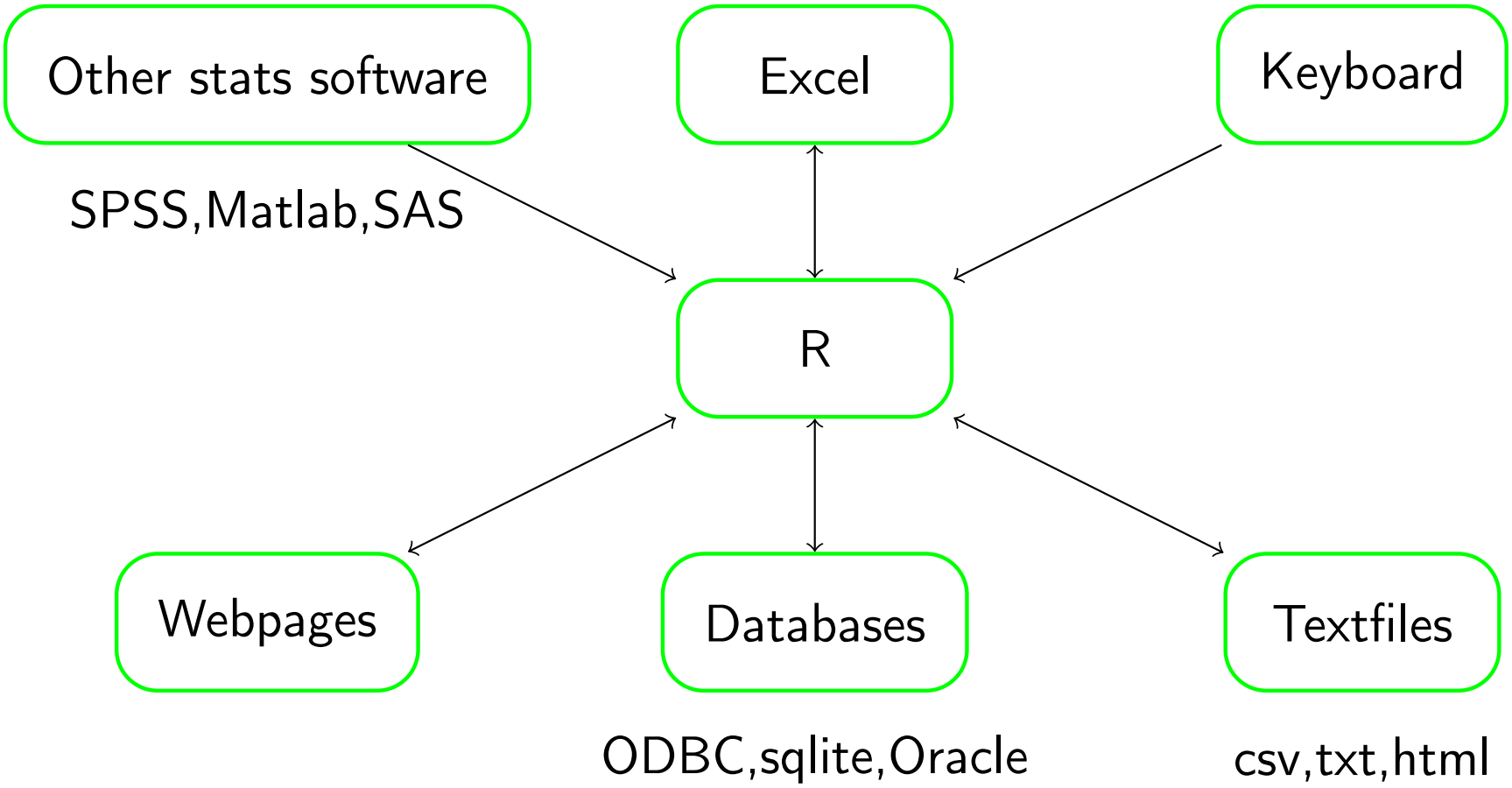
Data read into R

Bjarki Þór Elvarsson and Einar Hjörleifsson

Marine Research Institute

Where can we find data



Forming the data – rules of thumb

1. Does each variable have its own column and each subject it own line?
2. Are there any unecessary lines?
3. Do the data contain any non-US characters?
4. Are there gaps in the data?
5. Are the results entered consistently?
6. Does every variable have its own name?
7. Are the numbers correctly entered?
8. Are there any items that can cause misunderstanding?

Data entered directly into R

One quickly enter data into R:

```
weight <- c(1,5,3,2,6)
length <- c(10,17,14,12,18)
```

or if you want a more structured entry:

```
dat <- data.frame(id=numeric(0), species=character(0),
                  length=numeric(0),age=numeric(0),
                  lat = numeric(0),lon=numeric(0))
dat <- edit(dat)
```

but this only creates variable in R that, unless saved, will disappear when R is closed.

Text files - read in

- A lot of functions in R deal with reading in text files in various formats
- Most of these functions are derivatives of `read.table`, such as `read.csv` and `read.delim`

```
dat <-
  read.table(file = "nameOfFile", ## path to the file
             header = TRUE,       ## are column names
                                  ## at the top
             dec = '.',           ## decimal sign
             sep = ' ',           ## column separator symbol
             skip = 0,            ## num lines at top to skip
             stringsAsFactors = FALSE,
             comment.char = '#')  ## indicating comments
```

Other read functions

```
read.csv          ## US style CSV file
                  ## col sep ',' and dec '.'
read.csv2         ## European style CSV file
                  ## col sep ';' and dec ','
read.fwf          ## Fixed width input
                  ## (used in the olden days)
read.fortran      ## fotran formated text
readLines         ## raw lines from the file
scan              ## reads in a vector from the input
```

Data sanity check

```
head(dat)      ## shows the top 6 lines of dat
tail(dat)      ## shows the last 6 lines of dat
dim(dat)       ## shows the num row and col of dat
names(dat)     ## gives the column names of dat
summary(dat)   ## Quick summary statistics for the
               ## cols of dat
str(dat)       ## show the variable types of dat
glimpse(dat)   ## dplyr equivalent of str
```

Text files - written out

Analoguous to `read.table` we have `write.table`:

```
write.table(dat,
            file = 'nameOfFile', ## file name
            col.names = TRUE,    ## write header
            row.names = FALSE,   ## write row names
            quote = FALSE,       ## characters qouted?
            sep = ',',
            dec = '.')
```

Other write functions

```
write.csv       ## US style CSV file
                ## col sep ',' and dec '.'
write.csv2      ## European style CSV file
                ## col sep ';' and dec ','
write.fwf       ## Fixed width input
                ## from the gdata package
write           ## write raw lines to a file
```

Location of files

R is agnostic to file locations

- One can read a file in the working directory:

```
minke <- read.csv2('minke.csv')
```

Location of files

R is agnostic to file locations

- One can read a file in the working directory:

```
minke <- read.csv2('minke.csv')
```

- If minke whales are in folder called data within the working directory:

```
minke <- read.csv2('data/minke.csv')
```

Location of files

R is agnostic to file locations

- One can read a file in the working directory:

```
minke <- read.csv2('minke.csv')
```

- If minke whales are in folder called data within the working directory:

```
minke <- read.csv2('data/minke.csv')
```

- If it is somewhere on the computer one can use absolute positioning:

```
minke <- read.csv2('~/data/minke.csv') ## linux/mac
minke <- read.csv2('c:/data/minke.csv') ## windows
```

Location of files

R is agnostic to file locations

- One can read a file in the working directory:

```
minke <- read.csv2('minke.csv')
```

- If minke whales are in folder called data within the working directory:

```
minke <- read.csv2('data/minke.csv')
```

