Практическая работа №5

In [1]:
```
import numpy as np #Библиотека NumPy. Импорт библиотеки
#Создание массива. Функции array() и values().

F = np.random.randint(0,101,(4,4))
print(F)
G = np.transpose(F)

b = [sum(G[x]) for x in range(len(G))]
print(b)
index = b.index(min(b))
print(min(b))
mid = np.median(G[index])
print(mid, 'стандартная функция')



srt = np.sort(G[index])
coord = len(srt)//2
if len(G[index])%2==0:
    mean = (srt[coord]+srt[coord-1])/2
    print(mean, 'собственная функция')
else:
    print(srt[coord], 'собственная функция')
```

```
[[61 88 82 29]
 [30 52 63 64]
 [47 50 23  6]
 [31 44 94 69]]
[169, 234, 262, 168]
168
46.5 стандартная функция
46.5 собственная функция
```

Практическая работа №7

задание А

In [56]:
```
import numpy as np
import pandas as pd

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

g = list(map(lambda x,y: x == 'female' and y == "master's degree",
             df['gender'], df['parental level of education']))
c = filter(lambda x: x == True, g)
print(len(list(c)), 'девочек, у которых родители имеют степень баклавра')
```

```
36 девочек, у которых родители имеют степень баклавра
```

задание В

In [84]:
```
def cat(row):
    if row[2].find('degree')!=-1:
        return 'высшая категория'
    elif row[2].find('school')!=-1 and row[2].find('some')!=-1:
        return 'средняя категория'
    elif row[2].find('school')!=-1 and row[2].find('some')==-1:
        return 'выше среднего'
    elif row[2].find('college')!=-1:
        return 'низшая'


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

df['education_cat'] = list(cat_ed)
df
```

Out[84]:

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

1000 rows × 9 columns

Практическая работа №8

Задание

У какого из режиссеров самый высокий процент фильмов со сборами выше бюджета?

In [274]:
```
import pandas as pd
import numpy as np

def luck_film(budget, revenue):
    c = revenue - budget
    if c>0:
        return True
    else:
        return False

def percent_luck(row):
    if row[1] == 0:
        return row[0]/1*100
    else:
        return row[0]/(row[0]+row[1])*100

def analog_luck(row):
    return row[0]/(row[0]+row[1])*100
df = pd.read_csv('../films.csv')
```

In [275]:
```
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 [276]:
```
df.loc[:,'luck'] = list(map(luck_film, df['budget'], df['revenue']))



df1 = df.groupby('luck').get_group(True)
df2 = df.groupby('luck').get_group(False)
df1 = df1.groupby('director').agg({'luck':'value_counts'})
df2 = df2.groupby('director').agg({'luck':'value_counts'})
df1 = df1.reset_index(level='luck',drop=True)
df2 = df2.reset_index(level='luck',drop=True)


df1.columns=['value_luck']
df2.columns=['value_unluck']

df2

```

Out[276]:

director	value_unluck
Adam Shankman	1
Adam Wingard	1
Akiva Goldsman	1
Alan Parker	1
Alejandro AmenÃ¡bar	1
...	...
William Dear	1
William Friedkin	2
William Malone	1
William Monahan	1
Zal Batmanglij	1

354 rows × 1 columns

In [278]:
```
df3 = pd.concat([df1, df2], ignore_index=False)
df3['director'] = df3.index
df3 = df3.reset_index(level='director',drop=True)
df3 = df3.fillna(0)
listing = map(lambda x: percent_luck(list(df3.iloc[x])), range(len(df3['director'])))
df3['listing'] = list(listing)
m = max(df3['listing'])
df3.loc[df3['listing']==m]
```

Out[278]:

	value_luck	value_unluck	director	listing
596	12.0	0.0	Ridley Scott	1200.0

In [270]:
```
df3
```

Out[270]:

	value_luck	value_unluck	director	listing
0	1.0	0.0	Aaron Seltzer|Jason Friedberg	100.0
1	6.0	0.0	Adam McKay	600.0
2	7.0	0.0	Adam Shankman	700.0
3	1.0	0.0	Adrian Lyne	100.0
4	1.0	0.0	Alan Poul	100.0
...	...	...	...	...
1123	0.0	1.0	William Dear	0.0
1124	0.0	2.0	William Friedkin	0.0
1125	0.0	1.0	William Malone	0.0
1126	0.0	1.0	William Monahan	0.0
1127	0.0	1.0	Zal Batmanglij	0.0

1128 rows × 4 columns

In [ ]:
```

```