Week 1 Problem Based Learning and Practical Solutions

Jeffrey O. Hanson[1]
[1]*School of Biological Sciences, The University of Queensland, Brisbane, QLD, Australia*
Correspondance should be addressed to jeffrey.hanson@uqconnect.edu.au

21 March 2016

Contents

Problem based learning workshop

`R` practical session

General notes on using the teaching manual

- network information
- always use Windows 7
  - Map to L drive: `\\\\sci-tl01.soe.uq.edu.au\\Teaching\\BIOL\\BIOL2006`
  - Program name: `BIOL2006PracManual_2016.exe`
- sign on
  - must enter valid id
  - my sign on is `"Jeffrey"`
- GUI
  - popup links are bright green
  - arrows to navigate
  - home icon is home
  - R icon saves code to desktop and open up with Rstudio
  - R code is in blue
  - comments in black
  - questions also in green

  - tooltips appear in brown bar along the bottom of program
  - minimize question popups by clicking on question again
  - assessment questions turn black after they're done, but note that you might have to mouse over questions for them to turn black due to bug
  - students must click logout to store results, they cannot just close the program
  - students must answer all practice questions to get to final assessment questions
  - each question in the assessment page refers to a specific page with info on how to answer it

- R scripts
  - right click on the scripts
  - open with -> Rstudio

- Potential issues
  - the computers don't associate .R scripts with Rstudio: need to open Rstudio first then open the script
  - attaching datasets: **never to do this even though it looks handy**

P1. Using R as a calculator

```
1 + 2
```

```
## [1] 3
```

```
log(10)/( 41 + exp(33.8))
```

```
## [1] 4.820168e-15
```

```
x <- (48 + 34)/5.5       # store the result as 'x'
x          # look at the contents of 'x'
```

```
## [1] 14.90909
```

```
y <- x + 3
x*y
```

```
## [1] 267.0083
```

P2. Joining stuff together

```r
stuff <- c(1,"cat", 4.5, 6, "elephant")     # create some "stuff"
class(stuff)                        # what type is it?
```

```
## [1] "character"
```

```r
stuff               # have a look ; note that "4.5" is not a number!
```

```
## [1] "1"        "cat"      "4.5"      "6"        "elephant"
```

```r
numbers <- c(1,2.8,3,4.5,12.22,13)  # create "numbers"
class(numbers)                  # what type is it?
```

```
## [1] "numeric"
```

```r
paste(stuff,numbers)      # join things together as character strings
```

```
## [1] "1 1"            "cat 2.8"        "4.5 3"          "6 4.5"
## [5] "elephant 12.22" "1 13"
```

```r
3:7      # create a consecutive sequence of numbers
```

```
## [1] 3 4 5 6 7
```

```r
paste(1:4,stuff,numbers,"whatever!")
```

```
## [1] "1 1 1 whatever!"            "2 cat 2.8 whatever!"
## [3] "3 4.5 3 whatever!"          "4 6 4.5 whatever!"
## [5] "1 elephant 12.22 whatever!" "2 1 13 whatever!"
```

P3. Viewing data

```r
women # display the data set called 'women'
```

```
##    height weight
## 1      58    115
## 2      59    117
## 3      60    120
## 4      61    123
## 5      62    126
## 6      63    129
```

```
## 7       64    132
## 8       65    135
## 9       66    139
## 10      67    142
## 11      68    146
## 12      69    150
## 13      70    154
## 14      71    159
## 15      72    164
```

```r
dim(women) # display the numbers of rows and columns
```

```
## [1] 15  2
```

```r
names(women) # display the column headings
```

```
## [1] "height" "weight"
```

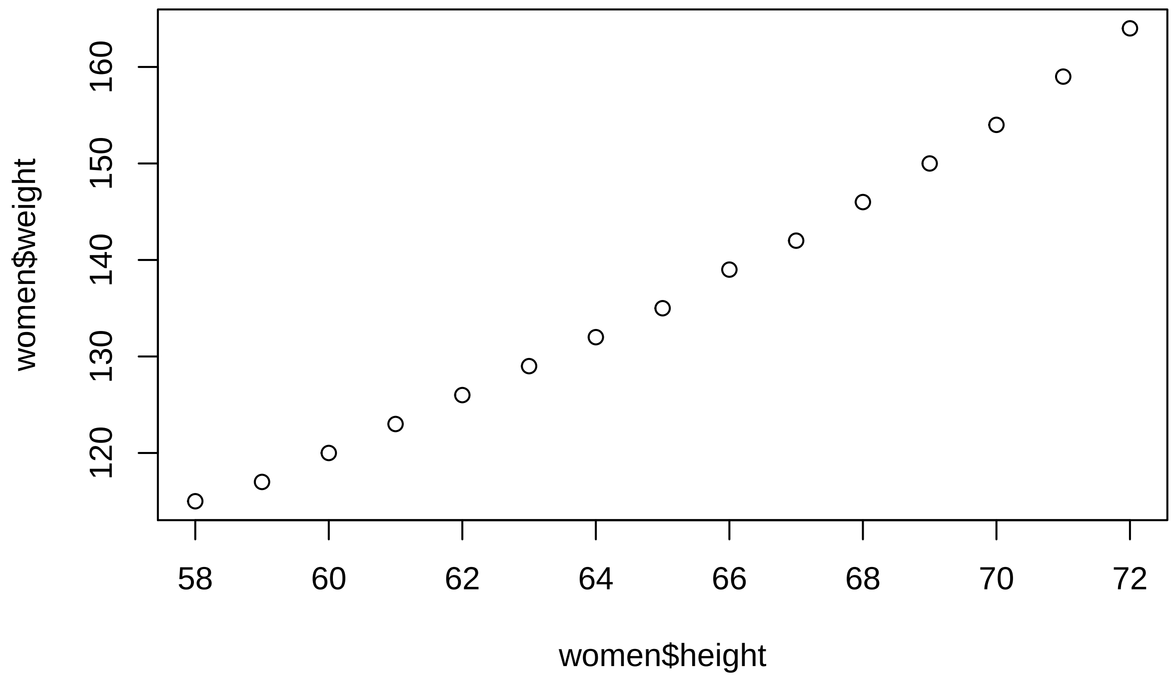
```r
women$weight # display the column called 'weight' within 'women'
```

```
##  [1] 115 117 120 123 126 129 132 135 139 142 146 150 154 159 164
```

