Share your work: Publishing your data analysis with R

Christine Zhang
The Baltimore Sun
@christinezhang | @baltsundata

Slides (& repo) here: https://bit.ly/2HgrsqF

Disclaimers / About me

- This session is about **how to publish** your data analyses with R (as the title says), but it will also touch on the related concept of **what makes a good data analysis workflow** (aka, what will make your life a lot easier)
- It is ***not*** about learning R for data analysis — see the R course track on the NICAR schedule for those classes
- I do not always "practice what I preach," nor am I an expert in all of this. But I have had some experience …

Why share?

https://github.com/baltimore-sun-data

- Builds trust among readers
- Enhances transparency
- Simplifies (self) and peer review
- Promotes community

— adapted from Andrew Ba Tran
https://andrewbtran.github.io/NICAR/2018/workflow/docs/01-workflow_intro.html

Why share?

"Baltimore's last weekend without a shooting was the weekend Freddie Gray was arrested," *Baltimore Sun* (October 18, 2018)

Weekend without shootings

» Baltimore hasn't seen a weekend without a shooting since Freddie Gray was arrested. The last weekend — a period defined as Friday through Sunday — in which Baltimore saw no shootings was the weekend of April 10-12, 2015, meaning there were zero shootings in Baltimore on those days. Gray was arrested April 12 and died from injuries suffered in police custody April 19.

'Bullets don't have a name': 11 shot, three fatally, in Baltimore City on Tuesday »

Daily shooting record

» The 11 shootings Baltimore saw Tuesday was the highest one-day total this year. The last day there were as many shootings was Sept. 24, 2016, when police data show 12 were shot.

Baltimore Sun reporter Justin Fenton contributed to this article.

To learn more about the methodology and review the computer code that generated the analysis, go to www.baltimoresun.com/2018-shootings-analysis.

https://www.baltimoresun.com/news/maryland/crime/bs-md-ci-violence-stats-20181018-story.html

Baltimore shootings analysis

Baltimore Sun analysis

By Christine Zhang and Caroline Pate

The Baltimore Sun conducted an analysis of Baltimore Police Department Part 1 Victim Based Crime Data posted on Open Baltimore on October 17, 2018. The report covers incidents between January 1, 2012, and October 13, 2018.

The analysis provided information for the October 18, 2018 Baltimore Sun story titled "Baltimore's last weekend without a shooting was the weekend Freddie Gray was arrested".

The Sun's findings and analysis are available in the "analysis" notebook in this repository, `shootings_analysis.ipynb` (note: viewable on desktop).

https://twitter.com/baltsundata

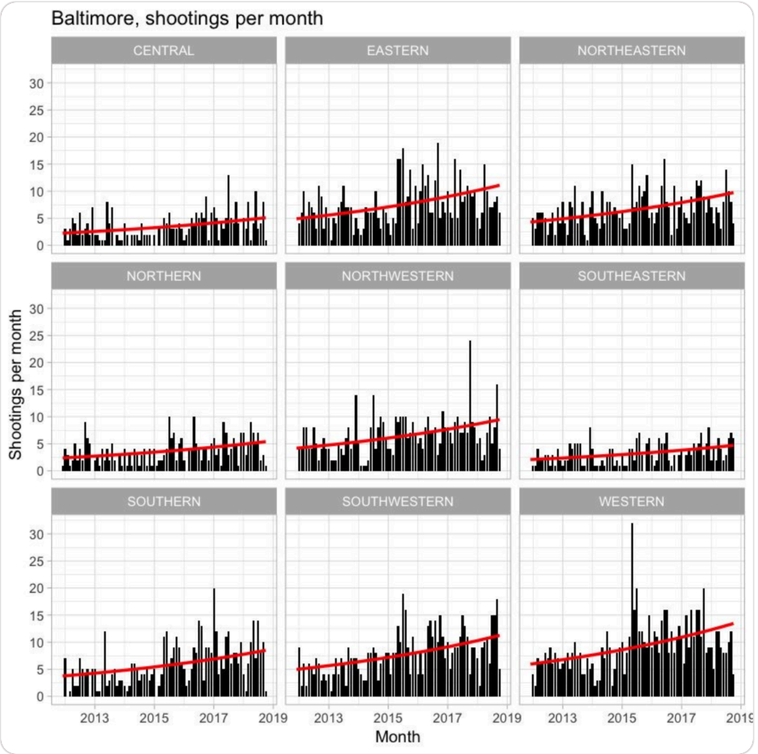
https://github.com/baltimore-sun-data/2018-shootings-analysis

Christopher Peters
@statwonk

Following

Replying to @statwonk @christinezhang and 2 others

I added a view split by district. The Western district has a shooting rate of about 2.6x of the Central district as viewed through this model + data: github.com/baltimore-sun- …



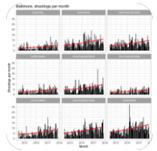
10:41 AM - 21 Oct 2018

Christine Zhang
@christinezhang

This is why we make a lot of our @baltsundata data & code on publicly available on @github.

Great to see civic-minded data scientists like @statwonk engaging with #opendata in this way. There are many angles to explore.

github.com/baltimore-sun-…



Christopher Peters @statwonk
Replying to @statwonk @christinezhang and 2 others
I added a view split by district. The Western district has a shooting rate of about 2.6x of the Central district as viewed through this model + data: github.com/baltimore-sun-…

11:16 AM - 21 Oct 2018 from Baltimore, MD

15 Retweets **36** Likes

3 15 36

https://twitter.com/christinezhang/status/1054028719873028101

0 Open ✓ 2 Closed Author Projects Labels Milestones Reviews Assignee Sort

Measure the shift in shootings that occurred before and after the death of Freddie Gray.
#2 by statwonk was merged on Oct 24, 2018

Add three models to describe the data 4
#1 by statwonk was merged on Oct 23, 2018

Community Contributions

By Christopher Peters aka Statwonk (so far!)

There are many angles to explore with this data, beyond just the ones we looked into for our story.

We made the `community contributions` folder as a place to store others' contributions. Here's what we have so far:

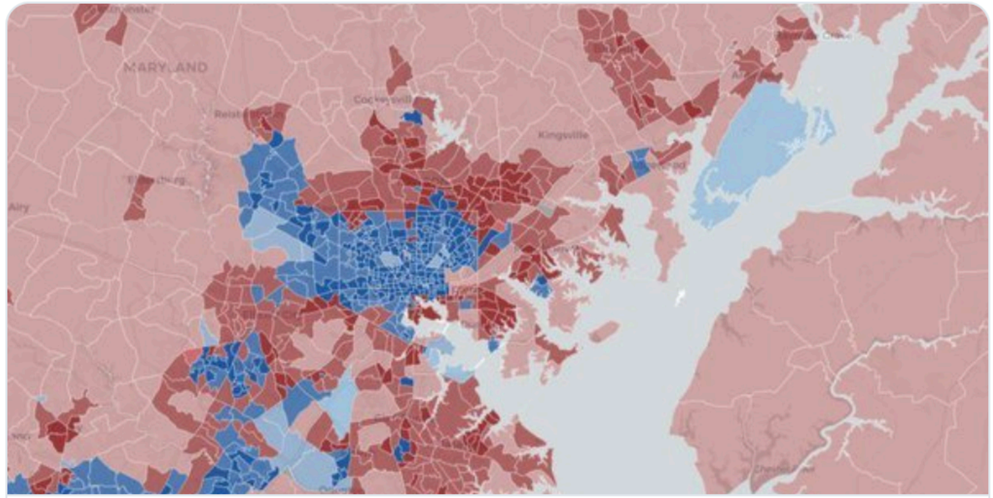
- shootings_analysis_by_district.ipynb: an analysis of the relationship between police district and shootings in Baltimore by Christopher Peters aka Statwonk

Have something to contribute? Send us a pull request or contact us on Twitter @baltsundata or via email (addresses linked above). You can also fork your own copy of this repo to your own account.

https://github.com/baltimore-sun-data/2018-shootings-analysis

Jordan Tessler @JZTessler · 7 Nov 2018

The **Baltimore Sun** has Maryland precinct data and I want it



Maryland governor precinct-by-precinct: Where Hogan and Jealous …

See how your neighborhood voted with this detailed map, which shows ballots cast for governor on Election Da

baltimoresun.com

2 4 18

Frank Hecker
@hecker
Following

After @baltsundata published detailed precinct-level data on the Maryland 2018 gubernatorial race between Larry Hogan and Ben Jealous, I had fun looking at how Hogan's vote share varied based on the number of registered voters per square mile rpubs.com/frankhecker/43…

1:19 AM - 10 Nov 2018

1 Retweet 4 Likes

1 1 4

Balt. Sun Data/Viz @baltsundata · 9 Nov 2018

Replying to @JZTessler

Here it is: twitter.com/baltsundata/st… (or, if you prefer, the raw JS files from @md_sbe are linked to in the repo as well)

Balt. Sun Data/Viz @baltsundata

ɔpy Friday, here's some **DATA** — precinct-level results for Maryland's ernor's race are now available on our GitHub repo. We have also uded GIS shapefiles so you can make your own maps. Check it out 🤓 bsun.md/2RM0vwW

1

Like · Reply · Remove Preview · 16w

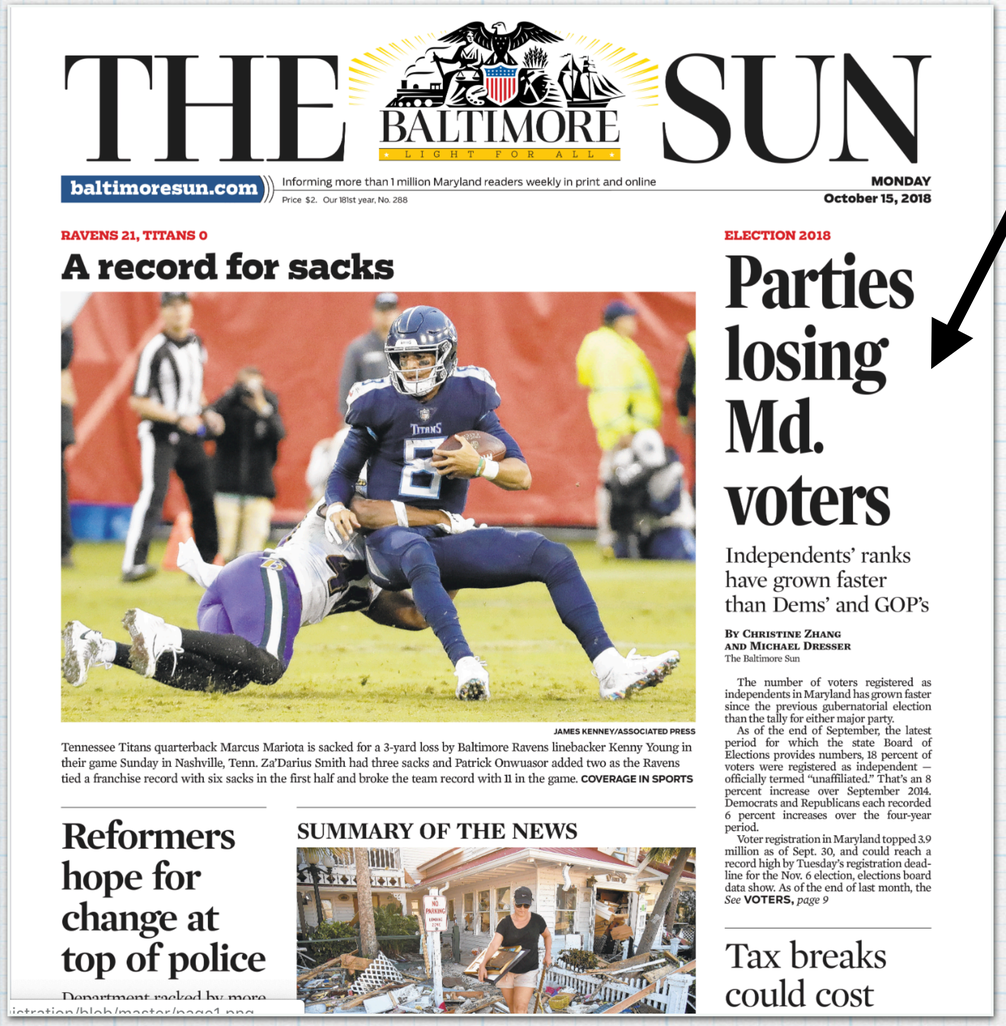
Owen Silverman Andrews Christine Zhang, thank you for engaging so extensively! I'm the son of a data person, not a data person myself, but I love to use it to tell stories and your visualization really did most of the storytelling for me. Thanks for making it!

3

Like · Reply · 16w

https://twitter.com/JZTessler/status/1060223143372034049

https://twitter.com/hecker/status/1061141198482669568



THE BALTIMORE SUN

LIGHT FOR ALL

baltimoresun.com

Informing more than 1 million Maryland readers weekly in print and online

Price $2. Our 181st year, No. 288

MONDAY
October 15, 2018

RAVENS 21, TITANS 0

A record for sacks

JAMES KENNEY/ASSOCIATED PRESS

Tennessee Titans quarterback Marcus Mariota is sacked for a 3-yard loss by Baltimore Ravens linebacker Kenny Young in their game Sunday in Nashville, Tenn. Za'Darius Smith had three sacks and Patrick Onwuasor added two as the Ravens tied a franchise record with six sacks in the first half and broke the team record with 11 in the game. COVERAGE IN SPORTS

Reformers hope for change at top of police

Department racked by more

SUMMARY OF THE NEWS

ELECTION 2018

Parties losing Md. voters

Independents' ranks have grown faster than Dems' and GOP's

By Christine Zhang and Michael Dresser
The Baltimore Sun

The number of voters registered as independents in Maryland has grown faster since the previous gubernatorial election than the tally for either major party.

As of the end of September, the latest period for which the state Board of Elections provides numbers, 18 percent of voters were registered as independent — officially termed "unaffiliated." That's an 8 percent increase over September 2014. Democrats and Republicans each recorded 6 percent increases over the four-year period.

Voter registration in Maryland topped 3.9 million as of Sept. 30, and could reach a record high by Tuesday's registration deadline for the Nov. 6 election, elections board data show. As of the end of last month, the

See VOTERS, page 9

Tax breaks could cost

Balt. Sun Data/Viz @baltsundata · 15 Oct 2018

For our voter registration story, we converted PDF reports to CSVs (h/t @TabulaPDF). Story by @christinezhang and @michaeltdresser: baltimoresun.com/news/maryland/…

Check out our GitHub repo here: github.com/baltimore-sun-… #opendata

baltimore-sun-data/2018-voter-registration
A Baltimore Sun analysis of Maryland's voter registration statistics. - baltimore-sun-data/2018-voter-registration
github.com

1 4 7

Rob Carlson
@vees

Following

Replying to @baltsundata @christinezhang and 2 others

This tweet/article/repo just made me re-subscribe. Keep up the great work.

12:03 PM - 15 Oct 2018

1 Retweet **6** Likes

1 1 6

https://twitter.com/vees/status/1051866074625572865

What I will do today

Using an example data analysis, I will:

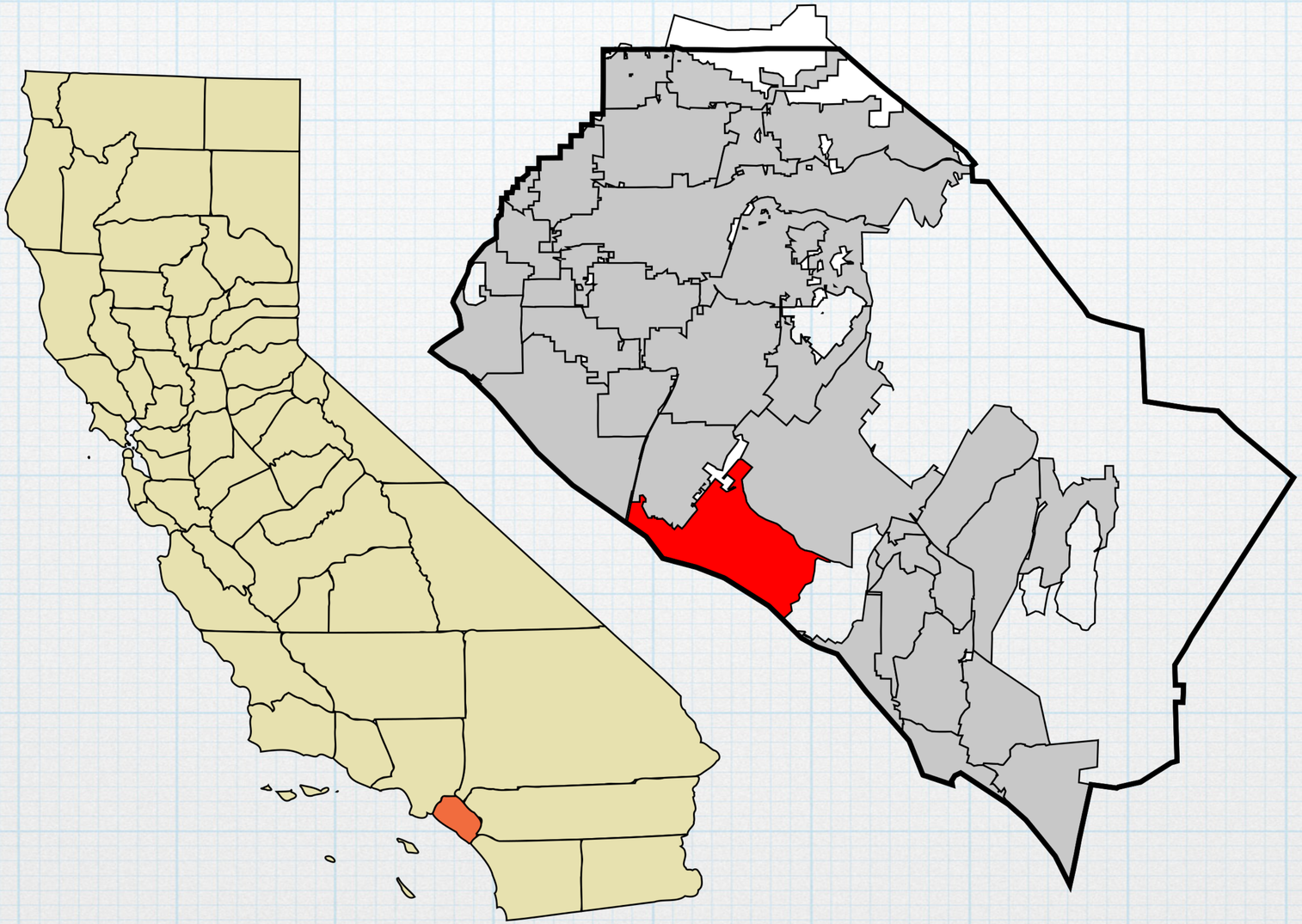
- Show how I organize my data and files prior to sharing

- Show 3 ways of sharing my analysis:
  1. **Straight-up code**
  2. **RMarkdown**
  3. **Jupyter notebook** (if there's time)

  + (hosting on GitHub, time permitting)

- Recap pros and cons of each of the 3 approaches

Newport Beach, CA



By Arkyan - My own work, based on public domain information. Based on similar map concepts by Ixnayonthetimmay, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=2486142

Newport Beach, CA

B03002

HISPANIC OR LATINO ORIGIN BY RACE
Universe: Total population
2013-2017 American Community Survey 5-Year Estimates

Table View

Actions: Modify Table | Add/Remove Geographies | Bookmark/Save | Print | Download | Create a Map

Although the American Community Survey (ACS) produces population, demographic and housing unit estimates, it is the Census Bureau's Population Estimates Program that produces and disseminates the official estir
housing units for states and counties.

Versions of this table are available for the following years:

- 2017
- 2016
- 2015
- 2014
- 2013
- 2012
- 2011
- 2010
- 2009

1 - 21 of 21

	Newport Beach city, California	
	Estimate	Margin of Error
Total:	86,793	+/-53
Not Hispanic or Latino:	79,878	+/-760
White alone	70,471	+/-1,097
Black or African American alone	536	+/-267
American Indian and Alaska Native alone	166	+/-115
Asian alone	6,604	+/-698
Native Hawaiian and Other Pacific Islander alone	93	+/-80
Some other race alone	122	+/-88
Two or more races:	1,886	+/-436
Two races including Some other race	18	+/-27
Two races excluding Some other race, and three or more races	1,868	+/-438
Hispanic or Latino:	6,915	+/-762
White alone	4,447	+/-627
Black or African American alone	11	+/-16
American Indian and Alaska Native alone	38	+/-42
Asian alone	71	+/-59
Native Hawaiian and Other Pacific Islander alone	19	+/-32
Some other race alone	1,575	+/-601
Two or more races:	754	+/-292
Two races including Some other race	441	+/-249
Two races excluding Some other race, and three or more races	313	+/-178

Source: U.S. Census Bureau, 2013-2017 American Community Survey 5-Year Estimates

https://factfinder.census.gov/bkmk/table/1.0/en/ACS/17_5YR/B03002/1600000US0651182

tidycensus 0.8.2 Reference Articles

tidycensus

build passing | CRAN 0.9 | downloads 2487/month

tidycensus is an R package that allows users to interface with the US Census Bureau's decennial Census and five-year American Community APIs and return tidyverse-ready data frames, optionally with simple feature geometry included. Install from CRAN with the following command:

```
install.packages("tidycensus")
```

New: To learn how to use **tidycensus**, take the DataCamp course Analyzing US Census Data in R!

Why tidycensus?

My work heavily involves the use of data from the US Census Bureau, and like many R users, I do most of my work within the **tidyverse**. Beyond this, the **sf** package now allows R users to work with spatial data in an integrated way with **tidyverse** tools, and updates to the **tigris** package provide access to Census boundary data as `sf` objects. Recently, I've found myself writing the same routines over and over to get Census data ready for use with **tidyverse** packages and **sf**. This motivated me to open-source these functions in a package in case other R users find them useful.

tidycensus is designed to help R users get Census data that is pre-prepared for exploration within the **tidyverse**, and optionally spatially with **sf**. To learn more about how the package works, I encourage you to read the following articles:

- Basic usage of **tidycensus**
- Spatial data in **tidycensus**
- Margins of error in the ACS
- Other Census Bureau datasets

Future development

To keep up with on-going development of **tidycensus** and get even more examples of how to use the package, subscribe to my email list below (no spam, I promise!). You'll also get updates on the development of my upcoming book with CRC Press, *Analyzing the US Census with R*.

Links

Download from CRAN at https://cloud.r-project.org/package=tidycensus

Browse source code at https://github.com/walkerke/tidycensus

Report a bug at https://github.com/walkerke/tidycensus/issues

License

MIT + file LICENSE

Developers

Kyle Walker
Author, maintainer

All authors...

https://walkerke.github.io/tidycensus/articles/basic-usage.html

```r
library('tidycensus')
library('tidyverse')

## More on tidycensus here: https://walkerke.github.io/tidycensus/articles/basic-usage.html

# census_api_key("YOUR API KEY GOES HERE")

ca <- get_acs("place",
              table = "B03002",
              summary_var = "B03002_001",
              state = "CA",
              year = 2017
              )

newport.beach <- ca %>%
  filter(NAME == 'Newport Beach city, California')

write_csv(newport.beach, './input/newport_beach.csv')
```

documentation/get_census_data.R

Let’s talk file organization

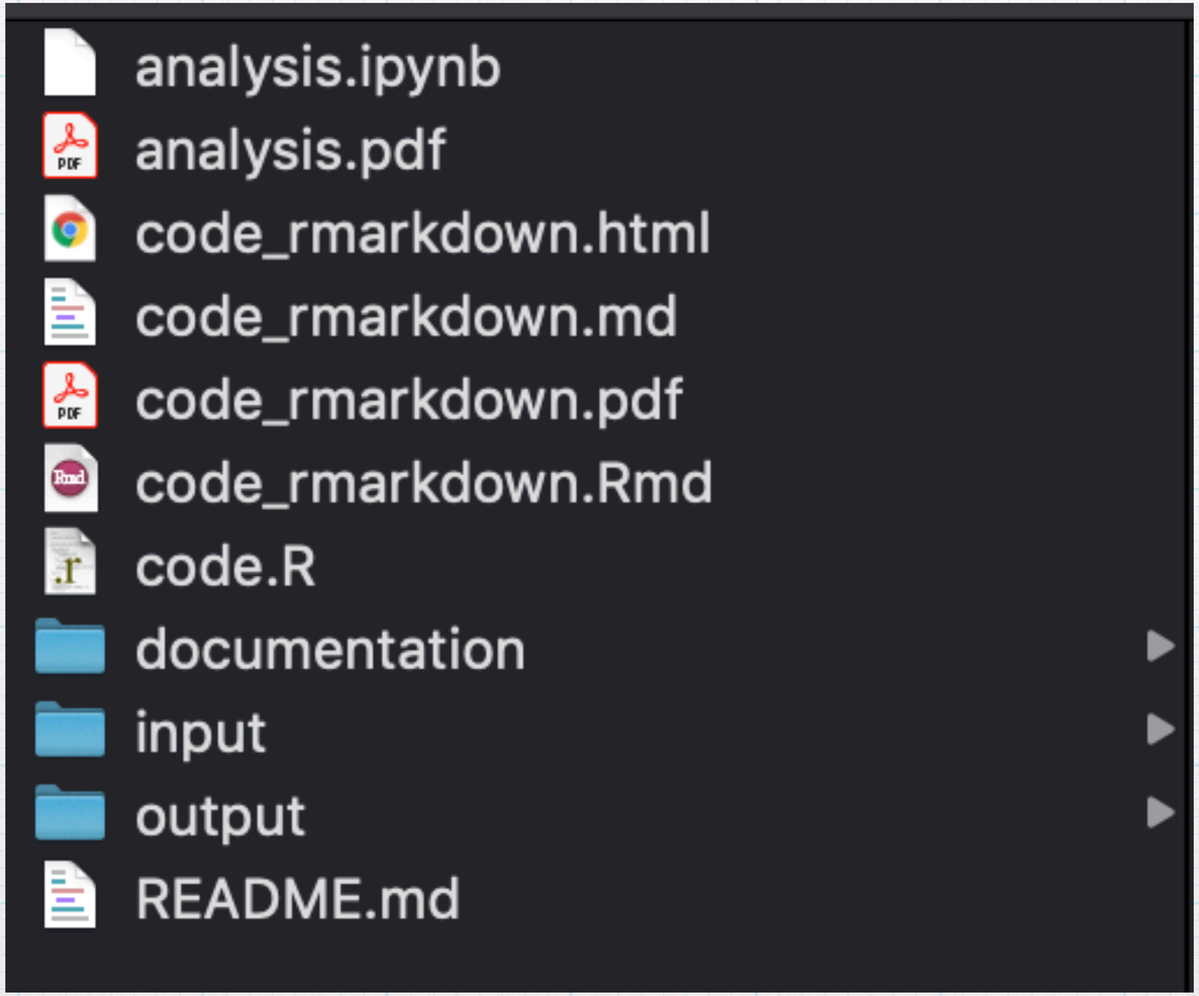
There are different opinions on how to organize your files. I’m only going to go through one way.

Here are some **general principles** to remember:

- Create a separate folder for each project

- Inside this folder should be:
  - Raw data (aka “input” data) — subfolder
  - Cleaned data (aka “output” data) — subfolder
  - Your processing and analysis code — either subfolder or in main folder
  - Notes or documentation — either subfolder or in main folder

Let’s talk file organization: example



What I have: the “**input**” data

GEOID	NAME	variable	estimate	moe	summary_est	summary_moe
651182	Newport Beach city, California	B03002_001	86793	53	86793	53
651182	Newport Beach city, California	B03002_002	79878	760	86793	53
651182	Newport Beach city, California	B03002_003	70471	1097	86793	53
651182	Newport Beach city, California	B03002_004	536	267	86793	53
651182	Newport Beach city, California	B03002_005	166	115	86793	53
651182	Newport Beach city, California	B03002_006	6604	698	86793	53
651182	Newport Beach city, California	B03002_007	93	80	86793	53
651182	Newport Beach city, California	B03002_008	122	88	86793	53
651182	Newport Beach city, California	B03002_009	1886	436	86793	53
651182	Newport Beach city, California	B03002_010	18	27	86793	53
651182	Newport Beach city, California	B03002_011	1868	438	86793	53
651182	Newport Beach city, California	B03002_012	6915	762	86793	53
651182	Newport Beach city, California	B03002_013	4447	627	86793	53
651182	Newport Beach city, California	B03002_014	11	16	86793	53
651182	Newport Beach city, California	B03002_015	38	42	86793	53
651182	Newport Beach city, California	B03002_016	71	59	86793	53
651182	Newport Beach city, California	B03002_017	19	32	86793	53
651182	Newport Beach city, California	B03002_018	1575	601	86793	53
651182	Newport Beach city, California	B03002_019	754	292	86793	53
651182	Newport Beach city, California	B03002_020	441	249	86793	53
651182	Newport Beach city, California	B03002_021	313	178	86793	53

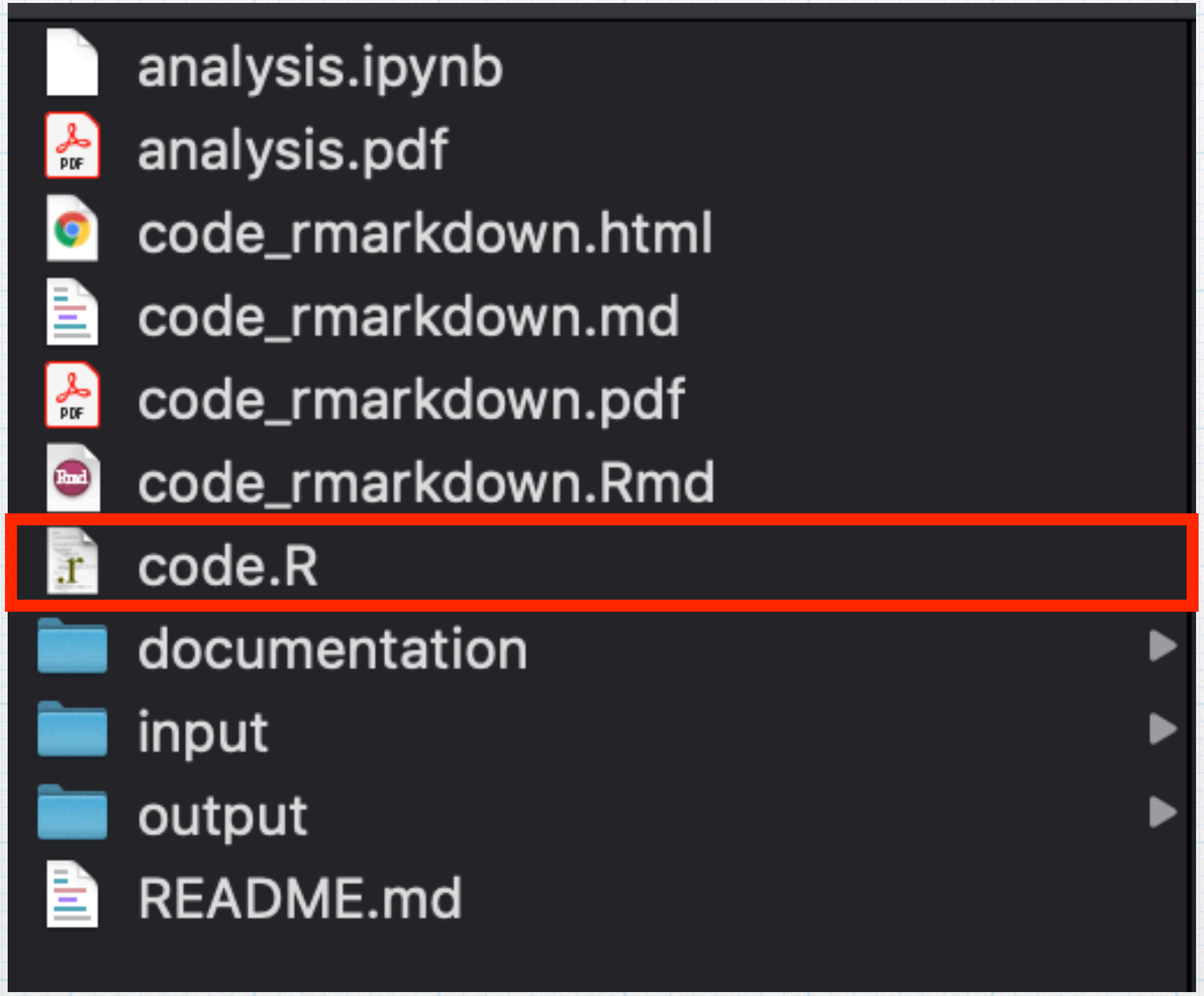
input/newport_beach.csv

What I want, post-cleaning: the "**output**" data

	A	B	C	D	E	F
1	GEOID	NAME	cat	summary_est	estimate	perc
2	651182	Newport Beach city, California	white	86793	70471	81.1943359
3	651182	Newport Beach city, California	hisp	86793	6915	7.96723238
4	651182	Newport Beach city, California	asian	86793	6604	7.60890855
5	651182	Newport Beach city, California	other	86793	2267	2.61196179
6	651182	Newport Beach city, California	black	86793	536	0.61756132
7						

output/newport_beach_cleaned.csv

Straight-up R code



Straight-up R code

```r
# Load libraries and read in data

library('tidyverse')

newport.beach <- read_csv('input/newport_beach.csv')

# v17 <- load_variables(2017, "acs5", cache = TRUE) # view variable names

# Create race/ethnicity categories

newport.beach <- newport.beach %>% select(-summary_moe) %>%
  mutate(cat = case_when(
    variable == 'B03002_003' ~ "white",
    variable == 'B03002_004' ~ "black",
    variable == 'B03002_005' | variable == 'B03002_007' | variable == 'B03002_008' | variable == 'B03002_009' ~ "other",
    variable == 'B03002_006' ~ "asian",
    variable == 'B03002_012' ~ "hisp"))

# Filter out unwanted categories and summarize the data by the newly-created categories

newport.beach <- newport.beach %>%
  filter(!is.na(cat)) %>%
  group_by(GEOID, NAME, cat, summary_est) %>%
  summarise(estimate = sum(estimate))

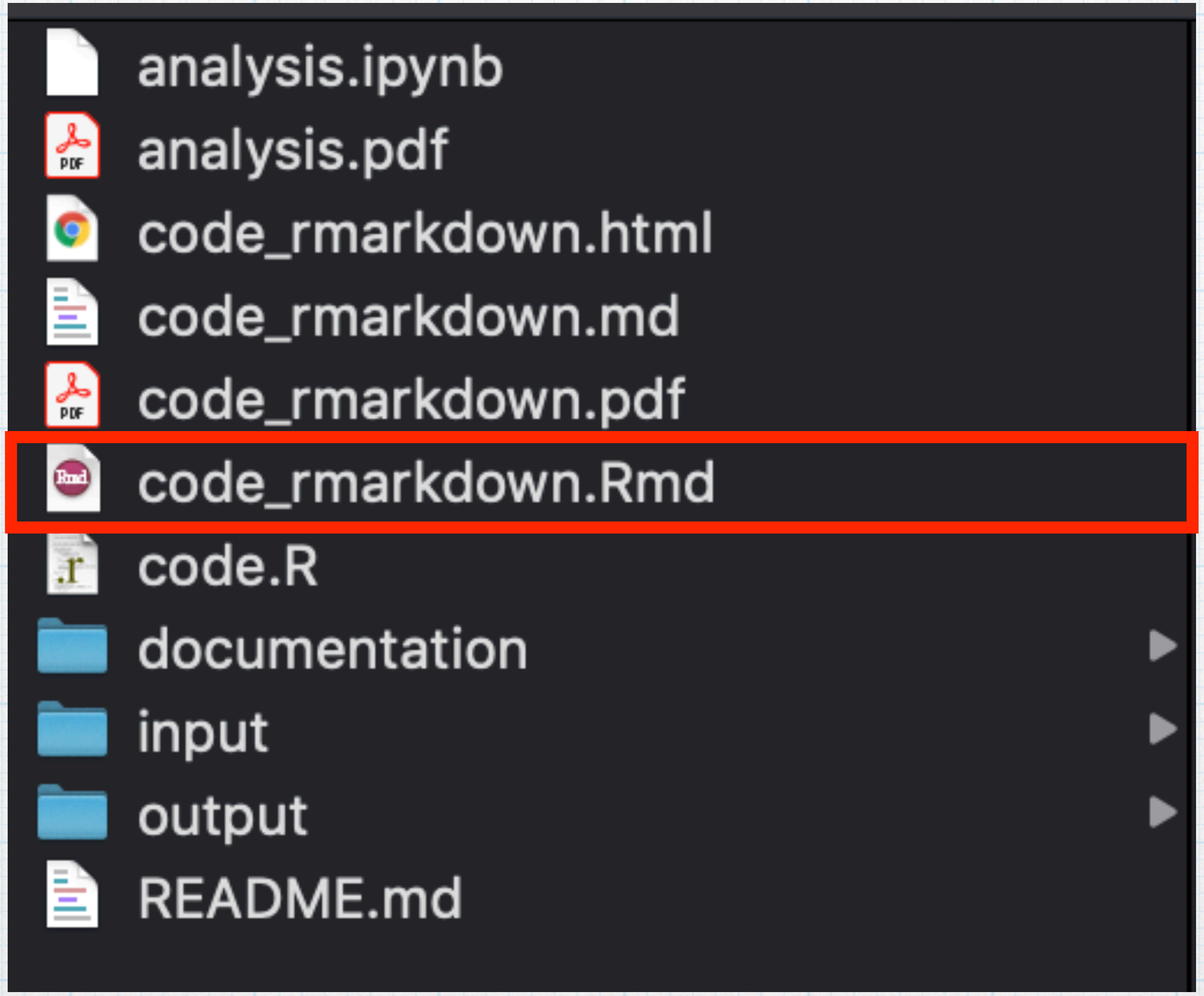
# Calculate each category's percentage of total population and sort categories by percentage, from highest to lowest
newport.beach <- newport.beach %>%
  mutate(perc = estimate/summary_est * 100) %>%
  arrange(desc(perc))

newport.beach

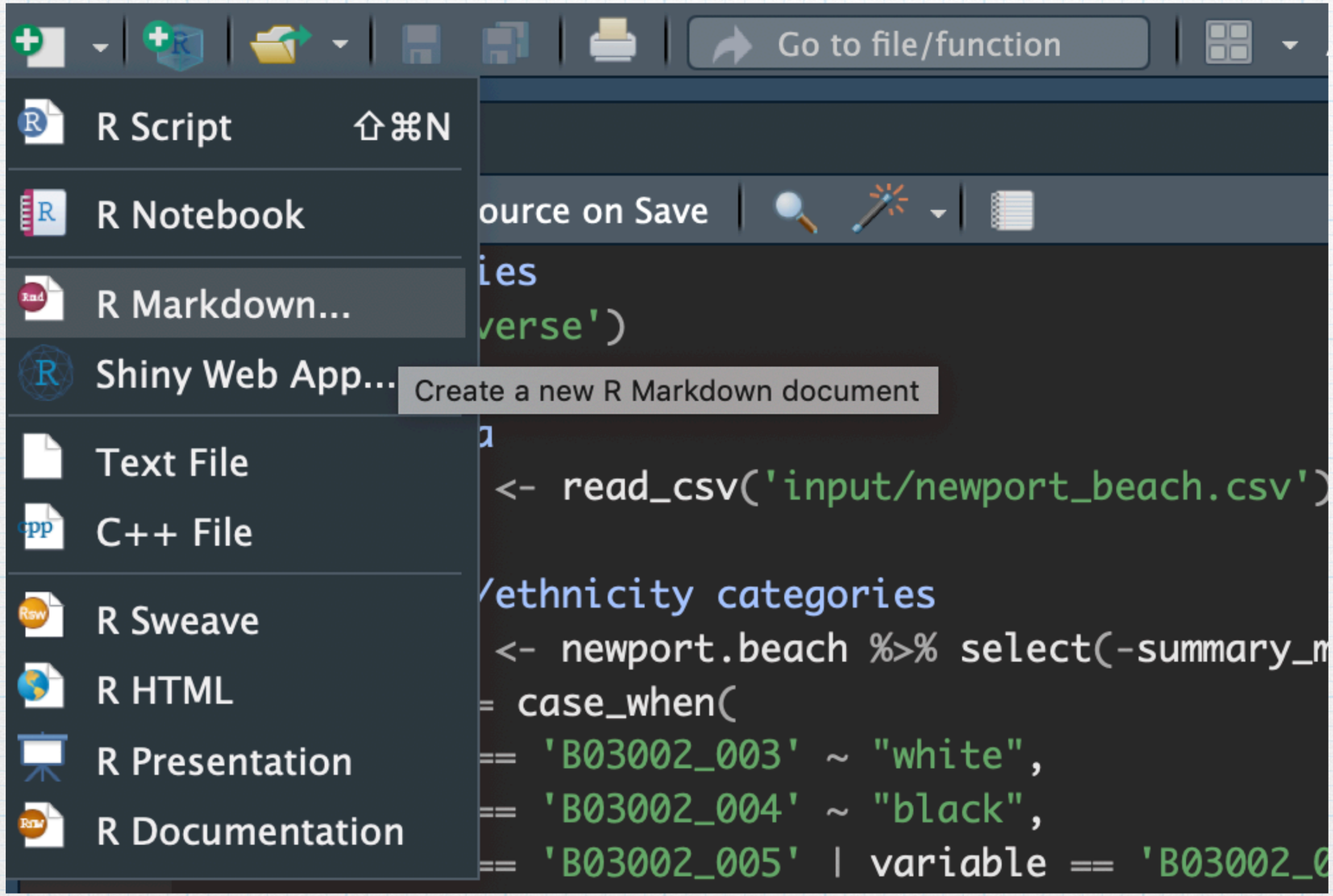
# Write to a CSV
write_csv(newport.beach, 'output/newport_beach_cleaned.csv')
```

code.R

R Markdown


analysis.ipynb
analysis.pdf
code_rmarkdown.html
code_rmarkdown.md
code_rmarkdown.pdf
code_rmarkdown.Rmd
code.R
documentation
input
output
README.md

RMarkdown



code_RMarkdown.Rmd

RMarkdown

```
---
title: "Newport Beach Analysis"
author: "Christine Zhang"
date: "3/6/2019"
output:
  html_document: default
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Load libraries and read in data

The data are from the Census Bureau's 2013-2017 American Community Survey (Five-Year Estimates), table B03002, [HISPANIC OR LATINO ORIGIN BY RACE](https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?src=bkmk). The numbers represent estimates over the five year period.

