Credit Card Segmentation

By

Trayan Das

Table of Contents:

Introduction:-

I am given a dataset which summarizes the usage behaviour of about 9000 active credit card holders during the last 6 months. This dataset has 18 variables, which represent different behavioural characteristics of the credit card holders. I have to derive intelligent KPIs from this data & use these KPIs to gain insights of the customer profiles. Apart from that, I'll have to reduce the dimensionality of the data & also use a clustering algorithm to reveal the behavioural segments of credit card holders.

Data Section:-

There are a total of 8950 observations & 18 variables. The observations denote different credit card holders. And these are the different variables.

1. CUST_ID: Credit card holder ID
2. BALANCE: Monthly average balance (based on daily balance averages)
3. BALANCE_FREQUENCY: Ratio of last 12 months with balance
4. PURCHASES: Total purchase amount spent during last 12 months
5. ONEOFF_PURCHASES: Total amount of one-off purchases
6. INSTALLMENTS_PURCHASES: Total amount of instalment purchases
7. CASH_ADVANCE: Total cash-advance amount
8. PURCHASES_ FREQUENCY: Frequency of purchases (percentage of months with at least on purchase)
9. ONEOFF_PURCHASES_FREQUENCY: Frequency of one-off-purchases
10. PURCHASES_INSTALLMENTS_FREQUENCY: Frequency of instalment purchases
11. CASH_ADVANCE_ FREQUENCY: Cash-Advance frequency
12. CASH_ADVANCE_TRX: Average amount per cash-advance transaction
13. PURCHASES_TRX: Average amount per purchase transaction
14. CREDIT_LIMIT: Maximum Credit limit for a customer
15. PAYMENTS: Total payments (due amount paid by the customer to decrease their statement balance) in the period
16. MINIMUM_PAYMENTS: Total minimum payments due in the period
17. PRC_FULL_PAYMENT: Percentage of months with full payment of the due statement balance
18. TENURE: Number of months as a customer

Objectives:-

1. Advanced data preparation. Build an 'enriched' customer profile by deriving 'intelligent' KPI's such as monthly average purchase and cash advance amount, purchases by type (one-off, instalments), average amount per purchase and cash advance transaction, limit usage (balance to credit limit ratio), payments to minimum payments ratio etc.
2. Advanced reporting. Use the derived KPI's to gain insight on the customer profiles.
3. Clustering. Apply a data reduction technique factor analysis for variable reduction technique and a clustering algorithm to reveal the behavioural segments of credit card holders.

Softwares Used:-

1. Python 3.7.1 for 64 bit.
2. R 3.6.3 for 64 bit.

Model Development:-

1. Model Development for Python

A) Exploratory Data Analysis

All the required libraries are loaded. After setting the working directory, the given 'credit card' dataset in CSV format is loaded into the 'credit' object. I can see that out of 18 variables, 'CUST_ID' is an object. And BALANCE, BALANCE_FREQUENCY, PURCHASES, ONEOFF_PURCHASES, INSTALLMENTS_PURCHASES, CASH_ADVANCE, PURCHASES_ FREQUENCY, ONEOFF_PURCHASES_FREQUENCY, PURCHASES_INSTALLMENTS_FREQUENCY, CASH_ADVANCE_ FREQUENCY, CREDIT_LIMIT, PAYMENTS, MINIMUM_PAYMENTS & PRC_FULL_PAYMENT are float values. And CASH_ADVANCE_TRX, PURCHASES_TRX, TENURE are int values. Then, I derived the number of unique values for each variable.

B) Missing Value Analysis

I checked & found out that there're a total of 314 missing values. So, I checked which variables among the dataset have missing values. I found out that, 'CREDIT_LIMIT' & 'MINIMUM_PAYMENTS' variables have missing values in them. Now, I've arranged the missing values for each variable with their respective percentage out of the whole variable & saved them in another dataset, named 'missing_Val'. And I have saved this dataframe into 'Missing_perc1' file in 'CSV' format.
So, now I have 2 options. I can either (i) drop the missing values (as according to industry standard, if the amount of missing value in a variable is less than 5%, we can drop these missing values without much affecting the information), or, I can impute them (as according to industry standard, if the amount of missing value in a variable is less than 30%, we can impute these missing values without much affecting the information). I chose to impute these missing values, in order to save the information. Here, we can impute these missing values using either central tendency (mean,

median) or using K-nearest Neighbour (knn) algorithm. I can't use mode method here, as the missing values have occurred in numerical variables. So, I've taken a known value (5th value in MINIMUM_PAYMENTS variable), made a note of its original value & assigned 'nan' to it, effectively making it a missing value. Then I've calculated the value of this specific variable using mean, median & knn algorithm. And, I found out, that using 'Median' method, I got the predicted value, which is closest to the original value, compared to mean & knn method. So, I've reloaded the dataset, and applied the 'median' method to impute the missing values in 'CREDIT_LIMIT' & 'MINIMUM_PAYMENTS' variables. But, we've to remember, that as the mean & median are constants for any given column, & knn will vary with different data points. I can get different best methods if I take another known value for method selection. After that, I've again checked which variables have missing values. This time, I got no missing values.

C) KPI derivation from the data

I have to make a total of 6 KPIs from the dataset. The derivation process of these KPIs is explained below.

(i) Monthly average purchase: It means how much amount worth of purchase a specific credit card holder has done on a monthly basis. I've divided the 'PURCHASES' variable by 'TENURE' variable. That's how I got this KPI for each credit card holder.

(ii) Monthly Cash advance amount: It means how much amount of advance cash was taken by a credit card holder on monthly basis. For this, I've divided 'CASH_ADVANCE' variable by 'TENURE' variable.

(iii) Purchases by type: It means whether a customer is doing instalment purchases/ one-off purchases/ both/ none with his/her credit card. I've created a function named 'purchase', which returns-

- none: if, for a customer, the amount of one-off purchase & instalment purchase is zero.
- Both_oneoff_installment: if, for a customer, the amount of both one-off purchase & instalment purchase is greater than zero.
- one_off: if, for a customer, the amount of one-off purchase is greater than zero & instalment purchase is zero.
- Instalment: if, for a customer, the amount of one-off purchase is zero & instalment purchase is greater than zero.

Then, I've applied this function on the credit dataset & saved the returned values into a newly Created 'purchase_type' variable. I found out that 2774 customers did both type of purchases, 2260 customers did only instalment purchase, 2042 customers did neither type of purchases, 1874 customers did only one-off purchase.

(iv) average amount per purchase: It means average amount per purchase transaction for a credit card holder. This KPI is already provided within the dataset in the 'PURCHASES_TRX' variable.

(v) average amount per cash advance transaction: It means average amount per cash-advance transaction for a credit card holder. This KPI is already provided within the dataset in the' CASH_ADVANCE_TRX' variable.

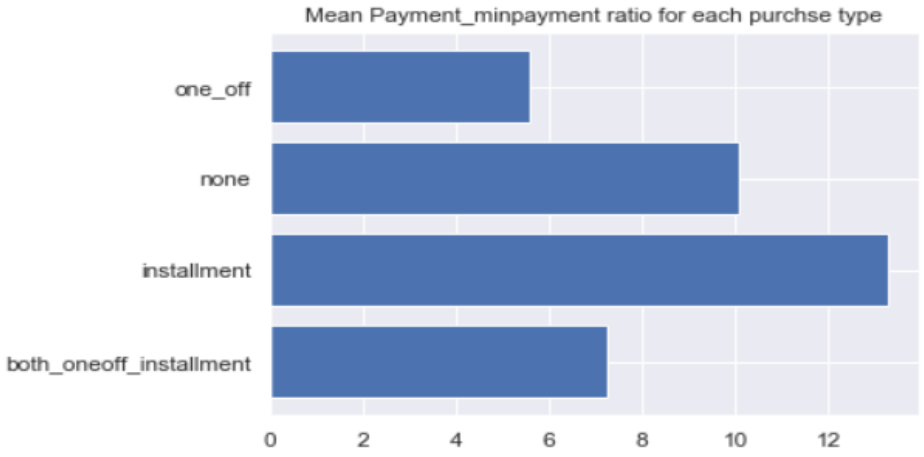
(vi) limit usage: It means balance to credit limit ratio for a credit card holder. Here, lower value implies lower balance remaining in the card, which in turn means that customers are spending more. For this, I've divided 'BALANCE' variable by 'TENURE' variable.

(vii) payments to minimum payments ratio: It means the ratio of total payments paid by the customer & total minimum payments due in the period for that customer. For this, I've divided the 'PAYMENTS' variable by 'MINIMUM_PAYMENTS' variable.

D) Insights from derived KPIs

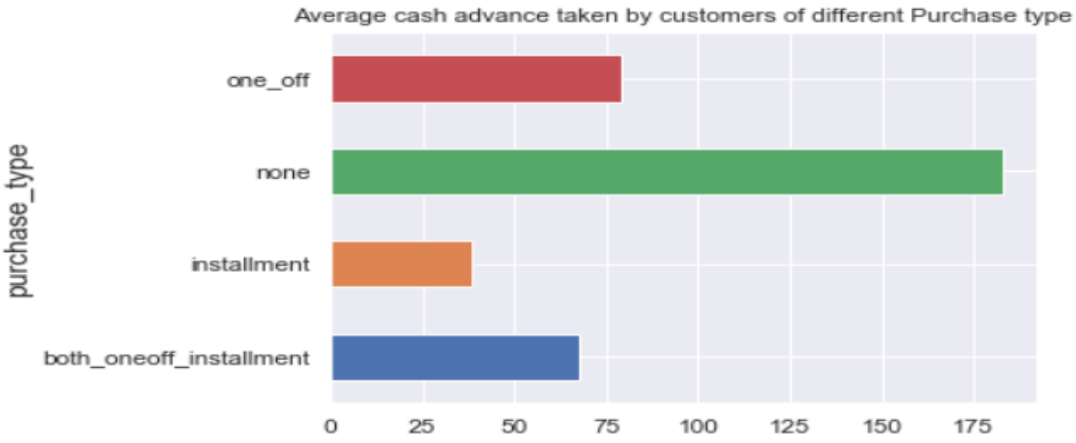
I have made a total 3 insights from these derived KPIs. These are-

(i) Insight 1: I've previously seen that the newly derived 'purchase_type' variable is a categorical variable, with 4 categories- one_off, none, instalment, both_oneoff_installment. I've derived the average of the 'payment_minpayment_ratio' variable for each of these categories. And I've plotted them using the subplots function in matplotlib library.



From here, it can be seen, that customers with installment purchases are paying more dues compared to other groups.

(ii) Insight 2: I've derived the average of the 'Monthly_cash_advance' variable for each of the categories in 'purchase_type'. And I've plotted them using the matplotlib library.

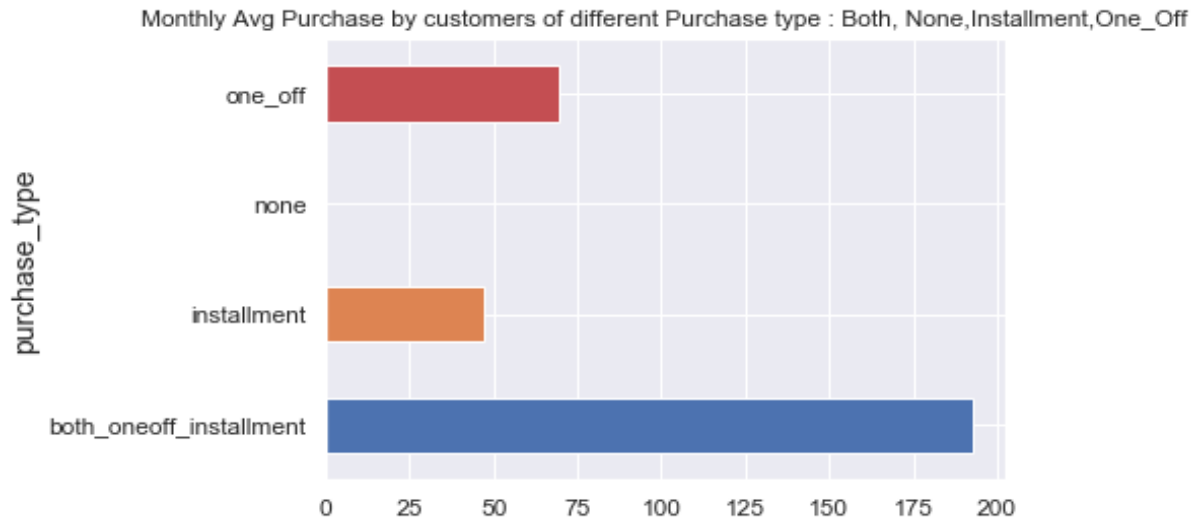


From here, I can see Customers who don't do either one-off or installment purchases, take more cash on advance**.**

(iii) Insight 3: I've derived the average of the 'limit_usage' variable for each of the categories in 'purchase_type'. And I've plotted them using the matplotlib library.

Here, 'limit_usage' represents the ratio of remaining balance & overall credit limit of a credit card. Lower value implies lower balance remaining in the card, which in turn means that customers are spending more. From the subplot, I can see that customers who're doing instalment purchase, are spending more compared to other groups.

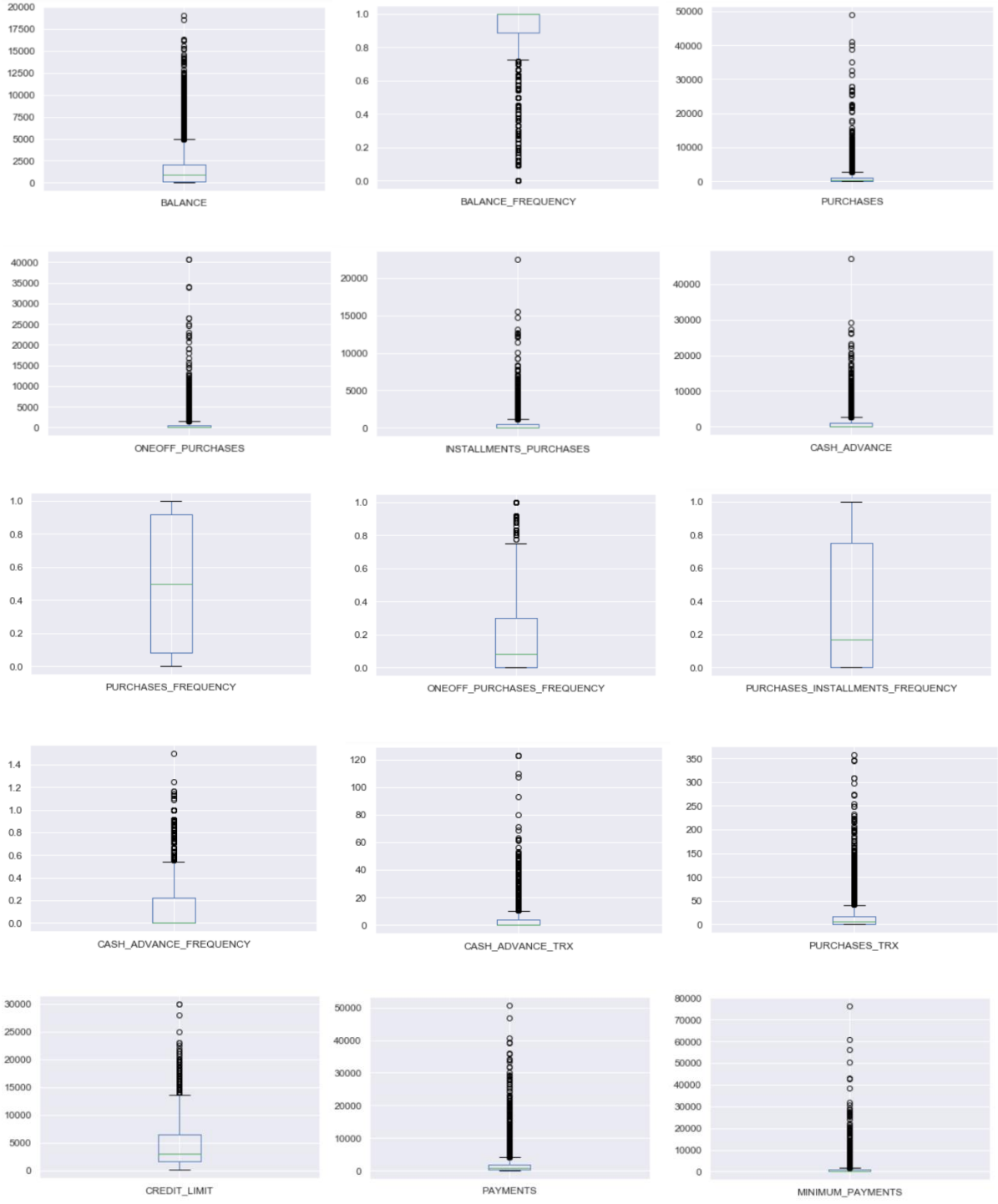
(iv) Insight 4: I've derived the average of the 'Monthly_avg_purchase' variable for each of the categories in 'purchase_type'. And I've plotted them using the matplotlib library.

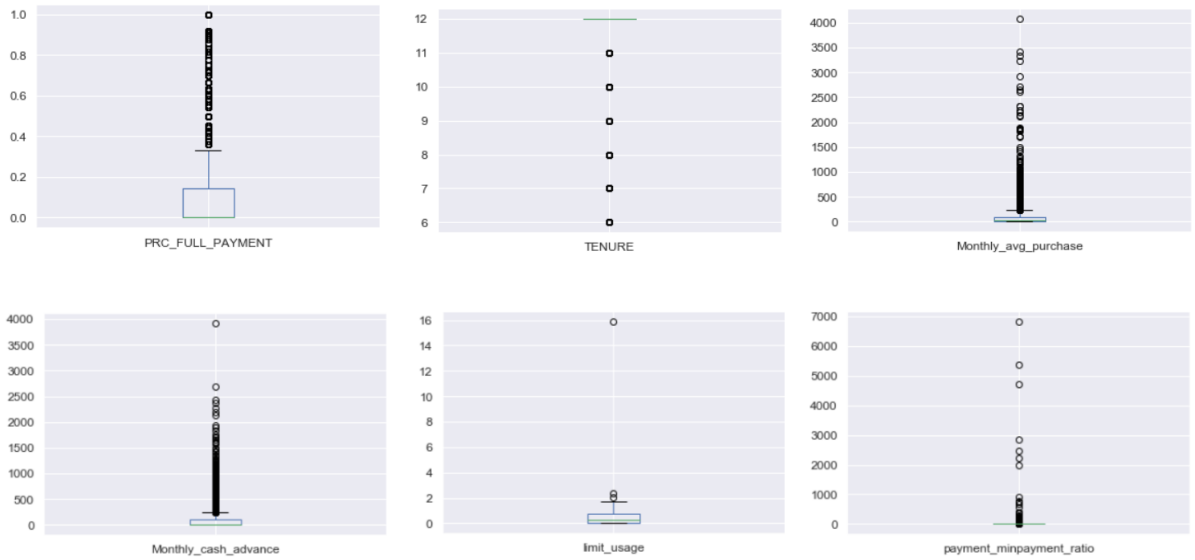


From here, I can see Customers who do installment purchases, do less purchase compared to other groups.

E) Outlier Analysis

In order to do outlier analysis, I've saved all the numerical variables into 'cr_num' object. Then, for all the columns in 'cr_num', I've generated boxplots to see if they contain outliers or not.





Here, I can see that apart from 'PURCHASES_FREQUENCY' & 'PURCHASES_INSTALLMENT_FREQUENCY', all the other variables contain outliers. So, I'll have to remove the outlier effect. But, its obvious, that these outliers were not made by human/machine errors & they create a significant association, and in turn, will contribute to the end result. That's why, I did not drop or impute them with central tendency or knn, I have used logarithmic transform to get rid of the extreme range of the dataset. I've log- transformed all the numeric variables of the 'credit' dataset by dropping the 2 categorical variables & saving the output in 'cr_log' dataset.

F) Data preparation for Machine Learning Algorithm

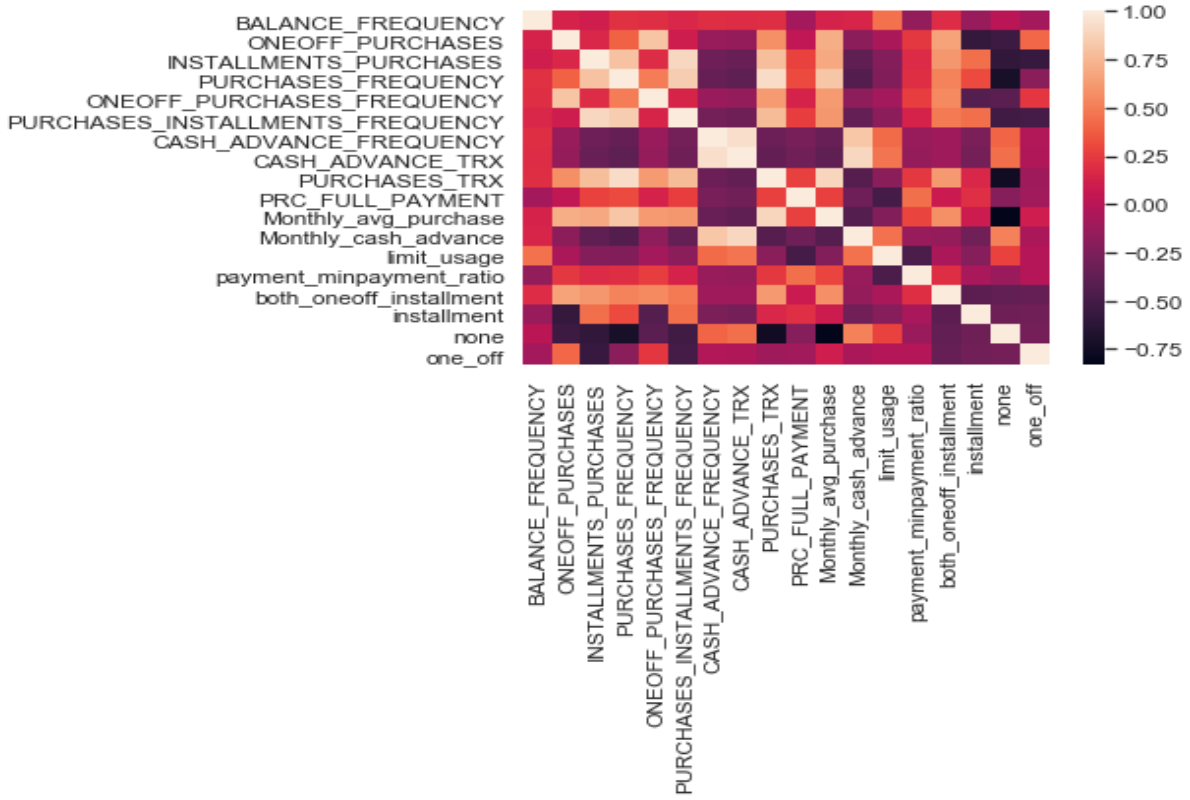
Before advancing further, I've subsetted those variables from cr_log, which have been used to derive KPIs, apart from ONEOFF_PURCHASES & INSTALLMENTS_PURCHASES, & saved the remaining variables into the 'cr_pre' dataset. I didn't exclude these 2 variables as these 2 variables don't have a linear relation with their derived KPI, purchase_type.

Now, in my original variable (credit), I've converted the categorical variable 'purchase_type' to number type using get_dummies function from pandas library & saved these resultant numeric variables (both_oneoff_installment,instalment,none,one_off) with the variables from 'credit' dataset into 'cre_original' variable.

Now, I need to join these same dummy numeric variables with my final log-transformed dataset, 'cr_pre'. So, I copied the 'purchase_type' categorical variable from 'credit' dataset to the 'cr_pre' dataset; & using dummies, I converted this variable to four numeric dummy variables & attached these variables with 'cr_pre' & saved the output into 'cr_dummy' dataset. Afterwards, I dropped the 'purchase_type' variable from 'cr_dummy', as this was a redundant variable (this variable & the derived dummies conveyed same information).

G) Correlation Analysis

I've generated a heatmap of correlation of the variables in 'cr_dummy' dataset using the seaborn library & 'corr()' function.



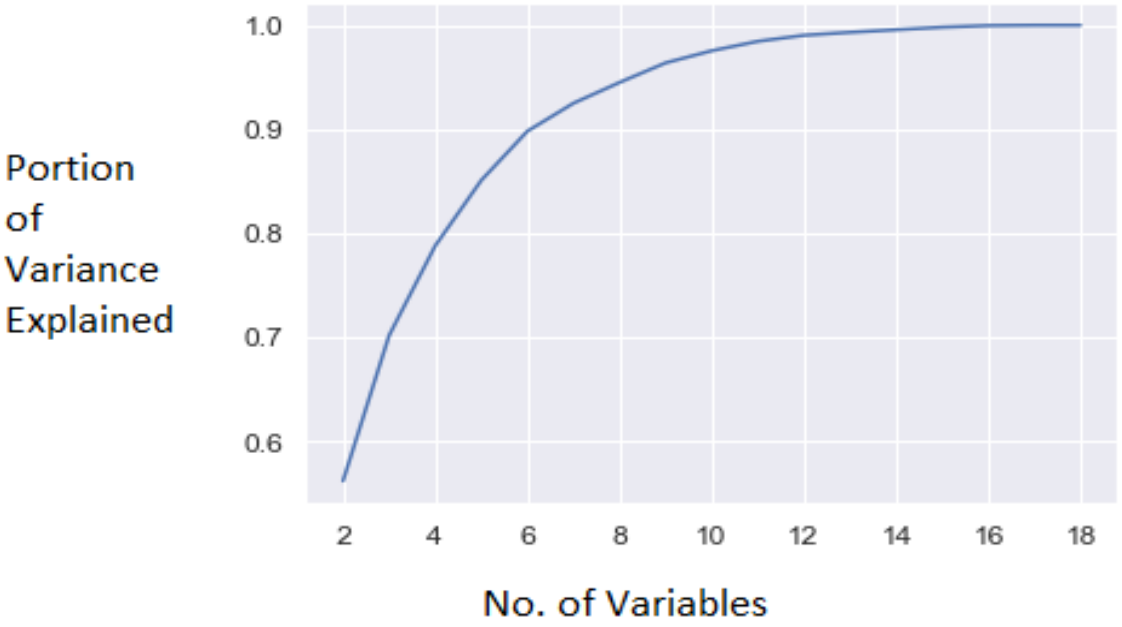
Here, its clearly visible that there's correlation between different variables in the dataset. So, I'll have to apply a dimensionality reduction technique to get rid of the multicollinearity in the dataset.

H) Dimension reduction using PCA

I'm choosing to apply Principal Component Analysis (PCA) as the dimension reduction method as by using PCA I'll be able to identify the direction of highest to smallest variance; which in turn will enable me to drop the less variant features.

Before applying PCA, I'll standardize the data to put the data on the same scale & so that the data has zero mean & unit variance, which is important for the optimal performance of PCA. So, I've imported the StandardScaler function from scikit-learn library; & using StandardScaler, I've scaled the 'cr_dummy' dataset, & saved the output into 'cr_scaled' object.

After that, I've imported PCA function from scikit-learn library, & applied PCA on the 'cr_scaled' data for its 18 variables & saved the result into 'cr_pca' object. Then I extracted the variance explained by the PCA variables & saved into 'var_ratio'; after that, I plotted 'var_ratio '.

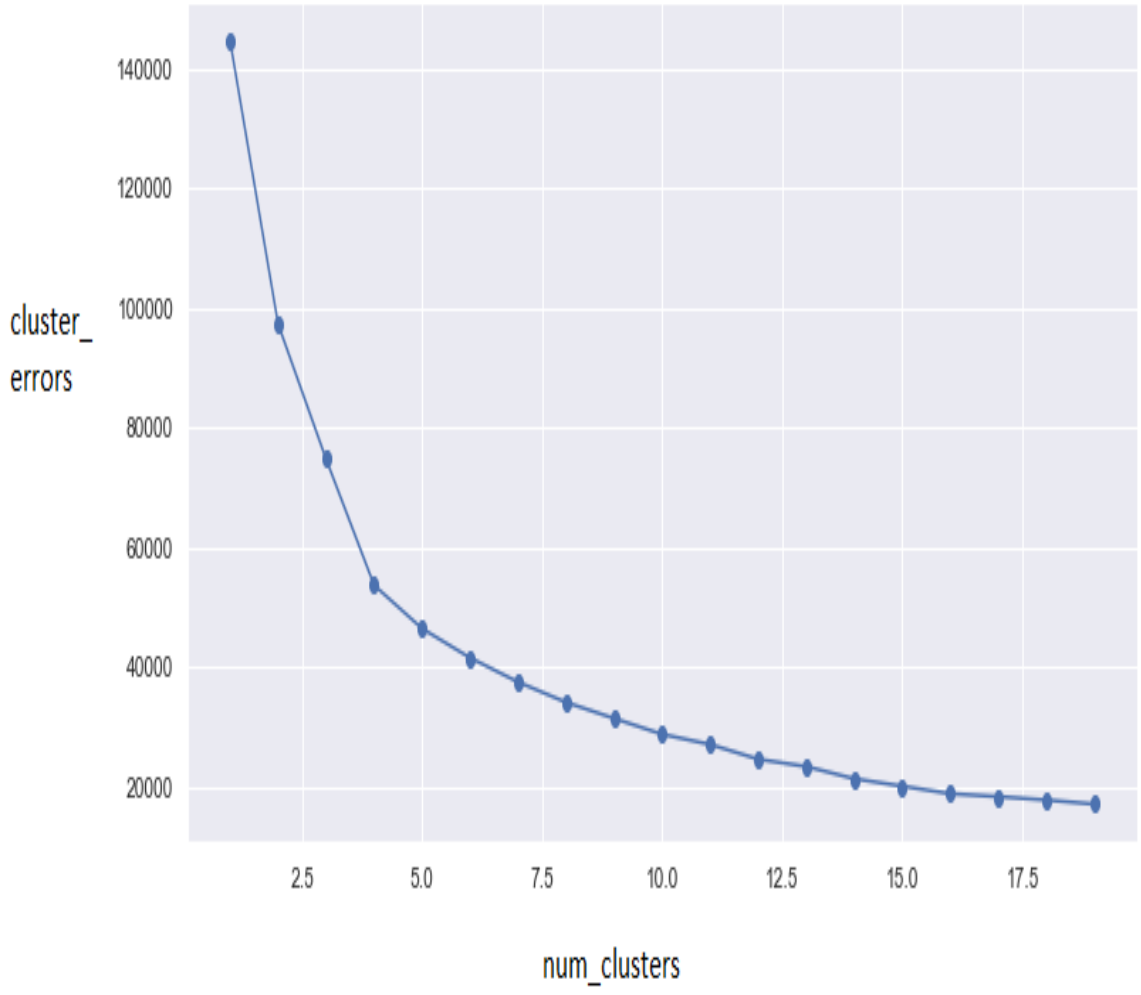


From there, I can see that 6 components are explaining more than 90% of variance, so I selected 6 components. Then, I applied PCA on the 'cr_scaled' data for 6 variables. Finally, I freezed no. of PCA variables = 6 & applied PCA on 'cr_scaled' data & saved the PCA array (with n = 6) as pc_final. Then I saved the result into 'reduced_cr' object. Then I saved 'reduced_cr' into 'dd' dataframe. So initially I had 18 variables, now I have 6 variables. I should note that after dimensionality reduction, there usually isn't a particular meaning assigned to each principal component. The new components are just the six main dimensions of variation.

Then I renamed the 6 different columns of pc_final into PC_0, PC_1, PC_2, PC_3, PC_4, PC_5 & saved it to a dataframe; & set the column names of 'cr_dummy' as the indexes of this dataframe. As the components of pc_final represented different directions of each original variable (eigen vectors, as per PCA), this newly created dataset represents eigen vector for each component. After that, I got the explained variance ratio for these 6 components.

I) Clustering Algorithm

In order to do clustering, first I'll have to estimate optimum no. of clusters to be built. I've used the 'Elbow Method' to estimate this. I've imported KMeans function from scikitlearn library. Then using KMeans, I've derived the errors for each no. of clusters in the range of (1-20) & saved the output in clusters_df dataframe. Then I've plotted cluster errors in the y-axis & no. of clusters in x-axis.



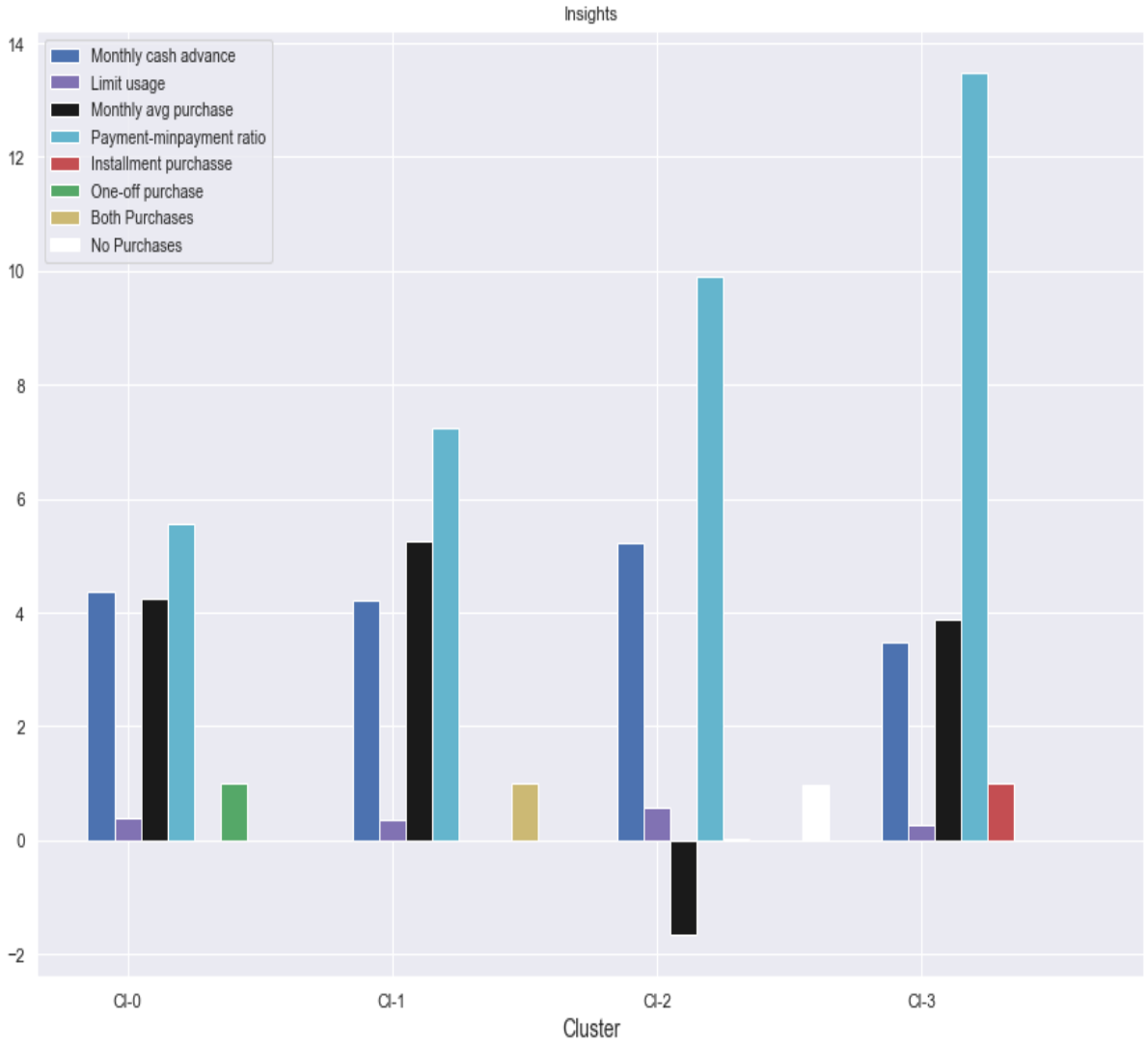
From this graph, I got the elbow range (3,4,5). So, I've specified no. of clusters as 4. Thereafter, I did clustering on the 'reduced_cr' object with no. of clusters set at 4; & saved the resultant clusters into km_4 object. After that, using the 'value_counts' function, I derived no. of observations in each cluster. I found that these clusters have 1870, 2764, 2101 & 2215 observations in each cluster respectively.

J) Behaviroul Analysis & Segmentation of credit card holders:

Now, I've selected only the KPI variables from the cre_original dataset, as I'm interested to analyse the behaviour of customers based on the KPIs. Then, I've merged the clusters with the cre_original dataframe & saved the result in cluster_df_4 object. I've selected this dataframe, as it's un-transformed, hence will represent the original data which was provided to us.

Then I've found the mean value for each variable for each cluster & saved the result into cluster_4 dataframe.

After that, I've generated a subplot where x-axis represents the 4 clusters & in y-axis, we have 'Monthly cash advance', 'Limit usage', 'Monthly avg purchase', 'Payment-minpayment ratio', 'Installment purchase', 'One-off purchase', 'Both Purchases' & 'No Purchases'.



Then I've derived the percentage of each cluster in the total customer base.

From the generated plot, we can divide the total customer base into 4 clusters based on their behaviour. The detailed behaviroul analysis is described below.

Customers from cluster 0: They're taking the high advance cash from their card & doing only one-off purchases. Their average purchase using this credit card is medium, they're spending less amount & are not paying their dues in time. They consist of approximately 21 % of total customers.

Customers from cluster 1: They're taking the medium amount of advance cash from their card & doing both types of purchases. Their average purchase is highest among groups. They're spending high amount but they're making comparatively less minimum payments. They consist of approximately 31 % of total customers.

Customers from cluster 2: They're taking highest amount of advance cash from their card & aren't doing any purchases. They're spending least amount on their card & are paying their dues in time. They consist of approximately 23 % of total customers.

Customers from cluster 3: They're taking the least amount of advance cash from their card & doing only instalment purchases. They're doing medium amount of purchase. They're spending highest amount on their card & are making highest minimum payments compared to others. They consist of approximately 25 % of total customers.

K) Recommended Business Strategy

Here, I'll suggest business strategies for each of the customer groups. They are as follows-

(i) Cluster 0: As these customers are spending less amount & doing medium purchases, we can give them discount offers on purchases to encourage them to increase their purchases. & as they aren't paying their dues in time, we shouldn't increase their credit limit.

(ii) Cluster 1: As they're spending high amount & doing highest amount of purchases, we can give them discount offers on purchases & reward points for the same, which will hopefully increase their spending further. We should be cautious before increasing their credit limit as they're making comparatively less minimum payments.

(iii) Cluster 2: As they're not doing any purchases & spending least amount on their card, we can give them discount offers on purchases & reward points for the same in order to encourage them to spend more.

(iv) Cluster 3: As they're doing medium amount of purchase, we can give them discount offers, in order to encourage them to do more purchases.

2. Model Development for R

A) Exploratory Data Analysis & Initial Data Preparation

All the required packages are loaded. After setting the working directory, the given 'credit card' dataset in CSV format is loaded into the 'credit' dataframe. I can see that out of 18 variables, 'CUST_ID' is a factor. And BALANCE, BALANCE_FREQUENCY, PURCHASES, ONEOFF_PURCHASES, INSTALLMENTS_PURCHASES, CASH_ADVANCE, PURCHASES_ FREQUENCY, ONEOFF_PURCHASES_FREQUENCY, PURCHASES_INSTALLMENTS_FREQUENCY, CASH_ADVANCE_ FREQUENCY, CREDIT_LIMIT, PAYMENTS, MINIMUM_PAYMENTS & PRC_FULL_PAYMENT are float values. And CASH_ADVANCE_TRX, PURCHASES_TRX, TENURE are int values. As the 'CUST_ID' variable is not important, I dropped it. And I've converted CASH_ADVANCE_TRX, PURCHASES_TRX, TENURE variables to float values.

B) Missing Value Analysis

I've arranged the missing values for each variable with their respective percentage out of a whole variable & saved them in another dataset, named 'missing_val', from which I can see that there're missing values in CREDIT_LIMIT & MINIMUM_PAYMENTS variable. And I have saved this dataframe into 'Miising_perc' file in 'CSV' format.
So, now I have 2 options. I can either (i) drop the missing values (as according to industry standard, if the amount of missing value in a variable is less than 5%, we can drop these missing values without much affecting the information), or, I can impute them (as according to industry standard, if the amount of missing value in a variable is less than 30%, we can impute these missing values without much affecting the information). I chose to impute these missing values, in order to save the information. Here, we can impute these missing values using either central tendency (mean, median) or using K-nearest Neighbour (knn) algorithm. I can't use mode method here, as the missing values have occurred in numerical variables. So, I've taken a known value (4^{th} value in CREDIT_LIMIT variable), made a note of its original value & assigned 'NA' to it, effectively making it a missing value. Then I've calculated the value of this specific variable using mean, median & knn algorithm. And, I found out, that using 'KNN imputation method', I got the predicted value, which is closest to the original value, compared to mean & median method. So, I've reloaded the dataset, and applied the 'knn' method to impute the missing values in 'CREDIT_LIMIT' & 'MINIMUM_PAYMENTS' variables. After that, I've again checked which variables have missing values. This time, I got no missing values.

C) Deriving New KPIs

I have to make a total of 6 KPIs from the dataset. The derivation process of these KPIs is explained below.

(i) Monthly average purchase: It means how much amount worth of purchase a specific credit card holder has done on a monthly basis. I've divided the 'PURCHASES' variable by 'TENURE' variable. That's how I got this KPI for each credit card holder.

(ii) Monthly Cash advance amount: It means how much amount of advance cash was taken by a credit card holder on monthly basis. For this, I've divided 'CASH_ADVANCE' variable by 'TENURE' variable.

(iii) Purchases by type: It means whether a customer is doing instalment purchases/ one-off purchases/ both/ none with his/her credit card. I've created a new variable named 'purchase_type', in 'credit' dataset. The assigned values to this variable are-

- Instalment: if, for a customer, the amount of one-off purchase is zero & instalment purchase is greater than zero.
- one_off: if, for a customer, the amount of one-off purchase is greater than zero & instalment purchase is zero.
- none: if, for a customer, the amount of one-off purchase & instalment purchase is zero.
- Both_oneoff_installment: if, for a customer, the amount of one-off purchase & instalment purchase is greater than zero.

(iv) average amount per purchase: It means average amount per purchase transaction for a credit card holder. This KPI is already provided within the dataset in the 'PURCHASES_TRX' variable.

(v) average amount per cash advance transaction: It means average amount per cash-advance transaction for a credit card holder. This KPI is already provided within the dataset in the' CASH_ADVANCE_TRX' variable.

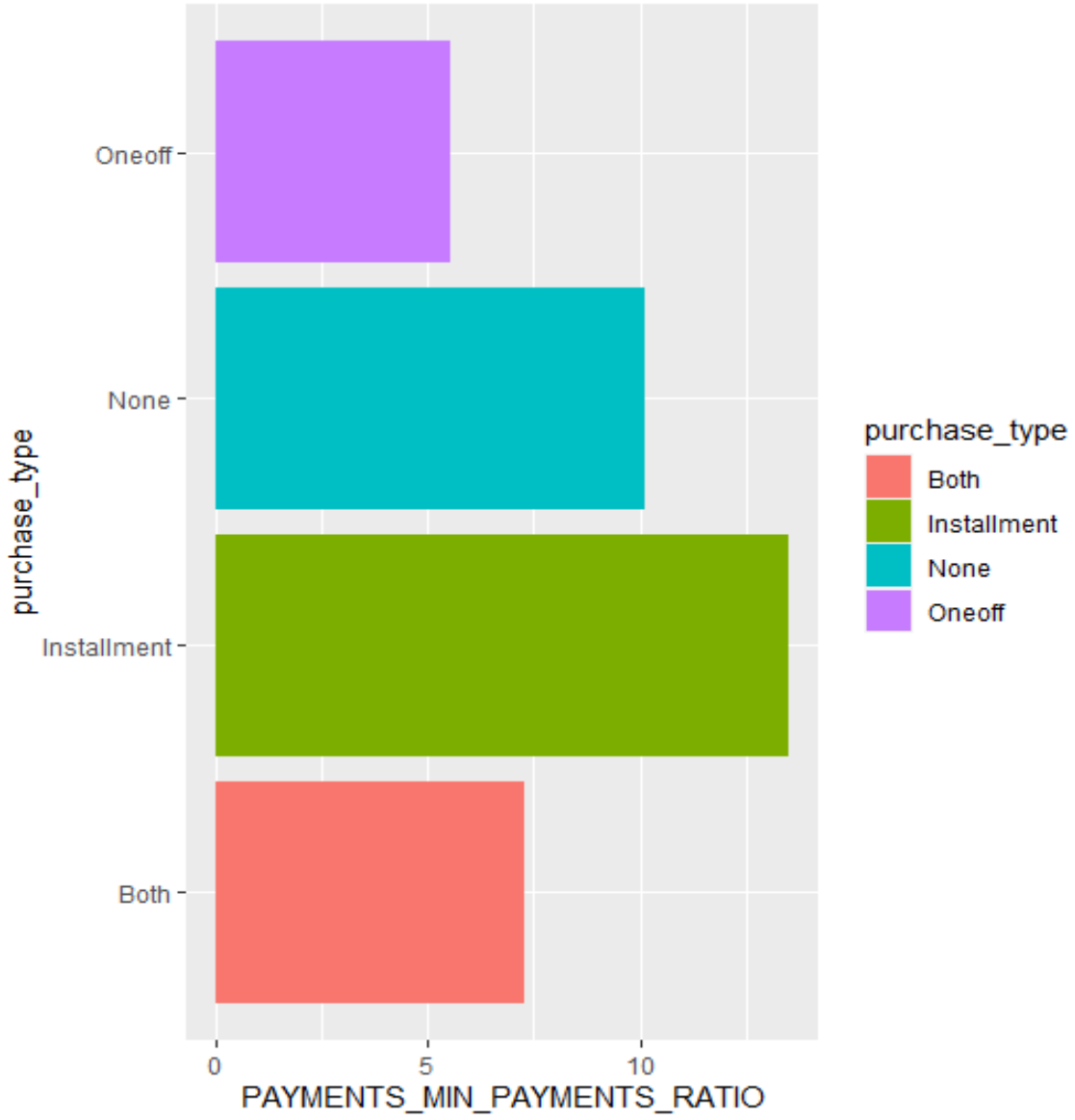
(vi) limit usage: It means balance to credit limit ratio for a credit card holder. Here, lower value implies lower balance remaining in the card, which in turn means that customers are spending more. For this, I've divided 'BALANCE' variable by 'TENURE' variable.

(vii) payments to minimum payments ratio: It means the ratio of total payments paid by the customer & total minimum payments due in the period for that customer. For this, I've divided the 'PAYMENTS' variable by 'MINIMUM_PAYMENTS' variable.

D) Insights from derived KPIs

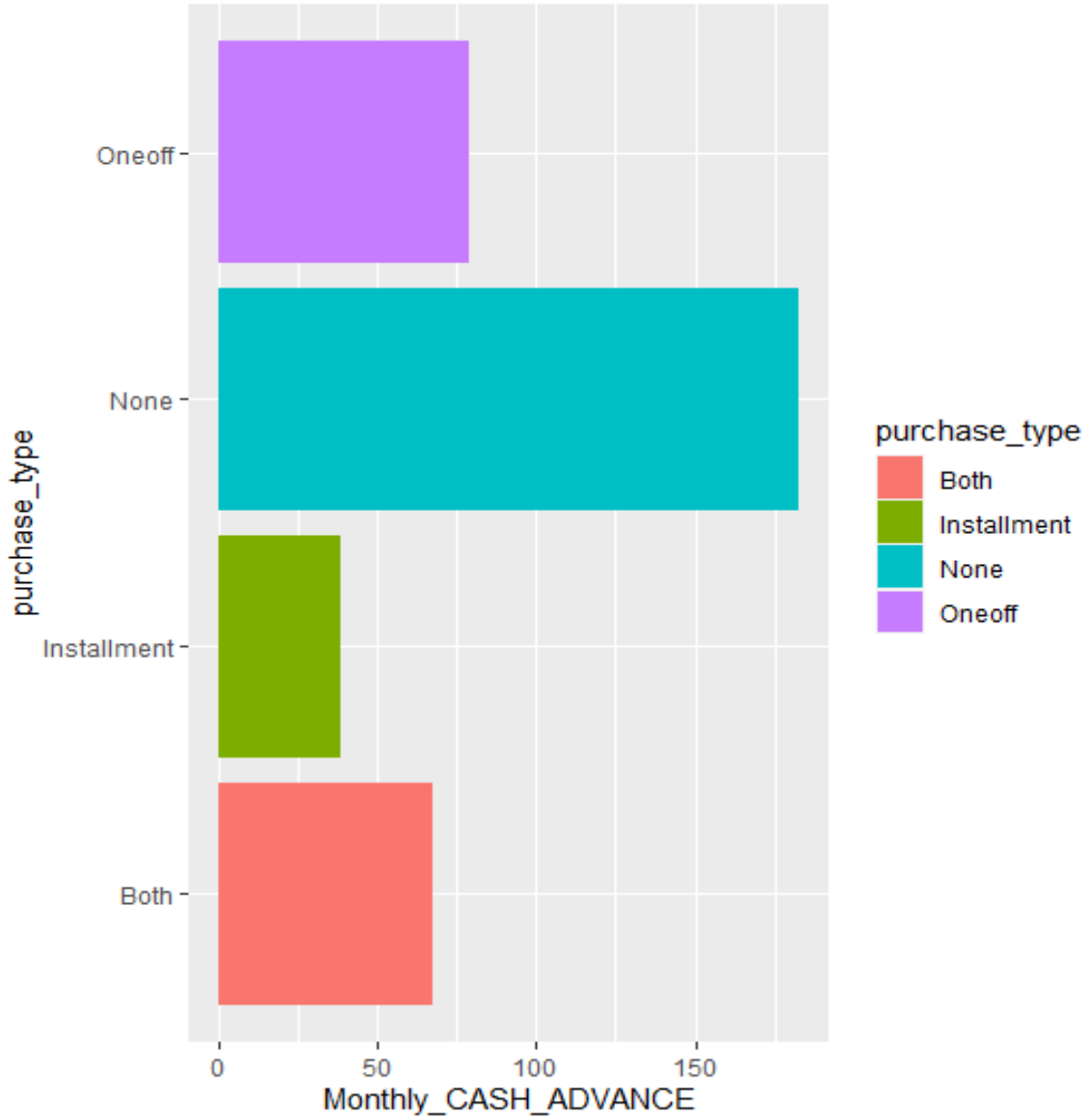
I have used the aggregate function to take the mean of "purchase_type", "Monthly_Avg_PURCHASES", "Monthly_CASH_ADVANCE", "LIMIT_USAGE", "PAYMENTS_MIN_PAYMENTS_RATIO" for each category (Instalment, Oneoff, None & Both) in purchase_type variable & saved the result into 'cr_selected' dataset. Then, using ggplot2 library, I made total 4 insights from these derived KPIs. These are-

(i) Insight 1:



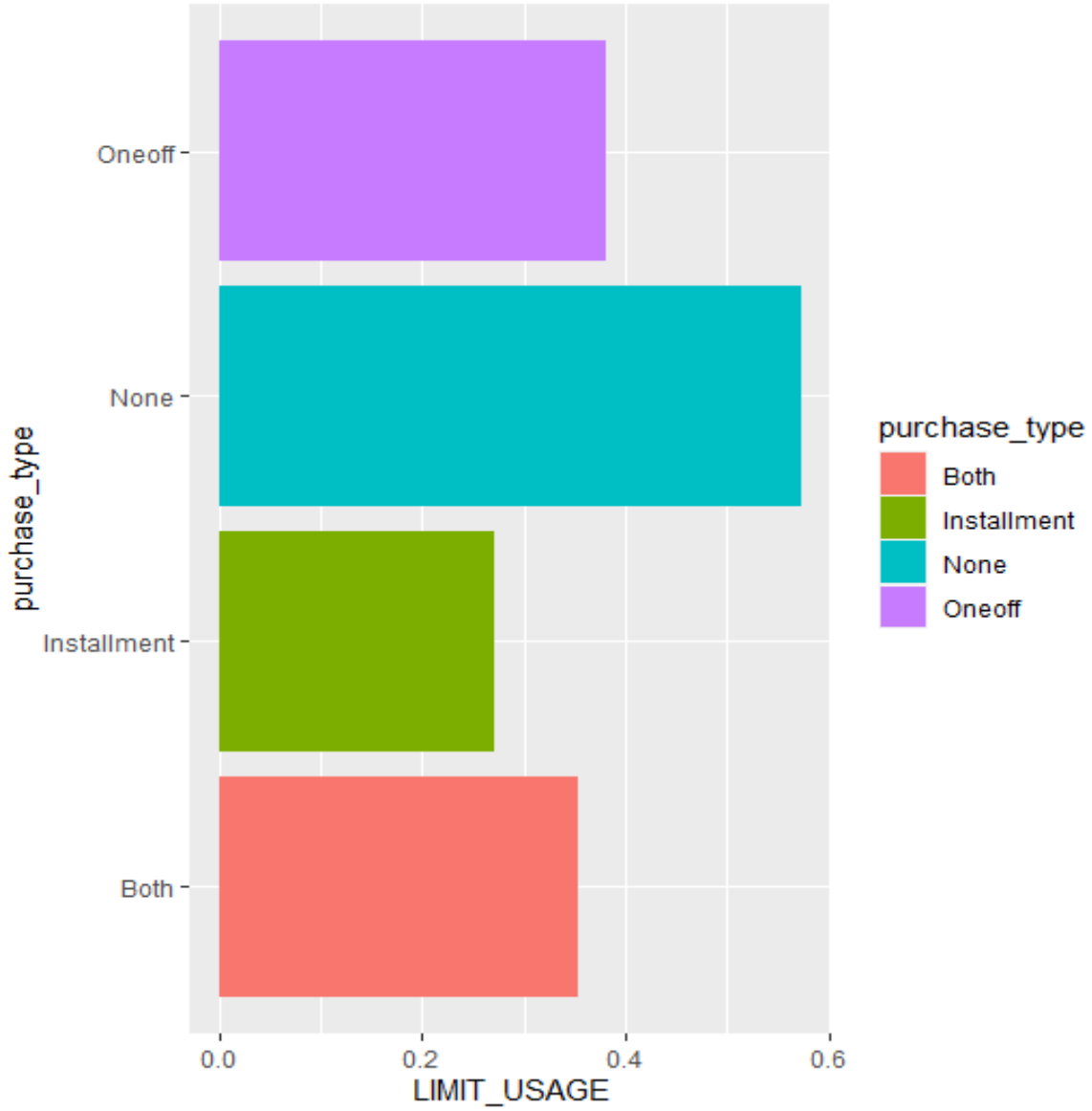
From here, it can be seen, that customers with installment purchases are paying more dues compared to other groups.

(ii) Insight 2:



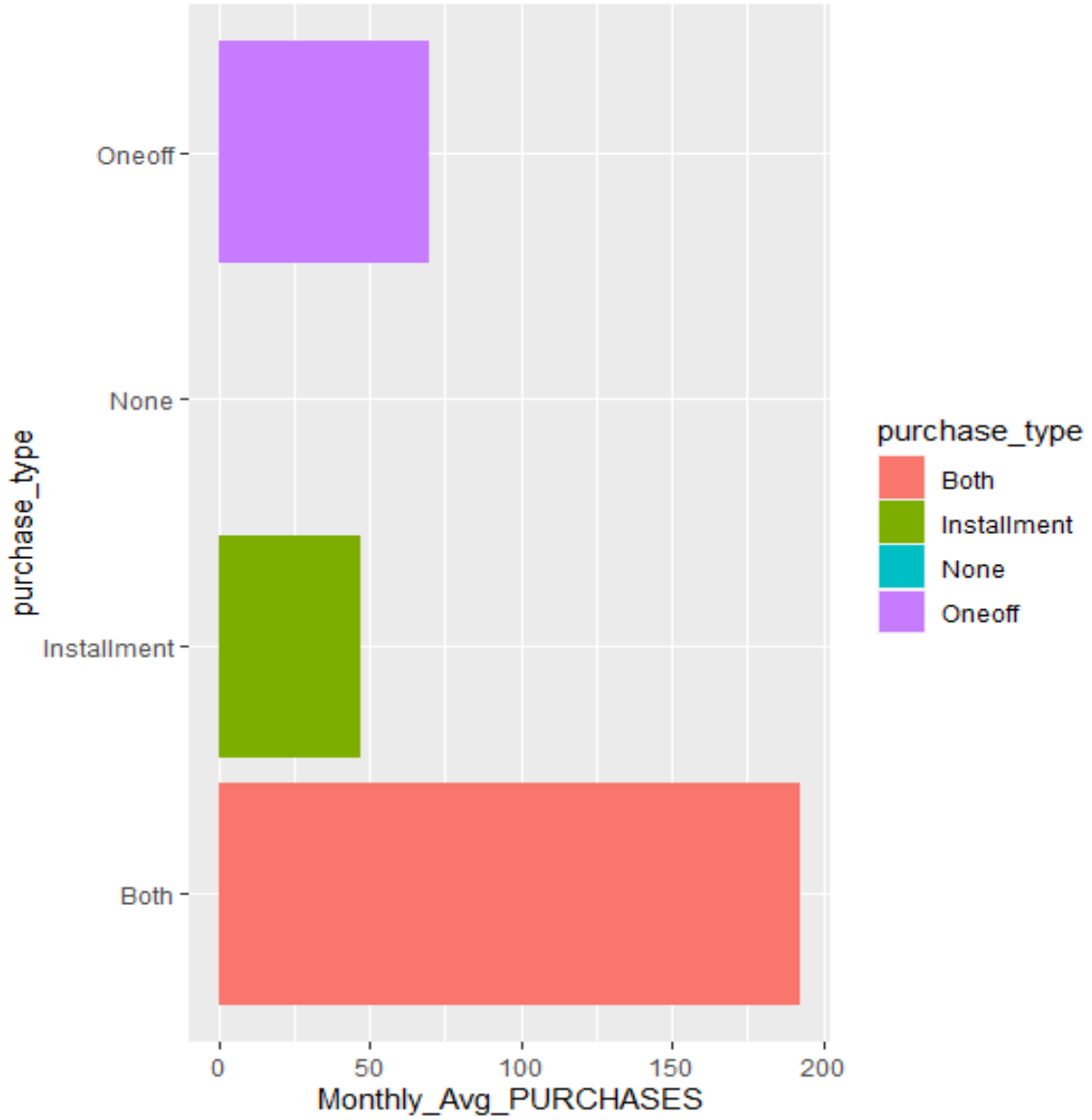
From here, I can see Customers who don't do either one-off or installment purchases, take more cash on advance**.**

(iii) Insight 3:



Here, 'limit_usage' represents the ratio of remaining balance & overall credit limit of a credit card. Lower value implies lower balance remaining in the card, which in turn means that customers are spending more. From the subplot, I can see that customers who're doing instalment purchase, are spending more compared to other groups.

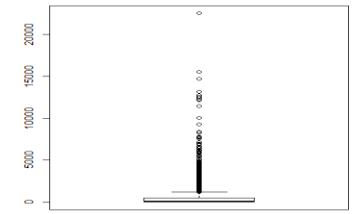
(iv) Insight 4:

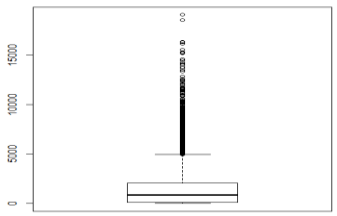


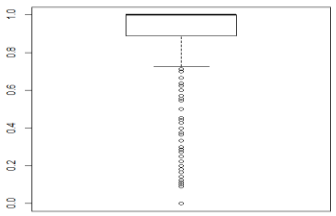
From here, I can see Customers who do instalment purchases, are doing less purchase compared to other purchase groups**.**

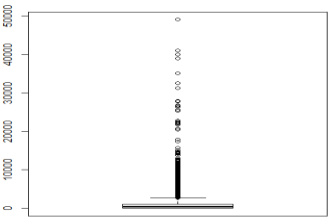
E) Outlier Treatment:

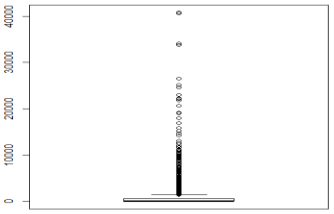
I've generated boxplots for all the numerical variables in the credit dataset to check if they contain outliers.

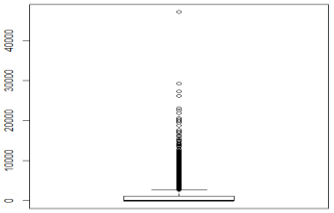


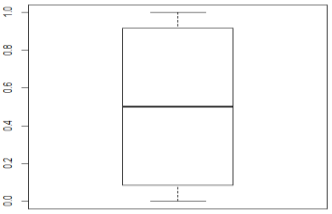


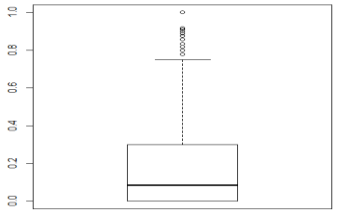


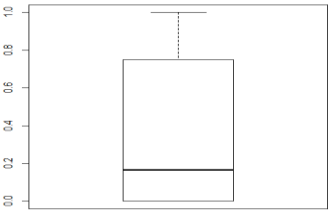


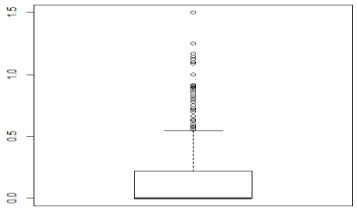


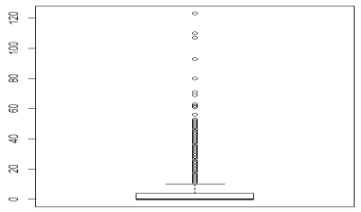


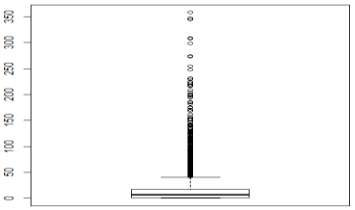


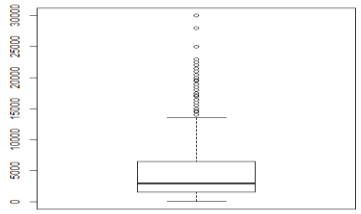


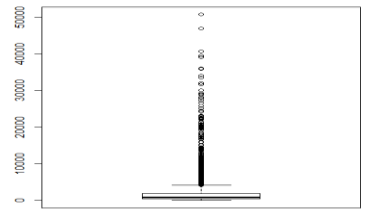


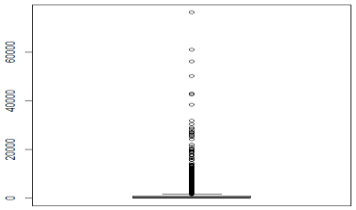


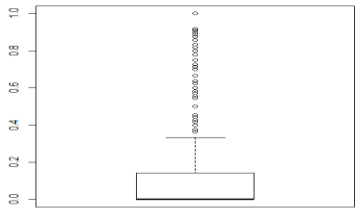


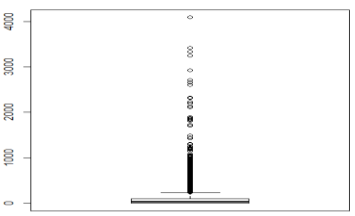


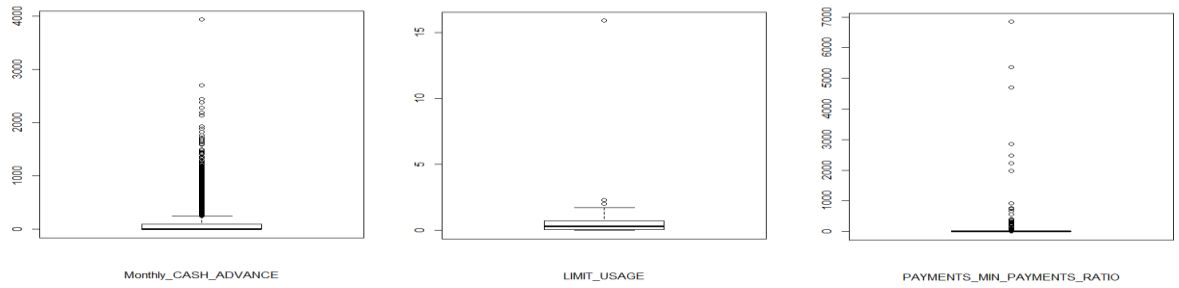












It's obvious, that these outliers were not made by human errors & they create a significant association, and in turn, will contribute to the end result. That's why, I did not drop or impute them (with central tendency or knn), I've used square- root transform to get rid of the extreme values of the dataset.

Before transforming the dataset, I've subsetted the KPI variables into 'cre_original' dataset, which will be required later to reveal the behavioural segments of credit card holders.

Then, I square-root transformed the numerical variables of the credit dataset & saved into the same credit dataset.

Then I've excluded those variables from cr_log, which have been used to derive KPIs, apart from ONEOFF_PURCHASES & INSTALLMENTS_PURCHASES, as these 2 variables don't have a linear relation with their derived KPI, purchase_type; & have saved the remaining variables into 'cr_pre' dataset.

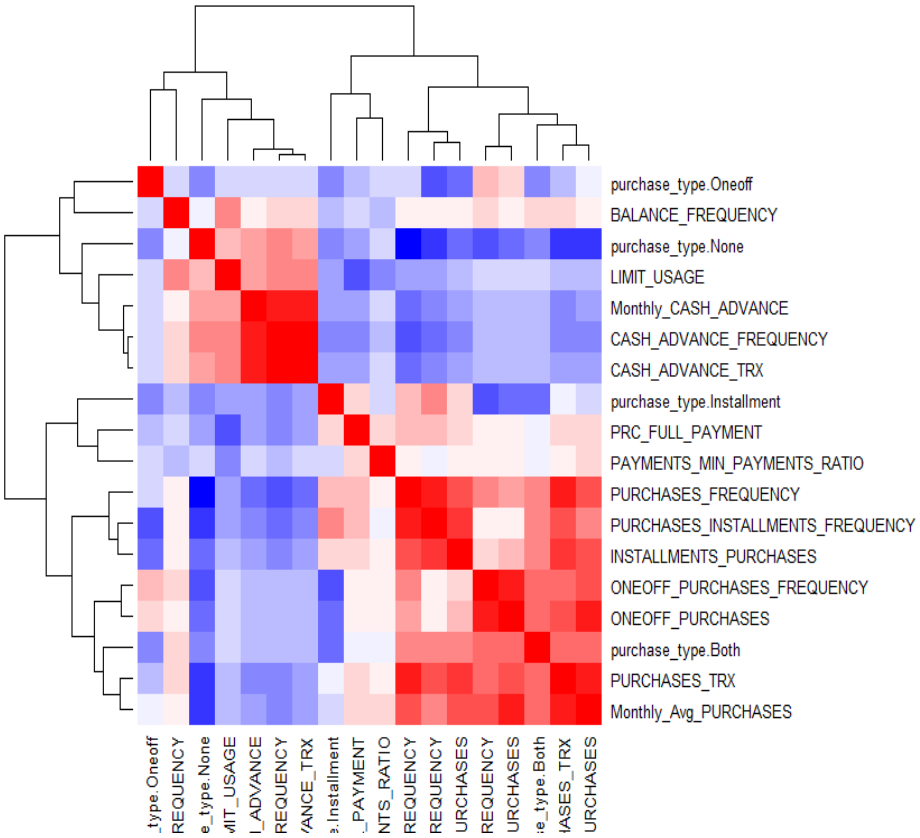
F) Data preparation for Machine Learning Algorithm

In 'cr_pre' dataset, I changed the purchase_type variable from character to factor. Then using 'dummy.data.frame' function from 'dummies' library, I converted the 'purchase_type' categorical variable into dummies, named 'purchase_type.Both', 'purchase_type.Installment', 'purchase_type.None', 'purchase_type.Oneoff' respectively; & saved the output variables in 'credit.new1' dataset. Then I converted these dummy variables from int type to float type.

After that, using the same process, I converted the 'purchase_type' variable in 'cre_origial' dataset from categorical to numeric.

G) Correlation Analysis

I used the 'cor' function to extract the correlation between variables in 'credit.new1' dataset & saved them in credit_cor object & also saved it to a csv file. Then I generated a heatmap to visualize the correlation between different variables in credit_cor.



From here, we can see that there's multicollinearity between different variables.

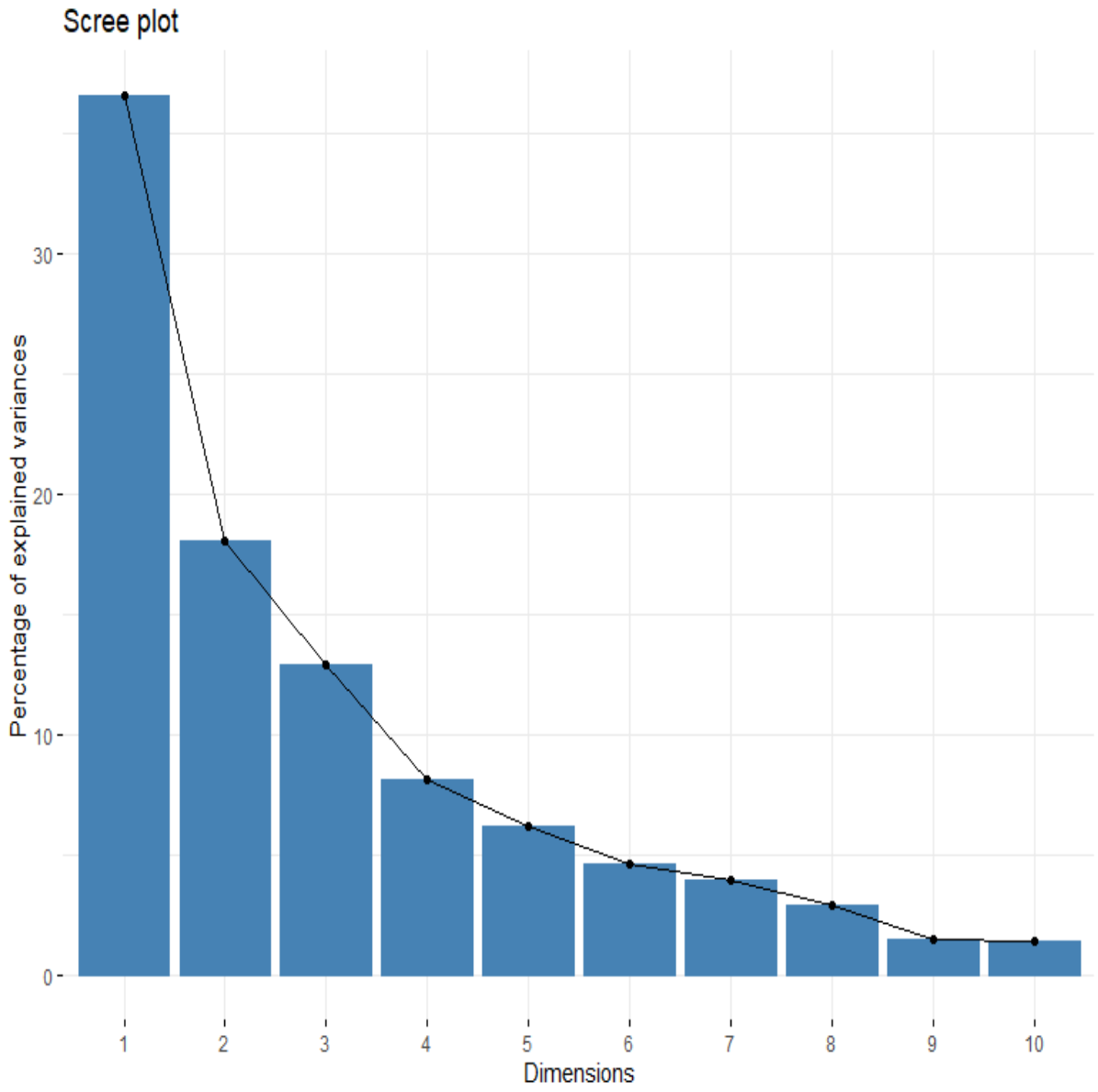
H) Dimensionality Reduction using PCA

I'm choosing to apply Principal Component Analysis (PCA) as the dimension reduction method as by using PCA I'll be able to identify the direction of highest to smallest variance; which in turn will enable me to drop the less variant features.

Before applying PCA, I'll standardize the data to put the data on the same scale & so that the data has zero mean & unit variance, which is important for the optimal performance of PCA. I've scaled the 'credit.new1' dataset using scale function & saved the output into 'cr_scaled' matrix. And, I've saved the same in a csv file named 'standardized data'.

Then, by using 'prcomp' function from 'factoextra' extra, I've performed PCA on 'cr_scaled' data set & saved the result in 'cred.pca' object.

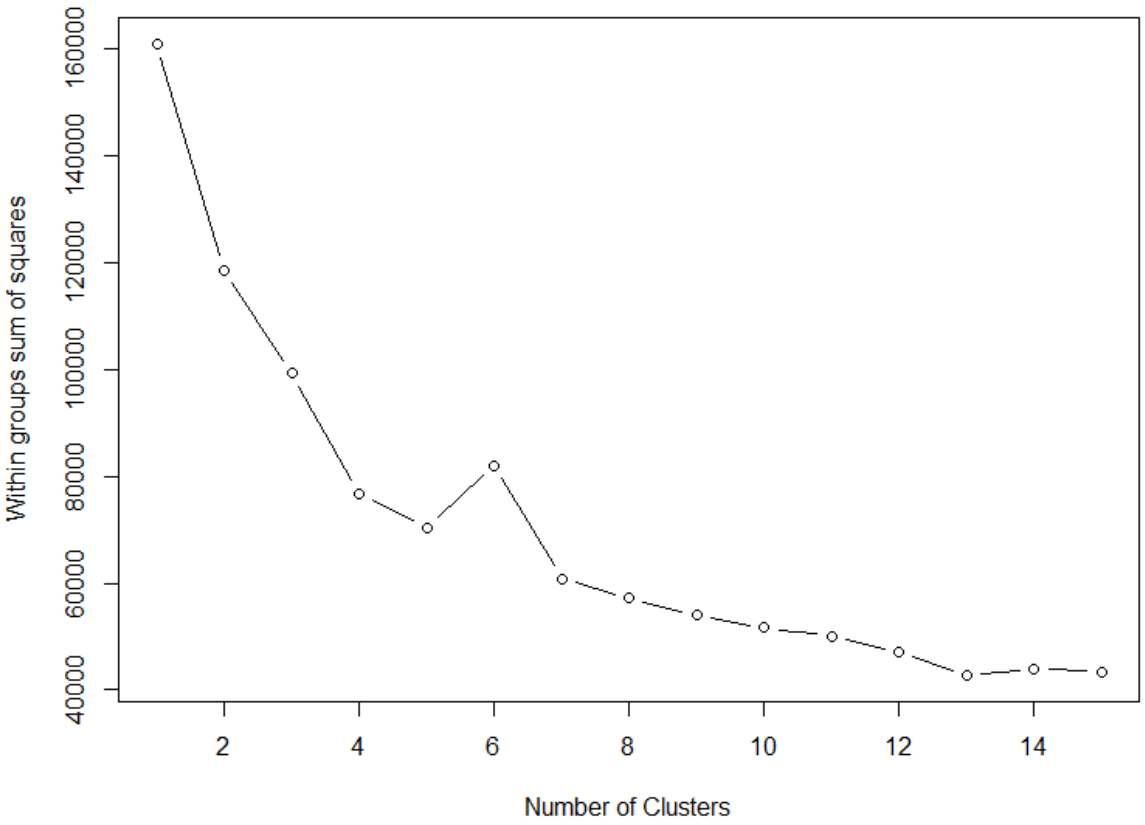
Then I used the ‘fviz_eig’ function on the ‘cred.pca’ object in order to plot the eigenvalues/ variances against the number of dimensions.



From here, we can see that 5 variables explain more than 85% of the variance (approximately), So, I selected 5 components. That’s how I can drop the other less variant variables. After that, I derived the rotation of ‘cred.pca’ into ’ cred_res’ matrix. Finally, I extracted the more varying 6 features from this into ‘cred_res_final’ dataframe, which represents the eigen vectors.

I) Clustering Algorithm

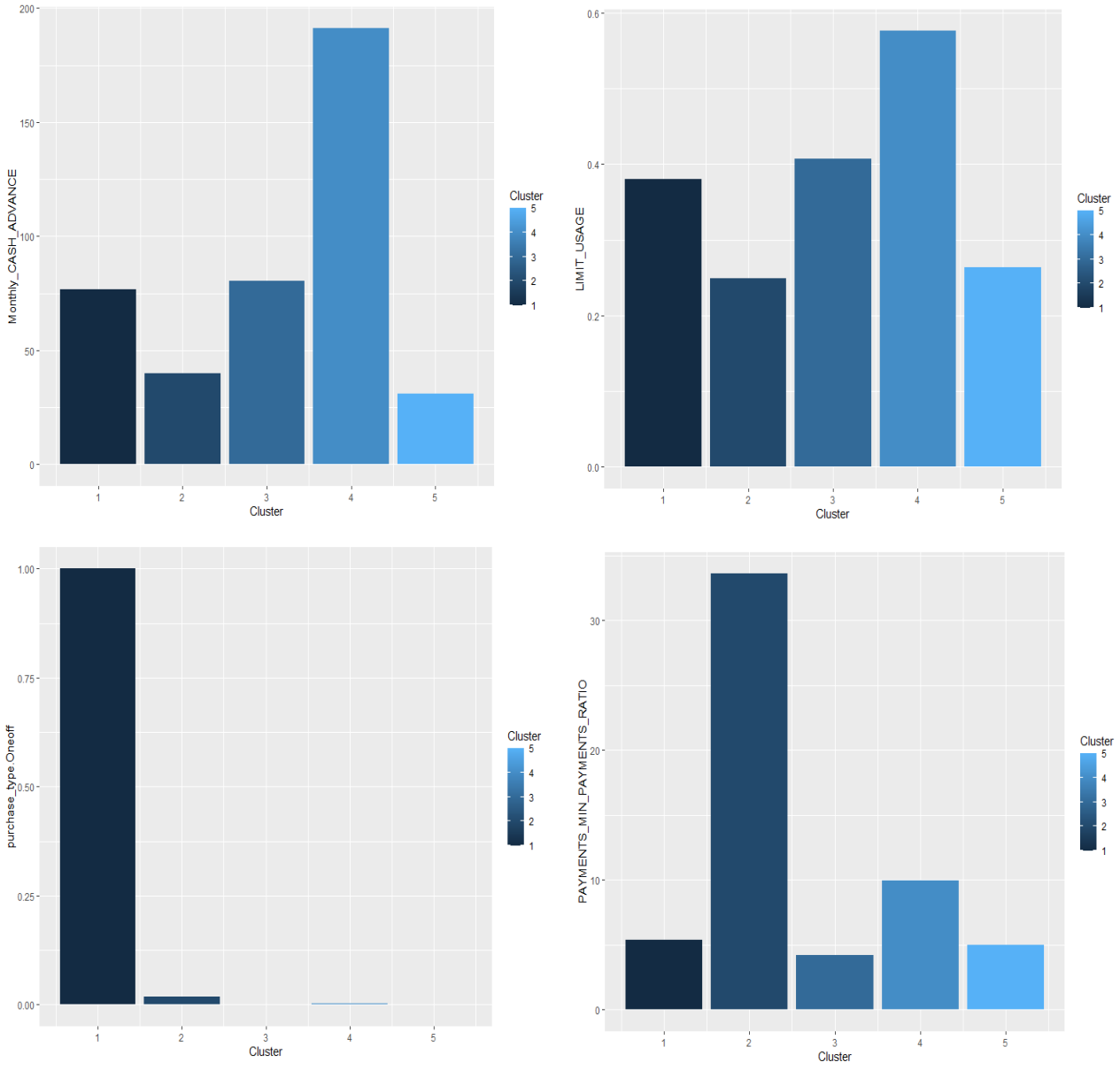
In order to do clustering, first I'll have to estimate optimum no. of clusters to be built. I've used 'within-cluster sum of squares' to determine the no. of clusters, which is a measure of the variability of the observations within each cluster. In general, a cluster that has a small sum of squares is more compact than a cluster that has a large sum of squares. Then, I've plotted "Number of Clusters" on x-axis & "Within groups sum of squares" on y-axis.

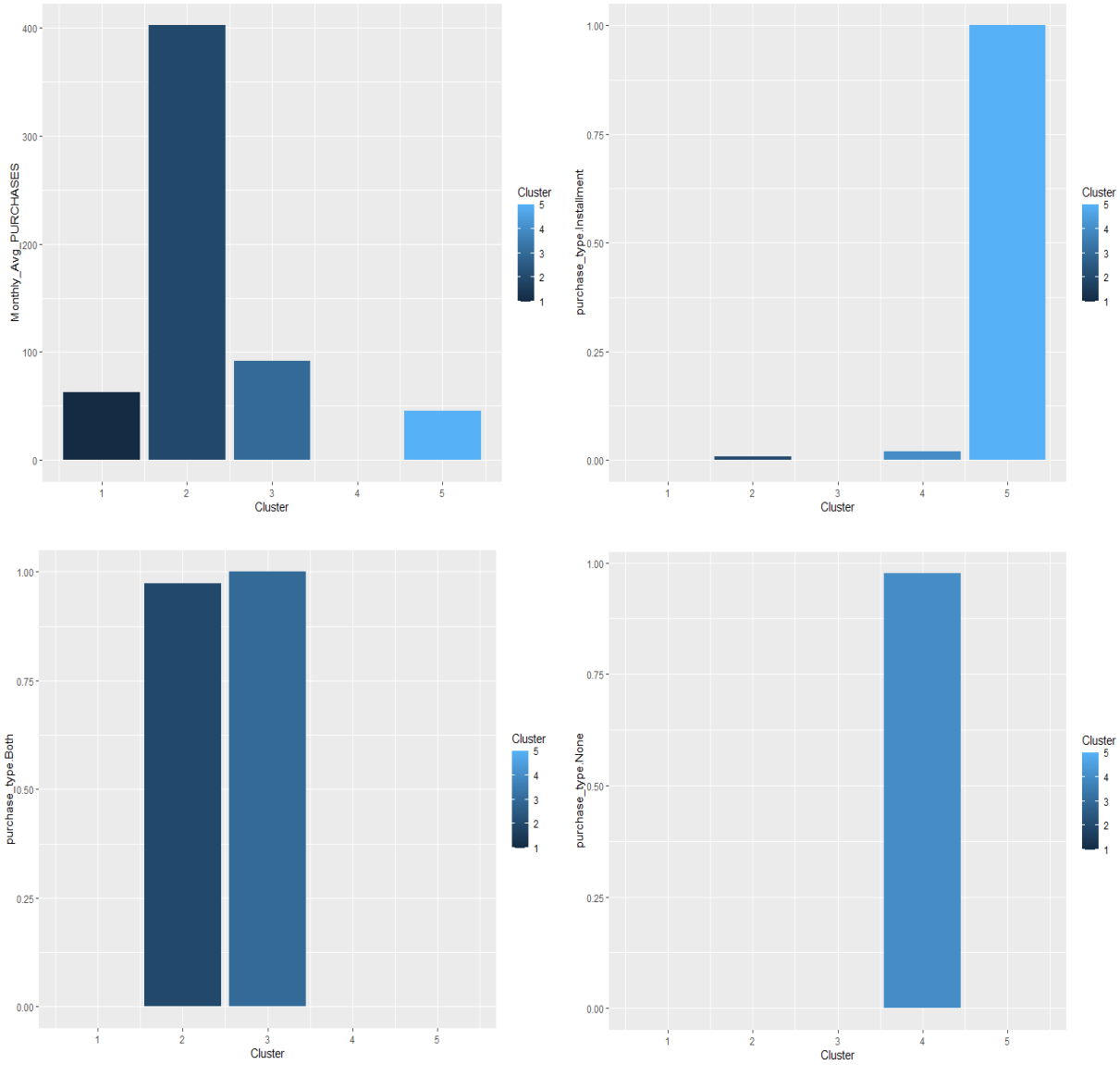


From here, I've taken no. of cluster as 5. Then, I've done clustering by using kmeans function on the 'cr_scaled' object & saved the clusters as 'clusters' object. After that, I've merged the names column in clusters(km_clust_5) with cre_original (non-transformed existing dataframe with KPI variables) & saved the result into 'credit_new2' dataset.

J) Behaviroul Analysis & Segmentation of credit card holders:

Using the aggregate function, I've taken the mean of all the variables in credit_new2 except the cluster variable, for all the different clusters & saved the results in 'cr_clust' dataset.
Then, by using ggplot2 library, I've plotted different bar charts with "Monthly_Avg_PURCHASES", "Monthly_CASH_ADVANCE", "LIMIT_USAGE", "PAYMENTS_MIN_PAYMENTS_RATIO", "purchase_type.Installment","purchase_type.Oneoff", "purchase_type.Both" & "purchase_type.None" on the y-axis and "cluster" on the x-axis.





Customers from cluster 1: They're taking medium advance cash from their card & doing only one-off purchases. Their average purchase using this credit card is less, they're spending medium amount & are not paying their dues in time.

Customers from cluster 2: They're taking less amount of advance cash from their card & doing both types of purchases. Their average purchase is highest among groups. They're spending highest amount & they're making high minimum payments.

Customers from cluster 3: They're taking medium advance cash from their card & doing both type of purchases. Their average purchase using this credit card is medium. They're spending medium amount on their card & are not paying their dues in time.

Customers from cluster 4: They're taking highest amount of advance cash from their card & not doing any purchases. They're spending least amount on their card & are making medium minimum payments compared to others.

Customers from cluster 5: They're taking least amount of advance cash from their card & doing only instalment purchases. They're doing less amount of purchase. They're spending high amount on their card & are not paying their dues in time.

K) Recommended Business Strategy

Here, I'll suggest business strategies for each of the customer groups. They are as follows-

(i) Cluster 1: As these customers are spending less amount & doing medium purchases, we can give them discount offers on purchases to encourage them to increase their purchases; & as they aren't paying their dues in time, we shouldn't increase their credit limit.

(ii) Cluster 2: As they're spending highest amount & doing highest amount of purchases, we can give them reward points, in order to encourage them to do more purchases. We can offer to increase their credit limit, which will hopefully increase their spending further. .

(iii) Cluster 3: As they're spending medium amount & doing medium amount of purchases, we can give them discount offers on purchases, which can increase their spending further. And as they aren't paying their dues in time, we shouldn't increase their credit limit.

(iv) Cluster 4: As they're not doing any purchase & spending least, we can give them discount offers, in order to encourage them to spend more. We should try to increase their spending before offering to increase their credit limit.

(v) Cluster 5: As they're spending high amount & doing less amount of purchases, we can give them discount offers on purchases, which can increase their spending further; & as they aren't paying their dues in time, we shouldn't increase their credit limit.