```
!pip install pyspark
```

```
Collecting pyspark
  Downloading pyspark-3.5.1.tar.gz (317.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 317.0/317.0 MB 2.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from py
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.5.1-py2.py3-none-any.whl size=317488491 sha256=b0
  Stored in directory: /root/.cache/pip/wheels/80/1d/60/2c256ed38dddce2fdd93be545214a63e02fbd8d74
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.5.1
```

- Pyspark Dataframe
- Reading the Data set
- Checking the data types of schema(column)
- Selecting the column and index
- Check describe option similar to pandas
- Adding column
- Dropping columns
- Renaming Columns

```
from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder.appName('Dataframe').getOrCreate()
```

```
spark
```

SparkSession - in-memory

SparkContext

Spark UI

Version
    v3.5.1
Master
    local[*]
AppName
    Practise

```
# reading the dataset
```

```
df_pyspark = spark.read.option('header','true').csv('vgsales2.csv', inferSchema = True)
```

```
# showing it
spark.read.option('header','true').csv('vgsales2.csv').show()
```

```
+----+--------------------+--------+----+------------+--------------------+--------+--------+----
|Rank|                Name|Platform|Year|       Genre|           Publisher|NA_Sales|EU_Sales|JP_Sa
+----+--------------------+--------+----+------------+--------------------+--------+--------+----
|   1|          Wii Sports|     Wii|2006|      Sports|            Nintendo|   41.49|   29.02|    
|   2|   Super Mario Bros.|     NES|1985|    Platform|            Nintendo|   29.08|    3.58|    
|   3|      Mario Kart Wii|     Wii|2008|      Racing|            Nintendo|   15.85|   12.88|    
|   4|   Wii Sports Resort|     Wii|2009|      Sports|            Nintendo|   15.75|   11.01|    
|   5|Pokemon Red/Pokem...|      GB|1996|Role-Playing|            Nintendo|   11.27|    8.89|   1
|   6|              Tetris|      GB|1989|      Puzzle|            Nintendo|    23.2|    2.26|    
|   7|New Super Mario B...|      DS|2006|    Platform|            Nintendo|   11.38|    9.23|
|   8|            Wii Play|     Wii|2006|        Misc|            Nintendo|   14.03|     9.2|    
|   9|New Super Mario B...|     Wii|2009|    Platform|            Nintendo|   14.59|    7.06|
|  10|           Duck Hunt|     NES|1984|     Shooter|            Nintendo|   26.93|    0.63|
|  11|          Nintendogs|      DS|2005|  Simulation|            Nintendo|    9.07|      11|
|  12|       Mario Kart DS|      DS|2005|      Racing|            Nintendo|    9.81|    7.57|
|  13|Pokemon Gold/Poke...|      GB|1999|Role-Playing|            Nintendo|       9|    6.18|
|  14|             Wii Fit|     Wii|2007|      Sports|            Nintendo|    8.94|    8.03|
|  15|        Wii Fit Plus|     Wii|2009|      Sports|            Nintendo|    9.09|    8.59|
|  16|  Kinect Adventures!|    X360|2010|        Misc|Microsoft Game St...|   14.97|    4.94|
|  17|  Grand Theft Auto V|     PS3|2013|      Action|Take-Two Interactive|    7.01|    9.27|
|  18|Grand Theft Auto:...|     PS2|2004|      Action|Take-Two Interactive|    9.43|     0.4|
|  19|   Super Mario World|    SNES|1990|    Platform|            Nintendo|   12.78|    3.75|
|  20|Brain Age: Train ...|      DS|2005|        Misc|            Nintendo|    4.75|    9.26|
+----+--------------------+--------+----+------------+--------------------+--------+--------+----
only showing top 20 rows
```

```
# Checking the schema
# By deafult it will read all column as string , so we use inferschema = true to handle this case
df_pyspark.printSchema()
```

```
root
 |-- Rank: integer (nullable = true)
 |-- Name: string (nullable = true)
 |-- Platform: string (nullable = true)
 |-- Year: string (nullable = true)
 |-- Genre: string (nullable = true)
 |-- Publisher: string (nullable = true)
 |-- NA_Sales: double (nullable = true)
 |-- EU_Sales: double (nullable = true)
 |-- JP_Sales: double (nullable = true)
 |-- Other_Sales: double (nullable = true)
```

```
# Another way to read the csv file or dataset
df_pyspark = spark.read.csv('vgsales2.csv', header = True , inferSchema = True)
df_pyspark.show()
```

```
----------+--------+----+------------+--------------------+--------+--------+--------+-----------+
      Name|Platform|Year|       Genre|           Publisher|NA_Sales|EU_Sales|JP_Sales|Other_Sales|
----------+--------+----+------------+--------------------+--------+--------+--------+-----------+
Wii Sports|     Wii|2006|      Sports|            Nintendo|   41.49|   29.02|    3.77|       8.46|
ario Bros.|     NES|1985|    Platform|            Nintendo|   29.08|    3.58|    6.81|       0.77|
o Kart Wii|     Wii|2008|      Racing|            Nintendo|   15.85|   12.88|    3.79|       3.31|
rts Resort|     Wii|2009|      Sports|            Nintendo|   15.75|   11.01|    3.28|       2.96|
d/Pokem...|      GB|1996|Role-Playing|            Nintendo|   11.27|    8.89|   10.22|        1.0|
    Tetris|      GB|1989|      Puzzle|            Nintendo|    23.2|    2.26|    4.22|       0.58|
Mario B...|      DS|2006|    Platform|            Nintendo|   11.38|    9.23|     6.5|        2.9|
  Wii Play|     Wii|2006|        Misc|            Nintendo|   14.03|     9.2|    2.93|       2.85|
Mario B...|     Wii|2009|    Platform|            Nintendo|   14.59|    7.06|     4.7|       2.26|
 Duck Hunt|     NES|1984|     Shooter|            Nintendo|   26.93|    0.63|    0.28|       0.47|
Nintendogs|      DS|2005|  Simulation|            Nintendo|    9.07|    11.0|    1.93|       2.75|
io Kart DS|      DS|2005|      Racing|            Nintendo|    9.81|    7.57|    4.13|       1.92|
ld/Poke...|      GB|1999|Role-Playing|            Nintendo|     9.0|    6.18|     7.2|       0.71|
   Wii Fit|     Wii|2007|      Sports|            Nintendo|    8.94|    8.03|     3.6|       2.15|
i Fit Plus|     Wii|2009|      Sports|            Nintendo|    9.09|    8.59|    2.53|       1.79|
dventures!|    X360|2010|        Misc|Microsoft Game St...|   14.97|    4.94|    0.24|       1.67|
eft Auto V|     PS3|2013|      Action|Take-Two Interactive|    7.01|    9.27|    0.97|       4.14|
t Auto:...|     PS2|2004|      Action|Take-Two Interactive|    9.43|     0.4|    0.41|      10.57|
ario World|    SNES|1990|    Platform|            Nintendo|   12.78|    3.75|    3.54|       0.55|
 Train ...|      DS|2005|        Misc|            Nintendo|    4.75|    9.26|    4.16|       2.05|
----------+--------+----+------------+--------------------+--------+--------+--------+-----------+
 20 rows
```

```
# checking the schema
df_pyspark.printSchema()
```

```
root
 |-- Rank: integer (nullable = true)
 |-- Name: string (nullable = true)
 |-- Platform: string (nullable = true)
 |-- Year: string (nullable = true)
 |-- Genre: string (nullable = true)
 |-- Publisher: string (nullable = true)
 |-- NA_Sales: double (nullable = true)
 |-- EU_Sales: double (nullable = true)
 |-- JP_Sales: double (nullable = true)
 |-- Other_Sales: double (nullable = true)
```

```
type( df_pyspark)
# it waill give dataframe as output which is a type of datastructure
```

```
pyspark.sql.dataframe.DataFrame
def __init__(jdf: JavaObject, sql_ctx: Union['SQLContext',
'SparkSession'])
    Supports Spark Connect.

Examples
--------
A :class:`DataFrame` is equivalent to a relational table in Spark SQL,
and can be created using various functions in :class:`SparkSession`:
```

```
#  see the columns
df_pyspark.columns
```

```
['Rank',
 'Name',
 'Platform',
 'Year',
 'Genre',
 'Publisher',
 'NA_Sales',
 'EU_Sales',
 'JP_Sales',
 'Other_Sales']
```

```
# see the hrad element of the dataset as a form of list
df_pyspark.head(3 )
```

```
[Row(Rank=1, Name='Wii Sports', Platform='Wii', Year='2006', Genre='Sports',
Publisher='Nintendo', NA_Sales=41.49, EU_Sales=29.02, JP_Sales=3.77, Other_Sales=8.46),
 Row(Rank=2, Name='Super Mario Bros.', Platform='NES', Year='1985', Genre='Platform',
Publisher='Nintendo', NA_Sales=29.08, EU_Sales=3.58, JP_Sales=6.81, Other_Sales=0.77),
 Row(Rank=3, Name='Mario Kart Wii', Platform='Wii', Year='2008', Genre='Racing',
Publisher='Nintendo', NA_Sales=15.85, EU_Sales=12.88, JP_Sales=3.79, Other_Sales=3.31)]
```

```
# Selecting a column
df_pyspark.select( 'Name')
# this will return the type of dataframe
```

```
DataFrame[Name: string]
```

```
df_pyspark.select( 'Name').show()
# this will return the entire data present in that column
```

```
+--------------------+
|                Name|
+--------------------+
|          Wii Sports|
|   Super Mario Bros.|
|      Mario Kart Wii|
|   Wii Sports Resort|
|Pokemon Red/Pokem...|
|              Tetris|
|New Super Mario B...|
|            Wii Play|
|New Super Mario B...|
|           Duck Hunt|
|          Nintendogs|
|       Mario Kart DS|
|Pokemon Gold/Poke...|
|             Wii Fit|
|        Wii Fit Plus|
|  Kinect Adventures!|
|  Grand Theft Auto V|
|Grand Theft Auto:...|
|   Super Mario World|
|Brain Age: Train ...|
+--------------------+
only showing top 20 rows
```

```
# Now selecting more than 1 columns
df_pyspark.select( ['Name' , 'publisher'])
```

```
DataFrame[Name: string, publisher: string]
```

```
df_pyspark.select( ['Name' , 'publisher']).show()
```

```
+--------------------+--------------------+
|                Name|           publisher|
+--------------------+--------------------+
|          Wii Sports|            Nintendo|
|   Super Mario Bros.|            Nintendo|
|      Mario Kart Wii|            Nintendo|
|   Wii Sports Resort|            Nintendo|
|Pokemon Red/Pokem...|            Nintendo|
|              Tetris|            Nintendo|
|New Super Mario B...|            Nintendo|
|            Wii Play|            Nintendo|
|New Super Mario B...|            Nintendo|
|           Duck Hunt|            Nintendo|
|          Nintendogs|            Nintendo|
|       Mario Kart DS|            Nintendo|
|Pokemon Gold/Poke...|            Nintendo|
|             Wii Fit|            Nintendo|
|        Wii Fit Plus|            Nintendo|
|  Kinect Adventures!|Microsoft Game St...|
|  Grand Theft Auto V|Take-Two Interactive|
|Grand Theft Auto:...|Take-Two Interactive|
|   Super Mario World|            Nintendo|
|Brain Age: Train ...|            Nintendo|
+--------------------+--------------------+
only showing top 20 rows
```

```
df_pyspark['Name']
# the return type of this is column
```

```
Column<'Name'>
```

```
# Now we will see how to check the datatypes
df_pyspark.dtypes
```

```
[('Rank', 'int'),
 ('Name', 'string'),
 ('Platform', 'string'),
 ('Year', 'string'),
 ('Genre', 'string'),
 ('Publisher', 'string'),
 ('NA_Sales', 'double'),
 ('EU_Sales', 'double'),
 ('JP_Sales', 'double'),
 ('Other_Sales', 'double')]
```

```
# Now checking the describe option similar to pandas
df_pyspark.describe()
```

```
DataFrame[summary: string, Rank: string, Name: string, Platform: string, Year: string, Genre:
string, Publisher: string, NA_Sales: string, EU_Sales: string, JP_Sales: string, Other_Sales:
string]
```

```
df_pyspark.describe().show()
# NULL values are shown because of mean and max and st dev of string will be NULL
```

```
+-------+-----------------+--------------------+--------+------------------+--------+-------------
|summary|             Rank|                Name|Platform|              Year|   Genre|      Publis
+-------+-----------------+--------------------+--------+------------------+--------+-------------
|  count|            16598|               16598|   16598|             16598|   16598|          165
|   mean|8300.605253645017|              1942.0|  2600.0|2006.4064433147546|    NULL|           NU
| stddev|  4791.8539328964|                NULL|     0.0| 5.828981114713253|    NULL|           NU
|    min|                1|.hack//G.U. Vol.1...|    2600|              1980|  Action|10TACLE Studi
|    max|            16600|¡Shin Chan Flipa ...|    XOne|               N/A|Strategy|   responDESI
+-------+-----------------+--------------------+--------+------------------+--------+-------------
```

```
# Adding column in dataframe
df_pyspark = df_pyspark.withColumn('Global Sales',df_pyspark['NA_Sales'] + df_pyspark['EU_Sales'] + d
# df_pyspark = df_pyspark.withColumn('Global Sales', col('NA_Sales') + col('EU_Sales') + col('JP_Sale
df_pyspark = df_pyspark.withColumn('Global + 10 ',df_pyspark['Global Sales']+ 10)

df_pyspark.show()
```

```
--------------------+--------+--------+--------+-----------+------------------+------------------+
           Publisher|NA_Sales|EU_Sales|JP_Sales|Other_Sales|      Global Sales|      Global + 10 |
--------------------+--------+--------+--------+-----------+------------------+------------------+
            Nintendo|   41.49|   29.02|    3.77|       8.46| 82.74000000000001| 92.74000000000001|
            Nintendo|   29.08|    3.58|    6.81|       0.77|             40.24|             50.24|
            Nintendo|   15.85|   12.88|    3.79|       3.31|35.830000000000005|45.830000000000005|
            Nintendo|   15.75|   11.01|    3.28|       2.96|              33.0|              43.0|
            Nintendo|   11.27|    8.89|   10.22|        1.0|31.380000000000003|             41.38|
            Nintendo|    23.2|    2.26|    4.22|       0.58|30.259999999999998|             40.26|
            Nintendo|   11.38|    9.23|     6.5|        2.9|30.009999999999998|             40.01|
            Nintendo|   14.03|     9.2|    2.93|       2.85|29.009999999999998|             39.01|
            Nintendo|   14.59|    7.06|     4.7|       2.26|             28.61|             38.61|
            Nintendo|   26.93|    0.63|    0.28|       0.47|             28.31|             38.31|
            Nintendo|    9.07|    11.0|    1.93|       2.75|             24.75|             34.75|
            Nintendo|    9.81|    7.57|    4.13|       1.92|             23.43|             33.43|
            Nintendo|     9.0|    6.18|     7.2|       0.71|             23.09|             33.09|
            Nintendo|    8.94|    8.03|     3.6|       2.15|             22.72|             32.72|
            Nintendo|    9.09|    8.59|    2.53|       1.79|              22.0|              32.0|
Microsoft Game St...|   14.97|    4.94|    0.24|       1.67|             21.82|             31.82|
Take-Two Interactive|    7.01|    9.27|    0.97|       4.14|             21.39|             31.39|
Take-Two Interactive|    9.43|     0.4|    0.41|      10.57|20.810000000000002|30.810000000000002|
            Nintendo|   12.78|    3.75|    3.54|       0.55|             20.62|             30.62|
            Nintendo|    4.75|    9.26|    4.16|       2.05|20.220000000000002|30.220000000000002|
--------------------+--------+--------+--------+-----------+------------------+------------------+
```

```
# Drop the columns
df_pyspark = df_pyspark.drop('Global + 10 ')
df_pyspark.show()
```

```
+----+--------------------+--------+----+------------+--------------------+--------+--------+-----
|Rank|                Name|Platform|Year|       Genre|           Publisher|NA_Sales|EU_Sales|JP_Sa
+----+--------------------+--------+----+------------+--------------------+--------+--------+-----
|   1|          Wii Sports|     Wii|2006|      Sports|            Nintendo|   41.49|   29.02|    3
|   2|   Super Mario Bros.|     NES|1985|    Platform|            Nintendo|   29.08|    3.58|    6
|   3|      Mario Kart Wii|     Wii|2008|      Racing|            Nintendo|   15.85|   12.88|    3
|   4|   Wii Sports Resort|     Wii|2009|      Sports|            Nintendo|   15.75|   11.01|    3
|   5|Pokemon Red/Pokem...|      GB|1996|Role-Playing|            Nintendo|   11.27|    8.89|   10
|   6|              Tetris|      GB|1989|      Puzzle|            Nintendo|    23.2|    2.26|    4
|   7|New Super Mario B...|      DS|2006|    Platform|            Nintendo|   11.38|    9.23|
|   8|            Wii Play|     Wii|2006|        Misc|            Nintendo|   14.03|     9.2|    2
|   9|New Super Mario B...|     Wii|2009|    Platform|            Nintendo|   14.59|    7.06|
|  10|           Duck Hunt|     NES|1984|     Shooter|            Nintendo|   26.93|    0.63|    0
|  11|          Nintendogs|      DS|2005|  Simulation|            Nintendo|    9.07|    11.0|    1
|  12|       Mario Kart DS|      DS|2005|      Racing|            Nintendo|    9.81|    7.57|
|  13|Pokemon Gold/Poke...|      GB|1999|Role-Playing|            Nintendo|     9.0|    6.18|
|  14|             Wii Fit|     Wii|2007|      Sports|            Nintendo|    8.94|    8.03|
|  15|        Wii Fit Plus|     Wii|2009|      Sports|            Nintendo|    9.09|    8.59|
|  16|  Kinect Adventures!|    X360|2010|        Misc|Microsoft Game St...|   14.97|    4.94|
|  17|  Grand Theft Auto V|     PS3|2013|      Action|Take-Two Interactive|    7.01|    9.27|
|  18|Grand Theft Auto:...|     PS2|2004|      Action|Take-Two Interactive|    9.43|     0.4|
|  19|   Super Mario World|    SNES|1990|    Platform|            Nintendo|   12.78|    3.75|
|  20|Brain Age: Train ...|      DS|2005|        Misc|            Nintendo|    4.75|    9.26|
+----+--------------------+--------+----+------------+--------------------+--------+--------+----
only showing top 20 rows
```

```
# Rename the column
df_pyspark.withColumnRenamed('Name', 'New Name').show()
```

```
+----+--------------------+--------+----+------------+--------------------+--------+--------+----
|Rank|            New Name|Platform|Year|       Genre|           Publisher|NA_Sales|EU_Sales|JP_S
+----+--------------------+--------+----+------------+--------------------+--------+--------+----
|   1|          Wii Sports|     Wii|2006|      Sports|            Nintendo|   41.49|   29.02|
|   2|   Super Mario Bros.|     NES|1985|    Platform|            Nintendo|   29.08|    3.58|
|   3|      Mario Kart Wii|     Wii|2008|      Racing|            Nintendo|   15.85|   12.88|
|   4|   Wii Sports Resort|     Wii|2009|      Sports|            Nintendo|   15.75|   11.01|
|   5|Pokemon Red/Pokem...|      GB|1996|Role-Playing|            Nintendo|   11.27|    8.89|   1
|   6|              Tetris|      GB|1989|      Puzzle|            Nintendo|    23.2|    2.26|
|   7|New Super Mario B...|      DS|2006|    Platform|            Nintendo|   11.38|    9.23|
|   8|            Wii Play|     Wii|2006|        Misc|            Nintendo|   14.03|     9.2|
|   9|New Super Mario B...|     Wii|2009|    Platform|            Nintendo|   14.59|    7.06|
|  10|           Duck Hunt|     NES|1984|     Shooter|            Nintendo|   26.93|    0.63|
|  11|          Nintendogs|      DS|2005|  Simulation|            Nintendo|    9.07|    11.0|
|  12|       Mario Kart DS|      DS|2005|      Racing|            Nintendo|    9.81|    7.57|
|  13|Pokemon Gold/Poke...|      GB|1999|Role-Playing|            Nintendo|     9.0|    6.18|
|  14|             Wii Fit|     Wii|2007|      Sports|            Nintendo|    8.94|    8.03|
|  15|        Wii Fit Plus|     Wii|2009|      Sports|            Nintendo|    9.09|    8.59|
|  16|  Kinect Adventures!|    X360|2010|        Misc|Microsoft Game St...|   14.97|    4.94|
|  17|  Grand Theft Auto V|     PS3|2013|      Action|Take-Two Interactive|    7.01|    9.27|
|  18|Grand Theft Auto:...|     PS2|2004|      Action|Take-Two Interactive|    9.43|     0.4|
|  19|   Super Mario World|    SNES|1990|    Platform|            Nintendo|   12.78|    3.75|
|  20|Brain Age: Train ...|      DS|2005|        Misc|            Nintendo|    4.75|    9.26|
+----+--------------------+--------+----+------------+--------------------+--------+--------+----
only showing top 20 rows
```

```
from pyspark.sql import SparkSession

# Initialize SparkSession
spark = SparkSession.builder.appName("Download DataFrame as CSV").getOrCreate()

# Assuming df_pyspark is your PySpark DataFrame

# Convert PySpark DataFrame to Pandas DataFrame
df_pandas = df_pyspark.toPandas()

# Save Pandas DataFrame as CSV file
df_pandas.to_csv("output.csv", index=False)

# Download the CSV file in Colab
from google.colab import files
files.download("output.csv")
```

Handling Missing values

- Drop Columns
- Dropping Rows
- Various parameters in Dropping functionalities

- Handling missing values by mean, median and mode

```
from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder.appName('Practise').getOrCreate()
```

```
df_pyspark = spark.read.csv('test2.csv', header = True , inferSchema = True )
```

```
df_pyspark.show()
```

```
+---------+----+----------+------+
|     Name| age|Experience|Salary|
+---------+----+----------+------+
|    Krish|  31|        10| 30000|
|Sudhanshu|  30|         8| 25000|
|    Sunny|  29|         4| 20000|
|     Paul|  24|         3| 20000|
|   Harsha|  21|         1| 15000|
|  Shubham|  23|         2| 18000|
|   Mahesh|NULL|      NULL| 40000|
|     NULL|  34|        10| 38000|
|     NULL|  36|      NULL|  NULL|
+---------+----+----------+------+
```

```
# drop the colums
df_pyspark.drop('Name').show()
```

```
+----+----------+------+
| age|Experience|Salary|
+----+----------+------+
|  31|        10| 30000|
|  30|         8| 25000|
|  29|         4| 20000|
|  24|         3| 20000|
|  21|         1| 15000|
|  23|         2| 18000|
|NULL|      NULL| 40000|
|  34|        10| 38000|
|  36|      NULL|  NULL|
+----+----------+------+
```

```
df_pyspark.show()
```

```
+---------+----+----------+------+
|     Name| age|Experience|Salary|
+---------+----+----------+------+
|    Krish|  31|        10| 30000|
|Sudhanshu|  30|         8| 25000|
|    Sunny|  29|         4| 20000|
|     Paul|  24|         3| 20000|
|   Harsha|  21|         1| 15000|
|  Shubham|  23|         2| 18000|
|   Mahesh|NULL|      NULL| 40000|
|     NULL|  34|        10| 38000|
|     NULL|  36|      NULL|  NULL|
+---------+----+----------+------+
```

```
# Deleting the row values
```

```
df_pyspark.na.drop().show()
# By using this without any parameter, null rows will be deleted containing it
# In this case last 3 rows will be deleted
```

```
+---------+---+----------+------+
|     Name|age|Experience|Salary|
+---------+---+----------+------+
|    Krish| 31|        10| 30000|
|Sudhanshu| 30|         8| 25000|
|    Sunny| 29|         4| 20000|
|     Paul| 24|         3| 20000|
|   Harsha| 21|         1| 15000|
|  Shubham| 23|         2| 18000|
+---------+---+----------+------+
```

```
#  if any = how
df_pyspark.na.drop( how = 'all').show()
# In this case the rows which are having all the null values in it will be dropped
```

```
+---------+----+----------+------+
|     Name| age|Experience|Salary|
+---------+----+----------+------+
|    Krish|  31|        10| 30000|
|Sudhanshu|  30|         8| 25000|
|    Sunny|  29|         4| 20000|
|     Paul|  24|         3| 20000|
|   Harsha|  21|         1| 15000|
|  Shubham|  23|         2| 18000|
|   Mahesh|NULL|      NULL| 40000|
|     NULL|  34|        10| 38000|
|     NULL|  36|      NULL|  NULL|
+---------+----+----------+------+
```

```
#  if how = any
df_pyspark.na.drop( how = 'any').show()
# Drop the rows having andy null value in it
```

```
+---------+---+----------+------+
|     Name|age|Experience|Salary|
+---------+---+----------+------+
|    Krish| 31|        10| 30000|
|Sudhanshu| 30|         8| 25000|
|    Sunny| 29|         4| 20000|
|     Paul| 24|         3| 20000|
|   Harsha| 21|         1| 15000|
|  Shubham| 23|         2| 18000|
+---------+---+----------+------+
```

```
#Using Threshold value
df_pyspark.na.drop( how = 'any' , thresh = 2).show()
# Drop the rows in which atleast 2 non null values are present for a row
```

```
+---------+----+----------+------+
|     Name| age|Experience|Salary|
+---------+----+----------+------+
|    Krish|  31|        10| 30000|
|Sudhanshu|  30|         8| 25000|
|    Sunny|  29|         4| 20000|
|     Paul|  24|         3| 20000|
|   Harsha|  21|         1| 15000|
|  Shubham|  23|         2| 18000|
|   Mahesh|NULL|      NULL| 40000|
|     NULL|  34|        10| 38000|
+---------+----+----------+------+
```

```
# Subset
df_pyspark.na.drop( how = 'all', subset = ['Experience']).show()
#  In this case the rows in which null values are present in Experience column got deleted
```

```
+---------+---+----------+------+
|     Name|age|Experience|Salary|
+---------+---+----------+------+
|    Krish| 31|        10| 30000|
|Sudhanshu| 30|         8| 25000|
|    Sunny| 29|         4| 20000|
|     Paul| 24|         3| 20000|
|   Harsha| 21|         1| 15000|
|  Shubham| 23|         2| 18000|
|     NULL| 34|        10| 38000|
+---------+---+----------+------+
```

```
# Filling the missing values
df_pyspark.na.fill('Missing Values', 'age').show()
```

```
+---------+----+----------+------+
|     Name| age|Experience|Salary|
+---------+----+----------+------+
|    Krish|  31|        10| 30000|
|Sudhanshu|  30|         8| 25000|
|    Sunny|  29|         4| 20000|
|     Paul|  24|         3| 20000|
|   Harsha|  21|         1| 15000|
|  Shubham|  23|         2| 18000|
|   Mahesh|NULL|      NULL| 40000|
|     NULL|  34|        10| 38000|
|     NULL|  36|      NULL|  NULL|
+---------+----+----------+------+
```

```
df_pyspark.na.fill('Missing Values').show()
```

```
+--------------+----+----------+------+
|          Name| age|Experience|Salary|
+--------------+----+----------+------+
|         Krish|  31|        10| 30000|
|     Sudhanshu|  30|         8| 25000|
|         Sunny|  29|         4| 20000|
|          Paul|  24|         3| 20000|
|        Harsha|  21|         1| 15000|
|       Shubham|  23|         2| 18000|
|        Mahesh|NULL|      NULL| 40000|
|Missing Values|  34|        10| 38000|
|Missing Values|  36|      NULL|  NULL|
+--------------+----+----------+------+
```

```
# df_pyspark.withColumn("Name", when(col("Name") == "NULL", None).otherwise(col("Text")))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-19-5e158572894d> in <cell line: 1>()
----> 1 df_pyspark.withColumn("Name", when(col("Name") == "NULL",
None).otherwise(col("Text")))

NameError: name 'when' is not defined
```

```
# Now Handling the NULL values with the Mean
df_pyspark.show()
```

```
+---------+----+----------+------+
|     Name| age|Experience|Salary|
+---------+----+----------+------+
|    Krish|  31|        10| 30000|
|Sudhanshu|  30|         8| 25000|
|    Sunny|  29|         4| 20000|
|     Paul|  24|         3| 20000|
|   Harsha|  21|         1| 15000|
|  Shubham|  23|         2| 18000|
|   Mahesh|NULL|      NULL| 40000|
|     NULL|  34|        10| 38000|
|     NULL|  36|      NULL|  NULL|
+---------+----+----------+------+
```

```
from pyspark.ml.feature import Imputer
imputer = Imputer(
    inputCols = ['age' , 'Experience' , 'Salary'],
    outputCols = ["{}_imputed".format(c) for c in ['age', 'Experience','Salary']]

).setStrategy("mean")


# Add imputation Col to df
imputer.fit(df_pyspark).transform(df_pyspark).show()
```

```
+---------+----+----------+------+-----------+------------------+--------------+
|     Name| age|Experience|Salary|age_imputed|Experience_imputed|Salary_imputed|
+---------+----+----------+------+-----------+------------------+--------------+
|    Krish|  31|        10| 30000|         31|                10|         30000|
|Sudhanshu|  30|         8| 25000|         30|                 8|         25000|
|    Sunny|  29|         4| 20000|         29|                 4|         20000|
|     Paul|  24|         3| 20000|         24|                 3|         20000|
|   Harsha|  21|         1| 15000|         21|                 1|         15000|
|  Shubham|  23|         2| 18000|         23|                 2|         18000|
|   Mahesh|NULL|      NULL| 40000|         28|                 5|         40000|
|     NULL|  34|        10| 38000|         34|                10|         38000|
|     NULL|  36|      NULL|  NULL|         36|                 5|         25750|
+---------+----+----------+------+-----------+------------------+--------------+
```

```
# Now Handling the NULL values with the Mode
df_pyspark.show()
```

```
+---------+----+----------+------+
|     Name| age|Experience|Salary|
+---------+----+----------+------+
|    Krish|  31|        10| 30000|
|Sudhanshu|  30|         8| 25000|
|    Sunny|  29|         4| 20000|
|     Paul|  24|         3| 20000|
|   Harsha|  21|         1| 15000|
|  Shubham|  23|         2| 18000|
|   Mahesh|NULL|      NULL| 40000|
|     NULL|  34|        10| 38000|
|     NULL|  36|      NULL|  NULL|
+---------+----+----------+------+
```

```
from pyspark.ml.feature import Imputer
imputer = Imputer(
    inputCols = ['age' , 'Experience' , 'Salary'],
    outputCols = ["{}_imputed".format(c) for c in ['age', 'Experience','Salary']]

).setStrategy("mode")


# Add imputation Col to df
imputer.fit(df_pyspark).transform(df_pyspark).show()
```

```
+---------+----+----------+------+-----------+------------------+--------------+
|     Name| age|Experience|Salary|age_imputed|Experience_imputed|Salary_imputed|
+---------+----+----------+------+-----------+------------------+--------------+
|    Krish|  31|        10| 30000|         31|                10|         30000|
|Sudhanshu|  30|         8| 25000|         30|                 8|         25000|
|    Sunny|  29|         4| 20000|         29|                 4|         20000|
|     Paul|  24|         3| 20000|         24|                 3|         20000|
|   Harsha|  21|         1| 15000|         21|                 1|         15000|
|  Shubham|  23|         2| 18000|         23|                 2|         18000|
|   Mahesh|NULL|      NULL| 40000|         21|                10|         40000|
|     NULL|  34|        10| 38000|         34|                10|         38000|
|     NULL|  36|      NULL|  NULL|         36|                10|         20000|
+---------+----+----------+------+-----------+------------------+--------------+
```

```
# Now Handling the NULL values with the Median
df_pyspark.show()
```

```
+---------+----+----------+------+
|     Name| age|Experience|Salary|
+---------+----+----------+------+
|    Krish|  31|        10| 30000|
|Sudhanshu|  30|         8| 25000|
|    Sunny|  29|         4| 20000|
|     Paul|  24|         3| 20000|
|   Harsha|  21|         1| 15000|
|  Shubham|  23|         2| 18000|
|   Mahesh|NULL|      NULL| 40000|
|     NULL|  34|        10| 38000|
|     NULL|  36|      NULL|  NULL|
+---------+----+----------+------+
```

```
from pyspark.ml.feature import Imputer
imputer = Imputer(
    inputCols = ['age' , 'Experience' , 'Salary'],
    outputCols = ["{}_imputed".format(c) for c in ['age', 'Experience','Salary']]

).setStrategy("median")
```

```
# Add imputation Col to df
imputer.fit(df_pyspark).transform(df_pyspark).show()
```

```
+---------+----+----------+------+-----------+------------------+--------------+
|     Name| age|Experience|Salary|age_imputed|Experience_imputed|Salary_imputed|
+---------+----+----------+------+-----------+------------------+--------------+
|    Krish|  31|        10| 30000|         31|                10|         30000|
|Sudhanshu|  30|         8| 25000|         30|                 8|         25000|
|    Sunny|  29|         4| 20000|         29|                 4|         20000|
|     Paul|  24|         3| 20000|         24|                 3|         20000|
|   Harsha|  21|         1| 15000|         21|                 1|         15000|
|  Shubham|  23|         2| 18000|         23|                 2|         18000|
|   Mahesh|NULL|      NULL| 40000|         29|                 4|         40000|
|     NULL|  34|        10| 38000|         34|                10|         38000|
|     NULL|  36|      NULL|  NULL|         36|                 4|         20000|
+---------+----+----------+------+-----------+------------------+--------------+
```

- Filter Operation
- & ,| , ==
- ~

```
from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder.appName('dataframe').getOrCreate()
```

```
df_pyspark = spark.read.csv('test1.csv', header = True , inferSchema = True )
```

```
df_pyspark.show()
```

```
+---------+---+----------+------+
|     Name|age|Experience|Salary|
+---------+---+----------+------+
|    Krish| 31|        10| 30000|
|Sudhanshu| 30|         8| 25000|
|    Sunny| 29|         4| 20000|
|     Paul| 24|         3| 20000|
|   Harsha| 21|         1| 15000|
|  Shubham| 23|         2| 18000|
+---------+---+----------+------+
```

```
# Filter Operations
# Salary of the people less than or equal 20000
df_pyspark.filter("Salary<=20000").show()
```

```
    +-------+---+----------+------+
    |   Name|age|Experience|Salary|
    +-------+---+----------+------+
    |  Sunny| 29|         4| 20000|
    |   Paul| 24|         3| 20000|
    | Harsha| 21|         1| 15000|
    |Shubham| 23|         2| 18000|
    +-------+---+----------+------+
```

```
# Aliter
df_pyspark.filter(df_pyspark['Salary'] <= 20000).show()
```

```
    +-------+---+----------+------+
    |   Name|age|Experience|Salary|
    +-------+---+----------+------+
    |  Sunny| 29|         4| 20000|
    |   Paul| 24|         3| 20000|
    | Harsha| 21|         1| 15000|
    |Shubham| 23|         2| 18000|
    +-------+---+----------+------+
```

```
df_pyspark.filter("Salary<=20000").select(['Name','age']).show()
```

```
    +-------+---+
    |   Name|age|
    +-------+---+
    |  Sunny| 29|
    |   Paul| 24|
    | Harsha| 21|
    |Shubham| 23|
    +-------+---+
```

```
# Use of multiple condition with &
df_pyspark.filter((df_pyspark['Salary'] <= 20000) &
                  (df_pyspark['Salary'] >= 18000)).show()
```

```
    +-------+---+----------+------+
    |   Name|age|Experience|Salary|
    +-------+---+----------+------+
    |  Sunny| 29|         4| 20000|
    |   Paul| 24|         3| 20000|
    |Shubham| 23|         2| 18000|
    +-------+---+----------+------+
```

```
# Use of multiple condition with |
df_pyspark.filter((df_pyspark['Salary'] <= 20000) |
                  (df_pyspark['Salary'] >= 18000)).show()
```

```
    +---------+---+----------+------+
    |     Name|age|Experience|Salary|
    +---------+---+----------+------+
    |    Krish| 31|        10| 30000|
    |Sudhanshu| 30|         8| 25000|
    |    Sunny| 29|         4| 20000|
    |     Paul| 24|         3| 20000|
    |   Harsha| 21|         1| 15000|
    |  Shubham| 23|         2| 18000|
    +---------+---+----------+------+
```

```
# Use of Not( ~ ) condition
df_pyspark.filter(~(df_pyspark['Salary'] <= 20000)).show()
```

```
+---------+---+----------+------+
|     Name|age|Experience|Salary|
+---------+---+----------+------+
|    Krish| 31|        10| 30000|
|Sudhanshu| 30|         8| 25000|
+---------+---+----------+------+
```

- Group by
- And Aggegate functions

```
from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder.appName('Agg').getOrCreate()
```

```
df_pyspark = spark.read.csv('test3.csv', header = True , inferSchema = True )
```

```
df_pyspark.show()
```

```
+---------+------------+------+
|     Name| Departments|salary|
+---------+------------+------+
|    Krish|Data Science| 10000|
|    Krish|         IOT|  5000|
|   Mahesh|    Big Data|  4000|
|    Krish|    Big Data|  4000|
|   Mahesh|Data Science|  3000|
|Sudhanshu|Data Science| 20000|
|Sudhanshu|         IOT| 10000|
|Sudhanshu|    Big Data|  5000|
|    Sunny|Data Science| 10000|
|    Sunny|    Big Data|  2000|
+---------+------------+------+
```

```
df_pyspark.printSchema()
```

```
root
 |-- Name: string (nullable = true)
 |-- Departments: string (nullable = true)
 |-- salary: integer (nullable = true)
```

```
# Group by operation
# gives the total salary by Name as group
df_pyspark.groupBy('Name').sum().show()
```

```
+---------+-----------+
|     Name|sum(salary)|
+---------+-----------+
|Sudhanshu|      35000|
|    Sunny|      12000|
|    Krish|      19000|
|   Mahesh|       7000|
    +---------+-----------+
```

```
# who is getting maximum salary
df_pyspark.groupBy('Name').max().show()
```

```
    +---------+-----------+
    |     Name|max(salary)|
    +---------+-----------+
    |Sudhanshu|      20000|
    |    Sunny|      10000|
    |    Krish|      10000|
    |   Mahesh|       4000|
    +---------+-----------+
```

```
# who is getting minimum salary
df_pyspark.groupBy('Name').min().show()
```

```
    +---------+-----------+
    |     Name|min(salary)|
    +---------+-----------+
    |Sudhanshu|       5000|
    |    Sunny|       2000|
    |    Krish|       4000|
    |   Mahesh|       3000|
    +---------+-----------+
```

```
# group by department to find the maximum salary
df_pyspark.groupBy('Departments').sum().show()
```

```
    +------------+-----------+
    | Departments|sum(salary)|
    +------------+-----------+
    |         IOT|      15000|
    |    Big Data|      15000|
    |Data Science|      43000|
    +------------+-----------+
```

```
#  to find the Department wise mean
df_pyspark.groupBy('Departments').mean().show()
```

```
    +------------+-----------+
    | Departments|avg(salary)|
    +------------+-----------+
    |         IOT|     7500.0|
    |    Big Data|     3750.0|
    |Data Science|    10750.0|
    +------------+-----------+
```

```
# Number of employees using count
df_pyspark.groupBy('Departments').count().show()
```

```
    +------------+-----+
    | Departments|count|
    +------------+-----+
    |         IOT|    2|
    |    Big Data|    4|
    |Data Science|    4|
    +------------+-----+
```

```
df_pyspark.agg({'Salary':'sum'}).show()
```

```
+-----------+
|sum(Salary)|
+-----------+
|      73000|
+-----------+
```