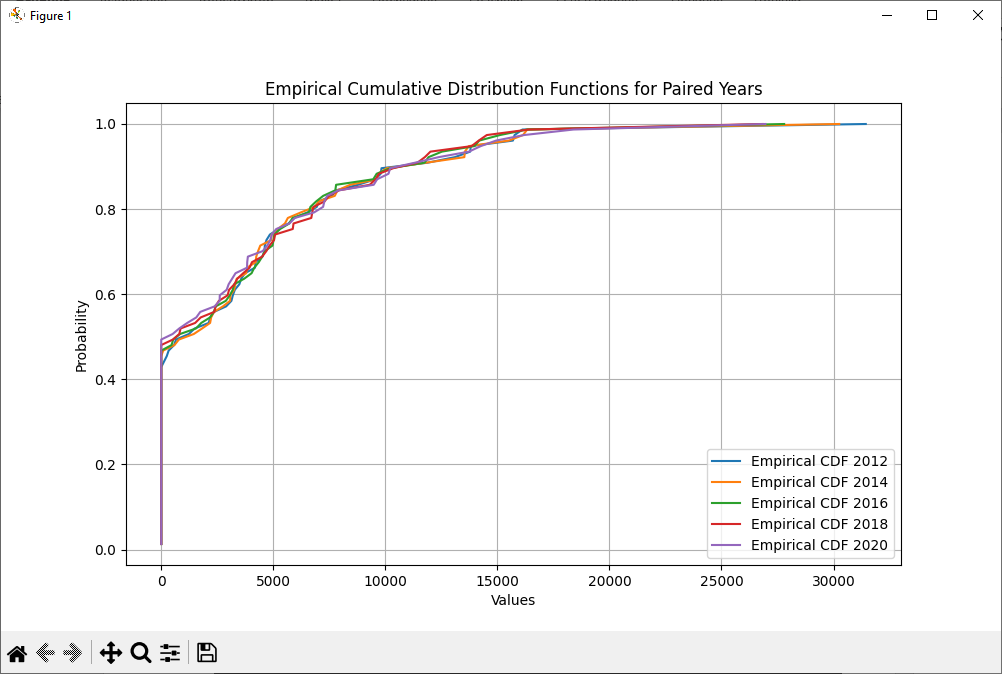
**Звіт до лабораторної роботи 5**

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
from scipy.stats import probplot  
import scipy

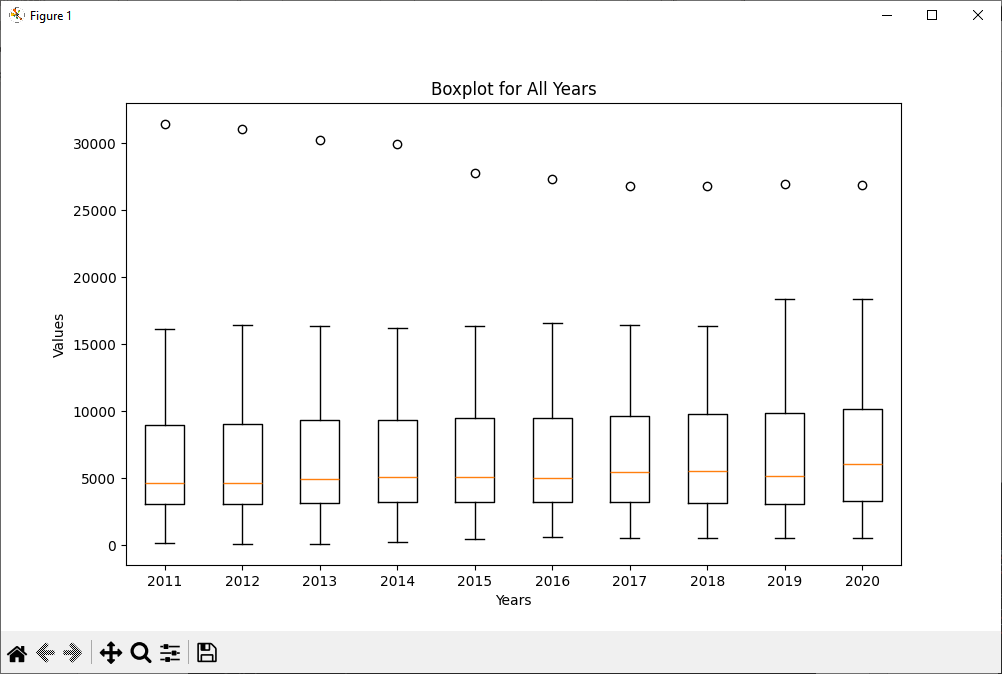
1.З даних таблиці, що використовувалася в попередніх завданнях, сформувати вектори, що відповідають кожному року за період 2011 – 2020 р

