Mall Customer Segmentation report

Subhajit Karmakar

2023-12-26

The data given to us contains information on customers in a particular supermarket. There are 5 variables in total. Our goal is to group the similar customers using **K-means** algorithm, based on the their purchase behavior.

Data source: https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python

Necessary libraries & data importation

```
library(tidyverse)
library(factoextra)
library(kableExtra)

df <- read_csv('D:/Internships/Prodigy/Task 2 (Clustering)/Data2.csv',
               show_col_types = F)
```

Glimpse at the data

```
## Rows: 200
## 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, …
```

Comment: There are 200 rows and 5 columns in the data.

Checking for missing value

```
df %>% is.na() %>% sum()
```

```
## [1] 0
```

∴ There is no missing value in the data.

Summary statistics

```
##    CustomerID        Gender               Age        Annual Income (k$)
##  Min.   :  1.00   Length:200         Min.   :18.00   Min.   : 15.00
##  1st Qu.: 50.75   Class :character   1st Qu.:28.75   1st Qu.: 41.50
##  Median :100.50   Mode  :character   Median :36.00   Median : 61.50
##  Mean   :100.50                      Mean   :38.85   Mean   : 60.56
##  3rd Qu.:150.25                      3rd Qu.:49.00   3rd Qu.: 78.00
##  Max.   :200.00                      Max.   :70.00   Max.   :137.00
##  Spending Score (1-100)
##  Min.   : 1.00
##  1st Qu.:34.75
##  Median :50.00
##  Mean   :50.20
##  3rd Qu.:73.00
##  Max.   :99.00
```

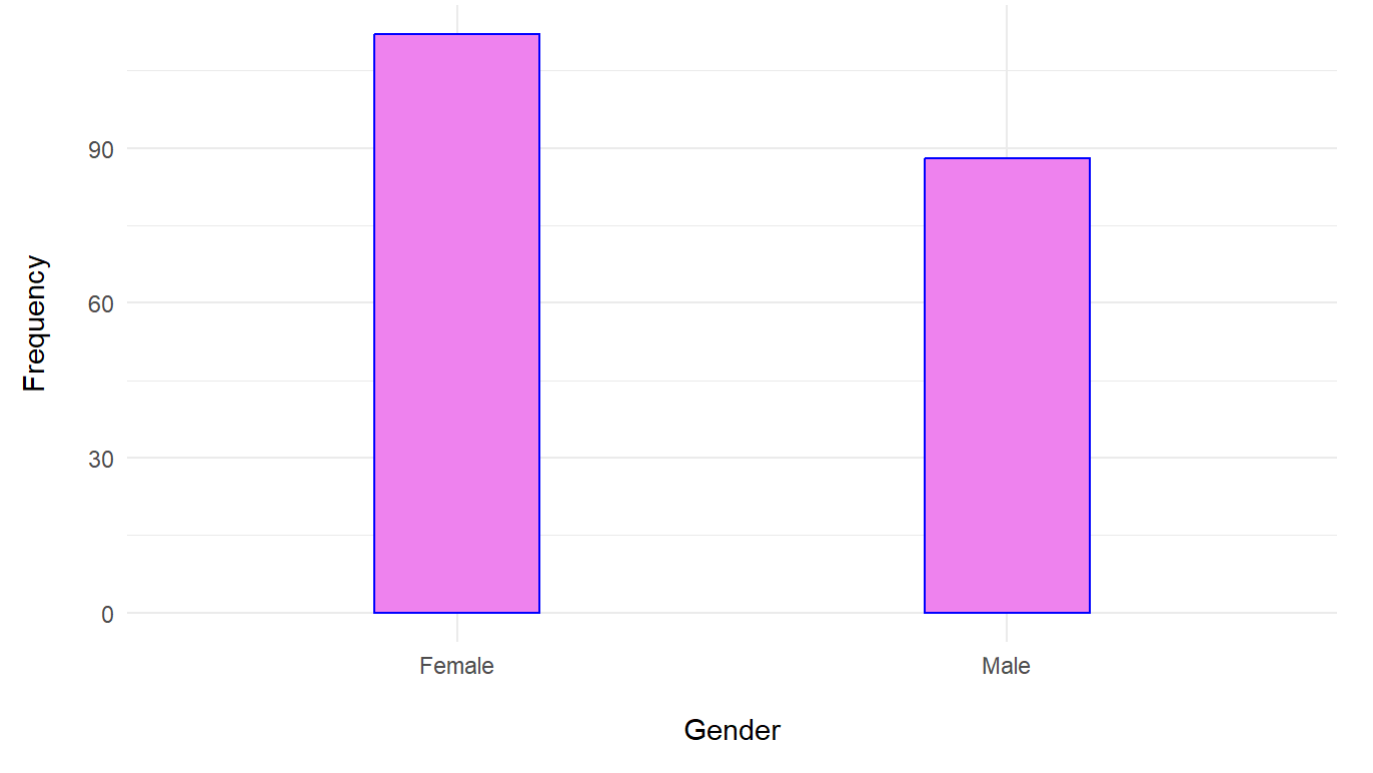
Here our task is to perform cluster analysis and we have two variables, first one is `CustomerID`, second one is `Gender` - IDs of customers is not useful in kmeans and gender is categorical variable, which we can not use in kmeans. So, we will ignore the `CustomerID` variable and will not analyze `Gender` variable too much.

Univariate Analysis

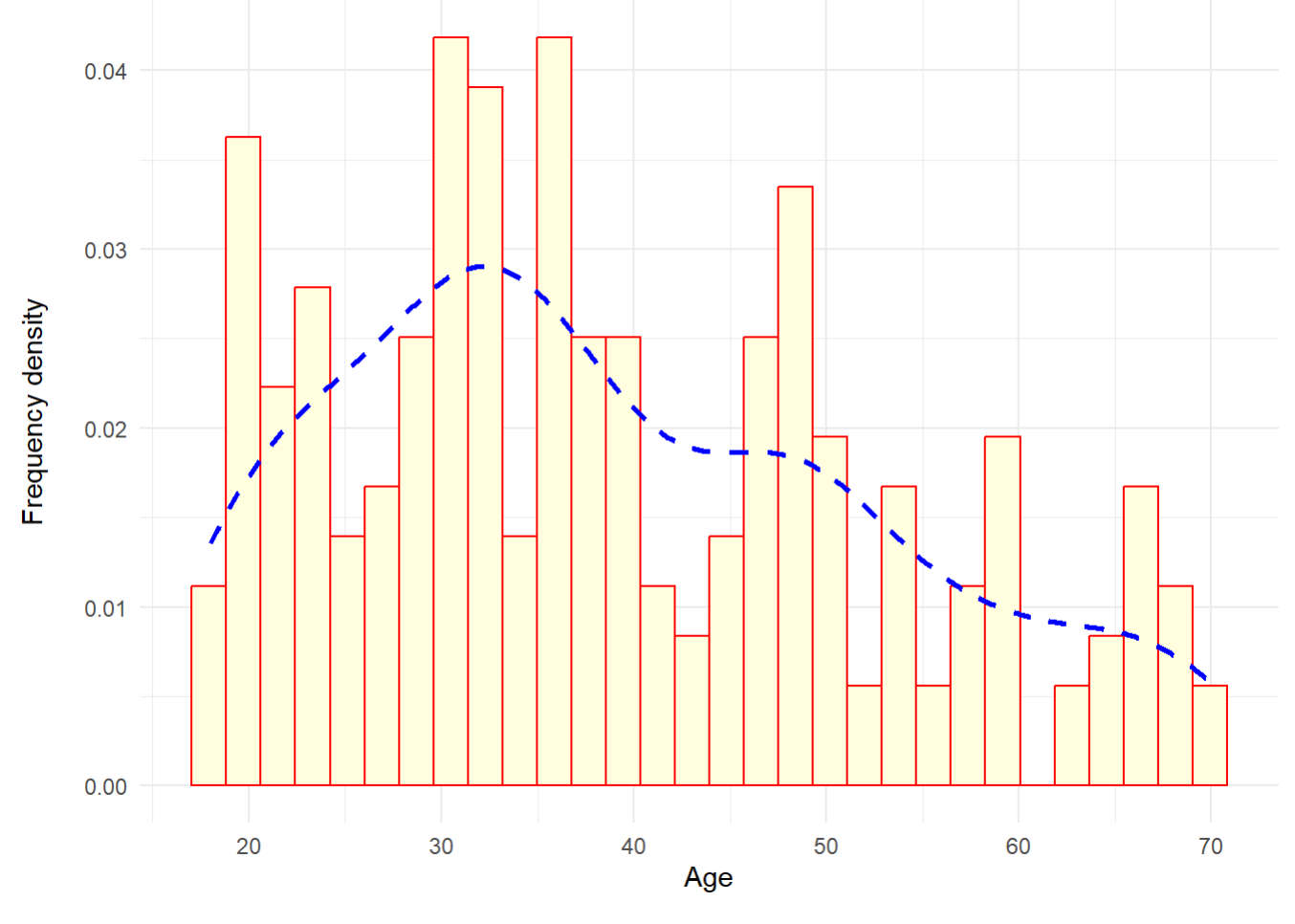
Plots to be used:

- **Categorical:** Count-Plot [Variables: `Gender`]
- **Continuous/Numerical:** Histogram [Variables: `Annual Income (k$)`, `Age`, `Spending Score (1-100)`]

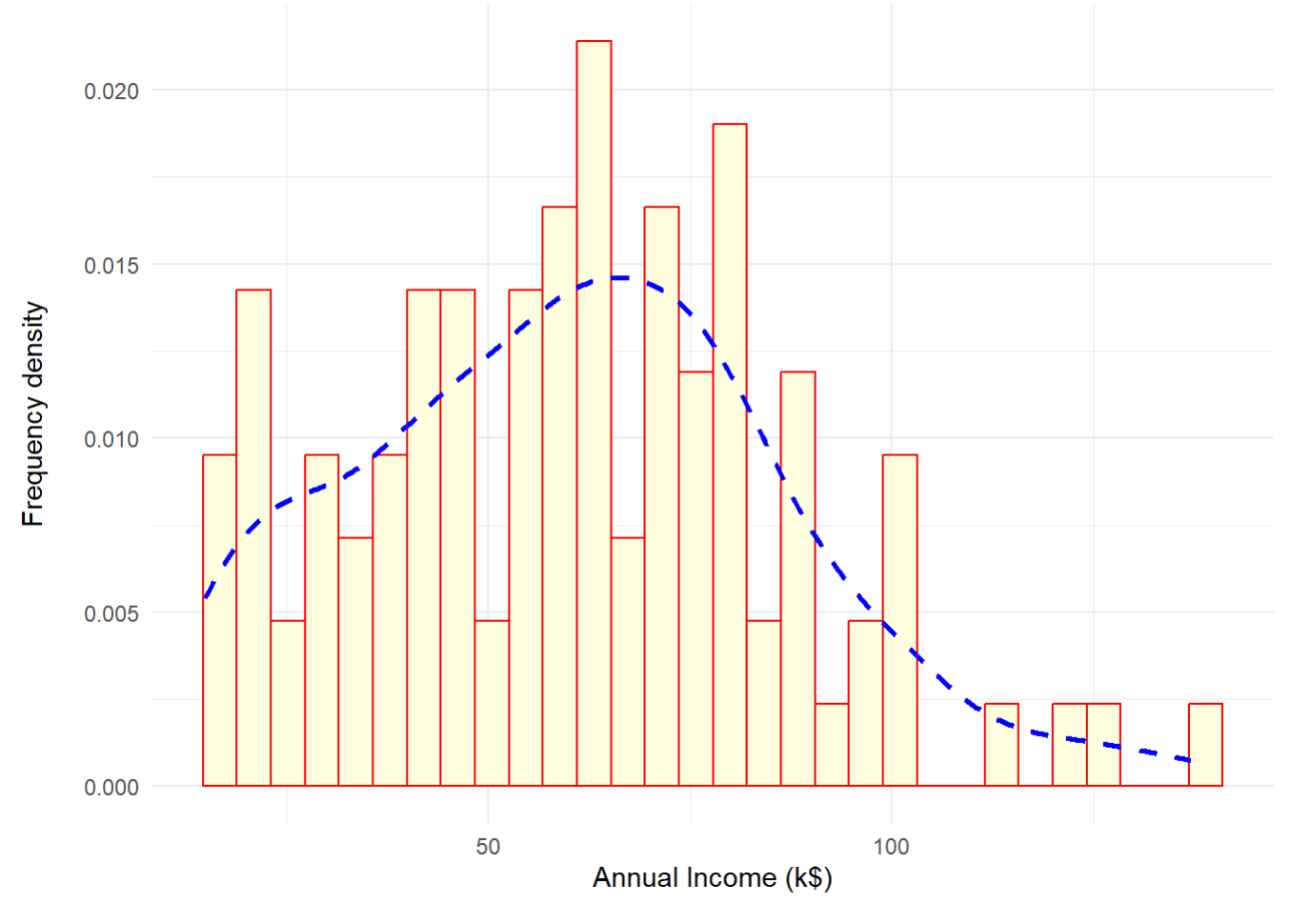
Distribution of Gender



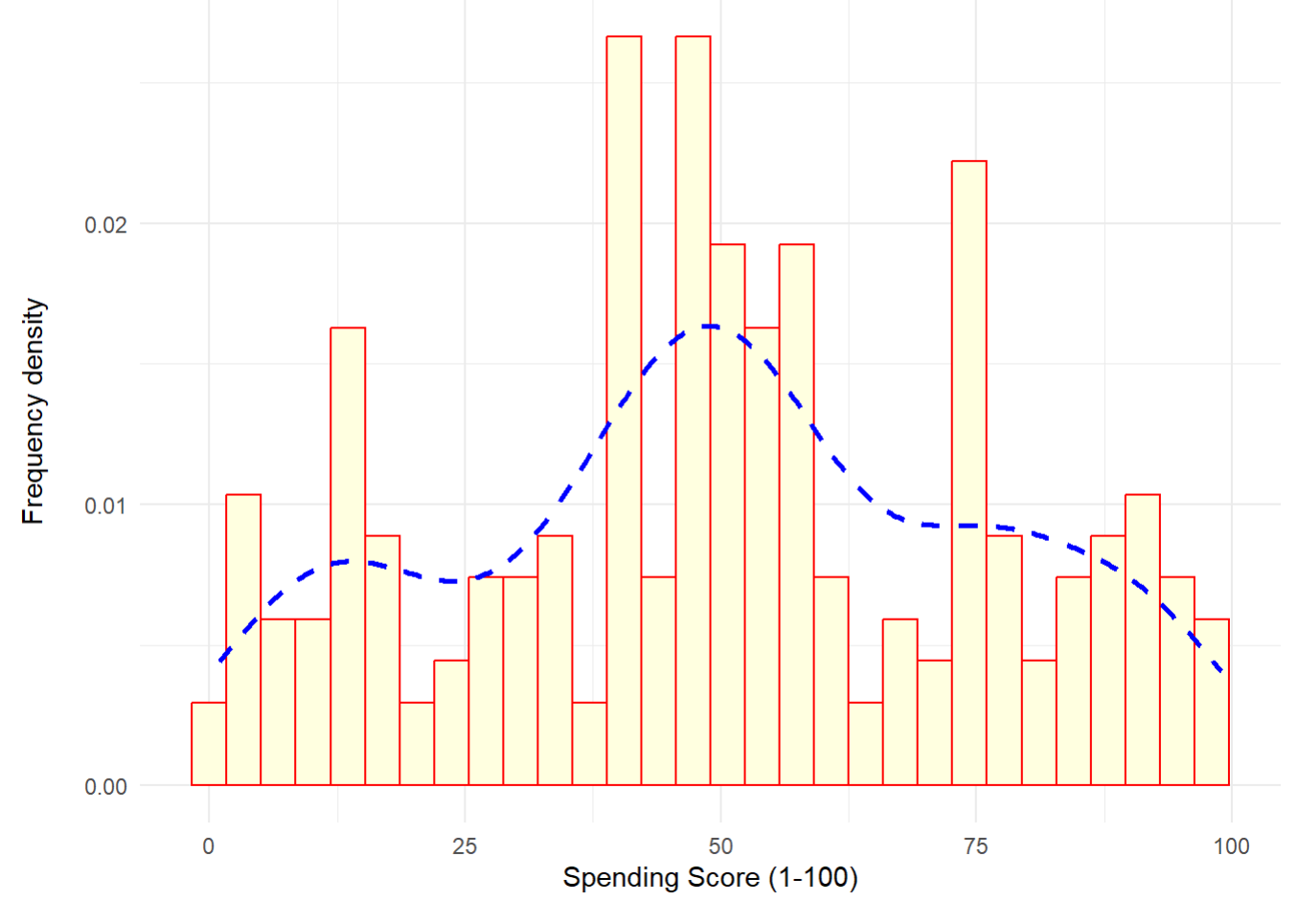
Distribution of Age



Distribution of Annual Income



Distribution of Spending Score

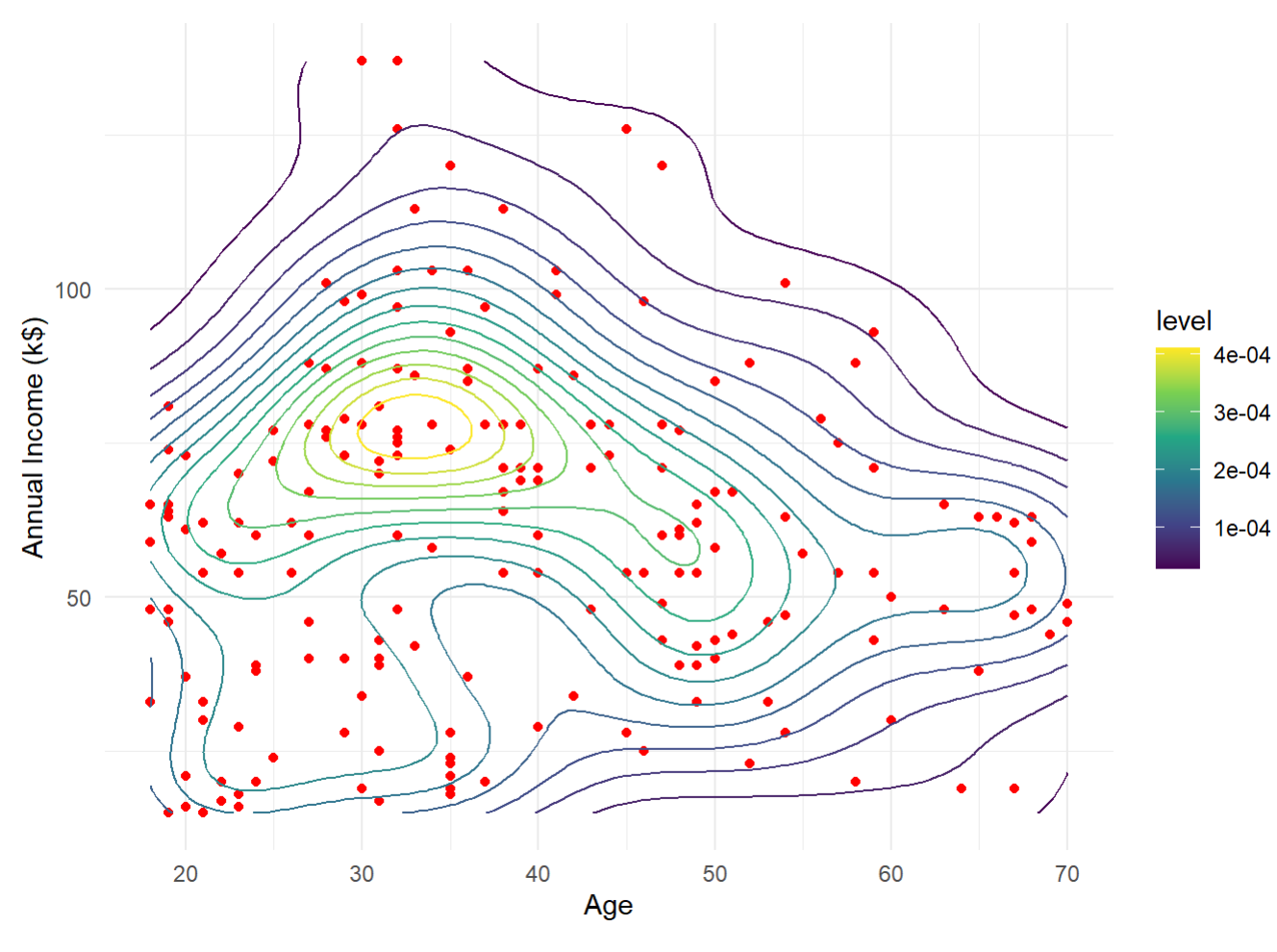


Bivariate Analysis

Plot to be used: Scatter-Plot and Contour-Plot.

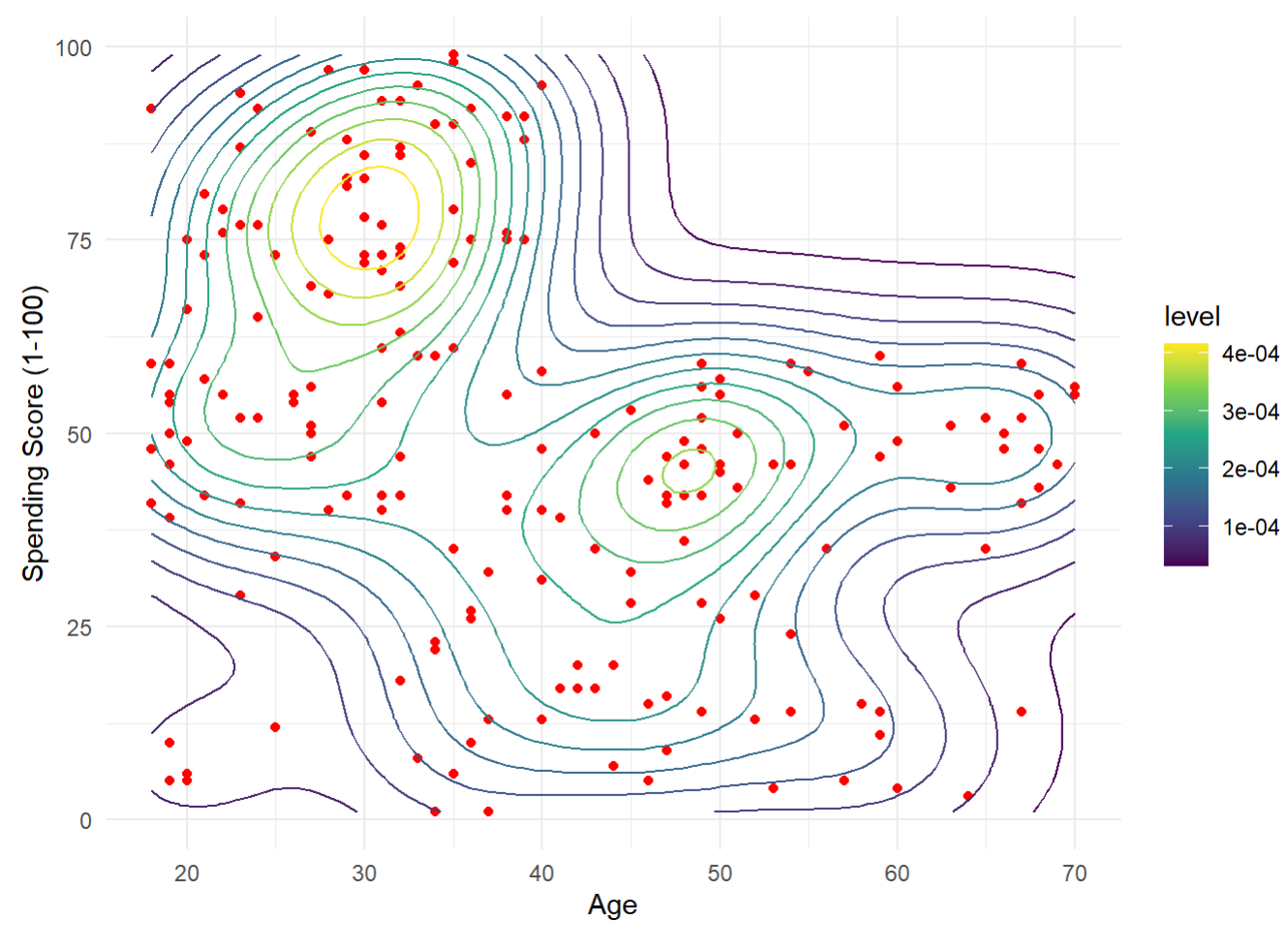
Here we are using contour plot to get an idea about the presence of clusters or peaks in the bivariate distributions of variables taking two at a time from the 3.

Plot 1: Distribution of Age and Annual Income



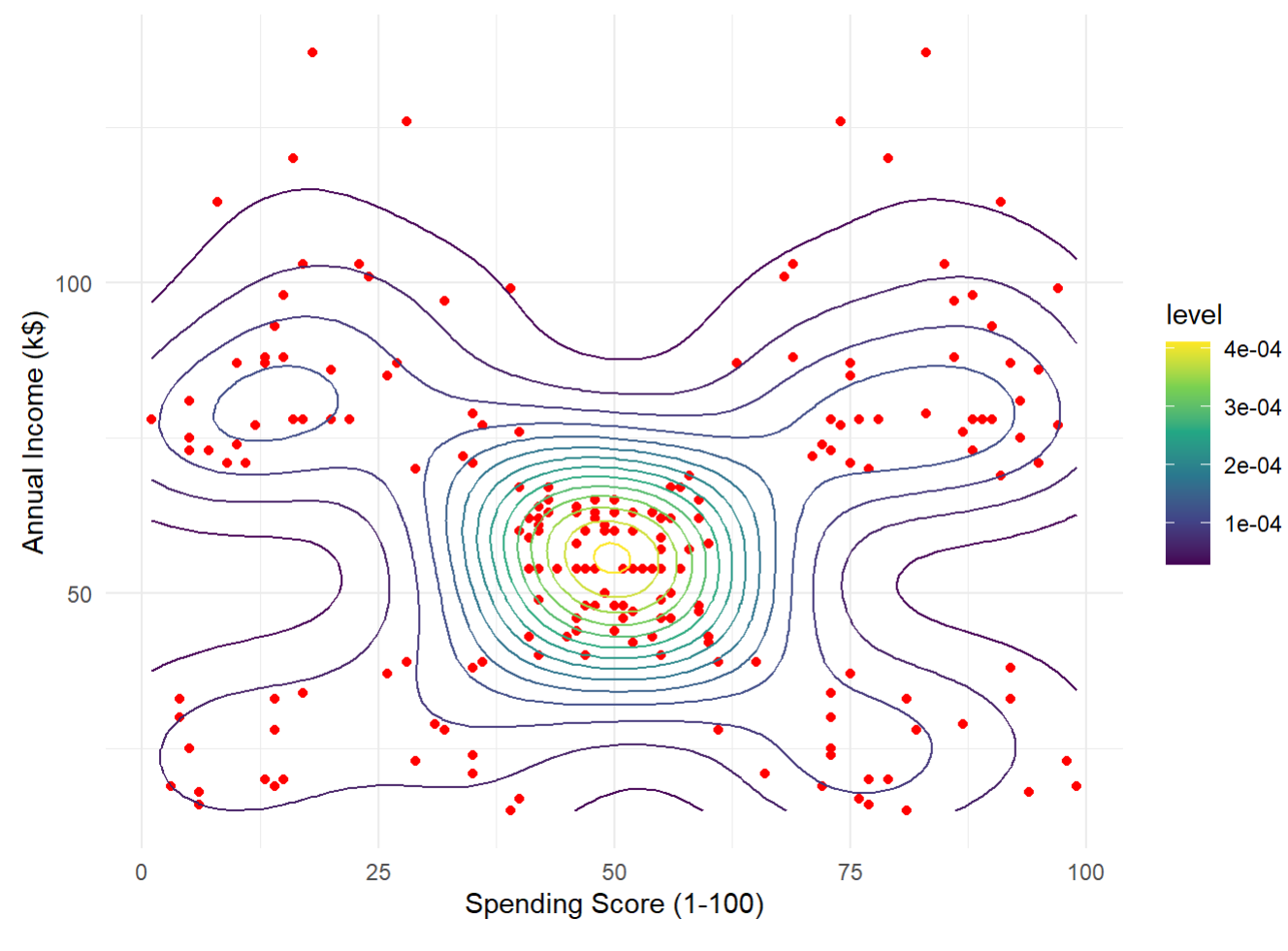
Comment: The plot is not so clear, it is roughly indicating 1 peak.

Plot 2: Distribution of Age and Spending Score



Comment: This plot is more clear than the previous one, two peaks are clear in the joint distribution.

Plot 3: Distribution of Spending Score and Annual Income



Comment: Clear indication of 5 clusters is there.

K-means Clustering

Now that we are done with the EDA, we now proceed with the KMeans clustering. Since Euclidean distance is used here to calculate the distance of the points from the centroids, this algorithm is sensitive to the scale of the data, for that we first scale the data. Also, we have to omit the `CustomerID` and `Gender` from the data (reason mentioned earlier).

- **Scaling method:** StandardScalar

```
df %>% select(-c(CustomerID, Gender)) %>%
  scale() %>% as_tibble() -> df_active
```

First few rows of the Transformed data

Age	Annual Income (k$)	Spending Score (1-100)
-1.4210029	-1.734646	-0.4337131
-1.2778288	-1.734646	1.1927111
-1.3494159	-1.696572	-1.7116178
-1.1346547	-1.696572	1.0378135
-0.5619583	-1.658499	-0.3949887

- Note that, `df_active` will be used for the clustering.

Optimum number of Clusters

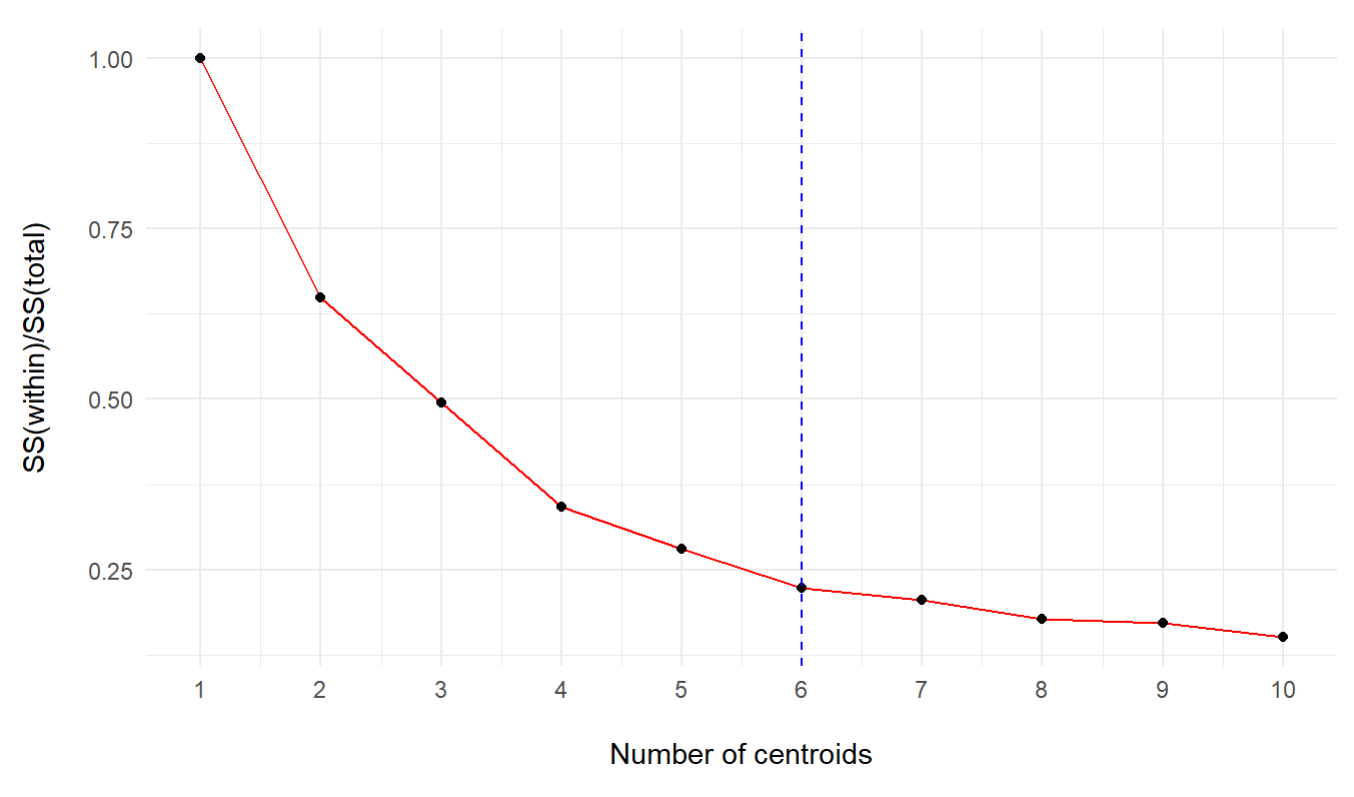
Now, we have to find the optimum number of clusters which suits the data. We will consider different number of clusters and for that number of clusters after which the *within sum of squares* of the clusters will not decrease significantly, will be considered as the optimum number of clusters.

