Assignment3

Weijie Gao

5 February 2017

Assignment 3 Part 1

```
dataPath <- "~/Documents/Chicago2016/Spring/Data Mining/week2"
# Germandata <- read.table(paste(dataPath,"german.data.csv",sep='/'),header=FALSE)

# Translate raw data into numerical for further analysis
# colnames(Germandata) <- c("Status","Duration","Credit_history","Purpose","Credit_Amount",
#                            "Savings_Account","Employment","Installment_rate","Status_Sex",
#                            "Other_guarantors","Present_residence","Property","Age","Other_installment","Housing",
#                            "Num_existingcredit","Job","Num_maintenance","Telephone","Foreign_worker","Class")
#
# Germandata$Status<- as.numeric(factor(Germandata$Status,levels=c("A11","A12",  "A13","A14")))
# Germandata$Credit_history <- as.numeric(factor(Germandata$Credit_history,levels=c("A30","A31","A32","A33","A34")))
# Germandata$Purpose <- as.numeric(factor(Germandata$Purpose,levels=c("A40","A41",  "A42","A43","A44","A45","A46","A47","A48","A49","A410")))
# Germandata$Savings_Account <- as.numeric(factor(Germandata$Savings_Account,levels=c("A61","A62","A63","A64","A65")))
# Germandata$Employment <- as.numeric(factor(Germandata$Employment,levels=c("A71","A72","A73","A74","A75")))
# Germandata$Status_Sex <- as.numeric(factor(Germandata$Status_Sex,levels=c("A91","A92","A93","A94","A95")))
# Germandata$Other_guarantors <- as.numeric(factor(Germandata$Other_guarantors,levels=c("A101","A102","A103")))
# Germandata$Property <- as.numeric(factor(Germandata$Property,levels=c("A121","A122","A123","A124")))
# Germandata$Other_installment <- as.numeric(factor(Germandata$Other_installment,levels=c("A141","A142","A143")))
# Germandata$Housing <- as.numeric(factor(Germandata$Housing,levels=c("A151","A152","A153")))
# Germandata$Job <- as.numeric(factor(Germandata$Job,levels=c("A171","A172","A173","A174")))
# Germandata$Telephone <- as.numeric(factor(Germandata$Telephone,levels=c("A191","A192")))
# Germandata$Foreign_worker <- as.numeric(factor(Germandata$Foreign_worker,levels=c("A201","A202")))
```

```r
# Store the translated data as comma separated values format
# write.table(Germandata, file = paste(dataPath,'Germancredit_numertic.
csv',sep = '/'), row.names = F)

# reload the translated data
Germandata <- read.table(paste(dataPath,"Germancredit_numertic.csv",sep
='/'),header=TRUE)

# Separate data set into train and test data
smp_size <- floor(0.632*nrow(Germandata))
set.seed(123)
train_ind <- sample(nrow(Germandata),size= smp_size)
train_data <- Germandata[train_ind,]
test_data <- Germandata[-train_ind,]

# Choose qualitative variable Credit_history, Savings_Account, Employme
nt and Status_Sex.
Credit_history <- train_data$Credit_history
Savings_Account <- train_data$Savings_Account
Employment <-train_data$Employment
Status_Sex <- train_data$Status_Sex

# Install packages poLCA
library(poLCA)

## Loading required package: scatterplot3d

## Warning: package 'scatterplot3d' was built under R version 3.3.2

## Loading required package: MASS

# define function
f1= cbind(Credit_history,Savings_Account,Employment,Status_Sex)~1

# Estimate the model with 2 to 7 clusters and runs every model 100 time
s
# and return the corresponding AIC and BIC value.
LCA_best_models<- function(data,formula,max.class=7){
  ret<-NULL
  min_aic<-100000
  min_bic<-100000
  clust_bic<-c()
  clust_aic<-c()
  for(i in 2:max.class){
    for(j in 1:100){
      res<-poLCA(formula,data,nclass=i,maxiter=1000,tol=.001,
                 verbose=FALSE)
      if(res$bic < min_bic){
        min_bic<-res$bic
        LCA_best_model_BIC<-res
      }
      if(res$aic < min_aic){
        min_aic<-res$aic
        LCA_best_model_AIC<-res
      }
    }
    clust_bic<-rbind(clust_bic,c(i,res$bic))
    clust_aic<-rbind(clust_aic,c(i,res$aic))
  }
  ret$LCA_best_model_BIC<-LCA_best_model_BIC
  ret$min_bic<-min_bic
  ret$LCA_best_model_AIC<-LCA_best_model_AIC
  ret$min_aic<-min_aic
  ret$clust_bic<-as.data.frame(clust_bic)
  ret$clust_aic<-as.data.frame(clust_aic)
  return(ret)
}

start.time <- Sys.time()
LCAresults <- LCA_best_models(train_data,f1,7)
end.time <- Sys.time()
(time.taken <- end.time - start.time)
```

```
## Time difference of 1.880263 mins
```

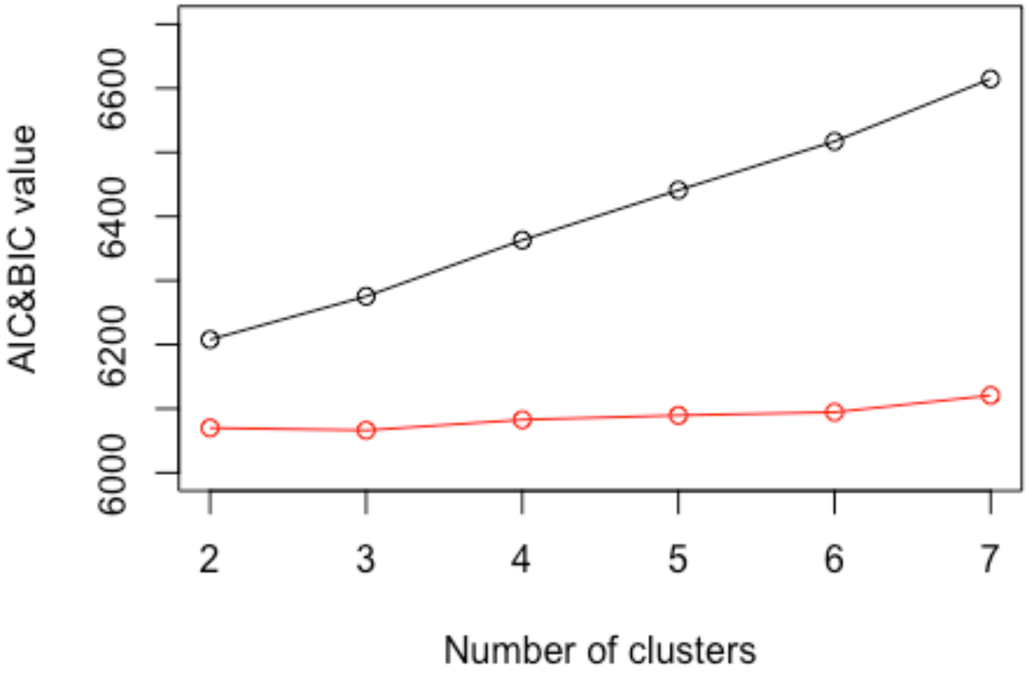
```
(aic <- LCAresults$clust_aic)
```

```
##   V1       V2
## 1  2 6069.966
## 2  3 6066.266
## 3  4 6082.756
## 4  5 6089.596
## 5  6 6094.517
## 6  7 6120.773
```

```
(bic <- LCAresults$clust_bic)
```

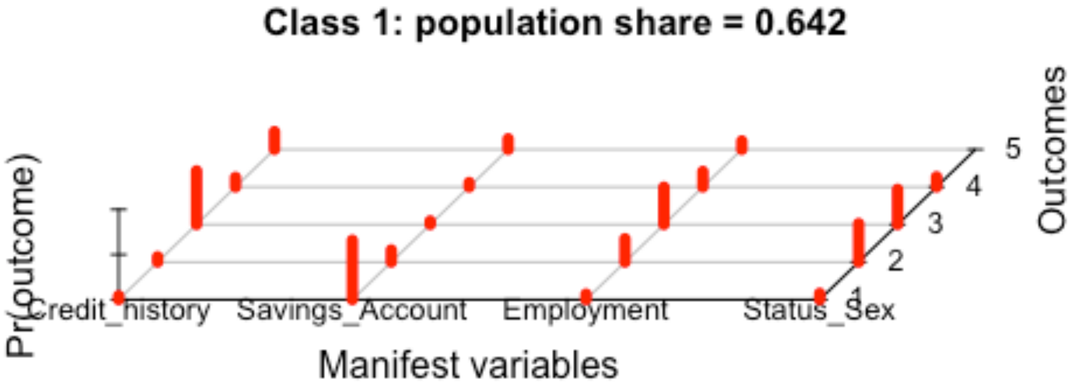
```
##   V1       V2
## 1  2 6207.881
## 2  3 6275.364
## 3  4 6363.036
## 4  5 6441.059
## 5  6 6517.161
## 6  7 6614.599
```

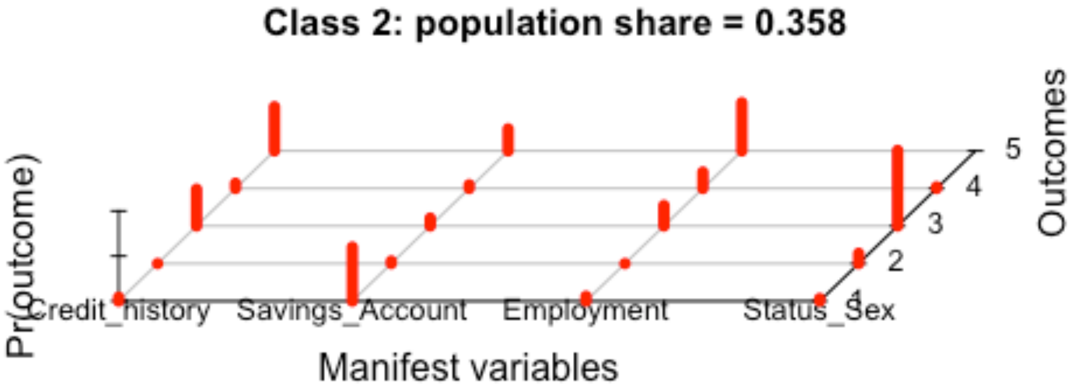
```
# generate the scree plot
plot(aic,type = "o",xlab="Number of clusters",ylab="AIC&BIC value",col=
"red",ylim=c(6000,6700))
points(bic,type="o")
```



The results above shows that AIC value reaches minimum when the number of cluster equals 3, while the BIC value is minimum when the number of cluster is 2. But from the generated scree plot, we could see that the difference between AIC and BIC is smallest when we choose two clusters. Hence, for the following analysis we will select two clusters.

```
# fit the data with best trained model
LCA_best_model <- poLCA(f1,train_data,nclass=2,nrep=100,tol=.001,verbos
e=FALSE,graphs=TRUE)
```

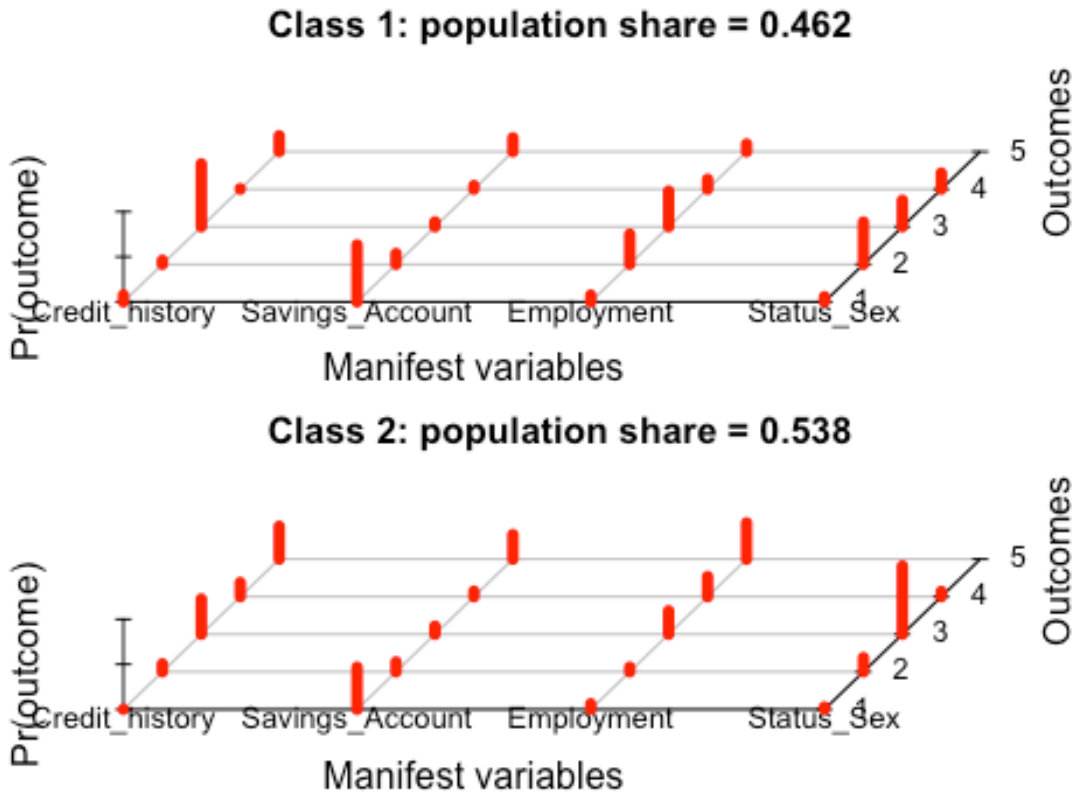




```
# attributes(LCA_best_model)
# LCA_best_model$npar
# table(LCA_best_model$predclass)
# LCA_best_model$posterior
# LCA_best_model$aic

# return the class-conditional probability from training set
(probs_train <- LCA_best_model$probs)

## $Credit_history
##                 Pr(1)        Pr(2)     Pr(3)      Pr(4)     Pr(5)
## class 1:  0.04239907 5.913140e-02 0.5952303 0.10591770 0.1973215
## class 2:  0.04330013 1.073432e-05 0.4087050 0.05754606 0.4904380
##
## $Savings_Account
##                Pr(1)      Pr(2)      Pr(3)      Pr(4)     Pr(5)
## class 1:  0.6589529 0.13450466 0.03997501 0.04886283 0.1177047
## class 2:  0.5994468 0.02835689 0.08744380 0.04054455 0.2442080
##
## $Employment
##                 Pr(1)        Pr(2)     Pr(3)     Pr(4)      Pr(5)
## class 1:  0.06114747 2.587027e-01 0.4146586 0.1720308 0.09346044
## class 2:  0.05387314 4.186548e-05 0.2286697 0.1820966 0.53531864
```

```
##
## $Status_Sex
##                  Pr(1)     Pr(2)     Pr(3)      Pr(4)
## class 1:  0.06993240 0.4270116 0.3899522 0.11310380
## class 2:  0.02926623 0.1136508 0.8389610 0.01812194
```

```
# perform holdout validation of trained LCA
Credit_history <- test_data$Credit_history
Savings_Account <- test_data$Savings_Account
Employment <-test_data$Employment
Status_Sex <- test_data$Status_Sex
# define function
f1= cbind(Credit_history,Savings_Account,Employment,Status_Sex)~1
LCA_test_model <- poLCA(f1, test_data, nclass=2, tol=0.001, na.rm=FALSE,
 probs.start=probs_train, verbose=TRUE,graphs=TRUE)
```



```
## Conditional item response (column) probabilities,
##  by outcome variable, for each class (row)
##
## $Credit_history
##              Pr(1)  Pr(2)  Pr(3)  Pr(4)  Pr(5)
## class 1:  0.0764 0.0446 0.6973 0.0059 0.1758
## class 2:  0.0000 0.0880 0.3909 0.1567 0.3644
```

```
##
## $Savings_Account
##            Pr(1)  Pr(2)  Pr(3)  Pr(4)  Pr(5)
## class 1:  0.6317 0.1198 0.0547 0.0408 0.1531
## class 2:  0.4676 0.1093 0.0895 0.0610 0.2727
##
## $Employment
##            Pr(1)  Pr(2)  Pr(3)  Pr(4)  Pr(5)
## class 1:  0.0749 0.3340 0.3964 0.1102 0.0845
## class 2:  0.0619 0.0514 0.2605 0.2237 0.4025
##
## $Status_Sex
##            Pr(1)  Pr(2)  Pr(3)  Pr(4)
## class 1:  0.0575 0.4692 0.2942 0.1792
## class 2:  0.0264 0.1575 0.7579 0.0582
##
## Estimated class population shares
##  0.4624 0.5376
##
## Predicted class memberships (by modal posterior prob.)
##  0.4674 0.5326
##
## =========================================================
## Fit for 2 latent classes:
## =========================================================
## number of observations: 368
## number of estimated parameters: 31
## residual degrees of freedom: 337
## maximum log-likelihood: -1828.19
##
## AIC(2): 3718.379
## BIC(2): 3839.53
## G^2(2): 279.1223 (Likelihood ratio/deviance statistic)
## X^2(2): 402.7962 (Chi-square goodness of fit)
##
```

```
LCA_best_model$P
```

```
## [1] 0.6421435 0.3578565
```

```
LCA_test_model$P
```

```
## [1] 0.4624308 0.5375692
```

```
LCA_test_model$aic
```

```
## [1] 3718.379
```

```
LCA_test_model$bic
```

```
## [1] 3839.53
```

```
LCA_test_model$probs

## $Credit_history
##                  Pr(1)      Pr(2)     Pr(3)       Pr(4)     Pr(5)
## class 1:  7.639216e-02 0.04463954 0.6973145 0.005899035 0.1757548
## class 2:  2.325608e-15 0.08797394 0.3909246 0.156684240 0.3644172
##
## $Savings_Account
##               Pr(1)     Pr(2)      Pr(3)      Pr(4)     Pr(5)
## class 1:  0.6317304 0.1197598 0.05465276 0.04075836 0.1530987
## class 2:  0.4675615 0.1092879 0.08947024 0.06098287 0.2726974
##
## $Employment
##                Pr(1)      Pr(2)     Pr(3)     Pr(4)      Pr(5)
## class 1:  0.07491328 0.33396697 0.3964048 0.1101877 0.08452724
## class 2:  0.06193170 0.05139546 0.2605428 0.2236763 0.40245381
##
## $Status_Sex
##                Pr(1)     Pr(2)     Pr(3)      Pr(4)
## class 1:  0.05745088 0.4691957 0.2942018 0.17915163
## class 2:  0.02640370 0.1574865 0.7579122 0.05819754
```

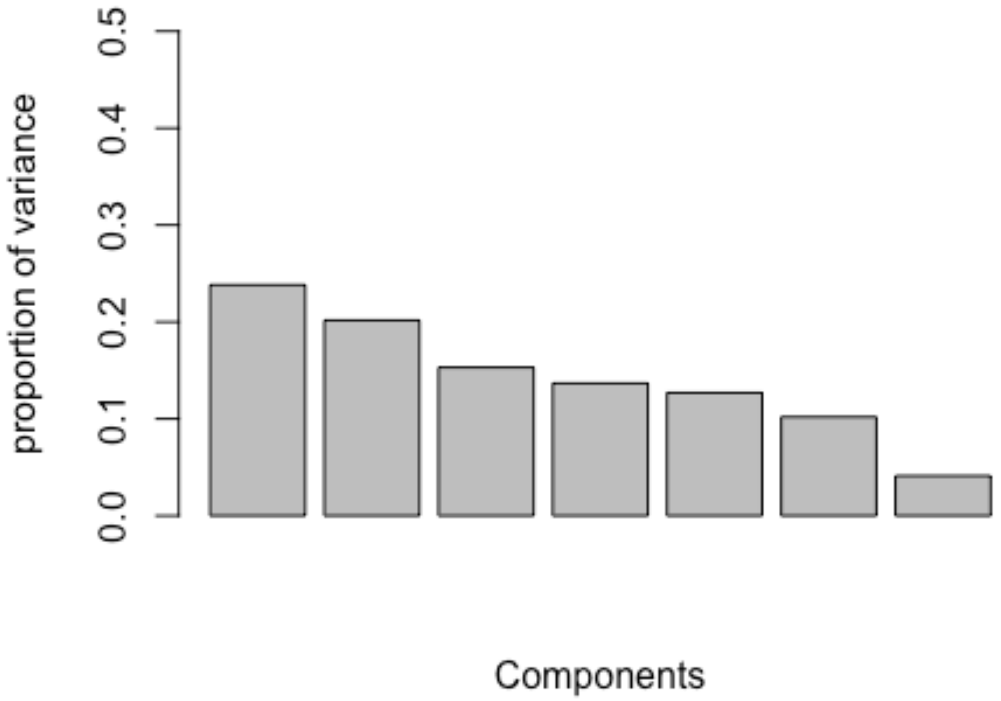
From the outputs we could see that the value of AIC reduced to 3718 and the value of BIC reduced to 3839, showing that the performance of fitted model is comparatively good. But when comparing the returned cluster sizes for training data and test data, it could see that the performance of fitted model is not very stable. And from the results of item response probabilities, it could be seen that for class 1, the majority ones are females who have existing credit paid back duly, with savings account less than 100 DM and have 1 to 4 years. And for class 2, most of them are single males who do not have existing credits history in our bank, with savings account less than 100 DM but have more than 7 years employment.

