Session 16 Assignment 1

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

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

Find the sum of all numbers

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

scala> numbers.sum
res4: Double = 55.0
```

Find the total elements in the list

a

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

scala> numbers.sum
res4: Double = 55.0

scala> numbers.count
res5: Long = 10
```

Calculate the average of the numbers in the list.

```
scala> val numbersavg = numbers.sum / numbers.count
numbersavg: Double = 5.5
```

Find the sum of all even numbers in the list.

```
scala> val evennumbers=numbers.filter(value => value% 2==0)
evennumbers: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[9] at filter at <console>:25

scala> evennumbers.foreach(println)
2
4
6
8
10

scala> println(evennumbers.sum)
30.0
```

Find the total number of elements divided by both 5 and 3.

```
scala> val divby5and3=numbers.filter(value => value% 5==0).intersection(numbers.filter(value => value%3==0)).cou
nt
divby5and3: Long = 0
```