Introduction to R
2020-21

First Session and Some Tips

This illustration Assumes that You Have Installed R and R-Studio

Dataset needed for this illustration

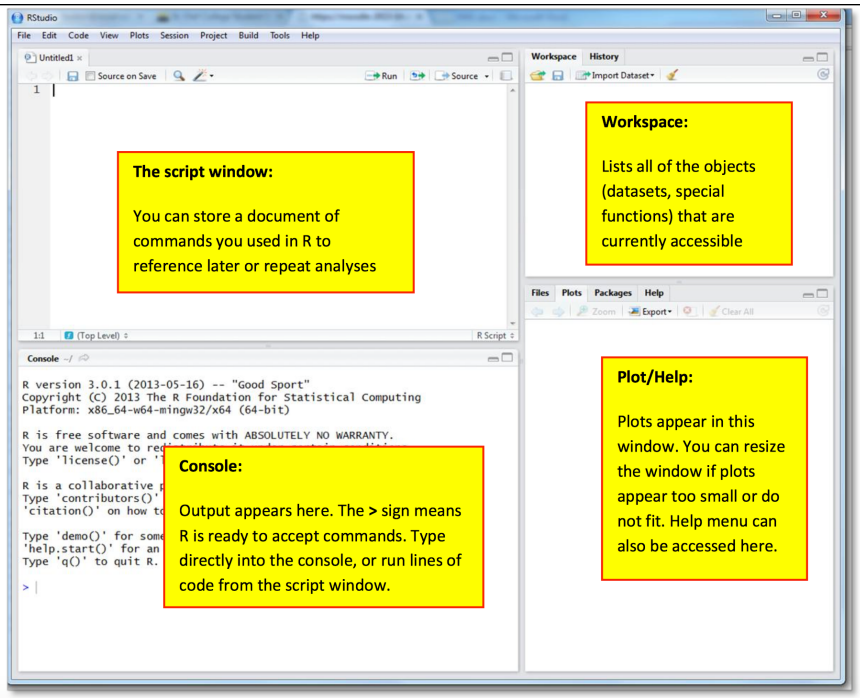
Right click to download from the BIOSTATS 540 course website (Welcome page > Illustrations) the following R dataset: framingham_1000.Rdata. Here is the direct link:

https://people.umass.edu/biep540w/datasets/framingham_1000.Rdata

IMPORTANT! Save it to your desktop.

Goals **Page**

1. The R Studio Environment *in more detail!*



2

Source: https://pages.stolaf.edu/boehm/files/2015/02/RStudioInfoFor272.pdf

Bottom Left: Console Window

- Here you type code directly. R then executes it.
- The prompt is indicated by a ">"
- HACK: To retrieve a previous command (*handy for editing!*): UP-arrow
- HACK: To clear contents of the console window: <control-L>

Top Left: R Script and R Markdown (if installed) Window

- Use FILE > NEW FILE > *your choice* to access R Script or R Markdown
- HACK: Use R script to create and save files of R commands (more on this later)
- HACK: Use R Markdown to create archives of R sessions (R commands + output)
- HACK: In R Script, to execute a command (good for troubleshooting): <control-ENTER>
- Tip: Different R Script files can be saved in different tabs
- HACK: Create R Script files that you save as "boilers" for future work

Top Right: Workspace Window

- Here you see all your stuff, called "objects" – datasets, variables, etc
- HACK: To view a "spreadsheet" version of your data: CLICK on the name in the workspace

Bottom Right: Plot/Help Window

- A lot of stuff here: plots, help w packages, importing from your computer, other help
- Be sure to "toggle" about to see what's here!

