PCA of Students' Dropout and Academic Success Dataset from Kaggle

Jeremy Haakenson

2024-06-10

The data can be found at: https://www.kaggle.com/datasets/mattop/predict-students-dropout-and-academic-success

Load packages.

Read in data.

```
aca1 = read.csv('academic.csv')
```

Check for NAs.

```
sum(is.na(aca1))

## [1] 0
```

There are no missing values.

Look at the structure of the data.

```
str(aca1)

## 'data.frame':    4424 obs. of  37 variables:
##  $ Marital.status                                : int  1 1 1 1 2 2
1 1 1 1 ...
##  $ Application.mode                              : int  17 15 1 17
39 39 1 18 1 1 ...
##  $ Application.order                             : int  5 1 5 2 1 1
1 4 3 1 ...
##  $ Course                                        : int  171 9254
9070 9773 8014 9991 9500 9254 9238 9238 ...
##  $ Daytime.evening.attendance                    : int  1 1 1 1 0 0
1 1 1 1 ...
##  $ Previous.qualification                        : int  1 1 1 1 1
19 1 1 1 1 ...
##  $ Previous.qualification..grade.                : num  122 160 122
122 100 ...
##  $ Nacionality                                   : int  1 1 1 1 1 1
1 1 62 1 ...
##  $ Mother.s.qualification                        : int  19 1 37 38
37 37 19 37 1 1 ...
##  $ Father.s.qualification                        : int  12 3 37 37
38 37 38 37 1 19 ...
```

```
##  $ Mother.s.occupation                           : int  5 3 9 5 9 9 
7 9 9 4 ...
##  $ Father.s.occupation                           : int  9 3 9 3 9 7 
10 9 9 7 ...
##  $ Admission.grade                               : num  127 142 125 
120 142 ...
##  $ Displaced                                     : int  1 1 1 1 0 0 
1 1 0 1 ...
##  $ Educational.special.needs                     : int  0 0 0 0 0 0 
0 0 0 0 ...
##  $ Debtor                                        : int  0 0 0 0 0 1 
0 0 0 1 ...
##  $ Tuition.fees.up.to.date                       : int  1 0 0 1 1 1 
1 0 1 0 ...
##  $ Gender                                        : int  1 1 1 0 0 1 
0 1 0 0 ...
##  $ Scholarship.holder                            : int  0 0 0 0 0 0 
1 0 1 0 ...
##  $ Age.at.enrollment                             : int  20 19 19 20 
45 50 18 22 21 18 ...
##  $ International                                 : int  0 0 0 0 0 0 
0 0 1 0 ...
##  $ Curricular.units.1st.sem..credited.           : int  0 0 0 0 0 0 
0 0 0 0 ...
##  $ Curricular.units.1st.sem..enrolled.           : int  0 6 6 6 6 5 
7 5 6 6 ...
##  $ Curricular.units.1st.sem..evaluations.        : int  0 6 0 8 9 
10 9 5 8 9 ...
##  $ Curricular.units.1st.sem..approved.           : int  0 6 0 6 5 5 
7 0 6 5 ...
##  $ Curricular.units.1st.sem..grade.              : num  0 14 0 13.4 
12.3 ...
##  $ Curricular.units.1st.sem..without.evaluations.: int  0 0 0 0 0 0 
0 0 0 0 ...
##  $ Curricular.units.2nd.sem..credited.           : int  0 0 0 0 0 0 
0 0 0 0 ...
##  $ Curricular.units.2nd.sem..enrolled.           : int  0 6 6 6 6 5 
8 5 6 6 ...
##  $ Curricular.units.2nd.sem..evaluations.        : int  0 6 0 10 6 
17 8 5 7 14 ...
##  $ Curricular.units.2nd.sem..approved.           : int  0 6 0 5 6 5 
8 0 6 2 ...
##  $ Curricular.units.2nd.sem..grade.              : num  0 13.7 0 
12.4 13 ...
##  $ Curricular.units.2nd.sem..without.evaluations.: int  0 0 0 0 0 5 
0 0 0 0 ...
##  $ Unemployment.rate                             : num  10.8 13.9 
10.8 9.4 13.9 16.2 15.5 15.5 16.2 8.9 ...
##  $ Inflation.rate                                : num  1.4 -0.3 
1.4 -0.8 -0.3 0.3 2.8 2.8 0.3 1.4 ...
```

```
##  $ GDP                                           : num  1.74 0.79
1.74 -3.12 0.79 -0.92 -4.06 -4.06 -0.92 3.51 ...
##  $ Target                                        : chr  "Dropout"
"Graduate" "Dropout" "Graduate" ...
```

All variables except Target are numeric.

Convert Target from character to categorical.

```
aca2 = aca1 %>%
  mutate(Target = as.factor(Target))
```

Although all features are encoded as numeric, some of them represent categorical data. I will drop these features for PCA.

```
aca3 = aca2[c(7, 13, 20, 22:37)]
```

Scale the data.

```
aca.scale = scale(aca3[1:18])
```

Perform PCA.

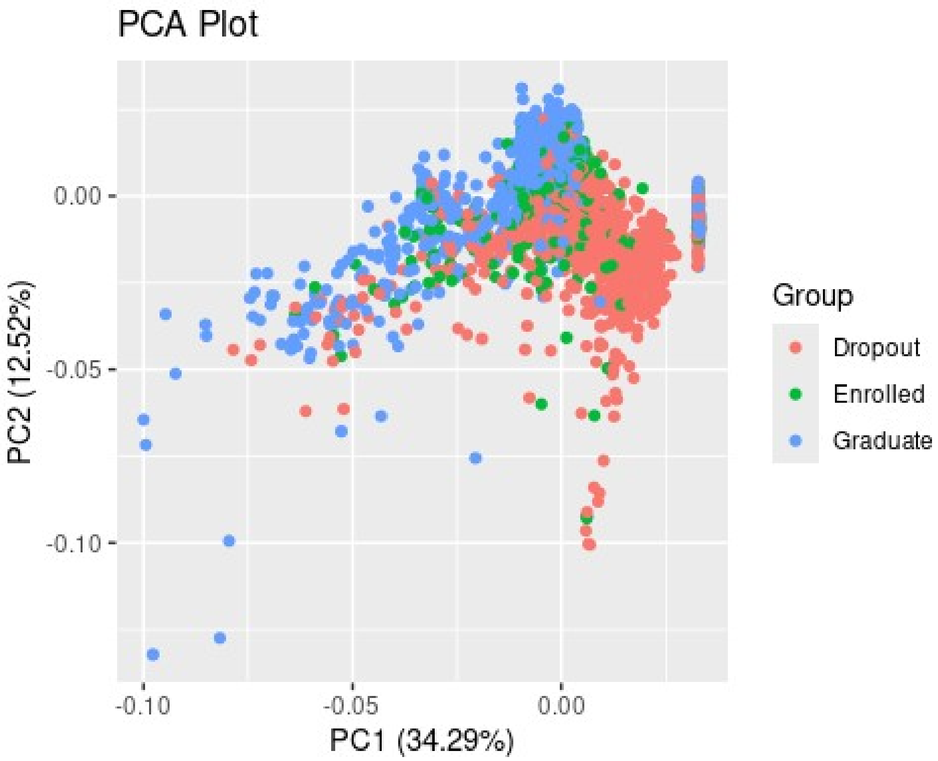
```
aca.pca = prcomp(aca.scale)
summary(aca.pca)

## Importance of components:
##                            PC1    PC2     PC3     PC4     PC5
PC6     PC7
## Standard deviation     2.4844 1.5011 1.27133 1.23484 1.14573
1.00481 0.93463
## Proportion of Variance 0.3429 0.1252 0.08979 0.08471 0.07293
0.05609 0.04853
## Cumulative Proportion  0.3429 0.4681 0.55788 0.64260 0.71553
0.77162 0.82015
##                             PC8     PC9    PC10    PC11    PC12
PC13    PC14
## Standard deviation     0.84767 0.79391 0.66307 0.64104 0.60613
0.46180 0.41598
## Proportion of Variance 0.03992 0.03502 0.02443 0.02283 0.02041
0.01185 0.00961
## Cumulative Proportion  0.86006 0.89508 0.91951 0.94234 0.96275
0.97459 0.98421
##                            PC15    PC16    PC17    PC18
## Standard deviation     0.36779 0.30361 0.18908 0.14515
## Proportion of Variance 0.00751 0.00512 0.00199 0.00117
## Cumulative Proportion  0.99172 0.99684 0.99883 1.00000
```

The first 12 principal components explain over 96% of the variance.

Plot PCA.

```
autoplot(aca.pca, data = aca3,
         colour = 'Target',
         label = F) +
  ggtitle('PCA Plot') +
  labs(colour = 'Group')
```



Calculate total variance explained by each principal component.

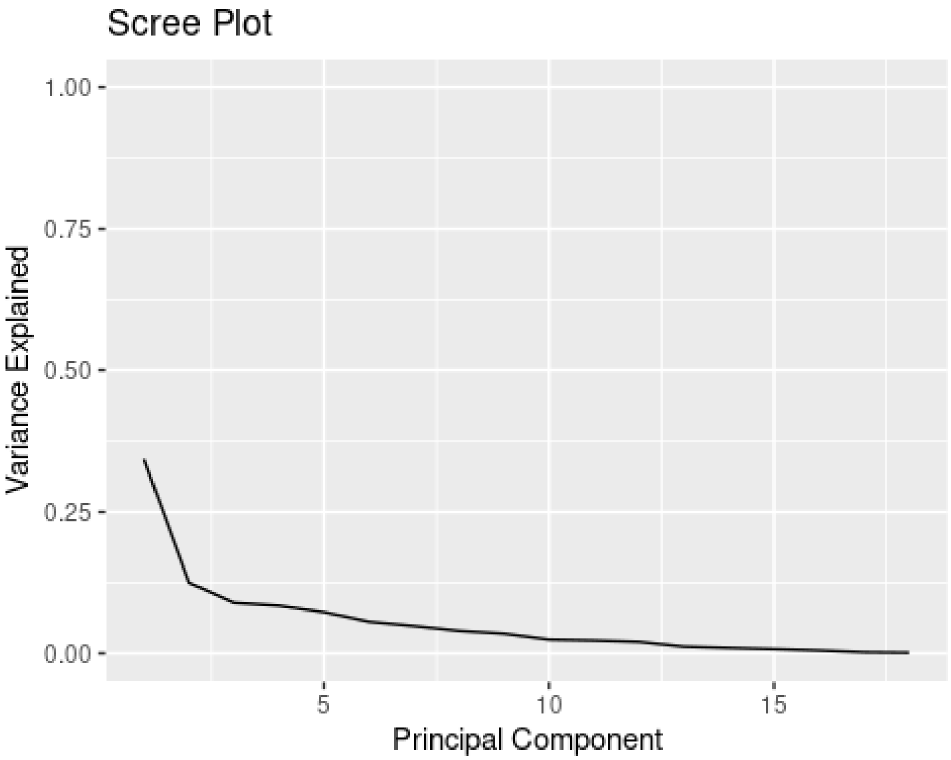
```
var_explained = aca.pca$sdev^2 / sum(aca.pca$sdev^2)
```

Make a dataframe for a scree plot.

```
scree.df = cbind.data.frame(1:length(colnames(aca.pca$x)),
var_explained)
colnames(scree.df) = c('PC', 'Var')
```

Make a scree plot.

```
ggplot(scree.df, aes(x = PC, y = Var)) +
  geom_line() +
  xlab("Principal Component") +
  ylab("Variance Explained") +
  ggtitle("Scree Plot") +
  ylim(0, 1)
```


Scree Plot
Variance Explained
1.00
0.75
0.50
0.25
0.00
5
10
15
Principal Component