Задание 1.
Исходные данные:
Какое соотношение мужчин и женщин в представленном наборе данных?
Решение:

```
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>> df.head()
   customerID  gender  SeniorCitizen  ... MonthlyCharges TotalCharges  Churn
0  7590-VHVEG  Female              0  ...          29.85         29.85     No
1  5575-GNVDE    Male              0  ...          56.95        1889.5     No
2  3668-QPYBK    Male              0  ...          53.85        108.15    Yes
3  7795-CFOCW    Male              0  ...          42.30       1840.75     No
4  9237-HQITU  Female              0  ...          70.70        151.65    Yes

[5 rows x 21 columns]
>>> df.shape
(7043, 21)
>>> df.dtypes
customerID          object
gender              object
SeniorCitizen        int64
Partner             object
Dependents          object
tenure               int64
PhoneService        object
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         object
StreamingMovies     object
Contract            object
PaperlessBilling    object
PaymentMethod       object
MonthlyCharges     float64
TotalCharges        object
Churn               object
dtype: object
>>> df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
```

```
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null object
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null object
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
TotalCharges        7043 non-null object
Churn               7043 non-null object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
>>> df[['gender']]
      gender
0     Female
1       Male
2       Male
3       Male
4     Female
...      ...
7038    Male
7039  Female
7040  Female
7041    Male
7042    Male

[7043 rows x 1 columns]
>>> df[['gender']].shape
(7043, 1)
>>> df[['gender']].isna().sum()
gender    0
dtype: int64
>>> df['gender'][df['gender'].sort_values() == 'Female']
0       Female
4       Female
5       Female
7       Female
8       Female
        ...
7034    Female
7036    Female
7037    Female
7039    Female
7040    Female
Name: gender, Length: 3488, dtype: object
```

```
>>> df['gender'][df['gender'].sort_values() == 'Male']
1       Male
2       Male
3       Male
6       Male
9       Male
       ...
7033    Male
7035    Male
7038    Male
7041    Male
7042    Male
Name: gender, Length: 3555, dtype: object
>>> wuman = 3488 / 7043
>>> man = 3555 / 7043
>>> print(wuman, man)
0.495243504188556 0.504756495811444
>>>
```

Задание 2.
Исходные данные:
Какое количество уникальных значений у поля InternetService?
Решение:

```
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>> df.head()
   customerID  gender  SeniorCitizen  ... MonthlyCharges TotalCharges  Churn
0  7590-VHVEG  Female              0  ...          29.85        29.85     No
1  5575-GNVDE    Male              0  ...          56.95       1889.5     No
2  3668-QPYBK    Male              0  ...          53.85       108.15    Yes
3  7795-CFOCW    Male              0  ...          42.30      1840.75     No
4  9237-HQITU  Female              0  ...          70.70       151.65    Yes

[5 rows x 21 columns]
>>> df.shape
(7043, 21)
>>> df.dtypes
customerID          object
gender              object
SeniorCitizen        int64
Partner             object
Dependents          object
tenure               int64
PhoneService        object
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
```

```
DeviceProtection     object
TechSupport          object
StreamingTV          object
StreamingMovies      object
Contract             object
PaperlessBilling     object
PaymentMethod        object
MonthlyCharges       float64
TotalCharges         object
Churn                object
dtype: object
>>> df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null object
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null object
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
TotalCharges        7043 non-null object
Churn               7043 non-null object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
>>> df[['InternetService']]
    InternetService
0               DSL
1               DSL
2               DSL
3               DSL
4       Fiber optic
...             ...
7038            DSL
7039    Fiber optic
7040            DSL
7041    Fiber optic
7042    Fiber optic
```

```
[7043 rows x 1 columns]
>>> df[['InternetService']].shape
(7043, 1)
>>> print(len(df['InternetService'].unique()))
3
>>>
```

