library(shiny)

library(shinydashboard)

library(plotly)

library(readxl)

library(dplyr)

library(reshape2)

library(lubridate)

library(ggplot2)

library(colourpicker)

library(datasets)

library(forecast)

library(randomForest)

library(prophet)

library(tidyverse)

library(data.table)

library(bindrcpp)

library(zoo)

library(lubridate)

data\_df <- read.csv("Walmart\_Store\_sales.csv")

summary(data\_df)

#get max sale for store. store 20 has max sale 301397792

library(dplyr)

by\_store <- data\_df %>%group\_by(data\_df$Store)

View(by\_store)

by\_store %>% summarise(sum\_Weekly\_Sales=sum(Weekly\_Sales))->by\_store1

View(by\_store1)

max\_store\_sale<-filter(by\_store1,by\_store1$sum\_Weekly\_Sales==max(by\_store1$sum\_Weekly\_Sales))

View(max\_store\_sale)

data\_df %>%

group\_by(Store, Weekly\_Sales)

mutate(growth =data\_df$Weekly\_Sales / lag(data\_df$Weekly\_Sales, 4) - 1)

data\_df %>%

group\_by(Weekly\_Sales) %>%

summarise(max.sales = max(Weekly\_Sales)) %>%

arrange(max.sales) %>%

summary(max.sales)

#max\_sd <- store\_sales <- aggregate(walmart\_data, by=list(walmart\_data$Store), FUN=sd, na.rm=TRUE)

#by\_store %>% summarise(sum\_Weekly\_Sales=mean(Weekly\_Sales))->by\_store2

#View(by\_store2

by\_store %>% summarise(sd\_Weekly\_Sales=sd(Weekly\_Sales))->max\_sd

View(max\_sd)

max\_sd2<-subset(max\_sd,max\_sd$sd\_Weekly\_Sales==max(max\_sd$sd\_Weekly\_Sales))

View(max\_sd2)

by\_store %>% summarise(sd\_Weekly\_Sales=sd(Weekly\_Sales))->by\_store2

View(by\_store2)

max\_sd\_store\_sale<-filter(by\_store2,by\_store2$sd\_Weekly\_Sales==max(by\_store2$sd\_Weekly\_Sales))

View(max\_sd\_store\_sale)

y1=c(391909,29661)

pie(y1 , labels = c("Holiday","Not Holiday") , col = c("cornsilk", "purple"), edges = 16)

my\_theme <- theme\_bw() +

theme(axis.title=element\_text(size=24),

plot.title=element\_text(size=36),

axis.text =element\_text(size=16))

data\_df %>%

group\_by(Store) %>%

summarise(mean.sales = mean(Weekly\_Sales)) %>%

arrange(mean.sales) %>%

mutate(store=factor(Store,levels=Store)) %>%

ggplot(aes(x=store,y=mean.sales)) +

geom\_point(color="#c60b1e") +

guides(color=FALSE) +

xlab("Store") +

ylab("Mean Sales") +

my\_theme +

theme(axis.text.x=element\_blank(), axis.ticks = element\_blank()) +

geom\_text(aes(x=c(1:45),y=mean.sales,label=Store), hjust=-.25, vjust=-2, size=2.5) +

theme(plot.background=element\_rect(fill="#c60b1e"),

panel.background=element\_rect(fill="#ffc400"),

panel.grid =element\_blank(),

axis.title =element\_text(color="#ffc400"),

axis.text =element\_text(color="#ffc400"),

plot.title =element\_text(color="#ffc400",size=32)) +

ylim(c(5000,30000)) +

ggtitle("Sales Distribution by Stores")

ggplot(data\_df,aes(x=Weekly\_Sales))+ geom\_histogram(aes(y=..density..),colour="red",fill="yellow", bins = 20)+ geom\_density(alpha=.2, fill="red")

model<-lm(Weekly\_Sales~., data\_df)

summary(model)

data\_df$Date <- as.Date(data\_df$Date,format="%d-%m-%Y")

data\_df

data\_df %>% mutate(qtr=quarters(data\_df$Date))->data\_df\_by\_qtr

view(data\_df\_by\_qtr)

data\_df\_by\_qtr\_q23<-filter(data\_df\_by\_qtr,(data\_df\_by\_qtr$qtr!='Q1'& data\_df\_by\_qtr$qtr!='Q4'& data\_df\_by\_qtr$qtr!='Q2') & year(data\_df\_by\_qtr$Date)==2012 )

