python_Basic_Infrential_Analysys

June 12, 2021

1 Running Basic Inferential Analyses

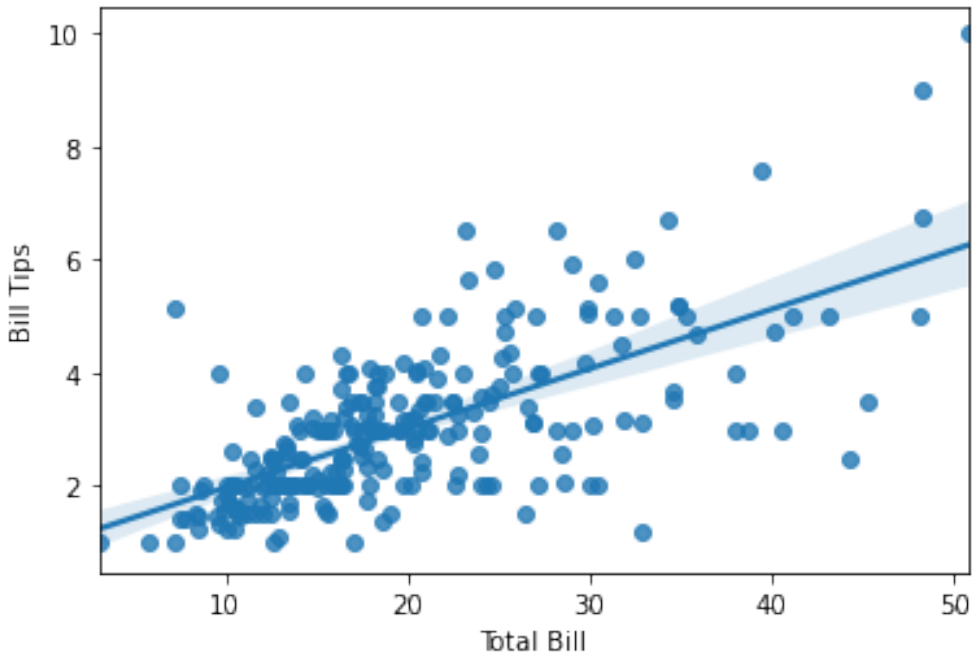
[2]:
```
# Statistical analysis can be done using Pandas, SciPy, and Numpy
#The following operations can be perfomed

#• Linear regression
#• Finding correlation
#• Measuring central tendency
#• Measuring variance
#• Normal distribution
#• Binomial distribution
#• Poisson distribution
#• Bernoulli distribution
#• Calculating p-value
#• Implementing a Chi-square test
```

1.1 linear regression

[4]:
```
#Linear regression between two variables represents a straight line when␣
↪plotted as a graph, where the exponent (power) of both of the variables is 1.
↪ A nonlinear relationship where the exponent of any variable is not equal to␣
↪1 creates a curve shape.
```

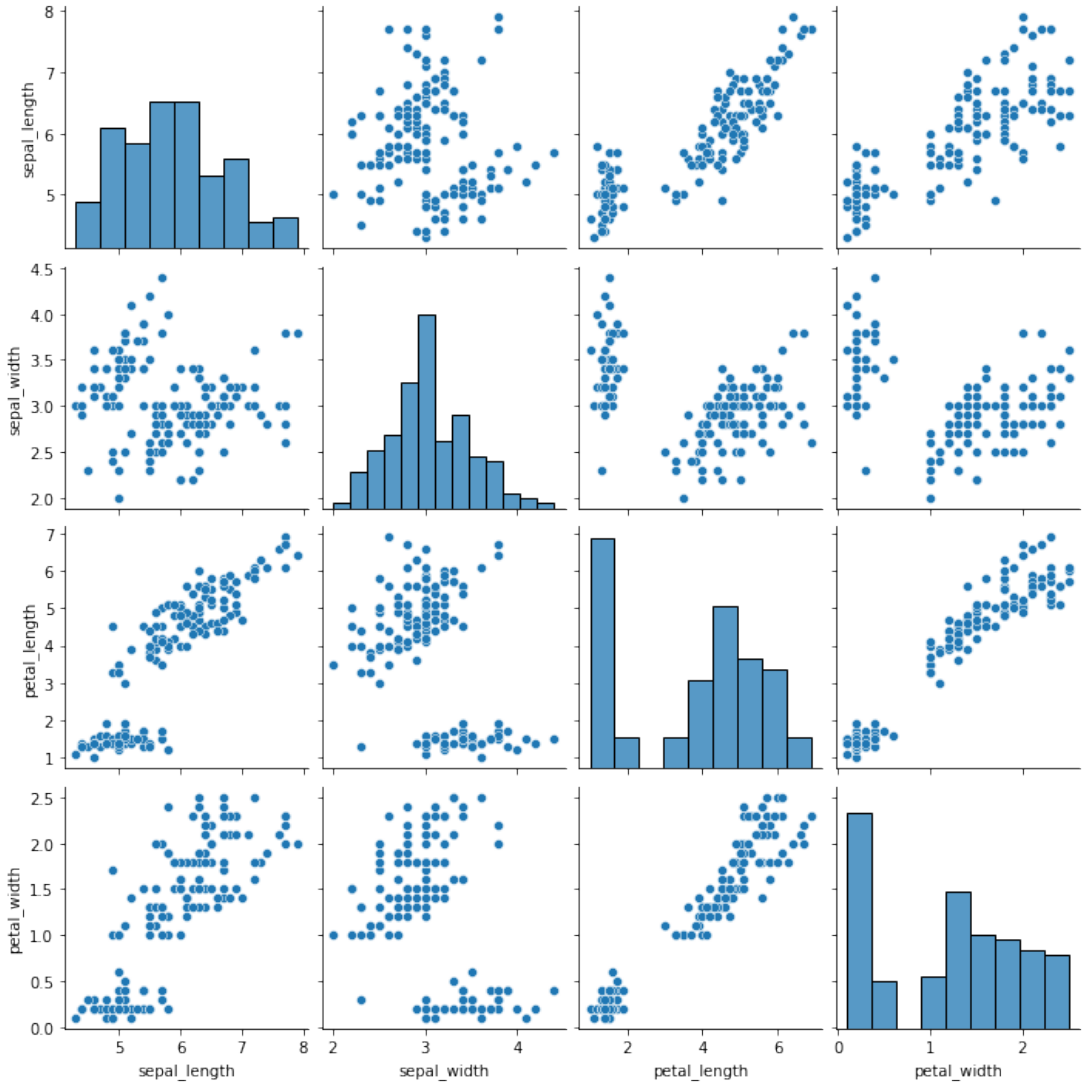
[3]:
```
#using tips inbuilt database of the Seaborn Python Library
import seaborn as sb
from matplotlib import pyplot as plt
df = sb.load_dataset('tips')
sb.regplot(x = "total_bill", y = "tip", data = df)
plt.xlabel('Total Bill')
plt.ylabel('Bill Tips')
plt.show()
```



1.2 Correlation

[ ]:
```
#refers to some statistical relationship involving dependence between two data␣
↪sets, such as the correlation between the price of a product and its sales␣
↪volume.
```

[5]:
```
#using the inbuilt iris dataset in Seaborn Python Library
import matplotlib.pyplot as plt
import seaborn as sns
df = sns.load_dataset('iris')
sns.pairplot(df, kind="scatter")
plt.show()
```



1.3 variance

[ ]:
```
#is a measure of how dispersed the values are from the mean value. Standard␣
↪deviation is the square root of variance. In other words, it is the average␣
↪of the squared difference of values in a data set from the mean value.
```

[6]:
```
import pandas as pd
d= {
'Name': pd.Series(['Ahmed','Omar','Ali','Salwa','Majid',
 'Othman','Gameel','Ziad','Ahlam','Zahrah',
 'Ayman','Alaa']),
'Age': pd.Series([34,26,25,27,30,54,23,43,40,30,28,46]),
'Height':pd.Series([114.23,173.24,153.98,172.0,153.20,164.6,
```

```
 183.8,163.78,172.0,164.8 ])}
df = pd.DataFrame(d) #Create a DataFrame
print (df.std()) # Calculate and print the standard deviation
```

```
Age         9.740574
Height     18.977090
dtype: float64
```

[7]:
```
print (df.describe())
```

```
              Age      Height
count   12.000000   10.00000
mean    33.833333  161.56300
std      9.740574   18.97709
min     23.000000  114.23000
25%     26.750000  156.43000
50%     30.000000  164.70000
75%     40.750000  172.00000
max     54.000000  183.80000
```

1.4 Central tendency

[ ]:
```
#measures the distribution of the location of values of a data set. It gives␣
 ↪you an idea of the average value of the data in the data set and an␣
 ↪indication of how widely the values are spread in the data set.
```

[8]:
```
print ("Mean Values in the Distribution")
print (df.mean())
print ("*******************************")
print ("Median Values in the Distribution")
print (df.median())
print ("*******************************")
print ("Mode Values in the Distribution")
print (df['Height'].mode())
```

```
Mean Values in the Distribution
Age         33.833333
Height     161.563000
dtype: float64
*******************************
Median Values in the Distribution
Age         30.0
Height     164.7
dtype: float64
*******************************
Mode Values in the Distribution
0    172.0
dtype: float64
```

[ ]: