Data visualization project

Final project

2023-12-01

```
# mergeing features2,traindata1 and stores into single csv file
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
```

```
dataset1 <- read.csv("features2.csv")
dataset2 <- read.csv("traindata1.csv")
dataset3 <- read.csv("stores.csv")

merged_data <- dataset1 %>%
  left_join(dataset2, by = c("Store", "Date"))
merged_data1 <- merged_data %>%
  left_join(dataset3, by = "Store")
names(merged_data1)
```

```
##  [1] "Store"        "Date"         "Temperature"  "Fuel_Price"   "MarkDown1"
##  [6] "MarkDown2"    "MarkDown3"    "MarkDown4"    "MarkDown5"    "CPI"
## [11] "Unemployment" "IsHoliday.x"  "Dept"         "Weekly_Sales" "IsHoliday.y"
## [16] "Type"         "Size"
```

```
head(merged_data1,2)
```

```
##   Store       Date Temperature Fuel_Price MarkDown1 MarkDown2 MarkDown3
## 1     1 05/02/2010       42.31      2.572        NA        NA        NA
## 2     1 05/02/2010       42.31      2.572        NA        NA        NA
##   MarkDown4 MarkDown5      CPI Unemployment IsHoliday.x Dept Weekly_Sales
## 1        NA        NA 211.0964        8.106       FALSE    1     24924.50
## 2        NA        NA 211.0964        8.106       FALSE    2     50605.27
##   IsHoliday.y Type   Size
## 1       FALSE    A 151315
## 2       FALSE    A 151315
```

```r
#Replacing nan value to 0 using user-defined function for smoother visualizations
NAN <- function(data) {
  columns <- c('MarkDown1', 'MarkDown2', 'MarkDown3', 'MarkDown4', 'MarkDown5')
  data[columns] <- replace(data[columns], is.na(data[columns]), 0)
  return(data)
}
data_without_Nan <-NAN(merged_data1)
head(data_without_Nan,2)
```

```
##   Store       Date Temperature Fuel_Price MarkDown1 MarkDown2 MarkDown3
## 1     1 05/02/2010       42.31      2.572         0         0         0
## 2     1 05/02/2010       42.31      2.572         0         0         0
##   MarkDown4 MarkDown5      CPI Unemployment IsHoliday.x Dept Weekly_Sales
## 1         0         0 211.0964        8.106       FALSE    1     24924.50
## 2         0         0 211.0964        8.106       FALSE    2     50605.27
##   IsHoliday.y Type   Size
## 1       FALSE    A 151315
## 2       FALSE    A 151315
```

```r
#Converting IsHoliday column boolean values to 1,0
data_without_Nan <- data_without_Nan %>%
  mutate(IsHoliday = as.integer(IsHoliday.x))
nullNA<- data_without_Nan[!rowSums(is.na(data_without_Nan)), ]
colSums(is.na(nullNA)) #Observed no NA values
```

```
##        Store         Date  Temperature   Fuel_Price    MarkDown1    MarkDown2
##            0            0            0            0            0            0
##    MarkDown3    MarkDown4    MarkDown5          CPI Unemployment  IsHoliday.x
##            0            0            0            0            0            0
##         Dept Weekly_Sales  IsHoliday.y         Type         Size    IsHoliday
##            0            0            0            0            0            0
```

```r
head(data_without_Nan,2)
```

```
##   Store       Date Temperature Fuel_Price MarkDown1 MarkDown2 MarkDown3
## 1     1 05/02/2010       42.31      2.572         0         0         0
## 2     1 05/02/2010       42.31      2.572         0         0         0
##   MarkDown4 MarkDown5      CPI Unemployment IsHoliday.x Dept Weekly_Sales
## 1         0         0 211.0964        8.106       FALSE    1     24924.50
## 2         0         0 211.0964        8.106       FALSE    2     50605.27
##   IsHoliday.y Type   Size IsHoliday
## 1       FALSE    A 151315         0
## 2       FALSE    A 151315         0
```

```r
#Extracting day,month and year
nullNA$Date <- as.Date(nullNA$Date, format="%d/%m/%Y")
nullNA$Day <- format(nullNA$Date, "%d")
nullNA$Month <- format(nullNA$Date, "%m")
nullNA$Year <- format(nullNA$Date, "%Y")
Walmart_dataset = nullNA
# write walmart dataset
```

```r
#write.csv(Walmart_dataset, "Walmart_dataset.csv", row.names = FALSE)
#Read Walmart_dataset
Walmart_dataset<- read.csv("Walmart_dataset.csv")

#1. How many stores are present in data?
Walmart_dataset %>% summarize(Total_stores = n_distinct(Store))
```

```
##   Total_stores
## 1           30
```

```r
#2. How many departments are present in data?
Walmart_dataset %>% summarize(Total_Dept = n_distinct(Dept))
```

```
##   Total_Dept
## 1         80
```

```r
#3. How many store-department combinations have all weeks of sales data?