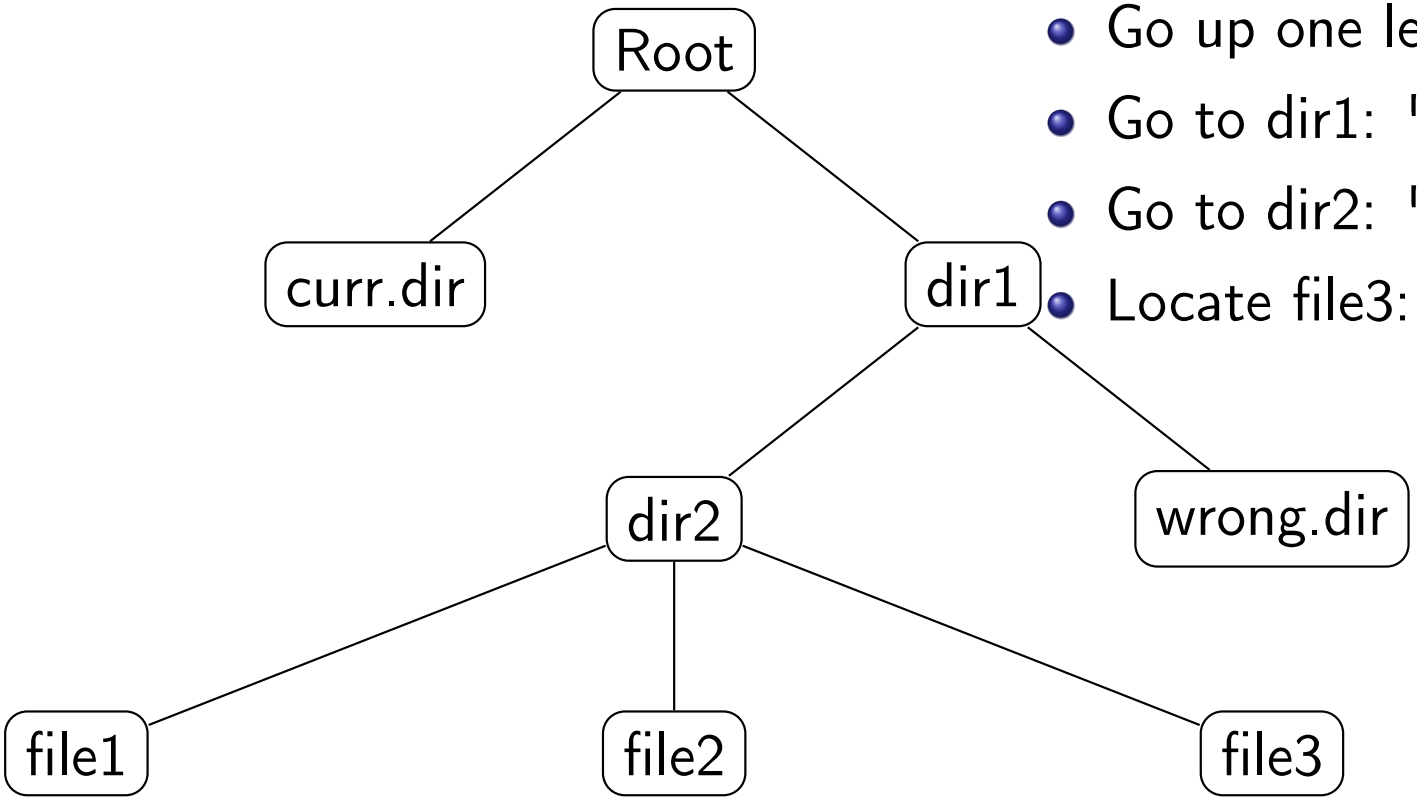
- If it is somewhere on the computer one can use absolute positioning:

```
minke <- read.csv2('~/data/minke.csv') ## linux/mac
minke <- read.csv2('c:/data/minke.csv') ## windows
```

- If it is on-line:

```
minke <-
  read.csv2('http://www.hafro.is/~bthe/data/minke.csv')
```

Location of files – relative positioning



Assume you want read "file3" from "curr.dir":

- Go up one level to "Root": ".."
- Go to dir1: "../dir1"
- Go to dir2: "../dir1/dir2"
- Locate file3: "../dir1/dir2/file3"

Excel files

The `readxl` package provides support to read in Excel files directly into R

```
# read_excel reads both xls and xlsx files
read_excel("minke.xls")
read_excel("minke.xlsx")

# Specify sheet with a number or name
read_excel("minke.xls", sheet = "data")
read_excel("minke.xls", sheet = 2)

# If NAs are represented by something other than blank cells,
# set the na argument
read_excel("minke.xls", na = "NA")

## list excel sheets
excel_sheets('minke.xls')
```

Writing to Excel–files

The openxlsx package can create Excel documents

```
## write single sheet to file
write.xlsx(minke,file='minke.xlsx',sheetName='Minke whales')

## multiple sheets
write.xlsx(list('Minke whales'=minke,'fish'=fish),
           file='minkeandfish.xlsx')
```

Databases

- Databases are commonly used to store (large amounts of) data and numerous software vendors provide database solutions, both general and specific
- Similarly numerous packages exist to interact with databases in R. Notably `DBI`, `RODBC` and `dplyr`
- Typically in an R session the user queries the database for the data needed for a particular analysis and loads it into memory. Larger datasets, that don't fit into memory will need to be subsampled

Connecting to an Access database

RODBC provides functions to connect to an Access database

```
# Load RODBC package
library(RODBC)

# Connect to Access db
db <-
  odbcConnectAccess("C:/Documents/NameOfMyAccessDatabase")

# Get data
data <- sqlQuery(db , "select *
 from Name_of_table_in_my_database")

# close connection
close(db)
```

General database connectivity

The 'dplyr' package has built in connectivity for a wide range of data base types:

- postgres
- mysql
- sqlite
- oracle (via dplyrOracle)

Interacting with databases

```
src_sqlite()    ## sets up a connection
                ## with an sqlite database
src_postgres()  ## sets up a connection with
                ## an postgres database
tbl()           ## calls a table from a database
sql()           ## prepares a sql query
copy_to()       ## saves a dataframe to a database
                ## as a table
```

Other software packages

- Package `haven` provides support for SPSS, STATA and SAS files

```
read_sas("path/to/file")  ## SAS files
read_por("path/to/file")  ## SPSS portable files
read_sav("path/to/file")  ## SPSS data files
read_dta("path/to/file")  ## Stata files
```

- Similarly `R.matlab` can read Matlab files

```
readMat('path/to/file') ## Matlab data files
```

DATRAS

- Data from European trawl surveys is available on-line from ICES
- Data is formated in the usual manner:
  - **HH**: haul/station data
  - **HL**: length data
  - **CA**: age (otolith) data
- Two R-packages connect to DATRAS, DATRAS and rICES
- For this course we will provide a script to obtain data from DATRAS

DATRAS script

```
source('datras.R')
## get station data
st <- get_datras(record = "HH", survey = 'NS-IBTS',
                 year = 2000, quarter = 1)

## get length data
le <- get_datras(record = "HL", survey = 'NS-IBTS',
                 year = 2000, quarter = 1)

## get age data
age <- get_datras(record = "CA", survey = 'NS-IBTS',
                  year = 2000, quarter = 1)
```

FishBase

The 'rfishbase' package allows access to FishBase directly from R:

```
library(rfishbase)
## query data on length weight relationship
lw <- length_weight('Gadus morhua')
## query growth parameters
vonB <- popgrowth('Gadus morhua')
## find common names (in many languages)
cod.names <- common_names('Gadus morhua')
## diet data
cod.diet <- diet('Gadus morhua')
## fecundity
cod.fec <- fecundity('Gadus morhua')
```

Class excercise

- Open notepad (or just create text file in Rstudio) and enter the following data, save it and read into R using `read.table`:

```
a   b   c
1   1.5 cod
2   2.5 haddock
```

- Save the minke whale dataset to an excel file
- Read in a datras dataset for a single year
- Save the datras dataset to a database
- Browse for your favorite species on FishBase with 'rfishbase'