Introduction to Biostatistics
The 1st Lecture

Hiroyoshi IWATA hiroiwata@g.ecc.u-tokyo.ac.jp

2021/4/9

Introduction

Recently, in the fields of agriculture and life sciences, various types of data are being collected and accumulated in large quantities. It is necessary to analyze the data using methods appropriate to the purpose of the research and the nature of the data, in order to ensure that we do not miss the "undiscovered knowledge" harbored in the data.

There are various methods for statistical analysis. In order to understand the features of the methods and the principles of analysis and also to interpret the analysis results properly, you should learn them accordingly. Also, in order to make your learning more effective, it is essential to have the experience of analyzing actual data. It is often the case that you can clearly understand what you have learned in lectures and textbooks when you analyze your data by yourself.

This course is intended to provide you with a "first step" to analyze their own data and improve their statistical analysis skills. Specifically, we will focus on practical data analysis methods using R, focusing on some of the statistical methods that may be required in your future research. The goal of this lecture is to acquire the skills to use it for data analysis of general-purpose statistical analysis methods, e.g., regression analysis, analysis of variance, and principal component analysis. Furthermore, the goal is to build a foothold to perform more advanced data analysis. Although it is a short course with four lectures, I will provide you this course so that you can be interested in the joy and skills of statistical analysis.

R

R is free software for statistical analysis. (To put it a little correctly, R is the name of a computer language. R installed on a PC as software is an "environment" for using R language). R has many functions, and its usages range from statistical analysis to pre-processing of data, overview of data, and creation of graphs for papers. In addition, various analysis can be easily performed by installing an extension program distributed as a "package". Newly developed statistical methods will be available quickly in R. Thus, the skills for using R have become useful to researchers in agronomics and life sciences.

In addition, for learning R, a large number of reference books are available. Introductory books I recommend are:

- Peter Dalgaard, Introductory Statistics with R (Statistics and Computing) Second Edition, Springer, 2008, ISBN: 978-0387790534

- Brian Everitt, Torsten Hothorn, An Introduction to Applied Multivariate Analysis with R (Use R!), Springer, 2011, ISBN: 978-1441996497

Simple calculation using R

In analysis using R, the analysis is basically progressed interactively while sequentially inputting commands (However, when you actually perform analysis, it is useful to prepare a series of commands, as an R script, prior to the analysis. Remember that it is more convenient to execute the R script, because it allows us to do partial corrections and to review the history of analysis).

Let's start with getting used to R while doing simple calculations with command input.

The easiest way to use R is to enter a simple arithmetic expression and get the answer. For example,

```
3 + 5 * 3
```

```
## [1] 18
```

If you want to perform the next calculation based on the obtained result, assign the resulted value to some variable as follows.

```
x <- 1 + 2
x
```

```
## [1] 3
```

The assigned value can be used for another calculation through the variable name.

```
x + 5 * x
```

```
## [1] 18
```

Various calculations can be performed using functions.

```
abs(x)
```

```
## [1] 3
```

```
sin(x)
```

```
## [1] 0.14112
```

```
atan(x)
```

```
## [1] 1.249046
```

```
log(x)
```

```
## [1] 1.098612
```

```
log10(x)
```

```
## [1] 0.4771213
```

Let's perform a bit more complicated calculation. The probability density function of the normal distribution of mean μ and the variance σ^2 (Figure 1) is

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Calculate this with R.

```r
mu <- 3
s2 <- 2
x <- 5
1 / sqrt(2 * pi * s2) * exp(- (x - mu)^2 / (2 * s2))

## [1] 0.1037769
```
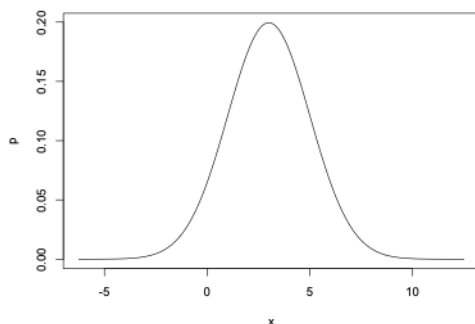

Fig1. Normal distribution with mean 3 and variance 2

If you use the "dnorm" function to calculate the probability density of the normal distribution for confirming your calculation, you can get the same result.

```r
dnorm(x, mu, sqrt(s2))

## [1] 0.1037769
```

Quiz1

Now, let's solve a practice question here. Practice questions will be presented in the lecture.

Go to https://www.menti.com/d9p29idxtw . Then, register your nickname and wait for the quiz to start.

Calculation using vector or matrix

A great advantage of R is that we can perform vector and matrix operations easily. Let's calculate some summary statistics using vector and matrix operations.

For example, we can easily create a vector of six elements as follows: this is the data which measured the grain length of six varieties/lines of rice in mm unit (the source of the data will be described later).

```r
length <- c(8.1, 7.7, 8.2, 9.7, 7.1, 7.3)    # mm scale
length

## [1] 8.1 7.7 8.2 9.7 7.1 7.3
```

Input grain widths of the same varieties/lines, and calculate the length-width ratio.

```r
width <- c(3.7, 3.0, 2.9, 2.4, 3.3, 2.5)
ratio <- length / width
ratio
```

```
## [1] 2.189189 2.566667 2.827586 4.041667 2.151515 2.920000
```

First, let's calculate the average of the length-width ratio of grains. The estimate of the population mean is

$$\sum_{i=1}^{n} x_i / n$$

where x_i is the value of the ith sample and n is the number of samples.

```
sum(ratio)
```

```
## [1] 16.69662
```

```
length(ratio)
```

```
## [1] 6
```

```
sum(ratio) / length(ratio)
```

```
## [1] 2.782771
```

The mean can be calculated using the "mean" function.

```
mean(ratio)
```

```
## [1] 2.782771
```

Next, let's calculate the variance. An estimate of the population variance is

$$\sum_{i=1}^{n} (x_i - \overline{x})^2 / (n - 1)$$

where $\overline{x}$ is the average calculated earlier.

```
xbar <- mean(ratio)
(ratio - xbar)^2
```

```
## [1] 0.352338947 0.046700930 0.002008434 1.584819189 0.398483500 0.01883189
5
```

```
sum((ratio - xbar)^2)
```

```
## [1] 2.403183
```

```
sum((ratio - xbar)^2) / (length(ratio) - 1)
```

```
## [1] 0.4806366
```

The variance can be calculated using the "var" function.

```
var(ratio)
```

```
## [1] 0.4806366
```

Next, let's calculate the covariance. An estimate of the covariance between bivariate x and y is

$$\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})/(n-1)$$

where $\overline{x}$ and $\overline{y}$ represent the mean of each variable.

```
xbar <- mean(length)
ybar <- mean(width)
sum((length - xbar) * (width - ybar)) / (length(length) - 1)
```

```
## [1] -0.1773333
```

The variance can be calculated using the "var" function.

```
cov(length, width)
```

```
## [1] -0.1773333
```

Let's calculate Pearson product moment correlation coefficient (hereinafter referred to just as correlation coefficient). The correlation coefficient is

$$\frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

```
s12 <- sum((length - xbar) * (width - ybar))
s1 <- sum((length - xbar)^2)
s2 <- sum((width - ybar)^2)
s12 / (sqrt(s1) * sqrt(s2))
```

```
## [1] -0.3901388
```

As you can see in the equation, the correlation coefficient is the covariance divided by the standard deviation of both variables. Let's actually calculate it and check the result.

```
cov(length, width) / (sd(length) * sd(width))
```

```
## [1] -0.3901388
```

The correlation coefficient is standardized by dividing it by the standard deviation of both variables. Unlike the covariance, we can understand the relationship between variables without being influenced by the scale of the measurement value. Thus, it is suitable for comparing the strength of relationships between variables measured at different scales (such as weight and length).

