Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Лабораторная работа №1
по дисциплине
«Методы машинного обучения»
на тему

«Создание "истории о данных" (Data Storytelling).»

Выполнил:
студент группы ИУ5-23М
Дин Но

Москва — 2021 г.

1. **Цель лабораторной работы**

   изучение различных методов визуализация данных и создание истории на основе данных.

2. Задание

Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:

1.История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.

2.На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.

3.Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.

4.Выбор графиков должен быть обоснован использованием методологии data-to-viz. Рекомендуется учитывать типичные ошибки построения выбранного вида графика по методологии data-to-viz. Если методология Вами отвергается, то просьба обосновать Ваше решение по выбору графика.

5.История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.

3. Ход выполнения работы

3.1. Текстовое описание набора данных

Последний год во всем мире ознаменовался пандемией вируса SARS-CoV-2, вызвавшего болезнь COVID-19. Это

привело к гибели миллионов людей, десятки миллионов заболели, сотни миллионов были помещены в карантин, и миллиарды людей изменили свою жизнь. Пандемия, хотя и продолжает развиваться, и новые люди все еще болеют в мире, есть свет в этом темном туннеле - вакцина. Хотя вирус новый, через год, благодаря работе ученых со всего мира, у нас есть несколько вакцин, которые являются безопасными и имеют высокую эффективность (против вирусов и / или с тяжелым течением болезни). Так человечество сталкивается с серьезным испытанием вакцинировать как можно больше людей, чтобы получить вируса и его последствий избавиться раз и навсегда (вакцина работает только в течение нескольких лет, поэтому важно, чтобы процесс массовой вакцинации проходит гладко).

3.2. Основные характеристики набора данных

Посмотрим на данные в данном наборе данных:

```
In [12]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         from scipy import stats
         import seaborn as sns

         df = pd.read_csv('covid19.csv')
         df.head()
```

Out[12]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hun
0	Albania	ALB	2021-01-10	0.0	0.0	NaN	NaN	NaN	
1	Albania	ALB	2021-01-11	NaN	NaN	NaN	NaN	64.0	
2	Albania	ALB	2021-01-12	128.0	128.0	NaN	NaN	64.0	
3	Albania	ALB	2021-01-13	188.0	188.0	NaN	60.0	63.0	
4	Albania	ALB	2021-01-14	266.0	266.0	NaN	78.0	66.0	

Проверим основные статистические характеристики набора данных:

```
df.describe()
```

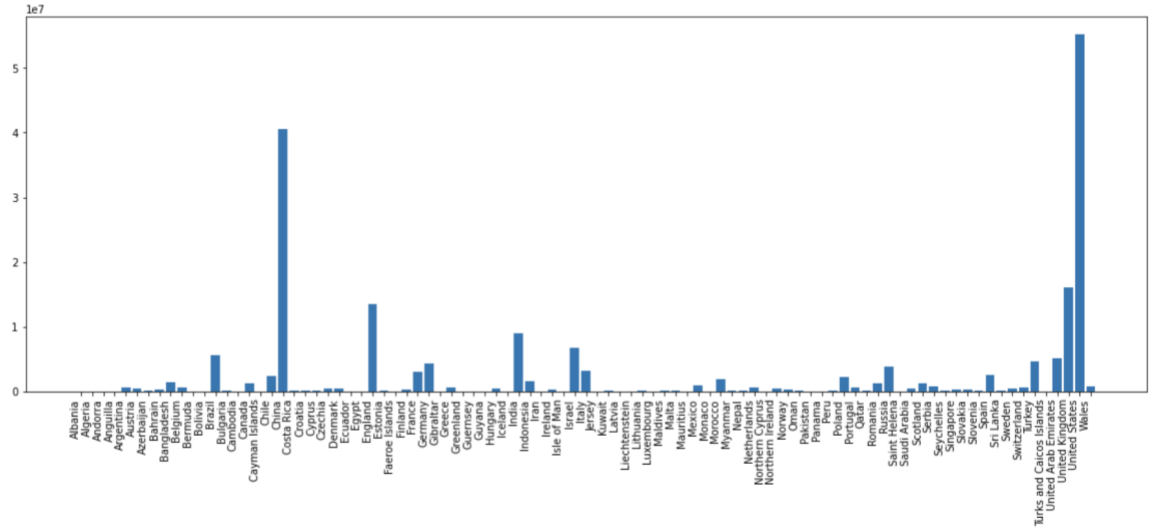
Out[49]:

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinate
count	2.145000e+03	1.774000e+03	1.137000e+03	1.784000e+03	3.186000e+03	2145.000000	
mean	1.393672e+06	1.168188e+06	3.434739e+05	7.205658e+04	5.656643e+04	5.619720	
std	4.577173e+06	3.801181e+06	1.248999e+06	2.033630e+05	1.760802e+05	10.661725	
min	0.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	1.000000e+00	0.000000	
25%	2.809500e+04	2.561775e+04	7.607000e+03	1.956500e+03	1.287500e+03	0.530000	
50%	1.729120e+05	1.563050e+05	2.918800e+04	1.067850e+04	6.257000e+03	2.130000	
75%	7.135170e+05	6.106792e+05	1.712700e+05	5.478875e+04	2.868750e+04	4.950000	
max	5.522036e+07	3.967055e+07	1.501543e+07	2.242472e+06	1.916190e+06	78.300000	

3.3. Визуальное исследование датасета

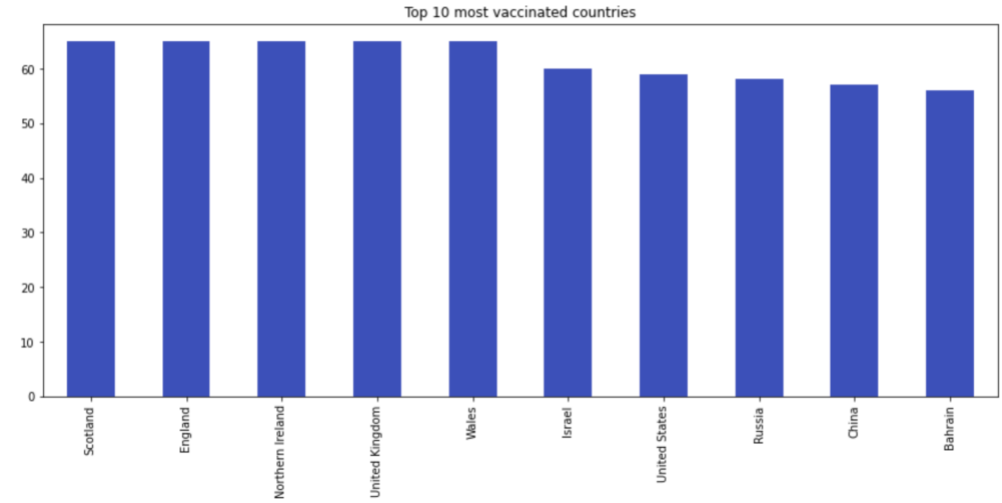
Давайте оценим распределение целевого атрибута - Рейтинг:

```
In [45]: gb_country = df.groupby('country')[['total_vaccinations']].max()

         fig = plt.figure(figsize=(15, 5))
         ax = fig.add_axes([0,0,1,1])

         ax.bar(gb_country.index, gb_country['total_vaccinations'])
         fig.autofmt_xdate(rotation=90)
         plt.show()
```



Эта диаграмма иллюстрирует до сих пор общее число больных в каждой стране.

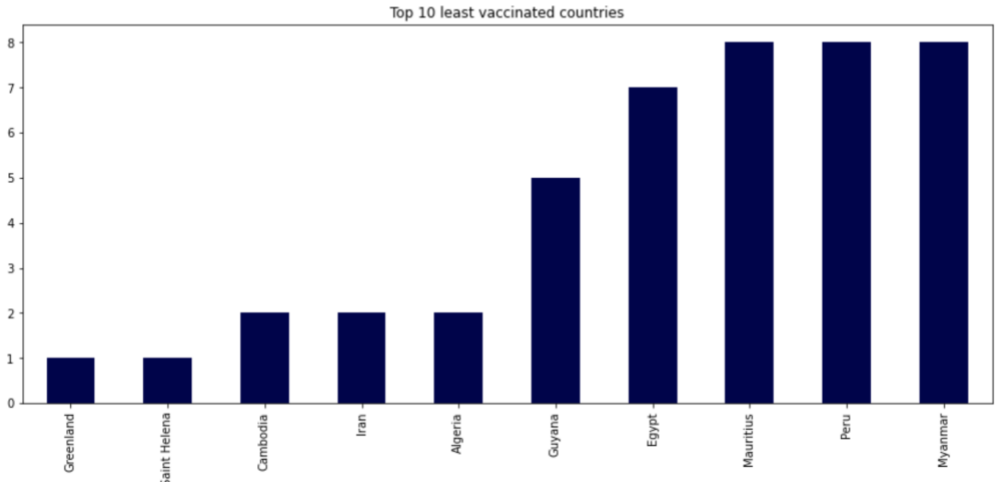
```
In [42]: df['country'].value_counts().sort_values(ascending=False).head(10).plot(kind='bar',figsize=(15,6),colormap='coolwarm')
         plt.title('Top 10 most vaccinated countries')

Out[42]: Text(0.5, 1.0, 'Top 10 most vaccinated countries')
```



Эта диаграмма иллюстрирует 10 стран с наибольшим количеством больных людей на сегодняшний день.

```
In [15]: df['country'].value_counts().sort_values(ascending=True).head(10).plot(kind='bar',figsize=(15,6),colormap='seismic')
         plt.title('Top 10 least vaccinated countries')
```

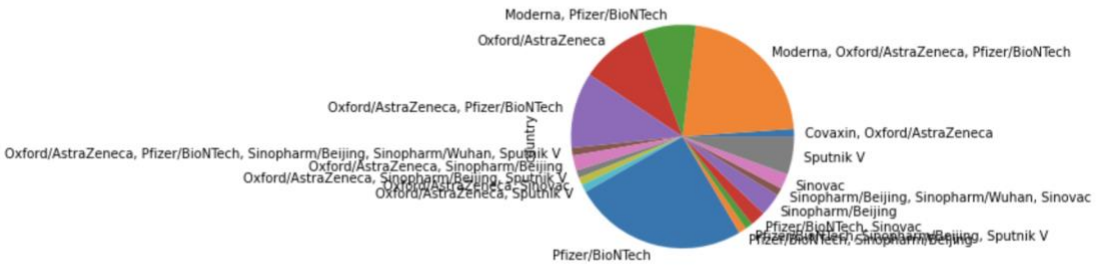
```
Out[15]: Text(0.5, 1.0, 'Top 10 least vaccinated countries')
```



Эта диаграмма иллюстрирует 10 стран с самым низким числом больных людей на сегодняшний день.

```
In [22]: df_vaccines = df.drop_duplicates(subset=['country','vaccines'], keep ='last')
         df_vaccines.groupby('vaccines')['country'].count().plot(kind='pie')
```

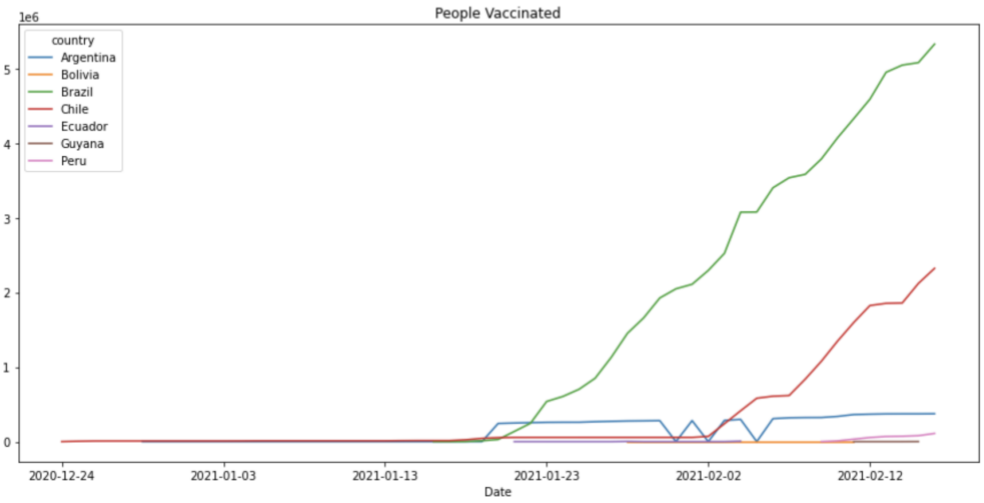
```
Out[22]: <AxesSubplot:ylabel='country'>
```



Эта диаграмма иллюстрирует сравнение числа больных в каждой стране с общим числом больных в мире.

```
In [25]: countries_sa = ['Brazil','Argentina','Chile','Venezuela','Bolivia','Colombia','Ecuador','Peru','Paraguay','Uruguay','Su
         fig,ax = plt.subplots(figsize=(15,7))
         south_america_ds = df[df.country.isin(countries_sa)]
         south_america_ds.groupby(['date','country']).sum()['people_vaccinated'].unstack().plot(ax=ax)
         plt.title('People Vaccinated')
         plt.xlabel('Date')
```

```
Out[25]: Text(0.5, 0, 'Date')
```



Эта диаграмма иллюстрирует количество новых случаев заболевания в день в семи странах.

3.4. Информация о корреляции признаков

Построим корреляционную матрицу по всему набору данных:

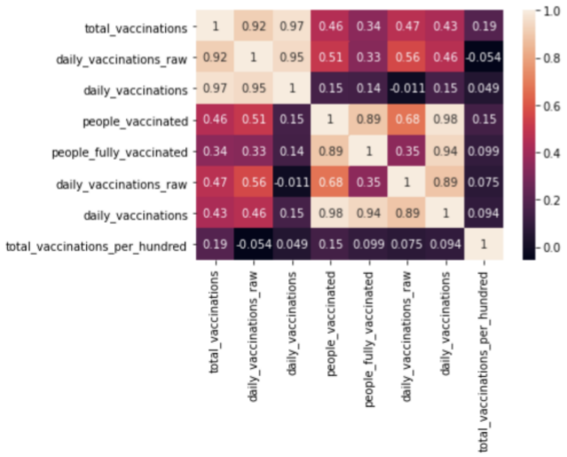
```
In [50]: df.corr()
```

```
Out[50]:
```

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_
total_vaccinations	1.000000	0.994298	0.925016	0.915930	0.967453	
people_vaccinated	0.994298	1.000000	0.879587	0.926011	0.976648	
people_fully_vaccinated	0.925016	0.879587	1.000000	0.848907	0.874140	
daily_vaccinations_raw	0.915930	0.926011	0.848907	1.000000	0.947794	
daily_vaccinations	0.967453	0.976648	0.874140	0.947794	1.000000	
total_vaccinations_per_hundred	0.198666	0.191345	0.198173	0.121487	0.129673	
people_vaccinated_per_hundred	0.205415	0.209374	0.172479	0.137509	0.153187	
people_fully_vaccinated_per_hundred	0.105679	0.073365	0.219562	0.039982	0.052051	
daily_vaccinations_per_million	0.133351	0.134202	0.138819	0.111706	0.081196	

Визуализируем корреляционную матрицу с помощью тепловой карты:

```
In [32]: import seaborn as sns

undt=pd.read_csv('covid19.csv')
new_df = df.sort_values('people_vaccinated_per_hundred', ascending=False).drop_duplicates('country').sort_index().reset
lists_countries_vaccine = list(new_df['country'])
dfun_new = undt[undt['country'].isin(lists_countries_vaccine)]
dfun_new.reset_index(drop=True, inplace=True)
new_df.reset_index(drop=True, inplace=True)
df3 = pd.concat([dfun_new.iloc[:,3],dfun_new.iloc[:,6:8],dfun_new.iloc[:,31:33],new_df.iloc[:,4:9]], axis = 1)
for x in range(4):
    df3.iloc[:,x] = df3.iloc[:,x].astype(float)
sns.heatmap(df3.corr(), annot= True)

Out[32]: <AxesSubplot:>
```



Список литературы

[1] Гапанюк Ю. Е. Лабораторная работа «Разведочный анализ данных. Исследование и визуализация данных» [Электронный ресурс] // GitHub. — 2019. — Режим доступа: https://github.com/ugapanyuk/ml_course/wiki/LAB_EDA_VISUALIZATION (дата обращения: 13.02.2019)

[2] https://www.kaggle.com/datasets