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

review of data structures -vectors

everything is based on vectors

a vector is a series of values of the same type

we’ve seen ’character’, ’numeric’ and ’logical’ vectors

a ’factor’ is a special type of vector for categorical data

review of data structures -attributes

attributes are information specific to an object

can be anything including your own notes

```
attr(mydata,’comment’)<-’deeply dubious data’
```

attributes not called ’comment’ are shown when you print

especially important attributes

dimensions

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

row and column names

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

review of data structures - lists, data frames

a table of data that’s all one vector must be all one type

your typical excel file isn’t like that

a list is a collection of any assorted objects

a dataframe is a list of assorted vectors all the same length

the vectors (columns) must also have unique names

```
a<-data.frame(a)
```

```
a$j
```

editing or entering data in R

you can make a data frame with `data.frame()`

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

```
edit(dogs) # !
```

```
dogs<-edit(data.frame())
```

the `read.table()` function

- the commonest way to input tabular data, eg Excel
  - many 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 lables, one line header

`read.table()`, continued

- the defaults don’t only work in the simplest cases
- safer to be explicit for general use

```
cfdata<-read.table(
’g:/regresssionof.txt’,header=T,sep=’\t’,as.is=T)
str(cfdata)
cfdata<-data.frame(cfdata,strain=factor(substr(cfdata$ID,1,1)))
plot(oftime~strain,cfdata)
summary(aov(oftime~strain,cfdata)
```

keeping your data under control

- both workspace and command history can be saved
- you can restart where you left off
- this gives a false sense of security
- a much better strategy is to keep a log
  - paste commands that have done what you want into an editor
  - (optional) name it something.R
  - put comments starting with #
- such a log can be used as a script to generate your analysis again

the foreign package

- installed but not loaded by default
  - `library(foreign)`
- has methods for importing data files from most stats packages
  - SAS,SPSS,STATA, not Statistica
- also some database like files
  - .DBF (xbase, foxpro, clipper, etc)
- corresponding export methods as well

```
anx34567<-
read.dta('g:/Behaviour/Batch7/Emotionality/anx34567
nosibs.dta')
```

Microsoft Excel and Access

RODBC R Open Database Connectivity

we will talk about databases later

easy to read in whole Excel or Access tables

```
library(RODBC)

ch<-odbcConnectExcel()

sqlTables(ch)

sqlFetch(ch,’table’)
```

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)
```

```
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
review of data structures -vectors
        everything is based on vectors
        a vector is a series of values of the same type
        we’ve seen ’character’, ’numeric’ and  ’logical’ vectors
        a ’factor’ is a special type of vector for categorical data
review of data structures -attributes
        attributes are information specific to an object
        can be anything including your own notes
                attr(mydata,’comment’)<-’deeply dubious data’
        attributes not called ’comment’ are shown when you print
especially important attributes
        dimensions
                a<-1:100
                dim(a)<-c(10,10)
                a[50]
                a[10,5]
        row and column names
                colnames(a)<-letters[1:10]
                a[,’j’]
review of data structures - lists, data frames
        a table of data that’s all one vector must be all one type
        your typical excel file isn’t like that
        a list is a collection of any assorted objects
        a dataframe is a list of assorted vectors all the same length
                the vectors (columns) must also have unique names
                a<-data.frame(a)
                a$j
editing or entering data in R
        you can make a data frame with data.frame()
                dogs<-data.frame() #empty data frame
                edit(dogs)         # !

                dogs<-edit(data.frame())
the read.table() function
        the commonest way to input tabular data, eg Excel
                many 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 lables, one line header
read.table(), continued
        the defaults don’t only work in the simplest cases
        safer to be explicit for general use
        cfdata<-read.table(
        ’g:/regresssionof.txt’,header=T,sep=’\t’,as.is=T)
        str(cfdata)
        cfdata<-data.frame(cfdata,strain=factor(substr(cfdata$ID,1,1)))
        plot(oftime~strain,cfdata)
        summary(aov(oftime~strain,cfdata)
keeping your data under control
        both workspace and command history can be saved
        you can restart where you left off
        this gives a false sense of security
        a much better strategy is to keep a log
                paste commands that have done what you want into an editor
```

```
                (optional) name it something.R
                put comments starting with #
        such a log can be used as a script to generate your analysis again
the foreign package
        installed but not loaded by default
                library(foreign)
        has methods for importing data files from most stats packages
                SAS,SPSS,STATA, not Statistica
        also some database like files
                .DBF (xbase, foxpro, clipper, etc)
        corresponding export methods as well
        anx34567<- read.dta(’g:/Behaviour/Batch7/Emotionality/anx34567 nosibs.dta’)
Microsoft Excel and Access
        RODBC R Open Database Connectivity
        we will talk about databases later
        easy to read in whole Excel or Access tables
        library(RODBC)
        ch<-odbcConnectExcel()
        sqlTables(ch)
        sqlFetch(ch,’table’)
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)
                        NAs introduced by coercion
```