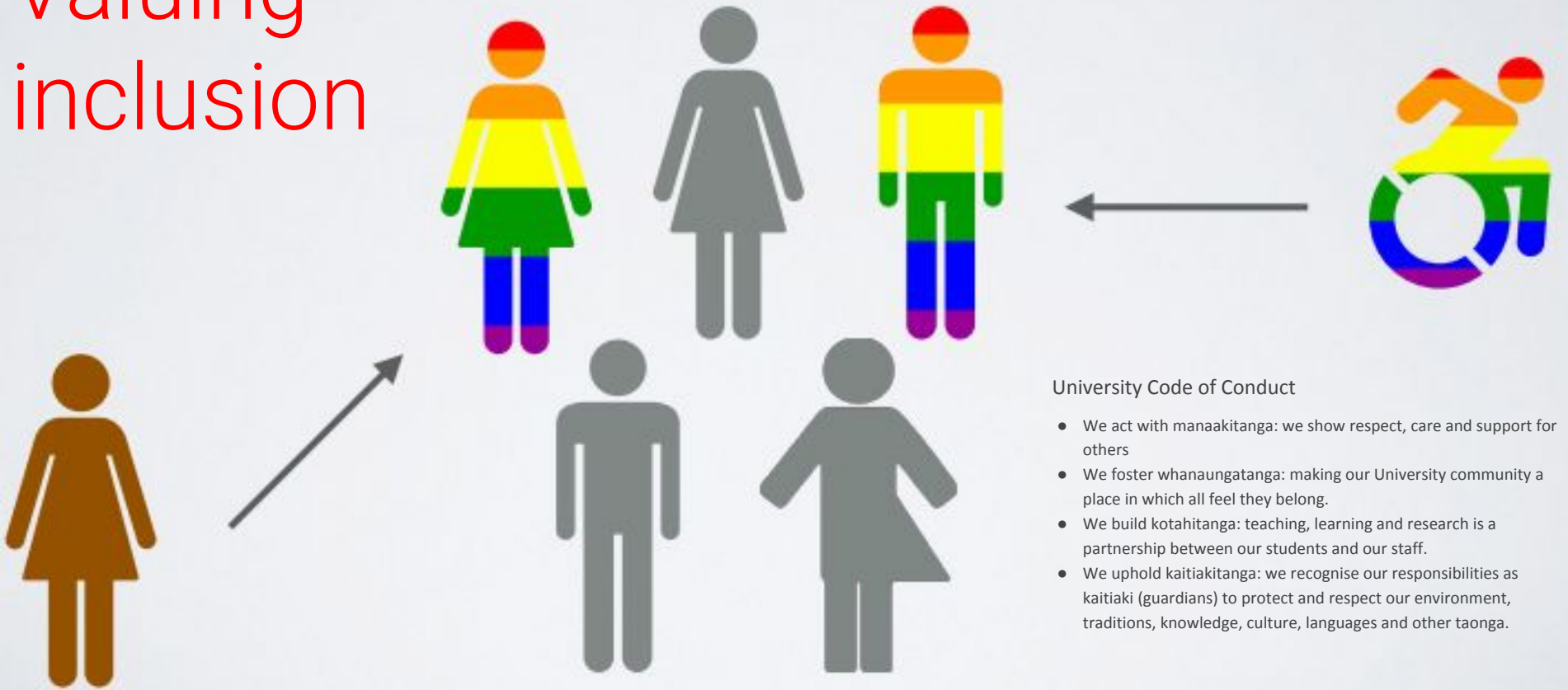
ResBaz 2020 Pick n Mix
Intro to R and RStudio

Yvette Wharton + Libby Li
Centre for eResearch
Te Whare Wānanga o Tāmaki Makaurau
The University of Auckland
Pronouns: she, her

#resbaz2020 #resbazpicnmix

Valuing inclusion



University Code of Conduct

- We act with manaakitanga: we show respect, care and support for others
- We foster whanaungatanga: making our University community a place in which all feel they belong.
- We build kotahitanga: teaching, learning and research is a partnership between our students and our staff.
- We uphold kaitiakitanga: we recognise our responsibilities as kaitiaki (guardians) to protect and respect our environment, traditions, knowledge, culture, languages and other taonga.

ResBaz 2020: Pick n Mix

23-27 Nov.

Key Stories - join us to listen over lunch

Tues. 12-1pm	*Harnessing the disruptive nature of portable sequencing for community empowerment* https://vuw.zoom.us/j/95143657235 Passcode: 718144
Wed. 12-1pm	*From Classics to Computer Science and back again* https://vuw.zoom.us/j/98866515882
Thur. 12-1pm	*Performance - Music performance and project showcase* https://vuw.zoom.us/j/98265221971
Fri. 12-1pm	*Exploring the cultural horizons of open science: your research and your life* https://auckland.zoom.us/j/98447731180 Passcode: 404029

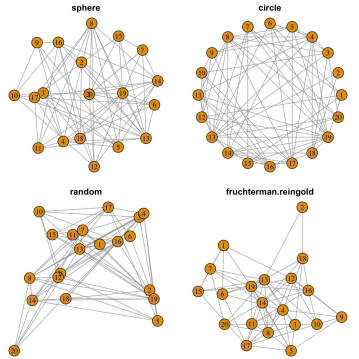
#resbaz2020 #resbazpicnmix email: researchdata@auckland.ac.nz

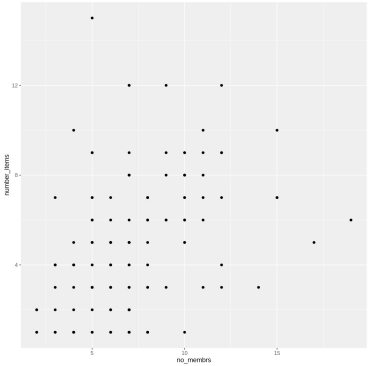
Outline

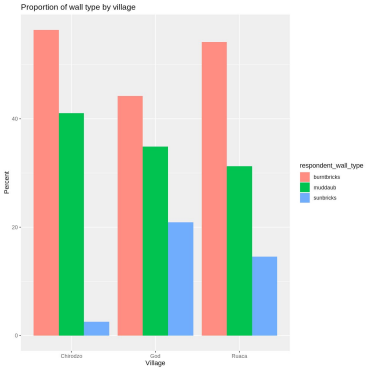
1. Overview of R Studio interface
2. Demystify some of the language in R
3. Work with a real dataset to do some cool things

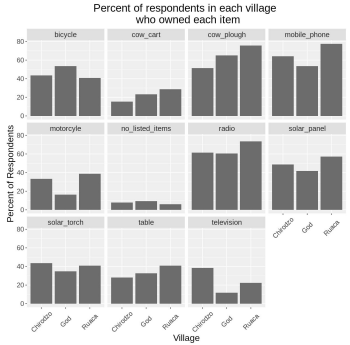
What is R

- A programming language and an environment
- Runs on Mac, Linux and Windows OS
- Open source free software
- Huge community
- wide variety of statistical and graphical techniques (https://www.r-graph-gallery.com/)



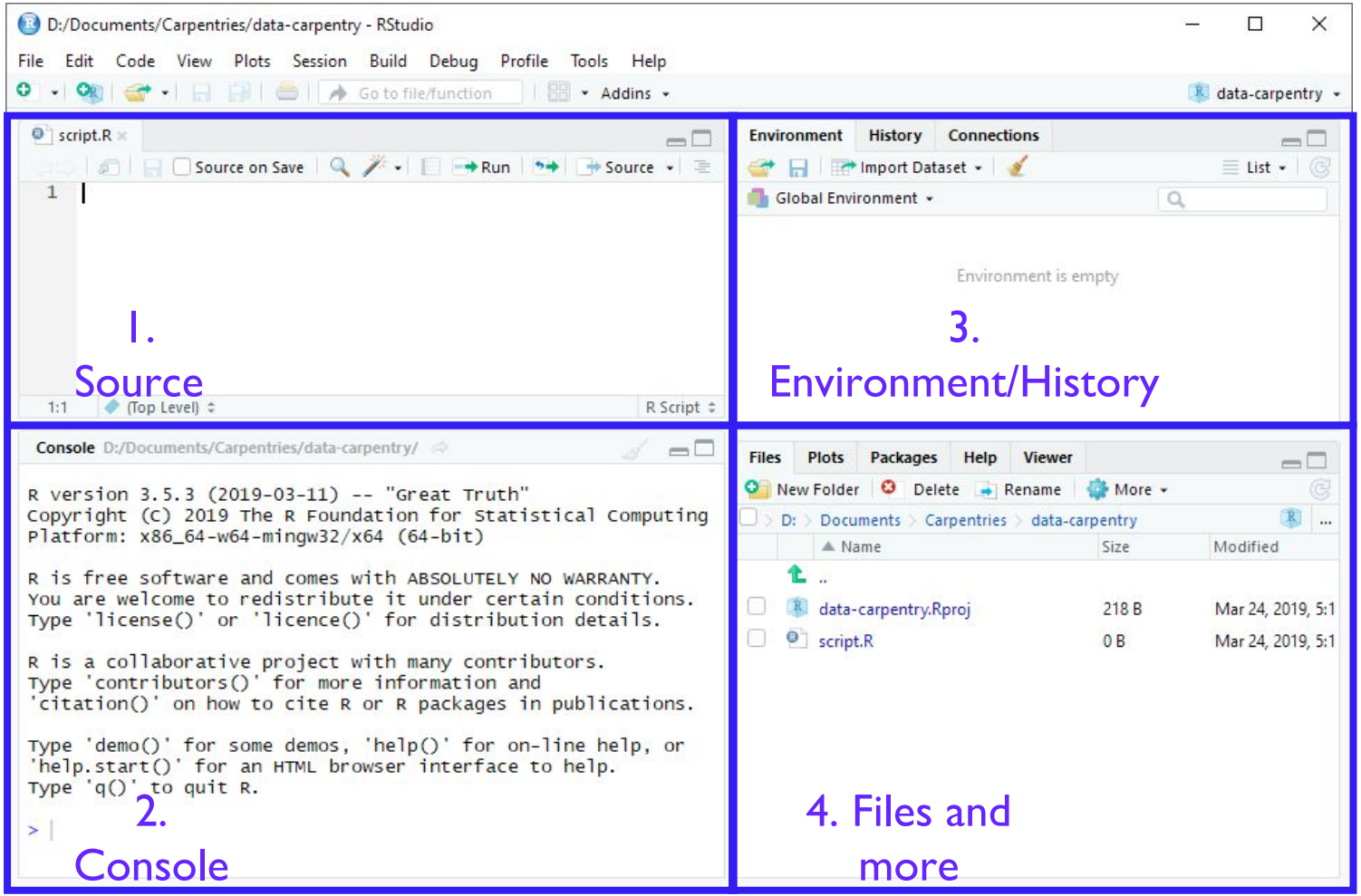


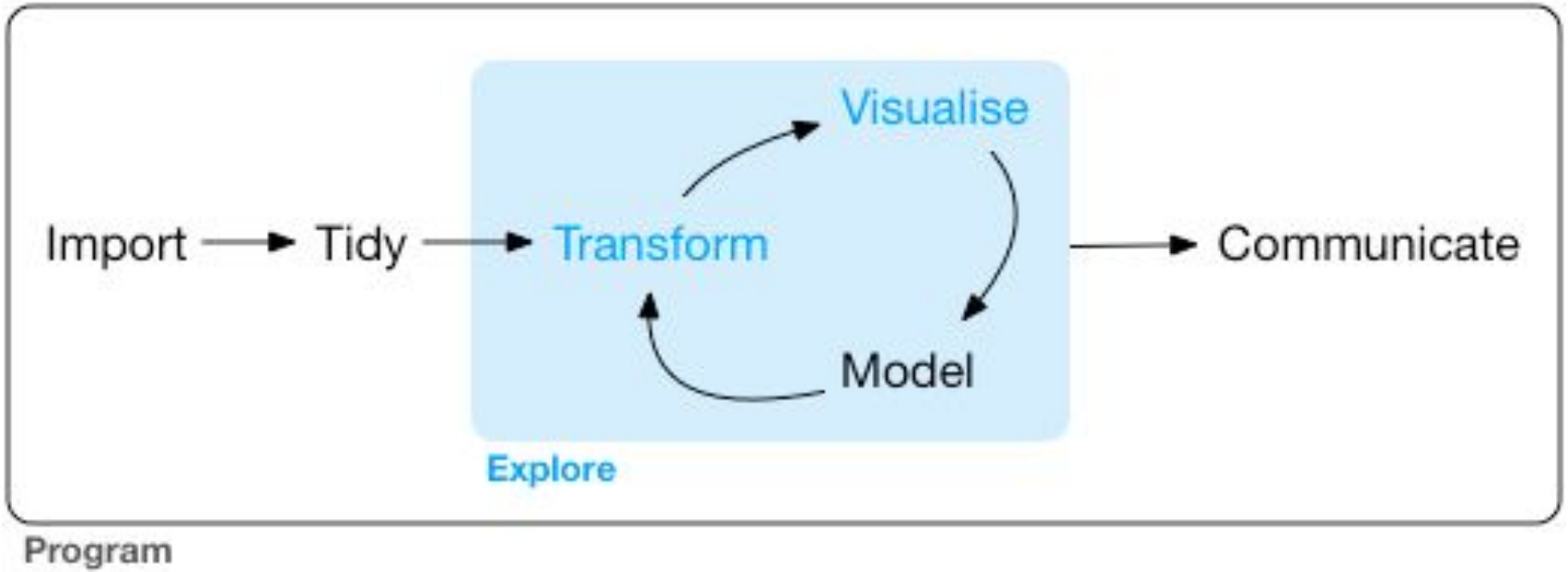




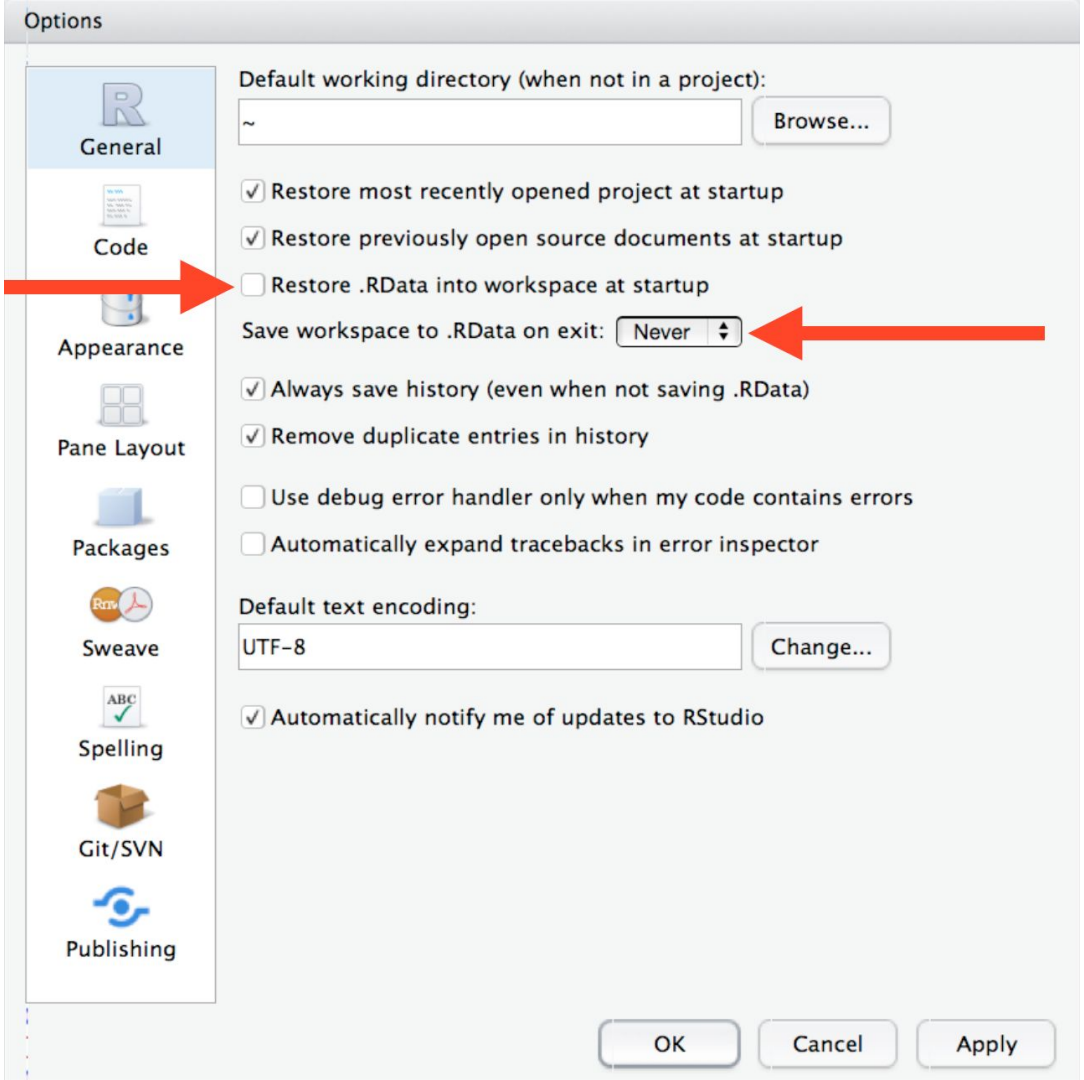
More info: https://www.r-project.org/about.html

R Studio


D:/Documents/Carpentries/data-carpentry - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
data-carpentry
script.R
Source on Save
Run
Source
1
1:1
(Top Level)
R Script
1.
Source
Environment History Connections
Import Dataset
List
Global Environment
Environment is empty
3.
Environment/History
Console D:/Documents/Carpentries/data-carpentry/
R version 3.5.3 (2019-03-11) -- "Great Truth"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
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.
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.
>
2.
Console
Files Plots Packages Help Viewer
New Folder
Delete
Rename
More
D: Documents Carpentries data-carpentry
Name
Size
Modified
..
data-carpentry.Rproj
218 B
Mar 24, 2019, 5:1
script.R
0 B
Mar 24, 2019, 5:1
4. Files and
more

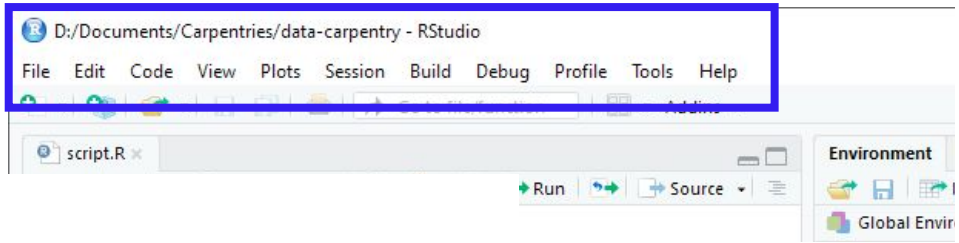


https://r4ds.had.co.nz/explore-intro.html


Options
General
Code
Appearance
Pane Layout
Packages
Sweave
Spelling
Git/SVN
Publishing
Default working directory (when not in a project):
~
Browse...
Restore most recently opened project at startup
Restore previously open source documents at startup
Restore .RData into workspace at startup
Save workspace to .RData on exit:
Never
Always save history (even when not saving .RData)
Remove duplicate entries in history
Use debug error handler only when my code contains errors
Automatically expand tracebacks in error inspector
Default text encoding:
UTF-8
Change...
Automatically notify me of updates to RStudio
OK
Cancel
Apply

Project / Working Directory

- Working directories



Example:
working directory
/users/yvette/documents/r/resbaz2020

absolute path
/users/yvette/documents/r/resbaz2020/scripts/rscript.R

relative path
- scripts/rscript.R

s or as a whole.

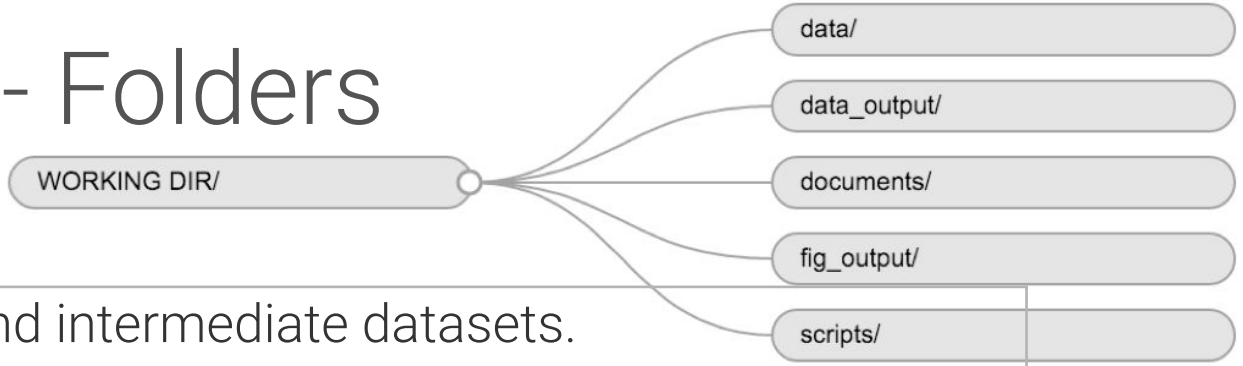
here.

aths.

one place, cleanly
e working on).

https://r4ds.had.co.nz/workflow-projects.html

Tidy Data Tip - Folders



data/	raw data and intermediate datasets. * keep a copy of your raw data accessible and do as much of your data cleanup and preprocessing programmatically (i.e., with scripts, rather than manually) as possible.
data_output/	modified versions of the datasets (e.g. cleaned data).
documents/	outlines, drafts, and other text.
fig_output/	store the graphics that are generated by your scripts.
scripts/	A place to keep your R scripts.

