**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчет по лабораторной работе №1

Выполнил:  
Студент группы ИУ5-62Б   
Андреев Илья

Подпись и дата:

Проверил:  
преподаватель каф. ИУ5   
Гапанюк Ю.Е.

Подпись и дата:

г. Москва, 2020 г.

**Лабораторная работа №1**

[**Разведочный анализ данных.**](https://en.wikipedia.org/wiki/Exploratory_data_analysis)**Исследование и визуализация данных.**

**Цель лабораторной работы:** изучение различных методов визуализация данных.

### Задание:

* Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов [здесь.](https://github.com/ugapanyuk/ml_course_2020/wiki/DSLIST)
* Для первой лабораторной работы рекомендуется использовать датасет без пропусков в данных, например из [Scikit-learn.](https://scikit-learn.org/stable/datasets/index.html" \l "toy-datasets) Пример преобразования датасетов Scikit-learn в Pandas Dataframe можно посмотреть [здесь.](https://github.com/ugapanyuk/ml_course_2020/blob/master/ds/sklearn_datasets.ipynb)
* 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.

## 1) Текстовое описание набора данных

В качестве набора данных использхуется набор данных со статистикой уровня счастья населения в различных странах.

Датасет содержит следующие колонки:

* Overall rank - место страны по уровню счастья населения
* Country or region - страна или регион
* Score - уровень счастья населения
* GPD per capita - ВВП на душу населения
* Social support - социальная поддержка
* Healthy life expectancy - оценка ожидаемой продолжительности жизни
* Freedom to make life choices - свобода выбора
* Generosity - щедрость
* Perceptions of corruption - ощущение коррупции

### Импорт библиотек

In [3]:

**import** **numpy** **as** **np**

**import** **pandas** **as** **pd**

**import** **seaborn** **as** **sns**

**import** **matplotlib.pyplot** **as** **plt**

**from** **pandas.plotting** **import** scatter\_matrix

**import** **warnings**

warnings.filterwarnings('ignore')

sns.set(style="ticks")

%**matplotlib** inline

### Загрузка данных

In [4]:

happy\_data = pd.read\_csv('2018.csv', sep = ',' )

## 2)Основные характеристики датасета

In [5]:

*# Первые пять строк датасета*

happy\_data.head()

Out[5]:

|  | **Overall rank** | **Country or region** | **Score** | **GDP per capita** | **Social support** | **Healthy life expectancy** | **Freedom to make life choices** | **Generosity** | **Perceptions of corruption** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 1 | Finland | 7.632 | 1.305 | 1.592 | 0.874 | 0.681 | 0.202 | 0.393 |
| **1** | 2 | Norway | 7.594 | 1.456 | 1.582 | 0.861 | 0.686 | 0.286 | 0.340 |
| **2** | 3 | Denmark | 7.555 | 1.351 | 1.590 | 0.868 | 0.683 | 0.284 | 0.408 |
| **3** | 4 | Iceland | 7.495 | 1.343 | 1.644 | 0.914 | 0.677 | 0.353 | 0.138 |
| **4** | 5 | Switzerland | 7.487 | 1.420 | 1.549 | 0.927 | 0.660 | 0.256 | 0.357 |

In [6]:

*# Размер датасета*

happy\_data.shape

Out[6]:

(156, 9)

In [7]:

*# Количество нулевых элементов*

happy\_data.isnull().sum()

Out[7]:

Overall rank 0

Country or region 0

Score 0

GDP per capita 0

Social support 0

Healthy life expectancy 0

Freedom to make life choices 0

Generosity 0

Perceptions of corruption 1

dtype: int64

In [8]:

*# Колонки и их типы данных*

happy\_data.dtypes

Out[8]:

Overall rank int64

Country or region object

Score float64

GDP per capita float64

Social support float64

Healthy life expectancy float64

Freedom to make life choices float64

Generosity float64

Perceptions of corruption float64

dtype: object

In [9]:

*# Описание датасета*

happy\_data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 156 entries, 0 to 155

Data columns (total 9 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Overall rank 156 non-null int64

