Using R and Python at *The Baltimore Sun*

Christine Zhang
Friday, June 7, 2019
czhang@baltsun.com
@christinezhang

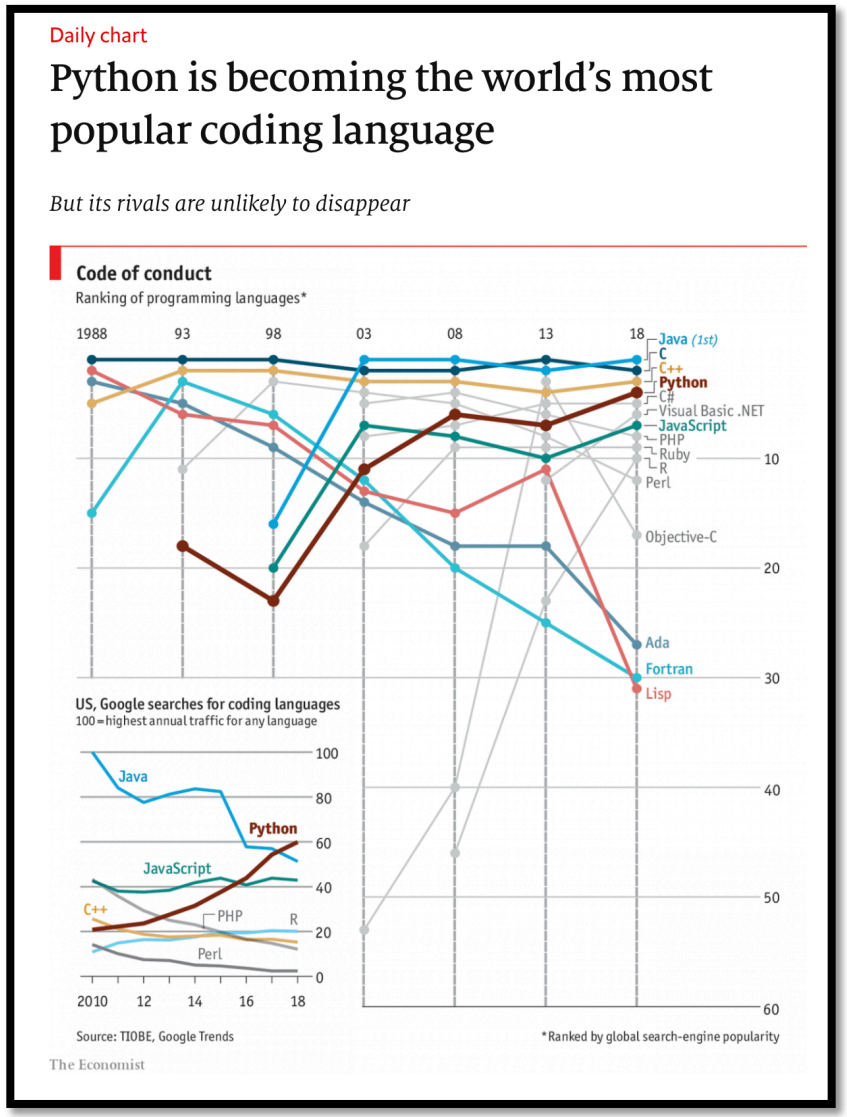
repo with data & code: http://bit.ly/2EXWPUG

About Baltimore Sun Data / Interactives

data.baltimoresun.com
twitter.com/baltsundata
github.com/baltimore-sun-data/

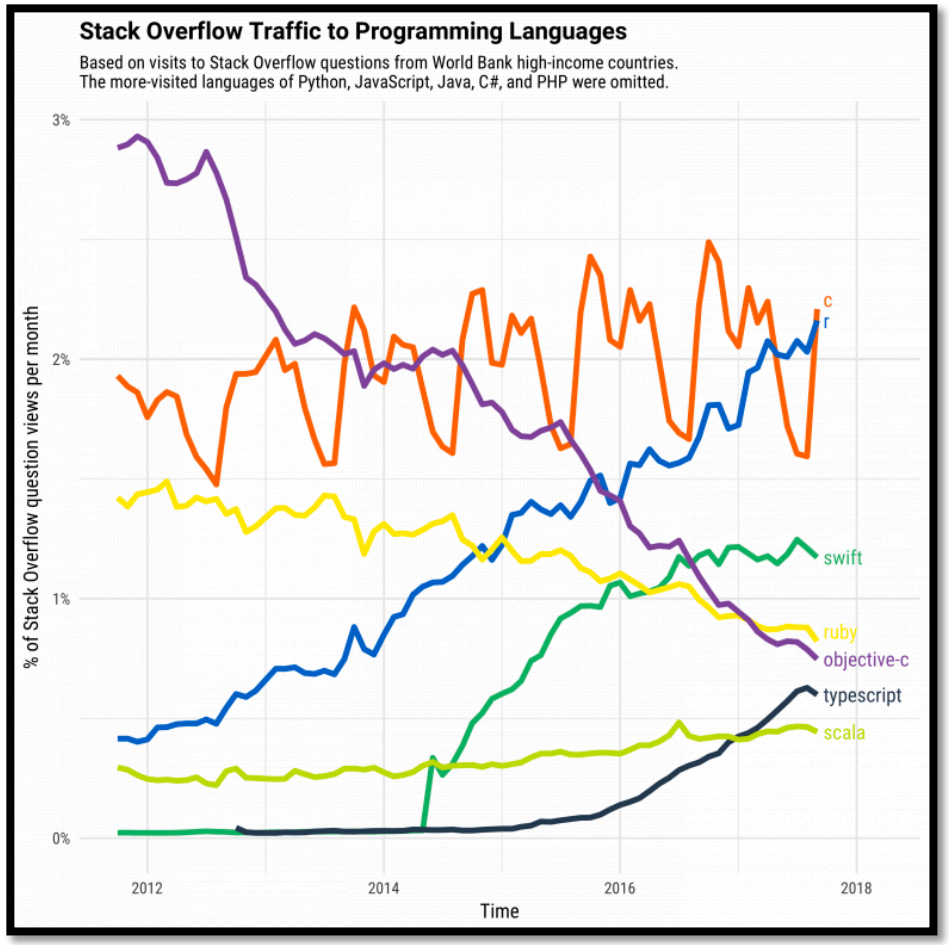
About me

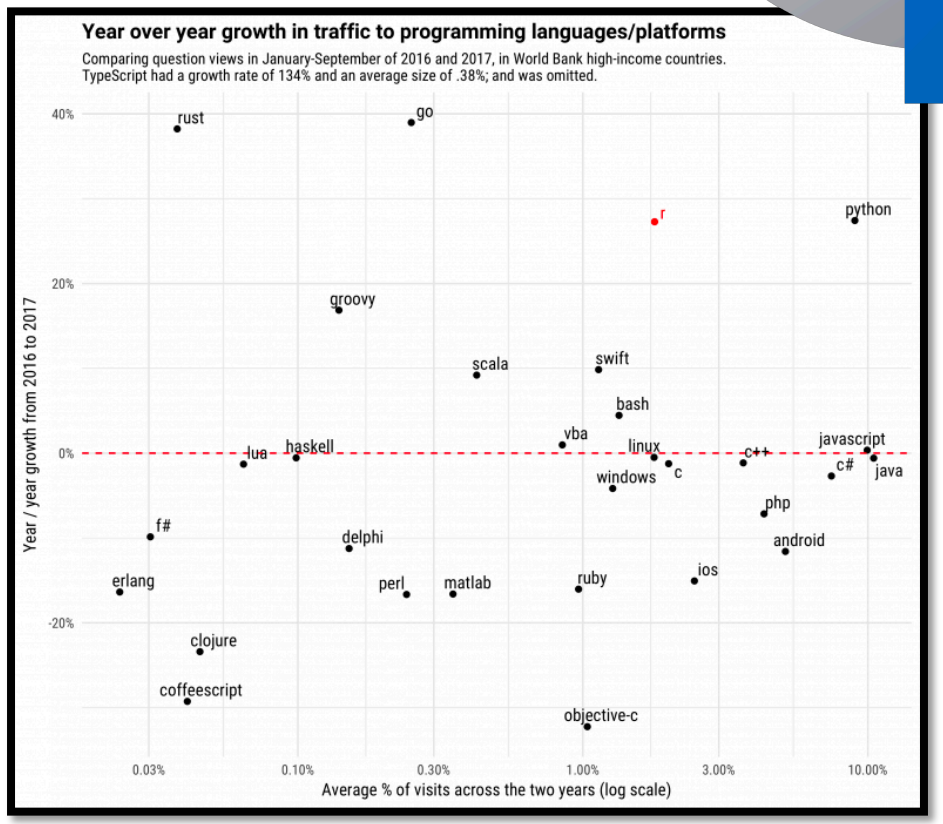
“Python vs R”



https://www.economist.com/graphic-detail/2018/07/26/python-is-becoming-the-worlds-most-popular-coding-language

“Python vs R”

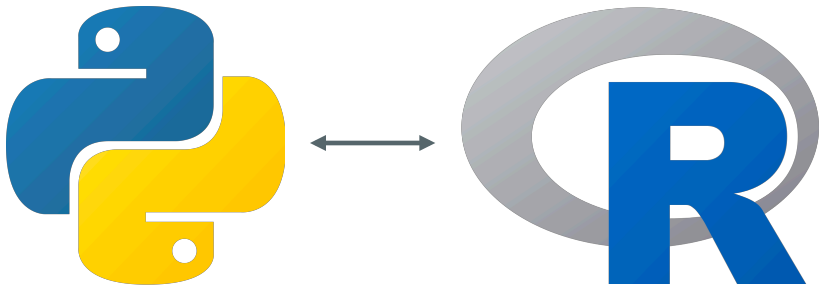




https://stackoverflow.blog/2017/10/10/impressive-growth-r/

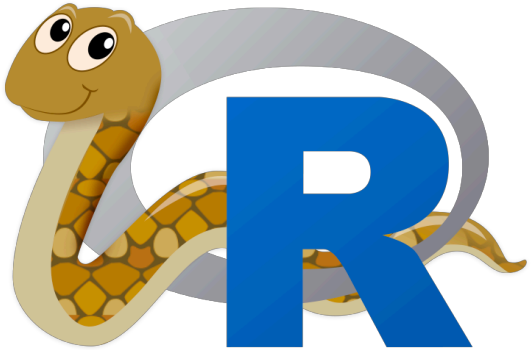
Three ways I use Python **and** R together

Separate workflows that feed into each other

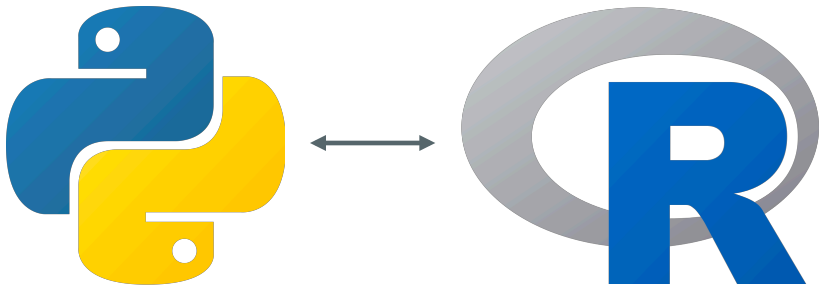


Running R inside a Python script

Running Python inside an R script



Separate workflows that feed into each other

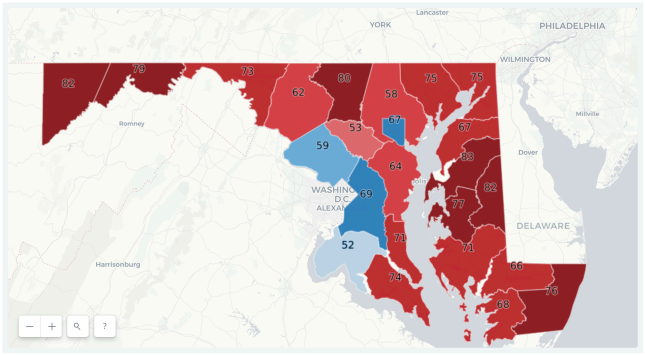


"Where Hogan and Jealous got their votes," November 26, 2018

Early voting for governor, by county

The map displays the percentage of Early Voting votes received by the winning candidate in each county. These include votes cast during the eight-day Early Voting period from Thursday, Oct. 25, 2018 through Thursday, Nov. 1. Download the data: baltimoresun.com/precinct-map.

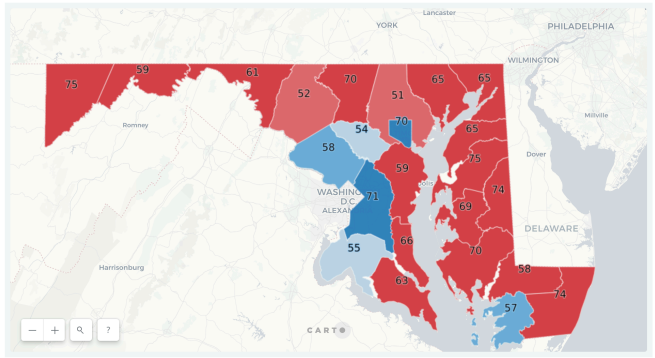
Updates: *The map has been updated to reflect results for all precincts.*

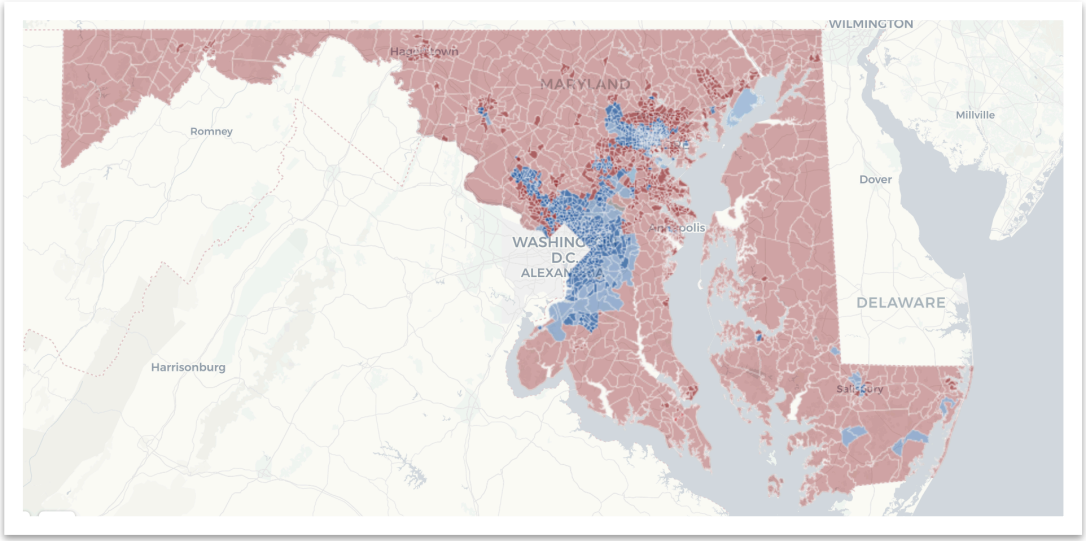


Absentee and Provisional voting for governor, by county

The map displays the percentage of received by the winning candidate in each county, based on Absentee and Provisional ballot results. Download the data: baltimoresun.com/precinct-map.

Note: *The map has been updated to reflect results for all precincts.*





https://www.baltimoresun.com/politics/bs-md-governor-election-precinct-map-20181107-htmlstory.html

https://github.com/baltimore-sun-data/maryland-2018-governor-precinct-map

The goal:

Make a county-level map of Maryland's Early Voting and Absentee voting results for the 2018 gubernatorial election.

The goal:

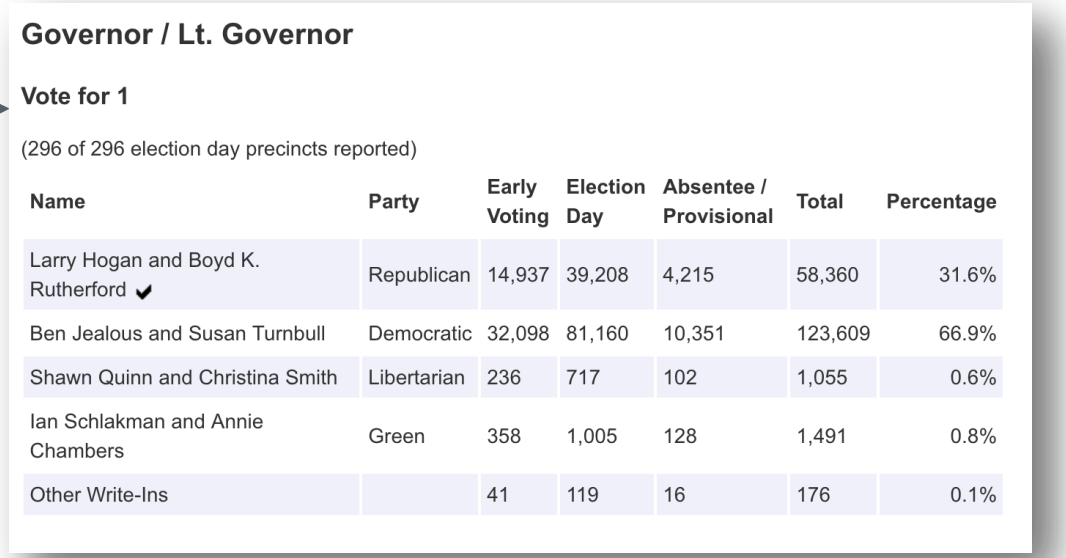
Make a county-level map of Maryland's Early Voting and Absentee voting results for the 2018 gubernatorial election.

The challenge:

The numbers on the website are not all in one place. Each county has its own page.

Results for State and Local Offices by County

Allegany County	Anne Arundel County	**Baltimore City**
Baltimore County	Calvert County	Caroline County
Carroll County	Cecil County	Charles County
Dorchester County	Frederick County	Garrett County
Harford County	Howard County	Kent County
Montgomery County	Prince George's County	Queen Anne's County
St. Mary's County	Somerset County	Talbot County
Washington County	Wicomico County	Worcester County



Governor / Lt. Governor

Vote for 1

(296 of 296 election day precincts reported)

Name	Party	Early Voting	Election Day	Absentee / Provisional	Total	Percentage
Larry Hogan and Boyd K. Rutherford ✔	Republican	14,937	39,208	4,215	58,360	31.6%
Ben Jealous and Susan Turnbull	Democratic	32,098	81,160	10,351	123,609	66.9%
Shawn Quinn and Christina Smith	Libertarian	236	717	102	1,055	0.6%
Ian Schlakman and Annie Chambers	Green	358	1,005	128	1,491	0.8%
Other Write-Ins		41	119	16	176	0.1%

https://elections.maryland.gov/elections/2018/results/general/index.html

https://elections.maryland.gov/elections/2018/results/general/gen_results_2018_2_by_county_03-1.html

The goal:

Make a county-level map of Maryland's Early Voting and Absentee voting results for the 2018 gubernatorial election.

The challenge:

The numbers on the website are not all in one place. Each county has its own page.

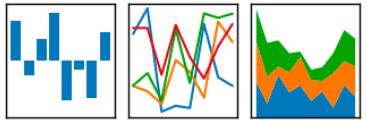
The process:

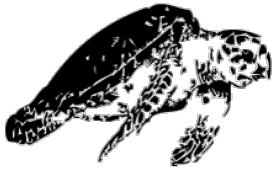
1. Scrape the data for each county from the state Board of Elections results page into a CSV.
2. Combine all the CSVs into one output file.
3. Combine the results data with GIS shapefiles in order to make a map.

1. Scrape the data for each county

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$





Requests

Beautiful Soup

```
import pandas as pd
import csv
import requests
from bs4 import BeautifulSoup
```

```
counties = pd.read_csv('input/countylist.csv', dtype = {'id':str})
counties['id'] = counties['id'].str.zfill(2)
```

```
for i in counties['id']:
    r = requests.get('https://elections.maryland.gov/elections/2018/results/general/gen_results_2018_2_by_county_' +
                     format(i) + '-1.html')

    soup = BeautifulSoup(r.content,'html.parser')

    tables = soup.find_all(lambda tag: tag.name == 'table')
    candidates = soup.select('td[headers="CandidateName003-"]')
    early_votes = soup.select('td[headers="EarlyVotes003-"]')
    election_day_votes = soup.select('td[headers="ElectionNightVotes003-"]')
    absentee_prov_votes = soup.select('td[headers="AbsenteeAndProvisionalVotes003-"]')
    total_votes = soup.select('td[headers="TotalVotes003-"]')
    percentages = soup.select('td[headers="Percentage003-"]')

    cands = []

    for candidate in candidates:
        # print(candidate.text.strip())
        cands.append(candidate.text.strip())

    early = []
    for early_vote in early_votes:
        # print(early_vote.text.strip())
        early.append(early_vote.text.strip())

    electday = []
    for election_day_vote in election_day_votes:
        # print(election_day_vote.text.strip())
        electday.append(election_day_vote.text.strip())

    absentee = []
    for absentee_prov_vote in absentee_prov_votes:
        # print(absentee_prov_vote.text.strip())
        absentee.append(absentee_prov_vote.text.strip())

    total = []
    for total_vote in total_votes:
        # print(total_vote.text.strip())
        total.append(total_vote.text.strip())

    perc = []
    for percentage in percentages:
        # print(percentage.text.strip())
        perc.append(percentage.text.strip())


    table_list = pd.DataFrame(
```

[scraping_results.ipynb]

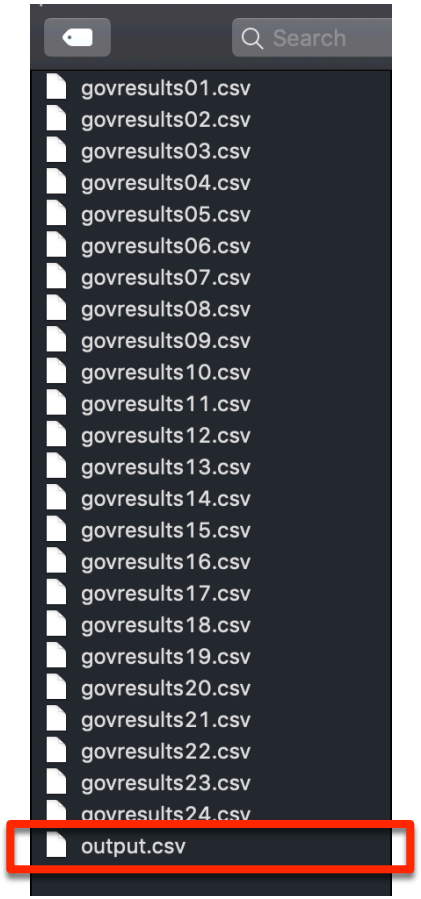
2. Combine the CSVs into one file

csvkit 1.0.4

```
pip install csvkit
```

A suite of command-line tools for working with CSV, the king of tabular file formats.

```
csvstack *.csv > output.csv
```

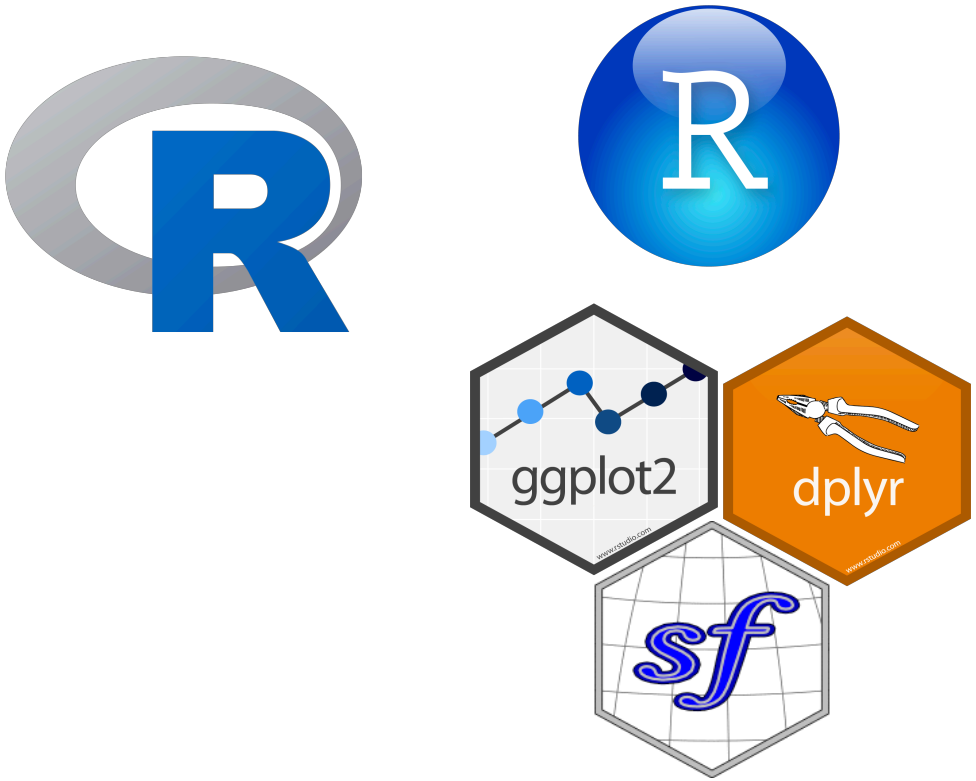


https://csvkit.readthedocs.io/en/latest/

2. Combine the CSVs into one file

A	B	C	D	E	F	G	H
candidate	early_vote	electday_vote	absentee_vote	total_vote	perc	id	county
Larry Hogan and Boyd K. Rutherford	2,055	15,993	1,176	19,224	81.70%	1	Allegany
Ben Jealous and Susan Turnbull	516	2,689	780	3,985	16.90%	1	Allegany
Shawn Quinn and Christina Smith	12	136	18	166	0.70%	1	Allegany
Ian Schlakman and Annie Chambers	17	100	22	139	0.60%	1	Allegany
Other Write-Ins	2	13	1	16	0.10%	1	Allegany
Larry Hogan and Boyd K. Rutherford	44,549	103,436	9,217	157,202	68.60%	2	Anne Arundel
Ben Jealous and Susan Turnbull	23,891	39,497	6,011	69,399	30.30%	2	Anne Arundel
Shawn Quinn and Christina Smith	318	913	135	1,366	0.60%	2	Anne Arundel
Ian Schlakman and Annie Chambers	315	619	118	1,052	0.50%	2	Anne Arundel
Other Write-Ins	60	113	14	187	0.10%	2	Anne Arundel
Larry Hogan and Boyd K. Rutherford	14,937	39,208	4,215	58,360	31.60%	3	Baltimore City
Ben Jealous and Susan Turnbull	32,098	81,160	10,351	123,609	66.90%	3	Baltimore City
Shawn Quinn and Christina Smith	236	717	102	1,055	0.60%	3	Baltimore City
Ian Schlakman and Annie Chambers	358	1,005	128	1,491	0.80%	3	Baltimore City
Other Write-Ins	41	119	16	176	0.10%	3	Baltimore City
Larry Hogan and Boyd K. Rutherford	59,150	128,960	10,012	198,122	61.10%	4	Baltimore
Ben Jealous and Susan Turnbull	41,257	72,308	9,208	122,773	37.90%	4	Baltimore
Shawn Quinn and Christina Smith	384	1,142	107	1,633	0.50%	4	Baltimore
Ian Schlakman and Annie Chambers	434	930	128	1,492	0.50%	4	Baltimore
Other Write-Ins	82	161	13	256	0.10%	4	Baltimore
Larry Hogan and Boyd K. Rutherford	6,176	21,923	1,511	29,610	75.40%	5	Calvert
Ben Jealous and Susan Turnbull	2,484	5,968	764	9,216	23.50%	5	Calvert
Shawn Quinn and Christina Smith	51	237	15	303	0.80%	5	Calvert
Ian Schlakman and Annie Chambers	26	98	6	130	0.30%	5	Calvert
Other Write-Ins	5	17	3	25	0.10%	5	Calvert
Larry Hogan and Boyd K. Rutherford	2,544	6,622	323	9,489	82.50%	6	Caroline

3. Combine the results data with GIS shapefiles



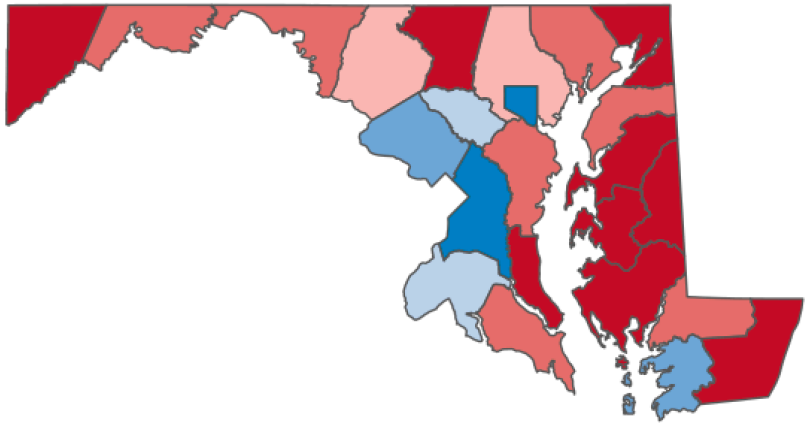

```r
library('tidyverse')
library('sf')

options(device = "X11")
X11.options(type = "cairo")

counties <- "input/Maryland_Physical_Boundaries__County_Boundaries_Generalized/Maryland_Physical_Boundaries__Coun
ty_Boundaries_Generalized.shp"
counties.read <- st_read(counties)

output <- read_csv('output/output.csv')

output <- output %>% mutate(cand = case_when(candidate == 'Ben  Jealous and Susan  Turnbull' ~ 'jealous',
                                             candidate == 'Larry  Hogan and Boyd K. Rutherford' ~ 'hogan',
                                             candidate == 'Ian  Schlakman and Annie  Chambers' ~ 'schlakman',
                                             candidate == 'Shawn  Quinn and Christina  Smith' ~ 'quinn',
                                             candidate == 'Other Write-Ins' ~ 'write_in'))

output <- output %>% group_by(county) %>% mutate(max_vote = max(absentee_vote),
                                                 total_abs_vote = sum(absentee_vote),
                                                 abs_perc = absentee_vote/total_abs_vote * 100)

output_wide <- output %>% spread(key = cand, value = absentee_vote)

output_wide <- output_wide %>% group_by(county) %>% mutate(jealous = max(jealous, na.rm = T),
                                                           hogan = max(hogan, na.rm = T),
                                                           schlakman = max(schlakman, na.rm = T),
                                                           quinn = max(quinn, na.rm = T),
                                                           write_in = max(write_in, na.rm = T))
```

https://data.imap.maryland.gov/datasets/4c172f80b626490ea2cff7b699febedb_1

absentee_code.R

3. Combine the results data with GIS shapefiles

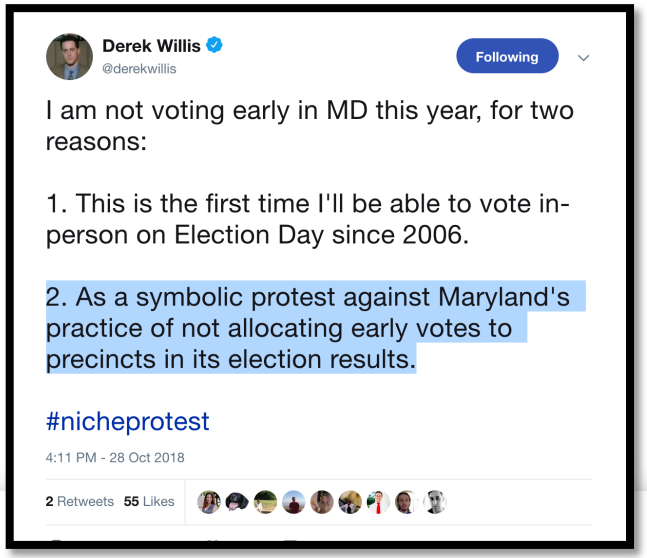
```
ggplot() + geom_sf(data = counties.merge,
                   aes(fill = perc_cats_map)) +
  scale_fill_gradient2(low = "#007dbe", mid = "white",
                       high = "#b8292f", space = "Lab", midpoint = 0,
                       na.value = "#afafaf", guide = "legend",
                       aesthetics = "fill") +
  theme_void() +
  theme(panel.grid.major = element_line(colour = 'transparent'))

st_write(counties.merge, 'output/counties_merge_absentee.shp')
```

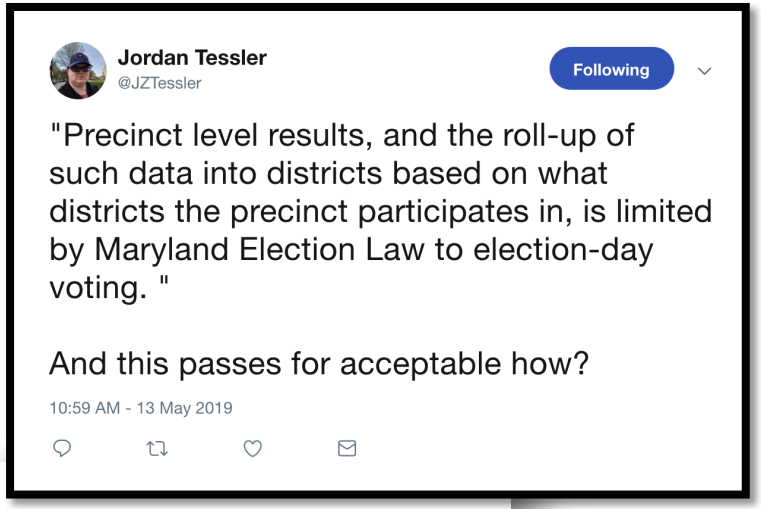


Note: the shapefile is saved so it can be used in other mapping software (in my case, CartoDB: https://baltsun.carto.com/builder/4bcdc78e-7567-4306-9bd6-2301b68ca5d0/embed)

absentee_code.R

Why aren't precinct-level results reported for Early and Absentee Voting?





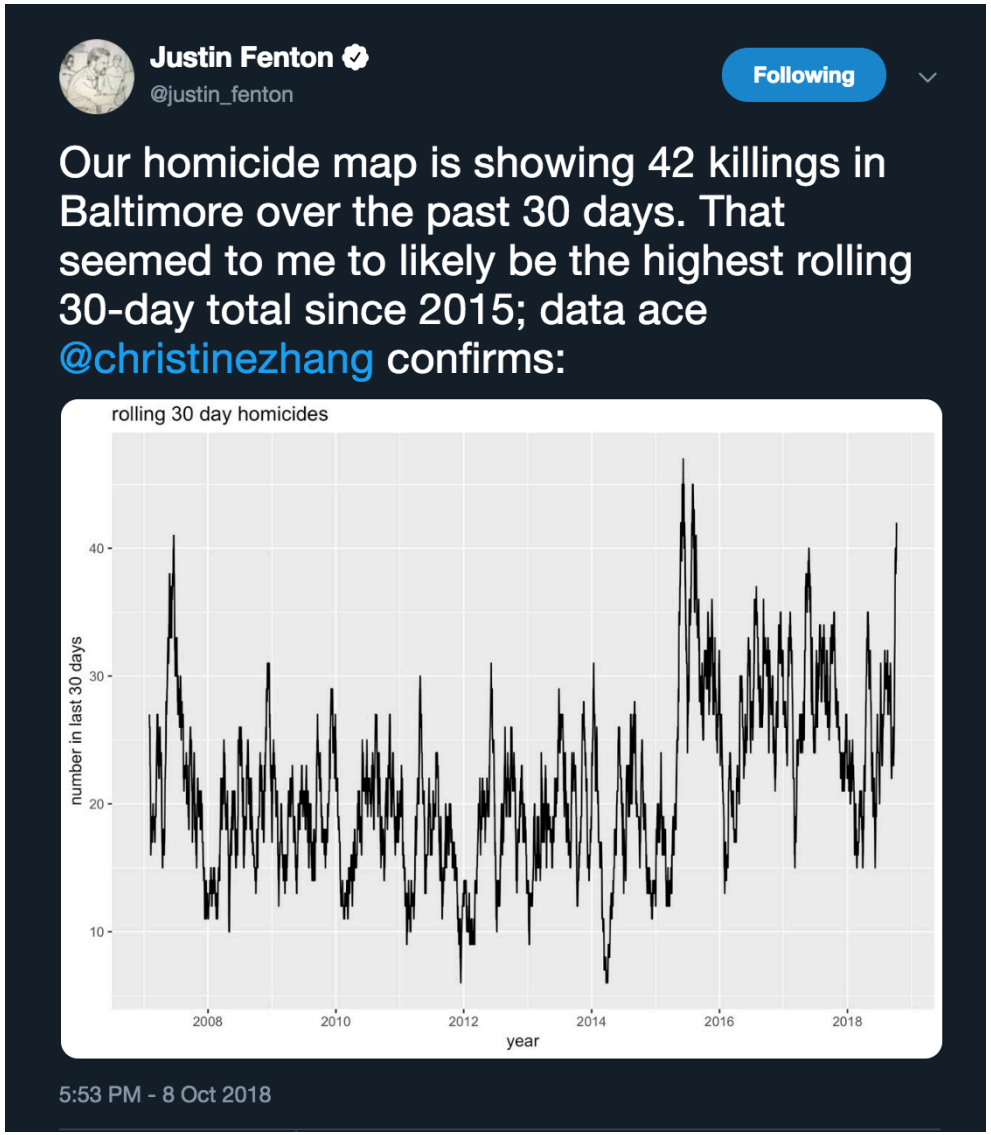
Assumptions and Constraints:

1. Precinct level results, and the roll-up of such data into districts based on what districts the precinct participates in, is limited by Maryland Election Law to election-day voting. Early Voting, Absentee and Provisional vote counts are not reported at the precinct level, so those counts are only included in the county level files.

https://elections.maryland.gov/elections/using_election_data_instructions.html

Three ways I use Python **and** R together

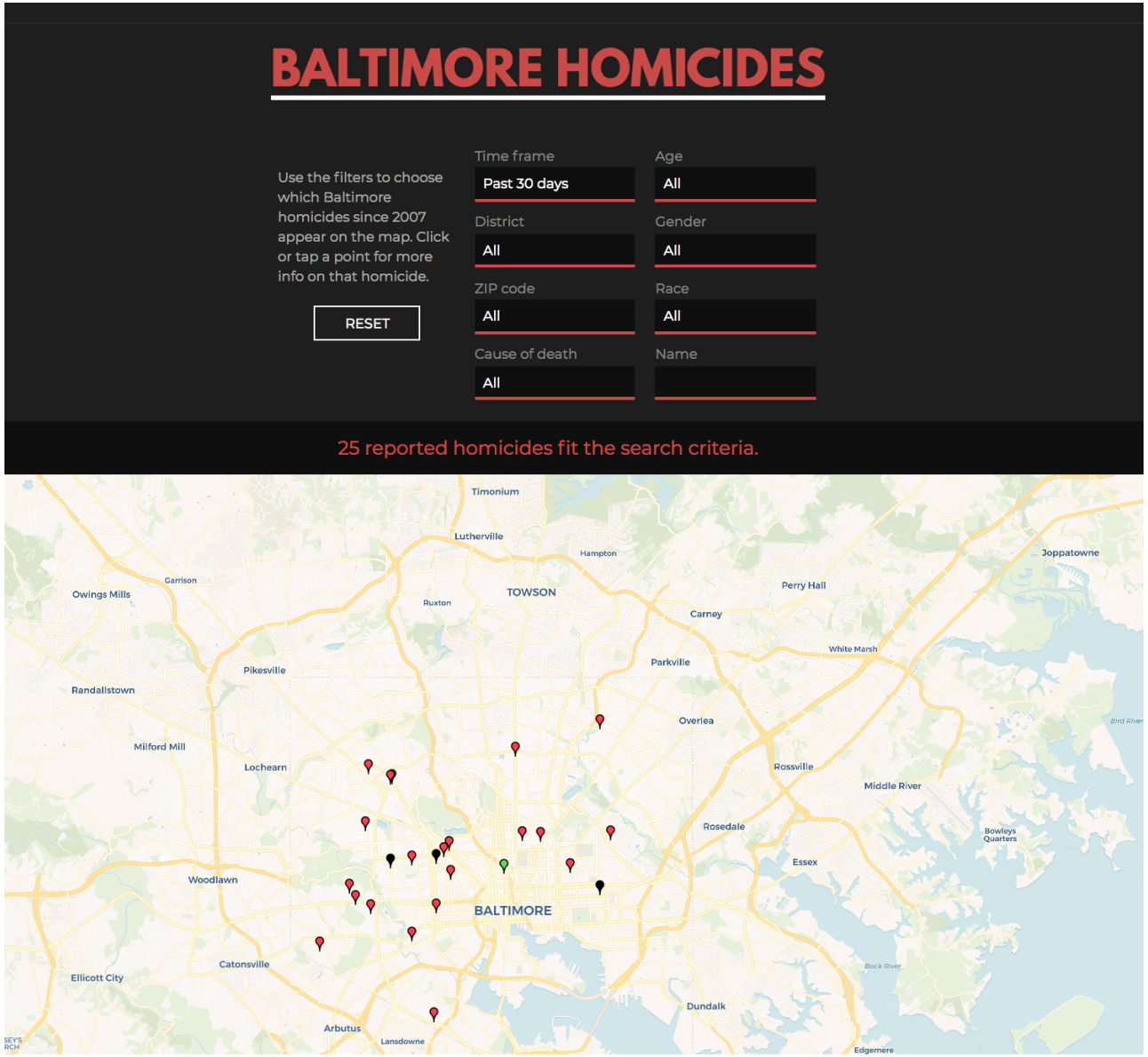
Running R inside a Python script



https://twitter.com/justin_fenton/status/1049417577968087040

The goal:

Use the homicides database to make a chart of the rolling 30-day sum of homicides since 2007.



https://baltimoresun.com/homicides

first_name	last_name	age	gender	race	cause	death_loc	district	street_address	zip_code	latitude	longitude	date_found	time_found
George	Phillips IV	19	male	black	shooting	unknown	NE	2800 Lake Ave	21213	39.32411	-76.57386	2018-10-08	*NA*
Brailynn	Ford	0	female	black	unknown	unknown	WD	1000 Mosher St	21217	39.30042	-76.63535	2018-10-08	*NA*
NA	*NA*	*NA*	male	unknown	shooting	unknown	WD	1800 N Fulton Ave	21217	39.30895	-76.64670	2018-10-07	23:30:00
Harry	Reckline	53	male	white	shooting	unknown	SW	2600 Cole St	21223	39.27841	-76.65591	2018-10-05	*NA*
Lawrence	Price	73	male	black	shooting	unknown	NW	4600 Liberty Heights Ave	21207	39.33129	-76.69434	2018-10-04	00:24:00
Andrew Omar	Allen	18	male	black	shooting	unknown	??	3300 Cardenas Ave	21213	39.32295	-76.57566	2018-10-03	23:05:00
Fanny	Machado	28	female	white	unknown	scene	WD	1300 N Fulton Ave	21217	39.30338	-76.64649	2018-10-03	14:27:00
Jason Reuben	Haynes	41	male	black	shooting	sinai-hospital	NW	4300 Park Heights Ave	21215	39.33781	-76.66544	2018-10-02	22:43:00
Trevos	Agard	41	male	black	shooting	unknown	ND	400 Homeland Ave	21212	39.35403	-76.61411	2018-10-01	20:23:00
Garfield	Leon Jr.	30	male	black	shooting	scene	NW	3000 Oakley Ave	21215	39.34801	-76.67054	2018-10-01	13:18:00

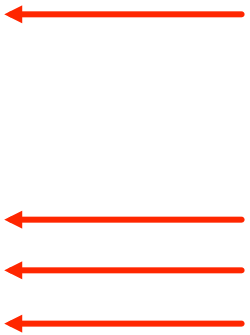
The goal:

Use the homicides database to make a chart of the rolling 30-day sum of homicides since 2007.

The challenge:

Days on which there were no homicides are not recorded as rows in the database.

date	number_of_homicides
2007-01-01	1
2007-01-02	2
2007-01-03	1
2007-01-05	3
2007-01-07	1
2007-01-08	2
2007-01-09	5
2007-01-13	1
2007-01-15	1
2007-01-18	1



The goal:

Use the homicides database to make a chart of the rolling 30-day sum of homicides since 2007.

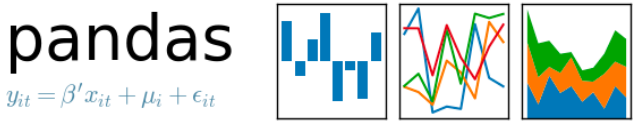
The challenge:

Days on which there were no homicides are not recorded as rows in the database.

The process:

1. Add a row for days on which there were no homicides, and attach a value of 0 for these rows.
2. Calculate a rolling 30-day sum of homicides since 2007.
3. Plot the data of rolling 30-day homicides.

0. Calculate the number of homicides by day



Read in the individual-level homicides data and group by date.

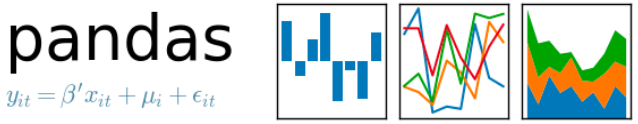
```
homicides = pd.read_csv('input/baltimore-homicide-victims_oct8_2018.csv')
homicides['date'] = pd.to_datetime(homicides['date_found'])
homicides['year'] = homicides['date'].dt.year
homicides = homicides[homicides['year'] >= 2007]

homicides_grouped = (homicides.groupby(['date'],
                                       as_index=False)['date_found']
                     .count())
homicides_grouped.rename(columns={"date_found": "number_of_homicides"}, inplace = True)

homicides_grouped.head()
```

	date	number_of_homicides
0	2007-01-01	1
1	2007-01-02	2
2	2007-01-03	1
3	2007-01-05	3
4	2007-01-07	1

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.core.resample.Resampler.fillna.html

rolling_homicides.ipynb

1. Add a row for days with no homicides




```
homicides_grouped.index = homicides_grouped['date']
filled = homicides_grouped.resample("1d").sum().fillna(0)
```

```
filled.head(15)
```

date	number_of_homicides
2007-01-01	1
2007-01-02	2
2007-01-03	1
2007-01-04	0
2007-01-05	3
2007-01-06	0
2007-01-07	1
2007-01-08	2
2007-01-09	5
2007-01-10	0
2007-01-11	0
2007-01-12	0
2007-01-13	1
2007-01-14	0
2007-01-15	1

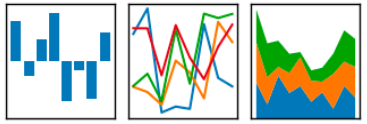
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.core.resample.Resampler.fillna.html

rolling_homicides.ipynb

2. Calculate a rolling 30-day sum of homicides

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



```
rolling = filled.rolling('30D', min_periods = 1).sum().reset_index()
```

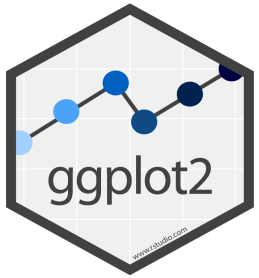
```
rolling.head(15)
```

	date	number_of_homicides
0	2007-01-01	1.0
1	2007-01-02	3.0
2	2007-01-03	4.0
3	2007-01-04	4.0
4	2007-01-05	7.0
5	2007-01-06	7.0
6	2007-01-07	8.0
7	2007-01-08	10.0
8	2007-01-09	15.0
9	2007-01-10	15.0
10	2007-01-11	15.0
11	2007-01-12	15.0
12	2007-01-13	16.0

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.rolling.html

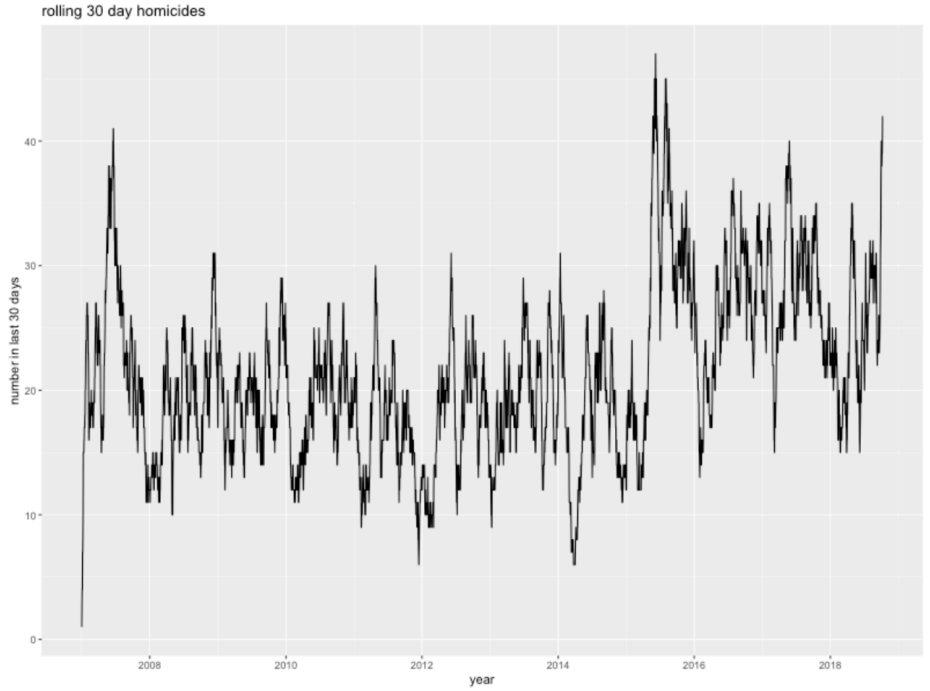
rolling_homicides.ipynb

3. Plot the data




```
%load_ext rpy2.ipython
```

```
%%R -i rolling -w 800 -h 600 -u px
library('ggplot2')

head(rolling)

ggplot(rolling, aes(x = date, y = number_of_homicides)) + geom_line() + labs(x = 'year',
                                                                             y = 'number in last 30 days',
                                                                             title = 'rolling 30 day homicides')
```

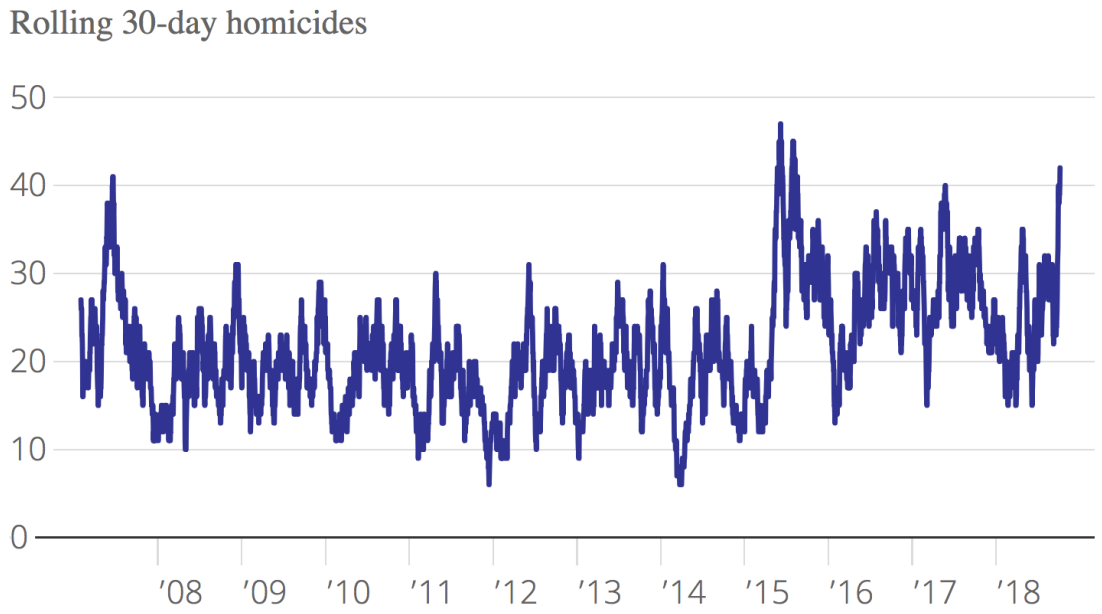


https://rpy2.bitbucket.io/
https://gist.github.com/simecek/019d87c55fec3839d95bbf8489dde61d

“The last 30 days in Baltimore have been the deadliest stretch since 2015” (by Sarah Meehan, Oct. 9, 2018)

The city saw similar high homicide spikes in 2007, 2015 and last year.

The below graph shows the number of homicides recorded over monthlong spans since early 2007. Each point is the sum of the previous 30 days' homicides.



In the story, the chart was made using Chartbuilder http://quartz.github.io/Chartbuilder/

https://www.baltimoresun.com/news/maryland/crime/bs-md-ci-crime-30-days-20181009-story.html

Three ways I use Python **and** R together

The same thing, but the other way around

Running Python inside an R script

The goal:

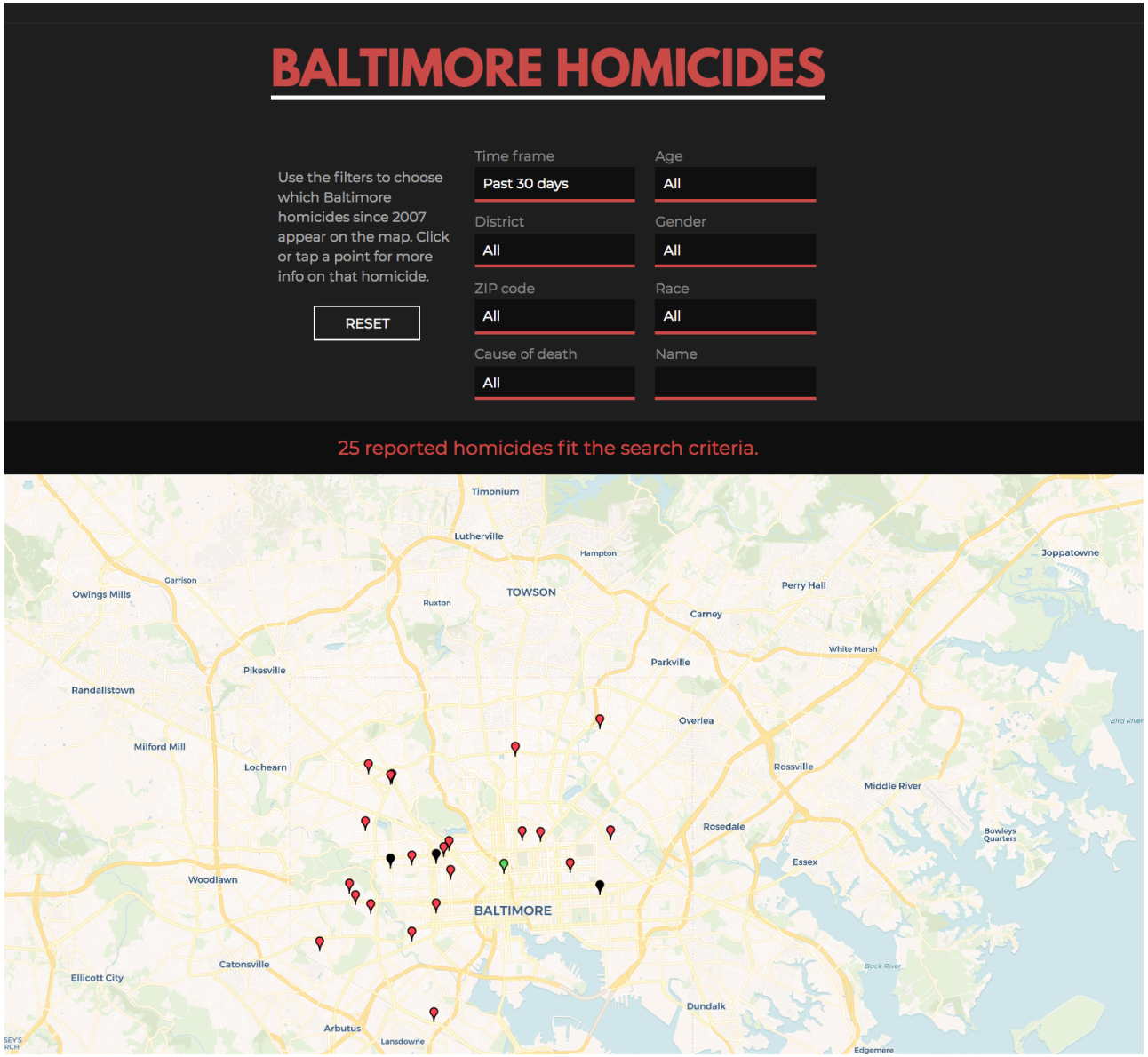
Use the homicides database to make a chart of the rolling 30-day sum of homicides since 2007.

The challenge:

Days on which there were no homicides are not recorded as rows in the database.

The process:

1. Add a row for days on which there were no homicides, and attach a value of 0 for these rows.
2. Calculate a rolling 30-day sum of homicides since 2007.
3. Plot the data of rolling 30-day homicides.



https://baltimoresun.com/homicides

first_name	last_name	age	gender	race	cause	death_loc	district	street_address	zip_code	latitude	longitude	date_found	time_found
George	Phillips IV	19	male	black	shooting	unknown	NE	2800 Lake Ave	21213	39.32411	-76.57386	2018-10-08	*NA*
Brailynn	Ford	0	female	black	unknown	unknown	WD	1000 Mosher St	21217	39.30042	-76.63535	2018-10-08	*NA*
NA	*NA*	*NA*	male	unknown	shooting	unknown	WD	1800 N Fulton Ave	21217	39.30895	-76.64670	2018-10-07	23:30:00
Harry	Reckline	53	male	white	shooting	unknown	SW	2600 Cole St	21223	39.27841	-76.65591	2018-10-05	*NA*
Lawrence	Price	73	male	black	shooting	unknown	NW	4600 Liberty Heights Ave	21207	39.33129	-76.69434	2018-10-04	00:24:00
Andrew Omar	Allen	18	male	black	shooting	unknown	??	3300 Cardenas Ave	21213	39.32295	-76.57566	2018-10-03	23:05:00
Fanny	Machado	28	female	white	unknown	scene	WD	1300 N Fulton Ave	21217	39.30338	-76.64649	2018-10-03	14:27:00
Jason Reuben	Haynes	41	male	black	shooting	sinai-hospital	NW	4300 Park Heights Ave	21215	39.33781	-76.66544	2018-10-02	22:43:00
Trevos	Agard	41	male	black	shooting	unknown	ND	400 Homeland Ave	21212	39.35403	-76.61411	2018-10-01	20:23:00
Garfield	Leon Jr.	30	male	black	shooting	scene	NW	3000 Oakley Ave	21215	39.34801	-76.67054	2018-10-01	13:18:00

0. Calculate the number of homicides by day

Read in individual-level homicides data and group by date.

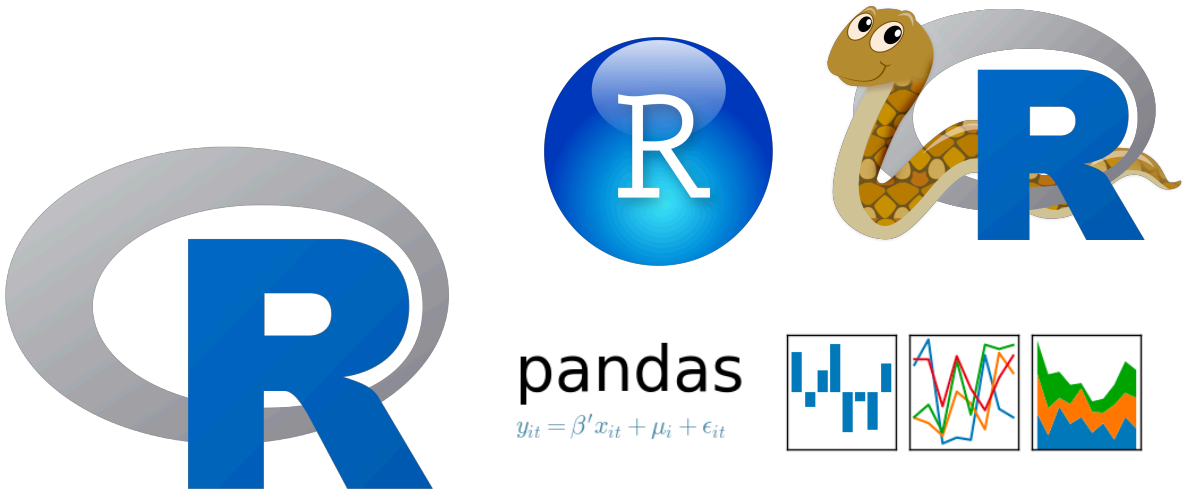
```{r, message=FALSE}
homicides <- read_csv('input/baltimore-homicide-victims_oct8_2018.csv')
homicides$date <- ymd(homicides$date_found)
homicides <- homicides %>% filter(year(date_found) >= 2007)
homicides_grouped <- homicides %>% group_by(date) %>% summarise(n = n())

head(homicides_grouped)
```

date <date>	n <int>
2007-01-01	1
2007-01-02	2
2007-01-03	1
2007-01-05	3
2007-01-07	1
2007-01-08	2

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.core.resample.Resampler.fillna.html

rolling_homicides.Rmd

1. Add a row for days with no homicides




```{python}
import pandas as pd
homicides_grouped = r.homicides_grouped
homicides_grouped['date'] = pd.to_datetime(homicides_grouped['date'])
homicides_grouped.index = homicides_grouped['date']
filled = homicides_grouped.resample("1d").sum().fillna(0)

filled.head()
```

```
            n
date
2007-01-01  1
2007-01-02  2
2007-01-03  1
2007-01-04  0
2007-01-05  3
```

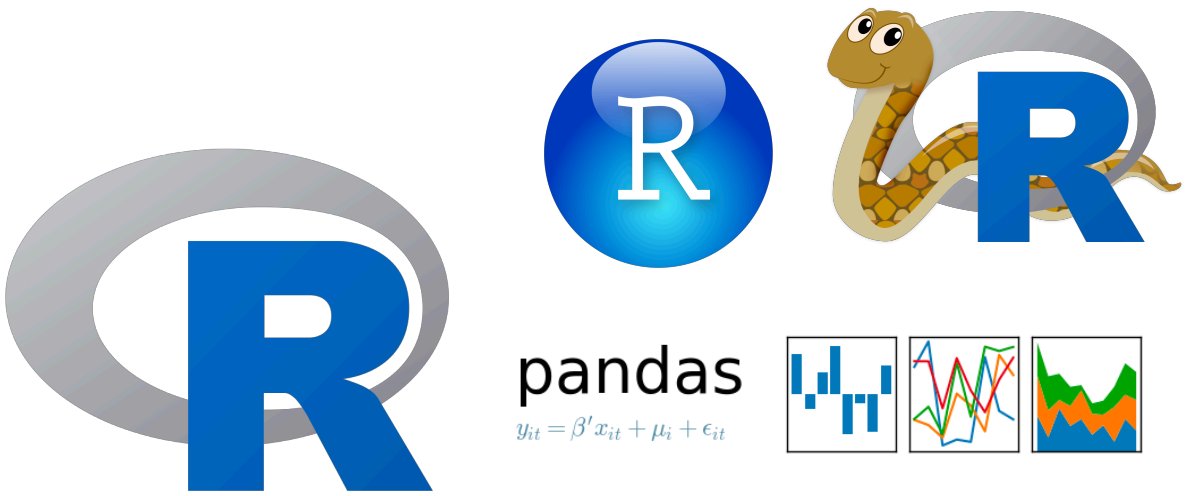
https://rstudio.github.io/reticulate/

https://www.rstudio.com/products/rstudio/download/preview/

→ Make sure you have the latest version of RStudio

rolling_homicides.Rmd

2. Calculate a rolling 30-day sum of homicides



Use pandas to calculate the rolling 30-day sum of homicides.

```{python}
rolling = filled.rolling('30D', min_periods = 1).sum().reset_index()
rolling.head()
```

	date	n
0	2007-01-01	1.0
1	2007-01-02	3.0
2	2007-01-03	4.0
3	2007-01-04	4.0
4	2007-01-05	7.0

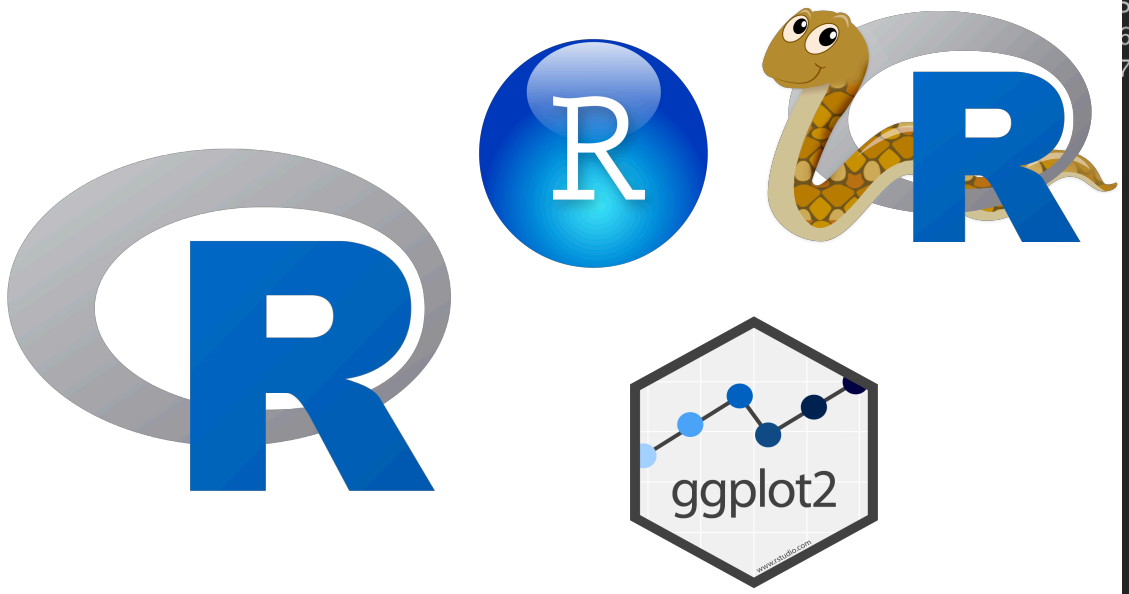
https://rstudio.github.io/reticulate/

https://www.rstudio.com/products/rstudio/download/preview/

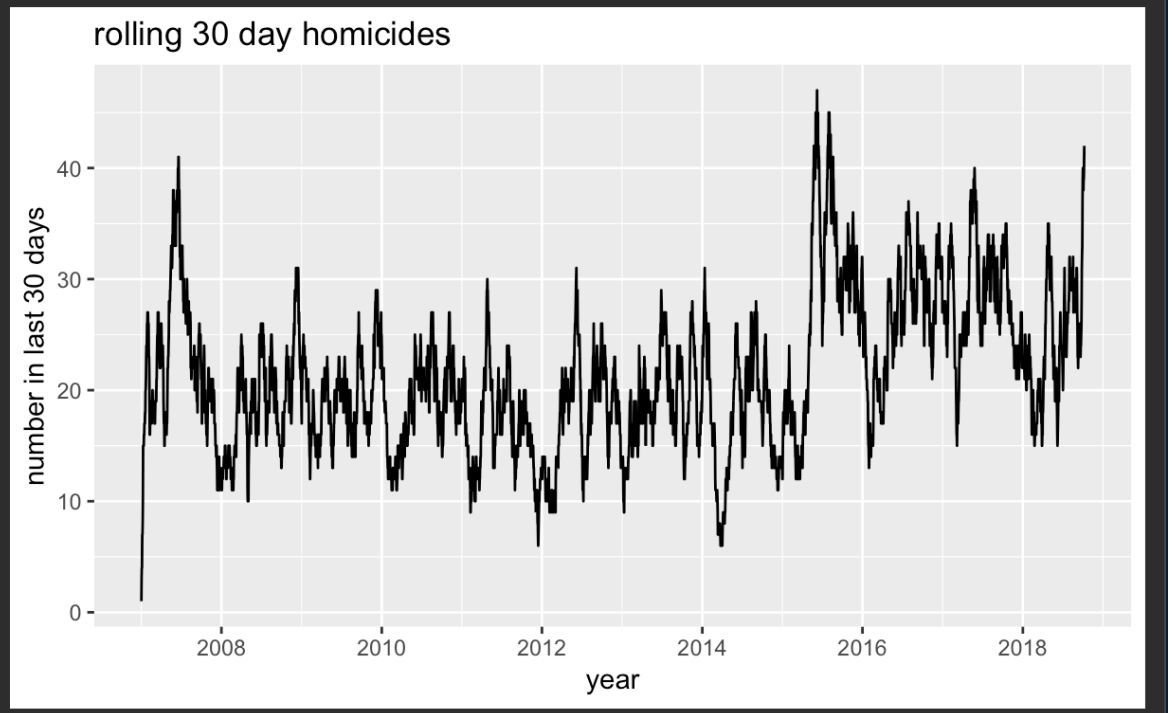
→ Make sure you have the latest version of RStudio

rolling_homicides.Rmd

3. Plot the data



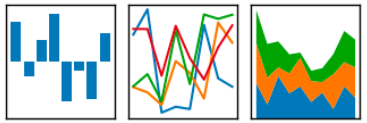

```r
ggplot(py$rolling, aes(x = date, y = n)) +
  geom_line() + labs(x = 'year',
                     y = 'number in last 30 days',
                     title = 'rolling 30 day homicides')
```



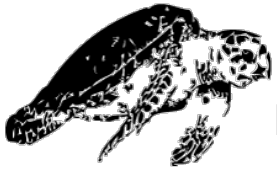
Thank you!

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

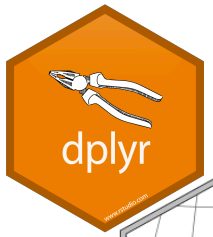


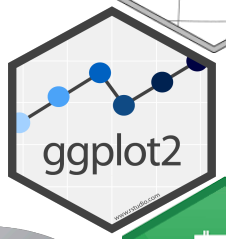
Beautiful Soup



Requests

jupyter





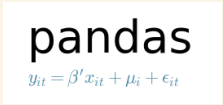
Glossary

Jupyter notebook: an interactive notebook environment, supporting multiple languages ("Jupyter" is named for Julia, Python, R).

RStudio: an integrated development environment for R.

rpy2: a Python package that allows for R access from within Python.

reticulate: an R package that allows for Python access from within R.



pandas: a Python library for data analysis.

requests: a Python library for making HTTP requests.

BeautifulSoup: a Python library for pulling data out of HTML and XML files ("scraping").

dplyr: an R library for data manipulation. Part of the tidyverse.

sf: an R library that provides simple features access (for spatial data).

ggplot2: an R library for creating graphics based on the "grammar of graphics." Part of the tidyverse.

lubridate: an R package for working with dates. Part of the tidyverse.