Happiest Countries in the World

Bogdanovich Alena

DA_2023

Планирование работ

ДАТА	ОПИСАНИЕ
21.09.2023	Предзащита идеи
25.09.2023	Сбор данных, определение метрик
02.10.2023	Оценка качества и объема данных
05.10.2023	Обработка, очистка данных
10.10.2023	Анализ данных, расчет показателей, поиск взаимосвязей
17.10.2023	Визуализация данных
24.10.2023	Выводы исследования
31.10.2023	Оформление презентации
17.11.2023	Защита проекта

Цели и задачи проекта

- Какая страна самая счастливая в мире?
- Какие факторы больше влияют на счастье страны?
- Как менялся показатель счастья с течением времени?

Инструменты и методы

Веб-приложение с открытым исходным кодом **Jupyter notebook.**

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

Язык программирования **Python** и его библиотеки (**Pandas, Openpyxl**, **Matplotlib, NumPy, Seaborn, Statsmodels**).

Язык структурированных запросов (**SQL**) - язык программирования для хранения и обработки информации в реляционной базе данных.

Apache Superset — открытое программное обеспечение для исследования и визуализации данных.

Сбор данных

Данные в формате csv и xlsx для проекта загружены с сайта https://worldhappiness.report:

- WHR2023.csv
- DataForTable2.1WHR2023.xlsx

Просмотр данных, определение метрик

- Для прочтения файла **csv** загружаем библиотеку **Pandas:**

```
import pandas as pd

happiness2023 = pd.read_csv("WHR2023.csv")
```

Подробно в 2023report.ipynb

- Для прочтения файла **xlsx** загружаем дополнительно библиотеку **Openpyxl:**

```
!pip install openpyxl

happiness = pd.read_excel("DataForTable2.1WHR2023.xlsx")
```

Подробно в happiness_total.ipynb

Оценка качества и объема данных

Отчет 2023:

```
happiness2023.shape
```

```
(137, 19)
```

```
happiness2023.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 137 entries, 0 to 136
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   Country name                          137 non-null    object 
 1   Ladder score                          137 non-null    float64
 2   Standard error of ladder score        137 non-null    float64
 3   upperwhisker                          137 non-null    float64
 4   lowerwhisker                          137 non-null    float64
 5   Logged GDP per capita                 137 non-null    float64
 6   Social support                        137 non-null    float64
 7   Healthy life expectancy               136 non-null    float64
 8   Freedom to make life choices          137 non-null    float64
 9   Generosity                            137 non-null    float64
 10  Perceptions of corruption             137 non-null    float64
 11  Ladder score in Dystopia              137 non-null    float64
 12  Explained by: Log GDP per capita      137 non-null    float64
 13  Explained by: Social support          137 non-null    float64
 14  Explained by: Healthy life expectancy 136 non-null    float64
 15  Explained by: Freedom to make life choices 137 non-null float64
 16  Explained by: Generosity              137 non-null    float64
 17  Explained by: Perceptions of corruption 137 non-null  float64
 18  Dystopia + residual                   136 non-null    float64
```

Подробно в 2023report.ipynb

Отчет 2005-2022:

Узнаем какой самый ранний год представлен в данной таблице. Также проверим последний год.

```
happiness.year.min(), happiness.year.max()
```

```
(2005, 2022)
```

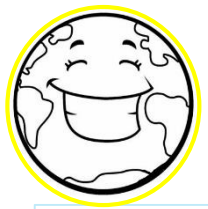
```
happiness.shape
```

```
(2199, 11)
```

```
happiness.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2199 entries, 0 to 2198
Data columns (total 11 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Country name                    2199 non-null   object 
 1   year                            2199 non-null   int64  
 2   Life Ladder                     2199 non-null   float64
 3   Log GDP per capita              2179 non-null   float64
 4   Social support                  2186 non-null   float64
 5   Healthy life expectancy at birth 2145 non-null  float64
 6   Freedom to make life choices    2166 non-null   float64
 7   Generosity                      2126 non-null   float64
 8   Perceptions of corruption       2083 non-null   float64
 9   Positive affect                 2175 non-null   float64
 10  Negative affect                 2183 non-null   float64
dtypes: float64(9), int64(1), object(1)
memory usage: 189.1+ KB
```

Подробно в happiness_total.ipynb



Обработка/очистка данных

Отчет 2023:

Подробно в 2023report.ipynb

- Удалим столбцы с лишними данными.

```
happiness2023 = happiness2023.drop(columns = ['Standard error of ladder score', 'upperwhisker', 'lowerwhisker', ...],  axis = 1)
```

- Переименуем столбцы датасета.

```
happiness2023.rename(columns = { 'Country name':'Страна', 'Ladder score':'Рейтинг', 'Logged GDP per capita':'ВВП на душу населения', 'Social support':'Социальная поддержка', 'Healthy life expectancy':'Ожидаемая продолжительность здоровой жизни', 'Freedom to make life choices':'Свобода жизненного выбора', 'Generosity':'Щедрость', 'Perceptions of corruption':'Восприятие коррупции'}, inplace = True)
```

