

バイオスタティクス基礎論 第2回講義テキスト

岩田洋佳 hiroiwata@g.ecc.u-tokyo.ac.jp

2021/4/16

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。

```
# this data set was analyzed in Zhao 2011 (Nature Communications 2:467)
pheno <- read.csv("RiceDiversityPheno.csv")
line <- read.csv("RiceDiversityLine.csv")
line.pheno <- merge(line, pheno, by.x = "NSFTV.ID", by.y = "NSFTVID")
head(line.pheno)[,1:12]

##   NSFTV.ID GSOR.ID          IRGC.ID Accession.Name Country.of.origin Latitu
## 1      1    301001 To be assigned        Agostano       Italy  41.8719
## 2      3    301003           117636 Ai-Chiao-Hong      China  27.9025
## 3      4    301004           117601     NSF-TV 4      India  22.9030
## 4      5    301005           117641     NSF-TV 5      India  30.4726
## 5      6    301006           117603        ARC 7229      India  22.9030
## 6      7    301007 To be assigned        Arias      Indonesia -0.7892
## 7      8    301008           117642      Tengku 10      Indonesia -0.7892
## 8      9    301009           117643      Tengku 11      Indonesia -0.7892
## 9     10    301010           117644      Tengku 12      Indonesia -0.7892
## 10    11    301011           117645      Tengku 13      Indonesia -0.7892
## 11    12    301012           117646      Tengku 14      Indonesia -0.7892
## 12    13    301013           117647      Tengku 15      Indonesia -0.7892
## 13    14    301014           117648      Tengku 16      Indonesia -0.7892
## 14    15    301015           117649      Tengku 17      Indonesia -0.7892
## 15    16    301016           117650      Tengku 18      Indonesia -0.7892
## 16    17    301017           117651      Tengku 19      Indonesia -0.7892
## 17    18    301018           117652      Tengku 20      Indonesia -0.7892
## 18    19    301019           117653      Tengku 21      Indonesia -0.7892
## 19    20    301020           117654      Tengku 22      Indonesia -0.7892
## 20    21    301021           117655      Tengku 23      Indonesia -0.7892
## 21    22    301022           117656      Tengku 24      Indonesia -0.7892
## 22    23    301023           117657      Tengku 25      Indonesia -0.7892
## 23    24    301024           117658      Tengku 26      Indonesia -0.7892
## 24    25    301025           117659      Tengku 27      Indonesia -0.7892
## 25    26    301026           117660      Tengku 28      Indonesia -0.7892
## 26    27    301027           117661      Tengku 29      Indonesia -0.7892
## 27    28    301028           117662      Tengku 30      Indonesia -0.7892
## 28    29    301029           117663      Tengku 31      Indonesia -0.7892
## 29    30    301030           117664      Tengku 32      Indonesia -0.7892
## 30    31    301031           117665      Tengku 33      Indonesia -0.7892
## 31    32    301032           117666      Tengku 34      Indonesia -0.7892
## 32    33    301033           117667      Tengku 35      Indonesia -0.7892
## 33    34    301034           117668      Tengku 36      Indonesia -0.7892
## 34    35    301035           117669      Tengku 37      Indonesia -0.7892
## 35    36    301036           117670      Tengku 38      Indonesia -0.7892
## 36    37    301037           117671      Tengku 39      Indonesia -0.7892
## 37    38    301038           117672      Tengku 40      Indonesia -0.7892
## 38    39    301039           117673      Tengku 41      Indonesia -0.7892
## 39    40    301040           117674      Tengku 42      Indonesia -0.7892
## 40    41    301041           117675      Tengku 43      Indonesia -0.7892
## 41    42    301042           117676      Tengku 44      Indonesia -0.7892
## 42    43    301043           117677      Tengku 45      Indonesia -0.7892
## 43    44    301044           117678      Tengku 46      Indonesia -0.7892
## 44    45    301045           117679      Tengku 47      Indonesia -0.7892
## 45    46    301046           117680      Tengku 48      Indonesia -0.7892
## 46    47    301047           117681      Tengku 49      Indonesia -0.7892
## 47    48    301048           117682      Tengku 50      Indonesia -0.7892
## 48    49    301049           117683      Tengku 51      Indonesia -0.7892
## 49    50    301050           117684      Tengku 52      Indonesia -0.7892
## 50    51    301051           117685      Tengku 53      Indonesia -0.7892
## 51    52    301052           117686      Tengku 54      Indonesia -0.7892
## 52    53    301053           117687      Tengku 55      Indonesia -0.7892
## 53    54    301054           117688      Tengku 56      Indonesia -0.7892
## 54    55    301055           117689      Tengku 57      Indonesia -0.7892
## 55    56    301056           117690      Tengku 58      Indonesia -0.7892
## 56    57    301057           117691      Tengku 59      Indonesia -0.7892
## 57    58    301058           117692      Tengku 60      Indonesia -0.7892
## 58    59    301059           117693      Tengku 61      Indonesia -0.7892
## 59    60    301060           117694      Tengku 62      Indonesia -0.7892
## 60    61    301061           117695      Tengku 63      Indonesia -0.7892
## 61    62    301062           117696      Tengku 64      Indonesia -0.7892
## 62    63    301063           117697      Tengku 65      Indonesia -0.7892
## 63    64    301064           117698      Tengku 66      Indonesia -0.7892
## 64    65    301065           117699      Tengku 67      Indonesia -0.7892
## 65    66    301066           117700      Tengku 68      Indonesia -0.7892
## 66    67    301067           117701      Tengku 69      Indonesia -0.7892
## 67    68    301068           117702      Tengku 70      Indonesia -0.7892
## 68    69    301069           117703      Tengku 71      Indonesia -0.7892
## 69    70    301070           117704      Tengku 72      Indonesia -0.7892
## 70    71    301071           117705      Tengku 73      Indonesia -0.7892
## 71    72    301072           117706      Tengku 74      Indonesia -0.7892
## 72    73    301073           117707      Tengku 75      Indonesia -0.7892
## 73    74    301074           117708      Tengku 76      Indonesia -0.7892
## 74    75    301075           117709      Tengku 77      Indonesia -0.7892
## 75    76    301076           117710      Tengku 78      Indonesia -0.7892
## 76    77    301077           117711      Tengku 79      Indonesia -0.7892
## 77    78    301078           117712      Tengku 80      Indonesia -0.7892
## 78    79    301079           117713      Tengku 81      Indonesia -0.7892
## 79    80    301080           117714      Tengku 82      Indonesia -0.7892
## 80    81    301081           117715      Tengku 83      Indonesia -0.7892
## 81    82    301082           117716      Tengku 84      Indonesia -0.7892
## 82    83    301083           117717      Tengku 85      Indonesia -0.7892
## 83    84    301084           117718      Tengku 86      Indonesia -0.7892
## 84    85    301085           117719      Tengku 87      Indonesia -0.7892
## 85    86    301086           117720      Tengku 88      Indonesia -0.7892
## 86    87    301087           117721      Tengku 89      Indonesia -0.7892
## 87    88    301088           117722      Tengku 90      Indonesia -0.7892
## 88    89    301089           117723      Tengku 91      Indonesia -0.7892
## 89    90    301090           117724      Tengku 92      Indonesia -0.7892
## 90    91    301091           117725      Tengku 93      Indonesia -0.7892
## 91    92    301092           117726      Tengku 94      Indonesia -0.7892
## 92    93    301093           117727      Tengku 95      Indonesia -0.7892
## 93    94    301094           117728      Tengku 96      Indonesia -0.7892
## 94    95    301095           117729      Tengku 97      Indonesia -0.7892
## 95    96    301096           117730      Tengku 98      Indonesia -0.7892
## 96    97    301097           117731      Tengku 99      Indonesia -0.7892
## 97    98    301098           117732      Tengku 100     Indonesia -0.7892
## 98    99    301099           117733      Tengku 101     Indonesia -0.7892
## 99    100   301100           117734      Tengku 102     Indonesia -0.7892
## 100   101   301101           117735      Tengku 103     Indonesia -0.7892
## 101   102   301102           117736      Tengku 104     Indonesia -0.7892
## 102   103   301103           117737      Tengku 105     Indonesia -0.7892
## 103   104   301104           117738      Tengku 106     Indonesia -0.7892
## 104   105   301105           117739      Tengku 107     Indonesia -0.7892
## 105   106   301106           117740      Tengku 108     Indonesia -0.7892
## 106   107   301107           117741      Tengku 109     Indonesia -0.7892
## 107   108   301108           117742      Tengku 110     Indonesia -0.7892
## 108   109   301109           117743      Tengku 111     Indonesia -0.7892
## 109   110   301110           117744      Tengku 112     Indonesia -0.7892
## 110   111   301111           117745      Tengku 113     Indonesia -0.7892
## 111   112   301112           117746      Tengku 114     Indonesia -0.7892
## 112   113   301113           117747      Tengku 115     Indonesia -0.7892
## 113   114   301114           117748      Tengku 116     Indonesia -0.7892
## 114   115   301115           117749      Tengku 117     Indonesia -0.7892
## 115   116   301116           117750      Tengku 118     Indonesia -0.7892
## 116   117   301117           117751      Tengku 119     Indonesia -0.7892
## 117   118   301118           117752      Tengku 120     Indonesia -0.7892
## 118   119   301119           117753      Tengku 121     Indonesia -0.7892
## 119   120   301120           117754      Tengku 122     Indonesia -0.7892
## 120   121   301121           117755      Tengku 123     Indonesia -0.7892
## 121   122   301122           117756      Tengku 124     Indonesia -0.7892
## 122   123   301123           117757      Tengku 125     Indonesia -0.7892
## 123   124   301124           117758      Tengku 126     Indonesia -0.7892
## 124   125   301125           117759      Tengku 127     Indonesia -0.7892
## 125   126   301126           117760      Tengku 128     Indonesia -0.7892
## 126   127   301127           117761      Tengku 129     Indonesia -0.7892
## 127   128   301128           117762      Tengku 130     Indonesia -0.7892
## 128   129   301129           117763      Tengku 131     Indonesia -0.7892
## 129   130   301130           117764      Tengku 132     Indonesia -0.7892
## 130   131   301131           117765      Tengku 133     Indonesia -0.7892
## 131   132   301132           117766      Tengku 134     Indonesia -0.7892
## 132   133   301133           117767      Tengku 135     Indonesia -0.7892
## 133   134   301134           117768      Tengku 136     Indonesia -0.7892
## 134   135   301135           117769      Tengku 137     Indonesia -0.7892
## 135   136   301136           117770      Tengku 138     Indonesia -0.7892
## 136   137   301137           117771      Tengku 139     Indonesia -0.7892
## 137   138   301138           117772      Tengku 140     Indonesia -0.7892
## 138   139   301139           117773      Tengku 141     Indonesia -0.7892
## 139   140   301140           117774      Tengku 142     Indonesia -0.7892
## 140   141   301141           117775      Tengku 143     Indonesia -0.7892
## 141   142   301142           117776      Tengku 144     Indonesia -0.7892
## 142   143   301143           117777      Tengku 145     Indonesia -0.7892
## 143   144   301144           117778      Tengku 146     Indonesia -0.7892
## 144   145   301145           117779      Tengku 147     Indonesia -0.7892
## 145   146   301146           117780      Tengku 148     Indonesia -0.7892
## 146   147   301147           117781      Tengku 149     Indonesia -0.7892
## 147   148   301148           117782      Tengku 150     Indonesia -0.7892
## 148   149   301149           117783      Tengku 151     Indonesia -0.7892
## 149   150   301150           117784      Tengku 152     Indonesia -0.7892
## 150   151   301151           117785      Tengku 153     Indonesia -0.7892
## 151   152   301152           117786      Tengku 154     Indonesia -0.7892
## 152   153   301153           117787      Tengku 155     Indonesia -0.7892
## 153   154   301154           117788      Tengku 156     Indonesia -0.7892
## 154   155   301155           117789      Tengku 157     Indonesia -0.7892
## 155   156   301156           117790      Tengku 158     Indonesia -0.7892
## 156   157   301157           117791      Tengku 159     Indonesia -0.7892
## 157   158   301158           117792      Tengku 160     Indonesia -0.7892
## 158   159   301159           117793      Tengku 161     Indonesia -0.7892
## 159   160   301160           117794      Tengku 162     Indonesia -0.7892
## 160   161   301161           117795      Tengku 163     Indonesia -0.7892
## 161   162   301162           117796      Tengku 164     Indonesia -0.7892
## 162   163   301163           117797      Tengku 165     Indonesia -0.7892
## 163   164   301164           117798      Tengku 166     Indonesia -0.7892
## 164   165   301165           117799      Tengku 167     Indonesia -0.7892
## 165   166   301166           117800      Tengku 168     Indonesia -0.7892
## 166   167   301167           117801      Tengku 169     Indonesia -0.7892
## 167   168   301168           117802      Tengku 170     Indonesia -0.7892
## 168   169   301169           117803      Tengku 171     Indonesia -0.7892
## 169   170   301170           117804      Tengku 172     Indonesia -0.7892
## 170   171   301171           117805      Tengku 173     Indonesia -0.7892
## 171   172   301172           117806      Tengku 174     Indonesia -0.7892
## 172   173   301173           117807      Tengku 175     Indonesia -0.7892
## 173   174   301174           117808      Tengku 176     Indonesia -0.7892
## 174   175   301175           117809      Tengku 177     Indonesia -0.7892
## 175   176   301176           117810      Tengku 178     Indonesia -0.7892
## 176   177   301177           117811      Tengku 179     Indonesia -0.7892
## 177   178   301178           117812      Tengku 180     Indonesia -0.7892
## 178   179   301179           117813      Tengku 181     Indonesia -0.7892
## 179   180   301180           117814      Tengku 182     Indonesia -0.7892
## 180   181   301181           117815      Tengku 183     Indonesia -0.7892
## 181   182   301182           117816      Tengku 184     Indonesia -0.7892
## 182   183   301183           117817      Tengku 185     Indonesia -0.7892
## 183   184   301184           117818      Tengku 186     Indonesia -0.7892
## 184   185   301185           117819      Tengku 187     Indonesia -0.7892
## 185   186   301186           117820      Tengku 188     Indonesia -0.7892
## 186   187   301187           117821      Tengku 189     Indonesia -0.7892
## 187   188   301188           117822      Tengku 190     Indonesia -0.7892
## 188   189   301189           117823      Tengku 191     Indonesia -0.7892
## 189   190   301190           117824      Tengku 192     Indonesia -0.7892
## 190   191   301191           117825      Tengku 193     Indonesia -0.7892
## 191   192   301192           117826      Tengku 194     Indonesia -0.7892
## 192   193   301193           117827      Tengku 195     Indonesia -0.7892
## 193   194   301194           117828      Tengku 196     Indonesia -0.7892
## 194   195   301195           117829      Tengku 197     Indonesia -0.7892
## 195   196   301196           117830      Tengku 198     Indonesia -0.7892
## 196   197   301197           117831      Tengku 199     Indonesia -0.7892
## 197   198   301198           117832      Tengku 200     Indonesia -0.7892
## 198   199   301199           117833      Tengku 201     Indonesia -0.7892
## 199   200   301200           117834      Tengku 202     Indonesia -0.7892
## 200   201   301201           117835      Tengku 203     Indonesia -0.7892
## 201   202   301202           117836      Tengku 204     Indonesia -0.7892
## 202   203   301203           117837      Tengku 205     Indonesia -0.7892
## 203   204   301204           117838      Tengku 206     Indonesia -0.7892
## 204   205   301205           117839      Tengku 207     Indonesia -0.7892
## 205   206   301206           117840      Tengku 208     Indonesia -0.7892
## 206   207   301207           117841      Tengku 209     Indonesia -0.7892
## 207   208   301208           117842      Tengku 210     Indonesia -0.7892
## 208   209   301209           117843      Tengku 211     Indonesia -0.7892
## 209   210   301210           117844      Tengku 212     Indonesia -0.7892
## 210   211   301211           117845      Tengku 213     Indonesia -0.7892
## 211   212   301212           117846      Tengku 214     Indonesia -0.7892
## 212   213   301213           117847      Tengku 215     Indonesia -0.7892
## 213   214   301214           117848      Tengku 216     Indonesia -0.7892
## 214   215   301215           117849      Tengku 217     Indonesia -0.7892
## 215   216   301216           117850      Tengku 218     Indonesia -0.7892
## 216   217   301217           117851      Tengku 219     Indonesia -0.7892
## 217   218   301218           117852      Tengku 220     Indonesia -0.7892
## 218   219   301219           117853      Tengku 221     Indonesia -0.7892
## 219   220   301220           117854      Tengku 222     Indonesia -0.7892
## 220   221   301221           117855      Tengku 223     Indonesia -0.7892
## 221   222   301222           117856      Tengku 224     Indonesia -0.7892
## 222   223   301223           117857      Tengku 225     Indonesia -0.7892
## 223   224   301224           117858      Tengku 226     Indonesia -0.7892
## 224   225   301225           117859      Tengku 227     Indonesia -0.7892
## 225   226   301226           117860      Tengku 228     Indonesia -0.7892
## 226   227   301227           117861      Tengku 229     Indonesia -0.7892
## 227   228   301228           117862      Tengku 230     Indonesia -0.7892
## 228   229   301229           117863      Tengku 231     Indonesia -0.7892
## 229   230   301230           117864      Tengku 232     Indonesia -0.7892
## 230   231   301231           117865      Tengku 233     Indonesia -0.7892
## 231   232   301232           117866      Tengku 234     Indonesia -0.7892
## 232   233   301233           117867      Tengku 235     Indonesia -0.7892
## 233   234   301234           117868      Tengku 236     Indonesia -0.7892
## 234   235   301235           117869      Tengku 237     Indonesia -0.7892
## 235   236   301236           117870      Tengku 238     Indonesia -0.7892
## 236   237   301237           117871      Tengku 239     Indonesia -0.7892
## 237   238   301238           117872      Tengku 240     Indonesia -0.7892
## 238   239   301239           117873      Tengku 241     Indonesia -0.7892
## 239   240   301240           117874      Tengku 242     Indonesia -0.7892
## 240   241   301241           117875      Tengku 243     Indonesia -0.7892
## 241   242   301242           117876      Tengku 244     Indonesia -0.7892
## 242   243   301243           117877      Tengku 245     Indonesia -0.7892
## 243   244   301244           117878      Tengku 246     Indonesia -0.7892
## 244   245   301245           117879      Tengku 247     Indonesia -0.7892
## 245   246   301246           117880      Tengku 248     Indonesia -0.7892
## 246   247   301247          
```

(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。

```
# extract variables for regression analysis
data <- data.frame(
  height = line.pheno$Plant.height,
  flower = line.pheno$Flowering.time.at.Arkansas,
  PC1 = line.pheno$PC1,
  PC2 = line.pheno$PC2,
  PC3 = line.pheno$PC3,
  PC4 = line.pheno$PC4)
data <- na.omit(data)
head(data)

##      height     flower      PC1      PC2      PC3      PC4
## 1 110.9167 75.08333 -0.0486  0.0030  0.0752 -0.0076
## 2 143.5000 89.50000  0.0672 -0.0733  0.0094 -0.0005
## 3 128.0833 94.50000  0.0544  0.0681 -0.0062 -0.0369
## 4 153.7500 87.50000 -0.0073  0.0224 -0.0121  0.2602
## 5 148.3333 89.08333  0.0509  0.0655 -0.0058 -0.0378
## 6 119.6000 105.00000 -0.0293 -0.0027 -0.0677 -0.0085
```

まずは、両者の関係を図示します。

```
# Look at the relationship between plant height and flowering time
plot(data$height ~ data$flower)
```

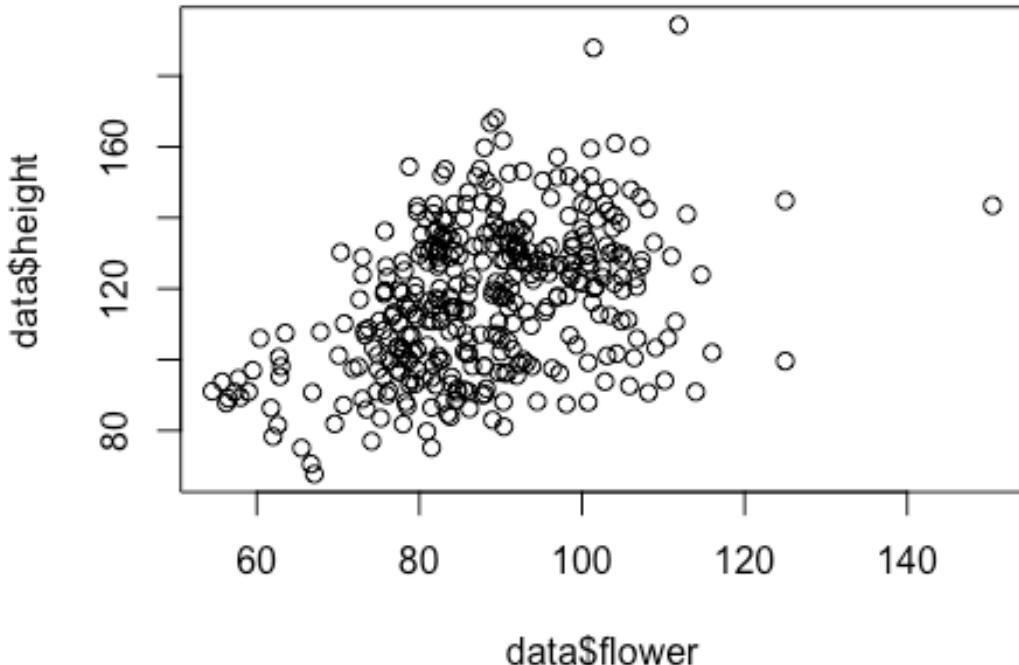


図 1 にも示されているように、開花が早いものほど草丈が小さく、遅くなるほど草丈が大きくなる傾向が見てとれます。

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。

```
# perform single Linear regression
model <- lm(height ~ flower, data = data)
```

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。

```
# show the result
summary(model)

##
## Call:
## lm(formula = height ~ flower, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -43.846 -13.718    0.295  13.409   61.594 
## 
## 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
```

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と思われます。しかし、後で述べるように複数の回帰モデルを作つ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、

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

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。

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るが、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

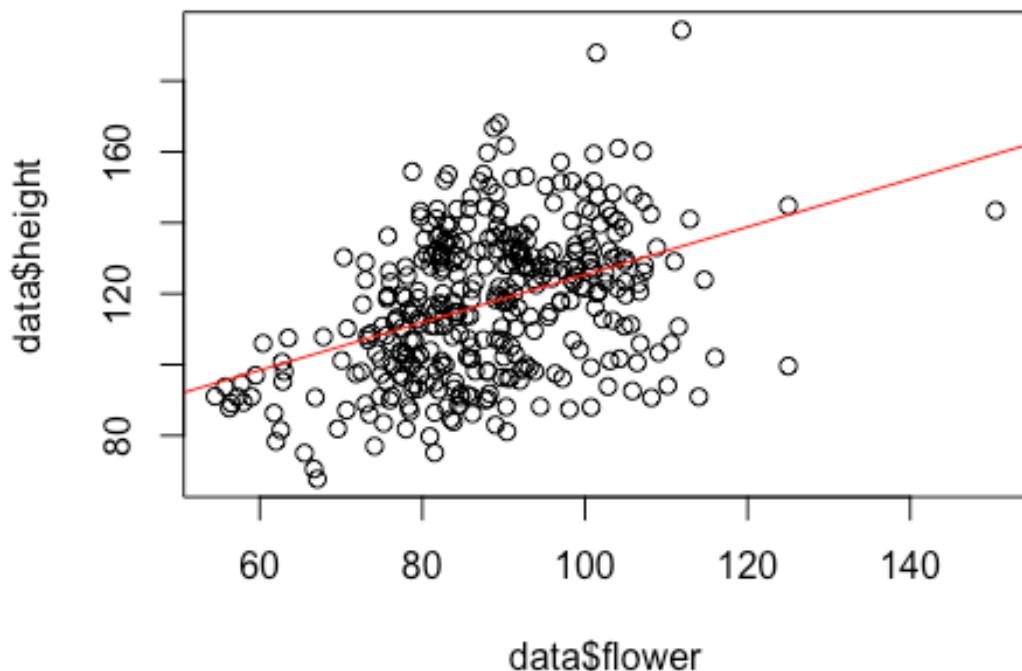
回帰モデルのパラメータ μ 、 β の推定値 m 、 b と、それに伴う標準誤差、 t 値、 p 値が表示されています。また、各行の最後の星印は、有意水準を視覚的に確認しやすくしたもので
す。1つ星は5%、2つ星は1%、3つ星は0.1%水準で有意であることを表しています。

```
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。全ての回帰係数が0で
あるという仮説（帰無仮説）のもとの検定であり、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。

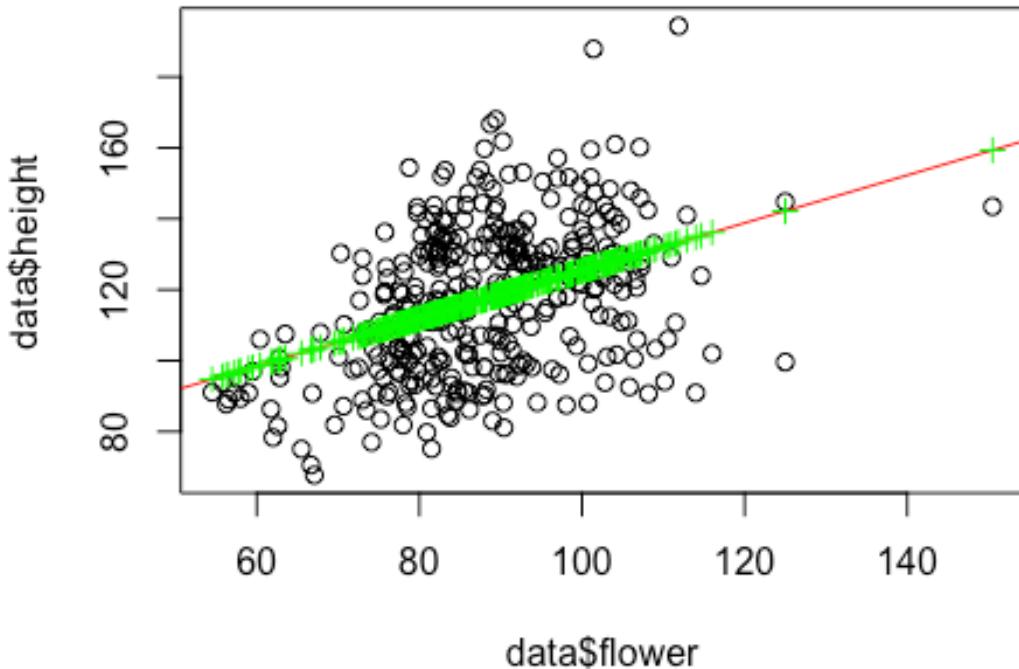
では、回帰分析の結果を図示して眺めてみましょう。まず、散布図を描き、そこに回帰直
線を引きます。

```
# again, plot the two variables
plot(data$height ~ data$flower)
abline(model, col = "red")
```



次に、回帰モデルにデータをあてはめたときの y の値を計算し、図示してみます。

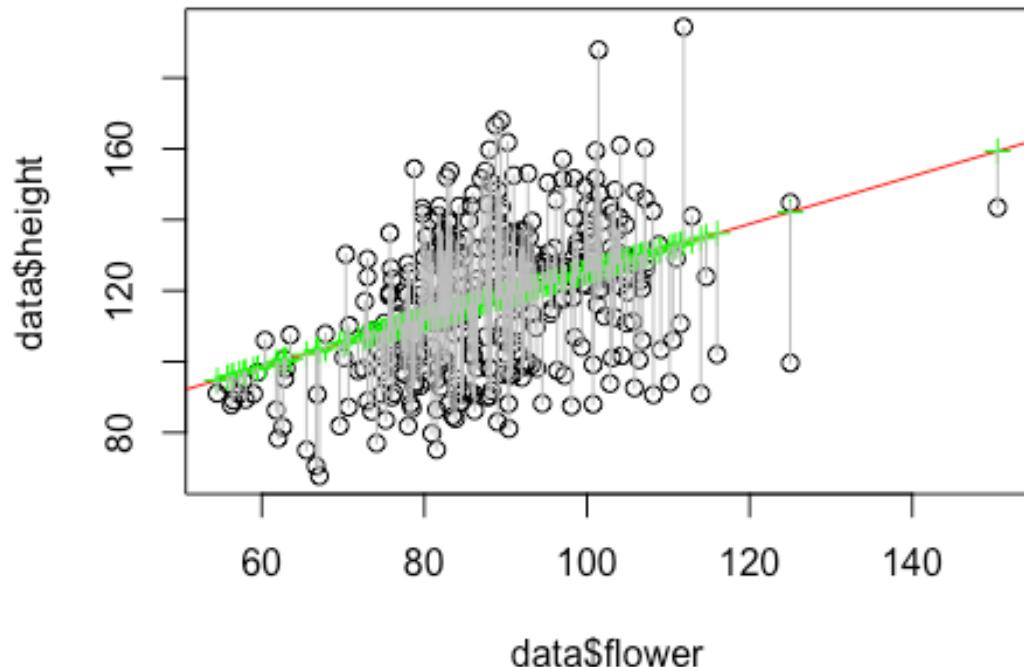
```
# calculate fitted values
height.fit <- fitted(model)
plot(data$height ~ data$flower)
abline(model, col = "red")
points(data$flower, height.fit, pch = 3, col = "green")
```



モデルをあてはめて計算される y の値は全て直線上に乘ります。

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。

```
# plot residuals
plot(data$height ~ data$flower)
abline(model, col = "red")
points(data$flower, height.fit, pch = 3, col = "green")
segments(data$flower, height.fit,
          data$flower, height.fit + resid(model), col = "gray")
```

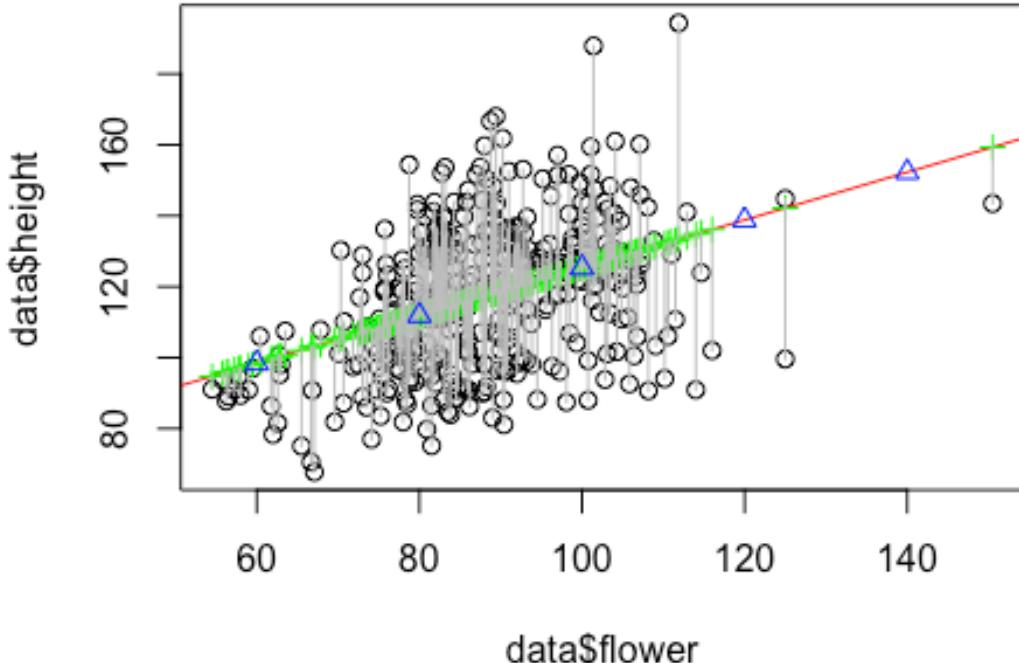


y の値は、モデルをあてはめて計算される y の値（緑色の点）とモデルの残差（灰色の線）の和として表されます。

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

```
# predict unknown data
height.pred <- predict(model, data.frame(flower = seq(60, 140, 20)))

plot(data$height ~ data$flower)
abline(model, col = "red")
points(data$flower, height.fit, pch = 3, col = "green")
segments(data$flower, height.fit,
          data$flower, height.fit + resid(model), col = "gray")
points(seq(60, 140, 20), height.pred, pch = 2, col = "blue")
```



やはり、予測値は全て回帰直線の上に乘ります。

Quiz 1

では、ここで練習問題を解いてみましょう。練習問題は、授業中に提示します。

<https://www.menti.com/ie9s1dzmdz> に移動して、授業中に指定する番号を入力して下さい。その後、ニックネームを登録してクイズが始まるまで待機していて下さい。

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。

先述したように单回帰のモデルは、

$$y_i = m + bx_i + \epsilon_i$$

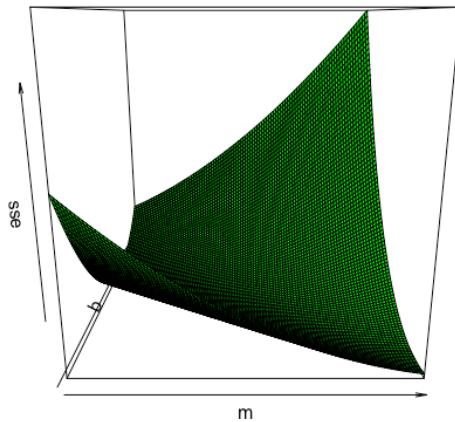
として表現されます。この式は、観察値 y_i が、回帰方程式で説明される部分 $m + bx_i$ と、回帰直線では説明されない誤差部分 ϵ_i から成ることを意味しています。上式の、 m や b を動かすと、それに伴って誤差 ϵ_i も変化します。では、どのようにして“最適な”パラメータを求めればよいのでしょうか。何をもって“最適”とするかについては様々な基準が考えられます。ここでは、誤差 ϵ_i をデータ全体で最小にすることを考えてみます。は正負両方の値をとりますので、単純に和をとると互いに相殺されてしまいます。そこで、 ϵ_i の 2 乗和 (sum of squared error: SSE) を最小にすることを考えます。すなわち、

$$SSE = \sum_{i=1}^n \epsilon_i^2 = \sum_{i=1}^n (y_i - (m + bx_i))^2$$

式(1) を最小にするような m と b を考えてみましょう。

以下の図は様々な m と b に対する SSE の変化を表した図です。この図を描くためのコマンドは少し複雑ですが次のようにになります。

```
#visualize the plane for optimization
x <- data$flower
y <- data$height
m <- seq(0, 100, 1)
b <- seq(0, 2, 0.02)
sse <- matrix(NA, length(m), length(b))
for(i in 1:length(m)) {
  for(j in 1:length(b)) {
    sse[i, j] <- sum((y - m[i] - b[j] * x)^2)
  }
}
persp(m, b, sse, col = "green")
```



パッケージ“plotly”を使うと、

```
# draw the figure with plotly
df <- data.frame(m, b, sse)
plot_ly(data = df, x = ~m, y = ~b, z = ~sse) %>% add_surface()
```

なお、上図において SSE が最小となる点では、 m や b 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 m および b 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、

$$\frac{\partial SSE}{\partial m} = 0, \frac{\partial SSE}{\partial b} = 0$$

としてこれを満たす m および b 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) と呼びます。

なお、SSE を最小化する m は、

$$\begin{aligned}\frac{\partial SSE}{\partial m} &= -2 \sum_{i=1}^n (y_i - m - bx_i) = 0 \\ \Leftrightarrow \sum_{i=1}^n y_i - nm - b \sum_{i=1}^n x_i &= 0 \\ \Leftrightarrow m &= \frac{\sum_{i=1}^n y_i}{n} - b \frac{\sum_{i=1}^n x_i}{n} = \bar{y} - b\bar{x}\end{aligned}$$

として計算されます。

また、SSE を最小化する b は、

$$\begin{aligned}\frac{\partial SSE}{\partial b} &= -2 \sum_{i=1}^n x_i (y_i - m - bx_i) = 0 \\ \Leftrightarrow \sum_{i=1}^n x_i y_i - m \sum_{i=1}^n x_i - b \sum_{i=1}^n x_i^2 &= 0 \\ \Leftrightarrow \sum_{i=1}^n x_i y_i - n(\bar{y} - b\bar{x})\bar{x} - b \sum_{i=1}^n x_i^2 &= 0 \\ \Leftrightarrow \sum_{i=1}^n x_i y_i - n\bar{x}\bar{y} - b(\sum_{i=1}^n x_i^2 - n\bar{x}^2) &= 0 \\ \Leftrightarrow b &= \frac{\sum_{i=1}^n x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^n x_i^2 - n\bar{x}^2} = \frac{SSXY}{SSX}\end{aligned}$$

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、

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

$$\begin{aligned}
&= \sum_{i=1}^n x_i y_i - n \bar{x} \bar{y} \\
SSX &= \sum_{i=1}^n (x_i - \bar{x})^2 \\
&= \sum_{i=1}^n x_i^2 - 2\bar{x} \sum_{i=1}^n x_i - n \bar{x}^2 \\
&= \sum_{i=1}^n x_i^2 - 2n \bar{x}^2 + n \bar{x}^2 \\
&= \sum_{i=1}^n x_i^2 - n \bar{x}^2
\end{aligned}$$

として計算されます。

SSE を最少にする m と b をこれらパラメータの推定値とし、以降、単に、 m と b で表すことになります。すなわち、

$$b = \frac{SSXY}{SSX}$$

$$m = \bar{y} - b \bar{x}$$

では、回帰係数を上述した式をもとに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。

```
# calculate sum of squares (ss) of x and ss of xy
n <- length(x)
ssx <- sum(x^2) - n * mean(x)^2
ssxy <- sum(x * y) - n * mean(x) * mean(y)
```

まずは傾き b を計算します。

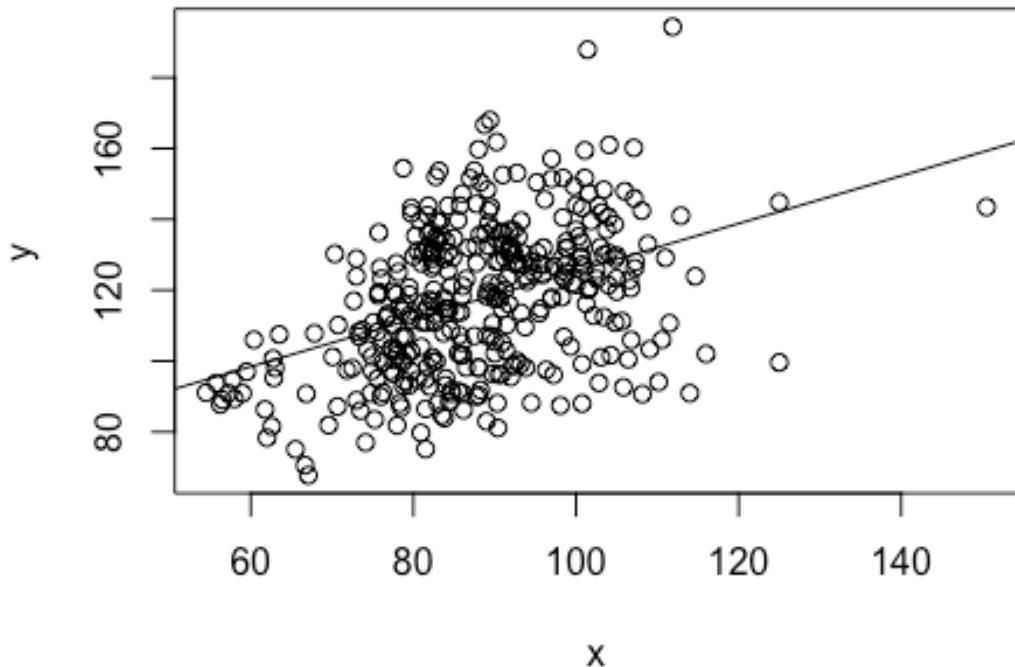
```
# calculate b
b <- ssxy / ssx
b
## [1] 0.6728746
```

次に切片 m を計算します。

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

計算された b と m をもとに回帰直線を描いてみましょう。

```
# draw scatter plot and regression line
plot(y ~ x)
abline(m, b)
```



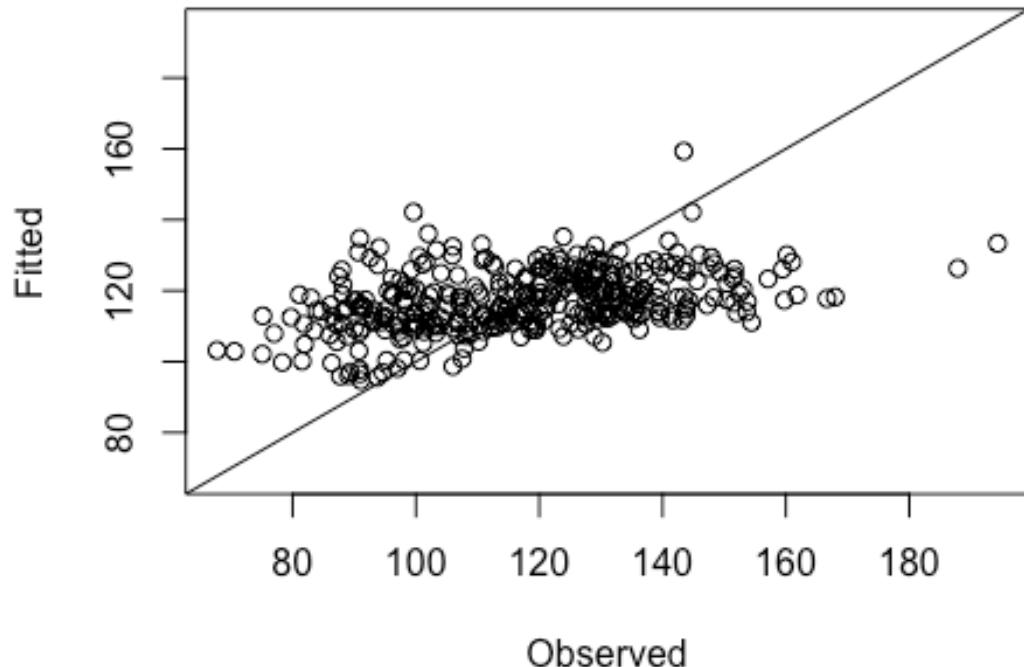
先ほど関数 `lm` を用いて計算された回帰直線と同じものが描かれていることを確認してみましょう。

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。

```
# calculate fitted values
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)
```



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。

```
# calculate correlation between observed and fitted values
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。

```
# compare the square of the correlation and R2
cor(y, y.hat)^2
## [1] 0.1671894
summary(model)

##
## Call:
## lm(formula = height ~ flower, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -43.846 -13.718   0.295  13.409  61.594 
## 
## 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

```

Quiz 2

では、ここで練習問題を解いてみましょう。練習問題は、授業中に提示します。

クイズのページを閉じてしまった人は、<https://www.menti.com/ie9s1dzmdz> に移動して、授業中に指定する番号を入力して下さい。

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。

```

# analysis of variance of regression
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。

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

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

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、

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

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。

```
# calculate sum of squares of regression and error
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。

```
# calculate mean squares of regression and error
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。

```
# calculate F value
f.value <- msr / mse
f.value

## [1] 74.47943

# calculate p value for the F value
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。

```
# check the summary of the result of regression analysis
summary(model)

##
## Call:
## lm(formula = height ~ flower, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -43.846 -13.718    0.295  13.409   61.594
##
## 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。

```
# square root of mse
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。

```
# R squared
ssr / ssy

## [1] 0.1671894
```

「Adjusted R-squared」 (R_{adj}^2) 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。

```
# adjusted R squared
(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 ($8.6302^2 = 74.477$)。

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

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

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

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_{adj}^2 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。

Quiz 3

では、ここで練習問題を解いてみましょう。練習問題は、授業中に提示します。

クイズのページを閉じてしまった人は、<https://www.menti.com/ie9s1dzmdz> に移動して、授業中に指定する番号を入力して下さい。

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 m と b 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。

ここでは詳細は省きますが、推定値 b は以下の正規分布に従います。

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

なお、ここで、 σ^2 は、誤差分散 $\sigma^2 = Var(y_i) = Var(e_i)$ です。

いっぽう、推定値 m は、以下の正規分布に従います。

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

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

$$t = \frac{b - b_0}{s / \sqrt{SSX}}$$

は、帰無仮説

$$H_0: \beta = b_0$$

のもとで、自由度 $n - 2$ の t 分布に従います。

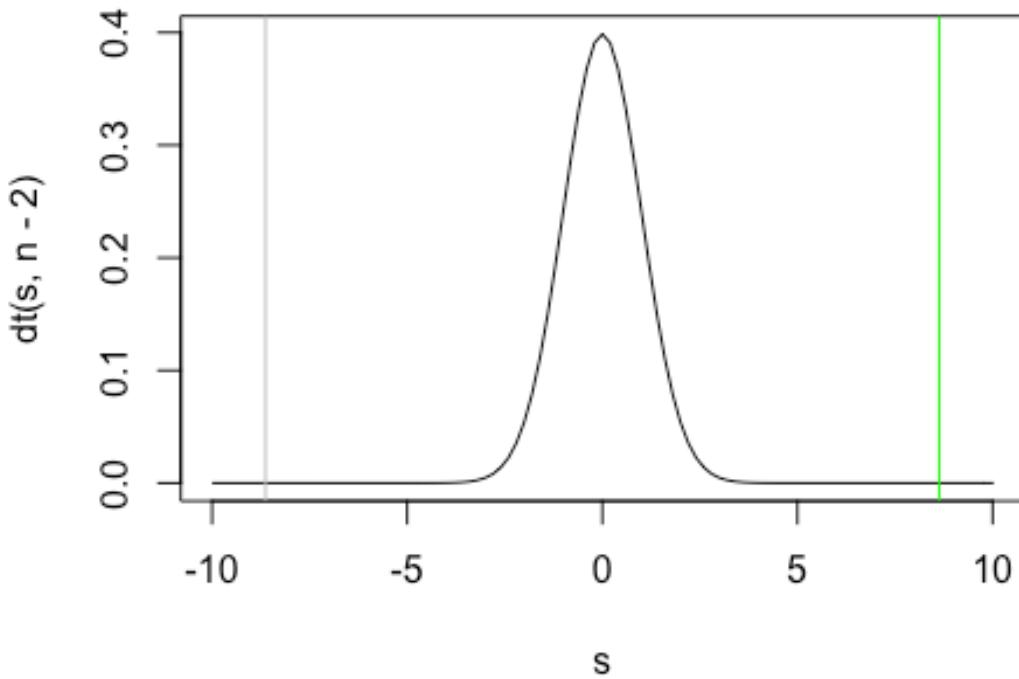
まず b について、帰無仮説 $H_0: \beta = 0$ について検定してみます。このとき、 $b_0 = 0$ であることに注意して、

```
# test beta = 0
t.value <- (b - 0) / sqrt(mse/ssx)
t.value

## [1] 8.630147
```

この統計量は、帰無仮説のもとで、自由度 371 ($n - 2$) の t 分布にしたがいます。図で描いてみると

```
# visualize the t distribution under H0
s <- seq(-10, 10, 0.2)
plot(s, dt(s, n - 2), type = "l")
abline(v = t.value, col = "green")
abline(v = -t.value, col = "gray")
```



帰無仮説のもとで従う分布からみると、得られてる t の値が大きいように見えます。実際に t 検定を行ってみます。両側検定であることに注意して、

```
# perform t test
2 * (1 - pt(abs(t.value), n - 2)) # two-sided test
## [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。

```
# check the summary of the model
summary(model)

##
## Call:
## lm(formula = height ~ flower, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -43.846  -13.718    0.295  13.409   61.594
##
## 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
```

上で行った仮説検定は、任意の b_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。

```
# test beta = 0.5  
t.value <- (b - 0.5) / sqrt(msse/ssx)  
t.value  
  
## [1] 2.217253  
  
2 * (1 - pt(abs(t.value), n - 2))  
## [1] 0.02721132
```

結果は、5%水準で有意です。

では、 m についても検定を行ってみましょう。まず、帰無仮説 $H_0: \mu = 0$ の検定を行ってみましょう。

```
# test mu = 0  
t.value <- (m - 0) / sqrt(msse * (1/n + mean(x)^2 / ssx))  
t.value  
  
## [1] 8.383389  
  
2 * (1 - pt(abs(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。

```
# check the summary of the model again  
summary(model)  
  
##  
## Call:  
## lm(formula = height ~ flower, data = data)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -43.846 -13.718    0.295  13.409   61.594  
##  
## 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
```

(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 $H_0: \mu = 70$ について検定してみましょう。

```
# test mu = 70
t.value <- (m - 70) / sqrt(msse * (1/n + mean(x)^2 / ssx))
t.value

## [1] -1.724971
2 * (1 - pt(abs(t.value), n - 2))

## [1] 0.08536545
```

結果は、5%水準でも有意ではありませんでした。

Quiz 4

では、ここで練習問題を解いてみましょう。練習問題は、授業中に提示します。

クイズのページを閉じてしまった人は、<https://www.menti.com/ie9s1dzmdz> に移動して、授業中に指定する番号を入力して下さい。

推定値や予測値の信頼区間

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 の推定値 \hat{y} が計算されます。その推定値は関数 `fitted` で計算されるものと全く同じです。

```
# fitted values
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
```

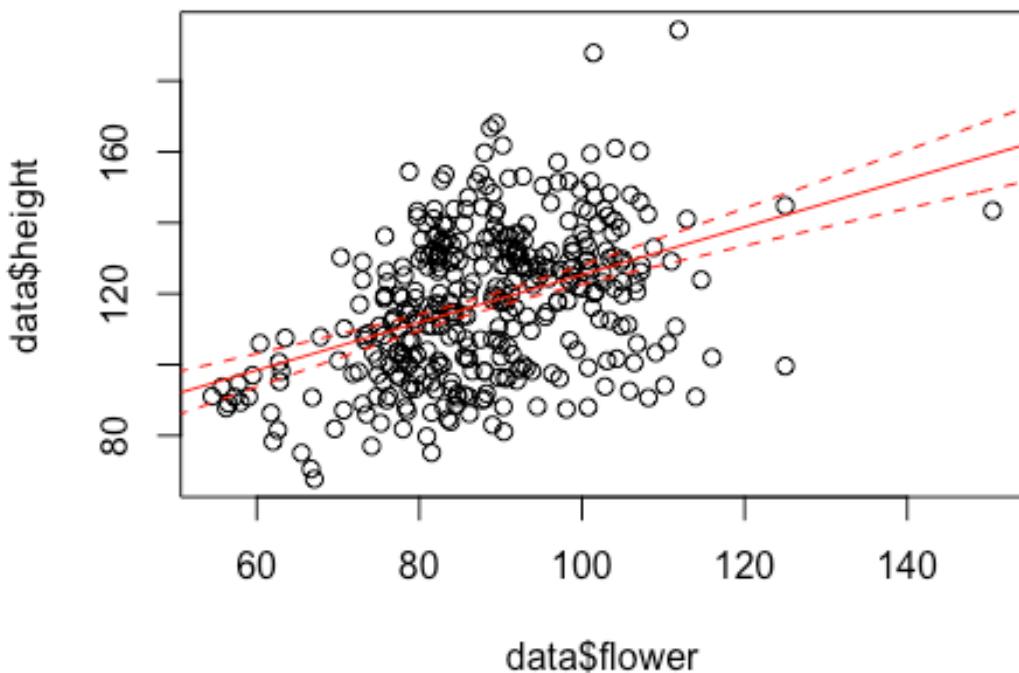
オプション `interval` と `level` を設定すると、モデルをあてはめたときの y の推定値 \hat{y} の信頼区間（デフォルト設定は 95% 信頼区間）を計算できます。

```
# calculate confidence interval
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
```

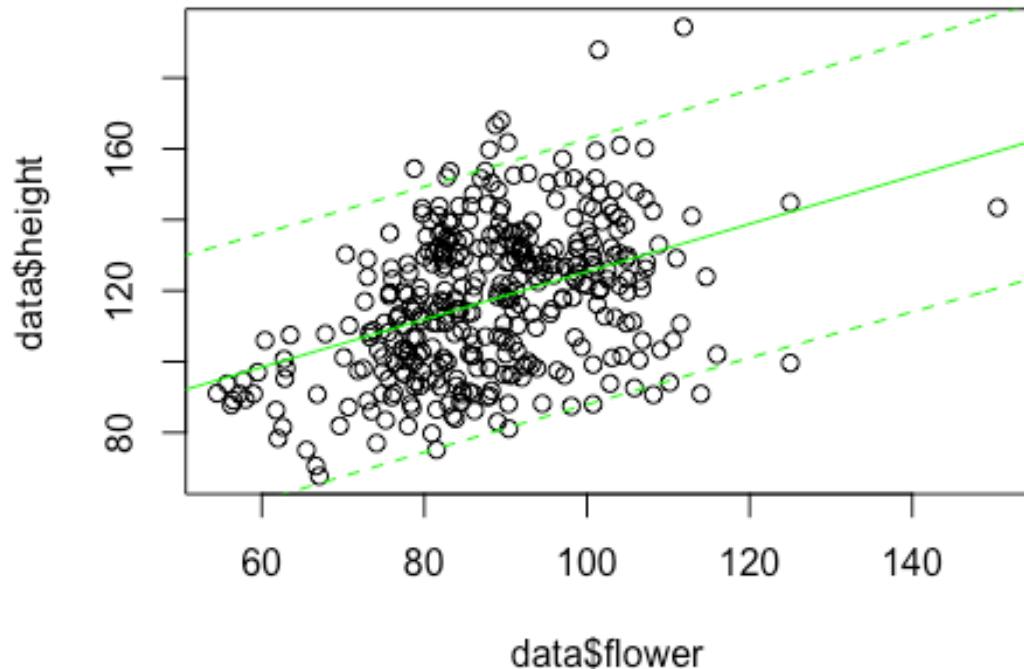
関数 `predict` を用いて y の推定値 \hat{y} の 95% 信頼区間を図示してみましょう。

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



次に、未知のデータを予測する場合について考えます。未知データについて観察されるであろう y の値、すなわち、予測値 \hat{y} には、誤差によるばらつきがさらに加わります。`predict`関数を用いて、予測値の95%信頼区間を図示してみましょう。

```
pc <- predict(model, interval= "prediction", newdata = pred)
plot(data$height ~ data$flower)
matlines(pred$flower, pc, lty = c(1, 2, 2), col = "green")
```



誤差が加わる分、予測値 \hat{y} のばらつきは推定値 \hat{y} に比べて大きくなります。

なお、ある特定の x についてだけ、推定値 \hat{y} の信頼区間と、予測値 \hat{y} の信頼区間を求めるには以下のようにします。ここでは、 $x = 120$ のときの99%信頼区間を求めてみます。

```
# estimate the confidence intervals for the estimate and prediction of y
pred <- data.frame(flower = 120)
predict(model, interval = "confidence", newdata = pred, level = 0.99)

##          fit      lwr      upr
## 1 138.7996 131.8403 145.7589

predict(model, interval = "prediction", newdata = pred, level = 0.99)

##          fit      lwr      upr
## 1 138.7996 89.12106 188.4781
```

Quiz 5

では、ここで練習問題を解いてみましょう。練習問題は、授業中に提示します。

クイズのページを閉じてしまった人は、<https://www.menti.com/ie9s1dzmdz> に移動して、授業中に指定する番号を入力して下さい。

多項式回帰モデルと重回帰モデル

ここまででは、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、

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

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。

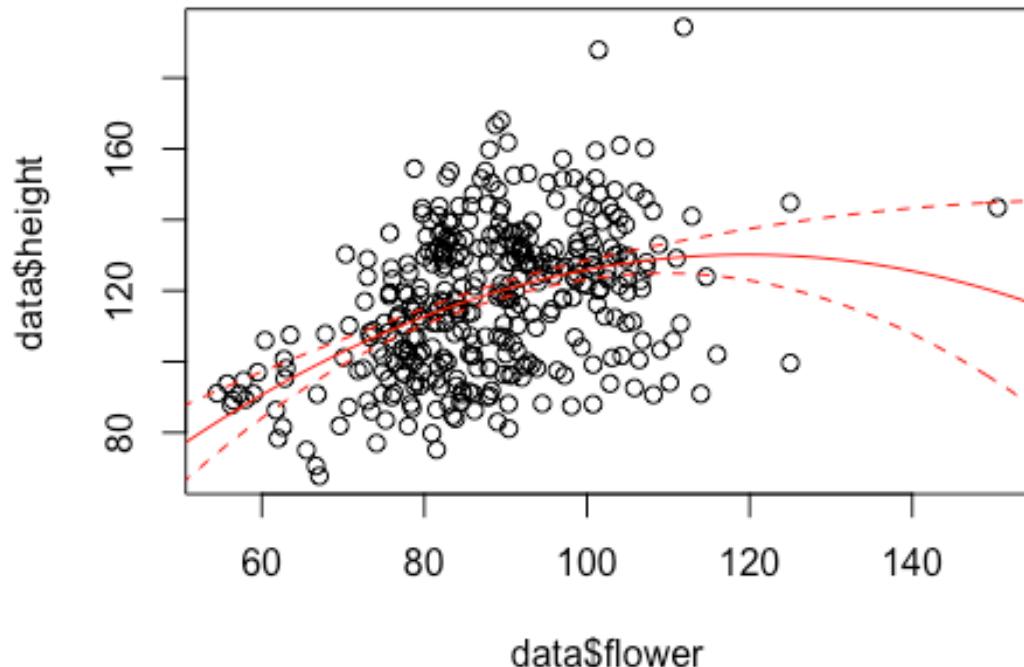
```
# polynomial regression
model.quad <- lm(height ~ flower + I(flower^2), data = data)
summary(model.quad)

##
## Call:
## lm(formula = height ~ flower + I(flower^2), data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max 
## -39.57 -13.60   0.97  12.91  64.83 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -29.082326  27.019440  -1.076 0.282473  
## flower       2.662663   0.601878   4.424 1.28e-05 *** 
## I(flower^2) -0.011130   0.003339  -3.333 0.000945 *** 
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 18.74 on 370 degrees of freedom
## Multiple R-squared:  0.1915, Adjusted R-squared:  0.1871 
## F-statistic: 43.81 on 2 and 370 DF,  p-value: < 2.2e-16
```

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当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。

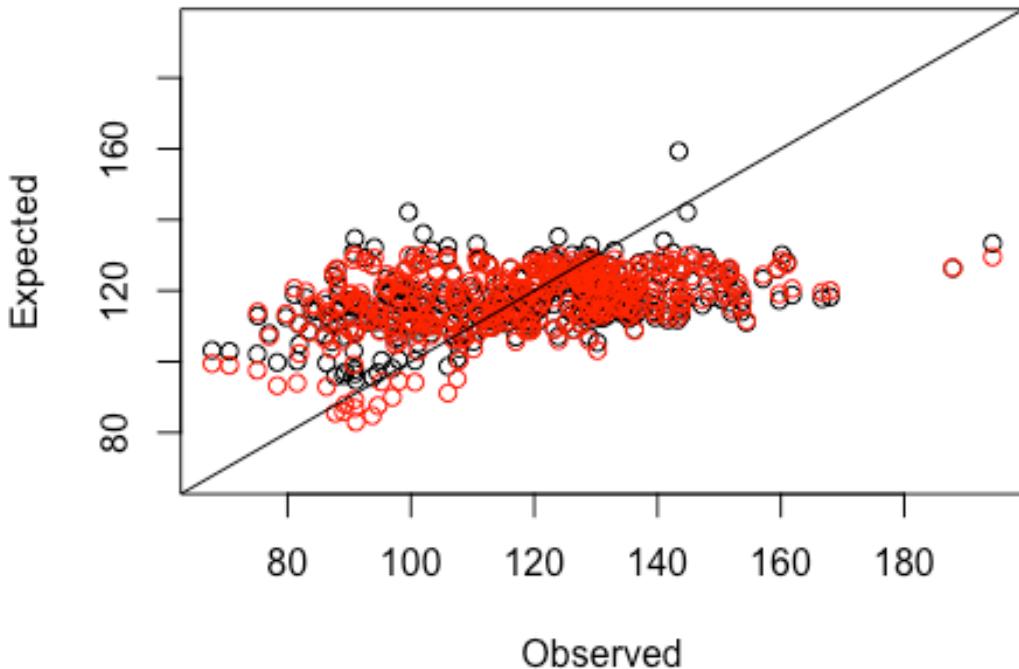
```
# plot(data$height ~ data$flower)
pred <- data.frame(flower = 50:160)
pc <- predict(model.quad, interval = "confidence", newdata = pred)
plot(data$height ~ data$flower)
matlines(pred$flower, pc, lty = c(1, 2, 2), col = "red")
```



多項式回帰では、開花のタイミングが播種後 120 日以上の場合には推定値の信頼性が低いことが分かります。

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。

```
# compare predicted and observed values
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)
```



上図は、単回帰モデル（黒）および2次の多項式モデル（赤）における推定値と観察値の関係を表しています。

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。

```
# compare error variance between two models
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。

```

# extend polynomial regression model to a higher dimensional one...
model.cube <- lm(height ~ flower + I(flower^2) + I(flower^3), data = data)
summary(model.cube)

##
## Call:
## lm(formula = height ~ flower + I(flower^2) + I(flower^3), data = data)
##
## Residuals:
##    Min      1Q  Median      3Q     Max 
## -39.699 -13.705   1.031  13.240  65.840 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.001e+02  8.541e+01  -1.172  0.2419    
## flower       5.029e+00  2.765e+00   1.818  0.0698 .  
## I(flower^2) -3.664e-02  2.929e-02  -1.251  0.2118    
## I(flower^3)  8.898e-05  1.015e-04   0.877  0.3813    
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
##
## Residual standard error: 18.75 on 369 degrees of freedom
## Multiple R-squared:  0.1931, Adjusted R-squared:  0.1866 
## F-statistic: 29.44 on 3 and 369 DF, p-value: < 2.2e-16

# compare error variance between two models
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、

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

というかたちで複数の説明変数 ($x_{1i}, x_{2i}, \dots, x_{pi}$) 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。

```

# multi-linear regression with genetic background
model.wgb <- lm(height ~ PC1 + PC2 + PC3 + PC4, data = data)
summary(model.wgb)

##
## Call:
## lm(formula = height ~ PC1 + PC2 + PC3 + PC4, data = data)
## 
```

```

## lm(formula = height ~ PC1 + PC2 + PC3 + PC4, data = data)
##
## Residuals:
##   Min     1Q Median     3Q    Max
## -45.89 -11.65   0.15  11.05  72.12
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 117.2608    0.8811 133.080 < 2e-16 ***
## PC1         181.6572   18.2977   9.928 < 2e-16 ***
## PC2         83.5334   17.9920   4.643 4.79e-06 ***
## PC3        -88.6432   18.1473  -4.885 1.55e-06 ***
## PC4         122.1351   18.2476   6.693 8.16e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17 on 368 degrees of freedom
## Multiple R-squared:  0.3388, Adjusted R-squared:  0.3316
## F-statistic: 47.14 on 4 and 368 DF, p-value: < 2.2e-16

anova(model.wgb)

## Analysis of Variance Table
##
## 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。

```

# multi-linear regression with all information
model.all <- lm(height ~ flower + I(flower^2) + PC1 + PC2 + PC3 + PC4, data =
  data)
summary(model.all)

##
## Call:
## lm(formula = height ~ flower + I(flower^2) + PC1 + PC2 + PC3 +
##     PC4, data = data)
##
## Residuals:
##   Min     1Q Median     3Q    Max
## -37.589 -10.431   0.542  10.326  65.390
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)

```

```

## (Intercept) 25.739160 24.725955 1.041 0.29857
## flower      1.633185 0.543172 3.007 0.00282 **
## I(flower^2) -0.006598 0.002974 -2.219 0.02713 *
## PC1        141.214491 18.547296 7.614 2.29e-13 ***
## PC2        83.552448 17.231568 4.849 1.84e-06 ***
## PC3       -45.310663 18.647979 -2.430 0.01559 *
## PC4       119.638954 17.369423 6.888 2.48e-11 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.17 on 366 degrees of freedom
## Multiple R-squared: 0.4045, Adjusted R-squared: 0.3947
## F-statistic: 41.43 on 6 and 366 DF, p-value: < 2.2e-16

# compare error variance
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

```

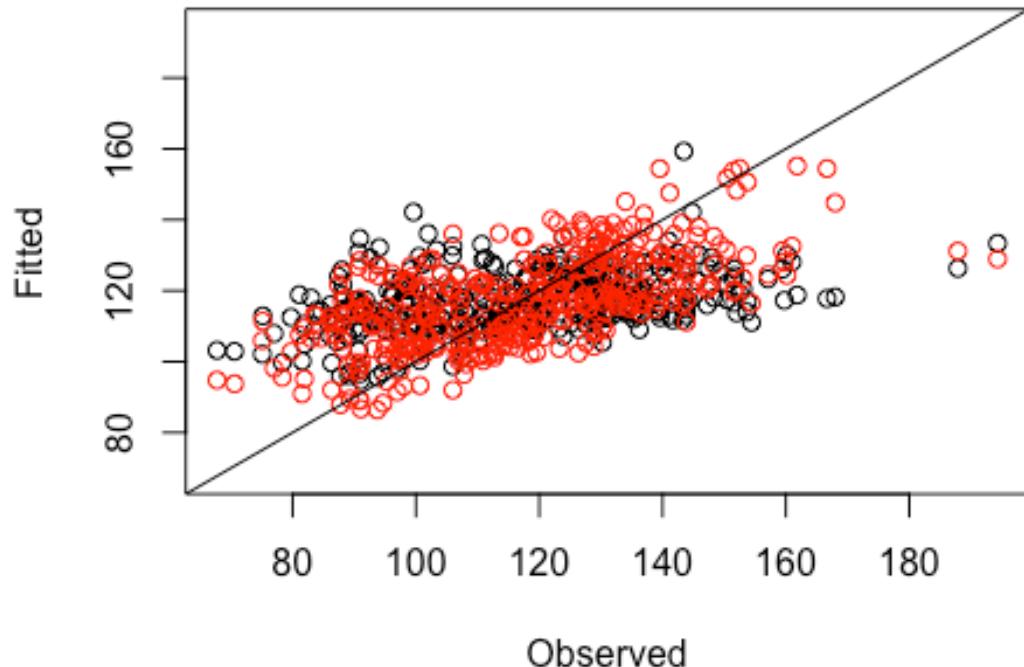
草丈に対する遺伝的背景の効果は非常に大きいのですが、それだけでなく、開花のタイミングの効果についても加えたほうが、モデルの説明力が向上することが分かります。

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、観察値と推定値の対散布を描いて比較してみましょう。

```

# compare between the simplest and final models
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)

```



x の値が大きいときのあてはまりの悪さは、解決していないことが分かります。

Quiz 6

では、ここで練習問題を解いてみましょう。練習問題は、授業中に提示します。

クイズのページを閉じてしまった人は、<https://www.menti.com/ie9s1dzmdz> に移動して、授業中に指定する番号を入力して下さい。

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) と呼びます。

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 a seed for random number generation
set.seed(12)
```

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。

```
# The blocks have unequal fertility among them
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。

```
# set block to consider the heterogeneity of field condition
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。

```
# assume that there are four varieties
variety <- c("A", "B", "C", "D")
# sample the order of the four varieties randomly
sample(variety)

## [1] "B" "D" "C" "A"

sample(variety)

## [1] "C" "B" "A" "D"
```

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。

```
# allocate the varieties randomly to each column of the field
varmat <- matrix(c(sample(variety), sample(variety),
                     sample(variety), sample(variety)), nrow = 4)
varmat

##      [,1] [,2] [,3] [,4]
## [1,] "D"  "B"  "B"  "D"
## [2,] "B"  "A"  "D"  "A"
## [3,] "C"  "D"  "C"  "B"
## [4,] "A"  "C"  "A"  "C"
```

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。

```
# simulate genetic ability of the varieties
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,]    -4     2     2    -4
## [2,]     2     4    -4     4
## [3,]    -2    -4    -2     2
## [4,]     4    -2     4    -2
```

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。

```
# simulate error variance (variation due to the heterogeneity of Local environment)
e.value <- matrix(rnorm(16, sd = 2.5), 4, 4)
e.value

##      [,1]      [,2]      [,3]      [,4]
## [1,] -1.547611  2.232424  0.1911757  1.8861892
## [2,] -2.207789  3.909922 -2.1251164 -0.8860432
## [3,]  1.098536  2.524477 -3.2760349 -1.1552031
## [4,]  3.110199 -0.690938 -4.2153578  4.7493152
```

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。

```
# simulate phenotypic values
grand.mean <- 50
simyield <- grand.mean + field.cond + g.value + e.value
simyield

##      [,1]     [,2]     [,3]     [,4]
## [1,] 48.45239 56.23242 50.19118 43.88619
## [2,] 53.79221 59.90992 41.87488 49.11396
## [3,] 53.09854 50.52448 42.72397 46.84480
## [4,] 61.11020 49.30906 47.78464 48.74932
```

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。

```
# unfold a matrix to a vector
as.vector(simyield)

## [1] 48.45239 53.79221 53.09854 61.11020 56.23242 59.90992 50.52448 49.309
06
## [9] 50.19118 41.87488 42.72397 47.78464 43.88619 49.11396 46.84480 48.749
32

as.vector(varmat)

## [1] "D" "B" "C" "A" "B" "A" "D" "C" "B" "D" "C" "A" "D" "A" "B" "C"

as.vector(blomat)

## [1] "I"   "I"   "I"   "I"   "II"  "II"  "II"  "III" "III" "III" "II
I"
## [13] "IV"  "IV"  "IV"  "IV"
```

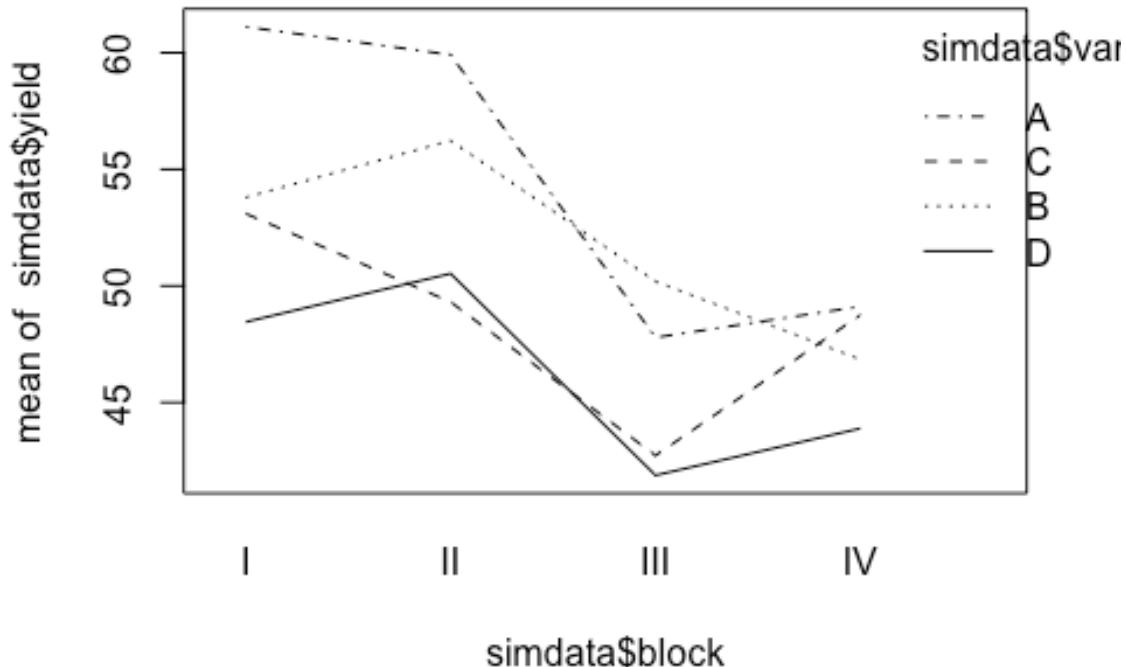
以下、データフレームとしてデータを束ねています。

```
# create a dataframe for the analysis of variance
simdata <- data.frame(variety = as.vector(varmat), block = as.vector(blomat),
yield = as.vector(simyield))
head(simdata, 10)

##   variety block   yield
## 1       D    I 48.45239
## 2       B    I 53.79221
## 3       C    I 53.09854
## 4       A    I 61.11020
## 5       B   II 56.23242
## 6       A   II 59.90992
## 7       D   II 50.52448
## 8       C   II 49.30906
## 9       B   III 50.19118
## 10      D   III 41.87488
```

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

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



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。

```
# perform the analysis of variance (ANOVA) with simulated data
res <- aov(yield ~ block + variety, data = simdata)
summary(res)

##           Df Sum Sq Mean Sq F value    Pr(>F)
## block      3 239.11   79.70   11.614 0.00190 ***
## variety     3 159.52   53.17    7.748 0.00728 **
## Residuals  9  61.77    6.86
## ---
## 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。

```
# perform ANOVA with a Linear model
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      3 239.109  79.703 11.6138 0.001898 **
## variety     3 159.518  53.173  7.7479 0.007285 **
## Residuals  9   61.765   6.863
## ---
## Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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) 」と呼びます。

```
# completely randomized the plots of each variety in a field
varmat.crd <- matrix(sample(varmat), nrow = 4)
varmat.crd

##      [,1] [,2] [,3] [,4]
## [1,] "A"  "C"  "C"  "D"
## [2,] "B"  "A"  "B"  "D"
## [3,] "A"  "C"  "B"  "A"
## [4,] "D"  "B"  "C"  "D"
```

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。

```
# simulate genetic ability of the varieties
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   -2   -2   -4
## [2,]    2    4    2   -4
## [3,]    4   -2    2    4
## [4,]   -4    2   -2   -4
```

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。

```
# simulate phenotypic values
simyield.crd <- grand.mean + g.value.crd + field.cond + e.value
simyield.crd

##      [,1]     [,2]     [,3]     [,4]
## [1,] 56.45239 52.23242 46.19118 43.88619
## [2,] 53.79221 59.90992 47.87488 41.11396
## [3,] 59.09854 52.52448 46.72397 48.84480
## [4,] 53.11020 53.30906 41.78464 46.74932
```

データフレームとしてデータを束ねます。

```
# create a dataframe for the analysis of variance
simdata.crd <- data.frame(variety = as.vector(varmat.crd),
                           yield = as.vector(simyield.crd))
head(simdata.crd, 10)

##   variety   yield
## 1       A 56.45239
## 2       B 53.79221
## 3       A 59.09854
## 4       D 53.11020
## 5       C 52.23242
## 6       A 59.90992
## 7       C 52.52448
## 8       B 53.30906
## 9       C 46.19118
## 10      B 47.87488
```

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。

```
# perform ANOVA
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      3 218.12  72.705  3.1663 0.06392 .
## Residuals 12 275.55  22.962
## ---
## Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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。

```
# perform multiple simulations
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(bloamat), 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] 0.54
sum(p.rbd < 0.01) / n.rep
## [1] 0.74
sum(p.crd < 0.01) / n.rep
## [1] 0.21
```

レポート課題

1. 一穂あたりの種子数 (Seed.number.per.panicle) を従属変数 y 、穂長 (Panicle.length) を独立変数 x として、単回帰モデル($y_i = \mu + \beta x_i + \epsilon_i$)をあてはめ、 μ と β の推定値である標本切片 m と標本回帰係数 b を求めよ。
2. 帰無仮説 $H_0: \beta = 0.02$ について検定を行え。
3. 帰無仮説 $H_0: \mu = 5$ について検定を行え。
4. $x = 27$ のときの y の推定値 \hat{y} と予測値 \tilde{y} の 95%信頼区間を答えよ。
5. x の 1 次の項と 2 次の項を用いた多項式回帰モデル $y_i = \mu + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$ をあてはめ、決定係数 R^2 と自由度決定済決定係数 R_{adj}^2 を答えよ。
6. 5 の回帰モデルと、1 の回帰モデルを分散分析で比較して、回帰モデルに x の 2 次の項を入れることの有効性について検討せよ。
7. x の 1~3 次の項を用いた多項式回帰モデル $y_i = \mu + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i$ をあてはめ、決定係数 R^2 と自由度決定済決定係数 R_{adj}^2 を答えよ。
8. 7 の回帰モデルと、5 の回帰モデルを分散分析で比較して、2 次の多項式モデルを 3 次の多項式モデルに拡張することの有効性について検討せよ。

提出方法 :

- レポートは「pdf ファイル」として作成し、ITC-LMS を通じて提出する。
- ただし、ITC-LMS が何らかの理由で利用できないときは、「report@iu.a.u-tokyo.ac.jp宛」にメール添付で送る。
- 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、5 月 14 日