In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:
```
df = pd.read_csv('Emp_EDA.csv')
df.head(10)
```

Out[3]:

	First Name	Gender	Salary	Team	Age	Experience	New_Salary	Bonus	Senior Management
0	Maria	Female	130590	Finance	NaN	5	146075.36220	20000	False
1	Angela	Female	54568	Business Development	27.0	5	64675.63064	19000	True
2	Allan	Male	125792	Client Services	28.0	6	132134.43260	18500	False
3	Rohan	Female	45906	Finance	28.0	7	51230.17788	18000	True
4	Douglas	Male	97308	Marketing	28.0	7	104066.04060	17000	True
5	Brandon	Male	112807	Human Resources	30.0	8	132539.20040	16000	True
6	Diana	Female	132940	Client Services	31.0	9	158307.61080	15800	False
7	Frances	NaN	139852	Business Development	34.0	10	150374.46450	15500	True
8	Matthew	Male	100612	Marketing	34.0	10	114340.50740	15000	False
9	Larry	Male	101004	Client Services	35.0	11	102406.94560	14700	True

1. Remove the irrelevant column 'Senior Management' and display top 5 rows (use inplace=True)

In [4]:
```
df.drop('Senior Management', axis=1, inplace=True)
top_5_rows = df.head(5)
print(top_5_rows)
```

```
  First Name  Gender  Salary                Team   Age  Experience  \
0      Maria  Female  130590             Finance   NaN           5
1     Angela  Female   54568  Business Development  27.0           5
2      Allan    Male  125792      Client Services  28.0           6
3      Rohan  Female   45906             Finance  28.0           7
4    Douglas    Male   97308           Marketing  28.0           7

     New_Salary  Bonus
0  146075.36220  20000
1   64675.63064  19000
2  132134.43260  18500
3   51230.17788  18000
4  104066.04060  17000
```

2. Remove the duplicate rows and display the shape of the dataframe (use inplace=True)

In [5]:
```
df.drop_duplicates(inplace=True)
df.shape
```

Out[5]:
```
(24, 8)
```

3) Rename the column 'Bonus' to 'Incentive' and display top 5 rows (use inplace=True)

In [6]:
```
df.rename(columns={'Bonus': 'Incentive'}, inplace=True)
df.head(5)
```

Out[6]:

	First Name	Gender	Salary	Team	Age	Experience	New_Salary	Incentive
0	Maria	Female	130590	Finance	NaN	5	146075.36220	20000
1	Angela	Female	54568	Business Development	27.0	5	64675.63064	19000
2	Allan	Male	125792	Client Services	28.0	6	132134.43260	18500
3	Rohan	Female	45906	Finance	28.0	7	51230.17788	18000
4	Douglas	Male	97308	Marketing	28.0	7	104066.04060	17000

4. Drop the missing value row-wise and display the shape of dataframe (use inplace=True)

In [7]:
```
df.dropna(inplace=True)
df.shape
```

Out[7]:
```
(22, 8)
```

5. Calculate the central tendency measures for 'Experience' and display the same

In [8]:
```
mean_exp = df['Experience'].mean()
median_exp = df['Experience'].median()
mode_exp = df['Experience'].mode().iloc[0]
print("Mean Experience:", mean_exp)
```

```
print("Median Experience:", median_exp)
print("Mode Experience:", mode_exp)
```

```
Mean Experience: 13.681818181818182
Median Experience: 12.5
Mode Experience: 7
```

6. Calculate the variability measures for 'Experience' and display the same

In [9]:
```
range_exp = df['Experience'].max() - df['Experience'].min()
variance_exp = df['Experience'].var()
stddev_exp = df['Experience'].std()
print("Range:", range_exp)
print("Variance:", variance_exp)
print("Standard:", stddev_exp)
```

```
Range: 21
Variance: 37.84632034632035
Standard: 6.151936308701542
```

7. Calculate the IQR using quantile for 'Experience' and display the same

In [27]:
```
q1 = df['Experience'].quantile(0.25)
q3 = df['Experience'].quantile(0.75)
print('iqr' , q3 - q1)
```

```
iqr 8.0
```

8. Calculate the z-score for 'Experience' and display the same

In [28]:
```
import scipy.stats as stats
zscore = stats.zscore(df['Experience'])
print('zscore',zscore)
```

```
zscore 0    -1.444443
1    -1.278068
2    -1.111692
3    -1.111692
4    -0.945316
5    -0.778941
6    -0.612565
7    -0.446189
8    -0.446189
9    -0.279814
10   -0.279814
11   -0.113438
12   -0.113438
13    0.052938
14    0.219313
15    0.219313
16    0.718440
17    1.051192
18    1.217568
19    1.550319
20    1.883070
21    2.049446
Name: Experience, dtype: float64
```

9. Plot the heatmap using the correlation ('Salary','Experience','Age')

In [ ]:
```
display = df[['Salary', 'Experience', 'Age']]
c = display.corr()
sns.heatmap(c,xticklabels=c.columns,yticklabels=c.columns, annot=True)
```

Out[ ]:
```
<Axes: >
```

10. Add 2 rows at the end of the dataframe with the given values and display last 5 rows

{'First Name':'Zion', 'Gender':'Male', 'Team':'Finance', 'Age':37, 'Experience':90,'New_Salary':146075.4, 'Incentive':20000} {'First Name':'Frances', 'Gender':'Male', 'Salary':139852, 'Team':'Business Development', 'Age':34, 'Experience':95, 'New_Salary':150374.5, 'Incentive':15500}

In [13]:
```
new_rows = [
    {'First Name': 'Zion', 'Gender': 'Male', 'Team': 'Finance', 'Age': 37, 'Experie
    {'First Name': 'Frances', 'Gender': 'Male', 'Salary': 139852, 'Team': 'Business
]
df = df.append(new_rows, ignore_index=True)
df.tail(5)
```

```
/tmp/ipykernel_3864579/2187930024.py:5: FutureWarning: The frame.append method is
deprecated and will be removed from pandas in a future version. Use pandas.concat
instead.
  df = df.append(new_rows, ignore_index=True)
```

Out[13]:

	First Name	Gender	Salary	Team	Age	Experience	New_Salary	Incentive	zscore
19	Donna	Female	81014.0	Product	49.0	23	82548.40516	10600	1.514675
20	Ruby	Female	65476.0	Product	54.0	25	72031.45712	10400	1.839776
21	Lillian	Female	59414.0	Product	55.0	26	60160.23984	10300	2.002326
22	Zion	Male	NaN	Finance	37.0	90	146075.40000	20000	NaN
23	Frances	Male	139852.0	Business Development	34.0	95	150374.50000	15500	NaN

11. Replace NaN value in 'Salary' with mean Salary and display last 5 rows

In [46]:
```
mean_salary = df['Salary'].mean()
df['Salary'].fillna(mean_salary, inplace=True)
df.tail(5)
```

Out[46]:

	First Name	Gender	Salary	Team	Age	Experience	New_Salary	Incentive	zscore
17	Kimberly	Female	41426.0	Finance	44.0	20	44512.23700	11000	1.027023
18	Louise	Female	63241.0	Business Development	45.0	21	72810.62812	10800	1.189574
19	Donna	Female	81014.0	Product	49.0	23	82548.40516	10600	1.514675
20	Ruby	Female	65476.0	Product	54.0	25	72031.45712	10400	1.839776
21	Lillian	Female	59414.0	Product	55.0	26	60160.23984	10300	2.002326

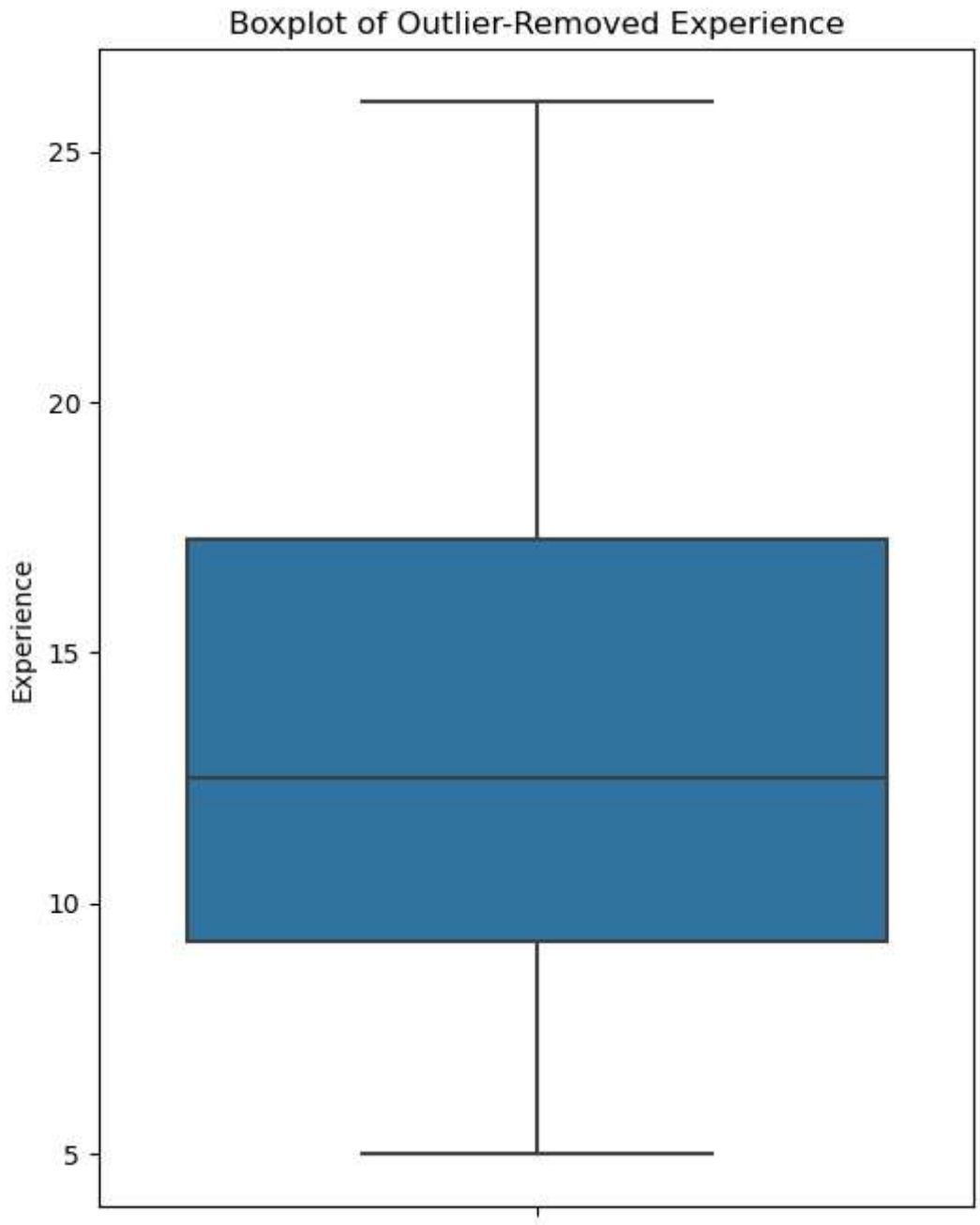
12. Detect the outliers in 'Experience' with boxplot

In [ ]:
```
sns.catplot(x='Experience' , kind = 'box' , data = df)
```

13. Remove the outliers using IQR and recalculate IQR in outlier removed 'Experience' column and analyse with boxplot (Use df.copy())

In [21]:
```
df_copy = df.copy()
q1 = df_copy['Experience'].quantile(0.25)
q3 = df_copy['Experience'].quantile(0.75)
iqr = q3 - q1
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr
df_copy = df_copy[(df_copy['Experience'] >= lower_bound) & (df_copy['Experience'] <
iqr_removed = df_copy['Experience'].quantile(0.75) - df_copy['Experience'].quantile
plt.figure(figsize=(6, 8))
sns.boxplot(data=df_copy, y='Experience')
plt.title("Boxplot Removed Experience")
```

```
plt.show()
iqr_removed
```



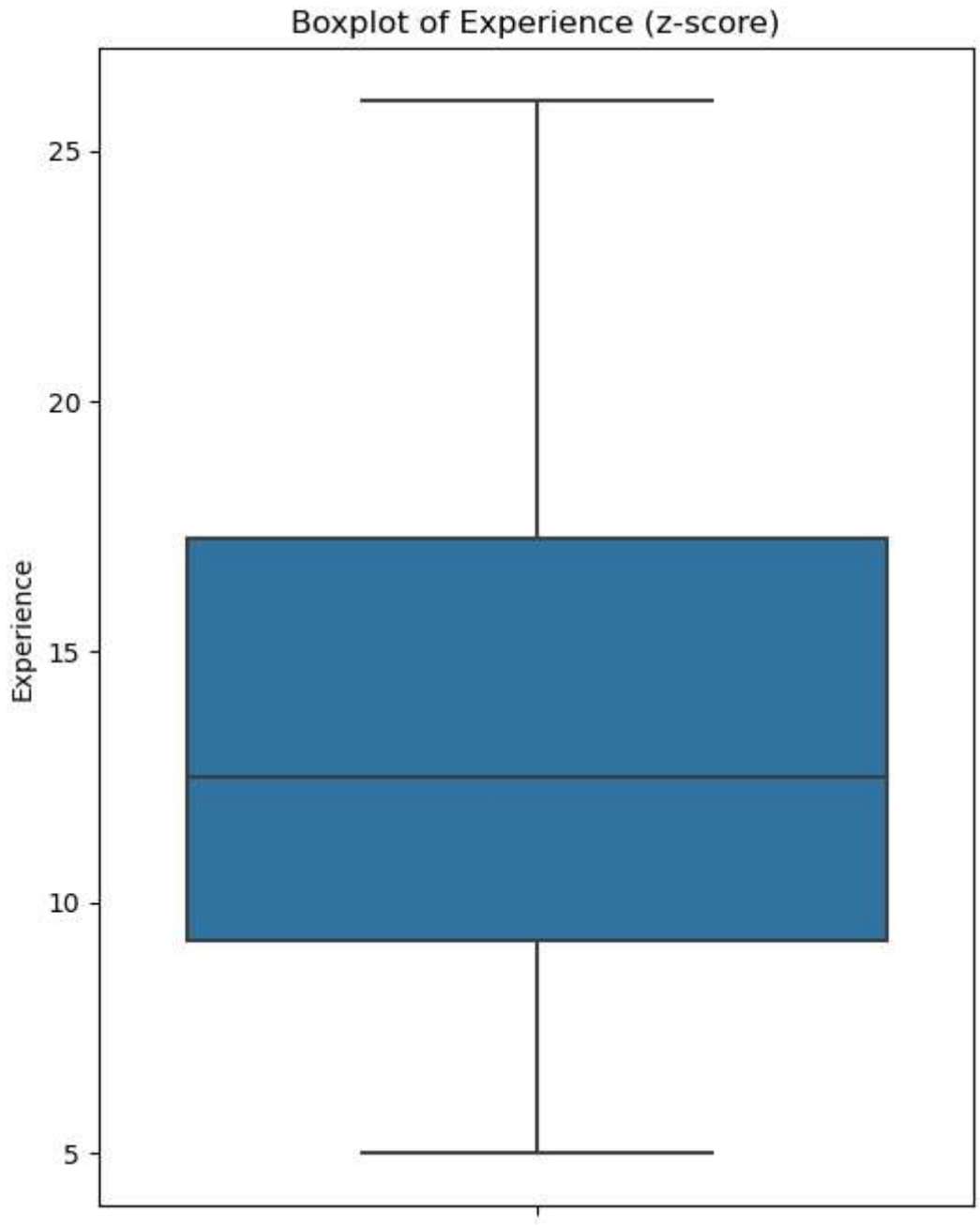
Out[21]: 8.0

14. Remove the outliers using z-score and recalculate z-score in outlier removed

'Experience' column and analyse with boxplot (Use df.copy())

In [23]:
```
df_copy = df.copy()
mean_exp = df_copy['Experience'].mean()
std_exp= df_copy['Experience'].std()
zscore = 3
df_copy['Experience_zscore'] = (df_copy['Experience'] - mean_exp) / std_exp
df_copy = df_copy[abs(df_copy['Experience_zscore']) <= zscore]
```

```
mean_exp_rem = df_copy['Experience'].mean()
stddev_exp_rem = df_copy['Experience'].std()
df_copy['Experizscore_rem'] = (df_copy['Experience'] - mean_experience_removed) / s
plt.figure(figsize=(6, 8))
sns.boxplot(data=df_copy, y='Experience')
plt.title("Boxplot of Experience (z-score)")
plt.show()
```



15. Drop the last two rows added in the dataframe

In [24]:
```
df.drop(df.index[-2:], inplace=True)
df
```

Out[24]:

	First Name	Gender	Salary	Team	Age	Experience	New_Salary	Incentive	zscore
0	Angela	Female	54568.0	Business Development	27.0	5	64675.63064	19000	-1.411233
1	Allan	Male	125792.0	Client Services	28.0	6	132134.43260	18500	-1.248683
2	Rohan	Female	45906.0	Finance	28.0	7	51230.17788	18000	-1.086133
3	Douglas	Male	97308.0	Marketing	28.0	7	104066.04060	17000	-1.086133
4	Brandon	Male	112807.0	Human Resources	30.0	8	132539.20040	16000	-0.923582
5	Diana	Female	132940.0	Client Services	31.0	9	158307.61080	15800	-0.761032
6	Matthew	Male	100612.0	Marketing	34.0	10	114340.50740	15000	-0.598481
7	Larry	Male	101004.0	Client Services	35.0	11	102406.94560	14700	-0.435931
8	Joshua	Male	90816.0	Client Services	35.0	11	107903.93860	14300	-0.435931
9	Jerry	Male	72000.0	Finance	35.0	12	78724.80000	14000	-0.273380
10	Lois	Female	64714.0	Legal	35.0	12	67906.98876	14000	-0.273380
11	Dennis	Male	115163.0	Legal	36.0	13	126823.25380	13000	-0.110830
12	John	Male	97950.0	Client Services	37.0	13	111538.60350	12000	-0.110830
13	Thomas	Male	61933.0	Marketing	38.0	14	68711.56685	11900	0.051721
14	Shawn	Male	111737.0	Human Resources	39.0	15	118903.81120	11500	0.214271
15	Gary	Male	109831.0	Product	39.0	15	116235.24560	11500	0.214271
16	Jeremy	Male	90370.0	Human Resources	42.0	18	97029.36530	11000	0.701922
17	Kimberly	Female	41426.0	Finance	44.0	20	44512.23700	11000	1.027023
18	Louise	Female	63241.0	Business Development	45.0	21	72810.62812	10800	1.189574
19	Donna	Female	81014.0	Product	49.0	23	82548.40516	10600	1.514675
20	Ruby	Female	65476.0	Product	54.0	25	72031.45712	10400	1.839776
21	Lillian	Female	59414.0	Product	55.0	26	60160.23984	10300	2.002326