МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



**Лабораторна робота №1**

**З дисципліни «Інтелектуальний аналіз даних»**

***Виконав:*** *студент групи КН-310*

*Бурак Марко*

***Викладач:***

*Якимишин Х.М.*

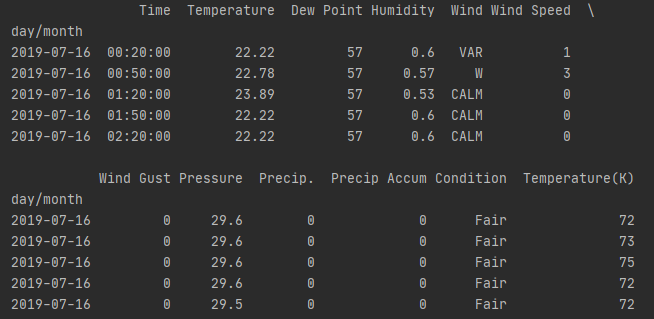
Львів – 2020

**Тема:** Візуалізація даних різного типу

**Хід Роботи**

1. Завантажуємо дані та досліджуємо їх:

df = pd.read\_csv('database.csv', sep=';')

****

*Рис. 1 Інформація про дані*

1. Пишу функції для конвертування даних



*Рис. 2 Функції*

1. Пишу парсер даних для правильної конвертації



*Рис. 3 Парсер*

1. Побудуємо точковий графік, для кожного можливого стовпця

if answ2 == 'pressure':

p\_max = df2.groupby('day/month').agg({'Pressure': 'max'})

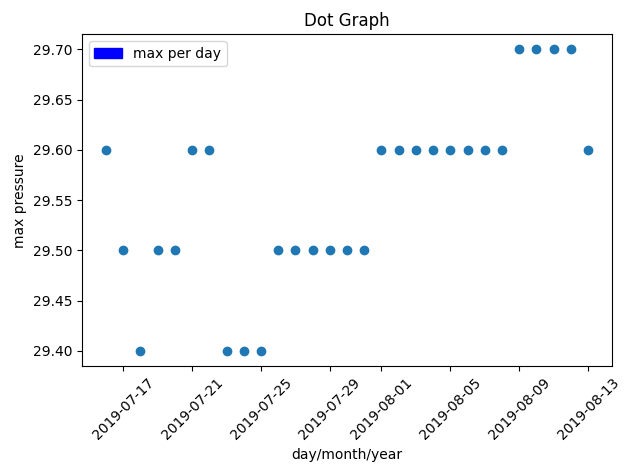
plt.plot(p\_max, 'o')

plt.xticks(rotation=45)

plt.xlabel('day/month/year')

plt.ylabel('max pressure')

plt.show()



*Рис. 4 точковий графік*

1. Побудуємо лінійний графік

pd.options.mode.chained\_assignment = None

t\_max = df2.groupby('day/month').agg({'Temperature': 'max'})

t\_max1 = t\_max.iloc[:int(len(t\_max) / 2)]

t\_max2 = t\_max.iloc[int(len(t\_max) / 2 + 1):]

index\_list = list(range(1, len(t\_max1) + 1))

t\_max1['count'] = index\_list

t\_max2['count'] = index\_list

ax = plt.gca()

t\_max1.plot(kind='line', x='count', y='Temperature', ax=ax)

t\_max2.plot(kind='line', x='count', y='Temperature', ax=ax)

ax.legend(["First half", "Second half"])

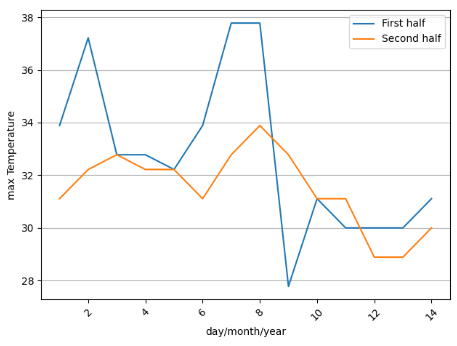
ax.grid('on', which='minor', axis='y')

ax.grid('off', which='major', axis='y')

plt.xticks(rotation=45)

plt.xlabel('day/month/year')

plt.ylabel('max Temperature')



*Рис. 5 Лінійний графік*

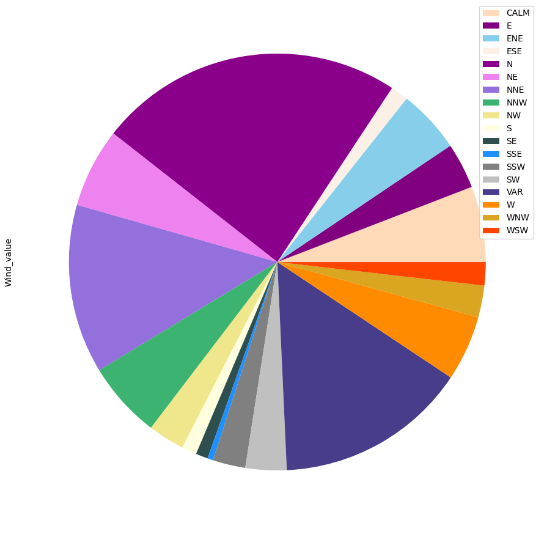
1. Графік Pie Chart

sums\_wind = df2.groupby('Wind').size().reset\_index(name='Wind\_value')

sums\_wind.set\_index('Wind', inplace=True)

sums\_wind.plot.pie(y='Wind\_value', figsize=(9, 9), labels=None, colors=colors)

plt.show()



*Рис. 6 Pie Chart*

1. Побудую гістограму для кожного стовпця

sums\_wind\_speed = df2.groupby('Wind Speed').size().reset\_index(name='Wind speed frequency')

sums\_wind\_speed.set\_index('Wind Speed', inplace=True)

sums\_wind\_speed.hist()

plt.xlabel('Velocity', fontsize=15)

plt.ylabel('Frequency per day', fontsize=15)

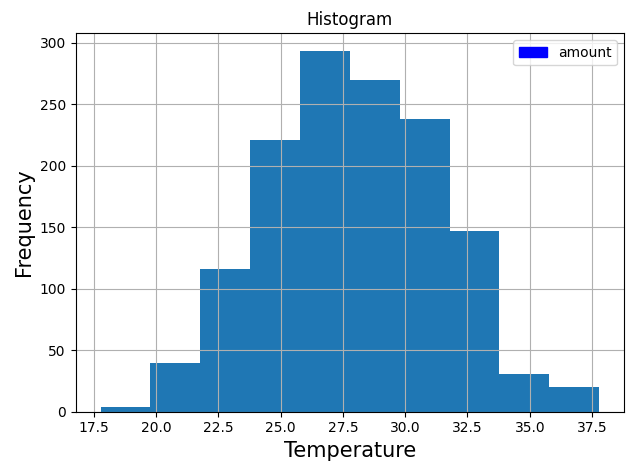
plt.xticks(fontsize=15)

plt.yticks(fontsize=15)

x\_new = [x for x in range(0, 140, 10)]

plt.xticks(x\_new)

plt.show()



*Рис. 7 Гістограма*

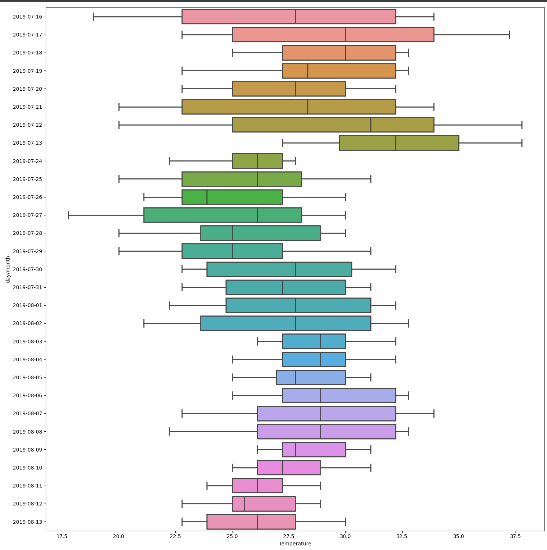
1. Коробчаста діаграма.

