

## Parallel organization

The project is structured so that most scripts (except one) are in sub-directories, including `analysis`, `build` and `maps`.

There is some in progress stuff like `simulate`.

The rest is utilities, in `myutil`, and `test`.

The database is at main level, and it comes in in two sizes, one for as many days back as there are files at main level in the source, and the other with previous files stashed by month.

The average script starts like this:

```
import sys, os, subprocess
base = os.environ.get('covid_base')
sys.path.insert(0,base)
```

Thus, you must set `covid_base` correctly. Everything is specified as a path from `covid_base`.

## Command line arguments

These can be viewed with `-h` or `--help` with any script.

Features that are currently supported are given by the `--help` flag:

```

> python scripts/one_state.py --help

flags
-h  --help      help
-n  <int>       display the last -n values, default: 7
-N  <int>       display -N rows of data, default: no limit
-c  <int>       --delta, change from x days ago, default: 1

-a  --all       use the complete db, starting 2020-03-22
-d  --deaths    display deaths rather than the default, cases
-g  --graph     plot a graph of the data
-m  --map       make a choropleth map
-p  --pop       normalize to population
-r  --rate      compute statistics (currently, over last 7 days)
-s  --sort
-t  --totals    (only)
-v  --verbose   debugging mode
-w, --write     text (if -g,-m present, output is normally silent)

to do:
-u  <int>       data slice ends this many days before yesterday

example:
> python one_state.py <state> -n 10 -sdr
>

```

I did not use the built-in Python module for parsing the command line arguments, but rolled my own, see `uinit.py`

The statistic is the slope of a linear regression, divided by the mean of the values.

So, for example, if a 10-day series goes smoothly from 100 to 110, then the slope is about  $10/10 = 1$  and the statistic is a bit less than 0.01. If the series goes from 1000 to 1100, then the slope is about  $100/10 = 10$ , but the statistic is still approximately 0.01.

## Approach

The idea for most scripts is to use the main part of the script to assemble the correct keys in order. This list is passed to `ucalc` and then to `ufmt` along with the `conf` dictionary.

All the trimming, sorting and stats happens in `ucalc`.

All the formatting happens in `ufmt`.

The code about keys does not know which database we're using. I found that too complicated to maintain since I added the option of building a `max` database.

So now the database is passed to `ukeys` functions as an argument.

## Examples (as of 2020-06-29)

This version of the database is 2020-06-01 to 2020-06-28, updated this morning.

From the `analysis` directory:

```
> python one_state.py SC -rs -n 3 -N 5
      06/26 06/27 06/28  stats
Bamberg      83    84    91  0.047
Anderson    525   552   563  0.035
Aiken       325   335   345  0.03
Abbeville   100   103   103  0.015
Allendale   47    48    48  0.01
total      1080  1122  1150  0.031
>
```

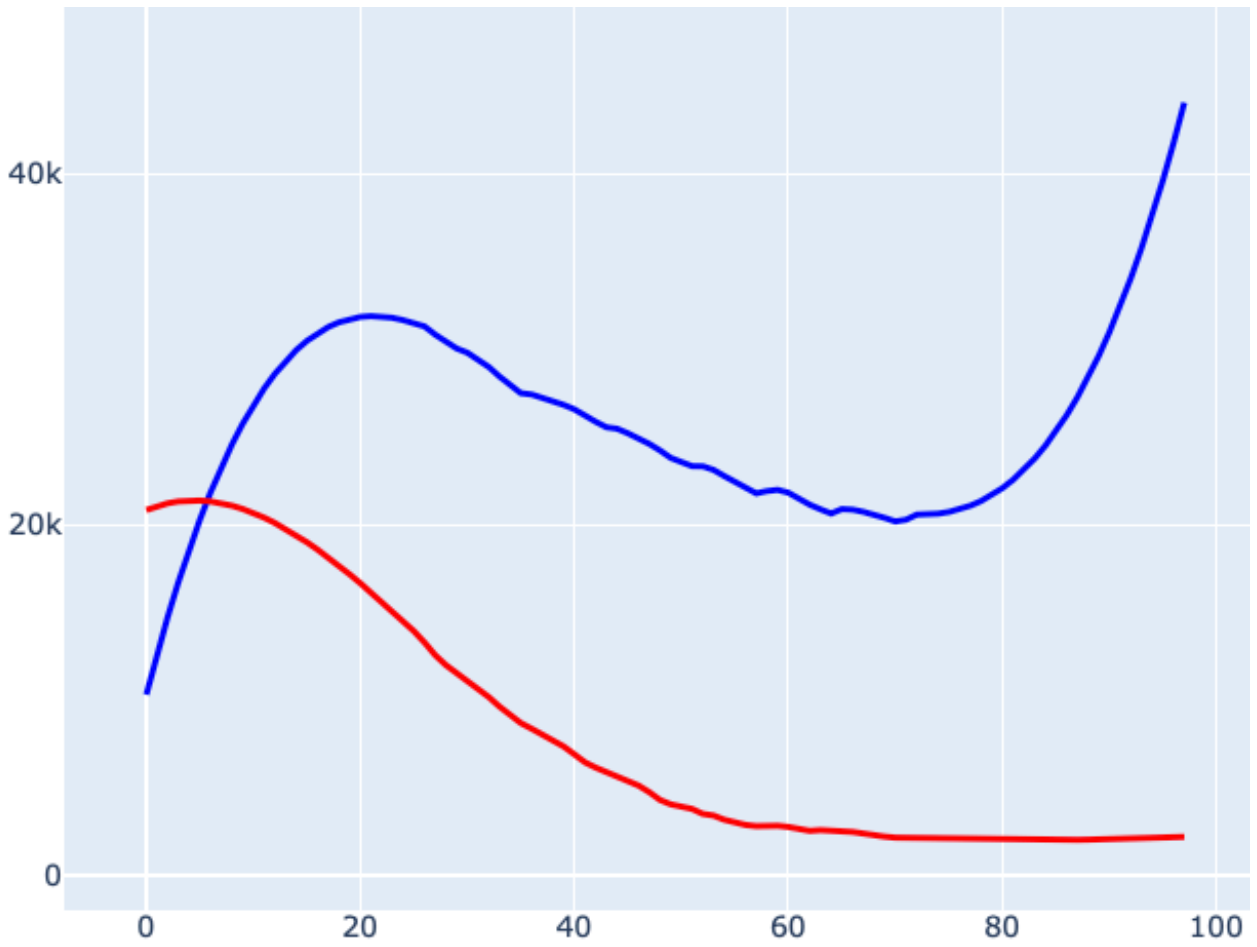
and

```
> python country.py Italy -n 3
```

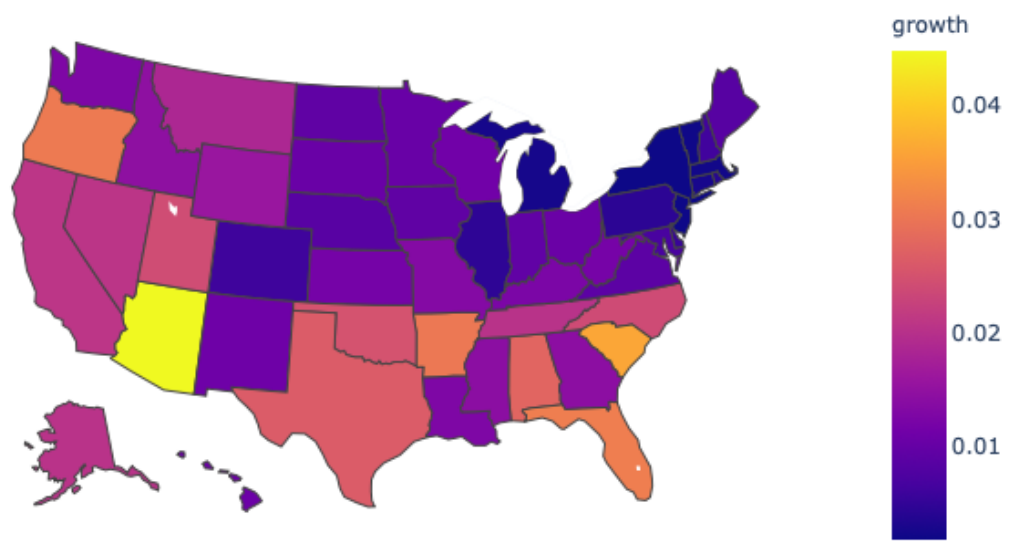
	06/26	06/27	06/28
Abruzzo, Italy	3285	3285	3286
Basilicata, Italy	401	401	401
Calabria, Italy	1178	1179	1180
Campania, Italy	4665	4665	4665
Emilia-Romagna, Italy	28393	28435	28456
Friuli Venezia Giulia, Italy	3307	3307	3308
Lazio, Italy	8064	8082	8096
Liguria, Italy	9958	9963	9967
Lombardia, Italy	93587	93664	93761
Marche, Italy	6783	6785	6785
Molise, Italy	445	445	445
P.A. Bolzano, Italy	2634	2636	2637
P.A. Trento, Italy	4859	4860	4863
Piemonte, Italy	31311	31322	31336
Puglia, Italy	4531	4531	4531
Sardegna, Italy	1362	1363	1364
Sicilia, Italy	3076	3077	3077
Toscana, Italy	10226	10238	10243
Umbria, Italy	1440	1440	1440
Valle d'Aosta, Italy	1194	1194	1194
Veneto, Italy	19262	19264	19275
total	239961	240136	240310

Results from `plot_eu_us.py`

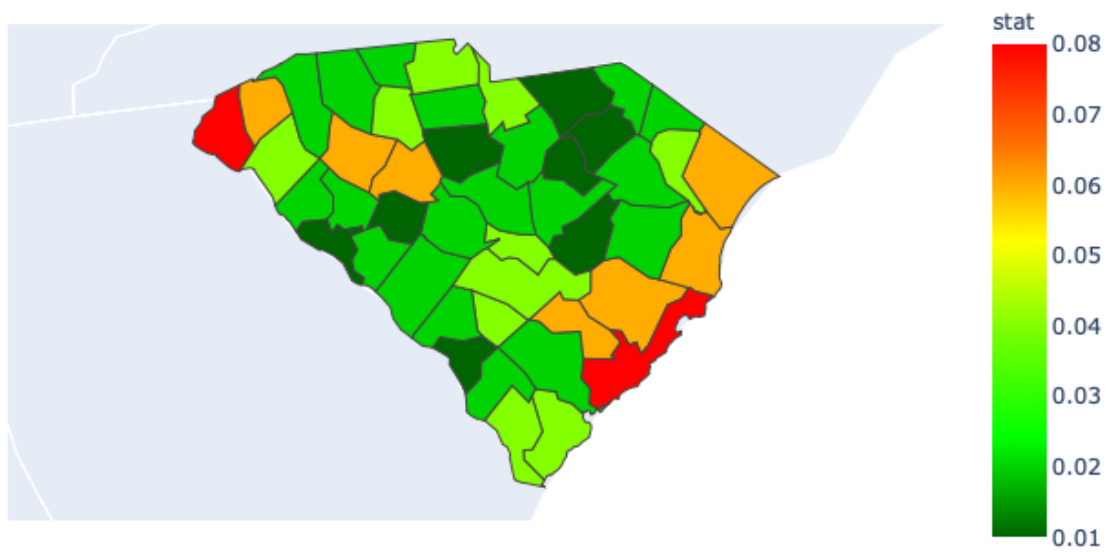
US v. EU new cases:



Choropleth 2020-06-19



and 2020-06-27



China new cases [2020-06-27](#).

```
python3 geo/one_state_map.py CA MN SC TX WY KY
```