1 Country or region 156 non-null object

2 Score 156 non-null float64

3 GDP per capita 156 non-null float64

4 Social support 156 non-null float64

5 Healthy life expectancy 156 non-null float64

6 Freedom to make life choices 156 non-null float64

7 Generosity 156 non-null float64

8 Perceptions of corruption 155 non-null float64

dtypes: float64(7), int64(1), object(1)

memory usage: 11.1+ KB

In [10]:

*# Статистические данные*

happy\_data.describe()

Out[10]:

|  | **Overall rank** | **Score** | **GDP per capita** | **Social support** | **Healthy life expectancy** | **Freedom to make life choices** | **Generosity** | **Perceptions of corruption** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 156.000000 | 156.000000 | 156.000000 | 156.000000 | 156.000000 | 156.000000 | 156.000000 | 155.000000 |
| **mean** | 78.500000 | 5.375917 | 0.891449 | 1.213237 | 0.597346 | 0.454506 | 0.181006 | 0.112000 |
| **std** | 45.177428 | 1.119506 | 0.391921 | 0.302372 | 0.247579 | 0.162424 | 0.098471 | 0.096492 |
| **min** | 1.000000 | 2.905000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 39.750000 | 4.453750 | 0.616250 | 1.066750 | 0.422250 | 0.356000 | 0.109500 | 0.051000 |
| **50%** | 78.500000 | 5.378000 | 0.949500 | 1.255000 | 0.644000 | 0.487000 | 0.174000 | 0.082000 |
| **75%** | 117.250000 | 6.168500 | 1.197750 | 1.463000 | 0.777250 | 0.578500 | 0.239000 | 0.137000 |
| **max** | 156.000000 | 7.632000 | 2.096000 | 1.644000 | 1.030000 | 0.724000 | 0.598000 | 0.457000 |

In [11]:

*# Удаляем столбец Overall rank*

happy\_data = happy\_data.drop('Overall rank', axis = 1)

## 3) Визуальное исследование датасета

In [12]:

*# Гистограммы для всех признаков*

happy\_data.hist(bins=30, figsize = (15,7))

Out[12]:

array([[<matplotlib.axes.\_subplots.AxesSubplot object at 0x0000019B2571CF48>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x0000019B25E3CBC8>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x0000019B25E757C8>],

[<matplotlib.axes.\_subplots.AxesSubplot object at 0x0000019B25EAB548>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x0000019B25EE12C8>,

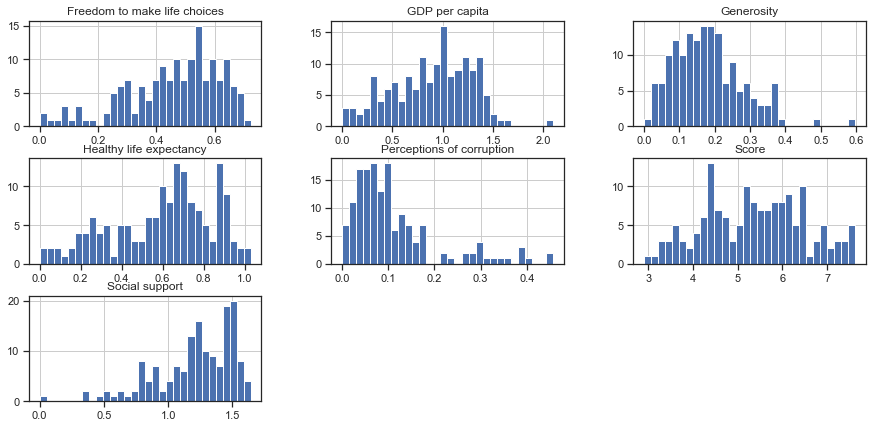
<matplotlib.axes.\_subplots.AxesSubplot object at 0x0000019B25F17048>],

[<matplotlib.axes.\_subplots.AxesSubplot object at 0x0000019B25F44D88>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x0000019B25F7EB08>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x0000019B25F84988>]],

dtype=object)



In [13]:

*# Диаграммы рассеяние для всех признаков*

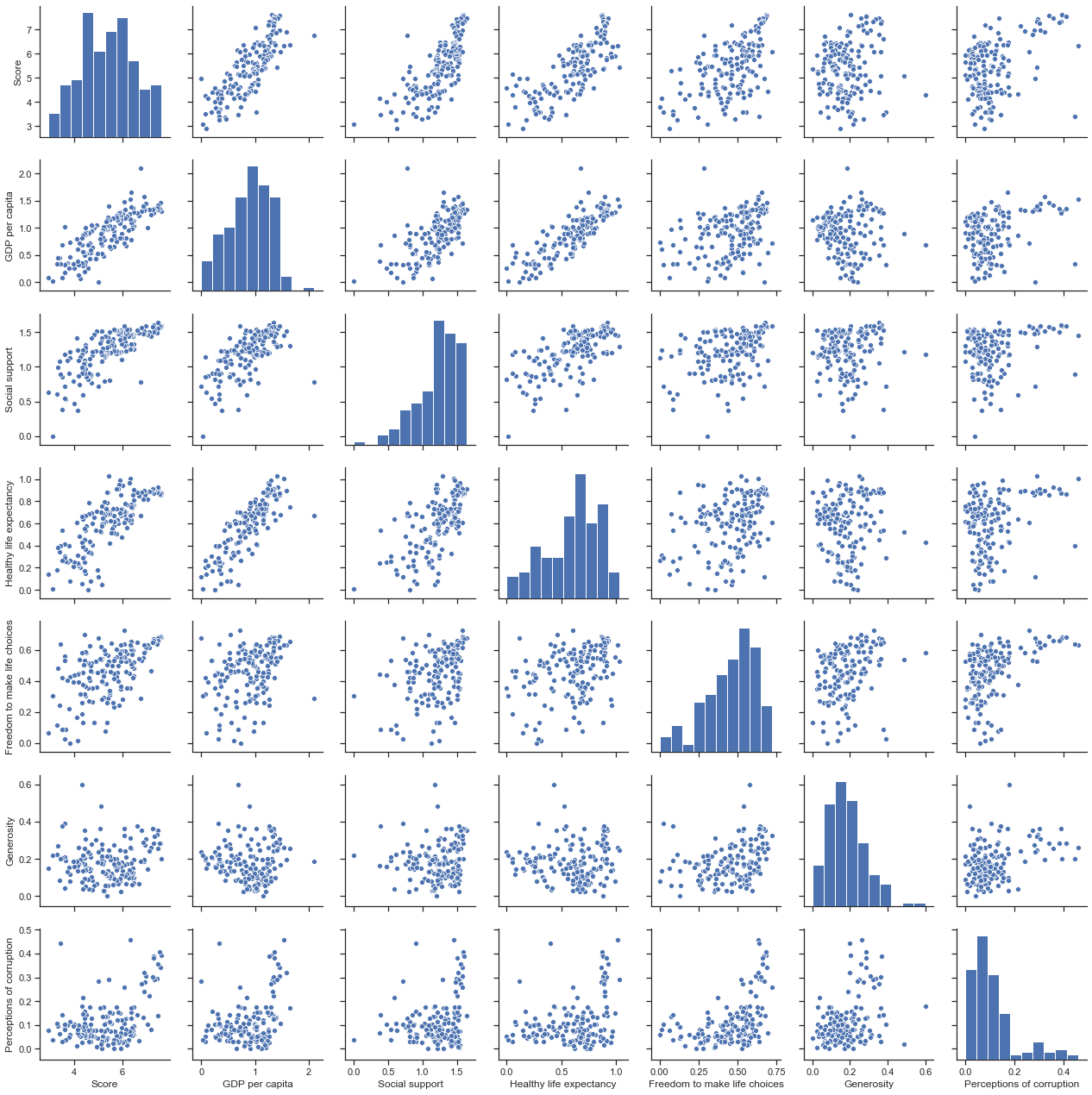
plt.figure(figsize=(12,6))

sns.pairplot(happy\_data)

Out[13]:

