```
{
  "authors": [
    {
      "name": "Алексеев Андрей Сергеевич"
    }
  ],
  "group": "ИУ5-62Б",
  "kernelspec": {
    "name": "python3",
    "display_name": "Python 3 (ipykernel)",
    "language": "python"
  },
  "lab_number": 1,
  "language_info": {
    "name": "python",
    "version": "3.9.7",
    "mimetype": "text/x-python",
    "codemirror_mode": {
      "name": "ipython",
      "version": 3
    },
    "pygments_lexer": "ipython3",
    "nbconvert_exporter": "python",
    "file_extension": ".py"
  },
  "title": "Разведочный анализ данных. Исследование и визуализация данных"
}
```


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

```
data = pd.read_csv('melb_data.csv', sep=",")
```

```
# Первые 5 строк датасета
data.head()
```

```
       Suburb           Address  Rooms Type      Price Method SellerG
\
0  Abbotsford      85 Turner St      2    h  1480000.0      S  Biggin

1  Abbotsford   25 Bloomburg St      2    h  1035000.0      S  Biggin

2  Abbotsford      5 Charles St      3    h  1465000.0     SP  Biggin

3  Abbotsford  40 Federation La      3    h   850000.0     PI  Biggin
```

```
4  Abbotsford       55a Park St      4    h  1600000.0     VB  Nelson


        Date  Distance  Postcode  ...  Bathroom  Car  Landsize
BuildingArea  \
0  3/12/2016       2.5    3067.0  ...       1.0  1.0     202.0
NaN
1  4/02/2016       2.5    3067.0  ...       1.0  0.0     156.0
79.0
2  4/03/2017       2.5    3067.0  ...       2.0  0.0     134.0
150.0
3  4/03/2017       2.5    3067.0  ...       2.0  1.0      94.0
NaN
4  4/06/2016       2.5    3067.0  ...       1.0  2.0     120.0
142.0

   YearBuilt  CouncilArea Lattitude  Longtitude             Regionname
\
0        NaN        Yarra  -37.7996    144.9984  Northern Metropolitan

1     1900.0        Yarra  -37.8079    144.9934  Northern Metropolitan

2     1900.0        Yarra  -37.8093    144.9944  Northern Metropolitan

3        NaN        Yarra  -37.7969    144.9969  Northern Metropolitan

4     2014.0        Yarra  -37.8072    144.9941  Northern Metropolitan


  Propertycount
0        4019.0
1        4019.0
2        4019.0
3        4019.0
4        4019.0

[5 rows x 21 columns]
```

```
# Размер датасета - 13580 строк, 21 колонок
data.shape
```

```
(13580, 21)
```

```
total_count = data.shape[0]
print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 13580
```

```
# Список колонок
data.columns
```

```
Index(['Suburb', 'Address', 'Rooms', 'Type', 'Price', 'Method',
'SellerG',
       'Date', 'Distance', 'Postcode', 'Bedroom2', 'Bathroom', 'Car',
       'Landsize', 'BuildingArea', 'YearBuilt', 'CouncilArea',
'Lattitude',
       'Longtitude', 'Regionname', 'Propertycount'],
      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))
```

```
Suburb - 0
Address - 0
Rooms - 0
Type - 0
Price - 0
Method - 0
SellerG - 0
Date - 0
Distance - 0
Postcode - 0
Bedroom2 - 0
Bathroom - 0
Car - 62
Landsize - 0
BuildingArea - 6450
YearBuilt - 5375
CouncilArea - 1369
Lattitude - 0
Longtitude - 0
Regionname - 0
Propertycount - 0
```

```
# Список колонок с типами данных
data.dtypes
```

```
Suburb           object
Address          object
Rooms             int64
Type             object
Price           float64
Method           object
SellerG          object
Date             object
Distance        float64
Postcode        float64
Bedroom2        float64
```

```
Bathroom         float64
Car              float64
Landsize         float64
BuildingArea     float64
YearBuilt        float64
CouncilArea       object
Lattitude        float64
Longtitude       float64
Regionname        object
Propertycount    float64
dtype: object
```

```
data.describe()
```

```
              Rooms         Price      Distance      Postcode
Bedroom2  \
count  13580.000000  1.358000e+04  13580.000000  13580.000000
13580.000000
mean       2.937997  1.075684e+06     10.137776   3105.301915
2.914728
std        0.955748  6.393107e+05      5.868725     90.676964
0.965921
min        1.000000  8.500000e+04      0.000000   3000.000000
0.000000
25%        2.000000  6.500000e+05      6.100000   3044.000000
2.000000
50%        3.000000  9.030000e+05      9.200000   3084.000000
3.000000
75%        3.000000  1.330000e+06     13.000000   3148.000000
3.000000
max       10.000000  9.000000e+06     48.100000   3977.000000
20.000000

           Bathroom           Car       Landsize  BuildingArea
YearBuilt  \
count  13580.000000  13518.000000   13580.000000   7130.000000
8205.000000
mean       1.534242      1.610075     558.416127    151.967650
1964.684217
std        0.691712      0.962634    3990.669241    541.014538
37.273762
min        0.000000      0.000000       0.000000      0.000000
1196.000000
25%        1.000000      1.000000     177.000000     93.000000
1940.000000
50%        1.000000      2.000000     440.000000    126.000000
1970.000000
75%        2.000000      2.000000     651.000000    174.000000
1999.000000
max        8.000000     10.000000  433014.000000  44515.000000
```

```
2018.000000
```

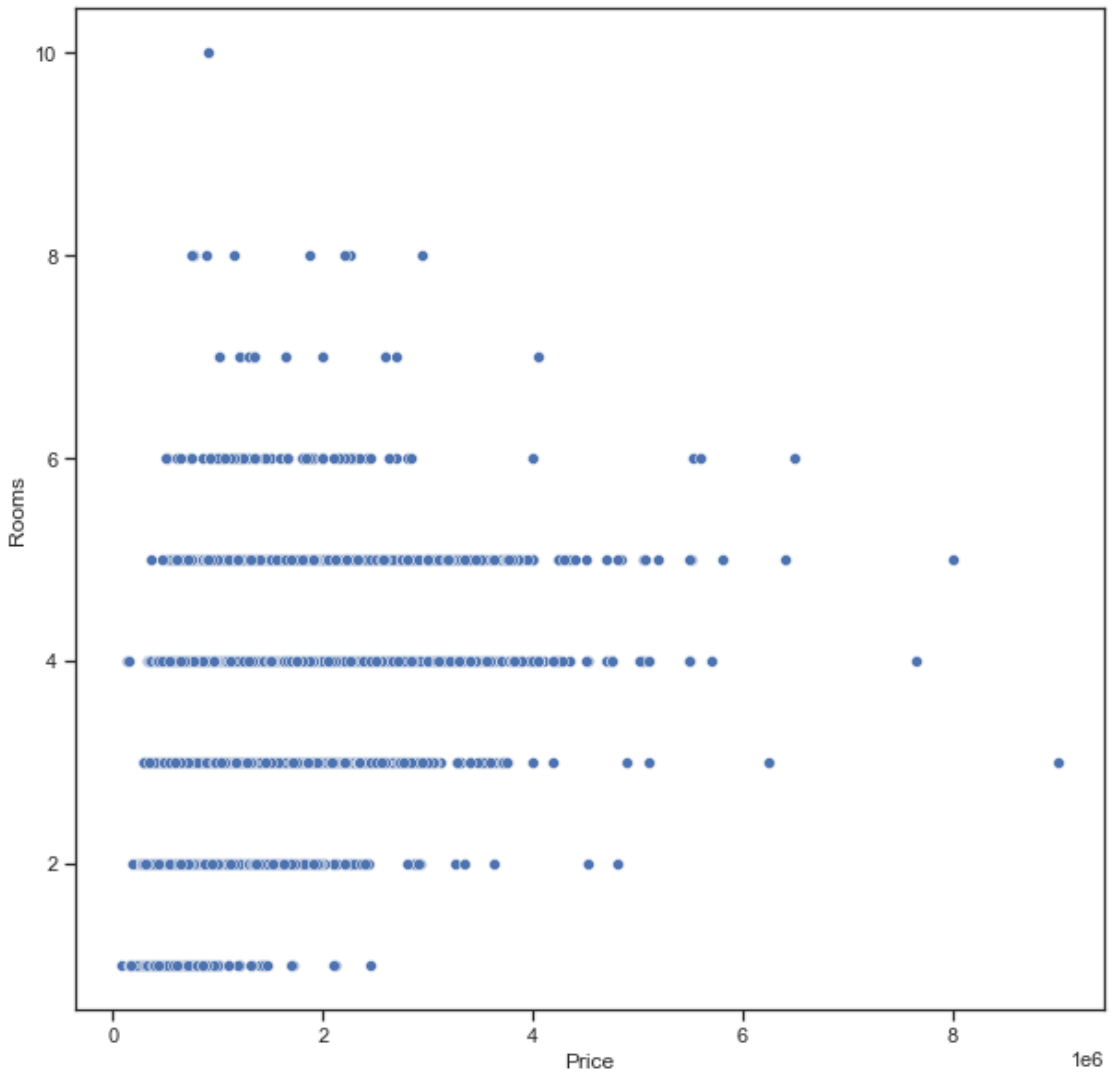
```
          Lattitude    Longtitude  Propertycount
count  13580.000000  13580.000000   13580.000000
mean     -37.809203    144.995216    7454.417378
std        0.079260      0.103916    4378.581772
min      -38.182550    144.431810     249.000000
25%      -37.856822    144.929600    4380.000000
50%      -37.802355    145.000100    6555.000000
75%      -37.756400    145.058305   10331.000000
max      -37.408530    145.526350   21650.000000
```

```
# Определим уникальные значения для целевого признака
data['Rooms'].unique()
```

```
array([ 2,  3,  4,  1,  6,  5,  8,  7, 10], dtype=int64)
```

