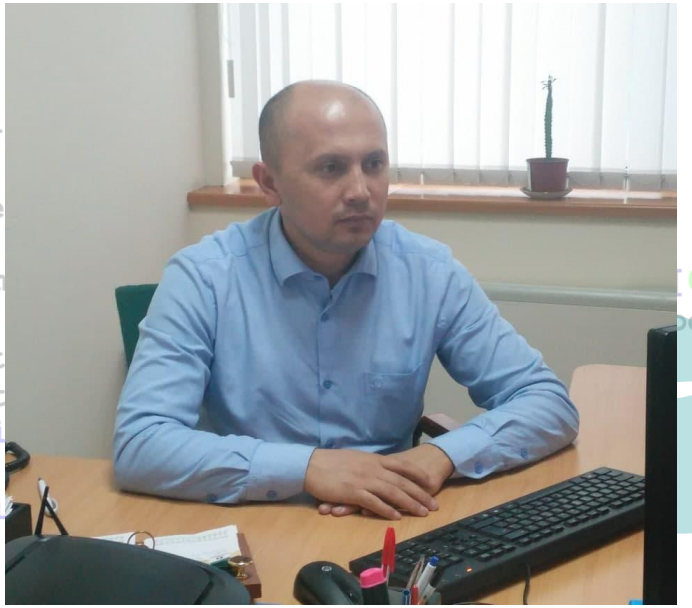
SUN’IY INTELLEKT ASOSLARI

SIA1046

02 Pandas kutubxonasi



Raxmanov Qurbon Sodikovich
t.f.n., raxmanov@gmail.com

Reja

1. Data analysisda Pandas kutubxonasi.
2. Series Ma’lumotlar tuzilmasi.
3. DataFrame Ma’lumotlar tuzilmasi.
4. Indekslar. Qator va ustunlarni tashlab yuborish.
5. Elementlarni tanlash.
6. Arifmetik amallar.
7. Funksiyalarni qo‘llash.
8. Tartiblash. Reytinglash
9. Dataset statistikasi: min, max, o‘rta qiymat, summa va korrelyasiya.
10. Ma’lumotlarni filtrlash.

Pandas kutubxonasi

Pandas - bu Python-da jadval ma’lumotlarini manipulyatsiya qilish uchun paket. Satrlar va ustunlar ko‘rinishidagi ma’lumotlar DataFrames (Excel, CSV, SQL ma’lumotlar bazalari va tezkor HDF5 formati) deb ham ataladi

Pandas funksionalligi ma’lumotlarni o‘zgartirishni o‘z ichiga oladi. Masalan, siz pandalardan qatorlarni saralash va pastki to‘plamlar, o‘rtachasi kabi umumiy statistikani hisoblash, freymlarni qayta shakllantirish va formalarni birlashtirish uchun foydalanishingiz mumkin.

Pandas kutubxonasi

Pandas ko‘pincha PyData ekotizim deb ataladigan boshqa mashhur Python ma’lumotlar paketlari bilan yaxshi ishlaydi:

NumPy - raqamli hisob-kitoblar uchun

Matplotlib, Seaborn, Plotly - ma’lumotlarni vizuallashtirish uchun

scikit-learn - mashinali o‘qitish uchun

Pandas kutubxonasi nima uchun ishlatiladi?

Ma’lumotlar to‘plamlarini ma’lumotlar bazalari, elektron jadvallar, CSV fayllari va boshqalardan import qilishda

Ma’lumotlar to‘plamlarini tozalashda. Masalan, yetishmayotgan qiymatlarni yo‘q qilish.

Ma’lumotlar to‘plamlarini ularning tuzilishini tahlil qilish uchun mos formatga aylantirish orqali tartibga solishda.

Ustunlarning o‘rtacha ko‘rsatkichi, ular orasidagi bog'liqlik va boshqalar kabi umumlashtirilgan statistik ma’lumotlarni hisoblash orqali ma’lumotlarni jamlash.

Ma’lumotlar to‘plamini tasavvur qilishda va yangi imkoniyatlarni kashf etishda.

