3110_DS

October 20, 2022

1 Internship on Data Science at InfraBIM

DS-01: Assignment - Descriptive Analysis on Interns Past Academic Performance

---

Team No.: Reg. No.: 3110 Name: Omkar Baburao Mane Date: 19-10-2022

[1]:
```
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
#Loading the csv file into  the dataframe
from google.colab import files # Use to load data on Google Colab
uploaded = files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving Enrollments_28092022.csv to Enrollments_28092022.csv
```

[3]:
```
df = pd.read_csv('Enrollments_28092022.csv')
df
```

[3]:

	StudentNo	DEGREE	INTERMEDIATE	SSC	INTERNSHIP
0	1001	8.10	76.0	92.0	Data Science
1	1002	8.10	76.0	92.0	MEAN Stack Web Development
2	1003	7.80	94.6	92.0	MEAN Stack Web Development
3	1004	9.03	89.5	89.0	Data Science
4	1005	8.38	87.0	90.0	MEAN Stack Web Development
..	…	…	…	…	…
292	2188	8.70	94.1	93.0	Data Science
293	2189	8.45	90.0	93.0	Data Science
294	2190	8.40	94.9	98.0	Data Science
295	2191	7.06	90.6	88.0	Cloud Computing Services (AWS)
296	2192	7.50	95.5	95.0	Cloud Computing Services (AWS)

[297 rows x 5 columns]

Q1) Identify Variables and their Types (Quantitative or Qualitative)

[4]:
```
#Using info function
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 297 entries, 0 to 296
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   StudentNo     297 non-null    int64
 1   DEGREE        297 non-null    float64
 2   INTERMEDIATE  297 non-null    float64
 3   SSC           297 non-null    float64
 4   INTERNSHIP    297 non-null    object
dtypes: float64(3), int64(1), object(1)
memory usage: 11.7+ KB
```

[5]:
```
df['StudentNo'] = df['StudentNo'].apply(str)
```

[6]:
```
#Using describe function
df.describe()
```

[6]:

	DEGREE	INTERMEDIATE	SSC
count	297.000000	297.000000	297.000000
mean	7.928081	88.662626	88.106734
std	0.785579	7.355733	9.027984
min	5.800000	65.000000	38.400000
25%	7.400000	83.000000	85.000000
50%	8.000000	90.800000	90.000000
75%	8.560000	94.600000	95.000000
max	9.530000	99.400000	99.000000

Q1. Answer Categorical Data:StudentNo Numerical Data:Degree, Intermediate, Ssc

---

Q2. Size of Data (No. of Rows and Columns)

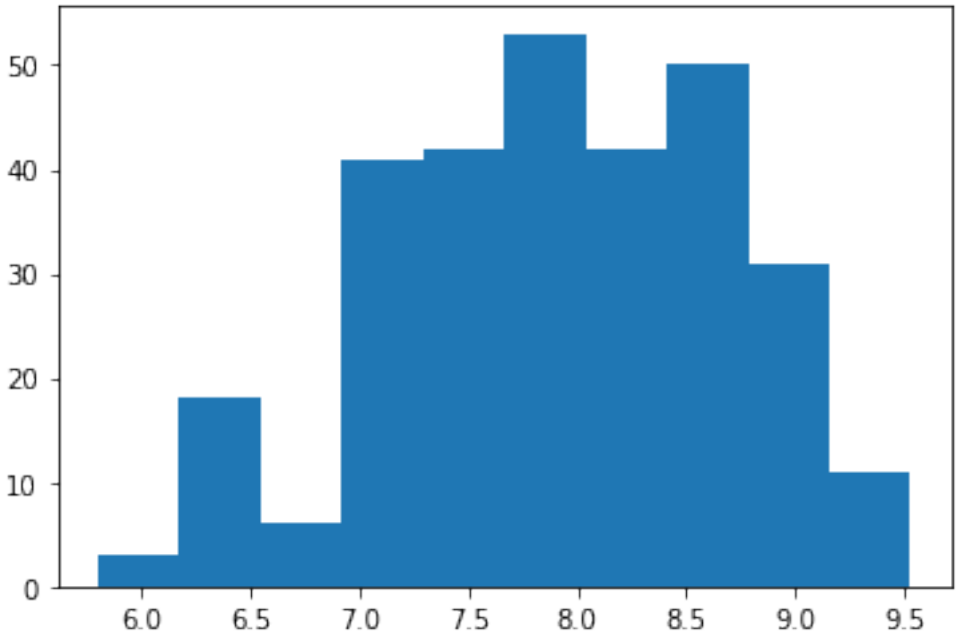
[7]:
```
#using shape function
df.shape
```

