

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
им. Н.Э. Баумана

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Рубежной контроль 1  
по курсу «Методы машинного обучения»

Вариант 4

ИСПОЛНИТЕЛЬ: Горбовцова К.М.  
ФИО

группа ИУ5-24М \_\_\_\_\_  
подпись

"\_\_" \_\_\_\_\_ 2020 г.

ПРЕПОДАВАТЕЛЬ: Гапнюк Ю.Е.  
ФИО

\_\_\_\_\_  
подпись

"\_\_" \_\_\_\_\_ 2020 г.

Москва – 2020

---

## Горбовцова Ксения, ИУ5-24М

### РК №1 по курсу ММО

```
In [40]: import numpy as np
import pandas as pd
import seaborn as sns

import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import matplotlib
plt.style.use('ggplot')
from matplotlib.pyplot import figure

%matplotlib inline
matplotlib.rcParams['figure.figsize'] = (12,8)
```

```
In [41]: filename = "toy_dataset.csv"
```

```
In [42]: data = pd.read_csv(filename)
```

```
In [43]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150000 entries, 0 to 149999
Data columns (total 6 columns):
Number      150000 non-null int64
City        150000 non-null int64
Gender      150000 non-null int64
Age         150000 non-null int64
Income      150000 non-null float64
Illness     150000 non-null int64
dtypes: float64(1), int64(5)
memory usage: 6.9 MB
```

In [44]: `data.describe()`

Out[44]:

	Number	City	Gender	Age	Income	Illness
<b>count</b>	150000.000000	150000.000000	150000.000000	150000.000000	150000.000000	150000.000000
<b>mean</b>	75000.500000	2.309627	0.441333	44.950200	91252.798273	0.003138
<b>std</b>	43301.414527	2.034138	0.496548	11.572486	24989.500948	0.002156
<b>min</b>	1.000000	0.000000	0.000000	25.000000	-654.000000	0.000000
<b>25%</b>	37500.750000	1.000000	0.000000	35.000000	80867.750000	0.000000
<b>50%</b>	75000.500000	2.000000	0.000000	45.000000	93655.000000	0.000000
<b>75%</b>	112500.250000	3.000000	1.000000	55.000000	104519.000000	0.000000
<b>max</b>	150000.000000	7.000000	1.000000	65.000000	177157.000000	1.000000

In [45]: `data.corr()`

Out[45]:

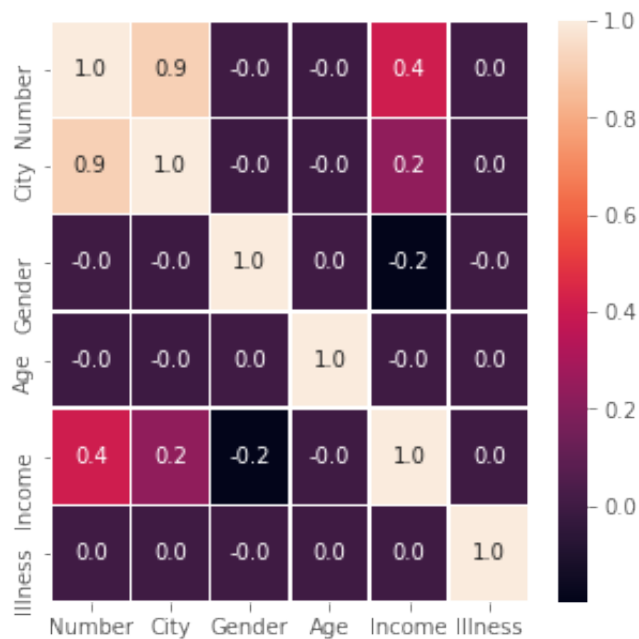
	Number	City	Gender	Age	Income	Illness
<b>Number</b>	1.000000	0.911749	-0.001272	-0.003448	0.410460	0.003138
<b>City</b>	0.911749	1.000000	-0.002404	-0.005615	0.234937	0.002156
<b>Gender</b>	-0.001272	-0.002404	1.000000	0.003653	-0.198888	-0.001297
<b>Age</b>	-0.003448	-0.005615	0.003653	1.000000	-0.001318	0.001811
<b>Income</b>	0.410460	0.234937	-0.198888	-0.001318	1.000000	0.000298
<b>Illness</b>	0.003138	0.002156	-0.001297	0.001811	0.000298	1.000000

```
In [46]: data.hist()
```

```
Out[46]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x1a22ff4f60>,  
                <matplotlib.axes._subplots.AxesSubplot object at 0x1a164abel0>],  
              [<matplotlib.axes._subplots.AxesSubplot object at 0x1a1dd9a6a0>,  
                <matplotlib.axes._subplots.AxesSubplot object at 0x1a1de02eb8>],  
              [<matplotlib.axes._subplots.AxesSubplot object at 0x1a1de6c278>,  
                <matplotlib.axes._subplots.AxesSubplot object at 0x1a1de6c2b0>]],  
          dtype=object)
```

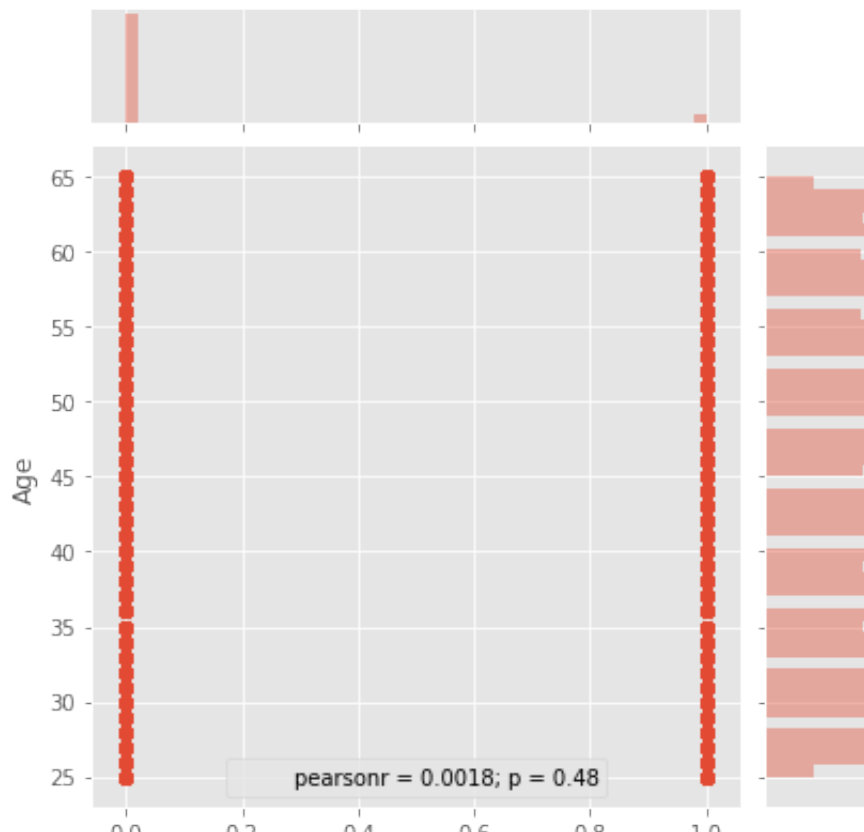


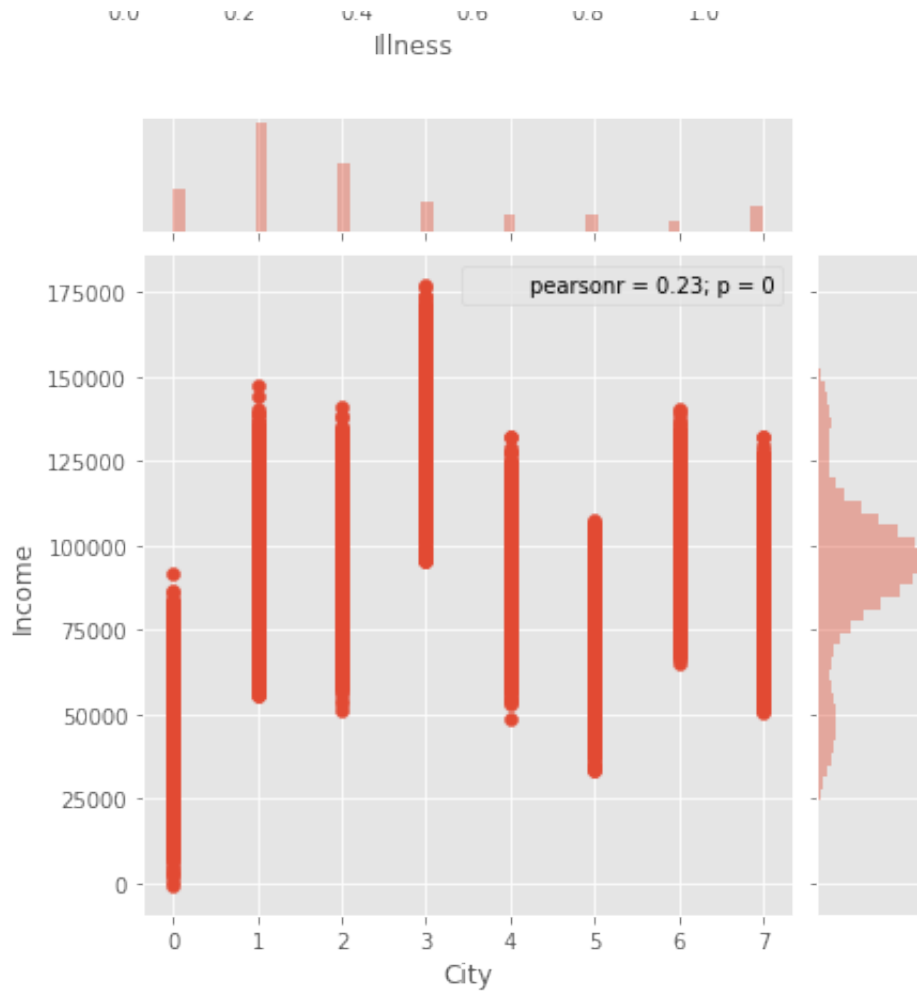
```
In [47]: f,ax = plt.subplots(figsize=(5, 5))
sns.heatmap(data.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
plt.show()
```



```
In [52]: sns.jointplot(x='Illness', y='Age', data=data)
sns.jointplot(x='City', y='Income', data=data)
```

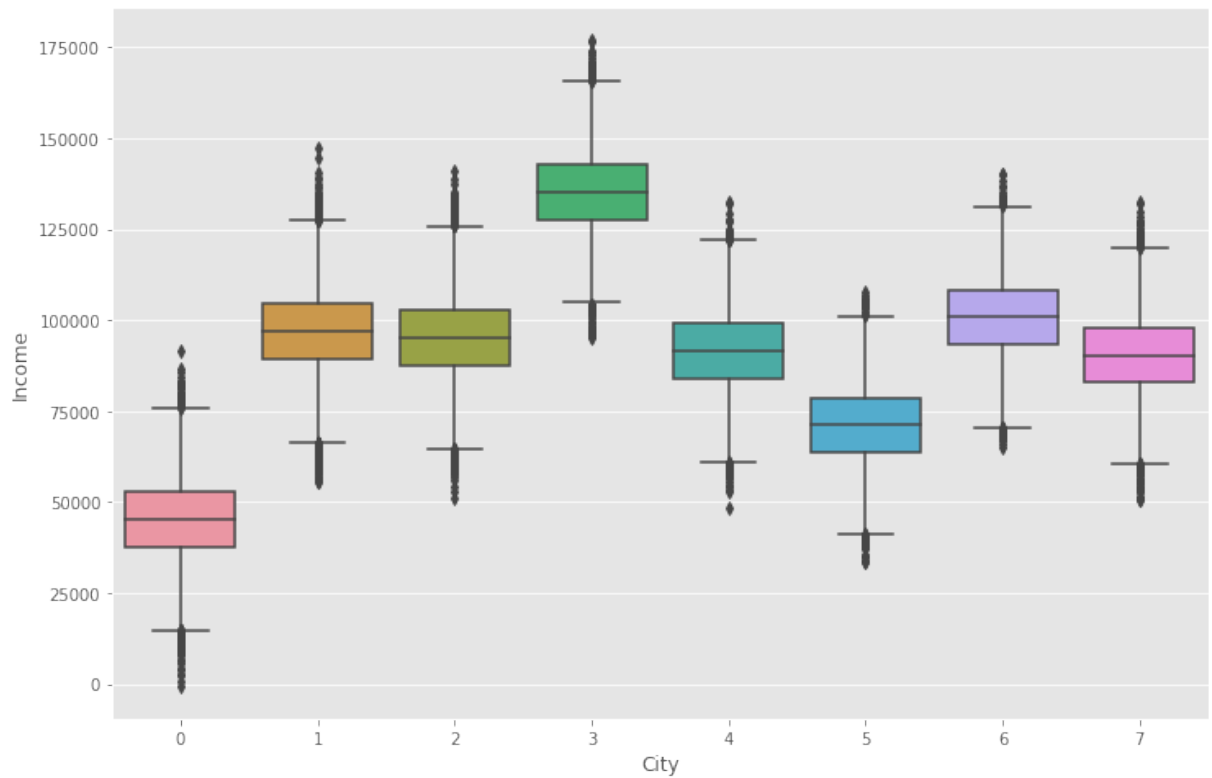
```
Out[52]: <seaborn.axisgrid.JointGrid at 0x1a241f8630>
```





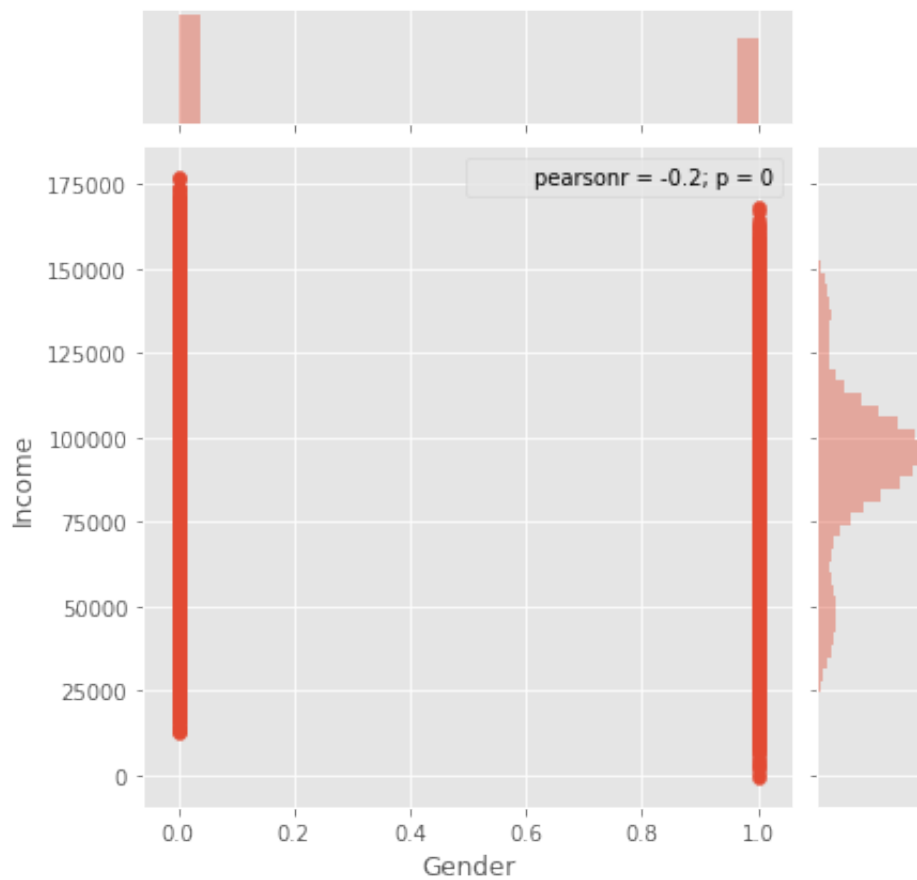
```
In [53]: sns.boxplot(x=data["City"], y=data["Income"])
```

```
Out[53]: <matplotlib.axes._subplots.AxesSubplot at 0x1a2274c048>
```



```
In [61]: sns.jointplot(x='Gender', y='Income', data=data)
```

