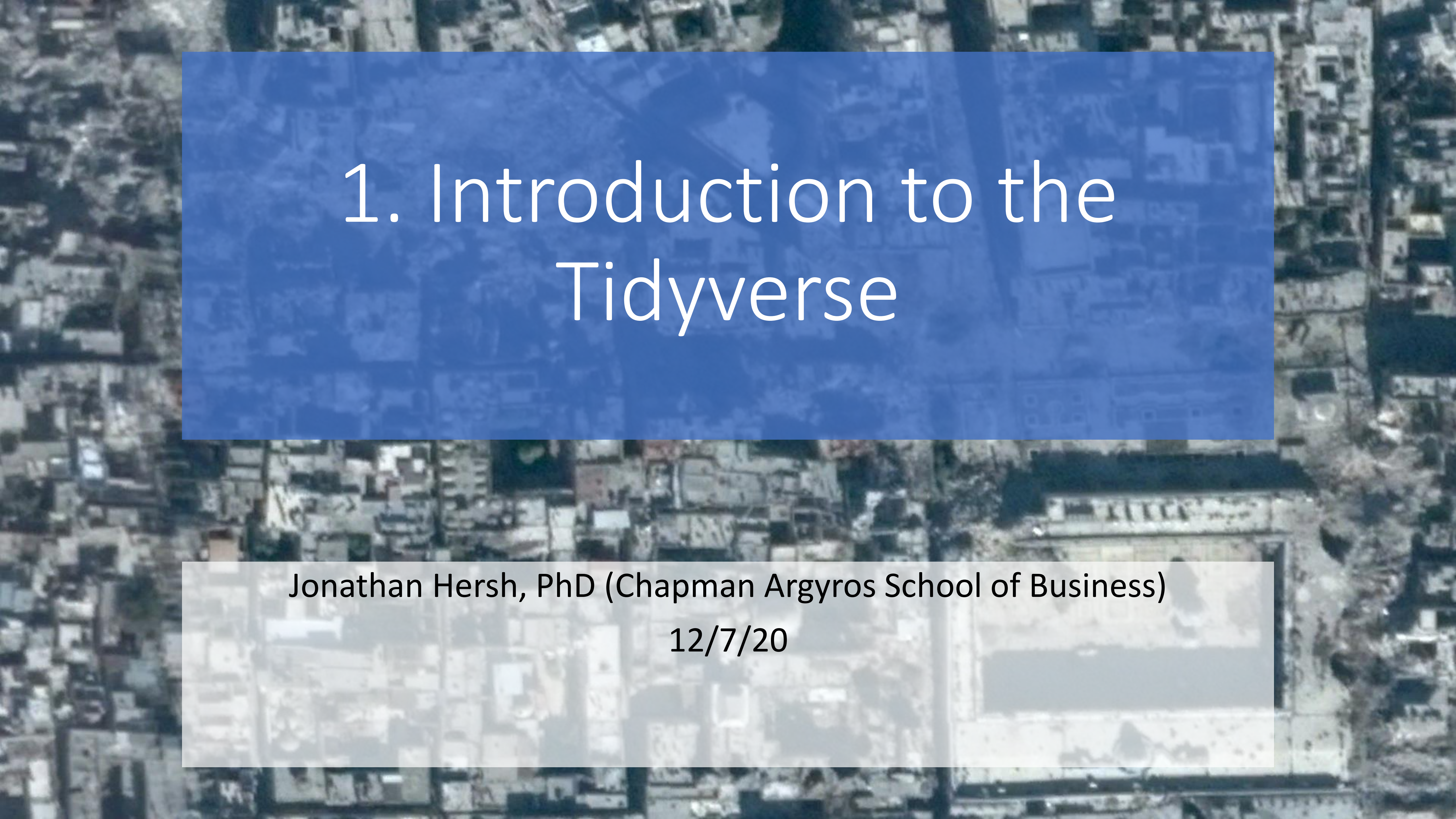


1. Introduction to the Tidyverse

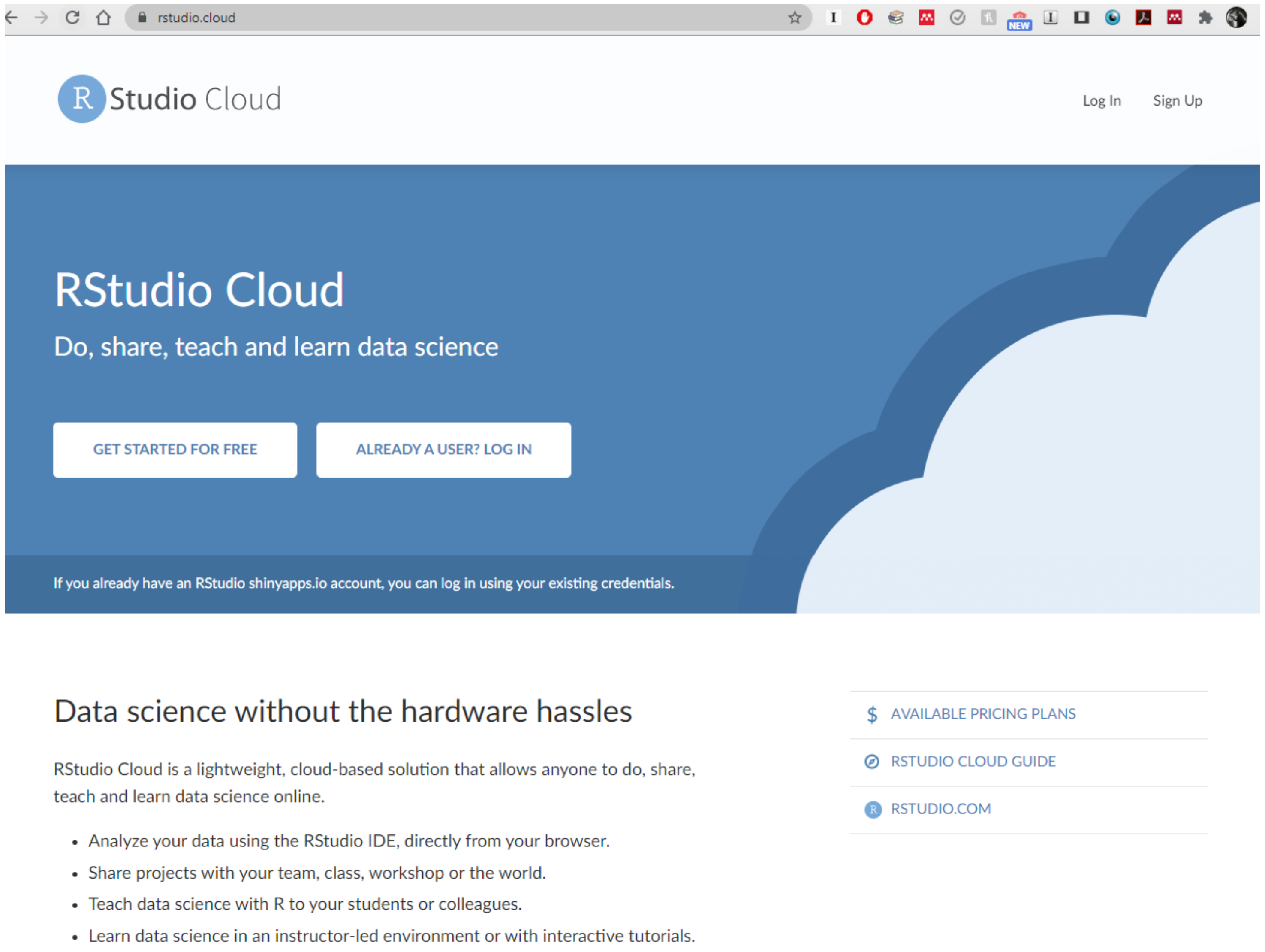
Jonathan Hersh, PhD (Chapman Argyros School of Business)

12/7/20

Section 1: Outline

- Loading data
- Glimpse to view
- Pipe operator
- slice() to select rows
- arrange() to order data frame
- select() to choose variables
- rename() to rename variables
- filter() to select rows matching characteristics
- mutate() to create new variables
- group_by() and summarize() to create group

R Studio Cloud



RStudio Cloud

Do, share, teach and learn data science

If you already have an RStudio shinyapps.io account, you can log in using your existing credentials.

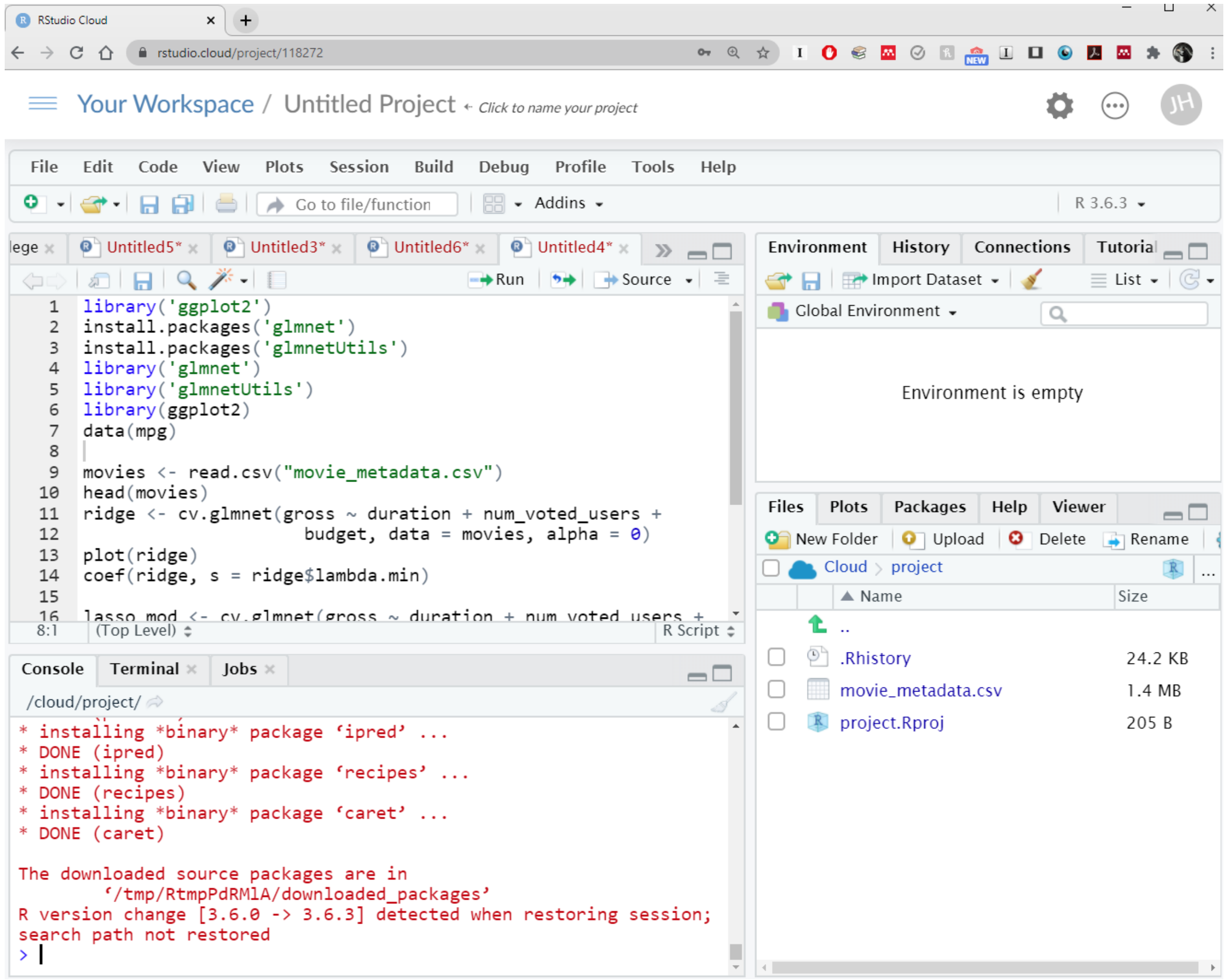
Data science without the hardware hassles

RStudio Cloud is a lightweight, cloud-based solution that allows anyone to do, share, teach and learn data science online.

- Analyze your data using the RStudio IDE, directly from your browser.
- Share projects with your team, class, workshop or the world.
- Teach data science with R to your students or colleagues.
- Learn data science in an instructor-led environment or with interactive tutorials.

- Go to rstudio.cloud if your version of R is ever not working

R Studio Cloud


RStudio Cloud
rstudio.cloud/project/118272
Your Workspace / Untitled Project ← Click to name your project
JH
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
R 3.6.3
lege Untitled5* Untitled3* Untitled6* Untitled4*
Run
Source
Environment History Connections Tutorial
Import Dataset
List
Global Environment
Environment is empty
8:1
(Top Level)
R Script
Console Terminal Jobs
/cloud/project/
Files Plots Packages Help Viewer
New Folder Upload Delete Rename
Cloud > project

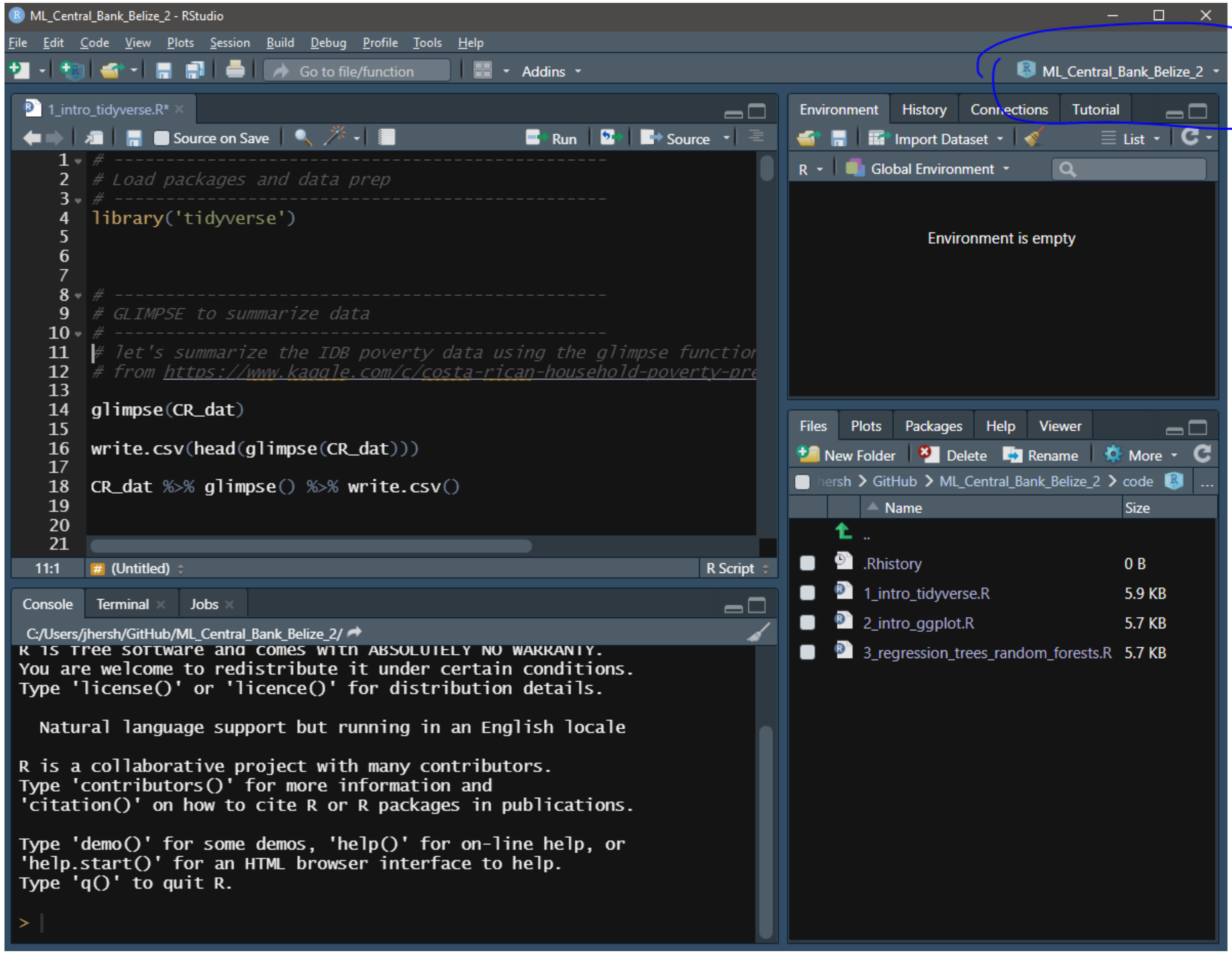
```
library('ggplot2')
install.packages('glmnet')
install.packages('glmnetUtils')
library('glmnet')
library('glmnetUtils')
library(ggplot2)
data(mpg)

movies <- read.csv("movie_metadata.csv")
head(movies)
ridge <- cv.glmnet(gross ~ duration + num_voted_users +
                     budget, data = movies, alpha = 0)
plot(ridge)
coef(ridge, s = ridge$lambda.min)

lasso_mod <- cv.glmnet(gross ~ duration + num_voted_users +
```

```
* installing *binary* package ‘ipred’ ...
* DONE (ipred)
* installing *binary* package ‘recipes’ ...
* DONE (recipes)
* installing *binary* package ‘caret’ ...
* DONE (caret)

The downloaded source packages are in
        ‘/tmp/RtmpPdRMlA/downloaded_packages’
R version change [3.6.0 -> 3.6.3] detected when restoring session;
search path not restored
>
```

Name	Size
..	
.Rhistory	24.2 KB
movie_metadata.csv	1.4 MB
project.Rproj	205 B

- R Studio Cloud is a full featured version of R in your browser!

R Studio Projects


ML_Central_Bank_Belize_2 - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
ML_Central_Bank_Belize_2
1_intro_tidyverse.R*
Source on Save
Run
Source
Environment History Connections Tutorial
Import Dataset
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder Delete Rename More
11:1
(Untitled)
R Script
Console Terminal Jobs

```
# ----------------------------------------------------------------
# Load packages and data prep
# ----------------------------------------------------------------
library('tidyverse')



# ----------------------------------------------------------------
# GLIMPSE to summarize data
# ----------------------------------------------------------------
# let's summarize the IDB poverty data using the glimpse functior
# from https://www.kaggle.com/c/costa-rican-household-poverty-pre

glimpse(CR_dat)

write.csv(head(glimpse(CR_dat)))

CR_dat %>% glimpse() %>% write.csv()
```

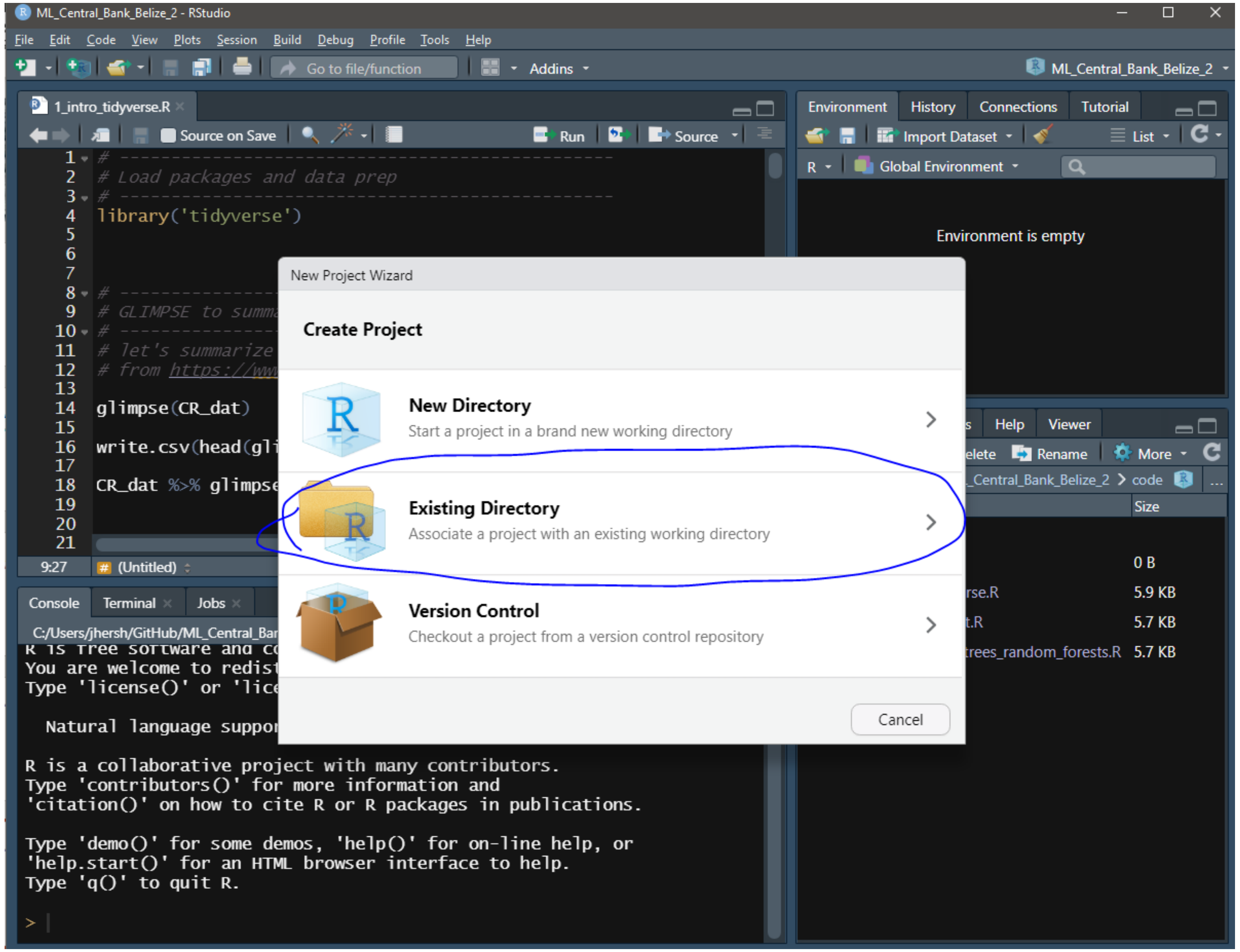
```
C:/Users/jhersh/GitHub/ML_Central_Bank_Belize_2/
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

hersh > GitHub > ML_Central_Bank_Belize_2 > code

Name	Size
..	
.Rhistory	0 B
1_intro_tidyverse.R	5.9 KB
2_intro_ggplot.R	5.7 KB
3_regression_trees_random_forests.R	5.7 KB

Be sure to work in a project file in R Studio

Creating an R Studio Project



```
# ----------------------------------------------------
# Load packages and data prep
# ----------------------------------------------------
library('tidyverse')



# ---------------
# GLIMPSE to summa
# ---------------
# let's summarize
# from https://ww

glimpse(CR_dat)

write.csv(head(gli

CR_dat %>% glimpse
```

New Project Wizard

Create Project

New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Cancel

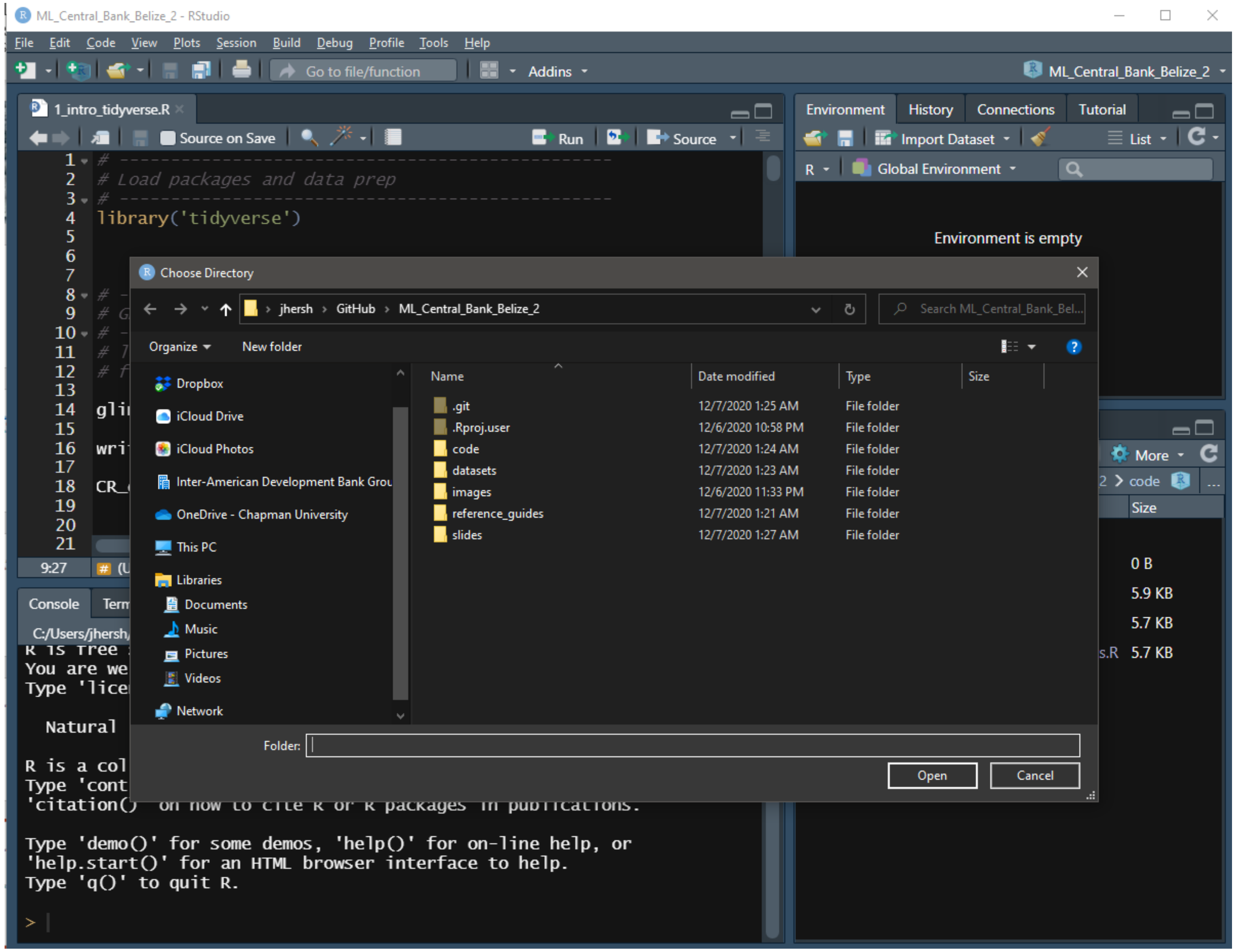
	0 B
rse.R	5.9 KB
t.R	5.7 KB
trees_random_forests.R	5.7 KB

```
R is free software and co
You are welcome to redist
Type 'license()' or 'lice

  Natural language suppor

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

- Be sure to work in a project file in R Studio

- Click file -> new project -> Existing Directory

Creating an R Studio Project



ML_Central_Bank_Belize_2 - RStudio

1_intro_tidyverse.R

```
# -----------------------------------------------
# Load packages and data prep
# -----------------------------------------------
library('tidyverse')
```

Environment is empty

Choose Directory

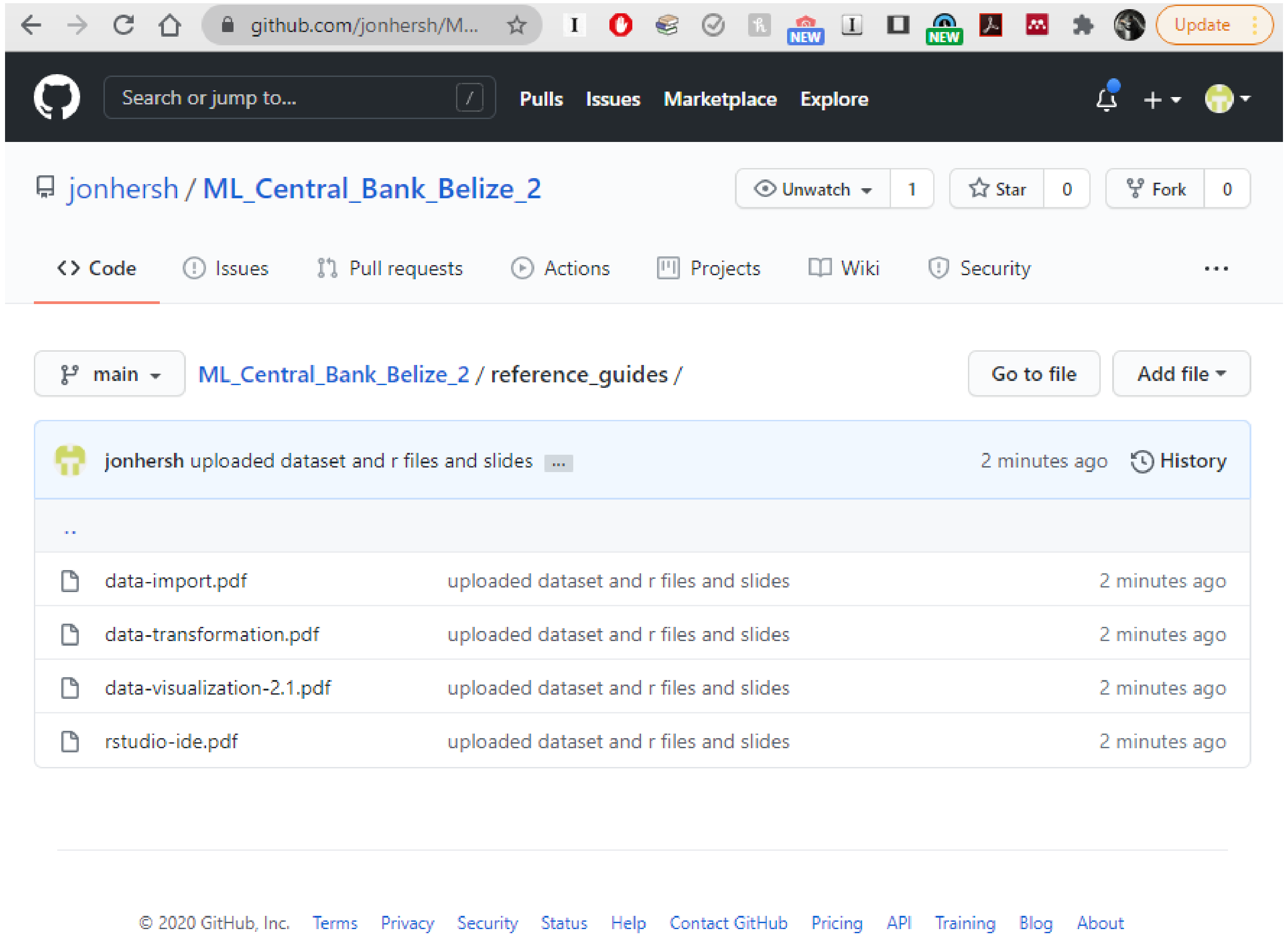
jhersh › GitHub › ML_Central_Bank_Belize_2

Name	Date modified	Type	Size
.git	12/7/2020 1:25 AM	File folder	
.Rproj.user	12/6/2020 10:58 PM	File folder	
code	12/7/2020 1:24 AM	File folder	
datasets	12/7/2020 1:23 AM	File folder	
images	12/6/2020 11:33 PM	File folder	
reference_guides	12/7/2020 1:21 AM	File folder	
slides	12/7/2020 1:27 AM	File folder	

```
R is free
You are we
Type 'lice

  Natural

R is a col
Type 'cont
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

- Be sure to work in a project file in R Studio
- Click file -> new project -> Existing Directory
- Then navigate to the folder where you are storing all of today's files

Reference Guides for Data Transformation, Import, Plotting and RStudio



jonhersh / ML_Central_Bank_Belize_2

ML_Central_Bank_Belize_2 / reference_guides /

jonhersh uploaded dataset and r files and slides ... 2 minutes ago

..

data-import.pdf	uploaded dataset and r files and slides	2 minutes ago
data-transformation.pdf	uploaded dataset and r files and slides	2 minutes ago
data-visualization-2.1.pdf	uploaded dataset and r files and slides	2 minutes ago
rstudio-ide.pdf	uploaded dataset and r files and slides	2 minutes ago

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

https://rstudio.com/resources/cheatsheets/

Data Import : : CHEAT SHEET

R's **tidyverse** is built around **tidy data** stored in **tibbles**, which are enhanced data frames.

The front side of this sheet shows how to read text files into R with **readr**.

The reverse side shows how to create tibbles with **tibble** and to layout tidy data with **tidyr**.

OTHER TYPES OF DATA

Try one of the following packages to import other types of files

- **haven** - SPSS, Stata, and SAS files
- **readxl** - excel files (.xls and .xlsx)
- **DBI** - databases
- **jsonlite** - json
- **xml2** - XML
- **httr** - Web APIs
- **rvest** - HTML (Web Scraping)

Save Data

Save **x**, an R object, to **path**, a file path, as:

Comma delimited file
write_csv(x, path, na = "NA", append = FALSE, col_names = !append)

File with arbitrary delimiter
write_delim(x, path, delim = " ", na = "NA", append = FALSE, col_names = !append)

CSV for excel
write_excel_csv(x, path, na = "NA", append = FALSE, col_names = !append)

String to file
write_file(x, path, append = FALSE)

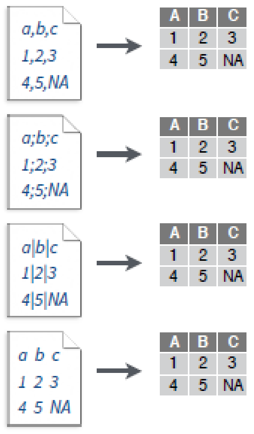
String vector to file, one element per line
write_lines(x,path, na = "NA", append = FALSE)

Object to RDS file
write_rds(x, path, compress = c("none", "gz", "bz2", "xz"), ...)

Tab delimited files
write_tsv(x, path, na = "NA", append = FALSE, col_names = !append)

Read Tabular Data - These functions share the common arguments:

```
read_*(file, col_names = TRUE, col_types = NULL, locale = default_locale(), na = c("", "NA"),
  quoted_na = TRUE, comment = "", trim_ws = TRUE, skip = 0, n_max = Inf, guess_max = min(1000,
  n_max), progress = interactive())
```



Comma Delimited Files
read_csv("file.csv")
To make file.csv run:
write_file(x = "a,b,c\n1,2,3\n4,5,NA", path = "file.csv")

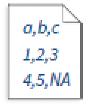
Semi-colon Delimited Files
read_csv2("file2.csv")
write_file(x = "a;b;c\n1;2;3\n4;5;NA", path = "file2.csv")

Files with Any Delimiter
read_delim("file.txt", delim = "|")
write_file(x = "a|b|c\n1|2|3\n4|5|NA", path = "file.txt")

Fixed Width Files
read_fwf("file.fwf", col_positions = c(1, 3, 5))
write_file(x = "a b c\n1 2 3\n4 5 NA", path = "file.fwf")

Tab Delimited Files
read_tsv("file.tsv") Also **read_table()**.
write_file(x = "a\tb\tc\n1\t2\t3\n4\t5\tNA", path = "file.tsv")

USEFUL ARGUMENTS



Example file
write_file("a,b,c\n1,2,3\n4,5,NA","file.csv")
f <- "file.csv"

1	2	3
4	5	NA

Skip lines
read_csv(f, **skip = 1**)

A	B	C
1	2	3
4	5	NA

No header
read_csv(f, **col_names = FALSE**)

A	B	C
1	2	3

Read in a subset
read_csv(f, **n_max = 1**)

x	y	z
A	B	C
1	2	3
4	5	NA

Provide header
read_csv(f, **col_names = c("x", "y", "z")**)

A	B	C
NA	2	3
4	5	NA

Missing Values
read_csv(f, **na = c("1", ".")**)

Read Non-Tabular Data

Read a file into a single string
read_file(file, locale = default_locale())

Read each line into its own string
read_lines(file, skip = 0, n_max = -1L, na = character(), locale = default_locale(), progress = interactive())

Read Apache style log files
read_log(file, col_names = FALSE, col_types = NULL, skip = 0, n_max = -1, progress = interactive())

Read a file into a raw vector
read_file_raw(file)

Read each line into a raw vector
read_lines_raw(file, skip = 0, n_max = -1L, progress = interactive())

Data types

readr functions guess the types of each column and convert types when appropriate (but will NOT convert strings to factors automatically).

A message shows the type of each column in the result.

```
## Parsed with column specification:
## cols(
##   age = col_integer(),
##   sex = col_character(),
##   earn = col_double()
## )
```

age is an integer

sex is a character

earn is a double (numeric)

1. Use **problems**() to diagnose problems.
x <- read_csv("file.csv"); problems(x)

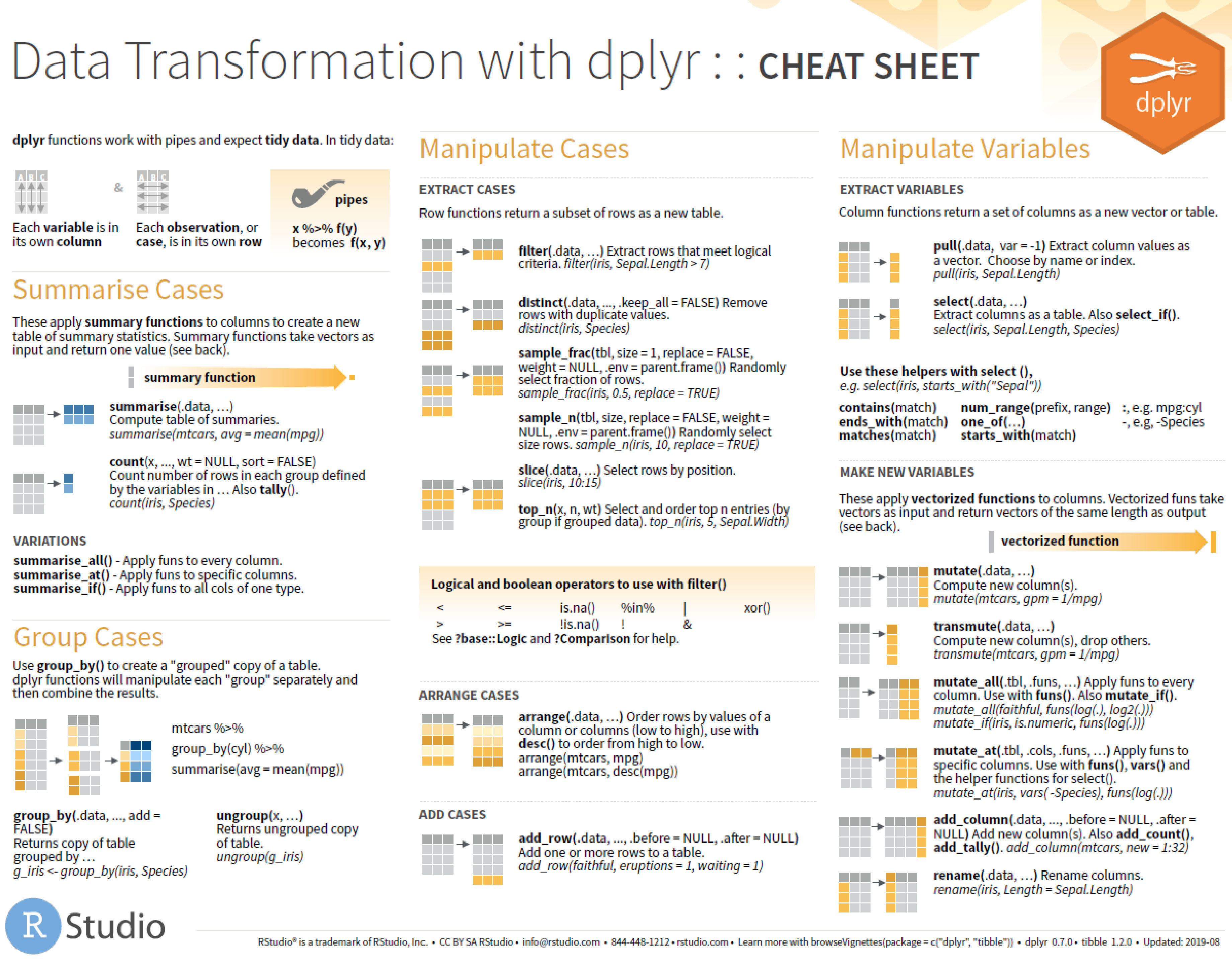
2. Use a col_ function to guide parsing.
- **col_guess()** - the default
- **col_character()**
- **col_double(), col_euro_double()**
- **col_datetime**(format = "") Also **col_date**(format = ""), **col_time**(format = "")
- **col_factor**(levels, ordered = FALSE)
- **col_integer()**
- **col_logical()**
- **col_number(), col_numeric()**
- **col_skip()**

x <- read_csv("file.csv", col_types = cols(
A = col_double(),
B = col_logical(),
C = col_factor()))

3. Else, read in as character vectors then parse with a parse_ function.
- **parse_guess()**
- **parse_character()**
- **parse_datetime()** Also **parse_date()** and **parse_time()**
- **parse_double()**
- **parse_factor()**
- **parse_integer()**
- **parse_logical()**
- **parse_number()**

x$A <- parse_number(x$A)

Studio



Data Transformation with dplyr : : CHEAT SHEET

dplyr

dplyr functions work with pipes and expect **tidy data**. In tidy data:

Each **variable** is in its own **column** & Each **observation**, or **case**, is in its own **row**

pipes

x %>% f(y) becomes **f(x, y)**

Summarise Cases

These apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).

summary function

summarise(.data, …)
Compute table of summaries.
summarise(mtcars, avg = mean(mpg))

count(x, ..., wt = NULL, sort = FALSE)
Count number of rows in each group defined by the variables in … Also **tally**().
count(iris, Species)

VARIATIONS

summarise_all() - Apply funs to every column.
summarise_at() - Apply funs to specific columns.
summarise_if() - Apply funs to all cols of one type.

Group Cases

Use **group_by()** to create a "grouped" copy of a table. dplyr functions will manipulate each "group" separately and then combine the results.

```
mtcars %>%
group_by(cyl) %>%
summarise(avg = mean(mpg))
```

group_by(.data, ..., add = FALSE)
Returns copy of table grouped by …
g_iris <- group_by(iris, Species)

ungroup(x, …)
Returns ungrouped copy of table.
ungroup(g_iris)

Manipulate Cases

EXTRACT CASES

Row functions return a subset of rows as a new table.

filter(.data, …) Extract rows that meet logical criteria. *filter(iris, Sepal.Length > 7)*

distinct(.data, ..., .keep_all = FALSE) Remove rows with duplicate values.
distinct(iris, Species)

sample_frac(tbl, size = 1, replace = FALSE, weight = NULL, .env = parent.frame()) Randomly select fraction of rows.
sample_frac(iris, 0.5, replace = TRUE)

sample_n(tbl, size, replace = FALSE, weight = NULL, .env = parent.frame()) Randomly select size rows. *sample_n(iris, 10, replace = TRUE)*

slice(.data, …) Select rows by position.
slice(iris, 10:15)

top_n(x, n, wt) Select and order top n entries (by group if grouped data). *top_n(iris, 5, Sepal.Width)*

Logical and boolean operators to use with filter()

<	<=	is.na()	%in%	|		xor()
>	>=	!is.na()	!		&	

See **?base::Logic** and **?Comparison** for help.

ARRANGE CASES

arrange(.data, …) Order rows by values of a column or columns (low to high), use with **desc()** to order from high to low.
arrange(mtcars, mpg)
arrange(mtcars, desc(mpg))

ADD CASES

add_row(.data, ..., .before = NULL, .after = NULL)
Add one or more rows to a table.
add_row(faithful, eruptions = 1, waiting = 1)

Manipulate Variables

EXTRACT VARIABLES

Column functions return a set of columns as a new vector or table.

pull(.data, var = -1) Extract column values as a vector. Choose by name or index.
pull(iris, Sepal.Length)

select(.data, …)
Extract columns as a table. Also **select_if()**.
select(iris, Sepal.Length, Species)

Use these helpers with select (),
e.g. select(iris, starts_with("Sepal"))

contains(match)	**num_range**(prefix, range)	:, e.g. mpg:cyl
ends_with(match)	**one_of**(…)	-, e.g, -Species
matches(match)	**starts_with**(match)	

MAKE NEW VARIABLES

These apply **vectorized functions** to columns. Vectorized funs take vectors as input and return vectors of the same length as output (see back).

vectorized function

mutate(.data, …)
Compute new column(s).
mutate(mtcars, gpm = 1/mpg)

transmute(.data, …)
Compute new column(s), drop others.
transmute(mtcars, gpm = 1/mpg)

mutate_all(.tbl, .funs, …) Apply funs to every column. Use with **funs()**. Also **mutate_if()**.
mutate_all(faithful, funs(log(.), log2(.)))
mutate_if(iris, is.numeric, funs(log(.)))

mutate_at(.tbl, .cols, .funs, …) Apply funs to specific columns. Use with **funs()**, **vars()** and the helper functions for select().
mutate_at(iris, vars(-Species), funs(log(.)))

add_column(.data, ..., .before = NULL, .after = NULL) Add new column(s). Also **add_count()**, **add_tally()**. *add_column(mtcars, new = 1:32)*

rename(.data, …) Rename columns.
rename(iris, Length = Sepal.Length)

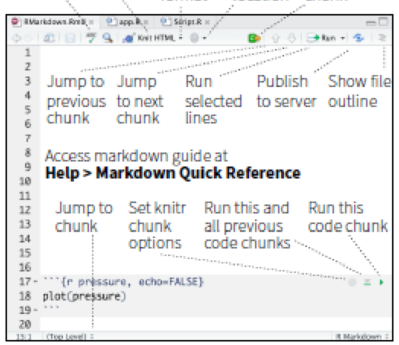
R Studio

RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more with browseVignettes(package = c("dplyr", "tibble")) • dplyr 0.7.0 • tibble 1.2.0 • Updated: 2019-08

RStudio IDE : : CHEAT SHEET

Documents and Apps

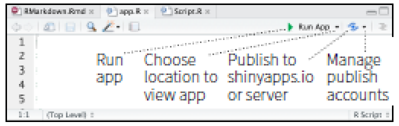
Open Shiny, R Markdown, knitr, Sweave, LaTeX, .Rd files and more in Source Pane



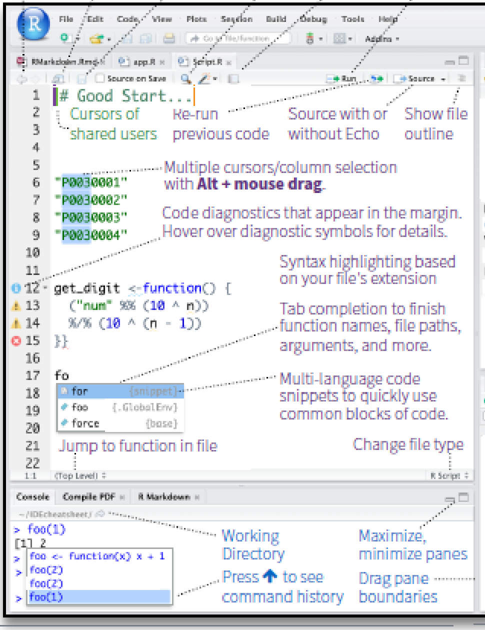
Access markdown guide at **Help > Markdown Quick Reference**

~~~
```{r pressure, echo=FALSE}
plot(pressure)
```
~~~

RStudio recognizes that files named **app.R**, **server.R**, **ui.R**, and **global.R** belong to a shiny app



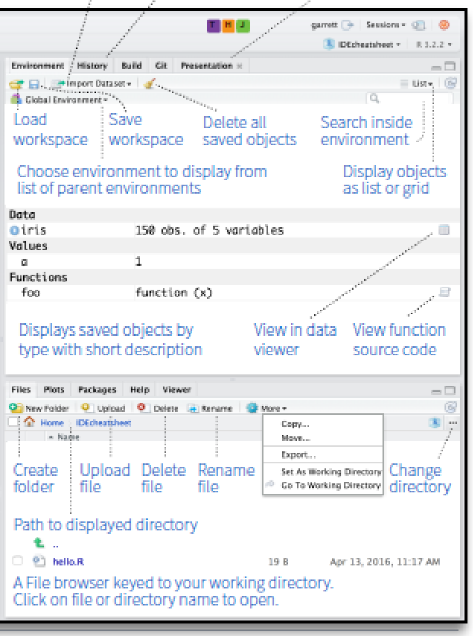
Write Code



```
# Good Start...

"P0030001"
"P0030002"
"P0030003"
"P0030004"

get_digit <-function() {
  ("num" %% (10 ^ n))
  %/% (10 ^ (n - 1))
}

fo
```

```
> foo(1)
[1] 2
> foo <- function(x) x + 1
> foo(2)
> foo(2)
> foo(1)
```

R Support



```
Data
iris        150 obs. of 5 variables
Values
a           1
Functions
foo         function (x)
```

Pro Features

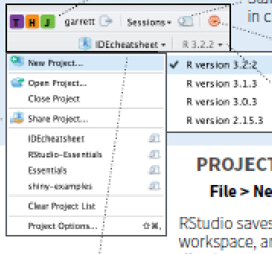
Share Project with Collaborators

Active shared collaborators

Start **new R Session** in current project

Close R Session in project

Select R Version



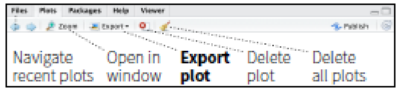
Name of current project

PROJECT SYSTEM

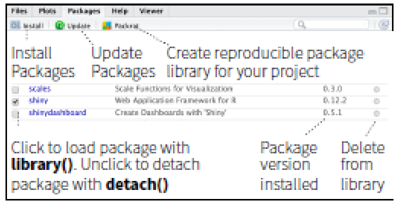
File > New Project

RStudio saves the call history, workspace, and working directory associated with a project. It reloads each when you re-open a project.

RStudio opens plots in a dedicated Plots pane

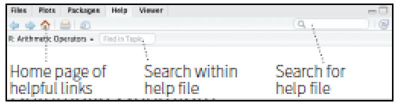


GUI Package manager lists every installed package

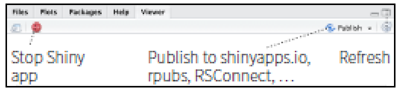


Click to load package with **library()**. Unclick to detach package with **detach()**

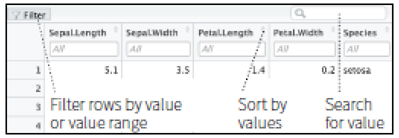
RStudio opens documentation in a dedicated Help pane



Viewer Pane displays HTML content, such as Shiny apps, RMarkdown reports, and interactive visualizations

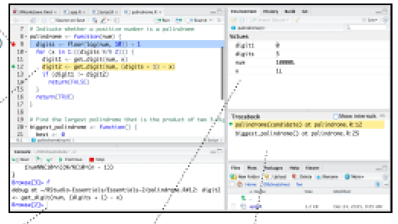


View(<data>) opens spreadsheet like view of data set



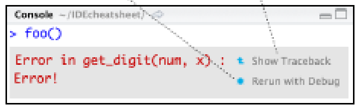
Debug Mode

Open with **debug()**, **browser()**, or a breakpoint. RStudio will open the debugger mode when it encounters a breakpoint while executing code.

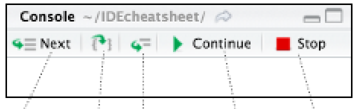


Launch debugger mode from origin of error

Open traceback to examine the functions that R called before the error occurred

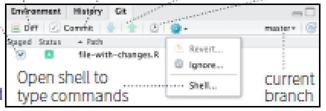


```
> foo()
Error in get_digit(num, x) :
Error!
```



Version Control with Git or SVN

Turn on at **Tools > Project Options > Git/SVN**



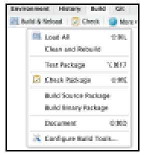
- 🟩 Added
- 🟥 Deleted
- 🟦 Modified
- 🟪 Renamed
- 🟨 Untracked

Package Writing

File > New Project > New Directory > R Package

Turn project into package, Enable roxygen documentation with **Tools > Project Options > Build Tools**

Roxygen guide at **Help > Roxygen Quick Reference**



RStudio