Задание 1

Задание №1

С помощью модуля ***pandas*** выведите статистику погибших/выживших отдельно для мужчин и женщин в каждом классе (*Pclass*).

```
import pandas as pd
data = pd.read_csv('train.csv')

# array of arrays
# stat[Pclass-1] = [dead_male, alive_male, dead_female, alive_female]
stat = [[0,0,0,0] for _ in range(3)]

for row in data.iterrows():
    row = row[1]
    if row.Sex == 'male':
        if row.Survived == 0:
            stat[row.Pclass-1][0] += 1
        else:
            stat[row.Pclass - 1][1] += 1
    else:
        if row.Survived == 0:
            stat[row.Pclass-1][2] += 1
        else:
            stat[row.Pclass - 1][3] += 1

for i in range(3):
    Pclass = stat[i]
    print('Pclass', i+1)
    print('Male:\n\tdead - %d\n\talive - %d' % (Pclass[0],
Pclass[1]))
    print('Female:\n\tdead - %d\n\talive - %d' % (Pclass[2],
Pclass[3]))
```

```
Pclass 1
Male:
    dead - 77
    alive - 45
Female:
    dead - 3
    alive - 91


Pclass 2
Male:
    dead - 91
    alive - 17
Female:
    dead - 6
    alive - 70


Pclass 3
Male:
    dead - 300
    alive - 47
Female:
    dead - 72
    alive - 72
```

Задание 2

Задание №2

С помощью модуля ***pandas*** выведите статистику по всем числовым полям, отдельно для мужчин и женщин.

```
import pandas as pd
data = pd.read_csv('train.csv')

males = data[data.Sex == 'male']
females = data[data.Sex == 'female']


print('                                              Male\n\n')
print(males.describe())

print('                                            Female\n\n')
print(females.describe())
```

```
                                Male


       PassengerId    Survived      Pclass  ...       SibSp       Parch        Fare
count   577.000000  577.000000  577.000000  ...  577.000000  577.000000  577.000000
mean    454.147314    0.188908    2.389948  ...    0.429809    0.235702   25.523893
std     257.486139    0.391775    0.813580  ...    1.061811    0.612294   43.138263
min       1.000000    0.000000    1.000000  ...    0.000000    0.000000    0.000000
25%     222.000000    0.000000    2.000000  ...    0.000000    0.000000    7.895800
50%     464.000000    0.000000    3.000000  ...    0.000000    0.000000   10.500000
75%     680.000000    0.000000    3.000000  ...    0.000000    0.000000   26.550000
max     891.000000    1.000000    3.000000  ...    8.000000    5.000000  512.329200

[8 rows x 7 columns]
                              Female


       PassengerId    Survived      Pclass  ...       SibSp       Parch        Fare
count   314.000000  314.000000  314.000000  ...  314.000000  314.000000  314.000000
mean    431.028662    0.742038    2.159236  ...    0.694268    0.649682   44.479818
std     256.846324    0.438211    0.857290  ...    1.156520    1.022846   57.997698
min       2.000000    0.000000    1.000000  ...    0.000000    0.000000    6.750000
25%     231.750000    0.000000    1.000000  ...    0.000000    0.000000   12.071875
50%     414.500000    1.000000    2.000000  ...    0.000000    0.000000   23.000000
75%     641.250000    1.000000    3.000000  ...    1.000000    1.000000   55.000000
max     889.000000    1.000000    3.000000  ...    8.000000    6.000000  512.329200

[8 rows x 7 columns]
```

Задание 3

Задание №3

Влияет ли порт посадки на выживаемость?

```
import pandas as pd
data = pd.read_csv('train.csv')

ports = ['C', 'Q', 'S']
# people = { port : [dead, alive] }
people = {'C': [0, 0], 'Q': [0, 0], 'S': [0, 0]}

for row in data.iterrows():
    row = row[1]
    if row.Embarked not in ports: # is NaN
        continue
    if row.Survived:
        people[row.Embarked][1] += 1
    else:
        people[row.Embarked][0] += 1

sum_dead = 0
for v in people.values():
    sum_dead += v[0]

for k,v in people.items():
    print(k, ':')
    print('\tdead - %s' % v[0])
    print('\talive - %s' % v[1])
    print('\tdead from all from this port %.2f%%' % (v[0] / (v[0] +
v[1]) * 100))
```

```
C :
    dead - 75
    alive - 93
    dead from all from this port 44.64%
Q :
    dead - 47
    alive - 30
    dead from all from this port 61.04%
S :
    dead - 427
    alive - 217
    dead from all from this port 66.30%
```

Можно заключить, что среди людей из порта С выживало больше, чем погибло. Чего нельзя сказать про остальные порты. Поэтому в некотором роде можно сказать, что порт влияет на выживаемость.

Задание 4

Задание №4

Выведите топ 10 популярных имён.

```
import pandas as pd
data = pd.read_csv('train.csv')

names = {}

for row in data.iterrows():
    row = row[1]
    name = row.Name.split(',')[0]
    names[name] = names.setdefault(name, 0) + 1

list_d = list(names.items())
list_d.sort(key=lambda i: i[1])

for i in range(10):
    print(list_d[-i - 1][0], ' - ', list_d[-i-1][1], ' times')
```

```
Andersson  -  9  times
Sage  -  7  times
Carter  -  6  times
Skoog  -  6  times
Goodwin  -  6  times
Panula  -  6  times
Johnson  -  6  times
Rice  -  5  times
Baclini  -  4  times
Hart  -  4  times
```

Задание 5

Задание №5

Заполните все отсутствующие в *train.csv* значения медианой (*по столбцу*).

```
import pandas as pd
data = pd.read_csv('train.csv')

print('Посмотрим какие столбцы вообще содержат NaN:')
print(data.loc[:, data.isnull().any()])

print('Видим, что всего 3 столбца содержат значения NaN')
print('Но 2 из них содержат текст, для них медианное значение не
актуально')
print('Поэтому остается только Age - заполним его медианным
значением\n')
data['Age'] = data['Age'].fillna(data['Age'].median())

print('Убедимся, что Age больше не содержит NaN')
print(data.isnull().any())
```

```
Посмотрим какие столбцы вообще содержат NaN:
      Age Cabin Embarked
0    22.0   NaN        S
1    38.0   C85        C
2    26.0   NaN        S
3    35.0  C123        S
4    35.0   NaN        S
..    ...   ...      ...
886  27.0   NaN        S
887  19.0   B42        S
888   NaN   NaN        S
889  26.0  C148        C
890  32.0   NaN        Q

[891 rows x 3 columns]
Видим, что всего 3 столбца содержат значения NaN
Но 2 из них содержат текст, для них медианное значение не актуально
Поэтому остается только Age - заполним его медианным значением
```

```
Убедимся, что Age больше не содержит NaN
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age            False
SibSp          False
Parch          False
Ticket         False
Fare           False
Cabin           True
Embarked        True
dtype: bool
```

Задание 6

Задание №6

На основе статистики попытайтесь предсказать выживаемость для пассажиров из файла *test.csv*.

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier


pd.options.mode.chained_assignment = None

data = pd.read_csv('train.csv')
data['Age'] = data['Age'].fillna(data['Age'].median())
data_to_train = data[['Pclass', 'Age', 'Sex']]
data_to_train['Sex'] = np.where(data_to_train['Sex'] == 'female', 0,
1)

test = pd.read_csv('test.csv')
test['Age'] = test['Age'].fillna(test['Age'].median())
data_to_test = test[['Pclass', 'Age', 'Sex']]
data_to_test['Sex'] = np.where(data_to_test['Sex'] == 'female', 0, 1)

expected_data = data['Survived']

rf = RandomForestClassifier(n_estimators=100)
rf.fit(data_to_train, expected_data)
pred = rf.predict(data_to_test)
with open('output.txt', 'w') as f:
    for p in pred:
        f.write(str(p) + '\n')
```

Результаты в файле output.txt