DataFrame Əməliyyatları və Pivot Table (Yekunlar Cədvəli):

Data -mızı yaratmaqdan başlayaq:

```
df.to_csv('azdatasaxta.csv')
```

```
df = pd.DataFrame(pd.read_csv("/content/azdatasaxta.csv"), columns=data.keys(),index = data_sira)
```

```
df = df.rename_axis(index='Sıra Nömrəsi')
df.dropna(inplace=True)
df
```

Sıra Nömrəsi	Ad	Soyad	Sinif	Cinsi	Təvəllüdü
1	Ceyms	Teymurov	10A	K	1990.0
2	Teymur	Akifoğlu	10A	K	1990.0
3	Qardaşxan	Elşadzadə	10B	K	1990.0
4	Mahir	Niyazlı	10B	K	1990.0
5	Mövlud	Hümmətli	10A	K	1991.0
6	Kənan	Altaylı	10A	K	1991.0
7	Kazuo	Teymurzadə	10B	K	1991.0
9	Qardaşxan	Teymurov	10B	K	1992.0
10	Zaman	Zauroğlu	9A	K	1989.0
12	Aqşin	Nərimanov	9A	K	1990.0
13	Niyaz	Kazımov	9B	K	1992.0
14	Qasımbəy	Elmanoğlu	9B	K	1990.0
15	Elmar	Abdullahlı	9A	K	1991.0
16	Rəşid	Emilzadə	9A	K	1991.0
17	Mark	Vahidli	9B	K	1991.0
18	Heysəm	Con	9A	K	1992.0
19	Ramin	Emilzadə	9B	K	1992.0
20	Xatirə	Ramizli	10B	Q	1989.0
21	Rusudan	Anatoli	10A	Q	1990.0
22	Mədinə	Elşadqızı	10A	Q	1990.0
23	Rəna	Çingizzadə	10B	Q	1990.0
24	Yekaterina	Əsgərli	10B	Q	1990.0
25	Pərvanə	Nəsimov	10A	Q	1991.0
26	Luiza	Elnurlu	10B	Q	1991.0
27	Lətafət	Babakişiyeva	10B	Q	1991.0
28	Tərlan	Babakişiyeva	10A	Q	1992.0
29	Şölə	Qazıyev	10B	Q	1992.0
30	Gülşən	Sultanzadə	9B	Q	1989.0
31	Hikari	Mustafa	9A	Q	1990.0
32	Xədicə	Fahirli	9A	Q	1990.0
33	Luiza	Bəhram	9B	Q	1990.0
34	Şəfa	Razi	9B	Q	1990.0
35	Marina	Edvard	9A	Q	1991.0
36	Gülüstan	Mikayılova	9B	Q	1991.0
37	Rusudan	Anarlı	9B	Q	1991.0
38	Flora	Kənanzadə	9A	Q	1992.0
39	Reyhan	Razi	9B	Q	1992.0

İlk öncə məqsədimizi qeyd edək ki, hərşey anlaşılan olsun. 3 sadə amma faydalılıq baxımından çox yüksək keyfiyyətli funksiya vardır:

1. head ( ) - bu funksiya bizə data haqqında anlayış verməsi baxımından defolt olaraq onun ilk 5 sətirini , bütun sütunlarla qarşımıza çıxardır və data haqqında ilkin biliklərə malik oluruq. Bu funksiyadan istifadədə yaranan çətin problemlərdən biri də SQL və digər DMS (Database management systems) -lərdə heç bir dataya müdaxiləyə izn olmamasından yaranır. Belə ki, test mühitində işləyən verilənlər bazaları ilkin məlumatları sınaq üçün yazılmış və birçox hissəsi düzgün göstəricilərlə əhatə olunmamış ola bilir belə ki, bir çox hallarda verilənlər bazasınıın test mühitinin belə toxunulmazlığı qaydaları (işləri idarə edən şirkət, qurum və mövcud ölkənin normativ hüquqi aktlarından irəli gələn tələblər) bu data -ları silməyə izn vermir. Yəni siz sorğu ilə SQL dən aldığınız datada head() işlədərsəniz buradakı məlumatlar sizə düzgün fikir verməyə bilər.
2. tail ( ) - bu funksiya (defolt olaraq son 5 sətiri alaraq) son dövrləri əhatə edər bu da ümumi datanı tam anlamamıza işıq tutmaya bilər.
3. sample ( ) - bu funksiya (defolt olaraq təsadüfi 1 ədəd sətiri alaraq) cədvəlin hər hansı hissəsindən sizə bir sətirlik məlumatı gətirər.

Ümumi baxdıqda bu üç funksiyanın hər birindən istifadədə öz problemləri var. Ona görə də hər üç funksiyadan baxış üçün istifadə edək:

```
pd.concat([df.head(5),df.sample(5),df.tail(5)])
```

Sıra Nömrəsi	Ad	Soyad	Sinif	Cinsi	Təvəllüdü
1	Ceyms	Teymurov	10A	K	1990.0
2	Teymur	Akifoğlu	10A	K	1990.0
3	Qardaşxan	Elşadzadə	10B	K	1990.0
4	Mahir	Niyazlı	10B	K	1990.0
5	Mövlud	Hümmətli	10A	K	1991.0
3	Qardaşxan	Elşadzadə	10B	K	1990.0
37	Rusudan	Anarlı	9B	Q	1991.0
29	Şölə	Qazıyev	10B	Q	1992.0
10	Zaman	Zauroğlu	9A	K	1989.0
2	Teymur	Akifoğlu	10A	K	1990.0
36	Gülüstan	Mikayılova	9B	Q	1991.0
37	Rusudan	Anarlı	9B	Q	1991.0
38	Flora	Kənanzadə	9A	Q	1992.0
39	Reyhan	Razi	9B	Q	1992.0
40	NaN	NaN	NaN	NaN	NaN

Dataframe daxilində olan ədədi göstəricilərin statistik xülasəsinə baxaq:

```
df.describe()
```

	Təvəllüdü
count	39.000000
mean	1990.641026
std	0.902837
min	1989.000000
25%	1990.000000
50%	1991.000000
75%	1991.000000
max	1992.000000

Sütun sayı az olduğundan xoş görünməsi üçün və ya digər məqsədlərlə biz cədvəli 90 dərəcə əyə bilərik, yəni Transpose edərək sütun və sətirlərin yerini dəyişə bilərik:

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Təvəllüdü	39.0	1990.641026	0.902837	1989.0	1990.0	1991.0	1991.0	1992.0

Beləliklə biz, '.T' işlədərək statistik dəyərləri daha rahat müşahidə etmiş oluruq.

Biz ümumi DataFrame -nin indeks sayını, sütunların data tipini, sütunlarda mövcud dolu xanaların sayını, data -nın ram -da tutduğu yerin miqdarını öyrənmək üçün ***info( )*** metodunu istifadə edərik:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 40 entries, 1 to 40
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Ad         37 non-null     object 
 1   Soyad      39 non-null     object 
 2   Sinif      39 non-null     object 
 3   Cinsi      39 non-null     object 
 4   Təvəllüdü  39 non-null     float64
dtypes: float64(1), object(4)
memory usage: 1.9+ KB
```

describe ( ) metodu statistik informasiyanı ekrana gətirərkən,

corr ( ) metodu sütunlar arasındakı korrelasiya haqqında bizə fikir verər. Lakin üzərində işlədiyimiz data korrelasiya əlaqələrini aydın görmək üçün uyğun deyil. Bu metoddan daha sonra, aşağıda istifadə edəcəyik.

Yuxarıda verdiyimiz nümunələrdə istifadə etdiyimiz metodlardan dataya ilkin baxış üçün yararlanmağı tövsiyə edirəm.

```
df['Soyad'].unique()
```

```
array(['Teymurov', 'Akifoğlu', 'Elşadzadə', 'Niyazlı', 'Hümmətli',
       'Altaylı', 'Teymurzadə', 'Davudov', 'Zauroğlu', 'Nərimanov',
       'Kazımov', 'Elmanoğlu', 'Abdullahlı', 'Emilzadə', 'Vahidli', 'Con',
       'Ramizli', 'Anatoli', 'Elşadqızı', 'Çingizzadə', 'Əsgərli',
       'Nəsimov', 'Elnurlu', 'Babakişiyeva', 'Qazıyev', 'Sultanzadə',
       'Mustafa', 'Fahirli', 'Bəhram', 'Razi', 'Edvard', 'Mikayılova',
       'Anarlı', 'Kənanzadə', nan], dtype=object)
```

```
df['Soyad'].nunique()
```

```
34
```

```
df.nunique()
```

```
Ad           34
Soyad        34
Sinif         4
Cinsi         2
Təvəllüdü     4
dtype: int64
```

unique ( ) metodu bizə sütundakı unikal dataları array (massiv) daxilində geri qaytarar.
nunique () metodu bizə neçə 'unique' (unikal) data verildiyini ifadə edər.

```
df['Təvəllüdü'].value_counts()
```

```
1990.0    16
1991.0    12
1992.0     8
1989.0     3
Name: Təvəllüdü, dtype: int64
```

value_counts () metodu hansı dəyərdən neçə ədəd mövcuddur onu bizə bildirər

```
df[df['Təvəllüdü'] >= 1992]
```

Sıra Nömrəsi	Ad	Soyad	Sinif	Cinsi	Təvəllüdü
9	Qardaşxan	Teymurov	10B	K	1992.0
13	Niyaz	Kazımov	9B	K	1992.0
18	Heysəm	Con	9A	K	1992.0
19	Ramin	Emilzadə	9B	K	1992.0
28	Tərlan	Babakişiyeva	10A	Q	1992.0
29	Şölə	Qazıyev	10B	Q	1992.0
38	Flora	Kənanzadə	9A	Q	1992.0
39	Reyhan	Razi	9B	Q	1992.0

```
df[(df['Sinif'] == '10A') & (df['Təvəllüdü'] == 1991) ]
```

Sıra Nömrəsi	Ad	Soyad	Sinif	Cinsi	Təvəllüdü
5	Mövlud	Hümmətli	10A	K	1991.0
6	Kənan	Altaylı	10A	K	1991.0
25	Pərvanə	Nəsimov	10A	Q	1991.0

Məlum olduğu kimi burada da DataFrame filtrləmə əməliyyatları edə bilirik.

```
def square(x):
    return x ** 2
square(2)
```

```
4
```

Biz burada aldığı parametrin kvadratını yaradıb bizə geri döndürən funksiya yaratdıq.

```
df['Təvəllüdü'].apply(square)
```

```
Sıra Nömrəsi
1     3960100.0
2     3960100.0
3     3960100.0
4     3960100.0
5     3964081.0
6     3964081.0
7     3964081.0
8     3960100.0
9     3968064.0
10    3956121.0
11    3960100.0
12    3960100.0
13    3968064.0
14    3960100.0
15    3964081.0
16    3964081.0
17    3964081.0
18    3968064.0
19    3968064.0
20    3956121.0
21    3960100.0
22    3960100.0
23    3960100.0
24    3960100.0
25    3964081.0
26    3964081.0
27    3964081.0
28    3968064.0
29    3968064.0
30    3956121.0
31    3960100.0
32    3960100.0
33    3960100.0
34    3960100.0
35    3964081.0
36    3964081.0
37    3964081.0
38    3968064.0
39    3968064.0
40          NaN
Name: Təvəllüdü, dtype: float64
```

Bu funksiyanı 'apply()' ilə həmin data sütununa yerləşdirdikdə o, bizə həmin sütunun elementlərinin kavadratını verdi.
Bu xüsusi yazılış qaydasından layihələrinizdə istifadə edə bilərsiniz. Məsələn, siz avtomobilin mühərrik hissəsinin datalarını araşdırırsınız və komandanız sizdən mühərrik haqqında diaqnostik dəyərləndirmə istədi. Datalardan lazım olan xüsusuiyyət (sütun elementlərini) -ləri götürüb riyazi düsturla bu misaldakı kimi istifadə edə və işinizi rahatlıqla görə bilərsiniz. Ancaq daha qısa bir yol var:

```
lambda x : x **2
```

```
<function __main__.<lambda>(x)>
```

```
df['Təvəllüdü'].apply(lambda x : x **2 )
```

```
Sıra Nömrəsi
1     3960100.0
2     3960100.0
3     3960100.0
4     3960100.0
5     3964081.0
6     3964081.0
7     3964081.0
8     3960100.0
9     3968064.0
10    3956121.0
11    3960100.0
12    3960100.0
13    3968064.0
14    3960100.0
15    3964081.0
16    3964081.0
17    3964081.0
18    3968064.0
19    3968064.0
20    3956121.0
21    3960100.0
22    3960100.0
23    3960100.0
24    3960100.0
25    3964081.0
26    3964081.0
27    3964081.0
28    3968064.0
29    3968064.0
30    3956121.0
31    3960100.0
32    3960100.0
33    3960100.0
34    3960100.0
35    3964081.0
36    3964081.0
37    3964081.0
38    3968064.0
39    3968064.0
40          NaN
Name: Təvəllüdü, dtype: float64
```

Adətən, ***lambda*** tipli tək sətirlik funksiyaya qısa olduğu üçün daha çox üstunlük verilir.
Baxmayaraq ki, biz yuxarıda olan formada kodu çalışdırdıq, data avtomatik olaraq yenilənməyəcəkdir. Şərhə aldığım kodda qeyd edildiyi formada biz onu (datanı) yeniləyə bilərik.

```
#  df['Təvəllüdü'] = df['Təvəllüdü'].apply(square)
```

```
df['Soyad'].apply(len) # Sütundakı dataların string uzunluğunu bizə geri döndürər
```

```
Sıra Nömrəsi
1      8
2      8
3      9
4      7
5      8
6      7
7     10
9      8
10     8
12     9
13     7
14     9
15    10
16     8
17     7
18     3
19     8
20     7
21     7
22     9
23    10
24     7
25     7
26     7
27    12
28    12
29     7
30    10
31     7
32     7
33     6
34     4
35     6
36    10
37     6
38     9
39     4
Name: Soyad, dtype: int64
```

```
df.drop('Təvəllüdü', axis = 1 ) #Təvəllüdü sütununu silirik
```

Sıra Nömrəsi	Ad	Soyad	Sinif	Cinsi
1	Ceyms	Teymurov	10A	K
2	Teymur	Akifoğlu	10A	K
3	Qardaşxan	Elşadzadə	10B	K
4	Mahir	Niyazlı	10B	K
5	Mövlud	Hümmətli	10A	K
6	Kənan	Altaylı	10A	K
7	Kazuo	Teymurzadə	10B	K
9	Qardaşxan	Teymurov	10B	K
10	Zaman	Zauroğlu	9A	K
12	Aqşin	Nərimanov	9A	K
13	Niyaz	Kazımov	9B	K
14	Qasımbəy	Elmanoğlu	9B	K
15	Elmar	Abdullahlı	9A	K
16	Rəşid	Emilzadə	9A	K
17	Mark	Vahidli	9B	K
18	Heysəm	Con	9A	K
19	Ramin	Emilzadə	9B	K
20	Xatirə	Ramizli	10B	Q
21	Rusudan	Anatoli	10A	Q
22	Mədinə	Elşadqızı	10A	Q
23	Rəna	Çingizzadə	10B	Q
24	Yekaterina	Əsgərli	10B	Q
25	Pərvanə	Nəsimov	10A	Q
26	Luiza	Elnurlu	10B	Q
27	Lətafət	Babakişiyeva	10B	Q
28	Tərlan	Babakişiyeva	10A	Q
29	Şölə	Qazıyev	10B	Q
30	Gülşən	Sultanzadə	9B	Q
31	Hikari	Mustafa	9A	Q
32	Xədicə	Fahirli	9A	Q
33	Luiza	Bəhram	9B	Q
34	Şəfa	Razi	9B	Q
35	Marina	Edvard	9A	Q
36	Gülüstan	Mikayılova	9B	Q

```
df.index
```

```
Int64Index([ 1,  2,  3,  4,  5,  6,  7,  9, 10, 12, 13, 14, 15, 16, 17, 18, 19,
             20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
             37, 38, 39],
            dtype='int64', name='Sıra Nömrəsi')
```

```
df.index.names
```

```
FrozenList(['Sıra Nömrəsi'])
```

index bizə defolt olaraq əgər indeks təyin edilməzsə index aralıqlarını bizə göstərər (başlanğıc, bitiş, addım) - formasında
Yox əgər index dataframe yaradılarkən təyin edilmişdirsə , təyin edilmiş array (sütun) -u bizə geri döndürər.
index.names index -ə təyin etdiyimiz sütun adını qaytarar.

Parametrlər

by : str və ya str -lardan ibarət list

> Sıralamaq üçün ad və ya adlar siyahısı qəbul edər.
>
> - if `axis` 0 və ya `'index'` -dirsa o zaman `by` index level -ləri və ya sütun başlıqlarını alar column labels.
> - if `axis` is 1 və ya `'columns'` -dirsa o zaman `by` column level -ləri və ya index başlıqları alar.

axis : {0 ovə ya 'index', 1 və ya 'columns'}, default 0

> Sıralamaq üçün axe və ya axislər verilər.

ascending : bool və ya bool -lardan ibarət list, defolt True

> Ascending və ya descending üzrə sıralaya bilərsiz. Çoxölçülü sıralama üçün bool -lardan ibarət list verilər. Amma, bu list by ilə göndərilən siyahı ilə eyni sayda element saxlamalıdır.

inplace : bool, defolt False

> Əgər True təyin edilərsə, əməliyyat in-place, yəni yerində - anındaca həyata heçirilir.

kind : {'quicksort', 'mergesort', 'heapsort', 'stable'}, defolt 'quicksort'

> Sıralama alqoritmi seçilir. Daha çox informasiya üçün həmçinin bax :funksiya: `numpy.sort` .
> `mergesort` və `stable` yeganə stabil alqoritmlərdir. Dataframe -lərdə bu seçim sıralama bir sütun və ya label üzərində tətbiq edilərkən istifadə edilir.

na_position : {'first', 'last'}, default 'last'

> Əgər `first` olarsa NaN dəyərləri başlanğıcda ; `last` olarsa NaNs dəyərləri cədvəl sonunda geri döndürər.

ignore_index : bool, default False

> Əgər True olarasa, nəticədə index `0, 1, …, n - 1` olaraq başlıqla sətirlərdə görünəcəkdir. `n` - burada datadakı sətirlərin sayıdır.

Qaytarır:

> Sıralanmış obyekt : DataFrame və ya None. Sıralanmış dəyərlərdən ibarət DataFrame və ya None əgər `inplace=True` dəyəri olarsa.

```
df.head()
```

Sıra Nömrəsi	Ad	Soyad	Sinif	Cinsi	Təvəllüdü
1	Ceyms	Teymurov	10A	K	1990.0
2	Teymur	Akifoğlu	10A	K	1990.0
3	Qardaşxan	Elşadzadə	10B	K	1990.0
4	Mahir	Niyazlı	10B	K	1990.0
5	Mövlud	Hümmətli	10A	K	1991.0

Sıralamaq üçün ad və ya adlar siyahısı qəbul edərsək:

```
df.sort_values(by=['Ad', 'Soyad'])
```

Sıra Nömrəsi	Ad	Soyad	Sinif	Cinsi	Təvəllüdü
12	Aqşin	Nərimanov	9A	K	1990.0
1	Ceyms	Teymurov	10A	K	1990.0
15	Elmar	Abdullahlı	9A	K	1991.0
38	Flora	Kənanzadə	9A	Q	1992.0
36	Gülüstan	Mikayılova	9B	Q	1991.0
30	Gülşən	Sultanzadə	9B	Q	1989.0
18	Heysəm	Con	9A	K	1992.0
31	Hikari	Mustafa	9A	Q	1990.0
7	Kazuo	Teymurzadə	10B	K	1991.0
6	Kənan	Altaylı	10A	K	1991.0
33	Luiza	Bəhram	9B	Q	1990.0
26	Luiza	Elnurlu	10B	Q	1991.0
27	Lətafət	Babakişiyeva	10B	Q	1991.0
4	Mahir	Niyazlı	10B	K	1990.0
35	Marina	Edvard	9A	Q	1991.0
17	Mark	Vahidli	9B	K	1991.0
5	Mövlud	Hümmətli	10A	K	1991.0
22	Mədinə	Elşadqızı	10A	Q	1990.0
13	Niyaz	Kazımov	9B	K	1992.0
25	Pərvanə	Nəsimov	10A	Q	1991.0
3	Qardaşxan	Elşadzadə	10B	K	1990.0
9	Qardaşxan	Teymurov	10B	K	1992.0
14	Qasımbəy	Elmanoğlu	9B	K	1990.0
19	Ramin	Emilzadə	9B	K	1992.0
39	Reyhan	Razi	9B	Q	1992.0
37	Rusudan	Anarlı	9B	Q	1991.0
21	Rusudan	Anatoli	10A	Q	1990.0
23	Rəna	Çingizzadə	10B	Q	1990.0
16	Rəşid	Emilzadə	9A	K	1991.0
2	Teymur	Akifoğlu	10A	K	1990.0
28	Tərlan	Babakişiyeva	10A	Q	1992.0
20	Xatirə	Ramizli	10B	Q	1989.0
32	Xədicə	Fahirli	9A	Q	1990.0
24	Yekaterina	Əsgərli	10B	Q	1990.0
10	Zaman	Zauroğlu	9A	K	1989.0
29	Şölə	Qazıyev	10B	Q	1992.0
34	Şəfa	Razi	9B	Q	1990.0

İlk öncə Ad sütununu əsas alaraq sıralayar.
Daha sonra, Soyad üzrə Hərf sıralaması səviyyələr üzrə aparar.

16 Elmar Abdullahlı 9A K 1991
9 Elmir Davudov 10A K 1992

Sıralamasından bunu aydın şəkildə görmək olur.

```
df.sort_values('Təvəllüdü', ascending = True) # kiçikdən böyüyə
```

```
df.sort_values('Təvəllüdü', ascending = False) # böyükdən kiçiyə
```

```
df.sort_values('Soyad', kind = 'heapsort')
```

```
df.sort_values('Ad', kind = 'mergesort')
```

```
df.sort_values('Sinif', kind = 'quicksort')
```

NaN dəyərləri ad sütunun ilk elementləri kimi sıralayaq:

```
df.sort_values('Ad', na_position = 'first')
```

NaN dəyərləri soyadad sütunun son elementləri kimi sıralayaq:

```
df.sort_values('Soyad', na_position = 'last')
```

Pivot Table (Yekunlar Cədvəli)

Əgər Exceli bilirsinizsə burda çəkinməyə heç bir əsas yoxdur. Çünki, buradakı anlayışlar tamamilə eynidir. Əgər bilmirsinizsə çox da narahat olmayın. Çünki istifadəsi çox sadədir.

```
df_2 = pd.DataFrame({'Aylar': ['January','February','March','January','February',
                               'March','January','February','March'],
                     'Ştat':['New York','New York','New York','Texas','Texas',
                             'Texas','Washington','Washington','Washington'],
                     'Nəmlik': [20,25,65,34,56,85,21,56,79]})
df_2
```

	Aylar	Ştat	Nəmlik
0	January	New York	20
1	February	New York	25
2	March	New York	65
3	January	Texas	34
4	February	Texas	56
5	March	Texas	85
6	January	Washington	21
7	February	Washington	56
8	March	Washington	79

Başlamazdan öncə corr() metodu haqqında ilkin fikir formalaşdıraq və aşağıdakı kodda onu çalışdıraq, bu metod bizə korrelasiya əlaqəsi haqqında fikir verir:

```
df_2.corr()
```

	Nəmlik
Nəmlik	1.0

İndi isə, 3 ədəd faiz nisbəti ilə ifadə edilmiş 3 ay üzrə 3 müxtəlif ştatdakı Nəmlik miqdarını Yekunlar cədvəlində ifadə edək.

```
df_2.pivot_table(index = 'Aylar', columns = 'Ştat', values = 'Nəmlik')
```

Ştat	New York	Texas	Washington
Aylar			
February	25	56	56
January	20	34	21
March	65	85	79

```
df_2.pivot_table(index = 'Ştat', columns = 'Aylar', values = 'Nəmlik')
```

Aylar	February	January	March
Ştat			
New York	25	20	65
Texas	56	34	85
Washington	56	21	79

Yuxarıdakı formada Pivot Table (Yekunlar Cədvəli) -ni asanlıqla yarada bilərsiniz.

DataSeti oxuma metodları

Bu məsələni aşağıdakı kimi ifadə edə bilərəm:

Kompüternizdə IDE -niz vasitəsi ilə bu kodu işlədə bilərsiniz

dataset = pd.read_csv ('File_Path \ Data_Name.csv')

Kaggle -da kernel yazmaq üçün (ancaq, bəzən aparılan yeniləmələr zamanı bu metod dəyişə bilər.)

df = pd.read_csv ("../ input / Data_Name.csv")

Dataframe -ni CSV formatına qaytarmaq üçün
dataset.to_csv ('Data_Name')
Excel faylı oxumaq üçün ilk öncə xlrd (XLRD) paketi yüklənilməlidir. Daha sonra bu kodu rahatlıqla istifadə edə bilərsiniz.
pip install xlrd
excelset = pd.read_excel ('Excels_Name.xlsx')
Mövcud Dataseti Excel faylına çevirir:
excelset.to_excel ('excelnewfile.xlsx')
Mövcud Dataseti Excel faylına çevirir:
excelset.to_excel ('excelnewfile.xlsx')
İnternettən xüsusilə HTML fayldan datanı oxumaq üçün bu formada kodu çalışdıra bilərsiniz:
New = pd.read_html ('Datasets URL.html')

Ydata_profiling

Normalda burada mövzunu bitirərdim, amma kifayət qədər aktual olduğunu düşündüyüm və son vaxtlar istifadəsi geniş vüsət alacaq bir xüsusiyyəti sizinlə bölüşmək istəyirəm: ***'Ydata_Profiling()'***

- İlk öncə modulu yükləyək :
  pip install ydata_profiling ilə paketi kompüternizə quraşdırın,
- daha sonra import edək:

```
!pip install ydata_profiling
```

```
import ydata_profiling as ydp
```

Datamızı hazır datasetlərdən olan diabetes datasetindən alaq:

```
import warnings
warnings.filterwarnings("ignore")
from sklearn.datasets import load_diabetes
diabetes = load_diabetes()
df_diabetes = pd.DataFrame(data = diabetes.data, columns = diabetes.feature_names)
df_diabetes.head()
```

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019908	-0.017646
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068330	-0.092204
2	0.085299	0.050680	0.044451	-0.005671	-0.045599	-0.034194	-0.032356	-0.002592	0.002864	-0.025930
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022692	-0.009362
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031991	-0.046641

Profiling report data analitiklərin üzərində çalışa biləcəyi bir hesabatı bizə geri qaytarar:

```
ydp.ProfileReport(df_diabetes)
```

Overview

Overview Alerts 5 Reproduction

Dataset statistics

Number of variables	10
Number of observations	442
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	34.7 KiB
Average record size in memory	80.3 B

Variable types

Numeric	9
Categorical	1

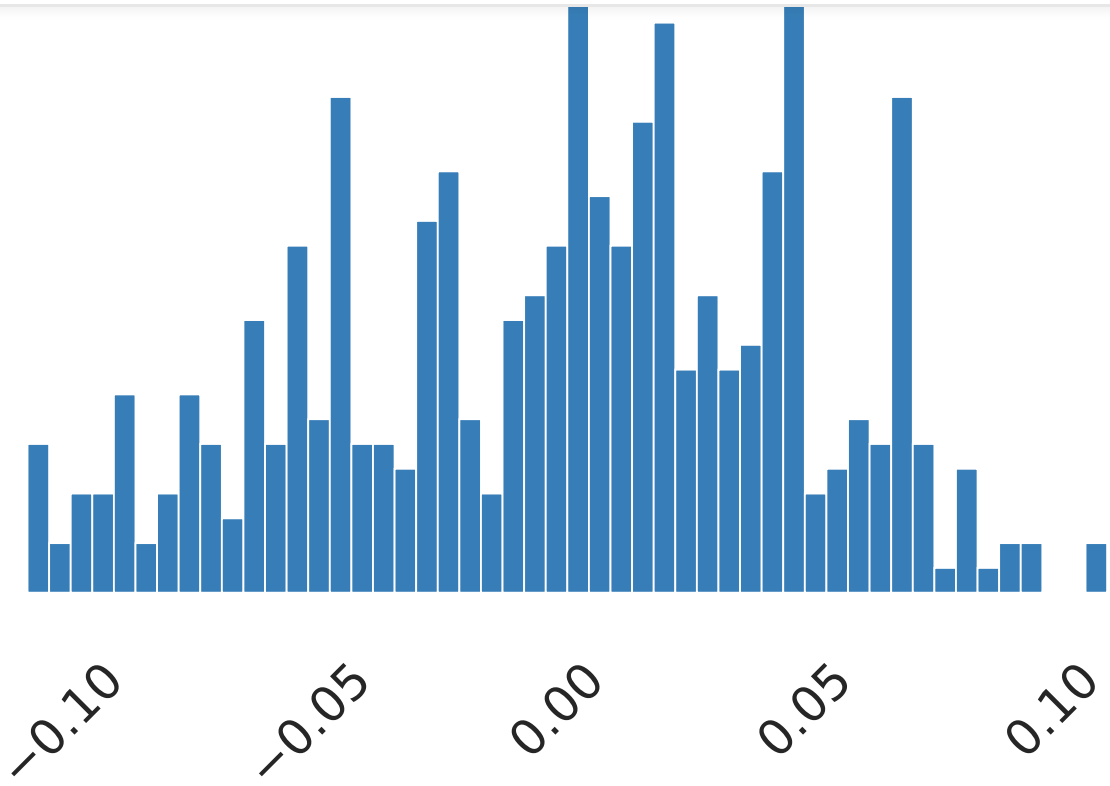
Variables

age

Real number (ℝ)

Distinct	58
Distinct (%)	13.1%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	$-3.6389946 \times 10^{-16}$

Minimum	-0.10722563
Maximum	0.11072668
Zeros	0
Zeros (%)	0.0%
Negative	202
Negative (%)	45.7%
Memory size	3.6 KiB



More details

sex

Categorical

Distinct	2
Distinct (%)	0.5%
Missing	0
Missing (%)	0.0%
Memory size	3.6 KiB

-0.04464163…	235
0.050680118…	207

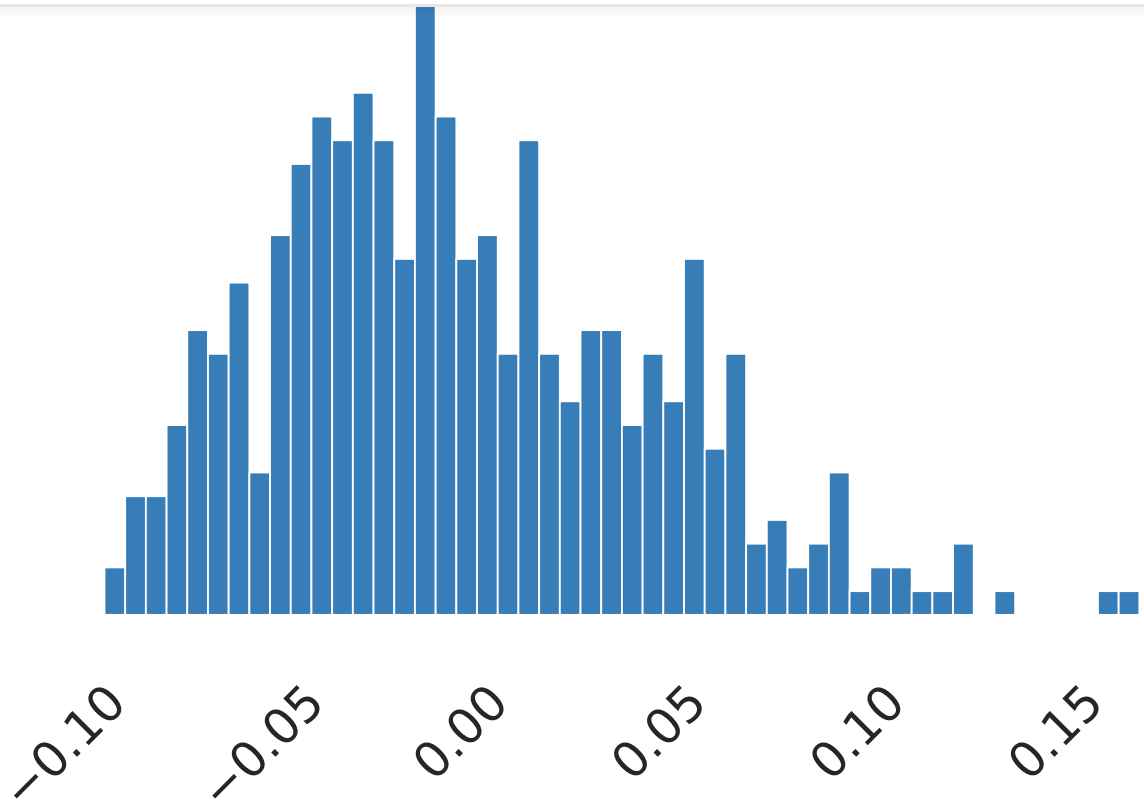
More details

bmi

Real number (ℝ)

Distinct	163
Distinct (%)	36.9%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	$-8.0227429 \times 10^{-16}$

Minimum	-0.090275296
Maximum	0.17055523
Zeros	0
Zeros (%)	0.0%
Negative	247
Negative (%)	55.9%
Memory size	3.6 KiB



More details

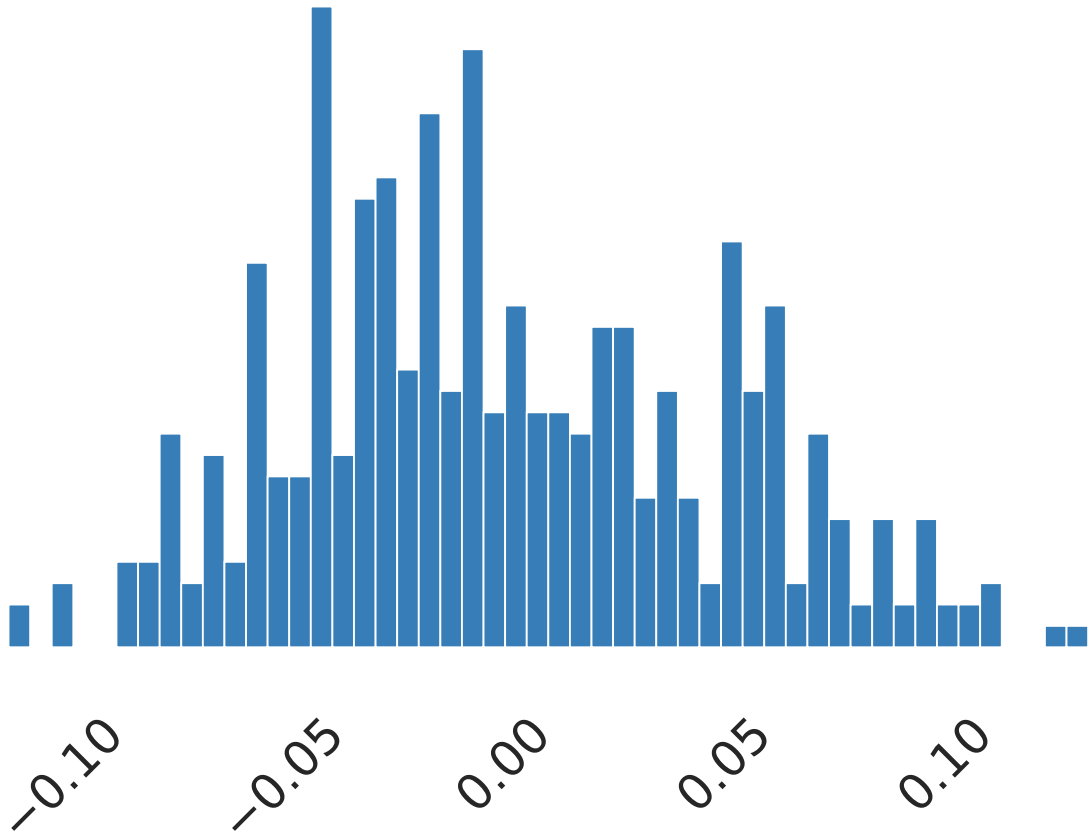
bp

Real number (ℝ)

Distinct	100
Distinct (%)	22.6%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	$1.2798492 \times 10^{-16}$

Minimum	-0.1123996

Zeros (%)	0.0%
Negative	244
Negative (%)	55.2%
Memory size	3.6 KiB



More details

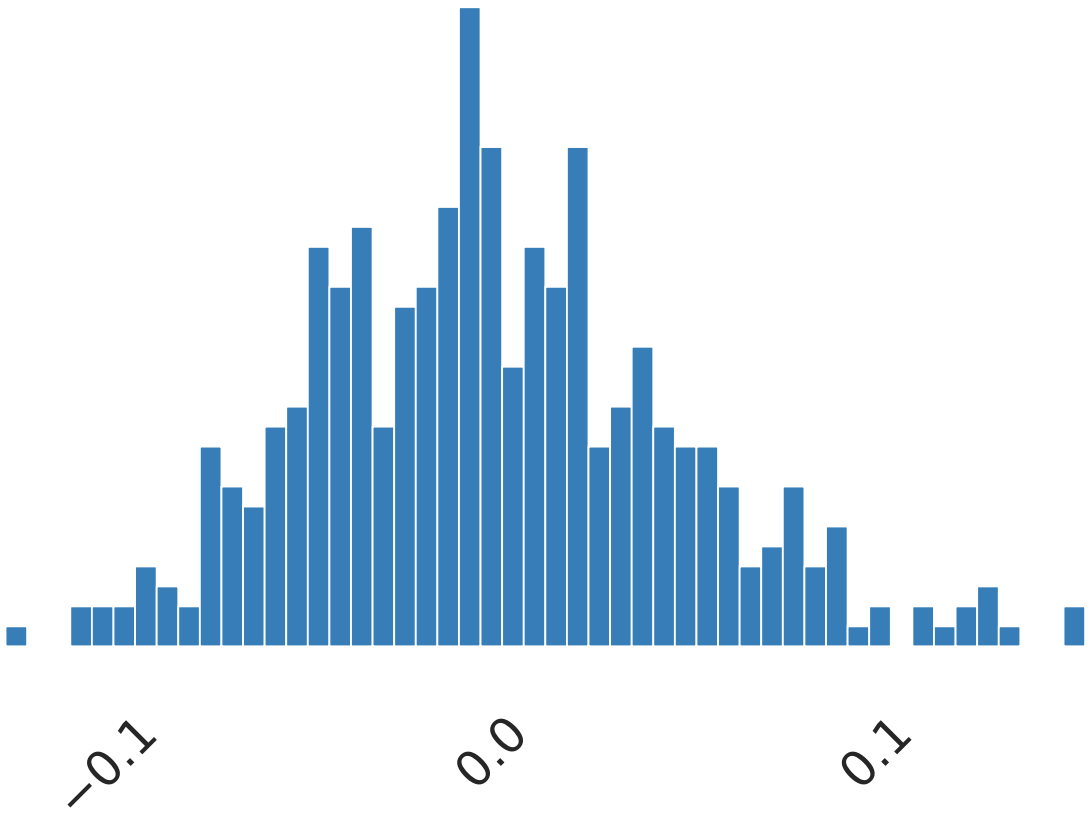
s1

Real number (ℝ)

Distinct	141
Distinct (%)	31.9%
Missing	0

Infinite	0
Infinite (%)	0.0%
Mean	$-9.0425405 \times 10^{-17}$

Minimum	-0.12678067
Maximum	0.15391371
Zeros	0
Zeros (%)	0.0%
Negative	240
Negative (%)	54.3%
Memory size	3.6 KiB

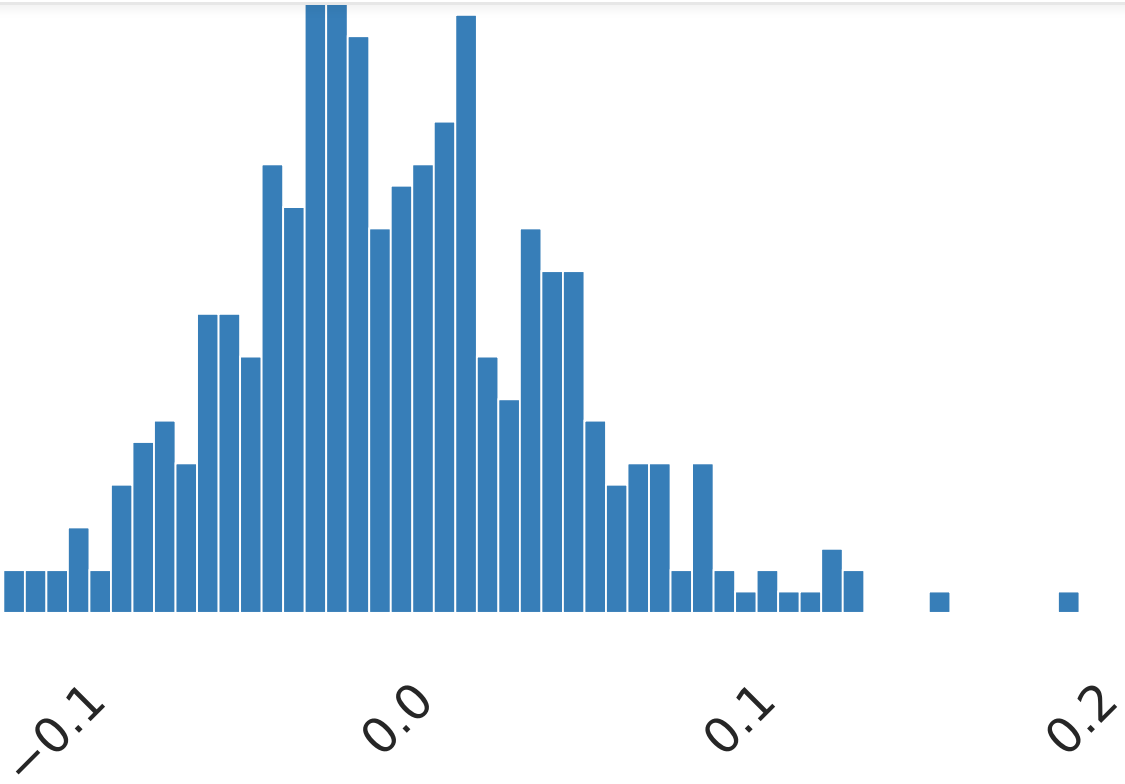


More details

Real number (ℝ)

Distinct	302
Distinct (%)	68.3%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	1.303005×10^{-16}

Minimum	-0.11561307
Maximum	0.19878799
Zeros	0
Zeros (%)	0.0%
Negative	239
Negative (%)	54.1%
Memory size	3.6 KiB



More details

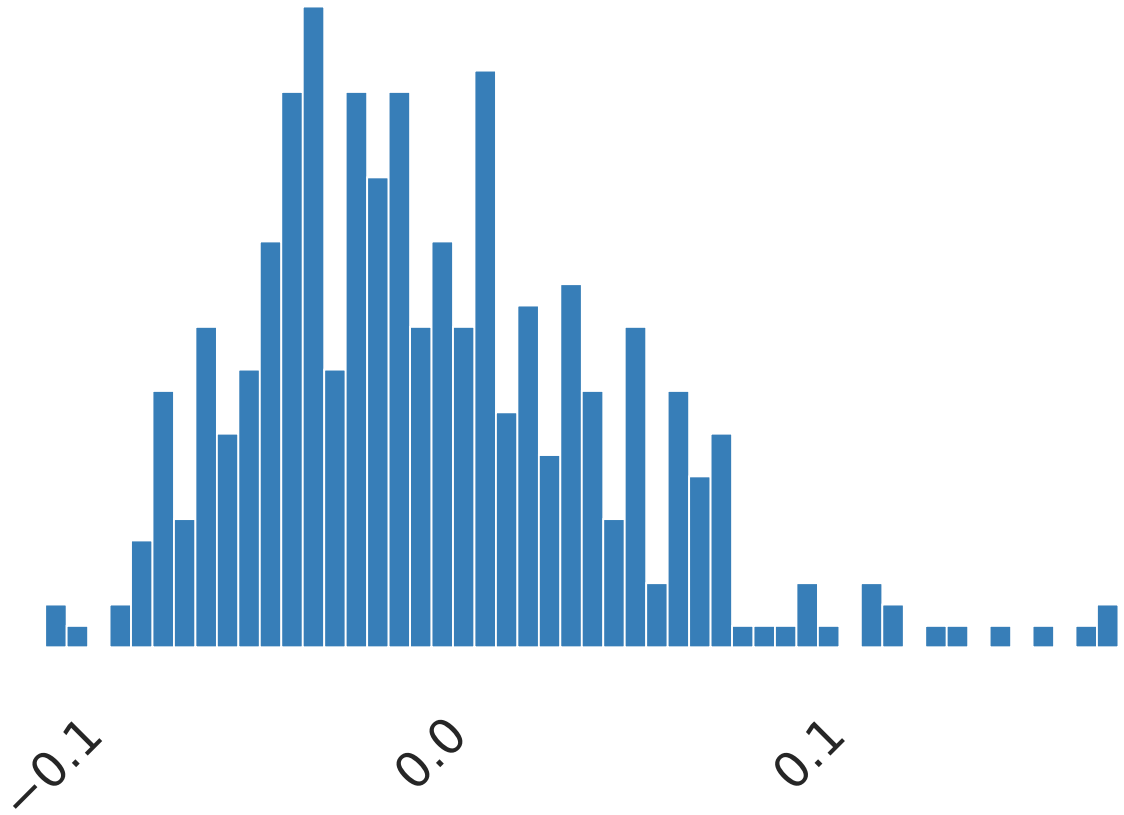
s3

Real number (ℝ)

Distinct	63
Distinct (%)	14.3%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	$-4.5583195 \times 10^{-16}$

Minimum	-0.10230705

Zeros (%)	0.0%
Negative	243
Negative (%)	55.0%
Memory size	3.6 KiB



More details

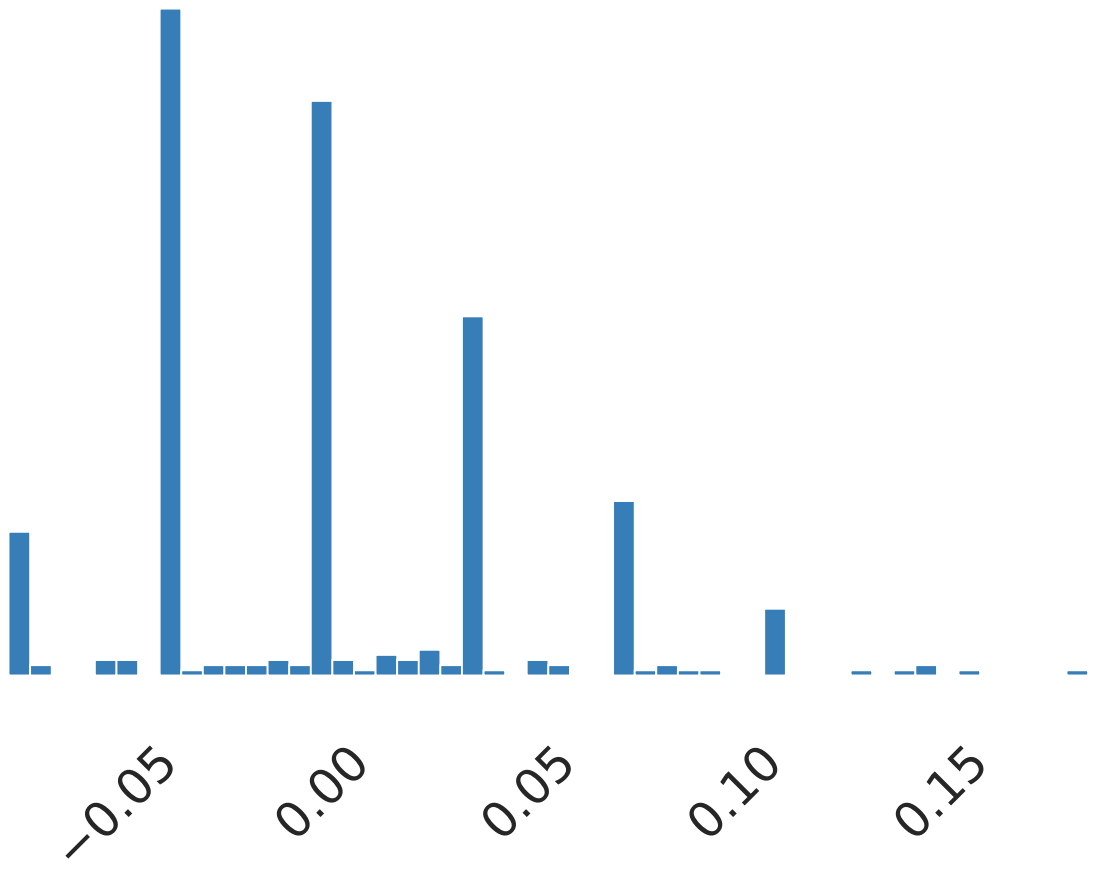
s4

Real number (ℝ)

Distinct	66
Distinct (%)	14.9%
Missing	0

Infinite	0
Infinite (%)	0.0%
Mean	$3.8623893 \times 10^{-16}$

Minimum	-0.076394504
Maximum	0.18523444
Zeros	0
Zeros (%)	0.0%
Negative	288
Negative (%)	65.2%
Memory size	3.6 KiB

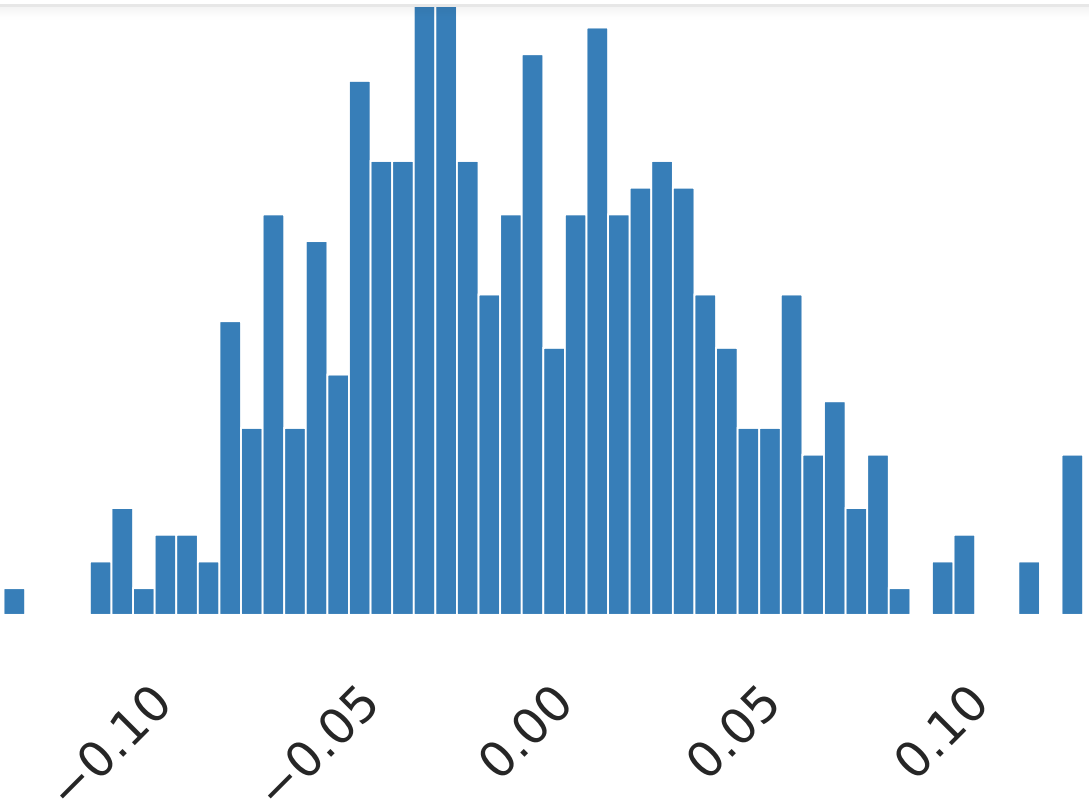


More details

Real number (ℝ)

Distinct	184
Distinct (%)	41.6%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	$-3.8280088 \times 10^{-16}$

Minimum	-0.12609739
Maximum	0.13359898
Zeros	0
Zeros (%)	0.0%
Negative	230
Negative (%)	52.0%
Memory size	3.6 KiB



More details

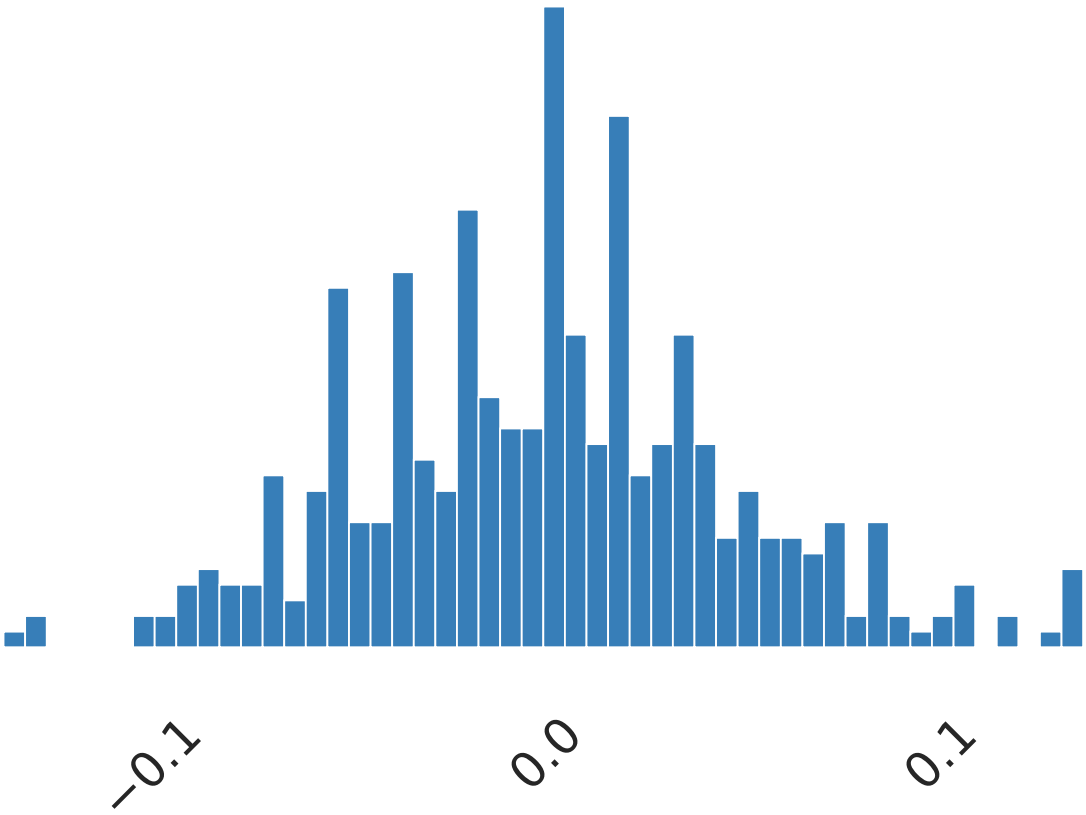
s6

Real number (ℝ)

Distinct	56
Distinct (%)	12.7%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	$-3.4009999 \times 10^{-16}$

Minimum	-0.13776723

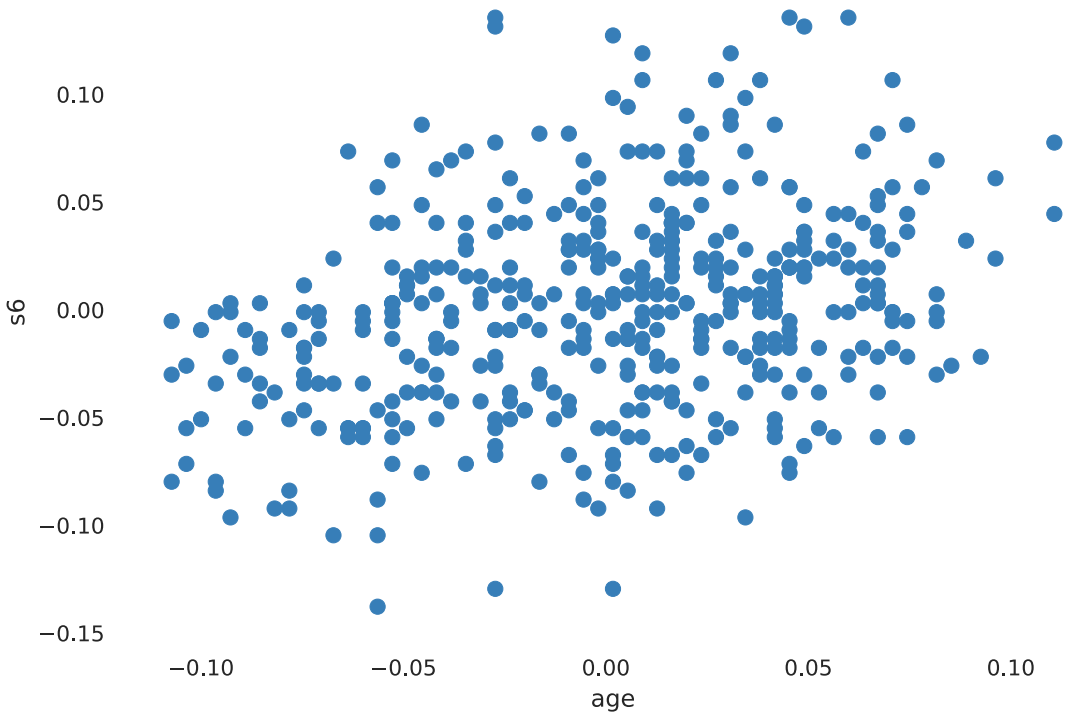
Zeros (%)	0.0%
Negative	224
Negative (%)	50.7%
Memory size	3.6 KiB



More details

Interactions

age | bmi | bp | s1 | s2 | s3 | s4 | s5 | s6



Bu data analitiklərin əldə edə biləcəyi ən gözəl alətlərdən biridir və bizə qalan yalnızca istifadə etməkdir.

Son söz

Bəli, dost. Bura qədər gəlib çatmısınızsa və haqqınızı vermişsinizsə, özünüzə "Mən Pandas -ı yaxşı səviyyədə bilirəm" deyə bilərsiniz. Amma bununla kifayətlənməməli, bir çox müxtəlif mənbələri araşdırıb öyrənməlisən.

Xüsusi Təşəkkürlər: Batuhan Bayraktar
Yazar: Nuhbalayev Ramazan
Kod Mənbəsi:
A'dan Z'ye Pandas Tutoriali (Başlangıç ve Orta Seviye və Kaggle: batuhan35