Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Лабораторна робота №3

Аналіз даних з використанням мови Python

Тема: Структури даних Pandas

Варіант: 1

Виконав

студент групи ІП-11:

Панченко С. В.

Перевірив:

Тимофєєва Ю. С

Київ 2023

ЗМІСТ

1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Ознайомитись з основними структурами даних бібліотеки Pandas: Series DataFrame, операціями над ними. Навчитись використовувати групування.

2 ЗАВДАННЯ

Створити програму, яка за даними файлу відповідно до варіантів лабораторної №2, виконує наступні завдання:

1. Виділити один зі стовпців (на вибір) з файлу як об’єкт Series, виділити з нього підмасив. Задати назви індексів цього об’єкту. Виділити підмасиви за допомогою прямої та непрямої індексацій.
2. До об’єкту DataFrame, в який записано вміст файлу, додати новий стовпець, що є результатом операцій над іншими стовпцями. Також продемонструвати додавання та видалення рядків, видалення стовпців.
3. Встановити один зі стовпців індексом. Визначити основні статистичні характеристики та типи даних всіх стовпців. Змінити тип даних для одного з стовпців. Згрупувати дані за одним зі стовпців, застосувати кілька агрегуючих функцій, виділити підмасив за певними ознаками.
4. Створити декілька власних об’єктів DataFrame за такою ж тематикою, що й файл. Наприклад, якщо тема файлу – жаби, можна створити об’єкти, що містять розміри жаб, вагу, стать, кількість особин в популяції і т.д. Використати описані в теоретичних відомостях параметри методів merge та concat для різних видів злиття та об’єднання даних цих об’єктів.

3 ВИКОНАННЯ

3.1 Виділити один зі стовпців (на вибір) з файлу як об’єкт Series, виділити з нього підмасив. Задати назви індексів цього об’єкту. Виділити підмасиви за допомогою прямої та непрямої індексацій

Імпортуємо модуль pandas та завантажимо датасет Birthweight.csv. Виведемо перші десять рядків за допомогою методу head.

```
In [156…
import pandas as pd
import numpy as np
df = pd.read_csv('data/Birthweight.csv')
df.head(10)
```

Out[156]:

	ID	Length	Birthweight	Headcirc	Gestation	smoker	mage	mnocig	mheight	mppwt	fage	fedyrs	fnocig
0	1360	56	4.55	34	44	0	20	0	162	57	23	10	35
1	1016	53	4.32	36	40	0	19	0	171	62	19	12	0
2	462	58	4.10	39	41	0	35	0	172	58	31	16	25
3	1187	53	4.07	38	44	0	20	0	174	68	26	14	25
4	553	54	3.94	37	42	0	24	0	175	66	30	12	0
5	1636	51	3.93	38	38	0	29	0	165	61	31	16	0
6	820	52	3.77	34	40	0	24	0	157	50	31	16	0
7	1191	53	3.65	33	42	0	21	0	165	61	21	10	25
8	1081	54	3.63	38	38	0	18	0	172	50	20	12	7
9	822	50	3.42	35	38	0	20	0	157	48	22	14	0

Рисунок 3.1 - Завантаження стовпця та виведення перших десяти рядків

Виділимо стовпець як об'єкт типу Series.

```
In [157…
bw = df.Birthweight
bw.__class__
```

```
Out[157]: pandas.core.series.Series
```

Рисунок 3.2 - Виділення стовпця як об'єкта типу Series

Виділимо підмасив з даного стовпця.

```
In [158…
sub = bw[:3]
```

Рисунок 3.3 - Виділення підмасиву

Задамо назви індексів даному підмасиву.

```
In [159…  sub.index = ['I', 'II', 'III']
          sub

Out[159]: I      4.55
          II     4.32
          III    4.10
          Name: Birthweight, dtype: float64
```

Рисунок 3.4 - Задання назви індексів

Виділимо підмасиви за допомогою прямої індексації. Використаємо атрибут loc та візьмемо елементи, індекси яких знаходяться до елемента з назвою індексу римського позначення двійки.

```
In [160…  sub2 = sub.loc[:'II']
          sub2

Out[160]: I     4.55
```

Рисунок 3.5 - Виділення підмасиву за допомогою прямої індексації

Виділимо підмасиви за допомогою непрямої індексації. Використаємо атрибут iloc та візьмемо елементи, індекси яких знаходяться до елементу індекс якого має порядковий номер 2.

```
In [161…  sub3 = sub.iloc[:2]
          sub3

Out[161]: I     4.55
          II    4.32
          Name: Birthweight, dtype: float64
```

