**Звіт до лабораторної роботи 3**

Завдання:

1. Для виконання цього завдання у таблиці КНТ-811 обрати країну, відповідно до номеру варіанта, і сформувати таблицю, що буде містити дані про кількість студентів у кожному ЗВО відповідної країни. При цьому по рядках мають бути дані кожного закладу за всі роки, а по стовпчиках – дані по всіх закладах для відповідного року. Зберегти таблицю як файл формату CSV (роздільники – коми),

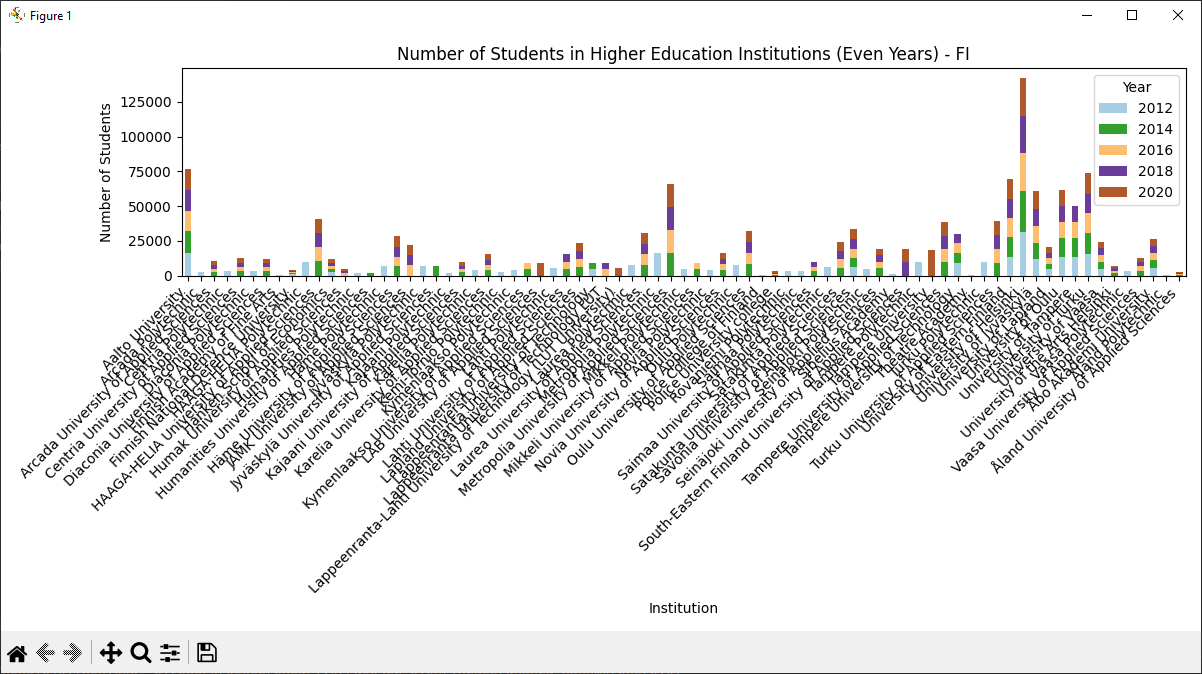
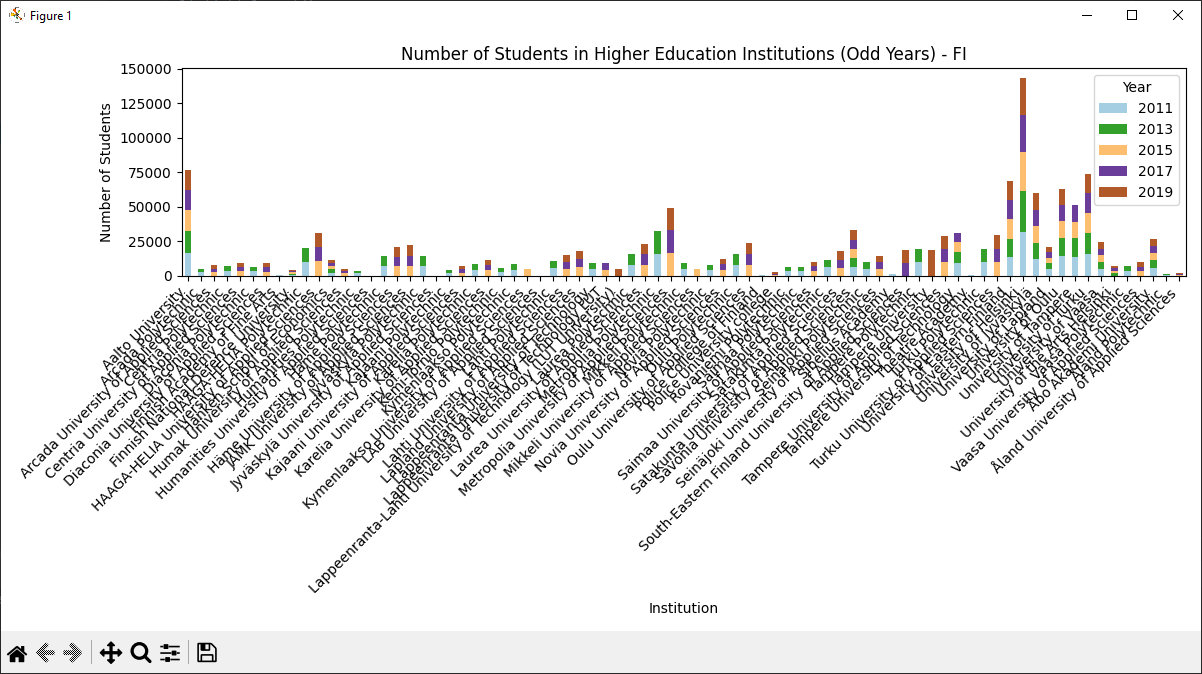
1.1. Для даних непарних років побудувати barplot діаграму. Назви років необхідно вказати у тексті легенди. Різні групи позначити різними кольорами. Обрати вертикальне розташування стовпчиків.

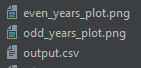
Створити загальний заголовок діаграми.

1.2. Для парних років побудувати barplot діаграму.

Код:

import pandas as pd  
import matplotlib.pyplot as plt  
  
# Зчитуємо дані з Excel-файлу  
df = pd.read\_excel('КНТ-811.xlsx')  
  
# Відфільтровуємо дані для країни FI (5 варіант)  
df\_fi = df[df['Country Code'] == 'FI']  
  
# Групуєємо дані за роками та інститутами, обчислюємо суму студентів  
df\_grouped = df\_fi.groupby(['Reference year', 'English Institution Name']).agg({'Total students enrolled ISCED 5-7': 'sum'}).reset\_index()  
  
# Перетворюємо таблицю для отримання кінцевого формату  
df\_pivot = df\_grouped.pivot(index='English Institution Name', columns='Reference year', values='Total students enrolled ISCED 5-7').fillna(0)  
  
# Зберігаємо таблицю у файл CSV  
df\_pivot.to\_csv('output.csv', sep=',')  
  
odd\_years\_data = df\_pivot[df\_pivot.columns[df\_pivot.columns % 2 != 0]]  
ax = odd\_years\_data.plot(kind='bar', colormap='Paired', stacked=True, rot=0, figsize=(12, 6))  
ax.set\_ylabel('Number of Students')  
ax.set\_xlabel('Institution')  
ax.set\_title('Number of Students in Higher Education Institutions (Odd Years) - {}'.format('FI'))  
ax.legend(title='Year')  
plt.xticks(rotation=45, ha='right')  
  
plt.tight\_layout()  
plt.savefig('odd\_years\_plot.png')  
  
plt.show()  
  
even\_years\_data = df\_pivot[df\_pivot.columns[df\_pivot.columns % 2 == 0]]  
ax = even\_years\_data.plot(kind='bar', colormap='Paired', stacked=True, rot=0, figsize=(12, 6))  
ax.set\_ylabel('Number of Students')  
ax.set\_xlabel('Institution')  
ax.set\_title('Number of Students in Higher Education Institutions (Even Years) - {}'.format('FI'))  
ax.legend(title='Year')  
  
plt.xticks(rotation=45, ha='right')  
  
plt.tight\_layout()  
  
plt.savefig('even\_years\_plot.png')  
  
# Show the plot  
plt.show()





Завдання: 2. Сформувати чотири вектори з такими параметрами:

x1 – містить 105 + n елементів, що підпорядковуються нормальному розподілу із середнім значенням 3n та стандартним відхиленням 0,35n;

x2 – містить 80 + 2n елементів, що підпорядковуються нормальному розподілу із середнім значенням 3,2n та стандартним відхиленням 0,4n;

x3 – містить 135 + 2n елементів, що підпорядковуються нормальному розподілу із середнім значенням 3,5n та стандартним відхиленням 0,4n;

x4 – містить 160 – 2n елементів, що підпорядковуються рівномірному розподілу на відрізку від 2,5n до 3,5n.

Побудувати такі boxplot діаграми:

2.1. Діаграма, стовпчики якої відповідають векторам x1, x2, x3, x4. Передбачити, що:

- на діаграмі треба зазначити назви компонентів;

- діаграми мають бути розташовані горизонтально для непарних n і вертикально для парних n;

- довжина вусів має дорівнювати 1,2 інтерквартильної відстані;

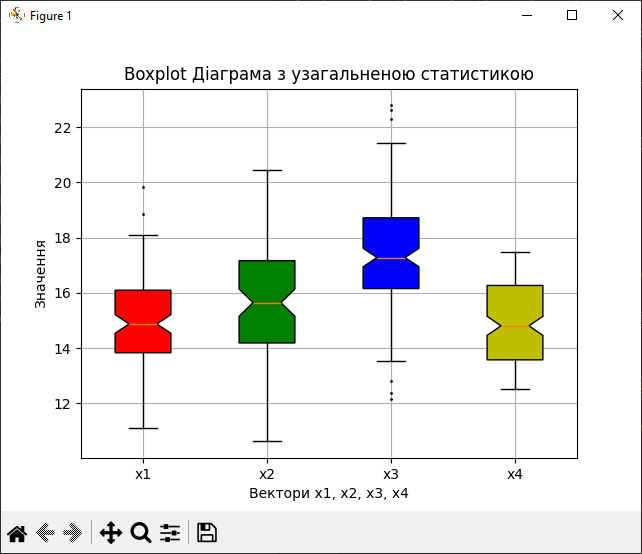
- довжина відрізка, що обмежує вуси, має бути рівною кількості спостережень для відповідного вектора поділеній на 120;

- необхідно показати довірчі інтервали для медіан у вигляді виїмок і точки, що виходять поза межі вусів;

- треба використати різнокольорову заливку прямокутників діаграми.

Код:

import numpy as np  
import matplotlib.pyplot as plt  
  
n = 5  
  
# Генеруємо масив x1, що підпорядковується нормальному розподілу  
x1 = np.random.normal(loc=3 \* n, scale=0.35 \* n, size=105 + n)  
x2 = np.random.normal(loc=3.2 \* n, scale=0.4 \* n, size=80 + 2 \* n)  
x3 = np.random.normal(loc=3.5 \* n, scale=0.4 \* n, size=135 + 2 \* n)  
x4 = np.random.uniform(low=2.5 \* n, high=3.5 \* n, size=160 - 2 \* n)  
  
# Побудова boxplot діаграми  
fig, ax = plt.subplots()  
  
data = [x1, x2, x3, x4]  
labels = ['x1', 'x2', 'x3', 'x4']  
colors = ['r', 'g', 'b', 'y']  
  
boxes = ax.boxplot(data, notch='True', vert=True, labels=labels, patch\_artist=True, whis=1.2, flierprops=dict(markersize=1))  
  
# Додавання узагальненої статистики  
for i, d in enumerate(data):  
 mean = np.mean(d)  
 median = np.median(d)  
 ax.text(i + 1, mean, f'Mean: {mean:.2f}', ha='center', va='bottom', fontweight='bold', color='black')  
 ax.text(i + 1, median, f'Median: {median:.2f}', ha='center', va='top', fontweight='bold', color='black')  
  
for patch, color in zip(boxes['boxes'], colors):  
 patch.set\_facecolor(color)  
  
ax.set\_title('Boxplot Діаграма з узагальненою статистикою')  
ax.set\_xlabel('Вектори x1, x2, x3, x4')  
ax.set\_ylabel('Значення')  
  
plt.grid(True)  
plt.show()



Завдання: 2.2. Діаграма, стовпчики якої відповідають векторам x1 + x4, об'єднанню векторів x2 та x4, (x2 + x3)/2. Передбачити, що:

- на діаграмі треба зазначити назви компонентів;

- розташування діаграм має бути горизонтальним для парних n і вертикальним для непарних n;

- довжина вусів має дорівнювати 0,9 інтерквартильної відстані;

- довжини відрізків, що обмежують вуси, мають дорівнювати n/(n + 2) від ширини прямокутників;

- ширина прямокутників має бути пропорційна квадратному кореню з кількості спостережень у групах;

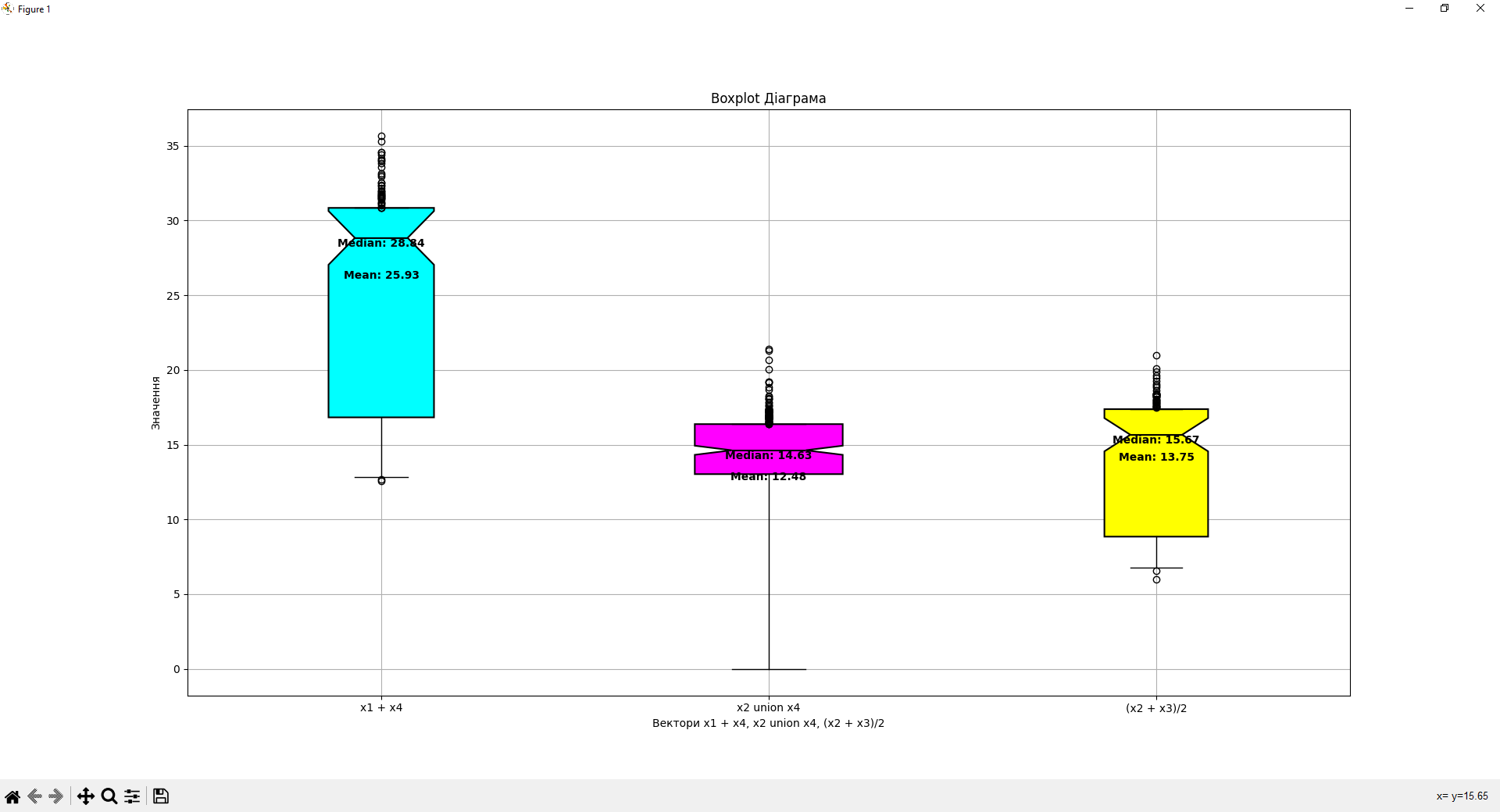
- кольори заливки прямокутників треба задати їх номерами від n + 16 n + 18;

- кольори ліній контурів треба задати назвами, а товщину ліній взяти рівною 1,5.

Для першої діаграми побудувати узагальнену статистику.

Код:

import numpy as np  
import matplotlib.pyplot as plt  
  
n = 5  
  
# Генеруємо дані для прикладу  
x1 = np.random.normal(loc=3 \* n, scale=0.35 \* n, size=105 + n)  
x2 = np.random.normal(loc=3.2 \* n, scale=0.4 \* n, size=80 + 2 \* n)  
x3 = np.random.normal(loc=3.5 \* n, scale=0.4 \* n, size=135 + 2 \* n)  
x4 = np.random.uniform(low=2.5 \* n, high=3.5 \* n, size=160 - 2 \* n)  
  
x1.resize(x4.shape)  
x2.resize(x3.shape)  
  
# Об'єднуємо вектори з допомогою concatenate  
x1\_plus\_x4 = x1 + x4  
x2\_union\_x4 = np.concatenate([x2, x4])  
avg\_x2\_x3 = (x2 + x3) / 2  
  
# Побудова boxplot діаграми для нових векторів  
fig, ax = plt.subplots()  
  
data = [x1\_plus\_x4, x2\_union\_x4, avg\_x2\_x3]  
labels = ['x1 + x4', 'x2 union x4', '(x2 + x3)/2']  
colors = ['cyan', 'magenta', 'yellow'] #кольори відповідні номерам 21-23 в мові Р  
  
# Визначаємо кількість спостережень у групах  
observations\_counts = [len(x1\_plus\_x4), len(x2\_union\_x4), len(avg\_x2\_x3)]  
widths = np.sqrt(observations\_counts)  
widths /= 45  
  
boxprops = dict(linewidth=1.5, color='black') # товщина ліній та колір контурів  
  
boxes = ax.boxplot(data, notch=True, vert=True, labels=labels, patch\_artist=True, widths=widths, whis=[0.9, n/(n+2)], boxprops=boxprops)  
  
for patch, color in zip(boxes['boxes'], colors):  
 patch.set\_facecolor(color)  
  
# Додаємо кольори ліній контурів та товщину ліній  
for median in boxes['medians']:  
 median.set(color='black', linewidth=1.5)  
  
# Додавання узагальненої статистики  
for i, d in enumerate(data):  
 mean = np.mean(d)  
 median = np.median(d)  
 ax.text(i + 1, mean, f'Mean: {mean:.2f}', ha='center', va='bottom', fontweight='bold', color='black')  
 ax.text(i + 1, median, f'Median: {median:.2f}', ha='center', va='top', fontweight='bold', color='black')  
  
ax.set\_title('Boxplot Діаграма')  
ax.set\_xlabel('Вектори x1 + x4, x2 union x4, (x2 + x3)/2')  
ax.set\_ylabel('Значення')  
  
plt.grid(True)  
plt.show()



Завдання:

3. Для даних 2020 р. свого варіанту побудувати кругову діаграму. Для цього впорядкувати вектор даних за спаданням і взяти дані для 5 найбільших закладів. Інших закладів створити групу «інші» Аргументи діаграми обрати самостійно.

Код:

import pandas as pd  
import matplotlib.pyplot as plt  
  
# Зчитуємо дані з Excel-файлу  
df = pd.read\_excel('КНТ-811.xlsx')  
  
  
df\_fi = df[df['Country Code'] == 'FI']  
  
df\_filtered = df\_fi[(df\_fi['Reference year'] == 2020)]  
  
df\_sorted = df\_filtered.sort\_values(by='Total students enrolled ISCED 5-7', ascending=False)  
  
top\_5 = df\_sorted.head(5)  
  
# Сумуємо дані для решти закладів і створюємо рядок "інші"  
other\_data = pd.Series({'English Institution Name': 'other', 'Total students enrolled ISCED 5-7': df\_sorted.iloc[5:]['Total students enrolled ISCED 5-7'].sum()})  
  
plot\_data = pd.concat([top\_5, pd.DataFrame([other\_data])])  
  
plt.figure(figsize=(8, 8))  
plt.pie(plot\_data['Total students enrolled ISCED 5-7'], labels=plot\_data['English Institution Name'], autopct='%1.1f%%', startangle=90, colors=plt.cm.Paired.colors)  
plt.title('Розподіл студентів за закладами (рік 2020, рівень ISCED 5-7)')  
plt.show()

