Project Superstore

Group 1

2023-06-01

```r
#install.packages("treemapify")
#install.packages("waffle")
#install.packages("plotly")

# Loading all required libraries
library(tidyverse)
library(lubridate)
library(treemapify)
library(waffle)
library(dplyr)
library(ggplot2)
library(viridis)
library(magrittr)
library(mapproj)
library(maps)
library(ggrepel)
library(httr)
library(readr)
library(scales)
library(plotly)

# Loading Data from csv file
Store <- read.csv("Sample - Superstore.csv", header = TRUE)

# Viewing the first 6 rows of the dataset
head(Store,5)
```

```
##   Row.ID       Order.ID Order.Date  Ship.Date      Ship.Mode Customer.ID
## 1      1 CA-2016-152156  11/8/2016 11/11/2016   Second Class    CG-12520
## 2      2 CA-2016-152156  11/8/2016 11/11/2016   Second Class    CG-12520
## 3      3 CA-2016-138688  6/12/2016  6/16/2016   Second Class    DV-13045
## 4      4 US-2015-108966 10/11/2015 10/18/2015 Standard Class    SO-20335
## 5      5 US-2015-108966 10/11/2015 10/18/2015 Standard Class    SO-20335
##     Customer.Name   Segment       Country            City      State
## 1     Claire Gute  Consumer United States       Henderson   Kentucky
## 2     Claire Gute  Consumer United States       Henderson   Kentucky
## 3 Darrin Van Huff Corporate United States     Los Angeles California
## 4  Sean O’Donnell  Consumer United States Fort Lauderdale    Florida
## 5  Sean O’Donnell  Consumer United States Fort Lauderdale    Florida
##   Postal.Code Region      Product.ID        Category Sub.Category
## 1       42420  South FUR-BO-10001798       Furniture    Bookcases
## 2       42420  South FUR-CH-10000454       Furniture       Chairs
## 3       90036   West OFF-LA-10000240 Office Supplies       Labels
## 4       33311  South FUR-TA-10000577       Furniture       Tables
## 5       33311  South OFF-ST-10000760 Office Supplies      Storage
##                                                   Product.Name    Sales Quantity
## 1                           Bush Somerset Collection Bookcase 261.9600        2
## 2 Hon Deluxe Fabric Upholstered Stacking Chairs, Rounded Back 731.9400        3
## 3   Self-Adhesive Address Labels for Typewriters by Universal  14.6200        2
## 4               Bretford CR4500 Series Slim Rectangular Table 957.5775        5
## 5                              Eldon Fold ’N Roll Cart System  22.3680        2
##   Discount    Profit
## 1     0.00   41.9136
## 2     0.00  219.5820
## 3     0.00    6.8714
## 4     0.45 -383.0310
## 5     0.20    2.5164
```

```
# Dataset Dimension
dim(Store)
```

```
## [1] 9994   21
```

```
# Details of column types
str(Store)
```

```
## ’data.frame':    9994 obs. of  21 variables:
##  $ Row.ID       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Order.ID     : chr  "CA-2016-152156" "CA-2016-152156" "CA-2016-138688" "US-2015-108966" ...
##  $ Order.Date   : chr  "11/8/2016" "11/8/2016" "6/12/2016" "10/11/2015" ...
##  $ Ship.Date    : chr  "11/11/2016" "11/11/2016" "6/16/2016" "10/18/2015" ...
##  $ Ship.Mode    : chr  "Second Class" "Second Class" "Second Class" "Standard Class" ...
##  $ Customer.ID  : chr  "CG-12520" "CG-12520" "DV-13045" "SO-20335" ...
##  $ Customer.Name: chr  "Claire Gute" "Claire Gute" "Darrin Van Huff" "Sean O’Donnell" ...
##  $ Segment      : chr  "Consumer" "Consumer" "Corporate" "Consumer" ...
##  $ Country      : chr  "United States" "United States" "United States" "United States" ...
##  $ City         : chr  "Henderson" "Henderson" "Los Angeles" "Fort Lauderdale" ...
##  $ State        : chr  "Kentucky" "Kentucky" "California" "Florida" ...
##  $ Postal.Code  : int  42420 42420 90036 33311 33311 90032 90032 90032 90032 90032 ...
##  $ Region       : chr  "South" "South" "West" "South" ...
##  $ Product.ID   : chr  "FUR-BO-10001798" "FUR-CH-10000454" "OFF-LA-10000240" "FUR-TA-10000577" ...
##  $ Category     : chr  "Furniture" "Furniture" "Office Supplies" "Furniture" ...
##  $ Sub.Category : chr  "Bookcases" "Chairs" "Labels" "Tables" ...
##  $ Product.Name : chr  "Bush Somerset Collection Bookcase" "Hon Deluxe Fabric Upholstered Stacking Ch
##  $ Sales        : num  262 731.9 14.6 957.6 22.4 ...
##  $ Quantity     : int  2 3 2 5 2 7 4 6 3 5 ...
##  $ Discount     : num  0 0 0 0.45 0.2 0 0 0.2 0.2 0 ...
##  $ Profit       : num  41.91 219.58 6.87 -383.03 2.52 ...
```

```
# Select only the numeric columns
numeric_cols <- Store[, sapply(Store, is.numeric)]
# Dataset datatype Summary
summary(numeric_cols)
```

```
##      Row.ID      Postal.Code        Sales              Quantity
##  Min.   :   1   Min.   : 1040   Min.   :    0.444   Min.   : 1.00
##  1st Qu.:2499   1st Qu.:23223   1st Qu.:   17.280   1st Qu.: 2.00
##  Median :4998   Median :56430   Median :   54.490   Median : 3.00
##  Mean   :4998   Mean   :55190   Mean   :  229.858   Mean   : 3.79
##  3rd Qu.:7496   3rd Qu.:90008   3rd Qu.:  209.940   3rd Qu.: 5.00
##  Max.   :9994   Max.   :99301   Max.   :22638.480   Max.   :14.00
##     Discount          Profit
##  Min.   :0.0000   Min.   :-6599.978
##  1st Qu.:0.0000   1st Qu.:    1.729
##  Median :0.2000   Median :    8.666
##  Mean   :0.1562   Mean   :   28.657
##  3rd Qu.:0.2000   3rd Qu.:   29.364
##  Max.   :0.8000   Max.   : 8399.976
```

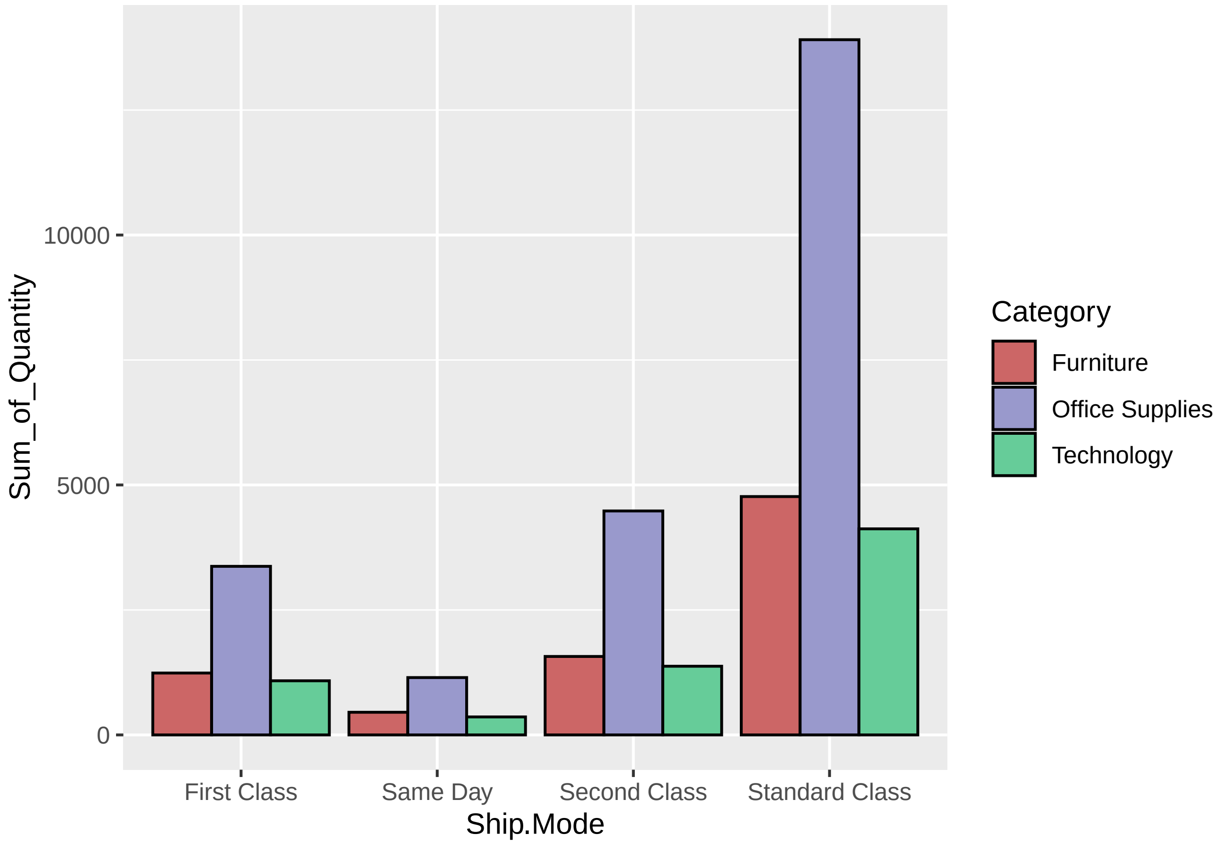
```r
# Missing Values
sapply(Store, function(x) sum(is.na(x)))
```

```
##        Row.ID      Order.ID    Order.Date     Ship.Date     Ship.Mode
##             0             0             0             0             0
##   Customer.ID Customer.Name       Segment       Country          City
##             0             0             0             0             0
##         State   Postal.Code        Region    Product.ID      Category
##             0             0             0             0             0
##  Sub.Category  Product.Name         Sales      Quantity      Discount
##             0             0             0             0             0
##        Profit
##             0
```

