

```
In [1]: import pandas as pd
import xlrd
import numpy as np
import random
```

#Import data from above in Environment

```
In [2]: c1=pd.read_excel(r"C:\Users\Aman Sharma\Desktop\python project\Employee-Master.xlsx")
c1.head()
```

Out[2]:

	EmployeeID	Employee Name	BirthDate	Today	Age	MaritalStatus	Gender	HireDate	Length of Service	Dept	Salary	Job Grade
0	1	Gustavo Achong	1982-05-15	2020-06-10	38		M	M 2002-07-31	17	Sales	2295	Admin
1	2	Catherine Abel	1987-06-03	2020-06-11	33		S	F 2010-02-26	10	Sales	962	Management
2	3	Kim Abercrombie	1964-12-13	2020-06-12	55		M	F 2007-12-12	12	Finance	4006	Admin
3	4	Humberto Acevedo	1965-01-23	2020-06-13	55		S	M 2007-01-05	13	Logistics	4547	Admin
4	5	Pilar Ackerman	1949-08-29	2020-06-14	70		M	F 2007-01-11	13	Human Resource	1932	Admin

Calculate the average age Department Name Wise

```
In [3]: avg = c1.groupby('Dept')['Age'].mean()
avg

Out[3]: Dept
Finance      44.181818
Human Resource  50.833333
Logistics     50.000000
Production    41.785714
Sales         42.272727
Name: Age, dtype: float64
```

Calculate the minimum, and maximum length of service Job Title Wise

```
In [4]: min_max=c1.agg({'Length of Service': ['min', 'max']})
min_max

Out[4]:
```

	Length of Service
min	6
max	22

Calculate the count of employees “Department Name” and “Gender” Wise

```
In [5]: count_ = c1.groupby('Dept')['Gender'].count()
count_

Out[5]: Dept
Finance      11
Human Resource  6
Logistics     8
Production    14
Sales        11
Name: Gender, dtype: int64
```

Get all rows from DataFrame where length of service is greater than 5 years and Gender is “Female”

```
In [6]: df = pd.DataFrame(c1)
df.head()
```

Out[6]:

	EmployeeID	Employee Name	BirthDate	Today	Age	MaritalStatus	Gender	HireDate	Length of Service	Dept	Salary	Job Grade
0	1	Gustavo Achong	1982-05-15	2020-06-10	38		M	M 2002-07-31	17	Sales	2295	Admin
1	2	Catherine Abel	1987-06-03	2020-06-11	33		S	F 2010-02-26	10	Sales	962	Management
2	3	Kim Abercrombie	1964-12-13	2020-06-12	55		M	F 2007-12-12	12	Finance	4006	Admin
3	4	Humberto Acevedo	1965-01-23	2020-06-13	55		S	M 2007-01-05	13	Logistics	4547	Admin
4	5	Pilar Ackerman	1949-08-29	2020-06-14	70		M	F 2007-01-11	13	Human Resource	1932	Admin

```
In [7]: df.columns=df.columns.str.replace(' ','_')

In [8]: ft=df[(df.Gender == 'F') & ( df.Length_of_Service > 5)]
ft.head(5)
```

Out[8]:

	EmployeeID	Employee_Name	BirthDate	Today	Age	MaritalStatus	Gender	HireDate	Length_of_Service	Dept	Salary	Job_Grade
1	2	Catherine Abel	1987-06-03	2020-06-11	33		S	F 2010-02-26	10	Sales	962	Management
2	3	Kim Abercrombie	1964-12-13	2020-06-12	55		M	F 2007-12-12	12	Finance	4006	Admin
4	5	Pilar Ackerman	1949-08-29	2020-06-14	70		M	F 2007-01-11	13	Human Resource	1932	Admin
5	6	Frances Adams	1965-04-19	2020-06-15	55		S	F 2005-01-20	15	Sales	3223	Management
6	7	Margaret Smith	1946-02-16	2020-06-16	74		S	F 2013-01-26	7	Sales	1329	Operations

Get the number of employees and mean of salary state name wise

```
In [9]: c1=pd.read_excel(r"C:\Users\Aman Sharma\Desktop\python project\Employee-Master.xlsx",sheet_name=0)
c2=pd.read_excel(r"C:\Users\Aman Sharma\Desktop\python project\Employee-Master.xlsx",sheet_name =1)
c3=pd.read_excel(r"C:\Users\Aman Sharma\Desktop\python project\Employee-Master.xlsx",sheet_name=2)
```

In [10]: df1=pd.merge(c1,c2,how='left')

In [11]: c13.columns=['StartDate', 'CourseCode', 'CourseName', 'EmployeeID', 'Cost', 'Supplier', 'Year', 'Month']

In [12]: df2=pd.merge(df1,c13,how='left')
df2.head(2)

Out[12]:

	EmployeeID	Employee Name	BirthDate	Today	Age	MaritalStatus	Gender	HireDate	Length of Service	Dept	...	City	State	Country	StartDate	CourseCode	CourseName	Cost	Supplier	Year
0	1	Gustavo Achong	1982-05-15	2020-06-10	38		M	M 2002-07-31	17	Sales	...	Surat	Gujarat	India	2012-04-12	1.0	Communication Workshop	1300.0	Communication Experts	2012.0
1	1	Gustavo Achong	1982-05-15	2020-06-10	38		M	M 2002-07-31	17	Sales	...	Surat	Gujarat	India	2014-07-19	16.0	Computer Literacy	2200.0	Computer Trainers	2014.0

2 rows × 22 columns

```
In [13]: df2.groupby('State').Salary.agg(['count','mean','min','max'])

Out[13]:
```

	count	mean	min	max
State				
Andhra Pradesh	6	3666.666667	2018	4491
Bihar	11	2893.909091	1161	3843
Chhattisgarh	4	3390.000000	3390	3390
Delhi	11	2565.363636	1395	4406
Gujarat	13	3090.692308	1808	3639
Haryana	11	1097.818182	962	1128
Jharkhand	6	3506.166667	548	4547
Karnataka	2	2939.000000	2679	3199
Kerala	15	3316.333333	1224	4096
Madhya Pradesh	7	2058.857143	1932	2154
Maharashtra	25	1613.240000	561	3978
Manipur	2	2298.000000	2298	2298
Rajasthan	9	1260.333333	1134	1513
Tamil Nadu	5	1589.000000	664	2533
Uttar Pradesh	13	2378.307692	600	3967
Uttarakhand	8	2108.625000	1329	3408
West Bengal	7	3913.000000	3723	3989

Get the department name, and training name wise number of trainings conducted

```
In [14]: df2.head(2)
```

Out[14]:

	EmployeeID	Employee Name	BirthDate	Today	Age	MaritalStatus	Gender	HireDate	Length of Service	Dept	...	City	State	Country	StartDate	CourseCode	CourseName	Cost	Supplier	Year
0	1	Gustavo Achong	1982-05-15	2020-06-10	38		M	M 2002-07-31	17	Sales	...	Surat	Gujarat	India	2012-04-12	1.0	Communication Workshop	1300.0	Communication Experts	2012.0
1	1	Gustavo Achong	1982-05-15	2020-06-10	38		M	M 2002-07-31	17	Sales	...	Surat	Gujarat	India	2014-07-19	16.0	Computer Literacy	2200.0	Computer Trainers	2014.0

2 rows × 22 columns

10. Make a bar chart based on Job Title wise average age and change the color as “RED” for the highest aged Job Title with text on top of the bar as “Most Experienced Employee Job Title”

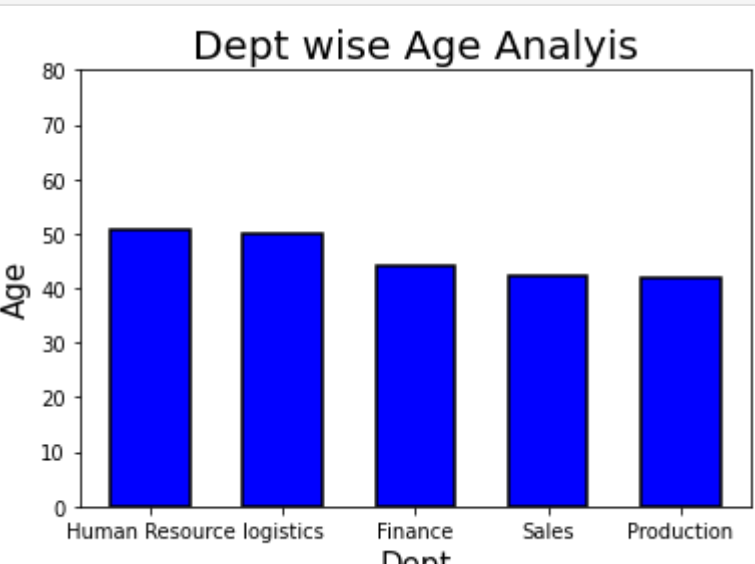
```
In [15]: import matplotlib.pyplot as plt

In [16]: bar = c1.groupby('Dept')['Age'].mean().sort_values(ascending=False)
name=['Human Resource','logistics','Finance','Sales','Production']

In [17]: bar

Out[17]: Dept
Human Resource    50.833333
Logistics         50.000000
Finance           44.181818
Sales             42.272727
Production        41.785714
Name: Age, dtype: float64

In [18]: plt.bar(name,bar,width=.6,align="center",color=["b","k"],edgecolor="k",
linewidth=1.5,linestyle="-",label='Dept')
plt.ylim(0,80)
plt.title("Dept wise Age Analysis",fontsize=20)
plt.xlabel("Dept",fontsize=15)
plt.ylabel("Age",fontsize=15)
plt.rcParams['figure.figsize'][0]=20
```



Make a donut chart for Department Wise Salary distribution

```
In [19]: pie=df2.groupby('Dept')['Salary'].count()
pie

Out[19]: Dept
Finance      30
Human Resource  21
Logistics     25
Production    38
Sales         41
Name: Salary, dtype: int64

In [20]: l = ['Finance','Human Rousouce','logistics','production','Sales']
color = ['red','green','blue','orange','yellow']
pie.pie(pie, startangle=180,radius=1.2,explode=None,counterClock=True,center=(0, 0),labels=1)
centre_circle = plt.Circle((0, 0), 0.70, fc='white')
fig = plt.gcf()
plt.title('Department wise Salary Distribution', fontsize=20, loc='left')

# Adding Circle in Pie chart
fig.gca().add_artist(centre_circle)

plt.show
```

Out[20]:

<function matplotlib.pyplot.show(close=None, block=None)>

Department wise Salary Distribution

Get the output in cross tabular format with Gender/s Job Title for an average length of service

```
In [21]: pivot=pd.pivot_table(df2,index=["Gender"],values=["Length of Service"],aggfunc=(['mean','count'])
pivot

Out[21]:
```

		Length of Service
	Gender	count mean
	F	62 12.677419
	M	93 14.279570

Save the result of the above Question Results in a text file with the name “d:/Sales_Analysis.csv”

```
In [22]: pivot.to_csv(r"C:\Users\Aman Sharma\Desktop\python project.csv', header=None, index=None, sep=' ', mode='a')
```

Make a TreeMap chart of the top 5 City Name v/s average of length of Service

```
In [23]: tree_map= df2.groupby('City')['Length of Service'].mean().sort_values(ascending=False).head(5)
tree_map

Out[23]: City
Agra      22.0
Bokaro    22.0
Dehra Dun 22.0
Kochi     22.0
Kanpur    21.0
Name: Length of Service, dtype: float64

In [24]: pip install squarify

Requirement already satisfied: squarify in c:\users\aman sharma\anaconda3\lib\site-packages (0.4.3)
Note: you may need to restart the kernel to use updated packages.

In [25]: import matplotlib.pyplot as plt
import squarify
color = ['red','green','blue','orange','yellow']
labels = ['Agra','Bokaro','Dehra Dun','Kochi','Kanpur']
squarify.plot(tree_map,color=color,alpha=0.9,label=labels)
plt.axis('off')
plt.show()
```



```
In [ ]:
In [ ]:
```