Customer Personality Analysis

Introduction

We are a team focus on the detailed analysis of a company's ideal customers. Our team strives to help company and business to modify products for customers with special needs. This dataset includes 27 features.

Variable Description

We can roughly categorized variable as 4 higher level group to describe the data:

People(Age, marital status, kids number...)

Product(Amount of spend on each product category last 2 year...)

Promotion(number of deals using promotion)

Places(where did customer purchases, online or in store)

Project Goal

Our goal is to perform clustering to summarize customer segments. We are achieving the goal through three steps: data cleaning, exploratory data analysis(EDA), and segmentation.

Data Cleaning

```
library(tidyverse)
library(dplyr)
library(naniar)
library(gridExtra)
library(lubridate)
library(FactoMineR)
library(factoextra)
library(ggfortify)
library(ggplot2)
```

```
df <- read.delim("marketing_campaign.csv", stringsAsFactors = FALSE)
head(df)
```

```
##     ID Year_Birth  Education Marital_Status Income Kidhome Teenhome Dt_Customer
## 1 5524       1957 Graduation         Single  58138       0        0  04-09-2012
## 2 2174       1954 Graduation         Single  46344       1        1  08-03-2014
## 3 4141       1965 Graduation       Together  71613       0        0  21-08-2013
## 4 6182       1984 Graduation       Together  26646       1        0  10-02-2014
```

```
## 5 5324       1981         PhD        Married  58293       1        0  19-01-2014
## 6 7446       1967      Master       Together  62513       0        1  09-09-2013
##   Recency MntWines MntFruits MntMeatProducts MntFishProducts MntSweetProducts
## 1      58      635        88             546             172               88
## 2      38       11         1               6               2                1
## 3      26      426        49             127             111               21
## 4      26       11         4              20              10                3
## 5      94      173        43             118              46               27
## 6      16      520        42              98               0               42
##   MntGoldProds NumDealsPurchases NumWebPurchases NumCatalogPurchases
## 1           88                 3               8                 10
## 2            6                 2               1                  1
## 3           42                 1               8                  2
## 4            5                 2               2                  0
## 5           15                 5               5                  3
## 6           14                 2               6                  4
##   NumStorePurchases NumWebVisitsMonth AcceptedCmp3 AcceptedCmp4 AcceptedCmp5
## 1                 4                 7            0            0            0
## 2                 2                 5            0            0            0
## 3                10                 4            0            0            0
## 4                 4                 6            0            0            0
## 5                 6                 5            0            0            0
## 6                10                 6            0            0            0
##   AcceptedCmp1 AcceptedCmp2 Complain Z_CostContact Z_Revenue Response
## 1            0            0        0             3        11        1
## 2            0            0        0             3        11        0
## 3            0            0        0             3        11        0
## 4            0            0        0             3        11        0
## 5            0            0        0             3        11        0
## 6            0            0        0             3        11        0
```

```
dim(df)
```

```
## [1] 2240   29
```

```
summary(df)
```

```
##        ID          Year_Birth    Education         Marital_Status
##  Min.   :    0   Min.   :1893   Length:2240        Length:2240
##  1st Qu.: 2828   1st Qu.:1959   Class :character   Class :character
##  Median : 5458   Median :1970   Mode  :character   Mode  :character
##  Mean   : 5592   Mean   :1969
##  3rd Qu.: 8428   3rd Qu.:1977
##  Max.   :11191   Max.   :1996
##
##      Income          Kidhome          Teenhome      Dt_Customer
##  Min.   :  1730   Min.   :0.0000   Min.   :0.0000   Length:2240
##  1st Qu.: 35303   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median : 51382   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   : 52247   Mean   :0.4442   Mean   :0.5062
##  3rd Qu.: 68522   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :666666   Max.   :2.0000   Max.   :2.0000
##  NA's   :24
```

```
##     Recency         MntWines         MntFruits     MntMeatProducts
##  Min.   : 0.00   Min.   :   0.00   Min.   :  0.0   Min.   :   0.0
##  1st Qu.:24.00   1st Qu.:  23.75   1st Qu.:  1.0   1st Qu.:  16.0
##  Median :49.00   Median : 173.50   Median :  8.0   Median :  67.0
##  Mean   :49.11   Mean   : 303.94   Mean   : 26.3   Mean   : 166.9
##  3rd Qu.:74.00   3rd Qu.: 504.25   3rd Qu.: 33.0   3rd Qu.: 232.0
##  Max.   :99.00   Max.   :1493.00   Max.   :199.0   Max.   :1725.0
##
##  MntFishProducts  MntSweetProducts  MntGoldProds    NumDealsPurchases
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:  3.00   1st Qu.:  1.00   1st Qu.:  9.00   1st Qu.: 1.000
##  Median : 12.00   Median :  8.00   Median : 24.00   Median : 2.000
##  Mean   : 37.53   Mean   : 27.06   Mean   : 44.02   Mean   : 2.325
##  3rd Qu.: 50.00   3rd Qu.: 33.00   3rd Qu.: 56.00   3rd Qu.: 3.000
##  Max.   :259.00   Max.   :263.00   Max.   :362.00   Max.   :15.000
##
##  NumWebPurchases  NumCatalogPurchases NumStorePurchases NumWebVisitsMonth
##  Min.   : 0.000   Min.   : 0.000      Min.   : 0.00     Min.   : 0.000
##  1st Qu.: 2.000   1st Qu.: 0.000      1st Qu.: 3.00     1st Qu.: 3.000
##  Median : 4.000   Median : 2.000      Median : 5.00     Median : 6.000
##  Mean   : 4.085   Mean   : 2.662      Mean   : 5.79     Mean   : 5.317
##  3rd Qu.: 6.000   3rd Qu.: 4.000      3rd Qu.: 8.00     3rd Qu.: 7.000
##  Max.   :27.000   Max.   :28.000      Max.   :13.00     Max.   :20.000
##
##   AcceptedCmp3      AcceptedCmp4      AcceptedCmp5      AcceptedCmp1
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.07277   Mean   :0.07455   Mean   :0.07277   Mean   :0.06429
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##
##   AcceptedCmp2        Complain        Z_CostContact   Z_Revenue
##  Min.   :0.00000   Min.   :0.000000   Min.   :3     Min.   :11
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:3     1st Qu.:11
##  Median :0.00000   Median :0.000000   Median :3     Median :11
##  Mean   :0.01339   Mean   :0.009375   Mean   :3     Mean   :11
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:3     3rd Qu.:11
##  Max.   :1.00000   Max.   :1.000000   Max.   :3     Max.   :11
##
##     Response
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1491
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
```

Missing Values

```r
# Counting the total number of missing values and each variables in the data
n_miss(df)
```

```
## [1] 24
```

```r
miss_var_summary(df)
```

```
## # A tibble: 29 x 3
##    variable       n_miss pct_miss
##    <chr>           <int>    <dbl>
##  1 Income             24     1.07
##  2 ID                  0     0
##  3 Year_Birth          0     0
##  4 Education           0     0
##  5 Marital_Status      0     0
##  6 Kidhome             0     0
##  7 Teenhome            0     0
##  8 Dt_Customer         0     0
##  9 Recency             0     0
## 10 MntWines            0     0
## # ... with 19 more rows
```

Notice that income is the only variable with missing data problem, we will just drop the customer data point with missing data.

```r
# Drop NA values
df_customers <- na.omit(df)
dim(df_customers)
```

```
## [1] 2216   29
```

Outliers

The variable Year_Birth gives us the birth year of customers, which is not very intuitive. So we create a new variable Age.

```r
# Creating a new variable Age from Year of Birth
customers_unfilter <- df_customers %>%
  mutate(Age = 2022 - Year_Birth)
customers_unfilter %>%
  select(Age) %>%
  arrange(desc(Age)) %>%
  top_n(10)
```

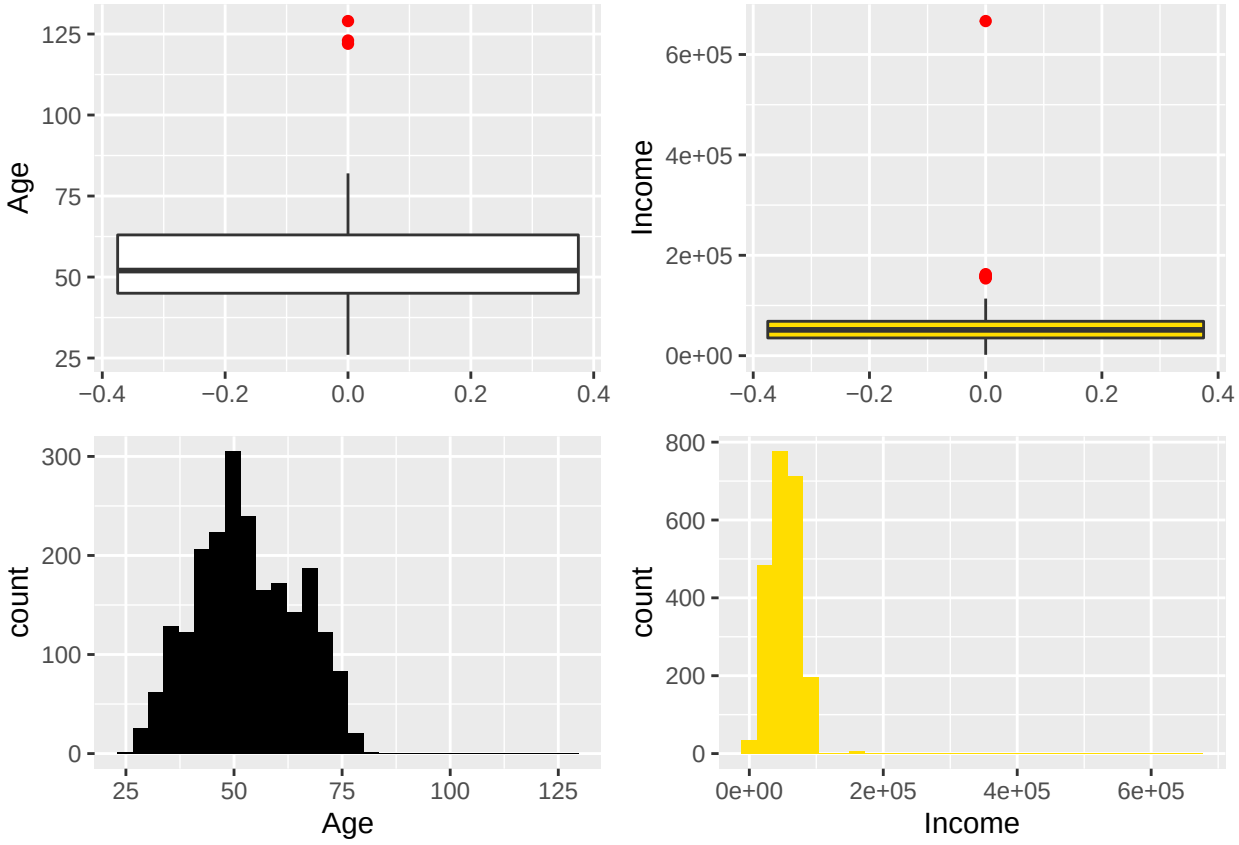
```
## Selecting by Age
```

```
##    Age
## 1  129
## 2  123
```

```
## 3  122
## 4   82
## 5   81
## 6   79
## 7   79
## 8   79
## 9   79
## 10  79
## 11  79
```

```
# identify the outliers
g1=ggplot(customers_unfilter, aes(y = Age)) + geom_boxplot(outlier.colour = 'red')
g2=ggplot(customers_unfilter, aes(y = Income)) + geom_boxplot(outlier.colour = 'red',fill='#FFDD00')
g3=ggplot(customers_unfilter, aes(x = Age)) + geom_histogram(fill='black')
g4=ggplot(customers_unfilter, aes(x = Income)) + geom_histogram(fill='#FFDD00')
grid.arrange(g1, g2, g3, g4, ncol=2)
```

