Imported Libraries

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

Loading and preparing the data

```
excelfile = pd.read_excel("Employee Sample Data.xlsx")
df = pd.DataFrame(excelfile)
```

DataFrame Basics

```
print (df.info())
print(df.head())
print(df.describe())
print(df.columns)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   EEID           1000 non-null   object
 1   Full Name      998 non-null    object
 2   Job Title      999 non-null    object
 3   Department     998 non-null    object
 4   Business Unit  1000 non-null   object
 5   Gender         999 non-null    object
 6   Ethnicity      993 non-null    object
 7   Age            994 non-null    float64
 8   Hire Date      993 non-null    datetime64[ns]
 9   Annual Salary  989 non-null    float64
 10  Bonus %        992 non-null    float64
 11  Country        998 non-null    object
 12  City           998 non-null    object
 13  Exit Date      85 non-null     datetime64[ns]
dtypes: datetime64[ns](2), float64(3), object(9)
memory usage: 109.5+ KB
None
     EEID        Full Name                 Job Title Department  \
0  E02387      Emily Davis                Sr. Manger         IT
1  E04105    Theodore Dinh       Technical Architect         IT
2  E02572     Luna Sanders                  Director    Finance
3  E02832  Penelope Jordan  Computer Systems Manager         IT
4  E01639        Austin Vo               Sr. Analyst    Finance

            Business Unit  Gender  Ethnicity   Age  Hire Date  Annual
Salary  \
```

```
0  Research & Development  Female      Black  55.0 2016-04-08
141604.0
1           Manufacturing    Male      Asian  59.0 1997-11-29
99975.0
2     Speciality Products  Female  Caucasian  50.0 2006-10-26
163099.0
3           Manufacturing  Female  Caucasian  26.0 2019-09-27
84913.0
4           Manufacturing    Male      Asian  55.0 1995-11-20
95409.0

   Bonus %        Country       City  Exit Date
0     0.15  United States    Seattle 2021-10-16
1     0.00          China  Chongqing        NaT
2     0.20  United States    Chicago        NaT
3     0.07  United States    Chicago        NaT
4     0.00  United States    Phoenix        NaT
              Age                          Hire Date  Annual Salary
Bonus %  \
count  994.000000                             993      989.00000
992.000000
mean    44.369215  2012-04-17 22:56:11.601208576   113372.62184
0.088972
min     25.000000            1992-01-09 00:00:00    40063.00000
0.000000
25%     35.000000            2007-02-24 00:00:00    71234.00000
0.000000
50%     45.000000            2014-02-20 00:00:00    96567.00000
0.000000
75%     54.000000            2018-06-25 00:00:00   151027.00000
0.150000
max     65.000000            2021-12-26 00:00:00   258498.00000
0.400000
std     11.248162                             NaN    53729.04678
0.118135

                           Exit Date
count                             85
mean   2016-11-02 18:04:14.117647104
min              1994-12-18 00:00:00
25%              2014-12-25 00:00:00
50%              2019-05-23 00:00:00
75%              2021-04-09 00:00:00
max              2022-08-17 00:00:00
std                              NaN
Index(['EEID', 'Full Name', 'Job Title', 'Department', 'Business
Unit',
       'Gender', 'Ethnicity', 'Age', 'Hire Date', 'Annual Salary',
'Bonus %',
```

```
       'Country', 'City', 'Exit Date'],
      dtype='object')
```

Filling the null values

```
df["Exit Date"].fillna("Unknown", inplace=True)
```

just to check

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 14 columns):
 #   Column         Non-Null Count  Dtype         
---  ------         --------------  -----         
 0   EEID           1000 non-null   object        
 1   Full Name      998 non-null    object        
 2   Job Title      999 non-null    object        
 3   Department     998 non-null    object        
 4   Business Unit  1000 non-null   object        
 5   Gender         999 non-null    object        
 6   Ethnicity      993 non-null    object        
 7   Age            994 non-null    float64       
 8   Hire Date      993 non-null    datetime64[ns]
 9   Annual Salary  989 non-null    float64       
 10  Bonus %        992 non-null    float64       
 11  Country        998 non-null    object        
 12  City           998 non-null    object        
 13  Exit Date      1000 non-null   object        
dtypes: datetime64[ns](1), float64(3), object(10)
memory usage: 109.5+ KB
None
```

Dropping the rows with null values

```
df.dropna()
```

```
       EEID        Full Name                 Job Title  Department  \
0    E02387      Emily Davis               Sr. Manger          IT   
1    E04105    Theodore Dinh      Technical Architect          IT   
2    E02572     Luna Sanders                  Director     Finance   
3    E02832  Penelope Jordan  Computer Systems Manager          IT   
4    E01639        Austin Vo              Sr. Analyst     Finance   
..      ...              ...                       ...         ...   
995  E03094     Wesley Young              Sr. Analyst   Marketing   
996  E01909     Lillian Khan                  Analyst     Finance   
997  E04398      Oliver Yang                 Director   Marketing   
998  E02521      Lily Nguyen              Sr. Analyst     Finance   
```

```
999  E03545      Sofia Cheng            Vice President  Accounting

               Business Unit  Gender  Ethnicity   Age  Hire Date  \
0    Research & Development  Female      Black  55.0 2016-04-08
1             Manufacturing    Male      Asian  59.0 1997-11-29
2       Speciality Products  Female  Caucasian  50.0 2006-10-26
3             Manufacturing  Female  Caucasian  26.0 2019-09-27
4             Manufacturing    Male      Asian  55.0 1995-11-20
..                      ...     ...        ...   ...        ...
995     Speciality Products    Male  Caucasian  33.0 2016-09-18
996     Speciality Products  Female      Asian  44.0 2010-05-31
997     Speciality Products    Male      Asian  31.0 2019-06-10
998     Speciality Products  Female      Asian  33.0 2012-01-28
999               Corporate  Female      Asian  63.0 2020-07-26

     Annual Salary  Bonus %        Country       City            Exit
Date
0         141604.0     0.15  United States    Seattle  2021-10-16
00:00:00
1          99975.0     0.00          China  Chongqing
Unknown
2         163099.0     0.20  United States    Chicago
Unknown
3          84913.0     0.07  United States    Chicago
Unknown
4          95409.0     0.00  United States    Phoenix
Unknown
..             ...      ...            ...        ...
...
995        98427.0     0.00  United States   Columbus
Unknown
996        47387.0     0.00          China    Chengdu  2018-01-08
00:00:00
997       176710.0     0.15  United States      Miami
Unknown
998        95960.0     0.00          China    Chengdu
Unknown
999       216195.0     0.31  United States      Miami
Unknown

[980 rows x 14 columns]
```

Grouping by Gender and calculating the Average Annual Salary

```
grp_by_gender = df.groupby("Gender")
avg_salary= grp_by_gender["Annual Salary"].mean()
print(avg_salary)
```

```
Gender
Female    112648.816406
Male      114232.334034
Name: Annual Salary, dtype: float64
```

Grouping by Ethnicity and calculating the Annual Salary

```
grp_by_ethnicity = df.groupby("Ethnicity")
avg_salary_for_ethnicity = grp_by_ethnicity["Annual Salary"].mean()
print(avg_salary_for_ethnicity)

Ethnicity
Asian         117803.925000
Black         109021.972973
Caucasian     109355.556818
Latino        111839.433735
Name: Annual Salary, dtype: float64
```

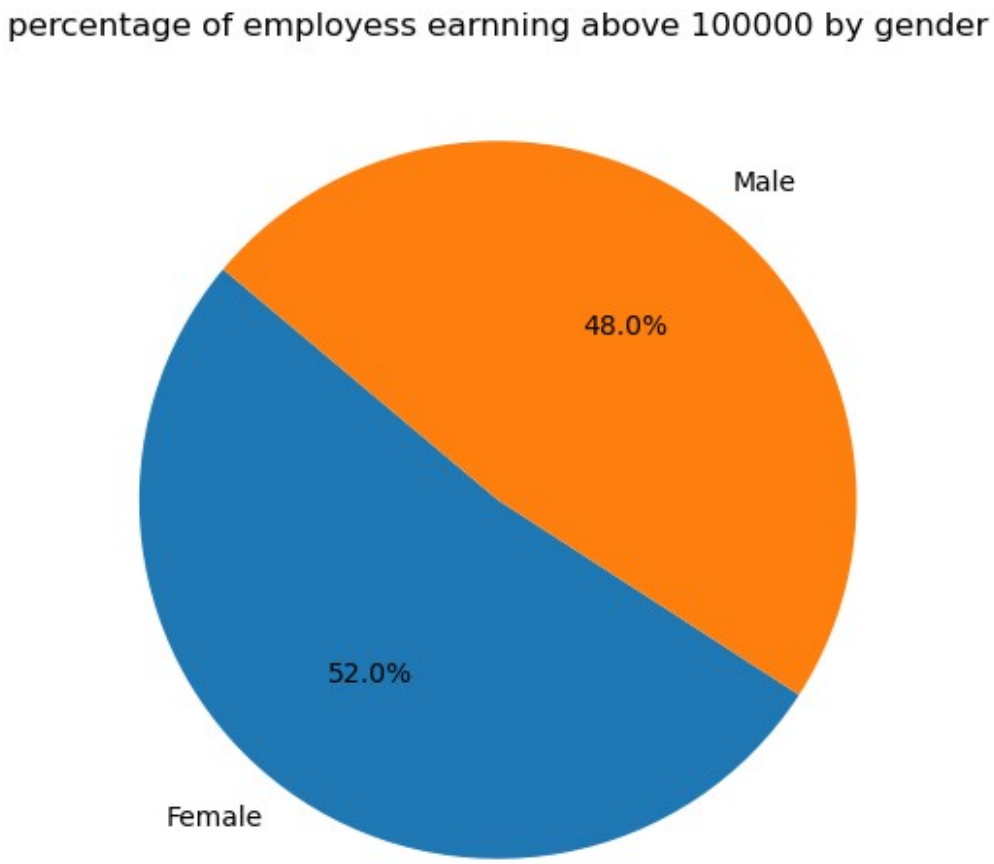
Excercise! group by job and find the mean

Filtering data for salaries > 100000 and creating a pie chart

```
new_df = df[["Gender", "Annual Salary"]].copy()
filtered_new_df = new_df[new_df["Annual Salary"]>100000]
group_by_gender = filtered_new_df.groupby("Gender").size()
print(group_by_gender)

colors_for_chart = ['#1f77b4', '#ff7f0e']
plt.figure(figsize=(6, 6))
plt.pie(group_by_gender, labels=group_by_gender.index , colors=
colors_for_chart, autopct='%1.1f%%', startangle=140)
plt.title('percentage of employess earnning above 100000 by gender')
plt.show()

Gender
Female    237
Male      219
dtype: int64
```



Counting Bonuses based on Ethnicities

```
ethnicity_count = df["Ethnicity"].value_counts()
print(ethnicity_count)

group_by_ethnicity = df.groupby("Ethnicity")
avg_bonuses = group_by_ethnicity["Bonus %"].mean()
print(avg_bonuses)
```

```
Ethnicity
Asian        401
Caucasian    269
Latino       249
Black         74
Name: count, dtype: int64
Ethnicity
Asian        0.095860
Black        0.086892
Caucasian    0.082210
Latino       0.086048
Name: Bonus %, dtype: float64
```