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 «Радиотехнический»

КАФЕДРА ИУ-5 «Системы обработки информации и управления»

Лабораторная работа №1 по курсу

Технологии машинного обучения

Тема работы: "Разведочный анализ данных. Исследование и визуализация данных."

Выполнил: Лисин А. В.
Группа: РТ5-61Б

Дата выполнения: «__» _____ 2021 г.

Подпись: _____

Проверил:

Дата проверки: «__» _____ 2021 г.

Подпись: _____

Москва, 2021 г.

Содержание

Описание задания

Цель лабораторной работы: изучение различных методов визуализация данных.

Краткое описание. Построение основных графиков, входящих в этап разведочного анализа данных.

Задание:

- Выбрать набор данных (датасет).
- Создать ноутбук, который содержит следующие разделы:

1. Текстовое описание выбранного Вами набора данных.
2. Основные характеристики датасета.
3. Визуальное исследование датасета.
4. Информация о корреляции признаков.

- Сформировать отчет и разместить его в своем репозитории на github.

Ход выполнения работы

1. Текстовое описание набора данных

В лабораторной работе используется датасет "2021 World Population".

Имеются следующие данные:

1. ISO код страны (iso_code)
2. Страна (country)
3. Обновленные данные за 2021-й год о населении (2021_last_updated)
4. Данные за 2020-й год о населении (2020_population)
5. Площадь страны (area)
6. Плотность населения на квадратный километр (density_sq_km)
7. Рост населения (growth_rate)
8. Процент населения относительно мира (world_%)
9. Номер по рейтингу (rank)

Загрузим выбранный датасет

In [126…
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

In [127…
```
data = pd.read_csv('2021_population.csv', sep = ",")
```

2. Основные характеристики набора данных

In [128…
```
data.shape
```

Out[128…
```
(229, 9)
```

In [129…
```
data.dtypes
```

Out[129…
```
iso_code             object
country              object
2021_last_updated    object
2020_population      object
area                 object
density_sq_km        object
growth_rate          object
world_%              object
rank                  int64
dtype: object
```

In [130…
```
data.isnull().sum()
```

Out[130…
```
iso_code             0
country              0
2021_last_updated    0
2020_population      0
area                 0
density_sq_km        0
growth_rate          0
world_%              0
rank                 0
dtype: int64
```

In [131…
```
data.head()
```

Out[131…

	iso_code	country	2021_last_updated	2020_population	area	density_sq_km	growth_rate	world_%	rank
0	CHN	China	1,443,813,474	1,439,323,776	9,706,961 sq_km	149/sq_km	0.34%	18.34%	1
1	IND	India	1,392,292,448	1,380,004,385	3,287,590 sq_km	424/sq_km	0.97%	17.69%	2
2	USA	United States	332,753,502	331,002,651	9,372,610 sq_km	36/sq_km	0.58%	4.23%	3
3	IDN	Indonesia	276,125,016	273,523,615	1,904,569 sq_km	145/sq_km	1.04%	3.51%	4
4	PAK	Pakistan	224,843,141	220,892,340	881,912 sq_km	255/sq_km	1.95%	2.86%	5

Преобразуем цифровые значения к рабочему виду:

In [132…
```
for i in range(len(data['2020_population'])):
    data['2020_population'][i] = data['2020_population'][i].replace(",", "")
    data['2020_population'][i] = str(int(float(data['2020_population'][i])))
```

```
<ipython-input-132-012f99fdf434>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['2020_population'][i] = data['2020_population'][i].replace(",", "")
<ipython-input-132-012f99fdf434>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['2020_population'][i] = str(int(float(data['2020_population'][i])))
```

In [133…
```
data['2020_population'] = data['2020_population'].astype(str).astype(int)
```

In [134…
```
for i in range(len(data['2021_last_updated'])):
    data['2021_last_updated'][i] = data['2021_last_updated'][i].replace(",", "")
    data['2021_last_updated'][i] = str(int(float(data['2021_last_updated'][i])))
```

```
<ipython-input-134-eeaf3535b530>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['2021_last_updated'][i] = data['2021_last_updated'][i].replace(",", "")
<ipython-input-134-eeaf3535b530>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['2021_last_updated'][i] = str(int(float(data['2021_last_updated'][i])))
```

In [135…
```
data['2021_last_updated'] = data['2021_last_updated'].astype(str).astype(int)
```

In [136…
```
for i in range(len(data['area'])):
    data['area'][i] = data['area'][i].replace(",", "")
    data['area'][i] = data['area'][i].replace(" sq_km", "")
    data['area'][i] = str(int(float(data['area'][i])))
```

```
<ipython-input-136-6ca7b74f9e1a>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['area'][i] = data['area'][i].replace(",", "")
<ipython-input-136-6ca7b74f9e1a>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['area'][i] = data['area'][i].replace(" sq_km", "")
<ipython-input-136-6ca7b74f9e1a>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['area'][i] = str(int(float(data['area'][i])))
```

In [137…
```
data['area'] = data['area'].astype(str).astype(int)
```

In [138…
```
for i in range(len(data['density_sq_km'])):
    data['density_sq_km'][i] = data['density_sq_km'][i].replace(",", "")
    data['density_sq_km'][i] = data['density_sq_km'][i].replace("/sq_km", "")
    data['density_sq_km'][i] = str(int(float(data['density_sq_km'][i])))
```

```
<ipython-input-138-3f16000d2a54>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['density_sq_km'][i] = data['density_sq_km'][i].replace(",", "")
<ipython-input-138-3f16000d2a54>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['density_sq_km'][i] = data['density_sq_km'][i].replace("/sq_km", "")
<ipython-input-138-3f16000d2a54>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['density_sq_km'][i] = str(int(float(data['density_sq_km'][i])))
```

In [139…
```
data['density_sq_km'] = data['density_sq_km'].astype(str).astype(int)
```

In [140…
```
for i in range(len(data['growth_rate'])):
    data['growth_rate'][i] = data['growth_rate'][i].replace("%", "")
    data['growth_rate'][i] = str(float(data['growth_rate'][i]))

data['growth_rate'] = data['growth_rate'].astype(str).astype(float)
```

```
<ipython-input-140-3f9e0a3df9f9>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['growth_rate'][i] = data['growth_rate'][i].replace("%", "")
<ipython-input-140-3f9e0a3df9f9>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['growth_rate'][i] = str(float(data['growth_rate'][i]))
```

In [141…
```
for i in range(len(data['world_%'])):
    data['world_%'][i] = data['world_%'][i].replace("%", "")
    data['world_%'][i] = str(float(data['world_%'][i]))

data['world_%'] = data['world_%'].astype(str).astype(float)
```

```
<ipython-input-141-27b1b79e590a>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['world_%'][i] = data['world_%'][i].replace("%", "")
<ipython-input-141-27b1b79e590a>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  data['world_%'][i] = str(float(data['world_%'][i]))
```

In [142…
```
data.dtypes
```

Out[142…
```
iso_code              object
country               object
2021_last_updated      int64
2020_population        int64
area                   int64
density_sq_km          int64
growth_rate          float64
world_%              float64
rank                   int64
dtype: object
```

In [143…
```
data.head()
```

Out[143…

	iso_code	country	2021_last_updated	2020_population	area	density_sq_km	growth_rate	world_%	rank
0	CHN	China	1443813474	1439323776	9706961	149	0.34	18.34	1
1	IND	India	1392292448	1380004385	3287590	424	0.97	17.69	2
2	USA	United States	332753502	331002651	9372610	36	0.58	4.23	3
3	IDN	Indonesia	276125016	273523615	1904569	145	1.04	3.51	4
4	PAK	Pakistan	224843141	220892340	881912	255	1.95	2.86	5

