In [23]:
```
#Part 1: How to load data file(s) using Pandas?
```

In [24]:
```
import pandas as pd
import numpy
import scipy
```

In [25]:
```
df=pd.read_csv("House_Rent.csv")
```

In [26]:
```
new_df=df
df
```

Out[26]:

	BHK	Rent	Size	Area Type	City	Furnishing Status	Bathroom
0	2	10000	1100	1	1	1	2
1	2	20000	800	1	1	2	1
2	2	17000	1000	1	1	2	1
3	2	10000	800	1	1	1	1
4	2	7500	850	2	1	1	1
...	...	...	...	...	...	...	...
4714	2	15000	1000	2	6	2	2
4715	3	29000	2000	1	6	2	3
4716	3	35000	1750	2	6	2	3
4717	3	45000	1500	2	6	2	2
4718	2	15000	1000	2	6	1	2

4719 rows × 7 columns

In [27]:
```
#Part 2: How to convert a variable to a different data type?
```

In [28]:
```
from datetime import datetime
char_date = 'Apr 1 2015 1:20PM' #creating example character date
date_obj = datetime.strptime(char_date, '%b %d %Y %I:%M%p')
print(date_obj)
```

```
2015-04-01 13:20:00
```

In [29]:
```
#Part 3: How to transpose a Data set or dataframe using Pandas?
```

In [30]:
```
#Transposing Pandas dataframe by a variable
df=pd.read_csv("House_Rent.csv")
print (df)
result= df.pivot( columns='City', values='BHK')
result
```

```
      BHK   Rent  Size Area Type  City  Furnishing Status  Bathroom
0       2  10000  1100         1     1                  1         2
1       2  20000   800         1     1                  2         1
2       2  17000  1000         1     1                  2         1
3       2  10000   800         1     1                  1         1
4       2   7500   850         2     1                  1         1
...   ...    ...   ...       ...   ...                ...       ...
4714    2  15000  1000         2     6                  2         2
4715    3  29000  2000         1     6                  2         3
4716    3  35000  1750         2     6                  2         3
4717    3  45000  1500         2     6                  2         2
4718    2  15000  1000         2     6                  1         2

[4719 rows x 7 columns]
```

Out[30]:

City	1	2	3	4	5	6
0	2.0	NaN	NaN	NaN	NaN	NaN
1	2.0	NaN	NaN	NaN	NaN	NaN
2	2.0	NaN	NaN	NaN	NaN	NaN
3	2.0	NaN	NaN	NaN	NaN	NaN
4	2.0	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...
4714	NaN	NaN	NaN	NaN	NaN	2.0
4715	NaN	NaN	NaN	NaN	NaN	3.0
4716	NaN	NaN	NaN	NaN	NaN	3.0
4717	NaN	NaN	NaN	NaN	NaN	3.0
4718	NaN	NaN	NaN	NaN	NaN	2.0

4719 rows × 6 columns

In [31]:
```
#Part 4: How to sort a Pandas DataFrame?
```

In [35]:
```
#Sorting Pandas Dataframe
df=pd.read_csv("House_Rent.csv")
df.sort_values(['Rent','Size'], ascending=[True, False])
```

