What is possible in R?

Almost everything:

- Statistics (tests, modeling, exploratory: PCA, clustering of the data)
- High throughput biological data:

  gene expression on arrays (Affymetrix, Agilent, custom arrays etc), qPCR, SAGE, flow cytometry, high throughput sequencing, microRNA, RNAi screens, mass spec data, CGH, Chip-chip, genome-wide SNP (Affymetrix, Ilumina)
- population genetics, sequence logos, google maps ... and tons of other small tools
- R-end in other software: SPSS, gene expression software,
- Graphics R graph gallery (http://addictedtor.free.fr/graphiques/)

Today

- vectors - names
- mode factor and function tapply()
- data frames
- matrices and arrays
- operations on matrices/data frames: apply()
- reading-in data from a text file
- writing data in a text file

Names

Should not start with number, should not contain blanks ",","_"

Case sensitivity: "Name" is different from "name"

It is allowed to give names only to certain elements of a vector

Avoid doubled names

Vector may be subset by names

Names may be subset, too

Names may be subset, too

```
> names(vector)

> names(vector) <- c(name1, name2...)


> vector[c(name1,name2...)]

> names(vector)[2] <- "newname"
```

Factor

Factor is another data mode. It is similar to character, but it is used in calculations as qualitative variable

```
> size <- factor(c("small","medium","big", "small",
"big"))

> size

[1] small  medium big  small  big

Levels: big medium small
```

tapply() allows to compute function values for each factor level separately

```
>tapply(vector, factor, function)

>applesFromTrees <- c(45,234,240,5,120)

>tapply(applesFromTrees,size,mean)
```

Data frame

Contains columns of data

Similar to spreadsheet: each column is a vector of data of the same mode

But different columns may have different modes

organism	genomeSizeBp	estGeneCount	zoology	inOurLab
“human”	3000000000	20000	Vertebrate	TRUE
“mouse”	3000000000	20000	Vertebrate	TRUE
“yeast”	12100000	6034	Unicellular	FALSE
“roundworm”	97000000	19099	Invertebrate	FALSE
“fruit fly”	135600000	13061	Invertebrate	TRUE

```
comparativeGenomics<-data.frame(organism=c ( “human”,”mouse”....),
                    genomeSizeBp = c(3000000000,3000000000,...),
                    estGeneCount=c(20000,20000,...),
                    zoology =
                    factor(c(“Vertebrate”,”Vertebrate”...)),
                    inOurLab = c(TRUE,TRUE,FALSE,...)
                         )
```

Working with dataframes

Accessing a column by name

```
comparativeGenomics$zoology
```

Accessing an element from a column

```
comparativeGenomics$zoology[2]
```

Also by indexes `dataframe[rows,columns]`

first row

```
comparativeGenomics[1,]
```

third column

```
comparativeGenomics[,3]
```

third elem. from second row

```
comparativeGenomics[2,3]
```

```
comparativeGenomics[2:4,3]
```

and by names

```
comparativeGenomics[“inOurLab”,1:2]
```

Reading data from a file

Reading data from a file with tabular data format. Data is read-in as a data frame:

```
data<-read.table(file,header=FALSE, skip = 0, sep = " ",
dec=".",...)

read.csv(), read.csv2(),read.delim()...
```

Reading data from a text file or console:

```
scan()
```

Writing data in a text file:

```
write.table()
```

Arrays and matrices

Matrices are two dimensional arrays

In matrices data are organized in rows and columns

All the data MUST have the same mode

All mathematical operations on matrices

```
> a<-matrix(1:9,nrow=3)
> a
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

Matrices and dataframes may be converted one into another

```
as.matrix()
as.data.frame()
```

```
> d<-array(1:24,dim=c(2,3,4))
> d
, , 1

     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

, , 2

     [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12

, , 3

     [,1] [,2] [,3]
[1,]   13   15   17
[2,]   14   16   18

, , 4

     [,1] [,2] [,3]
[1,]   19   21   23
[2,]   20   22   24
```

Data frames and matrices

Apply a function to the whole data frame

```
mean(data.frame)
```

Apply a function to a subset

```
mean(data.frame[1:3,4])
```

Apply a function to each row (1) or column (2) separately

```
apply(data.frame,1,mean)

apply(data.frame,2,mean)
```

Working with dataframes and matrices

Adding columns or rows

```
cbind(data.frame,vector)
rbind (matrix, vector)
```

Adding/changing rownames and column names

```
colnames (matrix)
colnames (matrix) <- c(...)
colnames (matrix)[3:5] <- c(...)
rownames (matrix)
```

Transposing matrix/data frame

```
t(data.frame)
```

Checking dimensions

```
dim(data.frame)
```

Sub-setting by row and column names

```
data.frame[c("row2","row40") , c("column1","column3")]
```