Московский государственный технический
университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчет по рубежному контролю №1

Вариант 11

Выполнил:
Студент группы ИУ5-63Б
Коноваликова Светлана
Руководители: Гапанюк Ю.Е.

Дата: 18.04.22

Москва, 2022 г.

Задача №2.

Для заданного набора данных проведите обработку пропусков в данных для одного категориального и одного количественного признака. Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали? Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему?

Дополнительные требования по группам:

Для студентов групп ИУ5-63Б, ИУ5Ц-83Б - для произвольной колонки данных построить график "Ящик с усами (boxplot)".

Загрузка и анализ данных

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

```
data = pd.read_csv('/Users/Kotos/Desktop/marvel-wikia-data.csv', sep=",")
```

```
data.shape
```

```
(16376, 13)
```

```
data.head()
```

	page_id	name	urlslug	ID	ALIGN	EYE	HAIR	SEX	GSM	ALIVE	APPEARANCES	FIRST APPEARANCE	Year
0	1678	Spider-Man (Peter Parker)	\/Spider-Man_(Peter_Parker)	Secret Identity	Good Characters	Hazel Eyes	Brown Hair	Male Characters	NaN	Living Characters	4043.0	Aug-62	1962.0
1	7139	Captain America (Steven Rogers)	\/Captain_America_(Steven_Rogers)	Public Identity	Good Characters	Blue Eyes	White Hair	Male Characters	NaN	Living Characters	3360.0	Mar-41	1941.0
2	64786	Wolverine (James \"Logan\" Howlett)	\/Wolverine_(James_%22Logan%22_Howlett)	Public Identity	Neutral Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	3061.0	Oct-74	1974.0
3	1868	Iron Man (Anthony \"Tony\" Stark)	\/Iron_Man_(Anthony_%22Tony%22_Stark)	Public Identity	Good Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	2961.0	Mar-63	1963.0
4	2460	Thor (Thor Odinson)	\/Thor_(Thor_Odinson)	No Dual Identity	Good Characters	Blue Eyes	Blond Hair	Male Characters	NaN	Living Characters	2258.0	Nov-50	1950.0

```
data.isnull().sum()
```

```
page_id              0
name                 0
urlslug              0
ID                3770
ALIGN             2812
EYE               9767
HAIR              4264
SEX                854
GSM              16286
ALIVE                3
APPEARANCES       1096
FIRST APPEARANCE   815
Year               815
dtype: int64
```

```
data.dtypes
```

```
page_id              int64
name                object
urlslug             object
ID                  object
ALIGN               object
EYE                 object
HAIR                object
SEX                 object
GSM                 object
ALIVE               object
APPEARANCES        float64
FIRST APPEARANCE    object
Year               float64
dtype: object
```

```
yorn_alive = data[['ALIVE']]
yorn_alive
```

Обработка пропусков данных в категориальных значениях.

```
yorn_alive = data[['ALIVE']]
yorn_alive
```

	ALIVE
0	Living Characters
1	Living Characters
2	Living Characters
3	Living Characters
4	Living Characters
...	...
16371	Living Characters
16372	Living Characters
16373	Living Characters
16374	Living Characters
16375	Living Characters

16376 rows × 1 columns

```
yorn_alive['ALIVE'].unique()
```

```
array(['Living Characters', 'Deceased Characters', nan], dtype=object)
```

```
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
imputer1 = SimpleImputer(missing_values=np.nan, strategy='constant', fill_value='No info')
f_yoren_alive = imputer1.fit_transform(yorn_alive)
f_yoren_alive
```

```
array([['Living Characters'],
       ['Living Characters'],
       ['Living Characters'],
       ...,
       ['Living Characters'],
       ['Living Characters'],
       ['Living Characters']], dtype=object)
```

```
f_yoren_alive.shape
```

```
(16376, 1)
```

```
np.unique(f_yoren_alive)
```

```
array(['Deceased Characters', 'Living Characters', 'No info'],
      dtype=object)
```

```
data.drop(['ALIVE'], axis = 1)
data.head()
```

