Skills Network

Data Wrangling Lab

Estimated time needed: **45** minutes

In this lab, you will perform data wrangling tasks to prepare raw data for analysis. Data wrangling involves cleaning, transforming, and organizing data into a structured format suitable for analysis. This lab focuses on tasks like identifying inconsistencies, encoding categorical variables, and feature transformation.

Objectives

After completing this lab, you will be able to:

- Identify and remove inconsistent data entries.
- Encode categorical variables for analysis.
- Handle missing values using multiple imputation strategies.
- Apply feature scaling and transformation techniques.

Intsall the required libraries

In [1]:

```
!pip install pandas
!pip install matplotlib
```

```
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)
```

Tasks

Step 1: Import the necessary module.

1. Load the Dataset

1.1 Import necessary libraries and load the dataset.

Ensure the dataset is loaded correctly by displaying the first few rows.

In [2]:

```
# Import necessary libraries
import pandas as pd

# Load the Stack Overflow survey data
dataset_url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/n01PQ9pSmiRX6520f
df = pd.read_csv(dataset_url)

# Display the first few rows
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]
```

2. Explore the Dataset

2.1 Summarize the dataset by displaying the column data types, counts, and missing values.

In [10]:

```python
# Write your code here

print("--- Dataset Summary (df.dtypes) ---")
print(df.dtypes)

print("--- Dataset Summary (df.info()) ---")
df.info()

print("\n--- Missing Values Count (df.isnull().sum()) ---")
print(df.isnull().sum())
```

```
--- Dataset Summary (df.dtypes) ---
ResponseId               int64
MainBranch              object
Age                     object
Employment              object
RemoteWork              object
                        ...   
JobSatPoints_11        float64
SurveyLength            object
SurveyEase              object
ConvertedCompYearly    float64
JobSat                 float64
Length: 114, dtype: object
--- Dataset Summary (df.info()) ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 65437 entries, 0 to 65436
Columns: 114 entries, ResponseId to JobSat
dtypes: float64(13), int64(1), object(100)
memory usage: 56.9+ MB

--- Missing Values Count (df.isnull().sum()) ---
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
```

2.2 Generate basic statistics for numerical columns.

In [9]:

```
# Write your code here

print("\n--- Summary Statistics (df.describe()) ---")
print(df.describe())
```

```
--- Summary Statistics (df.describe()) ---
         ResponseId      CompTotal       WorkExp  JobSatPoints_1  \
count  65437.000000   3.374000e+04  29658.000000    29324.000000   
mean   32719.000000  2.963841e+145     11.466957       18.581094   
std    18890.179119  5.444117e+147      9.168709       25.966221   
min        1.000000   0.000000e+00      0.000000        0.000000   
25%    16360.000000   6.000000e+04      4.000000        0.000000   
50%    32719.000000   1.100000e+05      9.000000       10.000000   
75%    49078.000000   2.500000e+05     16.000000       22.000000   
max    65437.000000  1.000000e+150     50.000000      100.000000   

       JobSatPoints_4  JobSatPoints_5  JobSatPoints_6  JobSatPoints_7  \
count    29393.000000    29411.000000    29450.000000     29448.00000   
mean         7.522140       10.060857       24.343232        22.96522   
std         18.422661       21.833836       27.089360        27.01774   
min          0.000000        0.000000        0.000000         0.00000   
25%          0.000000        0.000000        0.000000         0.00000   
50%          0.000000        0.000000       20.000000        15.00000   
75%          5.000000       10.000000       30.000000        30.00000   
max        100.000000      100.000000      100.000000       100.00000   

       JobSatPoints_8  JobSatPoints_9  JobSatPoints_10  JobSatPoints_11  \
count    29456.000000    29456.000000     29450.000000     29445.000000   
mean        20.278165       16.169432        10.955713         9.953948   
std         26.108110       24.845032        22.906263        21.775652   
min          0.000000        0.000000         0.000000         0.000000   
25%          0.000000        0.000000         0.000000         0.000000   
50%         10.000000        5.000000         0.000000         0.000000   
75%         25.000000       20.000000        10.000000        10.000000   
max        100.000000      100.000000       100.000000       100.000000   

       ConvertedCompYearly        JobSat  
count         2.343500e+04  29126.000000  
mean          8.615529e+04      6.935041  
std           1.867570e+05      2.088259  
min           1.000000e+00      0.000000  
25%           3.271200e+04      6.000000  
50%           6.500000e+04      7.000000  
75%           1.079715e+05      8.000000  
max           1.625660e+07     10.000000  
```

3. Identifying and Removing Inconsistencies

3.1 Identify inconsistent or irrelevant entries in specific columns (e.g., Country).

In [11]:

```python
# Write your code here
import pandas as pd
import numpy as np

print("--- Identifying Inconsistent or Irrelevant Entries in 'Country' Column ---")

# 1. Display all unique values in the 'Country' column
print("\nUnique values in 'Country' column (before cleaning):")
print(df['Country'].unique())

# 2. Display value counts to see frequencies and spot inconsistencies
print("\nValue counts for 'Country' column (before cleaning):")
print(df['Country'].value_counts())

# 3. (Optional but recommended) Standardize text data for better identification
# Convert to lowercase and strip whitespace to catch common inconsistencies
df['Country_Cleaned'] = df['Country'].astype(str).str.lower().str.strip()

print("\nUnique values in 'Country' column (after basic cleaning - lowercase, strip whitespace):")
print(df['Country_Cleaned'].unique())
print("\nValue counts for 'Country_Cleaned' column (after basic cleaning):")
print(df['Country_Cleaned'].value_counts())

# 4. Identify irrelevant entries (e.g., 'N/A', placeholder strings, or clearly wrong entries)
# This step is highly dependent on your knowledge of the expected data.
irrelevant_entries = ['n/a', 'unknown', 'not applicable'] # Define what you consider irrelevant

print("\nIdentifying irrelevant entries (e.g., 'n/a', 'unknown'):")
```

```
for entry in irrelevant_entries:
    if entry in df['Country_Cleaned'].unique():
        print(f"Found irrelevant entry: '{entry}'")

# Example of how to address them (e.g., replace with NaN or remove)
# df['Country_Cleaned'] = df['Country_Cleaned'].replace(irrelevant_entries, np.nan)
# print("\n'Country_Cleaned' after replacing irrelevant entries with NaN:")
# print(df['Country_Cleaned'].unique())

# You would then decide how to handle these inconsistencies (e.g., correct typos,
# map similar values to a single standard value, or remove irrelevant entries).
# Example mapping:
# country_mapping = {
#     'usa': 'United States',
#     'u.s.a.': 'United States',
#     'uk': 'United Kingdom',
#     'canada ': 'Canada' # Already handled by strip(), but good for explicit mapping
# }
# df['Country_Cleaned'] = df['Country_Cleaned'].replace(country_mapping)
# print("\nUnique values after mapping common inconsistencies:")
# print(df['Country_Cleaned'].unique())
```

```
--- Identifying Inconsistent or Irrelevant Entries in 'Country' Column ---

Unique values in 'Country' column (before cleaning):
['United States of America'
 'United Kingdom of Great Britain and Northern Ireland' 'Canada' 'Norway'
 'Uzbekistan' 'Serbia' 'Poland' 'Philippines' 'Bulgaria' 'Switzerland'
 'India' 'Germany' 'Ireland' 'Italy' 'Ukraine' 'Australia' 'Brazil'
 'Japan' 'Austria' 'Iran, Islamic Republic of...' 'France' 'Saudi Arabia'
 'Romania' 'Turkey' 'Nepal' 'Algeria' 'Sweden' 'Netherlands' 'Croatia'
 'Pakistan' 'Czech Republic' 'Republic of North Macedonia' 'Finland'
 'Slovakia' 'Russian Federation' 'Greece' 'Israel' 'Belgium' 'Mexico'
 'United Republic of Tanzania' 'Hungary' 'Argentina' 'Portugal'
 'Sri Lanka' 'Latvia' 'China' 'Singapore' 'Lebanon' 'Spain' 'South Africa'
 'Lithuania' 'Viet Nam' 'Dominican Republic' 'Indonesia' 'Kosovo'
 'Morocco' 'Taiwan' 'Georgia' 'San Marino' 'Tunisia' 'Bangladesh'
 'Nigeria' 'Liechtenstein' 'Denmark' 'Ecuador' 'Malaysia' 'Albania'
 'Azerbaijan' 'Chile' 'Ghana' 'Peru' 'Bolivia' 'Egypt' 'Luxembourg'
 'Montenegro' 'Cyprus' 'Paraguay' 'Kazakhstan' 'Slovenia' 'Jordan'
 'Venezuela, Bolivarian Republic of...' 'Costa Rica' 'Jamaica' 'Thailand'
 'Nicaragua' 'Myanmar' 'Republic of Korea' 'Rwanda'
 'Bosnia and Herzegovina' 'Benin' 'El Salvador' 'Zimbabwe' 'Afghanistan'
 'Estonia' 'Malta' 'Uruguay' 'Belarus' 'Colombia' 'Republic of Moldova'
 'Isle of Man' 'Nomadic' 'New Zealand' 'Palestine' 'Armenia'
 'United Arab Emirates' 'Maldives' 'Ethiopia' 'Fiji' 'Guatemala' 'Uganda'
 'Turkmenistan' 'Mauritius' 'Kenya' 'Cuba' 'Gabon' 'Bahamas' 'South Korea'
 'Iceland' 'Honduras' 'Hong Kong (S.A.R.)'
 "Lao People's Democratic Republic" 'Mongolia' 'Cambodia' 'Madagascar'
 'Angola' 'Democratic Republic of the Congo' 'Syrian Arab Republic' 'Iraq'
 'Namibia' 'Senegal' 'Kyrgyzstan' 'Zambia' 'Swaziland' "Côte d'Ivoire"
 'Kuwait' 'Tajikistan' 'Burundi' 'Trinidad and Tobago' 'Mauritania'
 'Sierra Leone' 'Panama' 'Somalia' 'North Korea' 'Dominica' 'Guyana'
 'Togo' 'Oman' 'Barbados' 'Andorra'
 "Democratic People's Republic of Korea" 'Qatar' 'Sudan' 'Cameroon'
 'Papua New Guinea' 'Bahrain' 'Yemen' 'Malawi' 'Burkina Faso'