2. How to Work with Packages

Introduction

- Many packages came pre-installed so you already have them, eg: base and stats
- In much of your work, you will want to use commands in packages which you must download

Working with Packages Requires TWO steps:

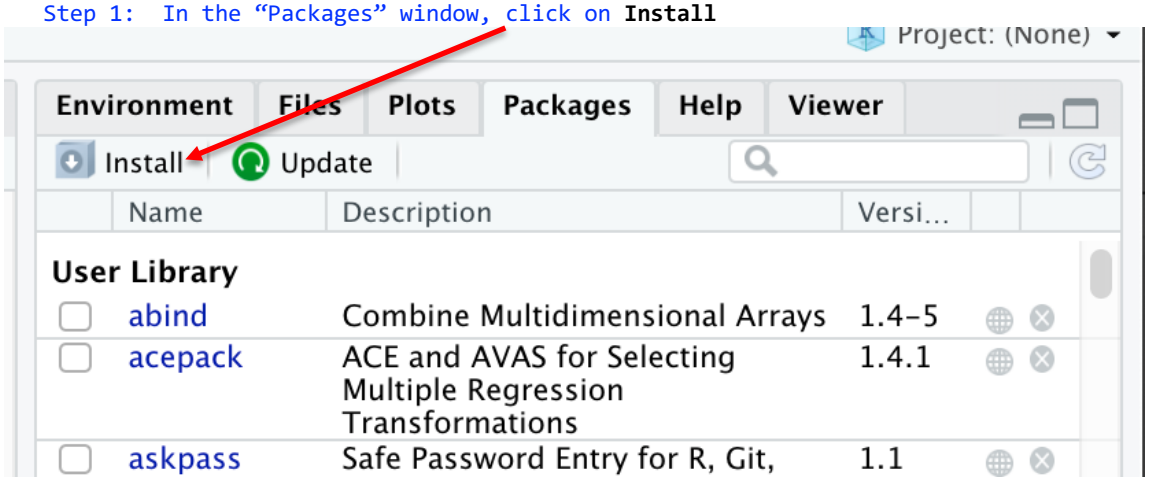
(1) A one-time installation, and;
(2) An every-time attachment of the package to your current R Studio session.

How to Install a Package

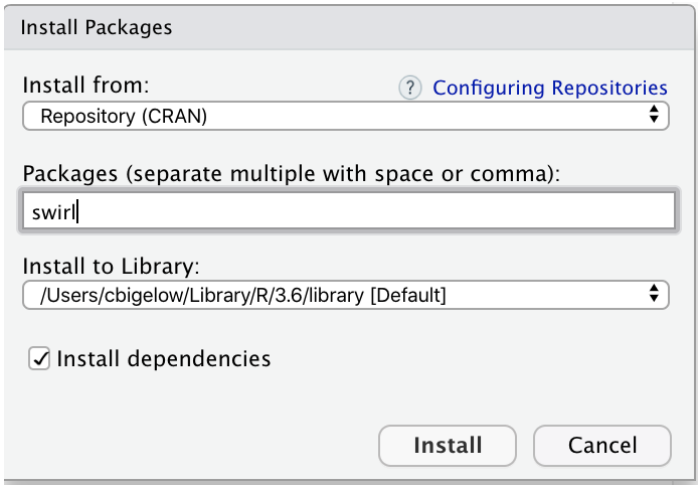
There's more than one way to download and install a package …

METHOD I: Menu Driven (Easy and recommended)

Step 1: In the "Packages" window, click on **Install**



Step 2: In the "Install Packages" window
Example: I want to install the package called swirl
In Install from: **default (Repository CRAN) is fine**
In Packages (separate multiple with space or comma:) **type in name of package (e.g. swirl)**
In Install to Library: **leave as is**
Check box for "Install dependencies": **check this**
At bottom, click **Install**



Step 3: Be patient. Wait. Then take a look at your console window. You should see:
At bottom you should see the prompt ">".

```
> install.packages("swirl")
Installing package into '/Users/cbigelow/Library/R/3.6/library'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/swirl_2.4.5.tgz'
Content type 'application/x-gzip' length 348039 bytes (339 KB)
==================================================
downloaded 339 KB


The downloaded binary packages are in
        /var/folders/rn/drwz6qbx3nb0ycr7vlppztt52nq3lw/T//RtmpSV27qs/downloaded_packages
>
```

METHOD II: Command Driven (From the console window)

How to Install a Package from the Console Window

install.packages("nameofpackageinquotes")

Example: install.packages("foreign")
HACK: Don't forget the period in install.packages
HACK: The name of the package MUST be enclosed in double quotes
HACK: Always install packages in the console window
HACK: Never install a package within a R Markdown file (more on this later)

How to Attach a Package to Your Session

How to Attach a Package to Your R Studio Session

library(nameofpackageNOTinquotes)

Example: library(foreign)

Some Useful help commands related to packages

- To see what packages you have installed now, from console: library()
- To see what packages are attached to your session, from console: search()
- To get location of library containing packages, from console: .libPaths()
- To get help, from console: help(package="foreign")
  NOTE: The help will appear in the help/plot window at bottom right

How to Upgrade a Package

- Tip: Be sure to visit https://www.datacamp.com/community/tutorials/r-packages-guide
- To see what packages are old and need updating: old.packages()
- To update ALL packages: update.packages()
- To update JUST ONE package is a 2 step procedure:
  STEP 1: Remove the old package. Example: update.packages("vioplot")
  STEP 2: Re-install the same package. Example: install.packages("vioplot")

3. Suggested Practices for Your R Studio Sessions

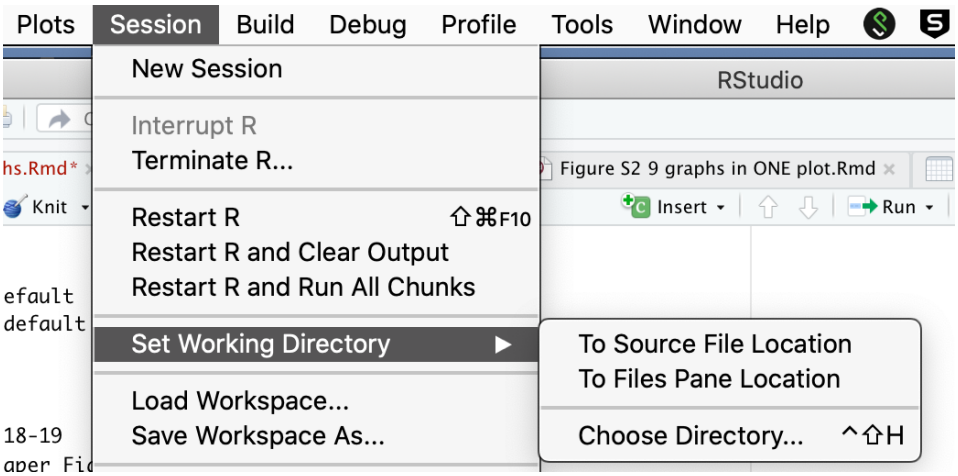
At the start of your R Studio Session

- Clear the workspace by issuing the following from the console: `rm(list=ls())`

Set Your Working Directory to a Path that is convenient and that you'll remember!
There's more than one way to download and install a package …

METHOD I: Menu Driven (Easy and recommended)

From the tool bar at top: **Session > Set Working Directory > Choose Directory**



- Check your working directory: `getwd()`

METHOD II: Command Driven (From the console window)

How to Set Your Working Directory
User provides path

Example:
Suppose you want your working directory to be your desktop (change yellow highlight):

MAC Users:
`setwd("/Users/cbigelow/Desktop/")`

WINDOWS 7 and Vista Users:
`setwd("c:/Users/cbigelow/Desktop/")`

WINDOWS 2000, 2003 and XP Users:
`setwd("c:/documents and settings/cbigelow/desktop/")`
OR: `setwd("c:/docume~1/cbigelow/desktop/")`

Check your working directory: `getwd()`

Create boilers! In particular, learn how to create and save R Script

- Use **FILE > NEW FILE > *your choice*** to access R Script
- In each R Studio session, you will be doing various things and troubleshooting as you go along.
- Once you have R code that runs successfully, save it in one or more R script files that you can use as boilers later.

Archive your work: In particular, learn How to Use R Markdown

- Use **FILE > NEW FILE > *your choice*** to access R Markdown
- We will learn how to work with R Markdown files in another lab
- But for now, consider writing out an analysis plan of each R session (modular is good!)
- Then create an R Markdown file of your work.

4. Opening and Saving R Data

BEFORE YOU BEGIN:
The following assumes that your working directory is your desktop

R Data (.RData)

- Place the R dataset that you downloaded on page 1 on your desktop (e.g: framingham_1000.Rdata)
- Set the working directory to your desktop, e.g: setwd("/Users/cbigelow/Desktop/")

How to Open R Data (.RData)

- LOAD your Rdata: load(file=" ").