	page_id	name	urlslug	ID	ALIGN	EYE	HAIR	SEX	GSM	ALIVE	APPEARANCES	FIRST APPEARANCE	Year
0	1678	Spider-Man (Peter Parker)	\/Spider-Man_(Peter_Parker)	Secret Identity	Good Characters	Hazel Eyes	Brown Hair	Male Characters	NaN	Living Characters	4043.0	Aug-62	1962.0
1	7139	Captain America (Steven Rogers)	\/Captain_America_(Steven_Rogers)	Public Identity	Good Characters	Blue Eyes	White Hair	Male Characters	NaN	Living Characters	3360.0	Mar-41	1941.0
2	64786	Wolverine (James \"Logan\" Howlett)	\/Wolverine_(James_%22Logan%22_Howlett)	Public Identity	Neutral Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	3061.0	Oct-74	1974.0
3	1868	Iron Man (Anthony \"Tony\" Stark)	\/Iron_Man_(Anthony_%22Tony%22_Stark)	Public Identity	Good Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	2961.0	Mar-63	1963.0
4	2460	Thor (Thor Odinson)	\/Thor_(Thor_Odinson)	No Dual Identity	Good Characters	Blue Eyes	Blond Hair	Male Characters	NaN	Living Characters	2258.0	Nov-50	1950.0

```
data['ALIVE'] = f_yoren_alive.reshape(-1)
data.head()
```

	page_id	name	urlslug	ID	ALIGN	EYE	HAIR	SEX	GSM	ALIVE	APPEARANCES	FIRST APPEARANCE	Year
0	1678	Spider-Man (Peter Parker)	\/Spider-Man_(Peter_Parker)	Secret Identity	Good Characters	Hazel Eyes	Brown Hair	Male Characters	NaN	Living Characters	4043.0	Aug-62	1962.0
1	7139	Captain America (Steven Rogers)	\/Captain_America_(Steven_Rogers)	Public Identity	Good Characters	Blue Eyes	White Hair	Male Characters	NaN	Living Characters	3360.0	Mar-41	1941.0
2	64786	Wolverine (James \"Logan\" Howlett)	\/Wolverine_(James_%22Logan%22_Howlett)	Public Identity	Neutral Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	3061.0	Oct-74	1974.0
3	1868	Iron Man (Anthony \"Tony\" Stark)	\/Iron_Man_(Anthony_%22Tony%22_Stark)	Public Identity	Good Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	2961.0	Mar-63	1963.0
4	2460	Thor (Thor Odinson)	\/Thor_(Thor_Odinson)	No Dual Identity	Good Characters	Blue Eyes	Blond Hair	Male Characters	NaN	Living Characters	2258.0	Nov-50	1950.0

Обработка пропусков данных в количественных значениях.

```
yorn_appear = data[['APPEARANCES']]
yorn_appear
```

	APPEARANCES
0	4043.0
1	3360.0
2	3061.0
3	2961.0
4	2258.0
...	...
16371	NaN
16372	NaN
16373	NaN
16374	NaN
16375	NaN

16376 rows × 1 columns

```
yorn_appear['APPEARANCES'].unique()
```

```
array([4.043e+03, 3.360e+03, 3.061e+03, 2.961e+03, 2.258e+03, 2.255e+03,
       2.072e+03, 2.017e+03, 1.955e+03, 1.934e+03, 1.825e+03, 1.713e+03,
       1.528e+03, 1.512e+03, 1.394e+03, 1.338e+03, 1.307e+03, 1.304e+03,
       1.266e+03, 1.265e+03, 1.237e+03, 1.233e+03, 1.230e+03, 1.162e+03,
       1.161e+03, 1.137e+03, 1.120e+03, 1.107e+03, 1.050e+03, 1.047e+03,
       1.007e+03, 1.000e+03, 8.860e+02, 8.810e+02, 8.800e+02, 8.780e+02,
       8.560e+02, 8.500e+02, 7.870e+02, 7.520e+02, 7.400e+02, 7.250e+02,
       7.210e+02, 7.090e+02, 7.080e+02, 6.960e+02, 6.920e+02, 6.890e+02,
       6.860e+02, 6.570e+02, 6.410e+02, 6.360e+02, 6.180e+02, 6.120e+02,
       5.990e+02, 5.900e+02, 5.670e+02, 5.580e+02, 5.500e+02, 5.480e+02,
       5.320e+02, 5.280e+02, 5.260e+02, 5.250e+02, 5.170e+02, 5.120e+02,
       5.030e+02, 5.020e+02, 5.000e+02, 4.780e+02, 4.740e+02, 4.730e+02,
       4.710e+02, 4.640e+02, 4.480e+02, 4.460e+02, 4.330e+02, 4.290e+02,
       4.260e+02, 4.220e+02, 4.150e+02, 4.080e+02, 4.060e+02, 4.030e+02,
       3.900e+02, 3.880e+02, 3.870e+02, 3.840e+02, 3.820e+02, 3.800e+02,
       3.760e+02, 3.750e+02, 3.740e+02, 3.730e+02, 3.710e+02, 3.680e+02,
       3.660e+02, 3.610e+02, 3.530e+02, 3.480e+02, 3.460e+02, 3.420e+02,
       3.400e+02, 3.380e+02, 3.320e+02, 3.290e+02, 3.270e+02, 3.250e+02,
       3.240e+02, 3.230e+02, 3.210e+02, 3.200e+02, 3.190e+02, 3.180e+02,
       3.170e+02, 3.160e+02, 3.130e+02, 3.110e+02, 3.100e+02, 3.060e+02,
       2.960e+02, 2.930e+02, 2.900e+02, 2.840e+02, 2.800e+02, 2.770e+02,
       2.760e+02, 2.740e+02, 2.730e+02, 2.720e+02, 2.710e+02, 2.700e+02,
       2.670e+02, 2.650e+02, 2.640e+02, 2.630e+02, 2.610e+02, 2.600e+02,
       2.590e+02, 2.570e+02, 2.550e+02, 2.520e+02, 2.500e+02, 2.490e+02,
       2.480e+02, 2.470e+02, 2.450e+02, 2.420e+02, 2.400e+02, 2.390e+02,
       2.380e+02, 2.370e+02, 2.360e+02, 2.340e+02, 2.310e+02, 2.290e+02,
       2.270e+02, 2.260e+02, 2.250e+02, 2.240e+02, 2.230e+02, 2.220e+02,
       2.210e+02, 2.180e+02, 2.170e+02, 2.120e+02, 2.110e+02, 2.100e+02,
       2.090e+02, 2.070e+02, 2.050e+02, 2.040e+02, 2.030e+02, 2.020e+02,
       2.000e+02, 1.980e+02, 1.970e+02, 1.950e+02, 1.940e+02, 1.930e+02,
       1.900e+02, 1.890e+02, 1.880e+02, 1.870e+02, 1.840e+02, 1.810e+02,
       1.800e+02, 1.790e+02, 1.780e+02, 1.770e+02, 1.760e+02, 1.750e+02,
       1.740e+02, 1.720e+02, 1.710e+02, 1.700e+02, 1.690e+02, 1.660e+02,
       1.650e+02, 1.640e+02, 1.630e+02, 1.610e+02, 1.600e+02, 1.580e+02,
       1.570e+02, 1.560e+02, 1.550e+02, 1.530e+02, 1.520e+02, 1.510e+02,
       1.500e+02, 1.490e+02, 1.480e+02, 1.470e+02, 1.460e+02, 1.450e+02,
       1.440e+02, 1.430e+02, 1.420e+02, 1.410e+02, 1.400e+02, 1.390e+02,
       1.380e+02, 1.370e+02, 1.360e+02, 1.350e+02, 1.340e+02, 1.330e+02,
       1.320e+02, 1.310e+02, 1.300e+02, 1.290e+02, 1.280e+02, 1.270e+02,
       1.260e+02, 1.250e+02, 1.240e+02, 1.220e+02, 1.210e+02, 1.200e+02,
       1.190e+02, 1.180e+02, 1.160e+02, 1.150e+02, 1.140e+02, 1.130e+02,
       1.120e+02, 1.110e+02, 1.100e+02, 1.090e+02, 1.080e+02, 1.070e+02,
       1.060e+02, 1.050e+02, 1.040e+02, 1.030e+02, 1.020e+02, 1.010e+02,
       1.000e+02, 9.900e+01, 9.800e+01, 9.700e+01, 9.600e+01, 9.500e+01,
       9.400e+01, 9.300e+01, 9.200e+01, 9.100e+01, 9.000e+01, 8.900e+01,
       8.800e+01, 8.700e+01, 8.600e+01, 8.500e+01, 8.400e+01, 8.300e+01,
       8.200e+01, 8.100e+01, 8.000e+01, 7.900e+01, 7.800e+01, 7.700e+01,
       7.600e+01, 7.500e+01, 7.400e+01, 7.300e+01, 7.200e+01, 7.100e+01,
       7.000e+01, 6.900e+01, 6.800e+01, 6.700e+01, 6.600e+01, 6.500e+01,
       6.400e+01, 6.300e+01, 6.200e+01, 6.100e+01, 6.000e+01, 5.900e+01,
       5.800e+01, 5.700e+01, 5.600e+01, 5.500e+01, 5.400e+01, 5.300e+01,
       5.200e+01, 5.100e+01, 5.000e+01, 4.900e+01, 4.800e+01, 4.700e+01,
       4.600e+01, 4.500e+01, 4.400e+01, 4.300e+01, 4.200e+01, 4.100e+01,
       4.000e+01, 3.900e+01, 3.800e+01, 3.700e+01, 3.600e+01, 3.500e+01,
       3.400e+01, 3.300e+01, 3.200e+01, 3.100e+01, 3.000e+01, 2.900e+01,
       2.800e+01, 2.700e+01, 2.600e+01, 2.500e+01, 2.400e+01, 2.300e+01,
       2.200e+01, 2.100e+01, 2.000e+01, 1.900e+01, 1.800e+01, 1.700e+01,
       1.600e+01, 1.500e+01, 1.400e+01, 1.300e+01, 1.200e+01, 1.100e+01,
       1.000e+01, 9.000e+00, 8.000e+00, 7.000e+00, 6.000e+00, 5.000e+00,
       4.000e+00, 3.000e+00, 2.000e+00, 1.000e+00,       nan])
```

```
imputer2 = SimpleImputer(missing_values=np.nan, strategy='median')
f_yoren_appear = imputer2.fit_transform(yorn_appear)
f_yoren_appear
```

```
array([[4.043e+03],
       [3.360e+03],
       [3.061e+03],
       ...,
       [3.000e+00],
       [3.000e+00],
       [3.000e+00]])
```

```
np.unique(f_yoren_appear)
```

```
array([1.000e+00, 2.000e+00, 3.000e+00, 4.000e+00, 5.000e+00, 6.000e+00,
       7.000e+00, 8.000e+00, 9.000e+00, 1.000e+01, 1.100e+01, 1.200e+01,
       1.300e+01, 1.400e+01, 1.500e+01, 1.600e+01, 1.700e+01, 1.800e+01,
       1.900e+01, 2.000e+01, 2.100e+01, 2.200e+01, 2.300e+01, 2.400e+01,
       2.500e+01, 2.600e+01, 2.700e+01, 2.800e+01, 2.900e+01, 3.000e+01,
       3.100e+01, 3.200e+01, 3.300e+01, 3.400e+01, 3.500e+01, 3.600e+01,
       3.700e+01, 3.800e+01, 3.900e+01, 4.000e+01, 4.100e+01, 4.200e+01,
       4.300e+01, 4.400e+01, 4.500e+01, 4.600e+01, 4.700e+01, 4.800e+01,
       4.900e+01, 5.000e+01, 5.100e+01, 5.200e+01, 5.300e+01, 5.400e+01,
       5.500e+01, 5.600e+01, 5.700e+01, 5.800e+01, 5.900e+01, 6.000e+01,
       6.100e+01, 6.200e+01, 6.300e+01, 6.400e+01, 6.500e+01, 6.600e+01,
       6.700e+01, 6.800e+01, 6.900e+01, 7.000e+01, 7.100e+01, 7.200e+01,
       7.300e+01, 7.400e+01, 7.500e+01, 7.600e+01, 7.700e+01, 7.800e+01,
       7.900e+01, 8.000e+01, 8.100e+01, 8.200e+01, 8.300e+01, 8.400e+01,
       8.500e+01, 8.600e+01, 8.700e+01, 8.800e+01, 8.900e+01, 9.000e+01,
       9.100e+01, 9.200e+01, 9.300e+01, 9.400e+01, 9.500e+01, 9.600e+01,
       9.700e+01, 9.800e+01, 9.900e+01, 1.000e+02, 1.010e+02, 1.020e+02,
       1.030e+02, 1.040e+02, 1.050e+02, 1.060e+02, 1.070e+02, 1.080e+02,
       1.090e+02, 1.100e+02, 1.110e+02, 1.120e+02, 1.130e+02, 1.140e+02,
       1.150e+02, 1.160e+02, 1.180e+02, 1.190e+02, 1.200e+02, 1.210e+02,
       1.220e+02, 1.240e+02, 1.250e+02, 1.260e+02, 1.270e+02, 1.280e+02,
       1.290e+02, 1.300e+02, 1.310e+02, 1.320e+02, 1.330e+02, 1.340e+02,
       1.350e+02, 1.360e+02, 1.370e+02, 1.380e+02, 1.390e+02, 1.400e+02,
       1.410e+02, 1.420e+02, 1.430e+02, 1.440e+02, 1.450e+02, 1.460e+02,
       1.470e+02, 1.480e+02, 1.490e+02, 1.500e+02, 1.510e+02, 1.520e+02,
       1.530e+02, 1.550e+02, 1.560e+02, 1.570e+02, 1.580e+02, 1.600e+02,
       1.610e+02, 1.630e+02, 1.640e+02, 1.650e+02, 1.660e+02, 1.690e+02,
       1.700e+02, 1.710e+02, 1.720e+02, 1.740e+02, 1.750e+02, 1.760e+02,
       1.770e+02, 1.780e+02, 1.790e+02, 1.800e+02, 1.810e+02, 1.840e+02,
       1.870e+02, 1.880e+02, 1.890e+02, 1.900e+02, 1.930e+02, 1.940e+02,
       1.950e+02, 1.970e+02, 1.980e+02, 2.000e+02, 2.020e+02, 2.030e+02,
       2.040e+02, 2.050e+02, 2.070e+02, 2.090e+02, 2.100e+02, 2.110e+02,
       2.120e+02, 2.170e+02, 2.180e+02, 2.210e+02, 2.220e+02, 2.230e+02,
       2.240e+02, 2.250e+02, 2.260e+02, 2.270e+02, 2.290e+02, 2.310e+02,
       2.340e+02, 2.360e+02, 2.370e+02, 2.380e+02, 2.390e+02, 2.400e+02,
       2.420e+02, 2.450e+02, 2.470e+02, 2.480e+02, 2.490e+02, 2.500e+02,
       2.520e+02, 2.550e+02, 2.570e+02, 2.590e+02, 2.600e+02, 2.610e+02,
       2.630e+02, 2.640e+02, 2.650e+02, 2.670e+02, 2.700e+02, 2.710e+02,
       2.720e+02, 2.730e+02, 2.740e+02, 2.760e+02, 2.770e+02, 2.800e+02,
       2.840e+02, 2.900e+02, 2.930e+02, 2.960e+02, 3.060e+02, 3.100e+02,
       3.110e+02, 3.130e+02, 3.160e+02, 3.170e+02, 3.180e+02, 3.190e+02,
       3.200e+02, 3.210e+02, 3.230e+02, 3.240e+02, 3.250e+02, 3.270e+02,
       3.290e+02, 3.320e+02, 3.380e+02, 3.400e+02, 3.420e+02, 3.460e+02,
       3.480e+02, 3.530e+02, 3.610e+02, 3.660e+02, 3.680e+02, 3.710e+02,
       3.730e+02, 3.740e+02, 3.750e+02, 3.760e+02, 3.800e+02, 3.820e+02,
       3.840e+02, 3.870e+02, 3.880e+02, 3.900e+02, 4.030e+02, 4.060e+02,
       4.080e+02, 4.150e+02, 4.220e+02, 4.260e+02, 4.290e+02, 4.330e+02,
       4.460e+02, 4.480e+02, 4.640e+02, 4.710e+02, 4.730e+02, 4.740e+02,
       4.780e+02, 5.000e+02, 5.020e+02, 5.030e+02, 5.120e+02, 5.170e+02,
       5.250e+02, 5.260e+02, 5.280e+02, 5.320e+02, 5.480e+02, 5.500e+02,
       5.580e+02, 5.670e+02, 5.900e+02, 5.990e+02, 6.120e+02, 6.180e+02,
       6.360e+02, 6.410e+02, 6.570e+02, 6.860e+02, 6.890e+02, 6.920e+02,
       6.960e+02, 7.080e+02, 7.090e+02, 7.210e+02, 7.250e+02, 7.400e+02,
       7.520e+02, 7.870e+02, 8.500e+02, 8.560e+02, 8.780e+02, 8.800e+02,
       8.810e+02, 8.860e+02, 1.000e+03, 1.007e+03, 1.047e+03, 1.050e+03,
       1.107e+03, 1.120e+03, 1.137e+03, 1.161e+03, 1.162e+03, 1.230e+03,
       1.233e+03, 1.237e+03, 1.265e+03, 1.266e+03, 1.304e+03, 1.307e+03,
       1.338e+03, 1.394e+03, 1.512e+03, 1.528e+03, 1.713e+03, 1.825e+03,
       1.934e+03, 1.955e+03, 2.017e+03, 2.072e+03, 2.255e+03, 2.258e+03,
       2.961e+03, 3.061e+03, 3.360e+03, 4.043e+03])
```

```
f_yoren_appear.shape
```

```
(16376, 1)
```

```
data.drop(['APPEARANCES'], axis = 1)
data.head()
```