Задание 3.
Исходные данные:
Выведите статистики по полю TotalCharges (median, mean, std).
Решение:

```
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>> df.head()
   customerID  gender  SeniorCitizen  ... MonthlyCharges Churn  TotalCharges
0  7590-VHVEG  Female              0  ...          29.85    No         29.85
1  5575-GNVDE    Male              0  ...          56.95    No       1889.50
2  3668-QPYBK    Male              0  ...          53.85   Yes        108.15
3  7795-CFOCW    Male              0  ...          42.30    No       1840.75
4  9237-HQITU  Female              0  ...          70.70   Yes        151.65

[5 rows x 21 columns]
>>> df.shape
(7043, 21)
>>> df.dtypes
customerID           object
gender               object
SeniorCitizen         int64
Partner              object
Dependents           object
tenure                int64
PhoneService         object
MultipleLines        object
InternetService      object
OnlineSecurity       object
OnlineBackup         object
DeviceProtection     object
TechSupport          object
StreamingTV          object
StreamingMovies      object
Contract             object
PaperlessBilling     object
PaymentMethod        object
MonthlyCharges      float64
Churn                object
TotalCharges        float64
dtype: object
>>> df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null object
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null object
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
Churn               7043 non-null object
TotalCharges        7032 non-null float64
dtypes: float64(2), int64(2), object(17)
memory usage: 1.1+ MB
>>> df['TotalCharges'].isna().sum()
11
>>> df['TotalCharges'].replace('isna().sum()', 'mean()')
0         29.85
1       1889.50
2        108.15
3       1840.75
4        151.65
         ...
7038    1990.50
7039    7362.90
7040     346.45
7041     306.60
7042    6844.50
Name: TotalCharges, Length: 7043, dtype: float64
>>> df['TotalCharges'].shape
(7043,)
>>> df['TotalCharges'].mean()
2283.3004408418656
>>> df['TotalCharges'].median()
1397.475
>>> df['TotalCharges'].std()
2266.771361883145
>>>
```

Задание 4.

Исходные данные:
Сделайте замену значений поля PhoneService на числовые (Yes->1, No->0)
Решение:

```
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>> df.head()
   customerID  gender  SeniorCitizen  ... MonthlyCharges TotalCharges  Churn
0  7590-VHVEG  Female              0  ...          29.85         29.85     No
1  5575-GNVDE    Male              0  ...          56.95        1889.5     No
2  3668-QPYBK    Male              0  ...          53.85        108.15    Yes
3  7795-CFOCW    Male              0  ...          42.30       1840.75     No
4  9237-HQITU  Female              0  ...          70.70        151.65    Yes

[5 rows x 21 columns]
>>> df.shape
(7043, 21)
>>> df.dtypes
customerID          object
gender              object
SeniorCitizen        int64
Partner             object
Dependents          object
tenure               int64
PhoneService        object
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         object
StreamingMovies     object
Contract            object
PaperlessBilling    object
PaymentMethod       object
MonthlyCharges     float64
TotalCharges        object
Churn               object
dtype: object
>>> df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
```

```
tenure            7043 non-null int64
PhoneService      7043 non-null object
MultipleLines     7043 non-null object
InternetService   7043 non-null object
OnlineSecurity    7043 non-null object
OnlineBackup      7043 non-null object
DeviceProtection  7043 non-null object
TechSupport       7043 non-null object
StreamingTV       7043 non-null object
StreamingMovies   7043 non-null object
Contract          7043 non-null object
PaperlessBilling  7043 non-null object
PaymentMethod     7043 non-null object
MonthlyCharges    7043 non-null float64
TotalCharges      7043 non-null object
Churn             7043 non-null object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
>>> df['PhoneService'].shape
(7043)
>>> df['PhoneService'].isna().sum()
0
>>> df['PhoneService'].dtypes
dtype('O')
>>> df = df['PhomeService'].isin([1.0,'Yes'])
>>> df['PhoneService'].dtypes
dtype('Bool')
>>> df = df['PhoneService'].replace({True: 1, False: 0})
>>> df['PhoneService'].dtypes
type('Int64')
>>> df.to_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>>
```

Задание 5.
Исходные данные:
Сделайте замену пробелов в поле TotalCharges на np.nan и приведите поле к типу данных float32. Затем заполните оставшиеся пропуски значением 0 с помощью метода fillna у столбца. Снова выведите статистики и сравните с тем, что вы видели в вопросе 3
Решение:

```
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>> df.head()
   customerID  gender  SeniorCitizen Partner  ... MonthlyCharges  Churn  TotalCharges
TotalCharges1
0  7590-VHVEG  Female              0     Yes  ...          29.85     No         29.85         29.85
1  5575-GNVDE    Male              0      No  ...          56.95     No       1889.50        1889.5
2  3668-QPYBK    Male              0      No  ...          53.85    Yes        108.15        108.15
3  7795-CFOCW    Male              0      No  ...          42.30     No       1840.75       1840.75
```

```
4  9237-HQITU  Female          0      No  ...         70.70    Yes       151.65       151.65

[5 rows x 22 columns]
>>> df.shape
(7043, 22)
>>> df.dtypes
customerID          object
gender              object
SeniorCitizen        int64
Partner             object
Dependents          object
tenure               int64
PhoneService         int64
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         object
StreamingMovies     object
Contract            object
PaperlessBilling    object
PaymentMethod       object
MonthlyCharges     float64
Churn               object
TotalCharges       float64
TotalCharges1       object
dtype: object
>>> df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 22 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null int64
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null object
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
```

```
Churn              7043 non-null object
TotalCharges       7032 non-null float64
TotalCharges1      7043 non-null object
dtypes: float64(2), int64(3), object(17)
memory usage: 1.2+ MB
>>> df['TotalCharges1'].shape
(7043,)
>>> df['TotalCharges1'].dtypes
dtype('O')
>>> df['TotalCharges1'].replace(' ', 'np.nan')
0         29.85
1        1889.5
2        108.15
3       1840.75
4        151.65
         ...
7038     1990.5
7039     7362.9
7040     346.45
7041      306.6
7042     6844.5
Name: TotalCharges1, Length: 7043, dtype: object
>>> df['TotalCharges1'].sort_values() == 'nan'
936     False
3826    False
4380    False
753     False
5218    False
        ...
6646    False
5598    False
3686    False
3353    False
2845    False
Name: TotalCharges1, Length: 7043, dtype: bool
>>> df['TotalCharges1'].fillna(0)
0         29.85
1        1889.5
2        108.15
3       1840.75
4        151.65
         ...
7038     1990.5
7039     7362.9
7040     346.45
7041      306.6
7042     6844.5
Name: TotalCharges1, Length: 7043, dtype: object
>>> df['TotalCharges1'].dtypes
dtype('O')
>>> df = pd.to_numeric(df['TotalCharges1'], errors='coerce')
>>> df['TotalCharges1'].dtypes
```

```
dtype('float64')
>>> df['TotalCharges1'].astype(np.float32)
0         29.850000
1       1889.500000
2        108.150002
3       1840.750000
4        151.649994
           ...
7027    1990.500000
7028    7362.899902
7029     346.450012
7030     306.600006
7031    6844.500000
Name: TotalCharges1, Length: 7032, dtype: float32
>>> df['TotalCharges1'].dtypes
dtype('float64')
>>> df['TotalCharges1'].mean()
2283.300440841866
>>> df['TotalCharges1'].median()
1397.475
>>> df['TotalCharges1'].std()
2266.771361883145
>>>
```

Задание 6.
Исходные данные:
Сделайте замену значений поля Churn на числовые (Yes -> 1, No — 0)
Решение:

```
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>> df.head()
   customerID  gender  SeniorCitizen Partner  ... MonthlyCharges  Churn  TotalCharges
TotalCharges1
0  7590-VHVEG  Female              0     Yes  ...          29.85     No         29.85         29.85
1  5575-GNVDE    Male              0      No  ...          56.95     No       1889.50        1889.5
2  3668-QPYBK    Male              0      No  ...          53.85    Yes        108.15        108.15
3  7795-CFOCW    Male              0      No  ...          42.30     No       1840.75       1840.75
4  9237-HQITU  Female              0      No  ...          70.70    Yes        151.65        151.65

[5 rows x 22 columns]
>>> df.shape
(7043, 22)
>>> df.dtypes
customerID          object
gender              object
SeniorCitizen        int64
Partner             object
Dependents          object
```

```
tenure              int64
PhoneService         int64
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         object
StreamingMovies     object
Contract            object
PaperlessBilling    object
PaymentMethod       object
MonthlyCharges     float64
Churn               object
TotalCharges       float64
TotalCharges1       object
dtype: object
>>> df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 22 columns):
customerID         7043 non-null object
gender             7043 non-null object
SeniorCitizen      7043 non-null int64
Partner            7043 non-null object
Dependents         7043 non-null object
tenure             7043 non-null int64
PhoneService       7043 non-null int64
MultipleLines      7043 non-null object
InternetService    7043 non-null object
OnlineSecurity     7043 non-null object
OnlineBackup       7043 non-null object
DeviceProtection   7043 non-null object
TechSupport        7043 non-null object
StreamingTV        7043 non-null object
StreamingMovies    7043 non-null object
Contract           7043 non-null object
PaperlessBilling   7043 non-null object
PaymentMethod      7043 non-null object
MonthlyCharges     7043 non-null float64
Churn              7043 non-null object
TotalCharges       7032 non-null float64
TotalCharges1      7043 non-null object
dtypes: float64(2), int64(3), object(17)
memory usage: 1.2+ MB
>>> df['Churn'].shape
(7043,)
>>> df['Churn'].isna().sum()
0
>>> df['Churn'].dtypes
dtype('O')
```

```
>>> df = df['Churn'].isin([1.0,'Yes'])
>>> df['Churn'].dtypes
dtype('bool')
>>> df = df['Churn'].replace({True: 1, False: 0})
>>> df['Churn'].dtypes
dtype('int64')
>>> df.to_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>>
```

