HW5_sela_amir

Amir Sela 2024-11-20

```
library(dplyr)
library(ggplot2)
library(cluster)
library(factoextra)
library(tidyclust)
library(tidymodels)
library(GGally)
library(plotly)
library(DT)
library(tidyverse)
library(rpart.plot)
library(lubridate)
library(ggridges)
```

```
retail_data_1 <- read_csv("online_retail_1.csv")
retail_data_2 <- read_csv("online_retail_2.csv")
```

```
retail_data_all <- rbind(retail_data_1,retail_data_2)

head(retail_data_all)
```

```
## # A tibble: 6 × 8
##   Invoice StockCode Description    Quantity InvoiceDate Price CustomerID Countr
##   <chr>   <chr>     <chr>             <dbl> <chr>       <dbl>      <dbl> <chr>
## 1 489434  85048     "15CM CHRISTM…       12 12/1/2009 …  6.95      13085 United
## 2 489434  79323P    "PINK CHERRY …       12 12/1/2009 …  6.75      13085 United
## 3 489434  79323W    "WHITE CHERRY…       12 12/1/2009 …  6.75      13085 United
## 4 489434  22041     "RECORD FRAME…       48 12/1/2009 …  2.1       13085 United
## 5 489434  21232     "STRAWBERRY C…       24 12/1/2009 …  1.25      13085 United
## 6 489434  22064     "PINK DOUGHNU…       24 12/1/2009 …  1.65      13085 United
```

###From the homework i will answer questions 3,10 and 11

Question 3:Which customer segments exist within our customer base?

To answer this question, i will approach it in two ways, first by grouping the data by certain conditions and visualizing it, and then with k means clustering

```
# first lets take a look at the data
summary(retail_data_all)
```

```
##    Invoice            StockCode          Description          Quantity
##  Length:1050922     Length:1050922     Length:1050922     Min.   :-9600.00
##  Class :character   Class :character   Class :character   1st Qu.:    1.00
##  Mode  :character   Mode  :character   Mode  :character   Median :    3.00
##                                                            Mean   :   10.34
##                                                            3rd Qu.:   10.00
##                                                            Max.   :19152.00
##
##  InvoiceDate            Price              CustomerID        Country
##  Length:1050922     Min.   :-53594.360   Min.   :12346    Length:1050922
##  Class :character   1st Qu.:     1.250   1st Qu.:13983    Class :character
##  Mode  :character   Median :     2.100   Median :15311    Mode  :character
##                     Mean   :     4.689   Mean   :15361
##                     3rd Qu.:     4.210   3rd Qu.:16799
##                     Max.   : 25111.090   Max.   :18287
##                                          NA's   :215854
```

```
# we can see that when it comes to numerical data there are some extreme data valu
# now lets check if there are any missing values
sapply(retail_data_all, function(x) sum(is.na(x))) # this gives us the sum of miss
```

```
##     Invoice   StockCode Description    Quantity InvoiceDate       Price
##           0           0        5856           0           0           0
##  CustomerID     Country
##      215854           0
```

```
retail_data_all_cleaned <- na.omit(retail_data_all) # dataframe without NA values
```

We have a lot of customerID(20.5394882 %) values missing, assuming that the missing data is random, it shouldnt affect the visualization a lot We have 4384 distinct customers