Walmart_dataset %>% summarize(min_date = min(Date), max_date = max(Date),
                   total_weeks = difftime(min_date,max_date, unit = "weeks"))
```

```
##     min_date   max_date    total_weeks
## 1 2010-02-05 2012-10-26 -141.994 weeks
```

```r
#which store has max sales
library(ggplot2)
library(dplyr)
store_sales <- Walmart_dataset %>%
  group_by(Store) %>%
  summarise(Total_Sales = sum(Weekly_Sales, na.rm = TRUE))
max_sales_store <- store_sales[which.max(store_sales$Total_Sales), ]
p <- ggplot(store_sales, aes(x = factor(Store), y = Total_Sales)) +
  geom_bar(stat = 'identity', fill = 'steelblue') +
  #geom_text(aes(label = Total_Sales), vjust = -0.3, size = 2.5) +
  theme_minimal() +
  labs(x = 'Store Number', y = 'Total Sales', title = 'Total Sales by Store') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
p <- p + geom_col(data = max_sales_store, aes(x = factor(Store),
                                              y = Total_Sales), fill = 'red')
ggsave("D:/R/work/Visuals DV/Hist.png", p)
```

```
## Saving 6.5 x 4.5 in image
```

```r
#Density Plot for Store 20
library(scales)
Store_20 <- Walmart_dataset[Walmart_dataset$Store == 20, ]
q<- ggplot(Store_20, aes(x = Weekly_Sales)) +
  geom_density(color = "darkblue", fill = "lightblue", alpha = 0.2) +
  geom_vline(aes(xintercept = mean(Weekly_Sales)), color = "steelblue",
             linetype = "dashed", size = 1) +
  theme(axis.text.x = element_text(vjust = 0.5, hjust = 0.5)) +
  scale_x_continuous(labels = label_number(suffix = " M", scale = 1e-6)) +
```

```
  ggtitle('Store 20 Sales Distribution') +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Weekly Sales") + ylab("Density")
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
ggsave("D:/R/work/Visuals DV/store20.png", q)
```

```
## Saving 6.5 x 4.5 in image
```

```
# #which year has max sales
# library(ggplot2)
# library(dplyr)
# year_2010 <- Walmart_dataset %>%
#   filter(format(Date, "%Y") %in% c("2010"))
# year_2011 <- Walmart_dataset %>%
#   filter(format(Date, "%Y") %in% c("2011"))
# year_2012 <- Walmart_dataset %>%
#   filter(format(Date, "%Y") %in% c("2012"))
# combined_data <- rbind(
#   mutate(year_2010, Quarter = "Q4 2010"),
#   mutate(year_2011, Quarter = "Q4 2011"),
#   mutate(year_2012, Quarter = "Q4 2012")
# )
# r=ggplot(Walmart_dataset, aes(x = Year, y = Weekly_Sales, fill = Year)) +
#   geom_bar(stat = "identity") +
#   labs(title = "Sales for the Years 2010, 2011 and 2012",
#        x = "Year", y = "Total Sales") +
#   theme_minimal()+
#   coord_cartesian(xlim = c(0,5))
# ggsave("D:/R/work/Visuals DV/maxyear.png", r)

#which store type have max sales and min sales
library(ggplot2)
s= ggplot(Walmart_dataset, aes(x = Type, y = Weekly_Sales, fill = Type)) +
  geom_boxplot(outlier.shape = NA) +
  labs(title = "Sales Comparison by Store Type",
       x = "Store Type",
       y = "Weekly Sales") +
  theme_minimal() +
  facet_wrap(~Type, scales = "free_y")+
  coord_cartesian(ylim = c(0, 50000))
ggsave("D:/R/work/Visuals DV/maxminstoretype.png", s)
```

```
## Saving 6.5 x 4.5 in image
```

```r
#Is there a relation between average temperature of the week and sales?
library(ggplot2)
t =ggplot(Walmart_dataset, aes(x = Temperature, y = Weekly_Sales,
                               color= factor(IsHoliday))) +
  geom_point() +
  labs(title = "Sales vs Temperature with Holiday Indicator",
       x = "Temperature",
       y = "Weekly Sales",
       color = "Is Holiday") +
  theme_minimal()
ggsave("D:/R/work/Visuals DV/tempsales.png", t)
```

```
## Saving 6.5 x 4.5 in image
```

```r
#Sales Comparison during Holidays vs. Non-Holidays
a=ggplot(Walmart_dataset, aes(x = factor(IsHoliday),
                              y = Weekly_Sales, fill = factor(IsHoliday))) +
  geom_violin() +
  facet_wrap(~Type, scales = "free_y") +
  labs(title = "Sales Comparison during Holidays vs. Non-Holidays",
       x = "Is Holiday",
       y = "Weekly Sales") +
  theme_minimal()+
  coord_cartesian(ylim=c(-10000,100000))
ggsave("D:/R/work/Visuals DV/holinotholisales.png", a)
```

```
## Saving 6.5 x 4.5 in image
```

```r
#Are there distinct seasonal patterns in sales?
b=ggplot(Walmart_dataset, aes(x = Month, y = Weekly_Sales,
                              group = Year, color = factor(Year))) +
  geom_line() +facet_wrap(~Year, scales = "free_y") +
  labs(title = "Seasonal Sales Patterns",
       x = "Month",
       y = "Weekly Sales",
       color = "Year") +
  theme_minimal()+  scale_color_manual(values = c("blue", "green", "red"))
ggsave("D:/R/work/Visuals DV/spsales.png", b)
```

```
## Saving 6.5 x 4.5 in image
```

```r
# pie chart for store type distribution
type_percentages <- prop.table(table(Walmart_dataset$Type)) * 100
c=ggplot(NULL, aes(x = "", y = type_percentages,
                   fill = names(type_percentages))) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y") +
  labs(title = "Store Type Distribution",
       fill = "Store Type") +
  scale_fill_manual(values = c("A" = "skyblue", "B" = "lightgreen",
                               "C" = "purple")) +
  theme_minimal()
ggsave("D:/R/work/Visuals DV/pie.png", c)
```

```
## Saving 6.5 x 4.5 in image
## Don't know how to automatically pick scale for object of type <table>.
## Defaulting to continuous.
```

```
#Distribution of stores by size
library(ggplot2)
d=ggplot(Walmart_dataset, aes(x = Size,fill=Type)) +
  geom_histogram(binwidth = 6000) + facet_grid(Type~.)
ggsave("D:/R/work/Visuals DV/storesize.png", d)
```

```
## Saving 6.5 x 4.5 in image
```

```
#relation between CPI and sales
library(ggplot2)
e=ggplot(Walmart_dataset, aes(x = CPI, y = Weekly_Sales)) +
  geom_point(alpha = 0.5, color = "blue") +
  labs(title = "Scatter Plot: CPI vs. Weekly Sales",
       x = "CPI",
       y = "Weekly Sales") +
  theme_minimal()
ggsave("D:/R/work/Visuals DV/salecpi.png", e)
```

```
## Saving 6.5 x 4.5 in image
```

```
#Department sales in each store type
department_sales <- aggregate(Weekly_Sales ~ Type + Dept,
                              data = Walmart_dataset, sum)
department_sales <- department_sales[order(department_sales$Type,
                                           -department_sales$Weekly_Sales), ]
g=ggplot(department_sales, aes(x = Dept, y = Weekly_Sales, color = Type)) +
  geom_point() +
  geom_line() +
  labs(title = "Department Sales in Each Store Type",
       x = "Department",
       y = "Total Weekly Sales") +
  theme_minimal()
ggsave("D:/R/work/Visuals DV/storetype_totalsales.png", g)
```

```
## Saving 6.5 x 4.5 in image
```

```
# Scatter plot to visualize the impact of Unemployment and CPI on Weekly Sales
h=ggplot(Walmart_dataset, aes(x = Unemployment, y = Weekly_Sales,
                              color = CPI)) +
  geom_point() +
  labs(title = "Unemployment and CPI Impact on Weekly Sales",
       x = "Unemployment",
       y = "Weekly Sales",
       color = "CPI") +
  theme_minimal()+
  scale_color_gradientn(colors = viridisLite::viridis(3))
ggsave("D:/R/work/Visuals DV/unempCPI.png", h)
```

```
## Saving 6.5 x 4.5 in image
```

```
#bar plot for department sales according to top 10 stores by total sales
library(ggplot2)
department_sales_by_store <- aggregate(Weekly_Sales ~ Dept + Store + Type,
                                       data = Walmart_dataset, sum)
top_stores <- head(department_sales_by_store[order(
  -department_sales_by_store$Weekly_Sales), ], 5)
Walmart_top5 <- subset(Walmart_dataset, Store %in% top_stores$Store)
f=ggplot(Walmart_top5, aes(x = Dept, y = Weekly_Sales, fill = Dept)) +
  geom_bar(stat = "summary", fun = "sum") +
  labs(title = "Department Sales According to Top 5 Stores",
       x = "Department",
       y = "Total Weekly Sales") +
  theme_minimal() +
  facet_wrap(~Store + Type, scales = "free_y")
ggsave("D:/R/work/Visuals DV/salecpi.png", f)
```

```
## Saving 6.5 x 4.5 in image
```

```
# created a loop function to get top 10,20,30 performing departments by
#weekly sales
library(ggplot2)
Top_perform_dept <- function(N) {
  top_departments <- head(department_sales[order(-department_sales$Weekly_Sales)
                                           , ], N)
  ggplot(top_departments, aes(x = reorder(Dept, -Weekly_Sales), y = Weekly_Sales
                              , fill = Dept)) +
    geom_bar(stat = "identity") +
    labs(title = paste("Top", N, "Performing Departments by Weekly Sales"),
         x = "Department",
         y = "Total Weekly Sales") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}
for (N in c(10, 20, 30)) {
  print(Top_perform_dept(N))
}
```