Задание 7.
Исходные данные:
Сделайте замену значений полей StreamingMovies, StreamingTV, TechSupport на числовые (Yes -> 1, No -> 0, No internet service → 0)
Решение:

```
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>> df.head()
   customerID  gender  SeniorCitizen Partner  ... MonthlyCharges  Churn  TotalCharges
TotalCharges1
0  7590-VHVEG  Female              0     Yes  ...          29.85      0         29.85         29.85
1  5575-GNVDE    Male              0      No  ...          56.95      0       1889.50       1889.50
2  3668-QPYBK    Male              0      No  ...          53.85      1        108.15        108.15
3  7795-CFOCW    Male              0      No  ...          42.30      0       1840.75       1840.75
4  9237-HQITU  Female              0      No  ...          70.70      1        151.65        151.65

[5 rows x 22 columns]
>>> df.shape
(7043, 22)
>>> df.dtypes
customerID          object
gender              object
SeniorCitizen        int64
Partner             object
Dependents          object
tenure               int64
PhoneService         int64
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         object
StreamingMovies     object
Contract            object
PaperlessBilling    object
PaymentMethod       object
MonthlyCharges     float64
```

```
Churn                int64
TotalCharges        float64
TotalCharges1       float64
dtype: object
>>> df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 22 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null int64
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null object
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
Churn               7043 non-null int64
TotalCharges        7032 non-null float64
TotalCharges1       7032 non-null float64
dtypes: float64(3), int64(4), object(15)
memory usage: 1.2+ MB
>>> df['StreamingMovies'].shape
(7043,)
>>> df['StreamingMovies'].isna().sum()
0
>>> df['StreamingMovies'].dtypes
dtype('O')
>>> df = df['StreamingMovies'].isin([1.0,'Yes'])
>>> df['StreamingMovies'].dtypes
dtype('Bool')
>>> df = df['StreamingMovies'].replace({True: 1, False: 0})
>>> df['StreamingMovies'].dtypes
dtype('int64')
>>> df.to_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>>

Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
```

```
>>> df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>> df.head()
   customerID  gender  SeniorCitizen Partner  ... MonthlyCharges  Churn  TotalCharges
TotalCharges1
0  7590-VHVEG  Female              0     Yes  ...          29.85      0         29.85         29.85
1  5575-GNVDE    Male              0      No  ...          56.95      0       1889.50       1889.50
2  3668-QPYBK    Male              0      No  ...          53.85      1        108.15        108.15
3  7795-CFOCW    Male              0      No  ...          42.30      0       1840.75       1840.75
4  9237-HQITU  Female              0      No  ...          70.70      1        151.65        151.65

[5 rows x 22 columns]
>>> df.shape
(7043, 22)
>>> df.dtypes
customerID          object
gender              object
SeniorCitizen        int64
Partner             object
Dependents          object
tenure               int64
PhoneService         int64
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         object
StreamingMovies      int64
Contract            object
PaperlessBilling    object
PaymentMethod       object
MonthlyCharges     float64
Churn                int64
TotalCharges       float64
TotalCharges1      float64
dtype: object
>>> df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 22 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null int64
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
```

```
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null int64
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
Churn               7043 non-null int64
TotalCharges        7032 non-null float64
TotalCharges1       7032 non-null float64
dtypes: float64(3), int64(5), object(14)
memory usage: 1.2+ MB
>>> df['StreamingTV'].shape
(7043,)
>>> df['StreamingTV'].isna().sum()
0
>>> df['StreamingTV'].dtypes
dtype('O')
>>> df = df['StreamingTV'].isin([1.0,'Yes'])
>>> df['StreamingTV'].dtypes
dtype('bool')
>>> df = df['StreamingTV'].replace({True: 1, False: 0})
>>> df['StreamingTV'].dtypes
dtype('Int64')
>>> df.to_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>>

Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>> df.head()
   customerID  gender  SeniorCitizen Partner  ... MonthlyCharges  Churn  TotalCharges
TotalCharges1
0  7590-VHVEG  Female              0     Yes  ...          29.85      0         29.85         29.85
1  5575-GNVDE    Male              0      No  ...          56.95      0       1889.50       1889.50
2  3668-QPYBK    Male              0      No  ...          53.85      1        108.15        108.15
3  7795-CFOCW    Male              0      No  ...          42.30      0       1840.75       1840.75
4  9237-HQITU  Female              0      No  ...          70.70      1        151.65        151.65

[5 rows x 22 columns]
>>> df.shape
(7043, 22)
>>> df.dtypes
customerID          object
gender              object
SeniorCitizen        int64
Partner             object
Dependents          object
```

```
tenure              int64
PhoneService        int64
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         int64
StreamingMovies     int64
Contract            object
PaperlessBilling    object
PaymentMethod       object
MonthlyCharges      float64
Churn               int64
TotalCharges        float64
TotalCharges1       float64
dtype: object
>>> df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 22 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null int64
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null int64
StreamingMovies     7043 non-null int64
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
Churn               7043 non-null int64
TotalCharges        7032 non-null float64
TotalCharges1       7032 non-null float64
dtypes: float64(3), int64(6), object(13)
memory usage: 1.2+ MB
>>> df['TechSupport'].shape
(7043,)
>>> df['TechSupport'].isna().sum()
0
>>> df['TechSupport'].dtypes
dtype('O')
```

```
>>> df = df['TechSupport'].isin([1.0,'Yes'])
>>> df['TechSupport'].dtypes
dtype('bool')
>>> df = df['TechSupport'].replace({True: 1, False: 0})
>>> df['TechSupport'].dtypes
dtype('int64')
>>> df.to_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
>>>
```

Задание 8.
Исходные данные:
Для нашего датасета оставьте только указанный ниже список полей, удалив все другие и выведите верхние 3 строки
columns = ['gender', 'tenure', 'PhoneService', 'TotalCharges', 'StreamingMovies', 'StreamingTV', 'TechSupport', 'Churn']
Решение:

```
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> df = pd.read_csv('1.csv')
>>> df.head(3)
   gender  tenure  PhoneService  TechSupport  StreamingTV  StreamingMovies  Churn
TotalCharges
0  Female       1             0            0            0                0      0
29.85
1    Male      34             1            0            0                0      0
1889.50
2    Male       2             1            0            0                0      1
108.15
>>> df.shape
(7043, 8)
>>> df.dtypes
gender             object
tenure              int64
PhoneService        int64
TechSupport         int64
StreamingTV         int64
StreamingMovies     int64
Churn               int64
TotalCharges      float64
dtype: object
>>> df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 8 columns):
gender            7043 non-null object
tenure            7043 non-null int64
PhoneService      7043 non-null int64
TechSupport       7043 non-null int64
StreamingTV       7043 non-null int64
StreamingMovies   7043 non-null int64
Churn             7043 non-null int64
TotalCharges      7032 non-null float64
```

dtypes: float64(1), int64(6), object(1)
memory usage: 440.3+ KB
>>>

Задание 9.
Исходные данные:
Разделите датасет на тренировочную и тестовую выборку (подсказка - воспользуйтесь train_test_split из sklearn.model_selection. Ссылка - https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html)
from sklearn.model_selection import train_test_split
features = ['gender', 'tenure', 'PhoneService', 'TotalCharges', 'StreamingMovies', 'StreamingTV', 'TechSupport']
target = 'Churn'
Решение:
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> import scipy
>>> import sklearn
>>> from sklearn.model_selection import train_test_split
>>> df = pd.read_csv('1.csv')
>>> df.head()
  gender tenure PhoneService TechSupport StreamingTV StreamingMovies Churn TotalCharges
0 Female   1   0   0   0   0   0   29.85
1   Male   34   1   0   0   0   0   1889.50
2   Male   2   1   0   0   0   1   108.15
3   Male   45   0   1   0   0   0   1840.75
4 Female   2   1   0   0   0   1   151.65
>>> df.shape
(7043, 8)
>>> df.dtypes
gender   object
tenure   int64
PhoneService   int64
TechSupport   int64
StreamingTV   int64
StreamingMovies   int64
Churn   int64
TotalCharges   float64
dtype: object
>>> df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 8 columns):
gender   7043 non-null object
tenure   7043 non-null int64
PhoneService   7043 non-null int64
TechSupport   7043 non-null int64
StreamingTV   7043 non-null int64

```
StreamingMovies    7043 non-null int64
Churn              7043 non-null int64
TotalCharges       7032 non-null float64
dtypes: float64(1), int64(6), object(1)
memory usage: 440.3+ KB
>>> X = df.iloc[:, 1:7]
>>> y = df.iloc[:, 0]
>>> X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25)
>>> X_train
      tenure  PhoneService  TechSupport  StreamingTV  StreamingMovies  Churn
5326      15             1            0            0                0      1
3045      48             1            0            0                1      0
760        1             1            0            0                0      0
4899       4             1            0            0                0      1
1335       2             1            0            0                0      1
...      ...           ...          ...          ...              ...    ...
6948      47             1            1            1                1      0
6763      71             1            1            0                1      0
807       71             1            1            1                1      0
2770      17             1            0            1                1      1
691       31             1            1            0                1      0

