ADA Groupwork Assignment

2292213, 5528141, 5581250, 5582755, 5587165, 5588409, 5589718

2024-03-08

```
# import loan data as df
df <- read_excel("loan_data_ADA_assignment.xlsx")
```

```
summary(df)
```

```
##        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
##
##
##
##
```

1. Define the Problem

1.1. Objective

- Utilising cluster analysisrouping borrowers with similar characteristics
- Identify distinct borrower's segments
- Enabling personalised loan products, targeted marketing strategies, and a better customer support process to serve the unique needs of each segment

1.2. Select cluster variables

1. annual_inc: The self-reported annual income provided by the borrower during registration.
2. 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.
3. int_rate: Interest Rate on the loan
4. total_rec_int: Interest received to date
5. total_rec_prncp: Principal received to date

6. 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.
7. total_rec_late_fee: Late fees received to date
8. term : The entire period of the loan
9. home_ownership: The home ownership status provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER
10. grade : The grade of loan

```
# create new df with essential variables (10 variables)
df_ess <- df %>% select(annual_inc,
                        loan_amnt,
                        int_rate,
                        total_rec_int,
                        total_rec_prncp,
                        dti,
                        total_rec_late_fee,
                        term,
                        home_ownership,
                        grade)
```

2. Pre-Analysis Decision

2.1. Data checking (duplicate, missing values, outliers)

- Check duplicate value

```
# total row of data
df_row <- nrow(df_ess)

# distinct data
unique_row <- nrow(distinct(df_ess))

# verification. No duplicated data
duplicated <- df_row - unique_row
print(duplicated)
```

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

- Check missing value

```
missing_counts <- numeric(length(df_ess))

for (col in names(df_ess)) {
  missing_counts[col] <- sum(is.na(df_ess[[col]]))
}

missing_counts
```

```
##
##                   0                   0                   0                   0
##
##                   0                   0                   0                   0
##                                                  annual_inc          loan_amnt
##                   0                   0                   0                   0
##            int_rate      total_rec_int    total_rec_prncp                 dti
##                   0                   0                   0                   0
## total_rec_late_fee                term     home_ownership               grade
```

```
##                    0                    0                    0                    0
```

No missing value in cluster variables.

- Detect outliers using graphical method

```
df_summary <- dfSummary(df_ess)
filename <- "df_summary.html"
view(df_summary, file = filename)
```

```
summary(df_ess)
```

```
##    annual_inc        loan_amnt        int_rate     total_rec_int
##  Min.   :   5000   Min.   : 1000   Min.   : 6.00   Min.   :    0
##  1st Qu.:  45000   1st Qu.: 8000   1st Qu.:11.14   1st Qu.: 1058
##  Median :  60000   Median :12000   Median :14.09   Median : 2047
##  Mean   :  71317   Mean   :13901   Mean   :14.00   Mean   : 3071
##  3rd Qu.:  85000   3rd Qu.:19200   3rd Qu.:17.27   3rd Qu.: 3737
##  Max.   :7141778   Max.   :35000   Max.   :24.89   Max.   :22778
##  total_rec_prncp      dti        total_rec_late_fee      term
##  Min.   :    0   Min.   : 0.00   Min.   :  0.0000   Min.   :36.00
##  1st Qu.: 6000   1st Qu.:11.51   1st Qu.:  0.0000   1st Qu.:36.00
##  Median :10000   Median :17.16   Median :  0.0000   Median :36.00
##  Mean   :11611   Mean   :17.37   Mean   :  0.8419   Mean   :40.49
##  3rd Qu.:15479   3rd Qu.:23.05   3rd Qu.:  0.0000   3rd Qu.:36.00
##  Max.   :35000   Max.   :34.99   Max.   :286.7476   Max.   :60.00
##  home_ownership        grade
##  Length:50000       Length:50000
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

2.2. Data encoding

1) Home-ownership: recode to metric variables - sequence order by financial stability (1-highest, 5-lowest)

1. OWN -1
2. MORTGAGE-2
3. RENT -3
4. OTHER-4
5. NONE-5

2) grade : 1 - 7 (A - G) 1: Highest (A) 7: Lowest (G)

3) Term: 0 -> 36 (short-term) 1 -> 60 (long-term)

```
# Change into factor
df_ess$home_ownership <- as.factor(df_ess$home_ownership)
df_ess$grade <- as.factor(df_ess$grade)
df_ess$term <- as.factor(df_ess$term)

# Recode
df_ess$home_ownership <- revalue(df_ess$home_ownership,
                                 c('OWN' = 1, 'MORTGAGE' = 2, 'RENT' = 3, 'OTHER' = 4, 'NONE' = 5)
```

```r
df_ess$grade <- revalue(df_ess$grade,
                            c('A' = 1, 'B' = 2, 'C' = 3, 'D' = 4, 'E' = 5, 'F' = 6, 'G' = 7))

df_ess$term <- revalue(df_ess$term,
                            c('36' = 0, '60' = 1))
```

```r
summary(df_ess)
```

```
##    annual_inc        loan_amnt        int_rate     total_rec_int
##  Min.   :   5000   Min.   : 1000   Min.   : 6.00   Min.   :    0
##  1st Qu.:  45000   1st Qu.: 8000   1st Qu.:11.14   1st Qu.: 1058
##  Median :  60000   Median :12000   Median :14.09   Median : 2047
##  Mean   :  71317   Mean   :13901   Mean   :14.00   Mean   : 3071
##  3rd Qu.:  85000   3rd Qu.:19200   3rd Qu.:17.27   3rd Qu.: 3737
##  Max.   :7141778   Max.   :35000   Max.   :24.89   Max.   :22778
##
##  total_rec_prncp      dti        total_rec_late_fee term      home_ownership
##  Min.   :    0   Min.   : 0.00   Min.   :  0.0000   0:40641   2:24784
##  1st Qu.: 6000   1st Qu.:11.51   1st Qu.:  0.0000   1: 9359   5:   42
##  Median :10000   Median :17.16   Median :  0.0000             4:   45
##  Mean   :11611   Mean   :17.37   Mean   :  0.8419             1: 3836
##  3rd Qu.:15479   3rd Qu.:23.05   3rd Qu.:  0.0000             3:21293
##  Max.   :35000   Max.   :34.99   Max.   :286.7476
##
##  grade
##  1: 8533
##  2:17859
##  3:12171
##  4: 7065
##  5: 2926
##  6: 1227
##  7:  219
```

```r
# Convert all selected columns to numeric
loanDataFiltered <- apply(df_ess, 2, as.numeric)
loanDataFiltered_df <- as.data.frame(loanDataFiltered)
summary(loanDataFiltered_df)
```

```
##    annual_inc        loan_amnt        int_rate     total_rec_int
##  Min.   :   5000   Min.   : 1000   Min.   : 6.00   Min.   :    0
##  1st Qu.:  45000   1st Qu.: 8000   1st Qu.:11.14   1st Qu.: 1058
##  Median :  60000   Median :12000   Median :14.09   Median : 2047
##  Mean   :  71317   Mean   :13901   Mean   :14.00   Mean   : 3071
##  3rd Qu.:  85000   3rd Qu.:19200   3rd Qu.:17.27   3rd Qu.: 3737
##  Max.   :7141778   Max.   :35000   Max.   :24.89   Max.   :22778
##  total_rec_prncp      dti        total_rec_late_fee      term
##  Min.   :    0   Min.   : 0.00   Min.   :  0.0000   Min.   :0.0000
##  1st Qu.: 6000   1st Qu.:11.51   1st Qu.:  0.0000   1st Qu.:0.0000
##  Median :10000   Median :17.16   Median :  0.0000   Median :0.0000
##  Mean   :11611   Mean   :17.37   Mean   :  0.8419   Mean   :0.1872
##  3rd Qu.:15479   3rd Qu.:23.05   3rd Qu.:  0.0000   3rd Qu.:0.0000
##  Max.   :35000   Max.   :34.99   Max.   :286.7476   Max.   :1.0000
##  home_ownership      grade
##  Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.000
```

```
##  Median :2.000   Median :2.000
##  Mean   :2.353   Mean   :2.651
##  3rd Qu.:3.000   3rd Qu.:3.000
##  Max.   :5.000   Max.   :7.000
```

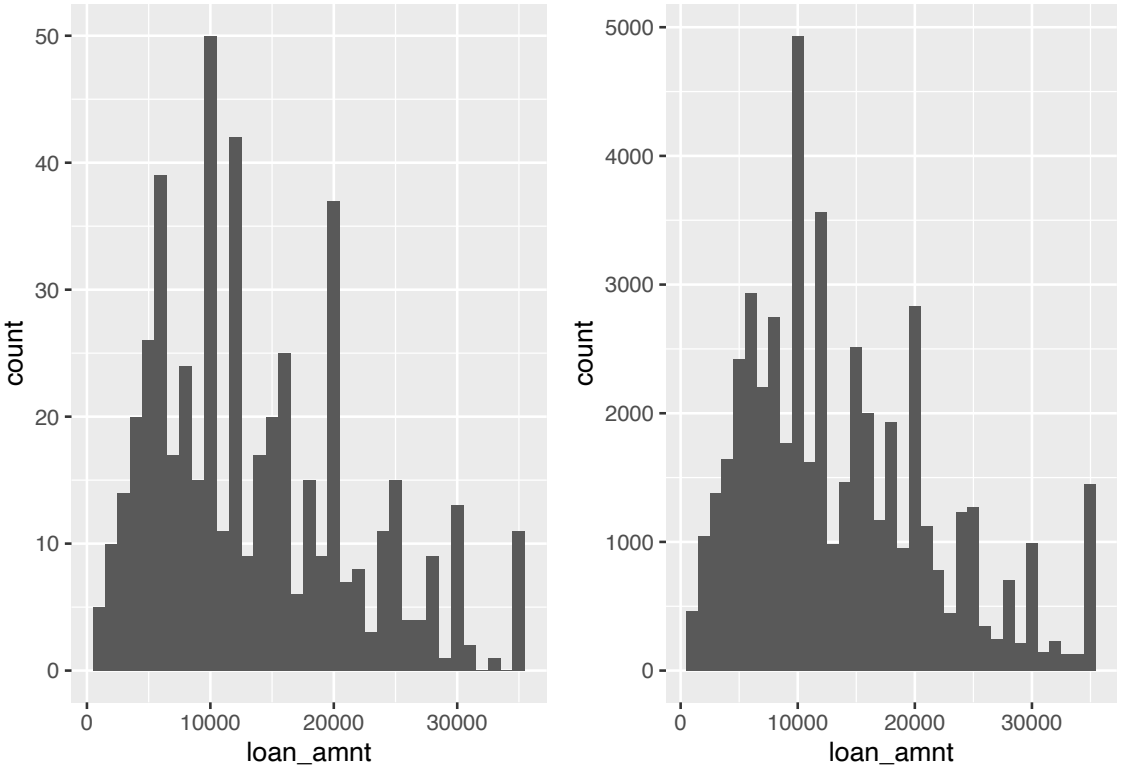
2.3. Sampling