#Grouping We will be grouping by costumer ID

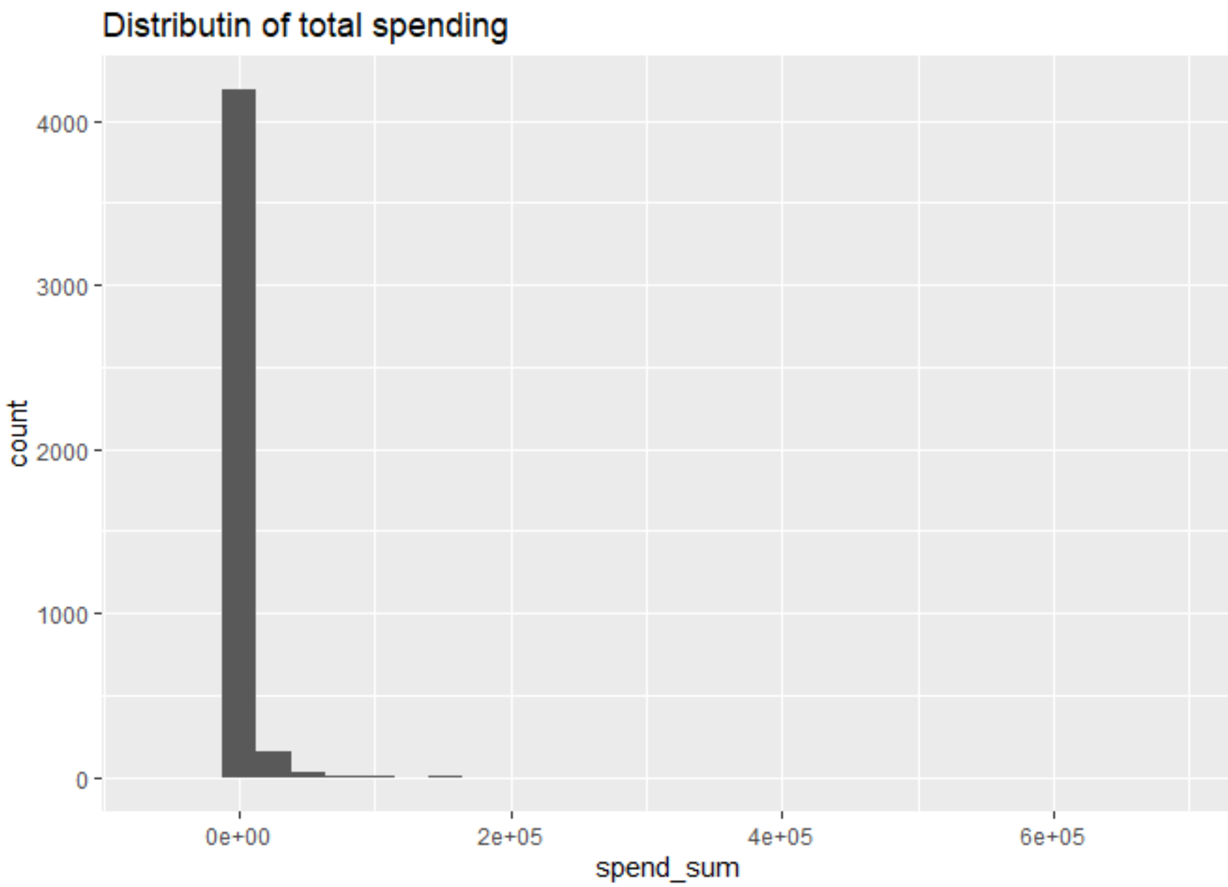
```
 last_date <- max(as.Date(retail_data_1$InvoiceDate, format = "%m/%d/%Y %H:%M"))#
retail_summary <- retail_data_all_cleaned %>%
  group_by(CustomerID) %>%  #grouping
  summarise(
    spend_sum = sum(Price * Quantity), # summary of how much they spend
    avg_quantity = mean(Quantity), # avg quantity
    sum_transactions = n(), # num of transaction
    recency = as.numeric(difftime(last_date,max(as.Date(InvoiceDate, format = "%m/
                              units = "days")) # how long from last date has i
    )

head(retail_summary)
```

```
## # A tibble: 6 × 5
##   CustomerID spend_sum avg_quantity sum_transactions recency
##        <dbl>     <dbl>        <dbl>            <int>   <dbl>
## 1      12346     -129.         1.13               92      66
## 2      12347     2647.        11.7               142       2
## 3      12348      444.        18.6                40      73
## 4      12349     5294.         9.23              214      42
## 5      12351      602.        12.4                42      10
## 6      12352      688.        10.4                36      10
```

Now that we have some data we can work with, lets visualize them and see what segments of costumers exist within our dataset