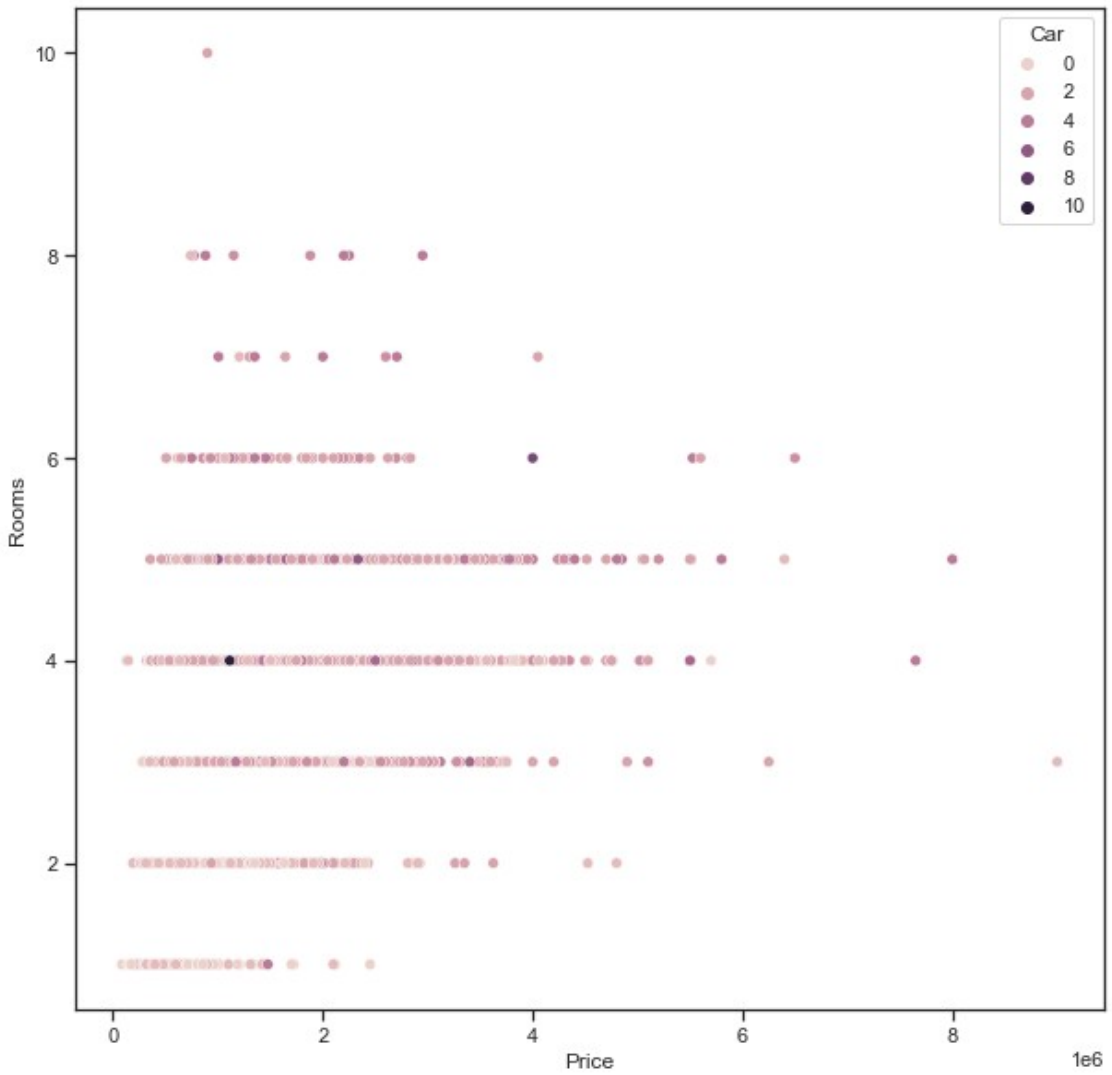
```
# Рассмотрим распределение цены за дом по количеству комнат
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Price', y='Rooms', data=data)
```

```
<AxesSubplot:xlabel='Price', ylabel='Rooms'>
```




```
# Распределение цен на дома по количеству комнат и количеству
парковочных мест
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Price', y='Rooms', data=data, hue='Car')
```

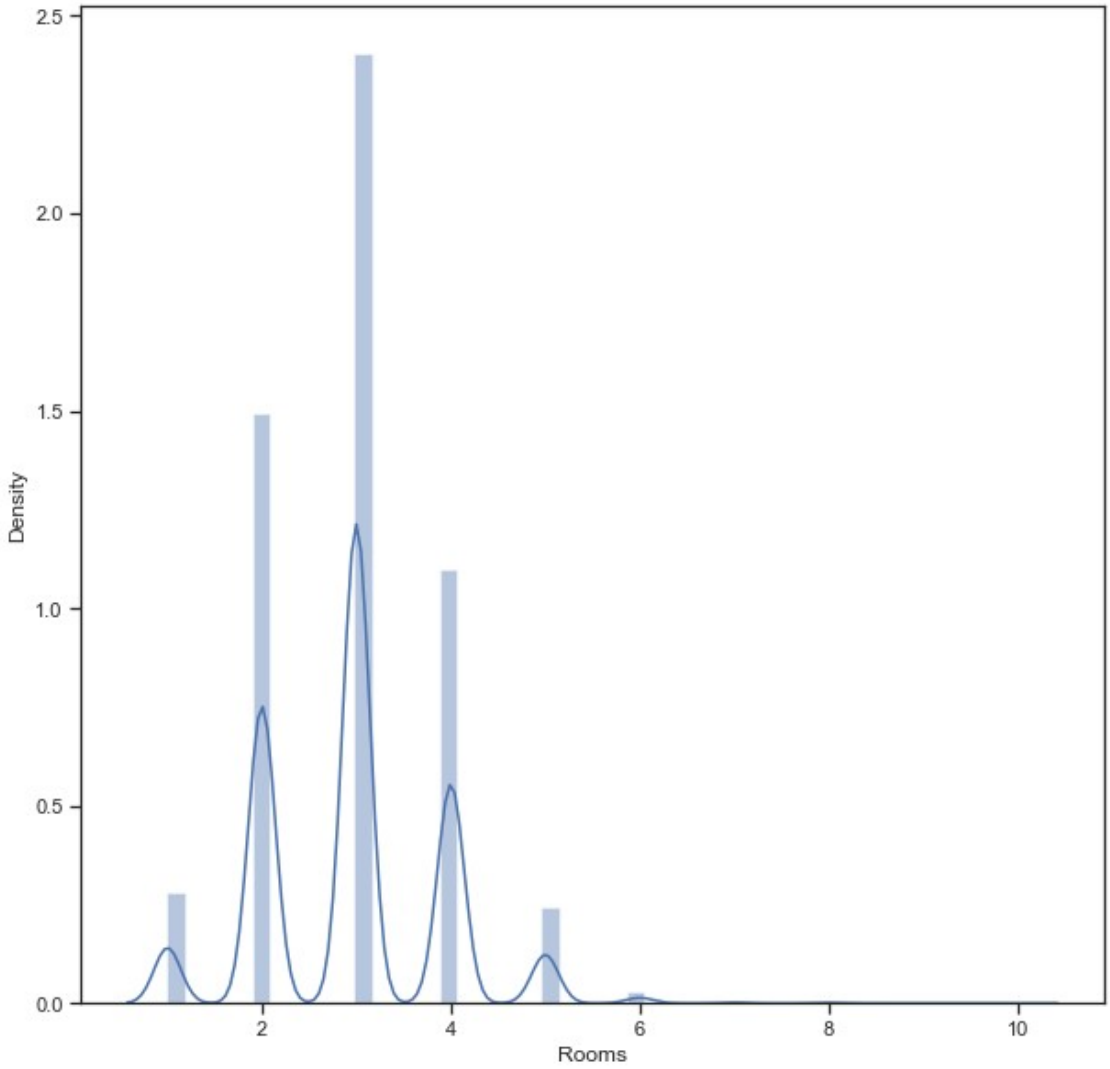
```
<AxesSubplot:xlabel='Price', ylabel='Rooms'>
```




```
# Гистограмма
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['Rooms'])
```

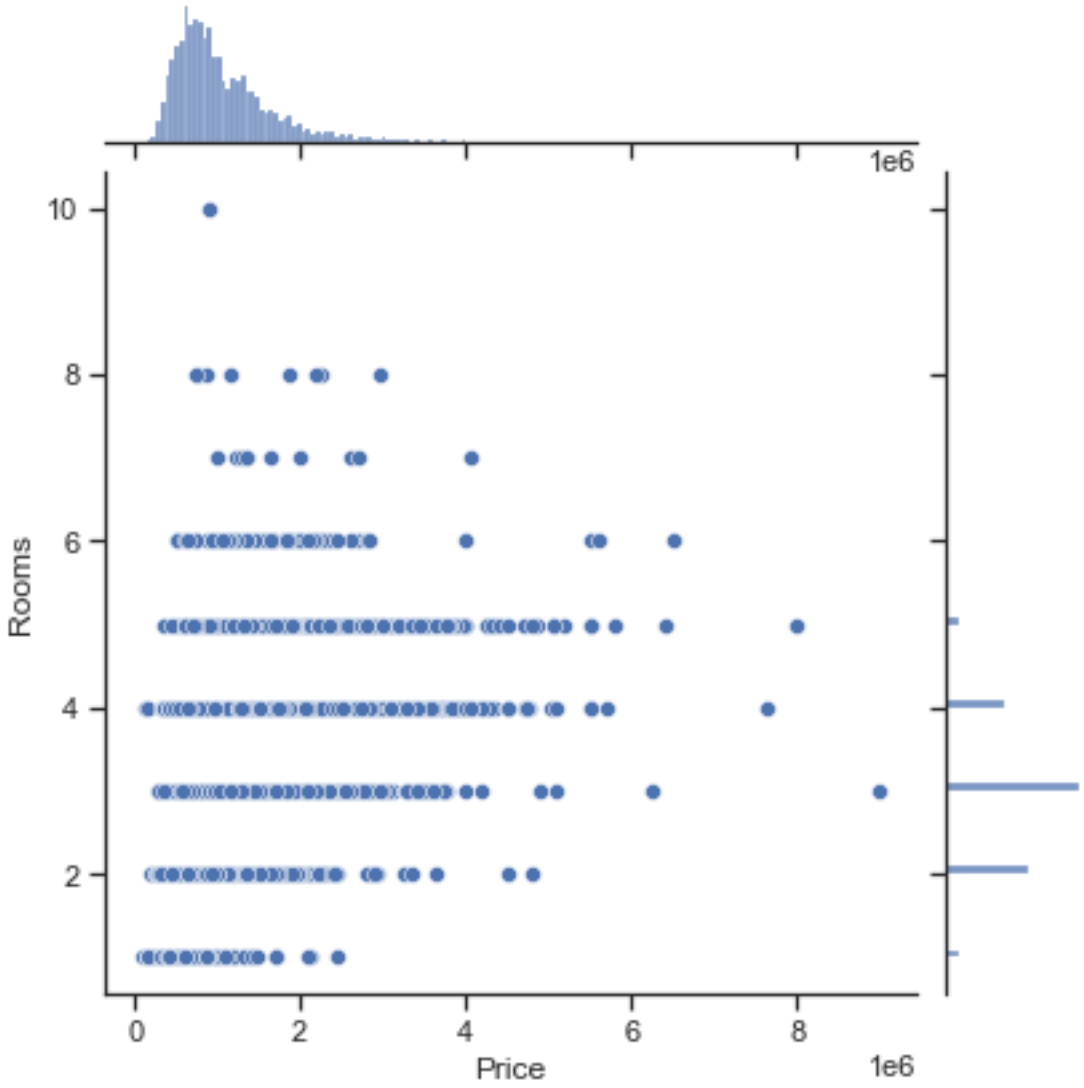
```
C:\Users\Admин\AppData\Local\Packages\
PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-
packages\Python39\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an
axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
<AxesSubplot:xlabel='Rooms', ylabel='Density'>
```



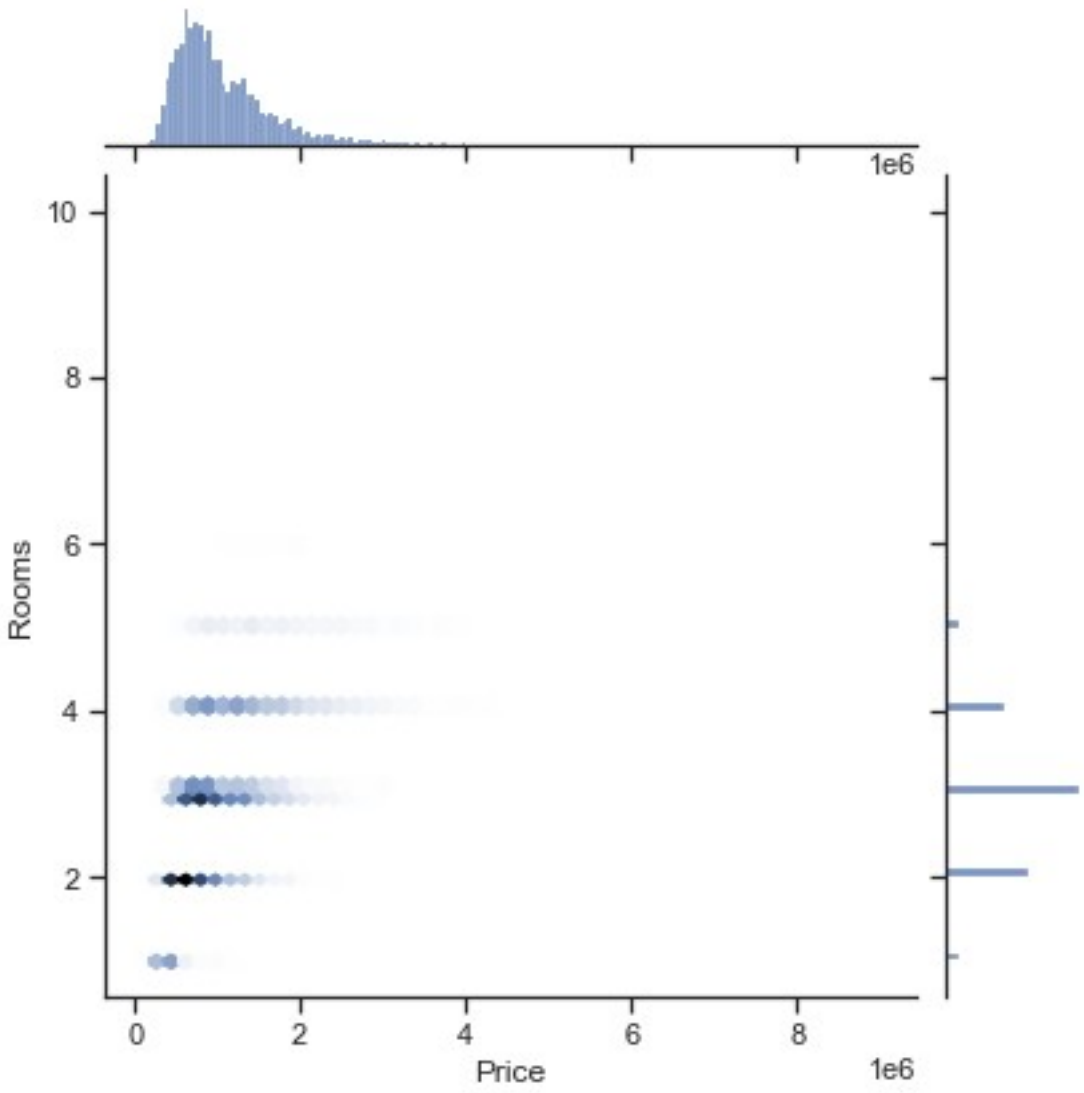

```
# Комбинация гистограмм и диаграмм рассеивания
sns.jointplot(x='Price', y='Rooms', data=data)
```

```
<seaborn.axisgrid.JointGrid at 0x1905529b550>
```



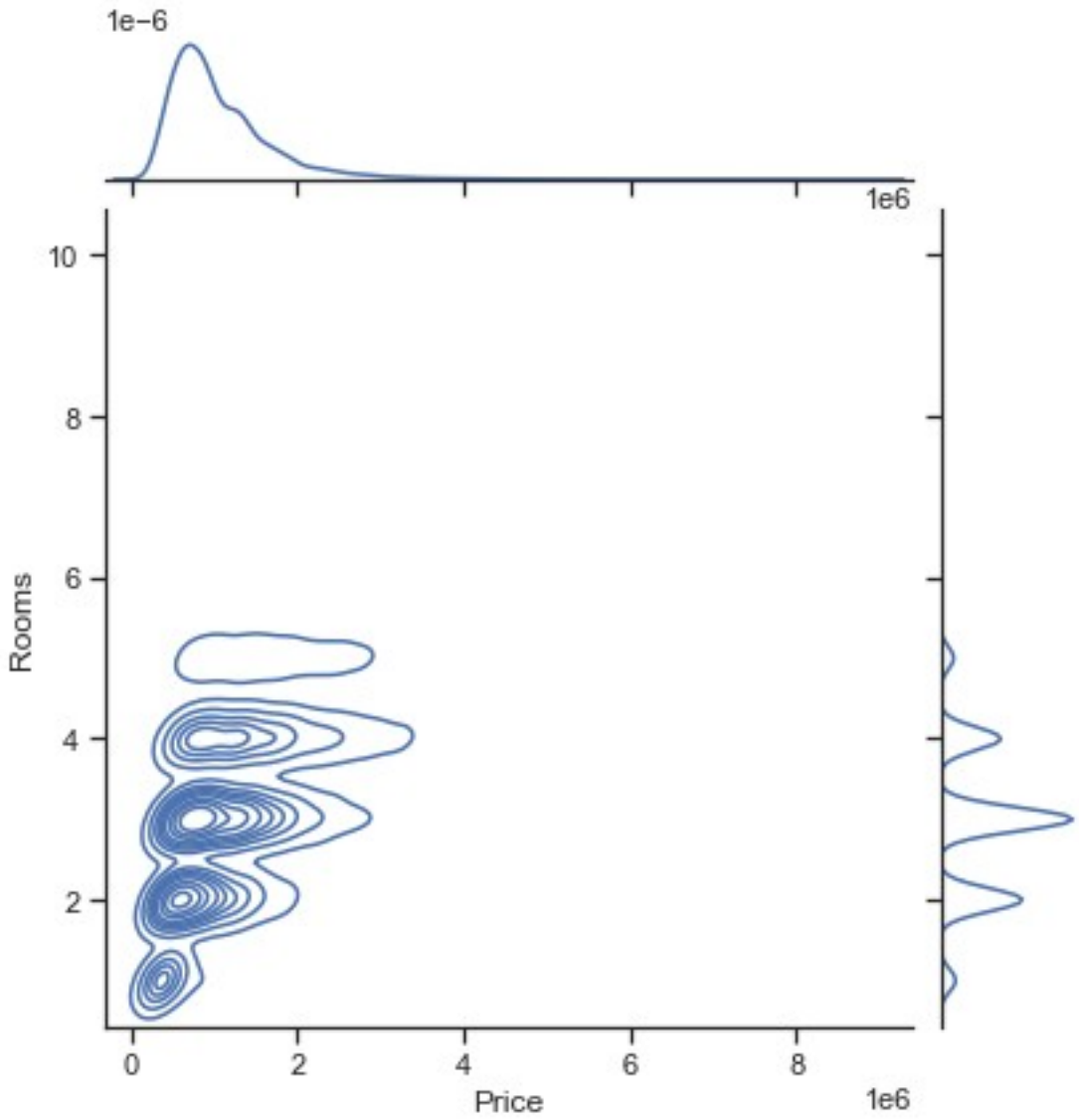

```
sns.jointplot(x='Price', y='Rooms', data=data, kind="hex")
```

```
<seaborn.axisgrid.JointGrid at 0x19051d12130>
```



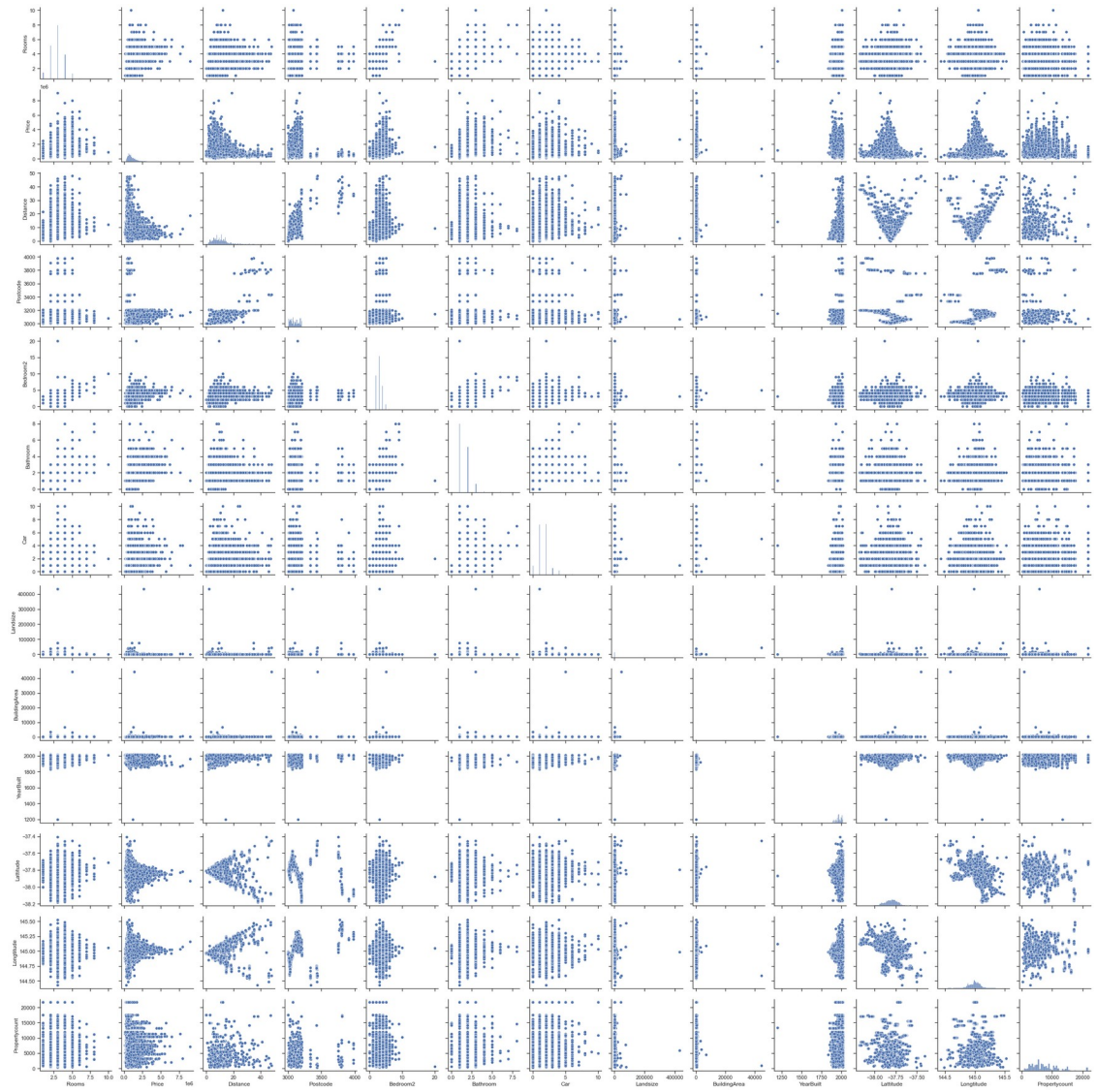

```
sns.jointplot(x='Price', y='Rooms', data=data, kind="kde")
```

```
<seaborn.axisgrid.JointGrid at 0x190552941c0>
```



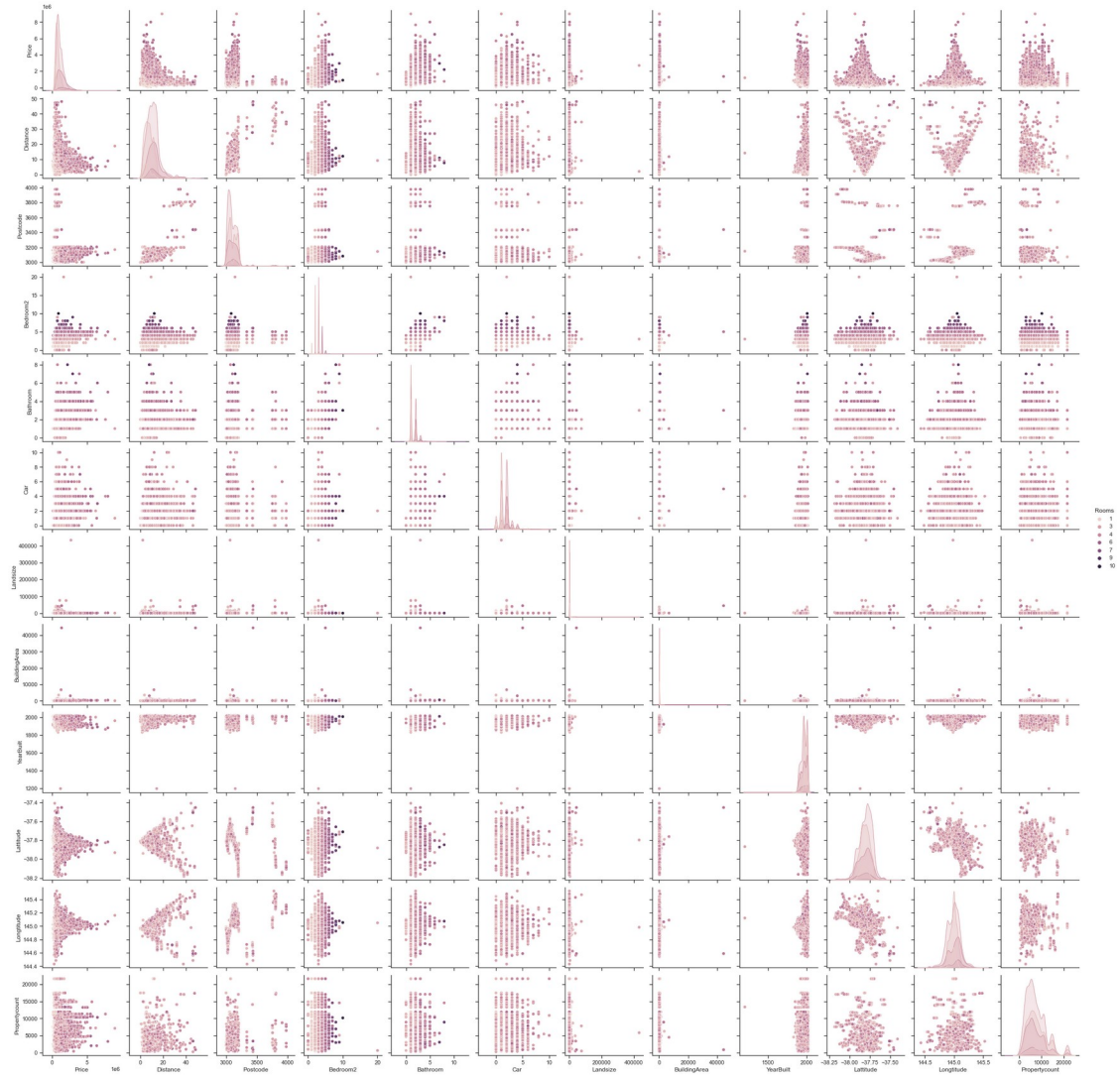

```
# Комбинация гистограмм и диаграмм рассеивания для всего набора данных
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x19055fb4f10>
```



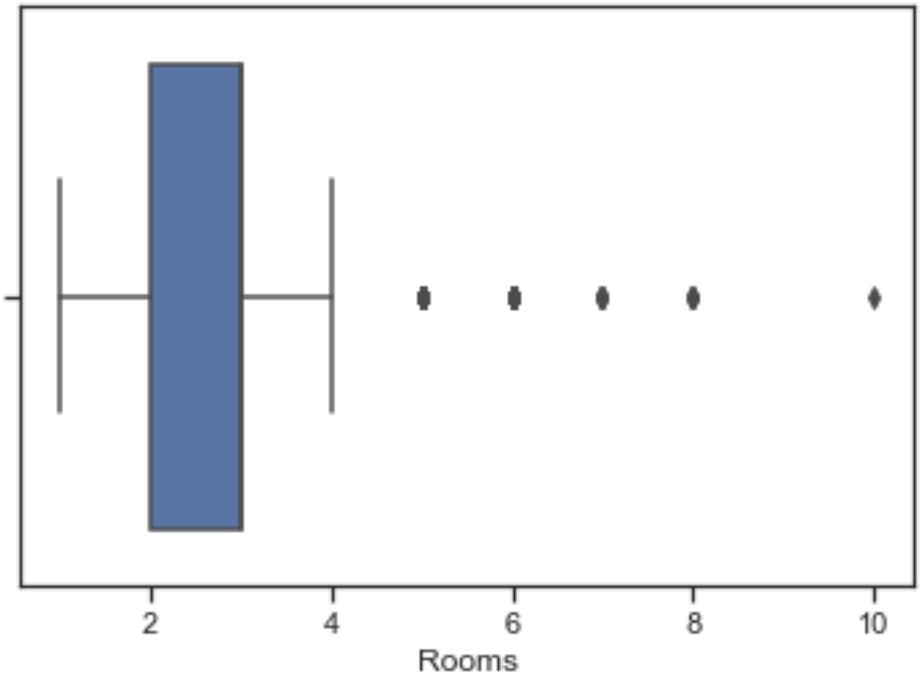

```
sns.pairplot(data, hue="Rooms")
```

```
<seaborn.axisgrid.PairGrid at 0x1906c83da60>
```



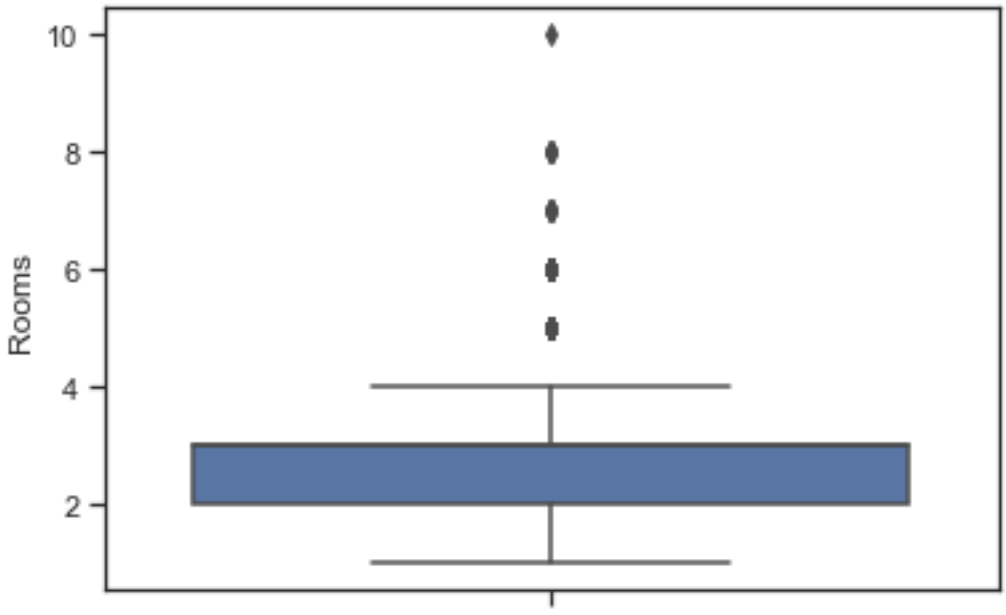

```
# Ящик с усами
sns.boxplot(x=data['Rooms'])
```

```
<AxesSubplot:xlabel='Rooms'>
```



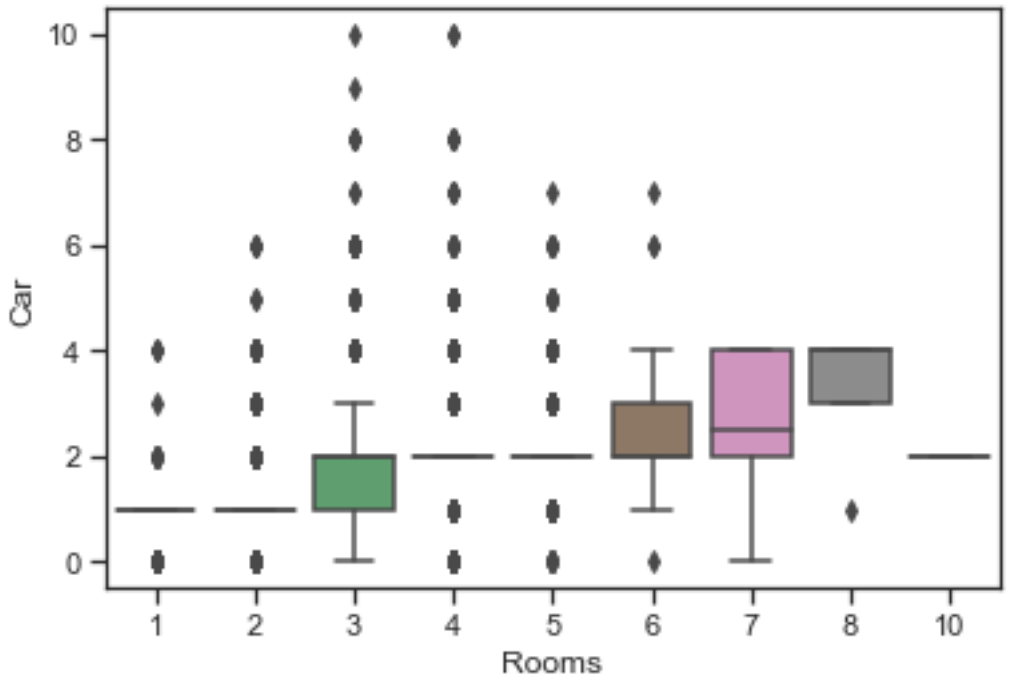

```
# По вертикали
sns.boxplot(y=data['Rooms'])
```

```
<AxesSubplot:ylabel='Rooms'>
```



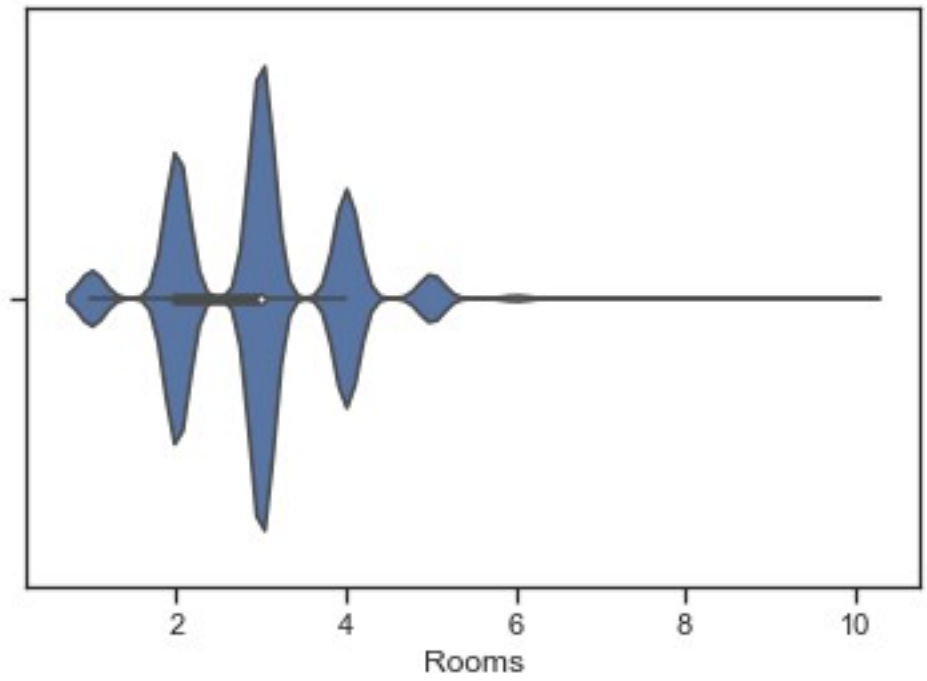

```
# Распределение параметра Rooms сгруппированные по Car.
sns.boxplot(x='Rooms', y='Car', data=data)
```

```
<AxesSubplot:xlabel='Rooms', ylabel='Car'>
```




```
# Violin plot
sns.violinplot(x=data['Rooms'])
```

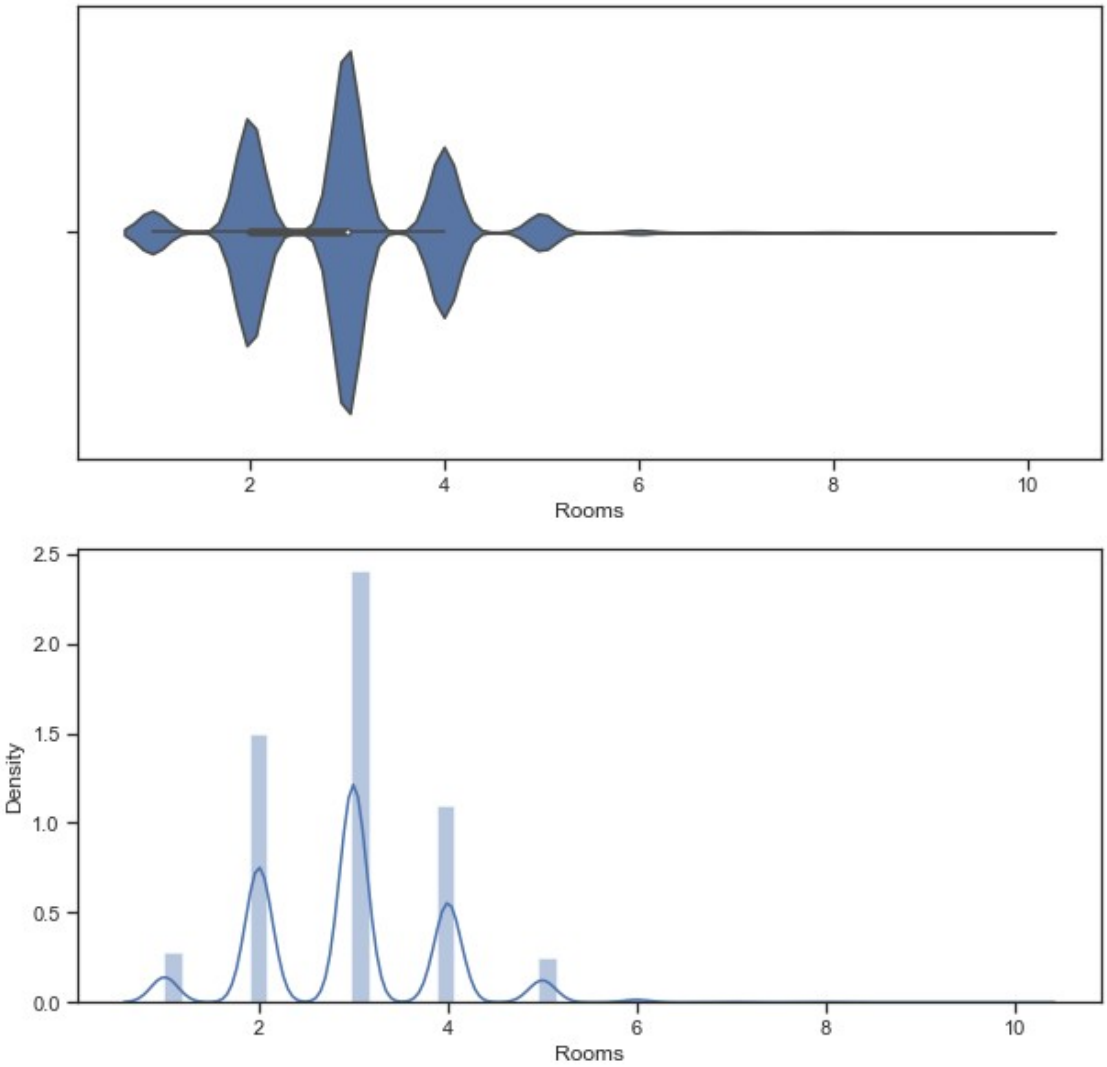
```
<AxesSubplot:xlabel='Rooms'>
```




```
fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=data['Rooms'])
sns.distplot(data['Rooms'], ax=ax[1])
```

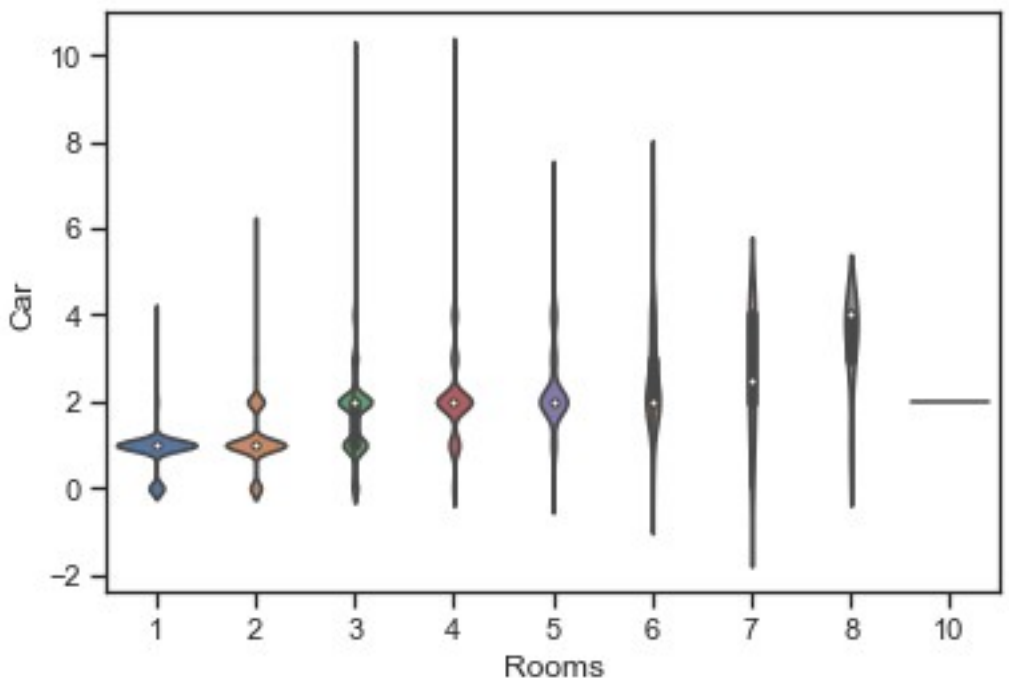
```
C:\Users\Admин\AppData\Local\Packages\
PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-
packages\Python39\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an
axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
<AxesSubplot:xlabel='Rooms', ylabel='Density'>
```



