Masters Programmes: Assignment Cover Sheet

Student Number:	2028065, 5518354, 5532431, 5531616, 5523853, 5521398, 5583269
Module Code:	IB98D0
Module Title:	Advanced Data Analysis
Submission Deadline:	12:00 (UK time) Monday 18 March 2024
Date Submitted:	16/03/2024
Word Count:	1987
Number of Pages:	19
Question Attempted: *question number/title, or description of assignment)*	1
Have you used Artificial Intelligence (AI) in any part of this assignment?	NO

Academic Integrity Declaration

We're part of an academic community at Warwick. Whether studying, teaching, or researching, we're all taking part in an expert conversation which must meet standards of academic integrity. When we all meet these standards, we can take pride in our own academic achievements, as individuals and as an academic community.

Academic integrity means committing to honesty in academic work, giving credit where we've used others' ideas and being proud of our own achievements.

In submitting my work, I confirm that:

- I have read the guidance on academic integrity provided in the Student Handbook and understand the University regulations in relation to Academic Integrity. I am aware of the potential consequences of Academic Misconduct.
- I declare that the work is all my own, except where I have stated otherwise.
- No substantial part(s) of the work submitted here has also been submitted by me in other credit bearing assessments courses of study (other than in certain cases of a resubmission of a piece of work), and I acknowledge that if this has been done this may lead to an appropriate sanction.
- Where a generative Artificial Intelligence such as ChatGPT has been used I confirm I have abided by both the University guidance and specific requirements as set out in the Student Handbook and the Assessment brief. I have clearly acknowledged the use of any generative Artificial Intelligence in my submission, my reasoning for using it and which generative AI (or AIs) I have used. Except where indicated the work is otherwise entirely my own.
- I understand that should this piece of work raise concerns requiring investigation in relation to any of points above, it is possible that other work I have submitted for assessment will be checked, even if marks (provisional or confirmed) have been published.
- Where a proof-reader, paid or unpaid was used, I confirm that the proof-reader was made aware of and has complied with the University's proofreading policy.

Upon electronic submission of your assessment, you will be required to agree to the statements above

Table of Contents

1. Executive Summary

This report evaluates the use of analytics in addressing the challenges that the company faces in the banking sector. The first part contains details on how clustering analysis was performed following all stages to segment customers based on their profile characteristics. In the second section, we categorise customers into three groups in to provide them with insights and results regarding which groups the bank should target and how to offer new products to them. Customers in the first group have a high risk of default, which we can closely monitor and supplement with financial assistance. The next group is potential customers who have a smaller loan size, and a lower risk profile could be incentivized further by extending larger loan. Finally, customers with substantial loan amounts who have a low default risk are those to whom we can offer more personalised products.

2. Introduction

In response to the bank's challenges regarding loan approval efficiency, risk assessments, and customer satisfaction, we analysed company's loan data. Our objective was to offer personalized loan products, refine marketing strategies, and enhance customer support. This was achieved by conducting clustering analysis to classify customers into distinct groups based on their behaviour, enabling targeted strategies to meet bank's needs.

3. Data Preparation

3.1. Variable selection

We started by analysing the original loan data, issued through the period of 2012-2013, contained 50,000 observations and 53 variables. Subsequently, data were removed based on the following criteria, resulting in 14 variables remaining for analysis (Appendix 8.1):

1. Non-numerical data types, such as strings and dates, were removed.
2. Categorical data, including both ordinal and nominal variables, were eliminated.
3. Numerical data that exhibited significant skewness, with one value representing the majority of observations, were excluded.
4. Variables exhibiting a correlation coefficient of 1 with others were identified, and only one of these highly correlated variables was retained.
5. Variables with substantial number of missing values (NA > 5%) were also removed.

3.2. Data Cleaning

Following our initial analysis, we proceeded to conduct a random sample comprising 500 observations. The collected data was then normalized using Z-statistics, allowing for a standardized comparison across variables. Outliers were identified by removing values with Z-scores exceeding -4 or falling below -4. Subsequently, in-depth analysis was conducted on both datasets, containing 500 and 480 observations, respectively, post-outlier removal. Detailed insights stemming from this analysis can be found in Appendix 8.2.

4. Doing PCA and Factor Analysis (FA)

Next, we examined whether if that was any multicollinearity that could potentially impact our results. To assess this assumption, we conducted Kaiser-Meyer-Olkin (KMO) tests and Bartlett

tests to evaluate the data's adequacy for Principal Component Analysis (PCA) and Factor Analysis (FA). Our target was a KMO value of 0.5 or higher and a Bartlett test p-value below 0.05. Both datasets, with and without outliers, met these criteria, affirming the suitability of proceeding with both methods, as it's evident in Appendix 8.3.

Continuing our analysis, we performed PCA and FA on both datasets. In PCA, we evaluated the proportion of variance explained by each new variable to determine the number of principal components (PCs) to retain. Generally, we retain all PCs with an eigenvalue of 1 or higher. While the results between the two datasets were not markedly different, the interpretation favoured the dataset excluding outliers, as it tended to explain more variance (Appendix 8.4).

For Factor Analysis, we employed PC and Maximum Likelihood (ML) extraction methods both before and after outlier removal. Our evaluation utilised the correlation matrix, percentage of variance explained, and scree plot to ascertain the optimal number of factors to extract. Additionally, we considered factor loadings in our assessment. We then compared these approaches to determine the most suitable extraction method. The models examined were as follows:

Model A FA – 3 MLs extraction – No rotation.
Model B FA – 3 MLs extraction – Orthogonal.
Model C FA – 5 PCs extraction – Orthogonal.

Among these methods, we found that the results of the three methods yielded superior outcomes. Detailed findings in Appendix 8.5.

In Model A, ML1 represents the loan amount received by the customer, encompassing principal, interest, and instalments. ML2 indicates both the payment and interest rate, serving as indicators of the lender's risk assessment towards the customer. Similarly, in Model B, ML1 signifies the loan size and repayment ability, while ML2 directly depicts the interest rate and amount, reflecting the customer's default risk. ML3 does not signify any specific characteristic in Models A and B as it only explained the remaining variance.

In Model C, five components are extracted: RC1 represents both loan and payment size, RC2 denotes the number and amount of customers' debts and revolving balances, RC3 signifies interest and revolving utilities RC4 reflects the number of inquiries and captures the remaining variance of interest rate. Additionally, several components, including interest (RC3), income (RC5), and debt utilization (RC2) identified DTI ratios.

5. Doing Cluster Analysis

5.1. Analysis

The initial stages of our cluster analysis, encompassing problem definition and pre-analysis decisions regarding sample size, were meticulously executed in early stages (Appendix 8.6). However, considering the sensitivity of the clustering process to outliers, we utilized Mahalanobis distance to identify and remove observations with p-values less than 0.001, deemed outliers (Appendix 8.7). This process was repeated for all three models extracted before. Subsequently, we continued by verifying the required assumptions. Multicollinearity assumption was already addressed through meticulous Factor Analysis and Principal Component Analysis, ensuring robustness in the 500-sample size. Consequently, we moved on creating clusters for each of the three models: A, B, C. After the linkage methods were defined

and agglomerative coefficients were computed for each model, ultimately leading to Ward's method, we proceeded to determine the optimal number of clusters using hierarchical approach. This approach employs the gap statistic technique. Appendix 8.8 illustrates the resulting output.

In all three models, the optimal number of clusters was determined to be 3, while for Model C, both 3 and 5 clusters were considered optimal. However, since hierarchical method may lead to misleading results if there are any undesirable combinations, we combined it with non-hierarchical method and continued by employing k-means clustering to put the observations into a given number of clusters. The cluster plots for each model are provided in Appendix 8.9.

The examination of observation counts within each cluster across three models revealed that every group was sufficiently sampled. To evaluate the quality of clustering, Silhouette scores were also computed for three models. Appendix 8.10 shows the detailed results including the Silhouette score and the number of observations in each cluster. Based on the results, models A and B seem to have better-defined clustering structures with higher average silhouette widths compared to Model C. Further analysis, including validation, is required to enhance our confidence in these conclusions.

Next, we proceeded with internal validation for each model by randomly selecting 5 subsamples, each containing 100 observations. Subsequently, we performed clustering on these subsamples and allocated observations to the designated clusters based on previously k-means results. We separately calculated the average clustering accuracy of five sub-samples in each of the three models. The table below illustrates the conclusive findings.

Model	**Average Clustering Accuracy**
A	95.4%
B	95.4%
C1 (3 clusters)	74.6%
C2 (5 clusters)	86.2%

Figure 1: Validation Result for each Model

It is evident that both models A and B exhibit equally high average accuracy, surpassing that of model C. Nevertheless, considering our earlier observation regarding fewer cross-loadings in model B, it is inferred that model B is the most suitable choice for cluster analysis. The result shows that only 4.6% of observations are assigned to different clusters, indicating extremely high clustering stability. Figure 2 shows the clustering result of Model B.

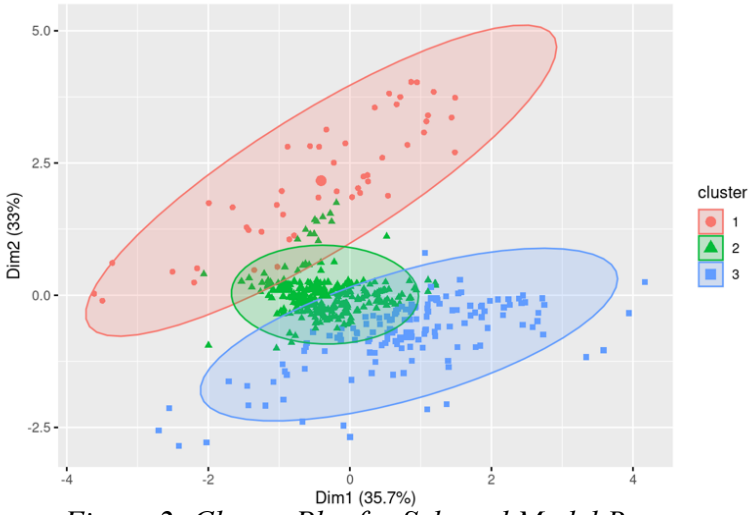


Figure 2: Cluster Plot for Selected Model B

Moreover, Appendix 8.11 shows the cluster centroids after performing Kmeans clustering for model B. The blue boxes highlight the highest values whereas the red ones the lowest ones. Based on that, cluster 1 has the highest value for ML2 (high default risk) and ML3, whereas cluster 2 has the lowest value for both ML1(small loan size and high repayment ability) and ML3. Cluster 3 represents a single case in each metric. To further understand customer profile in each cluster, we analysed characteristics from the original sample of 500 observations which included all 53 variables.

5.2. Interpretation

Our analysis aimed to uncover distinctive patterns among variables relevant to each cluster, excluding factors such as dates, addresses, and descriptions. Initial exploration involved plotting various variables, with detailed results provided in both Appendix 8.12.

Cluster 1 predominantly comprises customers seeking loans of lower grades (C-F) ranging from £10,000 to £27,000, primarily for debt consolidation and credit card payments. This cluster exhibits a significant proportion of charged-off loans (54%) and current loan statuses (33%), with approximately 65% of loans classified as bad.

In contrast, Cluster 2 customers typically request smaller loan amounts (below £15,000) with higher grades (A-D), with 65% of loans successfully paid off. Their loan purposes commonly include debt consolidation, credit cards, and home improvements. Additionally, this cluster has a stable financial profile, with only 9.4% of loans categorized as bad.

Cluster 3 customers seek loan amounts above £15,000, with instalments above £500. Despite a higher loan amount and instalments compared to Cluster 2, 35% of loans in this cluster are fully paid with zero recoveries. The primary loan purposes align with debt consolidation, credit cards, and home improvements. Further details regarding the characteristics of each cluster are provided in Appendix 8.13.

6. Recommendations

Cluster analysis provided insights into customer behaviour, facilitating the identification of potential initiatives to improve the bank's operations. Some of these initiatives include:

- **Loan Amount and Repayment Incentives:**

Implementing incentives to encourage customers in Clusters 2 and 3 to increase loan amounts could be advantageous due to their consistent loan repayment track record. This could generate additional interest income without elevating risk exposure while fostering customer loyalty, financial stability, and profitability.

- **Interest Rate Adjustments:**

A cautious increase in interest rates for cluster one customers is suggested to offset the risk. However, it's vital to maintain competitive rates to avoid customer discouragement, which could potentially exacerbate default rates. Conversely, for the more reliable Clusters 2 and 3, a slight reduction in interest rates may encourage larger loans without significantly impacting the bank's risk profile. To ensure any adjustments are both financially prudent and aligned with market conditions, a thorough cost-benefit analysis must be conducted, balancing the bank's risk management and growth objectives.

- **Customer Support and Debt Management:**

Prioritizing debt consolidation and loan restructuring, particularly for customers in Cluster 1, represents a strategic initiative. By assisting high-risk customers in managing their debt more effectively, the bank could increase the likelihood of recovering funds while simultaneously reducing the allowance for doubtful debts. As a result, the bank's net income might be positively influenced, mitigating the number of loans transitioning into the charged-off category.

- **Marketing Strategies and Product Offers:**

To enhance marketing strategies, there is a significant opportunity in launching targeted campaigns that showcase the personalized advantages of credit card products. Such campaigns could effectively attract both potential and existing customers by emphasizing the value and benefits of credit cards, tailored to their spending habits and repayment behaviour. Leveraging data analytics is crucial to ensure these efforts are relevant and effective, potentially increasing customer satisfaction and loyalty.

7. Conclusion

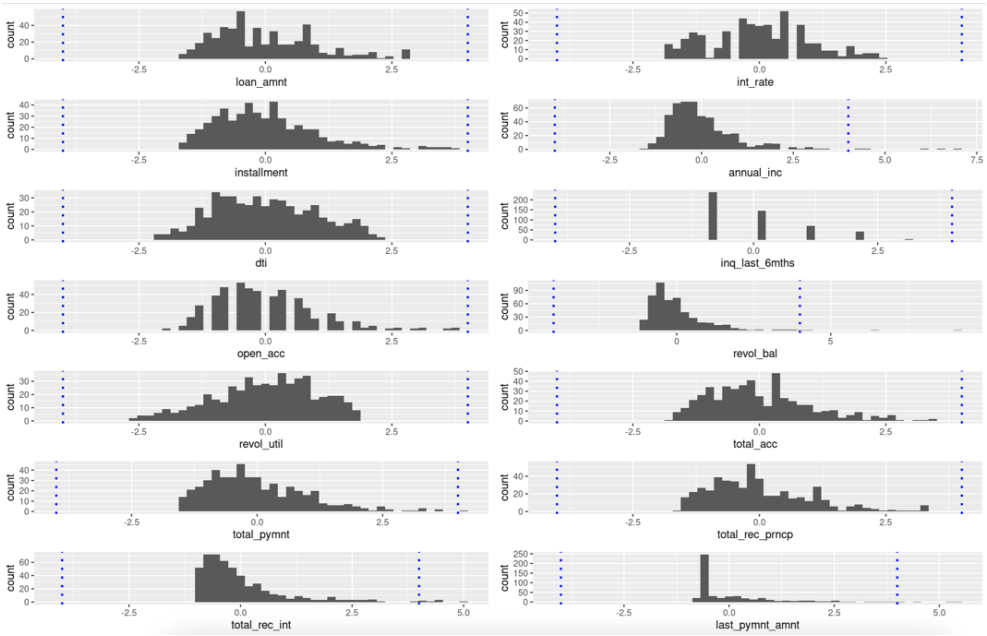
In conclusion, the cluster analysis of the company's loan data has provided valuable insights into borrower behaviour and characteristics, facilitating the identification of distinct segments within the customer base. Consequently, the bank can closely monitor high-risk default cases among customers in the first cluster, while incentivizing the remaining two groups to increase their loan amounts due to their financial stability and responsible behaviour. Implementing the recommendations derived from these insights can enhance loan portfolio management effectively. Therefore, adapting to a more data-driven approach will help improve the loan portfolio management and, in fact, deliver what we want to be - a more customer-centric organization.

8. Appendix

Appendix 8.1: Data Dictionary for Selected Variables

Variable Name	Description	Data Type
loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.	Numeric
int_rate	Interest Rate on the loan	Numeric
installment	The monthly payment owed by the borrower if the loan originates.	Numeric
annual_inc	The self-reported annual income provided by the borrower during registration.	Numeric
dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.	Numeric
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)	Numeric
open_acc	The number of open credit lines in the borrower's credit file.	Numeric
revol_bal	Total credit revolving balance	Numeric
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.	Numeric
total_acc	The total number of credit lines currently in the borrower's credit file	Numeric
total_pymnt	Payments received to date for total amount funded	Numeric
total_rec_prncp	Principal received to date	Numeric
total_rec_int	Interest received to date	Numeric
last_pymnt_amnt	Last total payment amount received	Numeric

Appendix 8.2: Plots for identifying potential outliers using Z- score



Appendix 8.3: Check multicollinearity assumption

```
KMO(z.df.corr)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = z.df.corr)
## Overall MSA =  0.67
## MSA for each item =
##       loan_amnt        int_rate     installment      annual_inc             dti
##            0.77            0.59            0.77            0.76            0.52
##  inq_last_6mths        open_acc       revol_bal      revol_util       total_acc
##            0.56            0.59            0.81            0.63            0.67
##     total_pymnt total_rec_prncp   total_rec_int last_pymnt_amnt
##            0.65            0.61            0.56            0.56
```

```
cortest.bartlett(z.df.corr, n=500)
```

```
## $chisq
## [1] 7304.685
##
## $p.value
## [1] 0
##
## $df
## [1] 91
```

```
KMO(z.df.outl.corr)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = z.df.outl.corr)
## Overall MSA =  0.66
## MSA for each item =
##       loan_amnt        int_rate     installment      annual_inc             dti
##            0.76            0.57            0.76            0.76            0.56
##  inq_last_6mths        open_acc       revol_bal      revol_util       total_acc
##            0.58            0.57            0.79            0.61            0.67
##     total_pymnt total_rec_prncp   total_rec_int last_pymnt_amnt
##            0.65            0.61            0.58            0.52
```

```
cortest.bartlett(z.df.outl.corr, n=480)
```

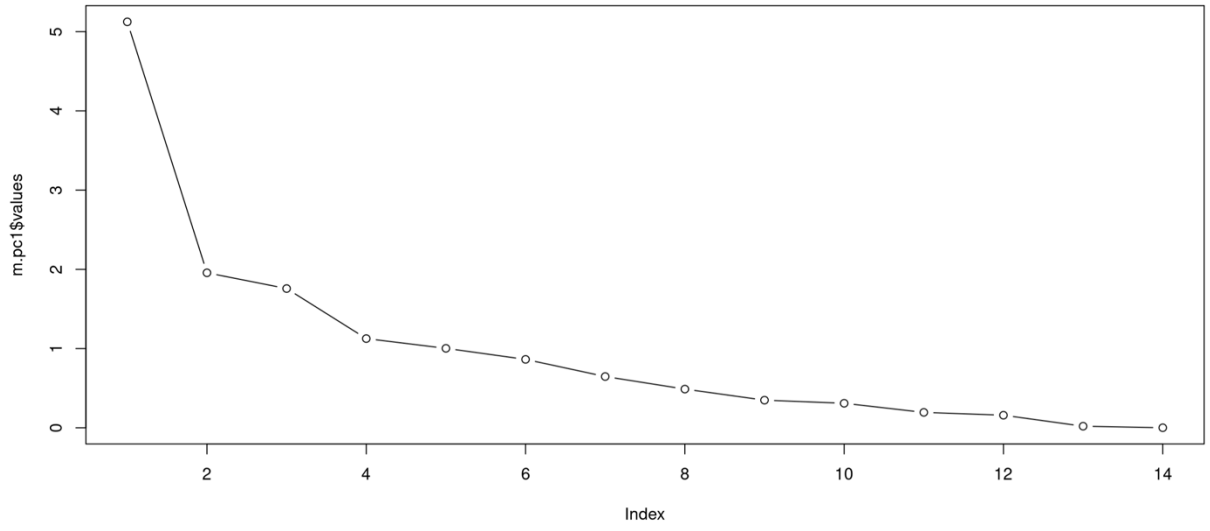
```
## $chisq
## [1] 6981.34
##
## $p.value
## [1] 0
##
## $df
## [1] 91
```

Appendix 8.4: PCA Results

Appendix 8.4.1: PCA for standardised data

```
Principal Components Analysis
Call: principal(r = z.df, nfactors = 14, rotate = "none", scores = TRUE,
    weights = TRUE)
```

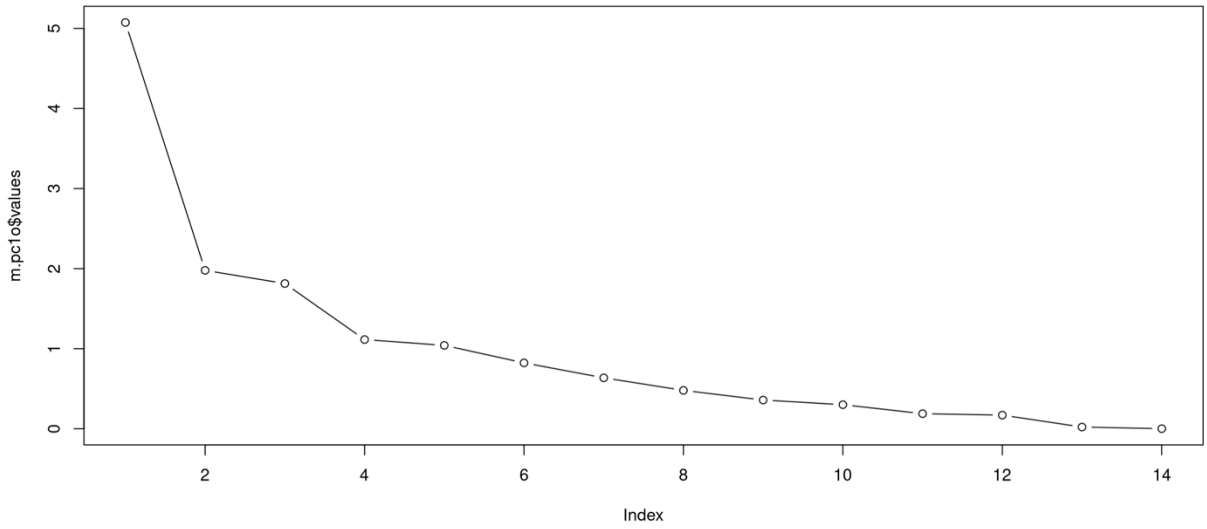
	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14
SS loadings	5.12	1.96	1.76	1.13	1.00	0.86	0.65	0.49	0.35	0.31	0.19	0.16	0.02	0
Proportion Var	0.37	0.14	0.13	0.08	0.07	0.06	0.05	0.03	0.02	0.02	0.01	0.01	0.00	0
Cumulative Var	0.37	0.51	0.63	0.71	0.78	0.85	0.89	0.93	0.95	0.97	0.99	1.00	1.00	1
Proportion Explained	0.37	0.14	0.13	0.08	0.07	0.06	0.05	0.03	0.02	0.02	0.01	0.01	0.00	0
Cumulative Proportion	0.37	0.51	0.63	0.71	0.78	0.85	0.89	0.93	0.95	0.97	0.99	1.00	1.00	1



Appendix 8.4.2: PCA for standardised data after removing outliers

```
Principal Components Analysis
Call: principal(r = z.outl.df, nfactors = 14, rotate = "none", scores = TRUE,
    weights = TRUE)
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14
SS loadings	5.07	1.98	1.81	1.11	1.04	0.82	0.64	0.48	0.36	0.30	0.19	0.17	0.02	0
Proportion Var	0.36	0.14	0.13	0.08	0.07	0.06	0.05	0.03	0.03	0.02	0.01	0.01	0.00	0
Cumulative Var	0.36	0.50	0.63	0.71	0.79	0.85	0.89	0.93	0.95	0.97	0.99	1.00	1.00	1
Proportion Explained	0.36	0.14	0.13	0.08	0.07	0.06	0.05	0.03	0.03	0.02	0.01	0.01	0.00	0
Cumulative Proportion	0.36	0.50	0.63	0.71	0.79	0.85	0.89	0.93	0.95	0.97	0.99	1.00	1.00	1

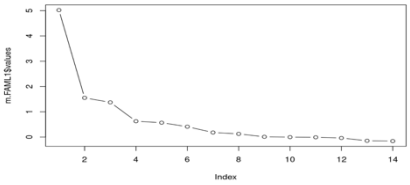


Appendix 8.5: Factor Analysis results

Appendix 8.5.1: FA – ML extraction – No rotation (Model A)

```
Call: fa(r = z.df, nfactors = 14, n.obs = 500, rotate = "none", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix

                       ML1  ML4  ML2  ML5  ML6  ML3  ML7  ML8  ML9 ML10 ML11 ML12 ML13 ML14
SS loadings           4.69 1.34 1.21 0.80 0.48 0.43 0.36 0.21 0.00 0.00 0.00 0.00 0.00 0.00
Proportion Var        0.34 0.10 0.09 0.06 0.03 0.03 0.03 0.01 0.00 0.00 0.00 0.00 0.00 0.00
Cumulative Var        0.34 0.43 0.52 0.57 0.61 0.64 0.66 0.68 0.68 0.68 0.68 0.68 0.68 0.68
Proportion Explained  0.49 0.14 0.13 0.08 0.05 0.04 0.04 0.02 0.00 0.00 0.00 0.00 0.00 0.00
Cumulative Proportion 0.49 0.63 0.76 0.85 0.90 0.94 0.98 1.00 1.00 1.00 1.00 1.00 1.00 1.00
```

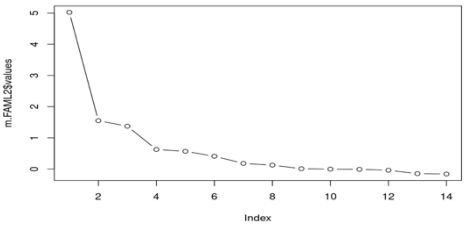


	item	ML1	ML4	ML2	ML5	ML6	ML3	ML7	ML8	ML9	ML10	ML11	ML12	ML13	ML14	h2	u2	com
	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<dbl>	<dbl>	<dbl>
total_pymnt	11	1.00														0.9971658	0.002834170	1.006755
total_rec_prncp	12	0.97														0.9962081	0.003791869	1.133966
installment	3	0.91														0.9674917	0.032508301	1.313022
loan_amnt	1	0.90														0.9701163	0.029883684	1.384764
last_pymnt_amnt	14	0.47														0.6126409	0.387359061	4.567315
annual_inc	4															0.5565568	0.443443152	5.248057
total_acc	10		0.71													0.5738123	0.426187719	1.320852
open_acc	7		0.69													0.5860263	0.413973677	1.477060
total_rec_int	13	0.69		0.71												0.9860684	0.013931562	2.036315
int_rate	2			0.58	0.48											0.7057926	0.294207392	2.907796
revol_util	9				0.43											0.5087220	0.491278019	3.781239
dti	5															0.4071697	0.592830315	3.406181
revol_bal	8															0.4554956	0.544504401	4.694645
inq_last_6mths	6															0.1891966	0.810803427	3.334258

Appendix 8.5.2: FA – ML extraction – Oblique rotation

```
Call: fa(r = z.df, nfactors = 14, n.obs = 500, rotate = "oblimin",
    fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix

                       ML1  ML3  ML4  ML2  ML6  ML8  ML7  ML5 ML11 ML12 ML13 ML14  ML9 ML10
SS loadings           2.24 1.57 1.48 1.39 0.94 0.74 0.67 0.47 0.00 0.00 0.00 0.00 0.00 0.00
Proportion Var        0.16 0.11 0.11 0.10 0.07 0.05 0.05 0.03 0.00 0.00 0.00 0.00 0.00 0.00
Cumulative Var        0.16 0.27 0.38 0.48 0.54 0.60 0.65 0.68 0.68 0.68 0.68 0.68 0.68 0.68
Proportion Explained  0.24 0.17 0.16 0.15 0.10 0.08 0.07 0.05 0.00 0.00 0.00 0.00 0.00 0.00
Cumulative Proportion 0.24 0.40 0.56 0.70 0.80 0.88 0.95 1.00 1.00 1.00 1.00 1.00 1.00 1.00
```

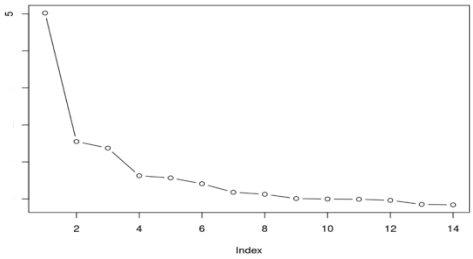


	item	ML1	ML3	ML4	ML2	ML6	ML8	ML7	ML5	ML11	ML12	ML13	ML14	ML9	ML10	h2	u2	com
	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<dbl>	<dbl>	<dbl>
installment	3	0.95														0.9674917	0.032508301	1.051629
loan_amnt	1	0.92														0.9701163	0.029883684	1.052687
total_rec_prncp	12		0.82													0.9962081	0.003791869	1.090420
total_pymnt	11		0.63													0.9971658	0.002834170	1.629906
open_acc	7			0.77												0.5860263	0.413973677	1.071341
total_acc	10			0.74												0.5738123	0.426187719	1.092655
total_rec_int	13				0.93											0.9860684	0.013931562	1.025882
int_rate	2															0.7057926	0.294207392	4.361158
revol_util	9					0.69										0.5087220	0.491278019	1.131250
revol_bal	8					0.41										0.4554956	0.544504401	3.040734
last_pymnt_amnt	14						0.73									0.6126409	0.387359061	1.078037
annual_inc	4							0.53								0.5565568	0.443443152	1.935541
dti	5							−0.52								0.4071697	0.592830315	2.067563
inq_last_6mths	6								0.43							0.1891966	0.810803427	2.000466

Appendix 8.5.3: FA – ML extraction – Orthogonal rotation (Model B)

```
Call: fa(r = z.df, nfactors = 14, n.obs = 500, rotate = "varimax",
    fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
```

	ML1	ML4	ML6	ML2	ML5	ML7	ML8	ML3	ML10	ML11	ML9	ML12	ML14	ML13
SS loadings	4.14	1.61	1.03	0.76	0.74	0.63	0.45	0.16	0.00	0.00	0.00	0.00	0.00	0.00
Proportion Var	0.30	0.11	0.07	0.05	0.05	0.05	0.03	0.01	0.00	0.00	0.00	0.00	0.00	0.00
Cumulative Var	0.30	0.41	0.48	0.54	0.59	0.64	0.67	0.68	0.68	0.68	0.68	0.68	0.68	0.68
Proportion Explained	0.44	0.17	0.11	0.08	0.08	0.07	0.05	0.02	0.00	0.00	0.00	0.00	0.00	0.00
Cumulative Proportion	0.44	0.60	0.71	0.79	0.87	0.94	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00

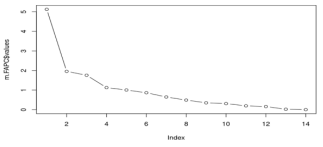


	item	ML1	ML4	ML6	ML2	ML5	ML7	ML8	ML3	ML10	ML11	ML9	ML12	ML14	ML13	h2	u2	com
	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<dbl>	<dbl>	<dbl>
installment	3	0.94														0.9674917	0.032508301	1.190878
total_pymnt	11	0.93														0.9971658	0.002834170	1.332702
loan_amnt	1	0.91														0.9701163	0.029883684	1.348417
total_rec_prncp	12	0.89														0.9962081	0.003791869	1.527733
total_acc	10		0.73													0.5738123	0.426187719	1.175159
open_acc	7		0.71													0.5860263	0.413973677	1.349054
annual_inc	4		0.46					-0.45								0.5565568	0.443443152	3.042070
revol_bal	8		0.46													0.4554956	0.544504401	3.009008
revol_util	9			0.68												0.5087220	0.491278019	1.230900
int_rate	2			0.49	0.45											0.7057926	0.294207392	4.002621
total_rec_int	13	0.65			0.68											0.9860684	0.013931562	2.461056
last_pymnt_amnt	14					0.72										0.6126409	0.387359061	1.379518
dti	5							0.56								0.4071697	0.592830315	1.599094
inq_last_6mths	6								0.41							0.1891966	0.810803427	1.203243

Appendix 8.5.4: FA – PC extraction – Oblique Rotation

```
Call: principal(r = z.df, nfactors = 14, rotate = "oblimin")
Standardized loadings (pattern matrix) based upon correlation matrix
```

	TC1	TC12	TC7	TC5	TC8	TC2	TC9	TC3	TC10	TC4	TC6	TC11	TC13	TC14
SS loadings	1.94	1.88	1.02	1.01	1.01	1.00	1.01	1.00	1.00	1.00	1.01	1.07	0.04	0.01
Proportion Var	0.14	0.13	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.08	0.00	0.00
Cumulative Var	0.14	0.27	0.35	0.42	0.49	0.56	0.63	0.71	0.78	0.85	0.92	1.00	1.00	1.00
Proportion Explained	0.14	0.13	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.08	0.00	0.00
Cumulative Proportion	0.14	0.27	0.35	0.42	0.49	0.56	0.63	0.71	0.78	0.85	0.92	1.00	1.00	1.00

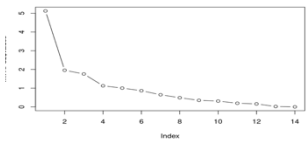


	item	TC1	TC12	TC7	TC5	TC8	TC2	TC9	TC3	TC10	TC4	TC6	TC11	TC13	TC14	h2	u2	com
	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<dbl>	<dbl>	<dbl>
installment	3	0.99														1	-1.776357e-15	1.024863
loan_amnt	1	0.91														1	3.330669e-16	1.053358
total_rec_prncp	12		1.03													1	-1.332268e-15	1.010860
total_pymnt	11		0.84													1	-1.998401e-15	1.079332
int_rate	2			1.00												1	6.661338e-16	1.000076
annual_inc	4				1.00											1	0.000000e+00	1.000011
revol_bal	8					1.00										1	4.440892e-16	1.000007
total_acc	10						1.00									1	-4.440892e-16	1.000005
revol_util	9							1.00								1	1.110223e-15	1.000001
dti	5								1							1	0.000000e+00	1.000002
open_acc	7									1.00						1	-1.110223e-15	1.000003
inq_last_6mths	6										1					1	-4.440892e-16	1.000002
last_pymnt_amnt	14											1.00				1	-8.881784e-16	1.000031
total_rec_int	13												0.99			1	-1.332268e-15	1.000330

Appendix 8.5.5: FA- PC extraction – Orthogonal Rotation (Model C)

```
Call: principal(r = z.df, nfactors = 14, rotate = "quartimax")
Standardized loadings (pattern matrix) based upon correlation matrix
```

	RC1	RC7	RC5	RC10	RC4	RC6	RC3	RC2	RC8	RC9	RC11	RC12	RC13	RC14
SS loadings	4.59	1.12	1.03	1.01	1.01	0.98	0.98	0.97	0.93	0.85	0.32	0.18	0.02	0
Proportion Var	0.33	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.02	0.01	0.00	0
Cumulative Var	0.33	0.41	0.48	0.55	0.63	0.70	0.77	0.84	0.90	0.96	0.99	1.00	1.00	1
Proportion Explained	0.33	0.08	0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.06	0.02	0.01	0.00	0
Cumulative Proportion	0.33	0.41	0.48	0.55	0.63	0.70	0.77	0.84	0.90	0.96	0.99	1.00	1.00	1



	item	RC1	RC7	RC5	RC10	RC4	RC6	RC3	RC2	RC8	RC9	RC11	RC12	RC13	RC14	h2	u2	com
	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<S3: AsIs>	<dbl>	<dbl>	<dbl>
total_pymnt	11	0.97														1	-1.998401e-15	1.121749
installment	3	0.96														1	-1.776357e-15	1.152698
loan_amnt	1	0.95														1	3.330669e-16	1.211056
total_rec_prncp	12	0.92														1	-1.332268e-15	1.409666
total_rec_int	13	0.72										0.53				1	-1.332268e-15	2.682837
int_rate	2		0.92													1	6.661338e-16	1.383730
dti	5			0.97												1	0.000000e+00	1.139224
open_acc	7				0.92											1	-1.110223e-15	1.376185
inq_last_6mths	6					0.99										1	-4.440892e-16	1.039657
last_pymnt_amnt	14						0.94									1	-8.881784e-16	1.279384
revol_util	9							0.93								1	1.110223e-15	1.327681
total_acc	10								0.91							1	-4.440892e-16	1.462275
revol_bal	8									0.91						1	4.440892e-16	1.415952
annual_inc	4										0.87					1	0.000000e+00	1.662765

Appendix 8.6: Assumption Check for Clustering (no multicollinearity)

```
# Check Correlation Matrix
lowerCor(CA.df.FAMLa)
```

```
##     ML1   ML2   ML3
## ML1  1.00
## ML2 -0.20  1.00
## ML3  0.01 -0.04  1.00
```

```
lowerCor(CA.df.FAMLb)
```

```
##     ML1   ML2   ML3
## ML1  1.00
## ML2 -0.05  1.00
## ML3  0.01 -0.04  1.00
```

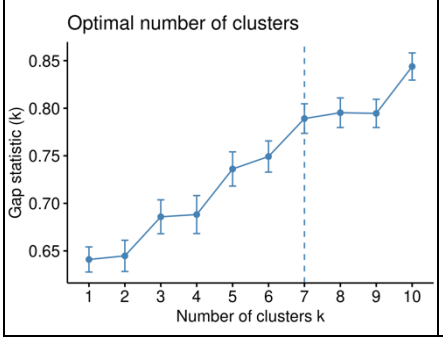
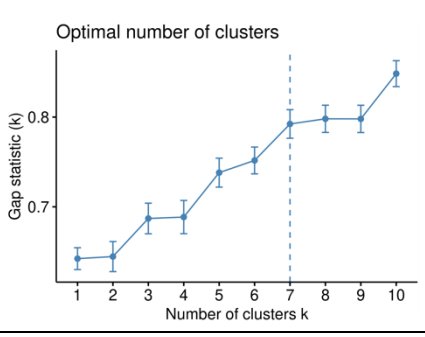
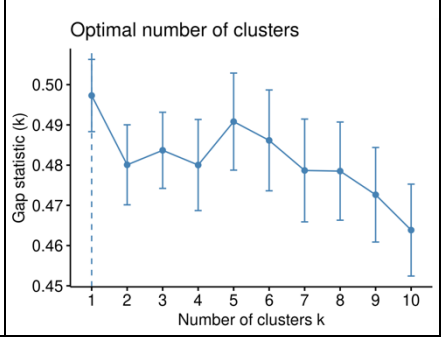
```
lowerCor(CA.df.FAPCc)
```

```
##     RC1   RC2   RC3   RC5   RC4
## RC1  1.00
## RC2 -0.01  1.00
## RC3  0.04  0.01  1.00
## RC5  0.01  0.12  0.05  1.00
## RC4 -0.03  0.06  0.05 -0.05  1.00
```

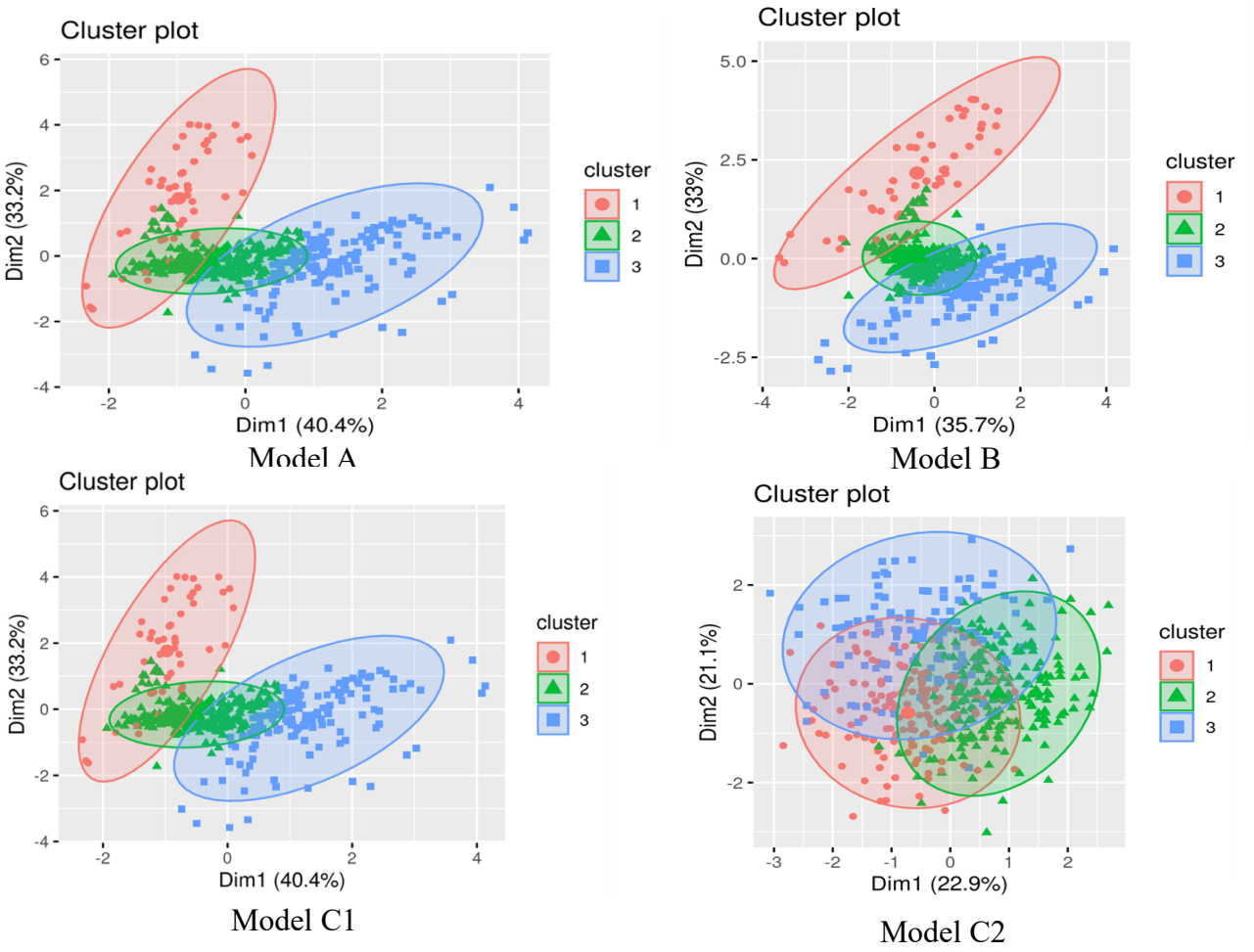
Appendix 8.7: Outliers in three models where Mahalanobis P Value less than 0.001

Model	Number of outliers removed
A	22
B	22
C	20

Appendix 8.8: Optimal number of Clusters for each Model

Model A	Model B	Model C
		

Appendix 8.9: Cluster Plot for each Model



Appendix 8.10: Table of Summary Results from Kmeans Clustering for all three models

Model	Optimal Number of Clusters	Number of Observations in each cluster	Average Silhouette Width
A	3	Cluster 1: 48	0.44
		Cluster 2: 286	
		Cluster 3: 144	
B	3	Cluster 1: 48	0.44
		Cluster 2: 286	
		Cluster 3: 144	
C	3	Cluster 1: 148	0.16
		Cluster 2: 208	
		Cluster 3: 129	
	5	Cluster 1: 116	0.17
		Cluster 2: 88	
		Cluster 3: 56	
		Cluster 4: 79	

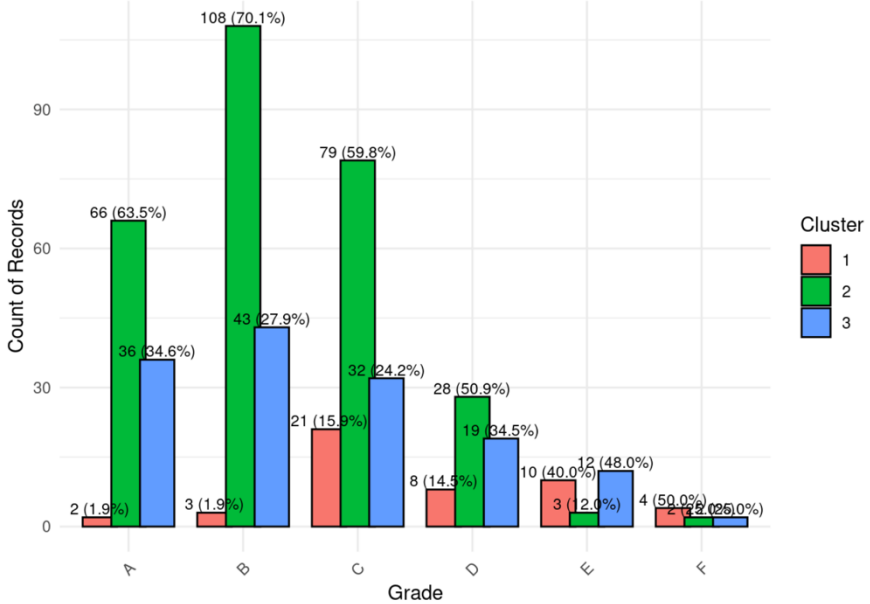
Appendix 8.11: Cluster centroids after performing Kmeans clustering in model B

```
## K-means clustering with 3 clusters of sizes 48, 286, 144
##
## Cluster means:
##          ML1        ML2        ML3
## 1 -0.5243660  1.1142198  1.5407388
## 2 -0.5603998 -0.2213520 -0.3446994
## 3  1.1670471 -0.2298315 -0.1219394
##
```

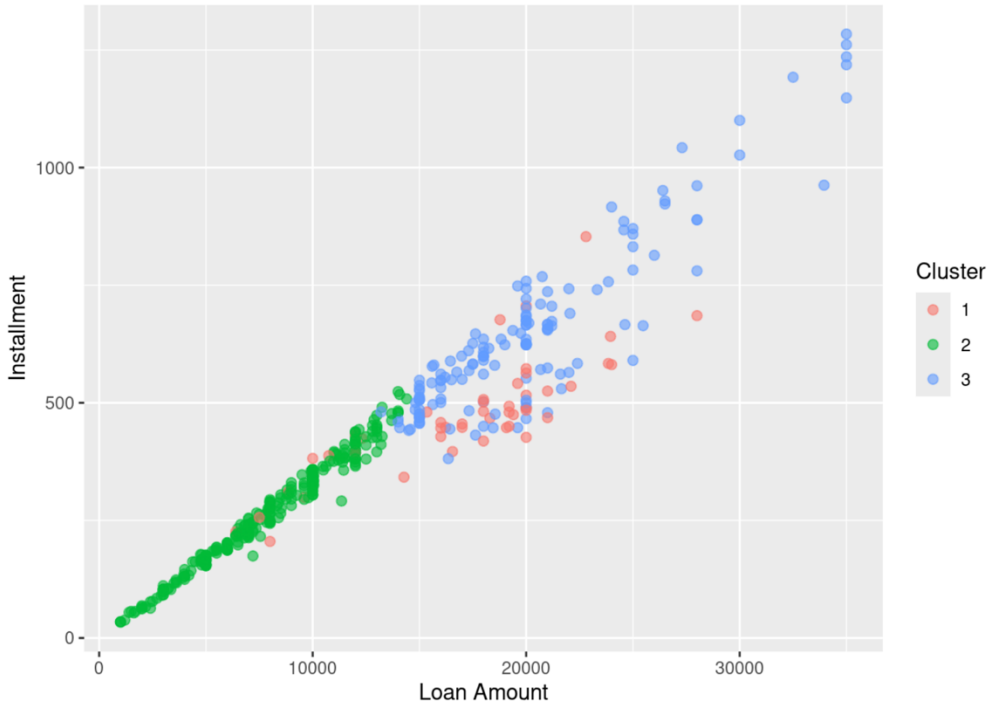
Appendix 8.12: Graph exploration for various variables

Appendix 8.12.1: The Loan Amount and Interest Rate of three clusters

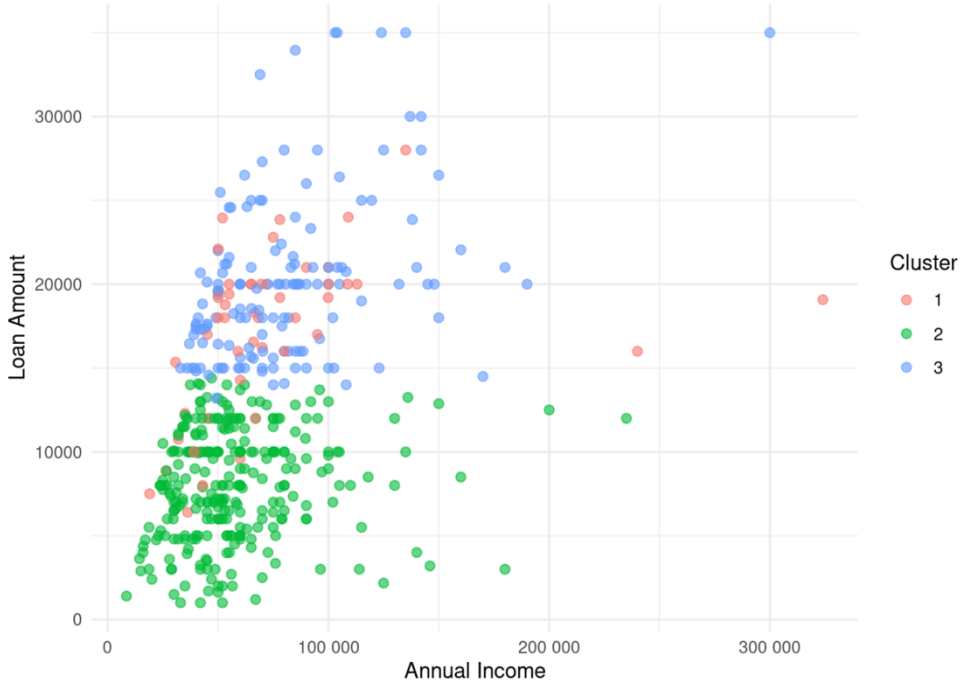
Appendix 8.12.2: The Grade and Loan Amount of three clusters



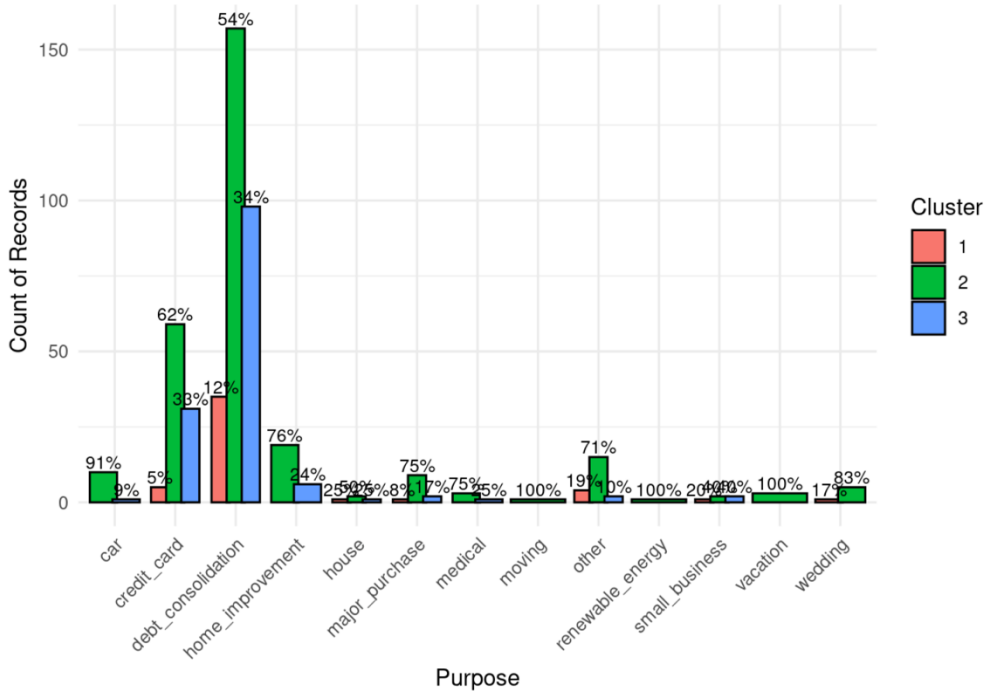
Appendix 8.12.3: The Instalment and Loan Amount of three clusters



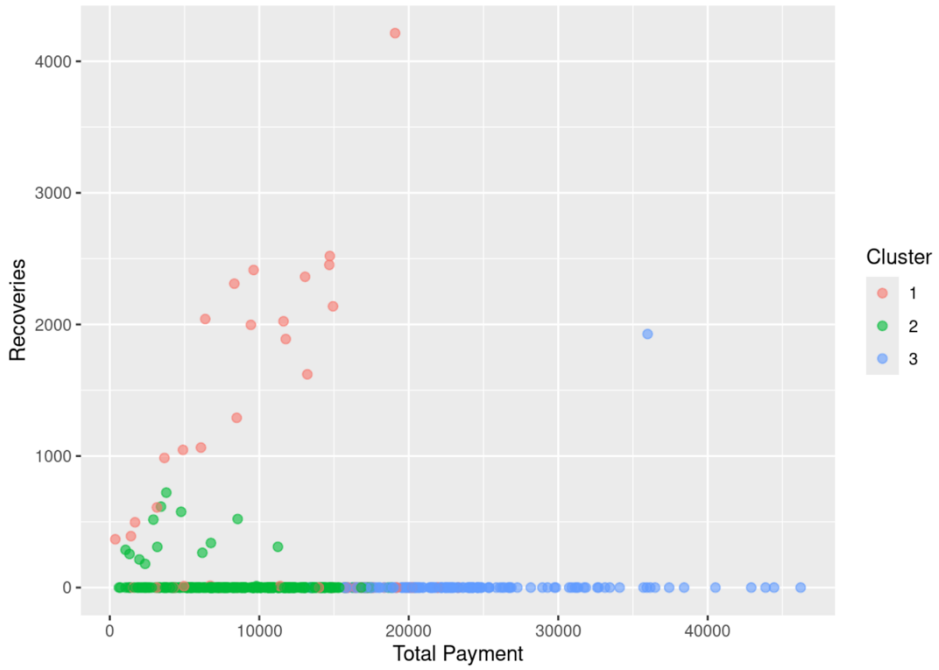
Appendix 8.12.4: The Annual Income and Loan Amount of three clusters



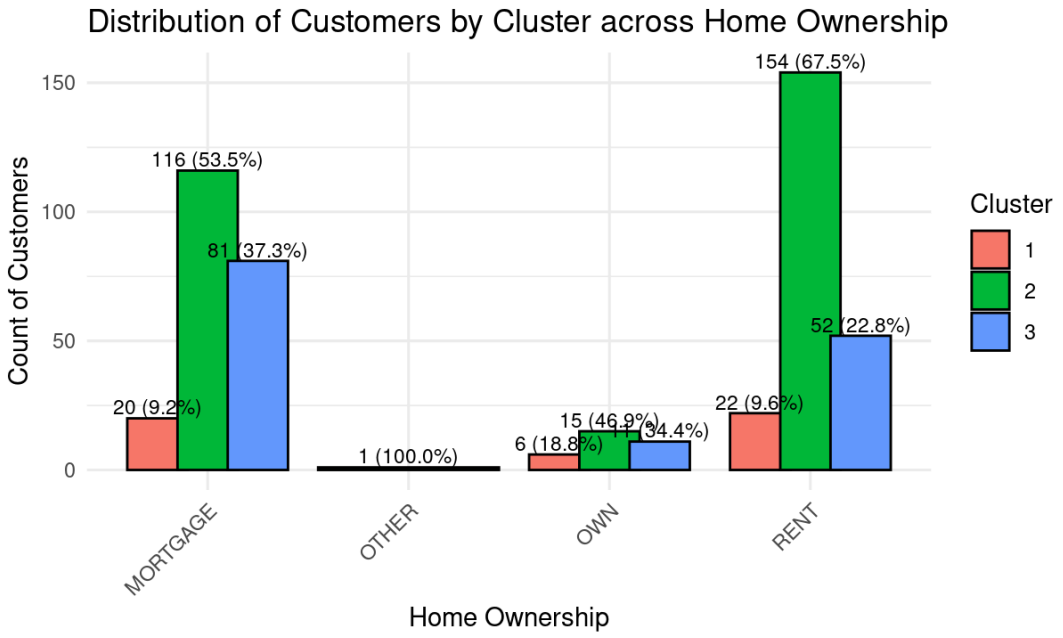
Appendix 8.12.5: The Loan Purpose of three clusters



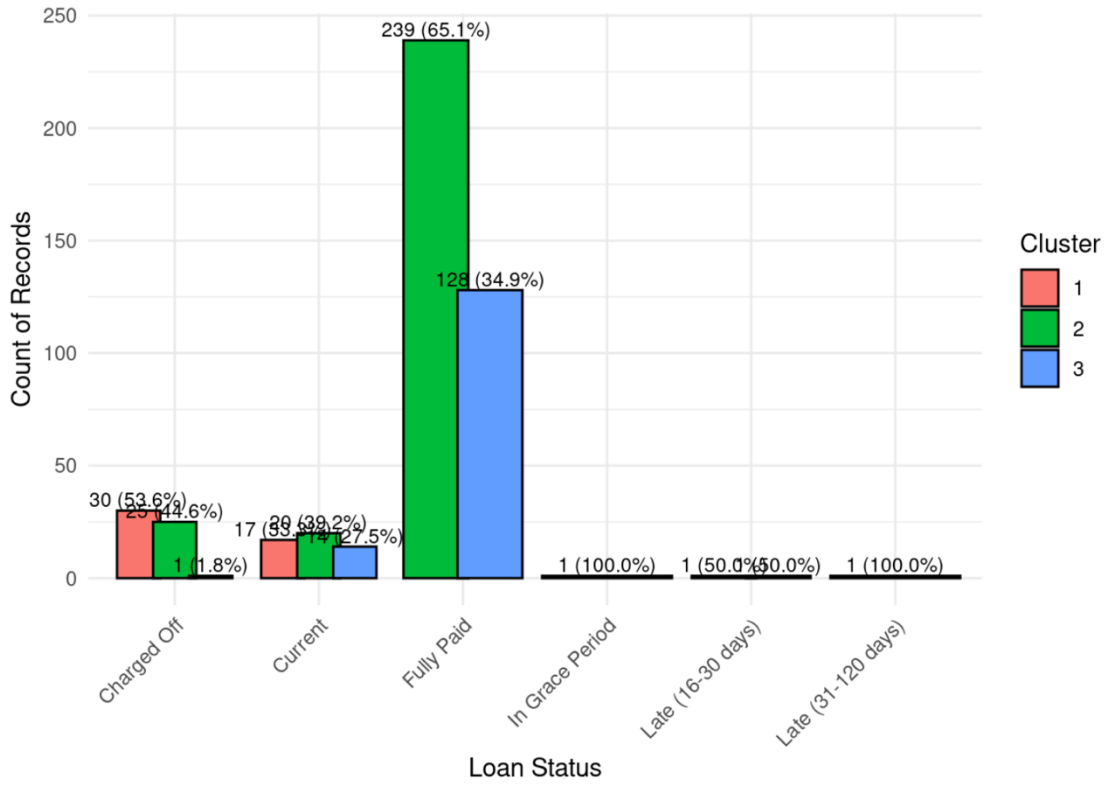
Appendix 8.12.6: Graph of Recoveries and Total Payment of three clusters



Appendix 8.12.7: Graph of Home Ownership of three clusters



Appendix 8.12.8: Graph of Loan Status of three clusters



Appendix 8.13: Characteristics of each cluster for all 10 variables

Variable	**Cluster 1**	**Cluster 2**	**Cluster 3**
Loan_amnt	£10 000 – £27 000	Below £15 000	Above £15 000
Interest_rate	Above 13%	Below 25%	Below 25%
Grade	Between C and F	Between A and D	Between A and D
Installments	£250 – £750	Below £500	Above £500
Annual Income	£50 000 – £100 000	Below £100 000	Between £50 000 and £100 000
Purpose	Debt: 12.2% Credit card: 5.3%	Debt 54.1% Credit card: 62.1% Home Improvement: 76%	Debt 54.1% Credit card: 32.6% Home Improvement: 24%
Recoveries	Below £2500	Few Below £800	Zero
Total Payment	Below £20 000	Below £20 000	£15 000 - £50 000
Home Ownership	Rent: 9.6% Mortgage: 9.2%	Rent: 67.5% Mortgage: 53.5%	Rent: 22.8% Mortgage: 37.3%
Loan Status	Charged off: 54% Current: 33%	Fully paid: 65%	Fully paid: 35%

	Cluster 1	**Cluster 2**	**Cluster 3**
Number of Good Loans	17	259	142
Number of Bad Loans	31	27	2
Percentage of Bad loans	**64.6%**	**9.4%**	**1.4%**

Appendix 8.14: Group Meeting Minutes

- **Meeting Date:** February 22nd , 2024

Participants: 2028065, 5518354, 5532431, 5531616, 5523853, 5521398, 5583269
Meeting Goal: Discussion and allocation of the tasks
Meeting Agenda/Topic: - Write the meeting agenda (or discussion points) here -

1. Meeting team members
2. Understanding Objective and Discussion on how to select variables
3. Task division
4. Selecting deadlines

Action Points:
All group members – Work on Data Understanding and Revision of Cluster Analysis.

- **Meeting Date:** February 27th, 2024

Participants: 2028065, 5518354, 5532431, 5531616, 5523853, 5521398, 5583269
Meeting Goal: Discussion and allocation of the tasks
Meeting Agenda/Topic:

1. Data Preparation and Selection of Variables
2. Define the process, and assign to do PCA, FA and cluster analysis

Action Points:

5532431 – Clustering using PCA
5531616, 5523853 – Clustering using FA with PCA
5521398, 5583269 – Clustering using FA with ML
2028065, 5518354 - – Prepare the Dataset for PCA and Factor Analysis and Data Preparation and Selection of Variables

- **Meeting Date:** March 5th, 2024

Participants: 2028065, 5518354, 5532431, 5531616, 5523853, 5521398, 5583269
Meeting Goal: Evaluation of Different Methods and Discussion of results
Meeting Agenda/Topic:

1. Discussion of the results
2. Review the Variables selection and Data preparation again

Action Points:

All group members – Discussion of the results and review the Variables selection and Data preparation again

- **Meeting Date:** March 12th, 2024

Participants: 2028065, 5518354, 5532431, 5531616, 5523853, 5521398, 5583269
Meeting Goal: Evaluation of Cluster Analysis and Interpretation
Meeting Agenda/Topic:

1. Evaluation of Cluster Analysis
2. Interpretation of Clusters

Action Points:

All group members – Work on suggestions and recommendations

- **Meeting Date:** March 14th, 2024

Participants: 2028065, 5518354, 5532431, 5531616, 5523853, 5521398, 5583269
Meeting Goal: Final code review, excel file and write the report
Meeting Agenda/Topic:

1. Final code review
2. Run the codes again and verify that everything works correctly
3. Plotting the clusters and finding insights
4. Draft the report

Action Points:

Work on the following parts in the report
Executive Summary - 2028065
Introduction - 5583269
Data Preparation - 5518354
Doing PCA and FA - 5532431
Doing Cluster Analysis – 2028065, 5523853
Recommendations - 5521398
Conclusion - 5531616

Group_9_IB98D0

2024-03-11

```
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(summarytools)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.4
## v ggplot2   3.4.4     v stringr   1.5.0
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tibble::view()  masks summarytools::view()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(psychTools)
```

```
##
## Attaching package: 'psychTools'
##
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(GPArotation)
```

```
## Warning: package 'GPArotation' was built under R version 4.3.2
```

```
##
## Attaching package: 'GPArotation'
##
## The following objects are masked from 'package:psych':
##
##     equamax, varimin
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following objects are masked from 'package:psych':
##
##     alpha, rescale
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

Data Integrity, and Data Cleaning

```
Original_data <- read_excel("loan_data_ADA_assignment.xlsx")
#Original_data <- read.csv("loan_data_ADA_assignment.csv")
```

Checking the summary to analyze any potential data entry errors

```
# Check summary of the data
summary(Original_data)
```

```
##        id            member_id         loan_amnt      funded_amnt
##  Min.   :  58524   Min.   : 149512   Min.   : 1000   Min.   : 1000
##  1st Qu.:1443048   1st Qu.:1695278   1st Qu.: 8000   1st Qu.: 8000
##  Median :1587758   Median :1857296   Median :12000   Median :12000
##  Mean   :1918444   Mean   :2283786   Mean   :13901   Mean   :13896
##  3rd Qu.:2311939   3rd Qu.:2744578   3rd Qu.:19200   3rd Qu.:19200
##  Max.   :3304574   Max.   :4076727   Max.   :35000   Max.   :35000
##
##  funded_amnt_inv      term          int_rate      installment
##  Min.   :  950   Min.   :36.00   Min.   : 6.00   Min.   :  25.81
##  1st Qu.: 7950   1st Qu.:36.00   1st Qu.:11.14   1st Qu.: 255.66
##  Median :12000   Median :36.00   Median :14.09   Median : 399.26
##  Mean   :13878   Mean   :40.49   Mean   :14.00   Mean   : 436.95
##  3rd Qu.:19175   3rd Qu.:36.00   3rd Qu.:17.27   3rd Qu.: 567.04
##  Max.   :35000   Max.   :60.00   Max.   :24.89   Max.   :1388.45
##
##     grade            sub_grade          emp_title           emp_length
##  Length:50000       Length:50000       Length:50000       Min.   : 1.000
##  Class :character   Class :character   Class :character   1st Qu.: 3.000
##  Mode  :character   Mode  :character   Mode  :character   Median : 6.000
##                                                           Mean   : 5.993
##                                                           3rd Qu.:10.000
##                                                           Max.   :10.000
##                                                           NA's   :1802
##  home_ownership       annual_inc      verification_status
##  Length:50000       Min.   :   5000   Length:50000
##  Class :character   1st Qu.:  45000   Class :character
##  Mode  :character   Median :  60000   Mode  :character
##                     Mean   :  71317
##                     3rd Qu.:  85000
##                     Max.   :7141778
##
##     issue_d                      loan_status         pymnt_plan
```

```
##  Min.   :2012-05-01 00:00:00.00   Length:50000       Length:50000
##  1st Qu.:2012-08-01 00:00:00.00   Class :character   Class :character
##  Median :2012-10-01 00:00:00.00   Mode  :character   Mode  :character
##  Mean   :2012-09-29 03:53:13.33
##  3rd Qu.:2012-12-01 00:00:00.00
##  Max.   :2013-02-01 00:00:00.00
##
##      desc             purpose             title             zip_code
##  Length:50000       Length:50000       Length:50000       Length:50000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   addr_state             dti         delinq_2yrs
##  Length:50000       Min.   : 0.00   Min.   : 0.0000
##  Class :character   1st Qu.:11.51   1st Qu.: 0.0000
##  Mode  :character   Median :17.16   Median : 0.0000
##                     Mean   :17.37   Mean   : 0.2244
##                     3rd Qu.:23.05   3rd Qu.: 0.0000
##                     Max.   :34.99   Max.   :18.0000
##
##  earliest_cr_line                 inq_last_6mths   mths_since_last_delinq
##  Min.   :1951-12-01 00:00:00.000   Min.   :0.0000   Min.   :  0.00
##  1st Qu.:1994-05-01 00:00:00.000   1st Qu.:0.0000   1st Qu.: 18.00
##  Median :1999-01-01 00:00:00.000   Median :1.0000   Median : 33.00
##  Mean   :1997-09-29 09:34:28.416   Mean   :0.8389   Mean   : 36.08
##  3rd Qu.:2002-05-01 00:00:00.000   3rd Qu.:1.0000   3rd Qu.: 52.00
##  Max.   :2009-12-01 00:00:00.000   Max.   :8.0000   Max.   :152.00
##                                                     NA's   :28126
##  mths_since_last_record    open_acc        pub_rec          revol_bal
##  Min.   :  2.0          Min.   : 0.00   Min.   :0.00000   Min.   :      0
##  1st Qu.: 76.0          1st Qu.: 8.00   1st Qu.:0.00000   1st Qu.:   7102
##  Median : 93.0          Median :10.00   Median :0.00000   Median :  12368
##  Mean   : 87.7          Mean   :11.01   Mean   :0.05648   Mean   :  16011
##  3rd Qu.:106.0          3rd Qu.:14.00   3rd Qu.:0.00000   3rd Qu.:  20515
##  Max.   :119.0          Max.   :53.00   Max.   :8.00000   Max.   :1743266
##  NA's   :47468
##    revol_util       total_acc      total_pymnt    total_pymnt_inv
##  Min.   :0.0000   Min.   : 2.00   Min.   :    0   Min.   :    0
##  1st Qu.:0.4310   1st Qu.:16.00   1st Qu.: 7614   1st Qu.: 7601
##  Median :0.6150   Median :23.00   Median :12858   Median :12842
##  Mean   :0.5885   Mean   :24.31   Mean   :14828   Mean   :14808
##  3rd Qu.:0.7750   3rd Qu.:31.00   3rd Qu.:20051   3rd Qu.:20024
##  Max.   :1.1390   Max.   :99.00   Max.   :57778   Max.   :57778
##  NA's   :31
##  total_rec_prncp total_rec_int   total_rec_late_fee   recoveries
##  Min.   :    0   Min.   :    0   Min.   :  0.0000   Min.   :    0.0
##  1st Qu.: 6000   1st Qu.: 1058   1st Qu.:  0.0000   1st Qu.:    0.0
##  Median :10000   Median : 2047   Median :  0.0000   Median :    0.0
##  Mean   :11611   Mean   : 3071   Mean   :  0.8419   Mean   :  144.2
##  3rd Qu.:15479   3rd Qu.: 3737   3rd Qu.:  0.0000   3rd Qu.:    0.0
##  Max.   :35000   Max.   :22778   Max.   :286.7476   Max.   :33520.3
```

```
##
##  collection_recovery_fee  last_pymnt_d                  last_pymnt_amnt
##  Min.   :   0.00         Min.   :2012-06-01 00:00:00.00   Min.   :    0.0
##  1st Qu.:   0.00         1st Qu.:2014-03-01 00:00:00.00   1st Qu.:  353.1
##  Median :   0.00         Median :2015-03-01 00:00:00.00   Median :  723.6
##  Mean   :  10.66         Mean   :2014-11-26 07:40:19.91   Mean   : 3569.0
##  3rd Qu.:   0.00         3rd Qu.:2015-10-01 00:00:00.00   3rd Qu.: 4675.9
##  Max.   :3896.24         Max.   :2015-12-01 00:00:00.00   Max.   :35683.2
##                          NA's   :43
##   next_pymnt_d                    last_credit_pull_d
##  Min.   :2016-01-01 00:00:00.00   Min.   :2012-05-01 00:00:00.00
##  1st Qu.:2016-01-01 00:00:00.00   1st Qu.:2015-03-01 00:00:00.00
##  Median :2016-01-01 00:00:00.00   Median :2015-11-01 00:00:00.00
##  Mean   :2016-01-06 08:08:08.33   Mean   :2015-06-01 13:41:50.21
##  3rd Qu.:2016-01-01 00:00:00.00   3rd Qu.:2015-12-01 00:00:00.00
##  Max.   :2016-02-01 00:00:00.00   Max.   :2015-12-01 00:00:00.00
##  NA's   :42864
##  collections_12_mths_ex_med mths_since_last_major_derog  policy_code
##  Min.   :0.00000            Min.   :  0.00              Min.   :1
##  1st Qu.:0.00000            1st Qu.: 25.00              1st Qu.:1
##  Median :0.00000            Median : 40.00              Median :1
##  Mean   :0.00114            Mean   : 42.31              Mean   :1
##  3rd Qu.:0.00000            3rd Qu.: 59.00              3rd Qu.:1
##  Max.   :2.00000            Max.   :152.00              Max.   :1
##                             NA's   :42880
##  acc_now_delinq     tot_coll_amt    tot_cur_bal      total_credit_rv
##  Min.   :0.00000   Min.   :    0   Min.   :      0   Min.   :      0
##  1st Qu.:0.00000   1st Qu.:    0   1st Qu.:  26298   1st Qu.:  14000
##  Median :0.00000   Median :    0   Median :  72117   Median :  22800
##  Mean   :0.00082   Mean   :   52   Mean   : 133594   Mean   :  29300
##  3rd Qu.:0.00000   3rd Qu.:    0   3rd Qu.: 202362   3rd Qu.:  36600
##  Max.   :4.00000   Max.   :55009   Max.   :8000078   Max.   :2013133
##                    NA's   :14618   NA's   :14618     NA's   :14618
##  loan_is_bad
##  Mode :logical
##  FALSE:42186
##  TRUE :7814
##
##
##
##
```

```
# Check structure of the data
str(Original_data)
```

```
## tibble [50,000 x 53] (S3: tbl_df/tbl/data.frame)
##  $ id                          : num [1:50000] 3296446 3286412 3286406 3296434 3286395 ...
##  $ member_id                   : num [1:50000] 4068857 4058853 4058848 4068843 4058836 ...
##  $ loan_amnt                   : num [1:50000] 11200 10000 8000 16000 4000 15000 8000 19800 4000 1440
##  $ funded_amnt                 : num [1:50000] 11200 10000 8000 16000 4000 15000 8000 19800 4000 1440
##  $ funded_amnt_inv             : num [1:50000] 11200 10000 8000 15950 4000 ...
##  $ term                        : num [1:50000] 36 36 36 36 36 36 36 60 36 36 ...
##  $ int_rate                    : num [1:50000] 6.62 11.14 16.29 7.9 7.9 ...
##  $ installment                 : num [1:50000] 344 328 282 501 125 ...
```

```
##  $ grade                       : chr [1:50000] "A" "B" "C" "A" ...
##  $ sub_grade                   : chr [1:50000] "A2" "B2" "C4" "A4" ...
##  $ emp_title                   : chr [1:50000] "Nokia Siemens Network" "creative financial group" "Te
##  $ emp_length                  : num [1:50000] 10 2 7 10 10 10 10 10 NA 3 ...
##  $ home_ownership              : chr [1:50000] "OWN" "MORTGAGE" "RENT" "MORTGAGE" ...
##  $ annual_inc                  : num [1:50000] 108000 65000 35000 110000 155000 ...
##  $ verification_status         : chr [1:50000] "Not Verified" "Not Verified" "Not Verified" "Verified
##  $ issue_d                     : POSIXct[1:50000], format: "2013-02-01" "2013-02-01" ...
##  $ loan_status                 : chr [1:50000] "Current" "Charged Off" "Current" "Fully Paid" ...
##  $ pymnt_plan                  : chr [1:50000] "n" "n" "n" "n" ...
##  $ desc                        : chr [1:50000] "Borrower added on 01/27/13 > Credit Card Refinancing<
##  $ purpose                     : chr [1:50000] "credit_card" "credit_card" "debt_consolidation" "debt_
##  $ title                       : chr [1:50000] "Credit Card" "my lending club Loan" "All in One" "Deb
##  $ zip_code                    : chr [1:50000] "750xx" "085xx" "440xx" "060xx" ...
##  $ addr_state                  : chr [1:50000] "TX" "NJ" "OH" "CT" ...
##  $ dti                         : num [1:50000] 12.52 9.58 27.84 28.87 17.87 ...
##  $ delinq_2yrs                 : num [1:50000] 0 0 0 0 0 0 1 0 0 0 ...
##  $ earliest_cr_line            : POSIXct[1:50000], format: "2002-10-01" "2000-03-01" ...
##  $ inq_last_6mths              : num [1:50000] 0 0 2 0 0 2 0 1 0 1 ...
##  $ mths_since_last_delinq      : num [1:50000] NA NA NA NA NA 67 19 NA NA NA ...
##  $ mths_since_last_record      : num [1:50000] NA NA NA NA NA NA NA NA NA NA ...
##  $ open_acc                    : num [1:50000] 9 9 12 21 7 9 7 18 9 10 ...
##  $ pub_rec                     : num [1:50000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ revol_bal                   : num [1:50000] 37822 16623 17938 23691 43945 ...
##  $ revol_util                  : num [1:50000] 0.662 0.742 0.72 0.752 0.955 0.681 0.476 0.767 0.873 0
##  $ total_acc                   : num [1:50000] 21 11 17 56 21 19 30 26 14 29 ...
##  $ total_pymnt                 : num [1:50000] 11676 4620 9602 16768 4252 ...
##  $ total_pymnt_inv             : num [1:50000] 11676 4620 9602 16716 4252 ...
##  $ total_rec_prncp             : num [1:50000] 10505 2711 7447 16000 3749 ...
##  $ total_rec_int               : num [1:50000] 1172 898 2155 768 503 ...
##  $ total_rec_late_fee          : num [1:50000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ recoveries                  : num [1:50000] 0 1012 0 0 0 ...
##  $ collection_recovery_fee     : num [1:50000] 0 10.1 0 0 0 ...
##  $ last_pymnt_d                : POSIXct[1:50000], format: "2015-12-01" "2014-01-01" ...
##  $ last_pymnt_amnt             : num [1:50000] 344 328 282 13269 125 ...
##  $ next_pymnt_d                : POSIXct[1:50000], format: "2016-01-01" NA ...
##  $ last_credit_pull_d          : POSIXct[1:50000], format: "2015-12-01" "2014-01-01" ...
##  $ collections_12_mths_ex_med  : num [1:50000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ mths_since_last_major_derog: num [1:50000] NA NA NA NA NA 67 19 NA NA NA ...
##  $ policy_code                 : num [1:50000] 1 1 1 1 1 1 1 1 1 1 ...
##  $ acc_now_delinq              : num [1:50000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ tot_coll_amt                : num [1:50000] 0 0 0 0 0 52 0 0 90 0 ...
##  $ tot_cur_bal                 : num [1:50000] 187717 16623 17938 372771 331205 ...
##  $ total_credit_rv             : num [1:50000] 66400 22400 24900 31500 46000 27100 31000 20800 13800 4
##  $ loan_is_bad                 : logi [1:50000] FALSE TRUE FALSE FALSE FALSE FALSE ...
```

```
# Check NA
(summarise_all(Original_data, ~ sum(is.na(.x))))
```

```
## # A tibble: 1 x 53
##      id member_id loan_amnt funded_amnt funded_amnt_inv  term int_rate
##   <int>     <int>     <int>       <int>           <int> <int>    <int>
## 1     0         0         0           0               0     0        0
## # i 46 more variables: installment <int>, grade <int>, sub_grade <int>,
```

```
## #   emp_title <int>, emp_length <int>, home_ownership <int>, annual_inc <int>,
## #   verification_status <int>, issue_d <int>, loan_status <int>,
## #   pymnt_plan <int>, desc <int>, purpose <int>, title <int>, zip_code <int>,
## #   addr_state <int>, dti <int>, delinq_2yrs <int>, earliest_cr_line <int>,
## #   inq_last_6mths <int>, mths_since_last_delinq <int>,
## #   mths_since_last_record <int>, open_acc <int>, pub_rec <int>, ...
```

Variables selection

```
# We selected the initials variables for analyse by removing string, categorical variable and remove va

df = subset(Original_data, select = -c(id, member_id, term, grade, sub_grade, emp_title, emp_length, hor
```

```
# Check overall statistics of remaining data
print(dfSummary(df), file = 'Summary.html')
```

```
## Output file written: /Users/savvinanicolaou/Downloads/Summary.html
```

```
# After checking overall statistics, we found 7 variables that have majority of observations near Zero,
summary(df)
```

```
##    loan_amnt      funded_amnt    funded_amnt_inv    int_rate
##  Min.   : 1000   Min.   : 1000   Min.   :  950   Min.   : 6.00
##  1st Qu.: 8000   1st Qu.: 8000   1st Qu.: 7950   1st Qu.:11.14
##  Median :12000   Median :12000   Median :12000   Median :14.09
##  Mean   :13901   Mean   :13896   Mean   :13878   Mean   :14.00
##  3rd Qu.:19200   3rd Qu.:19200   3rd Qu.:19175   3rd Qu.:17.27
##  Max.   :35000   Max.   :35000   Max.   :35000   Max.   :24.89
##
##   installment        annual_inc           dti         delinq_2yrs
##  Min.   :  25.81   Min.   :   5000   Min.   : 0.00   Min.   : 0.0000
##  1st Qu.: 255.66   1st Qu.:  45000   1st Qu.:11.51   1st Qu.: 0.0000
##  Median : 399.26   Median :  60000   Median :17.16   Median : 0.0000
##  Mean   : 436.95   Mean   :  71317   Mean   :17.37   Mean   : 0.2244
##  3rd Qu.: 567.04   3rd Qu.:  85000   3rd Qu.:23.05   3rd Qu.: 0.0000
##  Max.   :1388.45   Max.   :7141778   Max.   :34.99   Max.   :18.0000
##
##  inq_last_6mths      open_acc        pub_rec          revol_bal
##  Min.   :0.0000   Min.   : 0.00   Min.   :0.00000   Min.   :      0
##  1st Qu.:0.0000   1st Qu.: 8.00   1st Qu.:0.00000   1st Qu.:   7102
##  Median :1.0000   Median :10.00   Median :0.00000   Median :  12368
##  Mean   :0.8389   Mean   :11.01   Mean   :0.05648   Mean   :  16011
##  3rd Qu.:1.0000   3rd Qu.:14.00   3rd Qu.:0.00000   3rd Qu.:  20515
##  Max.   :8.0000   Max.   :53.00   Max.   :8.00000   Max.   :1743266
##
##    revol_util       total_acc      total_pymnt    total_pymnt_inv
##  Min.   :0.0000   Min.   : 2.00   Min.   :    0   Min.   :    0
##  1st Qu.:0.4310   1st Qu.:16.00   1st Qu.: 7614   1st Qu.: 7601
##  Median :0.6150   Median :23.00   Median :12858   Median :12842
##  Mean   :0.5885   Mean   :24.31   Mean   :14828   Mean   :14808
##  3rd Qu.:0.7750   3rd Qu.:31.00   3rd Qu.:20051   3rd Qu.:20024
```

```
##  Max.   :1.1390   Max.   :99.00   Max.   :57778   Max.   :57778
##  NA's   :31
##  total_rec_prncp total_rec_int   total_rec_late_fee   recoveries
##  Min.   :    0   Min.   :    0   Min.   :  0.0000   Min.   :    0.0
##  1st Qu.: 6000   1st Qu.: 1058   1st Qu.:  0.0000   1st Qu.:    0.0
##  Median :10000   Median : 2047   Median :  0.0000   Median :    0.0
##  Mean   :11611   Mean   : 3071   Mean   :  0.8419   Mean   :  144.2
##  3rd Qu.:15479   3rd Qu.: 3737   3rd Qu.:  0.0000   3rd Qu.:    0.0
##  Max.   :35000   Max.   :22778   Max.   :286.7476   Max.   :33520.3
##
##  collection_recovery_fee last_pymnt_amnt   collections_12_mths_ex_med
##  Min.   :   0.00         Min.   :    0.0   Min.   :0.00000
##  1st Qu.:   0.00         1st Qu.:  353.1   1st Qu.:0.00000
##  Median :   0.00         Median :  723.6   Median :0.00000
##  Mean   :  10.66         Mean   : 3569.0   Mean   :0.00114
##  3rd Qu.:   0.00         3rd Qu.: 4675.9   3rd Qu.:0.00000
##  Max.   :3896.24         Max.   :35683.2   Max.   :2.00000
##
##  acc_now_delinq
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.00082
##  3rd Qu.:0.00000
##  Max.   :4.00000
##
```

```
df = subset(df, select = -c(delinq_2yrs, pub_rec, total_rec_late_fee, recoveries, collection_recovery_f
```

```
# We observe correlation matrix of the remaining variables to see the variables that are represent the
lowerCor(df)
```

```
##                ln_mn fndd_ fnd__ int_r instl annl_ dti   in__6 opn_c rvl_b
## loan_amnt       1.00
## funded_amnt     1.00  1.00
## funded_amnt_inv 1.00  1.00  1.00
## int_rate        0.30  0.30  0.30  1.00
## installment     0.96  0.96  0.96  0.29  1.00
## annual_inc      0.29  0.29  0.29  0.00  0.28  1.00
## dti             0.04  0.04  0.04  0.15  0.04 -0.17  1.00
## inq_last_6mths  0.03  0.03  0.03  0.19  0.03  0.06  0.01  1.00
## open_acc        0.19  0.19  0.19  0.07  0.19  0.12  0.31  0.12  1.00
## revol_bal       0.32  0.32  0.32  0.05  0.30  0.32  0.15  0.01  0.23  1.00
## revol_util      0.09  0.09  0.09  0.44  0.13  0.02  0.25 -0.10 -0.08  0.20
## total_acc       0.25  0.25  0.25  0.04  0.24  0.19  0.24  0.14  0.66  0.23
## total_pymnt     0.89  0.89  0.89  0.26  0.89  0.27  0.02  0.01  0.17  0.29
## total_pymnt_inv 0.89  0.89  0.89  0.26  0.89  0.27  0.02  0.01  0.17  0.29
## total_rec_prncp 0.79  0.79  0.79  0.08  0.82  0.27 -0.02 -0.01  0.15  0.27
## total_rec_int   0.74  0.74  0.74  0.56  0.66  0.17  0.11  0.05  0.14  0.21
## last_pymnt_amnt 0.40  0.40  0.40  0.13  0.37  0.15 -0.04  0.05  0.07  0.13
##                rvl_t
## loan_amnt
## funded_amnt
```

```
## funded_amnt_inv
## int_rate
## installment
## annual_inc
## dti
## inq_last_6mths
## open_acc
## revol_bal
## revol_util       1.00
## total_acc       -0.05
## total_pymnt      0.10
## total_pymnt_inv  0.10
## total_rec_prncp  0.03
## total_rec_int    0.21
## last_pymnt_amnt -0.01
##                 ttl_c ttl_p ttl_p_ ttl_rc_p ttl_rc_n lst__
## total_acc        1.00
## total_pymnt      0.23  1.00
## total_pymnt_inv  0.23  1.00  1.00
## total_rec_prncp  0.21  0.95  0.95   1.00
## total_rec_int    0.17  0.73  0.73   0.49     1.00
## last_pymnt_amnt  0.15  0.49  0.49   0.58     0.10     1.00
```

```
# We found that, loan_amnt, funded_amnt and funded_amnt_inv are highly correlate with each other (Corre
df = subset(df, select = -c(funded_amnt, funded_amnt_inv, total_pymnt_inv))
```

```
summary(df)
```

```
##    loan_amnt        int_rate      installment        annual_inc
##  Min.   : 1000   Min.   : 6.00   Min.   :  25.81   Min.   :   5000
##  1st Qu.: 8000   1st Qu.:11.14   1st Qu.: 255.66   1st Qu.:  45000
##  Median :12000   Median :14.09   Median : 399.26   Median :  60000
##  Mean   :13901   Mean   :14.00   Mean   : 436.95   Mean   :  71317
##  3rd Qu.:19200   3rd Qu.:17.27   3rd Qu.: 567.04   3rd Qu.:  85000
##  Max.   :35000   Max.   :24.89   Max.   :1388.45   Max.   :7141778
##
##       dti        inq_last_6mths      open_acc       revol_bal
##  Min.   : 0.00   Min.   :0.0000   Min.   : 0.00   Min.   :      0
##  1st Qu.:11.51   1st Qu.:0.0000   1st Qu.: 8.00   1st Qu.:   7102
##  Median :17.16   Median :1.0000   Median :10.00   Median :  12368
##  Mean   :17.37   Mean   :0.8389   Mean   :11.01   Mean   :  16011
##  3rd Qu.:23.05   3rd Qu.:1.0000   3rd Qu.:14.00   3rd Qu.:  20515
##  Max.   :34.99   Max.   :8.0000   Max.   :53.00   Max.   :1743266
##
##    revol_util       total_acc      total_pymnt    total_rec_prncp
##  Min.   :0.0000   Min.   : 2.00   Min.   :    0   Min.   :    0
##  1st Qu.:0.4310   1st Qu.:16.00   1st Qu.: 7614   1st Qu.: 6000
##  Median :0.6150   Median :23.00   Median :12858   Median :10000
##  Mean   :0.5885   Mean   :24.31   Mean   :14828   Mean   :11611
##  3rd Qu.:0.7750   3rd Qu.:31.00   3rd Qu.:20051   3rd Qu.:15479
##  Max.   :1.1390   Max.   :99.00   Max.   :57778   Max.   :35000
##  NA's   :31
##  total_rec_int   last_pymnt_amnt
```

```
##  Min.   :    0   Min.   :    0.0
##  1st Qu.: 1058   1st Qu.:  353.1
##  Median : 2047   Median :  723.6
##  Mean   : 3071   Mean   : 3569.0
##  3rd Qu.: 3737   3rd Qu.: 4675.9
##  Max.   :22778   Max.   :35683.2
##
```

```
lowerCor(df)
```

```
##                ln_mn int_r instl annl_ dti   in__6 opn_c rvl_b rvl_t ttl_c
## loan_amnt       1.00
## int_rate        0.30  1.00
## installment     0.96  0.29  1.00
## annual_inc      0.29  0.00  0.28  1.00
## dti             0.04  0.15  0.04 -0.17  1.00
## inq_last_6mths  0.03  0.19  0.03  0.06  0.01  1.00
## open_acc        0.19  0.07  0.19  0.12  0.31  0.12  1.00
## revol_bal       0.32  0.05  0.30  0.32  0.15  0.01  0.23  1.00
## revol_util      0.09  0.44  0.13  0.02  0.25 -0.10 -0.08  0.20  1.00
## total_acc       0.25  0.04  0.24  0.19  0.24  0.14  0.66  0.23 -0.05  1.00
## total_pymnt     0.89  0.26  0.89  0.27  0.02  0.01  0.17  0.29  0.10  0.23
## total_rec_prncp 0.79  0.08  0.82  0.27 -0.02 -0.01  0.15  0.27  0.03  0.21
## total_rec_int   0.74  0.56  0.66  0.17  0.11  0.05  0.14  0.21  0.21  0.17
## last_pymnt_amnt 0.40  0.13  0.37  0.15 -0.04  0.05  0.07  0.13 -0.01  0.15
##                ttl_p
## loan_amnt
## int_rate
## installment
## annual_inc
## dti
## inq_last_6mths
## open_acc
## revol_bal
## revol_util
## total_acc
## total_pymnt     1.00
## total_rec_prncp 0.95
## total_rec_int   0.73
## last_pymnt_amnt 0.49
##                ttl_rc_p ttl_rc_n lst__
## total_rec_prncp  1.00
## total_rec_int    0.49     1.00
## last_pymnt_amnt  0.58     0.10     1.00
```

Data Cleaning

```
# NA removal
(summarise_all(df, ~ sum(is.na(.x))))
```

```
## # A tibble: 1 x 14
```

```
##   loan_amnt int_rate installment annual_inc   dti inq_last_6mths open_acc
##       <int>    <int>       <int>      <int> <int>          <int>    <int>
## 1         0        0           0          0     0              0        0
## # i 7 more variables: revol_bal <int>, revol_util <int>, total_acc <int>,
## #   total_pymnt <int>, total_rec_prncp <int>, total_rec_int <int>,
## #   last_pymnt_amnt <int>
```

```
# remove observations that have NA value, 31 observations were removed.
df <- df[!is.na(df$revol_util), ]
df$index <- 1:nrow(df)

Original_data_wo_outliers <- Original_data[!is.na(Original_data$revol_util), ]
Original_data_wo_outliers$index <- 1:nrow(Original_data_wo_outliers)
```

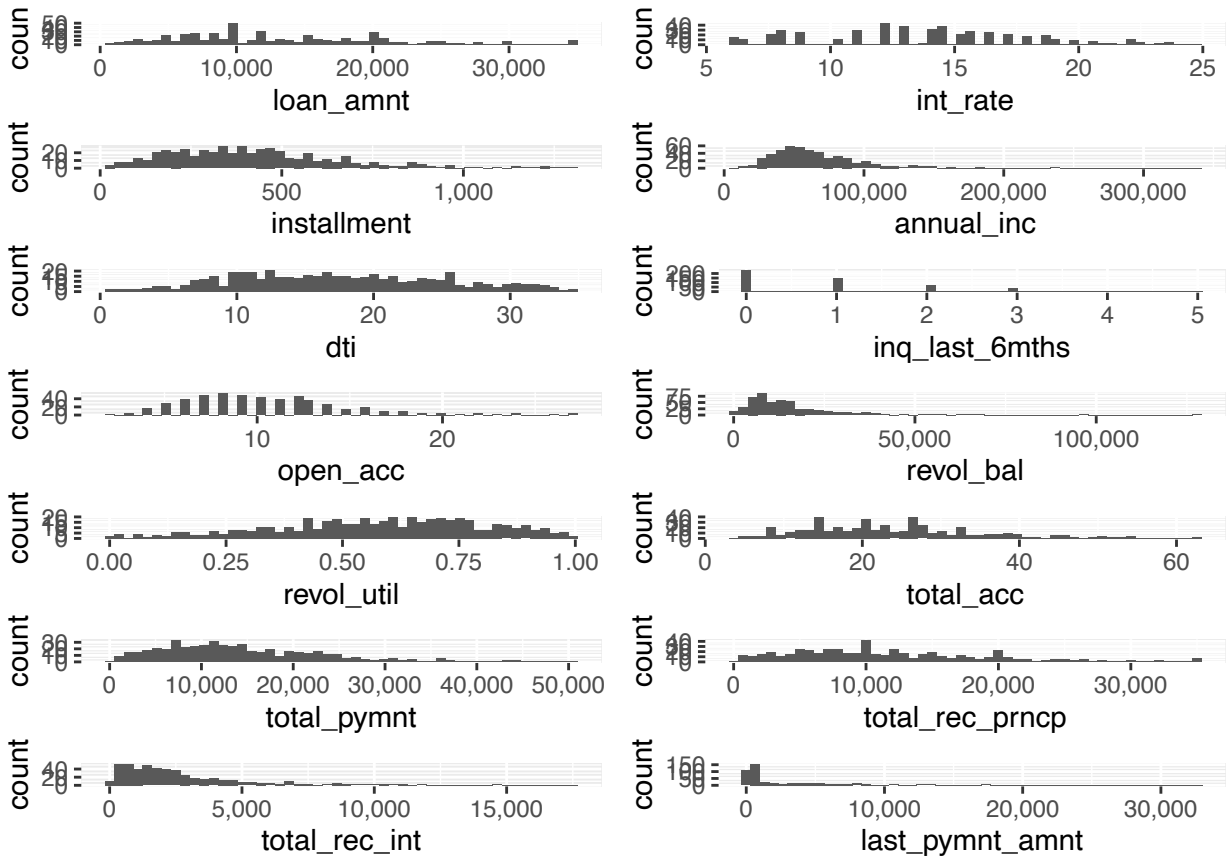
Random Sampling

```
# before doing any analysis, we make a random sampling from our data set to be representative of the wh
set.seed(123)
sp.df <- df[sample(nrow(df), 500), ] # Sampling Data
summary(sp.df)
```

```
##    loan_amnt        int_rate      installment        annual_inc
##  Min.   : 1000   Min.   : 6.03   Min.   :  33.75   Min.   :  8520
##  1st Qu.: 7369   1st Qu.:11.14   1st Qu.: 245.63   1st Qu.: 44420
##  Median :12000   Median :14.09   Median : 393.61   Median : 60000
##  Mean   :13307   Mean   :13.87   Mean   : 423.81   Mean   : 67931
##  3rd Qu.:18338   3rd Qu.:16.29   3rd Qu.: 548.21   3rd Qu.: 80000
##  Max.   :35000   Max.   :24.70   Max.   :1309.49   Max.   :340000
##       dti        inq_last_6mths     open_acc       revol_bal
##  Min.   : 0.86   Min.   :0.000   Min.   : 2.00   Min.   :     0
##  1st Qu.:11.33   1st Qu.:0.000   1st Qu.: 7.00   1st Qu.:  6664
##  Median :16.84   Median :1.000   Median :10.00   Median : 11664
##  Mean   :17.39   Mean   :0.862   Mean   :10.43   Mean   : 14387
##  3rd Qu.:23.20   3rd Qu.:1.000   3rd Qu.:13.00   3rd Qu.: 17838
##  Max.   :34.76   Max.   :5.000   Max.   :27.00   Max.   :127796
##    revol_util       total_acc      total_pymnt      total_rec_prncp
##  Min.   :0.0000   Min.   : 4.00   Min.   :  367.3   Min.   :    0
##  1st Qu.:0.4278   1st Qu.:15.75   1st Qu.: 7648.1   1st Qu.: 6000
##  Median :0.6005   Median :23.00   Median :12605.5   Median :10000
##  Mean   :0.5781   Mean   :24.11   Mean   :14390.0   Mean   :11413
##  3rd Qu.:0.7470   3rd Qu.:30.00   3rd Qu.:19883.4   3rd Qu.:15184
##  Max.   :0.9950   Max.   :63.00   Max.   :50846.5   Max.   :35000
##  total_rec_int     last_pymnt_amnt       index
##  Min.   :    0.0   Min.   :    0.0   Min.   :   31
##  1st Qu.:  995.2   1st Qu.:  364.8   1st Qu.:12049
##  Median : 2000.8   Median :  827.7   Median :25244
##  Mean   : 2865.2   Mean   : 3646.0   Mean   :24906
##  3rd Qu.: 3679.6   3rd Qu.: 5104.8   3rd Qu.:37276
##  Max.   :17508.2   Max.   :32665.3   Max.   :49929
```

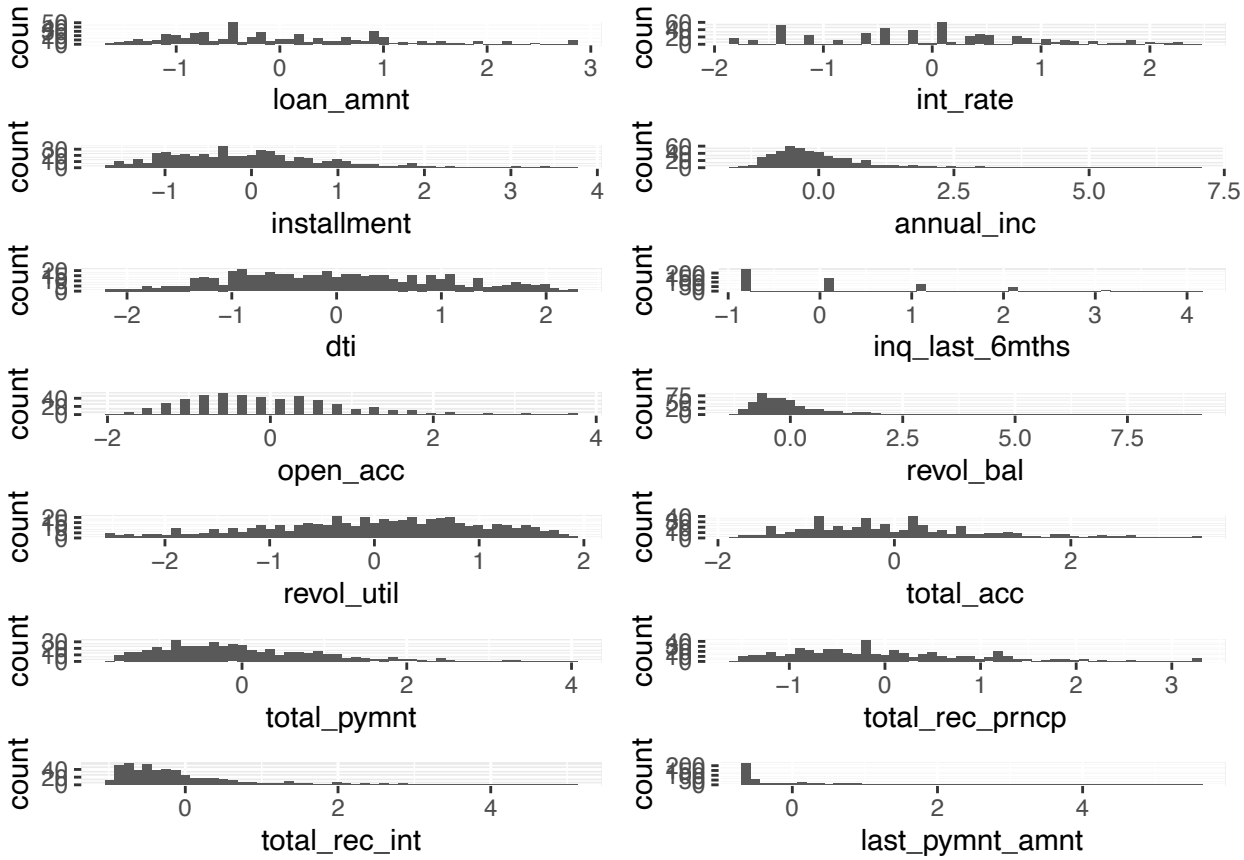
```r
sample_indices <- sp.df$index
Original_data_wo_outliers_sample <- Original_data_wo_outliers[sample_indices,]
sp.df$index <- NULL
Original_data_wo_outliers_sample$index <- NULL
```

```r
# plot the sampling data
grid.arrange(
  ggplot(sp.df, aes(x = loan_amnt))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales::comm
  ggplot(sp.df, aes(x = int_rate))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales::comma
  ggplot(sp.df, aes(x = installment))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales::co
  ggplot(sp.df, aes(x = annual_inc))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales::com
  ggplot(sp.df, aes(x = dti))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales::comma),
  ggplot(sp.df, aes(x = inq_last_6mths))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales
  ggplot(sp.df, aes(x = open_acc))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales::comma
  ggplot(sp.df, aes(x = revol_bal))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales::comm
  ggplot(sp.df, aes(x = revol_util))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales::com
  ggplot(sp.df, aes(x = total_acc))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales::comm
  ggplot(sp.df, aes(x = total_pymnt))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales::co
  ggplot(sp.df, aes(x = total_rec_prncp))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales
  ggplot(sp.df, aes(x = total_rec_int))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales:
  ggplot(sp.df, aes(x = last_pymnt_amnt))+ geom_histogram(bins = 50)+ scale_x_continuous(labels = scales
nrow=7)
```



```r
# standardlize data
preProcValues <- preProcess(sp.df, method = c("center", "scale"))
z.df <- predict(preProcValues, sp.df) # standardlized sampling data
```

```r
# plot the sampling data
grid.arrange(
  ggplot(z.df, aes(x = loan_amnt))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales::comm
  ggplot(z.df, aes(x = int_rate))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales::comma
  ggplot(z.df, aes(x = installment))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales::co
  ggplot(z.df, aes(x = annual_inc))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales::con
  ggplot(z.df, aes(x = dti))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales::comma),
  ggplot(z.df, aes(x = inq_last_6mths))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales
  ggplot(z.df, aes(x = open_acc))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales::comma
  ggplot(z.df, aes(x = revol_bal))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales::comm
  ggplot(z.df, aes(x = revol_util))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales::con
  ggplot(z.df, aes(x = total_acc))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales::comm
  ggplot(z.df, aes(x = total_pymnt))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales::co
  ggplot(z.df, aes(x = total_rec_prncp))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales
  ggplot(z.df, aes(x = total_rec_int))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales:
  ggplot(z.df, aes(x = last_pymnt_amnt))+ geom_histogram(bins = 50) + scale_x_continuous(labels = scales
nrow=7)
```




```r
# remove outlier z-score more than +- 4, for comparison the result (20 outliers)
z.outl.df <- z.df %>% filter(loan_amnt < 4 & loan_amnt > -4 &
                             int_rate < 4 & int_rate > -4 &
```

```
                          installment < 4 & installment > -4 &
                          annual_inc < 4 & annual_inc > -4 &
                          dti < 4 & dti > -4 &
                          inq_last_6mths < 4 & inq_last_6mths > -4 &
                          open_acc < 4 & open_acc > -4 &
                          revol_bal < 4 & revol_bal > -4 &
                          revol_util < 4 & revol_util > -4 &
                          total_acc < 4 & total_acc > -4 &
                          total_pymnt < 4 & total_pymnt > -4 &
                          total_rec_prncp < 4 & total_rec_prncp > -4 &
                          total_rec_int < 4 & total_rec_int > -4 &
                          last_pymnt_amnt < 4 & last_pymnt_amnt > -4
                             )

summary(z.outl.df)
```

```
##    loan_amnt          int_rate         installment         annual_inc
##  Min.   :-1.61832   Min.   :-1.78786   Min.   :-1.63850   Min.   :-1.5359
##  1st Qu.:-0.80304   1st Qu.:-0.62268   1st Qu.:-0.76631   1st Qu.:-0.6445
##  Median :-0.23760   Median : 0.04997   Median :-0.17298   Median :-0.2482
##  Mean   :-0.07122   Mean   :-0.03507   Mean   :-0.06727   Mean   :-0.0859
##  3rd Qu.: 0.61714   3rd Qu.: 0.55161   3rd Qu.: 0.44434   3rd Qu.: 0.3120
##  Max.   : 2.85260   Max.   : 2.46925   Max.   : 3.72043   Max.   : 3.4143
##       dti            inq_last_6mths         open_acc          revol_bal
##  Min.   :-2.156936   Min.   :-0.848111   Min.   :-1.91918   Min.   :-1.16074
##  1st Qu.:-0.788160   1st Qu.:-0.848111   1st Qu.:-0.78142   1st Qu.:-0.62990
##  Median :-0.068386   Median : 0.135776   Median :-0.09876   Median :-0.25207
##  Mean   : 0.000648   Mean   : 0.002542   Mean   :-0.01959   Mean   :-0.08496
##  3rd Qu.: 0.756402   3rd Qu.: 0.135776   3rd Qu.: 0.58390   3rd Qu.: 0.18262
##  Max.   : 2.265416   Max.   : 3.087437   Max.   : 3.76965   Max.   : 3.79265
##    revol_util         total_acc         total_pymnt       total_rec_prncp
##  Min.   :-2.57047   Min.   :-1.79039   Min.   :-1.56269   Min.   :-1.58019
##  1st Qu.:-0.67625   1st Qu.:-0.81106   1st Qu.:-0.77306   1st Qu.:-0.74944
##  Median : 0.06187   Median :-0.09882   Median :-0.23490   Median :-0.19561
##  Mean   :-0.01902   Mean   :-0.02723   Mean   :-0.06188   Mean   :-0.04724
##  3rd Qu.: 0.72885   3rd Qu.: 0.52438   3rd Qu.: 0.52859   3rd Qu.: 0.49668
##  Max.   : 1.85382   Max.   : 3.46237   Max.   : 3.64981   Max.   : 3.26583
##  total_rec_int      last_pymnt_amnt
##  Min.   :-0.99305   Min.   :-0.69433
##  1st Qu.:-0.65216   1st Qu.:-0.62711
##  Median :-0.31858   Median :-0.54409
##  Mean   :-0.07318   Mean   :-0.03837
##  3rd Qu.: 0.20676   3rd Qu.: 0.26749
##  Max.   : 3.88620   Max.   : 3.95006
```

Our objective is to doing clustering analysis for market segmentation, doing so, we want to specify the few variables that contains enough information for doing cluster, specifically is to reduce dimension. In order to reduce dimension, we doing PCA and Factor analysis to find suitable method for handle our loan dataset.

Check criteria for PCA and FA

```
# We check KMO and correlation of our dataset (now we test on 2 datasets, z.df without removing outlier

# Correlation Matrix of dataset without removing outlier
z.df.corr <- cor(z.df)
lowerCor(z.df.corr)
```

```
##                 ln_mn int_r instl annl_ dti   in__6 opn_c rvl_b rvl_t ttl_c
## loan_amnt        1.00
## int_rate         0.16  1.00
## installment      1.00  0.15  1.00
## annual_inc       0.45 -0.51  0.44  1.00
## dti             -0.46  0.16 -0.46 -0.70  1.00
## inq_last_6mths  -0.48  0.00 -0.46 -0.26 -0.04  1.00
## open_acc        -0.42 -0.52 -0.44  0.10  0.20  0.14  1.00
## revol_bal        0.26 -0.35  0.24  0.51 -0.11 -0.52  0.10  1.00
## revol_util       0.11  0.65  0.13 -0.28  0.19 -0.33 -0.64  0.10  1.00
## total_acc       -0.23 -0.67 -0.25  0.36 -0.03  0.04  0.83  0.17 -0.65  1.00
## total_pymnt      0.98  0.13  0.98  0.41 -0.47 -0.48 -0.46  0.21  0.09 -0.26
## total_rec_prncp  0.94 -0.05  0.95  0.47 -0.53 -0.48 -0.42  0.24 -0.01 -0.19
## total_rec_int    0.83  0.57  0.81  0.13 -0.18 -0.35 -0.46  0.07  0.36 -0.39
## last_pymnt_amnt  0.46 -0.13  0.46  0.15 -0.49 -0.33 -0.32  0.00 -0.20 -0.18
##                 ttl_p
## loan_amnt
## int_rate
## installment
## annual_inc
## dti
## inq_last_6mths
## open_acc
## revol_bal
## revol_util
## total_acc
## total_pymnt      1.00
## total_rec_prncp  0.98
## total_rec_int    0.79
## last_pymnt_amnt  0.55
##                 ttl_rc_p ttl_rc_n lst__
## total_rec_prncp  1.00
## total_rec_int    0.63     1.00
## last_pymnt_amnt  0.66     0.08     1.00
```

```
KMO(z.df.corr)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = z.df.corr)
## Overall MSA =  0.67
## MSA for each item =
##       loan_amnt        int_rate     installment      annual_inc             dti
##            0.77            0.59            0.77            0.76            0.52
##  inq_last_6mths        open_acc       revol_bal      revol_util       total_acc
```

```
##            0.56            0.59            0.81            0.63            0.67
##     total_pymnt total_rec_prncp   total_rec_int last_pymnt_amnt
##            0.65            0.61            0.56            0.56
```

```
cortest.bartlett(z.df.corr, n=500)
```

```
## $chisq
## [1] 7304.685
##
## $p.value
## [1] 0
##
## $df
## [1] 91
```

```
# We observe overall KMO at 0.67, and Bartlett test P-value 0 (below 0.05) which is suitable for PCA an
```

```
# Correlation Matrix of dataset that removed outlier
z.df.outl.corr <- cor(z.outl.df)
lowerCor(z.df.outl.corr)
```

```
##                ln_mn int_r instl annl_ dti   in__6 opn_c rvl_b rvl_t ttl_c
## loan_amnt       1.00
## int_rate        0.10  1.00
## installment     1.00  0.11  1.00
## annual_inc      0.43 -0.58  0.41  1.00
## dti            -0.42  0.18 -0.42 -0.72  1.00
## inq_last_6mths -0.51  0.06 -0.49 -0.23 -0.03  1.00
## open_acc       -0.46 -0.48 -0.49  0.04  0.23  0.15  1.00
## revol_bal       0.26 -0.31  0.23  0.39  0.00 -0.45  0.12  1.00
## revol_util      0.12  0.64  0.14 -0.31  0.19 -0.28 -0.63  0.17  1.00
## total_acc      -0.28 -0.69 -0.31  0.35 -0.03  0.06  0.83  0.15 -0.67  1.00
## total_pymnt     0.98  0.06  0.98  0.42 -0.45 -0.53 -0.49  0.22  0.09 -0.29
## total_rec_prncp 0.94 -0.10  0.94  0.47 -0.50 -0.54 -0.44  0.22 -0.01 -0.21
## total_rec_int   0.86  0.51  0.86  0.13 -0.20 -0.36 -0.53  0.16  0.37 -0.47
## last_pymnt_amnt 0.39 -0.16  0.37  0.14 -0.46 -0.35 -0.26 -0.20 -0.27 -0.11
##                ttl_p
## loan_amnt
## int_rate
## installment
## annual_inc
## dti
## inq_last_6mths
## open_acc
## revol_bal
## revol_util
## total_acc
## total_pymnt     1.00
## total_rec_prncp 0.98
## total_rec_int   0.82
## last_pymnt_amnt 0.49
##                ttl_rc_p ttl_rc_n lst__
```

```
## total_rec_prncp  1.00
## total_rec_int    0.69     1.00
## last_pymnt_amnt  0.59     0.09     1.00
```

```
KMO(z.df.outl.corr)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = z.df.outl.corr)
## Overall MSA =  0.66
## MSA for each item =
##       loan_amnt        int_rate     installment      annual_inc             dti
##            0.76            0.57            0.76            0.76            0.56
##  inq_last_6mths        open_acc       revol_bal      revol_util       total_acc
##            0.58            0.57            0.79            0.61            0.67
##     total_pymnt total_rec_prncp   total_rec_int last_pymnt_amnt
##            0.65            0.61            0.58            0.52
```

```
cortest.bartlett(z.df.outl.corr, n=480)
```

```
## $chisq
## [1] 6981.34
##
## $p.value
## [1] 0
##
## $df
## [1] 91
```

```
# The overall KMO of data which remove outlier also above 0.5, and Bartlett test also P-value 0 (below
```

PCA

```
# Create PCA for data without remove outlier
m.pc1 <-principal(z.df, 14, rotate="none",weights=TRUE, scores=TRUE)
print(m.pc1)
```

```
## Principal Components Analysis
## Call: principal(r = z.df, nfactors = 14, rotate = "none", scores = TRUE,
##     weights = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9  PC10
## loan_amnt       0.95 -0.08 -0.08  0.04 -0.03 -0.15  0.04 -0.02  0.01  0.03
## int_rate        0.40 -0.41  0.61  0.23 -0.10  0.13 -0.37 -0.16  0.09 -0.05
## installment     0.93 -0.10 -0.11  0.04 -0.02 -0.14  0.09  0.03 -0.04  0.10
## annual_inc      0.50  0.43 -0.27 -0.21 -0.50  0.14 -0.05  0.09  0.40  0.03
## dti             0.10  0.10  0.70 -0.14  0.50 -0.15  0.34  0.10  0.28 -0.03
## inq_last_6mths  0.06  0.20  0.30  0.70 -0.33  0.30  0.41  0.01 -0.05  0.01
## open_acc        0.20  0.77  0.32  0.11  0.13 -0.11 -0.24 -0.16 -0.06  0.36
## revol_bal       0.50  0.33  0.16 -0.53 -0.06  0.30  0.22 -0.40 -0.15 -0.13
```

```
## revol_util      0.35 -0.38  0.50 -0.39 -0.15  0.36 -0.06  0.35 -0.14  0.16
## total_acc       0.30  0.79  0.16  0.06  0.04 -0.05 -0.20  0.32 -0.16 -0.29
## total_pymnt     0.95 -0.12 -0.17  0.07  0.09 -0.10  0.07  0.04 -0.06  0.01
## total_rec_prncp 0.86 -0.06 -0.35  0.05  0.19  0.01  0.13  0.11 -0.08  0.08
## total_rec_int   0.77 -0.22  0.28  0.09 -0.18 -0.32 -0.12 -0.12  0.02 -0.18
## last_pymnt_amnt 0.45  0.00 -0.30  0.22  0.52  0.56 -0.19 -0.05  0.12 -0.07
##                  PC11  PC12  PC13  PC14 h2       u2 com
## loan_amnt       -0.11 -0.20 -0.10  0.00  1  5.6e-16 1.2
## int_rate        -0.18  0.15 -0.01  0.00  1  2.4e-15 4.7
## installment     -0.21 -0.10  0.10  0.00  1 -2.0e-15 1.3
## annual_inc       0.02  0.04  0.00  0.00  1  1.7e-15 5.2
## dti             -0.01  0.01  0.00  0.00  1 -1.8e-15 3.1
## inq_last_6mths   0.03 -0.02  0.00  0.00  1  3.3e-16 3.3
## open_acc         0.07  0.00  0.00  0.00  1 -3.3e-15 2.5
## revol_bal       -0.03  0.02  0.00  0.00  1 -2.2e-16 5.3
## revol_util       0.09 -0.06  0.00  0.00  1  2.3e-15 6.5
## total_acc       -0.07  0.01  0.00  0.00  1 -6.7e-16 2.4
## total_pymnt      0.09  0.13 -0.01 -0.02  1 -2.2e-15 1.2
## total_rec_prncp  0.02  0.22 -0.02  0.02  1 -1.3e-15 1.7
## total_rec_int    0.27 -0.07  0.03  0.01  1 -8.9e-16 2.7
## last_pymnt_amnt  0.07 -0.11  0.02  0.00  1 -6.7e-16 4.5
##
##                        PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9 PC10 PC11
## SS loadings           5.12 1.96 1.76 1.13 1.00 0.86 0.65 0.49 0.35 0.31 0.19
## Proportion Var        0.37 0.14 0.13 0.08 0.07 0.06 0.05 0.03 0.02 0.02 0.01
## Cumulative Var        0.37 0.51 0.63 0.71 0.78 0.85 0.89 0.93 0.95 0.97 0.99
## Proportion Explained  0.37 0.14 0.13 0.08 0.07 0.06 0.05 0.03 0.02 0.02 0.01
## Cumulative Proportion 0.37 0.51 0.63 0.71 0.78 0.85 0.89 0.93 0.95 0.97 0.99
##                       PC12 PC13 PC14
## SS loadings           0.16 0.02    0
## Proportion Var        0.01 0.00    0
## Cumulative Var        1.00 1.00    1
## Proportion Explained  0.01 0.00    0
## Cumulative Proportion 1.00 1.00    1
##
## Mean item complexity =  3.3
## Test of the hypothesis that 14 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

```
fscore_pc1 <- m.pc1$scores
fscorematrix <- cor(fscore_pc1)
lowerCor(fscore_pc1)
```

```
##      PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9 PC10 PC11
## PC1  1
## PC2  0   1
## PC3  0   0   1
## PC4  0   0   0   1
## PC5  0   0   0   0   1
## PC6  0   0   0   0   0   1
```

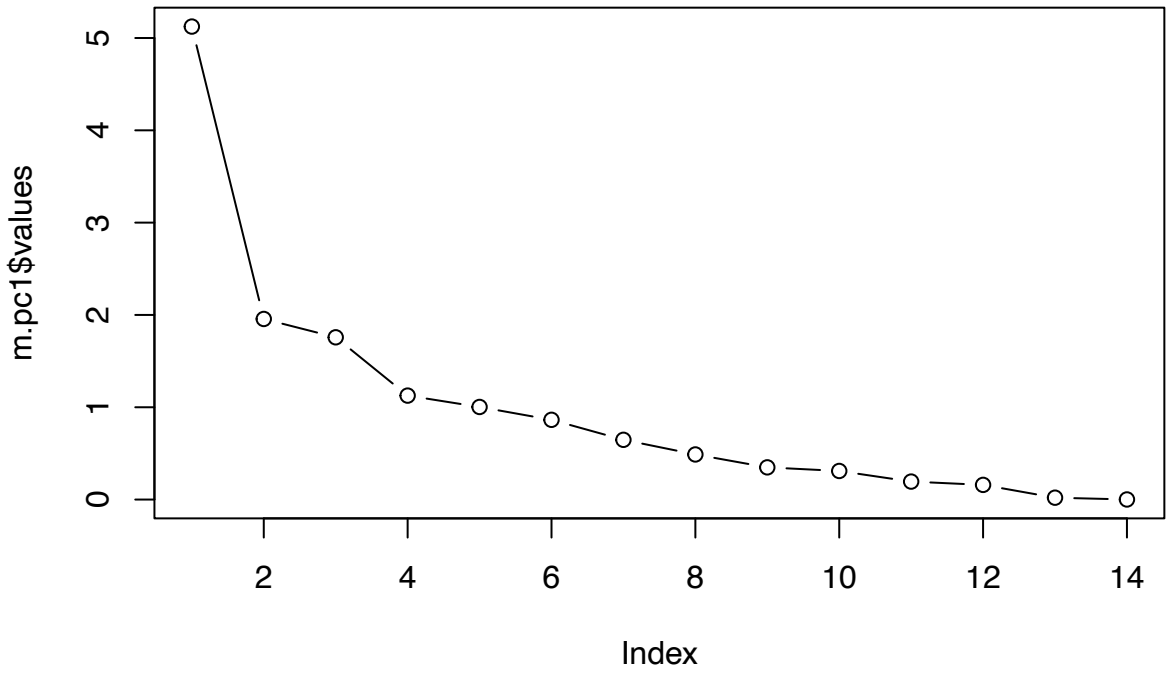
```
## PC7  0   0   0   0   0   0   1
## PC8  0   0   0   0   0   0   0   1
## PC9  0   0   0   0   0   0   0   0   1
## PC10 0   0   0   0   0   0   0   0   0   1
## PC11 0   0   0   0   0   0   0   0   0   0    1
## PC12 0   0   0   0   0   0   0   0   0   0    0
## PC13 0   0   0   0   0   0   0   0   0   0    0
## PC14 0   0   0   0   0   0   0   0   0   0    0
##      PC12 PC13 PC14
## PC12 1
## PC13 0    1
## PC14 0    0    1
```

```
print.psych(m.pc1, cut=0.4, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = z.df, nfactors = 14, rotate = "none", scores = TRUE,
##     weights = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  item  PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9 PC10
## loan_amnt           1 0.95
## total_pymnt        11 0.95
## installment         3 0.93
## total_rec_prncp    12 0.86
## total_rec_int      13 0.77
## annual_inc          4 0.50  0.43             -0.50                    0.40
## total_acc          10       0.79
## open_acc            7       0.77
## dti                 5             0.70        0.50
## int_rate            2      -0.41  0.61
## revol_util          9             0.50
## inq_last_6mths      6                   0.70              0.41
## revol_bal           8 0.50             -0.53                   -0.40
## last_pymnt_amnt    14 0.45                    0.52  0.56
##                  PC11 PC12 PC13 PC14 h2       u2 com
## loan_amnt                          1  5.6e-16 1.2
## total_pymnt                        1 -2.2e-15 1.2
## installment                        1 -2.0e-15 1.3
## total_rec_prncp                    1 -1.3e-15 1.7
## total_rec_int                      1 -8.9e-16 2.7
## annual_inc                         1  1.7e-15 5.2
## total_acc                          1 -6.7e-16 2.4
## open_acc                           1 -3.3e-15 2.5
## dti                                1 -1.8e-15 3.1
## int_rate                           1  2.4e-15 4.7
## revol_util                         1  2.3e-15 6.5
## inq_last_6mths                     1  3.3e-16 3.3
## revol_bal                          1 -2.2e-16 5.3
## last_pymnt_amnt                    1 -6.7e-16 4.5
##
##                        PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9 PC10 PC11
## SS loadings           5.12 1.96 1.76 1.13 1.00 0.86 0.65 0.49 0.35 0.31 0.19
## Proportion Var        0.37 0.14 0.13 0.08 0.07 0.06 0.05 0.03 0.02 0.02 0.01
## Cumulative Var        0.37 0.51 0.63 0.71 0.78 0.85 0.89 0.93 0.95 0.97 0.99
```

```
## Proportion Explained  0.37 0.14 0.13 0.08 0.07 0.06 0.05 0.03 0.02 0.02 0.01
## Cumulative Proportion 0.37 0.51 0.63 0.71 0.78 0.85 0.89 0.93 0.95 0.97 0.99
##                       PC12 PC13 PC14
## SS loadings           0.16 0.02    0
## Proportion Var        0.01 0.00    0
## Cumulative Var        1.00 1.00    1
## Proportion Explained  0.01 0.00    0
## Cumulative Proportion 1.00 1.00    1
##
## Mean item complexity =  3.3
## Test of the hypothesis that 14 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

```
plot(m.pc1$values,type="b")
```




```
# Create PCA for data which remove outlier
m.pc1o <- principal(z.outl.df, 14, rotate="none",weights=TRUE, scores=TRUE)
print(m.pc1o)
```

```
## Principal Components Analysis
## Call: principal(r = z.outl.df, nfactors = 14, rotate = "none", scores = TRUE,
```

```
##     weights = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##                   PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9  PC10
## loan_amnt        0.94 -0.08 -0.09  0.01  0.01 -0.16  0.01 -0.01  0.00  0.04
## int_rate         0.36 -0.37  0.64  0.31  0.09  0.12 -0.36 -0.09  0.10 -0.04
## installment      0.93 -0.12 -0.10  0.02  0.02 -0.18  0.05  0.04 -0.08  0.12
## annual_inc       0.47  0.41 -0.33 -0.19  0.50  0.17 -0.06  0.17  0.39  0.05
## dti              0.12  0.14  0.68 -0.14 -0.49 -0.17  0.33  0.20  0.26 -0.01
## inq_last_6mths   0.06  0.23  0.33  0.63  0.47  0.04  0.45 -0.02 -0.07  0.02
## open_acc         0.18  0.79  0.29  0.10 -0.15 -0.09 -0.26 -0.17 -0.03  0.34
## revol_bal        0.50  0.33  0.23 -0.51  0.09  0.22  0.20 -0.44 -0.05 -0.14
## revol_util       0.35 -0.34  0.53 -0.34  0.16  0.44 -0.02  0.27 -0.18  0.15
## total_acc        0.27  0.81  0.09  0.06 -0.04  0.02 -0.18  0.30 -0.21 -0.28
## total_pymnt      0.95 -0.12 -0.19  0.04 -0.09 -0.08  0.06  0.02 -0.05  0.00
## total_rec_prncp  0.87 -0.06 -0.34  0.03 -0.17  0.01  0.14  0.06 -0.10  0.07
## total_rec_int    0.80 -0.21  0.25  0.06  0.12 -0.27 -0.17 -0.08  0.10 -0.19
## last_pymnt_amnt  0.41  0.04 -0.26  0.40 -0.46  0.60  0.00 -0.07  0.12 -0.05
##                   PC11  PC12  PC13  PC14 h2       u2 com
## loan_amnt        -0.18 -0.16 -0.10  0.00  1  6.7e-16 1.3
## int_rate         -0.11  0.20 -0.01  0.00  1 -1.3e-15 4.3
## installment      -0.21 -0.02  0.10  0.00  1 -2.2e-15 1.3
## annual_inc        0.00  0.04  0.00  0.00  1  1.3e-15 5.7
## dti              -0.02  0.02  0.00  0.00  1 -8.9e-16 3.5
## inq_last_6mths    0.03 -0.02  0.00  0.00  1 -2.9e-15 3.8
## open_acc          0.08 -0.03  0.00  0.00  1 -4.4e-16 2.4
## revol_bal        -0.04  0.04  0.00  0.00  1 -2.2e-16 5.2
## revol_util        0.07 -0.09  0.00  0.00  1  8.9e-16 6.0
## total_acc        -0.05  0.02  0.00  0.00  1  0.0e+00 2.2
## total_pymnt       0.14  0.10 -0.01 -0.02  1 -2.2e-15 1.2
## total_rec_prncp   0.12  0.20 -0.02  0.02  1 -2.0e-15 1.7
## total_rec_int     0.22 -0.17  0.03  0.01  1 -2.4e-15 2.3
## last_pymnt_amnt   0.00 -0.11  0.02  0.00  1  1.6e-15 4.3
##
##                        PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9 PC10 PC11
## SS loadings           5.07 1.98 1.81 1.11 1.04 0.82 0.64 0.48 0.36 0.30 0.19
## Proportion Var        0.36 0.14 0.13 0.08 0.07 0.06 0.05 0.03 0.03 0.02 0.01
## Cumulative Var        0.36 0.50 0.63 0.71 0.79 0.85 0.89 0.93 0.95 0.97 0.99
## Proportion Explained  0.36 0.14 0.13 0.08 0.07 0.06 0.05 0.03 0.03 0.02 0.01
## Cumulative Proportion 0.36 0.50 0.63 0.71 0.79 0.85 0.89 0.93 0.95 0.97 0.99
##                       PC12 PC13 PC14
## SS loadings           0.17 0.02    0
## Proportion Var        0.01 0.00    0
## Cumulative Var        1.00 1.00    1
## Proportion Explained  0.01 0.00    0
## Cumulative Proportion 1.00 1.00    1
##
## Mean item complexity =  3.2
## Test of the hypothesis that 14 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

```r
fscore_pc1o <- m.pc1o$scores
fscorematrix <- cor(fscore_pc1o)
lowerCor(fscore_pc1o)
```

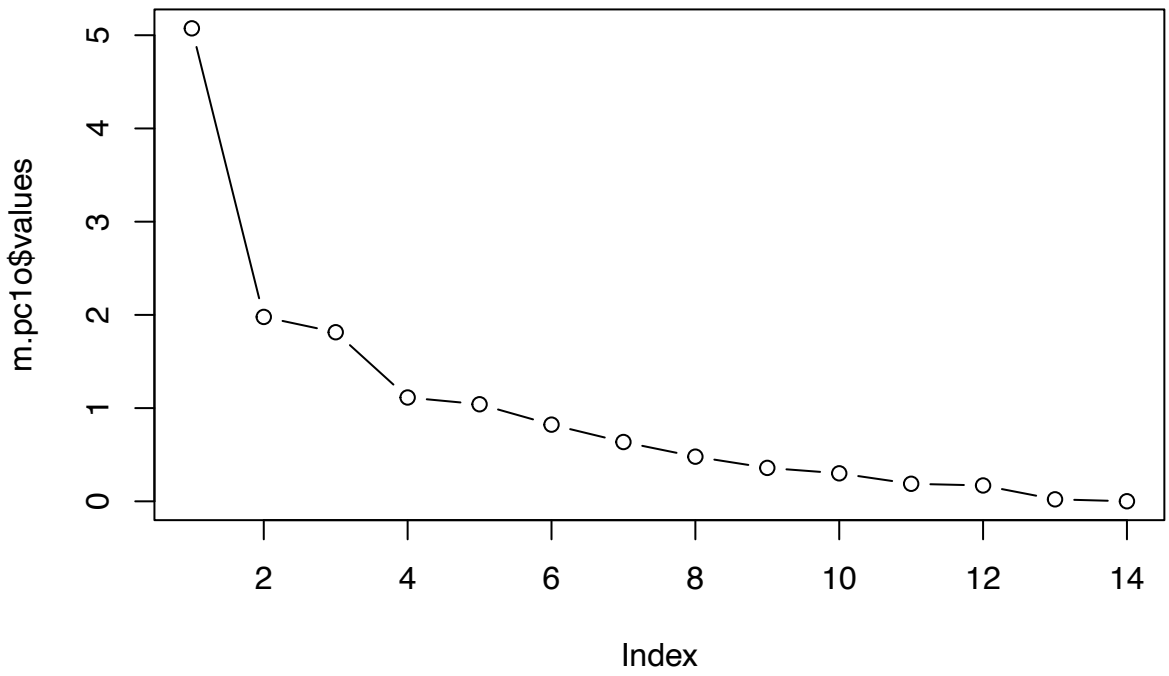
```
##      PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9 PC10 PC11
## PC1  1
## PC2  0   1
## PC3  0   0   1
## PC4  0   0   0   1
## PC5  0   0   0   0   1
## PC6  0   0   0   0   0   1
## PC7  0   0   0   0   0   0   1
## PC8  0   0   0   0   0   0   0   1
## PC9  0   0   0   0   0   0   0   0   1
## PC10 0   0   0   0   0   0   0   0   0   1
## PC11 0   0   0   0   0   0   0   0   0   0    1
## PC12 0   0   0   0   0   0   0   0   0   0    0
## PC13 0   0   0   0   0   0   0   0   0   0    0
## PC14 0   0   0   0   0   0   0   0   0   0    0
##      PC12 PC13 PC14
## PC12 1
## PC13 0    1
## PC14 0    0    1
```

```r
print.psych(m.pc1o, cut=0.4, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = z.outl.df, nfactors = 14, rotate = "none", scores = TRUE,
##     weights = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  item  PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8  PC9 PC10
## total_pymnt        11 0.95
## loan_amnt           1 0.94
## installment         3 0.93
## total_rec_prncp    12 0.87
## total_rec_int      13 0.80
## total_acc          10       0.81
## open_acc            7       0.79
## dti                 5             0.68       -0.49
## int_rate            2             0.64
## revol_util          9             0.53              0.44
## inq_last_6mths      6                   0.63  0.47        0.45
## revol_bal           8 0.50             -0.51                   -0.44
## annual_inc          4 0.47  0.41              0.50
## last_pymnt_amnt    14 0.41              0.40 -0.46  0.60
##                  PC11 PC12 PC13 PC14 h2       u2 com
## total_pymnt                          1 -2.2e-15 1.2
## loan_amnt                            1  6.7e-16 1.3
## installment                          1 -2.2e-15 1.3
## total_rec_prncp                      1 -2.0e-15 1.7
## total_rec_int                        1 -2.4e-15 2.3
## total_acc                            1  0.0e+00 2.2
```

```
## open_acc                            1 -4.4e-16 2.4
## dti                                 1 -8.9e-16 3.5
## int_rate                            1 -1.3e-15 4.3
## revol_util                          1  8.9e-16 6.0
## inq_last_6mths                      1 -2.9e-15 3.8
## revol_bal                           1 -2.2e-16 5.2
## annual_inc                          1  1.3e-15 5.7
## last_pymnt_amnt                     1  1.6e-15 4.3
##
##                         PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9 PC10 PC11
## SS loadings            5.07 1.98 1.81 1.11 1.04 0.82 0.64 0.48 0.36 0.30 0.19
## Proportion Var         0.36 0.14 0.13 0.08 0.07 0.06 0.05 0.03 0.03 0.02 0.01
## Cumulative Var         0.36 0.50 0.63 0.71 0.79 0.85 0.89 0.93 0.95 0.97 0.99
## Proportion Explained   0.36 0.14 0.13 0.08 0.07 0.06 0.05 0.03 0.03 0.02 0.01
## Cumulative Proportion  0.36 0.50 0.63 0.71 0.79 0.85 0.89 0.93 0.95 0.97 0.99
##                        PC12 PC13 PC14
## SS loadings            0.17 0.02    0
## Proportion Var         0.01 0.00    0
## Cumulative Var         1.00 1.00    1
## Proportion Explained   0.01 0.00    0
## Cumulative Proportion  1.00 1.00    1
##
## Mean item complexity =  3.2
## Test of the hypothesis that 14 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

```
plot(m.pc1o$values,type="b")
```



The results of PCA, for both remove and keep outlier is not much different, but for the removed outlier data, PC tend to explain more variance than the data that keep outlier.

FA

FA on PC extraction

```
# Try FA on data without removing outlier --- FA on PC - Oblique Rotation
m.FAPC <-principal(z.df, 14, rotate="oblimin")
```

```
## Warning in GPFoblq(A, Tmat = Tmat, normalize = normalize, eps = eps, maxit =
## maxit, : convergence not obtained in GPFoblq. 1000 iterations used.
```

```
print(m.FAPC)
```

```
## Principal Components Analysis
## Call: principal(r = z.df, nfactors = 14, rotate = "oblimin")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                   TC1 TC12   TC7  TC5  TC8  TC2  TC9 TC3 TC10 TC4   TC6  TC11
## loan_amnt        0.91 0.01  0.01 0.02 0.01 0.01 0.00   0 0.01   0  0.02  0.03
## int_rate         0.00 0.00  1.00 0.00 0.00 0.00 0.00   0 0.00   0  0.00  0.00
## installment      0.99 0.02  0.00 0.00 0.00 0.00 0.01   0 0.00   0  0.00 -0.01
## annual_inc       0.00 0.00  0.00 1.00 0.00 0.00 0.00   0 0.00   0  0.00  0.00
## dti              0.00 0.00  0.00 0.00 0.00 0.00 0.00   1 0.00   0  0.00  0.00
```

```
## inq_last_6mths   0.00 0.00  0.00 0.00 0.00 0.00 0.00   0 0.00   1  0.00  0.00
## open_acc         0.00 0.00  0.00 0.00 0.00 0.00 0.00   0 1.00   0  0.00  0.00
## revol_bal        0.00 0.00  0.00 0.00 1.00 0.00 0.00   0 0.00   0  0.00  0.00
## revol_util       0.00 0.00  0.00 0.00 0.00 0.00 1.00   0 0.00   0  0.00  0.00
## total_acc        0.00 0.00  0.00 0.00 0.00 1.00 0.00   0 0.00   0  0.00  0.00
## total_pymnt      0.07 0.84  0.05 0.01 0.01 0.01 0.00   0 0.00   0 -0.02  0.13
## total_rec_prncp -0.02 1.03 -0.02 0.00 0.00 0.00 0.00   0 0.00   0  0.02 -0.06
## total_rec_int    0.00 0.01  0.00 0.00 0.00 0.00 0.01   0 0.00   0  0.00  0.99
## last_pymnt_amnt  0.00 0.00  0.00 0.00 0.00 0.00 0.00   0 0.00   0  1.00  0.00
##                  TC13  TC14 h2       u2 com
## loan_amnt        0.14  0.01  1  5.6e-16 1.1
## int_rate         0.00  0.00  1  2.4e-15 1.0
## installment     -0.11 -0.01  1 -2.0e-15 1.0
## annual_inc       0.00  0.00  1  1.7e-15 1.0
## dti              0.00  0.00  1 -1.8e-15 1.0
## inq_last_6mths   0.00  0.00  1  3.3e-16 1.0
## open_acc         0.00  0.00  1 -3.3e-15 1.0
## revol_bal        0.00  0.00  1 -2.2e-16 1.0
## revol_util       0.00  0.00  1  2.3e-15 1.0
## total_acc        0.00  0.00  1 -6.7e-16 1.0
## total_pymnt      0.02  0.06  1 -2.2e-15 1.1
## total_rec_prncp -0.01 -0.03  1 -1.3e-15 1.0
## total_rec_int    0.00  0.00  1 -8.9e-16 1.0
## last_pymnt_amnt  0.00  0.00  1 -6.7e-16 1.0
##
##                         TC1 TC12  TC7  TC5  TC8  TC2  TC9  TC3 TC10  TC4  TC6
## SS loadings            1.94 1.88 1.02 1.01 1.01 1.00 1.01 1.00 1.00 1.00 1.01
## Proportion Var         0.14 0.13 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07
## Cumulative Var         0.14 0.27 0.35 0.42 0.49 0.56 0.63 0.71 0.78 0.85 0.92
## Proportion Explained   0.14 0.13 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07
## Cumulative Proportion  0.14 0.27 0.35 0.42 0.49 0.56 0.63 0.71 0.78 0.85 0.92
##                        TC11 TC13 TC14
## SS loadings            1.07 0.04 0.01
## Proportion Var         0.08 0.00 0.00
## Cumulative Var         1.00 1.00 1.00
## Proportion Explained   0.08 0.00 0.00
## Cumulative Proportion  1.00 1.00 1.00
##
##  With component correlations of
##       TC1  TC12   TC7   TC5   TC8   TC2   TC9   TC3  TC10   TC4   TC6 TC11
## TC1  1.00  0.86  0.32  0.42  0.36  0.19  0.25  0.04  0.10  0.02  0.34 0.71
## TC12 0.86  1.00  0.16  0.38  0.32  0.17  0.17 -0.02  0.03 -0.03  0.51 0.56
## TC7  0.32  0.16  1.00 -0.08  0.07 -0.06  0.49  0.21  0.03  0.19  0.13 0.58
## TC5  0.42  0.38 -0.08  1.00  0.42  0.38  0.07 -0.22  0.23  0.05  0.13 0.23
## TC8  0.36  0.32  0.07  0.42  1.00  0.28  0.29  0.19  0.28 -0.03  0.16 0.25
## TC2  0.19  0.17 -0.06  0.38  0.28  1.00 -0.06  0.17  0.63  0.16  0.11 0.13
## TC9  0.25  0.17  0.49  0.07  0.29 -0.06  1.00  0.24 -0.13 -0.03  0.02 0.32
## TC3  0.04 -0.02  0.21 -0.22  0.19  0.17  0.24  1.00  0.26  0.05 -0.06 0.16
## TC10 0.10  0.03  0.03  0.23  0.28  0.63 -0.13  0.26  1.00  0.17  0.05 0.09
## TC4  0.02 -0.03  0.19  0.05 -0.03  0.16 -0.03  0.05  0.17  1.00  0.01 0.06
## TC6  0.34  0.51  0.13  0.13  0.16  0.11  0.02 -0.06  0.05  0.01  1.00 0.06
## TC11 0.71  0.56  0.58  0.23  0.25  0.13  0.32  0.16  0.09  0.06  0.06 1.00
## TC13 0.11  0.00  0.04  0.08  0.06  0.03 -0.02  0.02  0.04 -0.04  0.11 0.28
## TC14 0.06 -0.03  0.08 -0.05 -0.04  0.00 -0.02  0.01  0.01  0.00 -0.10 0.52
```

```
##       TC13  TC14
## TC1   0.11  0.06
## TC12  0.00 -0.03
## TC7   0.04  0.08
## TC5   0.08 -0.05
## TC8   0.06 -0.04
## TC2   0.03  0.00
## TC9  -0.02 -0.02
## TC3   0.02  0.01
## TC10  0.04  0.01
## TC4  -0.04  0.00
## TC6   0.11 -0.10
## TC11  0.28  0.52
## TC13  1.00  0.13
## TC14  0.13  1.00
##
## Mean item complexity =  1
## Test of the hypothesis that 14 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

```
fscore_FAPC <- m.FAPC$scores
fscorematrix <- cor(fscore_FAPC)
lowerCor(fscore_FAPC)
```

```
##      TC1   TC12  TC7   TC5   TC8   TC2   TC9   TC3   TC10  TC4   TC6
## TC1   1.00
## TC12  0.86  1.00
## TC7   0.32  0.16  1.00
## TC5   0.42  0.38 -0.08  1.00
## TC8   0.36  0.32  0.07  0.42  1.00
## TC2   0.19  0.17 -0.06  0.38  0.28  1.00
## TC9   0.25  0.17  0.49  0.07  0.29 -0.06  1.00
## TC3   0.04 -0.02  0.21 -0.22  0.19  0.17  0.24  1.00
## TC10  0.10  0.03  0.03  0.23  0.28  0.63 -0.13  0.26  1.00
## TC4   0.02 -0.03  0.19  0.05 -0.03  0.16 -0.03  0.05  0.17  1.00
## TC6   0.34  0.51  0.13  0.13  0.16  0.11  0.02 -0.06  0.05  0.01  1.00
## TC11  0.71  0.56  0.58  0.23  0.25  0.13  0.32  0.16  0.09  0.06  0.06
## TC13  0.11  0.00  0.04  0.08  0.06  0.03 -0.02  0.02  0.04 -0.04  0.11
## TC14  0.06 -0.03  0.08 -0.05 -0.04  0.00 -0.02  0.01  0.01  0.00 -0.10
##      TC11  TC13  TC14
## TC11  1.00
## TC13  0.28  1.00
## TC14  0.52  0.13  1.00
```

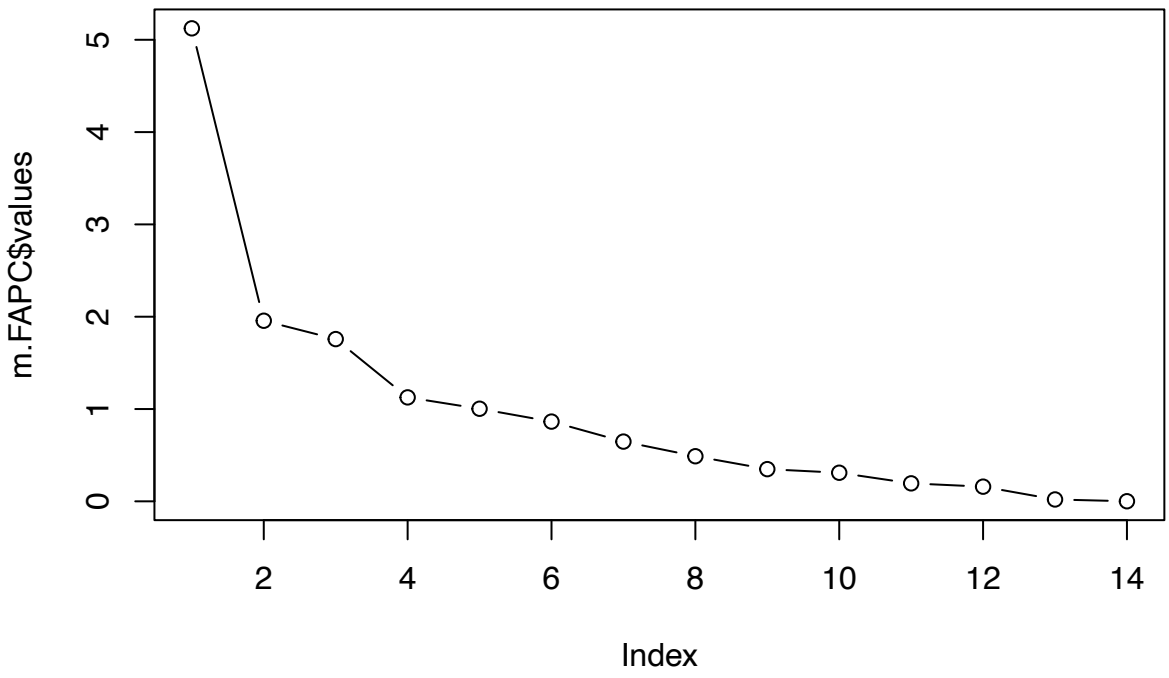
```
print.psych(m.FAPC, cut=0.4, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = z.df, nfactors = 14, rotate = "oblimin")
## Standardized loadings (pattern matrix) based upon correlation matrix
```

```
##                  item   TC1  TC12   TC7   TC5   TC8   TC2   TC9   TC3  TC10
## installment         3  0.99
## loan_amnt           1  0.91
## total_rec_prncp    12        1.03
## total_pymnt        11        0.84
## int_rate            2              1.00
## annual_inc          4                    1.00
## revol_bal           8                          1.00
## total_acc          10                                1.00
## revol_util          9                                      1.00
## dti                 5                                            1
## open_acc            7                                               1.00
## inq_last_6mths      6
## last_pymnt_amnt    14
## total_rec_int      13
##                   TC4   TC6  TC11  TC13  TC14 h2       u2 com
## installment                                   1 -2.0e-15 1.0
## loan_amnt                                     1  5.6e-16 1.1
## total_rec_prncp                               1 -1.3e-15 1.0
## total_pymnt                                   1 -2.2e-15 1.1
## int_rate                                      1  2.4e-15 1.0
## annual_inc                                    1  1.7e-15 1.0
## revol_bal                                     1 -2.2e-16 1.0
## total_acc                                     1 -6.7e-16 1.0
## revol_util                                    1  2.3e-15 1.0
## dti                                           1 -1.8e-15 1.0
## open_acc                                      1 -3.3e-15 1.0
## inq_last_6mths      1                         1  3.3e-16 1.0
## last_pymnt_amnt        1.00                   1 -6.7e-16 1.0
## total_rec_int                0.99             1 -8.9e-16 1.0
##
##                        TC1 TC12  TC7  TC5  TC8  TC2  TC9  TC3 TC10  TC4  TC6
## SS loadings           1.94 1.88 1.02 1.01 1.01 1.00 1.01 1.00 1.00 1.00 1.01
## Proportion Var        0.14 0.13 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07
## Cumulative Var        0.14 0.27 0.35 0.42 0.49 0.56 0.63 0.71 0.78 0.85 0.92
## Proportion Explained  0.14 0.13 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07
## Cumulative Proportion 0.14 0.27 0.35 0.42 0.49 0.56 0.63 0.71 0.78 0.85 0.92
##                       TC11 TC13 TC14
## SS loadings           1.07 0.04 0.01
## Proportion Var        0.08 0.00 0.00
## Cumulative Var        1.00 1.00 1.00
## Proportion Explained  0.08 0.00 0.00
## Cumulative Proportion 1.00 1.00 1.00
##
##  With component correlations of
##       TC1  TC12   TC7   TC5   TC8   TC2   TC9   TC3  TC10   TC4   TC6 TC11
## TC1  1.00  0.86  0.32  0.42  0.36  0.19  0.25  0.04  0.10  0.02  0.34 0.71
## TC12 0.86  1.00  0.16  0.38  0.32  0.17  0.17 -0.02  0.03 -0.03  0.51 0.56
## TC7  0.32  0.16  1.00 -0.08  0.07 -0.06  0.49  0.21  0.03  0.19  0.13 0.58
## TC5  0.42  0.38 -0.08  1.00  0.42  0.38  0.07 -0.22  0.23  0.05  0.13 0.23
## TC8  0.36  0.32  0.07  0.42  1.00  0.28  0.29  0.19  0.28 -0.03  0.16 0.25
## TC2  0.19  0.17 -0.06  0.38  0.28  1.00 -0.06  0.17  0.63  0.16  0.11 0.13
## TC9  0.25  0.17  0.49  0.07  0.29 -0.06  1.00  0.24 -0.13 -0.03  0.02 0.32
## TC3  0.04 -0.02  0.21 -0.22  0.19  0.17  0.24  1.00  0.26  0.05 -0.06 0.16
```

```
## TC10 0.10  0.03  0.03  0.23  0.28  0.63 -0.13  0.26  1.00  0.17  0.05 0.09
## TC4  0.02 -0.03  0.19  0.05 -0.03  0.16 -0.03  0.05  0.17  1.00  0.01 0.06
## TC6  0.34  0.51  0.13  0.13  0.16  0.11  0.02 -0.06  0.05  0.01  1.00 0.06
## TC11 0.71  0.56  0.58  0.23  0.25  0.13  0.32  0.16  0.09  0.06  0.06 1.00
## TC13 0.11  0.00  0.04  0.08  0.06  0.03 -0.02  0.02  0.04 -0.04  0.11 0.28
## TC14 0.06 -0.03  0.08 -0.05 -0.04  0.00 -0.02  0.01  0.01  0.00 -0.10 0.52
##       TC13  TC14
## TC1   0.11  0.06
## TC12  0.00 -0.03
## TC7   0.04  0.08
## TC5   0.08 -0.05
## TC8   0.06 -0.04
## TC2   0.03  0.00
## TC9  -0.02 -0.02
## TC3   0.02  0.01
## TC10  0.04  0.01
## TC4  -0.04  0.00
## TC6   0.11 -0.10
## TC11  0.28  0.52
## TC13  1.00  0.13
## TC14  0.13  1.00
##
## Mean item complexity =  1
## Test of the hypothesis that 14 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

```
plot(m.FAPC$values,type="b")
```




```
# Try FA on data which removing outlier --- FA on PC - Oblique Rotation
m.FAPCo <-principal(z.outl.df, 14, rotate="oblimin")
```

```
## Warning in GPFoblq(A, Tmat = Tmat, normalize = normalize, eps = eps, maxit =
## maxit, : convergence not obtained in GPFoblq. 1000 iterations used.
```

```
print(m.FAPCo)
```

```
## Principal Components Analysis
## Call: principal(r = z.outl.df, nfactors = 14, rotate = "oblimin")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                    TC1 TC12   TC3  TC8  TC5   TC6  TC2 TC7 TC10  TC9   TC4  TC11
## loan_amnt         0.91 0.01  0.00 0.02 0.02  0.02 0.01   0 0.01 0.00 -0.01  0.03
## int_rate          0.00 0.00  1.00 0.00 0.00  0.00 0.00   0 0.00 0.00  0.00  0.00
## installment       0.99 0.02  0.01 0.00 0.00 -0.01 0.00   0 0.00 0.01  0.00 -0.01
## annual_inc        0.00 0.00  0.00 0.00 1.00  0.00 0.00   0 0.00 0.00  0.00  0.00
## dti               0.00 0.00  0.00 0.00 0.00  0.00 0.00   1 0.00 0.00  0.00  0.00
## inq_last_6mths    0.00 0.00  0.00 0.00 0.00  0.00 0.00   0 0.00 0.00  1.00  0.00
## open_acc          0.00 0.00  0.00 0.00 0.00  0.00 0.00   0 1.00 0.00  0.00  0.00
## revol_bal         0.00 0.00  0.00 1.00 0.00  0.00 0.00   0 0.00 0.00  0.00  0.00
## revol_util        0.00 0.00  0.00 0.00 0.00  0.00 0.00   0 0.00 1.00  0.00  0.00
## total_acc         0.00 0.00  0.00 0.00 0.00  0.00 1.00   0 0.00 0.00  0.00  0.00
## total_pymnt       0.06 0.86  0.04 0.01 0.01 -0.01 0.00   0 0.00 0.00  0.00  0.12
## total_rec_prncp -0.02 1.03 -0.02 0.00 0.00  0.02 0.00   0 0.00 0.00  0.00 -0.06
## total_rec_int     0.00 0.01  0.00 0.00 0.00  0.00 0.00   0 0.00 0.01  0.00  0.99
```

```
## last_pymnt_amnt  0.00 0.00  0.00 0.00 0.00  1.00 0.00   0 0.00 0.00  0.00  0.00
##                  TC13  TC14 h2       u2 com
## loan_amnt        0.15  0.00  1  6.7e-16 1.1
## int_rate         0.00  0.00  1 -1.3e-15 1.0
## installment     -0.11 -0.01  1 -2.2e-15 1.0
## annual_inc       0.00  0.00  1  1.3e-15 1.0
## dti              0.00  0.00  1 -8.9e-16 1.0
## inq_last_6mths   0.00  0.00  1 -2.9e-15 1.0
## open_acc         0.00  0.00  1 -4.4e-16 1.0
## revol_bal        0.00  0.00  1 -2.2e-16 1.0
## revol_util       0.00  0.00  1  8.9e-16 1.0
## total_acc        0.00  0.00  1  0.0e+00 1.0
## total_pymnt      0.02  0.05  1 -2.2e-15 1.1
## total_rec_prncp -0.01 -0.03  1 -2.0e-15 1.0
## total_rec_int    0.00  0.00  1 -2.4e-15 1.0
## last_pymnt_amnt  0.00  0.00  1  1.6e-15 1.0
##
##                        TC1 TC12  TC3  TC8  TC5  TC6  TC2  TC7 TC10  TC9  TC4
## SS loadings           1.93 1.90 1.02 1.01 1.01 1.01 1.00 1.00 1.00 1.01 1.00
## Proportion Var        0.14 0.14 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07
## Cumulative Var        0.14 0.27 0.35 0.42 0.49 0.56 0.63 0.71 0.78 0.85 0.92
## Proportion Explained  0.14 0.14 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07
## Cumulative Proportion 0.14 0.27 0.35 0.42 0.49 0.56 0.63 0.71 0.78 0.85 0.92
##                       TC11 TC13 TC14
## SS loadings           1.06 0.04    0
## Proportion Var        0.08 0.00    0
## Cumulative Var        1.00 1.00    1
## Proportion Explained  0.08 0.00    0
## Cumulative Proportion 1.00 1.00    1
##
##  With component correlations of
##        TC1  TC12   TC3   TC8   TC5   TC6   TC2   TC7  TC10   TC9   TC4 TC11
## TC1   1.00  0.86  0.29  0.36  0.39  0.29  0.15  0.05  0.07  0.24  0.02 0.73
## TC12  0.86  1.00  0.13  0.32  0.38  0.47  0.15  0.00  0.02  0.17 -0.05 0.61
## TC3   0.29  0.13  1.00  0.07 -0.13  0.12 -0.09  0.21  0.04  0.48  0.22 0.56
## TC8   0.36  0.32  0.07  1.00  0.36  0.08  0.29  0.23  0.31  0.33  0.01 0.31
## TC5   0.39  0.38 -0.13  0.36  1.00  0.12  0.37 -0.24  0.19  0.05  0.08 0.22
## TC6   0.29  0.47  0.12  0.08  0.12  1.00  0.14 -0.04  0.08  0.00  0.01 0.09
## TC2   0.15  0.15 -0.09  0.29  0.37  0.14  1.00  0.17  0.63 -0.08  0.17 0.09
## TC7   0.05  0.00  0.21  0.23 -0.24 -0.04  0.17  1.00  0.28  0.24  0.06 0.16
## TC10  0.07  0.02  0.04  0.31  0.19  0.08  0.63  0.28  1.00 -0.13  0.17 0.07
## TC9   0.24  0.17  0.48  0.33  0.05  0.00 -0.08  0.24 -0.13  1.00  0.00 0.32
## TC4   0.02 -0.05  0.22  0.01  0.08  0.01  0.17  0.06  0.17  0.00  1.00 0.09
## TC11  0.73  0.61  0.56  0.31  0.22  0.09  0.09  0.16  0.07  0.32  0.09 1.00
## TC13  0.10 -0.01  0.01  0.10  0.08  0.13  0.03  0.01  0.03 -0.03 -0.04 0.25
## TC14 -0.11 -0.22  0.03 -0.07 -0.09 -0.14 -0.03  0.01  0.00 -0.05  0.01 0.32
##       TC13  TC14
## TC1   0.10 -0.11
## TC12 -0.01 -0.22
## TC3   0.01  0.03
## TC8   0.10 -0.07
## TC5   0.08 -0.09
## TC6   0.13 -0.14
## TC2   0.03 -0.03
```

```
## TC7   0.01  0.01
## TC10  0.03  0.00
## TC9  -0.03 -0.05
## TC4  -0.04  0.01
## TC11  0.25  0.32
## TC13  1.00  0.10
## TC14  0.10  1.00
##
## Mean item complexity =  1
## Test of the hypothesis that 14 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

```
fscore_FAPCo <- m.FAPCo$scores
fscorematrix <- cor(fscore_FAPCo)
lowerCor(fscore_FAPCo)
```

```
##      TC1   TC12  TC3   TC8   TC5   TC6   TC2   TC7   TC10  TC9   TC4
## TC1   1.00
## TC12  0.86  1.00
## TC3   0.29  0.13  1.00
## TC8   0.36  0.32  0.07  1.00
## TC5   0.39  0.38 -0.13  0.36  1.00
## TC6   0.29  0.47  0.12  0.08  0.12  1.00
## TC2   0.15  0.15 -0.09  0.29  0.37  0.14  1.00
## TC7   0.05  0.00  0.21  0.23 -0.24 -0.04  0.17  1.00
## TC10  0.07  0.02  0.04  0.31  0.19  0.08  0.63  0.28  1.00
## TC9   0.24  0.17  0.48  0.33  0.05  0.00 -0.08  0.24 -0.13  1.00
## TC4   0.02 -0.05  0.22  0.01  0.08  0.01  0.17  0.06  0.17  0.00  1.00
## TC11  0.73  0.61  0.56  0.31  0.22  0.09  0.09  0.16  0.07  0.32  0.09
## TC13  0.10 -0.01  0.01  0.10  0.08  0.13  0.03  0.01  0.03 -0.03 -0.04
## TC14 -0.11 -0.22  0.03 -0.07 -0.09 -0.14 -0.03  0.01  0.00 -0.05  0.01
##      TC11  TC13  TC14
## TC11  1.00
## TC13  0.25  1.00
## TC14  0.32  0.10  1.00
```

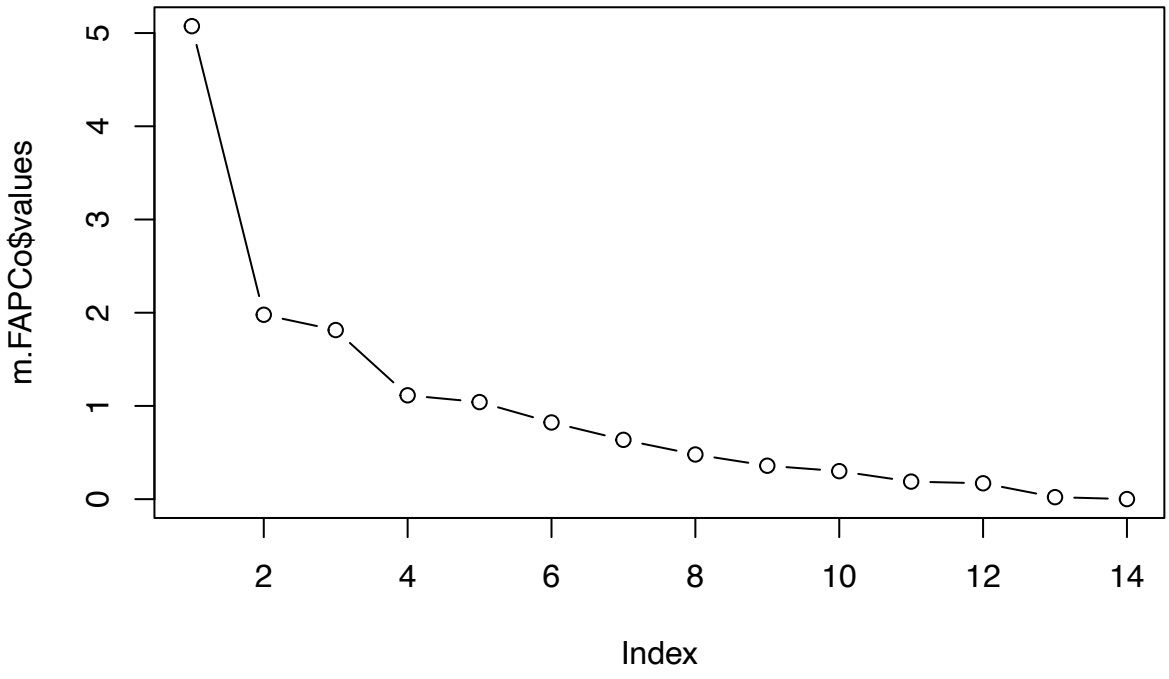
```
print.psych(m.FAPCo, cut=0.4, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = z.outl.df, nfactors = 14, rotate = "oblimin")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                 item   TC1  TC12   TC3   TC8   TC5   TC6   TC2   TC7  TC10
## installment        3  0.99
## loan_amnt          1  0.91
## total_rec_prncp   12       1.03
## total_pymnt       11       0.86
## int_rate           2             1.00
## revol_bal          8                   1.00
## annual_inc         4                         1.00
```

```
## last_pymnt_amnt   14                                1.00
## total_acc         10                                     1.00
## dti                5                                          1
## open_acc           7                                             1.00
## revol_util         9
## inq_last_6mths     6
## total_rec_int     13
##                   TC9   TC4  TC11  TC13  TC14 h2       u2 com
## installment                                   1 -2.2e-15 1.0
## loan_amnt                                     1  6.7e-16 1.1
## total_rec_prncp                               1 -2.0e-15 1.0
## total_pymnt                                   1 -2.2e-15 1.1
## int_rate                                      1 -1.3e-15 1.0
## revol_bal                                     1 -2.2e-16 1.0
## annual_inc                                    1  1.3e-15 1.0
## last_pymnt_amnt                               1  1.6e-15 1.0
## total_acc                                     1  0.0e+00 1.0
## dti                                           1 -8.9e-16 1.0
## open_acc                                      1 -4.4e-16 1.0
## revol_util       1.00                         1  8.9e-16 1.0
## inq_last_6mths         1.00                   1 -2.9e-15 1.0
## total_rec_int                0.99             1 -2.4e-15 1.0
##
##                        TC1 TC12  TC3  TC8  TC5  TC6  TC2  TC7 TC10  TC9  TC4
## SS loadings           1.93 1.90 1.02 1.01 1.01 1.01 1.00 1.00 1.00 1.01 1.00
## Proportion Var        0.14 0.14 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07
## Cumulative Var        0.14 0.27 0.35 0.42 0.49 0.56 0.63 0.71 0.78 0.85 0.92
## Proportion Explained  0.14 0.14 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.07
## Cumulative Proportion 0.14 0.27 0.35 0.42 0.49 0.56 0.63 0.71 0.78 0.85 0.92
##                       TC11 TC13 TC14
## SS loadings           1.06 0.04    0
## Proportion Var        0.08 0.00    0
## Cumulative Var        1.00 1.00    1
## Proportion Explained  0.08 0.00    0
## Cumulative Proportion 1.00 1.00    1
##
##  With component correlations of
##        TC1  TC12   TC3   TC8   TC5   TC6   TC2   TC7  TC10   TC9   TC4 TC11
## TC1   1.00  0.86  0.29  0.36  0.39  0.29  0.15  0.05  0.07  0.24  0.02 0.73
## TC12  0.86  1.00  0.13  0.32  0.38  0.47  0.15  0.00  0.02  0.17 -0.05 0.61
## TC3   0.29  0.13  1.00  0.07 -0.13  0.12 -0.09  0.21  0.04  0.48  0.22 0.56
## TC8   0.36  0.32  0.07  1.00  0.36  0.08  0.29  0.23  0.31  0.33  0.01 0.31
## TC5   0.39  0.38 -0.13  0.36  1.00  0.12  0.37 -0.24  0.19  0.05  0.08 0.22
## TC6   0.29  0.47  0.12  0.08  0.12  1.00  0.14 -0.04  0.08  0.00  0.01 0.09
## TC2   0.15  0.15 -0.09  0.29  0.37  0.14  1.00  0.17  0.63 -0.08  0.17 0.09
## TC7   0.05  0.00  0.21  0.23 -0.24 -0.04  0.17  1.00  0.28  0.24  0.06 0.16
## TC10  0.07  0.02  0.04  0.31  0.19  0.08  0.63  0.28  1.00 -0.13  0.17 0.07
## TC9   0.24  0.17  0.48  0.33  0.05  0.00 -0.08  0.24 -0.13  1.00  0.00 0.32
## TC4   0.02 -0.05  0.22  0.01  0.08  0.01  0.17  0.06  0.17  0.00  1.00 0.09
## TC11  0.73  0.61  0.56  0.31  0.22  0.09  0.09  0.16  0.07  0.32  0.09 1.00
## TC13  0.10 -0.01  0.01  0.10  0.08  0.13  0.03  0.01  0.03 -0.03 -0.04 0.25
## TC14 -0.11 -0.22  0.03 -0.07 -0.09 -0.14 -0.03  0.01  0.00 -0.05  0.01 0.32
##       TC13  TC14
## TC1   0.10 -0.11
```

```
## TC12 -0.01 -0.22
## TC3   0.01  0.03
## TC8   0.10 -0.07
## TC5   0.08 -0.09
## TC6   0.13 -0.14
## TC2   0.03 -0.03
## TC7   0.01  0.01
## TC10  0.03  0.00
## TC9  -0.03 -0.05
## TC4  -0.04  0.01
## TC11  0.25  0.32
## TC13  1.00  0.10
## TC14  0.10  1.00
##
## Mean item complexity =  1
## Test of the hypothesis that 14 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

```
plot(m.FAPCo$values,type="b")
```



```r
# Try FA on data without removing outlier --- FA on PC - Orthogonal Rotation -----> 3rd model
m.FAPC2 <-principal(z.df, 14, rotate="quartimax")
```

```
## Warning in GPForth(A, Tmat = Tmat, normalize = normalize, eps = eps, maxit =
## maxit, : convergence not obtained in GPForth. 1000 iterations used.
```

```r
print(m.FAPC2)
```

```
## Principal Components Analysis
## Call: principal(r = z.df, nfactors = 14, rotate = "quartimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                   RC1   RC7   RC5  RC10   RC4   RC6   RC3   RC2   RC8   RC9
## loan_amnt        0.95  0.09  0.02  0.04  0.00  0.03  0.03  0.03  0.08  0.09
## int_rate         0.25  0.92  0.10  0.02  0.12  0.04  0.24 -0.06 -0.02 -0.08
## installment      0.96  0.08  0.00  0.02  0.02  0.00  0.05  0.02  0.06  0.07
## annual_inc       0.36 -0.09 -0.17  0.10  0.03 -0.01  0.02  0.17  0.18  0.87
## dti              0.02  0.09  0.97  0.12  0.02 -0.04  0.11  0.07  0.08 -0.12
## inq_last_6mths   0.00  0.09  0.02  0.07  0.99  0.00 -0.02  0.06 -0.02  0.02
## open_acc         0.05  0.02  0.14  0.92  0.08  0.01 -0.09  0.31  0.12  0.08
## revol_bal        0.29 -0.02  0.10  0.13 -0.03  0.03  0.13  0.09  0.91  0.16
## revol_util       0.19  0.24  0.12 -0.09 -0.02 -0.02  0.93 -0.04  0.13  0.02
## total_acc        0.15 -0.06  0.08  0.33  0.08  0.03 -0.04  0.91  0.09  0.15
## total_pymnt      0.97  0.03  0.00 -0.01 -0.01  0.12  0.03  0.03  0.04  0.03
## total_rec_prncp  0.92 -0.13 -0.03 -0.03 -0.02  0.23  0.01  0.02  0.04  0.03
## total_rec_int    0.72  0.39  0.08  0.03  0.02 -0.18  0.08  0.03  0.03  0.01
## last_pymnt_amnt  0.34  0.03 -0.04  0.01  0.00  0.94 -0.02  0.03  0.03 -0.01
##                  RC11  RC12  RC13  RC14 h2       u2 com
## loan_amnt        0.04 -0.24  0.07  0.01  1  5.6e-16 1.2
## int_rate         0.03  0.00  0.00  0.00  1  2.4e-15 1.4
## installment     -0.10 -0.16 -0.13  0.01  1 -2.0e-15 1.2
## annual_inc       0.00  0.00  0.00  0.00  1  1.7e-15 1.7
## dti              0.01  0.00  0.00  0.00  1 -1.8e-15 1.1
## inq_last_6mths   0.00  0.00  0.00  0.00  1  3.3e-16 1.0
## open_acc         0.01  0.00  0.00  0.00  1 -3.3e-15 1.4
## revol_bal        0.00  0.00  0.00  0.00  1 -2.2e-16 1.4
## revol_util       0.01  0.00  0.00  0.00  1  2.3e-15 1.3
## total_acc        0.01  0.00  0.00  0.00  1 -6.7e-16 1.5
## total_pymnt      0.07  0.17  0.03 -0.03  1 -2.2e-15 1.1
## total_rec_prncp -0.13  0.26  0.03  0.02  1 -1.3e-15 1.4
## total_rec_int    0.53 -0.01  0.00  0.00  1 -8.9e-16 2.7
## last_pymnt_amnt -0.03  0.00  0.00  0.00  1 -6.7e-16 1.3
##
##                        RC1  RC7  RC5 RC10  RC4  RC6  RC3  RC2  RC8  RC9 RC11
## SS loadings           4.59 1.12 1.03 1.01 1.01 0.98 0.98 0.97 0.93 0.85 0.32
## Proportion Var        0.33 0.08 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.06 0.02
## Cumulative Var        0.33 0.41 0.48 0.55 0.63 0.70 0.77 0.84 0.90 0.96 0.99
## Proportion Explained  0.33 0.08 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.06 0.02
## Cumulative Proportion 0.33 0.41 0.48 0.55 0.63 0.70 0.77 0.84 0.90 0.96 0.99
##                       RC12 RC13 RC14
## SS loadings           0.18 0.02    0
## Proportion Var        0.01 0.00    0
## Cumulative Var        1.00 1.00    1
```

```
## Proportion Explained  0.01 0.00    0
## Cumulative Proportion 1.00 1.00    1
##
## Mean item complexity =  1.4
## Test of the hypothesis that 14 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

```
fscore_FAPC2 <- m.FAPC2$scores
fscorematrix <- cor(fscore_FAPC2)
lowerCor(fscore_FAPC2)
```

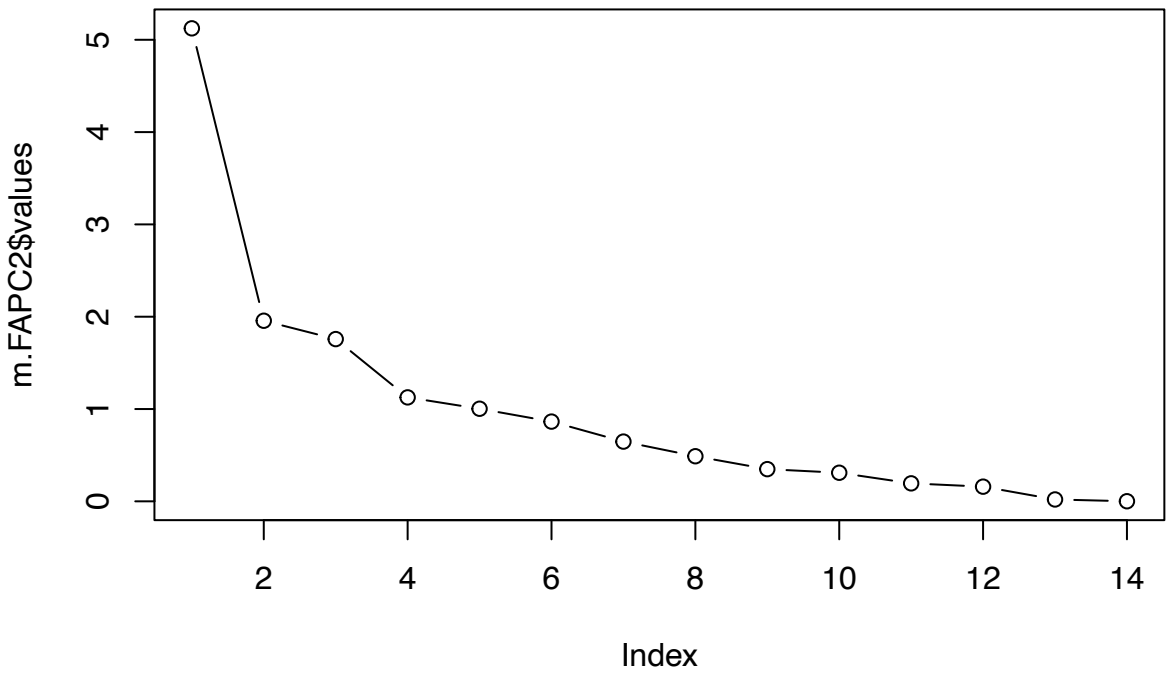
```
##      RC1 RC7 RC5 RC10 RC4 RC6 RC3 RC2 RC8 RC9 RC11
## RC1  1
## RC7  0   1
## RC5  0   0   1
## RC10 0   0   0   1
## RC4  0   0   0   0    1
## RC6  0   0   0   0    0   1
## RC3  0   0   0   0    0   0   1
## RC2  0   0   0   0    0   0   0   1
## RC8  0   0   0   0    0   0   0   0   1
## RC9  0   0   0   0    0   0   0   0   0   1
## RC11 0   0   0   0    0   0   0   0   0   0   1
## RC12 0   0   0   0    0   0   0   0   0   0   0
## RC13 0   0   0   0    0   0   0   0   0   0   0
## RC14 0   0   0   0    0   0   0   0   0   0   0
##      RC12 RC13 RC14
## RC12 1
## RC13 0    1
## RC14 0    0    1
```

```
print.psych(m.FAPC2, cut=0.4, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = z.df, nfactors = 14, rotate = "quartimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  item  RC1   RC7   RC5  RC10   RC4   RC6   RC3   RC2   RC8   RC9
## total_pymnt        11 0.97
## installment         3 0.96
## loan_amnt           1 0.95
## total_rec_prncp    12 0.92
## total_rec_int      13 0.72
## int_rate            2       0.92
## dti                 5             0.97
## open_acc            7                   0.92
## inq_last_6mths      6                         0.99
## last_pymnt_amnt    14                               0.94
## revol_util          9                                     0.93
## total_acc          10                                           0.91
```

```
## revol_bal          8                                                    0.91
## annual_inc         4                                                          0.87
##                  RC11 RC12 RC13 RC14 h2       u2 com
## total_pymnt                          1 -2.2e-15 1.1
## installment                          1 -2.0e-15 1.2
## loan_amnt                            1  5.6e-16 1.2
## total_rec_prncp                      1 -1.3e-15 1.4
## total_rec_int    0.53                1 -8.9e-16 2.7
## int_rate                             1  2.4e-15 1.4
## dti                                  1 -1.8e-15 1.1
## open_acc                             1 -3.3e-15 1.4
## inq_last_6mths                       1  3.3e-16 1.0
## last_pymnt_amnt                      1 -6.7e-16 1.3
## revol_util                           1  2.3e-15 1.3
## total_acc                            1 -6.7e-16 1.5
## revol_bal                            1 -2.2e-16 1.4
## annual_inc                           1  1.7e-15 1.7
##
##                         RC1  RC7  RC5 RC10  RC4  RC6  RC3  RC2  RC8  RC9 RC11
## SS loadings            4.59 1.12 1.03 1.01 1.01 0.98 0.98 0.97 0.93 0.85 0.32
## Proportion Var         0.33 0.08 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.06 0.02
## Cumulative Var         0.33 0.41 0.48 0.55 0.63 0.70 0.77 0.84 0.90 0.96 0.99
## Proportion Explained   0.33 0.08 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.06 0.02
## Cumulative Proportion  0.33 0.41 0.48 0.55 0.63 0.70 0.77 0.84 0.90 0.96 0.99
##                        RC12 RC13 RC14
## SS loadings            0.18 0.02    0
## Proportion Var         0.01 0.00    0
## Cumulative Var         1.00 1.00    1
## Proportion Explained   0.01 0.00    0
## Cumulative Proportion  1.00 1.00    1
##
## Mean item complexity =  1.4
## Test of the hypothesis that 14 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

```
plot(m.FAPC2$values,type="b")
```




```
# Try FA on data which removing outlier --- FA on PC - Orthogonal Rotation
m.FAPC2o <-principal(z.outl.df, 14, rotate="quartimax")
```

```
## Warning in GPForth(A, Tmat = Tmat, normalize = normalize, eps = eps, maxit =
## maxit, : convergence not obtained in GPForth. 1000 iterations used.
```

```
print(m.FAPC2o)
```

```
## Principal Components Analysis
## Call: principal(r = z.outl.df, nfactors = 14, rotate = "quartimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                    RC1   RC9   RC7   RC4   RC2   RC6   RC3  RC10   RC8   RC5
## loan_amnt         0.95  0.07  0.02  0.00  0.04  0.03  0.03  0.03  0.08  0.08
## int_rate          0.22  0.92  0.09  0.13  0.03  0.04  0.23 -0.06 -0.02 -0.10
## installment       0.97  0.07  0.01  0.02  0.02 -0.02  0.06  0.01  0.03  0.05
## annual_inc        0.34 -0.11 -0.17  0.05  0.08  0.00  0.02  0.17  0.14  0.89
## dti               0.03  0.08  0.97  0.02  0.13 -0.03  0.11  0.07  0.09 -0.13
## inq_last_6mths   -0.01  0.11  0.02  0.99  0.07  0.00 -0.01  0.07  0.00  0.04
## open_acc          0.04  0.03  0.14  0.08  0.92  0.03 -0.09  0.31  0.14  0.07
## revol_bal         0.31 -0.02  0.11 -0.01  0.15 -0.01  0.16  0.10  0.90  0.13
## revol_util        0.19  0.24  0.12 -0.01 -0.09 -0.03  0.93 -0.04  0.15  0.01
## total_acc         0.13 -0.07  0.08  0.09  0.33  0.06 -0.04  0.90  0.10  0.15
## total_pymnt       0.97  0.02  0.00 -0.02 -0.01  0.13  0.03  0.03  0.05  0.05
## total_rec_prncp   0.92 -0.13 -0.03 -0.05 -0.03  0.22  0.00  0.03  0.03  0.05
## total_rec_int     0.75  0.38  0.08  0.05  0.02 -0.13  0.07  0.01  0.07  0.01
```

```
## last_pymnt_amnt  0.31  0.03 -0.03  0.00  0.03  0.95 -0.03  0.05 -0.01  0.00
##                  RC11  RC12  RC13  RC14 h2       u2 com
## loan_amnt        0.03 -0.25  0.07  0.01  1  6.7e-16 1.2
## int_rate         0.02  0.00  0.00  0.00  1 -1.3e-15 1.4
## installment     -0.11 -0.15 -0.13  0.01  1 -2.2e-15 1.1
## annual_inc       0.00  0.00  0.00  0.00  1  1.3e-15 1.6
## dti              0.01  0.00  0.00  0.00  1 -8.9e-16 1.1
## inq_last_6mths   0.01  0.00  0.00  0.00  1 -2.9e-15 1.0
## open_acc         0.00  0.00  0.00  0.00  1 -4.4e-16 1.4
## revol_bal        0.01  0.00  0.00  0.00  1 -2.2e-16 1.5
## revol_util       0.01  0.00  0.00  0.00  1  8.9e-16 1.3
## total_acc        0.00  0.00  0.00  0.00  1  0.0e+00 1.5
## total_pymnt      0.05  0.18  0.03 -0.03  1 -2.2e-15 1.1
## total_rec_prncp -0.12  0.27  0.04  0.02  1 -2.0e-15 1.4
## total_rec_int    0.50 -0.02  0.00  0.00  1 -2.4e-15 2.5
## last_pymnt_amnt -0.02  0.00  0.00  0.00  1  1.6e-15 1.2
##
##                        RC1  RC9  RC7  RC4  RC2  RC6  RC3 RC10  RC8  RC5 RC11
## SS loadings           4.59 1.12 1.03 1.02 1.01 0.99 0.97 0.97 0.91 0.87 0.28
## Proportion Var        0.33 0.08 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.06 0.02
## Cumulative Var        0.33 0.41 0.48 0.55 0.63 0.70 0.77 0.84 0.90 0.96 0.98
## Proportion Explained  0.33 0.08 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.06 0.02
## Cumulative Proportion 0.33 0.41 0.48 0.55 0.63 0.70 0.77 0.84 0.90 0.96 0.98
##                       RC12 RC13 RC14
## SS loadings           0.19 0.03    0
## Proportion Var        0.01 0.00    0
## Cumulative Var        1.00 1.00    1
## Proportion Explained  0.01 0.00    0
## Cumulative Proportion 1.00 1.00    1
##
## Mean item complexity =  1.4
## Test of the hypothesis that 14 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

```
fscore_FAPC2o <- m.FAPC2o$scores
fscorematrix <- cor(fscore_FAPC2o)
lowerCor(fscore_FAPC2o)
```

```
##      RC1 RC9 RC7 RC4 RC2 RC6 RC3 RC10 RC8 RC5 RC11
## RC1  1
## RC9  0   1
## RC7  0   0   1
## RC4  0   0   0   1
## RC2  0   0   0   0   1
## RC6  0   0   0   0   0   1
## RC3  0   0   0   0   0   0   1
## RC10 0   0   0   0   0   0   0   1
## RC8  0   0   0   0   0   0   0   0    1
## RC5  0   0   0   0   0   0   0   0    0   1
## RC11 0   0   0   0   0   0   0   0    0   0   1
```

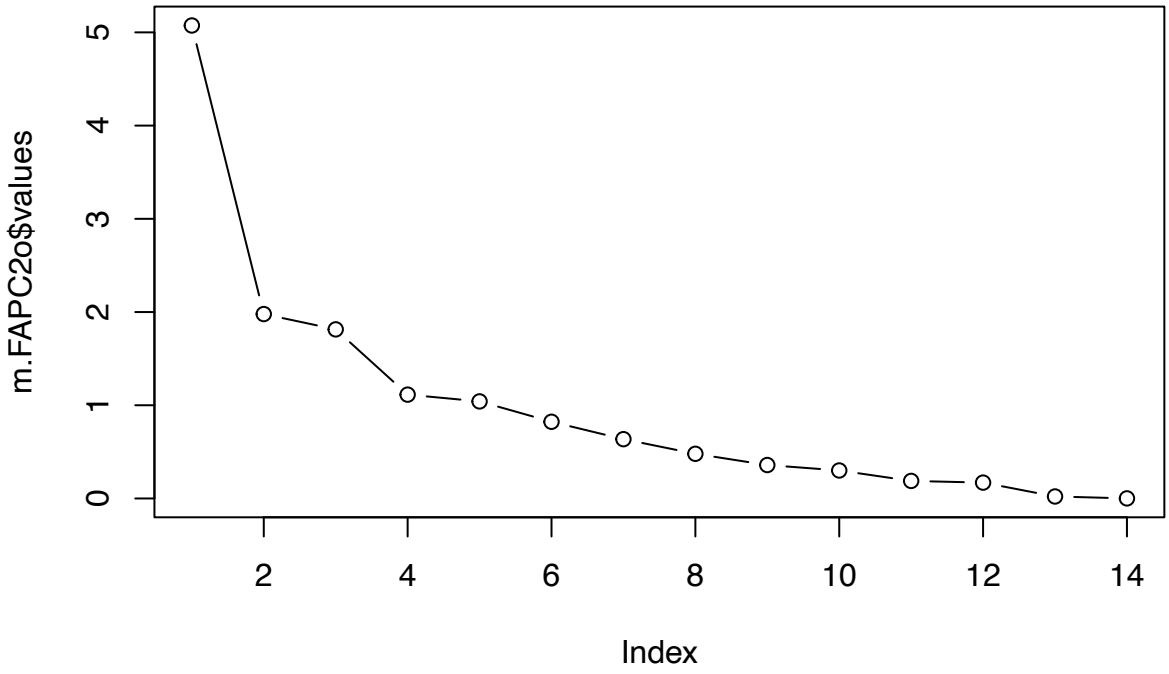
```
## RC12 0   0   0   0   0   0   0   0    0   0   0
## RC13 0   0   0   0   0   0   0   0    0   0   0
## RC14 0   0   0   0   0   0   0   0    0   0   0
##      RC12 RC13 RC14
## RC12 1
## RC13 0    1
## RC14 0    0    1
```

```
print.psych(m.FAPC2o, cut=0.4, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = z.outl.df, nfactors = 14, rotate = "quartimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  item   RC1   RC9   RC7   RC4   RC2   RC6   RC3  RC10   RC8
## total_pymnt        11  0.97
## installment         3  0.97
## loan_amnt           1  0.95
## total_rec_prncp    12  0.92
## total_rec_int      13  0.75
## int_rate            2        0.92
## dti                 5              0.97
## inq_last_6mths      6                    0.99
## open_acc            7                          0.92
## last_pymnt_amnt    14                                0.95
## revol_util          9                                      0.93
## total_acc          10                                            0.90
## revol_bal           8                                                  0.90
## annual_inc          4
##                   RC5  RC11  RC12  RC13  RC14 h2       u2 com
## total_pymnt                                    1 -2.2e-15 1.1
## installment                                    1 -2.2e-15 1.1
## loan_amnt                                      1  6.7e-16 1.2
## total_rec_prncp                                1 -2.0e-15 1.4
## total_rec_int          0.50                    1 -2.4e-15 2.5
## int_rate                                       1 -1.3e-15 1.4
## dti                                            1 -8.9e-16 1.1
## inq_last_6mths                                 1 -2.9e-15 1.0
## open_acc                                       1 -4.4e-16 1.4
## last_pymnt_amnt                                1  1.6e-15 1.2
## revol_util                                     1  8.9e-16 1.3
## total_acc                                      1  0.0e+00 1.5
## revol_bal                                      1 -2.2e-16 1.5
## annual_inc        0.89                         1  1.3e-15 1.6
##
##                         RC1  RC9  RC7  RC4  RC2  RC6  RC3 RC10  RC8  RC5 RC11
## SS loadings            4.59 1.12 1.03 1.02 1.01 0.99 0.97 0.97 0.91 0.87 0.28
## Proportion Var         0.33 0.08 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.06 0.02
## Cumulative Var         0.33 0.41 0.48 0.55 0.63 0.70 0.77 0.84 0.90 0.96 0.98
## Proportion Explained   0.33 0.08 0.07 0.07 0.07 0.07 0.07 0.07 0.07 0.06 0.02
## Cumulative Proportion  0.33 0.41 0.48 0.55 0.63 0.70 0.77 0.84 0.90 0.96 0.98
##                        RC12 RC13 RC14
## SS loadings            0.19 0.03    0
## Proportion Var         0.01 0.00    0
## Cumulative Var         1.00 1.00    1
```

```
## Proportion Explained  0.01 0.00    0
## Cumulative Proportion 1.00 1.00    1
##
## Mean item complexity =  1.4
## Test of the hypothesis that 14 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##  with the empirical chi square  0  with prob <  NA
##
## Fit based upon off diagonal values = 1
```

```
plot(m.FAPC2o$values,type="b")
```



FA Maximal Likelihood

```
# Try FA on data without removing outlier --- FA on ML - no rotate
m.FAML1 <- fa(z.df, 14, n.obs=500, rotate="none", fm="ml")
print(m.FAML1)
```

```
## Factor Analysis using method =  ml
## Call: fa(r = z.df, nfactors = 14, n.obs = 500, rotate = "none", fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                   ML1   ML4   ML2   ML5   ML6   ML3   ML7   ML8 ML9 ML10 ML11
```

```
## loan_amnt        0.90 -0.02  0.22 -0.02 -0.03  0.32  0.02 -0.04   0    0    0
## int_rate         0.25 -0.21  0.58  0.48 -0.07 -0.05  0.08  0.12   0    0    0
## installment      0.91 -0.07  0.12  0.01  0.00  0.33 -0.04  0.03   0    0    0
## annual_inc       0.39  0.35 -0.03 -0.18  0.23  0.29  0.31  0.12   0    0    0
## dti              0.02  0.17  0.22  0.39  0.07 -0.01 -0.38 -0.17   0    0    0
## inq_last_6mths  -0.01  0.14  0.11  0.17 -0.19  0.05  0.03  0.30   0    0    0
## open_acc         0.06  0.69  0.10  0.18 -0.12  0.19 -0.09  0.02   0    0    0
## revol_bal        0.34  0.31  0.05  0.17  0.37  0.21  0.10 -0.15   0    0    0
## revol_util       0.22 -0.21  0.26  0.43  0.40  0.01  0.05  0.02   0    0    0
## total_acc        0.18  0.71  0.02  0.09 -0.04  0.17 -0.02  0.06   0    0    0
## total_pymnt      1.00  0.00  0.05  0.00  0.00 -0.03  0.00  0.00   0    0    0
## total_rec_prncp  0.97  0.00 -0.25  0.00  0.00 -0.03  0.00  0.00   0    0    0
## total_rec_int    0.69  0.01  0.71 -0.02  0.00 -0.09  0.00  0.00   0    0    0
## last_pymnt_amnt  0.47  0.01 -0.34  0.30 -0.27 -0.05  0.29 -0.17   0    0    0
##                 ML12 ML13 ML14   h2     u2 com
## loan_amnt          0    0    0 0.97 0.0299 1.4
## int_rate           0    0    0 0.71 0.2942 2.9
## installment        0    0    0 0.97 0.0325 1.3
## annual_inc         0    0    0 0.56 0.4434 5.2
## dti                0    0    0 0.41 0.5928 3.4
## inq_last_6mths     0    0    0 0.19 0.8108 3.3
## open_acc           0    0    0 0.59 0.4140 1.5
## revol_bal          0    0    0 0.46 0.5445 4.7
## revol_util         0    0    0 0.51 0.4913 3.8
## total_acc          0    0    0 0.57 0.4262 1.3
## total_pymnt        0    0    0 1.00 0.0028 1.0
## total_rec_prncp    0    0    0 1.00 0.0038 1.1
## total_rec_int      0    0    0 0.99 0.0139 2.0
## last_pymnt_amnt    0    0    0 0.61 0.3874 4.6
##
##                        ML1  ML4  ML2  ML5  ML6  ML3  ML7  ML8  ML9 ML10 ML11
## SS loadings           4.69 1.34 1.21 0.80 0.48 0.43 0.36 0.21 0.00 0.00 0.00
## Proportion Var        0.34 0.10 0.09 0.06 0.03 0.03 0.03 0.01 0.00 0.00 0.00
## Cumulative Var        0.34 0.43 0.52 0.57 0.61 0.64 0.66 0.68 0.68 0.68 0.68
## Proportion Explained  0.49 0.14 0.13 0.08 0.05 0.04 0.04 0.02 0.00 0.00 0.00
## Cumulative Proportion 0.49 0.63 0.76 0.85 0.90 0.94 0.98 1.00 1.00 1.00 1.00
##                       ML12 ML13 ML14
## SS loadings           0.00 0.00 0.00
## Proportion Var        0.00 0.00 0.00
## Cumulative Var        0.68 0.68 0.68
## Proportion Explained  0.00 0.00 0.00
## Cumulative Proportion 1.00 1.00 1.00
##
## Mean item complexity =  2.7
## Test of the hypothesis that 14 factors are sufficient.
##
## df null model =  91  with the objective function =  14.8 with Chi Square =  7304.68
## df of  the model are -14  and the objective function was  0.96
##
## The root mean square of the residuals (RMSR) is  0.02
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  500 with the empirical chi square  41.9  with prob <  NA
## The total n.obs was  500  with Likelihood Chi Square =  462.93  with prob <  NA
```

```
##
## Tucker Lewis Index of factoring reliability =  1.438
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                  ML1  ML4  ML2  ML5   ML6  ML3
## Correlation of (regression) scores with factors    1 0.85 0.99 0.79  0.68 0.93
## Multiple R square of scores with factors           1 0.73 0.98 0.62  0.46 0.87
## Minimum correlation of possible factor scores      1 0.46 0.96 0.25 -0.08 0.74
##                                                   ML7   ML8 ML9 ML10 ML11
## Correlation of (regression) scores with factors  0.64  0.53   0    0    0
## Multiple R square of scores with factors         0.41  0.28   0    0    0
## Minimum correlation of possible factor scores   -0.19 -0.44  -1   -1   -1
##                                                  ML12 ML13 ML14
## Correlation of (regression) scores with factors     0    0    0
## Multiple R square of scores with factors            0    0    0
## Minimum correlation of possible factor scores      -1   -1   -1
```

```
fscore_FAML1 <- m.FAML1$scores
fscorematrix <- cor(fscore_FAML1)
lowerCor(fscore_FAML1)
```

```
##      ML1  ML4  ML2  ML5  ML6  ML3  ML7  ML8  ML9  ML10 ML11
## ML1  1.00
## ML4  0.00 1.00
## ML2  0.00 0.00 1.00
## ML5  0.00 0.00 0.00 1.00
## ML6  0.00 0.00 0.00 0.00 1.00
## ML3  0.00 0.00 0.00 0.00 0.00 1.00
## ML7  0.00 0.00 0.00 0.00 0.00 0.00 1.00
## ML8  0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00
## ML9  0.28 0.24 0.08 0.36 0.06 0.02 0.09 0.09 1.00
## ML10 0.28 0.24 0.08 0.36 0.06 0.02 0.09 0.09 1.00 1.00
## ML11 0.28 0.24 0.08 0.36 0.06 0.02 0.09 0.09 1.00 1.00 1.00
## ML12 0.28 0.24 0.08 0.36 0.06 0.02 0.09 0.09 1.00 1.00 1.00
## ML13 0.28 0.24 0.08 0.36 0.06 0.02 0.09 0.09 1.00 1.00 1.00
## ML14 0.28 0.24 0.08 0.36 0.06 0.02 0.09 0.09 1.00 1.00 1.00
##      ML12 ML13 ML14
## ML12 1.00
## ML13 1.00 1.00
## ML14 1.00 1.00 1.00
```

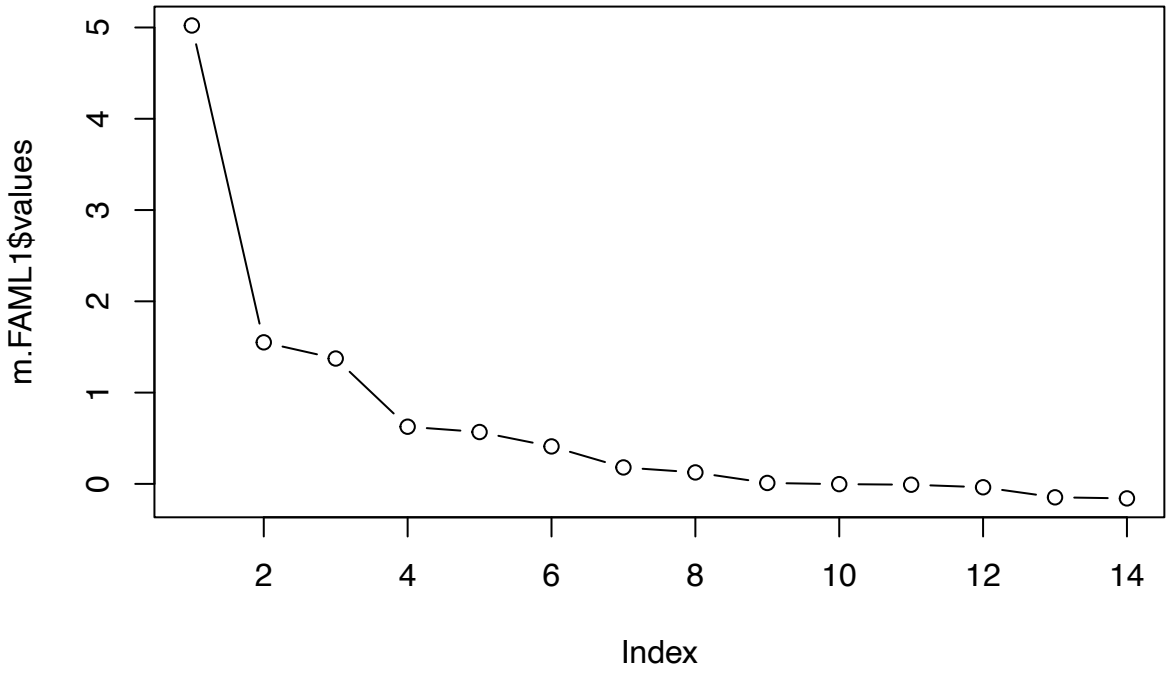
```
print.psych(m.FAML1, cut=0.4, sort=TRUE)
```

```
## Factor Analysis using method =  ml
## Call: fa(r = z.df, nfactors = 14, n.obs = 500, rotate = "none", fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  item   ML1   ML4   ML2   ML5   ML6   ML3   ML7   ML8   ML9
## total_pymnt        11  1.00
## total_rec_prncp    12  0.97
## installment         3  0.91
## loan_amnt           1  0.90
## last_pymnt_amnt    14  0.47
## annual_inc          4
```

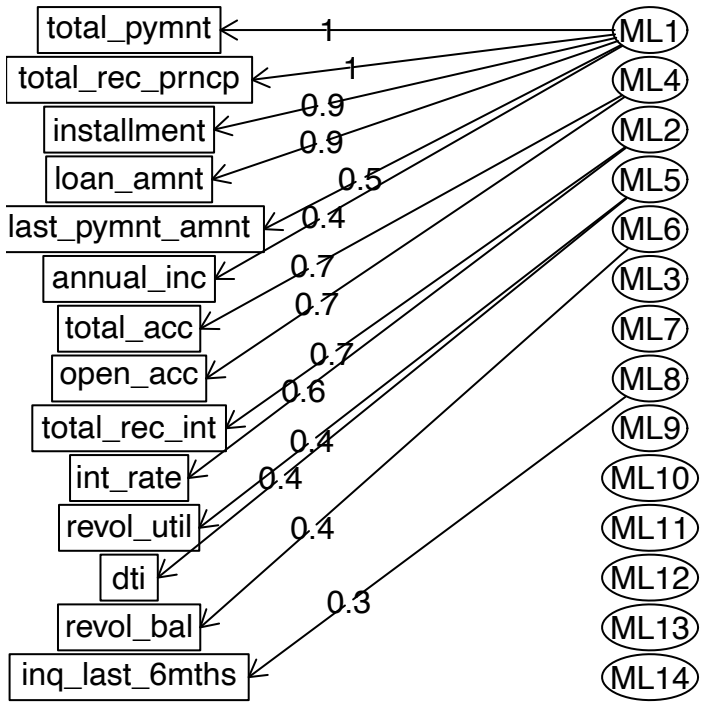
```
## total_acc         10        0.71
## open_acc           7        0.69
## total_rec_int     13  0.69        0.71
## int_rate           2               0.58  0.48
## revol_util         9                     0.43
## dti                5
## revol_bal          8
## inq_last_6mths     6
##                   ML10  ML11  ML12  ML13  ML14   h2     u2 com
## total_pymnt                                     1.00 0.0028 1.0
## total_rec_prncp                                 1.00 0.0038 1.1
## installment                                     0.97 0.0325 1.3
## loan_amnt                                       0.97 0.0299 1.4
## last_pymnt_amnt                                 0.61 0.3874 4.6
## annual_inc                                      0.56 0.4434 5.2
## total_acc                                       0.57 0.4262 1.3
## open_acc                                        0.59 0.4140 1.5
## total_rec_int                                   0.99 0.0139 2.0
## int_rate                                        0.71 0.2942 2.9
## revol_util                                      0.51 0.4913 3.8
## dti                                             0.41 0.5928 3.4
## revol_bal                                       0.46 0.5445 4.7
## inq_last_6mths                                  0.19 0.8108 3.3
##
##                        ML1  ML4  ML2  ML5  ML6  ML3  ML7  ML8  ML9 ML10 ML11
## SS loadings           4.69 1.34 1.21 0.80 0.48 0.43 0.36 0.21 0.00 0.00 0.00
## Proportion Var        0.34 0.10 0.09 0.06 0.03 0.03 0.03 0.01 0.00 0.00 0.00
## Cumulative Var        0.34 0.43 0.52 0.57 0.61 0.64 0.66 0.68 0.68 0.68 0.68
## Proportion Explained  0.49 0.14 0.13 0.08 0.05 0.04 0.04 0.02 0.00 0.00 0.00
## Cumulative Proportion 0.49 0.63 0.76 0.85 0.90 0.94 0.98 1.00 1.00 1.00 1.00
##                       ML12 ML13 ML14
## SS loadings           0.00 0.00 0.00
## Proportion Var        0.00 0.00 0.00
## Cumulative Var        0.68 0.68 0.68
## Proportion Explained  0.00 0.00 0.00
## Cumulative Proportion 1.00 1.00 1.00
##
## Mean item complexity =  2.7
## Test of the hypothesis that 14 factors are sufficient.
##
## df null model =  91  with the objective function =  14.8 with Chi Square =  7304.68
## df of  the model are -14  and the objective function was  0.96
##
## The root mean square of the residuals (RMSR) is  0.02
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  500 with the empirical chi square  41.9  with prob <  NA
## The total n.obs was  500  with Likelihood Chi Square =  462.93  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.438
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                  ML1  ML4  ML2  ML5   ML6  ML3
## Correlation of (regression) scores with factors    1 0.85 0.99 0.79  0.68 0.93
```

```
## Multiple R square of scores with factors           1 0.73 0.98 0.62  0.46 0.87
## Minimum correlation of possible factor scores      1 0.46 0.96 0.25 -0.08 0.74
##                                                  ML7   ML8 ML9 ML10 ML11
## Correlation of (regression) scores with factors  0.64  0.53   0    0    0
## Multiple R square of scores with factors         0.41  0.28   0    0    0
## Minimum correlation of possible factor scores   -0.19 -0.44  -1   -1   -1
##                                                 ML12 ML13 ML14
## Correlation of (regression) scores with factors    0    0    0
## Multiple R square of scores with factors           0    0    0
## Minimum correlation of possible factor scores     -1   -1   -1
```

```
plot(m.FAML1$values,type="b")
```




```
fa.diagram(m.FAML1)
```




```
# Try FA on data removing outlier --- FA on ML - no rotate ----------------------------- 1st, Model
m.FAML1o <-fa(z.outl.df, 14, n.obs=480, rotate="none", fm="ml")
print(m.FAML1o)
```

```
## Factor Analysis using method =  ml
## Call: fa(r = z.outl.df, nfactors = 14, n.obs = 480, rotate = "none",
##     fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                    ML1   ML4   ML2   ML5   ML6   ML7   ML3   ML8 ML9 ML10 ML11
## loan_amnt         0.89 -0.01  0.25 -0.02 -0.03  0.02  0.32 -0.05   0    0    0
## int_rate          0.21 -0.15  0.60  0.49 -0.11  0.09 -0.11  0.11   0    0    0
## installment       0.91 -0.07  0.15  0.03  0.01 -0.03  0.33  0.04   0    0    0
## annual_inc        0.39  0.33 -0.05 -0.25  0.14  0.36  0.22  0.10   0    0    0
## dti               0.02  0.21  0.22  0.38  0.15 -0.36 -0.01 -0.13   0    0    0
## inq_last_6mths   -0.03  0.18  0.17  0.16 -0.16  0.09  0.05  0.29   0    0    0
## open_acc          0.04  0.72  0.08  0.16 -0.10 -0.12  0.16  0.03   0    0    0
## revol_bal         0.35  0.36  0.12  0.12  0.39  0.10  0.15 -0.15   0    0    0
## revol_util        0.21 -0.17  0.29  0.44  0.39  0.17  0.00 -0.01   0    0    0
## total_acc         0.16  0.72 -0.02  0.05 -0.05  0.00  0.15  0.08   0    0    0
## total_pymnt       1.00  0.00  0.04  0.00  0.00  0.00 -0.03  0.00   0    0    0
## total_rec_prncp   0.97  0.00 -0.22  0.00  0.00  0.00 -0.03  0.00   0    0    0
## total_rec_int     0.71  0.01  0.68 -0.02  0.00  0.00 -0.11  0.00   0    0    0
## last_pymnt_amnt   0.44  0.08 -0.30  0.28 -0.35  0.22 -0.08 -0.22   0    0    0
##                  ML12 ML13 ML14   h2     u2 com
## loan_amnt           0    0    0 0.97 0.0324 1.4
## int_rate            0    0    0 0.71 0.2907 2.7
```

```
## installment        0    0    0 0.97 0.0343 1.3
## annual_inc         0    0    0 0.54 0.4649 4.9
## dti                0    0    0 0.40 0.5951 3.9
## inq_last_6mths     0    0    0 0.21 0.7942 4.2
## open_acc           0    0    0 0.59 0.4053 1.3
## revol_bal          0    0    0 0.49 0.5126 4.2
## revol_util         0    0    0 0.53 0.4728 3.9
## total_acc          0    0    0 0.58 0.4205 1.2
## total_pymnt        0    0    0 1.00 0.0028 1.0
## total_rec_prncp    0    0    0 1.00 0.0038 1.1
## total_rec_int      0    0    0 0.98 0.0162 2.0
## last_pymnt_amnt    0    0    0 0.59 0.4117 5.0
##
##                         ML1  ML4  ML2  ML5  ML6  ML7  ML3  ML8  ML9 ML10 ML11
## SS loadings            4.65 1.41 1.23 0.78 0.52 0.37 0.36 0.20 0.00 0.00 0.00
## Proportion Var         0.33 0.10 0.09 0.06 0.04 0.03 0.03 0.01 0.00 0.00 0.00
## Cumulative Var         0.33 0.43 0.52 0.58 0.61 0.64 0.67 0.68 0.68 0.68 0.68
## Proportion Explained   0.49 0.15 0.13 0.08 0.05 0.04 0.04 0.02 0.00 0.00 0.00
## Cumulative Proportion  0.49 0.64 0.76 0.85 0.90 0.94 0.98 1.00 1.00 1.00 1.00
##                        ML12 ML13 ML14
## SS loadings            0.00 0.00 0.00
## Proportion Var         0.00 0.00 0.00
## Cumulative Var         0.68 0.68 0.68
## Proportion Explained   0.00 0.00 0.00
## Cumulative Proportion  1.00 1.00 1.00
##
## Mean item complexity =  2.7
## Test of the hypothesis that 14 factors are sufficient.
##
## df null model =  91  with the objective function =  14.74 with Chi Square =  6981.34
## df of  the model are -14  and the objective function was  0.99
##
## The root mean square of the residuals (RMSR) is  0.02
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  480 with the empirical chi square  40.22  with prob <  NA
## The total n.obs was  480  with Likelihood Chi Square =  457.51  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.454
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                  ML1  ML4  ML2  ML5   ML6
## Correlation of (regression) scores with factors    1 0.86 0.99 0.79  0.70
## Multiple R square of scores with factors           1 0.74 0.97 0.63  0.49
## Minimum correlation of possible factor scores      1 0.48 0.95 0.25 -0.02
##                                                   ML7  ML3   ML8 ML9 ML10
## Correlation of (regression) scores with factors  0.63 0.93  0.55   0    0
## Multiple R square of scores with factors         0.40 0.86  0.30   0    0
## Minimum correlation of possible factor scores    -0.20 0.73 -0.39  -1   -1
##                                                  ML11 ML12 ML13 ML14
## Correlation of (regression) scores with factors     0    0    0    0
## Multiple R square of scores with factors            0    0    0    0
## Minimum correlation of possible factor scores      -1   -1   -1   -1
```

```r
fscore_FAML1o <- m.FAML1o$scores
fscorematrix <- cor(fscore_FAML1o)
lowerCor(fscore_FAML1o)
```

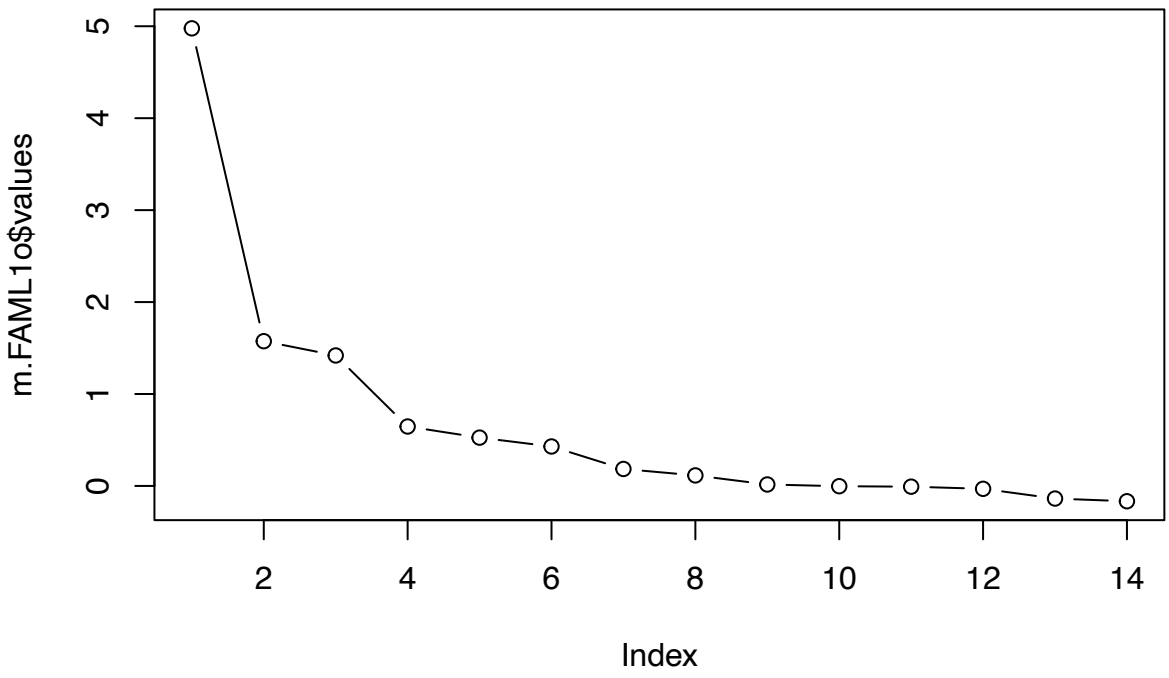
```
##      ML1  ML4  ML2  ML5  ML6  ML7  ML3  ML8  ML9  ML10 ML11
## ML1  1.00
## ML4  0.00 1.00
## ML2  0.00 0.00 1.00
## ML5  0.00 0.00 0.00 1.00
## ML6  0.00 0.00 0.00 0.00 1.00
## ML7  0.00 0.00 0.00 0.00 0.00 1.00
## ML3  0.00 0.00 0.00 0.00 0.00 0.00 1.00
## ML8  0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00
## ML9  0.30 0.30 0.09 0.36 0.04 0.14 0.02 0.08 1.00
## ML10 0.30 0.30 0.09 0.36 0.04 0.14 0.02 0.08 1.00 1.00
## ML11 0.30 0.30 0.09 0.36 0.04 0.14 0.02 0.08 1.00 1.00 1.00
## ML12 0.30 0.30 0.09 0.36 0.04 0.14 0.02 0.08 1.00 1.00 1.00
## ML13 0.30 0.30 0.09 0.36 0.04 0.14 0.02 0.08 1.00 1.00 1.00
## ML14 0.30 0.30 0.09 0.36 0.04 0.14 0.02 0.08 1.00 1.00 1.00
##      ML12 ML13 ML14
## ML12 1.00
## ML13 1.00 1.00
## ML14 1.00 1.00 1.00
```

```r
print.psych(m.FAML1o, cut=0.4, sort=TRUE)
```

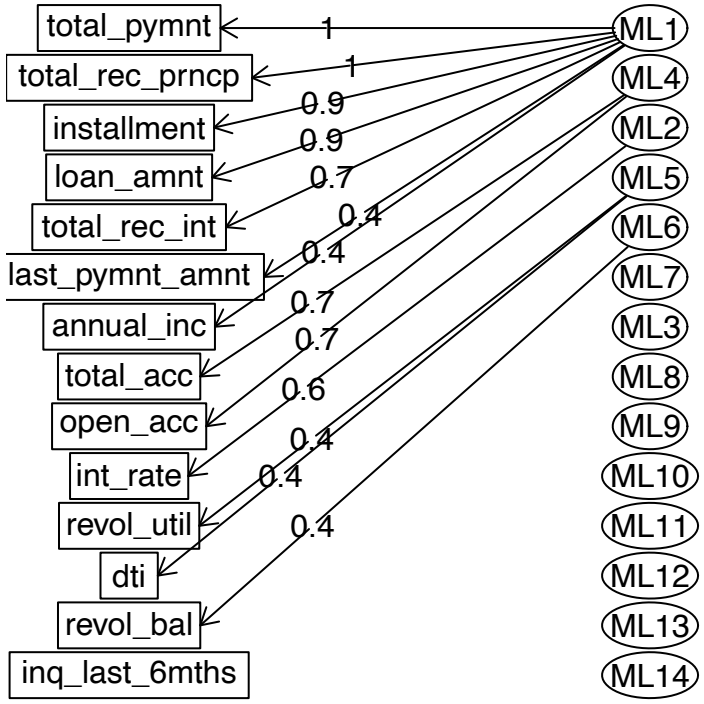
```
## Factor Analysis using method =  ml
## Call: fa(r = z.outl.df, nfactors = 14, n.obs = 480, rotate = "none",
##     fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  item   ML1   ML4   ML2   ML5   ML6   ML7   ML3   ML8   ML9
## total_pymnt        11  1.00
## total_rec_prncp    12  0.97
## installment         3  0.91
## loan_amnt           1  0.89
## total_rec_int      13  0.71        0.68
## last_pymnt_amnt    14  0.44
## annual_inc          4
## total_acc          10        0.72
## open_acc            7        0.72
## int_rate            2              0.60  0.49
## revol_util          9                    0.44
## dti                 5
## revol_bal           8
## inq_last_6mths      6
##                  ML10  ML11  ML12  ML13  ML14   h2     u2 com
## total_pymnt                                    1.00 0.0028 1.0
## total_rec_prncp                                1.00 0.0038 1.1
## installment                                    0.97 0.0343 1.3
## loan_amnt                                      0.97 0.0324 1.4
## total_rec_int                                  0.98 0.0162 2.0
## last_pymnt_amnt                                0.59 0.4117 5.0
```

```
## annual_inc                                      0.54 0.4649 4.9
## total_acc                                       0.58 0.4205 1.2
## open_acc                                        0.59 0.4053 1.3
## int_rate                                        0.71 0.2907 2.7
## revol_util                                      0.53 0.4728 3.9
## dti                                             0.40 0.5951 3.9
## revol_bal                                       0.49 0.5126 4.2
## inq_last_6mths                                  0.21 0.7942 4.2
##
##                         ML1  ML4  ML2  ML5  ML6  ML7  ML3  ML8  ML9 ML10 ML11
## SS loadings            4.65 1.41 1.23 0.78 0.52 0.37 0.36 0.20 0.00 0.00 0.00
## Proportion Var         0.33 0.10 0.09 0.06 0.04 0.03 0.03 0.01 0.00 0.00 0.00
## Cumulative Var         0.33 0.43 0.52 0.58 0.61 0.64 0.67 0.68 0.68 0.68 0.68
## Proportion Explained   0.49 0.15 0.13 0.08 0.05 0.04 0.04 0.02 0.00 0.00 0.00
## Cumulative Proportion  0.49 0.64 0.76 0.85 0.90 0.94 0.98 1.00 1.00 1.00 1.00
##                        ML12 ML13 ML14
## SS loadings            0.00 0.00 0.00
## Proportion Var         0.00 0.00 0.00
## Cumulative Var         0.68 0.68 0.68
## Proportion Explained   0.00 0.00 0.00
## Cumulative Proportion  1.00 1.00 1.00
##
## Mean item complexity =  2.7
## Test of the hypothesis that 14 factors are sufficient.
##
## df null model =  91  with the objective function =  14.74 with Chi Square =  6981.34
## df of  the model are -14  and the objective function was  0.99
##
## The root mean square of the residuals (RMSR) is  0.02
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  480 with the empirical chi square  40.22  with prob <  NA
## The total n.obs was  480  with Likelihood Chi Square =  457.51  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.454
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                  ML1  ML4  ML2  ML5   ML6
## Correlation of (regression) scores with factors    1 0.86 0.99 0.79  0.70
## Multiple R square of scores with factors           1 0.74 0.97 0.63  0.49
## Minimum correlation of possible factor scores      1 0.48 0.95 0.25 -0.02
##                                                   ML7  ML3   ML8 ML9 ML10
## Correlation of (regression) scores with factors  0.63 0.93  0.55   0    0
## Multiple R square of scores with factors         0.40 0.86  0.30   0    0
## Minimum correlation of possible factor scores   -0.20 0.73 -0.39  -1   -1
##                                                  ML11 ML12 ML13 ML14
## Correlation of (regression) scores with factors     0    0    0    0
## Multiple R square of scores with factors            0    0    0    0
## Minimum correlation of possible factor scores       -1   -1   -1   -1
```

```
plot(m.FAML1o$values,type="b")
```




```
fa.diagram(m.FAML1o)
```




```
# Try FA on data without removing outlier --- FA on ML - Oblique
m.FAML2 <- fa(z.df, 14, n.obs=500, rotate="oblimin", fm="ml")
print(m.FAML2)
```

```
## Factor Analysis using method =  ml
## Call: fa(r = z.df, nfactors = 14, n.obs = 500, rotate = "oblimin",
##     fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                    ML1   ML3   ML4   ML2   ML6   ML8   ML7   ML5 ML9 ML10 ML11
## loan_amnt         0.92 -0.05  0.00  0.11 -0.03  0.06  0.02 -0.04   0    0    0
## int_rate          0.10 -0.20 -0.02  0.37  0.32  0.16 -0.12  0.36   0    0    0
## installment       0.95  0.12 -0.01 -0.07  0.02 -0.03  0.00  0.04   0    0    0
## annual_inc        0.18  0.03  0.24  0.02  0.15 -0.03  0.53 -0.10   0    0    0
## dti               0.05  0.01  0.29  0.00  0.21 -0.07 -0.52 -0.09   0    0    0
## inq_last_6mths    0.01  0.05  0.30 -0.08  0.01 -0.05  0.06  0.43   0    0    0
## open_acc          0.05 -0.09  0.77  0.01 -0.06  0.03 -0.07  0.02   0    0    0
## revol_bal         0.11 -0.01  0.24  0.01  0.41  0.08  0.12 -0.30   0    0    0
## revol_util       -0.02  0.08 -0.14  0.02  0.69 -0.02 -0.04  0.05   0    0    0
## total_acc        -0.07  0.11  0.74  0.03 -0.02  0.00  0.08  0.00   0    0    0
## total_pymnt       0.16  0.63  0.01  0.30  0.02  0.09  0.01 -0.01   0    0    0
## total_rec_prncp   0.11  0.82  0.00  0.01  0.03  0.13  0.02 -0.01   0    0    0
## total_rec_int     0.04  0.08  0.02  0.93  0.01 -0.04  0.01  0.00   0    0    0
## last_pymnt_amnt   0.00  0.11  0.02 -0.08 -0.02  0.73  0.01 -0.01   0    0    0
##                  ML12 ML13 ML14   h2     u2 com
## loan_amnt           0    0    0 0.97 0.0299 1.1
## int_rate            0    0    0 0.71 0.2942 4.4
```

```
## installment        0    0    0 0.97 0.0325 1.1
## annual_inc         0    0    0 0.56 0.4434 1.9
## dti                0    0    0 0.41 0.5928 2.1
## inq_last_6mths     0    0    0 0.19 0.8108 2.0
## open_acc           0    0    0 0.59 0.4140 1.1
## revol_bal          0    0    0 0.46 0.5445 3.0
## revol_util         0    0    0 0.51 0.4913 1.1
## total_acc          0    0    0 0.57 0.4262 1.1
## total_pymnt        0    0    0 1.00 0.0028 1.6
## total_rec_prncp    0    0    0 1.00 0.0038 1.1
## total_rec_int      0    0    0 0.99 0.0139 1.0
## last_pymnt_amnt    0    0    0 0.61 0.3874 1.1
##
##                         ML1  ML3  ML4  ML2  ML6  ML8  ML7  ML5  ML9 ML10 ML11
## SS loadings            2.24 1.57 1.48 1.39 0.94 0.74 0.67 0.47 0.00 0.00 0.00
## Proportion Var         0.16 0.11 0.11 0.10 0.07 0.05 0.05 0.03 0.00 0.00 0.00
## Cumulative Var         0.16 0.27 0.38 0.48 0.54 0.60 0.65 0.68 0.68 0.68 0.68
## Proportion Explained   0.24 0.17 0.16 0.15 0.10 0.08 0.07 0.05 0.00 0.00 0.00
## Cumulative Proportion  0.24 0.40 0.56 0.70 0.80 0.88 0.95 1.00 1.00 1.00 1.00
##                        ML12 ML13 ML14
## SS loadings            0.00 0.00 0.00
## Proportion Var         0.00 0.00 0.00
## Cumulative Var         0.68 0.68 0.68
## Proportion Explained   0.00 0.00 0.00
## Cumulative Proportion  1.00 1.00 1.00
##
##  With factor correlations of
##        ML1   ML3   ML4   ML2   ML6  ML8   ML7   ML5 ML9 ML10 ML11 ML12 ML13
## ML1   1.00  0.78  0.20  0.68  0.36 0.44  0.23 -0.05   0    0    0    0    0
## ML3   0.78  1.00  0.08  0.38  0.10 0.56  0.25 -0.20   0    0    0    0    0
## ML4   0.20  0.08  1.00  0.11  0.10 0.08  0.12 -0.21   0    0    0    0    0
## ML2   0.68  0.38  0.11  1.00  0.43 0.15 -0.06  0.24   0    0    0    0    0
## ML6   0.36  0.10  0.10  0.43  1.00 0.15 -0.15  0.09   0    0    0    0    0
## ML8   0.44  0.56  0.08  0.15  0.15 1.00  0.06  0.09   0    0    0    0    0
## ML7   0.23  0.25  0.12 -0.06 -0.15 0.06  1.00 -0.26   0    0    0    0    0
## ML5  -0.05 -0.20 -0.21  0.24  0.09 0.09 -0.26  1.00   0    0    0    0    0
## ML9   0.00  0.00  0.00  0.00  0.00 0.00  0.00  0.00   1    0    0    0    0
## ML10  0.00  0.00  0.00  0.00  0.00 0.00  0.00  0.00   0    1    0    0    0
## ML11  0.00  0.00  0.00  0.00  0.00 0.00  0.00  0.00   0    0    1    0    0
## ML12  0.00  0.00  0.00  0.00  0.00 0.00  0.00  0.00   0    0    0    1    0
## ML13  0.00  0.00  0.00  0.00  0.00 0.00  0.00  0.00   0    0    0    0    1
## ML14  0.00  0.00  0.00  0.00  0.00 0.00  0.00  0.00   0    0    0    0    0
##      ML14
## ML1     0
## ML3     0
## ML4     0
## ML2     0
## ML6     0
## ML8     0
## ML7     0
## ML5     0
## ML9     0
## ML10    0
## ML11    0
```

```
## ML12    0
## ML13    0
## ML14    1
##
## Mean item complexity =  1.7
## Test of the hypothesis that 14 factors are sufficient.
##
## df null model =  91  with the objective function =  14.8 with Chi Square =  7304.68
## df of  the model are -14  and the objective function was  0.96
##
## The root mean square of the residuals (RMSR) is  0.02
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  500 with the empirical chi square  41.9  with prob <  NA
## The total n.obs was  500  with Likelihood Chi Square =  462.93  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.438
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                  ML1  ML3  ML4  ML2  ML6  ML8
## Correlation of (regression) scores with factors   0.99 0.99 0.86 0.99 0.80 0.84
## Multiple R square of scores with factors          0.98 0.99 0.74 0.98 0.64 0.71
## Minimum correlation of possible factor scores     0.96 0.98 0.47 0.97 0.29 0.41
##                                                  ML7  ML5 ML9 ML10 ML11 ML12
## Correlation of (regression) scores with factors   0.75 0.73   0    0    0    0
## Multiple R square of scores with factors          0.56 0.53   0    0    0    0
## Minimum correlation of possible factor scores     0.11 0.06  -1   -1   -1   -1
##                                                  ML13 ML14
## Correlation of (regression) scores with factors     0    0
## Multiple R square of scores with factors            0    0
## Minimum correlation of possible factor scores      -1   -1
```

```
fscore_FAML2 <- m.FAML2$scores
fscorematrix <- cor(fscore_FAML2)
lowerCor(fscore_FAML2)
```

```
##      ML1   ML3   ML4   ML2   ML6   ML8   ML7   ML5   ML9   ML10  ML11
## ML1   1.00
## ML3   0.80  1.00
## ML4   0.23  0.10  1.00
## ML2   0.69  0.39  0.14  1.00
## ML6   0.45  0.15  0.14  0.54  1.00
## ML8   0.53  0.71  0.10  0.17  0.20  1.00
## ML7   0.31  0.34  0.19 -0.07 -0.22  0.13  1.00
## ML5  -0.06 -0.28 -0.29  0.31  0.24  0.02 -0.50  1.00
## ML9   0.19 -0.11 -0.17  0.58  0.64  0.11 -0.60  0.87  1.00
## ML10  0.19 -0.11 -0.17  0.58  0.64  0.11 -0.60  0.87  1.00  1.00
## ML11  0.19 -0.11 -0.17  0.58  0.64  0.11 -0.60  0.87  1.00  1.00  1.00
## ML12  0.19 -0.11 -0.17  0.58  0.64  0.11 -0.60  0.87  1.00  1.00  1.00
## ML13  0.19 -0.11 -0.17  0.58  0.64  0.11 -0.60  0.87  1.00  1.00  1.00
## ML14  0.19 -0.11 -0.17  0.58  0.64  0.11 -0.60  0.87  1.00  1.00  1.00
##      ML12  ML13  ML14
## ML12  1.00
## ML13  1.00  1.00
```

```
## ML14  1.00  1.00  1.00
```

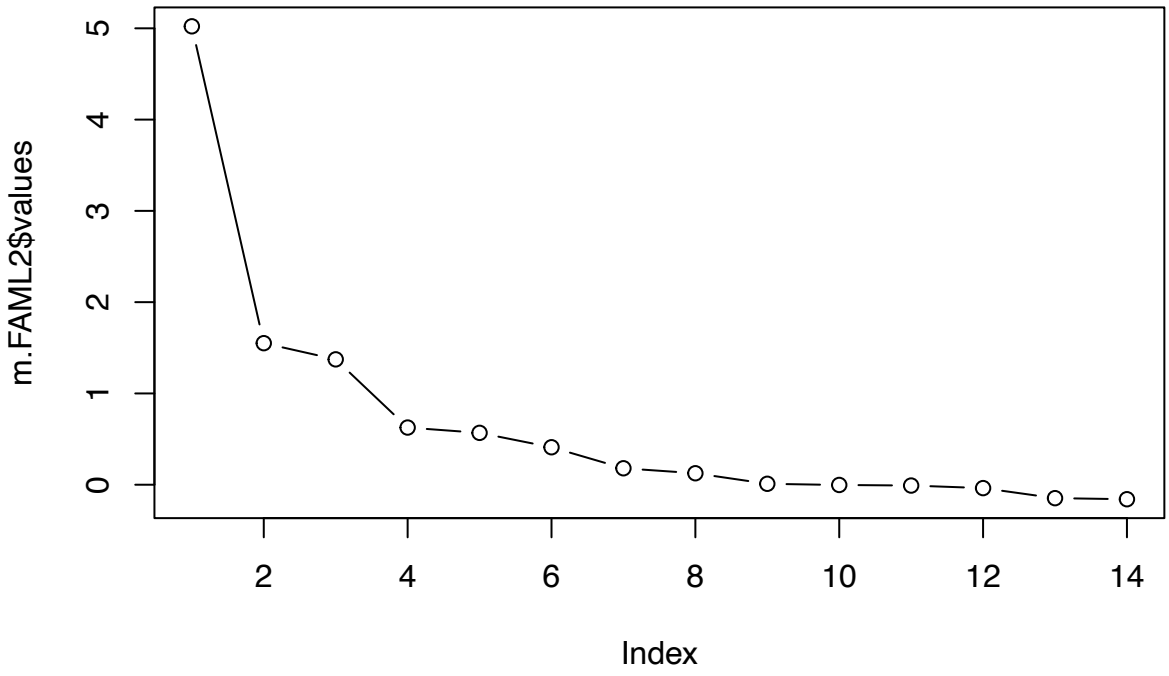
```
print.psych(m.FAML2, cut=0.4, sort=TRUE)
```

```
## Factor Analysis using method =  ml
## Call: fa(r = z.df, nfactors = 14, n.obs = 500, rotate = "oblimin",
##     fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  item   ML1   ML3   ML4   ML2   ML6   ML8   ML7   ML5   ML9
## installment         3  0.95
## loan_amnt           1  0.92
## total_rec_prncp    12        0.82
## total_pymnt        11        0.63
## open_acc            7              0.77
## total_acc          10              0.74
## total_rec_int      13                    0.93
## int_rate            2
## revol_util          9                          0.69
## revol_bal           8                          0.41
## last_pymnt_amnt    14                                0.73
## annual_inc          4                                      0.53
## dti                 5                                     -0.52
## inq_last_6mths      6                                            0.43
##                  ML10  ML11  ML12  ML13  ML14   h2     u2 com
## installment                                    0.97 0.0325 1.1
## loan_amnt                                      0.97 0.0299 1.1
## total_rec_prncp                                1.00 0.0038 1.1
## total_pymnt                                    1.00 0.0028 1.6
## open_acc                                       0.59 0.4140 1.1
## total_acc                                      0.57 0.4262 1.1
## total_rec_int                                  0.99 0.0139 1.0
## int_rate                                       0.71 0.2942 4.4
## revol_util                                     0.51 0.4913 1.1
## revol_bal                                      0.46 0.5445 3.0
## last_pymnt_amnt                                0.61 0.3874 1.1
## annual_inc                                     0.56 0.4434 1.9
## dti                                            0.41 0.5928 2.1
## inq_last_6mths                                 0.19 0.8108 2.0
##
##                        ML1  ML3  ML4  ML2  ML6  ML8  ML7  ML5  ML9 ML10 ML11
## SS loadings           2.24 1.57 1.48 1.39 0.94 0.74 0.67 0.47 0.00 0.00 0.00
## Proportion Var        0.16 0.11 0.11 0.10 0.07 0.05 0.05 0.03 0.00 0.00 0.00
## Cumulative Var        0.16 0.27 0.38 0.48 0.54 0.60 0.65 0.68 0.68 0.68 0.68
## Proportion Explained  0.24 0.17 0.16 0.15 0.10 0.08 0.07 0.05 0.00 0.00 0.00
## Cumulative Proportion 0.24 0.40 0.56 0.70 0.80 0.88 0.95 1.00 1.00 1.00 1.00
##                       ML12 ML13 ML14
## SS loadings           0.00 0.00 0.00
## Proportion Var        0.00 0.00 0.00
## Cumulative Var        0.68 0.68 0.68
## Proportion Explained  0.00 0.00 0.00
## Cumulative Proportion 1.00 1.00 1.00
##
##  With factor correlations of
##        ML1   ML3   ML4   ML2   ML6  ML8   ML7   ML5 ML9 ML10 ML11 ML12 ML13
```

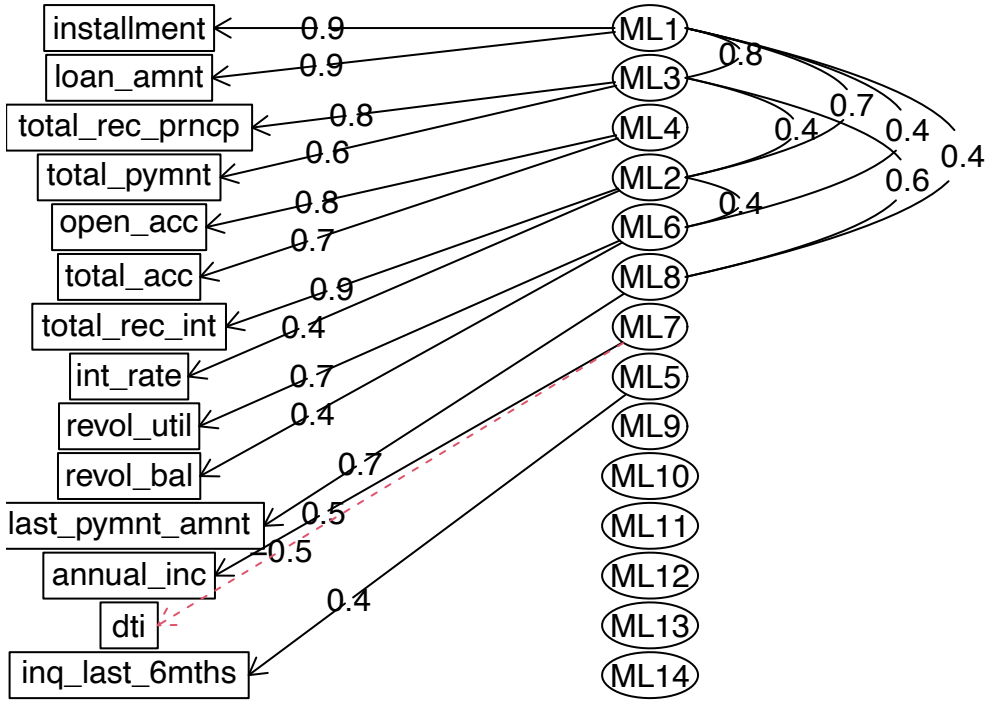
```
## ML1   1.00  0.78  0.20  0.68  0.36 0.44  0.23 -0.05   0    0    0    0    0
## ML3   0.78  1.00  0.08  0.38  0.10 0.56  0.25 -0.20   0    0    0    0    0
## ML4   0.20  0.08  1.00  0.11  0.10 0.08  0.12 -0.21   0    0    0    0    0
## ML2   0.68  0.38  0.11  1.00  0.43 0.15 -0.06  0.24   0    0    0    0    0
## ML6   0.36  0.10  0.10  0.43  1.00 0.15 -0.15  0.09   0    0    0    0    0
## ML8   0.44  0.56  0.08  0.15  0.15 1.00  0.06  0.09   0    0    0    0    0
## ML7   0.23  0.25  0.12 -0.06 -0.15 0.06  1.00 -0.26   0    0    0    0    0
## ML5  -0.05 -0.20 -0.21  0.24  0.09 0.09 -0.26  1.00   0    0    0    0    0
## ML9   0.00  0.00  0.00  0.00  0.00 0.00  0.00  0.00   1    0    0    0    0
## ML10  0.00  0.00  0.00  0.00  0.00 0.00  0.00  0.00   0    1    0    0    0
## ML11  0.00  0.00  0.00  0.00  0.00 0.00  0.00  0.00   0    0    1    0    0
## ML12  0.00  0.00  0.00  0.00  0.00 0.00  0.00  0.00   0    0    0    1    0
## ML13  0.00  0.00  0.00  0.00  0.00 0.00  0.00  0.00   0    0    0    0    1
## ML14  0.00  0.00  0.00  0.00  0.00 0.00  0.00  0.00   0    0    0    0    0
##      ML14
## ML1     0
## ML3     0
## ML4     0
## ML2     0
## ML6     0
## ML8     0
## ML7     0
## ML5     0
## ML9     0
## ML10    0
## ML11    0
## ML12    0
## ML13    0
## ML14    1
##
## Mean item complexity =  1.7
## Test of the hypothesis that 14 factors are sufficient.
##
## df null model =  91  with the objective function =  14.8 with Chi Square =  7304.68
## df of  the model are -14  and the objective function was  0.96
##
## The root mean square of the residuals (RMSR) is  0.02
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  500 with the empirical chi square  41.9  with prob <  NA
## The total n.obs was  500  with Likelihood Chi Square =  462.93  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.438
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                  ML1  ML3  ML4  ML2  ML6  ML8
## Correlation of (regression) scores with factors   0.99 0.99 0.86 0.99 0.80 0.84
## Multiple R square of scores with factors          0.98 0.99 0.74 0.98 0.64 0.71
## Minimum correlation of possible factor scores     0.96 0.98 0.47 0.97 0.29 0.41
##                                                  ML7  ML5 ML9 ML10 ML11 ML12
## Correlation of (regression) scores with factors   0.75 0.73   0    0    0    0
## Multiple R square of scores with factors          0.56 0.53   0    0    0    0
## Minimum correlation of possible factor scores     0.11 0.06  -1   -1   -1   -1
##                                                  ML13 ML14
```

```
## Correlation of (regression) scores with factors      0    0
## Multiple R square of scores with factors             0    0
## Minimum correlation of possible factor scores       -1   -1
```

```
plot(m.FAML2$values,type="b")
```




```
fa.diagram(m.FAML2)
```




```
# Try FA on data removing outlier --- FA on ML - Oblique
m.FAML2o <-fa(z.outl.df, 14, n.obs=480, rotate="oblimin", fm="ml")
print(m.FAML2o)
```

```
## Factor Analysis using method =  ml
## Call: fa(r = z.outl.df, nfactors = 14, n.obs = 480, rotate = "oblimin",
##     fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                    ML3   ML1   ML4   ML2   ML6   ML7   ML8   ML5 ML9 ML10 ML11
## loan_amnt         0.94 -0.07  0.00  0.11 -0.02  0.03  0.06 -0.05   0    0    0
## int_rate          0.06 -0.17 -0.01  0.38  0.29 -0.15  0.14  0.39   0    0    0
## installment       0.93  0.15  0.00 -0.08  0.02 -0.01 -0.04  0.06   0    0    0
## annual_inc        0.13  0.06  0.21  0.04  0.12  0.57 -0.01 -0.07   0    0    0
## dti               0.02  0.04  0.31  0.03  0.20 -0.51 -0.07 -0.10   0    0    0
## inq_last_6mths   -0.01  0.01  0.31 -0.01  0.02  0.09 -0.06  0.42   0    0    0
## open_acc          0.05 -0.09  0.78  0.00 -0.06 -0.08  0.04  0.01   0    0    0
## revol_bal         0.08  0.02  0.26  0.08  0.44  0.11  0.01 -0.30   0    0    0
## revol_util        0.01  0.06 -0.14 -0.01  0.70 -0.02 -0.01  0.07   0    0    0
## total_acc        -0.06  0.10  0.74  0.00 -0.03  0.11  0.02  0.00   0    0    0
## total_pymnt       0.13  0.70  0.00  0.26  0.02  0.01  0.06 -0.01   0    0    0
## total_rec_prncp   0.06  0.90  0.00 -0.01  0.02  0.01  0.08 -0.01   0    0    0
## total_rec_int     0.06  0.11  0.00  0.90  0.00  0.01 -0.03  0.00   0    0    0
## last_pymnt_amnt   0.01  0.11  0.03 -0.08 -0.02  0.01  0.71 -0.02   0    0    0
##                  ML12 ML13 ML14   h2     u2 com
## loan_amnt           0    0    0 0.97 0.0324 1.1
## int_rate            0    0    0 0.71 0.2907 3.9
```

```
## installment        0    0    0 0.97 0.0343 1.1
## annual_inc         0    0    0 0.54 0.4649 1.6
## dti                0    0    0 0.40 0.5951 2.2
## inq_last_6mths     0    0    0 0.21 0.7942 2.0
## open_acc           0    0    0 0.59 0.4053 1.1
## revol_bal          0    0    0 0.49 0.5126 2.8
## revol_util         0    0    0 0.53 0.4728 1.1
## total_acc          0    0    0 0.58 0.4205 1.1
## total_pymnt        0    0    0 1.00 0.0028 1.4
## total_rec_prncp    0    0    0 1.00 0.0038 1.0
## total_rec_int      0    0    0 0.98 0.0162 1.0
## last_pymnt_amnt    0    0    0 0.59 0.4117 1.1
##
##                         ML3  ML1  ML4  ML2  ML6  ML7  ML8  ML5  ML9 ML10 ML11
## SS loadings            2.14 1.79 1.50 1.33 0.95 0.70 0.65 0.50 0.00 0.00 0.00
## Proportion Var         0.15 0.13 0.11 0.09 0.07 0.05 0.05 0.04 0.00 0.00 0.00
## Cumulative Var         0.15 0.28 0.39 0.48 0.55 0.60 0.65 0.68 0.68 0.68 0.68
## Proportion Explained   0.22 0.19 0.16 0.14 0.10 0.07 0.07 0.05 0.00 0.00 0.00
## Cumulative Proportion  0.22 0.41 0.57 0.71 0.81 0.88 0.95 1.00 1.00 1.00 1.00
##                        ML12 ML13 ML14
## SS loadings            0.00 0.00 0.00
## Proportion Var         0.00 0.00 0.00
## Cumulative Var         0.68 0.68 0.68
## Proportion Explained   0.00 0.00 0.00
## Cumulative Proportion  1.00 1.00 1.00
##
##  With factor correlations of
##        ML3   ML1   ML4   ML2   ML6   ML7  ML8   ML5 ML9 ML10 ML11 ML12 ML13
## ML3   1.00  0.82  0.16  0.68  0.34  0.21 0.35 -0.04   0    0    0    0    0
## ML1   0.82  1.00  0.08  0.44  0.15  0.27 0.52 -0.20   0    0    0    0    0
## ML4   0.16  0.08  1.00  0.11  0.11  0.12 0.10 -0.20   0    0    0    0    0
## ML2   0.68  0.44  0.11  1.00  0.45 -0.07 0.14  0.23   0    0    0    0    0
## ML6   0.34  0.15  0.11  0.45  1.00 -0.18 0.07  0.11   0    0    0    0    0
## ML7   0.21  0.27  0.12 -0.07 -0.18  1.00 0.04 -0.25   0    0    0    0    0
## ML8   0.35  0.52  0.10  0.14  0.07  0.04 1.00  0.12   0    0    0    0    0
## ML5  -0.04 -0.20 -0.20  0.23  0.11 -0.25 0.12  1.00   0    0    0    0    0
## ML9   0.00  0.00  0.00  0.00  0.00  0.00 0.00  0.00   1    0    0    0    0
## ML10  0.00  0.00  0.00  0.00  0.00  0.00 0.00  0.00   0    1    0    0    0
## ML11  0.00  0.00  0.00  0.00  0.00  0.00 0.00  0.00   0    0    1    0    0
## ML12  0.00  0.00  0.00  0.00  0.00  0.00 0.00  0.00   0    0    0    1    0
## ML13  0.00  0.00  0.00  0.00  0.00  0.00 0.00  0.00   0    0    0    0    1
## ML14  0.00  0.00  0.00  0.00  0.00  0.00 0.00  0.00   0    0    0    0    0
##      ML14
## ML3     0
## ML1     0
## ML4     0
## ML2     0
## ML6     0
## ML7     0
## ML8     0
## ML5     0
## ML9     0
## ML10    0
## ML11    0
```

```
## ML12    0
## ML13    0
## ML14    1
##
## Mean item complexity =  1.6
## Test of the hypothesis that 14 factors are sufficient.
##
## df null model =  91  with the objective function =  14.74 with Chi Square =  6981.34
## df of  the model are -14  and the objective function was  0.99
##
## The root mean square of the residuals (RMSR) is  0.02
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  480 with the empirical chi square  40.22  with prob <  NA
## The total n.obs was  480  with Likelihood Chi Square =  457.51  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.454
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                  ML3  ML1  ML4  ML2  ML6  ML7
## Correlation of (regression) scores with factors   0.99 1.00 0.86 0.99 0.80 0.75
## Multiple R square of scores with factors          0.98 0.99 0.74 0.98 0.65 0.57
## Minimum correlation of possible factor scores     0.96 0.99 0.48 0.96 0.30 0.13
##                                                  ML8  ML5 ML9 ML10 ML11 ML12
## Correlation of (regression) scores with factors   0.82 0.75   0    0    0    0
## Multiple R square of scores with factors          0.67 0.56   0    0    0    0
## Minimum correlation of possible factor scores     0.34 0.12  -1   -1   -1   -1
##                                                  ML13 ML14
## Correlation of (regression) scores with factors     0    0
## Multiple R square of scores with factors            0    0
## Minimum correlation of possible factor scores      -1   -1
```

```
fscore_FAML2o <- m.FAML2o$scores
fscorematrix <- cor(fscore_FAML2o)
lowerCor(fscore_FAML2o)
```

```
##      ML3   ML1   ML4   ML2   ML6   ML7   ML8   ML5   ML9   ML10  ML11
## ML3   1.00
## ML1   0.83  1.00
## ML4   0.19  0.10  1.00
## ML2   0.70  0.45  0.13  1.00
## ML6   0.42  0.19  0.15  0.56  1.00
## ML7   0.28  0.36  0.19 -0.08 -0.26  1.00
## ML8   0.44  0.67  0.12  0.16  0.10  0.11  1.00
## ML5  -0.06 -0.27 -0.27  0.30  0.25 -0.47  0.05  1.00
## ML9   0.03 -0.21 -0.27  0.44  0.46 -0.62  0.03  0.96  1.00
## ML10  0.03 -0.21 -0.27  0.44  0.46 -0.62  0.03  0.96  1.00  1.00
## ML11  0.03 -0.21 -0.27  0.44  0.46 -0.62  0.03  0.96  1.00  1.00  1.00
## ML12  0.03 -0.21 -0.27  0.44  0.46 -0.62  0.03  0.96  1.00  1.00  1.00
## ML13  0.03 -0.21 -0.27  0.44  0.46 -0.62  0.03  0.96  1.00  1.00  1.00
## ML14  0.03 -0.21 -0.27  0.44  0.46 -0.62  0.03  0.96  1.00  1.00  1.00
##      ML12  ML13  ML14
## ML12  1.00
## ML13  1.00  1.00
```

```
## ML14  1.00  1.00  1.00
```

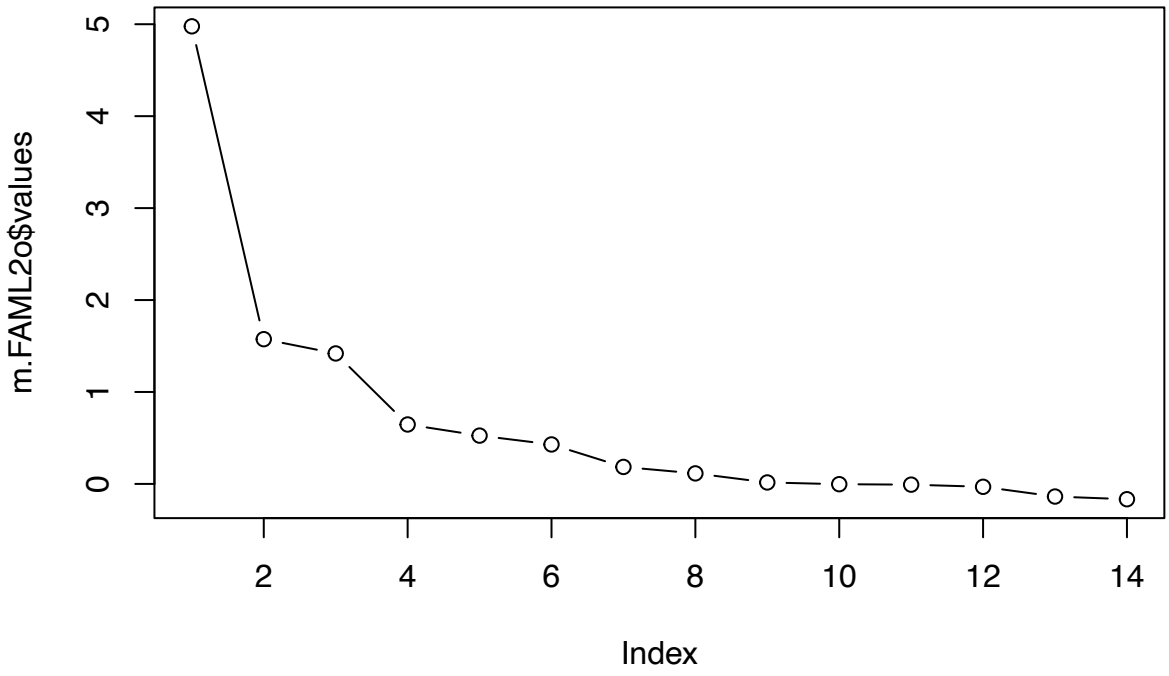
```
print.psych(m.FAML2o, cut=0.4, sort=TRUE)
```

```
## Factor Analysis using method =  ml
## Call: fa(r = z.outl.df, nfactors = 14, n.obs = 480, rotate = "oblimin",
##     fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  item   ML3   ML1   ML4   ML2   ML6   ML7   ML8   ML5   ML9
## loan_amnt           1  0.94
## installment         3  0.93
## total_rec_prncp    12        0.90
## total_pymnt        11        0.70
## open_acc            7              0.78
## total_acc          10              0.74
## total_rec_int      13                    0.90
## revol_util          9                          0.70
## revol_bal           8                          0.44
## annual_inc          4                                0.57
## dti                 5                               -0.51
## last_pymnt_amnt    14                                      0.71
## inq_last_6mths      6                                            0.42
## int_rate            2
##                  ML10  ML11  ML12  ML13  ML14   h2     u2 com
## loan_amnt                                     0.97 0.0324 1.1
## installment                                   0.97 0.0343 1.1
## total_rec_prncp                               1.00 0.0038 1.0
## total_pymnt                                   1.00 0.0028 1.4
## open_acc                                      0.59 0.4053 1.1
## total_acc                                     0.58 0.4205 1.1
## total_rec_int                                 0.98 0.0162 1.0
## revol_util                                    0.53 0.4728 1.1
## revol_bal                                     0.49 0.5126 2.8
## annual_inc                                    0.54 0.4649 1.6
## dti                                           0.40 0.5951 2.2
## last_pymnt_amnt                               0.59 0.4117 1.1
## inq_last_6mths                                0.21 0.7942 2.0
## int_rate                                      0.71 0.2907 3.9
##
##                        ML3  ML1  ML4  ML2  ML6  ML7  ML8  ML5  ML9 ML10 ML11
## SS loadings           2.14 1.79 1.50 1.33 0.95 0.70 0.65 0.50 0.00 0.00 0.00
## Proportion Var        0.15 0.13 0.11 0.09 0.07 0.05 0.05 0.04 0.00 0.00 0.00
## Cumulative Var        0.15 0.28 0.39 0.48 0.55 0.60 0.65 0.68 0.68 0.68 0.68
## Proportion Explained  0.22 0.19 0.16 0.14 0.10 0.07 0.07 0.05 0.00 0.00 0.00
## Cumulative Proportion 0.22 0.41 0.57 0.71 0.81 0.88 0.95 1.00 1.00 1.00 1.00
##                       ML12 ML13 ML14
## SS loadings           0.00 0.00 0.00
## Proportion Var        0.00 0.00 0.00
## Cumulative Var        0.68 0.68 0.68
## Proportion Explained  0.00 0.00 0.00
## Cumulative Proportion 1.00 1.00 1.00
##
##  With factor correlations of
##        ML3   ML1   ML4   ML2   ML6   ML7  ML8   ML5 ML9 ML10 ML11 ML12 ML13
```

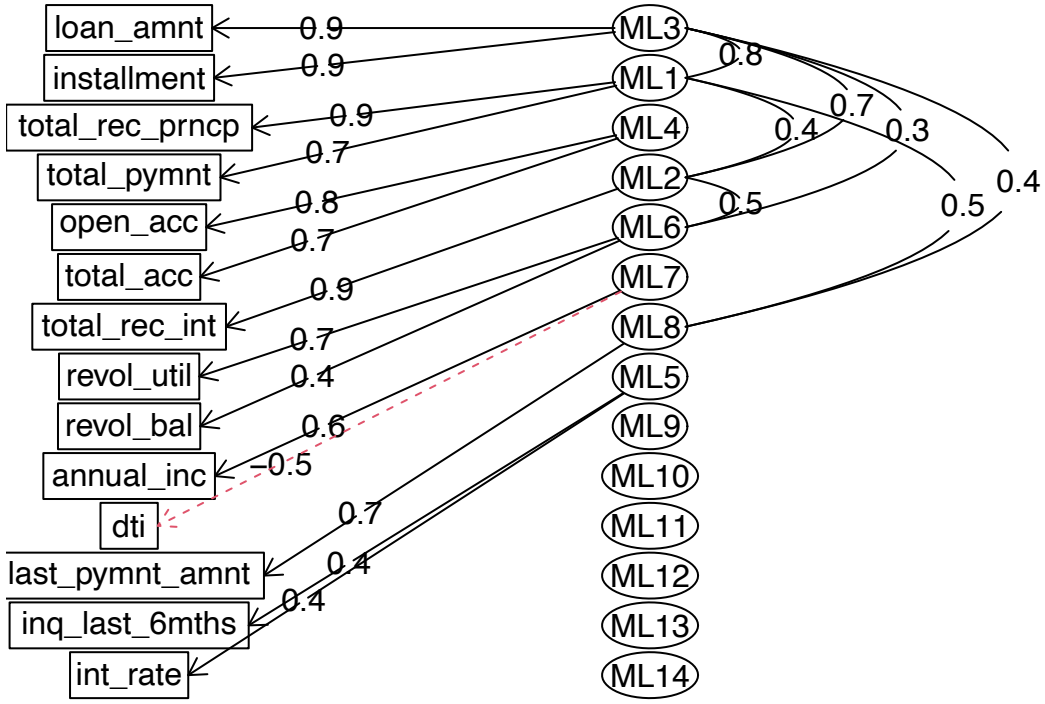
```
## ML3   1.00  0.82  0.16  0.68  0.34  0.21 0.35 -0.04   0    0    0    0    0
## ML1   0.82  1.00  0.08  0.44  0.15  0.27 0.52 -0.20   0    0    0    0    0
## ML4   0.16  0.08  1.00  0.11  0.11  0.12 0.10 -0.20   0    0    0    0    0
## ML2   0.68  0.44  0.11  1.00  0.45 -0.07 0.14  0.23   0    0    0    0    0
## ML6   0.34  0.15  0.11  0.45  1.00 -0.18 0.07  0.11   0    0    0    0    0
## ML7   0.21  0.27  0.12 -0.07 -0.18  1.00 0.04 -0.25   0    0    0    0    0
## ML8   0.35  0.52  0.10  0.14  0.07  0.04 1.00  0.12   0    0    0    0    0
## ML5  -0.04 -0.20 -0.20  0.23  0.11 -0.25 0.12  1.00   0    0    0    0    0
## ML9   0.00  0.00  0.00  0.00  0.00  0.00 0.00  0.00   1    0    0    0    0
## ML10  0.00  0.00  0.00  0.00  0.00  0.00 0.00  0.00   0    1    0    0    0
## ML11  0.00  0.00  0.00  0.00  0.00  0.00 0.00  0.00   0    0    1    0    0
## ML12  0.00  0.00  0.00  0.00  0.00  0.00 0.00  0.00   0    0    0    1    0
## ML13  0.00  0.00  0.00  0.00  0.00  0.00 0.00  0.00   0    0    0    0    1
## ML14  0.00  0.00  0.00  0.00  0.00  0.00 0.00  0.00   0    0    0    0    0
##      ML14
## ML3     0
## ML1     0
## ML4     0
## ML2     0
## ML6     0
## ML7     0
## ML8     0
## ML5     0
## ML9     0
## ML10    0
## ML11    0
## ML12    0
## ML13    0
## ML14    1
##
## Mean item complexity =  1.6
## Test of the hypothesis that 14 factors are sufficient.
##
## df null model =  91  with the objective function =  14.74 with Chi Square =  6981.34
## df of  the model are -14  and the objective function was  0.99
##
## The root mean square of the residuals (RMSR) is  0.02
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  480 with the empirical chi square  40.22  with prob <  NA
## The total n.obs was  480  with Likelihood Chi Square =  457.51  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.454
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                  ML3  ML1  ML4  ML2  ML6  ML7
## Correlation of (regression) scores with factors   0.99 1.00 0.86 0.99 0.80 0.75
## Multiple R square of scores with factors          0.98 0.99 0.74 0.98 0.65 0.57
## Minimum correlation of possible factor scores     0.96 0.99 0.48 0.96 0.30 0.13
##                                                  ML8  ML5 ML9 ML10 ML11 ML12
## Correlation of (regression) scores with factors   0.82 0.75   0    0    0    0
## Multiple R square of scores with factors          0.67 0.56   0    0    0    0
## Minimum correlation of possible factor scores     0.34 0.12  -1   -1   -1   -1
##                                                  ML13 ML14
```

```
## Correlation of (regression) scores with factors      0    0
## Multiple R square of scores with factors             0    0
## Minimum correlation of possible factor scores       -1   -1
```

```
plot(m.FAML2o$values,type="b")
```




```
fa.diagram(m.FAML2o)
```




```
# Try FA on data without removing outlier --- FA on ML - Orthogonal rotation --------- 2nd, model
m.FAML3 <- fa(z.df, 14, n.obs=500, rotate="varimax", fm="ml")
print(m.FAML3)
```

```
## Factor Analysis using method =  ml
## Call: fa(r = z.df, nfactors = 14, n.obs = 500, rotate = "varimax",
##     fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                    ML1   ML4   ML6   ML2   ML5   ML7   ML8   ML3 ML9 ML11 ML12
## loan_amnt         0.91  0.17  0.16  0.19  0.08 -0.05  0.02  0.19   0    0    0
## int_rate          0.19 -0.13  0.49  0.45  0.06  0.22  0.39  0.10   0    0    0
## installment       0.94  0.12  0.18  0.06  0.08 -0.04  0.05  0.15   0    0    0
## annual_inc        0.35  0.46  0.12 -0.03 -0.02 -0.45 -0.05  0.02   0    0    0
## dti               0.00  0.21  0.21  0.06 -0.05  0.56  0.02  0.01   0    0    0
## inq_last_6mths   -0.01  0.13  0.01  0.03  0.01  0.01  0.41  0.00   0    0    0
## open_acc          0.01  0.71 -0.09  0.03  0.02  0.18  0.20  0.04   0    0    0
## revol_bal         0.26  0.46  0.37 -0.02  0.06 -0.03 -0.19  0.03   0    0    0
## revol_util        0.15 -0.07  0.68  0.09  0.00  0.12  0.02 -0.01   0    0    0
## total_acc         0.11  0.73 -0.08  0.00  0.04  0.04  0.14 -0.05   0    0    0
## total_pymnt       0.93  0.10  0.12  0.18  0.23 -0.02 -0.03 -0.17   0    0    0
## total_rec_prncp   0.89  0.08  0.04 -0.06  0.36 -0.06 -0.08 -0.22   0    0    0
## total_rec_int     0.65  0.09  0.24  0.68 -0.14  0.09  0.10 -0.04   0    0    0
## last_pymnt_amnt   0.31  0.05  0.01 -0.03  0.72 -0.04  0.02  0.00   0    0    0
##                  ML13 ML10 ML14   h2     u2 com
## loan_amnt           0    0    0 0.97 0.0299 1.3
## int_rate            0    0    0 0.71 0.2942 4.0
```

```
## installment        0    0    0 0.97 0.0325 1.2
## annual_inc         0    0    0 0.56 0.4434 3.0
## dti                0    0    0 0.41 0.5928 1.6
## inq_last_6mths     0    0    0 0.19 0.8108 1.2
## open_acc           0    0    0 0.59 0.4140 1.3
## revol_bal          0    0    0 0.46 0.5445 3.0
## revol_util         0    0    0 0.51 0.4913 1.2
## total_acc          0    0    0 0.57 0.4262 1.2
## total_pymnt        0    0    0 1.00 0.0028 1.3
## total_rec_prncp    0    0    0 1.00 0.0038 1.5
## total_rec_int      0    0    0 0.99 0.0139 2.5
## last_pymnt_amnt    0    0    0 0.61 0.3874 1.4
##
##                         ML1  ML4  ML6  ML2  ML5  ML7  ML8  ML3  ML9 ML11 ML12
## SS loadings            4.14 1.61 1.03 0.76 0.74 0.63 0.45 0.16 0.00 0.00 0.00
## Proportion Var         0.30 0.11 0.07 0.05 0.05 0.05 0.03 0.01 0.00 0.00 0.00
## Cumulative Var         0.30 0.41 0.48 0.54 0.59 0.64 0.67 0.68 0.68 0.68 0.68
## Proportion Explained   0.44 0.17 0.11 0.08 0.08 0.07 0.05 0.02 0.00 0.00 0.00
## Cumulative Proportion  0.44 0.60 0.71 0.79 0.87 0.94 0.98 1.00 1.00 1.00 1.00
##                        ML13 ML10 ML14
## SS loadings            0.00 0.00 0.00
## Proportion Var         0.00 0.00 0.00
## Cumulative Var         0.68 0.68 0.68
## Proportion Explained   0.00 0.00 0.00
## Cumulative Proportion  1.00 1.00 1.00
##
## Mean item complexity =  1.8
## Test of the hypothesis that 14 factors are sufficient.
##
## df null model =  91  with the objective function =  14.8 with Chi Square =  7304.68
## df of  the model are -14  and the objective function was  0.96
##
## The root mean square of the residuals (RMSR) is  0.02
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  500 with the empirical chi square  41.9  with prob <  NA
## The total n.obs was  500  with Likelihood Chi Square =  462.93  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.438
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                  ML1  ML4  ML6  ML2  ML5
## Correlation of (regression) scores with factors   0.98 0.86 0.76 0.91 0.79
## Multiple R square of scores with factors          0.95 0.73 0.57 0.84 0.63
## Minimum correlation of possible factor scores     0.90 0.47 0.15 0.67 0.26
##                                                   ML7   ML8  ML3 ML9 ML11
## Correlation of (regression) scores with factors    0.68  0.63 0.87   0    0
## Multiple R square of scores with factors           0.47  0.40 0.76   0    0
## Minimum correlation of possible factor scores     -0.07 -0.20 0.52  -1   -1
##                                                  ML12 ML13 ML10 ML14
## Correlation of (regression) scores with factors     0    0    0    0
## Multiple R square of scores with factors            0    0    0    0
## Minimum correlation of possible factor scores      -1   -1   -1   -1
```

```r
fscore_FAML3 <- m.FAML3$scores
fscorematrix <- cor(fscore_FAML3)
lowerCor(fscore_FAML3)
```

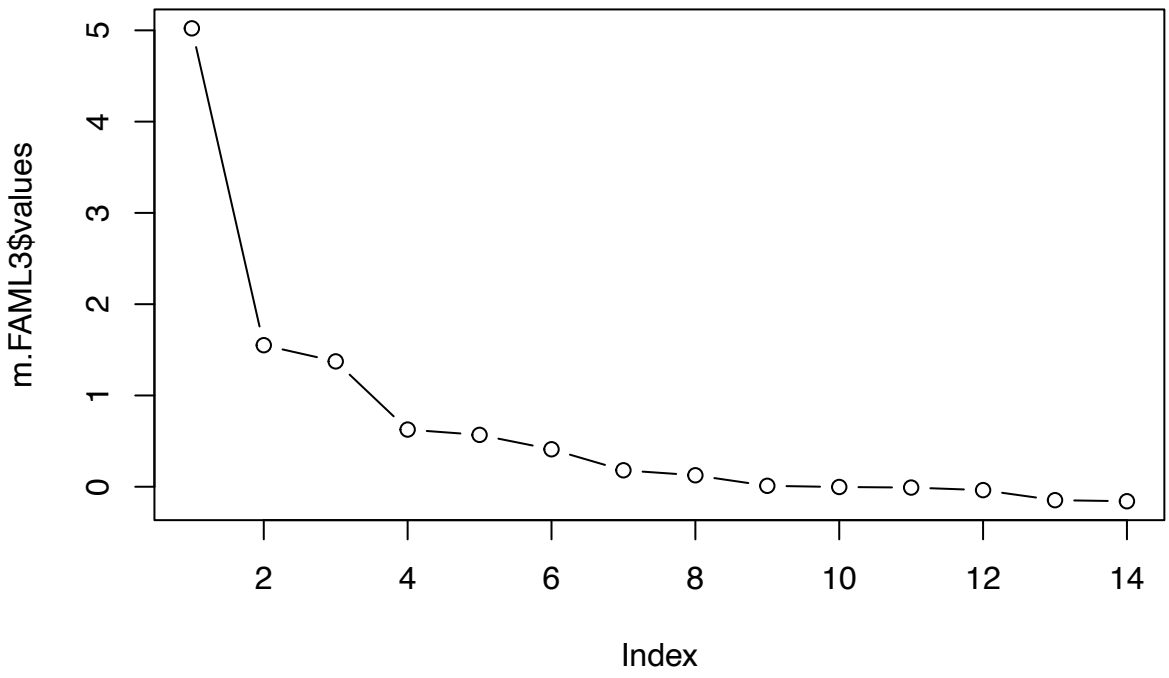
```
##       ML1   ML4   ML6   ML2   ML5   ML7   ML8   ML3   ML9   ML11  ML12
## ML1   1.00
## ML4   0.05  1.00
## ML6   0.08 -0.03  1.00
## ML2   0.06 -0.01  0.12  1.00
## ML5   0.13 -0.02 -0.01 -0.23  1.00
## ML7  -0.06  0.00  0.14  0.11  0.02  1.00
## ML8  -0.05  0.07  0.10  0.15  0.03  0.20  1.00
## ML3   0.01  0.04  0.10 -0.10 -0.23 -0.07  0.18  1.00
## ML9  -0.01  0.09  0.02  0.08 -0.09 -0.14 -0.19 -0.01  1.00
## ML11 -0.01  0.09  0.02  0.08 -0.09 -0.14 -0.19 -0.01  1.00  1.00
## ML12 -0.01  0.09  0.02  0.08 -0.09 -0.14 -0.19 -0.01  1.00  1.00  1.00
## ML13 -0.01  0.09  0.02  0.08 -0.09 -0.14 -0.19 -0.01  1.00  1.00  1.00
## ML10 -0.01  0.09  0.02  0.08 -0.09 -0.14 -0.19 -0.01  1.00  1.00  1.00
## ML14 -0.01  0.09  0.02  0.08 -0.09 -0.14 -0.19 -0.01  1.00  1.00  1.00
##       ML13  ML10  ML14
## ML13  1.00
## ML10  1.00  1.00
## ML14  1.00  1.00  1.00
```

```r
print.psych(m.FAML3, cut=0.4, sort=TRUE)
```

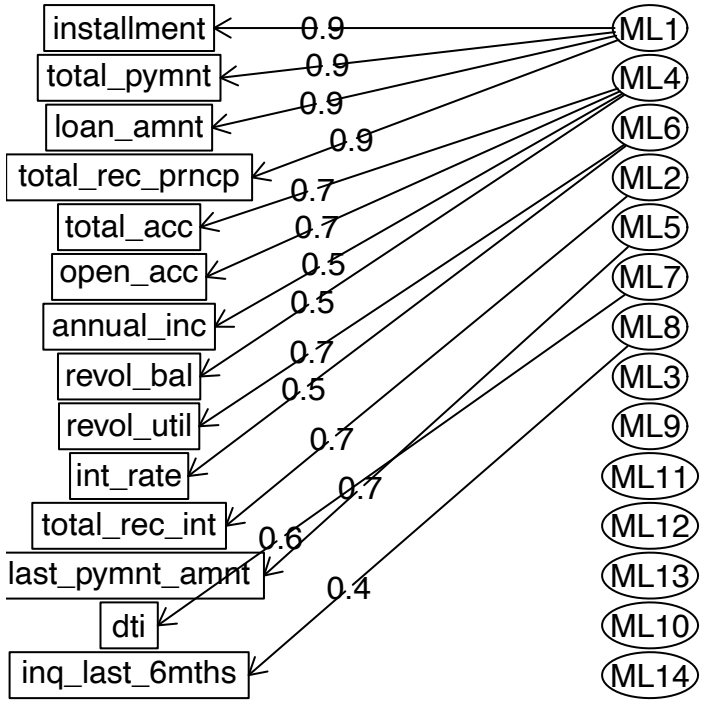
```
## Factor Analysis using method =  ml
## Call: fa(r = z.df, nfactors = 14, n.obs = 500, rotate = "varimax",
##     fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  item   ML1   ML4   ML6   ML2   ML5   ML7   ML8   ML3   ML9
## installment         3  0.94
## total_pymnt        11  0.93
## loan_amnt           1  0.91
## total_rec_prncp    12  0.89
## total_acc          10        0.73
## open_acc            7        0.71
## annual_inc          4        0.46                    -0.45
## revol_bal           8        0.46
## revol_util          9              0.68
## int_rate            2              0.49  0.45
## total_rec_int      13  0.65              0.68
## last_pymnt_amnt    14                          0.72
## dti                 5                                0.56
## inq_last_6mths      6                                      0.41
##                  ML11  ML12  ML13  ML10  ML14   h2     u2 com
## installment                                   0.97 0.0325 1.2
## total_pymnt                                   1.00 0.0028 1.3
## loan_amnt                                     0.97 0.0299 1.3
## total_rec_prncp                               1.00 0.0038 1.5
## total_acc                                     0.57 0.4262 1.2
## open_acc                                      0.59 0.4140 1.3
```

```
## annual_inc                                      0.56 0.4434 3.0
## revol_bal                                       0.46 0.5445 3.0
## revol_util                                      0.51 0.4913 1.2
## int_rate                                        0.71 0.2942 4.0
## total_rec_int                                   0.99 0.0139 2.5
## last_pymnt_amnt                                 0.61 0.3874 1.4
## dti                                             0.41 0.5928 1.6
## inq_last_6mths                                  0.19 0.8108 1.2
##
##                         ML1  ML4  ML6  ML2  ML5  ML7  ML8  ML3  ML9 ML11 ML12
## SS loadings            4.14 1.61 1.03 0.76 0.74 0.63 0.45 0.16 0.00 0.00 0.00
## Proportion Var         0.30 0.11 0.07 0.05 0.05 0.05 0.03 0.01 0.00 0.00 0.00
## Cumulative Var         0.30 0.41 0.48 0.54 0.59 0.64 0.67 0.68 0.68 0.68 0.68
## Proportion Explained   0.44 0.17 0.11 0.08 0.08 0.07 0.05 0.02 0.00 0.00 0.00
## Cumulative Proportion  0.44 0.60 0.71 0.79 0.87 0.94 0.98 1.00 1.00 1.00 1.00
##                        ML13 ML10 ML14
## SS loadings            0.00 0.00 0.00
## Proportion Var         0.00 0.00 0.00
## Cumulative Var         0.68 0.68 0.68
## Proportion Explained   0.00 0.00 0.00
## Cumulative Proportion  1.00 1.00 1.00
##
## Mean item complexity =  1.8
## Test of the hypothesis that 14 factors are sufficient.
##
## df null model =  91  with the objective function =  14.8 with Chi Square =  7304.68
## df of  the model are -14  and the objective function was  0.96
##
## The root mean square of the residuals (RMSR) is  0.02
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  500 with the empirical chi square  41.9  with prob <  NA
## The total n.obs was  500  with Likelihood Chi Square =  462.93  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.438
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                  ML1  ML4  ML6  ML2  ML5
## Correlation of (regression) scores with factors 0.98 0.86 0.76 0.91 0.79
## Multiple R square of scores with factors        0.95 0.73 0.57 0.84 0.63
## Minimum correlation of possible factor scores   0.90 0.47 0.15 0.67 0.26
##                                                  ML7   ML8  ML3 ML9 ML11
## Correlation of (regression) scores with factors 0.68  0.63 0.87   0    0
## Multiple R square of scores with factors        0.47  0.40 0.76   0    0
## Minimum correlation of possible factor scores  -0.07 -0.20 0.52  -1   -1
##                                                 ML12 ML13 ML10 ML14
## Correlation of (regression) scores with factors    0    0    0    0
## Multiple R square of scores with factors           0    0    0    0
## Minimum correlation of possible factor scores      -1   -1   -1   -1
```

```
plot(m.FAML3$values,type="b")
```




```
fa.diagram(m.FAML3)
```




```
# Try FA on data removing outlier --- FA on ML - Orthogonal rotation
m.FAML3o <- fa(z.outl.df, 14, n.obs=480, rotate="varimax", fm="ml")
print(m.FAML3o)
```

```
## Factor Analysis using method =  ml
## Call: fa(r = z.outl.df, nfactors = 14, n.obs = 480, rotate = "varimax",
##     fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                    ML1   ML4   ML5   ML6   ML7   ML8   ML2   ML3 ML13 ML14 ML9
## loan_amnt         0.93  0.13  0.16  0.08 -0.06  0.05  0.11 -0.20    0    0   0
## int_rate          0.20 -0.14  0.41  0.07  0.28  0.54  0.32 -0.08    0    0   0
## installment       0.95  0.08  0.16  0.07 -0.04  0.07 -0.03 -0.13    0    0   0
## annual_inc        0.34  0.38  0.09  0.00 -0.52 -0.04  0.00  0.00    0    0   0
## dti               0.02  0.26  0.23 -0.05  0.53  0.01  0.04  0.00    0    0   0
## inq_last_6mths   -0.02  0.16  0.01 -0.01 -0.02  0.42  0.00  0.01    0    0   0
## open_acc          0.00  0.73 -0.06  0.04  0.14  0.17  0.01 -0.04    0    0   0
## revol_bal         0.28  0.45  0.42 -0.02 -0.04 -0.16  0.06 -0.01    0    0   0
## revol_util        0.16 -0.10  0.68 -0.01  0.12  0.11  0.04  0.01    0    0   0
## total_acc         0.09  0.74 -0.06  0.07 -0.05  0.11 -0.02  0.04    0    0   0
## total_pymnt       0.93  0.07  0.12  0.24 -0.05 -0.03  0.13  0.17    0    0   0
## total_rec_prncp   0.88  0.07  0.06  0.36 -0.09 -0.13 -0.06  0.25    0    0   0
## total_rec_int     0.72  0.05  0.24 -0.10  0.09  0.22  0.58  0.01    0    0   0
## last_pymnt_amnt   0.26  0.07 -0.02  0.72 -0.03  0.01 -0.02  0.00    0    0   0
##                  ML11 ML12 ML10   h2     u2 com
## loan_amnt           0    0    0 0.97 0.0324 1.2
## int_rate            0    0    0 0.71 0.2907 3.9
```

```
## installment        0    0    0 0.97 0.0343 1.1
## annual_inc         0    0    0 0.54 0.4649 2.7
## dti                0    0    0 0.40 0.5951 1.9
## inq_last_6mths     0    0    0 0.21 0.7942 1.3
## open_acc           0    0    0 0.59 0.4053 1.2
## revol_bal          0    0    0 0.49 0.5126 3.0
## revol_util         0    0    0 0.53 0.4728 1.3
## total_acc          0    0    0 0.58 0.4205 1.1
## total_pymnt        0    0    0 1.00 0.0028 1.3
## total_rec_prncp    0    0    0 1.00 0.0038 1.6
## total_rec_int      0    0    0 0.98 0.0162 2.5
## last_pymnt_amnt    0    0    0 0.59 0.4117 1.3
##
##                         ML1  ML4  ML5  ML6  ML7  ML8  ML2  ML3 ML13 ML14  ML9
## SS loadings            4.26 1.59 1.01 0.73 0.69 0.62 0.48 0.15 0.00 0.00 0.00
## Proportion Var         0.30 0.11 0.07 0.05 0.05 0.04 0.03 0.01 0.00 0.00 0.00
## Cumulative Var         0.30 0.42 0.49 0.54 0.59 0.64 0.67 0.68 0.68 0.68 0.68
## Proportion Explained   0.45 0.17 0.11 0.08 0.07 0.06 0.05 0.02 0.00 0.00 0.00
## Cumulative Proportion  0.45 0.61 0.72 0.80 0.87 0.93 0.98 1.00 1.00 1.00 1.00
##                        ML11 ML12 ML10
## SS loadings            0.00 0.00 0.00
## Proportion Var         0.00 0.00 0.00
## Cumulative Var         0.68 0.68 0.68
## Proportion Explained   0.00 0.00 0.00
## Cumulative Proportion  1.00 1.00 1.00
##
## Mean item complexity =  1.8
## Test of the hypothesis that 14 factors are sufficient.
##
## df null model =  91  with the objective function =  14.74 with Chi Square =  6981.34
## df of  the model are -14  and the objective function was  0.99
##
## The root mean square of the residuals (RMSR) is  0.02
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  480 with the empirical chi square  40.22  with prob <  NA
## The total n.obs was  480  with Likelihood Chi Square =  457.51  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.454
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                  ML1  ML4  ML5  ML6   ML7
## Correlation of (regression) scores with factors   0.98 0.86 0.75 0.79  0.70
## Multiple R square of scores with factors          0.96 0.74 0.56 0.63  0.49
## Minimum correlation of possible factor scores     0.92 0.49 0.11 0.26 -0.02
##                                                   ML8  ML2  ML3 ML13 ML14 ML9
## Correlation of (regression) scores with factors    0.71 0.89 0.85    0    0   0
## Multiple R square of scores with factors           0.50 0.80 0.72    0    0   0
## Minimum correlation of possible factor scores     -0.01 0.60 0.44   -1   -1  -1
##                                                  ML11 ML12 ML10
## Correlation of (regression) scores with factors     0    0    0
## Multiple R square of scores with factors            0    0    0
## Minimum correlation of possible factor scores      -1   -1   -1
```

```r
fscore_FAML3o <- m.FAML3o$scores
fscorematrix <- cor(fscore_FAML3o)
lowerCor(fscore_FAML3o)
```

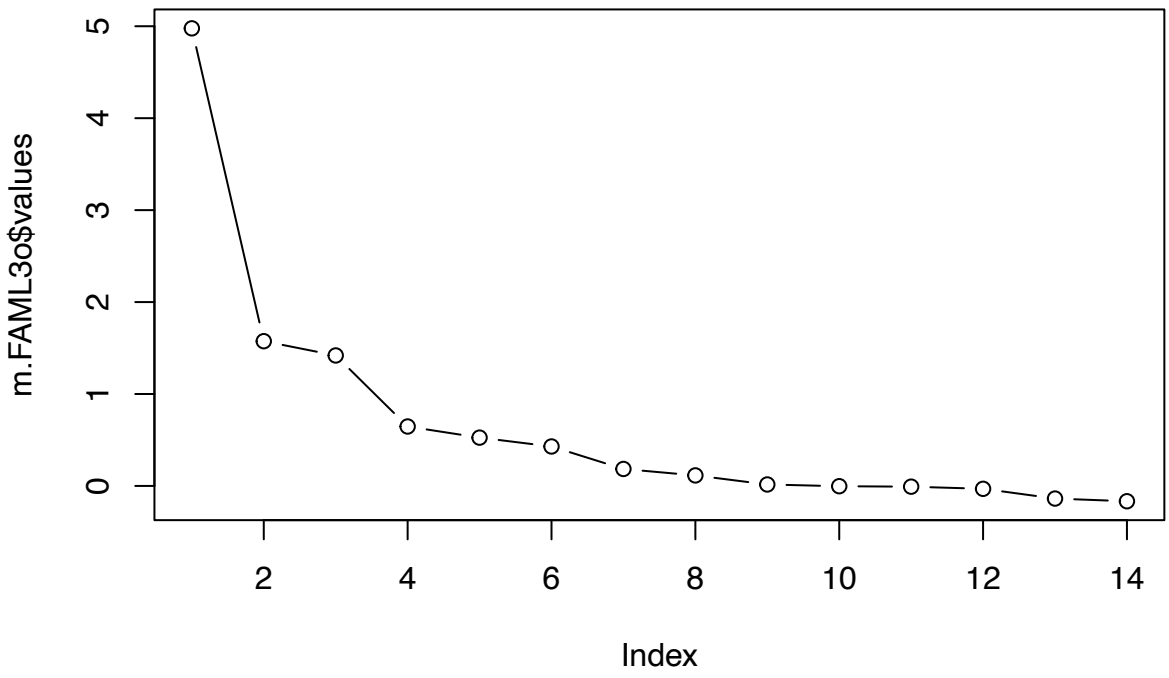
```
##       ML1   ML4   ML5   ML6   ML7   ML8   ML2   ML3   ML13  ML14  ML9
## ML1   1.00
## ML4   0.03  1.00
## ML5   0.10 -0.01  1.00
## ML6   0.11  0.01 -0.04  1.00
## ML7  -0.06 -0.03  0.16  0.01  1.00
## ML8  -0.02  0.03  0.15 -0.06  0.21  1.00
## ML2   0.05 -0.01  0.08 -0.19  0.10  0.23  1.00
## ML3  -0.01 -0.02 -0.04  0.27  0.02 -0.21  0.13  1.00
## ML13  0.00  0.09  0.02 -0.08 -0.12 -0.15  0.08  0.01  1.00
## ML14  0.00  0.09  0.02 -0.08 -0.12 -0.15  0.08  0.01  1.00  1.00
## ML9   0.00  0.09  0.02 -0.08 -0.12 -0.15  0.08  0.01  1.00  1.00  1.00
## ML11  0.00  0.09  0.02 -0.08 -0.12 -0.15  0.08  0.01  1.00  1.00  1.00
## ML12  0.00  0.09  0.02 -0.08 -0.12 -0.15  0.08  0.01  1.00  1.00  1.00
## ML10  0.00  0.09  0.02 -0.08 -0.12 -0.15  0.08  0.01  1.00  1.00  1.00
##       ML11  ML12  ML10
## ML11  1.00
## ML12  1.00  1.00
## ML10  1.00  1.00  1.00
```

```r
print.psych(m.FAML3o, cut=0.4, sort=TRUE)
```

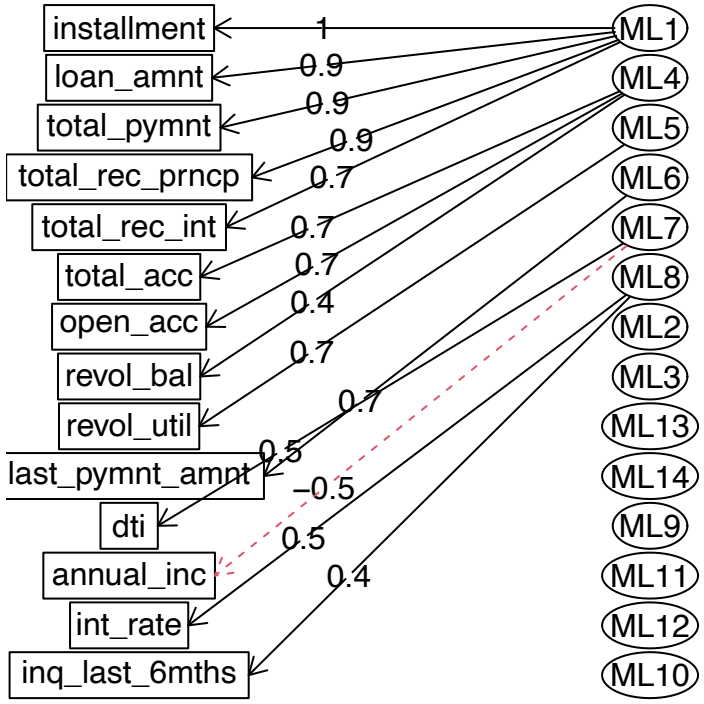
```
## Factor Analysis using method =  ml
## Call: fa(r = z.outl.df, nfactors = 14, n.obs = 480, rotate = "varimax",
##     fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  item   ML1   ML4   ML5   ML6   ML7   ML8   ML2   ML3  ML13
## installment         3  0.95
## loan_amnt           1  0.93
## total_pymnt        11  0.93
## total_rec_prncp    12  0.88
## total_rec_int      13  0.72                                 0.58
## total_acc          10        0.74
## open_acc            7        0.73
## revol_bal           8        0.45  0.42
## revol_util          9              0.68
## last_pymnt_amnt    14                    0.72
## dti                 5                          0.53
## annual_inc          4                         -0.52
## int_rate            2              0.41              0.54
## inq_last_6mths      6                                0.42
##                  ML14   ML9  ML11  ML12  ML10   h2     u2 com
## installment                                    0.97 0.0343 1.1
## loan_amnt                                      0.97 0.0324 1.2
## total_pymnt                                    1.00 0.0028 1.3
## total_rec_prncp                                1.00 0.0038 1.6
## total_rec_int                                  0.98 0.0162 2.5
## total_acc                                      0.58 0.4205 1.1
```

```
## open_acc                                        0.59 0.4053 1.2
## revol_bal                                       0.49 0.5126 3.0
## revol_util                                      0.53 0.4728 1.3
## last_pymnt_amnt                                 0.59 0.4117 1.3
## dti                                             0.40 0.5951 1.9
## annual_inc                                      0.54 0.4649 2.7
## int_rate                                        0.71 0.2907 3.9
## inq_last_6mths                                  0.21 0.7942 1.3
##
##                         ML1  ML4  ML5  ML6  ML7  ML8  ML2  ML3 ML13 ML14  ML9
## SS loadings            4.26 1.59 1.01 0.73 0.69 0.62 0.48 0.15 0.00 0.00 0.00
## Proportion Var         0.30 0.11 0.07 0.05 0.05 0.04 0.03 0.01 0.00 0.00 0.00
## Cumulative Var         0.30 0.42 0.49 0.54 0.59 0.64 0.67 0.68 0.68 0.68 0.68
## Proportion Explained   0.45 0.17 0.11 0.08 0.07 0.06 0.05 0.02 0.00 0.00 0.00
## Cumulative Proportion  0.45 0.61 0.72 0.80 0.87 0.93 0.98 1.00 1.00 1.00 1.00
##                        ML11 ML12 ML10
## SS loadings            0.00 0.00 0.00
## Proportion Var         0.00 0.00 0.00
## Cumulative Var         0.68 0.68 0.68
## Proportion Explained   0.00 0.00 0.00
## Cumulative Proportion  1.00 1.00 1.00
##
## Mean item complexity =  1.8
## Test of the hypothesis that 14 factors are sufficient.
##
## df null model =  91  with the objective function =  14.74 with Chi Square =  6981.34
## df of  the model are -14  and the objective function was  0.99
##
## The root mean square of the residuals (RMSR) is  0.02
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  480 with the empirical chi square  40.22  with prob <  NA
## The total n.obs was  480  with Likelihood Chi Square =  457.51  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  1.454
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##                                                  ML1  ML4  ML5  ML6   ML7
## Correlation of (regression) scores with factors 0.98 0.86 0.75 0.79  0.70
## Multiple R square of scores with factors        0.96 0.74 0.56 0.63  0.49
## Minimum correlation of possible factor scores   0.92 0.49 0.11 0.26 -0.02
##                                                  ML8  ML2  ML3 ML13 ML14 ML9
## Correlation of (regression) scores with factors 0.71 0.89 0.85    0    0   0
## Multiple R square of scores with factors        0.50 0.80 0.72    0    0   0
## Minimum correlation of possible factor scores  -0.01 0.60 0.44   -1   -1  -1
##                                                 ML11 ML12 ML10
## Correlation of (regression) scores with factors    0    0    0
## Multiple R square of scores with factors           0    0    0
## Minimum correlation of possible factor scores     -1   -1   -1
```

```
plot(m.FAML3o$values,type="b")
```




```
fa.diagram(m.FAML3o)
```



from the PCA and Factor Results, we think that the components of FA - Maximum Likelihood with no rotation and FA - Maximum Likelihood with Oblique rotation make sense for reduce dimension, both model can explain variance around 70% within 3 components while contain majority of variables information. Moreover, both model correlation matrix did not show highly multicollinearity (>0.8) between each component which is suitable for use in cluster analysis. And we consider between using data that keep outlier and remove outlier, we found that, fot the data that keep outlier, it has less cross-loading but also explain less variance compare to data that remove outlier and the cross-loading appear in total_record_interest, which basicly depend on loan amount and interest rate and make sense to cross-load on to 2 components we have. In conclusion, we dicide to do further analysis on the dataset that have less cross-loading (keep outlier) because the different in variance explained is not much compare to the cross-loading, furthermore, we will remove multivariate outlier again in cluster analysis.

Note –* if want to reduce dimensional > use ML > 2 - 3 components is enough to cover 60% - 70% variance, easy to interprete, have some correlation but still below 0.8**

***** if want to use FA on PC > will result in 5 components to cover 60%, no multi**

```
# Try FA on data without removing outlier --- FA on ML - no rotate - run again with 3 components
m.FAMLa <- fa(z.df, 3, n.obs=500, rotate="none", fm="ml", scores="regression")
print(m.FAMLa)
```

```
## Factor Analysis using method =  ml
## Call: fa(r = z.df, nfactors = 3, n.obs = 500, rotate = "none", scores = "regression",
##     fm = "ml")
```

```
## Standardized loadings (pattern matrix) based upon correlation matrix
##                   ML1   ML2   ML3    h2     u2 com
## loan_amnt        0.95  0.04  0.29 0.995 0.0050 1.2
## int_rate         0.34  0.51 -0.07 0.379 0.6211 1.8
## installment      0.94 -0.07  0.25 0.947 0.0530 1.1
## annual_inc       0.40 -0.10  0.22 0.215 0.7854 1.7
## dti              0.05  0.20 -0.02 0.044 0.9562 1.1
## inq_last_6mths   0.01  0.10  0.02 0.011 0.9893 1.1
## open_acc         0.08  0.08  0.12 0.028 0.9717 2.5
## revol_bal        0.36 -0.02  0.15 0.149 0.8508 1.4
## revol_util       0.26  0.20 -0.02 0.107 0.8926 1.9
## total_acc        0.19 -0.02  0.09 0.043 0.9567 1.5
## total_pymnt      0.99 -0.13 -0.10 0.997 0.0026 1.1
## total_rec_prncp  0.90 -0.42 -0.11 0.996 0.0037 1.4
## total_rec_int    0.80  0.58 -0.12 0.995 0.0048 1.9
## last_pymnt_amnt  0.40 -0.42 -0.02 0.335 0.6650 2.0
##
##                         ML1  ML2  ML3
## SS loadings            4.88 1.08 0.28
## Proportion Var         0.35 0.08 0.02
## Cumulative Var         0.35 0.43 0.45
## Proportion Explained   0.78 0.17 0.05
## Cumulative Proportion  0.78 0.95 1.00
##
## Mean item complexity =  1.5
## Test of the hypothesis that 3 factors are sufficient.
##
## df null model =  91  with the objective function =  14.8 with Chi Square =  7304.68
## df of  the model are 52  and the objective function was  2.5
##
## The root mean square of the residuals (RMSR) is  0.11
## The df corrected root mean square of the residuals is  0.15
##
## The harmonic n.obs is  500 with the empirical chi square  1201.61  with prob <  2.3e-217
## The total n.obs was  500  with Likelihood Chi Square =  1228.42  with prob <  6.1e-223
##
## Tucker Lewis Index of factoring reliability =  0.713
## RMSEA index =  0.213  and the 90 % confidence intervals are  0.203 0.223
## BIC =  905.26
## Fit based upon off diagonal values = 0.9
## Measures of factor score adequacy
##                                                 ML1  ML2  ML3
## Correlation of (regression) scores with factors   1 1.00 0.98
## Multiple R square of scores with factors          1 0.99 0.96
## Minimum correlation of possible factor scores     1 0.98 0.93
```

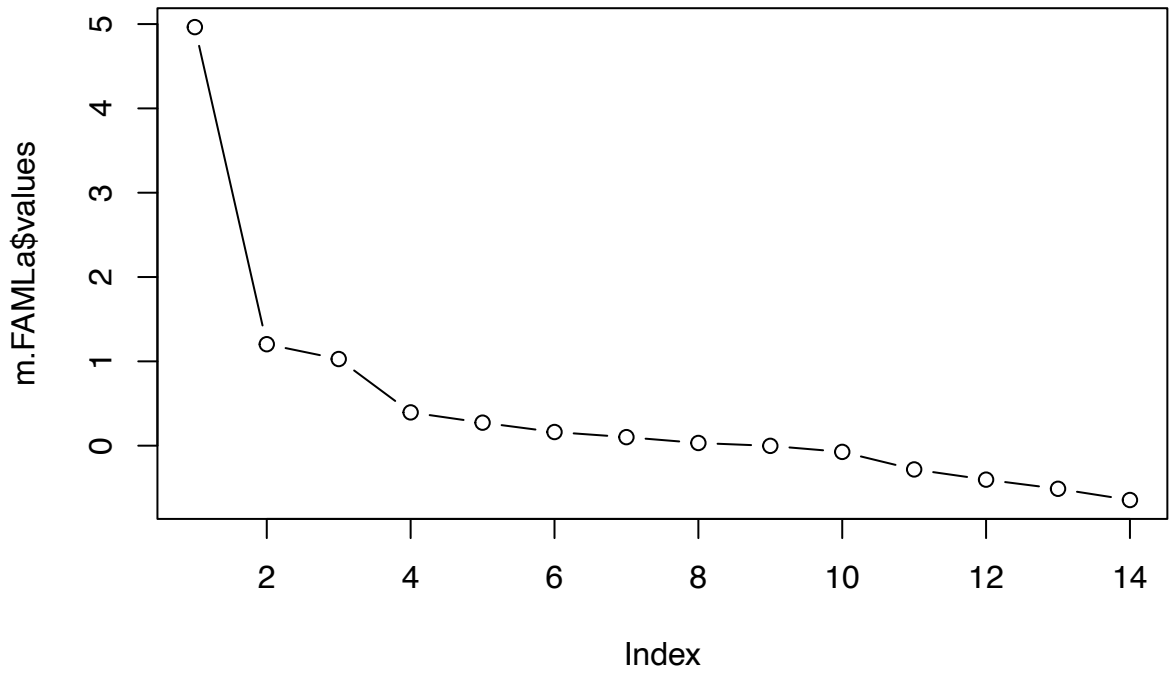
```
fscore_FAMLa <- m.FAMLa$scores
fscorematrix <- cor(fscore_FAMLa)
lowerCor(fscore_FAMLa)
```

```
##     ML1 ML2 ML3
## ML1 1
## ML2 0   1
## ML3 0   0   1
```

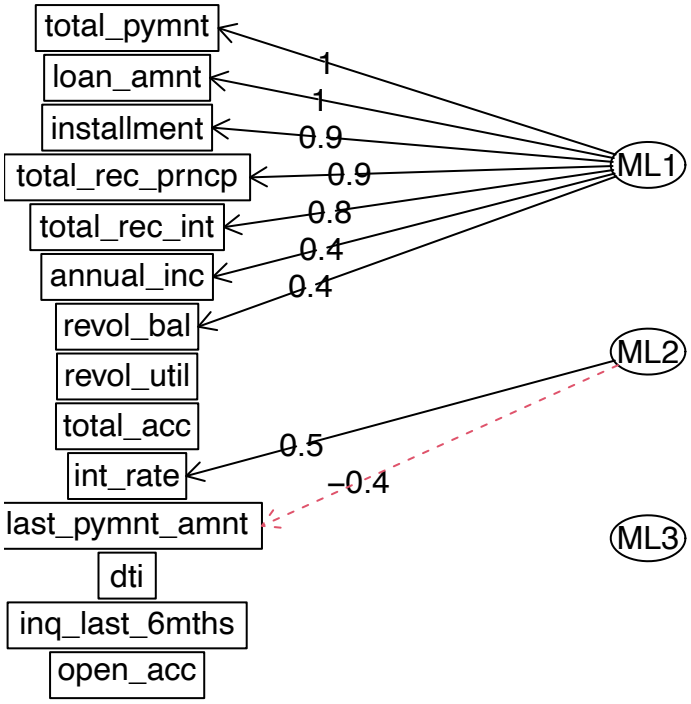
```
print.psych(m.FAMLa, cut=0.4, sort=TRUE)
```

```
## Factor Analysis using method =  ml
## Call: fa(r = z.df, nfactors = 3, n.obs = 500, rotate = "none", scores = "regression",
##     fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  item  ML1   ML2  ML3    h2     u2 com
## total_pymnt        11 0.99            0.997 0.0026 1.1
## loan_amnt           1 0.95            0.995 0.0050 1.2
## installment         3 0.94            0.947 0.0530 1.1
## total_rec_prncp    12 0.90 -0.42      0.996 0.0037 1.4
## total_rec_int      13 0.80  0.58      0.995 0.0048 1.9
## annual_inc          4                 0.215 0.7854 1.7
## revol_bal           8                 0.149 0.8508 1.4
## revol_util          9                 0.107 0.8926 1.9
## total_acc          10                 0.043 0.9567 1.5
## int_rate            2       0.51      0.379 0.6211 1.8
## last_pymnt_amnt    14      -0.42      0.335 0.6650 2.0
## dti                 5                 0.044 0.9562 1.1
## inq_last_6mths      6                 0.011 0.9893 1.1
## open_acc            7                 0.028 0.9717 2.5
##
##                        ML1  ML2  ML3
## SS loadings           4.88 1.08 0.28
## Proportion Var        0.35 0.08 0.02
## Cumulative Var        0.35 0.43 0.45
## Proportion Explained  0.78 0.17 0.05
## Cumulative Proportion 0.78 0.95 1.00
##
## Mean item complexity =  1.5
## Test of the hypothesis that 3 factors are sufficient.
##
## df null model =  91  with the objective function =  14.8 with Chi Square =  7304.68
## df of  the model are 52  and the objective function was  2.5
##
## The root mean square of the residuals (RMSR) is  0.11
## The df corrected root mean square of the residuals is  0.15
##
## The harmonic n.obs is  500 with the empirical chi square  1201.61  with prob <  2.3e-217
## The total n.obs was  500  with Likelihood Chi Square =  1228.42  with prob <  6.1e-223
##
## Tucker Lewis Index of factoring reliability =  0.713
## RMSEA index =  0.213  and the 90 % confidence intervals are  0.203 0.223
## BIC =  905.26
## Fit based upon off diagonal values = 0.9
## Measures of factor score adequacy
##                                                 ML1  ML2  ML3
## Correlation of (regression) scores with factors   1 1.00 0.98
## Multiple R square of scores with factors          1 0.99 0.96
## Minimum correlation of possible factor scores     1 0.98 0.93
```

```
plot(m.FAMLa$values,type="b")
```




```
fa.diagram(m.FAMLa)
```




```
# Prepare dataframe for further analysis - Model 1 - FA - ML - no rotate
head(m.FAMLa$scores, 10)
```

```
##              ML1          ML2          ML3
##  [1,] -0.8715353 -0.432381044  0.054168820
##  [2,] -0.1582354 -0.538062053  0.004499027
##  [3,]  2.5430851 -2.274164609  1.064604585
##  [4,] -0.8571602  0.493263577  0.314256408
##  [5,] -1.1056402  0.062417545 -0.408938769
##  [6,] -1.5103006  0.366496594 -0.630515157
##  [7,]  1.0575803 -0.479506536 -0.282573923
##  [8,]  1.8696597  3.752440413 -1.318641133
##  [9,]  1.1180960 -2.625993537  1.904244406
## [10,] -0.5835854 -0.001780086 -0.467082542
```

```
CA.df.FAMLa <- cbind(m.FAMLa$scores)
```

```
# Try FA on data without removing outlier --- FA on ML - Orthogonal rotation - run again with 2 compone
m.FAMLb <- fa(z.df, 3, n.obs=500, rotate="varimax", fm="ml")
print(m.FAMLb)
```

```
## Factor Analysis using method =  ml
## Call: fa(r = z.df, nfactors = 3, n.obs = 500, rotate = "varimax", fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
```

```
##                   ML1   ML2   ML3    h2     u2 com
## loan_amnt        0.87  0.33  0.37 0.995 0.0050 1.7
## int_rate         0.12  0.60  0.07 0.379 0.6211 1.1
## installment      0.89  0.24  0.30 0.947 0.0530 1.4
## annual_inc       0.41  0.01  0.22 0.215 0.7854 1.5
## dti             -0.03  0.21  0.03 0.044 0.9562 1.1
## inq_last_6mths  -0.03  0.09  0.04 0.011 0.9893 1.6
## open_acc         0.05  0.07  0.14 0.028 0.9717 1.7
## revol_bal        0.34  0.08  0.17 0.149 0.8508 1.6
## revol_util       0.16  0.28  0.05 0.107 0.8926 1.6
## total_acc        0.18  0.03  0.10 0.043 0.9567 1.6
## total_pymnt      0.96  0.28 -0.05 0.997 0.0026 1.2
## total_rec_prncp  0.99 -0.01 -0.12 0.996 0.0037 1.0
## total_rec_int    0.52  0.85  0.07 0.995 0.0048 1.7
## last_pymnt_amnt  0.53 -0.22 -0.08 0.335 0.6650 1.4
##
##                        ML1  ML2  ML3
## SS loadings           4.35 1.52 0.37
## Proportion Var        0.31 0.11 0.03
## Cumulative Var        0.31 0.42 0.45
## Proportion Explained  0.70 0.24 0.06
## Cumulative Proportion 0.70 0.94 1.00
##
## Mean item complexity =  1.4
## Test of the hypothesis that 3 factors are sufficient.
##
## df null model =  91  with the objective function =  14.8 with Chi Square =  7304.68
## df of  the model are 52  and the objective function was  2.5
##
## The root mean square of the residuals (RMSR) is  0.11
## The df corrected root mean square of the residuals is  0.15
##
## The harmonic n.obs is  500 with the empirical chi square  1201.61  with prob <  2.3e-217
## The total n.obs was  500  with Likelihood Chi Square =  1228.42  with prob <  6.1e-223
##
## Tucker Lewis Index of factoring reliability =  0.713
## RMSEA index =  0.213  and the 90 % confidence intervals are  0.203 0.223
## BIC =  905.26
## Fit based upon off diagonal values = 0.9
## Measures of factor score adequacy
##                                                 ML1  ML2  ML3
## Correlation of (regression) scores with factors   1.00 1.00 0.98
## Multiple R square of scores with factors          1.00 0.99 0.96
## Minimum correlation of possible factor scores     0.99 0.98 0.93
```

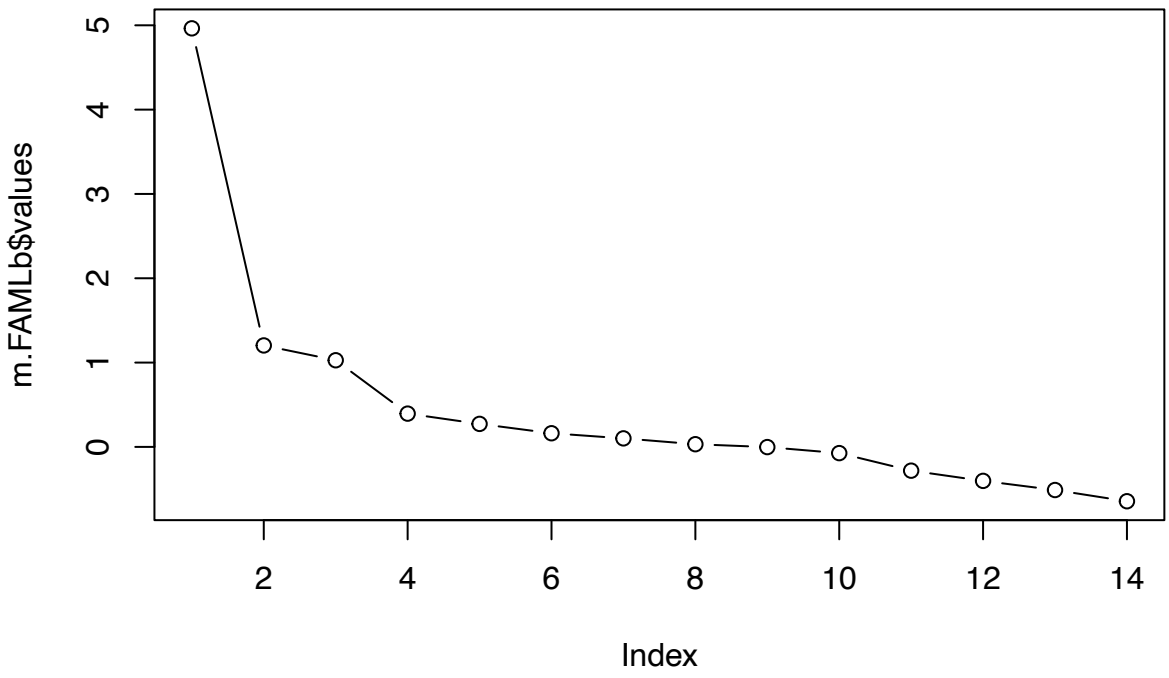
```
fscore_FAMLb <- m.FAMLb$scores
fscorematrix <- cor(fscore_FAMLb)
lowerCor(fscore_FAMLb)
```

```
##     ML1  ML2  ML3
## ML1 1.00
## ML2 0.00 1.00
## ML3 0.00 0.01 1.00
```

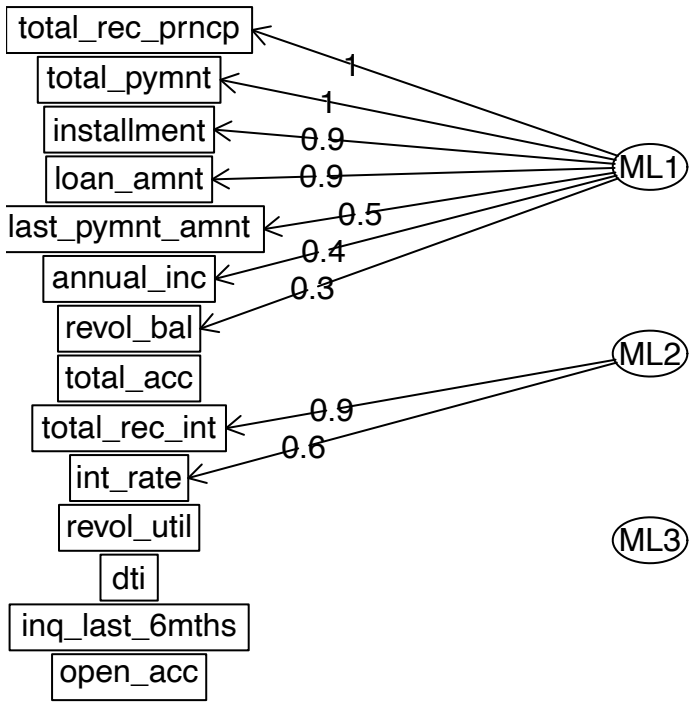
```
print.psych(m.FAMLb, cut=0.4, sort=TRUE)
```

```
## Factor Analysis using method =  ml
## Call: fa(r = z.df, nfactors = 3, n.obs = 500, rotate = "varimax", fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  item   ML1   ML2   ML3    h2     u2 com
## total_rec_prncp    12  0.99             0.996 0.0037 1.0
## total_pymnt        11  0.96             0.997 0.0026 1.2
## installment         3  0.89             0.947 0.0530 1.4
## loan_amnt           1  0.87             0.995 0.0050 1.7
## last_pymnt_amnt    14  0.53             0.335 0.6650 1.4
## annual_inc          4  0.41             0.215 0.7854 1.5
## revol_bal           8                   0.149 0.8508 1.6
## total_acc          10                   0.043 0.9567 1.6
## total_rec_int      13  0.52  0.85       0.995 0.0048 1.7
## int_rate            2        0.60       0.379 0.6211 1.1
## revol_util          9                   0.107 0.8926 1.6
## dti                 5                   0.044 0.9562 1.1
## inq_last_6mths      6                   0.011 0.9893 1.6
## open_acc            7                   0.028 0.9717 1.7
##
##                        ML1  ML2  ML3
## SS loadings           4.35 1.52 0.37
## Proportion Var        0.31 0.11 0.03
## Cumulative Var        0.31 0.42 0.45
## Proportion Explained  0.70 0.24 0.06
## Cumulative Proportion 0.70 0.94 1.00
##
## Mean item complexity =  1.4
## Test of the hypothesis that 3 factors are sufficient.
##
## df null model =  91  with the objective function =  14.8 with Chi Square =  7304.68
## df of  the model are 52  and the objective function was  2.5
##
## The root mean square of the residuals (RMSR) is  0.11
## The df corrected root mean square of the residuals is  0.15
##
## The harmonic n.obs is  500 with the empirical chi square  1201.61  with prob <  2.3e-217
## The total n.obs was  500  with Likelihood Chi Square =  1228.42  with prob <  6.1e-223
##
## Tucker Lewis Index of factoring reliability =  0.713
## RMSEA index =  0.213  and the 90 % confidence intervals are  0.203 0.223
## BIC =  905.26
## Fit based upon off diagonal values = 0.9
## Measures of factor score adequacy
##                                                 ML1  ML2  ML3
## Correlation of (regression) scores with factors   1.00 1.00 0.98
## Multiple R square of scores with factors          1.00 0.99 0.96
## Minimum correlation of possible factor scores     0.99 0.98 0.93
```

```
plot(m.FAMLb$values,type="b")
```




```
fa.diagram(m.FAMLb)
```




```
# Prepare dataframe for further analysis - Model 2 - FA - ML - Orthogonal rotation
head(m.FAMLb$scores, 10)
```

```
##               ML1         ML2        ML3
##  [1,] -0.63789064 -0.72883397 -0.1065739
##  [2,]  0.06092435 -0.54463707 -0.1192817
##  [3,]  3.22744562 -1.32524287  0.7744777
##  [4,] -0.97936769  0.05187441  0.3390246
##  [5,] -1.04660992 -0.26921207 -0.4750785
##  [6,] -1.53819847 -0.09878877 -0.6611076
##  [7,]  1.15923227  0.02887666 -0.2891321
##  [8,]  0.27587503  4.37130662 -0.3616935
##  [9,]  2.05161861 -2.36358073  1.4058715
## [10,] -0.54032276 -0.11774803 -0.5029194
```

```
CA.df.FAMLb <- cbind(m.FAMLb$scores)
```

```
# Try FA on data without removing outlier --- FA on PC - Orthogonal Rotation ----- > 3rd model
m.FAPCc <-principal(z.df, 5, rotate="quartimax")
print(m.FAPCc)
```

```
## Principal Components Analysis
## Call: principal(r = z.df, nfactors = 5, rotate = "quartimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
```

```
##                   RC1   RC2   RC3   RC5   RC4   h2    u2 com
## loan_amnt        0.94  0.07  0.13 -0.09  0.03 0.91 0.088 1.1
## int_rate         0.32 -0.10  0.63  0.27  0.41 0.76 0.241 2.8
## installment      0.93  0.04  0.12 -0.09  0.03 0.90 0.103 1.1
## annual_inc       0.41  0.33 -0.03 -0.72 -0.07 0.80 0.197 2.1
## dti             -0.02  0.44  0.44  0.61 -0.12 0.77 0.227 2.8
## inq_last_6mths   0.00  0.22  0.02 -0.05  0.83 0.74 0.264 1.1
## open_acc         0.05  0.86 -0.04  0.05  0.13 0.77 0.228 1.1
## revol_bal        0.35  0.47  0.32 -0.24 -0.40 0.66 0.339 4.3
## revol_util       0.23 -0.10  0.78  0.02 -0.15 0.69 0.313 1.3
## total_acc        0.17  0.83 -0.10 -0.11  0.08 0.74 0.260 1.2
## total_pymnt      0.98  0.02  0.03  0.00 -0.01 0.95 0.046 1.0
## total_rec_prncp  0.93  0.01 -0.16  0.01 -0.11 0.91 0.091 1.1
## total_rec_int    0.70  0.02  0.46 -0.02  0.23 0.76 0.244 2.0
## last_pymnt_amnt  0.56  0.03 -0.39  0.36 -0.07 0.61 0.391 2.6
##
##                        RC1  RC2  RC3  RC5  RC4
## SS loadings           4.86 2.02 1.74 1.19 1.15
## Proportion Var        0.35 0.14 0.12 0.09 0.08
## Cumulative Var        0.35 0.49 0.62 0.70 0.78
## Proportion Explained  0.44 0.18 0.16 0.11 0.11
## Cumulative Proportion 0.44 0.63 0.79 0.89 1.00
##
## Mean item complexity =  1.8
## Test of the hypothesis that 5 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##  with the empirical chi square  446.43  with prob <  4.2e-75
##
## Fit based upon off diagonal values = 0.96
```

```
fscore_FAPCc <- m.FAPCc$scores
fscorematrix <- cor(fscore_FAPCc)
lowerCor(fscore_FAPCc)
```

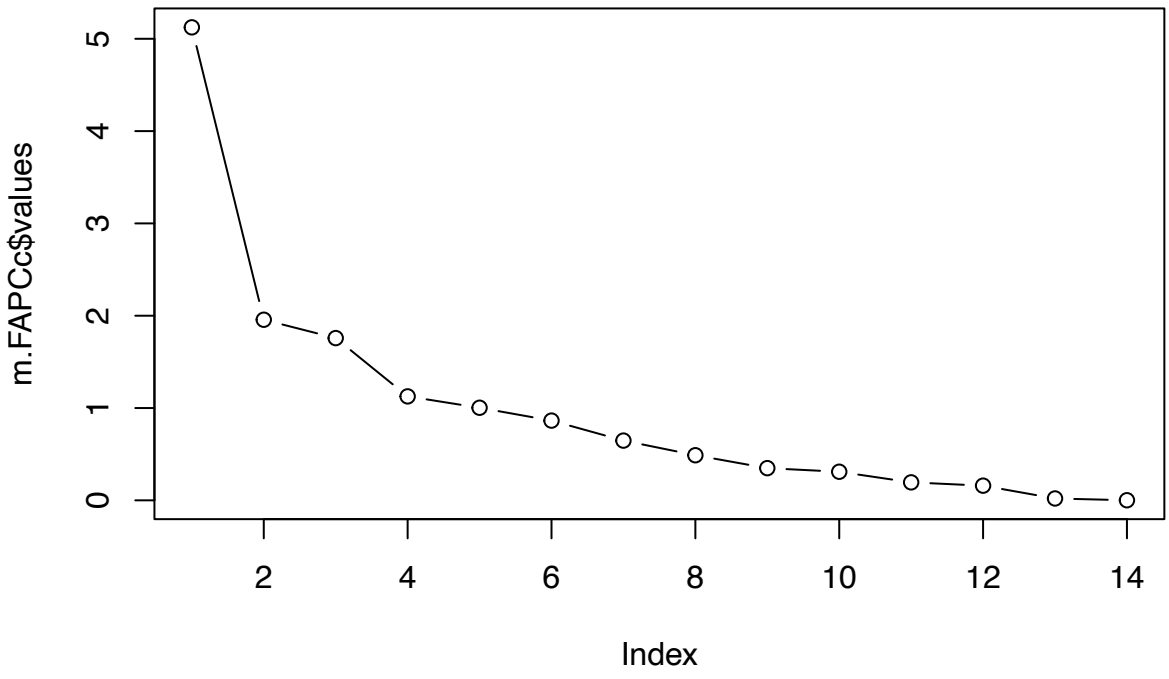
```
##     RC1 RC2 RC3 RC5 RC4
## RC1 1
## RC2 0   1
## RC3 0   0   1
## RC5 0   0   0   1
## RC4 0   0   0   0   1
```

```
print.psych(m.FAPCc, cut=0.4, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = z.df, nfactors = 5, rotate = "quartimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                  item   RC1   RC2   RC3   RC5   RC4   h2    u2 com
## total_pymnt        11  0.98                        0.95 0.046 1.0
## loan_amnt           1  0.94                        0.91 0.088 1.1
## installment         3  0.93                        0.90 0.103 1.1
## total_rec_prncp    12  0.93                        0.91 0.091 1.1
## total_rec_int      13  0.70        0.46            0.76 0.244 2.0
```

```
## last_pymnt_amnt   14  0.56                          0.61 0.391 2.6
## open_acc           7        0.86                    0.77 0.228 1.1
## total_acc         10        0.83                    0.74 0.260 1.2
## revol_bal          8        0.47                    0.66 0.339 4.3
## revol_util         9             0.78               0.69 0.313 1.3
## int_rate           2             0.63        0.41 0.76 0.241 2.8
## annual_inc         4  0.41            -0.72        0.80 0.197 2.1
## dti                5        0.44  0.44  0.61        0.77 0.227 2.8
## inq_last_6mths     6                          0.83 0.74 0.264 1.1
##
##                        RC1  RC2  RC3  RC5  RC4
## SS loadings           4.86 2.02 1.74 1.19 1.15
## Proportion Var        0.35 0.14 0.12 0.09 0.08
## Cumulative Var        0.35 0.49 0.62 0.70 0.78
## Proportion Explained  0.44 0.18 0.16 0.11 0.11
## Cumulative Proportion 0.44 0.63 0.79 0.89 1.00
##
## Mean item complexity =  1.8
## Test of the hypothesis that 5 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##  with the empirical chi square  446.43  with prob <  4.2e-75
##
## Fit based upon off diagonal values = 0.96
```

```
plot(m.FAPCc$values,type="b")
```



```r
# Prepare dataframe for further analysis - Model 3 - FA - PC - Orthogonal rotation
head(m.FAPCc$scores, 10)
```

```
##               RC1         RC2         RC3        RC5        RC4
##  [1,] -0.65518007 -0.22073608  0.06769853  1.4456059 -1.0097101
##  [2,] -0.02127861  0.38713324 -0.16007476  0.9223548 -1.0841590
##  [3,]  3.49638198 -1.39854422 -0.98559082  2.3570623 -0.1715037
##  [4,] -0.93207252  0.44239517  0.42429237  0.8562932  0.2541208
##  [5,] -0.94269044 -1.03354056  0.47719445  1.1980461 -0.2898919
##  [6,] -1.49403303 -0.80722407  0.89922381 -1.2374152  0.5097328
##  [7,]  0.96504199  0.59968597  0.89933201 -2.0314438 -1.0009010
##  [8,]  1.28599736 -0.04754573  2.34553715 -0.5568213  1.7054807
##  [9,]  2.12581923 -1.16585044 -2.64682920  1.3163154  0.2159588
## [10,] -0.51116068 -1.04515453 -0.41897960 -0.1179497  0.5309078
```

```r
CA.df.FAPCc <- cbind(m.FAPCc$scores)
```

Clustering

Model 1 - FA - ML - no rotate

```r
# Calculate Mahalanobis distance to identify multivariate outliers
Maha.FAMLa <- mahalanobis(CA.df.FAMLa, colMeans(CA.df.FAMLa),cov(CA.df.FAMLa))
print(Maha.FAMLa)
```

```
##   [1]  0.9524375  0.3173211 12.8738489  1.0839281  1.4017969  2.8326998
##   [7]  1.4351007 19.5189797 11.9775001  0.5675958  5.6257825  0.5596369
##  [13]  1.8384657  8.2809022  0.2568391  1.7470543  1.1927670  0.9135920
##  [19]  1.2788678  0.2783357  0.9502981  0.8301483  0.2479519  0.8813946
##  [25]  0.7342607  1.8168724  0.6194340  3.6984497  2.5458301  2.1584794
##  [31]  1.1986041  2.0049739  1.0159355  1.3431103  2.0330415  3.1197738
##  [37]  9.9109330  1.3492864 15.3982774 11.4988946  0.6607817  6.5237629
##  [43]  1.7565144  1.5850946  0.6968931  0.5652948  9.9420767  1.5371396
##  [49]  0.6522164 11.3638043 22.1633774  3.1926173 21.1811247  0.2584139
##  [55]  0.3240226  0.3172902  2.5860851 12.8913476  0.8465193  0.8064069
##  [61]  1.5952601  5.4488646  0.2702606  0.7726999  8.8180583  0.8898173
##  [67]  0.6240821  8.3825268  6.3214612  0.8649852 23.0900196  0.9475761
##  [73]  0.3260239  0.1771679  0.5056947  0.9501686  2.6830391  5.9699404
##  [79]  0.2666525  0.7601448  0.3780491  0.4138195  1.3504154  0.5082860
##  [85]  0.7472209  7.1855483  1.0907276  0.7289089  0.6195461  1.2913264
##  [91]  3.8224182  0.8558484  0.1865329  0.2935989  0.4202113  1.9824212
##  [97]  1.6308773  0.2150821  1.1230239  5.2391309  1.7427607  0.1911566
## [103]  0.6697158  0.4910127  1.9054971  1.4652937  1.7300908  0.5596246
## [109]  7.1892422  0.9151959  0.7225537 24.3638430  1.5938386  1.4757665
## [115]  0.6537978  1.1311591  1.3998171  3.6732627  6.7036155  2.2769480
## [121]  0.5931379  1.1154853  2.8422779  0.3722994  1.3794425  0.6011283
## [127]  0.7619071  4.7226072  0.4138485  3.1667208  6.6466713  1.5523832
## [133]  3.5489234  2.7033306  0.8673153 27.3039194  8.2017843  0.7661656
## [139]  2.4149411 29.3137379  0.7757434  0.8211953  0.8201295  2.9355405
## [145]  0.8224609  4.4925682  2.1152786  1.2173813  0.3143061  0.3213460
## [151]  1.2486938  0.9446047  5.5594797  1.0739809  3.5643625 32.6040525
```

```
## [157]  0.9439034  1.2035650  1.8469556  1.4433264  0.3152989  0.8018268
## [163]  6.0452219  4.1853316  0.2800341  5.3452057  0.9467471  1.0952549
## [169]  0.2173916  2.0834161  1.2633181  2.0044284  1.2964767 22.1175870
## [175]  0.1827717  0.8568306  0.1788543  0.3801964  0.4017934  4.1526745
## [181]  0.7898055  1.1291230  1.7501460  0.9490830  2.5233767  5.5218222
## [187]  0.5756720  0.6642002 21.5980246  1.2043098  2.7436730  2.4107786
## [193]  1.7702258  1.8177167  1.0510882  1.5270374  1.6598522  0.1883976
## [199]  0.9252071  5.5361485  4.5269379  0.4492540  0.6546037  0.4401747
## [205]  0.8665666 12.9414431  8.5161698  2.6819207  0.8336180  0.2657393
## [211]  2.7373615  4.1023885  0.1780054  5.4703107  0.3207318  9.4740834
## [217]  0.7273222  1.1760308  9.2050939  0.5147891 17.6727808  0.2526925
## [223]  0.5218078  2.4981201  0.6914766  4.4279341  1.8491023  8.5059998
## [229]  0.4827973  2.1874382  0.4322366 14.0597374  9.9599166  0.1684462
## [235]  2.1545606  2.3148058  0.9798965  1.9430125  1.4514703  2.0202469
## [241]  2.5632037  2.2606130  0.7082913  0.8835760  0.6207199  0.8455983
## [247]  2.7830310 11.5367999  0.3237869  0.8874692  0.8956258  0.5659017
## [253]  0.3555639  1.7874457  2.9736914  0.2877760  1.8269229  4.2869133
## [259]  1.7187167  0.9818200  0.4203262  1.4684180  3.1194592  0.5648997
## [265]  1.8806693  1.4333735  0.1433995  2.3910308  1.8944448  1.2425053
## [271]  0.2369103  0.1030979  1.9219282  0.3720820  0.5120626  0.2971361
## [277]  0.8079651  0.6193630  3.6488046  0.9264997  0.7321090  0.3458954
## [283]  1.0433058  0.3422752  1.2240731  0.6682820  0.4709115  0.5267688
## [289]  7.1812821 19.6399114  2.4786559  0.6699578  1.1638019  1.4647354
## [295]  0.8837238  0.5566248  2.8061752  0.2464435  1.9855051  6.0642974
## [301] 24.8242587  0.2149916  0.6277267  0.4626475  0.5167930  0.3495038
## [307] 11.7696614  0.5443210  4.0667240  3.7649219  0.6793437 15.5634846
## [313]  5.5316354  2.5250101  4.2729729 10.9737800  0.9198886  4.6133586
## [319]  1.1968333  1.5156535  0.3390322  3.7167464  2.3561360  0.3498400
## [325]  2.3432542  0.4578971  2.8158811  2.9836730  1.0590213  0.1951442
## [331]  1.4609888  6.0444617  0.7458461  0.7475374  0.9252699  0.5626256
## [337]  2.7151559  0.2657117  1.9290148  1.0091166  1.4328916  1.0394217
## [343]  0.4223791  1.4346429  0.2690442  0.3464670 10.0248184  6.3378734
## [349] 11.7487934  1.1946834  0.2768540 14.3565974  0.3417048  1.0081487
## [355]  1.7534414  0.1719162  0.2535056  4.1639208  0.5561302  0.5178311
## [361]  1.2800255  0.3706184  1.9198988  4.2012182  0.7473034 16.2922980
## [367]  1.5930876  3.5540961  0.3074061  1.5963047  6.0944312  2.0323676
## [373]  0.6023162  0.2703567  0.3750460  0.3249873  0.9376380  0.8819553
## [379]  2.0883319  0.6799053  6.5687678  0.7616541  0.4391751  0.2546757
## [385]  2.1729095  0.9749048  0.2720885  0.3590680  0.8221060 29.3428096
## [391] 13.7757401  0.8162036  0.9926666  0.1844399  0.4932371  0.2740223
## [397]  1.9854796  0.5093933 11.2823250 10.6769787  0.5741201  2.0170910
## [403] 10.2976792  0.5013148  0.9728970  0.6105507  0.7339667  0.2615994
## [409]  0.9337645  0.3446960  0.3492552  0.3085219  0.8332173  2.7269360
## [415]  2.3964596  0.2918705 10.0678495  0.2224624  0.2489291  1.2859151
## [421]  1.2329574  0.2905848  0.4202431  0.4923442  0.8197894  1.8133539
## [427]  0.2678865  0.6094284  0.7308811  3.9176810 11.0238803  0.8465925
## [433]  1.0817040  1.2586236  0.9711439  0.4634085  1.0179313  2.5154348
## [439]  1.5260983  1.4222009  1.2006463  2.2864282  0.3281610  0.8796222
## [445] 14.0567787  0.3186448  2.8201662  0.9228751  1.3595589  2.5626801
## [451]  0.5524441 13.5207956  4.0467456 32.2201911  1.1736301  3.6873459
## [457]  0.4712675  0.9092761  0.2528948  0.3990146  0.6495496  1.1653588
## [463]  0.7450839  0.2640378  0.5975902  0.2765440  6.5639852  0.9182319
## [469]  1.5142985  1.4350519  0.1096008  0.5190199  0.5893151  0.2384581
## [475]  0.5846148  0.5846080  4.5999915  0.7953013 13.1269541  0.3456125
```

```
## [481] 12.2416109  0.3155409  2.1613958  1.8367786  1.0316164  5.2640539
## [487] 24.8213762  3.7810732  2.5246922  0.5076377  0.2993707  2.1899555
## [493]  0.5354447  0.3199484  0.2610569  2.0372859  5.2785276  0.2009005
## [499]  0.3542274  0.1940355
```

```
Maha.FAMLa.Pv <-pchisq(Maha.FAMLa,df=2,lower.tail = FALSE)
print(Maha.FAMLa.Pv)
```

```
##   [1] 6.211276e-01 8.532859e-01 1.601324e-03 5.816048e-01 4.961393e-01
##   [6] 2.425979e-01 4.879461e-01 5.774407e-05 2.506795e-03 7.529188e-01
##  [11] 6.003118e-02 7.559209e-01 3.988249e-01 1.591567e-02 8.794843e-01
##  [16] 4.174765e-01 5.508000e-01 6.333095e-01 5.275910e-01 8.700820e-01
##  [21] 6.217924e-01 6.602913e-01 8.834011e-01 6.435875e-01 6.927193e-01
##  [26] 4.031542e-01 7.336545e-01 1.573591e-01 2.800142e-01 3.398538e-01
##  [31] 5.491948e-01 3.669657e-01 6.017172e-01 5.109134e-01 3.618517e-01
##  [36] 2.101598e-01 7.044793e-03 5.093381e-01 4.532174e-04 3.184540e-03
##  [41] 7.186428e-01 3.831624e-02 4.155064e-01 4.526902e-01 7.057836e-01
##  [46] 7.537855e-01 6.935942e-03 4.636757e-01 7.217271e-01 3.407071e-03
##  [51] 1.539160e-05 2.026432e-01 2.515227e-05 8.787921e-01 8.504316e-01
##  [56] 8.532991e-01 2.744345e-01 1.587375e-03 6.549086e-01 6.681762e-01
##  [61] 4.503951e-01 6.558342e-02 8.736021e-01 6.795327e-01 1.216698e-02
##  [66] 6.408828e-01 7.319515e-01 1.512716e-02 4.239476e-02 6.488897e-01
##  [71] 9.684249e-06 6.226392e-01 8.495811e-01 9.152263e-01 7.765864e-01
##  [76] 6.218326e-01 2.614481e-01 5.054101e-02 8.751795e-01 6.838119e-01
##  [81] 8.277662e-01 8.130930e-01 5.090507e-01 7.755809e-01 6.882450e-01
##  [86] 2.752187e-02 5.796309e-01 6.945755e-01 7.336134e-01 5.243147e-01
##  [91] 1.479015e-01 6.518608e-01 9.109508e-01 8.634671e-01 8.104986e-01
##  [96] 3.711271e-01 4.424452e-01 8.980396e-01 5.703461e-01 7.283450e-02
## [101] 4.183736e-01 9.088472e-01 7.154397e-01 7.823083e-01 3.856795e-01
## [106] 4.806351e-01 4.210324e-01 7.559256e-01 2.747109e-02 6.328018e-01
## [111] 6.967861e-01 5.122226e-06 4.507153e-01 4.781249e-01 7.211566e-01
## [116] 5.680308e-01 4.966307e-01 1.593533e-01 3.502099e-02 3.203074e-01
## [121] 7.433644e-01 5.725000e-01 2.414389e-01 8.301493e-01 5.017159e-01
## [126] 7.404004e-01 6.832096e-01 9.429722e-02 8.130812e-01 2.052841e-01
## [131] 3.603244e-02 4.601551e-01 1.695747e-01 2.588089e-01 6.481341e-01
## [136] 1.177685e-06 1.655790e-02 6.817564e-01 2.989525e-01 4.311245e-07
## [141] 6.784994e-01 6.632537e-01 6.636073e-01 2.304387e-01 6.628342e-01
## [146] 1.057916e-01 3.472747e-01 5.440628e-01 8.545733e-01 8.515705e-01
## [151] 5.356111e-01 6.235649e-01 6.205465e-02 5.845047e-01 1.682707e-01
## [156] 8.319935e-08 6.237836e-01 5.478342e-01 3.971355e-01 4.859434e-01
## [161] 8.541491e-01 6.697081e-01 4.867397e-02 1.233579e-01 8.693434e-01
## [166] 6.907221e-02 6.228974e-01 5.783203e-01 8.970032e-01 3.528515e-01
## [171] 5.317089e-01 3.670658e-01 5.229663e-01 1.574806e-05 9.126655e-01
## [176] 6.515408e-01 9.144549e-01 8.268779e-01 8.179969e-01 1.253886e-01
## [181] 6.737456e-01 5.686094e-01 4.168316e-01 6.221702e-01 2.831755e-01
## [186] 6.323413e-02 7.498846e-01 7.174155e-01 2.041966e-05 5.476303e-01
## [191] 2.536407e-01 2.995754e-01 4.126676e-01 4.029840e-01 5.912336e-01
## [196] 4.660237e-01 4.360815e-01 9.101018e-01 6.296422e-01 6.278279e-02
## [201] 1.039891e-01 7.988141e-01 7.208661e-01 8.024487e-01 6.483768e-01
## [206] 1.548108e-03 1.414937e-02 2.615943e-01 6.591468e-01 8.755792e-01
## [211] 2.544424e-01 1.285813e-01 9.148431e-01 6.488393e-02 8.518320e-01
## [216] 8.764536e-03 6.951267e-01 5.554285e-01 1.002627e-02 7.730632e-01
## [221] 1.453464e-04 8.813096e-01 7.703550e-01 2.867742e-01 7.076977e-01
## [226] 1.092663e-01 3.967094e-01 1.422151e-02 7.855284e-01 3.349684e-01
```

```
## [231] 8.056400e-01 8.850480e-04 6.874349e-03 9.192262e-01 3.405204e-01
## [236] 3.143014e-01 6.126581e-01 3.785125e-01 4.839687e-01 3.641740e-01
## [241] 2.775923e-01 3.229343e-01 7.017728e-01 6.428859e-01 7.331830e-01
## [246] 6.552102e-01 2.486981e-01 3.124753e-03 8.505318e-01 6.416357e-01
## [251] 6.390242e-01 7.535568e-01 8.371250e-01 4.091298e-01 2.260847e-01
## [256] 8.659847e-01 4.011333e-01 1.172489e-01 4.234337e-01 6.120691e-01
## [261] 8.104520e-01 4.798849e-01 2.101929e-01 7.539345e-01 3.904971e-01
## [266] 4.883677e-01 9.308103e-01 3.025480e-01 3.878167e-01 5.372710e-01
## [271] 8.882916e-01 9.497571e-01 3.825239e-01 8.302396e-01 7.741178e-01
## [276] 8.619414e-01 6.676558e-01 7.336806e-01 1.613140e-01 6.292354e-01
## [281] 6.934650e-01 8.411816e-01 5.935387e-01 8.427056e-01 5.422454e-01
## [286] 7.159528e-01 7.902106e-01 7.684465e-01 2.758064e-02 5.435599e-05
## [291] 2.895788e-01 7.153532e-01 5.588350e-01 4.807693e-01 6.428384e-01
## [296] 7.570603e-01 2.458367e-01 8.840676e-01 3.705553e-01 4.821193e-02
## [301] 4.068935e-06 8.980803e-01 7.306189e-01 7.934825e-01 7.722889e-01
## [306] 8.396653e-01 2.781317e-03 7.617320e-01 1.308947e-01 1.522151e-01
## [311] 7.120039e-01 4.172845e-04 6.292462e-02 2.829443e-01 1.180690e-01
## [316] 4.140702e-03 6.313188e-01 9.959142e-02 5.496813e-01 4.686839e-01
## [321] 8.440732e-01 1.559261e-01 3.078730e-01 8.395242e-01 3.098624e-01
## [326] 7.953694e-01 2.446466e-01 2.249591e-01 5.888931e-01 9.070369e-01
## [331] 4.816708e-01 4.869247e-02 6.887182e-01 6.881360e-01 6.296224e-01
## [336] 7.547922e-01 2.572832e-01 8.755913e-01 3.811709e-01 6.037722e-01
## [341] 4.884853e-01 5.946925e-01 8.096206e-01 4.880578e-01 8.741335e-01
## [346] 8.409412e-01 6.654851e-03 4.204828e-02 2.810489e-03 5.502725e-01
## [351] 8.707268e-01 7.629648e-04 8.429460e-01 6.040645e-01 4.161453e-01
## [356] 9.176327e-01 8.809514e-01 1.246855e-01 7.572475e-01 7.718882e-01
## [361] 5.272857e-01 8.308474e-01 3.829123e-01 1.223819e-01 6.882166e-01
## [366] 2.898494e-04 4.508846e-01 1.691367e-01 8.575266e-01 4.501599e-01
## [371] 4.749097e-02 3.619737e-01 7.399608e-01 8.735601e-01 8.290101e-01
## [376] 8.500215e-01 6.257408e-01 6.434071e-01 3.519853e-01 7.118040e-01
## [381] 3.746366e-02 6.832960e-01 8.028498e-01 8.804362e-01 3.374106e-01
## [386] 6.141891e-01 8.728040e-01 8.356595e-01 6.629518e-01 4.249031e-07
## [391] 1.020084e-03 6.649112e-01 6.087587e-01 9.119046e-01 7.814387e-01
## [396] 8.719605e-01 3.705600e-01 7.751516e-01 3.548741e-03 4.803121e-03
## [401] 7.504667e-01 3.647491e-01 5.806138e-03 7.782890e-01 6.148060e-01
## [406] 7.369204e-01 6.928212e-01 8.773935e-01 6.269539e-01 8.416862e-01
## [411] 8.397697e-01 8.570483e-01 6.592789e-01 2.557722e-01 3.017279e-01
## [416] 8.642137e-01 6.513198e-03 8.947318e-01 8.829696e-01 5.257352e-01
## [421] 5.398420e-01 8.647694e-01 8.104857e-01 7.817877e-01 6.637201e-01
## [426] 4.038641e-01 8.746397e-01 7.373341e-01 6.938909e-01 1.410218e-01
## [431] 4.038265e-03 6.548846e-01 5.822520e-01 5.329584e-01 6.153451e-01
## [436] 7.931807e-01 6.011170e-01 2.843022e-01 4.662426e-01 4.911035e-01
## [441] 5.486343e-01 3.187927e-01 8.486737e-01 6.441581e-01 8.863582e-04
## [446] 8.527214e-01 2.441230e-01 6.303768e-01 5.067288e-01 2.776650e-01
## [451] 7.586445e-01 1.158768e-03 1.322088e-01 1.008032e-07 5.560956e-01
## [456] 1.582352e-01 7.900700e-01 6.346776e-01 8.812205e-01 8.191343e-01
## [461] 7.226901e-01 5.584002e-01 6.889808e-01 8.763244e-01 7.417114e-01
## [466] 8.708618e-01 3.755335e-02 6.318420e-01 4.690015e-01 4.879580e-01
## [471] 9.466741e-01 7.714295e-01 7.447866e-01 8.876045e-01 7.465390e-01
## [476] 7.465416e-01 1.002593e-01 6.718967e-01 1.410971e-03 8.413006e-01
## [481] 2.196686e-03 8.540458e-01 3.393586e-01 3.991614e-01 5.970179e-01
## [486] 7.193251e-02 4.074803e-06 1.509908e-01 2.829893e-01 7.758323e-01
## [491] 8.609788e-01 3.345471e-01 7.651202e-01 8.521658e-01 8.776315e-01
## [496] 3.610846e-01 7.141383e-02 9.044301e-01 8.376845e-01 9.075399e-01
```

```r
print(sum(Maha.FAMLa.Pv<0.001))
```

```
## [1] 22
```

```r
# Remove the outlier where Mahalanobis P-Value less than 0.001 (22)
CA.df.FAMLa <- data.frame(cbind(CA.df.FAMLa, Maha.FAMLa, Maha.FAMLa.Pv))
CA.df.FAMLa <- CA.df.FAMLa %>% filter(Maha.FAMLa.Pv >= 0.001)
```

Model 2 - FA - ML - Orthogonal rotation

```r
# Calculate Mahalanobis distance to identify multivariate outliers
Maha.FAMLb <- mahalanobis(CA.df.FAMLb, colMeans(CA.df.FAMLb),cov(CA.df.FAMLb))
print(Maha.FAMLb)
```

```
##   [1]  0.9524375  0.3173211 12.8738489  1.0839281  1.4017969  2.8326998
##   [7]  1.4351007 19.5189797 11.9775001  0.5675958  5.6257825  0.5596369
##  [13]  1.8384657  8.2809022  0.2568391  1.7470543  1.1927670  0.9135920
##  [19]  1.2788678  0.2783357  0.9502981  0.8301483  0.2479519  0.8813946
##  [25]  0.7342607  1.8168724  0.6194340  3.6984497  2.5458301  2.1584794
##  [31]  1.1986041  2.0049739  1.0159355  1.3431103  2.0330415  3.1197738
##  [37]  9.9109330  1.3492864 15.3982774 11.4988946  0.6607817  6.5237629
##  [43]  1.7565144  1.5850946  0.6968931  0.5652948  9.9420767  1.5371396
##  [49]  0.6522164 11.3638043 22.1633774  3.1926173 21.1811247  0.2584139
##  [55]  0.3240226  0.3172902  2.5860851 12.8913476  0.8465193  0.8064069
##  [61]  1.5952601  5.4488646  0.2702606  0.7726999  8.8180583  0.8898173
##  [67]  0.6240821  8.3825268  6.3214612  0.8649852 23.0900196  0.9475761
##  [73]  0.3260239  0.1771679  0.5056947  0.9501686  2.6830391  5.9699404
##  [79]  0.2666525  0.7601448  0.3780491  0.4138195  1.3504154  0.5082860
##  [85]  0.7472209  7.1855483  1.0907276  0.7289089  0.6195461  1.2913264
##  [91]  3.8224182  0.8558484  0.1865329  0.2935989  0.4202113  1.9824212
##  [97]  1.6308773  0.2150821  1.1230239  5.2391309  1.7427607  0.1911566
## [103]  0.6697158  0.4910127  1.9054971  1.4652937  1.7300908  0.5596246
## [109]  7.1892422  0.9151959  0.7225537 24.3638430  1.5938386  1.4757665
## [115]  0.6537978  1.1311591  1.3998171  3.6732627  6.7036155  2.2769480
## [121]  0.5931379  1.1154853  2.8422779  0.3722994  1.3794425  0.6011283
## [127]  0.7619071  4.7226072  0.4138485  3.1667208  6.6466713  1.5523832
## [133]  3.5489234  2.7033306  0.8673153 27.3039194  8.2017843  0.7661656
## [139]  2.4149411 29.3137379  0.7757434  0.8211953  0.8201295  2.9355405
## [145]  0.8224609  4.4925682  2.1152786  1.2173813  0.3143061  0.3213460
## [151]  1.2486938  0.9446047  5.5594797  1.0739809  3.5643625 32.6040525
## [157]  0.9439034  1.2035650  1.8469556  1.4433264  0.3152989  0.8018268
## [163]  6.0452219  4.1853316  0.2800341  5.3452057  0.9467471  1.0952549
## [169]  0.2173916  2.0834161  1.2633181  2.0044284  1.2964767 22.1175870
## [175]  0.1827717  0.8568306  0.1788543  0.3801964  0.4017934  4.1526745
## [181]  0.7898055  1.1291230  1.7501460  0.9490830  2.5233767  5.5218222
## [187]  0.5756720  0.6642002 21.5980246  1.2043098  2.7436730  2.4107786
## [193]  1.7702258  1.8177167  1.0510882  1.5270374  1.6598522  0.1883976
## [199]  0.9252071  5.5361485  4.5269379  0.4492540  0.6546037  0.4401747
## [205]  0.8665666 12.9414431  8.5161698  2.6819207  0.8336180  0.2657393
## [211]  2.7373615  4.1023885  0.1780054  5.4703107  0.3207318  9.4740834
## [217]  0.7273222  1.1760308  9.2050939  0.5147891 17.6727808  0.2526925
## [223]  0.5218078  2.4981201  0.6914766  4.4279341  1.8491023  8.5059998
```

```
## [229]  0.4827973  2.1874382  0.4322366 14.0597374  9.9599166  0.1684462
## [235]  2.1545606  2.3148058  0.9798965  1.9430125  1.4514703  2.0202469
## [241]  2.5632037  2.2606130  0.7082913  0.8835760  0.6207199  0.8455983
## [247]  2.7830310 11.5367999  0.3237869  0.8874692  0.8956258  0.5659017
## [253]  0.3555639  1.7874457  2.9736914  0.2877760  1.8269229  4.2869133
## [259]  1.7187167  0.9818200  0.4203262  1.4684180  3.1194592  0.5648997
## [265]  1.8806693  1.4333735  0.1433995  2.3910308  1.8944448  1.2425053
## [271]  0.2369103  0.1030979  1.9219282  0.3720820  0.5120626  0.2971361
## [277]  0.8079651  0.6193630  3.6488046  0.9264997  0.7321090  0.3458954
## [283]  1.0433058  0.3422752  1.2240731  0.6682820  0.4709115  0.5267688
## [289]  7.1812821 19.6399114  2.4786559  0.6699578  1.1638019  1.4647354
## [295]  0.8837238  0.5566248  2.8061752  0.2464435  1.9855051  6.0642974
## [301] 24.8242587  0.2149916  0.6277267  0.4626475  0.5167930  0.3495038
## [307] 11.7696614  0.5443210  4.0667240  3.7649219  0.6793437 15.5634846
## [313]  5.5316354  2.5250101  4.2729729 10.9737800  0.9198886  4.6133586
## [319]  1.1968333  1.5156535  0.3390322  3.7167464  2.3561360  0.3498400
## [325]  2.3432542  0.4578971  2.8158811  2.9836730  1.0590213  0.1951442
## [331]  1.4609888  6.0444617  0.7458461  0.7475374  0.9252699  0.5626256
## [337]  2.7151559  0.2657117  1.9290148  1.0091166  1.4328916  1.0394217
## [343]  0.4223791  1.4346429  0.2690442  0.3464670 10.0248184  6.3378734
## [349] 11.7487934  1.1946834  0.2768540 14.3565974  0.3417048  1.0081487
## [355]  1.7534414  0.1719162  0.2535056  4.1639208  0.5561302  0.5178311
## [361]  1.2800255  0.3706184  1.9198988  4.2012182  0.7473034 16.2922980
## [367]  1.5930876  3.5540961  0.3074061  1.5963047  6.0944312  2.0323676
## [373]  0.6023162  0.2703567  0.3750460  0.3249873  0.9376380  0.8819553
## [379]  2.0883319  0.6799053  6.5687678  0.7616541  0.4391751  0.2546757
## [385]  2.1729095  0.9749048  0.2720885  0.3590680  0.8221060 29.3428096
## [391] 13.7757401  0.8162036  0.9926666  0.1844399  0.4932371  0.2740223
## [397]  1.9854796  0.5093933 11.2823250 10.6769787  0.5741201  2.0170910
## [403] 10.2976792  0.5013148  0.9728970  0.6105507  0.7339667  0.2615994
## [409]  0.9337645  0.3446960  0.3492552  0.3085219  0.8332173  2.7269360
## [415]  2.3964596  0.2918705 10.0678495  0.2224624  0.2489291  1.2859151
## [421]  1.2329574  0.2905848  0.4202431  0.4923442  0.8197894  1.8133539
## [427]  0.2678865  0.6094284  0.7308811  3.9176810 11.0238803  0.8465925
## [433]  1.0817040  1.2586236  0.9711439  0.4634085  1.0179313  2.5154348
## [439]  1.5260983  1.4222009  1.2006463  2.2864282  0.3281610  0.8796222
## [445] 14.0567787  0.3186448  2.8201662  0.9228751  1.3595589  2.5626801
## [451]  0.5524441 13.5207956  4.0467456 32.2201911  1.1736301  3.6873459
## [457]  0.4712675  0.9092761  0.2528948  0.3990146  0.6495496  1.1653588
## [463]  0.7450839  0.2640378  0.5975902  0.2765440  6.5639852  0.9182319
## [469]  1.5142985  1.4350519  0.1096008  0.5190199  0.5893151  0.2384581
## [475]  0.5846148  0.5846080  4.5999915  0.7953013 13.1269541  0.3456125
## [481] 12.2416109  0.3155409  2.1613958  1.8367786  1.0316164  5.2640539
## [487] 24.8213762  3.7810732  2.5246922  0.5076377  0.2993707  2.1899555
## [493]  0.5354447  0.3199484  0.2610569  2.0372859  5.2785276  0.2009005
## [499]  0.3542274  0.1940355
```

```
Maha.FAMLb.Pv <-pchisq(Maha.FAMLb,df=2,lower.tail = FALSE)
print(Maha.FAMLb.Pv)
```

```
##   [1] 6.211276e-01 8.532859e-01 1.601324e-03 5.816048e-01 4.961393e-01
##   [6] 2.425979e-01 4.879461e-01 5.774407e-05 2.506795e-03 7.529188e-01
##  [11] 6.003118e-02 7.559209e-01 3.988249e-01 1.591567e-02 8.794843e-01
##  [16] 4.174765e-01 5.508000e-01 6.333095e-01 5.275910e-01 8.700820e-01
```

```
##  [21] 6.217924e-01 6.602913e-01 8.834011e-01 6.435875e-01 6.927193e-01
##  [26] 4.031542e-01 7.336545e-01 1.573591e-01 2.800142e-01 3.398538e-01
##  [31] 5.491948e-01 3.669657e-01 6.017172e-01 5.109134e-01 3.618517e-01
##  [36] 2.101598e-01 7.044793e-03 5.093381e-01 4.532174e-04 3.184540e-03
##  [41] 7.186428e-01 3.831624e-02 4.155064e-01 4.526902e-01 7.057836e-01
##  [46] 7.537855e-01 6.935942e-03 4.636757e-01 7.217271e-01 3.407071e-03
##  [51] 1.539160e-05 2.026432e-01 2.515227e-05 8.787921e-01 8.504316e-01
##  [56] 8.532991e-01 2.744345e-01 1.587375e-03 6.549086e-01 6.681762e-01
##  [61] 4.503951e-01 6.558342e-02 8.736021e-01 6.795327e-01 1.216698e-02
##  [66] 6.408828e-01 7.319515e-01 1.512716e-02 4.239476e-02 6.488897e-01
##  [71] 9.684249e-06 6.226392e-01 8.495811e-01 9.152263e-01 7.765864e-01
##  [76] 6.218326e-01 2.614481e-01 5.054101e-02 8.751795e-01 6.838119e-01
##  [81] 8.277662e-01 8.130930e-01 5.090507e-01 7.755809e-01 6.882450e-01
##  [86] 2.752187e-02 5.796309e-01 6.945755e-01 7.336134e-01 5.243147e-01
##  [91] 1.479015e-01 6.518608e-01 9.109508e-01 8.634671e-01 8.104986e-01
##  [96] 3.711271e-01 4.424452e-01 8.980396e-01 5.703461e-01 7.283450e-02
## [101] 4.183736e-01 9.088472e-01 7.154397e-01 7.823083e-01 3.856795e-01
## [106] 4.806351e-01 4.210324e-01 7.559256e-01 2.747109e-02 6.328018e-01
## [111] 6.967861e-01 5.122226e-06 4.507153e-01 4.781249e-01 7.211566e-01
## [116] 5.680308e-01 4.966307e-01 1.593533e-01 3.502099e-02 3.203074e-01
## [121] 7.433644e-01 5.725000e-01 2.414389e-01 8.301493e-01 5.017159e-01
## [126] 7.404004e-01 6.832096e-01 9.429722e-02 8.130812e-01 2.052841e-01
## [131] 3.603244e-02 4.601551e-01 1.695747e-01 2.588089e-01 6.481341e-01
## [136] 1.177685e-06 1.655790e-02 6.817564e-01 2.989525e-01 4.311245e-07
## [141] 6.784994e-01 6.632537e-01 6.636073e-01 2.304387e-01 6.628342e-01
## [146] 1.057916e-01 3.472747e-01 5.440628e-01 8.545733e-01 8.515705e-01
## [151] 5.356111e-01 6.235649e-01 6.205465e-02 5.845047e-01 1.682707e-01
## [156] 8.319935e-08 6.237836e-01 5.478342e-01 3.971355e-01 4.859434e-01
## [161] 8.541491e-01 6.697081e-01 4.867397e-02 1.233579e-01 8.693434e-01
## [166] 6.907221e-02 6.228974e-01 5.783203e-01 8.970032e-01 3.528515e-01
## [171] 5.317089e-01 3.670658e-01 5.229663e-01 1.574806e-05 9.126655e-01
## [176] 6.515408e-01 9.144549e-01 8.268779e-01 8.179969e-01 1.253886e-01
## [181] 6.737456e-01 5.686094e-01 4.168316e-01 6.221702e-01 2.831755e-01
## [186] 6.323413e-02 7.498846e-01 7.174155e-01 2.041966e-05 5.476303e-01
## [191] 2.536407e-01 2.995754e-01 4.126676e-01 4.029840e-01 5.912336e-01
## [196] 4.660237e-01 4.360815e-01 9.101018e-01 6.296422e-01 6.278279e-02
## [201] 1.039891e-01 7.988141e-01 7.208661e-01 8.024487e-01 6.483768e-01
## [206] 1.548108e-03 1.414937e-02 2.615943e-01 6.591468e-01 8.755792e-01
## [211] 2.544424e-01 1.285813e-01 9.148431e-01 6.488393e-02 8.518320e-01
## [216] 8.764536e-03 6.951267e-01 5.554285e-01 1.002627e-02 7.730632e-01
## [221] 1.453464e-04 8.813096e-01 7.703550e-01 2.867742e-01 7.076977e-01
## [226] 1.092663e-01 3.967094e-01 1.422151e-02 7.855284e-01 3.349684e-01
## [231] 8.056400e-01 8.850480e-04 6.874349e-03 9.192262e-01 3.405204e-01
## [236] 3.143014e-01 6.126581e-01 3.785125e-01 4.839687e-01 3.641740e-01
## [241] 2.775923e-01 3.229343e-01 7.017728e-01 6.428859e-01 7.331830e-01
## [246] 6.552102e-01 2.486981e-01 3.124753e-03 8.505318e-01 6.416357e-01
## [251] 6.390242e-01 7.535568e-01 8.371250e-01 4.091298e-01 2.260847e-01
## [256] 8.659847e-01 4.011333e-01 1.172489e-01 4.234337e-01 6.120691e-01
## [261] 8.104520e-01 4.798849e-01 2.101929e-01 7.539345e-01 3.904971e-01
## [266] 4.883677e-01 9.308103e-01 3.025480e-01 3.878167e-01 5.372710e-01
## [271] 8.882916e-01 9.497571e-01 3.825239e-01 8.302396e-01 7.741178e-01
## [276] 8.619414e-01 6.676558e-01 7.336806e-01 1.613140e-01 6.292354e-01
## [281] 6.934650e-01 8.411816e-01 5.935387e-01 8.427056e-01 5.422454e-01
## [286] 7.159528e-01 7.902106e-01 7.684465e-01 2.758064e-02 5.435599e-05
```

```
## [291] 2.895788e-01 7.153532e-01 5.588350e-01 4.807693e-01 6.428384e-01
## [296] 7.570603e-01 2.458367e-01 8.840676e-01 3.705553e-01 4.821193e-02
## [301] 4.068935e-06 8.980803e-01 7.306189e-01 7.934825e-01 7.722889e-01
## [306] 8.396653e-01 2.781317e-03 7.617320e-01 1.308947e-01 1.522151e-01
## [311] 7.120039e-01 4.172845e-04 6.292462e-02 2.829443e-01 1.180690e-01
## [316] 4.140702e-03 6.313188e-01 9.959142e-02 5.496813e-01 4.686839e-01
## [321] 8.440732e-01 1.559261e-01 3.078730e-01 8.395242e-01 3.098624e-01
## [326] 7.953694e-01 2.446466e-01 2.249591e-01 5.888931e-01 9.070369e-01
## [331] 4.816708e-01 4.869247e-02 6.887182e-01 6.881360e-01 6.296224e-01
## [336] 7.547922e-01 2.572832e-01 8.755913e-01 3.811709e-01 6.037722e-01
## [341] 4.884853e-01 5.946925e-01 8.096206e-01 4.880578e-01 8.741335e-01
## [346] 8.409412e-01 6.654851e-03 4.204828e-02 2.810489e-03 5.502725e-01
## [351] 8.707268e-01 7.629648e-04 8.429460e-01 6.040645e-01 4.161453e-01
## [356] 9.176327e-01 8.809514e-01 1.246855e-01 7.572475e-01 7.718882e-01
## [361] 5.272857e-01 8.308474e-01 3.829123e-01 1.223819e-01 6.882166e-01
## [366] 2.898494e-04 4.508846e-01 1.691367e-01 8.575266e-01 4.501599e-01
## [371] 4.749097e-02 3.619737e-01 7.399608e-01 8.735601e-01 8.290101e-01
## [376] 8.500215e-01 6.257408e-01 6.434071e-01 3.519853e-01 7.118040e-01
## [381] 3.746366e-02 6.832960e-01 8.028498e-01 8.804362e-01 3.374106e-01
## [386] 6.141891e-01 8.728040e-01 8.356595e-01 6.629518e-01 4.249031e-07
## [391] 1.020084e-03 6.649112e-01 6.087587e-01 9.119046e-01 7.814387e-01
## [396] 8.719605e-01 3.705600e-01 7.751516e-01 3.548741e-03 4.803121e-03
## [401] 7.504667e-01 3.647491e-01 5.806138e-03 7.782890e-01 6.148060e-01
## [406] 7.369204e-01 6.928212e-01 8.773935e-01 6.269539e-01 8.416862e-01
## [411] 8.397697e-01 8.570483e-01 6.592789e-01 2.557722e-01 3.017279e-01
## [416] 8.642137e-01 6.513198e-03 8.947318e-01 8.829696e-01 5.257352e-01
## [421] 5.398420e-01 8.647694e-01 8.104857e-01 7.817877e-01 6.637201e-01
## [426] 4.038641e-01 8.746397e-01 7.373341e-01 6.938909e-01 1.410218e-01
## [431] 4.038265e-03 6.548846e-01 5.822520e-01 5.329584e-01 6.153451e-01
## [436] 7.931807e-01 6.011170e-01 2.843022e-01 4.662426e-01 4.911035e-01
## [441] 5.486343e-01 3.187927e-01 8.486737e-01 6.441581e-01 8.863582e-04
## [446] 8.527214e-01 2.441230e-01 6.303768e-01 5.067288e-01 2.776650e-01
## [451] 7.586445e-01 1.158768e-03 1.322088e-01 1.008032e-07 5.560956e-01
## [456] 1.582352e-01 7.900700e-01 6.346776e-01 8.812205e-01 8.191343e-01
## [461] 7.226901e-01 5.584002e-01 6.889808e-01 8.763244e-01 7.417114e-01
## [466] 8.708618e-01 3.755335e-02 6.318420e-01 4.690015e-01 4.879580e-01
## [471] 9.466741e-01 7.714295e-01 7.447866e-01 8.876045e-01 7.465390e-01
## [476] 7.465416e-01 1.002593e-01 6.718967e-01 1.410971e-03 8.413006e-01
## [481] 2.196686e-03 8.540458e-01 3.393586e-01 3.991614e-01 5.970179e-01
## [486] 7.193251e-02 4.074803e-06 1.509908e-01 2.829893e-01 7.758323e-01
## [491] 8.609788e-01 3.345471e-01 7.651202e-01 8.521658e-01 8.776315e-01
## [496] 3.610846e-01 7.141383e-02 9.044301e-01 8.376845e-01 9.075399e-01
```

```
print(sum(Maha.FAMLb.Pv<0.001))
```

```
## [1] 22
```

```
# Remove the outlier where Mahalanobis P-Value less than 0.001 (22)
original.df <- data.frame(cbind(CA.df.FAMLb, Maha.FAMLb, Maha.FAMLb.Pv))
original.df$index <- 1:nrow(original.df)
original.df <- original.df %>% filter(Maha.FAMLb.Pv >= 0.001)
```

Model 3 - FA - PC - Orthogonal rotation

```r
# Calculate Mahalanobis distance to identify multivariate outliers
Maha.FAPCc <- mahalanobis(CA.df.FAPCc, colMeans(CA.df.FAPCc),cov(CA.df.FAPCc))
print(Maha.FAPCc)
```

```
##   [1]  3.5918592  2.2020880 20.7371584  2.0423122  3.7039378  5.4833729
##   [7]  7.2282942 10.3763086 14.6633440  1.8249524 11.2913482  1.2466275
##  [13]  2.2473907  6.7455680  1.7276821  6.4844972  5.6298778  1.6934014
##  [19] 11.5446195  0.6305995  3.5909954  1.4645944  4.0253764  7.1487228
##  [25]  2.7671573  3.7725008  6.9208317  3.7827058  3.4832005  6.3717505
##  [31]  2.6477206  6.1737291  7.5832261  4.8526883  3.1783262  2.7523342
##  [37]  4.0738109  1.8310492 12.9542666  2.8304931  2.9574236  3.5116078
##  [43]  3.7807712  2.6326500  3.8336716  6.0922454  5.3699913  3.0411625
##  [49]  1.4178396  9.8234716  8.5816950  6.5138379  7.1595786  0.4154837
##  [55]  2.6279181  1.2107709  1.7753760 24.0543108  2.0034775  1.4049686
##  [61]  1.6810191  9.9070870  0.7604935  2.1749284 13.4467053  1.4278727
##  [67]  3.8912135  8.9510766  7.3036433  4.3155981 10.2949934  5.3723013
##  [73]  4.8436322  5.0501086  2.9121250  2.7499141  3.9613386  3.0969205
##  [79]  1.1029412  2.2556376  7.8030382  2.5963684  2.3766610  5.3264017
##  [85]  4.4725367  3.9398322  8.0155001  1.0345671  3.5575466  7.7682834
##  [91]  4.7835511  5.5301322  1.3199377  1.0323230  3.4484999  1.5936396
##  [97]  2.1039354  2.6900060  7.9813794  9.3598062  5.9159982  0.6485131
## [103]  2.5004309  4.6685567  4.4224362 14.4624471  5.4025582  1.4307830
## [109] 13.2782960  4.9959541  3.6441607  9.8329047  2.3063730  4.1377931
## [115]  0.9777952  1.0623071  3.1508338  2.9366755  4.9908255  3.9410914
## [121]  0.7437403  4.9472211  5.9785709  1.2699174  3.2399954  2.4740486
## [127]  4.3052666  6.1161684  1.5577534  2.9839822  9.8428662  5.2615876
## [133]  8.9381316 28.0979026  1.4136030 36.7132371  2.3617168  1.6498332
## [139]  7.7454289 18.1826504  4.6965914  6.6676495  1.1126489 10.6412719
## [145]  5.0920313  5.3700000  3.3628187  3.6690605  4.2033863  3.1672355
## [151]  7.6229194  1.5406790  5.8913671  7.7923248  8.7200656 10.5364802
## [157]  2.5214527  2.0171223 10.7993215  2.3554853  4.2893883  2.0603912
## [163]  6.9535782 16.0379485  1.1292346 10.2984143  2.4673199  4.5877254
## [169] 11.8934477  7.0680064  1.3112252  3.7226674  2.8886009 42.1396232
## [175]  2.3841865  3.5975059  3.7889817 11.6537118  1.2153853 11.6228191
## [181]  1.9311429  3.5542097  2.8443535  9.3135855  8.0159961  5.8456467
## [187]  0.7322737  6.7315697  0.8805711  5.1059473  6.1690298  1.6415538
## [193]  8.8835284 19.2392557  1.2503865 14.2205089  4.0226134  2.5221701
## [199]  2.5696999  2.3120614  3.3689803  3.9513783  3.1367841  2.7261402
## [205]  1.5857412 10.0127213  3.3890334  4.3421334  3.1305492  1.0497164
## [211]  6.5713241  6.3757676  4.8587748  3.3306790  1.4327947  4.6666000
## [217]  3.3301740  1.9038080 15.2220579  0.8673731  8.4627077  6.7937908
## [223]  1.7148541  7.6019915  3.7394817  6.6918410 12.4603458  1.1186023
## [229]  0.7550143 14.0953686  1.2452493  4.3947261  5.7187228  3.9026374
## [235]  1.9772995  4.0474386  1.8808818  5.4972816  1.3618163  3.6112954
## [241]  1.8986767  5.9543741  3.6509384  2.9396625  2.6170900  3.0903505
## [247]  4.3222171  4.1547296  0.5340527  5.6742813  7.4100936  5.6332621
## [253]  1.4973655 10.7885471  2.2235225  3.5602569  5.7408748  2.6393648
## [259]  4.4493712  5.2046150  3.9135652  5.2414612  7.3887145  3.2167309
## [265]  3.3362748  5.0881232  4.1874073  3.3872340  1.7924936  4.7394070
## [271]  5.0226541  0.3473088  4.3063257  4.1147081  2.8768615  8.9739178
## [277]  5.6111911  3.4333213  6.6701996  2.2229182  1.7668847  1.4819678
## [283]  3.0706078  1.8848609  4.4563646  2.2404020  1.5265028  4.6248940
## [289]  8.7759294  7.2655393  1.6291171  1.8571386  6.8595368  5.8905074
```

```
## [295]  2.0548555  2.5409737  3.1226418  2.3664751  5.4946787  7.1693324
## [301]  7.4343000  2.1040694  4.4842925  8.1067282  3.6880166  2.0964147
## [307] 11.2755625  1.6100872  3.0556224  5.0723947  5.0011908  5.5745817
## [313]  8.0324280  4.0293252 13.9159713 11.5991647  3.5444954  6.7821170
## [319]  5.4815597  2.9039601  4.8282060  4.9376586  3.9465822  1.5975254
## [325]  4.2393600  4.0628573  6.8630281  2.2940681  4.1475716  0.8787595
## [331]  4.6392203  8.2590520  1.4059850  3.9472219  3.6896443  5.2039604
## [337]  5.2519472  1.0477338  2.9861754  2.3972883  4.8862854  8.5194389
## [343]  7.6459252  2.9334801  2.9025834  1.1808645  3.7603666  4.4177041
## [349]  6.4229702  1.2592576  0.9876819  3.5034653  2.8471061  1.9812631
## [355]  6.4642164  2.3480617  2.6986778  2.9495096  4.4803350  4.1431981
## [361]  1.9234857  0.5632272  5.5109100 10.7840189  8.0884202  4.5651906
## [367]  8.4915068  8.5004910  8.8684618  2.1038826  9.5172026  3.6047908
## [373]  8.6045608  4.4928671  2.5710827  3.1044845  2.6529729  1.1808630
## [379]  3.8636643  0.9577731  4.8977791  1.8429585  2.5878390  1.5497349
## [385]  6.9849738 10.0862255  1.4401477  6.0553726  3.2246109  3.0995373
## [391]  7.6609465  3.6832833  1.3959393  5.2451043  2.5920365  5.2055562
## [397]  3.3058822  1.5630315 29.5100832 27.9494857  0.3649821  3.6924516
## [403]  8.0688155  2.0967501  1.4220194  1.2516342  1.8348304  0.9952504
## [409]  3.7936188  2.6711976  1.9954450  2.5171909  4.0778689  2.8961025
## [415]  6.4817571  3.8194372  3.3619521  5.5332869  0.9174744  4.3560502
## [421]  1.6876011  1.0401261  1.8207718  2.4219838  2.5616120  1.5924500
## [427] 22.3812101  3.9961193  2.7865667  5.1401282 10.4087987  6.7239390
## [433]  1.3452750  4.4331459  1.8111770  3.7260331  5.7457590  3.9709176
## [439]  1.6173260  2.5409756  3.0658796  4.0550858  0.3081362  2.6801042
## [445] 16.6727189  4.8532106  8.0429806  7.3970510  4.8330019  4.0084052
## [451]  7.8580445 15.3152599  8.8370179 12.1825331  4.8395303  2.1718443
## [457]  3.1801952  3.6461469  3.9956288  0.6166205  2.0908604  4.3083968
## [463] 16.6781237  1.9948130  1.4438323  4.6549818  3.8367277  3.4184020
## [469]  2.5328507  2.6684151  0.8875883  2.9658017  3.9227650  3.4574959
## [475]  2.3166348  3.7753460 12.4874720  1.9251825 10.9487654  4.3697099
## [481] 12.6129033  3.1446112  4.2144121  2.2211558  7.7455002  7.1712354
## [487] 10.5064711  8.0267543  5.8979695  6.7199432  4.0051797  4.6994136
## [493]  1.1725438  2.1055612  4.5705986  3.9593192  5.3090805  3.1276696
## [499]  2.3693614  0.4912698
```

```
Maha.FAPCc.Pv <-pchisq(Maha.FAPCc,df=2,lower.tail = FALSE)
print(Maha.FAPCc.Pv)
```

```
##   [1] 1.659731e-01 3.325238e-01 3.140388e-05 3.601783e-01 1.569279e-01
##   [6] 6.446154e-02 2.693989e-02 5.582301e-03 6.544784e-04 4.015287e-01
##  [11] 3.532766e-03 5.361648e-01 3.250763e-01 3.429403e-02 4.215398e-01
##  [16] 3.907593e-02 5.990838e-02 4.288274e-01 3.112560e-03 7.295702e-01
##  [21] 1.660448e-01 4.808032e-01 1.336290e-01 2.803332e-02 2.506798e-01
##  [26] 1.516393e-01 3.141669e-02 1.508676e-01 1.752398e-01 4.134205e-02
##  [31] 2.661061e-01 4.564485e-02 2.255918e-02 8.835927e-02 2.040963e-01
##  [36] 2.525447e-01 1.304317e-01 4.003066e-01 1.538214e-03 2.428657e-01
##  [41] 2.279311e-01 1.727683e-01 1.510136e-01 2.681188e-01 1.470716e-01
##  [46] 4.754291e-02 6.822149e-02 2.185848e-01 4.921756e-01 7.359702e-03
##  [51] 1.369332e-02 3.850686e-02 2.788157e-02 8.124167e-01 2.687539e-01
##  [56] 5.458640e-01 4.116063e-01 5.979609e-06 3.672403e-01 4.953532e-01
##  [61] 4.314906e-01 7.058353e-03 6.836927e-01 3.370701e-01 1.202500e-03
##  [66] 4.897127e-01 1.429005e-01 1.138409e-02 2.594383e-02 1.155792e-01
##  [71] 5.813940e-03 6.814274e-02 8.876027e-02 8.005397e-02 2.331525e-01
```

```
##  [76] 2.528504e-01 1.379769e-01 2.125750e-01 5.761020e-01 3.237386e-01
##  [81] 2.021119e-02 2.730271e-01 3.047296e-01 6.972469e-02 1.068565e-01
##  [86] 1.394686e-01 1.817424e-02 5.961377e-01 1.688451e-01 2.056547e-02
##  [91] 9.146714e-02 6.297194e-02 5.168674e-01 5.968070e-01 1.783067e-01
##  [96] 4.507602e-01 3.492499e-01 2.605389e-01 1.848696e-02 9.279913e-03
## [101] 5.192271e-02 7.230647e-01 2.864431e-01 9.688037e-02 1.095671e-01
## [106] 7.236349e-04 6.711961e-02 4.890006e-01 1.308141e-03 8.225122e-02
## [111] 1.616890e-01 7.325071e-03 3.156294e-01 1.263251e-01 6.133021e-01
## [116] 5.879264e-01 2.069213e-01 2.303080e-01 8.246241e-02 1.393808e-01
## [121] 6.894437e-01 8.428001e-02 5.032338e-02 5.299574e-01 1.978992e-01
## [126] 2.902466e-01 1.161778e-01 4.697761e-02 4.589212e-01 2.249244e-01
## [131] 7.288678e-03 7.202127e-02 1.145801e-02 7.918045e-07 4.932192e-01
## [136] 1.066159e-08 3.070151e-01 4.382715e-01 2.080183e-02 1.126387e-04
## [141] 9.553184e-02 3.565647e-02 5.733124e-01 4.889643e-03 7.839339e-02
## [146] 6.822119e-02 1.861115e-01 1.596885e-01 1.222493e-01 2.052313e-01
## [151] 2.211587e-02 4.628559e-01 5.256611e-02 2.031974e-02 1.277797e-02
## [156] 5.152671e-03 2.834481e-01 3.647434e-01 4.518113e-03 3.079732e-01
## [161] 1.171038e-01 3.569371e-01 3.090649e-02 3.291575e-04 5.685777e-01
## [166] 5.804005e-03 2.912248e-01 1.008761e-01 2.614392e-03 2.918784e-02
## [171] 5.191239e-01 1.554651e-01 2.359110e-01 7.071265e-10 3.035851e-01
## [176] 1.655052e-01 1.503949e-01 2.947329e-03 5.446060e-01 2.993208e-03
## [181] 3.807655e-01 1.691271e-01 2.411884e-01 9.496873e-03 1.816973e-02
## [186] 5.378163e-02 6.934079e-01 3.453490e-02 6.438525e-01 7.784982e-02
## [191] 4.575222e-02 4.400896e-01 1.177515e-02 6.641233e-05 5.351580e-01
## [196] 8.166871e-04 1.338137e-01 2.833464e-01 2.766921e-01 3.147330e-01
## [201] 1.855390e-01 1.386657e-01 2.083800e-01 2.558740e-01 4.525439e-01
## [206] 6.695225e-03 1.836880e-01 1.140559e-01 2.090306e-01 5.916392e-01
## [211] 3.741581e-02 4.125909e-02 8.809078e-02 1.891264e-01 4.885090e-01
## [216] 9.697520e-02 1.891742e-01 3.860054e-01 4.949623e-04 6.481154e-01
## [221] 1.453270e-02 3.347704e-02 4.242523e-01 2.234851e-02 1.541636e-01
## [226] 3.522777e-02 1.969111e-03 5.716084e-01 6.855683e-01 8.694199e-04
## [231] 5.365344e-01 1.110957e-01 5.730534e-02 1.420866e-01 3.720787e-01
## [236] 1.321630e-01 3.904556e-01 6.401481e-02 5.061571e-01 1.643680e-01
## [241] 3.869970e-01 5.093591e-02 1.611420e-01 2.299643e-01 2.702129e-01
## [246] 2.132745e-01 1.151973e-01 1.252599e-01 7.656529e-01 5.859296e-02
## [251] 2.459907e-02 5.980709e-02 4.729892e-01 4.542519e-03 3.289790e-01
## [256] 1.686165e-01 5.667413e-02 2.672202e-01 1.081014e-01 7.410239e-02
## [261] 1.413123e-01 7.274969e-02 2.486343e-02 2.002146e-01 1.885980e-01
## [266] 7.854672e-02 1.232299e-01 1.838533e-01 4.080985e-01 9.350845e-02
## [271] 8.116046e-02 8.405873e-01 1.161163e-01 1.277917e-01 2.372998e-01
## [276] 1.125482e-02 6.047075e-02 1.796651e-01 3.561103e-02 3.290785e-01
## [281] 4.133575e-01 4.766447e-01 2.153902e-01 3.896796e-01 1.077241e-01
## [286] 3.262142e-01 4.661483e-01 9.901866e-02 1.242599e-02 2.644285e-02
## [291] 4.428348e-01 3.951186e-01 3.239444e-02 5.258872e-02 3.579265e-01
## [296] 2.806949e-01 2.098587e-01 3.062855e-01 6.409818e-02 2.774593e-02
## [301] 2.430313e-02 3.492265e-01 1.062303e-01 1.736386e-02 1.581821e-01
## [306] 3.505656e-01 3.560760e-03 4.470684e-01 2.170101e-01 7.916687e-02
## [311] 8.203614e-02 6.158784e-02 1.802106e-02 1.333654e-01 9.510103e-04
## [316] 3.028819e-03 1.699506e-01 3.367302e-02 6.452001e-02 2.341063e-01
## [321] 8.944754e-02 8.468394e-02 1.389986e-01 4.498853e-01 1.200700e-01
## [326] 1.311480e-01 3.233794e-02 3.175773e-01 1.257090e-01 6.444360e-01
## [331] 9.831191e-02 1.609050e-02 4.951015e-01 1.389542e-01 1.580534e-01
## [336] 7.412665e-02 7.236926e-02 5.922260e-01 2.246778e-01 3.016029e-01
## [341] 8.688736e-02 1.412626e-02 2.186293e-02 2.306763e-01 2.342675e-01
```

```
## [346] 5.540877e-01 1.525621e-01 1.098266e-01 4.029672e-02 5.327895e-01
## [351] 6.102778e-01 1.734731e-01 2.408567e-01 3.713421e-01 3.947419e-02
## [356] 3.091184e-01 2.594117e-01 2.288348e-01 1.064407e-01 1.259842e-01
## [361] 3.822261e-01 7.545652e-01 6.358008e-02 4.552815e-03 1.752354e-02
## [366] 1.020191e-01 1.432494e-02 1.426073e-02 1.186419e-02 3.492591e-01
## [371] 8.577598e-03 1.649034e-01 1.353765e-02 1.057758e-01 2.765009e-01
## [376] 2.117726e-01 2.654082e-01 5.540882e-01 1.448825e-01 6.194728e-01
## [381] 8.638946e-02 3.979300e-01 2.741940e-01 4.607649e-01 3.042511e-02
## [386] 6.453629e-03 4.867163e-01 4.842755e-02 1.994273e-01 2.122971e-01
## [391] 2.169934e-02 1.585569e-01 4.975946e-01 7.261730e-02 2.736191e-01
## [396] 7.406752e-02 1.914859e-01 4.577117e-01 3.908111e-07 8.527982e-07
## [401] 8.331921e-01 1.578317e-01 1.769616e-02 3.505068e-01 4.911480e-01
## [406] 5.348242e-01 3.995505e-01 6.079728e-01 1.500466e-01 2.630006e-01
## [411] 3.687182e-01 2.840527e-01 1.301673e-01 2.350278e-01 3.912950e-02
## [416] 1.481221e-01 1.861922e-01 6.287269e-02 6.320813e-01 1.132650e-01
## [421] 4.300729e-01 5.944831e-01 4.023689e-01 2.979016e-01 2.778133e-01
## [426] 4.510284e-01 1.380327e-05 1.355981e-01 2.482588e-01 7.653064e-02
## [431] 5.492348e-03 3.466691e-02 5.103607e-01 1.089820e-01 4.043039e-01
## [436] 1.552037e-01 5.653590e-02 1.373176e-01 4.454532e-01 2.806947e-01
## [441] 2.159000e-01 1.316586e-01 8.572136e-01 2.618320e-01 2.396432e-04
## [446] 8.833620e-02 1.792623e-02 2.476001e-02 8.923330e-02 1.347677e-01
## [451] 1.966289e-02 4.724258e-04 1.205219e-02 2.262542e-03 8.894250e-02
## [456] 3.375903e-01 2.039057e-01 1.615285e-01 1.356314e-01 7.346873e-01
## [461] 3.515406e-01 1.159961e-01 2.389964e-04 3.688348e-01 4.858205e-01
## [466] 9.754018e-02 1.468470e-01 1.810104e-01 2.818373e-01 2.633668e-01
## [471] 6.415975e-01 2.269783e-01 1.406638e-01 1.775065e-01 3.140141e-01
## [476] 1.514238e-01 1.942584e-03 3.819020e-01 4.192816e-03 1.124941e-01
## [481] 1.824496e-03 2.075661e-01 1.215772e-01 3.293686e-01 2.080109e-02
## [486] 2.771954e-02 5.230567e-03 1.807226e-02 5.239287e-02 3.473625e-02
## [491] 1.349852e-01 9.539713e-02 5.563977e-01 3.489661e-01 1.017436e-01
## [496] 1.381162e-01 7.033117e-02 2.093318e-01 3.058438e-01 7.822078e-01
```

```
print(sum(Maha.FAPCc.Pv<0.001))
```

```
## [1] 20
```

```
# Remove the outlier where Mahalanobis P-Value less than 0.001 (20)
CA.df.FAPCc <- data.frame(cbind(CA.df.FAPCc, Maha.FAPCc, Maha.FAPCc.Pv))
CA.df.FAPCc <- CA.df.FAPCc %>% filter(Maha.FAPCc.Pv >= 0.001)
```

Check assumption for Clustering

```
# Drop column Mahalanobis distance and P-Value
CA.df.FAMLa = subset(CA.df.FAMLa, select = -c(Maha.FAMLa, Maha.FAMLa.Pv) )
CA.df.FAMLb = subset(original.df, select = -c(Maha.FAMLb, Maha.FAMLb.Pv, index) )
CA.df.FAPCc = subset(CA.df.FAPCc, select = -c(Maha.FAPCc, Maha.FAPCc.Pv) )

# Check Correlation Matrix
lowerCor(CA.df.FAMLa)
```

```
##      ML1   ML2   ML3
```

```
## ML1  1.00
## ML2 -0.20  1.00
## ML3  0.01 -0.04  1.00
```

```
lowerCor(CA.df.FAMLb)
```

```
##     ML1   ML2   ML3
## ML1  1.00
## ML2 -0.05  1.00
## ML3  0.01 -0.04  1.00
```

```
lowerCor(CA.df.FAPCc)
```

```
##     RC1   RC2   RC3   RC5   RC4
## RC1  1.00
## RC2 -0.01  1.00
## RC3  0.04  0.01  1.00
## RC5  0.01  0.12  0.05  1.00
## RC4 -0.03  0.06  0.05 -0.05  1.00
```

```
# Because we do cluster on PC/Factor, so the components is now shown high correlation between each vari
```

Clustering

Model A

```
# Define function to calculate agglomerative coefficient
m <- c("average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")
```

Model A

```
ac.a <- function(x) {
  agnes(CA.df.FAMLa, method = x)$ac
}
```

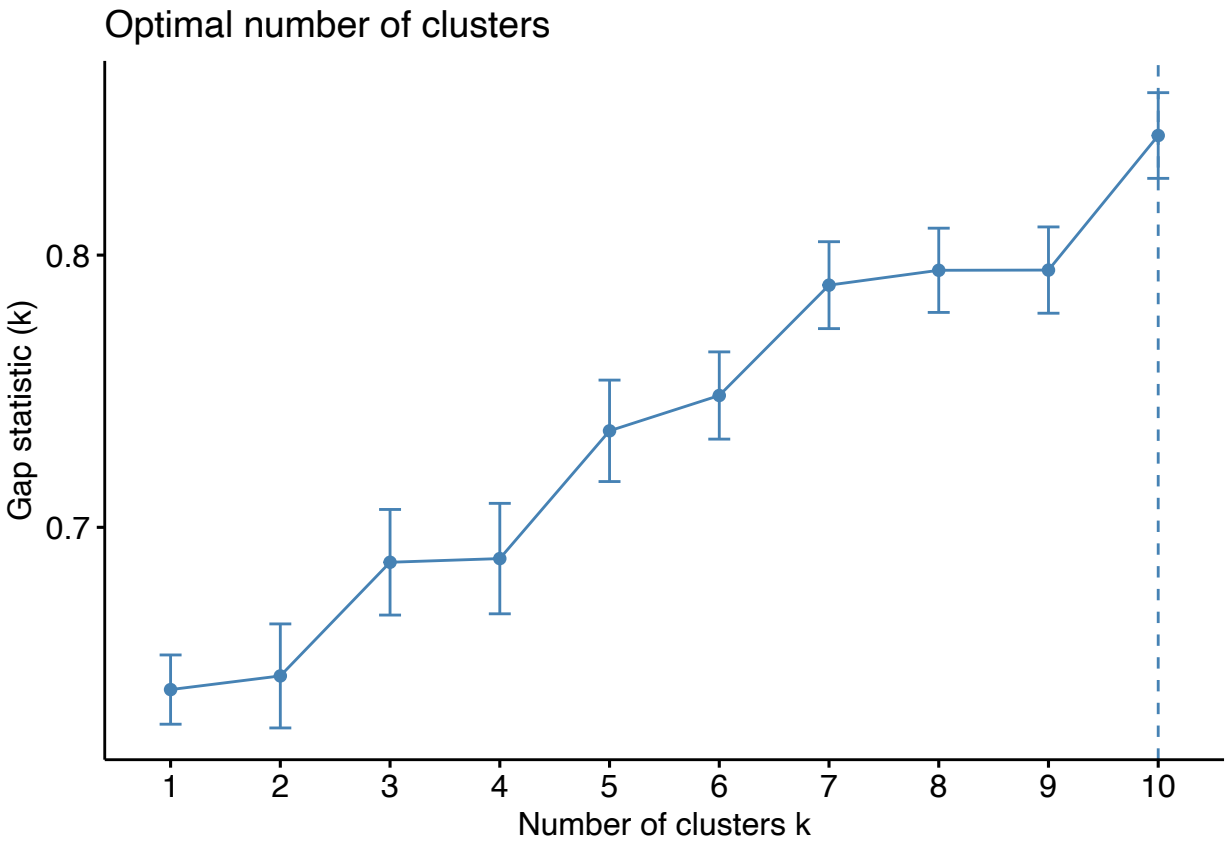
Calculate agglomerative coefficient

```
sapply(m, ac.a)
```

```
##   average    single  complete      ward
## 0.9534457 0.9358105 0.9777409 0.9935047
```

calculate gap statistic for each number of clusters (up to 10 clusters)

```
gap.h.a <- clusGap(CA.df.FAMLa, FUN = hcut, nstart = 25, K.max = 10, B = 50)
fviz_gap_stat(gap.h.a)
```



For Model A (FA - ML - no rotate), the agglomerative coefficient show that ward's distance method is the most suitable for hierarchical clustering, and the recommend number of K is 3 clusters (first peak)

Finding distance matrix

```
distance_mat.a <- dist(CA.df.FAMLa, method = 'euclidean')
```

Fitting Hierarchical clustering Model to dataset

```
set.seed(240)  # Setting seed
Hierar_cl.a <- hclust(distance_mat.a, method = "ward")
```

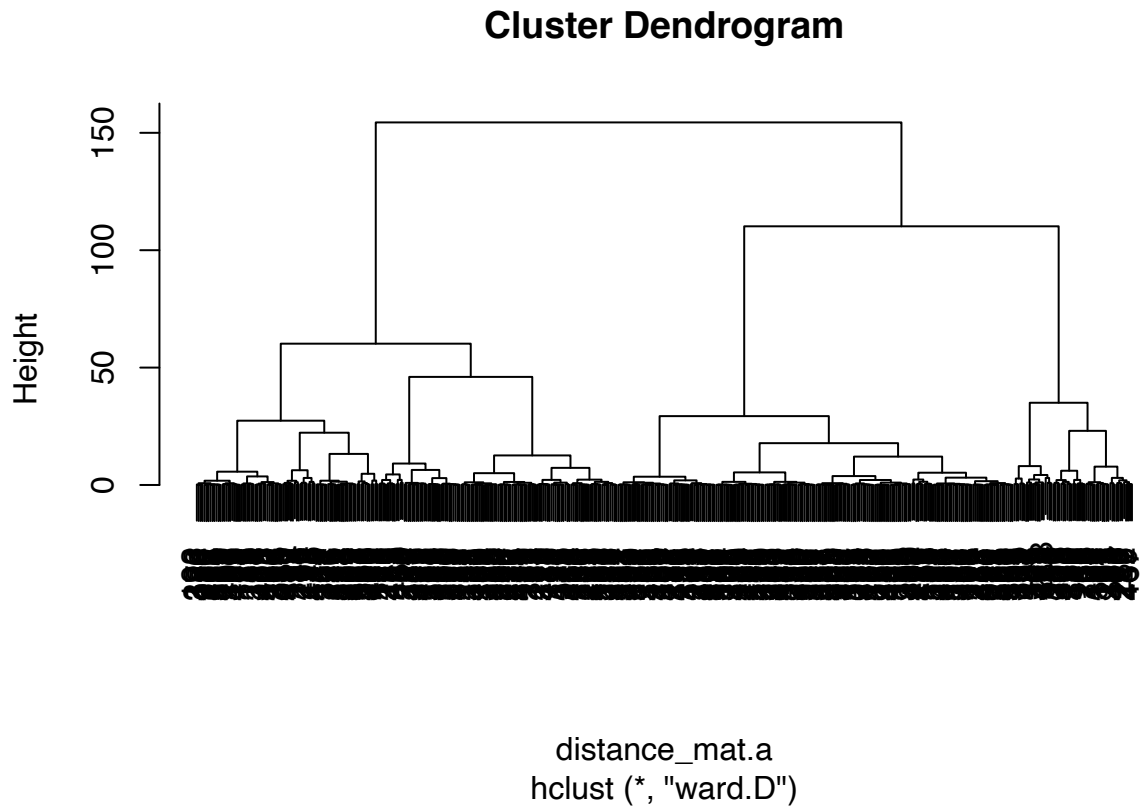
```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```
Hierar_cl.a
```

```
##
## Call:
## hclust(d = distance_mat.a, method = "ward")
##
## Cluster method   : ward.D
## Distance         : euclidean
## Number of objects: 478
```

Plotting dendrogram

```r
plot(Hierar_cl.a)
```



Choosing no. of clusters

Cutting tree by no. of clusters

```r
CA.a.fit_3 <- cutree(Hierar_cl.a, k = 3)
```

Find number of observations in each cluster

```r
table(CA.a.fit_3)
```

```
## CA.a.fit_3
##   1   2   3
## 202 215  61
```

```r
CA.a.final_data_3 <-cbind(CA.df.FAMLa, cluster = CA.a.fit_3)
```

Display first six rows of final data

```r
head(CA.a.final_data_3)
```

```
##          ML1         ML2          ML3 cluster
## 1 -0.8715353 -0.43238104  0.054168820       1
```

```
## 2 -0.1582354 -0.53806205  0.004499027       2
## 3  2.5430851 -2.27416461  1.064604585       2
## 4 -0.8571602  0.49326358  0.314256408       3
## 5 -1.1056402  0.06241754 -0.408938769       1
## 6 -1.5103006  0.36649659 -0.630515157       1
```

Find mean values for each cluster

```
CA.a.hcentres_3 <-aggregate(x=CA.a.final_data_3, by=list(cluster=CA.a.fit_3), FUN="mean")
print(CA.a.hcentres_3)
```

```
##   cluster        ML1        ML2         ML3 cluster
## 1       1 -0.7633917  0.0331800 -0.39124910       1
## 2       2  0.5897800 -0.5896435 -0.09320299       2
## 3       3 -0.1343229  1.3023936  1.10393901       3
```

Kmeans clustering

```
set.seed(240)
CA.a.k_3 <- kmeans(CA.df.FAMLa,3,nstart=25)
CA.a.k_3
```

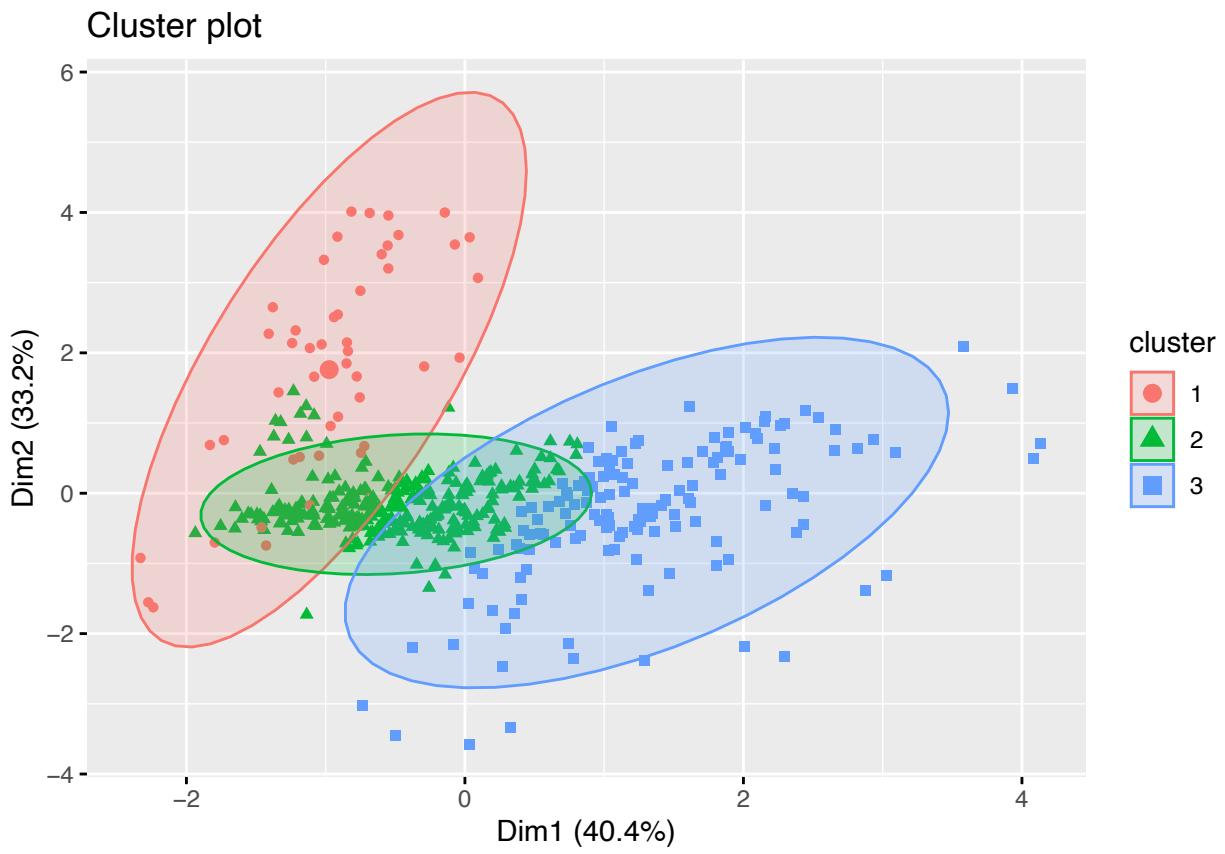
```
## K-means clustering with 3 clusters of sizes 48, 286, 144
##
## Cluster means:
##           ML1         ML2        ML3
## 1  0.05930453  1.52181119  1.2533535
## 2 -0.62832867 -0.05466208 -0.2899611
## 3  0.98096701 -0.68082332 -0.0622415
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   2   2   3   2   2   2   3   3   2   1   3   2   3   2   3   2   2   2   2   2
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   2   2   2   3   2   3   3   3   2   2   2   3   2   2   3   1   2   1   3   1
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   3   2   3   2   1   3   2   3   3   2   2   2   1   3   2   2   2   3   2   2
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   3   3   2   3   1   2   2   2   2   2   3   2   3   2   2   2   2   2   2   2
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   1   2   2   2   2   1   3   2   2   2   1   2   2   3   3   2   2   3   2   2
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   2   2   2   1   2   2   2   2   3   2   2   3   1   2   2   3   2   2   3   2
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   3   3   2   2   3   3   3   1   2   1   3   2   2   2   2   3   2   3   1   2
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   2   2   2   3   1   2   3   2   2   2   3   2   2   3   3   2   3   3   3   3
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   2   2   3   2   2   2   2   2   3   3   2   3   2   2   3   1   3   2   2   3
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
##   3   2   2   2   3   3   2   3   1   1   2   3   2   2   1   1   1   2   2   2
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220
```

```
##   3   2   1   2   1   3   2   3   3   2   2   3   2   3   2   1   3   2   2   1
## 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
##   2   3   2   2   2   3   2   1   3   2   2   3   2   2   1   2   2   2   2   2
## 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260
##   3   1   2   3   1   2   2   2   2   2   2   2   3   2   2   3   2   2   2   3
## 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280
##   3   2   2   2   2   1   2   3   2   2   2   3   2   3   2   3   3   2   3   2
## 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
##   2   3   3   2   3   3   2   2   2   3   2   3   2   3   3   2   3   2   3   3
## 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320
##   2   1   2   3   2   3   2   2   3   2   3   1   2   2   3   1   3   2   2   2
## 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340
##   2   2   2   2   3   2   2   3   2   2   1   1   1   2   2   3   2   2   2   2
## 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
##   1   2   2   2   2   3   3   3   3   3   2   3   3   2   2   2   2   2   2   2
## 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380
##   2   2   1   2   2   2   3   3   2   2   3   1   2   2   2   2   2   3   2   3
## 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400
##   1   3   3   1   2   2   3   2   3   2   2   2   2   2   3   2   2   1   2   2
## 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420
##   3   2   2   2   2   3   1   2   2   2   3   3   2   3   2   3   3   3   2   2
## 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440
##   2   2   2   2   3   2   2   2   2   2   2   3   3   2   1   2   3   2   2   2
## 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460
##   3   2   2   2   2   1   2   2   2   2   2   2   2   2   2   3   2   3   3   3
## 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478
##   2   2   2   3   3   3   1   3   2   3   2   2   2   2   1   2   2   2
##
## Within cluster sum of squares by cluster:
## [1] 119.7329 119.7347 231.3929
##  (between_SS / total_SS =  52.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

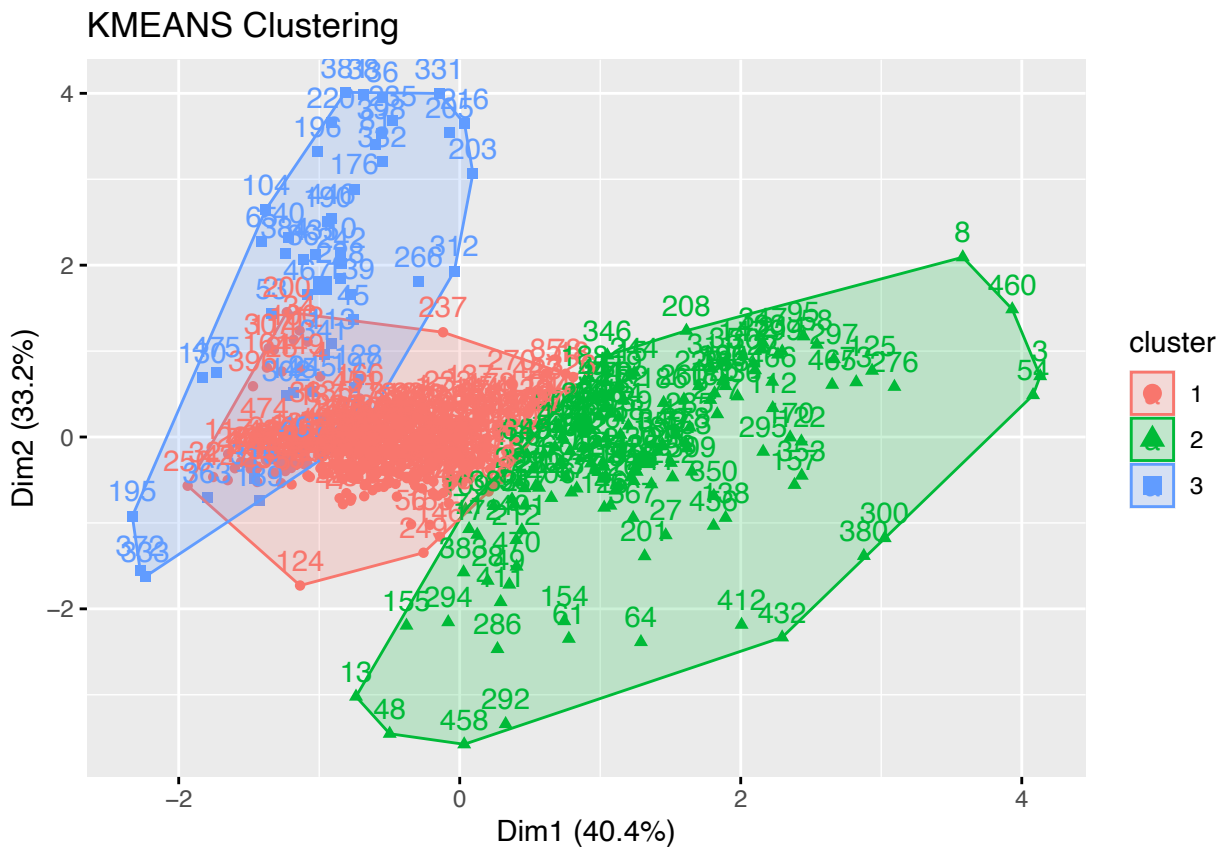
```
fviz_cluster(CA.a.k_3, data= CA.df.FAMLa, geom = "point", frame.type = "norm")
```

```
## Warning: argument frame is deprecated; please use ellipse instead.
```

```
## Warning: argument frame.type is deprecated; please use ellipse.type instead.
```

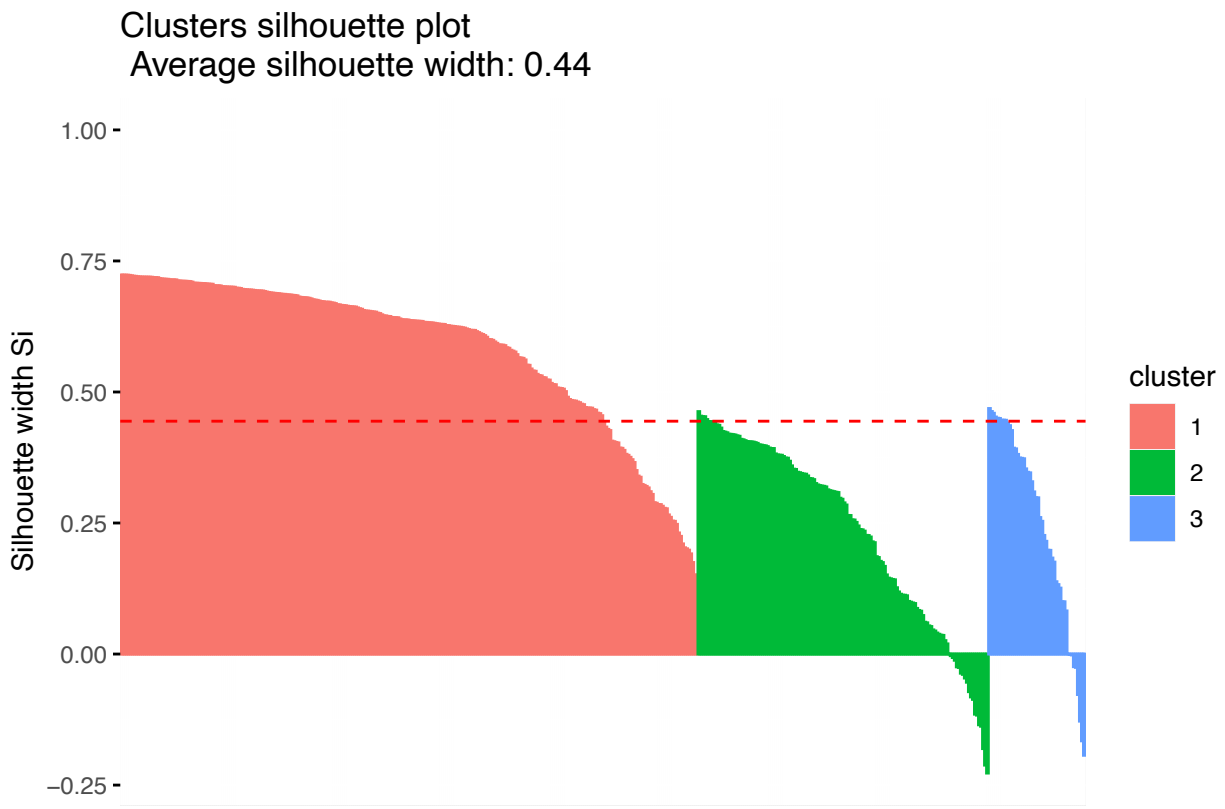



```
res.a.k_3 <-eclust(CA.df.FAMLa, "kmeans", nstart = 25)
```




```
fviz_silhouette(res.a.k_3)
```

```
##   cluster size ave.sil.width
## 1       1  286          0.58
## 2       2  144          0.23
## 3       3   48          0.24
```



Model B

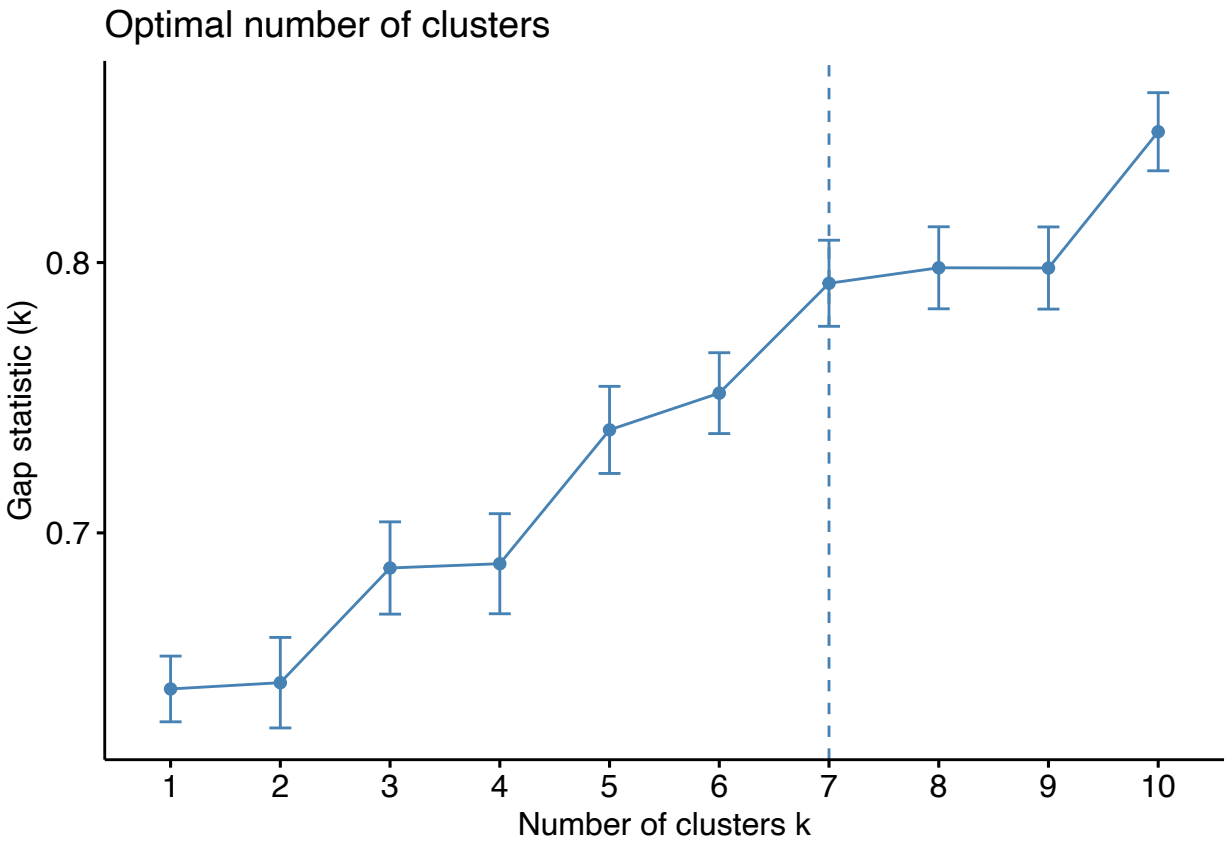
```
ac.b <- function(x) {
  agnes(CA.df.FAMLb, method = x)$ac
}
```

Calculate agglomerative coefficient

```
sapply(m, ac.b)
```

```
##   average    single  complete      ward
## 0.9534457 0.9358105 0.9777409 0.9935047
```

```
gap.h.b <- clusGap(CA.df.FAMLb, FUN = hcut, nstart = 25, K.max = 10, B = 50)
fviz_gap_stat(gap.h.b)
```



For Model B (FA - ML - Orthogonal rotation), the agglomerative coefficient also show that ward's distance method is the most suitable for hierarchical clustering, and the recommend number of K is 3 clusters (first peak)

Finding distance matrix

```
distance_mat.b <- dist(CA.df.FAMLb, method = 'euclidean')
```

Fitting Hierarchical clustering Model to dataset

```
set.seed(240)  # Setting seed
Hierar_cl.b <- hclust(distance_mat.b, method = "ward")
```

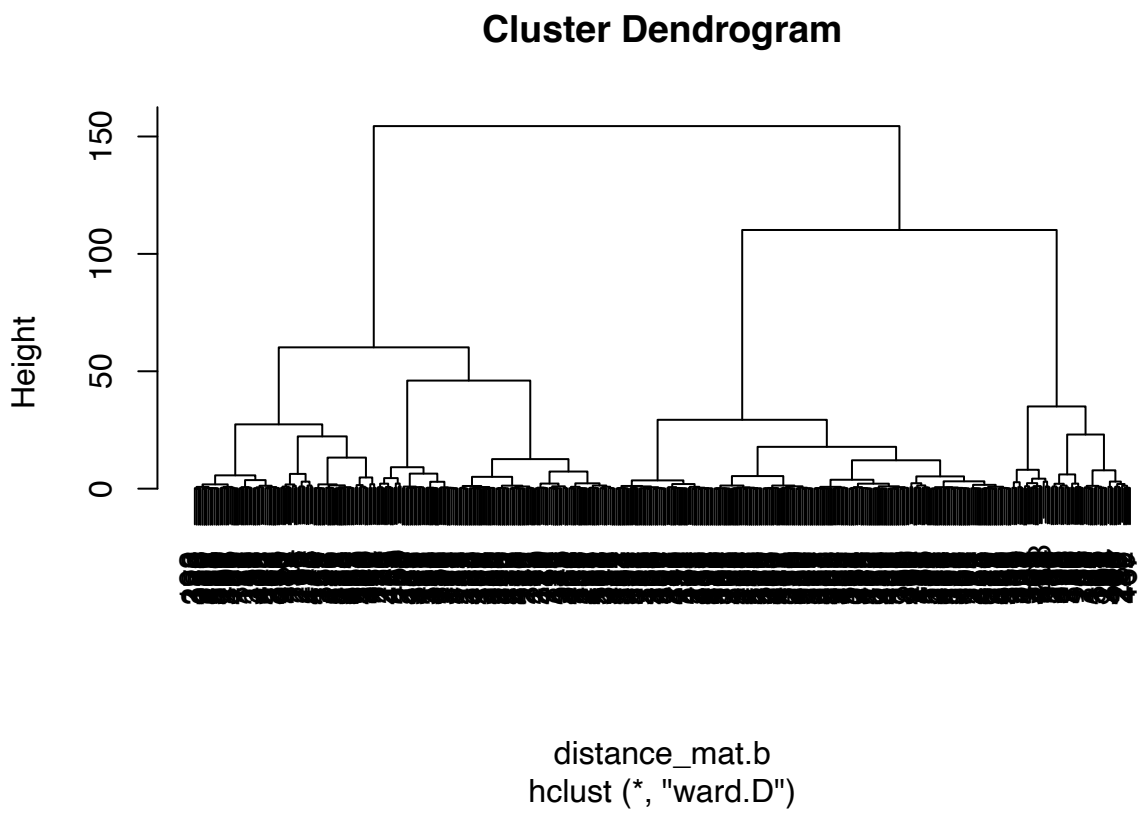
```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```
Hierar_cl.b
```

```
##
## Call:
## hclust(d = distance_mat.b, method = "ward")
##
## Cluster method   : ward.D
## Distance         : euclidean
## Number of objects: 478
```

Plotting dendrogram

```
plot(Hierar_cl.b)
```



Choosing no. of clusters

Cutting tree by no. of clusters

```
CA.b.fit_3 <- cutree(Hierar_cl.b, k = 3)
```

Find number of observations in each cluster

```
table(CA.b.fit_3)
```

```
## CA.b.fit_3
##   1   2   3
## 202 215  61
```

```
CA.b.final_data_3 <-cbind(CA.df.FAMLb, cluster = CA.b.fit_3)
```

Display first six rows of final data

```
head(CA.b.final_data_3)
```

```
##           ML1         ML2        ML3 cluster
## 1 -0.63789064 -0.72883397 -0.1065739       1
## 2  0.06092435 -0.54463707 -0.1192817       2
## 3  3.22744562 -1.32524287  0.7744777       2
## 4 -0.97936769  0.05187441  0.3390246       3
## 5 -1.04660992 -0.26921207 -0.4750785       1
## 6 -1.53819847 -0.09878877 -0.6611076       1
```

Find mean values for each cluster

```
CA.b.hcentres_3 <-aggregate(x=CA.b.final_data_3, by=list(cluster=CA.b.fit_3), FUN="mean")
print(CA.b.hcentres_3)
```

```
##   cluster        ML1        ML2        ML3 cluster
## 1       1 -0.7193626 -0.1707035 -0.4362585       1
## 2       2  0.7707351 -0.2880776 -0.1648798       2
## 3       3 -0.6194415  0.8770313  1.3341894       3
```

Kmeans clustering

```
set.seed(240)
CA.b.k_3 <- kmeans(CA.df.FAMLb,3,nstart=25)
CA.b.k_3
```

```
## K-means clustering with 3 clusters of sizes 48, 286, 144
##
## Cluster means:
##          ML1        ML2        ML3
## 1 -0.5243660  1.1142198  1.5407388
## 2 -0.5603998 -0.2213520 -0.3446994
## 3  1.1670471 -0.2298315 -0.1219394
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   2   2   3   2   2   2   3   3   2   1   3   2   3   2   3   2   2   2   2   2
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   2   2   2   3   2   3   3   3   2   2   2   3   2   2   3   1   2   1   3   1
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   3   2   3   2   1   3   2   3   3   2   2   2   1   3   2   2   2   3   2   2
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   3   3   2   3   1   2   2   2   2   2   3   2   3   2   2   2   2   2   2   2
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   1   2   2   2   2   1   3   2   2   2   1   2   2   3   3   2   2   3   2   2
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   2   2   2   1   2   2   2   2   3   2   2   3   1   2   2   3   2   2   3   2
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   3   3   2   2   3   3   3   1   2   1   3   2   2   2   2   3   2   3   1   2
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   2   2   2   3   1   2   3   2   2   2   3   2   2   3   3   2   3   3   3   3
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   2   2   3   2   2   2   2   2   3   3   2   3   2   2   3   1   3   2   2   3
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
```

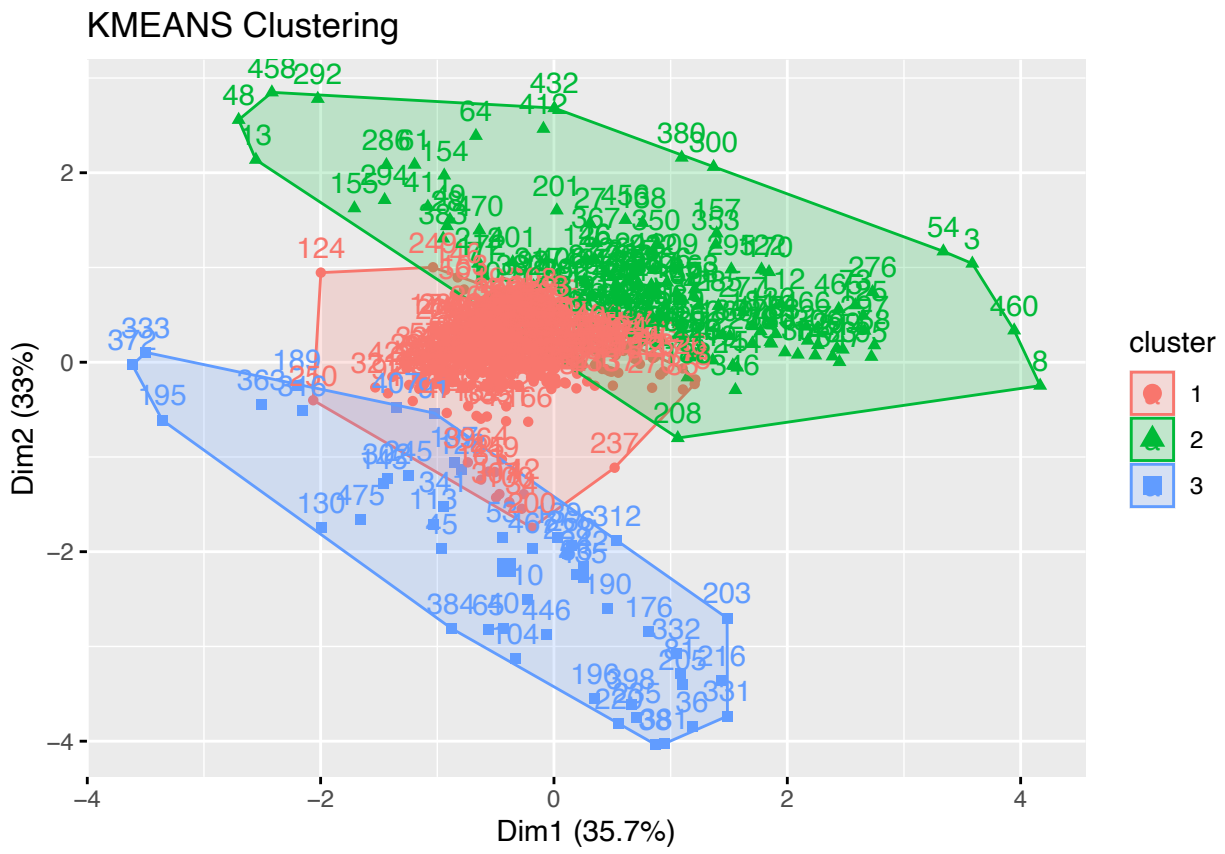
```
##   3   2   2   2   3   3   2   3   1   1   2   3   2   2   1   1   1   2   2   2
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220
##   3   2   1   2   1   3   2   3   3   2   2   3   2   3   2   1   3   2   2   1
## 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
##   2   3   2   2   2   3   2   1   3   2   2   3   2   2   1   2   2   2   2   2
## 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260
##   3   1   2   3   1   2   2   2   2   2   2   2   3   2   2   3   2   2   2   3
## 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280
##   3   2   2   2   2   1   2   3   2   2   2   3   2   3   2   3   3   2   3   2
## 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
##   2   3   3   2   3   3   2   2   2   3   2   3   2   3   3   2   3   2   3   3
## 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320
##   2   1   2   3   2   3   2   2   3   2   3   1   2   2   3   1   3   2   2   2
## 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340
##   2   2   2   2   3   2   2   3   2   2   1   1   1   2   2   3   2   2   2   2
## 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
##   1   2   2   2   2   3   3   3   3   3   2   3   3   2   2   2   2   2   2   2
## 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380
##   2   2   1   2   2   2   3   3   2   2   3   1   2   2   2   2   2   3   2   3
## 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400
##   1   3   3   1   2   2   3   2   3   2   2   2   2   2   3   2   2   1   2   2
## 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420
##   3   2   2   2   2   3   1   2   2   2   3   3   2   3   2   3   3   3   2   2
## 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440
##   2   2   2   2   3   2   2   2   2   2   2   3   3   2   1   2   3   2   2   2
## 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460
##   3   2   2   2   2   1   2   2   2   2   2   2   2   2   2   3   2   3   3   3
## 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478
##   2   2   2   3   3   3   1   3   2   3   2   2   2   2   1   2   2   2
##
## Within cluster sum of squares by cluster:
## [1] 119.7329 119.7347 231.3929
##  (between_SS / total_SS =  52.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
fviz_cluster(CA.b.k_3, data= CA.df.FAMLb, geom = "point", frame.type = "norm")
```

```
## Warning: argument frame is deprecated; please use ellipse instead.
```

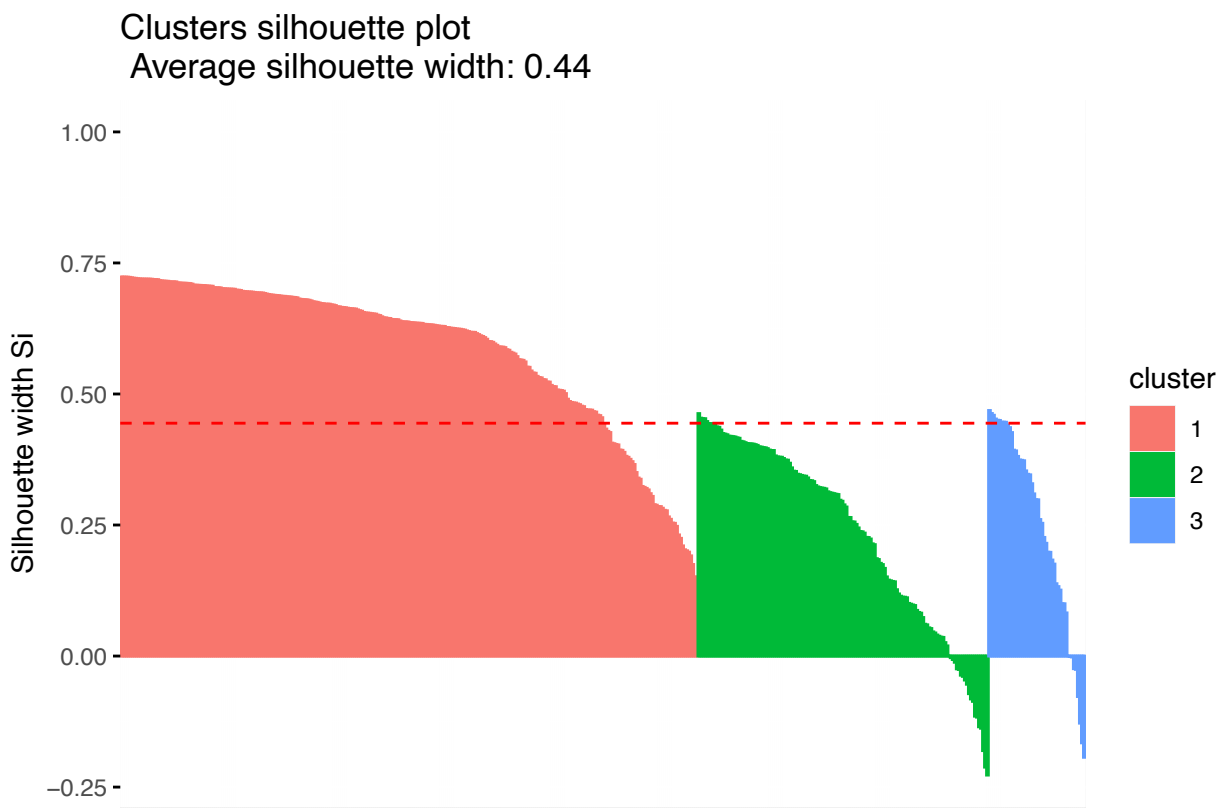
```
## Warning: argument frame.type is deprecated; please use ellipse.type instead.
```


```
res.b.k_3 <-eclust(CA.df.FAMLb, "kmeans", nstart = 25)
```




```
fviz_silhouette(res.b.k_3)
```

```
##   cluster size ave.sil.width
## 1       1  286          0.58
## 2       2  144          0.23
## 3       3   48          0.24
```



Model C

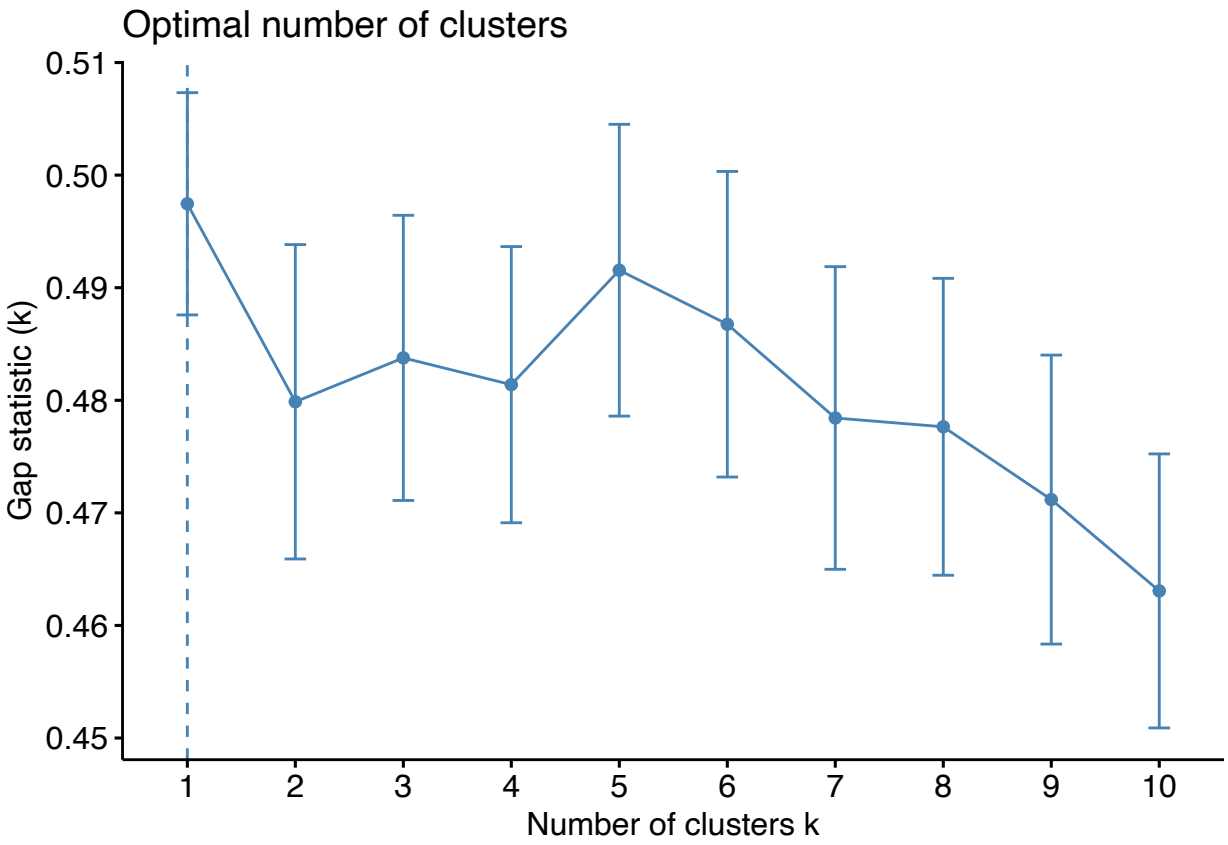
```
ac.c <- function(x) {
  agnes(CA.df.FAPCc, method = x)$ac
}
```

Calculate agglomerative coefficient

```
sapply(m, ac.c)
```

```
##   average    single  complete      ward
## 0.7971063 0.5684153 0.8820109 0.9638754
```

```
gap.h.c <- clusGap(CA.df.FAPCc, FUN = hcut, nstart = 25, K.max = 10, B = 50)
fviz_gap_stat(gap.h.c)
```



For Model C (FA - PC - Orthogonal rotation), the agglomerative coefficient also show that ward's distance method is the most suitable for hierarchical clustering, and the recommend number of K is 3 and 5 clusters, for the model C, we will use 3 and 5 to clustering.

Clustering for 3 groups

Finding distance matrix

```
distance_mat.c1 <- dist(CA.df.FAPCc, method = 'euclidean')
```

Fitting Hierarchical clustering Model to dataset

```
set.seed(240)  # Setting seed
Hierar_cl.c1 <- hclust(distance_mat.c1, method = "ward")
```

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

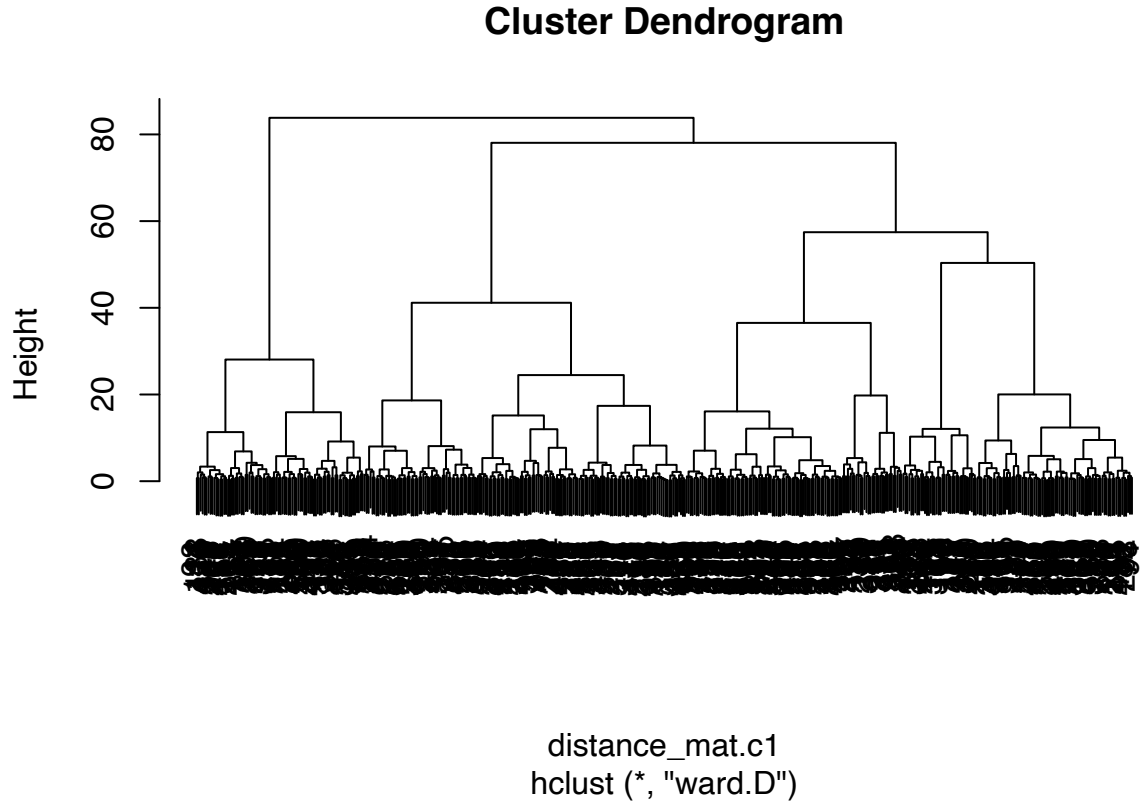
```
Hierar_cl.c1
```

```
##
## Call:
## hclust(d = distance_mat.c1, method = "ward")
##
## Cluster method   : ward.D
```

```
## Distance         : euclidean
## Number of objects: 480
```

Plotting dendrogram

```
plot(Hierar_cl.c1)
```



Choosing no. of clusters

Cutting tree by no. of clusters

```
CA.c1.fit_3 <- cutree(Hierar_cl.c1, k = 3 )
```

Find number of observations in each cluster

```
table(CA.c1.fit_3)
```

```
## CA.c1.fit_3
##   1   2   3
## 165 229  86
```

```
CA.c1.final_data_3 <-cbind(CA.df.FAPCc, cluster = CA.c1.fit_3)
```

Display first six rows of final data

```
head(CA.c1.final_data_3)
```

```
##           RC1        RC2         RC3        RC5        RC4 cluster
## 1 -0.65518007 -0.2207361  0.06769853  1.4456059 -1.0097101       1
## 2 -0.02127861  0.3871332 -0.16007476  0.9223548 -1.0841590       1
## 3 -0.93207252  0.4423952  0.42429237  0.8562932  0.2541208       2
## 4 -0.94269044 -1.0335406  0.47719445  1.1980461 -0.2898919       2
## 5 -1.49403303 -0.8072241  0.89922381 -1.2374152  0.5097328       1
## 6  0.96504199  0.5996860  0.89933201 -2.0314438 -1.0009010       2
```

Find mean values for each cluster

```
CA.c1.hcentres_3 <-aggregate(x=CA.c1.final_data_3, by=list(cluster=CA.c1.fit_3), FUN="mean")
print(CA.c1.hcentres_3)
```

```
##   cluster        RC1         RC2        RC3        RC5        RC4 cluster
## 1       1 -0.2789023 -0.34697305 -0.7152258 -0.3070687 -0.1721829       1
## 2       2  0.1174428  0.07746538  0.4225230  0.3889901 -0.3882362       2
## 3       3 -0.1066173  0.22666868  0.3465989 -0.1510918  1.3895277       3
```

Kmeans clustering

```
set.seed(240)
CA.c1.k_3 <- kmeans(CA.df.FAPCc,3,nstart=25)
CA.c1.k_3
```

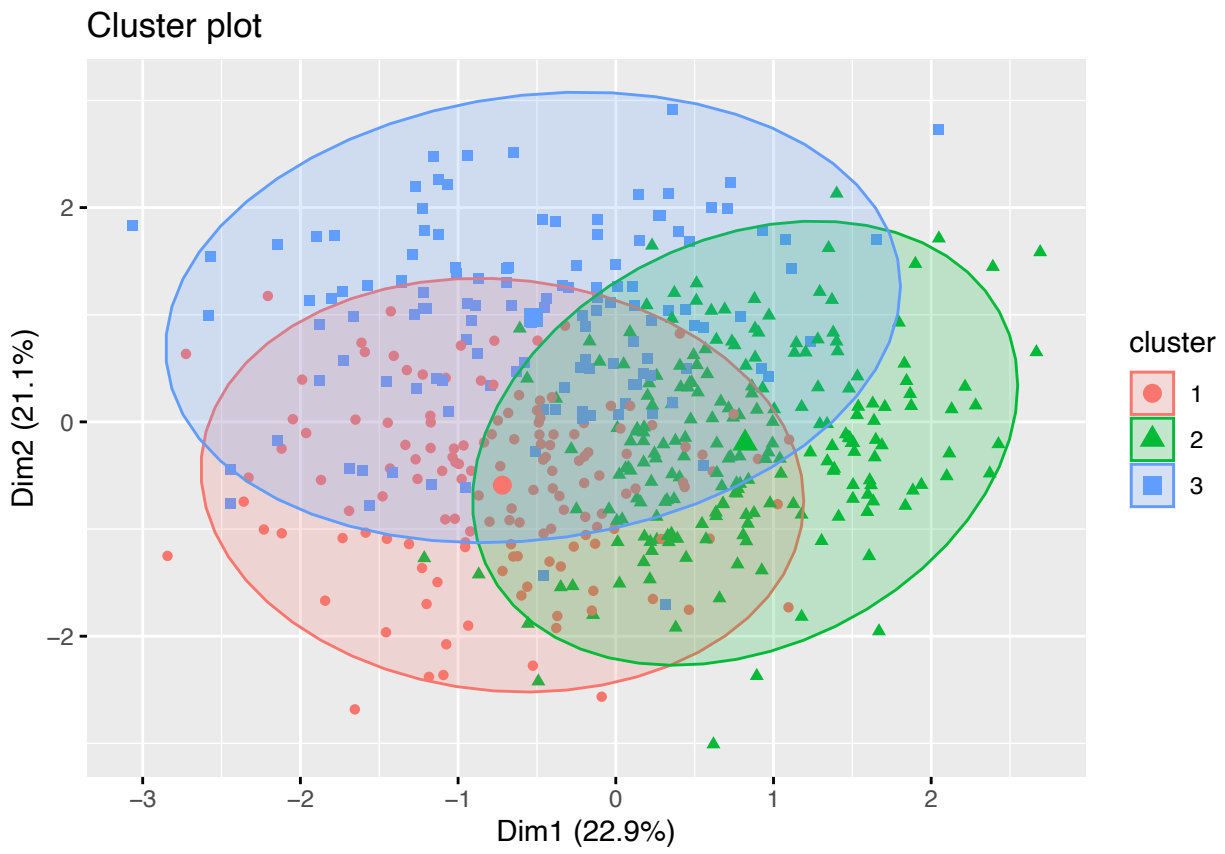
```
## K-means clustering with 3 clusters of sizes 143, 208, 129
##
## Cluster means:
##           RC1        RC2        RC3         RC5        RC4
## 1 -0.12727082  0.4104184  0.6343779  0.48533070 -0.6396614
## 2 -0.05848457 -0.4758724 -0.6688112 -0.22349543 -0.2654999
## 3  0.01605386  0.1571641  0.4414706  0.01940503  1.1540982
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   1   1   1   1   3   1   3   2   3   1   3   1   2   2   1   3   2   2   2   2
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   1   2   2   2   1   1   2   3   2   1   2   1   1   2   1   3   3   3   1   1
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   1   2   2   2   1   2   2   3   3   3   3   2   3   3   2   1   2   1   2   2
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   2   3   2   2   3   3   1   1   2   3   3   2   1   1   2   2   2   2   2   2
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   2   2   1   2   1   2   3   3   1   2   1   1   3   3   3   1   2   3   2   1
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   1   3   2   1   3   1   3   3   2   2   3   2   3   2   1   3   2   1   1   1
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   2   2   1   2   2   1   1   3   2   1   1   1   2   3   3   3   3   3   1   3
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
##   2   1   2   3   3   3   3   2   3   2   2   3   2   3   1   3   2   2   2   2
```

```
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   1   3   2   2   2   2   2   2   1   2   2   2   3   1   1   2   3   2   2   1
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
##   2   1   2   3   2   2   3   2   3   1   3   3   3   1   3   1   2   2   3   2
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220
##   1   1   1   2   3   3   3   2   3   1   1   3   2   2   2   3   2   2   1   3
## 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
##   1   2   2   3   2   1   2   1   2   2   1   2   2   3   1   3   3   2   2   1
## 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260
##   1   1   3   2   1   2   2   1   1   1   1   1   2   3   2   2   2   2   3   2
## 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280
##   1   3   1   3   1   1   3   2   1   2   1   1   1   2   3   2   1   2   2   2
## 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
##   1   3   2   2   2   2   1   3   2   2   2   2   2   3   3   2   1   2   3   2
## 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320
##   2   1   3   3   2   1   1   2   1   2   2   1   2   2   2   3   1   3   3   1
## 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340
##   1   2   1   2   1   2   2   2   2   3   1   2   1   1   3   1   1   3   1   1
## 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
##   3   2   3   3   1   1   2   3   3   1   3   1   3   1   3   2   2   2   1   1
## 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380
##   2   1   1   2   2   3   3   2   2   2   2   1   2   3   3   3   1   1   2   1
## 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400
##   2   2   2   3   3   1   3   1   2   1   1   2   2   2   1   2   2   2   3   3
## 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420
##   1   2   2   3   2   1   2   3   3   1   2   2   2   1   3   2   3   2   1   1
## 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440
##   2   2   1   3   2   2   2   1   1   3   2   1   2   2   3   2   2   2   2   1
## 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460
##   3   2   2   3   2   1   3   2   2   3   1   1   2   3   1   1   1   1   3   2
## 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480
##   2   2   2   2   2   2   3   2   3   1   2   2   2   3   3   2   3   3   2   3
##
## Within cluster sum of squares by cluster:
## [1] 449.4770 633.9987 448.9536
##  (between_SS / total_SS =  26.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

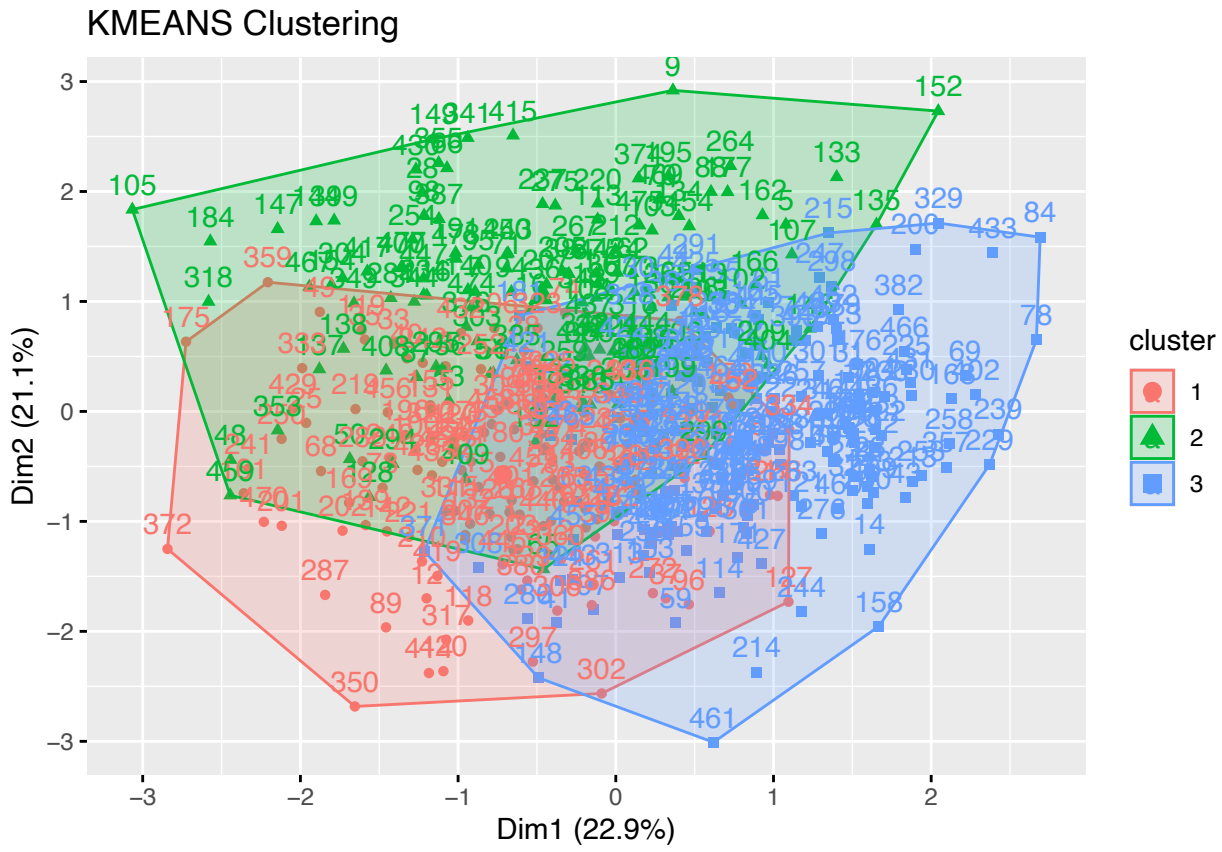
```
fviz_cluster(CA.c1.k_3, data= CA.df.FAPCc, geom = "point", frame.type = "norm")
```

```
## Warning: argument frame is deprecated; please use ellipse instead.
```

```
## Warning: argument frame.type is deprecated; please use ellipse.type instead.
```

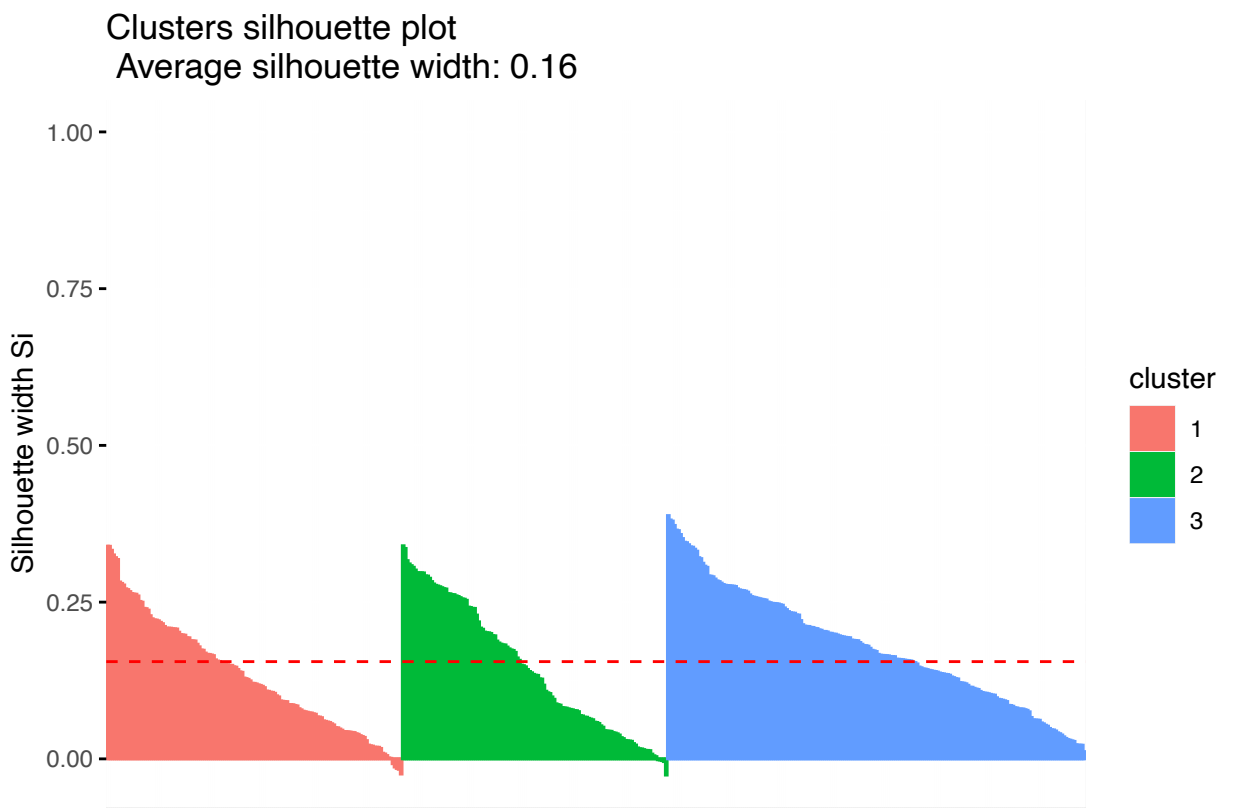



```
res.c1.k_3 <-eclust(CA.df.FAPCc, "kmeans", nstart = 25, k=3)
```




```
fviz_silhouette(res.c1.k_3)
```

```
##   cluster size ave.sil.width
## 1       1  145          0.13
## 2       2  130          0.14
## 3       3  205          0.18
```



Clustering for 5 groups

Finding distance matrix

```
distance_mat.c2 <- dist(CA.df.FAPCc, method = 'euclidean')
```

Fitting Hierarchical clustering Model to dataset

```
set.seed(240)  # Setting seed
Hierar_cl.c2 <- hclust(distance_mat.c2, method = "ward")
```

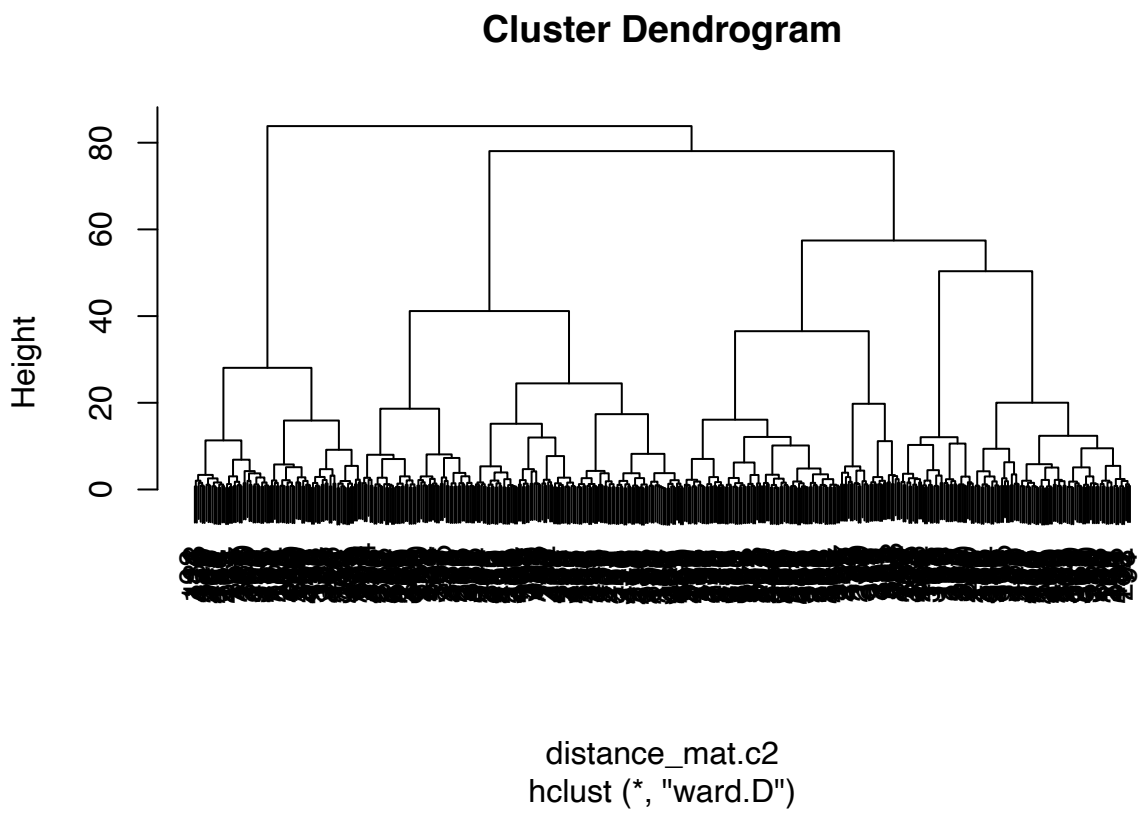
```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```
Hierar_cl.c2
```

```
##
## Call:
## hclust(d = distance_mat.c2, method = "ward")
##
## Cluster method   : ward.D
## Distance         : euclidean
## Number of objects: 480
```

Plotting dendrogram

```
plot(Hierar_cl.c2)
```



Choosing no. of clusters

Cutting tree by no. of clusters

```
CA.c2.fit_5 <- cutree(Hierar_cl.c2, k = 5 )
```

Find number of observations in each cluster

```
table(CA.c2.fit_5)
```

```
## CA.c2.fit_5
##   1   2   3   4   5
## 165  80 111  86  38
```

```
CA.c2.final_data_5 <-cbind(CA.df.FAPCc, cluster = CA.c2.fit_5)
```

Display first six rows of final data

```
head(CA.c2.final_data_5)
```

```
##           RC1        RC2         RC3        RC5        RC4 cluster
## 1 -0.65518007 -0.2207361  0.06769853  1.4456059 -1.0097101       1
## 2 -0.02127861  0.3871332 -0.16007476  0.9223548 -1.0841590       1
## 3 -0.93207252  0.4423952  0.42429237  0.8562932  0.2541208       2
## 4 -0.94269044 -1.0335406  0.47719445  1.1980461 -0.2898919       2
## 5 -1.49403303 -0.8072241  0.89922381 -1.2374152  0.5097328       1
## 6  0.96504199  0.5996860  0.89933201 -2.0314438 -1.0009010       3
```

Find mean values for each cluster

```
CA.c2.hcentres_5 <-aggregate(x=CA.c2.final_data_5, by=list(cluster=CA.c2.fit_5), FUN="mean")
print(CA.c2.hcentres_5)
```

```
##   cluster        RC1         RC2        RC3        RC5         RC4 cluster
## 1       1 -0.2789023 -0.34697305 -0.7152258 -0.3070687 -0.17218293       1
## 2       2 -0.6269338  0.79588734  0.5163794  0.6634279 -0.26117004       2
## 3       3  0.3103611 -0.43628643  0.6567676 -0.1426933 -0.63880077       3
## 4       4 -0.1066173  0.22666868  0.3465989 -0.1510918  1.38952767       4
## 5       5  1.1210273  0.06569418 -0.4593101  1.3643014  0.07616834       5
```

Kmeans clustering

```
set.seed(240)
CA.c2.k_5 <- kmeans(CA.df.FAPCc, 5, nstart=25)
CA.c2.k_5
```

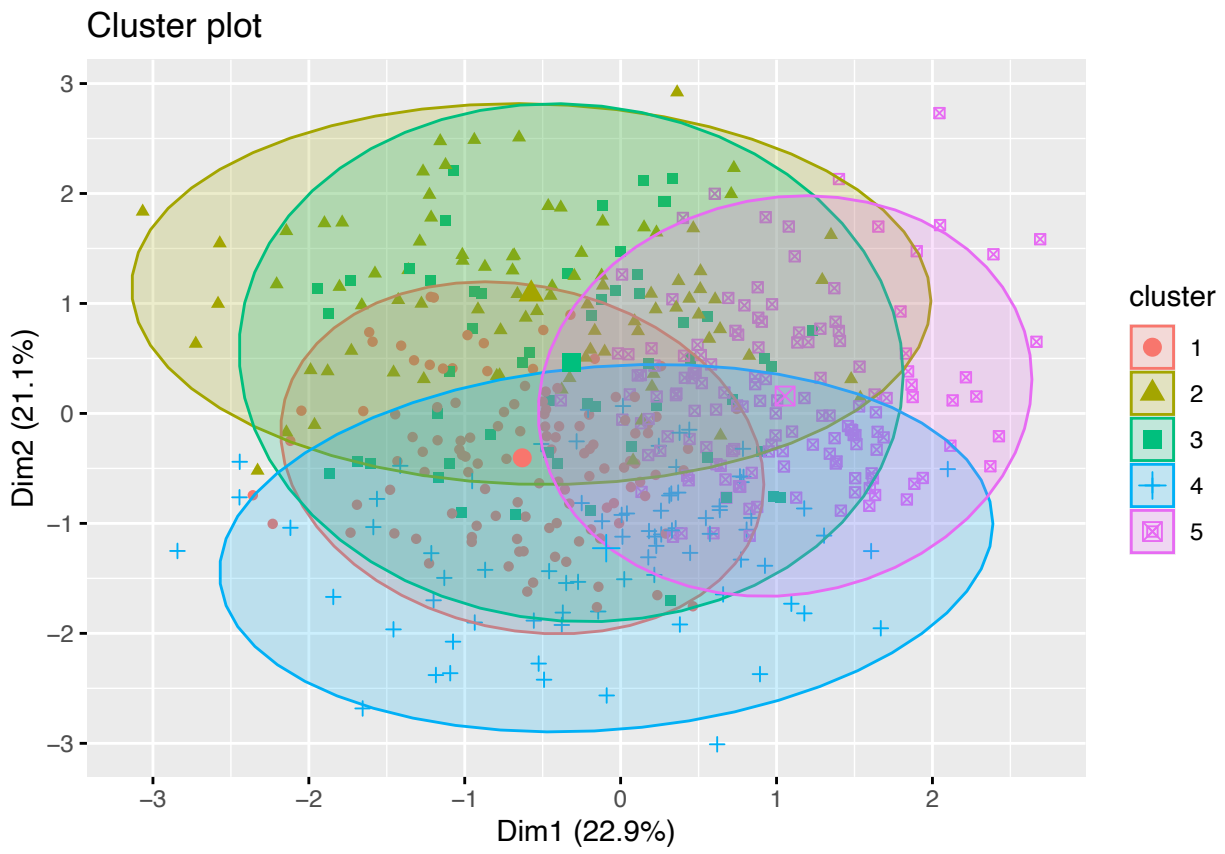
```
## K-means clustering with 5 clusters of sizes 116, 88, 56, 79, 141
##
## Cluster means:
##           RC1           RC2        RC3        RC5        RC4
## 1 -0.5696620  0.3059391087  0.4799930  0.5713276 -0.6281551
## 2 -0.4841647  0.8938039594 -0.4435413 -0.0137209  0.8992331
## 3  0.7538214  0.0002599473  1.3510013 -0.5215177  0.6860598
## 4  1.1527294 -0.0902603026 -0.6291118  0.5927522 -0.3580855
## 5 -0.3750773 -0.9010294511 -0.2414949 -0.4061738 -0.1008087
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   1   1   1   1   5   3   3   5   2   1   2   4   2   4   1   2   2   5   5   5
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   1   2   5   5   2   3   4   2   1   1   2   1   1   4   1   2   3   3   1   1
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   4   5   5   2   3   4   5   4   3   3   3   5   1   2   5   2   5   1   4   5
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   5   3   4   5   4   3   1   3   5   3   2   5   1   1   4   5   1   5   5   5
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   5   2   1   5   1   4   2   5   4   5   1   5   3   2   2   1   4   2   5   1
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   5   5   2   1   2   2   5   3   2   5   3   5   2   4   3   1   5   4   1   1
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
##   4   5   5   4   4   1   4   4   4   1   3   1   5   5   5   2   2   2   4   3
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
```

```
##   1   1   5   2   4   3   2   4   2   1   5   5   4   2   1   3   5   4   4   4
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   1   5   1   4   5   5   5   5   1   4   4   5   3   1   2   3   2   4   2   1
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
##   2   1   4   2   5   5   5   5   3   1   2   4   3   1   3   1   5   5   5   5
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220
##   4   1   1   5   3   2   2   5   3   1   1   3   5   4   2   5   4   5   1   3
## 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
##   1   4   5   5   5   4   5   1   5   5   1   5   2   1   3   5   2   2   5   1
## 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260
##   2   1   2   4   1   5   2   1   1   1   1   1   5   2   1   4   5   5   2   5
## 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280
##   1   5   1   2   1   3   2   1   1   5   5   1   1   4   2   4   3   4   1   5
## 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
##   2   5   5   4   5   4   4   3   5   2   2   5   5   4   1   4   4   5   3   4
## 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320
##   5   4   1   3   2   4   1   4   1   5   3   1   4   5   2   2   4   2   4   1
## 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340
##   1   5   1   1   1   5   5   2   5   3   1   5   2   5   3   1   1   3   1   1
## 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
##   2   5   2   2   1   1   5   2   2   4   2   1   2   3   2   4   4   5   2   1
## 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380
##   5   1   1   5   1   1   3   2   5   2   4   4   5   3   2   3   3   3   5   1
## 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400
##   5   5   4   2   3   1   3   1   5   5   1   5   5   5   1   5   5   5   2   2
## 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420
##   1   5   5   3   1   1   4   2   4   5   5   5   3   4   2   4   2   2   4   4
## 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440
##   1   5   1   5   2   5   4   1   1   2   5   1   5   4   3   5   2   1   5   1
## 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460
##   2   5   4   5   5   1   3   4   5   2   1   5   5   2   1   1   3   1   4   5
## 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480
##   4   5   5   5   4   5   3   4   3   1   5   4   1   2   2   5   3   2   5   5
##
## Within cluster sum of squares by cluster:
## [1] 240.7505 268.5284 171.1282 247.2878 291.1177
##  (between_SS / total_SS =  41.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

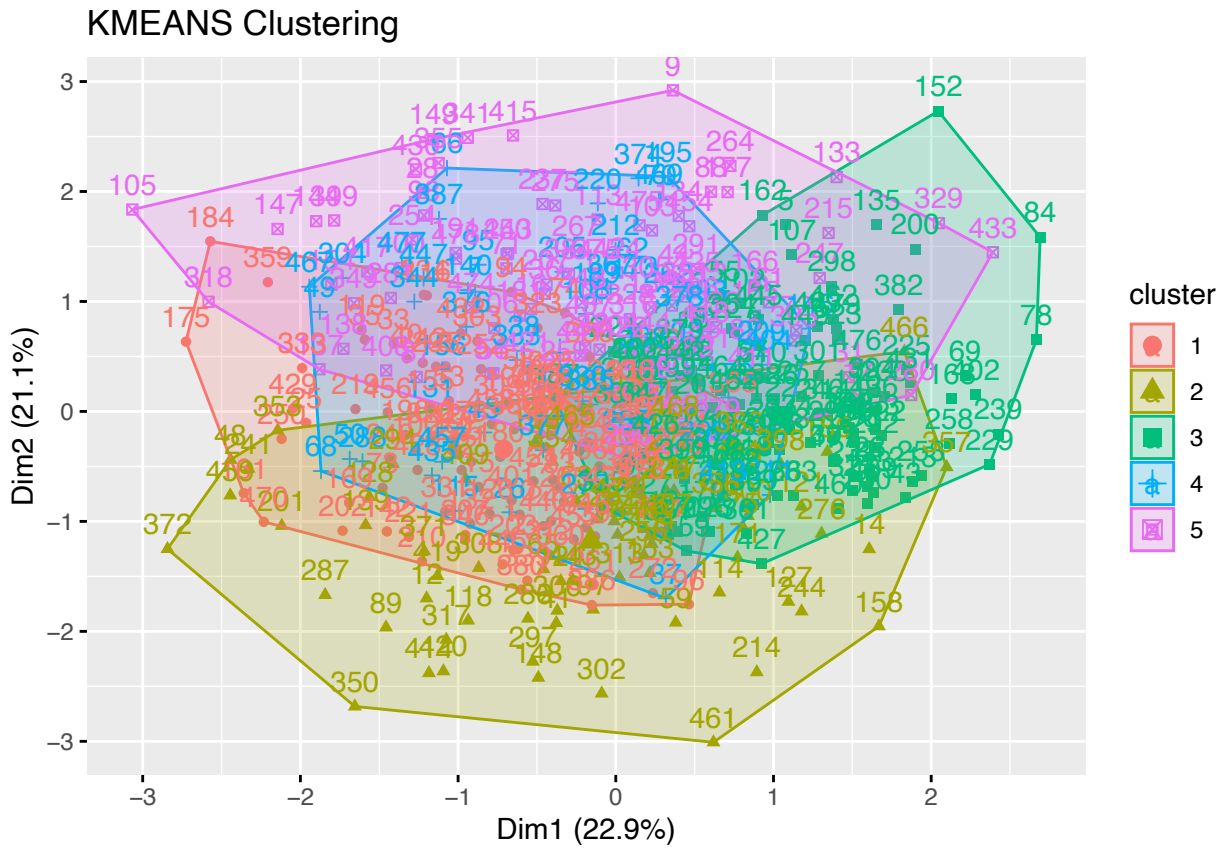
```
fviz_cluster(CA.c2.k_5, data= CA.df.FAPCc, geom = "point", frame.type = "norm")
```

```
## Warning: argument frame is deprecated; please use ellipse instead.
```

```
## Warning: argument frame.type is deprecated; please use ellipse.type instead.
```

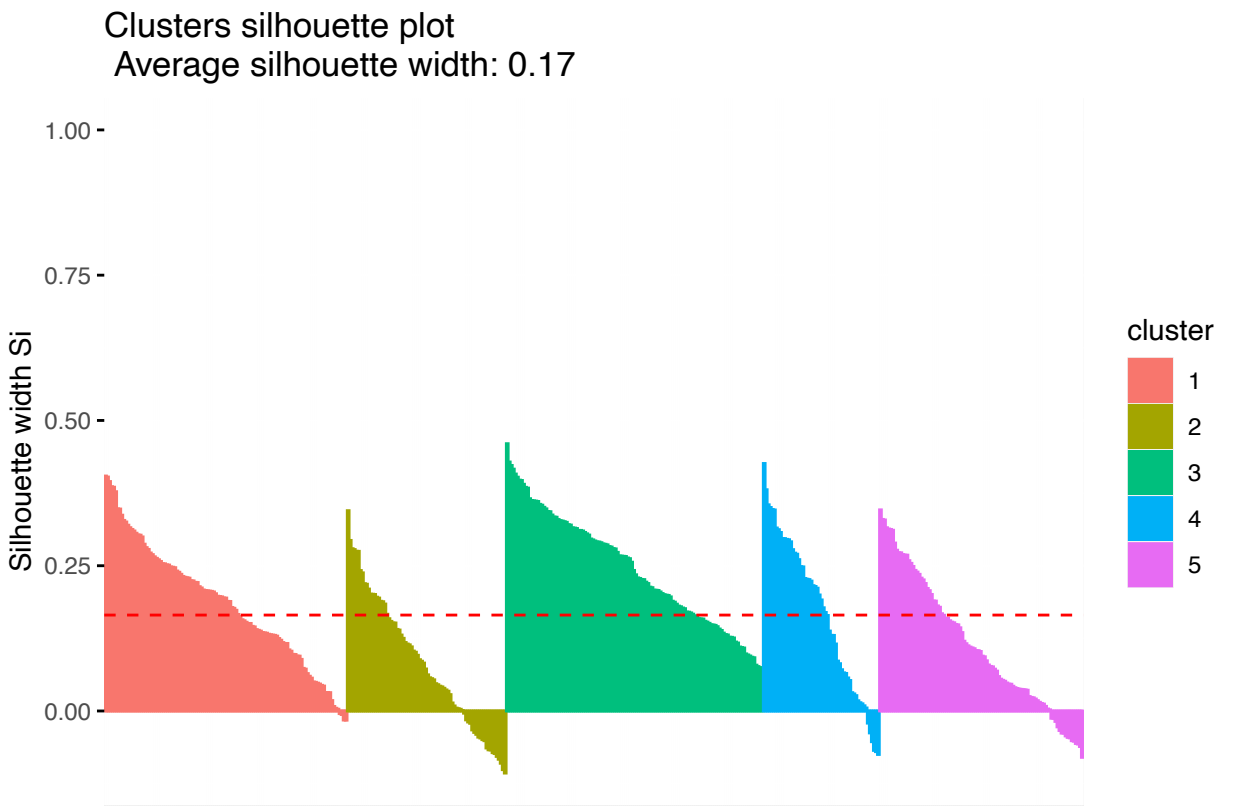



```
res.c2.k_5 <-eclust(CA.df.FAPCc, "kmeans", nstart = 25, k=5)
```




```
fviz_silhouette(res.c2.k_5)
```

```
##   cluster size ave.sil.width
## 1       1  119          0.18
## 2       2   78          0.08
## 3       3  126          0.24
## 4       4   57          0.17
## 5       5  100          0.11
```



Validation - Model A

```
# chosee the validation set
set.seed(20)
vali_FAMLa <- sample(1:nrow(CA.df.FAMLa), 100, replace = FALSE, prob = NULL)
validf.FAMLa <- CA.df.FAMLa[vali_FAMLa, ]

subset_cluster_assignments <- kmeans(validf.FAMLa, centers = CA.a.k_3$centers, iter.max = 10, nstart = 
validf.FAMLa <- as.data.frame(validf.FAMLa)
validf.FAMLa$validation_result <- subset_cluster_assignments
previous_result_FAMLa3 <- CA.a.k_3$cluster[vali_FAMLa]
validf.FAMLa$previous_result <- previous_result_FAMLa3
validf.FAMLa <- validf.FAMLa %>%
  mutate(difference = validation_result - previous_result_FAMLa3)
accuracy_modelA <- sum(validf.FAMLa$difference == 0) / nrow(validf.FAMLa)
accuracy_modelA
```

```
## [1] 0.97
```

Validation - Model A (5 iterations)

```r
set.seed(10) # Ensure reproducibility
total_accuracy <- numeric(5) # Initialize a vector to store accuracy for each iteration

for (i in 1:5) {
  # Randomly select 100 data points for the validation set
  vali_indices <- sample(1:nrow(CA.df.FAMLa), 100, replace = FALSE)
  validation_df <- CA.df.FAMLa[vali_indices, ]

  # Apply k-means clustering to the validation set using predefined centers
  subset_cluster_assignments <- kmeans(validation_df, centers = CA.a.k_3$centers, iter.max = 10, nstart

  # Add clustering results to the validation dataframe
  validation_df <- as.data.frame(validation_df)
  validation_df$validation_result <- subset_cluster_assignments

  # Retrieve the previous clustering results for the selected validation indices
  previous_result <- CA.a.k_3$cluster[vali_indices]
  validation_df$previous_result <- previous_result

  # Calculate the difference between new and previous clustering results
  validation_df <- validation_df %>% mutate(difference = validation_result - previous_result)

  # Calculate and store the accuracy for the current iteration
  accuracy <- sum(validation_df$difference == 0) / nrow(validation_df)
  total_accuracy[i] <- accuracy
}

# Calculate the mean accuracy across all iterations
mean_accuracy_modelA <- mean(total_accuracy)
mean_accuracy_modelA
```

```
## [1] 0.954
```

Validation - Model B

```r
# choose the validation set
set.seed(20)
vali_modelB <- sample(1:nrow(CA.df.FAMLb), 100, replace = FALSE, prob = NULL)
validf.modelB <- CA.df.FAMLb[vali_modelB, ]

subset_cluster_assignments <- kmeans(validf.modelB, centers = CA.b.k_3$centers, iter.max = 10, nstart =
validf.modelB <- as.data.frame(validf.modelB)
validf.modelB$validation_result <- subset_cluster_assignments
previous_result_modelB <- CA.b.k_3$cluster[vali_modelB]
validf.modelB$previous_result <- previous_result_modelB
validf.modelB <- validf.modelB %>%
  mutate(difference = validation_result - previous_result_modelB)
accuracy_modelB <- sum(validf.modelB$difference == 0) / nrow(validf.modelB)
accuracy_modelB
```

```
## [1] 0.97
```

Validation - Model B (5 iterations)

```
set.seed(10) # Ensure reproducibility
total_accuracy <- numeric(5) # Initialize a vector to store accuracy for each iteration

for (i in 1:5) {
  # Randomly select 100 data points for the validation set
  vali_indices <- sample(1:nrow(CA.df.FAMLb), 100, replace = FALSE)
  validation_df <- CA.df.FAMLb[vali_indices, ]

  # Apply k-means clustering to the validation set using predefined centers
  subset_cluster_assignments <- kmeans(validation_df, centers = CA.b.k_3$centers, iter.max = 10, nstart

  # Add clustering results to the validation dataframe
  validation_df <- as.data.frame(validation_df)
  validation_df$validation_result <- subset_cluster_assignments

  # Retrieve the previous clustering results for the selected validation indices
  previous_result <- CA.b.k_3$cluster[vali_indices]
  validation_df$previous_result <- previous_result

  # Calculate the difference between new and previous clustering results
  validation_df <- validation_df %>% mutate(difference = validation_result - previous_result)

  # Calculate and store the accuracy for the current iteration
  accuracy <- sum(validation_df$difference == 0) / nrow(validation_df)
  total_accuracy[i] <- accuracy
}

# Calculate the mean accuracy across all iterations
mean_accuracy_modelB <- mean(total_accuracy)
mean_accuracy_modelB
```

```
## [1] 0.954
```

Validation - Model C1 (3 clusters)

```
# choose the validation set
set.seed(10)
vali_modelC.1 <- sample(1:nrow(CA.df.FAPCc), 100, replace = FALSE, prob = NULL)
validf.modelC.1 <- CA.df.FAPCc[vali_modelC.1, ]

subset_cluster_assignments <- kmeans(validf.modelC.1, centers = CA.c1.k_3$centers, iter.max = 10, nstar
validf.modelC.1 <- as.data.frame(validf.modelC.1)
validf.modelC.1$validation_result <- subset_cluster_assignments
previous_result_modelC <- CA.c1.k_3$cluster[vali_modelC.1]
validf.modelC.1$previous_result <- previous_result_modelC
validf.modelC.1 <- validf.modelC.1 %>%
  mutate(difference = validation_result - previous_result_modelC)
accuracy_modelC1 <- sum(validf.modelC.1$difference == 0) / nrow(validf.modelC.1)
accuracy_modelC1
```

```
## [1] 0.61
```

Validation - Model C2 (5 clusters)

```
# choose the validation set
set.seed(10)
vali_modelC.2 <- sample(1:nrow(CA.df.FAPCc), 100, replace = FALSE, prob = NULL)
validf.modelC.2 <- CA.df.FAPCc[vali_modelC.2, ]

subset_cluster_assignments <- kmeans(validf.modelC.2, centers = CA.c2.k_5$centers, iter.max = 10, nstar
validf.modelC.2 <- as.data.frame(validf.modelC.2)
validf.modelC.2$validation_result <- subset_cluster_assignments
previous_result_modelC <- CA.c2.k_5$cluster[vali_modelC.2]
validf.modelC.2$previous_result <- previous_result_modelC
validf.modelC.2 <- validf.modelC.2 %>%
  mutate(difference = validation_result - previous_result_modelC)
accuracy_modelC2 <- sum(validf.modelC.2$difference == 0) / nrow(validf.modelC.2)
accuracy_modelC2
```

```
## [1] 0.86
```

The results show that the results of model A and model B are the same. But model A contains less cross-loadings, so we select Model B as the best model.

Interpretation (Vicky)

```
final_data <- sp.df[original.df$index, ]
final_data$cluster <- CA.b.k_3$cluster

final_data_with_categories <- Original_data_wo_outliers_sample[original.df$index, ]
final_data_with_categories$cluster <- CA.b.k_3$cluster

#Just to check if we have the same records
print(final_data$loan_amnt - final_data_with_categories$loan_amnt)
```

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [149] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [223] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [260] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [297] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [334] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [371] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [408] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [445] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```r
print(final_data$annual_inc - final_data_with_categories$annual_inc)
```

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [149] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [223] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [260] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [297] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [334] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [371] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [408] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [445] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

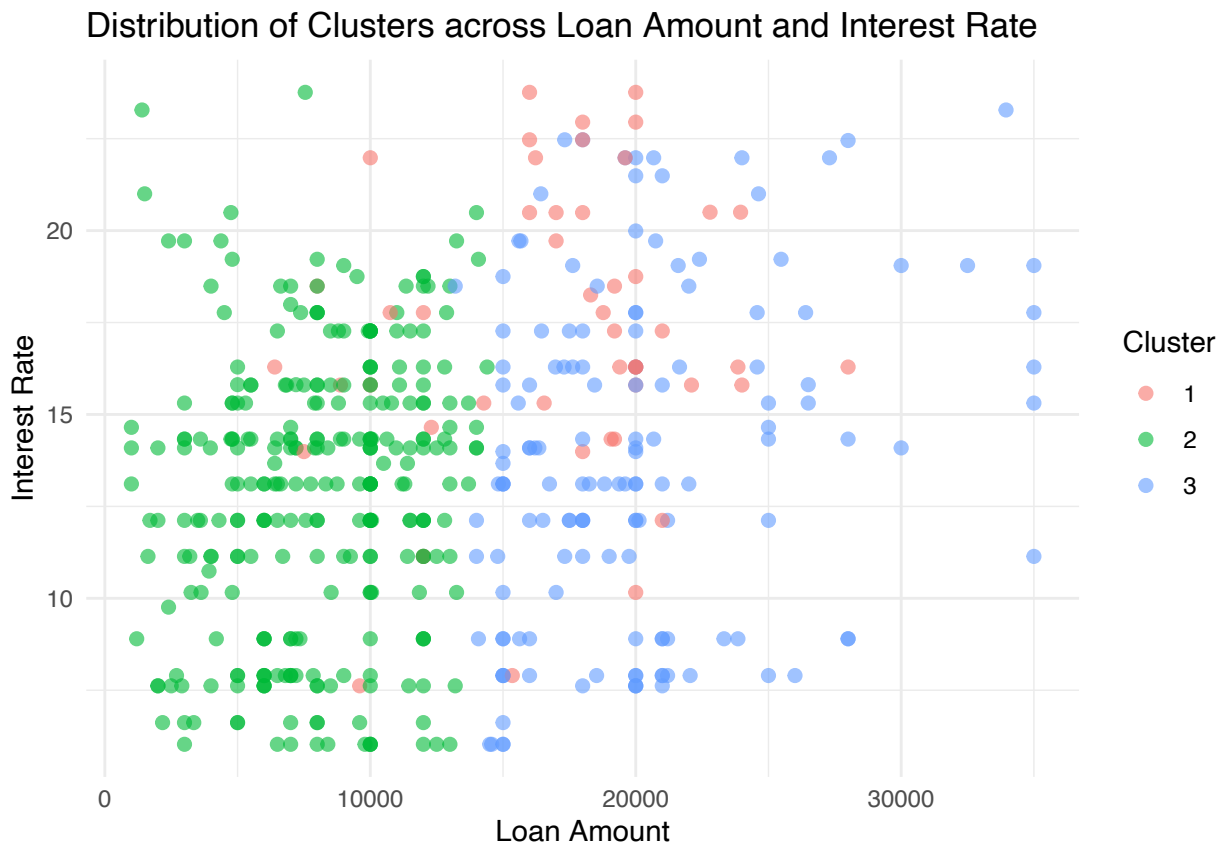
Cluster 1 - Loan amounts are in the mid range and interest rate is all over the place (low - high)

Cluster 2 - Loan amounts are in the lower range and interest rate is in the high range. (15%-20%)

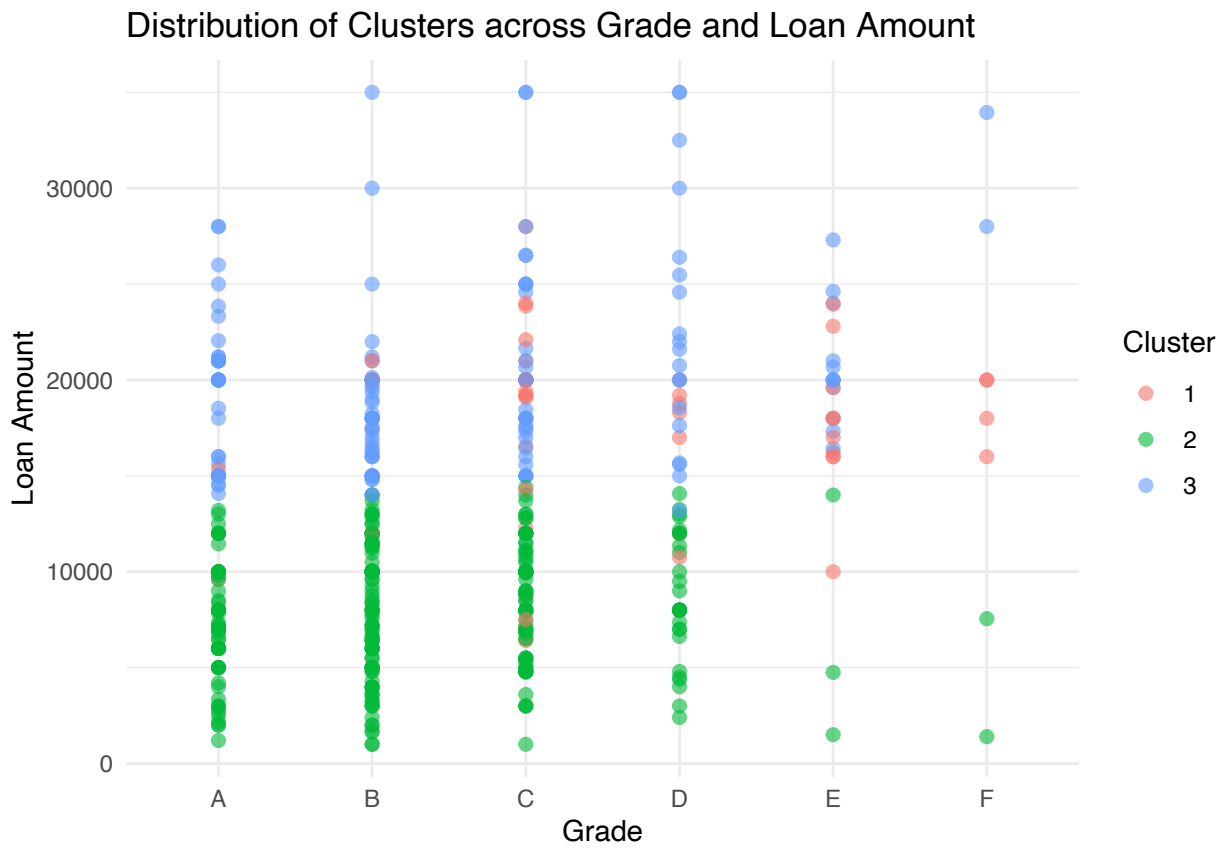
cluster 3 - Loan amounts are high and interest rate ranges from low to high

Customer loan proflie

```r
# Customer Loan Profile - Loan Amount & Interest Rate & Term
ggplot(final_data_with_categories,
       aes(x = loan_amnt, y = int_rate, color= as.factor(cluster))) +
  geom_point(alpha =0.6, size = 2 ) +
  labs(y = "Interest Rate", x = "Loan Amount", color = "Cluster", title = "Distribution of Clusters acr
  theme_minimal()
```

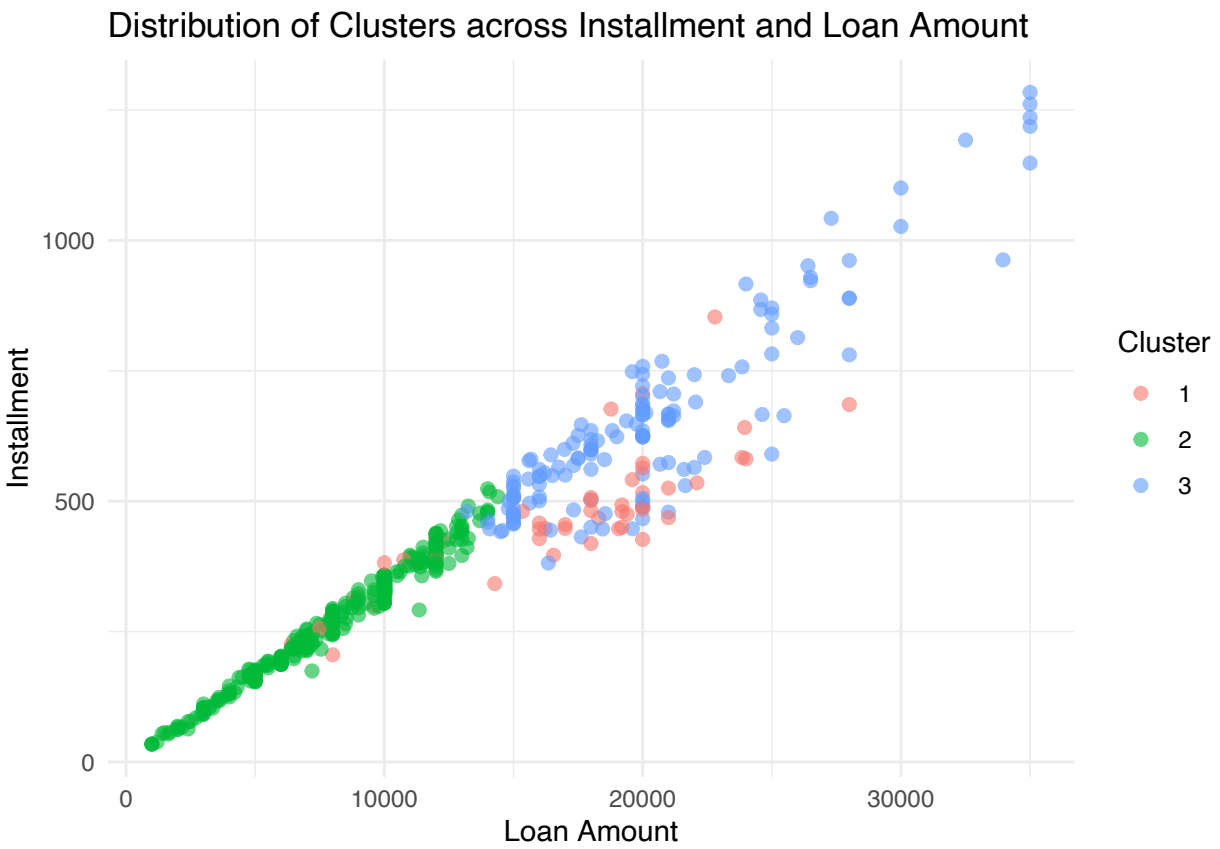



```
# Customer Loan Profile - Loan Amount & Grade
ggplot(final_data_with_categories,
       aes(y = loan_amnt, x = as.factor(grade), color= as.factor(cluster) )) +
  geom_point(alpha =0.6, size = 2  ) +
  labs(x = "Grade", y = "Loan Amount", color = "Cluster", title = "Distribution of Clusters across Grade
  theme_minimal()
```



Not much different between cluster in each grade, but it has characteristic in loan amount

```
# Customer Loan Profile - Loan Amount & Installment amount
ggplot(final_data_with_categories,
       aes(x = loan_amnt, y = installment, color= as.factor(cluster) )) +
  geom_point(alpha =0.6, size = 2  ) +
  labs(y = "Installment", x = "Loan Amount", color = "Cluster", title = "Distribution of Clusters across
  theme_minimal()
```

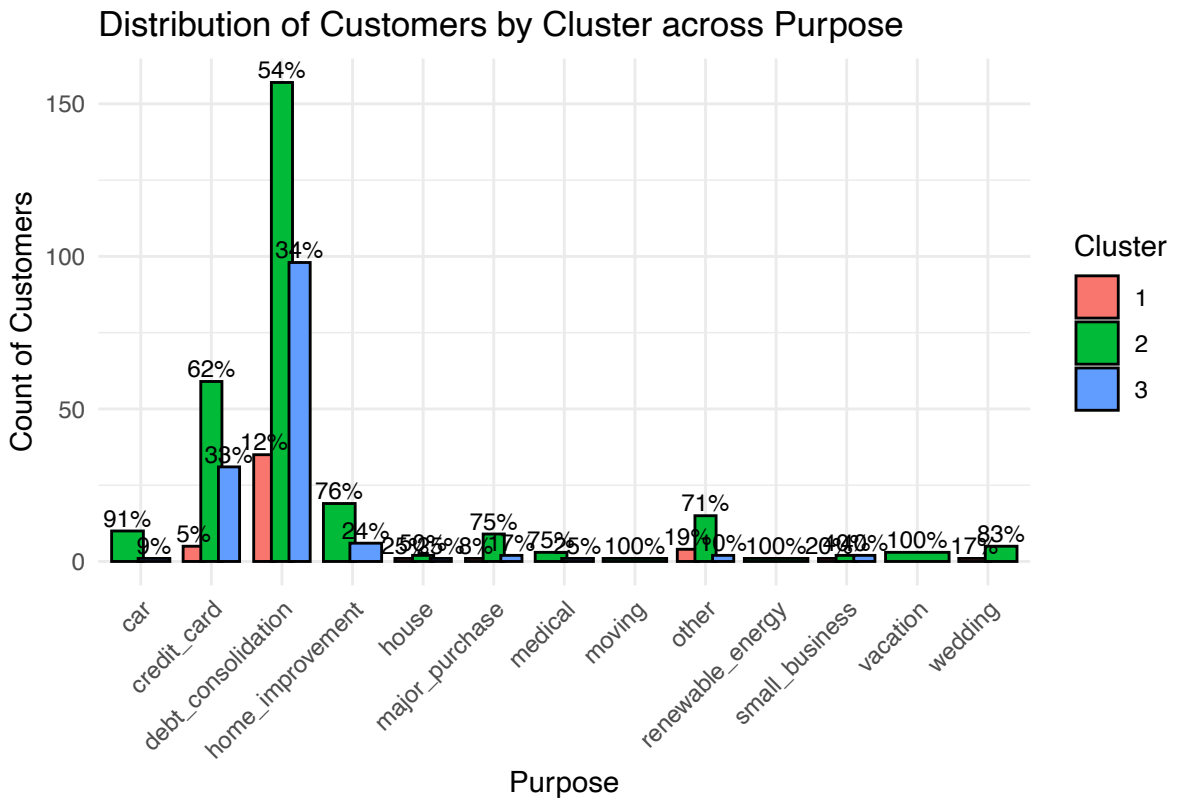


For loan_amount and Installment it's shown that cluster 1 is in the middle loan size and small installment, while cluster 2 is the small loan size and small paymenyt, last, cluster 3 is large loan size with high installment.

```
# Customer Loan Profile - Loan Amount & Annual Income & loan_is_bad status
ggplot(final_data_with_categories,
       aes(y = loan_amnt, x = annual_inc, color= as.factor(cluster), group=as.factor(loan_is_bad))) +
  geom_point(alpha = 0.6, size = 2, aes(shape = as.factor(loan_is_bad))) +
  scale_x_continuous(labels = label_number()) +  # Format x-axis labels as regular numbers
  labs(x = "Annual Income", y = "Loan Amount", color = "Cluster", shape = "Loan is Bad") +
  theme_minimal()
```


```r
# Customer Loan Profile - Purpose
summary_data <- final_data_with_categories %>%
  group_by(purpose, cluster) %>%
  summarise(count = n(), .groups = 'drop') %>%
  ungroup() %>%
  group_by(purpose) %>%
  mutate(total = sum(count),
         percentage = count / total * 100) %>%
  ungroup() # Ungroup if you're done with group-based calculations

# Step 2: Plot the bar chart and add the percentages as text
ggplot(summary_data, aes(x = purpose, y = count, fill = as.factor(cluster))) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.75), color = "black") +
  geom_text(aes(label = sprintf("%.0f%%", percentage), y = count),
            position = position_dodge(width = 0.75), vjust = -0.25, size = 3) +
  labs(x = "Purpose", y = "Count of Customers", fill = "Cluster", title ="Distribution of Customers by
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.margin = unit(c(1, 1, 1, 1), "lines"))
```



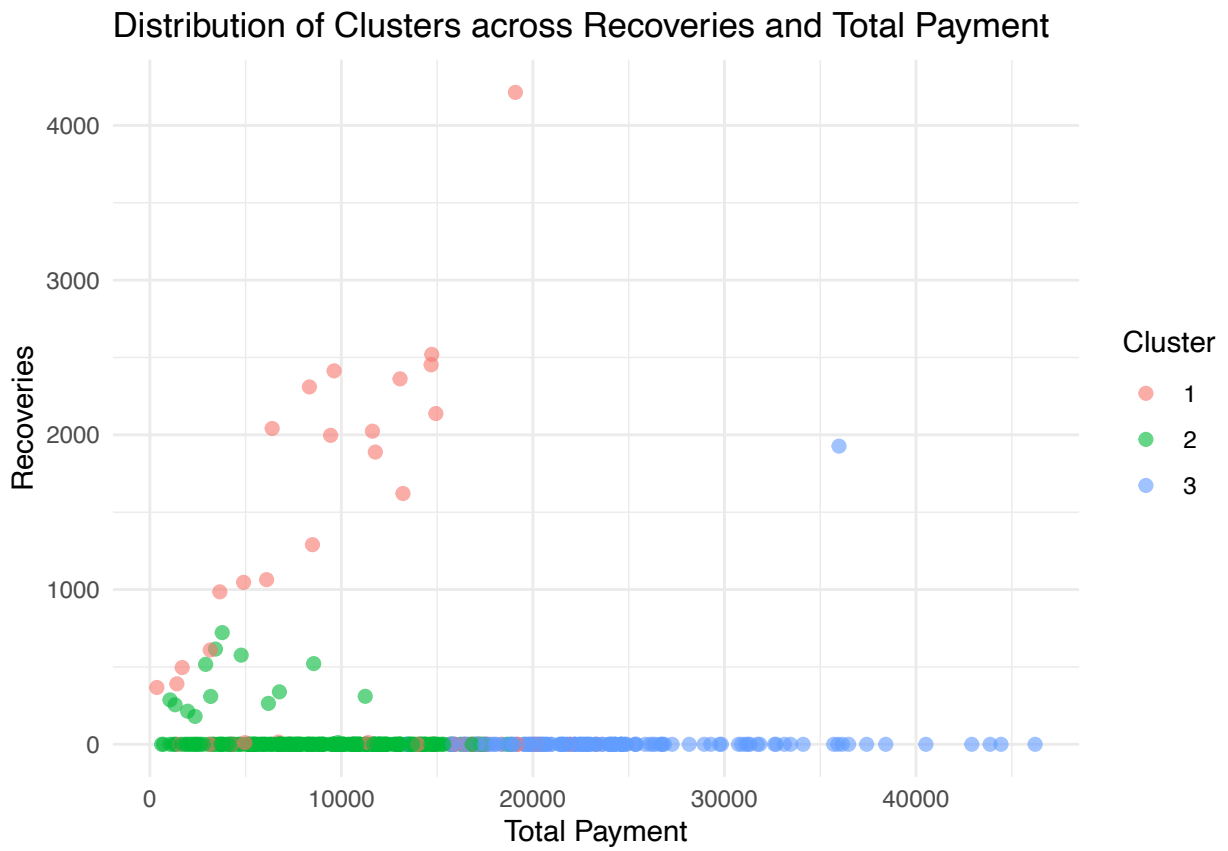

```
# change the bar chart
```

The cluster 1 and 3 show that they using loan for debt_consolidation and credit card ,while the group 2 spared across all purpose

Customer payment behaviour

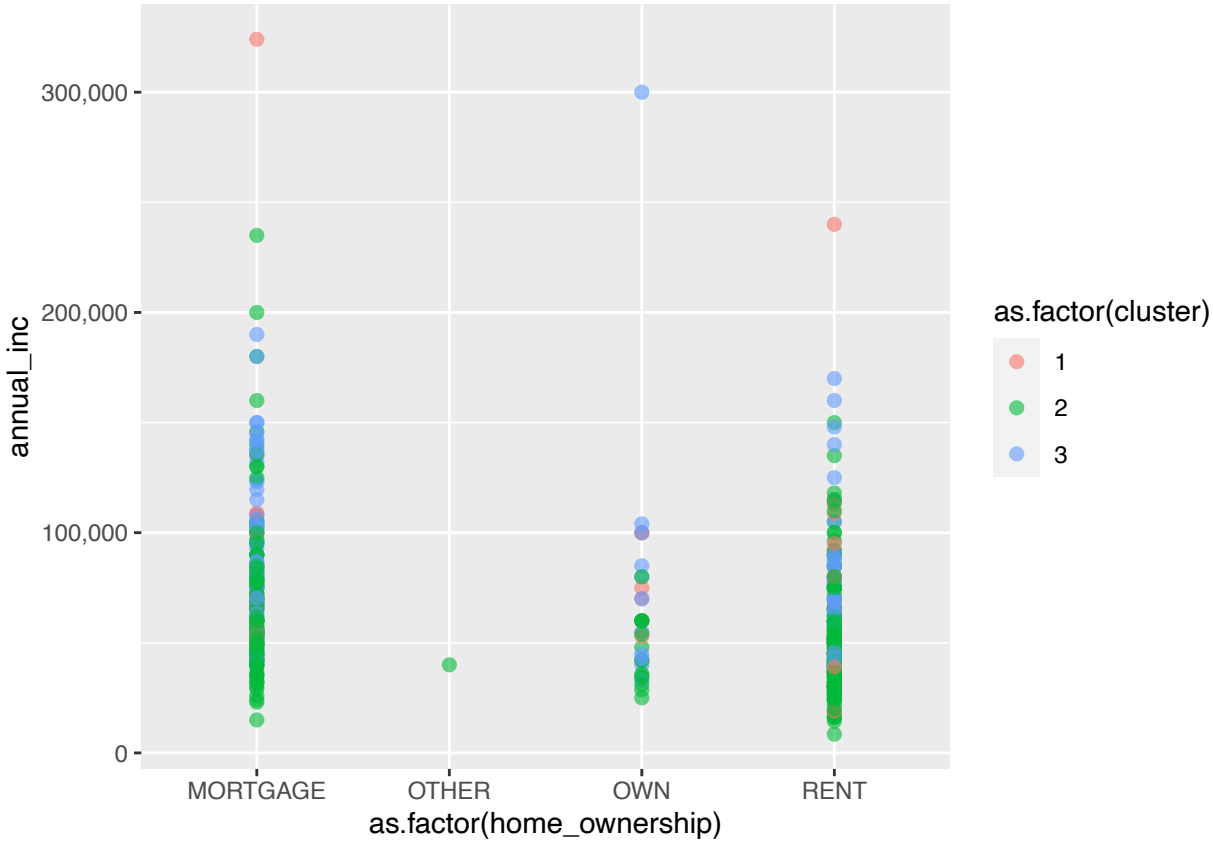
For cluster 1, there was no fully paid status, only for current and charged off, but for our sample, the number of in grace period until late is not have much observations to analyse. Moreover, the cluster 3 are the majority in Fully Paid, which can be conclude that the cluster 1 can be high risk loan, while cluster 3 is the loan with low risk of default.

```
# Customer Loan Profile - Recovery & total payment
ggplot(final_data_with_categories,
       aes(y = recoveries, x = total_pymnt, color= as.factor(cluster) )) +
  geom_point(alpha =0.6, size = 2  ) +
  labs(y = "Recoveries", x = "Total Payment", color = "Cluster", title = "Distribution of Clusters acros
  theme_minimal()
```



Customer Informations

```
# Customer Loan Profile - Loan Status & Total Payment
ggplot(final_data_with_categories,
       aes(y = annual_inc, x = as.factor(home_ownership), color= as.factor(cluster) )) +
  geom_point(alpha =0.6, size = 2  )+
  scale_y_continuous(labels = scales::comma)
```

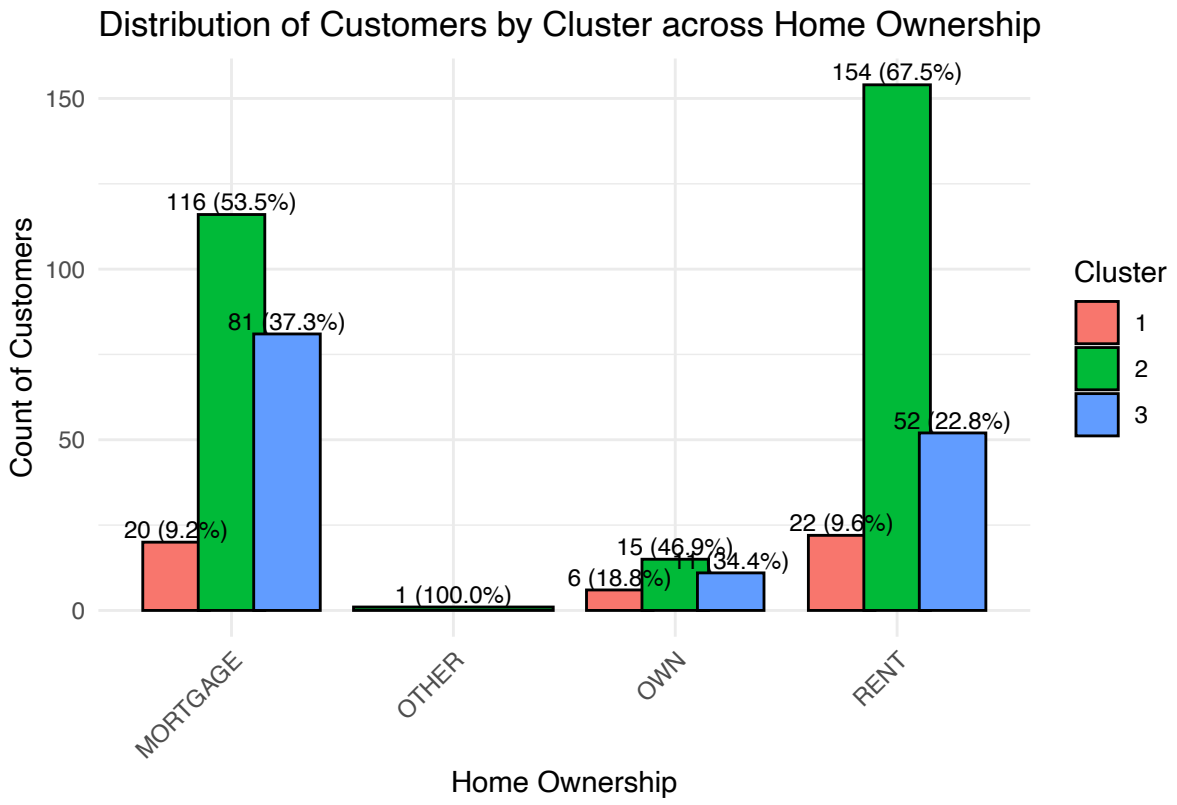



```
#try building a bar chart
```

Home Ownership by Clusters

```
summary_data <- final_data_with_categories %>%
  group_by(home_ownership, cluster) %>%
  summarise(count = n(), .groups = 'drop') %>%
  ungroup() %>%
  group_by(home_ownership) %>%
  mutate(total = sum(count),
         percentage = count / total * 100) %>%
  ungroup()

# Step 2: Plot the bar chart and add the percentages as text
ggplot(summary_data, aes(x = home_ownership, y = count, fill = as.factor(cluster))) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.75), color = "black") +
  geom_text(aes(label = sprintf("%.0f (%.1f%%)", count, percentage), y = count),
            position = position_dodge(width = 0.75), vjust = -0.25, size = 3)+
  labs(x = "Home Ownership", y = "Count of Customers", fill = "Cluster", title ="Distribution of Custome
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.margin = unit(c(1, 1, 1, 1), "lines"))
```



Home ownership and Employment length do not show the difference between cluster.

#Savvina

```
final_data_with_categories %>%
  group_by(cluster) %>%
  summarise(count_cluster = n())
```

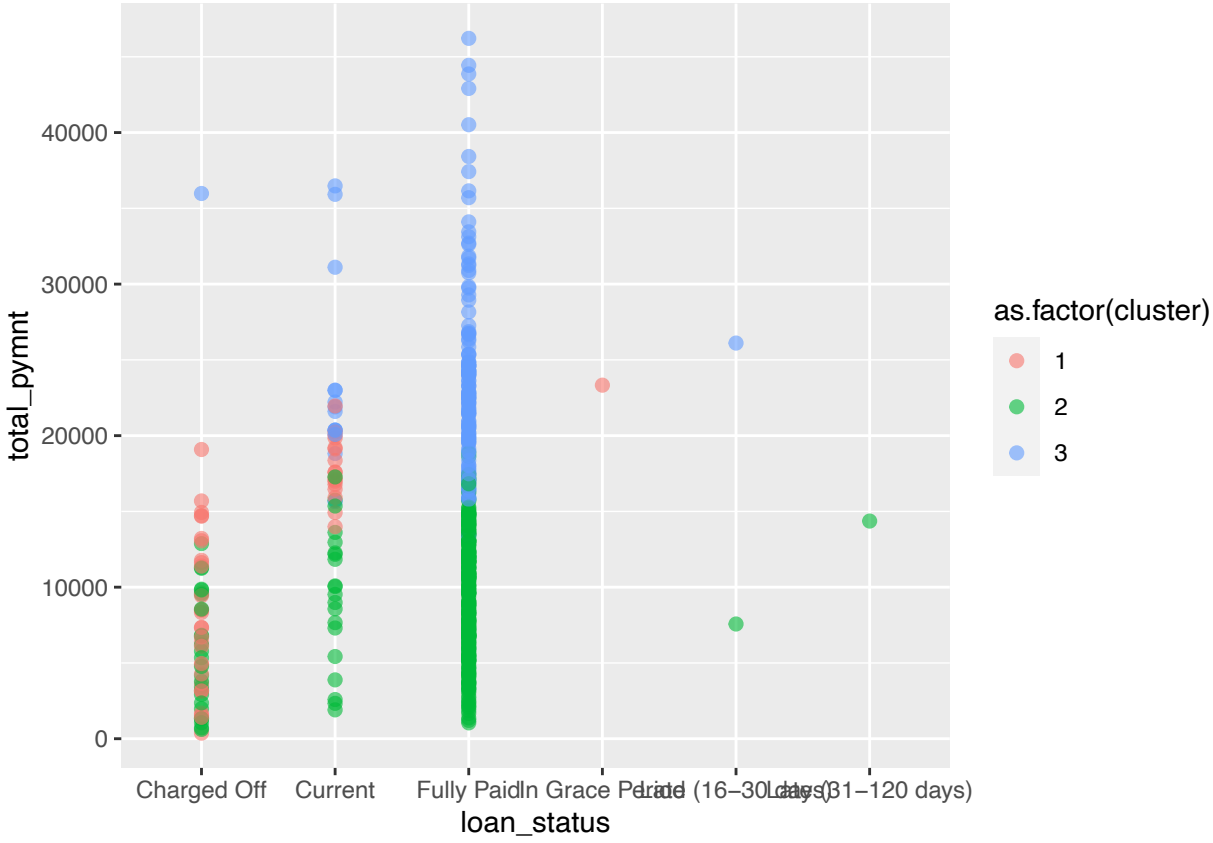
```
## # A tibble: 3 x 2
##   cluster count_cluster
##     <int>         <int>
## 1       1            48
## 2       2           286
## 3       3           144
```

```
#Calculate the number of bad and good loans and the percentage of bad loans in each cluster
final_data_with_categories %>%
  group_by(cluster) %>%
  summarise(count_false = sum(loan_is_bad == 0),
            count_true = sum(loan_is_bad == 1)) %>%
  mutate(percentage_false = count_false / (count_false + count_true) * 100,
         percentage_true = count_true / (count_false + count_true) * 100) %>%
  select(cluster, count_false, count_true, percentage_false, percentage_true)
```

```
## # A tibble: 3 x 5
##   cluster count_false count_true percentage_false percentage_true
```

```
##      <int>       <int>      <int>            <dbl>           <dbl>
## 1       1          17         31             35.4            64.6
## 2       2         259         27             90.6             9.44
## 3       3         142          2             98.6             1.39
```

```
# Customer Loan Profile - Loan Status & Total Payment
ggplot(final_data_with_categories,
       aes(y = total_pymnt, x = loan_status, color= as.factor(cluster) )) +
  geom_point(alpha =0.6, size = 2  )
```

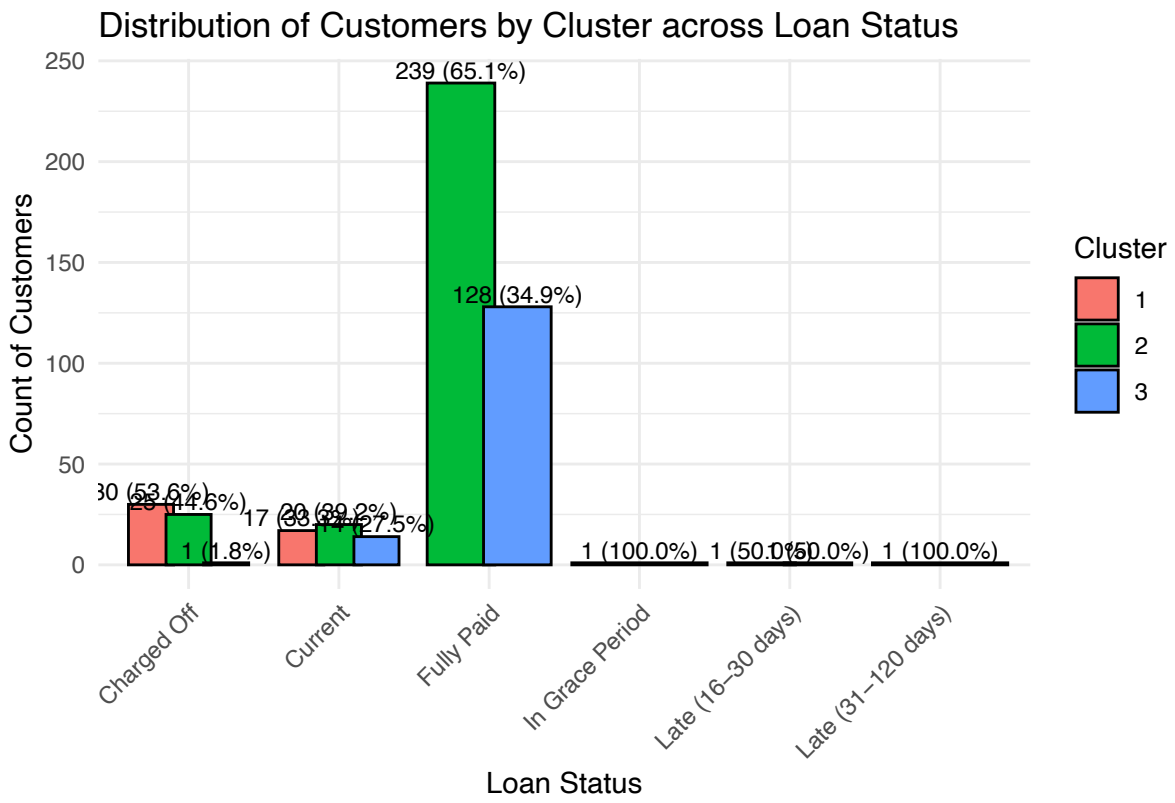


Loan Status by Clusters

```
summary_data <- final_data_with_categories %>%
  group_by(loan_status, cluster) %>%
  summarise(count = n(), .groups = 'drop') %>%
  ungroup() %>%
  group_by(loan_status) %>%
  mutate(total = sum(count),
         percentage = count / total * 100) %>%
  ungroup() # Ungroup if you're done with group-based calculations

# Step 2: Plot the bar chart and add the percentages as text
ggplot(summary_data, aes(x = loan_status, y = count, fill = as.factor(cluster))) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.75), color = "black") +
  geom_text(aes(label = sprintf("%.0f (%.1f%%)", count, percentage), y = count),
            position = position_dodge(width = 0.75), vjust = -0.25, size = 3) +
```

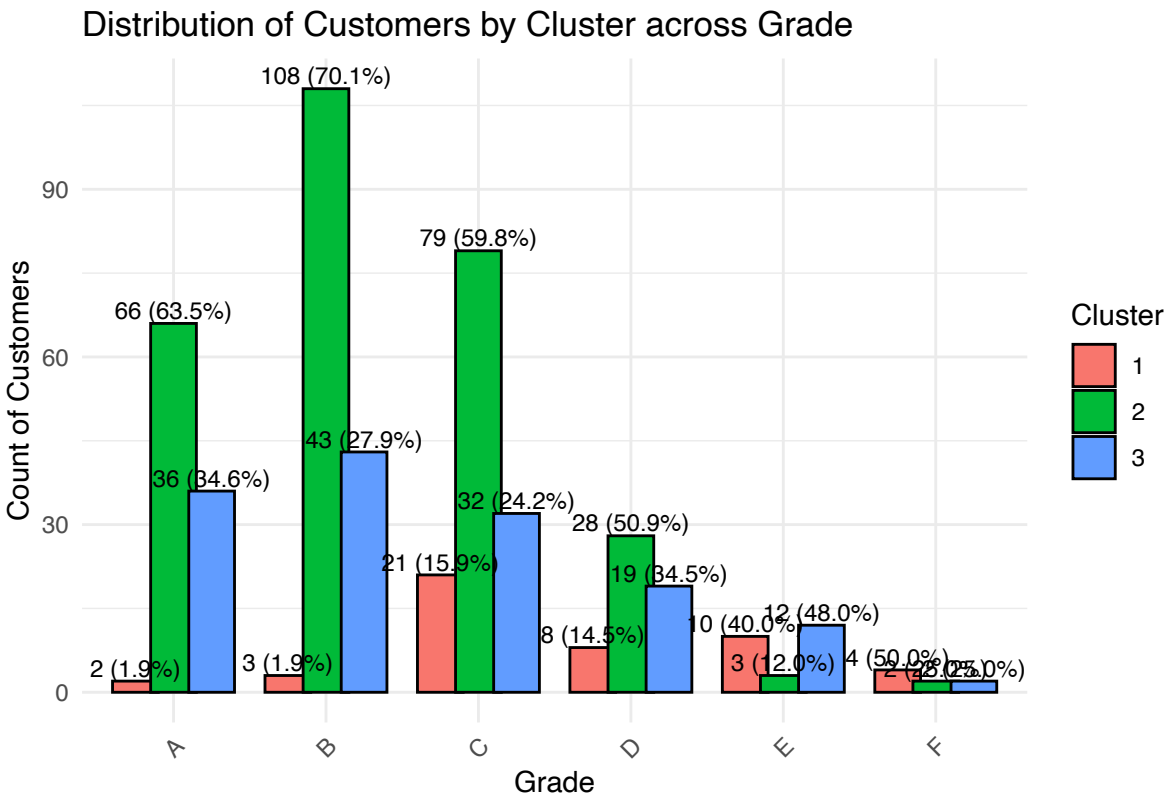
```
  labs(x = "Loan Status", y = "Count of Customers", fill = "Cluster", title ="Distribution of Customers
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.margin = unit(c(1, 1, 1, 1), "lines"))
```



Grade by Clusters

```
summary_data <- final_data_with_categories %>%
  group_by(grade, cluster) %>%
  summarise(count = n(), .groups = 'drop') %>%
  ungroup() %>%
  group_by(grade) %>%
  mutate(total = sum(count),
         percentage = count / total * 100) %>%
  ungroup() # Ungroup if you're done with group-based calculations

# Step 2: Plot the bar chart and add the percentages as text
ggplot(summary_data, aes(x = grade, y = count, fill = as.factor(cluster))) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.75), color = "black") +
  geom_text(aes(label = sprintf("%.0f (%.1f%%)", count, percentage), y = count),
            position = position_dodge(width = 0.75), vjust = -0.25, size = 3) +
  labs(x = "Grade", y = "Count of Customers", fill = "Cluster", title ="Distribution of Customers by Clu
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.margin = unit(c(1, 1, 1, 1), "lines"))
```


Distribution of Customers by Cluster across Grade
Count of Customers
Grade
Cluster
1
2
3
0
30
60
90
A
B
C
D
E
F
66 (63.5%)
36 (34.6%)
2 (1.9%)
108 (70.1%)
43 (27.9%)
3 (1.9%)
79 (59.8%)
32 (24.2%)
21 (15.9%)
28 (50.9%)
19 (34.5%)
8 (14.5%)
12 (48.0%)
10 (40.0%)
3 (12.0%)
4 (50.0%)