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




```
# Max Age is > 100
# Dropping outliers by setting a cap on Income and Age
customers <- customers_unfilter %>%
  filter(Income < 600000 & Age < 90)
dim(customers)
```

```
## [1] 2212   30
```

```
unique(customers$Marital_Status)
```

```
## [1] "Single"   "Together" "Married"  "Divorced" "Widow"    "Alone"    "Absurd"
## [8] "YOLO"
```

Collapsing Features

Some features have too many unnecessary categories, we can convert to only two categories.

```
# Collapsing marital Status into two categories: Single & Taken

customers <- customers %>%
  mutate(Marital_Status =
           replace(Marital_Status,
                   Marital_Status == "Divorced" | Marital_Status == "Widow" |
                     Marital_Status == "Alone" | Marital_Status == "Absurd" |
                     Marital_Status == "YOLO",
                   "Single"))
customers <- customers %>%
  mutate(Marital_Status =
           replace(Marital_Status,
                   Marital_Status == "Together" | Marital_Status == "Married",
                   "Taken"))
```

```
# Collapsing the Education into two Categories: graduate and non-graduate
unique(customers$Education)
```

```
## [1] "Graduation" "PhD"        "Master"     "Basic"      "2n Cycle"
```

```
customers <- customers %>%
  mutate(Education =
           replace(Education, Education == "Graduation"| Education == "PhD" |
                     Education == "Master" | Education == "2n Cycle",
                   "graduate"))
customers <- customers %>%
  mutate(Education =
           replace(Education, Education == "Basic", "non-graduate"))
```

```
# Converting them to factors
customers <- customers %>%
  mutate(Marital_Status = as.factor(Marital_Status),
         Education = as.factor(Education))
```

Renaming Features

```r
# Renaming the Mnt_ features
customers <- customers %>%
  rename(wines = MntWines, fruits = MntFruits, meat = MntMeatProducts,
         fish = MntFishProducts, sweet = MntSweetProducts, gold = MntGoldProds,
         webpurchase = NumWebPurchases, catalog = NumCatalogPurchases,
         Store = NumStorePurchases, webvisit = NumWebVisitsMonth,
         dealpurchase = NumDealsPurchases)
```

```r
# Creating a new variable:Total_spent
customers <- customers %>%
  mutate(Total_spent = wines + fruits + meat + fish + sweet + gold) %>%
  mutate(Total_num = webpurchase + catalog + Store + dealpurchase)
```

```r
# Creating a new variable:kids
customers <- customers %>%
  mutate(Kids = Kidhome + Teenhome)
```

```r
#Dropping some redundant features
customers <- customers %>%
  select(- ID, - Year_Birth, - Dt_Customer, - Z_CostContact,
         - Z_Revenue, - Kidhome, - Teenhome)
str(customers)
```

```
## 'data.frame':    2212 obs. of  26 variables:
##  $ Education     : Factor w/ 2 levels "graduate","non-graduate": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Marital_Status: Factor w/ 2 levels "Single","Taken": 1 1 2 2 2 2 1 2 2 2 ...
##  $ Income        : int  58138 46344 71613 26646 58293 62513 55635 33454 30351 5648 ...
##  $ Recency       : int  58 38 26 26 94 16 34 32 19 68 ...
##  $ wines         : int  635 11 426 11 173 520 235 76 14 28 ...
##  $ fruits        : int  88 1 49 4 43 42 65 10 0 0 ...
##  $ meat          : int  546 6 127 20 118 98 164 56 24 6 ...
##  $ fish          : int  172 2 111 10 46 0 50 3 3 1 ...
##  $ sweet         : int  88 1 21 3 27 42 49 1 3 1 ...
##  $ gold          : int  88 6 42 5 15 14 27 23 2 13 ...
##  $ dealpurchase  : int  3 2 1 2 5 2 4 2 1 1 ...
##  $ webpurchase   : int  8 1 8 2 5 6 7 4 3 1 ...
##  $ catalog       : int  10 1 2 0 3 4 3 0 0 0 ...
##  $ Store         : int  4 2 10 4 6 10 7 4 2 0 ...
##  $ webvisit      : int  7 5 4 6 5 6 6 8 9 20 ...
##  $ AcceptedCmp3  : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ AcceptedCmp4  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp5  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp1  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp2  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Complain      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Response      : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ Age           : num  65 68 57 38 41 55 51 37 48 72 ...
##  $ Total_spent   : int  1617 27 776 53 422 716 590 169 46 49 ...
##  $ Total_num     : int  25 6 21 8 19 22 21 10 6 2 ...
##  $ Kids          : int  0 2 0 1 1 1 1 1 1 2 ...
##  - attr(*, "na.action")= 'omit' Named int [1:24] 11 28 44 49 59 72 91 92 93 129 ...
##   ..- attr(*, "names")= chr [1:24] "11" "28" "44" "49" ...
```

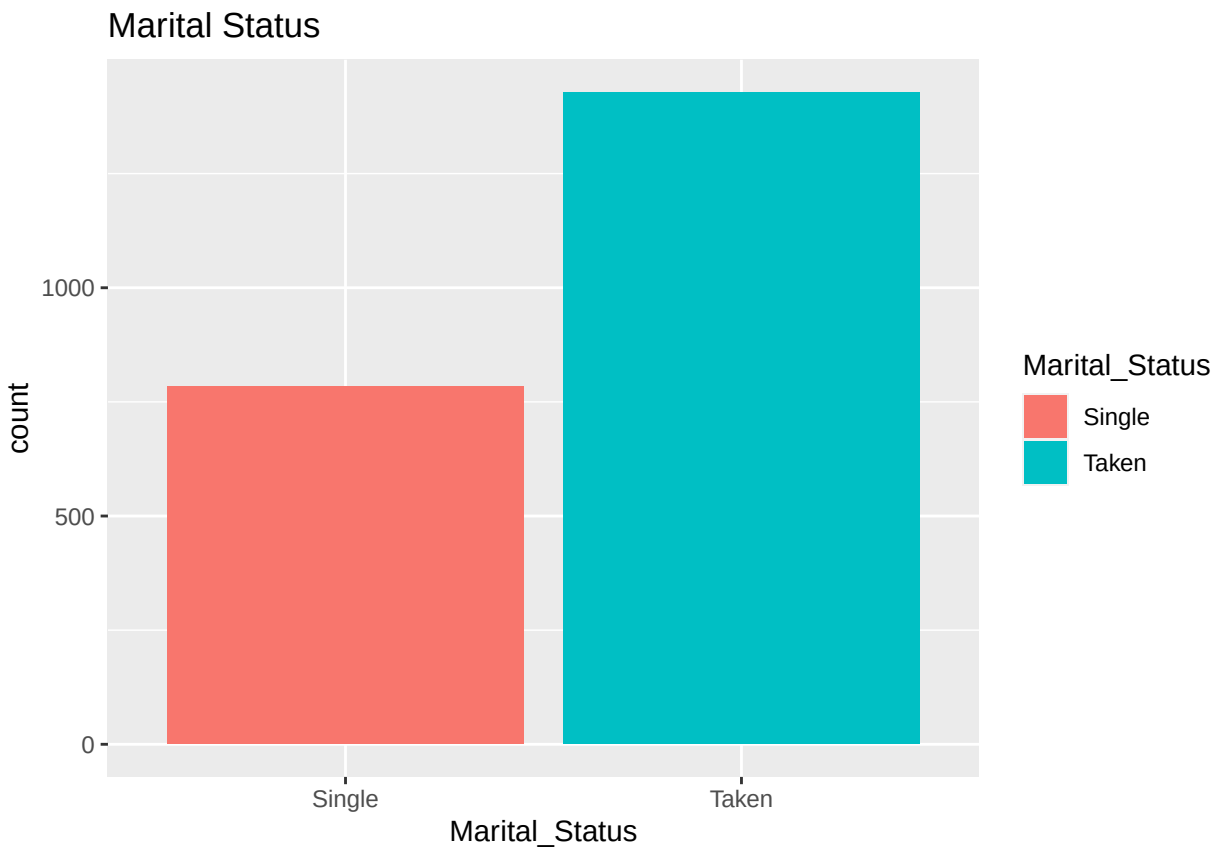
```r
head(customers, n = 5)
```

```
##   Education Marital_Status Income Recency wines fruits meat fish sweet gold
## 1  graduate         Single  58138      58   635     88  546  172    88   88
## 2  graduate         Single  46344      38    11      1    6    2     1    6
## 3  graduate          Taken  71613      26   426     49  127  111    21   42
## 4  graduate          Taken  26646      26    11      4   20   10     3    5
## 5  graduate          Taken  58293      94   173     43  118   46    27   15
##   dealpurchase webpurchase catalog Store webvisit AcceptedCmp3 AcceptedCmp4
## 1            3           8      10     4        7            0            0
## 2            2           1       1     2        5            0            0
## 3            1           8       2    10        4            0            0
## 4            2           2       0     4        6            0            0
## 5            5           5       3     6        5            0            0
##   AcceptedCmp5 AcceptedCmp1 AcceptedCmp2 Complain Response Age Total_spent
## 1            0            0            0        0        1  65        1617
## 2            0            0            0        0        0  68          27
## 3            0            0            0        0        0  57         776
## 4            0            0            0        0        0  38          53
## 5            0            0            0        0        0  41         422
##   Total_num Kids
## 1        25    0
## 2         6    2
## 3        21    0
## 4         8    1
## 5        19    1
```

EDA

1. Marital Status

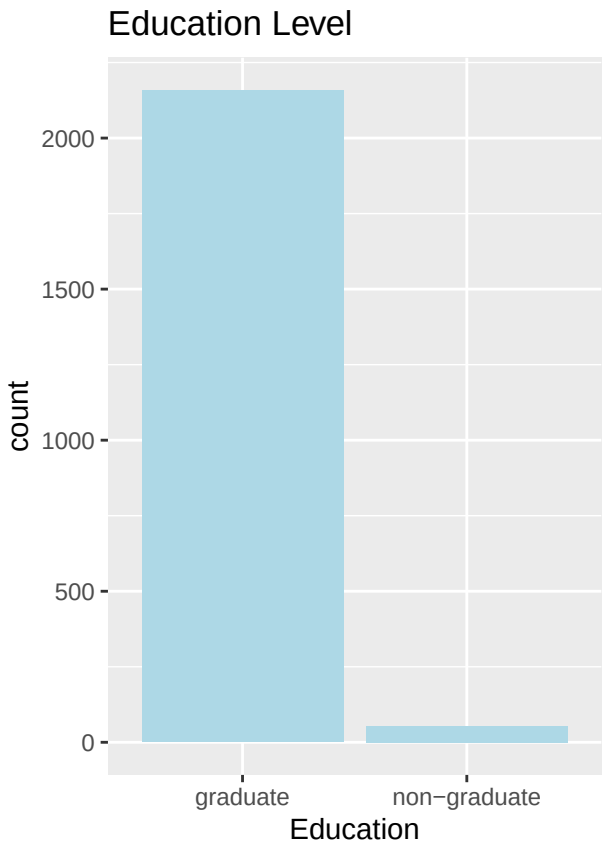
```r
ggplot(customers, aes(x=Marital_Status, fill=Marital_Status)) +
  geom_bar() + ggtitle("Marital Status")
```

