Lab 01 - Intro to R and RStudio

Dante Yasui

2024/04/03

Welcome to Lab!

Goals for this week:

- Install R and RStudio
- Learn about R basics and vectors
- Install the tidyverse package
- Learn how to use and complete Koans

- **Complete and submit `K01_vector.R`**
  - not due until next Friday

Setting up our R environment

Step 1. Download and install R

follow instructions from: https://cran.r-project.org/

- choose latest release for your OS
- Mac users:
  - know if your device has Apple silicon (M1-3 macs) or Intel cpu
  - also install xquartz: https://www.xquartz.org/

Step 2. Download and install RStudio

Go here: https://posit.co/download/rstudio-desktop/

- click on the link under **2. Install RStudio**
- follow the installer instructions

Open RStudio

You should see the default panes layout

- find the `Console` in the bottom left pane
- you can use this to run R code

```r
1 + 2
```

R as a fancy calculator

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

```
sqrt(64)
```

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

Everything is an object to R Objects have a name and value(s)

```
# we can define variables with either `<-` or `=`
a <- 9
b = 3
a / b
```

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

This is useful to for all sorts of reasons

- labeling things with memorable names
- having functions call on values which might change
- using shorter names for long values

e.g.,

```
pi <- 3.141593
```

```

name     = 'dante'
age      = 25
hometown = "Davis, CA"
```

```

typeof(name)
```

There are different *types* of objects

```
## [1] "character"
```

```
typeof(age)
```

```
## [1] "double"
```

```
typeof(hometown)
```

```
## [1] "character"
```

Vectors are lists To `combine` things into a vector, use the `c()` syntax:

```
c(1,2,3)
```

```
## [1] 1 2 3
```

Vectors are also objects:

```
stooges <- c("larry", "moe", "curly")
stooges
```

```
## [1] "larry" "moe"   "curly"
```

```

secret_message = "hello world!"
print(secret_message)
```

Functions do things with objects

```
## [1] "hello world!"
```

You can also define your own functions:

```
my_addition <- function(a, b) {
  return (a + b - 1)
}
```

R is really good at doing things with vector objects

```
long_vector <- 1:100
long_vector
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100
```

```
long_vector ^ 2
```

```
##   [1]     1     4     9    16    25    36    49    64    81   100   121   144
##  [13]   169   196   225   256   289   324   361   400   441   484   529   576
##  [25]   625   676   729   784   841   900   961  1024  1089  1156  1225  1296
##  [37]  1369  1444  1521  1600  1681  1764  1849  1936  2025  2116  2209  2304
##  [49]  2401  2500  2601  2704  2809  2916  3025  3136  3249  3364  3481  3600
##  [61]  3721  3844  3969  4096  4225  4356  4489  4624  4761  4900  5041  5184
##  [73]  5329  5476  5625  5776  5929  6084  6241  6400  6561  6724  6889  7056
##  [85]  7225  7396  7569  7744  7921  8100  8281  8464  8649  8836  9025  9216
##  [97]  9409  9604  9801 10000
```

Step 3. Install packages

User-defined functions come in **Packages**

Install the tidyverse Use the R function `install.packages()` to install the tidyverse from CRAN

```
install.packages("tidyverse", dependencies = TRUE)
```

Don't forget to always load your package before using:

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

Get beginner-friendly help from `qelp` Created by Colleen O'Briant: https://github.com/cobriant/qelp

```r
# we need these to install qelp:
install.packages("Rcpp", dependencies = TRUE)
install.packages("devtools", dependencies = TRUE)
```

answer `'yes'` when prompted in the console

Now you can install qelp:

```r
library(devtools)
install_github("cobriant/qelp")
```

Test that it worked:

```r
?qelp::install.packages
```

You can also use the default `R` documentation:

```r
?install.packages
```

See the difference?

Step 4. Download the Koans

Go to the github page: https://github.com/ajdickinson/tidyverse_koans

Click the `<> Code` button, then `Download ZIP`.

Unzip the files wherever you want them on your personal machine. This will be where you will keep all of your lab work for this class.

Double-click the file called `tidyverse_koans-master.Rproj`.

This should open a new `Project` in RStudio with all of the R files you need.

Open the `R` folder from the files you downloaded by clicking it in the `Files` tab in RStudio. Now click on the first lab assignment, `K01_vector.R`, and it will open in your `source` tab where you can view and edit it.

Complete Koan 1 - Vectors

You can use the hotkey `Ctrl/Cmd Shift C` to comment out the lines of code in between each set of question markers (e.g., from `#1@` to `@#1`).

Fill in the blanks by following the instructions.

Test your answers Use the shortcut `Shift Ctrl/Cmd T` to check whether your code is correct.

Compile and submit to Canvas Once your code has passed all of the tests, you will compile the R script as an html formatted output which you will upload to Canvas.

In the RStudio menu bar, go to *Tools*, then *Modify Keyboard Shortcuts*. Set the shortcut for *Compile Notebook* to be `Shift Ctrl/Cmd K` (or whatever shortcut you will remember).

You can also use the menu option in `File` for `Knit Document`.

Once RStudio is done compiling, you will have a file called `K01_vector.html` in the same folder as the original `.R` file.

Upload this to Canvas to get credit for this lab.