Import pandas and load a dataset:

In [2]:

```
import numpy as np
import pandas as pd
```

Load a CSV file into a pandas DataFrame.

In [7]:

```
df=pd.read_csv("HousingData.csv")
df
```

Out[7]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	NaN
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9.08
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6.48
505	0.04741	0.0	11.93	0.0	0.573	6.030	NaN	2.5050	1	273	21.0	396.90	7.88

506 rows × 14 columns

In [47]:

```
df3= pd.read_csv("credits.csv")
df3
```

```
C:\Users\shaw3\AppData\Local\Temp\ipykernel_50408\1418810855.py:1: DtypeWarning: C
olumns (0,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,
30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,5
7,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,8
4,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,10
8,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,
129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,14
9,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,
170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,19
0,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,
211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,23
1,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,
252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,27
2,273,274,275,276,277,278,279,280,281,282,283,284,285,286,287,288,289,290,291,292,
293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,310,311,312,31
3,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,
334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,35
4,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,
375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,39
5,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,
416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,43
6,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,
457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,47
7,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,
498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,51
8,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,
539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,55
9,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,
580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,60
0,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,
621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,64
1,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,
662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,68
2,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,
703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,72
3,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,
744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,76
4,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,
785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,80
5,806,807,808,809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,824,825,
826,827,828,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,84
6,847,848,849,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,
867,868,869,870,871,872,873,874,875,876,877,878,879,880,881,882,883,884,885,886,88
7,888,889,890,891,892,893,894,895,896,897,898,899,900,901,902,903,904,905,906,907,
908,909,910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,92
8,929,930,931,932,933,934,935,936,937,938,939,940,941,942,943,944,945,946,947,948,
949,950,951,952,953,954,955,956,957,958,959,960,961,962,963,964,965,966,967,968,96
9,970,971,972,973,974,975,976,977,978,979,980,981,982,983,984,985,986,987,988,989,
990,991,992,993,994,995,996,997,998,999,1000,1001,1002,1003,1004,1005,1006,1007,10
08,1009,1010,1011,1012,1013,1014,1015,1016,1017,1018,1019,1020,1021,1022,1023,102
4,1025,1026,1027,1028,1029,1030,1031,1032,1033,1034,1035,1036,1037,1038,1039,1040,
1041,1042,1043,1044,1045,1046,1047,1048,1049,1050,1051,1052,1053,1054,1055,1056,10
57,1058,1059,1060,1061,1062,1063,1064,1065,1066,1067,1068,1069,1070,1071,1072,107
3,1074,1075,1076,1077,1078,1079,1080,1081,1082,1083,1084,1085,1086,1087,1088,1089,
1090,1091,1092,1093,1094,1095,1096,1097,1098,1099,1100,1101,1102,1103,1104,1105,11
06,1107,1108,1109,1110,1111,1112,1113,1114,1115,1116,1117,1118,1119,1120,1121,112
2,1123,1124,1125,1126,1127,1128,1129,1130,1131,1132,1133,1134,1135,1136,1137,1138,
1139,1140,1141,1142,1143,1144,1145,1146,1147,1148,1149,1150,1151,1152,1153,1154,11
55,1156,1157,1158,1159,1160,1161,1162,1163,1164,1165,1166,1167,1168,1169,1170,117
1,1172,1173,1174,1175,1176,1177,1178,1179,1180,1181,1182,1183,1184,1185,1186,1187,
1188,1189,1190,1191,1192,1193,1194,1195,1196,1197,1198,1199,1200,1201,1202,1203,12
04,1205,1206,1207,1208,1209,1210,1211,1212,1213,1214,1215,1216,1217,1218,1219,122
0,1221,1222,1223,1224,1225,1226,1227,1228,1229,1230,1231,1232,1233,1234,1235,1236,
1237,1238,1239,1240,1241,1242,1243,1244,1245,1246,1247,1248,1249,1250,1251,1252,12
53,1254,1255,1256,1257,1258,1259,1260,1261,1262,1263,1264) have mixed types. Speci
fy dtype option on import or set low_memory=False.
  df3= pd.read_csv("credits.csv")
```

Out[47]:

	movie_id	title	cast	crew	Unnamed: 4	Unnamed: 5	Unr
0	19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...	NaN	NaN	
1	285	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...	NaN	NaN	
2	206647	Spectre	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...	NaN	NaN	
3	49026	The Dark Knight Rises	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...	NaN	NaN	
4	49529	John Carter	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...	NaN	NaN	
...	...	...	...	...	...	...	
4808	9367	El Mariachi	[{"cast_id": 1, "character": "El Mariachi", "c...	[{"credit_id": "52fe44eec3a36847f80b280b", "de...	NaN	NaN	
4809	72766	Newlyweds	[{"cast_id": 1, "character": "Buzzy", "credit_...	[{"credit_id": "52fe487dc3a368484e0fb013", "de...	NaN	NaN	
4810	231617	Signed, Sealed, Delivered	[{"cast_id": 8, "character": "Oliver O\u2019To...	[{"credit_id": "52fe4df3c3a36847f8275ecf", "de...	NaN	NaN	
4811	126186	Shanghai Calling	[{"cast_id": 3, "character": "Sam", "credit_id...	[{"credit_id": "52fe4ad9c3a368484e16a36b", "de...	NaN	NaN	
4812	25975	My Date with Drew	[{"cast_id": 3, "character": "Herself", "credi...	[{"credit_id": "58ce021b9251415a390165d9", "de...	NaN	NaN	

Inspect the DataFrame:

Display the first 5 rows of the DataFrame.

```
In [8]: df.head()
```

```
Out[8]:
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  PTRATIO       B  LSTAT  I
0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296     15.3  396.90   4.98
1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242     17.8  396.90   9.14
2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242     17.8  392.83   4.03
3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222     18.7  394.63   2.94
4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622    3  222     18.7  396.90    NaN
```

```
In [48]: df3.head()
```

Out[48]:

	movie_id	title	cast	crew	Unnamed: 4	Unnamed: 5	Unnamec
0	19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...	NaN	NaN	NaI
1	285	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...	NaN	NaN	NaI
2	206647	Spectre	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...	NaN	NaN	NaI
3	49026	The Dark Knight Rises	[{"cast_id": 2, "character": "Bruce Wayne / Ba...	[{"credit_id": "52fe4781c3a36847f81398c3", "de...	NaN	NaN	NaI
4	49529	John Carter	[{"cast_id": 5, "character": "John Carter", "c...	[{"credit_id": "52fe479ac3a36847f813eaa3", "de...	NaN	NaN	NaI

5 rows × 1265 columns

Get DataFrame information:

Display information about the DataFrame (e.g., column names, data types, number of non-null values).

In [10]:

```
df.shape
```

Out[10]:

```
(506, 14)
```

In [13]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   CRIM     486 non-null    float64
 1   ZN       486 non-null    float64
 2   INDUS    486 non-null    float64
 3   CHAS     486 non-null    float64
 4   NOX      506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      486 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    int64  
 9   TAX      506 non-null    int64  
 10  PTRATIO  506 non-null    float64
 11  B        506 non-null    float64
 12  LSTAT    486 non-null    float64
 13  MEDV     506 non-null    float64
dtypes: float64(12), int64(2)
memory usage: 55.5 KB
```

In [14]:

```
df.describe()
```

Out[14]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	
count	486.000000	486.000000	486.000000	486.000000	506.000000	506.000000	486.000000	506.00
mean	3.611874	11.211934	11.083992	0.069959	0.554695	6.284634	68.518519	3.79
std	8.720192	23.388876	6.835896	0.255340	0.115878	0.702617	27.999513	2.10
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.12
25%	0.081900	0.000000	5.190000	0.000000	0.449000	5.885500	45.175000	2.10
50%	0.253715	0.000000	9.690000	0.000000	0.538000	6.208500	76.800000	3.20
75%	3.560263	12.500000	18.100000	0.000000	0.624000	6.623500	93.975000	5.18
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.12

Check for missing vallue:

Check if there are any missing values in the DataFrame.

In [16]:

```
df.isnull().sum()
```

```
Out[16]: CRIM       20
         ZN         20
         INDUS      20
         CHAS       20
         NOX         0
         RM          0
         AGE        20
         DIS         0
         RAD         0
         TAX         0
         PTRATIO     0
         B           0
         LSTAT      20
         MEDV        0
         dtype: int64
```

Drop missing values:

```
In [17]: df_dropped = df.dropna()
         print(df_dropped.head())
```

```
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  PTRATIO  \
0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296     15.3   
1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242     17.8   
2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242     17.8   
3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222     18.7   
5  0.02985   0.0   2.18   0.0  0.458  6.430  58.7  6.0622    3  222     18.7   

        B  LSTAT  MEDV  
0  396.90   4.98  24.0  
1  396.90   9.14  21.6  
2  392.83   4.03  34.7  
3  394.63   2.94  33.4  
5  394.12   5.21  28.7  
```

Fill missing values:

```
In [18]: df_filled = df.fillna(0)
         print(df_filled.head())
```

```
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  PTRATIO  \
0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296     15.3   
1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242     17.8   
2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242     17.8   
3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222     18.7   
4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622    3  222     18.7   

        B  LSTAT  MEDV  
0  396.90   4.98  24.0  
1  396.90   9.14  21.6  
2  392.83   4.03  34.7  
3  394.63   2.94  33.4  
4  396.90   0.00  36.2  
```

Select specific columns:

Select and display specific columns from the DataFrame.

In [20]:
```
selected_columns= df[['CRIM','ZN']]
selected_columns
```

Out[20]:

	CRIM	ZN
0	0.00632	18.0
1	0.02731	0.0
2	0.02729	0.0
3	0.03237	0.0
4	0.06905	0.0
...	...	...
501	0.06263	0.0
502	0.04527	0.0
503	0.06076	0.0
504	0.10959	0.0
505	0.04741	0.0

506 rows × 2 columns

Filter rows:

In [27]:
```
filtered_df = df[df['ZN'] > 25]
print(filtered_df)
```

```
        CRIM    ZN  INDUS  CHAS    NOX     RM   AGE      DIS  RAD  TAX  \
39   0.02763  75.0   2.95   0.0  0.428  6.595  21.8   5.4011    3  252   
40   0.03359  75.0   2.95   0.0  0.428  7.024  15.8   5.4011    3  252   
54   0.01360  75.0   4.00   0.0  0.410  5.888  47.6   7.3197    3  469   
55   0.01311  90.0   1.22   0.0  0.403  7.249  21.9   8.6966    5  226   
56   0.02055  85.0   0.74   0.0  0.410  6.383  35.7   9.1876    2  313   
..       ...   ...    ...   ...    ...    ...   ...      ...  ...  ...   
349  0.02899  40.0   1.25   0.0  0.429  6.939  34.5   8.7921    1  335   
351  0.07950  60.0   1.69   0.0  0.411  6.579  35.9  10.7103    4  411   
352  0.07244  60.0   1.69   0.0  0.411  5.884  18.5  10.7103    4  411   
353  0.01709  90.0   2.02   0.0  0.410  6.728  36.1  12.1265    5  187   
354  0.04301  80.0   1.91   0.0  0.413  5.663  21.9  10.5857    4  334   

     PTRATIO       B  LSTAT  MEDV  
39      18.3  395.63   4.32  30.8  
40      18.3  395.62   1.98  34.9  
54      21.1  396.90  14.80  18.9  
55      17.9  395.93   4.81  35.4  
56      17.3  396.90   5.77  24.7  
..       ...     ...    ...   ...  
349     19.7  389.85    NaN  26.6  
351     18.3  370.78   5.49  24.1  
352     18.3  392.33   7.79  18.6  
353     17.0  384.46   4.50  30.1  
354     22.0  382.80   8.05  18.2  

[70 rows x 14 columns]
```

```
In [28]: filtered_df = df[df['ZN'] > 10]
         print(filtered_df)
```

```
        CRIM    ZN  INDUS  CHAS    NOX     RM    AGE      DIS  RAD  TAX  \
0    0.00632  18.0   2.31   0.0  0.538  6.575   65.2   4.0900    1  296   
6    0.08829  12.5   7.87   NaN  0.524  6.012   66.6   5.5605    5  311   
7    0.14455  12.5   7.87   0.0  0.524  6.172   96.1   5.9505    5  311   
8    0.21124  12.5   7.87   0.0  0.524  5.631  100.0   6.0821    5  311   
9    0.17004  12.5   7.87   NaN  0.524  6.004   85.9   6.5921    5  311   
..       ...   ...    ...   ...    ...    ...    ...      ...  ...  ...   
349  0.02899  40.0   1.25   0.0  0.429  6.939   34.5   8.7921    1  335   
351  0.07950  60.0   1.69   0.0  0.411  6.579   35.9  10.7103    4  411   
352  0.07244  60.0   1.69   0.0  0.411  5.884   18.5  10.7103    4  411   
353  0.01709  90.0   2.02   0.0  0.410  6.728   36.1  12.1265    5  187   
354  0.04301  80.0   1.91   0.0  0.413  5.663   21.9  10.5857    4  334   

     PTRATIO       B  LSTAT  MEDV  
0       15.3  396.90   4.98  24.0  
6       15.2  395.60  12.43  22.9  
7       15.2  396.90  19.15  27.1  
8       15.2  386.63  29.93  16.5  
9       15.2  386.71  17.10  18.9  
..       ...     ...    ...   ...  
349     19.7  389.85    NaN  26.6  
351     18.3  370.78   5.49  24.1  
352     18.3  392.33   7.79  18.6  
353     17.0  384.46   4.50  30.1  
354     22.0  382.80   8.05  18.2  

[126 rows x 14 columns]
```

Sort the DataFrame:

Sort the DataFrame by a specific column.

```
In [29]: sorted_df = df.sort_values(by='ZN')
         print(sorted_df.head())
```

```
         CRIM   ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  \
505   0.04741  0.0  11.93   0.0  0.573  6.030   NaN  2.5050    1  273   
438  13.67810  0.0  18.10   0.0  0.740  5.935  87.9  1.8206   24  666   
439   9.39063  0.0  18.10   0.0  0.740  5.627  93.9  1.8172   24  666   
440  22.05110  0.0  18.10   0.0  0.740  5.818  92.4  1.8662   24  666   
441   9.72418  0.0  18.10   0.0  0.740  6.406  97.2  2.0651   24  666   

     PTRATIO       B  LSTAT  MEDV  
505     21.0  396.90   7.88  11.9  
438     20.2   68.95  34.02   8.4  
439     20.2  396.90  22.88  12.8  
440     20.2  391.45    NaN  10.5  
441     20.2  385.96    NaN  17.1  
```

Group by and aggregate:

Group the DataFrame by a column and calculate aggregate statistics.

```
In [30]: grouped_df = df.groupby('CRIM').agg({'ZN': 'mean'})
         print(grouped_df)
```

```
                ZN
CRIM              
0.00632     18.0
0.00906     90.0
0.01096     55.0
0.01301     35.0
0.01311     90.0
...          ...
45.74610     0.0
51.13580     0.0
67.92080     0.0
73.53410     0.0
88.97620     0.0

[484 rows x 1 columns]
```

Create a new column:

Add a new column to the DataFrame based on existing data.

In [31]:

```
df['new_column'] = df['NOX'] + df['RM']
print(df)
```

```
        CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  \
0    0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296   
1    0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242   
2    0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242   
3    0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222   
4    0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622    3  222   
..       ...   ...    ...   ...    ...    ...   ...     ...  ...  ...   
501  0.06263   0.0  11.93   0.0  0.573  6.593  69.1  2.4786    1  273   
502  0.04527   0.0  11.93   0.0  0.573  6.120  76.7  2.2875    1  273   
503  0.06076   0.0  11.93   0.0  0.573  6.976  91.0  2.1675    1  273   
504  0.10959   0.0  11.93   0.0  0.573  6.794  89.3  2.3889    1  273   
505  0.04741   0.0  11.93   0.0  0.573  6.030   NaN  2.5050    1  273   

     PTRATIO       B  LSTAT  MEDV  new_column  
0       15.3  396.90   4.98  24.0       7.113  
1       17.8  396.90   9.14  21.6       6.890  
2       17.8  392.83   4.03  34.7       7.654  
3       18.7  394.63   2.94  33.4       7.456  
4       18.7  396.90    NaN  36.2       7.605  
..       ...     ...    ...   ...         ...  
501     21.0  391.99    NaN  22.4       7.166  
502     21.0  396.90   9.08  20.6       6.693  
503     21.0  396.90   5.64  23.9       7.549  
504     21.0  393.45   6.48  22.0       7.367  
505     21.0  396.90   7.88  11.9       6.603  

[506 rows x 15 columns]
```

Apply a function to a column:

Apply a custom function to a column.

In [32]:

```
df['new_column'] = df['ZN'].apply(lambda x: x * 2)
print(df)
```

```
        CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  \
0    0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296   
1    0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242   
2    0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242   
3    0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222   
4    0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622    3  222   
..       ...   ...    ...   ...    ...    ...   ...     ...  ...  ...   
501  0.06263   0.0  11.93   0.0  0.573  6.593  69.1  2.4786    1  273   
502  0.04527   0.0  11.93   0.0  0.573  6.120  76.7  2.2875    1  273   
503  0.06076   0.0  11.93   0.0  0.573  6.976  91.0  2.1675    1  273   
504  0.10959   0.0  11.93   0.0  0.573  6.794  89.3  2.3889    1  273   
505  0.04741   0.0  11.93   0.0  0.573  6.030   NaN  2.5050    1  273   

     PTRATIO       B  LSTAT  MEDV  new_column  
0       15.3  396.90   4.98  24.0        36.0  
1       17.8  396.90   9.14  21.6         0.0  
2       17.8  392.83   4.03  34.7         0.0  
3       18.7  394.63   2.94  33.4         0.0  
4       18.7  396.90    NaN  36.2         0.0  
..       ...     ...    ...   ...         ...  
501     21.0  391.99    NaN  22.4         0.0  
502     21.0  396.90   9.08  20.6         0.0  
503     21.0  396.90   5.64  23.9         0.0  
504     21.0  393.45   6.48  22.0         0.0  
505     21.0  396.90   7.88  11.9         0.0  

[506 rows x 15 columns]
```

Rename columns:

In [55]:

```
df = df.rename(columns={'ZN': 'OX'})
print(df)
```

```
         CRIM                            OX  INDUS  CHAS    NOX     RM   AGE  \
0     0.00632 1970-01-01 00:00:00.000000018   2.31   0.0  0.538  6.575  65.2   
1     0.02731 1970-01-01 00:00:00.000000000   7.07   0.0  0.469  6.421  78.9   
2     0.02729 1970-01-01 00:00:00.000000000   7.07   0.0  0.469  7.185  61.1   
3     0.03237 1970-01-01 00:00:00.000000000   2.18   0.0  0.458  6.998  45.8   
4     0.06905 1970-01-01 00:00:00.000000000   2.18   0.0  0.458  7.147  54.2   
..        ...                           ...    ...   ...    ...    ...   ...   
501   0.06263 1970-01-01 00:00:00.000000000  11.93   0.0  0.573  6.593  69.1   
502   0.04527 1970-01-01 00:00:00.000000000  11.93   0.0  0.573  6.120  76.7   
503   0.06076 1970-01-01 00:00:00.000000000  11.93   0.0  0.573  6.976  91.0   
504   0.10959 1970-01-01 00:00:00.000000000  11.93   0.0  0.573  6.794  89.3   
505   0.04741 1970-01-01 00:00:00.000000000  11.93   0.0  0.573  6.030   NaN   

        DIS  RAD  TAX  PTRATIO       B  LSTAT  MEDV  new_column    year  \
0    4.0900    1  296     15.3  396.90   4.98  24.0        36.0  1970.0   
1    4.9671    2  242     17.8  396.90   9.14  21.6         0.0  1970.0   
2    4.9671    2  242     17.8  392.83   4.03  34.7         0.0  1970.0   
3    6.0622    3  222     18.7  394.63   2.94  33.4         0.0  1970.0   
4    6.0622    3  222     18.7  396.90    NaN  36.2         0.0  1970.0   
..      ...  ...  ...      ...     ...    ...   ...         ...     ...   
501  2.4786    1  273     21.0  391.99    NaN  22.4         0.0  1970.0   
502  2.2875    1  273     21.0  396.90   9.08  20.6         0.0  1970.0   
503  2.1675    1  273     21.0  396.90   5.64  23.9         0.0  1970.0   
504  2.3889    1  273     21.0  393.45   6.48  22.0         0.0  1970.0   
505  2.5050    1  273     21.0  396.90   7.88  11.9         0.0  1970.0   

     month  day  
0      1.0  1.0  
1      1.0  1.0  
2      1.0  1.0  
3      1.0  1.0  
4      1.0  1.0  
..     ...  ...  
501    1.0  1.0  
502    1.0  1.0  
503    1.0  1.0  
504    1.0  1.0  
505    1.0  1.0  

[506 rows x 18 columns]
```

Remove duplicates:

In [39]:

```
df = df.drop_duplicates()
print(df.head())
```

```
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  PTRATIO  \
0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296     15.3   
1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242     17.8   
2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242     17.8   
3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222     18.7   
4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622    3  222     18.7   

        B  LSTAT  MEDV  new_column  
0  396.90   4.98  24.0        36.0  
1  396.90   9.14  21.6         0.0  
2  392.83   4.03  34.7         0.0  
3  394.63   2.94  33.4         0.0  
4  396.90    NaN  36.2         0.0  
```

Merge DataFrames:

Merge two DataFrames on a common column.

In [ ]:

```
merged_df = pd.merge(df, df3, on='common_column')
print(merged_df)
```

Concatenate DataFrames:

In [49]:

```
concatenated_df = pd.concat([df, df3], axis=0)
print(concatenated_df)
```

```
         CRIM                            ZN  INDUS  CHAS    NOX     RM   AGE  \
0     0.00632 1970-01-01 00:00:00.000000018   2.31   0.0  0.538  6.575  65.2
1     0.02731 1970-01-01 00:00:00.000000000   7.07   0.0  0.469  6.421  78.9
2     0.02729 1970-01-01 00:00:00.000000000   7.07   0.0  0.469  7.185  61.1
3     0.03237 1970-01-01 00:00:00.000000000   2.18   0.0  0.458  6.998  45.8
4     0.06905 1970-01-01 00:00:00.000000000   2.18   0.0  0.458  7.147  54.2
...       ...                           ...    ...   ...    ...    ...   ...
4808      NaN                           NaT    NaN   NaN    NaN    NaN   NaN
4809      NaN                           NaT    NaN   NaN    NaN    NaN   NaN
4810      NaN                           NaT    NaN   NaN    NaN    NaN   NaN
4811      NaN                           NaT    NaN   NaN    NaN    NaN   NaN
4812      NaN                           NaT    NaN   NaN    NaN    NaN   NaN

         DIS  RAD    TAX  ...  Unnamed: 1255  Unnamed: 1256  Unnamed: 1257  \
0     4.0900  1.0  296.0  ...            NaN            NaN            NaN
1     4.9671  2.0  242.0  ...            NaN            NaN            NaN
2     4.9671  2.0  242.0  ...            NaN            NaN            NaN
3     6.0622  3.0  222.0  ...            NaN            NaN            NaN
4     6.0622  3.0  222.0  ...            NaN            NaN            NaN
...      ...  ...    ...  ...            ...            ...            ...
4808     NaN  NaN    NaN  ...            NaN            NaN            NaN
4809     NaN  NaN    NaN  ...            NaN            NaN            NaN
4810     NaN  NaN    NaN  ...            NaN            NaN            NaN
4811     NaN  NaN    NaN  ...            NaN            NaN            NaN
4812     NaN  NaN    NaN  ...            NaN            NaN            NaN

      Unnamed: 1258  Unnamed: 1259 Unnamed: 1260 Unnamed: 1261 Unnamed: 1262  \
0               NaN            NaN           NaN           NaN           NaN
1               NaN            NaN           NaN           NaN           NaN
2               NaN            NaN           NaN           NaN           NaN
3               NaN            NaN           NaN           NaN           NaN
4               NaN            NaN           NaN           NaN           NaN
...             ...            ...           ...           ...           ...
4808            NaN            NaN           NaN           NaN           NaN
4809            NaN            NaN           NaN           NaN           NaN
4810            NaN            NaN           NaN           NaN           NaN
4811            NaN            NaN           NaN           NaN           NaN
4812            NaN            NaN           NaN           NaN           NaN

     Unnamed: 1263 Unnamed: 1264
0              NaN           NaN
1              NaN           NaN
2              NaN           NaN
3              NaN           NaN
4              NaN           NaN
...            ...           ...
4808           NaN           NaN
4809           NaN           NaN
4810           NaN           NaN
4811           NaN           NaN
4812           NaN           NaN

[5319 rows x 1280 columns]
```

Date and time operations:

Convert a column to datetime format and extract date parts.

In [50]:

```
df['ZN'] = pd.to_datetime(df['ZN'])
df['year'] = df['ZN'].dt.year
df['month'] = df['ZN'].dt.month
```

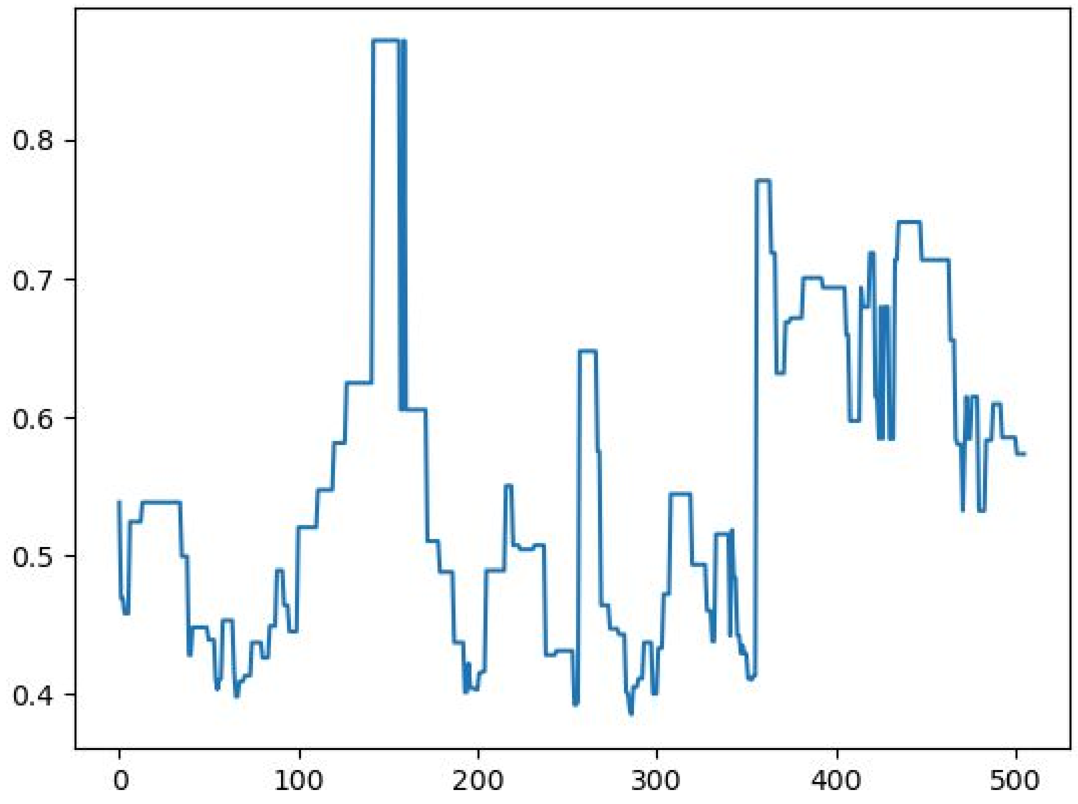
```
df['day'] = df['ZN'].dt.day
print(df)
```

```
        CRIM                            ZN  INDUS  CHAS    NOX     RM   AGE  \
0    0.00632 1970-01-01 00:00:00.000000018   2.31   0.0  0.538  6.575  65.2   
1    0.02731 1970-01-01 00:00:00.000000000   7.07   0.0  0.469  6.421  78.9   
2    0.02729 1970-01-01 00:00:00.000000000   7.07   0.0  0.469  7.185  61.1   
3    0.03237 1970-01-01 00:00:00.000000000   2.18   0.0  0.458  6.998  45.8   
4    0.06905 1970-01-01 00:00:00.000000000   2.18   0.0  0.458  7.147  54.2   
..       ...                           ...    ...   ...    ...    ...   ...   
501  0.06263 1970-01-01 00:00:00.000000000  11.93   0.0  0.573  6.593  69.1   
502  0.04527 1970-01-01 00:00:00.000000000  11.93   0.0  0.573  6.120  76.7   
503  0.06076 1970-01-01 00:00:00.000000000  11.93   0.0  0.573  6.976  91.0   
504  0.10959 1970-01-01 00:00:00.000000000  11.93   0.0  0.573  6.794  89.3   
505  0.04741 1970-01-01 00:00:00.000000000  11.93   0.0  0.573  6.030   NaN   

        DIS  RAD  TAX  PTRATIO       B  LSTAT  MEDV  new_column    year  \
0    4.0900    1  296     15.3  396.90   4.98  24.0        36.0  1970.0   
1    4.9671    2  242     17.8  396.90   9.14  21.6         0.0  1970.0   
2    4.9671    2  242     17.8  392.83   4.03  34.7         0.0  1970.0   
3    6.0622    3  222     18.7  394.63   2.94  33.4         0.0  1970.0   
4    6.0622    3  222     18.7  396.90    NaN  36.2         0.0  1970.0   
..      ...  ...  ...      ...     ...    ...   ...         ...     ...   
501  2.4786    1  273     21.0  391.99    NaN  22.4         0.0  1970.0   
502  2.2875    1  273     21.0  396.90   9.08  20.6         0.0  1970.0   
503  2.1675    1  273     21.0  396.90   5.64  23.9         0.0  1970.0   
504  2.3889    1  273     21.0  393.45   6.48  22.0         0.0  1970.0   
505  2.5050    1  273     21.0  396.90   7.88  11.9         0.0  1970.0   

     month  day  
0      1.0  1.0  
1      1.0  1.0  
2      1.0  1.0  
3      1.0  1.0  
4      1.0  1.0  
..     ...  ...  
501    1.0  1.0  
502    1.0  1.0  
503    1.0  1.0  
504    1.0  1.0  
505    1.0  1.0  

[506 rows x 18 columns]
```

Plot data:

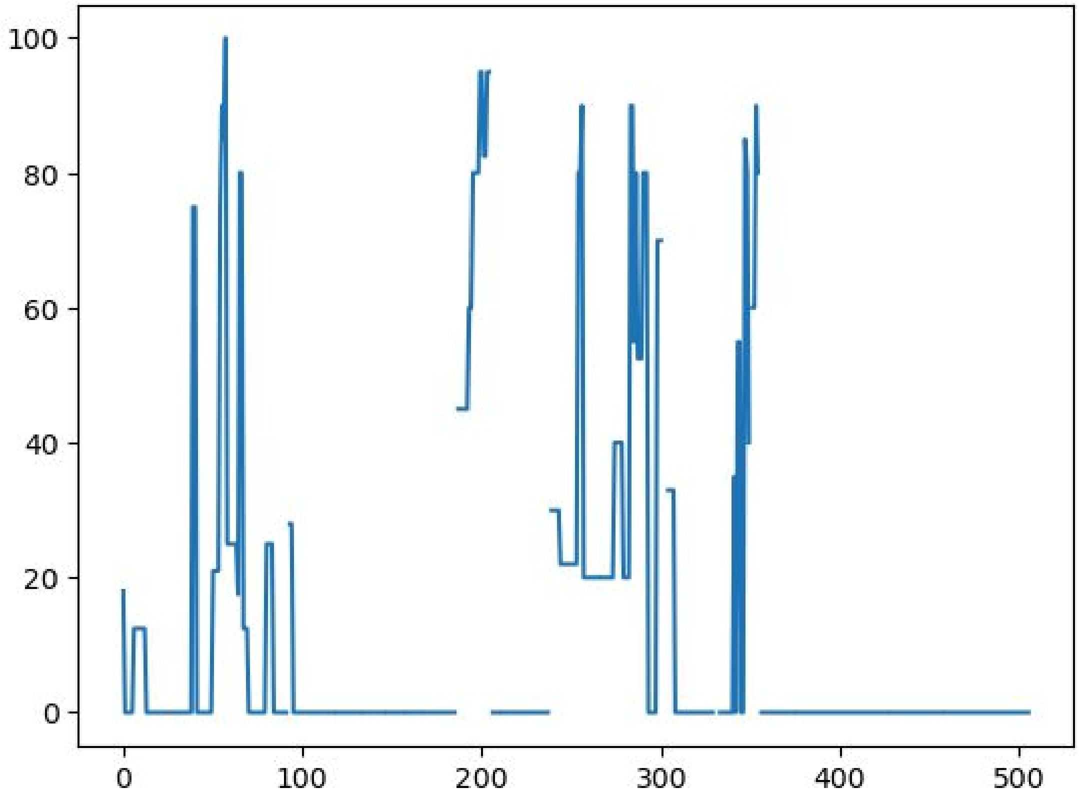
```
In [41]: df['NOX'].plot(kind='line')
```

```
Out[41]: <AxesSubplot:>
```



```
In [42]: df['ZN'].plot(kind='line')
```

```
Out[42]: <AxesSubplot:>
```



Save DataFrame to CSV:

```
In [44]: df.to_csv('new_file.csv', index=False)
```

In [ ]:

```
```