Demo

- RStudio
- write some code
- object variables
- functions
- good practice

Style Guide - Data Management

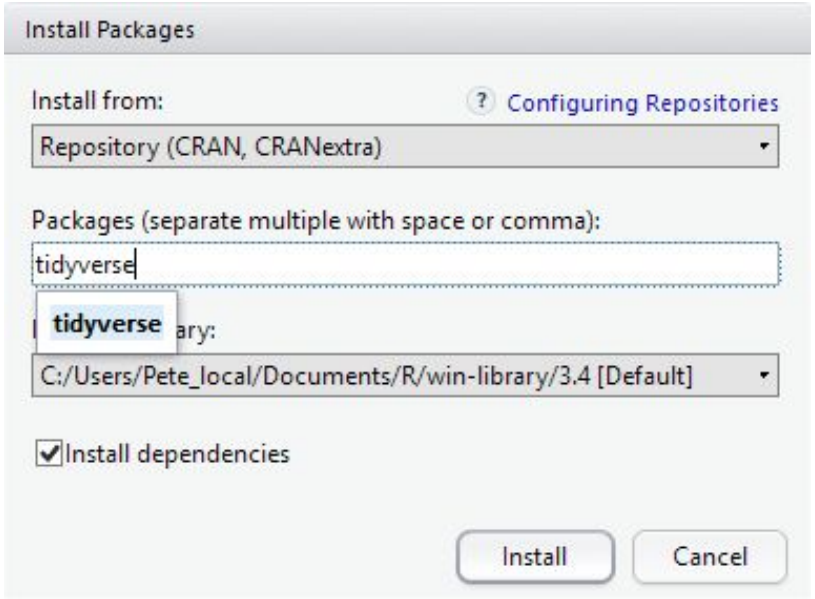
- Object variable and function names should be lowercase.
  - object variable names should be nouns
  - function names should be verbs
- Use an underscore (_) to separate words within a name (no spaces)
- Place spaces around all operators (=, +, -, <-, etc.), and when using = in function calls.
- Indenting your code, use two spaces.
- Comments - make notes for yourself and others
  - begin each line of code with the comment symbol and a single space (#)
  - Use commented lines of - and = to break up your file into easily readable chunks.

```
# Load data ----------------------------
X <- 1 # assign a reference to the value of 1 to the object variable x
# Plot data ----------------------------
```

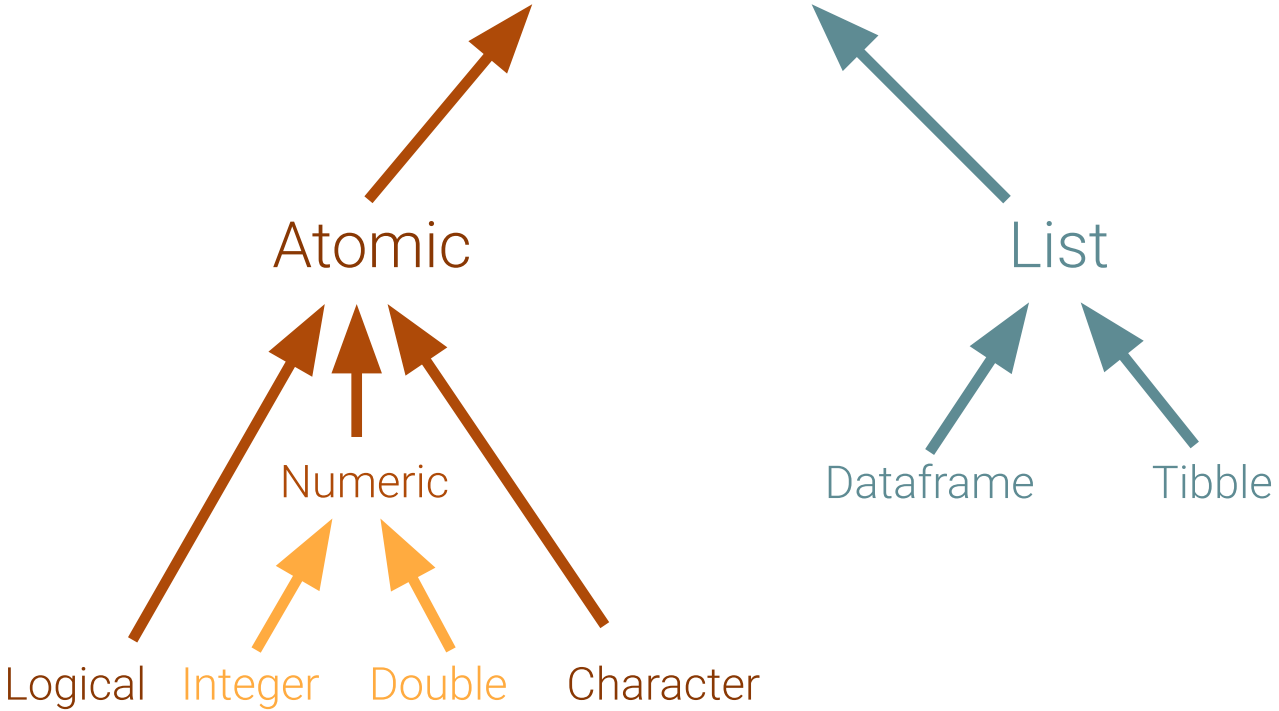
http://adv-r.had.co.nz/Style.html

Packages

- Install once (downloads to computer)
- Need to load into R memory every time
- Some useful packages
  - dplyr for manipulating data.
  - readr for reading .csv and .fwf file.
  - readxl for reading .xls / .xlsx files.
  - tidyr for tidying data.
  - stringr for working with strings.
  - lubridate for working with date/times.
  - ggplot2 for visualising data.
  - httr for talking to web APIs.
  - rvest for scraping websites.
  - rmarkdown notebook interface code and text.
  - pacman R package management tool.
  - formatR cleaning up code formatting