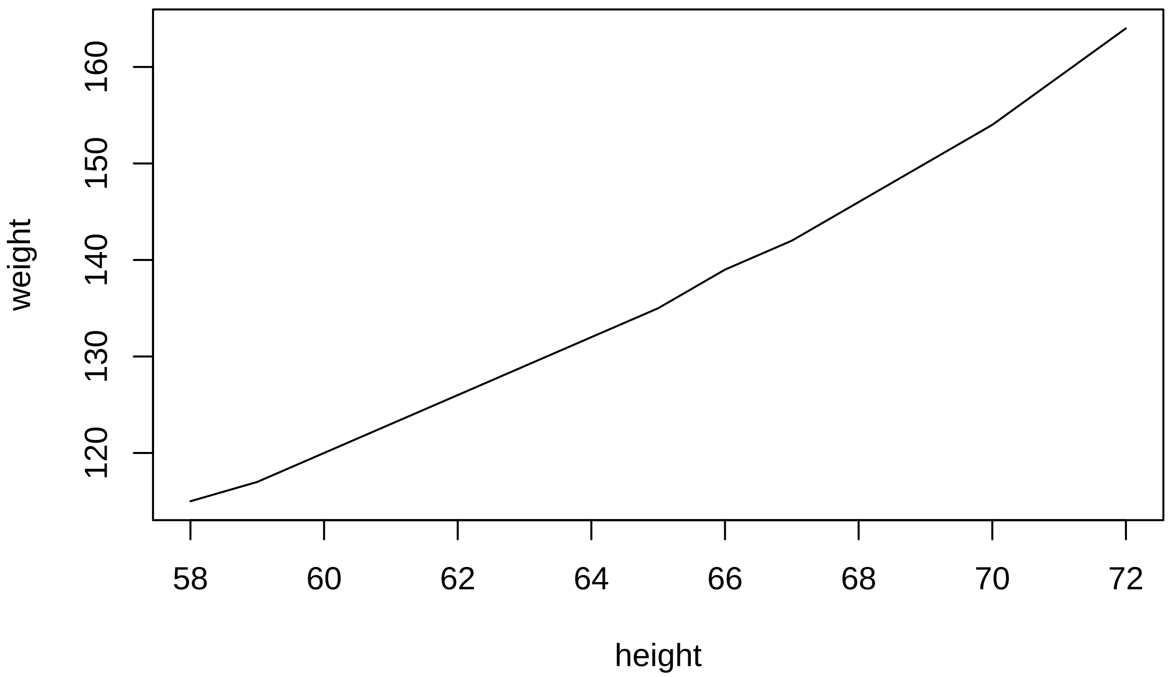
```r
women[4,2] # display row 4 column 2
```

```
## [1] 123
```

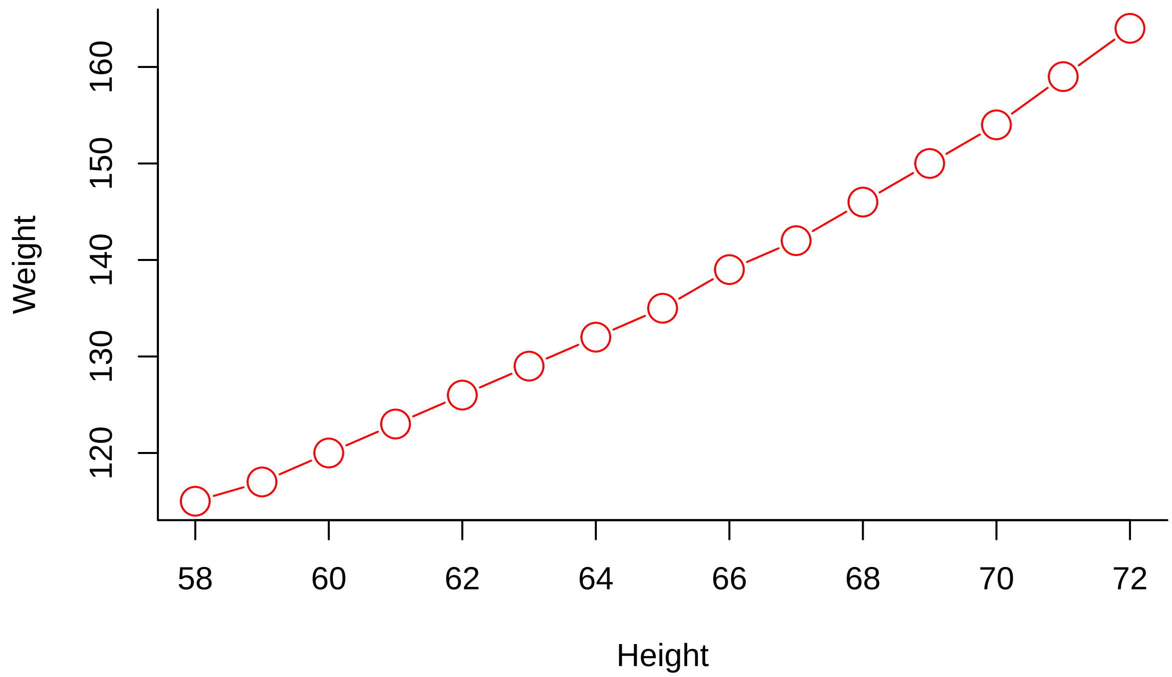
```r
plot(women$height, women$weight) # graph 'height' versus 'weight' for 'women'
```




```r
attach(women) # declare 'women' to be the default data set
plot(height, weight, type="l") # graph 'height' vs. 'weight', specifying the line type
```




```
plot(height, weight, type="b", col="red", bty="l", cex=2, xlab="Height", ylab="Weight")
```



P4. Importing your own data

```
schoolkids <- data.frame(height=rnorm(100), weight=rnorm(100),
    sex=sample(1:2, 100, replace=TRUE))
schoolkids            # look at the entire data set
```

```
##         height      weight sex
## 1    0.994858613 -1.10739913   2
## 2   -1.361242989 -1.36088680   2
## 3   -0.393041374 -0.01191303   2
## 4    0.170779218  1.46148744   1
## 5    0.430211543 -2.47266996   2
## 6   -0.763298351  0.13714048   1
## 7    0.872474903  0.50311276   2
## 8   -0.252777375  0.01744105   1
## 9   -0.573251416 -1.26152636   2
## 10  -0.191017082 -0.40979918   1
## 11  -0.162605965  0.58861391   1
## 12   1.279492437  1.30897673   1
## 13   0.309382536 -0.22422094   2
## 14  -0.287510366 -0.97967262   1
```

```
## 15   0.796088762 -0.11542627   1
## 16   1.092286933 -0.03411853   2
## 17  -0.016654187  0.29333187   2
## 18   1.829961234 -1.03260346   1
## 19   0.794455889 -1.95599632   1
## 20  -0.772706203 -0.77560081   2
## 21  -0.773729680 -0.10231106   2
## 22   0.888222985  2.31840316   1
## 23   0.835783874  0.31862030   1
## 24   3.127289390 -0.23618243   2
## 25  -0.369111592 -0.63747186   2
## 26  -0.527838645  2.53934737   1
## 27   0.574911178 -0.88837061   2
## 28   0.351226804  0.78281752   2
## 29  -0.621551765  1.21249175   2
## 30  -1.693475377  0.57375902   1
## 31   0.911220736  0.09431307   2
## 32   0.143396196  0.93770986   2
## 33  -0.146775757  2.12239372   2
## 34  -0.682440663  1.89850467   2
## 35  -1.599479830  1.10262123   2
## 36  -0.537746204  0.20740607   2
## 37  -0.763917408 -0.28371126   1
## 38   0.535094348  0.72748054   1
## 39   1.207536076  1.70414467   1
## 40  -0.303213195 -0.65646790   1
## 41   0.894367047 -1.15583857   2
## 42   1.323290311 -2.00232360   1
## 43   0.071094396  1.99143135   2
## 44  -0.696461871 -1.74618385   1
## 45  -0.252643461  0.59368615   2
## 46   0.342781917 -0.47151697   1
## 47   0.499799284 -1.31242166   1
## 48   0.009827627  1.36049826   2
## 49   2.121383357  0.49668248   2
## 50  -0.810644541 -0.78626420   2
## 51  -0.109123021 -0.23217445   2
## 52  -0.169543288  0.41700471   2
## 53   0.759168795  1.19733221   1
## 54   0.315846005  3.05991221   1
## 55  -1.646570900  3.10218149   2
## 56   0.452657345  1.08949521   1
## 57  -0.063036015  2.14835118   2
## 58   0.433615656  0.65253573   2
## 59   1.121792357  0.28886214   2
## 60   0.266183495 -0.18094793   2
## 61   1.393795582 -0.34466021   2
## 62  -0.882297963  0.23763324   1
```

```
## 63  -1.088812837  1.18596901   2
## 64  -0.198625940 -1.30968372   2
## 65  -1.027863811  2.70027714   1
## 66  -0.990563451  0.12442063   1
## 67   1.368447552 -1.37471097   1
## 68   2.245112734  0.40237741   2
## 69  -0.480431110  0.49937161   2
## 70   0.347338780 -0.23120193   1
## 71   0.795100108  0.04610843   1
## 72  -0.855282734 -0.37465430   2
## 73   0.854808296  1.51423777   1
## 74  -1.245184935 -0.10853691   1
## 75  -0.751207123  0.17284621   2
## 76   0.188254241 -0.69205934   2
## 77   1.190170968  0.42533786   2
## 78  -0.576083300 -0.15309103   1
## 79  -1.380768347  1.21405944   2
## 80  -1.550694145  1.24190360   1
## 81  -0.919286672  0.51612426   1
## 82  -0.360807276 -0.61382527   2
## 83   0.324054475 -0.01561877   1
## 84   1.190511511 -0.11396245   2
## 85  -0.252845810 -0.31661174   1
## 86   1.812372713  0.78121667   2
## 87  -0.330768557 -0.68609647   2
## 88   0.238181605  0.06898779   2
## 89   2.232106257  0.63446933   2
## 90   0.591753233  0.93345013   2
## 91  -1.468693745  0.35401263   1
## 92   0.405200028 -0.93182944   1
## 93  -1.179644801 -1.56892803   1
## 94  -0.076549056 -1.59863268   2
## 95   0.267096064  0.81781269   2
## 96  -0.968962878 -1.06748643   1
## 97   0.043604299  0.84681358   2
## 98   0.875895767  1.88589742   1
## 99  -1.071696360 -1.25049763   2
## 100  0.854150962  0.55334196   1
```

```r
names(schoolkids)       # list the variables
```

```
## [1] "height" "weight" "sex"
```

```r
attach(schoolkids)      # Make it the default data set
```

```
## The following objects are masked from women:
```

```
##
##     height, weight
```

```
summary(schoolkids)
```

```
##      height             weight             sex
##  Min.   :-1.69348   Min.   :-2.4727   Min.   :1.00
##  1st Qu.:-0.63677   1st Qu.:-0.6197   1st Qu.:1.00
##  Median : 0.02672   Median : 0.1308   Median :2.00
##  Mean   : 0.08776   Mean   : 0.1922   Mean   :1.56
##  3rd Qu.: 0.79535   3rd Qu.: 0.8251   3rd Qu.:2.00
##  Max.   : 3.12729   Max.   : 3.1022   Max.   :2.00
```

```
schoolkids$sex <- factor(sex)
# Deduce what each of these commands does:
schoolkids[2,4]
```

```
## NULL
```

```
schoolkids[,2]
```

```
##   [1] -1.10739913 -1.36088680 -0.01191303  1.46148744 -2.47266996
##   [6]  0.13714048  0.50311276  0.01744105 -1.26152636 -0.40979918
##  [11]  0.58861391  1.30897673 -0.22422094 -0.97967262 -0.11542627
##  [16] -0.03411853  0.29333187 -1.03260346 -1.95599632 -0.77560081
##  [21] -0.10231106  2.31840316  0.31862030 -0.23618243 -0.63747186
##  [26]  2.53934737 -0.88837061  0.78281752  1.21249175  0.57375902
##  [31]  0.09431307  0.93770986  2.12239372  1.89850467  1.10262123
##  [36]  0.20740607 -0.28371126  0.72748054  1.70414467 -0.65646790
##  [41] -1.15583857 -2.00232360  1.99143135 -1.74618385  0.59368615
##  [46] -0.47151697 -1.31242166  1.36049826  0.49668248 -0.78626420
##  [51] -0.23217445  0.41700471  1.19733221  3.05991221  3.10218149
##  [56]  1.08949521  2.14835118  0.65253573  0.28886214 -0.18094793
##  [61] -0.34466021  0.23763324  1.18596901 -1.30968372  2.70027714
##  [66]  0.12442063 -1.37471097  0.40237741  0.49937161 -0.23120193
##  [71]  0.04610843 -0.37465430  1.51423777 -0.10853691  0.17284621
##  [76] -0.69205934  0.42533786 -0.15309103  1.21405944  1.24190360
##  [81]  0.51612426 -0.61382527 -0.01561877 -0.11396245 -0.31661174
##  [86]  0.78121667 -0.68609647  0.06898779  0.63446933  0.93345013
##  [91]  0.35401263 -0.93182944 -1.56892803 -1.59863268  0.81781269
##  [96] -1.06748643  0.84681358  1.88589742 -1.25049763  0.55334196
```

```
schoolkids[2,]
```

```
##      height    weight sex
## 2 -1.361243 -1.360887   2
```

```
schoolkids[,1:3]
```

```
##          height      weight sex
## 1    0.994858613 -1.10739913   2
## 2   -1.361242989 -1.36088680   2
## 3   -0.393041374 -0.01191303   2
## 4    0.170779218  1.46148744   1
## 5    0.430211543 -2.47266996   2
## 6   -0.763298351  0.13714048   1
## 7    0.872474903  0.50311276   2
## 8   -0.252777375  0.01744105   1
## 9   -0.573251416 -1.26152636   2
## 10  -0.191017082 -0.40979918   1
## 11  -0.162605965  0.58861391   1
## 12   1.279492437  1.30897673   1
## 13   0.309382536 -0.22422094   2
## 14  -0.287510366 -0.97967262   1
## 15   0.796088762 -0.11542627   1
## 16   1.092286933 -0.03411853   2
## 17  -0.016654187  0.29333187   2
## 18   1.829961234 -1.03260346   1
## 19   0.794455889 -1.95599632   1
## 20  -0.772706203 -0.77560081   2
## 21  -0.773729680 -0.10231106   2
## 22   0.888222985  2.31840316   1
## 23   0.835783874  0.31862030   1
## 24   3.127289390 -0.23618243   2
## 25  -0.369111592 -0.63747186   2
## 26  -0.527838645  2.53934737   1
## 27   0.574911178 -0.88837061   2
## 28   0.351226804  0.78281752   2
## 29  -0.621551765  1.21249175   2
## 30  -1.693475377  0.57375902   1
## 31   0.911220736  0.09431307   2
## 32   0.143396196  0.93770986   2
## 33  -0.146775757  2.12239372   2
## 34  -0.682440663  1.89850467   2
## 35  -1.599479830  1.10262123   2
## 36  -0.537746204  0.20740607   2
## 37  -0.763917408 -0.28371126   1
## 38   0.535094348  0.72748054   1
## 39   1.207536076  1.70414467   1
## 40  -0.303213195 -0.65646790   1
## 41   0.894367047 -1.15583857   2
## 42   1.323290311 -2.00232360   1
## 43   0.071094396  1.99143135   2
## 44  -0.696461871 -1.74618385   1
```

```
## 45  -0.252643461  0.59368615   2
## 46   0.342781917 -0.47151697   1
## 47   0.499799284 -1.31242166   1
## 48   0.009827627  1.36049826   2
## 49   2.121383357  0.49668248   2
## 50  -0.810644541 -0.78626420   2
## 51  -0.109123021 -0.23217445   2
## 52  -0.169543288  0.41700471   2
## 53   0.759168795  1.19733221   1
## 54   0.315846005  3.05991221   1
## 55  -1.646570900  3.10218149   2
## 56   0.452657345  1.08949521   1
## 57  -0.063036015  2.14835118   2
## 58   0.433615656  0.65253573   2
## 59   1.121792357  0.28886214   2
## 60   0.266183495 -0.18094793   2
## 61   1.393795582 -0.34466021   2
## 62  -0.882297963  0.23763324   1
## 63  -1.088812837  1.18596901   2
## 64  -0.198625940 -1.30968372   2
## 65  -1.027863811  2.70027714   1
## 66  -0.990563451  0.12442063   1
## 67   1.368447552 -1.37471097   1
## 68   2.245112734  0.40237741   2
## 69  -0.480431110  0.49937161   2
## 70   0.347338780 -0.23120193   1
## 71   0.795100108  0.04610843   1
## 72  -0.855282734 -0.37465430   2
## 73   0.854808296  1.51423777   1
## 74  -1.245184935 -0.10853691   1
## 75  -0.751207123  0.17284621   2
## 76   0.188254241 -0.69205934   2
## 77   1.190170968  0.42533786   2
## 78  -0.576083300 -0.15309103   1
## 79  -1.380768347  1.21405944   2
## 80  -1.550694145  1.24190360   1
## 81  -0.919286672  0.51612426   1
## 82  -0.360807276 -0.61382527   2
## 83   0.324054475 -0.01561877   1
## 84   1.190511511 -0.11396245   2
## 85  -0.252845810 -0.31661174   1
## 86   1.812372713  0.78121667   2
## 87  -0.330768557 -0.68609647   2
## 88   0.238181605  0.06898779   2
## 89   2.232106257  0.63446933   2
## 90   0.591753233  0.93345013   2
## 91  -1.468693745  0.35401263   1
## 92   0.405200028 -0.93182944   1
```

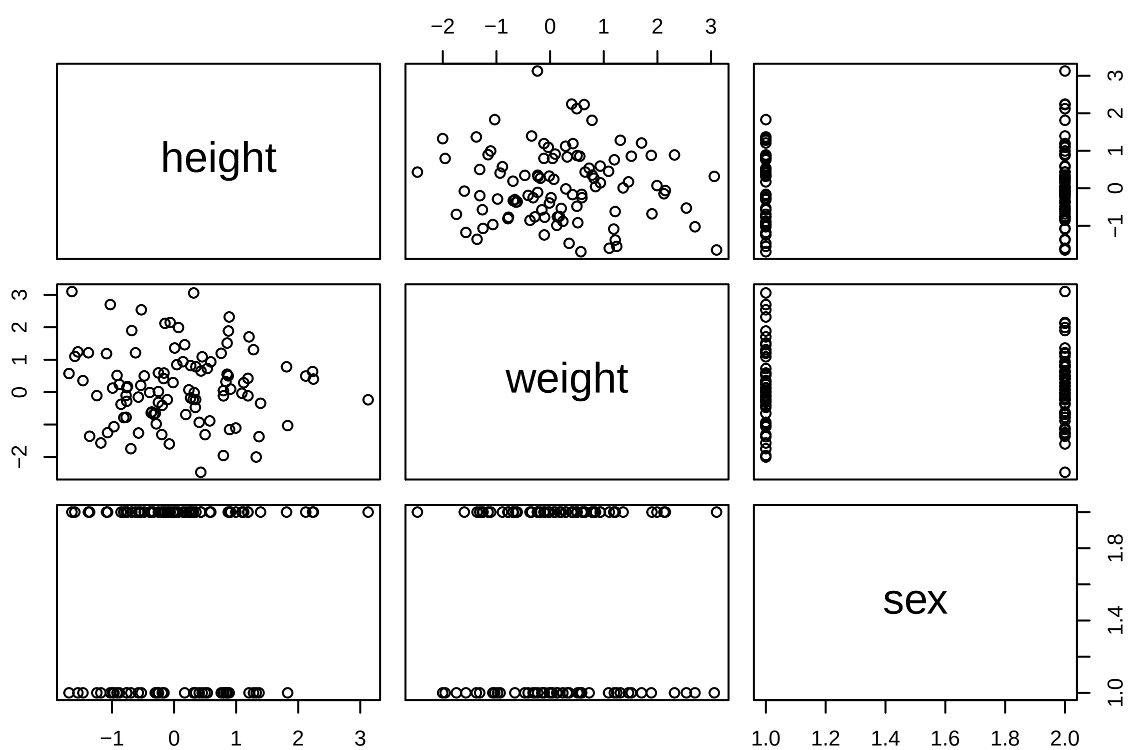
```
## 93  -1.179644801 -1.56892803   1
## 94  -0.076549056 -1.59863268   2
## 95   0.267096064  0.81781269   2
## 96  -0.968962878 -1.06748643   1
## 97   0.043604299  0.84681358   2
## 98   0.875895767  1.88589742   1
## 99  -1.071696360 -1.25049763   2
## 100  0.854150962  0.55334196   1
```

```
schoolkids[height>70,]
```

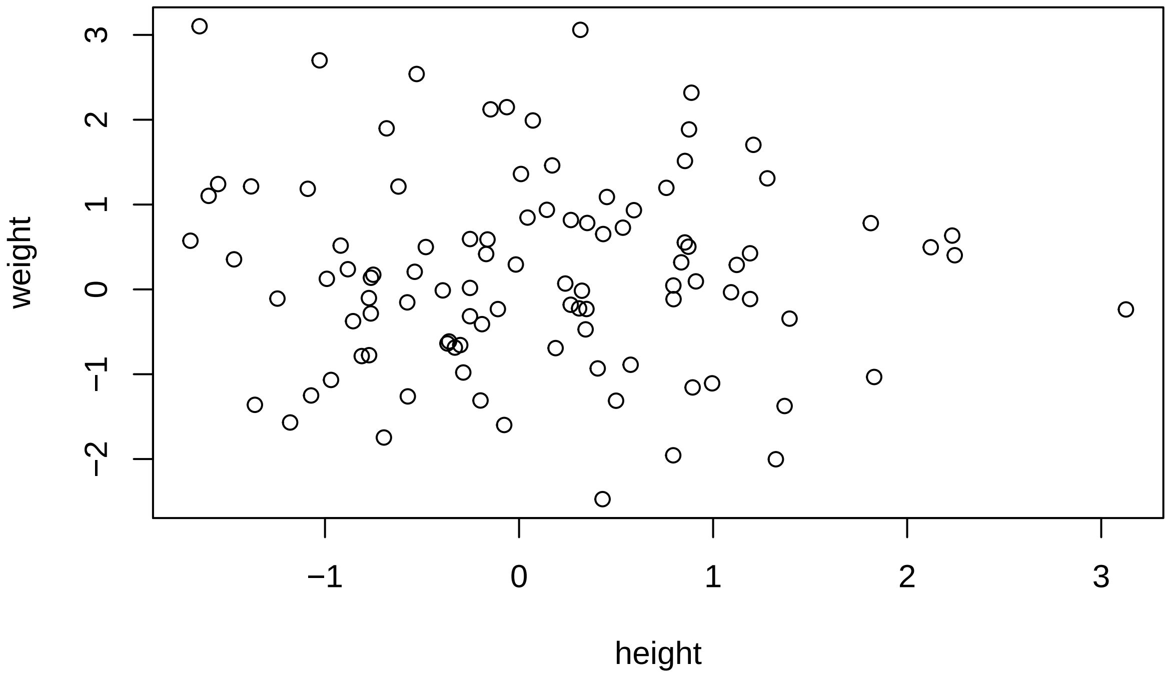
```
## [1] height weight sex
## <0 rows> (or 0-length row.names)
```

P5. Graphing data

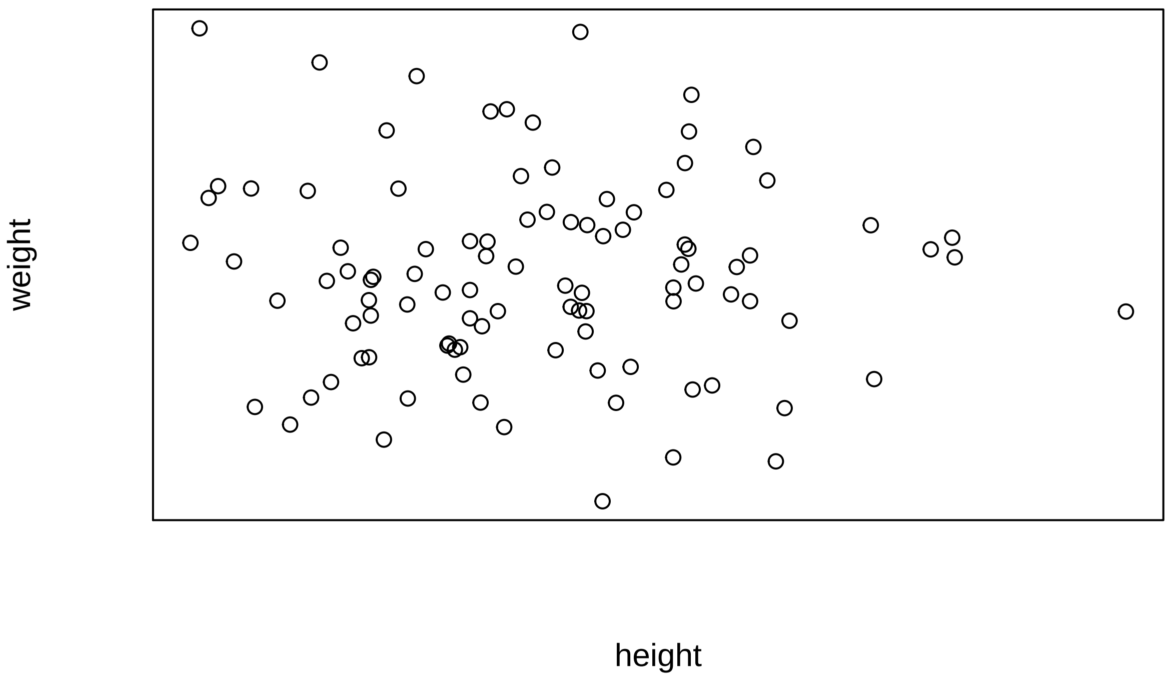
```
pairs(schoolkids)  # make pair-wise plots with columns in 'schoolkids'
```



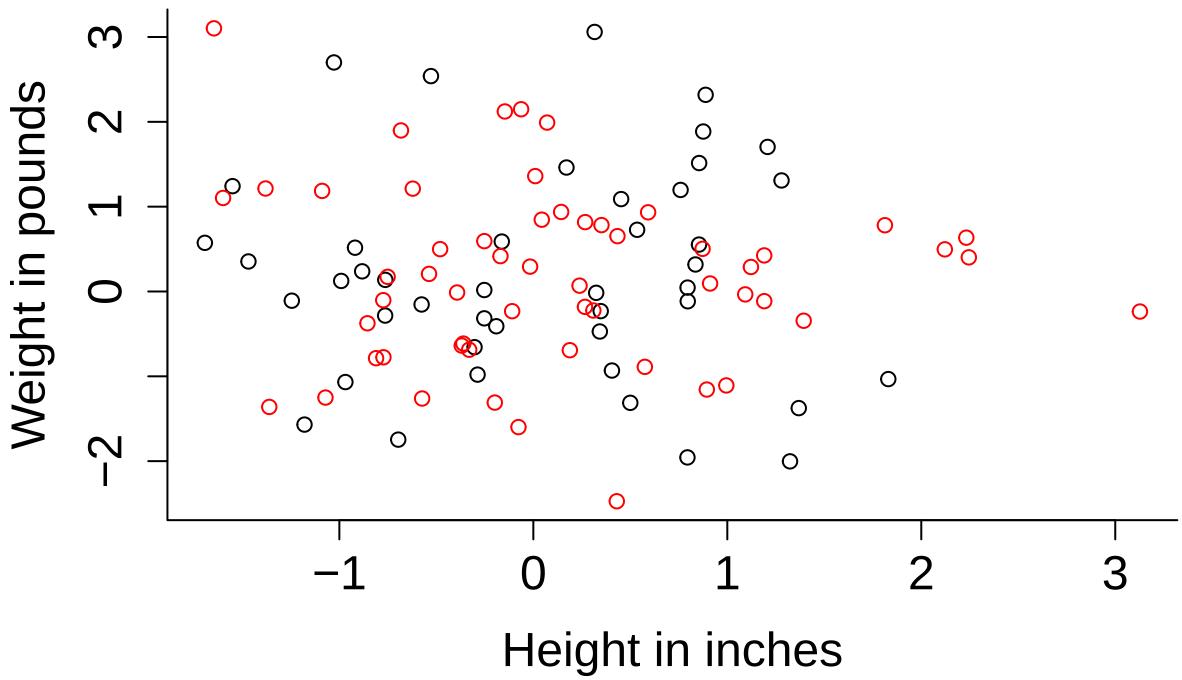
```r
plot(height, weight)          # Plot height (x-axis) against weight (y-axis)
```




```r
plot(height,weight,xaxt="n",yaxt="n")    # set up a blank graph with axes
```




```
plot(height,weight,type="n",
 bty="l",  # Remove the surrounding box
 xlab="Height in inches",  # set a label for the horizontal axis
 ylab="Weight in pounds",  # set a label for the vertical axis
 cex.axis=1.5,  # multiply the axis size by 1.5
 cex.lab=1.5)  # multiply the axis label size by 1.5
points(height[sex==1], weight[sex==1])  # add points for the 1st sex
points(height[sex==2],weight[sex==2],col="red")  # add red points for the 2nd sex
legend(52,160,c("Male","Female"),col=c("black","red"),pch=c(1,1) ) # add  a legend
```



P6. Simulating dice throws

```
die <- c(1:6)                # define the possible outcomes for a 6 sided die
die                          # have a look at your 'die'
```

```
## [1] 1 2 3 4 5 6
```

```
sample(die, 1, replace = TRUE)   # take a random 'throw of the of 'die'
```

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

```
sample(die, 10, replace = TRUE)      # observe 10 simulated dice throws
```

```
##  [1] 5 3 3 5 6 2 3 6 1 2
```

```
table(sample(die, size = 10000, replace = TRUE))
```

```
##
##    1    2    3    4    5    6
## 1616 1653 1664 1722 1694 1651
```

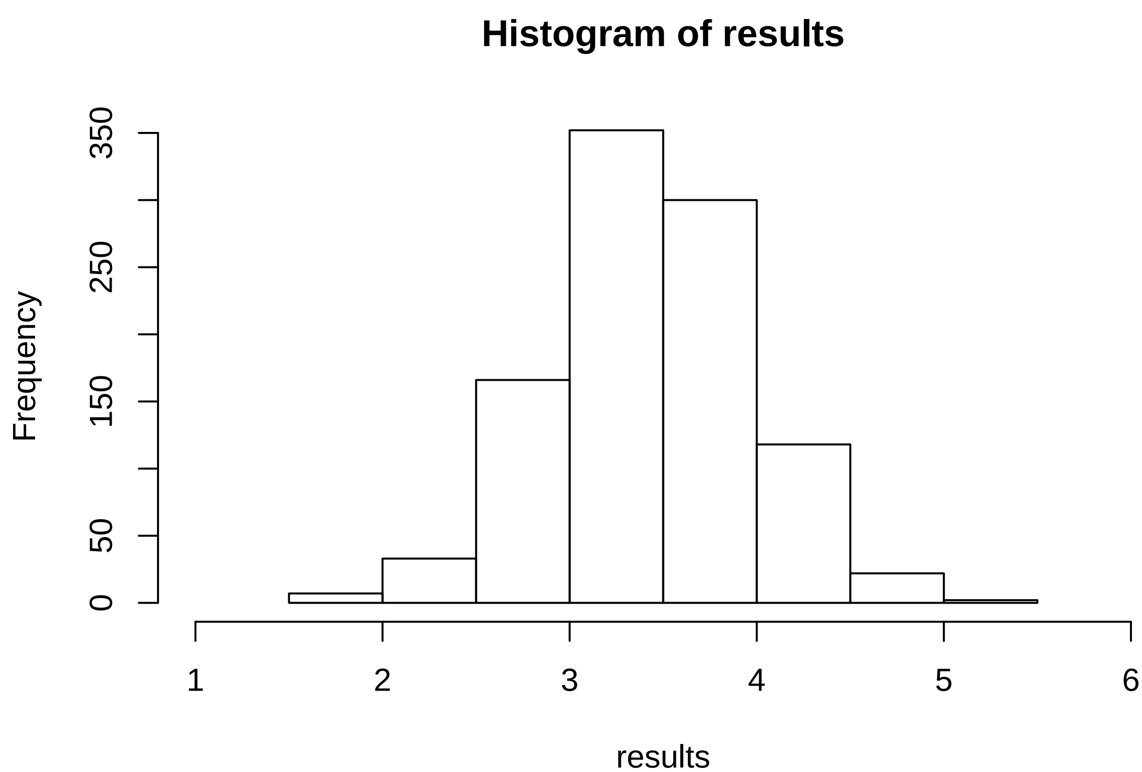
```r
# construct a frequency table for 10,000 dice throws
mean(sample(die, 10, replace = TRUE))    # observe the mean (average) of ten throws
```

```
## [1] 3.2
```

```r
# Repeat the previous command several times
# Observe that the sample mean is "scattered around" a "true mean" of 3.5
# Try this again, but with sample size increased to 100
# The "scatter around the true mean" decreases!
```

P7. Simulating data within a 'for loop'

```r
# In order to visualize the "scatter around a true mean",
# the following code will generate and store 1000 such means
results <- vector()                # set up space to hold results
length(results) <- 1000            # size it to hold exactly 1000 means
for (i in 1:1000) { # set up a 'for loop' that will iterate 1000 times
    # save the mean at location 'i' of 'results'
    results[i] <- mean(sample(die, 10, replace = TRUE))
}                 # repeat 'for loop' at the next value of 'i'
hist(results, xlim = c(1,6))     # A histogram of the 1000 means
```



```r
# Repeat, but with sample size increased to 100
# The "scatter around the true mean" decreases!
```