NPL Season 1 ANALYSIS

Section 1: General Practices (25 Questions)

1. Load the 'npl_ball_by_ball.csv' dataset into a pandas DataFrame and display the first five rows.
2. Check for any missing values in the dataset and report the count per column.
3. Convert the 'date' column to datetime format and extract year and month into separate columns.
4. Create a pivot table showing the total runs scored by each 'batting_team'.
5. Create a crosstab to show the frequency of each 'wicket_kind' per 'bowling_type'.
6. Group the data by 'bowler' and compute total runs conceded.
7. Filter out all dot balls and count them by each bowler.
8. Sort the dataset by 'match_id', then by 'inning', and then by 'over'.
9. Use multi-level indexing with 'match_id', 'inning', and 'over' and display a sample group.
10. Drop rows where 'runs_total' is null or missing.
11. Fill missing 'wicket_kind' entries with 'None'.
12. Merge the DataFrame with itself to get next delivery info for each row.
13. Filter all deliveries bowled by a particular bowler (e.g., 'Sompal Kami').
14. Create a column indicating whether the delivery was a dot ball.
15. Count the number of boundaries (4s and 6s) hit by each batsman.
16. Create a new column showing cumulative score per match using groupby and cumsum.
17. Identify how many unique venues are present in the dataset.
18. Get the top 5 batsmen with the highest total runs.
19. Group data by 'match_id' and 'inning' to compute total runs per innings.
20. Join this dataset with a dummy DataFrame of player roles based on 'batter'.
21. Filter data where the bowler is right-arm off-spinner and batsman is left-handed.
22. Rename the column 'runs_total' to 'total_runs_scored'.
23. Create a new column that flags whether a delivery was during powerplay overs.
24. Reset the index of the DataFrame after sorting.
25. Calculate the average number of balls bowled per match.

Section 2: Cricket Performance and Match Analysis (60 Questions)

26. Calculate each bowler's strike rate (total balls bowled divided by wickets taken).
27. Compute the bowling average (total runs conceded divided by wickets taken) for each bowler.
28. Determine the economy rate for each bowler (runs conceded per over).
29. Identify the top 5 bowlers with the best economy rates (minimum 5 overs).

30. Calculate the strike rate of each batsman (runs per 100 balls faced).
31. Determine the batting average (total runs divided by number of dismissals) for each batsman.
32. Identify the top 5 batsmen based on batting average (min. 30 balls faced).
33. Analyze bowler vs batsman matchups: compute total runs, balls faced, and dismissals.
34. Identify bowler-batsman pairs where batsman's strike rate exceeds 150.
35. Compare runs scored by right-handed vs. left-handed batsmen across all matches.
36. Analyze average runs conceded by pace bowlers vs. spin bowlers.
37. Compare right-handed batsmen's performance against pace and spin.
38. Determine which batting side (R/L) scores better against off-spinners.
39. Identify bowlers with better strike rates in death overs (17-20) vs powerplay (1-6).
40. Generate per-match bowling figures for each bowler (overs-runs-wickets format).
41. List batsmen who hit at least one boundary in more than 50% of innings.
42. Compute the dot ball percentage for each bowler.
43. Find players who were dismissed most often in the final 5 overs.
44. Calculate the average partnership runs between wickets for each team.
45. Compare home vs away performance of each team using total runs scored.
46. Identify which venues had the highest average score per innings.
47. Calculate total runs scored in powerplay, middle, and death overs for each match.
48. Identify batsmen with fastest 50s (least balls faced to reach 50).
49. Find bowlers with most wickets in a single match (best bowling spell).
50. Derive the win/loss record for each team from match outcomes.
51. Calculate standard deviation in runs conceded per over by each bowler.
52. Analyze a batsman's average runs against left-arm bowlers.
53. Compare a batsman's performance across different venues.
54. For each match, calculate required run rate vs actual run rate in 2nd innings.
55. Identify clutch knocks: batsmen scoring 30+ in the last 5 overs.
56. Compare economy rate of bowlers in pressure overs vs. rest of the innings.
57. Create a frequency table of all dismissal types ('wicket_kind').
58. Find teams where spinners contributed more than 50% of total wickets.
59. Analyze correlation between over number and average runs per delivery.
60. Use a rolling average (window=20 deliveries) to analyze bowler form over season.
61. Identify bowlers whose economy rate improved as the season progressed.
62. Calculate average runs scored by batsmen when playing as openers.

63. Identify matches with the highest total aggregate runs.
64. Create a boxplot-ready summary of runs per over for all innings.
65. Evaluate the impact of extras (runs_extras) on innings total.
66. Identify batsmen most frequently dismissed via a particular 'wicket_kind'.
67. Compute bowler wicket rates under low (<=150) vs high (>150) target conditions.
68. Rank all-rounders based on batting avg minus bowling avg (custom metric).
69. Analyze average delivery count per partnership before a wicket falls.
70. Find percentage of matches successfully chased in second innings.
71. Evaluate toss influence on match result (if toss data available).
72. Compare dot-ball frequency between spin and pace bowlers.
73. Compare run-scoring patterns against left-arm orthodox vs. unorthodox spin.
74. Determine powerplay wicket frequency for all bowlers.
75. List matches that were decided in the final over and key performers.
76. Compare average first innings vs second innings scores across matches.
77. Compute per-match net run rate and rank teams by it.
78. Identify batsmen with highest boundary percentage (boundaries per runs).
79. Analyze final wicket partnerships and their impact on total score.
80. Create a multi-level groupby summary: total runs per match and inning.
81. Identify the biggest win margins (by runs) in completed matches.
82. Find bowlers with negative impact (more runs conceded than wickets).
83. Compare batsmen's average against pace vs spin (check >=10% difference).
84. (Optional) Correlate scoring with venue altitude if such data exists.
85. Rank all-rounders using combined batting and bowling performance metrics.

Section 3: Data Visualization and Interpretation (15 Questions)

86. Create a bar chart showing total wickets taken by each bowling type.
87. Generate a line plot of cumulative runs over overs for a selected match.
88. Create a scatter plot comparing runs_batter and runs_extras for all deliveries.
89. Use a heatmap to show average runs conceded per over by bowling type.
90. Generate a boxplot comparing runs_total in powerplay vs. death overs.
91. Propose a suitable plot to compare batsman averages and justify your choice.
92. Design a plot to show which overs have the highest dot ball frequency.
93. Visualize the top 5 highest scoring deliveries and interpret the outcome.
94. Plot average partnership lengths per team and analyze the trend.

95. Compare scoring patterns of two rival teams using a visualization of your choice.

96. Create a plot showing run rate trends across matches over the season.

97. Visualize bowler performance under pressure (last 5 overs) and discuss results.

98. Chart extras conceded by each team and explain field discipline insights.

99. Display a comparison of dismissal types across different venues.

100. Create a visual ranking of top all-rounders using a custom metric.