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

ЗВІТ
про виконання лабораторної роботи №4
з дисципліни
“ Аналіз даних в інформаційно-управляючих системах”

СТВОРЕННЯ ВІ РІШЕННЯ

Виконав Студент
2 курсу групи ІП-11
Панченко Сергій

Київ 2023

ВИВІДНА СТАТИСТИКА

Мета роботи: ознайомитись з

- методами визначення точкових оцінок параметрів розподілу; дослідити, що впливає на якість точкових оцінок;
- методикою визначення інтервальних оцінок параметрів розподілу; дослідити, що впливає на якість інтервальних оцінок;
- методами перевірки статистичних гіпотез про вигляд закону розподілу; дослідити, що впливає на ширину критичної області.

Перелік корисних джерел

RStudio

- Довідкові матеріали для початку роботи з R
- Data visualization with ggplot2: cheatsheet
- 5 видеороликов о пакете dplyr языка R
- 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
- Первичный анализ данных с Pandas
- Інтерактивний міні-гайд по візуалізації даних на Python
- Визуальный анализ данных с Python
- Элбон Машинное обучение с использованием Python 2019
- Python NumPy beginners
- Campesato Pandas Basics
- Cheat sheets for Python
- Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython
- The Pandas DataFrame: Make Working With Data Delightful
- Data Visualization: A Practical Introduction
- Scientific Computing in Python: Introduction to NumPy and Matplotlib
- Interactive Data Visualization with Python
- Gayathri R. A Python Data Analyst's Toolkit
- Подборка статей о визуализации данных (англ.)

Запитання для самоперевірки

1. Назвіть основні вимоги до статистичних оцінок.
2. Як визначається проста середньоарифметична вибірки?
3. Як визначається вибіркова середня або зважена середньоарифметична вибірки

та які її властивості?

4. Як визначається степенева середня вибірки?
5. Як визначається вибіркова дисперсія та вибіркове середньоквадратичне відхилення? У яких випадках потрібна виправлена вибіркова дисперсія і як її визначити?
6. Для чого потрібні початкові та центральні моменти вибірки та як їх визначити?
7. В чому різниця між точковими та інтервальними оцінками параметрів ЗРВ?
8. Що таке надійність (довірча імовірність) оцінки параметра ЗРВ?
9. Чим визначається та для чого використовується довірчий інтервал?
10. Якою повинна бути підходяща статистика для визначення інтервальної оцінки?
11. Порядок визначення інтервальних оцінок математичного сподівання.
12. Порядок визначення інтервальних оцінок дисперсії.
13. Що називають статистичною гіпотезою?
14. Перерахуйте різновиди статистичні гіпотез.
15. Які бувають похибки перевірки гіпотез? Чим вони відрізняються?
16. Що таке критична область і від чого залежить її вигляд?
17. Що впливає на ширину критичної області?
18. Які критерії узгодження можна використовувати для перевірки гіпотез про вигляд розподілу?
19. Які критерії узгодження можна використовувати для перевірки гіпотез про параметри розподілу?
20. Перерахуйте обмеження, які накладаються на використання кожного з критеріїв.

ЗАВДАННЯ

Скачати потрібні дані.

Завдання для самоперевірки (не оцінюється, в звіт не включати)

Скачати дані по продажам авокадо в Америці Data3a.csv, Data3b.csv. Подивитись, проаналізувати структуру.

1. Об'єднати в один файл.
2. Створити стовпчик з прибутком.
3. Знайти загальний прибуток по органічному та неорганічному авокадо.
4. Який рік був найбільш успішним?
5. Побудувати 3 графіки залежностей середньої ціни від кількості упаковок різних розмірів. Чи є очевидна залежність?
6. Чи є викиди в обсягах продаж?
7. Ознайомитись з функцією pie(). Побудувати кругову діаграму по кількості проданих авокадо видів 4046, 4225, 4770 у 2016 році.
8. В якому штаті середня ціна за весь час була мінімальною, а в якому максимальною?

9. Які регіони схожі по продажам авокадо? Поясніть свою відповідь.

Приклад виконання RStudio, Python

Основне завдання

Скачати дані файлу Data2.csv.

1. Подивитись, проаналізувати структуру
2. Вказати, чи є параметри, що розподілені за нормальним законом
3. Перевірити гіпотезу про рівність середнього і медіани для одного з параметрів
4. Вказати, в якому регіоні розподіл викидів CO2 найбільш близький до нормального
5. Побудувати кругову діаграму населення по регіонам

Додаткове завдання

Особливості роботи з картинками та картами, різновиди форматів для збереження карт.

Генерація досліджуваних об'єктів.

Робота з растровим зображенням.

Робота з картами, що задані контуром (полігоном).

Інтерактивні карти.

Приклад 1 R

Робота з просторовим розподілом об'єктів (польові дослідження з мітками на карті).

Побудова карт щільностей.

Приклад 2 R

Робота з shape-файлами.

Побудова картограм.

Приклад 3 R

Скачати потрібні дані

1. Завантажити карту України Ukraine.jpg
2. Розмістити бульбашки, що відповідають їх населенню, на довільних 5 містах (статистику взяти в інтернеті)
3. Знайти найбільшу відстань між містами в пікселях та кілометрах

1. Завантажити файл з даними про конфлікти conflicts.csv
2. Побудувати просторовий розподіл конфліктів у Європі та Україні
3. Для Європи побудувати ізокліни для розподілу конфліктів
4. Для України:
    1. визначити, який регіон представлено на реальній карті;
    2. класифікувати точки по рокам;

3. побудувати розподіл щільності.

1. Завантажити shape-файл с областями України.
2. Побудувати картограми для прибутку населення на 1 особу і ВВП по регіонам за 2016 рік.
3. По даним за 2006-2015 роки для кожного регіону розрахувати коефіцієнт кореляції між прибутком населення на 1 особу та ВВП. Відобразити на картограмі.

Основне завдання

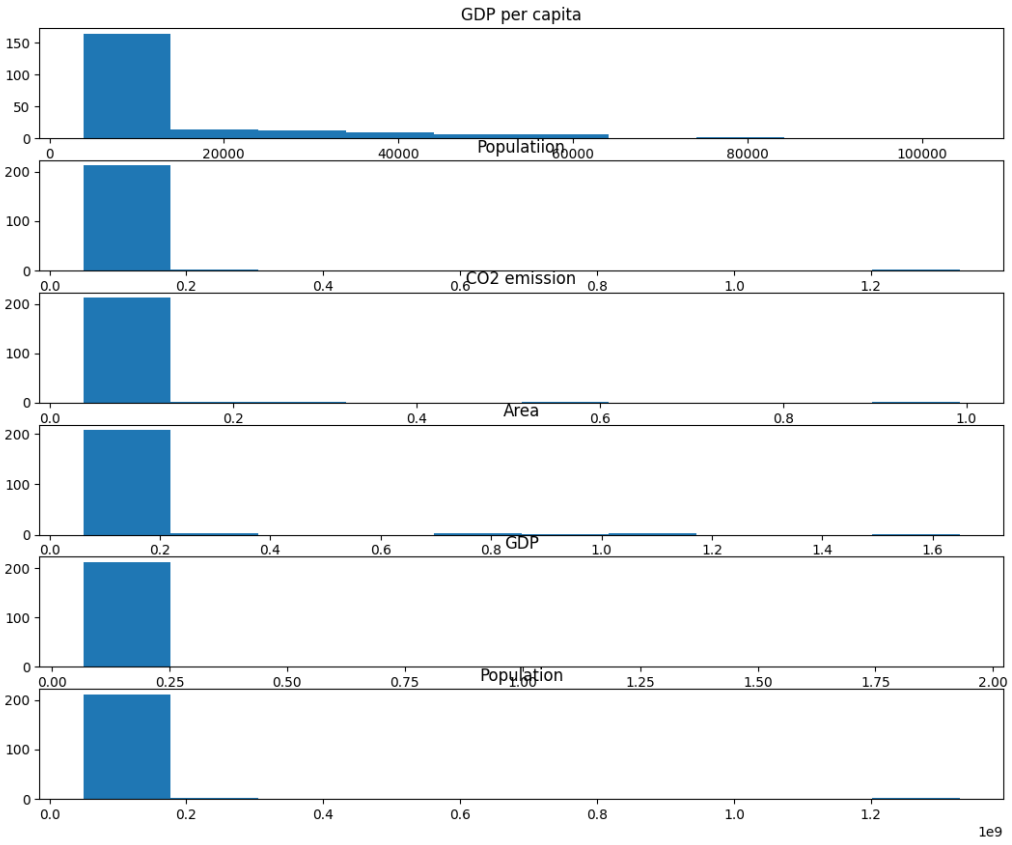
Вказати, чи є параметри, що розподілені за нормальним законом

Застосуємо тест Андерсона

```
def print_normality_check_result(sig_lev: float, crit_val: float, res_stat: float):
    res_val_msg = colorize(f'{res_stat}', bcolors.OKCYAN)
    crit_val_msg = colorize(f'{crit_val}', bcolors.OKCYAN)
    sig_val_msg = colorize(f'{sig_lev}', bcolors.OKCYAN)
    prob_normal = colorize('\t\tProbably Normal:', bcolors.OKGREEN)
    prob_not_normal = colorize('\t\tProbably Not Normal:', bcolors.FAIL)
    result_msg = colorize(f'\t\t\tExpected Value', bcolors.WARNING)
    critical_msg = colorize('\t\t\tCritical Value', bcolors.WARNING)
    significance_level_msg = colorize('\t\t\tSignificance Level', bcolors.WARNING)
    if res_stat < crit_val:
        print(f'{prob_normal}')
    else:
        print(f'{prob_not_normal}')
    print(result_msg, res_val_msg)
    print(critical_msg, crit_val_msg)
    print(significance_level_msg, sig_val_msg)


def check_normally_distributed(dataset: pd.DataFrame, column_name: str) -> None:
    result = stats.anderson(dataset[column_name])
    for i in range(len(result.critical_values)):
        sig_lev, crit_val = result.significance_level[i], result.critical_values[i]
        print_normality_check_result(sig_lev, crit_val, result.statistic)


def dataframe_col_to_interval_discrete(dataset: pd.DataFrame, column_name: str, step:
float = 0) -> list[float]:
    nums_count_freq =
myst.nums_count_frequency_tuples(dataset[column_name].values.tolist())
    intervals = myst.interval_sequence(nums_count_freq, step=step)
```

```
    discrete = myst.discrete_sequence_from_interval_count_frequency(intervals)
    hist_data = [row[0] for row in discrete for _ in range(row[1])]
    return hist_data


def check_columns_normally_distributed(dataset: pd.DataFrame, columns: list[str]):
    print(colorize(f'Check for normal distribution:', bcolors.HEADER))
    fig, axes = plt.subplots(len(columns), 1)

    for i, col in enumerate(columns):
        print(colorize(f'\t{col}:', bcolors.OKBLUE))
        check_normally_distributed(dataset, col)
        hist_data = dataframe_col_to_interval_discrete(dataset, col)
        axes[i].hist(hist_data)
        axes[i].set_title(col)

# ['GDP per capita', 'Populatiion', 'CO2 emission', 'Area', 'GDP', 'Population']
# 2 Вказати, чи є параметри, що розподілені за нормальним законом
check_columns_normally_distributed(dataset, numeric_cols)
```

Отримаємо такі гістограми та побачимо, що жожна з величин не розподілена за нормальним законом:



Check for normal distribution:

GDP per capita:

Probably Not Normal:

Expected Value 19.157195602761817

Critical Value 0.566

Significance Level 15.0

Probably Not Normal:

Expected Value 19.157195602761817

Critical Value 0.644

Significance Level 10.0

Probably Not Normal:

Expected Value 19.157195602761817

Critical Value 0.773

Significance Level 5.0

Probably Not Normal:

Expected Value 19.157195602761817

Critical Value 0.902

Significance Level 2.5

Probably Not Normal:

Expected Value 19.157195602761817

Critical Value 1.073

Significance Level 1.0

Populatiion:

Probably Not Normal:

Expected Value 54.013696488519145

Critical Value 0.566

Significance Level 15.0

Probably Not Normal:

Expected Value 54.013696488519145

Critical Value 0.644

Significance Level 10.0

Probably Not Normal:

Expected Value 54.013696488519145

Critical Value 0.773

Significance Level 5.0

Probably Not Normal:

Expected Value 54.013696488519145

Critical Value 0.902

Significance Level 2.5

Probably Not Normal:
Expected Value 54.013696488519145
Critical Value 1.073
Significance Level 1.0

CO2 emission:
Probably Not Normal:
Expected Value 60.53652028908277
Critical Value 0.566
Significance Level 15.0
Probably Not Normal:
Expected Value 60.53652028908277
Critical Value 0.644
Significance Level 10.0
Probably Not Normal:
Expected Value 60.53652028908277
Critical Value 0.773
Significance Level 5.0
Probably Not Normal:
Expected Value 60.53652028908277
Critical Value 0.902
Significance Level 2.5
Probably Not Normal:
Expected Value 60.53652028908277
Critical Value 1.073
Significance Level 1.0

Area:
Probably Not Normal:
Expected Value 47.71107909389252
Critical Value 0.566
Significance Level 15.0
Probably Not Normal:
Expected Value 47.71107909389252
Critical Value 0.644
Significance Level 10.0
Probably Not Normal:
Expected Value 47.71107909389252
Critical Value 0.773
Significance Level 5.0
Probably Not Normal:
Expected Value 47.71107909389252

Critical Value 0.902
Significance Level 2.5
Probably Not Normal:
Expected Value 47.71107909389252
Critical Value 1.073
Significance Level 1.0
GDP:
Probably Not Normal:
Expected Value 57.937022606921744
Critical Value 0.566
Significance Level 15.0
Probably Not Normal:
Expected Value 57.937022606921744
Critical Value 0.644
Significance Level 10.0
Probably Not Normal:
Expected Value 57.937022606921744
Critical Value 0.773
Significance Level 5.0
Probably Not Normal:
Expected Value 57.937022606921744
Critical Value 0.902
Significance Level 2.5
Probably Not Normal:
Expected Value 57.937022606921744
Critical Value 1.073
Significance Level 1.0
Population:
Probably Not Normal:
Expected Value 54.013696488519145
Critical Value 0.566
Significance Level 15.0
Probably Not Normal:
Expected Value 54.013696488519145
Critical Value 0.644
Significance Level 10.0
Probably Not Normal:
Expected Value 54.013696488519145
Critical Value 0.773
Significance Level 5.0

Probably Not Normal:

Expected Value 54.013696488519145

Critical Value 0.902

Significance Level 2.5

Probably Not Normal:

Expected Value 54.013696488519145

Critical Value 1.073

Significance Level 1.0

Перевірити гіпотезу про рівність середнього і медіани для одного з параметрів
Застосуємо одновибірковий Т-критерій Стьюдента

```
def check_mean_median_columns(dataset: pd.DataFrame, columns: list[str]):
    print(colorize(f'Mean-median:', bcolors.HEADER))
    hypothesis_accepted = colorize(f'\t\tAccepted:', bcolors.OKGREEN)
    hypothesis_rejected = colorize(f'\t\tRejected:', bcolors.FAIL)
    for col in columns:
        mean = dataset[col].mean()
        median = dataset[col].median()
        t_statistic, p_value = stats.ttest_1samp(a=dataset[col], popmean=median)
        mean_msg = colorize('\t\t\tMean: ', bcolors.WARNING)
        median_msg = colorize('\t\t\tMedian: ', bcolors.WARNING)
        mean_value = colorize(f'{mean}', bcolors.OKCYAN)
        median_value = colorize(f'{median}', bcolors.OKCYAN)
        colored_col = colorize(f'{col}:', bcolors.OKBLUE)
        print(f'\t{colored_col}')
        if p_value < 0.01:
            print(hypothesis_rejected)
        else:
            print(hypothesis_accepted)
        print(f'{mean_msg}{mean_value}')
        print(f'{median_msg}{median_value}')
```

Mean-median:

GDP per capita:

Rejected:

Mean: 13436.789145600322

Median: 7179.340661

Populatiion:

Rejected:

Mean: 34322559.875

Median: 6293253.0

CO2 emission:

Rejected:

Mean: 165114.1163365854

Median: 11562.051

Area:

Rejected:

Mean: 618844.1023041474

Median: 93030.0

GDP:

Rejected:

Mean: 353584564390.77295

Median: 24078931931.86415

Population:

Rejected:

Mean: 34322559.875

Median: 6293253.0

Вказати, в якому регіоні розподіл викидів CO2 найбільш близький до нормального

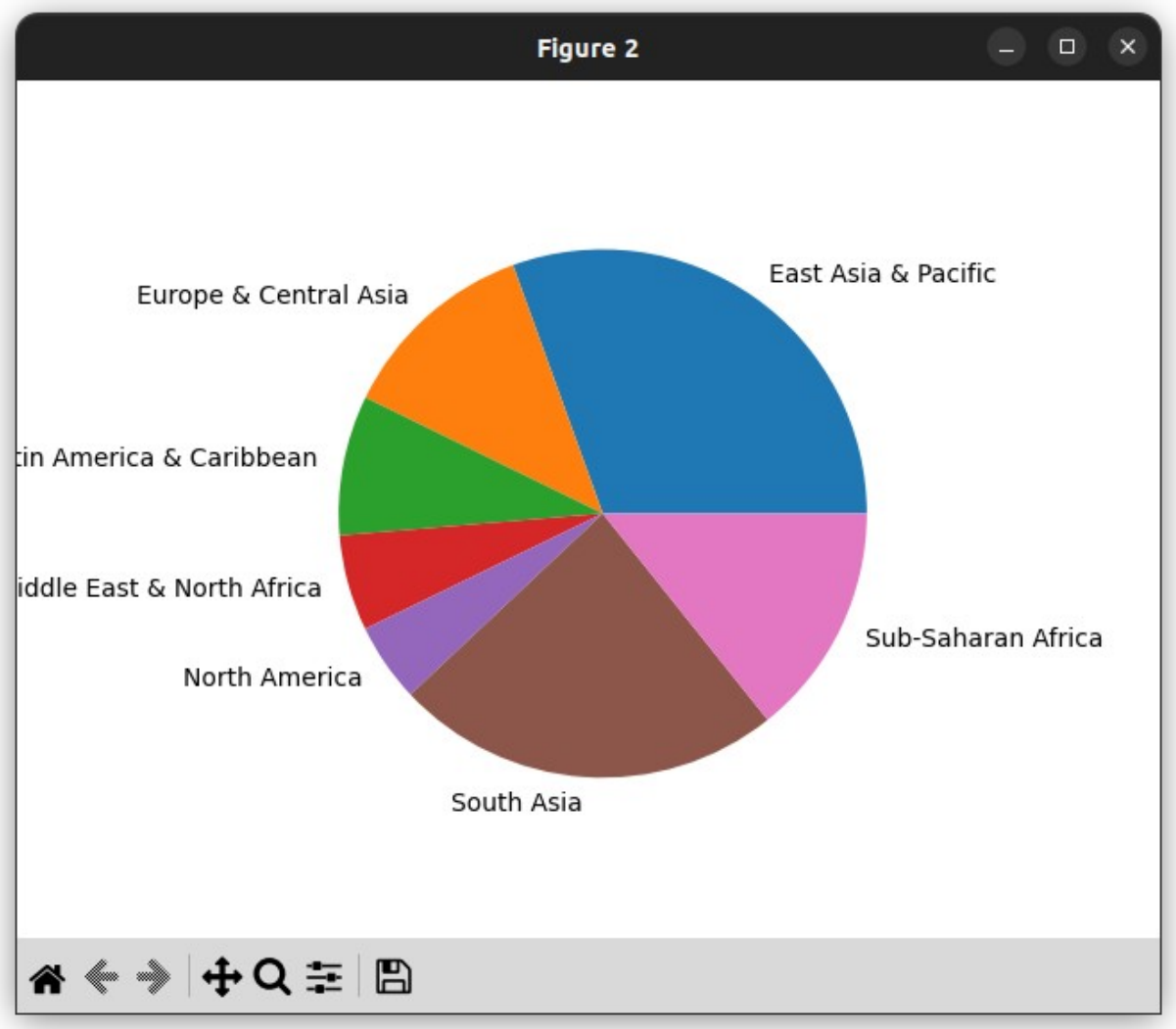
```
def group_by_column_normally_distibuted(dataset: pd.DataFrame, group_column: str,
normally_checked: str):
    print(colorize(f'Normally checked {normally_checked} by {group_column}:',
bcolors.HEADER))
    diffs = []
    groups = list(set(dataset[group_column].values.tolist()))
    for val in groups:
        data = dataset[dataset[group_column] == val]
        result = stats.anderson(data[normally_checked])
        expected = result.statistic
        sig_lev, crit_val = result.significance_level[0], result.critical_values[0]
        print(colorize(f'\t{val}:', bcolors.OKBLUE))
        print_normality_check_result(sig_lev, crit_val, expected)
        difference_msg = colorize('\t\t\tDifference: ', bcolors.WARNING)
        difference = abs(expected - crit_val)
        diff_val_msg = colorize(f'{difference}', bcolors.OKCYAN)
        diffs.append(difference)
        print(difference_msg, diff_val_msg)
    min_diff = min(diffs)
    index = diffs.index(min_diff)
    min_diff_msg = colorize('\tMin Difference: ', bcolors.OKBLUE)
    min_diff_key_val = groups[index]
    min_diff_key = colorize(f'{min_diff_key_val}', bcolors.WARNING)
    print(min_diff_msg, min_diff_key)
# 4 Вказати, в якому регіоні розподіл викидів CO2 найбільш близький до
нормального
group_by_column_normally_distibuted(dataset, 'Region', 'CO2 emission')
```

