ipl-data-analysis-spark

August 8, 2024

[0]:
```
spark
```

[0]:
```
from pyspark.sql.types import StructField, StructType, IntegerType, StringType,␣
↪BooleanType, DateType, DecimalType
from pyspark.sql.functions import col, when, sum, avg, row_number
from pyspark.sql.window import Window
```

[0]:
```
from pyspark.sql import SparkSession

#create session
spark = SparkSession.builder.appName("IPL Data Analysis").getOrCreate()
```

[0]:
```
spark
```

[0]:
```
ball_by_ball_schema = StructType([
    StructField("match_id", IntegerType(), True),
    StructField("over_id", IntegerType(), True),
    StructField("ball_id", IntegerType(), True),
    StructField("innings_no", IntegerType(), True),
    StructField("team_batting", StringType(), True),
    StructField("team_bowling", StringType(), True),
    StructField("striker_batting_position", IntegerType(), True),
    StructField("extra_type", StringType(), True),
    StructField("runs_scored", IntegerType(), True),
    StructField("extra_runs", IntegerType(), True),
    StructField("wides", IntegerType(), True),
    StructField("legbyes", IntegerType(), True),
    StructField("byes", IntegerType(), True),
    StructField("noballs", IntegerType(), True),
    StructField("penalty", IntegerType(), True),
    StructField("bowler_extras", IntegerType(), True),
    StructField("out_type", StringType(), True),
    StructField("caught", BooleanType(), True),
    StructField("bowled", BooleanType(), True),
    StructField("run_out", BooleanType(), True),
    StructField("lbw", BooleanType(), True),
    StructField("retired_hurt", BooleanType(), True),
```

```
    StructField("stumped", BooleanType(), True),
    StructField("caught_and_bowled", BooleanType(), True),
    StructField("hit_wicket", BooleanType(), True),
    StructField("obstructingfeild", BooleanType(), True),
    StructField("bowler_wicket", BooleanType(), True),
    StructField("match_date", DateType(), True),
    StructField("season", IntegerType(), True),
    StructField("striker", IntegerType(), True),
    StructField("non_striker", IntegerType(), True),
    StructField("bowler", IntegerType(), True),
    StructField("player_out", IntegerType(), True),
    StructField("fielders", IntegerType(), True),
    StructField("striker_match_sk", IntegerType(), True),
    StructField("strikersk", IntegerType(), True),
    StructField("nonstriker_match_sk", IntegerType(), True),
    StructField("nonstriker_sk", IntegerType(), True),
    StructField("fielder_match_sk", IntegerType(), True),
    StructField("fielder_sk", IntegerType(), True),
    StructField("bowler_match_sk", IntegerType(), True),
    StructField("bowler_sk", IntegerType(), True),
    StructField("playerout_match_sk", IntegerType(), True),
    StructField("battingteam_sk", IntegerType(), True),
    StructField("bowlingteam_sk", IntegerType(), True),
    StructField("keeper_catch", BooleanType(), True),
    StructField("player_out_sk", IntegerType(), True),
    StructField("matchdatesk", DateType(), True)
])
```

[0]:
```
ball_by_ball_df = spark.read.schema(ball_by_ball_schema).format("csv").
↪option("header","true").load("s3://ipl-data-analysis-project/Ball_By_Ball.
↪csv")
```

[0]:
```
match_schema = StructType([
    StructField("match_sk", IntegerType(), True),
    StructField("match_id", IntegerType(), True),
    StructField("team1", StringType(), True),
    StructField("team2", StringType(), True),
    StructField("match_date", DateType(), True),
    StructField("season_year", IntegerType(), True),
    StructField("venue_name", StringType(), True),
    StructField("city_name", StringType(), True),
    StructField("country_name", StringType(), True),
    StructField("toss_winner", StringType(), True),
    StructField("match_winner", StringType(), True),
    StructField("toss_name", StringType(), True),
    StructField("win_type", StringType(), True),
    StructField("outcome_type", StringType(), True),
```

```
    StructField("manofmach", StringType(), True),
    StructField("win_margin", IntegerType(), True),
    StructField("country_id", IntegerType(), True)
])
match_df = spark.read.schema(match_schema).format("csv").
 ↪option("header","true").load("s3://ipl-data-analysis-project/Match.csv")
```

[0]:
```
player_schema = StructType([
    StructField("player_sk", IntegerType(), True),
    StructField("player_id", IntegerType(), True),
    StructField("player_name", StringType(), True),
    StructField("dob", DateType(), True),
    StructField("batting_hand", StringType(), True),
    StructField("bowling_skill", StringType(), True),
    StructField("country_name", StringType(), True)
])

player_df = spark.read.schema(player_schema).format("csv").
 ↪option("header","true").load("s3://ipl-data-analysis-project/Player.csv")
```

[0]:
```
player_match_schema = StructType([
    StructField("player_match_sk", IntegerType(), True),
    StructField("playermatch_key", DecimalType(), True),
    StructField("match_id", IntegerType(), True),
    StructField("player_id", IntegerType(), True),
    StructField("player_name", StringType(), True),
    StructField("dob", DateType(), True),
    StructField("batting_hand", StringType(), True),
    StructField("bowling_skill", StringType(), True),
    StructField("country_name", StringType(), True),
    StructField("role_desc", StringType(), True),
    StructField("player_team", StringType(), True),
    StructField("opposit_team", StringType(), True),
    StructField("season_year", IntegerType(), True),
    StructField("is_manofthematch", BooleanType(), True),
    StructField("age_as_on_match", IntegerType(), True),
    StructField("isplayers_team_won", BooleanType(), True),
    StructField("batting_status", StringType(), True),
    StructField("bowling_status", StringType(), True),
    StructField("player_captain", StringType(), True),
    StructField("opposit_captain", StringType(), True),
    StructField("player_keeper", StringType(), True),
    StructField("opposit_keeper", StringType(), True)
])
```

```
player_match_df = spark.read.schema(player_match_schema).format("csv").
 ↪option("header","true").load("s3://ipl-data-analysis-project/Player_match.
 ↪csv")
```

[0]:
```
team_schema = StructType([
    StructField("team_sk", IntegerType(), True),
    StructField("team_id", IntegerType(), True),
    StructField("team_name", StringType(), True)
])

team_df = spark.read.schema(team_schema).format("csv").option("header","true").
 ↪load("s3://ipl-data-analysis-project/Team.csv")
```

[0]:
```
# Filter to include only valid deliveries (excluding extras like wides and no␣
 ↪balls for specific analyses)
ball_by_ball_df = ball_by_ball_df.filter((col("wides") == 0) &␣
 ↪(col("noballs")==0))

# Aggregation: Total and average runs scored in each match and inning
total_and_avg_runs = ball_by_ball_df.groupBy("match_id", "innings_no").agg(
    sum("runs_scored").alias("total_runs"),
    avg("runs_scored").alias("average_runs")
)
```

[0]:
```
# Window Function: Calculate running total of runs in each match for each over
windowSpec = Window.partitionBy("match_id","innings_no").orderBy("over_id")

ball_by_ball_df = ball_by_ball_df.withColumn(
    "running_total_runs",
    sum("runs_scored").over(windowSpec)
)
```

[0]:
```
# Conditional Column: Flag for high impact balls (either a wicket or more than␣
 ↪6 runs including extras)
ball_by_ball_df = ball_by_ball_df.withColumn(
    "high_impact",
    when((col("runs_scored") + col("extra_runs") > 6) | (col("bowler_wicket")␣
 ↪== True), True).otherwise(False)
)
```

[0]:
```
ball_by_ball_df.show(5)
```

```
+--------+-------+-------+----------+------------+------------+-----------------
-------+----------+-----------+----------+-----+-------+----+-------+-------+---
----------+-------------+------+------+-------+----+------------+-------+------
----------+----------+---------------+-------------+----------+------+-------+
----------+------+----------+--------+---------------+---------+-------------
```

```
-----+------------+---------------+---------+--------------+---------+------
-----------+-------------+-------------+-----------+------------+---------
-+-----------------+----------+
|match_id|over_id|ball_id|innings_no|team_batting|team_bowling|striker_batting_p
osition|extra_type|runs_scored|extra_runs|wides|legbyes|byes|noballs|penalty|bow
ler_extras|      out_type|caught|bowled|run_out| lbw|retired_hurt|stumped|caught
_and_bowled|hit_wicket|obstructingfeild|bowler_wicket|match_date|season|striker|
non_striker|bowler|player_out|fielders|striker_match_sk|strikersk|nonstriker_mat
ch_sk|nonstriker_sk|fielder_match_sk|fielder_sk|bowler_match_sk|bowler_sk|player
out_match_sk|battingteam_sk|bowlingteam_sk|keeper_catch|player_out_sk|matchdates
k|running_total_runs|high_impact|
+--------+-------+-------+----------+------------+------------+----------------
-------+----------+----------+----------+-----+-------+----+-------+-------+---
---------+-------------+------+------+-------+----+------------+-------+------
-----------+----------+---------------+-------------+----------+------+-------+
-----------+------+----------+--------+---------------+---------+-------------
-----+-------------+---------------+----------+--------------+---------+------
-----------+-------------+-------------+-----------+------------+---------
-+-----------------+----------+
|  335987|      1|      1|         1|           1|           2|
1|   legbyes|          0|         1|    0|      1|   0|      0|      0|
0|Not Applicable|  null|  null|   null|null|        null|   null|
null|      null|            null|         null|      null|  2008|      1|
2|    14|      null|    null|           12705|        0|             12706|
1|             -1|        -1|          12702|       13|               -1|
0|             1|        null|           0|       null|                0|
false|
|  335987|      1|      2|         1|           1|           2|
2| No Extras|          0|         0|    0|      0|   0|      0|      0|
0|Not Applicable|  null|  null|   null|null|        null|   null|
null|      null|            null|         null|      null|  2008|      2|
1|    14|      null|    null|           12706|        1|             12705|
0|             -1|        -1|          12702|       13|               -1|
0|             1|        null|           0|       null|                0|
false|
|  335987|      1|      4|         1|           1|           2|
2| No Extras|          0|         0|    0|      0|   0|      0|      0|
0|Not Applicable|  null|  null|   null|null|        null|   null|
null|      null|            null|         null|      null|  2008|      2|
1|    14|      null|    null|           12706|        1|             12705|
0|             -1|        -1|          12702|       13|               -1|
0|             1|        null|           0|       null|                0|
false|
|  335987|      1|      5|         1|           1|           2|
2| No Extras|          0|         0|    0|      0|   0|      0|      0|
0|Not Applicable|  null|  null|   null|null|        null|   null|
null|      null|            null|         null|      null|  2008|      2|
1|    14|      null|    null|           12706|        1|             12705|
```

```
0|              -1|        -1|          12702|       13|                -1|
0|             1|        null|            0|       null|                 0|
false|
|  335987|      1|      6|         1|           1|           2|
2| No Extras|          0|         0|    0|      0|   0|      0|      0|
0|Not Applicable|  null|  null|   null|null|        null|   null|
null|      null|            null|         null|      null|  2008|      2|
1|    14|      null|    null|           12706|        1|               12705|
0|              -1|        -1|          12702|       13|                -1|
0|             1|        null|            0|       null|                 0|
false|
+--------+-------+-------+----------+-----------+-----------+-----------------
-------+----------+-----------+----------+-----+-------+----+-------+-------+---
----------+-------------+------+------+-------+----+------------+-------+------
-----------+----------+---------------+-------------+----------+------+-------+
-----------+------+----------+--------+----------------+---------+--------------
-----+-------------+---------------+----------+--------------+---------+------
-----------+-------------+-------------+-----------+------------+----------
-+------------------+-----------+
only showing top 5 rows
```

[0]:
```python
from pyspark.sql.functions import year, month, dayofmonth, when

# Extracting year, month, and day from the match date for more detailed␣
 ↪time-based analysis
match_df = match_df.withColumn("year", year("match_date"))
match_df = match_df.withColumn("month", month("match_date"))
match_df = match_df.withColumn("day", dayofmonth("match_date"))

# High margin win: categorizing win margins into 'high', 'medium', and 'low'
match_df = match_df.withColumn(
    "win_margin_category",
    when(col("win_margin") >= 100, "High")
    .when((col("win_margin") >= 50) & (col("win_margin") < 100), "Medium")
    .otherwise("Low")
)

# Analyze the impact of the toss: who wins the toss and the match
match_df = match_df.withColumn(
    "toss_match_winner",
    when(col("toss_winner") == col("match_winner"), "Yes").otherwise("No")
)

# Show the enhanced match DataFrame
match_df.show(2)
```

```
+--------+--------+-------------------+--------------------+----------+--------
```

```
---+-------------------+---------+-----------+-------------------+---------
---------+---------+--------+-----------+-----------+---------+---------+---
-+-----+----+-------------------+----------------+
|match_sk|match_id|               team1|
team2|match_date|season_year|          venue_name| city_name|country_name|
toss_winner|        match_winner|toss_name|win_type|outcome_type|
manofmach|win_margin|country_id|year|month|
day|win_margin_category|toss_match_winner|
+--------+--------+-------------------+-------------------+---------+--------
---+-------------------+---------+-----------+-------------------+---------
---------+---------+--------+-----------+-----------+---------+---------+---
-+-----+----+-------------------+----------------+
|       0|  335987|Royal Challengers…|Kolkata Knight Ri…|      null|
2008|M Chinnaswamy Sta…| Bangalore|       India|Royal Challengers…|Kolkata
Knight Ri…|    field|    runs|      Result|BB McCullum|       140|
1|null| null|null|               High|               No|
|       1|  335988|     Kings XI Punjab| Chennai Super Kings|      null|
2008|Punjab Cricket As…|Chandigarh|       India| Chennai Super Kings| Chennai
Super Kings|      bat|    runs|      Result| MEK Hussey|        33|
1|null| null|null|                Low|              Yes|
+--------+--------+-------------------+-------------------+---------+--------
---+-------------------+---------+-----------+-------------------+---------
---------+---------+--------+-----------+-----------+---------+---------+---
-+-----+----+-------------------+----------------+
only showing top 2 rows
```

[0]:
```python
from pyspark.sql.functions import lower, regexp_replace

# Normalize and clean player names
player_df = player_df.withColumn("player_name",␣
 ↪lower(regexp_replace("player_name", "[^a-zA-Z0-9 ]", "")))

# Handle missing values in 'batting_hand' and 'bowling_skill' with a default␣
 ↪'unknown'
player_df = player_df.na.fill({"batting_hand": "unknown", "bowling_skill":␣
 ↪"unknown"})

# Categorizing players based on batting hand
player_df = player_df.withColumn(
    "batting_style",
    when(col("batting_hand").contains("left"), "Left-Handed").
 ↪otherwise("Right-Handed")
)

# Show the modified player DataFrame
player_df.show(2)
```

```
+---------+---------+-----------+----+-------------+---------------+---------
--+-------------+
|player_sk|player_id|player_name| dob|  batting_hand|
bowling_skill|country_name|batting_style|
+---------+---------+-----------+----+-------------+---------------+---------
--+-------------+
|        0|        1| sc ganguly|null| Left-hand bat|Right-arm medium|
India| Right-Handed|
|        1|        2|bb mccullum|null|Right-hand bat|Right-arm medium| New
Zealand| Right-Handed|
+---------+---------+-----------+----+-------------+---------------+---------
--+-------------+
only showing top 2 rows
```

