

# DAT ENG 300 Homework 1

Sabian Atmadja (sna3362)

## Part 1 - Missing Data

### 1. Carrier (IATA Code)

The CARRIER column represents the IATA code used to identify airlines. Upon inspection, all aircraft operated by North American Airlines were missing carrier codes. However, a quick google search revealed that the actual IATA code is "NA", which was misinterpreted by the program as a missing value due to its resemblance to NaN. To resolve this, all missing entries in the CARRIER column were explicitly replaced with the string "NA" to prevent future misclassification.

### 2. Carrier Name (Airline Name)

CARRIER\_NAME is closely tied to the CARRIER column. L4 and OH were CARRIERS found to be missing associated carrier names:

- L4 maps to **Lynx Aviation**, and this was consistent across the dataset.
- OH has been historically used by two airlines:
  - **Comair Inc.** prior to 2012, and **PSA Airlines Inc.** from 2012 onward.
  - For years before 2012 → "Comair Inc.
  - For 2012 and after → "PSA Airlines Inc.

### 3. Manufacturer Year

MANUFACTURE\_YEAR is a per-aircraft attribute and does not exhibit a pattern or grouping that would support a reliable imputation strategy. Unlike other variables, manufacture year is not easily inferred from other aircraft features such as model or airline. Thus no imputation can be done for this variable.

### 4. Number of Seats

Aircraft with missing values in NUMBER\_OF\_SEATS were examined and found to correspond to cargo planes. Since these aircraft legitimately have zero seats, missing values were imputed with 0 to reflect their operational role without making unrealistic assumptions.

### 5. Capacity in Pounds

CAPACITY\_IN\_POUNDS varies by aircraft type, which is identified by AIRCRAFT\_TYPE. While there may be multiple configurations within the same type, the median capacity across each AIRCRAFT\_TYPE group was used to impute missing values. Median imputation was chosen to reduce the impact of outliers and better reflect typical capacity values within each aircraft type.

### 6. Airline ID

AIRLINE\_ID functions similarly to CARRIER\_NAME, linking each aircraft to a specific carrier's BTS reporting ID. Missing values were limited to aircraft associated with **PSA Airlines** and **Lynx Aviation**. These IDs were imputed as follows:

- PSA Airlines → 20397
- Lynx Aviation → 21217

## Part 2 - Data Transformation and Standardization

### 1. Manufacturers

Here we can see 2 issues: inconsistent capitalization, punctuation and naming conventions. To handle this, I did both standardization and transformation:

- Standardization - All upper case, No dashes or full stops
- Transformation - Removed the words "The", "Co", "Company", "Industry", "Industrie", "Aerospace". These words sometimes appear and sometimes don't but represent the same company (i.e. Boeing and Boeing Co.). McDonnellDouglas had 2 spellings (McDonnellDouglas and McDonnelDouglas) so I combined them under one name McDonnelDouglas.

### 2. Model

Here I can see a few issues, namely with standardization:

- Removed spaces, dashes and slashes.
- Made names all uppercase

This ensures the same aircraft is counted under one bucket (737-800 and 737800)

### 3. Operating Status

Here we can see the only discrepancy is that there are 2 y's, one upper case and one lower case. I simply made all the letters uppercase.

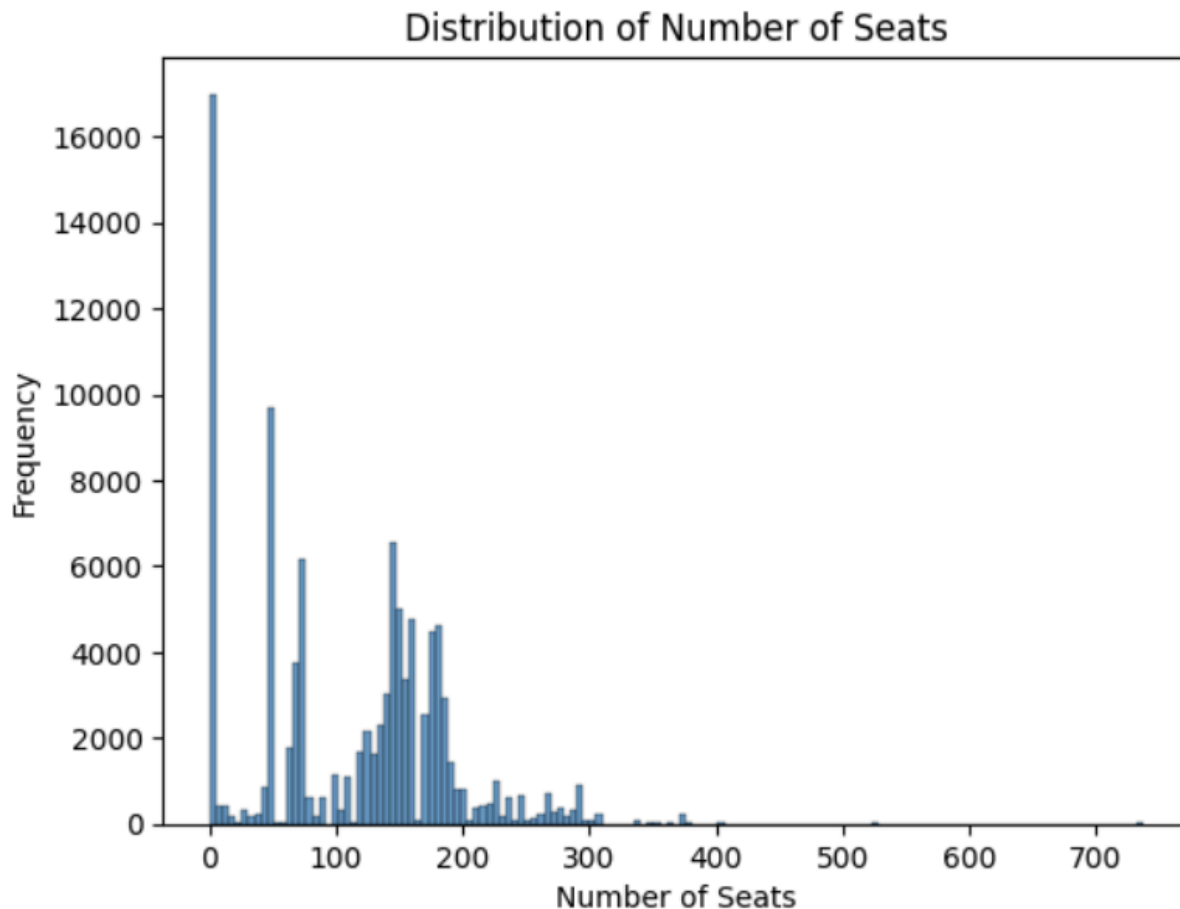
### 4. Aircraft Status

Here we can see the only discrepancy is that there are mixed upper and lowercase. I simply made all names uppercase so the same letters correspond to the same grouping.

## Part 3 - Number of Removed Data

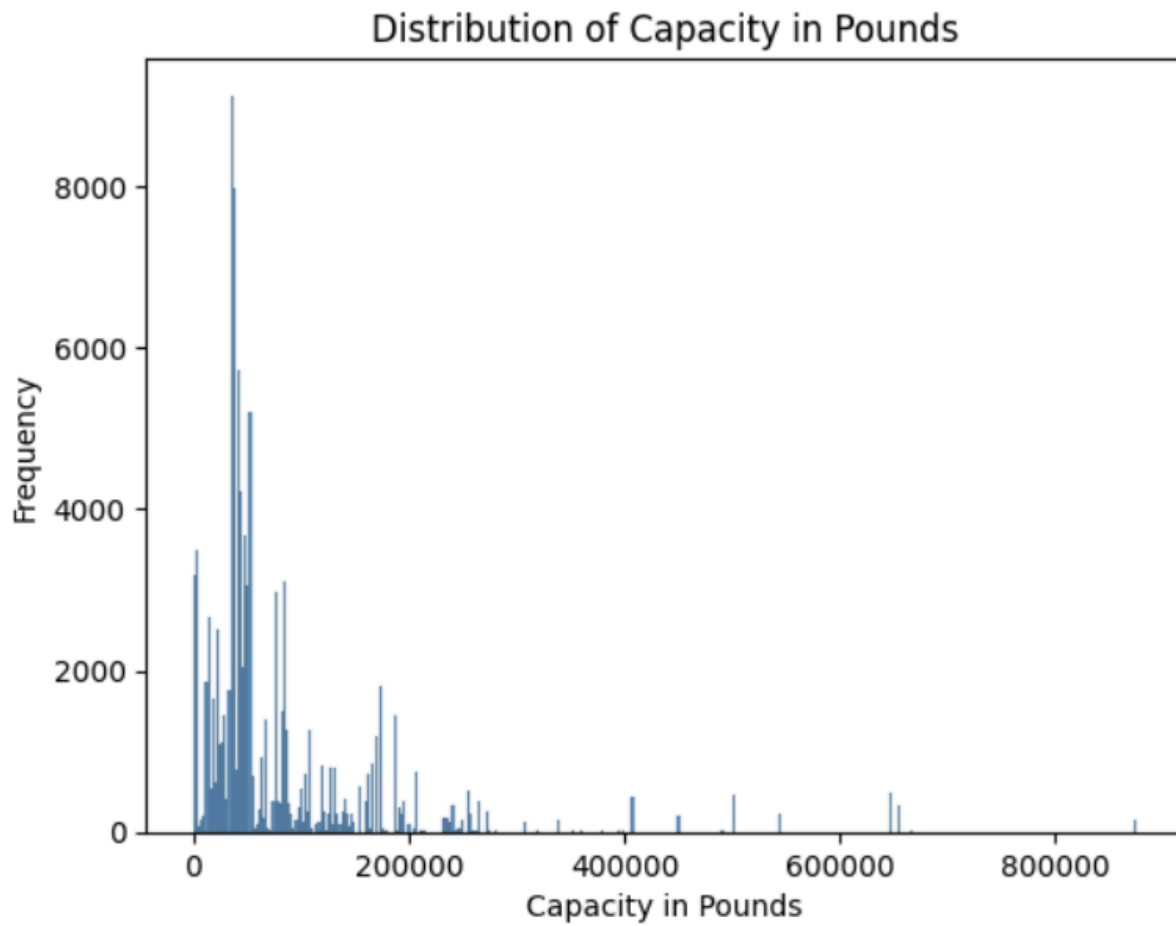
Removing all the rows with any missing data, we removed a total of 31038 out of the 132312 total data points. This represents around 23.4% of the total data

## Part 4 - Distribution Analysis



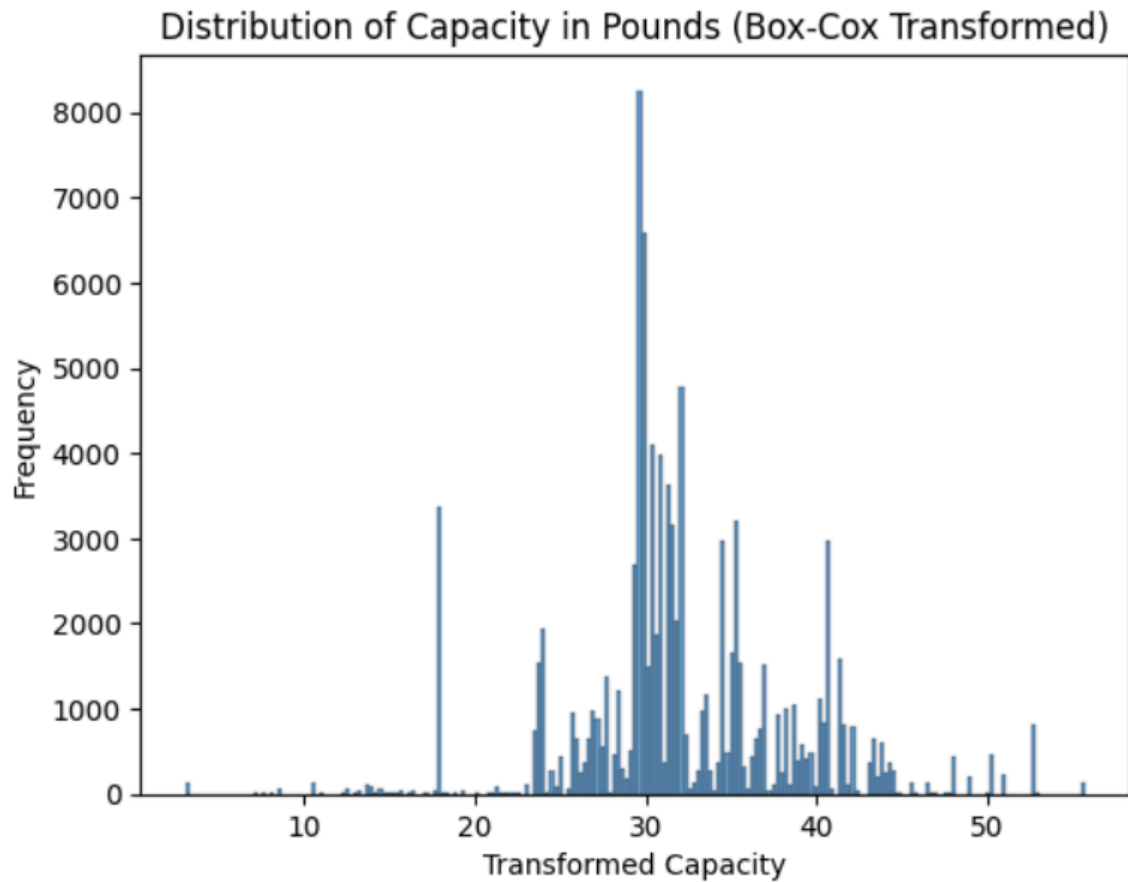
**Figure 1:** Graph showing the distribution of the number of seats

Here we can see that the graph is skewed to the right, with a skew of 0.378



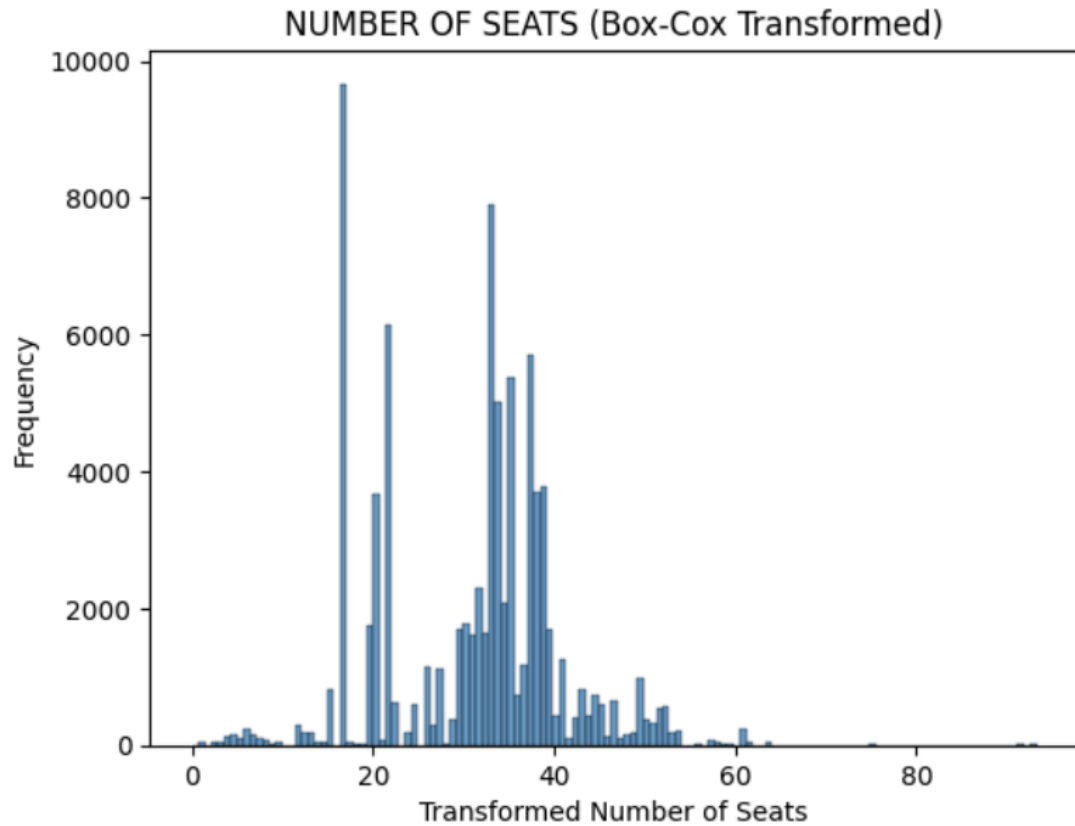
**Figure 2:** Graph showing the distribution of the capacity in pounds

Here we can also see the skew of the graph with a skew of 3.76.



**Figure 3:** Graph showing the distribution of the capacity in pounds with box cox transformation

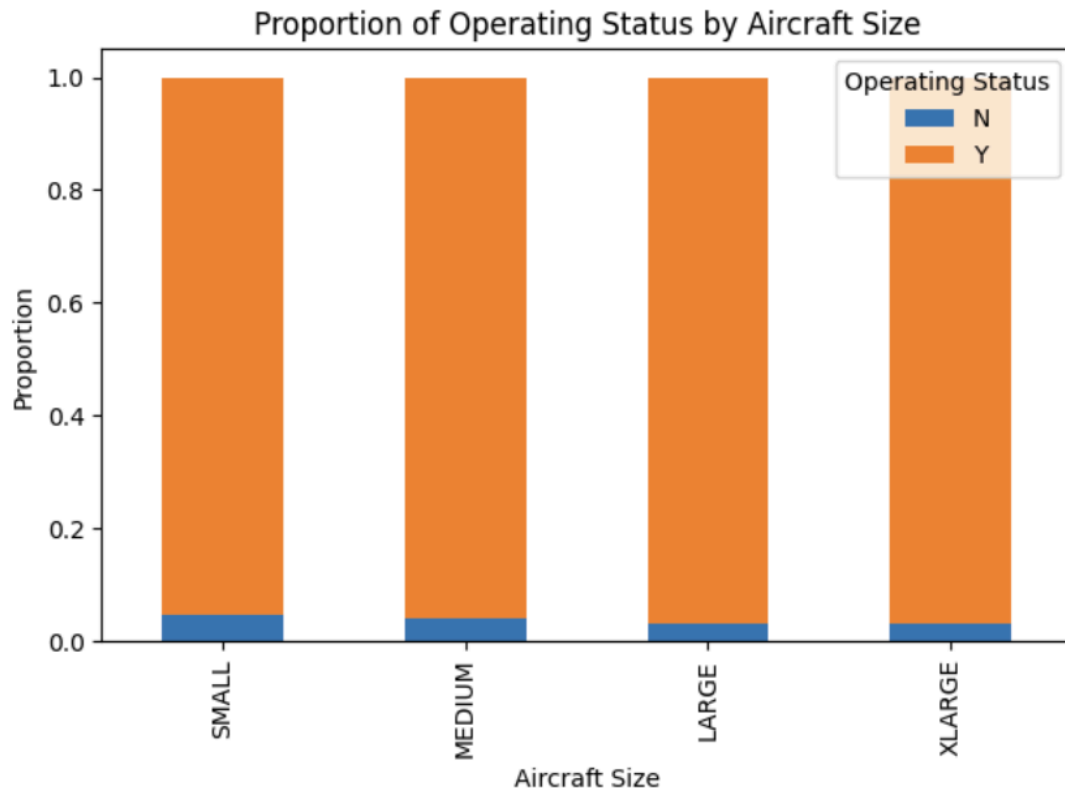
The graph now resembles that more of a normal distribution with much less skew at 0.118. Note that values of 0 were ignored as box cox can only take into account positive values.



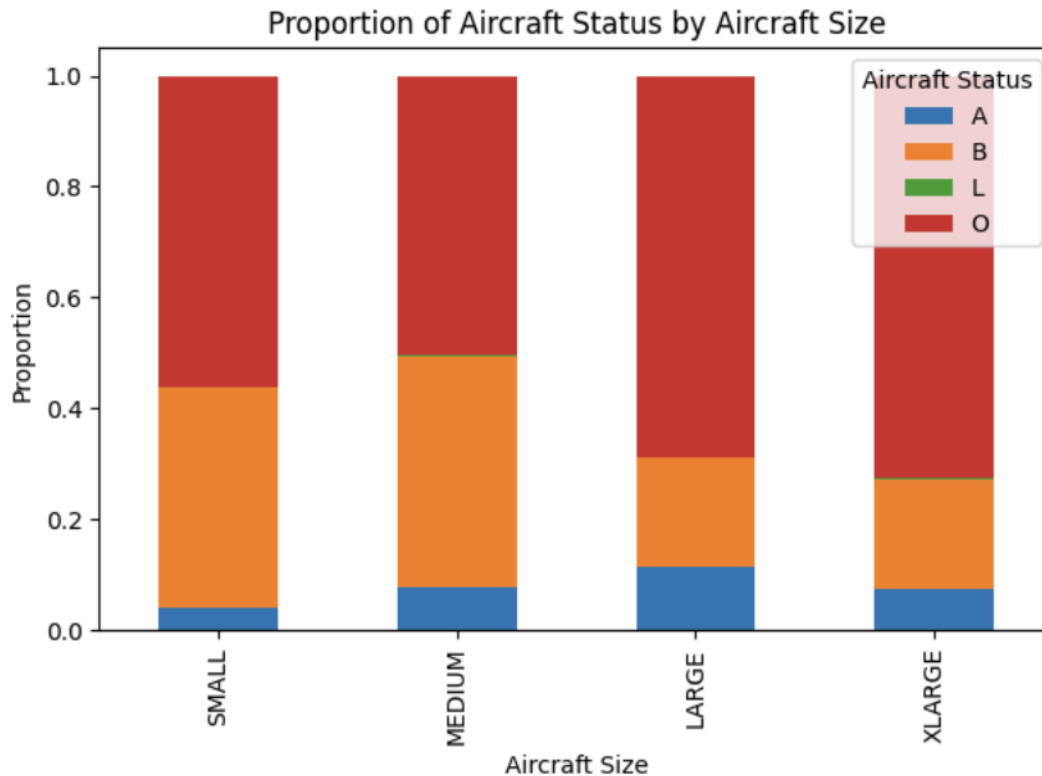
**Figure 4:** Graph showing the distribution of the number of seats with box cox transformation

The graph now resembles that more of a normal distribution with much less skew at -0.004. Note that values of 0 were ignored as box cox can only take into account positive values.

## Part 5 - Aircraft Size Analysis



**Figure 5:** Proportions of aircraft sizes with operating status



**Figure 6:** Proportions of aircraft sizes with aircraft status

Across all aircraft sizes, the proportion of operational aircraft (Y) is consistently high, with only a small share categorized as non-operational (N). This pattern holds steady from small to extra large, suggesting that most aircraft in the dataset remain in service regardless of size. In contrast, the distribution of aircraft status shows more variation. Small and medium aircraft have a larger proportion labeled O, while large and extra large aircraft have a more even split across A, B, and O, with a noticeable increase in A for larger sizes. This suggests differing usage or lifecycle trends based on aircraft size, particularly between large and extra large versus small and medium.



# Gen AI Disclosure

I asked ChatGPT how to reorder the bar chart. I was having difficulty on reordering the bar chart in order from smallest to largest.

Prompt:	Response:
<pre>op_status = clean_data.groupby(['SIZE', 'OPERATING_STATUS']).size().unstack() op_status_prop = op_status.div(op_status.sum(axis=1), axis=0) # Plot only the first two columns (assuming they correspond to valid statuses) op_status_prop.iloc[:, 1:3].plot(kind='bar', stacked=True, colormap='Set2') plt.title('Proportion of Operating Status by Aircraft Size') plt.ylabel('Proportion') plt.xlabel('Aircraft Size') plt.legend(title='Operating Status') plt.tight_layout() plt.show() can you help me  make the bar chart in order of small medium large xlarge please</pre>	<p>You're right to want consistency across both charts! Here's the modified code for the <b>AIRCRAFT_STATUS</b> plot, with aircraft sizes sorted in the order: <b>SMALL</b>, <b>MEDIUM</b>, <b>LARGE</b>, <b>XLARGE</b>:</p> <pre># Group and compute proportions op_status = clean_data.groupby(['SIZE', 'OPERATING_STATUS']).size().unstack()  # Define size order size_order = ['SMALL', 'MEDIUM', 'LARGE', 'XLARGE'] op_status = op_status.reindex(size_order)  # Normalize rows to get proportions op_status_prop = op_status.div(op_status.sum(axis=1), axis=0)  # Plot the chart op_status_prop.iloc[:, 1:3].plot(kind='bar', stacked=True) plt.title('Proportion of Operating Status by Aircraft Size') plt.ylabel('Proportion') plt.xlabel('Aircraft Size') plt.legend(title='Operating Status') plt.tight_layout() plt.show()</pre>