**50 Challenging Questions for Data Analysis Practice**

**Data Manipulation & Cleaning (Pandas & NumPy):**

1. Identify and handle missing values in the dataset.
2. Convert the order\_datetime column to a proper datetime format and extract the year, month, and day.
3. Remove duplicate entries based on order\_id and product\_id.
4. Standardize all text columns by converting them to lowercase and stripping extra spaces.
5. Replace incorrect values in the rating column, ensuring they fall between 1 and 5.
6. Find and replace all instances of a particular product category name with a corrected version.
7. Create a new column categorizing delivery time into "Fast" (<30 mins), "Medium" (30-60 mins), and "Slow" (>60 mins).
8. Check for outliers in the price column using the IQR method.
9. Normalize the price column using Min-Max scaling.
10. Calculate the total number of unique customers in the dataset.

**Exploratory Data Analysis (EDA):**

11. Find the top 5 most frequently ordered products.

12. Identify the average order value per customer segment.

13. Calculate the percentage of delayed deliveries.

14. Find the delivery partner with the highest average rating.

15. Determine the most popular payment method.

16. Identify the areas with the highest number of orders.

17. Find the most common feedback category given by customers.

18. Compute the average delivery time for each product category.

19. Determine the correlation between price and rating.

20. Identify the customer with the highest lifetime order value.

**Data Aggregation & Grouping:**

21. Find the total sales revenue generated by each product category.

22. Determine the total number of orders placed each month.

23. Compute the average rating for each product category.

24. Find the top 3 most ordered products in each area.

25. Group customers based on their total spending and classify them as "Low", "Medium", or "High" value customers.

26. Calculate the percentage contribution of each payment method to total sales.

27. Find the most common reason for delivery delays.

28. Analyze the trend of the total number of orders over time.

29. Determine the feedback category with the lowest average rating.

30. Identify the most profitable product by subtracting cost (if available) from price.

**Advanced Data Analysis & Feature Engineering:**

31. Create a new feature indicating whether an order was placed on a weekend or weekday.

32. Compute the moving average of sales over the last 7 days.

33. Create a pivot table showing average order value per payment method per customer segment.

34. Generate a feature representing the time difference between promised and actual delivery time.

35. Find the proportion of returning customers.

36. Identify the top 5 products with the highest refund or return rates (if such data exists).

37. Perform customer segmentation using k-means clustering based on order value and frequency.

38. Generate a ranking system for products based on their total sales and average rating.

39. Compute the average order size per day of the week.

40. Use NumPy to efficiently compute the median delivery time across all orders.

**Data Visualization (Matplotlib & Seaborn):**

41. Create a histogram showing the distribution of order values.

42. Generate a bar chart of the top 10 most ordered products.

43. Create a line plot showing the trend of daily sales.

44. Visualize the correlation matrix of numerical columns using a heatmap.

45. Plot a scatter plot between price and rating to identify any trends.

46. Create a boxplot of delivery time categorized by area.

47. Generate a pie chart showing the distribution of feedback categories.

48. Use a violin plot to visualize the distribution of order values per customer segment.

49. Create a stacked bar chart showing the proportion of payment methods used per area.

50. Generate a time series plot showing the change in order count over months.

These questions should help you practice various aspects of data manipulation, analysis, and visualization, leveraging your skills in Pandas, NumPy, Matplotlib, and Seaborn.