ЛАБОРАТОРНА РОБОТА 3

Тема: Візуалізація даних

Вивчення даних є обов’язковим початковим кроком. Візуалізація даних або графічне дослідження дозволяє краще зрозуміти структуру даних, на етапі очищення даних виявити певний їх дефіцит або неприйнятні значення, виявити викиди, встановити початкові закономірності, кореляції між змінними, виявити кластери даних, тощо. Графічне дослідження також може бути більш зосередженим, спрямованим на конкретні питання, що цікавлять. Графічне дослідження може варіюватися від створення базових графіків до використання таких операцій, як фільтрація та масштабування в інтерактивному режимі, щоб дослідити набір взаємопов’язаних візуалізацій, які включають розширені функції, такі як колір та декілька панелей.

Візуалізація даних в Python підтримується різноманітними бібліотеками. Найстарішою, але й найбільш гнучкою бібліотекою є matplotlib, яка широко використовується. Навколо її побудовано ряд інших бібліотек, які прискорюють генерацію графіків. Seaborn і pandas - це дві бібліотеки, які є обгортками навколо matplotlib. Кожна з них є достатньо корисними для швидкого створення графіків. Однак навіть при їх використанні знання matplotlib допомагає мати більший контроль над остаточним представленням.

Бібліотека ggplot змодельована на основі однаково названого пакета R від Хедлі Вікхема. Він базується на “Граматиці графіки” - терміні, який увів Леланд Вілкінсон для визначення системи теорії побудови та номенклатури. Ефективно вивчити ggplot - означає ознайомитися з цією філософією та технічною мовою планування.

Bokeh є достатньо новою розробкою для створення інтерактивних графіків, інформаційних панелей та програм для передачі даних для веб-браузерів.

Бібліотеки, які необхідні для типових візуалізацій за допомогою Python

```
import os
import calendar
import numpy as np
import pandas as pd
from pandas.plotting import scatter_matrix, parallel_coordinates
import seaborn as sns
from sklearn import preprocessing
import matplotlib.pylab as plt
```

Для візуалізації використано відомі дані про житло у Бостоні, що містять інформацію з книжок перепису в Бостоні, в яких проводиться кілька вимірів: рівень злочинності (crim), відсоток житлових земель для ділянок 25 000 футів^2 (zn), відсоток землі, яку зайнято нероздрібним бізнесом (indus), наявність межі з річкою Чарльз (1 — на межі з річкою, 0 -ні) (chas), концентрація оксиду азоту (кількість часток на 10 мільйонів) (nox), середня кількість кімнат на житло (rm), відсоток житла зайнятого власниками і яке побудовано до 1940 року (age), зважена відстань до п'яти центрів зайнятості в Бостона (dis), індекс доступності до радіальних магістралей (rad), повна ставка податку на нерухомість за 10 000 доларів США (tax), співвідношення учнів до вчителя за містами (ptratio), частка чорношкірих за містом (black), відсоток населення нижчого статусу (lstat), середня вартість будинків, що займаються власниками, у 1000 доларів США (medv).

```
housing_df = pd.read_csv('BostonHousing.csv')
housing_df
```

	ID	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0	1	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	2	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	4	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
3	5	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
4	7	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.60	12.43	22.9
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
328	500	0.17783	0.0	9.69	0	0.585	5.569	73.5	2.3999	6	391	19.2	395.77	15.10	17.5
329	502	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	9.67	22.4
330	503	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9.08	20.6
331	504	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64	23.9
332	506	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90	7.88	11.9

333 rows × 15 columns

Найбільш ефективними у аналізі діаграмами є стовпчасті діаграми, лінійні графіки та діаграми розсіювання. Базові діаграми представляють дані, відображаючи один або два стовпці даних (змінних) одночасно. Це корисно на перших етапах ознайомлення зі структурою даних, кількістю та типами змінних, обсягом та типом відсутніх значень тощо. Характер завдання обробки даних та отримання знання щодо даних впливатимуть на використання базових діаграм. Лінійні графіки найчастіше використовуються для показу часових рядів. Вибір часових рамок для побудови графіку, а також часової шкали повинні залежати від горизонту завдання прогнозування та від характеру даних. Для демонстрації побудови лінійного графіку з часовою віссю використаємо данні залізничної компанії Амтрак у США, яка регулярно збирає дані про поїздки. Файл Amtrak.csv має інформацію про рейси у період із січня 1991 року по березень 2004 року.

```
Amtrak_df = pd.read_csv('Amtrak.csv', squeeze=True)
Amtrak_df
```

