Some R- Basics

BST 226
Statistical Methods for Bioinformatics
David M. Rocke

R, Stata, and other packages

- R and other packages have many of the same functions
- Some of the others can be run more easily as "point and shoot"
- All should be run from command files to document analyses
- None is really harder than the others, but the syntax and overall conception is different

Origins

- S was a statistical and graphics language developed at Bell Labs in the “one letter” days (i.e., the c programming language)
- R is an implementation of S, as is S-Plus, a commercial statistical package
- R is free, open source, runs on Windows, OS X, and Linux

Why use R?

- Bioconductor is a project collecting packages for biological data analysis, graphics, and annotation
- Most of the better methods are only available in Bioconductor or stand-alone packages
- With some exceptions, commercial microarray analysis packages are not competitive
- For RNA-Seq, most methods are available only in R

Getting Data into R

- Many times the most direct method is to edit the data in Excel, Export as a .txt file, then import to R using read.delim
- We will do this two ways for some energy expenditure data
- Frequently, the data from studies I am involved in arrives in Excel

energy package:ISwR R Documentation

Energy expenditure

Description:

The 'energy' data frame has 22 rows and 2 columns. It contains data on the energy expenditure in groups of lean and obese women.

Format:

This data frame contains the following columns:

expend a numeric vector. 24 hour energy expenditure (MJ).

stature a factor with levels 'lean' and 'obese'.

Source:

D.G. Altman (1991), _Practical Statistics for Medical Research_, Table 9.4, Chapman & Hall.

```
> setwd("c:/td/classes/BST226 2014 Winter/RData/")
> source("energy.r",echo=T)
> eg <- read.delim("energy1.txt")
> eg
   Obese  Lean
1   9.21  7.53
2  11.51  7.48
3  12.79  8.08
4  11.85  8.09
5   9.97 10.15
6   8.79  8.40
7   9.69 10.88
8   9.68  6.13
9   9.19  7.90
10    NA  7.05
11    NA  7.48
12    NA  7.58
13    NA  8.11
```

```
> class(eg)
[1] "data.frame"
> t.test(eg$Obese,eg$Lean)

        Welch Two Sample t-test

data:  eg$Obese and eg$Lean
t = 3.8555, df = 15.919, p-value = 0.001411
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.004081 3.459167
sample estimates:
mean of x mean of y
10.297778  8.066154
```

```
> attach(eg)

> t.test(Obese, Lean)

        Welch Two Sample t-test

data:  Obese and Lean
t = 3.8555, df = 15.919, p-value = 0.001411
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.004081 3.459167
sample estimates:
mean of x mean of y
10.297778  8.066154


> detach(eg)
```

```
> eg2 <- read.delim("energy2.txt")
> eg2
   expend stature
1    9.21   Obese
2   11.51   Obese
3   12.79   Obese
4   11.85   Obese
5    9.97   Obese
6    8.79   Obese
7    9.69   Obese
8    9.68   Obese
9    9.19   Obese
10   7.53    Lean
11   7.48    Lean
12   8.08    Lean
13   8.09    Lean
14  10.15    Lean
15   8.40    Lean
16  10.88    Lean
17   6.13    Lean
18   7.90    Lean
19   7.05    Lean
20   7.48    Lean
21   7.58    Lean
22   8.11    Lean
```

For this data set we can read directly from the package ISwR:

1. Packages/Install/ISwR
2. library(ISwR)
3. data(energy)

```
> class(eg2)
[1] "data.frame"

> t.test(eg2$expend ~ eg2$stature)

        Welch Two Sample t-test

data:  eg2$expend by eg2$stature
t = -3.8555, df = 15.919, p-value = 0.001411
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.459167 -1.004081
sample estimates:
 mean in group Lean mean in group Obese
          8.066154           10.297778
```

```
> attach(eg2)

> t.test(expend ~ stature)

        Welch Two Sample t-test

data:  expend by stature
t = -3.8555, df = 15.919, p-value = 0.001411
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.459167 -1.004081
sample estimates:
 mean in group Lean mean in group Obese
          8.066154           10.297778

> mean(expend[stature == "Lean"])
[1] 8.066154

> mean(expend[stature == "Obese"])
[1] 10.29778

> detach(eg2)
```

```
> tapply(expend, stature, mean)
     Lean     Obese
 8.066154 10.297778

> tmp <- tapply(expend, stature, mean)

> class(tmp)
[1] "array"

> dim(tmp)
[1] 2

> tmp[1] - tmp[2]
     Lean
-2.231624

> detach(eg2)
```

```
source("myprogs.r")

--------------------------------
mystats <- function(df)
{
  groups <- sort(unique(df[,2]))
  m <- length(groups)
  for (i in 1:m)
  {
    subseti <- df[,2]==groups[i]
    print(mean(df[subseti,1]))
  }
}

--------------------------------

> mystats(eg2)
[1] 8.066154
[1] 10.29778
```

Using R for Linear Regression

- The lm() command is used to do linear regression
- In many statistical packages, execution of a regression command results in lots of output
- In R, the lm() command produces a linear models object that contains the results of the linear model

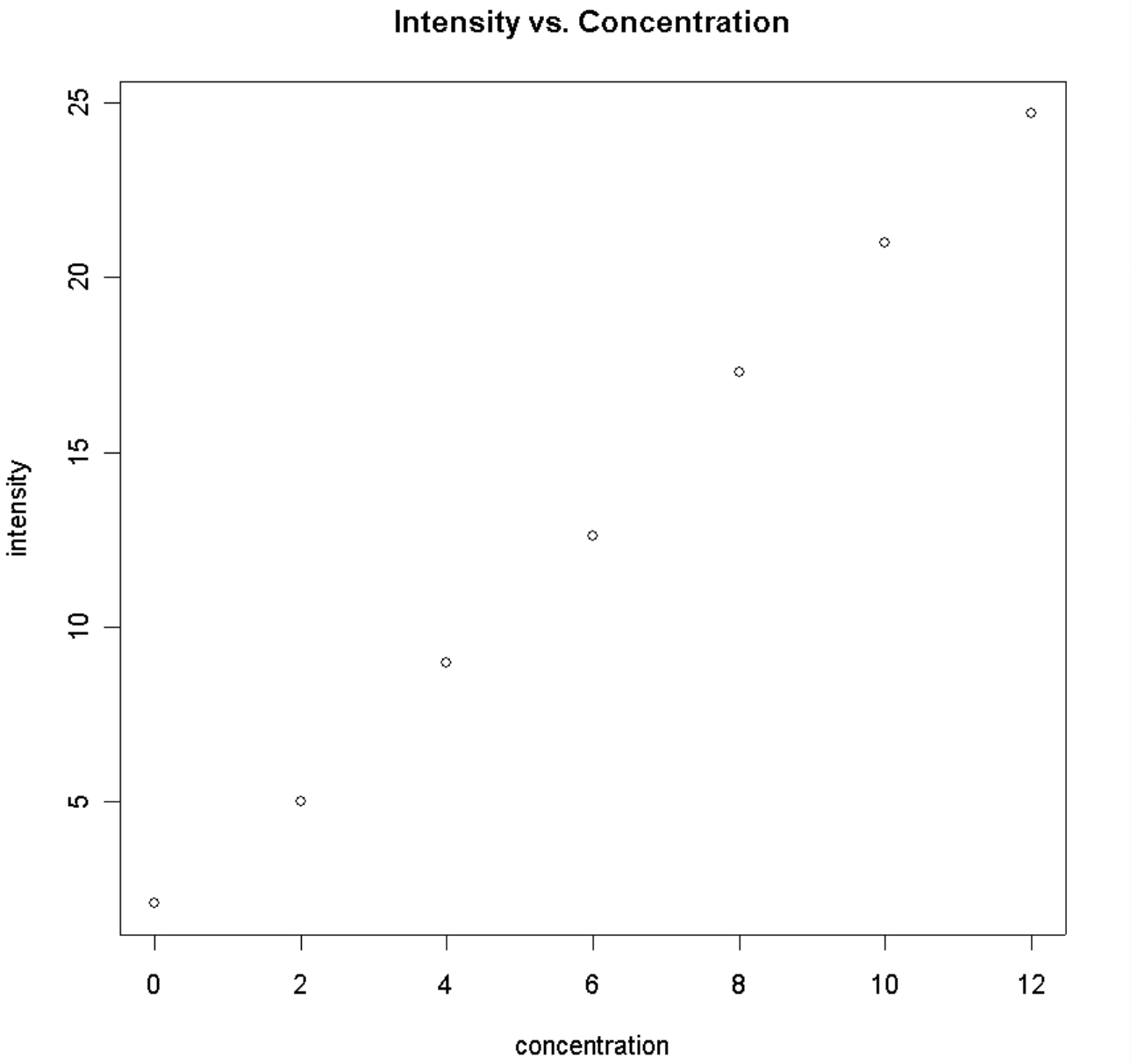
Formulas, output and extractors

- If `gene.exp` is a response, and `rads` is a level of radiation to which the cell culture is exposed, then `lm(gene.exp ~ rads)` computes the regression
- `lmobj <- lm(gene.exp ~ rads)`
- `Summary(lmobj)`
- `coef, resid(), fitted, …`

Example Analysis

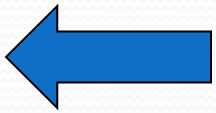
- Standard aqueous solutions of fluorescein (in pg/ml) are examined in a fluorescence spectrometer and the intensity (arbitrary units) is recorded
- What is the relationship of intensity to concentration
- Use later to infer concentration of labeled analyte

```
concentration <- c(0,2,4,6,8,10,12)
intensity <- c(2.1,5.0,9.0,12.6,17.3,21.0,24.7)
fluor <- data.frame(concentration,intensity)
> fluor
  concentration intensity
1             0       2.1
2             2       5.0
3             4       9.0
4             6      12.6
5             8      17.3
6            10      21.0
7            12      24.7

> attach(fluor)
> plot(concentration,intensity)
> title("Intensity vs. Concentration”)
```


Intensity vs. Concentration
intensity
5
10
15
20
25
0
2
4
6
8
10
12
concentration

```
> fluor.lm <- lm(intensity ~ concentration)
> summary(fluor.lm)

Call:
lm(formula = intensity ~ concentration)

Residuals:
       1        2        3        4        5        6        7
 0.58214 -0.37857 -0.23929 -0.50000  0.33929  0.17857  0.01786

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.5179     0.2949   5.146  0.00363 **
concentration  1.9304     0.0409  47.197 8.07e-08 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.4328 on 5 degrees of freedom
Multiple R-Squared: 0.9978,     Adjusted R-squared: 0.9973
F-statistic:  2228 on 1 and 5 DF,  p-value: 8.066e-08
```

```
> fluor.lm <- lm(intensity ~ concentration)
> summary(fluor.lm)

Call:
lm(formula = intensity ~ concentration)

Residuals:
      1       2       3       4       5       6       7
 0.5821 -0.3786 -0.2393 -0.5000  0.3393  0.1786  0.0179

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.5179     0.2949   5.146  0.00363 **
concentration  1.9304     0.0409  47.197 8.07e-08 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.4328 on 5 degrees of freedom
Multiple R-Squared: 0.9978,     Adjusted R-squared: 0.9973
F-statistic:  2228 on 1 and 5 DF,  p-value: 8.066e-08
```



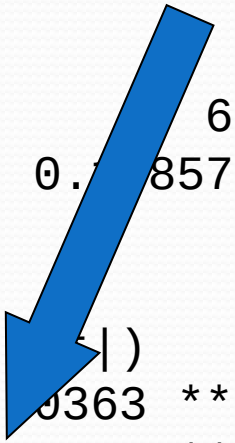
Formula

```
> fluor.lm <- lm(intensity ~ concentration)
> summary(fluor.lm)

Call:
lm(formula = intensity ~ concentration)

Residuals:
      1         2         3         4         5         6         7
 0.58214 -0.37857 -0.23929 -0.50000  0.33929  0.17857  0.01786

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.5179     0.2949   5.146  0.00363 **
concentration  1.9304     0.0409  47.197 8.07e-08 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.4328 on 5 degrees of freedom
Multiple R-Squared: 0.9978,     Adjusted R-squared: 0.9973
F-statistic:  2228 on 1 and 5 DF,  p-value: 8.066e-08
```

Residuals

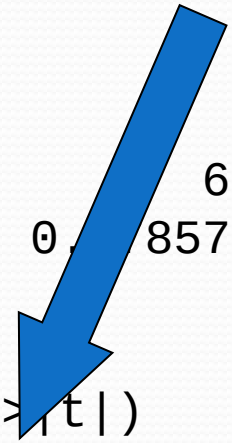
```
> fluor.lm <- lm(intensity ~ concentration)
> summary(fluor.lm)

Call:
lm(formula = intensity ~ concentration)

Residuals:
       1        2        3        4        5        6        7
 0.58214 -0.37857 -0.23929 -0.50000  0.33929  0.[illegible]857  0.01786

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.5179     0.2949   5.146  0.[illegible]0363 **
concentration  1.9304     0.0409  47.197 8.07e-08 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.4328 on 5 degrees of freedom
Multiple R-Squared: 0.9978,     Adjusted R-squared: 0.9973
F-statistic:  2228 on 1 and 5 DF,  p-value: 8.066e-08
```

Slope coefficient



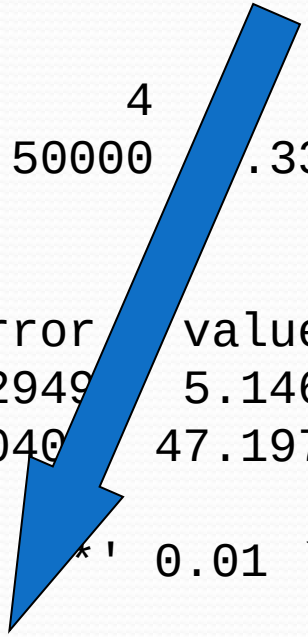
```
> fluor.lm <- lm(intensity ~ concentration)
> summary(fluor.lm)

Call:
lm(formula = intensity ~ concentration)

Residuals:
      1       2       3       4       5       6       7
 0.5821 -0.3786 -0.2393 -0.5000  0.3393  0.1857  0.0179

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.5179     0.2949   5.146  0.00363 **
concentration  1.9304     0.0409  47.197 8.07e-08 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.4328 on 5 degrees of freedom
Multiple R-Squared: 0.9978,     Adjusted R-squared: 0.9973
F-statistic:  2228 on 1 and 5 DF,  p-value: 8.066e-08
```

Intercept (intensity at zero concentration)



```
> fluor.lm <- lm(intensity ~ concentration)
> summary(fluor.lm)

Call:
lm(formula = intensity ~ concentration)

Residuals:
      1         2         3         4         5         6         7
 0.58214 -0.37857 -0.23929 -0.50000  0.33929  0.17857  0.01786

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.5179     0.2949   5.146  0.00363 **
concentration  1.9304     0.040   47.197 8.07e-08 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.4328 on 5 degrees of freedom
Multiple R-Squared: 0.9978,     Adjusted R-squared: 0.9973
F-statistic:  2228 on 1 and 5 DF,  p-value: 8.066e-08
```

Variability around regression line



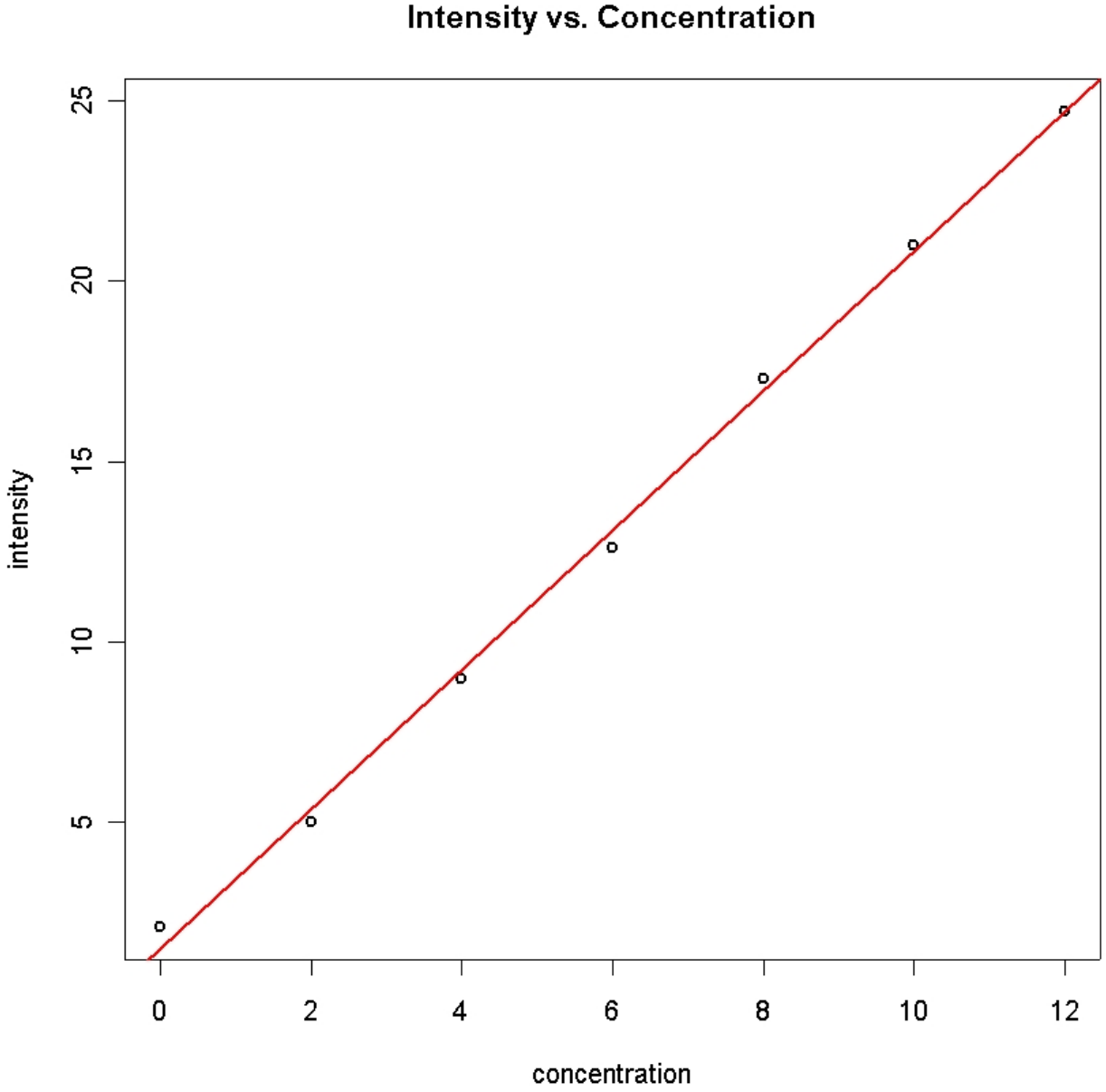
```
> fluor.lm <- lm(intensity ~ concentration)
> summary(fluor.lm)

Call:
lm(formula = intensity ~ concentration)

Residuals:
       1        2        3        4        5        6        7
 0.58214 -0.37857 -0.23929 -0.50000  0.33929  0.17857  0.01786

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.5179     0.2949   5.146  0.00363 **
concentration  1.9304     0.0409  47.197 8.07e-08 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 0.4328 on 5 degrees of freedom
Multiple R-Squared: 0.9978,     Adjusted R-squared: 0.9973
F-statistic:  2228 on 1 and 5 DF,  p-value: 8.066e-08
```

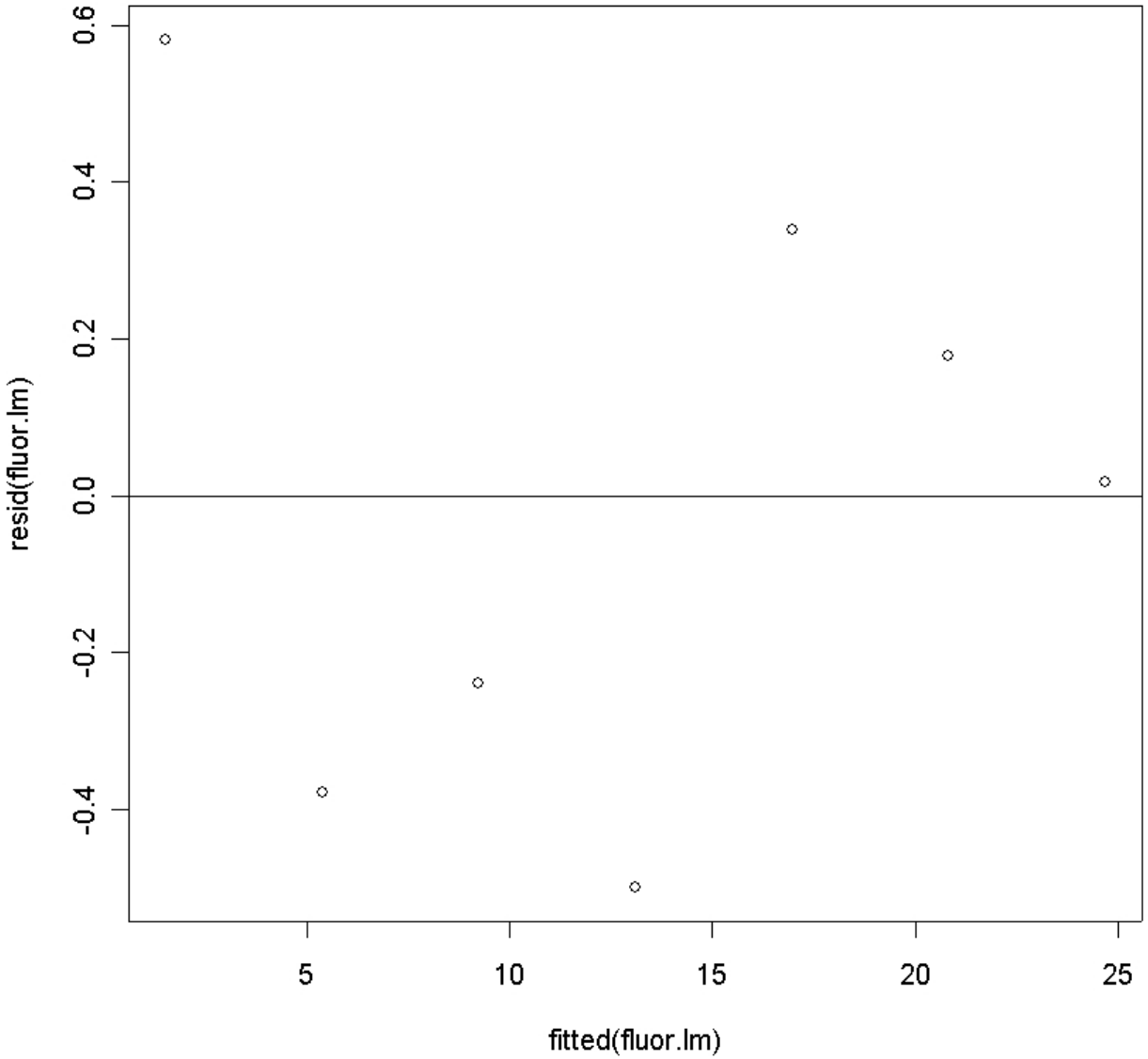
Test of overall significance of model

```
> plot(concentration,intensity,lw=2)
> title("Intensity vs. Concentration")
> abline(coef(fluor.lm),lwd=2,col="red")

> plot(fitted(fluor.lm),resid(fluor.lm))
> abline(h=0)
```

The first of these plots shows the data points and the regression line.

The second shows the residuals vs. fitted values, which is better at detecting nonlinearity

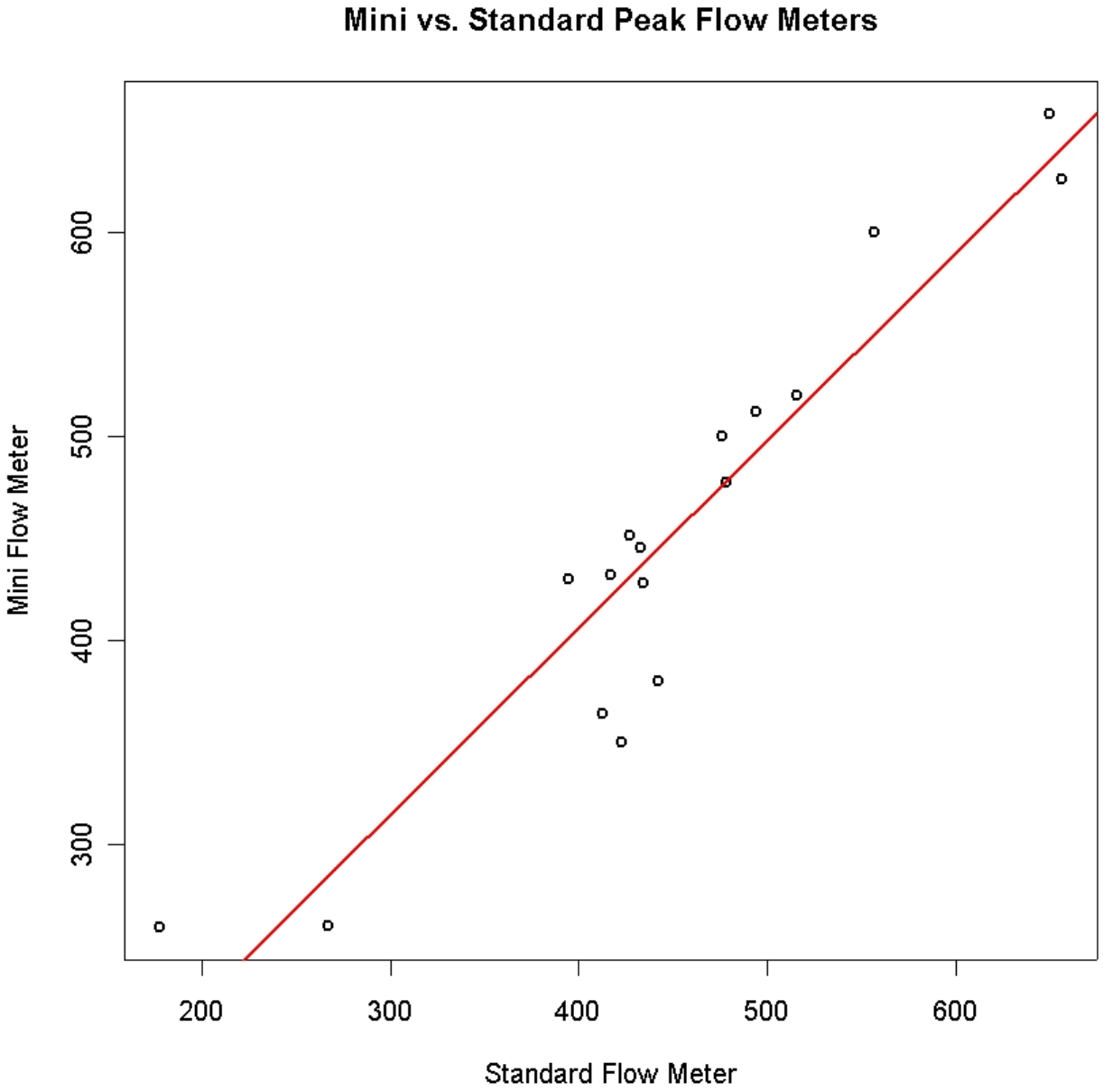

Intensity vs. Concentration
intensity
concentration
5
10
15
20
25
0
2
4
6
8
10
12


resid(fluor.lm)
0.6
0.4
0.2
0.0
-0.2
-0.4
5
10
15
20
25
fitted(fluor.lm)

```
> setwd("c:/td/classes/SPH247 2010 Spring/RData/")
> source("wright.r")
> cor(wright)
             std.wright mini.wright
std.wright    1.0000000   0.9432794
mini.wright   0.9432794   1.0000000

> wplot1()
```

File wright.r:

```
library(ISwR)
data(wright)
attach(wright)

wplot1 <- function()
{
  plot(std.wright,mini.wright,xlab="Standard Flow Meter",
     ylab="Mini Flow Meter",lwd=2)
  title("Mini vs. Standard Peak Flow Meters")
  wright.lm <- lm(mini.wright ~ std.wright)
  abline(coef(wright.lm),col="red",lwd=2)
}
detach(wright)
```


Mini vs. Standard Peak Flow Meters
Mini Flow Meter
300
400
500
600
200
300
400
500
600
Standard Flow Meter

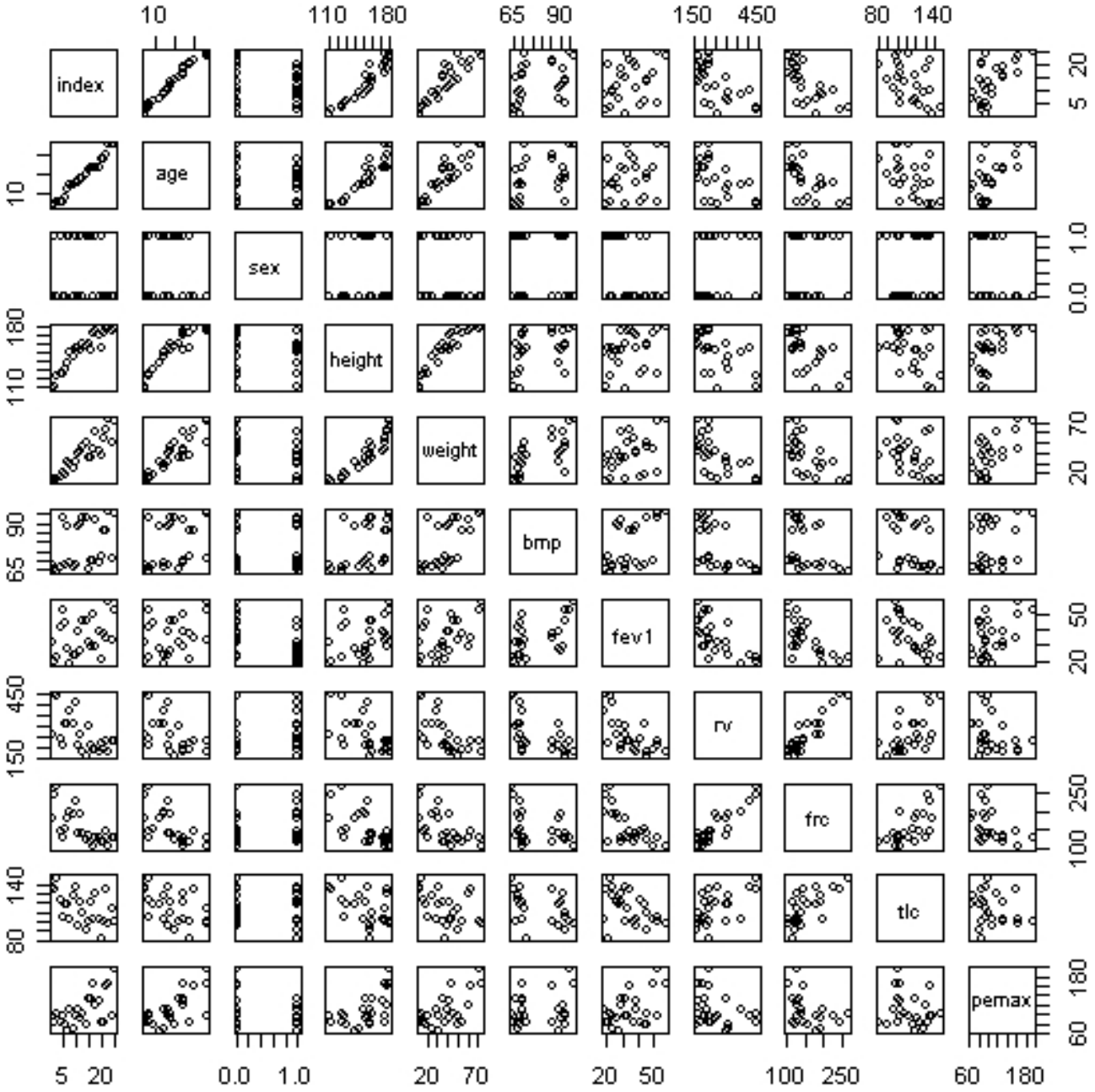
Cystic Fibrosis Data

Cystic fibrosis lung function data

lung function data for cystic fibrosis patients (7-23 years old)

age	a numeric vector. Age in years.
sex	a numeric vector code. 0: male, 1:female.
height	a numeric vector. Height (cm).
weight	a numeric vector. Weight (kg).
bmp	a numeric vector. Body mass (% of normal).
fev1	a numeric vector. Forced expiratory volume.
rv	a numeric vector. Residual volume.
frc	a numeric vector. Functional residual capacity.
tlc	a numeric vector. Total lung capacity.
pemax	a numeric vector. Maximum expiratory pressure.

```
cf <- read.csv("cystfibr.csv")
pairs(cf)
attach(cf)
cf.lm <- lm(pemax ~ age+sex+height+weight+bmp+fev1+rv+frc+tlc)
print(summary(cf.lm))
print(anova(cf.lm))
print(drop1(cf.lm,test="F"))
plot(cf.lm)
step(cf.lm)
detach(cf)
```


index
age
sex
height
weight
bmp
fev1
rv
frc
tlc
pemax

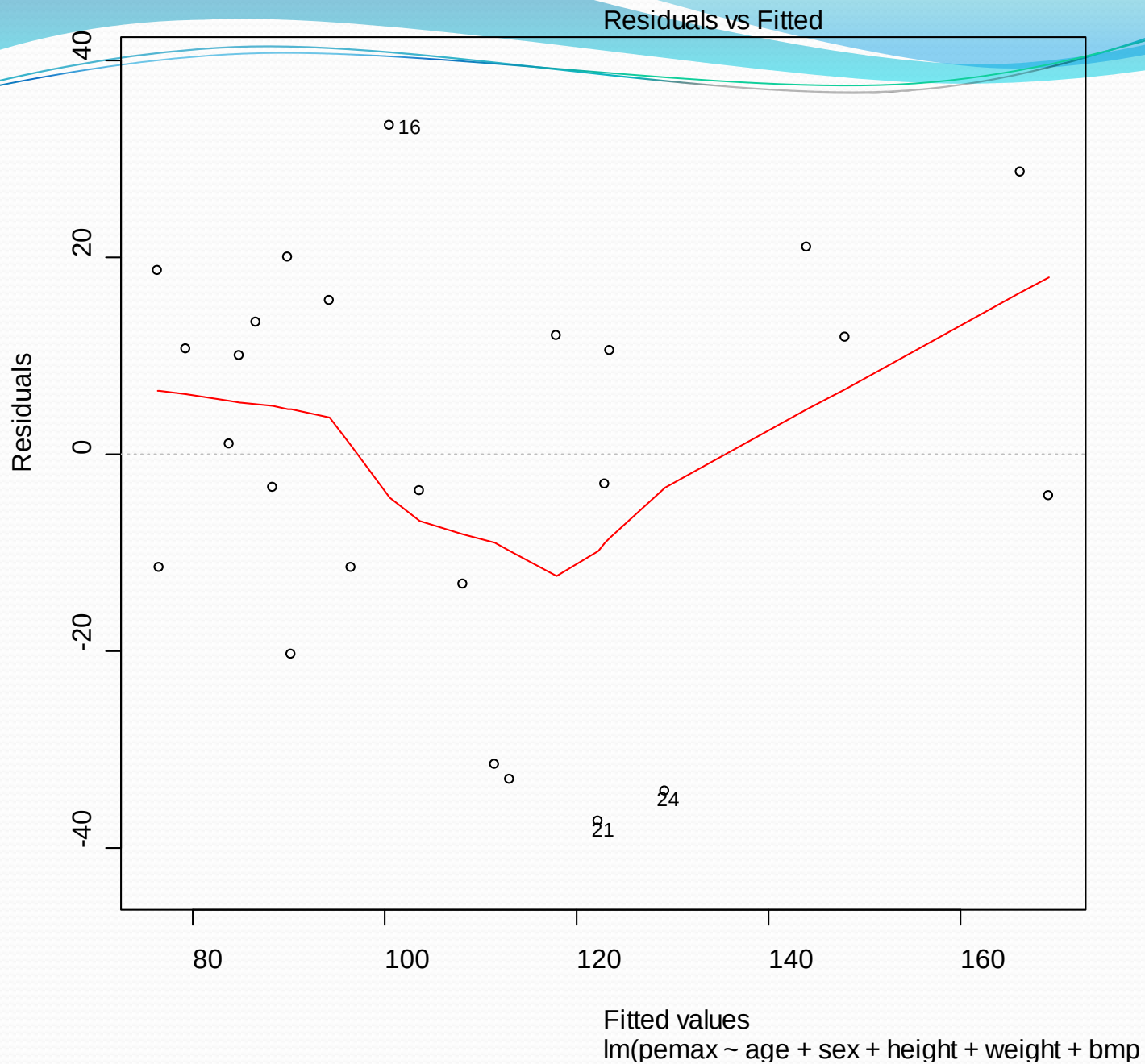
```
> source("cystfibr.r")
> cf.lm <- lm(pemax ~ age + sex + height + weight + bmp + fev1 +
  rv + frc + tlc)
> print(summary(cf.lm))
…

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 176.0582   225.8912   0.779    0.448
age          -2.5420     4.8017  -0.529    0.604
sex          -3.7368    15.4598  -0.242    0.812
height       -0.4463     0.9034  -0.494    0.628
weight        2.9928     2.0080   1.490    0.157
bmp          -1.7449     1.1552  -1.510    0.152
fev1          1.0807     1.0809   1.000    0.333
rv            0.1970     0.1962   1.004    0.331
frc          -0.3084     0.4924  -0.626    0.540
tlc           0.1886     0.4997   0.377    0.711

Residual standard error: 25.47 on 15 degrees of freedom
Multiple R-Squared: 0.6373,     Adjusted R-squared: 0.4197
F-statistic: 2.929 on 9 and 15 DF,  p-value: 0.03195
```

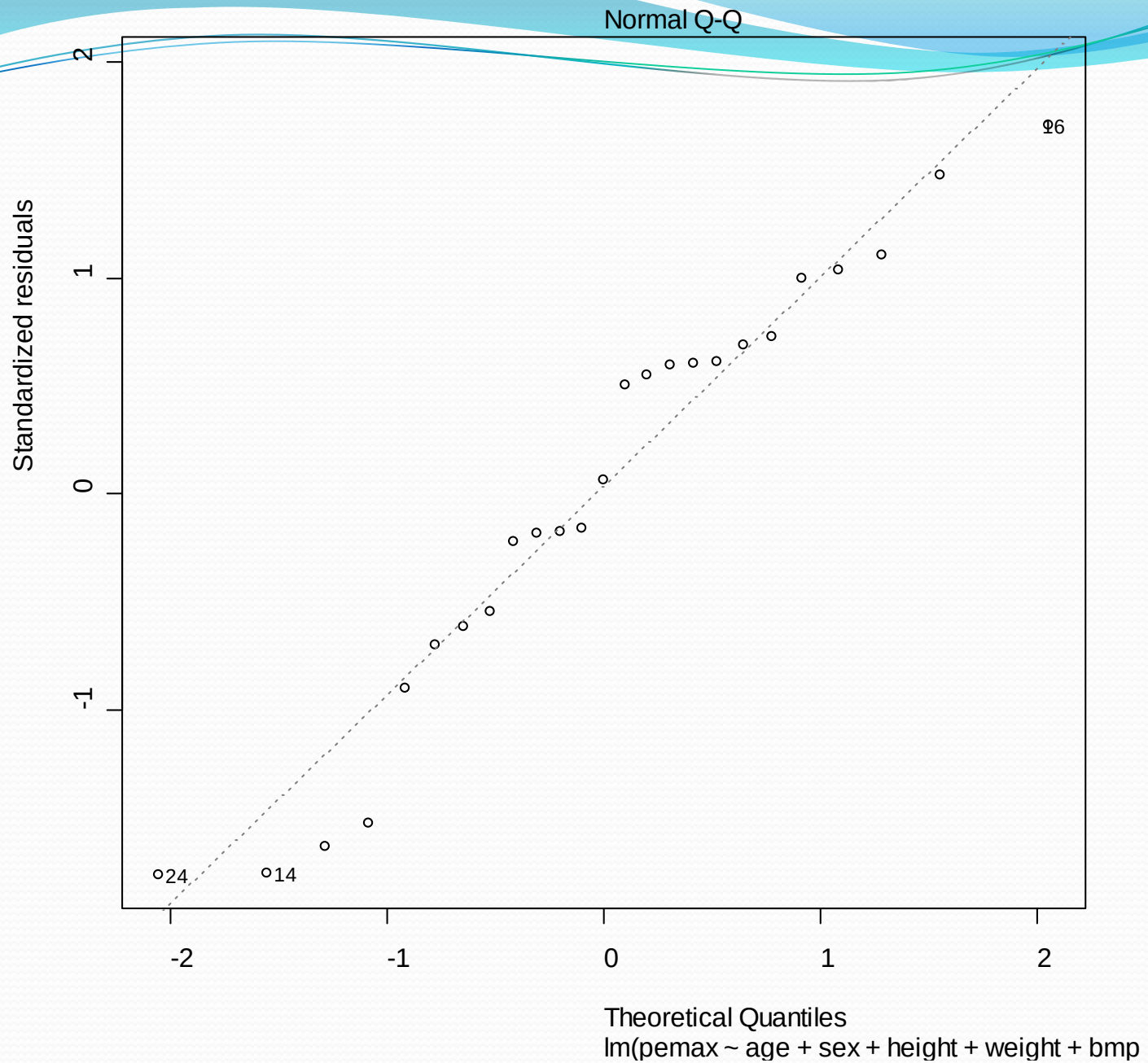
```
> print(anova(cf.lm))
Analysis of Variance Table

Response: pemax
          Df  Sum Sq Mean Sq F value   Pr(>F)
age        1 10098.5 10098.5 15.5661 0.001296 **
sex        1   955.4   955.4  1.4727 0.243680
height     1   155.0   155.0  0.2389 0.632089
weight     1   632.3   632.3  0.9747 0.339170
bmp        1  2862.2  2862.2  4.4119 0.053010 .
fev1       1  1549.1  1549.1  2.3878 0.143120
rv         1   561.9   561.9  0.8662 0.366757
frc        1   194.6   194.6  0.2999 0.592007
tlc        1    92.4    92.4  0.1424 0.711160
Residuals 15  9731.2   648.7
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

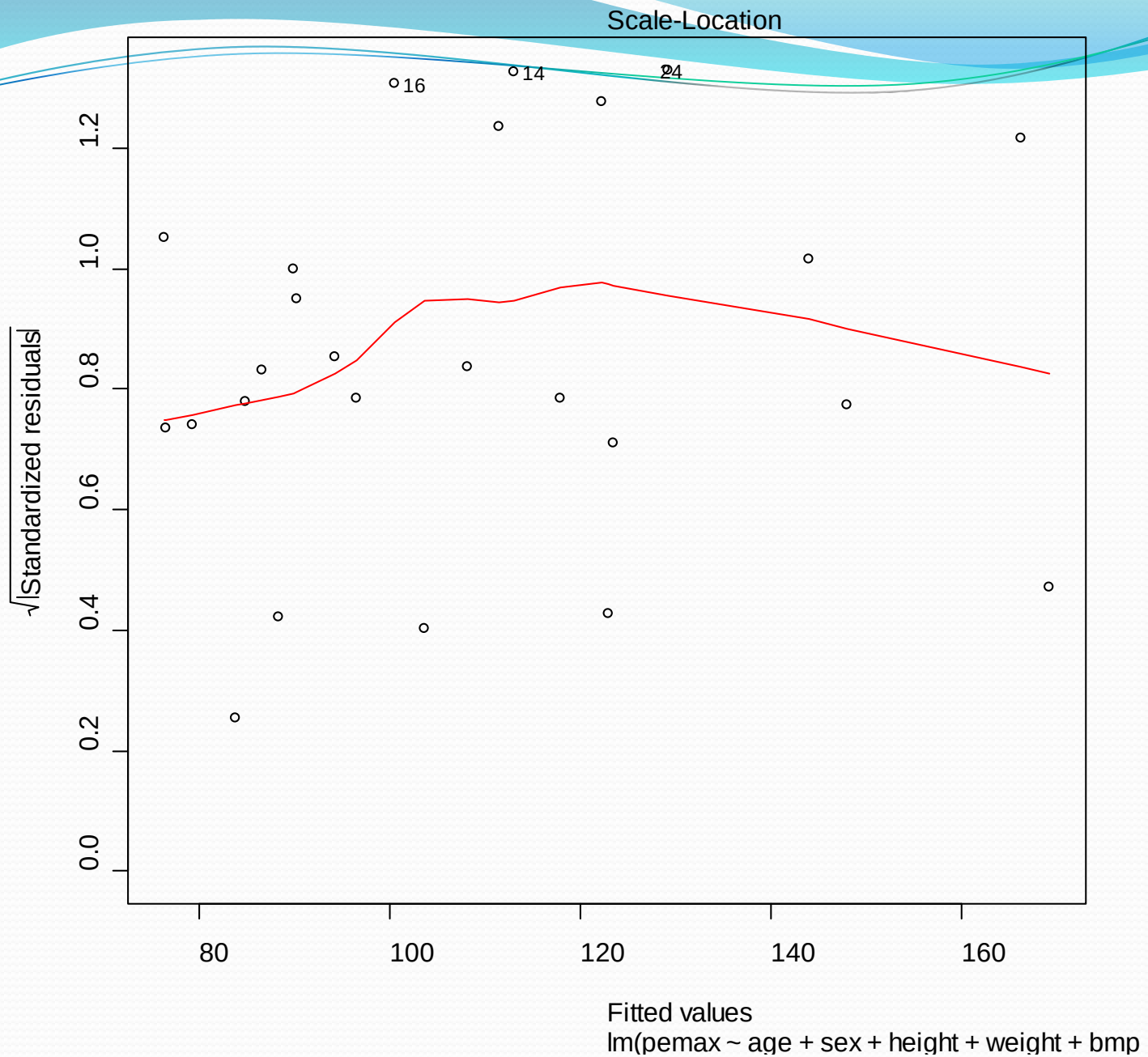
Performs sequential ANOVA

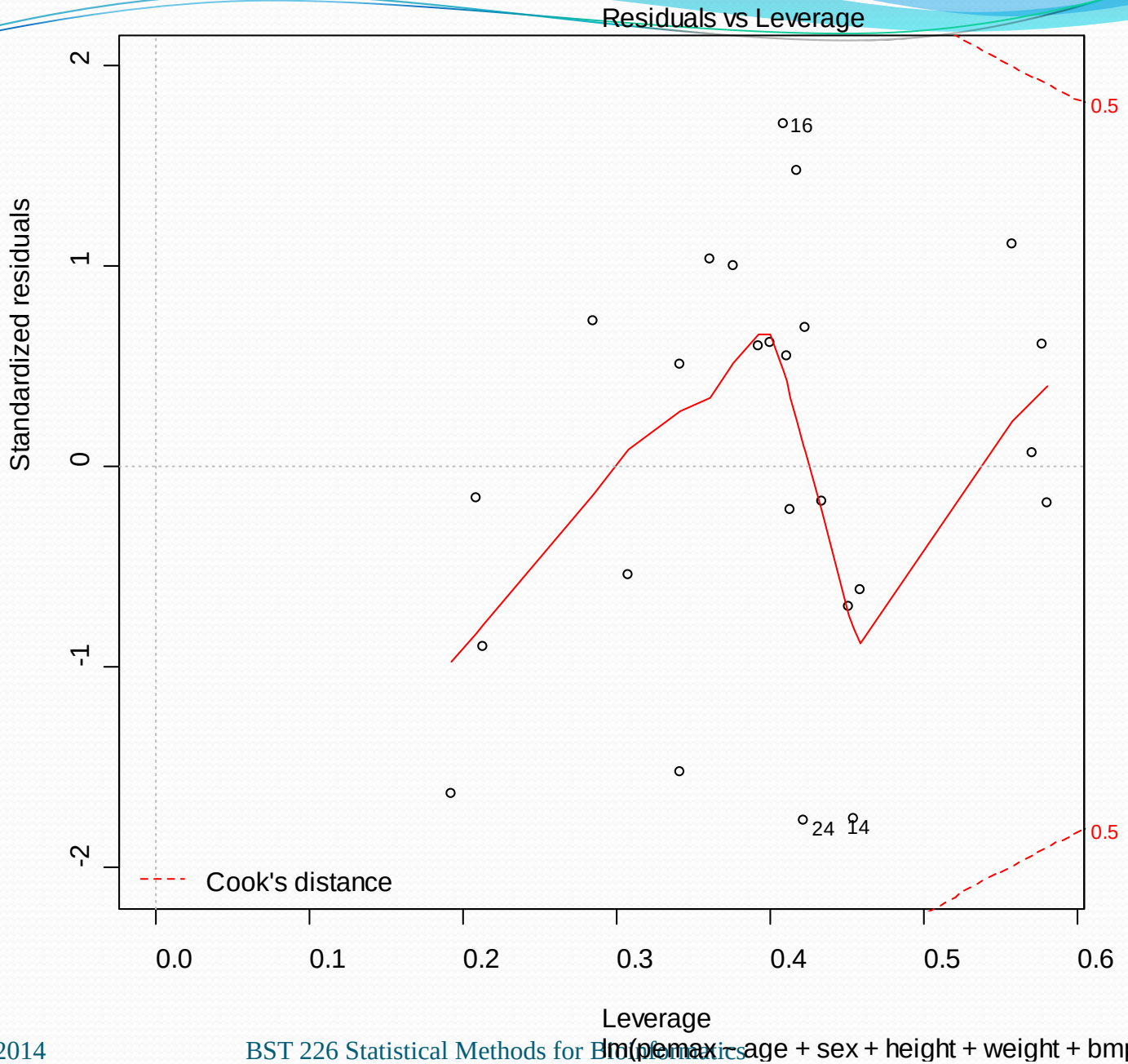
```
> print(drop1(cf.lm, test = "F"))

Single term deletions

Model:
pemax ~ age + sex + height + weight + bmp + fev1 + rv + frc +
    tlc
       Df Sum of Sq     RSS     AIC F value  Pr(F)
<none>                9731.2   169.1
age     1     181.8  9913.1   167.6  0.2803 0.6043
sex     1      37.9  9769.2   167.2  0.0584 0.8123
height  1     158.3  9889.6   167.5  0.2440 0.6285
weight  1    1441.2 11172.5   170.6  2.2215 0.1568
bmp     1    1480.1 11211.4   170.6  2.2815 0.1517
fev1    1     648.4 10379.7   168.7  0.9995 0.3333
rv      1     653.8 10385.0   168.7  1.0077 0.3314
frc     1     254.6  9985.8   167.8  0.3924 0.5405
tlc     1      92.4  9823.7   167.3  0.1424 0.7112
```

Performs Type III ANOVA


Residuals vs Fitted
Residuals
40
20
0
-20
-40
16
24
21
80
100
120
140
160
Fitted values
lm(pemax ~ age + sex + height + weight + bmp


Normal Q-Q
Standardized residuals
2
1
0
-1
16
24
14
-2
-1
0
1
2
Theoretical Quantiles
lm(pemax ~ age + sex + height + weight + bmp


Scale-Location
16
14
24
√|Standardized residuals|
0.0
0.2
0.4
0.6
0.8
1.0
1.2
80
100
120
140
160
Fitted values
lm(pemax ~ age + sex + height + weight + bmp


Residuals vs Leverage
Standardized residuals
Leverage
lm(pemax ~ age + sex + height + weight + bmp)
Cook's distance
0.5
0.5
16
24
14
2
1
0
-1
-2
0.0
0.1
0.2
0.3
0.4
0.5
0.6

```
> step(cf.lm)
Start:  AIC=169.11
pemax ~ age + sex + height + weight + bmp + fev1 + rv + frc +
    tlc

          Df Sum of Sq     RSS     AIC
- sex      1      37.9  9769.2   167.2
- tlc      1      92.4  9823.7   167.3
- height   1     158.3  9889.6   167.5
- age      1     181.8  9913.1   167.6
- frc      1     254.6  9985.8   167.8
- fev1     1     648.4 10379.7   168.7
- rv       1     653.8 10385.0   168.7
<none>                   9731.2   169.1
- weight   1    1441.2 11172.5   170.6
- bmp      1    1480.1 11211.4   170.6

Step:  AIC=167.2
pemax ~ age + height + weight + bmp + fev1 + rv + frc + tlc
```

……

```
Step:  AIC=160.66
pemax ~ weight + bmp + fev1 + rv

          Df Sum of Sq     RSS     AIC
<none>                  10354.6   160.7
- rv       1    1183.6 11538.2   161.4
- bmp      1    3072.6 13427.2   165.2
- fev1     1    3717.1 14071.7   166.3
- weight   1   10930.2 21284.8   176.7

Call:
lm(formula = pemax ~ weight + bmp + fev1 + rv)

Coefficients:
(Intercept)       weight          bmp         fev1           rv
    63.9467       1.7489      -1.3772       1.5477       0.1257
```

```
> cf.lm2 <- lm(pemax ~ rv+bmp+fev1+weight)
> summary(cf.lm2)

Call:
lm(formula = pemax ~ rv + bmp + fev1 + weight)

Residuals:
   Min     1Q Median     3Q    Max
-39.77 -11.74   4.33  15.66  35.07

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 63.94669   53.27673   1.200 0.244057
rv           0.12572    0.08315   1.512 0.146178
bmp         -1.37724    0.56534  -2.436 0.024322 *
fev1         1.54770    0.57761   2.679 0.014410 *
weight       1.74891    0.38063   4.595 0.000175 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 22.75 on 20 degrees of freedom
Multiple R-Squared: 0.6141,     Adjusted R-squared: 0.5369
F-statistic: 7.957 on 4 and 20 DF,  p-value: 0.000523
```

red.cell.folate package:ISwR R Documentation

Red cell folate data

Description:

The 'folate' data frame has 22 rows and 2 columns. It contains data on red cell folate levels in patients receiving three different methods of ventilation during anesthesia.

Format:

This data frame contains the following columns:

folate a numeric vector. Folate concentration (mug/l).

ventilation a factor with levels 'N2O+O2,24h': 50% nitrous oxide and 50% oxygen, continuously for 24~hours; 'N2O+O2,op': 50% nitrous oxide and 50% oxygen, only during operation; 'O2,24h': no nitrous oxide, but 35-50% oxygen for 24~hours.

```
> data(red.cell.folate)
> help(red.cell.folate)
> summary(red.cell.folate)
     folate            ventilation
 Min.   :206.0   N2O+O2,24h:8
 1st Qu.:249.5   N2O+O2,op :9
 Median :274.0   O2,24h    :5
 Mean   :283.2
 3rd Qu.:305.5
 Max.   :392.0
> attach(red.cell.folate)
> plot(folate ~ ventilation)
```

```
>  folate.lm <- lm(folate ~ ventilation)
> summary(folate.lm)

Call:
lm(formula = folate ~ ventilation)

Residuals:
    Min      1Q  Median      3Q     Max
-73.625 -35.361  -4.444  35.625  75.375

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)            316.62      16.16  19.588 4.65e-14 ***
ventilationN2O+O2,op   -60.18      22.22  -2.709   0.0139 *
ventilationO2,24h      -38.62      26.06  -1.482   0.1548
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 45.72 on 19 degrees of freedom
Multiple R-Squared: 0.2809,     Adjusted R-squared: 0.2052
F-statistic: 3.711 on 2 and 19 DF,  p-value: 0.04359
```

```
> anova(folate.lm)
Analysis of Variance Table

Response: folate
            Df Sum Sq Mean Sq F value  Pr(>F)
ventilation  2  15516    7758  3.7113 0.04359 *
Residuals   19  39716    2090
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
```

```
> data(heart.rate)
> attach(heart.rate)
> heart.rate
    hr subj time
1   96    1    0
2  110    2    0
3   89    3    0
4   95    4    0
5  128    5    0
6  100    6    0
7   72    7    0
8   79    8    0
9  100    9    0
10  92    1   30
......
18 106    9   30
19  86    1   60
......
27 104    9   60
28  92    1  120
......
36 102    9  120
```

```
> anova(hr.lm)
Analysis of Variance Table

Response: hr
          Df Sum Sq Mean Sq F value    Pr(>F)
subj       8 8966.6  1120.8 90.6391 4.863e-16 ***
time       3  151.0    50.3  4.0696   0.01802 *
Residuals 24  296.8    12.4
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
```

Note that when the design is orthogonal, the ANOVA results don't depend on the order of terms.

Exercises

- Download R and install from website http://cran.r-project.org/
- Also download BioConductor
  - source("http://bioconductor.org/biocLite.R")
  - biocLite()
- Install package ISwR
- Try to replicate the analyses in the presentation