import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import matplotlib.pyplot as plt

import plotly.express as px

import plotly.subplots as sp

import plotly.graph\_objects as go

import seaborn as sns

import random

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

for filename in filenames:

print(os.path.join(dirname, filename))

df = pd.read\_csv("/kaggle/input/job-postings-dataset/job\_data\_merged\_1.csv")

df.head()

df.drop("Unnamed: 0", axis = 1, inplace=True )

df.head()

df['Country'] = df['Location'].str.split(',').str[-1].str.strip()

df.head()

df.isnull().sum()

# Set the style for seaborn

sns.set(style="whitegrid")

# Plot histogram for job categories

plt.figure(figsize=(12, 8))

category\_counts = df['Category'].value\_counts()

sns.barplot(x=category\_counts.index, y=category\_counts, palette='viridis')

plt.title('Distribution of Job Postings Across Categories')

plt.xlabel('Job Category')

plt.ylabel('Count')

plt.xticks(rotation=45, ha='right')

plt.tight\_layout()

plt.show()

# Set the style for seaborn

sns.set(style="whitegrid")

# Plot histogram for job types

plt.figure(figsize=(10, 6))

type\_counts = df['Type'].value\_counts()

sns.barplot(x=type\_counts.index, y=type\_counts, palette='viridis')

plt.title('Distribution of Job Postings by Type')

plt.xlabel('Job Type')

plt.ylabel('Count')

plt.xticks(rotation=45, ha='right')

plt.tight\_layout()

plt.show()

# Set the style for seaborn

sns.set(style="whitegrid")

# Plot histogram for job Workplace

plt.figure(figsize=(10, 6))

type\_counts = df['Workplace'].value\_counts()

sns.barplot(x=type\_counts.index, y=type\_counts, palette='viridis')

plt.title('Distribution of Job Postings by Workplace')

plt.xlabel('Workplace')

plt.ylabel('Count')

plt.xticks(rotation=45, ha='right')

plt.tight\_layout()

plt.show()

sns.set(style="whitegrid")

# Plot histogram for job Country

plt.figure(figsize=(15, 6))

type\_counts = df['Country'].value\_counts()

sns.barplot(x=type\_counts.index, y=type\_counts, palette='viridis')

plt.title('Distribution of Job Postings by Country')

plt.xlabel('Job Type')

plt.ylabel('Count')

plt.xticks(rotation=45, ha='right')

plt.tight\_layout()

plt.show()

sns.set(style="whitegrid")

# Plot the top 10 most frequent Country locations

plt.figure(figsize=(12, 8))

top\_workplaces = df['Country'].value\_counts().nlargest(10)

sns.barplot(x=top\_workplaces.index, y=top\_workplaces, palette='viridis')

plt.title('Top 10 Most Frequent Country Locations')

plt.xlabel('Country Location')

plt.ylabel('Number of Job Postings')

plt.xticks(rotation=45, ha='right')

plt.tight\_layout()

plt.show()

# Set the style for seaborn

sns.set(style="whitegrid")

# Plot the distribution of job postings by department (Top 10)

plt.figure(figsize=(12, 8))

top\_departments = df['Department'].value\_counts().nlargest(10)

sns.barplot(x=top\_departments.index, y=top\_departments, palette='viridis')

plt.title('Distribution of Job Postings by Department (Top 10)')

plt.xlabel('Department')

plt.ylabel('Number of Job Postings')# Counting job postings by workplace

workplace\_counts = df['Workplace'].value\_counts()

# Creating a pie chart with Plotly Express

fig = px.pie(workplace\_counts, names=workplace\_counts.index, values=workplace\_counts.values,

title='Distribution of Job Postings by Workplace',

labels={'label': 'Workplace', 'value': 'Number of Job Postings'},

color\_discrete\_sequence=px.colors.qualitative.Set3, # Choose a color palette

hole=0.4, # Adjust the size of the center hole

)

fig.update\_traces(textinfo='percent+label', pull=[0.1] \* len(workplace\_counts), hoverinfo='percent+label') # Add percentage and label to each slice

fig.update\_layout(legend\_title\_text='Workplace', legend=dict(title=dict(text='Workplace')),

showlegend=True, margin=dict(l=0, r=0, b=0, t=40)) # Add legend and adjust layout

fig.show()

# Counting job postings by category

category\_counts = df['Category'].value\_counts()

# Creating a pie chart with Plotly Express

fig = px.pie(category\_counts, names=category\_counts.index, values=category\_counts.values,

title='Distribution of Job Postings by Category',

labels={'label': 'Category', 'value': 'Number of Job Postings'},

color\_discrete\_sequence=px.colors.qualitative.Set1, # Choose a color palette

hole=0.4, # Adjust the size of the center hole

)

fig.update\_traces(textinfo='percent+label', pull=[0.1] \* len(category\_counts), hoverinfo='percent+label') # Add percentage and label to each slice

fig.update\_layout(legend\_title\_text='Category', legend=dict(title=dict(text='Category')),

showlegend=True, margin=dict(l=0, r=0, b=0, t=40)) # Add legend and adjust layout

fig.show()

# Counting job postings by type

type\_counts = df['Type'].value\_counts()

# Creating a pie chart with Plotly Express

fig = px.pie(type\_counts, names=type\_counts.index, values=type\_counts.values,

title='Distribution of Job Postings by Type',

labels={'label': 'Type', 'value': 'Number of Job Postings'},

color\_discrete\_sequence=px.colors.qualitative.Set2, # Choose a color palette

hole=0.4, # Adjust the size of the center hole

)

fig.update\_traces(textinfo='percent+label', pull=[0.1] \* len(type\_counts), hoverinfo='percent+label') # Add percentage and label to each slice

fig.update\_layout(legend\_title\_text='Type', legend=dict(title=dict(text='Type')),

showlegend=True, margin=dict(l=0, r=0, b=0, t=40)) # Add legend and adjust layout

fig.show()

df['Category'].unique()

BA\_df = df[df['Category'] == 'Business Analyst']

cloud\_df = df[df['Category'] == 'Cloud']

DS\_df = df[df['Category'] == 'Data Scientist']

HR\_df = df[df['Category'] == 'HR']

SD\_df = df[df['Category'] == 'Software Developer']

UI\_df = df[df['Category'] == 'UI/UX']

BA\_df.isnull().sum()

def visualize\_bar\_plot(dataframe, column\_name):

"""

Visualize a bar plot for the frequency of values in a specific column.

Parameters:

- dataframe: pandas DataFrame

- column\_name: str, the name of the column to visualize

"""

plt.figure(figsize=(10, 6))

sns.countplot(x=column\_name, data=dataframe, palette='viridis')

plt.title(f'Frequency of {column\_name} for {dataframe["Category"].iloc[0]} Position')

plt.xlabel(column\_name)

plt.ylabel('Frequency')

plt.xticks(rotation=45, ha='right') # Adjust rotation for better visibility

plt.show()

visualize\_bar\_plot(BA\_df, 'Workplace')

visualize\_bar\_plot(BA\_df, 'Type')

visualize\_bar\_plot(BA\_df, 'Country')

plt.figure(figsize=(18, 5))

sns.countplot(x='Department', data=BA\_df, palette='viridis')

plt.title(f'Frequency of Department for {BA\_df["Category"].iloc[0]} Position')

plt.xlabel('Department')

plt.ylabel('Frequency')

plt.xticks(rotation=45, ha='right', fontsize=10) # Adjust rotation and fontsize

plt.tight\_layout() # Adjust layout for better spacing between subplots

plt.show()

fig = px.bar(BA\_df, x='Department', title=f'Frequency of Department for {BA\_df["Category"].iloc[0]} Position', labels={'Department': 'Frequency'}, height=400)

fig.update\_layout(xaxis=dict(tickangle=-45, tickfont=dict(size=10)), margin=dict(l=0, r=0, t=30, b=0))

fig.show()

cloud\_df.isnull().sum()

visualize\_bar\_plot(cloud\_df, 'Workplace')

visualize\_bar\_plot(cloud\_df, 'Type')

visualize\_bar\_plot(cloud\_df, 'Country')

plt.figure(figsize=(18, 5))

sns.countplot(x='Department', data=cloud\_df, palette='viridis')

plt.title(f'Frequency of Department for {cloud\_df["Category"].iloc[0]} Position')

plt.xlabel('Department')

plt.ylabel('Frequency')

plt.xticks(rotation=45, ha='right', fontsize=10) # Adjust rotation and fontsize

plt.tight\_layout() # Adjust layout for better spacing between subplots

plt.show()

fig = px.bar(cloud\_df, x='Department', title=f'Frequency of Department for {cloud\_df["Category"].iloc[0]} Position', labels={'Department': 'Frequency'}, height=400)

fig.update\_layout(xaxis=dict(tickangle=-45, tickfont=dict(size=10)), margin=dict(l=0, r=0, t=30, b=0))

fig.show()

DS\_df.isnull().sum()

visualize\_bar\_plot(DS\_df, 'Workplace')

visualize\_bar\_plot(DS\_df, 'Type')

visualize\_bar\_plot(DS\_df, 'Country')

plt.figure(figsize=(18, 5))

sns.countplot(x='Department', data=DS\_df, palette='viridis')

plt.title(f'Frequency of Department for {DS\_df["Category"].iloc[0]} Position')

plt.xlabel('Department')

plt.ylabel('Frequency')

plt.xticks(rotation=45, ha='right', fontsize=10) # Adjust rotation and fontsize

plt.tight\_layout() # Adjust layout for better spacing between subplots

plt.show()

fig = px.bar(DS\_df, x='Department', title=f'Frequency of Department for {DS\_df["Category"].iloc[0]} Position', labels={'Department': 'Frequency'}, height=400)

fig.update\_layout(xaxis=dict(tickangle=-45, tickfont=dict(size=10)), margin=dict(l=0, r=0, t=30, b=0))

fig.show()

visualize\_bar\_plot(HR\_df, 'Workplace')

visualize\_bar\_plot(HR\_df, 'Type')

visualize\_bar\_plot(HR\_df, 'Country')

plt.figure(figsize=(18, 5))

sns.countplot(x='Department', data=HR\_df, palette='viridis')

plt.title(f'Frequency of Department for {HR\_df["Category"].iloc[0]} Position')

plt.xlabel('Department')

plt.ylabel('Frequency')

plt.xticks(rotation=45, ha='right', fontsize=10) # Adjust rotation and fontsize

plt.tight\_layout() # Adjust layout for better spacing between subplots

plt.show()

fig = px.bar(HR\_df, x='Department', title=f'Frequency of Department for {HR\_df["Category"].iloc[0]} Position', labels={'Department': 'Frequency'}, height=400)

fig.update\_layout(xaxis=dict(tickangle=-45, tickfont=dict(size=10)), margin=dict(l=0, r=0, t=30, b=0))

fig.show()

SD\_df.isnull().sum()

visualize\_bar\_plot(SD\_df, 'Workplace')

visualize\_bar\_plot(SD\_df, 'Type')

visualize\_bar\_plot(SD\_df, 'Country')

plt.figure(figsize=(18, 5))

sns.countplot(x='Department', data=SD\_df, palette='viridis')

plt.title(f'Frequency of Department for {SD\_df["Category"].iloc[0]} Position')

plt.xlabel('Department')

plt.ylabel('Frequency')

plt.xticks(rotation=45, ha='right', fontsize=10) # Adjust rotation and fontsize

plt.tight\_layout() # Adjust layout for better spacing between subplots

plt.show()

fig = px.bar(SD\_df, x='Department', title=f'Frequency of Department for {SD\_df["Category"].iloc[0]} Position', labels={'Department': 'Frequency'}, height=400)

fig.update\_layout(xaxis=dict(tickangle=-45, tickfont=dict(size=10)), margin=dict(l=0, r=0, t=30, b=0))

fig.show()

UI\_df.isnull().sum()

visualize\_bar\_plot(UI\_df, 'Workplace')

visualize\_bar\_plot(UI\_df, 'Type')

visualize\_bar\_plot(UI\_df, 'Country')

plt.figure(figsize=(18, 5))

sns.countplot(x='Department', data=UI\_df, palette='viridis')

plt.title(f'Frequency of Department for {UI\_df["Category"].iloc[0]} Position')

plt.xlabel('Department')

plt.ylabel('Frequency')

plt.xticks(rotation=45, ha='right', fontsize=10) # Adjust rotation and fontsize

plt.tight\_layout() # Adjust layout for better spacing between subplots

plt.show()

fig = px.bar(UI\_df, x='Department', title=f'Frequency of Department for {UI\_df["Category"].iloc[0]} Position', labels={'Department': 'Frequency'}, height=400)

fig.update\_layout(xaxis=dict(tickangle=-45, tickfont=dict(size=10)), margin=dict(l=0, r=0, t=30, b=0))

fig.show()

colors = ['darkgray', 'darkgreen', 'darkblue', 'darkred','mediumblue', 'yellow']

def visualize\_factors(dataframe, colors):

# Create subplot with titles

fig = sp.make\_subplots(

rows=2, cols=2,

subplot\_titles=[

f'Frequency of Workplace for {dataframe["Category"].iloc[0]} Position',

f'Frequency of Type for {dataframe["Category"].iloc[0]} Position',

f'Frequency of Country for {dataframe["Category"].iloc[0]} Position',

f'Frequency of Department for {dataframe["Category"].iloc[0]} Position'

],

row\_heights=[0.5, 0.5]

)

# Randomly select a color for each subplot

subplot\_colors = random.sample(colors, 4)

# Subplot 1

fig.add\_trace(go.Bar(x=dataframe['Workplace'].value\_counts().index, y=dataframe['Workplace'].value\_counts(), marker=dict(color=subplot\_colors[0])),

row=1, col=1)

# Subplot 2

fig.add\_trace(go.Bar(x=dataframe['Type'].value\_counts().index, y=dataframe['Type'].value\_counts(), marker=dict(color=subplot\_colors[1])),

row=1, col=2)

# Subplot 3

fig.add\_trace(go.Bar(x=dataframe['Country'].value\_counts().index, y=dataframe['Country'].value\_counts(), marker=dict(color=subplot\_colors[2])),

row=2, col=1)

fig.update\_xaxes(title\_text='Country', tickangle=-45, tickfont=dict(size=10), row=2, col=1)

# Subplot 4

fig.add\_trace(go.Bar(x=dataframe['Department'].value\_counts().index, y=dataframe['Department'].value\_counts(), marker=dict(color=subplot\_colors[3])),

row=2, col=2)

fig.update\_xaxes(title\_text='Department', tickangle=-45, tickfont=dict(size=10), row=2, col=2)

# Update layout

fig.update\_layout(title\_text=f'Frequency of Different Factors for {dataframe["Category"].iloc[0]} Position', height=1000, showlegend=False)

# Show the interactive subplot

fig.show()

visualize\_factors(BA\_df, colors)

visualize\_factors(cloud\_df, colors)

visualize\_factors(DS\_df, colors)

visualize\_factors(HR\_df, colors)

visualize\_factors(SD\_df, colors)

visualize\_factors(UI\_df, colors)

plt.xticks(rotation=45, ha='right')

plt.tight\_layout()

plt.show()