Statistical Analysis using R

IITA
Transforming African Agriculture

Multivariate analysis: Principal Component Analysis

Ibnou Dieng
Kayode Fowobaje
Sam Ofodile
Moshood Bakare
Oluwafemi Oyedele

IITA Biometrics Unit

01-02 & 07-09 December 2021

Course overview

1. ~~Short Introduction to R and RStudio~~
2. ~~Preparation of Data for Statistical Analysis~~
3. ~~Data wrangling~~
4. ~~Experimental Designs for Plant Breeding~~
5. ~~ANOVA and MET analysis~~
6. Multivariate analysis
7. Graphics in R with ggplot2

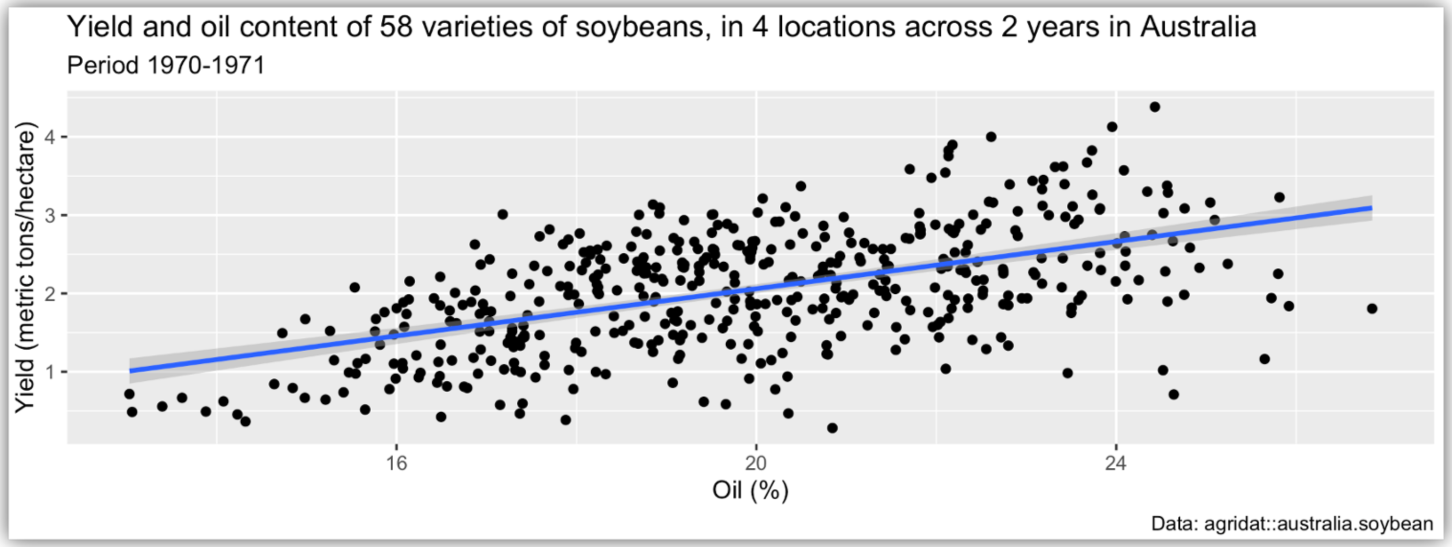
PCA

- Let's consider the `australia.soybean`, a multi-environment trial of 58 varieties of soybeans, in 4 locations across 2 years in Australia.
    - **env** (environment), 8 levels, location-year
    - **loc** (location)
    - **year**
    - **gen** (genotype) of soybeans,
    - **yield**, metric tons/hectare
    - **height**, meters
    - **lodging**
    - **size seed**, millimetres
    - **protein**, percentage
    - **oil**, percentage

```r
library(tidyverse)
library(agridat)
data(australia.soybean)
dat <- australia.soybean %>%
  as_tibble()
```

PCA

- To study the relationship between two variables, say `yield` and `oil`, we can use
  - Pearson correlation coefficient
  - Simple linear regression