[5282 rows x 6 columns]
>>> X_test
      tenure  PhoneService  TechSupport  StreamingTV  StreamingMovies  Churn
583        1             1            0            0                0      0
163       53             1            1            0                1      0
5271      13             1            0            1                1      0
360        5             1            0            0                0      0
3381      41             1            1            1                1      0
...      ...           ...          ...          ...              ...    ...
3147      15             1            0            1                1      0
4546      12             1            0            0                0      1
5913      26             1            1            0                0      0
6813      64             0            1            0                1      1
5165       1             1            0            0                0      1

[1761 rows x 6 columns]
>>> y_train
5326      Male
3045    Female
760       Male
4899    Female
1335      Male
        ...
6948    Female
6763      Male
807       Male
2770      Male
691       Male
Name: gender, Length: 5282, dtype: object
>>> y_test
```

583 Female
163 Male
5271 Male
360 Male
3381 Female
...
3147 Female
4546 Female
5913 Female
6813 Female
5165 Male
Name: gender, Length: 1761, dtype: object
>>>
Смена таргета была произведена по причине того, что терминал выбивал ошибку на тип данных, поэтому для большей устойчивости кода была произведена данная замена, на тип objects.

Задание 10.
Исходные данные:
соберите pipeline для поля gender (нужно разобраться и изучить https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html) из классов ColumnSelector и OHEEncoder, которые уже написаны ниже заранее

```
from sklearn.base import BaseEstimator, TransformerMixin
    "from sklearn.pipeline import Pipeline
    "class ColumnSelector(BaseEstimator, TransformerMixin):
    "    Transformer to select a single column from the data frame to perform additional transformations on
    "    def __init__(self, key):
    "        self.key = key
    "    def fit(self, X, y=None):
    "        return self
    "    def transform(self, X):
    "        return X[self.key]
    "class NumberSelector(BaseEstimator, TransformerMixin):
    "    Transformer to select a single column from the data frame to perform additional transformations on
    "    Use on numeric columns in the data
    "    def __init__(self, key):
    "        self.key = key
    "    def fit(self, X, y=None):
    "        return self
    "    def transform(self, X):
    "        return X[[self.key]]
    "class OHEEncoder(BaseEstimator, TransformerMixin):
    "    def __init__(self key):
    "        self.key = key
    "        self.columns = []
    "    def fit(self, X, y=None):
    "        self.columns = [col for col in pd.get_dummies(X, prefix=self.key).columns]
    "        return self
    "    def transform(self, X):
    "        X = pd.get_dummies(X, prefix=self.key)
    "        test_columns = [col for col in X.columns]
    "        for col_ in test_columns:
    "            if col_ not in self.columns:
    "                X[col_] = 0
    "        return X[self.columns]
    "gender = Pipeline([
```

```
    "                ('selector', ColumnSelector(key='gender'))
    "                ('ohe', OHEEncoder(key='gender'))])
    Решение:
```
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> import scipy
>>> import sklearn
>>> from sklearn.compose import ColumnTransformer
>>> from sklearn.feature_extraction.text import CountVectorizer
>>> from sklearn.preprocessing import OneHotEncoder
>>> from sklearn.compose import make_column_transformer
>>> df = pd.read_csv('1.csv')
>>> df.head()

	gender	tenure	PhoneService	TechSupport	StreamingTV	StreamingMovies	Churn	TotalCharges
0	Female	1	0	0	0	0	0	29.85
1	Male	34	1	0	0	0	0	1889.50
2	Male	2	1	0	0	0	1	108.15
3	Male	45	0	1	0	0	0	1840.75
4	Female	2	1	0	0	0	1	151.65

>>> df.shape
(7043, 8)
>>> df.dtypes
gender object
tenure int64
PhoneService int64
TechSupport int64
StreamingTV int64
StreamingMovies int64
Churn int64
TotalCharges float64
dtype: object
>>> df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 8 columns):
gender 7043 non-null object
tenure 7043 non-null int64
PhoneService 7043 non-null int64
TechSupport 7043 non-null int64
StreamingTV 7043 non-null int64
StreamingMovies 7043 non-null int64
Churn 7043 non-null int64
TotalCharges 7032 non-null float64
dtypes: float64(1), int64(6), object(1)
memory usage: 440.3+ KB
>>> column_gender = make_column_transformer(
... (OneHotEncoder(), ['gender']),
... (CountVectorizer(), '"Female", "Male"'),
... remainder='drop')

```
>>> column_gender
ColumnTransformer(transformers=[('onehotencoder', OneHotEncoder(), ['gender']),
                                ('countvectorizer', CountVectorizer(),
                                 '"Female", "Male"')])
>>>
```

Задание 11.
Исходные данные:
Вызовите метод fit_transform у пайплайна gender и передайте туда нашу тренировочную выборку (пример по ссылке из документации https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html#sklearn.pipeline.Pipeline.fit)
Решение:

```
Python 3.8.10 (default, Sep 28 2021, 16:10:42)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> import scipy
>>> import sklearn
>>> from sklearn.pipeline import make_pipeline
>>> from sklearn.preprocessing import StandardScaler
>>> from sklearn.datasets import make_classification
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.pipeline import Pipeline
>>> from sklearn.svm import SVC
>>> df = pd.read_csv('1.csv')
>>> df.head()
   gender  tenure  PhoneService  TechSupport  StreamingTV  StreamingMovies  Churn
TotalCharges
0  Female       1             0            0            0                0      0
29.85
1    Male      34             1            0            0                0      0
1889.50
2    Male       2             1            0            0                0      1
108.15
3    Male      45             0            1            0                0      0
1840.75
4  Female       2             1            0            0                0      1
151.65
>>> df.shape
(7043, 8)
>>> df.dtypes
gender             object
tenure              int64
PhoneService        int64
TechSupport         int64
StreamingTV         int64
StreamingMovies     int64
Churn               int64
TotalCharges      float64
dtype: object
>>> df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 8 columns):
gender            7043 non-null object
tenure            7043 non-null int64
```

PhoneService 7043 non-null int64
TechSupport 7043 non-null int64
StreamingTV 7043 non-null int64
StreamingMovies 7043 non-null int64
Churn 7043 non-null int64
TotalCharges 7032 non-null float64
dtypes: float64(1), int64(6), object(1)
memory usage: 440.3+ KB
>>> X, y = make_classification(random_state=0)
>>> X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
>>> pipe = Pipeline([('scaler', StandardScaler()), ('svc', SVC())])
>>> pipe.fit(X_train, y_train)
Pipeline(steps=[('scaler', StandardScaler()), ('svc', SVC())])
>>> pipe.score(X_test, y_test)
0.88
>>>

Задание 12.
Исходные данные:
Здесь код писать уже не нужно (все сделано за вас). К полю tenure применяем StandardScaler (нормируем и центрируем). Ссылка - https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html\n",

```
    "Вопрос - в каких случаях это может быть полезно?
from sklearn.preprocessing import StandardScaler
    "tenure =  Pipeline([
    ('selector', NumberSelector(key='tenure')),
('standard', StandardScaler()
```

Решение:
Данная операция будет полезна или необходима, когда необходимо убедиться в том что данные: во-первых не имеют пропусков и значений nan, во-вторых данные будут проверенны ещё раз на тип, чтобы они совпадали в данном столбце и были однородными, в третьих происходит центровка или стандартизация данных, которые проще обрабатывать моделе, чем уже иногда читать человеку.