In [29]:
```
%matplotlib inline

from numpy import arange
from matplotlib import pyplot as plt
from scipy.stats import norm
import pandas as pd

plt.rcParams['figure.figsize'] = [20,10]
```

In [28]:
```
columns = ['COUNTRY','ISO3','WHO_REGION','DATA_SOURCE','DATE_UPDATED','TOTA

df = pd.read_csv('Vaccination_Data.csv')
df.columns = columns
df.head()
```

Out[28]:

	COUNTRY	ISO3	WHO_REGION	DATA_SOURCE	DATE_UPDATED	TOTAL_VACCINATIONS
0	Afghanistan	AFG	EMRO	REPORTING	2021-08-20	1201286.0
1	Albania	ALB	EURO	OWID	2021-08-17	1340339.0
2	Algeria	DZA	AFRO	REPORTING	2021-08-23	4146091.0
3	American Samoa	ASM	WPRO	REPORTING	2021-08-09	52769.0
4	Andorra	AND	EURO	OWID	2021-08-15	91660.0

In [36]:
```
df.tail()
```

Out[36]:

	COUNTRY	ISO3	WHO_REGION	DATA_SOURCE	DATE_UPDATED	TOTAL_VACCINATION
222	Viet Nam	VNM	WPRO	REPORTING	2021-08-13	8061116
223	Wallis and Futuna	WLF	WPRO	REPORTING	2021-07-22	9276
224	Yemen	YEM	EMRO	REPORTING	2021-07-27	311483
225	Zambia	ZMB	AFRO	REPORTING	2021-08-23	503707
226	Zimbabwe	ZWE	AFRO	REPORTING	2021-08-23	3772579

In [30]:
```
df.dtypes
```

Out[30]:
```
COUNTRY                                object
ISO3                                   object
WHO_REGION                             object
DATA_SOURCE                            object
DATE_UPDATED                           object
TOTAL_VACCINATIONS                    float64
PERSONS_VACCINATED_1PLUS_DOSE         float64
TOTAL_VACCINATIONS_PER100             float64
PERSONS_VACCINATED_1PLUS_DOSE_PER100  float64
PERSONS_FULLY_VACCINATED              float64
PERSONS_FULLY_VACCINATED_PER100       float64
VACCINES_USED                          object
FIRST_VACCINE_DATE                     object
NUMBER_VACCINES_TYPES_USED            float64
dtype: object
```

