Reading in Data

Where's the spreadsheet?

- Statistical packages normally have a spreadsheet
- R has a minimal-but-usable spreadsheet
- More emphasis on data generated/curated externally
- Very powerful data import tools

Vectors

- Everything is based on vectors
- A vector is a series of values of the same type
- We've already seen 'character', 'numeric' and 'logical' vectors

```
x <- c('apple', 'banana', 'pear',)
y <- c(5,8,3,6,7,3,5)
z <- c(TRUE, FALSE,TRUE,FALSE,FALSE)
```

Lists and Data Frames

- In R, a table of data that's all one vector must be all one type
- Your typical excel file isn't like that
- List - A collection of any assorted objects
- Dataframe - A list of assorted vectors all the same length
- Vectors (columns) must also have unique names

Factor

- Special type of vector for categorical data

```
x <- c(1,1,1,1,2,2,2,3,3,4,4,4,4)
y <- as.factor(x)

a <- c(1,2,3,4)
b <- c(2,4,6,8)
levels <- factor(c("A","B","A","B"))
bubba <- data.frame(first=a, second=b, f=levels)

bubba$first
bubba$second
bubba$f
```

Attributes

- Attributes are information specific to an object

Dimensions

```
a<-1:100
dim(a)<-c(10,10)
a[50]
a[10,5]
```

Row and column names

```
colnames(a) <- letters[1:10]
rownames(a) <- letters[1:10]
a[‘e’,’j”]
```

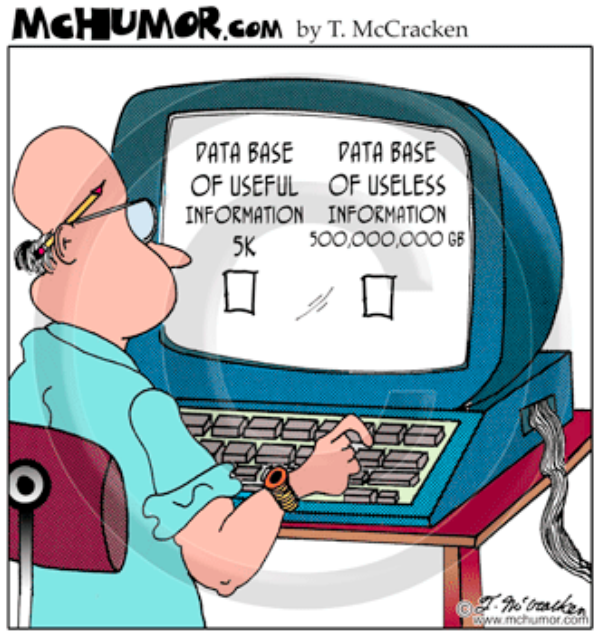
Attributes

- **Attributes can be anything including your own notes**

attr(a,’comment’)<-’deeply dubious data’

Editing and Entering data in R

- **Editing or entering data in R**
- **You can make a data frame with data.frame()**




```
dogs<-data.frame() # empty
edit(dogs) # !
dogs<-edit(data.frame())
```

Read.table() function

- Most common way to input tabular data, eg Excel (although other possibilities exist)
- Save data as tab-or comma separated text
- One sheet per file
- Use a version without too much extraneous junk
- Simple row and column labels, one line header

Read.table() function

- mydf <- read.table(‘mydata.txt’)
- Defaults only work in the simplest cases
- Safer to be explicit for general use

http://rcourse.iop.kcl.ac.uk/S3/regresssionof.txt

Read.table() function

```
cfdata<- read.table('http://rcourse.iop.kcl.ac.uk/S3/regresssionof.txt')

cfdata<- read.table('http://rcourse.iop.kcl.ac.uk/S3/regresssionof.txt',
    header=TRUE)

cfdata<- read.table('http://rcourse.iop.kcl.ac.uk/S3/regresssionof.txt',
    header=TRUE, sep="\t")

cfdata<- read.table('http://rcourse.iop.kcl.ac.uk/S3/regresssionof.txt',
    header=TRUE, sep="\t", as.is=TRUE)
```

Making a dataframe from your data

```
cfdata<-data.frame(cfdata,strain=factor(substr(cfdata$ID,1,1)))

plot(oftim~strain,cfdata)

summary(aov(oftim~strain,cfdata))

write.table(cfdata, 'test.txt')
```

foreign package

- Installed but not loaded by default

library(foreign)

- Methods for importing data files from most stats packages (e.g. SAS,SPSS,STATA, not Statistica)

- Methods for importing database files (e. .DBF xbase, foxpro, clipper, etc)

anx34567<- read.dta('http://rcourse.iop.kcl.ac.uk/S3/anx34567nosibs.dta', convert.factors=FALSE)

- Corresponding export methods to output the data.

foreign package

- Data may need cleanup
- Missing value codes often differ

```
anx34567$nest1[anx34567$nest1==’.’]<-NA
anx34567$nest1<-as.numeric(anx34567$nest1)
```