	page_id	name	urlslug	ID	ALIGN	EYE	HAIR	SEX	GSM	ALIVE	APPEARANCES	FIRST APPEARANCE	Year
0	1678	Spider-Man (Peter Parker)	\/Spider-Man_(Peter_Parker)	Secret Identity	Good Characters	Hazel Eyes	Brown Hair	Male Characters	NaN	Living Characters	4043.0	Aug-62	1962.0
1	7139	Captain America (Steven Rogers)	\/Captain_America_(Steven_Rogers)	Public Identity	Good Characters	Blue Eyes	White Hair	Male Characters	NaN	Living Characters	3360.0	Mar-41	1941.0
2	64786	Wolverine (James \"Logan\" Howlett)	\/Wolverine_(James_%22Logan%22_Howlett)	Public Identity	Neutral Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	3061.0	Oct-74	1974.0
3	1868	Iron Man (Anthony \"Tony\" Stark)	\/Iron_Man_(Anthony_%22Tony%22_Stark)	Public Identity	Good Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	2961.0	Mar-63	1963.0
4	2460	Thor (Thor Odinson)	\/Thor_(Thor_Odinson)	No Dual Identity	Good Characters	Blue Eyes	Blond Hair	Male Characters	NaN	Living Characters	2258.0	Nov-50	1950.0

```
data['APPEARANCES'] = f_yoren_appear.reshape(-1)
data.head()
```

	page_id	name	urlslug	ID	ALIGN	EYE	HAIR	SEX	GSM	ALIVE	APPEARANCES	FIRST APPEARANCE	Year
0	1678	Spider-Man (Peter Parker)	\/Spider-Man_(Peter_Parker)	Secret Identity	Good Characters	Hazel Eyes	Brown Hair	Male Characters	NaN	Living Characters	4043.0	Aug-62	1962.0
1	7139	Captain America (Steven Rogers)	\/Captain_America_(Steven_Rogers)	Public Identity	Good Characters	Blue Eyes	White Hair	Male Characters	NaN	Living Characters	3360.0	Mar-41	1941.0
2	64786	Wolverine (James \"Logan\" Howlett)	\/Wolverine_(James_%22Logan%22_Howlett)	Public Identity	Neutral Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	3061.0	Oct-74	1974.0
3	1868	Iron Man (Anthony \"Tony\" Stark)	\/Iron_Man_(Anthony_%22Tony%22_Stark)	Public Identity	Good Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	2961.0	Mar-63	1963.0
4	2460	Thor (Thor Odinson)	\/Thor_(Thor_Odinson)	No Dual Identity	Good Characters	Blue Eyes	Blond Hair	Male Characters	NaN	Living Characters	2258.0	Nov-50	1950.0

Дополнительное задание.

```
sns.boxplot(x=data['Year'])
```

```
<AxesSubplot:xlabel='Year'>
```