Introduction to R

Will Valdar
Oxford University

Pocket calculator

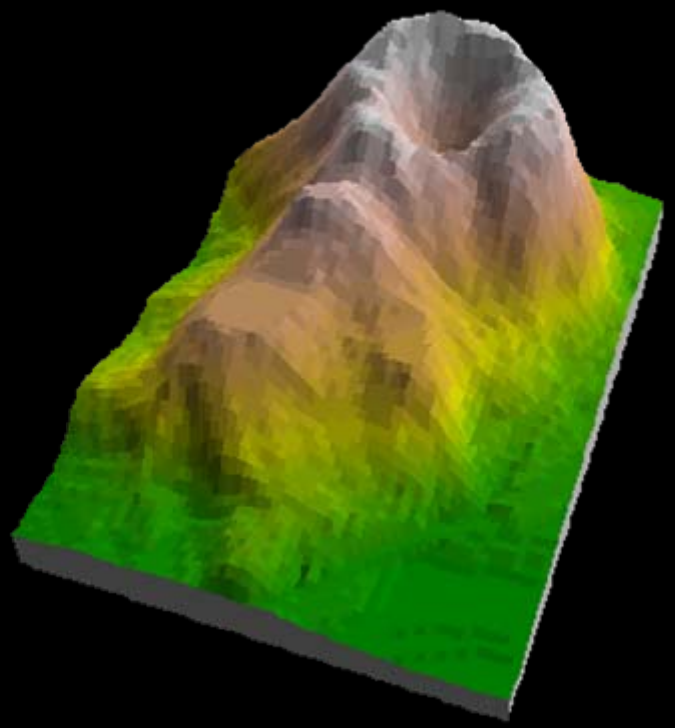
Data manipulation

Graphics

Pocket calculator

Data manipulation

Graphics



Pocket calculator

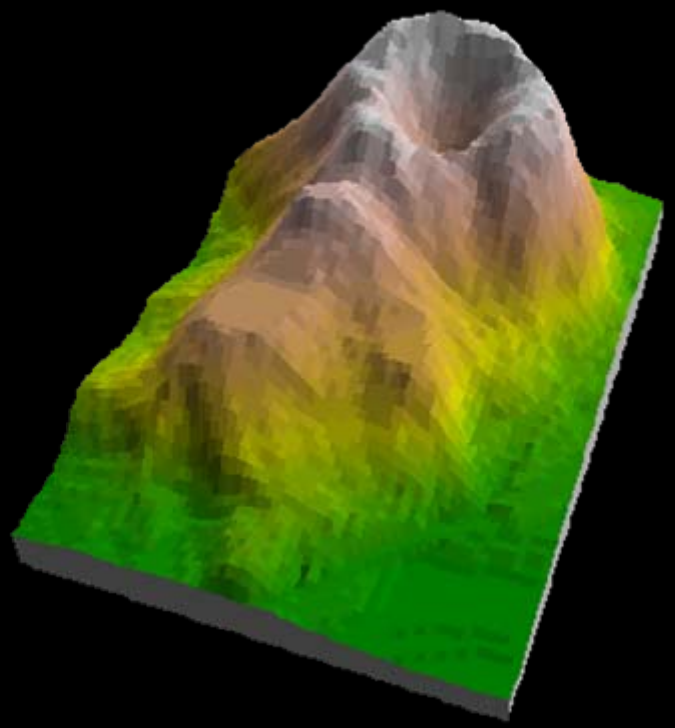
Data manipulation

Graphics

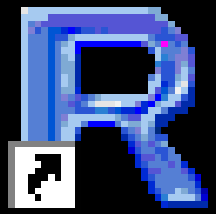
Statistical modelling

Computationally
Intensive Statistics

Programming



F:\valdar\TuesdayMorning\



R is a big calculator

Simple operations

```
3*4
20 + 4
20/4
9^2
log(100, base=10)
2.5e2 / 2.5
```

R is a big calculator

Simple operations

```
3*4
20 + 4
20/4
9^2
log(100, base=10)
2.5e2 / 2.5
```

Expressions

```
(3*4)/(20+4)
3*(4/20)+4
(5/9)*(48-32)
```

R is a big calculator

Variables

```
temp.f <- 48
temp.f
(5/9)*(temp.f-32)
temp.c <- (5/9)*(temp.f-32)
temp.c
```