<seaborn.axisgrid.PairGrid at 0x19b265745c8>

<Figure size 864x432 with 0 Axes>



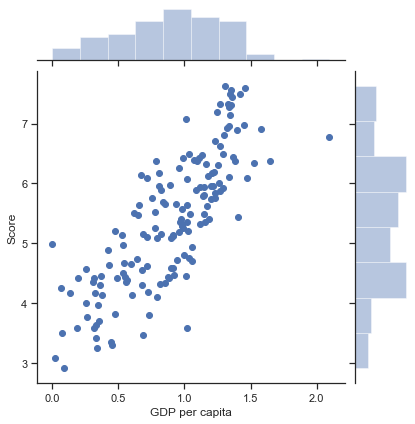
In [14]:

*# Увеличенные диаграммы рассеяния для признаков, которые имеют зависимость с уровнем счастья*

sns.jointplot(x = "GDP per capita", y = "Score", kind="scatter", data = happy\_data)

Out[14]:

<seaborn.axisgrid.JointGrid at 0x19b27a9e4c8>

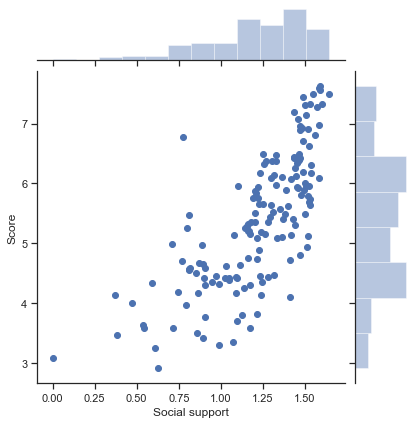


In [15]:

sns.jointplot(x = "Social support", y = "Score", kind="scatter", data = happy\_data)

Out[15]:

<seaborn.axisgrid.JointGrid at 0x19b2951f188>

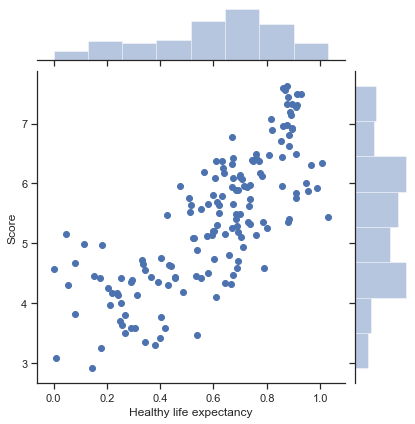


In [16]:

sns.jointplot(x = "Healthy life expectancy", y = "Score", kind="scatter", data = happy\_data)

Out[16]:

<seaborn.axisgrid.JointGrid at 0x19b2965d808>



In [17]:

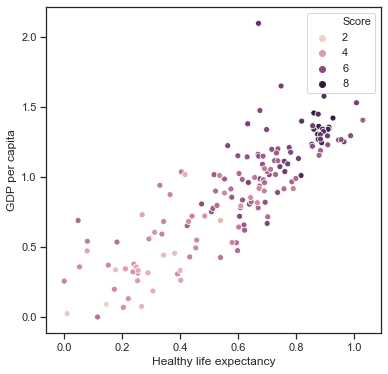
*# Влияние целевого признака*

fig, ax = plt.subplots(figsize=(6,6))

sns.scatterplot(ax=ax, x="Healthy life expectancy", y="GDP per capita", data=happy\_data, hue='Score')

Out[17]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x19b29769748>



In [18]:

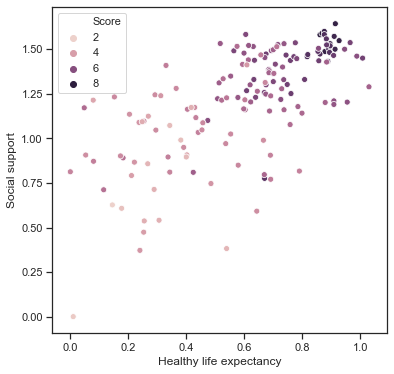
*# Влияние целевого признака*

fig, ax = plt.subplots(figsize=(6,6))

sns.scatterplot(ax=ax, x="Healthy life expectancy", y="Social support", data=happy\_data, hue='Score')

Out[18]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x19b2961f308>



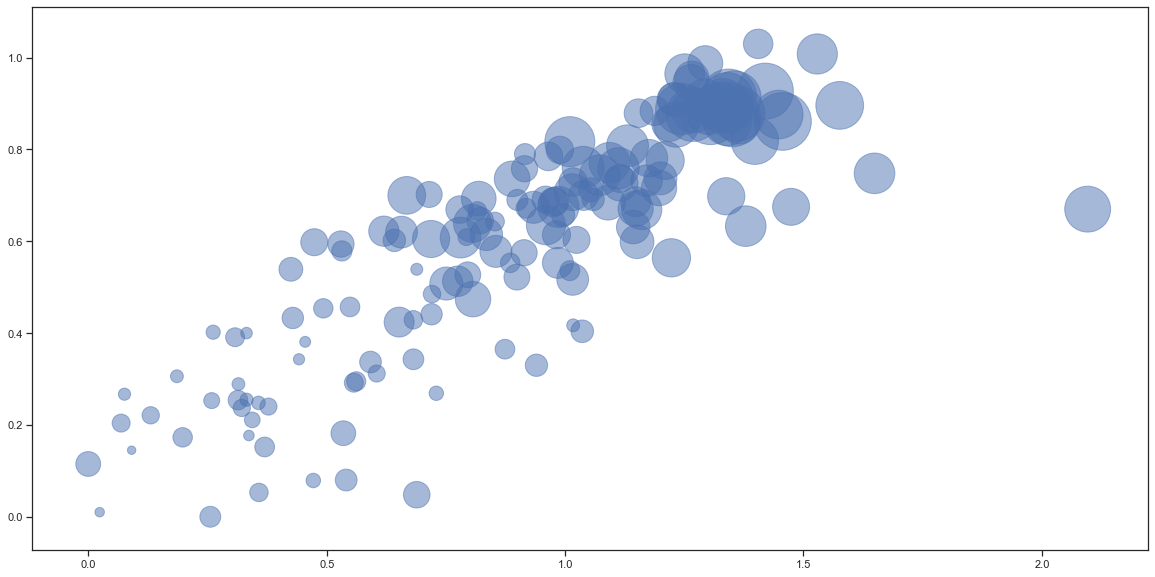
In [19]:

plt.figure(figsize=(20,10))

plt.scatter(happy\_data['GDP per capita'], happy\_data['Healthy life expectancy'], s=(happy\_data['Score']\*\*4), alpha=0.5)

Out[19]:

<matplotlib.collections.PathCollection at 0x19b2990b388>



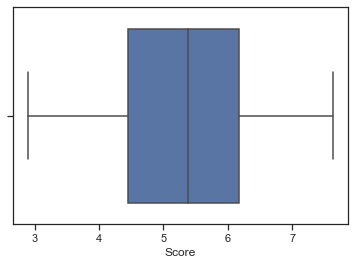
In [20]:

*# Одномерное распределение вероятности*

sns.boxplot(x=happy\_data['Score'])

Out[20]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x19b29d4a048>

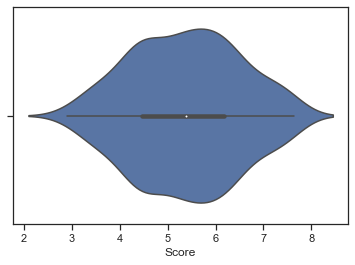


In [21]:

sns.violinplot(x=happy\_data['Score'])

Out[21]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x19b29d53f48>



## 4) Корреляции признаков

In [22]:

corr\_matrix = happy\_data.corr()

In [23]:

corr\_matrix['Score']

Out[23]:

Score 1.000000

GDP per capita 0.802124

Social support 0.745760

Healthy life expectancy 0.775814

Freedom to make life choices 0.544280

Generosity 0.135825

Perceptions of corruption 0.405292

Name: Score, dtype: float64

In [24]:

sns.heatmap(happy\_data.corr(), annot=**True**, fmt='.3f')

Out[24]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x19b299d61c8>