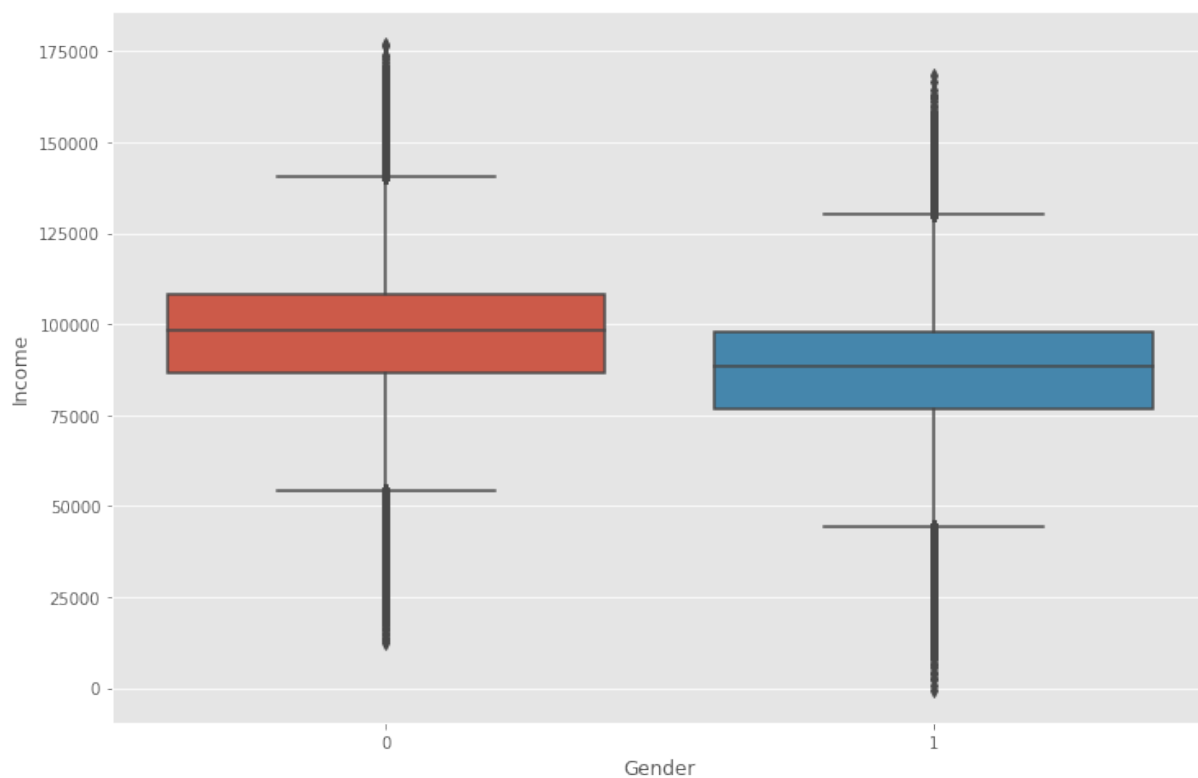
```
Out[61]: <seaborn.axisgrid.JointGrid at 0x1a263ed748>
```





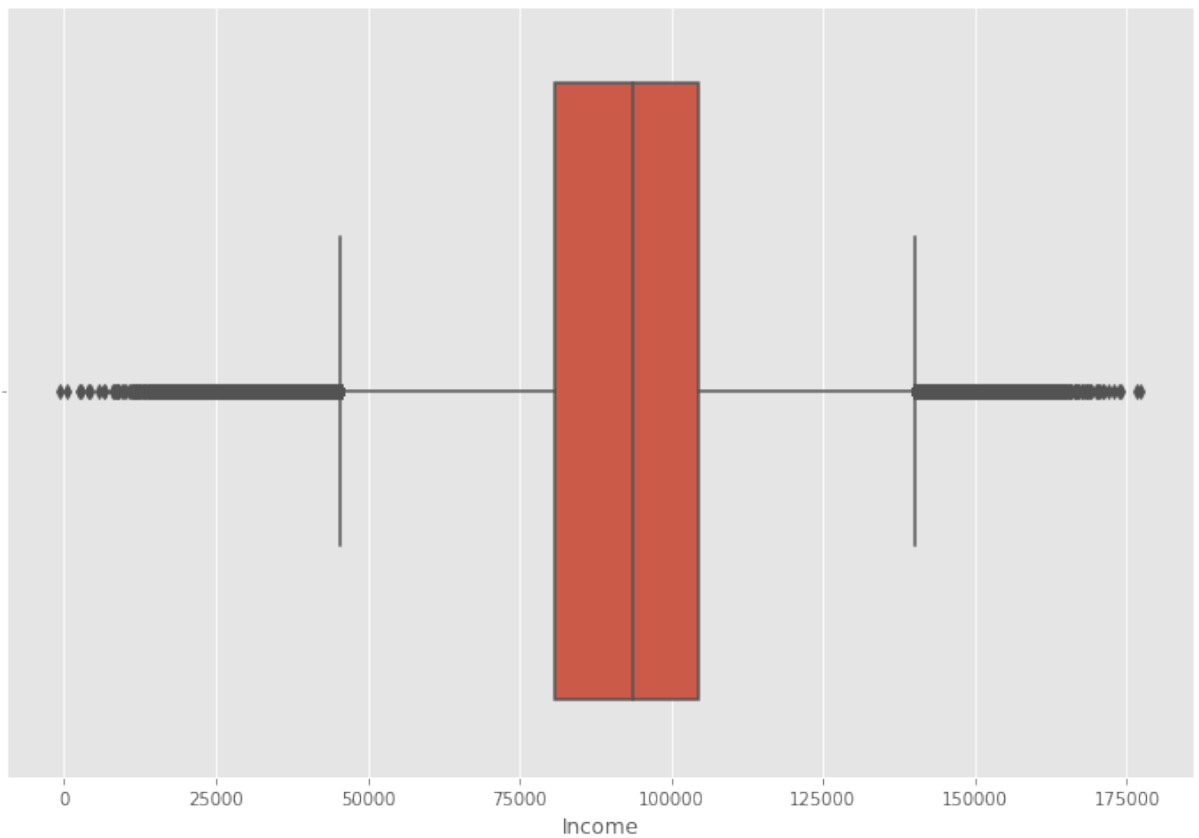
```
In [62]: sns.boxplot(x=data["Gender"], y=data["Income"])
```

```
Out[62]: <matplotlib.axes._subplots.AxesSubplot at 0x1a26f3ca90>
```



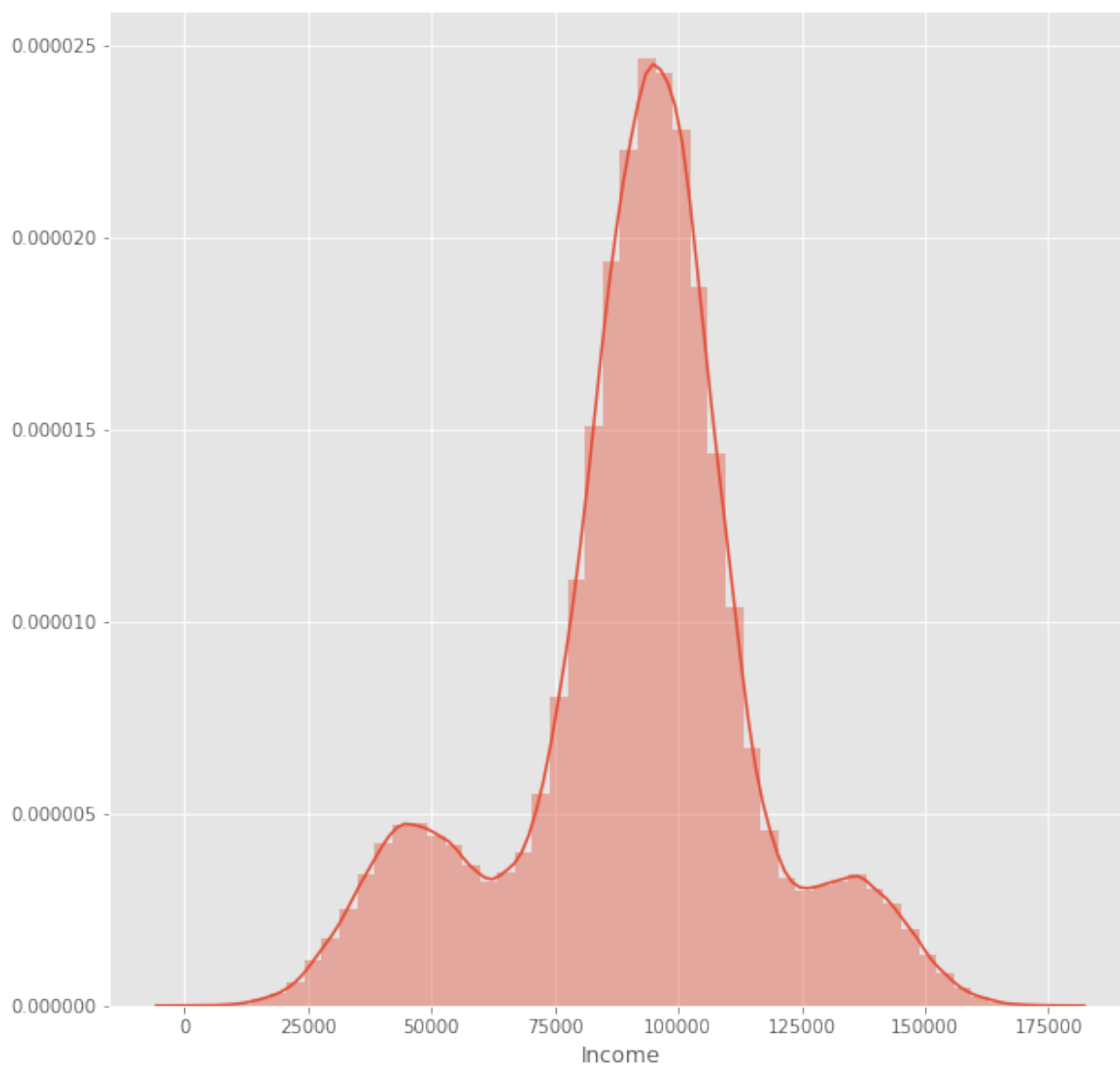
```
In [63]: sns.boxplot(x=data["Income"])
```

```
Out[63]: <matplotlib.axes._subplots.AxesSubplot at 0x1a27fea710>
```



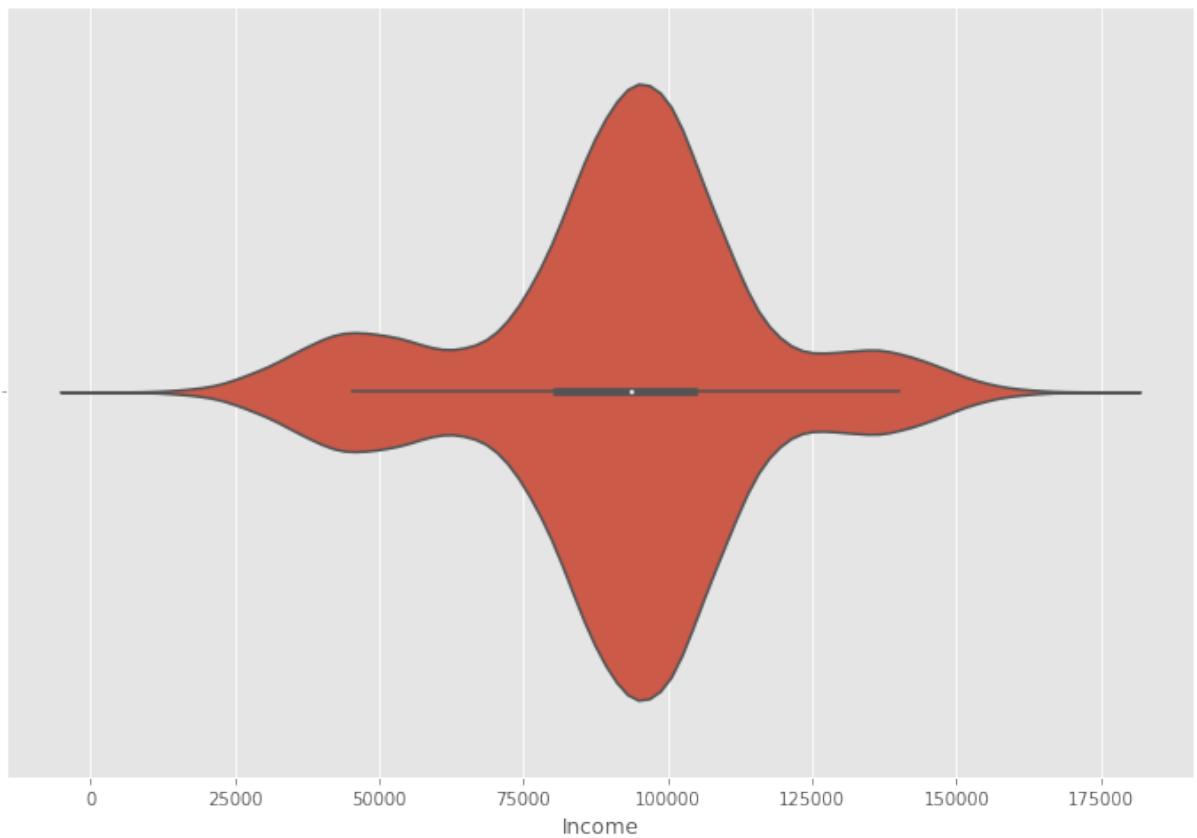
```
In [64]: fig, ax = plt.subplots(figsize=(10,10))  
sns.distplot(data['Income'])
```

```
Out[64]: <matplotlib.axes._subplots.AxesSubplot at 0x1a273b87f0>
```



```
In [66]: sns.violinplot(x=data["Income"])
```

```
Out[66]: <matplotlib.axes._subplots.AxesSubplot at 0x1a280cd978>
```



По данным датасета нельзя построить каких-либо точных предсказаний.

```
In [ ]:
```