

William Shakespeare's (1564 - 1616) plays have become a popular corpus for text analysis. Project Gutenberg (gutenberg.org) has compiled plays into a single text file that you can find in Moodle.

1.) To get a first idea, run a quick analysis on the text using Unix tools such as *wc* and *grep* to answer the following questions: [1 Pts]

- a) How big is the file?
 - 1. Size in kB?
 - 2. in words?
 - 3. in lines of text?
- b) How many plays does it contain?

2.) What does the following pipeline of shell-commands produce? [1 Pts]

```
grep -B 6 'by William Shakespeare' Shakespeare.txt | \
    grep -v -e '^$' | tr '\n' ' ' | sed 's/ -- /\n/g'
```

How many processes are involved in this execution?

3.) Design an Apache Spark pipeline that: [1 Pts]

- a) first removes (filters) the unwanted header (the text starts at line 245) off the text,
- b) then removes the Copyright-phrases (text between “<<THIS ELECTRONIC VERSION ... FOR MEMBERSHIP.>>” – there are multiple occurrences in the text),
- c) then splits the text in segments of plays,
- d) to spawn each play segment for separate (parallel) execution,
- e) collect and finally combine results from processing.

Name a Spark transformation or action for each stage and show how stages are connected (e.g. draw the pipeline and the a DAG as a sketch).

4.) Write Python functions for stages a) ... c) and test them separately.

5.) Write a python function *play_counts()* for stage d) for parallel analysis of all plays producing one line per play in the form:

“<play title>, xx lines, yy words.”

6.) Implement the pipeline in Spark. Summarize results for all plays in a resulting list of plays ordered by the number of lines a play has with the longest play first. Think about a convenient structure returned from play analysis that allows sorting in the final stage (avoiding parsing the result string).

Provide a log or screenshot of the execution and the resulting list. [4 Pts]

7.) Answer the question: What are stop words and why should they be removed or disregarded for text analysis? Find a list of English stop words on the Internet. Provide the URL and three examples. [1 Pts]

EP1, Extra Points: [+4 EP]

Select another analysis from: <https://verbingnouns.github.io/AdventuresInR/docs/shakespeare.nb.html> and implement in Spark. Demonstrate a working example.

Read through the material to see the scope of text analysis and visualization that can be done.