РК1 Гаврилов Леонид, ИУ5-63б

Вариант: 4; Задача: 1; Датасет: 4

Для заданного набора данных проведите корреляционный анализ. В случае наличия пропусков в данных удалите строки или колонки, содержащие пропуски. Сделайте выводы о возможности построения моделей машинного обучения и о возможном вкладе признаков в модель.

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

Сперва импортируем нужные библиотеки:

In [1]:
```
from sklearn.datasets import load_boston
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

Загружаем данные с датасета:

In [2]:
```
data = pd.read_csv('toy_dataset.csv')
```

Информация о датасете и поиск уникального поля:

In [3]:
```
# первые 5 столбцов таблицы
data.head()
```

Out[3]:

	Number	City	Gender	Age	Income	Illness
0	1	Dallas	Male	41	40367.0	No
1	2	Dallas	Male	54	45084.0	No
2	3	Dallas	Male	42	52483.0	No
3	4	Dallas	Male	40	40941.0	No
4	5	Dallas	Male	46	50289.0	No

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150000 entries, 0 to 149999
Data columns (total 6 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   Number   150000 non-null  int64  
 1   City     150000 non-null  object 
 2   Gender   150000 non-null  object 
 3   Age      150000 non-null  int64  
 4   Income   150000 non-null  float64
 5   Illness  150000 non-null  object 
dtypes: float64(1), int64(2), object(3)
memory usage: 6.9+ MB
```

In [5]:
```
data.describe()
```

Out[5]:

	Number	Age	Income
count	150000.000000	150000.000000	150000.000000
mean	75000.500000	44.950200	91252.798273
std	43301.414527	11.572486	24989.500948
min	1.000000	25.000000	-654.000000
25%	37500.750000	35.000000	80867.750000
50%	75000.500000	45.000000	93655.000000
75%	112500.250000	55.000000	104519.000000
max	150000.000000	65.000000	177157.000000

In [6]:
```
data.dtypes
```

Out[6]:
```
Number        int64
City         object
Gender       object
Age           int64
Income      float64
Illness      object
dtype: object
```

In [7]:
```
data.isnull().sum()
```

Out[7]:
```
Number     0
City       0
Gender     0
Age        0
Income     0
Illness    0
dtype: int64
```

In [18]:
```
data['Illness'].unique()
```

Out[18]:
```
array(['No', 'Yes'], dtype=object)
```

Видим, что Illness содержит только "Yes"/"No"

Переходим к корреляционному анализу

In [9]:
```
corr_matrix = data.corr()
```

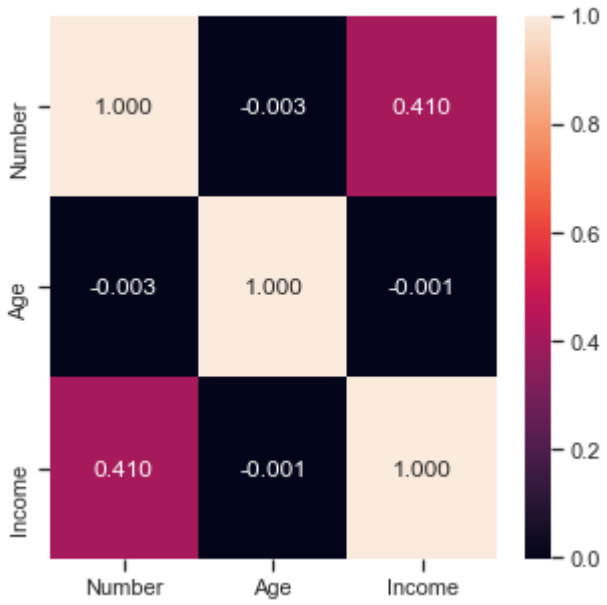
In [11]:
```
data.corr()
```

Out[11]:

	Number	Age	Income
Number	1.000000	-0.003448	0.410460
Age	-0.003448	1.000000	-0.001318
Income	0.410460	-0.001318	1.000000

In [15]:
```
plt.figure(figsize=(5,5))
sns.heatmap(corr_matrix, annot=True, fmt='.3f')
```

Out[15]: <AxesSubplot:>



Применим методы корреляции Пирсона, Кендалла и Спирмена

In [19]:
```
data.corr(method='pearson')
```

Out[19]:

	Number	Age	Income
Number	1.000000	-0.003448	0.410460
Age	-0.003448	1.000000	-0.001318
Income	0.410460	-0.001318	1.000000

In [20]:
```
data.corr(method='kendall')
```

Out[20]:

	Number	Age	Income
Number	1.000000	-0.002319	0.194147
Age	-0.002319	1.000000	-0.000978
Income	0.194147	-0.000978	1.000000

In [21]:
```
data.corr(method='spearman')
```

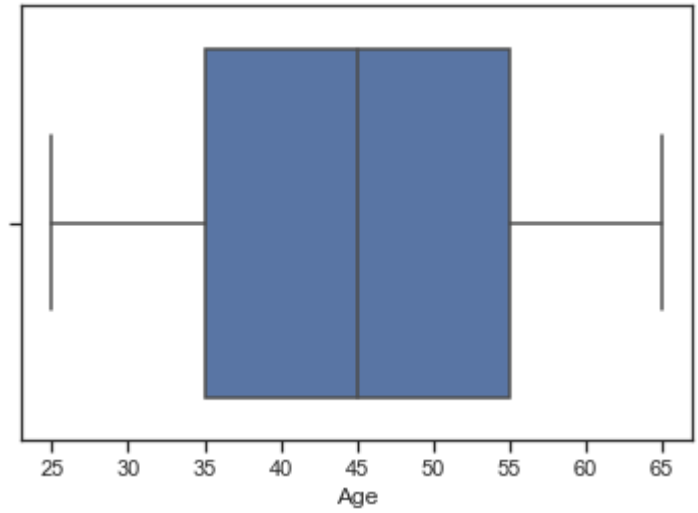
Out[21]:

	Number	Age	Income
Number	1.000000	-0.003441	0.286131
Age	-0.003441	1.000000	-0.001452
Income	0.286131	-0.001452	1.000000

Ящик с усами

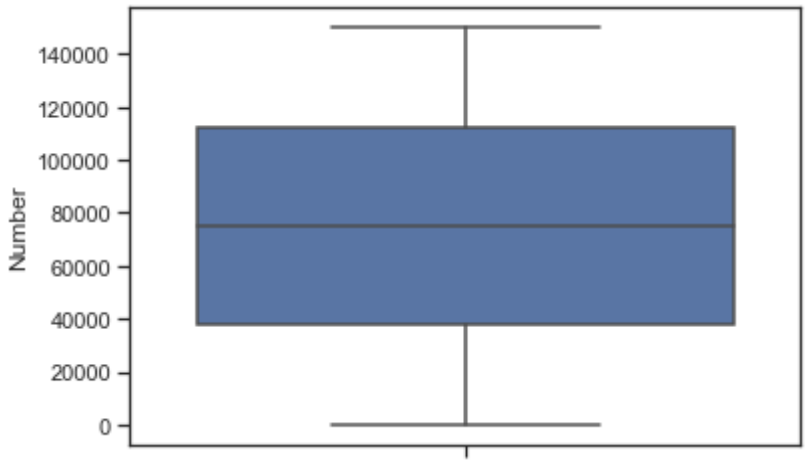
In [22]:
```
sns.boxplot(x=data['Age'])
```

Out[22]:
```
<AxesSubplot:xlabel='Age'>
```



In [25]:
```
sns.boxplot(y=data['Number'])
```

Out[25]:
```
<AxesSubplot:ylabel='Number'>
```



Выводы

Делаем вывод на основе слабой корреляции между полями (признаками) датасетаа, что модель машинного обучения построить будет проблематично.