In [144…
```
data.describe()
```

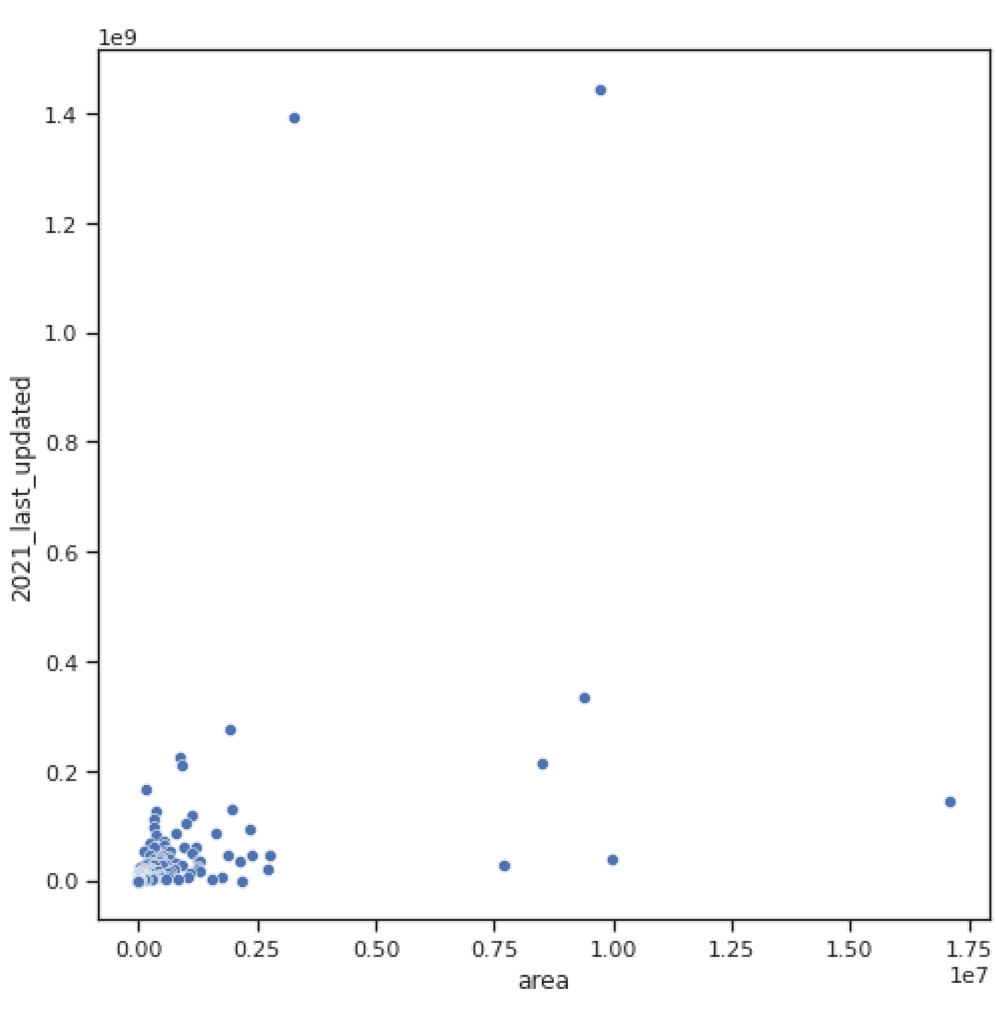
Out[144…

	2021_last_updated	2020_population	area	density_sq_km	growth_rate	world_%	rank
count	2.290000e+02	2.290000e+02	2.290000e+02	229.000000	229.000000	229.000000	229.000000
mean	3.435526e+07	3.403434e+07	5.937714e+05	454.524017	1.067336	0.436507	115.502183
std	1.376530e+08	1.367965e+08	1.779043e+06	2091.885175	1.057531	1.748844	66.947651
min	8.000000e+02	8.010000e+02	1.000000e+00	0.000000	-1.290000	0.000000	1.000000
25%	5.614110e+05	5.559870e+05	4.167000e+03	36.000000	0.300000	0.010000	58.000000
50%	5.811532e+06	5.792202e+06	8.387100e+04	95.000000	0.930000	0.070000	115.000000
75%	2.149028e+07	2.141325e+07	4.465500e+05	239.000000	1.830000	0.270000	172.000000
max	1.443813e+09	1.439324e+09	1.709824e+07	21946.000000	4.430000	18.340000	232.000000

3. Визуальное исследование набора данных

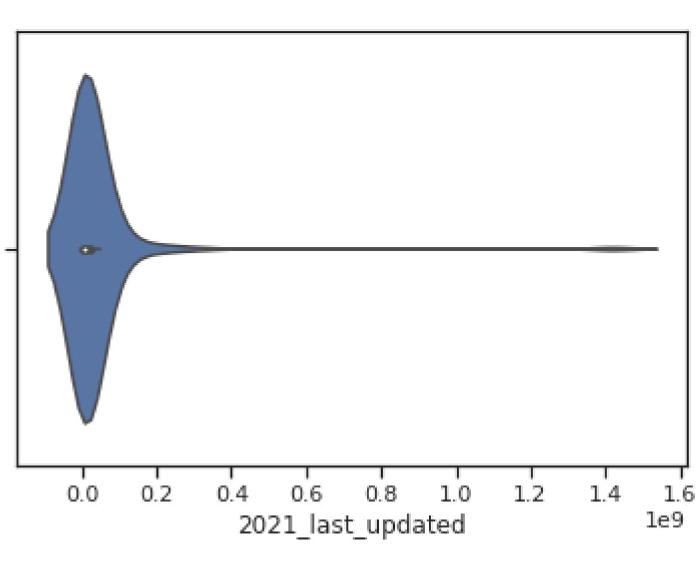
In [145…
```
fig, ax = plt.subplots(figsize=(8,8))
sns.scatterplot(ax=ax, x='area', y='2021_last_updated', data=data)
```

Out[145…
```
<AxesSubplot:xlabel='area', ylabel='2021_last_updated'>
```



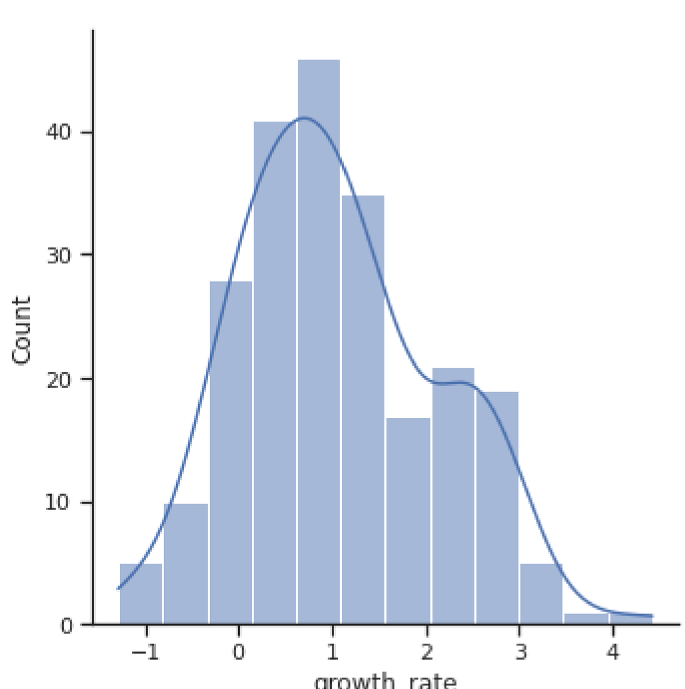
In [146…
```
sns.violinplot(data=data, x="2021_last_updated")
```

