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

```
Mounted at /content/gdrive
```

```
!apt-get install openjdk-11-jdk-headless -qq > /dev/null
!wget -q https://downloads.apache.org/spark/spark-3.5.1/spark-3.5.1-bin-hadoop3.tgz
!tar xf spark-3.5.1-bin-hadoop3.tgz
!pip install -q findspark
```

```
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-11-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.5.1-bin-hadoop3"
```

```
import findspark
findspark.init()
from pyspark.sql import SparkSession
spark = SparkSession.builder.master("local[*]").getOrCreate()
sc = spark.sparkContext
```

```
spark
```

SparkSession - in-memory

SparkContext

Spark UI

Version
v3.5.1
Master
local[*]
AppName
pyspark-shell

```
sc
```

SparkContext

Spark UI

Version
v3.5.1
Master
local[*]
AppName
pyspark-shell

```
df=spark.read.csv('gdrive/MyDrive/sample_data.csv',inferSchema=True,header=True)
```

```
df.columns
```

```
['ratings', 'age', 'experience', 'family', 'mobile']
```

```
len(df.columns)
```

```
5
```

```
df.count()
```

```
33
```

```
print((df.count(),len(df.columns)))
```

```
(33, 5)
```

```
df.printSchema()
```

```
root
 |-- ratings: integer (nullable = true)
 |-- age: integer (nullable = true)
 |-- experience: double (nullable = true)
 |-- family: integer (nullable = true)
 |-- mobile: string (nullable = true)
```

```
df.show(5)
```

```
+-------+---+----------+------+-------+
|ratings|age|experience|family| mobile|
+-------+---+----------+------+-------+
|      3| 32|       9.0|     3|   Vivo|
|      3| 27|      13.0|     3|  Apple|
|      4| 22|       2.5|     0|Samsung|
|      4| 37|      16.5|     4|  Apple|
|      5| 27|       9.0|     1|     MI|
+-------+---+----------+------+-------+
only showing top 5 rows
```

```
df.select('age','mobile').show(5)
```

```
+---+-------+
|age| mobile|
+---+-------+
| 32|   Vivo|
| 27|  Apple|
| 22|Samsung|
| 37|  Apple|
| 27|     MI|
+---+-------+
only showing top 5 rows
```

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

```
+-------+------------------+------------------+------------------+------------------+------+
|summary|           ratings|               age|        experience|            family|mobile|
+-------+------------------+------------------+------------------+------------------+------+
|  count|                33|                33|                33|                33|    33|
|   mean|3.5757575757575757|30.484848484848484|10.303030303030303|1.8181818181818181|  NULL|
| stddev|1.1188806636071336|  6.18527087180309| 6.770731351213326|1.8448330794164254|  NULL|
|    min|                 1|                22|               2.5|                 0| Apple|
|    max|                 5|                42|              23.0|                 5|  Vivo|
+-------+------------------+------------------+------------------+------------------+------+
```

```
from pyspark.sql.types import StringType,DoubleType,IntegerType
```

```
df.withColumn("age_after_10_yrs",(df["age"]+10)).show(10,False)
```

```
+-------+---+----------+------+-------+----------------+
|ratings|age|experience|family|mobile |age_after_10_yrs|
+-------+---+----------+------+-------+----------------+
|3      |32 |9.0       |3     |Vivo   |42              |
|3      |27 |13.0      |3     |Apple  |37              |
|4      |22 |2.5       |0     |Samsung|32              |
|4      |37 |16.5      |4     |Apple  |47              |
|5      |27 |9.0       |1     |MI     |37              |
|4      |27 |9.0       |0     |Oppo   |37              |
|5      |37 |23.0      |5     |Vivo   |47              |
|5      |37 |23.0      |5     |Samsung|47              |
|3      |22 |2.5       |0     |Apple  |32              |
|3      |27 |6.0       |0     |MI     |37              |
+-------+---+----------+------+-------+----------------+
only showing top 10 rows
```

```
df.withColumn('age_double',df['age'].cast(DoubleType())).show(10,False)
```

```
+-------+---+----------+------+-------+----------+
|ratings|age|experience|family|mobile |age_double|
+-------+---+----------+------+-------+----------+
|3      |32 |9.0       |3     |Vivo   |32.0      |
|3      |27 |13.0      |3     |Apple  |27.0      |
|4      |22 |2.5       |0     |Samsung|22.0      |
|4      |37 |16.5      |4     |Apple  |37.0      |
|5      |27 |9.0       |1     |MI     |27.0      |
|4      |27 |9.0       |0     |Oppo   |27.0      |
|5      |37 |23.0      |5     |Vivo   |37.0      |
|5      |37 |23.0      |5     |Samsung|37.0      |
|3      |22 |2.5       |0     |Apple  |22.0      |
|3      |27 |6.0       |0     |MI     |27.0      |
+-------+---+----------+------+-------+----------+
only showing top 10 rows
```

```
df.withColumn("age_after_10_yrs",(df["age"]+10)).show(10,False)
```

```
+-------+---+----------+------+-------+----------------+
|ratings|age|experience|family|mobile |age_after_10_yrs|
+-------+---+----------+------+-------+----------------+
|3      |32 |9.0       |3     |Vivo   |42              |
|3      |27 |13.0      |3     |Apple  |37              |
|4      |22 |2.5       |0     |Samsung|32              |
|4      |37 |16.5      |4     |Apple  |47              |
|5      |27 |9.0       |1     |MI     |37              |
|4      |27 |9.0       |0     |Oppo   |37              |
|5      |37 |23.0      |5     |Vivo   |47              |
|5      |37 |23.0      |5     |Samsung|47              |
|3      |22 |2.5       |0     |Apple  |32              |
|3      |27 |6.0       |0     |MI     |37              |
+-------+---+----------+------+-------+----------------+
only showing top 10 rows
```

```
df.filter(df['mobile']=='Vivo').show()
```

```
+-------+---+----------+------+------+
|ratings|age|experience|family|mobile|
+-------+---+----------+------+------+
|      3| 32|       9.0|     3|  Vivo|
|      5| 37|      23.0|     5|  Vivo|
|      4| 37|       6.0|     0|  Vivo|
|      5| 37|      13.0|     1|  Vivo|
|      4| 37|       6.0|     0|  Vivo|
+-------+---+----------+------+------+
```

```
df.filter(df['mobile']=='Vivo').select('age','ratings','mobile').show()
```

```
+---+-------+------+
|age|ratings|mobile|
+---+-------+------+
| 32|      3|  Vivo|
| 37|      5|  Vivo|
| 37|      4|  Vivo|
| 37|      5|  Vivo|
| 37|      4|  Vivo|
+---+-------+------+
```

```
df.filter(df['mobile']=='Vivo').filter(df['experience'] >10).show()
```

```
+-------+---+----------+------+------+
|ratings|age|experience|family|mobile|
+-------+---+----------+------+------+
|      5| 37|      23.0|     5|  Vivo|
|      5| 37|      13.0|     1|  Vivo|
+-------+---+----------+------+------+
```

```
df.filter((df['mobile']=='Vivo')&(df['experience'] >10)).show()
```

```
+-------+---+----------+------+------+
|ratings|age|experience|family|mobile|
+-------+---+----------+------+------+
|      5| 37|      23.0|     5|  Vivo|
|      5| 37|      13.0|     1|  Vivo|
+-------+---+----------+------+------+
```

```
df.select('mobile').distinct().show()
```

```
+-------+
| mobile|
+-------+
|     MI|
|   Oppo|
|Samsung|
|   Vivo|
|  Apple|
+-------+
```

```
df.select('mobile').distinct().count()
```

```
5
```

```
df.groupBy('mobile').count().show(5,False)
```

```
+-------+-----+
|mobile |count|
+-------+-----+
|MI     |8    |
|Oppo   |7    |
|Samsung|6    |
|Vivo   |5    |
|Apple  |7    |
+-------+-----+
```

```
df.groupBy('mobile').count().orderBy('count',ascending=False).show(5,False)
```

```
+-------+-----+
|mobile |count|
+-------+-----+
|MI     |8    |
|Oppo   |7    |
|Apple  |7    |
|Samsung|6    |
|Vivo   |5    |
+-------+-----+
```

```
df.groupBy('mobile').mean().show(5,False)
```

```
+-------+------------------+------------------+------------------+------------------+
|mobile |avg(ratings)      |avg(age)          |avg(experience)   |avg(family)       |
+-------+------------------+------------------+------------------+------------------+
|MI     |3.5               |30.125            |10.1875           |1.375             |
|Oppo   |2.857142857142857 |28.428571428571427|10.357142857142858|1.4285714285714286|
|Samsung|4.166666666666667 |28.666666666666668|8.666666666666666 |1.8333333333333333|
|Vivo   |4.2               |36.0              |11.4              |1.8               |
|Apple  |3.4285714285714284|30.571428571428573|11.0              |2.7142857142857144|
+-------+------------------+------------------+------------------+------------------+
```

```
df.groupBy('mobile').sum().show(5,False)
```

```
+-------+------------+--------+---------------+-----------+
|mobile |sum(ratings)|sum(age)|sum(experience)|sum(family)|
+-------+------------+--------+---------------+-----------+
|MI     |28          |241     |81.5           |11         |
|Oppo   |20          |199     |72.5           |10         |
|Samsung|25          |172     |52.0           |11         |
|Vivo   |21          |180     |57.0           |9          |
|Apple  |24          |214     |77.0           |19         |
+-------+------------+--------+---------------+-----------+
```

```
df.groupBy('mobile').max().show(5,False)
```

```
+-------+------------+--------+---------------+-----------+
|mobile |max(ratings)|max(age)|max(experience)|max(family)|
+-------+------------+--------+---------------+-----------+
|MI     |5           |42      |23.0           |5          |
|Oppo   |4           |42      |23.0           |2          |
|Samsung|5           |37      |23.0           |5          |
|Vivo   |5           |37      |23.0           |5          |
|Apple  |4           |37      |16.5           |5          |
+-------+------------+--------+---------------+-----------+
```

```
df.groupBy('mobile').min().show(5,False)
```

```
+-------+------------+--------+---------------+-----------+
|mobile |min(ratings)|min(age)|min(experience)|min(family)|
+-------+------------+--------+---------------+-----------+
|MI     |1           |27      |2.5            |0          |
|Oppo   |2           |22      |6.0            |0          |
|Samsung|2           |22      |2.5            |0          |
|Vivo   |3           |32      |6.0            |0          |
|Apple  |3           |22      |2.5            |0          |
+-------+------------+--------+---------------+-----------+
```

```
df.groupBy('mobile').agg({'experience':'sum'}).show(5,False)
```

```
+-------+---------------+
|mobile |sum(experience)|
+-------+---------------+
|MI     |81.5           |
|Oppo   |72.5           |
|Samsung|52.0           |
```

```
|Vivo   |57.0           |
|Apple  |77.0           |
+-------+---------------+
```

```
from pyspark.sql.functions import udf
```

```
def price_range(brand):
    if brand in ['Samsung','Apple']:
        return 'High Price'
    elif brand =='MI':
        return 'Mid Price'
    else:
        return 'Low Price'
```

```
brand_udf=udf(price_range,StringType())
#apply udf on dataframe
df.withColumn('price_range',brand_udf(df['mobile'])).show(10,False)
```

```
+-------+---+----------+------+-------+-----------+
|ratings|age|experience|family|mobile |price_range|
+-------+---+----------+------+-------+-----------+
|3      |32 |9.0       |3     |Vivo   |Low Price  |
|3      |27 |13.0      |3     |Apple  |High Price |
|4      |22 |2.5       |0     |Samsung|High Price |
|4      |37 |16.5      |4     |Apple  |High Price |
|5      |27 |9.0       |1     |MI     |Mid Price  |
|4      |27 |9.0       |0     |Oppo   |Low Price  |
|5      |37 |23.0      |5     |Vivo   |Low Price  |
|5      |37 |23.0      |5     |Samsung|High Price |
|3      |22 |2.5       |0     |Apple  |High Price |
|3      |27 |6.0       |0     |MI     |Mid Price  |
+-------+---+----------+------+-------+-----------+
only showing top 10 rows
```

```
age_udf = udf(lambda age: "young" if age <= 30 else "senior", StringType())
#apply udf on dataframe
df.withColumn("age_group", age_udf(df.age)).show(10,False)
```

```
+-------+---+----------+------+-------+---------+
|ratings|age|experience|family|mobile |age_group|
+-------+---+----------+------+-------+---------+
|3      |32 |9.0       |3     |Vivo   |senior   |
|3      |27 |13.0      |3     |Apple  |young    |
|4      |22 |2.5       |0     |Samsung|young    |
|4      |37 |16.5      |4     |Apple  |senior   |
|5      |27 |9.0       |1     |MI     |young    |
|4      |27 |9.0       |0     |Oppo   |young    |
|5      |37 |23.0      |5     |Vivo   |senior   |
|5      |37 |23.0      |5     |Samsung|senior   |
|3      |22 |2.5       |0     |Apple  |young    |
|3      |27 |6.0       |0     |MI     |young    |
+-------+---+----------+------+-------+---------+
only showing top 10 rows
```

```
from pyspark.sql.functions import pandas_udf, PandasUDFType
```

```
def remaining_yrs(age):
    yrs_left=100-age

    return yrs_left
```

```
#create udf using python function
length_udf = pandas_udf(remaining_yrs, IntegerType())
#apply pandas udf on dataframe
df.withColumn("yrs_left", length_udf(df['age'])).show(10,False)
```

```
+-------+---+----------+------+-------+--------+
|ratings|age|experience|family|mobile |yrs_left|
+-------+---+----------+------+-------+--------+
|3      |32 |9.0       |3     |Vivo   |68      |
|3      |27 |13.0      |3     |Apple  |73      |
|4      |22 |2.5       |0     |Samsung|78      |
|4      |37 |16.5      |4     |Apple  |63      |
|5      |27 |9.0       |1     |MI     |73      |
|4      |27 |9.0       |0     |Oppo   |73      |
|5      |37 |23.0      |5     |Vivo   |63      |
|5      |37 |23.0      |5     |Samsung|63      |
|3      |22 |2.5       |0     |Apple  |78      |
|3      |27 |6.0       |0     |MI     |73      |
```

```
+-------+---+----------+------+-------+--------+
only showing top 10 rows
```

```
def prod(rating,exp):
    x=rating*exp
    return x
```

```
prod_udf = pandas_udf(prod, DoubleType())
#apply pandas udf on multiple columns of dataframe
df.withColumn("product", prod_udf(df['ratings'],df['experience'])).show(10,False)
```

```
+-------+---+----------+------+-------+-------+
|ratings|age|experience|family|mobile |product|
+-------+---+----------+------+-------+-------+
|3      |32 |9.0       |3     |Vivo   |27.0   |
|3      |27 |13.0      |3     |Apple  |39.0   |
|4      |22 |2.5       |0     |Samsung|10.0   |
|4      |37 |16.5      |4     |Apple  |66.0   |
|5      |27 |9.0       |1     |MI     |45.0   |
|4      |27 |9.0       |0     |Oppo   |36.0   |
|5      |37 |23.0      |5     |Vivo   |115.0  |
|5      |37 |23.0      |5     |Samsung|115.0  |
|3      |22 |2.5       |0     |Apple  |7.5    |
|3      |27 |6.0       |0     |MI     |18.0   |
+-------+---+----------+------+-------+-------+
only showing top 10 rows
```

```
df.count()
```

```
33
```

```
df=df.dropDuplicates()
```

```
df.count()
```

```
26
```

```
df_new=df.drop('mobile')
```

```
df_new.show(10)
```

```
+-------+---+----------+------+
|ratings|age|experience|family|
+-------+---+----------+------+
|      3| 32|       9.0|     3|
|      4| 22|       2.5|     0|
|      5| 27|       6.0|     0|
|      4| 22|       6.0|     1|
|      3| 27|       6.0|     0|
|      2| 32|      16.5|     2|
|      4| 27|       9.0|     0|
|      2| 27|       9.0|     2|
|      3| 37|      16.5|     5|
|      4| 27|       6.0|     1|
+-------+---+----------+------+
only showing top 10 rows
```

```
parquet_path ='gdrive/MyDrive/sample_data.csv'
```

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

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
```

```
parquet_path = '/content/gdrive/MyDrive/df_parquet'
```

```
df.write.parquet(parquet_path)
```