Statistics, Random Data, and Sampling on DataFrames

Lesson Objectives

- After completing this lesson, you should be able to:
  - Compute column summary statistics
  - Compute pairwise statistics between series/columns
  - Perform standard sampling on any DataFrame
  - Split any DataFrame randomly into subsets
  - Perform stratified sampling on DataFrames
  - Generate Random Data from Uniform and Normal Distributions

Summary Statistics for DataFrames

- Column summary statistics for DataFrames are available through DataFrame's describe() method
- It returns another DataFrame, which contains column-wise results for:
  - min, max
  - mean, stddev
  - Count
- Column summary statistics can also be computed through DataFrame's groupBy() and agg() methods, but stddev is not supported
- It also returns another DataFrame with the results

Example

```
from collections import namedtuple
Record = namedtuple('Record',['desc','value1','value2'])
```

```
recDF = sc.parallelize([Record("first",1,3.7),Record("second",-2,2.1),Record("third",6,0.7)]).toDF()
```

```
recStats = recDF.describe()
recStatsPandas = recStats.toPandas().set_index('summary')
recStatsPandas
```

	value1	value2
summary		
count	3	3
mean	1.6666666666666667	2.166666666666667
stddev	4.041451884327381	1.5011106998930273
min	-2	0.7
max	6	3.7

```
recStatsPandas.loc['mean'].value1
```

```
u'1.6666666666666667'
```

More Statistics on DataFrames

- More statistics are available through the stats method in a DataFrame
- It returns a DataFrameStatsFunctions object, which has the following methods:
  - corr() - computes Pearson correlation between two columns
  - cov() - computes sample covariance between two columns
  - crosstab() - Computes a pair-wise frequency table of the given columns
  - freqItems() - finds frequent items for columns, possibly with false positives

A Simple Example of Statistics

```
recDF.stat.corr('value1','value2')
```

```
-0.5879120879120879
```

```
recDF.stat.cov('value1','value2')
```

```
-3.5666666666666664
```

```
recDF.stat.freqItems(['value1','value2']).toPandas()
```

	value1_freqItems	value2_freqItems
0	[1, -2, 6]	[0.7, 2.1, 3.7]

Sampling on DataFrames

- Can be performed on any DataFrame
- Returns a sampled subset of a DataFrame
- Sampling with or without replacement
- Fraction: expected fraction of rows to generate
- Can be used on bootstrapping procedures

A Simple Sampling

```
from pyspark.sql import SQLContext
sqlc = SQLContext(sc)

df = sqlc.createDataFrame([(1, 10), (1, 20), (2, 10),(2, 20),
                           (2, 30), (3, 20), (3, 30)]).toDF("key", "value")
```

```
dfSampled = df.sample(withReplacement=False, fraction=0.3, seed=11)
dfSampled.toPandas()
```

	key	value
0	1	10
1	2	10
2	2	30

Random Split on DataFrames

- Can be performed on any DataFrame
- Returns an array of DataFrames
- Weights for the split will be normalized if they do not add up to 1
- Useful for splitting a data set into training, test and validation sets

A Simple Random Split

```
dfSplit = df.randomSplit(weights=[0.3, 0.7], seed=11)
dfSplit[0].toPandas()
```

	key	value
0	1	10
1	2	10
2	2	30

```
dfSplit[1].toPandas()
```

	key	value
0	1	20
1	2	20
2	3	20
3	3	30

Stratified Sampling on DataFrames

- Can be performed on any DataFrame
- Any column may work as key
- Without replacement
- Fraction: specified by key
- Available as sampleBy function in DataFrameStatFunctions

A Simple Stratified Sampling

```
dfStrat = df.stat.sampleBy(col="key", fractions={1: 0.7, 2: 0.7, 3: 0.7}, seed=11)
dfStrat.show()
```

```
+---+-----+
|key|value|
+---+-----+
|  1|   10|
|  1|   20|
|  2|   10|
|  2|   30|
|  3|   20|
|  3|   30|
+---+-----+
```

Random Data Generation

- sql functions to generate columns filled with random values
- Two supported distributions: uniform and normal
- Useful for randomized algorithms, prototyping and performance testing

Simple Examples

```
from pyspark.sql.functions import rand, randn
```

```
df = sqlc.range(0, 5)
```

```
df2 = df.select("id").withColumn("uniform", rand(5)).withColumn("normal", randn(5))
```

```
df2.show()
```

```
+---+------------------+--------------------+
| id|           uniform|              normal|
+---+------------------+--------------------+
|  0|0.47611851579756026|-0.21311682946326227|
|  1|0.06498948189958098|-0.05248092572410684|
|  2| 0.7069655052310547|  1.3682472758997855|
|  3| 0.1982919638208397|  -0.256535324205377|
|  4|0.12030715258495939|  -0.506853671746243|
+---+------------------+--------------------+
```

Lesson Summary

- Having completed this lesson, you should be able to:
  - Compute column summary statistics
  - Compute pairwise statistics between series/columns
  - Perform standard sampling on any DataFrame
  - Split any DataFrame randomly into subsets
  - Perform stratified sampling on DataFrames
  - Generate Random Data from Uniform and Normal Distributions