SRC Summer Camp

Session 1

Introduction to R

This is an R markdown file summarizing the material from lecture 2. You can also use it as a template for writing your Problem Sets in R markdown.

First, when writing an R script you can give it a title, date, and name who the author is.

```
#---
# title: Lecture 2: Introduction to R
# author: Name
# date: Sep 16, 2019
#---
```

Arithmetic Operations

You can use R as a calculator. You can try summation, division, multiplication, power calculations.

```
5 + 3
```

```
## [1] 8
```

```
5 - 3
```

```
## [1] 2
```

```
5 / 3
```

```
## [1] 1.666667
```

```
5 * 3
```

```
## [1] 15
```

```
5 ^ 3
```

```
## [1] 125
```

Objects

R is an "object-oriented" programming language. An *object* is any piece of information stored by R. These can be anything, for example:

- A dataset (e.g. a country year dataset)
- A subset of a dataset (e.g. just the democracies in a country year dataset)
- A number (e.g. $2\pi + 1$)
- A phrase (e.g. "stanford is awesome")
- A function (e.g. a function that takes in x and gives you $x^2 + 8$)

R can store *objects* with a name of our choice. Use <- as an assignment operator for objects.

```
result <- 5 + 3

result
```

```
## [1] 8
```

If we assign a new value to the same object name, then we will overwite this object (so be careful when doing so!)

```
result <- 5 - 3
result
```

```
## [1] 2
```

R can also represent other types of values as objects, such as strings of characters:

```
s <- "stanford is awesome"
s
```

```
## [1] "stanford is awesome"
```

There are many other classes of data besides numeric and character, which we will talk about in class.

Vectors

A *vector* simply represents a collection of information stored in a specific order. We use the function c(), which stands for "concatenate," to enter a data vector (with commas separating elements of the vector):

```
world.pop <- c(2525779, 3026003, 3691173, 4449049, 5320817, 6127700, 6916183)
world.pop
```

```
## [1] 2525779 3026003 3691173 4449049 5320817 6127700 6916183
```

```
a <- c(1, 2, 3, 5)
```

To access specific elements of a vector, we use square brackets [ ]. This is called *indexing*:

```
world.pop[2]
```

```
## [1] 3026003
```

```
world.pop[-1]
```

```
## [1] 3026003 3691173 4449049 5320817 6127700 6916183
```

```
world.pop[c(2, 4)]
```

```
## [1] 3026003 4449049
```

```
world.pop[-c(1,3)]
```

```
## [1] 3026003 4449049 5320817 6127700 6916183
```

Since each element of this vector is a numeric value, we can apply arithmetic operations to it:

```
b <- world.pop /1000
```

Functions

A *function* takes input object(s) and returns an output object. In R, a function generally runs as funcname(input) where funcname is the function name and input is the input object. We often call these inputs *arguments*.

Some basic functions useful for summarizing data include:

- length(): length of a vector (number of elements)
- min(): minimum value
- max(): maximum value
- range(): range of data
- mean(): mean
- sum(): sum

```
length(world.pop)
```

```
## [1] 7
```

```
min(world.pop)
```

```
## [1] 2525779
```

```
max(world.pop)
```

```
## [1] 6916183
```

```
range(world.pop)[2]
```

```
## [1] 6916183
```

```
mean(world.pop)
```

```
## [1] 4579529
```

```
sum(world.pop)
```

```
## [1] 32056704
```

Data Files

Most of the time, we will load data from an external file. For this class, we will mostly use:

- *CSV*: comma-separated files. These are conceptually similar to Microsoft Excel or Google Spreadsheet.
- *RData*: collection of R objects including data sets.

Working Directory

The *working directory* is where R will by default load data from and save data to. You can use the function getwd() to display the current working directory.

```
getwd()
```

```
## [1] "/Users/xyq/Library/CloudStorage/Dropbox/Teaching/_Undergraduate/R_sessions/session1"
```

You can use the function setwd() to change the working directory

```
# setwd("") # Enter your directory here
```

In RStudio, you can also go to Session, Set Working Directory, To Source File Location to set it where your R file is located.

Clear memory

You can use the `rm` function to clear memory or remove objects.

```
rm(list = ls())
```

Reading in Files

- For *CSV* files:

```
d <- read.csv("nations.csv")
```

- For *RData* files:

```
rm(list = ls())
load("nations.RData")
```

Data Frames

A *data frame* is a collection of vectors, but we can think of it like a spreadsheet.

Useful functions for data frames include: - names(): return a vector of variable names - nrow(): return the number of rows - ncol(): return the number of columns - dim(): combine ncol() and nrow() into a vector - summary(): produce a summary

```
names(d)
```

```
## [1] "country" "isocode" "year"    "demo"    "gdppc"
```

```
nrow(d)
```

```
## [1] 192
```

```
ncol(d)
```

```
## [1] 5
```

```
dim(d)
```

```
## [1] 192   5
```

```
summary(d)
```

```
##    country            isocode               year           demo
##  Length:192         Length:192         Min.   :1960   Min.   :0.0000
##  Class :character   Class :character   1st Qu.:1960   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median :1984   Median :1.0000
##                                        Mean   :1984   Mean   :0.5208
##                                        3rd Qu.:2009   3rd Qu.:1.0000
##                                        Max.   :2009   Max.   :1.0000
##      gdppc
##  Min.   :   70.0
##  1st Qu.:  588.4
##  Median : 1920.2
##  Mean   : 7321.1
##  3rd Qu.: 7126.2
##  Max.   :65088.4
```

The $ operator is one way to access variables from a data frame:

```
table(d$demo)
```

```
##
##   0   1
##  92 100
```

```
table(d$year)
```

```
##
## 1960 2009
##   96   96
```

