00 Before we get started

http://www.datacarpentry.org/R-ecology-lesson/00-before-we-start.html

0. Learning Objectives:
    - Organize files and directories related to a particular set of analyses in an R Project within RStudio
    - Define the following (as they apply to R): Script, function, working directory, assign, object, variable
    - Describe the purpose of the RStudio script, console, environment, and plot windows
    - Assign values to variables -Call functions with zero or more named or unnamed arguments
    - Use the built-in RStudio help interface to search for more information on R functions -Ask for help from the R user community, providing sufficient information for the problem to be reproduced and troubleshooted
1. Working Directory
2. RStudio to manage WD
    - Open RStudio
    - Create Project data-carpentry
    - Create data/ and download data into it
    - Create data-carpentry-script.R
    - Show diagram
3. RStudio walk-through
    - Panes
    - Auto-complete
    - R Code
        - Code is instructions to computer: 2-ways
        - Console
            - prompt (> vs +)
            - ESC
        - Script to console (buttons and ctrl-enter)
4. Basics of R
    - Free and Open Source
    - Better than commercial
        - First implementations
        - widely extensible (nrow(available.packages()))
    - 1000's of developers vs 10's-100's
    - huge and (mostly) friendly community
    - functional and OO
    - not just for stats, general purpose programming too.
5. R Syntax
    - http://www.datacarpentry.org/R-ecology-lesson/00-before-we-start.html#the-r-syntax
    - comments
    - assignment operator
    - a function
    - = for args
    - $
    - quotes v not quotes
6. Using functions

- round
- args(round)
- show multiple args
  - named (orderd, not ordered)
  - not-named

7. The most important skill in coding: HELP!
   - https://twitter.com/ThePracticalDev/status/716390583217029124
   - When you know the function
     - ? or help
     - args
     - RStudio auto-complete
   - When you know the package
     - help(package="dplyr")
   - When you know a bit of the topic
     - ??smirnov
     - ??histogram
   - Google and Stackoverflow
     - Asking good questions (see http://www.datacarpentry.org/R-ecology-lesson/00-before-we-start.html#asking-for-help)
   - Mailing Lists
   - Task Views

01 Intro to R

http://www.datacarpentry.org/R-ecology-lesson/01-intro-to-R.html

0. Learning objectives:
   - Familiarize participants with R syntax
   - Understand the concepts of objects and assignment
   - Understand the concepts of vector and data types
   - Get exposed to a few functions
1. Creating Objects
   - Basic operators +, -, *, /
   - can save result to an object with <-
   - weight_kg <- 55
   - (weight_kg <- 55)
   - weight_kg <- 55; weight_kg
   - overwrite value
   - use math to change (*2.2 for lbs)
   - weight_lbs <- 2.2 * weight_kg
2. Challenge
   - work with person next to you
   - discuss what you think values would be
   - confirm using R
   - About 5 minutes
3. Vectors and data types
   - Vector

      - most basic structure
      - have length
      - have type (must be the same)
   - weight_g <- c(50, 60, 65, 82)
   - weight_g
   - animals <- c(“mouse”, “rat”, “dog”)
   - animals
   - length(), class(), str()
   - add to the vector (beginning and end)
      - c(animals,“”)
   - types
      - numeric
      - character
      - logical/boolean
      - integer
      - raw
      - complex
   - Other data structures
      - list
      - matrix
      - factor
      - data.frame

4. Challenge
   - Work on together
   - We’ve seen that atomic vectors can be of type character, numeric, integer, and logical. But what happens if we try to mix these types in a single vector?
      - coercion
   - What class will these be?
   - num_char <- c(1, 2, 3, ‘a’)
   - num_logical <- c(1, 2, 3, TRUE)
   - char_logical <- c(‘a’, ‘b’, ‘c’, TRUE)
   - tricky <- c(1, 2, 3, ‘4’)
   - c(3,“z”,TRUE)
   - so:
   - Logical > Numeric
   - Logical > Character
   - Character > Numeric
   - Character > Numeric + Logical
5. Subsetting
   - animals <- c(“mouse”, “rat”, “dog”, “cat”)
   - animals[2]
   - animals[c(3,2)]
   - animals[c(1, 2, 3, 2, 1, 4)]
   - animals[-3]
6. Conditional subsetting
   - weight_g <- c(21, 34, 39, 54, 55)
   - Subset based on logical
      - weight_g[c(TRUE, FALSE, TRUE, TRUE, FALSE)]
   - so we can use conditional statements such as

        - weight_g > 50
        - weight_g[weight_g > 50]
        - or: weight_g[weight_g < 30 | weight_g > 50]
        - and: weight_g[weight_g >= 30 & weight_g == 21]
    - Works with both numeric and character
        - animals[animals == “cat” | animals == “rat”]
        - %in%: animals %in% c(“rat”, “cat”, “dog”, “duck”)
        - animals[animals %in% c(“rat”, “cat”, “dog”, “duck”)]

7. Challenge:
    - what does “four” > “five” return?
    - why?

8. Missing Data
    - Should have already discussed during spreadsheet, if not say a few words
    - R’s values is NA
    - for example:
        - planets <- c(“Mercury”, “Venus”, “Earth”, “Mars”, “Jupiter”, “Saturn”, “Uranus”, “Neptune”, NA)
        - heights <- c(2, 4, 4, NA, 6)
        - mean(heights)
        - max(heights)
        - mean(heights, na.rm = TRUE)
        - max(heights, na.rm = TRUE)
    - is.na()
    - na.omit()
    - complete.cases()

9. Challenge
    - Try this: sample <- c(2, 4, 4, “NA”, 6); mean(sample, na.rm = TRUE)
    - What happens and why?
    - Why does the error message say the argument is not numeric?

02 Starting with data

http://www.datacarpentry.org/R-ecology-lesson/02-starting-with-data.html

0. Learning objectives:
    - load external data (CSV files) in memory using the survey table (surveys.csv) as an example
    - explore the structure and the content of a data frame in R
    - understand what factors are and how to manipulate them

1. Introduce the survey.csv dataset
    - it is data on species captured across many years and plots

Column	Description
record_id	Unique id for the observation
month	month of observation
day	day of observation
year	year of observation

Column	Description
plot_id	ID of a particular plot
species_id	2-letter code
sex	sex of animal (“M”, “F”)
hindfoot_length	length of the hindfoot in mm
weight	weight of the animal in grams
genus	genus of animal
species	species of animal
taxa	e.g. Rodent, Reptile, Bird, Rabbit
plot_type	type of plot

-

2. Download the data, store in data folder, and read in

   - download.file(“https://ndownloader.figshare.com/files/2292169”, “data/portal_data_joined.csv”)
   - surveys <- read.csv(‘data/portal_data_joined.csv’)
   - head(surveys): describe the result
   - str(surveys): describe the result

3. Challenge:

   - If not already done, download data
   - read into surveys
   - get result of str(): green sticky when you get there
   - Answer these:
     - What is the class of the object surveys?
     - How many rows and how many columns are in this object?
     - How many species have been recorded during these surveys?
   - Point out factors. Useful structure, but unique and need to be discussed before we dig more into data frames.

4.