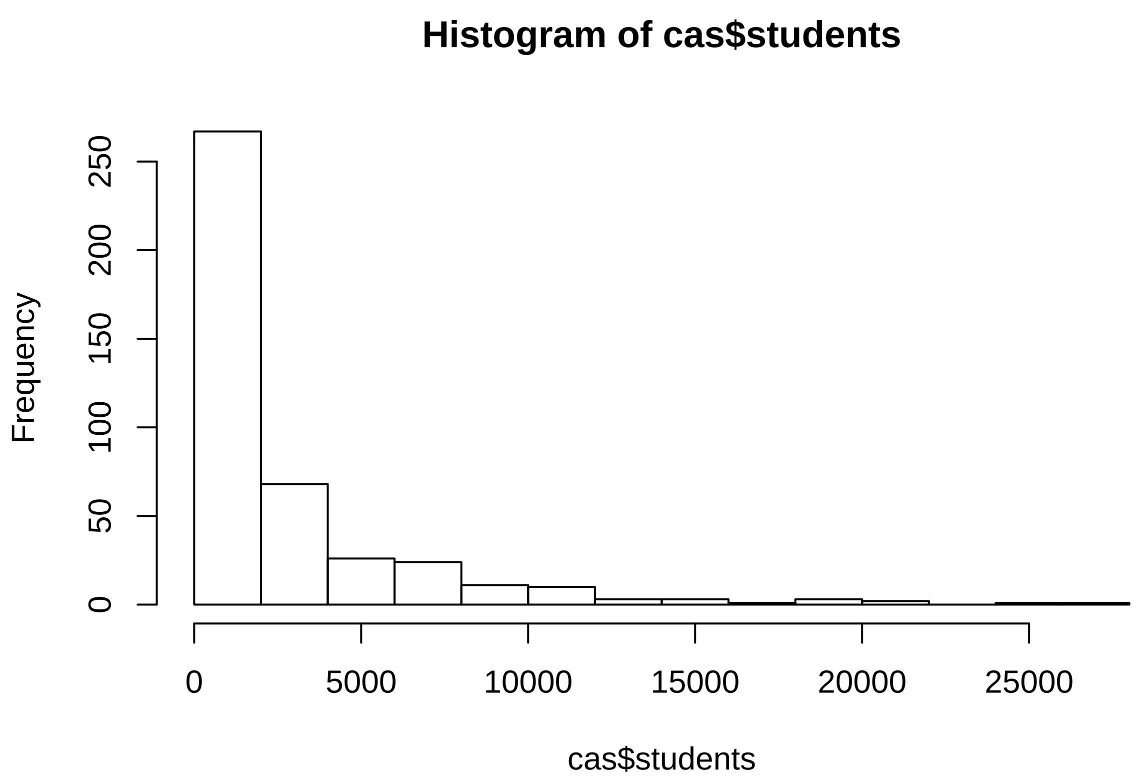
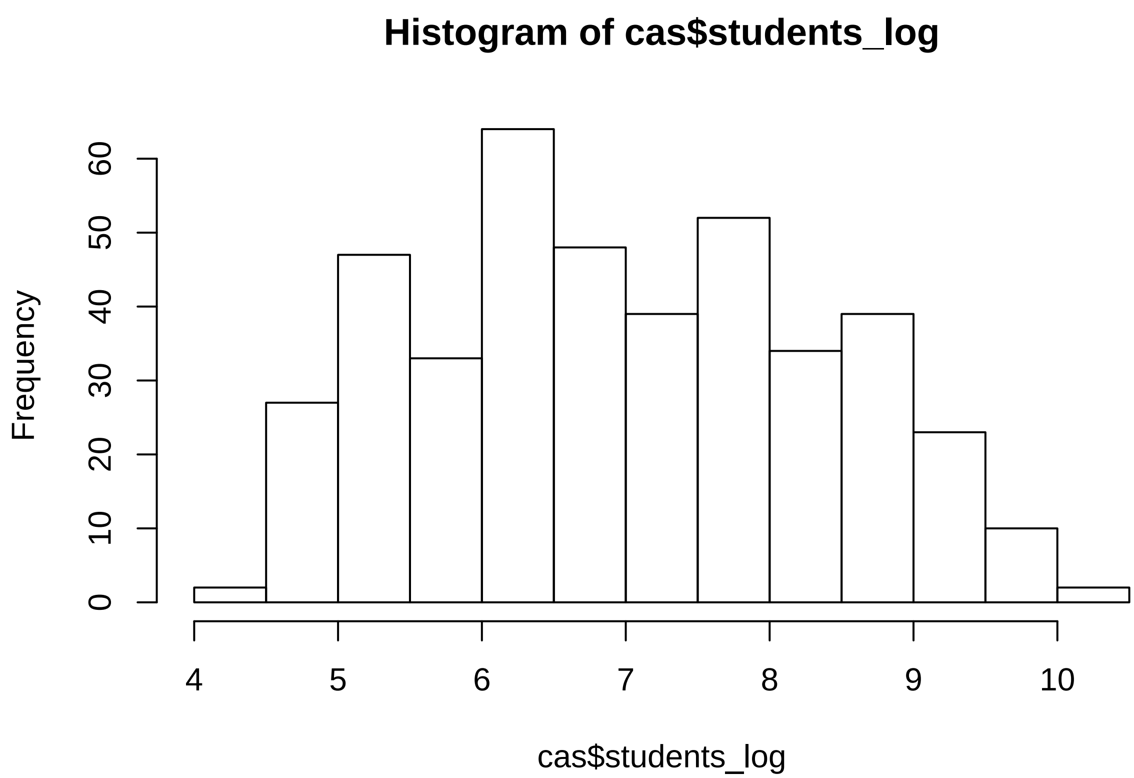
Introduction to R: Statistical Models Tutorial

Dr Jeromy Anglim

```
library(ProjectTemplate); load.project()

# And create some variables
library(AER)
data("CASchools")
?CASchools
cas <- CASchools

# create new vaiables
# academic performance as the sum of reading and mathematics
# performance
cas$performance <- as.numeric(scale(cas$read) + scale(cas$math))

# student-staff ratio
cas$student_teacher_ratio <- cas$students / cas$teachers

# computers per student
cas$computer_student_ratio <- cas$computer / cas$students

# Student size is quite skewed
hist(cas$students)
```



```r
# Let's log transform it
cas$students_log <- log(cas$students)
hist(cas$students_log)
```




```r
# same with average district income
cas$income_log <- log(cas$income)

dput(names(cas))
```

```
## c("district", "school", "county", "grades", "students", "teachers",
## "calworks", "lunch", "computer", "expenditure", "income", "english",
## "read", "math", "performance", "student_teacher_ratio", "computer_student_ratio",
## "students_log", "income_log")
```

```r
v <- list()

v$predictors <-
    c("calworks",      # percent of students qualifying for income assistance
    "lunch",          # percent qualifying for reduced price lunch
    "expenditure",    # expenditure per student
    "english",        # percent of english learners
    "student_teacher_ratio",
    "computer_student_ratio",
    "students_log",
    "income_log")
v$dv <- "performance"
v$all_variables <- c(v$predictors, v$dv)
```

Univariate statistics

```r
# sample size
nrow(cas)
```

```
## [1] 420
```

```r
# Frequencies or percentages on categorical variables
table(cas$grades) # frequency counts
```

```
##
## KK-06 KK-08
##    61   359
```

```r
prop.table(table(cas$grades)) # proportions
```

```
##
##     KK-06     KK-08
## 0.1452381 0.8547619
```

```r
# Descriptive statistics for continuous variables
round(psych::describe( cas[, v$all_variables]), 2)
```

```
##                         vars   n    mean     sd  median trimmed    mad
## calworks                   1 420   13.25  11.45   10.52   11.70  10.19
## lunch                      2 420   44.71  27.12   41.75   44.14  32.20
## expenditure                3 420 5312.41 633.94 5214.52 5252.95 487.17
## english                    4 420   15.77  18.29    8.78   12.54  11.76
## student_teacher_ratio      5 420   19.64   1.89   19.72   19.66   1.70
## computer_student_ratio     6 420    0.14   0.06    0.13    0.13   0.05
## students_log               7 420    6.99   1.38    6.86    6.96   1.57
## income_log                 8 420    2.64   0.39    2.62    2.62   0.38
## performance                9 420    0.00   1.96    0.03   -0.02   1.99
##                             min     max   range  skew kurtosis    se
## calworks                   0.00   78.99   78.99  1.68     4.55  0.56
## lunch                      0.00  100.00  100.00  0.18    -1.01  1.32
## expenditure             3926.07 7711.51 3785.44  1.06     1.85 30.93
## english                    0.00   85.54   85.54  1.42     1.41  0.89
## student_teacher_ratio     14.00   25.80   11.80 -0.03     0.59  0.09
## computer_student_ratio     0.00    0.42    0.42  0.92     1.41  0.00
## students_log               4.39   10.21    5.82  0.17    -0.94  0.07
## income_log                 1.67    4.01    2.34  0.65     0.76  0.02
## performance               -5.01    5.43   10.44  0.10    -0.26  0.10
```

```r
# Descriptive statistics for categorical and numeric variables
Hmisc::describe(cas)
```

```
## cas
##
```

```
##  19  Variables      420  Observations
## ---------------------------------------------------------------------------
## district
##       n missing  unique
##     420       0     420
##
## lowest : 61382 61457 61499 61507 61523
## highest: 75051 75085 75119 75135 75440
## ---------------------------------------------------------------------------
## school
##       n missing  unique
##     420       0     409
##
## lowest : Ackerman Elementary               Adelanto Elementary               Alexander Valley Union E
## highest: Woodlake Union Elementary         Woodside Elementary               Woodville Elementary
## ---------------------------------------------------------------------------
## county
##       n missing  unique
##     420       0      45
##
## lowest : Alameda      Butte        Calaveras    Contra Costa El Dorado
## highest: Trinity      Tulare       Tuolumne     Ventura      Yuba
## ---------------------------------------------------------------------------
## grades
##       n missing  unique
##     420       0       2
##
## KK-06 (61, 15%), KK-08 (359, 85%)
## ---------------------------------------------------------------------------
## students
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     391       1    2629   139.9   164.0   379.0   950.5
##     .75     .90     .95
##  3008.0  7119.5 10351.1
##
## lowest :    81    92   101   103   104
## highest: 19402 20927 21338 25151 27176
## ---------------------------------------------------------------------------
## teachers
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     374       1   129.1   7.076   9.000  19.662  48.565
##     .75     .90     .95
## 146.350 332.174 522.290
##
## lowest :    4.85    5.00    5.10    5.50    5.60
## highest:  924.57  953.50 1051.58 1186.70 1429.00
## ---------------------------------------------------------------------------
## calworks
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     411       1   13.25   0.745   1.996   4.395  10.520
##     .75     .90     .95
##  18.981  27.178  34.210
##
## lowest :  0.0000  0.0506  0.0800  0.1016  0.1517
```

```
## highest: 52.2199 55.0323 58.7522 71.7131 78.9942
## ---------------------------------------------------------------------------
## lunch
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     407       1   44.71   2.416  10.082  23.282  41.751
##     .75     .90     .95
##  66.865  83.123  90.302
##
## lowest :   0.0000   0.1239   0.1734   0.3033   0.5367
## highest:  94.9712  97.7597  98.1308  98.6056 100.0000
## ---------------------------------------------------------------------------
## computer
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     270       1   303.4    15.0    25.0    46.0   117.5
##     .75     .90     .95
##   375.2   790.1  1248.6
##
## lowest :    0    4    7    8   10, highest: 2001 2232 2401 2889 3324
## ---------------------------------------------------------------------------
## expenditure
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     420       1    5312    4441    4616    4906    5215
##     .75     .90     .95
##    5601    6108    6540
##
## lowest : 3926 4016 4024 4079 4136, highest: 7542 7593 7614 7668 7712
## ---------------------------------------------------------------------------
## income
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     337       1   15.32   7.751   8.930  10.639  13.728
##     .75     .90     .95
##  17.629  22.766  30.639
##
## lowest :  5.335  5.699  6.216  6.577  6.613
## highest: 41.734 43.230 49.939 50.677 55.328
## ---------------------------------------------------------------------------
## english
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     372       1   15.77   0.000   0.000   1.941   8.778
##     .75     .90     .95
##  22.970  43.784  53.440
##
## lowest :  0.00000  0.06333  0.11641  0.13298  0.14164
## highest: 76.66525 77.00581 80.12326 80.42009 85.53972
## ---------------------------------------------------------------------------
## read
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     322       1     655   620.7   629.4   640.4   655.8
##     .75     .90     .95
##   668.7   680.5   688.5
##
## lowest : 604.5 605.5 605.7 608.9 610.0
## highest: 698.9 699.1 700.9 701.3 704.0
## ---------------------------------------------------------------------------
```

```
## math
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     324       1   653.3   625.4   629.7   639.4   652.4
##     .75     .90     .95
##   665.8   676.8   685.0
##
## lowest : 605.4 609.0 612.5 613.4 616.0
## highest: 701.1 701.7 703.6 707.7 709.5
## ---------------------------------------------------------------------------
## performance
##         n   missing    unique      Info      Mean       .05       .10
##       420         0       420         1 1.196e-15  -3.17855  -2.44769
##       .25       .50       .75       .90       .95
##  -1.44908   0.03408   1.29520   2.54635   3.17658
##
## lowest : -5.007 -4.874 -4.638 -4.277 -4.272
## highest:  4.586  4.637  4.763  5.183  5.433
## ---------------------------------------------------------------------------
## student_teacher_ratio
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     413       1   19.64   16.43   17.35   18.58   19.72
##     .75     .90     .95
##   20.87   21.87   22.63
##
## lowest : 14.00 14.20 14.54 14.71 15.14
## highest: 24.89 24.95 25.05 25.79 25.80
## ---------------------------------------------------------------------------
## computer_student_ratio
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     412       1  0.1359 0.05471 0.06654 0.09377 0.12546
##     .75     .90     .95
## 0.16447 0.22494 0.24906
##
## lowest : 0.00000 0.01455 0.02266 0.02548 0.04167
## highest: 0.32770 0.34359 0.34979 0.35897 0.42083
## ---------------------------------------------------------------------------
## students_log
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     391       1   6.986   4.941   5.100   5.938   6.857
##     .75     .90     .95
##   8.009   8.871   9.245
##
## lowest :  4.394  4.522  4.615  4.635  4.644
## highest:  9.873  9.949  9.968 10.133 10.210
## ---------------------------------------------------------------------------
## income_log
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##     420       0     337       1   2.645   2.048   2.189   2.365   2.619
##     .75     .90     .95
##   2.870   3.125   3.422
##
## lowest : 1.674 1.740 1.827 1.884 1.889
## highest: 3.731 3.767 3.911 3.925 4.013
## ---------------------------------------------------------------------------
```

