Documentation: Cleaning Company Census File (20260121)

Purpose:

The Company Census dataset originally contained over **2 million rows** and more than 150 columns, many of which were administrative, redundant, or too granular for analysis. The goal of this cleaning step was to reduce the dataset to a lean, consistent structure that preserves carrier identity, fleet metrics, safety/compliance fields, and cargo specialization, while minimizing memory usage and ensuring analytical readiness.

Steps Taken

1. Reading the raw dataset

```
df = pd.read_csv('Company_Census_File_20260121.csv', low_memory=False)
```

- low_memory=False ensures pandas scans entire columns before assigning dtypes, avoiding DtypeWarning from mixed types.

2. Dropping redundant columns

```
df = df.drop(columns=cols_to_drop)
```

- Removed administrative metadata (MCS150_DATE, ADD_DATE, REVIEW_ID, etc.).
- Dropped contact details (PHONE, FAX, EMAIL_ADDRESS, etc.).
- Eliminated hyper-granular ownership/transfer columns (OWNTRUCK, TRMTRUCK, TRPTRUCK, etc.).
- Removed duplicate mailing address fields (CARRIER_MAILING_*).
- This reduced the dataset to **core analytical fields only**.

3. Handling missing values

```
df = df.fillna('NaN')
```

```
df.replace('NaN', np.nan, inplace=True)
```

- Standardized missing values during cleaning, then converted placeholders to proper NaN for numeric/statistical analysis.

4. Downcasting numeric columns

```
df_int = df.select_dtypes(include=['int64'])
df[df_int.columns] = df_int.apply(pd.to_numeric, downcast='integer')

df_float = df.select_dtypes(include=['float64'])
df[df_float.columns] = df_float.apply(pd.to_numeric, downcast='float')
```

- Converted int64 → int32 and float64 → float32 to reduce memory footprint.
- Ensured fleet counts, driver totals, and mileage values use efficient numeric types.

5. Optimizing cargo flags

```
cargo_cols = [col for col in df.columns if col.startswith("CRGO_")]
df[cargo_cols] = df[cargo_cols].astype("category")
```

- Converted all cargo specialization columns (CRGO_GENFREIGHT, CRGO_MEAT, CRGO_CARGOOTHR, etc.) to category.
- This saves memory and enables faster filtering/grouping.

6. Validating uniqueness

```
print(df[['DOT_NUMBER']].duplicated().sum())
```

- Checked for duplicate carrier identifiers to ensure each DOT number is unique.

7. Profiling the dataset

```
print(df.describe(include='all'))
print(df.dtypes)
```

```
print(df.info(memory_usage='deep'))
```

- Generated descriptive statistics and memory usage report to confirm data integrity and efficiency.

8. Saving the cleaned dataset

```
df.to_csv('Cleaned_Company_Census_File_20260121.csv', index=False)
```

- Exported the final cleaned dataset for downstream analysis.

Key Decisions and Rationale

- Dropped 50+ non-analytical columns to focus on carrier identity, fleet size, safety, and cargo specialization.
- Downcasted numeric types to reduce memory usage by ~30–50%.
- Converted cargo flags to categories for efficient filtering and grouping.
- Standardized missing values as NaN to ensure compatibility with pandas/numpy operations.
- Validated uniqueness of DOT_NUMBER to maintain dataset integrity.

Data Source: [Quarterly Census of Employment and Wages : U.S. Bureau of Labor Statistics]