Out[146…
```
<AxesSubplot:xlabel='2021_last_updated'>
```



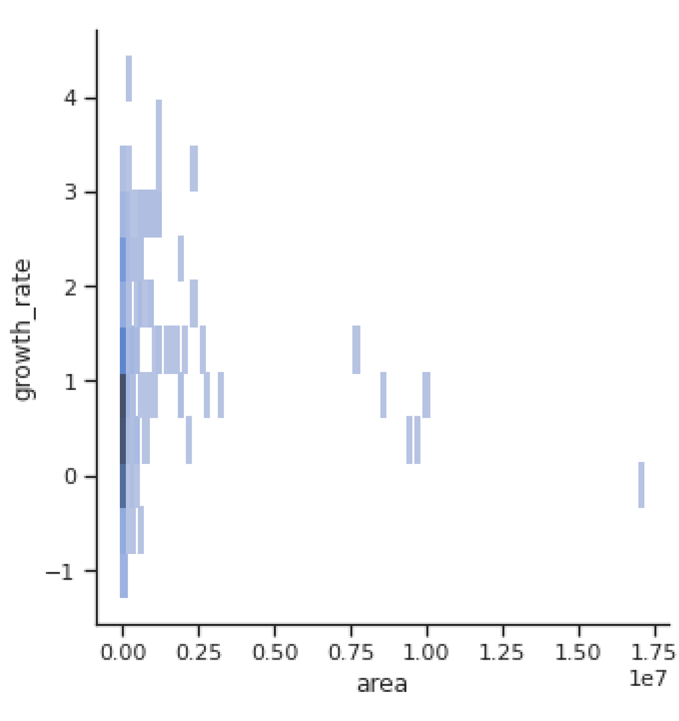
In [147…
```
sns.displot(data=data, x="growth_rate", kde = True)
```

Out[147…
```
<seaborn.axisgrid.FacetGrid at 0x7f28950f8550>
```



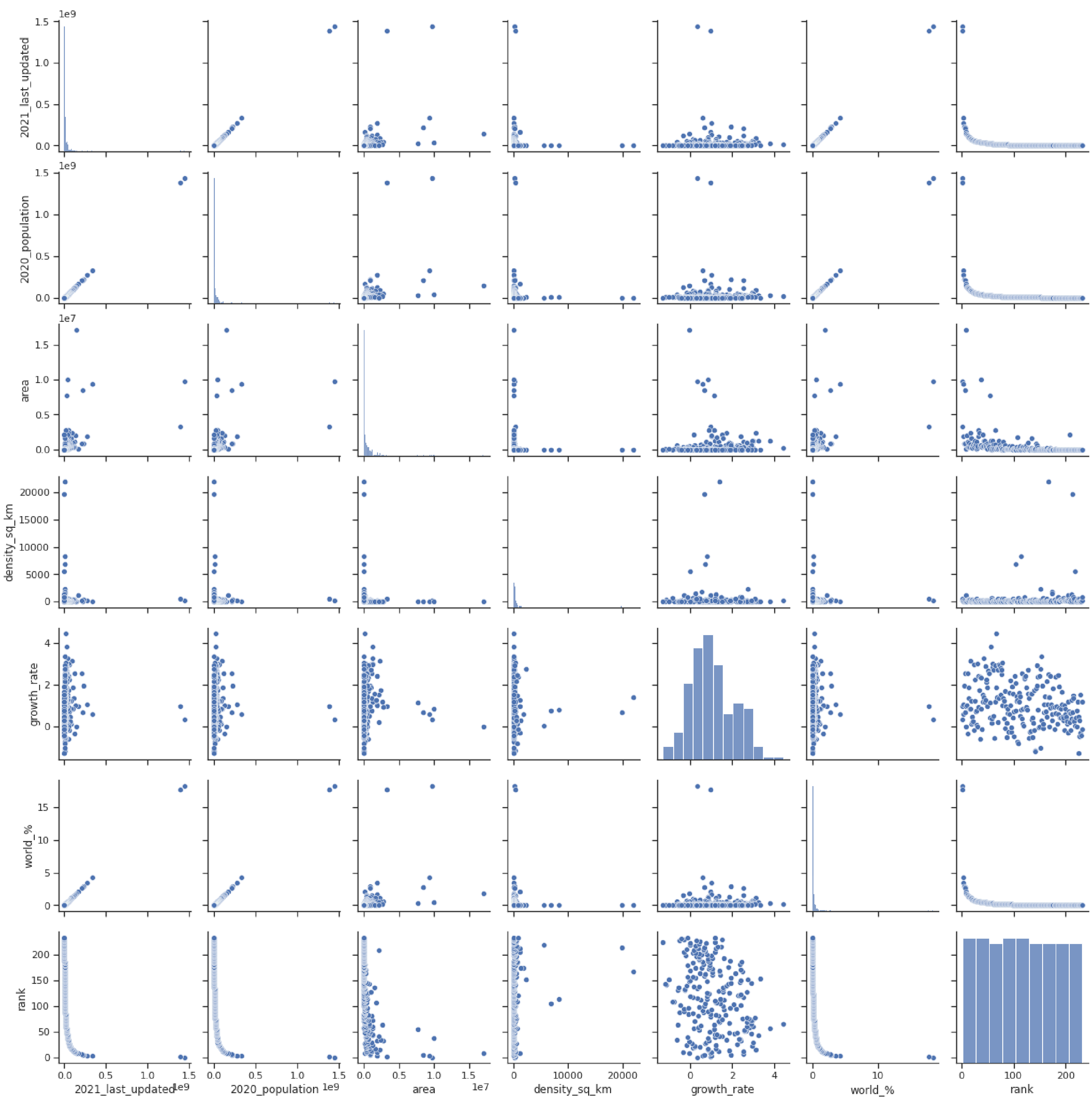
In [148…
```
sns.displot(data=data, x="area", y="growth_rate")
```

Out[148…
```
<seaborn.axisgrid.FacetGrid at 0x7f2895b40d60>
```



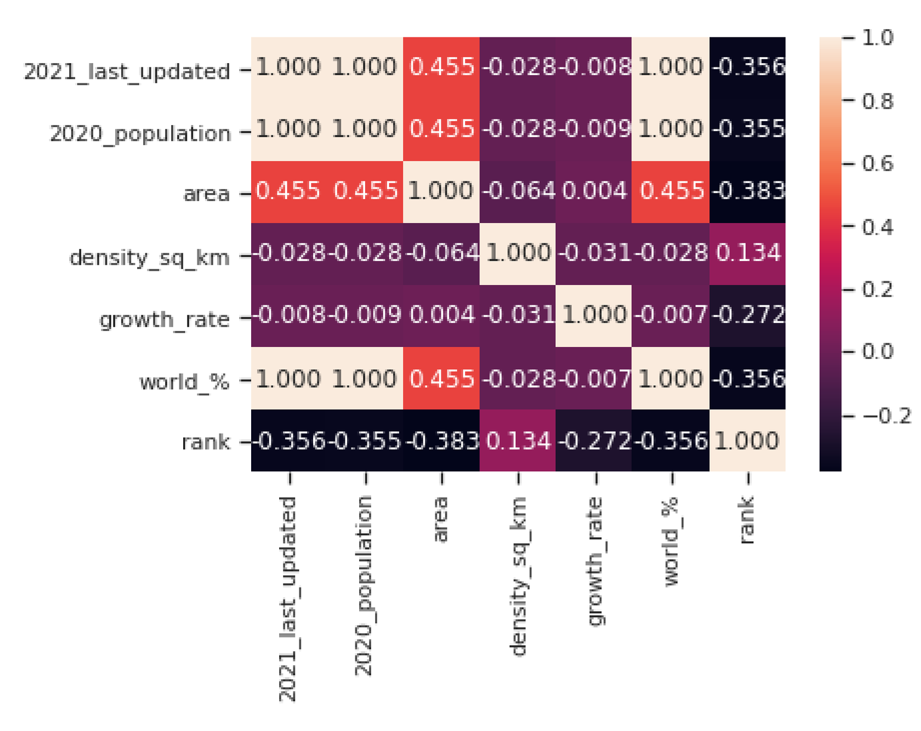
In [149…
```
sns.pairplot(data)
```

Out[149…
```
<seaborn.axisgrid.PairGrid at 0x7f2895637880>
```



In [150…
```
sns.heatmap(data.corr(), annot = True, fmt = '.3f')
```

Out[150…
```
<AxesSubplot:>
```



Судя по карте корреляции, есть смысл работать только с площадью и популяцией (в разных видах)

In [ ]: