```
sc

spark

dbutils.fs.ls("/FileStore/shared_uploads/noelroberttemp@outlook.com/")

Out[3]:
[FileInfo(path='dbfs:/FileStore/shared_uploads/noelroberttemp@outlook.
com/Customers.csv', name='Customers.csv', size=19786,
modificationTime=1727336006000),

FileInfo(path='dbfs:/FileStore/shared_uploads/noelroberttemp@outlook.c
om/sales_data_sample.csv', name='sales_data_sample.csv', size=527958,
modificationTime=1727091256000)]
```

dbutils.fs.ls(path) -> filenames list dbutils.fs.rm(path) -> remove a file

```
df = spark.read.format('csv').option('header',
'true').option('inferschema',
'true').load('dbfs:/FileStore/shared_uploads/noelroberttemp@outlook.co
m/sales_data_sample.csv')

df.show()

+-----------+--------------+---------+--------------+-------
+--------------+-------+------+--------+-------+-----------+----
+-----------+--------------------+---------------
+--------------------+------------+-------------+--------+----------
+---------+---------+--------------+----------------+--------+
|ORDERNUMBER|QUANTITYORDERED|PRICEEACH|ORDERLINENUMBER|  SALES|
ORDERDATE| STATUS|QTR_ID|MONTH_ID|YEAR_ID|PRODUCTLINE|MSRP|
PRODUCTCODE|        CUSTOMERNAME|           PHONE|
ADDRESSLINE1|ADDRESSLINE2|         CITY|   STATE|POSTALCODE|  COUNTRY|
TERRITORY|CONTACTLASTNAME|CONTACTFIRSTNAME|DEALSIZE|
+-----------+--------------+---------+--------------+-------
+--------------+-------+------+--------+-------+-----------+----
+-----------+--------------------+---------------
+--------------------+------------+-------------+--------+----------
+---------+---------+--------------+----------------+--------+
|      10107|             30|     95.7|             2| 2871.0|
2/24/2003 0:00|Shipped|     1|       2|   2003|Motorcycles|  95|
S10_1678|   Land of Toys Inc.|      2125557818|897 Long Airport ...|
null|          NYC|      NY|     10022|      USA|       NA|
Yu|            Kwai|   Small|
|      10121|             34|    81.35|             5| 2765.9|
5/7/2003 0:00|Shipped|     2|       5|   2003|Motorcycles|  95|
S10_1678|  Reims Collectables|      26.47.1555|  59 rue de l'Abbaye|
null|        Reims|    null|     51100|   France|     EMEA|
Henriot|            Paul|   Small|
|      10134|             41|    94.74|             2|3884.34|
```

```
7/1/2003 0:00|Shipped|     3|       7|   2003|Motorcycles|  95|
S10_1678|     Lyon Souveniers|+33 1 46 62 7555|27 rue du Colonel...|
null|        Paris|    null|     75508|   France|     EMEA|       Da
Cunha|          Daniel|  Medium|
|      10145|             45|    83.26|             6| 3746.7|
8/25/2003 0:00|Shipped|     3|       8|   2003|Motorcycles|  95|
S10_1678|   Toys4GrownUps.com|      6265557265|  78934 Hillside Dr.|
null|     Pasadena|      CA|     90003|      USA|       NA|
Young|           Julie|  Medium|
|      10159|             49|    100.0|            14|5205.27|
10/10/2003 0:00|Shipped|     4|      10|   2003|Motorcycles|  95|
S10_1678|Corporate Gift Id...|      6505551386|     7734 Strong St.|
null|San Francisco|      CA|      null|      USA|       NA|
Brown|           Julie|  Medium|
|      10168|             36|    96.66|             1|3479.76|
10/28/2003 0:00|Shipped|     4|      10|   2003|Motorcycles|  95|
S10_1678|Technics Stores Inc.|      6505556809|   9408 Furth Circle|
null|   Burlingame|      CA|     94217|      USA|       NA|
Hirano|            Juri|  Medium|
|      10180|             29|    86.13|             9|2497.77|
11/11/2003 0:00|Shipped|     4|      11|   2003|Motorcycles|  95|
S10_1678|Daedalus Designs ...|      20.16.1555|184, chausse de T...|
null|        Lille|    null|     59000|   France|     EMEA|
Rance|         Martine|   Small|
|      10188|             48|    100.0|             1|5512.32|
11/18/2003 0:00|Shipped|     4|      11|   2003|Motorcycles|  95|
S10_1678|        Herkku Gifts|   +47 2267 3215|Drammen 121, PR 7...|
null|       Bergen|    null|    N 5804|   Norway|     EMEA|
Oeztan|          Veysel|  Medium|
|      10201|             22|    98.57|             2|2168.54|
12/1/2003 0:00|Shipped|     4|      12|   2003|Motorcycles|  95|
S10_1678|     Mini Wheels Co.|      6505555787|5557 North Pendal...|
null|San Francisco|      CA|      null|      USA|       NA|
Murphy|           Julie|   Small|
|      10211|             41|    100.0|            14|4708.44|
1/15/2004 0:00|Shipped|     1|       1|   2004|Motorcycles|  95|
S10_1678|    Auto Canal Petit|  (1) 47.55.6555|   25, rue Lauriston|
null|        Paris|    null|     75016|   France|     EMEA|
Perrier|       Dominique|  Medium|
|      10223|             37|    100.0|             1|3965.66|
