**Министерство науки и высшего образования Российской Федерации**
**Федеральное государственное бюджетное образовательное учреждение**
**высшего образования**
**«Московский государственный технический университет**
**имени Н.Э. Баумана**
**(национальный исследовательский университет)»**
**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

# Отчёт по лабораторной работе №1

По дисциплине:
«Технологии машинного обучения»

Выполнил:

Студент группы ИУ5-65Б **Петренко С.С**

(Подпись, дата) (Фамилия И.О.)

Проверил:

**Гапанюк Ю. Е.**

(Подпись, дата) (Фамилия И.О.)

Москва, 2021

**Задание**

- Выбрать набор данных
- Создать ноутбук, который содержит следующие разделы:
    1. Текстовое описание набора данных
    2. Основные характеристики набора данных
    3. Визуальное исследование набора данных
    4. Информацию о корреляции признаков

# ЛР № 1

## 1) Текстовое описание набора данных

В качестве набора данных использхуется набор данных со статистикой уровня Мировой прогресс в вакцинации против COVID-19. Содержание

Эти данные содержат следующую информацию:

-Страна - это страна, для которой предоставляется информация о вакцинации; - Country ISO Code - код ISO для страны; Дата - дата ввода данных; для некоторых дат у нас есть только ежедневные прививки, для других - только (кумулятивная) общая сумма; Общее количество прививок - это абсолютное количество общих прививок в стране; Общее количество вакцинированных - человек, в зависимости от схемы иммунизации, получит одну или несколько (обычно 2) вакцин; в определенный момент количество вакцинированных может быть больше, чем количество людей; Общее число полностью вакцинированных людей - это число людей, получивших весь набор иммунизации по схеме иммунизации (обычно 2); в определенный момент времени может быть определенное число людей, получивших одну вакцину, и другое число (меньшее) людей, получивших все вакцины по схеме иммунизации.; Ежедневные прививки (raw) - для определенного ввода данных количество прививок на эту дату/страну; Ежедневные прививки - для определенного ввода данных, количество прививок на эту дату/страну; Общее количество прививок на сотню - соотношение (в процентах) между количеством прививок и общей численностью населения на сегодняшний день в стране; Общее количество вакцинированных на сто человек - соотношение (в процентах) между привитым населением и общей численностью населения на сегодняшний день в стране; Общее количество полностью вакцинированных людей на сотню - соотношение (в процентах) между полностью иммунизированным населением и общей численностью населения на сегодняшний день в стране; Количество прививок в день - количество ежедневных прививок за этот день и страну; Ежедневные прививки на миллион - соотношение (в промилле) между количеством прививок и общей численностью населения на текущую дату в стране; Вакцины, используемые в стране - общее количество вакцин, используемых в стране (на сегодняшний день); Название источника - источник информации (национальный орган, международная организация, местная организация и т. д.); Source website - сайт источника информации;

