hw6.R

yangz

2022-10-06

```
# import data
df <- read.delim("D:/GEORGIA INSTITUTE OF TECHNOLOGY/ISYE_6501/week5/hw5-SP22-1/data 8.2/uscrime.txt")
# header of data
head(df)
```

```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob    Time Crime
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602 26.2011   791
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599 25.2999  1635
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401 24.3006   578
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801 29.9012  1969
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399 21.2998  1234
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201 20.9995   682
```

```
# dimension of dataframe
dim(df)
```

```
## [1] 47 16
```

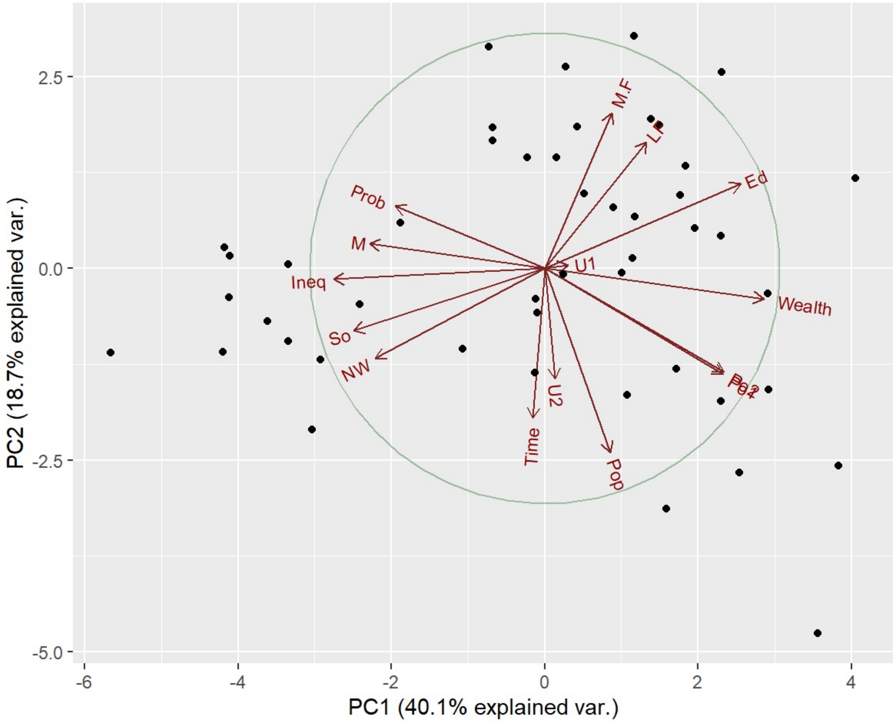
```
# set random seed
set.seed(9876)
# setting up pca and plot it in ggbiplot
pca <- prcomp(df[, 1:15], scale. = TRUE)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10    PC11
## Standard deviation     2.4534 1.6739 1.4160 1.07806 0.97893 0.74377 0.56729 0.55444 0.48493 0.44708 0.41915
## Proportion of Variance 0.4013 0.1868 0.1337 0.07748 0.06389 0.03688 0.02145 0.02049 0.01568 0.01333 0.01171
## Cumulative Proportion  0.4013 0.5880 0.7217 0.79920 0.86308 0.89996 0.92142 0.94191 0.95759 0.97091 0.98263
##                           PC12    PC13   PC14    PC15
## Standard deviation     0.35804 0.26333 0.2418 0.06793
## Proportion of Variance 0.00855 0.00462 0.0039 0.00031
## Cumulative Proportion  0.99117 0.99579 0.9997 1.00000
```

```
str(pca)
```

```
## List of 5
##  $ sdev    : num [1:15] 2.453 1.674 1.416 1.078 0.979 ...
##  $ rotation: num [1:15, 1:15] -0.304 -0.331 0.34 0.309 0.311 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:15] "M" "So" "Ed" "Po1" ...
##   .. ..$ : chr [1:15] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:15] 13.86 0.34 10.56 8.5 8.02 ...
##   ..- attr(*, "names")= chr [1:15] "M" "So" "Ed" "Po1" ...
##  $ scale   : Named num [1:15] 1.257 0.479 1.119 2.972 2.796 ...
##   ..- attr(*, "names")= chr [1:15] "M" "So" "Ed" "Po1" ...
##  $ x       : num [1:47, 1:15] -4.2 1.17 -4.17 3.83 1.84 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:15] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

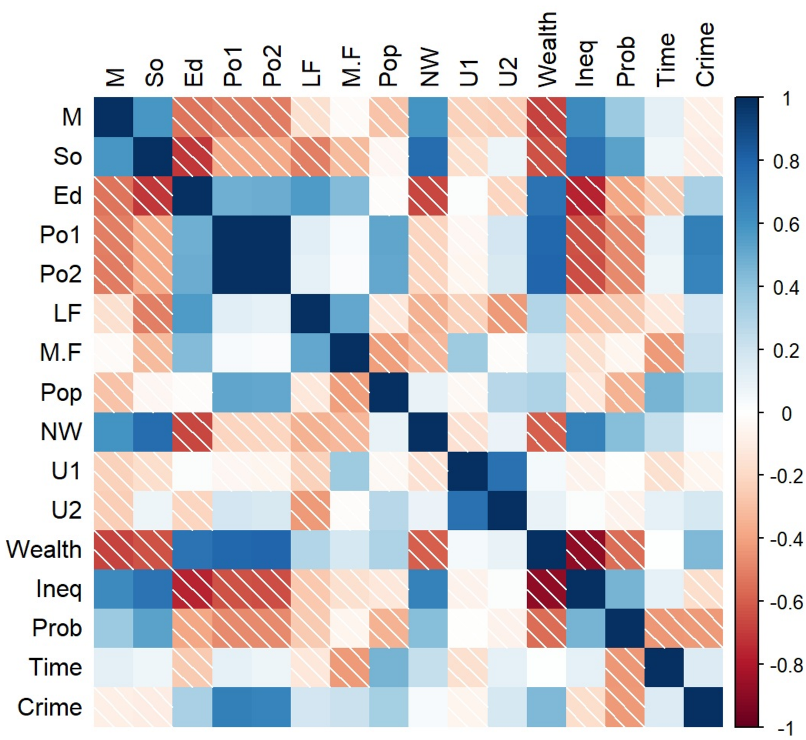
```
ggbiplot(pca, obs.scale = 1, var.scale = 1, circle = TRUE)
```



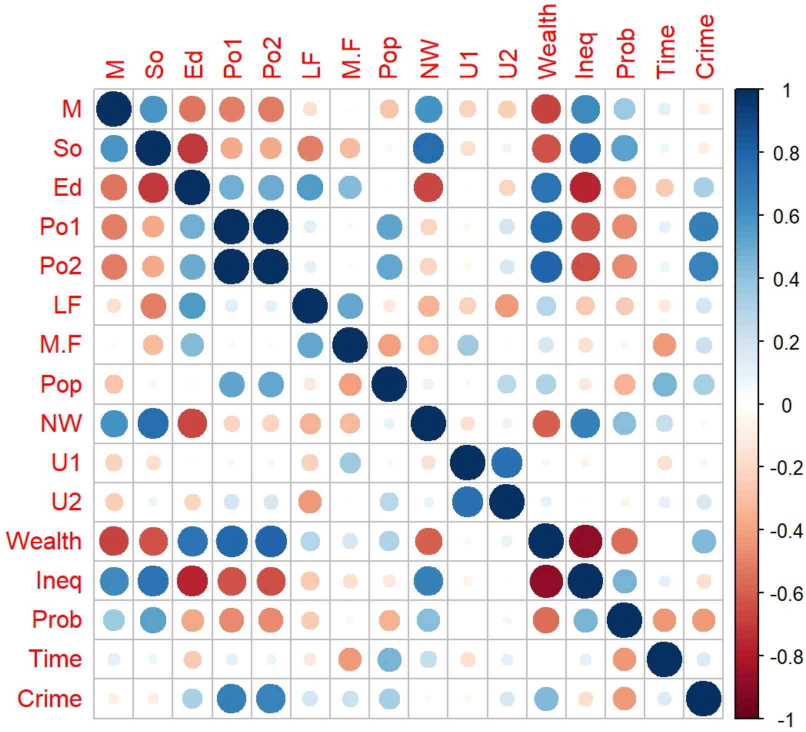

```
# calculate total variance explained by each principal component
explained_var <- pca$sdev^2 / sum(pca$sdev^2)
explained_var
```