Note that we can also calculate the correlation coefficient using the "cor" function.

```
cor(length, width)
```

```
## [1] -0.3901388
```

Now let's calculate variance and covariance using matrix calculations. First, combine length and width to create a 6 by 2 matrix.

```
x <- cbind(length, width)
x
```

```
##      length width
## [1,]    8.1   3.7
```

```
## [2,]    7.7    3.0
## [3,]    8.2    2.9
## [4,]    9.7    2.4
## [5,]    7.1    3.3
## [6,]    7.3    2.5
```

Then we use the function apply to find the average of each column.

```
m <- apply(x, 2, mean)
m

##    length     width
## 8.016667  2.966667
```

Subtract the column average from each column.

```
z <- sweep(x, 2, m)
z

##             length          width
## [1,]   0.08333333   0.73333333
## [2,]  -0.31666667   0.03333333
## [3,]   0.18333333  -0.06666667
## [4,]   1.68333333  -0.56666667
## [5,]  -0.91666667   0.33333333
## [6,]  -0.71666667  -0.46666667
```

After that, we can calculate variance and covariance (variance-covariance matrix) by using the product of the matrix.

```
t(z) %*% z / (nrow(z) - 1)

##              length       width
## length   0.8656667  -0.1773333
## width   -0.1773333   0.2386667
```

The diagonal components are variances, and the nondiagonal components are covariances.

The variance-covariance matrix can be calculated with the cov function.

```
cov(x)

##              length       width
## length   0.8656667  -0.1773333
## width   -0.1773333   0.2386667
```

Quiz2

Now, let's solve a practice question here. Practice questions will be presented in the lecture.

If you close the page of the quiz, go to https://www.menti.com/d9p29idxtw .

Import and analyze external data

If you use R for your own research, I think that most of the time you read and analyze data organized with spreadsheet software. Here, I will explain the procedure for reading data prepared by other software into R and analyzing it. Here, we will use the data which was used in genome-wide association studies of rice genetic resources (Zhao et al. 2011; Nature

Communications 2: 467). The data can be downloaded from Rice Diversity
(http://www.ricediversity.org/data/).

The function "read.csv" is used to read the file saved in csv format.

```
pheno <- read.csv("RiceDiversityPheno.csv")
```

To check the size of imported data or a part of the data as follows.

```
dim(pheno)
```

```
## [1] 413  38
```

```
head(pheno[,1:4])
```

```
##          HybID NSFTVID Flowering.time.at.Arkansas Flowering.time.at.Faridpur
## 1 081215-A05       1                    75.08333                         64
## 2 081215-A06       3                    89.50000                         66
## 3 081215-A07       4                    94.50000                         67
## 4 081215-A08       5                    87.50000                         70
## 5 090414-A09       6                    89.08333                         73
## 6 090414-A10       7                   105.00000                         NA
```

The data have a separate file that describes the origin of each genetic resource. Here, we load
the file and combine it with "pheno" data. First, read the file.

```
line <- read.csv("RiceDiversityLine.csv")
head(line)
```

```
##    GSOR.ID        IRGC.ID NSFTV.ID Accession.Name Country.of.origin  Latitu
de
## 1  301001 To be assigned        1       Agostano             Italy 41.8719
40
## 2  301003         117636        3  Ai-Chiao-Hong             China 27.9025
27
## 3  301004         117601        4        NSF-TV 4             India 22.9030
81
## 4  301005         117641        5        NSF-TV 5             India 30.4726
64
## 5  301006         117603        6        ARC 7229             India 22.9030
81
## 6  301007 To be assigned        7           Arias         Indonesia -0.7892
75
##    Longitude Sub.population     PC1     PC2     PC3     PC4
## 1  12.56738            TEJ -0.0486  0.0030  0.0752 -0.0076
## 2 116.87256            IND  0.0672 -0.0733  0.0094 -0.0005
## 3  87.12158            AUS  0.0544  0.0681 -0.0062 -0.0369
## 4  75.34424       AROMATIC -0.0073  0.0224 -0.0121  0.2602
## 5  87.12158            AUS  0.0509  0.0655 -0.0058 -0.0378
## 6 113.92133            TRJ -0.0293 -0.0027 -0.0677 -0.0085
```

Since NSFTV.ID in the "line" data and NSFTVID in the "pheno" data correspond to each other,
the two data are combined based on the information of these columns.

```
data <- merge(line, pheno, by.x = "NSFTV.ID", by.y = "NSFTVID")
head(data[,1:14])
```

```
##   NSFTV.ID GSOR.ID      IRGC.ID Accession.Name Country.of.origin  Latitu
de
## 1        1  301001 To be assigned      Agostano             Italy 41.8719
40
## 2        3  301003         117636  Ai-Chiao-Hong            China 27.9025
27
## 3        4  301004         117601        NSF-TV 4           India 22.9030
81
## 4        5  301005         117641        NSF-TV 5           India 30.4726
64
## 5        6  301006         117603        ARC 7229           India 22.9030
81
## 6        7  301007 To be assigned          Arias       Indonesia -0.7892
75
##    Longitude Sub.population     PC1     PC2     PC3     PC4      HybID
## 1  12.56738           TEJ -0.0486  0.0030  0.0752 -0.0076 081215-A05
## 2 116.87256           IND  0.0672 -0.0733  0.0094 -0.0005 081215-A06
## 3  87.12158           AUS  0.0544  0.0681 -0.0062 -0.0369 081215-A07
## 4  75.34424       AROMATIC -0.0073  0.0224 -0.0121  0.2602 081215-A08
## 5  87.12158           AUS  0.0509  0.0655 -0.0058 -0.0378 090414-A09
## 6 113.92133           TRJ -0.0293 -0.0027 -0.0677 -0.0085 090414-A10
##   Flowering.time.at.Arkansas
## 1                   75.08333
## 2                   89.50000
## 3                   94.50000
## 4                   87.50000
## 5                   89.08333
## 6                  105.00000
```

Quiz3

Now, let's solve a practice question here. Practice questions will be presented in the lecture.

If you close the page of the quiz, go to https://www.menti.com/d9p29idxtw .

Analysis of the data

We often want to analyze a large number of variables to look at their distributions and relationships among them. Measurement data, howover, often include missing entries with some experimental reasons. Here, we will analyze the data which we load from the csv file.

Let's calculate the length-width ratio of grains and its average in the same way as mentioned above.

```
ratio <- data$Seed.length / data$Seed.width
mean(ratio)
```

```
## [1] NA
```

We cannot calculate the average, and only get the value of "NA". Why is that?

This is because the ratio contains missing values (represented as NA in R).

```
ratio[1:14]
```

```
## [1] 2.188254 2.610704 2.814950 4.075973 2.168927 2.905327 3.229055 2.2706
83
## [9] 2.430681       NA 3.418122 3.061198 4.024255       NA
```

In such cases, specify the option na.rm in the calculation.

```
mean(ratio, na.rm = T)
```

```
## [1] 2.752084
```

Find the mean for all variables in the data with the "sapply" function.

```
sapply(data[, 1:14], mean, na.rm = T)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

##              NSFTV.ID                  GSOR.ID
##          2.340896e+02             3.016027e+05
##               IRGC.ID             Accession.Name
##                    NA                       NA
##      Country.of.origin                 Latitude
##                    NA             2.166591e+01
##             Longitude           Sub.population
##          4.359763e+01                       NA
##                   PC1                      PC2
##         -2.716707e-04            -1.280872e-04
##                   PC3                      PC4
##         -3.072639e-04            -2.106538e-05
##                 HybID Flowering.time.at.Arkansas
##                    NA             8.794439e+01
```

A warning message will be displayed for data that is not numeric data, and the calculated result will be NA.

Using the following command, we can calculate not only averages but also quartiles, minimum values and maximum values for numeric data, and can count up samples belonging to each class for factor data.

```
summary(data[,1:14])
```

```
##      NSFTV.ID          GSOR.ID                 IRGC.ID           Accession.Nam
e
##   Min.   :  1.0   Min.   :301001   To be assigned: 58   Azucena       :  2
```

```
## 1st Qu.:112.0   1st Qu.:301109   117616        :  2   Carolina Gold:  2

## Median :217.0   Median :301215   117638        :  2   Moroberekan  :  2

## Mean   :234.1   Mean   :301603   117756        :  2   N 22         :  2

## 3rd Qu.:322.0   3rd Qu.:301318   117808        :  2   Nipponbare   :  2

## Max.   :652.0   Max.   :312018   (Other)       :343   1021         :  1

##                 NA's   :4        NA's          :  4   (Other)      :402

##       Country.of.origin   Latitude         Longitude        Sub.population
## United States: 39   Min.   :-38.42   Min.   :-102.553   ADMIX   :62
## India        : 34   1st Qu.: 14.06   1st Qu.:  -7.093   AROMATIC:14
## China        : 31   Median : 23.70   Median :  71.276   AUS     :57
## Bangladesh   : 27   Mean   : 21.67   Mean   :  43.598   IND     :87
## Japan        : 19   3rd Qu.: 34.66   3rd Qu.: 113.921   TEJ     :96
## Taiwan       : 19   Max.   : 55.75   Max.   : 179.414   TRJ     :97
## (Other)      :244   NA's   :20       NA's   :20
##       PC1                 PC2                PC3
## Min.   :-0.0516000   Min.   :-0.0801000   Min.   :-0.0846000
## 1st Qu.:-0.0422000   1st Qu.:-0.0090000   1st Qu.:-0.0332000
## Median :-0.0326000   Median :-0.0027000   Median : 0.0045000
## Mean   :-0.0002717   Mean   :-0.0001281   Mean   :-0.0003073
## 3rd Qu.: 0.0599000   3rd Qu.: 0.0023000   3rd Qu.: 0.0266000
## Max.   : 0.0689000   Max.   : 0.1193000   Max.   : 0.0934000
##
##       PC4                                        HybID
## Min.   :-4.140e-02   @520672006494061024104086320922140:  1
## 1st Qu.:-1.670e-02   @520672006494061024104086320922210:  1
## Median :-9.400e-03   @520672006494061024104086320922250:  1
## Mean   :-2.107e-05   @520672006494061024104086320922270:  1
## 3rd Qu.:-5.000e-04   @520672006494061024104086320922310:  1
## Max.   : 2.784e-01   @520672006494061024104086320922330:  1
##                      (Other)                          :407
## Flowering.time.at.Arkansas
## Min.   : 54.50
## 1st Qu.: 79.75
## Median : 87.71
## Mean   : 87.94
## 3rd Qu.: 96.83
## Max.   :150.50
## NA's   :39
```

Let's calculate the correlation coefficient for grain length and width.

```
cor(data$Seed.length, data$Seed.width)
```

```
## [1] NA
```

The result is NA. This is due to missing data as before.

Specify the option for dealing with the missing value and try to calculate again.

```
cor(data$Seed.length, data$Seed.width, use = "pair")
```

```
## [1] -0.2837094
```

This time, the correlation is calculated successfully.

Quiz4

Now, let's solve a practice question here. Practice questions will be presented in the lecture.

If you close the page of the quiz, go to https://www.menti.com/d9p29idxtw .

Data visualization

It is very important to look at the data from different angles before actually performing statistical analysis. For example, statistics such as mean and variance mentioned above are statistics for summarization, and even with variables with similar mean and variance, the distribution of observed values may differ greatly. Therefore, looking at the data carefully is important to understand the characteristics of the data. Data visualization is also necessary when we prepare the results of the analysis for the publication of a paper. Here we will learn various data visualization techniques.

First, let's make it possible to directly call variables in the data before explaining the visualization methods.

```
attach(data)
```

With the "attach" command, for example, we can now specify Plant.height without data$~. Otherise we have to specify the variable data$Plant.height,

Let's draw a histogram first.

```
hist(Plant.height)
```

Histogram of Plant.height
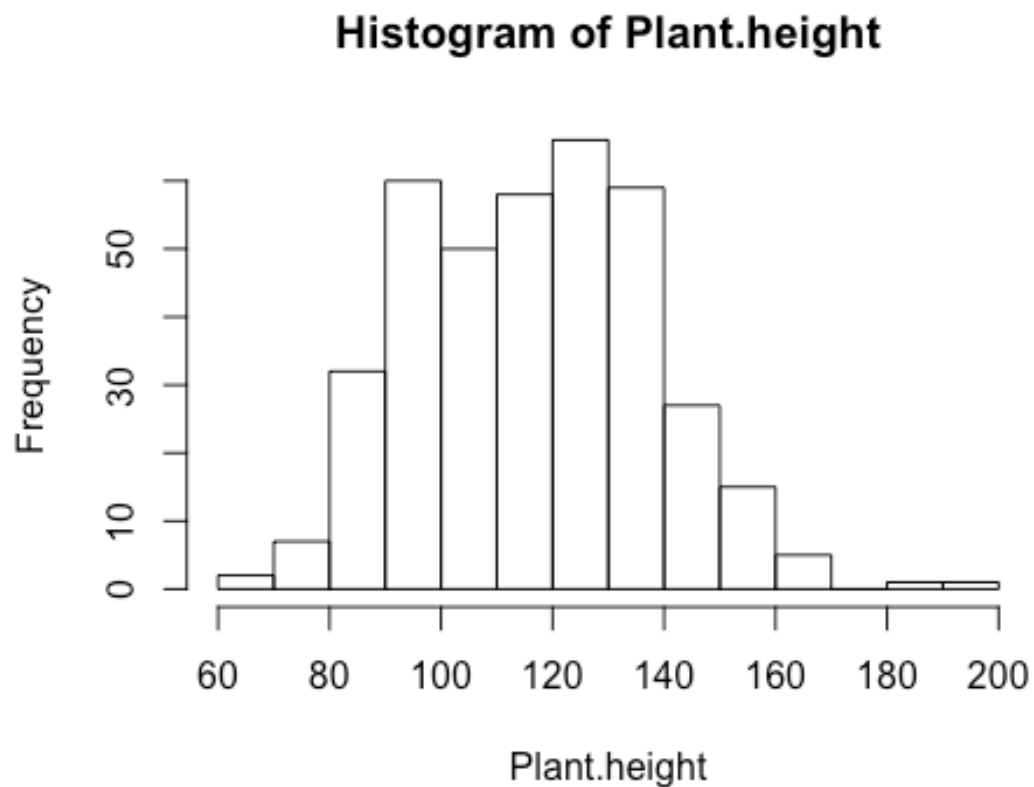
Let's draw a stem-and-leaf plot.

```
stem(Plant.height)

##
##   The decimal point is 1 digit(s) to the right of the |
##
##    6 | 8
##    7 | 0155578
##    8 | 012223344566677777888888899999
##    9 | 000111111111112223333344445555556666667777788888889999
##   10 | 000000001111112222222233333344455666666677777777778888899
##   11 | 0000111111112223333344444445555555566677778888888899999
##   12 | 00000000111111222223333333444445555666677777777778888888999999999
##   13 | 000000001111111122222222223333444444555556666667777779999
##   14 | 000001111222233344444444566778889
##   15 | 00122223344479
##   16 | 001278
##   17 |
##   18 | 8
##   19 | 4
```

This is not a graph. The result is shown with text.

Let's draw a box plot.

```
boxplot(Plant.height)
```
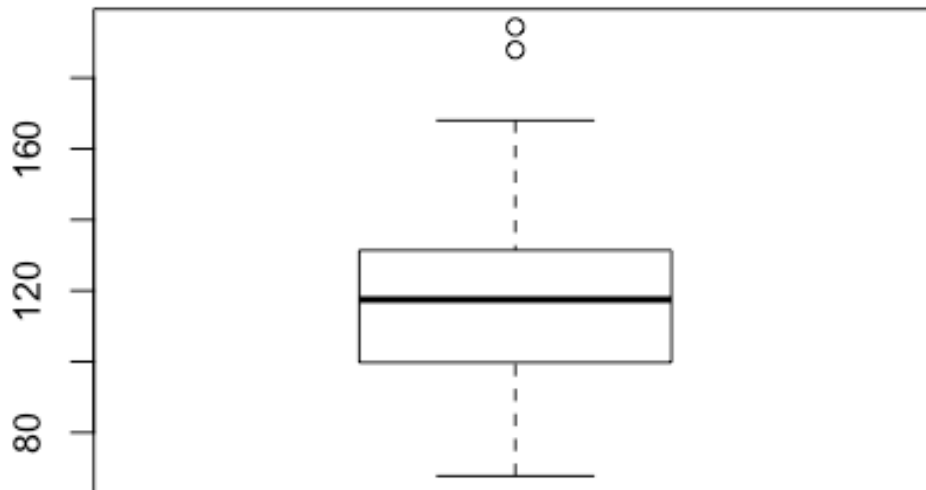
Next, we will draw a histogram of blast resistance (Blast.resistance).

```
hist(Blast.resistance)
```

Histogram of Blast.resistance

It looks like the distribution is visualized well, but there is a "pitfall".

The resistance to blast disease is expressed by the level of resistance with a score of 9 (0-9). Therefore, let's summarize how many accessions are included in each of 9 levels.

```
t <- table(Blast.resistance)
t

## Blast.resistance
##  0  1  2  3  4  5  6  7  8  9
##  3 77 23 34 36 24 39 36 52 61
```

You can see that the histogram we drew did not represent the whole class well.

You can draw a bar plot from the data summarized using the "table" function as described above.

```
plot(t)
```
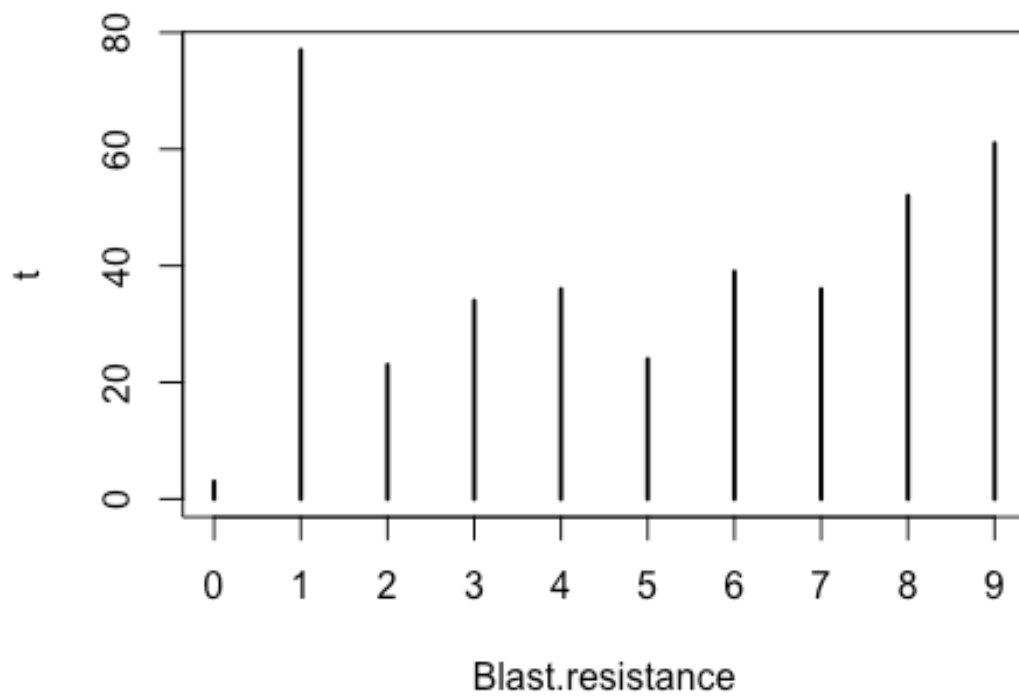
```r
plot(t, xlab = "Blast resistance scores", ylab = "Frequency")
```
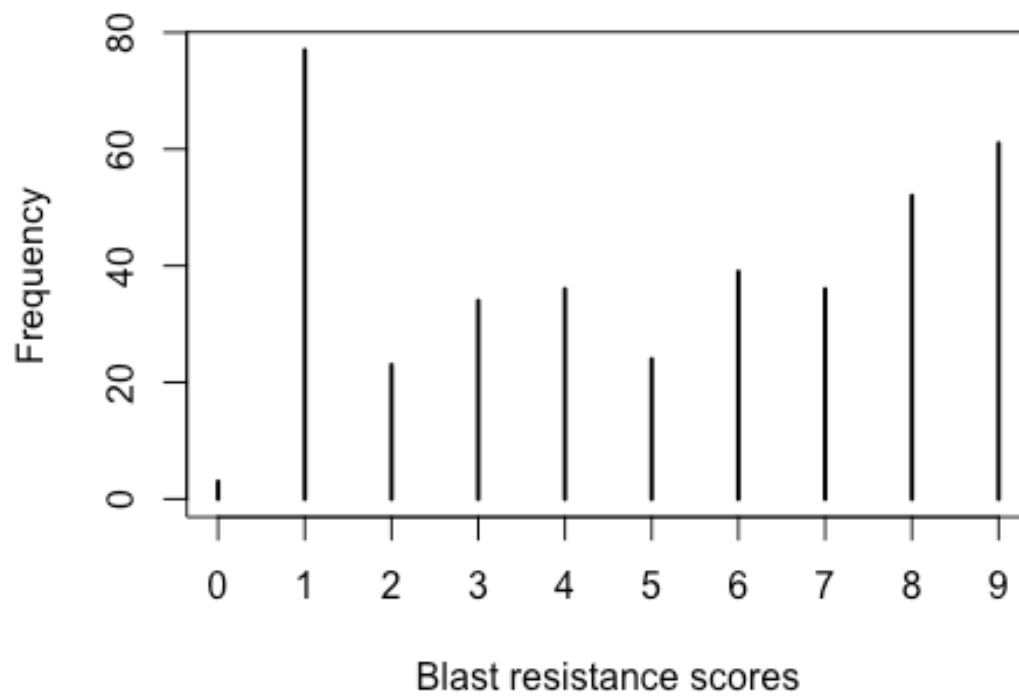
A bar chart can also be drawn using the "barplot" function. However, it looks a bit different from the bar chart drew above.

```
barplot(t)
```
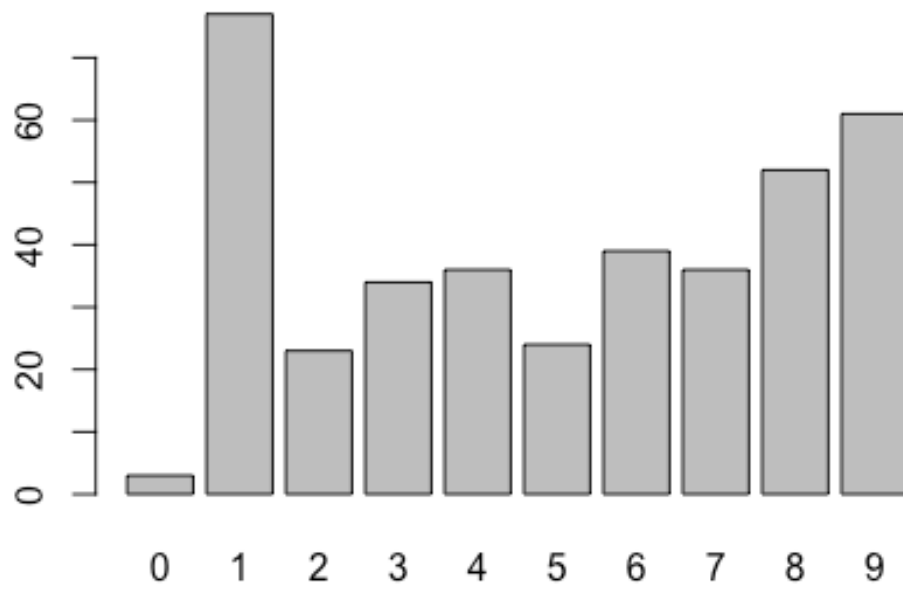
Drawing a pie chart allows you to illustrate the percentage of each score.

```
pie(t)
```

```
pie(t, main = "Blast resistance")
```

