Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 1 з дисципліни

«Прикладні задачі машинного навчання»

«Введення в data science»

Виконав *ІП-15, Дацьо Іван*

(шифр, прізвище, ім'я, по батькові)

Перевірив *Нестерук А.*

прізвище, ім'я, по батькові

Київ 2023

Завдання:

1. На сайті http://www.ukrstat.gov.ua/ обрати дані які для Вас є цікавими, можна використати будь-який ресурс з відкритими даними.
2. Знайти математичне сподівання, медіану, моду, дисперсію, середньоквадратичне відхилення (поясніть їх зміст)
3. Візуалізувати завантажені дані за допомогою гістограми
4. Для цих даних проробити всі дії з пункту колекції Series і DataFrame бібліотеки pandas
5. Прочитати набір даних катастрофи «Титаніка»
6. Завантажити набір даних катастрофи «Титаніка» за URL-адресою
7. Переглянути рядки набору даних катастрофи «Титаніка»
8. Налаштувати назви стовпців
9. Провести простий аналіз даних
10. Побудувати гістограму віку пасажирів

Виконання

1. Імпортуємо необхідні пакети:

In [1]:
```
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import statistics
```

Для виконання лабораторної роботи було використано набір даних, для дослідження країн світу. Завантажимо дані та виведемо дані

:

```
In [2]: data = pd.read_csv("Data1.csv")
        data
```

Out[2]:

	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3530
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200
...	...	...	...	...	...	...	...	...	...	...
163	Venezuela	17.1	28.5	4.91	17.6	16500	45.90	75.4	2.47	13500
164	Vietnam	23.3	72.0	6.84	80.2	4490	12.10	73.1	1.95	1310
165	Yemen	56.3	30.0	5.18	34.4	4480	23.60	67.5	4.67	1310
166	Zambia	83.1	37.0	5.89	30.9	3280	14.00	52.0	5.40	1460
167	Zambia	83.1	37.0	5.89	30.9	3280	14.00	52.0	5.40	1460

168 rows × 10 columns

2. Знайдемо медіану для стовпця child_mort(Death of children under 5 years of age per 1000 live births), тобто це буде значення яке стоїть по середині даних:

```
In [3]: data['child_mort'].median()
Out[3]: 19.5
```

Знайдемо математичне сподівання для стовпця child_mort те, що очікується в середньому :

```
In [4]: data['child_mort'].mean()
Out[4]: 38.53690476190476
```

Знайдемо мода для стовпця child_mort(Death of children under 5 years of age per 1000 live births), тобто це буде найчастіше число смертей дітей на 1000 народжувань:

```
In [5]: data['child_mort'].mode()
Out[5]: 0    4.5
        Name: child_mort, dtype: float64
```

Знайдемо дисперсію (наскільки далеко значення в даних розташовані від середнього значення)

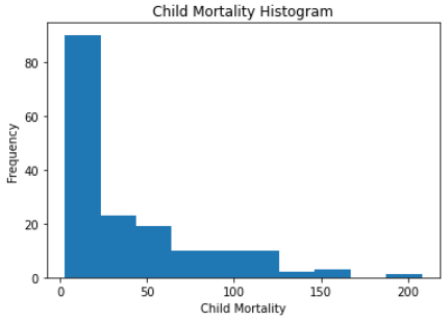
```
In [6]: statistics.pvariance(data['child_mort'])
Out[6]: 1618.951971371882
```

Знайдемо середньоквадратичне відхилення(наскільки сильно значення розкидані відносно середнього)

```
In [7]: statistics.pstdev(data['child_mort'])
Out[7]: 40.236202248371825
```

3. Візуалізуємо завантажені дані за допомогою гістограми

```
In [8]: plt.hist(data['child_mort'])
        plt.xlabel('Child Mortality')
        plt.ylabel('Frequency')
        plt.title('Child Mortality Histogram')
        plt.show()
```



4. Створюємо Series на основі даниих із минулого завдання

```
In [9]: child_mort = pd.Series(data['child_mort'])
        child_mort
Out[9]: 0       90.2
        1       16.6
        2       27.3
        3      119.0
        4       10.3
               ...
        163     17.1
        164     23.3
        165     56.3
        166     83.1
        167     83.1
        Name: child_mort, Length: 168, dtype: float64
```

Використання range()

```
In [10]: pd.Series(10, range(3))
Out[10]: 0    10
         1    10
         2    10
         dtype: int64
```

Звертання до елементу

```
In [11]: child_mort[3]
Out[11]: 119.0
```

Обчислення описових статистик

```
In [12]: child_mort.count()
Out[12]: 168

In [13]: child_mort.max()
Out[13]: 208.0

In [14]: child_mort.mean()
Out[14]: 38.53690476190476

In [15]: pd.Series(data['child_mort']).describe()
Out[15]: count    168.000000
         mean      38.536905
         std       40.356490
         min        2.600000
         25%        8.425000
         50%       19.500000
         75%       62.200000
         max      208.000000
         Name: child_mort, dtype: float64
```

Створення колекції з не типових індексів

```
In [16]: pd.Series([1, 2, 3], index=['index', 'index1', 'index2'])
Out[16]: index     1
         index1    2
         index2    3
         dtype: int64
```

Використання як інікалізатора словник

```
In [17]: d = {f'index{i+1}': i for i in range(10)}
         ser = pd.Series(d)
         ser
Out[17]: index1     0
         index2     1
         index3     2
         index4     3
         index5     4
         index6     5
         index7     6
         index8     7
         index9     8
         index10    9
         dtype: int64
```

Для Dataframe:

Створюємо dataframe використовуючи словник:

In [18]:
```
data = {
    'Subject': ['Mathematics', 'Physics', 'Chemistry', 'English'],
    'Grade1': [90, 85, 92, 88],
    'Grade2': [95, 88, 90, 92],
    'Grade3': [92, 90, 87, 95]
}
df = pd.DataFrame(data)
df
```

Out[18]:

	Subject	Grade1	Grade2	Grade3
0	Mathematics	90	95	92
1	Physics	85	88	90
2	Chemistry	92	90	87
3	English	88	92	95

Змінимо індекси:

In [19]:
```
df.index = [f'index{i+1}' for i in range(4)]
df
```

Out[19]:

	Subject	Grade1	Grade2	Grade3
index1	Mathematics	90	95	92
index2	Physics	85	88	90
index3	Chemistry	92	90	87
index4	English	88	92	95

Звернемося до стовпця `Subject`:

In [20]:
```
df['Subject']
```

Out[20]:
```
index1    Mathematics
index2        Physics
index3      Chemistry
index4        English
Name: Subject, dtype: object
```

Звернемося до значень першої оцінки Grade1:

In [21]:
```
df.Grade1
```

Out[21]:
```
index1    90
index2    85
index3    92
index4    88
Name: Grade1, dtype: int64
```

Виберемо рядки використовуючи loc та iloc:

```
In [22]: df.loc['index4']
Out[22]: Subject    English
         Grade1          88
         Grade2          92
         Grade3          95
         Name: index4, dtype: object

In [23]: df.iloc[3]
Out[23]: Subject    English
         Grade1          88
         Grade2          92
         Grade3          95
         Name: index4, dtype: object
```

Використаємо як індекс сегмент:

```
In [24]: df.loc['index2':'index4']
```

Out[24]:

	Subject	Grade1	Grade2	Grade3
index2	Physics	85	88	90
index3	Chemistry	92	90	87
index4	English	88	92	95

```
In [25]: df.iloc[1:4]
```

Out[25]:

	Subject	Grade1	Grade2	Grade3
index2	Physics	85	88	90
index3	Chemistry	92	90	87
index4	English	88	92	95

Для конкретних рядків використаємо список:

```
In [26]: df.loc[['index1','index4']]
```

Out[26]:

	Subject	Grade1	Grade2	Grade3
index1	Mathematics	90	95	92
index4	English	88	92	95

```
In [27]: df.iloc[[0,3]]
```

Out[27]:

	Subject	Grade1	Grade2	Grade3
index1	Mathematics	90	95	92
index4	English	88	92	95

Виберемо підмножину рядків та стовпців:

```
In [28]: df.loc[['index1','index4'], ['Grade1','Grade2']]
```

Out[28]:

	Grade1	Grade2
index1	90	95
index4	88	92

Видалимо колонку предмету для подальших дій:

```
In [29]: df = df.drop('Subject', axis=1)
         df
```

Out[29]:

	Grade1	Grade2	Grade3
index1	90	95	92
index2	85	88	90
index3	92	90	87
index4	88	92	95

Виберемо усі значення які більші за 90:

```
In [30]: df[df>90]
```

Out[30]:

	Grade1	Grade2	Grade3
index1	NaN	95.0	92.0
index2	NaN	NaN	NaN
index3	92.0	NaN	NaN
index4	NaN	92.0	95.0

Усі значення які більші за 85 але менші за 90:

```
In [31]: df[(df >= 85) & (df <= 90)]
```

Out[31]:

	Grade1	Grade2	Grade3
index1	90.0	NaN	NaN
index2	85.0	88.0	90.0
index3	NaN	90.0	87.0
index4	88.0	NaN	NaN

Виведення елементу по назві рядку та стовпця:

```
In [32]: df.at['index1', 'Grade1']
Out[32]: 90

In [33]: df.iat[2,2]
Out[33]: 87
```

Виведемо повну описову статистику:

```
In [34]: df.describe()
```

Out[34]:

	Grade1	Grade2	Grade3
count	4.000000	4.000000	4.000000
mean	88.750000	91.250000	91.000000
std	2.986079	2.986079	3.366502
min	85.000000	88.000000	87.000000
25%	87.250000	89.500000	89.250000
50%	89.000000	91.000000	91.000000
75%	90.500000	92.750000	92.750000
max	92.000000	95.000000	95.000000

Транспонуємо даний dataframe:

```
In [35]: df.T
```

Out[35]:

	index1	index2	index3	index4
Grade1	90	85	92	88
Grade2	95	88	90	92
Grade3	92	90	87	95

Виведемо описову статистику для транспонованого dataframe:

```
In [36]: df.T.describe()
```

Out[36]:

	index1	index2	index3	index4
count	3.000000	3.000000	3.000000	3.000000
mean	92.333333	87.666667	89.666667	91.666667
std	2.516611	2.516611	2.516611	3.511885
min	90.000000	85.000000	87.000000	88.000000
25%	91.000000	86.500000	88.500000	90.000000
50%	92.000000	88.000000	90.000000	92.000000
75%	93.500000	89.000000	91.000000	93.500000
max	95.000000	90.000000	92.000000	95.000000

Відсортуємо рядки dataframe за індексами:

```
In [37]: df.sort_index(ascending=False)
```

Out[37]:

	Grade1	Grade2	Grade3
index4	88	92	95
index3	92	90	87
index2	85	88	90
index1	90	95	92

та за стовпцями:

```
In [38]: df.sort_index(axis=1, ascending=False)
```

Out[38]:

	Grade3	Grade2	Grade1
index1	92	95	90
index2	90	88	85
index3	87	90	92
index4	95	92	88

Сортуємо dataframe за значеннями Grade1:

```
In [39]: df.sort_values(by='Grade1', ascending=False)
```

Out[39]:

	Grade1	Grade2	Grade3
index3	92	90	87
index1	90	95	92
index4	88	92	95
index2	85	88	90

5. 6. 7.

Прочитаємо, редагуємо та завантажимо набір даних катастрофи «Титаніка»:

```
In [40]: titanic = pd.read_csv('https://vincentarelbundock.github.io/Rdatasets/csv/carData/TitanicSurvival.csv')
```

```
In [40]: elbundock.github.io/Rdatasets/csv/carData/TitanicSurvival.csv').rename(columns={'Unnamed: 0': 'name'} )
```

8. Виведемо дані задопомогою методів head і tail:

```
In [41]: titanic.head()
```

Out[41]:

	name	survived	sex	age	passengerClass
0	Allen, Miss. Elisabeth Walton	yes	female	29.0000	1st
1	Allison, Master. Hudson Trevor	yes	male	0.9167	1st
2	Allison, Miss. Helen Loraine	no	female	2.0000	1st
3	Allison, Mr. Hudson Joshua Crei	no	male	30.0000	1st
4	Allison, Mrs. Hudson J C (Bessi	no	female	25.0000	1st

```
In [42]: titanic.tail()
```

Out[42]:

	name	survived	sex	age	passengerClass
1304	Zabour, Miss. Hileni	no	female	14.5	3rd
1305	Zabour, Miss. Thamine	no	female	NaN	3rd
1306	Zakarian, Mr. Mapriededer	no	male	26.5	3rd
1307	Zakarian, Mr. Ortin	no	male	27.0	3rd
1308	Zimmerman, Mr. Leo	no	male	29.0	3rd

9. Скоротимо 'passengerClass' до 'class':

```
In [43]: titanic.rename(columns={'passengerClass':'class'},inplace = True)
```

10. Визначимо наймолодшого пасажира що вижив:

```
In [44]: titanic[titanic['age']==titanic['age'].min()]
```

Out[44]:

	name	survived	sex	age	class
763	Dean, Miss. Elizabeth Gladys M	yes	female	0.1667	3rd

Найстаршого:

```
In [45]: titanic[titanic['age']==titanic['age'].max()]
```

Out[45]:

	name	survived	sex	age	class
14	Barkworth, Mr. Algernon Henry W	yes	male	80.0	1st

Середній вік тих хто вижив:

```
In [46]: titanic['age'].mean()
Out[46]: 29.881134512434034
```

Відсортуємо всіх жінок з кают 1-го класу:

```
In [47]: sub_class = titanic[(titanic['class']=='1st') & (titanic['sex']=='female')].sort_values(by='age', ascer
```

```
In [47]: titanic[(titanic['class']=='1st') & (titanic['sex']=='female')].sort_values(by='age', ascending=False)
```

Наймолодша жінка яка вижила із першого класу:

```
In [48]: sub_class.min()

Out[48]: name        Allen, Miss. Elisabeth Walton
         survived                               no
         sex                                female
         age                                   2.0
         class                                 1st
         dtype: object
```

Найстарша жінка яка вижила із першого класу:

```
In [49]: sub_class.max()

Out[49]: name        Young, Miss. Marie Grice
         survived                         yes
         sex                           female
         age                             76.0
         class                            1st
         dtype: object
```

Загальна кількість жінок що вижили:

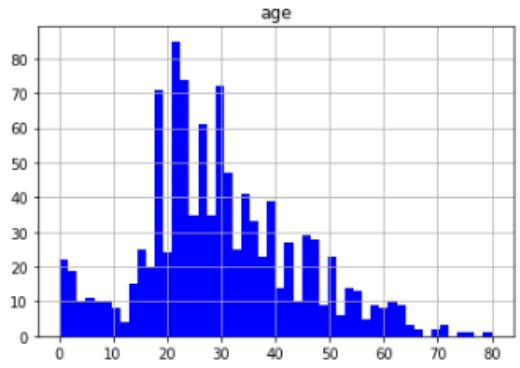
```
In [50]: from collections import Counter
         print(f"Total amount of survived women: {dict(Counter(sub_class['survived']))['yes']}")

         Total amount of survived women: 139
```

11. Побудуємо гістограму віку пасажирів:

```
In [51]: titanic.hist(color='blue', bins=50)

Out[51]: array([[<AxesSubplot:title={'center':'age'}>]], dtype=object)
```



Висновок

Виконавши дану лабораторну я ознайомився із такими бібліотеками як pandas та matplotlib. Були досліджено загальну інформацію про країни, а саме смерть дітей (віком до 5 років) на 1000 народжень. Для цих даних було знайдено: математичне сподівання, медіану, моду, дисперсію, середньоквадратичне відхилення. Проведена візуалізація даних та пророблені усі пункти над колекціями Series і DataFrame бібліотеки pandas. Також був проведений аналіз набору даних катастрофи лайнера “Титаніка”.