Finding Outliers

Estimated time needed: **30** minutes

In this lab, you will work with a cleaned dataset to perform exploratory data analysis or EDA. You will explore the distribution of key variables and focus on identifying outliers in this lab.

Objectives

In this lab, you will perform the following:

- Analyze the distribution of key variables in the dataset.
- Identify and remove outliers using statistical methods.
- Perform relevant statistical and correlation analysis.

Install and import the required libraries

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

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
Requirement already satisfied: pandas in /opt/conda/lib/python3.12/site-packages (2.3.0)
Requirement already satisfied: numpy>=1.26.0 in /opt/conda/lib/python3.12/site-packages (from panda
s) (2.3.0)
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)
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.12/site-packages (from panda
s) (2025.2)
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)
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)
Requirement already satisfied: seaborn in /opt/conda/lib/python3.12/site-packages (0.13.2)
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)
```

Step 1: Load and Explore the Dataset

Load the dataset into a DataFrame and examine the structure of the data.

In [2]:
```python
file_url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/n01PQ9pSmiRX6520fluj

#Create the dataframe
df = pd.read_csv(file_url)

#Display the top 10 records
df.head()
```

Out[2]:

	ResponseId	MainBranch	Age	Employment	RemoteWork	Check	CodingActivities	EdLevel	
0	1	I am a developer by profession	Under 18 years old	Employed, full-time	Remote	Apples	Hobby	Primary/elementary school	E
1	2	I am a developer by profession	35-44 years old	Employed, full-time	Remote	Apples	Hobby;Contribute to open-source projects;Other...	Bachelor's degree (B.A., B.S., B.Eng., etc.)	E medi
2	3	I am a developer by profession	45-54 years old	Employed, full-time	Remote	Apples	Hobby;Contribute to open-source projects;Other...	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	E medi
3	4	I am learning to code	18-24 years old	Student, full-time	NaN	Apples	NaN	Some college/university study without earning ...	vi
4	5	I am a developer by profession	18-24 years old	Student, full-time	NaN	Apples	NaN	Secondary school (e.g. American high school, G...	vi

5 rows × 114 columns

Step 2: Plot the Distribution of Industry

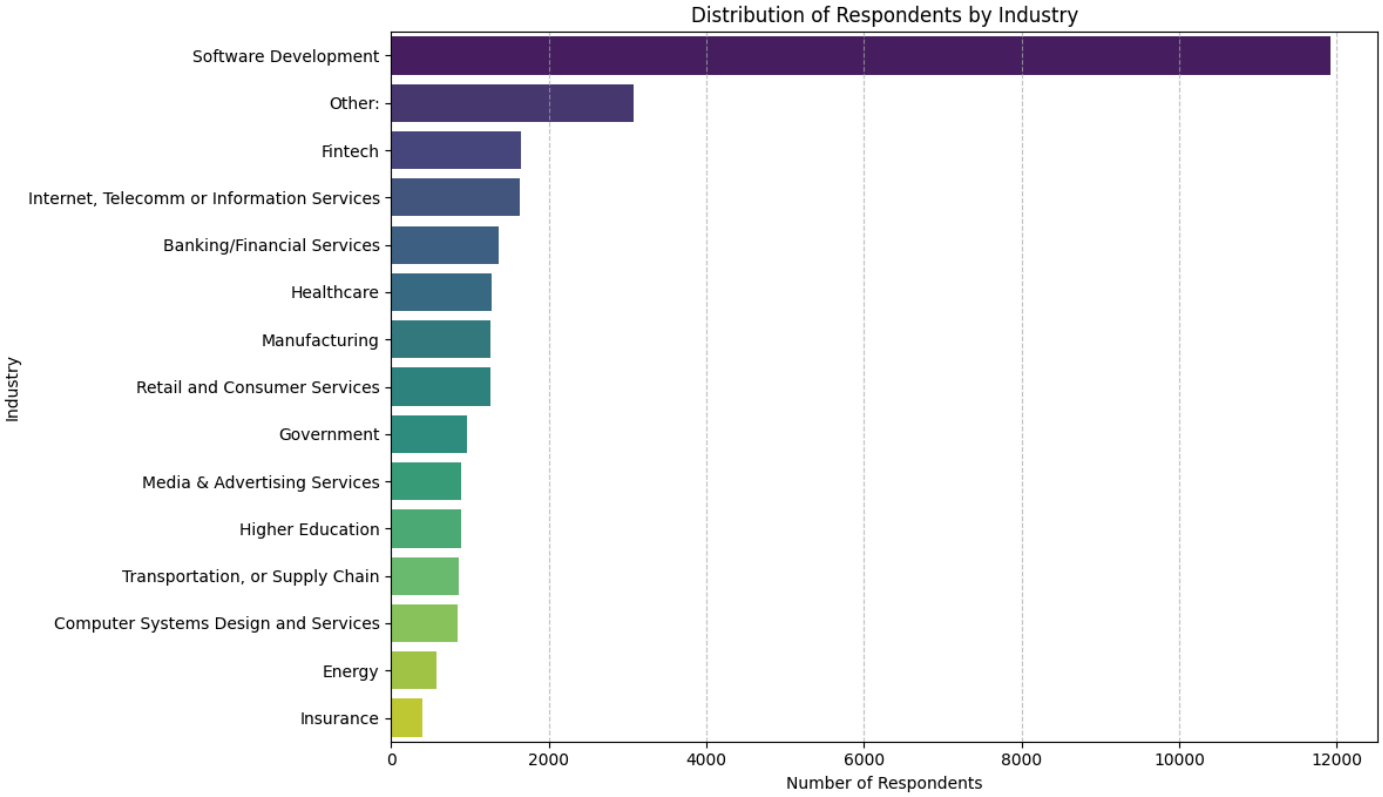
Explore how respondents are distributed across different industries.

- Plot a bar chart to visualize the distribution of respondents by industry.
- Highlight any notable trends.

In [3]:

```
##Write your code here
# --- Step 2: Plot the Distribution of Industry ---
print("\n--- Step 2: Plot the Distribution of Industry ---")
print("Explore how respondents are distributed across different industries.")

if 'Industry' in df.columns:
    plt.figure(figsize=(12, 7))
    # Count plot to visualize the distribution of respondents by industry
    # order by value_counts to show most frequent industries first
    sns.countplot(y='Industry', data=df, order=df['Industry'].value_counts().index, palette='viridi
    plt.title('Distribution of Respondents by Industry')
    plt.xlabel('Number of Respondents')
    plt.ylabel('Industry')
    plt.grid(axis='x', linestyle='--', alpha=0.7)
    plt.tight_layout()
    plt.show()

    print("\nNotable trends in Industry distribution:")
    print(df['Industry'].value_counts())
    # You can add more detailed interpretation based on the actual output after running.
    # For example: "The 'Software' industry has the highest number of respondents, followed by 'Fin
else:
    print("'Industry' column not found. Cannot plot its distribution.")
```

```
--- Step 2: Plot the Distribution of Industry ---
Explore how respondents are distributed across different industries.
/tmp/ipykernel_764/1462635402.py:10: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `
y` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(y='Industry', data=df, order=df['Industry'].value_counts().index, palette='viridis')
```




```
Notable trends in Industry distribution:
Industry
Software Development                          11918
Other:                                         3077
Fintech                                        1641
Internet, Telecomm or Information Services     1629
Banking/Financial Services                     1371
Healthcare                                     1277
Manufacturing                                  1265
Retail and Consumer Services                   1264
Government                                      962
Media & Advertising Services                    894
Higher Education                                890
Transportation, or Supply Chain                 859
Computer Systems Design and Services            844
Energy                                          578
Insurance                                       389
Name: count, dtype: int64
```

Step 3: Identify High Compensation Outliers

Identify respondents with extremely high yearly compensation.

- Calculate basic statistics (mean, median, and standard deviation) for `ConvertedCompYearly`.
- Identify compensation values exceeding a defined threshold (e.g., 3 standard deviations above the mean).

In [4]:

```python
##Write your code here
# --- Step 3: Identify High Compensation Outliers ---
print("\n--- Step 3: Identify High Compensation Outliers ---")
print("Identify respondents with extremely high yearly compensation.")

if 'ConvertedCompYearly' in df.columns and pd.api.types.is_numeric_dtype(df['ConvertedCompYearly']):
    # Calculate basic statistics
    mean_comp = df['ConvertedCompYearly'].mean()
    median_comp = df['ConvertedCompYearly'].median()
    std_comp = df['ConvertedCompYearly'].std()

    print(f"\nBasic Statistics for 'ConvertedCompYearly':")
    print(f"Mean: {mean_comp:.2f}")
    print(f"Median: {median_comp:.2f}")
    print(f"Standard Deviation: {std_comp:.2f}")

    # Identify compensation values exceeding a defined threshold (e.g., 3 standard deviations above
    threshold_3std = mean_comp + (3 * std_comp)
```

```
    high_compensation_outliers = df[df['ConvertedCompYearly'] > threshold_3std]

    print(f"\nThreshold for high compensation (Mean + 3*StdDev): {threshold_3std:.2f}")
    print(f"Number of high compensation outliers: {len(high_compensation_outliers)}")

    if not high_compensation_outliers.empty:
        print("\nHigh compensation outliers (first 5 records):")
        print(high_compensation_outliers.head())
    else:
        print("No compensation values found exceeding 3 standard deviations above the mean.")
else:
    print("'ConvertedCompYearly' column not found or is not numeric. Cannot identify high compensat
```

```
--- Step 3: Identify High Compensation Outliers ---
Identify respondents with extremely high yearly compensation.

Basic Statistics for 'ConvertedCompYearly':
Mean: 86155.29
Median: 65000.00
Standard Deviation: 186756.97

Threshold for high compensation (Mean + 3*StdDev): 646426.21
Number of high compensation outliers: 89

High compensation outliers (first 5 records):
      ResponseId                      MainBranch              Age  \
529          530  I am a developer by profession  25-34 years old
828          829  I am a developer by profession  35-44 years old
1932        1933  I am a developer by profession  25-34 years old
2171        2172  I am a developer by profession  35-44 years old
2187        2188  I am a developer by profession  35-44 years old

               Employment                              RemoteWork   Check  \
529   Employed, full-time                               In-person  Apples
828   Employed, full-time  Hybrid (some remote, some in-person)  Apples
1932  Employed, full-time                                  Remote  Apples
2171  Employed, full-time  Hybrid (some remote, some in-person)  Apples
2187  Employed, full-time                                  Remote  Apples

                                     CodingActivities  \
529                                             Hobby
828   Hobby;Bootstrapping a business;Professional de...
1932  Hobby;Professional development or self-paced l...
2171                                            Hobby
2187          Hobby;Contribute to open-source projects

                                          EdLevel  \
529      Bachelor’s degree (B.A., B.S., B.Eng., etc.)
828   Master’s degree (M.A., M.S., M.Eng., MBA, etc.)
1932     Bachelor’s degree (B.A., B.S., B.Eng., etc.)
2171     Bachelor’s degree (B.A., B.S., B.Eng., etc.)
2187     Bachelor’s degree (B.A., B.S., B.Eng., etc.)

                                          LearnCode  \
529   Books / Physical media;School (i.e., Universit...
828   Books / Physical media;Colleague;On the job tr...
1932  Books / Physical media;Colleague;Other online ...
2171  On the job training;Other online resources (e....
2187  On the job training;Other online resources (e....

                                    LearnCodeOnline  ... JobSatPoints_6  \
529                                             NaN  ...           40.0
828   Technical documentation;Blogs;Books;Written Tu...  ...           30.0
1932  Technical documentation;Blogs;Books;Written Tu...  ...           15.0
2171  Technical documentation;Blogs;Written Tutorial...  ...           20.0
2187  Technical documentation;Written Tutorials;Stac...  ...            NaN

     JobSatPoints_7 JobSatPoints_8 JobSatPoints_9 JobSatPoints_10  \
529            20.0            0.0           30.0            10.0
828            10.0            0.0            5.0             0.0
1932           10.0           15.0           15.0             0.0
2171           10.0           20.0           25.0             0.0
2187            NaN            NaN            NaN             NaN

     JobSatPoints_11           SurveyLength                  SurveyEase  \
529              0.0  Appropriate in length                        Easy
828              0.0  Appropriate in length  Neither easy nor difficult
1932            15.0  Appropriate in length                        Easy
2171             0.0  Appropriate in length                        Easy
2187             NaN  Appropriate in length                        Easy

     ConvertedCompYearly JobSat
529             650000.0    6.0
828            1000000.0    8.0
1932            945000.0    2.0
2171            750000.0    8.0
2187           2000000.0    NaN
```

```
[5 rows x 114 columns]
```

Step 4: Detect Outliers in Compensation

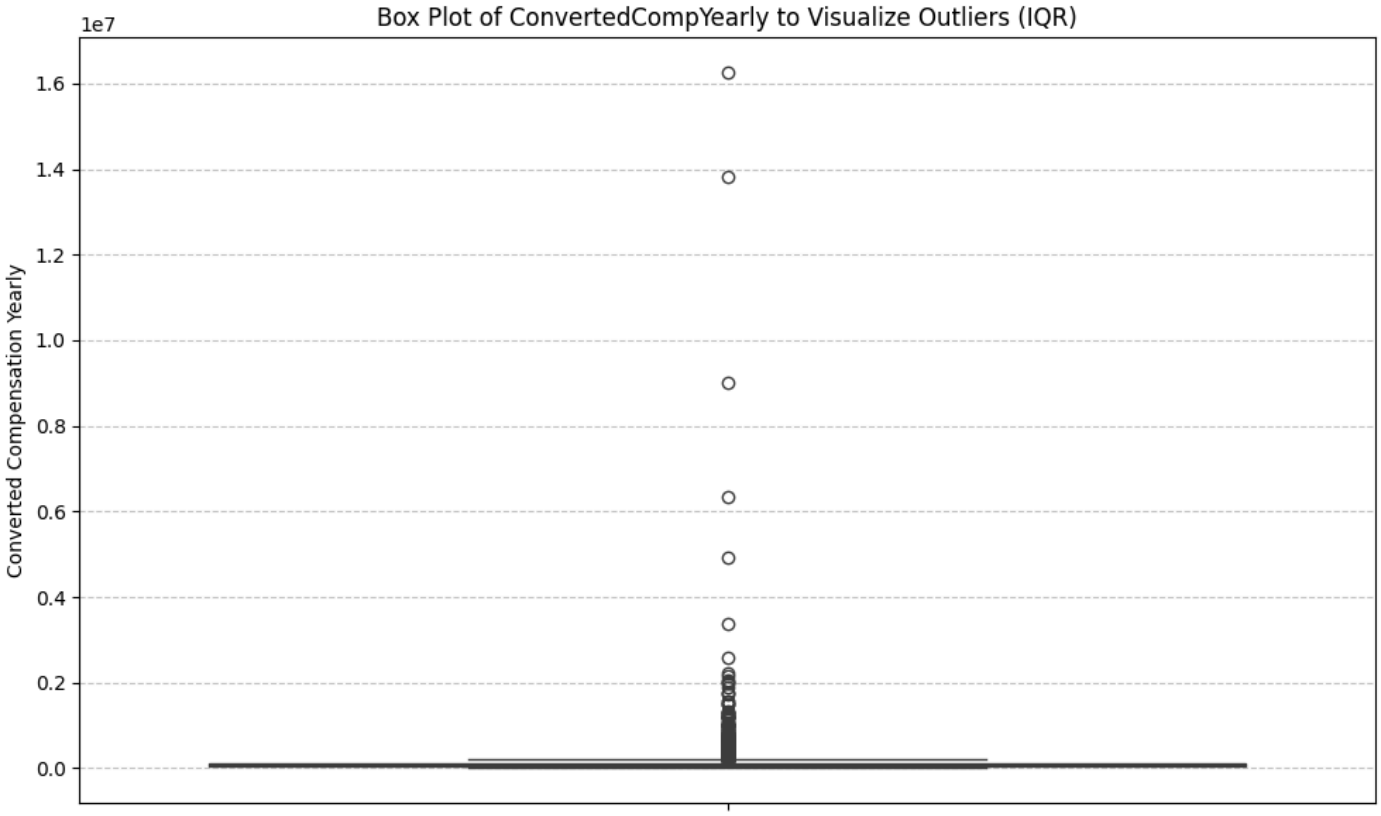
Identify outliers in the `ConvertedCompYearly` column using the IQR method.

- Calculate the Interquartile Range (IQR).
- Determine the upper and lower bounds for outliers.
- Count and visualize outliers using a box plot.

In [5]:
```python
##Write your code here
# --- Step 4: Detect Outliers in Compensation (IQR Method) ---
print("\n--- Step 4: Detect Outliers in Compensation (IQR Method) ---")
print("Identify outliers in the ConvertedCompYearly column using the IQR method.")

if 'ConvertedCompYearly' in df.columns and pd.api.types.is_numeric_dtype(df['ConvertedCompYearly']):
    Q1 = df['ConvertedCompYearly'].quantile(0.25)
    Q3 = df['ConvertedCompYearly'].quantile(0.75)
    IQR = Q3 - Q1

    upper_bound = Q3 + (1.5 * IQR)
    lower_bound = Q1 - (1.5 * IQR)

    print(f"\nQuartiles and IQR for 'ConvertedCompYearly':")
    print(f"Q1 (25th percentile): {Q1:.2f}")
    print(f"Q3 (75th percentile): {Q3:.2f}")
    print(f"IQR (Q3 - Q1): {IQR:.2f}")
    print(f"Upper Bound (Q3 + 1.5 * IQR): {upper_bound:.2f}")
    print(f"Lower Bound (Q1 - 1.5 * IQR): {lower_bound:.2f}")

    # Count outliers
    outliers_iqr = df[(df['ConvertedCompYearly'] < lower_bound) | (df['ConvertedCompYearly'] > uppe
    print(f"\nNumber of outliers detected by IQR method: {len(outliers_iqr)}")

    if not outliers_iqr.empty:
        print("\nOutliers detected by IQR method (first 5 records):")
        print(outliers_iqr.head())
    else:
        print("No outliers detected by IQR method.")

    # Visualize outliers using a box plot
    plt.figure(figsize=(10, 6))
    sns.boxplot(y=df['ConvertedCompYearly'])
    plt.title('Box Plot of ConvertedCompYearly to Visualize Outliers (IQR)')
    plt.ylabel('Converted Compensation Yearly')
    plt.grid(axis='y', linestyle='--', alpha=0.7)
    plt.tight_layout()
    plt.show()

else:
    print("'ConvertedCompYearly' column not found or is not numeric. Cannot detect outliers using I
```

```
--- Step 4: Detect Outliers in Compensation (IQR Method) ---
Identify outliers in the ConvertedCompYearly column using the IQR method.

Quartiles and IQR for 'ConvertedCompYearly':
Q1 (25th percentile): 32712.00
Q3 (75th percentile): 107971.50
IQR (Q3 - Q1): 75259.50
Upper Bound (Q3 + 1.5 * IQR): 220860.75
Lower Bound (Q1 - 1.5 * IQR): -80177.25

Number of outliers detected by IQR method: 978

Outliers detected by IQR method (first 5 records):
     ResponseId                      MainBranch              Age  \
428         429  I am a developer by profession  25-34 years old
456         457  I am a developer by profession  45-54 years old
461         462  I am a developer by profession  45-54 years old
529         530  I am a developer by profession  25-34 years old
545         546  I am a developer by profession  35-44 years old

              Employment                                RemoteWork   Check  \
428  Employed, full-time                                    Remote  Apples
456  Employed, full-time  Hybrid (some remote, some in-person)  Apples
461  Employed, full-time  Hybrid (some remote, some in-person)  Apples
529  Employed, full-time                             In-person  Apples
545  Employed, full-time                                    Remote  Apples

                                    CodingActivities  \
428  Hobby;Professional development or self-paced l...
456                      I don't code outside of work
461  Hobby;Professional development or self-paced l...
529                                             Hobby
545  Hobby;Contribute to open-source projects;Profe...

                                          EdLevel  \
428      Bachelor's degree (B.A., B.S., B.Eng., etc.)
456    Professional degree (JD, MD, Ph.D, Ed.D, etc.)
461   Master's degree (M.A., M.S., M.Eng., MBA, etc.)
529      Bachelor's degree (B.A., B.S., B.Eng., etc.)
545  Secondary school (e.g. American high school, G...

                                         LearnCode  \
428  Books / Physical media;On the job training;Oth...
456            School (i.e., University, College, etc)
461  Books / Physical media;Colleague;On the job tr...
529  Books / Physical media;School (i.e., Universit...
545  Books / Physical media;Colleague;On the job tr...

                                    LearnCodeOnline  ... JobSatPoints_6  \
428  Technical documentation;Blogs;Books;Written Tu...  ...            0.0
456                                                NaN  ...           10.0
461  Technical documentation;Blogs;Books;Written Tu...  ...           20.0
529                                                NaN  ...           40.0
545  Technical documentation;Blogs;Books;Written Tu...  ...           30.0

    JobSatPoints_7 JobSatPoints_8 JobSatPoints_9 JobSatPoints_10  \
428            0.0            0.0           25.0            25.0
456           10.0            0.0            0.0             0.0
461           50.0            0.0           10.0            10.0
529           20.0            0.0           30.0            10.0
545            5.0            5.0           10.0             0.0

    JobSatPoints_11          SurveyLength                SurveyEase  \
428            25.0  Appropriate in length  Neither easy nor difficult
456             0.0  Appropriate in length  Neither easy nor difficult
461             0.0  Appropriate in length  Neither easy nor difficult
529             0.0  Appropriate in length                      Easy
545             0.0  Appropriate in length                      Easy

    ConvertedCompYearly JobSat
428            230000.0    8.0
456            300000.0    1.0
461            254425.0    7.0
529            650000.0    6.0
545            400000.0    8.0
```

```
[5 rows x 114 columns]
```



Step 5: Remove Outliers and Create a New DataFrame

Remove outliers from the dataset.

- Create a new DataFrame excluding rows with outliers in `ConvertedCompYearly`.
- Validate the size of the new DataFrame.

In [6]:
```
##Write your code here
# --- Step 5: Remove Outliers and Create a New DataFrame ---
print("\n--- Step 5: Remove Outliers and Create a New DataFrame ---")
print("Remove outliers from the dataset. Create a new DataFrame excluding rows with outliers in Con

if 'ConvertedCompYearly' in df.columns and pd.api.types.is_numeric_dtype(df['ConvertedCompYearly']):
    Q1 = df['ConvertedCompYearly'].quantile(0.25)
    Q3 = df['ConvertedCompYearly'].quantile(0.75)
    IQR = Q3 - Q1
    upper_bound = Q3 + (1.5 * IQR)
    lower_bound = Q1 - (1.5 * IQR)

    # Create a new DataFrame excluding rows with outliers
    df_cleaned = df[(df['ConvertedCompYearly'] >= lower_bound) & (df['ConvertedCompYearly'] <= uppe

    print(f"\nOriginal DataFrame size: {len(df)} rows")
    print(f"Cleaned DataFrame size (after outlier removal): {len(df_cleaned)} rows")
    print(f"Number of rows removed (outliers): {len(df) - len(df_cleaned)}")

    print("\nFirst 5 rows of the new cleaned DataFrame:")
    print(df_cleaned.head())
else:
    print("'ConvertedCompYearly' column not found or is not numeric. Cannot remove outliers.")
    df_cleaned = df.copy() # Ensure df_cleaned exists for subsequent steps
```

```
--- Step 5: Remove Outliers and Create a New DataFrame ---
Remove outliers from the dataset. Create a new DataFrame excluding rows with outliers in ConvertedCo
mpYearly.

Original DataFrame size: 65437 rows
Cleaned DataFrame size (after outlier removal): 22457 rows
Number of rows removed (outliers): 42980

First 5 rows of the new cleaned DataFrame:
     ResponseId                                         MainBranch  \
72           73                     I am a developer by profession
374         375  I am not primarily a developer, but I write co...
379         380                     I am a developer by profession
385         386                     I am a developer by profession
389         390                     I am a developer by profession

                 Age                                         Employment  \
72   18-24 years old  Employed, full-time;Student, full-time;Indepen...
374  25-34 years old                                Employed, full-time
379  35-44 years old                                Employed, full-time
385  35-44 years old  Independent contractor, freelancer, or self-em...
389  25-34 years old            Employed, full-time;Student, part-time

                              RemoteWork   Check  \
72   Hybrid (some remote, some in-person)  Apples
374  Hybrid (some remote, some in-person)  Apples
379                              Remote  Apples
385                              Remote  Apples
389                              Remote  Apples

                                  CodingActivities  \
72   Hobby;School or academic work;Professional dev...
374  Hobby;School or academic work;Professional dev...
379                    Hobby;Bootstrapping a business
385                                             Hobby
389                     Hobby;School or academic work

                                           EdLevel  \
72   Secondary school (e.g. American high school, G...
374     Professional degree (JD, MD, Ph.D, Ed.D, etc.)
379    Master’s degree (M.A., M.S., M.Eng., MBA, etc.)
385    Master’s degree (M.A., M.S., M.Eng., MBA, etc.)
389  Some college/university study without earning ...

                                         LearnCode  \
72   On the job training;Other online resources (e....
374  Books / Physical media;Colleague;On the job tr...
379  Books / Physical media;Other online resources ...
385  Books / Physical media;On the job training;Oth...
389  Books / Physical media;Colleague;On the job tr...

                                    LearnCodeOnline  ... JobSatPoints_6  \
72   Technical documentation;Blogs;Written Tutorial...  ...          65.0
374  Written Tutorials;Stack Overflow;Written-based...  ...           NaN
379  Technical documentation;Books;Social Media;Wri...  ...           0.0
385  Technical documentation;Blogs;Written Tutorial...  ...           NaN
389  Written Tutorials;Stack Overflow;Coding sessio...  ...          20.0

     JobSatPoints_7 JobSatPoints_8 JobSatPoints_9 JobSatPoints_10  \
72            100.0          100.0          100.0            50.0
374             NaN            NaN            NaN             NaN
379             0.0            0.0            0.0             0.0
385             NaN            NaN            NaN             NaN
389            30.0            5.0           20.0            10.0

     JobSatPoints_11          SurveyLength                SurveyEase  \
72              90.0              Too long                      Easy
374              NaN  Appropriate in length  Neither easy nor difficult
379              0.0              Too long                 Difficult
385              NaN             Too short                      Easy
389              5.0              Too long                      Easy

     ConvertedCompYearly JobSat
72                7322.0   10.0
374              30074.0    NaN
```

```
379             91295.0   10.0
385             53703.0    NaN
389            110000.0   10.0

[5 rows x 114 columns]
```

Step 6: Correlation Analysis

Analyze the correlation between `Age` (transformed) and other numerical columns.

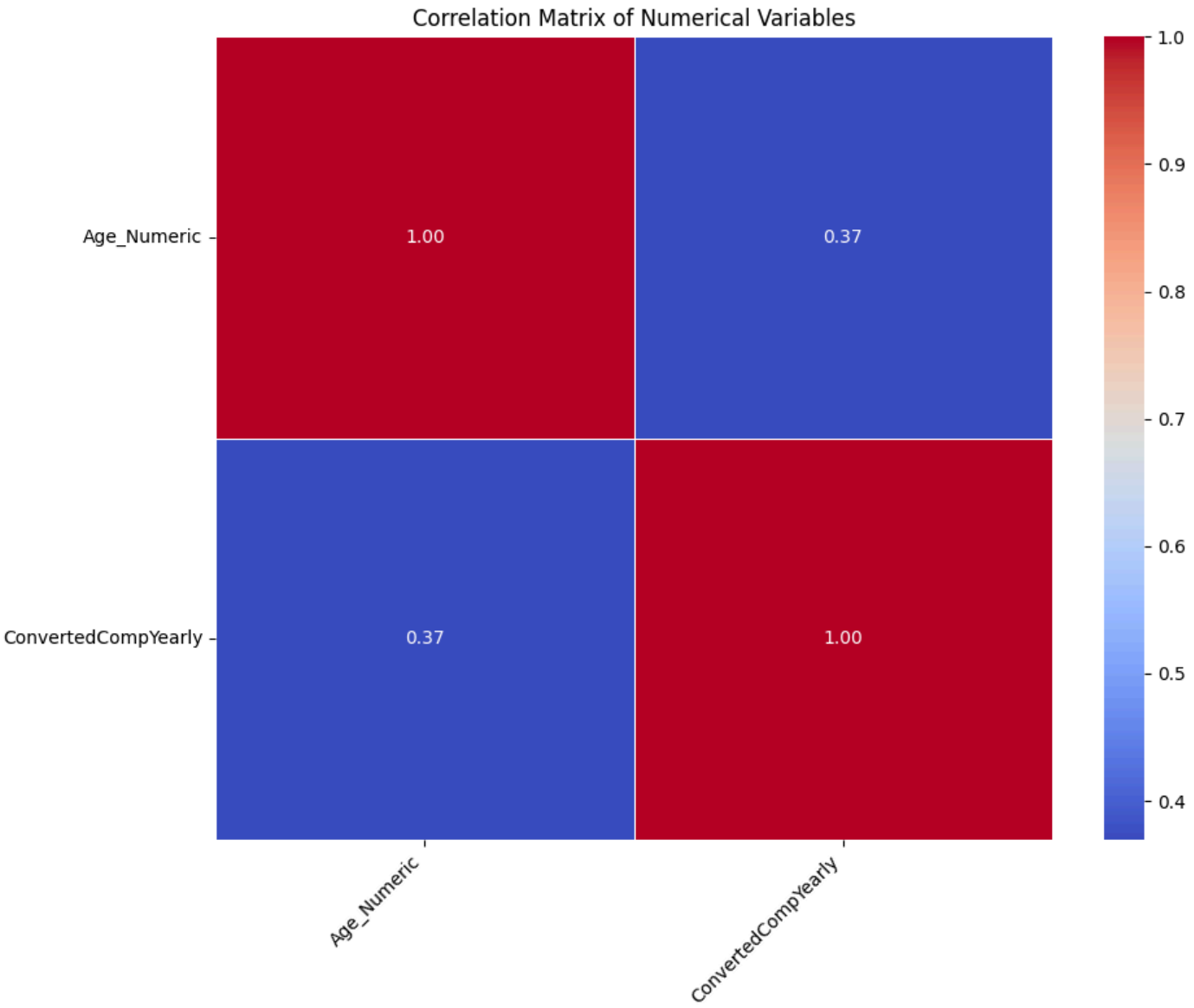
- Map the `Age` column to approximate numeric values.
- Compute correlations between `Age` and other numeric variables.
- Visualize the correlation matrix.

In [10]:

```
##Write your code here

!pip install numpy
import numpy as np

# --- Step 6: Correlation Analysis ---
print("\n--- Step 6: Correlation Analysis ---")
print("Analyze the correlation between Age (transformed) and other numerical columns.")

if 'Age' in df_cleaned.columns:
    # 1. Map the Age column to approximate numeric values.
    # Define a mapping for Age ranges to their approximate median values
    age_mapping = {
        'Under 18 years old': 17,
        '18-24 years old': 21,
        '25-34 years old': 29,
        '35-44 years old': 39,
        '45-54 years old': 49,
        '55-64 years old': 59,
        '65 years or older': 65
    }

    # Apply the mapping to create a new numeric 'Age_Numeric' column
    # Use .map() and fill any NaNs if they exist in the original 'Age'
    df_cleaned['Age_Numeric'] = df_cleaned['Age'].map(age_mapping)

    # If any Age values didn't map (e.g., if there were NaNs or unexpected strings),
    # they will become NaN after mapping. Impute these if necessary.
    if df_cleaned['Age_Numeric'].isnull().any():
        median_age_numeric = df_cleaned['Age_Numeric'].median()
        df_cleaned['Age_Numeric'].fillna(median_age_numeric, inplace=True)
        print(f"Missing or unmapped 'Age' values imputed with median numeric age: {median_age_numer

    print("\n'Age' column successfully transformed to 'Age_Numeric'.")
    print("Sample of 'Age' and 'Age_Numeric':")
    print(df_cleaned[['Age', 'Age_Numeric']].head())

    # 2. Compute correlations between Age_Numeric and other numeric variables.
    # Explicitly define expected numeric columns for correlation
    expected_numeric_for_corr = ['Age_Numeric', 'ConvertedCompYearly', 'YearsCodePro']

    # Filter df_cleaned to include only existing, numeric columns from the expected list
    actual_numeric_cols_for_corr = [
        col for col in expected_numeric_for_corr
        if col in df_cleaned.columns and pd.api.types.is_numeric_dtype(df_cleaned[col])
    ]
    numeric_cols = df_cleaned[actual_numeric_cols_for_corr]

    print(f"\nDiagnostic: Numeric columns selected for correlation: {numeric_cols.columns.tolist()}
    print(f"Diagnostic: Shape of numeric_cols: {numeric_cols.shape}")
    print(f"Diagnostic: Head of numeric_cols:\n{numeric_cols.head()}")
    print(f"Diagnostic: dtypes of numeric_cols:\n{numeric_cols.dtypes}")


    # Ensure there are at least two numeric columns for a meaningful correlation matrix
    # and that the DataFrame is not empty.
    if not numeric_cols.empty and numeric_cols.shape[1] >= 2:
```

```
        correlation_matrix = numeric_cols.corr() # .corr() with no args calculates correlation betw

        # Extract correlations with 'Age_Numeric'
        if 'Age_Numeric' in correlation_matrix.columns:
            age_correlations = correlation_matrix['Age_Numeric'].sort_values(ascending=False)
            print("\nCorrelation of 'Age_Numeric' with other numerical columns:")
            print(age_correlations)
        else:
            print("Warning: 'Age_Numeric' not found in the correlation matrix after calculation. Th
            print("Correlation matrix:\n", correlation_matrix) # Print full matrix for debugging

        # 3. Visualize the correlation matrix.
        if not correlation_matrix.empty: # Ensure the matrix is not empty before plotting
            plt.figure(figsize=(10, 8))
            sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
            plt.title('Correlation Matrix of Numerical Variables')
            plt.xticks(rotation=45, ha='right')
            plt.yticks(rotation=0)
            plt.tight_layout()
            plt.show()
        else:
            print("Correlation matrix is empty. Cannot visualize.")
    else:
        print("Warning: Insufficient numeric columns or empty DataFrame for meaningful correlation
        print(f"Number of numeric columns: {numeric_cols.shape[1]}. DataFrame empty: {numeric_cols.
        print("Consider checking your data loading and previous cleaning steps if this message appe

else:
    print("'Age' column not found in the DataFrame. Cannot perform Step 6 correlation analysis.")
```

```
Requirement already satisfied: numpy in /opt/conda/lib/python3.12/site-packages (2.3.0)

--- Step 6: Correlation Analysis ---
Analyze the correlation between Age (transformed) and other numerical columns.
Missing or unmapped 'Age' values imputed with median numeric age: 29

'Age' column successfully transformed to 'Age_Numeric'.
Sample of 'Age' and 'Age_Numeric':
                 Age  Age_Numeric
72   18-24 years old         21.0
374  25-34 years old         29.0
379  35-44 years old         39.0
385  35-44 years old         39.0
389  25-34 years old         29.0

Diagnostic: Numeric columns selected for correlation: ['Age_Numeric', 'ConvertedCompYearly']
Diagnostic: Shape of numeric_cols: (22457, 2)
Diagnostic: Head of numeric_cols:
     Age_Numeric  ConvertedCompYearly
72          21.0               7322.0
374         29.0              30074.0
379         39.0              91295.0
385         39.0              53703.0
389         29.0             110000.0
Diagnostic: dtypes of numeric_cols:
Age_Numeric            float64
ConvertedCompYearly    float64
dtype: object

Correlation of 'Age_Numeric' with other numerical columns:
Age_Numeric            1.0000
ConvertedCompYearly    0.3693
Name: Age_Numeric, dtype: float64
/tmp/ipykernel_764/1993684001.py:31: FutureWarning: A value is trying to be set on a copy of a DataF
rame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate
object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, in
place=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the ori
ginal object.


  df_cleaned['Age_Numeric'].fillna(median_age_numeric, inplace=True)
```



Summary

In this lab, you developed essential skills in **Exploratory Data Analysis (EDA)** with a focus on outlier detection and removal. Specifically, you:

- Loaded and explored the dataset to understand its structure.
- Analyzed the distribution of respondents across industries.
- Identified and removed high compensation outliers using statistical thresholds and the Interquartile Range (IQR) method.
- Performed correlation analysis, including transforming the `Age` column into numeric values for better analysis.

<!-- ## Change Log |Date (YYYY-MM-DD)|Version|Changed By|Change Description| |-|-|-|-| |2024-10-1|1.1|Madhusudan Moole|Reviewed and updated lab| |2024-09-29|1.0|Raghul Ramesh|Created lab| --!>