Blast resistance

From here, let's look at the relationship between two variables.

```
plot(Plant.height, Panicle.length)
```
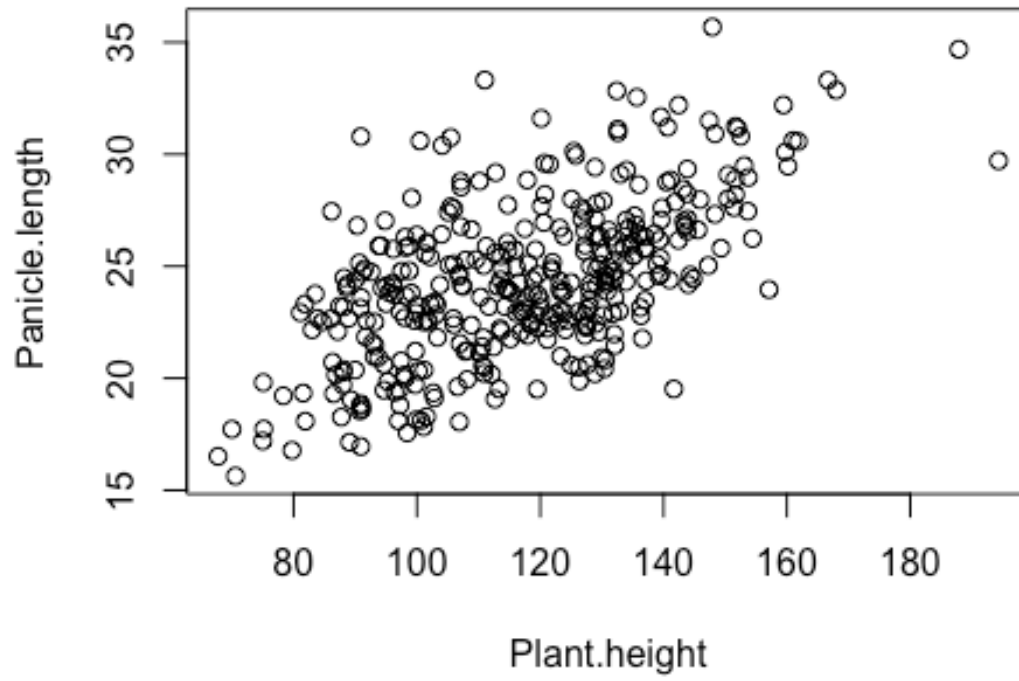
Fit a straight line to the data by regression analysis.

```
plot(Plant.height, Panicle.length)
abline(lm(Panicle.length ~ Plant.height))
```
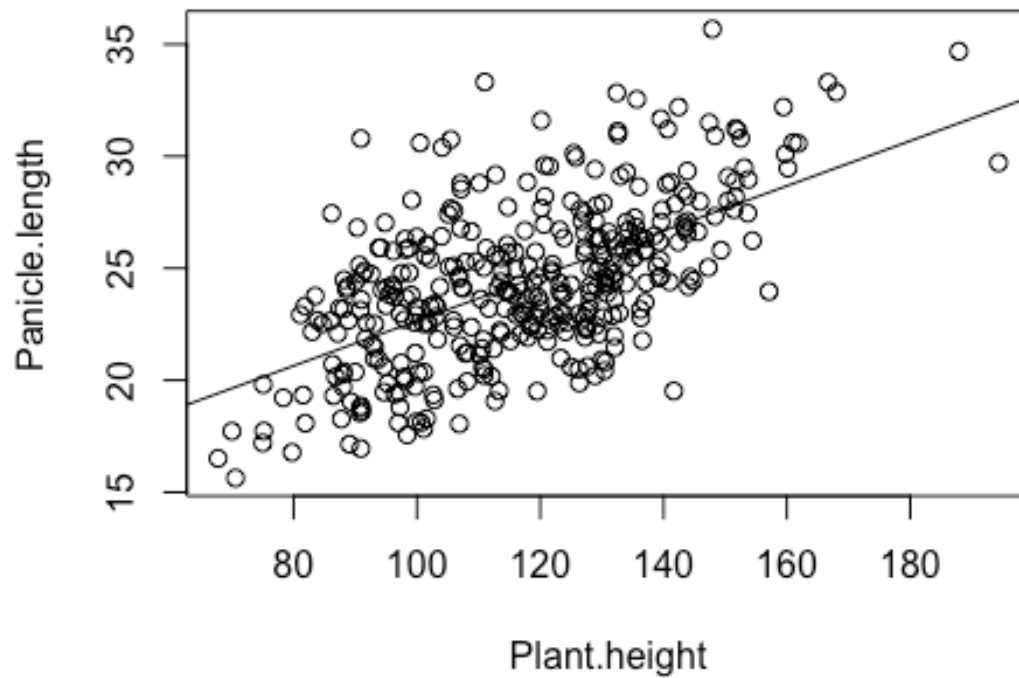
Overlap the rug (textile) plot. This is useful for visualizing distribution density.

```
plot(Plant.height, Panicle.length)
abline(lm(Panicle.length ~ Plant.height))
rug(Plant.height, side = 1)
rug(Panicle.length, side = 2)
```
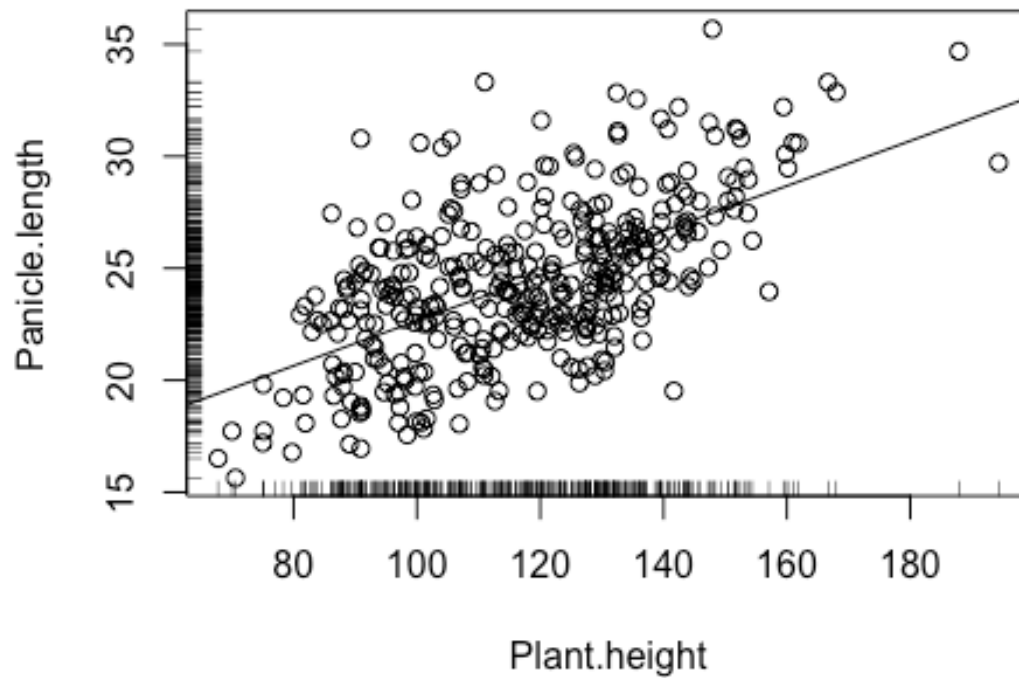
Let's illustrate the relationship between the two variables using kernel smoothing.

```r
library("KernSmooth")

## KernSmooth 2.23 loaded
## Copyright M. P. Wand 1997-2009

x <- data.frame(Plant.height, Panicle.length)
x <- na.omit(x)
d <- bkde2D(x, bandwidth = 4)
plot(x)
contour(d$x1, d$x2, d$fhat, add = T)
```
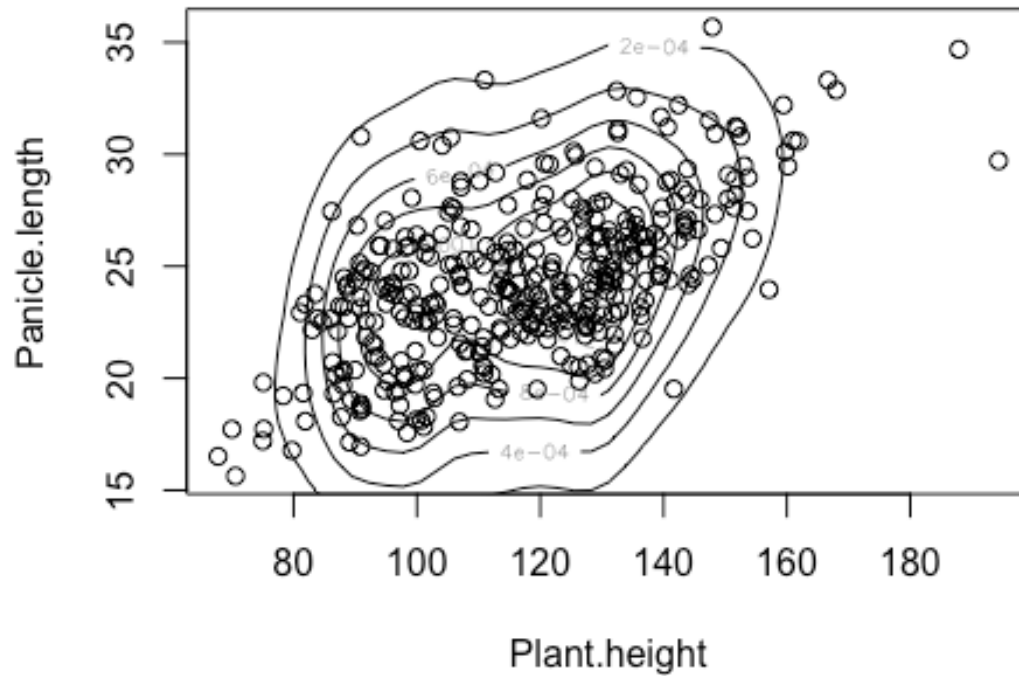
The contours represent the density of the points smoothed by the kernel.