  Example: load(file=" framingham_1000.Rdata")

  HACK: Take care to enclose name of data set in quotes

How to Save R Data (.RData)

- SAVE your Rdata (you will need to supply the correct object name): save(object, file=" ").

  Example: save(framingham, file=" framingham_1000.Rdata")

  HACK: The name of the .Rdata does NOT have to match the name of your workspace R object.

5. ILLUSTRATION: Data Description

> **BEFORE YOU BEGIN:**
> **The following assumes that your working directory is your desktop**

BEFORE YOU BEGIN: How R keeps track of dataset names and variable names.
R does this with a two level name: dataframe$variablename.

Example: framingham$sbp
dataframe = framingham
Variable name = sbp

Download framingham_1000.Rdata from the course website. Place on desktop.

Launch R Studio. Set working directory (yours will be slightly different). Load data

- setwd("/Users/cbigelow/Desktop/")
- load(file="framingham_1000.Rdata")

Create an R object (this will be a dataframe) called framingham so save having to type Framingham_1000

- framingham <- framingham_1000

One Time Install Packages (Remove leading comment indicator "#") at the console window:

- # install.packages("stargazer")
- # install.packages("summarytools")

Attach to your R Studio Session, the packages you will be using here

- library(stargazer)
- library(summarytools)

Data Description: Quick look for ALL variables Method I (no package needed)
summary(framingham)

```
summary(framingham)
```

```
##     sex           sbp             scl             age
##  Men  :443   Min.   : 80.0   Min.   :115.0   Min.   :30.00
##  Women:557   1st Qu.:116.0   1st Qu.:197.0   1st Qu.:38.75
##              Median :128.0   Median :225.0   Median :45.00
##              Mean   :132.3   Mean   :227.8   Mean   :45.92
##              3rd Qu.:144.0   3rd Qu.:255.0   3rd Qu.:53.00
##              Max.   :270.0   Max.   :493.0   Max.   :66.00
##                              NA's   :4
##       bmi              id           ln_bmi          ln_sbp
##  Min.   :16.40   Min.   :   1   Min.   :2.797   Min.   :4.382
##  1st Qu.:23.00   1st Qu.:1246   1st Qu.:3.135   1st Qu.:4.754
##  Median :25.10   Median :2488   Median :3.223   Median :4.852
##  Mean   :25.57   Mean   :2410   Mean   :3.230   Mean   :4.872
##  3rd Qu.:27.80   3rd Qu.:3605   3rd Qu.:3.325   3rd Qu.:4.970
##  Max.   :43.40   Max.   :4697   Max.   :3.770   Max.   :5.598
##  NA's   :2                      NA's   :2
##      ln_scl
##  Min.   :4.745
##  1st Qu.:5.283
##  Median :5.416
##  Mean   :5.410
##  3rd Qu.:5.541
##  Max.   :6.201
##  NA's   :4
```

Data Description: Quick look for ALL variables Method II (uses package stargazer, command stargazer)
library(stargazer)
stargazer**(framingham, type="text", median=TRUE)**

```
library(stargazer)
stargazer(framingham, type="text", median=TRUE)
```

```
##
## ===========================================================
## Statistic   N     Mean     St. Dev.   Min   Median   Max
## -----------------------------------------------------------
## sbp       1,000  132.350   23.043     80     128    270
## scl        996   227.846   45.087    115     225    493
## age       1,000  45.922     8.545     30     45      66
## bmi        998   25.566     3.848   16.400 25.100  43.400
## id        1,000 2,410.031 1,363.439   1    2,487.5 4,697
## ln_bmi     998    3.230     0.147   2.797   3.223  3.770
## ln_sbp    1,000   4.872     0.163   4.382   4.852  5.598
## ln_scl     996    5.410     0.195   4.745   5.416  6.201
## -----------------------------------------------------------
```

Data Description: CONTINUOUS variable – detail (uses package summarytools, command descr)

```
library(summarytools)
descr(framingham$sbp, stats = c("n.valid","mean", "sd", "min","q1", "med", "q3", "max","CV"),
 transpose = TRUE)
```

```
descr(framingham$sbp, stats = c("n.valid","mean", "sd", "min","q1", "med", "q3", "max","CV"),
 transpose = TRUE)

## Descriptive Statistics
## framingham$sbp
## N: 1000
##
##             N.Valid     Mean   Std.Dev     Min       Q1   Median       Q3      Max     CV
## --------- --------- -------- --------- ------- -------- -------- -------- -------- ------
##       sbp   1000.00   132.35     23.04   80.00   116.00   128.00   144.00   270.00   0.17
```

Data Description: DISCRETE variable – detail (uses package summarytools, command freq)

```
library(summarytools)
freq(framingham$sex)
```

```
freq(framingham$sex)

## Frequencies
## framingham$sex
## Type: Factor (unordered)
##
##               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ----------- ------ --------- -------------- --------- --------------
##         Men    443     44.30          44.30     44.30          44.30
##       Women    557     55.70         100.00     55.70         100.00
##        <NA>      0                               0.00         100.00
##       Total   1000    100.00         100.00    100.00         100.00
```

6. ILLUSTRATION: ggplot2 for graphs

I highly encourage you to learn ggplot2 (Layer by layer)

One Time (must be from console window): Install Packages (Remove leading comment indicator "#"):

- `# install.packages("ggplot2")`

Attach to your R Studio Session, the package you will be using here

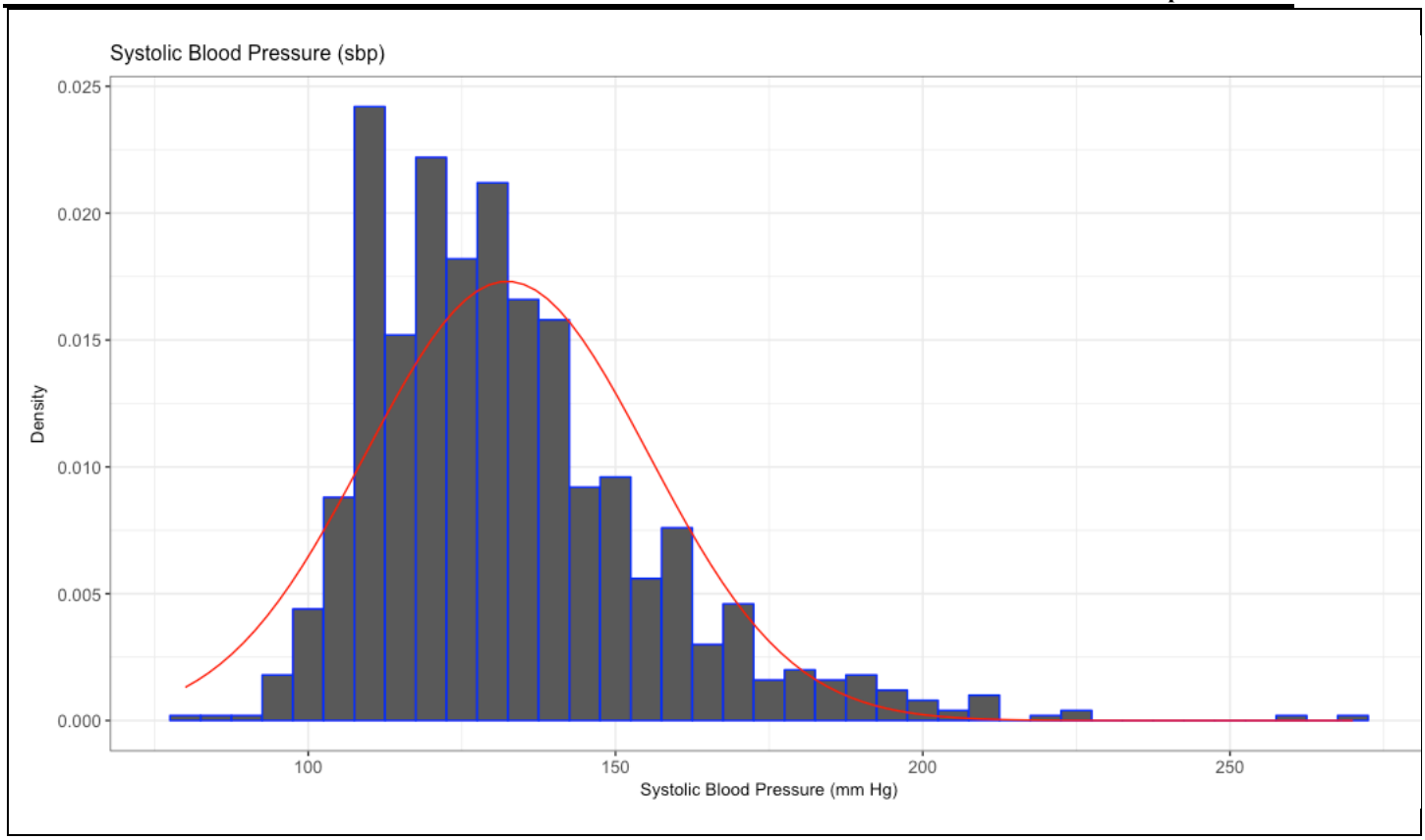
- `library(ggplot2)`

> **BEFORE YOU BEGIN – Learning ggplot2 "layer by layer":**
> ***Key:*** *Because I make lots of mistakes, I build my plots "layer by layer". To do this, I create a layer and assign it to an object that I call "p". Then I ask R to show me "p". If I'm lucky it works. If not, I get an error message. But it's just one layer to correct, not so hard to do. Once I've done all the layers and they all work, then I get rid of all the intermediate displays of layer by later. Last but not least, if I'm really happy with what I've just developed, I'll save it in one of my R Script boilers>*
>
> *See below and you'll see what I mean – cb.*

GRAPH: Histogram with Overlay Normal (uses package ggplot2, command ggplot w lots of "extras")

```
# ggplot(DATAFRAME, aes(x=VARIABLENAME)) + stuff below

p <- ggplot2::ggplot(framingham, aes(x=sbp))
p
p <- p + geom_histogram(binwidth=5, colour="blue", aes(y=..density..))
p
p <- p + stat_function(fun=dnorm,
                       color="red",
                       args=list(mean=mean(framingham$sbp),sd=sd(framingham$sbp)))
p
p <- p + ggtitle("Systolic Blood Pressure (sbp)")
p
p <- p + xlab("Systolic Blood Pressure (mm Hg)") + ylab("Density")
p
p <- p + theme_bw()
p
p <- p + theme(axis.text = element_text(size = 10),
        axis.title = element_text(size = 10),
        plot.title = element_text(size = 12))
p
```


Systolic Blood Pressure (sbp)
0.025
0.020
0.015
0.010
0.005
0.000
Density
100
150
200
250
Systolic Blood Pressure (mm Hg)

Summary

1. R-Studio Windows

Console Window (typically located lower left)	
Retrieve a previous command for editing and re-execution Clear contents of entire console window	UP-arrow <control-L>
R Script, R Markdown, if installed (typically located upper left)	
To open a new R Script file To open a new R Markdown In R Script, execute immediately	FILE > NEW FILE > R Script FILE > NEW FILE > R Markdown Highlight then <control-ENTER>

2. Working with Packages

Download/install new package (must do from console window) editing Attach package you want to use To see what packages you have installed (from console) To sew what packages are attached to this session To get help with package (from console) To see what packages need updating To update all packages	install.packages("nameinquotes") library(nameofpackagenoquotes) library() search() help(package="NAMEOFPACKAGEINQUOTES") old.packages() update.packages()
Hack: Preface a package specific command with the package name followed by two colons. *REASON: Sometimes multiple packages may have the same command. To be sure you execute the command you want from the package you want, this hack does the trick.*	**Command:** Packagename::command **Example:** Hmisc::label(Framingham$female) <- "female01"

3. Loading and Saving R Data

Hack: Set the working directory to be your desktop (not mine)	**Example:** setwd("/Users/cbigelow/Desktop/")
Load R data from your working directory	**Command:** load(file="FILLINNAME") **Example:** load(file="blood_storage.Rdata")
Save R data to your working directory	**Command:** save(object, file="FILLINNAME") **Example:** save(bloodfinal,file="blood_storage.Rdata")