Deepnote

Johnny / time_series Published at Oct 31, 2022 Unlisted

```
pip install covidcast
```

```
Requirement already satisfied: covidcast in /usr/local/lib/python3.9/site-packages (0.1.5)
Requirement already satisfied: delphi-epidata>=0.0.11 in /usr/local/lib/python3.9/site-packages (from covid
Requirement already satisfied: tqdm in /shared-libs/python3.9/py/lib/python3.9/site-packages (from covidcas
Requirement already satisfied: pandas in /shared-libs/python3.9/py/lib/python3.9/site-packages (from covidc
Requirement already satisfied: epiweeks in /usr/local/lib/python3.9/site-packages (from covidcast) (2.1.4)
Requirement already satisfied: geopandas in /shared-libs/python3.9/py/lib/python3.9/site-packages (from cov
Requirement already satisfied: numpy in /shared-libs/python3.9/py/lib/python3.9/site-packages (from covidca
Requirement already satisfied: imageio in /usr/local/lib/python3.9/site-packages (from covidcast) (2.22.2)
Requirement already satisfied: requests in /shared-libs/python3.9/py/lib/python3.9/site-packages (from covi
Requirement already satisfied: descartes in /usr/local/lib/python3.9/site-packages (from covidcast) (1.1.0)
Requirement already satisfied: matplotlib in /shared-libs/python3.9/py/lib/python3.9/site-packages (from co
Requirement already satisfied: imageio-ffmpeg in /usr/local/lib/python3.9/site-packages (from covidcast) (0
Requirement already satisfied: tenacity in /shared-libs/python3.9/py/lib/python3.9/site-packages (from delp
Requirement already satisfied: aiohttp in /shared-libs/python3.9/py-core/lib/python3.9/site-packages (from
Requirement already satisfied: charset-normalizer<3,>=2 in /shared-libs/python3.9/py-core/lib/python3.9/sit
Requirement already satisfied: idna<4,>=2.5 in /shared-libs/python3.9/py-core/lib/python3.9/site-packages (
Requirement already satisfied: certifi>=2017.4.17 in /shared-libs/python3.9/py/lib/python3.9/site-packages
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /shared-libs/python3.9/py/lib/python3.9/site-packag
Requirement already satisfied: packaging in /shared-libs/python3.9/py-core/lib/python3.9/site-packages (fro
Requirement already satisfied: pyproj>=2.6.1.post1 in /shared-libs/python3.9/py/lib/python3.9/site-packages
Requirement already satisfied: fiona>=1.8 in /shared-libs/python3.9/py/lib/python3.9/site-packages (from ge
Requirement already satisfied: shapely<2,>=1.7 in /shared-libs/python3.9/py/lib/python3.9/site-packages (fr
Requirement already satisfied: python-dateutil>=2.7.3 in /shared-libs/python3.9/py-core/lib/python3.9/site-
Requirement already satisfied: pytz>=2017.3 in /shared-libs/python3.9/py/lib/python3.9/site-packages (from
Requirement already satisfied: pillow>=8.3.2 in /shared-libs/python3.9/py/lib/python3.9/site-packages (from
Requirement already satisfied: cycler>=0.10 in /shared-libs/python3.9/py/lib/python3.9/site-packages (from
Requirement already satisfied: fonttools>=4.22.0 in /shared-libs/python3.9/py/lib/python3.9/site-packages (
Requirement already satisfied: contourpy>=1.0.1 in /shared-libs/python3.9/py/lib/python3.9/site-packages (f
Requirement already satisfied: pyparsing>=2.2.1 in /shared-libs/python3.9/py-core/lib/python3.9/site-packag
Requirement already satisfied: kiwisolver>=1.0.1 in /shared-libs/python3.9/py/lib/python3.9/site-packages (
```

```
from datetime import date
import covidcast
```

Date may have a wider range later on for more training data

```
start = date(2020, 3, 1)
end = date(2021, 3, 1)
```

```
CA_counties_to_fips = covidcast.fips_to_name('^06.*', ties_method='all')
CA_counties_to_fips = {value[0]: key for key, value in CA_counties_to_fips[0].items()}

# CA_counties_to_fips
```

```
CA_counties = list(CA_counties_to_fips.keys())
CA_counties = covidcast.name_to_fips(CA_counties)[1:]

# CA_counties
```

```
/usr/local/lib/python3.9/site-packages/covidcast/geography.py:314: UserWarning: Some inputs were not uniquel
  warnings.warn("Some inputs were not uniquely matched; returning only the first match "
```

Indicator Combination: ground truth

```
indicator_combination = covidcast.signal(
    data_source='indicator-combination',
    signal='confirmed_incidence_num',
    start_day=start, end_day=end, geo_values=CA_counties
)

indicator_combination = indicator_combination.drop([0,1])
indicator_combination['geo_value'].unique()
```

```
array(['06001', '06003', '06005', '06007', '06009', '0601
       '06015', '06017', '06019', '06021', '06023', '0602
       '06029', '06031', '06033', '06035', '06037', '0603
       '06043', '06045', '06047', '06049', '06051', '0605
       '06059', '06061', '06063', '06065', '06067', '0606
       '06073', '06075', '06077', '06079', '06081', '0608
       '06089', '06091', '06093', '06095', '06097', '0609
       '06103', '06105', '06107', '06109', '06111', '0611
      dtype=object)
```

Change Healthcare: % of confirmed cases at doctor visit

```
change_health = covidcast.signal(
    data_source='chng',
    signal='smoothed_adj_outpatient_covid',
    start_day=start, end_day=end, geo_values=CA_counties
)
```

```
change_health = change_health.drop([0,1])
change_health['geo_value'].unique()
```

```
array(['06001', '06005', '06007', '06009', '06011', '0601
       '06017', '06019', '06021', '06023', '06025', '0602
       '06031', '06033', '06035', '06037', '06039', '0604
       '06045', '06047', '06049', '06051', '06053', '0605
       '06061', '06063', '06065', '06067', '06069', '0607
       '06075', '06077', '06079', '06081', '06083', '0608
       '06091', '06093', '06095', '06097', '06099', '0610
       '06105', '06107', '06109', '06111', '06113', '0611
```

Hospital Admissions: % of new hospital admissions with COVID-associated diagnoses, based on claims data from health system partners, smoothed in time using a Gaussian linear smoother

```
hospital_admit = covidcast.signal(
    data_source='hospital-admissions',
    signal='smoothed_adj_covid19_from_claims',
    start_day=start, end_day=end, geo_values=CA_counties
)

hospital_admit['geo_value'].unique()
```

```
array(['06001', '06013', '06029', '06037', '06059', '0606
       '06067', '06071', '06073', '06075', '06081', '0608
       '06083', '06077', '06019', '06031', '06099', '0604
       '06041', '06079', '06097', '06053', '06107', '0402
       '06113', '06017'], dtype=object)
```

```
hospital_admit = hospital_admit[hospital_admit['geo_value']!='04023']
hospital_admit['geo_value'].unique()
```

```
array(['06001', '06013', '06029', '06037', '06059', '0606
       '06067', '06071', '06073', '06075', '06081', '0608
       '06083', '06077', '06019', '06031', '06099', '0604
       '06041', '06079', '06097', '06053', '06107', '0606
       '06017'], dtype=object)
```

Doctor Visits: % of confirmed cases at doctor visit (comes from another source)

```
doc_visits = covidcast.signal(
    data_source="doctor-visits",
    signal="smoothed_adj_cli",
    start_day=start, end_day=end, geo_values=CA_counties
)

doc_visits = doc_visits.drop([0])
doc_visits['geo_value'].unique()
```

```
array(['06001', '06005', '06007', '06011', '06013', '0601
       '06023', '06025', '06029', '06031', '06037', '0603
       '06045', '06047', '06053', '06055', '06059', '0606
       '06067', '06069', '06071', '06073', '06075', '0607
       '06081', '06083', '06085', '06089', '06095', '0609
       '06101', '06107', '06111', '06113', '06009', '0603
       '06103', '06109', '06043', '06093', '06021', '0606
       '06035'], dtype=object)
```

Mobility data

```
restaurants_prop = covidcast.signal(
    data_source="safegraph",
    signal="restaurants_visit_prop",
    start_day=start, end_day=end, geo_values=CA_counties
)

restaurants_prop = restaurants_prop.drop([0])
restaurants_prop['geo_value'].unique()
```

```
/usr/local/lib/python3.9/site-packages/covidcast/covidcast.py:423: NoDataWarning: No safegraph restaurants_
  warnings.warn(f"No {data_source} {signal} data found on {day_str} "
/usr/local/lib/python3.9/site-packages/covidcast/covidcast.py:423: NoDataWarning: No safegraph restaurants_
  warnings.warn(f"No {data_source} {signal} data found on {day_str} "
/usr/local/lib/python3.9/site-packages/covidcast/covidcast.py:423: NoDataWarning: No safegraph restaurants_
```

```
  warnings.warn(f"No {data_source} {signal} data found on {day_str} "
/usr/local/lib/python3.9/site-packages/covidcast/covidcast.py:423: NoDataWarning: No safegraph restaurants_\
  warnings.warn(f"No {data_source} {signal} data found on {day_str} "
/usr/local/lib/python3.9/site-packages/covidcast/covidcast.py:423: NoDataWarning: No safegraph restaurants_\
  warnings.warn(f"No {data_source} {signal} data found on {day_str} "
/usr/local/lib/python3.9/site-packages/covidcast/covidcast.py:423: NoDataWarning: No safegraph restaurants_\
  warnings.warn(f"No {data_source} {signal} data found on {day_str} "
/usr/local/lib/python3.9/site-packages/covidcast/covidcast.py:423: NoDataWarning: No safegraph restaurants_\
  warnings.warn(f"No {data_source} {signal} data found on {day_str} "
```

```
array(['06001', '06003', '06005', '06007', '06009', '06011', '06013',
       '06015', '06017', '06019', '06021', '06023', '06025', '06027',
       '06029', '06031', '06033', '06035', '06037', '06039', '06041',
       '06045', '06047', '06051', '06053', '06055', '06059', '06061',
       '06063', '06065', '06067', '06069', '06071', '06073', '06075',
       '06077', '06079', '06081', '06083', '06085', '06089', '06091',
       '06093', '06095', '06097', '06099', '06101', '06103', '06105',
       '06107', '06109', '06111', '06113', '06115', '06043'], dtype=object)
```

Merge

```
# df_list = [change_health, hospital_admit, doc_visits, restaurants_prop, indicator_combinati
df_list = [change_health, hospital_admit, doc_visits, indicator_combination]

merged = covidcast.aggregate_signals(df_list)
```

```
import numpy as np

merged = merged.rename(
    columns={
        'chng_smoothed_adj_outpatient_covid_0_value': 'change_health',
        'hospital-admissions_smoothed_adj_covid19_from_claims_1_value': 'hospital_admit',
#          'fb-survey_smoothed_cli_3_value': 'survey',
        'doctor-visits_smoothed_adj_cli_2_value': 'doc_visits',
#          'safegraph_restaurants_visit_prop_3_value': 'restaurants_prop',
        'indicator-combination_confirmed_incidence_num_3_value': 'ground_truth'
    }
)

# keep_list = ['geo_value', 'time_value',
#              'change_health', 'hospital_admit',
#              'doc_visits', 'restaurants_prop', 'ground_truth']
keep_list = ['geo_value', 'time_value',
             'change_health', 'hospital_admit',
             'doc_visits', 'ground_truth']
merged = merged[keep_list]
```

```
merged.loc[:, 'ground_truth'] = merged.loc[:, 'ground_truth'].abs()
np.sort(merged['ground_truth'].unique())
```

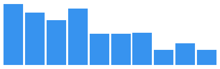
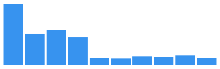
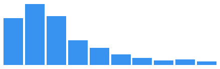
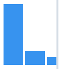
```
array([0.0000e+00, 1.0000e+00, 2.0000e+00, ..., 2.1902e+0
       2.8549e+04])
```

Missing values are caused by different sources of data having different counties they keep track of. We decided to find the average of the respective column values for every day and give the NaN values the value of the average.

```
# for every day, we took the mean values of every column with values of that day
# and gave the NaN values their respective mean values for that day
for date in merged['time_value'].unique():
    change_mean = merged[merged['time_value']==date]['change_health'].mean()
    hosp_mean = merged[merged['time_value']==date]['hospital_admit'].mean()
#     survey_mean = merged[merged['time_value']==date]['survey'].mean()
    doc_mean = merged[merged['time_value']==date]['doc_visits'].mean()
#     rest_mean = merged[merged['time_value']==date]['restaurants_prop'].mean()
    ground_mean = int(merged[merged['time_value']==date]['ground_truth'].mean())


    merged.loc[merged['time_value']==date, 'change_health'] = merged.loc[merged['time_value']]
    merged.loc[merged['time_value']==date, 'hospital_admit'] = merged.loc[merged['time_value'
#     merged.loc[merged['time_value']==date, 'survey'] = merged.loc[merged['time_value']==dat
    merged.loc[merged['time_value']==date, 'doc_visits'] = merged.loc[merged['time_value']==c
#     merged.loc[merged['time_value']==date, 'restaurants_prop'] = merged.loc[merged['time_va
    merged.loc[merged['time_value']==date, 'ground_truth'] = merged.loc[merged['time_value']]=

merged = merged.sort_values(['time_value', 'geo_value'])
merged[merged['geo_value']=='06001']
```

	geo_value object	time_value dateti...	change_health fl...	hospital_admit fl...	doc_visits float64	grou
	06001 100%	2020-03-01 00:00... 	0.0078964 - 1.31... 	0.088433 - 11.71... 	0.0 - 25.733473 	0.0 - 
0	06001	2020-03-01 00:00:00	0.0374813	0.119646	0.0	
56	06001	2020-03-02 00:00:00	0.0078964	0.119067	0.0	
112	06001	2020-03-03 00:00:00	0.0084559	0.119366	0.0	
168	06001	2020-03-04 00:00:00	0.0083222	0.119776	0.019267	
224	06001	2020-03-05 00:00:00	0.008752	0.119895	0.016927	
280	06001	2020-03-06	0.0088013	0.11982	0.014341	

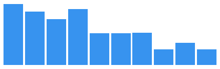
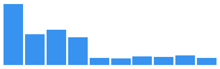
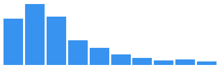
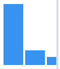
336	06001	2020-03-07 00:00:00	0.0358777	0.185543	0.011658	
392	06001	2020-03-08 00:00:00	0.0772792	0.245598	0.019996	
448	06001	2020-03-09 00:00:00	0.0156716	0.298664	0.006554	
504	06001	2020-03-10 00:00:00	0.0146484	0.343336	0.018026	

```
data_shift = len(merged['geo_value'].unique())
# today_list = ['change_health', 'hospital_admit', 'doc_visits', 'restaurants_prop']
# yesterday_list = ['change_health-1', 'hospital_admit-1', 'doc_visits-1', 'restaurants_prop-
today_list = ['change_health', 'hospital_admit', 'doc_visits']
yesterday_list = ['change_health-1', 'hospital_admit-1', 'doc_visits-1']

# before_yesterday_list = ['change_health-2', 'hospital_admit-2', 'doc_visits-2', 'restaurant

merged['ground_truth+1'] = merged['ground_truth'].shift(-1*data_shift)
for today, yesterday, in zip(today_list, yesterday_list):
    merged[yesterday] = merged[today].shift(data_shift)
#     merged[before_yesterday] = merged[today].shift(2*data_shift)

time_series = merged.dropna()
time_series[time_series['geo_value']=='06001']
```

	geo_value object	time_value dateti...	change_health fl...	hospital_admit fl...	doc_visits float64	grou
		2020-03-02 00:00...	0.0078964 - 1.31...	0.088433 - 11.71...	0.0 - 25.733473	0.0 -
	06001 .... 100%					
56	06001	2020-03-02 00:00:00	0.0078964	0.119067	0.0	
112	06001	2020-03-03 00:00:00	0.0084559	0.119366	0.0	
168	06001	2020-03-04 00:00:00	0.0083222	0.119776	0.019267	
224	06001	2020-03-05 00:00:00	0.008752	0.119895	0.016927	
280	06001	2020-03-06 00:00:00	0.0088013	0.11982	0.014341	
336	06001	2020-03-07 00:00:00	0.0358777	0.185543	0.011658	
392	06001	2020-03-08 00:00:00	0.0772792	0.245598	0.019996	
448	06001	2020-03-09 00:00:00	0.0156716	0.298664	0.006554	
504	06001	2020-03-10 00:00:00	0.0146484	0.343336	0.018026	

560	06001	2020-03-11 00:00:00	0.0134069	0.377819	0.087875

```
# export as a csv
# import pandas as pd
# compression_opts = dict(method='zip',
#                         archive_name='time_series.csv')
# time_series.to_csv('time_series.zip', index=False,
#           compression=compression_opts)
time_series.to_csv('time_series.csv', index=False)
```

Drop NaN values

```
# # df_list = [change_health, hospital_admit, doc_visits, restaurants_prop, indicator_combina
# df_list = [change_health, hospital_admit, doc_visits, indicator_combination]

# merged = covidcast.aggregate_signals(df_list)
```

```
# import numpy as np

# merged = merged.rename(
#     columns={
#         'chng_smoothed_adj_outpatient_covid_0_value': 'change_health',
#         'hospital-admissions_smoothed_adj_covid19_from_claims_1_value': 'hospital_admit',
# #         'fb-survey_smoothed_cli_3_value': 'survey',
#         'doctor-visits_smoothed_adj_cli_2_value': 'doc_visits',
# #         'safegraph_restaurants_visit_prop_3_value': 'restaurants_prop',
#         'indicator-combination_confirmed_incidence_num_3_value': 'ground_truth'
#     }
# )

# # keep_list = ['geo_value', 'time_value',
# #              'change_health', 'hospital_admit',
# #              'doc_visits', 'restaurants_prop', 'ground_truth']
# keep_list = ['geo_value', 'time_value',
#              'change_health', 'hospital_admit',
#              'doc_visits', 'ground_truth']
# merged = merged[keep_list]
# merged = merged.dropna().sort_values(by=['geo_value', 'time_value'])
# merged = merged.drop([4982,5038])
# merged
```

```
# today_list = ['change_health', 'hospital_admit', 'doc_visits']
# yesterday_list = ['change_health-1', 'hospital_admit-1', 'doc_visits-1']
# # before_yesterday_list = ['change_health-2', 'hospital_admit-2', 'doc_visits-2', 'restaura

# merged['ground_truth+1'] = merged['ground_truth'].shift(-1)
# for county in merged['geo_value'].unique():
#     merged.loc[merged['geo_value']==county, 'ground_truth+1'] = merged.loc[merged['geo_valu
#     for i in range(len(today_list)):
#         merged.loc[merged['geo_value']==county, yesterday_list[i]] = merged.loc[merged['geo

# drop_na = merged.drop(columns='ground_truth').dropna()
# drop_na
```

```
# # export as a csv
# import pandas as pd
# compression_opts = dict(method='zip',
#                         archive_name='drop_na.csv')
# drop_na.to_csv('drop_na.zip', index=False,
#           compression=compression_opts)
```