Министерство образования и науки 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**

Отчет по лабораторной работе № 1
«Разведочный анализ данных. Исследование и визуализация данных»
по курсу “Технологии машинного обучения”

Исполнитель:
Студент группы ИУ5-63
Желанкина А.С.
_____ 09.02.2018

Москва, 2019

Задание лабораторной работы

Выбрать набор данных (датасет). Для лабораторных работ не рекомендуется выбирать датасеты большого размера.

Создать ноутбук, который содержит следующие разделы:

- текстовое описание выбранного Вами набора данных,
- основные характеристики датасета,
- визуальное исследование датасета,
- информация о корреляции признаков.

Сформировать отчет и разместить его в своем репозитории на github.

Экранные формы с текстом программы и примерами её выполнения

In [31]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline

sns.set(style="ticks")
```

WHO Suicide Statistics

Основные исторические (1979-2016) данные по странам, годам и демографическим группам.

Основные совокупные числа. Количество самоубийств и численность населения измеряются в людях, поэтому 1 означает просто одного человека, а не тысячу.

In [32]:
```
data = pd.read_csv('C:/Anaconda/who_suicide_statistics.csv', sep=",")
```

In [33]:
```
#Основные характеристики датасета
```

In [34]:
```
#Последние 5 строк датасета
data.tail()
```

Out[34]:

	country	year	sex	age	suicides_no	population
43771	Zimbabwe	1990	male	25-34 years	150.0	NaN
43772	Zimbabwe	1990	male	35-54 years	132.0	NaN
43773	Zimbabwe	1990	male	5-14 years	6.0	NaN
43774	Zimbabwe	1990	male	55-74 years	74.0	NaN
43775	Zimbabwe	1990	male	75+ years	13.0	NaN

In [35]:
```
#Количество строк и столбцов
data.shape
```

Out[35]:
```
(43776, 6)
```

Общая информация о датасете

In [36]:
```
data.describe()
```

Out[36]:

	year	suicides_no	population
count	43776.000000	41520.000000	3.831600e+04
mean	1998.502467	193.315390	1.664091e+06
std	10.338711	800.589926	3.647231e+06
min	1979.000000	0.000000	2.590000e+02
25%	1990.000000	1.000000	8.511275e+04
50%	1999.000000	14.000000	3.806550e+05
75%	2007.000000	91.000000	1.305698e+06
max	2016.000000	22338.000000	4.380521e+07

In [37]:
```
#Колонки с типами данных
data.dtypes
```

Out[37]:
```
country         object
year             int64
sex             object
age             object
suicides_no    float64
population     float64
dtype: object
```

In [38]:
```
#Проверим наличие пустых значений
for col in data.columns:
    #Количество пустых значений - все значения заполнены
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
country - 0
year - 0
sex - 0
age - 0
suicides_no - 2256
population - 5460
```

In [39]:
```
#Заполнили пропуски медианными значениями
data = data.fillna(data.median(axis=0), axis=0)
```

In [40]:
```
#Проверка, что всё заполнили
data.count(axis=0)
```

Out[40]:
```
country        43776
year           43776
sex            43776
age            43776
suicides_no    43776
population     43776
dtype: int64
```

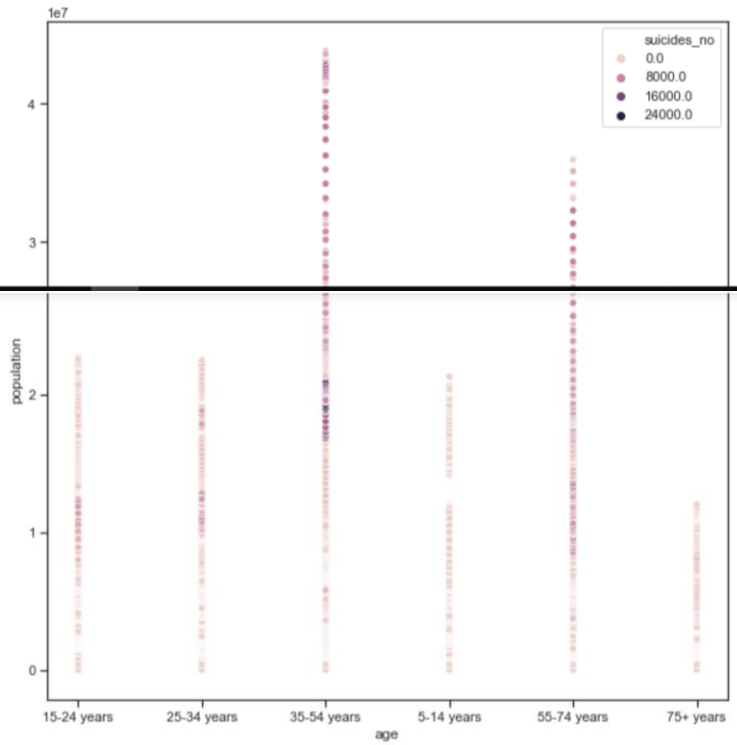
In [41]:
```
#Определим уникальные значения для целевого признака
data['suicides_no'].unique()
```

Out[41]:
```
array([1.4000e+01, 4.0000e+00, 6.0000e+00, ..., 1.1634e+04, 9.0680e+03,
       3.1710e+03])
```

Визуальное исследование датасета

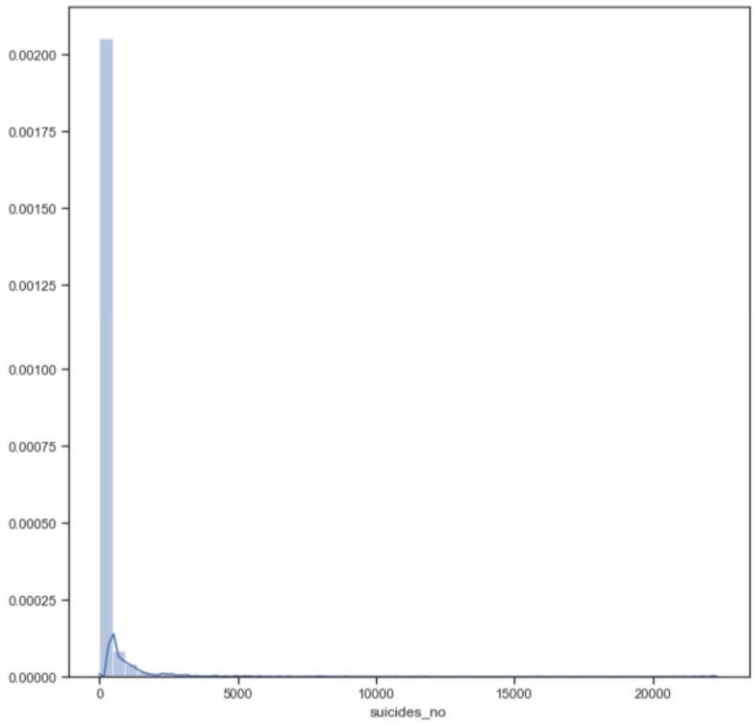
In [42]:
```
#Диаграмма рассеяния
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='age', y='population', data=data, hue='suicides_no')
```

Out[42]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24fe65dc908>
```



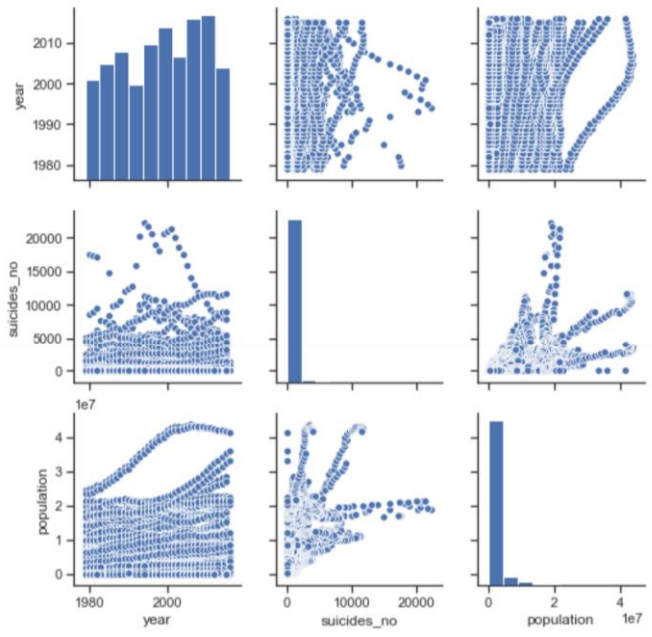
In [43]:
```
#Гистограмма
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['suicides_no'])
```

Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x24fdb4f9630>



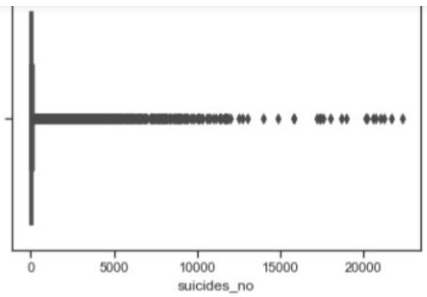
In [44]:
```
#Парные диаграммы
sns.pairplot(data)
```

Out[44]: <seaborn.axisgrid.PairGrid at 0x24fdb54ff60>



In [45]:
```
#одномерное распределение вероятности
sns.boxplot(x=data['suicides_no'])
```

Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x24fe6697978>



In [46]:
```
#Выделение категориальных и числовых признаков
data_new = data
categorical_columns = [c for c in data_new.columns if data_new[c].dtype.name == 'object']
numerical_columns   = [c for c in data_new.columns if data_new[c].dtype.name != 'object']
print(categorical_columns)
print(numerical_columns)
```

```
['country', 'sex', 'age']
['year', 'suicides_no', 'population']
```

In [47]:
```
#Выделение бинарных и небинарных признаков
data_describe = data_new.describe(include=[object])
binary_columns    = [c for c in categorical_columns if data_describe[c]['unique'] == 2]
nonbinary_columns = [c for c in categorical_columns if data_describe[c]['unique'] > 2]
print(binary_columns, nonbinary_columns)
```

```
['sex'] ['country', 'age']
```

In [48]:
```
#Обработка бинарных признаков
data_new.at[data_new['sex'] == 'female', 'sex'] = 0
data_new.at[data_new['sex'] == 'male', 'sex'] = 1
data_new['sex'].describe()
```

Out[48]:
```
count    43776.000000
mean         0.500000
std          0.500006
min          0.000000
25%          0.000000
50%          0.500000
75%          1.000000
max          1.000000
Name: sex, dtype: float64
```

In [49]:
```
#Обрарботка небинарных признаков
data_nonbinary = pd.get_dummies(data_new[nonbinary_columns])
print(data_nonbinary.columns)
```

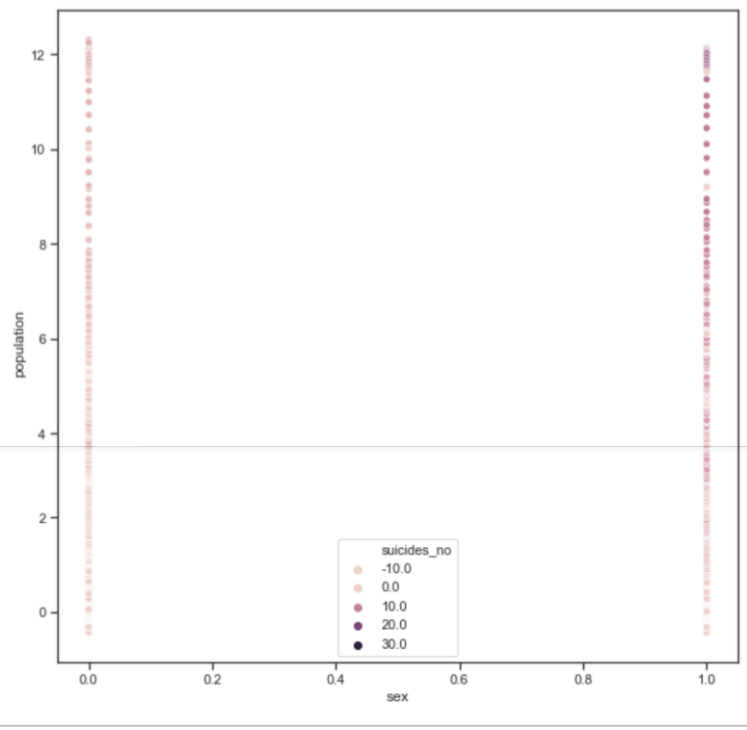
```
Index(['country_Albania', 'country_Anguilla', 'country_Antigua and Barbuda',
       'country_Argentina', 'country_Armenia', 'country_Aruba',
       'country_Australia', 'country_Austria', 'country_Azerbaijan',
       'country_Bahamas',
       ...
       'country_Uzbekistan', 'country_Venezuela (Bolivarian Republic of)',
       'country_Virgin Islands (USA)', 'country_Zimbabwe', 'age_15-24 years',
       'age_25-34 years', 'age_35-54 years', 'age_5-14 years',
       'age_55-74 years', 'age_75+ years'],
      dtype='object', length=147)
```

In [50]:
```
#Нормализация количественных признаков и создание одного нового датасета
data_numerical = data_new[numerical_columns]
data_numerical = (data_numerical - data_numerical.mean()) / data_numerical.std()
data_new = pd.concat((data_numerical, data_new[binary_columns], data_nonbinary), axis=1)
data_new = pd.DataFrame(data_new, dtype=float)
print(data_new.shape)
print(data_new.columns)
```

```
(43776, 151)
Index(['year', 'suicides_no', 'population', 'sex', 'country_Albania',
       'country_Anguilla', 'country_Antigua and Barbuda', 'country_Argentina',
       'country_Armenia', 'country_Aruba',
       ...
       'country_Uzbekistan', 'country_Venezuela (Bolivarian Republic of)',
       'country_Virgin Islands (USA)', 'country_Zimbabwe', 'age_15-24 years',
       'age_25-34 years', 'age_35-54 years', 'age_5-14 years',
       'age_55-74 years', 'age_75+ years'],
      dtype='object', length=151)
```

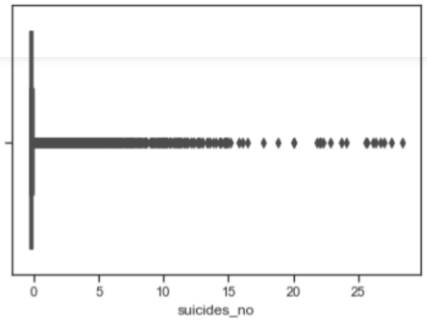
In [51]:
```
#Диаграмма рассеяния
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='sex', y='population', data=data_new, hue='suicides_no')
```

Out[51]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24fe6601320>
```



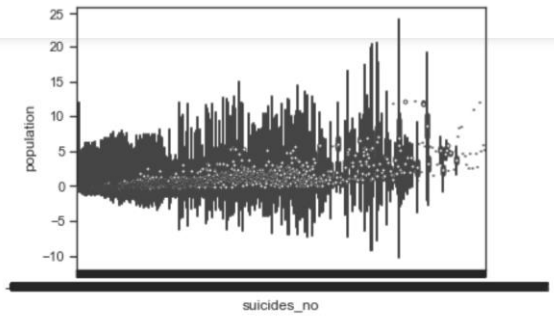
In [52]:
```
#одномерное распределение вероятности
sns.boxplot(x=data_new['suicides_no'])
```

Out[52]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24fe6631048>
```



In [53]:
```
#Violin plot
# Распределение параметра population с группированные по suicides_no
sns.violinplot(x='suicides_no', y='population', data=data_new)
```

Out[53]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24fdadc3438>
```



Информация о корреляции признаков

In [54]:
```
data.corr()
```

Out[54]:

	year	sex	suicides_no	population
year	1.000000	0.000000	-0.000962	0.021357
sex	0.000000	1.000000	0.123537	-0.009960
suicides_no	-0.000962	0.123537	1.000000	0.606450
population	0.021357	-0.009960	0.606450	1.000000

In [55]:
```
data_new.corr()
```

Out[55]:

	year	suicides_no	population	sex	country_Albania	country_Anguilla	country_Antigua and Barbu
year	1.000000e+00	-0.000962	0.021357	4.326714e-18	1.340978e-02	8.786435e-03	3.320948e
suicides_no	-9.621454e-04	1.000000	0.606450	1.235369e-01	-2.071851e-02	-2.110611e-02	-2.073312e
population	2.135677e-02	0.606450	1.000000	-9.959590e-03	-3.419856e-02	-2.924591e-02	-3.831325e

In [56]:
```
data.corr(method='kendall')
```

Out[56]:

	year	sex	suicides_no	population
year	1.000000	0.000000	-0.022183	-0.003620
sex	0.000000	1.000000	0.142904	-0.015135
suicides_no	-0.022183	0.142904	1.000000	0.493486
population	-0.003620	-0.015135	0.493486	1.000000

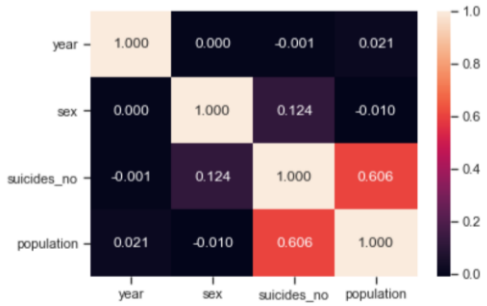
In [57]:
```
data.corr(method='spearman')
```

Out[57]:

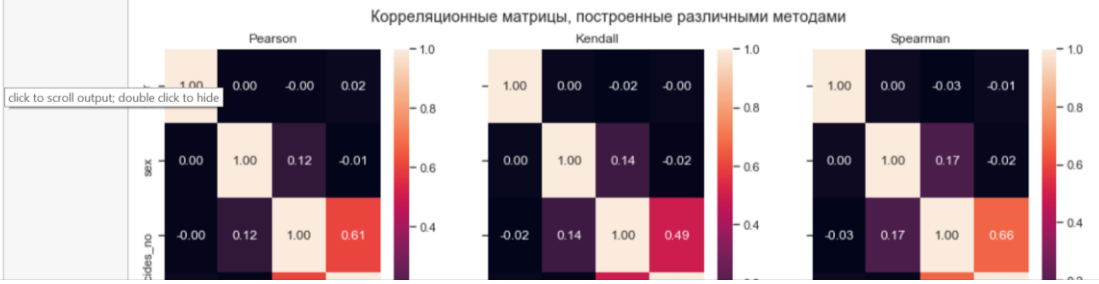
	year	sex	suicides_no	population
year	1.000000	0.000000	-0.033021	-0.005582
sex	0.000000	1.000000	0.170777	-0.018409
suicides_no	-0.033021	0.170777	1.000000	0.662356
population	-0.005582	-0.018409	0.662356	1.000000

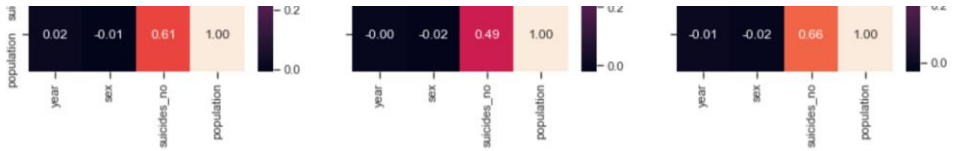
In [58]:
```
#Вывод значений в ячейках в тепловой карте
sns.heatmap(data.corr(), annot=True, fmt='.3f')
```

Out[58]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x24ff0205748>
```



In [60]:
```
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(15,5))
sns.heatmap(data.corr(method='pearson'), ax=ax[0], annot=True, fmt='.2f')
sns.heatmap(data.corr(method='kendall'), ax=ax[1], annot=True, fmt='.2f')
sns.heatmap(data.corr(method='spearman'), ax=ax[2], annot=True, fmt='.2f')
fig.suptitle('Корреляционные матрицы, построенные различными методами')
ax[0].title.set_text('Pearson')
ax[1].title.set_text('Kendall')
ax[2].title.set_text('Spearman')
```




population
0.02
-0.01
0.61
1.00
year
sex
suicides_no
population
0.2
0.0
-0.00
-0.02
0.49
1.00
year
sex
suicides_no
population
0.2
0.0
-0.01
-0.02
0.66
1.00
year
sex
suicides_no
population
0.0