# Зчитуємо файл output.csv  
data = pd.read\_csv('output.csv')  
# Отримаємо вектор(список) для нашого варіанту де дані це English Institution Name,2011,2012,...,2020  
data = data.values.tolist()  
  
  
 # №2  
# Отримаймо роки та дані для парних років  
years = [2012, 2014, 2016, 2018, 2020]  
data\_pairs = [row[1::2] for row in data]  
  
# Ініціалізуймо графік  
plt.figure(figsize=(10, 6))  
  
# Побудуймо емпіричні функції розподілу для парних років  
for i in range(0, len(data\_pairs[0])):  
 year = years[i]  
 values = [row[i] for row in data\_pairs]  
  
 # Створемо емпіричну функцію розподілу за допомогою numpy  
 x = np.sort(values)  
 y = np.arange(1, len(x) + 1) / len(x)  
  
 # Побудуймо графік для кожного парного року  
 plt.plot(x, y, label=f'Empirical CDF {year}')  
  
# Налаштуймо графік  
plt.title('Empirical Cumulative Distribution Functions for Paired Years')  
plt.xlabel('Values')  
plt.ylabel('Probability')  
plt.legend()  
plt.grid(True)  
  
plt.show()



3. Побудувати на одному графіку діаграми типу "ящик з вусами" для всіх вибірок.

# №3  
# Отримаймо дані для всіх років  
data\_all\_years = [row[1:] for row in data]  
years = [2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020]  
  
# Переробимо data\_all\_years  
transformed\_data = list(map(list, zip(\*data\_all\_years)))  
transformed\_data = [  
 [value for value in row if value != 0] for row in transformed\_data  
]  
  
# Ініціалізуймо графік  
plt.figure(figsize=(10, 6))  
  
# Побудуймо діаграму "ящик з вусами" для всіх років  
plt.boxplot(transformed\_data, labels=[str(year) for year in years])  
  
# Налаштуймо графік  
plt.title('Boxplot for All Years')  
plt.xlabel('Years')  
plt.ylabel('Values')  
  
plt.show()



4. Розрахувати і порівняти квартилі розподілу для всіх вибірок  
 # № 4  
# Розрахуймо квартилі для всіх років  
quartiles = np.percentile(data\_all\_years, [25, 50, 75], axis=0)  
  
# Виведемо результати  
for i, year in enumerate(range(2011, 2021)):  
 print(f"\nQuartiles for {year}:")  
 print(f"Q1: {quartiles[0, i]}")  
 print(f"Q2 (Median): {quartiles[1, i]}")  
 print(f"Q3: {quartiles[2, i]}")

Quartiles for 2011:

Q1: 0.0

Q2 (Median): 1234.0

Q3: 5235.0

Quartiles for 2012:

Q1: 0.0

Q2 (Median): 1199.0

Q3: 5244.0

Quartiles for 2013:

Q1: 0.0

Q2 (Median): 1453.0

Q3: 5205.0

Quartiles for 2014:

Q1: 0.0

Q2 (Median): 861.0

Q3: 5417.0

Quartiles for 2015:

Q1: 0.0

Q2 (Median): 840.0

Q3: 5305.0

Quartiles for 2016:

