

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.moun

```
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
```

```
!wget -q https://mirrors.estointernet.in/apache/spark/spark-3.1.2/spark-3.1.2-bin-hadoop3.2.tgz
```

```
mv spark-3.1.2-bin-hadoop3.2.tgz spark3.tgz
```

```
!tar xf spark3.tgz
!pip install -q findspark
```

```
!pip install -q findspark
```

```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.1.2-bin-hadoop3.2"
```

```
import findspark
findspark.init()
```

```
findspark.find()
```

```
'/content/spark-3.1.2-bin-hadoop3.2'
```

we can import SparkSession from pyspark.sql and create a SparkSession, which is the entry point to Spark.

```
from pyspark.sql import SparkSession
spark = SparkSession.builder\
    .master("local")\
    .appName("Colab")\
    .config('spark.ui.port', '4050')\
    .getOrCreate()
```

Finally, print the SparkSession variable.

```
spark
```

SparkSession - in-memory

SparkContext

[Spark UI](#)

Version

v3.1.2

Master

local

AppName

Colab

```
df = spark.read.csv("/content/drive/MyDrive/deliveries.csv", header=True, inferSchema=True)
```

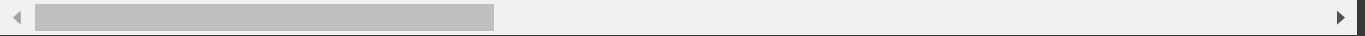
```
df.printSchema()
```

```
root
|-- match_id: integer (nullable = true)
|-- inning: integer (nullable = true)
|-- batting_team: string (nullable = true)
|-- bowling_team: string (nullable = true)
|-- over: integer (nullable = true)
|-- ball: integer (nullable = true)
|-- batsman: string (nullable = true)
|-- non_striker: string (nullable = true)
|-- bowler: string (nullable = true)
|-- is_super_over: integer (nullable = true)
|-- wide_runs: integer (nullable = true)
|-- bye_runs: integer (nullable = true)
|-- legbye_runs: integer (nullable = true)
|-- noball_runs: integer (nullable = true)
|-- penalty_runs: integer (nullable = true)
|-- batsman_runs: integer (nullable = true)
|-- extra_runs: integer (nullable = true)
|-- total_runs: integer (nullable = true)
|-- player_dismissed: string (nullable = true)
|-- dismissal_kind: string (nullable = true)
|-- fielder: string (nullable = true)
```

```
df.show(10)
```

match_id	inning	batting_team	bowling_team	over	ball	batsman	non_str
1	1	Kolkata Knight Ri...	Royal Challengers...	1	1	SC Ganguly	BB McCu
1	1	Kolkata Knight Ri...	Royal Challengers...	1	2	BB McCullum	SC Gar
1	1	Kolkata Knight Ri...	Royal Challengers...	1	3	BB McCullum	SC Gar
1	1	Kolkata Knight Ri...	Royal Challengers...	1	4	BB McCullum	SC Gar
1	1	Kolkata Knight Ri...	Royal Challengers...	1	5	BB McCullum	SC Gar
1	1	Kolkata Knight Ri...	Royal Challengers...	1	6	BB McCullum	SC Gar
1	1	Kolkata Knight Ri...	Royal Challengers...	1	7	BB McCullum	SC Gar
1	1	Kolkata Knight Ri...	Royal Challengers...	2	1	BB McCullum	SC Gar
1	1	Kolkata Knight Ri...	Royal Challengers...	2	2	BB McCullum	SC Gar
1	1	Kolkata Knight Ri...	Royal Challengers...	2	3	BB McCullum	SC Gar

only showing top 10 rows



```
df.count()
```

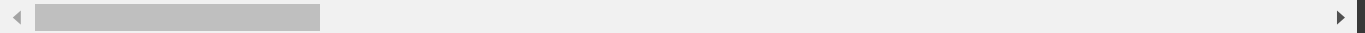
```
136598
```

```
df.select()
```

```
DataFrame[]
```

```
df.describe().show()
```

```
+-----+-----+-----+-----+-----+
|summary|      match_id|      inning|      batting_team|      bowling_team|
+-----+-----+-----+-----+-----+
|  count|      136598|      136598|      136598|      136598|
|   mean|  288.564678838636|  1.4827376681942634|      null|      null|
| stddev| 165.92986501567495|  0.5015754838499102|      null|      null|
|    min|           1|           1|Chennai Super Kings|Chennai Super Kings|
|    max|          577|           4|Sunrisers Hyderabad|Sunrisers Hyderabad|
+-----+-----+-----+-----+-----+
```



Implementation: Try finding the count max and min of all the columns in your dataframe.

```
# Extra runs given by all teams
runs = df.groupby(['bowling_team']).sum("extra_runs")
runs.show()
```

```
+-----+-----+
|      bowling_team|sum(extra_runs)|
+-----+-----+
| Sunrisers Hyderabad|      403|
| Chennai Super Kings|     1002|
| Deccan Chargers|      659|
| Kochi Tuskers Kerala|      110|
| Rajasthan Royals|     1058|
| Gujarat Lions|       98|
| Royal Challengers...|     1205|
| Kolkata Knight Ri...|     1073|
| Rising Pune Super...|      108|
| Kings XI Punjab|     1139|
| Pune Warriors|      335|
| Delhi Daredevils|     1069|
| Mumbai Indians|     1260|
+-----+-----+
```

```
# Most wide runs
df.groupby(['bowling_team']).sum("wide_runs").sort(desc("sum(wide_runs)")).show()
```

```
+-----+
|      bowling_team|sum(wide_runs)|
+-----+
|      Mumbai Indians|      715|
|Royal Challengers...|      667|
|      Kings XI Punjab|      616|
|Kolkata Knight Ri...|      586|
|      Rajasthan Royals|      586|
|      Delhi Daredevils|      551|
|Chennai Super Kings|      526|
|      Deccan Chargers|      328|
|Sunrisers Hyderabad|      223|
|      Pune Warriors|      174|
|Rising Pune Super...|       77|
|Kochi Tuskers Kerala|       56|
|      Gujarat Lions|       56|
+-----+
```

```
# No ball runs
df.groupby(['bowling_team']).sum("noball_runs").sort(desc("sum(noball_runs)")).show()
```

```
+-----+
|      bowling_team|sum(noball_runs)|
+-----+
|      Delhi Daredevils|      84|
|      Kings XI Punjab|      81|
|      Mumbai Indians|      78|
|Royal Challengers...|      74|
|      Rajasthan Royals|      65|
|Kolkata Knight Ri...|      61|
|Chennai Super Kings|      56|
|      Deccan Chargers|      49|
|      Pune Warriors|      26|
|Sunrisers Hyderabad|      19|
|Kochi Tuskers Kerala|       8|
|      Gujarat Lions|       7|
|Rising Pune Super...|       4|
+-----+
```

```
# Most balls bowled by a bowler
df.groupby(['bowler']).count().sort(desc("count")).show()
```

```
+-----+
|      bowler|count|
+-----+
|Harbhajan Singh| 2742|
|      P Kumar| 2529|
|      PP Chawla| 2472|
|      A Mishra| 2466|
|      SL Malinga| 2407|
|      R Ashwin| 2359|
+-----+
```

```

|      DW Steyn| 2159|
| R Vinay Kumar| 2141|
|      DJ Bravo| 2110|
|      IK Pathan| 2101|
|      Z Khan| 2030|
|      PP Ojha| 1945|
|      RP Singh| 1874|
|      A Nehra| 1842|
|      JA Morkel| 1807|
|      JH Kallis| 1799|
|      SR Watson| 1796|
|      RA Jadeja| 1733|
|      B Kumar| 1730|
|      UT Yadav| 1729|
+-----+
only showing top 20 rows

```

```
# Average runs scored by every team in powerplay
```

```
df[df['over']<6].groupby(['match_id','batting_team']).sum("total_runs").groupby('batting_team').mean().show()
```

```

+-----+-----+-----+
|      batting_team|      avg(match_id)|      avg(sum(total_runs))|
+-----+-----+-----+
| Sunrisers Hyderabad| 457.30645161290323|      37.435483870967744|
| Chennai Super Kings| 262.9465648854962|      37.20610687022901|
| Deccan Chargers| 152.06666666666666|      37.586666666666666|
| Kochi Tuskers Kerala| 208.57142857142858|      38.357142857142854|
| Rajasthan Royals| 253.42735042735043|      35.136752136752136|
| Gujarat Lions| 548.0|      40.0625|
| Royal Challengers...| 289.8201438848921|      36.35971223021583|
| Kolkata Knight Ri...| 292.65909090909093|      37.053030303030305|
| Rising Pune Super...| 544.4285714285714|      35.07142857142857|
| Kings XI Punjab| 289.2313432835821|      37.38059701492537|
| Pune Warriors| 287.66666666666667|      34.822222222222222|
| Delhi Daredevils| 285.56390977443607|      37.75187969924812|
| Mumbai Indians| 292.95714285714286|      34.792857142857144|
+-----+-----+-----+

```

