Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота №1
Підсумкові функції та відображення

Виконала

студентка групи ІТ-91: Перевірив:

Луцай Катерина Новотарський М. А.

Київ 2023

Мета: навчитися вибрати відповідні дані з DataFrame або Series, вибирати правильні дані з набору даних.

Варіант: 15 – dataset “Cancer rate by countries” (https://www.kaggle.com/dianapratiwi/cancer-rate-by-countries)

Хід виконання роботи:

```
[1]  import pandas as pd
     import numpy as np
```

```
[16] # connecting to gdrive
     from google.colab import drive
     drive.mount('/content/gdrive', force_remount=True)
     gdrive_path = f"/content/gdrive/MyDrive/ds/"
```

```
Mounted at /content/gdrive
```

```
[18] # reading dataset to pandas dataframe
     data = pd.read_csv("/content/gdrive/MyDrive/ds/Cancer_rate_by_countries.csv", index_col=0)
     # show first 20 rows
     data.head(20)
```

	Rank	Country	Cancer_Rate
0	1	Australia	468.0
1	2	New Zealand	438.1
2	3	Ireland	373.7
3	4	Hungary	368.1
4	5	United States	352.2
5	6	Belgium	345.8
6	7	France	344.1
7	8	Denmark	340.4
8	9	Norway	337.8
9	10	Netherlands	334.1
10	11	Canada	334.0
11	12	France (New Caledonia)	324.2
12	13	United Kingdom	319.2
13	14	South Korea	313.5
14	15	Germany	313.1

```
[4]  # show dataframe general info
     data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50 entries, 0 to 49
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Rank         50 non-null     int64
 1   Country      50 non-null     object
 2   Cancer_Rate  50 non-null     float64
dtypes: float64(1), int64(1), object(1)
memory usage: 1.6+ KB
```

Dataset has 3 columns and all rows are filled with values

```
# show dataframe column Cancer_Rate
data.Cancer_Rate
```

```
0     468.0
1     438.1
2     373.7
3     368.1
4     352.2
```

```
[17] # show columns description
     data.Cancer_Rate.describe()
```

```
count     50.000000
mean     294.082000
std       50.026628
min      233.600000
25%      253.900000
50%      286.500000
75%      317.775000
max      468.000000
Name: Cancer_Rate, dtype: float64
```

Column has a numerical data type, thus description shows its size, mean, standard deviation, min and max, and quartiles.

```
[6]  data.Country.describe()
```

```
count           50
unique          50
top       Australia
freq             1
Name: Country, dtype: object
```

Column has a string data type, thus description shows its size, number of unique values, the most frequent entry, and its frequency.

```
[7]  # show column mean value
     data.Cancer_Rate.mean()
```

```
294.08200000000005
```

```
[8]  # show a list of column's unique values
     data.Country.unique()
```

```
array(['Australia', 'New Zealand', 'Ireland', 'Hungary', 'United States',
       'Belgium', 'France', 'Denmark', 'Norway', 'Netherlands', 'Canada',
       'France (New Caledonia)', 'United Kingdom', 'South Korea',
       'Germany', 'Switzerland', 'Luxembourg', 'Serbia', 'Slovenia',
       'Latvia', 'Slovakia', 'Czech Republic', 'Sweden', 'Italy',
       'Croatia', 'Lithuania', 'Estonia', 'Greece', 'Spain', 'Finland',
       'Uruguay', 'Belarus', 'Portugal', 'Iceland', 'France (Guadeloupe)',
       'United States (Puerto Rico)', 'Moldova', 'Poland', 'Cyprus',
       'France (Martinique)', 'Malta', 'Singapore', 'Japan', 'Austria',
       'Barbados', 'France (French Guiana)', 'Bulgaria', 'Lebanon',
       'France (French Polynesia)', 'Israel'], dtype=object)
```

```
[9]  # show column's entries and how many times they appear
     data.Country.value_counts()
```

```
Australia                  1
Poland                     1
Greece                     1
Spain                      1
Finland                    1
Uruguay                    1
Belarus                    1
```

```
[12] # create a column of mapped values from another column series by substracting
     # column's mean value from all its rows
     cancer_mean = data.Cancer_Rate.mean()
     data["Cancer_div"] = data.Cancer_Rate.map(lambda p: p - cancer_mean)
     data
```

18	19	Slovenia	304.9	10.818
19	20	Latvia	302.2	8.118
20	21	Slovakia	297.5	3.418
21	22	Czech Republic	296.7	2.618
22	23	Sweden	294.7	0.618
23	24	Italy	290.6	-3.482
24	25	Croatia	287.2	-6.882
25	26	Lithuania	285.8	-8.282
26	27	Estonia	283.2	-10.782

Above is Cancer Rate diviation next to Cancer Rate

```
# function to make Country column entries in the upper case, and
# reset Cancer_div to the absolute values (negatives change to positives)
def remean_points(row):
    row.Cancer_div = np.abs(row.Cancer_Rate - cancer_mean)
    row.Country = row.Country.upper()
    return row

# applying the defined function to all dataframe rows
data = data.apply(remean_points, axis=1)
data
```

18	19	SLOVENIA	304.9	10.818
19	20	LATVIA	302.2	8.118
20	21	SLOVAKIA	297.5	3.418
21	22	CZECH REPUBLIC	296.7	2.618
22	23	SWEDEN	294.7	0.618
23	24	ITALY	290.6	3.482
24	25	CROATIA	287.2	6.882
25	26	LITHUANIA	285.8	8.282
26	27	ESTONIA	283.3	10.782

Above is Cancer Rate diviation absolute values next to Cancer Rate

```
[14] # show difference between a Cancer_rate series and its mean
     data.Cancer_Rate - cancer_mean
```

```
0     173.918
1     144.018
2      79.618
3      74.018
4      58.118
5      51.718
6      50.018
7      46.318
8      43.718
```

```
[15] # concat Country and stringified Rank values
     data.Country + " " + data.Rank.astype("string")
```

```
0                        AUSTRALIA 1
1                      NEW ZEALAND 2
2                          IRELAND 3
3                          HUNGARY 4
4                    UNITED STATES 5
5                          BELGIUM 6
6                           FRANCE 7
7                          DENMARK 8
8                           NORWAY 9
9                      NETHERLANDS 10
10                          CANADA 11
11         FRANCE (NEW CALEDONIA) 12
12                 UNITED KINGDOM 13
13                    SOUTH KOREA 14
14                        GERMANY 15
15                    SWITZERLAND 16
16                     LUXEMBOURG 17
17                         SERBIA 18
18                       SLOVENIA 19
19                         LATVIA 20
20                       SLOVAKIA 21
21                 CZECH REPUBLIC 22
22                         SWEDEN 23
23                          ITALY 24
24                        CROATIA 25
25                      LITHUANIA 26
26                        ESTONIA 27
27                         GREECE 28
28                          SPAIN 29
29                        FINLAND 30
30                        URUGUAY 31
31                        BELARUS 32
```

Вихідний код у jupyter notebook: https://colab.research.google.com/drive/13WVuqk6b3vXyhtyhHg5heAE5LD1muZDh?usp=sharing

Висновки: було розглянуто основні методи бібліотеки pandas на мові Python для вибору відповідних даних із DataFrame або Series, вибору правильних даних із набору даних, такі як: describe(), mean(), unique(), value_counts(), map(), apply() та інші.