Практическая работа №6

In [31]:
```
import numpy as np
import pandas as pd

df = pd.read_csv('../football.csv')

A = np.mean(df['Composure'])
B = df[['Nationality','Penalties']]
C = df.groupby('Nationality').agg({'Penalties':'sum', 'Composure': 'mean'})
G = C.loc[C['Composure'] > A]
G
```

Out[31]:

Nationality	Penalties	Composure
Afghanistan	206	57.250000
Albania	1346	59.413793
Algeria	1925	64.486486
Angola	572	59.500000
Argentina	28573	57.491289
...	...	...
Uruguay	4156	60.891566
Uzbekistan	106	59.500000
Venezuela	1836	57.083333
Zambia	337	56.333333
Zimbabwe	449	63.111111

94 rows × 2 columns

In [6]:
```
df['Composure'].describe()
```

Out[6]:
```
count    12897.000000
mean        55.942932
std         11.280631
min         12.000000
25%         49.000000
50%         56.000000
75%         63.000000
max         96.000000
Name: Composure, dtype: float64
```

Практическая работа №7

задание A

In [44]:
```
import numpy as np
import pandas as pd

df = pd.read_csv('../StudentsPerformance.csv')

g = list(df['parental level of education'])
g=set(g)
print(g)
len(g)
```

```
{'some college', 'some high school', "master's degree", "bachelor's degree", "associate's degree", 'high school'}
```

Out[44]:
```
6
```

задание B

In [101]:
```
def concatinate(row):
    f=list(row[2])
    s=list(row[3])
    fst=str()
    g=0
    k=0
    for i in f:
        if i != ' ' and g<5:
            fst += i
            g+=1
    for i in range(-5,len(s)):
        if s[i] != ' ' and k<5:
            fst += str(s[i])
            k+=1
    return fst

conc = map(lambda x: concatinate(list(df.iloc[x])), range(len(df['parental level of education'])))

df['concatinate'] = list(conc)
df
```

Out[101]:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	concatinate
0	female	group B	bachelor's degree	standard	none	72	72	74	bachendard
1	female	group C	some college	standard	completed	69	90	88	somecndard
2	female	group B	master's degree	standard	none	90	95	93	mastendard
3	male	group A	associate's degree	free/reduced	none	47	57	44	assocduced
4	male	group C	some college	standard	none	76	78	75	somecndard
...	...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88	99	95	mastendard
996	male	group C	high school	free/reduced	none	62	55	55	highsduced
997	female	group C	high school	free/reduced	completed	59	71	65	highsduced
998	female	group D	some college	standard	completed	68	78	77	somecndard
999	female	group D	some college	free/reduced	none	77	86	86	somecduced

1000 rows × 9 columns

Практическая работа №8

Задание

Самый убыточный фильм за период с 2012 по 2014 гг. (включительно)?

In [163]:
```
import pandas as pd
import numpy as np

def cost_cast(budget, revenue):
    c = revenue - budget
    return c


df = pd.read_csv('../films.csv')
```

In [164]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1890 entries, 0 to 1889
Data columns (total 16 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   imdb_id               1890 non-null   object 
 1   popularity            1890 non-null   float64
 2   budget                1890 non-null   int64  
 3   revenue               1890 non-null   int64  
 4   original_title        1890 non-null   object 
 5   cast                  1890 non-null   object 
 6   director              1890 non-null   object 
 7   tagline               1890 non-null   object 
 8   overview              1890 non-null   object 
 9   runtime               1890 non-null   int64  
 10  genres                1890 non-null   object 
 11  production_companies  1890 non-null   object 
 12  release_date          1890 non-null   object 
 13  vote_count            1890 non-null   int64  
 14  vote_average          1890 non-null   float64
 15  release_year          1890 non-null   int64  
dtypes: float64(2), int64(5), object(9)
memory usage: 236.4+ KB
```

In [165]:
```
1 df = df.loc[df['release_year'] <= 2014]
2 df = df.loc[df['release_year'] >= 2012]
3 
4 df.loc[:,'cash'] = list(map(cost_cast, df['budget'], df['revenue']))
5 
6 b =min(df['cash'])
7 
8 
9 df['original_title'].loc[df['cash']==b]
```

Out[165]:
```
1246    The Lone Ranger
Name: original_title, dtype: object
```

In [ ]:
```
1 
```