PCA

- With multivariate data (`yield`, `height`, `lodging`, `seed size`, `protein`, and `oil`), how can we easily see and visualize the relationships between the variables all together?
- Principal component analysis (PCA) is a useful technique for exploratory data analysis, allowing to better visualize the variation present in a data set with many variables
- It is particularly useful in the case of large data sets, where there are many variables: difficult to represent all the data in their raw format, which makes it difficult to understand the trends present inside

PCA

- PCA allows to see the general `shape` of a data, identifying which observations are similar

- This can allow us to identify groups of similar observations and determine which variables make one group different from another

- The basics of PCA:
  - take a dataset with many variables
  - reduce the number of variables to a smaller number of `principal components`
  - visualize the correlations between the variables
  - visualize the similarities between observations (individuals)

PCA

- The principal components are the `directions` where there is the most variance, the directions where the data is the most distributed
- This means that we are trying to find the straight line that best distributes the data as it is projected along it
- The first main component is the straight line which shows the largest variation in the data
- PCA is a type of linear transformation that adapts a dataset to a new coordinate system such that the most significant variance is on the first coordinate, and each subsequent coordinate is orthogonal to the last and has a variance lesser
- In this way, we transform a set of `x` correlated variables on `y` observations into a set of `p` uncorrelated principal components on the same observations

PCA

- When many variables are correlated, they will all contribute strongly to the same principal component

- Each principal component summarizes a certain percentage of the total variation in the data set. When the initial variables are highly correlated with each other, you will be able to approach most of the complexity of your dataset with just a few main components

- Having additional components makes the visualization of the data set more accurate, but also cumbersome

PCA

- An `eigenvector` is a direction such as horizontal 'or 30 degrees', while an `eigenvalue` is a number indicating the variance of the data in that direction
- The `eigenvector` with the largest `eigenvalue` is, therefore, the first principal component
- The `eigenvectors` and `eigenvalues` are linked: each `eigenvector` has a corresponding `eigenvalue`

PCA

- The total number of principal components is equal to the total number of variables
- As PCA aims to reduce the information of a large data set, it would be unnecessary and even tedious to consider/interpret all the principal components
- **But then how many principal components to retain?**
    - Keep the number of components that explains a satisfactory amount of the total variance
    - In other words, select the number of axes for which the cumulative percentage of variance is high enough (more than 50%)
    - Identify the eigenvalues on the histogram where the most obvious change in slope occurs (sharp decrease in inertia)
    - Keep the principal components for which the variance (or the eigenvalues) is greater than 1

PCA

- The graph of the variables is interpreted in terms of correlation. When two variables have an angle close to:
  - 0, positive correlation
  - 90, no correlation
  - 180, negative correlation
- The graph of observations is interpreted in terms of proximity

PCA

- Let's consider one location `Brookstead` in the `australia.soybean` data

```r
dat.Brookstead <- dat %>%
  filter(loc=="Brookstead")
str(dat.Brookstead)
```

```
## tibble [116 × 10] (S3: tbl_df/tbl/data.frame)
##  $ env    : Factor w/ 8 levels "B70","B71","L70",..: 1 1 1 1 1 1 1 1 1 1 .
##  $ loc    : Factor w/ 4 levels "Brookstead","Lawes",..: 1 1 1 1 1 1 1 1 1
##  $ year   : int [1:116] 1970 1970 1970 1970 1970 1970 1970 1970 1970 1970
##  $ gen    : Factor w/ 58 levels "G01","G02","G03",..: 1 2 3 4 5 6 7 8 9 10
##  $ yield  : num [1:116] 1.253 1.167 0.468 1.445 1.338 ...
##  $ height : num [1:116] 1.01 1.13 1.16 1.24 1.12 ...
##  $ lodging: num [1:116] 3.25 2.75 2.25 1.5 2 2.25 2 2.25 1.75 2 ...
##  $ size   : num [1:116] 8.85 8.9 10.8 10.6 11.95 ...
##  $ protein: num [1:116] 39.5 38.6 37.8 38.7 37.8 ...
##  $ oil    : num [1:116] 18.9 19.8 20.4 20.4 20.8 ...
```

PCA

- We store the names of the genotypes in a vector

```
labels.gen <- dat.Brookstead$gen
```

- And only consider the quantitative variables

```
dat.Brookstead <- dat.Brookstead %>%
  select(-c(env, loc, year, gen))
```

PCA

- We run the PCA using the prcomp function

```
dat.Brookstead.pca<-prcomp(dat.Brookstead, center=TRUE, scale=TRUE)
summary(dat.Brookstead.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     1.8780 0.9949 0.9176 0.60462 0.42702 0.30580
## Proportion of Variance 0.5878 0.1650 0.1403 0.06093 0.03039 0.01559
## Cumulative Proportion  0.5878 0.7528 0.8931 0.95402 0.98441 1.00000
```

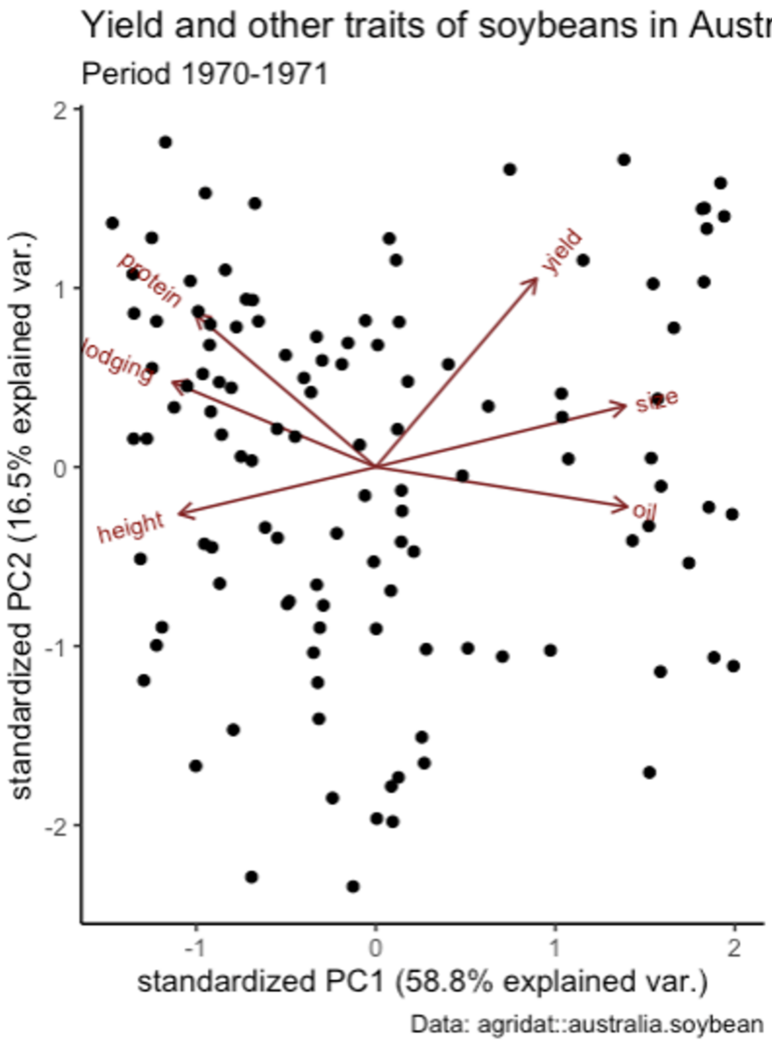
- The first 2 axes explain more than 75% of the information in the dataset

PCA

- We visualize the PCA using a `biplot`
- A `biplot` is a type of graph that allows simultaneously visualize the observations and the variables
- We use the `ggbiplot` package to visualize the biplot

```r
library(ggbiplot)
ggbiplot(dat.Brookstead.pca) +
  labs(title = "Yield and other traits of soybeans in Australia",
       subtitle = "Period 1970-1971",
       caption = "Data: agridat::australia.soybean"
  ) +
  theme_classic()
```

PCA


Yield and other traits of soybeans in Australia
Period 1970-1971
standardized PC2 (16.5% explained var.)
standardized PC1 (58.8% explained var.)
2
1
0
-1
-2
-1
0
1
2
protein
lodging
height
yield
size
oil
Data: agridat::australia.soybean

PCA

- The components are seen as arrows coming from the origin. Here, all the variables contribute to PC1, with:
- higher values of `oil`, `size` and `yield` to the right on this graph and lower values to the left
- vice versa for `protein`, `lodging` and `height`
- This allows to see how the observations relate to the components
- However, this is not very informative if we do not identify the observations

PCA

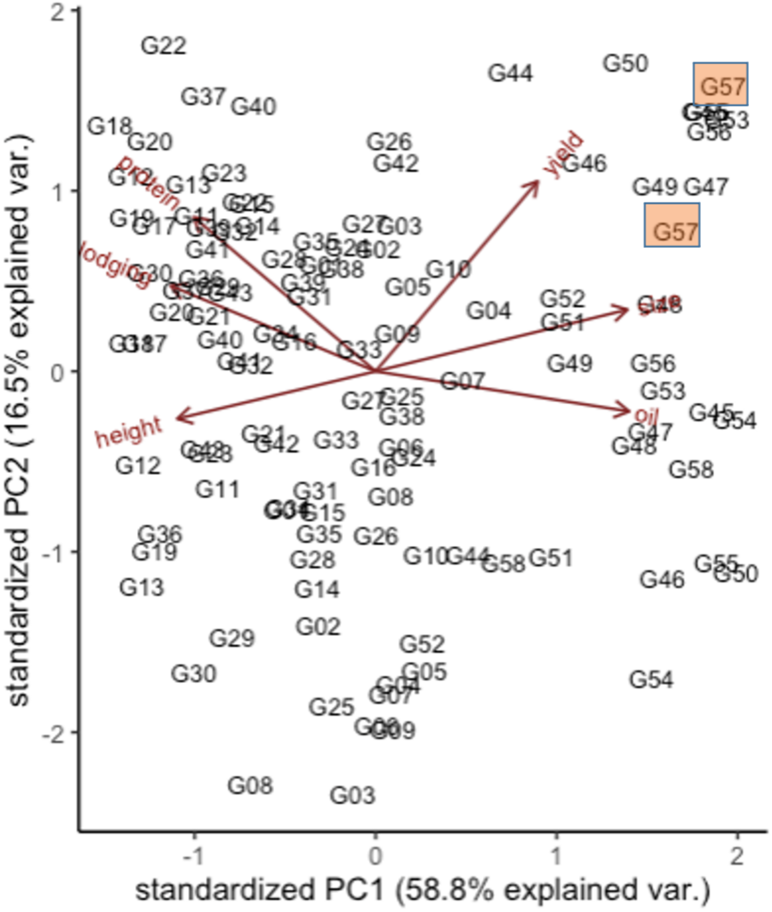
- We can display the names of the genotypes

```
library(ggbiplot)
ggbiplot(dat.Brookstead.pca, labels=labels.gen) +
  labs(title = "Yield and other traits of soybeans in Australia",
       subtitle = "Period 1970-1971",
       caption = "Data: agridat::australia.soybean"
  ) +
  theme_classic()
```

PCA

Yield and other traits of soybeans in Australia

Period 1970-1971



Data: agridat::australia.soybean

PCA

- We have two years of testing, 1970 and 1971
- Could be interesting to identify the genotypes by year

```r
library(ggbiplot)
ggbiplot(dat.Brookstead.pca, labels=labels.gen, groups=labels.year)+
  labs(title = "Yield and other traits of soybeans in Australia",
       subtitle = "Period 1970-1971",
       caption = "Data: agridat::australia.soybean"
  ) +
  theme_classic()
```

PCA