ЛАБОРАТОРНА РОБОТА 2

Тема: Підготування даних

Підготування даних є важливим етапом їх аналізу. До основних видів підготування даних можна віднести:

- Завантаження
- Збирання
- Злиття (merge)
- Конкатенацію (concatenating)
- Комбінування (combining)
- Зміну
- Видалення

Дані із об’єктів pandas можна збирати декількома методами.
Злиття — функція pandas.merge() з'єднує рядки у Dataframe на основі одного або декількох ключів. Цей метод широко використовується у база даних з SQL.
Конкатенація — функція pandas.concat() конкатенує об’єкти за віссю.
Комбінування — функція pandas.DataFrame.combine_first() є методом, який дозволяє з’єднати дані, що перетинаються, для заповнення відсутніх значень у структурі за допомогою даних з іншої структури.

Операція злиття об'єднує дані на основі одного або декількох ключів. Наприклад:

```
import numpy as np
import pandas as pd
frame1 = pd.DataFrame({'id':['ball','pencil','pen','mug','ashtray'],'price': [12.33,11.44,33.21,13.23,33.62]})
frame2 = pd.DataFrame({'id':['pencil','pencil','ball','pen'],'color': ['white','red','red','black']})
```

```
Ввод [2]: frame1
```

Out[2]:

	id	price
0	ball	12.33
1	pencil	11.44
2	pen	33.21
3	mug	13.23
4	ashtray	33.62

```
Ввод [3]: frame2
```

Out[3]:

	id	color
0	pencil	white
1	pencil	red
2	ball	red
3	pen	black

злиття за допомогою merge():

```
Ввод [4]: pd.merge(frame1,frame2)
```

Out[4]:

	id	price	color
0	ball	12.33	red
1	pencil	11.44	white
2	pencil	11.44	red
3	pen	33.21	black

У наведеному випадку злиття виконано за загальним id, який сприймався за замовчуванням. Функція merge() дозволяє явно вказати за якою колонкою буде виконано злиття:

```
Ввод [5]: pd.merge(frame1,frame2,on='id')
```

Out[5]:

	id	price	color
0	ball	12.33	red
1	pencil	11.44	white
2	pencil	11.44	red
3	pen	33.21	black

Дещо змінимо данні для демонстрації цих можливостей:

```
Ввод [7]: frame1 = pd.DataFrame({'id':['ball','pencil','pen','mug','ashtray'],
                                 'color': ['white','red','red','black','green'],
                                 'brand': ['OMG','ABC','ABC','POD','POD']})
          frame2 = pd.DataFrame({'id':['pencil','pencil','ball','pen'],
                                 'brand': ['OMG','POD','ABC','POD']})
          pd.merge(frame1,frame2,on='id')
```

Out[7]:

	id	color	brand_x	brand_y
0	ball	white	OMG	ABC
1	pencil	red	ABC	OMG
2	pencil	red	ABC	POD
3	pen	red	ABC	POD

злиття за колонкою brand:

```
Ввод [8]: pd.merge(frame1,frame2,on='brand')
```

Out[8]:

	id_x	color	brand	id_y
0	ball	white	OMG	pencil
1	pencil	red	ABC	ball
2	pen	red	ABC	ball
3	mug	black	POD	pencil
4	mug	black	POD	pen
5	ashtray	green	POD	pencil
6	ashtray	green	POD	pen

За допомогою left_on та right_on можна визначити ключові колонки для першого та другого об’єктів Dataframe. Це застосовується у тому випадку, коли два об’єкти мають різне найменування для однотипних даних. Наприклад:

```
frame1 = pd.DataFrame({'id':['ball','pencil','pen','mug','ashtray'],
                       'color': ['white','red','red','black','green'],
                       'brand': ['OMG','ABC','ABC','POD','POD']})
frame2 = pd.DataFrame({'sid':['pencil','pencil','ball','pen'],
                       'brand': ['OMG','POD','ABC','POD']})
```

У першому об’єкті використовується id, а другому — sid. Злиття можна виконати за допомогою наступного рядка:

```
pd.merge(frame1, frame2, left_on='id', right_on='sid')
```

Out[12]:

	id	color	brand_x	sid	brand_y
0	ball	white	OMG	ball	ABC
1	pencil	red	ABC	pencil	OMG
2	pencil	red	ABC	pencil	POD
3	pen	red	ABC	pen	POD

За замовчуванням функція merge() виконує inner join (внутрішнє з'єднання). Ключ у фінальному об'єкті є результатом перетинання. Іншими варіантами є: left join, right join та outer join — з'єднання ліворуч, праворуч та зовнішнє. Зовнішнє об'єднує усі ключі, при цьому комбінуються праве та ліве з'єднання. Для обрання типу необхідно використовувати параметр how. Наприклад:

```
frame1 = pd.DataFrame({'id':['ball','pencil','pen','mug','ashtray'],
                       'color': ['white','red','red','black','green'],
                       'brand': ['OMG','ABC','ABC','POD','POD']})
frame2 = pd.DataFrame({'id':['pencil','pencil','ball','pen'],
                       'brand': ['OMG','POD','ABC','POD']})
```

```
pd.merge(frame1,frame2,on='id')
```

	id	color	brand_x	brand_y
0	ball	white	OMG	ABC
1	pencil	red	ABC	OMG
2	pencil	red	ABC	POD
3	pen	red	ABC	POD

```
pd.merge(frame1,frame2,on='id',how='outer')
```

	id	color	brand_x	brand_y
0	ball	white	OMG	ABC
1	pencil	red	ABC	OMG
2	pencil	red	ABC	POD
3	pen	red	ABC	POD
4	mug	black	POD	NaN
5	ashtray	green	POD	NaN

```
pd.merge(frame1,frame2,on='id',how='left')
```

	id	color	brand_x	brand_y
0	ball	white	OMG	ABC
1	pencil	red	ABC	OMG
2	pencil	red	ABC	POD
3	pen	red	ABC	POD
4	mug	black	POD	NaN
5	ashtray	green	POD	NaN

```
pd.merge(frame1,frame2,on='id',how='right')
```

	id	color	brand_x	brand_y
0	pencil	red	ABC	OMG
1	pencil	red	ABC	POD
2	ball	white	OMG	ABC
3	pen	red	ABC	POD

```
pd.merge(frame1,frame2,on=['id','brand'],how='outer')
```

	id	color	brand
0	ball	white	OMG
1	pencil	red	ABC
2	pen	red	ABC
3	mug	black	POD
4	ashtray	green	POD
5	pencil	NaN	OMG
6	pencil	NaN	POD
7	ball	NaN	ABC
8	pen	NaN	POD

У деяких випадках замість використання колонок об'єкта Dataframe в якості ключів для цих цілей можна задіяти індекси. Потім для вибору конкретних індексів потрібно задати значення True для left_join або right_join. Вони можуть бути використані і разом.

```
pd.merge(frame1,frame2,right_index=True, left_index=True)
```

	id_x	color	brand_x	id_y	brand_y
0	ball	white	OMG	pencil	OMG
1	pencil	red	ABC	pencil	POD
2	pen	red	ABC	ball	ABC
3	mug	black	POD	pen	POD

Зрозуміло, що такого роду злиття слід використовувати з розумінням результату. У наведеному прикладі зверніть увагу на рядок з індексом 1 — один предмет, але два різні бренди та одна додаткова ознака.

Ще один тип об'єднання даних — конкатенація.

```
frame1 = pd.DataFrame({'id':['ball','pencil','pen','mug','ashtray'],
                       'color': ['white','red','red','black','green'],
                       'brand': ['OMG','ABC','ABC','POD','POD']})
frame2 = pd.DataFrame({'id':['pencil','pencil','ball','pen'],
                       'brand': ['OMG','POD','ABC','POD']})
```

```
pd.concat([frame1,frame2])
```

Out[37]:

	id	color	brand
0	ball	white	OMG
1	pencil	red	ABC
2	pen	red	ABC
3	mug	black	POD
4	ashtray	green	POD
0	pencil	NaN	OMG
1	pencil	NaN	POD
2	ball	NaN	ABC
3	pen	NaN	POD

```
pd.concat([frame1, frame2], axis=1)
```

	id	color	brand	id	brand
0	ball	white	OMG	pencil	OMG
1	pencil	red	ABC	pencil	POD
2	pen	red	ABC	ball	ABC
3	mug	black	POD	pen	POD
4	ashtray	green	POD	NaN	NaN

Якщо у даних виконано примусове індексування:

```
frame1 = pd.DataFrame({'id':['ball','pencil','pen','mug','ashtray'],
                       'color': ['white','red','red','black','green'],
                       'brand': ['OMG','ABC','ABC','POD','POD']},index=[1,2,3,4,5])
frame2 = pd.DataFrame({'id':['pencil','pencil','ball','pen'],
                       'brand': ['OMG','POD','ABC','POD']},index=[6,7,8,9])
```

```
frame1
```

	id	color	brand
1	ball	white	OMG
2	pencil	red	ABC
3	pen	red	ABC
4	mug	black	POD
5	ashtray	green	POD

```
frame2
```

	id	brand
6	pencil	OMG
7	pencil	POD
8	ball	ABC
9	pen	POD

то конкатенація буде виглядати наступним чином:

```
pd.concat([frame1, frame2])
```

	id	color	brand
1	ball	white	OMG
2	pencil	red	ABC
3	pen	red	ABC
4	mug	black	POD
5	ashtray	green	POD
6	pencil	NaN	OMG
7	pencil	NaN	POD
8	ball	NaN	ABC
9	pen	NaN	POD

```
pd.concat([frame1, frame2], axis=1)
```

	id	color	brand	id	brand
1	ball	white	OMG	NaN	NaN
2	pencil	red	ABC	NaN	NaN
3	pen	red	ABC	NaN	NaN
4	mug	black	POD	NaN	NaN
5	ashtray	green	POD	NaN	NaN
6	NaN	NaN	NaN	pencil	OMG
7	NaN	NaN	NaN	pencil	POD
8	NaN	NaN	NaN	ball	ABC
9	NaN	NaN	NaN	pen	POD

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. Завантажте файли у відповідні DataFrame.
3. Виконайте пошук найкращого та найгіршого гравця для кожного сезону за такими показниками як:
   - кількість забитих голів;
   - загальна кількість гольових передач;
   - загальна сума балів;
   - кількість зіграних хвилин;
   - загальні кількість голів, пропущених командою, поки гравець був на полі;

- творчість;
- вплив;
- загроза;
- загальна кількість чистих аркушів.

4. Знайдіть найгірших гравців у кожному сезоні за отриманими жовтими та червоними картками.
5. Знайдіть трьох найкращих гравців у кожному сезоні за такими двома показниками:
   - загальна кількість забитих голів та кількості гольових передач;
   - загальна кількість забитих голів та кількість чистих аркушів.
6. Знайдіть трьох найгірших гравців у кожному сезоні за такими двома показниками:
   - загальна кількість отриманих жовтих карток та загальна кількість голів, пропущених командою, поки гравець був на полі;
   - загальна кількість отриманих червоних карток та загальна кількість голів, пропущених командою, поки гравець був на полі.
7. Виконайте злиття даних у DataFrame за іменами гравців.
8. Виконайте злиття даних у DataFrame за кількістю балів.
9. Виконайте злиття даних у DataFrame за кількістю забитих голів.
10. Виконайте злиття даних у DataFrame за кількістю гольових передач.
11. Виконайте злиття даних у один файл з додаванням колонки сезону.
12. Визначить гравців, які грали усі сезони.
13. Визначить трьох найкращих гравців за усі сезони за такими показниками як:
    - кількість забитих голів;
    - загальна кількість гольових передач;
    - загальна сума балів;
    - кількість зіграних хвилин;
    - загальна кількість чистих аркушів.
14. Визначить трьох найгірших гравців за усі сезони за такими показниками як:
    - загальні кількість голів, пропущених командою, поки гравець був на полі;
    - кількість отриманих жовтих карток;
    - кількість отриманих червоних карток.
15. Визначить динаміку трьох найкращих та трьох найгірших гравців на протязі усіх сезонів.
16. Підготуйте звіт з виконання практичних завдань.