```
#!/usr/bin/env python
# coding: utf-8

# ### Aqila_Habib(47162)
#
#
# ITC_300: Final Project
#
#
# Professor Ahmad Al-Janad

# # Employee Dataset
#     This data conatains the employees infromation from three major locations in India. This dataset consists of 9 columns and 4653 rows.This dataset is from 2012-2018 that discuss about locations, gender, payment tier, Age, Experience in current domain , leave, everbenched ,and years. This data is important because the columns is easier to analyze and plot. Additionally, I can uderstand what are the outcomes which I am searching for. For this dataset I used bar plot and density plot which is easy and helps me to share my knowldge based on the outcomes. Most impotantly, the challenges I faced was how to organize questions and build a new columns. The main purpose of analyzing this dataset made me enthusiastic to understand the qualification, job opportunities and above that how youngs are involved in day to day company's tasks in order to contribute helping their wonderful country to have a bright future for many generations ahead. Therefore, I choose this dataset to analyize it.

# In[ ]:


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
import seaborn as sns
sns.set(style = 'white', color_codes = True)



# In[69]:


df= pd.read_csv("Employee.csv")


# In[70]:


df.columns


# In[71]:


df.head(10)


# In[ ]:
```

```
#To find the total years involved in this dataset
df['JoiningYear'].unique()


# In[ ]:


df['JoiningYear'].unique()


# In[72]:


# to confirm the columns and rows of the dataset
df.shape


# In[73]:


# Finding the data type of the column
df['Age'].dtype


# ## Q1. What is the mimunimum age of employees?
# There are a lot of employees but I want to confirm what the lowest age of
employees in the dataset.

# In[290]:


#The youngest employee is 22 years old in the company.
minmunum_EmpAge= df['Age'].unique().min()
minmunum_EmpAge


# ## Question02. List down the names of the city that employees belong to?
# This codes list down all the employees from three locations available in the
dataset.

# In[291]:


# Employees belongs to three locations Bangalore, Pune and New Dehli
Emp_byLocation= df['City'].value_counts()
Emp_byLocation


# ## Question3. What is the average age of emaployees in this dataset?
#

# In[292]:


#Average age of employee are 31 who are all belong to youngers.
Average_age= df['Age'].unique().mean()
Average_age
```

```
# ## Question4. Who is the oldest employee with experience?
#

# In[96]:


#  the oldest employee is 41 years who has 2 years of experience in current domain
with holding master degree.
oldest_with_experience = df[df['ExperienceInCurrentDomain'] > 0].nlargest(1, 'Age')
oldest_with_experience


# ## Question5. list down employees with the hieghest salary tier in this dataset?
#

# In[343]:


# Identify the highest payment tier
highest_payment_tier = df['PaymentTier'].max()

# Filter the DataFrame to get employees with the highest payment tier
employees_highest_tier = df[df['PaymentTier'] == highest_payment_tier]
employees_highest_tier
# Display the result
#print("Employees with the highest payment tier:")
#print(employees_highest_tier)


# In[98]:


df.dropna(inplace=True)


# ## Question6: show the number of education status by gender?

# In[100]:


# most of the employees in this company are male whose education ratio in all
status is hiegher than female.
EducationGen= df.groupby('Gender')[['Education']].value_counts()
EducationGen


# ## Q7. What is the percentage of female over male in this dataset? show by plot

# In[349]:


grouped_df = df.groupby(['Gender', 'Education']).size().unstack()

# Plot grouped bar plot
fig, ax = plt.subplots(figsize=(10, 6))
grouped_df.plot(kind='bar', ax=ax, width=0.4)

plt.title('Level of Education Based on Gender')
plt.xlabel('Education Level')
```

```
plt.ylabel('Count')
plt.xlim(-1, 6)
plt.xticks(rotation= 0)
plt.legend(title='Gender', loc='upper right')
#plt.show()


# ## Q8. Show the education percentage fully by pie chart?
# This diagram shows that most of the employees are bechalorate and very least
number of employeers PHD holders.

# In[390]:


Comparison_Education = df['Education'].value_counts()

# Calculate the percentage
Comparison_Education_percentage = Comparison_Education/ Comparison_Education.sum()
* 100

# Plot a pie chart
plt.figure(figsize=(5, 3))
plt.pie(Comparison_Education_percentage,
labels=Comparison_Education_percentage.index, autopct='%1.1f%%', startangle=90,
colors=['lightcoral', 'lightgreen', 'blue'])
plt.title('Percentage ofEmployees with Education status')
plt.show()


# In[391]:


Comparison_Education = df['Gender'].value_counts()

# Calculate the percentage
Comparison_Education_percentage = Comparison_Education/ Comparison_Education.sum()
* 100

# Plot a pie chart
plt.figure(figsize=(5, 4))
plt.pie(Comparison_Education_percentage,
labels=Comparison_Education_percentage.index, autopct='%1.1f%%', startangle=90,
colors=['brown', 'blue'])
plt.title('Percentage of Employees with Education status')
plt.show()


# In[173]:


#Question9. list down the joining year employees joined the job?
df['JoiningYear'].mean()


# ## Question9. Show the percentage mployees who got different paymentTier
annually? show by plot

# In[353]:
```

```
df.groupby('JoiningYear')['Gender'].count()

fig, ax = plt.subplots(figsize=(10, 6))
grouped_df.plot(kind='bar', ax=ax, width=0.4)

plt.title('Joining anually Based on Gender')
plt.xlabel('Year')
plt.ylabel('Count')
plt.xlim(-1, 6)
plt.xticks(rotation= 0)
plt.legend(title='Gender', loc='upper right')
#plt.show()


# ## Question9. Show the number of employees joining the job anuulay?
#

# In[357]:


df.groupby('JoiningYear')['Gender'].count()


# ## Question10. which year has the highest year job availability? show by plot
# Based on the codes a large number of employees had job availability in 2017. I
used denisty plot which is used for numeric variable and clearly it shows the peak
of the dataset.

# In[370]:


#The chances of job was much accessible in 2018
HighestEmployers= df.groupby('JoiningYear')[['PaymentTier']].count()
#HighestEmployers.sort_values('Gender', ascending= False,  inplace = True)
HighestEmployers.max()


# In[213]:


years= HighestEmployers.index
years


# In[371]:


# The Graph directly give the exact result that the hieghest job availability was
in 2017 with max value 1108 employeers.
#However The
plt.figure(figsize=(14, 10))
X = np.arange(len(HighestEmployers))
plt.plot(X, HighestEmployers['PaymentTier'], '--d', linewidth=1.5, markersize=8,
label='Number of 10 employers')

# Assuming HighestEmployers.index returns the years
years = HighestEmployers.index
```

```
plt.xticks(X, years, rotation=45)

max_v = HighestEmployers['PaymentTier'].max()
min_v = HighestEmployers['PaymentTier'].min()

# Ensure that you set the yticks range appropriately
plt.yticks(np.arange(min_v, max_v + 1000, step=1000))

plt.xlabel('Year')
plt.ylabel('Payment Tier')
plt.title('Number of 10 Employers Over Years')

plt.legend()
plt.grid(True)
plt.show()


# ## Question11. Show the hieghest and percentage of currentexperience domain
achieved by gender? show by plot
#
# To answer this question first I take out the number of female and male
individually than I used pie chart to answer the question clearly.

# In[372]:


GenExperience= df.groupby('Gender')[['ExperienceInCurrentDomain']].count()
GenExperience


# In[247]:


df['ExperienceInCurrentDomain'].unique()


# In[393]:


Cuurent_Experience_domain = df['ExperienceInCurrentDomain'].value_counts()

# Calculate the percentage
Cuurent_Experience_domain_percentage = Cuurent_Experience_domain/
Cuurent_Experience_domain.sum() * 100

# Plot a pie chart
plt.figure(figsize=(8, 5))
plt.pie(Cuurent_Experience_domain_percentage,
labels=Cuurent_Experience_domain_percentage.index, autopct='%1.1f%%',
startangle=90, colors=['purple', 'green', 'red', 'lightblue', 'black', 'brown',
'blue'])
plt.title('Percentage of Employees with Education status')
plt.show()


# ## Question12. Show the hieghest currentexperience domain? by year
# To answer this question I used bar charts which shows every attributes in
different color based on locations and the year through which I can predict about
the result by individually.
```

```
# In[281]:


#step1. I just sort them and find the total number of locations.
df_bylocation = df
df_bylocation.City.value_counts(sort = True, ascending = False)


# In[380]:


# groupby this two elements.
annual_joining_by_location = df.groupby(['City',
'JoiningYear']).size().unstack(fill_value=0)

# Calculate the percentage by dividing each row by the sum of the row
percentage_by_location =
annual_joining_by_location.div(annual_joining_by_location.sum(axis=1), axis=0) *
100

# Plotting the data as a stacked bar plot
plt.figure(figsize=(10, 9))
percentage_by_location.plot(kind='bar')
plt.xlabel('Location')
plt.ylabel('Percentage of Annual Joining')
plt.title('Percentage of Annual Joining by Location')
plt.xticks(rotation= 0)
plt.legend(title='Joining Year', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()



# ## Question13. find the percentage of the locations where employees belong to?
# Pie chart is usually used to identify the percentage clearly as compare to bar
chart. Therefore, for this qualtative variable I used pie chart which clearly state
about the percentage by location.

# In[388]:


Location = df['City'].value_counts()

# Calculate the percentage
Location_percentage = Location/ Location.sum() * 100

# Plot a pie chart
plt.figure(figsize=(8, 5))
plt.pie(Location_percentage, labels=Location_percentage.index, autopct='%1.1f%%',
startangle=90, colors=['blue', 'red', 'green'])
plt.title('Percentage of Employees based on city')
plt.show()


# In[333]:


def Get_count_gender(row):
    gender = row["Gender"]
```

```
    age = row["Age"]

    if gender == "Female":
        return 1
    else:
        return 0



# In[328]:


df["count_gender"]= df.apply(Get_count_gender, axis= 1)


# In[330]:


df['count_gender'].count()


# In[381]:


def get_gender_label(row):
    gender = row["Gender"]
    if gender == "Female":
        return 1
    elif gender == "Male":
        return 0
    else:
        return "Unknown"

# Apply the function to each row and create a new column
df['Gender_Label'] = df.apply(get_gender_label, axis=1)



# In[382]:


df


# ## Conclusion and Future work: Summarize your results, the strengths and
shortcomings of your results, and speculate on how you might address these
shortcomings if given more time.
#
# This dataset consists of the emplyee from three major locations where most of the
employees are male and younger than 41 years old. This indicates that mostly male
are searching for job as compare to women and holding different educational
degrees. However this data set does not consists many columns but I tried that much
to analyzes in depth that I could. I solved that much that I can build questions on
all columns of the dataset.I think this dataset is not more analayzable to take
time.
#
# Thank you.

# In[ ]:
```

```
# In[ ]:
```