Customer segmentation for products marketing

Reagan Kesseku

2023-01-17

```r
# set working directory ---------------------
setwd("D:/Ph.D_materials/Programming/R_programming/mdsr/customer-segmentation_analysis")
```

```r
# Load functions and packages ---------------------------
source("pkg.R")
```

DATA VISUALIZATION

```r
# Import the Mall customers data --------------------------------
customers <- vroom::vroom("Mall_Customers.csv", col_names = T)

customers_old <- customers

# Take a glimpse of the data sets
customers %>%
    glimpse()
```

```
## Rows: 400
## Columns: 5
## $ CustomerID               <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14~
## $ Gender                   <chr> "Male", "Male", "Female", "Female", "Female",~
## $ Age                      <dbl> 19, 21, 20, 23, 31, 22, 35, 23, 64, 30, 67, 3~
## $ `Annual Income (k$)`     <dbl> 15, 15, 16, 16, 17, 17, 18, 18, 19, 19, 19, 1~
## $ `Spending Score (1-100)` <dbl> 39, 81, 6, 77, 40, 76, 6, 94, 3, 72, 14, 99, ~
```

```r
# Change gender to factor -----------------------
customers <- customers %>%
    mutate(Gender = factor(Gender))



# rename income and spending variables -----------------------------------
customers <- customers %>%
    rename(annual_income = "Annual Income (k$)", spending_score = "Spending Score (1-100)")
```

There are 400 observations and 5 variables in the movies data. Additionally, all variables were numerical. However, we convert the class variables to factors.

```
# check the five number summary and other measures of Amount
# ---------------------------------------------------------
d <- favstats(Age ~ Gender, data = customers)

knitr::kable(d, digits = 3, format.args = list(scientific = FALSE), caption = "Descriptive summary of ag
```

EXPLORATORY DATA ANAYSIS

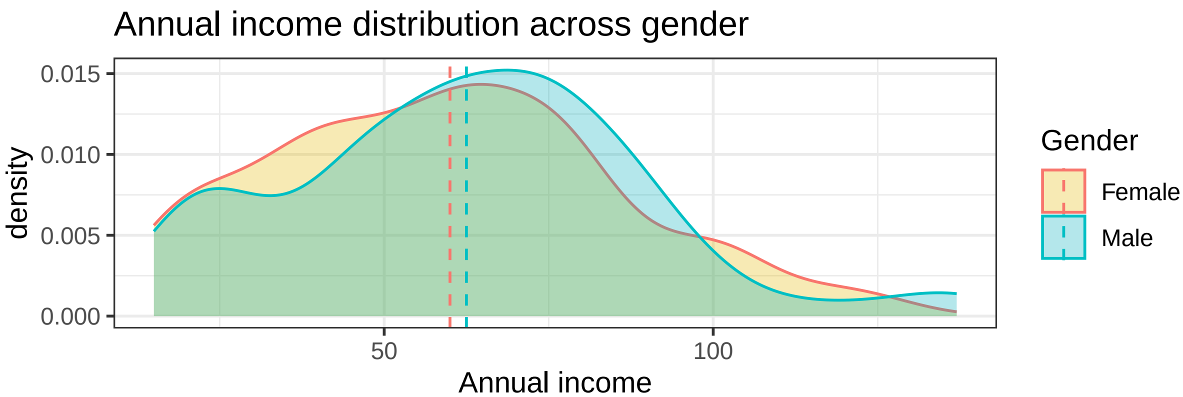
Table 1: Descriptive summary of age by gender.

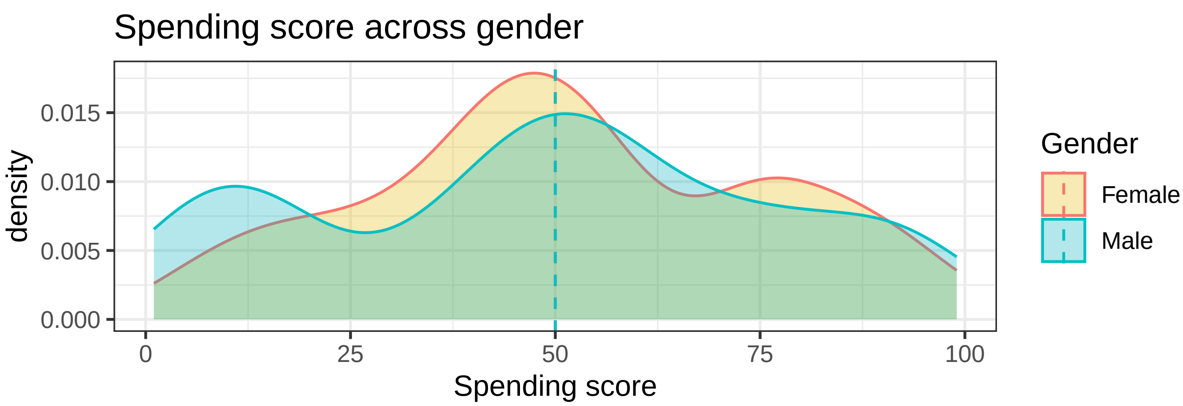
Gender	min	Q1	median	Q3	max	mean	sd	n	missing
Female	18	29.0	35	47.5	68	38.1	12.6	224	0
Male	18	27.8	37	50.5	70	39.8	15.5	176	0

```
tt <- ggplot(data = customers, aes(x = Gender)) + geom_bar(aes(fill = Gender)) +
    labs(y = "Number of values in class", title = "Bar graph of the target variable class") +
    theme_bw()
```

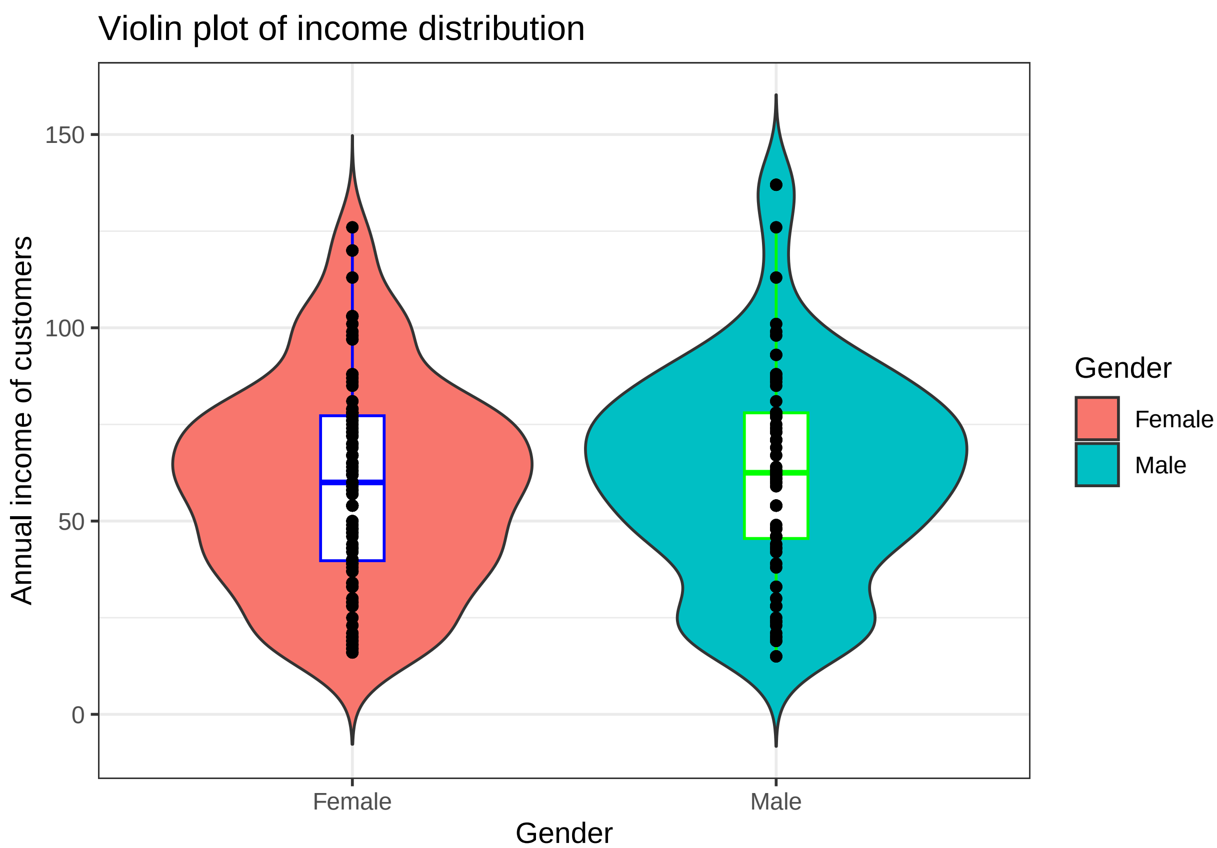
Clearly, this shows a highly imbalanced classification problem.

```
# calculate median income and spending across gender
mu_income <- plyr::ddply(customers, "Gender", summarise, grp.median = median(annual_income))

mu_spend <- plyr::ddply(customers, "Gender", summarise, grp.median = median(spending_score))

# plot graph -----------
p1 <- ggplot(data = customers, aes(x = annual_income, color = Gender, fill = Gender)) +
    geom_density(alpha = 0.3) + geom_vline(data = mu_income, aes(xintercept = grp.median,
    color = Gender), linetype = "dashed") + scale_fill_manual(values = c("#E7B800",
    "#00AFBB")) + labs(title = "Annual income distribution across gender", x = "Annual income") +
    theme_bw()


p2 <- ggplot(data = customers, aes(x = spending_score, color = Gender, fill = Gender)) +
    geom_density(alpha = 0.3) + geom_vline(data = mu_spend, aes(xintercept = grp.median,
    color = Gender), linetype = "dashed") + scale_fill_manual(values = c("#E7B800",
    "#00AFBB")) + labs(title = "Spending score across gender", x = "Spending score") +
    theme_bw()

p1/p2
```






```r
# Change violin plot colors by gender ------------------------------------
p <- ggplot(data = customers, aes(x = Gender, y = annual_income)) + geom_violin(trim = FALSE,
    aes(fill = Gender)) + geom_boxplot(width = 0.15, color = c("blue", "green"),
    fill = c("white", "white")) + geom_point() + labs(y = "Annual income of customers",
    title = "Violin plot of income distribution") + theme_bw()
p
```




```
customers %>%
    select(Gender) %>%
    unique()
```

```
## # A tibble: 2 x 1
##   Gender
##   <fct>
## 1 Male
## 2 Female
```

```
# scale spending and income variable -----------------------------------
customers <- customers %>%
    mutate(annual_income = scale(annual_income), spending_score = scale(spending_score))
```

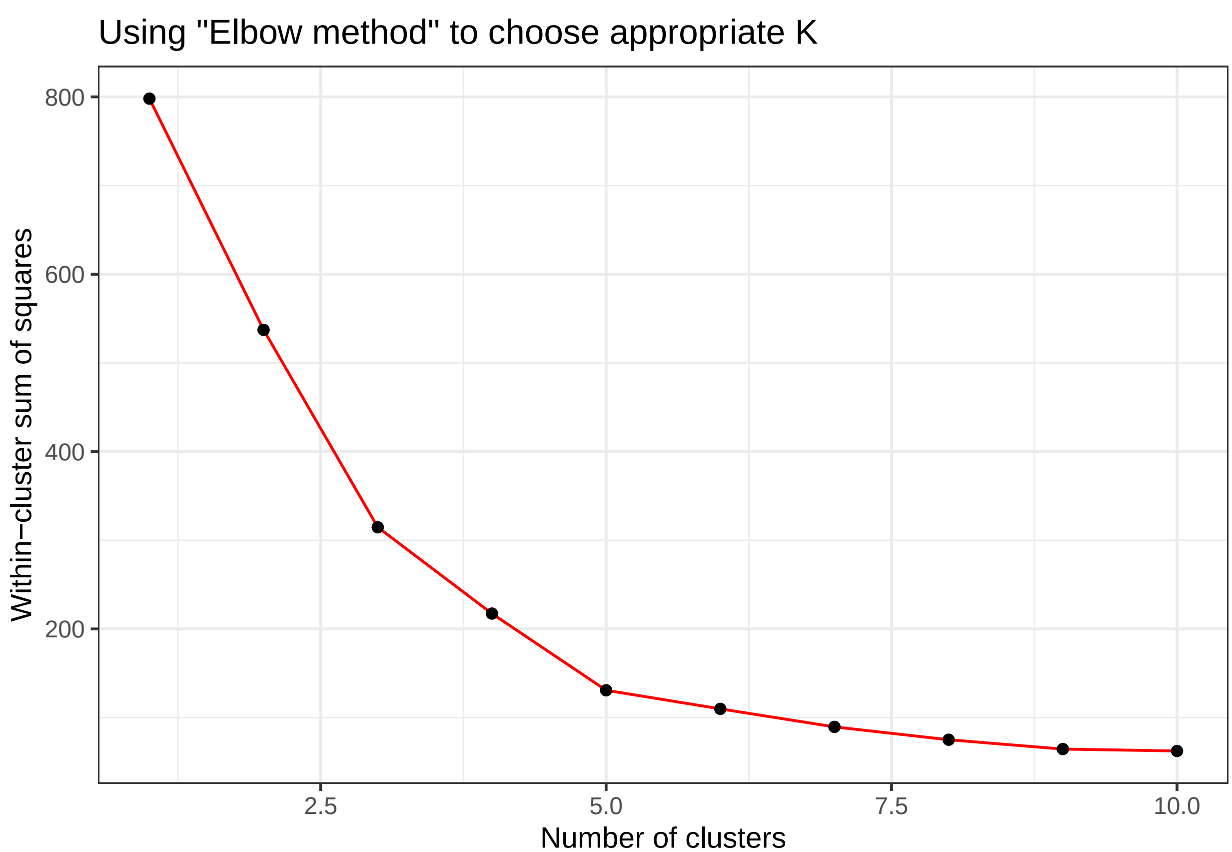
The descriptive statistics show that the amount values are highly variable. This suggests the we scale the data as it helps with most machine learning algorithms.

```
# Elbow Method for finding the optimal number of clusters
set.seed(123)
# Compute and plot wss for k = 2 to k = 15.
k.max <- 10
kk = customers %>%
    select(annual_income, spending_score)
data <- kk
wcss <- sapply(1:k.max, function(k) {
```

```
    kmeans(data, k, nstart = 10, iter.max = 350)$tot.withinss
})
wcss
```

```
##  [1] 798.0 537.3 314.6 217.3 130.8 109.8  89.5  75.0  64.5  62.3
```

```
kt <- data.frame(k.max = 1:k.max, wcss = wcss)

# plot graph
ggplot(kt, aes(x = k.max, y = wcss)) + geom_line(color = "red") + geom_point() +
    labs(y = "Within-cluster sum of squares", x = "Number of clusters", title = "Using \"Elbow method\"
    theme_bw()
```




```
set.seed(6)
library(cluster)

clust_Variables <- customers %>%
    select(annual_income, spending_score) %>%
    kmeans(5, iter.max = 300, nstart = 2) %>%
    fitted("classes") %>%
    as.factor()


customers <- customers %>%
```

```r
    mutate(clust_Variables = clust_Variables)


p3 <- customers %>%
    ggplot(aes(x = annual_income, y = spending_score)) + geom_point(aes(color = clust_Variables),
    alpha = 0.5) + scale_color_brewer(palette = "Set2") + labs(y = "Spending score of customers",
    x = "Annual income of customers", title = "K-Means clustering of customers income and spending") +
    theme_bw()


p4 <- customers %>%
    ggplot(aes(x = annual_income, y = spending_score)) + geom_point(aes(color = clust_Variables),
    alpha = 0.5) + scale_color_brewer(palette = "Set2") + labs(y = "Spending score of customers",
    x = "Annual income of customers", title = "K-Means clustering of customers income and spending by Ge
    facet_wrap(~Gender, nrow = 1) + theme_bw()
p3/p4
```


```r
# selecting cluster 4
customers %>%
    select(CustomerID) %>%
    filter(clust_Variables == "4") %>%
    as.vector()
```

```
## $CustomerID
##  [1] 124 126 128 130 132 134 136 138 140 142 144 146 148 150 152 154 156 158 160
```

```
## [20] 162 164 166 168 170 172 174 176 178 180 182 184 186 188 190 192 194 196 198
## [39] 200 324 326 328 330 332 334 336 338 340 342 344 346 348 350 352 354 356 358
## [58] 360 362 364 366 368 370 372 374 376 378 380 382 384 386 388 390 392 394 396
## [77] 398 400
```