- Сохраним обработанный датасет в csv-файл для дальнейшего анализа.

```
happiness2023.to_csv('happiness2023.csv', index=False)
```

- Проверим наличие нулевых значение в датасете и удалим при наличии.

```
happiness2023.isnull().sum()
```

```
happiness2023 = happiness2023.dropna()
```

Отчет 2005-2022:

Подробно в happiness_total.ipynb

- Удалим столбцы с лишними данными.

```
happiness = happiness.drop(columns = ['Positive affect', 'Negative affect'], axis = 1)
```

- Переименуем столбцы датасета как в отчете 2023 года.

```
happiness.rename(columns = { 'Country name':'Страна', 'year':'Год', 'Life Ladder':'Рейтинг', 'Log GDP per capita':'ВВП на душу населения', 'Social support':'Социальная поддержка', 'Healthy life expectancy at birth':'Ожидаемая продолжительность здоровой жизни', 'Freedom to make life choices':'Свобода жизненного выбора', 'Generosity':'Щедрость', 'Perceptions of corruption':'Восприятие коррупции'}, inplace = True)
```

- В отчете 2023 года добавим столбец с годом и заполним его.

```
happiness2023['Год'] = 2023
```

- Объединим отчеты с 2005 года по 2023 год.

```
happiness_total = pd.concat([happiness, happiness2023])
```

- Проверим количество строк объединенного отчета.

```
happiness.count(), happiness2023.count(), happiness_total.count()
```

Анализ данных, расчет показателей, поиск взаимосвязей

Отчет 2023:

- Получим первичное представление о статистических характеристиках нашего датасета.

```
happiness2023.describe()
```

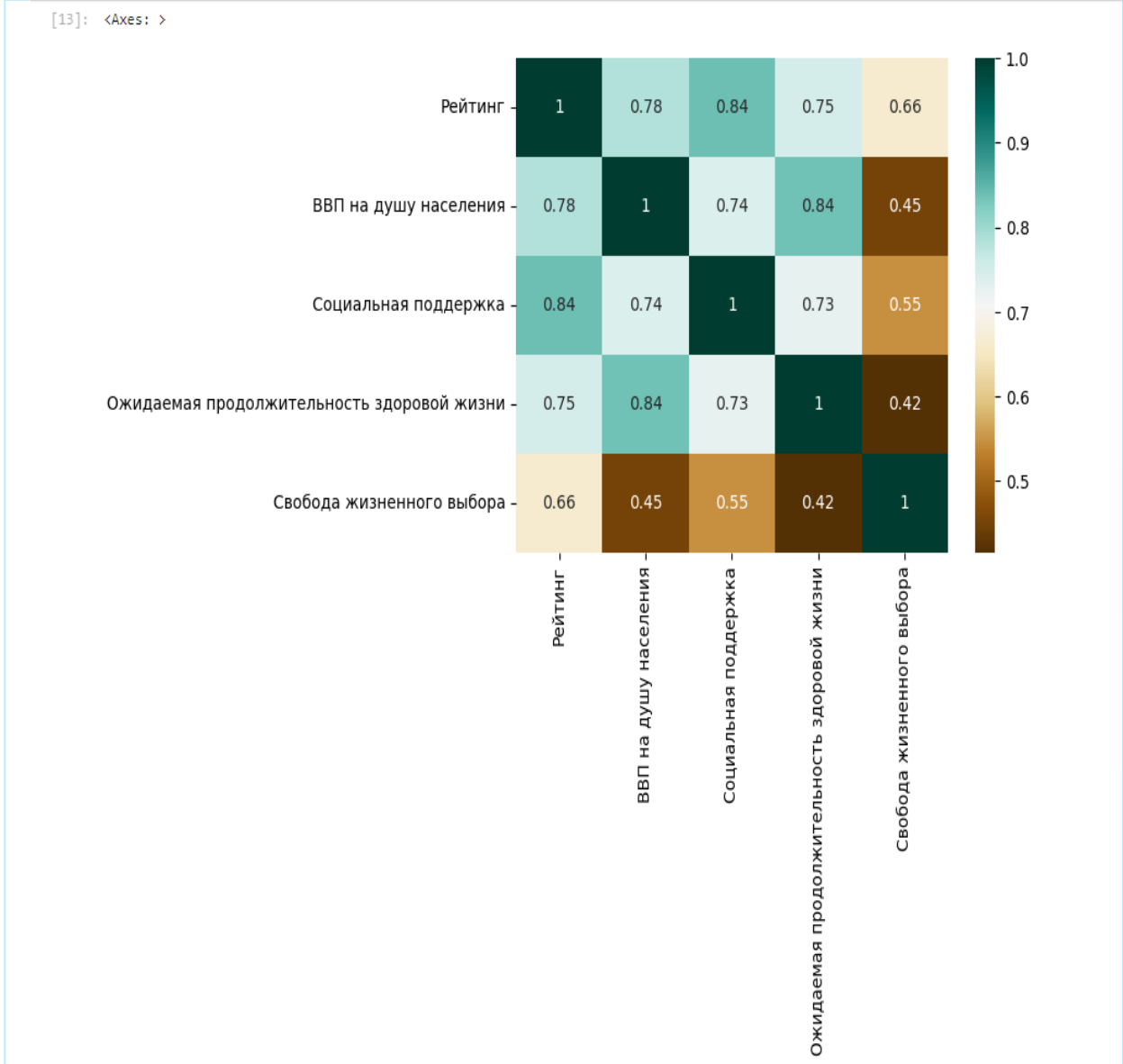
- Для изучения отношений между числовыми столбцами датасета загрузим библиотеки NumPy, Matplotlib и Seaborn.

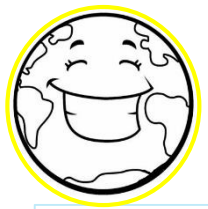
```
import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

sns.pairplot(happiness2023, corner=True)
```

- Для значений из диаграмм, в которых видна линейная зависимость, нарисуем тепловую карту.

```
df_1 = happiness2023[['Рейтинг', 'ВВП на душу населения', 'Социальная поддержка', 'Ожидаемая продолжительность здоровой жизни', 'Свобода жизненного выбора']]

sns.heatmap(df_1.corr(), annot=True, cmap='BrBG')
```

Подробно в 2023report.ipynb





Анализ данных, расчет показателей, поиск взаимосвязей

Отчет 2023:

- Для пар с самой сильной корреляцией применим линейную регрессию используя библиотеку Statsmodels и входящий в нее модуль линейной регрессии Linear Regression.

```
!pip install statsmodels

import statsmodels.formula.api as smf
```

Рейтинг и Социальной поддержка

```
df_2 = df_1[['Рейтинг','Социальная поддержка']]

df_2.rename(columns = { 'Рейтинг':'Score', 'Социальная поддержка':'SocialSupport'}, inplace = True)

model = smf.ols('Score ~ SocialSupport' ,data = df_2)

result = model.fit()

print(result.summary())
```

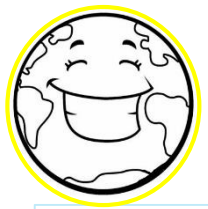
ВВП на душу населения и Ожидаемая продолжительность здоровой жизни

```
df_3 = df_1[['ВВП на душу населения','Ожидаемая продолжительность здоровой жизни']]

df_3.rename(columns = { 'ВВП на душу населения':'GDP', 'Ожидаемая продолжительность здоровой жизни':'HealthyLife'}, inplace = True)

model = smf.ols('HealthyLife ~ GDP' ,data = df_3)

result = model.fit()

print(result.summary())
```

Подробно в 2023report.ipynb

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Score   R-squared:                       0.702
Model:                            OLS   Adj. R-squared:                  0.700
Method:                 Least Squares   F-statistic:                     316.2
Date:                Thu, 26 Oct 2023   Prob (F-statistic):           4.49e-37
Time:                        13:06:37   Log-Likelihood:                -128.23
No. Observations:                 136   AIC:                             260.5
Df Residuals:                     134   BIC:                             266.3
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.3577      0.336     -1.064      0.289      -1.023       0.307
SocialSupport  7.3904      0.416     17.781      0.000       6.568       8.212
==============================================================================
Omnibus:                        2.196   Durbin-Watson:                   1.419
Prob(Omnibus):                  0.334   Jarque-Bera (JB):                2.190
Skew:                          -0.301   Prob(JB):                        0.334
Kurtosis:                       2.843   Cond. No.                         12.7
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            HealthyLife   R-squared:                       0.701
Model:                            OLS   Adj. R-squared:                  0.699
Method:                 Least Squares   F-statistic:                     314.9
Date:                Thu, 26 Oct 2023   Prob (F-statistic):           5.47e-37
Time:                        13:06:59   Log-Likelihood:                -348.17
No. Observations:                 136   AIC:                             700.3
Df Residuals:                     134   BIC:                             706.2
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     27.3366      2.138     12.786      0.000      23.108      31.565
GDP            3.9799      0.224     17.744      0.000       3.536       4.424
==============================================================================
Omnibus:                       17.212   Durbin-Watson:                   1.725
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               70.484
Skew:                           0.094   Prob(JB):                     4.95e-16
Kurtosis:                       6.522   Cond. No.                         76.2
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```



Анализ данных, расчет показателей, поиск взаимосвязей

Отчет 2005-2023:

- Узнаем сколько стран за все годы оценки счастья с мире попадали в рейтинги и их названия.

```
happiness_total['Страна'].value_counts()
```

```
happiness_total['Страна'].unique()
```

- Посмотрим как изменялся максимальный, минимальный и средний уровень счастья в разные годы.

```
happiness_total.groupby(['Год']).agg({'Рейтинг': ['max', 'min', 'mean']})
```

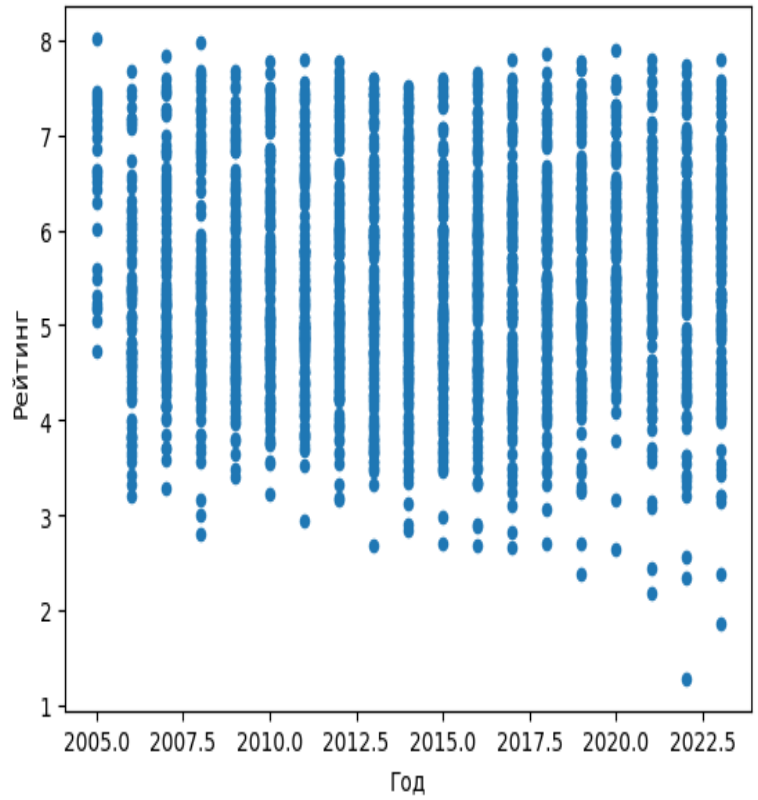
```
happiness_total.plot(kind='scatter', x='Год', y='Рейтинг')
```

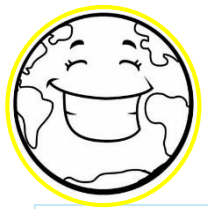
Подробно в happiness_total.ipynb

[41]:

	Рейтинг		
	max	min	mean
Год			
2005	8.018934	4.718734	6.446164
2006	7.672449	3.202429	5.196935
2007	7.834233	3.280247	5.418291
2008	7.970892	2.807855	5.418554
2009	7.683359	3.407508	5.457640
2010	7.770515	3.229129	5.496782
2011	7.788232	2.936221	5.424088
2012	7.776209	3.164491	5.443612
2013	7.593794	2.687553	5.393302
2014	7.507559	2.838959	5.386267
2015	7.603434	2.701591	5.400948
2016	7.659843	2.693061	5.396381
2017	7.788252	2.661718	5.460421
2018	7.858107	2.694303	5.498683
2019	7.780348	2.375092	5.570995
2020	7.889350	2.633753	5.727539
2021	7.794378	2.178809	5.636193
2022	7.728998	1.281271	5.585126
2023	7.804000	1.859000	5.539796

[30]: <Axes: xlabel='Год', ylabel='Рейтинг'>





Анализ данных, расчет показателей, поиск взаимосвязей

Отчет 2005-2023:

- Выберем в помощью .loc и текстового среза нужные столбцы:

```
df_1 = happiness_total.loc[:,'Страна':'Рейтинг']
```

- Создадим на основании нового датафрейма сводную таблицу:

```
df_pivot = df_1.pivot_table('Рейтинг', index='Год', columns='Страна', aggfunc = 'max').reset_index()
```

- Выберем данные по странам, которые входили в head(5) и tail(5) отчета 2023 года, а также Беларусь:

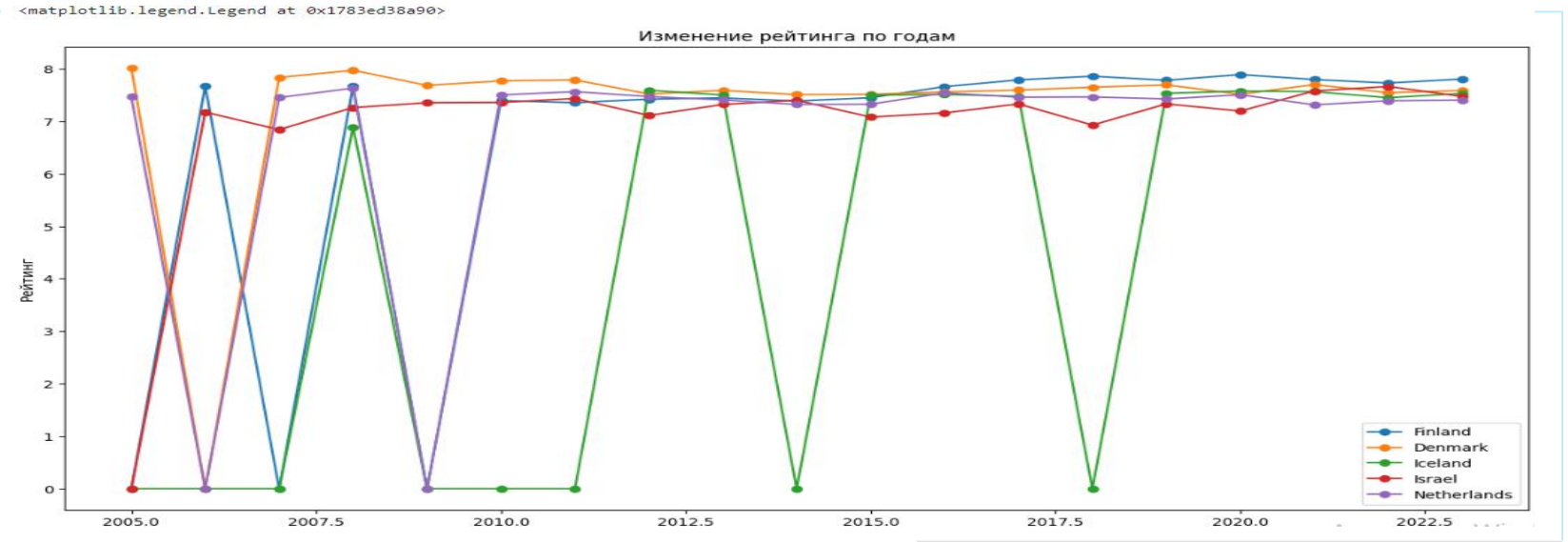
```
df = df_pivot.loc[:,('Год','Finland','Denmark','Iceland','Israel', 'Netherlands', 'Congo (Kinshasa)', 'Zimbabwe', 'Sierra Leone',
'Lebanon', 'Afghanistan', 'Belarus')]
```

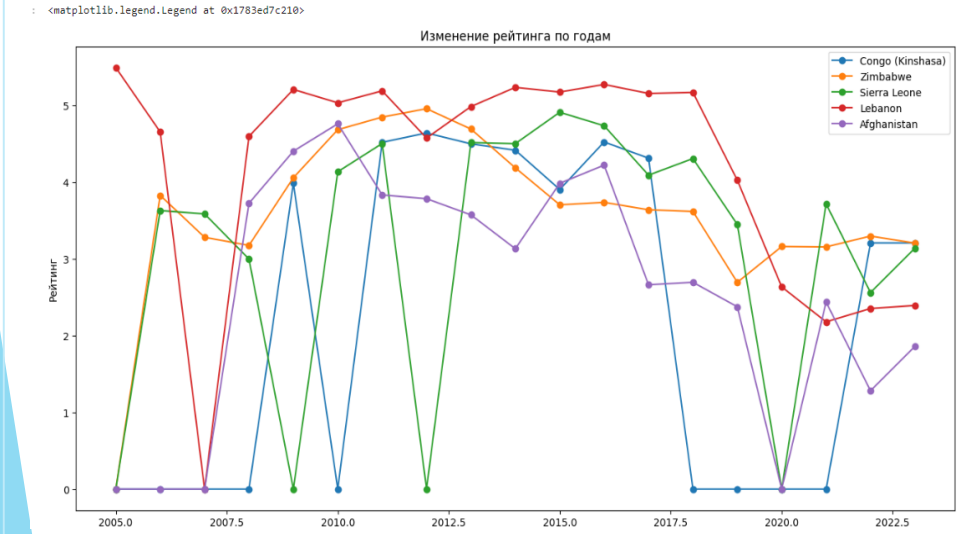
- Заполним ячейки, которые не имеют значение, когда страна не попадала в рейтинг:

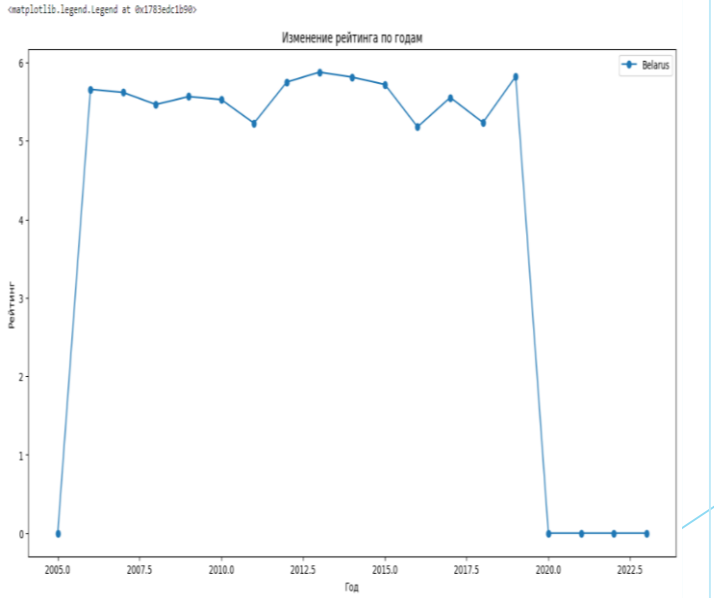
```
df.fillna(0, inplace=True)
```

Подробно в Change_score_year_country.ipynb

Анализ данных, расчет показателей, поиск взаимосвязей







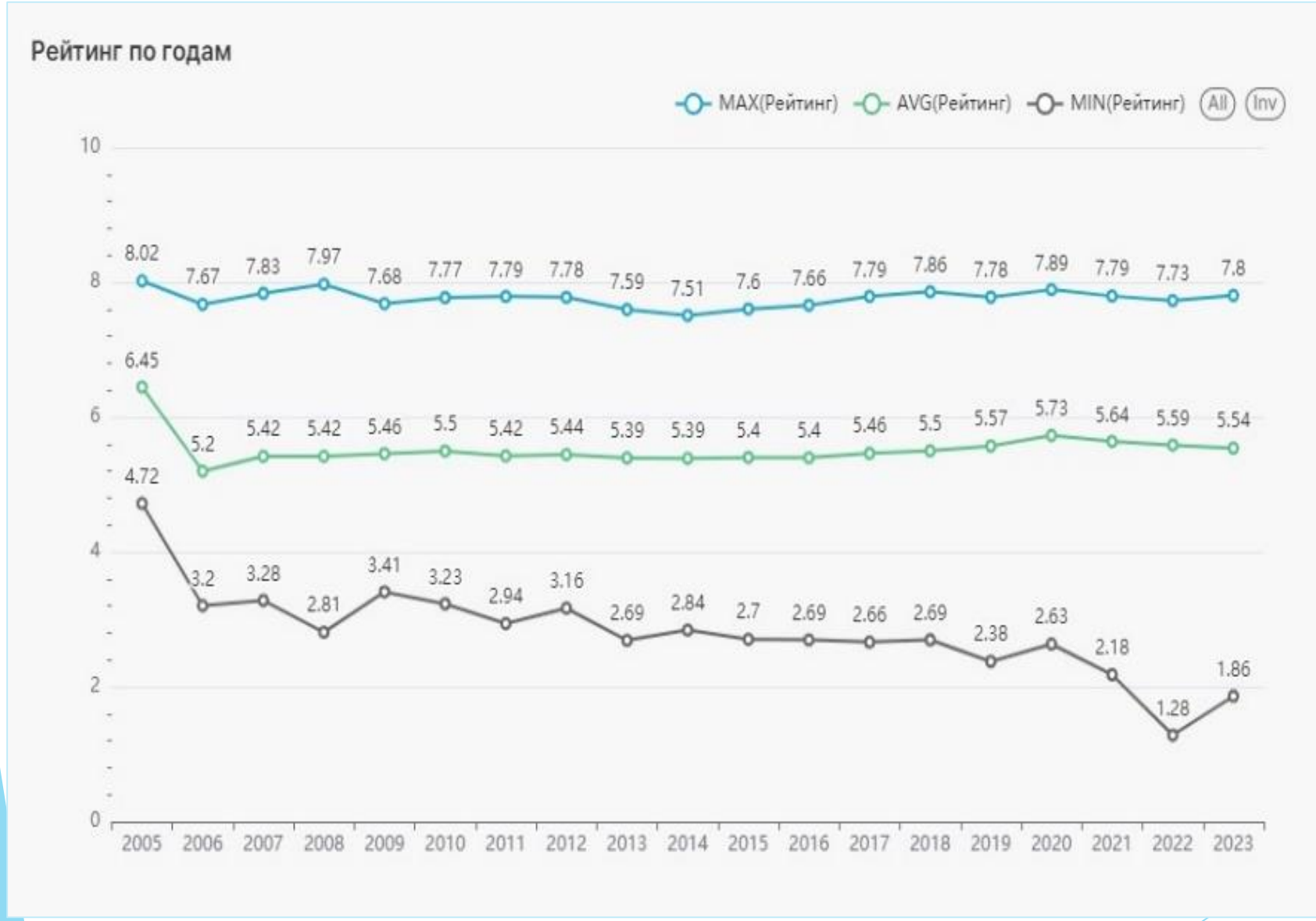
Отчет 2005-2023:

- Построим графики отдельно для каждой группы, чтобы увидеть изменение рейтинга по годам.

```
plt.figure(figsize=(16,8))
plt.plot(df['Год'], df['Finland'], marker='o')
plt.plot(df['Год'], df['Denmark'], marker='o')
plt.plot(df['Год'], df['Iceland'], marker='o')
plt.plot(df['Год'], df['Israel'], marker='o')
plt.plot(df['Год'], df['Netherlands'], marker='o')
plt.plot(df['Год'], df['Congo (Kinshasa)'], marker='o')
plt.plot(df['Год'], df['Zimbabwe'], marker='o')
plt.plot(df['Год'], df['Sierra Leone'], marker='o')
plt.plot(df['Год'], df['Lebanon'], marker='o')
plt.plot(df['Год'], df['Afghanistan'], marker='o')
plt.plot(df['Год'], df['Belarus'], marker='o')
plt.xlabel('Год')
plt.ylabel('Рейтинг')
plt.title('Изменение рейтинга по годам и странам')
plt.legend(['Finland', 'Denmark','Iceland','Israel',
'Netherlands', 'Congo (Kinshasa)','Zimbabwe','Sierra
Leone', 'Lebanon', 'Afghanistan','Belarus'])
```

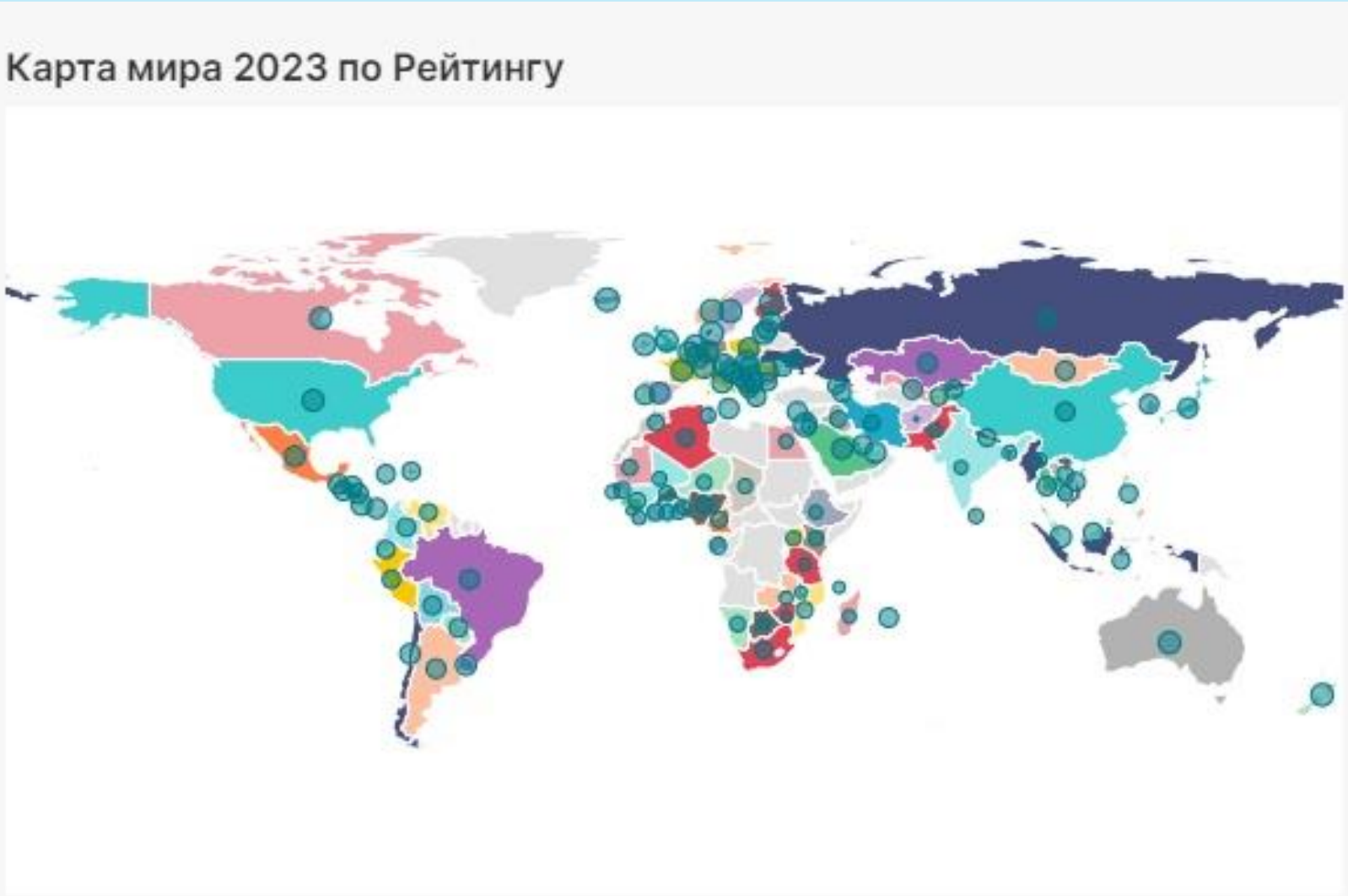
Подробно в Change_score_year_country.ipynb

Визуализация данных



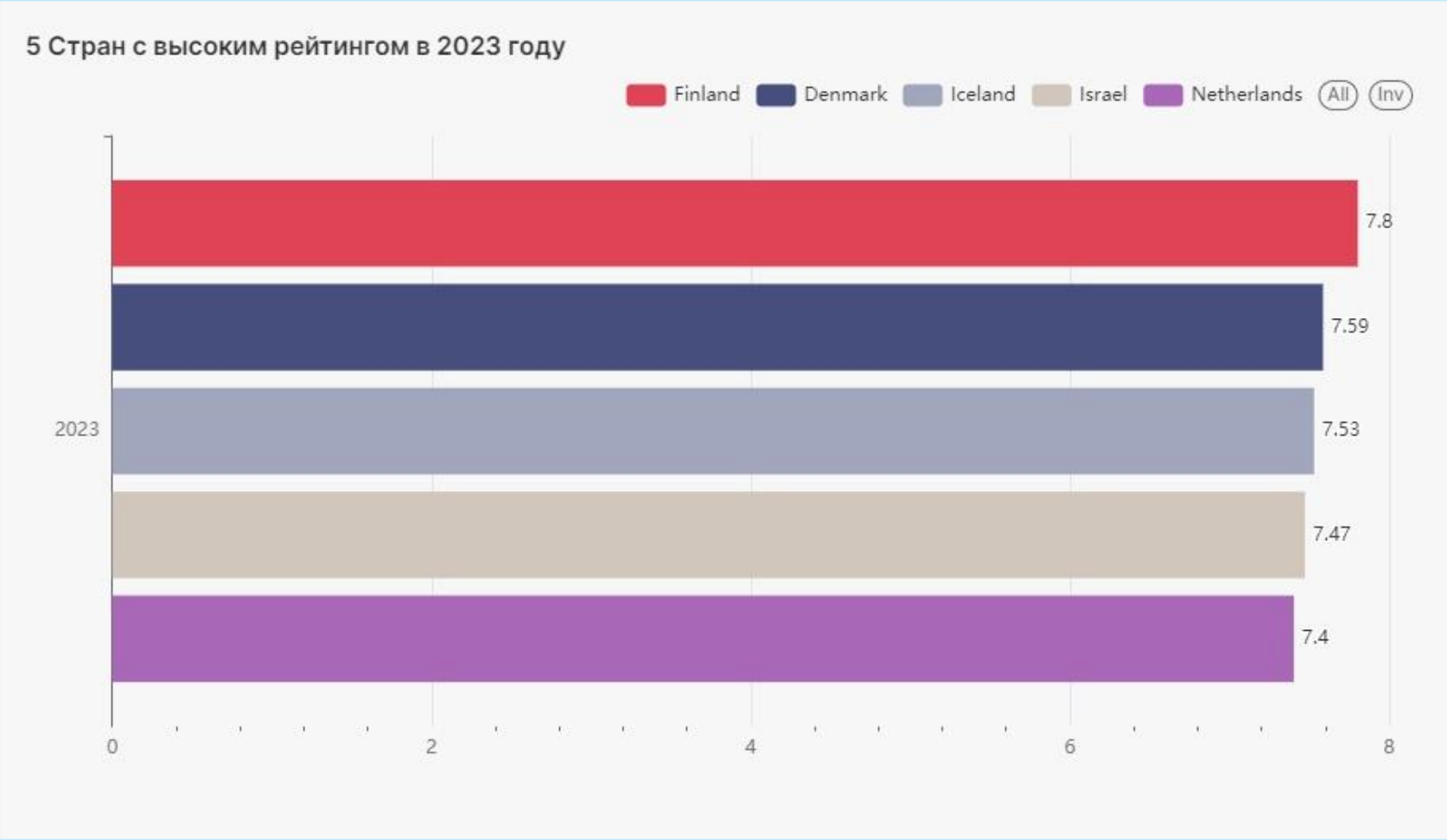

```
SELECT "Год" AS "Год",
   max("Рейтинг") AS "MAX(Рейтинг)",
   min("Рейтинг") AS "MIN(Рейтинг)",
   AVG("Рейтинг") AS "AVG(Рейтинг)"
FROM tres."Happiness"
GROUP BY "Год"
ORDER BY "MAX(Рейтинг)" ;
```

Визуализация данных




```
SELECT "Страна" AS "Страна",
count("Страна") AS "COUNT(Страна)",
max("Рейтинг") AS "MAX(Рейтинг)",
FROM tres."Happiness"
WHERE "Год" = 2023
GROUP BY "Страна"
ORDER BY "COUNT(Страна)";
```

Визуализация данных




```
SELECT "Год" AS "Год",
       "Страна" AS "Страна",
       max("Рейтинг") AS "MAX(Рейтинг)"
FROM tres."Happiness"
JOIN
 (SELECT "Страна" AS "Страна",
       max("Рейтинг") AS mme_inner__
  FROM tres."Happiness"
  WHERE "Год" = 2023
  GROUP BY "Страна"
  ORDER BY mme_inner__)
WHERE "Год" = 2023
GROUP BY "Год“,
                "Страна"
ORDER BY "MAX(Рейтинг)" ;
```

Визуализация данных

- График самых высоких рейтингов по годам и странам.

```
SELECT "Год" AS "Год",
    "Страна" AS "Страна",
    max("Рейтинг") AS "MAX(Рейтинг)"
FROM tres."Happiness"
GROUP BY Страна"
            "Год“
ORDER BY "MAX(Рейтинг)" ;
```

Выводы исследования

- Самой счастливой страной в мире является Финляндия. Как видно из исследования она, как и Дания, имели самые высокие рейтинги за всю историю вычислений.
- В пятерку лидеров по счастью в 2023 году вошли страны Северной Европы и Израиль. На показатель счастья согласно анализа в большей мере влияет Социальная поддержка, а не уровень ВВП. Так самые богатые страны занимают более низкие места в рейтинге.
- Показатель счастья в среднем снижается. В рейтинги попадают страны с очень низкими значениями.

Погода не оказывает большого влияния на счастье, также как и уровень богатства. Сильное чувство общественной поддержки и взаимного доверия делает людей более счастливыми.