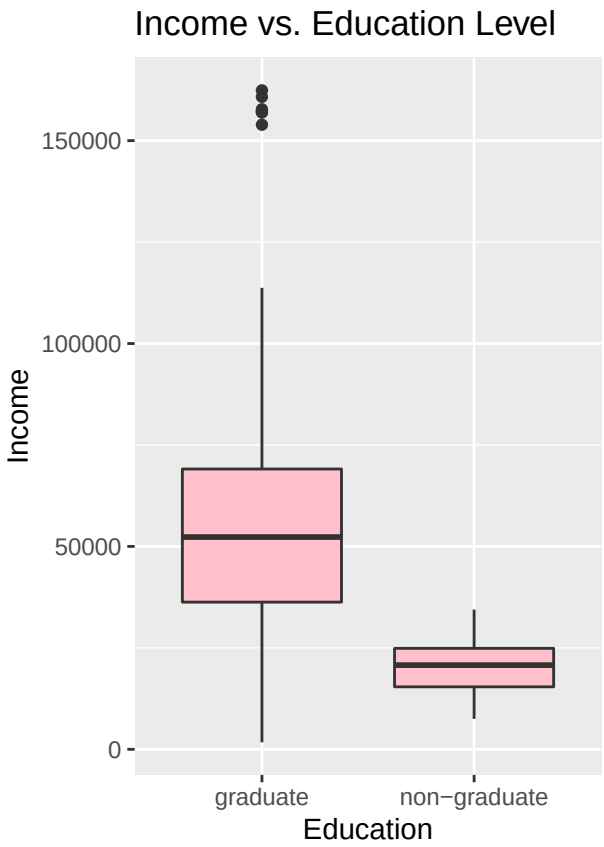



```
g1=ggplot(customers, mapping=aes(x=Education)) +
  geom_bar(fill = "lightblue") + ggtitle("Education Level")
```

2. Education

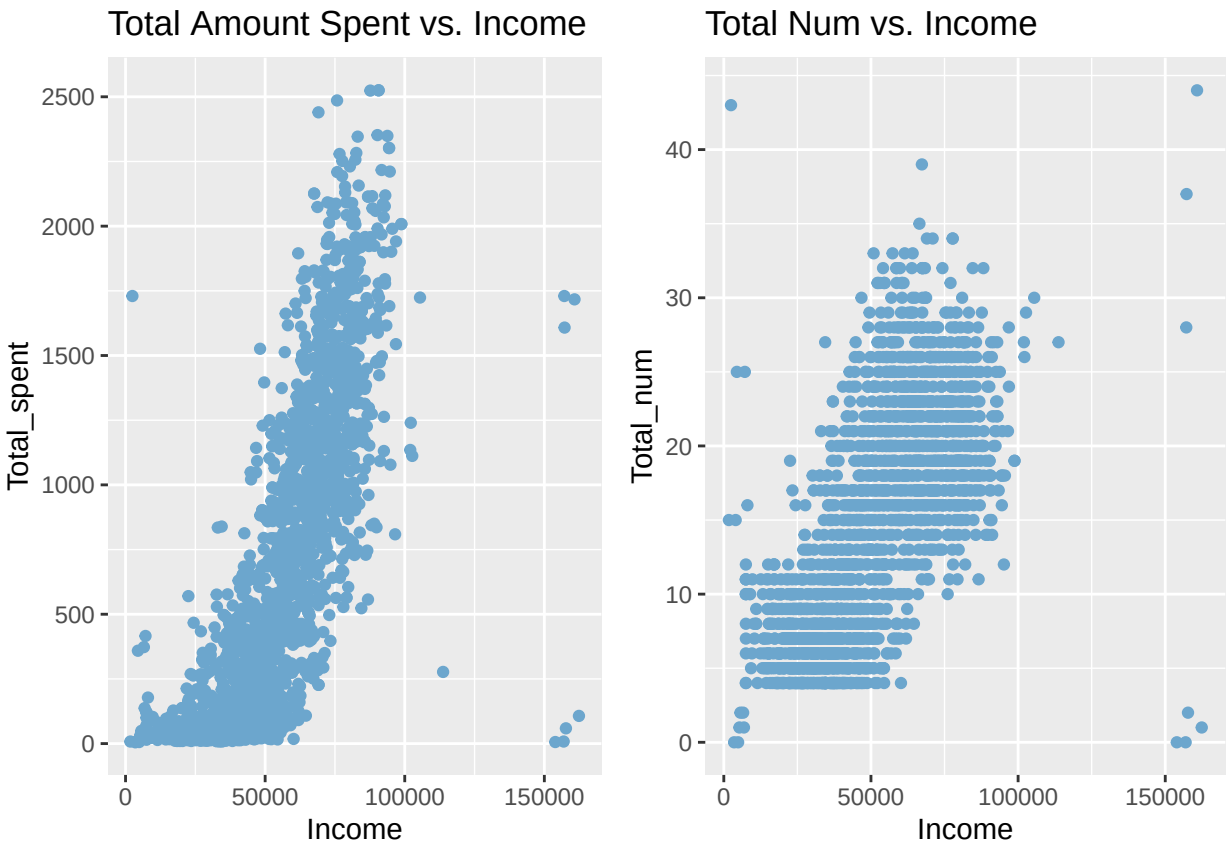
```
g2=ggplot(customers, aes(x=Education, y=Income)) +
  geom_boxplot(fill = "pink") + ggtitle("Income vs. Education Level")
grid.arrange(g1, g2, ncol=2)
```





3. Income

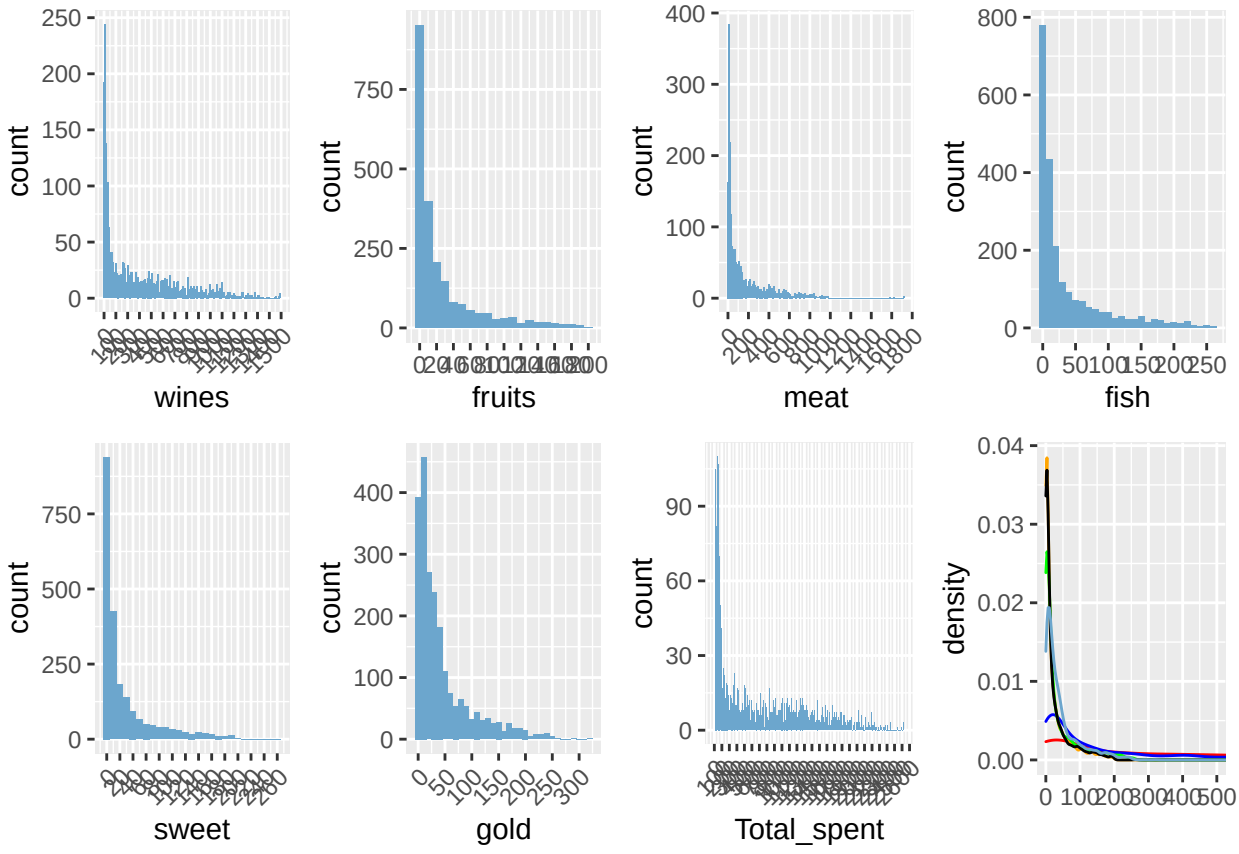
```
g1=ggplot(customers, aes(x=Income, y=Total_spent)) +
  geom_point(col='Sky Blue 3')+
  scale_x_continuous()+
  ggtitle("Total Amount Spent vs. Income")
g2=ggplot(customers, aes(x=Income, y=Total_num)) +
  geom_point(col='Sky Blue 3')+
  scale_x_continuous()+
  ggtitle("Total Num vs. Income")
grid.arrange(g1, g2, ncol=2)
```



4. Products

```
g1 <- ggplot(customers, aes(x = wines)) +
  geom_histogram(fill = "Sky Blue 3", binwidth = 10)+
  scale_x_continuous(breaks = seq(0, 1600, 100)) +
  theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1))
g2 <- ggplot(customers, aes(x = fruits)) +
  geom_histogram(fill = "Sky Blue 3", binwidth = 10)+
  scale_x_continuous(breaks = seq(0, 250, 20))
g3 <- ggplot(customers, aes(x = meat)) +
  geom_histogram(fill = "Sky Blue 3", binwidth = 10)+
  scale_x_continuous(breaks = seq(0, 1800, 200)) +
  theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1))
g4 <- ggplot(customers, aes(x = fish)) +
  geom_histogram(fill = "Sky Blue 3", binwidth = 10)+
  scale_x_continuous(breaks = seq(0, 300, 50))
g5 <- ggplot(customers, aes(x = sweet)) +
  geom_histogram(fill = "Sky Blue 3", binwidth = 10)+
  scale_x_continuous(breaks = seq(0, 300, 20)) +
  theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1))
g6 <- ggplot(customers, aes(x = gold)) +
  geom_histogram(fill = "Sky Blue 3", binwidth = 10)+
  scale_x_continuous(breaks = seq(0, 400, 50)) +
  theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1))
```

```
g7 <- ggplot(customers, aes(x = Total_spent)) +
  geom_histogram(fill = "Sky Blue 3", binwidth = 10)+
  scale_x_continuous(breaks = seq(0, 3000, 100)) +
  theme(axis.text.x = element_text(angle=45, vjust=1, hjust=1))
g8 <- ggplot(customers)+
  geom_density(aes(x = wines), color = "Red", fill = 0.7)+
  geom_density(aes(x = fruits), color = "Orange", fill = 0.7)+
  geom_density(aes(x = meat), color = "Blue", fill = 0.7)+
  geom_density(aes(x = fish), color = "Green", fill = 0.7)+
  geom_density(aes(x = sweet), color = "Black", fill = 0.7)+
  geom_density(aes(x = gold), color = "Sky Blue 3", fill = 0.7)+
  coord_cartesian(xlim = c(0, 500))+
  xlab("")

grid.arrange(g1, g2, g3, g4, g5, g6, g7, g8, ncol=4)
```



