TopCaptains

default

FINISHED

```
%md

## Working with Spark Scala RDD APIs

* This notebook gives on overview of how to read data into Spark Framework and apply basic RDD operations like parsing, filtering, grouping, sorting keys and values.

* This dataset is taken from cricinfo.com. There are two datasets available: [Captains ODIs](http://www.awesomestats.in/assets/data/captains_ODI.csv Tests](http://www.awesomestats.in/assets/data/captains_Test.csv)
```

Working with Spark Scala RDD APIs

- This notebook gives on overview of how to read data into Spark Framework and apply basic RDD operations like parsing, filtering, grouping, sorting and map reduce operations using keys and values.
- This dataset is taken from cricinfo.com. There are two datasets available: Captains ODIs (http://www.awesomestats.in/assets/data/captains_ODI.csv) & Captains Tests (http://www.awesomestats.in/assets/data/captains_Test.csv)

Took 1 seconds (outdated)

FINISHED

```
sc
```

```
res138: org.apache.spark.SparkContext = org.apache.spark.SparkContext@11160bec
```

Took 0 seconds (outdated)

FINISHED

```
// Read the ODI captains file. The file format is csv
var captains_odis = sc.textFile( "file:///home/hadoop/lab/data/captains_ODI.csv" )
```

```
captains_odis: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[20] at textFile at <console>:29
```

Took 0 seconds (outdated)

READY

TopCaptains

READY

default

FINISHED

```
// Print the first line
captains_odis.first()
```

```
res68: String = Ponting  R T,Australia,1995-2012,230,165,51,14,124
```

Took 0 seconds (outdated)

FINISHED

```
// Print the first 10 records
captains_odis.take( 10 ).foreach( println )
```

```
Ponting  R T,Australia,1995-2012,230,165,51,14,124
Fleming  S P,New Zealand,1994-2007,218,98,106,14,105
Ranatunga  A,Sri Lanka,1982-1999,193,89,95,9,102
Dhoni  M S*,India,2004-,186,103,68,15,88
Border  A R,Australia,1979-1994,178,107,67,4,86
Azharuddin  M,India,1985-2000,174,89,77,8,96
Smith  G C,South Africa,2002-2013,149,91,51,7,74
Ganguly  S C,India,1992-2007,147,76,66,5,74
Cronje  W J,South Africa,1992-2000,140,99,37,4,74
Imran Khan,Pakistan,1974-1992,139,75,59,5,70
```

Took 0 seconds (outdated)

FINISHED

```
// Count total number of records in the RDD
captains_odis.count()
```

```
res77: Long = 98
```

Took 0 seconds (outdated)

FINISHED

```
// define an object to represent each records in the csv file
case class Captain( name:String, country:String, career: String, matches: Int, won: Int, lost: Int, ties: Int, toss: Int )
```

```
defined class Captain
```

Took 1 seconds (outdated)

FINISHED

```
// define a function to actually parse these lines and convert into scala objects
def parse( line:String ) = {
    val pieces = line.split(",")
    val name = pieces(0)
    val country = pieces(1)
    val career = pieces(2)
    val matches = pieces(3).toInt
    val won = pieces(4).toInt
    val lost = pieces(5).toInt
    val ties = pieces(6).toInt
    val toss = pieces(7).toInt

    Captain( name, country, career, matches, won, lost, ties, toss )
}
```

```
parse: (line: String)Captain
```

Took 1 seconds (outdated)

FINISHED

```
var captains = captains_odis.map( line => parse( line ) )
```

```
captains: org.apache.spark.rdd.RDD[Captain] = MapPartitionsRDD[21] at map at <console>:35
```

Took 1 seconds (outdated)

FINISHED

```
captains.take( 10 ).foreach( println )
```

```
Captain(Ponting  R T,Australia,1995-2012,230,165,51,14,124)
Captain(Fleming  S P,New Zealand,1994-2007,218,98,106,14,105)
Captain(Ranatunga  A,Sri Lanka,1982-1999,193,89,95,9,102)
Captain(Dhoni  M S*,India,2004-,186,103,68,15,88)
Captain(Border  A R,Australia,1979-1994,178,107,67,4,86)
Captain(Azharuddin  M,India,1985-2000,174,89,77,8,96)
Captain(Smith  G C,South Africa,2002-2013,149,91,51,7,74)
Captain(Ganguly  S C,India,1992-2007,147,76,66,5,74)
Captain(Cronje  W J,South Africa,1992-2000,140,99,37,4,74)
Captain(Imran Khan,Pakistan,1974-1992,139,75,59,5,70)
```

Took 1 seconds (outdated)

FINISHED

```
// Captains who have played more than 100 matches
var captains_100 = captains.filter( rec => rec.matches > 100 )
```

```
captains_100: org.apache.spark.rdd.RDD[Captain] = MapPartitionsRDD[22] at filter at <console>:37
```

Took 1 seconds (outdated)

FINISHED

```
captains_100.count()
```

```
res96: Long = 16
```

Took 1 seconds (outdated)

FINISHED

```
captains_100.take( 16 ).foreach( println )
```

```
Captain(Ponting  R T,Australia,1995-2012,230,165,51,14,124)
Captain(Fleming  S P,New Zealand,1994-2007,218,98,106,14,105)
Captain(Ranatunga  A,Sri Lanka,1982-1999,193,89,95,9,102)
Captain(Dhoni  M S*,India,2004-,186,103,68,15,88)
Captain(Border  A R,Australia,1979-1994,178,107,67,4,86)
Captain(Azharuddin  M,India,1985-2000,174,89,77,8,96)
Captain(Smith  G C,South Africa,2002-2013,149,91,51,7,74)
Captain(Ganguly  S C,India,1992-2007,147,76,66,5,74)
Captain(Cronje  W J,South Africa,1992-2000,140,99,37,4,74)
Captain(Imran Khan,Pakistan,1974-1992,139,75,59,5,70)
Captain(Jayawardene  D P M,Sri Lanka,1998-2015,130,72,49,9,60)
Captain(Lara  B C,West Indies,1990-2007,125,59,59,7,57)
Captain(Jayasuriya  S T,Sri Lanka,1989-2011,117,65,47,5,61)
Captain(Wasim Akram,Pakistan,1984-2003,109,66,41,2,58)
Captain(Waugh  S R,Australia,1986-2002,106,67,35,4,48)
Captain(Richards  I V A,West Indies,1975-1991,105,67,36,2,43)
```

Took 1 seconds (outdated)

FINISHED

```
// Captains with more wins than losses
var captains_more_wins = captains_100.filter( rec => rec.won > rec.lost )
```

```
captains_more_wins: org.apache.spark.rdd.RDD[Captain] = MapPartitionsRDD[23] at filter at <console>:39
```

Took 0 seconds (outdated)

FINISHED

```
captains_more_wins.count()
```

```
res107: Long = 13
```

Took 0 seconds (outdated)

```
captains_more_wins.map( rec => rec.name ).collect()
```

FINISHED

```
res109: Array[String] = Array(Ponting  R T, Dhoni  M S*, Border  A R, Azharuddin  M, Smith  G C, Ganguly  S C, Cronje  W J, Imran Khan, Jayawardene
 D P M, Jayasuriya  S T, Wasim Akram, Waugh  S R, Richards  I V A)
```

Took 1 seconds (outdated)

```
// Captains with less wins than losses
var captains_more_losts = captains_100.filter( rec => rec.won <= rec.lost )
captains_more_losts.map( rec => rec.name ).collect()
```

FINISHED

```
captains_more_losts: org.apache.spark.rdd.RDD[Captain] = MapPartitionsRDD[27] at filter at <console>:39
res113: Array[String] = Array(Fleming  S P, Ranatunga  A, Lara  B C)
```

Took 1 seconds (outdated)

```
// Which country has played how many matches..
var countries = captains.map( rec => ( rec.country , rec.matches ) )
```

FINISHED

```
countries: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[29] at map at <console>:38
```

Took 0 seconds (outdated)

```
countries.take( 10 ).foreach( println )
```

FINISHED

```
(Australia,230)
(New Zealand,218)
(Sri Lanka,193)
(India,186)
(Australia,178)
(India,174)
(South Africa,149)
(India,147)
(South Africa,140)
(Pakistan,139)
```

Took 0 seconds (outdated)

```
var matches_countries = countries.reduceByKey( _+_ )
```

FINISHED

```
matches_countries: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[30] at reduceByKey at <console>:39
```

Took 0 seconds (outdated)

```
matches_countries.take( 20 ).foreach( println )
```

FINISHED

```
(Australia,832)
(Canada,27)
(Netherlands,31)
(Kenya,114)
(Bermuda,31)
(Afghanistan,50)
(Pakistan,781)
(Zimbabwe,394)
(Sri Lanka,710)
(Ireland,93)
(New Zealand,608)
(South Africa,463)
(Bangladesh,251)
(West Indies,658)
(India,770)
(England,554)
```

Took 1 seconds (outdated)

```
// Sort the countries by the number of matches they played. Sort by names...(sort by key)
matches_countries.sortByKey().take( 10 ).foreach( println )
```

FINISHED

```
(Afghanistan,50)
(Australia,832)
(Bangladesh,251)
(Bermuda,31)
(Canada,27)
(England,554)
(India,770)
(Ireland,93)
(Kenya,114)
(Netherlands,31)
```

Took 0 seconds (outdated)

```
// Sort the countries by the number of matches they played by descending order.. by names (key)
matches_countries.sortByKey( ascending = false ).take( 10 ).foreach( println )
```

FINISHED

```
(Zimbabwe,394)
```

```
(West Indies,658)
(Sri Lanka,710)
(South Africa,463)
(Pakistan,781)
(New Zealand,608)
(Netherlands,31)
(Kenya,114)
(Ireland,93)
(India,770)
```

Took 0 seconds (outdated)

FINISHED

```
// Sort by values.. default is ascending... min to max
var countries_by_matches = matches_countries.sortBy( rec => rec._2 )
```

```
countries_by_matches: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[68] at sortBy at <console>:42
```

Took 0 seconds (outdated)

FINISHED

```
countries_by_matches.collect().foreach( println )
```

```
(Canada,27)
(Netherlands,31)
(Bermuda,31)
(Afghanistan,50)
(Ireland,93)
(Kenya,114)
(Bangladesh,251)
(Zimbabwe,394)
(South Africa,463)
(England,554)
(New Zealand,608)
(West Indies,658)
(Sri Lanka,710)
(India,770)
(Pakistan,781)
(Australia,832)
```

Took 1 seconds (outdated)

FINISHED

```
// Sort by values.. set ascending to false... most to least
var countries_by_matches = matches_countries.sortBy( rec => rec._2, ascending = false )
```

```
countries_by_matches: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[73] at sortBy at <console>:41
```

Took 1 seconds (outdated)

FINISHED

```
countries_by_matches.collect().foreach( println )
```

```
(Australia,832)
(Pakistan,781)
(India,770)
(Sri Lanka,710)
(West Indies,658)
(New Zealand,608)
(England,554)
(South Africa,463)
(Zimbabwe,394)
(Bangladesh,251)
(Kenya,114)
(Ireland,93)
(Afghanistan,50)
(Netherlands,31)
(Bermuda,31)
(Canada,27)
```

Took 1 seconds (outdated)

FINISHED

```
// Captains by percentage of wins
import scala.math._

var captains_100_percent_wins = captains_100.map( rec => ( rec.name, ( rec.won.toFloat / rec.matches.toFloat ) ) )
```

```
import scala.math._
captains_100_percent_wins: org.apache.spark.rdd.RDD[(String, Float)] = MapPartitionsRDD[86] at map at <console>:61
```

Took 1 seconds (outdated)

FINISHED

```
captains_100_percent_wins.take( 10 ).foreach( println )
```

```
(Ponting  R T,0.7173913)
(Fleming  S P,0.44954127)
(Ranatunga  A,0.4611399)
(Dhoni  M S*,0.55376345)
(Border  A R,0.6011236)
(Azharuddin  M,0.5114943)
```

```
(Smith  G C,0.6107383)
(Ganguly  S C,0.5170068)
(Cronje  W J,0.70714283)
(Imran Khan,0.53956836)
```

Took 1 seconds (outdated)

FINISHED

```
// Sort by percentage wins
var best_odi_captains = captains_100_percent_wins.sortBy( rec => rec._2, ascending = false )
```

```
best_odi_captains: org.apache.spark.rdd.RDD[(String, Float)] = MapPartitionsRDD[96] at sortBy at <console>:63
```

Took 0 seconds (outdated)

FINISHED

```
best_odi_captains.take( 10 ).foreach( println )
```

```
(Ponting  R T,0.7173913)
(Cronje  W J,0.70714283)
(Richards  I V A,0.63809526)
(Waugh  S R,0.6320755)
(Smith  G C,0.6107383)
(Wasim Akram,0.6055046)
(Border  A R,0.6011236)
(Jayasuriya  S T,0.5555556)
(Jayawardene  D P M,0.5538462)
(Dhoni  M S*,0.55376345)
```

Took 1 seconds (outdated)

FINISHED

```
%md

## Exercise for the participants
#### Filter countries which have played more than 100 matches
#### Sort countries by the percentage of matches they lost
#### Find the top10 lucky captains in terms of TOSSes they have won
```

Exercise for the participants

Filter countries which have played more than 100 matches

Sort countries by the percentage of matches they lost

Find the top10 lucky captains in terms of TOSSes they have won

Took 0 seconds (outdated)

FINISHED

```
// Read the Captains_Test.csv file
var captains_tests = sc.textFile( "file:///home/hadoop/lab/data/captains_Test.csv" )
```

```
captains_tests: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[98] at textFile at <console>:51
```

Took 1 seconds (outdated)

FINISHED

```
var captains_tests_recs = captains_tests.map( line => parse( line ) )
```

```
captains_tests_recs: org.apache.spark.rdd.RDD[Captain] = MapPartitionsRDD[100] at map at <console>:56
```

Took 0 seconds (outdated)

FINISHED

```
captains_tests_recs.take( 10 ).foreach( println )
```

```
Captain(Smith  G C,South Africa,2002-2014,109,53,29,27,58)
Captain(Border  A R,Australia,1978-1994,93,32,22,38,47)
Captain(Fleming  S P,New Zealand,1994-2008,80,28,27,25,38)
Captain(Ponting  R T,Australia,1995-2012,77,48,16,13,37)
Captain(Lloyd  C H,West Indies,1966-1984,74,36,12,26,35)
Captain(Dhoni  M S*,India,2005-,60,27,18,15,27)
Captain(Waugh  S R,Australia,1985-2004,57,41,9,7,31)
Captain(Ranatunga  A,Sri Lanka,1982-2000,56,12,19,25,29)
Captain(Atherton  M A,England,1989-2001,54,13,21,20,23)
Captain(Cronje  W J,South Africa,1992-2000,53,27,11,15,22)
```

Took 1 seconds (outdated)

FINISHED

```
// Filter the captains who have captained for more than 100 tests
var captains_tests_100 = captains_tests_recs.filter( rec => rec.matches > 100 )
```

```
captains_tests_100: org.apache.spark.rdd.RDD[Captain] = MapPartitionsRDD[101] at filter at <console>:59
```

Took 1 seconds (outdated)

FINISHED

```
// How many captains?
captains_tests_100.take( 10 )
```

```
res268: Array[Captain] = Array(Captain(Smith  G C,South Africa,2002-2014,109,53,29,27,58))
```

Took 0 seconds (outdated)

FINISHED

```
// There is only one captain who have played more than 100 tests, so lets downgrade the filter criteria to 50 test matches

// Filter the captains who have captained for more than 50 tests
var captains_tests_50 = captains_tests_recs.filter( rec => rec.matches > 50 )
```

```
captains_tests_50: org.apache.spark.rdd.RDD[Captain] = MapPartitionsRDD[102] at filter at <console>:61
```

Took 0 seconds (outdated)

FINISHED

```
captains_tests_50.take( 10 ).foreach( println )
```

```
Captain(Smith  G C,South Africa,2002-2014,109,53,29,27,58)
Captain(Border  A R,Australia,1978-1994,93,32,22,38,47)
Captain(Fleming  S P,New Zealand,1994-2008,80,28,27,25,38)
Captain(Ponting  R T,Australia,1995-2012,77,48,16,13,37)
Captain(Lloyd  C H,West Indies,1966-1984,74,36,12,26,35)
Captain(Dhoni  M S*,India,2005-,60,27,18,15,27)
Captain(Waugh  S R,Australia,1985-2004,57,41,9,7,31)
Captain(Ranatunga  A,Sri Lanka,1982-2000,56,12,19,25,29)
Captain(Atherton  M A,England,1989-2001,54,13,21,20,23)
Captain(Cronje  W J,South Africa,1992-2000,53,27,11,15,22)
```

Took 1 seconds (outdated)

FINISHED

```
// Sort the captains by percentage of wins
var captain_top = captains_tests_50.map( rec => ( rec.name, ( rec.won.toFloat / rec.matches.toFloat) ) ).sortBy( rec => rec._2, ascending = false )
```

```
captain_top: org.apache.spark.rdd.RDD[(String, Float)] = MapPartitionsRDD[108] at sortBy at <console>:61
```

Took 0 seconds (outdated)

FINISHED

```
captain_top.collect()
```

```
res280: Array[(String, Float)] = Array((Waugh  S R,0.71929824), (Ponting  R T,0.6233766), (Vaughan  M P,0.50980395), (Cronje  W J,0.509434), (Lloyd
 C H,0.4864865), (Smith  G C,0.48623854), (Dhoni  M S*,0.45), (Fleming  S P,0.35), (Border  A R,0.34408602), (Atherton  M A,0.24074075), (Ranatunga
 A,0.21428572))
```

Took 0 seconds (outdated)

```
// Lets join both ODI and Test captaincy details. Default is inner join...
var all_time_best_captains = captains_100_percent_wins_join( captain_top )
```

FINISHED

```
all_time_best_captains: org.apache.spark.rdd.RDD[(String, (Float, Float))] = MapPartitionsRDD[111] at join at <console>:71
```

Took 0 seconds (outdated)

```
all_time_best_captains.take( 10 ).foreach( println )
```

FINISHED

```
(Dhoni  M S*,(0.55376345,0.45))
(Ranatunga  A,(0.4611399,0.21428572))
(Cronje  W J,(0.70714283,0.509434))
(Border  A R,(0.6011236,0.34408602))
(Fleming  S P,(0.44954127,0.35))
(Waugh  S R,(0.6320755,0.71929824))
(Smith  G C,(0.6107383,0.48623854))
(Ponting  R T,(0.7173913,0.6233766))
```

Took 1 seconds (outdated)

```
// Let's sort by test match wins
all_time_best_captains.sortBy( rec => rec._2._2, ascending = false ).collect().foreach( println )
```

FINISHED

```
(Waugh  S R,(0.6320755,0.71929824))
(Ponting  R T,(0.7173913,0.6233766))
(Cronje  W J,(0.70714283,0.509434))
(Smith  G C,(0.6107383,0.48623854))
(Dhoni  M S*,(0.55376345,0.45))
(Fleming  S P,(0.44954127,0.35))
(Border  A R,(0.6011236,0.34408602))
(Ranatunga  A,(0.4611399,0.21428572))
```

Took 1 seconds (outdated)

```
// What if we give 60% weightage to test wins and 40% weightage to odi wins
var best_captains = all_time_best_captains.map( rec => ( rec._1, ( rec._2._1 * 0.4 + rec._2._2 * .6 ) ) )
```

FINISHED

```
best_captains: org.apache.spark.rdd.RDD[(String, Double)] = MapPartitionsRDD[125] at map at <console>:73
```

Took 0 seconds (outdated)

```
best_captains.sortBy( rec => rec._2, ascending = false ).take( 10 ).foreach( println )
```

FINISHED

```
(Waugh  S R,0.6844091415405273)
```

```
(Ponting  R T,0.6609824895858765)
(Cronje  W J,0.5885175228118896)
(Smith  G C,0.5360384345054626)
(Dhoni  M S*,0.4915053725242615)
(Border  A R,0.44690104126930236)
(Fleming  S P,0.3898165047168732)
(Ranatunga  A,0.31302738487720494)
```

Took 1 seconds (outdated)

FINISHED

```
%md

## Exercise

#### Find out top 5 worst captains in ODI and test, who have played more than 50 matches
```

Exercise

Find out top 5 worst captains in ODI and test, who have played more than 50 matches

Took 0 seconds (outdated)

READY

```

```