post01

Yawen Sun

10/21/2017

```
# packages
library(readr)     # importing data
library(dplyr)     # data wrangling
```

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

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

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

```
library(ggplot2)  # graphics

setwd("/Users/sunsebrina/Documents/fall 2017courses/stat133/stat133-hws-fall17/post01")
```

Introduction:

In the lecture, the professor gave us an introduction on principal components analysis(PCA). In hw03, we also did exercises on PCA to rank NBA teams on scaled PC1. In this post, I will introduce more on principal components analysis(PCA), and its use in statistics and machine learning.

Motivation:

PCA is a hard topic for beginners. Although I met it both in the lecture and hw3, I am still confused about how to use this method to help us analyze data. Thus, I will explore more on this topic in post01.

Background:

Given a data set, a table, we can analyze it in two perspectives: one is objects – rows of the table, the other is variables – the columns of the table. We can study relationship among column variables and similarity between individual objects to explore a specific data set.

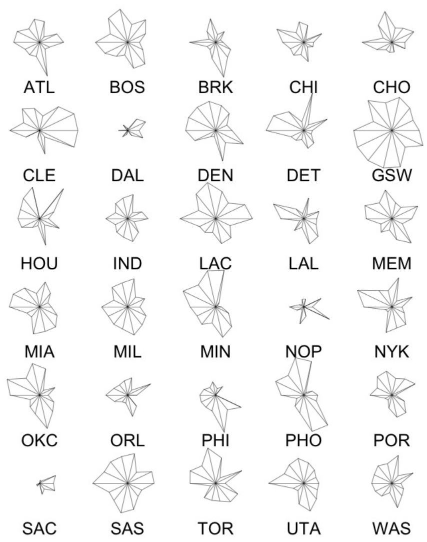
To find relationship among column variables, we can calculate correlations between each two variables.

To find out resemblance among individuals. We can draw star plots of objects. See example of a star plot of NBA teams:

```
teams <- data.frame(read_csv(file = "data/nba2017-teams.csv"))
```

```
## Parsed with column specification:
## cols(
##   team = col_character(),
##   experience = col_integer(),
##   salary = col_double(),
##   points3 = col_integer(),
##   points2 = col_integer(),
##   free_throws = col_integer(),
##   points = col_integer(),
##   off_rebounds = col_integer(),
##   def_rebounds = col_integer(),
##   assists = col_integer(),
##   steals = col_integer(),
##   blocks = col_integer(),
##   turnovers = col_integer(),
##   fouls = col_integer(),
##   efficiency = col_double()
## )
```

```
# star plot of the teams
stars(teams[ ,-1], labels = teams$team)
```



However, to summarize the systematic variation of the variables, we see methods including calculating correlations, standardizing the variables, or using correlations of transformed and standardized variables.

A better way to study the relationship of variables is to use principal componets analysis(PCA), a multivariate method and to look at PCs: linear combinations of the original variables.

We often see very high-dimensional data, so we use PCA as an unsupervised dimensionaly reduction technique to reduce dimensionality of data. A lower-dimensional representation of data is useful because it is easier for visualization (with 2 or 3 dimensions), and reduce computational load and noise in machine learning.

Given a matrix of data points, PCA finds one or more orthogonal directions that capture the largest amount of variance in the data, so PCA can reduce the dimensionality of a data set while keeping as much as possible of the variation present in the data. Intuitively, the directions with less variance contain less information and may be discarded without introducing too much error.

Steps of calculation: First, we subtract the mean to make the data points zero-mean. Then, we transform the original variables into a smaller set of new variables that summarize the variation in data – the principal componets, which can be seen as linear combinations of the original variables. After diagonaliztion, we do eigenvalue decomposition(EVD) of data.

Examples (still working with NBA teams):

Perform a principal components analysis (PCA)

```
part <- select(teams, points3, points2, free_throws, free_throws, off_rebounds, def_rebounds, assists, steals, blo
cks, turnovers, fouls)
pca <- prcomp(part, scale. = TRUE)

eigs <- data.frame(
  eigenvalue = round(pca$sdev^2, 4),
  prop = round(pca$sdev^2 / sum(pca$sdev^2), 4)
)
eigs <- mutate(eigs, cumprop = cumsum(prop))

eigs
```

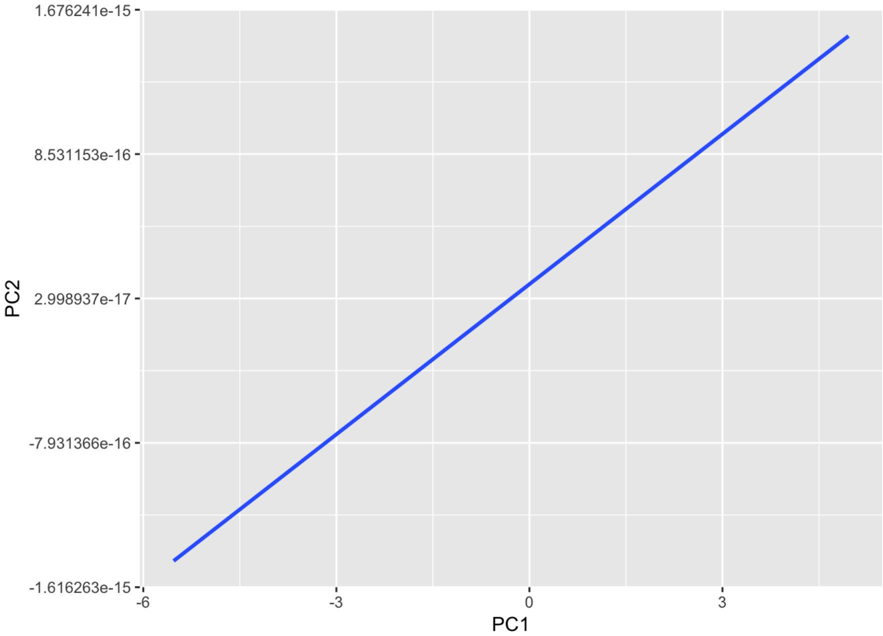
```
##    eigenvalue   prop cumprop
## 1      4.6959 0.4696  0.4696
## 2      1.7020 0.1702  0.6398
## 3      0.9795 0.0980  0.7378
## 4      0.7717 0.0772  0.8150
## 5      0.5341 0.0534  0.8684
## 6      0.4780 0.0478  0.9162
## 7      0.3822 0.0382  0.9544
## 8      0.2603 0.0260  0.9804
## 9      0.1336 0.0134  0.9938
## 10     0.0627 0.0063  1.0001
```

```
pc <- data.frame(
  PC1 = pca$x[,1],
  PC2 = pca$x[,2],
  name = teams$team
)
pc
```

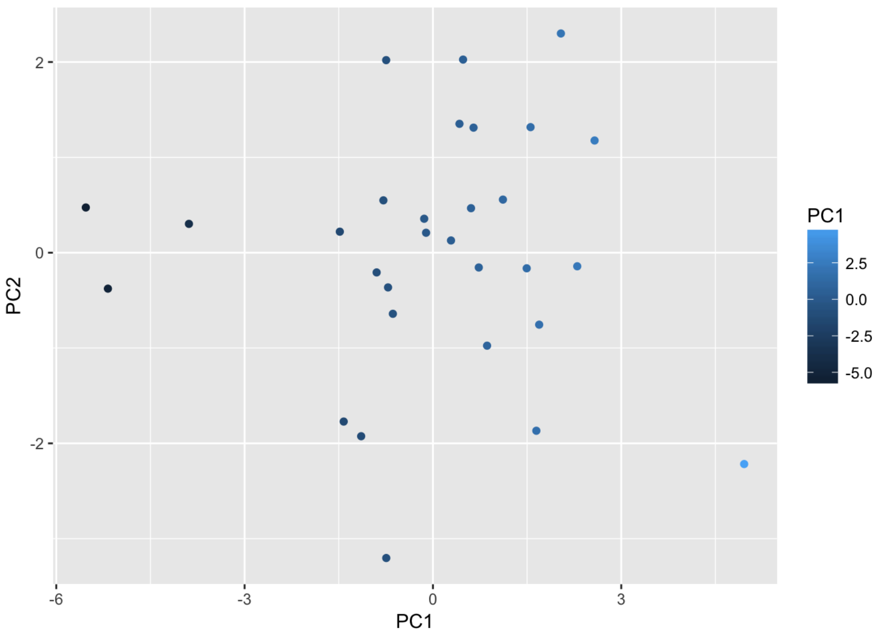
```
##            PC1        PC2 name
## 1   0.2883171  0.1281265  ATL
## 2   1.6475677 -1.8678932  BOS
## 3  -0.6378694 -0.6410895  BRK
## 4  -0.7889514  0.5491124  CHI
## 5  -1.4213891 -1.7716179  CHO
## 6  -1.1429197 -1.9254795  CLE
## 7  -5.1770470 -0.3771922  DAL
## 8   0.8628216 -0.9755539  DEN
## 9   0.4228059  1.3520635  DET
## 10  4.9580722 -2.2173199  GSW
## 11 -0.7434842 -3.2031420  HOU
## 12 -0.1393098  0.3561238  IND
## 13  1.6926408 -0.7550453  LAC
## 14 -0.7449230  2.0200116  LAL
## 15  0.6071090  0.4667924  MEM
## 16  1.1154708  0.5570744  MIA
## 17  1.4939629 -0.1637954  MIL
## 18  2.5754284  1.1769429  MIN
## 19 -3.8867632  0.3023898  NOP
## 20  0.4804728  2.0259452  NYK
## 21  1.5554071  1.3170619  OKC
## 22 -1.4831168  0.2204544  ORL
## 23 -0.7149664 -0.3641317  PHI
## 24  2.0387934  2.2997473  PHO
## 25 -0.8965058 -0.2071566  POR
## 26 -5.5291364  0.4742780  SAC
## 27  2.2990719 -0.1427248  SAS
## 28  0.6469827  1.3120040  TOR
## 29  0.7307586 -0.1550934  UTA
## 30 -0.1093009  0.2091070  WAS
```

We can explore the data by drawing some graphs:

```
ggplot(data = pc, aes(x = PC1, y = PC2)) +
   geom_smooth(method = "lm", se = FALSE)
```

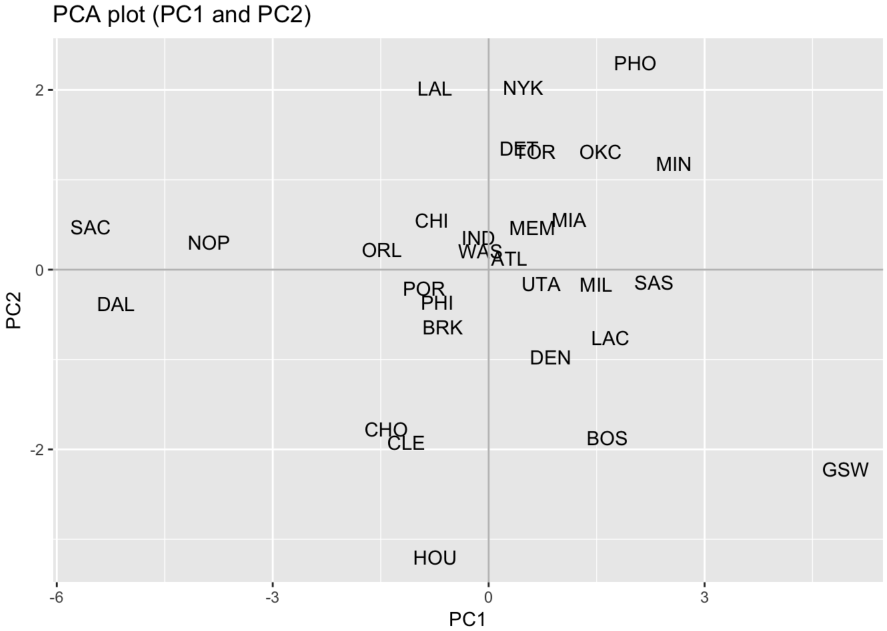



```
ggplot(data = pc, aes(x = PC1, y = PC2, color = PC1)) +
   geom_point()
```



Use the first two PCs to get a scatterplot of the teams

```
ggplot(data = pc, aes(x = PC1, y = PC2)) +
  geom_text(aes(label = name)) +
  ggtitle("PCA plot (PC1 and PC2)") +
  geom_hline(aes(yintercept = 0), color ="gray") +
  geom_vline(aes(xintercept = 0), color ="gray")
```

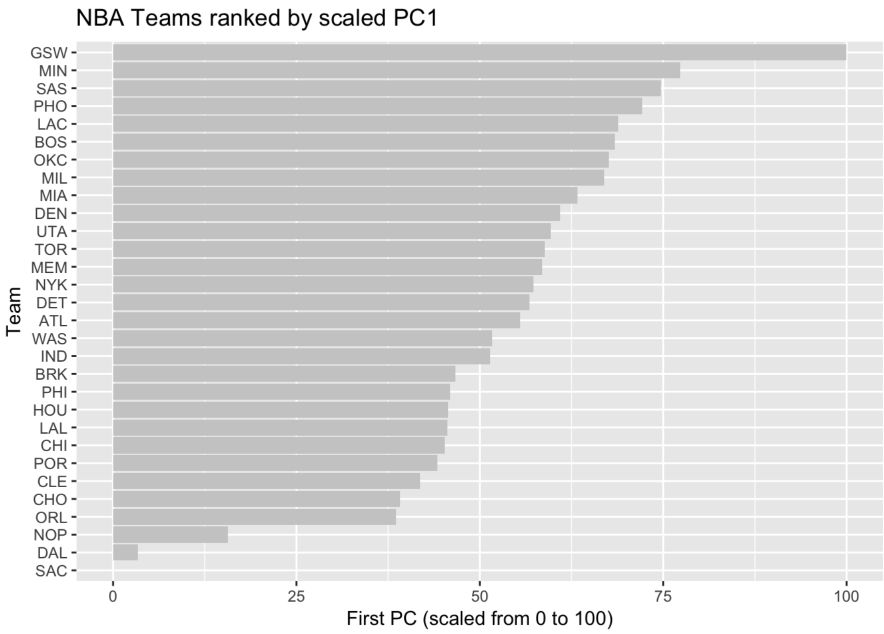


Index based on PC1

```
pc <- mutate(pc, s1 = 100*(PC1 - min(PC1))/(max(PC1) - min(PC1)))
pc
```

```
##          PC1        PC2 name         s1
## 1   0.2883171  0.1281265  ATL  55.471897
## 2   1.6475677 -1.8678932  BOS  68.432930
## 3  -0.6378694 -0.6410895  BRK  46.640314
## 4  -0.7889514  0.5491124  CHI  45.199683
## 5  -1.4213891 -1.7716179  CHO  39.169120
## 6  -1.1429197 -1.9254795  CLE  41.824445
## 7  -5.1770470 -0.3771922  DAL   3.357322
## 8   0.8628216 -0.9755539  DEN  60.950042
## 9   0.4228059  1.3520635  DET  56.754305
## 10  4.9580722 -2.2173199  GSW 100.000000
## 11 -0.7434842 -3.2031420  HOU  45.633232
## 12 -0.1393098  0.3561238  IND  51.394292
## 13  1.6926408 -0.7550453  LAC  68.862721
## 14 -0.7449230  2.0200116  LAL  45.619512
## 15  0.6071090  0.4667924  MEM  58.511713
## 16  1.1154708  0.5570744  MIA  63.359160
## 17  1.4939629 -0.1637954  MIL  66.968243
## 18  2.5754284  1.1769429  MIN  77.280477
## 19 -3.8867632  0.3023898  NOP  15.660728
## 20  0.4804728  2.0259452  NYK  57.304183
## 21  1.5554071  1.3170619  OKC  67.554140
## 22 -1.4831168  0.2204544  ORL  38.580520
## 23 -0.7149664 -0.3641317  PHI  45.905162
## 24  2.0387934  2.2997473  PHO  72.163434
## 25 -0.8965058 -0.2071566  POR  44.174106
## 26 -5.5291364  0.4742780  SAC   0.000000
## 27  2.2990719 -0.1427248  SAS  74.645300
## 28  0.6469827  1.3120040  TOR  58.891926
## 29  0.7307586 -0.1550934  UTA  59.690765
## 30 -0.1093009  0.2091070  WAS  51.680440
```

```
# create an associated bar chart
ggplot(data = pc, aes(x = reorder(name, s1), y = s1)) +
  geom_bar(stat='identity', fill = 'grey80') +
  coord_flip() +
  labs(title = "NBA Teams ranked by scaled PC1",
       x = "Team", y = "First PC (scaled from 0 to 100)")
```



Discussion:

There are many pca functions and packages in R:

prcomp(), princomp(), PCA(), etc.

Conclusion:

PCA is a dimentinality reduction (eg:reduce a data set with dimention 4 to dimention 2) to help us better explore the data.

References:

1. “data/nba2017-teams.csv”, which is from hw03
2. Intro to PCA, lecture slide: https://github.com/ucb-stat133/stat133-fall-2017/blob/master/slides/15-principal-components1.pdf
3. Principal Component Analysis note from Berkeley cs189 course website: http://www.eecs189.org/static/notes/n9.pdf

4. Wikepedia on PCA: https://en.wikipedia.org/wiki/Principal_component_analysis#First_component
5. Overview video of Principal Componets Analysis (PCA) and why use PCA as part of your machine learning toolset: https://www.youtube.com/watch?v=NLrb41ls4qo
6. Another overview: https://www.utdallas.edu/~herve/abdi-awPCA2010.pdf
7. Brief introduction: http://www.itl.nist.gov/div898/handbook/pmc/section5/pmc55.htm
8. ftp://statgen.ncsu.edu/pub/thorne/molevoclass/AtchleyOct19.pdf