Primer of Biostatistics
The 2nd Lecture

Hiroyoshi Iwata
aiwata@mail.ecc.u-tokyo.ac.jp

<Simple regression analysis>
Changes in one variable may affect another, such as the relationship between breeding and cultivation conditions and the growth of animals and plants. One of the statistical methods to model the relationship between these variables is regression analysis. By statistically modeling the relationship between variables, it becomes possible to understand the causal relationship that exists between variables, and to predict one variable from another.
Here, first, we will discuss simple regression analysis that models the relationship between two variables as a "linear relationship". In this case, the mechanism of single regression analysis will be explained using the analysis of rice data (Zhao et al. 2011, Nature Communications 2: 467) as an example.

First, read the rice data in the same way as before. Before entering the following command, change your R working directory to the directory (folder) where the two input files (RiceDiversityPheno.csv, RiceDiversityLine.csv) are located.

```
> pheno <- read.csv("RiceDiversityPheno.csv")        # read csv file
> line <- read.csv("RiceDiversityLine.csv")
> line.pheno <- merge(line, pheno, by.x = "NSFTV.ID", by.y = "NSFTVID")
                  # merge data with NSFTV.ID in line NSFTVID in pheno
> head(line.pheno)                  #  the first six samples
 (the result is omitted)
```

Prepare analysis data by extracting only the data used for simple regression analysis from the read data. Here we analyze the relationship between plant height (Plant.height) and flowering timing (Flowering.time.at.Arkansas). In addition, principal component scores (PC1 to PC4) representing the genetic

background to be used later are also extracted. Also, remove samples with missing values in advance.

```
> data <- data.frame(
         height = line.pheno$Plant.height,           #  Plant height
         flower = line.pheno$Flowering.time.at.Arkansas,   #  flowering time
         PC1 = line.pheno$PC1,                       #  PC1
         PC2 = line.pheno$PC2,                       #  2
         PC3 = line.pheno$PC3,                       #  3
         PC4 = line.pheno$PC4)                       #  4
> data <- na.omit(data)                      #  remove missing data
```

First, visualize the relationship between two variables.

```
> plot(data$height ~ data$flower)
                           #  make flower as x and height as y
                           #  Please be familiar with the expression with ~
```

As shown in Figure 1, the earlier the flowering time, the shorter the plant height, while the later the flowering time, the taller the plant height.

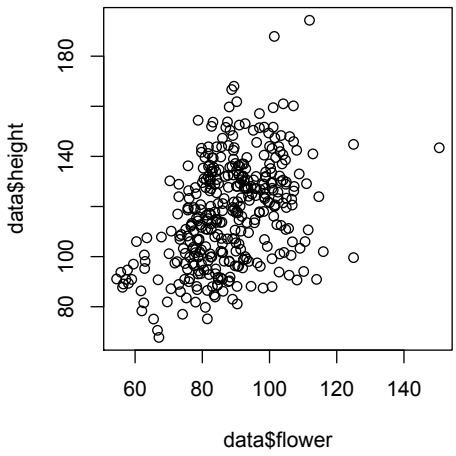


Figure 1. Relationship between flowering timing (x) and plant height (y)

Let's create a simple regression model that explains the variation in plant height by the difference in flowering timing.

```
> model <- lm(height ~ flower, data = data)
         #  make flower as x and height as y
```

The result of regression analysis (estimated model) is assigned to “model”.

Use the function “summary” to display the result of regression analysis.

```
> summary(model)          # display the result of regression analysis
```

I will explain the results displayed by executing the above command in order.

```
Call:
lm(formula = height ~ flower, data = data)
```

This is a repeat of the command you entered earlier. If you get this output right after you type it, it does not seem to be useful information. However, if you make multiple regression models and compare them as described later, it may be useful because you can reconfirm the model employed in the analysis. Here, assuming that the plant height is y_i and the flowering timing is x_i, the regression analysis is performed with the model

$$y_i = \mu + \beta x_i + \varepsilon_i.$$

As mentioned earlier, x_i is called independent variable or explanatory variable, and y_i is called dependent variable or response variable. μ and β are called parameters of the regression model, and ε_i is called error. Also, μ is called population intercept and β is called population regression coefficient.

In addition, since it is not possible to directly know the true values of the parameters μ and β of the regression model, estimation is performed based on samples. The estimates of the parameters μ and β, which are estimated from the sample, are called sample intercept and sample regression coefficient, respectively. The values of μ and β estimated from the samples are denoted by m and b, respectively. Since m and b are values estimated from the samples, they are random variables that vary depending on the samples selected by chance. Therefore, it follows a probability distribution. Details will be described later.

```
Residuals:
   Min     1Q  Median     3Q    Max
-43.846 -13.718   0.295  13.409  61.594
```

This output gives an overview of the distribution of residuals. You can use this information to check the regression model. For example, the model assumes that the expected value (average) of the error is 0. You can check whether the median is close to it. You can also check whether the distribution is symmetrical around 0, i.e., whether the maximum and minimum or the first and third quantiles have almost the same value. In this example, the maximum value is slightly larger than the minimum value, but otherwise no major issues are found.

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 58.05464    6.92496   8.383 1.08e-15 ***
flower       0.67287    0.07797   8.630  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The estimates of parameters μ and β, i.e., m and b, and their standard errors, t values and p values are shown. Asterisks at the endo of each line represent significance levels. One star represents 5%, two stars 1%, and three stars 0.1%.

```
Residual standard error: 19 on 371 degrees of freedom
Multiple R-squared: 0.1672,         Adjusted R-squared: 0.1649
F-statistic: 74.48 on 1 and 371 DF,  p-value: < 2.2e-16
```

The first line shows the standard deviation of the residuals. This is the value represented by s, where s^2 is the estimated value of the error variance σ^2. The second line is the determination coefficient R^2. This index and the adjusted R^2 represent how well the regression explain the variation of y.
The third line is the result of the F test that represents the significance of the regression model. It is a test under the hypothesis (null hypothesis) that all regression coefficients are 0, and if this p value is very small, the null hypothesis is rejected and the alternative hypothesis (regression coefficient is

not 0) is taken to be adopted.

Let's look at the results of regression analysis graphically. First, draw a scatter plot and draw a regression line.

```
> plot(data$height ~ data$flower)
> abline(model, col = "red")
```

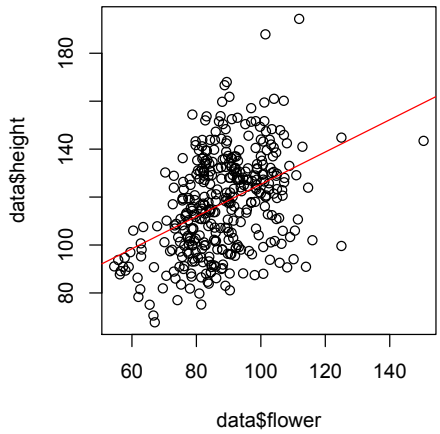


Figure 2. Scatter plot with the regression line

Next, calculate and plot the value of y when the data is fitted to the regression model.

```
> height.fit <- fitted(model)       #  calculation of fitted y values
> points(data$flower, height.fit, pch = 3, col = "green")
                                    #  fitted values were shown in green
```

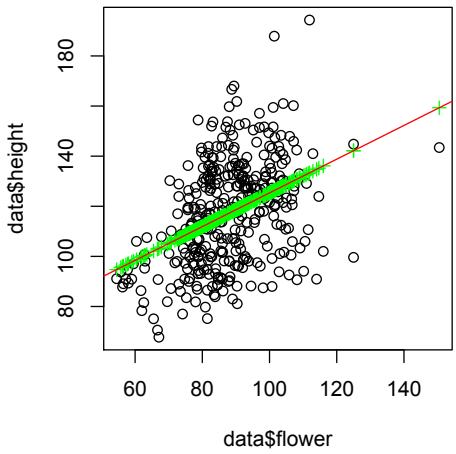


Figure 3. The values of *y* calculated by fitting the model all lie on a straight line

An observed value *y* is expressed as the sum of the variation explained by the regression model and the error which is not explained by the regression. Let's visualize the error in the figure and check the relationship.

```
> segments(data$flower, height.fit,
            data$flower, height.fit + resid(model), col = "gray")
          # segments is a function for draw a line segment between
          # (x1, y1), (x2, y2)
```

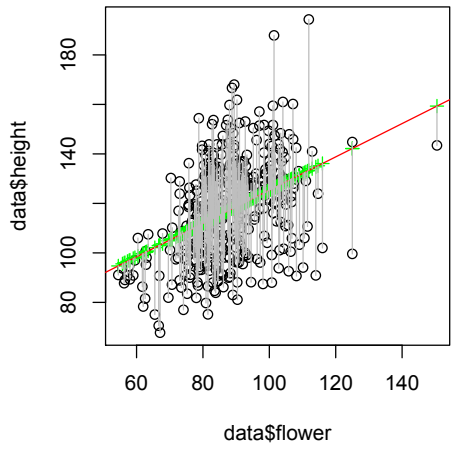


Figure 4. The value of *y* is expressed as the sum of

the values of y calculated by fitting the model (green points)
and the residuals of the model (gray line segments)

Let's use a regression model to predict y for x (60, 80, ..., 140), which are not actually observed.

```
> height.pred <- predict(model, data.frame(flower = seq(60, 140, 20)))
> points(seq(60, 140, 20), height.pred, pch = 2, col = "blue")
```

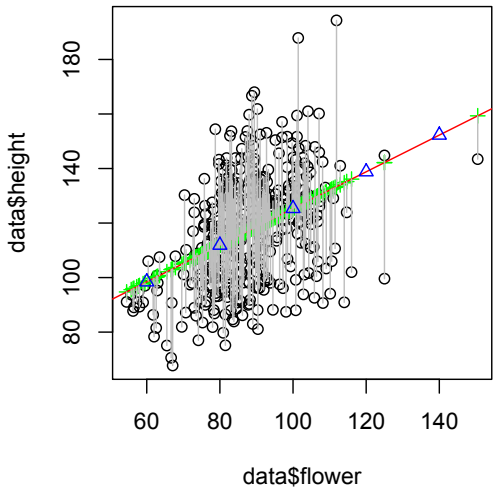


Fig. 5. All the predicted values will locate on the line

<Method for calculating the parameters of a regression model>
Here we will explain how to calculate a regression model. Also, let's calculate the regression coefficients while actually using the R command.

As mentioned earlier, the simple regression model is

$$y_i = \mu + \beta x_i + \varepsilon_i .$$

This equation implies that the observed value y_i consists of the variation $\mu + \beta x_i$ explained by the regression model and the error variation ε_i which is not explained by the regression model. As you move μ and β in the above equation, the error changes accordingly. So how can we find the "best" parameters?

There are various criteria for what is considered “optimal”, but here we consider minimizing the error across the data. Since errors can take both positive and negative values, errors cancels each other in their simple sum. So, we consider minimizing the sum of squared error (SSE). That is, consider μ and ß that minimize the following equation:

$$SSE = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - (\mu + \beta x_i))^2 \quad (1)$$

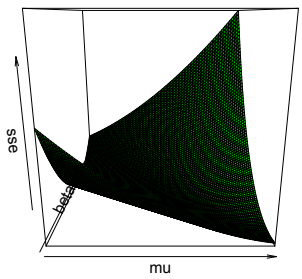


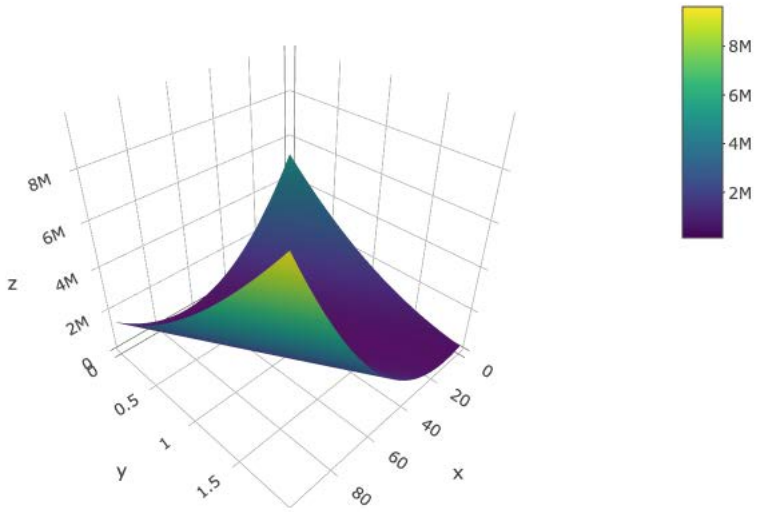
Figure 6. Relationship between regression parameter values and residual sum of squares

Figure 6 shows the change in SSE for various values of μ and β. The commands to draw Figure 6 is a little complicated, but they are as follows:

```
> x <- data$flower
> y <- data$height
> mu <- seq(0, 100, 1)
> beta <- seq(0, 2, 0.02)
> sse <- matrix(NA, length(mu), length(beta))
> for(i in 1:length(mu)) {
        for(j in 1:length(beta)) {
                sse[i, j] <- sum((y - mu[i] - beta[j] * x)^2)
        }
 }
> persp(mu, beta, sse, col = "green")
```

Draw the graph using “plotly” package

```
> # draw with plotly
> require(plotly)
> plot_ly(x = mu, y = beta, z = sse) %>% add_surface()
```



It should be noted that at the point where SSE becomes the minimum in Figure 3, SSE should not change (the slope of the tangent is zero) even when μ or β changes slightly. Therefore, the coordinates of the minimum point can be determined by partially differentiating the equation (1) with μ and β, and setting the value to zero. That is,

$$\frac{\partial SSE}{\partial \mu} = 0, \frac{\partial SSE}{\partial \beta} = 0$$

We should obtain the values of μ and β to satisfy these. The method of calculating the parameters of a regression model through minimizing the sum of squares of errors in this way is called the least squares method.

Note that μ minimizing SSE is

$$\frac{\partial SSE}{\partial \mu} = -2\sum_{i=1}^{n}(y_i - \mu - \beta x_i) = 0$$
$$\Leftrightarrow \sum_{i=1}^{n} y_i - n\mu - \beta\sum_{i=1}^{n} x_i = 0$$
$$\Leftrightarrow \mu = \frac{\sum_{i=1}^{n} y_i}{n} - \beta\frac{\sum_{i=1}^{n} x_i}{n} = \bar{y} - \beta\bar{x}$$

Also, β minimizing SSE is

$$\frac{\partial SSE}{\partial \beta} = -2\sum_{i=1}^{n} x_i(y_i - \mu - \beta x_i) = 0$$
$$\Leftrightarrow \sum_{i=1}^{n} x_i y_i - \mu\sum_{i=1}^{n} x_i - \beta\sum_{i=1}^{n} {x_i}^2 = 0$$
$$\Leftrightarrow \sum_{i=1}^{n} x_i y_i - n(\bar{y} - \beta\bar{x})\bar{x} - \beta\sum_{i=1}^{n} {x_i}^2 = 0$$
$$\Leftrightarrow \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y} - \beta\left(\sum_{i=1}^{n} {x_i}^2 - n\bar{x}^2\right) = 0$$
$$\Leftrightarrow \beta = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} {x_i}^2 - n\bar{x}^2} = \frac{SSXY}{SSX}$$

Here, SSXY and SSX are sum of products of deviation in x and y and deviation the sum of squares of deviation in x, respectively.

$$SSXY = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$
$$= \sum_{i=1}^{n} x_i y_i - \bar{x}\sum_{i=1}^{n} y_i - \bar{y}\sum_{i=1}^{n} x_i + n\bar{x}\bar{y}$$
$$= \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y} - n\bar{y}\bar{x} + n\bar{x}\bar{y}$$
$$= \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}$$

$$SSX = \sum_{i=1}^{n}(x_i - \bar{x})^2$$
$$= \sum_{i=1}^{n} x_i^2 - 2\bar{x}\sum_{i=1}^{n} x_i + n\bar{x}^2$$
$$= \sum_{i=1}^{n} x_i^2 - 2n\bar{x}^2 + n\bar{x}^2$$
$$= \sum_{i=1}^{n} x_i^2 - n\bar{x}^2$$

The values of μ and β minimizing SSE are the estimates of the parameters, and let the estimates be represented by m and b. That is,

$$b = \frac{SSXY}{SSX}$$
$$m = \bar{y} - b\bar{x}$$

Now let's calculate the regression coefficients based on the above equation. First, calculate the sum of products f deviation and the sum of squares of deviation.

```
> n <- length(x)                    #  substitute sample number for n
> ssxy <- sum(x * y) - n * mean(x) * mean(y)
> ssx <- sum(x^2) - n * mean(x)^2
```

First we calculate the slope b.

```
> b <- ssxy / ssx
> b
[1] 0.6728746
```

Then calculate the intercept μ.

```
> m <- mean(y) - b * mean(x)
> m
[1] 58.05464
```

Let's draw a regression line based on the calculated estimates.

```
> plot(y ~ x)
> abline(m, b)    # draw the line of intercept m and slope b
```

Let's make sure that the same regression line was obtained as the function lm which we used earlier.

Note that once the regression parameters μ and β are estimated, it is possible to calculate $\hat{y}_i$, which is the value of y corresponding to a given x_i. That is,

$$\hat{y}_i = m + bx_i.$$

This makes it possible to calculate the value of y when the model is fitted to the observed x, or to predict y if only the value of x is known. Here, let's calculate the value of y when the model is fitted to the observed x, and draw scatter points on the figure drawn earlier.

```
> y.hat <- m + b * x
> lim <- range(c(y, y.hat))
> plot(y, y.hat, xlab = "Observed", ylab = "Fitted", xlim = lim, ylim = lim)
> abline(0, 1)
```

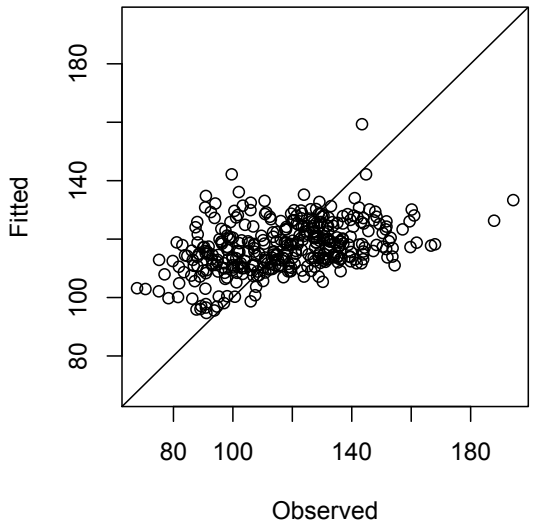


Figure 7. Relationship between observed and fitted values

Let's calculate the correlation coefficient between the two to find the degree of agreement between the observed and fitted values.

```
> cor(y, y.hat)
[1] 0.408888
```

In fact, the square of this correlation coefficient is the proportion of the variation of y explained by the regression (coefficient of determination, R^2 value). Let's compare these two statistics.

```
> cor(y, y.hat)^2
[1] 0.1671894
> summary(model)
 (omitted)
Multiple R-squared: 0.1672,         Adjusted R-squared: 0.1649
 (omitted)
```

<Significance test of a regression model>

When the linear relationship between variables is strong, a regression line fit well to the observations, and the relationship between both variables can be well modeled by a regression line. However, when a linear relationship between variables is not clear, modeling with a regression line does not work well. Here, as a method to objectively confirm the goodness of fit of the estimated regression model, we will explain a test using analysis of variance.

First, let's go back to the simple regression again.

```
model <- lm(height ~ flower, data = data)
```

The significance of the obtained regression model can be tested using the function anova.

```
> anova(model)
Analysis of Variance Table

Response: height
          Df Sum Sq Mean Sq F value    Pr(>F)
flower     1  26881 26881.5  74.479 < 2.2e-16 ***
Residuals 371 133903   360.9
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

As a result of the analysis of variance, the term of flowering time is highly significant ($p < 0.001$), and the goodness of fit of the regression model that the flowering timing influences plant height is confirmed.

Analysis of variance for regression models involves the following calculations: First of all, “Sum of squares explained by regression” can be calculated as follows.

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

$$= \sum_{i=1}^{n} \left(\mu + bx_i - (\mu + b\bar{x})\right)^2$$
$$= b^2 \sum_{i=1}^{n} (x_i - \bar{x})^2$$
$$= b^2 \cdot SSX = b \cdot SSXY$$

Also, the sum of squares of deviation from the mean of the observed values y is expressed as the sum of the sum of squares SSR explained by the regression and the residual sum of squares SSE. That is,

$$SSY = \sum_{i=1}^{n} (y_i - \bar{y})^2$$
$$= \sum_{i=1}^{n} (y_i - \hat{y}_i + \hat{y}_i - \bar{y})^2$$
$$= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$
$$= SSE + SSR$$

$$\because 2 \sum_{i=1}^{n} (y_i - \hat{y}_i)(\hat{y}_i - \bar{y})$$
$$= 2 \sum_{i=1}^{n} (y_i - m - bx_i)\left(m + bx_i - (m + b\bar{x})\right)$$
$$= 2b \sum_{i=1}^{n} (y_i - (\bar{y} - b\bar{x}) - bx_i)(x_i - \bar{x})$$
$$= 2b \sum_{i=1}^{n} \left(y_i - \bar{y} - b(x_i - \bar{x})\right)(x_i - \bar{x})$$
$$= 2b(SSXY - b \cdot SSX) = 0$$

Let's actually calculate it using the above equation. First, calculate SSR and SSE.

```
> ssr <- b * ssxy
> ssr
[1] 26881.49
> ssy <- sum(y^2) - n * mean(y)^2
> sse <- ssy - ssr
> sse
[1] 133903.2
```

Next, calculate the mean squares, which is of the sum of squares divided by the degrees of freedom.

```
> msr <- ssr / 1
> msr
[1] 26881.49
> mse <- sse / (n - 2)
> mse
[1] 360.9251
```

Finally, the mean square of the regression is divided by the mean square of the error to calculate the F value. Furthermore, calculate the p value corresponding to the calculated F value.

```
> f.value <- msr / mse
> f.value
[1] 74.47943
>
> 1 - pf(f.value, 1, n - 2)
[1] 2.220446e-16
```

The results obtained are in agreement with the results calculated earlier using the function anova.

The results of regression analysis are included in the results of regression analysis displayed using the function “summary”.

```
> summary(model)
 (omitted)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 58.05464    6.92496   8.383 1.08e-15 ***
flower       0.67287    0.07797   8.630  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19 on 371 degrees of freedom
Multiple R-squared: 0.1672,         Adjusted R-squared: 0.1649
F-statistic: 74.48 on 1 and 371 DF,  p-value: < 2.2e-16
```

"Residual standard error" is the square root of the mean square of the residual.

```
> sqrt(mse)
[1] 18.99803
```

"Multiple R-squared" (R^2) is a value called the coefficient of determination, which is the ratio of SSR to SSY.

```
> ssr / ssy
[1] 0.1671894
```

"Adjusted R-squared" (R^2_{adj}) is a value called the adjusted coefficient of determination, which can be calculated as follows.