Pandas kutubxonasi imkoniyatlari

Python uchun yaratilgan. Python - bu mashinani o‘rganish va ma’lumotlar fanlari uchun dunyodagi eng mashhur til.

Operatsiya birligi uchun kamroq aniqlik. Pandasda yozilgan kod ixcham va kerakli natijani olish uchun kamroq satrlarni talab qiladi.

Ma’lumotlarning intuitive shaklda taqdim etadi. Pandas ma’lumotlarning juda intuitiv taqdimotini taklif qiladi, bu tushunish va tahlil qilishni osonlashtiradi.

Funktsiyalarning keng to‘plami. Pandas keng ko‘lamli operatsiyalarni qo‘llab-quvvatlaydi: ma’lumotlarni tahlil qilish, etishmayotgan qiymatlar bilan ishlash, statistikani hisoblash, 1D va 2D ma’lumotlarini vizualizatsiya qilish va boshqalar.

Katta ma’lumotlar bilan ishlash. Pandalar katta ma’lumotlar to‘plamini osonlik bilan boshqaradi. U mashinaga qarab millionlab yozuvlar va yuzlab ustunlarni o‘z ichiga olgan ma’lumotlar to‘plamlari bilan ishlashda tezlik va samaradorlikni ta’minlaydi.

Pandas kutubxonasi tarixi

2008 yil : pandas ishlab chiqarila boshlandi

2009 yil : pandas ochiq manbaga aylandi

2012 : Ma’lumotlarni tahlil qilish uchun Pythonning birinchi nashri nashr etildi

2015 : NumFOCUS pandasga homiylik qildi

Pandas kutubxonasining afzalliklari

Integratsiyalashgan indekslash, ma’lumotlarni boshqarish uchun tezkor va samarali DataFrame ob'ekti;

Xotira ichidagi ma’lumotlar tuzilmalari va turli formatlari o‘rtasida **ma’lumotlarni o‘qish va yozish** vositalari : CSV va matnli fayllar, Microsoft Excel, SQL ma’lumotlar bazalari va tezkor HDF5 formati;

Sun’iy intellekt uchun ma’lumotlarni moslashtirish: Ma’lumotlarni oqilona va osongina tartiblash;

Ma’lumotlar to‘plamini moslashuvchan qayta shakllantirish va o‘zgartirish, zamonaviy indeksatsiya va katta ma’lumotlar to‘plamlarini kichik guruhlarga bo‘lish;

Pandas kutubxonasining afzalliklari

Jadvallar uchun ustunlar kiritish va o‘chirish;

Ma’lumotlar to‘plamida bo‘linadigan va birlashtiriladigan operatsiyalarni bajarishga imkon beradigan kuchli guruhlash dvigatelidan foydalangan holda ma’lumotlarni yig'ish yoki o‘zgartirish;

Ma’lumotlar to‘plamini yuqori darajada **birlashtirish**;

Ierarxik o‘qni indekslash quyi o‘lchovli ma’lumotlar tarkibida yuqori o‘lchovli ma’lumotlar bilan ishlashning intuitiv usulini ta’minlash;

Pandas kutubxonasining afzalliklari

Vaqt seriyasining funktsionalligi: sana diapazonini yaratish va chastotani o‘zgartirish harakatlanuvchi oyna statistikasi va sana o‘zgarishi. Ma’lumotlarni yo‘qotmasdan, hatto domenga xos vaqt oralig'ini yarating va vaqt qatorlariga qo‘shing;

Cython yoki C da yozilgan muhim kod yo‘llari bilan ishlash uchun juda optimallashtirilgan .

Pandalar bilan turli xil ilmiy va tijorat sohalarida, jumladan, moliya, nevrologiya, iqtisod, statistika, reklama, veb-tahlil va boshqalarda qo‘llaniladi.

Cython sizga Python va C ning umumiy quvvatini beradi

Pandas kutubxonasining ko‘rinishi

Hamma uchun ochiq

Foydalanuvchilar foydalanishi va o‘zgartirishi uchun bepul

Moslashuvchan

Kuchli

Foydalanish oson

Tez

Pandasdan foydalanish

Pandas kutubxonasi Data Sciensening eng muhim ish quroli hisoblanadi. Data Science loyihalar ustida ishlar ekanmiz, 90% amallar aynan Pandas yordamida amalga oshiriladi.

