

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет «Информатика и система управления»
Кафедра ИУ-5 «Системы обработки информации и управления»

ОТЧЕТ

**Рубежный контроль №1 по курсу
«Методы машинного обучения»**

Исполнитель - студент группы ИУ5-21М:

Кауров Максим _____

Москва – 2020 год

Кауров Максим ИУ5-21М

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

```
↳ /usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarnin
import pandas.util.testing as tm
```

```
#Загружаем датасет
from google.colab import files
files.upload()
```

```
↳  toy_dataset.csv
• toy_dataset.csv(application/vnd.ms-excel) - 5735570 bytes, last modified: 15.10.2019 - 100% done
Saving toy_dataset.csv to toy_dataset.csv
{'toy_dataset.csv': b'Number,City,Gender,Age,Income,Illness\n1,Dallas,Male,41,40367.0
```

```
#Выбираем датасет для работы
data = pd.read_csv('toy_dataset.csv', sep=",")
```

```
print(data.head())
print("\nРазмер датасета: " + str(data.shape))
```

```
↳
```

	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

```
Размер датасета: (150000, 6)
```

```
print("Список колонок с типами данных")
print(data.dtypes)
```

```
↳ Список колонок с типами данных
Number      int64
City        object
Gender       object
Age          int64
Income      float64
Illness      object
dtype: object
```

```
# Проверим наличие пустых значений
# Цикл по колонкам датасета
for col in data.columns:
    # Количество пустых значений - все значения заполнены
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
↳ Number - 0
   City - 0
   Gender - 0
   Age - 0
   Income - 0
   Illness - 0
```

```
# Выведем уникальные значения для всех столбцов
print("Number: " + str(data['Number'].unique()) + "\n")
print("City: " + str(data['City'].unique()) + "\n")
print("Gender: " + str(data['Gender'].unique()) + "\n")
print("Age: " + str(data['Age'].unique()) + "\n")
print("Income: " + str(data['Income'].unique()) + "\n")
print("Illness: " + str(data['Illness'].unique()))
```

```
↳ Number: [      1      2      3 ... 149998 149999 150000]

   City: ['Dallas' 'New York City' 'Los Angeles' 'Mountain View' 'Boston'
         'Washington D.C.' 'San Diego' 'Austin']

   Gender: ['Male' 'Female']

   Age: [41 54 42 40 46 36 32 39 51 30 48 47 61 43 27 38 35 57 33 58 64 44 34 45
        55 63 59 26 56 62 31 49 53 29 28 25 37 65 60 50 52]

   Income: [ 40367.  45084.  52483. ... 107123.  62501.  77823.]

   Illness: ['No' 'Yes']
```

```
# Основные статистические характеристики набора данных
data.describe()
```

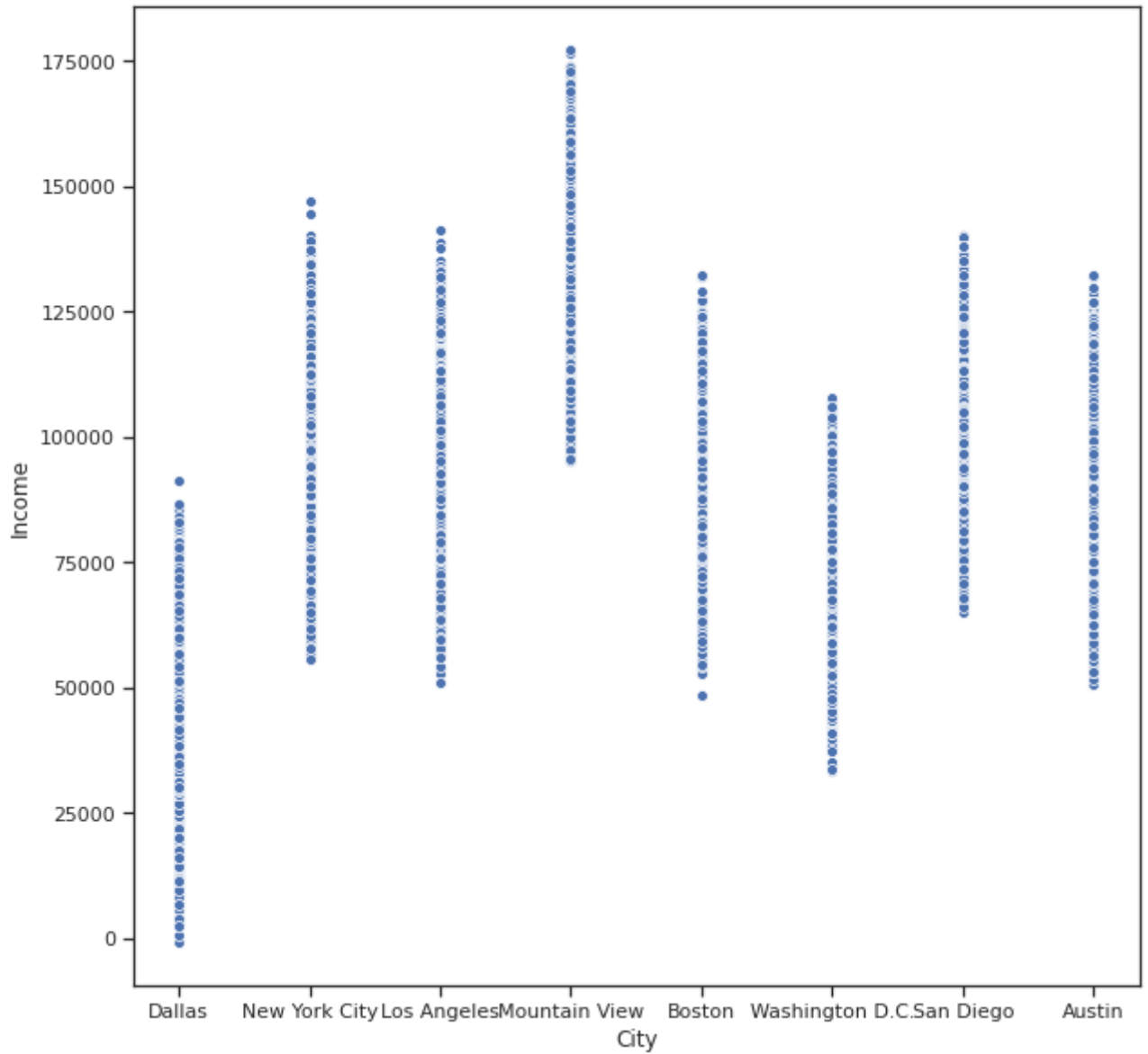
Строим диаграмму корреляции между городом и доходом

Как видно из диаграммы разные города имеют разные доходы, можно судить что "Dallas" - самым богатым

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='City', y='Income', data=data)
```

```
↳
```

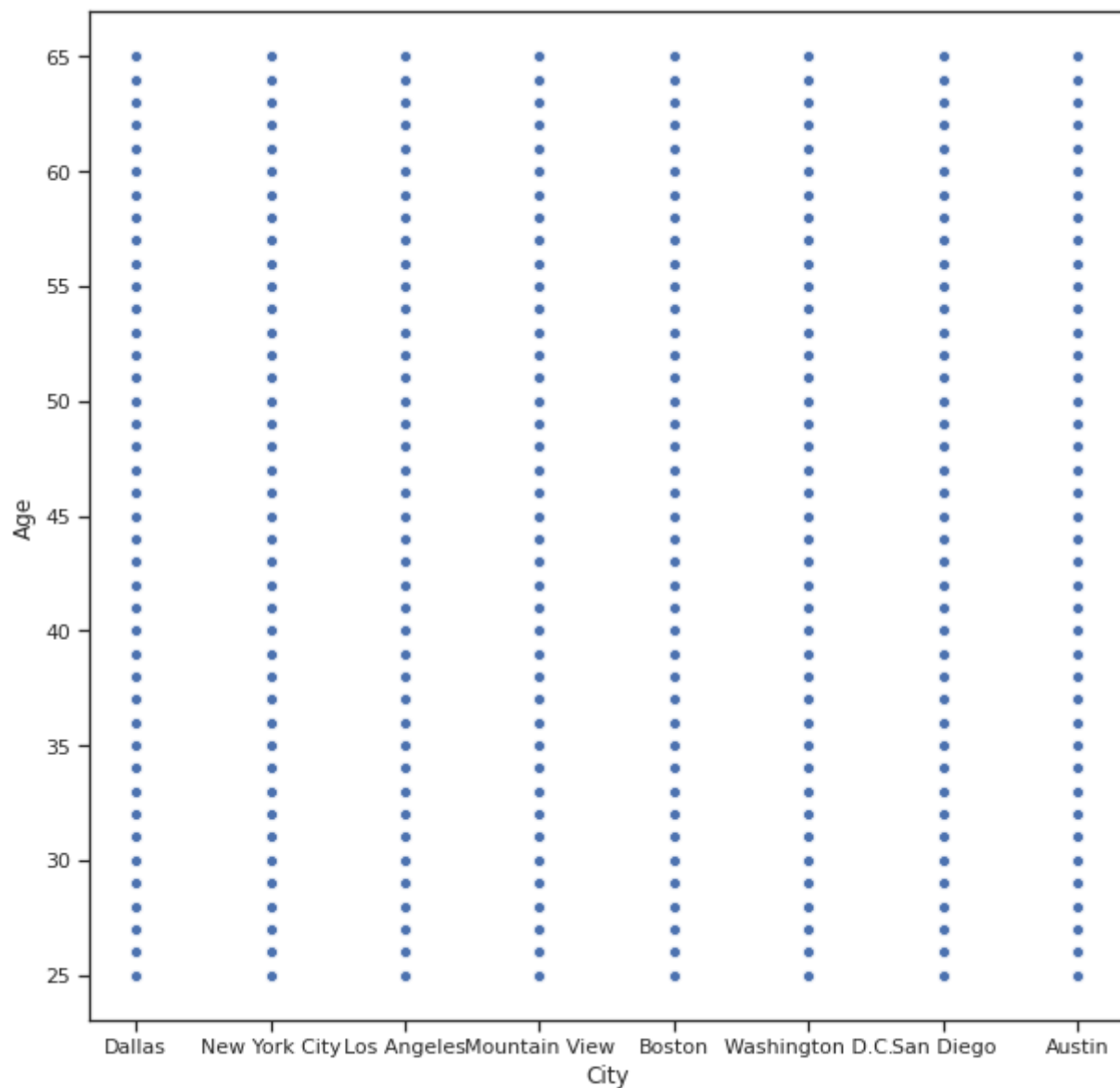
<matplotlib.axes._subplots.AxesSubplot at 0x7f16219ac7b8>



```
# Строим диаграмму корреляции между городом и доходом
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='City', y='Age', data=data)
```



<matplotlib.axes._subplots.AxesSubplot at 0x7f1621654f60>

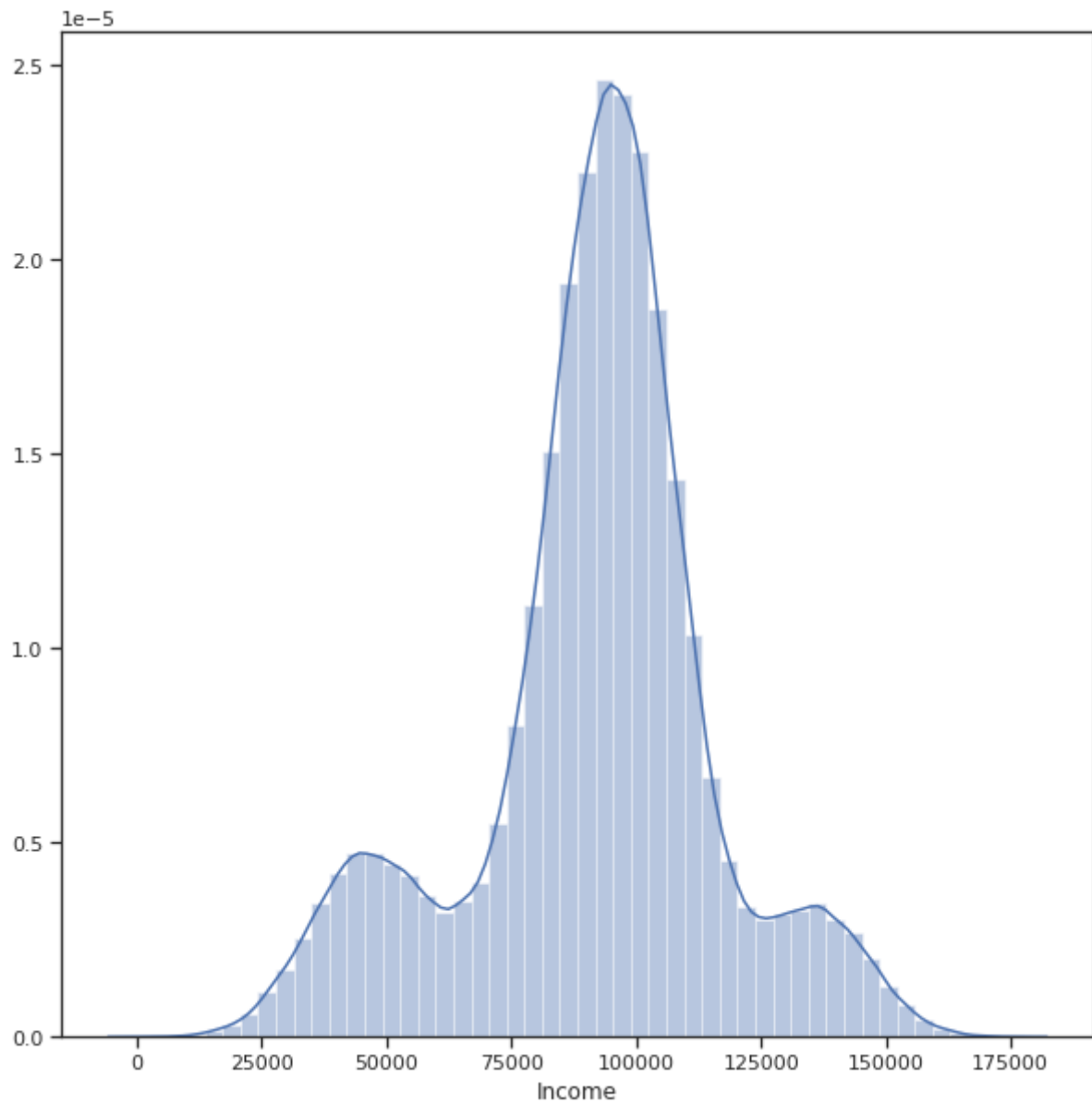


Как видно из диаграммы выше люди всех возрастов живут во всех городах

```
#Строим гистограмму дохода
fig, ax = plt.subplots(figsize=(10,10))
print(sns.distplot(data['Income']))
```



```
AxesSubplot(0.125,0.125;0.775x0.755)
```

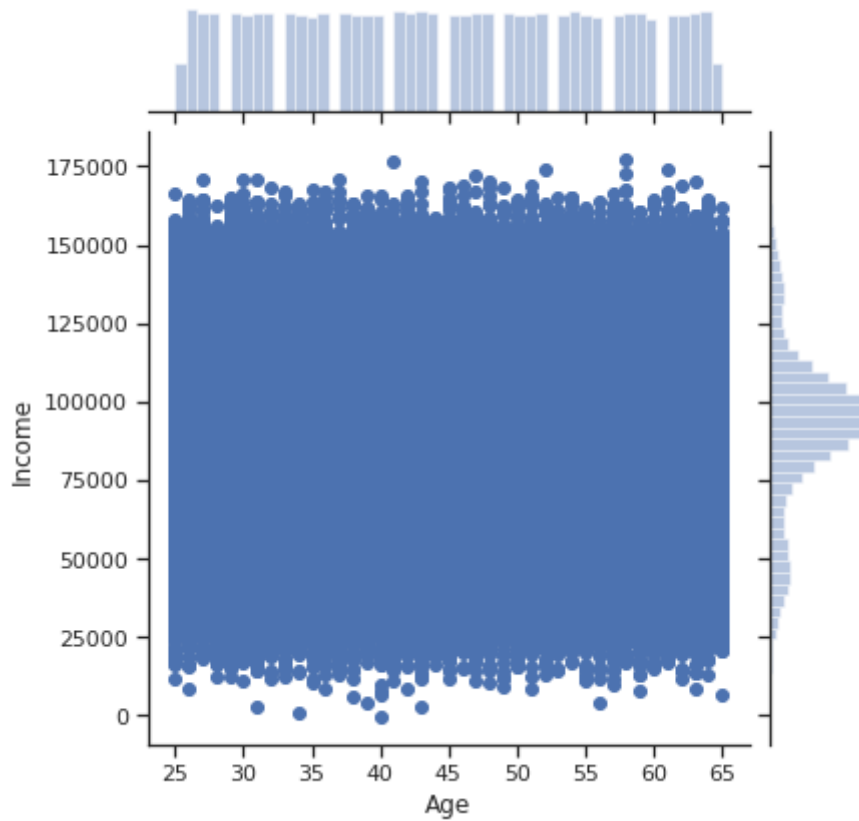


Построенная гистограмма позволяет понять, что наиболее распространенный доход среди

```
#Комбинация гистограмм и диаграммы рассеивания для возраста и дохода  
sns.jointplot(x='Age', y='Income', data=data)
```



```
<seaborn.axisgrid.JointGrid at 0x7f1622a25780>
```

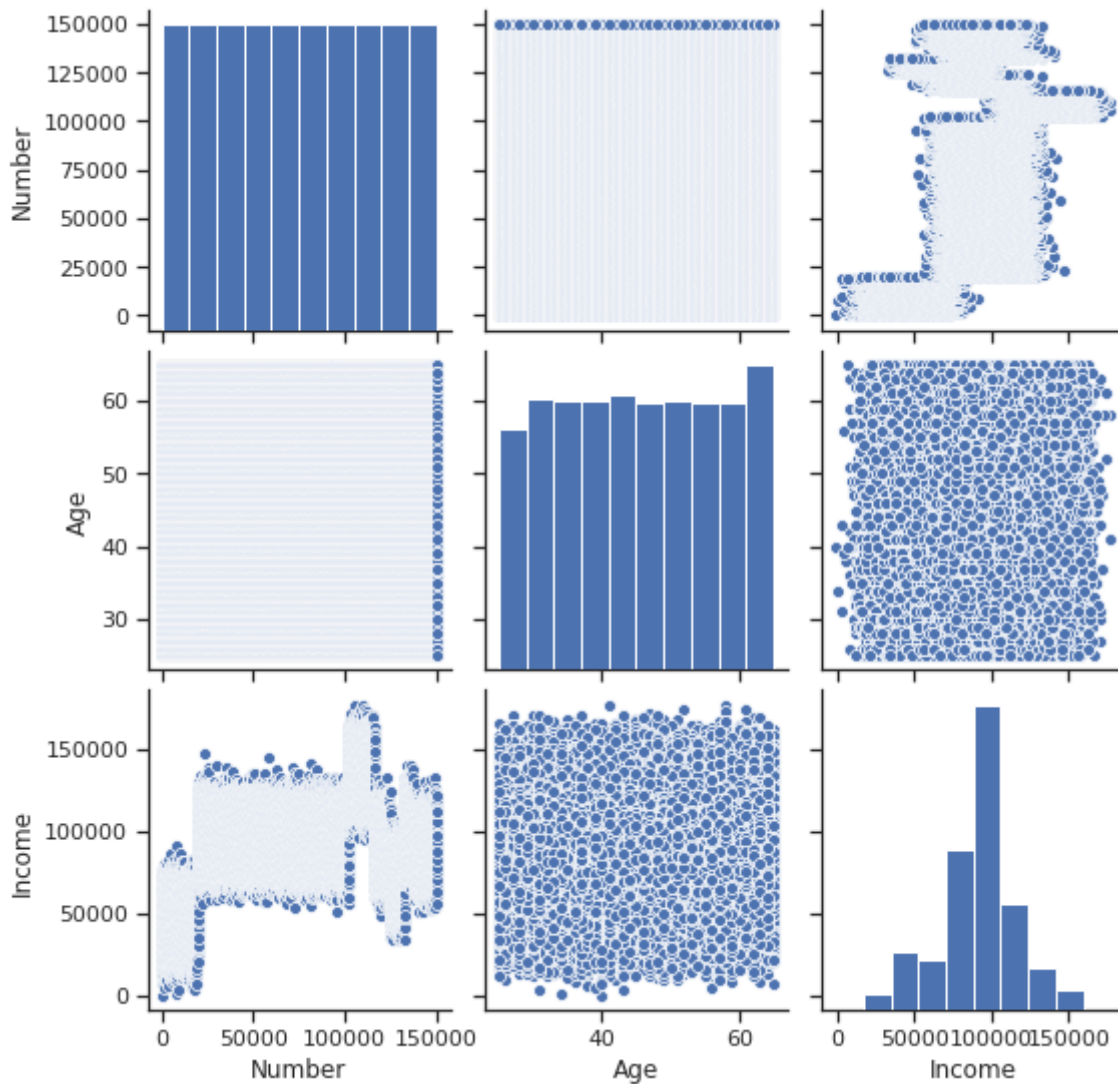


Приведенная выше диаграмма дает понять, что возраст очень слабо коррелирует с доходом. Любого возраста может иметь любой доход.

```
#Строим парные диаграммы для всего набора данных  
sns.pairplot(data)
```



```
<seaborn.axisgrid.PairGrid at 0x7f1621204278>
```



После построения парных диаграмм для всего набора данных стало понятно, что объекты типа object из которых состоит половина столбцов в моем датасете (или я не нал

```
#Получим информацию о корреляции признаков
data.corr()
```

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

Как видно из полученной таблицы, оставшиеся признаки коррелируют крайне слабо (все < 0.5). Проявляется между индексом (number) и доходом. Так что, на мой взгляд, для приведенной модели ММО.

Также, единственным выводом из построенных диаграмм является вывод об уровнях дохода, приведенный в этом ноутбуке.