In [31]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 227 entries, 0 to 226
Data columns (total 14 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   COUNTRY                               227 non-null    object 
 1   ISO3                                  227 non-null    object 
 2   WHO_REGION                            227 non-null    object 
 3   DATA_SOURCE                           227 non-null    object 
 4   DATE_UPDATED                          227 non-null    object 
 5   TOTAL_VACCINATIONS                    226 non-null    float64
 6   PERSONS_VACCINATED_1PLUS_DOSE         218 non-null    float64
 7   TOTAL_VACCINATIONS_PER100             226 non-null    float64
 8   PERSONS_VACCINATED_1PLUS_DOSE_PER100  218 non-null    float64
 9   PERSONS_FULLY_VACCINATED              216 non-null    float64
 10  PERSONS_FULLY_VACCINATED_PER100       216 non-null    float64
 11  VACCINES_USED                         223 non-null    object 
 12  FIRST_VACCINE_DATE                    155 non-null    object 
 13  NUMBER_VACCINES_TYPES_USED            223 non-null    float64
dtypes: float64(7), object(7)
memory usage: 25.0+ KB
```

In [32]:
```
df.memory_usage() # in bytes
```

Out[32]:
```
Index                                  128
COUNTRY                               1816
ISO3                                  1816
WHO_REGION                            1816
DATA_SOURCE                           1816
DATE_UPDATED                          1816
TOTAL_VACCINATIONS                    1816
PERSONS_VACCINATED_1PLUS_DOSE         1816
TOTAL_VACCINATIONS_PER100             1816
PERSONS_VACCINATED_1PLUS_DOSE_PER100  1816
PERSONS_FULLY_VACCINATED              1816
PERSONS_FULLY_VACCINATED_PER100       1816
VACCINES_USED                         1816
FIRST_VACCINE_DATE                    1816
NUMBER_VACCINES_TYPES_USED            1816
dtype: int64
```

In [33]:
```
df.memory_usage().sum()
```

Out[33]:
```
25552
```

In [34]:
```
df.describe()
```

Out[34]:

	TOTAL_VACCINATIONS	PERSONS_VACCINATED_1PLUS_DOSE	TOTAL_VACCINATIONS_
count	2.260000e+02	2.180000e+02	226
mean	2.044237e+07	8.253955e+06	63
std	1.268902e+08	3.473510e+07	51
min	7.300000e+01	3.700000e+01	0
25%	9.987875e+04	6.933075e+04	13
50%	8.988110e+05	5.660505e+05	53
75%	6.258963e+06	3.764875e+06	109
max	1.776816e+09	4.357201e+08	233

In [41]:
```
df.mean()
```

Out[41]:
```
TOTAL_VACCINATIONS                    2.044237e+07
PERSONS_VACCINATED_1PLUS_DOSE         8.253955e+06
TOTAL_VACCINATIONS_PER100             6.312819e+01
PERSONS_VACCINATED_1PLUS_DOSE_PER100  3.585499e+01
PERSONS_FULLY_VACCINATED              4.889511e+06
PERSONS_FULLY_VACCINATED_PER100       2.812897e+01
NUMBER_VACCINES_TYPES_USED            3.219731e+00
dtype: float64
```

In [42]:
```
df['NUMBER_VACCINES_TYPES_USED'].mean()
```

Out[42]:
```
3.219730941704036
```

In [46]:
```
df['PERSONS_FULLY_VACCINATED'].sum()
```

Out[46]:
```
1056134385.0
```

In [47]:
```
df.var()
```

Out[47]:
```
TOTAL_VACCINATIONS                      1.610112e+16
PERSONS_VACCINATED_1PLUS_DOSE           1.206527e+15
TOTAL_VACCINATIONS_PER100               2.663460e+03
PERSONS_VACCINATED_1PLUS_DOSE_PER100    7.228041e+02
PERSONS_FULLY_VACCINATED                2.694942e+14
PERSONS_FULLY_VACCINATED_PER100         6.370459e+02
NUMBER_VACCINES_TYPES_USED              2.901951e+00
dtype: float64
```

In [48]:
```
df.skew()
```

Out[48]:
```
TOTAL_VACCINATIONS                      12.334141
PERSONS_VACCINATED_1PLUS_DOSE            9.736706
TOTAL_VACCINATIONS_PER100                0.441166
PERSONS_VACCINATED_1PLUS_DOSE_PER100     0.226214
PERSONS_FULLY_VACCINATED                 7.062170
PERSONS_FULLY_VACCINATED_PER100          0.574913
NUMBER_VACCINES_TYPES_USED               0.650929
dtype: float64
```

In [49]:
```
df.kurtosis()
```

Out[49]:
```
TOTAL_VACCINATIONS                      166.261644
PERSONS_VACCINATED_1PLUS_DOSE           111.104264
TOTAL_VACCINATIONS_PER100                -0.910117
PERSONS_VACCINATED_1PLUS_DOSE_PER100     -1.150897
PERSONS_FULLY_VACCINATED                 60.759231
PERSONS_FULLY_VACCINATED_PER100          -0.757014
NUMBER_VACCINES_TYPES_USED                0.204438
dtype: float64
```

In [60]:
```
df['TOTAL_VACCINATIONS'].max()
```

Out[60]:
```
1776816416.0
```

In [61]:
```
df['TOTAL_VACCINATIONS'].min()
```

Out[61]:
```
73.0
```

In [50]:
```
df.min()
```

Out[50]:
```
COUNTRY                                 Afghanistan
ISO3                                            ABW
WHO_REGION                                     AFRO
DATA_SOURCE                                    OWID
DATE_UPDATED                             2021-04-04
TOTAL_VACCINATIONS                               73
PERSONS_VACCINATED_1PLUS_DOSE                    37
TOTAL_VACCINATIONS_PER100                     0.078
PERSONS_VACCINATED_1PLUS_DOSE_PER100          0.067
PERSONS_FULLY_VACCINATED                         36
PERSONS_FULLY_VACCINATED_PER100               0.007
NUMBER_VACCINES_TYPES_USED                        1
dtype: object
```

In [51]:
```
df.max()
```

Out[51]:
```
COUNTRY                                 occupied Palestinian territory
ISO3                                                               ZWE
WHO_REGION                                                        WPRO
DATA_SOURCE                                                  REPORTING
DATE_UPDATED                                              2021-08-23
TOTAL_VACCINATIONS                                       1.77682e+09
PERSONS_VACCINATED_1PLUS_DOSE                             4.3572e+08
TOTAL_VACCINATIONS_PER100                                    233.208
PERSONS_VACCINATED_1PLUS_DOSE_PER100                         116.933
PERSONS_FULLY_VACCINATED                                 1.70139e+08
PERSONS_FULLY_VACCINATED_PER100                              116.274
NUMBER_VACCINES_TYPES_USED                                         9
dtype: object
```

In [52]:
```
df.median()
```

Out[52]:
```
TOTAL_VACCINATIONS                 898811.0000
PERSONS_VACCINATED_1PLUS_DOSE      566050.5000
TOTAL_VACCINATIONS_PER100              53.3145
PERSONS_VACCINATED_1PLUS_DOSE_PER100   32.5940
PERSONS_FULLY_VACCINATED           324729.0000
PERSONS_FULLY_VACCINATED_PER100        22.0230
NUMBER_VACCINES_TYPES_USED              3.0000
dtype: float64
```

In [53]:
```
df.corr()
```

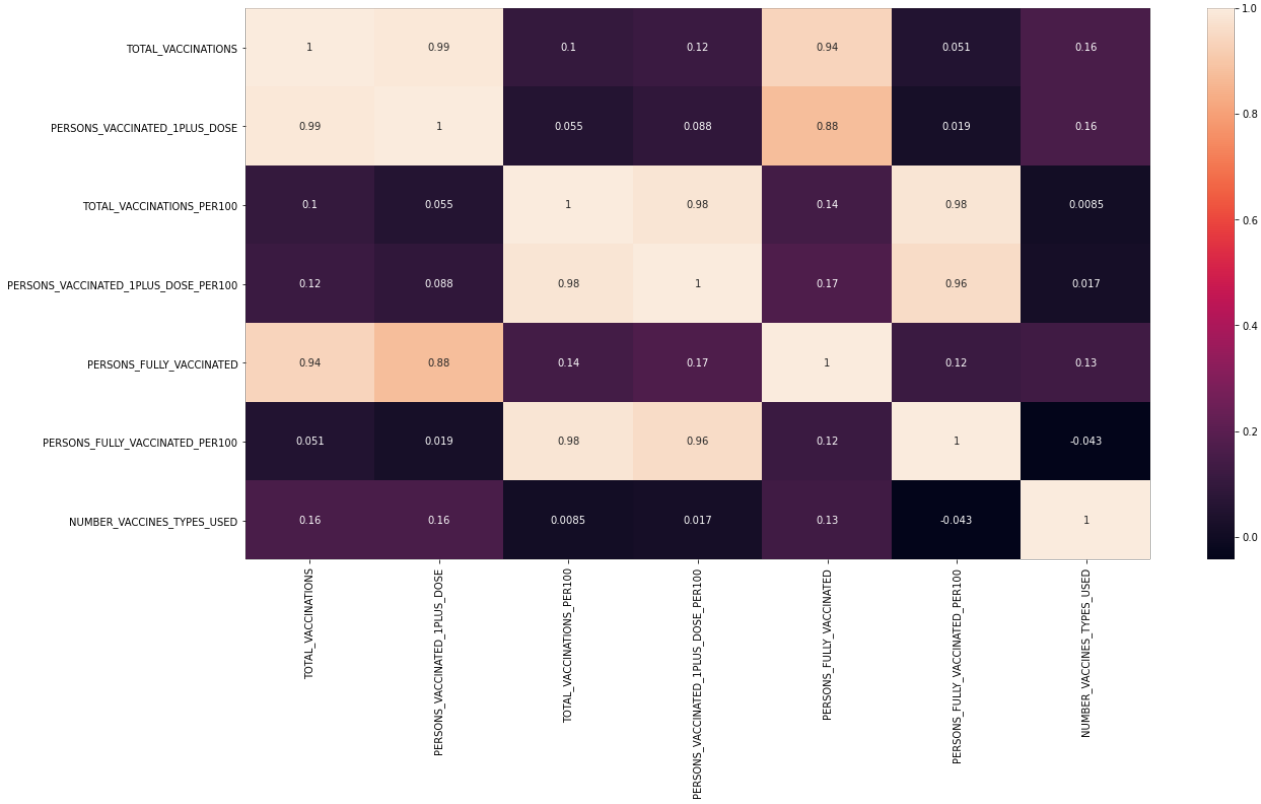
Out[53]:

	TOTAL_VACCINATIONS	PERSONS_VACCINATEI
TOTAL_VACCINATIONS	1.000000	
PERSONS_VACCINATED_1PLUS_DOSE	0.989262	
TOTAL_VACCINATIONS_PER100	0.101607	
PERSONS_VACCINATED_1PLUS_DOSE_PER100	0.116021	
PERSONS_FULLY_VACCINATED	0.936786	
PERSONS_FULLY_VACCINATED_PER100	0.050727	
NUMBER_VACCINES_TYPES_USED	0.160203	

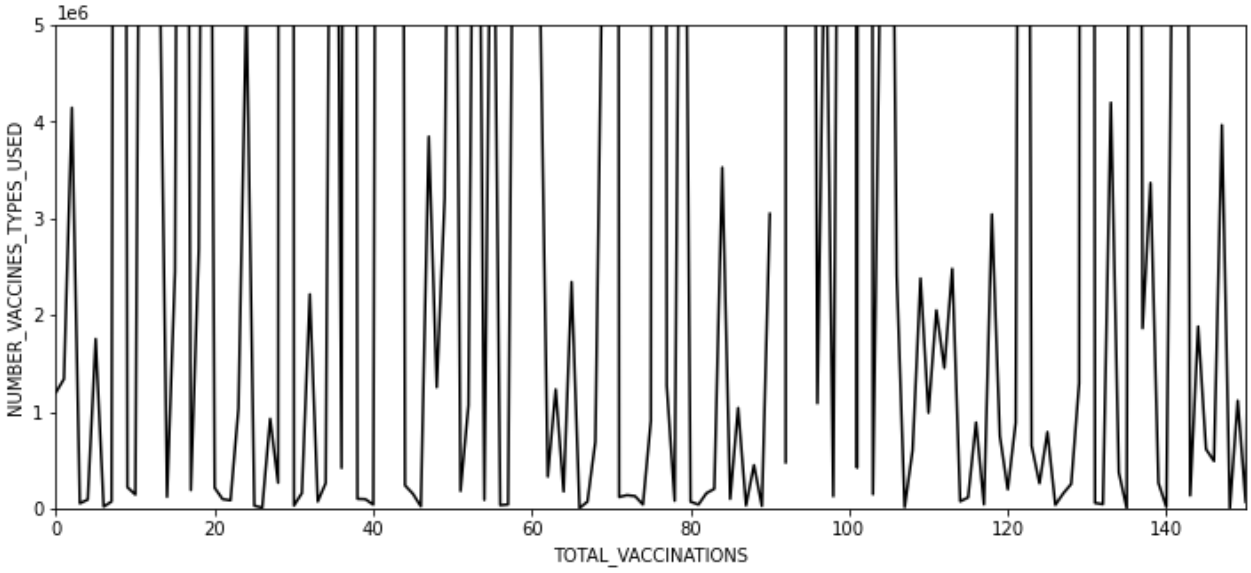
In [54]:
```
import seaborn as sns
```

In [55]:
```
sns.heatmap(df.corr(), annot=True)
```

Out[55]: <AxesSubplot:>

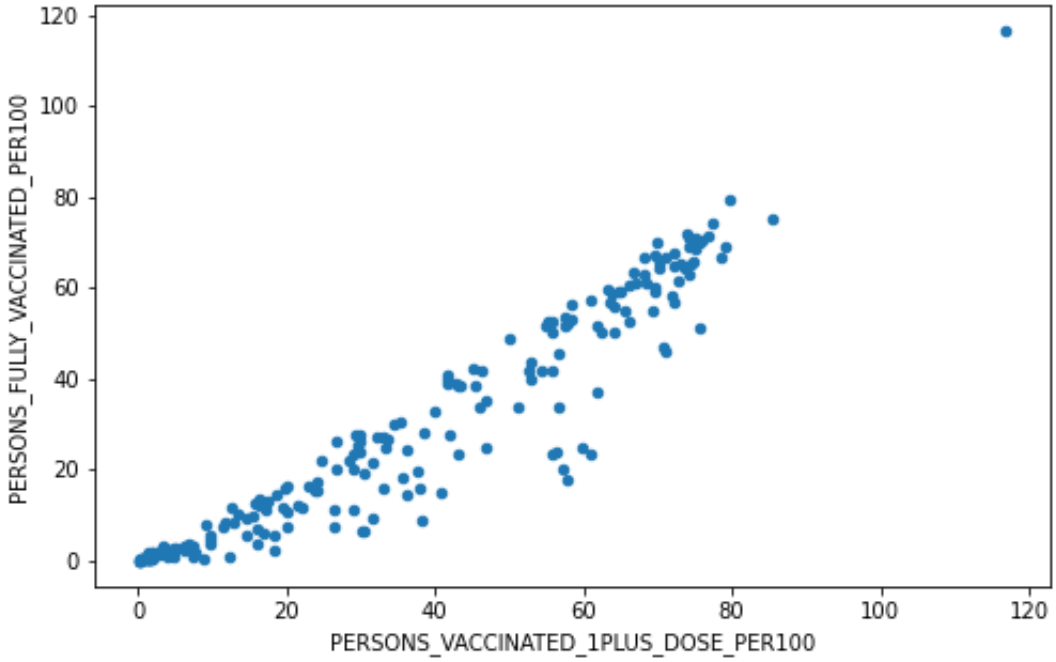


In [83]:
```
# Plotting with index along the x-axis
df['TOTAL_VACCINATIONS'].plot(figsize=(12, 5), color='black') # color and

