ASSIGNMENT – 1

**Assessing the Safety of Municipal Drinking Water**

**Name :Gurumoorthi.K**

**Register number : 922520106047**

**Naan Muthalvan Team: Analytical Data Science**

### Team ID: NM2023TMID19767

### Assignmentlink in Workwl

a

{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"provenance": []

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

}

},

"cells": [

{

"cell\_type": "code",

"execution\_count": 5,

"metadata": {

"id": "\_H0t62PMh9tN"

},

"outputs": [],

"source": [

"#Naan Mudhalvan AI Assignment 1 - Vignesh N - 922520106170 \n",

"\n",

"#Import data\n",

"\n",

"import pandas as pd\n",

"data = pd.read\_csv(\"/content/House Price India.csv\")"

]

},

{

"cell\_type": "code",

"source": [

"#Check the data: Use the head() function to check the first few rows of data and make sure it is loaded correctly.\n",

"\n",

"print(data.head())\n"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "MAKxvR81i7h3",

"outputId": "f6079cfe-5369-4ed6-c87a-65312d9cac4d"

},

"execution\_count": 8,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

" id Date number of bedrooms number of bathrooms living area \\\n",

"0 6762810145 42491 5 2.50 3650 \n",

"1 6762810635 42491 4 2.50 2920 \n",

"2 6762810998 42491 5 2.75 2910 \n",

"3 6762812605 42491 4 2.50 3310 \n",

"4 6762812919 42491 3 2.00 2710 \n",

"\n",

" lot area number of floors waterfront present number of views \\\n",

"0 9050 2.0 0 4 \n",

"1 4000 1.5 0 0 \n",

"2 9480 1.5 0 0 \n",

"3 42998 2.0 0 0 \n",

"4 4500 1.5 0 0 \n",

"\n",

" condition of the house ... Built Year Renovation Year Postal Code \\\n",

"0 5 ... 1921 0 122003 \n",

"1 5 ... 1909 0 122004 \n",

"2 3 ... 1939 0 122004 \n",

"3 3 ... 2001 0 122005 \n",

"4 4 ... 1929 0 122006 \n",

"\n",

" Lattitude Longitude living\_area\_renov lot\_area\_renov \\\n",

"0 52.8645 -114.557 2880 5400 \n",

"1 52.8878 -114.470 2470 4000 \n",

"2 52.8852 -114.468 2940 6600 \n",

"3 52.9532 -114.321 3350 42847 \n",

"4 52.9047 -114.485 2060 4500 \n",

"\n",

" Number of schools nearby Distance from the airport Price \n",

"0 2 58 2380000 \n",

"1 2 51 1400000 \n",

"2 1 53 1200000 \n",

"3 3 76 838000 \n",

"4 1 51 805000 \n",

"\n",

"[5 rows x 23 columns]\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"# Get basic information: Use the info() function to get basic information about \n",

"# the dataset, including the number of rows and columns, data type of each column,\n",

"# and the number of non-null values.\n",

"\n",

"print(data.info())\n"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "chEVqvjejqoe",

"outputId": "f32b6ff5-45b5-4478-8dc3-21adba4a6e34"

},

"execution\_count": 9,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"RangeIndex: 14620 entries, 0 to 14619\n",

"Data columns (total 23 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 id 14620 non-null int64 \n",

" 1 Date 14620 non-null int64 \n",

" 2 number of bedrooms 14620 non-null int64 \n",

" 3 number of bathrooms 14620 non-null float64\n",

" 4 living area 14620 non-null int64 \n",

" 5 lot area 14620 non-null int64 \n",

" 6 number of floors 14620 non-null float64\n",

" 7 waterfront present 14620 non-null int64 \n",

" 8 number of views 14620 non-null int64 \n",

" 9 condition of the house 14620 non-null int64 \n",

" 10 grade of the house 14620 non-null int64 \n",

" 11 Area of the house(excluding basement) 14620 non-null int64 \n",

" 12 Area of the basement 14620 non-null int64 \n",

" 13 Built Year 14620 non-null int64 \n",

" 14 Renovation Year 14620 non-null int64 \n",

" 15 Postal Code 14620 non-null int64 \n",

" 16 Lattitude 14620 non-null float64\n",

" 17 Longitude 14620 non-null float64\n",

" 18 living\_area\_renov 14620 non-null int64 \n",

" 19 lot\_area\_renov 14620 non-null int64 \n",

" 20 Number of schools nearby 14620 non-null int64 \n",

" 21 Distance from the airport 14620 non-null int64 \n",

" 22 Price 14620 non-null int64 \n",

"dtypes: float64(4), int64(19)\n",

"memory usage: 2.6 MB\n",

"None\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"# Describe the data: Use the describe() function to get descriptive statistics \n",

"# for each column, such as the mean, standard deviation, minimum, and maximum.\n",

"\n",

"print(data.describe())\n"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "-oSOgkejj4XD",

"outputId": "6cc24686-18a8-4a7d-e7a0-4cac0bf36fe0"

},

"execution\_count": 10,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

" id Date number of bedrooms number of bathrooms \\\n",

"count 1.462000e+04 14620.000000 14620.000000 14620.000000 \n",

"mean 6.762821e+09 42604.538646 3.379343 2.129583 \n",

"std 6.237575e+03 67.347991 0.938719 0.769934 \n",

"min 6.762810e+09 42491.000000 1.000000 0.500000 \n",

"25% 6.762815e+09 42546.000000 3.000000 1.750000 \n",

"50% 6.762821e+09 42600.000000 3.000000 2.250000 \n",

"75% 6.762826e+09 42662.000000 4.000000 2.500000 \n",

"max 6.762832e+09 42734.000000 33.000000 8.000000 \n",

"\n",

" living area lot area number of floors waterfront present \\\n",

"count 14620.000000 1.462000e+04 14620.000000 14620.000000 \n",

"mean 2098.262996 1.509328e+04 1.502360 0.007661 \n",

"std 928.275721 3.791962e+04 0.540239 0.087193 \n",

"min 370.000000 5.200000e+02 1.000000 0.000000 \n",

"25% 1440.000000 5.010750e+03 1.000000 0.000000 \n",

"50% 1930.000000 7.620000e+03 1.500000 0.000000 \n",

"75% 2570.000000 1.080000e+04 2.000000 0.000000 \n",

"max 13540.000000 1.074218e+06 3.500000 1.000000 \n",

"\n",

" number of views condition of the house ... Built Year \\\n",

"count 14620.000000 14620.000000 ... 14620.000000 \n",

"mean 0.233105 3.430506 ... 1970.926402 \n",

"std 0.766259 0.664151 ... 29.493625 \n",

"min 0.000000 1.000000 ... 1900.000000 \n",

"25% 0.000000 3.000000 ... 1951.000000 \n",

"50% 0.000000 3.000000 ... 1975.000000 \n",

"75% 0.000000 4.000000 ... 1997.000000 \n",

"max 4.000000 5.000000 ... 2015.000000 \n",

"\n",

" Renovation Year Postal Code Lattitude Longitude \\\n",

"count 14620.000000 14620.000000 14620.000000 14620.000000 \n",

"mean 90.924008 122033.062244 52.792848 -114.404007 \n",

"std 416.216661 19.082418 0.137522 0.141326 \n",

"min 0.000000 122003.000000 52.385900 -114.709000 \n",

"25% 0.000000 122017.000000 52.707600 -114.519000 \n",

"50% 0.000000 122032.000000 52.806400 -114.421000 \n",

"75% 0.000000 122048.000000 52.908900 -114.315000 \n",

"max 2015.000000 122072.000000 53.007600 -113.505000 \n",

"\n",

" living\_area\_renov lot\_area\_renov Number of schools nearby \\\n",

"count 14620.000000 14620.000000 14620.000000 \n",

"mean 1996.702257 12753.500068 2.012244 \n",

"std 691.093366 26058.414467 0.817284 \n",

"min 460.000000 651.000000 1.000000 \n",

"25% 1490.000000 5097.750000 1.000000 \n",

"50% 1850.000000 7620.000000 2.000000 \n",

"75% 2380.000000 10125.000000 3.000000 \n",

"max 6110.000000 560617.000000 3.000000 \n",

"\n",

" Distance from the airport Price \n",

"count 14620.000000 1.462000e+04 \n",

"mean 64.950958 5.389322e+05 \n",

"std 8.936008 3.675324e+05 \n",

"min 50.000000 7.800000e+04 \n",

"25% 57.000000 3.200000e+05 \n",

"50% 65.000000 4.500000e+05 \n",

"75% 73.000000 6.450000e+05 \n",

"max 80.000000 7.700000e+06 \n",

"\n",

"[8 rows x 23 columns]\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"# Visualize the data: Use the matplotlib or seaborn libraries to create visualizations like histograms, box plots, and\n",

"# density plots to explore the distribution of each variable.\n",

"\n",

"import matplotlib.pyplot as plt\n",

"\n",

"\n",

"# Uni-Variant Analysis\n",

"\n",

"plt.hist(data['number of bedrooms'])\n",

"plt.show()\n"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 430

},

"id": "Fbpl9Vd6kDt1",

"outputId": "970c387a-05f4-4087-9694-96c7d03f0fe6"

},

"execution\_count": 12,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 640x480 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"plt.hist(data['living area'])\n",

"plt.show()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 430

},

"id": "DhVcLwtekY6V",

"outputId": "de199b35-6d8f-47d7-bcf9-aa42210aeb65"

},

"execution\_count": 13,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 640x480 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"plt.hist(data['Price'])\n",

"plt.show()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 448

},

"id": "K1xbeOqrkigF",

"outputId": "03cbf84f-1e38-4fcf-c9ec-be3ee96c0e50"

},

"execution\_count": 14,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 640x480 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"print(data.describe())"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "g7j3Fzsql\_\_p",

"outputId": "b8ced6e9-0a98-4d5c-c722-0172bfd80d43"

},

"execution\_count": 17,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

" id Date number of bedrooms number of bathrooms \\\n",

"count 1.462000e+04 14620.000000 14620.000000 14620.000000 \n",

"mean 6.762821e+09 42604.538646 3.379343 2.129583 \n",

"std 6.237575e+03 67.347991 0.938719 0.769934 \n",

"min 6.762810e+09 42491.000000 1.000000 0.500000 \n",

"25% 6.762815e+09 42546.000000 3.000000 1.750000 \n",

"50% 6.762821e+09 42600.000000 3.000000 2.250000 \n",

"75% 6.762826e+09 42662.000000 4.000000 2.500000 \n",

"max 6.762832e+09 42734.000000 33.000000 8.000000 \n",

"\n",

" living area lot area number of floors waterfront present \\\n",

"count 14620.000000 1.462000e+04 14620.000000 14620.000000 \n",

"mean 2098.262996 1.509328e+04 1.502360 0.007661 \n",

"std 928.275721 3.791962e+04 0.540239 0.087193 \n",

"min 370.000000 5.200000e+02 1.000000 0.000000 \n",

"25% 1440.000000 5.010750e+03 1.000000 0.000000 \n",

"50% 1930.000000 7.620000e+03 1.500000 0.000000 \n",

"75% 2570.000000 1.080000e+04 2.000000 0.000000 \n",

"max 13540.000000 1.074218e+06 3.500000 1.000000 \n",

"\n",

" number of views condition of the house ... Built Year \\\n",

"count 14620.000000 14620.000000 ... 14620.000000 \n",

"mean 0.233105 3.430506 ... 1970.926402 \n",

"std 0.766259 0.664151 ... 29.493625 \n",

"min 0.000000 1.000000 ... 1900.000000 \n",

"25% 0.000000 3.000000 ... 1951.000000 \n",

"50% 0.000000 3.000000 ... 1975.000000 \n",

"75% 0.000000 4.000000 ... 1997.000000 \n",

"max 4.000000 5.000000 ... 2015.000000 \n",

"\n",

" Renovation Year Postal Code Lattitude Longitude \\\n",

"count 14620.000000 14620.000000 14620.000000 14620.000000 \n",

"mean 90.924008 122033.062244 52.792848 -114.404007 \n",

"std 416.216661 19.082418 0.137522 0.141326 \n",

"min 0.000000 122003.000000 52.385900 -114.709000 \n",

"25% 0.000000 122017.000000 52.707600 -114.519000 \n",

"50% 0.000000 122032.000000 52.806400 -114.421000 \n",

"75% 0.000000 122048.000000 52.908900 -114.315000 \n",

"max 2015.000000 122072.000000 53.007600 -113.505000 \n",

"\n",

" living\_area\_renov lot\_area\_renov Number of schools nearby \\\n",

"count 14620.000000 14620.000000 14620.000000 \n",

"mean 1996.702257 12753.500068 2.012244 \n",

"std 691.093366 26058.414467 0.817284 \n",

"min 460.000000 651.000000 1.000000 \n",

"25% 1490.000000 5097.750000 1.000000 \n",

"50% 1850.000000 7620.000000 2.000000 \n",

"75% 2380.000000 10125.000000 3.000000 \n",

"max 6110.000000 560617.000000 3.000000 \n",

"\n",

" Distance from the airport Price \n",

"count 14620.000000 1.462000e+04 \n",

"mean 64.950958 5.389322e+05 \n",

"std 8.936008 3.675324e+05 \n",

"min 50.000000 7.800000e+04 \n",

"25% 57.000000 3.200000e+05 \n",

"50% 65.000000 4.500000e+05 \n",

"75% 73.000000 6.450000e+05 \n",

"max 80.000000 7.700000e+06 \n",

"\n",

"[8 rows x 23 columns]\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"# Bi- Variant Analysis\n",

"\n",

"import matplotlib.pyplot as plt\n",

"\n",

"plt.scatter(data['condition of the house'], data['Built Year'])\n",

"plt.show()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 433

},

"id": "kRVOLgZrlTNc",

"outputId": "297a84f9-d2b0-436c-dbf4-7869c8e10735"