Рисунок 3.6 - Виділення підмасиву за допомогою непрямої індексації

3.2 До об’єкту DataFrame, в який записано вміст файлу, додати новий стовпець, що є результатом операцій над іншими стовпцями. Також продемонструвати додавання та видалення рядків, видалення стовпців

Додамо новий стовпець, що покаже відношення віку батька до віку матері.

```
In [162… df['fmratio'] = df['fage'] / df['mage']
         df['fmratio']

Out[162]: 0     1.150000
          1     1.000000
          2     0.885714
          3     1.300000
          4     1.250000
          5     1.068966
          6     1.291667
          7     1.000000
          8     1.111111
          9     1.100000
          10    1.370370
          11    1.208333
          12    1.129032
          13    1.190476
          14    1.034483
          15    1.392857
          16    1.000000
          17    0.961538
          18    1.210526
          19    1.083333
          20    1.200000
          21    1.090909
          22    1.052632
          23    1.045455
          24    1.000000
          25    0.902439
          26    1.000000
          27    1.187500
          28    1.322581
          29    1.370370
          30    1.266667
          31    1.391304
          32    1.142857
          33    1.071429
          34    1.333333
          35    1.243243
          36    1.153846
          37    1.000000
          38    1.032258
          39    1.148148
          40    1.150000
          41    0.837838
          Name: fmratio, dtype: float64
```

Рисунок 3.7 - Стовпець, що позначає відношення віку батька до віку матері

Для зручності виділимо частину датафрейму та будемо працювати з ним.

```
In [163…
sub = df.loc[:3, :'Headcirc'].copy()
sub
```

Out[163]:

	ID	Length	Birthweight	Headcirc
0	1360	56	4.55	34
1	1016	53	4.32	36
2	462	58	4.10	39
3	1187	53	4.07	38

Рисунок 3.8 - Виділення чатини датафрейму

Додамо рядки до датафрейму за допомогою метода concat. За допомогою параметру за замовчуванням з'єднаємо два датафрейми, переіндексувавши новоутворений.

```
In [164…
new_row = dict(ID=[1, 2], Length=[1, 2],
               Birthweight=[55, 45], Headcirc=[1, 1])
sub = pd.concat([sub, pd.DataFrame(new_row)], axis=0, ignore_index=True)
sub
```

Out[164]:

	ID	Length	Birthweight	Headcirc
0	1360	56	4.55	34
1	1016	53	4.32	36
2	462	58	4.10	39
3	1187	53	4.07	38
4	1	1	55.00	1
5	2	2	45.00	1

Рисунок 3.9 - Додавання рядків функцією concat

Додамо рядки до датафрейму за допомогою методу append.

```
In [165…
new_row = dict(ID=[99, 99], Length=[83, 11],
               Birthweight=[52, 35], Headcirc=[4, 6])
sub = sub.append(pd.DataFrame(new_row), ignore_index=True)
sub
```

```
/tmp/ipykernel_7289/1953853348.py:3: FutureWarning: The frame.append method is deprecated
and will be removed from pandas in a future version. Use pandas.concat instead.
  sub = sub.append(pd.DataFrame(new_row), ignore_index=True)
```

Out[165]:

	ID	Length	Birthweight	Headcirc
0	1360	56	4.55	34
1	1016	53	4.32	36
2	462	58	4.10	39
3	1187	53	4.07	38
4	1	1	55.00	1
5	2	2	45.00	1
6	99	83	52.00	4
7	99	11	35.00	6

Рисунок 3.10 - Додавання рядків методом append

Видалимо чотири рядки з датафрейму за допомогою методу drop.

```
In [166…  sub.drop(index=[4, 5, 6, 7], inplace=True)
          sub
```

Out[166]:

	ID	Length	Birthweight	Headcirc
0	1360	56	4.55	34
1	1016	53	4.32	36
2	462	58	4.10	39
3	1187	53	4.07	38

Рисунок 3.11 - Видалення рядків

Додамо новий стовпець до датафрейму та проініціалізуємо його.

```
In [167…  sub['new_col'] = 4
          sub
```

Out[167]:

