Data frames and tibbles: tables in R

Introduction to R - Day 1

Instructor: Selina Baldauf

Freie Universität Berlin - Theoretical Ecology

Data frames

The built-in data structure for tables in R is a `data frame`.

- vectors in R can't represent a table with data that is connected via rows

  > Data frames are one of the **biggest and most important ideas** in R, and one of the things that make R different from other programming languages.
  > (Wickham, Advanced R)

cities	population	area_km2
Istanbul	15100000	2576
Moscow	12500000	2561
London	9000000	1572
Saint Petersburg	5400000	1439
Berlin	3800000	891
Madrid	3200000	604
Kyiv	3000000	839
Rome	2800000	1285
Bucharest	2200000	228
Paris	2100000	105

Data frames

A data frame is a **named list of vectors** of the same length.

Basic properties of a data frame

- every column is a vector
- columns have a header
  - this is the name of the vector in the list
- within one column, all values are of the same data type
- every column has the same length

character		numeric
cities	population	area_km2
Istanbul	15100000	2576
Moscow	12500000	2561
London	9000000	1572
Saint Petersburg	5400000	1439
Berlin	3800000	891
Madrid	3200000	604
Kyiv	3000000	839
Rome	2800000	1285
Bucharest	2200000	228
Paris	2100000	105

Data frames

Data frames are created with the function `data.frame()`:

```r
cities <- c(
  "Istanbul", "Moscow", "London",
  "Saint Petersburg", "Berlin","Madrid",
  "Kyiv", "Rome", "Bucharest","Paris")

population <- c(
  15.1e6, 12.5e6, 9e6, 5.4e6, 3.8e6,
  3.2e6, 3e6, 2.8e6, 2.2e6, 2.1e6)

area_km2 <- c(2576, 2561, 1572, 1439,
  891, 604, 839, 1285, 228, 105)
```

```
##               cities population area_km2
## 1           Istanbul   15100000     2576
## 2             Moscow   12500000     2561
## 3             London    9000000     1572
## 4   Saint Petersburg    5400000     1439
## 5             Berlin    3800000      891
## 6             Madrid    3200000      604
## 7               Kyiv    3000000      839
## 8               Rome    2800000     1285
## 9          Bucharest    2200000      228
## 10             Paris    2100000      105
```

```r
data.frame(
  cities = cities,
  population = population,
  area_km2 = area_km2
  )
```
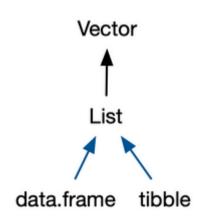
Tibbles

Tibbles are

> a **modern reimagining of the data frame**. Tibbles are designed to be (as much as possible) **drop-in replacements** for data frames.
>
> (Wickham, Advanced R)

Have a look at this book chapter for a full list of the differences between data frames and tibbles and the advantages of using tibbles.

- tibbles have the same basic properties as data frames (named list of vectors)

- everything that you can do with data frames, you can do with tibbles

Tibbles

Tibbles are a available from the `tibble` package.

Before we use tibbles, we need to install the package once using the function `install.packages`:

```
# This has do be once
install.packages("tibble")
```

Then, we need to load and attach the package to our current R session using `library`:

```
# This has to be done every time R restarts
# Put it at the beginning of a script
library(tibble)
```

Tibbles

Create a tibble using the `tibble()` function:

```r
library(tibble)

tibble(
  cities = cities,
  population = population,
  area_km2 = area_km2
)
```

```
## # A tibble: 10 x 3
##    cities          population area_km2
##    <chr>                <dbl>    <dbl>
##  1 Istanbul          15100000     2576
##  2 Moscow            12500000     2561
##  3 London             9000000     1572
##  4 Saint Petersburg   5400000     1439
##  5 Berlin             3800000      891
##  6 Madrid             3200000      604
##  7 Kyiv               3000000      839
##  8 Rome               2800000     1285
##  9 Bucharest          2200000      228
## 10 Paris              2100000      105
```

Note: If you want to use a function from a package you can attach it using `library(package)` or you can use `package::function` to tell R where a function is from (e.g. `tibble::tibble()`). I will sometimes do this to clearly distinguish between base R and package functions.

Exploring tibbles

Look at the structure of an object using `str()`:

```
str(cities_tbl)
```

```
## tibble [10 x 3] (S3: tbl_df/tbl/data.frame)
##  $ cities    : chr [1:10] "Istanbul" "Moscow" "London" "Saint Petersburg" ...
##  $ population: num [1:10] 15100000 12500000 9000000 5400000 3800000 3200000 3000000 2800000
2200000 2100000
##  $ area_km2  : num [1:10] 2576 2561 1572 1439 891 ...
```

- This function shows you:

    - data type of object (`tbl_df/tbl/data.frame`)
    - extent of the data (10 rows times 3 columns)
    - column names and data types

- This function works for every R object and is very useful if code doesn't work and you don't know why

Exploring tibbles

How many rows?

```
nrow(cities_tbl)
```

```
## [1] 10
```

How many columns?

```
ncol(cities_tbl)
```

```
## [1] 3
```

What are the column headers?

```
names(cities_tbl)
```

```
## [1] "cities"     "population" "area_km2"
```

Exploring tibbles

Look at the entire table in a separate window with `view()`:

```
tibble::view(cities_tbl)
```

Get a quick summary of all columns:

```
summary(cities_tbl)
```

```
##      cities            population            area_km2
##   Length:10         Min.   : 2100000    Min.    : 105.0
##   Class :character  1st Qu.: 2850000    1st Qu.: 662.8
##   Mode  :character  Median : 3500000    Median :1088.0
##                     Mean   : 5910000    Mean    :1210.0
##                     3rd Qu.: 8100000    3rd Qu.:1538.8
##                     Max.   :15100000    Max.    :2576.0
```

Now you

Task 3: Tibbles

Find the task description here

Summary I

data frames and tibbles

- can be used to represent tables in R
- are pretty similar, however tibbles are slightly conventient and modern
- are **named lists of vectors of the same length**
  - every column is a vector
  - columns have a header which is the name of the vector in the list
  - within one column, values are of same data type
  - every column has the same length

tibbles

- to use tibbles, install the package once with `install.packages("tibble")`
- put `library(tibble)` at the beginning of your script to load package

Summary II

Creating tibbles and data frames

```r
# data frame
data.frame(
  a = 1:3,
  b = c("a", "b", "c"),
  c = c(TRUE, FALSE, FALSE)
)
# tibble
tibble(
  a = 1:3,
  b = c("a", "b", "c"),
  c = c(TRUE, FALSE, FALSE)
)
# convert data frame to tibble
as_tibble(df)
```

Summary III

Looking at tibble structure

```r
# structure of tibble and data types of columns
str(tbl)
# number of rows
nrow(tbl)
# number of columns
ncol(tbl)
# column headers
names(tbl)
# look at the data in a new window
tibble::view(tbl)
# summary of values from each column
summary(tbl)
```