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

Отчет

Лабораторная работа № 1

По курсу «Технологии машинного обучения»

«Разведочный анализ данных. Исследование и визуализация данных»

ИСПОЛНИТЕЛЬ:

Сергеев И.В.
Группа ИУ5-64Б

_____

"__"_____2020 г.

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

_____

"__"_____2020 г.

Москва 2020

Описание задания

Цель лабораторной работы - изучение различных методов визуализация данных.

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

Для выполнения данного задания я взял набор данных (датасет), отражающий экономические показатели для большинства стран мира.

Датасет содержит следующие колонки:

- Year – год актуализации
- ISO_code – международный код страны
- Countries – название страны
- ECONOMIC FREEDOM – показатель экономической независимости страны
- Rank – позиция страны в мировом экономическом рейтинге
- И также много различных точечных экономических показателей

Текст программы

Программа разрабатывалась в IDE PyCharm. Ниже приведён полный листинг программы:

```
#%%

import numpy
import pandas
import seaborn
import matplotlib.pyplot as plt
%matplotlib inline
seaborn.set(style="ticks")

#%%

# This is our dataset from .csv file
data = pandas.read_csv('./dataset.csv', sep=",")

#%%

data.head()

#%%

data.shape

#%%
```

```
total_count = data.shape[0]

#%%

print('Strings number: {}'.format(total_count))

#%%

data.columns

#%%

data.dtypes

#%%

# Num of empty cells in 'data'
for col in data.columns:
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))

#%%

data2 = data.dropna(axis=0, how='any')
(data.shape, data2.shape)

#%%

#Num of empty cells in 'data2'
for col in data2.columns:
    temp_null_count = data2[data2[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))

#%%

data.describe()

#%%

data['countries'].unique()

#%%

fig, ax = plt.subplots(figsize=(10,10))
seaborn.scatterplot(ax=ax, x='rank', y='ECONOMIC FREEDOM', data=data2)

#%%

fig, ax = plt.subplots(figsize=(10,10))
seaborn.scatterplot(ax=ax, x='3_sound_money', y='3c_inflation', data=data2,
hue='rank')

#%%

fig, ax = plt.subplots(figsize=(10,10))
seaborn.distplot(data2[('ECONOMIC FREEDOM')])

#%%

fig, ax = plt.subplots(figsize=(10,10))
seaborn.distplot(data2[('1d_top_marg_tax_rate')])
```

```
#%%

seaborn.jointplot(x='3_sound_money', y='5_regulation', data=data2)

#%%

seaborn.jointplot(x='3_sound_money', y='5c_business_reg', data=data2)

#%%

seaborn.jointplot(x='rank', y='ECONOMIC FREEDOM', data=data2, kind="hex")

#%%

seaborn.jointplot(x='1_size_government', y='ECONOMIC FREEDOM', data=data2,
kind="kde")

#%%

seaborn.pairplot(data2)

#%%

seaborn.pairplot(data2, vars=['ECONOMIC
FREEDOM','1_size_government','2a_judicial_independence','2d_military_interference'

,'2_property_rights','3c_inflation','3_sound_money','5_regulation'])

#%%

seaborn.pairplot(data2, vars=['ECONOMIC
FREEDOM','1_size_government','2a_judicial_independence','2d_military_interference'

,'2_property_rights','3c_inflation','3_sound_money','5_regulation'], hue="rank")

#%%

seaborn.boxplot(x=data['2_property_rights'])

#%%

# Vertical scrolling
seaborn.boxplot(y=data['2_property_rights'])

#%%

seaborn.violinplot(x=data2['3_sound_money'])

#%%

fig, ax = plt.subplots(2, 1, figsize=(10,10))
seaborn.violinplot(ax=ax[0], x=data2['3c_inflation'])
seaborn.distplot(data2['3c_inflation'], ax=ax[1])

#%%

seaborn.violinplot(x='year', y='3c_inflation', data=data2)

#%%

data2.corr()
```

```
#%%

seaborn.heatmap(data2.corr())

#%%

d2 = data2[['year','ECONOMIC
FREEDOM','1_size_government','2a_judicial_independence','2j_gender_adjustment',

'2_property_rights','3c_inflation','3_sound_money','4_trade','5_regulation']]
seaborn.heatmap(d2.corr())

#%%

seaborn.heatmap(d2.corr(), cmap='YlGnBu', annot=True, fmt='.3f')

#%%

# Triangle matrix
mask = numpy.zeros_like(d2.corr(), dtype=numpy.bool)
# mask[np.triu_indices_from(mask)] = True
mask[numpy.tril_indices_from(mask)] = True
seaborn.heatmap(d2.corr(), mask=mask, annot=True, fmt='.3f')

#%%

fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(15,5))
seaborn.heatmap(d2.corr(method='pearson'), ax=ax[0], annot=True, fmt='.2f')
seaborn.heatmap(d2.corr(method='kendall'), ax=ax[1], annot=True, fmt='.2f')
seaborn.heatmap(d2.corr(method='spearman'), ax=ax[2], annot=True, fmt='.2f')
fig.suptitle('Correlation matrix by some methods:')
ax[0].title.set_text('Pearson')
ax[1].title.set_text('Kendall')
ax[2].title.set_text('Spearman')
```

Примеры выполнения программы

Выполнение программы, а также наглядная демонстрация входных и выходных данных (таблиц, графиков и тд) осуществлялась на базе Jupyter Notebook, сервер которого запускался из-под PyCharm. Ниже приведены скриншоты, отражающие работу программы:

jupyter lab1 (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

In [1]:
```
import numpy
import pandas
import seaborn
import matplotlib.pyplot as plt
%matplotlib inline
seaborn.set(style="ticks")
```

In [2]:
```
# This is our dataset from .csv file
data = pandas.read_csv('./dataset.csv', sep=",")
```

In [3]:
```
data.head()
```

Out[3]:

	year	ISO_code	countries	ECONOMIC FREEDOM	rank	quartile	1a_government_consumption	1b_transfers	1c_gov_enterprises	1d_top_marg_tax_rate	...	3_sound_
0	2016	ALB	Albania	7.54	34.0	1.0	8.232353	7.509902	8.0	8.0	...	9.5
1	2016	DZA	Algeria	4.99	159.0	4.0	2.150000	7.817129	0.0	4.5	...	7.2
2	2016	AGO	Angola	5.17	155.0	4.0	7.600000	8.886739	0.0	9.5	...	5.6
3	2016	ARG	Argentina	4.84	160.0	4.0	5.335294	6.048930	6.0	4.0	...	5.6
4	2016	ARM	Armenia	7.57	29.0	1.0	7.264706	7.748532	8.0	5.0	...	9.5

5 rows × 36 columns

In [4]:
```
data.shape
```

Out[4]:
```
(3726, 36)
```

In [5]:
```
total_count = data.shape[0]
```

jupyter lab1 (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

In [6]:
```
print('Strings number: {}'.format(total_count))
```

```
Strings number: 3726
```

In [7]:
```
data.columns
```

Out[7]:
```
Index(['year', 'ISO_code', 'countries', 'ECONOMIC FREEDOM', 'rank', 'quartile',
       '1a_government_consumption', '1b_transfers', '1c_gov_enterprises',
       '1d_top_marg_tax_rate', '1_size_government', '2a_judicial_independence',
       '2b_impartial_courts', '2c_protection_property_rights',
       '2d_military_interference', '2e_integrity_legal_system',
       '2f_legal_enforcement_contracts', '2g_restrictions_sale_real_property',
       '2h_reliability_police', '2i_business_costs_crime',
       '2j_gender_adjustment', '2_property_rights', '3a_money_growth',
       '3b_std_inflation', '3c_inflation', '3d_freedom_own_foreign_currency',
       '3_sound_money', '4a_tariffs', '4b_regulatory_trade_barriers',
       '4c_black_market', '4d_control_movement_capital_ppl', '4_trade',
       '5a_credit_market_reg', '5b_labor_market_reg', '5c_business_reg',
       '5_regulation'],
      dtype='object')
```

In [8]:
```
data.dtypes
```

Out[8]:
```
year                              int64
ISO_code                         object
countries                        object
ECONOMIC FREEDOM                float64
rank                            float64
quartile                        float64
1a_government_consumption       float64
1b_transfers                    float64
1c_gov_enterprises              float64
1d_top_marg_tax_rate            float64
1_size_government               float64
2a_judicial_independence        float64
```

jupyter lab1 (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

In [9]:
```
# Num of empty cells in 'data'
for col in data.columns:
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
year - 0
ISO_code - 0
countries - 0
ECONOMIC FREEDOM - 723
rank - 723
quartile - 723
1a_government_consumption - 589
1b_transfers - 960
1c_gov_enterprises - 646
1d_top_marg_tax_rate - 1047
1_size_government - 647
2a_judicial_independence - 1553
2b_impartial_courts - 1179
2c_protection_property_rights - 1129
2d_military_interference - 1186
2e_integrity_legal_system - 1153
2f_legal_enforcement_contracts - 1361
2g_restrictions_sale_real_property - 1384
2h_reliability_police - 2071
2i_business_costs_crime - 2071
2j_gender_adjustment - 66
2_property_rights - 755
3a_money_growth - 644
3b_std_inflation - 607
3c_inflation - 607
3d_freedom_own_foreign_currency - 621
3_sound_money - 611
4a_tariffs - 681
4b_regulatory_trade_barriers - 1348
4c_black_market - 617
```

jupyter lab1 (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

In [10]:
```
data2 = data.dropna(axis=0, how='any')
(data.shape, data2.shape)
```

Out[10]: ((3726, 36), (1316, 36))

In [11]:
```
#Num of empty cells in 'data2'
for col in data2.columns:
    temp_null_count = data2[data2[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
year - 0
ISO_code - 0
countries - 0
ECONOMIC FREEDOM - 0
rank - 0
quartile - 0
1a_government_consumption - 0
1b_transfers - 0
1c_gov_enterprises - 0
1d_top_marg_tax_rate - 0
1_size_government - 0
2a_judicial_independence - 0
2b_impartial_courts - 0
2c_protection_property_rights - 0
2d_military_interference - 0
2e_integrity_legal_system - 0
2f_legal_enforcement_contracts - 0
2g_restrictions_sale_real_property - 0
2h_reliability_police - 0
2i_business_costs_crime - 0
2j_gender_adjustment - 0
2_property_rights - 0
3a_money_growth - 0
3b_std_inflation - 0
3c_inflation - 0
```

jupyter lab1 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

In [12]:
```
data.describe()
```

Out[12]:

	year	ECONOMIC FREEDOM	rank	quartile	1a_government_consumption	1b_transfers	1c_gov_enterprises	1d_top_marg_tax_rate	1_size_gover
count	3726.000000	3003.000000	3003.000000	3003.000000	3137.000000	2766.000000	3080.000000	2679.000000	3079.(
mean	2001.347826	6.519640	68.307026	2.497835	5.862426	7.672901	5.737987	5.813177	6.2
std	12.735125	1.133638	41.343417	1.118963	2.270241	2.138957	3.242377	2.654083	1.4
min	1970.000000	1.970000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.6
25%	1995.000000	5.855000	33.000000	1.000000	4.450000	6.207809	4.000000	4.000000	5.2
50%	2005.000000	6.680000	66.000000	3.000000	6.082353	8.432251	7.000000	6.000000	6.3
75%	2011.000000	7.350000	102.000000	3.000000	7.571360	9.482289	8.000000	8.000000	7.2
max	2016.000000	9.190000	162.000000	4.000000	10.000000	10.000000	10.000000	10.000000	9.9

8 rows × 34 columns

In [13]:
```
data['countries'].unique()
```

Out[13]:
```
array(['Albania', 'Algeria', 'Angola', 'Argentina', 'Armenia',
       'Australia', 'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain',
       'Bangladesh', 'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin',
       'Bhutan', 'Bolivia', 'Bosnia and Herzegovina', 'Botswana',
       'Brazil', 'Brunei Darussalam', 'Bulgaria', 'Burkina Faso',
       'Burundi', 'Cambodia', 'Cameroon', 'Canada', 'Cape Verde',
       'Central Afr. Rep.', 'Chad', 'Chile', 'China', 'Colombia',
       'Congo, Dem. R.', 'Congo, Rep. Of', 'Costa Rica', "Cote d'Ivoire",
       'Croatia', 'Cyprus', 'Czech Rep.', 'Denmark', 'Dominican Rep.',
       'Ecuador', 'Egypt', 'El Salvador', 'Estonia', 'Ethiopia', 'Fiji',
       'Finland', 'France', 'Gabon', 'Gambia, The', 'Georgia', 'Germany',
       'Ghana', 'Greece', 'Guatemala', 'Guinea', 'Guinea-Bissau',
       'Guyana', 'Haiti', 'Honduras', 'Hong Kong', 'Hungary', 'Iceland',
```

jupyter lab1 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

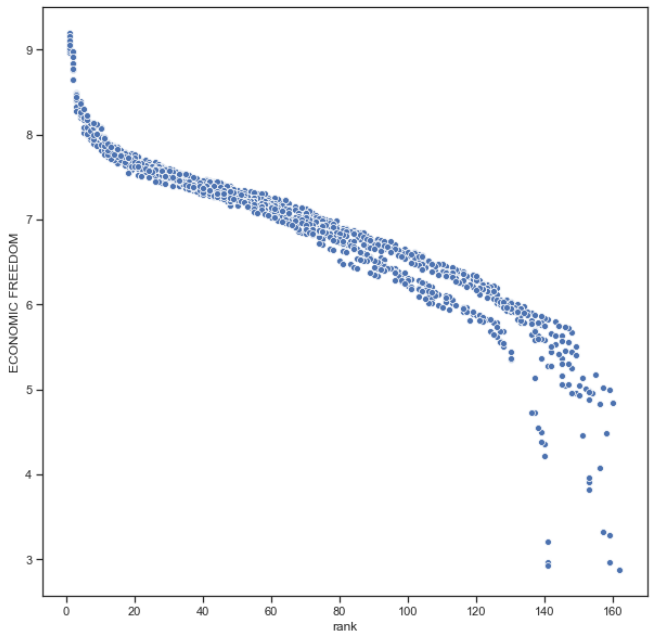
Not Trusted | Python 3

Run Code

```
       'Zimbabwe'], dtype=object)
```

In [14]:
```
fig, ax = plt.subplots(figsize=(10,10))
seaborn.scatterplot(ax=ax, x='rank', y='ECONOMIC FREEDOM', data=data2)
```

Out[14]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xd8065b0>
```



In [15]:
```
fig, ax = plt.subplots(figsize=(10,10))
seaborn.scatterplot(ax=ax, x='3_sound_money', y='3c_inflation', data=data2, hue='rank')
```

jupyter lab1 (autosaved)

Logout

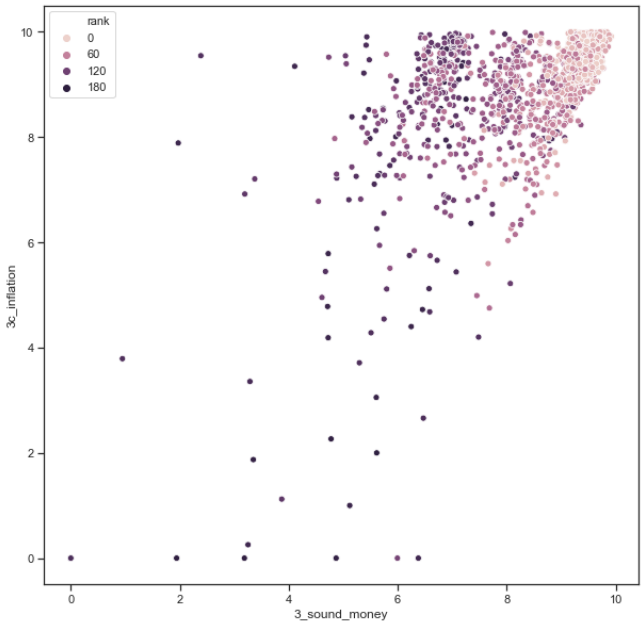
File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

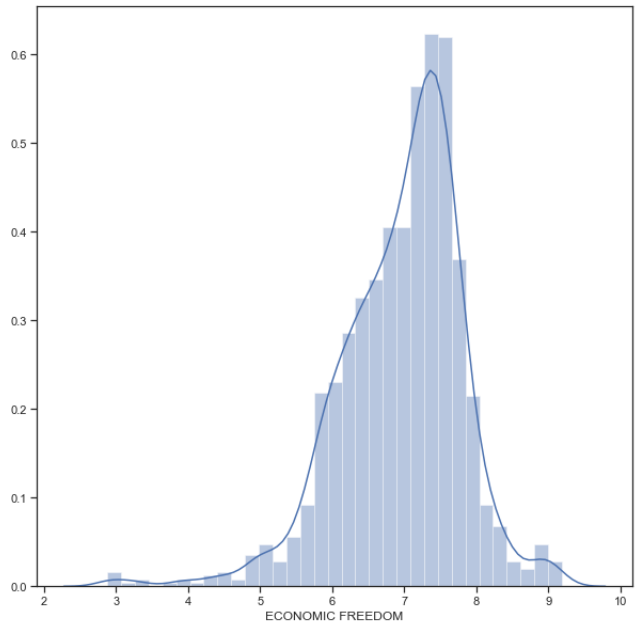
In [15]:
```
fig, ax = plt.subplots(figsize=(10,10))
seaborn.scatterplot(ax=ax, x='3_sound_money', y='3c_inflation', data=data2, hue='rank')
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0xd873c50>



In [16]:
```
fig, ax = plt.subplots(figsize=(10,10))
seaborn.distplot(data2[('ECONOMIC FREEDOM')])
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0xd81ed90>



In [17]:
```
fig, ax = plt.subplots(figsize=(10,10))
seaborn.distplot(data2[('1d_top_marg_tax_rate')])
```

jupyter lab1 (autosaved)

Logout

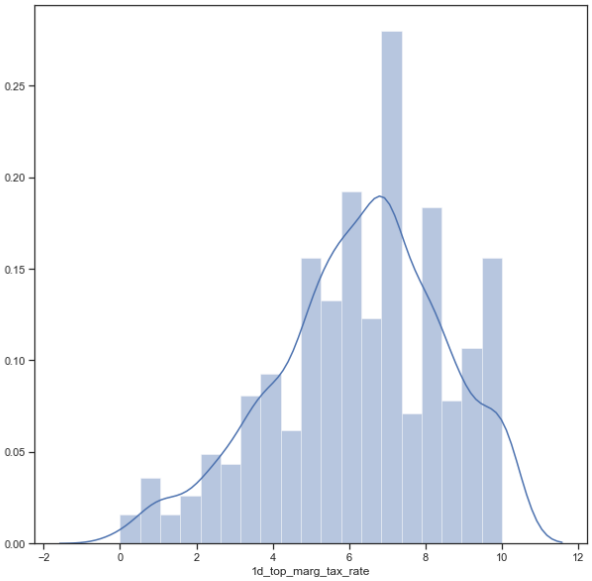
File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

In [17]:
```
fig, ax = plt.subplots(figsize=(10,10))
seaborn.distplot(data2[('1d_top_marg_tax_rate')])
```

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0xf594530>



In [18]:
```
seaborn.jointplot(x='3_sound_money', y='5_regulation', data=data2)
```

Out[18]: <seaborn.axisgrid.JointGrid at 0xf3b17d0>

jupyter lab1 (autosaved)

Logout

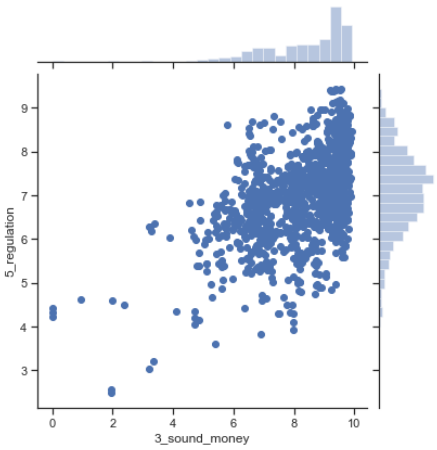
File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

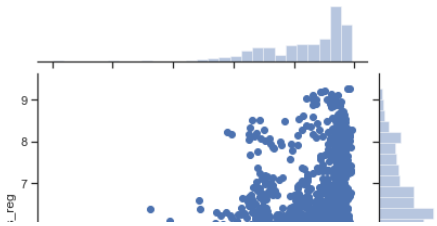
In [18]:
```
seaborn.jointplot(x='3_sound_money', y='5_regulation', data=data2)
```

Out[18]: <seaborn.axisgrid.JointGrid at 0xf3b17d0>



In [19]:
```
seaborn.jointplot(x='3_sound_money', y='5c_business_reg', data=data2)
```

Out[19]: <seaborn.axisgrid.JointGrid at 0xd93a1b0>



jupyter lab1 (unsaved changes)

Logout

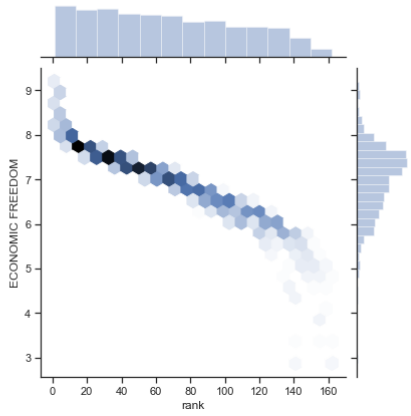
File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run Code

In [20]:
```
seaborn.jointplot(x='rank', y='ECONOMIC FREEDOM', data=data2, kind="hex")
```

Out[20]: <seaborn.axisgrid.JointGrid at 0xf911eb0>



In [21]:
```
seaborn.jointplot(x='1_size_government', y='ECONOMIC FREEDOM', data=data2, kind="kde")
```

Out[21]: <seaborn.axisgrid.JointGrid at 0xfb94410>