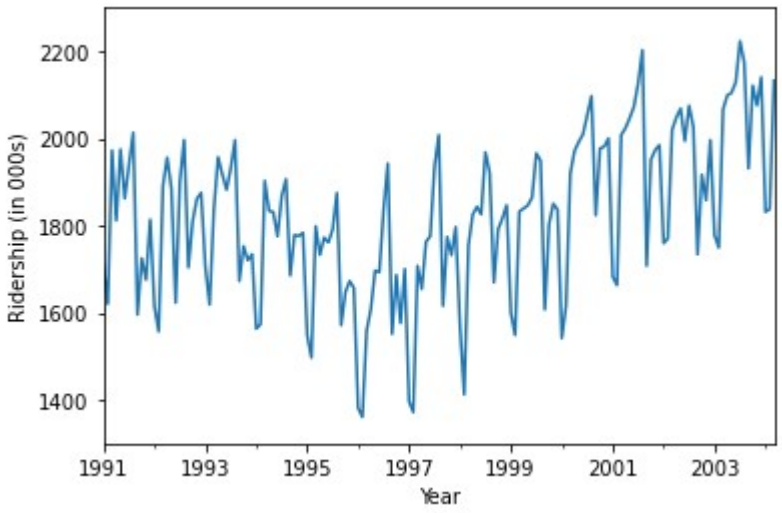
	Month	Ridership
0	01/01/1991	1708.917
1	01/02/1991	1620.586
2	01/03/1991	1972.715
3	01/04/1991	1811.665
4	01/05/1991	1974.964
...	...	...

Для побудови лінійного графіку необхідно отримати часові данні, які у файлі представлено у форматі День / Місяць / Рік. Тому для читання даних з відповідного стовпчика необхідно вказати формат представлення цих даних:

```
Amtrak_df['Date'] = pd.to_datetime(Amtrak_df.Month, format="%d/%m/%Y")
ridership_ts = pd.Series(Amtrak_df.Ridership.values, index=Amtrak_df.Date)
```

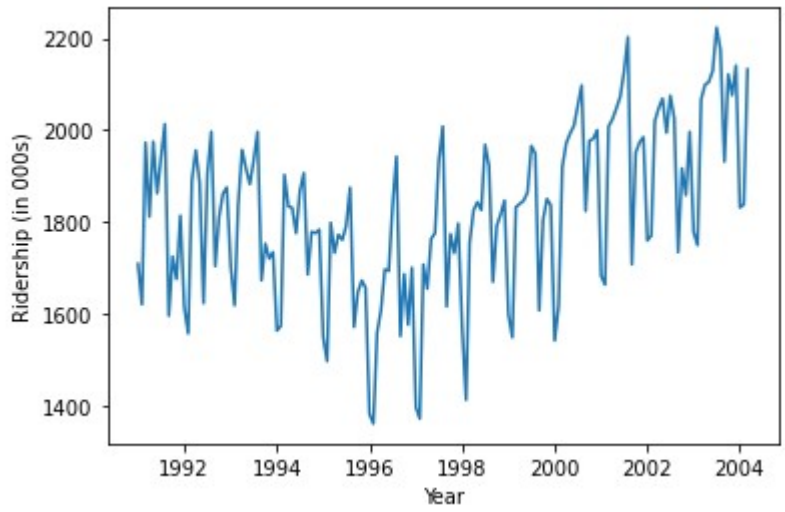
Для побудови лінійного графіку можна використати як інструменти Pandas:

```
## line graph
ridership_ts.plot(ylim=[1300, 2300], legend=False)
plt.xlabel('Year')  # set x-axis label
plt.ylabel('Ridership (in 000s)')  # set y-axis label
```



