Overview
For this exercise you'll be working with the publicly available Enron Email dataset.
Dataset home page: https://www.cs.cmu.edu/~./enron/
Direct link to the dataset: https://www.cs.cmu.edu/~./enron/enron_mail_20150507.tgz

The goal is to produce a list of word counts for words that appear in the bodies of the Enron emails (ignoring the headers.) Only words that appear 10 or more times will be included. The list must be sorted in order of count descending. You can produce the output in whatever format you like (CSV, MySQL table, parquet file, etc.)

If you feel a simple word count is not interesting enough you can add additional features such as stop words, stemming, or per-sender counts. However note that the main purpose of the exercise is not to demonstrate skills in areas like NLP. You should focus on demonstrating construction of a robust, maintainable, and scalable data pipeline.

You can spend as long as you like on your solution, however we recommend blocking out 4-12 hours.

Example Output

word	**count**
foo	8,938
bar	5,392
baz	4,392
fnord	2,918

Non-functional Requirements
Write your solution as if it was for production (unit tests, documentation, etc.). Given that this is a code challenge don't put in an inordinate amount of time, but at least demonstrate how you would test and document. For example if comprehensive unit and integration test coverage will take too long, at least write a few good example tests.

Reliability is also a major concern. Assume that the accuracy of the word counts is business critical, and the viability of the business could be put at risk if the counts are incorrect (or are perceived as incorrect by end users.) Your solution should produce some evidence to increase confidence that the output is accurate. You may run into issues that you can't resolve in a limited time period (e.g. it may not be viable to 100% confidently distinguish email body from headers.) The critical thing is to document these types of issues and quantify them as much as

possible. Also document any assumptions (e.g. “each file represents a single email message”) that you don’t have time to fully validate.

In addition to processing the Enron dataset, assume that your solution will need to eventually scale to handle a dataset 10,000x larger (10s of TBs.) You may or may not actually implement and test this level of scalability but you should be prepared to discuss what work would be involved to scale your solution.

Development Environment

If your personal machine isn’t up to the challenge of processing the whole Enron dataset, feel free to just process some more manageable subset of it. If you choose a distributed platform like Hadoop or Spark to implement your solution don’t feel like you need to actually spin up a cluster in the cloud, but be prepared to discuss how that would work.