Q1: 0.0

Q2 (Median): 824.0

Q3: 5146.0

Quartiles for 2017:

Q1: 0.0

Q2 (Median): 806.0

Q3: 5869.0

Quartiles for 2018:

Q1: 0.0

Q2 (Median): 878.0

Q3: 5841.0

Quartiles for 2019:

Q1: 0.0

Q2 (Median): 509.0

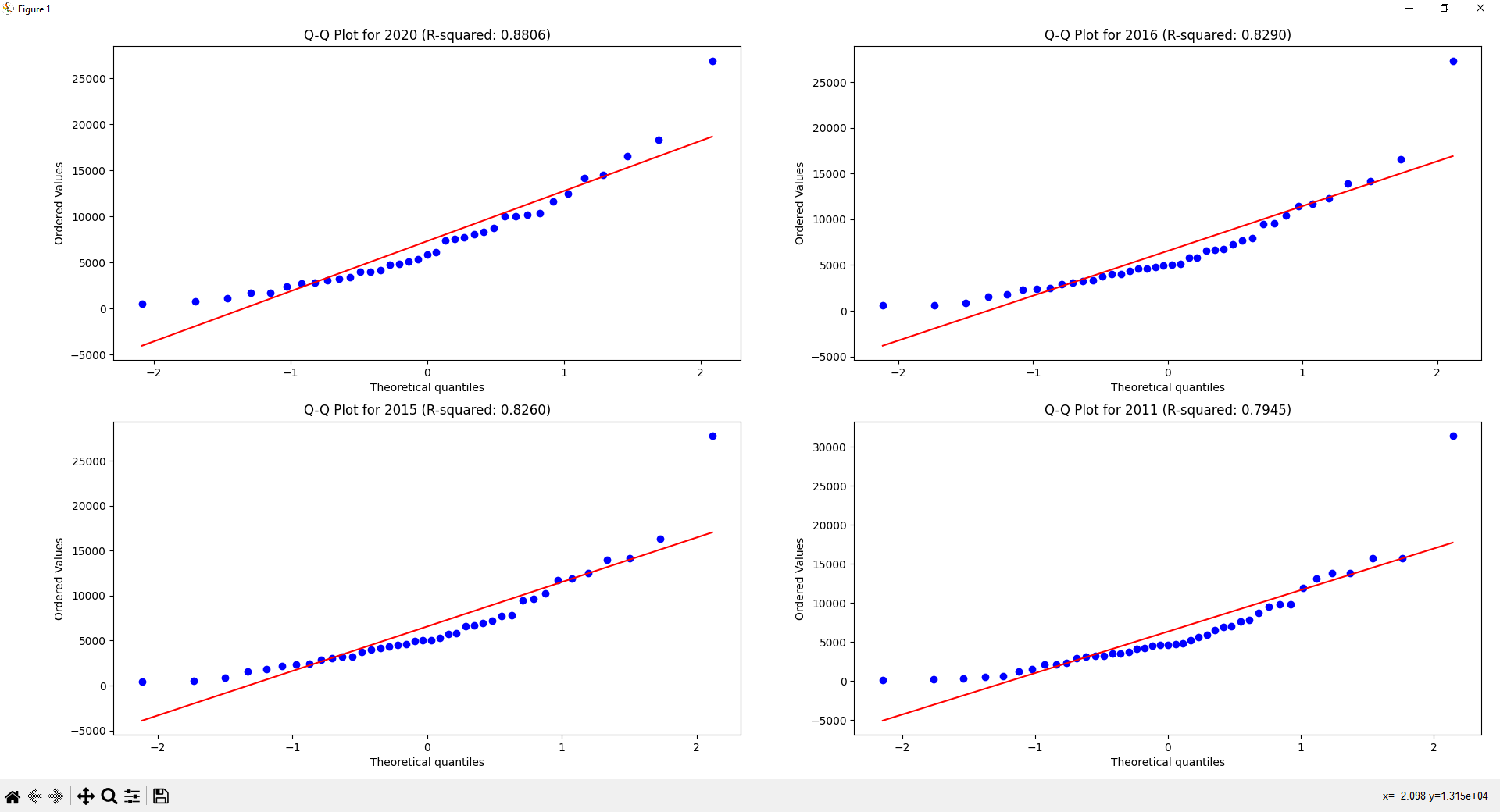
Q3: 5132.0

Quartiles for 2020:

Q1: 0.0

Q2 (Median): 0.0

Q3: 5901.0

5. Побудувати Q-Q діаграми вибірок 2020, 2016, 2015 та 2011 р. і зробити попередній висновок про їх відповідність нормальному розподілу.  
  
  
 # № 5  
data\_2020 = np.array(list(filter(lambda x: x != 0, [row[10] for row in data[1:]])))  
data\_2016 = np.array(list(filter(lambda x: x != 0, [row[6] for row in data[1:]])))  
data\_2015 = np.array(list(filter(lambda x: x != 0, [row[5] for row in data[1:]])))  
data\_2011 = np.array(list(filter(lambda x: x != 0, [row[1] for row in data[1:]])))  
  
# Побудуймо Q-Q діаграми  
fig, axes = plt.subplots(2, 2, figsize=(12, 10))  
axes = axes.flatten()  
  
for ax, data, year in zip(axes, [data\_2020, data\_2016, data\_2015, data\_2011], [2020, 2016, 2015, 2011]):  
 \_, (\_, \_, r) = probplot(data, plot=ax, fit=True, dist='norm')  
 ax.set\_title(f'Q-Q Plot for {year} (R-squared: {r\*\*2:.4f})')  
  
plt.tight\_layout()  
plt.show()  
  


6. Перевірити відповідність зазначених вибірок нормальному розподілу за критеріями Шапіро-Уїлка, Андерсона-Дарлінга та Ліллієфорса і зробити відповідний висновок.  
 # № 6  
print('\n'\*5)  
# не знайшов функції для Ліллієфорса тому використав Колмагорова-Смірнова  
  
# Перевірка відповідності нормальному розподілу за критерієм Шапіро-Уїлка  
stat\_shapiro, p\_shapiro = scipy.stats.shapiro(data\_2020)  
print(f'Shapiro-Wilk Test for 2020: Statistic={stat\_shapiro:.4f}, p-value={p\_shapiro:.4f}')  
if p\_shapiro > 0.05:  
 print('можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
else:  
 print('не можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
  
  
# Перевірка відповідності нормальному розподілу за критерієм Андерсона-Дарлінга  
result\_anderson = scipy.stats.anderson(data\_2020)  
print(f'Anderson-Darling Test for 2020: Statistic={result\_anderson.statistic:.4f}, Critical Value={result\_anderson.critical\_values[2]}, Significance Level={result\_anderson.significance\_level[2]}')  
if result\_anderson.statistic > result\_anderson.critical\_values[2]:  
 print('не можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
else:  
 print('можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
  
  
# Перевірка відповідності нормальному розподілу за критерієм Ліллієфорса  
stat\_kstest, p\_kstest = scipy.stats.kstest(data\_2020, 'norm')  
print(f'Kolmogorov-Smirnov Test for 2020: Statistic={stat\_kstest:.4f}, p-value={p\_kstest:.4f}')  
if p\_kstest > 0.05:  
 print('можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
else:  
 print('не можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
  
print('\n'\*2)  
  
# Перевірка відповідності нормальному розподілу за критерієм Шапіро-Уїлка  
stat\_shapiro, p\_shapiro = scipy.stats.shapiro(data\_2016)  
print(f'Shapiro-Wilk Test for 2016: Statistic={stat\_shapiro:.4f}, p-value={p\_shapiro:.4f}')  
if p\_shapiro > 0.05:  
 print('можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
else:  
 print('не можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
  
# Перевірка відповідності нормальному розподілу за критерієм Андерсона-Дарлінга  
result\_anderson = scipy.stats.anderson(data\_2016)  
print(  
 f'Anderson-Darling Test for 2016: Statistic={result\_anderson.statistic:.4f}, Critical Value={result\_anderson.critical\_values[2]}, Significance Level={result\_anderson.significance\_level[2]}')  
if result\_anderson.statistic > result\_anderson.critical\_values[2]:  
 print('не можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
else:  
 print('можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
  
# Перевірка відповідності нормальному розподілу за критерієм Ліллієфорса  
stat\_kstest, p\_kstest = scipy.stats.kstest(data\_2016, 'norm')  
print(f'Kolmogorov-Smirnov Test for 2016: Statistic={stat\_kstest:.4f}, p-value={p\_kstest:.4f}')  
if p\_kstest > 0.05:  
 print('можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
else:  
 print('не можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
  
print('\n' \* 2)  
  
# Перевірка відповідності нормальному розподілу за критерієм Шапіро-Уїлка  
stat\_shapiro, p\_shapiro = scipy.stats.shapiro(data\_2015)  
print(f'Shapiro-Wilk Test for 2015: Statistic={stat\_shapiro:.4f}, p-value={p\_shapiro:.4f}')  
if p\_shapiro > 0.05:  
 print('можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
else:  
 print('не можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
  
# Перевірка відповідності нормальному розподілу за критерієм Андерсона-Дарлінга  
result\_anderson = scipy.stats.anderson(data\_2015)  
print(  
 f'Anderson-Darling Test for 2015: Statistic={result\_anderson.statistic:.4f}, Critical Value={result\_anderson.critical\_values[2]}, Significance Level={result\_anderson.significance\_level[2]}')  
if result\_anderson.statistic > result\_anderson.critical\_values[2]:  
 print('не можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
else:  
 print('можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
  
# Перевірка відповідності нормальному розподілу за критерієм Ліллієфорса  
stat\_kstest, p\_kstest = scipy.stats.kstest(data\_2015, 'norm')  
print(f'Kolmogorov-Smirnov Test for 2015: Statistic={stat\_kstest:.4f}, p-value={p\_kstest:.4f}')  
if p\_kstest > 0.05:  
 print('можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
else:  
 print('не можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
  
print('\n' \* 2)  
  
# Перевірка відповідності нормальному розподілу за критерієм Шапіро-Уїлка  
stat\_shapiro, p\_shapiro = scipy.stats.shapiro(data\_2011)  
print(f'Shapiro-Wilk Test for 2011: Statistic={stat\_shapiro:.4f}, p-value={p\_shapiro:.4f}')  
if p\_shapiro > 0.05:  
 print('можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
else:  
 print('не можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
  
# Перевірка відповідності нормальному розподілу за критерієм Андерсона-Дарлінга  
result\_anderson = scipy.stats.anderson(data\_2011)  
print(  
 f'Anderson-Darling Test for 2011: Statistic={result\_anderson.statistic:.4f}, Critical Value={result\_anderson.critical\_values[2]}, Significance Level={result\_anderson.significance\_level[2]}')  
if result\_anderson.statistic > result\_anderson.critical\_values[2]:  
 print('не можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
else:  
 print('можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
  
# Перевірка відповідності нормальному розподілу за критерієм Ліллієфорса  
stat\_kstest, p\_kstest = scipy.stats.kstest(data\_2011, 'norm')  
print(f'Kolmogorov-Smirnov Test for 2011: Statistic={stat\_kstest:.4f}, p-value={p\_kstest:.4f}')  
if p\_kstest > 0.05:  
 print('можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
else:  
 print('не можна припустити що дані розподілено нормально з рівнем значимості 0.05')  
  
print('\n' \* 2)

Shapiro-Wilk Test for 2020: Statistic=0.8868, p-value=0.0013

не можна припустити що дані розподілено нормально з рівнем значимості 0.05

Anderson-Darling Test for 2020: Statistic=1.0308, Critical Value=0.722, Significance Level=5.0

не можна припустити що дані розподілено нормально з рівнем значимості 0.05

Kolmogorov-Smirnov Test for 2020: Statistic=1.0000, p-value=0.0000

не можна припустити що дані розподілено нормально з рівнем значимості 0.05

Shapiro-Wilk Test for 2016: Statistic=0.8405, p-value=0.0001

не можна припустити що дані розподілено нормально з рівнем значимості 0.05

Anderson-Darling Test for 2016: Statistic=1.5490, Critical Value=0.726, Significance Level=5.0

не можна припустити що дані розподілено нормально з рівнем значимості 0.05

Kolmogorov-Smirnov Test for 2016: Statistic=1.0000, p-value=0.0000

не можна припустити що дані розподілено нормально з рівнем значимості 0.05

Shapiro-Wilk Test for 2015: Statistic=0.8381, p-value=0.0000

не можна припустити що дані розподілено нормально з рівнем значимості 0.05

Anderson-Darling Test for 2015: Statistic=1.5534, Critical Value=0.726, Significance Level=5.0

не можна припустити що дані розподілено нормально з рівнем значимості 0.05

Kolmogorov-Smirnov Test for 2015: Statistic=1.0000, p-value=0.0000

не можна припустити що дані розподілено нормально з рівнем значимості 0.05

Shapiro-Wilk Test for 2011: Statistic=0.8079, p-value=0.0000

не можна припустити що дані розподілено нормально з рівнем значимості 0.05

Anderson-Darling Test for 2011: Statistic=1.8755, Critical Value=0.729, Significance Level=5.0

не можна припустити що дані розподілено нормально з рівнем значимості 0.05

Kolmogorov-Smirnov Test for 2011: Statistic=1.0000, p-value=0.0000

не можна припустити що дані розподілено нормально з рівнем значимості 0.05

7. Взяти пари вибірок, які відповідають 2020 та 2015 р., 2016 та2011 р. Якщо всі ці вибірки підпорядковуються нормальному розподілу, перевірити їх однорідність за критеріями Уелча та Фішера. Якщо ця умова не виконується виконати перевірку однорідності за критеріями Колмогорова-Смирнова, Уїлкоксона та Ансарі-Бредлі. Зробити висновок про однорідність для кожної пари вибірок.  
  
 # № 7  
from scipy.stats import ttest\_ind, f\_oneway, ks\_2samp, mannwhitneyu, ansari, shapiro  
  
  
# Перевірка нормальності за критерієм Шапіро-Уїлка (необхідно вже перевірити перед використанням критеріїв Уелча та Фішера)  
\_, p\_shapiro\_2020 = shapiro(data\_2020)  
\_, p\_shapiro\_2015 = shapiro(data\_2015)  
\_, p\_shapiro\_2016 = shapiro(data\_2016)  
\_, p\_shapiro\_2011 = shapiro(data\_2011)  
  
# Визначення однорідності за критеріями Уелча та Фішера (якщо всі вибірки нормальні)  
if all(p > 0.05 for p in [p\_shapiro\_2020, p\_shapiro\_2015, p\_shapiro\_2016, p\_shapiro\_2011]):  
 # Перевірка за критерієм Уелча  
 stat\_welch, p\_welch = ttest\_ind(data\_2020, data\_2015, equal\_var=False)  
 print(f'Welch Test for 2020 and 2015: Statistic={stat\_welch:.4f}, p-value={p\_welch:.4f}')  
 if p\_welch > 0.05:  
 print('Можна припустити однорідність.')  
 else:  
 print('Не можна припустити однорідність.')  
  
 # Перевірка за критерієм Уелча  
 stat\_welch, p\_welch = ttest\_ind(data\_2016, data\_2011, equal\_var=False)  
 print(f'Welch Test for 2016 and 2011: Statistic={stat\_welch:.4f}, p-value={p\_welch:.4f}')  
 if p\_welch > 0.05:  
 print('Можна припустити однорідність.')  
 else:  
 print('Не можна припустити однорідність.')  
  
 # Перевірка за критерієм Фішера  
 stat\_f, p\_f = f\_oneway(data\_2020, data\_2015, data\_2016, data\_2011)  
 print(f'Fisher Test for all pairs: Statistic={stat\_f:.4f}, p-value={p\_f:.4f}')  
 if p\_f > 0.05:  
 print('Можна припустити однорідність.')  
 else:  
 print('Не можна припустити однорідність.')  
  
else:  
 print('Не всі вибірки мають нормальний розподіл.')  
  
 # Перевірка однорідності за критеріями Колмогорова-Смірнова, Уілкоксона та Ансарі-Бредлі  
 stat\_ks, p\_ks = ks\_2samp(data\_2020, data\_2015)  
 print(f'Kolmogorov-Smirnov Test for 2020 and 2015: Statistic={stat\_ks:.4f}, p-value={p\_ks:.4f}')  
 if p\_ks > 0.05:  
 print('Можна припустити однорідність.')  
 else:  
 print('Не можна припустити однорідність.')  
  
 stat\_mw, p\_mw = mannwhitneyu(data\_2020, data\_2015, alternative='two-sided')  
 print(f'Mann-Whitney U Test for 2020 and 2015: Statistic={stat\_mw:.4f}, p-value={p\_mw:.4f}')  
 if p\_mw > 0.05:  
 print('Можна припустити однорідність.')  
 else:  
 print('Не можна припустити однорідність.')  
  
 stat\_ansari, p\_ansari = ansari(data\_2020, data\_2015)  
 print(f'Ansari-Bradley Test for 2020 and 2015: Statistic={stat\_ansari:.4f}, p-value={p\_ansari:.4f}')  
 if p\_ansari > 0.05:  
 print('Можна припустити однорідність.')  
 else:  
 print('Не можна припустити однорідність.')  
  
 # Перевірка однорідності за критеріями Колмогорова-Смірнова, Уілкоксона та Ансарі-Бредлі  
 stat\_ks, p\_ks = ks\_2samp(data\_2016, data\_2011)  
 print(f'Kolmogorov-Smirnov Test for 2016 and 2011: Statistic={stat\_ks:.4f}, p-value={p\_ks:.4f}')  
 if p\_ks > 0.05:  
 print('Можна припустити однорідність.')  
 else:  
 print('Не можна припустити однорідність.')  
  
 stat\_mw, p\_mw = mannwhitneyu(data\_2016, data\_2011, alternative='two-sided')  
 print(f'Mann-Whitney U Test for 2016 and 2011: Statistic={stat\_mw:.4f}, p-value={p\_mw:.4f}')  
 if p\_mw > 0.05:  
 print('Можна припустити однорідність.')  
 else:  
 print('Не можна припустити однорідність.')  
  
 stat\_ansari, p\_ansari = ansari(data\_2016, data\_2011)  
 print(f'Ansari-Bradley Test for 2016 and 2011: Statistic={stat\_ansari:.4f}, p-value={p\_ansari:.4f}')  
 if p\_ansari > 0.05:  
 print('Можна припустити однорідність.')  
 else:  
 print('Не можна припустити однорідність.')

Не всі вибірки мають нормальний розподіл.

Kolmogorov-Smirnov Test for 2020 and 2015: Statistic=0.1595, p-value=0.6376

Можна припустити однорідність.

Mann-Whitney U Test for 2020 and 2015: Statistic=803.0000, p-value=0.5240

Можна припустити однорідність.

Ansari-Bradley Test for 2020 and 2015: Statistic=698.0000, p-value=0.5115

Можна припустити однорідність.

Kolmogorov-Smirnov Test for 2016 and 2011: Statistic=0.0953, p-value=0.9760

Можна припустити однорідність.

Mann-Whitney U Test for 2016 and 2011: Statistic=914.0000, p-value=0.6259

Можна припустити однорідність.

Ansari-Bradley Test for 2016 and 2011: Statistic=884.0000, p-value=0.5449

Можна припустити однорідність.