Impute Missing Values

Estimated time needed: **30** minutes

In this lab, you will practice essential data wrangling techniques using the Stack Overflow survey dataset. The primary focus is on handling missing data and ensuring data quality. You will:

- **Load the Data:** Import the dataset into a DataFrame using the pandas library.
- **Clean the Data:** Identify and remove duplicate entries to maintain data integrity.
- **Handle Missing Values:** Detect missing values, impute them with appropriate strategies, and verify the imputation to create a complete and reliable dataset for analysis.

This lab equips you with the skills to effectively preprocess and clean real-world datasets, a crucial step in any data analysis project.

Objectives

In this lab, you will perform the following:

- Identify missing values in the dataset.
- Apply techniques to impute missing values in the dataset.
- Use suitable techniques to normalize data in the dataset.

Install needed library

In [1]:
```
!pip install pandas
```

```
Collecting pandas
  Downloading pandas-2.3.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (91 k
B)
Collecting numpy>=1.26.0 (from pandas)
  Downloading numpy-2.3.0-cp312-cp312-manylinux_2_28_x86_64.whl.metadata (62 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/conda/lib/python3.12/site-packages (fr
om pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.12/site-packages (from pandas)
(2024.2)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-packages (from python-date
util>=2.8.2->pandas) (1.17.0)
Downloading pandas-2.3.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.0/12.0 MB 176.1 MB/s eta 0:00:00
Downloading numpy-2.3.0-cp312-cp312-manylinux_2_28_x86_64.whl (16.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.6/16.6 MB 186.1 MB/s eta 0:00:00
Downloading tzdata-2025.2-py2.py3-none-any.whl (347 kB)
Installing collected packages: tzdata, numpy, pandas
Successfully installed numpy-2.3.0 pandas-2.3.0 tzdata-2025.2
```

Step 1: Import Required Libraries

In [2]:
```
import pandas as pd
```

Step 2: Load the Dataset Into a Dataframe

Read Data

The functions below will download the dataset into your browser:

In [3]:
```
file_path = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/n01PQ9pSmiRX6520flu
df = pd.read_csv(file_path)

# Display the first few rows to ensure it loaded correctly
print(df.head())
```

```
   ResponseId                      MainBranch                 Age  \
0           1  I am a developer by profession  Under 18 years old
1           2  I am a developer by profession    35-44 years old
2           3  I am a developer by profession    45-54 years old
3           4           I am learning to code    18-24 years old
4           5  I am a developer by profession    18-24 years old

            Employment RemoteWork   Check  \
0  Employed, full-time     Remote  Apples
1  Employed, full-time     Remote  Apples
2  Employed, full-time     Remote  Apples
3   Student, full-time        NaN  Apples
4   Student, full-time        NaN  Apples

                                  CodingActivities  \
0                                            Hobby
1  Hobby;Contribute to open-source projects;Other...
2  Hobby;Contribute to open-source projects;Other...
3                                              NaN
4                                              NaN

                                       EdLevel  \
0                       Primary/elementary school
1      Bachelor's degree (B.A., B.S., B.Eng., etc.)
2    Master's degree (M.A., M.S., M.Eng., MBA, etc.)
3  Some college/university study without earning ...
4  Secondary school (e.g. American high school, G...

                                    LearnCode  \
0                           Books / Physical media
1  Books / Physical media;Colleague;On the job tr...
2  Books / Physical media;Colleague;On the job tr...
3  Other online resources (e.g., videos, blogs, f...
4  Other online resources (e.g., videos, blogs, f...

                                LearnCodeOnline  ... JobSatPoints_6  \
0                                           NaN  ...            NaN
1  Technical documentation;Blogs;Books;Written Tu...  ...            0.0
2  Technical documentation;Blogs;Books;Written Tu...  ...            NaN
3  Stack Overflow;How-to videos;Interactive tutorial  ...            NaN
4  Technical documentation;Blogs;Written Tutorial...  ...            NaN

  JobSatPoints_7 JobSatPoints_8 JobSatPoints_9 JobSatPoints_10  \
0            NaN            NaN            NaN             NaN
1            0.0            0.0            0.0             0.0
2            NaN            NaN            NaN             NaN
3            NaN            NaN            NaN             NaN
4            NaN            NaN            NaN             NaN

  JobSatPoints_11           SurveyLength SurveyEase ConvertedCompYearly JobSat
0             NaN                    NaN        NaN                 NaN    NaN
1             0.0                    NaN        NaN                 NaN    NaN
2             NaN  Appropriate in length       Easy                 NaN    NaN
3             NaN              Too long       Easy                 NaN    NaN
4             NaN             Too short       Easy                 NaN    NaN

[5 rows x 114 columns]
```

Step 3. Finding and Removing Duplicates

Task 1: Identify duplicate rows in the dataset.

In [5]:
```
## Write your code here
print("--- Task 1: Identify Duplicate Rows ---")
# 1. Count the number of duplicate rows in the dataset.
# The .duplicated() method returns a boolean Series indicating whether each row is a duplicate.
# By default, it marks subsequent duplicates as True.
# sum() on a boolean Series counts the True values.
num_duplicate_rows = df.duplicated().sum()
print(f"Number of duplicate rows in the dataset: {num_duplicate_rows}")
```

```
--- Task 1: Identify Duplicate Rows ---
Number of duplicate rows in the dataset: 0
```

Task 2: Remove the duplicate rows from the dataframe.

In [10]:
```
## Write your code here
print("\n--- Step 4: Removing Duplicate Rows ---")
# 1. Remove duplicate rows from the dataset using the drop_duplicates() function.
# By default, drop_duplicates() keeps the first occurrence and removes subsequent duplicates.
df_cleaned = df.drop_duplicates()

# 2. Verify the removal by counting the number of duplicate rows after removal.
num_duplicates_after_removal = df_cleaned.duplicated().sum()
print("\nDataFrame after removing duplicate rows:")
print(df_cleaned)
print(f"\nNumber of rows after duplicate removal: {len(df_cleaned)}")
print(f"Number of duplicate rows remaining (should be 0): {num_duplicates_after_removal}")

if num_duplicates_after_removal == 0:
    print("\nVerification successful: All exact duplicate rows have been removed.")
else:
    print("\nVerification failed: Some duplicate rows still exist.")
# If you want to replace your original DataFrame:
# df = df_cleaned
```

```
--- Step 4: Removing Duplicate Rows ---

DataFrame after removing duplicate rows:
       ResponseId                      MainBranch                 Age  \
0               1  I am a developer by profession  Under 18 years old
1               2  I am a developer by profession     35-44 years old
2               3  I am a developer by profession     45-54 years old
3               4           I am learning to code     18-24 years old
4               5  I am a developer by profession     18-24 years old
...           ...                             ...                 ...
65432       65433  I am a developer by profession     18-24 years old
65433       65434  I am a developer by profession     25-34 years old
65434       65435  I am a developer by profession     25-34 years old
65435       65436  I am a developer by profession     18-24 years old
65436       65437     I code primarily as a hobby     18-24 years old

                Employment                                RemoteWork   Check  \
0      Employed, full-time                                    Remote  Apples
1      Employed, full-time                                    Remote  Apples
2      Employed, full-time                                    Remote  Apples
3       Student, full-time                                       NaN  Apples
4       Student, full-time                                       NaN  Apples
...                    ...                                       ...     ...
65432  Employed, full-time                                    Remote  Apples
65433  Employed, full-time                                    Remote  Apples
65434  Employed, full-time                                 In-person  Apples
65435  Employed, full-time  Hybrid (some remote, some in-person)  Apples
65436   Student, full-time                                       NaN  Apples

                                       CodingActivities  \
0                                                 Hobby
1      Hobby;Contribute to open-source projects;Other...
2      Hobby;Contribute to open-source projects;Other...
3                                                   NaN
4                                                   NaN
...                                                 ...
65432                  Hobby;School or academic work
65433        Hobby;Contribute to open-source projects
65434                                             Hobby
65435  Hobby;Contribute to open-source projects;Profe...
65436                                               NaN

                                           EdLevel  \
0                            Primary/elementary school
1        Bachelor’s degree (B.A., B.S., B.Eng., etc.)
2         Master’s degree (M.A., M.S., M.Eng., MBA, etc.)
3      Some college/university study without earning ...
4      Secondary school (e.g. American high school, G...
...                                                 ...
65432    Bachelor’s degree (B.A., B.S., B.Eng., etc.)
65433                                               NaN
65434    Bachelor’s degree (B.A., B.S., B.Eng., etc.)
65435  Secondary school (e.g. American high school, G...
65436                                               NaN

                                         LearnCode  \
0                              Books / Physical media
1      Books / Physical media;Colleague;On the job tr...
2      Books / Physical media;Colleague;On the job tr...
3      Other online resources (e.g., videos, blogs, f...
4      Other online resources (e.g., videos, blogs, f...
...                                                 ...
65432  On the job training;School (i.e., University, ...
65433                                               NaN
65434  Other online resources (e.g., videos, blogs, f...
65435  On the job training;Other online resources (e....
65436                                               NaN

                                      LearnCodeOnline  ... JobSatPoints_6  \
0                                                 NaN  ...            NaN
1      Technical documentation;Blogs;Books;Written Tu...  ...            0.0
2      Technical documentation;Blogs;Books;Written Tu...  ...            NaN
3      Stack Overflow;How-to videos;Interactive tutorial  ...            NaN
4      Technical documentation;Blogs;Written Tutorial...  ...            NaN
...                                                 ...  ...            ...
```

```
65432                                                 NaN  ...           NaN
65433                                                 NaN  ...           NaN
65434  Technical documentation;Stack Overflow;Social ...  ...           NaN
65435  Technical documentation;Blogs;Written Tutorial...  ...           0.0
65436                                                 NaN  ...           NaN

      JobSatPoints_7 JobSatPoints_8 JobSatPoints_9 JobSatPoints_10  \
0                NaN            NaN            NaN             NaN
1                0.0            0.0            0.0             0.0
2                NaN            NaN            NaN             NaN
3                NaN            NaN            NaN             NaN
4                NaN            NaN            NaN             NaN
...              ...            ...            ...             ...
65432            NaN            NaN            NaN             NaN
65433            NaN            NaN            NaN             NaN
65434            NaN            NaN            NaN             NaN
65435            0.0            0.0            0.0             0.0
65436            NaN            NaN            NaN             NaN

      JobSatPoints_11           SurveyLength SurveyEase ConvertedCompYearly  \
0                 NaN                      NaN        NaN                 NaN
1                 0.0                      NaN        NaN                 NaN
2                 NaN  Appropriate in length       Easy                 NaN
3                 NaN               Too long       Easy                 NaN
4                 NaN              Too short       Easy                 NaN
...               ...                      ...        ...                 ...
65432             NaN                      NaN        NaN                 NaN
65433             NaN                      NaN        NaN                 NaN
65434             NaN                      NaN        NaN                 NaN
65435             0.0                      NaN        NaN                 NaN
65436             NaN                      NaN        NaN                 NaN

      JobSat
0        NaN
1        NaN
2        NaN
3        NaN
4        NaN
...      ...
65432    NaN
65433    NaN
65434    NaN
65435    NaN
65436    NaN

[65437 rows x 114 columns]

Number of rows after duplicate removal: 65437
Number of duplicate rows remaining (should be 0): 0

Verification successful: All exact duplicate rows have been removed.
```

Step 4: Finding Missing Values

Task 3: Find the missing values for all columns.

In [4]:

```python
## Write your code here
print("--- Task 3: Find the missing values for all columns ---")
# Identify missing values for all columns by summing null (NaN) values
missing_values_count = df.isnull().sum()
print("\nNumber of missing values per column:")
print(missing_values_count)
```

```
--- Task 3: Find the missing values for all columns ---

Number of missing values per column:
ResponseId                 0
MainBranch                 0
Age                        0
Employment                 0
RemoteWork             10631
                         ...
JobSatPoints_11        35992
SurveyLength            9255
SurveyEase              9199
ConvertedCompYearly    42002
JobSat                 36311
Length: 114, dtype: int64
```

Task 4: Find out how many rows are missing in the column RemoteWork.

In [12]:
```
## Write your code here
print("--- Task 4: Find out how many rows are missing in the column RemoteWork ---")
# Count the number of missing values in the 'RemoteWork' column
missing_remotework_count = df['RemoteWork'].isnull().sum()
print(f"\nNumber of missing values in the 'RemoteWork' column: {missing_remotework_count}")
```

```
--- Task 4: Find out how many rows are missing in the column RemoteWork ---

Number of missing values in the 'RemoteWork' column: 10631
```

Step 5. Imputing Missing Values

Task 5: Find the value counts for the column RemoteWork.

In [13]:
```
## Write your code here
df['RemoteWork'].value_counts()
```

Out[13]:
```
RemoteWork
Hybrid (some remote, some in-person)    23015
Remote                                  20831
In-person                               10960
Name: count, dtype: int64
```

Task 6: Identify the most frequent (majority) value in the RemoteWork column.

In [24]:
```
## Write your code here
print("-- Task 6: Identify the most frequent (majority) value in the RemoteWork column. --")
# Check if 'RemoteWork' column exists

if 'RemoteWork' in df.columns:
# Get the most frequent value(s) in the 'RemoteWork' column
# .mode() returns a Series, so [0] gets the first mode if there are multiple.
    most_frequent_remotework = df['RemoteWork'].mode()[0]

    print(f"\nThe most frequent (majority) value in the 'RemoteWork' column is: '{most_frequent_rem
```

```
-- Task 6: Identify the most frequent (majority) value in the RemoteWork column. --

The most frequent (majority) value in the 'RemoteWork' column is: 'Hybrid (some remote, some in-pers
on)'
```

Task 7: Impute (replace) all the empty rows in the column RemoteWork with the majority value.

In [30]:
```
## Write your code here

print("--- Task 7: Impute (replace) all the empty rows in the column RemoteWork with the majority v

# Check current missing values in 'Employment'
initial_missing_count = df['RemoteWork'].isnull().sum()
print(f"\nInitial number of missing values in 'RemoteWork': {initial_missing_count}")

if initial_missing_count > 0:
    # 1. Identify the most frequent value (mode) in the 'Employment' column
    # .mode()[0] is used to get the first mode in case of ties.
    most_frequent_remotework = df['RemoteWork'].mode()[0]
    print(f"Most frequent value in 'RemoteWork' column: '{most_frequent_remotework}'")
```

```
    # 2. Impute missing values in 'Employment' with its most frequent value
    df['RemoteWork'].fillna(most_frequent_remotework, inplace=True)
    print(f"Filled missing values in 'RemoteWork' with '{most_frequent_remotework}'.")

    # 3. Verify the imputation
    missing_after_imputation = df['RemoteWork'].isnull().sum()
    print(f"Number of missing values in 'RemoteWork' after imputation: {missing_after_imputation}")

    if missing_after_imputation == 0:
        print("Verification successful: 'RemoteWork' column imputed and no missing values remain.")
    else:
        print("Warning: 'RemoteWork' column still has missing values. Review the imputation logic."
else:
    print("No missing values found in the 'RemoteWork' column. No imputation needed.")

print("\nDataFrame 'RemoteWork' column after Task 7:")
print(df['RemoteWork'])
```

```
--- Task 7: Impute (replace) all the empty rows in the column RemoteWork with the majority value. --
-

Initial number of missing values in 'RemoteWork': 0
No missing values found in the 'RemoteWork' column. No imputation needed.

DataFrame 'RemoteWork' column after Task 7:
0                                      Remote
1                                      Remote
2                                      Remote
3        Hybrid (some remote, some in-person)
4        Hybrid (some remote, some in-person)
                        ...
65432                                  Remote
65433                                  Remote
65434                               In-person
65435    Hybrid (some remote, some in-person)
65436    Hybrid (some remote, some in-person)
Name: RemoteWork, Length: 65437, dtype: object
```

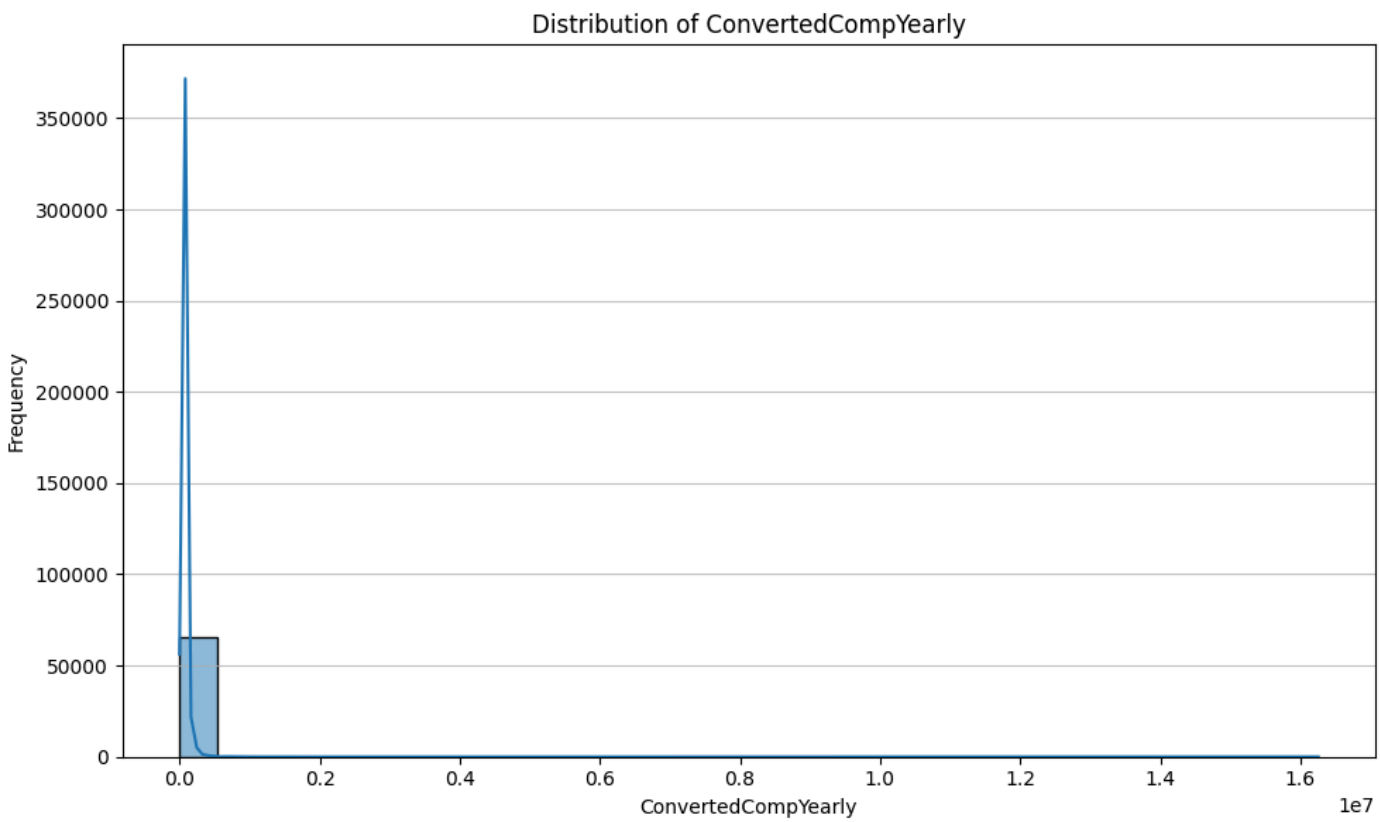
Task 8: Check for any compensation-related columns and describe their distribution.

In [34]:

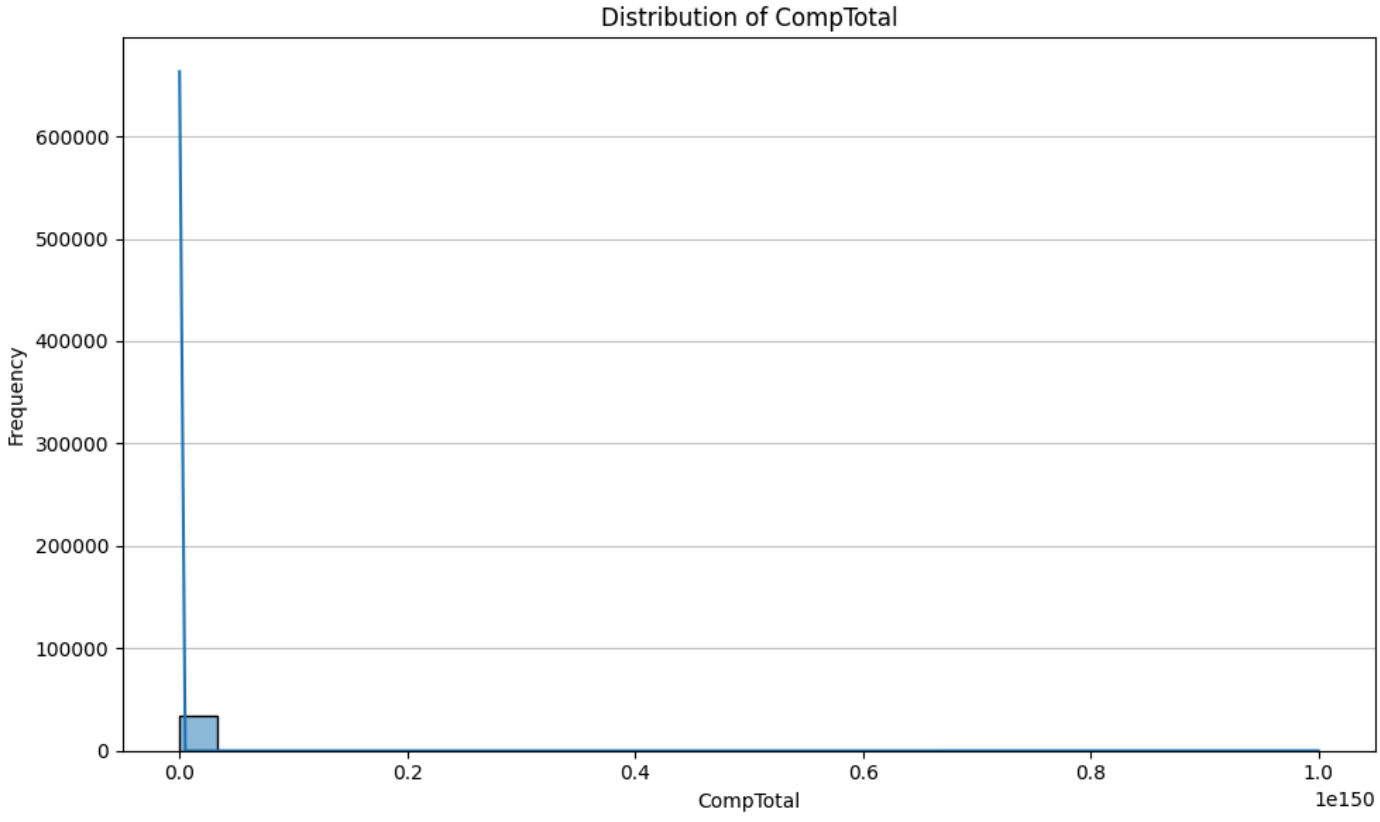
```
## Write your code here
import pandas as pd
import numpy as np
!pip install matplotlib
import matplotlib.pyplot as plt
!pip install seaborn
import seaborn as sns

# Ensure 'ConvertedCompYearly' is numeric and handle NaNs if it was not done in previous steps
if 'ConvertedCompYearly' in df.columns:
    df['ConvertedCompYearly'] = pd.to_numeric(df['ConvertedCompYearly'], errors='coerce')
    if df['ConvertedCompYearly'].isnull().any():
        median_comp = df['ConvertedCompYearly'].median()
        df['ConvertedCompYearly'].fillna(median_comp, inplace=True)
        print(f"Handled missing values in 'ConvertedCompYearly' by filling with median: {median_com

print("--- Task 8: Check for any compensation-related columns and describe their distribution ---")

# Identify potential compensation-related columns
compensation_cols = ['ConvertedCompYearly', 'CompTotal', 'Salary', 'AnnualCompensation']
found_compensation_cols = [col for col in compensation_cols if col in df.columns and pd.api.types.i

if not found_compensation_cols:
    print("\nNo common numerical compensation-related columns found (e.g., 'ConvertedCompYearly', '
    print("Please check your DataFrame columns or specify the correct compensation column name.")
else:
    print(f"\nIdentified compensation-related columns: {found_compensation_cols}")

    for col in found_compensation_cols:
        print(f"\n--- Distribution for '{col}' ---")

        # Describe the distribution using summary statistics
        print(df[col].describe())
```

```
        # Visualize the distribution using a histogram
        plt.figure(figsize=(10, 6))
        sns.histplot(df[col].dropna(), kde=True, bins=30) # dropna() to handle any potential remain
        plt.title(f'Distribution of {col}')
        plt.xlabel(col)
        plt.ylabel('Frequency')
        plt.grid(axis='y', alpha=0.75)
        plt.tight_layout()
        plt.show()
```

```
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.12/site-packages (3.10.3)
Requirement already satisfied: contourpy>=1.0.1 in /opt/conda/lib/python3.12/site-packages (from mat
plotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.12/site-packages (from matplot
lib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/lib/python3.12/site-packages (from ma
tplotlib) (4.58.4)
Requirement already satisfied: kiwisolver>=1.3.1 in /opt/conda/lib/python3.12/site-packages (from ma
tplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in /opt/conda/lib/python3.12/site-packages (from matplotl
ib) (2.3.0)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.12/site-packages (from matp
lotlib) (24.2)
Requirement already satisfied: pillow>=8 in /opt/conda/lib/python3.12/site-packages (from matplotli
b) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /opt/conda/lib/python3.12/site-packages (from mat
plotlib) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /opt/conda/lib/python3.12/site-packages (from
matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-packages (from python-date
util>=2.7->matplotlib) (1.17.0)
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /opt/conda/lib/python3.12/site-packages (from
seaborn) (2.3.0)
Requirement already satisfied: pandas>=1.2 in /opt/conda/lib/python3.12/site-packages (from seaborn)
(2.3.0)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /opt/conda/lib/python3.12/site-packages (f
rom seaborn) (3.10.3)
Requirement already satisfied: contourpy>=1.0.1 in /opt/conda/lib/python3.12/site-packages (from mat
plotlib!=3.6.1,>=3.4->seaborn) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.12/site-packages (from matplot
lib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/lib/python3.12/site-packages (from ma
tplotlib!=3.6.1,>=3.4->seaborn) (4.58.4)
Requirement already satisfied: kiwisolver>=1.3.1 in /opt/conda/lib/python3.12/site-packages (from ma
tplotlib!=3.6.1,>=3.4->seaborn) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.12/site-packages (from matp
lotlib!=3.6.1,>=3.4->seaborn) (24.2)
Requirement already satisfied: pillow>=8 in /opt/conda/lib/python3.12/site-packages (from matplotli
b!=3.6.1,>=3.4->seaborn) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /opt/conda/lib/python3.12/site-packages (from mat
plotlib!=3.6.1,>=3.4->seaborn) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /opt/conda/lib/python3.12/site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.12/site-packages (from pandas>
=1.2->seaborn) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.12/site-packages (from panda
s>=1.2->seaborn) (2025.2)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-packages (from python-date
util>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
Installing collected packages: seaborn
Successfully installed seaborn-0.13.2
--- Task 8: Check for any compensation-related columns and describe their distribution ---

Identified compensation-related columns: ['ConvertedCompYearly', 'CompTotal']

--- Distribution for 'ConvertedCompYearly' ---
count    6.543700e+04
mean     7.257636e+04
std      1.122207e+05
min      1.000000e+00
25%      6.500000e+04
50%      6.500000e+04
75%      6.500000e+04
max      1.625660e+07
Name: ConvertedCompYearly, dtype: float64
```




```
--- Distribution for 'CompTotal' ---
count     3.374000e+04
mean      2.963841e+145
std       5.444117e+147
min        0.000000e+00
25%        6.000000e+04
50%        1.100000e+05
75%        2.500000e+05
max       1.000000e+150
Name: CompTotal, dtype: float64
```



Summary

In this lab, you focused on imputing missing values in the dataset.

- Use the `pandas.read_csv()` function to load a dataset from a CSV file into a DataFrame.

- Download the dataset if it's not available online and specify the correct file path.

<!-- ## Change Log |Date (YYYY-MM-DD)|Version|Changed By|Change Description| |-|-|-|-| |2024-11-05|1.3|Madhusudhan Moole|Updated lab| |2024-10-29|1.2|Madhusudhan Moole|Updated lab| |2024-09-27|1.1|Madhusudhan Moole|Updated lab| |2024-09-26|1.0|Raghul Ramesh|Created lab| --!>