Problem Statement

Given a list of numbers - List[Int] (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

- find the sum of all numbers

- find the total elements in the list

- calculate the average of the numbers in the list

- find the sum of all the even numbers in the list

- find the total number of elements in the list divisible by both 5 and 3

Sum of all numbers

```
scala> val nums = sc.parallelize(List(1, 2, 3, 4,5,6,7,8,9,10))
nums: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val sum = nums.sum()
sum: Double = 55.0

scala>
```

Total elements in the list

```
scala> val count = nums.count()
count: Long = 10

scala>
```

Aaverage of the numbers in the list

```
scala> val avg = nums.mean()
avg: Double = 5.5

scala>
```

Acadgild_Session_16_Assignment_16.1_Solution

Sum of all the even numbers in the list

```
scala> val even = nums.filter(i => (i%2==0))
even: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[4] at filter at <console>:26

scala> val sumofeven = even.sum()
sumofeven: Double = 30.0

scala>
```

Total number of elements in the list divisible by both 5 and 3

```
scala> val multiple = nums.filter(i => (i%3==0 && i%5==0))
multiple: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[6] at filter at <console>:26

scala> val sumofmultiple = multiple.count()
sumofmultiple: Long = 0

scala>
```


Submitted By

Shishir Jha