**Word Counter - Test Case 1 - Working**

**RDD**

**words.txt = echo “one two three”**

sc.textFile("words.txt")

.map(line => line.split(" ").size)

.reduce( (a:Int, b:Int) => a+b )

Output: Int = 3 (cannot .collect() this)

**Dataset**

spark.read.textFile("words.txt")

.map(l => l.split(" ").size)

.select(reduceAggregator( (a:Int, b:Int) => a+b )).collect()

Output: Array[Int] = Array(3)

**Dataframe**

spark.read.text("words.txt")

.selectExpr("sum(size(split(value,' ')))")

.as[Long].first

Output: Long = 3

\*\* add .toInt at the end to get Int = 3

**Dataframe Example - Test Case 2 - Working**

**RDD**

sc.range(10, 24)

.map( i => {val j=i%3; (I, if (j==0) i\*10 else i\*2) })

.map(r => r.\_1+r.\_2)

.collect()

Output: Array[Long] =

Array(30, 33, 132, 39, 42, 165, 48, 51 198, 57, 60, 231, 66, 69)

**Dataset**

spark.range(10, 24)

.map( i => {val j=i%3; (I, if (j==0) i\*10 else i\*2) })

.map(r => r.\_1+r.\_2)

.collect()

Output: Array[Long] =

Array(30, 33, 132, 39, 42, 165, 48, 51 198, 57, 60, 231, 66, 69)

**Dataframe**

spark.range(10, 24).selectExpr(“id as \_1”)

.selectExpr(“\_1 as \_1”, “if (\_1%3==0), \_1\*10, \_1\*2) as \_2”)

.selectExpr(“\_1+\_2 as \_1”)

.collect()

Output: Array[org.apache.spark.sql.Row] = Array([30], [33], [132], [39], [42], [165], [48], [51], [198], [57], [60], [231], [66], [69])

**Sort Least to Greatest - Test Case 3**

**RDD**

sc.textFile("numbers.txt")

.map(line => line.split(","))

.sortBy( (a:Long) => a )

.collect()

// creates another array, but don't know how else to view answers on terminal window easily

**Dataset**

Spark.read.textFile(“numbers.txt”)

.map(line => line.split(“,”))

..map(row=>(( ( (a:Long) => a ))(row), row)).orderBy("\_1").map(\_.\_2)

**Dataframe**

**ReduceByKey - Test Case 4**

**RDD**

sc.range(1,100)

.map(i=>(i%11, 1))

.reduceByKey((a:Int, b:Int) => a+b)

.collect()

**Dataset**

spark.range(1,100)

.map(i=>(i%11, 1))

.groupByKey(\_.\_1)

.agg(reduceByKeyAggregator((a:Int, b:Int) => a+b))

.collect()

**Dataframe**

**Even Numbers - Test Case 5 - Working**

**RDD**

sc.range(0, 51)

.filter( i => (i%2 == 0) )

.collect()

Output: Array[Long] = Array(0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50)

**Dataset**

spark.range(0, 101)

.filter( i => (i%2 == 0) )

.collect()

Output: Array[Long] = Array(0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50)

**Dataframe**

spark.range(0, 51).selectExpr("id as \_1")

.where(“\_1%2==0”)

.collect()

Output: Array[org.apache.spark.sql.Row] = Array([0], [2], [4], [6], [8], [10], [12], [14], [16], [18], [20], [22], [24], [26], [28], [30], [32], [34], [36], [38], [40], [42], [44], [46], [48], [50])

**Find Sum - Test Case 6**

**RDD**

sc.range(0, 10).as[Long[

.reduce((a:Long, b:Long) => a + b)

.collect()

**Dataset**

spark .range ( 0 , 10 ).as[Long]

.select(reduceAggregator ( ( a : Long , b : Long ) => a + b ))

.collect()

**Dataframe**