バイオスタティスティクス基礎論
第 2 回　講義テキスト

＜単回帰分析＞

飼育・栽培条件と動植物の生長の関係など、ある変数の変化が別の変数に影響を与える場合があります。このような変数間の関係をモデル化するための統計手法として回帰分析（regression analysis）が挙げられます。変数間の関係を統計的にモデル化することで、変数間に存在する因果関係について理解したり、一方の変数から他方の変数を予測したりすることができるようになります。

ここでは、まず、2 つの変数間の関係を“直線的な関係として”モデル化する単回帰分析（simple regression analysis）について解説します。なお、今回も前回と同様にイネのデータ（Zhao et al. 2011, Nature Communications 2:467）の解析を例に、単回帰分析の仕組みについて説明していきます。

まずは、前回と同じようにしてイネのデータを読み込みます。以下のコマンドを入力する前に、R の作業ディレクトリを 2 つの入力ファイル（RiceDiversityPheno.csv, RiceDiversityLine.csv）があるディレクトリ（フォルダ）に変更しておく必要があります。

```
> pheno <- read.csv("RiceDiversityPheno.csv")        # csv ファイルの読み込み
> line <- read.csv("RiceDiversityLine.csv")
> line.pheno <- merge(line, pheno, by.x = "NSFTV.ID", by.y = "NSFTVID")
                  # line の NSFTV.ID と pheno の NSFTVID をもとにデータを結合
> head(line.pheno)                    #  最初の 6 サンプルを示す
（結果は省略）
```

読み込んだデータから単回帰分析に用いるデータだけを抜き出して、解析データの準備を行います。ここでは、草丈（Plant.height）と開花タイミング（Flowering.time.at.Arkansas）間の関係を解析します。なお、後ほど使う遺伝的背景を表す主成分得点（PC1～PC4）も抜き出しておきます。また、欠測値をもつサンプルについてもあらかじめ取り除いておきます。

```
> data <- data.frame(
         height = line.pheno$Plant.height,            #  草丈
         flower = line.pheno$Flowering.time.at.Arkansas,     #  開花タイミング
         PC1 = line.pheno$PC1,                        #  第 1 主成分
         PC2 = line.pheno$PC2,                        #  第 2 主成分
         PC3 = line.pheno$PC3,                        #  第 3 主成分
         PC4 = line.pheno$PC4)                        #  第 4 主成分
> data <- na.omit(data)                          #  欠測データの除去
```

まずは、両者の関係を図示します。

```
> plot(data$height ~ data$flower)
                            #  flower を x, height を y として散布図を描く
                            #  ~（tilde）を使った指定の仕方に注意！
```

図 1 にも示されているように、開花が早いものほど草丈が小さく、遅くなるほど草丈が大きくなる傾向が見てとれます。

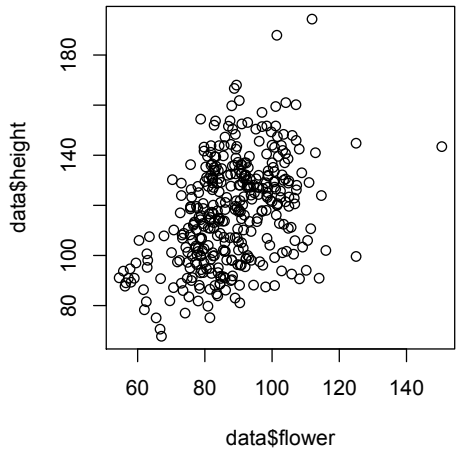

図 1. 開花のタイミング（x）と草丈（y）の関係

では、草丈の変異を開花のタイミングの違いによって説明する単回帰モデルを作成してみよう。

```
> model <- lm(height ~ flower, data = data)
         #  flower が独立変数 x、height を従属変数 y として回帰
         #  data = data は、data というオプションに data と名付けたデータを指定
```

回帰分析の結果（推定されたモデル）は、model に代入されています。

回帰分析の結果を表示させるには関数 summary を用います。

```
> summary(model)              # 関数 summary で回帰分析の結果を表示
（結果は以下に示す）
```

では上のコマンドを実行して表示された結果について順に説明していきます。

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

これは先ほど入力したコマンドが繰り返されたものです。入力した直後にこの出力が得られても、有用な情報でないように思われます。しかし、後で述べるように複数の回帰モデルを作って比較をする場合などには、どのようなモデルを想定して得られた結果であるかを再確認するのに有用だと思われます。なお、ここでは、草丈をy_i、開花のタイミングをx_iとして、

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

というモデルを想定して回帰分析を行っています。先述したように、x_iのことを独立変数（independent variable）または説明変数（explanatory variable）、y_iのことを従属変数（dependent variable）または応答変数（response variable）とよびます。μやβを回帰モデルのパラメータ（parameter）または母数、ε_iを誤差（error）とよびます。また、μを母切片（population intercept）、βを母回帰係数（population regression coefficient）とよびます。

なお、回帰モデルのパラメータμやβの真の値を直接知ることはできないため、標本をもとに推定を行います。標本をもとに推定されたパラメータμやβの推定値を、それぞれ、標本切片（sample intercept）および標本回帰係数（sample regression coefficient）とよびます。標本から推定されたμ、βの値を、以降、それぞれ、m、bで表します。m、bは、標本から推定される値であるため、偶然選ばれる標本に左右されて変動する確率変数です。したがって、ある確率分布に従います。詳細については後述します。

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

この出力は、残差の分布の概略を表しています。これを使うと簡単に回帰モデ

ルのチェックができます。例えば、モデルでは誤差の期待値（平均）は 0 となることを想定していますが、中央値（median）がそこから大幅にはずれていないか確認することができます。また、誤差の最大値と最小値、または、25%点と 75%点がほぼ同じ値をとっているかどうかで、0 を中心として左右対称の分布をしているかを確認できます。この例では、最大値が最小値に比べて少し大きめですが、それ以外は特に大きな問題は見られません。

```
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
```

回帰モデルのパラメータμ、βの推定値m、bと、それに伴う標準誤差、t値、p値が表示されています。また、各行の最後の星印は、有意水準を視覚的に確認しやすくしたものです。1 つ星は 5%、2 つ星は 1%、3 つ星は 0.1%水準で有意であることを表しています。

```
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
```

最初の行は、残差の標準偏差を表しています。これは、誤差分散σ^2の推定値をs^2とすると、sで表される値です。

2 行目は、決定係数R^2です。また、補正R^2は、自由度調整済み決定係数とよばれる統計量です。いずれも回帰が説明する変動の度合いを表しています。

3 行目は、回帰モデルの有意性を表す F検定の結果です。全ての回帰係数βが 0 であるという仮説（帰無仮説）のもとでの検定であり、この p 値が非常に小さい場合には、帰無仮説を棄却して対立仮説（回帰係数βは 0 でない）を採択すべきであると解釈されます。

では、回帰分析の結果を図示して眺めてみましょう。まず、散布図を描き、そこに回帰直線を引きます。

```
> plot(data$height ~ data$flower)
> abline(model, col = "red")    #  回帰分析の結果を abline に代入すると直線が描ける
```

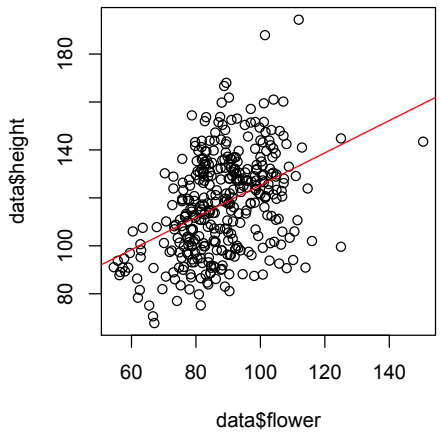

図 2. 散布図に回帰直線を加えた図

次に、回帰モデルにデータをあてはめた場合のyの値を計算し、図示してみます。

```
> height.fit <- fitted(model)        #  モデルをあてはめたときの y の値の計算
> points(data$flower, height.fit, pch = 3, col = "green")
                                     #  あてはめた値を緑色の＋で表示
```

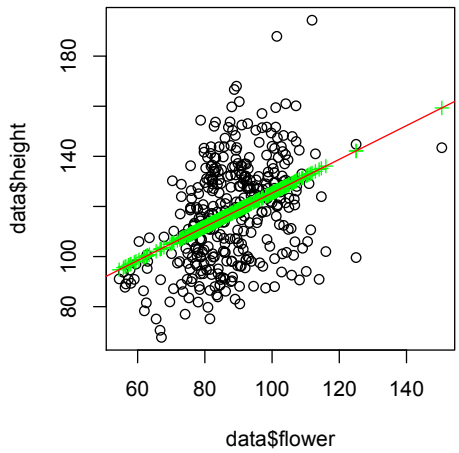

図 3. モデルをあてはめて計算される y の値は全て直線上に乗る

観察値yは、回帰モデルで説明される部分（モデルを当てはめたときの値）と、回帰で説明されない誤差部分の和として表されます。誤差部分について図示して、その関係を確認してみましょう。

```
> segments(data$flower, height.fit,
           data$flower, height.fit + resid(model), col = "gray")
         # segments は(x1, y1), (x2, y2)間で線分を描くための関数
         # x1,y1,x2,y2 は全てベクトルで表すことができ、複数の線分を一度に描ける
```

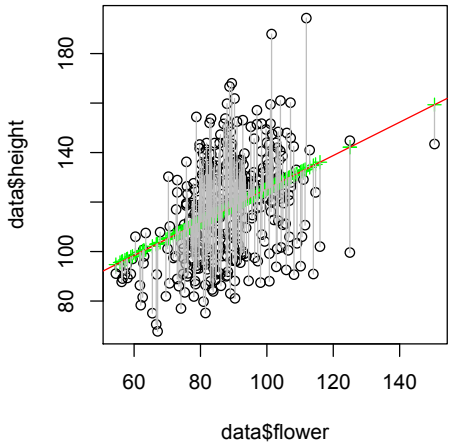

図 4. yの値は、モデルをあてはめて計算されるyの値（緑色の点）とモデルの残差（灰色の線）の和として表される

実際には観察されていない x（60, 80, …, 140）に対して、回帰モデルを用いて y を予測してみましょう。

```
> height.pred <- predict(model, data.frame(flower = seq(60, 140, 20)))
                  #  関数 predict で予測値を計算できる
                  #  data.frame(flower…)で y を予測させる新データを作成している
> points(seq(60, 140, 20), height.pred, pch = 2, col = "blue")
                  # 予測値を青（col = “blue”）い三角（pch = 2）でプロット
```

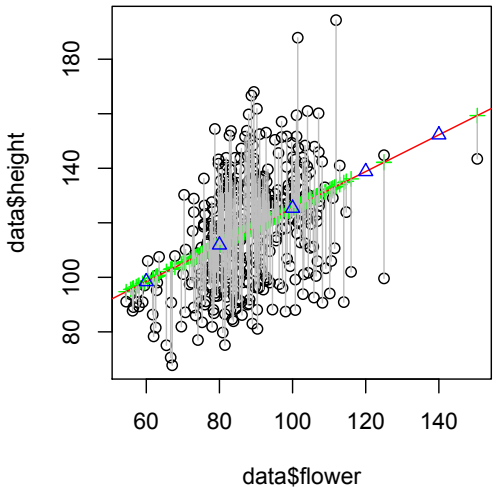

図 5. 予測値は全て回帰直線の上に乗る

＜回帰モデルのパラメータの計算方法＞

ここでは、回帰モデルの計算法について解説します。また、実際に R のコマンドを使いながら回帰係数を計算してみます。

先述したように単回帰のモデルは、

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

として表現されます。この式は、観察値y_iが、回帰方程式で説明される部分$\mu + \beta x_i$と、回帰直線では説明されない誤差部分ε_iから成ることを意味しています。上式の、μやβを動かすと、それに伴って誤差ε_iも変化します。では、どのようにして“最適な”パラメータを求めればよいのでしょうか。

何をもって“最適”とするかについては様々な基準が考えられますが、ここでは、誤差ε_iをデータ全体で最小にすることを考えてみます。ε_iは正負両方の値をとりますので、単純に和をとると互いに相殺されてしまいます。そこで、ε_iの 2 乗和（sum of squared error: SSE）を最小にすることを考えます。すなわち、

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

を最小にするようなμとβを考えてみましょう。

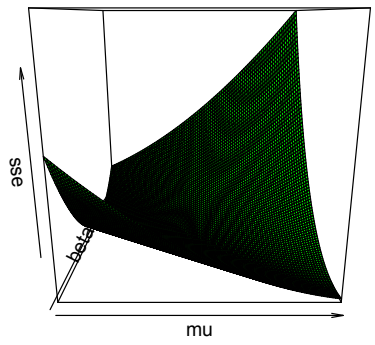

図 6. 回帰パラメータの値と残差平方和の関係

図 6 は様々なμとβに対する *SSE* の変化を表した図です。図 6 を描くためのコマンドは少し複雑ですが次のようになります。

```
> 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)
         }
 }
> persp3d(mu, beta, sse, col = "green")
```

なお、図 3 において *SSE* が最小となる点では、μやβが微小に変化しても *SSE* が変化しない（傾きがゼロ）状態になっているはずです。そこで、式(1)をμおよびβで偏微分して、その値をゼロとすることにより、最小点の座標を求めることができます。すなわち、

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

としてこれを満たすμおよびβを求めればよいということになります。このように誤差の 2 乗和を最小にするという基準にしたがって回帰モデルのパラメータを計算する方法のことを最小二乗法（least squares method）とよびます。

なお、*SSE* を最小化するμは、

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

$$\Leftrightarrow \sum_{i=1}^{n} y_i - n\mu - \beta\sum_{i=1}^{n} x_i = 0$$

$$\Leftrightarrow \mu = \frac{\sum_{i=1}^{n} y_i}{n} - \beta\frac{\sum_{i=1}^{n} x_i}{n} = \bar{y} - \beta\bar{x}$$

として計算されます。

また、*SSE* を最小化するβは、

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

として計算されます。

ここで、*SSXY* と *SSX* は、*x* と *y* の偏差積和と *x* の偏差平方和で、それぞれ、

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

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

として計算されます。

SSEを最少にするμとβをこれらパラメータの推定値とし、以降、推定値をmおよびbで表すことにします。すなわち、

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

です。

では、回帰係数を上述した式をもとにして計算してみましょう。まずは、偏差積和と偏差平方和を計算します。

```
> n <- length(x)                    #  サンプル数を n に代入
> ssxy <- sum(x * y) - n * mean(x) * mean(y)          #  偏差積和
> ssx <- sum(x^2) - n * mean(x)^2                     #  偏差平方和
```

まずは傾きbを計算します。

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

次に切片 μ を計算します。

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

計算された μ と β をもとに回帰直線を描いてみましょう。

```
> plot(y ~ x)
> abline(m, b)    # 切片 mu、傾き beta の直線を描く
```

先ほど関数 lm を用いて計算された回帰直線と同じものが描かれていることを確認してみましょう。

なお、回帰パラメータが推定されれば、与えられたx_iに対応するyの値$\hat{y}_i$を計算することができるようになります。すなわち、

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

として計算できます。これにより、観察されたxにモデルをあてはめたときのyの値を計算したり、xのみが既知の場合にyを予測したりすることができます。ここでは、観察されたxにモデルをあてはめたときのyの値を計算し、先ほど描いた図の上に点を散布してみましょう。

```
> y.hat <- m + b * x                    #  x にモデルをあてはめたときの y の値を計算
> lim <- range(c(y, y.hat))             #  y と y.hat の値の範囲を調べる
> plot(y, y.hat, xlab = "Observed", ylab = "Fitted", xlim = lim, ylim = lim)
         # y と y.hat の散布図を描く。横軸が観察値、縦軸があてはめ値
         # 計算しておいた y と y.hat の値の範囲を、x および y 軸の範囲として指定
> abline(0, 1)
                                       #  切片が 0、傾き 1 の直線（y = x）を描く
```

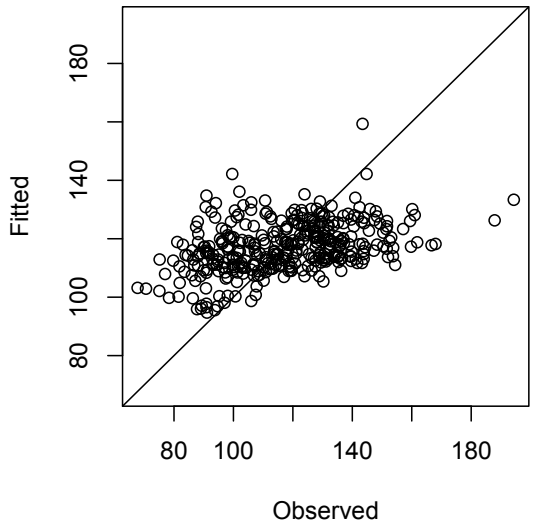

図 7. 観測値とあてはめ値の間の関係

観察値とあてはめ値の一致の度合いを調べるために両者の相関係数を計算してみましょう。

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

実は、この相関係数の 2 乗が、回帰が説明するyの変動の割合（決定係数、R^2値）になっています。両者を見比べてみましょう。

```
> cor(y, y.hat)^2
[1] 0.1671894
> summary(model)
（結果を一部省略）
Multiple R-squared: 0.1672,        Adjusted R-squared: 0.1649
（結果を一部省略）
```

＜回帰モデルの有意性検定＞

変数間の直線的な関係が強い場合には回帰直線がよくあてはまり、両変数間の関係を回帰直線でうまくモデル化できます。しかし、変数間の直線的な関係が明瞭でない場合には、回帰直線によるモデル化がうまく行きません。ここでは、推定された回帰モデルの有効性を客観的に確認するための方法として、分散分析を用いた検定法について説明します。

まずは、再度、単回帰を行ってみましょう。

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

得られた回帰モデルの有意性は、関数 anova を用いて検定できます。

```
> 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
```

分散分析の結果、変数 flower の項は高度に有意（$p < 0.001$）であり、開花のタイミング flower が草丈 height に影響を与えるという回帰モデルの有効性が確認できます。

回帰モデルの分散分析では、以下に示すような計算が行われます。まず、「回帰で説明される平方和」（回帰モデルをあてはめて計算される値 $\hat{y}_i$ の偏差平方和）は、以下のようにして計算できます。

$$
\begin{aligned}
SSR &= \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 \\
&= \sum_{i=1}^{n} \left(\mu + bx_i - (\mu + b\bar{x})\right)^2 \\
&= b^2 \sum_{i=1}^{n} (x_i - \bar{x})^2 \\
&= b^2 \cdot SSX = b \cdot SSXY
\end{aligned}
$$

また、観察値 y の平均からの偏差の平方和は、回帰で説明される平方和 *SSR* と残差平方和 *SSE* の和として表されます。すなわち、

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

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

では、上の式を用いて実際に計算してみましょう。まずは、回帰で説明される平方和 *SSR* と残差平方和 *SSE* を計算します。

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

次に、平方和を自由度で割った平均平方を計算します。

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

最後に回帰の平均平方を誤差の平均平方で割り、F値を計算します。さらに、計算された F値に対応する p 値を計算します。

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

得られる結果は、先ほど関数 anova を用いて計算された結果と一致しています。

なお、回帰の分散分析の結果は、関数 summary を用いて表示される回帰分析の結果の中にも含まれています。

```
> summary(model)
（結果を一部省略）
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

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」は、残差の平均平方の平方根となっています。

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

「Multiple R-squared」（R^2）は、決定係数（coefficient of determination）とよばれる値で、*SSR* と *SSY* の比です。

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

「Adjusted R-squared」（R^2_{adj}）は、自由度調整済決定係数とよばれる値で、次のように計算できます。

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

また、「*F*-statistic」は、分散分析で flower の効果として表されている F 値とその p 値に一致します。また、flower の回帰係数について計算されている t 値を 2 乗すると F 値になります（$8.630^2 = 74.477$）。

なお、R^2およびR^2_{adj}は、*SSR*、*SSY*、*SSE* を用いて以下のように表すこともできます。

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

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

ここで、pはモデルに含まれるパラメータの数で、単回帰モデルでは、$p = 2$になります。R^2_{adj}は、モデルに含まれるパラメータの数が多ければ多いほど、調整量が大きくなる（残差平方和の小ささを低く見積もる）ことが分かります。

＜回帰係数の推定値が従う分布＞

先述したように回帰係数μとβの推定値bとmは、標本から推定される値であり、偶然選ばれた標本に左右される確率変数です。したがって、推定値bとmは確率分布をもちます。ここでは、推定値の従う分布について考えます。

まず、bについて考えます。bは、

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

と表すことができます。

したがって、 推定値bの平均は、

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

です。すなわち、推定値bの平均は、真の値βに一致します。ここで、μ^*は、y_iをx_iでなく$x_i - \bar{x}$に回帰した場合の定数項で、

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

と表されます。

推定値bの分散は、

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

となります。なお、ここでσ^2は、残差分散$\sigma^2 = \mathbb{V}(y_i) = \mathbb{V}(e_i)$ です。

なお、推定値bは、y_iの線形結合

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

です。y_iは正規分布に従うため、その線形結合であるbもまた正規分布に従います。すなわち、

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

です。

いっぽう、推定値mは、$m = \bar{y} - \beta\bar{x}$と表せます。

したがって、平均は、

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

となります。推定値mの平均も、やはり、真の値μに一致します。

次に分散は、

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

となります。

y_iは正規分布に従うため、$m = \bar{y} - \beta\bar{x}$と表される$m$もまた正規分布に従います。すなわち、

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

なお、誤差分散σ^2の真の値は未知ですが、これを残差分散s^2で置き換えることができます。すなわち、

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

です。この値は分散分析の際に計算した残差の平均平方です。

このとき、bに関する統計量

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

は、帰無仮説

$$H_0: \beta = \beta_0$$

のもとで、自由度$n-2$のt分布に従います。

このとき、β（すなわちβ_0）が$1-\alpha$の確率で含まれる区間、すなわち、$(1-\alpha)100\%$信頼区間は以下のように計算されます。

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

ここで、$t_{n-2,1-\alpha/2}$は自由度$n-2$における両側5%（$\alpha = 0.05$）または1%（$\alpha = 0.01$）水準の棄却限界値です。

また、mについても統計量

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

は、帰無仮説

$$H_0: m = \mu_0$$

のもとで、自由度$n-2$のt分布に従います。

このとき、μ（すなわちμ_0）が$1-\alpha$の確率で含まれる区間、すなわち、$(1-\alpha)100\%$信頼区間は以下のように計算されます。

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

では、ここまでに求めたbとmについて検定や信頼区間の計算を行ってみましょう。

まずbについて、帰無仮説$H_0: \beta = 0$について検定してみます。

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

この検定の結果は、回帰分析結果として既に表示されていたものです。

```
> 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）
```

上で行った仮説検定は、任意のβ_0について行うことができます。例えば、帰無仮説$H_0: \beta = 0.5$について検定してみましょう。

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

結果は、5%水準で有意です。これは、上述した 95%信頼区間に 0.5 が「含まれていない」ことを意味しています。

では、mについても検定と信頼区間の計算を行ってみましょう。まず、帰無仮説$H_0: m = 0$の検定を行ってみましょう。

```
> 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
```

この結果もやはり、既に計算されていたものでした。

```
> 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）
```

（p 値が若干合わないのは丸め誤差によるものかもしれません）

最後に、帰無仮説$H_0: m = 50$について検定してみいましょう。

```
> 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
```

結果は、5%水準でも有意ではありませんでした。これは、上述した 95%信頼区間に 50 が「含まれている」ことを意味しています。

＜回帰係数やあてはめ値の信頼区間＞

関数 predict には様々な機能があります。まずは回帰モデルを単純に引数として関数を使ってみましょう。するとモデルをあてはめたときのyの値が計算されます。その値は関数 fitted で計算されるものと全く同じです。

```
> 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
```

オプション interval と level を設定すると、モデルをあてはめたときのyの信頼区間を計算できます。

```
> 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
```

関数 predict を用いてyの信頼区間を図示してみましょう。

```
> 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")
```

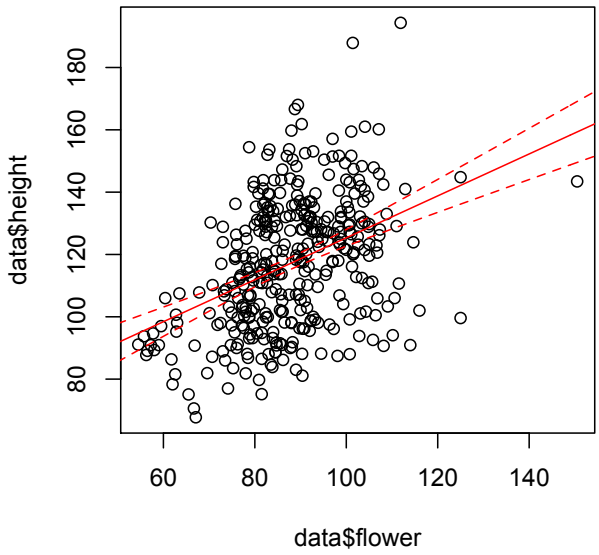

図 8. モデルをあてはめたときのyの信頼区間
xの平均付近は狭く、そこから離れるほど広くなる

yの信頼区間は次のように計算できます。まず、x^*を与えたときのy、すなわち、$y_m = \mathbb{E}(y|x = x^*) = \mu + \beta x^*$を推定することを考えます。標本から推定された回帰係数をbとすると、y_mの推定値は、$\hat{y}_m = m + bx^*$となります。ここで、mもbも確率変数であるために、$\hat{y}_m$もまた確率変数となります。$\hat{y}_m$は

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

と表され、> 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

では、上式にしたがって、x^*をあたえたときのyの推定値の信頼区間を図示してみましょう。

```
> 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")
```

図 8 と同じ図が描かれることを確認してみましょう。

＜多項式回帰モデルと重回帰モデル＞

ここまでは、2 つの変数間の関係を直線で表す回帰モデルをデータに適用してきました。ここでは、回帰モデルを少し拡張してみましょう。

まず、多項式回帰（polynomial regression)とよばれる方法で回帰を行ってみましょう。多項式回帰では、

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

というかたちでxの 2 次以上の項も用いて回帰を行います。まずは、xの 1 次の項と 2 次の項を用いて回帰を行ってみましょう。

```
> model.quad <- lm(height ~ flower + I(flower^2), data = data)
> summary(model.quad)
（結果を一部省略）
Multiple R-squared: 0.1915,        Adjusted R-squared: 0.1871
（結果を一部省略）
```

多項式回帰モデルで説明されるyの変動の割合（決定係数R^2）が、単回帰モデルに比べて向上していることが分かります。

なお、後述しますがこの値だけで多項式回帰モデルが優れていると判断してはいけません。なぜなら、多項式回帰モデルのほうが単回帰モデルに比べてパラメータが多く、データへモデルの当てはめを行う場合の柔軟性が高くなっているからです。柔軟性を上げることでモデルのデータへのあてはまりを向上させるのは簡単なことで、極端な例を挙げるとデータ数と同じだけのパラメータがあればモデルをデータに完全にあてはめることができます（その場合、決定係数R^2は完全に 1 に一致します)。したがって、最適なモデルを選択する場合には、何らかの統計的基準による注意深い検討が必要となります。これについては後述します。

では、多項式回帰の結果を信頼区間付きで図示してみましょう。

```
> pred <- data.frame(flower = 50:160)
                    #  計算範囲の指定（独立変数 x を与える）
> pc <- predict(model.quad, int = "c", newdata = pred)
                    #  与えられた x に対して、あてはめ値と信頼区間を計算する
> plot(data$height ~ data$flower)            #  散布図の描画
> matlines(pred$flower, pc, lty = c(1, 2, 2), col = "red")
                    #  あてはめ値（多項式回帰曲線）およびその信頼区間の描画
```

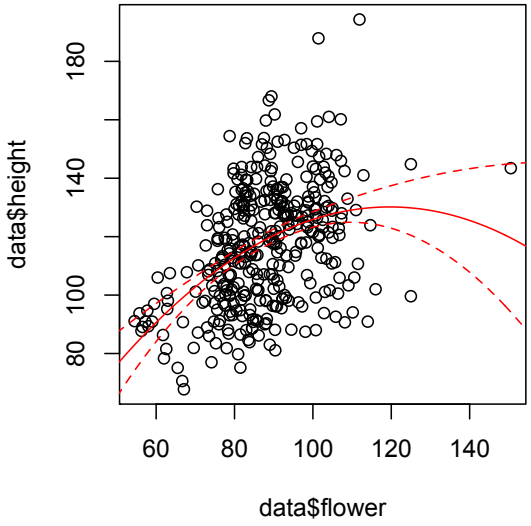

図 9. 2 次の多項式回帰の結果。開花のタイミングが播種後 120 日以上の場合には信頼性が低いことが分かる。

では、多項式回帰モデルと単回帰モデルの説明力を視覚的に比較してみましょう。

```
> pred <- data.frame(flower = 50:160)
                    #  計算範囲の指定（独立変数 x を与える）
> pc <- predict(model.quad, int = "c", newdata = pred)
                    #  与えられた x に対して、あてはめ値と信頼区間を計算する
> plot(data$height ~ data$flower)              #  散布図の描画
> matlines(pred$flower, pc, lty = c(1, 2, 2), col = "red")
                    #  あてはめ値（多項式回帰曲線）およびその信頼区間の描画
```

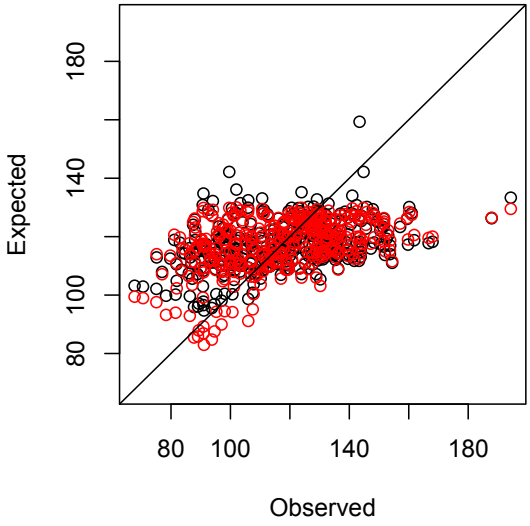

図 10. 単回帰モデル（黒）および 2 次の多項式モデル（赤）における
あてはめ値と観察値の関係

では、2 次の多項式モデルの説明力の向上が統計的に有意かどうか検定してみましょう。有意性は、2 つのモデルの残差平方和の違いが、一方を内包している側のモデル（ここでは Model 2 が Model 1 を内包している）の残差平方和に比べて十分大きいかを*F*検定によって検定します。

```
> 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
```

結果、両モデルの残差分散の違いは高度に有意（$p < 0.001$）であることが分かります。すなわち、Model 2 が Model 1 に比べて有意に説明力が高いといえます。

では、3 次の多項式回帰モデルをあてはめ、2 次のモデルに比べて有意に説明力が高いか検定してみましょう。

```
> model.cube <- lm(height ~ flower + I(flower^2) + I(flower^3), data = data)
> summary(model.cube)
（結果を一部省略）
Multiple R-squared: 0.1931,         Adjusted R-squared: 0.1866
（結果を一部省略）
> 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
```

2 次のモデルに比べ、3 次のモデルは説明力が少しだけ向上しています。しかし、その差は統計的に有意ではありません。すなわち、2 次のモデルを 3 次のモデルに拡張するのは良策でないことが分かります。

最後に、重回帰（multiple linear regression)モデルをあてはめてみましょう。重回帰では、

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

というかたちで複数の説明変数（$x_{1i}, x_{2i}, \ldots, x_{pi}$）を用いて回帰を行います。第 1 回の講義において、草丈（height）が遺伝的背景の違いによっても異なることをグラフで確認しました。ここでは 4 主成分の得点として表された遺伝的背景（PC1～PC4）を用いて草丈を説明する重回帰モデルを作成してみます。

```
> model.wgb <- lm(height ~ PC1 + PC2 + PC3 + PC4, data = data)
> summary(model.wgb)
（結果を一部省略）
Multiple R-squared: 0.3388,         Adjusted R-squared: 0.3316
（結果を一部省略）
> anova(model.wgb)
（結果を一部省略）
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
```

回帰モデルの決定係数が、先ほどの多項式回帰モデルに比べても高いことが分かります。分散分析の結果を見てもいずれの主成分も有意で、回帰に含める必要があることが分かります。

最後に、多項式回帰モデルと重回帰モデルを組合せてみましょう。

```
> model.all <- lm(height ~ flower + I(flower^2) + PC1 + PC2 + PC3 + PC4,
                  data = data)
> summary(model.all)
（結果を一部省略）
Multiple R-squared: 0.4045,         Adjusted R-squared: 0.3947
（結果を一部省略）
> 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
```

草丈に対する遺伝的背景の効果は非常に大きいのですが、それだけでなく、開花のタイミングの効果についても加えたほうが、モデルの説明力が向上することが分かります。

最後に、最初に作成した単回帰モデルと最後に作成した重回帰モデルを、観察値とあてはめ値の対散布を描いて比較してみましょう。

```
> 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)
```

結果、遺伝的背景や 2 次の項を考慮することなどにより大幅にモデルの説明力が上がっていることが分かります。しかし、一方で、開花のタイミングが遅い（180 日以降）2 つの品種・系統については、最終的に得られたモデルでも十分に説明できていないことも分かります。新たな要因を独立変数として加えるなどして、モデルを改良する余地が残っているのかもしれません。

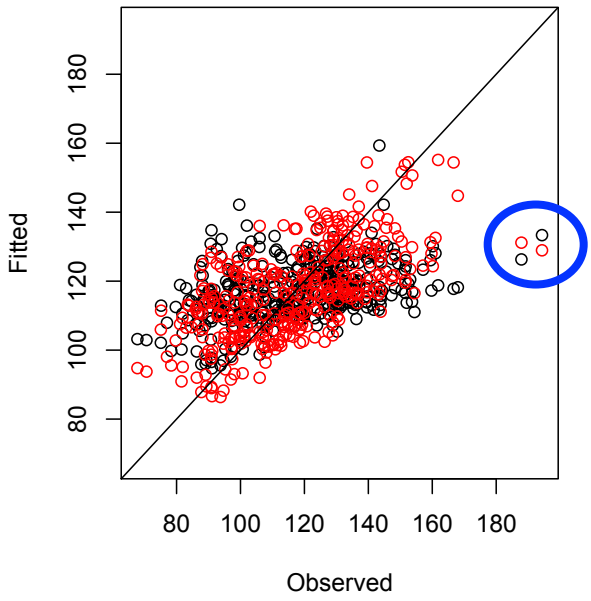

図 11. 単回帰モデル（黒）と重回帰モデル（赤）の比較
横軸が観察値で縦軸がモデルをあてはめた値
青い円内のサンプルでは当てはまりの悪さが解決していない

＜実験計画法と分散分析＞

実験結果をもとに結論を得ようとする場合に、いつも問題になるのが観察値に含まれる誤差の存在です。どれほど精密な実験を行っても誤差は不可避なものであり、特に圃場での実験では圃場内にみられる微細な環境の違いによって誤差が生じます。したがって、誤差があってもそれに影響されずに客観的な結論を得るために工夫された方法が実験計画法（experimental design）です。

まず、実験を計画する上で、非常に重要なのは以下に示す Fisher の 3 原則（Fisher's three principles）です。

(1) 反復（replication）：実験結果について統計的検定ができるようにするために、同じ処理について反復を設けます。例えば、１つの品種を複数回評価するようにします。1 反復分に相当する実験単位のことをプロット（plot）とよびます。

(2) 無作為化（randomization）：誤差の影響がランダムになるようにする操作のことを無作為化といいます。例えば、圃場試験の例では、品種を圃場内のプロットにサイコロや乱数を用いてランダムに割り付けます。

(3) 局所管理（local control）：局所管理とは圃場をブロック（block）とよばれる区画に分け、各ブロッック内の環境条件ができるだけ均質になるように管理することです。圃場試験の例では、圃場のあるまとまった区画をブロックという小さな単位に分割することで、ブロック内の栽培環境ができるだけ均質になるようにします。圃場全体の栽培環境を均質にするより、ブロック毎に均質かするほうが容易です。

なお、圃場をいくつかのブロックに分割して、ブロック内ではできるだけ栽培環境が均質になるようにして行う実験法を乱塊法（randomized block design）といいます。乱塊法では圃場をブロックに分割して、各ブロック内での品種の割り付けは無作為に行います。ブロックの数が反復数となります。

では、簡単なシミュレーションを通して、乱塊法における統計検定の方法について説明します。まずはシミュレーションを開始する前に、乱数の「種」を設定しましょう。乱数の種とは擬似乱数を発生するためのもとになる値です。

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

では、早速シミュレーションを始めましょう。なお、ここでは、16 個のプロット（plot）が 4×4 で配置されている圃場を考えます。そして、その圃場に地力の勾配がある状況を考えます。

```
> 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
```

もっとも地力が高いところでは+4, 低いところでは-4 の効果があるとしました。

ここで、Fisher の 3 原則にしたがってブロックを配置します。ブロックは、地力の違いをうまく反映できるように配置します。

```
> 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"
```

次に、Fisher の 3 原則にしたがって品種を各ブロックに無作為に配置します。まずはそのための準備をしましょう。

```
> variety <- c("A", "B", "C", "D")           #  4 つの品種を試験する
> sample(variety)
[1] "B" "C" "A" "D"        # 関数 sample で 4 品種を無作為に並べることができる
> sample(variety)          # 実行する毎に無作為に並び替えられる
[1] "D" "A" "B" "C"
```

では、各ブロックに無作為に品種を割り付けてみましょう。

```
> 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"
```

4 品種にみられる遺伝的能力の違いを考えます。A～D 品種の遺伝的能力をそれぞれ+4, +2, -2, -4 とします。

```
> 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
```

環境によるばらつきを平均 0、標準偏差 2.5 の正規分布からの乱数として生成します。

```
> 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
```

なお、上のコマンドでは乱数を発生していますが、皆さんも教科書と同じ値が得られると思います。これは、発生される乱数が疑似乱数であり、ある一定の規則に従って生成されているためです。なお、先に設定した乱数の種の値を変えると、上に示されている値と同じものは生成されません。また、実行する毎に異なる数値が生成されます。

最後に、全体平均、地力の勾配、品種の遺伝的能力、環境によるばらつきを足し合わせ、形質の観察値を模擬的に生成します。

```
> 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
```

模擬的に作成したデータを視覚化してみましょう。

```
> 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)
```

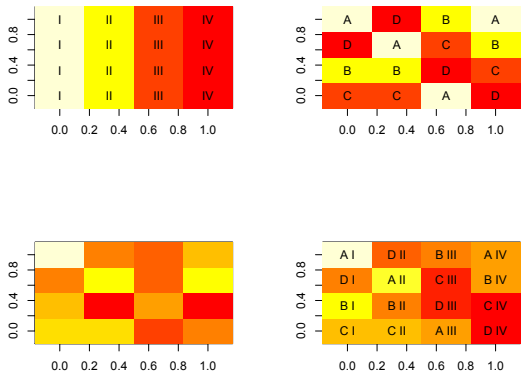

図 12. 地力の勾配（左上）、品種の遺伝効果（右上）、
環境によるばらつき（左下）および形質の観察値（右下）

分散分析を行う前に行列のかたちになっているデータを列データに直し、束ね直します。

```
> 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
（結果は省略）
```

作成したデータを関数 interaction.plot を使って図示してみます。

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

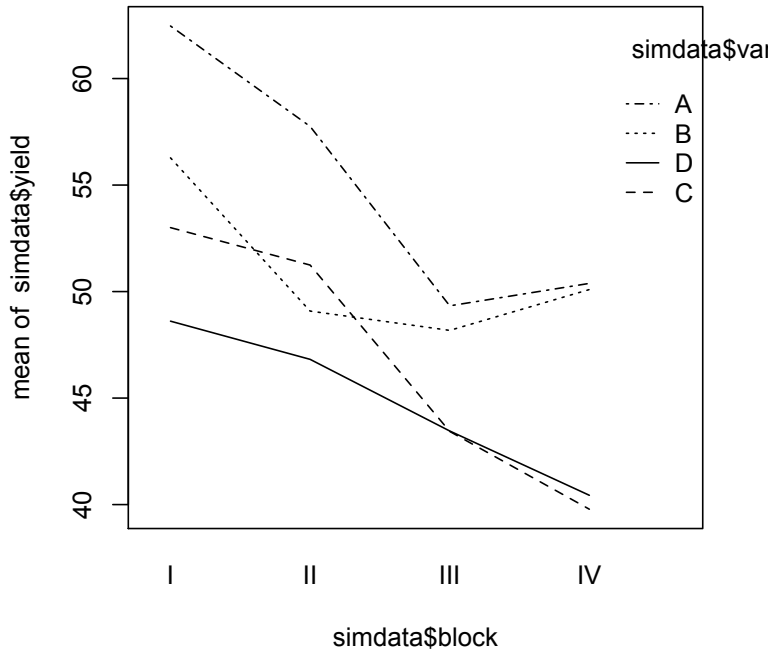

図 13. 模擬的に生成された品種・ブロック毎の収量データ
品種間差と同じようにブロック間差が大きいことが見てとれる

では、準備したデータを用いて分散分析を行ってみましょう。

```
> 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
```

ブロック効果も品種効果も高度に有意であることが分かります。なお、前者は検証の対象ではなく、あくまで品種効果を正しく推定するためにモデルに組み込まれていることに注意しましょう。

上述した分散分析は、回帰モデルの推定のための関数 lm を用いても行うことができます。

```
> 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
```

関数 lm では、ダミー変数を用いて回帰の枠組みの中で分散分析を行っています。なお、関数 model.matrix を使うとダミー変数の設定を確認することができます。

```
> model.matrix(res)
（結果を省略）
> summary(res)
（結果を省略）
```

＜分散分析の計算法＞

いま、i 番目の品種の j 番目のブロックにおける形質の観測値を x_{ij} とします。このとき、x_{ij} は次のように書くことができます。

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

ここで、$\bar{x}_i.$，$\bar{x}._j$，$\bar{x}..$ はそれぞれ、i 番目の品種についての平均、j 番目のプロットにおける平均、総平均を表します。すなわち、

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

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

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

となります。ここで、mは品種数、rはブロック数です。

観察値の総平均からの差の平方の和（平方和, sum of squares）は、

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

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

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

と分割することができます。1 項目が品種に起因する平方和、2 項目がブロックに起因する平方和、3 項目が誤差に起因する平方和です。

分割された平方和を自由度で割ったものを平均平方（mean square）といいます。平均平方はそれぞれの変動をもたらす原因による不偏分散（unbiased variance）に対応します。分散分析では品種の平均平方を誤差の平均平方で割った比を計算し、その比が帰無仮説（品種に起因する分散は 0）のもとで自由度 $m-1$、$(m-1)(r-1)$ の F 分布に従うことを利用して品種の効果の有意性検定を行います。

次のページに関数 aov を使わずに分散分析を行うための R のコードを示します。

```
> 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
```

＜乱塊法と完全無作為配置＞

Fisher の 3 原則の 1 つである局所管理は、プロット間の異質性が高い圃場で精度の高い実験を行うために非常に重要です。ここでは、先ほどと同じ環境条件を想定して、ブロックを設けずに実験を行うことを考えてみます。

先ほどのシミュレーション実験では列毎にブロック化し、そのブロック内で A, B, C, D を無作為に配置しました。ここでは 4 品種×4 反復のプロットを、圃場全体に完全に無作為に配置します。このようにブロックを配置せず、完全に無作為に配置して行う実験を「完全無作為配置（completely randomized design）」とよびます。

```
> 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"
```

今回は、圃場全体に無作為に割り振っているので、列毎に品種の出現頻度が異なることに注意しましょう。

完全無作為配置の品種の並びに合わせて遺伝効果を割り当てます。

```
> 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
```

先ほどのシミュレーション実験と同様に、全体平均、地力の勾配、品種の遺伝効果、環境によるばらつきを足し合わせます。

```
> 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
```

では、模擬的に生成されたデータについて分散分析を行ってみましょう。先ほどの実験とは異なりブロックを設定していないのでブロック効果は含めないで品種効果だけを含むモデルで回帰分析を行います。

```
> 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）
```

上の例では、品種効果は、有意ではありません。これは地力の勾配により誤差が大きくなり、品種間の遺伝的な違いが十分な精度で推定できなくなっているためだと考えられます。

なお、上述したシミュレーション実験を 100 回繰り返して行ってみました（次ページに示します)。その結果、乱塊法を用いた実験では 100 回のうち 94 回の実験で品種効果を検出（有意水準 5%）できましたが、完全無作為配置では 66 回しか検出できませんでした。また、有意水準を 1%に設定すると、品種効果が検出される回数がそれぞれ 70 回、30 回となりました（完全無作為配置では 70 回品種効果を見逃す！)。このことからも、地力の勾配など、ブロックを設定することである程度制御ができるような場合には、乱塊法の採用が非常に有効であることが分かります。時間と労力をかけて行う実験をできるだけ有効なものにするためには、実験計画を適切に組むことが非常に重要なのです。

```
> 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
```

＜レポート課題＞

講義で学んだ回帰分析法を用いて、いくつかの形質で、形質と遺伝的背景間の関係について解析してみて下さい。

提出方法：

- レポートは「pdf ファイル」として作成し、メール添付で提出する。
- メールは、「report@iu.a.u-tokyo.ac.jp 宛」に送る。
- レポートの最初に、「所属、学生番号、名前を忘れず」に。
- 提出期限は、4 月 30 日

K.W.

ギリシャ文字などからできています

気づきましたか？