R is a big calculator

```
Variables
   temp.f <- 48
   temp.f
   (5/9)*(temp.f-32)
   temp.c <- (5/9)*(temp.f-32)
   temp.c

Write your own functions
   will.fun <- function(x) { (5/9)*(x-32) }
   will.fun(48)
   will.fun(temp.f)
   thing <- will.fun(26)
```

Data types

```
class(will.fun)
class(temp.c)

my.name <- "Will"
class(my.name)
```

Data types

```
class(will.fun)
class(temp.c)

my.name <- "Will"
class(my.name)

Vectors
  a <- c("Will", "Ben")
  x <- c(1,2,3)
  sum(x)
  mean(x)
  range(x)
  var(x)
  1:3
```

Matrices

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 5 & 1 & 1 \\ 5 & 2 & 14 \\ 5 & 3 & -1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$$

```
A <- matrix(1:9, byrow=TRUE, ncol=3)
B <- rbind( c(5,1,1), c(5,2,14), c(5,3,-1) )
x <- c(7,8,9)
```

Matrices

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 5 & 1 & 1 \\ 5 & 2 & 14 \\ 5 & 3 & -1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$$

```
A <- matrix(1:9, byrow=TRUE, ncol=3)
B <- rbind( c(5,1,1), c(5,2,14), c(5,3,-1) )
x <- c(7,8,9)
```

```
t(A)
diag(A)
A * B
A %*% B
A + 1
```

```
median(x)
range(B)
max(x)
B.inv <- solve(B)
3*A
```

Matrices

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 5 & 1 & 1 \\ 5 & 2 & 14 \\ 5 & 3 & -1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$$

Subscripting is easy

```
A[1,1]
A[3,2]
x[2]
A[3,]
A[,3]
A[3, c(1,2)]
A[, c(1,2)]
A[2,2] <- 1975
```

Exercises 1

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 11 & 12 \\ 21 & 22 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 4 \\ 5.2 \end{bmatrix}$$

1) $(3.5+6)\times(7.9)^2$

2) $\mathbf{A}+\mathbf{B}$

3) $\bar{x}$

4) Convert **B** from Farenheit to Centigrade

5) $\mathbf{Ax}$

*6) $\mathbf{A}^{-1}\mathbf{Bx}$

Answers 1

```
A <- matrix(1:4, byrow=TRUE, ncol=2)
B <- cbind(c(11,21), c(12,22))
x <- c(4, 5.2)

1)          (3.5+6)*((7.9)^2)
2)          A + B
3)          mean(x)
4)          (5/9)*(B-32)
      or
            will.fun(B)
5)          A %*% x
6)          solve(A) %*% B %*% x
```

Random numbers

```
rnorm(10)
rnorm(10, mean=5, sd=2)
rnorm(100, mean=5, sd=2)
?rnorm
```

Random numbers

```
rnorm(10)
rnorm(10, mean=5, sd=2)
rnorm(100, mean=5, sd=2)

library(lattice)
histogram( rnorm(100, mean=5, sd=2) )
```

Simulation

```
args(sample)
sample(c("Heads", "Tails"), size=1)
```

Simulation

```
args(sample)
sample(c("Heads", "Tails"), size=1)

0 allele does nothing, 1 makes you taller

sample(0:1, size=1)
sample(0:1, size=2, replace=TRUE)
sum( sample(0:1, size=2, replace=TRUE ) )
```

Simulation

```
  args(sample)
  sample(c("Heads", "Tails"), size=1)

0 allele does nothing, 1 makes you taller

  sample(0:1, size=1)
  sample(0:1, size=2, replace=TRUE)
  sum( sample(0:1, size=2, replace=TRUE ) )

Function to make a fake genotype
  make.genotype <- function(){ sum(sample(0:1,
  size=2, replace=TRUE) )}
  make.genotype()
```

Simulating the genetic basis of height

```
gmat <- matrix(ncol=5, nrow=100)
gmat
```

Simulating the genetic basis of height

```
gmat <- matrix(ncol=5, nrow=100)
gmat
for (person in 1:100) {
for (gene in 1:5) {
    gmat[person, gene] <- make.genotype()
  }}
gmat
```

Simulating the genetic basis of height

```
gmat <- matrix(ncol=5, nrow=100)
gmat
for (person in 1:100) {
for (gene in 1:5) {
    gmat[person, gene] <- make.genotype()
  }}
gmat
head(gmat)
```

Simulating the genetic basis of height

```
  gmat <- matrix(ncol=5, nrow=100)
  gmat
  for (person in 1:100) {
  for (gene in 1:5) {
     gmat[person, gene] <- make.genotype()
    }}
  gmat
  head(gmat)
Genotype of person 1:
  gmat[1,]
```

Simulating the genetic basis of height

```
  gmat <- matrix(ncol=5, nrow=100)
  gmat
  for (person in 1:100) {
  for (gene in 1:5) {
     gmat[person, gene] <- make.genotype()
    }}
  gmat
  head(gmat)
Genotype of person 1:
  gmat[1,]
Height of person 1:
  gmat[1,] %*% c( 1, 1, 1, 1, 1)
  gmat[1,] %*% c( 3, 5, 2, 3, 1)
```

Simulating the genetic basis of height

All heights

```
gmat %*% c( 3, 5, 2, 3, 1)

histogram(gmat %*% c( 3, 5, 2, 3, 1))
```

Further reading

Books:

Introductory Statistics with R

by Peter Dalgaard

Modern Applied Statistics with S

by Venables and Ripley

Web sites:

http://www.r-project.org/

http://wiki.r-project.org/rwiki/doku.php

http://news.gmane.org/gmane.comp.lang.r.general

Graph gallery:

http://addictedtor.free.fr/graphiques/

	A	B	C	D	E	F	G	H	I
1	sex	weight	country	age	snp.A	snp.B	family	phenotype	status
2	F	91.29	Europe	42	0	1	Family3	-2.6134576	FALSE
3	M	154.22	US	42	1	2	Family6	-7.7237116	FALSE
4	F	95.74	US	40	2	0	Family1	3.2361416	TRUE
5	F	110.25	Australia	30	0	1	Family10	-0.026201	TRUE
6	M	170.47	Australia	36	0	2	Family1	-11.132778	FALSE
7	M	127.57	US	45	1	1	Family6	-3.6016196	FALSE
8	M	125.56	Europe	44	1	1	Family9	-7.0376314	FALSE

data.frame

	A	B	C	D	E	F	G	H	I
1	sex	weight	country	age	snp.A	snp.B	family	phenotype	status
2	F	91.29	Europe	42	0	1	Family3	-2.6134576	FALSE
3	M	154.22	US	42	1	2	Family6	-7.7237116	FALSE
4	F	95.74	US	40	2	0	Family1	3.2361416	TRUE
5	F	110.25	Australia	30	0	1	Family10	-0.026201	TRUE
6	M	170.47	Australia	36	0	2	Family1	-11.132778	FALSE
7	M	127.57	US	45	1	1	Family6	-3.6016196	FALSE
8	M	125.56	Europe	44	1	1	Family9	-7.0376314	FALSE

data.frame

	A	B	C	D	E	F	G	H	I
1	sex	weight	country	age	snp.A	snp.B	family	phenotype	status
2	F	91.29	Europe	42	0	1	Family3	-2.6134576	FALSE
3	M	154.22	US	42	1	2	Family6	-7.7237116	FALSE
4	F	95.74	US	40	2	0	Family1	3.2361416	TRUE
5	F	110.25	Australia	30	0	1	Family10	-0.026201	TRUE
6	M	170.47	Australia	36	0	2	Family1	-11.132778	FALSE
7	M	127.57	US	45	1	1	Family6	-3.6016196	FALSE
8	M	125.56	Europe	44	1	1	Family9	-7.0376314	FALSE

```
data <- read.delim("F:valdar/disease.txt")
class(data)
head(data)
```

data.frame

	A	B	C	D	E	F	G	H	I
1	sex	weight	country	age	snp.A	snp.B	family	phenotype	status
2	F	91.29	Europe	42	0	1	Family3	-2.6134576	FALSE
3	M	154.22	US	42	1	2	Family6	-7.7237116	FALSE
4	F	95.74	US	40	2	0	Family1	3.2361416	TRUE
5	F	110.25	Australia	30	0	1	Family10	-0.026201	TRUE
6	M	170.47	Australia	36	0	2	Family1	-11.132778	FALSE
7	M	127.57	US	45	1	1	Family6	-3.6016196	FALSE
8	M	125.56	Europe	44	1	1	Family9	-7.0376314	FALSE

```
  sex weight   country age snp.A snp.B   family    phenotype status
1   F  91.29    Europe  42     0     1  Family3  -2.61345764  FALSE
2   M 154.22        US  42     1     2  Family6  -7.72371165  FALSE
3   F  95.74        US  40     2     0  Family1   3.23614160   TRUE
4   F 110.25 Australia  30     0     1 Family10  -0.02620101   TRUE
5   M 170.47 Australia  36     0     2  Family1 -11.13277756  FALSE
6   M 127.57        US  45     1     1  Family6  -3.60161956  FALSE
```

a variable with other variables hanging off it

```
str(data)
'data.frame':   400 obs. of  9 variables:
 $ sex      : Factor w/ 2 levels "F","M": 1 2 1 1 2  ...
 $ weight   : num   91.3 154.2  95.7 110.2 170.5 ...
 $ country  : Factor w/ 3 levels "Australia","Europe",..: 2 3 ...
 $ age      : int  42 42 40 30 36 45 44 38 40 34 ...
 $ snp.A    : int  0 1 2 0 0 1 1 0 2 1 ...
 $ snp.B    : int  1 2 0 1 2 1 1 0 1 1 ...
 $ family   : Factor w/ 10 levels "Family1",..: 4 7 ...
 $ phenotype: num   -2.6135  -7.7237   3.2361  ...
 $ status   : logi  FALSE FALSE  TRUE  TRUE FALSE FALSE …
```

a variable with other variables hanging off it

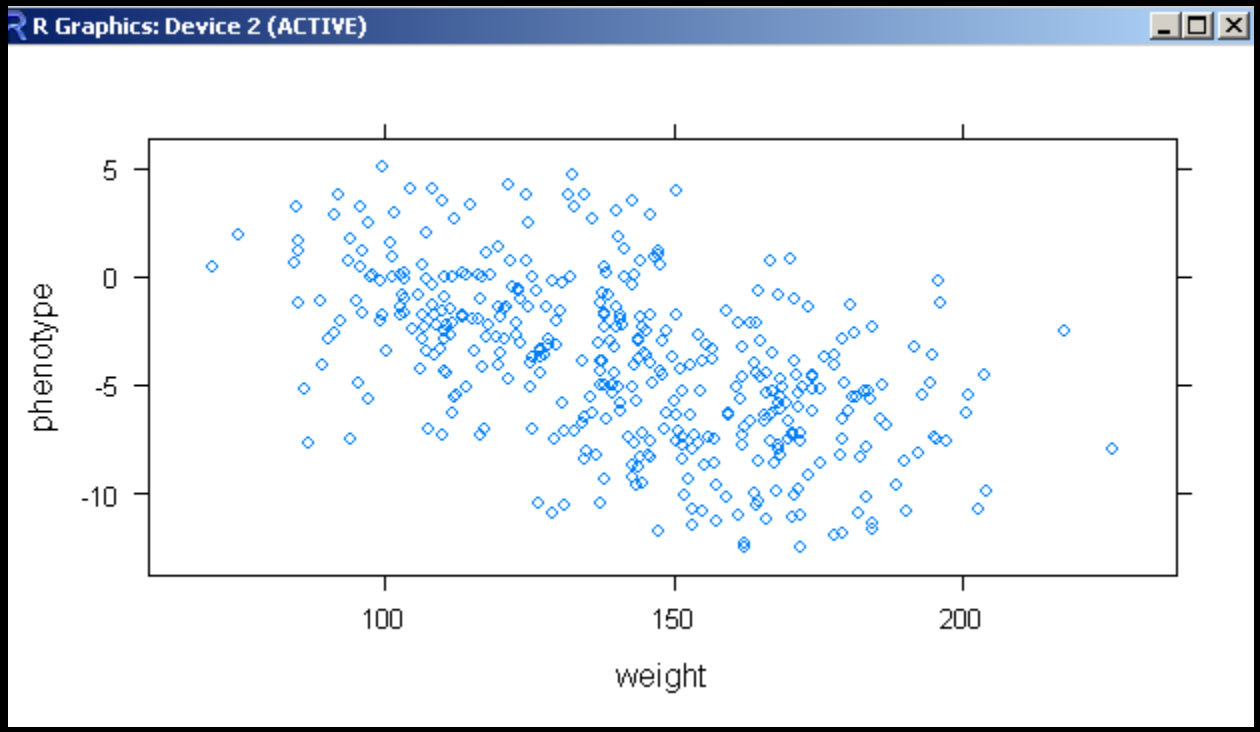
```
str(data)
'data.frame':   400 obs. of  9 variables:
 $ sex      : Factor w/ 2 levels "F","M": 1 2 1 1 2  ...
 $ weight   : num   91.3 154.2  95.7 110.2 170.5 ...
 $ country  : Factor w/ 3 levels "Australia","Europe",..: 2 3 ...
 $ age      : int  42 42 40 30 36 45 44 38 40 34 ...
 $ snp.A    : int  0 1 2 0 0 1 1 0 2 1 ...
 $ snp.B    : int  1 2 0 1 2 1 1 0 1 1 ...
 $ family   : Factor w/ 10 levels "Family1",..: 4 7 ...
 $ phenotype: num   -2.6135  -7.7237   3.2361  ...
 $ status   : logi  FALSE FALSE  TRUE  TRUE FALSE FALSE …
```

```
data$weight
data$weight[3:10]
mean(data$weight)
```

```
library(lattice)
```

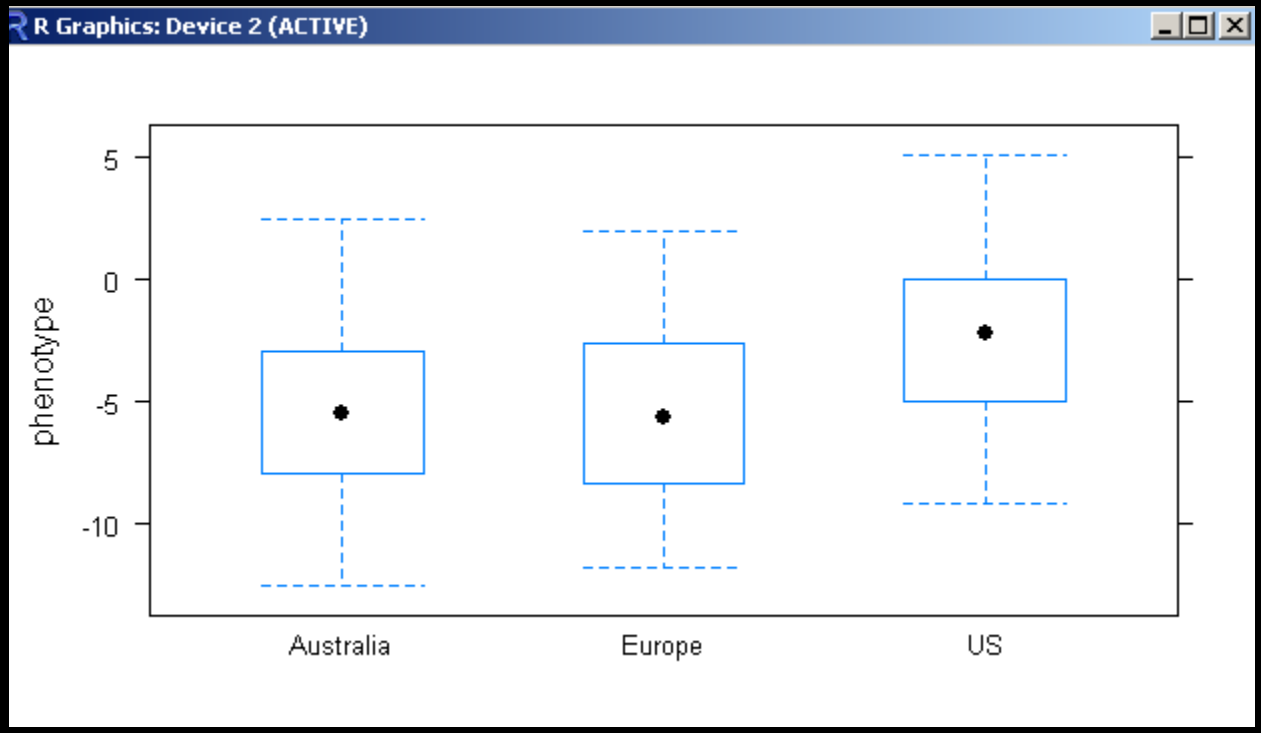
xyplot()

```
xyplot( phenotype ~ weight, data=data)
```



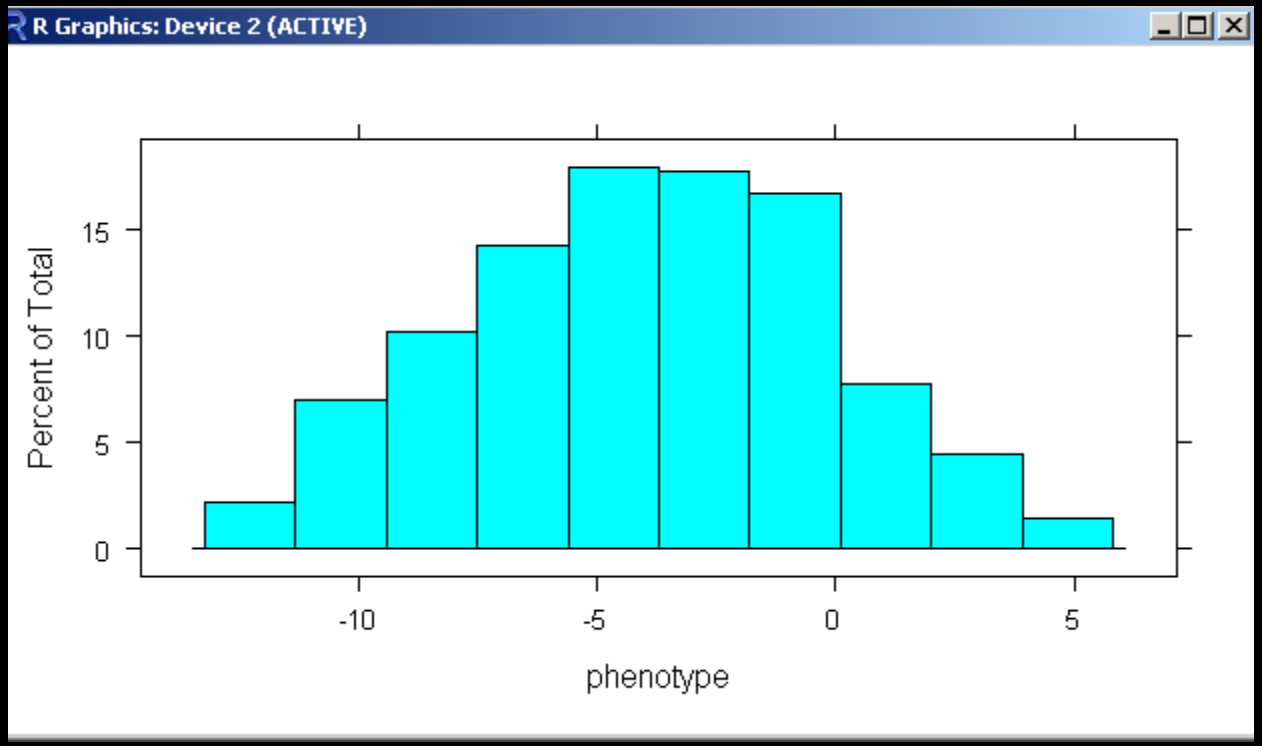
bwplot() , ie, box and whisker plot

```
bwplot( phenotype ~ country, data=data)
```



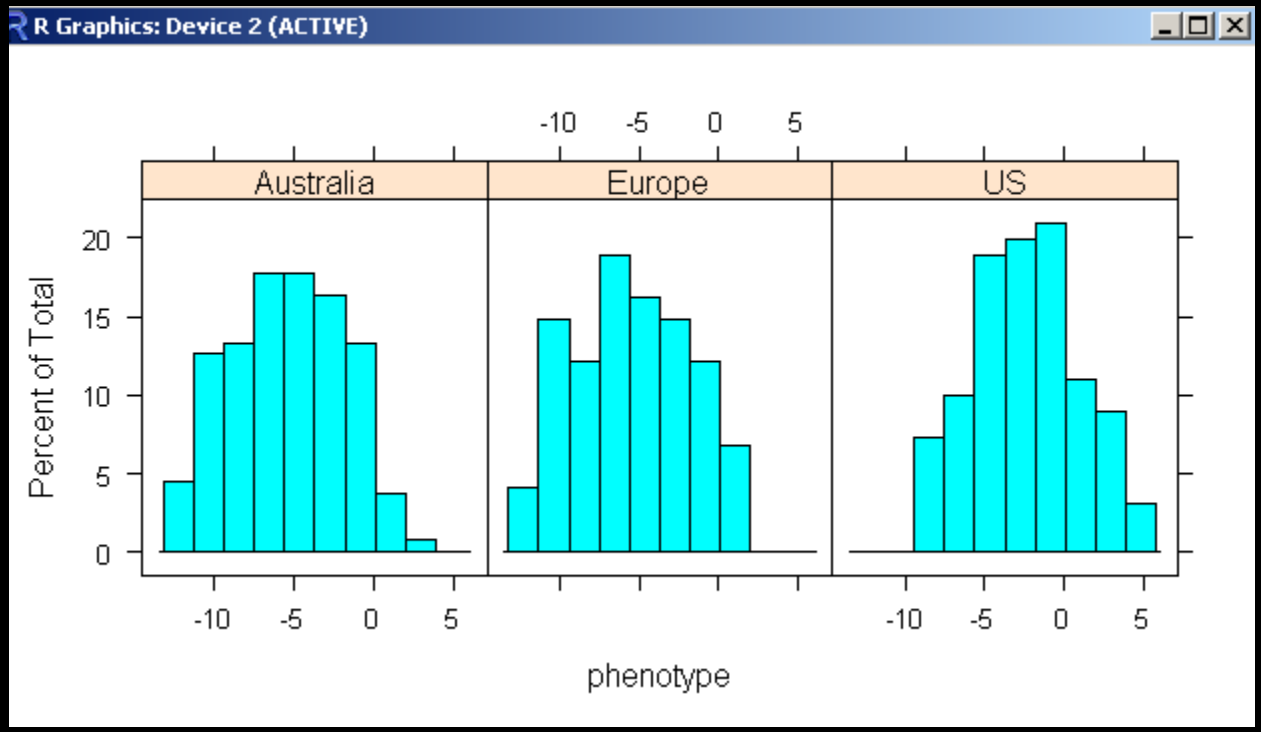
histogram()

```
histogram( ~ phenotype, data=data)
```



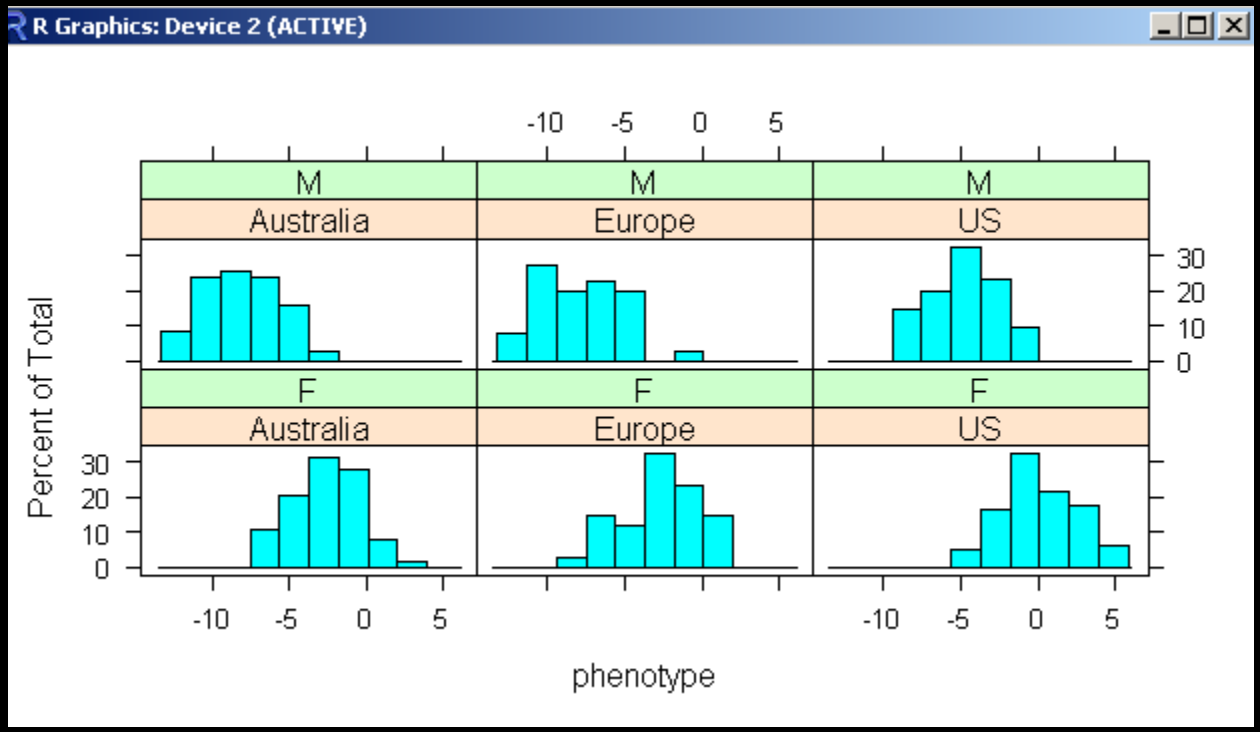
Conditioning

```
histogram( ~ phenotype | country, data=data)
```



Conditioning

```
histogram( ~ phenotype | sex * country, data=data)
```



Conditioning

Examples:

```
xyplot( phenotype ~ weight | sex, data=data)
bwplot( phenotype ~ country | snp.A, data=data)
```

Exercises 2

1) What do you think the status variable means and how is it defined? *(conditional histogram)*

2) How does snp.B affect the phenotype? *(cond. hist)*

3) Is there an effect of age on the phenotype? Are you sure? (*conditional xyplot*)

4) Does weight affect the phenotype? Does sex? Does sex affect weight? Does weight affect sex? What's going on here? *(conditional xyplot)*

Answers 2

1) `histogram( ~ phenotype | status, data=data)`

2) `xyplot(phenotype ~ snp.B, data=data)`
   `histogram( ~ phenotype | snp.B, data=data)`

3) `xyplot(phenotype ~ age, data=data)`
   `xyplot(phenotype ~ age | sex, data=data)`

4) `bwplot(phenotype ~ sex, data=data)`
   `xyplot(phenotype ~ weight, data=data)`
   `xyplot(phenotype ~ weight | sex, data=data)`

Answers 2: question 4