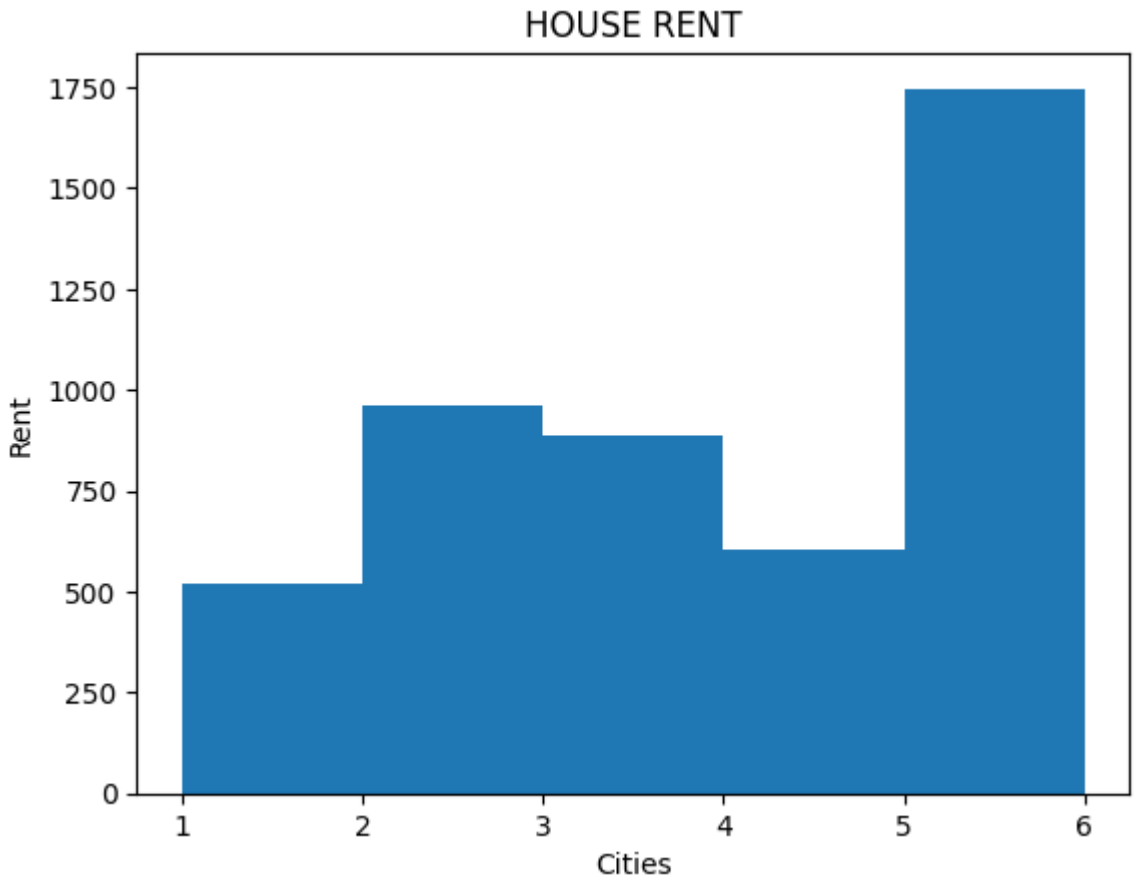
Out[35]:

	BHK	Rent	Size	Area Type	City	Furnishing Status	Bathroom
4055	3	1200	2100	2	6	2	3
284	1	1500	200	1	1	2	1
469	1	1800	500	1	1	2	1
2461	2	2000	60	1	4	1	1
504	1	2200	700	1	1	1	1
...	...	...	...	...	...	...	...
1445	4	700000	3200	2	2	2	4
1317	4	850000	3200	2	2	2	4
821	4	1000000	3064	2	2	2	4
993	4	1200000	5000	2	2	2	4
1823	3	3500000	2500	2	3	2	3

4719 rows × 7 columns

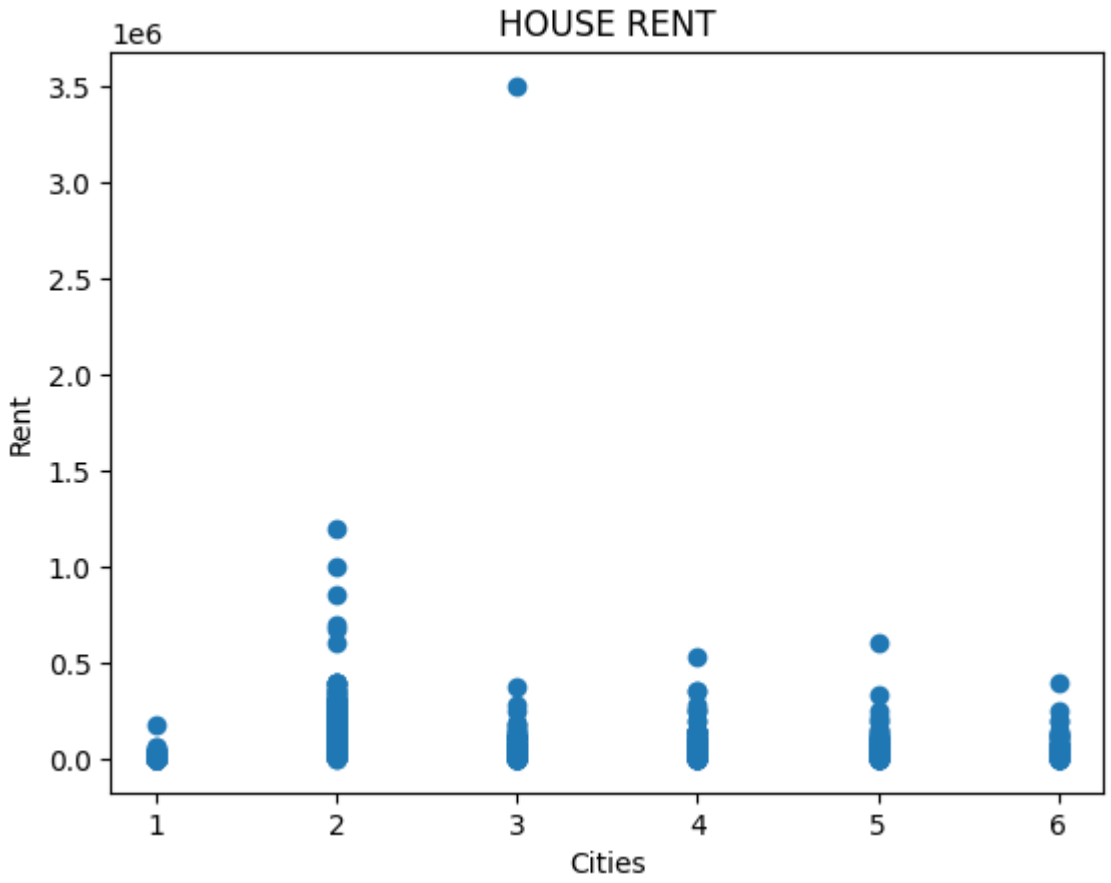
In [36]:
```
#Part 5: How to create plots (Histogram, Scatter, Box Plot)?
```

