# **Part 1 – Data Preparation and Pre-processing [8 points]**

## 1) Describe the dataset.

### For example:

### What are the categories/domains of the dataset? Id, Title, Company, Date, Location, Area, Classification, Sub-classification, Requirement, Full Description, Lowest Salary, Highest Salary, Job Type.

### What is the dataset size of each variation?

### What is dataset structure/format?

### What are attributes/features of review data you are going to use?

### What are attributes/features of product data you are going to use?

### Which parts of the dataset will you use or all of them?

### [1-2 paragraphs, 3 points]

## This project will explore and analyse SEEK job market data. The dataset contains 318,477 data entries. It includes 13 categories: Id, Title, Company, Date, Location, Area, Classification, Sub-classification, Requirement, Full Description, Lowest Salary, Highest Salary, and Job Type. All categories were formatted as a string except for the *HighestSalary* and *LowestSalary* categories which were formatted as integers.

## Some parts of the dataset were excluded and other parts appended to improve analysis. Firstly, it was found that all Classification and Location NaN records were shared, and so these values were removed. This decreased the total data entries to 197,229. In addition, rows with a Sub-Classification value of ‘Other’ was removed. This is because ‘Other’ does not tell us anything useful about the data. It decreased the total data entries to 184,841. After this, the Id and Full Description columns were removed because they provided no useful information for analysis and created problems with duplicate data. The *Id* and *Full Description* revealed 1,039 duplicate entries. After deleting these duplicates, the total data entries were lowered to 183,802. Additionally, an *AvgSalary* column was appended. The column values are derived from the mean of the *HighestSalary* and *LowestSalary* columns. It was added to allow for an easier way to rank job salaries later in the analysis.

## 2) Describe the steps you used for data preparation and pre-processing.

### For example:

### How do you load the data using Pandas?

### How do you normalize the data?

### How do you clean the data?

### [2-3 paragraphs, 4 points]

Data preparation and pre-processing was done in a Jupyter notebook using the Pandas library. Because the dataset has been formatted as a CSV, the Pandas *read\_csv( <filename> )* function has been used. It should be noted that a *low\_memory* flag was used to fix importing issues caused by columns with mixed types. From this, we could read the CSV file into a Pandas DataFrame. The name of the DataFrame is *df* as it is short and an easy way of referring to the data.

Data cleaning was an important phase of the data preparation process. Firstly, we needed to get a big picture of the data by displaying its head. From this, we looked at how the data was formatted. It was noticed that the Dates column was formatted as an object type and was converted into a DateTime type. This was done to allow for easier manipulation of the dates. Next, we checked for NULL values. This gave us a clearer picture of the parts of the data that was missing. Fortunately, there were no NULL values for the LowestSalary and HighestSalary columns and so nothing was dropped. Additionally, it was found that 19.53% of jobs had a LowestSalary value of 0k, and 4.35% had a HighestSalary value of 999k (although none had both a LowestSalary of 0k and HighestSalary of 999k). These salaries appeared to be anomalies as no reasonable salary pays no nothing, nor do so many pay exactly 999k. In addition, the SEEK website allows job searches from 30k to 200k+. As such, to keep closer to these amounts 0k *LowestSalary* rows were converted to 15k and 999k *HighestSalary* rows were converted to 300k. Accounting for these anomalies will help reduce inaccurate skewing of the data. Moreover, salary values were scaled from 1 to 1,000. This allowed us to interpret the data as $1000 instead of 1k which was found to be clearer and less ambiguous. Furthermore, duplicate data removed. At first, there appeared to be no duplicate data. However, upon removing redundant Id and Full Description columns, 1039 duplicate entries were discovered and then removed. Lastly, to make the Title columns easier to read, any matching Area column that was found inside the title was removed. This affected 8,011 rows.

Normalization was also performed during data preparation. Normalisation is important as it allows us to standardise the metric data. To do this, the Sci-kit learn pre-processing library *StandardScaler* has been used. *StandardScaler* imposes a mean of 0 and a standard deviation of 1 onto the data. In application, we declared a StandardScaler object and then used it to fit and transformed our metric columns. This was then saved into a DataFrame called *df\_norm* – a clone of *df* but with normalized metric values.

## 3) What is your hypothesis (expectation) about the analysis outcome?

### [1-2 paragraphs, 1 point]

Some predictions have been made regarding the analysis outcomes:

* **Hypothesis 1:** It is believed … cities will have the most well-paying salaries. This is because ….
* **Hypothesis 2:** It is believed the hottest job sectors in each city will be …. This is because ….
* **Hypothesis 3:** It is believed there will be more jobs posted at the beginning of the month. This is because ….
* **Hypothesis 4:** It is believed jobs like … will pay the highest salary. This is because ….

# Not yet complete. These are the kinds of questions we should be exploring in the Analysis phase.

# **Part 2 – Data Analysis and Interpretation [17 points]**

## 1) Job metadata

First impressions about the dataset.

## 2) Locations

## 3) Sectors

## 3) What is your hypothesis (expectation) about the analysis outcome?