plt.xlim(0, 150) # range for x-axis
plt.ylim(0, 5000000) # range for x-axis
plt.xlabel('TOTAL_VACCINATIONS')
plt.ylabel('NUMBER_VACCINES_TYPES_USED'); # ";" prevents object info from
```

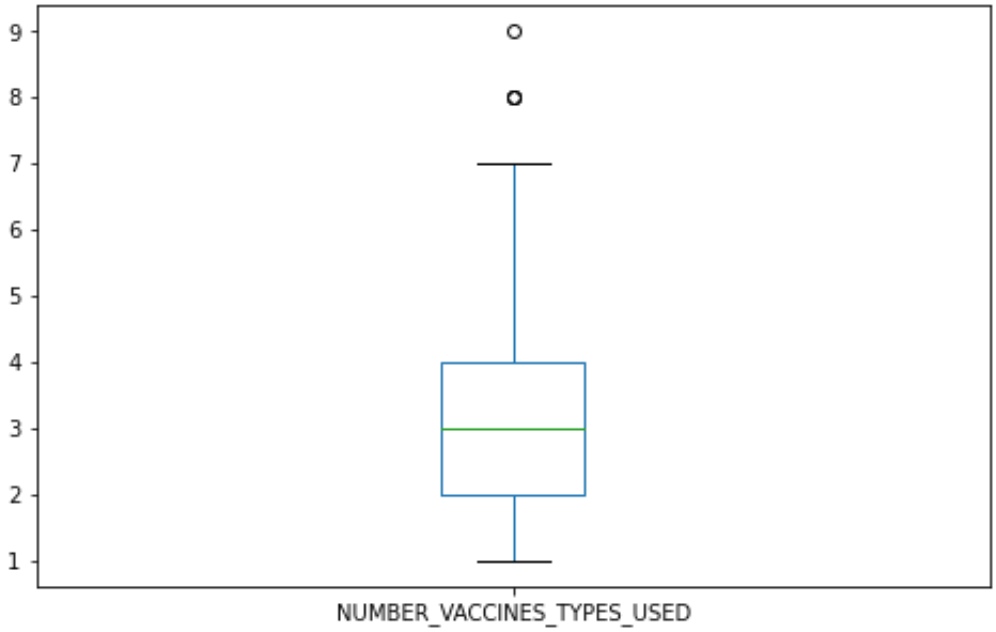


In [84]:
```
# plotting one variable against the other
df.plot.scatter('PERSONS_VACCINATED_1PLUS_DOSE_PER100', 'PERSONS_FULLY_VAC

# The x and y labels are automatically taken from the column names
```

Out[84]:
```
<AxesSubplot:xlabel='PERSONS_VACCINATED_1PLUS_DOSE_PER100', ylabel='PERSONS
_FULLY_VACCINATED_PER100'>
```

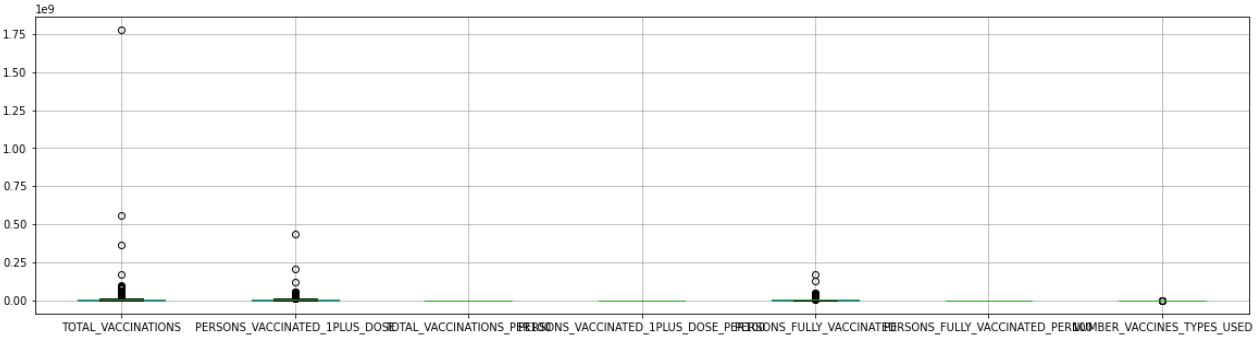


In [85]:
```
# Box plot of a column
df['NUMBER_VACCINES_TYPES_USED'].plot.box(figsize=(8, 5));
```

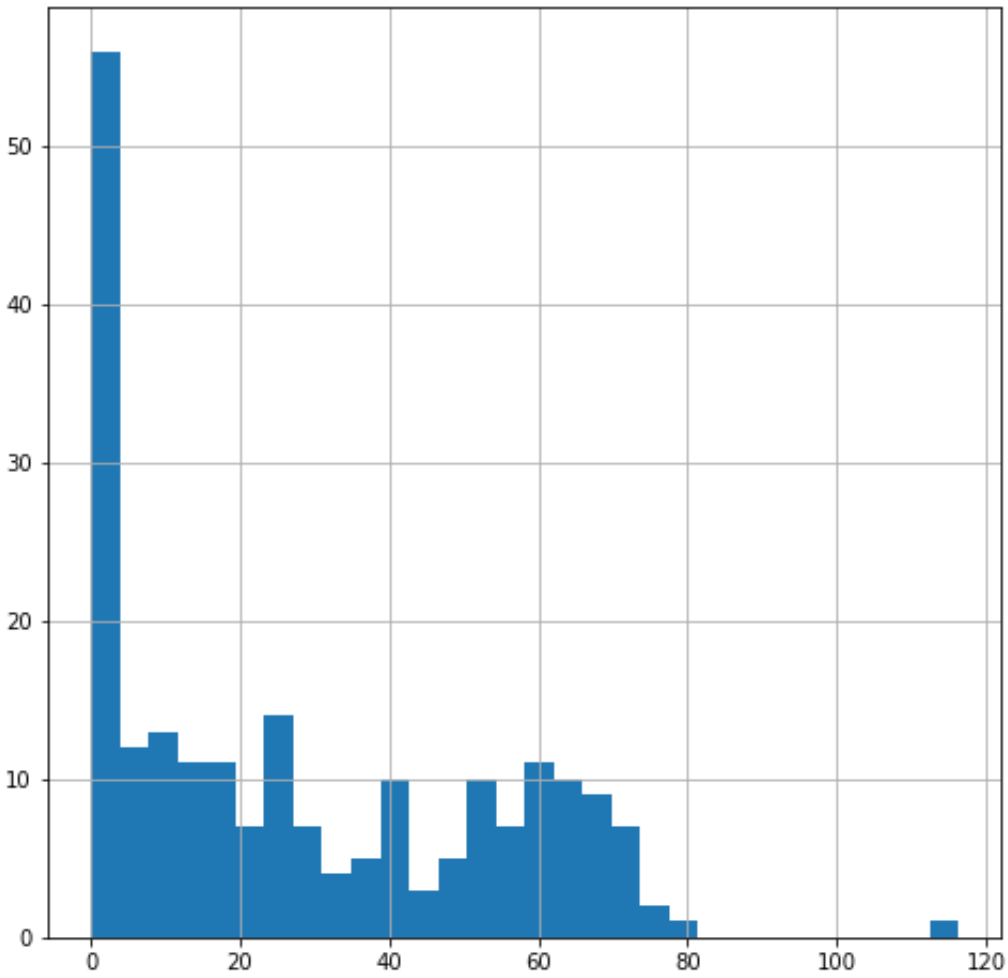


In [91]:
```
# Box plot of all the columns with numerical data
df.boxplot(figsize=(20, 5)) # or df.plot.box()
```

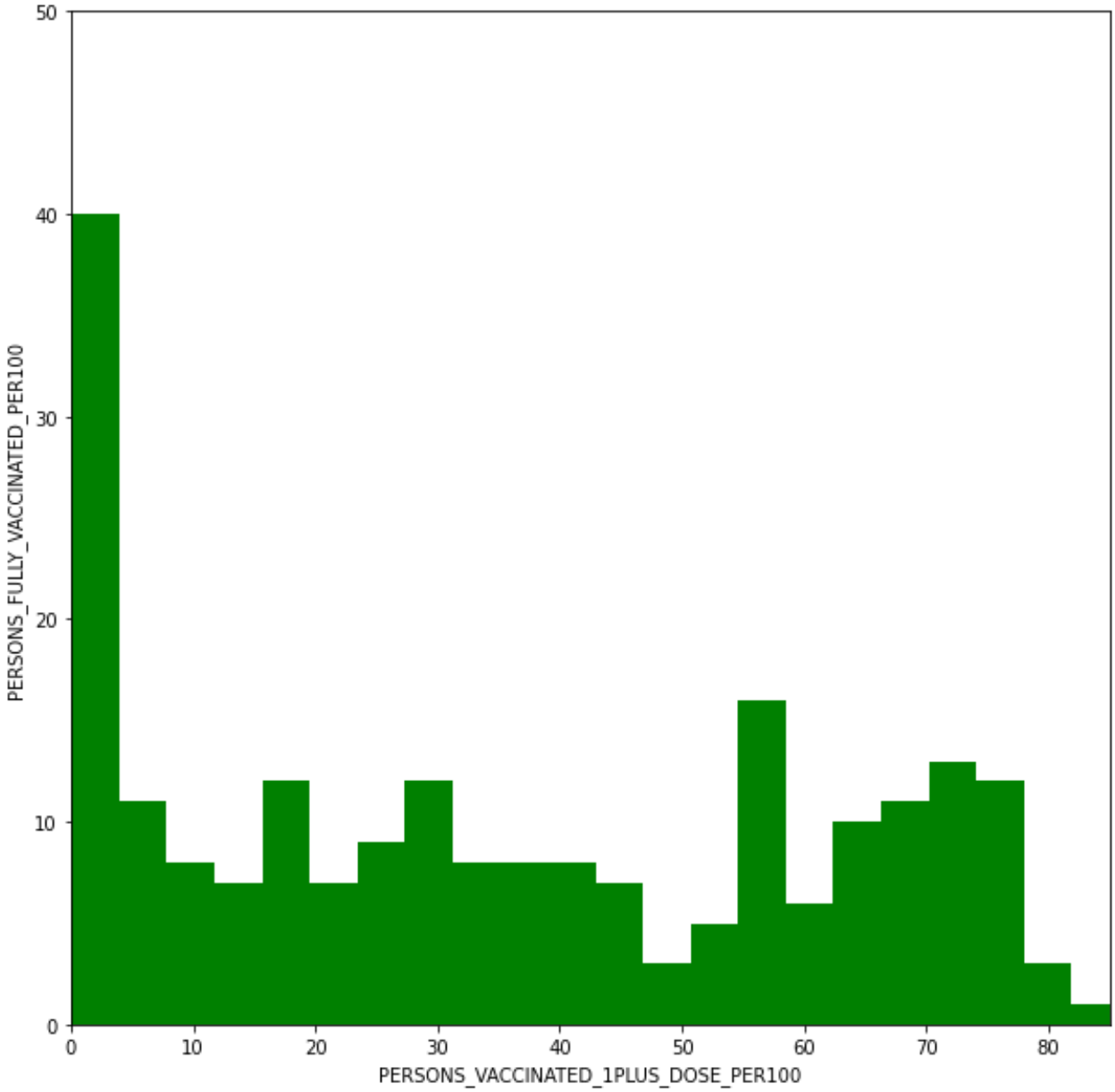
Out[91]: <AxesSubplot:>



In [97]:
```
df['PERSONS_FULLY_VACCINATED_PER100'].hist(bins=30, figsize=(8, 8)); # we
```

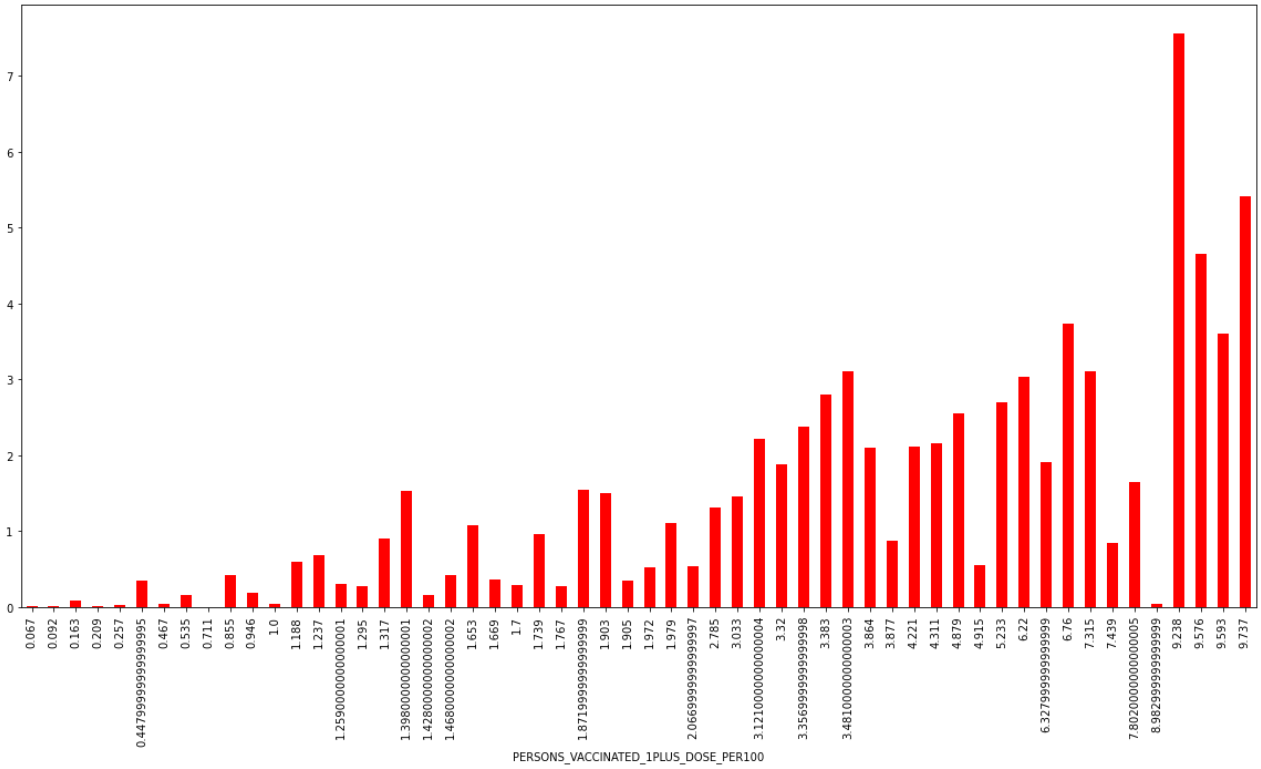


In [102…
```
ax = df['PERSONS_VACCINATED_1PLUS_DOSE_PER100'].hist(bins=30, grid=False, 

ax.set_xlabel('PERSONS_VACCINATED_1PLUS_DOSE_PER100')
ax.set_ylabel('PERSONS_FULLY_VACCINATED_PER100')

ax.set_xlim(0, 85) # limiting display range to 0-70 for the x-axis
ax.set_ylim(0, 50); # limiting display range to 0-120 for the y-axis
```



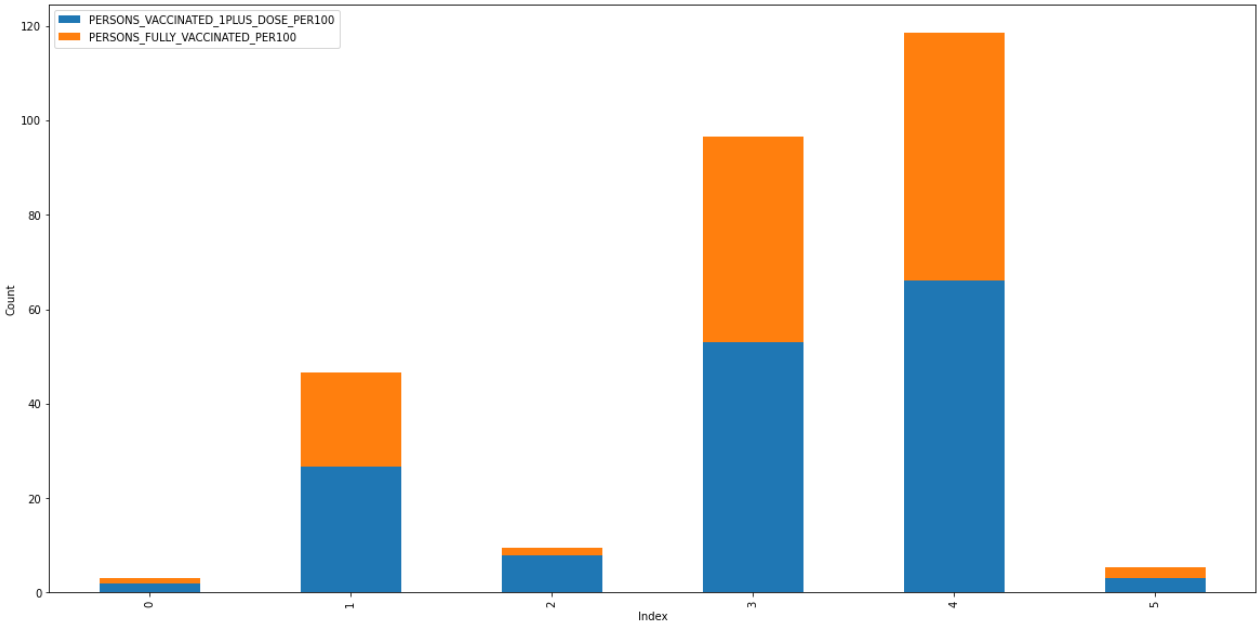
In [103…

```
df_avg_BP = df.groupby('PERSONS_VACCINATED_1PLUS_DOSE_PER100')['PERSONS_FU
df_avg_BP[:10].plot.bar(color='red');
```



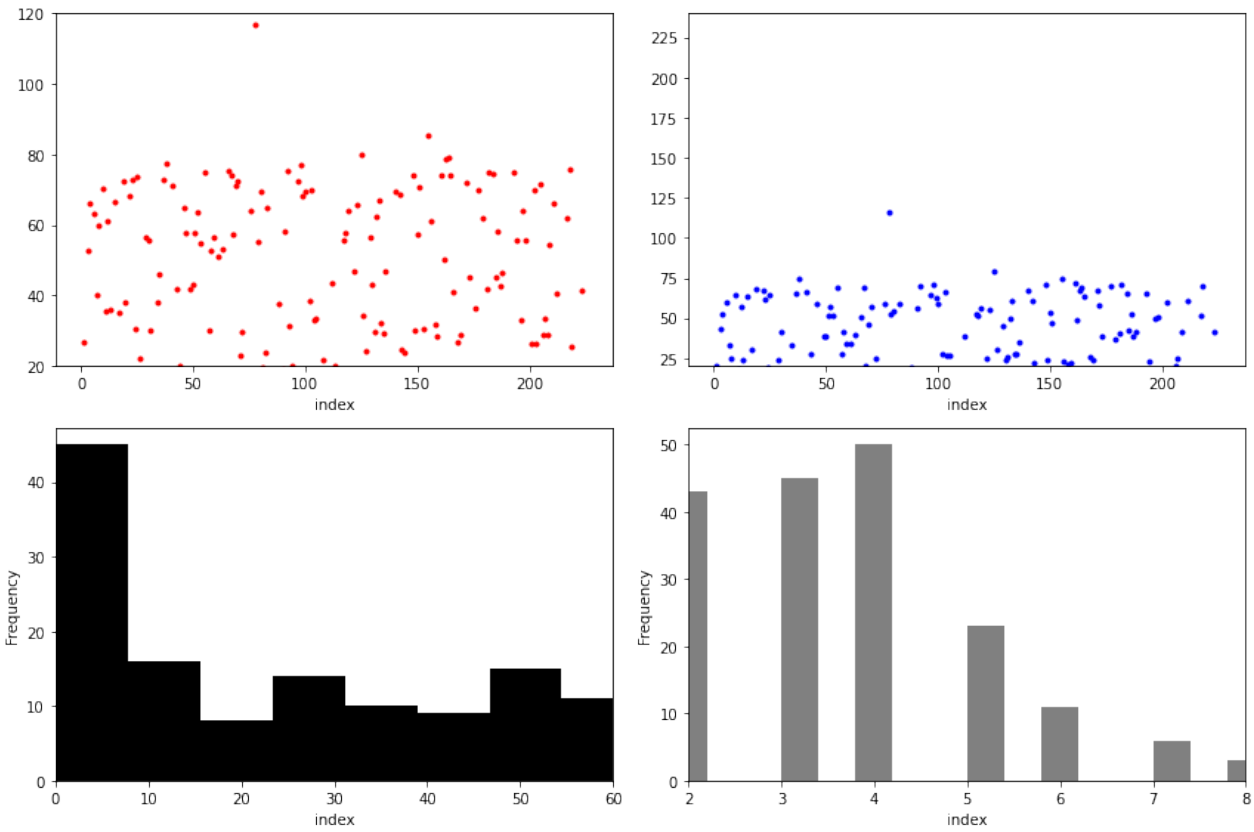
In [105…

```
# To stack values from multiple columns
ax = df[['PERSONS_VACCINATED_1PLUS_DOSE_PER100', 'PERSONS_FULLY_VACCINATED_

ax.set_xlabel("Index")
ax.set_ylabel("Count");
```



In [111…

```
fig, axes = plt.subplots(2, 2, figsize=(12, 8))
# or fig, (ax1, ax2, ax3, ax4) = plt.subplots(2, 2, figsize=(12, 8))

# axes is the axes object(s). It can be a single object or an array of obj
# In this case, it is an array of dimension 2-by-2

df['PERSONS_VACCINATED_1PLUS_DOSE_PER100'].plot(ax = axes[0][0], style='.'
df['PERSONS_FULLY_VACCINATED_PER100'].plot(ax = axes[0][1], style='.', col

df['TOTAL_VACCINATIONS_PER100'].plot.hist(bins=30, ax = axes[1][0], color=
df['NUMBER_VACCINES_TYPES_USED'].plot.hist(bins=20, ax = axes[1][1], color

axes[0][0].set_xlabel('index')
axes[0][1].set_xlabel('index')
axes[1][0].set_xlabel('index')
axes[1][1].set_xlabel('index')


axes[0][0].set_ylim(20, 120)
axes[0][1].set_ylim(20, 240)

axes[1][0].set_xlim(0, 60)
axes[1][1].set_xlim(2, 8)

fig.tight_layout()
```



In [ ]: