IBM_Data_Science_Professional_Certificate_-Statistics_for_Data_Science_with_Python

October 23, 2023

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats
```

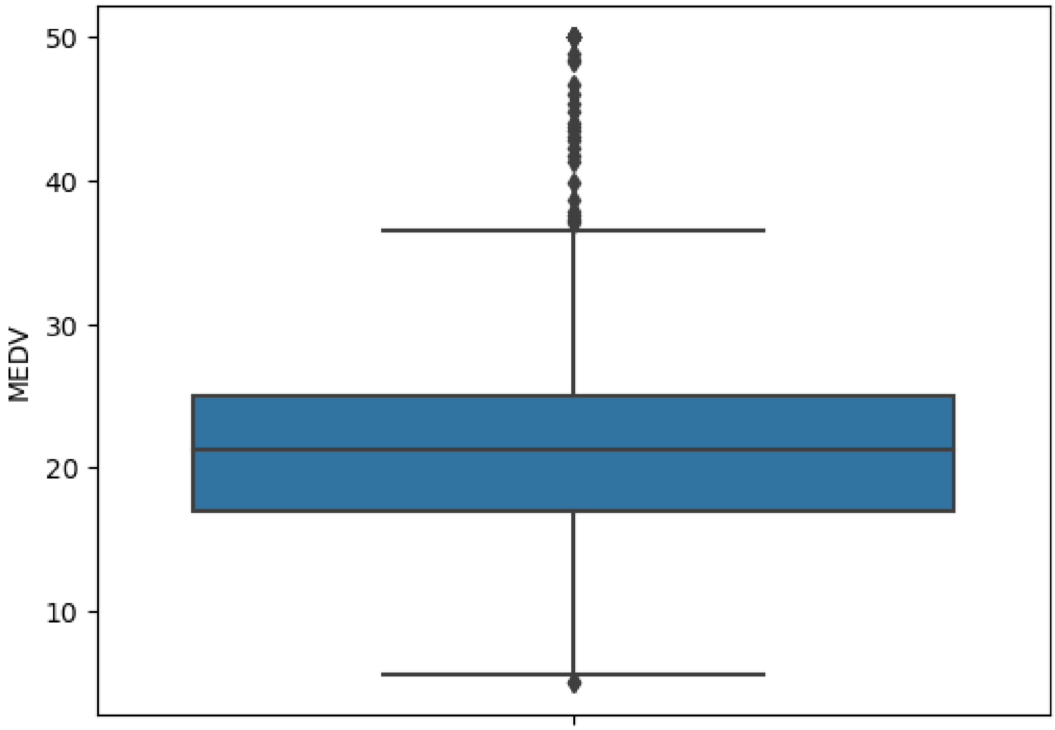
[2]:
```
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

[3]:
```
#TASK4
```

[4]:
```
boston_df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.
 ↪appdomain.cloud/IBMDeveloperSkillsNetwork-ST0151EN-SkillsNetwork/labs/
 ↪boston_housing.csv")
boston_df.head()
```

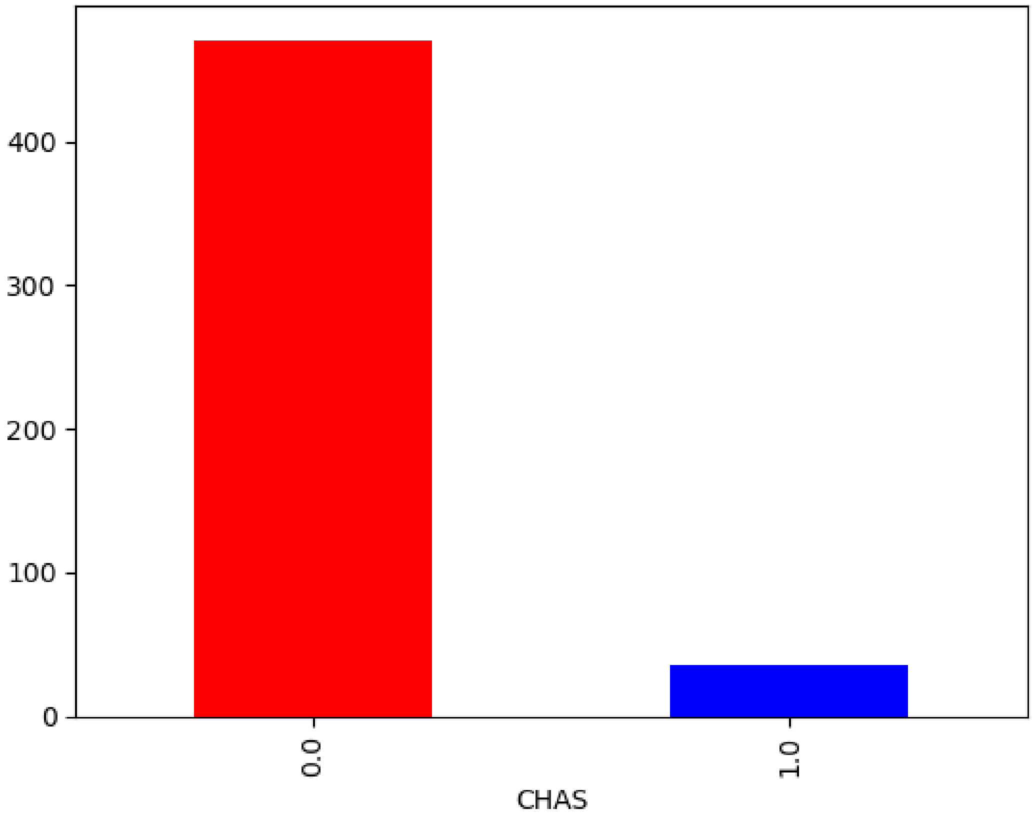
[4]:
```
   Unnamed: 0     CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  \
0           0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900  1.0
1           1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671  2.0
2           2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671  2.0
3           3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622  3.0
4           4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622  3.0

     TAX  PTRATIO  LSTAT  MEDV
0  296.0     15.3   4.98  24.0
1  242.0     17.8   9.14  21.6
2  242.0     17.8   4.03  34.7
3  222.0     18.7   2.94  33.4
4  222.0     18.7   5.33  36.2
```

[5]:
```
#For the "Median value of owner-occupied homes" provide a boxplot:
owner_occupied_homes=sns.boxplot(y="MEDV",data=boston_df)
plt.show()
```



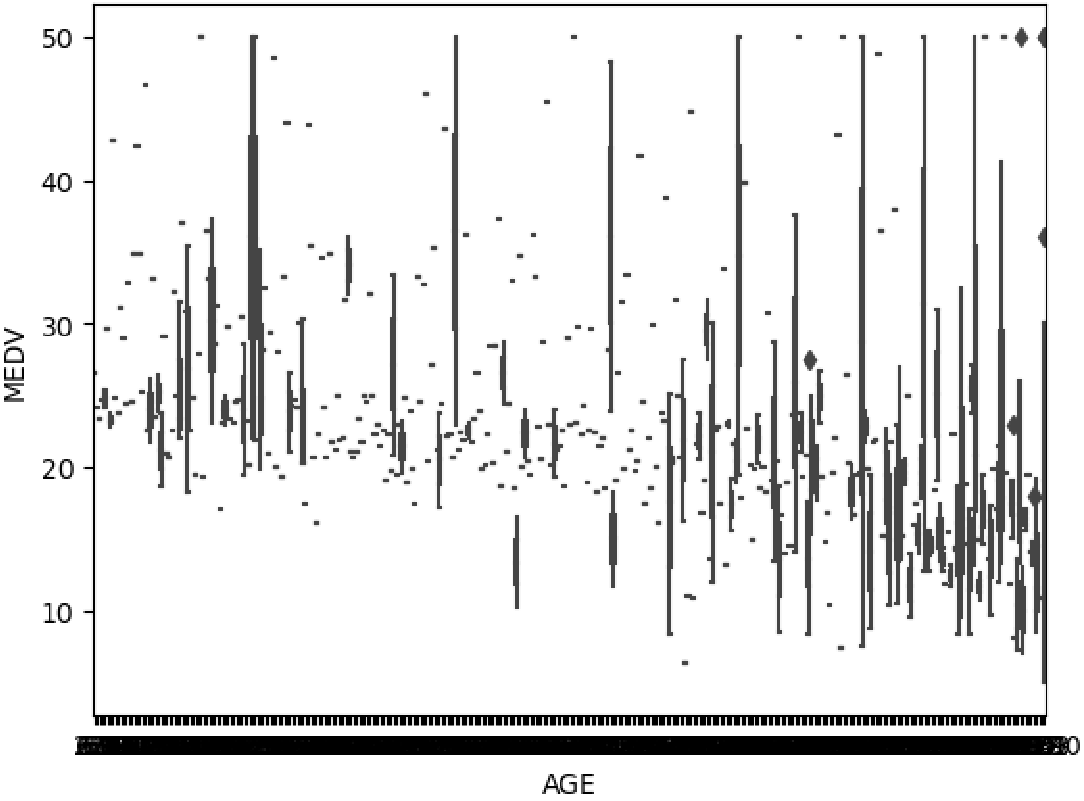
```
[6]: #Provide a  bar plot for the Charles river variable
     boston_df.groupby("CHAS").size().plot(kind="bar",color=["red","blue"])
```

```
[6]: <Axes: xlabel='CHAS'>
```



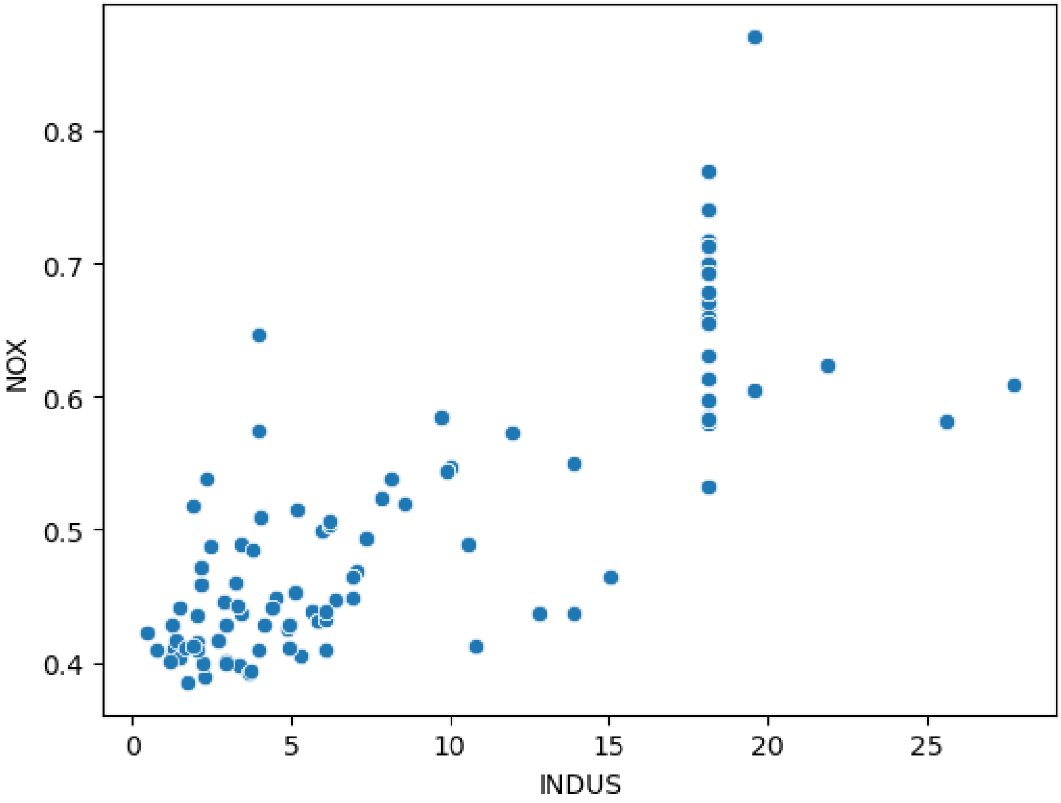
[7]:

```
#Provide a boxplot for the MEDV variable vs the AGE variable.
#(Discretize the age variable into three groups of 35 years and younger,
#between 35 and 70 years and 70 years and older):
Age=sns.boxplot(x="AGE",y="MEDV",data=boston_df)
plt.show()
```



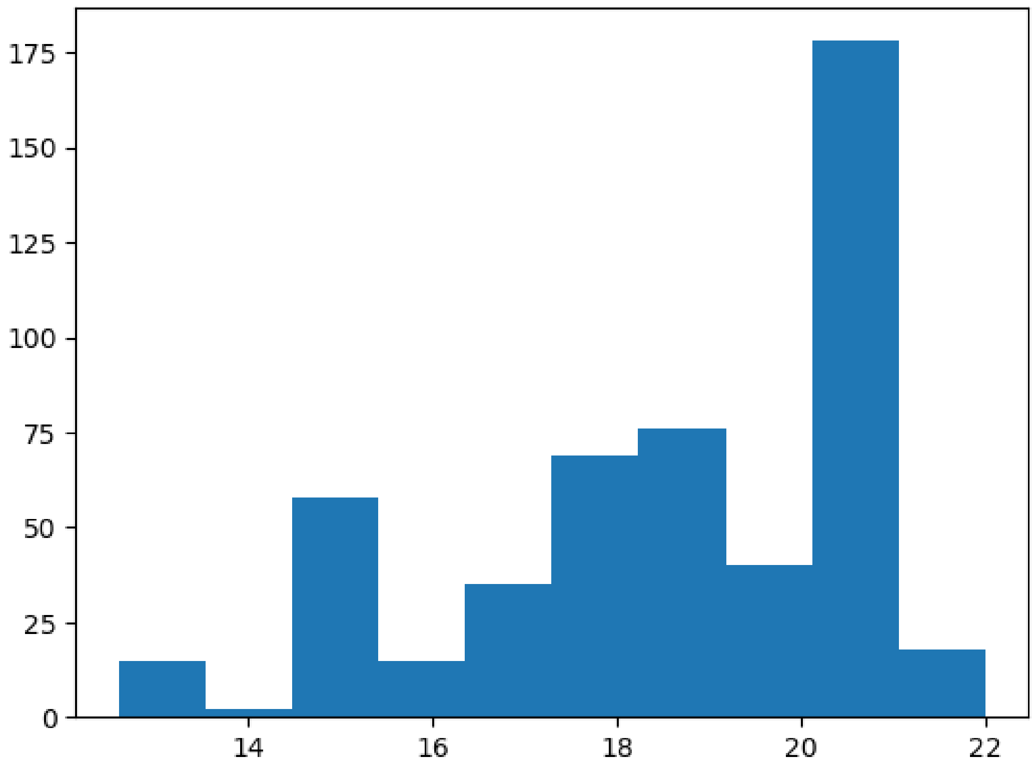
[8]:

```
#Provide a scatter plot to show the relationship between Nitric oxide␣
  ↪concentrations
#and the proportion of non-retail business acres per town. What can you say␣
  ↪about the relationship?
NO=sns.scatterplot(x="INDUS",y="NOX",data=boston_df)
plt.show()
```



```
[9]: #Create a histogram for the pupil to teacher ratio variabl
     plt.hist(boston_df["PTRATIO"])
```

```
[9]: (array([ 15.,   2.,  58.,  15.,  35.,  69.,  76.,  40., 178.,  18.]),
      array([12.6 , 13.54, 14.48, 15.42, 16.36, 17.3 , 18.24, 19.18, 20.12,
             21.06, 22.  ]),
      <BarContainer object of 10 artists>)
```



```
[10]: #TASK 5
```

```
[11]: #Is there a significant difference in median value of houses bounded by the␣
        ↪Charles river or not?
      #(T-test for independent samples)
      #H0: The average values of houses on the river bank and those not on the river␣
        ↪bank are equal to each other.
      #H1: The average values of houses on the river bank and those not on the river␣
        ↪bank are not equal to each other.
      scipy.stats.ttest_ind(boston_df[boston_df["CHAS"]==1.
        ↪0]["MEDV"],boston_df[boston_df["CHAS"]==0.0]["MEDV"],equal_var = True)
      #H0 is rejected because the alpha value (0.05) is greater than the p value;␣
        ↪that is, there is a difference between
      #the two average values.
```

```
[11]: Ttest_indResult(statistic=3.996437466090509, pvalue=7.390623170519905e-05)
```

```
[12]: #Is there a difference in Median values of houses (MEDV) for each proportion of␣
        ↪owner occupied
      #units built prior to 1940 (AGE)? (ANOVA)
      #H0: MEDV average values for the three age groups are equal to each other.
      #H1: At least one of the MEDV values is different for three age groups.
```

```
boston_df.loc[(boston_df['AGE'] <= 35), 'age_group'] = '35 years and younger'
boston_df.loc[(boston_df['AGE'] > 35)&(boston_df['AGE'] < 70), 'age_group'] =␣
 ↪'between 35 and 70 years'
boston_df.loc[(boston_df['AGE'] >= 57), 'age_group'] = '70 years and older'
```