[0]:
```python
from pyspark.sql.functions import col, when, current_date, expr

# Add a 'veteran_status' column based on player age
player_match_df = player_match_df.withColumn(
    "veteran_status",
    when(col("age_as_on_match") >= 35, "Veteran").otherwise("Non-Veteran")
)

# Dynamic column to calculate years since debut
player_match_df = player_match_df.withColumn(
    "years_since_debut",
    (year(current_date()) - col("season_year"))
)

# Show the enriched DataFrame
player_match_df.show()
```

```
+--------------+--------------+--------+---------+--------------+----+-------
-------+-------------------+------------+---------+--------------------+-------
-------------+-----------+----------------+--------------+------------------+--
-----------+-------------+-------------+--------------+-------------+-------
-------+--------------+-----------------+
|player_match_sk|playermatch_key|match_id|player_id|    player_name| dob|
batting_hand|       bowling_skill|country_name|role_desc|         player_team|
opposit_team|season_year|is_manofthematch|age_as_on_match|isplayers_team_won|bat
ting_status|bowling_status|player_captain|opposit_captain|player_keeper|opposit_
keeper|veteran_status|years_since_debut|
+--------------+--------------+--------+---------+--------------+----+-------
-------+-------------------+------------+---------+--------------------+-------
-------------+-----------+----------------+--------------+------------------+--
-----------+-------------+-------------+--------------+-------------+-------
-------+--------------+-----------------+
|             -1|            -1|      -1|       -1|           N/A|null|
```

```
null|                null|        null|     null|                null|
null|       null|            null|           null|              null|
null|          null|          null|           null|         null|          null|
Non-Veteran|             null|
|          12694|           null|  335987|        6|       R Dravid|null|Right-
hand bat|  Right-arm offbreak|       India|  Captain|Royal
Challengers…|Kolkata Knight Ri…|       2008|            null|
35|              null|          null|          null|      R Dravid|     SC
Ganguly|   MV Boucher|       WP Saha|       Veteran|               16|
|          12695|           null|  335987|        7|       W Jaffer|null|Right-
hand bat|  Right-arm offbreak|       India|   Player|Royal
Challengers…|Kolkata Knight Ri…|       2008|            null|
30|              null|          null|          null|      R Dravid|     SC
Ganguly|   MV Boucher|       WP Saha|   Non-Veteran|               16|
|          12696|           null|  335987|        8|        V Kohli|null|Right-
hand bat|    Right-arm medium|       India|   Player|Royal
Challengers…|Kolkata Knight Ri…|       2008|            null|
20|              null|          null|          null|      R Dravid|     SC
Ganguly|   MV Boucher|       WP Saha|   Non-Veteran|               16|
|          12697|           null|  335987|        9|      JH Kallis|null|Right-
hand bat|Right-arm fast-me…|South Africa|   Player|Royal
Challengers…|Kolkata Knight Ri…|       2008|            null|
33|              null|          null|          null|      R Dravid|     SC
Ganguly|   MV Boucher|       WP Saha|   Non-Veteran|               16|
|          12698|           null|  335987|       10|       CL White|null|Right-
hand bat|     Legbreak googly|   Australia|   Player|Royal
Challengers…|Kolkata Knight Ri…|       2008|            null|
25|              null|          null|          null|      R Dravid|     SC
Ganguly|   MV Boucher|       WP Saha|   Non-Veteran|               16|
|          12699|           null|  335987|       11|     MV Boucher|null|Right-
hand bat|    Right-arm medium|South Africa|   Keeper|Royal
Challengers…|Kolkata Knight Ri…|       2008|            null|
32|              null|          null|          null|      R Dravid|     SC
Ganguly|   MV Boucher|       WP Saha|   Non-Veteran|               16|
|          12700|           null|  335987|       12|        B Akhil|null|Right-
hand bat|Right-arm medium-…|       India|   Player|Royal
Challengers…|Kolkata Knight Ri…|       2008|            null|
31|              null|          null|          null|      R Dravid|     SC
Ganguly|   MV Boucher|       WP Saha|   Non-Veteran|               16|
|          12701|           null|  335987|       13|      AA Noffke|null|Right-
hand bat|Right-arm fast-me…|   Australia|   Player|Royal
Challengers…|Kolkata Knight Ri…|       2008|            null|
31|              null|          null|          null|      R Dravid|     SC
Ganguly|   MV Boucher|       WP Saha|   Non-Veteran|               16|
|          12702|           null|  335987|       14|        P Kumar|null|Right-
hand bat|    Right-arm medium|       India|   Player|Royal
Challengers…|Kolkata Knight Ri…|       2008|            null|
22|              null|          null|          null|      R Dravid|     SC
```

```
Ganguly|   MV Boucher|       WP Saha|   Non-Veteran|               16|
|          12703|           null|  335987|       15|         Z Khan|null|Right-
hand bat|Left-arm fast-medium|       India|   Player|Royal
Challengers…|Kolkata Knight Ri…|       2008|            null|
30|              null|          null|          null|      R Dravid|     SC
Ganguly|   MV Boucher|       WP Saha|   Non-Veteran|               16|
|          12704|           null|  335987|       16|       SB Joshi|null| Left-
hand bat|Slow left-arm ort…|       India|   Player|Royal
Challengers…|Kolkata Knight Ri…|       2008|            null|
38|              null|          null|          null|      R Dravid|     SC
Ganguly|   MV Boucher|       WP Saha|       Veteran|               16|
|          12705|           null|  335987|        1|     SC Ganguly|null| Left-
hand bat|    Right-arm medium|       India|  Captain|Kolkata Knight Ri…|Royal
Challengers…|       2008|            null|             36|              null|
null|          null|    SC Ganguly|       R Dravid|      WP Saha|    MV Boucher|
Veteran|               16|
|          12706|           null|  335987|        2|    BB McCullum|null|Right-
hand bat|    Right-arm medium| New Zealand|   Player|Kolkata Knight Ri…|Royal
Challengers…|       2008|            null|             27|              null|
null|          null|    SC Ganguly|       R Dravid|      WP Saha|    MV Boucher|
Non-Veteran|               16|
|          12707|           null|  335987|        3|     RT Ponting|null|Right-
hand bat|    Right-arm medium|   Australia|   Player|Kolkata Knight Ri…|Royal
Challengers…|       2008|            null|             34|              null|
null|          null|    SC Ganguly|       R Dravid|      WP Saha|    MV Boucher|
Non-Veteran|               16|
|          12708|           null|  335987|        4|      DJ Hussey|null|Right-
hand bat|  Right-arm offbreak|   Australia|   Player|Kolkata Knight Ri…|Royal
Challengers…|       2008|            null|             31|              null|
null|          null|    SC Ganguly|       R Dravid|      WP Saha|    MV Boucher|
Non-Veteran|               16|
|          12709|           null|  335987|        5|Mohammad Hafeez|null|Right-
hand bat|  Right-arm offbreak|    Pakistan|   Player|Kolkata Knight Ri…|Royal
Challengers…|       2008|            null|             28|              null|
null|          null|    SC Ganguly|       R Dravid|      WP Saha|    MV Boucher|
Non-Veteran|               16|
|          12710|           null|  335987|       62|        WP Saha|null|Right-
hand bat|                N/A|       India|   Keeper|Kolkata Knight Ri…|Royal
Challengers…|       2008|            null|             24|              null|
null|          null|    SC Ganguly|       R Dravid|      WP Saha|    MV Boucher|
Non-Veteran|               16|
|          12711|           null|  335987|       63|      LR Shukla|null|Right-
hand bat|    Right-arm medium|       India|   Player|Kolkata Knight Ri…|Royal
Challengers…|       2008|            null|             27|              null|
null|          null|    SC Ganguly|       R Dravid|      WP Saha|    MV Boucher|
Non-Veteran|               16|
|          12712|           null|  335987|       82|     AB Agarkar|null|Right-
hand bat|      Right-arm fast|       India|   Player|Kolkata Knight Ri…|Royal
```

```
Challengers…|       2008|            null|             31|              null|
null|          null|    SC Ganguly|       R Dravid|      WP Saha|    MV Boucher|
Non-Veteran|               16|
+--------------+---------------+--------+---------+--------------+----+-------
-------+--------------------+------------+---------+--------------------+-------
-------------+-----------+----------------+--------------+------------------+--
------------+-------------+--------------+--------------+-------------+-------
-------+-------------+----------------+
only showing top 20 rows
```

[0]:
```
ball_by_ball_df.createOrReplaceTempView("ball_by_ball")
match_df.createOrReplaceTempView("match")
player_df.createOrReplaceTempView("player")
player_match_df.createOrReplaceTempView("player_match")
team_df.createOrReplaceTempView("team")
```

[0]:
```
ball_by_ball_df.columns
```

```
Out[24]: ['match_id',
 'over_id',
 'ball_id',
 'innings_no',
 'team_batting',
 'team_bowling',
 'striker_batting_position',
 'extra_type',
 'runs_scored',
 'extra_runs',
 'wides',
 'legbyes',
 'byes',
 'noballs',
 'penalty',
 'bowler_extras',
 'out_type',
 'caught',
 'bowled',
 'run_out',
 'lbw',
 'retired_hurt',
 'stumped',
 'caught_and_bowled',
 'hit_wicket',
 'obstructingfeild',
 'bowler_wicket',
 'match_date',
 'season',
```

```
 'striker',
 'non_striker',
 'bowler',
 'player_out',
 'fielders',
 'striker_match_sk',
 'strikersk',
 'nonstriker_match_sk',
 'nonstriker_sk',
 'fielder_match_sk',
 'fielder_sk',
 'bowler_match_sk',
 'bowler_sk',
 'playerout_match_sk',
 'battingteam_sk',
 'bowlingteam_sk',
 'keeper_catch',
 'player_out_sk',
 'matchdatesk',
 'running_total_runs',
 'high_impact']
```

[0]:
```
top_scoring_batsmen_per_season = spark.sql("""
SELECT
p.player_name,
m.season_year,
SUM(b.runs_scored) AS total_runs
FROM ball_by_ball b
JOIN match m ON b.match_id = m.match_id
JOIN player_match pm ON m.match_id = pm.match_id AND b.striker = pm.player_id
JOIN player p ON p.player_id = pm.player_id
GROUP BY p.player_name, m.season_year
ORDER BY m.season_year, total_runs DESC
""")
```

[0]:
```
top_scoring_batsmen_per_season.show(30)
```

```
+--------------+-----------+----------+
|    player_name|season_year|total_runs|
+--------------+-----------+----------+
|       se marsh|       2008|       614|
|      g gambhir|       2008|       532|
|  st jayasuriya|       2008|       508|
|      sr watson|       2008|       463|
|       gc smith|       2008|       437|
|   ac gilchrist|       2008|       431|
|      yk pathan|       2008|       430|
```

```
|       sk raina|       2008|       420|
|       ms dhoni|       2008|       414|
|       v sehwag|       2008|       399|
|      rg sharma|       2008|       399|
|       r dravid|       2008|       370|
|     sc ganguly|       2008|       349|
|       s dhawan|       2008|       340|
|  kc sangakkara|       2008|       319|
|      dj hussey|       2008|       318|
|     rv uthappa|       2008|       316|
|    sa asnodkar|       2008|       311|
|   yuvraj singh|       2008|       299|
|       pa patel|       2008|       297|
|y venugopal rao|       2008|       283|
|      ja morkel|       2008|       235|
|     mv boucher|       2008|       225|
|       jr hopes|       2008|       221|
|       am nayar|       2008|       205|
|      jh kallis|       2008|       199|
|    salman butt|       2008|       192|
|     sp fleming|       2008|       192|
|    s badrinath|       2008|       192|
|   sr tendulkar|       2008|       188|
+---------------+-----------+----------+
only showing top 30 rows
```

[0]:
```
economical_bowlers_powerplay = spark.sql("""
SELECT
p.player_name,
AVG(b.runs_scored) AS avg_runs_per_ball,
COUNT(b.bowler_wicket) AS total_wickets
FROM ball_by_ball b
JOIN player_match pm ON b.match_id = pm.match_id AND b.bowler = pm.player_id
JOIN player p ON pm.player_id = p.player_id
WHERE b.over_id <= 6
GROUP BY p.player_name
HAVING COUNT(*) >= 1
ORDER BY avg_runs_per_ball, total_wickets DESC
""")
economical_bowlers_powerplay.show()
```

```
+---------------+------------------+-------------+
|    player_name| avg_runs_per_ball|total_wickets|
+---------------+------------------+-------------+
|     sm harwood|0.3333333333333333|            0|
|     ankit soni|               0.5|            0|
|      gr napier|               0.5|            0|
```

```
|       aj finch|               0.5|            0|
|        a zampa|               0.5|            0|
|     avesh khan|               0.5|            0|
|       nb singh|0.5833333333333334|            0|
|     ag murtaza|0.6538461538461539|            0|
|      sb bangar|0.6666666666666666|            0|
|     d du preez|0.6666666666666666|            0|
|        s gopal|0.6666666666666666|            0|
|     fh edwards|0.6923076923076923|            0|
|       a kumble|0.7685185185185185|            0|
|j syed mohammad|0.7777777777777778|            0|
|   kp pietersen|0.7777777777777778|            0|
|       umar gul|0.7777777777777778|            0|
|  la carseldine|0.8333333333333334|            0|
|    rj peterson|0.8333333333333334|            0|
|      ss mundhe|0.8333333333333334|            0|
|       tl suman|0.8333333333333334|            0|
+---------------+------------------+-------------+
only showing top 20 rows
```

[0]:
```
toss_impact_individual_matches = spark.sql("""
SELECT m.match_id, m.toss_winner, m.toss_name, m.match_winner,
       CASE WHEN m.toss_winner = m.match_winner THEN 'Won' ELSE 'Lost' END AS␣
 ↪match_outcome
FROM match m
WHERE m.toss_name IS NOT NULL
ORDER BY m.match_id
""")
toss_impact_individual_matches.show()
```

```
+--------+--------------------+---------+--------------------+-------------+
|match_id|         toss_winner|toss_name|        match_winner|match_outcome|
+--------+--------------------+---------+--------------------+-------------+
|  335987|Royal Challengers…|    field|Kolkata Knight Ri…|         Lost|
|  335988| Chennai Super Kings|      bat| Chennai Super Kings|          Won|
|  335989|    Rajasthan Royals|      bat|    Delhi Daredevils|         Lost|
|  335990|      Mumbai Indians|      bat|Royal Challengers…|         Lost|
|  335991|     Deccan Chargers|      bat|Kolkata Knight Ri…|         Lost|
|  335992|     Kings XI Punjab|      bat|    Rajasthan Royals|         Lost|
|  335993|     Deccan Chargers|      bat|    Delhi Daredevils|         Lost|
|  335994|      Mumbai Indians|    field| Chennai Super Kings|         Lost|
|  335995|    Rajasthan Royals|    field|    Rajasthan Royals|          Won|
|  335996|      Mumbai Indians|    field|     Kings XI Punjab|         Lost|
|  335997|    Rajasthan Royals|    field|    Rajasthan Royals|          Won|
|  335998|Kolkata Knight Ri…|      bat| Chennai Super Kings|         Lost|
|  335999|     Deccan Chargers|    field|     Deccan Chargers|          Won|
|  336000|    Delhi Daredevils|      bat|     Kings XI Punjab|         Lost|
```

```
|  336001| Chennai Super Kings|      bat| Chennai Super Kings|          Won|
|  336002|Kolkata Knight Ri…|      bat|      Mumbai Indians|         Lost|
|  336003|Royal Challengers…|    field|    Delhi Daredevils|         Lost|
|  336004|     Kings XI Punjab|    field|     Kings XI Punjab|          Won|
|  336005|    Rajasthan Royals|      bat|    Rajasthan Royals|          Won|
|  336006| Chennai Super Kings|      bat|    Delhi Daredevils|         Lost|
+--------+--------------------+---------+--------------------+-------------+
only showing top 20 rows
```

[0]:
```
average_runs_in_wins = spark.sql("""
SELECT p.player_name, AVG(b.runs_scored) AS avg_runs_in_wins, COUNT(*) AS␣
 ↪innings_played
FROM ball_by_ball b
JOIN player_match pm ON b.match_id = pm.match_id AND b.striker = pm.player_id
JOIN player p ON pm.player_id = p.player_id
JOIN match m ON pm.match_id = m.match_id
WHERE m.match_winner = pm.player_team
GROUP BY p.player_name
ORDER BY avg_runs_in_wins ASC
""")
average_runs_in_wins.show()
```

```
+--------------+-------------------+--------------+
|   player_name|   avg_runs_in_wins|innings_played|
+--------------+-------------------+--------------+
|       a nehra|                0.0|             2|
|    kp appanna|                0.0|             1|
|     jj bumrah|                0.0|             2|
|      i sharma|                0.0|             1|
|      ts mills|                0.0|             3|
|      j theron|                0.0|             1|
|      vr aaron|                0.0|             5|
|     sn thakur|                0.0|             2|
| anirudh singh|                0.0|             1|
|    t thushara|                0.2|             5|
|     sa abbott|               0.25|             4|
| yashpal singh| 0.3157894736842105|            19|
|   s sreesanth| 0.3333333333333333|             3|
|   kc cariappa| 0.3333333333333333|             3|
|    jd unadkat|                0.4|             5|
|    sm harwood|0.42857142857142855|             7|
|      r shukla|                0.5|             2|
|       jm kemp|                0.5|             8|
|       b kumar|                0.5|            10|
|sandeep sharma|                0.5|             6|
+--------------+-------------------+--------------+
only showing top 20 rows
```

[0]:
```
import plotly.express as px
```

[0]:
```
# Convert Spark DataFrame to Pandas DataFrame
economical_bowlers_pd = economical_bowlers_powerplay.toPandas()

# Limiting to top 10 most economical bowlers
top_economical_bowlers = economical_bowlers_pd.nsmallest(10,␣
 ↪'avg_runs_per_ball')

# Create a bar plot using Plotly
fig = px.bar(
    top_economical_bowlers,
    x='player_name',
    y='avg_runs_per_ball',
    title='Most Economical Bowlers in Powerplay Overs (Top 10)',
    labels={'player_name': 'Bowler Name', 'avg_runs_per_ball': 'Average Runs␣
 ↪per Ball'},
    color='avg_runs_per_ball',
    color_continuous_scale='Viridis'
)

# Update layout for better clarity
fig.update_layout(
    xaxis_title='Bowler Name',
    yaxis_title='Average Runs per Ball',
    xaxis_tickangle=-45,
    template='plotly_white',
    width=800,
    height=500
)

# Show the figure
fig.show()
```

[0]:
```
toss_impact_pd = toss_impact_individual_matches.toPandas()

fig = px.bar(
    toss_impact_pd,
    x='toss_winner',
    color='match_outcome',
    title='Impact of Winning Toss on Match Outcomes',
    labels={'toss_winner': 'Toss Winner', 'match_outcome': 'Match Outcome'},
    barmode='group',
    height=600,
    width=800
```

```
)

fig.update_layout(
    xaxis_title='Toss Winner',
    yaxis_title='Number of Matches',
    xaxis_tickangle=-45,
    template='plotly_white'
)

fig.show()
```

[0]:
```
average_runs_pd = average_runs_in_wins.toPandas()
top_scorers = average_runs_pd.nlargest(10, 'avg_runs_in_wins')

fig = px.bar(
    top_scorers,
    x='player_name',
    y='avg_runs_in_wins',
    title='Average Runs Scored by Batsmen in Winning Matches (Top 10 Scorers)',
    labels={'player_name': 'Player Name', 'avg_runs_in_wins': 'Average Runs in␣
 ↪Wins'},
    color='avg_runs_in_wins',
    color_continuous_scale='Viridis',
    height=600,
    width=800
)

fig.update_layout(
    xaxis_title='Player Name',
    yaxis_title='Average Runs in Wins',
    xaxis_tickangle=-45,
    template='plotly_white'
)

fig.show()
```

[0]:
```
scores_by_venue = spark.sql("""
SELECT venue_name, AVG(total_runs) AS average_score, MAX(total_runs) AS␣
 ↪highest_score
FROM (
    SELECT ball_by_ball.match_id, match.venue_name, SUM(runs_scored) AS␣
 ↪total_runs
    FROM ball_by_ball
    JOIN match ON ball_by_ball.match_id = match.match_id
    GROUP BY ball_by_ball.match_id, match.venue_name
)
GROUP BY venue_name
```

```
ORDER BY average_score DESC
""")
```

[0]:
```
scores_by_venue_pd = scores_by_venue.toPandas()

fig = px.bar(
    scores_by_venue_pd,
    x='average_score',
    y='venue_name',
    orientation='h',
    title='Distribution of Scores by Venue',
    labels={'average_score': 'Average Score', 'venue_name': 'Venue'},
    color='average_score',
    color_continuous_scale='Viridis',
    height=600,
    width=900
)

fig.update_layout(
    xaxis_title='Average Score',
    yaxis_title='Venue',
    template='plotly_white'
)

fig.show()
```

[0]:
```
# Execute SQL Query
dismissal_types = spark.sql("""
SELECT out_type, COUNT(*) AS frequency
FROM ball_by_ball
WHERE out_type IS NOT NULL
GROUP BY out_type
ORDER BY frequency DESC
""")
```

[0]:
```
dismissal_types_pd = dismissal_types.toPandas()

fig = px.bar(
    dismissal_types_pd,
    x='frequency',
    y='out_type',
    orientation='h',
    title='Most Frequent Dismissal Types',
    labels={'frequency': 'Frequency', 'out_type': 'Dismissal Type'},
    color='frequency',
    color_continuous_scale='Viridis',
    height=600,
```

```
    width=900
)

fig.update_layout(
    xaxis_title='Frequency',
    yaxis_title='Dismissal Type',
    template='plotly_white'
)

fig.show()
```

[0]:
```
# Execute SQL Query
team_toss_win_performance = spark.sql("""
SELECT team1, COUNT(*) AS matches_played, SUM(CASE WHEN toss_winner =␣
 ↪match_winner THEN 1 ELSE 0 END) AS wins_after_toss
FROM match
WHERE toss_winner = team1
GROUP BY team1
ORDER BY wins_after_toss DESC
""")
```

[0]:
```
team_toss_win_pd = team_toss_win_performance.toPandas()

fig = px.bar(
    team_toss_win_pd,
    x='wins_after_toss',
    y='team1',
    orientation='h',
    title='Team Performance After Winning Toss',
    labels={'wins_after_toss': 'Wins After Winning Toss', 'team1': 'Team'},
    color='wins_after_toss',
    color_continuous_scale='Viridis',
    height=600,
    width=900
)

fig.update_layout(
    xaxis_title='Wins After Winning Toss',
    yaxis_title='Team',
    template='plotly_white'
)

fig.show()
```