-- DA Final Project

# -\*- coding: utf-8 -\*-

"""1\_New\_DA\_Project.ipynb

Automatically generated by Colab.

Original file is located at

https://colab.research.google.com/drive/1FGuonhJ4iInGh90qXST9w5kJ3VSJhPo7

"""

from google.colab import files

uploaded = files.upload()

import pandas as pd

df = pd.read\_csv('Employee\_Attendance\_Productivity.csv')

df

df.head()

df.tail()

df.dtypes

df.isnull()

df.duplicated()

df.drop\_duplicates()

df.describe()

# Remove extraspace from Name column

df.Name = df.Name.str.strip()

# Remove extrapace from Weekday column

df.Weekday = df.Weekday.str.lstrip()

# Remove extrapace from Department column

df.Department = df.Department.str.strip()

# Remove extrapace from Department column

df.Gender = df.Gender.str.strip()

text\_cols = ['Status', 'Department', 'LeaveType', 'RemoteWork']

for col in text\_cols:

df[col] = df[col].str.strip().str.title()

print(df.to\_string())

df['Date'] = pd.to\_datetime(df['Date'], format='%d-%m-%Y', errors='coerce')

df['CheckIn'] = pd.to\_datetime(df['CheckIn'], format='%H:%M:%S', errors='coerce').dt.time

df['CheckOut'] = pd.to\_datetime(df['CheckOut'], format='%H:%M:%S', errors='coerce').dt.time

df

df.duplicated().sum() # To see if any complete duplicate rows exist

df.fillna({'LeaveType': 'Present'}, inplace=True)

df

"""\*\*Analysis of Data from the dataset\*\*"""

# 1. Employees with the Most Absences

absences = df[df['Status'] == 'Absent']

absent\_counts = absences['Employee ID'].value\_counts()

absent\_counts.head()

# 2. Average Working Hours per Employee

avg\_hours = df[df['Status'] == 'Present'].groupby('Employee ID')['HoursWorked'].mean()

avg\_hours.head()

# Average Hours Worked by Gender

df[df['Status'] == 'Present'].groupby('Gender')['HoursWorked'].mean()

#Total Absences by Gender

df[df['Status'] == 'Absent'].groupby('Gender').size()

# Average Productivity Score by Gender

df[df['Status'] == 'Present'].groupby('Gender')['ProductivityScore'].mean()

# Average Salary by Gender

df.groupby('Gender')['Salary'].mean()

# Gender-wise Summary Report

gender\_summary = df.groupby('Gender').agg(

Total\_Records=('Status', 'count'),

Total\_Absent=('Status', lambda x: (x == 'Absent').sum()),

Total\_Present=('Status', lambda x: (x == 'Present').sum()),

Avg\_Hours=('HoursWorked', 'mean'),

Avg\_Productivity=('ProductivityScore', 'mean')

)

print(gender\_summary)

# 1. Bar Chart – Average Hours Worked by Gender

import seaborn as sns

import matplotlib.pyplot as plt

avg\_hours = df[df['Status'] == 'Present'].groupby('Gender')['HoursWorked'].mean().reset\_index()

plt.figure(figsize=(5,5))

sns.barplot(data=avg\_hours, x='Gender', y='HoursWorked',palette='Set1')

plt.title('Average Hours Worked by Gender')

plt.ylabel('Avg Hours Worked')

plt.show()

# 2. Countplot – Total Absences by Gender

absent = df[df['Status'] == 'Absent']

plt.figure(figsize=(5,5))

sns.countplot(data=absent, x='Gender', palette='pastel')

plt.title('Total Absences by Gender')

plt.xlabel('Gender')

plt.ylabel('Number of Absences')

plt.show()

# 3. Bar Chart – Average Productivity Score by Gender

avg\_productivity = df[df['Status'] == 'Present'].groupby('Gender')['ProductivityScore'].mean().reset\_index()

plt.figure(figsize=(5,5))

sns.barplot(data=avg\_productivity, x='Gender', y='ProductivityScore', palette='Set1')

plt.title('Average Productivity Score by Gender')

plt.ylabel('Avg Productivity Score')

plt.show()

# 4. Boxplot – Distribution of Hours Worked by Gender

sns.boxplot(data=df[df['Status'] == 'Present'], x='Gender', y='HoursWorked', palette='dark')

plt.title('Hours Worked Distribution by Gender')

plt.figure(figsize=(5,5))

plt.show()

# 5. Pie Chart – Gender Distribution in Dataset

gender\_counts = df['Gender'].value\_counts()

plt.figure(figsize=(5,5))

plt.pie(gender\_counts, labels=gender\_counts.index, autopct='%1.1f%%', startangle=90, colors=sns.color\_palette('pastel'))

plt.title('Gender Distribution')

plt.axis('equal')

plt.show()

# 6. Clustered Bar Chart – Absences by Gender and Department

absences = df[df['Status'] == 'Absent'].groupby(['Department', 'Gender']).size().reset\_index(name='Absence\_Count')

plt.figure(figsize=(8,6))

sns.barplot(data=absences, x='Department', y='Absence\_Count', hue='Gender', palette='pastel')

plt.title('Total Absences by Gender & Department')

plt.xticks(rotation=45)

plt.show()

# 7.Bar Plot – Average Productivity Score by Department

avg\_prod = df[df['Status'] == 'Present'].groupby('Department')['ProductivityScore'].mean().reset\_index()

plt.figure(figsize=(6, 6))

ax = sns.barplot(data=avg\_prod, x='Department', y='ProductivityScore', palette='viridis')

for container in ax.containers:

ax.bar\_label(container, fmt='%.2f', label\_type='edge', padding=3)

plt.title('Average Productivity Score by Department')

plt.ylabel('Avg Productivity Score')

plt.xlabel('Department')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

# 8. Pie chart: Attendance Status Distribution

plt.figure(figsize=(5,5))

status\_counts = df['Status'].value\_counts()

colors = sns.color\_palette('pastel')

plt.pie(status\_counts, labels=status\_counts.index, autopct='%1.1f%%', startangle=140, colors=colors)

plt.title('Employee Attendance Status Distribution')

plt.tight\_layout()

plt.show()

# 9. Box Plot: Hours Worked by Department

plt.figure(figsize=(5,5))

sns.boxplot(x='Department', y='HoursWorked', data=df, palette='Set3')

plt.title('Hours Worked by Department')

plt.xlabel('Department')

plt.ylabel('Hours Worked')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

# Save as any name you like

df.to\_csv('Employee\_Attendance\_Productivity\_cleaned.csv', index=False)

# Download using the same file name

files.download('Employee\_Attendance\_Productivity\_cleaned.csv')

-- SQL Queries

# To display all the data from dataset

SELECT \* FROM employee\_attendance\_productivity\_cleaned;

-- Project Tasks

-- SQL: Identify employees with frequent absences .

-- 1. Identify Frequent Absentees

# Displays total no. of employee absent in 3 months

select \* from employee\_attendance\_productivity\_cleaned where Status = 'Absent';

# Displays the name , department and total no. of absent count of Employees

SELECT Name, Department, COUNT(\*) as AbsenceCount FROM employee\_attendance\_productivity\_cleaned WHERE Status = 'Absent'

GROUP BY Department, Name ORDER BY AbsenceCount DESC;

# Displays the total Absences from each department

SELECT Department, COUNT(\*) AS TotalAbsences FROM employee\_attendance\_productivity\_cleaned WHERE Status = 'Absent'

GROUP BY Department ORDER BY TotalAbsences DESC;

# Displays the Absences count of Gender

SELECT Gender ,COUNT(\*) as AbsenceCount FROM employee\_attendance\_productivity\_cleaned WHERE Status = 'Absent'

GROUP BY Gender ORDER BY AbsenceCount DESC;

# Displays details of absent employees

SELECT Name,Gender,Department,Salary,COUNT(\*) AS TotalAbsences FROM employee\_attendance\_productivity\_cleaned WHERE Status = 'Absent'

GROUP BY Name, Gender, Department, Salary ORDER BY TotalAbsences DESC;

-- SQL: Calculate average hours worked.

-- 2. Calculate average hours worked.

# Display total Average hours worked of all employees

SELECT AVG(HoursWorked) AS Average\_Hours\_Worked FROM employee\_attendance\_productivity\_cleaned ;

# Display Average hours worked for present employees based on gender

SELECT AVG(HoursWorked) AS Average\_Hours\_Worked FROM employee\_attendance\_productivity\_cleaned WHERE Gender = 'Female';

SELECT AVG(HoursWorked) AS Average\_Hours\_Worked FROM employee\_attendance\_productivity\_cleaned WHERE Gender = 'Male';

# Department wise average hours of work

SELECT Department,AVG(HoursWorked) AS Avg\_Hours\_Worked FROM employee\_attendance\_productivity\_cleaned group by Department;

# Displays Average Hours Worked per Employee based on Gender

SELECT Department, Name, Gender, AVG(HoursWorked) as AverageHoursWorked FROM employee\_attendance\_productivity\_cleaned where Department = 'Engineering' and Gender = 'Male' GROUP BY Department, Name;

# Displays the avg work hour status of present employee with respect to department

SELECT Department,Status,AVG(HoursWorked) AS AvgHoursWorked FROM employee\_attendance\_productivity\_cleaned WHERE Status = 'Present' group by Department;

# Displays the avg work hour status of absent employee with respect to department

SELECT Department,Status,AVG(HoursWorked) AS AvgHoursWorked FROM employee\_attendance\_productivity\_cleaned WHERE Status = 'Absent' group by Department;

# Display all details of employees based on average work hours

SELECT Department,`Employee ID` AS Employee\_ID,Name, Salary , Gender,AVG(HoursWorked) AS AverageHoursWorked FROM employee\_attendance\_productivity\_cleaned

WHERE Status = 'Present' GROUP BY Department, `Employee ID`, Name,Salary,Gender ORDER BY Department, AverageHoursWorked DESC limit 20;