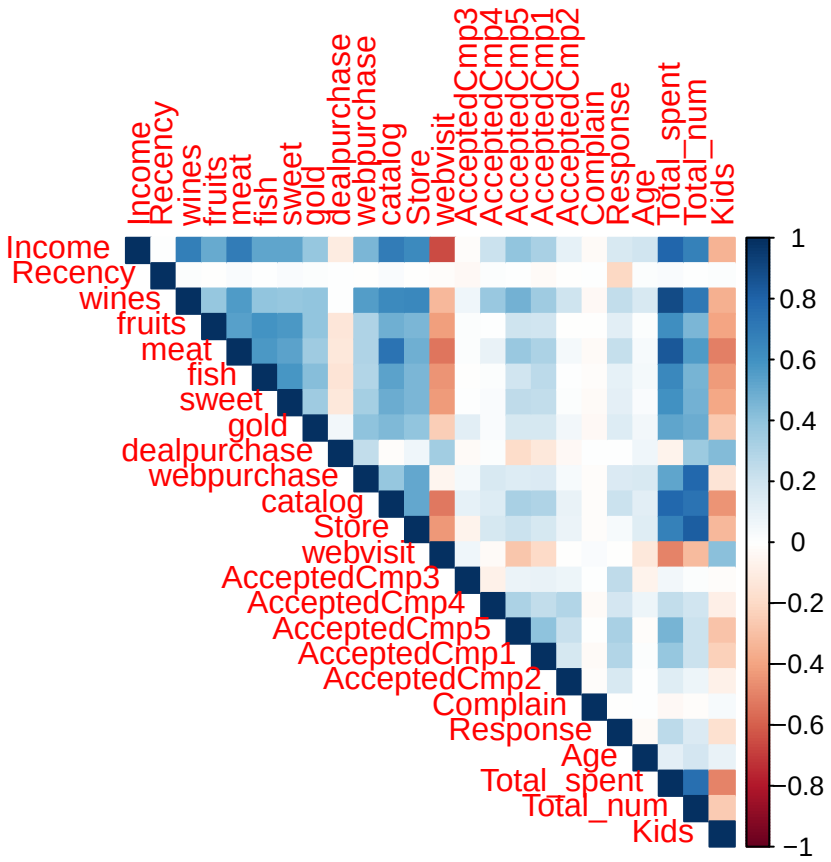
5. Correlation Plot

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.1.3
```

```
## corrplot 0.92 loaded
```

```
#Getting correlation matrix
cust_cor <- cor(customers[,3:26])
corrplot(cust_cor,
         method = "color",
         type = "upper")
```




```
         # order = "hclust",
         # col = brewer.pal(n=10, name="RdYlBu"))
# png(file="corr.png", res=300, width=4500, height=4500)
# dev.off()
```

Here, based on the correlation plot, we could find that AcceptCmp*(1 if customer accepted the offer in the 1st campaign, 0 otherwise) are very much likely not to have any correlation with other variables, thus, we will remove these when we do the segementation. Secondly, there are some interesting findings. For example, the Income is highly negativly related to webvisit, while dealpurchase seems like the only variable to have the positive relationship with webvist. It could tell us people who like are not with high income usually will go to website to to some deal searching. And it is consistant result with our real world problem.

Categorical Data

```
# Encoding the categorical features to numeric
customers_copy <- customers
```

```
customers_copy <- customers_copy %>%
  mutate(Education = case_when(
    Education == "graduate" ~ 1, Education == "non-graduate" ~ 0))
customers_copy <- customers_copy %>%
  mutate(Marital_Status = case_when(
    Marital_Status == "Taken" ~ 1, Marital_Status == "Single" ~ 0))

str(customers_copy$Education)
```

```
##  num [1:2212] 1 1 1 1 1 1 1 1 1 1 ...
```

```
str(customers_copy$Marital_Status)
```

```
##  num [1:2212] 0 0 1 1 1 1 0 1 1 1 ...
```

```
dim(customers_copy)
```

```
## [1] 2212   26
```

```
# glimpse(customers_copy)
```

```
miss_var_summary(customers_copy)
```

```
## # A tibble: 26 x 3
##    variable       n_miss pct_miss
##    <chr>           <int>    <dbl>
##  1 Education           0        0
##  2 Marital_Status      0        0
##  3 Income              0        0
##  4 Recency             0        0
##  5 wines               0        0
##  6 fruits              0        0
##  7 meat                0        0
##  8 fish                0        0
##  9 sweet               0        0
## 10 gold                0        0
## # ... with 16 more rows
```

Segmentation

Pre-processing Data

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
# preprocessing the data, use numerical data to do dimension reduction
customers_copy_pre <-
  preProcess(customers_copy[,c(3:15, 23:25)], method = c("center", "scale"))

# normalizing
customers_copy <- predict(customers_copy_pre, customers_copy[,c(3:15, 23:25)])
summary(customers_copy)
```

```
##      Income           Recency              wines             fruits
##  Min.   :-2.3333   Min.   :-1.6936473   Min.   :-0.9050   Min.   :-0.6625
##  1st Qu.:-0.7769   1st Qu.:-0.8644347   1st Qu.:-0.8339   1st Qu.:-0.6122
##  Median :-0.0273   Median :-0.0006716   Median :-0.3848   Median :-0.4612
##  Mean   : 0.0000   Mean   : 0.0000000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.7678   3rd Qu.: 0.8630914   3rd Qu.: 0.5921   3rd Qu.: 0.1678
##  Max.   : 5.1302   Max.   : 1.7268545   Max.   : 3.5210   Max.   : 4.3446
##       meat              fish              sweet              gold
##  Min.   :-0.7448   Min.   :-0.6874   Min.   :-0.6582   Min.   :-0.8495
##  1st Qu.:-0.6735   1st Qu.:-0.6326   1st Qu.:-0.6339   1st Qu.:-0.6755
##  Median :-0.4416   Median :-0.4683   Median :-0.4635   Median :-0.3757
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.2908   3rd Qu.: 0.2255   3rd Qu.: 0.1449   3rd Qu.: 0.2335
##  Max.   : 6.9473   Max.   : 4.0413   Max.   : 5.7179   Max.   : 5.3585
##   dealpurchase      webpurchase          catalog            Store
##  Min.   :-1.2079   Min.   :-1.49084   Min.   :-0.9128   Min.   :-1.7861
##  1st Qu.:-0.6883   1st Qu.:-0.76149   1st Qu.:-0.9128   1st Qu.:-0.8633
##  Median :-0.1687   Median :-0.03215   Median :-0.2296   Median :-0.2481
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.3510   3rd Qu.: 0.69720   3rd Qu.: 0.4535   3rd Qu.: 0.6747
##  Max.   : 6.5863   Max.   : 8.35532   Max.   : 8.6515   Max.   : 2.2127
##     webvisit            Age            Total_spent        Total_num
##  Min.   :-2.1939   Min.   :-2.31476   Min.   :-0.9996   Min.   :-1.94111
##  1st Qu.:-0.9571   1st Qu.:-0.69105   1st Qu.:-0.8934   1st Qu.:-0.89831
##  Median : 0.2798   Median :-0.09284   Median :-0.3490   Median : 0.01414
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.6920   3rd Qu.: 0.84720   3rd Qu.: 0.7315   3rd Qu.: 0.79624
##  Max.   : 6.0515   Max.   : 2.47091   Max.   : 3.1829   Max.   : 3.79431
```

PCA: Dimensionality Reduction

We use PCA as the segmentation method in this part, it is an unsupervised learning method.

Firstly, we need to standardize some of the features for PCA: Centering and Scaling

We anticipate to have 2 or 3 specific customer groups for the results

```
#Running a PCA.
customers_copy_pca <- PCA(customers_copy, graph = FALSE)

#Exploring PCA()
```

```r
# Getting the summary of the pca
summary(customers_copy_pca)
```

```
##
## Call:
## PCA(X = customers_copy, graph = FALSE)
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
## Variance               7.434   1.829   1.079   1.000   0.827   0.673   0.639
## % of var.             46.465  11.431   6.741   6.252   5.168   4.205   3.993
## Cumulative % of var.  46.465  57.896  64.637  70.889  76.056  80.261  84.254
##                        Dim.8   Dim.9  Dim.10  Dim.11  Dim.12  Dim.13  Dim.14
## Variance               0.575   0.436   0.399   0.377   0.292   0.256   0.184
## % of var.              3.592   2.728   2.493   2.359   1.828   1.597   1.149
## Cumulative % of var.  87.846  90.574  93.067  95.426  97.254  98.851 100.000
##                       Dim.15  Dim.16
## Variance               0.000   0.000
## % of var.              0.000   0.000
## Cumulative % of var. 100.000 100.000
##
## Individuals (the 10 first)
##                   Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3
## 1              |  5.491 |  4.272  0.111  0.605 |  0.396  0.004  0.005 |  0.697
## 2              |  3.189 | -2.539  0.039  0.634 | -0.654  0.011  0.042 | -1.422
## 3              |  3.020 |  1.883  0.022  0.389 |  0.182  0.001  0.004 | -0.150
## 4              |  3.136 | -2.742  0.046  0.764 | -0.568  0.008  0.033 |  0.979
## 5              |  2.597 |  0.085  0.000  0.001 |  0.715  0.013  0.076 |  0.714
## 6              |  2.554 |  1.091  0.007  0.183 |  0.891  0.020  0.122 | -0.201
## 7              |  2.124 |  0.883  0.005  0.173 |  0.972  0.023  0.210 |  0.843
## 8              |  2.854 | -2.347  0.033  0.676 |  0.136  0.000  0.002 |  1.290
## 9              |  3.552 | -3.145  0.060  0.784 | -0.267  0.002  0.006 |  0.585
## 10             |  7.523 | -4.849  0.143  0.415 |  1.338  0.044  0.032 |  0.240
##                   ctr   cos2
## 1              0.020  0.016 |
## 2              0.085  0.199 |
## 3              0.001  0.002 |
## 4              0.040  0.097 |
## 5              0.021  0.076 |
## 6              0.002  0.006 |
## 7              0.030  0.158 |
## 8              0.070  0.204 |
## 9              0.014  0.027 |
## 10             0.002  0.001 |
##
## Variables (the 10 first)
##                  Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr
## Income        |  0.853  9.796  0.728 | -0.095  0.495  0.009 | -0.226  4.750
## Recency       |  0.018  0.005  0.000 | -0.013  0.009  0.000 | -0.179  2.986
## wines         |  0.794  8.488  0.631 |  0.204  2.269  0.041 | -0.146  1.981
## fruits        |  0.681  6.236  0.464 | -0.253  3.513  0.064 |  0.266  6.567
## meat          |  0.805  8.707  0.647 | -0.234  2.989  0.055 |  0.009  0.008
```

```
## fish         |  0.704  6.674  0.496 | -0.266  3.858  0.071 |  0.223  4.625
## sweet        |  0.683  6.274  0.466 | -0.230  2.891  0.053 |  0.229  4.859
## gold         |  0.580  4.529  0.337 |  0.126  0.869  0.016 |  0.254  5.972
## dealpurchase | -0.030  0.012  0.001 |  0.794 34.508  0.631 |  0.164  2.500
## webpurchase  |  0.606  4.942  0.367 |  0.560 17.128  0.313 |  0.083  0.634
##                cos2
## Income        0.051 |
## Recency       0.032 |
## wines         0.021 |
## fruits        0.071 |
## meat          0.000 |
## fish          0.050 |
## sweet         0.052 |
## gold          0.064 |
## dealpurchase  0.027 |
## webpurchase   0.007 |
```

```
#Getting the variance of the first 8 new dimensions
customers_copy_pca$eig[,2][1:8]
```

```
##    comp 1    comp 2    comp 3    comp 4    comp 5    comp 6    comp 7    comp 8
## 46.464788 11.431160  6.741227  6.251565  5.167577  4.204525  3.992843  3.592394
```

```
#Getting the cummulative variance
customers_copy_pca$eig[,3][1:8]
```

```
##   comp 1   comp 2   comp 3   comp 4   comp 5   comp 6   comp 7   comp 8
## 46.46479 57.89595 64.63717 70.88874 76.05632 80.26084 84.25368 87.84608
```

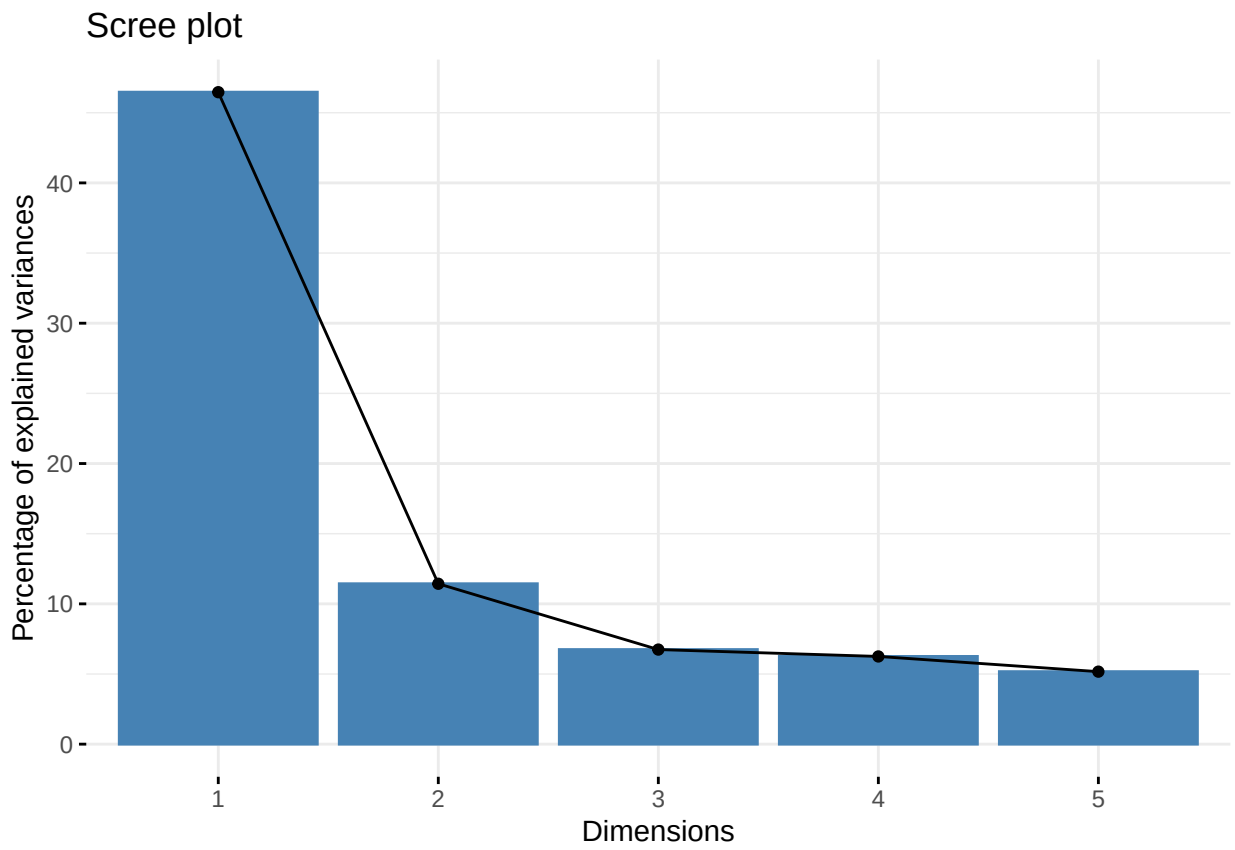
```
#Tracing variable contributions in customers_pca
customers_copy_pca$var$contrib
```

```
##                    Dim.1        Dim.2        Dim.3        Dim.4        Dim.5
## Income        9.795757924  0.495384008 4.749615e+00 2.948303e-01  2.214556314
## Recency       0.004579413  0.008693595 2.986338e+00 9.633489e+01  0.002858636
## wines         8.487984211  2.268563515 1.980708e+00 6.861465e-02 10.333163167
## fruits        6.236085347  3.513309806 6.567481e+00 2.095683e-02  9.944335904
## meat          8.706831579  2.988581730 7.649743e-03 4.109597e-02  4.915749537
## fish          6.673976602  3.858364653 4.624684e+00 4.701107e-02 10.669902888
## sweet         6.273728695  2.891186359 4.858516e+00 3.424587e-01  8.051085971
## gold          4.528624178  0.869438785 5.972192e+00 5.726921e-01 14.739096490
## dealpurchase  0.011781204 34.508008842 2.500131e+00 3.646655e-01  0.115970224
## webpurchase   4.941673488 17.127570714 6.340136e-01 1.421811e-02  0.551695066
## catalog       9.213729804  0.075283716 4.555626e-01 1.764697e-02  3.066526464
## Store         8.045872720  2.183760872 2.410467e-01 1.793061e-01  0.548661958
## webvisit      4.804018733 14.473495324 7.659895e+00 2.513166e-01  0.245821064
## Age           0.365799504  2.647375878 5.671471e+01 1.445942e+00 31.651862827
## Total_spent  12.126992089  0.019137688 9.480278e-04 3.958298e-03  2.550285584
## Total_num     9.782564508 12.071844516 4.650595e-02 3.952069e-04  0.398427908
```

Visualizing PCA

1. Eigenvalues/variances vs. the Number of Dimensions

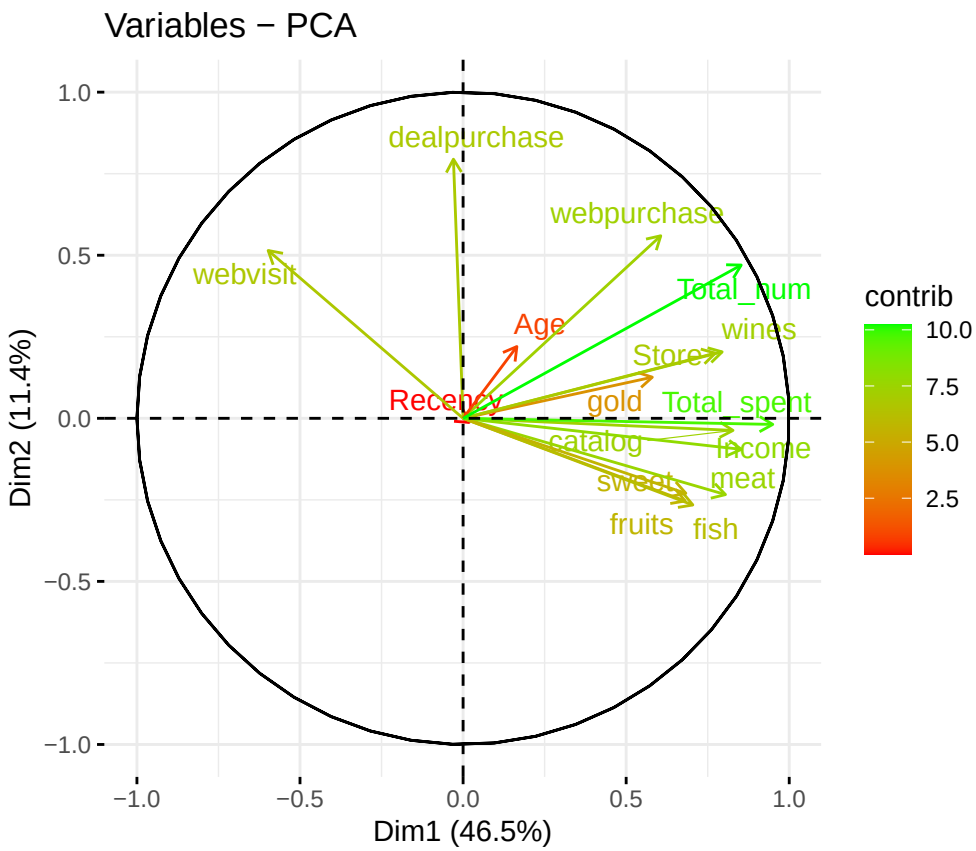
```
# Plot the eigenvalues/variances against the number of dimensions
fviz_screeplot(customers_copy_pca, ncp=5)
```



In this plot, we could find that the first principal component will explain about 46% variance, second principal component is about 11% and the third principal component is about 6%. Thus, we will use first two principal component to do the following analysis.

2. Variable Contributions(loading plots)

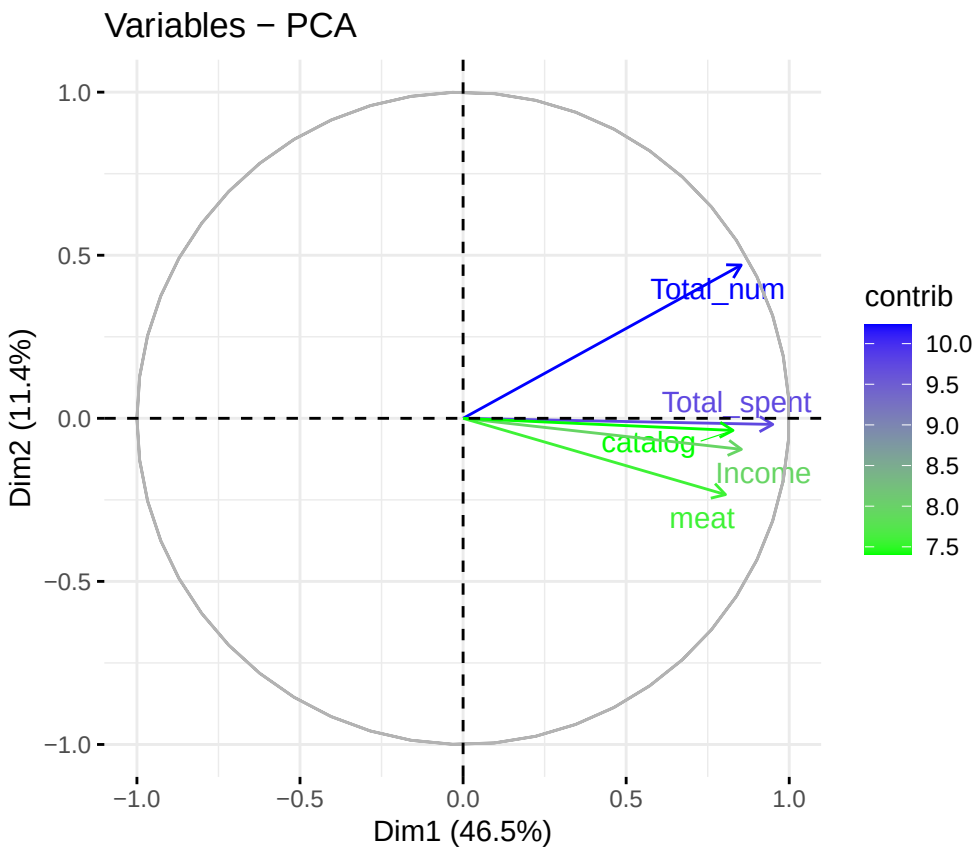
```
#Creating a factor map for the variable contributions
fviz_pca_var(customers_copy_pca,
             col.circle = 'black',
             col.var = "contrib",
             gradient.cols = c("red", "green"),
             repel = TRUE)
```



Age and recency are have the lowest contribution, while total spentand total number has the largest

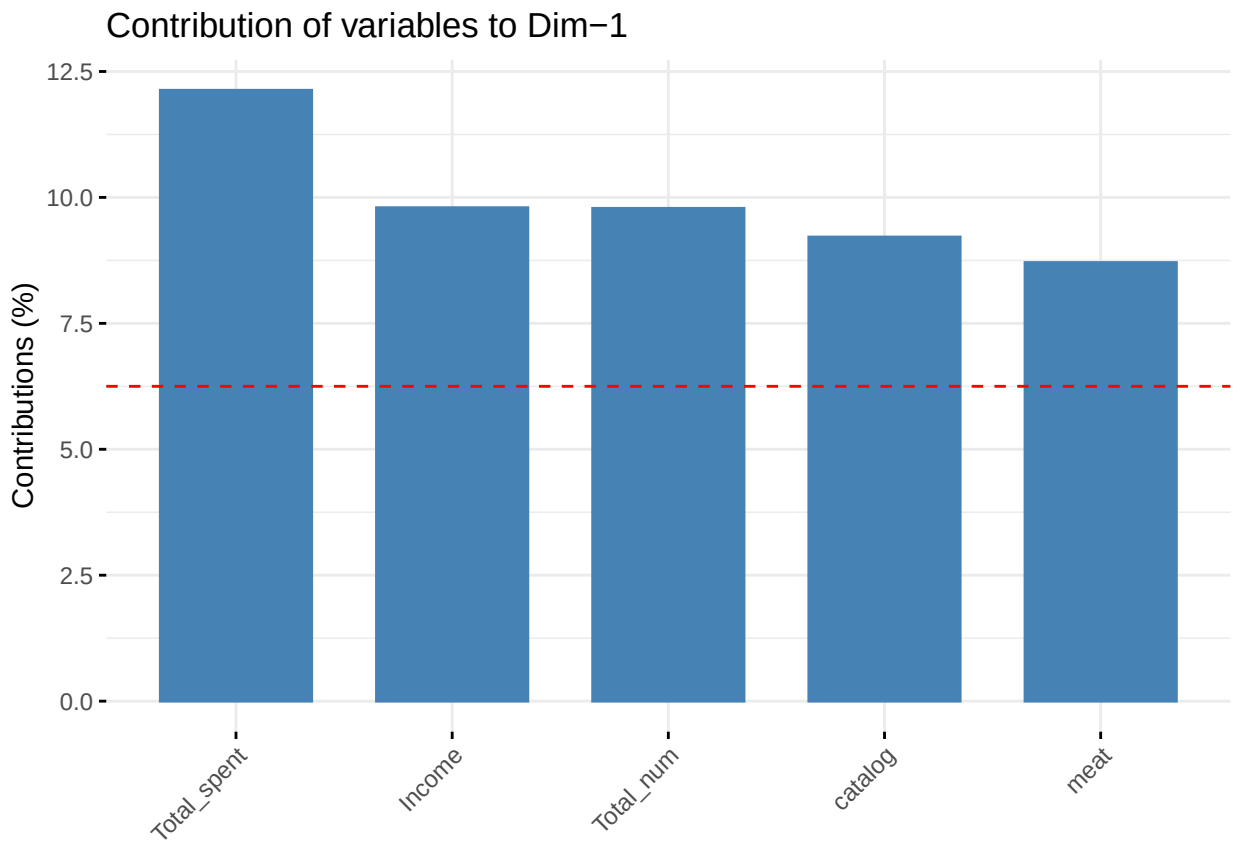
3. Top 5 Variable Contributions

```
#Creating a factor map for the top 5 variables with the highest contributions.
fviz_pca_var(customers_copy_pca,
             select.var = list(contrib = 5),
             col.var = "contrib",
             gradient.cols = c("green", "blue"),
             repel = TRUE)
```



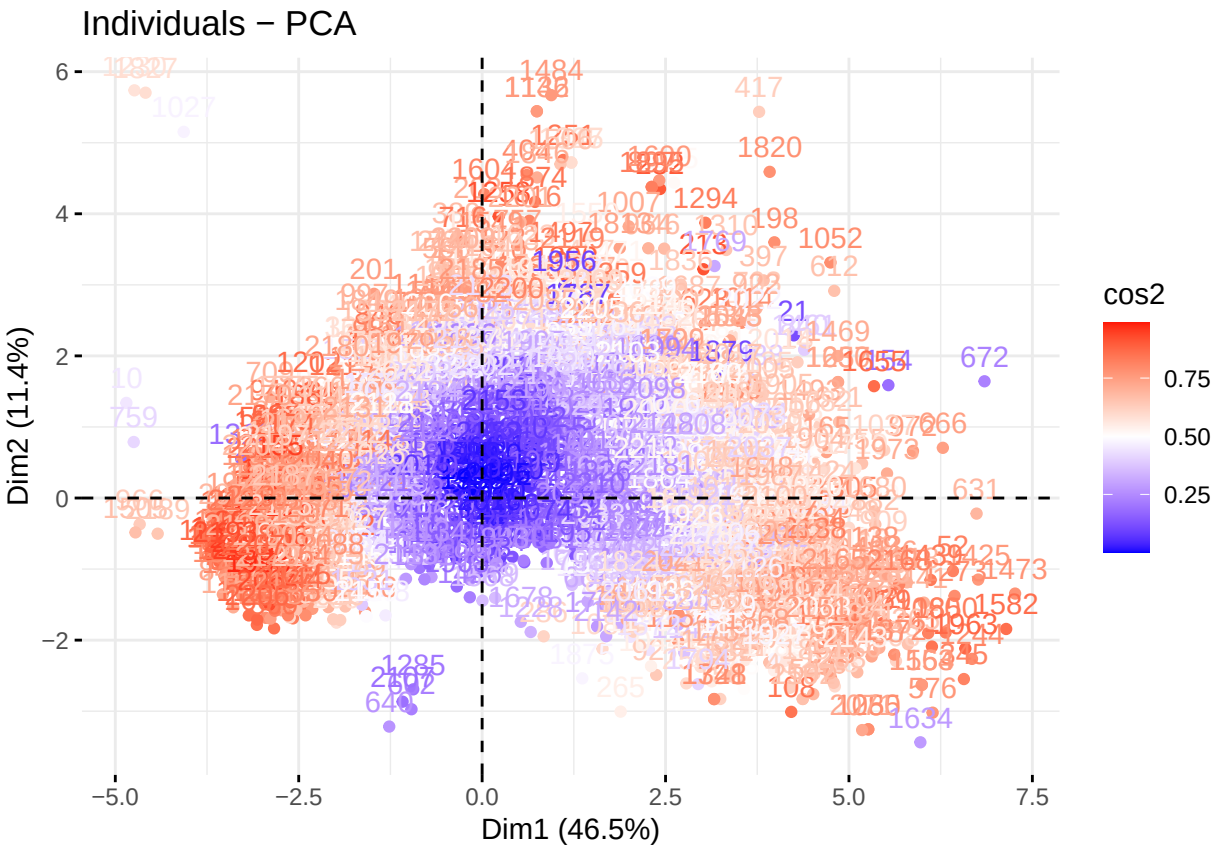
4. Top 5 Variable Contributions in Barplot

```
fviz_contrib(customers_copy_pca, choice = "var", axes = 1, top = 5)
```



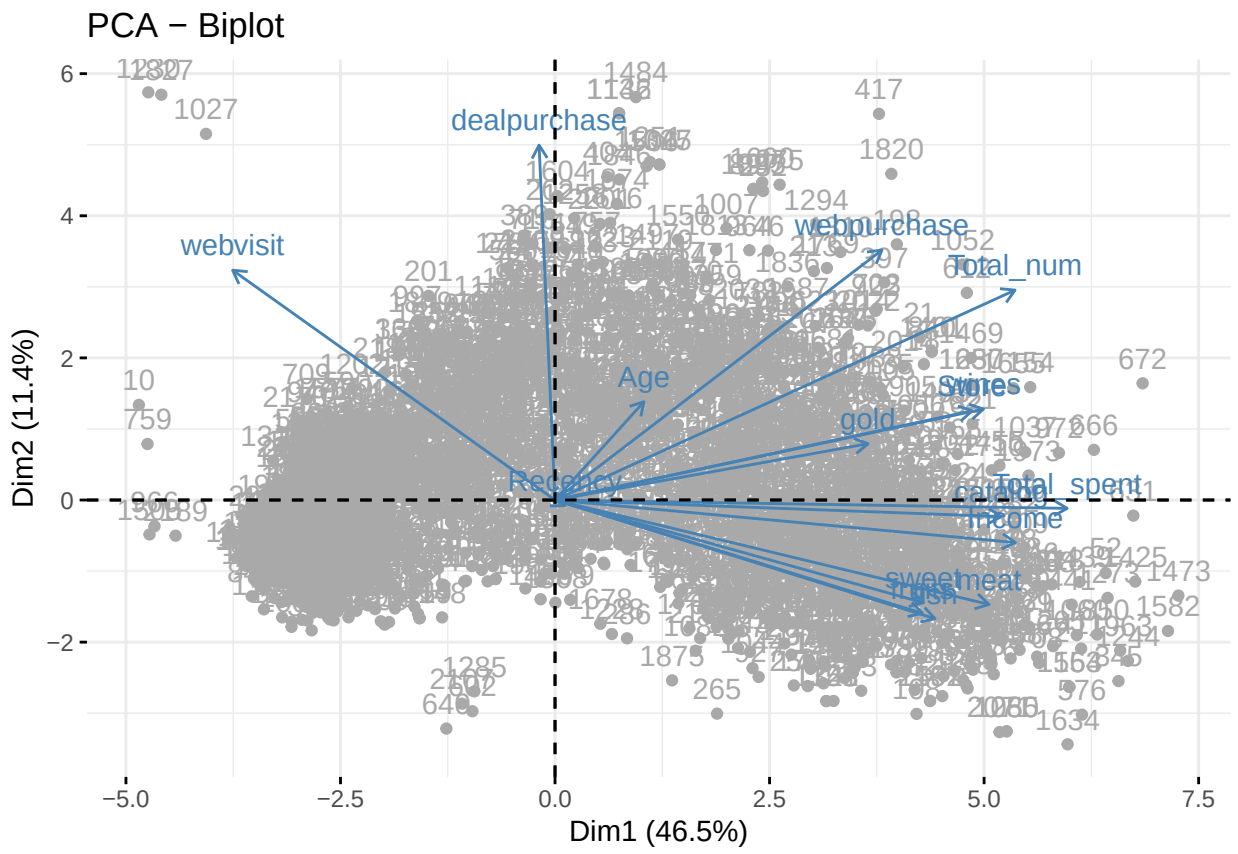
5. Graph of Individuals(Score plot)

```
fviz_pca_ind(customers_copy_pca, col.ind="cos2") +
  scale_color_gradient2(low="blue", mid="white",high="red", midpoint=0.50)
```



6. Biplot

```
fviz_pca_biplot(customers_copy_pca, col.ind = 'dark grey')
```



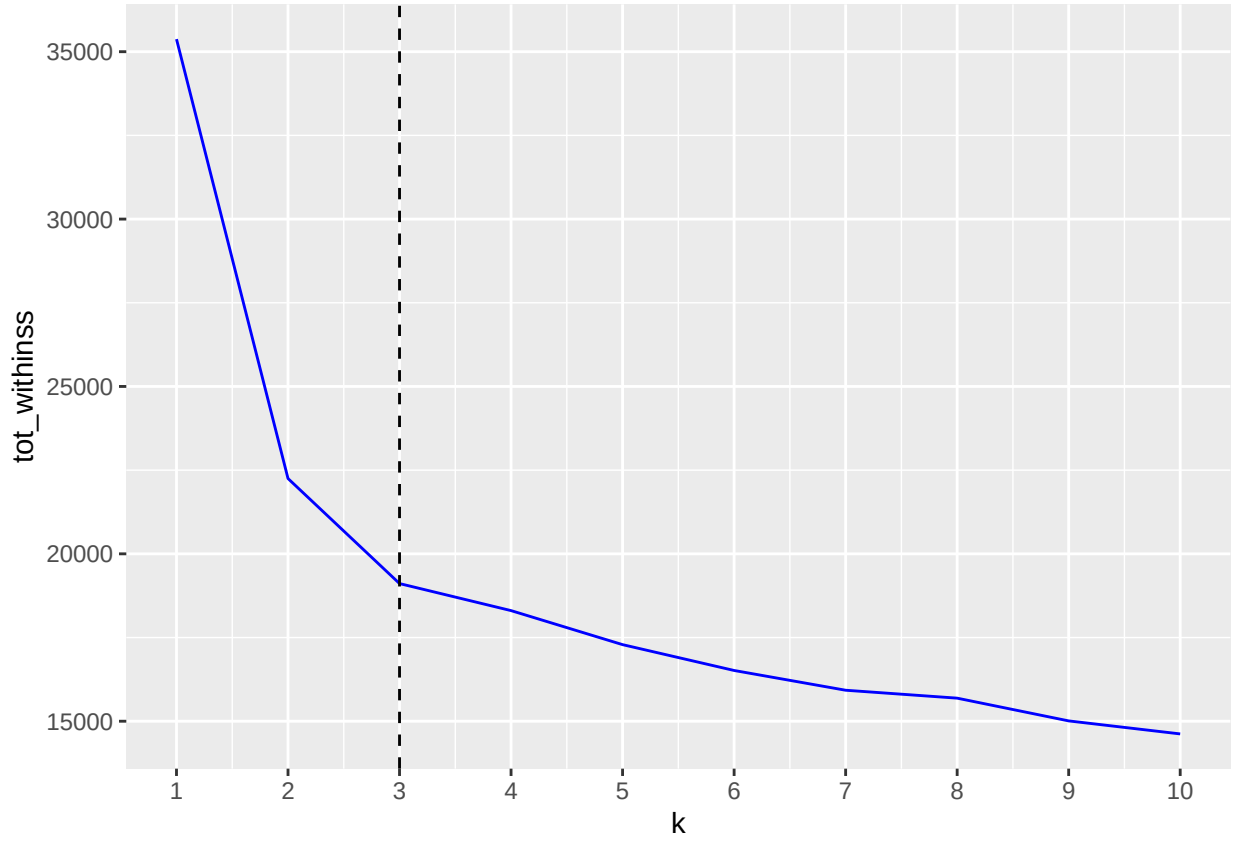
Biplot is the combination of score plot and loading plot

K-Means Clustering

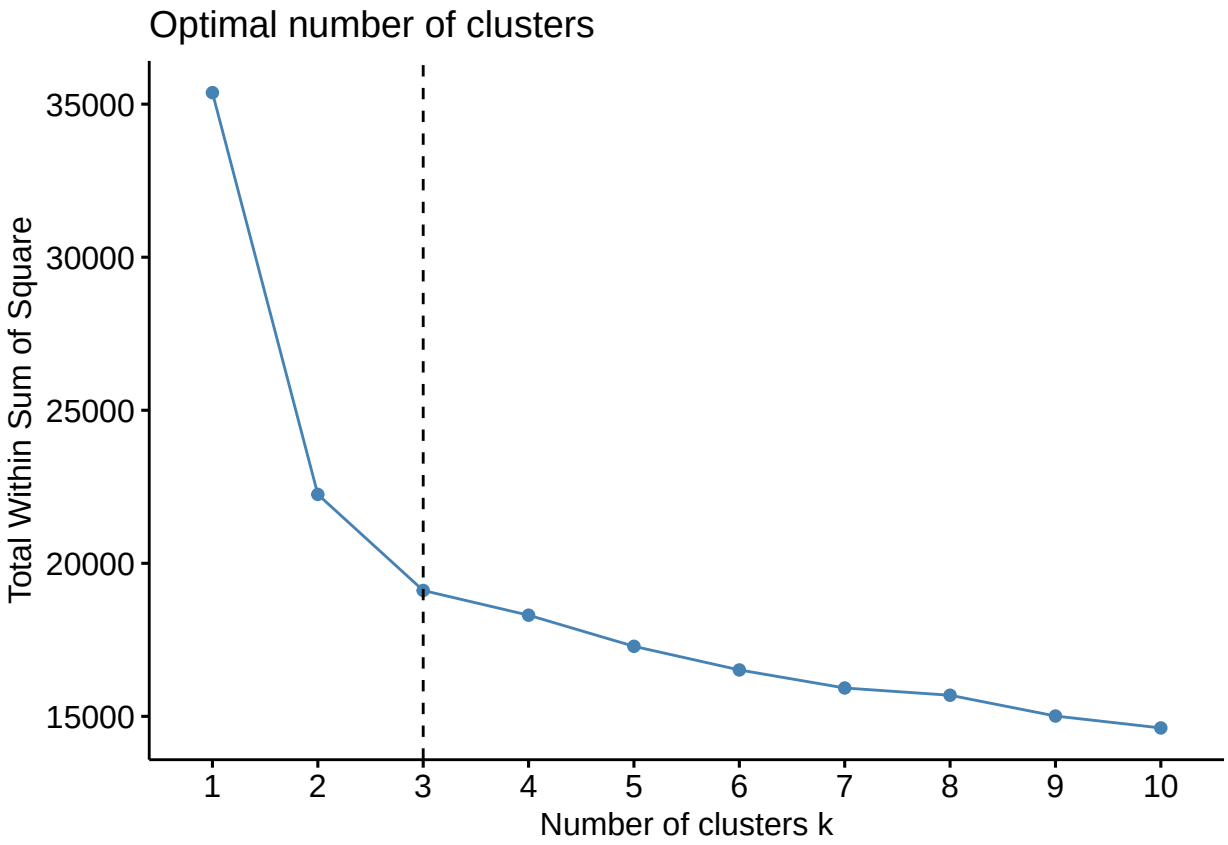
```
#The elbow method to find cluster number with kmean method
library(purrr)
tot_withinss <- map_dbl(1:10, function(k){
  model <- kmeans(x = customers_copy, centers = k)
  model$tot.withinss
})

elbow_df <- data.frame(
       k = 1:10,
       tot_withinss = tot_withinss)
head(elbow_df)
```

```
##   k tot_withinss
## 1 1     35376.00
## 2 2     22250.00
## 3 3     19112.24
## 4 4     18304.32
## 5 5     17287.71
## 6 6     16515.84
```

```
#plotting the elbow plot
ggplot(elbow_df, aes(k, tot_withinss)) +
  geom_line(col='blue') +
  scale_x_continuous(breaks = 1:10)+
  geom_vline(xintercept=3,linetype=2)
```

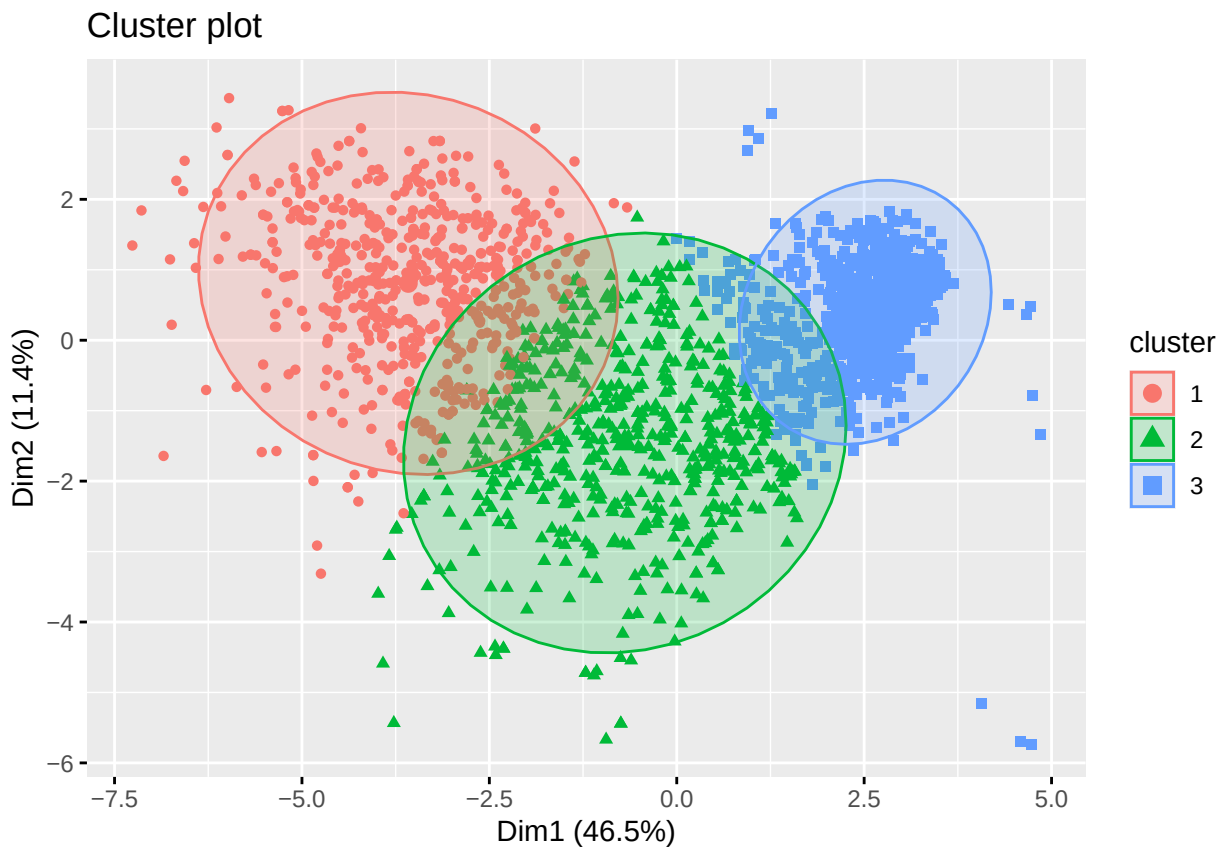



```
#plot elbow with package function
fviz_nbclust(customers_copy,kmeans,method="wss")+
  geom_vline(xintercept=3,linetype=2)
```



We can see that two way of the scree plot methods will plot the same result. And we will choose three as the cluster number.

```
set.seed(1313)
res_kmeans <- kmeans(customers_copy, centers = 3, nstart = 10)
# autoplot(res_kmeans, data=customers_copy)
fviz_cluster(res_kmeans,
             customers_copy,
             geom = "point",
             ellipse.type = "norm",
             repel = TRUE)
```




```
set.seed(3)

#Building a k-means model with a k of 2
customers_md <- kmeans(customers_copy, center = 3)

#Extracting the vector of cluster assignment from the model
clust_customers <- as.factor(customers_md$cluster)

#Building the segment_customers dataframe
segment_customers <- mutate(customers_copy, cluster = clust_customers)

#Calculating the mean for each category
count(segment_customers, cluster)
```

```
##   cluster    n
## 1       1 1013
## 2       2  605
## 3       3  594
```

```
#Adding the cluster variable to the original dataframe
customers <- customers %>% mutate(cluster = segment_customers$cluster)
head(customers, n = 3)
```

```
##   Education Marital_Status Income Recency wines fruits meat fish sweet gold
## 1  graduate         Single  58138      58   635     88  546  172    88   88
```

```
## 2  graduate         Single  46344      38    11      1    6    2     1    6
## 3  graduate          Taken  71613      26   426     49  127  111    21   42
##   dealpurchase webpurchase catalog Store webvisit AcceptedCmp3 AcceptedCmp4
## 1            3           8      10     4        7            0            0
## 2            2           1       1     2        5            0            0
## 3            1           8       2    10        4            0            0
##   AcceptedCmp5 AcceptedCmp1 AcceptedCmp2 Complain Response Age Total_spent
## 1            0            0            0        0        1  65        1617
## 2            0            0            0        0        0  68          27
## 3            0            0            0        0        0  57         776
##   Total_num Kids cluster
## 1        25    0       3
## 2         6    2       1
## 3        21    0       2
```

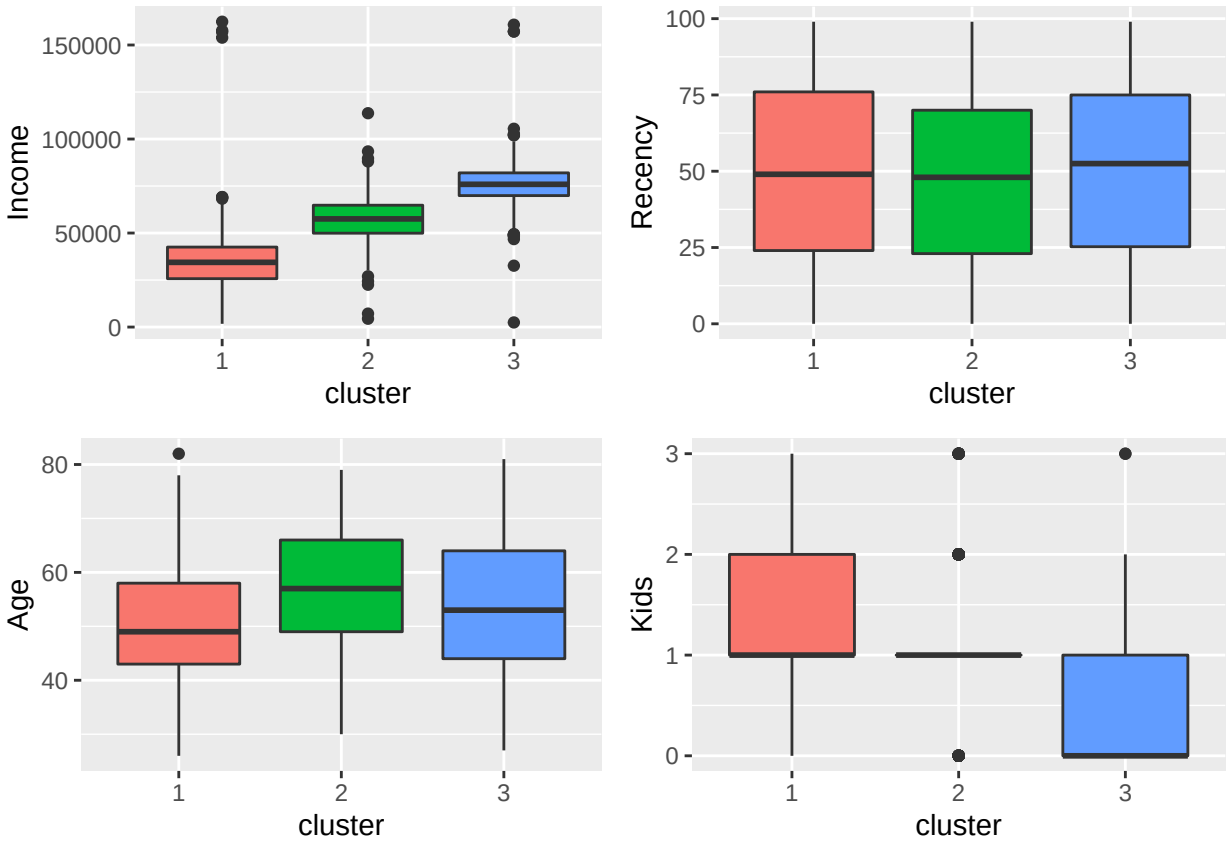
```
#confirming
count(customers, cluster)
```

```
##   cluster    n
## 1       1 1013
## 2       2  605
## 3       3  594
```

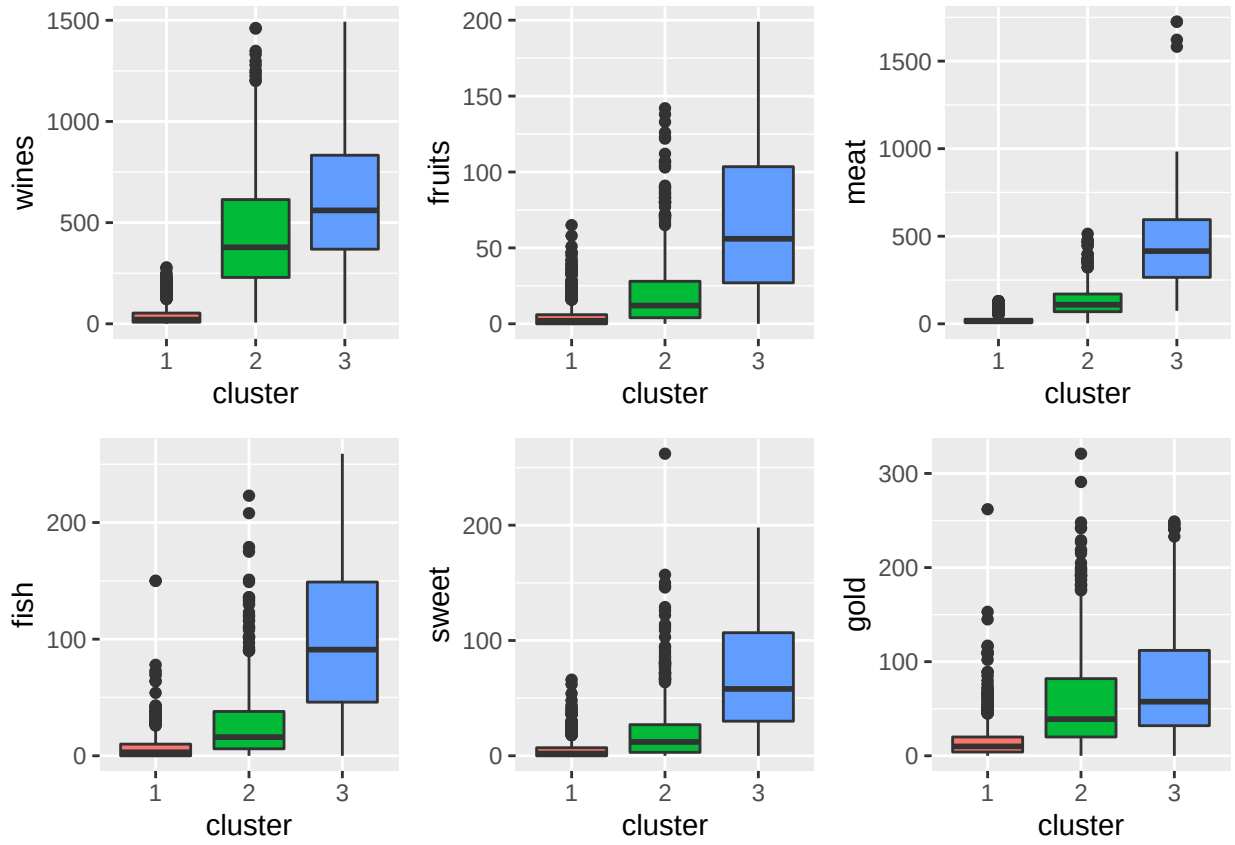
Segmenting Results

```
g1=ggplot(data = customers, aes(x = cluster, y = Income, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

g2=ggplot(data = customers, aes(x = cluster, y = Recency, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

g111=ggplot(data = customers, aes(x = cluster, y = Age, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

g112=ggplot(data = customers, aes(x = cluster, y = Kids, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

g3=ggplot(data = customers, aes(x = cluster, y = wines, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

g4=ggplot(data = customers, aes(x = cluster, y = fruits, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

g5=ggplot(data = customers, aes(x = cluster, y = meat, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")
```

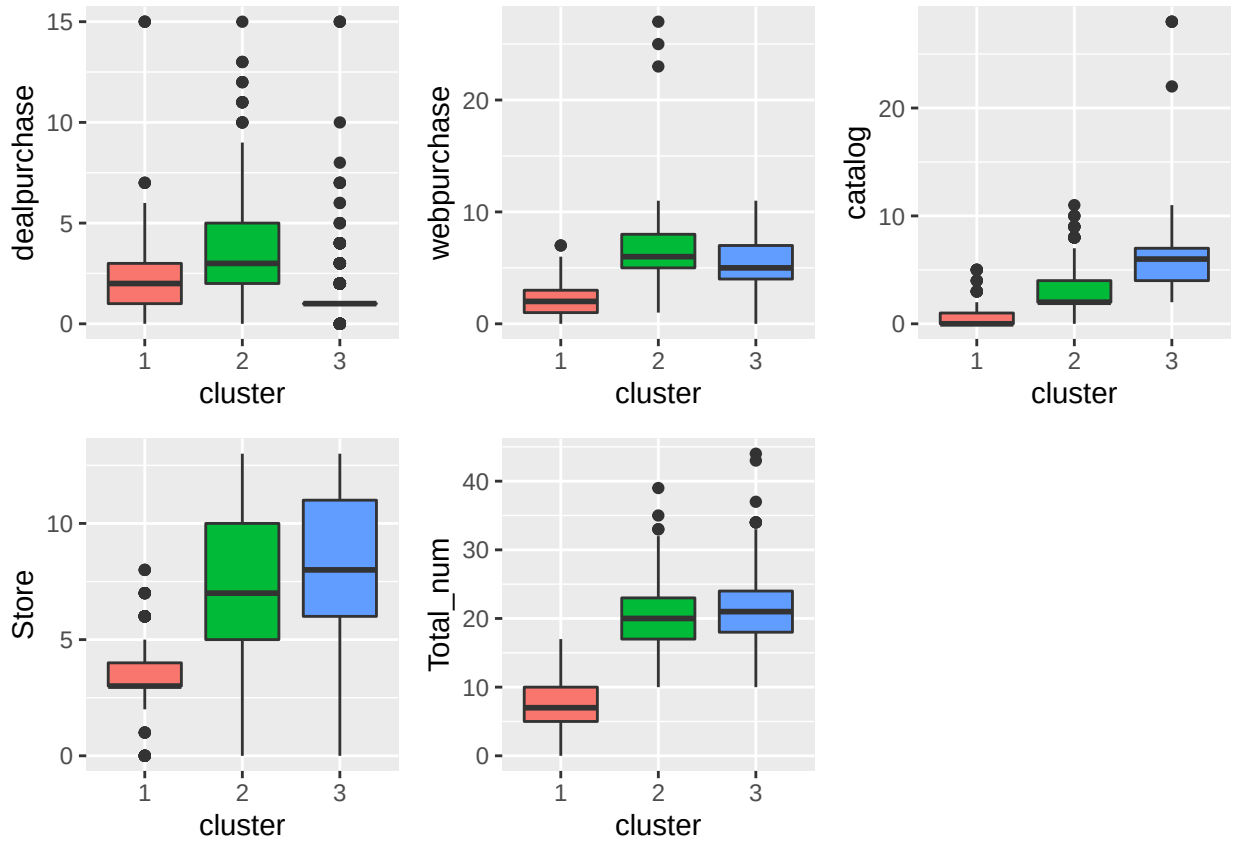
```
g6=ggplot(data = customers, aes(x = cluster, y = fish, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

g7=ggplot(data = customers, aes(x = cluster, y = sweet, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

g8=ggplot(data = customers, aes(x = cluster, y = gold, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

g9=ggplot(data = customers, aes(x = cluster, y = dealpurchase, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

g10=ggplot(data = customers, aes(x = cluster, y = webpurchase, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

g11=ggplot(data = customers, aes(x = cluster, y = catalog, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

g12=ggplot(data = customers, aes(x = cluster, y = Store, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

g13=ggplot(data = customers, aes(x = cluster, y = Total_num, fill = cluster))+
  geom_boxplot() +
  theme(legend.position = "none")

grid.arrange(g1, g2, g111, g112, ncol=2)
```




```
grid.arrange(g3, g4, g5, g6, g7, g8, ncol=3)
```




```
grid.arrange(g9, g10, g11, g12, g13, ncol=3)
```



Conclusion

Basically, from the k-means clustering, we can separate customers into three groups with the following characteristics:

Cluster 1:

Low purchasing power customer

No wine/meat

time gap between the going to market is large

have more kids

Cluster 2:

Median purchasing power customer

prefer deal/web, but have some level of purchasing potential

love buying wines and gold

Cluster 3:

High purchasing power customer

Prefer catalog/store

Purchasing more product

Future steps

We will explore factor analysis which is one of the most populare analysis technique in market analysis