**Describe the steps for data preparation and preprocessing**

The dataset was loaded using the Pandas library with the following command:

import pandas as pd

df = pd.read\_csv("data\_v1.csv", engine="python", on\_bad\_lines='skip')

After loading, I verified the dataset structure:

* Checked the number of rows and columns using df.shape.
* Reviewed column names and their data types with df.info().
* Displayed basic statistics of numerical fields using df.describe().
* Listed all columns and checked for missing values using df.isnull().sum().

Instead of dropping rows with missing values, I applied a flexible missing data strategy:

* Retained important fields like Title, Company, Requirement, LowestSalary, and HighestSalary even if they had missing data, because their presence was still valuable for the analysis.
* Skipped direct analysis on fields with substantial missing data (Location, Area, Classification, and SubClassification), but kept them for potential inference tasks.

To enhance the dataset, I created a new column CompanyGroupID:

* This column was initially copied from the existing Company field.
* For rows where Company was missing, I applied an inference process:
  + Developed a mapping from the Location field to infer Australian states (such as NSW, VIC, QLD).
  + Created a helper function infer\_company() using Classification, SubClassification, and Requirement to guess the general company group if Company data was missing.
  + Updated CompanyGroupID accordingly based on job categories and state mappings and it can be applied instead of Company with lower missing values (Figure 1).

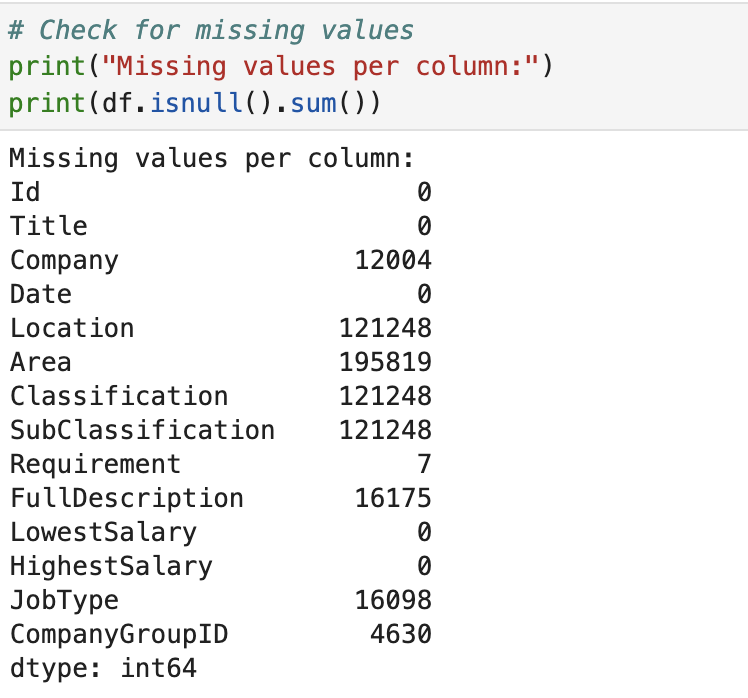


Figure Numbers of missing values after infer\_company function is applied

In terms of data normalization:

* Although I did not apply mathematical normalization such as Min-Max scaling, I performed data normalization in the form of textual and categorical standardization.
* This included converting inconsistent Location values into standard Australian state codes and binning the LowestSalary and HighestSalary columns into defined salary ranges.
* Salary values were categorized into 5 bins (ranges): 'Very Low' (0–39k), 'Low' (40–69k), 'Medium' (70–99k), 'High' (100–149k), and 'Very High' (150k+), to reduce the effect of extreme values and improve interpretability as shown in Figure 2.

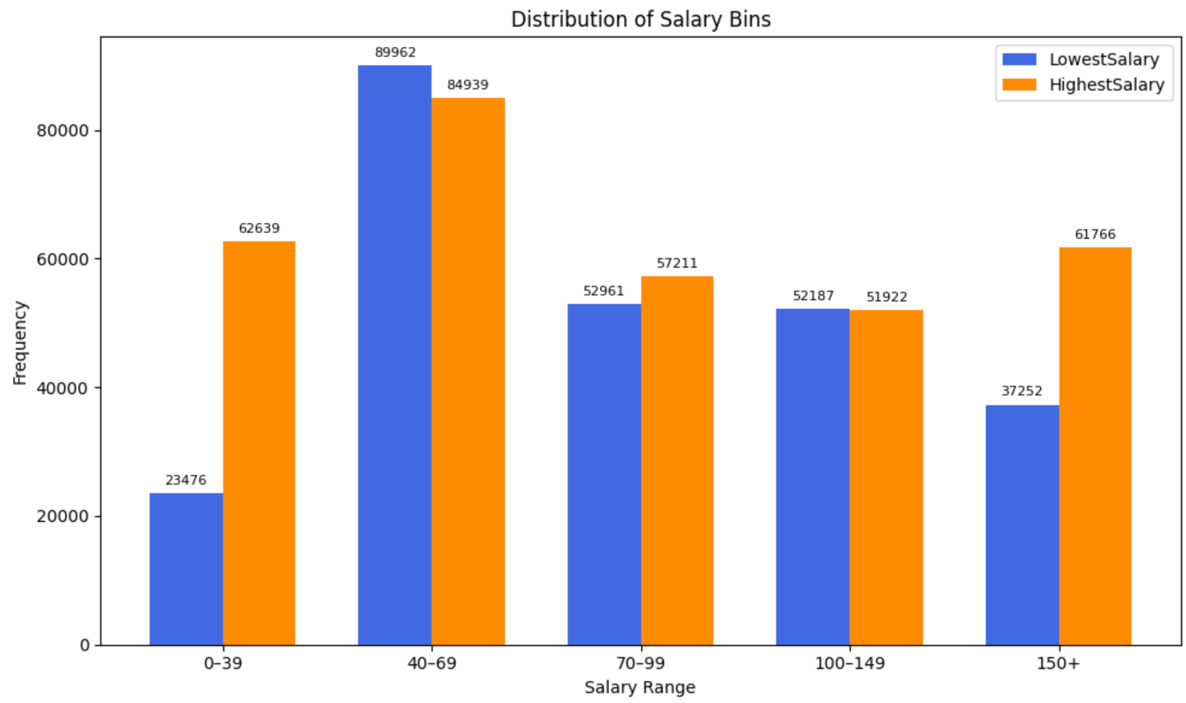


Figure 2 A salary bin chart shows a skewed distribution

To verify the salary distribution, I applied the Shapiro-Wilk test to the LowestSalary field:

* The resulting p-value was approximately 9.9e-127, which is significantly less than the standard threshold (p-value < 0.05).
* Therefore, I reject the null hypothesis and conclude that the salary data is not normally distributed.
* This finding is consistent with the salary bin chart, which shows a skewed distribution with a concentration around the 40–69k range and a long right tail. Most values are clustered on the lower end (between 30k–120k), but some values extend very far to the right, indicating a few extremely high values (almost all 999k).

Finally, the cleaned and enhanced dataset was saved into a new CSV file:

df.to\_csv('data\_withComGID.csv', index=False)

This processed dataset was then used for further analysis in Task 2.