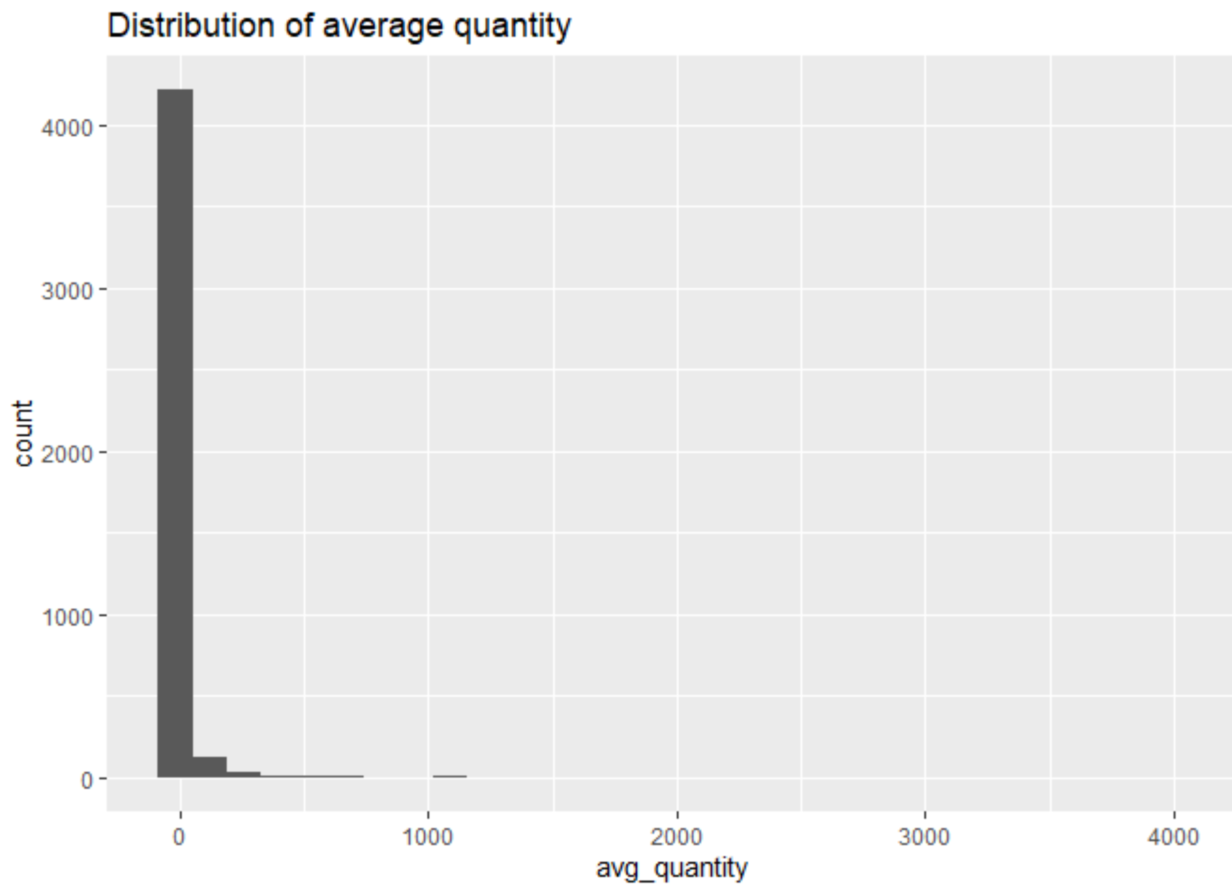
```
ggplot(retail_summary, aes(x = spend_sum)) +
  geom_histogram() +
  labs(
    title = "Distributin of total spending"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```




```
ggplot(retail_summary, aes(x = avg_quantity)) +
  geom_histogram()+
  labs(
    title = "Distribution of average quantity"
  )
```

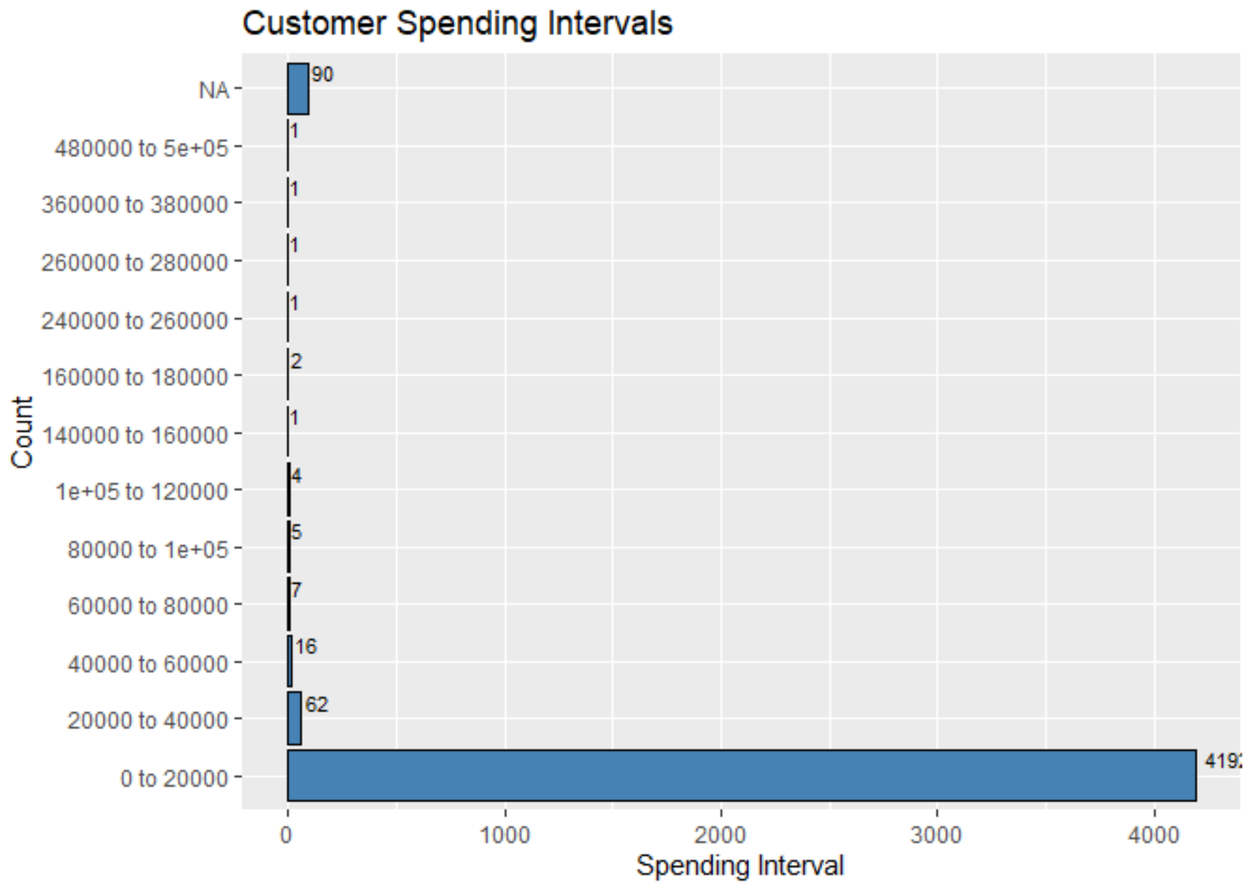
```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



by the visualization we can only see few bars, but we know that there are other bars in the histogram because it is stretched in the y axis, but because they are so small we can not see them, so lets create a bar visualization

```
retail_summary <- retail_summary %>%
  mutate(
    spending_interval = cut(
      spend_sum,
      breaks = seq(0, max(spend_sum, na.rm = TRUE), by = 20000),
      labels = paste0(seq(0, max(spend_sum, na.rm = TRUE) - 20000, by = 20000),
                      " to ",
                      seq(20000, max(spend_sum, na.rm = TRUE), by = 20000)),
      include.lowest = TRUE
    )# this add labels to each costumer by which interval the fall into, the inter
  )

ggplot(retail_summary, aes(y = spending_interval)) +
  geom_bar(fill = "steelblue", color = "black") +
  geom_text(stat = "count", aes(label = after_stat(count)), vjust = -0.5,hjust = -
  labs(
    title = "Customer Spending Intervals",
    x = "Spending Interval",
    y = "Count"
```

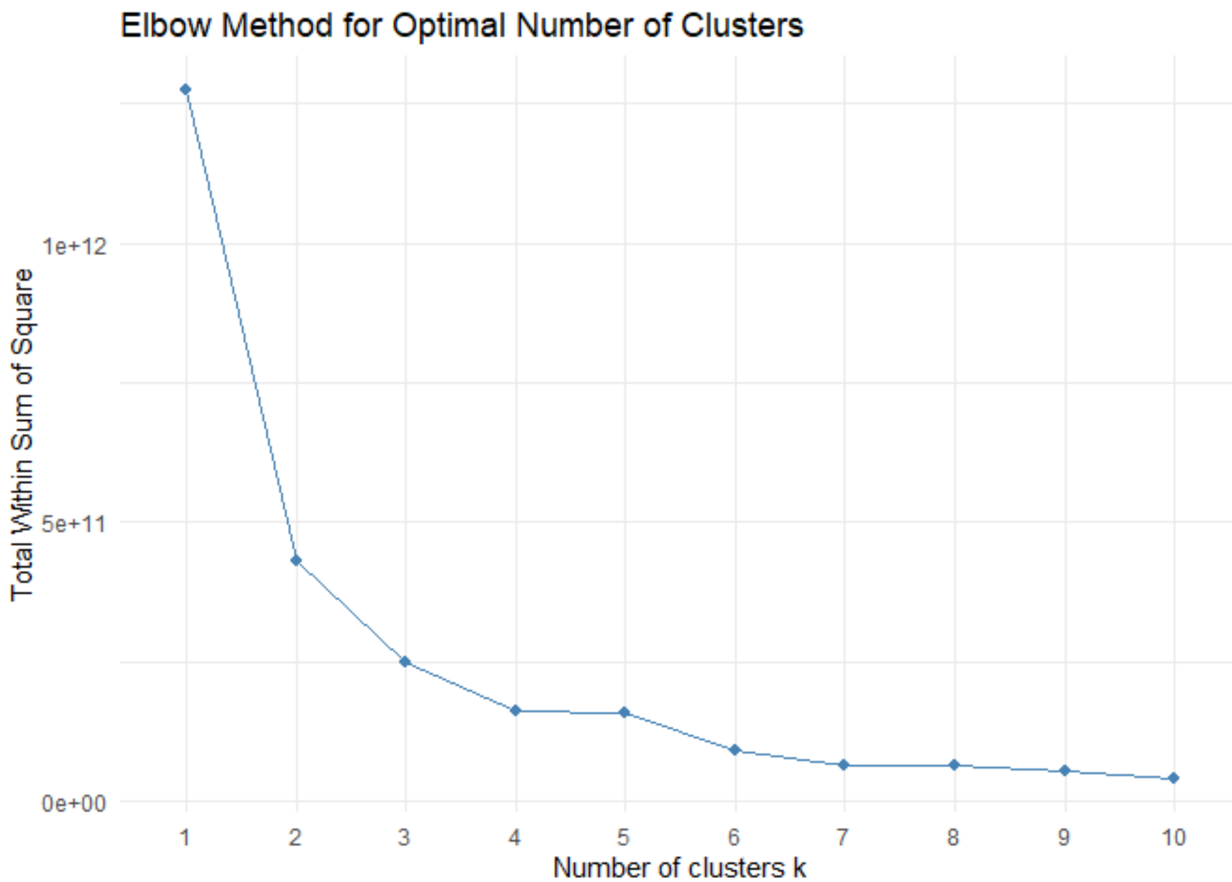
```
)
```



This is one way we can create costumer segments, by putting the in intervals.

Now lets use K-means clustering to create costumer segments

```
retail_data_all_scaled <- retail_summary %>%
  select(spend_sum,avg_quantity,sum_transactions,recency) %>% # selecting and scal
  as.data.frame(scale())


fviz_nbclust(retail_data_all_scaled, kmeans, method = "wss") +
  labs(title = "Elbow Method for Optimal Number of Clusters") +
  theme_minimal()
```



The optimal amount of clusters to use would be 5

```
set.seed(123)

# Define and fit the model
kmeans_spec <- k_means(num_clusters = 5) %>% set_engine("stats")
kmeans_fit_2d <- kmeans_spec %>% fit(~ spend_sum + recency, data = retail_data_all

# Add clusters to the original data
retail_data_all_scaled$cluster_2d <- as.factor(predict(kmeans_fit_2d, new_data = r

# Visualize the clusters
ggplot(retail_data_all_scaled, aes(x = spend_sum, y = recency, color = cluster_2d)
  geom_point(size = 1) +
  labs(title = "K-means Clustering Results on Retail data", x = "Total Spending",
  theme_minimal()
```

As we can see from the cluster analysis, we can divide the costumers into 5 groups, where the group with green labelling are the ones who have returned products, orange group show us that if the spending total is low, the costumer could be an repeating costumer or a returning costumer at the same time. as the spedning total increaset we can see that the recency is low, meaning that the highger the spending total, we could predict that the costumer is a repeat costumer who buys in bulk.

Question 11

Are there any distinct paterns in costumer spending behaviour

```
patterns_summary <- retail_data_all_cleaned %>%
  mutate(InvoiceDate = sub(" .*", "", InvoiceDate)) %>%
  group_by(InvoiceDate) %>% # groups it by ID and date, if there are more than one
  summarise(
    spent_sum = sum(Price * Quantity, na.rm = TRUE), # total purchased
  )
patterns_summary <- patterns_summary %>%
  rename(date = 1) # renames the first column with name "date"
```

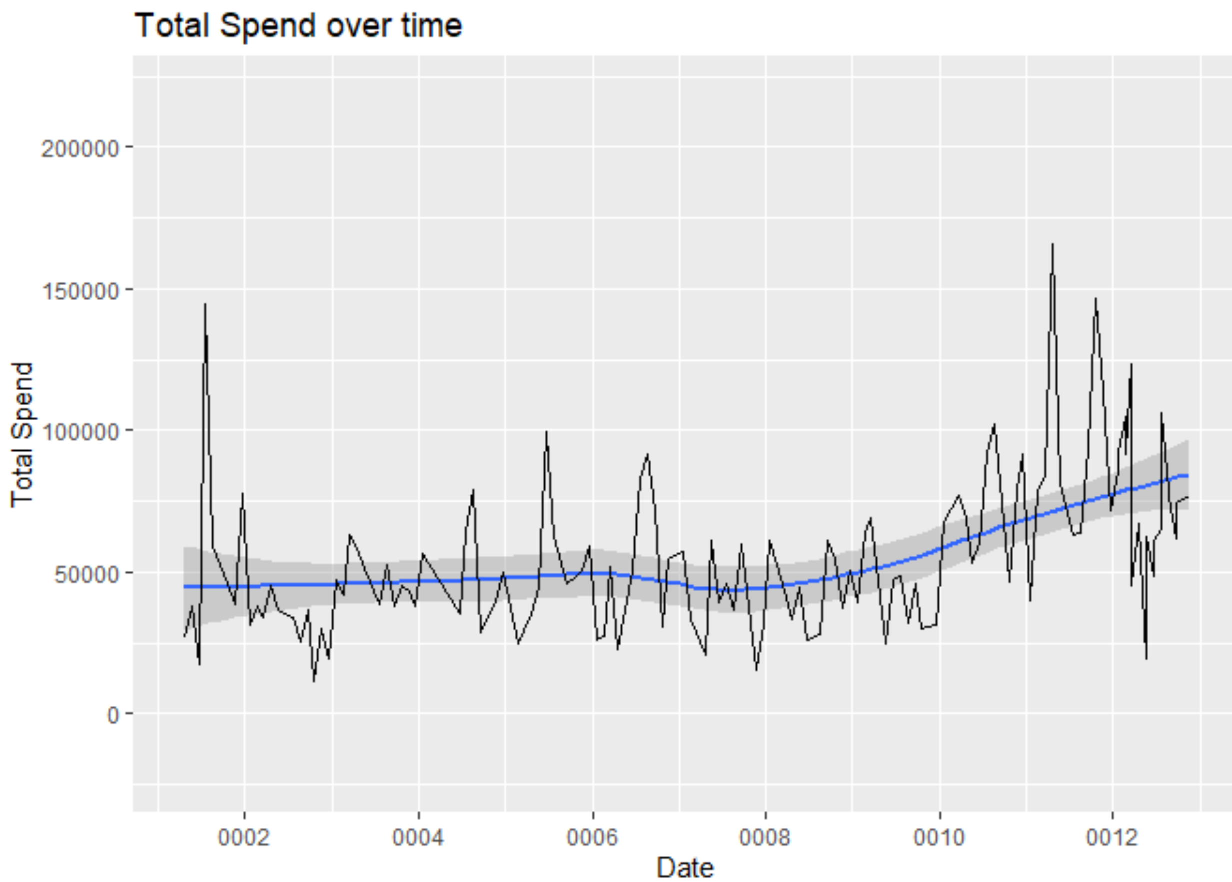
```r
head(patterns_summary)
```

```
## # A tibble: 6 × 2
##   date      spent_sum
##   <chr>         <dbl>
## 1 1/10/2010    46159.
## 2 1/11/2010    38586.
## 3 1/12/2010    77475.
## 4 1/13/2010    18642.
## 5 1/14/2010    46598.
## 6 1/15/2010    26580.
```

Now that we have data we can work with lets visualize and see if we can find any patterns.

```r
ggplot(patterns_summary, aes(x = as.Date(date), y = spent_sum)) + # over time plot
  geom_smooth() +
  geom_line() +
  labs(
    title = "Total Spend over time",
    x = "Date",
    y = "Total Spend"
  )
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'

## Warning: Removed 180 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 180 rows containing missing values or values outside the scale
## (`geom_line()`).
```



Judging by the scatterplot, we can say that the spending pattern stays stable during the year, but increases at the end of the year, this could be due to holidays and costumers are buying gifts

Question 10 ### Can we predict if a costumer will buy a specific product category

To predict if a costumer will buy a certain category, we will use a decision tree

```
retail_data_tree <- retail_data_all_cleaned %>%
  select(Price, Quantity, InvoiceDate) %>%  # select only needed columns
  mutate(price_range = cut(Price,
                           breaks = quantile(Price, probs = seq(0, 1, 0.2), na.rm
                           labels = c("Very Low", "Low", "Medium", "High", "Very H
                           include.lowest = TRUE))
```

Splitting Data

```
#|label: Spliting data

set.seed(123)
```

```
retail_split <- initial_split(retail_data_tree, prop = 0.8)  # using 80/20 split
retail_train <- training(retail_split)
retail_test <- testing(retail_split)
```

Defining and training decision tree model

```
retail_tree_model <- decision_tree( # defining tree model
  mode = "classification",
  tree_depth = 3
) %>%
  set_engine("rpart")

retail_recipe <- recipe(price_range ~ Quantity, data = retail_train) # recipe
retail_workflow <- workflow() %>%
  add_model(retail_tree_model) %>%
  add_recipe(retail_recipe)

retail_fit <- retail_workflow %>%  # adding the workflow for the trained data
  fit(data = retail_train)
```

Evaluating the model

```
retail_predictions <- retail_fit %>%
  predict(retail_test) %>%  # making predictions
  bind_cols(retail_test)

retail_evaluation_metric <- retail_predictions %>%  # see how accurate our predict
  metrics(truth = price_range, estimate = .pred_class)

retail_confusion_matrix <- retail_predictions %>%
  conf_mat(truth = price_range, estimate = .pred_class)

print(retail_evaluation_metric)
```

```
## # A tibble: 2 × 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy multiclass     0.366
## 2 kap      multiclass     0.201
```

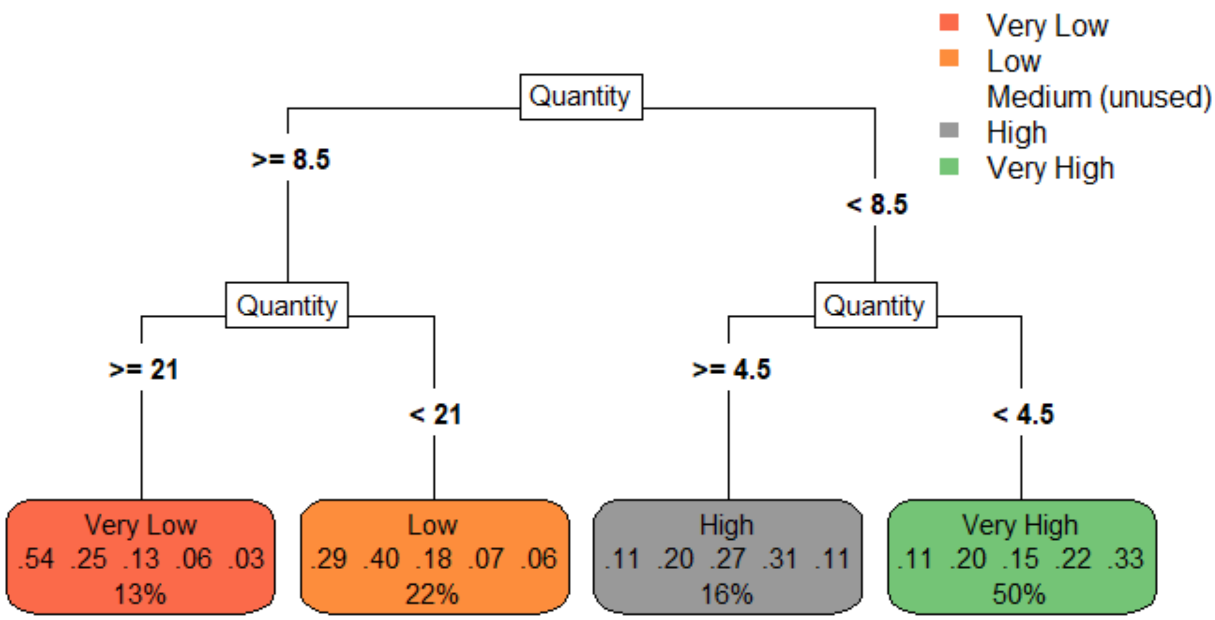
```r
print(retail_confusion_matrix)
```

```
##             Truth
## Prediction  Very Low   Low Medium  High Very High
##   Very Low     11558  5393   2697  1267       602
##   Low          10267 14408   6579  2589      2021
##   Medium           0     0      0     0         0
##   High          2902  5133   7086  7925      2784
##   Very High     8850 17197  12247 18295     27214
```

Visualizing the Decision Tree

```r
retail_tree_viz <- retail_fit %>% # visualizing
  extract_fit_engine()

rpart.plot(retail_tree_viz, type = 5, extra = 104)
```

```
## Warning: Cannot retrieve the data used to build the model (so cannot determine
## To silence this warning:
##     Call rpart.plot with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```



The confusion and the evaluation matrix shows us that we can only predict 1/3 of the data with the decision model tree, which is not a good model. In my opinion its hard to create a very accurate decision tree with this data because its under fitting, it only has little numerical columns which we can use to predict a categorical variable, in this case price_range