Transformations - GroupBy

GroupBy

df.groupBy(cols) returns a funny thing called:

<class 'pyspark.sql.group.GroupedData'>

```
df1 = df.groupBy("dept")
print(type(df1))
```

```
<class 'pyspark.sql.group.GroupedData'>
```

Notebook: DfGroupBy

Grouped Data

You can’t “show” it, you can’t “collect” it.

```
df.groupBy("dept").show()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-7-493f09bee249> in <module>
----> 1 df.groupBy("dept").show()

AttributeError: 'GroupedData' object has no attribute 'show'
```

Dealing with GroupData

- A GroupBy function returns a RelationalGroupedBy dataset
- It has a standard set of aggregation functions defined on it:
  - count
  - sum
  - min
  - max
  - avg

  Follow GroupBy with an aggregation …..

Grouped Data

- It will count everything in its sub-categories

```
df.groupBy("dept").count().show()
```

```
+---------+-----+
|     dept|count|
+---------+-----+
|     Math|    2|
|Business|    4|
| Biology|    3|
+---------+-----+
```

GroupBy … OrderBy .. a very common pattern

```
df.groupBy("dept", "age").count().orderBy("age").show()
```

```
+--------+---+-----+
|    dept|age|count|
+--------+---+-----+
|    Math| 18|    1|
|Business| 18|    1|
| Biology| 18|    1|
|Business| 19|    1|
|    Math| 19|    1|
| Biology| 20|    1|
|Business| 20|    1|
|Business| 21|    1|
| Biology| 23|    1|
+--------+---+-----+
```

GroupBy with Aggregate

```
df.groupBy("dept").agg({'age':'avg', 'hours':'sum', 'studentId' : 'count'}).show()
```

```
+--------+----------------+----------+------------------+
|    dept|count(studentId)|sum(hours)|          avg(age)|
+--------+----------------+----------+------------------+
|    Math|               2|        28|              18.5|
|Business|               4|        54|              19.5|
| Biology|               3|        36|20.333333333333332|
+--------+----------------+----------+------------------+
```