Another way of retrieving variables is to use brackets [ ] with a comma in the form [rows, columns]:

```
d[, "country"]
```

```
##   [1] "Afghanistan"          "Afghanistan"          "Albania"
##   [4] "Albania"              "Argentina"            "Argentina"
##   [7] "Australia"            "Australia"            "Austria"
##  [10] "Austria"              "Belgium"              "Belgium"
##  [13] "Benin"                "Benin"                "Bolivia"
##  [16] "Bolivia"              "Brazil"               "Brazil"
##  [19] "Burkina Faso"         "Burkina Faso"         "Cambodia"
##  [22] "Cambodia"             "Cameroon"             "Cameroon"
##  [25] "Canada"               "Canada"               "Central African Rep."
##  [28] "Central African Rep." "Chad"                 "Chad"
##  [31] "Chile"                "Chile"                "China"
##  [34] "China"                "Colombia"             "Colombia"
##  [37] "Congo, Dem Rep"       "Congo, Dem Rep"       "Congo, Rep."
##  [40] "Congo, Rep."          "Costa Rica"           "Costa Rica"
##  [43] "Cote d'Ivoire"        "Cote d'Ivoire"        "Cuba"
##  [46] "Cuba"                 "Cyprus"               "Cyprus"
##  [49] "Denmark"              "Denmark"              "Dominican Rep."
##  [52] "Dominican Rep."       "Ecuador"              "Ecuador"
##  [55] "Egypt"                "Egypt"                "El Salvador"
##  [58] "El Salvador"          "Finland"              "Finland"
##  [61] "France"               "France"               "Gabon"
##  [64] "Gabon"                "Ghana"                "Ghana"
##  [67] "Greece"               "Greece"               "Guatemala"
##  [70] "Guatemala"            "Guinea"               "Guinea"
##  [73] "Haiti"                "Haiti"                "Honduras"
##  [76] "Honduras"             "Hungary"              "Hungary"
##  [79] "India"                "India"                "Indonesia"
##  [82] "Indonesia"            "Iran"                 "Iran"
##  [85] "Iraq"                 "Iraq"                 "Ireland"
##  [88] "Ireland"              "Israel"               "Israel"
##  [91] "Italy"                "Italy"                "Jamaica"
##  [94] "Jamaica"              "Japan"                "Japan"
##  [97] "Jordan"               "Jordan"               "Korea, South"
## [100] "Korea, South"         "Laos"                 "Laos"
## [103] "Lebanon"              "Lebanon"              "Liberia"
## [106] "Liberia"              "Libya"                "Libya"
## [109] "Madagascar"           "Madagascar"           "Malaysia"
## [112] "Malaysia"             "Mali"                 "Mali"
## [115] "Mauritania"           "Mauritania"           "Mexico"
## [118] "Mexico"               "Mongolia"             "Mongolia"
## [121] "Morocco"              "Morocco"              "Myanmar"
## [124] "Myanmar"              "Nepal"                "Nepal"
## [127] "Netherlands"          "Netherlands"          "New Zealand"
## [130] "New Zealand"          "Nicaragua"            "Nicaragua"
## [133] "Niger"                "Niger"                "Nigeria"
## [136] "Nigeria"              "Norway"               "Norway"
## [139] "Oman"                 "Oman"                 "Panama"
```

```
## [142] "Panama"              "Paraguay"            "Paraguay"
## [145] "Peru"                "Peru"                "Philippines"
## [148] "Philippines"         "Poland"              "Poland"
## [151] "Portugal"            "Portugal"            "Romania"
## [154] "Romania"             "Saudi Arabia"        "Saudi Arabia"
## [157] "Senegal"             "Senegal"             "Singapore"
## [160] "Singapore"           "Somalia"             "Somalia"
## [163] "South Africa"        "South Africa"        "Spain"
## [166] "Spain"               "Sri Lanka"           "Sri Lanka"
## [169] "Sudan"               "Sudan"               "Sweden"
## [172] "Sweden"              "Switzerland"         "Switzerland"
## [175] "Syria"               "Syria"               "Thailand"
## [178] "Thailand"            "Togo"                "Togo"
## [181] "Tunisia"             "Tunisia"             "Turkey"
## [184] "Turkey"              "United Kingdom"      "United Kingdom"
## [187] "United States"       "United States"       "Uruguay"
## [190] "Uruguay"             "Venezuela"           "Venezuela"
```

```
d[1:4, "year"]
```

```
## [1] 1960 2009 1960 2009
```

Saving Objects

When you quit RStudio, you will be asked whether you would like to save the workspace. You should answer *no* to this in general: we only want to save what we want!

- To export *CSV*:

```
write.csv(d, file = "nations_new.csv")
```

- To export *RData*:

```
save(d, file = "d.RData")
```

Packages

One of R's strengths is the existence of a large community of R users who contribute various functionalities as R packages. For example, the foreign package is useful when dealing with files from other statistical software.

```
# install.packages("foreign") # install package
library(foreign) # load package
```

Programming Tips

First, use the text editor in RStudio to write your code rather than directly typing it into the R console. That way you can save a record of your program as a R file.

Second, we can annotate our R code so that it is easily understandable to ourselves and others using #

```
# File: d.R
# Author: Charlie McClean
# This code loads the UN population data

d <- read.csv("nations.csv")
```

Third, we should follow a certain set of coding rules:

- Use informative names for files, variables, and functions

- Use systematic spacing and indentation

```
# File: d.R
# Author: Charlie McClean
# This code loads the UN population data

d <- read.csv("nations.csv")
```