2/20/2004 0:00|Shipped|     1|       2|   2004|Motorcycles|  95|
S10_1678|Australian Collec...|    03 9520 4555|   636 St Kilda Road|
Level 3|    Melbourne|Victoria|      3004|Australia|     APAC|
Ferguson|           Peter|  Medium|
|      10237|             23|    100.0|             7|2333.12|
4/5/2004 0:00|Shipped|     2|       4|   2004|Motorcycles|  95|
S10_1678|     Vitachrome Inc.|      2125551500|   2678 Kingston Rd.|
Suite 101|          NYC|      NY|     10022|      USA|       NA|
Frick|         Michael|   Small|
```

```
|      10251|              28|    100.0|              2|3188.64|
5/18/2004 0:00|Shipped|     2|       5|   2004|Motorcycles|  95|
S10_1678|Tekni Collectable...|      2015559350|       7476 Moss Rd.|
null|       Newark|      NJ|     94019|      USA|       NA|
Brown|         William|  Medium|
|      10263|              34|    100.0|              2|3676.76|
6/28/2004 0:00|Shipped|     2|       6|   2004|Motorcycles|  95|
S10_1678|     Gift Depot Inc.|      2035552570| 25593 South Bay Ln.|
null|  Bridgewater|      CT|     97562|      USA|       NA|
King|           Julie|  Medium|
|      10275|              45|    92.83|              1|4177.35|
7/23/2004 0:00|Shipped|     3|       7|   2004|Motorcycles|  95|
S10_1678|   La Rochelle Gifts|      40.67.8555|67, rue des Cinqu...|
null|       Nantes|    null|     44000|   France|     EMEA|
Labrune|          Janine|  Medium|
|      10285|              36|    100.0|              6|4099.68|
8/27/2004 0:00|Shipped|     3|       8|   2004|Motorcycles|  95|
S10_1678|Marta's Replicas Co.|      6175558555| 39323 Spinnaker Dr.|
null|    Cambridge|      MA|     51247|      USA|       NA|
Hernandez|           Marta|  Medium|
|      10299|              23|    100.0|              9|2597.39|
9/30/2004 0:00|Shipped|     3|       9|   2004|Motorcycles|  95|
S10_1678|Toys of Finland, Co.|     90-224 8555|       Keskuskatu 45|
null|     Helsinki|    null|     21240|  Finland|     EMEA|
Karttunen|           Matti|   Small|
|      10309|              41|    100.0|              5|4394.38|
10/15/2004 0:00|Shipped|     4|      10|   2004|Motorcycles|  95|
S10_1678|  Baane Mini Imports|      07-98 9555|Erling Skakkes ga...|
null|      Stavern|    null|      4110|   Norway|     EMEA|
Bergulfsen|           Jonas|  Medium|
|      10318|              46|    94.74|              1|4358.04|
11/2/2004 0:00|Shipped|     4|      11|   2004|Motorcycles|  95|
S10_1678|Diecast Classics ...|      2155551555|    7586 Pompton St.|
null|    Allentown|      PA|     70267|      USA|       NA|
Yu|           Kyung|  Medium|
|      10329|              42|    100.0|              1|4396.14|
11/15/2004 0:00|Shipped|     4|      11|   2004|Motorcycles|  95|
S10_1678|   Land of Toys Inc.|      2125557818|897 Long Airport ...|
null|          NYC|      NY|     10022|      USA|       NA|
Yu|           Kwai|  Medium|
+-----------+--------------+---------+--------------+-------
+--------------+-------+------+--------+-------+-----------+----
+-----------+--------------------+----------------
+--------------------+------------+-------------+--------+----------
+---------+---------+--------------+----------------+--------+
only showing top 20 rows


df.printSchema()
```

```
root
 |-- ORDERNUMBER: integer (nullable = true)
 |-- QUANTITYORDERED: integer (nullable = true)
 |-- PRICEEACH: double (nullable = true)
 |-- ORDERLINENUMBER: integer (nullable = true)
 |-- SALES: double (nullable = true)
 |-- ORDERDATE: string (nullable = true)
 |-- STATUS: string (nullable = true)
 |-- QTR_ID: integer (nullable = true)
 |-- MONTH_ID: integer (nullable = true)
 |-- YEAR_ID: integer (nullable = true)
 |-- PRODUCTLINE: string (nullable = true)
 |-- MSRP: integer (nullable = true)
 |-- PRODUCTCODE: string (nullable = true)
 |-- CUSTOMERNAME: string (nullable = true)
 |-- PHONE: string (nullable = true)
 |-- ADDRESSLINE1: string (nullable = true)
 |-- ADDRESSLINE2: string (nullable = true)
 |-- CITY: string (nullable = true)
 |-- STATE: string (nullable = true)
 |-- POSTALCODE: string (nullable = true)
 |-- COUNTRY: string (nullable = true)
 |-- TERRITORY: string (nullable = true)
 |-- CONTACTLASTNAME: string (nullable = true)
 |-- CONTACTFIRSTNAME: string (nullable = true)
 |-- DEALSIZE: string (nullable = true)
```

```
from pyspark.sql.functions import to_timestamp, col, dayofweek

df = df.withColumn('ORDERDATE', to_timestamp(col('ORDERDATE'),
'M/d/yyyy H:mm'))
df = df.withColumn('DAY_OF_WEEK', dayofweek(col('ORDERDATE')))

df.createOrReplaceTempView('productSales')
```

1. Summary of Product Sales

```
%sql
SELECT COUNT(*) as sales_count FROM productSales

Databricks visualization. Run in Databricks to view.

%sql
SELECT ROUND(SUM(SALES), 2) AS total_sales FROM productSales

Databricks visualization. Run in Databricks to view.
```

```
%sql
SELECT ROUND(SUM(SALES - PRICEEACH*QUANTITYORDERED), 2) AS
total_profit FROM productSales

Databricks visualization. Run in Databricks to view.

%sql
SELECT COUNT(DISTINCT CUSTOMERNAME) AS total_customers FROM
productSales

Databricks visualization. Run in Databricks to view.

%sql
SELECT COUNT(DISTINCT PRODUCTCODE) AS total_products FROM productSales

Databricks visualization. Run in Databricks to view.
```

2. Quantity Sold per Product

```
%sql
SELECT PRODUCTLINE, SUM(QUANTITYORDERED) AS ProductsSold FROM
productSales
GROUP BY PRODUCTLINE
ORDER BY PRODUCTLINE

Databricks visualization. Run in Databricks to view.
```

3. Quantity Sold per Month

```
%sql
SELECT MONTH_ID, SUM(QUANTITYORDERED) AS ProductsSold,
      CASE
          WHEN MONTH_ID = 1 THEN 'January'
          WHEN MONTH_ID = 2 THEN 'February'
          WHEN MONTH_ID = 3 THEN 'March'
          WHEN MONTH_ID = 4 THEN 'April'
          WHEN MONTH_ID = 5 THEN 'May'
          WHEN MONTH_ID = 6 THEN 'June'
          WHEN MONTH_ID = 7 THEN 'July'
          WHEN MONTH_ID = 8 THEN 'August'
          WHEN MONTH_ID = 9 THEN 'September'
          WHEN MONTH_ID = 10 THEN 'October'
          WHEN MONTH_ID = 11 THEN 'November'
          WHEN MONTH_ID = 12 THEN 'December'
      END AS month_name
FROM productSales
```

```
GROUP BY MONTH_ID
ORDER BY MONTH_ID
```

```
Databricks visualization. Run in Databricks to view.
```

4. Month-on-month Sales growth

```
%sql
SELECT CAST(DAY(ORDERDATE) AS int) AS day, CAST(MONTH(ORDERDATE) AS
int) AS month,
        SUM(QUANTITYORDERED) AS ProductsSold
FROM productSales
GROUP BY month, day
ORDER BY month, day
```

```
Databricks visualization. Run in Databricks to view.
```

5. Weekly Sales growth

```
%sql
SELECT DAY_OF_WEEK, CAST(MONTH(ORDERDATE) AS int) AS month,
        CASE
            WHEN DAY_OF_WEEK=1 THEN 'Sunday'
            WHEN DAY_OF_WEEK=2 THEN 'Monday'
            WHEN DAY_OF_WEEK=3 THEN 'Tuesday'
            WHEN DAY_OF_WEEK=4 THEN 'Wednesday'
            WHEN DAY_OF_WEEK=5 THEN 'Thursday'
            WHEN DAY_OF_WEEK=6 THEN 'Friday'
            WHEN DAY_OF_WEEK=7 THEN 'Saturday'
        END as DAY_NAME,
        SUM(Sales) AS TotalSales
FROM productSales
GROUP BY month, DAY_OF_WEEK
ORDER BY month, DAY_OF_WEEK
```

```
Databricks visualization. Run in Databricks to view.
```

6. Top 20 cities with Most Quantity Orders

```
%sql
SELECT CITY, SUM(QUANTITYORDERED) AS TotalQuantityOrdered
FROM productSales
GROUP BY CITY
ORDER BY TotalQuantityOrdered DESC
LIMIT 20
```

```
Databricks visualization. Run in Databricks to view.
```

7. Customer Sales Contribution

```
%sql
SELECT CUSTOMERNAME, ROUND(SUM(SALES), 3) AS sum_sales,
        ROUND(SUM(SALES)*100/(SELECT SUM(SALES) FROM productSales), 4)
AS contribution_percent
FROM productSales
GROUP BY CUSTOMERNAME
ORDER BY sum_sales DESC
LIMIT 10
```

```
Databricks visualization. Run in Databricks to view.
```

8. Seasonal Trends

```
# spark.sql("""
#     SELECT MONTH_ID,
#           CASE
#             WHEN QTR_ID=1 THEN 'Winter'
#             WHEN QTR_ID=2 THEN 'Spring'
#             WHEN QTR_ID=3 THEN 'Summer'
#             WHEN QTR_ID=4 THEN 'Fall'
#           END AS season,
#           ROUND(SUM(SALES), 3) AS TotalSales
#     FROM productSales
#     GROUP BY MONTH_ID, QTR_ID
#     ORDER BY MONTH_ID ASC
# """).show()
```

```
%sql
SELECT MONTH_ID,
      CASE
        WHEN QTR_ID=1 THEN 'Winter'
        WHEN QTR_ID=2 THEN 'Spring'
        WHEN QTR_ID=3 THEN 'Summer'
        WHEN QTR_ID=4 THEN 'Fall'
      END AS season,
      ROUND(SUM(SALES), 3) AS TotalSales,
      CASE
              WHEN MONTH_ID=1 THEN 'January'
              WHEN MONTH_ID=2 THEN 'February'
              WHEN MONTH_ID=3 THEN 'March'
              WHEN MONTH_ID=4 THEN 'April'
              WHEN MONTH_ID=5 THEN 'May'
```

```
              WHEN MONTH_ID=6 THEN 'June'
              WHEN MONTH_ID=7 THEN 'July'
              WHEN MONTH_ID=8 THEN 'August'
              WHEN MONTH_ID=9 THEN 'September'
              WHEN MONTH_ID=10 THEN 'October'
              WHEN MONTH_ID=11 THEN 'November'
              WHEN MONTH_ID=12 THEN 'December'
       END AS month_name
FROM productSales
GROUP BY MONTH_ID, QTR_ID
ORDER BY MONTH_ID ASC
```

```
Databricks visualization. Run in Databricks to view.
```

```
Databricks visualization. Run in Databricks to view.
```

9. Categorize PRODUCTLINE as 'Hot' or 'Cold' based on ORDERQUANTITY in each quarter using CASE WHEN

```
%sql
SELECT PRODUCTLINE, quantity, QTR_ID,
CASE
  WHEN quantity=max THEN 'Hot'
  WHEN quantity=min THEN 'Cold'
  else 'medium'
END AS category
FROM (
    SELECT QTR_ID, PRODUCTLINE,
          SUM(QUANTITYORDERED) AS quantity,
          MAX(SUM(QUANTITYORDERED)) OVER(PARTITION BY QTR_ID) AS max,
          MIN(SUM(QUANTITYORDERED)) OVER(PARTITION BY QTR_ID) AS min
    FROM productSales
    GROUP BY QTR_ID, PRODUCTLINE
) AS _
WHERE quantity = max OR quantity = min
ORDER BY QTR_ID, quantity DESC
```

```
Databricks visualization. Run in Databricks to view.
```

10. Identify products with outlier sales in each quarter

```
%sql
SELECT PRODUCTLINE, SALES, QTR_ID, YEAR_ID,
      PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY SALES) OVER
(PARTITION BY QTR_ID) AS Q1,
      PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY SALES) OVER
(PARTITION BY QTR_ID) AS Q3,
      CASE
          WHEN (SALES > Q3+1.5*(Q3-Q1)) OR (SALES < Q1-1.5*(Q3-Q1))
THEN 1
          ELSE 0
        END AS isOutlier
FROM productSales
```

```
Databricks visualization. Run in Databricks to view.
```

11. Monthly sales Performance by each Territory

```
%sql
SELECT TERRITORY, MONTH_ID,
       ROUND(SUM(SALES - PRICEEACH*QUANTITYORDERED), 3) AS PROFIT,
       ROUND(PROFIT*100/SUM(SALES), 5) AS PROFIT_PERCENTAGE,
       CASE
              WHEN MONTH_ID=1 THEN 'January'
              WHEN MONTH_ID=2 THEN 'February'
              WHEN MONTH_ID=3 THEN 'March'
              WHEN MONTH_ID=4 THEN 'April'
              WHEN MONTH_ID=5 THEN 'May'
              WHEN MONTH_ID=6 THEN 'June'
              WHEN MONTH_ID=7 THEN 'July'
              WHEN MONTH_ID=8 THEN 'August'
              WHEN MONTH_ID=9 THEN 'September'
              WHEN MONTH_ID=10 THEN 'October'
              WHEN MONTH_ID=11 THEN 'November'
              WHEN MONTH_ID=12 THEN 'December'
       END AS month_name
FROM productSales
GROUP BY TERRITORY, MONTH_ID
ORDER BY TERRITORY, MONTH_ID
```

```
Databricks visualization. Run in Databricks to view.
```

```
Databricks visualization. Run in Databricks to view.
```

12. Identify consecutive quarters of sales decline for each product

```
%sql
SELECT *, LAG(Total_Quarterly_Sales, 1, 0) OVER(PARTITION BY
PRODUCTLINE ORDER BY PRODUCTLINE, QTR_ID) AS prev_qtr_sales,
          (Total_Quarterly_Sales-prev_qtr_sales) AS difference,
          ROUND(difference*100/Total_Quarterly_Sales, 3) AS
difference_percentage
FROM (
    SELECT PRODUCTLINE, QTR_ID,
          ROUND(SUM(SALES), 3) AS Total_Quarterly_Sales
    FROM productSales
    GROUP BY PRODUCTLINE, QTR_ID
    ORDER BY PRODUCTLINE, QTR_ID
) AS _
```

```
Databricks visualization. Run in Databricks to view.
```

```
Databricks visualization. Run in Databricks to view.
```

13. median sales for each product line in every quarter

```
%sql
SELECT PRODUCTLINE, QTR_ID, ROUND(MAX(Median), 3) AS Median_Value
FROM (
    SELECT PRODUCTLINE, QTR_ID, SALES,
    PERCENTILE_CONT(0.5) WITHIN GROUP(ORDER BY SALES) OVER (PARTITION
BY PRODUCTLINE, QTR_ID) as Median
    FROM productSales
) AS _
group BY PRODUCTLINE, QTR_ID
ORDER BY PRODUCTLINE, QTR_ID
```

```
Databricks visualization. Run in Databricks to view.
```

14. Percent rank of Products based on Gross sales amount in each quarter

```
%sql
SELECT PRODUCTLINE, QTR_ID, GrossSales, PERCENT_RANK() OVER(PARTITION
BY QTR_ID ORDER BY GrossSales DESC) AS percent_rank
```

```
FROM (
  SELECT PRODUCTLINE, QTR_ID, ROUND(SUM(SALES), 3) AS GrossSales
  FROM productSales
  GROUP BY QTR_ID, PRODUCTLINE
  ORDER BY QTR_ID, PRODUCTLINE
) AS _
```

```
Databricks visualization. Run in Databricks to view.
```

15. categorize product category sales performance as 'High', 'Medium' and 'Low' based on percentile

```
%sql
-- SELECT PRODUCTLINE, SALES,
--        NTILE(3) OVER (PARTITION BY PRODUCTLINE ORDER BY SALES) AS
ntile,
--        CASE
--            WHEN ntile=1 THEN 'Low'
--            WHEN ntile=2 THEN 'Medium'
--            WHEN ntile=3 THEN 'High'
--        END AS sales_performance_category
-- FROM productSales

SELECT PRODUCTLINE, total_sales_per_prodline, NTILE(3) OVER (ORDER BY
total_sales_per_prodline) AS ntile,
        CASE
            WHEN ntile=1 THEN 'Low'
            WHEN ntile=2 THEN 'Medium'
            WHEN ntile=3 THEN 'High'
        END AS sales_performance_category
FROM (
    SELECT PRODUCTLINE, ROUND(SUM(SALES), 3) AS
total_sales_per_prodline
    FROM productSales
    GROUP BY PRODUCTLINE
) AS _;
```

```
Databricks visualization. Run in Databricks to view.
```

16. Rank 5 customers by total purchase in every quarter of each year

```
%sql
SELECT CUSTOMERNAME, YEAR_ID, QTR_ID, Total_Purchase, rank,
       CONCAT(YEAR_ID, ',', QTR_ID) AS year_and_quarter
FROM (
  SELECT CUSTOMERNAME, YEAR_ID, QTR_ID, ROUND(SUM(QUANTITYORDERED *
PRICEEACH), 3) AS Total_Purchase,
        DENSE_RANK() OVER(PARTITION BY YEAR_ID, QTR_ID ORDER BY
SUM(QUANTITYORDERED*PRICEEACH) DESC) AS rank
  FROM productSales
  GROUP BY CUSTOMERNAME, YEAR_ID, QTR_ID
  ORDER BY YEAR_ID, QTR_ID
) AS _
WHERE rank <= 5
```

```
Databricks visualization. Run in Databricks to view.
```

```
Databricks visualization. Run in Databricks to view.
```

17. Identify and Present the Trends of Sales Growth by Month. [bar graph, in percentage form]

```
%sql
SELECT MONTH_ID, ROUND(SUM(SALES), 3) AS MonthlySales,
       LAG(ROUND(SUM(SALES), 3), 1, 0) OVER(ORDER BY MONTH_ID) AS
Prev_MonthlySales,
       ROUND((MonthlySales-Prev_MonthlySales)*100/MonthlySales, 3) AS
PercentGrowth,
       CASE
              WHEN MONTH_ID=1 THEN 'January'
              WHEN MONTH_ID=2 THEN 'February'
              WHEN MONTH_ID=3 THEN 'March'
              WHEN MONTH_ID=4 THEN 'April'
              WHEN MONTH_ID=5 THEN 'May'
              WHEN MONTH_ID=6 THEN 'June'
              WHEN MONTH_ID=7 THEN 'July'
              WHEN MONTH_ID=8 THEN 'August'
              WHEN MONTH_ID=9 THEN 'September'
              WHEN MONTH_ID=10 THEN 'October'
              WHEN MONTH_ID=11 THEN 'November'
              WHEN MONTH_ID=12 THEN 'December'
       END AS month_name
```

```
FROM productSales
GROUP BY MONTH_ID
ORDER BY MONTH_ID

Databricks visualization. Run in Databricks to view.
```