так і інструменти matplotlib

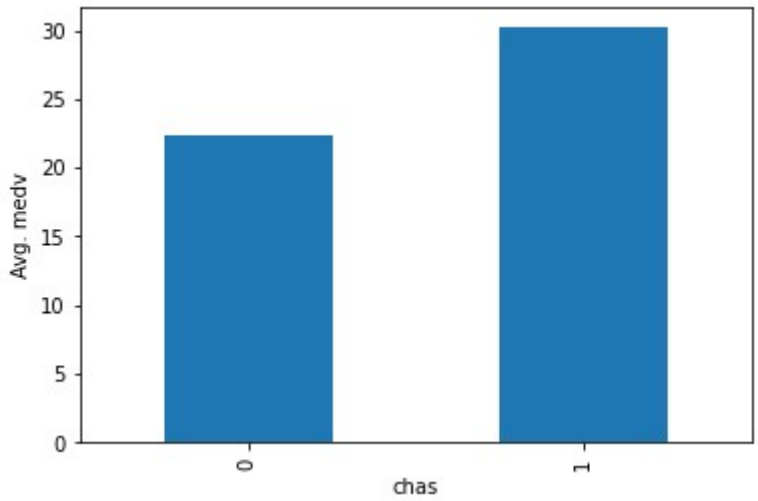
```
## line graph
plt.plot(ridership_ts.index, ridership_ts)
plt.xlabel('Year')  # set x-axis label
plt.ylabel('Ridership (in 000s)')  # set y-axis label
```



Стовпчасті діаграми корисні для порівняння єдиної статистики між групами , наприклад, середньої, кількості, відсотка. Висота смуги (або довжина в горизонтальному дисплеї) представляє

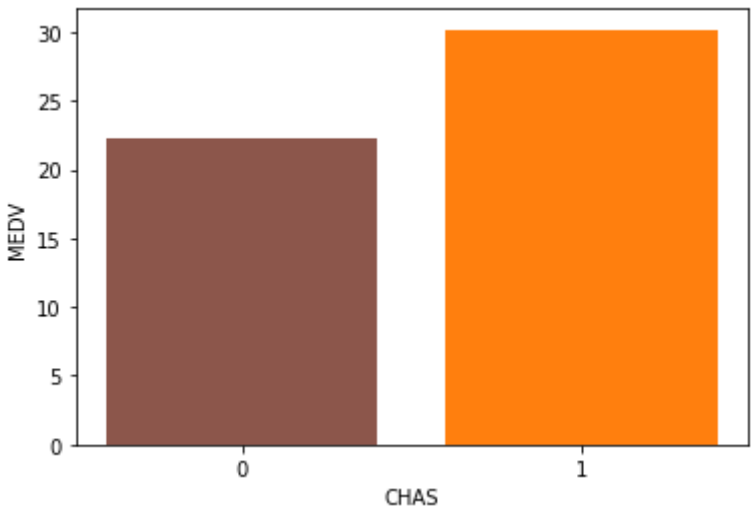
значення статистики, а різні смуги відповідають різним групам. Приклад побудови стовпчастої діаграми за допомогою Pandas:

```
ax = housing_df.groupby('chas').mean().medv.plot(kind='bar')
ax.set_ylabel('Avg. medv')
```



та за допомогою matplotlib:

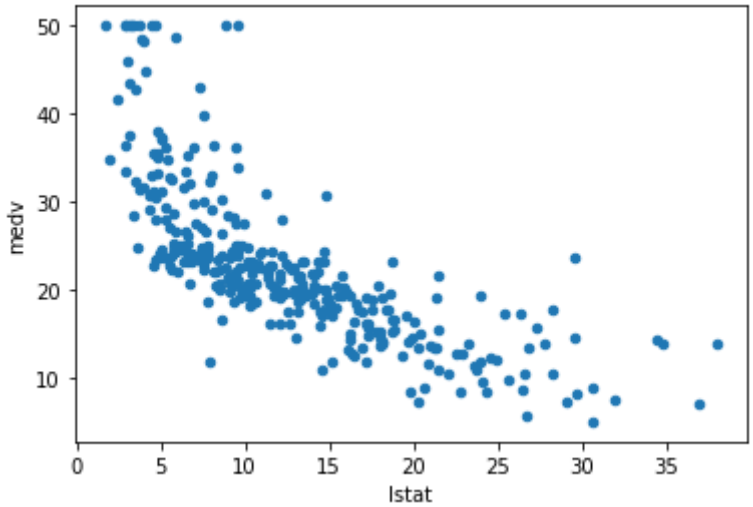
```
dataForPlot = housing_df.groupby('chas').mean().medv
fig, ax = plt.subplots()
ax.bar(dataForPlot.index, dataForPlot, color=['C5', 'C1'])
ax.set_xticks([0, 1])
ax.set_xlabel('CHAS')
ax.set_ylabel('MEDV')
```



На графіку одна смуга відповідає будинкам у Бостоні, які знаходяться поблизу річки Чарльз, а інша - тим, що не знаходяться неї.

