databricks Basic Data Cleaning with PySpark

(https://databricks.com)

Finding the datasets from local data sets

```
%fs

ls
```

Table

	path	name	size	modificationTime
1	dbfs:/FileStore/	FileStore/	0	0
2	dbfs:/databricks-datasets/	databricks-datasets/	0	0
3	dbfs:/databricks-results/	databricks-results/	0	0
4	dbfs:/user/	user/	0	0

4 rows

```
%fs

ls dbfs:/databricks-datasets/online_retail/data-001/
```

Table

	path	name	size	modificationTim
1	dbfs:/databricks-datasets/online_retail/data-001/data.csv	data.csv	5357240	1466107812000

1 row

Loading the datasets

```
data = spark.read.csv("dbfs:/databricks-datasets/online_retail/data-001/data.csv", header=True, inferSchema=True)
```

Create duplicate copy of original datasets in df

```
df = data.alias('copy')


df.show()  # show datasets
```

```
+---------+---------+--------------------+--------+------------+---------+----------+---------
-----+
```

```
|InvoiceNo|StockCode|         Description|Quantity| InvoiceDate|UnitPrice|CustomerID|       Co
untry|
+---------+---------+--------------------+--------+------------+---------+----------+---------
-----+
|   536365|   85123A|WHITE HANGING HEA...|       6|12/1/10 8:26|     2.55|     17850|United Ki
ngdom|
|   536365|    71053| WHITE METAL LANTERN|       6|12/1/10 8:26|     3.39|     17850|United Ki
ngdom|
|   536365|   84406B|CREAM CUPID HEART...|       8|12/1/10 8:26|     2.75|     17850|United Ki
ngdom|
|   536365|   84029G|KNITTED UNION FLA...|       6|12/1/10 8:26|     3.39|     17850|United Ki
ngdom|
|   536365|   84029E|RED WOOLLY HOTTIE...|       6|12/1/10 8:26|     3.39|     17850|United Ki
ngdom|
|   536365|    22752|SET 7 BABUSHKA NE...|       2|12/1/10 8:26|     7.65|     17850|United Ki
ngdom|
|   536365|    21730|GLASS STAR FROSTE...|       6|12/1/10 8:26|     4.25|     17850|United Ki
ngdom|
|   536366|    22633|HAND WARMER UNION...|       6|12/1/10 8:28|     1.85|     17850|United Ki
```

```
df.count()  # check the dataset size
```

```
Out[128]: 65499
```

```
df.columns   # check the columns
```

```
Out[129]: ['InvoiceNo',
 'StockCode',
 'Description',
 'Quantity',
 'InvoiceDate',
 'UnitPrice',
 'CustomerID',
 'Country']
```

```
df.printSchema()    # check the data types of each columns
```

```
root
 |-- InvoiceNo: string (nullable = true)
 |-- StockCode: string (nullable = true)
 |-- Description: string (nullable = true)
 |-- Quantity: integer (nullable = true)
 |-- InvoiceDate: string (nullable = true)
 |-- UnitPrice: double (nullable = true)
 |-- CustomerID: integer (nullable = true)
 |-- Country: string (nullable = true)
```

```
from pyspark.sql import functions as f   # import functions class
```

```
null_values = df.filter(f.col("InvoiceNo").isNull())
null_values.show()

# filter the null values in "InvoiceNo" columns
```

```
+---------+---------+-----------+--------+-----------+---------+----------+-------+
|InvoiceNo|StockCode|Description|Quantity|InvoiceDate|UnitPrice|CustomerID|Country|
+---------+---------+-----------+--------+-----------+---------+----------+-------+
```

```
+---------+---------+-----------+--------+-----------+---------+----------+-------+
```

```
df.select([f.isnull(c).alias(c) for c in df.columns]).groupBy(df.columns).count().show()

# check the null values in each columns
```

```
+---------+---------+-----------+--------+-----------+---------+----------+-------+-----+
|InvoiceNo|StockCode|Description|Quantity|InvoiceDate|UnitPrice|CustomerID|Country|count|
+---------+---------+-----------+--------+-----------+---------+----------+-------+-----+
|    false|    false|      false|   false|      false|    false|     false|  false|40218|
|    false|    false|      false|   false|      false|    false|      true|  false|25115|
|    false|    false|       true|   false|      false|    false|      true|  false|  166|
+---------+---------+-----------+--------+-----------+---------+----------+-------+-----+
```

```
# Check for null values with defferent way
df.select([f.count(f.when(f.isnull(c), c)).alias(c) for c in df.columns]).show()
```

```
+---------+---------+-----------+--------+-----------+---------+----------+-------+
|InvoiceNo|StockCode|Description|Quantity|InvoiceDate|UnitPrice|CustomerID|Country|
+---------+---------+-----------+--------+-----------+---------+----------+-------+
|        0|        0|        166|       0|          0|        0|     25281|      0|
+---------+---------+-----------+--------+-----------+---------+----------+-------+
```

```
null_values = df.filter(F.col("CustomerID").isNull())  # filter the CustomerID columns as per
null values
null_values.show()
```

```
+---------+---------+--------------------+--------+-------------+---------+----------+--------
------+
|InvoiceNo|StockCode|         Description|Quantity|  InvoiceDate|UnitPrice|CustomerID|       C
ountry|
+---------+---------+--------------------+--------+-------------+---------+----------+--------
------+
|   536414|    22139|                null|      56|12/1/10 11:52|      0.0|      null|United K
ingdom|
|   536544|    21773|DECORATIVE ROSE B...|       1|12/1/10 14:32|     2.51|      null|United K
ingdom|
|   536544|    21774|DECORATIVE CATS B...|       2|12/1/10 14:32|     2.51|      null|United K
ingdom|
|   536544|    21786|  POLKADOT RAIN HAT |       4|12/1/10 14:32|     0.85|      null|United K
ingdom|
|   536544|    21787|RAIN PONCHO RETRO...|       2|12/1/10 14:32|     1.66|      null|United K
ingdom|
|   536544|    21790|  VINTAGE SNAP CARDS|       9|12/1/10 14:32|     1.66|      null|United K
ingdom|
|   536544|    21791|VINTAGE HEADS AND...|       2|12/1/10 14:32|     2.51|      null|United K
ingdom|
|   536544|    21801|CHRISTMAS TREE DE...|      10|12/1/10 14:32|     0.43|      null|United K
```

```
df.select('CustomerID').show()  # show the CustomerID columns
```

```
+----------+
|CustomerID|
+----------+
|     17850|
|     17850|
```

```
|     17850|
|     17850|
|     17850|
|     17850|
|     17850|
|     17850|
|     17850|
|     13047|
|     13047|
|     13047|
|     13047|
|     13047|
|     13047|
|     13047|
|     13047|
```

```
CustomerIDUpdate = df.select('CustomerID').fillna(0)
CustomerIDUpdate.show()    # replace  the null values with 0 in customerid
```

```
+----------+
|CustomerID|
+----------+
|     17850|
|     17850|
|     17850|
|     17850|
|     17850|
|     17850|
|     17850|
|     17850|
|     17850|
|     13047|
|     13047|
|     13047|
|     13047|
|     13047|
|     13047|
|     13047|
|     13047|
```

```
df = df.withColumn("CustomerID", F.coalesce(df["CustomerID"], F.lit(0)))    # update the
CustomerID with 0 where is the null values
```

```
null_values = df.filter(F.col("CustomerID").isNull()) # filter the CustomerID where is null
values
null_values.show()
```

```
+---------+---------+-----------+--------+-----------+---------+----------+-------+
|InvoiceNo|StockCode|Description|Quantity|InvoiceDate|UnitPrice|CustomerID|Country|
+---------+---------+-----------+--------+-----------+---------+----------+-------+
+---------+---------+-----------+--------+-----------+---------+----------+-------+
```

```
df = df.dropna() # now i drop the rest null columns as per description columns
```

```
# Check for null values
df.select([f.count(f.when(f.isnull(c), c)).alias(c) for c in df.columns]).show()  # after
cleaning teh datasets check the null values in each columns
```

```
+---------+---------+-----------+--------+-----------+---------+----------+-------+
|InvoiceNo|StockCode|Description|Quantity|InvoiceDate|UnitPrice|CustomerID|Country|
+---------+---------+-----------+--------+-----------+---------+----------+-------+
|        0|        0|          0|       0|          0|        0|         0|      0|
+---------+---------+-----------+--------+-----------+---------+----------+-------+
```

```
df.show() # final dataset show after cleaning
```

```
+---------+---------+--------------------+--------+------------+---------+----------+---------
-----+
|InvoiceNo|StockCode|         Description|Quantity| InvoiceDate|UnitPrice|CustomerID|       Co
untry|
+---------+---------+--------------------+--------+------------+---------+----------+---------
-----+
|   536365|   85123A|WHITE HANGING HEA...|       6|12/1/10 8:26|     2.55|     17850|United Ki
ngdom|
|   536365|    71053| WHITE METAL LANTERN|       6|12/1/10 8:26|     3.39|     17850|United Ki
ngdom|
|   536365|   84406B|CREAM CUPID HEART...|       8|12/1/10 8:26|     2.75|     17850|United Ki
ngdom|
|   536365|   84029G|KNITTED UNION FLA...|       6|12/1/10 8:26|     3.39|     17850|United Ki
ngdom|
|   536365|   84029E|RED WOOLLY HOTTIE...|       6|12/1/10 8:26|     3.39|     17850|United Ki
ngdom|
|   536365|    22752|SET 7 BABUSHKA NE...|       2|12/1/10 8:26|     7.65|     17850|United Ki
ngdom|
|   536365|    21730|GLASS STAR FROSTE...|       6|12/1/10 8:26|     4.25|     17850|United Ki
ngdom|
|   536366|    22633|HAND WARMER UNION...|       6|12/1/10 8:28|     1.85|     17850|United Ki
```

```
df.printSchema()
```

```
root
 |-- InvoiceNo: string (nullable = true)
 |-- StockCode: string (nullable = true)
 |-- Description: string (nullable = true)
 |-- Quantity: integer (nullable = true)
 |-- InvoiceDate: string (nullable = true)
 |-- UnitPrice: double (nullable = true)
 |-- CustomerID: integer (nullable = false)
 |-- Country: string (nullable = true)
```

```
from pyspark.sql import types as t
```

```
df = df.withColumn("InvoiceDate", df["InvoiceDate"].cast(t.TimestampType()))
```

```
df.printSchema()
```

```
root
 |-- InvoiceNo: string (nullable = true)
```

```
 |-- StockCode: string (nullable = true)
 |-- Description: string (nullable = true)
 |-- Quantity: integer (nullable = true)
 |-- InvoiceDate: timestamp (nullable = true)
 |-- UnitPrice: double (nullable = true)
 |-- CustomerID: integer (nullable = false)
 |-- Country: string (nullable = true)
```

```
import matplotlib.pyplot as plt
```

```
df1 = df.groupBy('Country').agg(F.sum('UnitPrice').alias('TotalUnitPrice'))
df1.show()
```

```
+--------------+------------------+
|       Country|    TotalUnitPrice|
+--------------+------------------+
|        Sweden|144.77000000000004|
|       Germany| 3569.779999999995|
|        France| 3366.180000000005|
|       Belgium| 408.7099999999998|
|       Finland|              49.6|
|         Italy|418.64999999999986|
|          EIRE|2418.5600000000036|
|     Lithuania| 99.44000000000001|
|        Norway|238.82999999999996|
|         Spain| 2367.680000000003|
|       Denmark|             92.97|
|       Iceland| 89.59000000000002|
|        Israel|              68.0|
|Channel Islands|160.64000000000001|
|        Cyprus| 490.6899999999998|
|   Switzerland|            464.05|
|         Japan|            131.35|
|        Poland|115.03000000000002|
```

```
df = df.withColumn('TotalSales',df['UnitPrice'] * df['Quantity'])
```

```
df.show() # check the datatype of the datasets
```

```
+---------+---------+--------------------+--------+-----------+---------+----------+----------
----+------------------+
|InvoiceNo|StockCode|         Description|Quantity|InvoiceDate|UnitPrice|CustomerID|       Cou
ntry|        TotalSales|
+---------+---------+--------------------+--------+-----------+---------+----------+----------
----+------------------+
|   536365|   85123A|WHITE HANGING HEA...|       6|       null|     2.55|     17850|United Kin
gdom|15.299999999999999|
|   536365|    71053| WHITE METAL LANTERN|       6|       null|     3.39|     17850|United Kin
gdom|             20.34|
|   536365|   84406B|CREAM CUPID HEART...|       8|       null|     2.75|     17850|United Kin
gdom|              22.0|
|   536365|   84029G|KNITTED UNION FLA...|       6|       null|     3.39|     17850|United Kin
gdom|             20.34|
|   536365|   84029E|RED WOOLLY HOTTIE...|       6|       null|     3.39|     17850|United Kin
gdom|             20.34|
|   536365|    22752|SET 7 BABUSHKA NE...|       2|       null|     7.65|     17850|United Kin
gdom|              15.3|
|   536365|    21730|GLASS STAR FROSTE...|       6|       null|     4.25|     17850|United Kin
gdom|              25.5|
|   536366|    22633|HAND WARMER UNION...|       6|       null|     1.85|     17850|United Kin
```

```
df1 = df.groupBy('Country').agg(F.sum('TotalSales').alias('TotalSales'))   # create the total
saels columns
df1.show()
```

```
+--------------+------------------+
|       Country|        TotalSales|
+--------------+------------------+
|        Sweden| 3153.859999999999|
|       Germany|22237.810000000005|
|        France|21773.329999999998|
|       Belgium|2640.5199999999995|
|       Finland| 892.8000000000001|
|         Italy|2395.5099999999993|
|          EIRE|29037.420000000002|
|     Lithuania|           1661.06|
|        Norway|3787.1199999999994|
|         Spain| 8864.819999999998|
|       Denmark|1281.5000000000002|
|       Iceland|            711.79|
|        Israel|152.40000000000003|
|Channel Islands|            363.53|
|        Cyprus|2138.3199999999997|
|   Switzerland| 4909.549999999999|
|         Japan| 7595.270000000001|
|        Poland|            861.38|
```