```
set.seed(10)
sample <- sample_n(loanDataFiltered_df, 500)
```

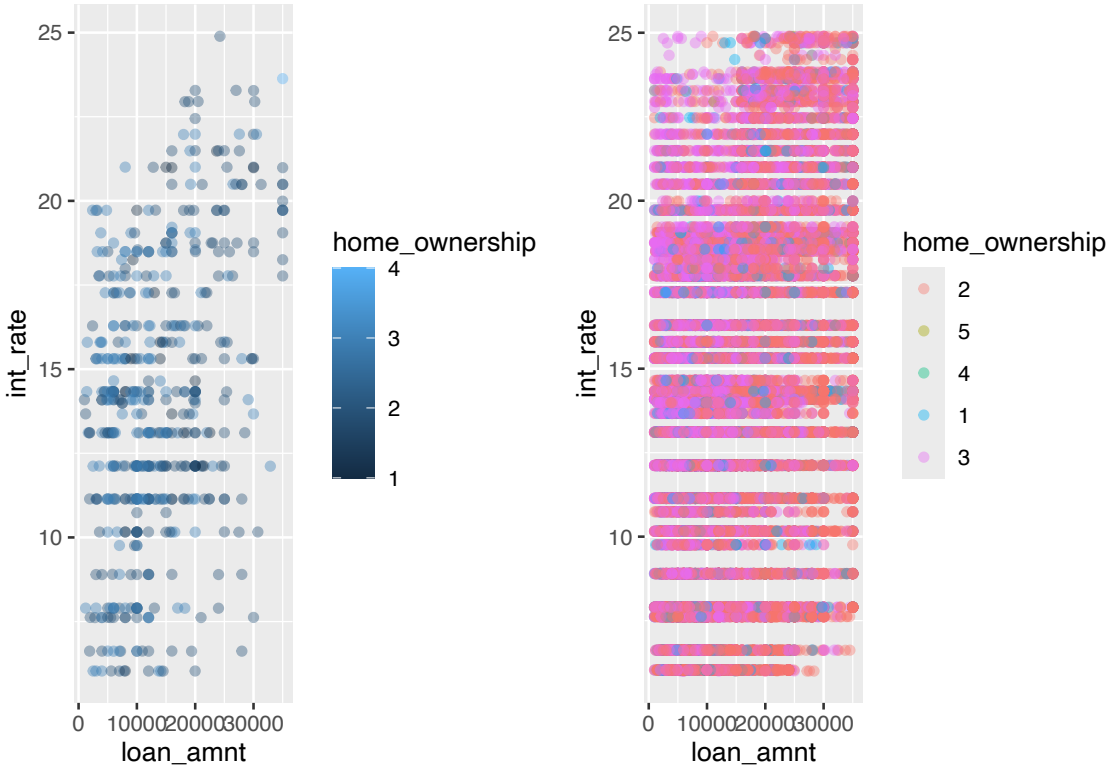
3. Check Assumptions

3.1. Data checking

```
# Distribution of loan amount of sample vs population
grid.arrange(ggplot(sample, aes(x = loan_amnt)) + geom_histogram(binwidth = 1000),
                        ggplot(df_ess, aes(x = loan_amnt)) + geom_histogram(binwidth = 1000),
                        ncol = 2)
```




```
# Scatter plot of interest rate and loan amount based on home ownership
grid.arrange(ggplot(sample, aes(x = loan_amnt, y = int_rate, color = home_ownership)) +
                          geom_point(position = "jitter", alpha = 0.4),
                        ggplot(df_ess, aes(x = loan_amnt, y = int_rate, color = home_ownership)) +
                          geom_point(position = "jitter", alpha = 0.4),
                        ncol = 2)
```



3.2. Multicollinearity: Pairwise Correlation

```
sampleMatrix<-cor(sample)
```

```
# round(sampleMatrix, 2)
```

```
lowerCor(sample)
```

```
##                   annl_ ln_mn int_r ttl_rc_n ttl_rc_p dti   tt___ term  hm_wn
## annual_inc         1.00
## loan_amnt          0.29  1.00
## int_rate           0.06  0.37  1.00
## total_rec_int      0.27  0.76  0.55  1.00
## total_rec_prncp    0.20  0.81  0.17  0.52     1.00
## dti               -0.22  0.05  0.13  0.06     0.03     1.00
## total_rec_late_fee 0.02  0.10  0.03  0.09     0.11     0.01  1.00
## term               0.15  0.50  0.49  0.60     0.22     0.02 -0.05  1.00
## home_ownership    -0.07 -0.21 -0.01 -0.16    -0.20     0.03  0.03 -0.15  1.00
## grade              0.07  0.39  0.96  0.56     0.20     0.11  0.04  0.51 -0.02
##                   grade
## annual_inc
## loan_amnt
## int_rate
## total_rec_int
## total_rec_prncp
## dti
## total_rec_late_fee
```

```
## term
## home_ownership
## grade               1.00
```

3.3. Multicollinearity: Kaiser-Meyer-Olkin (KMO)

```
KMO(sample)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = sample)
## Overall MSA =  0.71
## MSA for each item =
##         annual_inc          loan_amnt           int_rate      total_rec_int
##               0.75               0.67               0.65               0.86
##    total_rec_prncp                dti total_rec_late_fee               term
##               0.61               0.55               0.60               0.83
##     home_ownership              grade
##               0.88               0.65
```

Kaiser-Meyer-Olkin (KMO) test is a standard to assess the suitability of a data set for factor analysis. We are looking for a KMO value of 0.5 or more. Here it is 0.71, so this is good.

3.4. Multicollinearity: Bartlett's test

```
cortest.bartlett(sample)
```

```
## R was not square, finding R from data
```

```
## $chisq
## [1] 2917.021
##
## $p.value
## [1] 0
##
## $df
## [1] 45
```

P-value is 0, which means there is a sufficient correlation between variables. We can use factor analysis in this case.

4. Perform Principal Component Analysis (PCA)

```
pcModel<-principal(sample, 5, rotate="none", weights=TRUE, scores=TRUE)
print(pcModel)
```

```
## Principal Components Analysis
## Call: principal(r = sample, nfactors = 5, rotate = "none", scores = TRUE,
##     weights = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##                      PC1   PC2   PC3   PC4   PC5   h2    u2 com
## annual_inc          0.30 -0.46 -0.51  0.22  0.19 0.64 0.361 3.4
## loan_amnt           0.84 -0.39  0.19 -0.01  0.16 0.92 0.081 1.6
## int_rate            0.75  0.56 -0.14  0.08 -0.10 0.92 0.083 2.0
## total_rec_int       0.88 -0.09  0.01  0.02  0.05 0.79 0.212 1.0
## total_rec_prncp     0.63 -0.53  0.34  0.01  0.21 0.83 0.167 2.8
```

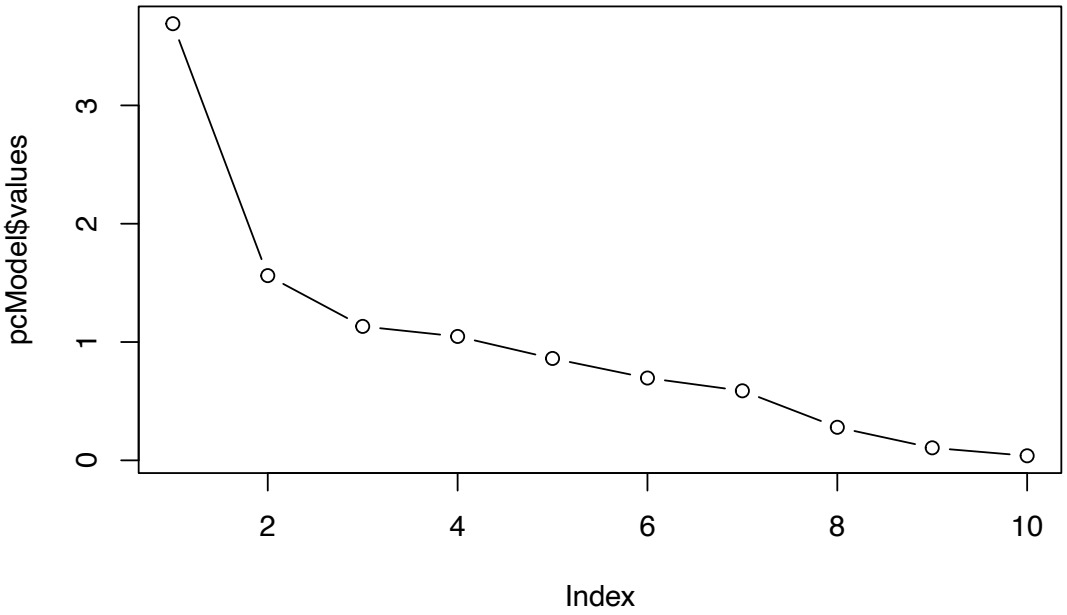
```
## dti                 0.09  0.37  0.72 -0.20  0.24 0.76 0.244 2.0
## total_rec_late_fee  0.09 -0.12  0.37  0.77 -0.47 0.98 0.025 2.3
## term                0.71  0.14 -0.19 -0.17 -0.06 0.60 0.402 1.4
## home_ownership     -0.22  0.36 -0.07  0.57  0.67 0.95 0.048 2.8
## grade               0.77  0.54 -0.14  0.08 -0.11 0.92 0.085 1.9
##
##                        PC1  PC2  PC3  PC4  PC5
## SS loadings           3.69 1.56 1.13 1.05 0.86
## Proportion Var        0.37 0.16 0.11 0.10 0.09
## Cumulative Var        0.37 0.53 0.64 0.74 0.83
## Proportion Explained  0.45 0.19 0.14 0.13 0.10
## Cumulative Proportion 0.45 0.63 0.77 0.90 1.00
##
## Mean item complexity =  2.1
## 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  236.16  with prob <  5.1e-49
##
## Fit based upon off diagonal values = 0.95
```

```
print.psych(pcModel, cut=0.3, sort=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = sample, nfactors = 5, rotate = "none", scores = TRUE,
##     weights = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item   PC1   PC2   PC3   PC4   PC5   h2    u2 com
## total_rec_int          4  0.88                       0.79 0.212 1.0
## loan_amnt              2  0.84 -0.39                 0.92 0.081 1.6
## grade                 10  0.77  0.54                 0.92 0.085 1.9
## int_rate               3  0.75  0.56                 0.92 0.083 2.0
## term                   8  0.71                       0.60 0.402 1.4
## total_rec_prncp        5  0.63 -0.53  0.34           0.83 0.167 2.8
## annual_inc             1       -0.46 -0.51           0.64 0.361 3.4
## dti                    6        0.37  0.72           0.76 0.244 2.0
## total_rec_late_fee     7              0.37  0.77 -0.47 0.98 0.025 2.3
## home_ownership         9        0.36        0.57  0.67 0.95 0.048 2.8
##
##                        PC1  PC2  PC3  PC4  PC5
## SS loadings           3.69 1.56 1.13 1.05 0.86
## Proportion Var        0.37 0.16 0.11 0.10 0.09
## Cumulative Var        0.37 0.53 0.64 0.74 0.83
## Proportion Explained  0.45 0.19 0.14 0.13 0.10
## Cumulative Proportion 0.45 0.63 0.77 0.90 1.00
##
## Mean item complexity =  2.1
## 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  236.16  with prob <  5.1e-49
##
## Fit based upon off diagonal values = 0.95
```

To produce the scree plot

```
plot(pcModel$values,type="b")
```




```
pcModel$weights
```

```
##                           PC1         PC2         PC3          PC4         PC5
## annual_inc          0.08077978 -0.29386453 -0.44702185  0.211336565  0.21505306
## loan_amnt           0.22724608 -0.25017527  0.16852750 -0.012608901  0.19006420
## int_rate            0.20388818  0.35981820 -0.12090356  0.075981427 -0.11638677
## total_rec_int       0.23875150 -0.05686243  0.01128327  0.016732186  0.06257646
## total_rec_prncp     0.16944389 -0.34076013  0.30067456  0.005422551  0.24069513
## dti                 0.02405031  0.24000061  0.63163640 -0.192318501  0.27421682
## total_rec_late_fee  0.02435222 -0.07721672  0.32343541  0.735819109 -0.55046943
## term                0.19347312  0.08751756 -0.16911285 -0.164221480 -0.06470958
## home_ownership     -0.05908698  0.23224573 -0.05769586  0.541198712  0.77708951
## grade               0.20839761  0.34281299 -0.11965181  0.076032355 -0.13105886
```

We can then access these scores by using

```
head(pcModel$scores, 10)
```

```
##              PC1        PC2        PC3         PC4        PC5
##  [1,] -0.9362977  0.5689859 -0.4954174  0.46310447  0.3189131
##  [2,] -0.4007959 -1.0803219 -0.1582872 -1.95481543 -1.9494720
##  [3,] -0.9696421 -1.4644904 -0.5948107  0.62450428  1.1336938
##  [4,] -0.3813357  0.5408023 -0.5210639  0.59892546  0.3624829
##  [5,] -0.4367595  0.5162278  1.1664280  0.03463419  1.3807775
##  [6,]  1.2747127 -1.3723158  1.0637235 -0.31856468  0.6282800
##  [7,] -0.2453597 -0.4118408  1.0225689  0.24853833  1.7307205
##  [8,] -0.6120398  0.3670978  0.9451526  0.10681232  1.1682691
##  [9,] -0.1046778 -0.2993457  0.6658561  0.38867458  1.5204344
## [10,] -0.6689182  1.3380139 -0.1416283  0.39483528  0.3665061
```

We can use the principal component scores for further analysis, before doing that we need to add them into our dataframe:

```
sample_pca <- cbind(sample, pcModel$scores)
```

```
summary(sample_pca)
```

```
##    annual_inc        loan_amnt        int_rate     total_rec_int
##  Min.   :  14000   Min.   : 1000   Min.   : 6.03   Min.   :   59.7
##  1st Qu.:  45000   1st Qu.: 7200   1st Qu.:11.14   1st Qu.:  998.4
##  Median :  60000   Median :12000   Median :14.09   Median : 1970.6
##  Mean   :  73102   Mean   :13662   Mean   :14.03   Mean   : 2982.6
##  3rd Qu.:  85000   3rd Qu.:19600   3rd Qu.:17.27   3rd Qu.: 3813.5
##  Max.   :1250000   Max.   :35000   Max.   :24.89   Max.   :17175.1
##  total_rec_prncp        dti        total_rec_late_fee      term
##  Min.   :  265.9   Min.   : 1.02   Min.   : 0.0000    Min.   :0.000
##  1st Qu.: 5971.4   1st Qu.:11.22   1st Qu.: 0.0000    1st Qu.:0.000
##  Median :10000.0   Median :16.48   Median : 0.0000    Median :0.000
##  Mean   :11545.0   Mean   :17.02   Mean   : 0.9973    Mean   :0.186
##  3rd Qu.:16000.0   3rd Qu.:23.02   3rd Qu.: 0.0000    3rd Qu.:0.000
##  Max.   :35000.0   Max.   :34.88   Max.   :64.3400    Max.   :1.000
##  home_ownership      grade            PC1                PC2
##  Min.   :1.000   Min.   :1.000   Min.   :-1.6576   Min.   :-4.815475
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:-0.6791   1st Qu.:-0.638184
##  Median :2.000   Median :2.000   Median :-0.2757   Median : 0.002815
##  Mean   :2.368   Mean   :2.676   Mean   : 0.0000   Mean   : 0.000000
##  3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.: 0.4054   3rd Qu.: 0.741524
##  Max.   :4.000   Max.   :7.000   Max.   : 3.7382   Max.   : 2.345764
##       PC3                PC4                PC5
##  Min.   :-8.30745   Min.   :-1.9548   Min.   :-6.5452
##  1st Qu.:-0.60411   1st Qu.:-0.5538   1st Qu.:-0.5323
##  Median :-0.04166   Median :-0.1339   Median : 0.1074
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.56248   3rd Qu.: 0.4210   3rd Qu.: 0.6567
##  Max.   : 4.22046   Max.   : 7.6188   Max.   : 4.1670
```

5. Perform Factor Analysis

Two Factors Solution with Orthogonal rotation (Varimax Rotation)

```
fa3v<-(fa(sample,2, n.obs=500, rotate="varimax", fm="pa"))
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect.  Try a
## different factor score estimation method.
```

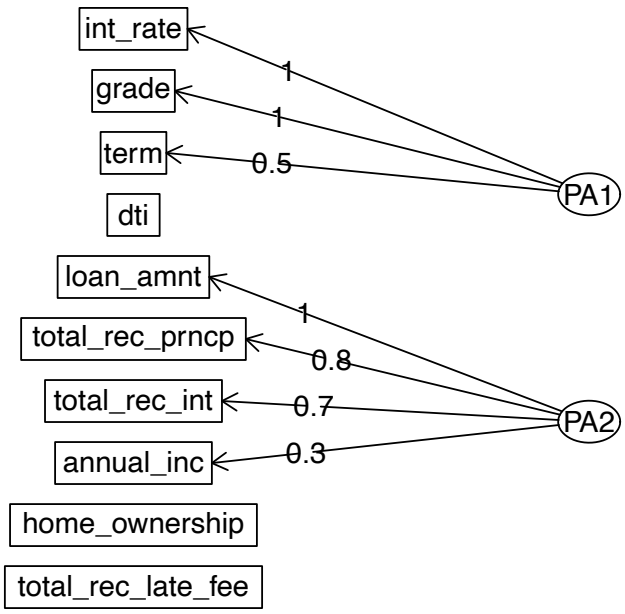
```
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected.  Examine the results carefully
```

```
print.psych(fa3v, cut=0.3,sort="TRUE")
```

```
## Factor Analysis using method =  pa
## Call: fa(r = sample, nfactors = 2, n.obs = 500, rotate = "varimax",
##     fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                    item  PA1   PA2     h2     u2 com
## int_rate              3 0.97       0.9538  0.046 1.0
## grade                10 0.95       0.9310  0.069 1.1
## term                  8 0.48  0.39 0.3831  0.617 1.9
```

```
## dti                   6            0.0178  0.982 1.1
## loan_amnt             2       1.00 1.0487 -0.049 1.1
## total_rec_prncp       5       0.75 0.5704  0.430 1.0
## total_rec_int         4 0.50  0.68 0.7128  0.287 1.8
## annual_inc            1       0.30 0.0925  0.908 1.0
## home_ownership        9            0.0587  0.941 1.0
## total_rec_late_fee    7            0.0087  0.991 1.0
##
##                        PA1  PA2
## SS loadings           2.41 2.37
## Proportion Var        0.24 0.24
## Cumulative Var        0.24 0.48
## Proportion Explained  0.50 0.50
## Cumulative Proportion 0.50 1.00
##
## Mean item complexity =  1.2
## Test of the hypothesis that 2 factors are sufficient.
##
## df null model =  45  with the objective function =  5.89 with Chi Square =  2917.02
## df of  the model are 26  and the objective function was  0.43
##
## The root mean square of the residuals (RMSR) is  0.05
## The df corrected root mean square of the residuals is  0.06
##
## The harmonic n.obs is  500 with the empirical chi square  100.75  with prob <  9.6e-11
## The total n.obs was  500  with Likelihood Chi Square =  211.73  with prob <  4.9e-31
##
## Tucker Lewis Index of factoring reliability =  0.888
## RMSEA index =  0.12  and the 90 % confidence intervals are  0.105 0.135
## BIC =  50.15
## Fit based upon off diagonal values = 0.98
```

```
fa.diagram(fa3v)
```



Once we have decided best solution, we can set scores to regression.

```
fa3v<-(fa(sample,2, n.obs=500, rotate="varimax", fm="pa", scores="regression"))
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect.  Try a
## different factor score estimation method.
```

```
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected.  Examine the results carefully
```

```
head(fa3v$scores, 10)
```

```
##                PA1        PA2
##  [1,] -0.18085623 -1.1381673
##  [2,] -1.25971881 -0.3675690
##  [3,] -1.84634926  0.5655897
##  [4,]  0.19832737 -0.5813052
##  [5,] -0.73987157  1.5818343
##  [6,] -0.03012164  2.0481682
##  [7,] -0.90231757  0.5370203
##  [8,] -0.61937263 -0.1590261
##  [9,] -0.45304492  0.4046091
## [10,]  0.48182208 -1.4589098
```

We can use the factor scores for further analysis, before doing that we need to add them into our dataframe:

```
sample_fa3 <- fa3v$scores
```

```
summary(sample_fa3)
```

```
##       PA1                PA2
##  Min.   :-2.0672   Min.   :-1.9379
```

```
##  1st Qu.:-0.6513   1st Qu.:-0.8054
##  Median :-0.1008   Median :-0.1588
##  Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.6534   3rd Qu.: 0.7085
##  Max.   : 2.5941   Max.   : 3.4486
```

6. Create Clusters

1) Check outliers again after selecting 3 factors

- Calculate Mahalanobis distance to identify potential outliers

```
Maha <- mahalanobis(sample_fa3,colMeans(sample_fa3),cov(sample_fa3))
```

```
MahaPvalue <-pchisq(Maha,df=10,lower.tail = FALSE)
# print(MahaPvalue)
print(sum(MahaPvalue<0.001)) # no outlier
```

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

2) Check multicollinearity again after selecting 3 factors

```
SampleMatrix<-cor(sample_fa3)
```

```
#round(sample_fa3, 2)
```

```
lowerCor(sample_fa3)
```

```
##     PA1   PA2
## PA1  1.00
## PA2 -0.01  1.00
```

There is no substantial number of correlations > 0.3

6.1. Standardise/normalise data

```
# Standardize data set
sample_s <- scale(sample_fa3)

# View the standardized data
head(sample_s)
```

```
##             PA1        PA2
## [1,] -0.1837921 -1.0875018
## [2,] -1.2801680 -0.3512067
## [3,] -1.8763213  0.5404125
## [4,]  0.2015468 -0.5554284
## [5,] -0.7518820  1.5114189
## [6,] -0.0306106  1.9569939
```

6.2. Select clustering method

- **Find the Linkage Method to Use**
- Define linkage methods

```
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")
```

- Function to compute agglomerative coefficient

```
ac <- function(x) {
  agnes(sample_s, method = x)$ac
}
```

- Calculate agglomerative coefficient for each clustering linkage method

```
sapply(m, ac)
```

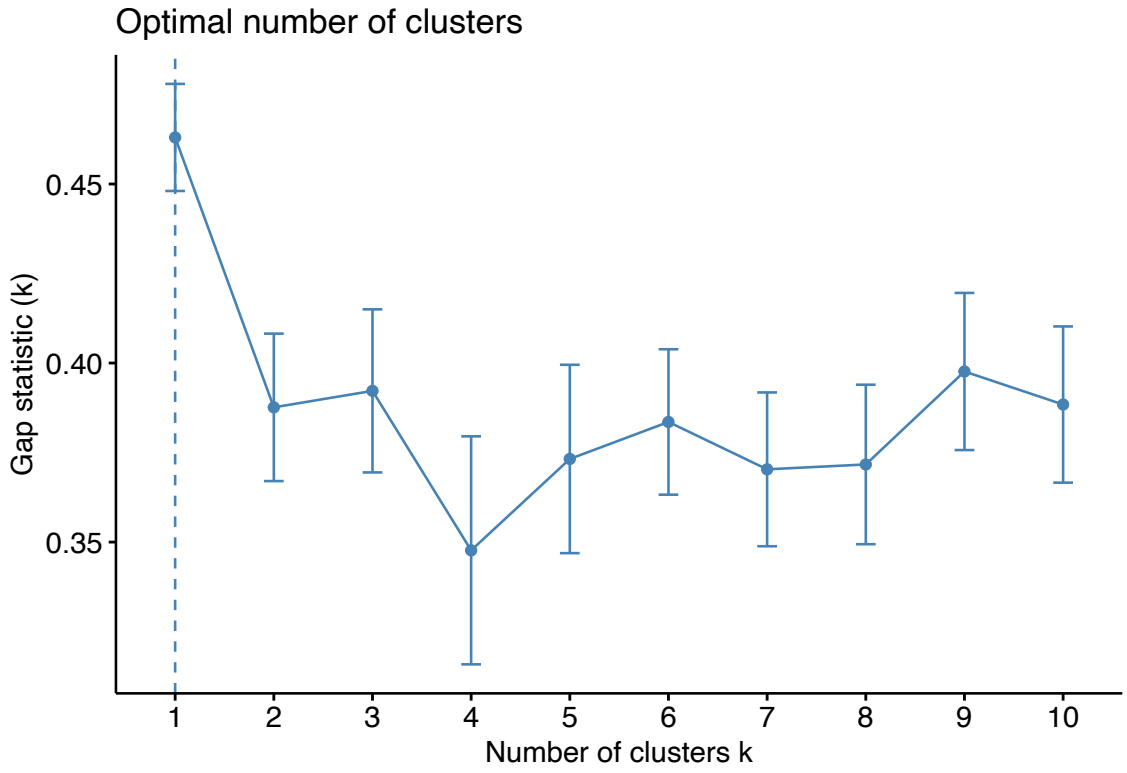
```
##   average    single  complete      ward
## 0.9643149 0.8638402 0.9834904 0.9961041
```

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

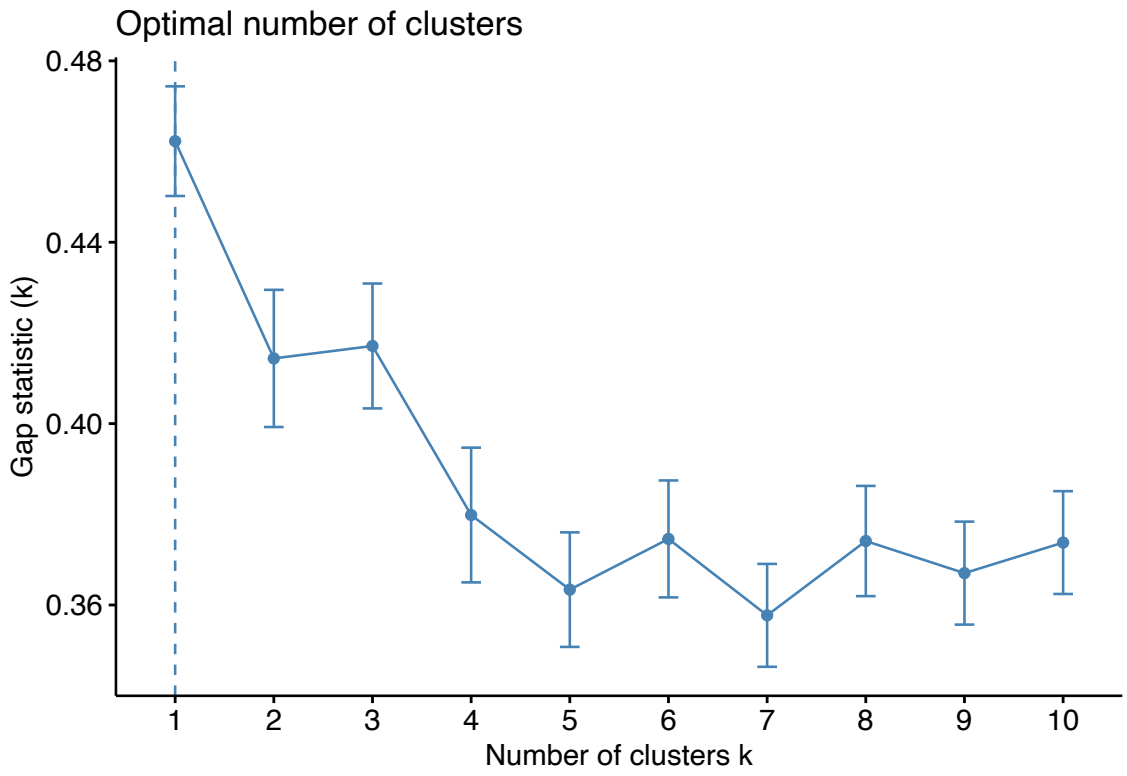
```
gap_stat_h_fa3 <- clusGap(sample_s, FUN = hcut, nstart = 25, K.max = 10, B = 50)
gap_stat_k_fa3 <- clusGap(sample_s, FUN = kmeans, nstart = 25, K.max = 10, B = 50)
```

- Produce plot of optimal number of clusters using gap statistic method

```
fviz_gap_stat(gap_stat_h_fa3) # 3 clusters for hierarchical
```




```
fviz_gap_stat(gap_stat_k_fa3) # 3 clusters for Non-hierarchical (Kmeans)
```



- Finding distance matrix

```
#using Euclidean distance
distance_mat_fa3 <- dist(sample_s, method = 'euclidean')
```

7. Comparing results & choosing solution

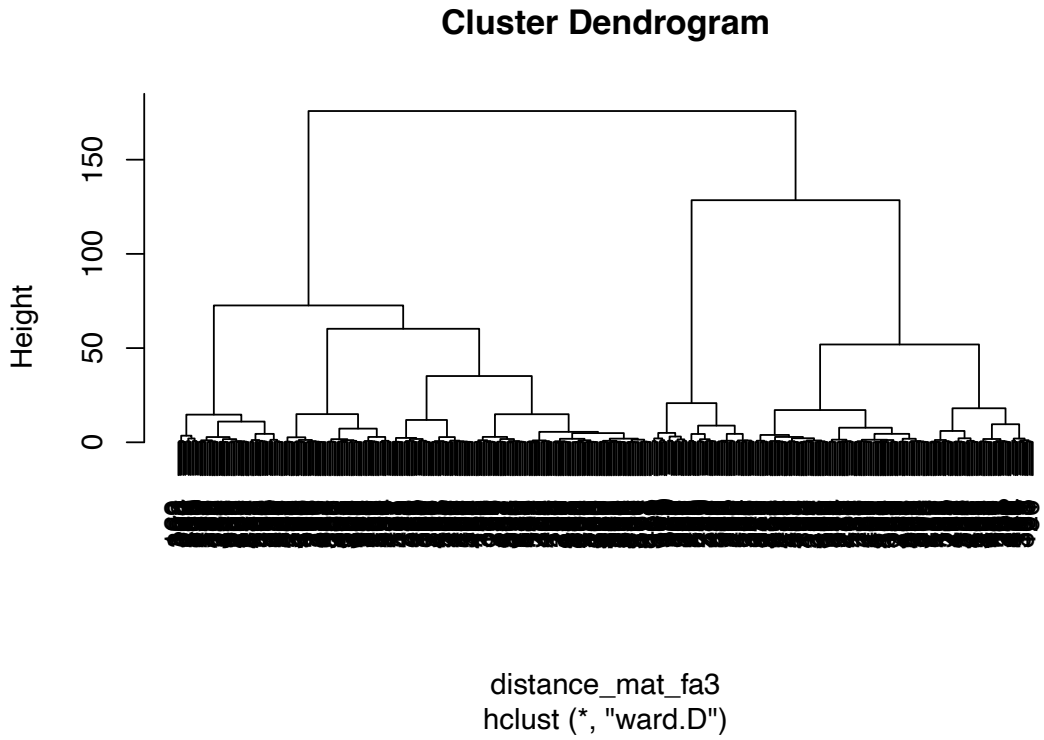
- **Hierarchical clustering Model**
- Fitting Hierarchical clustering Model to dataset

```
set.seed(240)  # Setting seed
Hierar_cl_fa3 <- hclust(distance_mat_fa3, method = "ward.D")
Hierar_cl_fa3
```

```
##
## Call:
## hclust(d = distance_mat_fa3, method = "ward.D")
##
## Cluster method   : ward.D
## Distance         : euclidean
## Number of objects: 500
```

- Plotting dendrogram

```
plot(Hierar_cl_fa3)
```



- Choosing no. of clusters: 3 clusters

```
fit_fa3 <- cutree(Hierar_cl_fa3, k = 3)
fit_fa3
```

```
##   [1] 1 2 2 1 2 2 2 2 2 1 3 1 1 1 2 2 3 2 1 2 1 3 2 2 1 3 2 1 2 2 2 1 2 1 1 2 2
##  [38] 1 2 3 1 2 2 2 2 1 1 1 2 3 1 2 3 1 2 3 2 3 3 2 2 2 2 2 2 2 2 1 1 2 2 2 3 2
##  [75] 2 2 2 2 3 2 2 1 2 2 2 1 1 1 2 1 2 1 2 1 1 1 2 3 3 1 2 2 1 3 2 2 1 1 3 1 2
## [112] 2 2 2 1 2 2 2 1 2 2 1 1 1 3 1 2 1 3 2 2 2 3 1 2 2 2 1 1 3 2 1 1 2 1 1 1 2
## [149] 2 2 1 1 1 1 2 1 1 2 2 2 1 2 3 3 1 2 2 1 2 2 2 1 1 3 2 2 2 2 2 2 1 2 1 2 3
## [186] 1 2 2 1 2 1 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 2 2 2 1 1 2 1 1 1 1 2 3 2 1
## [223] 2 2 2 1 3 2 2 2 2 3 2 1 2 2 2 1 2 2 1 2 2 2 2 1 2 2 1 2 1 2 1 1 2 2 2 2 3
## [260] 3 1 2 2 2 2 2 2 1 2 1 2 1 2 1 2 2 1 3 1 2 1 2 2 1 2 2 2 2 3 1 1 2 2 2 1 2
## [297] 2 3 1 2 2 2 2 2 2 3 2 2 3 2 2 3 3 2 1 2 2 1 3 1 2 1 2 2 1 3 2 2 2 2 2 2 3
## [334] 1 1 3 2 1 1 2 1 1 1 1 3 2 3 1 2 1 2 2 2 1 2 2 2 2 3 2 2 3 1 1 2 1 1 2 2 2
## [371] 2 2 2 2 1 3 2 2 1 2 2 2 1 3 2 2 2 1 1 2 2 1 2 2 1 2 2 3 3 2 1 2 1 2 1 1 1
## [408] 2 1 1 2 1 2 2 1 1 2 2 2 2 2 2 1 2 2 1 3 2 2 3 1 2 1 1 1 1 2 3 3 1 1 2 2 2
## [445] 2 2 2 2 1 2 2 3 2 2 1 2 3 3 3 2 2 2 1 1 1 3 1 1 2 2 1 2 1 2 1 1 1 2 2 2 1
## [482] 2 1 2 2 3 2 2 2 2 2 2 3 2 2 2 1 1 1 3 2
```

- Find number of observations in each cluster

```
table(fit_fa3)
```

```
## fit_fa3
##   1   2   3
## 163 277  60
```

```
final_data_fa3 <-cbind(sample_s, cluster = fit_fa3)
```

- Display first six rows of final data

```r
head(final_data_fa3)
```

```
##             PA1        PA2 cluster
## [1,] -0.1837921 -1.0875018       1
## [2,] -1.2801680 -0.3512067       2
## [3,] -1.8763213  0.5404125       2
## [4,]  0.2015468 -0.5554284       1
## [5,] -0.7518820  1.5114189       2
## [6,] -0.0306106  1.9569939       2
```

- Find mean values for each cluster

```r
hcentres_fa3<-aggregate(x=final_data_fa3, by=list(cluster=fit_fa3), FUN="mean")
print(hcentres_fa3)
```

```
##   cluster        PA1        PA2 cluster
## 1       1  0.5967203 -0.9183870       1
## 2       2 -0.6591268  0.2544071       2
## 3       3  1.4218784  1.3204386       3
```

- **Non-hierarchical clustering Model (Kmeans)**

```r
# 3 clusters
set.seed(55)
k_cl_fa3 <- kmeans(sample_s, 3, nstart=25)
k_cl_fa3
```

```
## K-means clustering with 3 clusters of sizes 93, 187, 220
##
## Cluster means:
##          PA1        PA2
## 1  1.1785205  1.1181371
## 2  0.4430026 -0.8371817
## 3 -0.8747450  0.2389374
##
## Clustering vector:
##   [1] 2 3 3 2 3 1 3 3 3 2 1 2 2 2 3 3 1 3 2 1 1 1 3 3 2 1 2 2 3 3 3 2 3 2 2 3 3
##  [38] 2 3 1 2 3 3 1 3 2 2 2 3 1 2 3 1 2 3 1 2 1 1 3 3 3 3 2 1 3 3 2 1 3 2 2 1 1
##  [75] 1 3 3 3 1 3 3 2 3 3 3 2 2 2 3 2 3 2 3 2 2 1 3 1 1 2 3 3 2 1 3 3 1 2 1 2 3
## [112] 3 1 3 2 1 3 3 2 3 3 2 2 2 1 2 3 2 1 2 3 3 1 2 3 3 3 2 2 1 3 2 2 2 2 2 2 3
## [149] 3 3 2 2 2 2 3 2 2 3 1 3 2 3 1 1 1 3 3 2 3 2 3 2 2 1 2 3 3 3 3 3 2 3 2 2 1
## [186] 2 3 3 1 3 2 3 2 3 3 3 3 3 2 3 3 2 2 1 3 3 3 2 1 3 1 2 2 3 2 2 2 2 3 1 3 2
## [223] 1 3 3 2 1 3 1 3 3 1 3 2 3 3 3 2 3 3 2 3 3 3 3 2 3 3 2 3 1 3 2 2 3 3 3 3 1
## [260] 1 2 3 3 3 3 3 3 2 2 2 3 2 3 2 3 3 2 1 2 2 2 3 3 2 1 3 2 3 1 2 2 3 3 2 2 3
## [297] 1 1 2 3 2 3 3 3 3 1 3 2 1 3 3 1 1 2 1 3 3 2 1 2 2 2 3 3 2 1 2 1 3 3 3 3 1
## [334] 2 2 1 3 2 2 3 2 2 2 2 1 3 1 2 3 2 3 3 3 2 3 3 3 3 1 1 1 1 2 2 3 2 2 2 3 2
## [371] 3 3 3 3 2 1 3 3 2 3 3 3 2 1 3 3 2 2 2 3 3 2 3 3 1 3 3 1 1 2 2 2 2 3 2 2 2
## [408] 3 2 2 3 2 3 3 2 2 3 1 2 3 1 3 2 2 3 2 1 1 3 1 2 3 2 2 2 2 3 1 1 2 2 3 3 3
## [445] 3 3 3 3 2 3 3 1 3 3 2 3 1 1 1 3 2 2 2 2 2 1 2 2 3 3 2 2 2 2 1 2 2 3 1 3 2
## [482] 3 2 3 2 1 3 3 3 3 3 1 2 2 3 2 2 2 1 3
##
## Within cluster sum of squares by cluster:
## [1] 107.2371 107.8363 188.8248
##  (between_SS / total_SS =  59.5 %)
##
## Available components:
```

```
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

- **Choose final cluster solution**

Similarity measure

- Using Silhouette Coefficient Index to compare hierarchical and Non-hierarchical solutions

```
# Hierarchical model
silhouette_score_hierar <- silhouette(fit_fa3, dist(sample_s))
avg_sil_width_hierar <- mean(silhouette_score_hierar[, 'sil_width'])

# Non-hierarchical model  (K-means)
silhouette_score_kmeans <- silhouette(k_cl_fa3$cluster, dist(sample_s))
avg_sil_width_kmeans <- mean(silhouette_score_kmeans[, 'sil_width'])

# Compare Silhouette Coefficient Index
print(paste("Hierarchical clustering average silhouette width:", avg_sil_width_hierar))
```

```
## [1] "Hierarchical clustering average silhouette width: 0.365886930628537"
```

