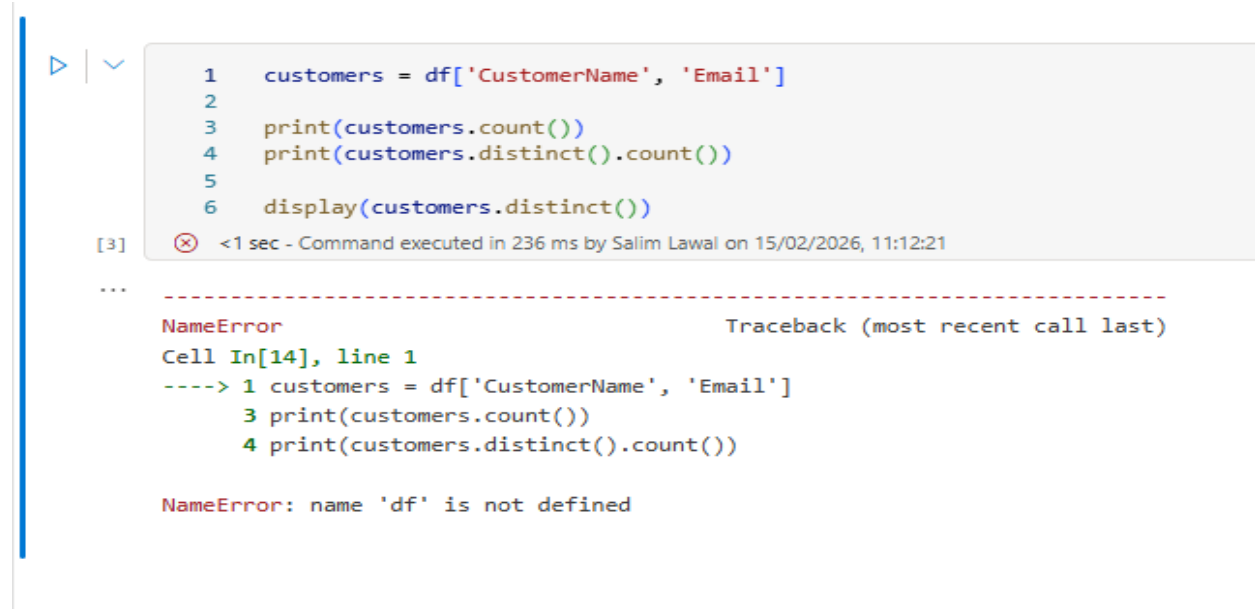


## Exercise Error: Filtering and Transforming Dataframe

When trying to transform dataframe into a new dataframe, I encountered this error.



The screenshot shows a Jupyter Notebook interface. A code cell contains the following Python code:

```
1 customers = df['CustomerName', 'Email']
2
3 print(customers.count())
4 print(customers.distinct().count())
5
6 display(customers.distinct())
```

Below the code, the execution status is shown as [3] with a red 'x' icon and the text: <1 sec - Command executed in 236 ms by Salim Lawal on 15/02/2026, 11:12:21.

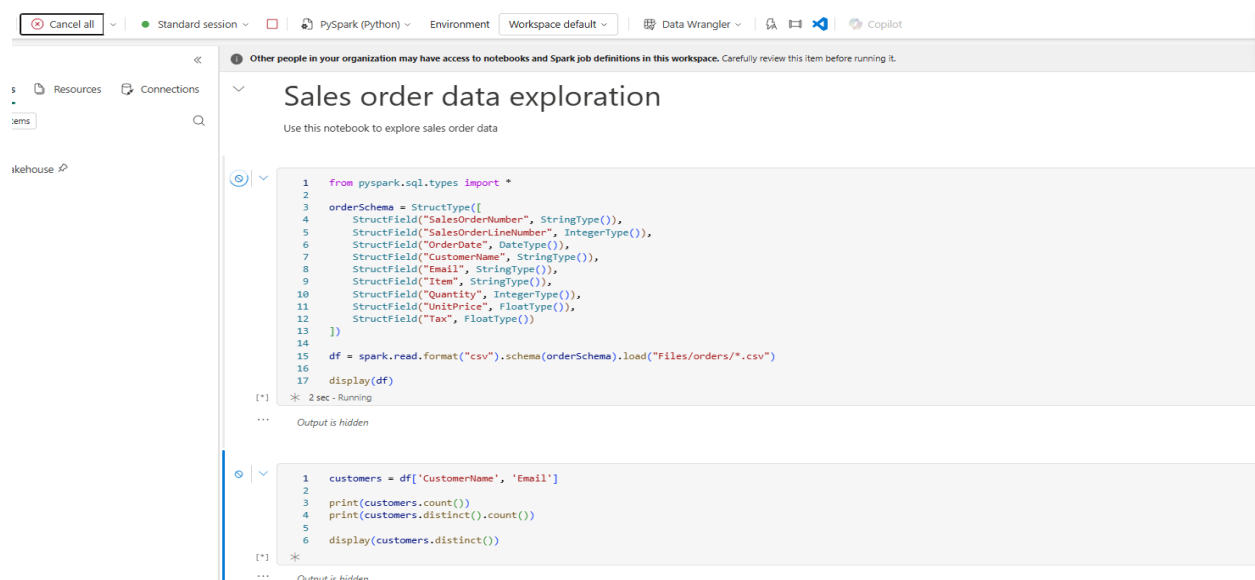
Below the code cell, the error message is displayed:

```
-----
NameError                                Traceback (most recent call last)
Cell In[14], line 1
----> 1 customers = df['CustomerName', 'Email']
      3 print(customers.count())
      4 print(customers.distinct().count())

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

Error occurred because I was running this code alone instead of running all the code including the code for the first dataframe.

Hence the NameError: name 'df' is not defined



The screenshot shows a Databricks workspace interface. The top bar includes a 'Cancel all' button, a 'Standard session' indicator, and a 'PySpark (Python)' environment selection. The workspace title is 'Sales order data exploration'.

The first code cell contains the following Python code:

```
1 from pyspark.sql.types import *
2
3 orderSchema = StructType([
4     StructField("SalesOrderNumber", StringType()),
5     StructField("SalesOrderLineNumber", IntegerType()),
6     StructField("OrderDate", DateType()),
7     StructField("CustomerName", StringType()),
8     StructField("Email", StringType()),
9     StructField("Item", StringType()),
10    StructField("Quantity", IntegerType()),
11    StructField("UnitPrice", FloatType()),
12    StructField("Tax", FloatType())
13 ])
14
15 df = spark.read.format("csv").schema(orderSchema).load("Files/orders/*.csv")
16
17 display(df)
```

The second code cell contains the following Python code:

```
1 customers = df['CustomerName', 'Email']
2
3 print(customers.count())
4 print(customers.distinct().count())
5
6 display(customers.distinct())
```

Ran all the code and output created a new dataframe with applied filter