```{r loading, warning = FALSE, message = FALSE}
library('tidyverse')
library('knitr')
newport.beach <- read_csv('input/newport_beach.csv')
```

Create race/ethnicity categories

Rename the race/ethnicity variables we want to keep into more intuitive names; e.g., "white" for non-Hispanic white, "black" for non-Hispanic black, etc. The "other" category includes non-Hispanic American Indian and Alaskan Native, Native Hawaiian and Other Pacific Islander, Some other race alone, and Two or more races.
```

code_rmarkdown.Rmd

RMarkdown structure

YAML HEADER

```
---
title: "Newport Beach Analysis"
author: "Christine Zhang"
date: "3/6/2019"
output:
  html_document: default
---
```

OPTIONS

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

TEXT

```
### Load libraries and read in data

The data are from the Census Bureau's 2013-2017 Americ
LATINO ORIGIN BY RACE](https://factfinder.census.gov/f
represent estimates over the five year period.
```

CODE CHUNK

```{r loading, warning = FALSE, message = FALSE}
library('tidyverse')
library('knitr')
newport.beach <- read_csv('input/newport_beach.csv')
```

TEXT

```
### Create race/ethnicity categories

Rename the race/ethnicity variables we want to keep in
"black" for non-Hispanic black, etc. The "other" categ
Hawaiian and Other Pacific Islander, Some other race a
```

Via Andrew Ba Tran: https://andrewbtran.github.io/NICAR/2018/workflow/docs/02-rmarkdown.html

R Markdown Cheat Sheet

learn more at rmarkdown.rstudio.com

rmarkdown 0.2.50 Updated: 8/14

RStudio

1. Workflow

R Markdown is a format for writing reproducible, dynamic reports with R. Use it to embed R code and results into slideshows, pdfs, html documents, Word files and more. To make a report:

i. Open - Open a file that uses the .Rmd extension.

ii. Write - Write content with the easy to use R Markdown syntax

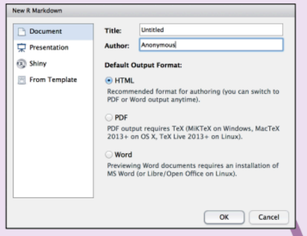
iii. Embed - Embed R code that creates output to include in the report

iv. Render - Replace R code with its output and transform the report into a slideshow, pdf, html or ms Word file.

2. Open File

Start by saving a text file with the extension .Rmd, or open an RStudio Rmd template

- In the menu bar, click **File ▸ New File ▸ R Markdown...**
- A window will open. Select the class of output you would like to make with your .Rmd file
- Select the specific type of output to make with the radio buttons (you can change this later)
- Click OK



3. Markdown

Next, write your report in plain text. Use markdown syntax to describe how to format text in the final report.

syntax

```
Plain text
End a line with two spaces to start a new paragraph.
*italics* and _italics_
**bold** and __bold__
superscript^2^
~~strikethrough~~
[link](www.rstudio.com)

# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

###### Header 6

endash: --
emdash: ---
ellipsis: ...
inline equation: $A = \pi*r^{2}$
image: ![](path/to/smallorb.png)

horizontal rule (or slide break):

***

> block quote

* unordered list
* item 2
    + sub-item 1
    + sub-item 2

1. ordered list
2. item 2
    + sub-item 1
    + sub-item 2

Table Header   | Second Header
-------------- | -------------
Table Cell     | Cell 2
Cell 3         | Cell 4
```

becomes

Plain text
End a line with two spaces to start a new paragraph.
italics and *italics*
bold and **bold**
superscript^2
~~strikethrough~~
link

Header 1

Header 2

Header 3

Header 4

Header 5

Header 6

endash: –
emdash: —
ellipsis: …
inline equation: $A = \pi * r^2$
image:

horizontal rule (or slide break):

> block quote

- unordered list
- item 2
    - sub-item 1
    - sub-item 2

1. ordered list
2. item 2
    - sub-item 1
    - sub-item 2

Table Header	Second Header
Table Cell	Cell 2
Cell 3	Cell 4

4. Choose Output

Write a YAML header that explains what type of document to build from your R Markdown file.

YAML

A YAML header is a set of key: value pairs at the start of your file. Begin and end the header with a line of three dashes (- - -)

```
---
title: "Untitled"
author: "Anonymous"
output: html_document
---

This is the start of my
report. The above is metadata
saved in a YAML header.
```

The RStudio template writes the YAML header for you

The output value determines which type of file R will build from your .Rmd file (in Step 6)

- **output: html_document** ····· html file (web page)
- **output: pdf_document** ····· pdf document
- **output: word_document** ····· Microsoft Word .docx
- **output: beamer_presentation** ····· beamer slideshow (pdf)
- **output: ioslides_presentation** ····· ioslides slideshow (html)

RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1212 • rstudio.com

https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf

code.Rmd

Knit

Knit the current docume

```
1 ---
2 title: "Newport Beach Analysis
3 author: "Christine Zhang"
4 date: "3/6/2019"
5 output:
```

code_rmarkdown.html

code_rmarkdown.Rmd

“Knitting” creates an html file

code_rmarkdown.html | Open in Browser | Find | Publish

Newport Beach Analysis

Christine Zhang

3/6/2019

Load libraries and read in data

The data are from the Census Bureau’s 2013-2017 American Community Survey (Five-Year Estimates), table B03002, HISPANIC OR LATINO ORIGIN BY RACE. The numbers represent estimates over the five year period.

```
library('tidyverse')
library('knitr')
newport.beach <- read_csv('input/newport_beach.csv')
```

Create race/ethnicity categories

Rename the race/ethnicity variables we want to keep into more intuitive names; e.g., “white” for non-Hispanic white, “black” for non-Hispanic black, etc. The “other” category includes non-Hispanic American Indian and Alaskan Native, Native Hawaiian and Other Pacific Islander, Some other race alone, and Two or more races.

```
# v17 <- load_variables(2017, "acs5", cache = TRUE) # view variable names

newport.beach <- newport.beach %>% select(-summary_moe) %>%
  mutate(cat = case_when(
    variable == 'B03002_003' ~ "white",
    variable == 'B03002_004' ~ "black",
    variable == 'B03002_005' | variable == 'B03002_007' | variable == 'B03002_008' | variable == 'B03002_009
' ~ "other",
    variable == 'B03002_006' ~ "asian",
    variable == 'B03002_012' ~ "hisp"))
```

Filter out unwanted categories and summarize the data by the newly-created categories

```
newport.beach <- newport.beach %>%
```

code_rmarkdown.html | Open in Browser | Find | Publish

Newport Beach Analysis

Christine Zhang

3/6/2019

Load libraries and read in data

The data are from the Census Bureau's 2013-2017 American Community Survey (Five-Year Estimates), table B03002 ORIGIN BY RACE. The numbers represent estimates over the five year period.

```
library('tidyverse')
library('knitr')
newport.beach <- read_csv('input/newport_beach.csv')
```

Create race/ethnicity categories

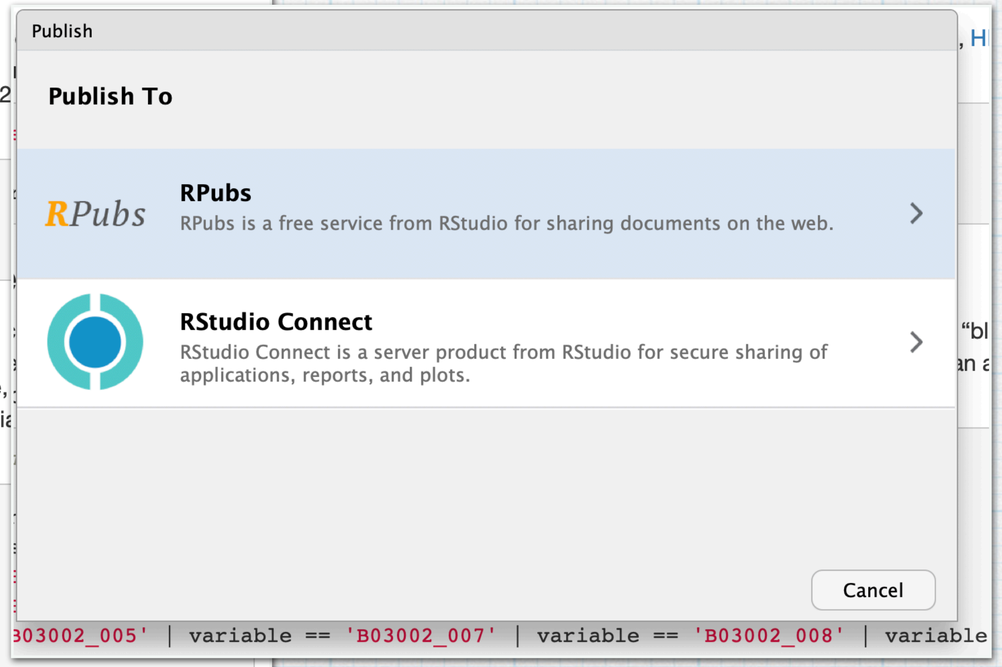
Rename the race/ethnicity variables we want to keep into more intuitive names; e.g., "white" for non-Hispanic white, Hispanic black, etc. The "other" category includes non-Hispanic American Indian and Alaskan Native, Native Hawaiia Islander, Some other race alone, and Two or more races.

```
# v17 <- load_variables(2017, "acs5", cache = TRUE) # view variable names

newport.beach <- newport.beach %>% select(-summary_moe) %>%
  mutate(cat = case_when(
    variable == 'B03002_003' ~ "white",
    variable == 'B03002_004' ~ "black",
    variable == 'B03002_005' | variable == 'B03002_007' | variable == 'B03002_008' | variable == 'B03002_009
' ~ "other",
    variable == 'B03002_006' ~ "asian",
    variable == 'B03002_012' ~ "hisp"))
```

Filter out unwanted categories and summarize the data by the newly-created categories

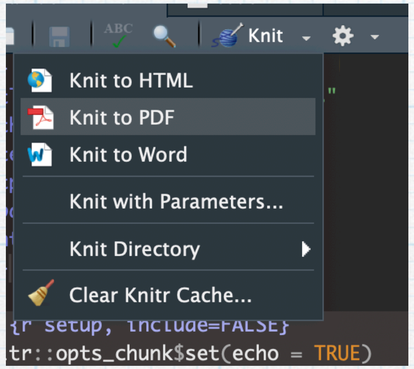
```
newport.beach <- newport.beach %>%
```



http://rpubs.com/rpubs.com/ychristine/newportbeachanalysis/

RMarkdown: knitting to other formats

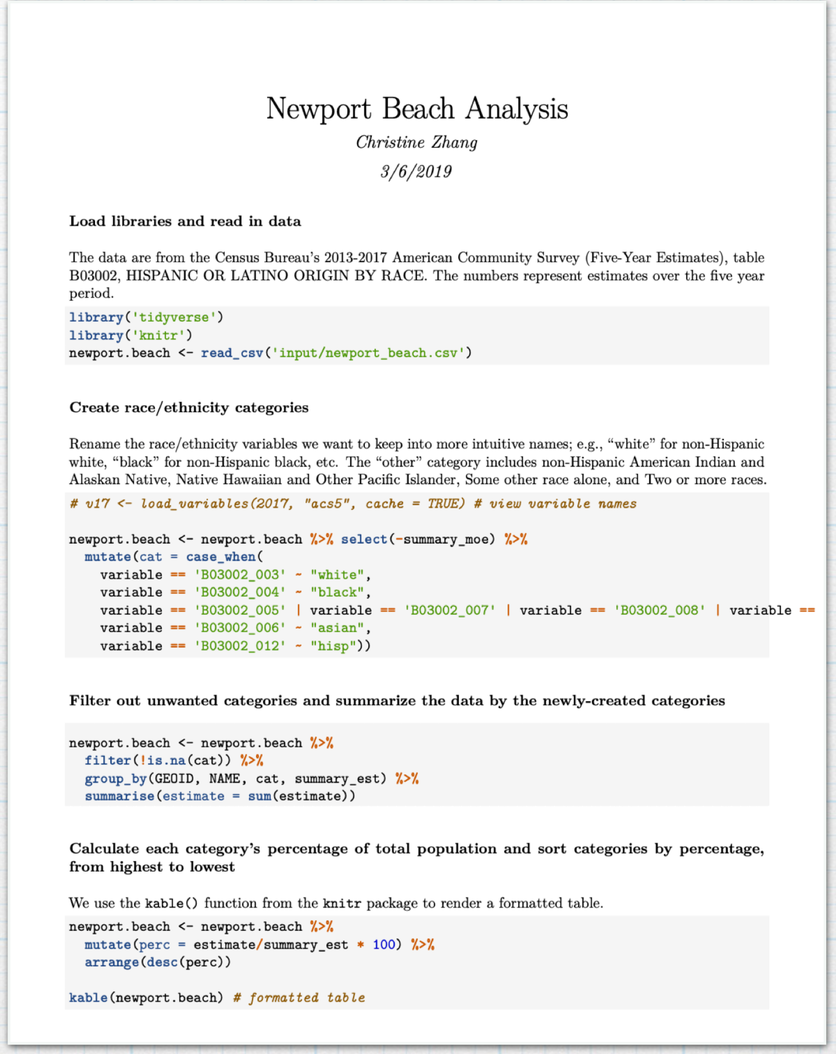
You can also knit to PDF, but you'll need to install LaTeX first: https://www.latex-project.org/get/



Also, GitHub-"flavored" Markdown: (more on that later)

```
---
title: "Newport Beach Analysis"
author: "Christine Zhang"
date: "3/6/2019"
output:
  md_document:
    variant: markdown_github
---
```

code_rmarkdown.md



Newport Beach Analysis

Christine Zhang

3/6/2019

Load libraries and read in data

The data are from the Census Bureau's 2013-2017 American Community Survey (Five-Year Estimates), table B03002, HISPANIC OR LATINO ORIGIN BY RACE. The numbers represent estimates over the five year period.

```
library('tidyverse')
library('knitr')
newport.beach <- read_csv('input/newport_beach.csv')
```

Create race/ethnicity categories

Rename the race/ethnicity variables we want to keep into more intuitive names; e.g., "white" for non-Hispanic white, "black" for non-Hispanic black, etc. The "other" category includes non-Hispanic American Indian and Alaskan Native, Native Hawaiian and Other Pacific Islander, Some other race alone, and Two or more races.

```
# v17 <- load_variables(2017, "acs5", cache = TRUE) # view variable names

newport.beach <- newport.beach %>% select(-summary_moe) %>%
  mutate(cat = case_when(
    variable == 'B03002_003' ~ "white",
    variable == 'B03002_004' ~ "black",
    variable == 'B03002_005' | variable == 'B03002_007' | variable == 'B03002_008' | variable ==
    variable == 'B03002_006' ~ "asian",
    variable == 'B03002_012' ~ "hisp"))
```

Filter out unwanted categories and summarize the data by the newly-created categories

```
newport.beach <- newport.beach %>%
  filter(!is.na(cat)) %>%
  group_by(GEOID, NAME, cat, summary_est) %>%
  summarise(estimate = sum(estimate))
```

Calculate each category's percentage of total population and sort categories by percentage, from highest to lowest

We use the `kable()` function from the `knitr` package to render a formatted table.

```
newport.beach <- newport.beach %>%
  mutate(perc = estimate/summary_est * 100) %>%
  arrange(desc(perc))

kable(newport.beach) # formatted table
```

code_rmarkdown.pdf

RMarkdown: convert to straight-up R code

You'll need the `purl()` function in the `knitr` package for this

```
purl("code_rmarkdown.Rmd", documentation = 2)
```

How much of the interspersed text to include:

0: include only R code
1: include code chunk labels as comments
2: include code chunk labels and all interspersed text as comments

Jupyter notebook

Replace this with the location of your files

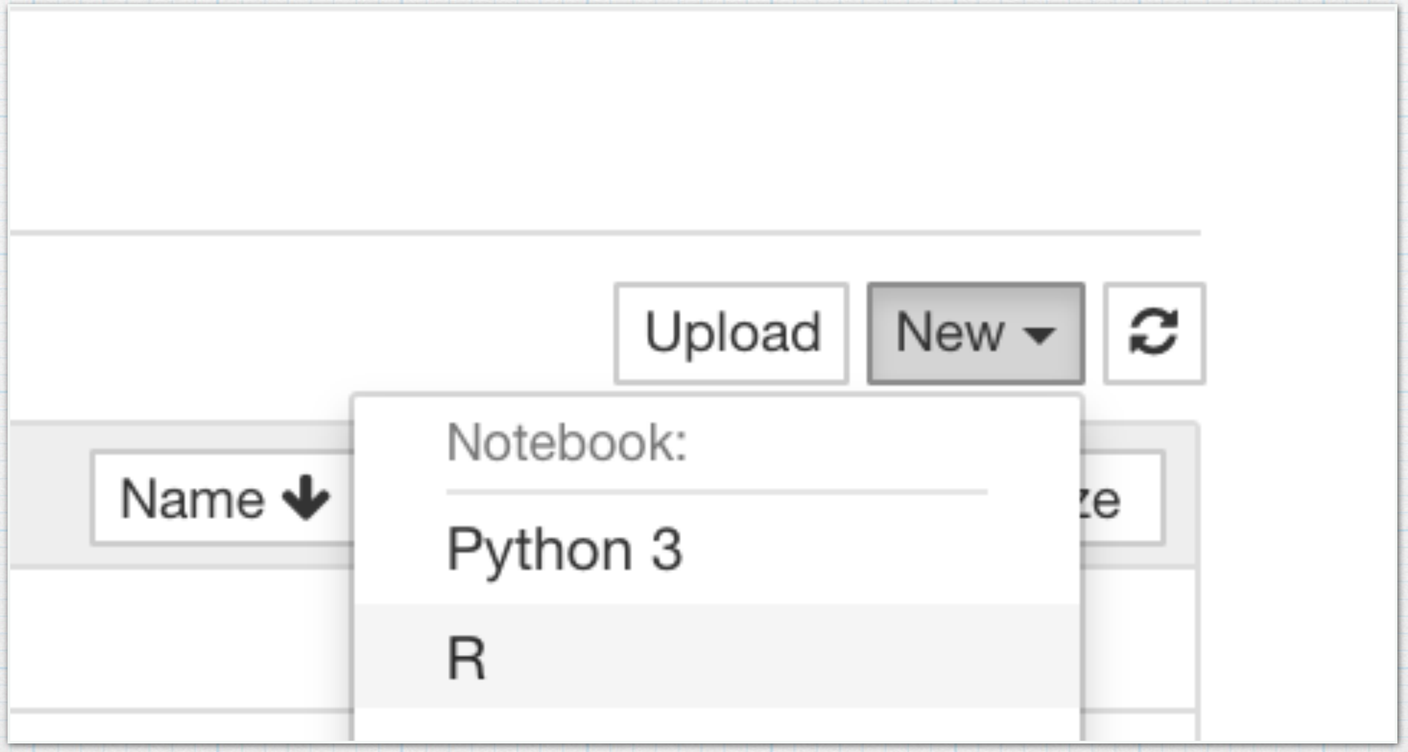
In terminal:

```
:~ czhang$ cd Documents/NICAR
:NICAR czhang$ jupyter notebook
```

This will open up in your browser:

Jupyter notebook

Notebook requires installing R kernel for Jupyter:
https://irkernel.github.io/



Newport Beach Analysis

An analysis of race/ethnicity categories in Newport Beach, California.

Load libraries and read in data

The data are from the Census Bureau's 2013-2017 American Community Survey (Five-Year Estimates), table B03002, [HISPANIC OR LATINO ORIGIN BY RACE](https://factfinder.census.gov/bkmk/table/1.0/en/ACS/17_5YR/B03002/1600000US0651182). The numbers represent estimates over the five year period.

```
suppressMessages(library('tidyverse'))
newport.beach <- suppressMessages(read_csv('input/newport_beach.csv'))
```

Create race/ethnicity categories

Rename the race/ethnicity variables we want to keep into more intuitive names; e.g., "white" for non-Hispanic white, "black" for non-Hispanic black, etc. The "other" category includes non-Hispanic American Indian and Alaskan Native, Native Hawaiian and Other Pacific Islander, Some other race alone, and Two or more races.

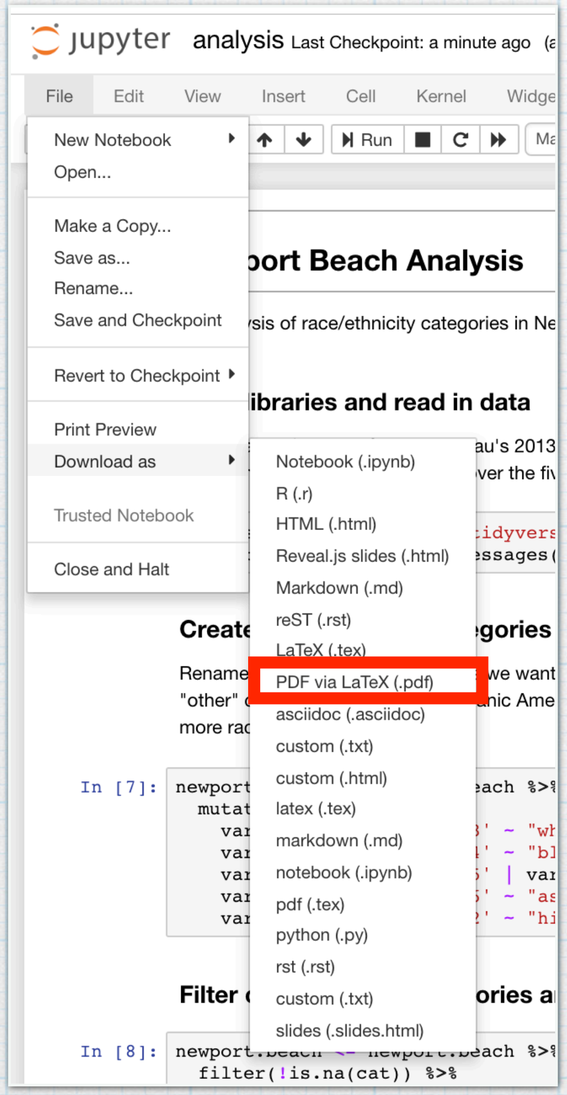
```
newport.beach <- newport.beach %>% select(-summary_moe) %>%
  mutate(cat = case_when(
    variable == 'B03002_003' ~ "white",
    variable == 'B03002_004' ~ "black",
    variable == 'B03002_005' | variable == 'B03002_007' | variable == 'B03002_008' | variable == 'B03002_009' ~ "other",
    variable == 'B03002_006' ~ "asian",
    variable == 'B03002_012' ~ "hisp"))
```

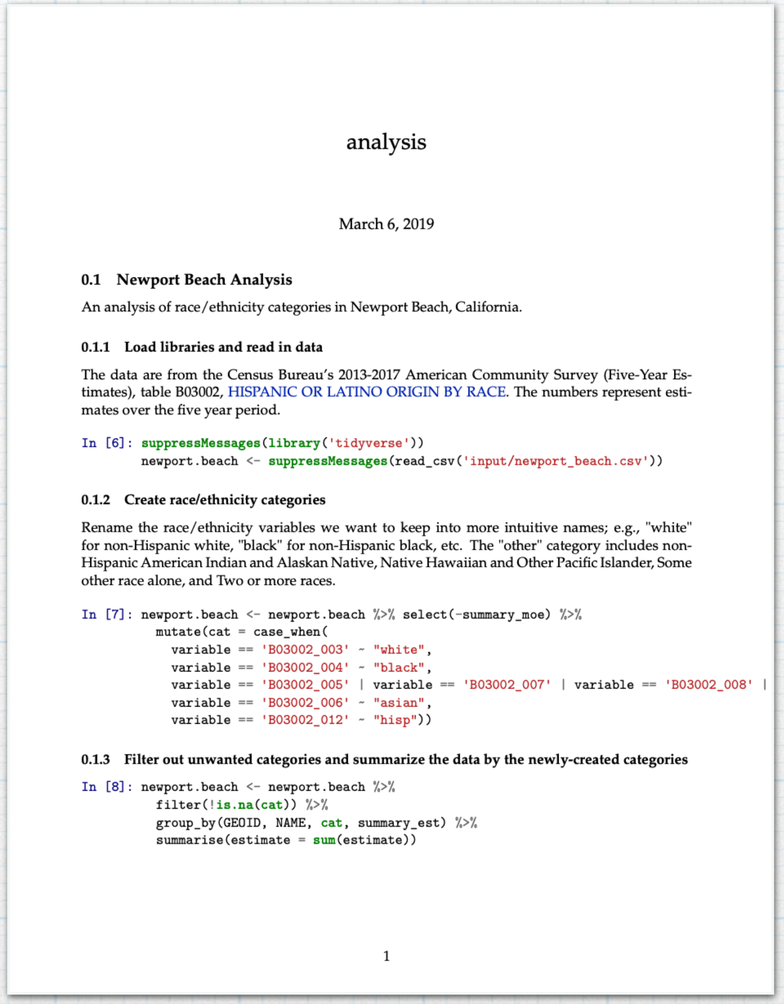
Filter out unwanted categories and summarize the data by the newly-created categories

```
newport.beach <- newport.beach %>%
  filter(!is.na(cat)) %>%
```

analysis.ipynb

Jupyter notebook: saving as other formats





analysis

March 6, 2019

0.1 Newport Beach Analysis

An analysis of race/ethnicity categories in Newport Beach, California.

0.1.1 Load libraries and read in data

The data are from the Census Bureau's 2013-2017 American Community Survey (Five-Year Estimates), table B03002, HISPANIC OR LATINO ORIGIN BY RACE. The numbers represent estimates over the five year period.

```
In [6]: suppressMessages(library('tidyverse'))
        newport.beach <- suppressMessages(read_csv('input/newport_beach.csv'))
```

0.1.2 Create race/ethnicity categories

Rename the race/ethnicity variables we want to keep into more intuitive names; e.g., "white" for non-Hispanic white, "black" for non-Hispanic black, etc. The "other" category includes non-Hispanic American Indian and Alaskan Native, Native Hawaiian and Other Pacific Islander, Some other race alone, and Two or more races.

```
In [7]: newport.beach <- newport.beach %>% select(-summary_moe) %>%
          mutate(cat = case_when(
            variable == 'B03002_003' ~ "white",
            variable == 'B03002_004' ~ "black",
            variable == 'B03002_005' | variable == 'B03002_007' | variable == 'B03002_008' |
            variable == 'B03002_006' ~ "asian",
            variable == 'B03002_012' ~ "hisp"))
```

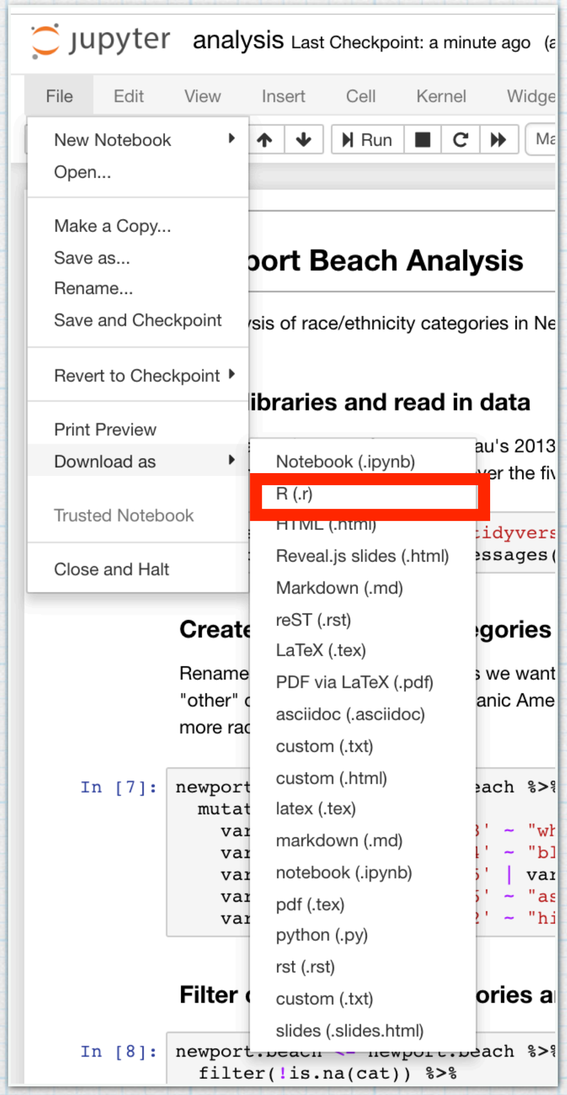
0.1.3 Filter out unwanted categories and summarize the data by the newly-created categories

```
In [8]: newport.beach <- newport.beach %>%
          filter(!is.na(cat)) %>%
          group_by(GEOID, NAME, cat, summary_est) %>%
          summarise(estimate = sum(estimate))
```

1

analysis.pdf

Jupyter notebook: saving as other formats




```r
suppressMessages(library('tidyverse'))
newport.beach <- suppressMessages(read_csv('input/newport_beach.csv'))

newport.beach <- newport.beach %>% select(-summary_moe) %>%
  mutate(cat = case_when(
    variable == 'B03002_003' ~ "white",
    variable == 'B03002_004' ~ "black",
    variable == 'B03002_005' | variable == 'B03002_007' | variable == 'B03002_008' | variable == 'B03002_009' ~ "other",
    variable == 'B03002_006' ~ "asian",
    variable == 'B03002_012' ~ "hisp"))

newport.beach <- newport.beach %>%
  filter(!is.na(cat)) %>%
  group_by(GEOID, NAME, cat, summary_est) %>%
  summarise(estimate = sum(estimate))

newport.beach <- newport.beach %>%
  mutate(perc = estimate/summary_est * 100) %>%
  arrange(desc(perc))

newport.beach

write_csv(newport.beach, 'output/newport_beach_cleaned.csv')
```

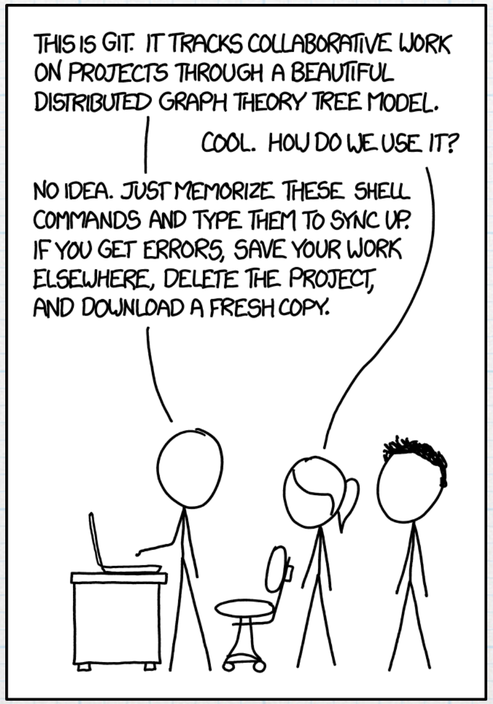
analysis.R

Jupyter + GitHub

So you have a Jupyter notebook called “analysis.ipynb” …. Now what??

Share it on GitHub* (or GitLab, etc.)

Here is a walkthrough of how it works: https://happygitwithr.com/



underthecurve / NICAR

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

No description, website, or topics provided.

Edit

Manage topics

4 commits | 1 branch | 0 releases | 0 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

Zhang and Zhang locations — Latest commit b6ab021 4 minutes ago

documentation	locations	4 minutes ago
input	locations	4 minutes ago
output	locations	4 minutes ago
.gitignore	locations	4 minutes ago
README.md	locations	4 minutes ago
analysis.ipynb	locations	4 minutes ago
analysis.pdf	locations	4 minutes ago
code.R	locations	4 minutes ago
code_rmarkdown.Rmd	locations	4 minutes ago
code_rmarkdown.html	locations	4 minutes ago
code_rmarkdown.md	locations	4 minutes ago
code_rmarkdown.pdf	locations	4 minutes ago

https://github.com/underthecurve/NICAR

underthecurve / NICAR

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

Branch: master | NICAR / analysis.ipynb | Find file | Copy path

Zhang locations b6ab021 8 minutes ago

0 contributors

190 lines (189 sloc) | 8.2 KB | Raw | Blame | History

Newport Beach Analysis

An analysis of race/ethnicity categories in Newport Beach, California.

Load libraries and read in data

The data are from the Census Bureau's 2013-2017 American Community Survey (Five-Year Estimates), table B03002, HISPANIC OR LATINO ORIGIN BY RACE. The numbers represent estimates over the five year period.

In [6]:
```
suppressMessages(library('tidyverse'))
newport.beach <- suppressMessages(read_csv('input/newport_beach.csv'))
```

Create race/ethnicity categories

Rename the race/ethnicity variables we want to keep into more intuitive names; e.g., "white" for non-Hispanic white, "black" for non-Hispanic black, etc. The "other" category includes non-Hispanic American Indian and Alaskan Native, Native Hawaiian and Other Pacific Islander, Some other race alone, and Two or more races.

In [7]:
```
newport.beach <- newport.beach %>% select(-summary_moe) %>%
  mutate(cat = case_when(
    variable == 'B03002_003' ~ "white",
    variable == 'B03002_004' ~ "black",
    variable == 'B03002_005' | variable == 'B03002_007' | variable == 'B03002_008' | variable ==
'B03002_009' ~ "other",
    variable == 'B03002_006' ~ "asian",
    variable == 'B03002_012' ~ "hisp"))
```

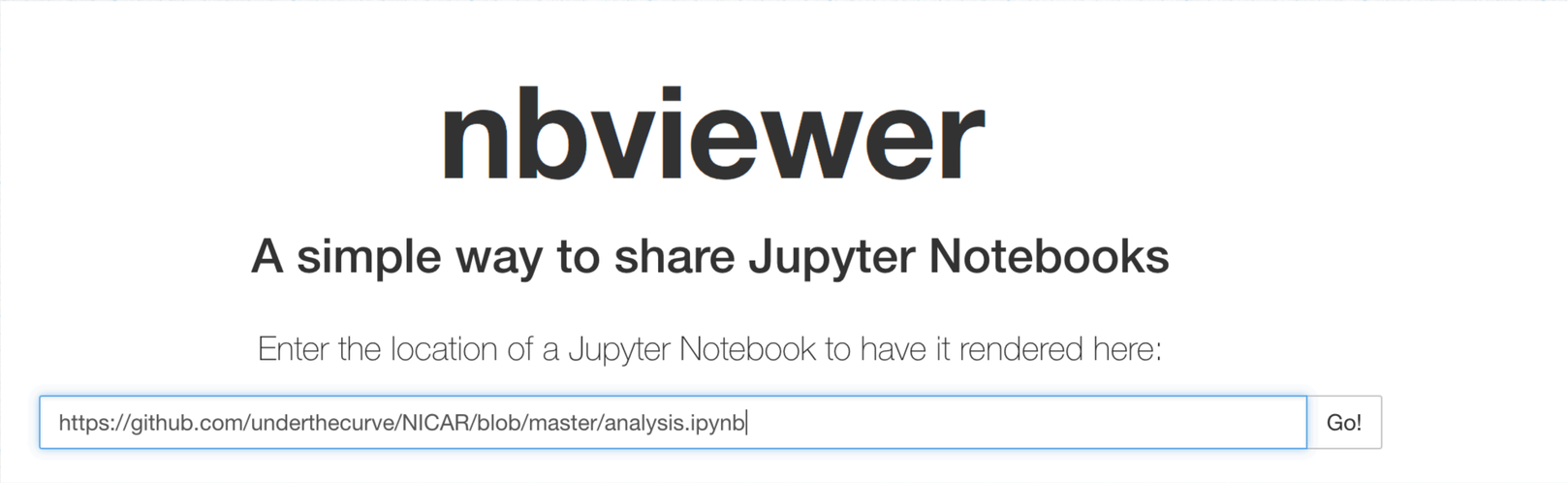
Filter out unwanted categories and summarize the data by the newly-created categories

In [8]:
```
newport.beach <- newport.beach %>%
  filter(!is.na(cat)) %>%
  group_by(GEOID, NAME, cat, summary_est) %>%
  summarise(estimate = sum(estimate))
```

https://github.com/underthecurve/NICAR/blob/master/analysis.ipynb

Jupyter + GitHub + NBViewer

Sometimes Jupyter notebooks take a long time to render on GitHub. NBViewer makes things a bit easier: https://nbviewer.jupyter.org/



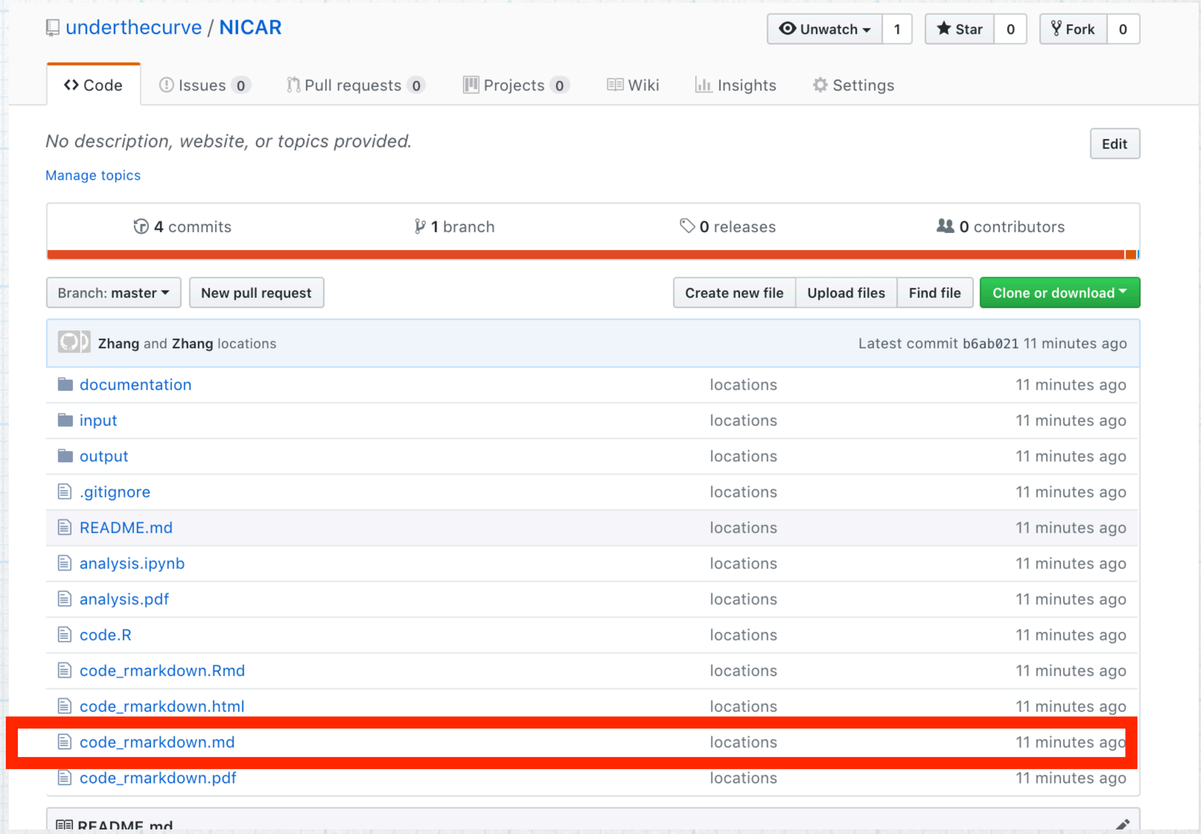
https://nbviewer.jupyter.org/github/underthecurve/NICAR/blob/master/analysis.ipynb

What about RMarkdown + GitHub?

Remember this?

```yaml
---
title: "Newport Beach Analysis"
author: "Christine Zhang"
date: "3/6/2019"
output:
  md_document:
    variant: markdown_github
---
```

code_rmarkdown.md



Branch: master ▾ NICAR / code_rmarkdown.md Find file Copy path

Zhang locations b6ab021 12 minutes ago

0 contributors

61 lines (45 sloc) | 2.54 KB Raw Blame History

Load libraries and read in data

The data are from the Census Bureau's 2013-2017 American Community Survey (Five-Year Estimates), table B03002, HISPANIC OR LATINO ORIGIN BY RACE. The numbers represent estimates over the five year period.

```
library('tidyverse')
library('knitr')
newport.beach <- read_csv('input/newport_beach.csv')
```

Create race/ethnicity categories

Rename the race/ethnicity variables we want to keep into more intuitive names; e.g., "white" for non-Hispanic white, "black" for non-Hispanic black, etc. The "other" category includes non-Hispanic American Indian and Alaskan Native, Native Hawaiian and Other Pacific Islander, Some other race alone, and Two or more races.

```
# v17 <- load_variables(2017, "acs5", cache = TRUE) # view variable names

newport.beach <- newport.beach %>% select(-summary_moe) %>%
  mutate(cat = case_when(
    variable == 'B03002_003' ~ "white",
    variable == 'B03002_004' ~ "black",
    variable == 'B03002_005' | variable == 'B03002_007' | variable == 'B03002_008' | variable == 'B03002_0
    variable == 'B03002_006' ~ "asian",
    variable == 'B03002_012' ~ "hisp"))
```

Filter out unwanted categories and summarize the data by the newly-created categories

```
newport.beach <- newport.beach %>%
  filter(!is.na(cat)) %>%
  group_by(GEOID, NAME, cat, summary_est) %>%
  summarise(estimate = sum(estimate))
```

https://github.com/underthecurve/NICAR/blob/master/code_rmarkdown.md

Conclusion

We went through three main ways of sharing your work in R. Each has pros and cons.

Straight-up code

Pros:

Cons:

Conclusion

We went through three main ways of sharing your work in R. Each has pros and cons.

Straight-up code

Pros: not dependent on any other programs (you only need R), simple

Cons: comments can't be formatted, looks less slick/accessible, if your code is long or complicated, it's not going to be a pretty picture

Conclusion

We went through three main ways of sharing your work in R. Each has pros and cons.

RMarkdown

Pros:

Cons:

Conclusion

We went through three main ways of sharing your work in R. Each has pros and cons.

RMarkdown

Pros: can be run from within RStudio, code can be interspersed with prose, you don't need to know GitHub to publish —> can publish directly to RPubs if you sign up, can save as different formats

Cons: each code chunk must be labeled with ```{r} CODE ```, publishing to GitHub requires exporting to a different format (the "GitHub-flavored" markdown)

Conclusion

We went through three main ways of sharing your work in R. Each has pros and cons.

Jupyter notebook

Pros: code can be interspersed with prose, can save as different formats, keyboard shortcuts are a little easier to get used to, renders directly in GitHub / works well with NBViewer, Jupyter also has other kernels like Python, tables formatted automatically

Cons: if you want to publish online you need to know GitHub (or GitLab, etc.), you have to install Jupyter (and the R Kernel for Jupyter)

Thomas Mock @thomas_mock · Jan 28

Anyone else got a list of News sources that have most if not all of their raw data on GitHub (or elsewhere)?

My known list:
@BuzzFeedNews
@FiveThirtyEight
@baltsundata
@TheEconomist (only recently)
@UpshotNYT

2 9 30

Christine Zhang
@christinezhang

Replying to @thomas_mock @BuzzFeedNews and 4 others

Check out @LATdatadesk 🤓

10:13 AM - 28 Jan 2019

1 Like

1 1

Let's add to this list!

https://twitter.com/christinezhang/status/1089904357070594050

Thank you!

Christine Zhang
The Baltimore Sun
@christinezhang | @baltsundata