In [44]:
```
#Plot Histogram
import matplotlib.pyplot as plt
import pandas as pd
df=pd.read_csv("House_Rent.csv")
#Plots in matplotlib reside within a figure object, use plt.figure to create n
fig=plt.figure()
#Create one or more subplots using add_subplot, because you can't create blank
ax = fig.add_subplot(1,1,1)
#Variable
ax.hist(df['City'],bins = 5)
#Labels and Tit
plt.title('HOUSE RENT')
plt.xlabel('Cities')
plt.ylabel('Rent')
plt.show()
```



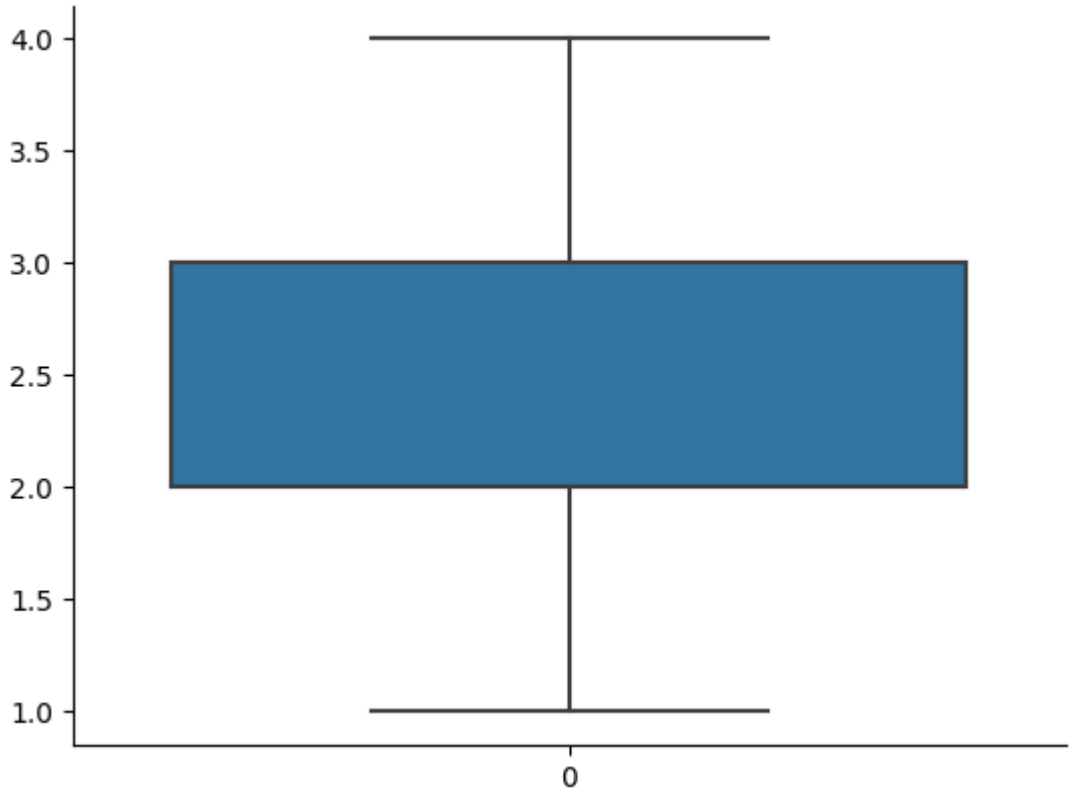
In [45]:
```
#Scatter plot
```

In [49]:
```
#Plots in matplotlib reside within a figure object, use plt.figure to create n
fig=plt.figure()
#Create one or more subplots using add_subplot, because you can't create blank
ax = fig.add_subplot(1,1,1)
#Variable
ax.scatter(df['City'],df['Rent'])
#Labels and Tit
plt.title('HOUSE RENT')
plt.xlabel('Cities')
plt.ylabel('Rent')
plt.show()
```



In [50]:
```
#Box-plot
```

In [52]:
```
import seaborn as sns
sns.boxplot(df['BHK'])
sns.despine()
```



In [53]:
```
#Part 6: How to generate frequency tables with Pandas?
```

In [55]:
```
import pandas as pd
df=pd.read_csv("House_Rent.csv")
test= df.groupby(['BHK','Rent'])
test.size()
```

Out[55]:
```
BHK  Rent
1    1500       1
     1800       1
     2200       1
     3000       8
     3200       1
                ..
4    680000     1
     700000     1
     850000     1
     1000000    1
     1200000    1
Length: 467, dtype: int64
```

In [56]:
```
#Part 7: How to do sample Data set in Python?
```

In [62]:
```
#Create Sample dataframe
import numpy as np
import pandas as pd
from random import sample
dfr = df.sample(n=5)
print (dfr)
```

	BHK	Rent	Size	Area Type	City	Furnishing Status	Bathroom
2426	1	9500	600	1	4	2	2
2026	2	12000	700	2	3	2	1
4496	3	9000	1100	2	6	2	1
2996	2	15000	850	1	5	2	1
2808	4	250000	2800	2	4	1	4

In [63]:
```
#Part 8: How to remove duplicate values of a variable in a Pandas Dataframe?
```

In [65]:
```
#Remove Duplicate Values based on values of variables "Gender" and "BMI"
rem_dup=df.drop_duplicates(['BHK', 'Rent'])
print (rem_dup)
```

	BHK	Rent	Size	Area Type	City	Furnishing Status	Bathroom
0	2	10000	1100	1	1	1	2
1	2	20000	800	1	1	2	1
2	2	17000	1000	1	1	2	1
4	2	7500	850	2	1	1	1
5	2	7000	600	1	1	1	2
...	...	...	...	...	...	...	...
4312	4	52000	2700	1	6	2	4
4354	3	21467	200	1	6	1	1
4501	2	5800	760	1	6	1	2
4528	4	17000	1600	1	6	1	4
4687	4	16000	1000	1	6	2	2

[467 rows x 7 columns]

In [66]:
```
#Part 9: How to group variables in Pandas to calculate count, average, sum?
```

In [68]:
```
test= df.groupby(['BHK'])
test.describe()
```

Out[68]:

	Rent								Si
	count	mean	std	min	25%	50%	75%	max	cc
BHK									
1	1167.0	14139.223650	13514.982134	1500.0	6500.0	9000.0	17000.0	200000.0	11
2	2265.0	22113.864018	25803.382742	2000.0	10000.0	15000.0	22000.0	600000.0	22
3	1098.0	55863.062842	117555.074963	1200.0	20000.0	32000.0	65000.0	3500000.0	10
4	189.0	168864.555556	165788.401565	10000.0	60000.0	130000.0	250000.0	1200000.0	1

4 rows × 40 columns

In [69]:
```
#Part 10: How to recognize and Treat missing values and outliers in Pandas?
```

In [70]:
```
# Identify missing values of dataframe
df.isnull()
```

Out[70]:

	BHK	Rent	Size	Area Type	City	Furnishing Status	Bathroom
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
4714	False	False	False	False	False	False	False
4715	False	False	False	False	False	False	False
4716	False	False	False	False	False	False	False
4717	False	False	False	False	False	False	False
4718	False	False	False	False	False	False	False

4719 rows × 7 columns

In [71]:
```
#Part 11: How to merge / join data sets and Pandas dataframes?
```

In [74]:
```
df_new = pd.merge(df, dfr, how = 'inner', left_index = True, right_index = Tru
# By changing how = 'outer', you can do outer join.
# Similarly how = 'left' will do a left join
#You can also specify the columns to join instead of indexes, which are used b
df_new
```

Out[74]:

	BHK_x	Rent_x	Size_x	Area Type_x	City_x	Furnishing Status_x	Bathroom_x	BHK_y	Rent_y	Size_y
2026	2	12000	700	2	3	2	1	2	12000	700
2426	1	9500	600	1	4	2	2	1	9500	600
2808	4	250000	2800	2	4	1	4	4	250000	2800
2996	2	15000	850	1	5	2	1	2	15000	850
4496	3	9000	1100	2	6	2	1	3	9000	1100

In [ ]:
```
```