CS 696 Intro to Big Data
Fall Semester, 2016
Assignment 2

© 2016, All Rights Reserved, SDSU & Roger Whitney

San Diego State University -- This page last updated 9/11/17

Due Sept 24 23:59

In problems 1 - 4 you are to use higher order functions (map, reduce, filter, …) to solve the problems. You do not need to use for or while statements. The solutions with map, reduce and/or filter may not be as efficient as solutions using a for or while. However the core operations in Hadoop are map and reduce. While there is a difference in the Hadoop mapReduce from what you will use hear it is important to get used to using those constructs.

In each of the solutions to the problems you are to include tests that show that your code works.

1. If we list all the natural numbers below 20 that are multiples of 3 or 5 but not multiples both of 3 and 5, we get 3, 5, 6, 9, 10, 12, 18. The sum of these multiples is 63. Write a function, sum_multiples_3_5, that returns the sum of the multiples of 3 and 5 but not both less than N.

2. Write a function, pattern_count with two arguments. The first arguments is a string, lets call it text, and the second argument is also a string, call it pattern. The function pattern_count returns the number of times the pattern occurs in the text. For example

   pattern_count(“abababa”, “aba”) == 3
   pattern_count(“aaaaa”, “aa”) == 4
   pattern_count(“Abcde”, “abc”) == 0

3. Implement factorial using “to” and “reduceLeft” without a loop or recursion.

4. Write a function that receives a collection of strings and a map from strings to integers. Return a collection of integers that are values of the map corresponding to one of the strings in the collection. For example, given Array("Tom", "Fred", "Harry") and Map("Tom" -> 3, "Dick" -> 4, "Harry" -> 5), return Array(3, 5).

5. Implement a function sampleNoRepeats with two arguments. The first argument is a range, the second an integer, n, greater that zero. The function returns n random numbers in the given range with no repeats.

6. Create an array with random 100,000 integers between 0 and 50,000. Compute the mean value of the array, call it M. Take a random sample of 100 different elements (sampling without replacement) of the array and compute the mean of the sample. What is the error?

7. Using the same array as in #6 take 1,000 samples of 100 random elements of the array. Compute the mean of each sample. Find the value r where interval (M-r, M+r) contains 95% of the means of the 1000 samples.

8. Repeat #7 but use the sample size of 1,000 rather than 100.

9. Repeat #8 but use the sample size of 10,000 rather than 100.

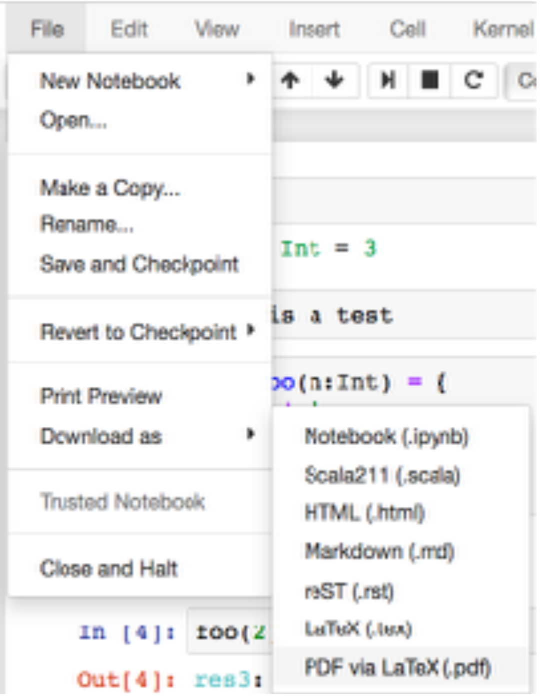
Grading

Each problem is worth 10 points.

What to turn in

You are to turn in a Jupyter Jupyter-scala notebook containing the answers to the questions above. Since Jupyter notebooks can contain text and code, before each problem indicate which problem it is in text, not in code comment.

To turn in your assignment download your Jupyter notebook as an Notebook (.ipynb). See image below. This will allow me to run your assignment in Jupyter. Do not download it as a Scala file (.ji) as this will not run in Jupyter and removes all the text (markdown).



```
Int = 3
is a test
oo(n:Int) = {
In [4]: foo(2
Out[4]: res3:
```

Once you have downloaded the assignment zip it up and then upload the zip file to the course portal.

Late Penalty

An assignment turned in 1-7 days late, will lose 5% of the total value of the assignment per day late. The eight day late the penalty will be 40% of the assignment, the ninth day late the penalty will be 60%, after the ninth day late the penalty will be 90%. Once a solution to an assignment has been posted or discussed in class, the assignment will no longer be accepted. Late penalties are always rounded up to the next integer value.