},

"execution\_count": 19,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 640x480 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"plt.scatter(data['number of bedrooms'], data['living area'])\n",

"plt.show()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 430

},

"id": "yvxK-r4EmSGh",

"outputId": "2ddf0f71-a03c-45e6-9031-1acb565499be"

},

"execution\_count": 20,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 640x480 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"plt.scatter(data['number of bedrooms'], data['Price'])\n",

"plt.show()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 445

},

"id": "v\_dEaYRumiiD",

"outputId": "a9a7ab51-955c-4b47-830e-1901213cce8b"

},

"execution\_count": 21,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 640x480 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"# Multi-Variate Analysis\n",

"import pandas as pd\n",

"import seaborn as sns\n",

"import matplotlib.pyplot as plt\n",

"\n",

"data = pd.read\_csv('/content/House Price India.csv', nrows=10)\n",

"sns.heatmap(data.corr(), annot=True, cmap='coolwarm')\n",

"plt.show()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 696

},

"id": "8-iYH12Amt4Q",

"outputId": "fcc20331-5f9b-4b44-82e7-c7d593ae43c9"

},

"execution\_count": 4,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 640x480 with 2 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"import pandas as pd\n",

"import seaborn as sns\n",

"import matplotlib.pyplot as plt\n",

"\n",

"# Load the dataset\n",

"data = pd.read\_csv('/content/House Price India.csv')\n",

"\n",

"#\n",

"# Compute the correlation matrix\n",

"corr = data.corr()\n",

"\n",

"# Visualize the correlation matrix as a heatmap\n",

"sns.heatmap(corr, annot=True, cmap='coolwarm')\n",

"plt.show()\n",

"\n",

"# Cluster the variables based on the correlation matrix\n",

"sns.clustermap(corr, cmap='coolwarm')\n",

"plt.show()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 1000

},

"id": "zCJQBjc1qI-1",

"outputId": "1ed7cab8-b8a8-4ff1-95db-e388c04707c5"

},

"execution\_count": 5,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 640x480 with 2 Axes>"

],

"image/png": "\n"

},

"metadata": {}

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1000x1000 with 4 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"import pandas as pd\n",

"import seaborn as sns\n",

"import matplotlib.pyplot as plt\n",

"\n",

"# Load the dataset\n",

"data = pd.read\_csv('/content/House Price India.csv',nrows = 10)\n",

"\n",

"# Create a scatter matrix\n",

"pd.plotting.scatter\_matrix(data, alpha=0.2, figsize=(10,10))\n",

"plt.show()\n",

"\n",

"# Compute the correlation matrix\n",

"corr = data.corr()\n",

"\n",

"# Visualize the correlation matrix as a heatmap\n",

"sns.heatmap(corr, annot=True, cmap='coolwarm')\n",

"plt.show()\n",

"\n",

"# Create a pairplot\n",

"sns.pairplot(data, hue='Price')\n",

"plt.show()\n"

],

"metadata": {

"id": "X5Rwt6qGsTYT"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"import pandas as pd\n",

"import seaborn as sns\n",

"import matplotlib.pyplot as plt\n",

"\n",

"# Load the dataset\n",

"data = pd.read\_csv('/content/House Price India.csv',nrows = 10)\n",

"\n",

"# Create a scatter matrix\n",

"pd.plotting.scatter\_matrix(data, alpha=0.2, figsize=(10,10))\n",

"plt.show()\n",

"\n",

"# Compute the correlation matrix\n",

"corr = data.corr()\n",

"\n",

"# Visualize the correlation matrix as a heatmap\n",

"sns.heatmap(corr, annot=True, cmap='coolwarm')\n",

"plt.show()\n",

"\n",

"# Create a pairplot\n",

"sns.pairplot(data, hue='Price')\n",

"plt.show()"

],

"metadata": {

"id": "YzXDrxwytC5d"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"import pandas as pd\n",

"\n",

"# Load the dataset\n",

"data = pd.read\_csv('/content/House Price India.csv')\n",

"\n",

"# Display the summary statistics\n",

"print(data.describe())\n",

"\n",

"# Display the column-wise data types\n",

"print(data.dtypes)\n",

"\n",

"# Display the number of unique values in each column\n",

"print(data.nunique())"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "yafFKIRFtybx",

"outputId": "39cd8637-257e-4518-82f2-d273330fe408"

