**Challenge 1**

In this challenge, you get to be the *boss*. You oversee hundreds of employees across the country developing Tuna 2.0, a world-changing snack food based on canned tuna fish. Alas, being the boss isn't all fun, games, and self-adulation. The company recently decided to purchase a new HR system, and unfortunately for you, the new system requires employee records be stored completely differently.

Your task is to help bridge the gap by creating a Python script able to convert your employee records to the required format. Your script will need to do the following:

* Import the employee\_data.csv file, which currently holds employee records like the below:

Emp ID,Name,DOB,SSN,State

214,Sarah Simpson,1985-12-04,282-01-8166,Florida

15,Samantha Lara,1993-09-08,848-80-7526,Colorado

411,Stacy Charles,1957-12-20,658-75-8526,Pennsylvania

* Then convert and export the data to use the following format instead:

Emp ID,First Name,Last Name,DOB,SSN,State

214,Sarah,Simpson,12/04/1985,\*\*\*-\*\*-8166,FL

15,Samantha,Lara,09/08/1993,\*\*\*-\*\*-7526,CO

411,Stacy,Charles,12/20/1957,\*\*\*-\*\*-8526,PA

* In summary, the required conversions are as follows:
  + The Name column should be split into separate First Name and Last Name columns.
  + The DOB data should be re-written into MM/DD/YYYY format.
  + The SSN data should be re-written such that the first five numbers are hidden from view.
  + The State data should be re-written as simple two-letter abbreviations.
* Special Hint: You may find this link to be helpful—[Python Dictionary for State Abbreviations](https://gist.github.com/afhaque/29f0f4f37463c447770517a6c17d08f5).

**Challenge 2**

In this challenge, you get to play the role of chief linguist at a local learning academy. As chief linguist, you are responsible for assessing the complexity of various passages of writing, ranging from the sophomoric Twilight novel to the nauseatingly high-minded research article. Having read so many passages, you've since come up with a fairly simple set of metrics for assessing complexity.

Your task is to create a Python script to automate the analysis of any such passage using these metrics. Your script will need to do the following:

* Import a text file filled with a paragraph of your choosing.
* Assess the passage for each of the following:
  + Approximate word count
  + Approximate sentence count
  + Approximate letter count (per word)
  + Average sentence length (in words)
* As an example, this passage:

“Adam Wayne, the conqueror, with his face flung back and his mane like a lion's, stood with his great sword point upwards, the red raiment of his office flapping around him like the red wings of an archangel. And the King saw, he knew not how, something new and overwhelming. The great green trees and the great red robes swung together in the wind. The preposterous masquerade, born of his own mockery, towered over him and embraced the world. This was the normal, this was sanity, this was nature, and he himself, with his rationality, and his detachment and his black frock-coat, he was the exception and the accident a blot of black upon a world of crimson and gold.”

...would yield these results:

Paragraph Analysis

-----------------

Approximate Word Count: 122

Approximate Sentence Count: 5

Average Letter Count: 4.6

Average Sentence Length: 24.0

* **Special Hint:** You may find this code snippet helpful when determining sentence length (look into [regular expressions](https://en.wikipedia.org/wiki/Regular_expression) if interested in learning more):

import re

re.split("(?<=[.!?]) +", paragraph)