## ` **Project Report**

11.17.2022

**─**  
Human-Computer Interaction  
  
Walker Daniels - Main Coder & Report  
Oscar Salinas - Writer (Report & Slides) & Coder  
Robert Taylor - Coder & Report   
Zachary Urrutia - Writer (Report & Slides) & Coder

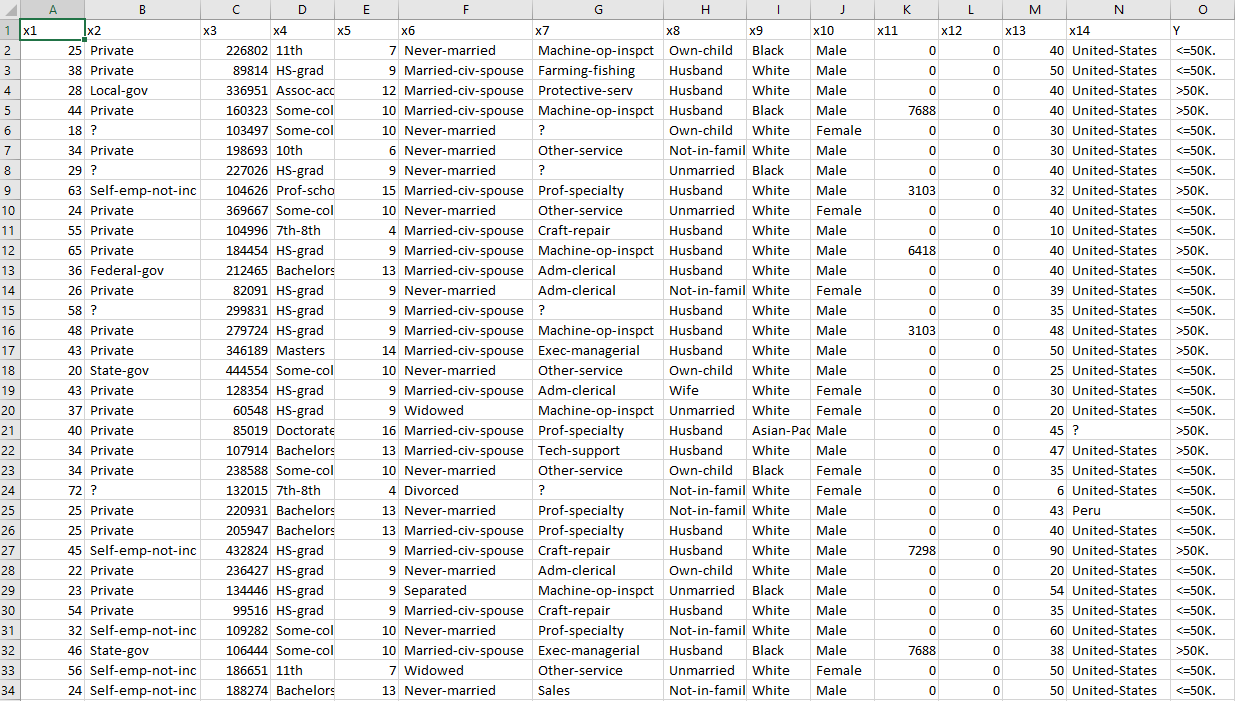
# **Overview**

1. Introduction
2. Provided Data Instance
3. Preparing Data
4. Determining Occurrences
5. Algorithmic Description and Implementation
6. Implementation Results
7. Pros and Con
8. Further Improvements

# 

# **Introduction**

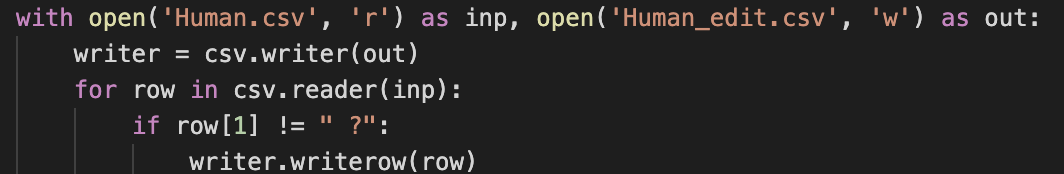
Dr. Huang has tasked our group with completing specific tasks regarding the manipulation of data within a CSV file. As a group, we are to Identify the Gross Pay of an individual on Unigram Language Model by Corpus Cross Entropy, as well as providing additional data, dissecting the Human.csv file into more digestible/readable data. Demonstrating this will give our audience a better understanding of Corpus Cross Entropy in this project. Using our knowledge from Human-Computer Interaction and our coding knowledge as a group, we used the programming language Python to implement all this.

**Provided Data Instances**Data instance - Data used in our project

The provided data instance is a simple CSV file titled ‘human.csv’ it contains typical information one might expect, including education level, race, sexuality, gender.

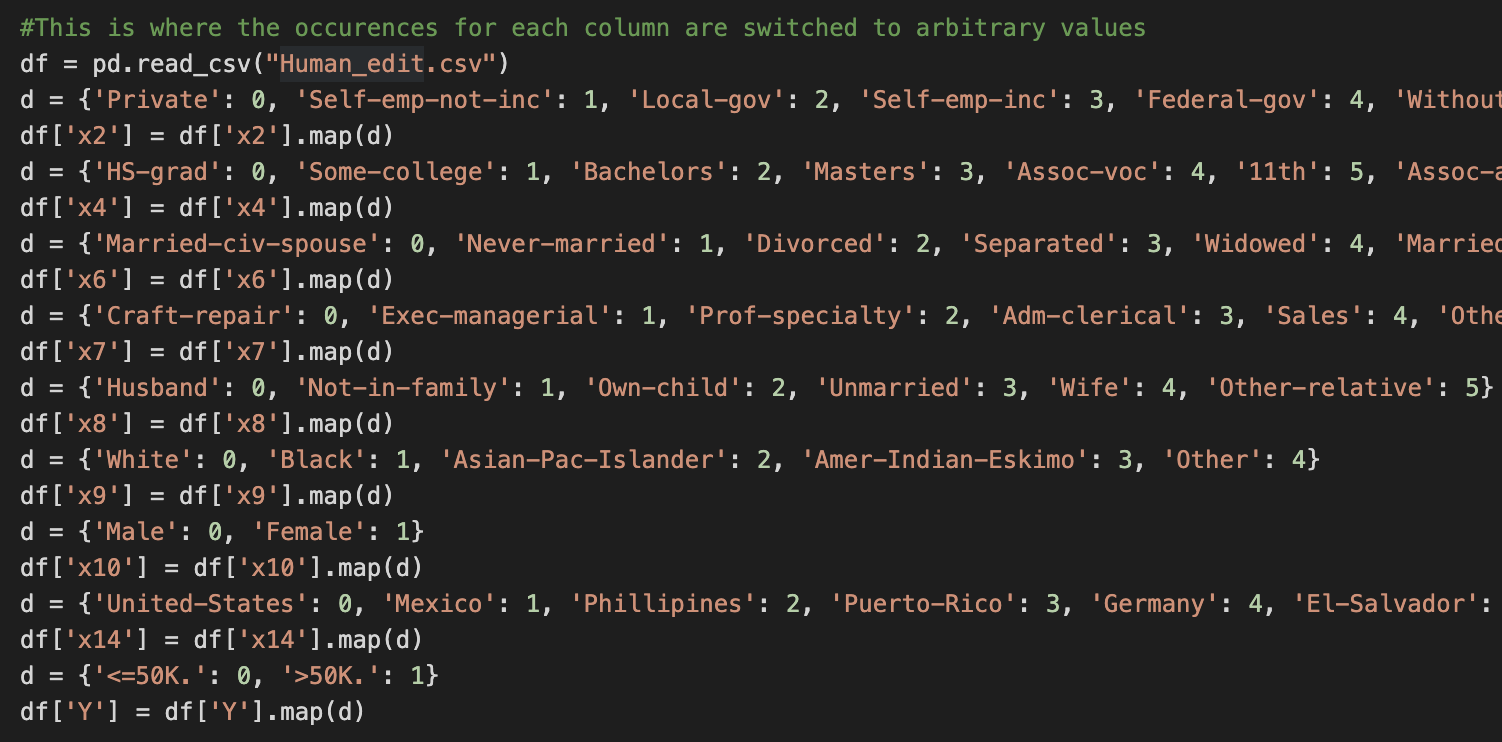
**Preparing Data**

The first task needed to complete this project was to prepare the human.csv dataset. The provided data has a few blank sections in its columns with missing values. These are marked with ‘?’. To remove them we ran a for loop and wrote into a new csv file titled ‘human\_edit’ for every line that didn’t contain a ‘?’.

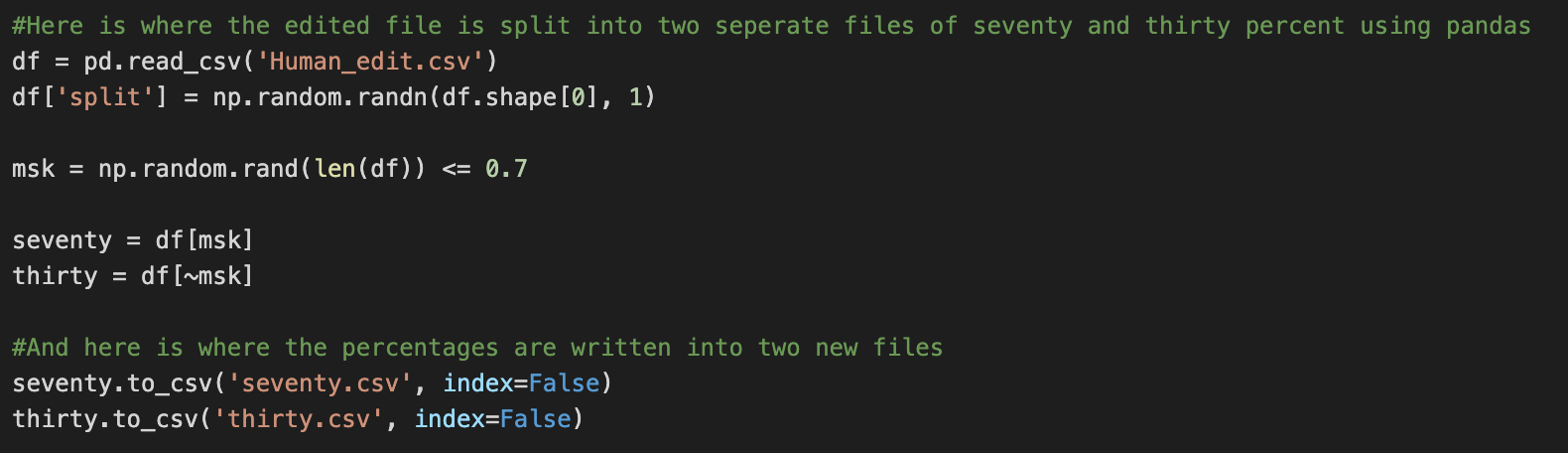
CSV Element Replace Code - gets rid of “?” in the data set ****

The second requirement was to change all of the string info into arbitrary code. This was done by reading our file into an instance variable, using the replace function to switch every unique string with a number, and then mapping them back into the instance.