For last assignment, two numerical variables age and credit amount are selected to perform a k-overlapping means clustering and the results shows that young people with the lowest amount belong to cluster 1 and middle-age people with a comparatively highest amount belong to cluster 3 and the elder-age people with comparatively low amount belong to cluster 2. For latent class analysis, we choose four categorical variables Credit_history, Savings_Account,Employment and Status_Sex and group the data into two clusters. The objective of these two methods are the same, aiming to target the potential customers. And by combining these two results, we may considering pay more attention to elder single males who have a comparatively long employment but have no existing credits history in our bank.

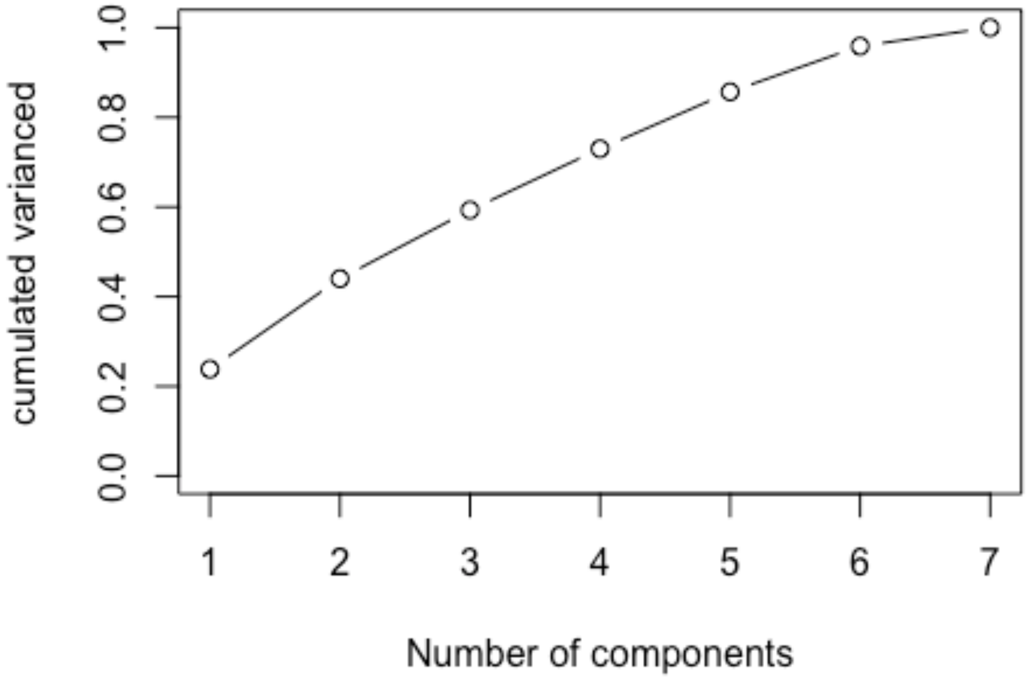
Assignment 3 Part 2

```
# install.packages("caret")
data(GermanCredit,package="caret")

# split sample into two random samples of sizes 70% and 30%
```

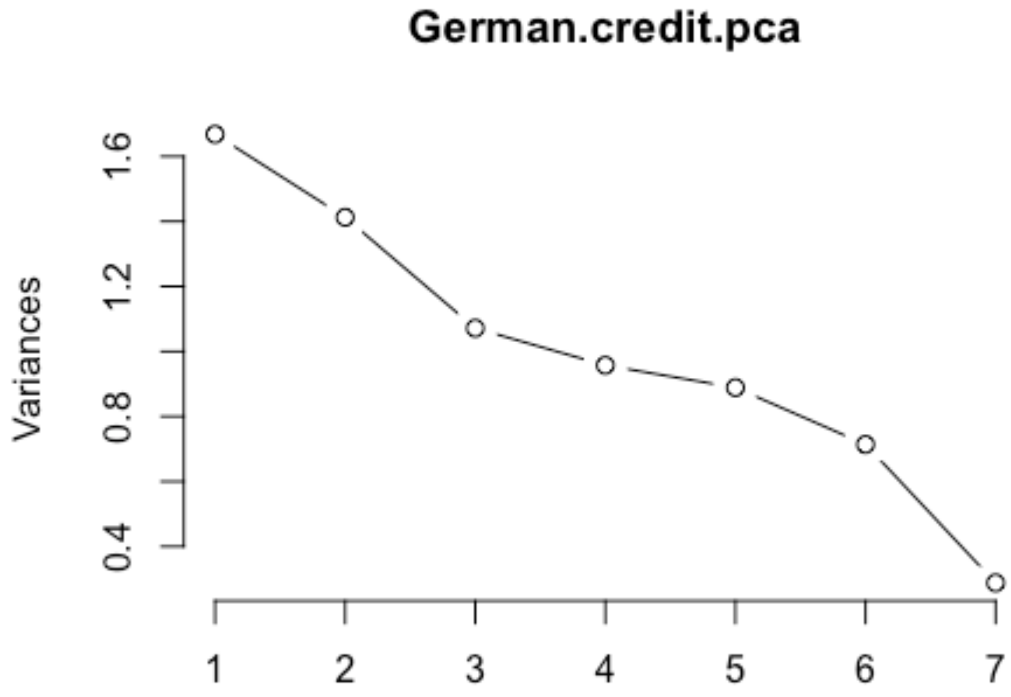
```
smp_size_pca <- floor(0.7*nrow(GermanCredit))

# seperate the data set into train and test data
set.seed(123)
train_ind <- sample(nrow(GermanCredit),size= smp_size_pca)
train_pca <- GermanCredit[train_ind,]
test_pca <- GermanCredit[-train_ind,]

# choose the first seven variables from the data
train_pca <- scale(train_pca[,1:7])
test_pca <- scale(test_pca[,1:7])

# perform principle component analysis on traing data
German.credit.pca <- prcomp(train_pca,center = TRUE,scale. = TRUE)

# return the corresponding importance of components
summary(German.credit.pca)

## Importance of components:
##                            PC1    PC2    PC3    PC4    PC5    PC6    
 PC7
## Standard deviation     1.2915 1.1885 1.0351 0.9784 0.9427 0.8449 0.5
3674
## Proportion of Variance 0.2383 0.2018 0.1531 0.1368 0.1270 0.1020 0.0
4116
## Cumulative Proportion  0.2383 0.4401 0.5931 0.7299 0.8569 0.9588 1.0
0000

# proportion of variance explained for different component
x.pvar <- (German.credit.pca$sdev^2)/sum(German.credit.pca$sdev^2)
barplot(x.pvar,ylim=c(0,0.5),xlab="Components",ylab="proportion of vari
ance")
```



```
# cumulated varianced explained
plot(cumsum(x.pvar),ylim=c(0,1), type='b',xlab="Number of components",y
lab="cumulated varianced")
```



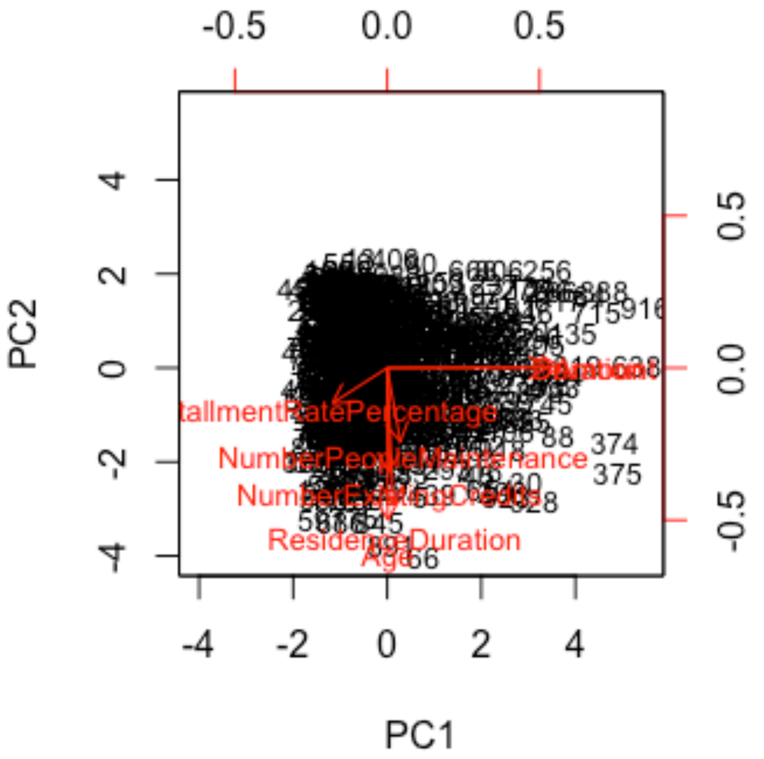
```
# Generate the scree plot
screeplot(German.credit.pca,type="l")
```



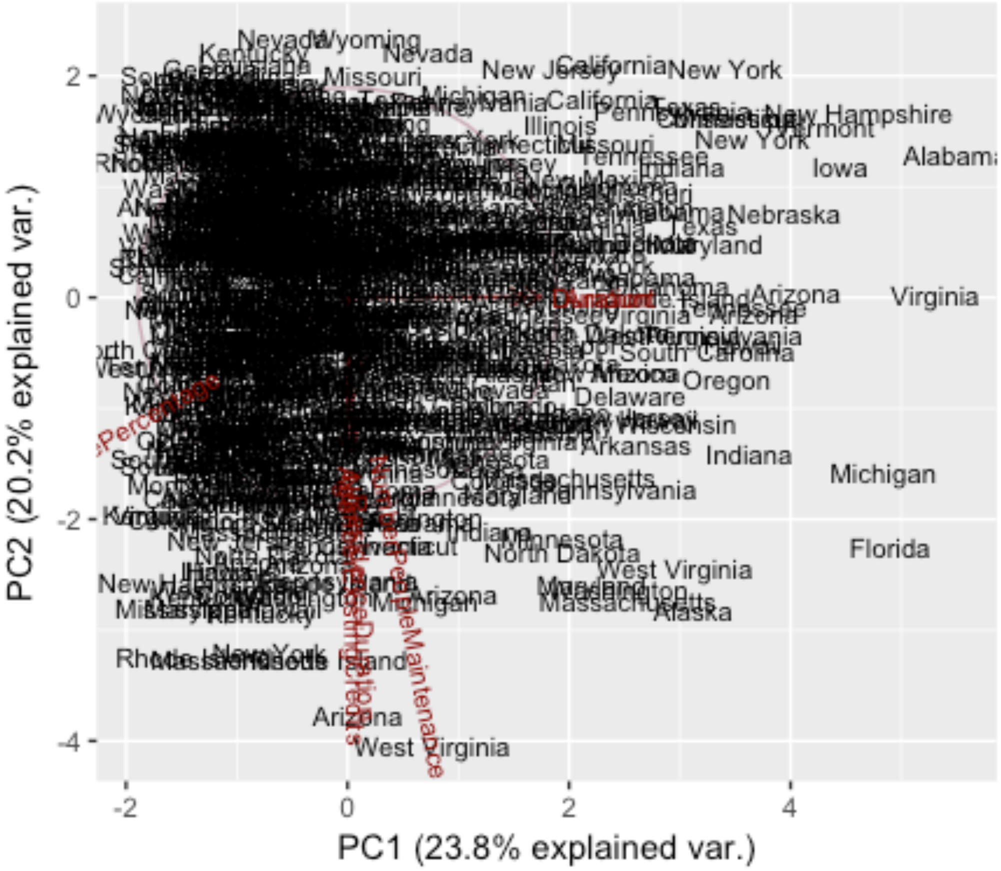
According to the cumulative proportion, more than 85 percent of the variance could be explained including 5 principle components, hence we choose to use the first five principle components.

```
# generate the biplot of the first two components
biplot(German.credit.pca,scale=0, cex=0.8)

# generate the biplot using ggbiplot
# install.packages("devtools")
library(devtools)
# install_github("ggbiplot","vqv")
library(ggbiplot)
```

```
## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.3.2

## Loading required package: plyr

## Loading required package: scales

## Warning: package 'scales' was built under R version 3.3.2

## Loading required package: grid
```



