

```
pip install pyspark
```

```
Requirement already satisfied: pyspark in /usr/local/lib/python3.10/dist-packages (3.5.1)
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
```

```
import pyspark
```

```
import pandas as pd
pd.read_csv('Sales_data.csv')
```



	Sr.No.	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	Hour
0	0	295665	Macbook Pro Laptop	1	1700.00	30-12-2019 00:01	136 Church St, New York City, NY 10001	12	1700.00	New York City	0
1	1	295666	LG Washing Machine	1	600.00	29-12-2019 07:03	562 2nd St, New York City, NY 10001	12	600.00	New York City	7
2	2	295667	USB-C Charging Cable	1	11.95	12-12-2019 18:21	277 Main St, New York City, NY 10001	12	11.95	New York City	18
3	3	295668	27in FHD Monitor	1	149.99	22-12-2019 15:13	410 6th St, San Francisco, CA 94016	12	149.99	San Francisco	15
4	4	295669	USB-C Charging Cable	1	11.95	18-12-2019 12:38	43 Hill St, Atlanta, GA 30301	12	11.95	Atlanta	12
...
185945	13617	222905	AAA Batteries (4-pack)	1	2.99	07-06-2019 19:02	795 Pine St, Boston, MA 02215	6	2.99	Boston	19
185946	13618	222906	27in FHD Monitor	1	149.99	01-06-2019 19:29	495 North St, New York City, NY 10001	6	149.99	New York City	19

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

```
from pyspark.sql.functions import col, sum
from pyspark.sql.types import FloatType, IntegerType
```

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

```
spark
```

SparkSession - in-memory

SparkContext

[Spark UI](#)

Version

v3.5.1

Master

local[*]

AppName

Practise

```
df_pyspark=spark.read.csv('Sales_data.csv')
```

```
df_pyspark=spark.read.option('header','true').csv('Sales_data.csv')
```

```
type(df_pyspark)
```

pyspark.sql.dataframe.DataFrame

def __init__(jdf: JavaObject, sql_ctx: Union['SQLContext', 'SparkSession'])

A distributed collection of data grouped into named columns.

.. versionadded:: 1.3.0

.. versionchanged:: 3.4.0

```
df_pyspark.printSchema()
```

```
root
|-- Sr.No.: string (nullable = true)
|-- Order ID: string (nullable = true)
|-- Product: string (nullable = true)
|-- Quantity Ordered: string (nullable = true)
|-- Price Each: string (nullable = true)
|-- Order Date: string (nullable = true)
|-- Purchase Address: string (nullable = true)
```

```
-- Month: string (nullable = true)
|-- Sales: string (nullable = true)
|-- City: string (nullable = true)
|-- Hour: string (nullable = true)

df_pyspark.select(['Product','Order ID'])

DataFrame[Product: string, Order ID: string]

df_pyspark.select(['Product','Order ID']).show()
```

Product	Order ID
Macbook Pro Laptop	295665
LG Washing Machine	295666
USB-C Charging Cable	295667
27in FHD Monitor	295668
USB-C Charging Cable	295669
AA Batteries (4-p...	295670
USB-C Charging Cable	295671
USB-C Charging Cable	295672
Bose SoundSport H...	295673
AAA Batteries (4-...	295674
USB-C Charging Cable	295675
ThinkPad Laptop	295676
AA Batteries (4-p...	295677
AAA Batteries (4-...	295678
USB-C Charging Cable	295679
Lightning Chargin...	295680
Google Phone	295681
USB-C Charging Cable	295681
Bose SoundSport H...	295681
Wired Headphones	295681

only showing top 20 rows

```
df_pyspark['Product']

Column<'Product'>
```

```
df_pyspark.dtypes

[('Sr.No.', 'string'),
 ('Order ID', 'string'),
 ('Product', 'string'),
 ('Quantity Ordered', 'string'),
 ('Price Each', 'string'),
 ('Order Date', 'string'),
 ('Purchase Address', 'string'),
 ('Month', 'string'),
 ('Sales', 'string'),
 ('City', 'string'),
 ('Hour', 'string')]
```

```
### Adding columns in data frame
df_pyspark=df_pyspark.withColumn('Sales After 1 year',df_pyspark['Sales']*2)
```

```
df_pyspark.show()
```

Sr.No.	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City
0	295665	Macbook Pro Laptop	1	1700	30-12-2019 00:01	136 Church St, Ne...	12	1700	New York City
1	295666	LG Washing Machine	1	600	29-12-2019 07:03	562 2nd St, New Y...	12	600	New York City
2	295667	USB-C Charging Cable	1	11.95	12-12-2019 18:21	277 Main St, New ...	12	11.95	New York City
3	295668	27in FHD Monitor	1	149.99	22-12-2019 15:13	410 6th St, San F...	12	149.99	San Francisco
4	295669	USB-C Charging Cable	1	11.95	18-12-2019 12:38	43 Hill St, Atl...	12	11.95	Atlanta
5	295670	AA Batteries (4-p...	1	3.84	31-12-2019 22:58	200 Jefferson St,...	12	3.84	New York City
6	295671	USB-C Charging Cable	1	11.95	16-12-2019 15:10	928 12th St, Port...	12	11.95	Portland
7	295672	USB-C Charging Cable	2	11.95	13-12-2019 09:29	813 Hickory St, D...	12	23.9	Dallas
8	295673	Bose SoundSport H...	1	99.99	15-12-2019 23:26	718 Wilson St, Da...	12	99.99	Dallas
9	295674	AAA Batteries (4-...	4	2.99	28-12-2019 11:51	77 7th St, Dallas...	12	11.96	Dallas
10	295675	USB-C Charging Cable	2	11.95	13-12-2019 13:52	594 1st St, San F...	12	23.9	San Francisco
11	295676	ThinkPad Laptop	1	999.99	28-12-2019 17:19	410 Lincoln St, L...	12	999.99	Los Angeles
12	295677	AA Batteries (4-p...	2	3.84	20-12-2019 19:19	866 Pine St, Bost...	12	7.68	Boston
13	295678	AAA Batteries (4-...	2	2.99	06-12-2019 09:38	187 Lincoln St, D...	12	5.98	Dallas
14	295679	USB-C Charging Cable	1	11.95	25-12-2019 09:39	902 2nd St, Dalla...	12	11.95	Dallas
15	295680	Lightning Chargin...	1	14.95	01-12-2019 14:30	338 Main St, Aust...	12	14.95	Austin
16	295681	Google Phone	1	600	25-12-2019 12:37	79 Elm St, Boston...	12	600	Boston
17	295681	USB-C Charging Cable	1	11.95	25-12-2019 12:37	79 Elm St, Boston...	12	11.95	Boston
18	295681	Bose SoundSport H...	1	99.99	25-12-2019 12:37	79 Elm St, Boston...	12	99.99	Boston
19	295681	Wired Headphones	1	11.99	25-12-2019 12:37	79 Elm St, Boston...	12	11.99	Boston

only showing top 20 rows

```
### Drop the columns
df_pyspark=df_pyspark.drop('Sales After 1 year')
```

```
df_pyspark.show()
```

Sr.No.	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Sales	City	H
0	295665	Macbook Pro Laptop	1	1700	30-12-2019 00:01	136 Church St, Ne...	12	1700	New York City	
1	295666	LG Washing Machine	1	600	29-12-2019 07:03	562 2nd St, New Y...	12	600	New York City	
2	295667	USB-C Charging Cable	1	11.95	12-12-2019 18:21	277 Main St, New ...	12	11.95	New York City	
3	295668	27in FHD Monitor	1	149.99	22-12-2019 15:13	410 6th St, San F...	12	149.99	San Francisco	
4	295669	USB-C Charging Cable	1	11.95	18-12-2019 12:38	43 Hill St, Atlan...	12	11.95	Atlanta	
5	295670	AA Batteries (4-p...	1	3.84	31-12-2019 22:58	200 Jefferson St,...	12	3.84	New York City	
6	295671	USB-C Charging Cable	1	11.95	16-12-2019 15:10	928 12th St, Port...	12	11.95	Portland	
7	295672	USB-C Charging Cable	2	11.95	13-12-2019 09:29	813 Hickory St, D...	12	23.9	Dallas	
8	295673	Bose SoundSport H...	1	99.99	15-12-2019 23:26	718 Wilson St, Da...	12	99.99	Dallas	
9	295674	AAA Batteries (4-...	4	2.99	28-12-2019 11:51	77 7th St, Dallas...	12	11.96	Dallas	
10	295675	USB-C Charging Cable	2	11.95	13-12-2019 13:52	594 1st St, San F...	12	23.9	San Francisco	
11	295676	ThinkPad Laptop	1	999.99	28-12-2019 17:19	410 Lincoln St, L...	12	999.99	Los Angeles	
12	295677	AA Batteries (4-p...	2	3.84	20-12-2019 19:19	866 Pine St, Bost...	12	7.68	Boston	
13	295678	AAA Batteries (4-...	2	2.99	06-12-2019 09:38	187 Lincoln St, D...	12	5.98	Dallas	
14	295679	USB-C Charging Cable	1	11.95	25-12-2019 09:39	902 2nd St, Dalla...	12	11.95	Dallas	
15	295680	Lightning Chargin...	1	14.95	01-12-2019 14:30	338 Main St, Aust...	12	14.95	Austin	
16	295681	Google Phone	1	600	25-12-2019 12:37	79 Elm St, Boston...	12	600	Boston	
17	295681	USB-C Charging Cable	1	11.95	25-12-2019 12:37	79 Elm St, Boston...	12	11.95	Boston	
18	295681	Bose SoundSport H...	1	99.99	25-12-2019 12:37	79 Elm St, Boston...	12	99.99	Boston	
19	295681	Wired Headphones	1	11.99	25-12-2019 12:37	79 Elm St, Boston...	12	11.99	Boston	

only showing top 20 rows



```
df_pyspark = df_pyspark.withColumnRenamed("Sr.No.", "SrNo")

# Data cleaning
# Convert columns to appropriate data types
df_pyspark= df_pyspark.withColumn("Quantity Ordered", df_pyspark["Quantity Ordered"].cast(IntegerType()))
df_pyspark = df_pyspark.withColumn("Price Each", df_pyspark["Price Each"].cast(FloatType()))
df_pyspark = df_pyspark.withColumn("Sales", df_pyspark["Sales"].cast(FloatType()))

# Handling missing values
df_pyspark = df_pyspark.dropna()

# Removing duplicates
df_pyspark = df_pyspark.dropDuplicates()

# Calculate total sales amount for each product
product_sales = df_pyspark.groupBy("Product").agg(sum("Sales").alias("TotalSales"))

# Output the results to a new CSV file with overwrite mode
product_sales.coalesce(1).write.mode("overwrite").csv("output_path", header=True)

# Stop SparkSession
spark.stop()
```