Let's display this smoothed density in three dimensions.

```r
persp(d$x1, d$x2, d$fhat, xlab = "Plant.height", ylab = "Panicle.length", zlab = "density", theta = -30, phi = 30)
```
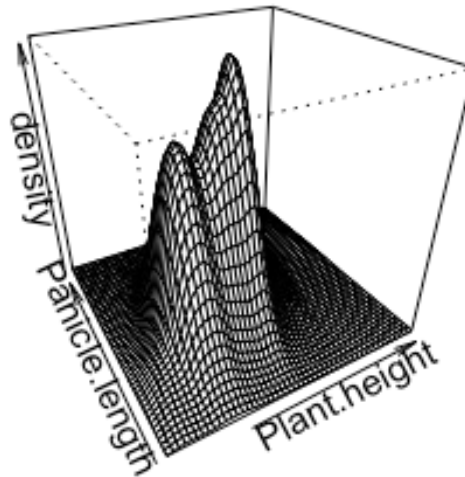
The data of Zhao et al. (2011) read from the file contain not only trait data but also data of the genetic background of genetic resources. Let's visualize both data together and investigate what kind of relationship between genetic background and trait.

SThe variable Sub.population represents differences in the genetic background of genetic resources. This is estimated using Structure analysis (Pritchard et al. 2000, Genetics 155: 945). Now, let's visualize and see what kind of relationship between genetic background and plant height and ear length.

```r
pop.id <- as.numeric(Sub.population)
plot(Plant.height, Panicle.length, col = pop.id)
levels(Sub.population)

## [1] "ADMIX"    "AROMATIC" "AUS"      "IND"      "TEJ"      "TRJ"

legend("bottomright", levels(Sub.population), col = 1:nlevels(Sub.populatio
n), pch = 1)
```
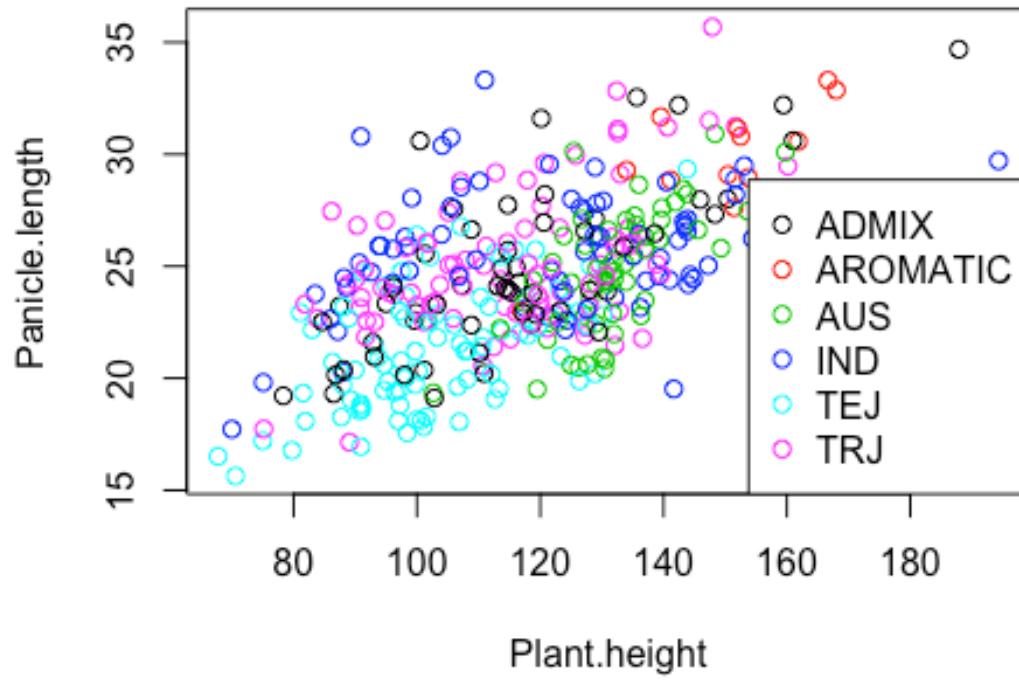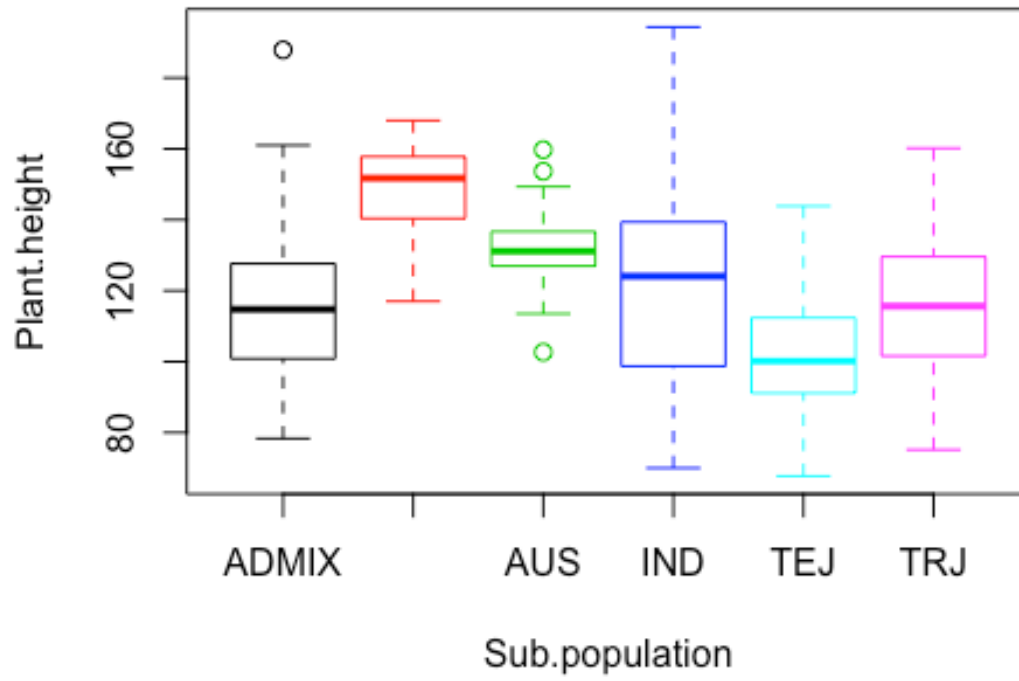
Let's show in boxplots how there are differences in values due to differences in genetic background.

```r
boxplot(Plant.height ~ Sub.population)
```
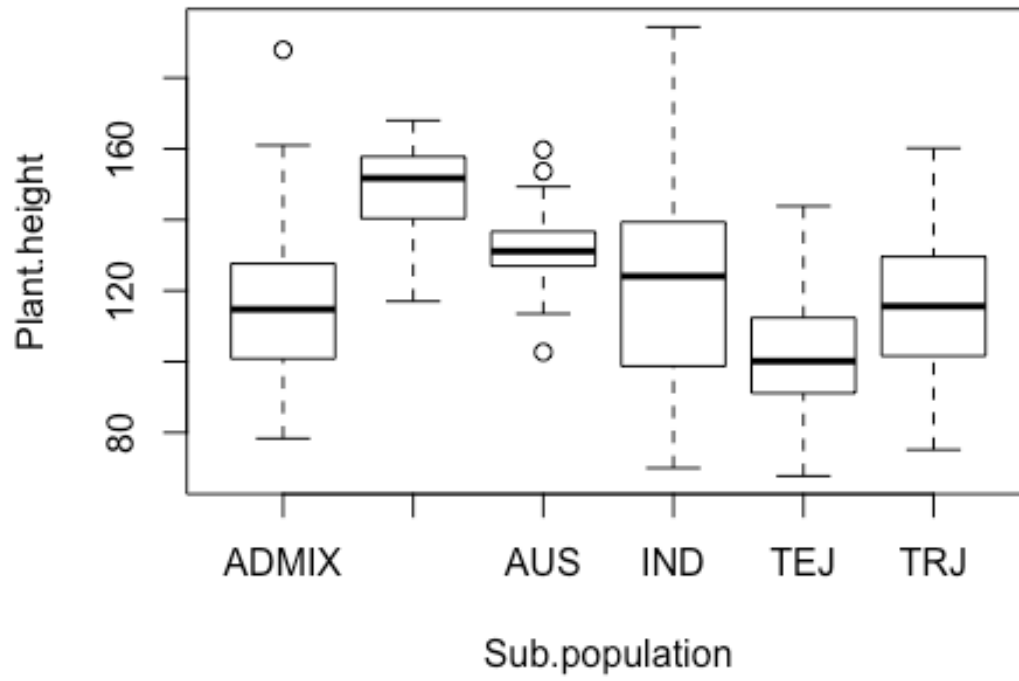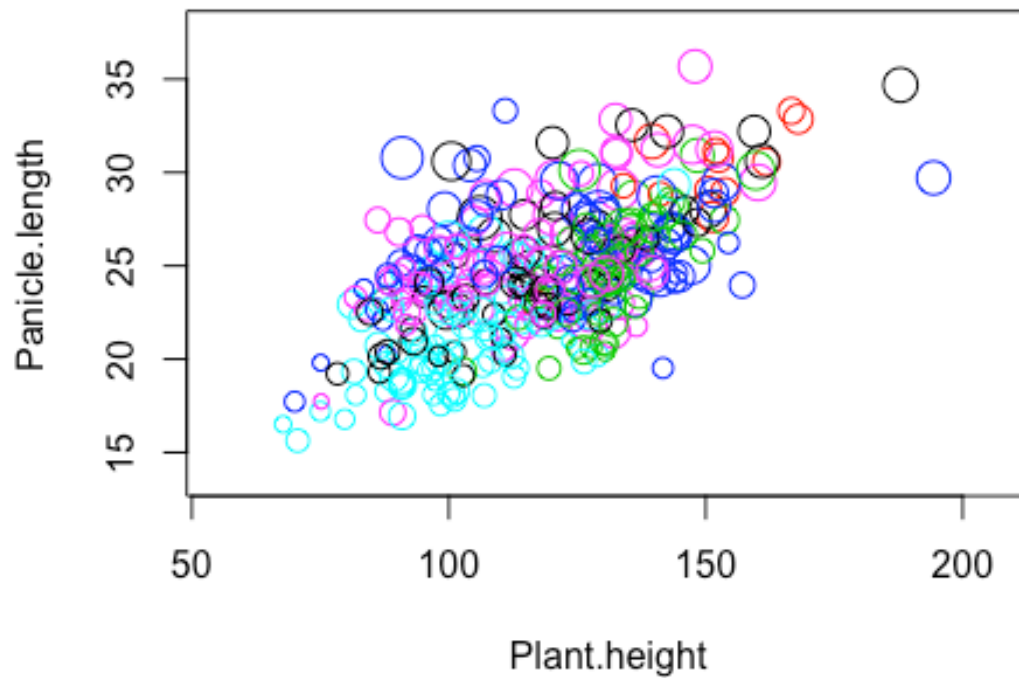
```r
boxplot(Plant.height ~ Sub.population, border = 1:nlevels(Sub.population))
```

Let's see the relationships among plant height, panicle length, and flag leaf length with a bubble plot. Here, the size of the bubble represents the flag leaf length.

```
symbols(Plant.height, Panicle.length, circles = Flag.leaf.length, inches = 0.
1, fg = pop.id)
```
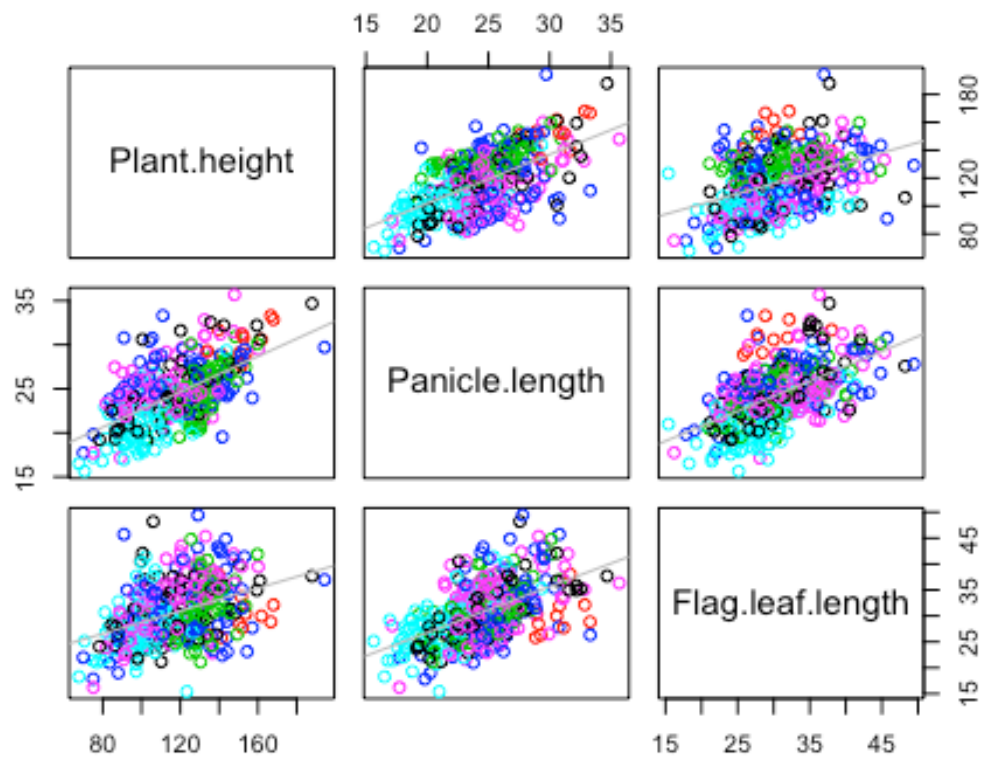
Let's draw a scatterplot of the relationship between the three variables for all possible combinations.

```
x <- data.frame(Plant.height, Panicle.length, Flag.leaf.length)
pairs(x, col = pop.id)
```
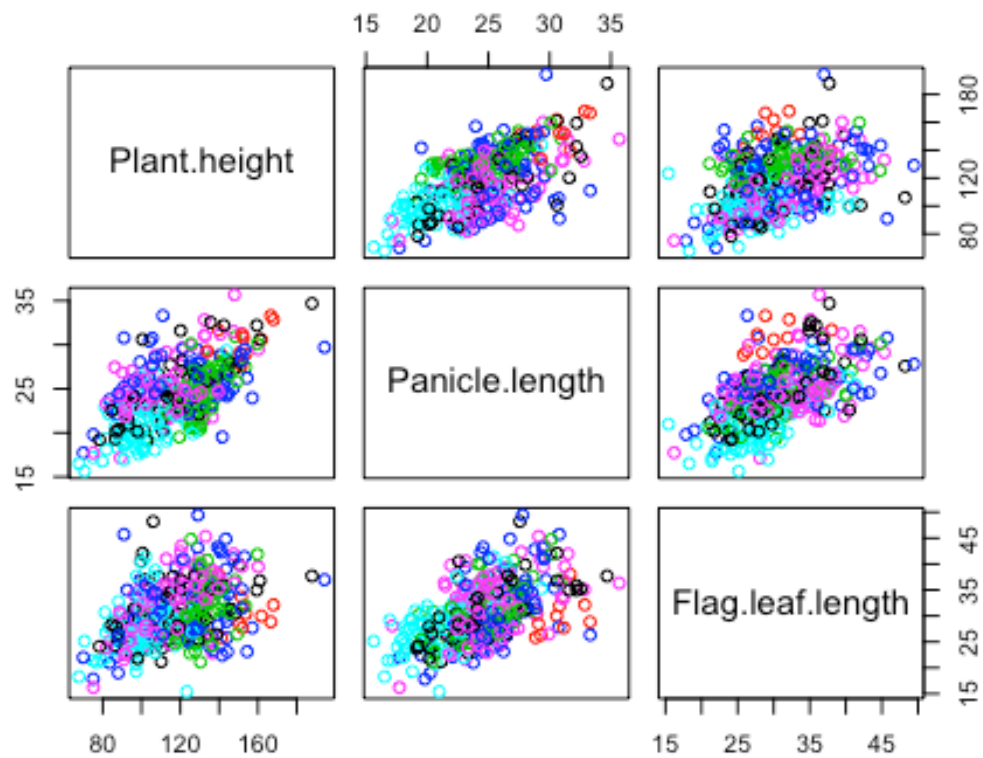
Let's make it a bit more complex scatterplot through adding regression lines.

```
pairs(x, panel = function(x, y, ...) {
    points(x, y, ...)
    abline(lm(y ~ x), col = "gray")
}, col = pop.id)
```
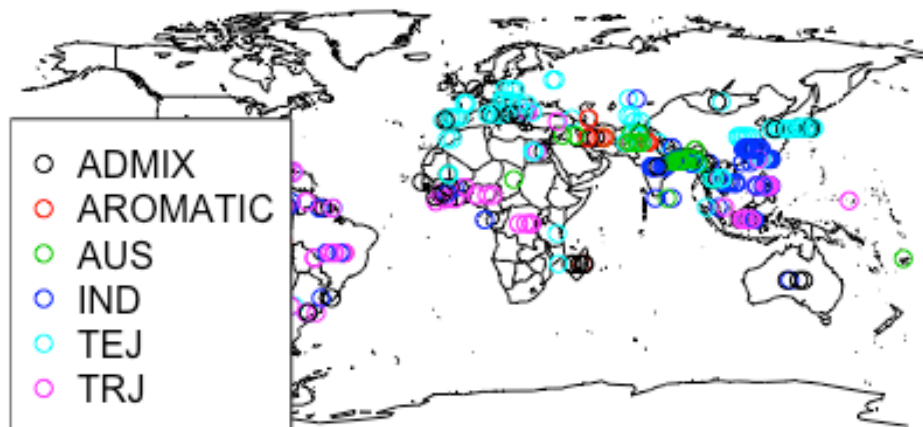
The data also have the latitude and longitude of the place where each genetic resource originates. Let's map and confirm the origin of each genetic resource on the world map.

```r
library(maps)
library(mapdata)
map('worldHires')
points(Longitude, Latitude, col = pop.id)
legend("bottomleft", levels(Sub.population), col = 1:nlevels(Sub.population),
 pch = 1)
```
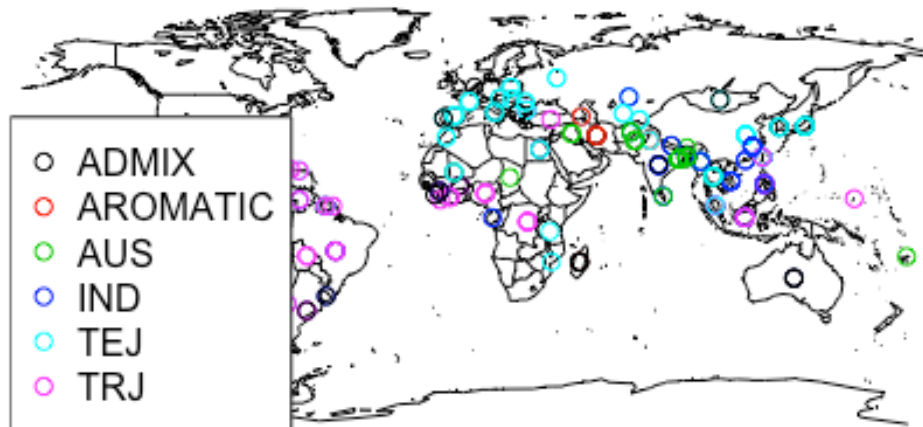
The above command only draws a much smaller number of points than the number of genetic resources. This is because genetic resources from the same area overlap each other. In order to prevent the overlapping, move overlapped points a little with the function "jitter".

```
map('worldHires')
points(jitter(Longitude, 200), Latitude, col = pop.id)
legend("bottomleft", levels(Sub.population), col = 1:nlevels(Sub.population),
 pch = 1)
```

Quiz5

Now, let's solve a practice question here. Practice questions will be presented in the lecture.

If you close the page of the quiz, go to https://www.menti.com/d9p29idxtw .

Output to file of diagram

It is useful to be able to output a graph to a PDF file in order to use the graph in a paper or presentation. Here, the method to do that will be explained briefly.

Let's output the figure we drew earlier to a file called map.pdf.

```
pdf("map.pdf")
map('worldHires')
points(jitter(Longitude, 200), Latitude, col = pop.id)
legend("bottomleft", levels(Sub.population), col = 1:nlevels(Sub.population),
 pch = 1)
dev.off()

## quartz_off_screen
##                 2
```

When the above command is executed, a file called map.pdf is output to the working directory of R.

Function "pdf" can specify the size of the output graph. When a larger size is required as is this figure, it is better to specify the size of the graph output explicitly.

```
pdf("map_large.pdf", width = 20, height = 10)
map('worldHires')
points(jitter(Longitude, 200), Latitude, col = pop.id)
legend("bottomleft", levels(Sub.population), col = 1:nlevels(Sub.population),
 pch = 1)
dev.off()

## quartz_off_screen
##                  2
```

In addition, if multiple figures are repeatedly output to the same pdf file, they will be saved in a pdf file with multiple pages. If you want to output similar and a large number of figures, it may be convenient to combine them into a single pdf file.

Draw an interactive figure

You can use the package "plotly" to draw interactive diagrams. Here, let's draw the figure drawn earlier using plotly function plot_ly.

Try to display the density of data in three dimensions.

```
require(plotly)
plot_ly(data = data.frame(d), x = d$x1, y = d$x2, z = d$fhat) %>%
add_surface()
```

Finally, we look at the relationship between the three variables.

```
df <- data.frame(Sub.population, Plant.height, Panicle.length, Flag.leaf.leng
th)
df <- na.omit(df)
plot_ly(data = df, x = ~Plant.height, y = ~Panicle.length, z = ~Flag.leaf.len
gth, color = ~Sub.population, type = "scatter3d", mode = "markers")
```

Quiz 6

Now, let's solve a practice question here. Practice questions will be presented in the lecture.

If you close the page of the quiz, go to https://www.menti.com/d9p29idxtw .

Report assignment

Use the various data visualization methods learned in the lecture to illustrate the relationship between traits and the relationship between traits and genetic background. Describe the relationships that can be read from the graphs (figures) that you drew.

Submission method:

- Create a report as a pdf file and submit it to ITC-LMS.
- When you cannot submit your report to ITC-LMS with some issues, send the report to report@iu.a.u-tokyo.ac.jp
- Make sure to write the affiliation, student number, and name at the beginning of the report.

- The deadline for submission is May 7th.