},

"execution\_count": 9,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

" id Date number of bedrooms number of bathrooms \\\n",

"count 1.462000e+04 14620.000000 14620.000000 14620.000000 \n",

"mean 6.762821e+09 42604.538646 3.379343 2.129583 \n",

"std 6.237575e+03 67.347991 0.938719 0.769934 \n",

"min 6.762810e+09 42491.000000 1.000000 0.500000 \n",

"25% 6.762815e+09 42546.000000 3.000000 1.750000 \n",

"50% 6.762821e+09 42600.000000 3.000000 2.250000 \n",

"75% 6.762826e+09 42662.000000 4.000000 2.500000 \n",

"max 6.762832e+09 42734.000000 33.000000 8.000000 \n",

"\n",

" living area lot area number of floors waterfront present \\\n",

"count 14620.000000 1.462000e+04 14620.000000 14620.000000 \n",

"mean 2098.262996 1.509328e+04 1.502360 0.007661 \n",

"std 928.275721 3.791962e+04 0.540239 0.087193 \n",

"min 370.000000 5.200000e+02 1.000000 0.000000 \n",

"25% 1440.000000 5.010750e+03 1.000000 0.000000 \n",

"50% 1930.000000 7.620000e+03 1.500000 0.000000 \n",

"75% 2570.000000 1.080000e+04 2.000000 0.000000 \n",

"max 13540.000000 1.074218e+06 3.500000 1.000000 \n",

"\n",

" number of views condition of the house ... Built Year \\\n",

"count 14620.000000 14620.000000 ... 14620.000000 \n",

"mean 0.233105 3.430506 ... 1970.926402 \n",

"std 0.766259 0.664151 ... 29.493625 \n",

"min 0.000000 1.000000 ... 1900.000000 \n",

"25% 0.000000 3.000000 ... 1951.000000 \n",

"50% 0.000000 3.000000 ... 1975.000000 \n",

"75% 0.000000 4.000000 ... 1997.000000 \n",

"max 4.000000 5.000000 ... 2015.000000 \n",

"\n",

" Renovation Year Postal Code Lattitude Longitude \\\n",

"count 14620.000000 14620.000000 14620.000000 14620.000000 \n",

"mean 90.924008 122033.062244 52.792848 -114.404007 \n",

"std 416.216661 19.082418 0.137522 0.141326 \n",

"min 0.000000 122003.000000 52.385900 -114.709000 \n",

"25% 0.000000 122017.000000 52.707600 -114.519000 \n",

"50% 0.000000 122032.000000 52.806400 -114.421000 \n",

"75% 0.000000 122048.000000 52.908900 -114.315000 \n",

"max 2015.000000 122072.000000 53.007600 -113.505000 \n",

"\n",

" living\_area\_renov lot\_area\_renov Number of schools nearby \\\n",

"count 14620.000000 14620.000000 14620.000000 \n",

"mean 1996.702257 12753.500068 2.012244 \n",

"std 691.093366 26058.414467 0.817284 \n",

"min 460.000000 651.000000 1.000000 \n",

"25% 1490.000000 5097.750000 1.000000 \n",

"50% 1850.000000 7620.000000 2.000000 \n",

"75% 2380.000000 10125.000000 3.000000 \n",

"max 6110.000000 560617.000000 3.000000 \n",

"\n",

" Distance from the airport Price \n",

"count 14620.000000 1.462000e+04 \n",

"mean 64.950958 5.389322e+05 \n",

"std 8.936008 3.675324e+05 \n",

"min 50.000000 7.800000e+04 \n",

"25% 57.000000 3.200000e+05 \n",

"50% 65.000000 4.500000e+05 \n",

"75% 73.000000 6.450000e+05 \n",

"max 80.000000 7.700000e+06 \n",

"\n",

"[8 rows x 23 columns]\n",

"id int64\n",

"Date int64\n",

"number of bedrooms int64\n",

"number of bathrooms float64\n",

"living area int64\n",

"lot area int64\n",

"number of floors float64\n",

"waterfront present int64\n",

"number of views int64\n",

"condition of the house int64\n",

"grade of the house int64\n",

"Area of the house(excluding basement) int64\n",

"Area of the basement int64\n",

"Built Year int64\n",

"Renovation Year int64\n",

"Postal Code int64\n",

"Lattitude float64\n",

"Longitude float64\n",

"living\_area\_renov int64\n",

"lot\_area\_renov int64\n",

"Number of schools nearby int64\n",

"Distance from the airport int64\n",

"Price int64\n",

"dtype: object\n",

"id 14620\n",

"Date 241\n",

"number of bedrooms 12\n",

"number of bathrooms 29\n",

"living area 865\n",

"lot area 7451\n",

"number of floors 6\n",

"waterfront present 2\n",

"number of views 5\n",

"condition of the house 5\n",

"grade of the house 10\n",

"Area of the house(excluding basement) 781\n",

"Area of the basement 280\n",

"Built Year 116\n",

"Renovation Year 68\n",

"Postal Code 70\n",

"Lattitude 4662\n",

"Longitude 716\n",

"living\_area\_renov 665\n",

"lot\_area\_renov 6835\n",

"Number of schools nearby 3\n",

"Distance from the airport 31\n",

"Price 2901\n",

"dtype: int64\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"# Drop rows with missing values\n",

"data.dropna(inplace=True)\n",

"\n",

"# Drop columns with missing values\n",

"data.dropna(axis=1, inplace=True)\n"

],

"metadata": {

"id": "zSELGVXqu6H7"

},

"execution\_count": 10,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"# Replace missing values with the mean of the column\n",

"data.fillna(data.mean(), inplace=True)\n",

"\n",

"# Replace missing values with the median of the column\n",

"data.fillna(data.median(), inplace=True)\n",

"\n",

"# Replace missing values with the mode of the column\n",

"data.fillna(data.mode(), inplace=True)\n",

"\n",

"# Replace missing values with a custom value\n",

"data.fillna('Unknown', inplace=True)\n"

],

"metadata": {

"id": "dJQCLqiWu-Q9"

},

"execution\_count": 11,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"import pandas as pd\n",

"import matplotlib.pyplot as plt\n",

"\n",

"# Load the dataset\n",

"data = pd.read\_csv('/content/House Price India.csv',nrows=10)\n",

"\n",

"# Create a scatter plot matrix to visualize the relationship between pairs of variables\n",

"pd.plotting.scatter\_matrix(data, figsize=(12, 12), diagonal='hist')\n",

"\n",

"# Create a heatmap to visualize the correlation between variables\n",

"corr\_matrix = data.corr()\n",

"plt.imshow(corr\_matrix, cmap='coolwarm', interpolation='none')\n",

"plt.colorbar()\n",

"plt.xticks(range(len(corr\_matrix)), corr\_matrix.columns, rotation=90)\n",

"plt.yticks(range(len(corr\_matrix)), corr\_matrix.columns)\n",

"plt.show()\n"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 1000

},

"id": "dOHgIn6ivM7s",

"outputId": "3f16c8b5-08d0-496c-c382-79b836d49968"

},

"execution\_count": 13,

"outputs": [

{

"output\_type": "stream",

"name": "stderr",

"text": [

"/usr/local/lib/python3.10/dist-packages/pandas/plotting/\_matplotlib/misc.py:101: UserWarning: Attempting to set identical low and high xlims makes transformation singular; automatically expanding.\n",

" ax.set\_xlim(boundaries\_list[j])\n",

"/usr/local/lib/python3.10/dist-packages/pandas/plotting/\_matplotlib/misc.py:102: UserWarning: Attempting to set identical low and high ylims makes transformation singular; automatically expanding.\n",

" ax.set\_ylim(boundaries\_list[i])\n",

"/usr/local/lib/python3.10/dist-packages/pandas/plotting/\_matplotlib/misc.py:92: UserWarning: Attempting to set identical low and high xlims makes transformation singular; automatically expanding.\n",

" ax.set\_xlim(boundaries\_list[i])\n"

]

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 1200x1200 with 530 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"metadata": {

"id": "061f72c5"

},

"outputs": [],

"source": [

"import pandas as pd\n",

"import matplotlib.pyplot as plt\n",

"import numpy as np\n",

"import seaborn as sns\n",

"from scipy import stats"

]

},

{

"cell\_type": "code",

"execution\_count": 16,

"metadata": {

"id": "99d210dd"

},

"outputs": [],

"source": [

"df = pd.read\_csv(\"/content/House Price India.csv\")"

]

},

{

"cell\_type": "code",

"source": [

"df.head()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 352

},

"id": "YoIOLzGi4WIU",

"outputId": "9470af74-22a8-4cc1-d152-8c88c704036d"

},

"execution\_count": 17,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" id Date number of bedrooms number of bathrooms living area \\\n",

"0 6762810145 42491 5 2.50 3650 \n",

"1 6762810635 42491 4 2.50 2920 \n",

"2 6762810998 42491 5 2.75 2910 \n",

"3 6762812605 42491 4 2.50 3310 \n",

"4 6762812919 42491 3 2.00 2710 \n",

"\n",

" lot area number of floors waterfront present number of views \\\n",

"0 9050 2.0 0 4 \n",

"1 4000 1.5 0 0 \n",

"2 9480 1.5 0 0 \n",

"3 42998 2.0 0 0 \n",

"4 4500 1.5 0 0 \n",

"\n",

" condition of the house ... Built Year Renovation Year Postal Code \\\n",

"0 5 ... 1921 0 122003 \n",

"1 5 ... 1909 0 122004 \n",

"2 3 ... 1939 0 122004 \n",

"3 3 ... 2001 0 122005 \n",

"4 4 ... 1929 0 122006 \n",

"\n",

" Lattitude Longitude living\_area\_renov lot\_area\_renov \\\n",

"0 52.8645 -114.557 2880 5400 \n",

"1 52.8878 -114.470 2470 4000 \n",

"2 52.8852 -114.468 2940 6600 \n",

"3 52.9532 -114.321 3350 42847 \n",

"4 52.9047 -114.485 2060 4500 \n",

"\n",

" Number of schools nearby Distance from the airport Price \n",

"0 2 58 2380000 \n",

"1 2 51 1400000 \n",

"2 1 53 1200000 \n",

"3 3 76 838000 \n",

"4 1 51 805000 \n",

"\n",

"[5 rows x 23 columns]"

],

"text/html": [

"\n",

" <div id=\"df-7800adef-924e-47b8-9aef-de84deec7df0\">\n",

" <div class=\"colab-df-container\">\n",

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>id</th>\n",

" <th>Date</th>\n",

" <th>number of bedrooms</th>\n",

" <th>number of bathrooms</th>\n",

" <th>living area</th>\n",

" <th>lot area</th>\n",

" <th>number of floors</th>\n",

" <th>waterfront present</th>\n",

" <th>number of views</th>\n",

" <th>condition of the house</th>\n",

" <th>...</th>\n",

" <th>Built Year</th>\n",

" <th>Renovation Year</th>\n",

" <th>Postal Code</th>\n",

" <th>Lattitude</th>\n",

" <th>Longitude</th>\n",

" <th>living\_area\_renov</th>\n",

" <th>lot\_area\_renov</th>\n",

" <th>Number of schools nearby</th>\n",

" <th>Distance from the airport</th>\n",

" <th>Price</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>6762810145</td>\n",

" <td>42491</td>\n",

" <td>5</td>\n",

" <td>2.50</td>\n",

" <td>3650</td>\n",

" <td>9050</td>\n",

" <td>2.0</td>\n",

" <td>0</td>\n",

" <td>4</td>\n",

" <td>5</td>\n",

" <td>...</td>\n",

" <td>1921</td>\n",

" <td>0</td>\n",

" <td>122003</td>\n",

" <td>52.8645</td>\n",

" <td>-114.557</td>\n",

" <td>2880</td>\n",

" <td>5400</td>\n",

" <td>2</td>\n",

" <td>58</td>\n",

" <td>2380000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>6762810635</td>\n",

" <td>42491</td>\n",

" <td>4</td>\n",

" <td>2.50</td>\n",

" <td>2920</td>\n",

" <td>4000</td>\n",

" <td>1.5</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>5</td>\n",

" <td>...</td>\n",

" <td>1909</td>\n",

" <td>0</td>\n",

" <td>122004</td>\n",

" <td>52.8878</td>\n",

" <td>-114.470</td>\n",

" <td>2470</td>\n",

" <td>4000</td>\n",

" <td>2</td>\n",

" <td>51</td>\n",

" <td>1400000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>6762810998</td>\n",

" <td>42491</td>\n",

" <td>5</td>\n",

" <td>2.75</td>\n",

" <td>2910</td>\n",

" <td>9480</td>\n",

" <td>1.5</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>3</td>\n",

" <td>...</td>\n",

" <td>1939</td>\n",

" <td>0</td>\n",

" <td>122004</td>\n",

" <td>52.8852</td>\n",

" <td>-114.468</td>\n",

" <td>2940</td>\n",

" <td>6600</td>\n",

" <td>1</td>\n",

" <td>53</td>\n",

" <td>1200000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>6762812605</td>\n",

" <td>42491</td>\n",

" <td>4</td>\n",

" <td>2.50</td>\n",

" <td>3310</td>\n",

" <td>42998</td>\n",

" <td>2.0</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>3</td>\n",

" <td>...</td>\n",

" <td>2001</td>\n",

" <td>0</td>\n",

" <td>122005</td>\n",

" <td>52.9532</td>\n",

" <td>-114.321</td>\n",

" <td>3350</td>\n",

" <td>42847</td>\n",

" <td>3</td>\n",

" <td>76</td>\n",

" <td>838000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>6762812919</td>\n",

" <td>42491</td>\n",

" <td>3</td>\n",

" <td>2.00</td>\n",

" <td>2710</td>\n",

" <td>4500</td>\n",

" <td>1.5</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" <td>4</td>\n",

" <td>...</td>\n",

" <td>1929</td>\n",

" <td>0</td>\n",

" <td>122006</td>\n",

" <td>52.9047</td>\n",

" <td>-114.485</td>\n",

" <td>2060</td>\n",

" <td>4500</td>\n",

" <td>1</td>\n",

" <td>51</td>\n",

" <td>805000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>5 rows × 23 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-7800adef-924e-47b8-9aef-de84deec7df0')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-7800adef-924e-47b8-9aef-de84deec7df0 button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-7800adef-924e-47b8-9aef-de84deec7df0');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 17

}

]

},

{

"cell\_type": "code",

"source": [

"df.shape"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "Hm27BCGW4ZVL",

"outputId": "87cdd596-a8a0-4bfc-8e44-3df3bcb2d97b"

},

"execution\_count": 18,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"(14620, 23)"

]

},

"metadata": {},

"execution\_count": 18

}

]

},

{

"cell\_type": "code",

"source": [

"df.info()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "gZt9WVuc4YMp",

"outputId": "82241228-f506-4f4e-8966-077c36324b67"

},

"execution\_count": 19,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"RangeIndex: 14620 entries, 0 to 14619\n",

"Data columns (total 23 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 id 14620 non-null int64 \n",

" 1 Date 14620 non-null int64 \n",

" 2 number of bedrooms 14620 non-null int64 \n",

" 3 number of bathrooms 14620 non-null float64\n",

" 4 living area 14620 non-null int64 \n",

" 5 lot area 14620 non-null int64 \n",

" 6 number of floors 14620 non-null float64\n",

" 7 waterfront present 14620 non-null int64 \n",

" 8 number of views 14620 non-null int64 \n",

" 9 condition of the house 14620 non-null int64 \n",

" 10 grade of the house 14620 non-null int64 \n",

" 11 Area of the house(excluding basement) 14620 non-null int64 \n",

" 12 Area of the basement 14620 non-null int64 \n",

" 13 Built Year 14620 non-null int64 \n",

" 14 Renovation Year 14620 non-null int64 \n",

" 15 Postal Code 14620 non-null int64 \n",

" 16 Lattitude 14620 non-null float64\n",

" 17 Longitude 14620 non-null float64\n",

" 18 living\_area\_renov 14620 non-null int64 \n",

" 19 lot\_area\_renov 14620 non-null int64 \n",

" 20 Number of schools nearby 14620 non-null int64 \n",

" 21 Distance from the airport 14620 non-null int64 \n",

" 22 Price 14620 non-null int64 \n",

"dtypes: float64(4), int64(19)\n",

"memory usage: 2.6 MB\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"df.isna().sum()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "O-8TKQBk4dmB",