- Consider the following code for the implementation of the above task

```
centers <- 1:10
r <- 0

for(i in 1:length(centers)){
  set.seed(42)
  k <- kmeans(df_active, centers[i])
  r[i] <- k$tot.withinss/k$totss
}

ggplot(NULL, aes(x = centers, y = r)) +
  geom_line(colour = 'red') + geom_point() +
  geom_vline(xintercept = 6, lty = 2, colour = 'blue') +
  theme_minimal() +
  labs(x = '\nNumber of centroids', y = 'SS(within)/SS(total)\n') +
  scale_x_continuous(n.breaks = 10)
```



Comment: So the optimum number of clusters we consider is 6. So, finally we go with 6 clusters and here $\frac{\text{Between Sum of Square}}{\text{Total Sum of Square}} = 0.78$

Final Grouping

```
set.seed(42)

K <- kmeans(df_active, centers = 6)
df %>% mutate('Clusters' = K$cluster) -> df
```

The **centroids** of the clusters are listed below:

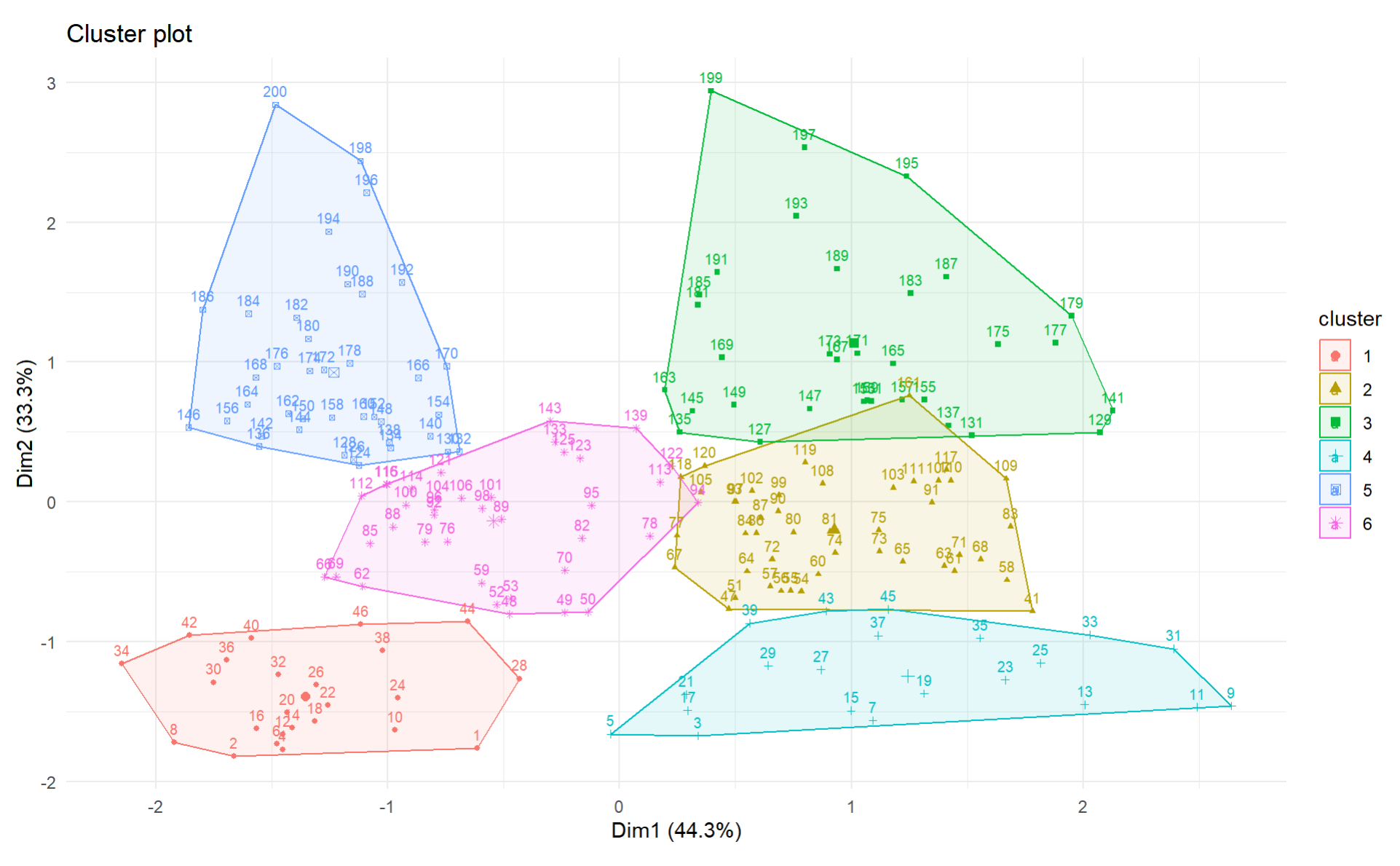
Cluster centroids

Age	Annual Income (k$)	Spending Score (1-100)
-0.9735839	-1.3221791	1.0345865
1.2515802	-0.2396117	-0.0438876
0.2211606	1.0805138	-1.2868231
0.4777583	-1.3049552	-1.1934487
-0.4408110	0.9891010	1.2364001
-0.8709130	-0.1135003	-0.0933461

Cluster Visualization

The clusters along with the customer IDs are shown below, from this one can get an idea about the status of the customers belonging to different groups/clusters.

```
fviz_cluster(K, data = df_active,
             pointsize = 1, labelsize = 8,
             ellipse.alpha = 0.1) +
  theme_minimal()
```



Thank You!