Pandas o‘zining maxsus ma’lumotlar tuzilmalariga ega va bu tuzilmalar ma’lumotlarga ishlov berish, tozalash va boshqa amllarni bir necha barobar osonlashtiradi.

NumPy kutubxonasidan farqli ravishda, Pandas kutubxonasi faqatgina sonlardan iborat massivlar bilan emas, balki turli ko‘rinishdagi ma’lumotlardan iborat jadvallar bilan ishlashga moslashgan.

Pandasdan foydalanish

Pandas ikkita juda muhim loyihalar uchun mos tushuvchi Series va DataFrame ma’lumotlar tuzilmalariga ega.

```
#NumPy va Pandas kutubxonaslarini chaqirib olish
import pandas as pd
import numpy as np
```

```
#Pandasdan Series va DataFrameni esa qisqacha df deb chariqib olish
from pandas import Series, DataFrame as df
```

Series bir o‘lchovli massivga o‘xshash ma’lumotlar tuzilmasi bo‘lib qiymatlar va ularga mos keluvchi indekslardan iborat bo‘ladi.

Pandasdan foydalanish

```
from pandas import Series, DataFrame as df
```

```
obj = Series([4, 5, -7, 3.2, 2023, 11, 27])
obj
```

```
0       4.0
1       5.0
2      -7.0
3       3.2
4    2023.0
5      11.0
6      27.0
dtype: float64
```

Konsolga chop etganda, Series yuqoridagi kabi indekslar (chap ustun) va qiymatlar (o‘ng ustun) ko‘rinishida chiqadi.

Series obyektini yaratishda indekslarni bermaganimiz sababli indekslar 0 dan N-1 gacha qiymatlarni oladi (N ma’lumotlar uzunligi).

Pandasdan foydalanish

```
obj.values
```

```
array([   4. ,    5. ,   -7. ,    3.2, 2023. ,   11. ,   27. ])
```

```
obj.index
```

```
RangeIndex(start=0, stop=7, step=1)
```

```
obj[2]
```

```
-7.0
```

```
obj[5]
```

```
11.0
```

Series obyekti index va qiymatlarini alohida ajratib ko‘rish ham mumkin.

Pandasdan foydalanish

```
obj1 = Series([4, 5, -7, 3.2, 2023, 11, 27], index=['a', 'b', 'c', 'd', 'e', 'f', 'h'])
obj1
```

```
a       4.0
b       5.0
c      -7.0
d       3.2
e    2023.0
f      11.0
h      27.0
dtype: float64
```

```
obj1.index
```

```
Index(['a', 'b', 'c', 'd', 'e', 'f', 'h'], dtype='object')
```

```
obj1['b']
```

```
5.0
```

Series obyektini yaratishda indekslarga o‘zimiz ham nom berishimiz mumkin

Pandasdan foydalanish

```
obj1['b']= 505 # yangi qiymat berish
```

```
obj1
```

```
a       4.0
b     505.0
c      -7.0
d       3.2
e    2023.0
f      11.0
h      27.0
dtype: float64
```

```
obj1['j']=2540 #yangi index va qiymat berish
```

```
obj1
```

```
a       4.0
b     505.0
c      -7.0
d       3.2
e    2023.0
f      11.0
h      27.0
j    2540.0
dtype: float64
```

Obyekt ichidagi qiymatlarga esa indeks orqali murojaat qilish va yangi qiymat berish

Obyekt ichidagi qiymatlarga yangi indeks qo‘shish va qiymat berish

Pandasdan foydalanish

```
obj1[['a', 'c', 'e', 'h']]
```

```
a       4.0
c      -7.0
e    2023.0
h      27.0
dtype: float64
```

Birdaniga bir nechta qiymatlarni chiqarish

```
obj1*2
```

```
a       8.0
b    1010.0
c     -14.0
d       6.4
e    4046.0
f      22.0
h      54.0
j    5080.0
dtype: float64
```

Barcha qiymatlarni ko‘paytirish

```
obj1[obj1>50]
```

```
b     505.0
e    2023.0
j    2540.0
dtype: float64
```

Barcha qiymatlarga shart qo‘yish

```
obj1[obj1<50]
```

```
a     4.0
c    -7.0
d     3.2
f    11.0
h    27.0
dtype: float64
```

Pandasdan foydalanish

```
np.exp(obj1)
```

```
/usr/local/lib/python3.10/d
  result = getattr(ufunc, m
a     5.459815e+01
b    2.083116e+219
c     9.118820e-04
d     2.453253e+01
e              inf
f     5.987414e+04
h     5.320482e+11
j              inf
dtype: float64
```

Obj1 barcha qiymatlarini exponentasini topish

Series obyektini Pythondagi lug'at (dictionary) ma’lumot turiga ham o‘xshatish mumkin. Yodingizda bo‘lsa lug'at elementlari ham 2 qismdan, kalit va qiymatdan iborat edi.

```
'a' in obj1
```

```
True
```

```
'x' in obj1
```

```
False
```

o‘rni kelganda bu turdagi obyektlardan lug'atdan foydalangandek ham foydalanish mumkin:

Pandasdan foydalanish Series

```
mashina_dict = {'Malibu':30000, 'Lacetti':18000, 'Toyota':45000,
'Monza':17000, 'Cobalt':14000, 'BYD':35000} # bu lug'at
```

```
avtosalon = Series(mashina_dict)
```

```
avtosalon
```

```
Malibu     30000
Lacetti    18000
Toyota     45000
Monza      17000
Cobalt     14000
BYD        35000
dtype: int64
```

```
avtosalon['Cobalt']
```

```
14000
```

Lug'atdan Series obyektini yaratish ham juda oson:

Lug'atdan Series obyektini yaratish

Pandasdan foydalanish Series

```
models = ['Monza','Cobalt','Lacetti','Jetour', 'BYD', 'Malibu','Toyota']
# bu yerda Jetour modeli qo'shimcha qilib yozilgan
#Lug'atdan Series yaratishda yangi elementlar indeksi lug'atda berilgan ketma-ketlikda bo'ladi.
#Bu ketma-ketlikni biz avvaldan ro'yxat ko'rinishida ham berishimiz mumkin.
```

```
avtosalon2 = Series(mashina_dict, index=models)
#Modellar ro'yxatida qo'shimcha Jetour elementi ham bor,
#Series elementi yaratilganda ro'yxtada mavjud bo'lmagan qiymatlar o'rniga
#NaN (Not a number, son emas) qiymati qo'yib ketiladi.
avtosalon2
```

```
Monza      17000.0
Cobalt     14000.0
Lacetti    18000.0
Jetour         NaN
BYD        35000.0
Malibu     30000.0
Toyota     45000.0
dtype: float64
```

Pandasdan foydalanish Series

```
#Ma'lumotlar bilan ishlaganda NaN qiymatlarni ajratish muhim ahamiyatga ega.
#Buning uchun Series obyekti bir maxsus metodlarga ega:
avtosalon2.isnull() # qiymat null borligini tekshirish uchun
```

```
Monza      False
Cobalt     False
Lacetti    False
Jetour      True
BYD        False
Malibu     False
Toyota     False
dtype: bool
```

```
avtosalon2.notnull() # qiymat null emasligini tekshirish uchun
```

```
Monza       True
Cobalt      True
Lacetti     True
Jetour     False
BYD         True
Malibu      True
Toyota      True
dtype: bool
```

Pandasdan foydalanish Series +

```
avtosalon2
# Series obyektining muhim jihatlaridan biri, ikkita series obyektini qo'shishda (ayirishda) qiymatlar indeks bo'yicha qo'shiladi.
```

```
Monza      17000.0
Cobalt     14000.0
Lacetti    18000.0
Jetour         NaN
BYD        35000.0
Malibu     30000.0
Toyota     45000.0
dtype: float64
```

```
yangi_narx = Series({'Malibu':-3000, 'Lacetti':-1800, 'Toyota':-5000,'Monza':-700, 'Cobalt':-1000, 'BYD':-8000})
```

```
avtosalon2 + yangi_narx
```

```
BYD        27000.0
Cobalt     13000.0
Jetour         NaN
Lacetti    16200.0
Malibu     27000.0
Monza      16300.0
Toyota     40000.0
dtype: float64
```

Pandasdan foydalanish Series +

```
] #Series obyektining o'zini ham indekslarini ham nomlash mumkin:
  avtosalon2.name = 'Avtosalon'
  avtosalon2.index.name = 'Mashinalar modeli va narxi'
```

```
] avtosalon2
```

```
Mashinalar modeli va narxi
Monza       17000.0
Cobalt      14000.0
Lacetti     18000.0
Jetour          NaN
BYD         35000.0
Malibu      30000.0
Toyota      45000.0
Name: Avtosalon, dtype: float64
```

```
] avtosalon2.name
```

```
'Avtosalon'
```

Pandasdan foydalanish Series +

```
avtosalon2.values
```

```
array([17000., 14000., 18000.,    nan, 35000., 30000., 45000.])
```

```
avtosalon2.hasnans # NaN qiymatlar bor (True) yo'qligi (False)
```

```
True
```

```
avtosalon2.dtype # qiymatlarning ma'lumot turi
```

```
dtype('float64')
```

```
avtosalon2.is_unique # qiymatlar takrorlanmas ekanini tekshirish
```

```
True
```

```
avtosalon2.shape # Series hajmi
```

```
(7,)
```

```
avtosalon2.size # Series o'lchami
```

```
7
```

Pandasdan foydalanish Series +

```
avtosalon2.iloc[2] # ma'lumotlarga indeks tartib raqami bo'yicha murojaat qilish
```

```
18000.0
```

```
avtosalon2.iloc[-2]
```

```
30000.0
```

```
avtosalon2['Cobalt'] # ma'lumotlarga indeks bo'yicha murojaat qilish
```

```
14000.0
```

```
avtosalon2.max() # max qiymat
```

```
45000.0
```

```
avtosalon2.min() # max qiymat
```

```
14000.0
```

```
avtosalon2.mean() # o'rtacha qiymat
```

```
26500.0
```

.head() va .tail() yordamida ma’lumotlarni ko‘rish

.head() va .tail() usullaridan foydalanib, siz mos ravishda DataFrame ning birinchi yoki oxirgi qatorlarini ko‘rish mumkin. Qatorlar sonini n argumenti yordamida aniqlash mumkin (standart 5).

```
amazon = pd.read_csv("amazon_categories.csv")
TIMES = pd.read_csv("TIMES_WorldUniversityRankings_2024.csv")
```

```
amazon.head
```

```
<bound method NDFrame.head of       id
0      1                          Beading & Jewelry Making
1      2                                   Fabric Decorating
2      3                        Knitting & Crochet Supplies
3      4                                Printmaking Supplies
4      5                  Scrapbooking & Stamping Supplies
..   ...                                                ...
243  262  Xbox Series X & S Consoles, Games & Accessories
244  263                              PC Games & Accessories
245  264                       Baby Girls' Clothing & Shoes
246  265                              Boys' School Uniforms
247  270                                        Toys & Games

[248 rows x 2 columns]>
```

```
amazon.tail(n = 10)
```

	id	category_name
238	255	Video Games
239	256	Online Video Game Services
240	259	Virtual Reality Hardware & Accessories

.head() va .tail() yordamida ma’lumotlarni ko‘rish

	id	category_name
238	255	Video Games
239	256	Online Video Game Services
240	259	Virtual Reality Hardware & Accessories

```
amazon.tail(n = 10)
```

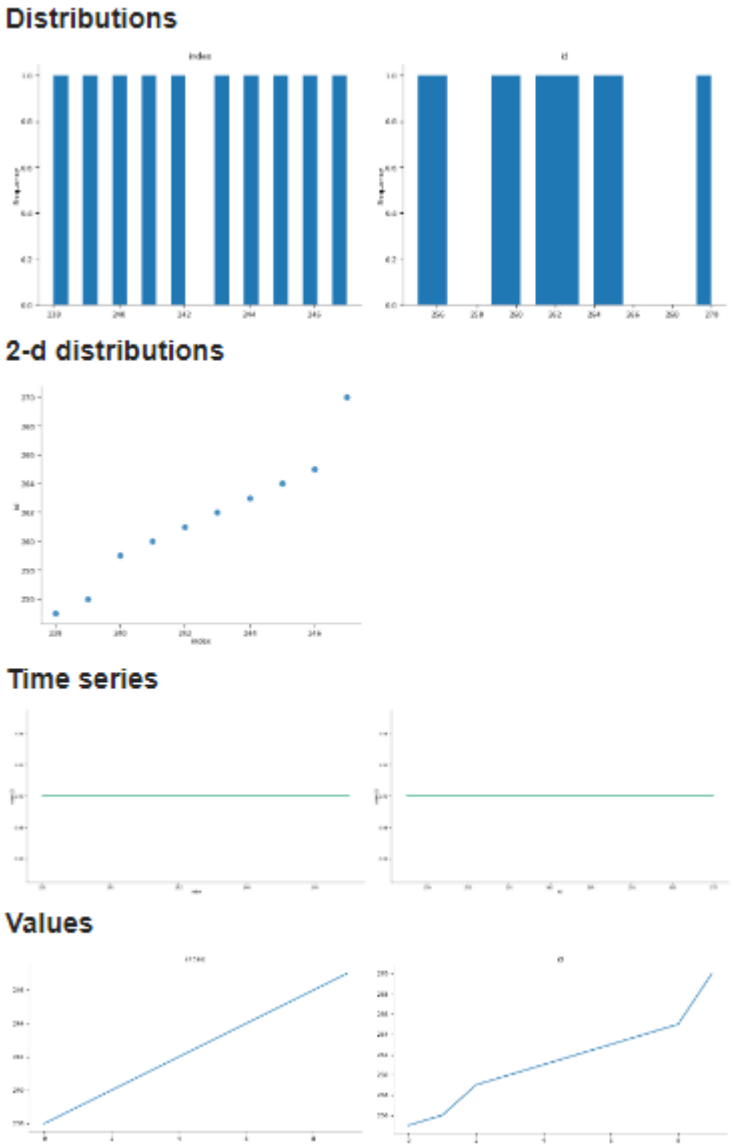
1 to 10 of 10 entries Filter

index	id	category_name
238	255	Video Games
239	256	Online Video Game Services
240	259	Virtual Reality Hardware & Accessories
241	260	Nintendo Switch Consoles, Games & Accessories
242	261	PlayStation 5 Consoles, Games & Accessories
243	262	Xbox Series X & S Consoles, Games & Accessories
244	263	PC Games & Accessories
245	264	Baby Girls' Clothing & Shoes
246	265	Boys' School Uniforms
247	270	Toys & Games

Show 25 per page

.describe() usuli yordamida ma’lumotlar tahlili

	id	category_name
238	255	Video Games
239	256	Online Video Game Services
240	259	Virtual Reality Hardware & Accessories



.head() va .tail() yordamida ma’lumotlarni ko‘rish

.describe() usuli barcha son ustunlarning umumiy statistikasini ko‘rsatadi. Masalan, soni, o‘rtacha, standart og'ish, diapazon va kvartillar.

```
amazon.describe()
```

	id
count	248.000000
mean	133.875000
std	77.132441
min	1.000000
25%	67.500000
50%	137.500000
75%	199.250000
max	270.000000

```
TIMES.describe()
```

	scores_overall_rank	scores_teaching	scores_teaching_rank	scores_research	scores
count	2.673000e+03	1904.000000	2673.000000	1904.000000	
mean	2.945872e+05	29.060662	678.473625	23.416176	
std	4.486570e+05	13.967201	633.450078	16.697923	
min	1.000000e+01	9.400000	0.000000	4.600000	
25%	6.690000e+03	19.400000	0.000000	11.800000	
50%	1.337000e+04	25.800000	568.000000	17.250000	
75%	1.000101e+06	34.500000	1236.000000	29.900000	
max	1.000769e+06	99.000000	1904.000000	100.000000	

percentiles yordamida ma’lumotlarni ko‘rish

Bu usul sizga raqamli ma’lumotlarning masshtabini, dispersiyasini va diapazonini tezda baholash imkonini beradi.
Bundan tashqari, foizlar argumentidan foydalanib, kvartillarni o‘zgartirishingiz mumkin. Bu erda, masalan, DataFrame df-dagi raqamli ustunlarning 30%, 50% va 70% foizli nisbatlarini ko‘rib chiqamiz.

```
TIMES.describe(percentiles=[0.3, 0.5, 0.7])
```

	scores_overall_rank	scores_teaching	scores_teaching_rank	scores_research	scores_research_rank	scores_citations	s
count	2.673000e+03	1904.000000	2673.000000	1904.000000	2673.000000	1904.000000	
mean	2.945872e+05	29.060662	678.473625	23.416176	678.473625	52.189706	
std	4.486570e+05	13.967201	633.450078	16.697923	633.450078	25.071342	
min	1.000000e+01	9.400000	0.000000	4.600000	0.000000	3.400000	
30%	8.026000e+03	20.590000	33.600000	12.700000	33.600000	34.590000	
50%	1.337000e+04	25.800000	568.000000	17.250000	568.000000	52.400000	
70%	1.871400e+04	32.410000	1102.400000	26.000000	1102.400000	69.300000	
max	1.000769e+06	99.000000	1904.000000	100.000000	1904.000000	99.700000	

include yordamida ma’lumotlarni ko‘rish

Include ma’lumotlar turlarini ajratib ko‘rsatish yoki argumentdan ham foydalanish mumkin. Masalan, faqat butun sonli ma’lumotlar turiga ega ustunlarni yig’ish.

```
TIMES.describe(include=[int])
```

	scores_overall_rank	scores_teaching_rank	scores_research_rank	scores_citations_rank	scores_industry_income_rank	scor
count	2.673000e+03	2673.000000	2673.000000	2673.000000	2673.000000	
mean	2.945872e+05	678.473625	678.473625	678.473625	678.473625	
std	4.486570e+05	633.450078	633.450078	633.450078	633.450078	
min	1.000000e+01	0.000000	0.000000	0.000000	0.000000	
25%	6.690000e+03	0.000000	0.000000	0.000000	0.000000	
50%	1.337000e+04	568.000000	568.000000	568.000000	568.000000	
75%	1.000101e+06	1236.000000	1236.000000	1236.000000	1236.000000	
max	1.000769e+06	1904.000000	1904.000000	1904.000000	1904.000000	

exclude yordamida ma’lumotlarni ko‘rish

Xuddi shunday, exclude argument yordamida ma’lum turdagi ma’lumotlarni chiqarib tashlash mumkin.

```
TIMES.describe(exclude=[int])
```

	rank	name	scores_overall	scores_teaching	scores_research	scores_citations	scores_industry_income	sc
count	2673	2673	1904	1904.000000	1904.000000	1904.000000	1904.000000	
unique	160	2673	159	NaN	NaN	NaN	NaN	
top	Reporter	University of Oxford	9.7–22.7	NaN	NaN	NaN	NaN	
freq	769	1	395	NaN	NaN	NaN	NaN	
mean	NaN	NaN	NaN	29.060662	23.416176	52.189706	47.057405	
std	NaN	NaN	NaN	13.967201	16.697923	25.071342	26.150454	
min	NaN	NaN	NaN	9.400000	4.600000	3.400000	15.600000	
25%	NaN	NaN	NaN	19.400000	11.800000	30.400000	22.275000	
50%	NaN	NaN	NaN	25.800000	17.250000	52.400000	41.050000	
75%	NaN	NaN	NaN	34.500000	29.900000	73.225000	68.200000	
max	NaN	NaN	NaN	99.000000	100.000000	99.700000	100.000000	

11 rows × 21 columns

.T yordamida ma’lumotlarni ko‘rish

Ko‘pincha amaliyotchilar uchun bunday statistik ma’lumotlarni .T atributidan foydalanib ko‘chirish orqali ko‘rish qulay.

```
TIMES.describe().T
```

	count	mean	std	min	25%	50%	75%	max
scores_overall_rank	2673.0	294587.229704	448657.024118	10.0	6690.000	13370.00	1000101.000	1000769.0
scores_teaching	1904.0	29.060662	13.967201	9.4	19.400	25.80	34.500	99.0
scores_teaching_rank	2673.0	678.473625	633.450078	0.0	0.000	568.00	1236.000	1904.0
scores_research	1904.0	23.416176	16.697923	4.6	11.800	17.25	29.900	100.0
scores_research_rank	2673.0	678.473625	633.450078	0.0	0.000	568.00	1236.000	1904.0
scores_citations	1904.0	52.189706	25.071342	3.4	30.400	52.40	73.225	99.7
scores_citations_rank	2673.0	678.473625	633.450078	0.0	0.000	568.00	1236.000	1904.0
scores_industry_income	1904.0	47.057405	26.150454	15.6	22.275	41.05	68.200	100.0
scores_industry_income_rank	2673.0	678.473625	633.450078	0.0	0.000	568.00	1236.000	1904.0
scores_international_outlook	1904.0	49.907143	21.828249	16.1	31.975	45.30	65.400	98.8
scores_international_outlook_rank	2673.0	678.473625	633.450078	0.0	0.000	568.00	1236.000	1904.0
member_level	2673.0	0.550318	2.231166	0.0	0.000	0.00	0.000	11.0
nid	2673.0	437757.592593	292321.295190	466.0	131435.000	622251.00	675423.000	726248.0
stats_student_staff_ratio	2673.0	19.130527	13.450010	0.3	12.300	16.60	22.500	333.3

.info() usuli yordamida ma’lumotlarni tushunish

.info() usuli DataFrame-dagi ma’lumotlar turlarini, etishmayotgan qiymatlarini va hajmini ko‘rishning tezkor usulidir. show_counts argumentini True ga o‘rnatsak, har bir ustunda etishmayotgan qiymatlarning umumiy sonini ko‘rish imkonini beradi.

memory_usage=True. memory_usage DataFrame elementlarining umumiy xotiradan foydalanishini ko‘rsatadi. Agar batafsil ma’lumot "True" ga o‘rnatilgan bo‘lsa, .info() dan to‘liq xulosa chop etiladi.

.info() usuli yordamida ma’lumotlarni tushunish

```
TIMES.info(show_counts=True, memory_usage=True, verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2673 entries, 0 to 2672
Data columns (total 29 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   rank                            2673 non-null   object
 1   name                            2673 non-null   object
 2   scores_overall                  1904 non-null   object
 3   scores_overall_rank             2673 non-null   int64
 4   scores_teaching                 1904 non-null   float64
 5   scores_teaching_rank            2673 non-null   int64
 6   scores_research                 1904 non-null   float64
 7   scores_research_rank            2673 non-null   int64
 8   scores_citations                1904 non-null   float64
 9   scores_citations_rank           2673 non-null   int64
 10  scores_industry_income          1904 non-null   float64
 11  scores_industry_income_rank     2673 non-null   int64
 12  scores_international_outlook    1904 non-null   float64
 13  scores_international_outlook_rank 2673 non-null   int64
 14  record_type                     2673 non-null   object
 15  member_level                    2673 non-null   int64
 16  url                             2673 non-null   object
 17  nid                             2673 non-null   int64
 18  location                        2673 non-null   object
 19  stats_number_students           2673 non-null   object
 20  stats_student_staff_ratio       2673 non-null   float64

 26  unaccredited                    2673 non-null   bool
 27  disabled                        2673 non-null   bool
 28  website_url                     344 non-null    object
dtypes: bool(3), float64(6), int64(8), object(12)
memory usage: 550.9+ KB
```

```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodename()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '    %s [label="%s' % (nodename, label),
    print '"];'
    children = []
    for n, child
        children.ap
    print '    %s
    for name in children:
        print '%s' % name,
```

E’tiboringiz uchun rahmat!

Raxmanov Qurbon Sodikovich
t.f.n., raxmanov@gmail.com