Normally checked CO2 emission by Region:

Middle East & North Africa:

Probably Not Normal:
Expected Value 2.592605193403763
Critical Value 0.508
Significance Level 15.0
Difference: 2.084605193403763
North America:
Probably Not Normal:
Expected Value 0.3928299107602231
Critical Value -1.296
Significance Level 15.0
Difference: 1.6888299107602232
Sub-Saharan Africa:
Probably Not Normal:
Expected Value 14.403652452170114
Critical Value 0.537
Significance Level 15.0
Difference: 13.866652452170113
East Asia & Pacific:
Probably Not Normal:
Expected Value 11.205740854501428
Critical Value 0.529
Significance Level 15.0
Difference: 10.676740854501428
Latin America & Caribbean:
Probably Not Normal:
Expected Value 7.185303306711631
Critical Value 0.533
Significance Level 15.0
Difference: 6.652303306711631
Europe & Central Asia:
Probably Not Normal:
Expected Value 8.695484468560878
Critical Value 0.543
Significance Level 15.0
Difference: 8.152484468560878
South Asia:
Probably Not Normal:
Expected Value 2.084604572240668
Critical Value 0.519
Significance Level 15.0
Difference: 1.5656045722406677
Min Difference: South Asia

Побудувати кругову діаграму населення по регіонам



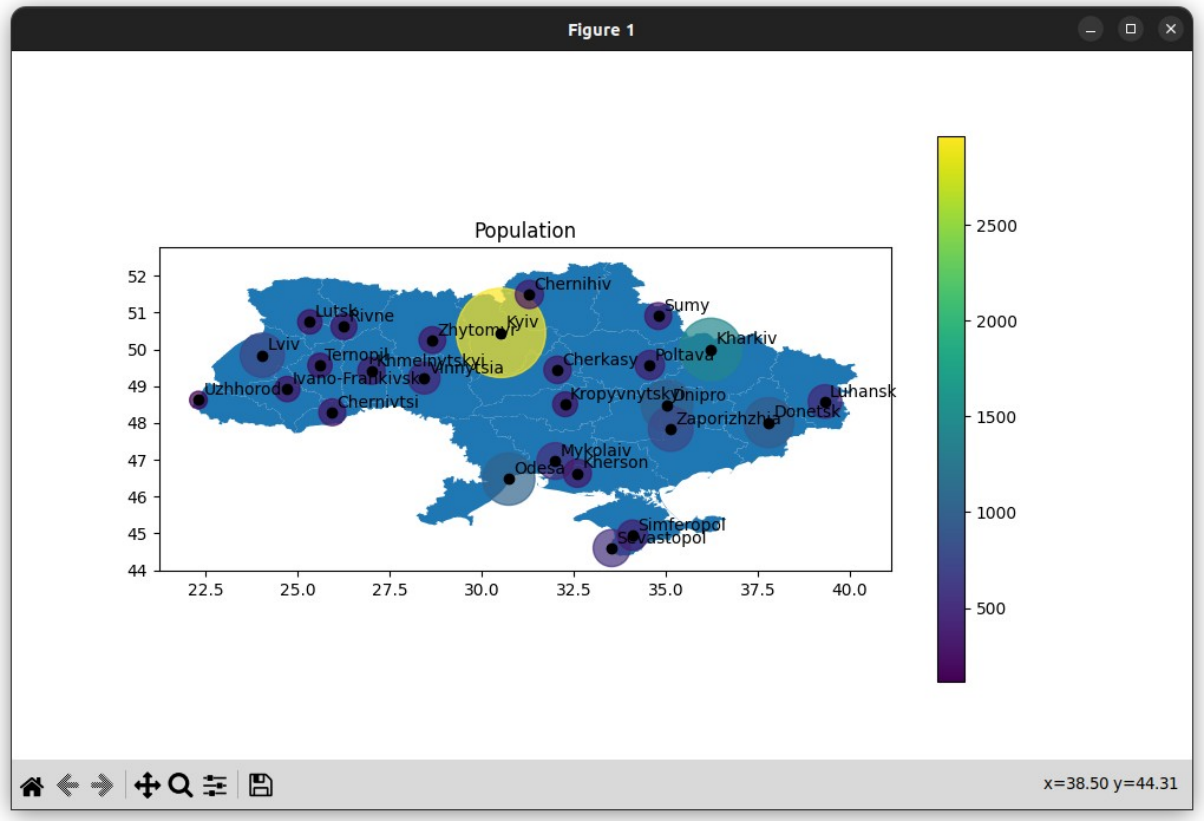
Додаткове завдання:

Перше:

Розмістити бульбашки, що відповідають їх населенню, на довільних 5 містах (статистику взяти в інтернеті)

```
# cities
cities = pd.read_csv('data/worldcities.csv')
cities = cities[(cities['country'] == 'Ukraine') & ((cities['capital'] == 'primary') |
(cities['capital'] == 'admin'))]
cities = gpd.GeoDataFrame(cities, geometry=gpd.points_from_xy(cities.lng, cities.lat))


def plot_cities(axis):
    cities.plot(ax=axis, color='black')
```

```
    for x, y, label in zip(cities.geometry.x, cities.geometry.y, cities.city):
        axis.annotate(label, xy=(x, y), xytext=(3, 3), textcoords="offset points")
# population bubbles
cities_bubbles = cities.copy()
cities_bubbles['geometry'] = cities_bubbles['geometry'].centroid
country = gpd.read_file('data/ukraine/ukr_admbnda_adm1_sspe_20230201.shp')
fig, axis = plt.subplots(figsize=(10, 6))
axis.set_title('Population')
country.plot(ax=axis)

cities_bubbles['population'] /= 1000
cities_bubbles.plot(ax=axis, column='population', markersize='population',
                    alpha=0.7, categorical=False, legend=True)
plot_cities(axis)
```



Знайти найбільшу відстань між містами в пікселях та кілометрах

```
def haversine(lon1, lat1, lon2, lat2):
    # convert decimal degrees to radians
    lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])
    # haversine formula
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = sin(dlat / 2) ** 2 + cos(lat1) * cos(lat2) * sin(dlon / 2) ** 2
    c = 2 * asin(sqrt(a))
    r = 6371
```

```
    return c * r

differences = []
for index_one, row_one in cities.iterrows():
    for index_two, row_two in cities.iterrows():
        sq_diff_lat = math.pow(row_one.lat - row_two.lat, 2)
        sq_diff_lng = math.pow(row_one.lng - row_two.lng, 2)
        dist_geo = math.sqrt(sq_diff_lat + sq_diff_lng)
        dist_km = haversine(row_one.lng, row_one.lat, row_two.lng, row_two.lat)
        differences.append((row_one.city, row_two.city, dist_geo, dist_km))

# max distance
differences.sort(key=lambda x: x[2])
max_diff = differences[-1]
print(max_diff)
```

('Uzhhorod', 'Luhansk', 17.03834837212809, 1250.2159211865564)

Друге

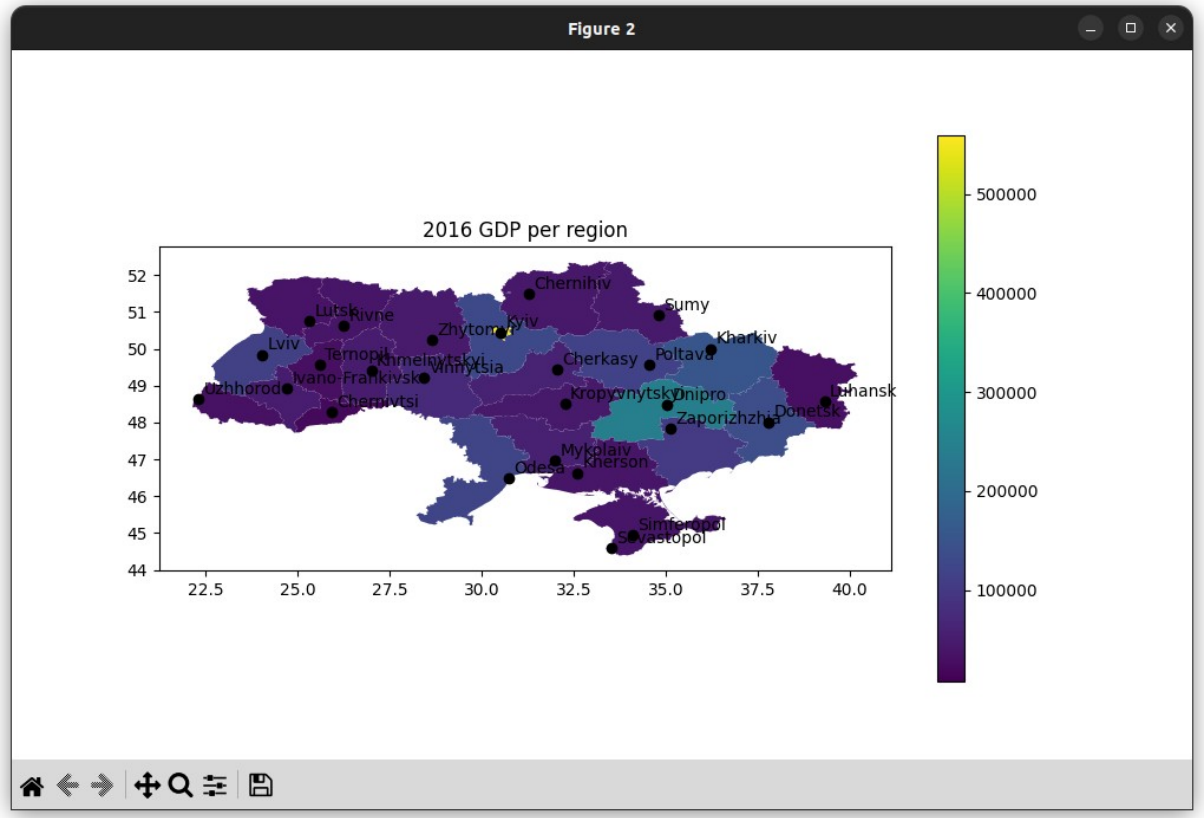
Перед тим, як будувати графіки ми маємо заповнити пропущені місця для ВВП у датафреймі. Для цього позначимо пусті місця -1, а потім за допомогою поліноміальної регресії знайдемо функцію, за допомогою якої спрогнозуємо можливі значення за відомими

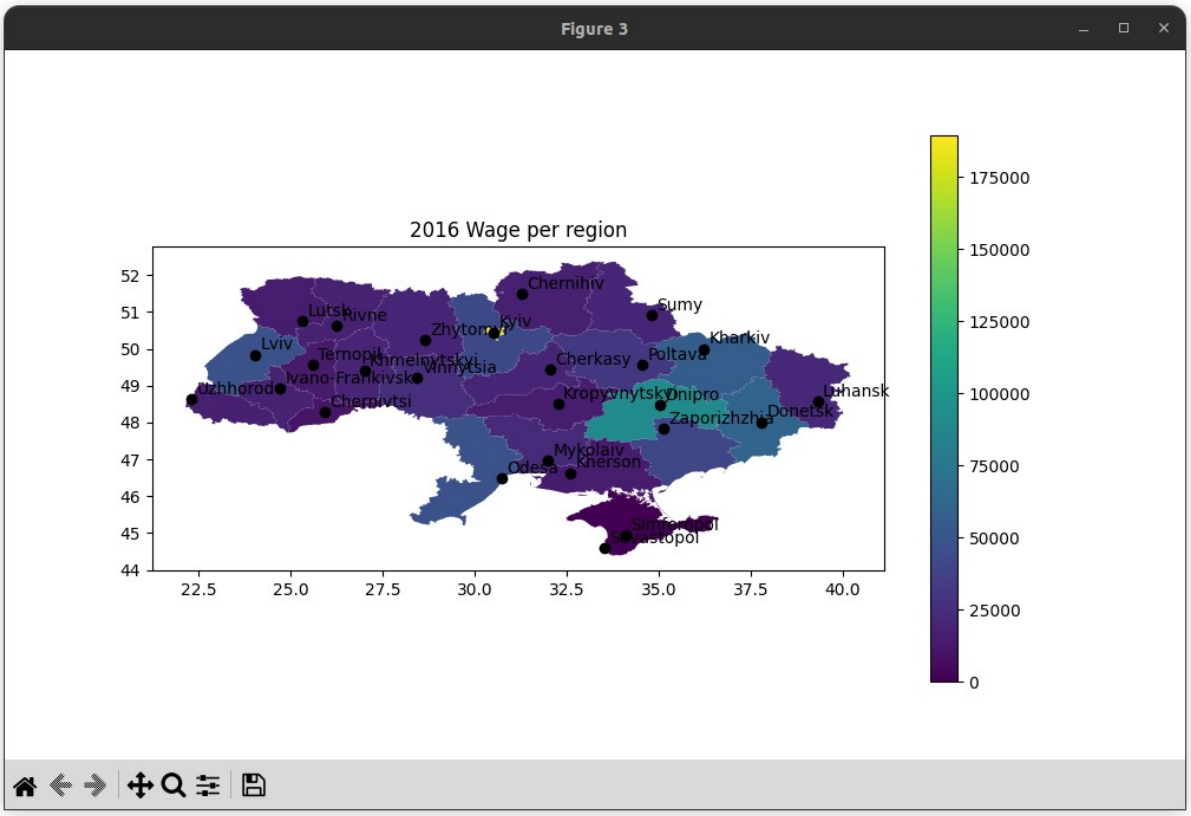
```
gdp = pd.read_csv('data/ukr_GDP.csv')

def build_regression_empty_value():
    for index, row in gdp.iterrows():
        nums = row.iloc[1:]
        positive = [(int(x) - 2006, y) for x, y in nums.items() if y >= 0]
        y = [y for x, y in positive]
        y = np.array(y)
        x = [x for x, y in positive]
        pow_deg = 2
        pow_x = [pow_deg for _ in x]
        x = np.array(x)
        degree = 2
        model = np.poly1d(np.polyfit(np.power(x, np.array(pow_x)), y, degree))
        for column, element in nums.items():
            if element < 0:
                gdp.at[index, column] = model(int(column) - 2006)
```

```
build_regression_empty_value()
```

Побудувати картограми для прибутку населення на 1 особу і ВВП по регіонам за 2016 рік.

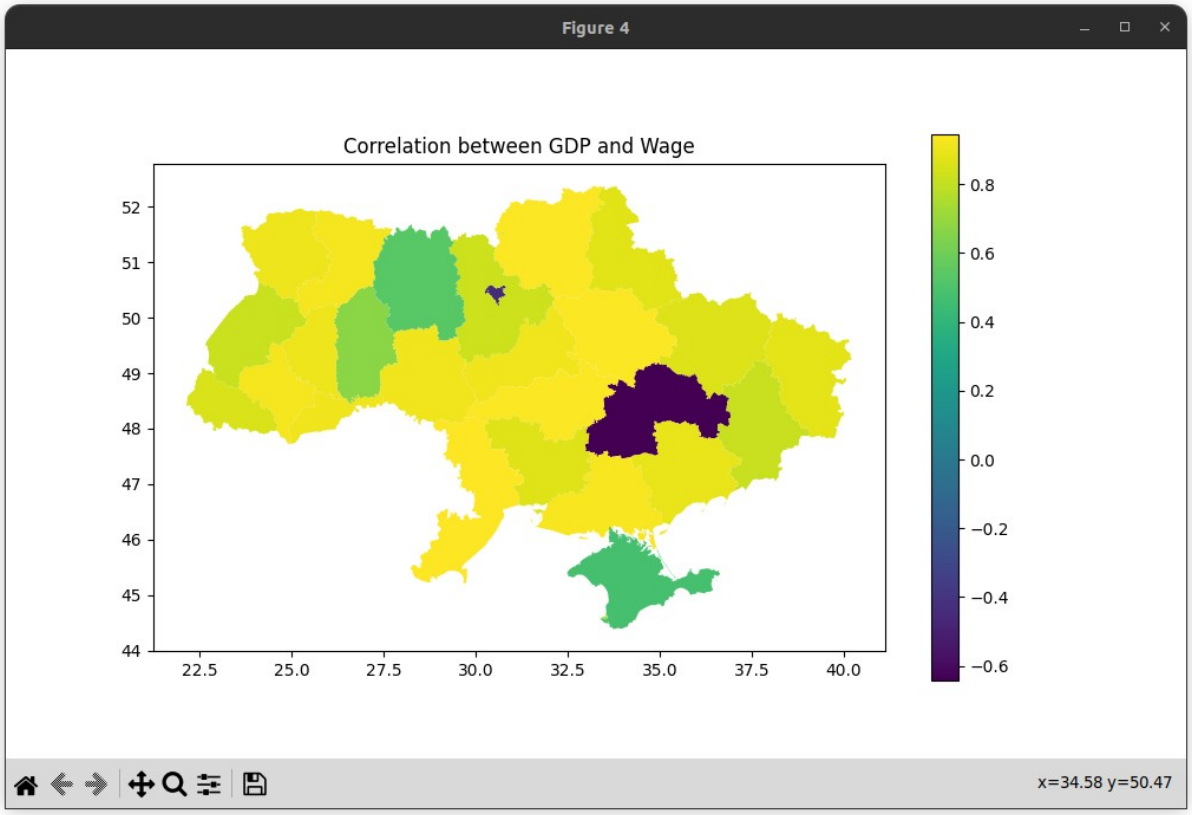
```
wages = pd.read_csv('data/ukr_ZP.csv')

country_gdp = country.copy()
country_wages = country.copy()

country_gdp = country_gdp.merge(gdp, how='left', on='ADM1_EN')
country_wages = country_wages.merge(wages, how='left', on='ADM1_EN')

   # GDP
fig, axis = plt.subplots(figsize=(10, 6))
axis.set_title('2016 GDP per region')
country_gdp.plot(ax=axis, column='2016', categorical=False, legend=True)
plot_cities(axis)


   # Wage
fig, axis = plt.subplots(figsize=(10, 6))
axis.set_title('2016 Wage per region')
country_wages.plot(ax=axis, column='2016', categorical=False, legend=True)
plot_cities(axis)
```





По даним за 2006-2015 роки для кожного регіону розрахувати коефіцієнт кореляції між прибутком населення на 1 особу та ВВП. Відобразити на картограмі.

```
fig, axis = plt.subplots(figsize=(10, 6))
axis.set_title('Correlation between GDP and Wage')
correlation = country.copy()
gdp_raw = gdp.loc[:, '2006':'2016']
wages_raw = wages.loc[:, '2006':'2016']
correlation_raw = gdp.corrwith(wages_raw, axis=1, numeric_only=True)
correlation['Correlation'] = correlation_raw
correlation.plot(ax=axis, column='Correlation', categorical=False, legend=True)
```



Висновок:

Під час лабораторної роботи проаналізував атрибути датасету на нормальний розподіл за допомогою тесту андерсона: жоден не виявився нормально розподіленим. Також провів дослід на рівність середнього значення(математичного очікування) медіані за допомогою одновибірково Т-критерію Стьюдента. Жоден не пройшов гіпотезу. Далі у додатковому завдання пропущені значення в датасеті зарплат були догенеровані за допомогою побудови поліному на основі вже відомих значень. Кореляція показує, що є чітка залежність між ВВП та зарплатами по регіонах. Однак є відхилення у Києва та Запоріжжя, що можна вважати статистичною аномалією. Графіки та код наведені.