Goals for today

By the end of today you will be able to load data into R, inspect it, subset data using arbitrary criteria, write functions to manipulate data.

Format: live demo instruction, followed by exercises you will do yourself

We will largely follow Software Carpentry's "R for Reproducible Scientific Analysis" lesson

Caveats

Impossible to learn R in 4 hrs. We will be skipping a LOT.

We will focus on **fundamentals** and getting comfortable with R so that you can learn on your own later

What is R?

A programming language

A statistical computing environment consisting of a command line and an interpreter

Free and open source

Widely used across academia and industry

R or Python?

Both are interpreted programming languages commonly used in biology

R for data analysis and statistics, data visualization

Python for general-purpose, performance, scaling to large projects, more software engineering-heavy tasks (e.g., serving thousands of requests a minute in a web server)

Programming concepts are the same between R and Python; syntax is slightly different

Use whatever language has the best tools for the job

Exercises

Which of the following are valid variable names?

```
min_height
max.height
_age
.mass
MaxLength
min-length
2widths
celsius2kelvin
```

What is the value of each at the end?

```
mass <- 50
age <- 122
mass <- mass * 3
age <- age - 20
```

How to determine if `mass` is greater than `age`?

Install the following packages:

`ggplot2`

`dplyr`

`gapminder`

Download data

`https://helix.nih.gov/~dalerr/gapminder.csv`

Create a new "`data`" directory in your current project and save the file there.

Make sure it's called "`gapminder.csv`"

We will load and inspect it later.

Know what you're working with

Using your command line knowledge:

1. What is the size of the file?
2. How many rows of data does it contain?
3. What kinds of values are stored in this file?

Getting help

Read the help for the `c()` function. What kind of vector do you expect from each of the following:

```
c(1, 2, 3)
c('d', 'e', 'f')
c(1, 2, 'f')
```

Getting help (2)

Look up the help for the `paste()` function, which we will use later.

What is the difference between the "`sep`" and "`collapse`" arguments?

Toy data set

Create a file called `data/feline-data.csv` with the following contents:

```
coat,weight,likes_string
calico,2.1,1
black,5.0,0
tabby,3.2,1
```

Named vector

Make a vector with the numbers 1 through 26

Multiply it by 2

Give the resulting vector the names A through Z.

(hint: there is a built-in vector called LETTERS)

Looking for factors

Is there a factor in our `cats` data.frame? Where is it?

Use `?read.csv` to find a way to keep text columns as character vectors instead of factors

Try it out and verify it works.

Extracting

```
cats[1]

cats[[1]]

cats$coat

cats["coat"]

cats[1, 1]

cats[, 1]

cats[1, ]
```

Making dataframes

```
df <- data.frame(id = c('a', 'b', 'c'),
                 x = 1:3,
                 y = c(TRUE, TRUE, FALSE),
                 stringsAsFactors = FALSE)
```

Make a data frame holding the following information about yourself:

- First name
- Last name
- Lucky number

Then use rbind() to add an entry for the people sitting next to you. Then use cbind() to add a column with each person's answer to: "is it time for coffee?"

A script to load

Write an R script to load in the gapminder dataset we downloaded previously.

Put the script in a `scripts/` directory

Run the script using the `source()` function, using the path to the file as the argument

Inspection

Read the output of `str(gapminder)`. Use what you've learned about factors, lists, vectors, and the output of `colnames()` and `dim()` to explain everything that `str()` prints out.

Discuss with your neighbors if there are parts you can't interpret.

Subsetting

Given:

```
x <- c(5.4, 6.2, 7.1, 4.8, 7.5)
names(x) <- c('a', 'b', 'c', 'd', 'e')
print(x)
```

```
  a   b   c   d   e
5.4 6.2 7.1 4.8 7.5
```

Come up with 3 different commands that produce the following output:

```
  b   c   d
6.2 7.1 4.8
```

Subsetting 2

Given:

```
x <- c(5.4, 6.2, 7.1, 4.8, 7.5)
names(x) <- c('a', 'b', 'c', 'd', 'e')
print(x)
```

```
  a   b   c   d   e
5.4 6.2 7.1 4.8 7.5
```

Write a subsetting command to return the values in x that are greater than 4 and less than 7

Subsetting 3

```
seAsia <- c("Myanmar","Thailand","Cambodia","Vietnam","Laos")
gapminder <- read.csv("data/gapminder.csv", header=TRUE)
countries <- unique(as.character(gapminder$country))
```

How to get a logical vector that is TRUE for each row in `gapminder` for all countries in SE Asia, and FALSE otherwise?

Subsetting 4

Given:

```
xlist <- list(a = "immunology", b = 1:10, data = head(iris))
```

Use your knowledge of list and vector subsetting to extract the number 2 from `xlist`.

(Hint: it's contained within the "b" item in the list)

Exploring unknown objects

Run a linear model:

```
mod <- aov(pop ~ lifeExp, data=gapminder)
```

Extract the Y intercept of the linear model.

(Hint: `attributes()` will be helpful; `str()` will print a lot of output but could also be helpful)

Find the errors

Extract observations from 1957

```
gapminder[gapminder$year = 1957,]
```

Extract all columns except 1 through 4

```
gapminder[,-1:4]
```

Extract rows where life exp is greater than 80 years

```
gapminder[gapminder$lifeExp > 80]
```

Extract the first row and fourth and fifth columns

```
gapminder[1, 4, 5]
```

Extract rows from 2002 and 2007

```
gapminder[gapminder$year == 2002 | 2007,]
```

Write a function

Write a function called `kelvin_to_celsius` that takes a temperature in Kelvin and returns the temperature in Celsius (i.e. subtract 273.15)

Write a function 2

Write a function called `fence` that takes two vectors as arguments called "text" and "wrapper", and prints out the "text" wrapped with "wrapper". Example input and output:

```
fence("hooray for T cells", "***")
[1] "*** hooray for T cells ***"
```

```
fence("we're almost done", "!")
[1] "! we're almost done !"
```

(Hint: the `paste()` function will be helpful for this)