Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут”
Кафедра АСОІУ

ЗВІТ
про виконання лабораторної роботи №2
з дисципліни
“ Аналіз даних в інформаційно-управляючих системах”

СТВОРЕННЯ ВІ РІШЕННЯ

Виконав Студент
2 курсу групи ІП-11
Панченко Сергій

Київ 2023

ОПИСОВА СТАТИСТИКА

Мета роботи: ознайомитись з методикою первинної обробки статистичних даних; проаналізувати вплив способу представлення даних на їх інформативність.

Перелік корисних джерел

RStudio

+ Довідкові матеріали для початку роботи з R
+ Data visualization with ggplot2: cheatsheet
+ Гнатюк В. Вступ до R на прикладах
+ Кабаков Р.И. R в действии. Анализ и визуализация данных в программе R
+ Data Mining for Business Analytics

Python

+ Довідкові матеріали для початку роботи з Python
+ Учебник по Python
+ Подборка блокнотов по командам IPython
+ Інтерактивний міні-гайд по візуалізації даних на Python
+ Элбон Машинное обучение с использованием Python 2019
+ Python NumPy beginners
+ Campesato Pandas Basics
+ Пособие по Matplotlib
+ Первичный анализ данных с Pandas
+ Python Data Science Handbook

Запитання для самоперевірки

1. Чим відрізняються генеральна та вибіркова сукупності?
2. Які бувають способи відбору даних?
3. Як побудувати статистичний розподіл вибірки?
4. Що відносять до числових характеристик вибірки?
5. Як визначається мода та медіана?
6. Які бувають способи графічного зображення статистичних розподілів?
7. Що таке емпірична функція розподілу та які її властивості?

ЗАВДАННЯ

Скачати потрібні дані.

Завдання для самоперевірки

Ознайомитися з:

- Підключення бібліотек
- Можливі формати вхідних даних
- Перетворення форматів
- Отримання інформації про структуру даних
- Перетворення датафреймів
    - Додавання, видалення ознак (стовпчиків)
    - Перевірка та перетворення типу даних
- Групування даних
    - Сортування

Фільтрація
Виділення підмножини
Об'єднання кількох датафреймів в один
Обчислення числових характеристик
Застосування функцій до елемента, стовпчика, рядка
Приклад R
Поглиблено про графічне представлення інформації
кругові діаграми
діаграми розсіювання
діаграми розмаху
Приклад R

Скачати дані із файлу Data1.csv

1. дослідити їх структуру
2. вивести перші 5 рядків
3. вивести останні 6 рядків
4. видалити стовпчик з абревіатурами
5. додати стовпчик з повним GDP, пропуски замінити нулями
6. вивести все summary
7. побудувати діаграму розмаху для GDP per capita
8. побудувати графік залежності High-technology exports від GDP

Приклад виконання RStudio, Python

Основне завдання

Скачати дані із файлу Data2.csv

1. Записати дані у data frame
2. Дослідити структуру даних
3. Виправити помилки в даних
4. Побудувати діаграми розмаху та гістограми
5. Додати стовпчик із щільністю населення

Додаткове завдання

Відповісти на питання (файл Data2.csv):

1. Чи є пропущені значення? Якщо є, замінити середніми
2. Яка країна має найбільший ВВП на людину (GDP per capita)? Яка має найменшу площу?
3. В якому регіоні середня площа країни найбільша?
4. З яким населенням найчастіше зустрічаються країни у світі? У Європі?
5. Чи співпадає в якомусь регіоні середнє та медіана ВВП?
6. Вивести топ 5 країн та 5 останніх країн по ВВП та кількості CO2 на душу населення.

Завантажити вхідні дані в середовище, що буде використовуватись для їх обробки (розрахунки можна проводити з використанням R-Studio, Python, в пакетах STATISTICA, MathCad, MathLab, Excel тощо). Виконати необхідні розрахунки та оформити звіт.

В звіт включити основне та додаткове завдання (самі завдання, числові та графічні відповіді на них, а також висновки по виконаному дослідженню; файл з кодом

окремо).
Відповіді на теоретичні питання включати в звіт не потрібно - це просто підказка для вас, з чим треба розібратися до того, як виконувати роботу.
Загальний висновок по роботі у вигляді "розібрався, навчився" не потрібен, залишайте лише ваші висновки по проведеному дослідженню (наприклад, висновок щодо прийняття або відхилення гіпотези, яку ви перевіряли).

Код:
1) Записати дані у дата фрейм:

	Country Name	Region	GDP per capita	Populatiion	CO2 emission	Area
0	Afghanistan	South Asia	561,7787463	3.465603e+07	9809,225	652860
1	Albania	Europe & Central Asia	4124,98239	2.876101e+06	5716,853	28750
2	Algeria	Middle East & North Africa	3916,881571	4.060605e+07	145400,217	2381740
3	American Samoa	East Asia & Pacific	11834,74523	5.559900e+04	NaN	200
4	Andorra	Europe & Central Asia	36988,62203	7.728100e+04	462,042	470
5	Angola	Sub-Saharan Africa	3308,700233	2.881346e+07	34763,16	1246700
6	Antigua and Barbuda	Latin America & Caribbean	14462,17628	1.009630e+05	531,715	440
7	Argentina	Latin America & Caribbean	12440,32098	4.384743e+07	204024,546	2780400
8	Armenia	Europe & Central Asia	3614,688357	2.924816e+06	5529,836	29740

2) Дослідити структуру даних:

Бачимо, що датафрейм має NAN-значення, пусті клітинки. Також варто помітити, що «Populatiion» написана з помилкою, треба виправити на «Population». Також серед значень є від'ємні.

3) Виправити помилки в даних:
Додаткове 1:

Замінимо пропущені значення середніми

```
for column_name in dataset.columns[2:]:
  replace_comma_with_dots(dataset, column_name)
  convert_column_to_float(dataset, column_name)
  replace_nan_with_mean(dataset, column_name)
  convert_float_with_positive(dataset, column_name)
def replace_comma_with_dots(dataset: pd.DataFrame, column_name: str) -> None:
   dataset[column_name] = dataset[column_name].astype(str)
   dataset[column_name] = dataset[column_name].str.replace(',', '.')


def convert_column_to_float(dataset: pd.DataFrame, column_name: str) -> None:
   dataset[column_name] = dataset[column_name].astype(float)


def replace_nan_with_mean(dataset: pd.DataFrame, column_name: str):
   mean_value = dataset[column_name].mean()
   dataset[column_name].fillna(value=mean_value, inplace=True)

def convert_float_with_positive(dataset: pd.DataFrame, column_name: str):
   dataset[column_name] = dataset[column_name].abs()
```

Додаткове 2:

```
# 2 Яка країна має найбільший ВВП на людину (GDP per capita)? Яка має найменшу
```

```
площу?
df_max_gdp = dataset.nlargest(1, ['GDP per capita'])
max_gdp = df_max_gdp['GDP per capita'].values[0]
country_max_gdp = df_max_gdp['Country Name'].values[0]
print(f'{bcolors.HEADER}Max GDP:\n{bcolors.ENDC}'
      f'\t{bcolors.OKBLUE}Country: '
      f'{bcolors.OKGREEN}{country_max_gdp}\n'
      f'\t{bcolors.OKBLUE}GDP: {bcolors.OKGREEN}{max_gdp}')
```

Max GDP:
    Country: Luxembourg
    GDP: 100738.6842

Додаткове 3:

```
# 3 В якому регіоні середня площа країни найбільша?
df_group_by_region = dataset.groupby('Region')
df_region_area_sum = df_group_by_region.sum(numeric_only=True)[['Area']]
df_region_count = df_group_by_region.count()[['Area']]
df_average_area = df_region_area_sum / df_region_count
df_max_average_area = df_average_area.nlargest(1, ['Area'])
max_average_area_region = df_max_average_area.axes[0][0]
max_average_area = df_max_average_area['Area'].values[0]
print(f'{bcolors.HEADER}Max average area:{bcolors.ENDC}\n'
      f'\t{bcolors.OKBLUE}Region: '
      f'{bcolors.OKGREEN}{max_average_area_region}\n'
      f'\t{bcolors.OKBLUE}Value: '
      f'{bcolors.OKGREEN}{max_average_area}')
```

Min Area:
    Country: Monaco
    Area: 2.0
Max average area:
    Region: North America
    Value: 6605410.0

5. Додати стовпчик із щільністю населення та додаткове 4:

```
# 4 знайдіть країну з найбільшою щільністю населення, у світі та в Європі
dataset['Population'] = dataset['Populatiion']
dataset.drop(['Populatiion'], axis=1)
dataset['Density'] = dataset['Population'] / dataset['Area']
df_max_density_world = dataset.nlargest(1, ['Density'])
max_density_world = df_max_density_world['Density'].values[0]
max_density_world_country = df_max_density_world['Country Name'].values[0]
df_europe = dataset[dataset['Region'] == 'Europe & Central Asia']
df_europe_max_density = df_europe.nlargest(1, ['Density'])
max_density_europe_country = df_europe_max_density['Country Name'].values[0]
max_density_europe = df_europe_max_density['Density'].values[0]
print(f'{bcolors.HEADER}Max density:{bcolors.ENDC}\n'
      f'\t{bcolors.OKBLUE}World: '
      f'{bcolors.OKGREEN}{max_density_world_country}\n'
      f'\t{bcolors.OKBLUE}World Value: '
      f'{bcolors.OKGREEN}{max_density_world}\n'
```

```
    f'\t{bcolors.OKBLUE}Europe Country: '
    f'{bcolors.OKGREEN}{max_density_europe_country}\n'
    f'\t{bcolors.OKBLUE}Europe Value: '
    f'{bcolors.OKGREEN}{max_density_europe}')
```

Max density:

World: Macao SAR, China
World Value: 20203.531353135313
Europe Country: Monaco
Europe Value: 19249.5

Додаткове 5:

```
# 5 Чи співпадає в якомусь регіоні середнє та медіана ВВП?
df_region_gdp_mean = pd.DataFrame()
df_region_gdp_average = pd.DataFrame()
df_region_gdp_mean['Mean'] = df_group_by_region.mean(numeric_only=True)[['GDP per
capita']]
df_region_gdp_average['Average'] = df_group_by_region.median(numeric_only=True)
[['GDP per capita']]
df_region_mean_average = pd.concat([df_region_gdp_mean, df_region_gdp_average],
axis=1)
df_region_mean_average['Difference'] = df_region_mean_average['Mean'] -
df_region_mean_average['Average']
df_region_mean_average['Difference'] = df_region_mean_average['Difference'].abs()
df_smallest_mean_average_difference = df_region_mean_average.nsmallest(1,
['Difference'])
mean_average_info = bcolors.HEADER + 'Mean-Median Equality:\n' + bcolors.ENDC
mean_average_info += f'\t{bcolors.OKBLUE}Region: {bcolors.OKGREEN}
{df_smallest_mean_average_difference.axes[0][0]}'
for column_name in ['Mean', 'Average', 'Difference']:
    mean_average_info += f'\n\t{bcolors.OKBLUE}{column_name}: ' \
                         f'{bcolors.OKGREEN}' \
                         f'{df_smallest_mean_average_difference[column_name].values[0]}' \
                         f'{bcolors.ENDC}'
print(mean_average_info)
```

Mean-Median Equality:

Region: Latin America & Caribbean
Mean: 10468.495457604762
Average: 10833.201075
Difference: 364.70561739523873

Додаткове 6:

```
# 6 Вивести топ 5 країн та 5 останніх країн по ВВП та кількості CO2 на душу
населення.
dataset_gdp_desc = dataset.sort_values('GDP per capita', ascending=False)
dataset_gdp_asc = dataset.sort_values('GDP per capita', ascending=True)
dataset_co2_desc = dataset.sort_values('CO2 emission', ascending=False)
dataset_co2_asc = dataset.sort_values('CO2 emission', ascending=True)
print(f'{bcolors.HEADER}GDP Top 5:{bcolors.ENDC}\n',
dataset_gdp_desc.head(5).to_string())
```

```
print(f'{bcolors.HEADER}GDP Bottom 5:{bcolors.ENDC}\n',
dataset_gdp_asc.head(5).to_string())
print(f'{bcolors.HEADER}CO2 Top 5:{bcolors.ENDC}\n',
dataset_co2_desc.head(5).to_string())
print(f'{bcolors.HEADER}CO2 Bottom 5:{bcolors.ENDC}\n',
dataset_co2_asc.head(5).to_string())
```

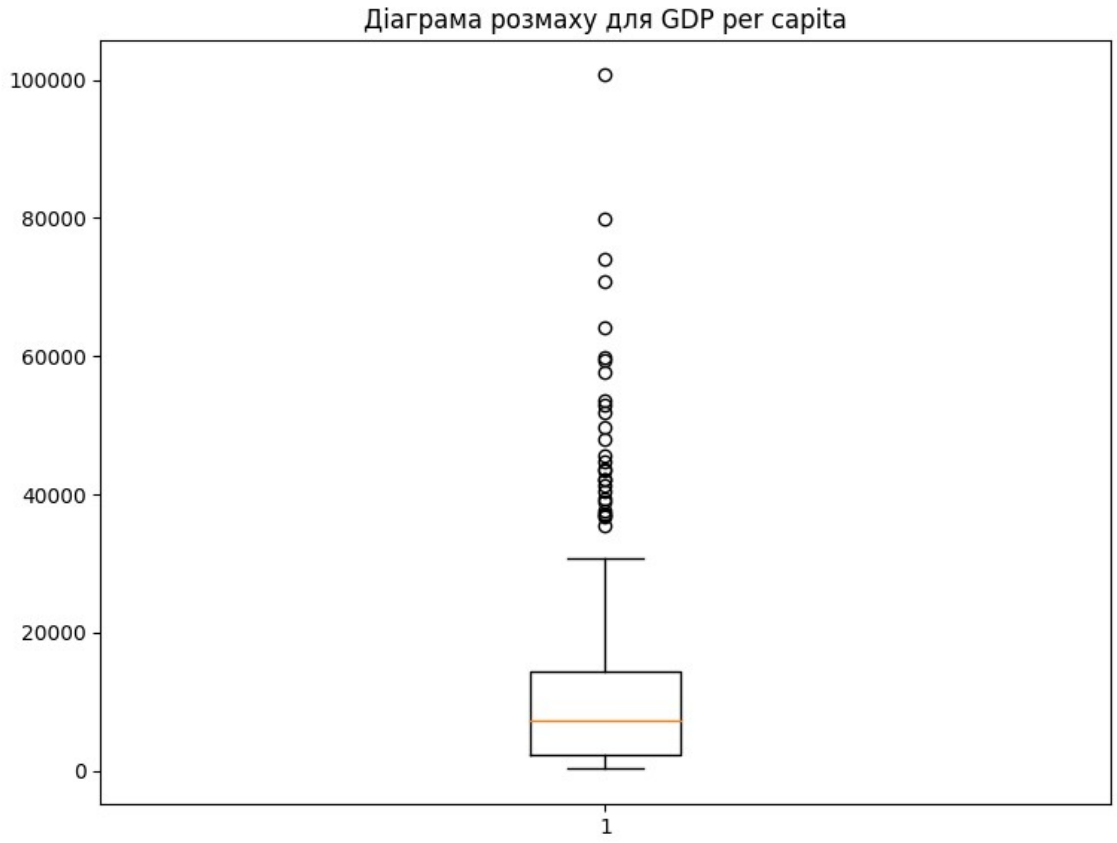
```
GDP Top 5:
       Country Name  GDP per capita
115      Luxembourg    100738.68420
188     Switzerland     79887.51824
116  Macao SAR, China     74017.18471
146          Norway     70868.12250
92          Ireland     64175.43824
GDP Bottom 5:
            Country Name  GDP per capita
31               Burundi     285.727442
119               Malawi     300.307665
134            Mozambique     382.069330
37   Central African Republic     382.213174
118            Madagascar     401.742270
CO2 Top 5:
         Country Name  CO2 emission
41              China  1.029193e+07
206     United States  5.254279e+06
88              India  2.238377e+06
160  Russian Federation  1.705346e+06
97              Japan  1.214048e+06
CO2 Bottom 5:
       Country Name  CO2 emission
201          Tuvalu       11.001
113    Liechtenstein       44.004
137            Nauru       47.671
101         Kiribati       62.339
124  Marshall Islands      102.676
```

6. Побудувати діаграми розмаху та гістограми

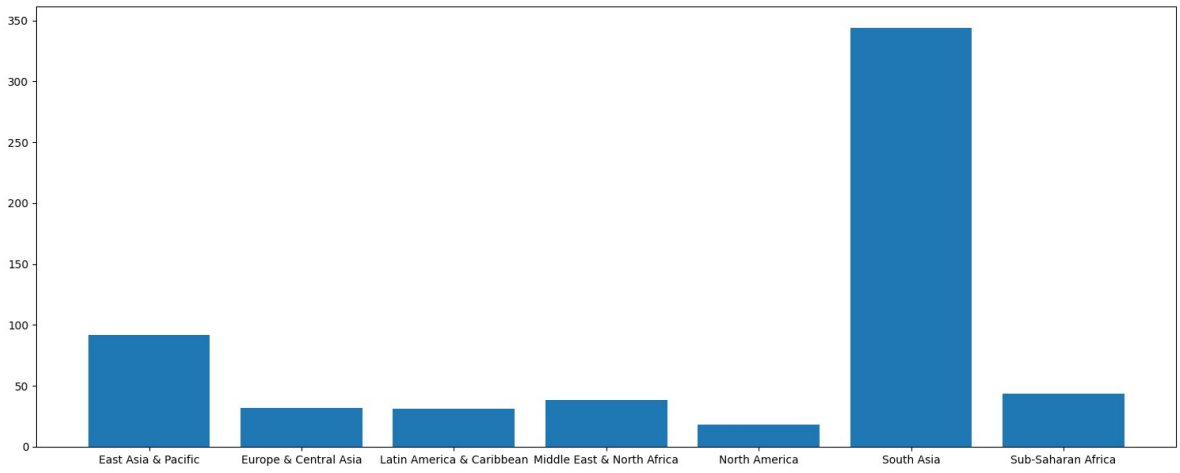
```
#Plot average density by regions
df_group_by_region = dataset.groupby('Region')
df_region_area_sum = df_group_by_region.sum(numeric_only=True)[['Area']]
df_region_population_sum = df_group_by_region.sum(numeric_only=True)[['Population']]
df_region_density = df_region_population_sum['Population'] / df_region_area_sum['Area']
df_region_density: pd.DataFrame = df_region_density.to_frame()
df_region_density.columns = ['Density']
print(df_region_density, df_region_density.axes)



plt.bar(df_region_density.index, df_region_density['Density'])
```

```
boxplot(dataset, 'GDP per capita')
plt.show()
```

Побудувати діаграми розмаху GDP per capita:



Гістограма щільності населення по регіонах



Висновок:
Відредагував дані, знайшов середні значення, медіани стопчиків датафрейму. Код написаний на пайтоні, графіки наведені.