# Credit card dataset for clustering

Fatima Muhammed CST/19/COM/00318

2025-03-24

## Introduction

The sample Dataset summarizes the usage behavior of about 9000 active credit card holders during the last 6 months. The file is at a customer level with 18 behavioral variables. You need to develop a customer segmentation to define marketing strategy from the dataset.

## Data Description

Following is the Data Dictionary for Credit Card dataset: * CUST_ID: Identification of Credit Card holder (Categorical) * BALANCE: Balance amount left in their account to make purchases * BALANCE_FREQUENCY: How frequently the Balance is updated, score between 0 and 1 (1 = frequently updated, 0 = not frequently updated) * PURCHASES: Amount of purchases made from account * ONEOFF_PURCHASES: Maximum purchase amount done in one-go * INSTALLMENTS_PURCHASES: Amount of purchase done in installment * CASH_ADVANCE: Cash in advance given by the user * PURCHASES_FREQUENCY: How frequently the Purchases are being made, score between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased) * ONEOFFPURCHASESFREQUENCY: How frequently Purchases are happening in one-go (1 = frequently purchased, 0 = not frequently purchased) * PURCHASESINSTALLMENTSFREQUENCY: How frequently purchases in installments are being done (1 = frequently done, 0 = not frequently done) * CASHADVANCEFREQUENCY: How frequently the cash in advance being paid * CASHADVANCETRX: Number of Transactions made with "Cash in Advanced" * PURCHASES_TRX: Number of purchase transactions made * CREDIT_LIMIT: Limit of Credit Card for user * PAYMENTS: Amount of Payment done by user * MINIMUM_PAYMENTS: Minimum amount of payments made by user * PRCFULLPAYMENT: Percent of full payment paid by user * TENURE: Tenure of credit card service for user

## Load Data

```
# Load the data
data <- read.csv('CC GENERAL.csv')
```

### Data overview

```
cat('Data shape: ', dim(data), '\n')
```

```
## Data shape:  8950 18
```

```
head(data)
```

```
##   CUST_ID    BALANCE BALANCE_FREQUENCY PURCHASES ONEOFF_PURCHASES
## 1  C10001   40.90075          0.818182     95.40             0.00
## 2  C10002 3202.46742          0.909091      0.00             0.00
## 3  C10003 2495.14886          1.000000    773.17           773.17
## 4  C10004 1666.67054          0.636364   1499.00          1499.00
## 5  C10005  817.71434          1.000000     16.00            16.00
## 6  C10006 1809.82875          1.000000   1333.28             0.00
```

```
##   INSTALLMENTS_PURCHASES CASH_ADVANCE PURCHASES_FREQUENCY
## 1                  95.40        0.000            0.166667
## 2                   0.00     6442.945            0.000000
## 3                   0.00        0.000            1.000000
## 4                   0.00      205.788            0.083333
## 5                   0.00        0.000            0.083333
## 6                1333.28        0.000            0.666667
##   ONEOFF_PURCHASES_FREQUENCY PURCHASES_INSTALLMENTS_FREQUENCY
## 1                   0.000000                         0.083333
## 2                   0.000000                         0.000000
## 3                   1.000000                         0.000000
## 4                   0.083333                         0.000000
## 5                   0.083333                         0.000000
## 6                   0.000000                         0.583333
##   CASH_ADVANCE_FREQUENCY CASH_ADVANCE_TRX PURCHASES_TRX CREDIT_LIMIT  PAYMENTS
## 1               0.000000                0             2         1000  201.8021
## 2               0.250000                4             0         7000 4103.0326
## 3               0.000000                0            12         7500  622.0667
## 4               0.083333                1             1         7500    0.0000
## 5               0.000000                0             1         1200  678.3348
## 6               0.000000                0             8         1800 1400.0578
##   MINIMUM_PAYMENTS PRC_FULL_PAYMENT TENURE
## 1         139.5098         0.000000     12
## 2        1072.3402         0.222222     12
## 3         627.2848         0.000000     12
## 4               NA         0.000000     12
## 5         244.7912         0.000000     12
## 6        2407.2460         0.000000     12
```

```
library(psych)
describe(data)
```

```
##                                  vars    n    mean      sd  median trimmed
## CUST_ID*                            1 8950 4475.50 2583.79 4475.50 4475.50
## BALANCE                             2 8950 1564.47 2081.53  873.39 1128.17
## BALANCE_FREQUENCY                   3 8950    0.88    0.24    1.00    0.94
## PURCHASES                           4 8950 1003.20 2136.63  361.28  583.10
## ONEOFF_PURCHASES                    5 8950  592.44 1659.89   38.00  263.82
## INSTALLMENTS_PURCHASES              6 8950  411.07  904.34   89.00  223.03
## CASH_ADVANCE                        7 8950  978.87 2097.16    0.00  494.90
## PURCHASES_FREQUENCY                 8 8950    0.49    0.40    0.50    0.49
## ONEOFF_PURCHASES_FREQUENCY          9 8950    0.20    0.30    0.08    0.14
## PURCHASES_INSTALLMENTS_FREQUENCY   10 8950    0.36    0.40    0.17    0.33
## CASH_ADVANCE_FREQUENCY             11 8950    0.14    0.20    0.00    0.09
## CASH_ADVANCE_TRX                   12 8950    3.25    6.82    0.00    1.73
## PURCHASES_TRX                      13 8950   14.71   24.86    7.00    9.43
## CREDIT_LIMIT                       14 8949 4494.45 3638.82 3000.00 3927.53
## PAYMENTS                           15 8950 1733.14 2895.06  856.90 1152.89
## MINIMUM_PAYMENTS                   16 8637  864.21 2372.45  312.34  484.54
## PRC_FULL_PAYMENT                   17 8950    0.15    0.29    0.00    0.08
## TENURE                             18 8950   11.52    1.34   12.00   11.92
##                                     mad   min      max    range  skew kurtosis
## CUST_ID*                        3317.32  1.00  8950.00  8949.00  0.00    -1.20
## BALANCE                         1185.88  0.00 19043.14 19043.14  2.39     7.67
## BALANCE_FREQUENCY                  0.00  0.00     1.00     1.00 -2.02     3.09
```

```
## PURCHASES                          535.63  0.00 49039.57 49039.57  8.14   111.30
## ONEOFF_PURCHASES                    56.34  0.00 40761.25 40761.25 10.04   164.06
## INSTALLMENTS_PURCHASES             131.95  0.00 22500.00 22500.00  7.30    96.50
## CASH_ADVANCE                         0.00  0.00 47137.21 47137.21  5.16    52.86
## PURCHASES_FREQUENCY                  0.62  0.00     1.00     1.00  0.06    -1.64
## ONEOFF_PURCHASES_FREQUENCY           0.12  0.00     1.00     1.00  1.54     1.16
## PURCHASES_INSTALLMENTS_FREQUENCY     0.25  0.00     1.00     1.00  0.51    -1.40
## CASH_ADVANCE_FREQUENCY               0.00  0.00     1.50     1.50  1.83     3.33
## CASH_ADVANCE_TRX                     0.00  0.00   123.00   123.00  5.72    61.60
## PURCHASES_TRX                       10.38  0.00   358.00   358.00  4.63    34.76
## CREDIT_LIMIT                      2668.68 50.00 30000.00 29950.00  1.52     2.83
## PAYMENTS                           861.91  0.00 50721.48 50721.48  5.91    54.73
## MINIMUM_PAYMENTS                   282.25  0.02 76406.21 76406.19 13.62   283.76
## PRC_FULL_PAYMENT                     0.00  0.00     1.00     1.00  1.94     2.43
## TENURE                               0.00  6.00    12.00     6.00 -2.94     7.69
##                                        se
## CUST_ID*                            27.31
## BALANCE                             22.00
## BALANCE_FREQUENCY                    0.00
## PURCHASES                           22.58
## ONEOFF_PURCHASES                    17.55
## INSTALLMENTS_PURCHASES               9.56
## CASH_ADVANCE                        22.17
## PURCHASES_FREQUENCY                  0.00
## ONEOFF_PURCHASES_FREQUENCY           0.00
## PURCHASES_INSTALLMENTS_FREQUENCY     0.00
## CASH_ADVANCE_FREQUENCY               0.00
## CASH_ADVANCE_TRX                     0.07
## PURCHASES_TRX                        0.26
## CREDIT_LIMIT                        38.47
## PAYMENTS                            30.60
## MINIMUM_PAYMENTS                    25.53
## PRC_FULL_PAYMENT                     0.00
## TENURE                               0.01
```

## Data Cleaning

**First, we check the missing/corrupted values.**

```
# Count missing values in each column
colSums(is.na(data))
```

```
##                          CUST_ID                          BALANCE
##                                0                                0
##                  BALANCE_FREQUENCY                        PURCHASES
##                                0                                0
##                   ONEOFF_PURCHASES             INSTALLMENTS_PURCHASES
##                                0                                0
##                      CASH_ADVANCE               PURCHASES_FREQUENCY
##                                0                                0
##       ONEOFF_PURCHASES_FREQUENCY PURCHASES_INSTALLMENTS_FREQUENCY
##                                0                                0
##            CASH_ADVANCE_FREQUENCY                  CASH_ADVANCE_TRX
##                                0                                0
##                     PURCHASES_TRX                     CREDIT_LIMIT
```

```
##                                0                                1
##                         PAYMENTS                 MINIMUM_PAYMENTS
##                                0                              313
##                  PRC_FULL_PAYMENT                           TENURE
##                                0                                0
```

We will impute these missing values with the median value.

```
# Impute missing values with median
data$MINIMUM_PAYMENTS[is.na(data$MINIMUM_PAYMENTS)] <- median(data$MINIMUM_PAYMENTS, na.rm = TRUE)
data$CREDIT_LIMIT[is.na(data$CREDIT_LIMIT)] <- median(data$CREDIT_LIMIT, na.rm = TRUE)

colSums(is.na(data))
```

```
##                            CUST_ID                            BALANCE
##                                  0                                  0
##                  BALANCE_FREQUENCY                          PURCHASES
##                                  0                                  0
##                   ONEOFF_PURCHASES             INSTALLMENTS_PURCHASES
##                                  0                                  0
##                       CASH_ADVANCE                PURCHASES_FREQUENCY
##                                  0                                  0
##       ONEOFF_PURCHASES_FREQUENCY PURCHASES_INSTALLMENTS_FREQUENCY
##                                  0                                  0
##            CASH_ADVANCE_FREQUENCY                  CASH_ADVANCE_TRX
##                                  0                                  0
##                      PURCHASES_TRX                      CREDIT_LIMIT
##                                  0                                  0
##                           PAYMENTS                 MINIMUM_PAYMENTS
##                                  0                                  0
##                  PRC_FULL_PAYMENT                           TENURE
##                                  0                                  0
```

No more missing values

**Now we drop CUST_ID column, then normalize the input values using StandardScaler().**

```
# Drop ID column
data <- data[, !names(data) %in% 'CUST_ID']

# Normalize values
library(scales)
data_scaled <- scale(data)

# Replace data with scaled data
data <- as.data.frame(data_scaled)

# Check shape
dim(data)
```

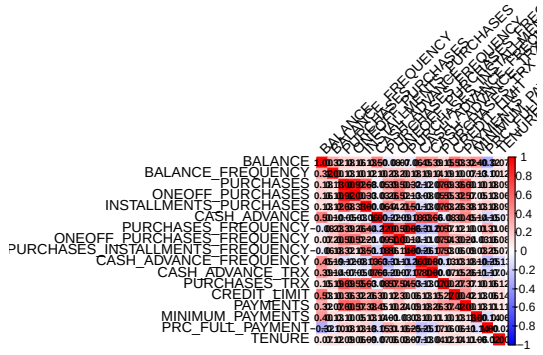
```
## [1] 8950   17
```

## Clustering

### Correlation Check

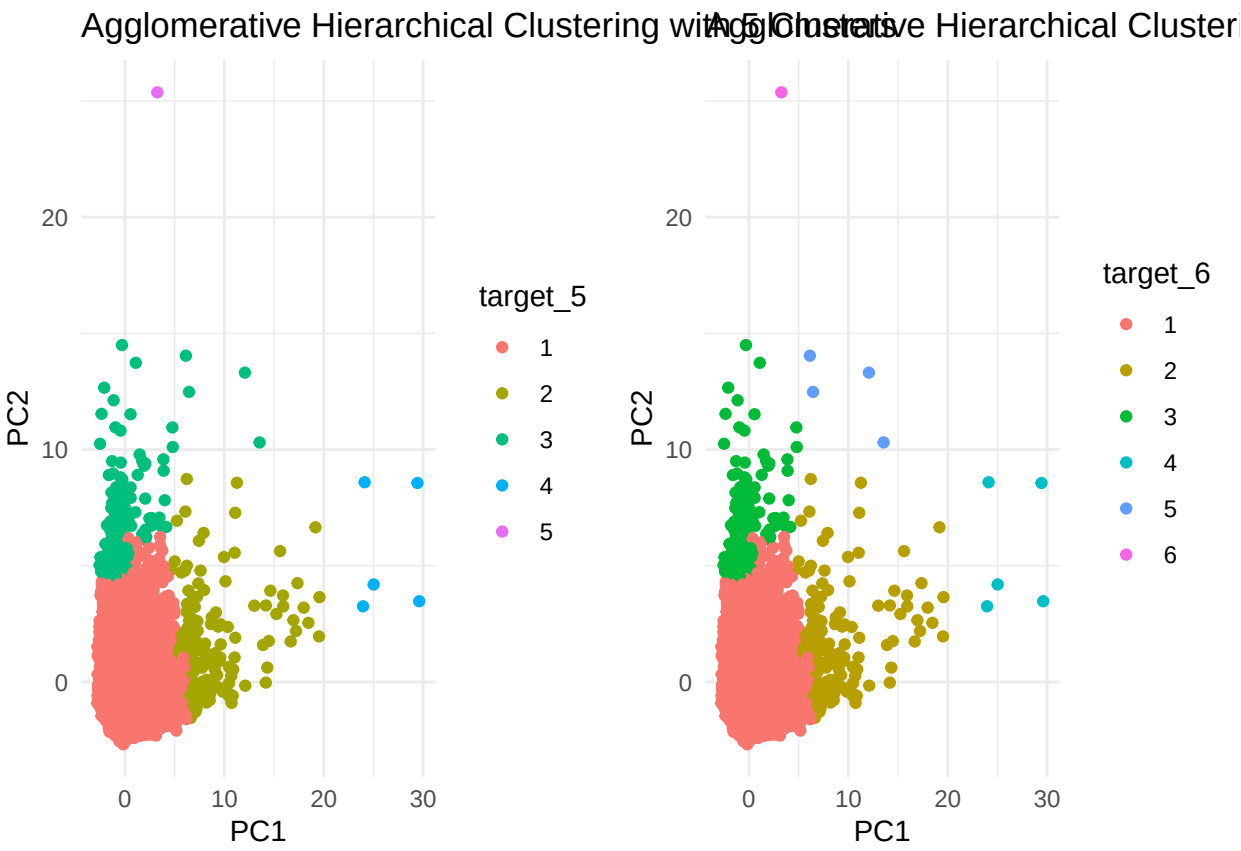
```
library(ggplot2)
library(corrplot)
```

```r
# Compute correlation matrix
cor_matrix <- cor(data, use = "complete.obs")

# Plot heatmap
corrplot(cor_matrix, method = "color", col = colorRampPalette(c("blue", "white", "red"))(200),
         tl.col = "black", tl.srt = 45, addCoef.col = "black", number.cex = 0.7, mar = c(1, 1, 1, 1))
```


BALANCE
BALANCE_FREQUENCY
PURCHASES
ONEOFF_PURCHASES
INSTALLMENTS_PURCHASES
CASH_ADVANCE
PURCHASES_FREQUENCY
ONEOFF_PURCHASES_FREQUENCY
PURCHASES_INSTALLMENTS_FREQUENCY
CASH_ADVANCE_FREQUENCY
CASH_ADVANCE_TRX
PURCHASES_TRX
CREDIT_LIMIT
PAYMENTS
MINIMUM_PAYMENTS
PRC_FULL_PAYMENT
TENURE
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

**Agglomerative Hierarchical Clustering with PCA**

```r
# Load necessary libraries
library(FactoMineR)
library(ggplot2)
library(dplyr)

# Perform PCA
pca_result <- PCA(data, scale.unit = TRUE, ncp = 2, graph = FALSE)
pca_data <- as.data.frame(pca_result$ind$coord)

# Agglomerative Clustering with 5 Clusters
hc_5 <- hclust(dist(pca_data))
clusters_5 <- cutree(hc_5, k = 5)
pca_data$target_5 <- as.factor(clusters_5)

# Agglomerative Clustering with 6 Clusters
hc_6 <- hclust(dist(pca_data))
clusters_6 <- cutree(hc_6, k = 6)
pca_data$target_6 <- as.factor(clusters_6)

# Plot for 5 Clusters
plot_5 <- ggplot(pca_data, aes(x = Dim.1, y = Dim.2, color = target_5)) +
  geom_point() +
  labs(title = 'Agglomerative Hierarchical Clustering with 5 Clusters',
       x = 'PC1', y = 'PC2') +
  theme_minimal()

# Plot for 6 Clusters
plot_6 <- ggplot(pca_data, aes(x = Dim.1, y = Dim.2, color = target_6)) +
  geom_point() +
  labs(title = 'Agglomerative Hierarchical Clustering with 6 Clusters',
       x = 'PC1', y = 'PC2') +
  theme_minimal()

# Display plots side by side
library(gridExtra)
grid.arrange(plot_5, plot_6, ncol = 2)
```



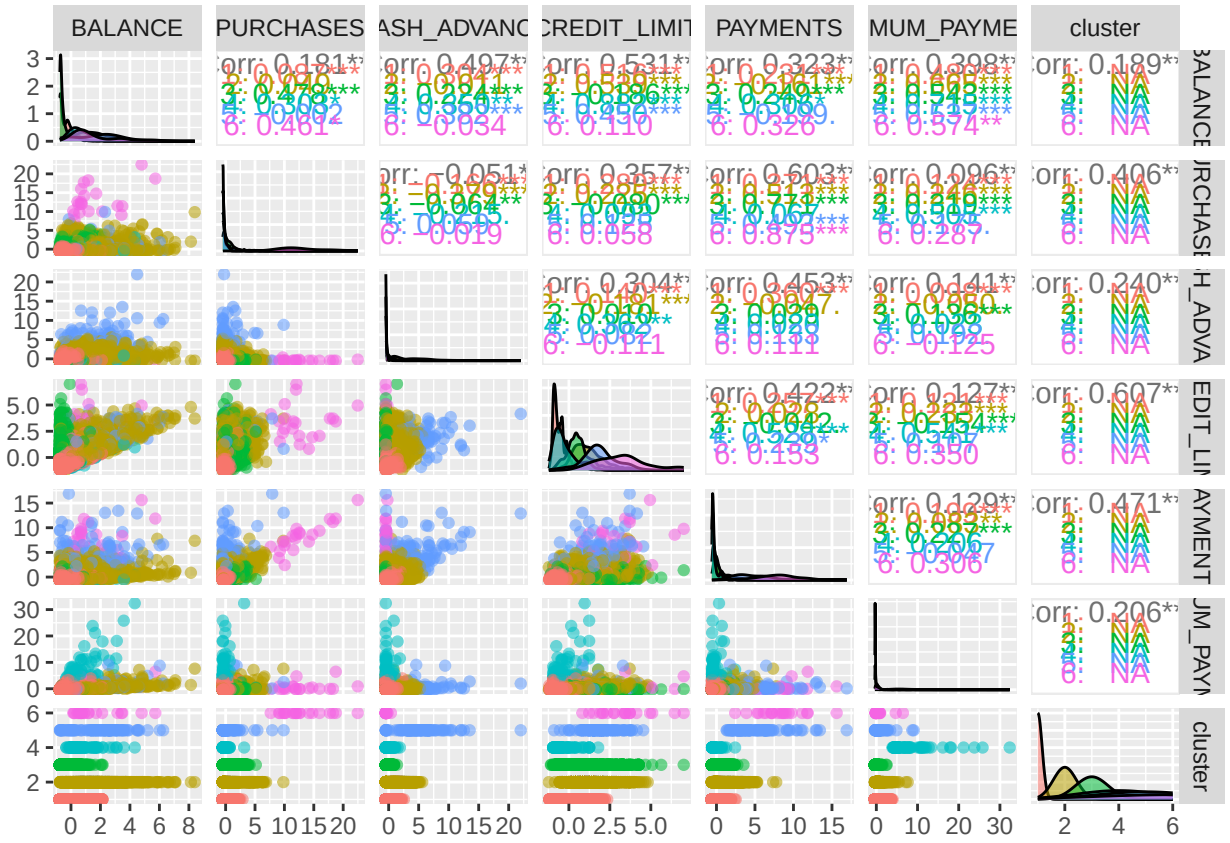

## Exploratory Data Analysis

**We are picking 6 clusters for this EDA. Let's make a Seaborn pairplot with selected/best columns to show how the clusters are segmenting the samples:**

```
# Select best columns
best_cols <- c("BALANCE", "PURCHASES", "CASH_ADVANCE", "CREDIT_LIMIT", "PAYMENTS", "MINIMUM_PAYMENTS")

# Dataframe with best columns
data_final <- data[, best_cols]

cat("New dataframe with best columns has just been created. Data shape:", dim(data_final), "\n")
```

```
## New dataframe with best columns has just been created. Data shape: 8950 6
```

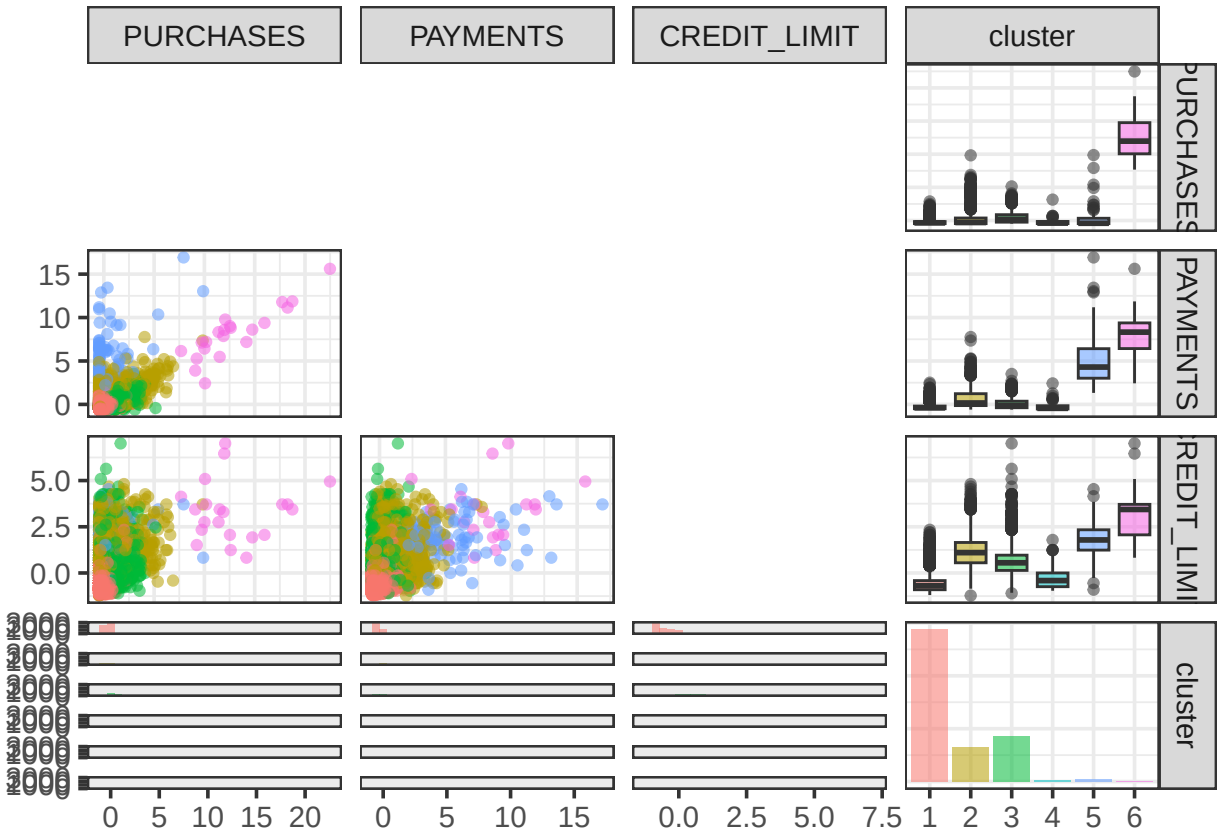
```
# Apply hierarchical clustering (Agglomerative)
d <- dist(data_final)  # Calculate distance matrix
hc <- hclust(d, method = "ward.D2")  # Agglomerative clustering
data_final$cluster <- cutree(hc, k = 6)  # Cut tree into 6 clusters

# Load GGally for pairplot
library(GGally)

# Create a pairplot with clusters
ggpairs(data_final, aes(color = as.factor(cluster), alpha = 0.5))
```




We can see some interesting correlations between features and clusters that we have made above. Let's get into detailed analysis.

**Cluster 0 (Blue): The Average Joe**

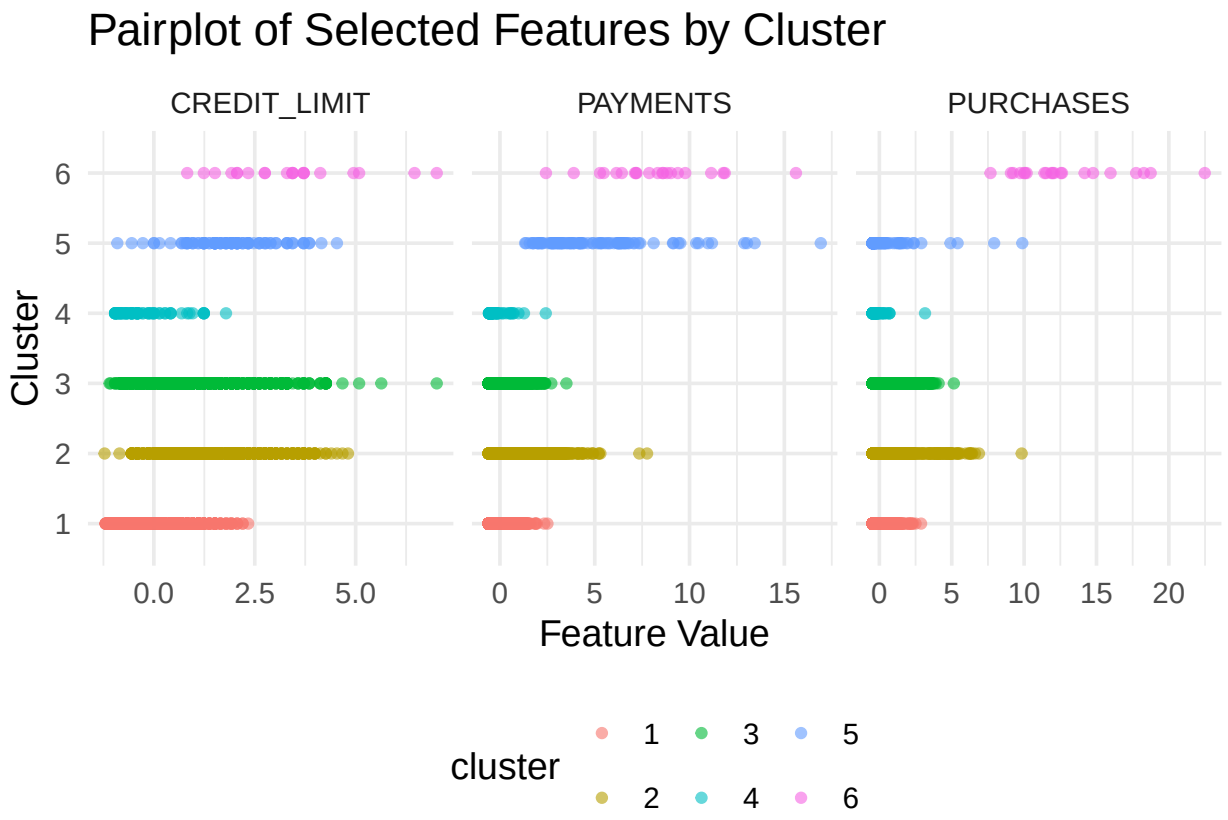
```
# Load libraries
library(GGally)
library(ggplot2)

# Convert cluster to factor for coloring
data_final$cluster <- as.factor(data_final$cluster)

# Create a pairplot with specified x and y variables
ggpairs(data_final,
        columns = c("PURCHASES", "PAYMENTS", "CREDIT_LIMIT", "cluster"),
        aes(color = cluster, alpha = 0.5),
        upper = list(continuous = "blank"),
        diag = list(continuous = "blank"),
        lower = list(continuous = "points")) +
  theme_bw(base_size = 14)
```




This group of users, while having the highest number of users by far, is fairly frugal: they have lowest purchases, second lowest payments, and lowest credit limit. The bank would not make much profit from this group, so there should be some sorts of strategy to attract these people more.

**Cluster 1 (Orange): The Active Users**

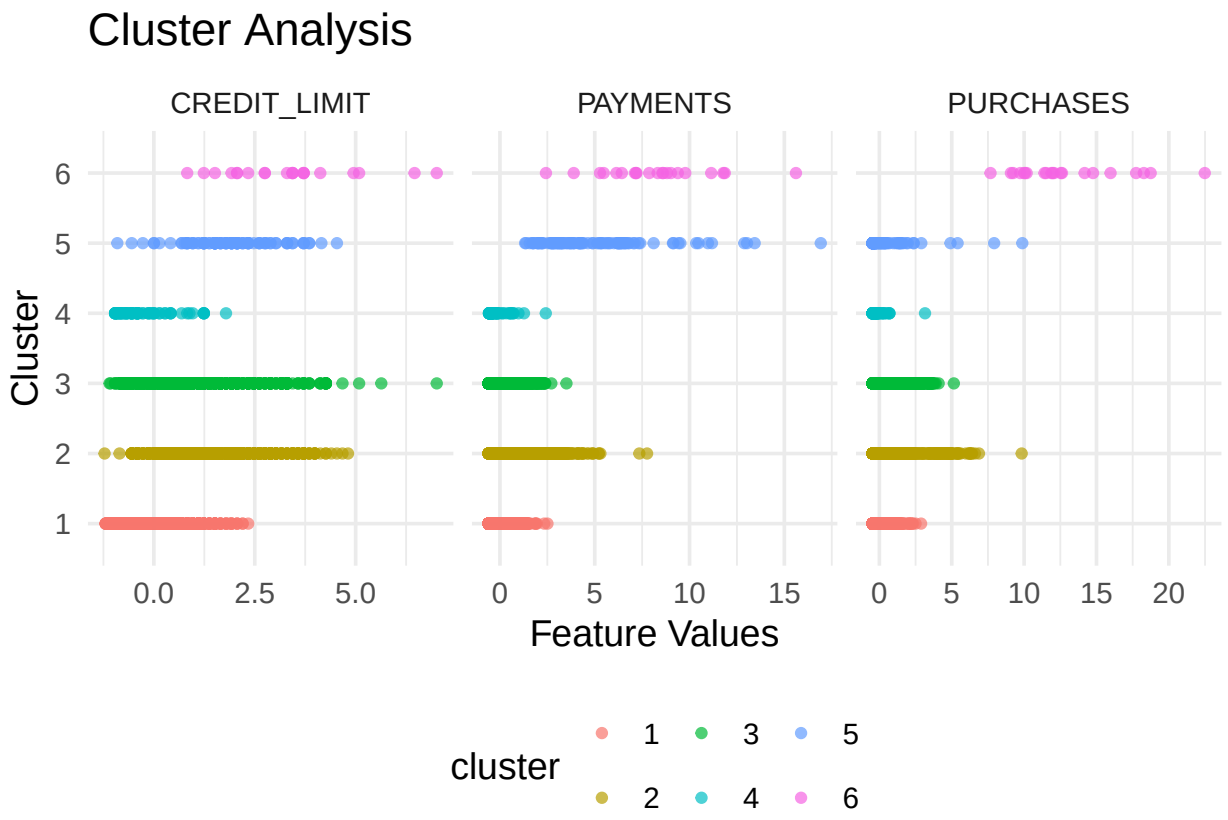
```
# Load libraries
library(ggplot2)
library(tidyr)

# Convert cluster to factor
data_final$cluster <- as.factor(data_final$cluster)

# Pivot data for easier plotting
data_long <- pivot_longer(data_final, cols = c("PURCHASES", "PAYMENTS", "CREDIT_LIMIT"),
                          names_to = "Feature", values_to = "Value")

# Plot
ggplot(data_long, aes(x = Value, y = cluster, color = cluster)) +
  geom_point(alpha = 0.6) +
  facet_wrap(~Feature, scales = "free_x") +
  theme_minimal(base_size = 14) +
  labs(title = "Pairplot of Selected Features by Cluster",
       x = "Feature Value", y = "Cluster") +
  theme(legend.position = "bottom")
```




This group of users is very active in general: they have second highest purchases, third highest payments, and the most varied credit limit values. This type of credit card users is the type you should spend the least time and effort on, as they are already the ideal one.

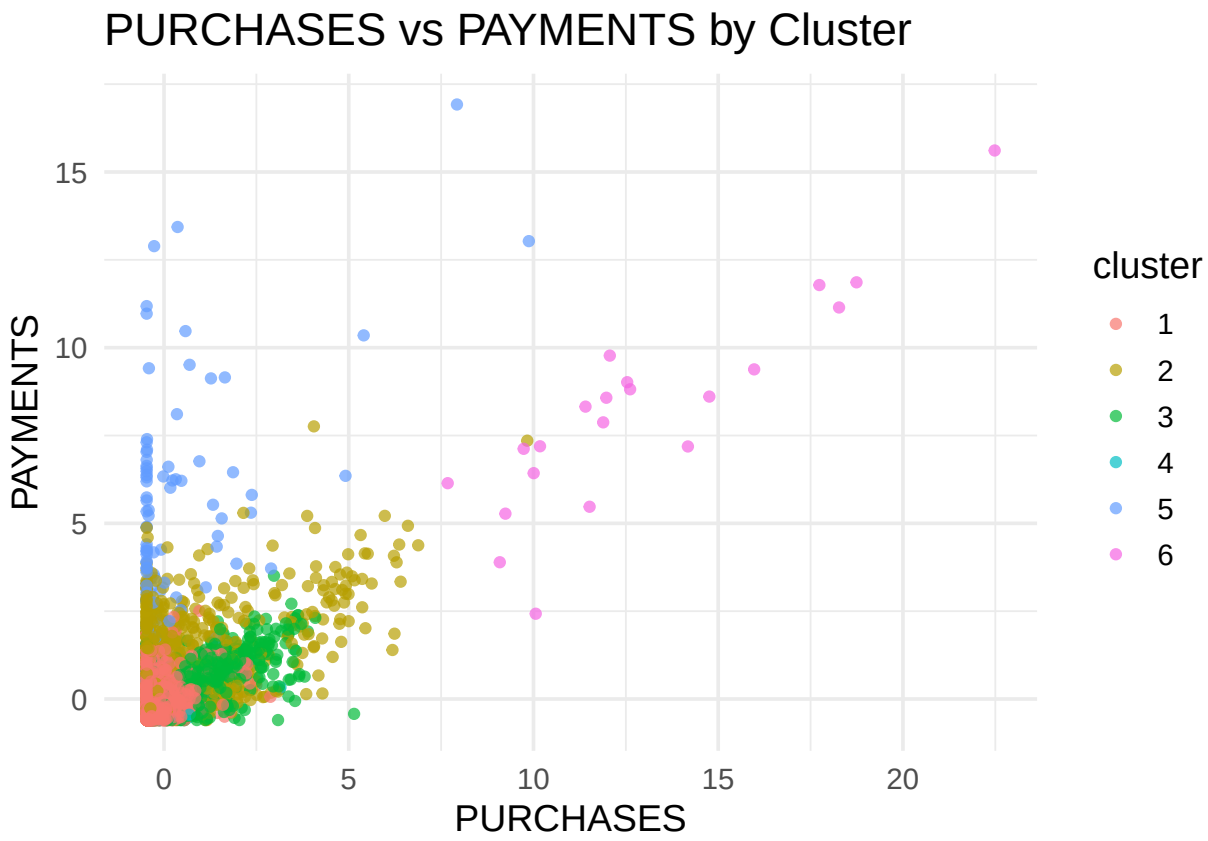
**Cluster 2 (Green): The Big Spenders** The Big Spenders. This group is by far the most interesting to analyze, since they do not only have the highest number of purchases, highest payments, highest minimum payments, but the other features are also wildly varied in values. Let's take a quick look at the pairplots.

```r
# Load necessary libraries
library(ggplot2)
library(tidyr)

# Convert 'cluster' to a factor for proper coloring
data_final$cluster <- as.factor(data_final$cluster)

# Select relevant columns and reshape data for plotting
plot_data <- data_final[, c("PURCHASES", "PAYMENTS", "CREDIT_LIMIT", "cluster")]
plot_data_long <- pivot_longer(plot_data, cols = c("PURCHASES", "PAYMENTS", "CREDIT_LIMIT"),
                               names_to = "Feature", values_to = "Value")

# Plot using ggplot
ggplot(plot_data_long, aes(x = Value, y = cluster, color = cluster)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~ Feature, scales = "free_x") +
  theme_minimal(base_size = 14) +
  labs(title = "Cluster Analysis", x = "Feature Values", y = "Cluster") +
  theme(legend.position = "bottom")
```



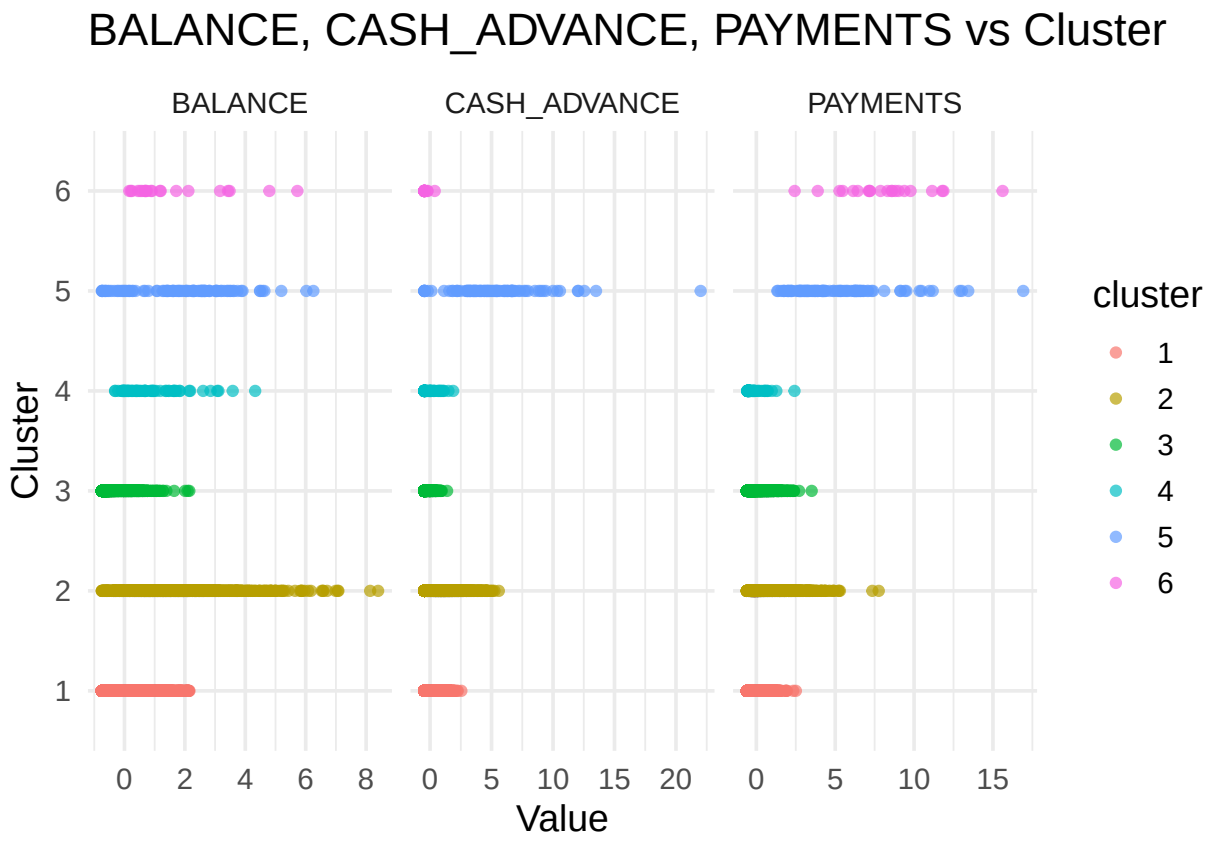

As a nature of the "Big Spenders", there are many outliers in this cluster: people who have/make abnormally high balance, purchases, cash advance, and payment. The graph below will give you an impression of how outlier-heavy this cluster is - almost all the green dots are outliers relatively compared to the rest of the whole dataset.

```r
# Load necessary libraries
library(ggplot2)

# Convert 'cluster' to a factor for proper coloring
data_final$cluster <- as.factor(data_final$cluster)

# Plot PURCHASES vs PAYMENTS colored by cluster
ggplot(data_final, aes(x = PURCHASES, y = PAYMENTS, color = cluster)) +
  geom_point(alpha = 0.7) +
  theme_minimal(base_size = 14) +
  labs(title = "PURCHASES vs PAYMENTS by Cluster", x = "PURCHASES", y = "PAYMENTS") +
  theme(legend.position = "right")
```




**Cluster 3 (Red): The Money Borrowers**

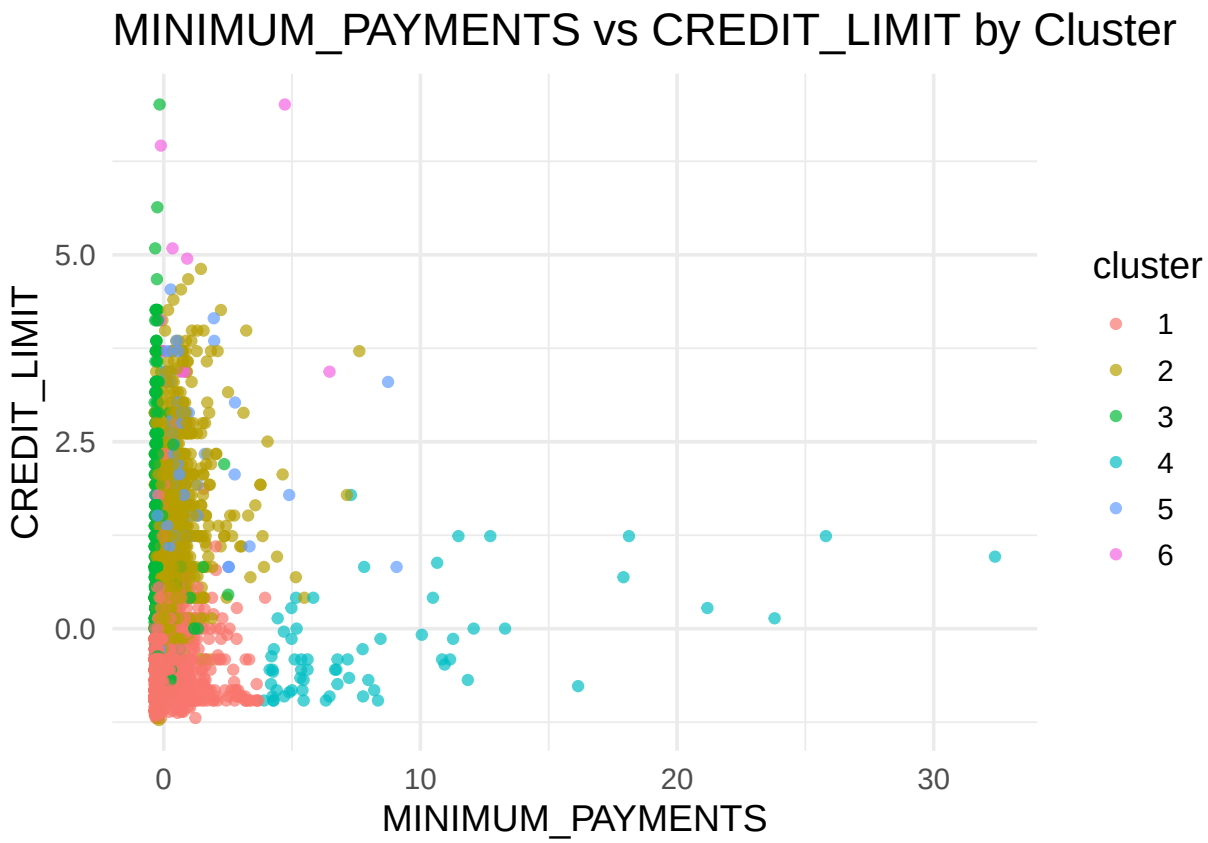
```r
# Load necessary libraries
library(ggplot2)
library(tidyr)

# Convert 'cluster' to a factor for proper coloring
data_final$cluster <- as.factor(data_final$cluster)

# Reshape the data into long format for ggplot
data_long <- data_final %>%
  pivot_longer(cols = c("BALANCE", "CASH_ADVANCE", "PAYMENTS"),
               names_to = "Variable", values_to = "Value")

# Plot each variable against cluster
ggplot(data_long, aes(x = Value, y = cluster, color = cluster)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~ Variable, scales = "free_x") +
  theme_minimal(base_size = 14) +
  labs(title = "BALANCE, CASH_ADVANCE, PAYMENTS vs Cluster", x = "Value", y = "Cluster") +
  theme(legend.position = "right")
```




Wildly varied balance, second highest payments, average purchases. The special thing about this cluster is that these people have the highest cash advance by far - there is even one extreme case that has like 25 cash advance points. We call these people "The Money Borrowers".

**Cluster 4 (Purple): The High Riskers**

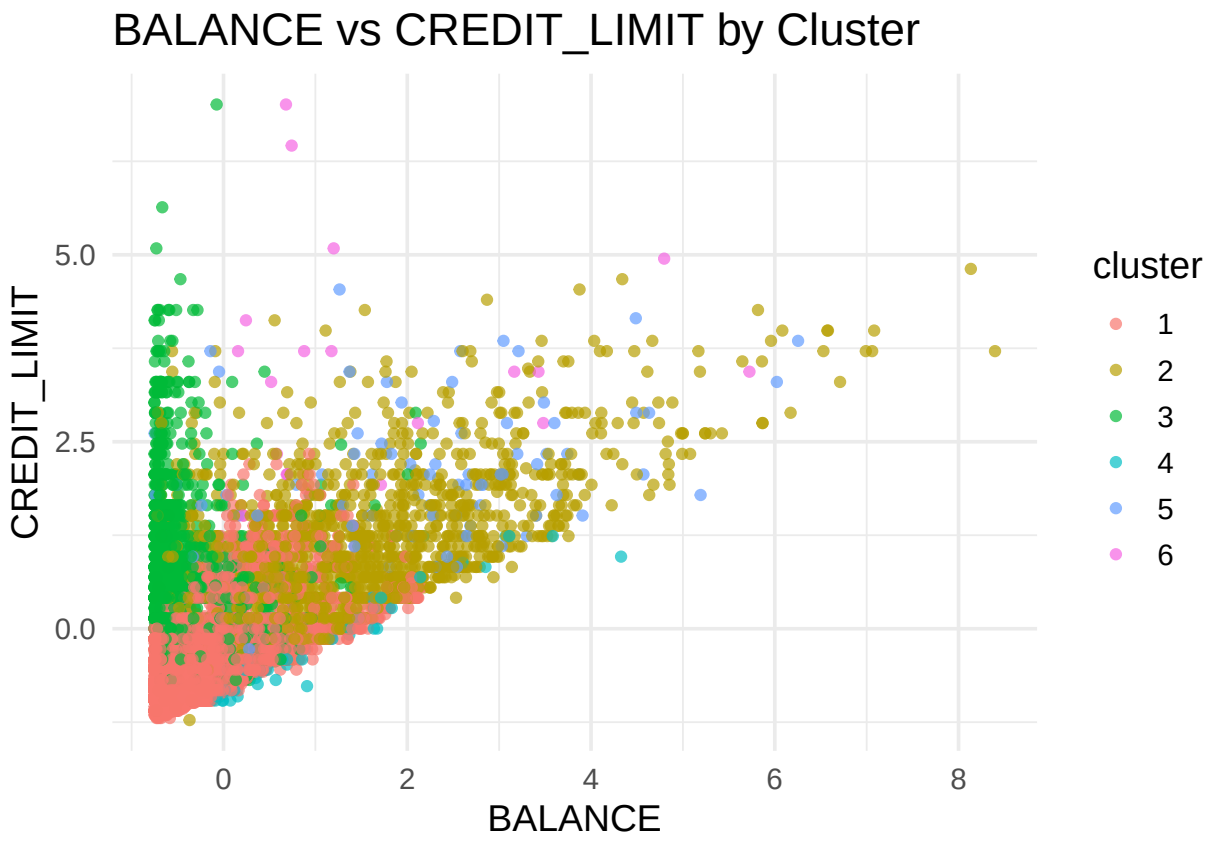
```r
# Load necessary libraries
library(ggplot2)

# Convert 'cluster' to a factor for proper coloring
data_final$cluster <- as.factor(data_final$cluster)

# Plot MINIMUM_PAYMENTS vs CREDIT_LIMIT colored by cluster
ggplot(data_final, aes(x = MINIMUM_PAYMENTS, y = CREDIT_LIMIT, color = cluster)) +
  geom_point(alpha = 0.7) +
  theme_minimal(base_size = 14) +
  labs(title = "MINIMUM_PAYMENTS vs CREDIT_LIMIT by Cluster",
       x = "MINIMUM_PAYMENTS", y = "CREDIT_LIMIT") +
  theme(legend.position = "right")
```