```
##  [1] 0.401263510 0.186789802 0.133662956 0.077480520 0.063886598 0.036879593 0.021454579 0.020493418 0.0156770
19
## [10] 0.013325395 0.011712360 0.008546007 0.004622779 0.003897851 0.000307611
```

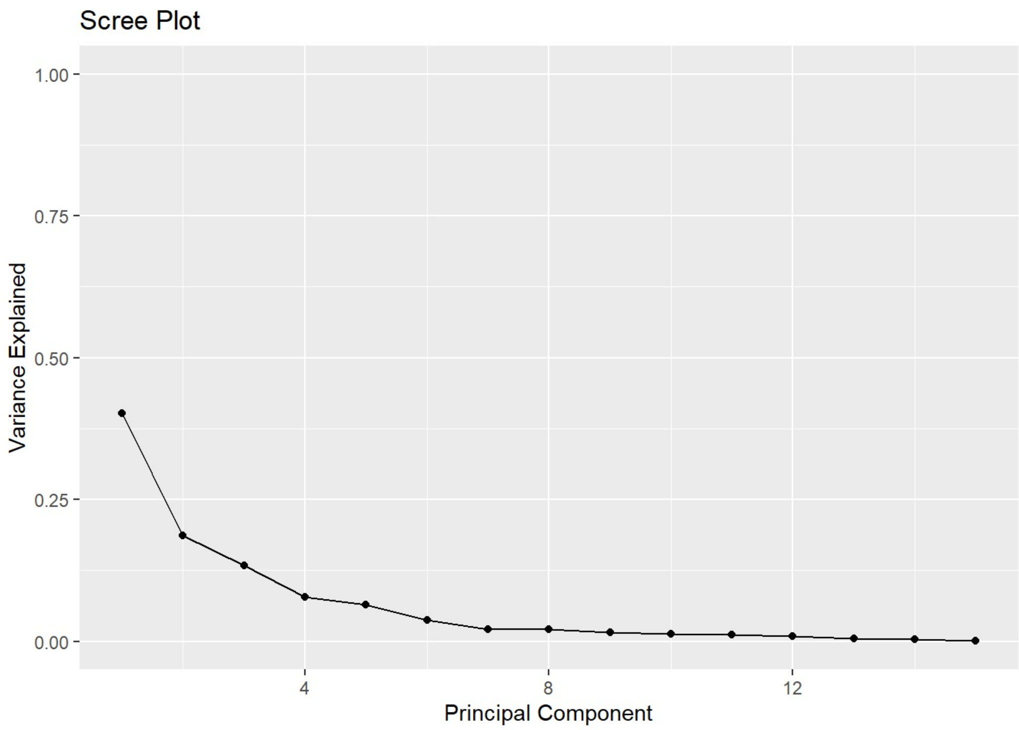
```
# plot correlation plot
corrplot(cor(df), method = "shade", type = "full", diag = TRUE, tl.col = "black", bg = "white", title = "", col =
NULL)
```




```
# Ignore correlations with p-value > 0.01(insignificant); change to dot for clear view
corrplot(cor(df), type="full", sig.level = 0.01, insig = "blank")
```




```
# plot relations between explained variance and principla component
qplot(c(1:15), explained_var) + geom_line() + xlab("Principal Component") + ylab("Variance Explained") + ggtitle(
"Scree Plot") + ylim(0, 1)
```




```
# choose first 5 parameters
pca_data <- cbind(pca$x[,1:5], df['Crime'])
glm_model <- glm(Crime~., data = pca_data, family = 'gaussian')
summary(glm_model)
```

```
##
## Call:
## glm(formula = Crime ~ ., family = "gaussian", data = pca_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -420.79  -185.01    12.21   146.24   447.86
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      35.59  25.428  < 2e-16 ***
## PC1            65.22      14.67   4.447 6.51e-05 ***
## PC2           -70.08      21.49  -3.261  0.00224 **
## PC3            25.19      25.41   0.992  0.32725
## PC4            69.45      33.37   2.081  0.04374 *
## PC5          -229.04      36.75  -6.232 2.02e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 59546.2)
##
##     Null deviance: 6880928  on 46  degrees of freedom
## Residual deviance: 2441394  on 41  degrees of freedom
## AIC: 657.7
##
## Number of Fisher Scoring iterations: 2
```

```
glm_cv <- cv.glm(pca_data, glm_model, K=5)
str(glm_cv)
```

```
## List of 4
##  $ call : language cv.glm(data = pca_data, glmfit = glm_model, K = 5)
##  $ K    : num 5
##  $ delta: num [1:2] 65605 63964
##  $ seed : int [1:626] 10403 624 -150993744 -806309647 1839918590 1844738663 -245088388 -1543757235 1089626282
-1314256349 ...
```

```
# calculate errors of deviation
sst = sum((df$Crime - mean(df$Crime))^2)
ssr = sum(glm_model$residuals^2)
# calculate r square
1- ssr/sst
```

```
## [1] 0.6451941
```

```
# calculate r square for cross validated model
1 - glm_cv$delta[1]*nrow(df)/sst
```

```
## [1] 0.5518838
```

```
# extract intercept and betas from the model
intercept <- glm_model$coefficients[1]
intercept
```

```
## (Intercept)
##    905.0851
```

```
betas <- glm_model$coefficients[2:6]
betas
```

```
##        PC1        PC2        PC3        PC4        PC5
##   65.21593  -70.08312   25.19408   69.44603 -229.04282
```

```
# reverse to get unscaled betas
betas_s <- pca$rotation[, 1:5] %*% betas
betas_s
```

```
##              [,1]
## M       60.794349
## So      37.848243
## Ed      19.947757
## Po1    117.344887
## Po2    111.450787
## LF      76.254902
## M.F    108.126558
## Pop     58.880237
## NW      98.071790
## U1       2.866783
## U2      32.345508
## Wealth  35.933362
## Ineq    22.103697
## Prob   -34.640264
## Time    27.205022
```

```
beta = glm_model$coefficients
betas_scaled = pca$rotation[,1:5] %*% beta[2:6]

betas_unscaled = betas_scaled/sapply(df[,1:15], sd)
intercept_unscaled = beta[1] - sum(betas_scaled*sapply(df[,1:15], mean)/
                            sapply(df[,1:15], sd))
intercept_unscaled
```

```
## (Intercept)
##   -5933.837
```

```
betas_unscaled
```

```
##                 [,1]
## M       4.837374e+01
## So      7.901922e+01
## Ed      1.783120e+01
## Po1     3.948484e+01
## Po2     3.985892e+01
## LF      1.886946e+03
## M.F     3.669366e+01
## Pop     1.546583e+00
## NW      9.537384e+00
## U1      1.590115e+02
## U2      3.829933e+01
## Wealth  3.724014e-02
## Ineq    5.540321e+00
## Prob   -1.523521e+03
## Time    3.838779e+00
```

```
# print betas
M <- 4.837374e+01
So <- 7.901922e+01
Ed <- 1.783120e+01
Po1 <- 3.948484e+01
Po2 <- 3.985892e+01
LF <- 1.886946e+03
M.F <- 3.669366e+01
Pop <- 1.546583e+00
NW <- 9.537384e+00
U1 <- 1.590115e+02
U2 <- 3.829933e+01
Wealth <- 3.724014e-02
Ineq <- 5.540321e+00
Prob <- -1.523521e+03
Time <- 3.838779e+00

# make prediction & calculate r square
prediction <- as.matrix(df[,1:15]) %*% betas_unscaled + intercept_unscaled
sse_ = sum((prediction - df[,16])^2)
sst_ = sum((df - mean(df[,16]))^2)
1 - sse_/sst_
```

```
## [1] 0.9983257
```

```
prediction
```

```
##             [,1]
##  [1,]  713.6803
##  [2,] 1195.7066
##  [3,]  506.4008
##  [4,] 1744.8151
##  [5,] 1004.3223
##  [6,]  901.3083
##  [7,]  817.7618
##  [8,] 1158.0158
##  [9,]  862.6600
## [10,]  906.1942
## [11,] 1309.8473
## [12,]  831.7397
## [13,]  668.7175
## [14,]  653.8079
## [15,]  663.3242
## [16,]  933.7860
## [17,]  467.7924
## [18,] 1097.8331
## [19,]  975.2212
## [20,] 1238.8452
## [21,]  805.7895
## [22,]  769.6724
## [23,]  768.1369
## [24,]  928.9523
## [25,]  604.2355
## [26,] 1845.7567
## [27,]  480.4270
## [28,] 1015.0839
## [29,] 1463.7936
## [30,]  801.6455
## [31,]  687.8542
## [32,]  969.6941
## [33,]  722.6822
## [34,]  841.7013
## [35,]  914.9564
## [36,]  977.8353
## [37,] 1211.6890
## [38,]  604.2928
## [39,]  627.6148
## [40,] 1069.8938
## [41,]  841.4929
## [42,]  272.2545
## [43,] 1043.4520
## [44,] 1126.3430
## [45,]  425.4541
## [46,]  927.1627
## [47,] 1139.3538
```