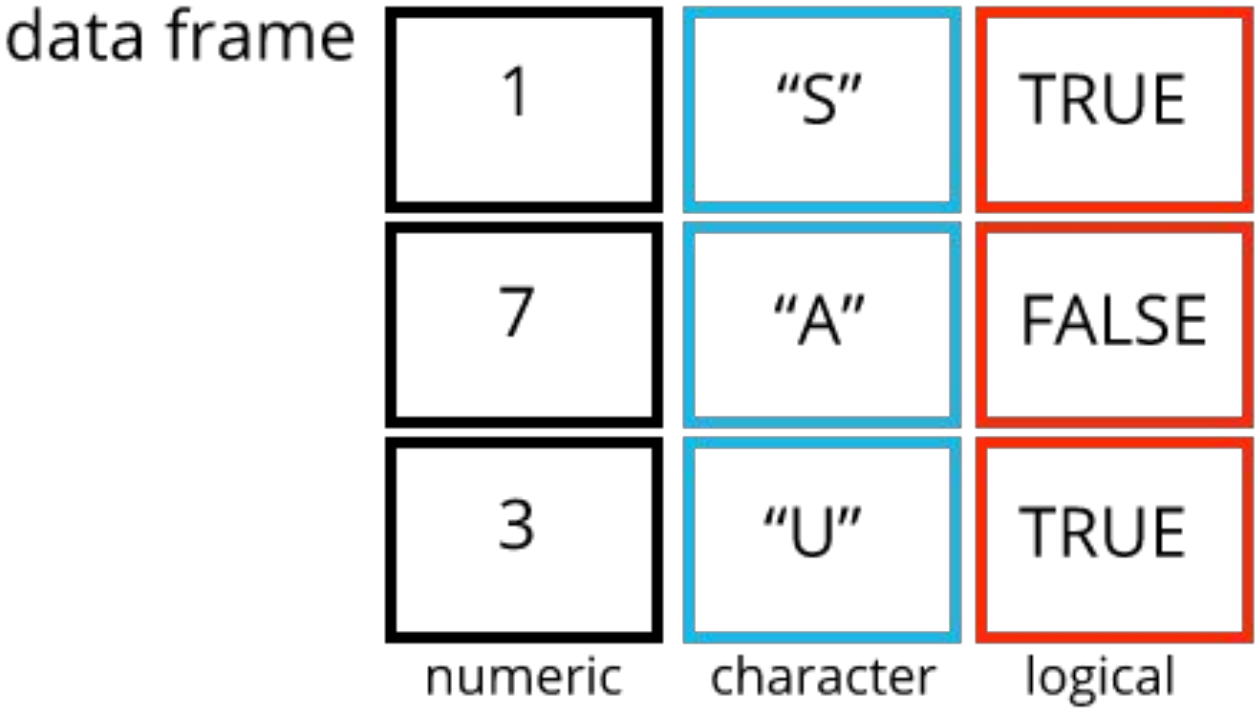


DataTypes: Vectors



https://adv-r.hadley.nz/vectors-chap.html

Data Frames + Tibbles



More information: https://adv-r.hadley.nz/vectors-chap.html#tibble

Subsetting

hh_members <- c(3, 7, 10, 6)

[ ] to access individual values

e.g hh_members[2] *# returns 7*

- [c (…)] to access sets of data

  e.g hh_members[c(3, 1)] *# returns 10 3*

- logical operators and logical vectors to get subsets.

  hh_members >5 *# returns FALSE TRUE TRUE TRUE*

  E.g. hh_members[hh_members > 5] *# returns 7 10 6*

Demo

- Packages
- vectors
- subsetting

SAFI Dataset

- Farming and irrigation methods in Africa
- Interview data Nov 2016-June 2017
- This demonstration a subset of a cleaned version (in OpenRefine) of the dataset
- Rows respondents

Available on FigShare: https://figshare.com/articles/SAFI_Survey_Results/6262019

CITATION: Woodhouse, Philip; Veldwisch, Gert Jan; Brockington, Daniel; Komakech, Hans C.; Manjichi, Angela; Venot, Jean-Philippe (2018): SAFI Survey Results. doi:10.6084/m9.figshare.6262019.v1

Image from: https://degrp.odi.org/project/studying-african-farmer-led-irrigation-safi/

Read/Import data into R

read_csv() function (part of readr package)

```
> library(tidyverse)
> interviews <- read_csv ("data/SAFI_clean.csv", na = “NULL" )
```

Into the Object called interview ← ASSIGN ← what you read from the csv file ← called SAFI_clean.csv located in the data folder ← as you do this change all null entries to NA

Note there’s also a function called read.csv

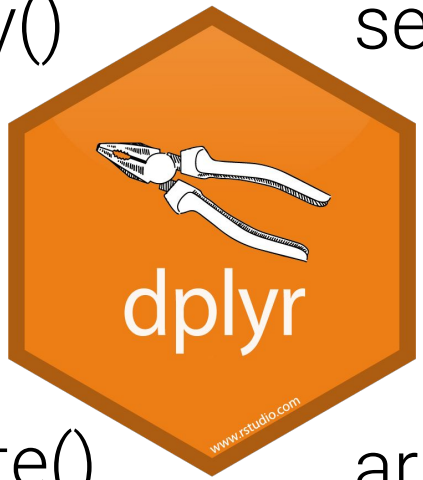
Manipulating Data Frames and Tibbles

dplyr package functions helps you manipulate dataframes

group_by()

select()

summarise()



filter()

mutate()

arrange()

