



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Отчёт по рубежному контролю №1

Вариант 11

Выполнила:
Студентка группы ИУ5-65Б
Крайчиков О.Д.

Задание

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?

Набор данных: <https://www.kaggle.com/fivethirtyeight/fivethirtyeight-comic-characters-dataset> (файл marvel-wikia-data.csv).

Дополнительное требование: для набора данных построить "парные диаграммы".

Текст программы

Рубежный контроль №1

Крайчиков О.Д.

ИУ5-61Б

Импорт библиотек

```
In [20]: import pandas as pd
import numpy as np
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
import seaborn as sns
import matplotlib.pyplot as plt
sns.set(style="ticks")
```

```
In [21]: data = pd.read_csv('marvel-wikia-data.csv', sep=',')
```

Характеристики датасета

```
In [7]: data.head()
```

Out[7]:

	page_id	name	urlslug	ID	ALIGN	EYE	HAIR	SEX	GSM	ALIVE	APPEARANCES	FIRST APPEARANCE
0	1878	Spider-Man (Peter Parker)	VSpider-Man_(Peter_Parker)	Secret Identity	Good Characters	Hazel Eyes	Brown Hair	Male Characters	NaN	Living Characters	4043.0	Aug-6
1	7139	Captain America (Steven Rogers)	VCaptain_America_(Steven_Rogers)	Public Identity	Good Characters	Blue Eyes	White Hair	Male Characters	NaN	Living Characters	3380.0	Mar-4
2	84788	Wolverine (James "Logan" Howlett)	VWolverine_(James_%22Logan%22_Howlett)	Public Identity	Neutral Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	3081.0	Oct-7
3	1888	Iron Man (Anthony "Tony" Stark)	VIron_Man_(Anthony_%22Tony%22_Stark)	Public Identity	Good Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	2981.0	Mar-6
4	2480	Thor (Thor Odinson)	VThor_(Thor_Odinson)	No Dual Identity	Good Characters	Blue Eyes	Blond Hair	Male Characters	NaN	Living Characters	2258.0	Nov-6

```
In [8]: # Колонки с пропусками
cols_with_na = [c for c in data.columns if data[c].isnull().sum() > 0]
cols_with_na
```

Out[8]:

```
['ID',
 'ALIGN',
 'EYE',
 'HAIR',
 'SEX',
 'GSM',
 'ALIVE',
 'APPEARANCES',
 'FIRST APPEARANCE',
 'Year']
```

```
In [9]: # Доля (процент) пропусков
[[c, data[c].isnull().mean()] for c in cols_with_na]
```

```
Out[9]: [('ID', 0.23021494870542256),
 ('ALIGN', 0.17171470444553005),
 ('EYE', 0.5964215925744992),
 ('HAIR', 0.26038104543234003),
 ('SEX', 0.052149487054225695),
 ('GSM', 0.9945041524181729),
 ('ALIVE', 0.00018319491939423546),
 ('APPEARANCES', 0.06692721055202736),
 ('FIRST APPEARANCE', 0.04976795310210064),
 ('Year', 0.04976795310210064)]
```

Обработка пропусков для категориального признака

Можно заметить, что для признака "GSM" пропущенных данных слишком много (около 99%), следовательно нужно удалить признак (колонку) целиком.

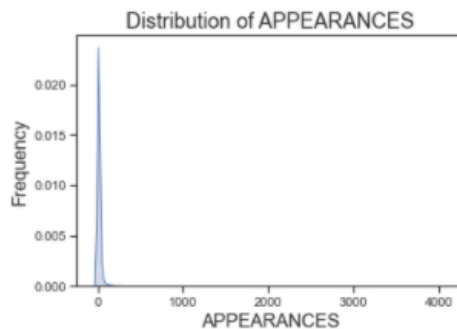
```
In [10]: data.drop(['GSM'], axis=1, inplace=True)
```

Обработка пропусков для количественного признака

Поскольку в исследуемом датасете один количественный признак "Appearances" и процент пропусков для него составляет <5%, то будем использовать метод заполнения пропущенных значений показателями центра распределения.

```
In [22]: g = sns.kdeplot(data=data, x="APPEARANCES", shade=True)
g.set_xlabel("APPEARANCES", size = 16)
g.set_ylabel("Frequency", size = 16)
plt.title('Distribution of APPEARANCES', size = 18)
```

```
Out[22]: Text(0.5, 1.0, 'Distribution of APPEARANCES')
```



```
In [23]: data[['APPEARANCES']].describe()
```

```
Out[23]:
```

APPEARANCES	
count	15280.000000
mean	17.033377
std	98.372959
min	1.000000
25%	1.000000
50%	3.000000
75%	8.000000

min	1.000000
25%	1.000000
50%	3.000000
75%	8.000000
max	4043.000000

Получаем одномодальное распределение, поэтому будем использовать моду для заполнения пустых значений.

```
In [24]: indicator = MissingIndicator()
mask_missing_values_only = indicator.fit_transform(data[['APPEARANCES']])
imp_num = SimpleImputer(strategy='most_frequent')
data_num_imp = imp_num.fit_transform(data[['APPEARANCES']])
data[['APPEARANCES']] = data_num_imp
```

Характеристики датасета после обработки пропусков

```
In [25]: # Колонки с пропусками
cols_with_na = [c for c in data.columns if data[c].isnull().sum() > 0]
cols_with_na
```

```
Out[25]: ['ID',
'ALIGN',
'EYE',
'HAIR',
'SEX',
'GSM',
'ALIVE',
'FIRST APPEARANCE',
'Year']
```

```
In [26]: # Доля (процент) пропусков
[(c, data[c].isnull().mean()) for c in cols_with_na]
```

```
Out[26]: [('ID', 0.23021494870542256),
('ALIGN', 0.17171470444553005),
('EYE', 0.5964215925744992),
('HAIR', 0.26038104543234003),
('SEX', 0.052149487054225695),
('GSM', 0.9945041524181729),
('ALIVE', 0.00018319491939423546),
('FIRST APPEARANCE', 0.04976795310210064),
('Year', 0.04976795310210064)]
```

```
In [27]: data.head()
```

Out[27]:

	page_id	name	urlslug	ID	ALIGN	EYE	HAIR	SEX	GSM	ALIVE	APPEARANCES	FIRS APPEARANC
	0	Spider-Man (Peter Parker)	VSpider-Man_(Peter_Parker)	Secret Identity	Good Characters	Hazel Eyes	Brown Hair	Male Characters	NaN	Living Characters	4043.0	Aug-6
	1	Captain America (Steven Rogers)	VCaptain_America_(Steven_Rogers)	Public Identity	Good Characters	Blue Eyes	White Hair	Male Characters	NaN	Living Characters	3380.0	Mar-4
	2	Wolverine (James "Logan" Howlett)	VWolverine_(James_%22Logan%22_Howlett)	Public Identity	Neutral Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	3061.0	Oct-7
	3	Iron Man (Anthony "Tony" Stark)	VIron_Man_(Anthony_%22Tony%22_Stark)	Public Identity	Good Characters	Blue Eyes	Black Hair	Male Characters	NaN	Living Characters	2961.0	Mar-6

Дополнительное требование

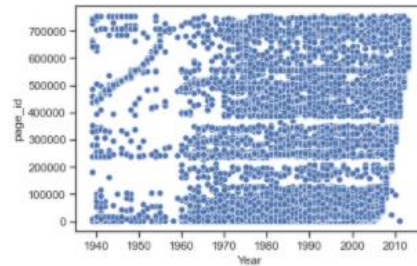
```
In [37]: plt.figure(figsize=(12,6))
```

```
Out[37]: <Figure size 864x432 with 0 Axes>
```

```
<Figure size 864x432 with 0 Axes>
```

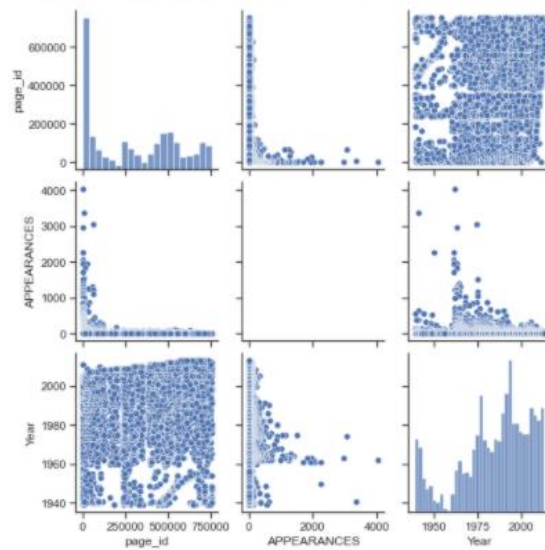
```
In [40]: sns.scatterplot(x='Year', y='page_id', data=data)
```

```
Out[40]: <AxesSubplot:xlabel='Year', ylabel='page_id'>
```



```
In [41]: sns.pairplot(data)
```

```
Out[41]: <seaborn.axisgrid.PairGrid at 0x1e9911f4b20>
```



```
In [ ]:
```