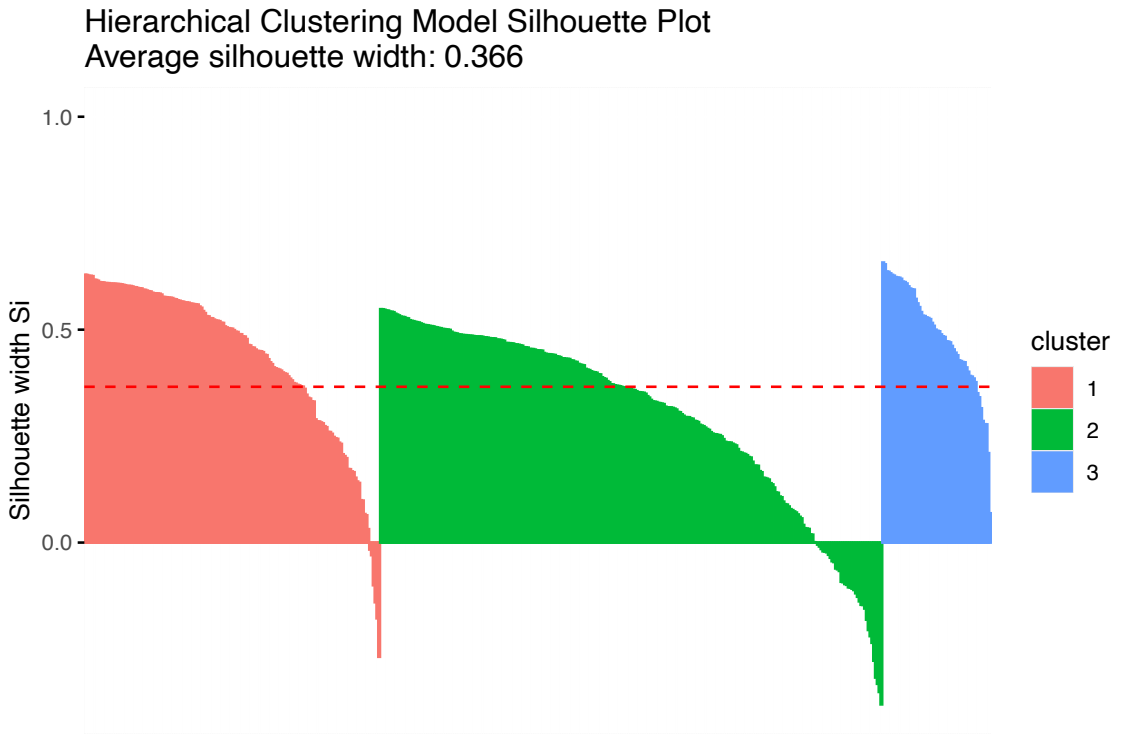
```
print(paste("K-means clustering average silhouette width:", avg_sil_width_kmeans))
```

```
## [1] "K-means clustering average silhouette width: 0.385485019803922"
```

Based on the result above, non-hierarchical model has a higher Silhouette Coefficient Index, which means clusters are closely connected within each other, while clusters are relatively separated from each other.

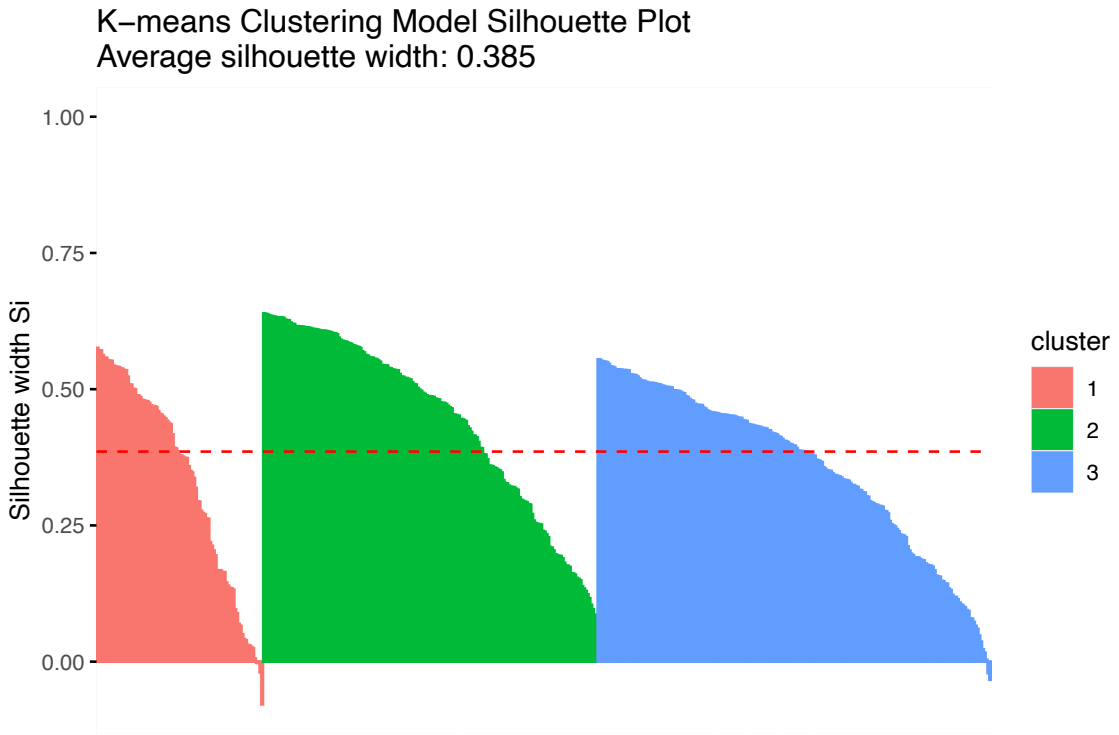
```
# Silhouette plot
fviz_silhouette(silhouette_score_hierar) +
    ggtitle(paste("Hierarchical Clustering Model Silhouette Plot\nAverage silhouette width:", round(avg_
```

```
##   cluster size ave.sil.width
## 1       1  163          0.44
## 2       2  277          0.30
## 3       3   60          0.49
```




```
fviz_silhouette(silhouette_score_kmeans) +
    ggtitle(paste("K-means Clustering Model Silhouette Plot\nAverage silhouette width:", round(avg_sil_
```

```
##   cluster size ave.sil.width
## 1       1   93          0.33
## 2       2  187          0.44
## 3       3  220          0.36
```



According to these two figures, the K-means clustering model provides better clustering quality, with higher average silhouette width and more consistent intra-group similarity. In hierarchical clustering model, it can be seen that many samples, in cluster 1 and 2, have a negative silhouette coefficient. This means that they are not in the right cluster.

Hence, we choose to use K-means clustering (3 clusters) solution in our case.

8. Validate & Profile cluster solution

8.1. Stability check

1) Using Silhouette Coefficient Index again to test internal new cluster (subset)

```
#Save the initial cluster results
initial_cluster <- k_cl_fa3$cluster

set.seed(55)  # Ensure reproducibility
indices <- sample(1:nrow(sample_s), size = 100)  # Random indices for the subset
subset <- sample_s[indices, ]  # Extract the subset
new_cluster <- kmeans(subset, 3, nstart = 25)  # Perform KMeans clustering on the subset
```

```
silhouette_score_new_cluster <- silhouette(new_cluster$cluster, dist(subset))
avg_sil_width_new_cluster <- mean(silhouette_score_new_cluster[, 'sil_width'])
```

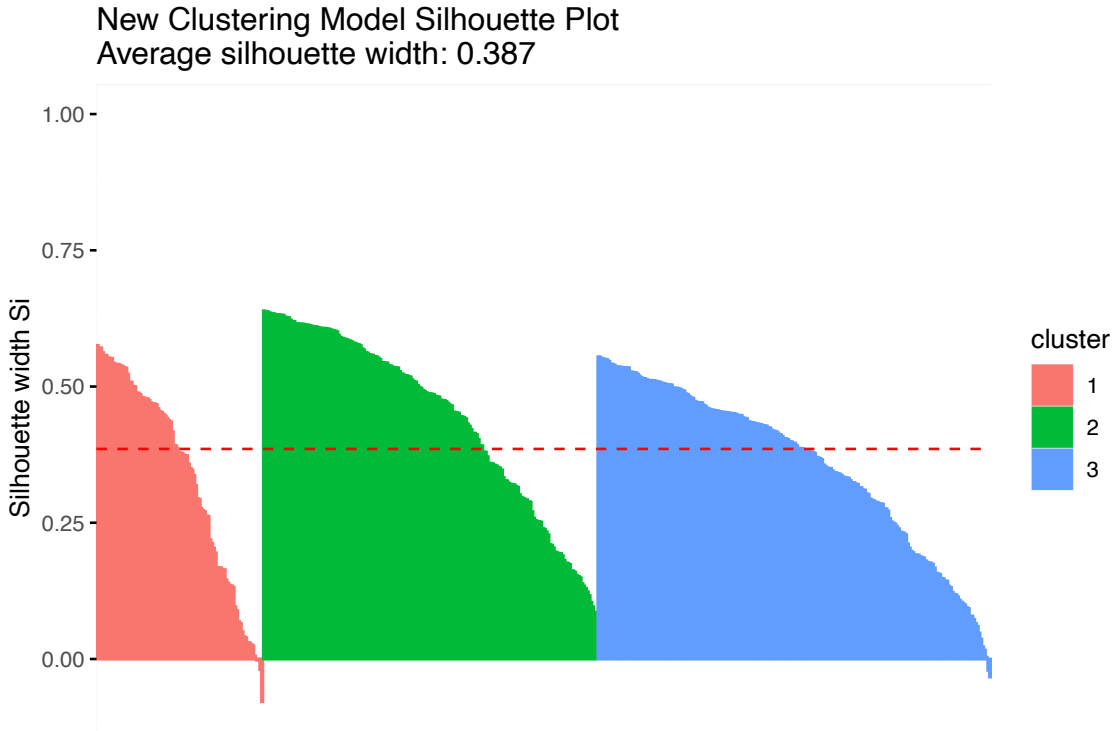
```
print(paste("New clustering average silhouette width:", avg_sil_width_new_cluster))
```

```
## [1] "New clustering average silhouette width: 0.387436909751364"
```

- Silhouette plot

```
fviz_silhouette(silhouette_score_kmeans) +
    ggtitle(paste("New Clustering Model Silhouette Plot\nAverage silhouette width:", round(avg_sil_width
```

```
##   cluster size ave.sil.width
## 1       1   93          0.33
## 2       2  187          0.44
## 3       3  220          0.36
```



2) Choosing different sample Sample size: 500 (given) No explicit pattern detected from the previous section. Therefore, random sampling will be used to sample 500.

```
set.seed(1)
sample_2 <- sample_n(loanDataFiltered_df, 500)
```

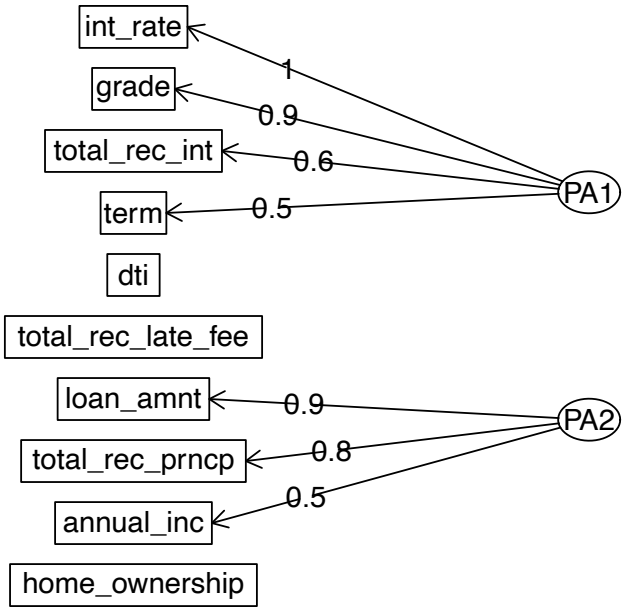
3) Performing Factor Analysis to new sample Two Factors Solution with Orthogonal rotation (Varimax Rotation)

```
fa3v_2<-(fa(sample_2,2, n.obs=500, rotate="varimax", fm="pa"))
print.psych(fa3v_2, cut=0.3,sort="TRUE")
```

```
## Factor Analysis using method =  pa
## Call: fa(r = sample_2, nfactors = 2, n.obs = 500, rotate = "varimax",
##     fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                     item   PA1   PA2     h2    u2 com
## int_rate               3  0.95       0.9136 0.086 1.0
## grade                 10  0.94       0.8929 0.107 1.0
## total_rec_int          4  0.63  0.55 0.7009 0.299 2.0
## term                   8  0.49       0.2846 0.715 1.4
## dti                    6             0.0406 0.959 1.2
## total_rec_late_fee     7             0.0026 0.997 1.5
## loan_amnt              2  0.31  0.93 0.9684 0.032 1.2
```

```
## total_rec_prncp       5        0.83 0.6993 0.301 1.0
## annual_inc            1        0.53 0.2778 0.722 1.0
## home_ownership        9             0.0643 0.936 1.3
##
##                        PA1  PA2
## SS loadings           2.57 2.27
## Proportion Var        0.26 0.23
## Cumulative Var        0.26 0.48
## Proportion Explained  0.53 0.47
## Cumulative Proportion 0.53 1.00
##
## Mean item complexity =  1.3
## Test of the hypothesis that 2 factors are sufficient.
##
## df null model =  45  with the objective function =  6.22 with Chi Square =  3076.63
## df of  the model are 26  and the objective function was  0.96
##
## The root mean square of the residuals (RMSR) is  0.05
## The df corrected root mean square of the residuals is  0.07
##
## The harmonic n.obs is  500 with the empirical chi square  133.56  with prob <  2e-16
## The total n.obs was  500  with Likelihood Chi Square =  474.92  with prob <  5.3e-84
##
## Tucker Lewis Index of factoring reliability =  0.743
## RMSEA index =  0.186  and the 90 % confidence intervals are  0.172 0.201
## BIC =  313.34
## Fit based upon off diagonal values = 0.97
## Measures of factor score adequacy
##                                                 PA1  PA2
## Correlation of (regression) scores with factors   0.97 0.98
## Multiple R square of scores with factors          0.95 0.96
## Minimum correlation of possible factor scores     0.90 0.93
```

```
fa.diagram(fa3v_2)
```



Once we have decided best solution, we can set scores to regression.

```
fa3v_2<-(fa(sample_2,2, n.obs=500, rotate="varimax", fm="pa", scores="regression"))
head(fa3v_2$scores, 10)
```

```
##               PA1        PA2
##  [1,] -0.7191392  1.9333363
##  [2,]  1.3926625 -1.8066480
##  [3,]  0.7798019 -1.3619908
##  [4,]  1.1832440 -1.3560982
##  [5,] -0.5811087 -0.8124614
##  [6,] -0.4392972  0.8691897
##  [7,]  0.6707179 -1.4386772
##  [8,] -0.3613264 -0.1282618
##  [9,] -0.8703624 -0.8852356
## [10,]  0.3600808  0.0441758
```

We can use the factor scores for further analysis, before doing that we need to add them into our dataframe:

```
sample_fa3_2 <- fa3v_2$scores
```

4) Create clusters

- Standardize data

```
# Standardise data set
sample_fa3_s_2 <- scale(sample_fa3_2)

# View the standardized data
head(sample_fa3_s_2)
```

```
##             PA1        PA2
```

```
## [1,] -0.7378579  1.9681100
## [2,]  1.4289125 -1.8391430
## [3,]  0.8000996 -1.3864881
## [4,]  1.2140430 -1.3804895
## [5,] -0.5962346 -0.8270747
## [6,] -0.4507318  0.8848233
```

- **Find the Linkage Method to Use**
- Define linkage methods

```
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")
```

- Function to compute agglomerative coefficient

```
ac <- function(x) {
  agnes(sample_fa3_s_2, method = x)$ac
}
```

- Calculate agglomerative coefficient for each clustering linkage method

```
sapply(m, ac)
```

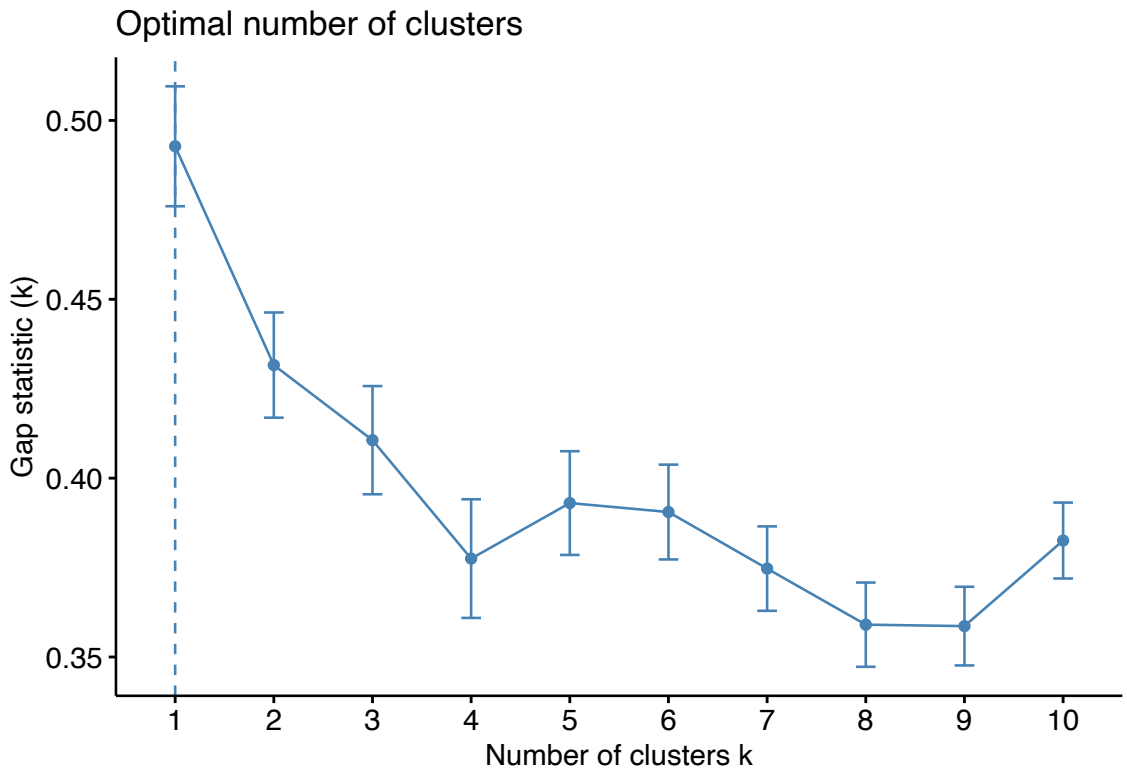
```
##   average    single  complete      ward
## 0.9715927 0.8857328 0.9827171 0.9959920
```

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

```
gap_stat_k_fa3_2 <- clusGap(sample_fa3_s_2, FUN = kmeans, nstart = 25, K.max = 10, B = 50)
```

- Produce plot of potential number of clusters using gap statistic method

```
fviz_gap_stat(gap_stat_k_fa3_2) # 5 clusters for Non-hierarchical (Kmeans)
```



