# SRG Coding Challenge

As part of our interview assesment we want to make sure that you have a curious mind, are able to use internet resources successfully and have a good understanding of Python (the primary coding language for SRG projects) The exercise below should be committed to your own github https://github.com (https://github.com) in a public repository or emailed back to us.

We recommend the use of Jupyter notebooks https://jupyter.org (https://jupyter.org) as they are useful for producing good display outputs and are also a tool frequently used by the team.

When thinking about the challenge it is worth paying particular attention to:

1. A new user viewing your repository and ensuring they can clone and run your code
2. Prevention of commiting secrets or hard coded artefacts to your repository
3. Algorithm efficiency
4. Visualisations

You will be asked to explain your code and thought process, so although there are existing solutions to this problem you will need to be able to understand the code and explain to SRG team members how it works.

This challenge should take no longer than 6 hours.

## Challenge

In tmdb_5000_movies.csv you'll find the top movies as listed by https://www.themoviedb.org/ (https://www.themoviedb.org/). Read in the file and then:

1. **Produce a summary of the dataset** - size/shape, column headers, unique values etc.
2. **Calculate basic statistics of the data** - (max, mins, count, mean, std, etc) and examine data and state your observations.
3. **Self led exploration** - Do any data exploration you think would be of interest and create visualisations that show your work and support observations.
4. **Create your own top 250** - Use the formula below to create a new top 250 and check if where it maps with the ranking in the database provided(source https://en.wikipedia.org/wiki/IMDb#Rankings (https://en.wikipedia.org/wiki/IMDb#Rankings)):

$$W = \frac{R * v + C * m}{v + m}$$

Where:

```
W = weighted rating
R = average rating for the movie as a number from 1 to 10 (vote_average)
v = number of votes for the movie (vote_count)
m = minimum votes required to be listed in the Top 250 (currently 25,000)
C = the mean vote across the whole report
```