plt.figure(figsize=(15, 15))

df2 = df2.reset\_index()

box = sns.boxplot(x='Temperature', y='day/month', orient='h', data=df2, linewidth=2)

plt.show()



*Рис. 8 коробчаста діаграма*

1. Бонус завдання. Написав парсер для отримання даних з сайту у data frame

r = requests.get(

'https://api.weather.com/v1/location/LGAV:9:GR/observations/historical.json?apiKey'

'=6532d6454b8aa370768e63d6ba5a832e&units=e&startDate=20201001&endDate=20201031')

x = r.json()

weather = pd.DataFrame(x['observations'])

# func to clear starting json

def clear\_observation(observation):

res = {}

res["Time"] = [datetime.fromtimestamp(time).strftime('%d/%m/%Y') for time in observation["valid\_time\_gmt"]]

res["Temperature"] = observation["temp"]

res["Dew Point"] = observation["dewPt"]

res["Pressure"] = observation["pressure"]

res["Wind Speed"] = observation["wspd"]

res["Precipitation"] = observation["precip\_hrly"]

res["Humidity"] = observation["rh"]

return res

weather = pd.DataFrame(clear\_observation(weather))

# new final dataframe

final\_weather = weather.groupby('Time').agg({'Temperature': 'mean'})

final\_weather['maxTemp'] = weather.groupby('Time').agg({'Temperature': 'max'})

final\_weather['minTemp'] = weather.groupby('Time').agg({'Temperature': 'min'})

final\_weather['Dew Point'] = weather.groupby('Time').agg({'Dew Point': 'mean'})

final\_weather['maxDew'] = weather.groupby('Time').agg({'Dew Point': 'max'})

final\_weather['minDew'] = weather.groupby('Time').agg({'Dew Point': 'min'})

final\_weather['Humidity'] = weather.groupby('Time').agg({'Humidity': 'mean'})

final\_weather['maxHum'] = weather.groupby('Time').agg({'Humidity': 'max'})

final\_weather['minHum'] = weather.groupby('Time').agg({'Humidity': 'min'})

final\_weather['Wind Speed'] = weather.groupby('Time').agg({'Wind Speed': 'mean'})

final\_weather['maxWind'] = weather.groupby('Time').agg({'Wind Speed': 'max'})

final\_weather['minWind'] = weather.groupby('Time').agg({'Wind Speed': 'min'})

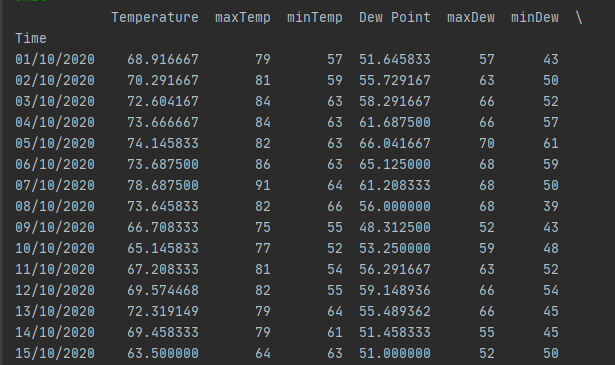
final\_weather['Pressure'] = weather.groupby('Time').agg({'Pressure': 'mean'})

final\_weather['maxPressure'] = weather.groupby('Time').agg({'Pressure': 'max'})

final\_weather['minPressure'] = weather.groupby('Time').agg({'Pressure': 'min'})

final\_weather['Precipitation'] = weather.groupby('Time').agg({'Precipitation': 'min'})

print(final\_weather)



*Рис. 9 Кінцевий data frame*

**Висновок:** В дані лабораторній роботі проаналізував та візуалізував дані з файлу “database.csv”, а також навчився основаб мови програмування Python і бібліотек “ pandas” та “ matplotlib”.