Getting the data in R

Alex Douglas

a.douglas@abdn.ac.uk

@Scedacity

UNIVERSITY OF ABERDEEN

1

classes of data

- two fundamental types of data
  - numeric: numbers (integers or real numbers)
  - strings: alphanumerics, characters
- anything not a number is a string
- several types of strings
  - generic: “It takes 2 to tango”
  - factors: strings with a limited No. values - levels
  - logical: special kind of factor with 2 levels, TRUE or FALSE
  - missing data: NA

2

structures for data

- vectors
  - one dimensional collection objects
  - can contain numbers or characters or factor levels etc
  - objects in vector must be all of the same class

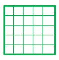
```
> numbers <- c(2,3,4,5,6)
> numbers
[1] 2 3 4 5 6
```

- matrices (arrays)
  - simply a vector that has extra dimensions
  - again, objects must be of the same class
  - arrays are multidimensional matrices

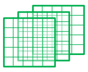
3

structures for data

matrix



array



```
> mat.1 <- matrix(1:12, nrow=4)
> mat.1
     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
```

```
> array.1 <- array(1:16, dim=c(2,4,2))
> array.1
, , 1

     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
[2,]    2    4    6    8

, , 2

     [,1] [,2] [,3] [,4]
[1,]    9   11   13   15
[2,]   10   12   14   16
```

4

structures for data

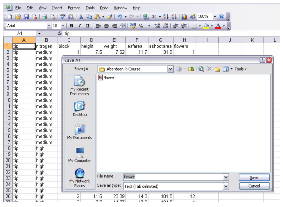
- data frames
  - most commonly used for statistical data analysis
  - powerful 2-dimensional vector holding structure
  - each column represents a variable
  - each row represents an observation
  - dataframes can hold vectors of any of the basic classes of data

```
  treat nitrogen block height weight leafarea shootarea flowers
1   tip      low     1    8.0   6.88      9.3      16.1       4
2   tip      low     1    8.0  10.23     11.9      88.1       4
3   tip      low     1    6.4   5.97      8.7       7.3       2
4   tip      low     1    7.6  13.05      7.2      47.2       8
5   tip      low     1    9.7   6.49      8.1      18.0       3
```

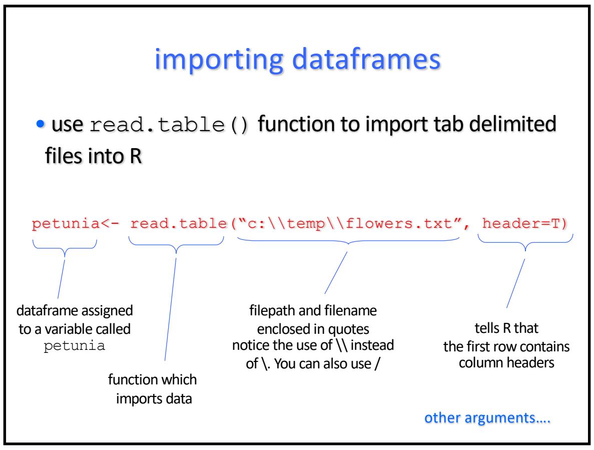
5

importing dataframes

- simplest method is to use Excel and then import data into R
- use Excel to save as a tab delimited file (*.txt)
- good practice:
  - missing data represented with NA
  - no spaces in variable names (replaced with .)
  - keep variable names short



6



importing dataframes

- use `read.table()` function to import tab delimited files into R

```
petunia<- read.table("c:\\temp\\flowers.txt", header=T)
```

dataframe assigned to a variable called `petunia`

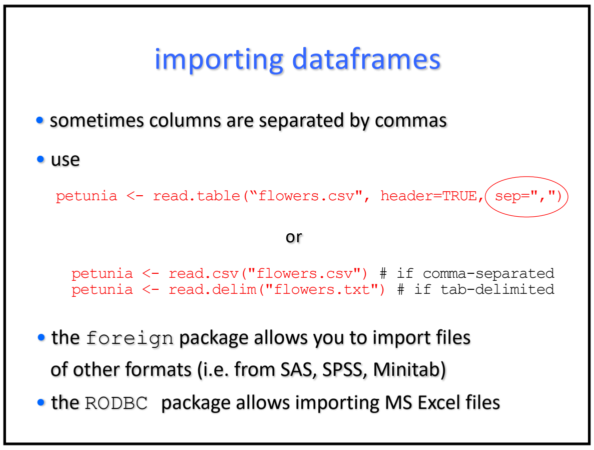
function which imports data

filepath and filename enclosed in quotes notice the use of \\ instead of \. You can also use /

tells R that the first row contains column headers

other arguments....

7



importing dataframes

- sometimes columns are separated by commas
- use

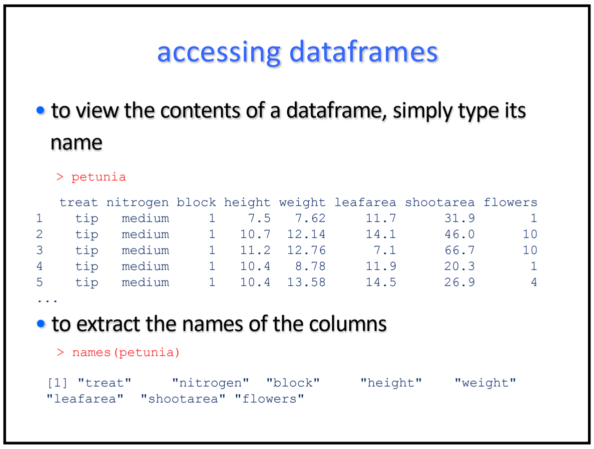
```
petunia <- read.table("flowers.csv", header=TRUE, sep=",")
```

or

```
petunia <- read.csv("flowers.csv") # if comma-separated
petunia <- read.delim("flowers.txt") # if tab-delimited
```

- the `foreign` package allows you to import files of other formats (i.e. from SAS, SPSS, Minitab)
- the `RODBC` package allows importing MS Excel files

8



accessing dataframes

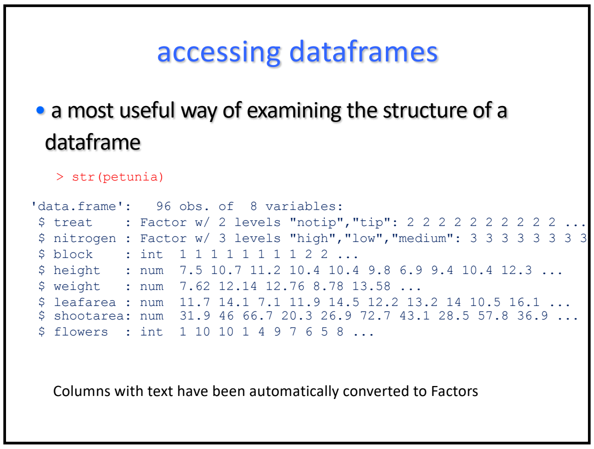
- to view the contents of a dataframe, simply type its name

```
> petunia

  treat nitrogen block height weight leafarea shootarea flowers
1   tip   medium     1    7.5   7.62     11.7      31.9       1
2   tip   medium     1   10.7  12.14     14.1      46.0      10
3   tip   medium     1   11.2  12.76      7.1      66.7      10
4   tip   medium     1   10.4   8.78     11.9      20.3       1
5   tip   medium     1   10.4  13.58     14.5      26.9       4
...
```

- to extract the names of the columns

```
> names(petunia)

[1] "treat"     "nitrogen"  "block"     "height"    "weight"
"leafarea"  "shootarea" "flowers"
```

9



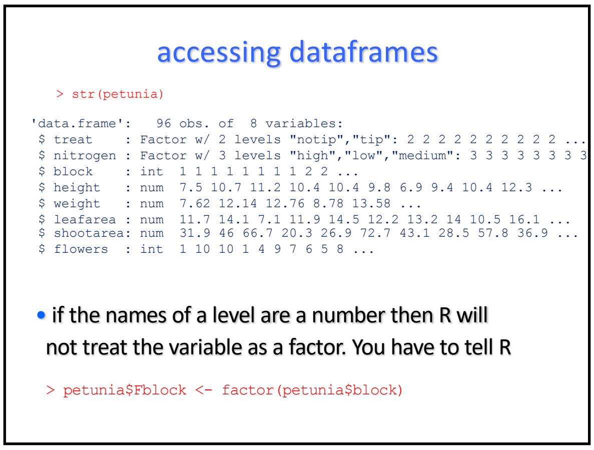
accessing dataframes

- a most useful way of examining the structure of a dataframe

```
> str(petunia)

'data.frame':   96 obs. of  8 variables:
 $ treat    : Factor w/ 2 levels "notip","tip": 2 2 2 2 2 2 2 2 2 2 ...
 $ nitrogen : Factor w/ 3 levels "high","low","medium": 3 3 3 3 3 3 3 3
 $ block    : int  1 1 1 1 1 1 1 1 2 2 ...
 $ height   : num  7.5 10.7 11.2 10.4 10.4 9.8 6.9 9.4 10.4 12.3 ...
 $ weight   : num  7.62 12.14 12.76 8.78 13.58 ...
 $ leafarea : num  11.7 14.1 7.1 11.9 14.5 12.2 13.2 14 10.5 16.1 ...
 $ shootarea: num  31.9 46 66.7 20.3 26.9 72.7 43.1 28.5 57.8 36.9 ...
 $ flowers  : int  1 10 10 1 4 9 7 6 5 8 ...
```

Columns with text have been automatically converted to Factors

10



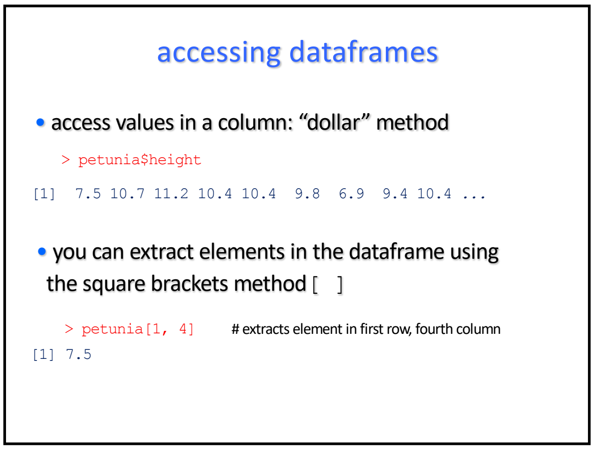
accessing dataframes

```
> str(petunia)

'data.frame':   96 obs. of  8 variables:
 $ treat    : Factor w/ 2 levels "notip","tip": 2 2 2 2 2 2 2 2 2 2 ...
 $ nitrogen : Factor w/ 3 levels "high","low","medium": 3 3 3 3 3 3 3 3
 $ block    : int  1 1 1 1 1 1 1 1 2 2 ...
 $ height   : num  7.5 10.7 11.2 10.4 10.4 9.8 6.9 9.4 10.4 12.3 ...
 $ weight   : num  7.62 12.14 12.76 8.78 13.58 ...
 $ leafarea : num  11.7 14.1 7.1 11.9 14.5 12.2 13.2 14 10.5 16.1 ...
 $ shootarea: num  31.9 46 66.7 20.3 26.9 72.7 43.1 28.5 57.8 36.9 ...
 $ flowers  : int  1 10 10 1 4 9 7 6 5 8 ...
```

- if the names of a level are a number then R will not treat the variable as a factor. You have to tell R

```
> petunia$Fblock <- factor(petunia$block)
```

11



accessing dataframes

- access values in a column: "dollar" method

```
> petunia$height

[1]  7.5 10.7 11.2 10.4 10.4  9.8  6.9  9.4 10.4 ...
```

- you can extract elements in the dataframe using the square brackets method [ ]

```
> petunia[1, 4]      # extracts element in first row, fourth column
[1] 7.5
```

12

accessing dataframes

- also extract more than one element

```
> petunia[1:3, 1:4]
  treat nitrogen block height
1   tip   medium     1    7.5
2   tip   medium     1   10.7
3   tip   medium     1   11.2
```

- all columns

notice

```
> petunia[c(1,3),]

  treat nitrogen block height weight leafarea shootarea flowers
1   tip   medium     1    7.5   7.62     11.7      31.9       1
3   tip   medium     1   11.2  12.76      7.1      66.7      10
```

13

accessing dataframes

- all rows

notice

```
> petunia[, 1:3]
  treat nitrogen block
1   tip   medium     1
2   tip   medium     1
3   tip   medium     1
4   tip   medium     1
5   tip   medium     1
6   tip   medium     1
7   tip   medium     1
8   tip   medium     1
9   tip   medium     2
...
```

14

indexing with []

- numbering method

```
> c(1, 2, 4)
[1] 1 2 4

> petunia[1:3, c(1, 2, 4)]
  treat nitrogen height
1   tip   medium    7.5
2   tip   medium   10.7
3   tip   medium   11.2
```

- naming method

```
> petunia[1:2, c("treat", "nitrogen", "height")]
  treat nitrogen height
1   tip   medium    7.5
2   tip   medium   10.7
```

15

accessing dataframes

- query using a logical test

notice

```
> petunia[petunia$height>10.5 & petunia$nitrogen=="medium",]
   treat nitrogen block height weight leafarea shootarea flowers
2    tip   medium     1   10.7  12.14     14.1      46.0      10
3    tip   medium     1   11.2  12.76      7.1      66.7      10
10   tip   medium     2   12.3  13.48     16.1      36.9       8
12   tip   medium     2   11.0  11.56     12.6      31.3       6

> petunia$height>10.5 & petunia$nitrogen=="medium"
 [1] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE
[14] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[27] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[40] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[53] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[66] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[79] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[92] FALSE FALSE FALSE FALSE FALSE
```

16

accessing dataframes

- query using a logical test

notice

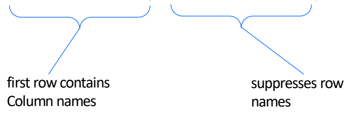
```
> petunia[petunia$height>10.5 & petunia$nitrogen=="medium",]
   treat nitrogen block height weight leafarea shootarea flowers
2    tip   medium     1   10.7  12.14     14.1      46.0      10
3    tip   medium     1   11.2  12.76      7.1      66.7      10
10   tip   medium     2   12.3  13.48     16.1      36.9       8
12   tip   medium     2   11.0  11.56     12.6      31.3       6

> petunia$height>10.5 & petunia$nitrogen=="medium"
 [1] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE
[14] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[27] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[40] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[53] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[66] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[79] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[92] FALSE FALSE FALSE FALSE FALSE
```

17

getting dataframes out

- use `write.table()` function

```
write.table(petunia, “c:\\Rdata\\petunia.txt”,
      col.names=TRUE, row.names=FALSE)
```



first row contains
Column names

suppresses row
names

- saves as tab delimited as default (other options available)

18