View(data\_df\_by\_qtr\_q23)

by\_store\_qtr <- data\_df\_by\_qtr\_q23%>%group\_by(Store,qtr)

View(by\_store\_qtr)

by\_store\_qtr %>% summarise(sum\_Weekly\_Sales=sum(Weekly\_Sales))->by\_store\_qtr2

View(by\_store\_qtr2)

by\_store\_qtr2 %>% arrange(Store,qtr)

#Growth rate = (Present value - Past value)/ Past Value \* 100

growth\_rate = by\_store\_qtr2 %>%

# first sort by store and qtr

arrange(Store,qtr) %>%

mutate(Diff\_growth = sum\_Weekly\_Sales - lag(sum\_Weekly\_Sales), # Difference in route between years

Rate\_percent = (Diff\_growth)/lag(sum\_Weekly\_Sales) \* 100) # growth rate in percent

View(growth\_rate)

#get store for max growth

max\_growth\_store<-subset(growth\_rate,Rate\_percent==max(growth\_rate$Rate\_percent, na.rm = TRUE))

View(max\_growth\_store)

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

walmart\_nh\_data <- filter(data\_df, data\_df$Holiday\_Flag == 0)#non holiday data <- data\_df%>%group\_by(data\_df$Store)

mean(walmart\_nh\_data$Weekly\_Sales)

walmart\_h\_data <- filter(data\_df, data\_df$Holiday\_Flag ==1)# holiday data <- data\_df%>%group\_by(data\_df$Store)

walmart\_h\_data1 <- filter(walmart\_h\_data, walmart\_h\_data$Weekly\_Sales>mean(walmart\_nh\_data$Weekly\_Sales))

unique(walmart\_h\_data1$Date)

#total <- merge(walmart\_h\_sum\_data,walmart\_nh\_mean\_data,by="Store")

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

library(ggplot2)

library(scales)

monthly\_sales <- data\_df%>%

group\_by(floor\_date(data\_df$Date,unit = 'month'))%>% summarise(sales = sum(Weekly\_Sales))

names(monthly\_sales)[1] <- "MonthWise"

View(monthly\_sales)

# graph by month:

ggplot(data = monthly\_sales,

aes(MonthWise, sales)) +

stat\_summary(fun.y = sum, # adds up all observations for the month

geom = "bar") + # or "line"

scale\_x\_date(

labels = date\_format("%Y-%m"),

breaks = "1 month") # custom x-axis labels

ggplot(data = monthly\_sales,

aes(MonthWise, sales)) +

stat\_summary(fun.y = sum, # adds up all observations for the month

geom = "line") + # or "line"

scale\_x\_date(

labels = date\_format("%Y-%m"),

breaks = "1 month") # custom x-axis labels

#semseter view

ggplot(data = monthly\_sales,

aes(MonthWise, sales)) +

stat\_summary(fun.y = sum, # adds up all observations for the month

geom = "bar") + # or "line"

scale\_x\_date(

labels = date\_format("%Y-%m"),

breaks = "6 month") # custom x-axis labels

ggplot(data = monthly\_sales,

aes(MonthWise, sales)) +

stat\_summary(fun.y = sum, # adds up all observations for the month

geom = "line") + # or "line"

scale\_x\_date(

labels = date\_format("%Y-%m"),

breaks = "6 month") # custom x-axis labels

ts\_data <- ts(data\_df$Weekly\_Sales,start = 2010,end = 2012,frequency = 12)

plot(ts\_data)

fit\_arima <- auto.arima(ts\_data,seasonal = TRUE,stepwise = FALSE,stationary = FALSE,allowmean = TRUE,parallel = TRUE)

plot(forecast(fit\_arima,h=10))

w <- prophet(ts\_data)