[7]:
```
(297, 5)
```

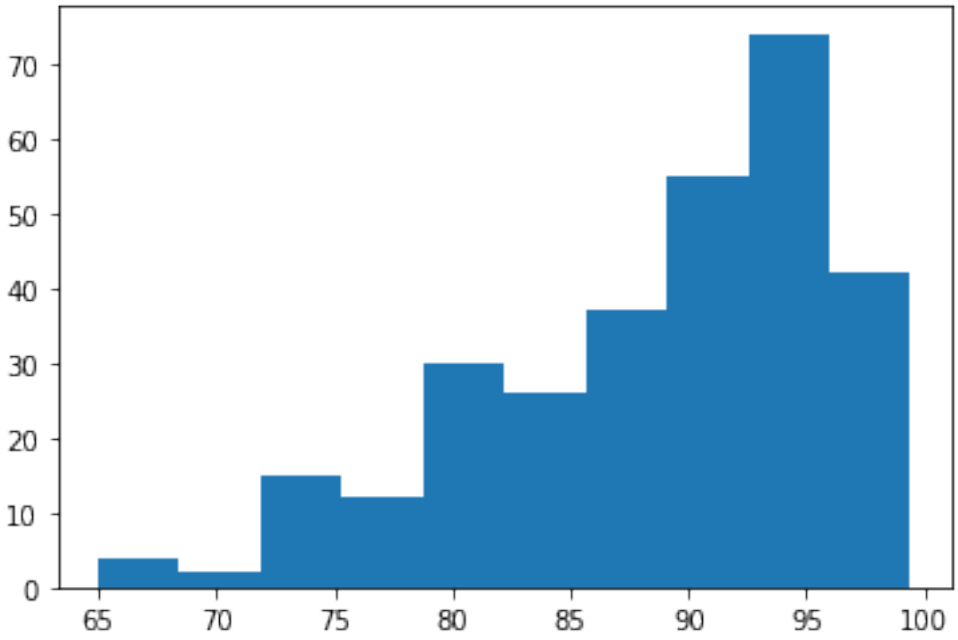
Q2. Answer Rows: 297, Attributes: 5

Q3. Create Histogram

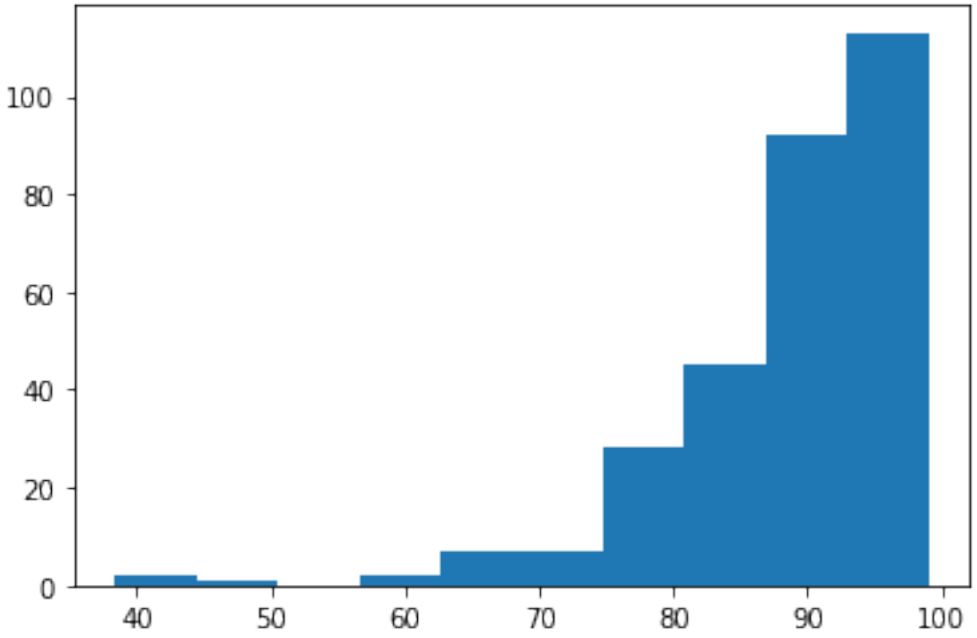
[8]:
```
# generating histogram using hist function from matplotlib library
plt.hist(df['DEGREE'])
plt.show()
```



[9]:
```
plt.hist(df['INTERMEDIATE'])
plt.show()
```

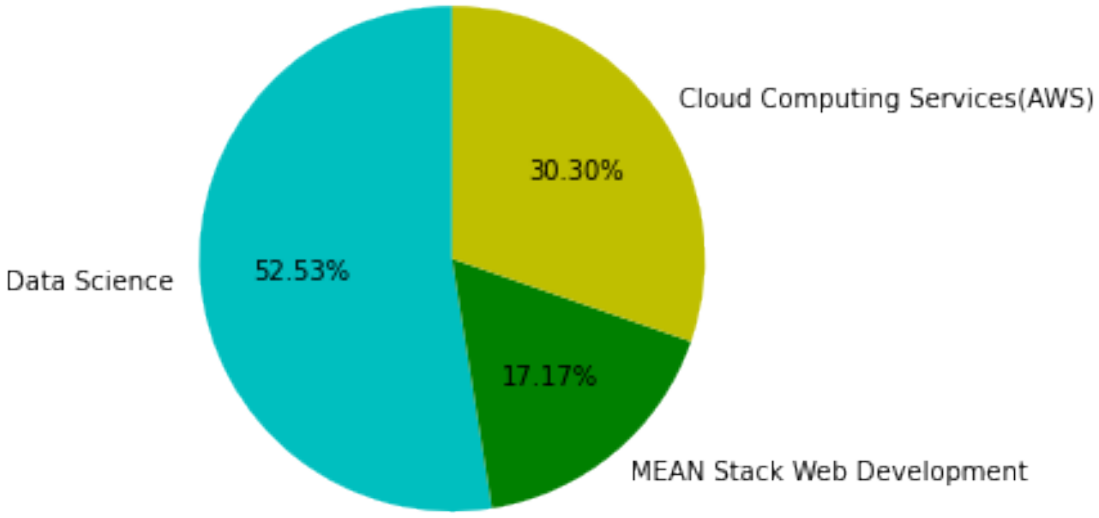


[10]:
```
plt.hist(df['SSC'])
plt.show()
```



Q4. Create Pie-Chart to represent the Enrollments for each Internship Program

[11]:
```
courses = ['Data Science','MEAN Stack Web Development ','Cloud Computing␣
 ↪Services(AWS)']
students= [156,51,90]
colors = ['c','g','y']
plt.
 ↪pie(students,labels=courses,colors=colors,startangle=90,explode=(0,0,0),autopct␣
 ↪= '%1.2f%%')
plt.axis('equal')
plt.show()
```



Q5. Find No. of Enrollments for each Internship Program

[12]:
```
df['INTERNSHIP'].value_counts()
```

[12]:
```
Data Science                       156
Cloud Computing Services (AWS)      90
MEAN Stack Web Development          51
Name: INTERNSHIP, dtype: int64
```

Q6. Find Measure of Central Tendency: MEAN, MEDIAN, MODE

[13]:
```
#finding the MEAN
print(df.mean(numeric_only= True))
```

```
DEGREE           7.928081
INTERMEDIATE    88.662626
SSC             88.106734
dtype: float64
```

[14]:
```
#finding the MEDIAN
print(df.median(numeric_only= True))
```

```
DEGREE           8.0
INTERMEDIATE    90.8
SSC             90.0
dtype: float64
```

[15]:
```
#finding the MODE
print(df.mode(numeric_only= True))
```

```
   DEGREE  INTERMEDIATE   SSC
0     7.0          95.0  95.0
```

Q7. Find Measure of Variance: Minimum, Maximum, Range, Mean Deviation, Standard Deviation, Co-efficient of Variation

[16]:
```
# Measure of Variance : Minimum
print(df.min(numeric_only= True))
```

```
DEGREE           5.8
INTERMEDIATE    65.0
SSC             38.4
dtype: float64
```

[17]:
```
# Measure of Variance : Maximum
print(df.max(numeric_only= True))
```

```
DEGREE           9.53
INTERMEDIATE    99.40
SSC             99.00
dtype: float64
```

[18]:
```
# Measure of Variance : Range
print(df.max(numeric_only= True)-df.min(numeric_only= True))
```

```
DEGREE           3.73
INTERMEDIATE    34.40
SSC             60.60
dtype: float64
```

[19]:
```
#Measure of Variance : Standard Deviation
print(df.std(numeric_only= True))
```

```
DEGREE          0.785579
INTERMEDIATE    7.355733
SSC             9.027984
dtype: float64
```

[20]:
```
# Co-effienct of Variation
print(df.std(numeric_only= True)/df.mean(numeric_only= True))
```

```
DEGREE          0.099088
INTERMEDIATE    0.082963
```

```
SSC             0.102466
dtype: float64
```

Q8. Measures of Position: Standard Scores, Inter-quartile Range for Degree, Inter and 10th

[21]:
```
#Frist Quartile
print(df.quantile(q=0.25, numeric_only= True))
```

```
DEGREE           7.4
INTERMEDIATE    83.0
SSC             85.0
Name: 0.25, dtype: float64
```

[22]:
```
#Second Quartile or Median
print(df.quantile(q=0.5, numeric_only= True))
```

```
DEGREE           8.0
INTERMEDIATE    90.8
SSC             90.0
Name: 0.5, dtype: float64
```

[23]:
```
#Third Quartile
print(df.quantile(q=0.75, numeric_only= True))
```

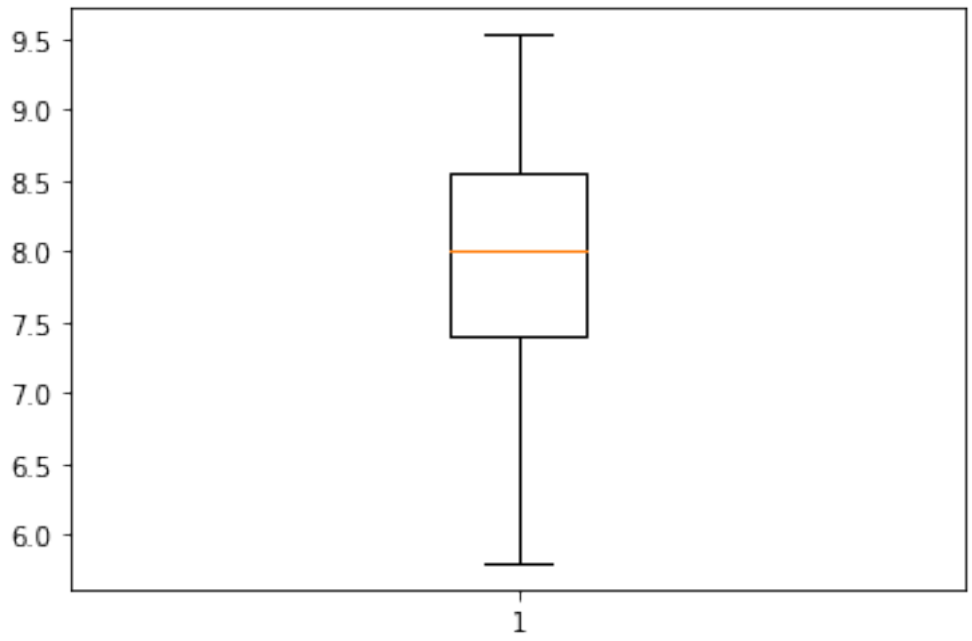
```
DEGREE           8.56
INTERMEDIATE    94.60
SSC             95.00
Name: 0.75, dtype: float64
```

[24]:
```
# Inter-Quartile = Q3 - Q1
a= df.quantile(q=0.75, numeric_only= True)-df.quantile(q=0.25, numeric_only=␣
 ↪True)
a
```

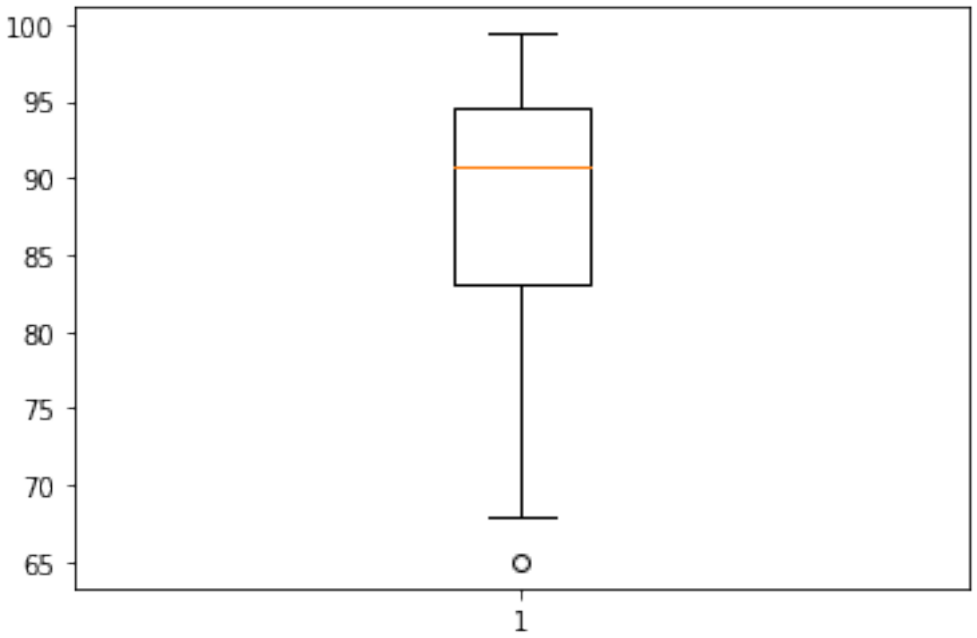
[24]:
```
DEGREE           1.16
INTERMEDIATE    11.60
SSC             10.00
dtype: float64
```

Q9. Create Box Plot and Identify Outliers

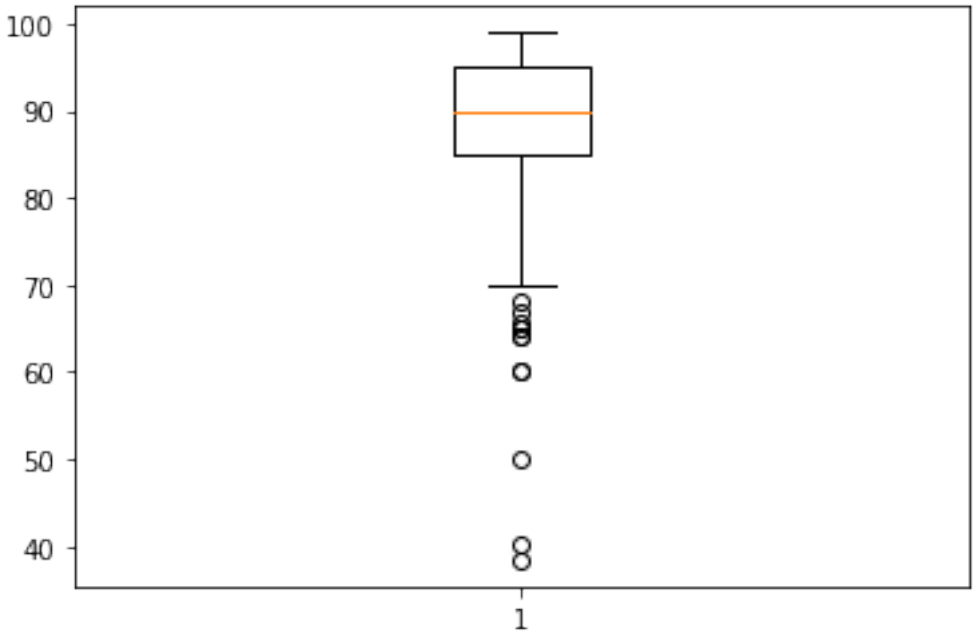
[25]:
```
plt.boxplot(df['DEGREE'])
plt.show()
```



[26]:
```
plt.boxplot(df['INTERMEDIATE'])
plt.show()
```



[27]:
```
plt.boxplot(df['SSC'])
plt.show()
```



Q10. Identify No. of Students with 90% percentile for Degree, Inter and 10th Class

[28]:
```
# 90th Percentile or Quantile
def outlier(a):
  q1 = np.quantile(a,0.25)
  q3 = np.quantile(a,0.75)
  med = np.median(a)
  iqr = q3-q1
  upper_bound = q3+(1.5*iqr)
  lower_bound = q1-(1.5*iqr)
  print(iqr,upper_bound,lower_bound)
  print("Inter Quartile Range:",iqr)
  outliers = a[(a<= lower_bound) |(a>= upper_bound) ]
  print("the following are the outliers in boxplot:\n{}".format(outliers))
```

[29]:
```
outlier(df['DEGREE'])
```

```
1.1600000000000001 10.3 5.66
Inter Quartile Range: 1.1600000000000001
the following are the outliers in boxplot:
Series([], Name: DEGREE, dtype: float64)
```

[30]:
```
outlier(df['INTERMEDIATE'])
```

```
11.599999999999994 111.99999999999999 65.60000000000001
Inter Quartile Range: 11.599999999999994
the following are the outliers in boxplot:
271    65.0
Name: INTERMEDIATE, dtype: float64
```

[31]:
```
outlier(df['SSC'])
```

```
10.0 110.0 70.0
Inter Quartile Range: 10.0
the following are the outliers in boxplot:
5      64.0
7      70.0
31     60.0
51     68.0
69     60.0
82     65.6
86     50.0
107    64.0
236    38.4
237    67.0
243    40.2
270    65.0
288    65.0
Name: SSC, dtype: float64
```

[32]:
```
def func(b):
  q9 = np.quantile(b,0.9)
  li = b[b==q9]
  print("no.of students with 90% percentile:",li.count())
```

[33]:
```
func(df['DEGREE'])
```

```
no.of students with 90% percentile: 3
```

[34]:
```
func(df['INTERMEDIATE'])
```

```
no.of students with 90% percentile: 3
```

[35]:
```
func(df['SSC'])
```

```
no.of students with 90% percentile: 19
```
