In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
from sklearn.datasets import load_boston
boston = load_boston()
```

In [3]:
```
data = pd.DataFrame(boston.data)
```

In [4]:
```
data.head()
```

Out[4]:

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

In [5]:
```
data.columns = boston.feature_names
```

In [6]:
```
data['PRICE'] = boston.target
```

In [7]:
```
data.head(n=10)
```

Out[7]:

	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	1.0	296.0	15.3	39
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	39
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	39
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	39
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	39
5	0.02985	0.0	2.18	0.0	0.458	6.430	58.7	6.0622	3.0	222.0	18.7	39
6	0.08829	12.5	7.87	0.0	0.524	6.012	66.6	5.5605	5.0	311.0	15.2	39
7	0.14455	12.5	7.87	0.0	0.524	6.172	96.1	5.9505	5.0	311.0	15.2	39
8	0.21124	12.5	7.87	0.0	0.524	5.631	100.0	6.0821	5.0	311.0	15.2	38
9	0.17004	12.5	7.87	0.0	0.524	6.004	85.9	6.5921	5.0	311.0	15.2	38

In [8]:

```
#Shape of the data
print(data.shape)
```

```
(506, 14)
```

In [9]:

```
data.isnull().sum()
```

Out[9]:

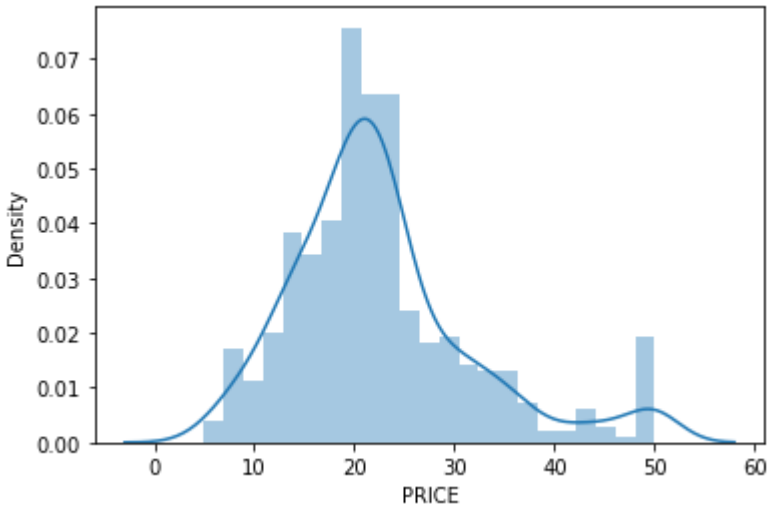
```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
PRICE      0
dtype: int64
```

In [10]:

```
import seaborn as sns
sns.distplot(data.PRICE)
```

```
/home/sitrc/anaconda3/lib/python3.9/site-packages/seaborn/distributi
ons.py:2619: FutureWarning: `distplot` is a deprecated function and
will be removed in a future version. Please adapt your code to use e
ither `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

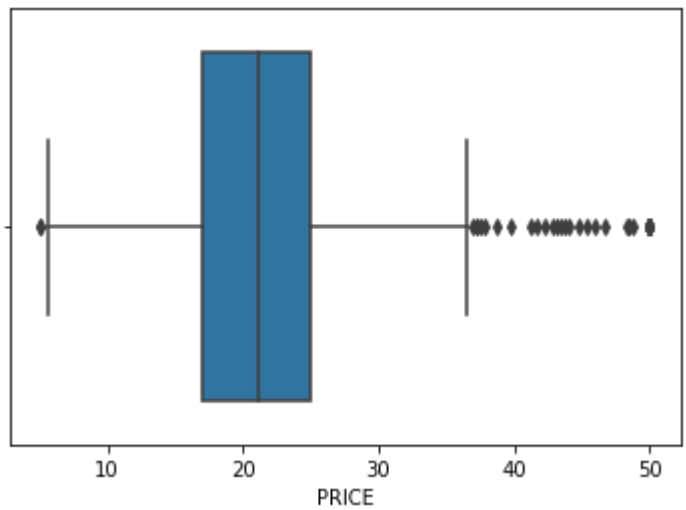
Out[10]:
```
<AxesSubplot:xlabel='PRICE', ylabel='Density'>
```



In [11]:
```
sns.boxplot(data.PRICE)
```

```
/home/sitrc/anaconda3/lib/python3.9/site-packages/seaborn/_decorator
s.py:36: FutureWarning: Pass the following variable as a keyword arg
: x. From version 0.12, the only valid positional argument will be `
data`, and passing other arguments without an explicit keyword will
result in an error or misinterpretation.
  warnings.warn(
```

Out[11]:
```
<AxesSubplot:xlabel='PRICE'>
```



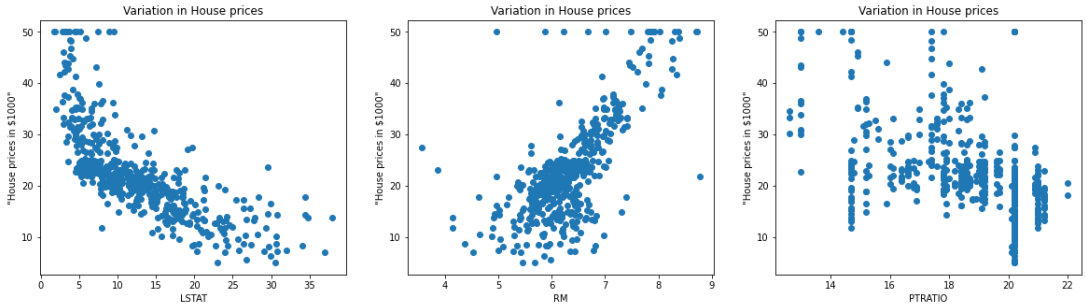
In [12]:
```
correlation = data.corr()
correlation.loc['PRICE']
```

Out[12]:
```
CRIM      -0.388305
ZN         0.360445
INDUS     -0.483725
CHAS       0.175260
NOX       -0.427321
RM         0.695360
AGE       -0.376955
DIS        0.249929
RAD       -0.381626
TAX       -0.468536
PTRATIO   -0.507787
B          0.333461
LSTAT     -0.737663
PRICE      1.000000
Name: PRICE, dtype: float64
```

In [14]:
```
import matplotlib.pyplot as plt
fig,axes = plt.subplots(figsize=(15,12))
sns.heatmap(correlation,square =True,annot =True)
```

Out[14]: <AxesSubplot:>

In [16]:
```
plt.figure(figsize = (20,5))
features = ['LSTAT','RM','PTRATIO']
for i, col in enumerate(features):
    plt.subplot(1,len(features) , i+1)
    x = data[col]
    y = data.PRICE
    plt.scatter(x, y, marker='o')
    plt.title("Variation in House prices")
    plt.xlabel(col)
    plt.ylabel('"House prices in $1000"')
```



In [17]:
```
X = data.iloc[:,:-1]
y= data.PRICE
```

In [19]:
```
mean = X_train.mean(axis=0)
std = X_train.std(axis=0)
X_train = (X_train - mean) / std
X_test = (X_test - mean) / std
```

```
---------------------------------------------------------------------------
-------
NameError                                 Traceback (most recent cal
l last)
/tmp/ipykernel_3653/626873035.py in <module>
----> 1 mean = X_train.mean(axis=0)
      2 std = X_train.std(axis=0)
      3 X_train = (X_train - mean) / std
      4 X_test = (X_test - mean) / std

NameError: name 'X_train' is not defined
```

In [20]:
```
from sklearn.linear_model
import LinearRegression
```

```
  File "/tmp/ipykernel_3653/696221428.py", line 1
    from sklearn.linear_model
                              ^
SyntaxError: invalid syntax
```

In [ ]: