

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 analyze.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 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`, and all the output 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 2020-07-09)

```
python3 analyze.py US -c
```

	07/02	07/03	07/04	07/05	07/06	07/07	07/08
Alabama	1162	1758	997	1091	925	888	1161
Alaska	39	46	47	27	28	18	42
Arizona	3340	4427	2695	3536	3352	3639	3520
Arkansas	847	501	483	585	381	310	703
California	7869	3964	2381	11786	6354	12977	8548
Colorado	323	260	254	199	192	407	452
Connecticut	72	69	0	0	245	57	74
Delaware	210	176	69	124	155	114	45
District of Columbia	25	45	12	35	33	54	73
Florida	10104	9473	11444	10034	6332	7346	9969
Georgia	2512	3376	3184	1835	1672	2749	2852
Guam	13	0	0	0	21	2	4
Hawaii	20	28	24	24	7	41	23
Idaho	223	400	377	363	319	487	430
Illinois	0	1742	861	632	612	588	986
Indiana	435	528	517	576	323	295	437
Iowa	742	204	589	456	240	396	467
Kansas	172	657	252	170	557	222	472
Kentucky	237	297	0	0	779	368	399
Louisiana	1381	1726	0	1936	1101	1911	1921
Maine	30	46	26	18	9	14	23
Maryland	505	538	380	291	272	492	465
Massachusetts	457	285	211	136	162	199	260
Michigan	578	474	395	343	302	617	646
Minnesota	493	420	0	511	432	557	454
Mississippi	870	914	990	226	357	957	674
Missouri	1600	525	423	233	394	707	672
Montana	67	45	39	45	37	78	44
Nebraska	152	214	161	100	119	195	227
Nevada	632	985	857	843	491	876	516
New Hampshire	20	34	16	22	18	20	20
New Jersey	399	400	270	363	199	280	187
New Mexico	243	241	286	193	249	219	285
New York	875	918	726	533	518	588	692
North Carolina	1465	2046	1408	1322	1783	1515	1397

North Dakota	42	65	57	37	33	49	73
Northern Mariana Islands	1	0	0	0	0	0	0
Ohio	0	2392	926	967	806	948	1277
Oklahoma	412	538	579	283	431	858	674
Oregon	363	342	294	300	165	210	212
Pennsylvania	837	780	530	481	697	798	827
Puerto Rico	71	75	104	129	669	129	31
Rhode Island	0	274	0	0	0	0	0
South Carolina	1782	1831	1854	1461	1533	972	1557
South Dakota	67	85	50	35	42	58	79
Tennessee	1519	1701	1275	1155	710	1482	2238
Texas	6769	6454	6562	2779	10710	10384	8903
Utah	554	596	676	410	517	564	722
Vermont	18	9	2	11	2	3	2
Virgin Islands	2	6	13	0	1	4	6
Virginia	532	658	716	639	354	638	635
Washington	799	598	501	596	1041	505	500
West Virginia	74	73	79	57	180	63	202
Wisconsin	539	579	738	522	484	495	598
Wyoming	36	32	24	28	41	34	31
total	52529	54850	45354	48478	47386	58377	57707

running:

python3 analyze.py US -rs -N 3

	07/02	07/03	07/04	07/05	07/06	07/07	07/08
Idaho	6592	6992	7369	7732	8051	8538	8968
0.05							
Florida	168934	178407	189851	199885	206217	213563	223532
0.045							
Virgin Islands	92	98	111	111	112	116	122
0.042							
total	2720164	2775014	2820368	2868846	2916232	2974609	3032316
0.018							

running:

python3 analyze.py HI -rs

	07/02	07/03	07/04	07/05	07/06	07/07	07/08
Honolulu, HI	676	701	720	744	750	788	808 0.029
Kauai, HI	38	38	40	40	40	42	42 0.018
Hawaii, HI	90	91	93	93	94	95	96 0.01
Maui, HI	125	127	128	128	128	128	130 0.005
total	929	957	981	1005	1012	1053	1076 0.024

running:

```
python3 analyze.py SC -N 3
```

	07/02	07/03	07/04	07/05	07/06	07/07	07/08
Abbeville, SC	119	118	119	124	135	134	137
Aiken, SC	434	451	488	507	516	530	545
Allendale, SC	58	58	58	61	64	64	64
total	39701	41532	43386	44847	46380	47352	48909

running:

```
python3 analyze.py Mexico -c -N 3
```

	07/02	07/03	07/04	07/05	07/06	07/07	07/08
Aguascalientes	36	16	23	2	90	94	107
Baja California	202	216	141	74	38	251	143
Baja California Sur	53	49	64	40	27	86	40
total	6741	6740	6914	4683	4902	6258	6995

running:

```
python3 analyze.py Switzerland
```

	07/02	07/03	07/04	07/05	07/06	07/07	07/08
Switzerland	31967	32101	32198	32268	32315	32369	32498
total	31967	32101	32198	32268	32315	32369	32498

running:

```
python3 analyze.py Germany -N 3 -o
```

	07/02	07/03	07/04	07/05	07/06	07/07	07/08
Germany	195359	195817	196190	196413	196708	197071	197485

running:

```
python3 analyze.py Russia -p -q
```

KeyError, key not in pop_dict:

;Adygea Republic;;Russia

running:

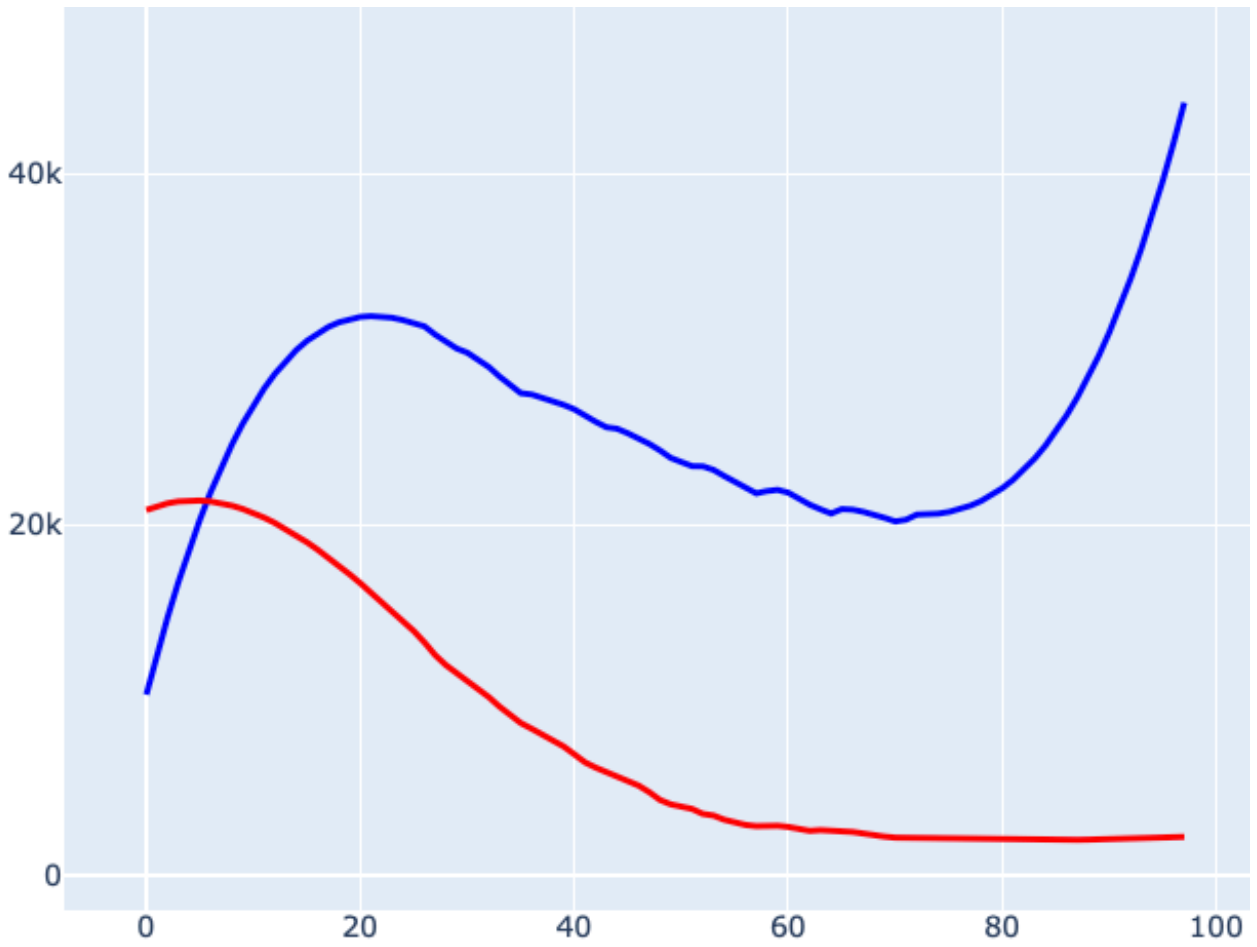
```
python3 analyze.py counties -s -c 10 -N 30
```

	07/02	07/03	07/04	07/05	07/06	07/07	07/08
Los Angeles, CA	21775	19294	18193	23447	23179	25379	25338
Maricopa, AZ	23107	24046	24837	25161	25289	25169	25166
Miami-Dade, FL	14026	15489	16950	18347	18796	19496	20260
Harris, TX	11061	9687	8322	7092	9522	10149	10736
Dallas, TX	5291	5931	6643	7302	8020	8536	9565

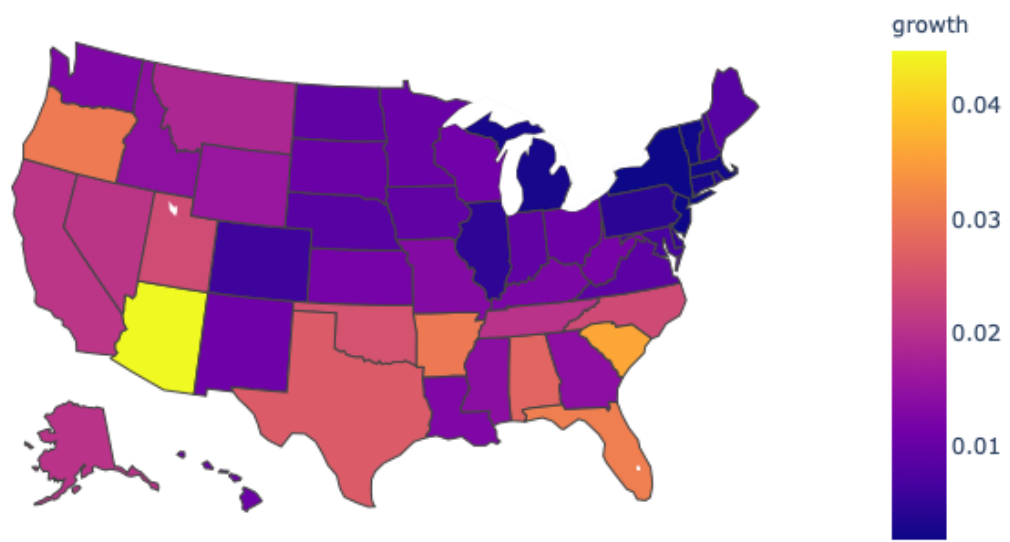
Broward, FL	5789	6485	7358	8655	8536	8549	9161
Orange, CA	4470	5041	5175	5343	5922	6430	7617
Bexar, TX	5622	5722	6745	6739	6650	7023	6578
Riverside, CA	4920	5019	4545	4308	4912	5458	6314
Clark, NV	5406	6254	6933	6564	6614	6480	6216
Hillsborough, FL	6403	6868	6808	7007	6659	6232	6181
Orange, FL	6301	6576	7206	7246	6559	5931	5523
San Bernardino, CA	3791	3666	3269	4153	4055	4642	5085
Duval, FL	4162	4759	5195	5422	5316	4987	4851
Palm Beach, FL	3916	4144	4613	4996	4744	4710	4842
Travis, TX	3914	4356	4413	4217	4582	4829	4583
San Diego, CA	4111	4402	4070	4765	4325	4741	4508
Tarrant, TX	3995	4037	4037	3060	4122	4502	4109
Cook, IL	3487	3990	4102	3882	3713	3646	3741
Davidson, TN	2409	2583	2664	2831	2658	2924	3559
Shelby, TN	2701	2877	3070	3150	2948	3037	3410
Pinellas, FL	3395	3664	3729	3864	3660	3319	3375
Nueces, TX	2385	2530	2459	2565	2797	2973	3190
New York City, NY	3161	3306	3382	3270	3146	3054	3027
Pima, AZ	2990	2935	2995	3075	3037	3098	2960
Charleston, SC	2571	2782	2959	3022	3118	2931	2956
Lee, FL	2557	2923	3271	3436	2904	2903	2937
Mecklenburg, NC	2847	3147	3243	3192	3167	3030	2897
Hidalgo, TX	2563	2638	2668	2295	2632	2555	2890
Fresno, CA	2141	1954	1959	2057	1736	2450	2807
total	421548	441569	452345	461094	463226	479246	499427

Results from `plot_eu_us.py`

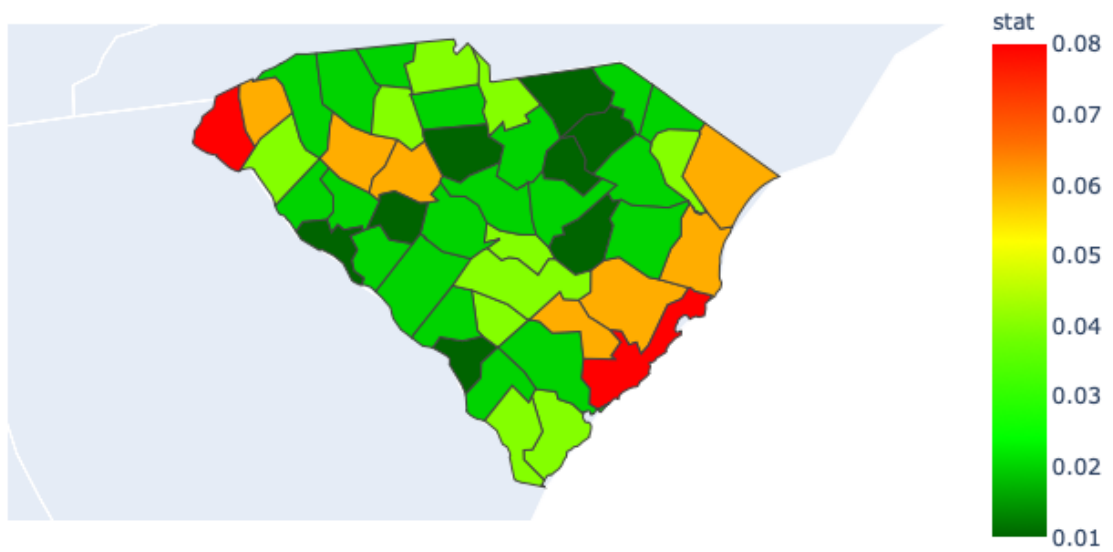
US v. EU new cases:



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
```

