Covid Vaccination

December 20, 2021

1 Analysis of Coronavirus Vaccination Rates

1.1 Olivia Taylor

I chose to explore topic of coronavirus vaccination rates. Over the past two years we have collectively experienced a trauma due to the novel Coronavirus and the various efforts to lessen its impact. Multiple vaccines were developed in record time, some of which were mRNA vaccines which is a new technology that will create . Since then we have been working to mitigate the effects of the coronavirus by distributing vaccines and as of November 24, 2021 over half of the global population has received at least one dose of a vaccine, and 42.7% of the world is now fully vaccinated.

While more and more people are receiving the vaccines, these vaccines are not distributed uniformly and countries have varying vaccination rates with the United Arab Emirates as of 11/15/2021 having 98% of its population at least partially vaccinated leading. And some countries having extremely low rates such as the Democratic Republic of Congo where less than one percent of the population have received a dose as of 12/7/2021.

I want to explore the potential factors that impact vaccination rates with a special focus on political violence. While it is easy to assume that markers such as GDP and human development index have a significant impact on vaccination rates globally, there is little study on political violence and how that has impacted coronavirus vaccination rates, despite the amount of violence that is occurring throughout the world. In the past couple of years, political violence has been prominent in the news in many countries around the world. Myanmar had a military coup last February and the country has been cut off from the rest of the world since https://www.bbc.com/news/world-asia-55902070. There has been civil war going on in Ethiopia with the country on the brink of collapse https://www.nytimes.com/article/ethiopia-tigray-conflict-explained.html. And even in the US we had the events of January 6th which no one will forget at any point soon.

The goal will be to create a model that will predict vaccination rates within countries provided certain data including political violence and HDI.

1.2 Data Collection:

There are two relevant datasets that I have found. Our World in Data (OWID) has comprehensive data on coronavirus vaccinations and the data is updated on a daily basis. I will be using the OWID Covid data as well as data from the Armed Conflict Location & Event Data Project (ACLED) whose dataset covers conflict internationally with many features including location, date of event, actors, casualties, and source of information among others.

OWID Covid site: https://ourworldindata.org/coronavirus

OWID Covid data: https://github.com/owid/covid-19-data

ACLED site: https://acleddata.com/#/dashboard

ACLED data can be obtained by making a free account.

[1]:
```
import seaborn as sns
import numpy as np
import csv
import pandas as pd
import datetime
import statsmodels.api as sm
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt

from sklearn.svm import SVC
from scipy.stats import norm
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.linear_model import LogisticRegression
from numpy import mean
from sklearn import tree
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OrdinalEncoder
from sklearn.preprocessing import OneHotEncoder
from sklearn.neighbors import KNeighborsRegressor
```

[2]:
```
latest = pd.read_csv('owid-covid-latest.txt')
total = pd.read_csv('owid-covid-data.csv')
acled = pd.read_csv('acled.csv')
```

1.3 Data Processing

In this next section we will process the data so that it can be easily explored and manipulated in the future. The result will be two datasets. The first one will contain the total vaccination rates along with the total fatalities due to political violence for each country, where the fatalities are normalized by population. The second dataset will be a time series that has vaccination rates taken at one week intervals in each country with a column added for the number of deaths due to violence that has occurred in that country since the last interval.

The OWID vaccination data has 138,727 observations and 67 features and the ACLED data has 731,342 observations and 31 features, and the cleaned data that we will be working with needs to shorten and combine these in a way that is workable.

[3]:

```python
# first we remove the irrelevant columns and make sure that only countries and␣
↪not regions are being represented in
# the data. Then we drop the entries missing data that is relevant. We then␣
↪work with the the date info to make
# it easy to use in the future
latest = latest[['iso_code', 'continent', 'location', 'population',␣
↪'last_updated_date', 'total_cases_per_million',␣
↪'people_vaccinated_per_hundred', 'people_fully_vaccinated_per_hundred',␣
↪'total_vaccinations_per_hundred','gdp_per_capita',␣
↪'human_development_index']]
latest.dropna(subset=['total_vaccinations_per_hundred',␣
↪'people_vaccinated_per_hundred','last_updated_date',␣
↪'location','iso_code','continent'], inplace=True)
latest['last_updated_date']=pd.to_datetime(latest['last_updated_date'])

total = total[['iso_code', 'continent', 'location',␣
↪'population_density','population', 'date', 'total_cases_per_million',␣
↪'people_vaccinated_per_hundred',␣
↪'total_vaccinations_per_hundred','gdp_per_capita',␣
↪'human_development_index']]
total.dropna(subset=['total_vaccinations_per_hundred',␣
↪'people_vaccinated_per_hundred','date', 'location','iso_code','continent'],␣
↪inplace=True)
total['date']=pd.to_datetime(total['date'])
total['month']= total.apply(lambda row: row['date'].month, axis=1)
total['year']= total.apply(lambda row: row['date'].year, axis=1)

# create a string date variable for just the month and year
def year_month(val):
    if val['month'] < 10:
        return str(val['year']) + "/0" + str(val['month'])
    return str(val['year']) + "/" + str(val['month'])
# compresses the data into one month chunks rather than all of the entries we␣
↪had before
def vax_month_per_hundred(val):
    month = val['month']
    year = val['year']
    iso = val['iso_code']
    v = total[total['year'] == year]
    v = v[v['month'] == month]
    return v[v['iso_code'] == iso]['total_vaccinations_per_hundred'].max()
total['year_month'] = total.apply(lambda row: year_month(row), axis = 1)
total['vax_month_per_hundred'] = total.apply(lambda row:␣
↪vax_month_per_hundred(row), axis=1)
total = total.drop_duplicates(subset=['year_month', 'iso_code'], keep='last')
```

[4]:
```
# we are only interested in the violent events that resulted in deaths
acled = acled[acled['fatalities'] > 0]
acled = acled[['iso3','event_date','event_type','fatalities']]
acled['event_date']=pd.to_datetime(acled['event_date'])
acled['month']=acled.apply(lambda row: row['event_date'].month, axis=1)
acled['year']=acled.apply(lambda row: row['event_date'].year, axis=1)

# gives the sum per country of deaths due to political violence
def violence(val):
    iso = val['iso_code']
    return acled[acled['iso3'] == iso]['fatalities'].sum()
latest['fatalities'] = latest.apply(lambda row: violence(row),axis=1)
latest['death_per_million'] = (latest['fatalities'] /␣
↪latest['population'])*1000000
latest['fatalities_over_100'] = latest['fatalities'] > 100

# creates a sum for deaths due to political violence for the month for each␣
↪country
def violence(val):
    iso = val['iso_code']
    month = val['month']
    year = val['year']
    c = acled[acled['month'] == month]
    c = c[c['year'] == year]
    return c[c['iso3']==iso]['fatalities'].sum()
total['fatalities'] = total.apply(lambda row: violence(row), axis=1)
total['death_per_million'] = (total['fatalities'] / total['population'])*1000000
total = total[['iso_code', 'continent', 'location', 'year_month',␣
↪'vax_month_per_hundred','fatalities','death_per_million',␣
↪'population_density','gdp_per_capita', 'human_development_index']]
```

Now we have cleaned and consolidated the data. There is now an entry for every country for every month in the larger dataset we are working with with entries for political deaths normalized by population.

[5]:
```
total.head(10)
```

[5]:
```
    iso_code continent     location year_month  vax_month_per_hundred  \
370      AFG      Asia  Afghanistan    2021/02                   0.02
386      AFG      Asia  Afghanistan    2021/03                   0.14
423      AFG      Asia  Afghanistan    2021/04                   0.60
461      AFG      Asia  Afghanistan    2021/05                   1.51
492      AFG      Asia  Afghanistan    2021/06                   2.23
503      AFG      Asia  Afghanistan    2021/07                   2.42
554      AFG      Asia  Afghanistan    2021/08                   4.97
584      AFG      Asia  Afghanistan    2021/09                   5.95
642      AFG      Asia  Afghanistan    2021/11                  13.13
```

```
1650      ALB     Europe      Albania     2021/01                  0.02

      fatalities  death_per_million  population_density  gdp_per_capita  \
370         2751          69.059130              54.422        1803.987
386         2399          60.222775              54.422        1803.987
423         3586          90.020371              54.422        1803.987
461         6378         160.108735              54.422        1803.987
492         8551         214.658168              54.422        1803.987
503         8763         219.980064              54.422        1803.987
554         5874         147.456681              54.422        1803.987
584          176           4.418178              54.422        1803.987
642          163           4.091835              54.422        1803.987
1650           0           0.000000             104.871       11803.431

      human_development_index
370                     0.511
386                     0.511
423                     0.511
461                     0.511
492                     0.511
503                     0.511
554                     0.511
584                     0.511
642                     0.511
1650                    0.795
```

[6]:
```
latest.head(10)
```

[6]:
```
   iso_code      continent             location  population last_updated_date  \
2       ALB         Europe              Albania   2872934.0        2021-12-11
3       DZA         Africa              Algeria  44616626.0        2021-12-11
4       AND         Europe              Andorra     77354.0        2021-12-11
5       AGO         Africa               Angola  33933611.0        2021-12-11
6       AIA  North America             Anguilla     15125.0        2021-12-08
7       ATG  North America  Antigua and Barbuda     98728.0        2021-12-11
8       ARG  South America            Argentina  45605823.0        2021-12-11
9       ARM           Asia              Armenia   2968128.0        2021-12-11
10      ABW  North America                Aruba    107195.0        2021-12-10
12      AUS        Oceania            Australia  25788217.0        2021-12-11

    total_cases_per_million  people_vaccinated_per_hundred  \
2                 70841.864                          38.02
3                  4766.205                          15.41
4                251312.149                          72.52
5                  1927.204                          20.64
6                       NaN                          66.64
7                 42125.841                          62.21
```

```
8               117460.549                          81.78
9               115360.591                          28.16
10                     NaN                          78.30
12                8877.116                          78.35

    people_fully_vaccinated_per_hundred  total_vaccinations_per_hundred  \
2                                 34.20                           75.40
3                                 12.08                           27.55
4                                 65.07                          137.59
5                                  9.77                           30.41
6                                 60.98                          134.11
7                                 58.08                          120.30
8                                 67.85                          156.55
9                                 17.42                           45.58
10                                73.14                          151.45
12                                74.69                          155.71

    gdp_per_capita  human_development_index  fatalities  death_per_million  \
2        11803.431                    0.795           1           0.348076
3        13913.839                    0.748          81           1.815467
4              NaN                    0.868           0           0.000000
5         5819.495                    0.581         155           4.567743
6              NaN                      NaN           0           0.000000
7        21490.943                    0.778           1          10.128839
8        18933.907                    0.845          51           1.118278
9         8787.580                    0.776          52          17.519460
10       35973.781                      NaN           0           0.000000
12       44648.710                    0.944           0           0.000000

    fatalities_over_100
2                 False
3                 False
4                 False
5                  True
6                 False
7                 False
8                 False
9                 False
10                False
12                False
```

1.4 Exploratory Analysis & Data Visualization

First we will create a quick regression model for both datasets to get a preliminary glance at what variables might be have important relationships that we may want to further explore visually.

[7]:
```
fit1 = smf.ols(formula="vax_month_per_hundred ~ death_per_million + continent +␣
↪year_month + population_density + gdp_per_capita + human_development_index +␣
↪iso_code", data=total).fit()
fit1.summary()
```

[7]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                              OLS Regression Results
================================================================================
=
Dep. Variable:     vax_month_per_hundred   R-squared:
0.855
Model:                               OLS   Adj. R-squared:
0.838
Method:                    Least Squares   F-statistic:
50.07
Date:                   Mon, 20 Dec 2021   Prob (F-statistic):
0.00
Time:                           23:44:03   Log-Likelihood:
-8009.0
No. Observations:                   1817   AIC:
1.640e+04
Df Residuals:                       1624   BIC:
1.747e+04
Df Model:                            192
Covariance Type:               nonrobust
================================================================================
==============
                                 coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------------------
--------------
Intercept                   -107.7025      4.028    -26.741      0.000
-115.602     -99.803
continent[T.Asia]             20.0775      1.506     13.329      0.000
17.123      23.032
continent[T.Europe]           21.4040      1.621     13.206      0.000
18.225      24.583
continent[T.North America]    15.3547      1.664      9.230      0.000
12.092      18.618
continent[T.Oceania]           8.2633      2.370      3.487      0.001
3.615      12.911
continent[T.South America]    31.2075      1.891     16.505      0.000
27.499      34.916
year_month[T.2021/01]          7.4211      4.730      1.569      0.117
-1.856      16.699
year_month[T.2021/02]         21.5911      4.410      4.896      0.000
```

```
12.941      30.241
year_month[T.2021/03]         40.9969      4.278      9.584      0.000
32.606      49.387
year_month[T.2021/04]         49.4924      4.262     11.614      0.000
41.134      57.851
year_month[T.2021/05]         61.3201      4.242     14.457      0.000
53.000      69.640
year_month[T.2021/06]         72.0174      4.236     17.002      0.000
63.709      80.325
year_month[T.2021/07]         83.8472      4.242     19.767      0.000
75.527      92.167
year_month[T.2021/08]         96.5628      4.233     22.811      0.000
88.260     104.866
year_month[T.2021/09]        106.5237      4.230     25.182      0.000
98.227     114.821
year_month[T.2021/10]        115.7834      4.236     27.331      0.000
107.474     124.093
year_month[T.2021/11]        125.2460      4.234     29.579      0.000
116.941     133.551
year_month[T.2021/12]        129.5811      4.288     30.218      0.000
121.170     137.992
iso_code[T.AFG]              -10.3671     10.246     -1.012      0.312
-30.465       9.731
iso_code[T.AGO]               -8.0687      6.594     -1.224      0.221
-21.003       4.865
iso_code[T.AIA]            -2.269e-10   1.07e-10     -2.123      0.034
-4.37e-10   -1.73e-11
iso_code[T.ALB]               -8.0308      6.532     -1.230      0.219
-20.842       4.781
iso_code[T.AND]            -5.908e-10    3.5e-10     -1.687      0.092
-1.28e-09    9.61e-11
iso_code[T.ARE]               45.7648      7.616      6.009      0.000
30.826      60.703
iso_code[T.ARG]                3.7572      5.817      0.646      0.518
-7.653      15.167
iso_code[T.ARM]              -39.8834      6.898     -5.782      0.000
-53.413     -26.354
iso_code[T.ATG]               16.4232      6.198      2.650      0.008
4.266      28.580
iso_code[T.AUS]               -9.3751      6.001     -1.562      0.118
-21.146       2.396
iso_code[T.AUT]                0.2911      5.758      0.051      0.960
-11.003      11.585
iso_code[T.AZE]                2.8927      6.025      0.480      0.631
-8.925      14.710
iso_code[T.BDI]              -40.2362     11.892     -3.384      0.001
-63.561     -16.911
```

iso_code[T.BEL]	10.7993	5.760	1.875	0.061	-0.499	22.098
iso_code[T.BEN]	-17.4256	7.863	-2.216	0.027	-32.848	-2.003
iso_code[T.BES]	4.439e-10	1.03e-10	4.301	0.000	2.41e-10	6.46e-10
iso_code[T.BFA]	-20.5967	9.272	-2.221	0.026	-38.783	-2.411
iso_code[T.BGD]	-5.5380	5.993	-0.924	0.356	-17.293	6.217
iso_code[T.BGR]	-14.4330	6.549	-2.204	0.028	-27.278	-1.587
iso_code[T.BHR]	33.9317	5.978	5.677	0.000	22.207	45.656
iso_code[T.BHS]	-29.8441	6.813	-4.380	0.000	-43.208	-16.480
iso_code[T.BIH]	-26.9689	6.875	-3.923	0.000	-40.454	-13.484
iso_code[T.BLR]	-25.2743	5.988	-4.221	0.000	-37.020	-13.529
iso_code[T.BLZ]	12.3210	6.210	1.984	0.047	0.140	24.502
iso_code[T.BMU]	-6.641e-10	2e-10	-3.314	0.001	-1.06e-09	-2.71e-10
iso_code[T.BOL]	-15.2849	6.062	-2.522	0.012	-27.174	-3.395
iso_code[T.BRA]	12.3867	5.829	2.125	0.034	0.954	23.820
iso_code[T.BRB]	11.9319	6.184	1.929	0.054	-0.198	24.061
iso_code[T.BRN]	-36.0644	6.729	-5.360	0.000	-49.262	-22.866
iso_code[T.BTN]	62.2083	6.582	9.452	0.000	49.298	75.118
iso_code[T.BWA]	-12.7181	6.543	-1.944	0.052	-25.551	0.115
iso_code[T.CAF]	-11.2017	8.701	-1.287	0.198	-28.267	5.864
iso_code[T.CAN]	15.9242	5.690	2.798	0.005	4.763	27.085
iso_code[T.CHE]	-16.4092	5.744	-2.857	0.004	-27.676	-5.142
iso_code[T.CHL]	49.2546	5.618	8.767	0.000	38.235	60.275
iso_code[T.CHN]	59.9414	10.325	5.805	0.000	39.690	80.193
iso_code[T.CIV]	-7.6220	6.597	-1.155	0.248		

```
-20.561       5.317
iso_code[T.CMR]              -18.6386      6.945     -2.684      0.007
-32.262      -5.016
iso_code[T.COD]              -15.7714      7.350     -2.146      0.032
-30.189      -1.354
iso_code[T.COG]              -17.0251      6.945     -2.451      0.014
-30.647      -3.403
iso_code[T.COK]             8.334e-10   1.83e-10      4.548      0.000
4.74e-10    1.19e-09
iso_code[T.COL]               -7.5110      6.341     -1.185      0.236
-19.948       4.926
iso_code[T.COM]               12.3693      7.355      1.682      0.093
-2.058      26.796
iso_code[T.CPV]               23.8078      6.919      3.441      0.001
10.236      37.380
iso_code[T.CRI]               21.2486      5.728      3.710      0.000
10.014      32.484
iso_code[T.CUB]            -1.211e-11   1.17e-10     -0.104      0.917
-2.41e-10    2.17e-10
iso_code[T.CUW]            -5.694e-10   1.75e-10     -3.259      0.001
-9.12e-10   -2.27e-10
iso_code[T.CYM]            -1.518e-10   5.26e-11     -2.886      0.004
-2.55e-10   -4.86e-11
iso_code[T.CYP]               15.1042      5.993      2.520      0.012
3.349      26.860
iso_code[T.CZE]                3.0840      5.763      0.535      0.593
-8.221      14.389
iso_code[T.DEU]               -0.0860      5.758     -0.015      0.988
-11.381      11.209
iso_code[T.DJI]              -10.2448      7.863     -1.303      0.193
-25.667       5.177
iso_code[T.DMA]               12.7425      6.199      2.056      0.040
0.585      24.900
iso_code[T.DNK]               12.5749      6.248      2.013      0.044
0.320      24.829
iso_code[T.DOM]               28.8695      6.200      4.656      0.000
16.709      41.030
iso_code[T.DZA]              -11.6935      8.417     -1.389      0.165
-28.203       4.816
iso_code[T.ECU]               11.6885      5.827      2.006      0.045
0.260      23.117
iso_code[T.EGY]               -9.0917      6.260     -1.452      0.147
-21.369       3.186
iso_code[T.ESP]               20.8540      5.992      3.480      0.001
9.101      32.607
iso_code[T.EST]                1.5945      5.763      0.277      0.782
-9.709      12.898
```

iso_code[T.ETH]	-10.6936	6.940	-1.541	0.124	-24.305	2.918
iso_code[T.FIN]	4.1757	5.988	0.697	0.486	-7.569	15.921
iso_code[T.FJI]	33.9191	6.344	5.346	0.000	21.475	46.363
iso_code[T.FLK]	3.333e-10	1.13e-10	2.940	0.003	1.11e-10	5.56e-10
iso_code[T.FRA]	10.9716	5.762	1.904	0.057	-0.331	22.274
iso_code[T.FRO]	-7.02e-12	1.03e-10	-0.068	0.946	-2.09e-10	1.95e-10
iso_code[T.GAB]	-29.1920	6.560	-4.450	0.000	-42.060	-16.324
iso_code[T.GBR]	28.3233	5.990	4.728	0.000	16.574	40.072
iso_code[T.GEO]	-29.0731	6.539	-4.446	0.000	-41.899	-16.248
iso_code[T.GGY]	1.921e-10	1.17e-10	1.641	0.101	-3.75e-11	4.22e-10
iso_code[T.GHA]	-6.9981	7.355	-0.952	0.341	-21.424	7.427
iso_code[T.GIB]	4.575e-10	1.32e-10	3.464	0.001	1.98e-10	7.17e-10
iso_code[T.GIN]	0.1236	6.590	0.019	0.985	-12.802	13.049
iso_code[T.GMB]	-4.0083	6.593	-0.608	0.543	-16.940	8.924
iso_code[T.GNB]	-12.7521	7.353	-1.734	0.083	-27.175	1.671
iso_code[T.GNQ]	-13.8721	6.573	-2.111	0.035	-26.764	-0.980
iso_code[T.GRC]	15.4428	5.761	2.681	0.007	4.144	26.742
iso_code[T.GRD]	-10.0180	6.195	-1.617	0.106	-22.169	2.133
iso_code[T.GRL]	-4.99e-10	1.78e-10	-2.802	0.005	-8.48e-10	-1.5e-10
iso_code[T.GTM]	-11.3026	6.194	-1.825	0.068	-23.451	0.846
iso_code[T.GUY]	-2.9731	6.058	-0.491	0.624	-14.855	8.909
iso_code[T.HKG]	-20.6491	4.902	-4.212	0.000	-30.265	-11.034
iso_code[T.HND]	2.8958	6.186	0.468	0.640	-9.238	15.030
iso_code[T.HRV]	0.3856	5.762	0.067	0.947		

```
-10.916      11.688
iso_code[T.HTI]            -42.1064      8.936     -4.712      0.000
-59.635     -24.578
iso_code[T.HUN]             26.1313      6.554      3.987      0.000
13.276      38.986
iso_code[T.IDN]             -5.8712      6.028     -0.974      0.330
-17.695       5.952
iso_code[T.IMN]           2.054e-10    1.7e-10      1.206      0.228
-1.29e-10    5.39e-10
iso_code[T.IND]              8.3153      6.027      1.380      0.168
-3.507      20.137
iso_code[T.IRL]            -13.6520      5.720     -2.387      0.017
-24.871      -2.433
iso_code[T.IRN]            -13.3047      6.277     -2.119      0.034
-25.617      -0.992
iso_code[T.IRQ]            -41.1926      7.336     -5.615      0.000
-55.581     -26.804
iso_code[T.ISL]             18.4708      5.756      3.209      0.001
7.181      29.760
iso_code[T.ISR]             56.0560      5.772      9.712      0.000
44.735      67.377
iso_code[T.ITA]             16.2975      5.764      2.828      0.005
4.993      27.603
iso_code[T.JAM]            -25.9613      6.492     -3.999      0.000
-38.694     -13.228
iso_code[T.JEY]           4.013e-10   1.52e-10      2.643      0.008
1.03e-10    6.99e-10
iso_code[T.JOR]              3.3589      6.021      0.558      0.577
-8.450      15.168
iso_code[T.JPN]             -6.0183      6.245     -0.964      0.335
-18.268       6.231
iso_code[T.KAZ]            -14.7597      6.013     -2.455      0.014
-26.553      -2.967
iso_code[T.KEN]            -10.9764      6.588     -1.666      0.096
-23.899       1.946
iso_code[T.KGZ]            -28.6915      6.564     -4.371      0.000
-41.566     -15.817
iso_code[T.KHM]             52.6178      6.276      8.384      0.000
40.308      64.927
iso_code[T.KIR]             -8.1935      8.686     -0.943      0.346
-25.230       8.843
iso_code[T.KNA]              8.7682      6.196      1.415      0.157
-3.385      20.921
iso_code[T.KOR]             -4.0939      6.246     -0.655      0.512
-16.345       8.157
iso_code[T.KWT]            -32.6982      8.928     -3.662      0.000
-50.210     -15.187
```

iso_code[T.LAO]	0.1603	7.329	0.022	0.983	-14.214	14.535
iso_code[T.LBN]	-23.1157	6.576	-3.515	0.000	-36.014	-10.218
iso_code[T.LBR]	-7.7797	10.360	-0.751	0.453	-28.099	12.540
iso_code[T.LBY]	-25.9529	6.899	-3.762	0.000	-39.485	-12.420
iso_code[T.LCA]	-13.9509	6.488	-2.150	0.032	-26.678	-1.224
iso_code[T.LIE]	-3.062e-10	7.96e-11	-3.848	0.000	-4.62e-10	-1.5e-10
iso_code[T.LKA]	17.3952	6.010	2.894	0.004	5.606	29.184
iso_code[T.LSO]	-2.9348	6.947	-0.422	0.673	-16.562	10.692
iso_code[T.LTU]	12.6750	5.764	2.199	0.028	1.370	23.980
iso_code[T.LUX]	-45.7779	5.581	-8.202	0.000	-56.725	-34.830
iso_code[T.LVA]	-0.4979	5.763	-0.086	0.931	-11.801	10.805
iso_code[T.MAC]	3.929e-10	9.36e-11	4.198	0.000	2.09e-10	5.76e-10
iso_code[T.MAR]	53.0337	6.264	8.467	0.000	40.748	65.320
iso_code[T.MCO]	2.361e-11	8.7e-11	0.271	0.786	-1.47e-10	1.94e-10
iso_code[T.MDA]	-20.7806	7.253	-2.865	0.004	-35.007	-6.554
iso_code[T.MDG]	-23.3631	8.484	-2.754	0.006	-40.003	-6.723
iso_code[T.MDV]	52.9694	6.248	8.478	0.000	40.714	65.225
iso_code[T.MEX]	7.3837	5.739	1.287	0.198	-3.872	18.640
iso_code[T.MKD]	-8.4893	6.534	-1.299	0.194	-21.306	4.327
iso_code[T.MLI]	-4.7374	6.572	-0.721	0.471	-17.629	8.154
iso_code[T.MLT]	42.0555	5.952	7.065	0.000	30.380	53.730
iso_code[T.MMR]	-5.6925	6.583	-0.865	0.387	-18.604	7.219
iso_code[T.MNE]	-8.9417	6.243	-1.432	0.152	-21.188	3.304
iso_code[T.MNG]	47.7031	6.279	7.597	0.000		

```
35.387      60.019
iso_code[T.MOZ]              0.6122      6.928      0.088      0.930
-12.976      14.201
iso_code[T.MRT]             -1.6316      6.597     -0.247      0.805
-14.570      11.307
iso_code[T.MSR]           1.679e-10   3.02e-11      5.566      0.000
1.09e-10    2.27e-10
iso_code[T.MUS]             43.6356      6.508      6.705      0.000
30.871      56.400
iso_code[T.MWI]             -4.2838      6.591     -0.650      0.516
-17.211       8.644
iso_code[T.MYS]             11.5190      6.278      1.835      0.067
-0.796      23.834
iso_code[T.NAM]            -13.4310      6.580     -2.041      0.041
-26.338      -0.525
iso_code[T.NCL]           1.411e-10   7.58e-11      1.862      0.063
-7.52e-12     2.9e-10
iso_code[T.NER]              3.0109      7.304      0.412      0.680
-11.316      17.338
iso_code[T.NGA]            -12.7795      6.595     -1.938      0.053
-25.714       0.155
iso_code[T.NIC]            -12.3332      6.820     -1.808      0.071
-25.711       1.044
iso_code[T.NIU]          -9.805e-11   7.32e-11     -1.339      0.181
-2.42e-10    4.56e-11
iso_code[T.NLD]            -14.0243      7.791     -1.800      0.072
-29.305       1.257
iso_code[T.NOR]            -17.2168      5.726     -3.007      0.003
-28.448      -5.986
iso_code[T.NPL]             -1.9769      6.275     -0.315      0.753
-14.286      10.332
iso_code[T.NRU]          -2.076e-10   8.77e-11     -2.367      0.018
-3.8e-10   -3.56e-11
iso_code[T.NZL]             -5.4697      6.019     -0.909      0.364
-17.276       6.337
iso_code[T.OMN]            -19.5952      6.028     -3.251      0.001
-31.418      -7.772
iso_code[T.OWID_CYN]     -1.209e-10   7.69e-11     -1.572      0.116
-2.72e-10       3e-11
iso_code[T.OWID_KOS]      2.004e-10   5.86e-11      3.418      0.001
8.54e-11    3.15e-10
iso_code[T.PAK]             -7.9278      6.264     -1.266      0.206
-20.215       4.359
iso_code[T.PAN]             12.7376      5.942      2.144      0.032
1.083      24.392
iso_code[T.PCN]          -8.172e-11   5.48e-11     -1.490      0.136
-1.89e-10    2.58e-11
```

iso_code[T.PER]	-6.6406	6.062	-1.095	0.273	-18.531	5.250
iso_code[T.PHL]	-15.5030	6.571	-2.359	0.018	-28.392	-2.614
iso_code[T.PNG]	-17.2374	6.623	-2.603	0.009	-30.227	-4.248
iso_code[T.POL]	4.0433	5.763	0.702	0.483	-7.261	15.347
iso_code[T.PRT]	29.8837	5.764	5.185	0.000	18.578	41.189
iso_code[T.PRY]	-16.3279	6.062	-2.693	0.007	-28.219	-4.437
iso_code[T.PSE]	-12.7520	6.891	-1.850	0.064	-26.268	0.765
iso_code[T.PYF]	1.593e-10	3.87e-11	4.120	0.000	8.35e-11	2.35e-10
iso_code[T.QAT]	-30.5658	6.972	-4.384	0.000	-44.240	-16.891
iso_code[T.ROU]	-5.6926	6.554	-0.869	0.385	-18.549	7.163
iso_code[T.RUS]	-16.7745	5.993	-2.799	0.005	-28.529	-5.020
iso_code[T.RWA]	16.0169	6.293	2.545	0.011	3.673	28.360
iso_code[T.SAU]	-2.0935	7.751	-0.270	0.787	-17.297	13.110
iso_code[T.SDN]	-6.5734	6.943	-0.947	0.344	-20.191	7.044
iso_code[T.SEN]	1.0679	6.298	0.170	0.865	-11.285	13.420
iso_code[T.SGP]	-7.8926	4.421	-1.785	0.074	-16.563	0.778
iso_code[T.SHN]	-8.933e-12	3.72e-11	-0.240	0.810	-8.18e-11	6.4e-11
iso_code[T.SLB]	-10.1110	6.986	-1.447	0.148	-23.813	3.591
iso_code[T.SLE]	-3.4771	6.932	-0.502	0.616	-17.073	10.119
iso_code[T.SLV]	37.1611	6.193	6.000	0.000	25.014	49.308
iso_code[T.SMR]	-3.474e-11	4.18e-11	-0.831	0.406	-1.17e-10	4.73e-11
iso_code[T.SOM]	1.034e-10	6.15e-11	1.683	0.093	-1.71e-11	2.24e-10
iso_code[T.SRB]	23.3955	5.982	3.911	0.000	11.662	35.129
iso_code[T.SSD]	1.197e-10	7.29e-11	1.642	0.101		

```
-2.33e-11    2.63e-10
iso_code[T.STP]             6.8167      6.931      0.984      0.325
-6.778      20.411
iso_code[T.SUR]           -17.5265      6.063     -2.891      0.004
-29.419      -5.634
iso_code[T.SVK]            -7.4189      5.994     -1.238      0.216
-19.176       4.338
iso_code[T.SVN]             1.5878      5.761      0.276      0.783
-9.713      12.888
iso_code[T.SWE]            -0.7868      5.984     -0.131      0.895
-12.524      10.951
iso_code[T.SWZ]            -8.5245      6.590     -1.294      0.196
-21.450       4.401
iso_code[T.SXM]          3.729e-11    2.1e-11      1.773      0.076
-3.97e-12    7.85e-11
iso_code[T.SYC]            95.9975      6.223     15.426      0.000
83.791     108.204
iso_code[T.SYR]         -1.014e-11   8.81e-12     -1.151      0.250
-2.74e-11    7.14e-12
iso_code[T.TCA]          2.882e-11   2.53e-11      1.139      0.255
-2.08e-11    7.84e-11
iso_code[T.TCD]           -19.5750      7.803     -2.509      0.012
-34.880      -4.270
iso_code[T.TGO]            -1.2405      6.596     -0.188      0.851
-14.178      11.697
iso_code[T.THA]            -4.1532      6.277     -0.662      0.508
-16.465       8.158
iso_code[T.TJK]           -20.5953      7.322     -2.813      0.005
-34.958      -6.233
iso_code[T.TKL]         -3.364e-16   3.77e-16     -0.892      0.372
-1.08e-15    4.03e-16
iso_code[T.TKM]            35.8731     14.589      2.459      0.014
7.258       64.488
iso_code[T.TLS]            -3.3960      6.920     -0.491      0.624
-16.968      10.176
iso_code[T.TON]            19.2093      7.003      2.743      0.006
5.473       32.946
iso_code[T.TTO]           -14.6119      6.521     -2.241      0.025
-27.403      -1.821
iso_code[T.TUN]            10.3560      6.534      1.585      0.113
-2.460      23.171
iso_code[T.TUR]            18.1728      6.273      2.897      0.004
5.868       30.477
iso_code[T.TUV]                  0          0        nan        nan
0           0
iso_code[T.TWN]                  0          0        nan        nan
0           0
```

```
iso_code[T.TZA]              -33.9642     11.973     -2.837      0.005
-57.448     -10.480
iso_code[T.UGA]               -5.5366      7.863     -0.704      0.481
-20.958       9.885
iso_code[T.UKR]              -25.4818      6.227     -4.092      0.000
-37.696     -13.267
iso_code[T.URY]               50.7239      6.059      8.372      0.000
38.840      62.608
iso_code[T.USA]                7.0065      5.675      1.235      0.217
-4.125      18.138
iso_code[T.UZB]               -4.9153      6.569     -0.748      0.454
-17.800       7.969
iso_code[T.VCT]              -19.9305      6.826     -2.920      0.004
-33.319      -6.542
iso_code[T.VEN]              -30.3395      6.332     -4.791      0.000
-42.760     -17.919
iso_code[T.VGB]                     0          0        nan        nan
0           0
iso_code[T.VNM]               -3.7024      6.574     -0.563      0.573
-16.596       9.191
iso_code[T.VUT]              -19.1314      7.436     -2.573      0.010
-33.716      -4.546
iso_code[T.WLF]                     0          0        nan        nan
0           0
iso_code[T.WSM]               24.6529      7.444      3.312      0.001
10.053      39.253
iso_code[T.YEM]              -36.7199      8.631     -4.255      0.000
-53.648     -19.792
iso_code[T.ZAF]               -7.6244      6.258     -1.218      0.223
-19.899       4.651
iso_code[T.ZMB]              -11.1901      7.862     -1.423      0.155
-26.611       4.230
iso_code[T.ZWE]               12.5687      6.295      1.996      0.046
0.221       24.917
death_per_million             -0.0175      0.077     -0.228      0.820
-0.168       0.133
population_density            -0.0012      0.001     -1.794      0.073
-0.002       0.000
gdp_per_capita                 0.0009   3.76e-05     24.261      0.000
0.001       0.001
human_development_index       56.1112      3.382     16.590      0.000
49.477      62.745
==============================================================================
Omnibus:                       17.921   Durbin-Watson:                   0.578
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               15.110
Skew:                           0.156   Prob(JB):                     0.000523
Kurtosis:                       2.681   Cond. No.                     7.31e+20
```

```
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The smallest eigenvalue is 2.56e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
"""
```

[8]:
```
fit2 = smf.ols(formula="total_vaccinations_per_hundred ~ death_per_million +␣
 ↪continent + gdp_per_capita + human_development_index", data=latest).fit()
fit2.summary()
```

[8]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                                  OLS Regression Results
================================================================================
==========
Dep. Variable:     total_vaccinations_per_hundred   R-squared:
0.748
Model:                                        OLS   Adj. R-squared:
0.735
Method:                             Least Squares   F-statistic:
57.92
Date:                            Mon, 20 Dec 2021   Prob (F-statistic):
7.07e-43
Time:                                    23:44:03   Log-Likelihood:
-794.13
No. Observations:                             165   AIC:
1606.
Df Residuals:                                 156   BIC:
1634.
Df Model:                                       8
Covariance Type:                        nonrobust
================================================================================
==============
                                 coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------------------
--------------
Intercept                    -92.0948     21.241     -4.336      0.000
-134.051     -50.139
continent[T.Asia]             30.4825      8.588      3.550      0.001
13.520      47.445
continent[T.Europe]           16.4353     11.272      1.458      0.147
-5.830      38.701
continent[T.North America]    16.8918     10.235      1.650      0.101
```

```
-3.325      37.109
continent[T.Oceania]           23.3353     12.084      1.931      0.055
-0.534      47.204
continent[T.South America]     47.3670     11.800      4.014      0.000
24.059      70.674
death_per_million             -0.0165      0.012     -1.345      0.181
-0.041       0.008
gdp_per_capita                 0.0007      0.000      3.019      0.003
0.000        0.001
human_development_index      213.8837     38.195      5.600      0.000
138.437     289.330
==============================================================================
Omnibus:                        7.153   Durbin-Watson:                   2.093
Prob(Omnibus):                  0.028   Jarque-Bera (JB):                6.914
Skew:                           0.429   Prob(JB):                       0.0315
Kurtosis:                       3.519   Cond. No.                     4.98e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 4.98e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

From these results it appears that the biggest factor that affects vaccination rates is the time in the larger dataset and in both GDP and HDI are statistically significant features.

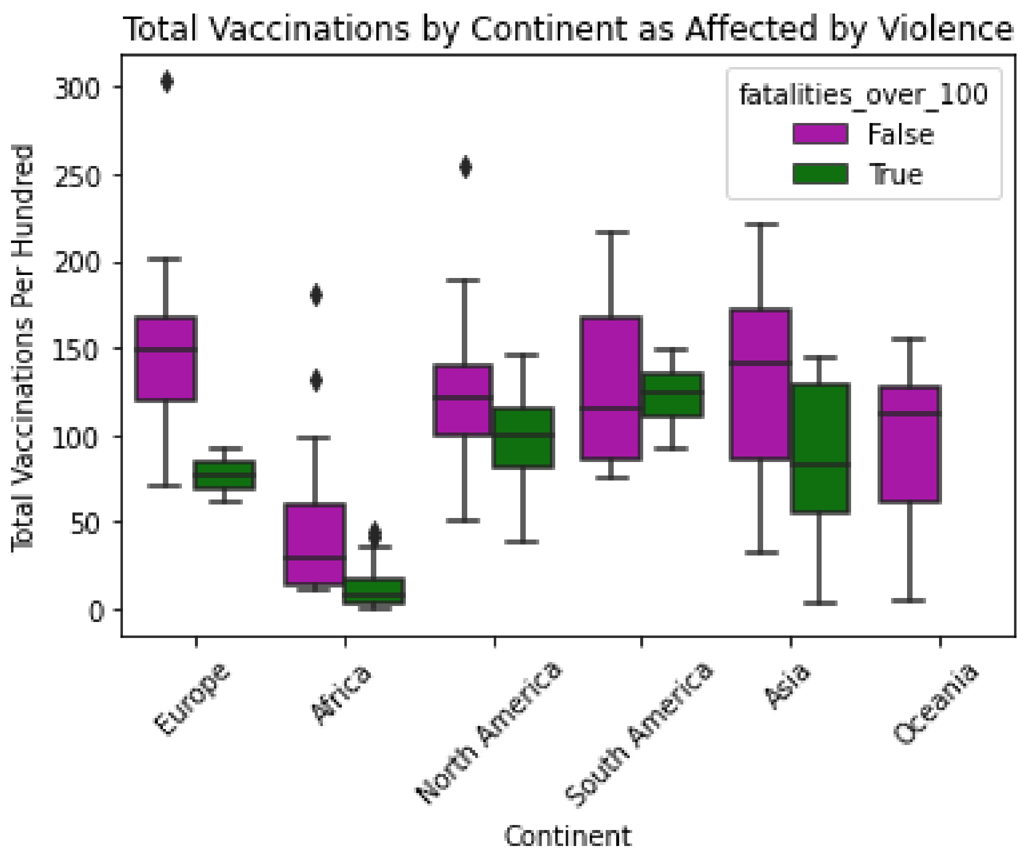
Before exploring these features, let's first look at the violence and see if we can find a relationship between that and vaccination rates. Since it seems more important in the dataset representing the recent snapshot, we will start with that one.

[9]:

```python
plot = sns.boxplot(x="continent", y="total_vaccinations_per_hundred",
            hue="fatalities_over_100", palette=["m", "g"],
            data=latest)
plot.set_xlabel('Continent')
plot.set_ylabel('Total Vaccinations Per Hundred')
plot.set_title('Total Vaccinations by Continent as Affected by Violence')
plt.xticks(rotation=45)
plot
```

[9]:

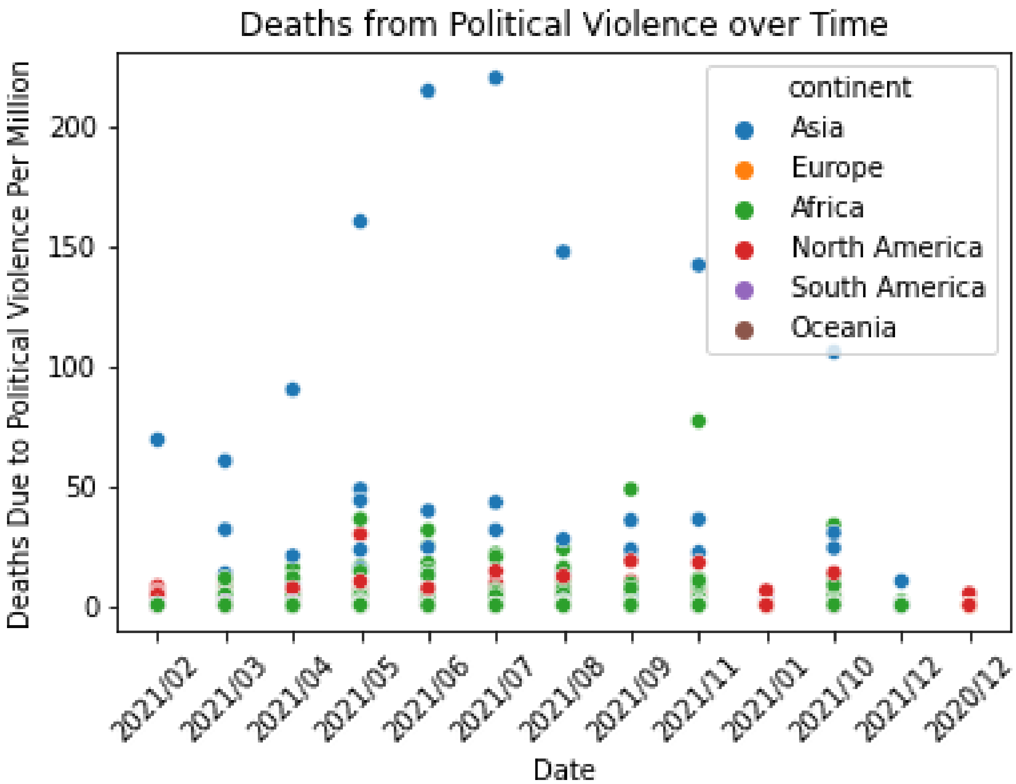
```
<AxesSubplot:title={'center':'Total Vaccinations by Continent as Affected by
Violence'}, xlabel='Continent', ylabel='Total Vaccinations Per Hundred'>
```



This plot shows that when separated by continent, mean vaccination rates are higher in countries that have fewer fatalities due to political violence. Let's look at violence over time now.

[10]:
```
plot = sns.scatterplot(x='year_month', y='death_per_million', data=total,␣
 ↪hue='continent')
plot.set_xlabel('Date')
plot.set_ylabel('Deaths Due to Political Violence Per Million')
plot.set_title('Deaths from Political Violence over Time')
plt.xticks(rotation=45)
plot
```

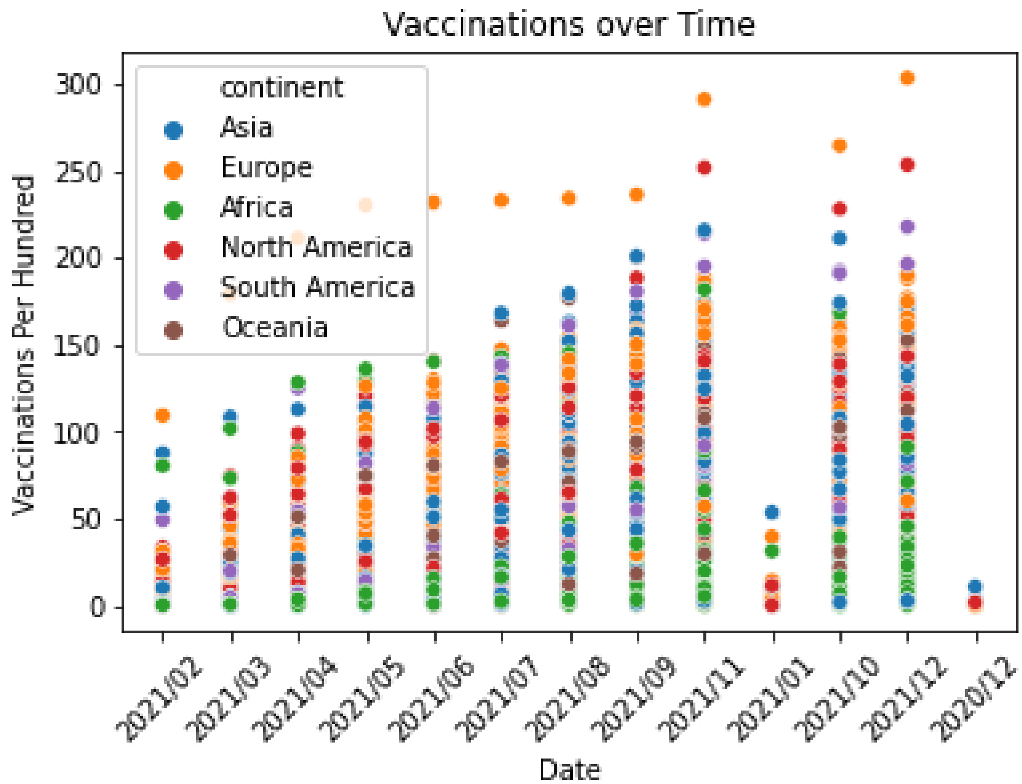
[10]:
```
<AxesSubplot:title={'center':'Deaths from Political Violence over Time'},
xlabel='Date', ylabel='Deaths Due to Political Violence Per Million'>
```



There does not seem to be a trend for political violence over time. There are several consistent outliers, but that is it. Let's look at vaccinations over time. I would expect it to appear linear and increasing, but maybe it will have some similarities with the political violence.

[11]:
```
plot = sns.scatterplot(x='year_month', y='vax_month_per_hundred',␣
↪hue='continent', data=total)
plot.set_xlabel('Date')
plot.set_ylabel('Vaccinations Per Hundred')
plot.set_title('Vaccinations over Time')
plt.xticks(rotation=45)
plot
```

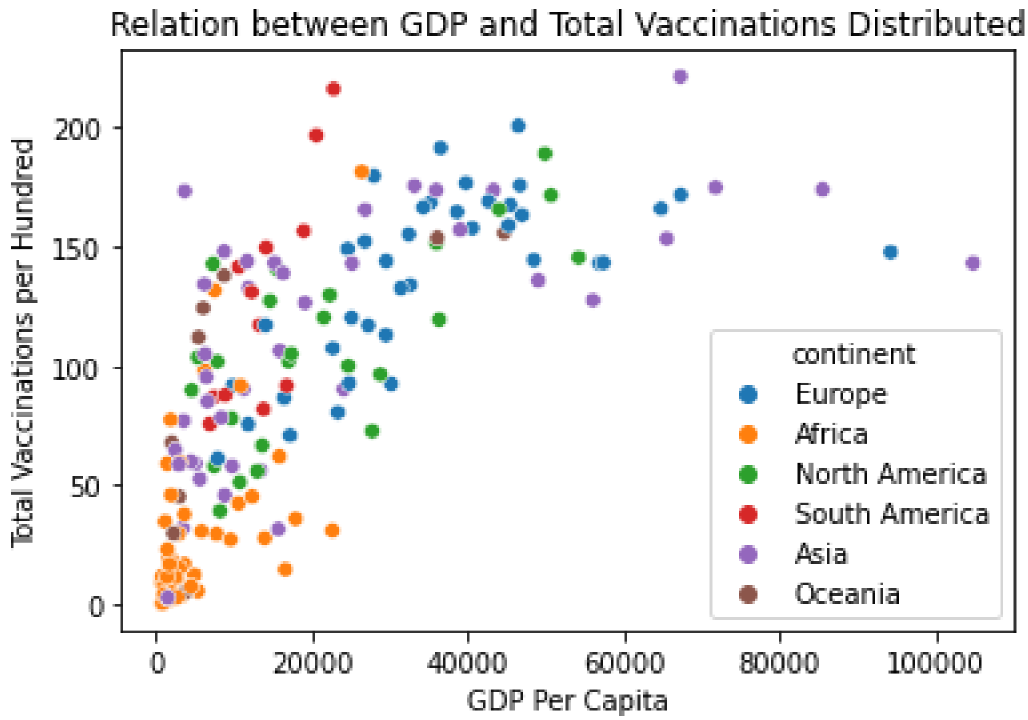
[11]:
```
<AxesSubplot:title={'center':'Vaccinations over Time'}, xlabel='Date',
ylabel='Vaccinations Per Hundred'>
```



The graph looks as expected. The last entry appears to be an outlier because there are fewer data points from December going towards the total since that month is still in progress. Let's now look at GDP and HDI.

[12]:
```
plot = sns.
 ↪scatterplot(x='gdp_per_capita',y='total_vaccinations_per_hundred',data=latest,hue='continen
plot.set_xlabel('GDP Per Capita')
plot.set_ylabel('Total Vaccinations per Hundred')
plot.set_title('Relation between GDP and Total Vaccinations Distributed')
```

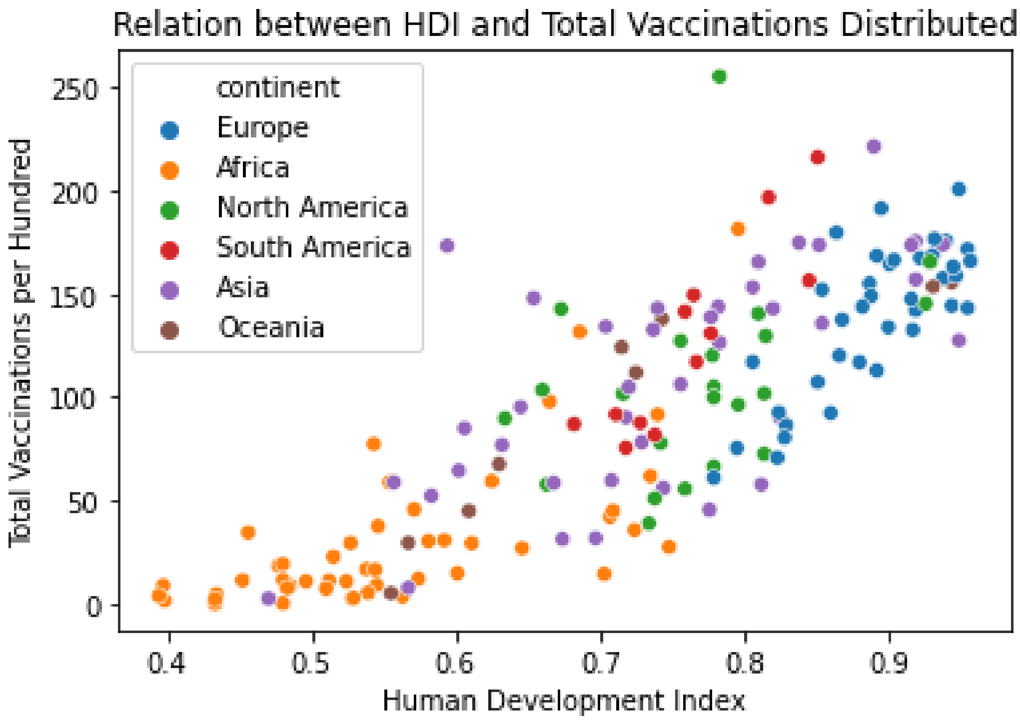
[12]: Text(0.5, 1.0, 'Relation between GDP and Total Vaccinations Distributed')



It looks like GDP has a correlation, though it does not appear that the relation between GDP and vaccinations is linear, it appears logarithmic. Now lets look at HDI.

[13]:
```
plot = sns.scatterplot(x='human_development_index',␣
 ↪y='total_vaccinations_per_hundred', data=latest,hue='continent',legend=True)
plot.set_xlabel('Human Development Index')
plot.set_ylabel('Total Vaccinations per Hundred')
plot.set_title('Relation between HDI and Total Vaccinations Distributed')
```

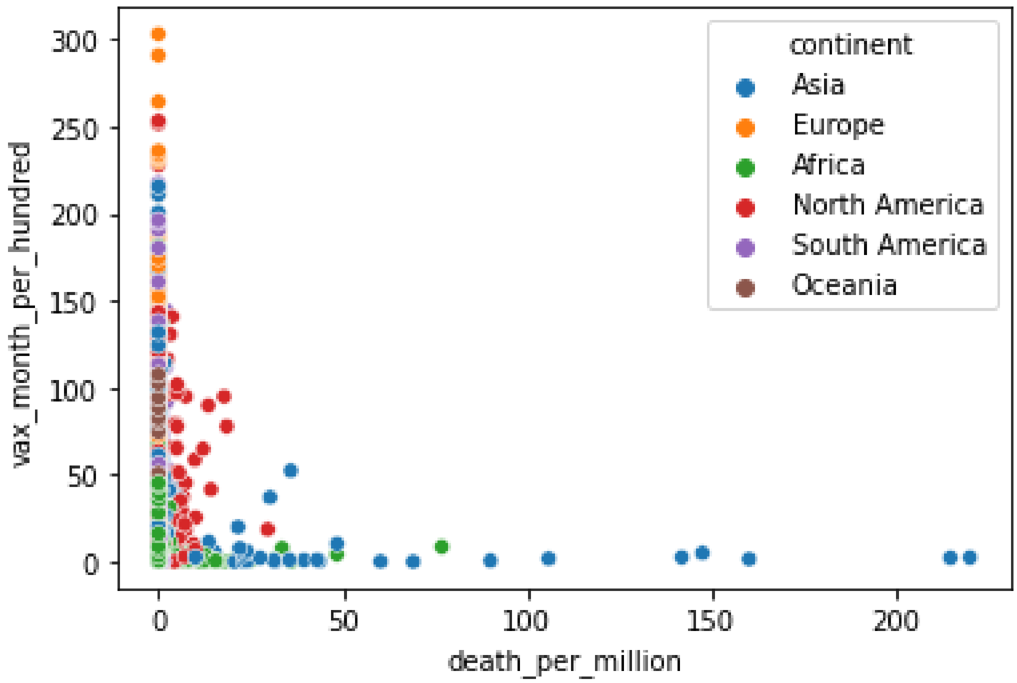
[13]:
```
Text(0.5, 1.0, 'Relation between HDI and Total Vaccinations Distributed')
```



There is clearly a strong correlation and a linear relationship between HDI and vaccinations. Let's now see the relation between deaths from political violence and vaccinations.

[14]:
```
sns.scatterplot(x='death_per_million',␣
 ↪y='vax_month_per_hundred',data=total,hue='continent')
plot.set_xlabel('Deaths From Violence Per Million')
plot.set_ylabel('Total Vaccinations per Hundred')
plot.set_title('Relation between Political Violence and Total Vaccinations␣
 ↪Distributed')
```

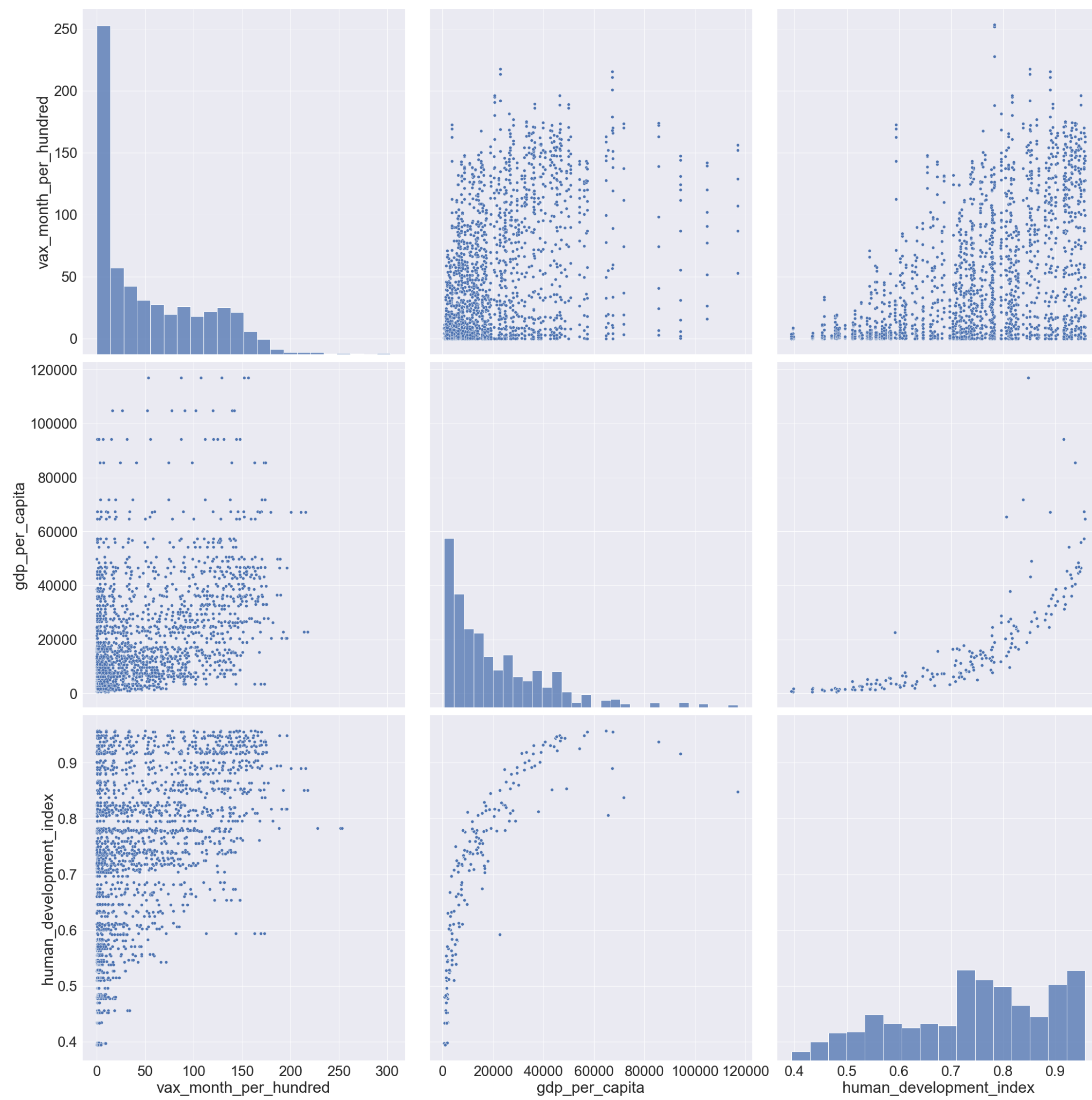
[14]:
```
Text(0.5, 1.0, 'Relation between Political Violence and Total Vaccinations
Distributed')
```



While it looks like for most observations there is not a strong relationship, for the countries with large amounts of deaths from political violence, the total vaccinations are very low across the board.

Since GDP and HDI are most likely the most important features, let's create a plot to show any correlation that might occur between total vaccinations and these two features.

[15]:
```
sns.set(font_scale=2.5)
plot = sns.pairplot(total[['vax_month_per_hundred','gdp_per_capita',␣
↪'human_development_index']], height=10)
```



There seems to be correlations among all three of these features. We will now see how these features can be used in creating predictive models for vaccination rates.

1.5 Model Analysis, Hypothesis Testing, and ML

We will now take the features of date, continent, country, human development index, and GDP and use them to create a model that fits the data and will have the ability to predict vaccination rates. In order to be able to use the categorical variables of date, continent and country, we first need to use encodings. For country and continent I chose to use one hot encoding since using an ordinal encoding would negatively impact the algorithms.

[16]:
```
# creating the kfold object to do the cross validation computations
cv = KFold(n_splits=20, shuffle=True)
# use ordinal encoder to encode the date feature
```

```
ord_enc = OrdinalEncoder()
tot=total
tot.reset_index(inplace = True)
tot["year_month code"] = ord_enc.fit_transform(tot[["year_month"]])
# use one hot encoder to encode the continent and country features
# although this adds over 200 features, it allows for these categorical␣
↪variables
# to be distinguished from one another without implying order
oe_style = OneHotEncoder()
oe_results = oe_style.fit_transform(tot[["continent"]])
to_add = pd.DataFrame(oe_results.toarray(), columns=oe_style.categories_[0])
oe_results2 = oe_style.fit_transform(tot[["iso_code"]])
to_add = pd.DataFrame(oe_results2.toarray(), columns=oe_style.categories_[0])
tot = pd.concat([tot, to_add], axis=1)
```

[17]:
```
# create the sets for the features and the values meant to be predicted
# all features non-numeric features are removed
y = tot.dropna()['vax_month_per_hundred']
X = tot.dropna().drop(['index','continent', 'year_month','fatalities',␣
↪'continent','iso_code','location', 'vax_month_per_hundred'],axis=1)
```

We can now begin to create models for the data!

For each model in this section the effectiveness will be assessed through the mean of a 20-fold cross validation with the score coming from the R2 value. While the mean will differ every time the code is run, a 20-fold validation should lower the variance of the mean.

The first model that I would like to use is the linear regression model.

[18]:
```
clf = LinearRegression()
cross_val_score(clf, X, y, scoring='r2', cv=cv, n_jobs=-1).mean()
```

[18]: 0.805784550771128

The mean is approximately .809 during this run of the code. These results are decent for the model. Let's now look at the coefficients for the non-categorical variables.

[19]:
```
clf = LinearRegression().fit(X, y)
dir(clf)
d = pd.DataFrame(clf.coef_, columns=['coefficients'])
d['feature name'] = pd.DataFrame(clf.feature_names_in_)
d.head(5)
```

[19]:

	coefficients	feature name
0	0.007239	death_per_million
1	-0.001051	population_density
2	0.000801	gdp_per_capita
3	113.728603	human_development_index
4	10.983976	year_month code

From these results we can see that the human development index was the most important variable. Now let's create a Bayesian ridge model!

[20]:
```
clf = linear_model.BayesianRidge()
cross_val_score(clf, X, y, scoring='r2', cv=cv, n_jobs=-1).mean()
```

[20]: 0.8122832828306722

This model performs slightly better than the linear regression model with a mean score of .811 this run of the code. Let's see if the coefficients of the Bayesian ridge model show similar results to the linear regression model.

[21]:
```
clf = linear_model.BayesianRidge().fit(X, y)
dir(clf)
d = pd.DataFrame(clf.coef_, columns=['coefficients'])
d['feature name'] = pd.DataFrame(clf.feature_names_in_)
d.head(5)
```

[21]:

	coefficients	feature name
0	-0.010657	death_per_million
1	-0.001048	population_density
2	0.000823	gdp_per_capita
3	108.788314	human_development_index
4	10.938041	year_month code

As expected HDI is the most heavily weighted feature. Now let's make a model using K-nearest neighbors regression.

[22]:
```
clf = KNeighborsRegressor(n_neighbors=2)
cross_val_score(clf, X, y, cv=cv,scoring='r2', n_jobs=-1).mean()
```

[22]: 0.9635986867209005

This model performs the best of all with a mean R2 of .963 this run of the code. This is a solid model that can hopefully be used to predict future vaccination rates. Let's now make a decision tree regression model.

[23]:
```
clf = tree.DecisionTreeRegressor()
cross_val_score(clf, X, y, cv=cv,scoring='r2', n_jobs=-1).mean()
```

[23]: 0.8960866174509237

This model performs better than both the linear regression and Bayesian ridge models and worse than the K-nearest neighbors model with a mean score of .899. This model performs decently, but the k-neighbors model is the one that will be used. Let's see which features are most important.

[24]:
```
clf = tree.DecisionTreeRegressor().fit(X, y)
d = pd.DataFrame(clf.feature_importances_, columns=['feature importance'])
d['feature name'] = pd.DataFrame(clf.feature_names_in_)
```

```
d.sort_values('feature importance', ascending=False).head(15)
```

[24]:

	feature importance	feature name
3	0.413802	human_development_index
4	0.374051	year_month code
2	0.077682	gdp_per_capita
1	0.028678	population_density
191	0.012693	SYC
35	0.009992	BTN
210	0.007528	URY
108	0.006447	KHM
119	0.004851	LKA
40	0.004814	CHL
0	0.004573	death_per_million
99	0.004242	ISR
11	0.003980	ARE
61	0.003451	DZA
153	0.002918	NZL

As with the other models the HDI is the most important feature.

1.6 Interpretation

In this exploration of data on Covid vaccines and political violence I was hoping to be able to make some connection between the two. While I was able to create a solid predictive model, the importance of political violence as a feature was small. A couple reasons for this result are that my awareness of current events increased during the pandemic due to the isolation and I had my experience with political violence. I am a DC resident and was in the city during January 6th and the BLM protests. I was pepperspraved by the police a couple of times and barely avoided the photo op tear gas incident. I think that my experience made me overestimate the data. Another reason is that political unrest tends to be an outcome of pandemics and it is still too early for the peak of the unrest https://www.economist.com/the-world-ahead/2021/11/08/the-aftermath-of-the-pandemic-will-make-politics-more-turbulent.

The main outcome of this data is that the Human Development Index is the most important predictive feature of vaccinations. There are three indicators that go into the computations of the HDI: education, life expectancy, and per capita income https://en.wikipedia.org/wiki/Human_Development_Index#Dimensions_and_calculation. My recommendation is to work to increase and improve education in the countries that are lagging. This will not help for vaccinations during this pandemic, but it can help us better prepare to tackle the next one so that we can return to normalcy sooner.