Manipulating Data Frames and Tibbles

dplyr package functions helps you manipulate dataframes

- select() to subset columns

key_ID	village	no_meals	rooms	no_membrs
1	God	2	1	3
2	God	2	1	7
3	Chirodzo	2	3	12
4	Chirodzo	3	1	8
5	Chirodzo	3	5	12
6	God	3	1	3

key_ID	village	no_membrs
1	God	3
2	God	7
3	Chirodzo	12
4	Chirodzo	8
5	Chirodzo	12
6	God	3

select(interviews, key_ID, village, no_members)

Manipulating Data Frames and Tibbles

dplyr package functions helps you manipulate dataframes

- filter() to subset rows
- select() to subset columns

key_ID	village	no_meals	rooms	no_membrs
1	God	2	1	3
2	God	2	1	7
3	Chirodzo	2	3	12
4	Chirodzo	3	1	8
5	Chirodzo	3	5	12
6	God	3	1	3

filter(int

key_ID	village	no_meals	rooms	no_membrs
3	Chirodzo	2	3	12
4	Chirodzo	3	1	8
5	Chirodzo	3	5	12

Manipulating Data Frames and Tibbles

dplyr package functions helps you manipulate dataframes

- filter() to subset rows
- select() to subset columns

key_ID	village	no_meals	rooms	no_membrs
1	God	2	1	3
2	God	2	1	7
3	Chirodzo	2	3	12
4	Chirodzo	3	1	8
5	Chirodzo	3	5	12
6	God	3	1	3

key_ID	village	no_meals	rooms	no_membrs
3	Chirodzo	2	3	12
5	Chirodzo	3	5	12

filter(interviews, village == "Chirodzo", no_membrs > 10)

Manipulating Data Frames and Tibbles

dplyr package functions helps you manipulate dataframes

- arrange () to reorder rows
- filter() to subset rows

key_ID	village	no_meals	rooms	no_membrs
1	God	2	1	3
2	God	2	1	7
3	Chirodzo	2	3	12
4	Chirodzo	3	1	8
5	Chirodzo	3	5	12
6	God	3	1	3

key_ID	village	no_meals	rooms	no_membrs
3	Chirodzo	2	3	12
5	Chirodzo	3	5	12
4	Chirodzo	3	1	8
2	God	2	1	7
1	God	2	1	3
6	God	3	1	3

arrange(interviews, desc(no_members))

Manipulating Data Frames and Tibbles

dplyr package functions helps you manipulate dataframes

- mutate() to create new columns using info from other columns
- arrange () to reorder rows

s

lumns

key_ID	village	no_meals	rooms	no_membrs
1	God	2	1	3
2	God	2	1	7
3	Chirodzo	2	3	12
4	Chirodzo	3	1	8
5	Chirodzo	3	5	12
6	God	3	1	3

key_ID	village	no_meals	rooms	no_membrs	peop_per_room
1	God	2	1	3	3
2	God	2	1	7	1
3	Chirodzo	2	3	12	4
4	Chirodzo	3	1	8	8
5	Chirodzo	3	5	12	2.4
6	God	3	1	3	3

mutate(interviews, peop_per_room = no_membrs / rooms)

Manipulating Data Frames and Tibbles

dplyr package functions helps you manipulate dataframes

- summarize() creates summary statistics
- r… …g info from other columns
- c…
- f…
- s…

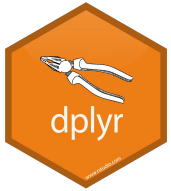
key_ID	village	no_meals	rooms	no_membrs
1	God	2	1	3
2	God	2	1	7
3	Chirodzo	2	3	12
4	Chirodzo	3	1	8
5	Chirodzo	3	5	12
6	God	3	1	3

→

mean_no_membrs
7.5

summarise(interviews, mean_no_membrs = mean(no_membrs))

Combine functions?



dplyr package functions helps you manipulate dataframes

- summarize() with group_by() creates summary statistics
- mutate() to create

key_ID	village	rooms	no_membrs
1	God	1	3
2	God	1	7
3	Chirodzo	3	12
4	Chirodzo	1	8
5	Chirodzo	5	12
6	God	1	3

key_ID	village	rooms	no_membrs
1	God	1	3
2	God	1	7
6	God	1	3

key_ID	village	rooms	no_membrs
3	Chirodzo	3	12
4	Chirodzo	1	8
5	Chirodzo	5	12

mean_	sum	n
	13	3
	32	3

Combine functions (e.g. filter & select)

1. Intermediate steps

```
interviews2 <- filter(interviews, village == "Chirodzo")

interviews_god <- select(interviews2, village:respondent_wall_type)
```

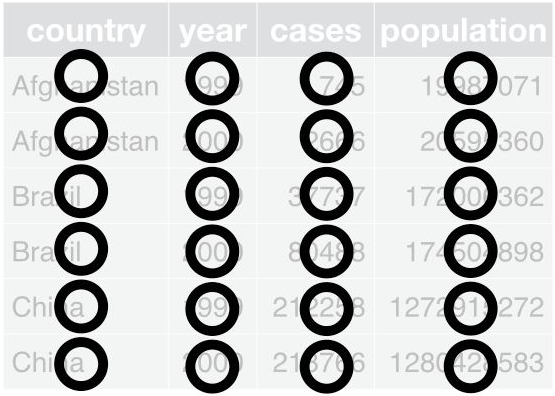
2. Nested functions

```
interviews_god <- select(filter(interviews, village == "Chirodzo"),
   village:respondent_wall_type)
```

3. Pipes %>%

```
interviews %>%
    filter(village == "Chirodzo") %>%
    select(village:respondent_wall_type)
```

Tidy Data Principles



https://r4ds.had.co.nz/tidy-data.html?q=long#longer

pivot_wider()

key_ID	sunbricks	burntbricks	cement	muddaub
1	true	false	false	false
2	false	true	false	false
3	false	false	true	false
4	false	false	false	true
5	false	true	false	false
6	false	false	false	true

Pivoting into a “wider”, tidy form.

pivot_longer()

key_ID	village	wall type	wall_logic
1	God	sunbricks	true
2	God	burntbricks	true
3	Chirodzo	cement	true
4	Chirodzo	muddaub	true
5	Chirodzo	burntbricks	true
6	God	muddaub	true

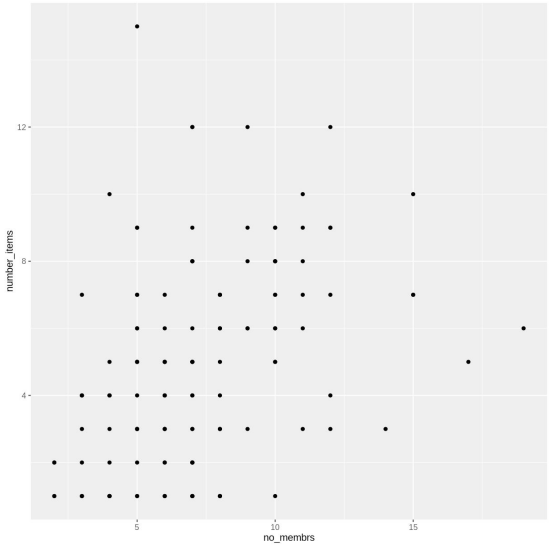
Pivoting into a longer, tidy form.

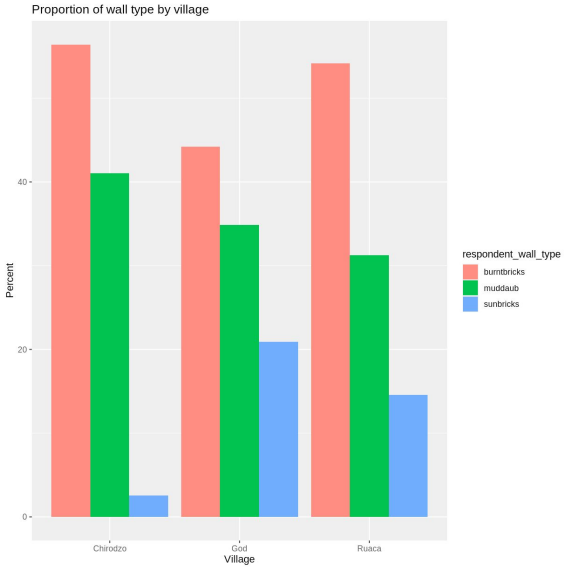
More information: https://r4ds.had.co.nz/tidy-data.html?q=long#longer

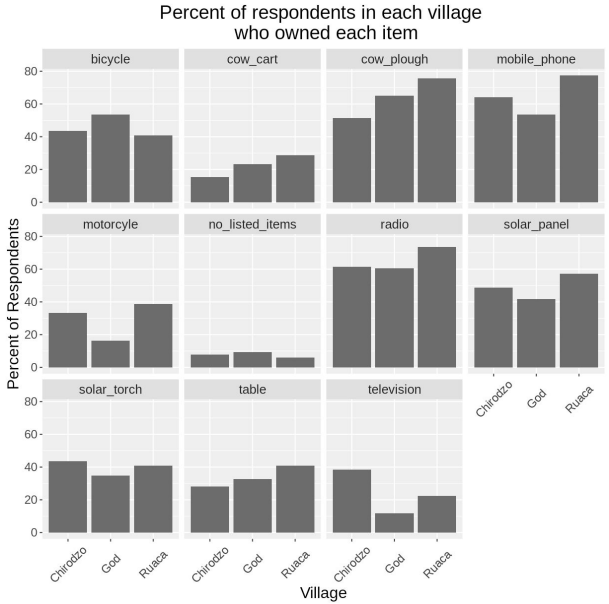
Demo

- data manipulation functions
- pipes
- pivot_wider()
- pivot_longer()

Data Visualisation with ggplot2







https://r4ds.had.co.nz/data-visualisation.html

ggplot example

```
interviews_plotting %>%
    ggplot(aes(x = no_membrs, y = number_items)) +
   geom_point()
```

ggplot()
Create an empty plot using the interviews plotting ile

geom_point ()
add a layer of points (creates a scatterplot)

Set x axis as no. members

Set y axis as number of items

```
<DATA> %>%
    ggplot(aes(<MAPPINGS>)) +
     <GEOM_FUNCTION>()
```

Demo

- ggplot()

Useful Shortcuts

- Cmd/Ctrl Enter to run code
- Cmd/Ctrl + Shift + F10 to restart RStudio.
- Cmd/Ctrl + Shift + S to rerun the current script
- Cmd/Ctrl + L to clear the console
- Ctrl 1 to change to source window focus
- Ctrl 2 to change to console window focus
- Opt - to add an assignment operator
- Opt/Alt + Shift + K - to show all shortcuts

RStudio cheatsheet about the RStudio IDE.

Useful Resources

- Data Carpentry Lesson - R for Social Scientists https://datacarpentry.org/r-socialsci
- “R for Data Science” by Hadley Wickham and Garrett Grolemund r4ds.had.co.nz.
- Advanced R - Hadley Wickham https://adv-r.hadley.nz/