Bivariate correlations

```r
cor(cas[ , v$all_variables])
```

```
##                           calworks       lunch expenditure     english
## calworks                1.00000000  0.73942180  0.06788857  0.31957593
## lunch                   0.73942180  1.00000000 -0.06103871  0.65306072
## expenditure             0.06788857 -0.06103871  1.00000000 -0.07139604
## english                 0.31957593  0.65306072 -0.07139604  1.00000000
## student_teacher_ratio   0.01827610  0.13520340 -0.61998216  0.18764237
## computer_student_ratio -0.15196751 -0.20395342  0.28655958 -0.25100695
## students_log            0.07597949  0.08926736 -0.15718872  0.37765895
## income_log             -0.56870132 -0.76388309  0.25113384 -0.38512630
## performance            -0.62697238 -0.86780205  0.19015943 -0.64197938
##                        student_teacher_ratio computer_student_ratio
## calworks                           0.0182761             -0.1519675
## lunch                              0.1352034             -0.2039534
## expenditure                       -0.6199822              0.2865596
## english                            0.1876424             -0.2510070
## student_teacher_ratio              1.0000000             -0.3070702
## computer_student_ratio            -0.3070702              1.0000000
## students_log                       0.3310482             -0.3352406
## income_log                        -0.1896905              0.1593155
## performance                       -0.2254616              0.2701315
##                        students_log income_log performance
## calworks                 0.07597949 -0.5687013  -0.6269724
## lunch                    0.08926736 -0.7638831  -0.8678020
## expenditure             -0.15718872  0.2511338   0.1901594
## english                  0.37765895 -0.3851263  -0.6419794
## student_teacher_ratio    0.33104818 -0.1896905  -0.2254616
## computer_student_ratio  -0.33524063  0.1593155   0.2701315
## students_log             1.00000000  0.1486931  -0.1206251
## income_log               0.14869307  1.0000000   0.7496733
## performance             -0.12062512  0.7496733   1.0000000
```

```r
round(cor(cas[ , v$all_variables]), 2) # round to 2 decimal places
```

```
##                        calworks lunch expenditure english
## calworks                   1.00  0.74        0.07    0.32
## lunch                      0.74  1.00       -0.06    0.65
## expenditure                0.07 -0.06        1.00   -0.07
## english                    0.32  0.65       -0.07    1.00
## student_teacher_ratio      0.02  0.14       -0.62    0.19
## computer_student_ratio    -0.15 -0.20        0.29   -0.25
## students_log               0.08  0.09       -0.16    0.38
## income_log                -0.57 -0.76        0.25   -0.39
## performance               -0.63 -0.87        0.19   -0.64
##                        student_teacher_ratio computer_student_ratio
## calworks                                0.02                  -0.15
## lunch                                   0.14                  -0.20
## expenditure                            -0.62                   0.29
```

```
## english                                   0.19                    -0.25
## student_teacher_ratio                     1.00                    -0.31
## computer_student_ratio                   -0.31                     1.00
## students_log                              0.33                    -0.34
## income_log                               -0.19                     0.16
## performance                              -0.23                     0.27
##                        students_log income_log performance
## calworks                       0.08      -0.57       -0.63
## lunch                          0.09      -0.76       -0.87
## expenditure                   -0.16       0.25        0.19
## english                        0.38      -0.39       -0.64
## student_teacher_ratio          0.33      -0.19       -0.23
## computer_student_ratio        -0.34       0.16        0.27
## students_log                   1.00       0.15       -0.12
## income_log                     0.15       1.00        0.75
## performance                   -0.12       0.75        1.00
```

```r
rp <- Hmisc::rcorr(as.matrix(cas[,v$all_variables])) # significance test on correlations
rp
```

```
##                        calworks lunch expenditure english
## calworks                   1.00  0.74        0.07    0.32
## lunch                      0.74  1.00       -0.06    0.65
## expenditure                0.07 -0.06        1.00   -0.07
## english                    0.32  0.65       -0.07    1.00
## student_teacher_ratio      0.02  0.14       -0.62    0.19
## computer_student_ratio    -0.15 -0.20        0.29   -0.25
## students_log               0.08  0.09       -0.16    0.38
## income_log                -0.57 -0.76        0.25   -0.39
## performance               -0.63 -0.87        0.19   -0.64
##                        student_teacher_ratio computer_student_ratio
## calworks                                0.02                  -0.15
## lunch                                   0.14                  -0.20
## expenditure                            -0.62                   0.29
## english                                 0.19                  -0.25
## student_teacher_ratio                   1.00                  -0.31
## computer_student_ratio                 -0.31                   1.00
## students_log                            0.33                  -0.34
## income_log                             -0.19                   0.16
## performance                            -0.23                   0.27
##                        students_log income_log performance
## calworks                       0.08      -0.57       -0.63
## lunch                          0.09      -0.76       -0.87
## expenditure                   -0.16       0.25        0.19
## english                        0.38      -0.39       -0.64
## student_teacher_ratio          0.33      -0.19       -0.23
## computer_student_ratio        -0.34       0.16        0.27
## students_log                   1.00       0.15       -0.12
## income_log                     0.15       1.00        0.75
## performance                   -0.12       0.75        1.00
##
## n= 420
##
##
```

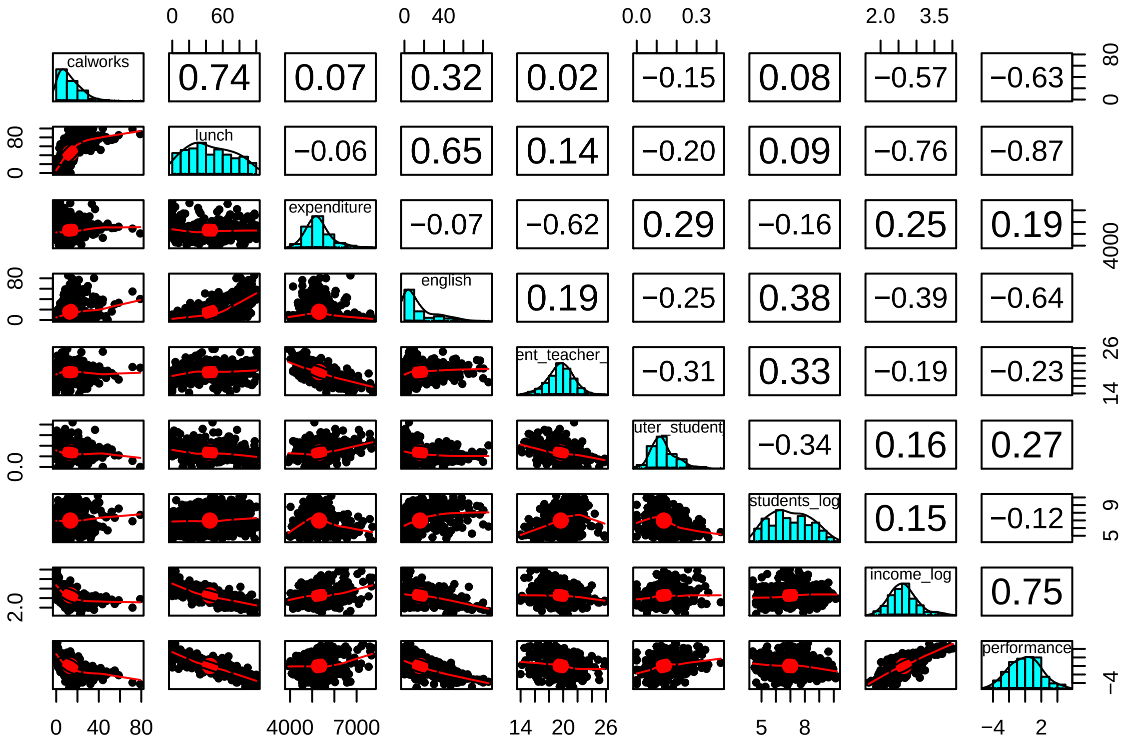
```
## P
##                       calworks lunch  expenditure english
## calworks                       0.0000 0.1649      0.0000
## lunch                 0.0000          0.2119      0.0000
## expenditure           0.1649   0.2119             0.1441
## english               0.0000   0.0000 0.1441
## student_teacher_ratio  0.7088   0.0055 0.0000      0.0001
## computer_student_ratio 0.0018   0.0000 0.0000      0.0000
## students_log          0.1200   0.0676 0.0012      0.0000
## income_log            0.0000   0.0000 0.0000      0.0000
## performance           0.0000   0.0000 0.0000      0.0000
##                       student_teacher_ratio computer_student_ratio
## calworks              0.7088                0.0018
## lunch                 0.0055                0.0000
## expenditure           0.0000                0.0000
## english               0.0001                0.0000
## student_teacher_ratio                       0.0000
## computer_student_ratio 0.0000
## students_log          0.0000                0.0000
## income_log            0.0000                0.0011
## performance           0.0000                0.0000
##                       students_log income_log performance
## calworks              0.1200       0.0000     0.0000
## lunch                 0.0676       0.0000     0.0000
## expenditure           0.0012       0.0000     0.0000
## english               0.0000       0.0000     0.0000
## student_teacher_ratio  0.0000       0.0000     0.0000
## computer_student_ratio 0.0000       0.0011     0.0000
## students_log                       0.0022     0.0134
## income_log            0.0022                  0.0000
## performance           0.0134       0.0000
```

```r
ifelse(rp$P < .05, "*", "")
```

```
##                       calworks lunch expenditure english
## calworks              NA       "*"   ""          "*"
## lunch                 "*"      NA    ""          "*"
## expenditure           ""       ""    NA          ""
## english               "*"      "*"   ""          NA
## student_teacher_ratio  ""       "*"   "*"         "*"
## computer_student_ratio "*"      "*"   "*"         "*"
## students_log          ""       ""    "*"         "*"
## income_log            "*"      "*"   "*"         "*"
## performance           "*"      "*"   "*"         "*"
##                       student_teacher_ratio computer_student_ratio
## calworks              ""                    "*"
## lunch                 "*"                   "*"
## expenditure           "*"                   "*"
## english               "*"                   "*"
## student_teacher_ratio  NA                    "*"
## computer_student_ratio "*"                   NA
## students_log          "*"                   "*"
## income_log            "*"                   "*"
## performance           "*"                   "*"
```

```
##                         students_log income_log performance
## calworks                ""           "*"        "*"
## lunch                   ""           "*"        "*"
## expenditure             "*"          "*"        "*"
## english                 "*"          "*"        "*"
## student_teacher_ratio   "*"          "*"        "*"
## computer_student_ratio  "*"          "*"        "*"
## students_log            NA           "*"        "*"
## income_log              "*"          NA         "*"
## performance             "*"          "*"        NA
```

```r
# Scatterplot matrix with correlations
pairs.panels(cas[ , v$all_variables])
```



Regression models

```r
# By default, you don't get much output
lm(performance ~ expenditure + calworks + lunch, data = cas)
```

```
##
## Call:
## lm(formula = performance ~ expenditure + calworks + lunch, data = cas)
##
## Coefficients:
## (Intercept)  expenditure     calworks        lunch
##   0.5108774    0.0004267   -0.0003359   -0.0620302
```

```r
# You need to save the model to an object
fit <- lm(performance ~ expenditure + calworks + lunch, cas)

# this object stores the results of analyses.
# You can extract elements directly from this object
str(fit) # show the structure of the object
```

```
## List of 12
##  $ coefficients : Named num [1:4] 0.510877 0.000427 -0.000336 -0.06203
##   ..- attr(*, "names")= chr [1:4] "(Intercept)" "expenditure" "calworks" "lunch"
##  $ residuals    : Named num [1:420] 0.668 1.022 0.836 0.573 0.759 ...
##   ..- attr(*, "names")= chr [1:420] "1" "2" "3" "4" ...
##  $ effects      : Named num [1:420] -1.69e-14 7.63 2.57e+01 2.29e+01 6.55e-01 ...
##   ..- attr(*, "names")= chr [1:420] "(Intercept)" "expenditure" "calworks" "lunch" ...
##  $ rank         : int 4
##  $ fitted.values: Named num [1:420] 3.108 -0.291 -1.894 -1.251 -2.131 ...
##   ..- attr(*, "names")= chr [1:420] "1" "2" "3" "4" ...
##  $ assign       : int [1:4] 0 1 2 3
##  $ qr           :List of 5
##   ..$ qr   : num [1:420, 1:4] -20.4939 0.0488 0.0488 0.0488 0.0488 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:420] "1" "2" "3" "4" ...
##   .. .. ..$ : chr [1:4] "(Intercept)" "expenditure" "calworks" "lunch"
##   .. ..- attr(*, "assign")= int [1:4] 0 1 2 3
##   ..$ qraux: num [1:4] 1.05 1.02 1.18 1.01
##   ..$ pivot: int [1:4] 1 2 3 4
##   ..$ tol  : num 1e-07
##   ..$ rank : int 4
##   ..- attr(*, "class")= chr "qr"
##  $ df.residual  : int 416
##  $ xlevels      : Named list()
##  $ call         : language lm(formula = performance ~ expenditure + calworks + lunch, data = cas)
##  $ terms        :Classes 'terms', 'formula'  language performance ~ expenditure + calworks + lunch
##   .. ..- attr(*, "variables")= language list(performance, expenditure, calworks, lunch)
##   .. ..- attr(*, "factors")= int [1:4, 1:3] 0 1 0 0 0 0 1 0 0 0 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:4] "performance" "expenditure" "calworks" "lunch"
##   .. .. .. ..$ : chr [1:3] "expenditure" "calworks" "lunch"
##   .. ..- attr(*, "term.labels")= chr [1:3] "expenditure" "calworks" "lunch"
##   .. ..- attr(*, "order")= int [1:3] 1 1 1
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(performance, expenditure, calworks, lunch)
##   .. ..- attr(*, "dataClasses")= Named chr [1:4] "numeric" "numeric" "numeric" "numeric"
##   .. .. ..- attr(*, "names")= chr [1:4] "performance" "expenditure" "calworks" "lunch"
##  $ model        :'data.frame':   420 obs. of  4 variables:
##   ..$ performance: num [1:420] 3.776 0.731 -1.059 -0.678 -1.372 ...
##   ..$ expenditure: num [1:420] 6385 5099 5502 7102 5236 ...
##   ..$ calworks   : num [1:420] 0.51 15.42 55.03 36.48 33.11 ...
##   ..$ lunch      : num [1:420] 2.04 47.92 76.32 77.05 78.43 ...
##   ..- attr(*, "terms")=Classes 'terms', 'formula'  language performance ~ expenditure + calworks + lu
##   .. .. ..- attr(*, "variables")= language list(performance, expenditure, calworks, lunch)
```

```
##   .. .. ..- attr(*, "factors")= int [1:4, 1:3] 0 1 0 0 0 0 1 0 0 0 ...
##   .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. ..$ : chr [1:4] "performance" "expenditure" "calworks" "lunch"
##   .. .. .. .. ..$ : chr [1:3] "expenditure" "calworks" "lunch"
##   .. .. ..- attr(*, "term.labels")= chr [1:3] "expenditure" "calworks" "lunch"
##   .. .. ..- attr(*, "order")= int [1:3] 1 1 1
##   .. .. ..- attr(*, "intercept")= int 1
##   .. .. ..- attr(*, "response")= int 1
##   .. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. .. ..- attr(*, "predvars")= language list(performance, expenditure, calworks, lunch)
##   .. .. ..- attr(*, "dataClasses")= Named chr [1:4] "numeric" "numeric" "numeric" "numeric"
##   .. .. .. ..- attr(*, "names")= chr [1:4] "performance" "expenditure" "calworks" "lunch"
##  - attr(*, "class")= chr "lm"
```

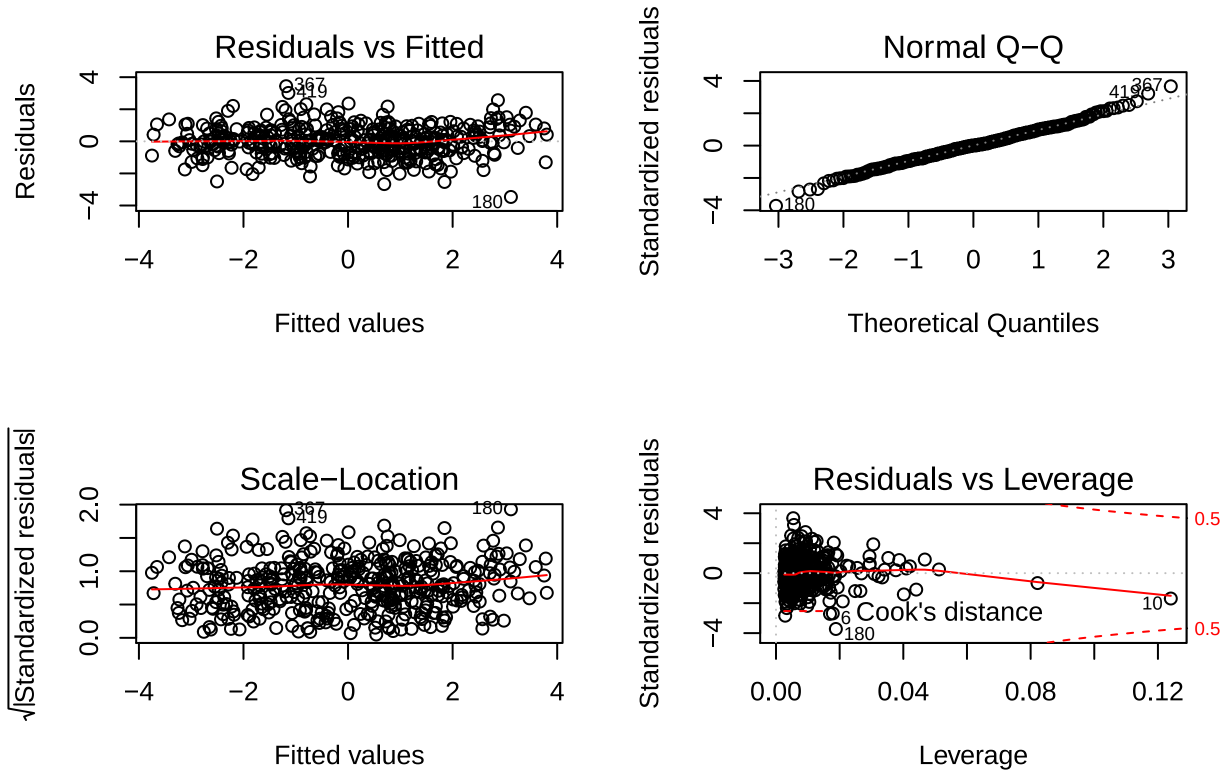
```
fit$coefficients
```

```
##   (Intercept)   expenditure      calworks         lunch
##  0.5108773871  0.0004266699 -0.0003358747 -0.0620301538
```

```
# But more commonly you apply a set of "methods"
summary(fit) # summary info
```

```
##
## Call:
## lm(formula = performance ~ expenditure + calworks + lunch, data = cas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4663 -0.5953  0.0060  0.6150  3.4391
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.109e-01  4.062e-01   1.258    0.209
## expenditure  4.267e-04  7.361e-05   5.796 1.34e-08 ***
## calworks    -3.359e-04  6.040e-03  -0.056    0.956
## lunch       -6.203e-02  2.550e-03 -24.330   < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9398 on 416 degrees of freedom
## Multiple R-squared:  0.772,  Adjusted R-squared:  0.7703
## F-statistic: 469.4 on 3 and 416 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
plot(fit)
```




```r
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: performance
##              Df Sum Sq Mean Sq F value    Pr(>F)
## expenditure   1  58.27   58.27   65.97  5.26e-15 ***
## calworks      1 662.84  662.84  750.44 < 2.2e-16 ***
## lunch         1 522.85  522.85  591.94 < 2.2e-16 ***
## Residuals   416 367.44    0.88
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
inf <- influence.measures(fit) # various influence and outlier measures

confint(fit) # confidence intervals on coeficients
```

```
##                      2.5 %        97.5 %
## (Intercept) -0.2876649944  1.3094197685
## expenditure  0.0002819765  0.0005713632
## calworks    -0.0122078876  0.0115361382
## lunch       -0.0670417523 -0.0570185553
```

```r
# You can create plots yourself
# Check normality and homoscedsaticity of residuals
# plot predicted by residuals
plot(predict(fit), residuals(fit))
```

```r
abline(h=0)

# standardised coefficients
library(QuantPsyc)
```

```
## Loading required package: boot
##
## Attaching package: 'boot'
##
## The following object is masked from 'package:car':
##
##     logit
##
## The following object is masked from 'package:survival':
##
##     aml
##
## The following object is masked from 'package:lattice':
##
##     melanoma
##
## The following object is masked from 'package:psych':
##
##     logit
##
##
## Attaching package: 'QuantPsyc'
##
## The following object is masked from 'package:base':
##
##     norm
```

```r
QuantPsyc::lm.beta(fit)
```

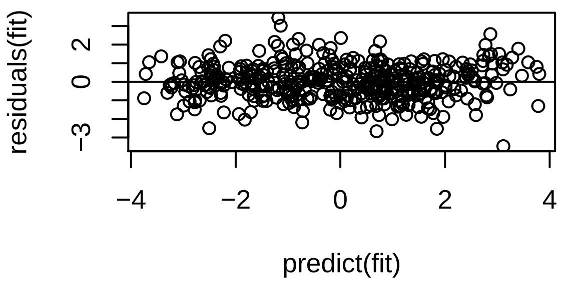
```
##  expenditure     calworks        lunch
##  0.137925516 -0.001961878 -0.857932597
```

```r
fit_standardised <- lm(scale(performance) ~ scale(expenditure) + scale(calworks) + scale(lunch), cas)
summary(fit_standardised)
```

```
##
## Call:
## lm(formula = scale(performance) ~ scale(expenditure) + scale(calworks) +
##     scale(lunch), data = cas)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.76754 -0.30356  0.00306  0.31362  1.75369
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)         1.569e-17  2.338e-02   0.000    1.000    
## scale(expenditure)  1.379e-01  2.380e-02   5.796 1.34e-08 ***
## scale(calworks)    -1.962e-03  3.528e-02  -0.056    0.956    
## scale(lunch)       -8.579e-01  3.526e-02 -24.330  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4792 on 416 degrees of freedom
## Multiple R-squared:  0.772,  Adjusted R-squared:  0.7703 
## F-statistic: 469.4 on 3 and 416 DF,  p-value: < 2.2e-16
```

```r
# more information on regression diagnostics
# http://www.statmethods.net/stats/rdiagnostics.html
```



Comparing regression models

```r
# model 1 include poverty variables
v$predictors
```

```
## [1] "calworks"                "lunch"                  
## [3] "expenditure"             "english"                
## [5] "student_teacher_ratio"   "computer_student_ratio" 
## [7] "students_log"            "income_log"
```

```r
fit1 <- lm(performance ~ calworks + lunch + expenditure + income_log, cas)
# Model 2 adds school features
fit2 <- lm(performance ~ calworks + lunch + expenditure + income_log + 
               student_teacher_ratio + students_log + 
               computer_student_ratio, cas)

summary(fit1)
```

```
## 
## Call:
## lm(formula = performance ~ calworks + lunch + expenditure + income_log, 
##     data = cas)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3.3949 -0.5867 -0.0192  0.5470  3.3424
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.3975377  0.6062637  -2.305   0.0216 *
## calworks     0.0013168  0.0059369   0.222   0.8246
## lunch       -0.0540187  0.0031516 -17.140  < 2e-16 ***
## expenditure  0.0003235  0.0000763   4.240 2.75e-05 ***
## income_log   0.7850026  0.1879584   4.176 3.61e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9218 on 415 degrees of freedom
## Multiple R-squared:  0.7812, Adjusted R-squared:  0.7791
## F-statistic: 370.4 on 4 and 415 DF,  p-value: < 2.2e-16
```

```r
summary(fit2)
```

```
##
## Call:
## lm(formula = performance ~ calworks + lunch + expenditure + income_log +
##     student_teacher_ratio + students_log + computer_student_ratio,
##     data = cas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6805 -0.5905  0.0154  0.5004  2.9066
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -6.466e-01  1.051e+00  -0.615   0.5388
## calworks                2.989e-03  5.882e-03   0.508   0.6116
## lunch                  -5.076e-02  3.250e-03 -15.621  < 2e-16 ***
## expenditure             1.758e-04  9.578e-05   1.836   0.0671 .
## income_log              1.021e+00  2.041e-01   5.000 8.48e-07 ***
## student_teacher_ratio  -2.238e-02  3.169e-02  -0.706   0.4804
## students_log           -7.825e-02  3.900e-02  -2.007   0.0454 *
## computer_student_ratio  1.683e+00  7.650e-01   2.200   0.0284 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.909 on 412 degrees of freedom
## Multiple R-squared:  0.7888, Adjusted R-squared:  0.7852
## F-statistic: 219.8 on 7 and 412 DF,  p-value: < 2.2e-16
```

```r
# Does second model explain significantly more variance?
anova(fit1, fit2)
```

```
## Analysis of Variance Table
##
## Model 1: performance ~ calworks + lunch + expenditure + income_log
## Model 2: performance ~ calworks + lunch + expenditure + income_log + student_teacher_ratio +
##     students_log + computer_student_ratio
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
```

```
## 1    415 352.62
## 2    412 340.40  3    12.217 4.9287 0.00225 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Formula notation

```r
# For teaching purposes let's name the variables in a general way
x <- cas[, c("performance", "student_teacher_ratio", "students_log", "income_log")]
head(x)
```

```
##   performance student_teacher_ratio students_log income_log
## 1   3.7762622              17.88991     5.273000   3.121924
## 2   0.7312842              21.52466     5.480639   2.284828
## 3  -1.0587539              18.69723     7.346010   2.194777
## 4  -0.6775202              17.35714     5.493061   2.194777
## 5  -1.3717659              18.67133     7.196687   2.206111
## 6  -5.0066595              21.40625     4.919981   2.343247
```

```r
names(x)  <- c("dv", "A", "B", "C")
head(x)
```

```
##           dv        A        B        C
## 1  3.7762622 17.88991 5.273000 3.121924
## 2  0.7312842 21.52466 5.480639 2.284828
## 3 -1.0587539 18.69723 7.346010 2.194777
## 4 -0.6775202 17.35714 5.493061 2.194777
## 5 -1.3717659 18.67133 7.196687 2.206111
## 6 -5.0066595 21.40625 4.919981 2.343247
```

```r
# Overview
?formula
# http://faculty.chicagobooth.edu/richard.hahn/teaching/FormulaNotation.pdf

# 1 intercept
# -1 exclude intercept
# The intercept is included by default in linear models,
# but in other contexts you need to specify it.

lm(dv ~ A, x) # intercept included by default
```

```
##
## Call:
## lm(formula = dv ~ A, data = x)
##
## Coefficients:
## (Intercept)            A
##      4.5903      -0.2337
```

```r
lm(dv ~ 1 + A, x) # intercept explicitly included (same as above)
```

```
##
## Call:
## lm(formula = dv ~ 1 + A, data = x)
##
## Coefficients:
## (Intercept)            A
##      4.5903      -0.2337
```

```r
lm(dv ~ -1 + A, x) # exclude intercept
```

```
##
## Call:
## lm(formula = dv ~ -1 + A, data = x)
##
## Coefficients:
##         A
## -0.002143
```

```r
# + main effect
lm(dv ~ A + B, x) # main effect of A and B
```

```
##
## Call:
## lm(formula = dv ~ A + B, data = x)
##
## Coefficients:
## (Intercept)            A            B
##     4.75670     -0.21599     -0.07365
```

```r
# * include interaction and main effects
# : just main effect without interactions
lm(dv ~ A * B, x) # main effects and interactions
```

```
##
## Call:
## lm(formula = dv ~ A * B, data = x)
##
## Coefficients:
## (Intercept)            A            B          A:B
##    -6.93585      0.36541      1.76101     -0.09085
```

```r
lm(dv ~ A:B, x) # no main effects but interaction
```

```
##
## Call:
## lm(formula = dv ~ A:B, data = x)
##
## Coefficients:
## (Intercept)          A:B
##     1.50395     -0.01089
```

```r
lm(dv ~ A + B + A:B, x) # main effects explicitly specified
```

```
##
## Call:
## lm(formula = dv ~ A + B + A:B, data = x)
##
## Coefficients:
## (Intercept)            A            B          A:B
##    -6.93585      0.36541      1.76101     -0.09085
```

```r
lm(dv ~ A*B*C, x) # main effects, two-way interactions, three-way interaction
```

```
##
## Call:
## lm(formula = dv ~ A * B * C, data = x)
##
## Coefficients:
## (Intercept)            A            B            C          A:B
##   -15.23512      0.58174      1.15797      5.65525     -0.10694
##         A:C          B:C        A:B:C
##    -0.17159     -0.40625      0.03268
```

```r
lm(dv ~ (A + B + C)^3, x) # main as above
```

```
##
## Call:
## lm(formula = dv ~ (A + B + C)^3, data = x)
##
## Coefficients:
## (Intercept)            A            B            C          A:B
##   -15.23512      0.58174      1.15797      5.65525     -0.10694
##         A:C          B:C        A:B:C
##    -0.17159     -0.40625      0.03268
```

```r
lm(dv ~ (A + B + C)^2, x) # main effects but only two-way interactions
```

```
##
## Call:
## lm(formula = dv ~ (A + B + C)^2, data = x)
##
## Coefficients:
## (Intercept)            A            B            C          A:B
##  -3.838e+00   -9.812e-05   -5.595e-01    1.371e+00   -1.971e-02
##         A:C          B:C
##   4.823e-02    2.342e-01
```

```r
# You can apply transformations to variables in place
lm(dv ~ scale(A), x) # main effects but only two-way interactions
```

```
##
## Call:
## lm(formula = dv ~ scale(A), data = x)
##
## Coefficients:
## (Intercept)     scale(A)
##   7.516e-16   -4.421e-01
```

```
# this is the same as creating a new variable
# and using he new variable in the model
x$zA <- scale(x$A)
lm(dv ~ zA, x)
```

```
##
## Call:
## lm(formula = dv ~ zA, data = x)
##
## Coefficients:
## (Intercept)           zA
##   7.516e-16   -4.421e-01
```

```
# However if the transformation involves symbols that
# have special meaning in the context of R formulas
# i.e., +, -, *, ^, |, :
# then you  # have to wrap it in the I()
# I stands for Inhibit Interpretation or AsIs

# Polynomial regression
lm(dv ~ A + I(A^2), x) # include quadratic effect of A
```

```
##
## Call:
## lm(formula = dv ~ A + I(A^2), data = x)
##
## Coefficients:
## (Intercept)            A       I(A^2)
##     8.76464     -0.66330      0.01095
```

```
lm(dv ~ A + I(A^2) + I(A^3), x) # include quadratic and cubic effect of A
```

```
##
## Call:
## lm(formula = dv ~ A + I(A^2) + I(A^3), data = x)
##
## Coefficients:
## (Intercept)            A       I(A^2)       I(A^3)
##  -55.127071     9.231085    -0.493990     0.008495
```

```
# interaction effects with centering
lm(dv ~ A + B + I(scale(A) * scale(B)), x) # z-score centre before creating interaction
```

```
##
## Call:
## lm(formula = dv ~ A + B + I(scale(A) * scale(B)), data = x)
##
## Coefficients:
##           (Intercept)                      A                      B
##               5.52963               -0.26928               -0.02331
## I(scale(A) * scale(B))
##              -0.23636
```

```
# composites
lm(dv ~ I(A + B), x) # include the sum of two variables as a predictor
```

```
##
## Call:
## lm(formula = dv ~ I(A + B), data = x)
##
## Coefficients:
## (Intercept)     I(A + B)
##      4.3007      -0.1615
```

```
lm(dv ~ I(2 * A + 5 * B), x) # include the weighted coposte as a predictor
```

```
##
## Call:
## lm(formula = dv ~ I(2 * A + 5 * B), data = x)
##
## Coefficients:
##      (Intercept)  I(2 * A + 5 * B)
##          3.10648          -0.04186
```

R Factors: Categorical predictors

```
# Factors can be used for categorical variables

# http://www.ats.ucla.edu/stat/r/modules/factor_variables.htm
library(MASS)
data(survey)
csurvey <- na.omit(survey)
# let's assume a few variables were string variables
csurvey$Sex_character <- as.character(csurvey$Sex)
csurvey$Smoke_character <- as.character(csurvey$Smoke)

# by default character variables will be converted to factors in regression models
lm(Height ~ Sex_character, csurvey)
```

```
##
## Call:
## lm(formula = Height ~ Sex_character, data = csurvey)
```

```
##
## Coefficients:
##      (Intercept)  Sex_characterMale
##           165.60              13.75
```

```r
# by default it performs dummy coding with the first category as the reference category
# By default the ordering of a categorical variable is alphabetical

# levels shows the levels of a factor variable
# Thus, if we convert a sex as a character variable to a factor
# F is before M to it is Female then Male

csurvey$Sex_factor <-  factor(csurvey$Sex_character)
levels(csurvey$Sex_factor)
```

```
## [1] "Female" "Male"
```

```r
lm(Height ~ Sex_factor, csurvey)
```

```
##
## Call:
## lm(formula = Height ~ Sex_factor, data = csurvey)
##
## Coefficients:
##   (Intercept)  Sex_factorMale
##        165.60           13.75
```

```r
# Factors also influence the ordering of categorical variables
# in plots
par(mfrow=c(2,1))
plot(Height ~ Sex_factor, csurvey)
# and the order in tables
table(csurvey$Sex_factor)
```

```
##
## Female   Male
##     84     84
```

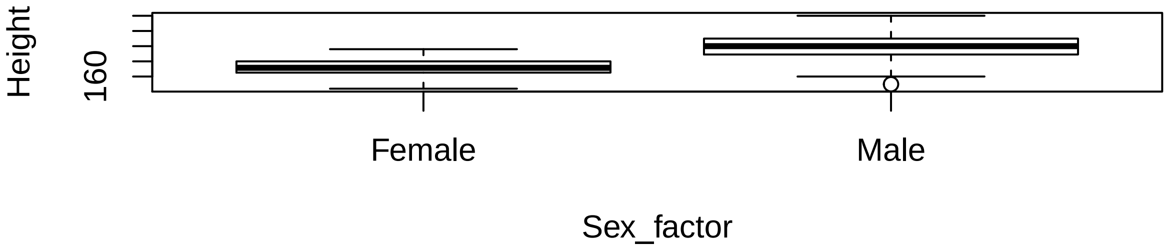
```r
# If we wanted to change the order to Male then Female
csurvey$Sex_factor <- factor(csurvey$Sex_character, levels = c("Male", "Female"))
levels(csurvey$Sex_factor)
```

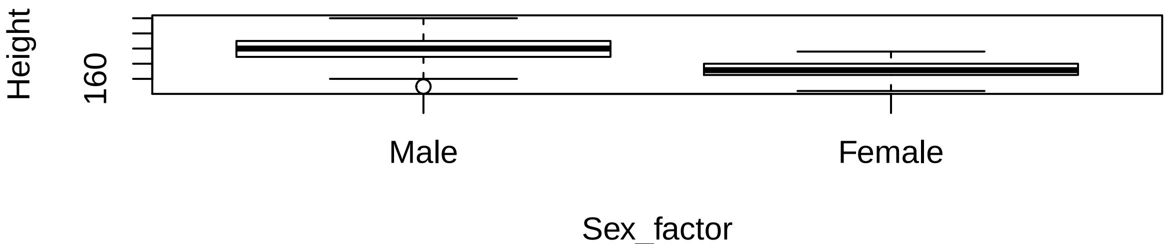
```
## [1] "Male"   "Female"
```

```r
lm(Height ~ Sex_factor, csurvey) # now male is the reference category
```

```
##
## Call:
## lm(formula = Height ~ Sex_factor, data = csurvey)
##
## Coefficients:
##     (Intercept)  Sex_factorFemale
##          179.35            -13.75
```

```r
plot(Height ~ Sex_factor, csurvey)
```






```r
table(csurvey$Sex_factor)
```

```
##
##   Male Female
##     84     84
```

```r
# Ordered factors
# Factors
# some factors reflect an ordinal relationship
# e.g., survey frequency-agreement type scales
# For example, see this smoking frequency items
csurvey$Smoke_factor <- factor(csurvey$Smoke)
table(csurvey$Smoke_factor)
```

```
##
## Heavy Never Occas Regul
##     7   134    13    14
```

```r
# By default it is in the wrong order
csurvey$Smoke_factor <- factor(csurvey$Smoke, c("Never", "Occas", "Regul", "Heavy"))
table(csurvey$Smoke_factor)
```

```
##
## Never Occas Regul Heavy
##   134    13    14     7
```

```r
# However, we can also influence the type of contrasts performed
csurvey$Smoke_ordered <- factor(csurvey$Smoke, c("Never", "Occas", "Regul", "Heavy"),
                                ordered = TRUE)
# or equivalently
csurvey$Smoke_ordered <- ordered(csurvey$Smoke, c("Never", "Occas", "Regul", "Heavy"))

# When included in linear model, we get
# polynomial contrasts for ordered factors
lm(Pulse ~ Smoke_ordered, csurvey)
```

```
##
## Call:
## lm(formula = Pulse ~ Smoke_ordered, data = csurvey)
##
## Coefficients:
##     (Intercept)  Smoke_ordered.L  Smoke_ordered.Q  Smoke_ordered.C
##          75.265            4.092            1.436           -1.974
```

```r
# Many data import functions have the option of
# importing string variables as characters or factors
# Some use a general configuration option:
opt <- options()
opt$stringsAsFactors
```

```
## [1] FALSE
```

```r
# e.g.,
# read.table(..., stringsAsFactors = ...)
# read.csv(..., stringsAsFactors = ...)

# other functions have explicit options to import as factors
# foreign::read.spss(..., use.value.labels = ...

# Tip: My preference is to import string variables as character variables
# If I want to use factors I prefer to explicitly create them.
```

Exercise 1

```r
library(AER)
help(package = AER)
data("GSS7402")
?GSS7402 # to learn about the dataset
# It might be easier to work with a shorter variable name

# 1. Run a t-test on whether participants who lived in a city
#    at age 16 (i.e, city16) have more or less education
#    than those those who did not

# 2. Get correlations between education, number of kids (kids)
```

```
#    year, and number of siblings (siblings)

# 3. Run a multiple regresion predicting education from
#    year, kids, and siblings.
# 3.1 Run the model and save the fit

# 3.2 Get a summary of the results

# 3.3 the standardised coefficients

# 3.4 Check whether the residuals are normally distributed

# 3.5 Plot predicted values by residuals


# 4. Factors
# 4.1 create a table of values for ethnicity

# 4.2 Run a regression predicting education from ethnicity

# 4.3 Make a new factor variable where cauc is the reference value
#     and check that this worked by running a regression with
#     this new ethncity variable as the predictor.


# 5. Comparing models
# 5.1 Fit a model predicting education from
#     (a) year and siblings
#     (b) year, siblings, and the interaction
#     and compare the fit of these two models
```

Answers 1

```
library(AER)
help(package = AER)
data("GSS7402")
?GSS7402 # to learn about the dataset
# It might be easier to work with a shorter variable name
gss <- GSS7402

# 1. Run a t-test on whether participants who lived in a city
#    at age 16 (i.e, city16) have more or less education
#    than those those who did not
t.test(education ~ city16, gss)
```

```
##
##  Welch Two Sample t-test
##
## data:  education by city16
## t = -18.4921, df = 8832.927, p-value < 2.2e-16
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.234686 -0.998011
## sample estimates:
##  mean in group no mean in group yes
##          12.16088          13.27723
```

```r
# 2. Get correlations between education, number of kids (kids)
#    year, and number of siblings (siblings)
cor( gss[ ,c("education", "kids", "year", "siblings")])
```

```
##            education        kids        year    siblings
## education  1.0000000 -0.29051084  0.21216834 -0.29060307
## kids      -0.2905108  1.00000000 -0.08267769  0.18001462
## year       0.2121683 -0.08267769  1.00000000 -0.07925257
## siblings  -0.2906031  0.18001462 -0.07925257  1.00000000
```

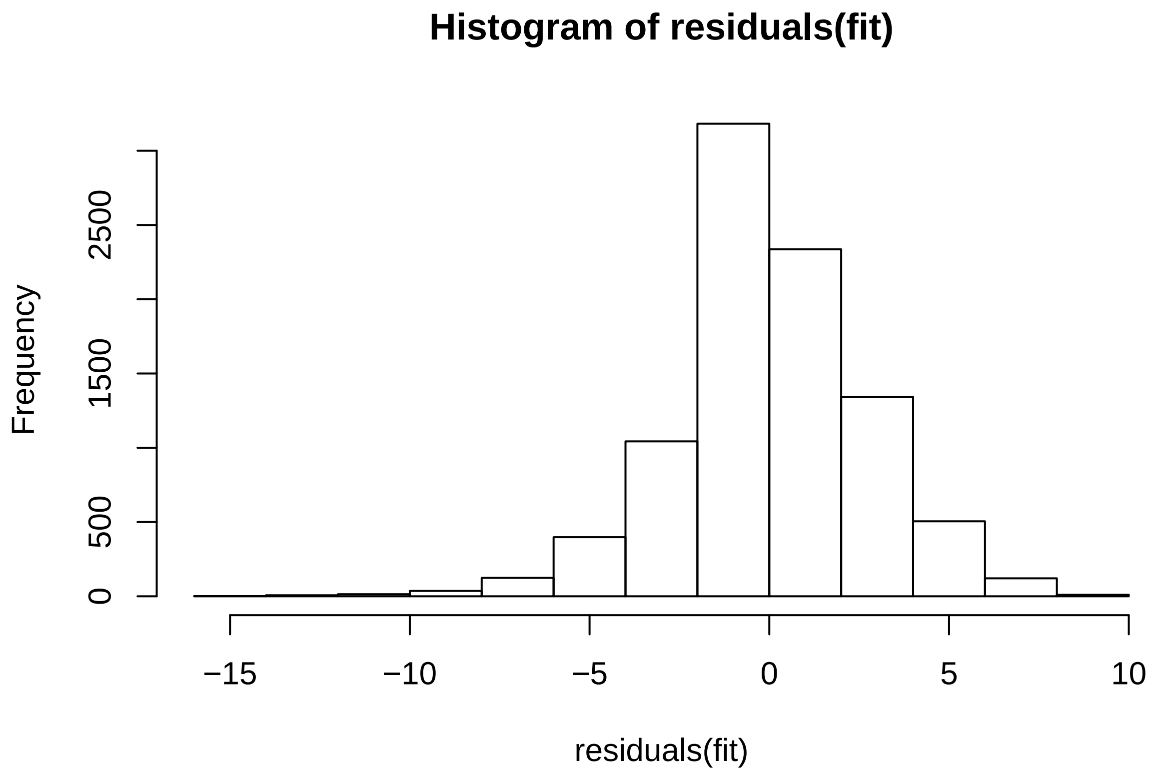
```r
# 3. Run a multiple regression predicting education from
#    year, kids, and siblings.
# 3.1 Run the model and save the fit
fit <- lm(education ~ year + kids + siblings, gss)

# 3.2 Get a summary of the results
summary(fit)
```

```
##
## Call:
## lm(formula = education ~ year + kids + siblings, data = gss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.5055  -1.5182  -0.1563   1.6827   9.6598
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -98.356468   6.197245  -15.87   <2e-16 ***
## year          0.056601   0.003111   18.19   <2e-16 ***
## kids         -0.382855   0.015890  -24.09   <2e-16 ***
## siblings     -0.213661   0.008833  -24.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.688 on 9116 degrees of freedom
## Multiple R-squared:  0.1731, Adjusted R-squared:  0.1728
## F-statistic: 636.1 on 3 and 9116 DF,  p-value: < 2.2e-16
```

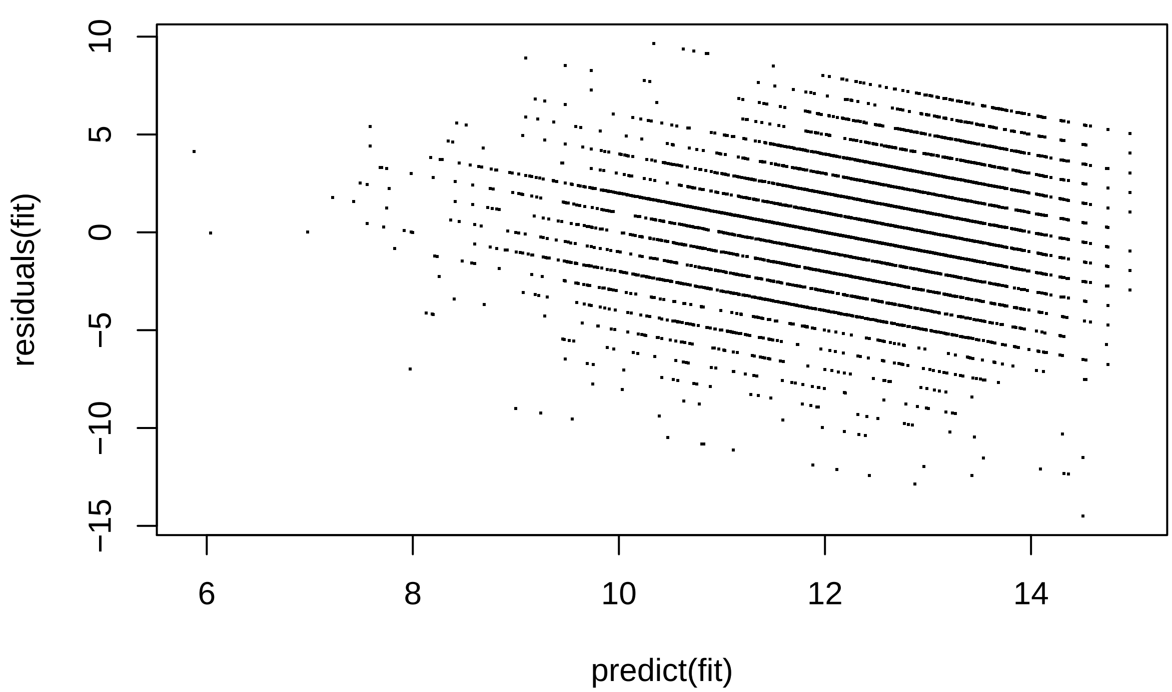
```r
# 3.3 the standardised coefficients
QuantPsyc::lm.beta(fit)
```

```
##       year       kids   siblings
##  0.1742333 -0.2338567 -0.2346970
```

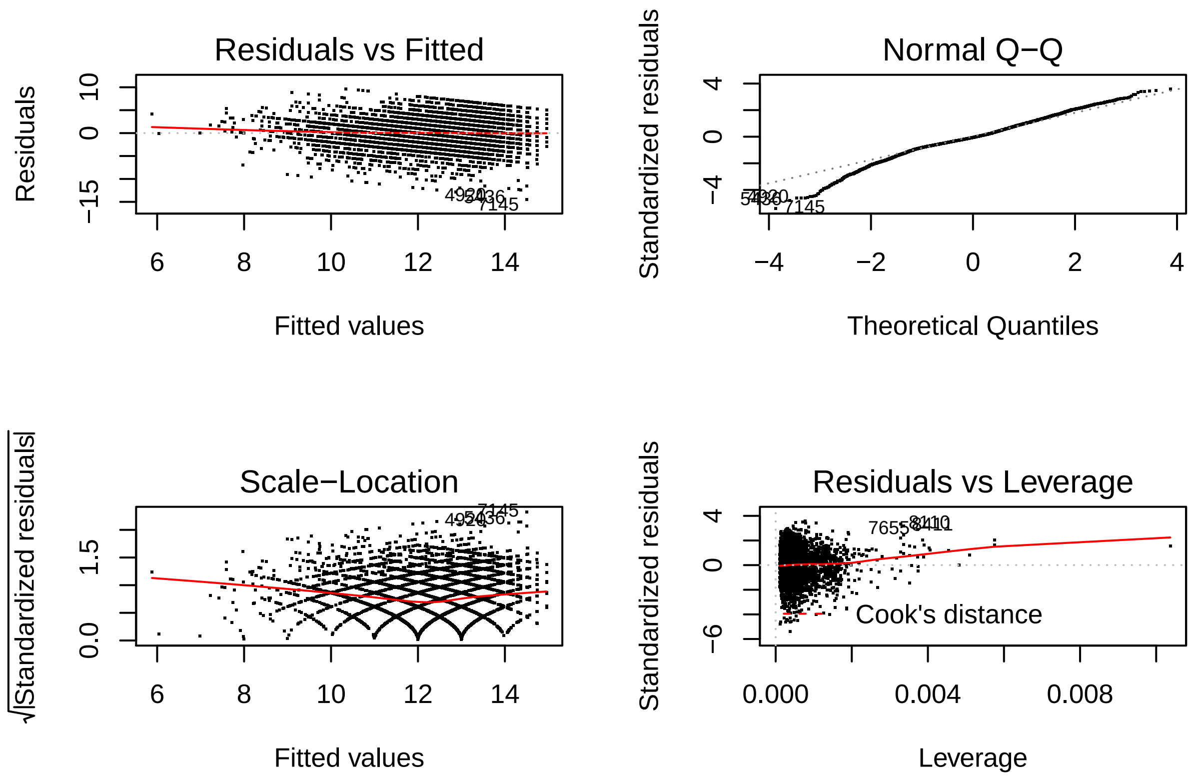
```r
# 3.4 Check whether the residuals are normally distributed
hist(residuals(fit))
```




