## **Задание**

Создайте фрейм с именами колонок 'col\_1', 'col\_2', 'col\_3' и индексами по алфавиту ('A', 'B', 'C', ... или 'A', 'Б', 'В', ...).

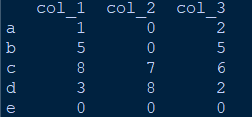
import pandas as pd

import numpy as np

arr = np.random.randint(0, 10, size=(5, 3))

arr1=pd.DataFrame(data=arr,index=['a','b','c','d','e'],columns=['col\_1','col\_2','col\_3'])

print(arr1)



## **Задание**

Выведите часть фрейма со второго ряда по девятый (индексы с 'C' по 'I') и только feature1 и feature3:

import pandas as pd

import numpy as np

data = {

'feature1': np.arange(15),

'feature2': np.linspace(0, 10, 15),

'feature3': 'string',

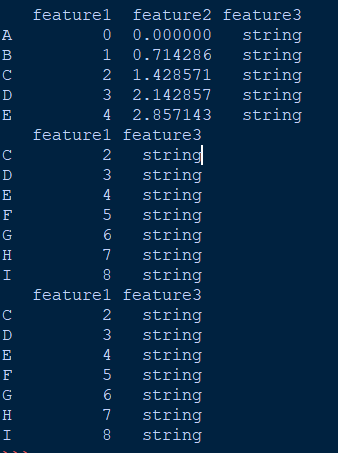
}

df = pd.DataFrame(data, index=list('ABCDEFGHIJKLMNO'))

print(df.head())

print(df.iloc[2:9,0:3:2])

print(df.loc['C':'I',['feature1','feature3']])



## **Задание**

Выберите записи во фрейме, которые по признаку x1 имеют значение больше 3, а по признаку x3 - меньше 8.

import pandas as pd

import numpy as np

df = pd.DataFrame(

data=np.random.randint(0, 10, size=(15, 3)),

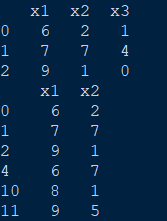
columns=['x1', 'x2', 'x3']

)

print(df.head(3))

mask=(df['x1']>3)&(df['x3']<8)

print(df[mask].loc[:,('x1','x2')])



## **Задание**

С помощью метода DataFrame.query() получите записи, которые соответсвуют условию сравнения колонок IMG_256:

import pandas as pd

import numpy as np

df = pd.DataFrame(

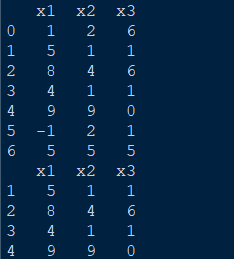
[[1, 2, 6], [5, 1, 1], [8, 4, 6], [4, 1, 1], [9, 9, 0], [-1, 2, 1], [5, 5, 5]],

columns = ['x1', 'x2', 'x3']

)

print(df)

print(df.query('x1>x3'))



### **Задание**

Переименуйте индексы и колонки во фрейме

import pandas as pd

import numpy as np

df = pd.DataFrame(

data=np.random.randint(0, 10, size=(5, 3)),

index=['A', 'B', 'C', 'D', 'E'],

columns=['x1', 'x2', 'x3']

)

df.rename(columns={

'x1':'new1',

'x3':'new3'

},

index={

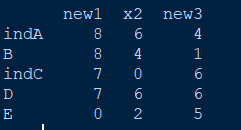
'A':'indA',

'C':'indC'

},

inplace=True)

print(df)



### **Задание**

Отфильруйте записи по строке, содержащейся в имени индекса (аргумент like):

import pandas as pd

import numpy as np

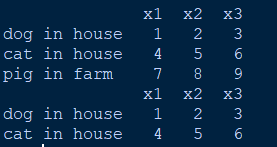
df = pd.DataFrame(np.array(([1, 2, 3], [4, 5, 6], [7, 8, 9])),

index=['dog in house', 'cat in house', 'pig in farm'],

columns=['x1', 'x2', 'x3'])

print(df)

print(df.filter(like='house',axis='index'))



### **Задание**

Отсортируйте данные:

import pandas as pd

import numpy as np

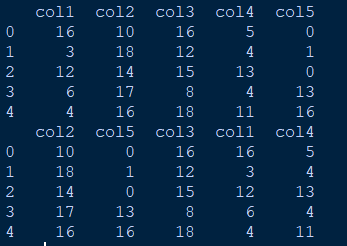
df = pd.DataFrame(

np.random.randint(0, 20, size=(5, 5)),

columns=['col1', 'col2', 'col3', 'col4', 'col5'])

print(df)

print(df.sort\_values(by=3,axis=1,ascending=False))



### **Задание**

Разделите показатель качества в диапазоне [0,100] на три группы: плохое [0-20], среднее(20-45], хорошее (45,85], отличное (85,100]. Выясните, что происходит со значениями, которые не попадают в диапазоны категорий:

import pandas as pd

import numpy as np

df = pd.DataFrame({

'quality': [ 7, 67, 21, 38, 2, -2, 79, 22, 5, 31, 77, 72, 23, 64, 99]

})

print(df.head())

df['quality\_cat'] = pd.cut(

x=df['quality'],

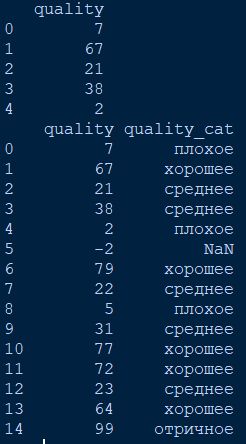
bins=[0, 20, 45, 85, 100],

labels=['плохое', 'среднее', 'хорошее','отричное'],

include\_lowest=True

)

print(df)



### **Задание**

Произведите агрегацию средним по рядам

import pandas as pd

import numpy as np

df = pd.DataFrame(

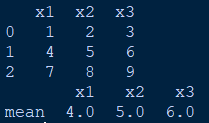
([1, 2, 3], [4, 5, 6], [7, 8, 9]),

columns=['x1', 'x2', 'x3'])

df

print(df)

print(df.aggregate(['mean'], axis='rows'))



### **Задание**

Сгруппируйте данные по признаку x2 и найдите медиану и максимальное значения по признакам x1, x3:

import pandas as pd

import numpy as np

df = pd.DataFrame({

'x1': np.random.randint(10, 100, size=(200, )),

'x2': np.random.choice(['low', 'high', 'medium'], size=(200, )),

'x3': np.random.randint(-20, 30, size=(200, )),

})

print(df.head())

group\_df=df.groupby(by='x2')

res = group\_df.agg({

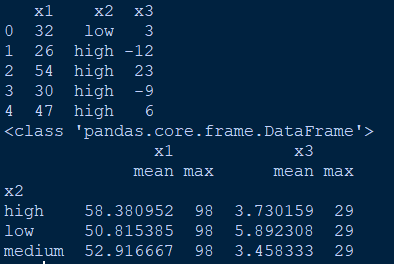
'x1': ['mean', 'max'],

'x3': ['mean', 'max']

})

print(type(res))

print(res)



## **Задание**

Разберитесь в аргументах DataFrame.to\_csv() и сохраните данные в файл так, чтобы в файл индекс не сохранялся.

import pandas as pd

import numpy as np

import string

df = pd.DataFrame(

data=np.random.randint(0, 10, size=(15, 3)),

columns=['x1', 'x2', 'x3'],

index=list(string.ascii\_uppercase)[:15])

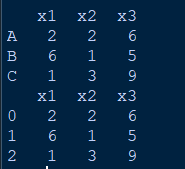
df.head(3)

print(df.head(3))

df.to\_csv('my\_second\_fail.csv',index=False)

df = pd.read\_csv('my\_second\_fail.csv')

print(df.head(3))



# **Задачки**

Создайте фрейм с тремя колонками:

* Колонка с именами (тип - объекты);
* Колонка с стажем работы (тип - вещественный);
* Колонка с возрастом (тип - целочисленный);
* Колонка с названием любимого цвета (тип - категориальный);

Имена колонок и значения любые, не менее трех записей (строк) в фрейме.

import pandas as pd

import numpy as np

#Задачки

#создайте фрейм и настройте правильные типы колонок

df = pd.DataFrame({'name': ['xyj','xyj1','xyj2','xyj3'],

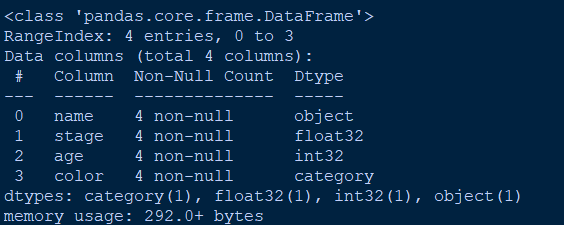
'stage': np.array([2.] \* 4, dtype='float32'),

'age': np.array([22] \* 4, dtype='int32'),

'color': pd.Categorical(["blue", "blue", "blue", "синий"])})

df

df.info()



Выберите числа в ряду ds1, которых нет в ряду ds2:

Почитайте и примение метод Series.isin()

import pandas as pd

import numpy as np

#Задачки

#создайте фрейм и настройте правильные типы колонок

ds1 = pd.Series([1, 2, 3, 4, 5])

ds2 = pd.Series([4, 5, 6, 7, 8, 1, 9])

mask=ds1.isin(ds2)

print(ds1[~mask])



Оставьте в ряду два наиболее частых значения, остальные замените значением 'Другое':

import pandas as pd

import numpy as np

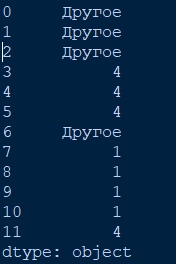
ds = pd.Series([2, 2, 2, 4, 4, 4, 3, 1, 1, 1, 1, 4])

mask=ds.value\_counts()

new\_mask=max(mask)==mask

mask1=ds.replace(mask[~new\_mask].index,'Другое')

print(mask1)



Сделайте каждую первую букву в словах ряда заглавной:

import pandas as pd

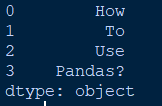
import numpy as np

ds = pd.Series(['how', 'to', 'use', 'pandas?'])

def change(x):

return x.title()

print(ds.apply(change))



Выберите записи с максимальным значением по колонке x1:

import pandas as pd

import numpy as np

df = pd.DataFrame(

np.random.randint(0, 4, size=(15, 3)),

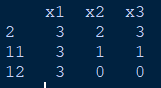
columns=['x1', 'x2', 'x3']

)

mask=(df['x1']==df['x1'].max())

df=df[mask]

print(df)



Выведите количество пропусков в каждой колонке данных:

import pandas as pd

import numpy as np

df = pd.read\_csv('https://raw.githubusercontent.com/Kail4eK/ml\_edu/master/datasets/Cars93\_miss.csv')

df.isna().sum()

Manufacturer 4

Model 1

Type 3

Min.Price 7

Price 2

Max.Price 5

MPG.city 9

MPG.highway 2

AirBags 6

DriveTrain 7

Cylinders 5

EngineSize 2

Horsepower 7

RPM 3

Rev.per.mile 6

Man.trans.avail 5

Fuel.tank.capacity 8

Passengers 2

Length 4

Wheelbase 1

Width 6

Turn.circle 5

Rear.seat.room 4

Luggage.room 19

Weight 7

Origin 5

Make 3

dtype: int64

Замените пропущенные значения в колонке Min.Price средним значениям по этой колонке:

import pandas as pd

import numpy as np

df = pd.read\_csv('https://raw.githubusercontent.com/Kail4eK/ml\_edu/master/datasets/Cars93\_miss.csv')

mean=df['Min.Price'].mean()

df['Min.Price'].fillna(mean,inplace=True)

df['Min.Price'].isna().sum()

0

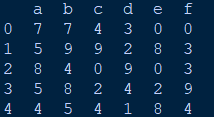
Отсортируйте и выведите фрейм с колонками в алфавитном порядке:

import pandas as pd

import numpy as np

df = pd.DataFrame(np.random.randint(0, 10, size=(5, 6)), columns=list('fbecda'))

print(df.reindex(sorted(df.columns),axis=1))



Отобразите каждую 20ю запись во фрейме и только колонки Manufacturer, Model, Type:

import pandas as pd

import numpy as np

df = pd.read\_csv('https://raw.githubusercontent.com/Kail4eK/ml\_edu/master/datasets/Cars93\_miss.csv')

df1=df.loc[20::20,['Manufacturer', 'Model', 'Type']]

print(df1)

| **Manufacturer** | **Model** | **Type** |
| --- | --- | --- |
| **20** | Chrysler | LeBaron | Compact |
| **40** | Honda | Prelude | Sporty |
| **60** | Mercury | Cougar | Midsize |
| **80** | Subaru | Loyale | Small |

Получите ряд, который содержит длины строк:

import pandas as pd

import numpy as np

ds = pd.Series(['how', 'to', 'use', 'pandas?'])

print(ds.str.len())