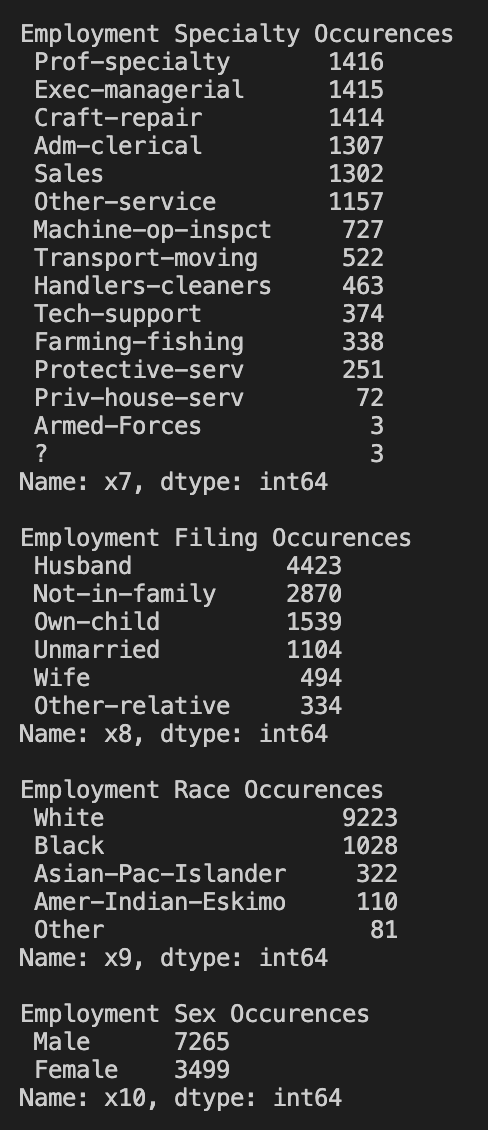
Arbitrary Code

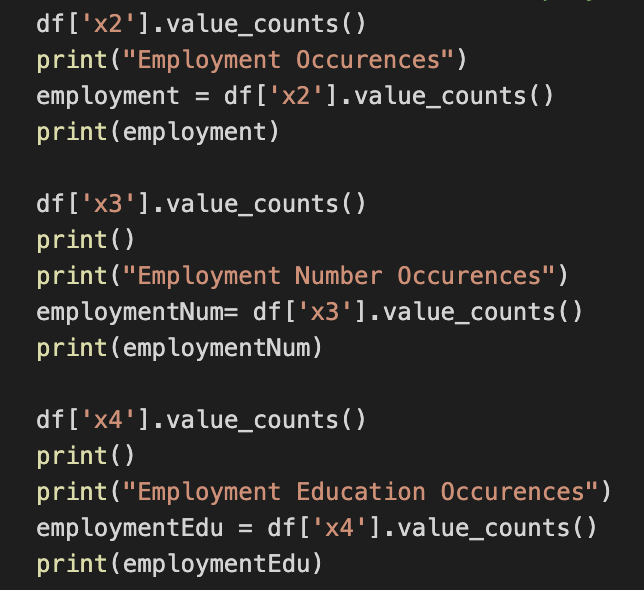


The last requirement needed to prepare the data was to split our csv file into two separate files of 70% and 30%. This was achieved by randomly choosing 70% of a file and then writing that and its composite into two new files.

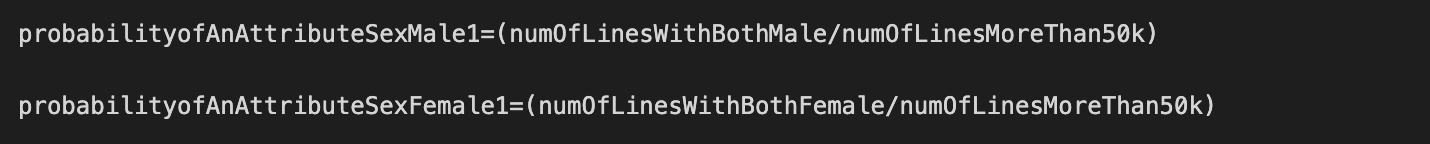
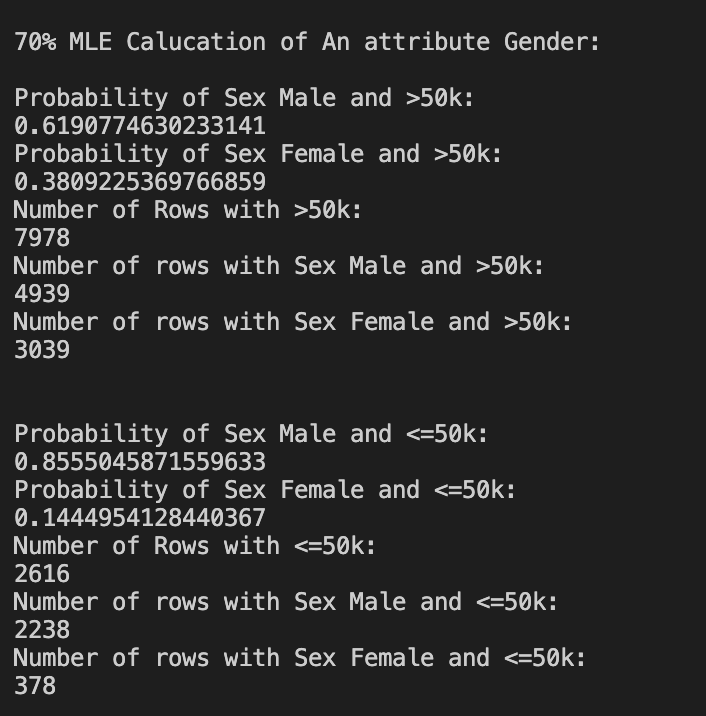
File Split Code - 70% and 30% respectively  