- Type coersion is often all you need

```
anx34567$nest2<-as.numeric(anx34567$nest2)
```

Microsoft Excel and Access

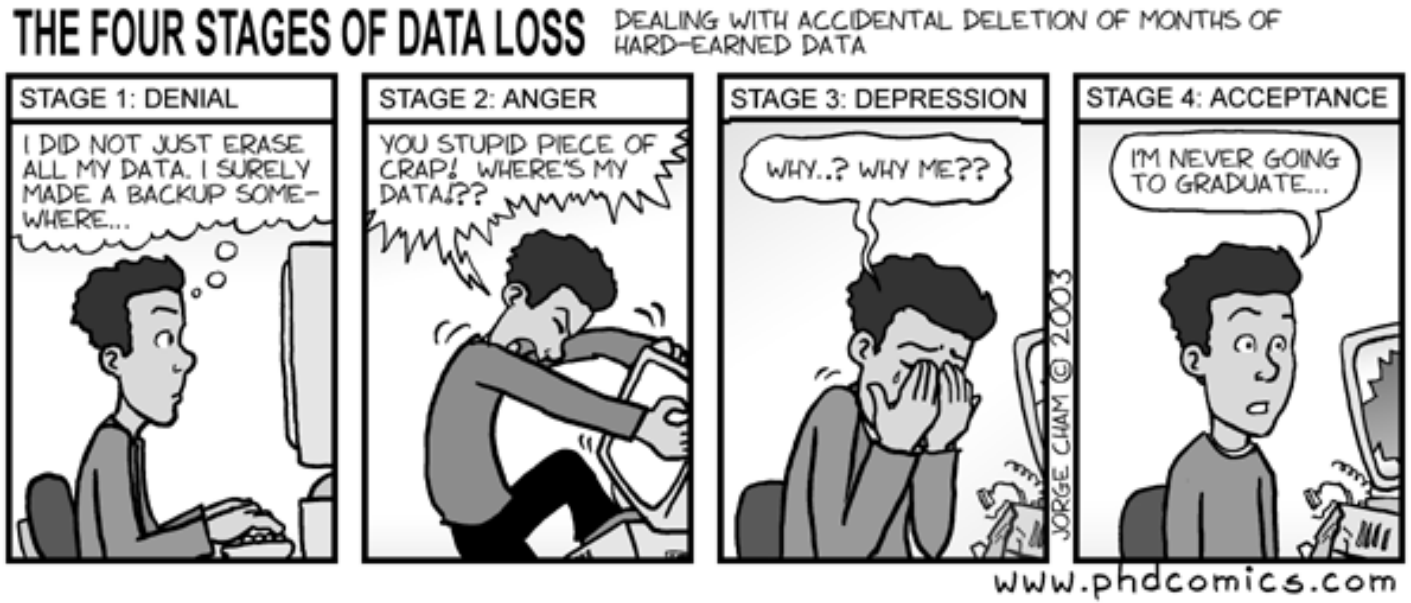
- RODBC R Open Database Connectivity
- Easy to read in whole Excel or Access tables

```
install.packages("RODBC")
library(RODBC)
download.file(url='http://rcourse.iop.kcl.ac.uk/2011/2tue/session1/genotest.xls',
    'genotest.xls',mode='wb')

myexcelchannel <- odbcConnectExcel("genotest.xls")

sqlTables(myexcelchannel)
sheet1 <- sqlFetch(myexcelchannel, "ABC")
sheet2 <- sqlFetch(myexcelchannel, “DEF")
sheet3 <- cbind(sheet1, sheet2)
```

Keeping your data under control



- Both workspace and command history can be saved so you can restart where you left off

savehistory(file = “file_name.Rhistory")
save(file="d:/file_name.RData")

- **This gives a false sense of security**

Why not write your own script?

- A much better strategy is to keep a log of your commands
- Use a text editor and record commands as you go along
- ### Annotate the script

Problems

1: (i) Create a matrix of 1:676 with dimensions 26 x 26

(ii) Add row names and column names of ‘a-z’ and ‘a-z’

(iii) Find the six numbers that relate to the six positions in the matrix:

‘If it is to be so’ (‘i’,’f’ ‘i’,’t’ ‘i’,’s’, ‘t’,’o’ ‘b’,’e’ ‘s’,’o’)

Problems

2: (i) Load in data from

http://rcourse.iop.kcl.ac.uk/2011/2tue/session1/testsemicolon.txt

Header = TRUE

Separated by ‘;’

(ii) Plot a graph of weight against price

(iii) Create a histogram of weights

Problems

3: The file ‘ExamResults.xls’ contains Geography and History exam results for 53 pupils. The results for the two subjects are on two separate worksheets in the file.

(i) Load in the file to R using the RODBC program.

download.file(url='http://rcourse.iop.kcl.ac.uk/2011/2tue/session1/ExamResults.xls', ‘ExamResults.xls',mode='wb')

(ii) Combine the data from the two worksheets into a single data frame showing the name, age and combined exam results of each pupil.