Practical No:3

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
```

In [3]:
```
df=pd.read_csv("D:\DSBDA/Placement_Data.csv")
```

In [4]:
```
df
```

Out[4]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degre
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&T
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&T
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&M
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&T
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&M
...	...	...	...	...	...	...	...	...	
210	211	M	80.60	Others	82.00	Others	Commerce	77.60	Comm&M
211	212	M	58.00	Others	60.00	Others	Science	72.00	Sci&T
212	213	M	67.00	Others	67.00	Others	Commerce	73.00	Comm&M
213	214	F	74.00	Others	66.00	Others	Commerce	58.00	Comm&M
214	215	M		Central	58.00	Others	Science	53.00	Comm&M

62.00 mns

215 rows × 15 colu

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----0
 sl_no           215 non-null    int64
  1 gender          215 non-null    object
  2 ssc_p           215 non-null    float64
  3 ssc_b           215 non-null    object
  4 hsc_p           215 non-null    float64
```

```
 5 hsc_b           215 non-null    object
 6 hsc_s           215 non-null    object
 7 degree_p        215 non-null    float64
 8 degree_t        215 non-null    object
 9 workex          215 non-null    object
 10      etest_p         215 non-null    float64
 11      specialisation  215 non-null    object
 12      mba_p           215 non-null    float64
 13      status          215 non-null    object
 14      salary          148 non-null    float64 dtypes: float64(6),
int64(1), object(8) memory usage: 25.3+ KB
```

In [6]:
```
df.shape
```

Out[6]: (215,

15)

In [7]:
```
df.describe
```

Out[7]:

```
<bound method NDFrame.describe of      sl_no gender  ssc_p    ssc_b  hsc_p
hsc_b     hsc_s  degree_p  \
0         1      M  67.00   Others  91.00   Others  Commerce     58.00
1         2      M  79.33  Central  78.33   Others   Science     77.48
2         3      M  65.00  Central  68.00  Central      Arts     64.00
3         4      M  56.00  Central  52.00  Central   Science     52.00
4         5      M  85.80  Central  73.60  Central  Commerce     73.30 ..
          ...    ...    ...      ...    ...      ...       ...       ...
210     211      M  80.60   Others  82.00   Others  Commerce     77.60
211     212      M  58.00   Others  60.00   Others   Science     72.00
212     213      M  67.00   Others  67.00   Others  Commerce     73.00
213     214      F  74.00   Others  66.00   Others  Commerce     58.00
214     215      M  62.00  Central  58.00   Others   Science     53.00

     degree_t workex  etest_p specialisation  mba_p      status    salary
0     Sci&Tech     No     55.0         Mkt&HR  58.80      Placed  270000.0
1     Sci&Tech    Yes     86.5        Mkt&Fin  66.28      Placed  200000.0
2     Comm&Mgmt     No     75.0        Mkt&Fin  57.80      Placed
      250000.0
3     Sci&Tech     No     66.0         Mkt&HR  59.43  Not Placed       NaN
4     Comm&Mgmt     No     96.8        Mkt&Fin  55.50      Placed
      425000.0 ..          ...    ...      ...            ...    ...
      ...       ...
210   Comm&Mgmt     No     91.0        Mkt&Fin  74.49      Placed
      400000.0
211   Sci&Tech     No     74.0        Mkt&Fin  53.62      Placed  275000.0
212   Comm&Mgmt    Yes     59.0        Mkt&Fin  69.72      Placed
      295000.0
213   Comm&Mgmt     No     70.0         Mkt&HR  60.23      Placed
      204000.0
214   Comm&Mgmt     No     89.0         Mkt&HR  60.22  Not Placed
      NaN
```

```
[215 rows x 15 columns]>
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
sl_no             0
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
hsc_b             0
hsc_s             0
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
salary           67
dtype: int64
```

In [33]:
```
df['salary'].mean()
```

Out[33]:
```
np.float64(288655.40540540544)
```

In [35]:
```
df['ssc_p'].median()
```

Out[35]:
```
67.0
```

In [36]:
```
df['salary'] = df['salary'].fillna(df['salary'].mean())
```

In [37]:

Out[37]:
```
df.head()
```

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_sdegree_p	degree_
0	1	1	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tec
1	2	1	79.33	Central	78.33	Others	Science	77.48	Sci&Tec
2	3	1	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgm
3	4	1	56.00	Central	52.00	Central	Science	52.00	Sci&Tec
4	5	1	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgm

In [38]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['gender'] = le.fit_transform(df['gender'])
```

In [39]:

In [40]:
```
df['status'] = le.fit_transform(df['status'])
```

In [41]:
```
df['workex'] = le.fit_transform(df['workex'])
```

Out[41]:

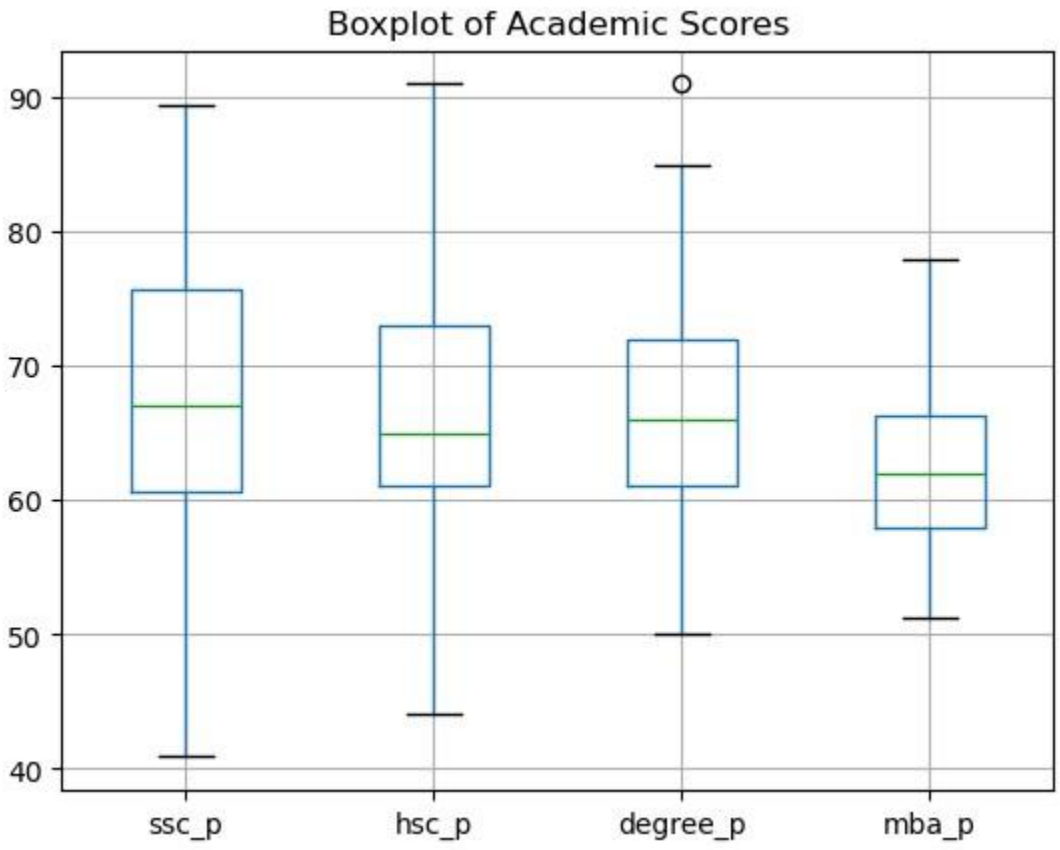
```
df['specialisation'] = le.fit_transform(df['specialisation'])
df.head()
```

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_
0	1	1	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tec

```
df.boxplot(column=['ssc_p','hsc_p','degree_p','mba_p'])
plt.title("Boxplot of Academic Scores")
plt.show()
```

1	2	1	79.33	Central	78.33	Others	Science	77.48	Sci&Tec
2	3	1	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgm
3	4	1	56.00	Central	52.00	Central	Science	52.00	Sci&Tec
4	5	1	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgm

In [42]:



In [43]:
```
z = np.abs(stats.zscore(df['mba_p']))
df[z > 3]
```

Out[43]:

sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex

In [44]:
```
Q1 = df['hsc_p'].quantile(0.25)
Q3 = df['hsc_p'].quantile(0.75)
IQR = Q3 - Q1 lower =
Q1 - 1.5 * IQR upper
= Q3 + 1.5 * IQR
print("Lower Bound:", lower)
print("Upper Bound:", upper)
```

```
Lower Bound: 43.0
Upper Bound: 91.0
```

In [45]:
```
df[(df['hsc_p'] < lower) | (df['hsc_p'] > upper)]
```

Out[45]:

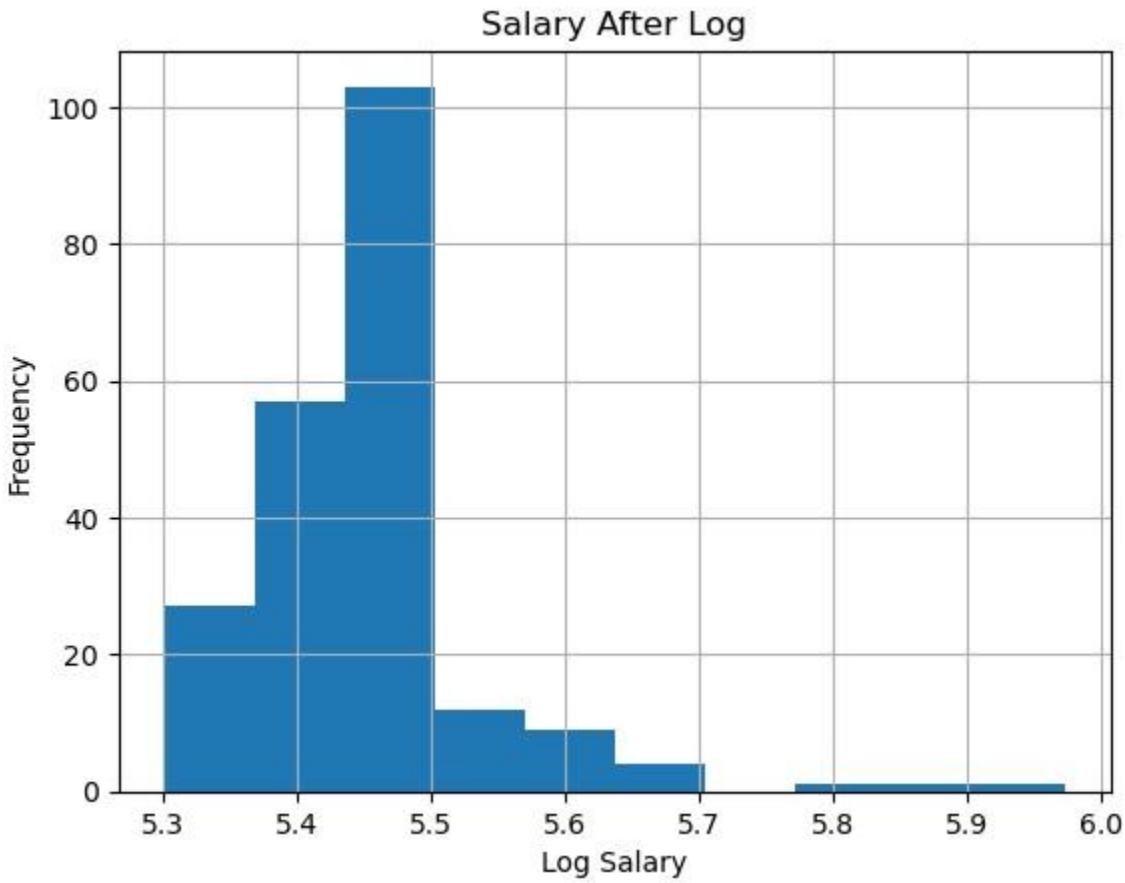
sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex

In [46]:
```
median = df['hsc_p'].median()
df['hsc_p'] = np.where((df['hsc_p'] < lower) | (df['hsc_p'] > upper),median,df
```

In [47]:
```
df['salary'].hist()
plt.title("Salary Before Log")
plt.xlabel("Salary")
plt.ylabel("Frequency")
plt.show()
df['salary'].skew()
```

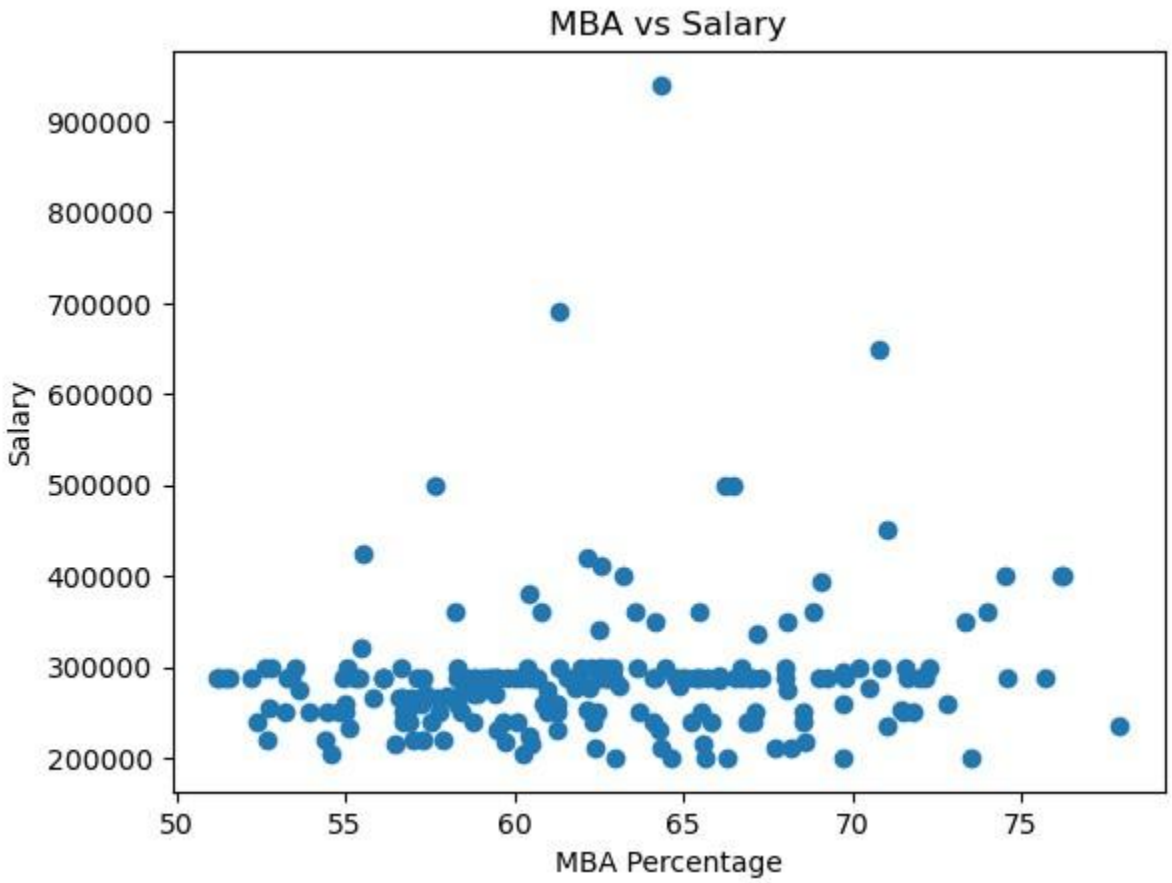
Out[47]: np.float64(4.288798850070048)

In [48]:
```
df['log_salary'] =
np.log10(df['salary'])
df['log_salary'].hist()
plt.title("Salary After Log")
plt.xlabel("Log Salary")
plt.ylabel("Frequency")
plt.show()
df['log_salary'].skew()
```



Out[48]: np.float64(1.9500125921488516)

In [49]:
```
plt.scatter(df['mba_p'], df['salary'])
plt.xlabel("MBA Percentage")
plt.ylabel("Salary")
plt.title("MBA vs Salary")
plt.show()
```



In [54]:
```
mad = (df['salary'] - df['salary'].mean()).abs().mean()
print("Mean Absolute Deviation:", mad)
```

```
Mean Absolute Deviation: 39441.1062225016
```

In [55]:
```
variance = df['salary'].var()
print("Variance:", variance)
```

```
Variance: 5999726288.204086
```

In [56]:
```
std_dev = df['salary'].std()
print("Standard Deviation:", std_dev)
```

```
Standard Deviation: 77457.90010195272
```

In [57]:
```
data_range = df['salary'].max() - df['salary'].min()
print("Range:", data_range)
```

```
Range: 740000.0
```

In [ ]:
```

```