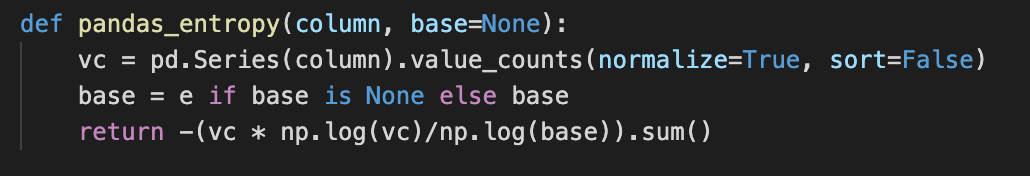
**Determining Occurrences**

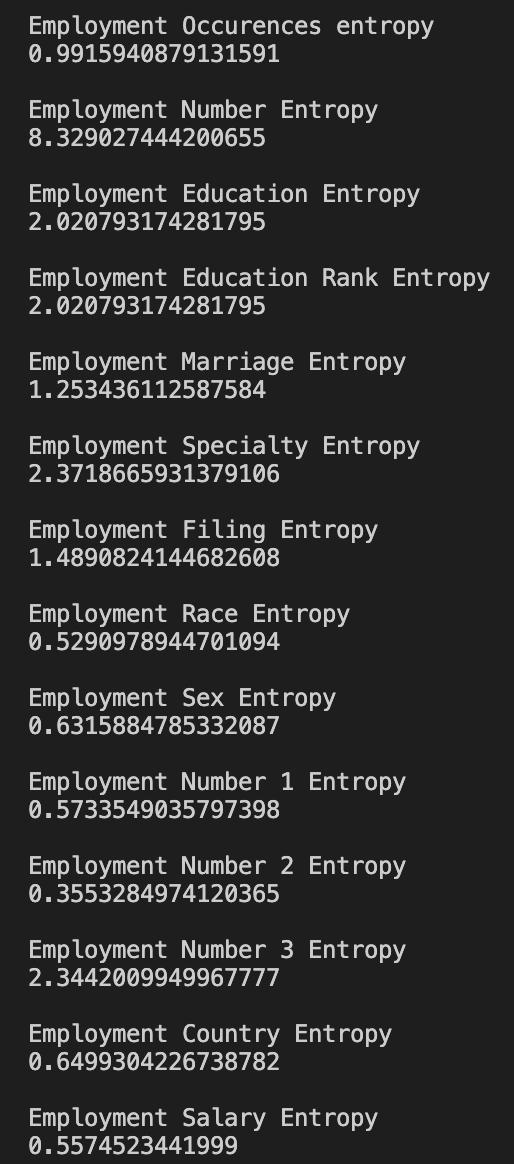
The second requirement of our project was to determine the number of occurrences for each unique human in every given column. This was achieved with the built-in python function value\_count() that reads a given column in a data instance and displays relevant information. 

Number of Occurrences Code & Output  


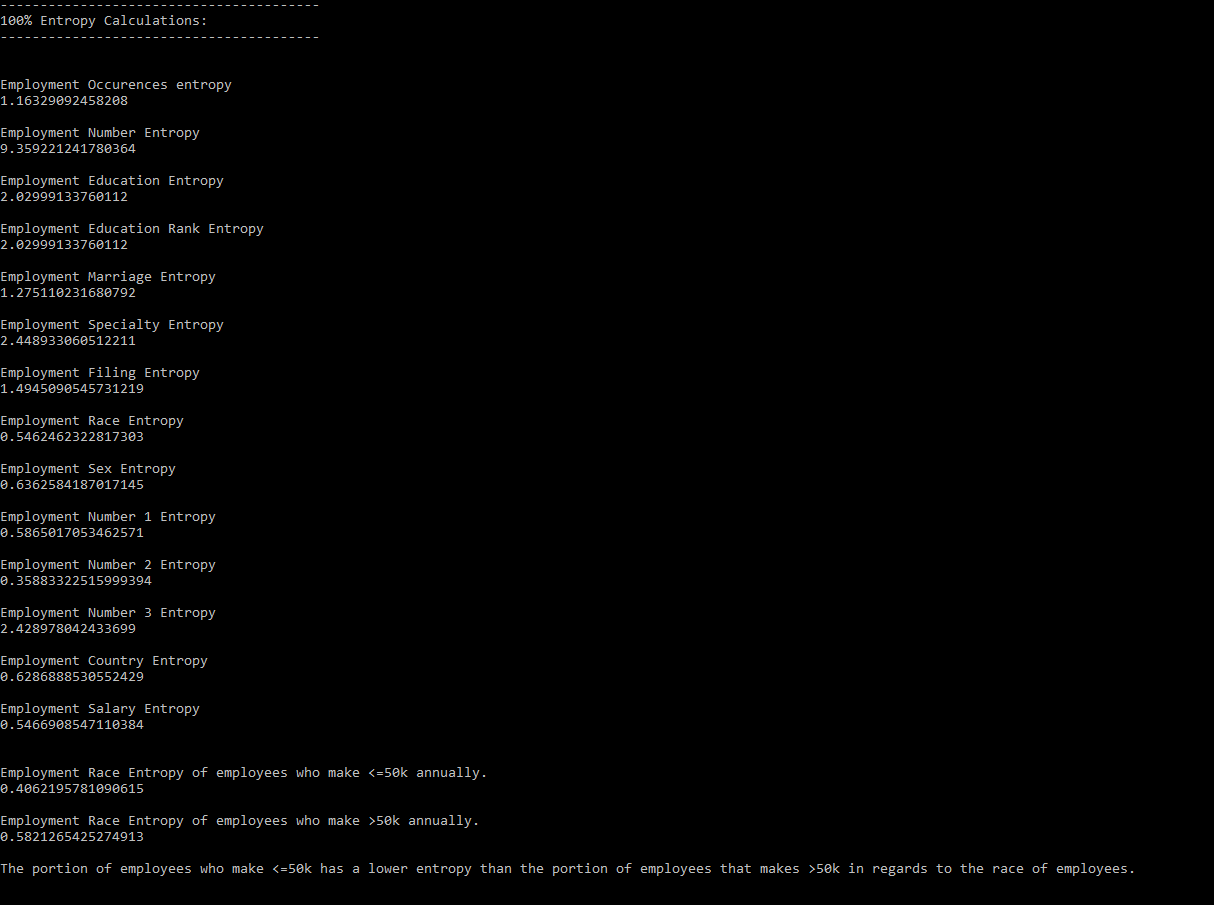
**Algorithmic Description and Implementation**

Once the edited file had been split, our first algorithmic implementation was to apply the Maximum Likelihood Estimation to the testing file based around humans sex and annual salary. This was achieved by running the data row by row through a series of if statements. If a particular row met a requirement, the corresponding count would be increased. Once the data instance had finished its run through these statements the counts were then applied to the MLE algorithm as the A and B variables.   
MLE Algorithm - Calculation - Using the MLE equation   
****MLE Output

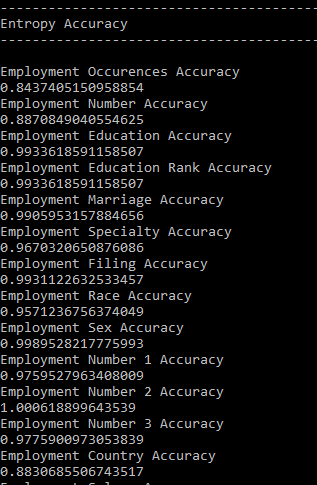
The second algorithm required the representation of an entropy calculation. This calculation was performed using pandas, a feature that can be installed in python3. Pandas is used to provide an easy and efficient data structure. In this case the structure stores the occurrences using *value\_counts*, however in this case the function calculates entropy using those stored occurences and returns that instead.   


Entropy Output - 30% dataset ****

**Implementation Results**The Entropy accuracy was calculated using the *Entropy Output* of the 100% and 30% dataset. The calculation was accomplished by using the entropy output of the 30% dataset and dividing it by the entropy output of the 100% dataset. This is important as it gives the user the accuracy of utilizing a portion of the dataset informing them whether or not the limited information at hand is reliable and efficient.

Entropy Output - 100% dataset ****

Entropy Accuracy - Calculated using the 30% dataset



**Pros**

* The program runs efficiently. Should in theory work with any given csv file.
* Information displayed is correct.

**Cons**

* Information is not displayed in a particularly aesthetic manner.
* Furthermore it can be quite difficult to find the particular information you are looking for once it has been displayed in a terminal.

**Further Improvement**

* Visual output of the information. The product could be neater and more visually appealing to the audience.
* An interactive UI So the information isn’t displayed all at once, this ties in with visual output.
* Design an instruction manual or guide for this program, for the user.