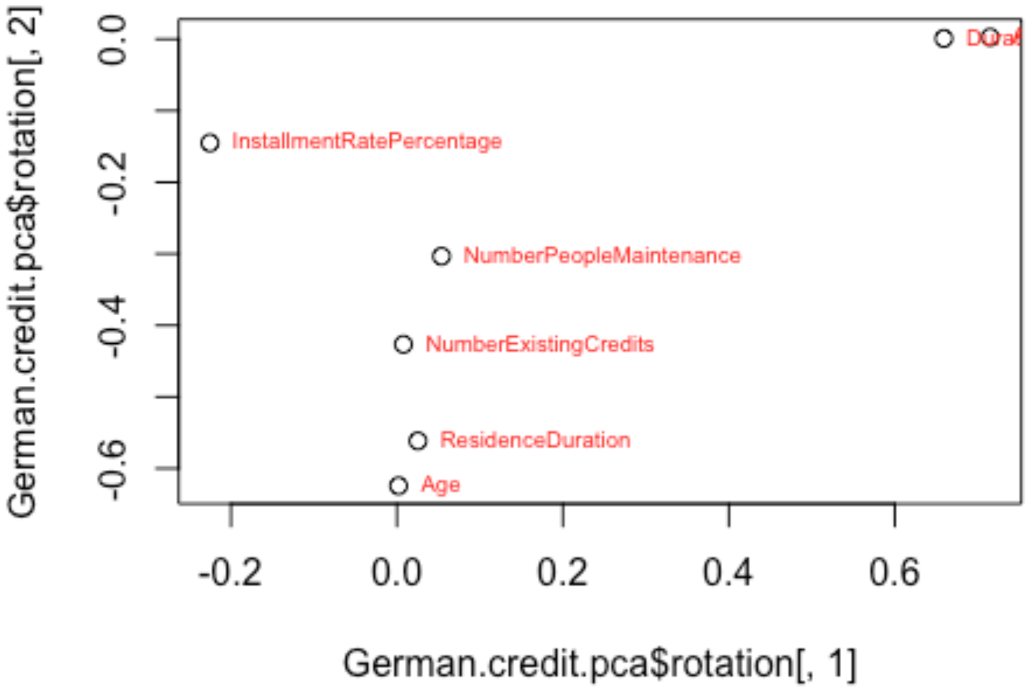
```
g <- ggbiplot(German.credit.pca, obs.scale = 1, var.scale = 1, labels=r
ow.names(USArrests),
              ellipse = TRUE, circle = TRUE)
g <- g + scale_color_discrete(name = '')
g <- g + theme(legend.direction = 'horizontal',
               legend.position = 'top')
print(g)
```



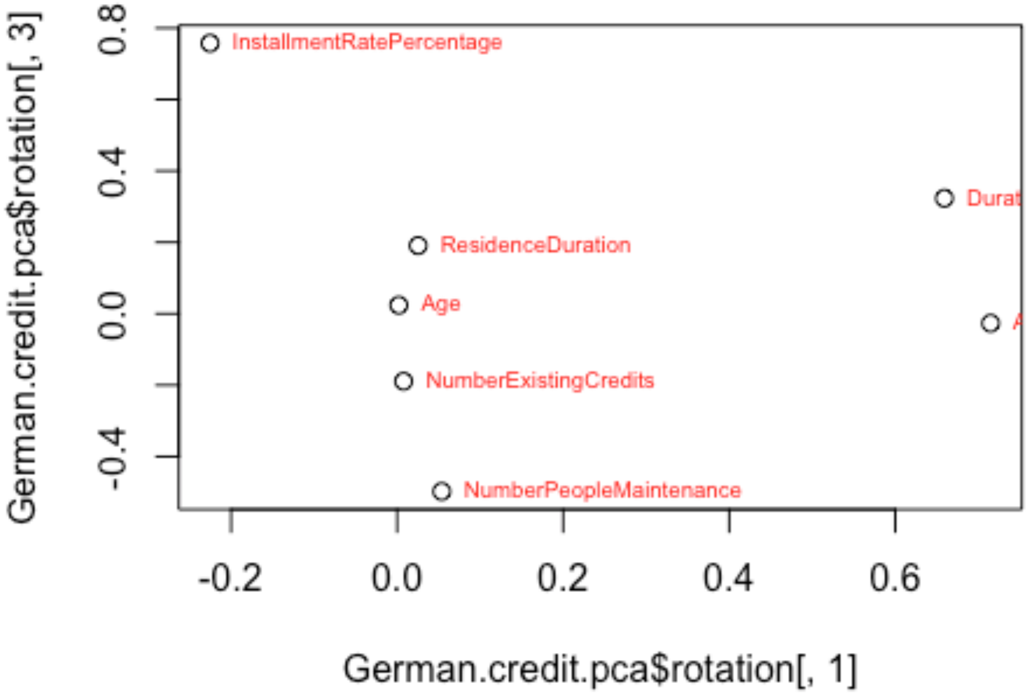
```
# check the correlation between training data and first three factors
cor((train_pca),German.credit.pca$x[,c(1:3)])

##                                  PC1          PC2         PC3
## Duration                 0.851356241  0.001088617  0.33473991
## Amount                   0.923304794  0.003156222 -0.02763820
## InstallmentRatePercentage -0.290980310 -0.172387151  0.78425617
## ResidenceDuration        0.032718075 -0.667327996  0.19758364
## Age                      0.002651075 -0.741910322  0.02473823
## NumberExistingCredits    0.010142276 -0.506973834 -0.19581819
## NumberPeopleMaintenance  0.069483461 -0.360658591 -0.51530855
```

Both of these two types of biplot is not clear to see the results, hence we choose to check the correlation between training data and the first three factors, and the table above shows that first principle component places approximatedly equal weight on Duration and Amount,with much less weight on the rest five variables. Hence this component roughly corresponds to a measure of overall duration and amount. The second component places most of it weight on ResidenceDuration and age, hence this component roughly corresponds to the ResidenceDuration and age. And the third component places most of it weight on InstallmentRatePercentage, hence this component roughly corresponds to the level of installment rate. Overall, we see that the Amount, Residence Duration, Age, NumberExistingCredits and NumberPeopleMaintenance varaibales are located close to each other, and that the Duration and Installmentrate percentage is far from other five. This indicates hat the these variables are correlated with each other-people with high credit amount tend to had longer residence duration,elder age and large number of existing credits. Duration and Installmentrate percentage variable is less correlated with the other five.

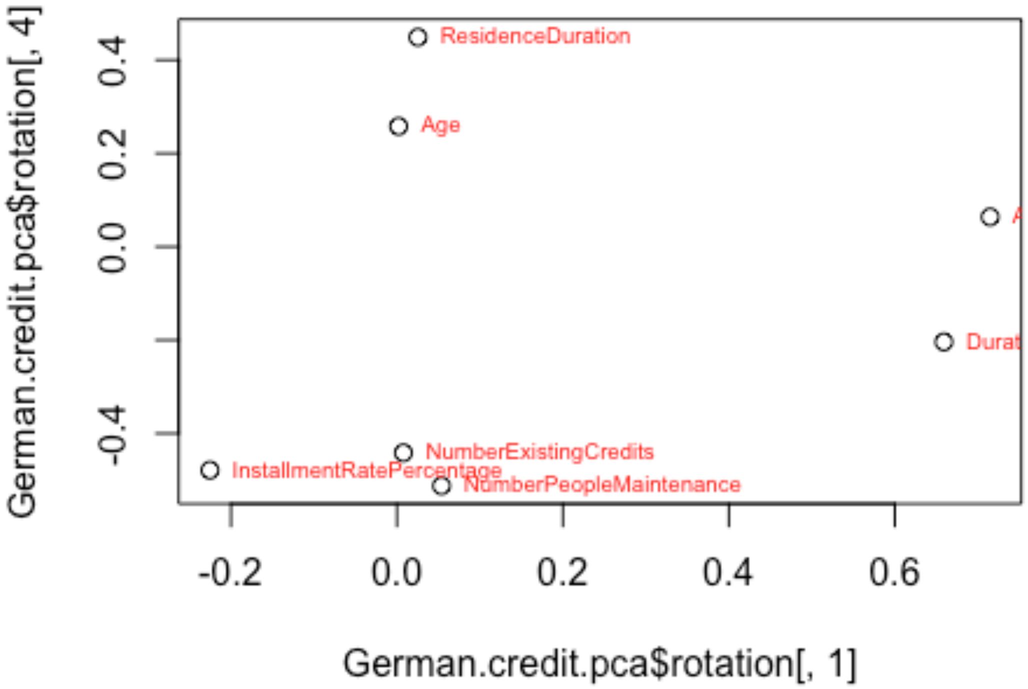
```
# Return the component loadings
loadings <- German.credit.pca$rotation

# Plot Component 1 loadings versus Component 2
plot(German.credit.pca$rotation[,1],German.credit.pca$rotation[,2])
text(German.credit.pca$rotation[,1],German.credit.pca$rotation[,2], row.
names(German.credit.pca$rotation), cex=0.6, pos=4, col="red")
```



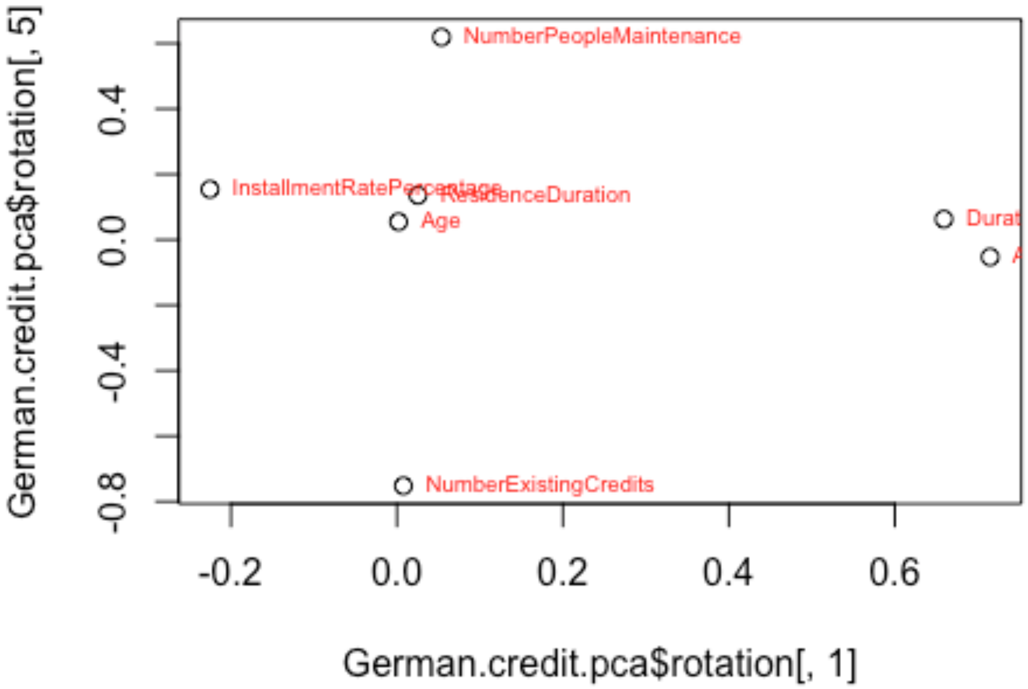
```
# Plot Component 1 loadings versus Component 3
plot(German.credit.pca$rotation[,1],German.credit.pca$rotation[,3])
text(German.credit.pca$rotation[,1],German.credit.pca$rotation[,3], row.
names(German.credit.pca$rotation), cex=0.6, pos=4, col="red")
```



```r
# Plot Component 1 loadings versus Component 4
plot(German.credit.pca$rotation[,1],German.credit.pca$rotation[,4])
text(German.credit.pca$rotation[,1],German.credit.pca$rotation[,4], row.
names(German.credit.pca$rotation), cex=0.6, pos=4, col="red")
```



```
# Plot Component 1 loadings versus Component 5
plot(German.credit.pca$rotation[,1],German.credit.pca$rotation[,5])
text(German.credit.pca$rotation[,1],German.credit.pca$rotation[,5], row.
names(German.credit.pca$rotation), cex=0.6, pos=4, col="red")
```



The above four plots shows the relative weight for different variables in component 1 to 5. And we could notice that variable Amount and Duration has a comparatively high weight in component 1. And other weight of variables could also be seen from these plots.

```
# Show that component score are orthogonal
round(t(German.credit.pca$x)%*%(German.credit.pca$x),2)

##         PC1   PC2    PC3    PC4    PC5    PC6    PC7
## PC1 1165.92   0.0   0.00   0.00   0.00   0.00   0.00
## PC2    0.00 987.4   0.00   0.00   0.00   0.00   0.00
## PC3    0.00   0.0 748.92   0.00   0.00   0.00   0.00
## PC4    0.00   0.0   0.00 669.19   0.00   0.00   0.00
## PC5    0.00   0.0   0.00   0.00 621.22   0.00   0.00
## PC6    0.00   0.0   0.00   0.00   0.00 498.99   0.00
## PC7    0.00   0.0   0.00   0.00   0.00   0.00 201.38

# Show that component loadings are orthogonal
round(t(loadings)%*%(loadings),2)

##     PC1 PC2 PC3 PC4 PC5 PC6 PC7
## PC1   1   0   0   0   0   0   0
## PC2   0   1   0   0   0   0   0
## PC3   0   0   1   0   0   0   0
## PC4   0   0   0   1   0   0   0
## PC5   0   0   0   0   1   0   0
## PC6   0   0   0   0   0   1   0
## PC7   0   0   0   0   0   0   1
```

The inner product of component score are diagonal matrix, showing that they are orthogonal. Similarly, the component loadings are orthogonal as well.

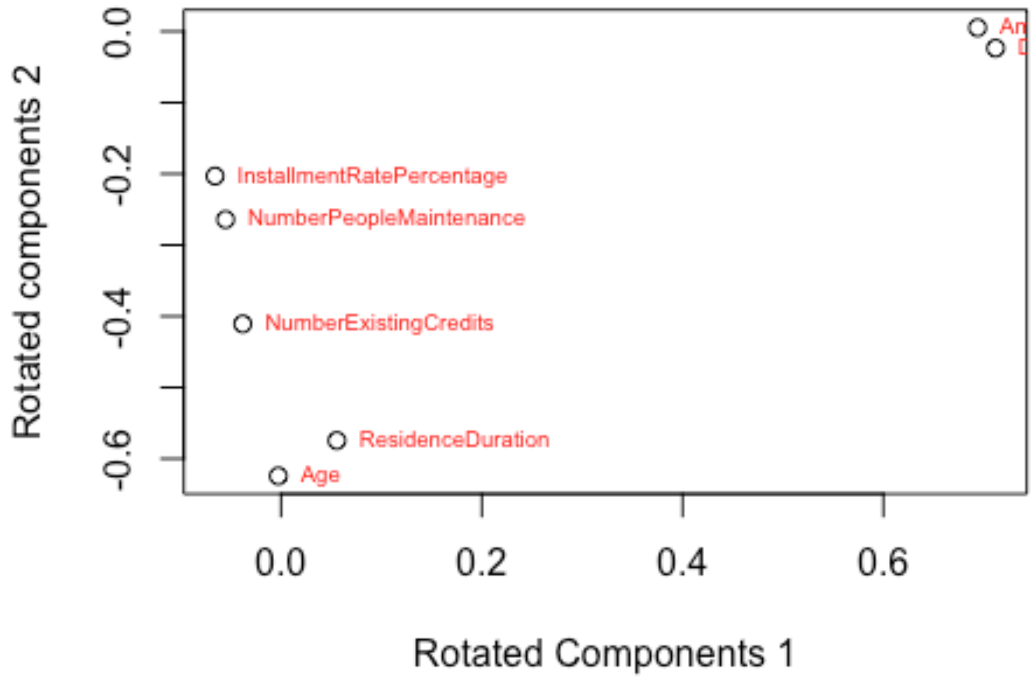
```
# Perfrom holdout validation of principal components solution
# predict the component score
predicted.factor.score <- predict(German.credit.pca, newdata=test_pca)

# matrix multiply the predicted component score from above with transpo
se of component loadings
predicted.data <- predicted.factor.score%*%t(German.credit.pca$rotation)

# compute the R square in the holdout sample
# method 1: calculate r square based on defition
(residuals.ss <- sum((test_pca-predicted.data)^2))

## [1] 1.177598e-27

y_bar <- mean(test_pca)
total_ss <- sum((test_pca-y_bar)^2)
r_square <-1-residuals.ss/total_ss
r_square

## [1] 1

# method 2: calculate r square based on correlation
cor(as.vector(test_pca),as.vector(predicted.data))^2

## [1] 1
```

Both of these two methods shows that the R squares is quite close to 1, implying that the model fitted is good and stable.

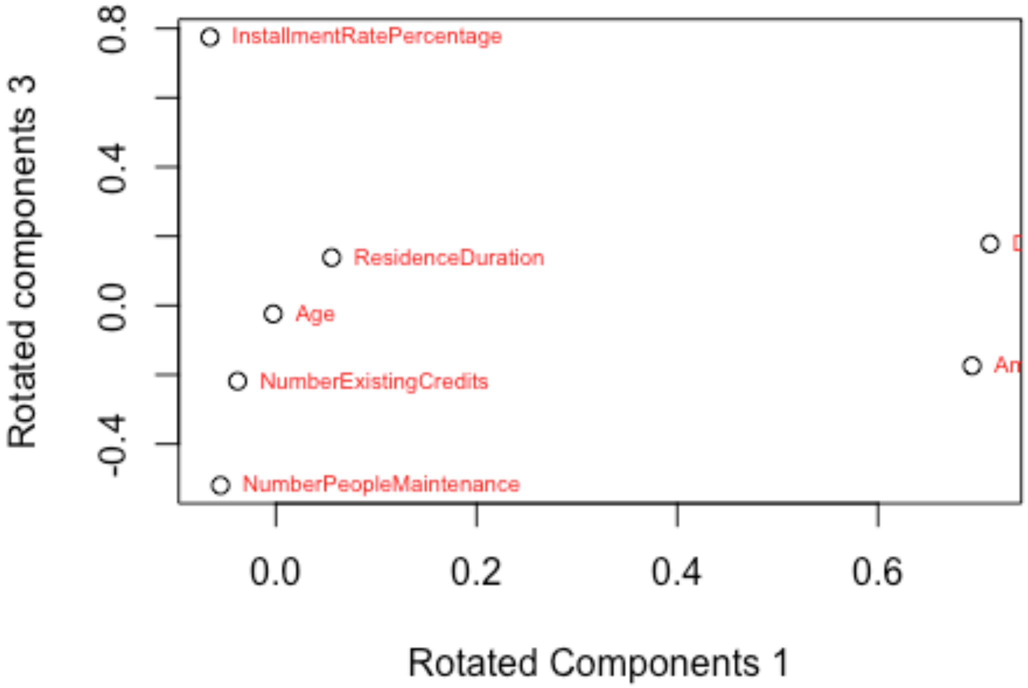
```
# return the vaf value of first 5 components
total_variance <-(var(predicted.factor.score[,1])
  +var(predicted.factor.score[,2])
  +var(predicted.factor.score[,3])
  +var(predicted.factor.score[,4])
  +var(predicted.factor.score[,5])
  +var(predicted.factor.score[,6])
  +var(predicted.factor.score[,7]))

components <- (var(predicted.factor.score[,6])
+var(predicted.factor.score[,7]))

(vaf <- 1-components/total_variance)

## [1] 0.855048
```

The value of test data vaf also shows that five factor components should be selected.

```
# Original component loadings
German.credit.pca$rotation[,1:3]

##                                    PC1           PC2         PC3
## Duration                   0.659198618  0.0009159422  0.32339208
## Amount                     0.714907832  0.0026555874 -0.02670126
## InstallmentRatePercentage -0.225303826 -0.1450433702  0.75766953
## ResidenceDuration          0.025333355 -0.5614774708  0.19088547
## Age                        0.002052707 -0.6242296645  0.02389959
## NumberExistingCredits      0.007853086 -0.4265584361 -0.18917987
## NumberPeopleMaintenance    0.053800512 -0.3034514887 -0.49783935

# Rotate the component loadings using varimax rotation
rotated.components <- varimax(German.credit.pca$rotation[,1:3])
rotated.components$loadings

## 
## Loadings:
##                            PC1    PC2    PC3   
## Duration                   0.712         0.179
## Amount                     0.694        -0.174
## InstallmentRatePercentage        -0.203  0.775
## ResidenceDuration                -0.574  0.139
## Age                              -0.624       
## NumberExistingCredits            -0.411 -0.218
## NumberPeopleMaintenance          -0.264 -0.520
## 
##                  PC1   PC2   PC3
## SS loadings    1.000 1.000 1.000
## Proportion Var 0.143 0.143 0.143
## Cumulative Var 0.143 0.286 0.429

# Plot rotated loadings 1 versus rotated loadings 2 and 3.
plot(rotated.components$loadings[,1],rotated.components$loadings[,2],xl
ab="Rotated Components 1",ylab="Rotated components 2")
text(rotated.components$loadings[,1],rotated.components$loadings[,2], r
ow.names(rotated.components$loadings), cex=0.6, pos=4, col="red")
```



```
plot(rotated.components$loadings[,1],rotated.components$loadings[,3],xl
ab="Rotated Components 1",ylab="Rotated components 3")
text(rotated.components$loadings[,1],rotated.components$loadings[,3], r
ow.names(rotated.components$loadings), cex=0.6, pos=4, col="red")
```



The rotated component loadings show that the first principle component places high weight on Duration and Amount,but a higher weight on Duration and lower weight on Amount comparing to original loadings. The second component also places most of its weight on ResidenceDuration and age, but increase the weight of InstallmentRatePercentage varible and reduce the weight of NumberPeopleMaintenance variable. For the third component it also has the highest weight on InstallmentRatePercentage. By comparing to the original loadings, the weight of InstallmentRatePercentage increased to 0.775 but the weight of Duration decreased to 0.179. Generally speaking, the principle components does not reduce the data significantly but it does help to outline most of the useful data and make our analysis more efficient.