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
Е. И. Мащенко

2021

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

Мащенко Е.И.

ИУ5-65Б

Импорт библиотек

В [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")
```

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

Характеристики датасета ¶

В [22]:
```
data.head()
```

Out[22]:

name	urlslug	ID	ALIGN	EYE	HAIR	SEX	GSM	ALIVE	APPEARANCES	FIRST APPEARANCE	Year
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
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
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
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
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

В [23]:
```
# Колонки с пропусками
cols_with_na = [c for c in data.columns if data[c].isnull().sum() > 0]
cols_with_na
```

Out[23]:
```
['ID',
 'ALIGN',
 'EYE',
 'HAIR',
 'SEX',
 'GSM',
 'ALIVE',
 'APPEARANCES',
 'FIRST APPEARANCE',
 'Year']
```

In [24]:
```
# Доля (процент) пропусков
[(c, data[c].isnull().mean()) for c in cols_with_na]
```

Out[24]:
```
[('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.

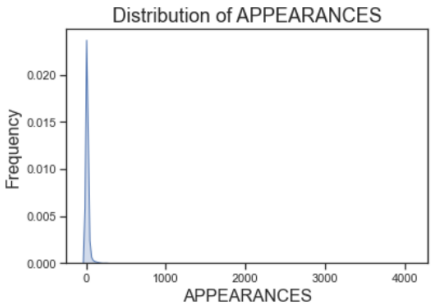
In [25]:
```
data.drop(['GSM'], axis=1, inplace=True)
```

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 [26]:
```
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[26]:
```
Text(0.5, 1.0, 'Distribution of APPEARANCES')
```



In [27]:
```
data[['APPEARANCES']].describe()
```

Out[27]:

	APPEARANCES
count	15280.000000
mean	17.033377
std	96.372959
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 [28]:
```
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 [29]:
```
# Колонки с пропусками
cols_with_na = [c for c in data.columns if data[c].isnull().sum() > 0]
cols_with_na
```

Out[29]:
```
['ID', 'ALIGN', 'EYE', 'HAIR', 'SEX', 'ALIVE', 'FIRST APPEARANCE', 'Year']
```

In [30]:
```
# Доля (процент) пропусков
[(c, data[c].isnull().mean()) for c in cols_with_na]
```

Out[30]:
```
[('ID', 0.23021494870542256),
 ('ALIGN', 0.17171470444553005),
 ('EYE', 0.5964215925744992),
 ('HAIR', 0.26038104543234003),
 ('SEX', 0.052149487054225695),
 ('ALIVE', 0.00018319491939423546),
 ('FIRST APPEARANCE', 0.04976795310210064),
 ('Year', 0.04976795310210064)]
```

B [31]:
```
data.head()
```

Out[31]:

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

Дополнительное требование

B [34]:
```
data.columns
```

Out[34]:
```
Index(['page_id', 'name', 'urlslug', 'ID', 'ALIGN', 'EYE', 'HAIR', 'SEX',
       'ALIVE', 'APPEARANCES', 'FIRST APPEARANCE', 'Year'],
      dtype='object')
```

B [35]:
```
sns.pairplot(data)
```

Out[35]:
```
<seaborn.axisgrid.PairGrid at 0x16513af6d30>
```