This group has absurdly high minimum payments while having the second lowest credit limit. It looks like the bank has identified them as higher risk.

**Cluster 5 (Brown): The Wildcards**

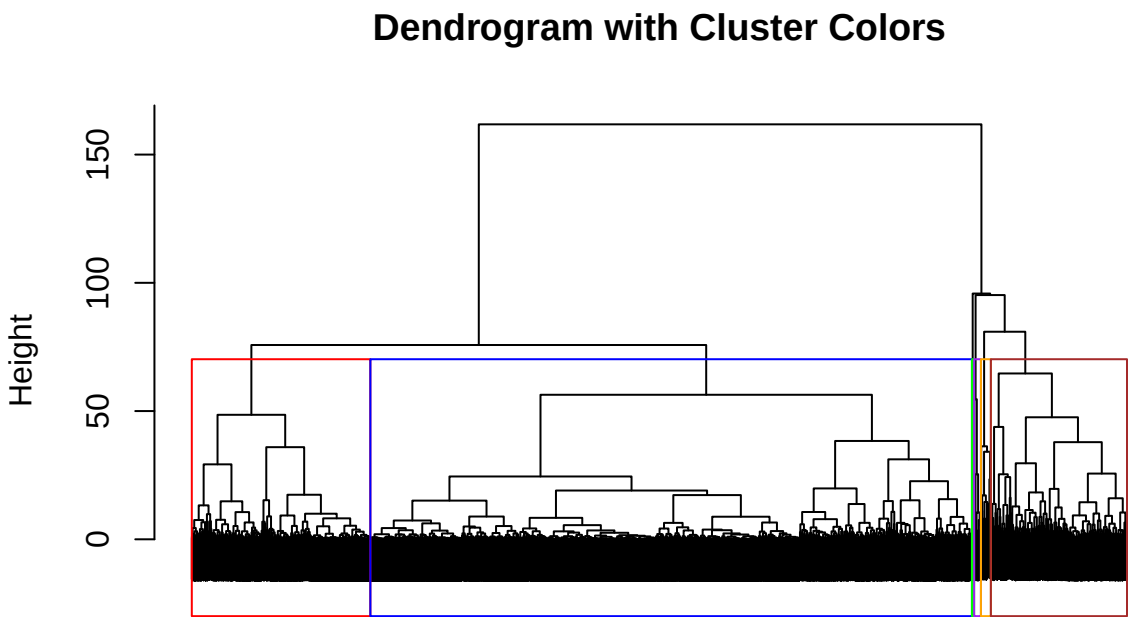
```
# Load necessary libraries
library(ggplot2)

# Convert 'cluster' to a factor for proper coloring
data_final$cluster <- as.factor(data_final$cluster)

# Plot BALANCE vs CREDIT_LIMIT colored by cluster
ggplot(data_final, aes(x = BALANCE, y = CREDIT_LIMIT, color = cluster)) +
  geom_point(alpha = 0.7) +
  theme_minimal(base_size = 14) +
  labs(title = "BALANCE vs CREDIT_LIMIT by Cluster",
       x = "BALANCE", y = "CREDIT_LIMIT") +
  theme(legend.position = "right")
```




This group is troublesome to analyze and to come up with a good marketing strategy towards, as both their credit limit and balance values are wildly varied. As you can see, the above graph looks like half of it was made of the color brown!

**Dendogram**

```
# Load necessary libraries
library(cluster)

# Remove 'cluster' column for clustering
data_no_cluster <- data_final[, !(names(data_final) %in% 'cluster')]

# Compute the distance matrix
dist_matrix <- dist(data_no_cluster, method = "euclidean")

# Perform hierarchical clustering
hc <- hclust(dist_matrix, method = "ward.D2")

# Cut the tree into 6 clusters (based on your earlier choice)
clusters <- cutree(hc, k = 6)

# Plot the dendrogram with colored clusters
plot(hc, labels = FALSE, main = "Dendrogram with Cluster Colors", xlab = "", sub = "", cex = 0.7)
rect.hclust(hc, k = 6, border = c("red", "blue", "green", "purple", "orange", "brown"))
```




## Summary and Possible Marketing Strategy

**Summary**

We have learned a lot from this dataset by segmenting the customers into six smaller groups: the Average Joe, the Active Users, the Big Spenders, the Money Borrowers, the High Riskers, and the Wildcards. To conclude this cluster analysis, let's sum up what we have learned and some possible marketing strategies: * The Average Joe do not use credit card very much in their daily life. They have healthy finances and low debts. While encouraging these people to use credit cards more is necessary for the company's profit, business ethics and social responsibility should also be considered. * Identify active customers in order to apply proper marketing strategy towards them. These people are the main group that we should focus on. * Some people are just bad at finance management - for example, the Money Borrowers. This should not be taken lightly. * Although we are currently doing a good job at managing the High Riskers by giving them low credit limits, more marketing strategies targeting this group of customers should be considered.

## Conclusion

In this project, we have performed data preprocessing, feature extraction, experimented with the Clustering algorithm (Agglomerative Hierarchical Clustering), data visualizations, and business analytics.