# ASSIGNMENT 6

## Part I

In this assignment, you will use Pig to find the results of the same queries as in assignment 5. Submit just your code.

---

The dataset files are located in hdfs in the following path,

**/yelpdatafall/business/business.csv.**
**/yelpdatafall/review/review.csv.**
**/yelpdatafall/user/user.csv.**

**Dataset Description.**
The dataset comprises of **three** csv files, namely user.csv, business.csv and review.csv. Note that some of the content, such as id fields are encoded. Note that the files are separated by "^" character.

**1. Business.csv** file contain basic information about local businesses.
**Business.csv** file contains the following columns
"business_id","full_address","categories"

'business_id': (a unique identifier for the business)
'full_address': (localized address),
'categories': [(localized category names)]

**2. Review.csv** file contains the star rating given by a user to a business. Use user_id to associate this review with others by the same user. Use business_id to associate this review with others of the same business.

**review.csv** file contains the following columns
"review_id","user_id","business_id","stars"
'review_id': (a unique identifier for the review)
'user_id': (the identifier of the reviewed business),
'business_id': (the identifier of the authoring user),
'stars': (star rating, integer 1-5),the rating given by the user to a business

**3. user.csv file** contains aggregate information about a single user across all of Yelp
**user.csv file** contains the following columns "user_id","name","url"
user_id': (unique user identifier),
'name': (first name, last initial, like 'Matt J.'), this column has been made anonymous to preserve privacy
'url': url of the user on yelp

---

** Remember you have to use Pig to solve these questions**
**Hint: You can load a "^" separated file as:**
**business = LOAD '/yelpdatafall/business/business.csv' using PigStorage('^') as (businessId, fullAddress, categories);**

**Q1. List the business_id , full address and categories of the Top 10 highest rated businesses using the average ratings.**

This will require you to use **review.csv** and **business.csv** and join them on the common key (business_id)

**Sample output:**

```
business id         full address        categories                          avg rating
xdf12344444444,      CA 91711      List['Local Services', 'Carpet Cleaning']     5.0
```

**Q2. Read a user name from the command line and find the average of their review rating.**

For example, if the command line argument is "Matt J", you need to output the average review ratings of that user.

Q3. **List the 'user id' and 'stars' of users that reviewed businesses located in Stanford.**

You would need to filter the business.csv files by addresses that contain the word "Stanford". There is no need for any aggregation operation.

**Q4. List the user_id , and name of the top 10 users who have written the most reviews.**

**Q5. List the business_id, and count of each business's ratings for the businesses that are located in the state of TX**

---

## Part II

A link to selected books of Shakespeare is available here:
http://www.utdallas.edu/~axn112530/cs6350/shakespeare.tar.gz
(you can combine the multiple files into one big file)

and a link to the King James Bible is available here:
http://www.utdallas.edu/~axn112530/cs6350/bible.tar.gz

You would like to find the top 10 frequently occurring common words and their counts in these two books that are of a specified length k, where k = 5, 6, 7, or 8. By keeping the word length large, you are minimizing the presence of stop words.

You have to do this using:
1. Spark
2. Pig

Your code should output the top 10 most commonly occurring words and their counts for four word lengths. It is up to you how you implement this – either have the word length as a parameter or hard code it.

We are also interested in the speed of computation, so you have to time the execution of the code for Pig and Spark.

Submit your code and the output of the programs for 4 word length values and also how long it took each language to output the results.