- Finding distance matrix

```
#using Euclidean distance
distance_mat_fa3_2 <- dist(sample_fa3_s_2, method = 'euclidean')
```

```
# 5 clusters
#(between_SS / total_SS =  73.6 %)
set.seed(55)
k_cl_fa3_2 <- kmeans(sample_fa3_s_2, 5, nstart=25)
k_cl_fa3_2
```

```
## K-means clustering with 5 clusters of sizes 151, 79, 104, 114, 52
##
## Cluster means:
##          PA1        PA2
## 1 -0.8712874 -0.4062146
## 2 -0.7169792  1.4760163
## 3  0.3841964  0.2305968
## 4  0.4457889 -1.0753481
## 5  1.8736423  0.8334758
##
## Clustering vector:
##   [1] 2 4 4 4 1 2 4 1 1 3 3 3 4 3 4 1 1 3 1 1 5 4 1 2 5 1 1 4 5 4 3 2 2 1 3 5 5
##  [38] 4 2 3 1 3 1 1 4 3 1 4 3 1 1 4 1 5 3 1 1 4 1 4 1 4 2 2 2 3 1 3 1 1 1 3 3 1
##  [75] 2 1 3 2 1 3 3 4 3 5 1 1 1 5 2 4 3 1 3 1 1 3 3 4 1 1 1 4 3 2 1 5 1 1 3 4 2
## [112] 4 3 4 3 3 1 2 4 4 2 3 1 1 1 4 1 3 3 4 2 1 5 2 2 1 1 2 4 1 1 1 2 5 3 3 2 1
## [149] 4 1 2 4 3 1 3 2 1 2 3 3 4 3 4 4 4 4 4 4 4 4 4 2 1 5 4 5 4 4 3 5 1 3 1 1 4 3
## [186] 5 2 1 5 4 3 1 4 1 1 3 3 1 2 4 4 3 4 3 3 1 3 1 2 4 3 3 3 5 5 3 5 4 2 5 1 3
```

```
## [223] 1 4 5 5 1 2 3 2 1 1 4 1 1 2 4 5 3 3 5 1 4 1 1 1 4 3 5 1 1 4 2 4 3 2 3 3 3
## [260] 3 1 3 3 2 1 4 2 1 3 1 4 2 2 1 2 3 3 4 1 3 4 1 3 2 4 1 4 2 5 2 2 2 1 2 4 1
## [297] 2 3 1 1 4 5 1 1 2 2 3 5 2 5 3 1 1 4 5 1 3 1 2 4 1 1 5 2 4 1 4 4 1 1 3 4 5
## [334] 2 3 1 1 2 3 3 1 1 1 2 2 3 2 5 2 4 3 3 1 4 1 2 3 4 2 5 3 3 5 1 5 3 5 5 5 1
## [371] 1 3 1 2 1 4 5 4 5 1 1 1 2 4 4 1 4 1 1 4 2 3 2 1 1 1 4 1 3 1 4 5 2 3 1 4 2
## [408] 4 4 5 1 3 4 4 2 4 5 3 5 2 1 5 5 4 2 4 3 4 4 4 3 1 3 1 5 1 1 4 4 4 4 3 1 2
## [445] 4 4 4 1 4 1 1 4 4 2 1 3 4 5 5 2 1 4 4 4 1 1 2 1 2 4 4 1 2 3 3 1 4 3 5 1 3
## [482] 3 4 5 1 2 1 1 2 4 2 4 1 1 4 3 3 3 2 1
##
## Within cluster sum of squares by cluster:
## [1] 61.38244 64.81063 32.52642 39.65720 53.32022
##  (between_SS / total_SS =  74.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
k_cl_fa3
```

```
## K-means clustering with 3 clusters of sizes 93, 187, 220
##
## Cluster means:
##          PA1        PA2
## 1  1.1785205  1.1181371
## 2  0.4430026 -0.8371817
## 3 -0.8747450  0.2389374
##
## Clustering vector:
##   [1] 2 3 3 2 3 1 3 3 3 2 1 2 2 2 3 3 1 3 2 1 1 1 3 3 2 1 2 2 3 3 3 2 3 2 2 3 3
##  [38] 2 3 1 2 3 3 1 3 2 2 2 3 1 2 3 1 2 3 1 2 1 1 3 3 3 3 2 1 3 3 2 1 3 2 2 1 1
##  [75] 1 3 3 3 1 3 3 2 3 3 3 2 2 2 3 2 3 2 3 2 2 1 3 1 1 2 3 3 2 1 3 3 1 2 1 2 3
## [112] 3 1 3 2 1 3 3 2 3 3 2 2 2 1 2 3 2 1 2 3 3 1 2 3 3 3 2 2 1 3 2 2 2 2 2 2 3
## [149] 3 3 2 2 2 2 3 2 2 3 1 3 2 3 1 1 1 3 3 2 3 2 3 2 2 1 2 3 3 3 3 3 2 3 2 2 1
## [186] 2 3 3 1 3 2 3 2 3 3 3 3 3 2 3 3 2 2 1 3 3 3 2 1 3 1 2 2 3 2 2 2 2 3 1 3 2
## [223] 1 3 3 2 1 3 1 3 3 1 3 2 3 3 3 2 3 3 2 3 3 3 3 2 3 3 2 3 1 3 2 2 3 3 3 3 1
## [260] 1 2 3 3 3 3 3 3 2 2 2 3 2 3 2 3 3 2 1 2 2 2 3 3 2 1 3 2 3 1 2 2 3 3 2 2 3
## [297] 1 1 2 3 2 3 3 3 3 1 3 2 1 3 3 1 1 2 1 3 3 2 1 2 2 2 3 3 2 1 2 1 3 3 3 3 1
## [334] 2 2 1 3 2 2 3 2 2 2 2 1 3 1 2 3 2 3 3 3 2 3 3 3 3 1 1 1 1 2 2 3 2 2 2 3 2
## [371] 3 3 3 3 2 1 3 3 2 3 3 3 2 1 3 3 2 2 2 3 3 2 3 3 1 3 3 1 1 2 2 2 2 3 2 2 2
## [408] 3 2 2 3 2 3 3 2 2 3 1 2 3 1 3 2 2 3 2 1 1 3 1 2 3 2 2 2 2 3 1 1 2 2 3 3 3
## [445] 3 3 3 3 2 3 3 1 3 3 2 3 1 1 1 3 2 2 2 2 2 1 2 2 3 3 2 2 2 2 1 2 2 3 1 3 2
## [482] 3 2 3 2 1 3 3 3 3 3 1 2 2 3 2 2 2 1 3
##
## Within cluster sum of squares by cluster:
## [1] 107.2371 107.8363 188.8248
##  (between_SS / total_SS =  59.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

- Compare two clusters using Adjusted Rand Index (ARI)

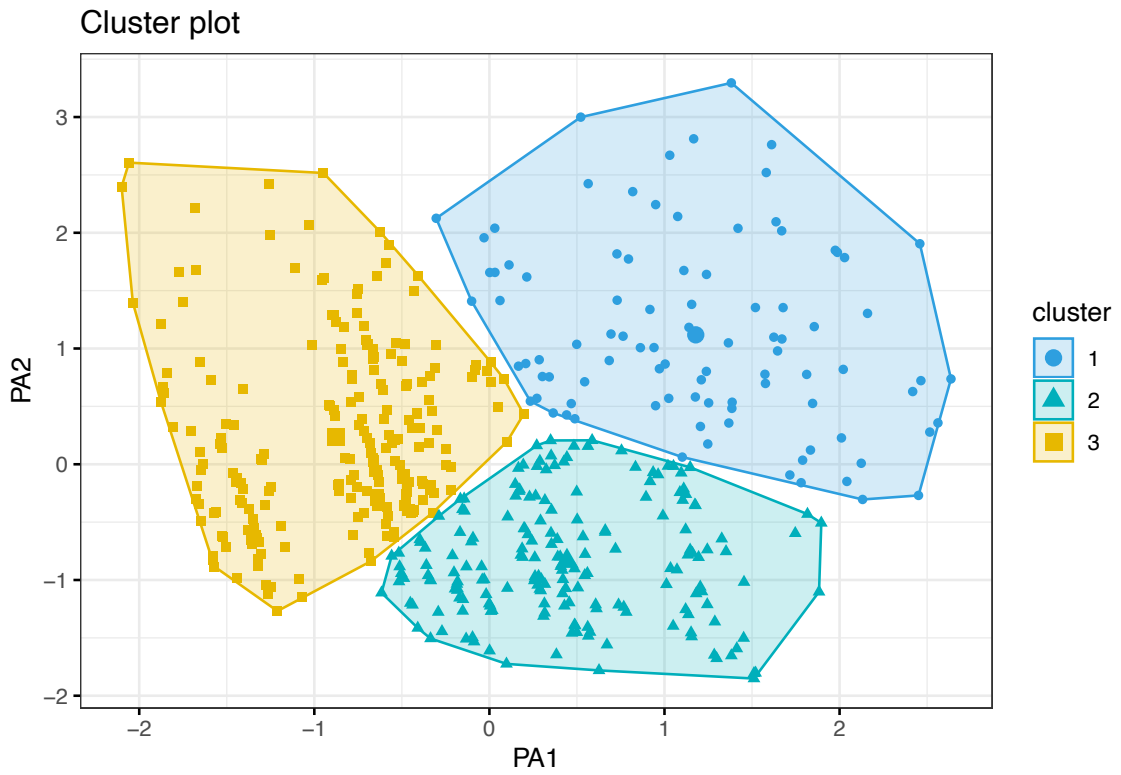
```r
# ARI
adjustedRandIndex(k_cl_fa3$cluster, k_cl_fa3_2$cluster)
```

```
## [1] -6.749255e-05
```

8.2. Cluster segmentation

Cluster plot

```r
# Plot the result of CA for the initial sample
fviz_cluster(k_cl_fa3, data = sample_s,
                         palette = c("#2E9FDF", "#00AFBB", "#E7B800", "red", "green", "purple", "grey",
                         geom = "point",
                         ellipse.type = "convex",
                         ggtheme= theme_bw())
```



Cluster characteristics

From the cluster plot above, we have 3 clusters for the initial sample with the following characteristics:

- Cluster 1: representing customers with higher creditworthiness, as it correlates with variables like interest rate, grade, and term, which are typically better for customers with good credit.
- Cluster 2: Customers in this cluster could have lower credit scores but relatively higher current financial activities or responsibilities.
- Cluster 3: representing customers with lower creditworthiness and less active or lower financial status.

Recommendation

Based on the clustering outcomes, here are some recommendations for the loan company tailored to each customer cluster:

- **Cluster 1 (Higher Creditworthiness):**

1. Premium Services: To retain low-risk customers, offer high-quality loan options with attractive interest rates and adjustable repayment terms.
2. Loyalty Programs: Encourage customer loyalty by implementing loyalty programs or offering incentives to retain their business and encourage referrals.
3. Credit Line Increases: Consider offering higher credit lines or larger loans based on the excellent credit standing of such customers.
4. Cross-Selling Opportunities: Given their probable financial stability, target these customers by cross-selling other financial products, such as investment opportunities or insurance.

- **Cluster 2 (Lower Credit Scores, Higher Financial Activity):**

1. Financial Planning Services: Provide financial planning services to help these customers manage their finances and improve their credit scores.
2. Credit Education: To aid these customers in improving their credit scores over time, offer credit counseling or educational resources.
3. Monitoring and Alerts: Introduce monitoring services that notify customers of possible credit issues or opportunities to improve their credit status.

- **Cluster 3 (Lower Creditworthiness and Financial Status):**

1. Secured Loan Options: Offer secured loans that require collateral, which can help mitigate risk while providing these customers access to credit.
2. Credit Building Products: Create products aimed at helping customers build or rebuild their credit, such as secured credit cards or small credit-builder loans.
3. Risk-Adjusted Pricing: Use risk-adjusted pricing models to ensure that the interest rates and fees compensate for the higher risk associated with this group.
4. Financial Assistance Programs: Implement hardship programs or financial assistance to support customers who may struggle with repayments.

By tailoring our approach to each customer segment, the lending company can more effectively manage risk, maximise profitability, and enhance customer satisfaction.