[13]:

```
#Is there a difference in Median values of houses (MEDV) for each proportion of␣
 ↪owner occupied
#units built prior to 1940 (AGE)? (ANOVA)

thirtyfive_lower=boston_df[boston_df['age_group'] == '35 years and␣
 ↪younger']['MEDV']
thirtyfive_seventy= boston_df[boston_df['age_group'] == 'between 35 and 70␣
 ↪years']['MEDV']
seventy_older= boston_df[boston_df['age_group'] == '70 years and older']['MEDV']

scipy.stats.f_oneway(thirtyfive_lower,thirtyfive_seventy,seventy_older)

#Since the alpha value (0.05) is greater than p, H0 is rejected; That is, at␣
 ↪least one of the
#average values is different from the others.
```

[13]:

```
F_onewayResult(statistic=29.028583208977576, pvalue=1.1723548056383608e-12)
```

[14]:

```
#Can we conclude that there is no relationship between Nitric oxide␣
 ↪concentrations and proportion of
#non-retail business acres per town? (Pearson Correlation)
#H0: there is no relationship between the two data
#H1: There is a relationship between two data.

scipy.stats.pearsonr(boston_df['NOX'], boston_df['INDUS'])

#Since the alpha value (0.05) is greater than the p value, the H0 hypothesis is␣
 ↪rejected;
#That is, there is a relationship between two data.
```

[14]:

```
PearsonRResult(statistic=0.7636514469209149, pvalue=7.913361061242812e-98)
```

[15]:

```
#What is the impact of an additional weighted distance  to the five Boston␣
 ↪employment centres on the
#median value of owner occupied homes? (Regression analysis)
#H0:beta1 equal to zero (DIS has no effect on MEDV)
#H1:beta1 not equal to zero(DIS has an effect on MEDV)

y=boston_df["MEDV"]
x=boston_df["DIS"]
x=sm.add_constant(x)
```

```python
model=sm.OLS(y,x).fit()
predictions=model.predict(x)
model.summary()

#Since alpha(0.05) is greater than the p value, H0 is rejected; i.e. DIS has an impact on MEDV
```

[15]:

Dep. Variable:	MEDV	**R-squared:**	0.062
Model:	OLS	**Adj. R-squared:**	0.061
Method:	Least Squares	**F-statistic:**	33.58
Date:	Mon, 23 Oct 2023	**Prob (F-statistic):**	1.21e-08
Time:	11:11:33	**Log-Likelihood:**	-1823.9
No. Observations:	506	**AIC:**	3652.
Df Residuals:	504	**BIC:**	3660.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
const	18.3901	0.817	22.499	0.000	16.784	19.996
DIS	1.0916	0.188	5.795	0.000	0.722	1.462

Omnibus:	139.779	**Durbin-Watson:**	0.570
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	305.104
Skew:	1.466	**Prob(JB):**	5.59e-67
Kurtosis:	5.424	**Cond. No.**	9.32

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[ ]:

[ ]: