It looks like you're working on a Data Science assignment involving both the *Flight Price* and *Google Playstore* datasets. Below are the steps and methodologies you can use to answer the questions related to the datasets:

**Flight Price Dataset**

**Q1. Load the flight price dataset and examine its dimensions.**

* **Solution**: Use the pandas library to load the dataset and examine its shape using the .shape attribute.

import pandas as pd

flight\_data = pd.read\_csv('flight\_price.csv')

print(flight\_data.shape)

**Q2. What is the distribution of flight prices in the dataset? Create a histogram to visualize the distribution.**

* **Solution**: Use matplotlib or seaborn to plot the histogram of the flight prices.

import matplotlib.pyplot as plt

import seaborn as sns

sns.histplot(flight\_data['Price'], kde=True)

plt.title('Distribution of Flight Prices')

plt.xlabel('Price')

plt.ylabel('Frequency')

plt.show()

**Q3. What is the range of prices in the dataset? What is the minimum and maximum price?**

* **Solution**: Use .min() and .max() functions to find the minimum and maximum price in the dataset.

min\_price = flight\_data['Price'].min()

max\_price = flight\_data['Price'].max()

print(f"Minimum price: {min\_price}, Maximum price: {max\_price}")

**Q4. How does the price of flights vary by airline? Create a boxplot to compare the prices of different airlines.**

* **Solution**: Use seaborn to create a boxplot comparing prices by airline.

sns.boxplot(x='Airline', y='Price', data=flight\_data)

plt.title('Flight Price Distribution by Airline')

plt.xlabel('Airline')

plt.ylabel('Price')

plt.show()

**Q5. Are there any outliers in the dataset? Identify any potential outliers using a boxplot and describe how they may impact your analysis.**

* **Solution**: Boxplots will show outliers as points outside the "whiskers". You can interpret the results and mention how these outliers could skew your analysis.

sns.boxplot(x='Price', data=flight\_data)

plt.title('Boxplot of Flight Prices')

plt.show()

**Q6. Identify the peak travel season.**

* **Solution**: Analyze features like departure month, holidays, or weekends. Use a time series or bar plot to identify peak travel periods.

flight\_data['Departure\_month'] = pd.to\_datetime(flight\_data['Departure\_date']).dt.month

sns.barplot(x='Departure\_month', y='Price', data=flight\_data)

plt.title('Average Price by Departure Month')

plt.xlabel('Month')

plt.ylabel('Average Price')

plt.show()

**Q7. Analyze trends in flight prices.**

* **Solution**: Analyze time-related features (e.g., day of the week, season) to identify trends and use line plots to visualize them.

sns.lineplot(x='Date', y='Price', data=flight\_data)

plt.title('Flight Prices Trend Over Time')

plt.xlabel('Date')

plt.ylabel('Price')

plt.show()

**Q8. Identify factors that affect flight prices.**

* **Solution**: Analyze features such as distance, flight duration, airline, etc., using correlation heatmaps and regression models.

sns.heatmap(flight\_data.corr(), annot=True)

plt.title('Correlation Matrix of Features')

plt.show()

**Google Playstore Dataset**

**Q9. Load the Google Playstore dataset and examine its dimensions.**

* **Solution**: Use the .shape attribute to examine the dimensions.

playstore\_data = pd.read\_csv('google\_playstore.csv')

print(playstore\_data.shape)

**Q10. How does the rating of apps vary by category? Create a boxplot to compare the ratings of different app categories.**

* **Solution**: Create a boxplot comparing the ratings of different app categories.

sns.boxplot(x='Category', y='Rating', data=playstore\_data)

plt.title('App Ratings by Category')

plt.xlabel('Category')

plt.ylabel('Rating')

plt.show()

**Q11. Are there any missing values in the dataset?**

* **Solution**: Use .isnull() to check for missing values and summarize their count.

missing\_values = playstore\_data.isnull().sum()

print(missing\_values)

**Q12. What is the relationship between the size of an app and its rating?**

* **Solution**: Create a scatter plot to visualize the relationship.

sns.scatterplot(x='Size', y='Rating', data=playstore\_data)

plt.title('Relationship Between Size and Rating')

plt.xlabel('Size')

plt.ylabel('Rating')

plt.show()

**Q13. How does the type of app affect its price?**

* **Solution**: Create a bar chart to compare the average prices by app type (free vs paid).

sns.barplot(x='Type', y='Price', data=playstore\_data)

plt.title('Average Price by App Type')

plt.xlabel('Type')

plt.ylabel('Price')

plt.show()

**Q14. What are the top 10 most popular apps in the dataset?**

* **Solution**: Use the Installs column to sort and display the top 10 most installed apps.

top\_apps = playstore\_data.sort\_values('Installs', ascending=False).head(10)

print(top\_apps[['App', 'Installs']])

**Q15. Identify the most popular app categories.**

* **Solution**: Use the Category and Installs columns to calculate the most popular categories.

popular\_categories = playstore\_data.groupby('Category')['Installs'].sum().sort\_values(ascending=False)

print(popular\_categories)

**Q16. Identify the most successful app developers.**

* **Solution**: Use the Developer column to group apps and calculate success metrics, like average ratings or installs.

top\_developers = playstore\_data.groupby('Developer')['Installs'].sum().sort\_values(ascending=False)

print(top\_developers)

**Q17. Identify the best time to launch a new app.**

* **Solution**: Analyze app trends over time, considering seasonal spikes, using visualizations like line plots or heatmaps based on app installation trends.

sns.lineplot(x='Year', y='Installs', data=playstore\_data)

plt.title('Trend of Installs Over Time')

plt.xlabel('Year')

plt.ylabel('Installs')

plt.show()

These steps should help you answer the questions in your assignment. Be sure to clean the datasets if necessary (e.g., handling missing values, converting data types) before starting your analysis.