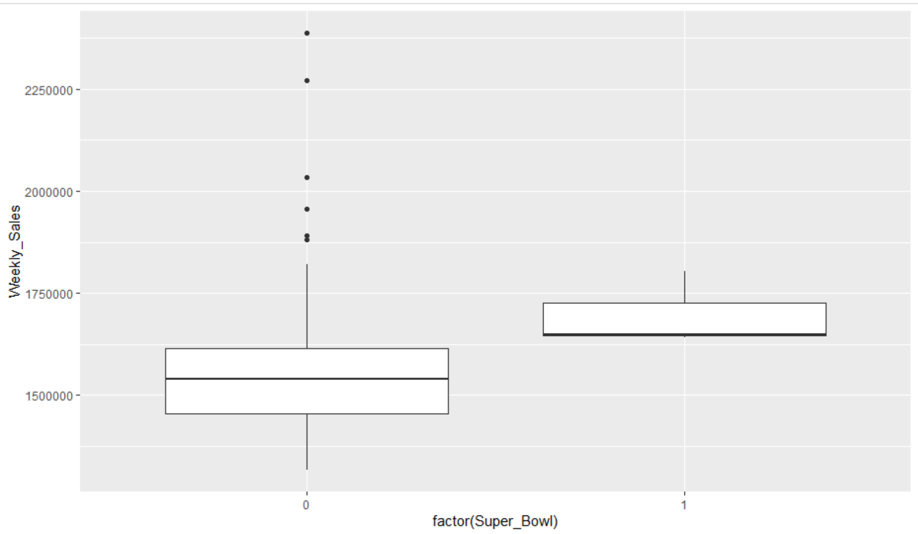
```
> cor.test(train_data$Day_number,train_data$Weekly_Sales,method='spearman')

        Spearman's rank correlation rho

data:  train_data$Day_number and train_data$Weekly_Sales
S = 173870, p-value = 0.001458
alternative hypothesis: true rho is not equal to 0
sample estimates:
     rho
0.295802
```

```r
# Impact of Thanksgiving season on sales
ggplot(train_data,aes(factor(Thanksgiving),Weekly_Sales))+geom_boxplot()
cor.test(train_data$Thanksgiving,train_data$Weekly_Sales,method='spearman')

```




```
> ggplot(train_data,aes(factor(Thanksgiving),Weekly_Sales))+geom_boxplot()
> cor.test(train_data$Thanksgiving,train_data$Weekly_Sales,method='spearman')

        Spearman's rank correlation rho

data:  train_data$Thanksgiving and train_data$Weekly_Sales
S = 192763, p-value = 0.01907
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.2192816
```

```
# Impact of Super Bowl season on sales
ggplot(train_data,aes(factor(Super_Bowl),Weekly_Sales))+geom_boxplot()
cor.test(train_data$Super_Bowl,train_data$Weekly_Sales,method='spearman')
```




```
> cor.test(train_data$Super_Bowl,train_data$Weekly_Sales,method='spearman')

        Spearman's rank correlation rho

data:  train_data$Super_Bowl and train_data$Weekly_Sales
S = 196127, p-value = 0.02815
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.2056588
```

Based on training dataset, Weekly Sales of Store 1 has correlation with factors like CPI, Fuel Price, year, Temperature, Thanksgiving Season, Super Bowl Season and Day Number. Since, probability of these features being uncorrelated with Weekly_Sales<0.05, we can reject the Null Hypothesis which states that they are not correlated with Weekly_Sales.

However, Unemployment column has p-value >0.05 and therefore we cannot reject Null Hypothesis for Unemployment column for Training Dataset and do not include it in Linear Regaression Model

Now, Variance Inflation Factor (VIF) is calculated to check for Multi-colinearity.

Features with greatest VIF are removed one by one and VIF is recomputed till all remaining features have VIF<10

```r
# Sales of Store 1 depend on factors like CPI,Fuel Price, year, Temperature,
#Thanksgiving Season, Super Bowl season,Day Number

vif(train_data[c('CPI','Fuel_Price','year','Temperature','Thanksgiving','Super_Bowl','Day_number')])
vif(train_data[c('CPI','Fuel_Price','year','Temperature','Thanksgiving','Super_Bowl')])
vif(train_data[c('CPI','Fuel_Price','Temperature','Thanksgiving','Super_Bowl')])
```

```
cannot compute exact p-value with ties
> vif(train_data[c('CPI','Fuel_Price','year','Temperature','Thanksgiving','Super_Bowl','Day_number')])
    Variables       VIF
1         CPI 22.122533
2  Fuel_Price  3.359369
3        year 13.707501
4 Temperature  1.296530
5 Thanksgiving  1.057822
6  Super_Bowl  1.122562
7  Day_number 23.204299
> vif(train_data[c('CPI','Fuel_Price','year','Temperature','Thanksgiving','Super_Bowl')])
    Variables       VIF
1         CPI 10.012136
2  Fuel_Price  3.252955
3        year 12.759756
4 Temperature  1.255378
5 Thanksgiving  1.033406
6  Super_Bowl  1.112749
> vif(train_data[c('CPI','Fuel_Price','Temperature','Thanksgiving','Super_Bowl')])
    Variables      VIF
1         CPI 2.427314
2  Fuel_Price 2.482621
3 Temperature 1.155696
4 Thanksgiving 1.008151
5  Super_Bowl 1.099281
>
```

After feature reduction using VIF, the remaining Features are CPI, Fuel_Price, Temperature, Thanksgiving, Super_Bowl

Now, we perform Linear Regression and remove features with p-val>0.05, till essential features for Linear Regression model having p-value<0.05 are remaining

```
Sales_estimate<-lm(Weekly_Sales~CPI+Fuel_Price+Temperature+Thanksgiving+Super_Bowl,data=train_data)
Sales_estimate
summary(Sales_estimate)
```

```
> Sales_estimate<-lm(Weekly_Sales~CPI+Fuel_Price+Temperature+Thanksgiving+Super_Bowl,data=train_data)
> Sales_estimate

Call:
lm(formula = Weekly_Sales ~ CPI + Fuel_Price + Temperature + 
    Thanksgiving + Super_Bowl, data = train_data)

Coefficients:
 (Intercept)           CPI    Fuel_Price   Temperature  Thanksgiving    Super_Bowl  
     -699213         11506        -15186         -2714        433227         82296  

> summary(Sales_estimate)

Call:
lm(formula = Weekly_Sales ~ CPI + Fuel_Price + Temperature + 
    Thanksgiving + Super_Bowl, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-260692  -84196  -10910   63496  840658 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -699213.4   932574.6  -0.750  0.45503    
CPI            11505.7     4872.9   2.361  0.02001 *  
Fuel_Price    -15185.5    51867.6  -0.293  0.77026    
Temperature    -2714.1      989.6  -2.743  0.00714 ** 
Thanksgiving  433227.5   104163.5   4.159 6.43e-05 ***
Super_Bowl     82296.5    89209.1   0.923  0.35832    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 145400 on 108 degrees of freedom
Multiple R-squared:  0.2542,	Adjusted R-squared:  0.2197 
F-statistic: 7.361 on 5 and 108 DF,  p-value: 5.676e-06
```

```
Sales_estimate<-lm(Weekly_Sales~CPI+Temperature+Thanksgiving+Super_Bowl,data=train_data)
Sales_estimate
summary(Sales_estimate)
```

```
> Sales_estimate<-lm(Weekly_Sales~CPI+Temperature+Thanksgiving+Super_Bowl,data=train_data)
> Sales_estimate

Call:
lm(formula = Weekly_Sales ~ CPI + Temperature + Thanksgiving + 
    Super_Bowl, data = train_data)

Coefficients:
 (Intercept)           CPI   Temperature  Thanksgiving    Super_Bowl  
     -511947         10426         -2750        435187         83053  

> summary(Sales_estimate)

Call:
lm(formula = Weekly_Sales ~ CPI + Temperature + Thanksgiving + 
    Super_Bowl, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-262917  -84075   -9555   64438  839767 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -511946.9   675804.1  -0.758  0.45036    
CPI            10425.7     3170.7   3.288  0.00136 ** 
Temperature    -2750.3      977.7  -2.813  0.00582 ** 
Thanksgiving  435187.1   103511.4   4.204 5.39e-05 ***
Super_Bowl     83052.5    88796.9   0.935  0.35170    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 144800 on 109 degrees of freedom
Multiple R-squared:  0.2536,	Adjusted R-squared:  0.2262 
F-statistic: 9.258 on 4 and 109 DF,  p-value: 1.771e-06
```

```
Sales_estimate<-lm(Weekly_Sales~CPI+Temperature+Thanksgiving,data=train_data)
Sales_estimate
summary(Sales_estimate)
```

```
> Sales_estimate<-lm(Weekly_Sales~CPI+Temperature+Thanksgiving,data=train_data)
> Sales_estimate

Call:
lm(formula = Weekly_Sales ~ CPI + Temperature + Thanksgiving,
    data = train_data)

Coefficients:
 (Intercept)          CPI   Temperature  Thanksgiving
     -501424        10472         -3021        431527

> summary(Sales_estimate)

Call:
lm(formula = Weekly_Sales ~ CPI + Temperature + Thanksgiving,
    data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-271450  -84654   -6415   60996  833573

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -501423.9   675325.8  -0.742  0.45937
CPI            10472.3     3168.5   3.305  0.00128 **
Temperature    -3021.0      933.4  -3.237  0.00160 **
Thanksgiving  431526.8   103378.5   4.174    6e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 144700 on 110 degrees of freedom
Multiple R-squared:  0.2476,	Adjusted R-squared:  0.2271
F-statistic: 12.07 on 3 and 110 DF,  p-value: 6.896e-07
```

Final Linear Regression Model

CPI→ Prevailing consumer price index

Temperature→Temperature on the day of sale

Thanksgiving→ This value is ‘1’ if Thanksgiving holiday lies during the week of sale. Otherwise, it is ‘0’

Predicted Weekly Sales → Prediction of Weekly Sales

$$\boxed{Predicted\ Weekly\ Sales = 10472.3 \times CPI - 3021 \times Temperature + 431526.8 \times Thanksgiving}$$

Training Data has R^2 value of 0.248 and Adjusted R^2 value of 0.227

Performance on Test Dataset

```
sales_prediction<-predict(Sales_estimate,test_data[c('CPI','Temperature','Thanksgiving')])
testing_prediction<-cbind(sales_prediction,test_data$Weekly_Sales)
df.test.predict<-data.frame(testing_prediction)
colnames(df.test.predict)<-c('Prediction','True_Value')
MSE_value<-mean((df.test.predict$Prediction-df.test.predict$True_Value)^2)
cat("Mean Square Error is", MSE_value)
RMSE_value<-sqrt(mean((df.test.predict$Prediction-df.test.predict$True_Value)^2))
cat("Root Mean Square Error is", RMSE_value)
SSR_value<-sum((df.test.predict$Prediction-mean(df.test.predict$True_Value))^2);
SST_value<-sum((df.test.predict$True_Value-mean(df.test.predict$True_Value))^2);
test_R2_value<-SSR_value/SST_value;
cat("Coefficient of determination (R^2) on Test Data is equal to",test_R2_value )
```

```
> sales_prediction<-predict(Sales_estimate,test_data[c('CPI','Temperature','Thanksgiving')])
> testing_prediction<-cbind(sales_prediction,test_data$Weekly_Sales)
> df.test.predict<-data.frame(testing_prediction)
> colnames(df.test.predict)<-c('Prediction','True_Value')
> MSE_value<-mean((df.test.predict$Prediction-df.test.predict$True_Value)^2)
> cat("Mean Square Error is", MSE_value)
Mean Square Error is 13477465480
> RMSE_value<-sqrt(mean((df.test.predict$Prediction-df.test.predict$True_Value)^2))
> cat("Root Mean Square Error is", RMSE_value)
Root Mean Square Error is 116092.5
> SSR_value<-sum((df.test.predict$Prediction-mean(df.test.predict$True_Value))^2);
> SST_value<-sum((df.test.predict$True_Value-mean(df.test.predict$True_Value))^2);
> test_R2_value<-SSR_value/SST_value;
> cat("Coefficient of determination (R^2) on Test Data is equal to",test_R2_value )
Coefficient of determination (R^2) on Test Data is equal to 0.3594749
>
```

Test Data has R^2 value of 0.359