# TI2736-B: Assignment 5
# Big Data Processing

Due date: 8.01.2017 @ 11.59pm

For each exercise, submit on Blackboard your code and the output of running this code. We will execute all submitted scripts and validate them. Do **not** copy & paste your code into a PDF or Word document, submit the source code files. Make sure to include your name and student number in your submission.

In this assignment, you will practice your Pig skills once more. For the exercises in this assignment, we make use of one of the example datasets from the *Pig Programming* book (*baseball*). The schema of the baseball dataset is described at:
`https://github.com/alanfgates/programmingpig/tree/master/data`.
Each line in the dataset refers to a distinct player. If the same name appears in multiple lines, you should consider those to be distinct players.
To start writing scripts, download the dataset(s) to a local directory within your virtual machine (if you use CDH). Open a terminal, move to the directory you stored the datasets in and type `pig -x local`. This opens the interactive shell (called Grunt) in which you can test and type out your Pig Latin scripts. Note that starting the shell from the same directory as your data is just for convenience, as it saves you typing effort when loading data from file.
To avoid a possible reoccurring error in the Grunt shell, type as first command `set io.sort.mb 5;` (this ensures that Pig does not use too much memory when sorting).

1. [*baseball*] Write a Pig Latin script that outputs for each team, the average number of games their players played for them.

2. [*baseball*] Write a Pig Latin script that outputs for each player with more than 100 games under his belt the following:

$$r = \frac{\textit{number of homeruns}}{\textit{number of games}} \tag{1}$$

   In the output, the players should be ranked by $r$ in descending order.

3. [*baseball*] Write a Pig Latin script that outputs for each position the number of distinct players that held it.

4. [*baseball*] Write a Pig Latin script that outputs the list of distinct positions.

5. [*baseball*] Write a *FilterFunc* UDF that takes that takes a player tuple as input and filters out (i.e. removes) those players that are not *Pitchers* and players that are *Pitchers* but whose fraction of strikeouts per game, i.e.

$$x = \frac{\textit{number of strikeouts}}{\textit{number of games}} \tag{2}$$

   is less than the average of $x$ computed across all *Pitchers*. Write a Pig Latin script that uses this UDF and outputs in the end all players that held the position of *Pitcher* and whose strikeout fraction is above average.

6. [*baseball+majorleague-payroll.csv*] You now received a second dataset from a colleague who asks you to list all players that play for one of the high-paying teams mentioned in *majorleague-payroll.csv* (i.e. teams that report an average salary of more than 3 million dollars). Write a Pig Latin script to that effect.
   Note: do **not** manually change the payroll data set, use pig functionality to extract the required information.

7. [*baseball+majorleague-payroll.csv+majorleague-payroll2.csv*] Another colleague of yours disagrees with the colleague from the previous exercise and produces another salary overview of baseball teams (similar but slightly different to the previous one). To satisfy both of your colleagues, you decide to only list all those players that play for one of the teams mentioned in both payroll overviews **and** that in **both** payrolls report an average salary of more than 3 million dollars. Write a Pig Latin script to that effect.
   Note: do **not** manually change the payroll data sets, use pig functionality to extract the required information.