	ID	Length	Birthweight	Headcirc	new_col
0	1360	56	4.55	34	4
1	1016	53	4.32	36	4
2	462	58	4.10	39	4
3	1187	53	4.07	38	4

Рисунок 3.12 - Додавання стовпця за допомогою передачі індексу

Додамо новий стовпець за допомогою методу concat.

```
In [168…  new_col = dict(concat_col=[777, 888, 999, 000])
          sub = pd.concat([sub, pd.DataFrame(new_col)], axis=1)
          sub
```

Out[168]:

	ID	Length	Birthweight	Headcirc	new_col	concat_col
0	1360	56	4.55	34	4	777
1	1016	53	4.32	36	4	888
2	462	58	4.10	39	4	999
3	1187	53	4.07	38	4	0

Рисунок 3.13 - Додавання стовпця за допомогою методу concat

Видалимо новостворені стовпці за допомогою методу drop.

```
In [169…  sub.drop(['new_col', 'concat_col'], axis=1, inplace=True)
          sub
```

Out[169]:

	ID	Length	Birthweight	Headcirc
0	1360	56	4.55	34
1	1016	53	4.32	36
2	462	58	4.10	39
3	1187	53	4.07	38

Рисунок 3.14 - Видалення стовпців методом drop

3.3 Встановити один зі стовпців індексом. Визначити основні статистичні характеристики та типи даних всіх стовпців. Змінити тип даних для одного з стовпців. Згрупувати дані за одним зі стовпців, застосувати кілька агрегуючих функцій, виділити підмасив за певними ознаками.

Встановимо стовпець ID індексом датафрейму за допомогою методу set_index.

```
In [170…  df.set_index('ID', inplace=True)
          df.head()
```

Out[170]:

ID	Length	Birthweight	Headcirc	Gestation	smoker	mage	mnocig	mheight	mppwt	fage	fedyrs	fnocig	fl
1360	56	4.55	34	44	0	20	0	162	57	23	10	35	
1016	53	4.32	36	40	0	19	0	171	62	19	12	0	
462	58	4.10	39	41	0	35	0	172	58	31	16	25	
1187	53	4.07	38	44	0	20	0	174	68	26	14	25	
553	54	3.94	37	42	0	24	0	175	66	30	12	0	

Рисунок 3.15 - Встановлення індексу методом set_index

Виведемо основні статистичні характеристики методом describe.

```
In [171… df.describe()
```

Out[171]:

	Length	Birthweight	Headcirc	Gestation	smoker	mage	mnocig	mheight	mppwt	
count	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.000000	42.0
mean	51.333333	3.312857	34.595238	39.190476	0.523810	25.547619	9.428571	164.452381	57.500000	28.9
std	2.935624	0.603895	2.399792	2.643336	0.505487	5.666342	12.511737	6.504041	7.198408	6.8
min	43.000000	1.920000	30.000000	33.000000	0.000000	18.000000	0.000000	149.000000	45.000000	19.0
25%	50.000000	2.940000	33.000000	38.000000	0.000000	20.250000	0.000000	161.000000	52.250000	23.0
50%	52.000000	3.295000	34.000000	39.500000	1.000000	24.000000	4.500000	164.500000	57.000000	29.5
75%	53.000000	3.647500	36.000000	41.000000	1.000000	29.000000	15.750000	169.500000	62.000000	32.0
max	58.000000	4.570000	39.000000	45.000000	1.000000	41.000000	50.000000	181.000000	78.000000	46.0

Рисунок 3.16 - Визначення основних статистичних характеристик

Виведемо типи даних всіх стовпців за допомогою атрибуту dtypes.

```
In [172… df.dtypes
```

```
Out[172]: Length           int64
          Birthweight    float64
          Headcirc         int64
          Gestation        int64
          smoker           int64
          mage             int64
          mnocig           int64
          mheight          int64
          mppwt            int64
          fage             int64
          fedyrs           int64
          fnocig           int64
          fheight          int64
          lowbwt           int64
          mage35           int64
          fmratio        float64
          dtype: object
```

Рисунок 3.17 - Виведення даних стовпців за допомогою атрибуту dtypes

Змінимо тип даних для стовпця Headcirc з int64 до float64 за допомогою методу astype.

```
In [173… df['Headcirc'] = df['Headcirc'].astype('float64')
         df.Headcirc.dtype
```

```
Out[173]: dtype('float64')
```

Рисунок 3.18 - Зміна типу даних стовпця методом astype

Згрупуємо дані за стовпцем 'smoker' та застосуємо на стовпці 'Birthweight'

функцію np.mean, а на 'Gestation' np.median.

```
In [174… df.groupby('smoker').agg({'Birthweight': np.mean, 'Gestation': np.median})
```

Out[174]:

smoker	Birthweight	Gestation
0	3.509500	40.0
1	3.134091	39.0

Рисунок 3.19 - Групування даних та застосування агрегуючих функцій

Виділимо підмасив за певними ознаками, наприклад, матері, що палять, та страші 30.

```
In [176… df[(df['smoker'] == 1) & (df['mage'] >= 30)]
```

Out[176]:

ID	Length	Birthweight	Headcirc	Gestation	smoker	mage	mnocig	mheight	mppwt	fage	fedyrs	fnocig	fl
300	46	2.05	32.0	35	1	41	7	166	57	37	14	25	
1764	58	4.57	39.0	41	1	32	12	173	70	38	14	25	
532	53	3.59	34.0	40	1	31	12	163	49	41	12	50	
1023	52	3.00	35.0	38	1	30	12	165	64	38	14	50	
272	52	3.86	36.0	39	1	30	25	170	78	40	16	50	
27	53	3.55	37.0	41	1	37	25	161	66	46	16	0	
1369	49	3.18	34.0	38	1	31	25	162	57	32	16	50	
1272	53	2.75	32.0	40	1	37	50	168	61	31	16	0	

Рисунок 3.20 - Виділення підмасиву за ознаками

3.4 Створити декілька власних об’єктів DataFrame за такою ж тематикою, що й файл. Наприклад, якщо тема файлу – жаби, можна створити об’єкти, що містять розміри жаб, вагу, стать, кількість особин в популяції і т.д. Використати описані в теоретичних відомостях параметри методів merge та concat для різних видів злиття та об’єднання даних цих об’єктів

Створимо декілька датафреймів: mothers - містить id матері, її ім'я; smokers - містить id матері та поле smoker, чи палить вона; children - містить id дитини, її вагу, зріст; mothers_children - містить поле child_id та mother_id, що позначає

родинний зв'язок.

```
In [182…
mothers = pd.DataFrame(dict(id=[1, 2, 3, 4], mother_name=[
    'Kate Smith', 'Daisy Piligrim', 'Sam McClusky', 'Susan Vesper']))
smokers = pd.DataFrame(dict(id=[1, 2, 3, 4], smoker=[1, 0, 0, 1]))
children = pd.DataFrame(dict(id=[1, 2, 3, 4, 5, 6, 7],
                             weight=[3, 4, 3, 3, 3, 2, 5],
                             height=[55, 47, 65, 53, 65, 48, 49]))
mothers_children = pd.DataFrame(dict(child_id=[1, 2, 3, 4, 5, 6, 7],
                        mother_id=[1, 1, 1, 2, 3, 4, 4]))
m_smokers = pd.merge(mothers, smokers, left_on='id', right_on='id')
m_c =  pd.merge(children, mothers_children,
                left_on='id', right_on='child_id').drop('id', axis=1)
pd.merge(m_smokers, m_c, left_on='id',
         right_on='mother_id').drop('id', axis=1)
```

Out[182]:

	mother_name	smoker	weight	height	child_id	mother_id
0	Kate Smith	1	3	55	1	1
1	Kate Smith	1	4	47	2	1
2	Kate Smith	1	3	65	3	1
3	Daisy Piligrim	0	3	53	4	2
4	Sam McClusky	0	3	65	5	3
5	Susan Vesper	1	2	48	6	4
6	Susan Vesper	1	5	49	7	4

Рисунок 3.21 - Застосування функції merge для join-операцій

4 ВИСНОВОК

Під час виконання даної лабораторної роботи я ознайомився з основними структурами даних бібліотеки Pandas: Series DataFrame, операціями над ними.

У першому завданні виділив підмасиви за допомогою прямої та непрямої індексації — атрибутами loc та iloc відповідно. До того ж задав індекси методом set_index.

У другому завданні продемонстрував роботу з рядками та стовпцями: додавання(concat, append) і видалення(drop).

У третьому завданні встановив за індекс стовпець “ID”, визначив основні характеристики методом describe, застосував агрегуючи функції(agg).

У четвертому завданні створив декілька датафреймів і застосував на них аналог SQL-join метод merge.