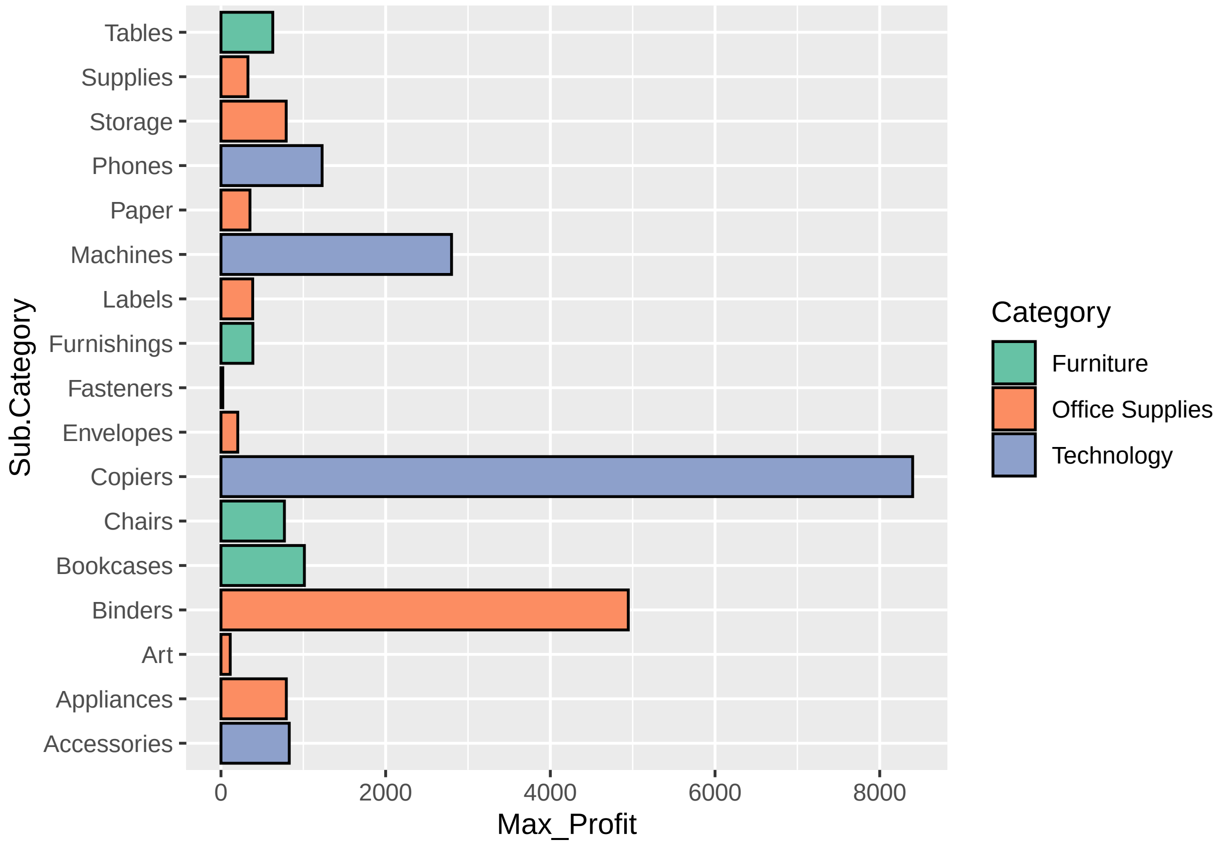
```r
# Maximum Sales in each State
state_max_sales <- Store %>%
  group_by(State) %>%
  summarise(Maximum_Sales = max(Sales),
            .groups = "drop") %>%
  arrange(desc(Maximum_Sales))

#top 5 states with max sales
head(state_max_sales)
```

```
## # A tibble: 6 x 2
##   State      Maximum_Sales
##   <chr>              <dbl>
## 1 Florida           22638.
## 2 Indiana           17500.
## 3 Washington        14000.
## 4 New York          11200.
## 5 Delaware          10500.
## 6 Michigan           9893.
```

```r
# Total quantity of each category across different shipment modes
mode_quantity <- Store %>%
  group_by(Ship.Mode, Category) %>%
  summarize(Sum_of_Quantity = sum(Quantity),
            .groups = 'drop')

ggplot(mode_quantity, aes(fill = Category, y = Sum_of_Quantity, x = Ship.Mode)) +
  geom_bar(position = "dodge", stat = "identity", color = "black") +
  scale_fill_manual(values=c("#CC6666", "#9999CC", "#66CC99"))
```



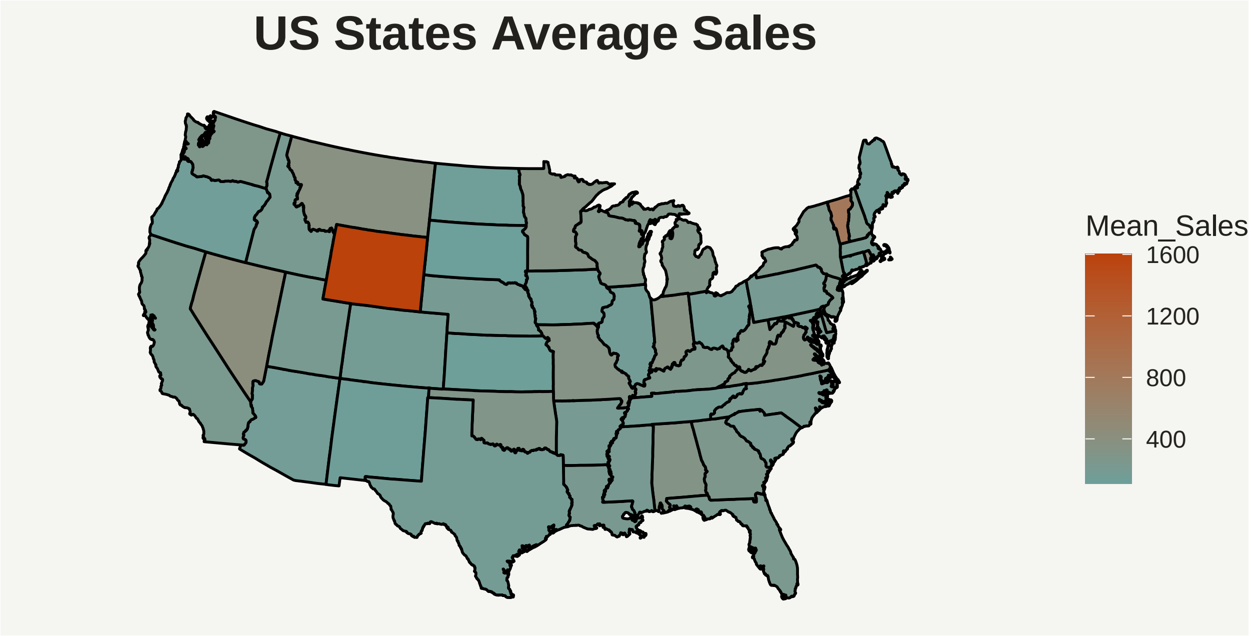
```
# Maximum Profit for different categires and sub-categories
max_profit <- Store %>%
  group_by(Category, Sub.Category) %>%
  summarize(Max_Profit = max(Profit),
            .groups = "drop")

ggplot(max_profit, aes(fill = Category, y = Sub.Category, x = Max_Profit)) +
  geom_bar(position = "dodge", stat = "identity", color = "black") +
  scale_fill_brewer(palette="Set2")
```



```
#Group by state to get the average of sales and profit
agg_tbl <- Store %>%
  group_by(State) %>%
  summarise(across(c(Sales, Profit), mean)) %>%
  rename (Mean_Sales = Sales) %>%
  rename(Mean_Profit = Profit)

agg_tbl <- agg_tbl %>%mutate(State = tolower((State)))

#Top 2 states with high sales
head(agg_tbl[order(-agg_tbl$Mean_Sales), ],2)
```

```
## # A tibble: 2 x 3
##   State   Mean_Sales Mean_Profit
##   <chr>        <dbl>       <dbl>
## 1 wyoming      1603.        100.
## 2 vermont       812.        204.
```

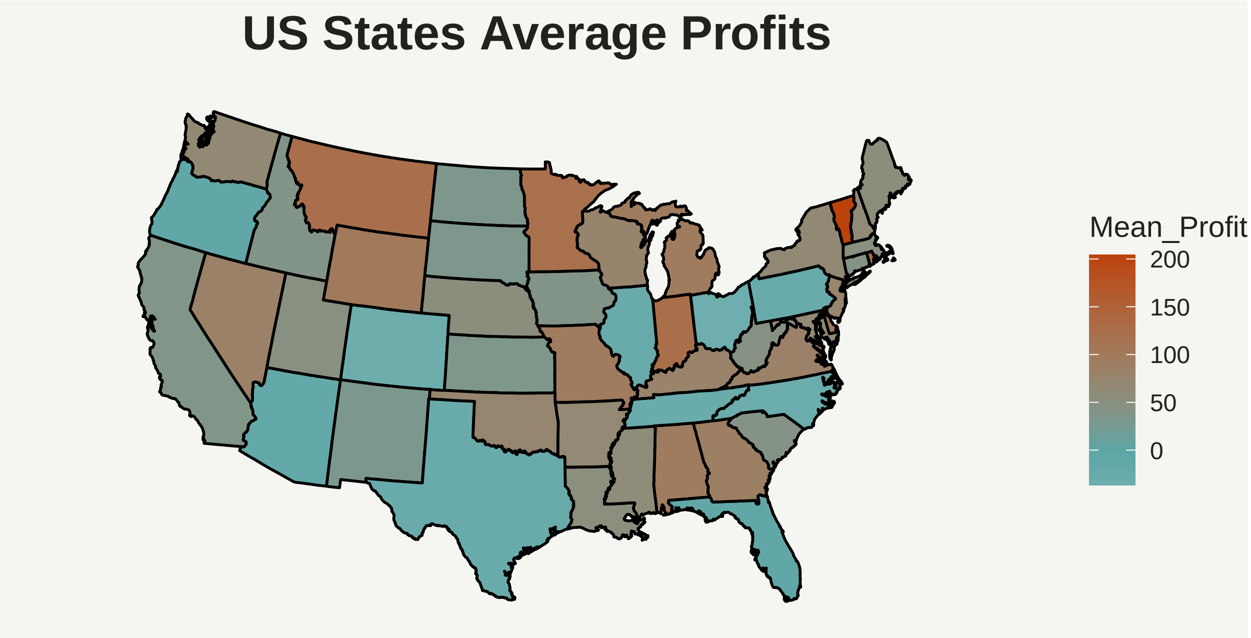
```
states_map <- map_data("state")

#Merge with State map to get the latitude and longitude to plot the graph
sales_map <- merge(agg_tbl, states_map, by.x = "State", by.y = "region", all.x = TRUE)

plot1 <- ggplot(sales_map, aes(x = long, y = lat, group = group, fill = Mean_Sales)) +
  geom_polygon(colour = "black") +
  coord_map("polyconic") +
  scale_fill_gradient2(low = "#B5D6D6", mid = "#5DA5A5", high = "#BB420B") +
  theme_void() +
  theme(
    text = element_text(color = "#22211d"),
    plot.background = element_rect(fill = "#f5f5f2", color = NA),
    panel.background = element_rect(fill = "#f5f5f2", color = NA),
    legend.background = element_rect(fill = "#f5f5f2", color = NA),
    ) +
  labs(title='US States Average Sales') +
  theme(plot.title=element_text(size=18, face='bold', hjust=0.5))

plot1
```



```
#show top 2 highly profitable states
head(agg_tbl[order(-agg_tbl$Mean_Profit), ],2)
```

```
## # A tibble: 2 x 3
##   State        Mean_Sales Mean_Profit
##   <chr>             <dbl>       <dbl>
## 1 vermont            812.        204.
## 2 rhode island       404.        130.
```

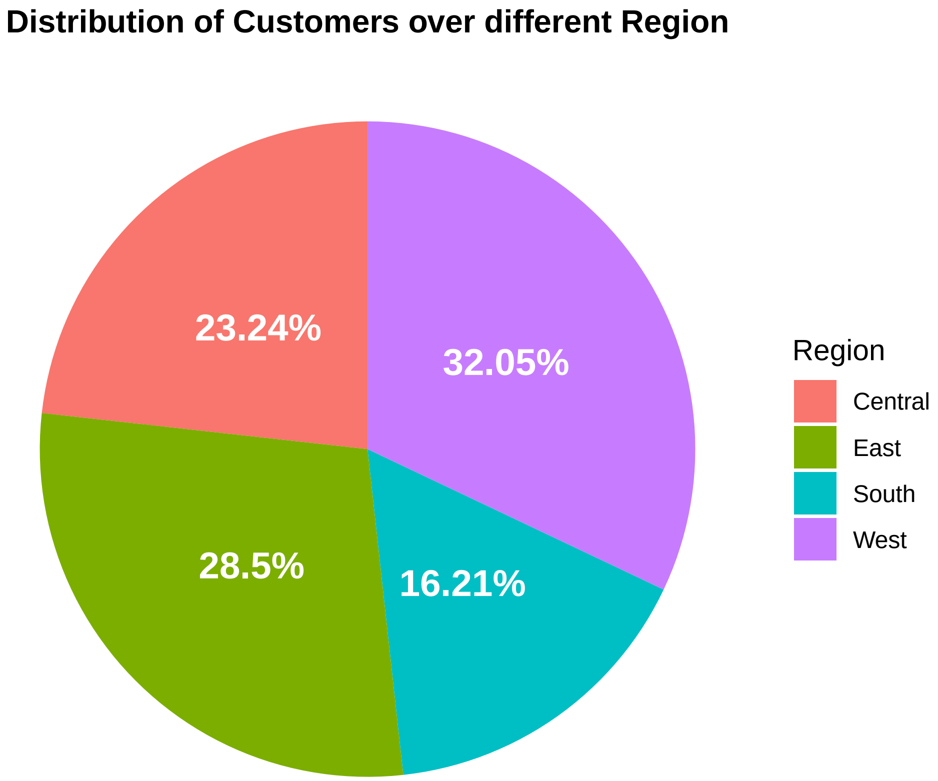
```
plot2 <- ggplot(sales_map, aes(x = long, y = lat, group = group, fill = Mean_Profit)) +
  geom_polygon(colour = "black") +
  coord_map("polyconic") +
  scale_fill_gradient2(low = "#B5D6D6", mid = "#5DA5A5", high = "#BB420B") +
  theme_void() +
  theme(
    text = element_text(color = "#22211d"),
    plot.background = element_rect(fill = "#f5f5f2", color = NA),
    panel.background = element_rect(fill = "#f5f5f2", color = NA),
    legend.background = element_rect(fill = "#f5f5f2", color = NA),
    ) +
  labs(title='US States Average Profits') +
  theme(plot.title=element_text(size=18, face='bold', hjust=0.5))

plot2
```



```
#group the store by Region
Store_region <- Store %>%
    group_by(Region) %>%
    summarize(count=n())

Store_region
```

```
## # A tibble: 4 x 2
##   Region  count
##   <chr>   <int>
## 1 Central  2323
## 2 East     2848
## 3 South    1620
## 4 West     3203
```

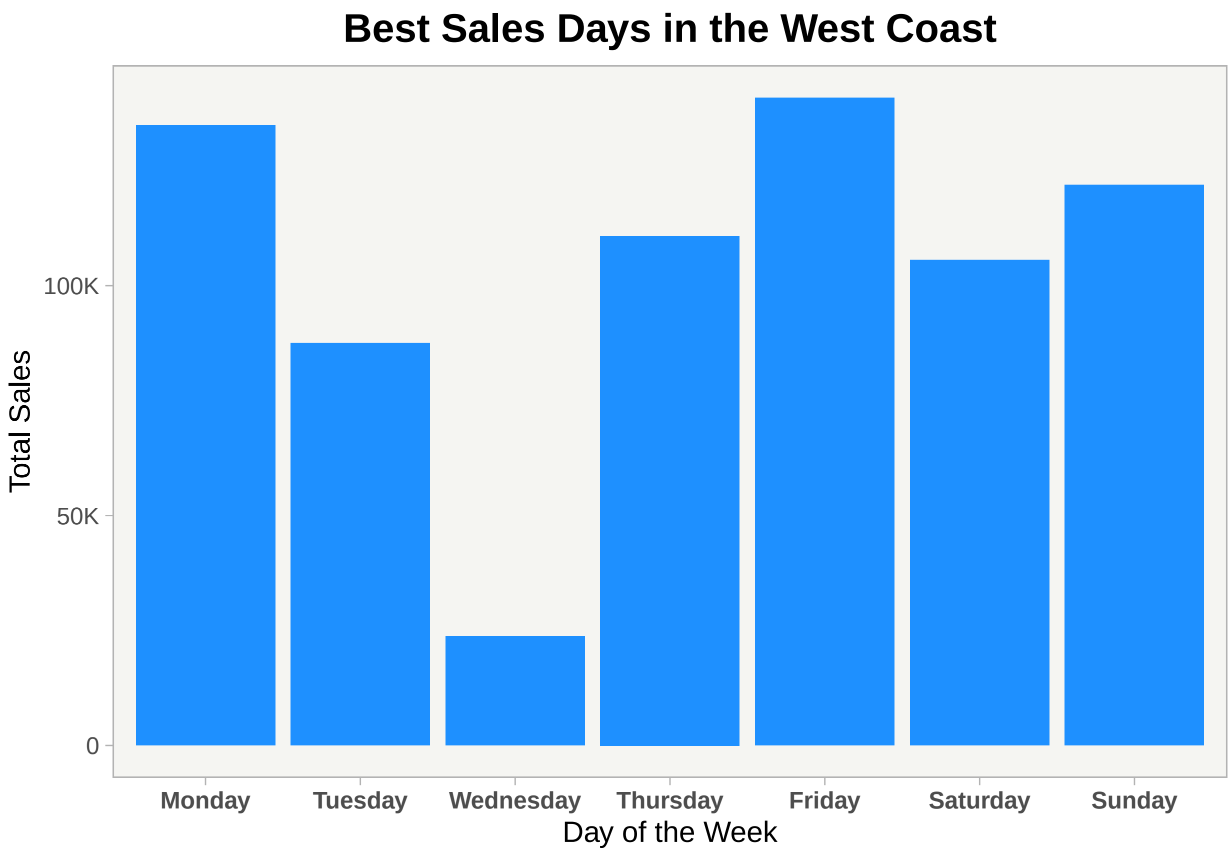
```
# Calculate proportions
Store_region %<>%
    mutate(position=cumsum(Store_region$count)-(0.5*count),
           percent=(count/sum(Store_region$count)*100)) %>%
    # To be able to use position_stack in geom_text
    as.data.frame()

plot3<-Store_region %>%
    ggplot(aes(x='', y=count, fill=Region))+
    geom_bar(stat='identity', width=1)+
    geom_text(aes(label=paste0(round(percent, 2), '%')), size=5, fontface='bold', color='white', positi
    coord_polar(theta='y', start=0)+
    labs(title='Distribution of Customers over different Region')+
    theme_void()+
    theme(plot.title=element_text(size=12, hjust=0.5, face='bold'))

plot3
```



```r
#filter by stores on west
Store_west <- Store %>% filter(Region == "West")
Store_west$Order.Date<-mdy(Store_west$Order.Date);
Store_west$Day<-weekdays(Store_west$Order.Date);

Day_agg <- Store_west %>%
  group_by(Day) %>%
  summarise(across(c(Sales, Profit), sum)) %>%
  arrange(Day);

week <- c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday")

plot4 <- ggplot(data = Day_agg, aes(x = factor(Day,level=week), y = Sales)) +
         geom_bar(stat = "sum", fill = "dodgerblue") +
         scale_y_continuous(labels = label_number_si()) +
         labs(title = 'Best Sales Days in the West Coast',x = 'Day of the Week',y = 'Total Sales') +
         theme_light()+
         theme(plot.title=element_text(size=15, face='bold', hjust=0.5),axis.text.x=element_text(face='
               legend.position = "none",panel.background = element_rect(fill = "#f5f5f2"),panel.grid =
```

```
## Warning: ‘label_number_si()‘ was deprecated in scales 1.2.0.
## i Please use the ‘scale_cut‘ argument of ‘label_number()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

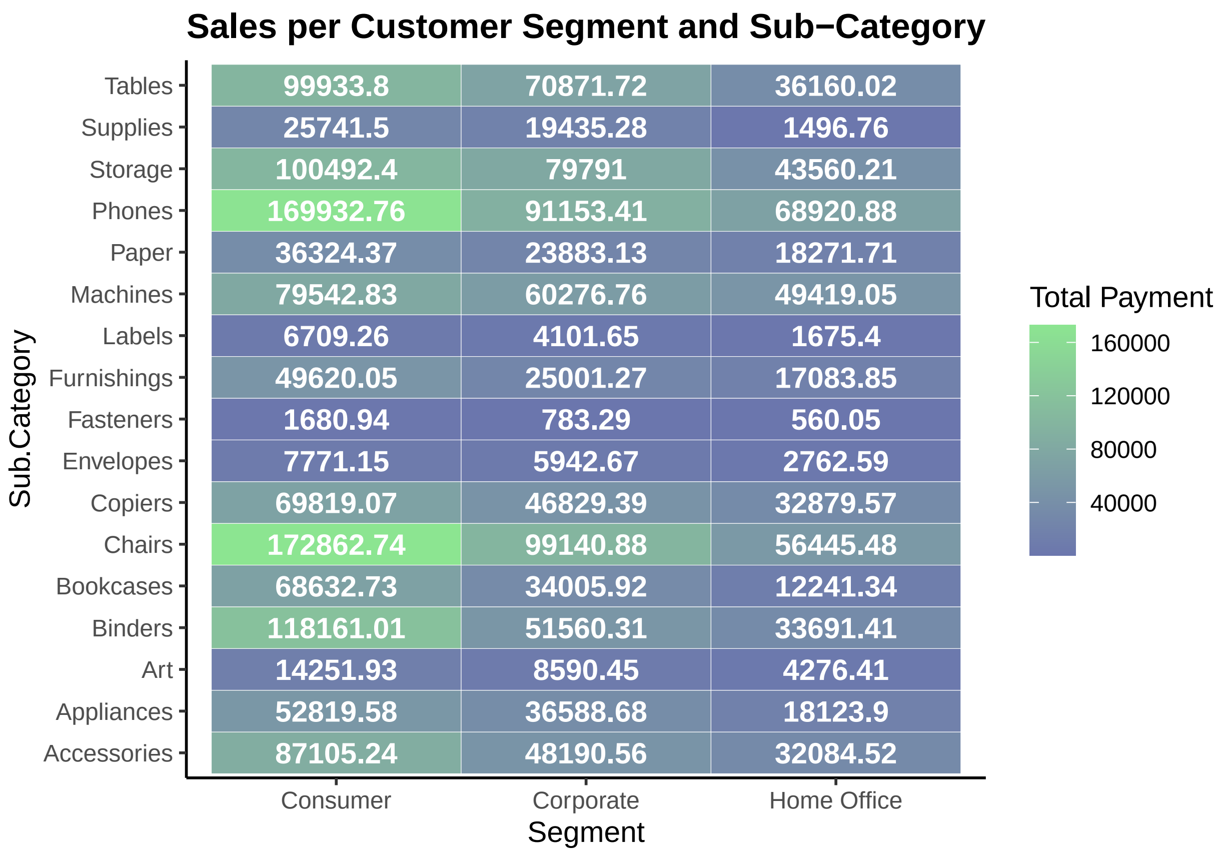
```r
plot4
```



```
plot5 <- Store %>%
  group_by(`Segment`, `Sub.Category`) %>%
  summarize("Total Payment"= sum(Sales)) %>%
  ggplot(aes(x=Segment, y=`Sub.Category`, fill=`Total Payment`)) +
  scale_fill_viridis_b(option = 'D') +
  geom_tile(color='white') +
  geom_text(aes(label=paste0(round(`Total Payment`, 2))), color='white', fontface='bold') +
  labs(title='Sales per Customer Segment and Sub-Category') +
  theme_classic() +
  theme(plot.title=element_text(size=13, face='bold', hjust=0.5)) +
  scale_fill_gradient2(low = "#7bae9f", mid = "#6b76ad", high = "#8ce591")
```

```
## ‘summarise()‘ has grouped output by ’Segment’. You can override using the
## ‘.groups‘ argument.
## Scale for fill is already present. Adding another scale for fill, which will
## replace the existing scale.
```

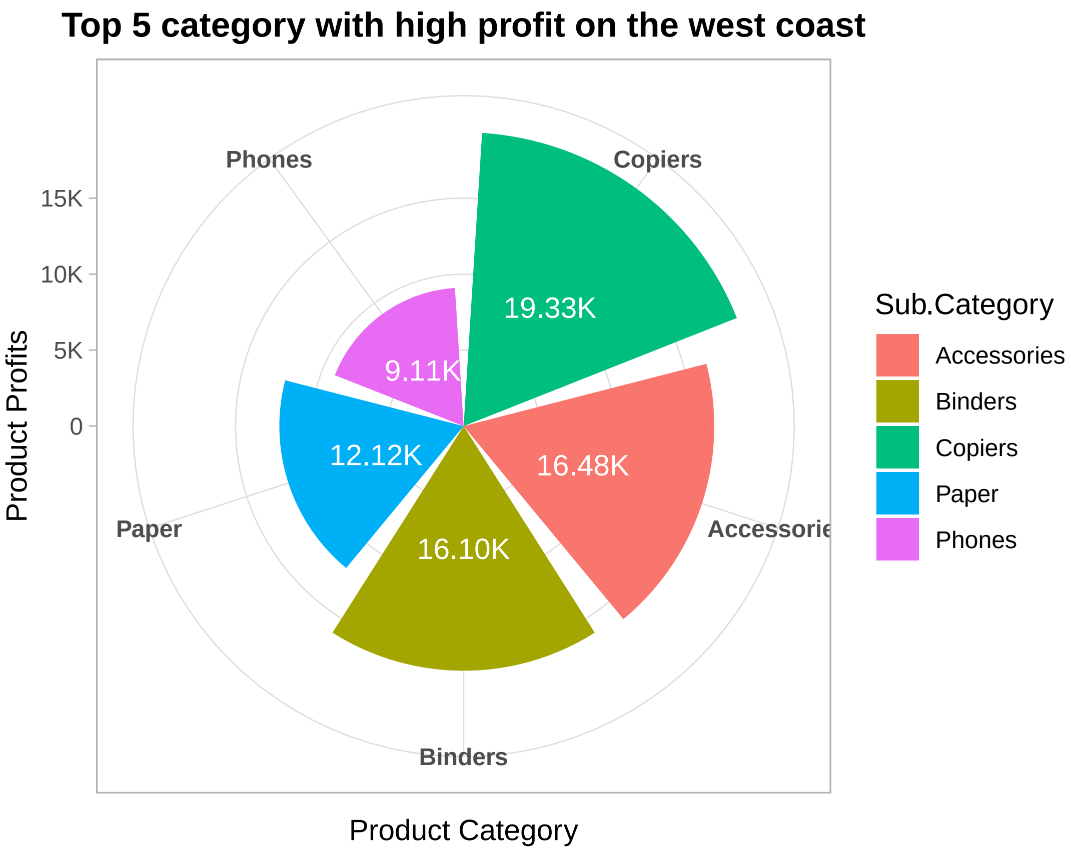
```
plot5
```



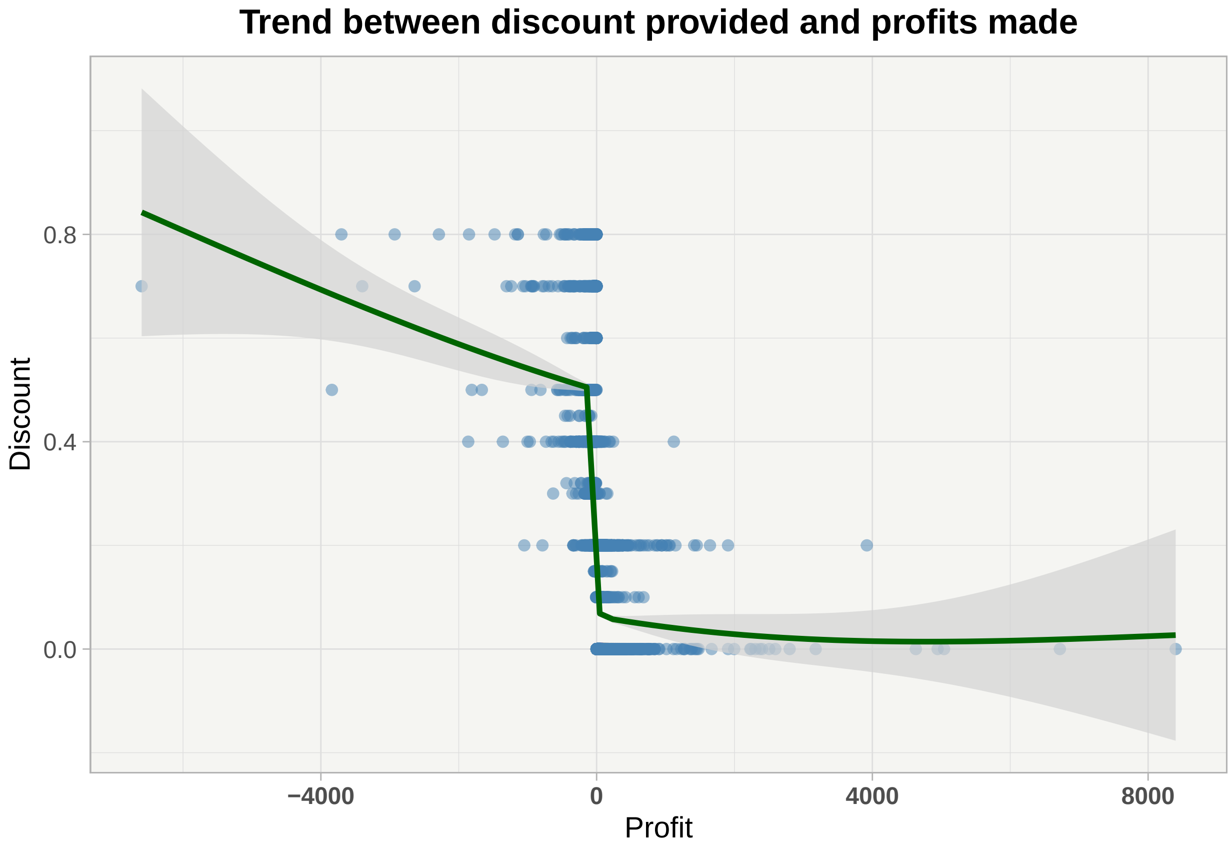
```
#group by sub category to find the profits made by stores on west
West_profit <- Store_west %>%group_by(`Sub.Category`) %>%
    summarize(product_profit=sum(Profit)) %>%
    arrange(-`product_profit`)

PlotData <- slice(West_profit, 1:5)[c('Sub.Category', 'product_profit')] %>%mutate(position=product_pro

PlotData$p_profit_formatted <- label_number_si()(PlotData$product_profit)
PlotData
```

```
## # A tibble: 5 x 4
##   Sub.Category product_profit position p_profit_formatted
##   <chr>                 <dbl>    <dbl> <chr>
## 1 Copiers              19327.    9664. 19.33K
## 2 Accessories          16485.    8242. 16.48K
## 3 Binders              16097.    8048. 16.10K
## 4 Paper                12119.    6060. 12.12K
## 5 Phones                9111.    4555. 9.11K
```

```
plot6<-PlotData %>%
    ggplot(aes(x=reorder(Sub.Category, -product_profit), y=product_profit, fill=Sub.Category, order_by=p
    geom_bar(stat='identity')+
    scale_y_continuous(labels = label_number_si()) +
    geom_text(aes(y=position, label=p_profit_formatted), color='white')+
    coord_polar()+
    labs(title='Top 5 category with high profit on the west coast',x='Product Category', y='Product Pro
    theme_light()+
    theme(plot.title=element_text(size=13, face='bold', hjust=0.5),
          axis.text.x=element_text(face='bold'))

plot6
```



```
## ‘geom_smooth()‘ using method = ’gam’ and formula = ’y ~ s(x, bs = "cs")’
```


Trend between discount provided and profits made
Discount
0.8
0.4
0.0
Profit
−4000
0
4000
8000