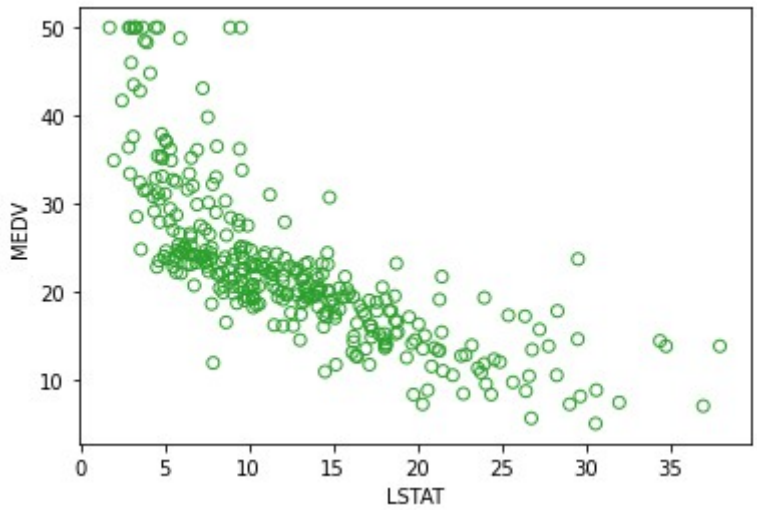
Графік розсіювання допомагає вивчити зв'язок між двома числовими змінними з точки зору перекриття інформації, а також ідентифікувати групи спостережень. Приклад побудови графіку розсіювання за допомогою Pandas:

```
housing_df.plot.scatter(x='lstat', y='medv', legend=False)
```



та за допомогою matplotlib:

```
plt.scatter(housing_df.lstat, housing_df.medv, color='C2', facecolor='none')
plt.xlabel('LSTAT'); plt.ylabel('MEDV')
```



На рисунку зображено діаграму розсіювання MEDV проти LSTAT. Цей графік є важливим для задач передбачення.

Усі три основні діаграми висвітлюють загальну інформацію, таку як середня вартість будинків, а також зміни, що відбуваються у часі (лінійна діаграма), відмінності між підгрупами (стовпчаста діаграма) та відносини між числовими змінними (діаграма розсіювання).

Практичні завдання

1. У заданих файлах розташовано статистичні дані Футбольної Прем’єр Ліги (FPL): успіхи кожного гравця за певний сезон у Лізі. Назва файлу відповідає певному сезону. Вміст файлу поділяється на колонки:
   - first_name — ім’я гравця
   - second_name — прізвище гравця
   - goals_scored - загальна кількість забитих голів за цей сезон
   - assists - загальна кількість гольових передач - присуджується гравцю з команди забиття воріт, який робить остаточний пас до того, як забити гол, включаючи автоголи.
   - total_points - загальна сума балів, зароблених у цьому сезоні
   - minutes - загальна кількість зіграних хвилин цього сезону
   - goals_conceded - загальна кількість голів, пропущених командою, поки гравець був на полі
   - creativity - творчість, оцінює ефективність гравців з точки зору створення можливостей для оцінки голів для інших гравців. Частина індексу ICT
   - influence - вплив, оцінює вплив гравця на матч, враховуючи дії, які можуть прямо чи побічно вплинути на результат матчу. Частина індексу ICT.
   - threat - загроза, вимірює гравців, які, швидше за все, заб'ють голи. Частина індексу ICT
   - bonus - троє найкращих гравців у кожному матчі відповідно до BPS отримають додаткові бонусні бали - 3 бали будуть нараховані гравцеві з найвищою оцінкою, 2 - другому кращому та 1 - третьому.
   - bps - система бонусних балів (BPS) використовує ряд статистичних даних для створення оцінки BPS для кожного гравця. Троє найкращих гравців у кожному матчі отримають бонусні бали.
   - ict_index - статистичний індекс, розроблений спеціально для оцінки гравця як активу FPL, що поєднує показники впливу, творчості та загроз.
   - clean_sheets - загальна кількість чистих аркушів - присуджується гравцям, які не пропустили гол і зіграли принаймні 60 хвилин.
   - red_cards - кількість отриманих за сезон червоних карток.
   - yellow_cards - кількість отриманих за сезон жовтих карток.
2. Використовуючи матеріал попередньої лабораторної роботи проілюструйте графіками отримані результати та їх динаміку.
3. Підготуйте звіт з виконання практичних завдань.