"outputId": "58dc39b0-1002-4a97-8544-a408f118691b"

},

"execution\_count": 20,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"id 0\n",

"Date 0\n",

"number of bedrooms 0\n",

"number of bathrooms 0\n",

"living area 0\n",

"lot area 0\n",

"number of floors 0\n",

"waterfront present 0\n",

"number of views 0\n",

"condition of the house 0\n",

"grade of the house 0\n",

"Area of the house(excluding basement) 0\n",

"Area of the basement 0\n",

"Built Year 0\n",

"Renovation Year 0\n",

"Postal Code 0\n",

"Lattitude 0\n",

"Longitude 0\n",

"living\_area\_renov 0\n",

"lot\_area\_renov 0\n",

"Number of schools nearby 0\n",

"Distance from the airport 0\n",

"Price 0\n",

"dtype: int64"

]

},

"metadata": {},

"execution\_count": 20

}

]

},

{

"cell\_type": "code",

"source": [

"df[\"number of floors\"]"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "U3qgL0we4je5",

"outputId": "c068ebdc-8833-45ea-80fd-5c8c26f90518"

},

"execution\_count": 21,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"0 2.0\n",

"1 1.5\n",

"2 1.5\n",

"3 2.0\n",

"4 1.5\n",

" ... \n",

"14615 1.0\n",

"14616 1.5\n",

"14617 1.0\n",

"14618 1.0\n",

"14619 1.0\n",

"Name: number of floors, Length: 14620, dtype: float64"

]

},

"metadata": {},

"execution\_count": 21

}

]

},

{

"cell\_type": "code",

"source": [

"df[\"number of floors\"].unique()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "zIvpHESN4vDp",

"outputId": "5144c4db-5914-463b-843f-38f435a5930c"

},

"execution\_count": 23,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"array([2. , 1.5, 1. , 2.5, 3. , 3.5])"

]

},

"metadata": {},

"execution\_count": 23

}

]

},

{

"cell\_type": "code",

"source": [

"sns.displot(df[\"Price\"])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 523

},

"id": "Xj\_b7O7d400B",

"outputId": "8288ff15-2445-4862-d1b5-fedf7a64a638"

},

"execution\_count": 25,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<seaborn.axisgrid.FacetGrid at 0x7f0a6bb21bd0>"

]

},

"metadata": {},

"execution\_count": 25

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 500x500 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"sns.barplot(data = df,x = \"number of bedrooms\",y = \"Price\")"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 482

},

"id": "JE4HbHrA5CYR",

"outputId": "a706b7cf-5689-410b-e4db-273ce698e51f"

},

"execution\_count": 27,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<Axes: xlabel='number of bedrooms', ylabel='Price'>"

]

},

"metadata": {},

"execution\_count": 27

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 640x480 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"execution\_count": 28,

"metadata": {

"id": "b902c9db",

"outputId": "24036e0f-302f-4903-9117-284c90b144fc",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 482

}

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<Axes: xlabel='number of floors', ylabel='Price'>"

]

},

"metadata": {},

"execution\_count": 28

},

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 640x480 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

],

"source": [

"sns.barplot(data=df,x='number of floors',y='Price')"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "3f646c3d",

"outputId": "a6ef5855-819c-448c-9ba3-63605a3723cc"

},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:>"

]

},

"execution\_count": 19,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 640x480 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"sns.boxplot(data=df['living area'])"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"id": "7ac1c622",

"outputId": "6574cec2-3922-44b5-a3fb-1cb02a9b6733"

},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:>"

]

},

"execution\_count": 16,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 640x480 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"sns.boxplot(data=df['number of bedrooms'])"

]

},

{

"cell\_type": "code",

"execution\_count": 29,

"metadata": {

"id": "e2a7a8ab",

"outputId": "6269d54e-f8f7-47e3-fb3b-c6813ecd9e6b",

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 69

}

},

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

" id number of bedrooms number of bathrooms living area \\\n",

"0 1.713148 1.726515 0.481119 1.671691 \n",

"1 1.634590 0.661197 0.481119 0.885260 \n",

"2 1.576392 1.726515 0.805833 0.874487 \n",

"3 1.318751 0.661197 0.481119 1.305408 \n",

"4 1.268409 0.404121 0.168309 0.659026 \n",

"... ... ... ... ... \n",

"14615 1.510170 1.469439 0.817737 0.584182 \n",

"14616 1.524439 0.404121 0.168309 0.450596 \n",

"14617 1.569169 1.469439 1.467166 1.107751 \n",

"14618 1.583759 0.661197 1.467166 1.150843 \n",

"14619 1.704643 0.404121 1.467166 1.290892 \n",

"\n",

" lot area number of floors waterfront present number of views \\\n",

"0 0.159376 0.921180 0.087863 4.916126 \n",

"1 0.292557 0.004368 0.087863 0.304223 \n",

"2 0.148036 0.004368 0.087863 0.304223 \n",

"3 0.735916 0.921180 0.087863 0.304223 \n",

"4 0.279371 0.004368 0.087863 0.304223 \n",

"... ... ... ... ... \n",

"14615 0.129402 0.929917 0.087863 0.304223 \n",

"14616 0.213440 0.004368 0.087863 0.304223 \n",

"14617 0.236648 0.929917 0.087863 0.304223 \n",

"14618 0.223435 0.929917 0.087863 0.304223 \n",

"14619 0.272250 0.929917 0.087863 0.304223 \n",

"\n",

" condition of the house grade of the house ... Built Year \\\n",

"0 2.363238 1.972420 ... 1.692844 \n",

"1 2.363238 0.270281 ... 2.099726 \n",

"2 0.648227 0.270281 ... 1.082522 \n",

"3 0.648227 1.121351 ... 1.019699 \n",

"4 0.857506 0.270281 ... 1.421590 \n",

"... ... ... ... ... \n",

"14615 0.857506 0.580788 ... 0.472200 \n",

"14616 0.857506 0.580788 ... 0.099225 \n",

"14617 0.648227 1.431857 ... 0.302666 \n",

"14618 0.857506 1.431857 ... 0.540013 \n",

"14619 0.648227 1.431857 ... 0.065318 \n",

"\n",

" Renovation Year Postal Code Lattitude Longitude living\_area\_renov \\\n",

"0 0.218461 1.575444 0.521043 1.082593 1.278160 \n",

"1 0.218461 1.523037 0.690476 0.466974 0.684877 \n",

"2 0.218461 1.523037 0.671569 0.452822 1.364982 \n",

"3 0.218461 1.470631 1.166052 0.587363 1.958265 \n",

"4 0.218461 1.418225 0.813370 0.573115 0.091594 \n",

"... ... ... ... ... ... \n",

"14615 0.218461 1.726138 1.263460 0.481126 0.366530 \n",

"14616 0.218461 2.040574 2.074994 0.077885 0.660863 \n",

"14617 0.218461 1.202077 0.465015 0.728789 1.254146 \n",

"14618 0.218461 0.468393 0.561003 0.049485 0.834507 \n",

"14619 4.608517 0.789353 1.883745 1.047213 1.586963 \n",

"\n",

" lot\_area\_renov Number of schools nearby Distance from the airport \\\n",

"0 0.282203 0.014981 0.777886 \n",

"1 0.335930 0.014981 1.561260 \n",

"2 0.236151 1.238588 1.337439 \n",

"3 1.154887 1.208625 1.236505 \n",

"4 0.316742 1.238588 1.561260 \n",

"... ... ... ... \n",

"14615 0.173942 1.208625 1.236505 \n",

"14616 0.202379 1.208625 0.665975 \n",

"14617 0.254571 0.014981 0.106422 \n",

"14618 0.234961 1.208625 1.225528 \n",

"14619 0.355886 0.014981 1.113618 \n",

"\n",

" Price \n",

"0 5.009438 \n",

"1 2.342915 \n",

"2 1.798727 \n",

"3 0.813746 \n",

"4 0.723955 \n",

"... ... \n",

"14615 0.863170 \n",

"14616 0.869973 \n",

"14617 0.897726 \n",

"14618 0.908610 \n",

"14619 1.069146 \n",

"\n",

"[14620 rows x 22 columns]"

],

"text/html": [

"\n",

" <div id=\"df-7732e605-cc9c-4d20-989f-a6d4ad3328df\">\n",

" <div class=\"colab-df-container\">\n",

" <div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>id</th>\n",

" <th>number of bedrooms</th>\n",

" <th>number of bathrooms</th>\n",

" <th>living area</th>\n",

" <th>lot area</th>\n",

" <th>number of floors</th>\n",

" <th>waterfront present</th>\n",

" <th>number of views</th>\n",

" <th>condition of the house</th>\n",

" <th>grade of the house</th>\n",

" <th>...</th>\n",

" <th>Built Year</th>\n",

" <th>Renovation Year</th>\n",

" <th>Postal Code</th>\n",

" <th>Lattitude</th>\n",

" <th>Longitude</th>\n",

" <th>living\_area\_renov</th>\n",

" <th>lot\_area\_renov</th>\n",

" <th>Number of schools nearby</th>\n",

" <th>Distance from the airport</th>\n",

" <th>Price</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>1.713148</td>\n",

" <td>1.726515</td>\n",

" <td>0.481119</td>\n",

" <td>1.671691</td>\n",

" <td>0.159376</td>\n",

" <td>0.921180</td>\n",

" <td>0.087863</td>\n",

" <td>4.916126</td>\n",

" <td>2.363238</td>\n",

" <td>1.972420</td>\n",

" <td>...</td>\n",

" <td>1.692844</td>\n",

" <td>0.218461</td>\n",

" <td>1.575444</td>\n",

" <td>0.521043</td>\n",

" <td>1.082593</td>\n",

" <td>1.278160</td>\n",

" <td>0.282203</td>\n",

" <td>0.014981</td>\n",

" <td>0.777886</td>\n",

" <td>5.009438</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>1.634590</td>\n",

" <td>0.661197</td>\n",

" <td>0.481119</td>\n",

" <td>0.885260</td>\n",

" <td>0.292557</td>\n",

" <td>0.004368</td>\n",

" <td>0.087863</td>\n",

" <td>0.304223</td>\n",

" <td>2.363238</td>\n",

" <td>0.270281</td>\n",

" <td>...</td>\n",

" <td>2.099726</td>\n",

" <td>0.218461</td>\n",

" <td>1.523037</td>\n",

" <td>0.690476</td>\n",

" <td>0.466974</td>\n",

" <td>0.684877</td>\n",

" <td>0.335930</td>\n",

" <td>0.014981</td>\n",

" <td>1.561260</td>\n",

" <td>2.342915</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>1.576392</td>\n",

" <td>1.726515</td>\n",

" <td>0.805833</td>\n",

" <td>0.874487</td>\n",

" <td>0.148036</td>\n",

" <td>0.004368</td>\n",

" <td>0.087863</td>\n",

" <td>0.304223</td>\n",

" <td>0.648227</td>\n",

" <td>0.270281</td>\n",

" <td>...</td>\n",

" <td>1.082522</td>\n",

" <td>0.218461</td>\n",

" <td>1.523037</td>\n",

" <td>0.671569</td>\n",

" <td>0.452822</td>\n",

" <td>1.364982</td>\n",

" <td>0.236151</td>\n",

" <td>1.238588</td>\n",

" <td>1.337439</td>\n",

" <td>1.798727</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>1.318751</td>\n",

" <td>0.661197</td>\n",

" <td>0.481119</td>\n",

" <td>1.305408</td>\n",

" <td>0.735916</td>\n",

" <td>0.921180</td>\n",

" <td>0.087863</td>\n",

" <td>0.304223</td>\n",

" <td>0.648227</td>\n",

" <td>1.121351</td>\n",

" <td>...</td>\n",

" <td>1.019699</td>\n",

" <td>0.218461</td>\n",

" <td>1.470631</td>\n",

" <td>1.166052</td>\n",

" <td>0.587363</td>\n",

" <td>1.958265</td>\n",

" <td>1.154887</td>\n",

" <td>1.208625</td>\n",

" <td>1.236505</td>\n",

" <td>0.813746</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>1.268409</td>\n",

" <td>0.404121</td>\n",

" <td>0.168309</td>\n",

" <td>0.659026</td>\n",

" <td>0.279371</td>\n",

" <td>0.004368</td>\n",

" <td>0.087863</td>\n",

" <td>0.304223</td>\n",

" <td>0.857506</td>\n",

" <td>0.270281</td>\n",

" <td>...</td>\n",

" <td>1.421590</td>\n",

" <td>0.218461</td>\n",

" <td>1.418225</td>\n",

" <td>0.813370</td>\n",

" <td>0.573115</td>\n",

" <td>0.091594</td>\n",

" <td>0.316742</td>\n",

" <td>1.238588</td>\n",

" <td>1.561260</td>\n",

" <td>0.723955</td>\n",

" </tr>\n",

" <tr>\n",

" <th>...</th>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>14615</th>\n",

" <td>1.510170</td>\n",

" <td>1.469439</td>\n",

" <td>0.817737</td>\n",

" <td>0.584182</td>\n",

" <td>0.129402</td>\n",

" <td>0.929917</td>\n",

" <td>0.087863</td>\n",

" <td>0.304223</td>\n",

" <td>0.857506</td>\n",

" <td>0.580788</td>\n",

" <td>...</td>\n",

" <td>0.472200</td>\n",

" <td>0.218461</td>\n",

" <td>1.726138</td>\n",

" <td>1.263460</td>\n",

" <td>0.481126</td>\n",

" <td>0.366530</td>\n",

" <td>0.173942</td>\n",

" <td>1.208625</td>\n",

" <td>1.236505</td>\n",

" <td>0.863170</td>\n",

" </tr>\n",

" <tr>\n",

" <th>14616</th>\n",

" <td>1.524439</td>\n",

" <td>0.404121</td>\n",

" <td>0.168309</td>\n",

" <td>0.450596</td>\n",

" <td>0.213440</td>\n",

" <td>0.004368</td>\n",

" <td>0.087863</td>\n",

" <td>0.304223</td>\n",

" <td>0.857506</td>\n",

" <td>0.580788</td>\n",

" <td>...</td>\n",

" <td>0.099225</td>\n",

" <td>0.218461</td>\n",

" <td>2.040574</td>\n",

" <td>2.074994</td>\n",

" <td>0.077885</td>\n",

" <td>0.660863</td>\n",

" <td>0.202379</td>\n",

" <td>1.208625</td>\n",

" <td>0.665975</td>\n",

" <td>0.869973</td>\n",

" </tr>\n",

" <tr>\n",

" <th>14617</th>\n",

" <td>1.569169</td>\n",

" <td>1.469439</td>\n",

" <td>1.467166</td>\n",

" <td>1.107751</td>\n",

" <td>0.236648</td>\n",

" <td>0.929917</td>\n",

" <td>0.087863</td>\n",

" <td>0.304223</td>\n",

" <td>0.648227</td>\n",

" <td>1.431857</td>\n",

" <td>...</td>\n",

" <td>0.302666</td>\n",

" <td>0.218461</td>\n",

" <td>1.202077</td>\n",

" <td>0.465015</td>\n",

" <td>0.728789</td>\n",

" <td>1.254146</td>\n",

" <td>0.254571</td>\n",

" <td>0.014981</td>\n",

" <td>0.106422</td>\n",

" <td>0.897726</td>\n",

" </tr>\n",

" <tr>\n",

" <th>14618</th>\n",

" <td>1.583759</td>\n",

" <td>0.661197</td>\n",

" <td>1.467166</td>\n",

" <td>1.150843</td>\n",

" <td>0.223435</td>\n",

" <td>0.929917</td>\n",

" <td>0.087863</td>\n",

" <td>0.304223</td>\n",

" <td>0.857506</td>\n",

" <td>1.431857</td>\n",

" <td>...</td>\n",

" <td>0.540013</td>\n",

" <td>0.218461</td>\n",

" <td>0.468393</td>\n",

" <td>0.561003</td>\n",

" <td>0.049485</td>\n",

" <td>0.834507</td>\n",

" <td>0.234961</td>\n",

" <td>1.208625</td>\n",

" <td>1.225528</td>\n",

" <td>0.908610</td>\n",

" </tr>\n",

" <tr>\n",

" <th>14619</th>\n",

" <td>1.704643</td>\n",

" <td>0.404121</td>\n",

" <td>1.467166</td>\n",

" <td>1.290892</td>\n",

" <td>0.272250</td>\n",

" <td>0.929917</td>\n",

" <td>0.087863</td>\n",

" <td>0.304223</td>\n",

" <td>0.648227</td>\n",

" <td>1.431857</td>\n",

" <td>...</td>\n",

" <td>0.065318</td>\n",

" <td>4.608517</td>\n",

" <td>0.789353</td>\n",

" <td>1.883745</td>\n",

" <td>1.047213</td>\n",

" <td>1.586963</td>\n",

" <td>0.355886</td>\n",

" <td>0.014981</td>\n",

" <td>1.113618</td>\n",

" <td>1.069146</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>14620 rows × 22 columns</p>\n",

"</div>\n",

" <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-7732e605-cc9c-4d20-989f-a6d4ad3328df')\"\n",

" title=\"Convert this dataframe to an interactive table.\"\n",

" style=\"display:none;\">\n",

" \n",

" <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",

" width=\"24px\">\n",

" <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

" <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",

" </svg>\n",

" </button>\n",

" \n",

" <style>\n",

" .colab-df-container {\n",

" display:flex;\n",

" flex-wrap:wrap;\n",

" gap: 12px;\n",

" }\n",

"\n",

" .colab-df-convert {\n",

" background-color: #E8F0FE;\n",

" border: none;\n",

" border-radius: 50%;\n",

" cursor: pointer;\n",

" display: none;\n",

" fill: #1967D2;\n",

" height: 32px;\n",

" padding: 0 0 0 0;\n",

" width: 32px;\n",

" }\n",

"\n",

" .colab-df-convert:hover {\n",

" background-color: #E2EBFA;\n",

" box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",

" fill: #174EA6;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert {\n",

" background-color: #3B4455;\n",

" fill: #D2E3FC;\n",

" }\n",

"\n",

" [theme=dark] .colab-df-convert:hover {\n",

" background-color: #434B5C;\n",

" box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",

" filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",

" fill: #FFFFFF;\n",

" }\n",

" </style>\n",

"\n",

" <script>\n",

" const buttonEl =\n",

" document.querySelector('#df-7732e605-cc9c-4d20-989f-a6d4ad3328df button.colab-df-convert');\n",

" buttonEl.style.display =\n",

" google.colab.kernel.accessAllowed ? 'block' : 'none';\n",

"\n",

" async function convertToInteractive(key) {\n",

" const element = document.querySelector('#df-7732e605-cc9c-4d20-989f-a6d4ad3328df');\n",

" const dataTable =\n",

" await google.colab.kernel.invokeFunction('convertToInteractive',\n",

" [key], {});\n",

" if (!dataTable) return;\n",

"\n",

" const docLinkHtml = 'Like what you see? Visit the ' +\n",

" '<a target=\"\_blank\" href=https://colab.research.google.com/notebooks/data\_table.ipynb>data table notebook</a>'\n",

" + ' to learn more about interactive tables.';\n",

" element.innerHTML = '';\n",

" dataTable['output\_type'] = 'display\_data';\n",

" await google.colab.output.renderOutput(dataTable, element);\n",

" const docLink = document.createElement('div');\n",

" docLink.innerHTML = docLinkHtml;\n",

" element.appendChild(docLink);\n",

" }\n",

" </script>\n",

" </div>\n",

" </div>\n",

" "

]

},

"metadata": {},

"execution\_count": 29

},

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Warning: Total number of columns (22) exceeds max\_columns (20) limiting to first (20) columns.\n",

"Warning: Total number of columns (22) exceeds max\_columns (20) limiting to first (20) columns.\n"

]

}

],

"source": [

"df1 = df.drop('Date',axis=1)\n",

"z\_score = np.abs(stats.zscore(df1))\n",

"z\_score"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"metadata": {

"scrolled": true,

"id": "a9c2516b",

"outputId": "ee479680-f60c-43d8-9947-fb08caf1aa5d"

},

"outputs": [

{

"data": {

"text/plain": [

"<AxesSubplot:xlabel='number of floors', ylabel='Price'>"

]

},

"execution\_count": 30,

"metadata": {},

"output\_type": "execute\_result"

},

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 640x480 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"sns.boxplot(x=df\_out['number of floors'],y=df\_out['Price'])"

]

}

]

}