```r
# 3.5 Plot predicted values by residuals
plot(predict(fit), residuals(fit), pch =".")
```



```r
par(mfrow = c(2, 2))
plot(fit, pch=".")
```




```r
par(mfrow = c(1,1))


# 4. Factors
# 4.1 create a table of values for ethnicity
table(gss$ethnicity)
```

```
##
## other  cauc
##  1785  7335
```

```r
# 4.2 Run a regression predicting education from ethnicity
lm(education ~ ethnicity, gss)
```

```
##
## Call:
## lm(formula = education ~ ethnicity, data = gss)
##
## Coefficients:
##   (Intercept)  ethnicitycauc
##       12.0773         0.6935
```

```r
# 4.3 Make a new factor variable where cauc is the reference value
#     and check that this worked by running a regression with
#     this new ethncity variable as the predictor.
gss$ethnicity_other <- factor( gss$ethnicity, c("cauc", "other"))
lm(education ~ ethnicity_other, gss)
```

```
##
## Call:
## lm(formula = education ~ ethnicity_other, data = gss)
##
## Coefficients:
##         (Intercept)  ethnicity_otherother
##             12.7708               -0.6935
```

```r
# 5. Comparing models
# 5.1 Fit a model predicting education from
#     (a) year and siblings
#     (b) year, siblings, and the interaction
# and compare the fit of these two models
fit1 <- lm(education ~ year + siblings, gss)
fit2 <- lm(education ~ year * siblings, gss)
summary(fit1)
```

```
##
## Call:
## lm(formula = education ~ year + siblings, data = gss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8806  -1.3896  -0.1353   1.6314   9.6240
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.094e+02  6.374e+00  -17.17   <2e-16 ***
## year         6.183e-02  3.201e-03   19.32   <2e-16 ***
## siblings    -2.508e-01  8.970e-03  -27.96   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.772 on 9117 degrees of freedom
## Multiple R-squared:  0.1204, Adjusted R-squared:  0.1203
## F-statistic: 624.3 on 2 and 9117 DF,  p-value: < 2.2e-16
```

```r
summary(fit2)
```

```
##
## Call:
## lm(formula = education ~ year * siblings, data = gss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8456  -1.4599  -0.1789   1.7660   9.6978
```

```
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -8.882e+01  1.029e+01  -8.635   <2e-16 ***
## year            5.149e-02  5.167e-03   9.965   <2e-16 ***
## siblings       -5.229e+00  1.952e+00  -2.679   0.0074 **
## year:siblings   2.502e-03  9.808e-04   2.551   0.0108 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.771 on 9116 degrees of freedom
## Multiple R-squared:  0.1211, Adjusted R-squared:  0.1208
## F-statistic: 418.6 on 3 and 9116 DF,  p-value: < 2.2e-16
```

```
anova(fit1, fit2)
```

```
## Analysis of Variance Table
##
## Model 1: education ~ year + siblings
## Model 2: education ~ year * siblings
##   Res.Df   RSS Df Sum of Sq      F  Pr(>F)
## 1   9117 70045
## 2   9116 69995  1     49.95 6.5053 0.01077 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Illustration of how ideas generalise to other kinds of models

Generalised linear models

```
# Don't create median splits
# but for the sake of example assume that we have
# a binary outcome
cas$high_performance <- as.numeric(cas$performance > median(cas$performance))

# glm: generalised linear models
# E.g., logistic regression
fit <- glm(high_performance ~ calworks + lunch, cas, family = binomial())
summary(fit)
```

```
##
## Call:
## glm(formula = high_performance ~ calworks + lunch, family = binomial(),
##     data = cas)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.78738  -0.40069   0.06019   0.50807   2.28800
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.41173    0.42663  10.341  < 2e-16 ***
## calworks    -0.04045    0.02686  -1.506    0.132
## lunch       -0.09038    0.01212  -7.458 8.76e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 582.24  on 419  degrees of freedom
## Residual deviance: 284.65  on 417  degrees of freedom
## AIC: 290.65
##
## Number of Fisher Scoring iterations: 6
```

```r
exp(coef(fit)) # exp beta coefficients
```

```
## (Intercept)    calworks       lunch
##  82.4120333   0.9603571   0.9135838
```

Multilevel modelling

```r
# Main multilevel modelling package
library(lme4)
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:base':
##
##     crossprod, tcrossprod
##
## Loading required package: Rcpp
```

```r
# also see older package
# library(nlme)

# Let's look at the built-in sleepstudy dataset
data(sleepstudy)
?sleepstudy
# long format dat
head(sleepstudy, 20)
```

```
##    Reaction Days Subject
## 1  249.5600    0     308
## 2  258.7047    1     308
## 3  250.8006    2     308
```

```
## 4  321.4398    3      308
## 5  356.8519    4      308
## 6  414.6901    5      308
## 7  382.2038    6      308
## 8  290.1486    7      308
## 9  430.5853    8      308
## 10 466.3535    9      308
## 11 222.7339    0      309
## 12 205.2658    1      309
## 13 202.9778    2      309
## 14 204.7070    3      309
## 15 207.7161    4      309
## 16 215.9618    5      309
## 17 213.6303    6      309
## 18 217.7272    7      309
## 19 224.2957    8      309
## 20 237.3142    9      309
```

```r
table(sleepstudy$Subject) # number of observations per participant
```

```
##
## 308 309 310 330 331 332 333 334 335 337 349 350 351 352 369 370 371 372
##  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10  10
```

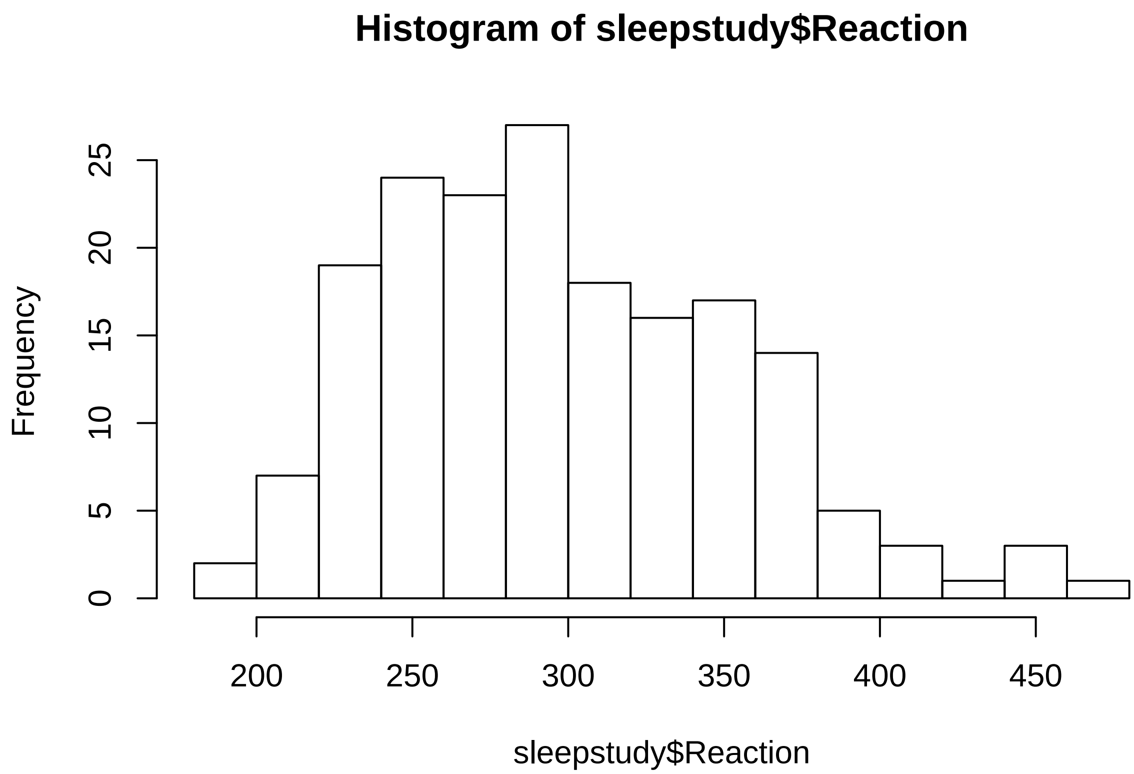
```r
length(table(sleepstudy$Subject)) # number of participants
```

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

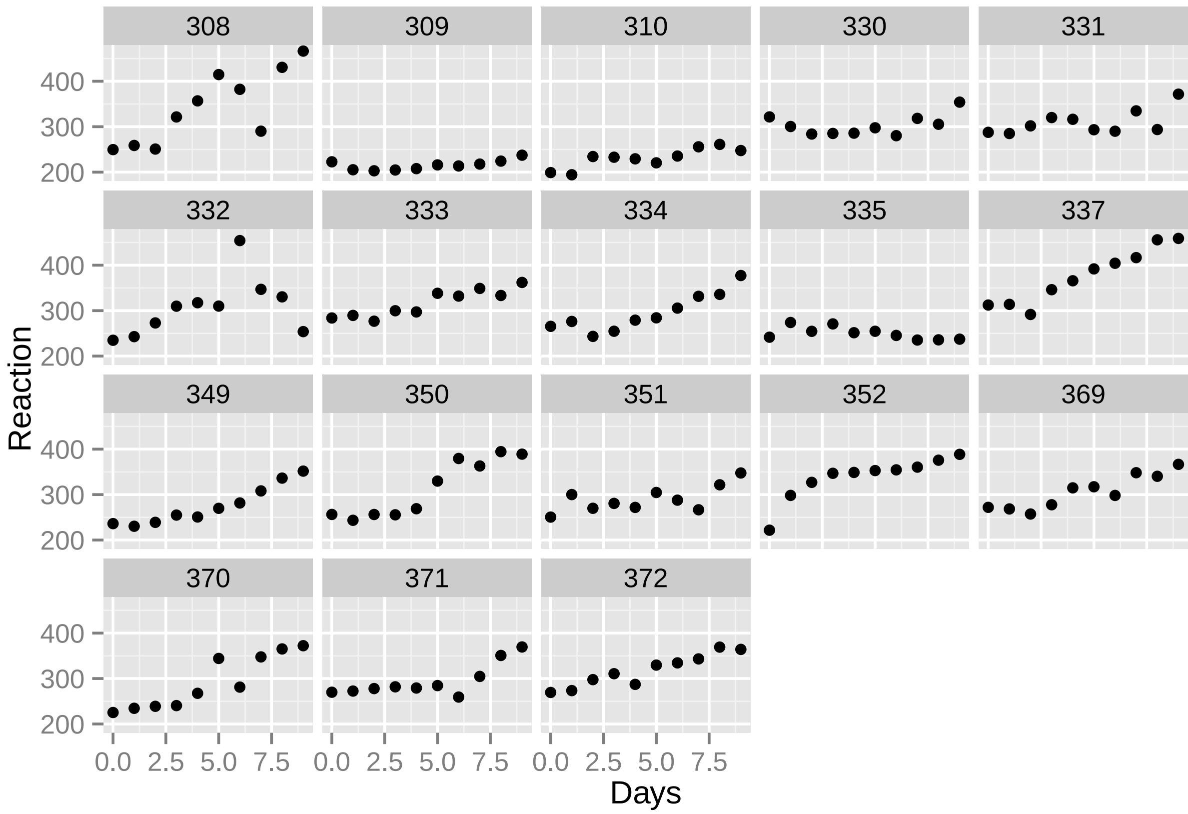
```r
table(sleepstudy$Days) # each participants observed at times 0 to 9
```

```
##
##  0  1  2  3  4  5  6  7  8  9
## 18 18 18 18 18 18 18 18 18 18
```

```r
# histogram of reaction time
hist(sleepstudy$Reaction, 10)
```




```r
# Reaction time over days of sleep deprivation
# each cell is one subject
ggplot(sleepstudy, aes(x = Days, y = Reaction)) +
    geom_point() +
    facet_wrap( ~ Subject)
```




```r
# Random intercept
fit1 <- lmer(Reaction ~ 1 + (1  | Subject),  data = sleepstudy)

# Random intercept + fixed Days effect
fit2 <- lmer(Reaction ~ 1 + Days + (1  | Subject),  data=sleepstudy)

# Random intercept and random Days effect
fit3 <- lmer(Reaction ~ 1 + Days + (1 + Days | Subject),  data=sleepstudy)

# # Random intercept and linear Days effect, fixed quadratic Days effect
fit4 <- lmer(Reaction ~ 1 + Days + I(Days^2) + (1 + Days | Subject),  data=sleepstudy)

# Compare models
anova(fit1, fit2)
```

```
## refitting model(s) with ML (instead of REML)

## Data: sleepstudy
## Models:
## fit1: Reaction ~ 1 + (1 | Subject)
## fit2: Reaction ~ 1 + Days + (1 | Subject)
##      Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## fit1  3 1916.5 1926.1 -955.27   1910.5
## fit2  4 1802.1 1814.8 -897.04   1794.1 116.46      1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit2, fit3)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: sleepstudy
## Models:
## fit2: Reaction ~ 1 + Days + (1 | Subject)
## fit3: Reaction ~ 1 + Days + (1 + Days | Subject)
##      Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## fit2  4 1802.1 1814.8 -897.04   1794.1
## fit3  6 1763.9 1783.1 -875.97   1751.9 42.139      2  7.072e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit3, fit4)
```

```
## refitting model(s) with ML (instead of REML)
```

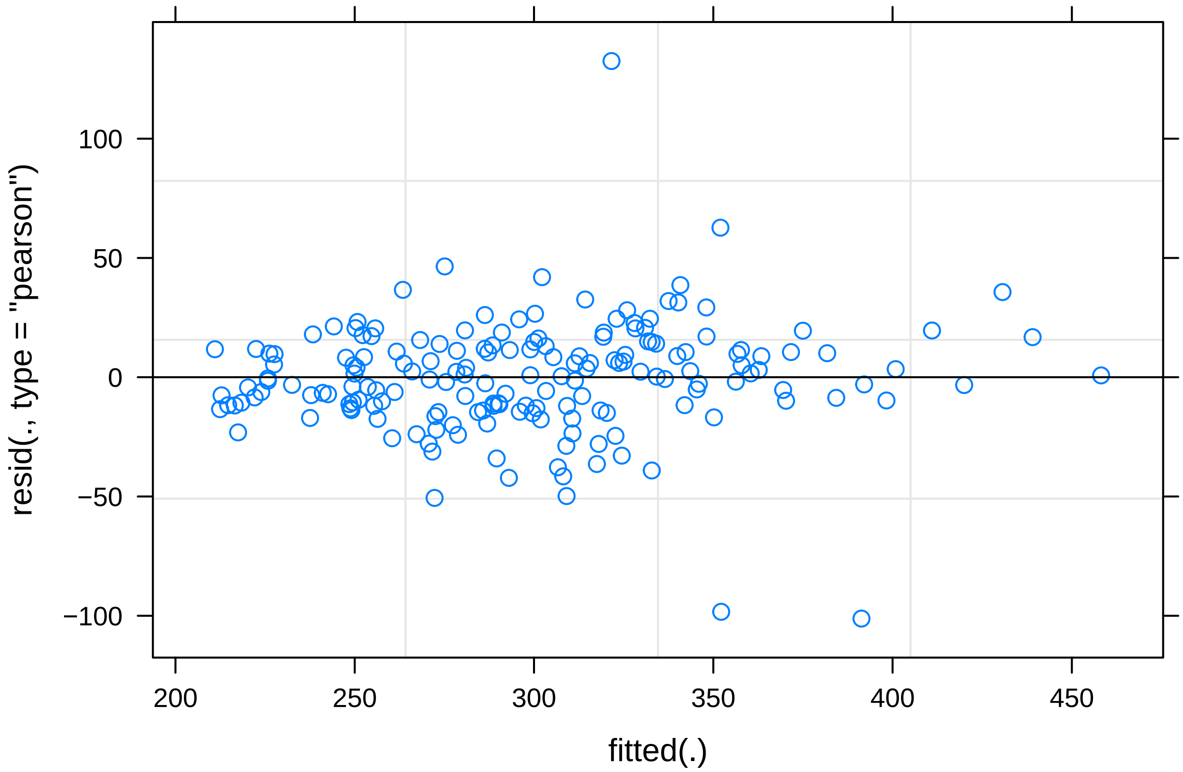
```
## Data: sleepstudy
## Models:
## fit3: Reaction ~ 1 + Days + (1 + Days | Subject)
## fit4: Reaction ~ 1 + Days + I(Days^2) + (1 + Days | Subject)
##      Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## fit3  6 1763.9 1783.1 -875.97   1751.9
## fit4  7 1764.3 1786.6 -875.14   1750.3 1.6577      1     0.1979
```

```r
# Summary of best fitting model
summary(fit3)
```

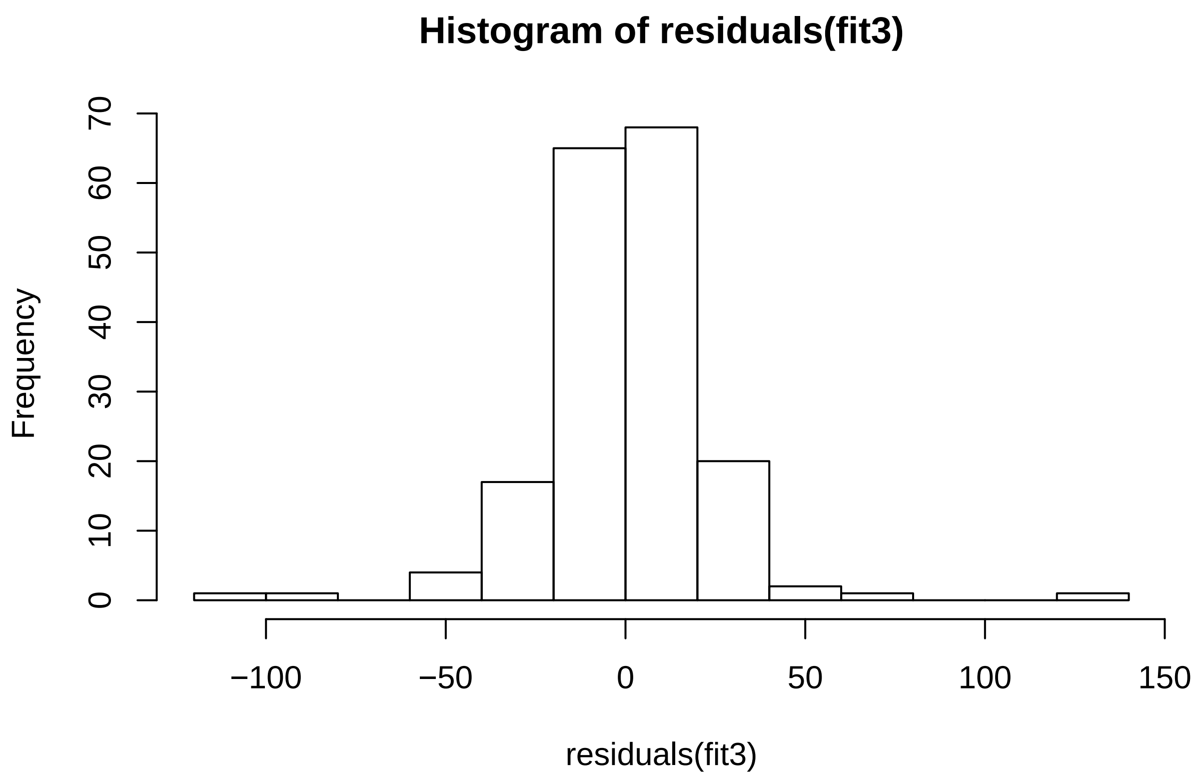
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Reaction ~ 1 + Days + (1 + Days | Subject)
##    Data: sleepstudy
##
## REML criterion at convergence: 1743.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9536 -0.4634  0.0231  0.4634  5.1793
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Subject  (Intercept) 612.09   24.740
##           Days         35.07    5.922   0.07
##  Residual             654.94   25.592
## Number of obs: 180, groups:  Subject, 18
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  251.405      6.825   36.84
## Days          10.467      1.546    6.77
##
```

```
## Correlation of Fixed Effects:
##      (Intr)
## Days -0.138
```

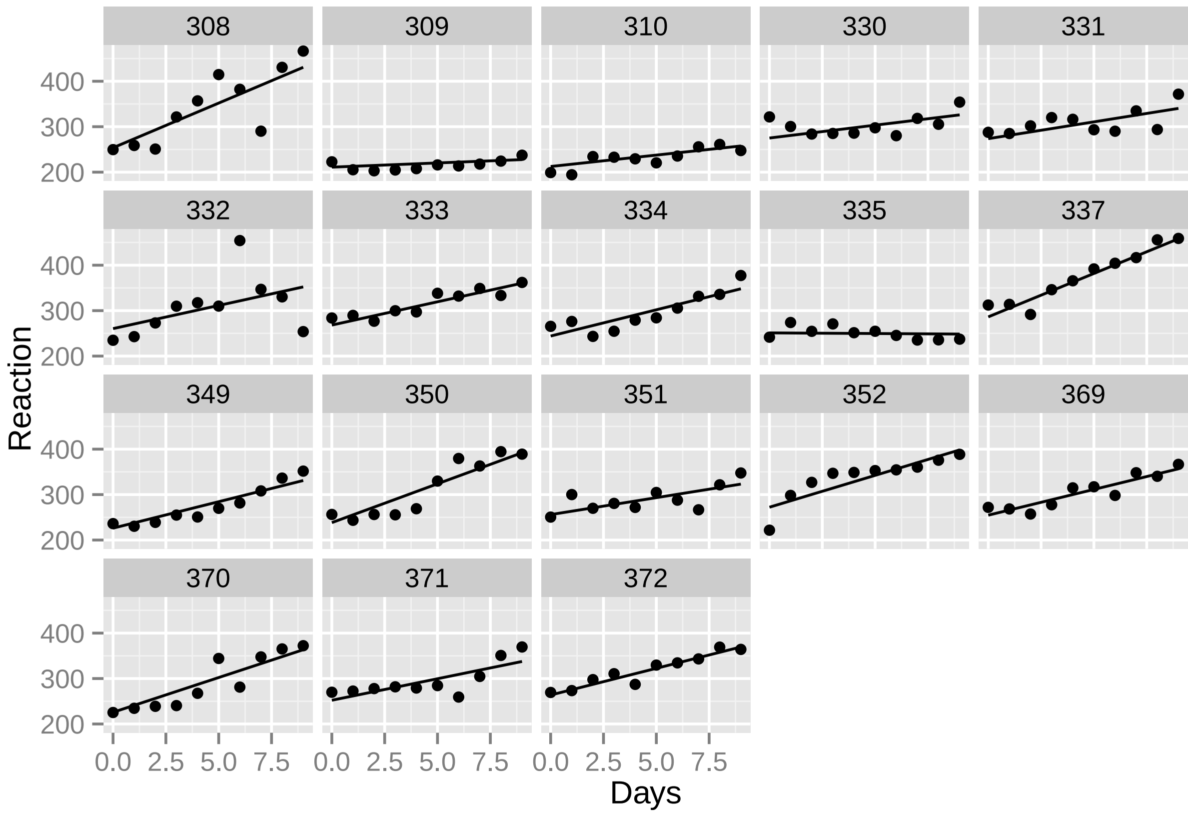
```r
# Most standard methods from lm also apply
plot(fit3) # plot fitted by residuals
```




```r
hist(residuals(fit3)) # histogram of residuals
```




```
# Save and plot predicted values
sleepstudy$predicted_fit3 <- predict(fit3)
ggplot(sleepstudy, aes(x = Days, y = Reaction)) +
    geom_point() + geom_line(aes(y=predicted_fit3))  +
    facet_wrap( ~ Subject)
```

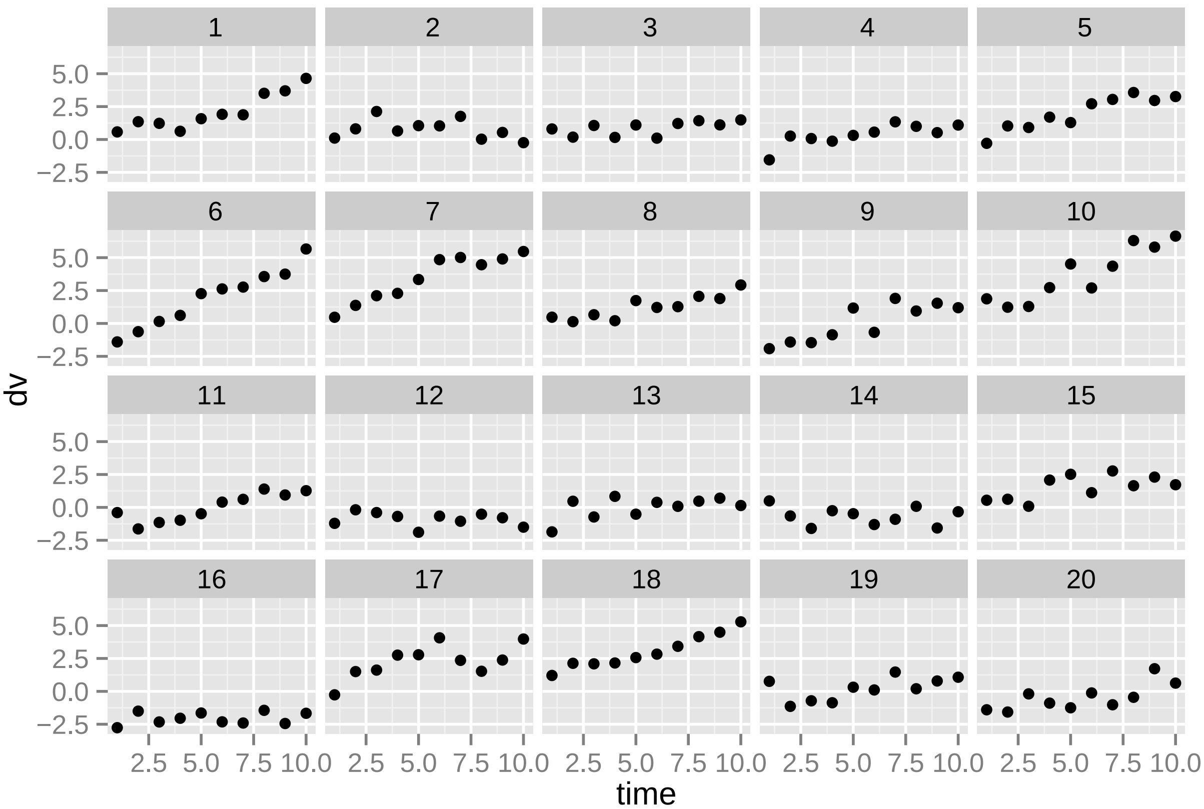


Exercise 2

```
# Let's create some simulated data with a random intercept
# and random slope.
set.seed <- 1234 # ensures we get the same results
sim <- expand.grid(subject = 1:20, time = 1:10)
sim_subject <- data.frame(subject = 1:20,
                          intercept = rnorm(20, 0, 1),
                          beta = rnorm(20, .3, .2))
sim <- merge(sim, sim_subject)
sim$dv <- rnorm(nrow(sim),  sim$intercept + sim$beta * sim$time,  .6)

# 1. Plot the the effect of the dv by time over subjects

# 2. Fit models predicting dv from time by subject
#    (1) a random intercept model
#    (b) a random intercept plus fixed slope model
#    (c) a rndom intercept and random slope model

# 3. Get summary information for model 3

# Compare the fits of the three models
# which is best?
```

Answers

```r
# Let's create some simulated data with a random intercept
# and random slope.
sset.seed <- 1234 # ensures we get the same results
sim <- expand.grid(subject = 1:20, time = 1:10)
sim_subject <- data.frame(subject = 1:20,
                          intercept = rnorm(20, 0, 1),
                          beta = rnorm(20, .3, .2))
sim <- merge(sim, sim_subject)
sim$dv <- rnorm(nrow(sim),  sim$intercept + sim$beta * sim$time,  .6)

# 1. Plot the the effect of the dv by time over subjects
ggplot(sim, aes(x = time, y = dv)) +
    geom_point() + facet_wrap( ~ subject)
```




```r
# 2. Fit models predicting dv from time by subject
#    (1) a random intercept model
#    (b) a random intercept plus fixed slope model
#    (c) a rndom intercept and random slope model

fit1 <- lmer(dv ~ 1 + (1  | subject),  data = sim)
fit2 <- lmer(dv ~ 1 + time + (1  | subject),  data=sim)
fit3 <- lmer(dv ~ 1 + time + (1 + time | subject),  data=sim)

# 3. Get summary information for model 3
summary(fit3)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: dv ~ 1 + time + (1 + time | subject)
##    Data: sim
##
## REML criterion at convergence: 510.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.19681 -0.64747 -0.04348  0.57064  2.60329
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  subject  (Intercept) 0.92295  0.9607
##           time        0.04312  0.2077   0.03
##  Residual             0.42664  0.6532
## Number of obs: 200, groups:  subject, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) -0.47896    0.23686  -2.022
## time         0.26438    0.04914   5.380
##
## Correlation of Fixed Effects:
##      (Intr)
## time -0.095
```

```r
# Compare the fits of the three models
# which is best
anova(fit1, fit2)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: sim
## Models:
## fit1: dv ~ 1 + (1 | subject)
## fit2: dv ~ 1 + time + (1 | subject)
##      Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## fit1  3 701.29 711.18 -347.64   695.29
## fit2  4 597.38 610.57 -294.69   589.38 105.91      1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(fit2, fit3) # model 3 is best
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: sim
## Models:
## fit2: dv ~ 1 + time + (1 | subject)
## fit3: dv ~ 1 + time + (1 + time | subject)
```

```
##      Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## fit2  4 597.38 610.57 -294.69   589.38
## fit3  6 517.05 536.84 -252.53   505.05 84.327      2  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Structural equation modelling

```r
# There are three main options for SEM
# library(sem): this is the original one
#
# library(OpenMx): Very powerful but more complicated
# http://openmx.psyc.virginia.edu/
#
# library(lavaan):
# This is my first choice when it comes to doing
# all the standard things that you might do in a program like Amos
# Lots of user friendly documentation on:
# http://lavaan.ugent.be/
# I also have a cheat sheet
# http://jeromyanglim.tumblr.com/post/33556941601/lavaan-cheat-sheet

library(lavaan)
```

```
## This is lavaan 0.5-18
## lavaan is BETA software! Please report any bugs.
```

```r
library(psych)
data(bfi)

cbfi <- na.omit(bfi)

dim(cbfi)
```

```
## [1] 2236   28
```

```r
head(cbfi)
```

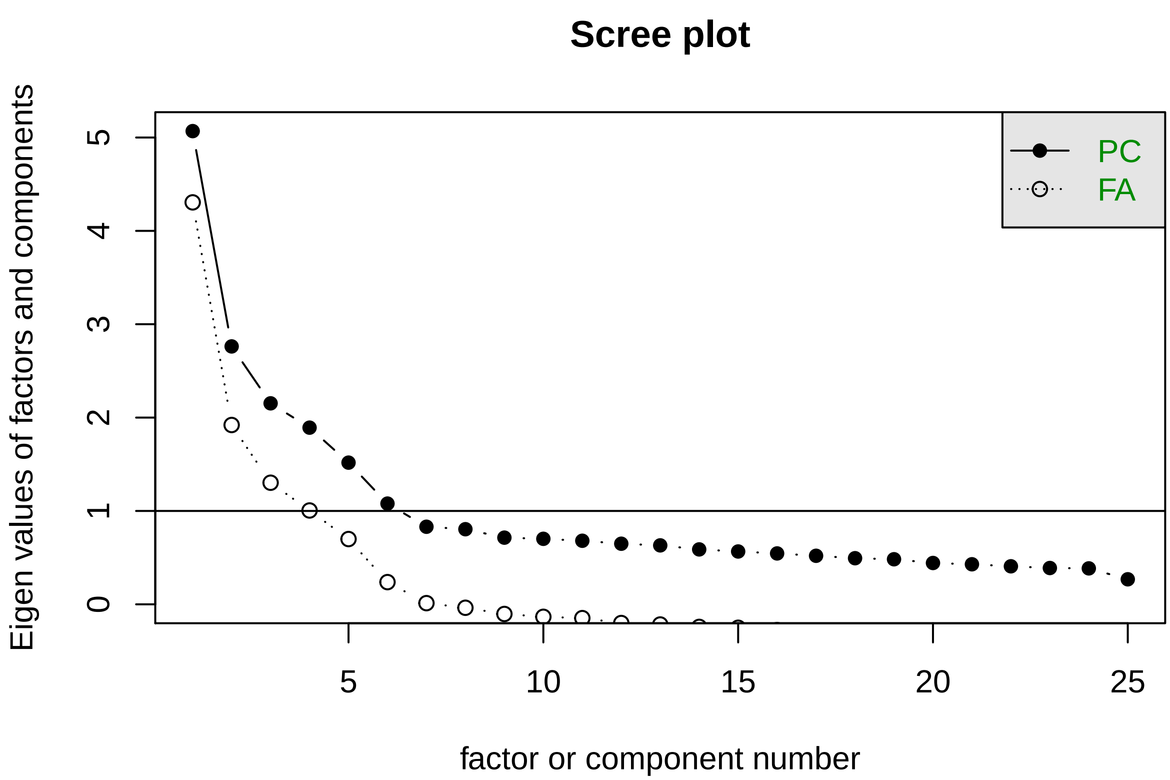
```
##       A1 A2 A3 A4 A5 C1 C2 C3 C4 C5 E1 E2 E3 E4 E5 N1 N2 N3 N4 N5 O1 O2 O3
## 61623  6  6  5  6  5  6  6  6  1  3  2  1  6  5  6  3  5  2  2  3  4  3  5
## 61629  4  3  1  5  1  3  2  4  2  4  3  6  4  2  1  6  3  2  6  4  3  2  4
## 61634  4  4  5  6  5  4  3  5  3  2  1  3  2  5  4  3  3  4  2  3  5  3  5
## 61640  4  5  2  2  1  5  5  5  2  2  3  4  3  6  5  2  4  2  2  3  5  2  5
## 61661  1  5  6  5  6  4  3  2  4  5  2  1  2  5  2  2  2  2  2  2  6  1  5
## 61664  2  6  5  6  5  3  5  6  3  6  2  2  4  6  6  4  4  4  6  6  6  1  5
##       O4 O5 gender education age
## 61623  6  1      2         3  21
## 61629  5  3      1         2  19
## 61634  6  3      1         1  21
## 61640  5  5      1         1  17
```

```
## 61661  5  2        1          5  68
## 61664  6  1        2          2  27
```

```r
dput(names(cbfi))
```

```
## c("A1", "A2", "A3", "A4", "A5", "C1", "C2", "C3", "C4", "C5",
## "E1", "E2", "E3", "E4", "E5", "N1", "N2", "N3", "N4", "N5", "O1",
## "O2", "O3", "O4", "O5", "gender", "education", "age")
```

```r
v$sem <- c("A1", "A2", "A3", "A4", "A5", "C1", "C2", "C3", "C4", "C5",
    "E1", "E2", "E3", "E4", "E5", "N1", "N2", "N3", "N4", "N5", "O1",
    "O2", "O3", "O4", "O5")

# Exploratory factor analysis
# Extract 5 factors with promax rotation
psych::scree(cbfi[ v$sem]) # scree plot
```




```r
fa <- factanal(cbfi[ v$sem], factors = 5, rotation = "promax")
print(fa, cutoff=.3) # print results hiding loadings below .3
```

```
##
## Call:
## factanal(x = cbfi[v$sem], factors = 5, rotation = "promax")
##
## Uniquenesses:
##    A1    A2    A3    A4    A5    C1    C2    C3    C4    C5    E1    E2
## 0.843 0.602 0.485 0.694 0.525 0.669 0.579 0.675 0.516 0.561 0.640 0.454
```

```
##    E3    E4    E5    N1    N2    N3    N4    N5    O1    O2    O3    O4
## 0.543 0.461 0.585 0.277 0.341 0.474 0.502 0.657 0.676 0.725 0.516 0.758
##    O5
## 0.714
##
## Loadings:
##    Factor1 Factor2 Factor3 Factor4 Factor5
## A1                         -0.387
## A2                          0.582
## A3                          0.646
## A4                          0.453
## A5                          0.558
## C1                  0.549
## C2                  0.658
## C3                  0.593
## C4                 -0.675
## C5                 -0.581
## E1         -0.632
## E2         -0.715
## E3          0.468                   0.302
## E4          0.605           0.338
## E5          0.473
## N1  0.909
## N2  0.860
## N3  0.682
## N4  0.398  -0.393
## N5  0.433
## O1                                  0.525
## O2                                 -0.473
## O3                                  0.629
## O4                                  0.369
## O5                                 -0.533
##
##                Factor1 Factor2 Factor3 Factor4 Factor5
## SS loadings      2.617   2.293   2.038   1.807   1.576
## Proportion Var   0.105   0.092   0.082   0.072   0.063
## Cumulative Var   0.105   0.196   0.278   0.350   0.413
##
## Factor Correlations:
##         Factor1 Factor2 Factor3 Factor4 Factor5
## Factor1   1.000  0.3698   0.376  0.1253   0.234
## Factor2   0.370  1.0000   0.247 -0.0245  -0.088
## Factor3   0.376  0.2468   1.000  0.2205   0.198
## Factor4   0.125 -0.0245   0.221  1.0000   0.183
## Factor5   0.234 -0.0880   0.198  0.1826   1.000
##
## Test of the hypothesis that 5 factors are sufficient.
## The chi square statistic is 1357.5 on 185 degrees of freedom.
## The p-value is 1.88e-177
```

```
# Confirmatory factor analysis
# Write out SEM using model notation
model1 <- "
    # latent variable definitions
```

```r
    # side point: first item gets loading of 1 so
    # it is clearer if this is a positively worded item
    agreeableness =~ A2 + A1 + A3 + A4 + 1 * A5
    conscientiousnes =~ C1 + C2 + C3 + C4 + C5
    extraversion =~ E3 + E1 + E2  + E4 + E5
    neuroticism =~ N1 + N2 + N3 + N4 + N5
    openness =~  O1 + O2 + O3 + O4 + O5
"

# fit model
fit1 <- cfa(model1, data=cbfi[ v$sem])
summary(fit1, fit.measures=TRUE)
```

```
## lavaan (0.5-18) converged normally after  67 iterations
##
##   Number of observations                          2236
##
##   Estimator                                         ML
##   Minimum Function Test Statistic             3855.328
##   Degrees of freedom                               266
##   P-value (Chi-square)                           0.000
##
## Model test baseline model:
##
##   Minimum Function Test Statistic            16560.077
##   Degrees of freedom                               300
##   P-value                                        0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    0.779
##   Tucker-Lewis Index (TLI)                       0.751
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -91295.294
##   Loglikelihood unrestricted model (H1)     -89367.630
##
##   Number of free parameters                         59
##   Akaike (AIC)                              182708.587
##   Bayesian (BIC)                            183045.621
##   Sample-size adjusted Bayesian (BIC)       182858.169
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.078
##   90 Percent Confidence Interval          0.076  0.080
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.077
##
```

```
## Parameter estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
##                      Estimate  Std.err  Z-value  P(>|z|)
## Latent variables:
##   agreeableness =~
##     A2                 1.000
##     A1                -0.595    0.042  -14.297    0.000
##     A3                 1.215    0.039   30.982    0.000
##     A4                 0.927    0.043   21.577    0.000
##     A5                 1.000
##   conscientiousnes =~
##     C1                 1.000
##     C2                 1.162    0.063   18.571    0.000
##     C3                 1.085    0.060   18.024    0.000
##     C4                -1.457    0.072  -20.319    0.000
##     C5                -1.555    0.080  -19.335    0.000
##   extraversion =~
##     E3                 1.000
##     E1                -1.052    0.048  -21.819    0.000
##     E2                -1.292    0.050  -25.670    0.000
##     E4                 1.186    0.046   25.849    0.000
##     E5                 0.866    0.040   21.844    0.000
##   neuroticism =~
##     N1                 1.000
##     N2                 0.951    0.025   37.526    0.000
##     N3                 0.898    0.026   34.192    0.000
##     N4                 0.694    0.026   26.365    0.000
##     N5                 0.643    0.028   23.217    0.000
##   openness =~
##     O1                 1.000
##     O2                -1.058    0.072  -14.657    0.000
##     O3                 1.368    0.075   18.182    0.000
##     O4                 0.413    0.049    8.388    0.000
##     O5                -1.006    0.064  -15.719    0.000
##
## Covariances:
##   agreeableness ~~
##     conscientisns      0.168    0.016   10.268    0.000
##     extraversion       0.467    0.025   18.352    0.000
##     neuroticism       -0.202    0.027   -7.418    0.000
##     openness           0.132    0.016    8.334    0.000
##   conscientiousnes ~~
##     extraversion       0.203    0.019   10.871    0.000
##     neuroticism       -0.234    0.025   -9.501    0.000
##     openness           0.117    0.014    8.374    0.000
##   extraversion ~~
##     neuroticism       -0.259    0.030   -8.559    0.000
##     openness           0.244    0.020   12.126    0.000
##   neuroticism ~~
##     openness          -0.092    0.023   -4.039    0.000
##
```

```
## Variances:
##     A2                0.772    0.029
##     A1                1.717    0.053
##     A3                0.744    0.033
##     A4                1.561    0.051
##     A5                0.891    0.032
##     C1                1.054    0.036
##     C2                1.144    0.041
##     C3                1.156    0.040
##     C4                0.955    0.041
##     C5                1.627    0.061
##     E3                1.055    0.038
##     E1                1.792    0.060
##     E2                1.332    0.051
##     E4                1.078    0.042
##     E5                1.209    0.041
##     N1                0.798    0.038
##     N2                0.862    0.038
##     N3                1.219    0.045
##     N4                1.639    0.054
##     N5                1.949    0.062
##     O1                0.858    0.033
##     O2                1.945    0.065
##     O3                0.682    0.040
##     O4                1.313    0.040
##     O5                1.366    0.047
##     agreeableness     0.621    0.031
##     conscientisns     0.425    0.036
##     extraversion      0.746    0.048
##     neuroticism       1.648    0.075
##     openness          0.396    0.034
```

```r
# Suggest modifications
mod_ind <- modificationindices(fit1)
split(head(mod_ind[order(mod_ind$mi, decreasing=TRUE), ], 20),
      head(mod_ind[order(mod_ind$mi, decreasing=TRUE), "op"], 20))
```

```
## $`=~`
##                  lhs op rhs      mi    epc sepc.lv sepc.all sepc.nox
## 1      extraversion =~  N4 193.348 -0.526  -0.455   -0.291   -0.291
## 2          openness =~  E3 133.324  0.644   0.406    0.302    0.302
## 3          openness =~  E4 126.514 -0.669  -0.421   -0.289   -0.289
## 4  conscientiousnes =~  E5 109.333  0.516   0.336    0.253    0.253
## 5      extraversion =~  O3 107.380  0.446   0.385    0.323    0.323
## 6      extraversion =~  O4 101.990 -0.383  -0.331   -0.282   -0.282
## 7       neuroticism =~  O4  95.511  0.204   0.263    0.223    0.223
## 8       neuroticism =~  C2  94.252  0.218   0.279    0.213    0.213
## 9       neuroticism =~  C5  90.725  0.262   0.337    0.207    0.207
## 10 conscientiousnes =~  N4  89.721 -0.503  -0.328   -0.210   -0.210
##
## $`~~`
##    lhs op rhs      mi   epc sepc.lv sepc.all sepc.nox
## 1   N1 ~~  N2 371.089 0.819   0.819    0.341    0.341
## 2   N3 ~~  N4 115.971 0.391   0.391    0.157    0.157
```

```
## 3   C1 ~~  C2  98.826  0.286   0.286    0.179    0.179
## 4   E2 ~~  O4  91.887  0.298   0.298    0.158    0.158
## 5   N4 ~~  O4  87.266  0.303   0.303    0.165    0.165
## 6   A2 ~~  A1  85.774 -0.261  -0.261   -0.159   -0.159
## 7   N1 ~~  N4  81.332 -0.318  -0.318   -0.130   -0.130
## 8   A5 ~~  E4  79.097  0.223   0.223    0.125    0.125
## 9   O2 ~~  O5  77.587  0.357   0.357    0.174    0.174
## 10  N2 ~~  N4  75.885 -0.300  -0.300   -0.125   -0.125
```

```
# Refine model
model2 <- "
    # latent variable definitions
    # side point: first item gets loading of 1 so
    # it is clearer if this is a positively worded item
    agreeableness =~ A2 + A1 + A3 + A4 + 1 * A5
    conscientiousnes =~ C1 + C2 + C3 + C4 + C5
    extraversion =~ E3 + E1 + E2  + E4 + E5
    neuroticism =~ N1 + N2 + N3 + N4 + N5
    openness =~  O1 + O2 + O3 + O4 + O5

    # add some correlated items that are very similar
    N1 ~~ N2
    N3 ~~ N4
    C1 ~~ C2
"

fit2 <- cfa(model2, data=cbfi[ v$sem])
```

```
## Warning in lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats, : lavaan WARNING: co
##   lavaan NOTE: this may be a symptom that the model is not identified.

## Warning in lavaan::lavaan(model = model2, data = cbfi[v$sem], model.type =
## "cfa", : lavaan WARNING: some estimated variances are negative

## Warning in lavaan::lavaan(model = model2, data = cbfi[v$sem], model.type
## = "cfa", : lavaan WARNING: covariance matrix of latent variables is not
## positive definite; use inspect(fit,"cov.lv") to investigate.
```

```
summary(fit2, fit.measures=TRUE)
```

```
## lavaan (0.5-18) converged normally after 2680 iterations
##
##   Number of observations                          2236
##
##   Estimator                                         ML
##   Minimum Function Test Statistic             4434.123
##   Degrees of freedom                               263
##   P-value (Chi-square)                           0.000
##
## Model test baseline model:
##
##   Minimum Function Test Statistic            16560.077
```

```
##   Degrees of freedom                               300
##   P-value                                        0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    0.743
##   Tucker-Lewis Index (TLI)                       0.707
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -91584.691
##   Loglikelihood unrestricted model (H1)     -89367.630
##
##   Number of free parameters                         62
##   Akaike (AIC)                              183293.382
##   Bayesian (BIC)                            183647.554
##   Sample-size adjusted Bayesian (BIC)       183450.570
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.084
##   90 Percent Confidence Interval          0.082  0.086
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.105
##
## Parameter estimates:
##
##   Information                                 Expected
##   Standard Errors                             Standard
##
##                    Estimate  Std.err  Z-value  P(>|z|)
## Latent variables:
##   agreeableness =~
##     A2                1.000
##     A1               -0.598
##     A3                1.223
##     A4                0.914
##     A5                1.000
##   conscientiousnes =~
##     C1                1.000
##     C2                1.190
##     C3                0.563
##     C4             2260.858
##     C5               -1.796
##   extraversion =~
##     E3                1.000
##     E1               -1.069
##     E2               -1.311
##     E4                1.200
##     E5                0.855
##   neuroticism =~
```

```
##     N1                 1.000
##     N2                 0.935
##     N3                 1.237
##     N4                 1.019
##     N5                 0.872
##   openness =~
##     O1                 1.000
##     O2                -1.053
##     O3                 1.403
##     O4                 0.425
##     O5                -1.014
##
## Covariances:
##   N1 ~~
##     N2                 0.693
##   N3 ~~
##     N4                -0.076
##   C1 ~~
##     C2                 0.682
##   agreeableness ~~
##     conscientisns     -0.000
##     extraversion       0.462
##     neuroticism       -0.152
##     openness           0.131
##   conscientiousnes ~~
##     extraversion      -0.000
##     neuroticism        0.000
##     openness          -0.000
##   extraversion ~~
##     neuroticism       -0.268
##     openness           0.238
##   neuroticism ~~
##     openness          -0.076
##
## Variances:
##     A2                 0.771
##     A1                 1.714
##     A3                 0.733
##     A4                 1.577
##     A5                 0.893
##     C1                 1.481
##     C2                 1.720
##     C3                 1.656
##     C4               832.184
##     C5                 2.655
##     E3                 1.065
##     E1                 1.775
##     E2                 1.311
##     E4                 1.069
##     E5                 1.230
##     N1                 1.361
##     N2                 1.405
##     N3                 0.886
##     N4                 1.306
```

```
##     N5                1.804
##     O1                0.867
##     O2                1.960
##     O3                0.662
##     O4                1.311
##     O5                1.369
##     agreeableness     0.621
##     conscientisns    -0.000
##     extraversion      0.736
##     neuroticism       1.086
##     openness          0.387
```

```r
ff1 <- fitMeasures(fit1)
ff2 <- fitMeasures(fit2)
ff1
```

```
##                npar                fmin               chisq
##              59.000               0.862            3855.328
##                  df              pvalue      baseline.chisq
##             266.000               0.000           16560.077
##         baseline.df     baseline.pvalue                 cfi
##             300.000               0.000               0.779
##                 tli                nnfi                 rfi
##               0.751               0.751               0.737
##                 nfi                pnfi                 ifi
##               0.767               0.680               0.780
##                 rni                logl   unrestricted.logl
##               0.779          -91295.294          -89367.630
##                 aic                 bic              ntotal
##          182708.587          183045.621            2236.000
##                bic2               rmsea      rmsea.ci.lower
##          182858.169               0.078               0.076
##      rmsea.ci.upper        rmsea.pvalue                 rmr
##               0.080               0.000               0.157
##          rmr_nomean                srmr        srmr_bentler
##               0.157               0.077               0.077
## srmr_bentler_nomean         srmr_bollen  srmr_bollen_nomean
##               0.077               0.076               0.076
##          srmr_mplus   srmr_mplus_nomean               cn_05
##               0.077               0.077             177.917
##               cn_01                 gfi                agfi
##             188.088               0.861               0.830
##                pgfi                 mfi                ecvi
##               0.705               0.448               1.777
```

```r
# show measures you want
dput(names(ff1))
```

```
## c("npar", "fmin", "chisq", "df", "pvalue", "baseline.chisq",
## "baseline.df", "baseline.pvalue", "cfi", "tli", "nnfi", "rfi",
## "nfi", "pnfi", "ifi", "rni", "logl", "unrestricted.logl", "aic",
## "bic", "ntotal", "bic2", "rmsea", "rmsea.ci.lower", "rmsea.ci.upper",
## "rmsea.pvalue", "rmr", "rmr_nomean", "srmr", "srmr_bentler",
```

```
## "srmr_bentler_nomean", "srmr_bollen", "srmr_bollen_nomean", "srmr_mplus",
## "srmr_mplus_nomean", "cn_05", "cn_01", "gfi", "agfi", "pgfi",
## "mfi", "ecvi")
```

```r
v$stats <-  c("npar", "chisq", "df", "pvalue",
   "cfi", "rmsea", "rmsea.ci.lower", "rmsea.ci.upper")

# compare stats
round(data.frame(ff1[v$stats],  ff2[v$stats]), 3)
```

```
##                ff1.v.stats. ff2.v.stats.
## npar                 59.000       62.000
## chisq              3855.328     4434.123
## df                  266.000      263.000
## pvalue                0.000        0.000
## cfi                   0.779        0.743
## rmsea                 0.078        0.084
## rmsea.ci.lower        0.076        0.082
## rmsea.ci.upper        0.080        0.086
```

Meta analysis

```r
# Lots of meta-analysis options
# http://cran.r-project.org/web/views/MetaAnalysis.html
# meta, rmeta, and metafor are all fairly general meta-analysis packages
library(metafor)
```

```
## Loading 'metafor' package (version 1.9-6). For an overview
## and introduction to the package please type: help(metafor).
```

```r
# Example is based on
# http://www.metafor-project.org/doku.php/analyses:normand1999
data("dat.normand1999")
?dat.normand1999
# compares mean length of stay for stroke patients
# in speialised care (group 1) and routine care (group 2)
dat.normand1999
```

```
##   study             source n1i m1i sd1i n2i m2i sd2i
## 1     1          Edinburgh 155  55   47 156  75   64
## 2     2     Orpington-Mild  31  27    7  32  29    4
## 3     3 Orpington-Moderate  75  64   17  71 119   29
## 4     4   Orpington-Severe  18  66   20  18 137   48
## 5     5      Montreal-Home   8  14    8  13  18   11
## 6     6  Montreal-Transfer  57  19    7  52  18    4
## 7     7          Newcastle  34  52   45  33  41   34
## 8     8               Umea 110  21   16 183  31   27
## 9     9            Uppsala  60  30   27  52  23   20
```

```r
mean(dat.normand1999$m1i) # mean over studies length of time in specialised care
```

```
## [1] 38.66667
```

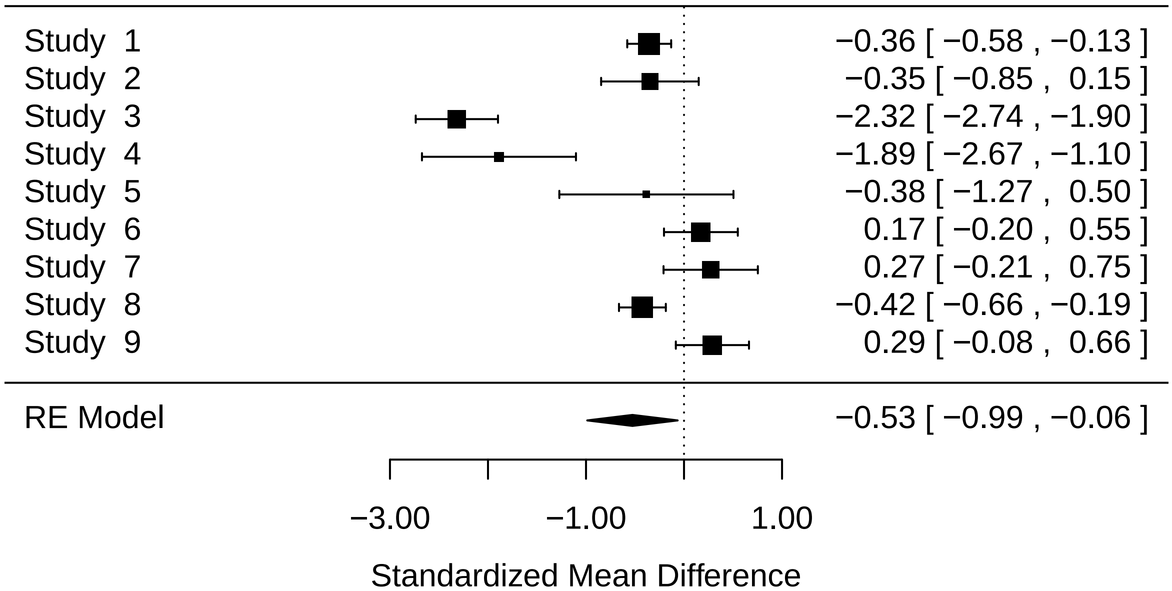
```r
mean(dat.normand1999$m2i) # ...........                         in routine care
```

```
## [1] 54.55556
```

```r
# calculate pooled standard deviation
dat.normand1999$sdpi <- with(dat.normand1999,
                             sqrt(((n1i - 1) * sd1i^2 + (n2i - 1) * sd2i^2) /
                                  (n1i + n2i - 2)))
# Compare standard mean differences
dat <- escalc(m1i=m1i, sd1i=sdpi, n1i=n1i, m2i=m2i, sd2i=sdpi, n2i=n2i,
              measure="SMD", data=dat.normand1999, digits=2)
# Fit random effects meta analysis
fit <- rma(yi, vi, data=dat, method="HS", digits=2)
summary(fit) # Estimate of mean and sd of effect
```

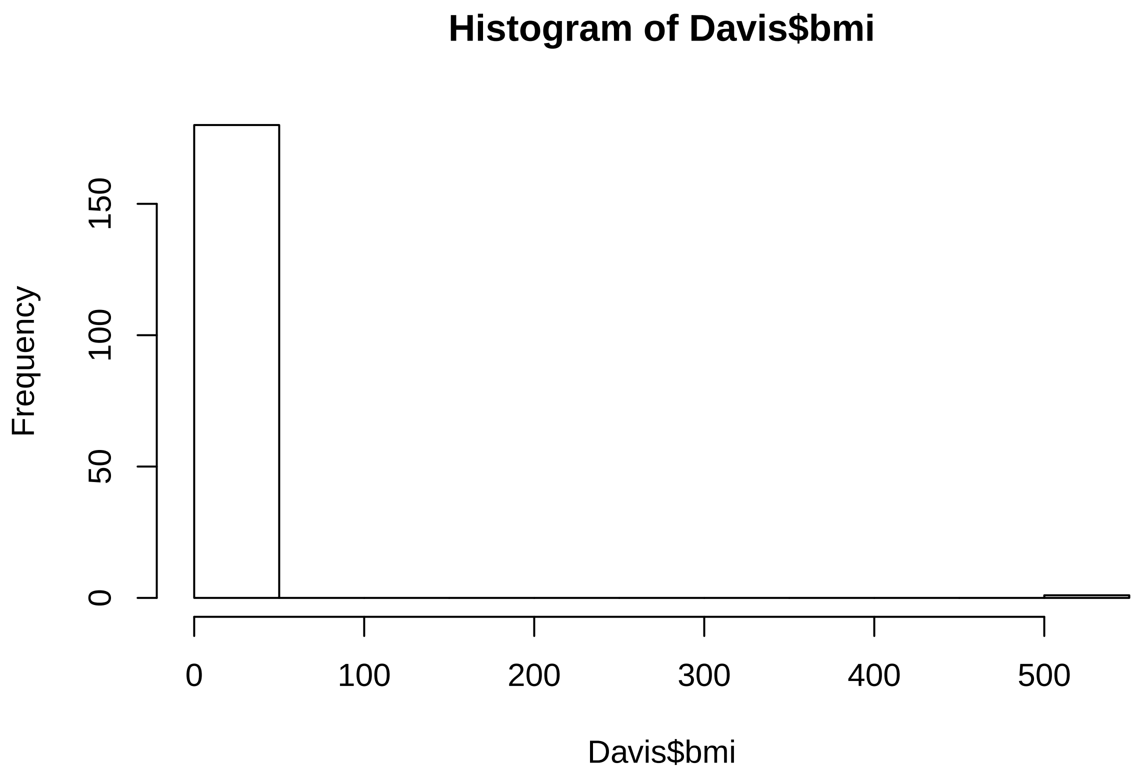
```
##
## Random-Effects Model (k = 9; tau^2 estimator: HS)
##
##   logLik  deviance       AIC       BIC      AICc
##   -12.02     34.71     28.04     28.44     30.04
##
## tau^2 (estimated amount of total heterogeneity): 0.44 (SE = 0.24)
## tau (square root of estimated tau^2 value):      0.66
## I^2 (total heterogeneity / total variability):   92.11%
## H^2 (total variability / sampling variability):  12.67
##
## Test for Heterogeneity:
## Q(df = 8) = 123.73, p-val < .01
##
## Model Results:
##
## estimate       se     zval     pval    ci.lb    ci.ub
##    -0.53     0.24    -2.23     0.03    -0.99    -0.06        *
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
forest(fit) # Plot of effect size estimates
```



Bootstrapping

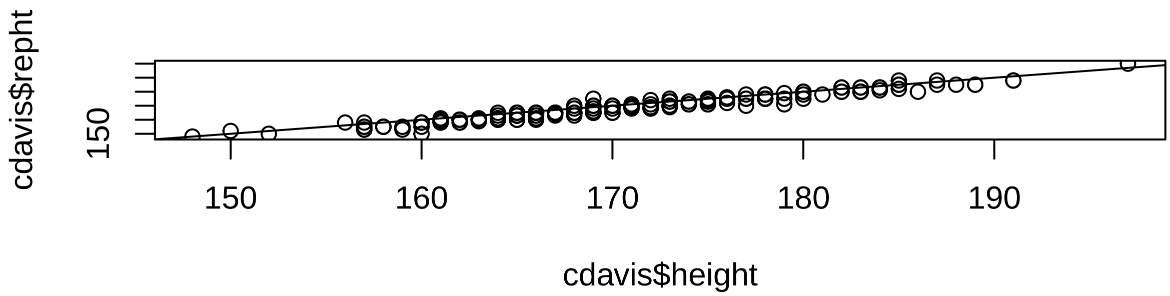
```r
library(boot)
# see also
# http://www.statmethods.net/advstats/bootstrapping.html


library(car)
# Use height and weight data of university students
data(Davis)
Davis <- na.omit(Davis)
Davis$bmi <- with(Davis, weight/(height/100)^2)
hist(Davis$bmi)
```

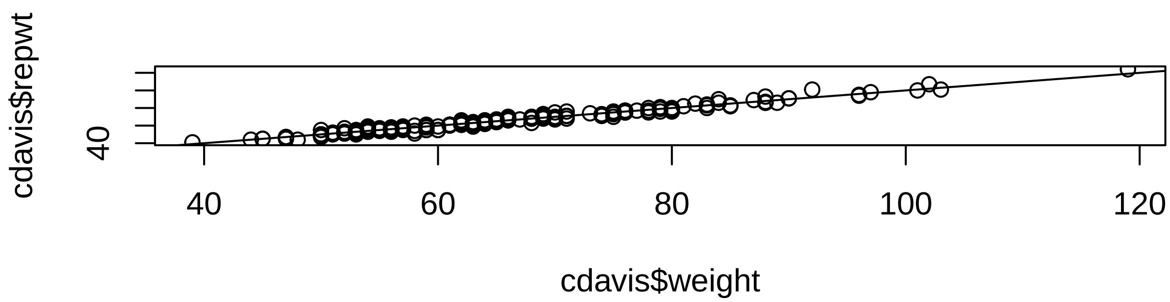



```r
# looks like data entry error
Davis[ Davis$bmi > 100, ]
```

```
##    sex weight height repwt repht      bmi
## 12   F    166     57    56   163 510.9264
```

```r
# let's remove and work with cleaned data
cdavis <-  Davis[ Davis$bmi < 100, ]

# Which correlation is larger
# Correlation between actual and report height
# or correlation between actual and reported weight
par(mfrow=c(2,1))
plot(cdavis$height, cdavis$repht)
abline(a = 0, b = 1)
plot(cdavis$weight, cdavis$repwt)
abline(a = 0, b = 1)
```






```
# look at sample data
# correlation for weight looks a tiny bit bigger
# but is it significant
cor(cdavis$height, cdavis$repht)
```

```
## [1] 0.9755571
```

```
cor(cdavis$weight, cdavis$repwt)
```

```
## [1] 0.9860954
```

```
# How could we test this using a bootstrap?

# function receives
cordif <- function(data, i) {
    cidavis <- data[i, ]
    cor1 <- cor(cidavis$height, cidavis$repht)
    cor2 <- cor(cidavis$weight, cidavis$repwt)
    cor1 - cor2
}

fit <- boot(data = cdavis, statistic = cordif, R = 2000)
fit
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
```

```
## boot(data = cdavis, statistic = cordif, R = 2000)
##
##
## Bootstrap Statistics :
##        original        bias    std. error
## t1* -0.01053833 -4.795489e-05 0.004212112
```

```r
boot.ci(fit)
```

```
## Warning in boot.ci(fit): bootstrap variances needed for studentized
## intervals
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = fit)
##
## Intervals :
## Level      Normal              Basic
## 95%   (-0.0187, -0.0022 )   (-0.0182, -0.0016 )
##
## Level     Percentile            BCa
## 95%   (-0.0195, -0.0029 )   (-0.0207, -0.0034 )
## Calculations and Intervals on Original Scale
```

Bayesian modelling

```r
# See interfaces with Bayesian modelling language like
# library(rjags)  # JAGS
# and
# library(rstan) # Stan
#
# See example project:
# Anglim, J., & Wynton, S. K. (2015). Hierarchical Bayesian Models of
# Subtask Learning. Journal of experimental psychology. Learning, memory, and cognition.
# Full repository with R code available at
# https://github.com/jeromyanglim/anglim-wynton-2014-subtasks
```