### Импорт библиотек

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
import warnings
warnings.filterwarnings('ignore')
sns.set(style="ticks")
%matplotlib inline
```

### Загрузка данных

In [2]:
```
happy_data = pd.read_csv('country_vaccinations.csv', sep = ',' )
```

## 2)Основные характеристики датасета

In [3]:
```
# Первые пять строк датасета
happy_data.head()
```

Out[3]:

| | country | iso_code | date | total_vaccinations | people_vaccinated | people_fully_vaccinated |
|---|---|---|---|---|---|---|
| 0 | Albania | ALB | 2021-01-10 | 0.0 | 0.0 | NaN |
| 1 | Albania | ALB | 2021-01-11 | NaN | NaN | NaN |
| 2 | Albania | ALB | 2021-01-12 | 128.0 | 128.0 | NaN |
| 3 | Albania | ALB | 2021-01-13 | 188.0 | 188.0 | NaN |
| 4 | Albania | ALB | 2021-01-14 | 266.0 | 266.0 | NaN |

In [4]:
```
# Размер датасета
happy_data.shape
```

Out[4]:
```
(3555, 15)
```

In [5]:
```
# Количество нулевых элементов
happy_data.isnull().sum()
```

Out[5]:
```
country                                  0
iso_code                               272
date                                     0
total_vaccinations                    1214
people_vaccinated                     1615
people_fully_vaccinated               2277
daily_vaccinations_raw                1583
daily_vaccinations                     135
total_vaccinations_per_hundred        1214
people_vaccinated_per_hundred         1615
people_fully_vaccinated_per_hundred   2277
daily_vaccinations_per_million         135
vaccines                                 0
source_name                              0
source_website                           0
dtype: int64
```

In [6]:
```
# Колонки и их типы данных
happy_data.dtypes
```

Out[6]:
```
country                                 object
iso_code                                object
date                                    object
total_vaccinations                     float64
people_vaccinated                      float64
people_fully_vaccinated                float64
daily_vaccinations_raw                 float64
daily_vaccinations                     float64
total_vaccinations_per_hundred         float64
people_vaccinated_per_hundred          float64
people_fully_vaccinated_per_hundred    float64
daily_vaccinations_per_million         float64
vaccines                                object
source_name                             object
source_website                          object
dtype: object
```

In [7]:
```
# Описание датасета
happy_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3555 entries, 0 to 3554
Data columns (total 15 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   country                               3555 non-null   object
 1   iso_code                              3283 non-null   object
 2   date                                  3555 non-null   object
 3   total_vaccinations                    2341 non-null   float64
 4   people_vaccinated                     1940 non-null   float64
 5   people_fully_vaccinated               1278 non-null   float64
 6   daily_vaccinations_raw                1972 non-null   float64
 7   daily_vaccinations                    3420 non-null   float64
 8   total_vaccinations_per_hundred        2341 non-null   float64
 9   people_vaccinated_per_hundred         1940 non-null   float64
 10  people_fully_vaccinated_per_hundred   1278 non-null   float64
 11  daily_vaccinations_per_million        3420 non-null   float64
 12  vaccines                              3555 non-null   object
 13  source_name                           3555 non-null   object
 14  source_website                        3555 non-null   object
dtypes: float64(9), object(6)
memory usage: 416.7+ KB
```

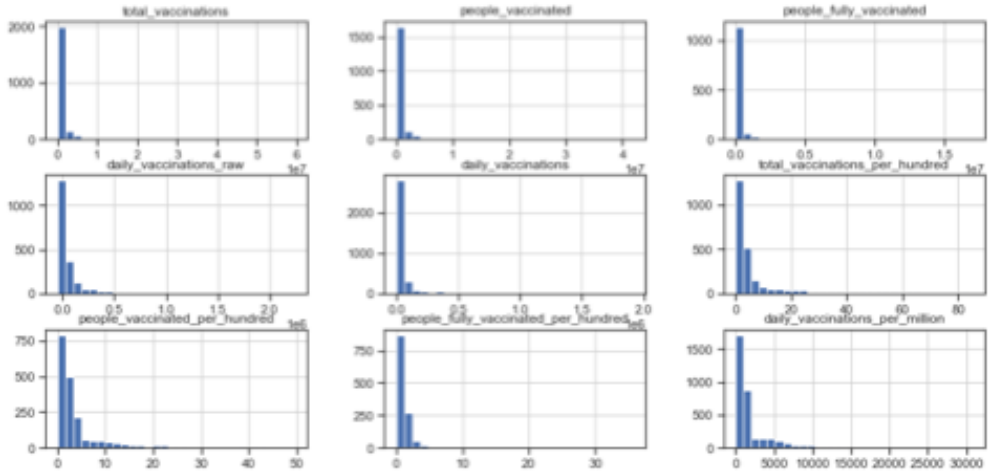
4.250e+00, 4.600e+00, 4.950e+00, 5.250e+00, 5.440e+00, 5.750e+00,
5.960e+00, 6.440e+00, 7.220e+00, 8.060e+00, 8.320e+00, 8.390e+00,
8.440e+00, 8.470e+00, 9.750e+00, 9.860e+00, 9.960e+00, 9.980e+00,
1.002e+01, 1.008e+01, 1.016e+01, 1.023e+01, 1.029e+01, 1.042e+01,
1.055e+01, 1.073e+01, 1.125e+01, 1.182e+01, 1.251e+01, 1.291e+01,
1.322e+01, 1.367e+01, 1.424e+01, 1.462e+01, 1.475e+01, 1.487e+01,
1.514e+01, 1.548e+01, 1.577e+01, 2.100e-01, 4.500e-01, 6.900e-01,
9.600e-01, 1.120e+00, 1.560e+00, 2.330e+00, 3.170e+00, 5.000e-02,
1.200e-01, 2.000e-01, 2.300e-01, 2.800e-01, 4.100e-01, 7.900e-01,
1.030e+00, 1.060e+00, 1.150e+00, 1.320e+00, 1.490e+00, 1.750e+00,
2.000e+00, 2.070e+00, 2.090e+00, 2.180e+00, 2.280e+00, 2.420e+00,
2.610e+00, 2.780e+00, 2.820e+00, 2.880e+00, 3.320e+00, 3.590e+00,
3.840e+00, 3.890e+00, 3.920e+00, 4.000e+00, 4.220e+00, 4.420e+00,
4.700e+00, 4.970e+00, 5.030e+00, 5.050e+00, 5.150e+00, 5.370e+00,
2.670e+00, 4.710e+00, 7.860e+00, 1.461e+01, 1.606e+01, 2.112e+01,
3.000e-02, 2.500e-01, 3.300e-01, 4.000e-01, 5.300e-01, 7.800e-01,
9.400e-01, 1.000e+00, 1.190e+00, 1.450e+00, 1.600e+00, 1.800e+00,
1.940e+00, 2.210e+00, 2.410e+00, 2.460e+00, 2.490e+00, 2.640e+00,
2.770e+00, 2.920e+00, 3.070e+00, 8.000e-02, 1.600e-01, 1.900e-01,
2.600e-01, 2.700e-01, 3.200e-01, 3.500e-01, 3.600e-01, 3.800e-01,
3.900e-01, 4.200e-01, 4.700e-01, 5.400e-01, 6.100e-01, 6.600e-01,
7.200e-01, 8.200e-01, 8.500e-01, 9.000e-01, 9.500e-01, 1.020e+00,
1.070e+00, 1.170e+00, 1.240e+00, 1.300e+00, 1.500e+00, 7.100e-01,
1.220e+00, 1.290e+00, 1.990e+00, 2.190e+00, 2.360e+00, 2.480e+00,
2.580e+00, 2.650e+00, 2.690e+00, 2.760e+00, 3.060e+00, 3.140e+00,
3.240e+00, 3.520e+00, 3.720e+00, 1.675e+01, 2.071e+01, 2.143e+01,
2.365e+01, 2.545e+01, 2.630e+01, 1.000e-01, 3.400e-01, 2.230e+00,
3.110e+00, 3.250e+00, 3.300e+00, 4.510e+00, 5.870e+00, 7.320e+00,
8.630e+00, 9.820e+00, 1.000e+01, 1.137e+01, 1.243e+01, 3.100e-01,
6.300e-01, 1.580e+00, 2.170e+00, 4.900e-01, 5.800e-01, 1.470e+00,
3.000e-01, 5.000e-01, 9.300e-01, 1.010e+00, 1.950e+00, 2.040e+00,
2.140e+00, 2.750e+00, 2.030e+00, 5.070e+00, 1.300e-01, 4.300e-01,
4.600e-01, 5.600e-01, 1.720e+00, 1.860e+00, 1.930e+00, 2.050e+00,
2.300e+00, 2.430e+00, 2.550e+00, 2.570e+00, 2.590e+00, 2.830e+00,
3.130e+00, 3.280e+00, 3.340e+00, 3.480e+00, 3.630e+00, 3.800e+00,
4.190e+00, 4.240e+00, 4.260e+00, 4.400e+00, 4.560e+00, 4.890e+00,
1.500e-01, 8.000e-01, 1.520e+00, 2.010e+00, 2.250e+00, 2.560e+00,
2.960e+00, 3.200e+00, 3.270e+00, 3.380e+00, 3.640e+00, 3.690e+00,
3.950e+00, 4.280e+00, 4.530e+00, 4.760e+00, 4.910e+00, 5.000e+00,
5.130e+00, 5.350e+00, 5.610e+00, 5.740e+00, 5.840e+00, 6.000e+00,
6.150e+00, 6.270e+00, 6.410e+00, 6.750e+00, 6.880e+00, 7.000e+00,
7.290e+00, 7.540e+00, 7.730e+00, 7.920e+00, 2.120e+00, 4.730e+00,
5.670e+00, 6.240e+00, 6.740e+00, 7.010e+00, 7.850e+00, 8.420e+00,
9.060e+00, 1.061e+01, 1.096e+01, 1.138e+01, 1.184e+01, 1.229e+01,
1.290e+01, 1.368e+01, 1.466e+01, 1.518e+01, 1.568e+01, 1.621e+01,
1.689e+01, 1.759e+01, 1.830e+01, 1.912e+01, 1.953e+01, 1.998e+01,
2.054e+01, 2.114e+01, 2.184e+01, 2.262e+01, 2.339e+01, 2.372e+01,
2.412e+01, 2.468e+01, 2.544e+01, 2.615e+01, 1.400e-01, 6.000e-01,
7.700e-01, 1.260e+00, 1.370e+00, 1.380e+00, 1.460e+00, 1.540e+00,
1.660e+00, 1.960e+00, 2.380e+00, 2.870e+00, 3.990e+00, 4.030e+00,
4.390e+00, 4.660e+00, 4.960e+00, 5.140e+00, 5.160e+00, 5.230e+00,
5.410e+00, 5.730e+00, 6.210e+00, 8.170e+00, 9.530e+00, 9.790e+00,
1.650e+00, 2.110e+00, 2.740e+00, 2.930e+00, 3.500e+00, 3.740e+00,
3.930e+00, 3.970e+00, 3.980e+00, 4.100e+00, 4.720e+00, 4.930e+00,
4.990e+00, 5.120e+00, 5.380e+00, 6.020e+00, 1.800e-01, 8.600e-01,
1.050e+00, 1.250e+00, 1.730e+00, 1.880e+00, 2.060e+00, 2.260e+00,
2.370e+00, 2.890e+00, 3.470e+00, 3.960e+00, 4.270e+00, 4.440e+00,
4.680e+00, 5.180e+00, 2.200e-01, 6.700e-01, 7.400e-01, 1.420e+00,
1.510e+00, 1.620e+00, 1.780e+00, 1.890e+00, 2.710e+00, 2.840e+00,
3.010e+00, 3.780e+00, 3.900e+00, 4.590e+00, 4.780e+00, 5.300e+00,
5.470e+00, 5.630e+00, 4.290e+00, 7.420e+00, 1.524e+01, 1.735e+01,
1.910e+01, 2.083e+01, 2.359e+01, 2.635e+01, 2.911e+01, 3.069e+01,
3.128e+01, 3.287e+01, 3.445e+01, 3.660e+01, 3.819e+01, 3.943e+01,
4.244e+01, 4.568e+01, 4.802e+01, 5.102e+01, 5.424e+01, 5.661e+01,
5.949e+01, 6.239e+01, 6.530e+01, 6.857e+01, 7.181e+01, 7.506e+01,

```
1.180e+01, 1.410e+01, 1.545e+01, 1.684e+01, 1.818e+01, 1.904e+01,
1.993e+01, 2.088e+01, 2.185e+01, 2.271e+01, 2.515e+01, 2.600e+01,
2.707e+01, 2.795e+01, 2.900e+01, 3.040e+01, 3.149e+01, 3.371e+01,
3.479e+01, 3.604e+01, 3.892e+01, 4.248e+01, 4.362e+01, 4.463e+01,
4.577e+01, 4.737e+01, 4.845e+01, 4.956e+01, 5.061e+01, 5.111e+01,
5.143e+01, 5.256e+01, 5.343e+01, 5.433e+01, 5.527e+01, 4.520e+00,
5.420e+00, 5.930e+00, 6.960e+00, 7.470e+00, 8.010e+00, 8.620e+00,
9.320e+00, 1.005e+01, 1.038e+01, 1.079e+01, 1.172e+01, 1.233e+01,
1.305e+01, 1.395e+01, 1.442e+01, 1.494e+01, 1.550e+01, 1.619e+01,
1.691e+01, 1.764e+01, 1.845e+01, 1.886e+01, 1.939e+01, 2.000e+01,
2.067e+01, 2.142e+01, 2.223e+01, 2.298e+01, 2.333e+01, 2.375e+01,
2.430e+01, 2.504e+01, 2.573e+01, 2.790e+00, 3.330e+00, 3.670e+00,
6.140e+00, 6.530e+00, 6.800e+00, 7.040e+00, 7.370e+00, 7.830e+00,
8.340e+00, 8.840e+00, 9.310e+00, 9.800e+00, 1.013e+01, 1.053e+01,
1.101e+01, 1.167e+01, 1.232e+01, 1.268e+01, 1.292e+01, 1.387e+01,
1.448e+01, 1.581e+01, 1.651e+01, 1.683e+01, 1.726e+01, 1.782e+01,
7.640e+00, 8.410e+00, 8.610e+00, 9.200e+00, 9.930e+00, 1.151e+01,
1.282e+01, 1.398e+01, 1.471e+01, 1.560e+01, 1.664e+01, 1.774e+01,
1.878e+01, 1.924e+01, 2.005e+01, 2.090e+01, 2.182e+01, 2.283e+01,
2.390e+01, 2.462e+01, 2.506e+01, 2.547e+01, 2.670e+01, 2.742e+01])
```

## 3) Визуальное исследование датасета

In [12]:
```
# Гистограммы для всех признаков
happy_data.hist(bins=30, figsize = (15,7))
```

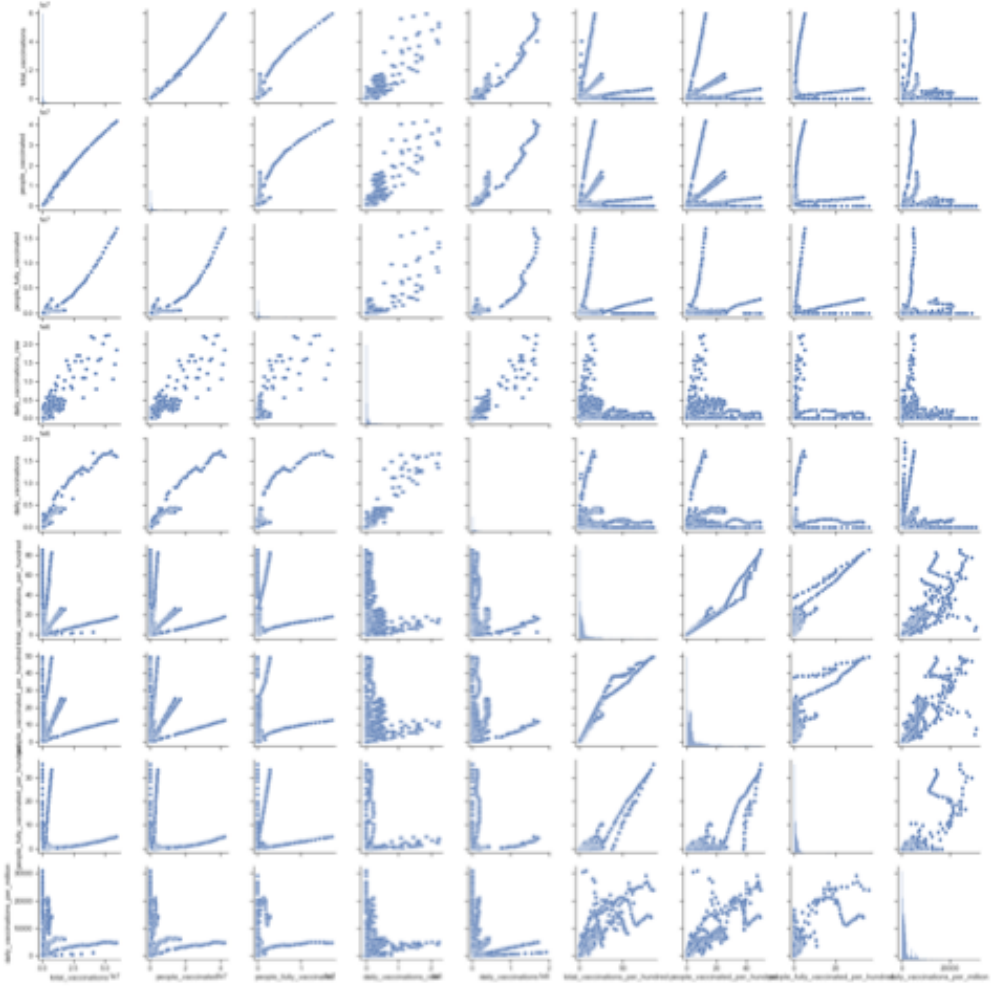
Out[12]:
```
array([[<AxesSubplot:title={'center':'total_vaccinations'}>,
        <AxesSubplot:title={'center':'people_vaccinated'}>,
        <AxesSubplot:title={'center':'people_fully_vaccinated'}>],
       [<AxesSubplot:title={'center':'daily_vaccinations_raw'}>,
        <AxesSubplot:title={'center':'daily_vaccinations'}>,
        <AxesSubplot:title={'center':'total_vaccinations_per_hundred'}>],
       [<AxesSubplot:title={'center':'people_vaccinated_per_hundred'}>,
        <AxesSubplot:title={'center':'people_fully_vaccinated_per_hundred'}
>,
        <AxesSubplot:title={'center':'daily_vaccinations_per_million'}>]],
      dtype=object)
```




In [13]:
```
# Диаграммы рассеяние для всех признаков
plt.figure(figsize=(12,6))
sns.pairplot(happy_data)
```

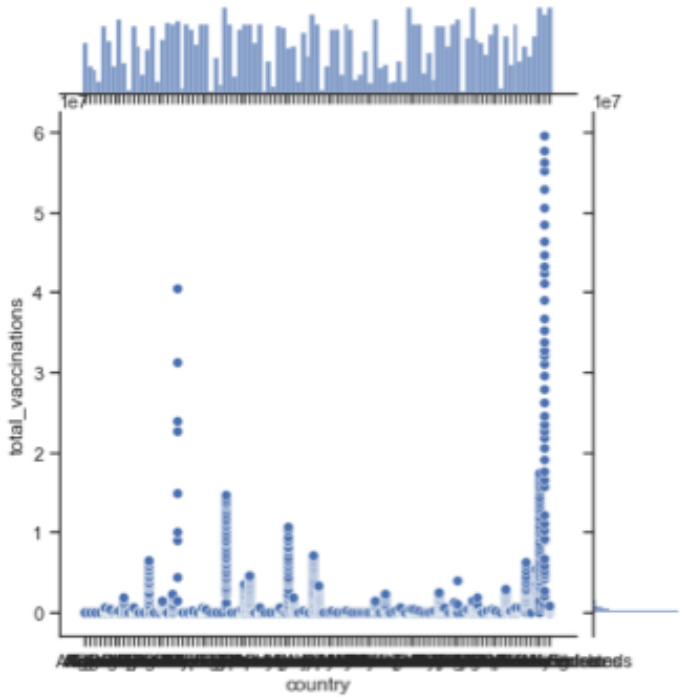
Out[13]: <seaborn.axisgrid.PairGrid at 0x7f8f0ebb7ee0>

<Figure size 864x432 with 0 Axes>



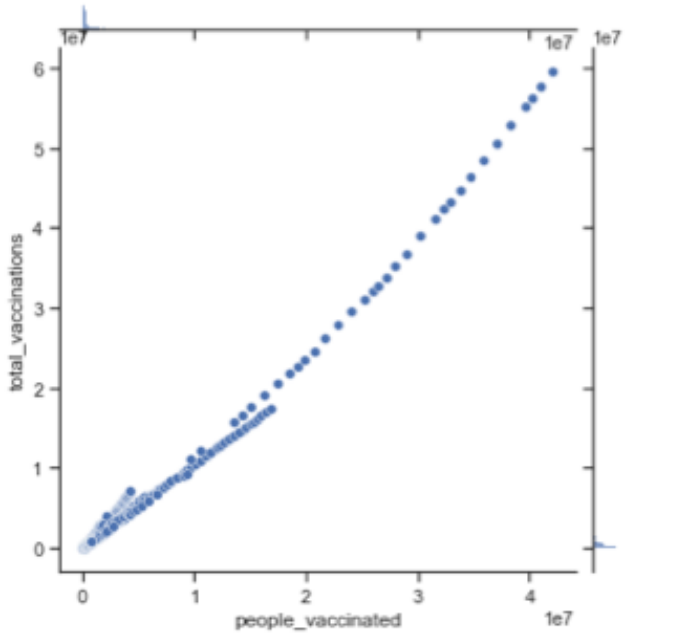
In [14]:
```
# Увеличенные диаграммы рассеяния для признаков, которые имеют зависимость
sns.jointplot(x = "country", y = "total_vaccinations", kind="scatter", data
```

Out[14]: <seaborn.axisgrid.JointGrid at 0x7f8f11d49730>



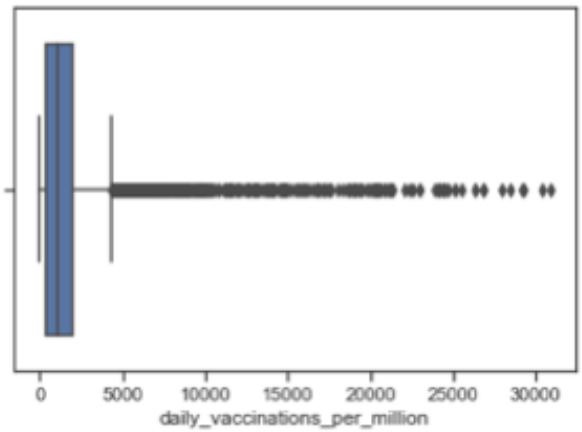

In [15]:
```
sns.jointplot(x = "people_vaccinated", y = "total_vaccinations", kind="sca
```

Out[15]: <seaborn.axisgrid.JointGrid at 0x7f8ef7422e20>



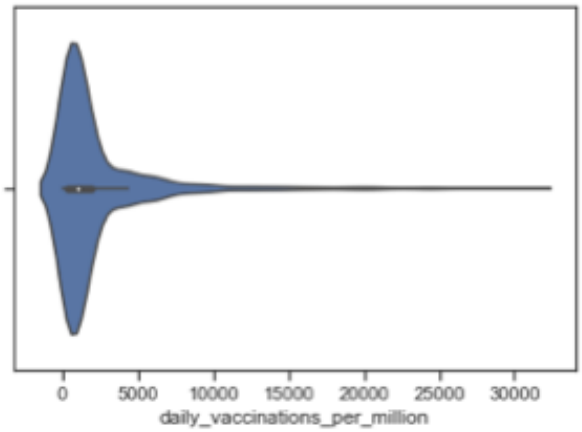
In [16]:
```
# Одномерное распределение вероятности
sns.boxplot(x=happy_data['daily_vaccinations_per_million'])
```

Out[16]:
```
<AxesSubplot:xlabel='daily_vaccinations_per_million'>
```




In [17]:
```
sns.violinplot(x=happy_data['daily_vaccinations_per_million'])
```

Out[17]:
```
<AxesSubplot:xlabel='daily_vaccinations_per_million'>
```




## 4) Корреляции признаков

In [18]:
```
corr_matrix = happy_data.corr()
```

In [19]:
```
corr_matrix['total_vaccinations']
```