```
> (ssy / (n - 1) - mse) / (ssy / (n - 1))
[1] 0.1649446
```

Also, "F-statistic" matches the F value and its p value which are expressed as the effect of flowering time in the analysis of variance. In addition, the t value calculated for the regression coefficient of the flowering time term is squared to obtain the F value ($8.6302 = 74.477$).

R^2 and R^2_{adj} can also be expressed using SSR, SSY, and SSE as follows.

$$R^2 = \frac{SSR}{SSY} = 1 - \frac{SSE}{SSY}$$

$$R^2_{adj} = 1 - \frac{n-1}{n-p} \cdot \frac{SSE}{SSY}$$

Here, p is the number of parameters included in the model, and $p = 2$ for a simple regression model. It can be seen that R^2_{adj} has a larger amount of adjustment (the residual sum of squares is underestimated) as the number of parameters included in the model increases.

<Distribution that estimated value of regression coefficient follows>

As mentioned earlier, the estimates b and m of the regression coefficients μ and β are values estimated from samples and are random variables that depend on the samples chosen by chance. Thus, estimates b and m have probabilistic distributions. Here we consider the distributions that the estimates follow.

First, think about b. We can express b as

$$b = \sum_{i=1}^{n} \frac{(x_i - \bar{x})(y_i - \bar{y})}{SSX}$$
$$= \sum_{i=1}^{n} \frac{(x_i - \bar{x})y_i}{SSX} - \bar{y} \sum_{i=1}^{n} \frac{(x_i - \bar{x})}{SSX}$$
$$= \frac{1}{SSX} \sum_{i=1}^{n} y_i(x_i - \bar{x})$$

Thus, the mean of the estimate b is

$$\mathbb{E}(b) = \frac{1}{SSX} \mathbb{E}\left(\sum_{i=1}^{n} y_i(x_i - \bar{x}) \right)$$
$$= \frac{1}{SSX} \mathbb{E}\left(\sum_{i=1}^{n} (\mu + \beta x_i + \varepsilon_i)(x_i - \bar{x}) \right)$$
$$= \frac{1}{SSX} \mathbb{E}\left(\sum_{i=1}^{n} (\mu^* + \beta(x_i - \bar{x}) + \varepsilon_i)(x_i - \bar{x}) \right)$$
$$= \frac{1}{SSX} \left[\mu^* \sum_{i=1}^{n} (x_i - \bar{x}) + \beta \sum_{i=1}^{n} (x_i - \bar{x})^2 + \mathbb{E}\left(\sum_{i=1}^{n} \varepsilon_i(x_i - \bar{x}) \right) \right]$$
$$= \frac{1}{SSX} [0 + \beta SSX + 0] = \beta$$

That is, the mean of the estimated value b matches the true value β. Here, μ^* is a constant term when y_i is regressed not to x_i but to $x_i - \bar{x}$,

$$y_i = \mu^* + \beta(x_i - \bar{x})$$

The variance of the estimated value b is

$$\mathbb{V}(b) = \frac{1}{SSX^2}\mathbb{V}\left(\sum_{i=1}^{n} y_i(x_i - \bar{x})\right)$$
$$= \frac{1}{SSX^2}\sum_{i=1}^{n}(x_i - \bar{x})^2\,\mathbb{V}(y_i)$$
$$= \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{SSX^2}\sigma^2 = \frac{\sigma^2}{SSX}$$

Here, σ^2 is the residual variance $\sigma^2 = \mathbb{V}(y_i) = \mathbb{V}(e_i)$.

The estimated value b is a linear combination of y_i

$$b = \sum_{i=1}^{n} a_i y_i\,, \qquad a_i = \frac{x_i - \bar{x}}{SSX}$$

Since y_i follows a normal distribution, its linear combination b also follows a normal distribution. That is,

$$b \sim N(\beta, \frac{\sigma^2}{SSX})$$

On the other hand, the estimated value m can be expressed as $m = \bar{y} - \beta\bar{x}$.

Thus, the average is

$$\mathbb{E}(m) = \mathbb{E}(\bar{y} - \beta\bar{x}) = \mu + \beta\bar{x} - \beta\bar{x} = \mu$$

The mean of the estimated value m also matches the true value μ.

Then the variance is

$$\mathbb{V}(m) = \mathbb{V}(\bar{y}) + \mathbb{V}(b\bar{x}) - 2\mathbb{C}\mathrm{ov}(\bar{y}, b\bar{x}) = \frac{\sigma^2}{n} + (\bar{x})^2\frac{\sigma^2}{SSX} - 2\cdot 0 = \sigma^2\left[\frac{1}{n} + \frac{(\bar{x})^2}{SSX}\right]$$

Since y_i follows normal distribution, m represented as $m = \bar{y} - \beta\bar{x}$ also follows normal distribution. That is,

$$m \sim N(\mu, \sigma^2\left[\frac{1}{n} + \frac{(\bar{x})^2}{SSX}\right])$$

Although the true value of the error variance σ^2 is unknown, it can be replaced by the residual variance s^2. That is,

$$s^2 = \frac{SSE}{n-2}$$

This value is the mean square of the residuals calculated during the analysis of variance.

At this time, statistics on b

$$t = \frac{b-\beta_0}{s/\sqrt{SSX}}$$

follows the t distribution with n – 2 degrees of freedom

$$H_0: \beta = \beta_0$$

under the null hypothesis.

At this time, an interval in which β (that is, β_0) is included with a probability of $1-\alpha$, that is, a $(1-\alpha)$ 100% confidence interval is calculated as follows.

$$\left[b - t_{n-2,1-\frac{\alpha}{2}}\frac{s}{\sqrt{SSX}}, \quad b + t_{n-2,1-\frac{\alpha}{2}}\frac{s}{\sqrt{SSX}}\right]$$

Here, $t_{n-2,1-\alpha/2}$ is the rejection limits at both sides of the 5% (α = 0.05) or the 1% (α = 0.01) level in the degree of freedom.

Also for m, statistics

$$t = \frac{m-\mu_0}{s\sqrt{\frac{1}{n}+\frac{(\bar{x})^2}{SSX}}}$$

follows the t distribution with n – 2 degrees of freedom

$$H_0: m = \mu_0$$

under the null hypothesis.

At this time, an interval in which μ (that is, μ_0) is included with a probability of 1-α, that is, a (1-α) 100% confidence interval is calculated as follows.

$$\left[m - t_{n-2,1-\frac{\alpha}{2}} s\sqrt{\frac{1}{n} + \frac{(\bar{x})^2}{SSX}},\quad m + t_{n-2,1-\frac{\alpha}{2}} s\sqrt{\frac{1}{n} + \frac{(\bar{x})^2}{SSX}}\right]$$

Now let's calculate the test and confidence intervals for b and m that we have found so far.

First, test for null hypothesis H_0: ß = 0 for b.

```
> t.value <- (b - 0) / sqrt(mse/ssx)
> t.value
[1] 8.630147
> 2 * (1 - pt(t.value, n - 2))
[1] 2.220446e-16
```

The results of this test were already displayed as regression analysis results.

```
> summary(model)
（省略）
oefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 58.05464    6.92496   8.383 1.08e-15 ***
flower       0.67287    0.07797   8.630  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
（省略）
```

The hypothesis test performed above can be performed for any β_0. For example, let's test for null hypothesis $H_0: \beta = 0.5$.

```
> t.value <- (b - 0.5) / sqrt(mse/ssx)
> t.value
[1] 2.217253
> 2 * (1 - pt(t.value, n - 2))
[1] 0.02721132
```

The result is significant at the 5% level. This means that 0.5 is not included in the 95% confidence interval mentioned above.

Now let us test and calculate confidence intervals for m. First, let's test the null hypothesis $H_0: m = 0$.

```
> t.value <- (m  - 0) / sqrt(mse * (1/n + mean(x)^2 / ssx))
> t.value
[1] 8.383389
> 2 * (1 - pt(t.value, n - 2))
[1] 1.110223e-15
```

This result was also already calculated.

```
> summary(model)
（省略）
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 58.05464    6.92496   8.383 1.08e-15 ***
flower       0.67287    0.07797   8.630  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
（省略）
```

(It may be due to the rounding error that the p value does not match completely)

Finally, let's test for the null hypothesis $H_0: m = 50$.

```
> t.value <- (m  - 50) / sqrt(mse * (1/n + mean(x)^2 / ssx))
> t.value
[1] 1.163132
> 2 * (1 - pt(t.value, n - 2))
[1] 0.2455237
```

The result was not significant even at the 5% level. This means that 50 is "included" in the 95% confidence interval mentioned above.

<Confidence intervals for regression coefficients and fitted values>

Function “predict” has various functions. First, let's use the function with the estimated regression model. Then, the values of y when the model fitted to observed data are calculated. The values are exactly the same as calculated by the function “fitted”.

```
> pred <- predict(model)
> head(pred)
      1       2       3       4       5       6
108.5763 118.2769 121.6413 116.9312 117.9966 128.7065
> head(fitted(model))
      1       2       3       4       5       6
108.5763 118.2769 121.6413 116.9312 117.9966 128.7065
```

By setting the options “interval” and “level”, you can calculate the confidence interval of *y* at the specified significance level when fitting the model.

```
> pred <- predict(model, interval = "confidence", level = 0.95)
> head(pred)
      fit     lwr     upr
1 108.5763 105.8171 111.3355
2 118.2769 116.3275 120.2264
3 121.6413 119.4596 123.8230
4 116.9312 114.9958 118.8665
5 117.9966 116.0540 119.9391
6 128.7065 125.4506 131.9623
```

Let's vidualize the confidence interval of y using the function “predict”.

```
> pred <- data.frame(flower = 50:160)
> pc <- predict(model, int = "c", newdata = pred)
> plot(data$height ~ data$flower)
> matlines(pred$flower, pc, lty = c(1, 2, 2), col = "red")
```

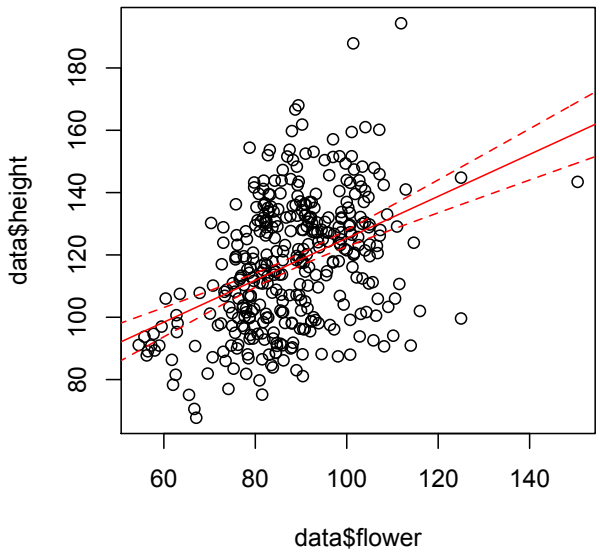


Figure 8. Confidence interval of y when fitting the model
The interval becomes narrow at a point close to the mean of x, while becomes wider at a point far from the mean

The confidence interval of y can be calculated as follows. First, consider estimating y given x^*, that is, $y_m = \mathbb{E}(y|x = x^*) = \mu + \beta x^*$. Assuming that the regression coefficient estimated from the sample is b, the estimated value of y_m is $\hat{y}_m = m + bx^*$. Here, since m and b are random variables, $\hat{y}_m$ is also a random variable. $\hat{y}_m$ is represented as

$$\hat{y}_m = m + bx^* = \bar{y} + b(x^* - \bar{x})$$

And its variance is calculated as:

$$V(\hat{y}_m) = V(\bar{y}) + (x^* - \bar{x})^2 V(b)$$

$$= \frac{\sigma^2}{n} + \frac{(x^* - \bar{x})^2 \sigma^2}{SSX}$$

Replacing the residual variance s^2 calculated from the sample for the residual variance σ^2, as before, statistics

$$t = \frac{\hat{y}_m - y_m}{s\sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{SSX}}}$$

Follows the t distribution with $n-2$ degrees of freedom. Therefore, the interval in which the true value $y_m = \mu + \beta x^*$ is included with the probability of 1-α, that is, the (1-α) 100% confidence interval is calculated as follows.

$$\left[\hat{y}_m - t_{n-2,1-\frac{\alpha}{2}} s\sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{SSX}}, \hat{y}_m + t_{n-2,1-\frac{\alpha}{2}} s\sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{SSX}}\right]$$

Now, let's draw the confidence intervals of the estimate of y given x^* according to the above equation.

```
> x <- 50:160
> tv <- qt(0.975, n - 2)
> y.hat <- mu + beta * x
> mean.x <- mean(data$flower)
> y.hat.upper <- y.hat + tv * sqrt(mse) * sqrt(1/n + (x - mean.x)^2 / ssx)
> y.hat.lower <- y.hat - tv * sqrt(mse) * sqrt(1/n + (x - mean.x)^2 / ssx)
> plot(data$height ~ data$flower)
> matlines(x, cbind(y.hat, y.hat.upper, y.hat.lower),
                            lty = c(1, 2, 2), col = "red")
```

Let's confirm that the same figure as Figure 8 is drawn.

<Polynomial regression model and multiple regression model>
So far, we have applied to the data a regression model that represents the relationship between the two variables with a straight line. Let's extend the regression model a bit.

First, let's perform regression by a method called polynomial regression. In polynomial regression,

$$y_i = \mu + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_p x_i^p + \varepsilon_i$$

In this way, regression is performed using the second or higher order terms of x. First, let's perform regression using the first and second terms of x.

```
> model.quad <- lm(height ~ flower + I(flower^2), data = data)
> summary(model.quad)
 (omitted)
Multiple R-squared: 0.1915,         Adjusted R-squared: 0.1871
 (omitted)
```

It can be seen that the proportion of variation of y (coefficient of determination R^2) explained by the polynomial regression model is larger than that of the simple regression model.

Although this will be mentioned later, you should not judge that the polynomial regression model is excellent only with this value. This is because a polynomial regression model has more parameters than a simple regression model, and you have more flexibility when fitting the model to data. It is easy to improve the fit of the model to the data by increasing the flexibility. In extreme cases, the model can be completely fitted to the data with as many parameters as the size of data (In that case, the coefficient pf determination R^2 completely matches 1). Therefore, careful selection of some statistical criteria is required when selecting the best model. This will be discussed later.

Now let's draw the result of polynomial regression with confidence intervals.

```
> pred <- data.frame(flower = 50:160)
                  #  set the range for the calculation (giving x)
> pc <- predict(model.quad, int = "c", newdata = pred)
                  #  calculate fitted values for the x
> plot(data$height ~ data$flower)           #  draw a scatterplot
> matlines(pred$flower, pc, lty = c(1, 2, 2), col = "red")
         #  draw the polynomial regression curve and their confidence interval
```

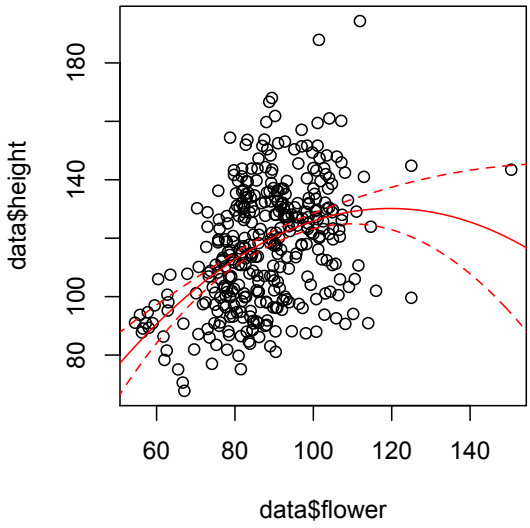


Figure 9. Second-order polynomial regression results.
When the timing of flowering is over 120 days after sowing,
it can be seen that the reliability is low.

Let's visually compare the explanatory power of the polynomial regression model and the simple regression model.

```
> lim <- range(c(data$height, fitted(model), fitted(model.quad)))
> plot(data$height, fitted(model),
+                  xlab = "Observed", ylab = "Expected",
+                  xlim = lim, ylim = lim)
> points(data$height, fitted(model.quad), col = "red")
> abline(0, 1)
```

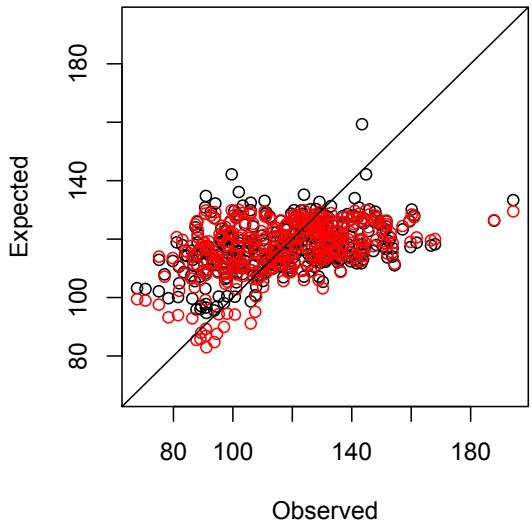


Figure 10. For a simple regression model (black) and
a second-order polynomial model (red)
Relationship between fitted value and observed value

Let's test if the improvement in the explanatory power of the second-order polynomial model is statistically significant. The significance is tested with F test whether the difference between the residual sum of squares of the two models is sufficiently large compared to the residual sum of squares of the model containing the other (here, Model 2 contains Model 1).

```
> anova(model, model.quad)
Analysis of Variance Table

Model 1: height ~ flower
Model 2: height ~ flower + I(flower^2)
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    371 133903
2    370 129999  1    3903.8 11.111 0.0009449 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The results show that the difference in residual variance between the two models is highly significant ($p < 0.001$). In other words, Model 2 has significantly more explanatory power than Model 1.

Now let's fit a third-order polynomial regression model and test if it is significantly more descriptive than a second-order model.

```
> model.cube <- lm(height ~ flower + I(flower^2) + I(flower^3), data = data)
> summary(model.cube)
 (omitted)
Multiple R-squared: 0.1931,         Adjusted R-squared: 0.1866
 (omitted)
> anova(model.quad, model.cube)
Analysis of Variance Table

Model 1: height ~ flower + I(flower^2)
Model 2: height ~ flower + I(flower^2) + I(flower^3)
  Res.Df    RSS Df Sum of Sq     F Pr(>F)
1    370 129999
2    369 129729  1    270.17 0.7685 0.3813
```

The 3rd-order model has a slightly better explanatory power than the 2nd-order model. However, the difference is not statistically significant. In other words, it turns out that extending a second-order model to a third-order model is not a good idea.

Finally, let's apply the multiple linear regression model.
In multiple linear regression,

$$y_i = \mu + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi} + \varepsilon_i$$

In this way, regression is performed using multiple explanatory variables ($x_{1i}, x_{2i}, \ldots, x_{pi}$). In the first lecture, I confirmed in the graph that the height varies depending on the difference in genetic background. Here, we will create a multiple regression model that explains plant height using genetic backgrounds (PC1 to PC4) expressed as the scores of four principal components.

```
> model.wgb <- lm(height ~ PC1 + PC2 + PC3 + PC4, data = data)
> summary(model.wgb)
 (omitted)
Multiple R-squared: 0.3388,         Adjusted R-squared: 0.3316
 (omitted)
> anova(model.wgb)
 (omitted)
Response: height
          Df Sum Sq Mean Sq F value    Pr(>F)
PC1        1  28881 28881.3  99.971 < 2.2e-16 ***
PC2        1   5924  5924.2  20.506 8.040e-06 ***
PC3        1   6723  6723.2  23.272 2.063e-06 ***
PC4        1  12942 12942.3  44.799 8.163e-11 ***
Residuals 368 106314   288.9
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

You can see that the coefficient of determination of the regression model is higher than that of the polynomial regression model. The results of analysis of variance show that all principal components are significant and need to be included in the regression.

Finally, let's combine the polynomial regression model with the multiple regression model.

```
> model.all <- lm(height ~ flower + I(flower^2) + PC1 + PC2 + PC3 + PC4,
                  data = data)
> summary(model.all)
 (omitted)
Multiple R-squared: 0.4045,         Adjusted R-squared: 0.3947
 (omitted)
> anova(model.all, model.wgb)
Analysis of Variance Table

Model 1: height ~ flower + I(flower^2) + PC1 + PC2 + PC3 + PC4
Model 2: height ~ PC1 + PC2 + PC3 + PC4
  Res.Df    RSS Df Sum of Sq     F   Pr(>F)
1    366  95753
2    368 106314 -2    -10561 20.184 4.84e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The effect of the genetic background on plant height is very large, but it can also be seen that the model's explanatory power improves if the effect of flowering timing is also added.

Lastly, let's compare the first regression model and the last created multiple regression model by plotting the scatter of the observed value and the fitted value.

```
> lim <- range(data$height, fitted(model), fitted(model.all))
> plot(data$height, fitted(model), xlab = "Observed",
                  ylab = "Fitted", xlim = lim, ylim = lim)
> points(data$height, fitted(model.all), col = "red")
> abline(0,1)
```

As a result, we can see that the explanatory power of the model is significantly improved by considering the genetic background and the second order terms. However, on the other hand, it can also be seen that the two varieties and lines whose flowering timing is late (after 180 days) can not be sufficiently explained even by the finally obtained model. There may be room to improve the model, such as adding new factors as independent variables.

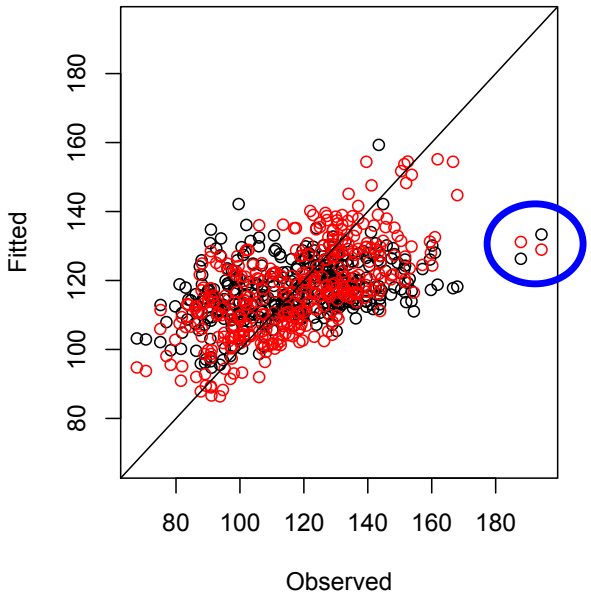


Figure 11. Comparison of simple regression model (black) and multiple regression model (red)

Horizontal axis is observed value and vertical axis is model fitted value

In the sample in the blue circle, the badness of fit has not been resolved

<Experimental design and analysis of variance>
When trying to draw conclusions based on experimental results, it is always the presence of errors in the observed values. Errors are inevitable no matter how precise the experiment is, especially in field experiments, errors are caused by small environmental variations in the field. Therefore, experimental design is a method devised to obtain objective conclusions without being affected by errors.

First of all, what is most important in planning experiments is the Fisher's three principles:
(1) Replication: In order to be able to perform statistical tests on experimental results, we repeat the same process. For example, evaluate one variety multiple times. The experimental unit equivalent to one replication is called a plot.
(2) Randomization: An operation that makes the effect of errors random is called randomization. For example, in the field test example, varieties are randomly assigned to plots in the field using dice or random numbers.
(3) Local control: Local control means dividing the field into blocks and managing the environmental conditions in each block to be as homogeneous as possible. In the example of the field test, the grouped area of the field is divided into small units called blocks to make the cultivation environment in the block as homogeneous as possible. It is easier to homogenize each block rather than homogenizing the cultivation environment of the whole field.

The experimental method of dividing the field into several blocks and making the cultivation environment as homogeneous as possible in the blocks is called the randomized block design. In the randomized block design, the field is divided into blocks, and varieties are randomly assigned within each block. The number of blocks is equal to the number of replications.

Next, I will explain the method of statistical test in the randomized block design through a simple simulation. First, let's set the "seed" of the random number before starting the simulation. A random seed is a source value for generating pseudorandom numbers.

```
> set.seed(123)
```

Let's start the simulation. Here, consider a field where 16 plots are arranged in 4×4. And think about the situation that there is a slope of the soil fertility in the field.

```
> field.cond <- matrix(rep(c(4,2,-2,-4), each = 4), nrow = 4)
> field.cond
     [,1] [,2] [,3] [,4]
[1,]    4    2   -2   -4
[2,]    4    2   -2   -4
[3,]    4    2   -2   -4
[4,]    4    2   -2   -4
```

However, it is assumed that there is an effect of +4 where the soil fertility is high and -4 where it is low.

Here, we arrange blocks according to Fisher's three principles. The blocks are arranged to reflect the difference in the soil fertility well.

```
> block <- c("I", "II", "III", "IV")
> blomat <- matrix(rep(block, each = 4), nrow = 4)
> blomat
     [,1] [,2] [,3]  [,4]
[1,] "I"  "II" "III" "IV"
[2,] "I"  "II" "III" "IV"
[3,] "I"  "II" "III" "IV"
[4,] "I"  "II" "III" "IV"
```

Next, randomly arrange varieties in each block according to Fisher's three principles. Let's prepare for that first.

```
> variety <- c("A", "B", "C", "D")          #  4 varieteis
> sample(variety)
[1] "B" "C" "A" "D"        # function “sample” can sort entries randomly
> sample(variety)
        # each time you execute the function, the order is randomized
```

Let's allocate varieties randomly to each block.

```
> varmat <- matrix(c(sample(variety), sample(variety),
                           sample(variety), sample(variety)), nrow = 4)
> varmat
     [,1] [,2] [,3] [,4]
[1,] "C"  "C"  "A"  "D"
[2,] "B"  "B"  "D"  "C"
[3,] "D"  "A"  "C"  "B"
[4,] "A"  "D"  "B"  "A"
```

Consider the differences in genetic values of the four varieties. Let the genetic values of the A to D varieties be +4, +2, -2, -4, respectively.

```
> g.value <- matrix(NA, 4, 4)
> g.value[varmat == "A"] <- 4
> g.value[varmat == "B"] <- 2
> g.value[varmat == "C"] <- -2
> g.value[varmat == "D"] <- -4
> g.value
     [,1] [,2] [,3] [,4]
[1,]   -2   -2    4   -4
[2,]    2    2   -4   -2
[3,]   -4    4   -2    2
[4,]    4   -4    2    4
```

Environmental variations are generated as random numbers from a normal distribution with an average of 0 and a standard deviation of 2.5.

```
> e.value <- matrix(rnorm(16, sd = 2.5), 4, 4)
> e.value
          [,1]     [,2]      [,3]      [,4]
[1,]  1.0019286  1.244626 -2.6695593 -1.5625982
[2,]  0.2767068 -4.916543 -0.5449373 -4.2167333
[3,] -1.3896028  1.753390 -2.5650111  2.0944676
[4,]  4.4672828 -1.181979 -1.8222281  0.3834328
```

Although the above command generates random numbers, I think you will get the same value as the textbook. This is because the random numbers

generated are pseudo random numbers and are generated according to certain rules. Note that if you change the value of the random seed, the same value as above will not be generated. Also, different random numbers are generated each time you run.

Finally, the overall average, the gradient of soil fertility, the genetic values of varieties, and the variation due to the local environment are added together to generate a simulated observed value of the trait.

```
> grand.mean <- 50
> simyield <- grand.mean + field.cond + g.value + e.value
> simyield
         [,1]     [,2]     [,3]     [,4]
[1,] 53.00193 51.24463 49.33044 40.43740
[2,] 56.27671 49.08346 43.45506 39.78327
[3,] 48.61040 57.75339 43.43499 50.09447
[4,] 62.46728 46.81802 48.17777 50.38343
```

Let's visualize the simulated data.

```
> op <- par(mfrow = c(2, 2))
> image(t(field.cond))
> for(i in 1:4) text((i-1) / 3, 0:3 / 3, blomat[,i])
> image(t(g.value))
> for(i in 1:4) text((i-1) / 3, 0:3 / 3, varmat[,i])
> image(t(e.value))
> image(t(simyield))
> for(i in 1:4) text((i-1) / 3, 0:3 / 3, paste(varmat[,i], blomat[,i]))
> par(op)
```

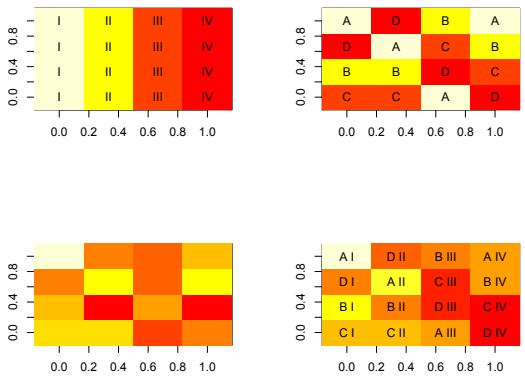


Figure 12. Gradient of soil fertility (upper left), the genetic values of varieties (upper right), Environmental variation (lower left) and observed values of the trait (lower right)

Before performing analysis of variance, reshape data in the form of matrices into vectors and rebundle them.

```
> as.vector(simyield)
 [1] 53.00193 56.27671 48.61040 62.46728 51.24463 49.08346 57.75339 46.81802
49.33044 43.45506 43.43499
[12] 48.17777 40.43740 39.78327 50.09447 50.38343
> as.vector(varmat)
 [1] "C" "B" "D" "A" "C" "B" "A" "D" "A" "D" "C" "B" "D" "C" "B" "A"
> as.vector(blomat)
 [1] "I"   "I"   "I"   "I"   "II"  "II"  "II"  "II"  "III" "III" "III" "III"
"IV"  "IV"  "IV"  "IV"
> simdata <- data.frame(variety = as.vector(varmat),
                  block = as.vector(blomat), yield = as.vector(simyield))
> simdata
 (omitted)
```

Let's plot the created data using the function interaction.plot.

```
> interaction.plot(simdata$block, simdata$variety, simdata$yield)
```

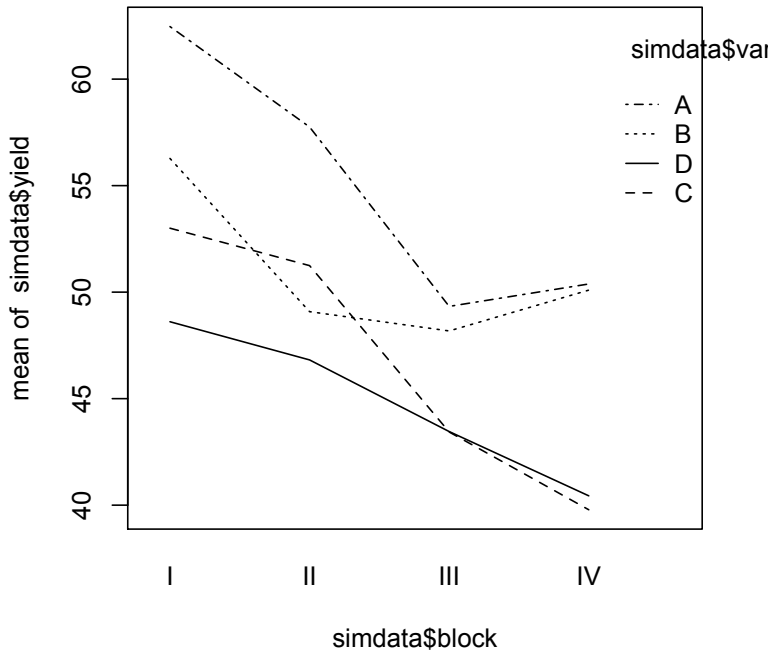


Fig. 13. Simulated yield data in varieties and blocks.

It can be seen that the difference between blocks is as large as the difference between varieties

Let's perform an analysis of variance using the prepared data.

```
> res <- aov(yield ~ block + variety, data = simdata)
> summary(res)
          Df Sum Sq Mean Sq F value  Pr(>F)
block      3 257.77   85.92   13.45 0.00113 **
variety    3 243.02   81.01   12.68 0.00139 **
Residuals  9  57.48    6.39
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

It can be seen that both the block and variety effects are highly significant. Note that the former is not the subject of verification, and is incorporated into the model in order to estimate the variety effect correctly.

The analysis of variance described above can also be performed using the function “lm” for estimating regression models.

```
> res <- lm(yield ~ block + variety, data = simdata)
> anova(res)
Analysis of Variance Table

Response: yield
          Df  Sum Sq Mean Sq F value   Pr(>F)
block      3 257.769  85.923  13.453 0.001126 **
variety    3 243.017  81.006  12.683 0.001391 **
Residuals  9  57.484   6.387
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

In the function “lm”, analysis of variance is performed within the framework of regression analysis using dummy variables. In addition, you can check the setting of the dummy variables by using the function model.matrix.

```
> model.matrix(res)
 (omitted)
> summary(res)
 (omitted)
```

<How to perform the analysis of variance>

Now, let x_{ij} be the observed value of the trait in the jth block of the ith breed. Then, x_{ij} can be written as follows.

$$x_{ij} = \bar{x}.. + (\bar{x}_i. - \bar{x}..) + (\bar{x}._j - \bar{x}..) + (x_{ij} - \bar{x}_i. - \bar{x}._j + \bar{x}..)$$

Here, $\bar{x}_i.$, $\bar{x}._j$, $\bar{x}..$represents the mean for the ith variety, the mean for the jth plot, and the total mean, respectively. That is,

$$\bar{x}_i. = \sum_{i=1}^{r} x_{ij} / r$$

$$\bar{x}._j = \sum_{j=1}^{m} x_{ij} / m$$

$$\bar{x}.. = \sum_{i=1}^{r}\sum_{j=1}^{m} x_{ij} / (mr) = \sum_{i=1}^{m} \bar{x}_i. / m = \sum_{j=1}^{r} \bar{x}._j / r$$

Here, m is the number of varieties and r is the number of blocks.

The sum of squares of differences from the average of the observed values can be split into:

$$\sum_{i=1}^{m}\sum_{j=1}^{r}(x_{ij} - \bar{x}..)^2$$
$$= \sum_{i=1}^{m}\sum_{j=1}^{r}(\bar{x}_i. - \bar{x}..)^2 + \sum_{i=1}^{m}\sum_{j=1}^{r}(\bar{x}._j - \bar{x}..)^2 + \sum_{i=1}^{m}\sum_{j=1}^{r}(x_{ij} - \bar{x}_i. - \bar{x}._j + \bar{x}..)^2$$
$$= r\sum_{i=1}^{m}(\bar{x}_i. - \bar{x}..)^2 + m\sum_{j=1}^{r}(\bar{x}._j - \bar{x}..)^2 + \sum_{i=1}^{m}\sum_{j=1}^{r}(x_{ij} - \bar{x}_i. - \bar{x}._j + \bar{x}..)^2$$

The first item is the sum of squares due to varieties, the second item is the sum of squares due to blocks, and the third item is the sum of squares due to errors.

The sum of squares divided by the degrees of freedom is called the mean square. The mean square corresponds to the unbiased variance due to each variation. Analysis of variance calculates the ratio of the mean square of the variety divided by the mean square of the error, and tests the significance of the effect of the varieties using the fact that the ratio follows the F distribution with the degrees of freedom $m-1$ and $(m-1)(r-1)$, under the null hypothesis.

The following shows the R code to do an analysis of variance without using the function aov.

```
> simdata <- simdata[order(simdata$block, simdata$variety),]
> simdata
（結果を省略）
> xij <- matrix(simdata$yield, nrow = 4)
> xij
       [,1]    [,2]    [,3]    [,4]
[1,] 62.46728 57.75339 49.33044 50.38343
[2,] 56.27671 49.08346 48.17777 50.09447
[3,] 53.00193 51.24463 43.43499 39.78327
[4,] 48.61040 46.81802 43.45506 40.43740
> x.. <- mean(xij)
> xi. <- apply(xij, 1, mean)
> x.j <- apply(xij, 2, mean)
>
> m <- nrow(xij)
> r <- ncol(xij)
> ss.blo <- sum((x.j - x..)^2) * m
> ss.blo
[1] 257.769
> ss.var <- sum((xi. - x..)^2) * r
> ss.var
[1] 243.0174
> ss.err <- sum((sweep(sweep(xij, 1, xi.), 2, x.j) + x..)^2)
> ss.err
[1] 57.48384
>
> ms.blo <- ss.blo / (r - 1)
> ms.blo
[1] 85.92301
> ms.var <- ss.var / (m - 1)
> ms.var
[1] 81.00579
> ms.err <- ss.err / ((m - 1) * (r - 1))
> ms.err
[1] 6.387094
>
> f.value <- ms.var / ms.err
> f.value
[1] 12.68273
>
> qf(1 - c(0.05, 0.01, 0.001), m - 1, (m - 1) * (r - 1))
[1]  3.862548  6.991917 13.901803
>
> p.value <- 1 - pf(f.value, m - 1, (m - 1) * (r - 1))
> p.value
[1] 0.001391247
```

<Complete random block design and completely randomized design>
The local control, one of Fisher's three principles, is very important for performing highly accurate experiments in fields under high heterogeneity between plots. Here, assuming the same environmental conditions as before, let's consider performing an experiment without setting up a block.

In the previous simulation experiment, we blocked each column and placed A, B, C, D randomly in that block. Here we will assign the plots with 4 varieties x 4 replicates completely randomly across the field. An experiment in which blocks are not arranged in the expliment and arranged completely randomly is called "completely randomized design."

```
> varmat.crd <- matrix(sample(varmat), nrow = 4)
> varmat.crd
     [,1] [,2] [,3] [,4]
[1,] "D"  "D"  "A"  "B"
[2,] "B"  "B"  "D"  "C"
[3,] "D"  "A"  "C"  "A"
[4,] "C"  "A"  "B"  "C"
```

This time, you should careful that the frequency of appearance of variety varies from row to row, since varieties are randomly assigned to the entire field. $x_{1i}, x_{2i}, ..., x_{pi}$

The genetic effect is assigned according to the order of varieties in a completely random arrangement.

```
> g.value.crd <- matrix(NA, 4, 4)
> g.value.crd[varmat.crd == "A"] <- 4
> g.value.crd[varmat.crd == "B"] <- 2
> g.value.crd[varmat.crd == "C"] <- -2
> g.value.crd[varmat.crd == "D"] <- -4
> g.value.crd
     [,1] [,2] [,3] [,4]
[1,]   -4   -4    4    2
[2,]    2    2   -4   -2
[3,]   -4    4   -2    4
[4,]   -2    4    2   -2
```

As in the previous simulation experiment, the overall average, the gradient of soil fertility, the genetic effect of varieties, and the variation due to the local environment are summed up.

```
> simyield.crd <- grand.mean + g.value.crd + field.cond + e.value
> simyield.crd
        [,1]    [,2]    [,3]    [,4]
[1,] 51.00193 49.24463 49.33044 46.43740
[2,] 56.27671 49.08346 43.45506 39.78327
[3,] 48.61040 57.75339 43.43499 52.09447
[4,] 56.46728 54.81802 48.17777 44.38343
```

Now let's perform analysis of variance on the data generated in the simulation. Unlike the previous experiment, we do not set blocks. Thus, we perform regression analysis with the model that only includes the varietal effect and does not include the block effect.

```
> res <- lm(yield ~ variety, data = simdata.crd)
> anova(res)
Analysis of Variance Table

Response: yield
          Df Sum Sq Mean Sq F value Pr(>F)
variety    3 121.38  40.461  1.7072 0.2185
Residuals 12 284.41  23.701
> summary(res)
（結果は省略）
```

In the above example, the varietal effect is not significant. This is considered to be due to the fact that the spatial heterogeneity in the field causes the error to be large and the genetic difference between varieties cannot be estimated with sufficient accuracy.

The above simulation experiment was repeated 100 times (shown on the next page). As a result, in the experiment using the random complete block design, the varietal effect was detected (the significance level was set to 5%) in 94 experiments out of 100, but it was detected only 66 times in the completely random arrangement. In addition, when the significance level was set to 1%, the number of the varietal effect detected was 70 and 30, respectively (in the case of completely random arrangement, the varietal effect was missed 70 times!). From this result, it can be seen that the adoption of the random complete block design is effective when there is

among-replication heterogeneity such as the slope of soil fertility. In order to make a time-consuming and labor-intensive experiment as efficient as possible, it is important to design the experiment properly.

```
> n.rep <- 100
> p.rbd <- rep(NA, n.rep)
> p.crd <- rep(NA, n.rep)
> for(i in 1:n.rep) {
         # experiment with randomized block design
         varmat <- matrix(c(sample(variety), sample(variety),
                            sample(variety), sample(variety)), nrow = 4)
         g.value <- matrix(NA, 4, 4)
         g.value[varmat == "A"] <- 4
         g.value[varmat == "B"] <- 2
         g.value[varmat == "C"] <- -2
         g.value[varmat == "D"] <- -4
         e.value <- matrix(rnorm(16, sd = 2.5), 4, 4)
         simyield <- grand.mean + field.cond + g.value + e.value
         simdata <- data.frame(variety = as.vector(varmat),
                            block = as.vector(blomat),
                            yield = as.vector(simyield))
         res <- lm(yield ~ block + variety, data = simdata)
         p.rbd[i] <- anova(res)$Pr[2]

         # experiment with completed randomized design
         varmat.crd <- matrix(sample(varmat), nrow = 4)
         g.value.crd <- matrix(NA, 4, 4)
         g.value.crd[varmat.crd == "A"] <- 4
         g.value.crd[varmat.crd == "B"] <- 2
         g.value.crd[varmat.crd == "C"] <- -2
         g.value.crd[varmat.crd == "D"] <- -4
         simyield.crd <- grand.mean + g.value.crd + field.cond + e.value
         simdata.crd <- data.frame(variety = as.vector(varmat.crd),
                                   yield = as.vector(simyield.crd))
         res <- lm(yield ~ variety, data = simdata.crd)
         p.crd[i] <- anova(res)$Pr[1]
 }
> sum(p.rbd < 0.05) / n.rep
[1] 0.94
> sum(p.crd < 0.05) / n.rep
[1] 0.66
> sum(p.rbd < 0.01) / n.rep
[1] 0.7
> sum(p.crd < 0.01) / n.rep
[1] 0.3
```

<Report assignment>

Analyze the relationship between a trait and the genetic background for several traits using the regression analysis you learned in the lecture.

Submission procedure:

· Create a report as a pdf file and submit it as an email attachment.

· Send an e-mail to "report@iu.a.u-tokyo.ac.jp".

· At the beginning of the report, do not forget to write your affiliation, student number, and name.

· The deadline of the submission is May 24th.

K.W.

The cat was made of Greek letters!

Did you notice that?