```
from google.colab import drive

drive.mount("/content/drive")
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sn
```

```
df = pd.read_csv("/content/drive/MyDrive/HR Analytics/train_LZdllcl.csv")
```

+ Code + Text

```
df
```

	employee_id	department	region	education	gender	recruitment_channel	no_
0	65438	Sales & Marketing	region_7	Master's & above	f	sourcing	
1	65141	Operations	region_22	Bachelor's	m	other	
2	7513	Sales & Marketing	region_19	Bachelor's	m	sourcing	
3	2542	Sales & Marketing	region_23	Bachelor's	m	other	
4	48945	Technology	region_26	Bachelor's	m	other	
...	...	...	...	...	...	...	
54803	3030	Technology	region_14	Bachelor's	m	sourcing	
54804	74592	Operations	region_27	Master's & above	f	other	
54805	13918	Analytics	region_1	Bachelor's	m	other	
54806	13614	Sales & Marketing	region_9	NaN	m	sourcing	
54807	51526	HR	region_22	Bachelor's	m	other	

54808 rows × 14 columns

```
df.shape
```

```
(54808, 14)
```

```
df.columns
```

```
Index(['employee_id', 'department', 'region', 'education', 'gender',
       'recruitment_channel', 'no_of_trainings', 'age', 'previous_year_rating',
       'length_of_service', 'KPIs_met >80%', 'awards_won?',
       'avg_training_score', 'is_promoted'],
      dtype='object')
```

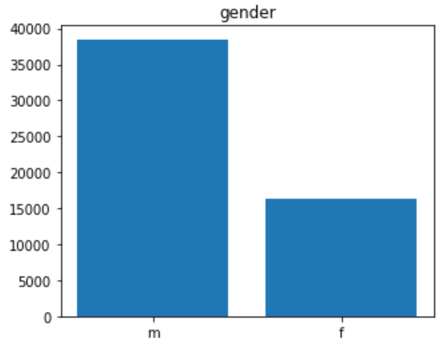
```
df.education.value_counts()
```

```
Bachelor's          36669
Master's & above    14925
Below Secondary       805
Name: education, dtype: int64
```

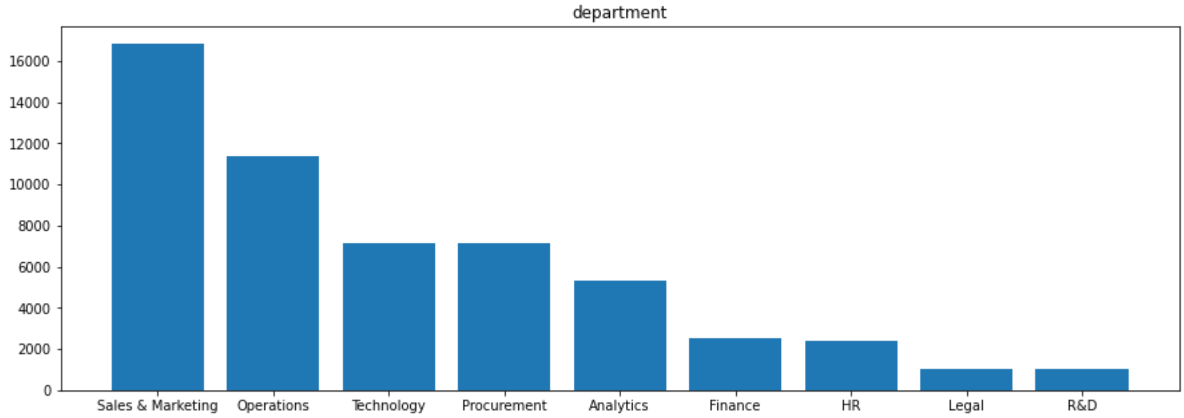
```
def barplot_count(column,x=5,y=5,df=df):

    plt.figure(figsize=(x,y))
    cnt_column = df[column].value_counts().to_dict()
    plt.title(column)
    plt.bar(cnt_column.keys(),cnt_column.values())
```

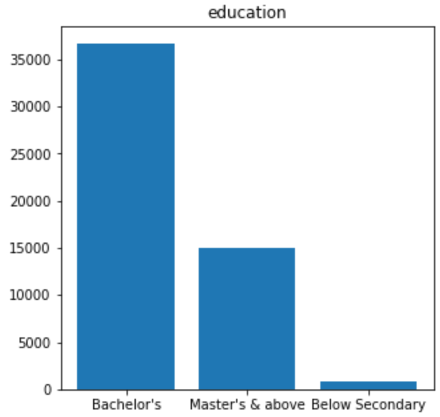
```
barplot_count('gender',5,4)
```



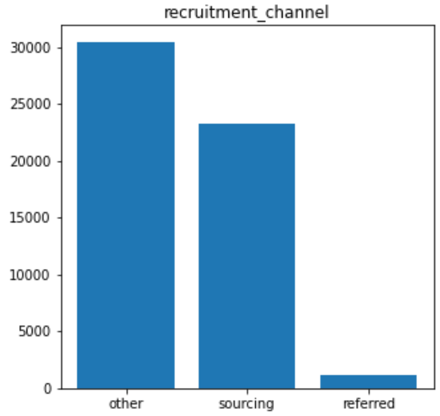
```
barplot_count('department',15,5)
```



```
barplot_count('education')
```

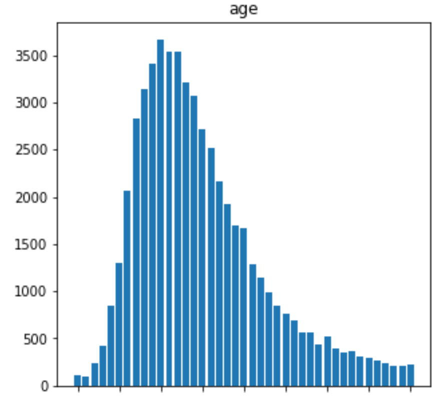


```
df.recruitment_channel.unique()
barplot_count('recruitment_channel')
```



```
barplot_count('age')
df.age.describe()
```

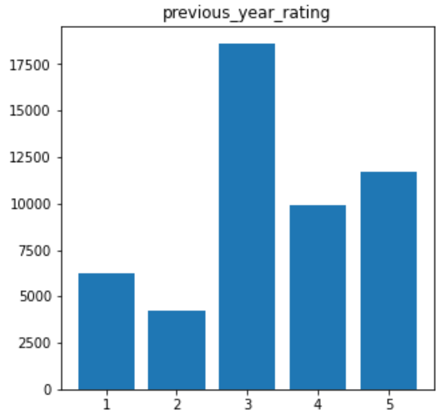
```
count    54808.000000
mean        34.803915
std          7.660169
min         20.000000
25%         29.000000
50%         33.000000
75%         39.000000
max         60.000000
Name: age, dtype: float64
```



```
agefrom25_40_in_percentage = (df.loc[ (df.age >= 25) & (df.age < 40)].shape[0])/df.shape[0]*100
agefrom25_40_in_percentage
```

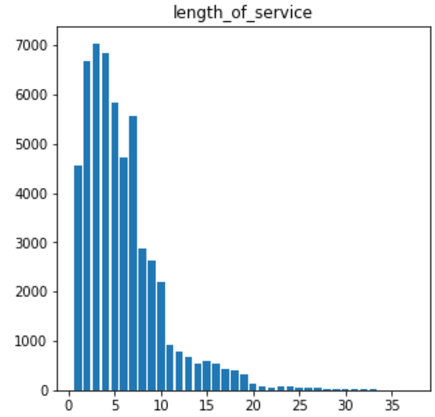
```
74.38330170777988
```

```
barplot_count('previous_year_rating')
```

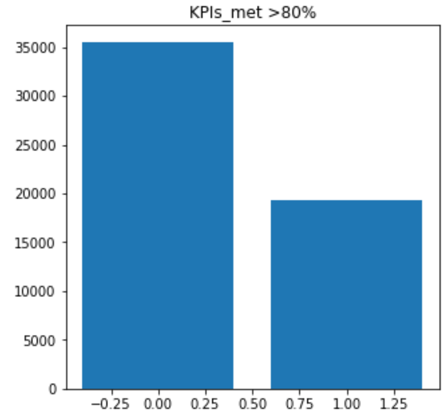


```
barplot_count('length_of_service')
df['length_of_service'].describe()
```

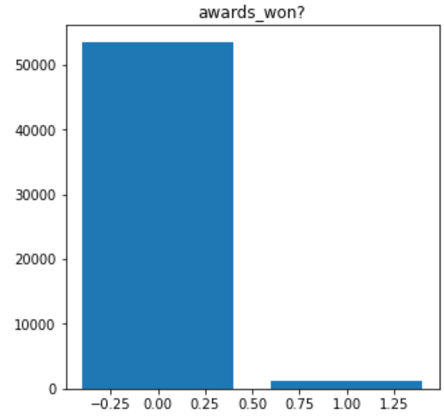
```
count    54808.000000
mean         5.865512
std          4.265094
min          1.000000
25%          3.000000
50%          5.000000
75%          7.000000
max         37.000000
Name: length_of_service, dtype: float64
```



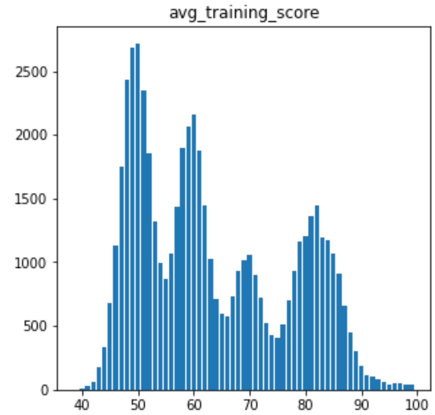
```
barplot_count('KPIs_met >80%')
```



```
barplot_count('awards_won?')
```



```
barplot_count('avg_training_score')
```



```
df['avg_training_score'].unique()
```

```
array([49, 60, 50, 73, 85, 59, 63, 83, 54, 77, 80, 84, 51, 46, 75, 57, 70,
       68, 79, 44, 72, 61, 48, 58, 87, 47, 52, 88, 71, 65, 62, 53, 78, 91,
       82, 69, 55, 74, 86, 90, 92, 67, 89, 56, 76, 81, 45, 64, 39, 94, 93,
       66, 95, 42, 96, 40, 99, 43, 97, 41, 98])
```

```
df['avg_training_score'].describe()
```

```
count    54808.000000
mean        63.386750
std         13.371559
min         39.000000
25%         51.000000
50%         60.000000
75%         76.000000
max         99.000000
Name: avg_training_score, dtype: float64
```

```
Total_promotedf = df.loc[df.is_promoted == 1]
Total_promoted = Total_promotedf.shape[0]
Total_promotedf.describe()
```

	employee_id	no_of_trainings	age	previous_year_rating	length_of_service	KPIs_met >80%	awards_won?	avg_training_sc
count	4668.000000	4668.000000	4668.000000	4329.000000	4668.000000	4668.000000	4668.000000	4668.00
mean	39285.092331	1.203299	34.372965	3.986140	5.716367	0.698800	0.119751	71.32
std	22754.632174	0.513805	7.081449	1.052149	4.007772	0.458828	0.324706	14.74
min	39.000000	1.000000	20.000000	1.000000	1.000000	0.000000	0.000000	41.00
25%	19390.500000	1.000000	29.000000	3.000000	3.000000	0.000000	0.000000	59.00
50%	39439.500000	1.000000	33.000000	4.000000	5.000000	1.000000	0.000000	71.00
75%	59082.250000	1.000000	38.000000	5.000000	7.000000	1.000000	0.000000	84.00

```
FullDataATSMedian = df.avg_training_score.quantile(.50)
LessmedianPromoted=Total_promotedf.loc[Total_promotedf.avg_training_score < FullDataATSMedian]
Lmp=(LessmedianPromoted.shape[0]/Total_promoted)*100
Lmp
```

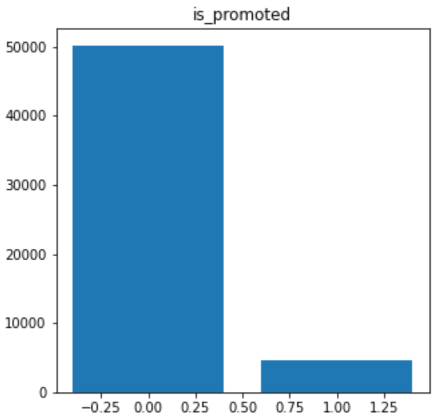
```
26.028277634961437
```

```
Hmp = 100.0 - Lmp
Hmp
```

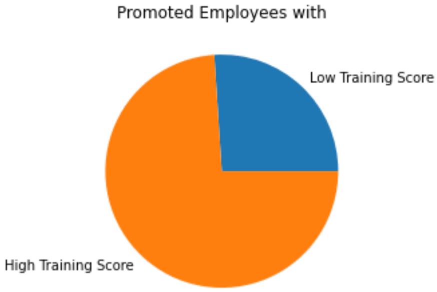
```
73.97172236503856
```

Hence if Employee has high training score, chances for him to be promoted is 73%

```
barplot_count('is_promoted')
```



```
Pie_dict={'Low Training Score':Lmp, 'High Training Score':Hmp}
plt.title('Promoted Employees with')
plt.pie([Lmp, Hmp], labels=Pie_dict.keys())
plt.show()
```



```
EmployeeWithAwards = Total_promotedf.loc[Total_promotedf['awards_won?'] == 1].shape[0]
```

```
PEmpPromoplusAwards = (EmployeeWithAwards/Total_promoted)*100
PEmpPromoplusAwards
```

```
    11.975149957155098
```

```
PromoWithKpi = Total_promotedf.loc[Total_promotedf['KPIs_met >80%'] == 1].shape[0]
```

```
PpromoWithKpi=(PromoWithKpi/Total_promoted)*100
PpromoWithKpi
```

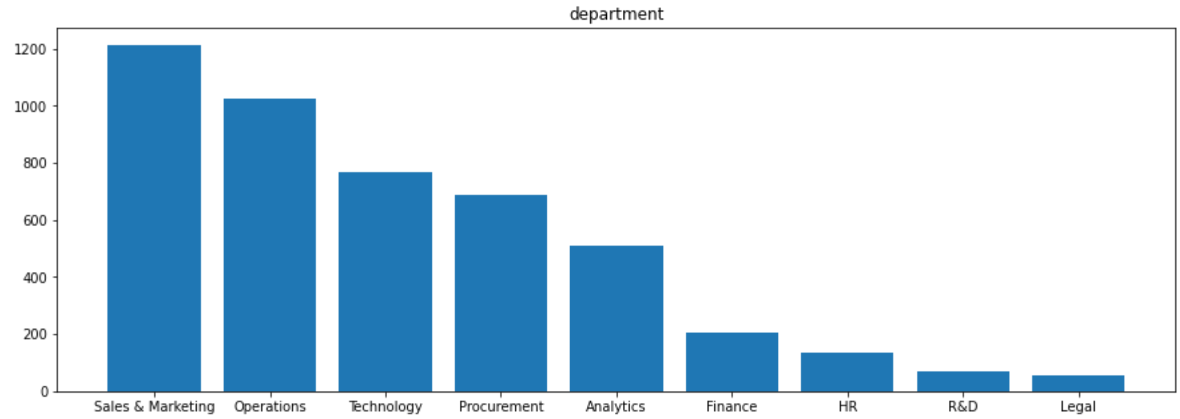
```
    69.88003427592116
```

```
FullDataLOSMedian = df.length_of_service.median()
LOSmorethan7 = Total_promotedf.loc[Total_promotedf.length_of_service > FullDataLOSMedian ].shape[0]
```

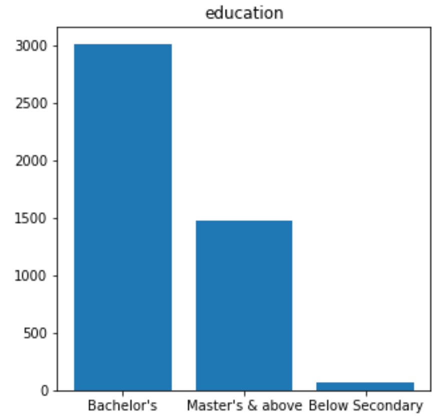
```
PpromoWithLOS7 =(LOSmorethan7/Total_promoted)*100
PpromoWithLOS7
```

```
    43.14481576692374
```

```
barplot_count('department',15,5, df=Total_promotedf)
```



```
barplot_count('education', df=Total_promotedf)
```



```
AllBachelor=df.loc[df.education == "Bachelor's" ].shape[0]
PromotedBachelor=Total_promotedf.loc[Total_promotedf.education == "Bachelor's" ].shape[0]
ShareofBachelor=(PromotedBachelor/AllBachelor)*100
ShareofBachelor
```

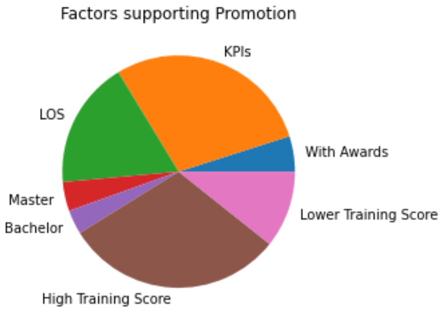
```
    8.20311434726881
```

```
AllMasters=df.loc[df.education == "Master's & above" ].shape[0]
PromotedMasters=Total_promotedf.loc[Total_promotedf.education == "Master's & above" ].shape[0]
ShareofMasters=(PromotedMasters/AllMasters)*100
ShareofMasters
```

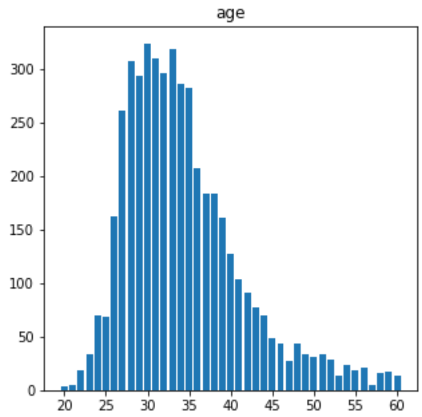
```
    9.855946398659967
```

Master's have higher chances of getting promoted!

```
plt.title("Factors supporting Promotion")
plt.pie([
PEmpPromoplusAwards,
PpromoWithKpi,
PpromoWithLOS7,
ShareofMasters,
ShareofBachelor, Hmp, Lmp
],labels=['With Awards','KPIs','LOS','Master','Bachelor','High Training Score','Lower Training Score'])
plt.show()
```

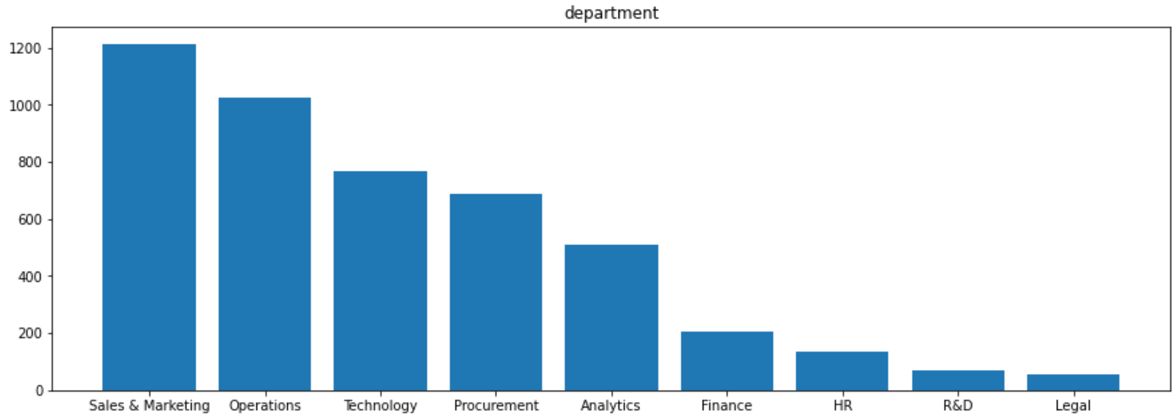


```
barplot_count('age', df=Total_promotedf)
```



Age between 26 to 38 are have seen more promotion

```
barplot_count('department',x=15,y=5,df=Total_promotedf)
```



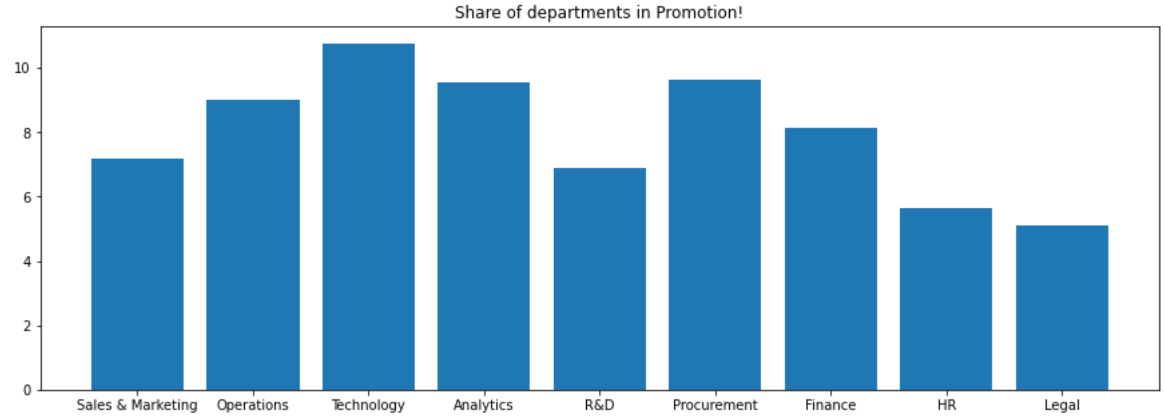
```
Lis=[]

for i in df.department.unique():
  Dpt=df.loc[df.department == i ].shape[0]
  PromotedDPT=Total_promotedf.loc[Total_promotedf.department == i ].shape[0]
  ShareofDPT=(PromotedDPT/Dpt)*100
  Lis.append(ShareofDPT)
Gg = {'Department':df.department.unique(),
'Share':Lis}
pd.DataFrame(Gg,index=range(len(Lis)))
```

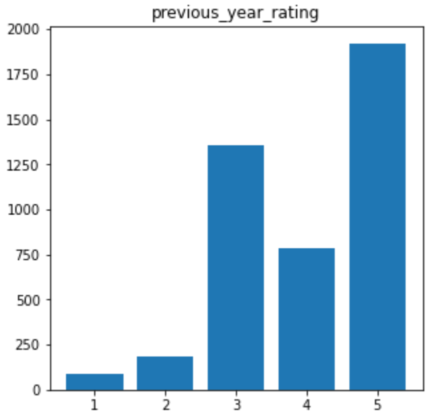
	Department	Share
0	Sales & Marketing	7.203088
1	Operations	9.014804
2	Technology	10.759316
3	Analytics	9.566517
4	R&D	6.906907
5	Procurement	9.638554
6	Finance	8.123028
7	HR	5.624483
8	Legal	5.101059

```
plt.figure(figsize=(15,5))
plt.title("Share of departments in Promotion!")
plt.bar(df.department.unique(),Lis)
```

```
<BarContainer object of 9 artists>
```

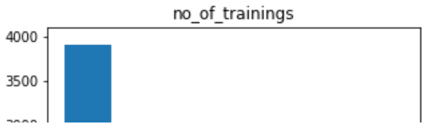


```
barplot_count('previous_year_rating',df=Total_promotedf)
```

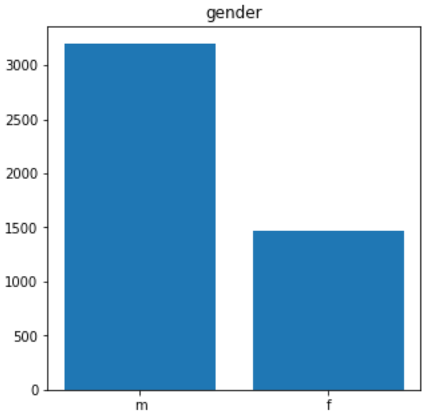


Employees with previous year rating of 5 have highest chances of getting promoted, which is quite obvious!

```
barplot_count('no_of_trainings',df=Total_promotedf)
```



```
barplot_count('gender',df=Total_promotedf)
```



```
Allgender=df.loc[df.gender == "m" ].shape[0]
PromotedGenders=Total_promotedf.loc[Total_promotedf.gender== "m" ].shape[0]
ShareofMales=(PromotedGenders/Allgender)*100
ShareofMales
```

```
8.315149625935161
```

```
Allgender=df.loc[df.gender == "f" ].shape[0]
PromotedGenders=Total_promotedf.loc[Total_promotedf.gender== "f" ].shape[0]
ShareofFemales=(PromotedGenders/Allgender)*100
ShareofFemales
```

```
8.993379107405591
```

```
df.columns
```

```
Index(['employee_id', 'department', 'region', 'education', 'gender',
       'recruitment_channel', 'no_of_trainings', 'age', 'previous_year_rating',
       'length_of_service', 'KPIs_met >80%', 'awards_won?',
       'avg_training_score', 'is_promoted'],
      dtype='object')
```

```
Total_promotedf.recruitment_channel.value_counts()
```

```
other       2556
sourcing    1974
referred     138
Name: recruitment_channel, dtype: int64
```

✓ 0s completed at 1:06 PM