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 - и**зучение различных методов визуализация данных.

**Краткое описание - п**остроение 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:





















