04_Groupby_agg

March 28, 2020

0.1 Load data

[1]:
```
from pyspark.sql import SparkSession
```

[2]:
```
spark = SparkSession.builder.appName("GroupByAgg").getOrCreate()
```

[3]:
```
path = "Python-and-Spark-for-Big-Data-master/Spark_DataFrames/sales_info.csv"
```

[4]:
```
df = spark.read.csv(path, inferSchema=True, header=True)
```

[5]:
```
df.show()
```

```
+-------+-------+-----+
|Company| Person|Sales|
+-------+-------+-----+
|   GOOG|    Sam|200.0|
|   GOOG|Charlie|120.0|
|   GOOG|  Frank|340.0|
|   MSFT|   Tina|600.0|
|   MSFT|    Amy|124.0|
|   MSFT|Vanessa|243.0|
|     FB|   Carl|870.0|
|     FB|  Sarah|350.0|
|   APPL|   John|250.0|
|   APPL|  Linda|130.0|
|   APPL|   Mike|750.0|
|   APPL|  Chris|350.0|
+-------+-------+-----+
```

[6]:
```
df.printSchema()
```

```
root
 |-- Company: string (nullable = true)
 |-- Person: string (nullable = true)
 |-- Sales: double (nullable = true)
```

0.2 Groupby and aggregation

[7]:
```
# Just groupby return a GroupedData object, not a data frame
type(df.groupby("company"))
```

[7]:
```
pyspark.sql.group.GroupedData
```

[8]:
```
df.groupby(["company", "person"])
```

[8]:
```
<pyspark.sql.group.GroupedData at 0x7fa84862ffd0>
```

[9]:
```
# groupby and agg will return data frame
df.groupby("company").mean().show()
```

```
+-------+-----------------+
|company|       avg(Sales)|
+-------+-----------------+
|   APPL|            370.0|
|   GOOG|            220.0|
|     FB|            610.0|
|   MSFT|322.3333333333333|
+-------+-----------------+
```

[10]:
```
df.groupby("company").max().show()
```

```
+-------+----------+
|company|max(Sales)|
+-------+----------+
|   APPL|     750.0|
|   GOOG|     340.0|
|     FB|     870.0|
|   MSFT|     600.0|
+-------+----------+
```

[11]:
```
df.groupby("company").count().show()    # how many rows
```

```
+-------+-----+
|company|count|
+-------+-----+
|   APPL|    4|
|   GOOG|    3|
|     FB|    2|
|   MSFT|    3|
+-------+-----+
```

[12]:
```
# agg() can be used without groupby
# df.agg({"Sales":"mean"}).show()
# df.agg({"Sales":"max"}).show()
df.agg({"Sales":"sum"}).show()
```

```
+----------+
|sum(Sales)|
+----------+
|    4327.0|
+----------+
```

[13]:
```
# combine groupby and agg
df.groupby("company").agg({"Sales":"mean"}).show()
```

```
+-------+-----------------+
|company|       avg(Sales)|
+-------+-----------------+
|   APPL|            370.0|
|   GOOG|            220.0|
|     FB|            610.0|
|   MSFT|322.3333333333333|
+-------+-----------------+
```

0.3 Using functions from pyspark with select()

[14]:
```
from pyspark.sql.functions import countDistinct, avg, stddev
```

[15]:
```
df.select(countDistinct("company")).show()
```

```
+-----------------------+
|count(DISTINCT company)|
+-----------------------+
|                      4|
+-----------------------+
```

[16]:
```
df.select(avg("Sales")).show()
```

```
+-----------------+
|       avg(Sales)|
+-----------------+
|360.5833333333333|
+-----------------+
```

[17]:
```
# using alias to rename the column inside select
df.select(avg("Sales").alias("Averaged sales")).show()
```

```
+-----------------+
|   Averaged sales|
+-----------------+
|360.5833333333333|
+-----------------+
```

0.4 Format number while showing data

[18]:
```
from pyspark.sql.functions import format_number
```

[19]:
```
sales_std = df.select(stddev("Sales"))
```

[20]:
```
sales_std.show()
```

```
+------------------+
|stddev_samp(Sales)|
+------------------+
|250.08742410799007|
+------------------+
```

[21]:
```
sales_std.select(format_number("stddev_samp(Sales)", 2).alias("std")).show()
```

```
+------+
|   std|
+------+
|250.09|
+------+
```

0.5 Ordering

[22]:
```
# ordering ascending is easy
df.orderBy("Sales").show()
```

```
+-------+-------+-----+
|Company| Person|Sales|
+-------+-------+-----+
|   GOOG|Charlie|120.0|
|   MSFT|    Amy|124.0|
|   APPL|  Linda|130.0|
|   GOOG|    Sam|200.0|
|   MSFT|Vanessa|243.0|
```

```
|   APPL|   John|250.0|
|   GOOG|  Frank|340.0|
|     FB|  Sarah|350.0|
|   APPL|  Chris|350.0|
|   MSFT|   Tina|600.0|
|   APPL|   Mike|750.0|
|     FB|   Carl|870.0|
+-------+-------+-----+
```

[23]:
```
# for descending
df.orderBy("Sales", ascending=False).show()
```

```
+-------+-------+-----+
|Company| Person|Sales|
+-------+-------+-----+
|     FB|   Carl|870.0|
|   APPL|   Mike|750.0|
|   MSFT|   Tina|600.0|
|     FB|  Sarah|350.0|
|   APPL|  Chris|350.0|
|   GOOG|  Frank|340.0|
|   APPL|   John|250.0|
|   MSFT|Vanessa|243.0|
|   GOOG|    Sam|200.0|
|   APPL|  Linda|130.0|
|   MSFT|    Amy|124.0|
|   GOOG|Charlie|120.0|
+-------+-------+-----+
```

[24]:
```
# we can also use the desc() method
df.orderBy(df["Sales"].desc()).show()
```

```
+-------+-------+-----+
|Company| Person|Sales|
+-------+-------+-----+
|     FB|   Carl|870.0|
|   APPL|   Mike|750.0|
|   MSFT|   Tina|600.0|
|     FB|  Sarah|350.0|
|   APPL|  Chris|350.0|
|   GOOG|  Frank|340.0|
|   APPL|   John|250.0|
|   MSFT|Vanessa|243.0|
|   GOOG|    Sam|200.0|
|   APPL|  Linda|130.0|
```

```
|   MSFT|    Amy|124.0|
|   GOOG|Charlie|120.0|
+-------+-------+-----+
```