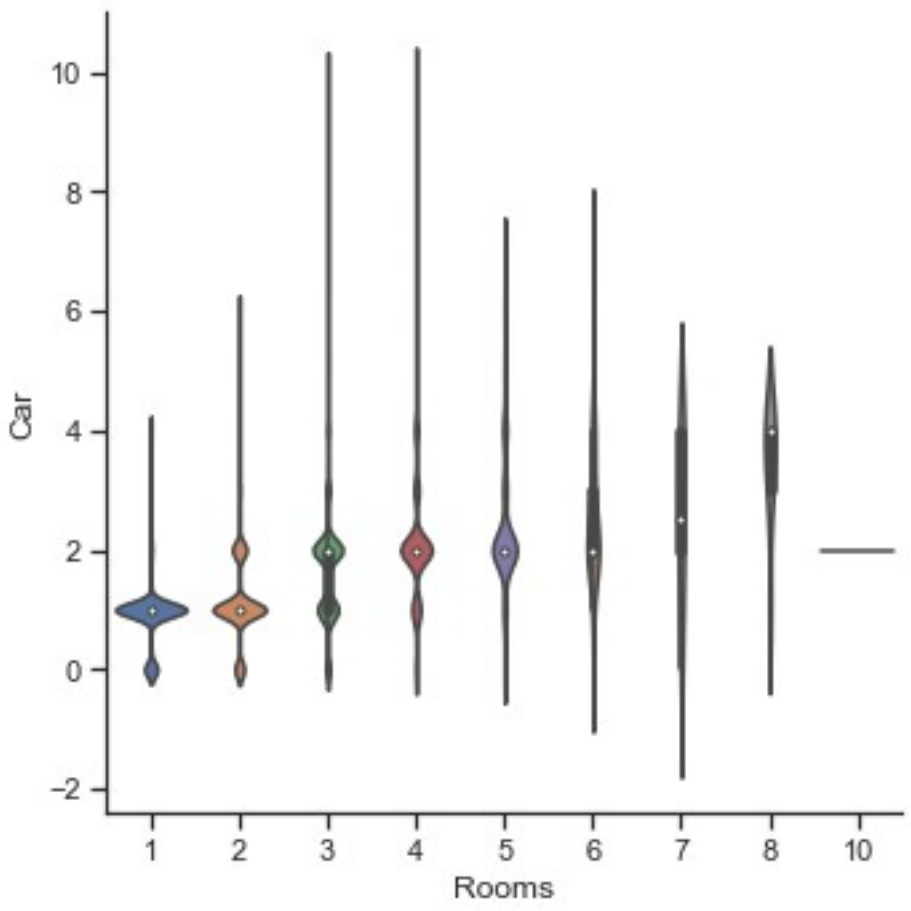

```
# Распределение параметра Rooms сгруппированные по Car.
sns.violinplot(x='Rooms', y='Car', data=data)
```

```
<AxesSubplot:xlabel='Rooms', ylabel='Car'>
```




```
sns.catplot(y='Car', x='Rooms', data=data, kind="violin", split=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x1900e819100>
```




```
data.corr()
```

	Rooms	Price	Distance	Postcode	Bedroom2	Bathroom
Rooms	1.000000	0.496634	0.294203	0.055303	0.944190	0.592934
Price	0.496634	1.000000	-0.162522	0.107867	0.475951	0.467038
Distance	0.294203	-0.162522	1.000000	0.431514	0.295927	0.127155
Postcode	0.055303	0.107867	0.431514	1.000000	0.060584	0.113664
Bedroom2	0.944190	0.475951	0.295927	0.060584	1.000000	0.584685
Bathroom	0.592934	0.467038	0.127155	0.113664	0.584685	1.000000
Car	0.408483	0.238979	0.262994	0.050289	0.405325	0.322246
Landsize	0.025678	0.037507	0.025004	0.024558	0.025646	0.037130
BuildingArea	0.124127	0.090981	0.099481	0.055475	0.122319	0.111933

```
YearBuilt     -0.065413 -0.323617  0.246379  0.032863 -0.053319
0.152702
Lattitude      0.015948 -0.212934 -0.130723 -0.406104  0.015925 -
0.070594
Longtitude     0.100771  0.203656  0.239425  0.445357  0.102238
0.118971
Propertycount -0.081530 -0.042153 -0.054910  0.062304 -0.081350 -
0.052201

                    Car  Landsize  BuildingArea  YearBuilt  Lattitude
\
Rooms          0.408483  0.025678      0.124127  -0.065413   0.015948

Price          0.238979  0.037507      0.090981  -0.323617  -0.212934

Distance       0.262994  0.025004      0.099481   0.246379  -0.130723

Postcode       0.050289  0.024558      0.055475   0.032863  -0.406104

Bedroom2       0.405325  0.025646      0.122319  -0.053319   0.015925

Bathroom       0.322246  0.037130      0.111933   0.152702  -0.070594

Car            1.000000  0.026770      0.096101   0.104515  -0.001963

Landsize       0.026770  1.000000      0.500485   0.036451   0.009695

BuildingArea   0.096101  0.500485      1.000000   0.019665   0.043420

YearBuilt      0.104515  0.036451      0.019665   1.000000   0.060445

Lattitude     -0.001963  0.009695      0.043420   0.060445   1.000000

Longtitude     0.063395  0.010833     -0.023810  -0.003470  -0.357634

Propertycount -0.024295 -0.006854     -0.028840   0.006361   0.047086


               Longtitude  Propertycount
Rooms            0.100771      -0.081530
Price            0.203656      -0.042153
Distance         0.239425      -0.054910
Postcode         0.445357       0.062304
Bedroom2         0.102238      -0.081350
Bathroom         0.118971      -0.052201
Car              0.063395      -0.024295
Landsize         0.010833      -0.006854
BuildingArea    -0.023810      -0.028840
YearBuilt       -0.003470       0.006361
```

```
Lattitude       -0.357634       0.047086
Longtitude       1.000000       0.065988
Propertycount    0.065988       1.000000
```

```
data.corr(method='pearson')
```

```
                  Rooms     Price  Distance  Postcode  Bedroom2
Bathroom  \
Rooms          1.000000  0.496634  0.294203  0.055303  0.944190
0.592934
Price          0.496634  1.000000 -0.162522  0.107867  0.475951
0.467038
Distance       0.294203 -0.162522  1.000000  0.431514  0.295927
0.127155
Postcode       0.055303  0.107867  0.431514  1.000000  0.060584
0.113664
Bedroom2       0.944190  0.475951  0.295927  0.060584  1.000000
0.584685
Bathroom       0.592934  0.467038  0.127155  0.113664  0.584685
1.000000
Car            0.408483  0.238979  0.262994  0.050289  0.405325
0.322246
Landsize       0.025678  0.037507  0.025004  0.024558  0.025646
0.037130
BuildingArea   0.124127  0.090981  0.099481  0.055475  0.122319
0.111933
YearBuilt     -0.065413 -0.323617  0.246379  0.032863 -0.053319
0.152702
Lattitude      0.015948 -0.212934 -0.130723 -0.406104  0.015925 -
0.070594
Longtitude     0.100771  0.203656  0.239425  0.445357  0.102238
0.118971
Propertycount -0.081530 -0.042153 -0.054910  0.062304 -0.081350 -
0.052201

                   Car  Landsize  BuildingArea  YearBuilt  Lattitude
\
Rooms          0.408483  0.025678      0.124127  -0.065413   0.015948

Price          0.238979  0.037507      0.090981  -0.323617  -0.212934

Distance       0.262994  0.025004      0.099481   0.246379  -0.130723

Postcode       0.050289  0.024558      0.055475   0.032863  -0.406104

Bedroom2       0.405325  0.025646      0.122319  -0.053319   0.015925

Bathroom       0.322246  0.037130      0.111933   0.152702  -0.070594

Car            1.000000  0.026770      0.096101   0.104515  -0.001963
```

```
Landsize       0.026770  1.000000      0.500485   0.036451   0.009695

BuildingArea   0.096101  0.500485      1.000000   0.019665   0.043420

YearBuilt      0.104515  0.036451      0.019665   1.000000   0.060445

Lattitude     -0.001963  0.009695      0.043420   0.060445   1.000000

Longtitude     0.063395  0.010833     -0.023810  -0.003470  -0.357634

Propertycount -0.024295 -0.006854     -0.028840   0.006361   0.047086


               Longtitude  Propertycount
Rooms            0.100771      -0.081530
Price            0.203656      -0.042153
Distance         0.239425      -0.054910
Postcode         0.445357       0.062304
Bedroom2         0.102238      -0.081350
Bathroom         0.118971      -0.052201
Car              0.063395      -0.024295
Landsize         0.010833      -0.006854
BuildingArea    -0.023810      -0.028840
YearBuilt       -0.003470       0.006361
Lattitude       -0.357634       0.047086
Longtitude       1.000000       0.065988
Propertycount    0.065988       1.000000
```

```
data.corr(method='kendall')
```

```
                  Rooms     Price  Distance  Postcode  Bedroom2
Bathroom  \
Rooms          1.000000  0.428875  0.266915  0.022762  0.948765
0.537258
Price          0.428875  1.000000 -0.089653  0.153939  0.415596
0.343686
Distance       0.266915 -0.089653  1.000000  0.159089  0.272685
0.123625
Postcode       0.022762  0.153939  0.159089  1.000000  0.026893
0.098119
Bedroom2       0.948765  0.415596  0.272685  0.026893  1.000000
0.531571
Bathroom       0.537258  0.343686  0.123625  0.098119  0.531571
1.000000
Car            0.415043  0.221869  0.268727  0.043592  0.416386
0.341120
Landsize       0.393809  0.229082  0.290712  0.044693  0.388514
0.171010
BuildingArea   0.654305  0.463391  0.191654  0.052439  0.641065
```

```
0.534286
YearBuilt     -0.055820 -0.254514  0.172429 -0.018382 -0.045410
0.164226
Lattitude      0.030120 -0.177821  0.008626 -0.425359  0.031868 -
0.063523
Longtitude     0.096776  0.171984  0.215278  0.519327  0.099180
0.113078
Propertycount -0.058574 -0.006296 -0.101447  0.084464 -0.058064 -
0.027915

                    Car  Landsize  BuildingArea  YearBuilt  Lattitude
\
Rooms          0.415043  0.393809      0.654305  -0.055820   0.030120

Price          0.221869  0.229082      0.463391  -0.254514  -0.177821

Distance       0.268727  0.290712      0.191654   0.172429   0.008626

Postcode       0.043592  0.044693      0.052439  -0.018382  -0.425359

Bedroom2       0.416386  0.388514      0.641065  -0.045410   0.031868

Bathroom       0.341120  0.171010      0.534286   0.164226  -0.063523

Car            1.000000  0.316381      0.364504   0.082646   0.005668

Landsize       0.316381  1.000000      0.344596  -0.071377   0.043693

BuildingArea   0.364504  0.344596      1.000000   0.000136  -0.011603

YearBuilt      0.082646 -0.071377      0.000136   1.000000   0.046977

Lattitude      0.005668  0.043693     -0.011603   0.046977   1.000000

Longtitude     0.103110  0.126201      0.088549   0.003019  -0.239012

Propertycount -0.033326 -0.050365     -0.054403  -0.003697  -0.022690


               Longtitude  Propertycount
Rooms            0.096776      -0.058574
Price            0.171984      -0.006296
Distance         0.215278      -0.101447
Postcode         0.519327       0.084464
Bedroom2         0.099180      -0.058064
Bathroom         0.113078      -0.027915
Car              0.103110      -0.033326
Landsize         0.126201      -0.050365
BuildingArea     0.088549      -0.054403
```

```
YearBuilt        0.003019      -0.003697
Lattitude       -0.239012      -0.022690
Longtitude       1.000000       0.057458
Propertycount    0.057458       1.000000
```

```
data.corr(method='spearman')
```

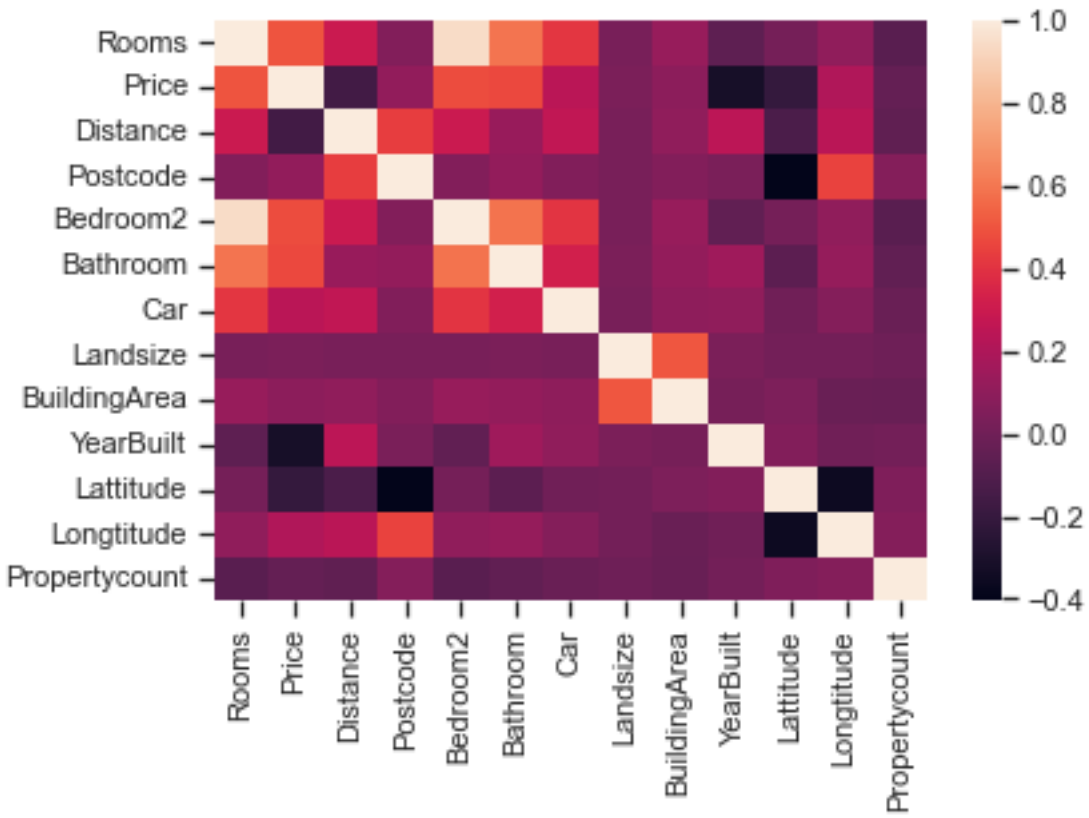
```
                 Rooms     Price  Distance  Postcode  Bedroom2
Bathroom  \
Rooms          1.000000  0.539886  0.351416  0.029471  0.959668
0.586860
Price          0.539886  1.000000 -0.129990  0.229903  0.524029
0.427199
Distance       0.351416 -0.129990  1.000000  0.209811  0.358071
0.156650
Postcode       0.029471  0.229903  0.209811  1.000000  0.034787
0.124291
Bedroom2       0.959668  0.524029  0.358071  0.034787  1.000000
0.580364
Bathroom       0.586860  0.427199  0.156650  0.124291  0.580364
1.000000
Car            0.476219  0.288263  0.347235  0.056521  0.477307
0.372330
Landsize       0.485742  0.327200  0.417379  0.061388  0.479792
0.212134
BuildingArea   0.775193  0.631425  0.287116  0.078542  0.761968
0.650893
YearBuilt     -0.072058 -0.368080  0.228634 -0.029741 -0.058499
0.202215
Lattitude      0.037347 -0.260322 -0.009697 -0.587603  0.039482 -
0.080748
Longtitude     0.133109  0.261787  0.311872  0.679639  0.136435
0.144319
Propertycount -0.077651 -0.011409 -0.141644  0.133281 -0.076816 -
0.035286

                   Car  Landsize  BuildingArea  YearBuilt  Lattitude
\
Rooms          0.476219  0.485742      0.775193  -0.072058   0.037347

Price          0.288263  0.327200      0.631425  -0.368080  -0.260322

Distance       0.347235  0.417379      0.287116   0.228634  -0.009697

Postcode       0.056521  0.061388      0.078542  -0.029741  -0.587603

Bedroom2       0.477307  0.479792      0.761968  -0.058499   0.039482

Bathroom       0.372330  0.212134      0.650893   0.202215  -0.080748
```

Car	1.000000	0.407369	0.472484	0.107163	0.007798
Landsize	0.407369	1.000000	0.470785	-0.128734	0.056585
BuildingArea	0.472484	0.470785	1.000000	0.003002	-0.021483
YearBuilt	0.107163	-0.128734	0.003002	1.000000	0.066924
Lattitude	0.007798	0.056585	-0.021483	0.066924	1.000000
Longtitude	0.132211	0.197717	0.141481	-0.004554	-0.355989
Propertycount	-0.043160	-0.074200	-0.081563	-0.005500	-0.031668

	Longtitude	Propertycount
Rooms	0.133109	-0.077651
Price	0.261787	-0.011409
Distance	0.311872	-0.141644
Postcode	0.679639	0.133281
Bedroom2	0.136435	-0.076816
Bathroom	0.144319	-0.035286
Car	0.132211	-0.043160
Landsize	0.197717	-0.074200
BuildingArea	0.141481	-0.081563
YearBuilt	-0.004554	-0.005500
Lattitude	-0.355989	-0.031668
Longtitude	1.000000	0.083054
Propertycount	0.083054	1.000000

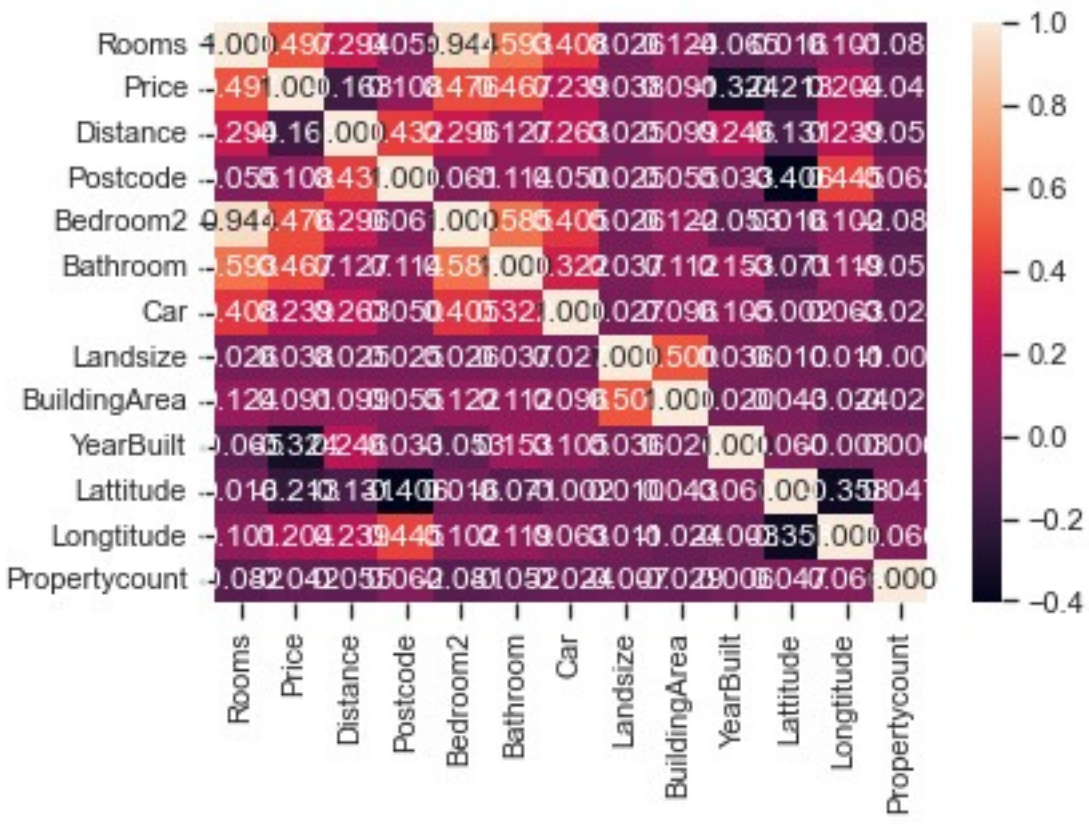
```
sns.heatmap(data.corr())
```

```
<AxesSubplot:>
```



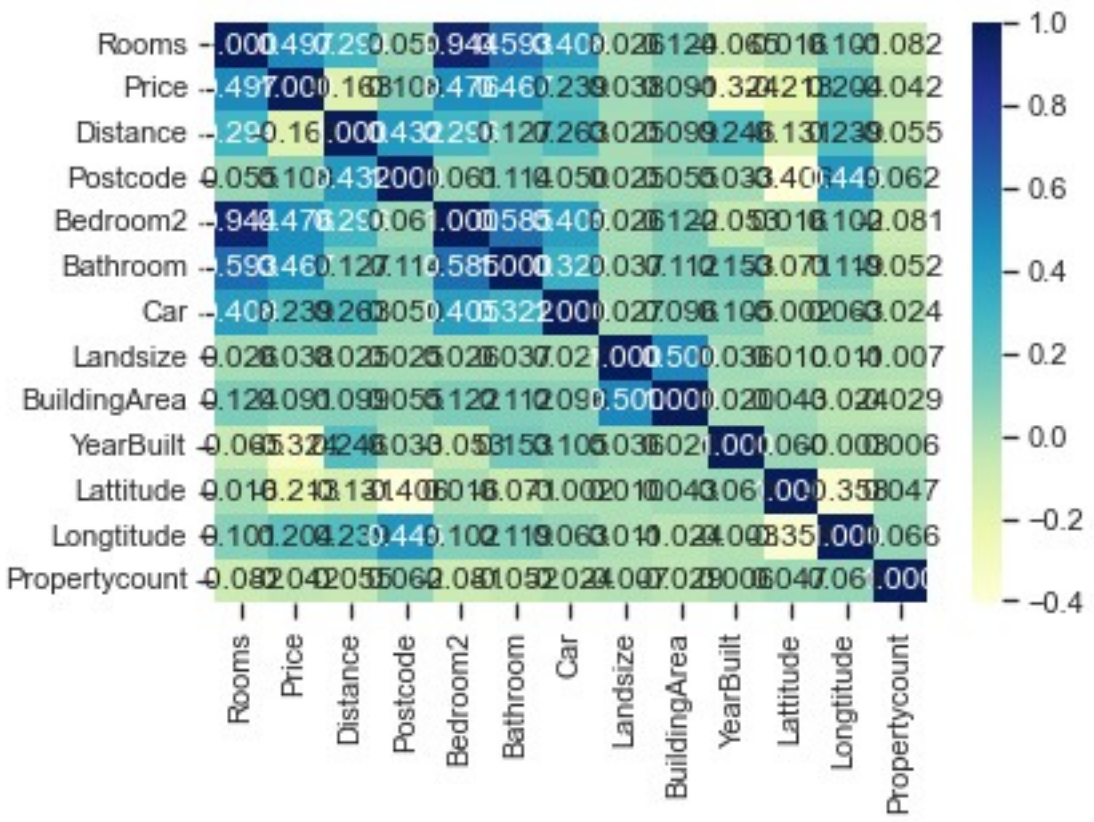

```
# Вывод значений в ячейках
sns.heatmap(data.corr(), annot=True, fmt='.3f')
```

```
<AxesSubplot:>
```



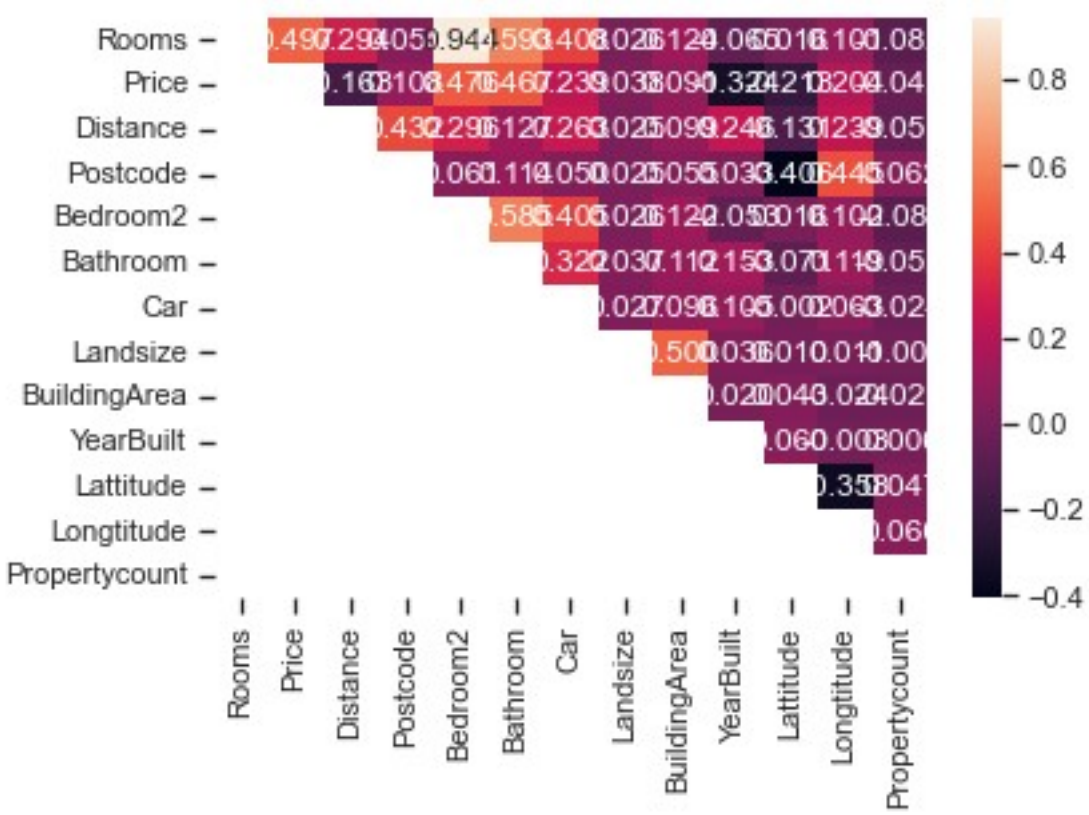

```
# Изменение цветовой гаммы
sns.heatmap(data.corr(), cmap='YlGnBu', annot=True, fmt='.3f')
```

```
<AxesSubplot:>
```

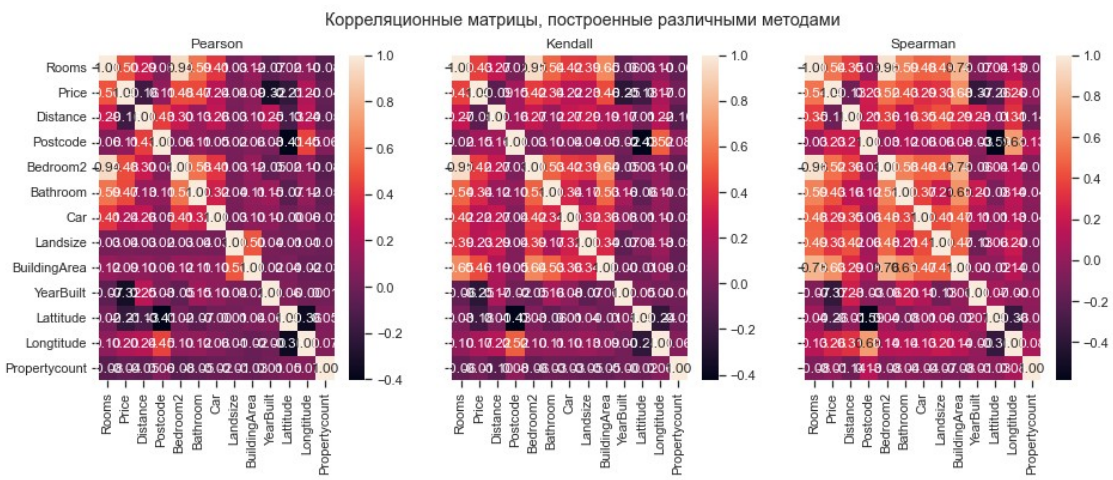



```
# Треугольный вариант матрицы
mask = np.zeros_like(data.corr(), dtype=np.bool)
# чтобы оставить нижнюю часть матрицы
# mask[np.triu_indices_from(mask)] = True
# чтобы оставить верхнюю часть матрицы
mask[np.tril_indices_from(mask)] = True
sns.heatmap(data.corr(), mask=mask, annot=True, fmt='.3f')
```

```
C:\Users\7272~1\AppData\Local\Temp/ipykernel_17288/3444845879.py:2:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin
`bool`. To silence this warning, use `bool` by itself. Doing this will
not modify any behavior and is safe. If you specifically wanted the
numpy scalar type, use `np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance:
https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  mask = np.zeros_like(data.corr(), dtype=np.bool)

<AxesSubplot:>
```




```
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row',
figsize=(15,5))
sns.heatmap(data.corr(method='pearson'), ax=ax[0], annot=True,
fmt='.2f')
sns.heatmap(data.corr(method='kendall'), ax=ax[1], annot=True,
fmt='.2f')
sns.heatmap(data.corr(method='spearman'), ax=ax[2], annot=True,
fmt='.2f')
fig.suptitle('Корреляционные матрицы, построенные различными
методами')
ax[0].title.set_text('Pearson')
ax[1].title.set_text('Kendall')
ax[2].title.set_text('Spearman')
```


Корреляционные матрицы, построенные различными методами
Pearson
Kendall
Spearman
Rooms
Price
Distance
Postcode
Bedroom2
Bathroom
Car
Landsize
BuildingArea
YearBuilt
Lattitude
Longtitude
Propertycount
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4