 'Congo, Republic of the...' 'Botswana' 'Guinea-Bissau' 'Mozambique'
 'Central African Republic' 'Equatorial Guinea' 'Suriname' 'Belize'
 'Libyan Arab Jamahiriya' 'Cape Verde' 'Brunei Darussalam' 'Bhutan'
 'Guinea' 'Niger' 'Antigua and Barbuda' 'Mali' 'Samoa' 'Lesotho'
 'Saint Kitts and Nevis' 'Monaco' 'Micronesia, Federated States of...'
 'Haiti' nan 'Nauru' 'Liberia' 'Chad' 'Djibouti' 'Solomon Islands']

Value counts for 'Country' column (before cleaning):
Country
United States of America                                11095
Germany                                                  4947
India                                                    4231
United Kingdom of Great Britain and Northern Ireland     3224
Ukraine                                                  2672
                                                        ...  
Micronesia, Federated States of...                          1
Nauru                                                       1
Chad                                                        1
Djibouti                                                    1
Solomon Islands                                             1
Name: count, Length: 185, dtype: int64

Unique values in 'Country' column (after basic cleaning - lowercase, strip whitespace):
['united states of america'
 'united kingdom of great britain and northern ireland' 'canada' 'norway'
 'uzbekistan' 'serbia' 'poland' 'philippines' 'bulgaria' 'switzerland'
 'india' 'germany' 'ireland' 'italy' 'ukraine' 'australia' 'brazil'
 'japan' 'austria' 'iran, islamic republic of...' 'france' 'saudi arabia'
 'romania' 'turkey' 'nepal' 'algeria' 'sweden' 'netherlands' 'croatia'
 'pakistan' 'czech republic' 'republic of north macedonia' 'finland'
 'slovakia' 'russian federation' 'greece' 'israel' 'belgium' 'mexico'
 'united republic of tanzania' 'hungary' 'argentina' 'portugal'
 'sri lanka' 'latvia' 'china' 'singapore' 'lebanon' 'spain' 'south africa'
 'lithuania' 'viet nam' 'dominican republic' 'indonesia' 'kosovo'
 'morocco' 'taiwan' 'georgia' 'san marino' 'tunisia' 'bangladesh'
 'nigeria' 'liechtenstein' 'denmark' 'ecuador' 'malaysia' 'albania'
 'azerbaijan' 'chile' 'ghana' 'peru' 'bolivia' 'egypt' 'luxembourg'
 'montenegro' 'cyprus' 'paraguay' 'kazakhstan' 'slovenia' 'jordan'
 'venezuela, bolivarian republic of...' 'costa rica' 'jamaica' 'thailand'
 'nicaragua' 'myanmar' 'republic of korea' 'rwanda'
 'bosnia and herzegovina' 'benin' 'el salvador' 'zimbabwe' 'afghanistan'
 'estonia' 'malta' 'uruguay' 'belarus' 'colombia' 'republic of moldova'
 'isle of man' 'nomadic' 'new zealand' 'palestine' 'armenia'
 'united arab emirates' 'maldives' 'ethiopia' 'fiji' 'guatemala' 'uganda'
 'turkmenistan' 'mauritius' 'kenya' 'cuba' 'gabon' 'bahamas' 'south korea'
 'iceland' 'honduras' 'hong kong (s.a.r.)'
 "lao people's democratic republic" 'mongolia' 'cambodia' 'madagascar'
 'angola' 'democratic republic of the congo' 'syrian arab republic' 'iraq'
 'namibia' 'senegal' 'kyrgyzstan' 'zambia' 'swaziland' "côte d'ivoire"
 'kuwait' 'tajikistan' 'burundi' 'trinidad and tobago' 'mauritania'
 'sierra leone' 'panama' 'somalia' 'north korea' 'dominica' 'guyana'
 'togo' 'oman' 'barbados' 'andorra'
 "democratic people's republic of korea" 'qatar' 'sudan' 'cameroon'
 'papua new guinea' 'bahrain' 'yemen' 'malawi' 'burkina faso'
 'congo, republic of the...' 'botswana' 'guinea-bissau' 'mozambique'
 'central african republic' 'equatorial guinea' 'suriname' 'belize'
 'libyan arab jamahiriya' 'cape verde' 'brunei darussalam' 'bhutan'
 'guinea' 'niger' 'antigua and barbuda' 'mali' 'samoa' 'lesotho'
 'saint kitts and nevis' 'monaco' 'micronesia, federated states of...'
 'haiti' 'nan' 'nauru' 'liberia' 'chad' 'djibouti' 'solomon islands']

Value counts for 'Country_Cleaned' column (after basic cleaning):
Country_Cleaned
united states of america                               11095
nan                                                     6507
germany                                                 4947
india                                                   4231
united kingdom of great britain and northern ireland    3224
                                                       ...  
micronesia, federated states of...                         1
nauru                                                      1
chad                                                       1
djibouti                                                   1
solomon islands                                            1
Name: count, Length: 186, dtype: int64

Identifying irrelevant entries (e.g., 'n/a', 'unknown'):
```

3.2 Standardize entries in columns like Country or EdLevel by mapping inconsistent values to a consistent format.

In [12]:

```
## Write your code here
# --- New Section: 3.2 Standardize Entries in Categorical Columns ---
print("\n--- 3.2 Standardize entries in columns like Country or EdLevel by mapping inconsistent val

# --- Standardizing 'Country' Column ---
print("\nStandardizing 'Country' column:")
print("Original unique values in 'Country':", df['Country'].unique())

# Step 1: Convert to string, lowercase, and strip whitespace for initial consistency
df['Country'] = df['Country'].astype(str).str.lower().str.strip()

# Step 2: Define mapping for common inconsistencies
country_mapping = {
    'usa': 'United States',
    'u.s.a.': 'United States',
    'uk': 'United Kingdom',
    'n/a': np.nan # Treat 'N/A' as a missing value
}
df['Country'] = df['Country'].replace(country_mapping)

print("Unique values in 'Country' after standardization:")
print(df['Country'].unique())
print("Value counts for 'Country' after standardization:")
print(df['Country'].value_counts(dropna=False)) # Show counts including NaN if any

# --- Standardizing 'EdLevel' Column ---
print("\nStandardizing 'EdLevel' column:")
print("Original unique values in 'EdLevel':", df['EdLevel'].unique())

# Step 1: Convert to string, lowercase, and strip whitespace
df['EdLevel'] = df['EdLevel'].astype(str).str.lower().str.strip()

# Step 2: Define mapping for common inconsistencies
edlevel_mapping = {
    'some college': 'Some college/university study without earning a degree',
    'masters degree': 'Masters degree',
    'phd': 'PhD',
    'high school': 'Less than a Bachelors degree', # Assuming this maps to 'Less than a Bachelor's
    # Add more mappings as needed based on your dataset's specific inconsistencies
}
df['EdLevel'] = df['EdLevel'].replace(edlevel_mapping)

# Capitalize first letter of each word if desired for presentation
df['EdLevel'] = df['EdLevel'].str.title()

print("Unique values in 'EdLevel' after standardization:")
print(df['EdLevel'].unique())
print("Value counts for 'EdLevel' after standardization:")
print(df['EdLevel'].value_counts(dropna=False)) # Show counts including NaN if any

print("\nStandardization of categorical columns 'Country' and 'EdLevel' complete.")
print("This process helps in ensuring data consistency for accurate analysis and visualization.")
```

```
--- 3.2 Standardize entries in columns like Country or EdLevel by mapping inconsistent values to a c
onsistent format ---

Standardizing 'Country' column:
Original unique values in 'Country': ['United States of America'
 'United Kingdom of Great Britain and Northern Ireland' 'Canada' 'Norway'
 'Uzbekistan' 'Serbia' 'Poland' 'Philippines' 'Bulgaria' 'Switzerland'
 'India' 'Germany' 'Ireland' 'Italy' 'Ukraine' 'Australia' 'Brazil'
 'Japan' 'Austria' 'Iran, Islamic Republic of...' 'France' 'Saudi Arabia'
 'Romania' 'Turkey' 'Nepal' 'Algeria' 'Sweden' 'Netherlands' 'Croatia'
 'Pakistan' 'Czech Republic' 'Republic of North Macedonia' 'Finland'
 'Slovakia' 'Russian Federation' 'Greece' 'Israel' 'Belgium' 'Mexico'
 'United Republic of Tanzania' 'Hungary' 'Argentina' 'Portugal'
 'Sri Lanka' 'Latvia' 'China' 'Singapore' 'Lebanon' 'Spain' 'South Africa'
 'Lithuania' 'Viet Nam' 'Dominican Republic' 'Indonesia' 'Kosovo'
 'Morocco' 'Taiwan' 'Georgia' 'San Marino' 'Tunisia' 'Bangladesh'
 'Nigeria' 'Liechtenstein' 'Denmark' 'Ecuador' 'Malaysia' 'Albania'
 'Azerbaijan' 'Chile' 'Ghana' 'Peru' 'Bolivia' 'Egypt' 'Luxembourg'
 'Montenegro' 'Cyprus' 'Paraguay' 'Kazakhstan' 'Slovenia' 'Jordan'
 'Venezuela, Bolivarian Republic of...' 'Costa Rica' 'Jamaica' 'Thailand'
 'Nicaragua' 'Myanmar' 'Republic of Korea' 'Rwanda'
 'Bosnia and Herzegovina' 'Benin' 'El Salvador' 'Zimbabwe' 'Afghanistan'
 'Estonia' 'Malta' 'Uruguay' 'Belarus' 'Colombia' 'Republic of Moldova'
 'Isle of Man' 'Nomadic' 'New Zealand' 'Palestine' 'Armenia'
 'United Arab Emirates' 'Maldives' 'Ethiopia' 'Fiji' 'Guatemala' 'Uganda'
 'Turkmenistan' 'Mauritius' 'Kenya' 'Cuba' 'Gabon' 'Bahamas' 'South Korea'
 'Iceland' 'Honduras' 'Hong Kong (S.A.R.)'
 "Lao People's Democratic Republic" 'Mongolia' 'Cambodia' 'Madagascar'
 'Angola' 'Democratic Republic of the Congo' 'Syrian Arab Republic' 'Iraq'
 'Namibia' 'Senegal' 'Kyrgyzstan' 'Zambia' 'Swaziland' "Côte d'Ivoire"
 'Kuwait' 'Tajikistan' 'Burundi' 'Trinidad and Tobago' 'Mauritania'
 'Sierra Leone' 'Panama' 'Somalia' 'North Korea' 'Dominica' 'Guyana'
 'Togo' 'Oman' 'Barbados' 'Andorra'
 "Democratic People's Republic of Korea" 'Qatar' 'Sudan' 'Cameroon'
 'Papua New Guinea' 'Bahrain' 'Yemen' 'Malawi' 'Burkina Faso'
 'Congo, Republic of the...' 'Botswana' 'Guinea-Bissau' 'Mozambique'
 'Central African Republic' 'Equatorial Guinea' 'Suriname' 'Belize'
 'Libyan Arab Jamahiriya' 'Cape Verde' 'Brunei Darussalam' 'Bhutan'
 'Guinea' 'Niger' 'Antigua and Barbuda' 'Mali' 'Samoa' 'Lesotho'
 'Saint Kitts and Nevis' 'Monaco' 'Micronesia, Federated States of...'
 'Haiti' nan 'Nauru' 'Liberia' 'Chad' 'Djibouti' 'Solomon Islands']
Unique values in 'Country' after standardization:
['united states of america'
 'united kingdom of great britain and northern ireland' 'canada' 'norway'
 'uzbekistan' 'serbia' 'poland' 'philippines' 'bulgaria' 'switzerland'
 'india' 'germany' 'ireland' 'italy' 'ukraine' 'australia' 'brazil'
 'japan' 'austria' 'iran, islamic republic of...' 'france' 'saudi arabia'
 'romania' 'turkey' 'nepal' 'algeria' 'sweden' 'netherlands' 'croatia'
 'pakistan' 'czech republic' 'republic of north macedonia' 'finland'
 'slovakia' 'russian federation' 'greece' 'israel' 'belgium' 'mexico'
 'united republic of tanzania' 'hungary' 'argentina' 'portugal'
 'sri lanka' 'latvia' 'china' 'singapore' 'lebanon' 'spain' 'south africa'
 'lithuania' 'viet nam' 'dominican republic' 'indonesia' 'kosovo'
 'morocco' 'taiwan' 'georgia' 'san marino' 'tunisia' 'bangladesh'
 'nigeria' 'liechtenstein' 'denmark' 'ecuador' 'malaysia' 'albania'
 'azerbaijan' 'chile' 'ghana' 'peru' 'bolivia' 'egypt' 'luxembourg'
 'montenegro' 'cyprus' 'paraguay' 'kazakhstan' 'slovenia' 'jordan'
 'venezuela, bolivarian republic of...' 'costa rica' 'jamaica' 'thailand'
 'nicaragua' 'myanmar' 'republic of korea' 'rwanda'
 'bosnia and herzegovina' 'benin' 'el salvador' 'zimbabwe' 'afghanistan'
 'estonia' 'malta' 'uruguay' 'belarus' 'colombia' 'republic of moldova'
 'isle of man' 'nomadic' 'new zealand' 'palestine' 'armenia'
 'united arab emirates' 'maldives' 'ethiopia' 'fiji' 'guatemala' 'uganda'
 'turkmenistan' 'mauritius' 'kenya' 'cuba' 'gabon' 'bahamas' 'south korea'
 'iceland' 'honduras' 'hong kong (s.a.r.)'
 "lao people's democratic republic" 'mongolia' 'cambodia' 'madagascar'
 'angola' 'democratic republic of the congo' 'syrian arab republic' 'iraq'
 'namibia' 'senegal' 'kyrgyzstan' 'zambia' 'swaziland' "côte d'ivoire"
 'kuwait' 'tajikistan' 'burundi' 'trinidad and tobago' 'mauritania'
 'sierra leone' 'panama' 'somalia' 'north korea' 'dominica' 'guyana'
 'togo' 'oman' 'barbados' 'andorra'
 "democratic people's republic of korea" 'qatar' 'sudan' 'cameroon'
 'papua new guinea' 'bahrain' 'yemen' 'malawi' 'burkina faso'
 'congo, republic of the...' 'botswana' 'guinea-bissau' 'mozambique'
 'central african republic' 'equatorial guinea' 'suriname' 'belize'
 'libyan arab jamahiriya' 'cape verde' 'brunei darussalam' 'bhutan'
 'guinea' 'niger' 'antigua and barbuda' 'mali' 'samoa' 'lesotho'
 'saint kitts and nevis' 'monaco' 'micronesia, federated states of...'
 'haiti' 'nan' 'nauru' 'liberia' 'chad' 'djibouti' 'solomon islands']
Value counts for 'Country' after standardization:
Country
united states of america                                11095
nan                                                      6507
germany                                                  4947
india                                                    4231
united kingdom of great britain and northern ireland     3224
                                                        ...  
micronesia, federated states of...                          1
nauru                                                       1
chad                                                        1
djibouti                                                    1
solomon islands                                             1
Name: count, Length: 186, dtype: int64

Standardizing 'EdLevel' column:
Original unique values in 'EdLevel': ['Primary/elementary school'
 'Bachelor’s degree (B.A., B.S., B.Eng., etc.)'
 'Master’s degree (M.A., M.S., M.Eng., MBA, etc.)'
 'Some college/university study without earning a degree'
 'Secondary school (e.g. American high school, German Realschule or Gymnasium, etc.)'
 'Professional degree (JD, MD, Ph.D, Ed.D, etc.)'
 'Associate degree (A.A., A.S., etc.)' 'Something else' nan]
Unique values in 'EdLevel' after standardization:
['Primary/Elementary School'
 'Bachelor’S Degree (B.A., B.S., B.Eng., Etc.)'
 'Master’S Degree (M.A., M.S., M.Eng., Mba, Etc.)'
 'Some College/University Study Without Earning A Degree'
 'Secondary School (E.G. American High School, German Realschule Or Gymnasium, Etc.)'
 'Professional Degree (Jd, Md, Ph.D, Ed.D, Etc.)'
 'Associate Degree (A.A., A.S., Etc.)' 'Something Else' 'Nan']
Value counts for 'EdLevel' after standardization:
EdLevel
Bachelor’S Degree (B.A., B.S., B.Eng., Etc.)                                          24942
Master’S Degree (M.A., M.S., M.Eng., Mba, Etc.)                                       15557
Some College/University Study Without Earning A Degree                                 7651
Secondary School (E.G. American High School, German Realschule Or Gymnasium, Etc.)     5793
Nan                                                                                    4653
Professional Degree (Jd, Md, Ph.D, Ed.D, Etc.)                                         2970
Associate Degree (A.A., A.S., Etc.)                                                    1793
Primary/Elementary School                                                              1146
Something Else                                                                          932
Name: count, dtype: int64

Standardization of categorical columns 'Country' and 'EdLevel' complete.
This process helps in ensuring data consistency for accurate analysis and visualization.
```

4. Encoding Categorical Variables

4.1 Encode the Employment column using one-hot encoding.

In [13]:

```python
## Write your code here
# --- New Section: 4.1 Encode the Employment column using one-hot encoding ---
print("\n--- 4.1 Encode the Employment column using one-hot encoding ---")

# Display original Employment column info and head
print("\nOriginal 'Employment' column head:")
print(df['Employment'].head())
print("\nOriginal 'Employment' unique values:")
print(df['Employment'].unique())

# Perform one-hot encoding on the 'Employment' column
# drop_first=True is often used to avoid multicollinearity if you're using this for modeling.
# It drops one of the generated columns, as the information is implicitly contained in the others.
employment_encoded = pd.get_dummies(df['Employment'], prefix='Employment', drop_first=False)

# Concatenate the new one-hot encoded columns with the original DataFrame
df = pd.concat([df, employment_encoded], axis=1)
```

```
# Drop the original 'Employment' column if you no longer need it
# df = df.drop('Employment', axis=1)

print("\nDataFrame head after one-hot encoding 'Employment' column (showing new columns):")
print(df.head())

print(f"\nNew columns created from 'Employment' encoding: {employment_encoded.columns.tolist()}")
print("\nShape of DataFrame after one-hot encoding:", df.shape)
print("One-hot encoding of 'Employment' column complete. This converts categorical data into a nume
--- 4.1 Encode the Employment column using one-hot encoding ---

Original 'Employment' column head:
0    Employed, full-time
1    Employed, full-time
2    Employed, full-time
3     Student, full-time
4     Student, full-time
Name: Employment, dtype: object

Original 'Employment' unique values:
['Employed, full-time' 'Student, full-time'
 'Student, full-time;Not employed, but looking for work'
 'Independent contractor, freelancer, or self-employed'
 'Not employed, and not looking for work'
 'Employed, full-time;Student, part-time'
 'Employed, full-time;Independent contractor, freelancer, or self-employed'
 'Employed, full-time;Student, full-time' 'Employed, part-time'
 'Student, full-time;Employed, part-time'
 'Student, part-time;Employed, part-time' 'I prefer not to say'
 'Not employed, but looking for work' 'Student, part-time'
 'Employed, full-time;Student, full-time;Independent contractor, freelancer, or self-employed;Employ
ed, part-time'
 'Employed, full-time;Independent contractor, freelancer, or self-employed;Student, part-time'
 'Independent contractor, freelancer, or self-employed;Employed, part-time'
 'Independent contractor, freelancer, or self-employed;Student, part-time;Employed, part-time'
 'Student, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-
employed'
 'Student, full-time;Independent contractor, freelancer, or self-employed'
 'Employed, full-time;Employed, part-time'
 'Not employed, but looking for work;Independent contractor, freelancer, or self-employed'
 'Student, full-time;Not employed, and not looking for work' 'Retired'
 'Independent contractor, freelancer, or self-employed;Student, part-time'
 'Employed, full-time;Independent contractor, freelancer, or self-employed;Employed, part-time'
 'Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Student, p
art-time'
 'Not employed, but looking for work;Student, part-time'
 'Not employed, but looking for work;Not employed, and not looking for work'
 'Independent contractor, freelancer, or self-employed;Retired'
 'Not employed, but looking for work;Student, part-time;Employed, part-time'
 'Student, full-time;Not employed, but looking for work;Not employed, and not looking for work'
 'Employed, full-time;Not employed, but looking for work'
 'Student, full-time;Not employed, and not looking for work;Student, part-time'
 'Employed, full-time;Retired'
 'Employed, full-time;Independent contractor, freelancer, or self-employed;Student, part-time;Employ
ed, part-time'
 'Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Not employ
ed, and not looking for work'
 'Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Employed,
part-time'
 'Not employed, but looking for work;Employed, part-time'
 'Employed, full-time;Student, full-time;Employed, part-time'
 'Independent contractor, freelancer, or self-employed;Not employed, and not looking for work'
 'Not employed, and not looking for work;Student, part-time'
 'Student, full-time;Independent contractor, freelancer, or self-employed;Employed, part-time'
 'Student, full-time;Student, part-time'
 'Student, full-time;Not employed, but looking for work;Student, part-time'
 'Independent contractor, freelancer, or self-employed;Not employed, and not looking for work;Retire
d'
 'Employed, full-time;Independent contractor, freelancer, or self-employed;Not employed, and not loo
king for work'
 'Employed, full-time;Student, full-time;Independent contractor, freelancer, or self-employed'
 'Employed, full-time;Student, full-time;Student, part-time'
 'Not employed, but looking for work;Retired'
 'Employed, full-time;Student, full-time;Not employed, but looking for work'
 'Not employed, and not looking for work;Retired'
 'Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Not employ
ed, and not looking for work;Retired'
 'Employed, full-time;Not employed, but looking for work;Employed, part-time'
 'Student, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-
employed;Student, part-time;Employed, part-time;Retired'
 'Employed, full-time;Independent contractor, freelancer, or self-employed;Not employed, and not loo
king for work;Employed, part-time'
 'Student, full-time;Independent contractor, freelancer, or self-employed;Not employed, and not look
ing for work'
```

```
 'Employed, full-time;Student, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Not employed, and not looking for work;Student, part-time;Employed, part-time;Retired'
 'Employed, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-employed'
 'Independent contractor, freelancer, or self-employed;Not employed, and not looking for work;Student, part-time'
 'Student, full-time;Not employed, but looking for work;Retired'
 'Student, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Student, part-time'
 'Student, part-time;Retired'
 'Student, full-time;Not employed, but looking for work;Not employed, and not looking for work;Student, part-time'
 'Employed, full-time;Student, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Student, part-time;Employed, part-time'
 'Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Retired'
 'Employed, full-time;Student, full-time;Student, part-time;Employed, part-time'
 'Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Student, part-time;Employed, part-time'
 'Student, full-time;Not employed, but looking for work;Employed, part-time'
 'Employed, full-time;Independent contractor, freelancer, or self-employed;Not employed, and not looking for work;Student, part-time'
 'Independent contractor, freelancer, or self-employed;Student, part-time;Retired'
 'Student, full-time;Independent contractor, freelancer, or self-employed;Student, part-time;Employed, part-time'
 'Employed, full-time;Independent contractor, freelancer, or self-employed;Student, part-time;Retired'
 'Student, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Not employed, and not looking for work'
 'Student, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Employed, part-time'
 'Student, full-time;Independent contractor, freelancer, or self-employed;Student, part-time'
 'Independent contractor, freelancer, or self-employed;Employed, part-time;Retired'
 'Employed, full-time;Not employed, and not looking for work'
 'Employed, full-time;Independent contractor, freelancer, or self-employed;Retired'
 'Student, full-time;Student, part-time;Employed, part-time'
 'Employed, part-time;Retired'
 'Employed, full-time;Independent contractor, freelancer, or self-employed;Employed, part-time;Retired'
 'Employed, full-time;Student, part-time;Employed, part-time'
 'Employed, full-time;Student, full-time;Independent contractor, freelancer, or self-employed;Student, part-time;Employed, part-time;Retired'
 'Student, full-time;Student, part-time;Retired'
 'Student, full-time;Not employed, and not looking for work;Employed, part-time'
 'Employed, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Employed, part-time'
 'Not employed, but looking for work;Not employed, and not looking for work;Student, part-time;Employed, part-time'
 'Independent contractor, freelancer, or self-employed;Not employed, and not looking for work;Employed, part-time'
 'Employed, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Not employed, and not looking for work;Employed, part-time'
 'Employed, full-time;Student, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Not employed, and not looking for work;Student, part-time;Employed, part-time'
 'Employed, full-time;Student, full-time;Independent contractor, freelancer, or self-employed;Student, part-time;Employed, part-time'
 'Not employed, and not looking for work;Employed, part-time'
 'Employed, full-time;Student, full-time;Not employed, but looking for work;Student, part-time'
 'Employed, full-time;Student, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Employed, part-time'
 'Employed, full-time;Not employed, but looking for work;Not employed, and not looking for work;Employed, part-time'
 'Student, full-time;Independent contractor, freelancer, or self-employed;Employed, part-time;Retired'
 'Not employed, but looking for work;Student, part-time;Retired'
 'Independent contractor, freelancer, or self-employed;Not employed, and not looking for work;Student, part-time;Retired'
 'Employed, full-time;Student, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-employed'
 'Not employed, but looking for work;Not employed, and not looking for work;Student, part-time'
 'Employed, full-time;Student, full-time;Independent contractor, freelancer, or self-employed;Student, part-time;Retired'
 'Employed, full-time;Student, full-time;Not employed, but looking for work;Student, part-time;Employed, part-time'
 'Student, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-
employed;Not employed, and not looking for work;Student, part-time'
 'Employed, full-time;Student, full-time;Not employed, but looking for work;Independent contractor,
freelancer, or self-employed;Student, part-time;Employed, part-time;Retired'
 'Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Not employ
ed, and not looking for work;Employed, part-time'
 'Student, full-time;Retired'
 'Employed, full-time;Not employed, but looking for work;Student, part-time'
 'Not employed, and not looking for work;Student, part-time;Employed, part-time'
 'Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Student, p
art-time;Retired']

DataFrame head after one-hot encoding 'Employment' column (showing new columns):
   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  ...  \
0                                                NaN  ...
1  Technical documentation;Blogs;Books;Written Tu...  ...
2  Technical documentation;Blogs;Books;Written Tu...  ...
3  Stack Overflow;How-to videos;Interactive tutorial  ...
4  Technical documentation;Blogs;Written Tutorial...  ...

  Employment_Student, full-time;Not employed, but looking for work;Not employed, and not looking for
work;Student, part-time  \
0                                              False
1                                              False
2                                              False
3                                              False
4                                              False

  Employment_Student, full-time;Not employed, but looking for work;Retired  \
0                                              False
1                                              False
2                                              False
3                                              False
4                                              False

  Employment_Student, full-time;Not employed, but looking for work;Student, part-time  \
0                                              False
1                                              False
2                                              False
3                                              False
```

```
4                                              False

  Employment_Student, full-time;Retired  \
0                                 False
1                                 False
2                                 False
3                                 False
4                                 False

  Employment_Student, full-time;Student, part-time  \
0                                            False
1                                            False
2                                            False
3                                            False
4                                            False

  Employment_Student, full-time;Student, part-time;Employed, part-time  \
0                                              False
1                                              False
2                                              False
3                                              False
4                                              False

  Employment_Student, full-time;Student, part-time;Retired  \
0                                              False
1                                              False
2                                              False
3                                              False
4                                              False

  Employment_Student, part-time  \
0                         False
1                         False
2                         False
3                         False
4                         False

  Employment_Student, part-time;Employed, part-time  \
0                                             False
1                                             False
2                                             False
3                                             False
4                                             False

  Employment_Student, part-time;Retired
0                                 False
1                                 False
2                                 False
3                                 False
4                                 False

[5 rows x 225 columns]

New columns created from 'Employment' encoding: ['Employment_Employed, full-time', 'Employment_Emplo
yed, full-time;Employed, part-time', 'Employment_Employed, full-time;Independent contractor, freelan
cer, or self-employed', 'Employment_Employed, full-time;Independent contractor, freelancer, or self-
employed;Employed, part-time', 'Employment_Employed, full-time;Independent contractor, freelancer, o
r self-employed;Employed, part-time;Retired', 'Employment_Employed, full-time;Independent contracto
r, freelancer, or self-employed;Not employed, and not looking for work', 'Employment_Employed, full-
time;Independent contractor, freelancer, or self-employed;Not employed, and not looking for work;Emp
loyed, part-time', 'Employment_Employed, full-time;Independent contractor, freelancer, or self-emplo
yed;Not employed, and not looking for work;Student, part-time', 'Employment_Employed, full-time;Inde
pendent contractor, freelancer, or self-employed;Retired', 'Employment_Employed, full-time;Independe
nt contractor, freelancer, or self-employed;Student, part-time', 'Employment_Employed, full-time;Ind
ependent contractor, freelancer, or self-employed;Student, part-time;Employed, part-time', 'Employme
nt_Employed, full-time;Independent contractor, freelancer, or self-employed;Student, part-time;Retir
ed', 'Employment_Employed, full-time;Not employed, and not looking for work', 'Employment_Employed, 
full-time;Not employed, but looking for work', 'Employment_Employed, full-time;Not employed, but loo
king for work;Employed, part-time', 'Employment_Employed, full-time;Not employed, but looking for wo
rk;Independent contractor, freelancer, or self-employed', 'Employment_Employed, full-time;Not employ
ed, but looking for work;Independent contractor, freelancer, or self-employed;Employed, part-time', 
'Employment_Employed, full-time;Not employed, but looking for work;Independent contractor, freelance
r, or self-employed;Not employed, and not looking for work;Employed, part-time', 'Employment_Employe
d, full-time;Not employed, but looking for work;Not employed, and not looking for work;Employed, par
t-time', 'Employment_Employed, full-time;Not employed, but looking for work;Student, part-time', 'Em
```

```
ployment_Employed, full-time;Retired', 'Employment_Employed, full-time;Student, full-time', 'Employm
ent_Employed, full-time;Student, full-time;Employed, part-time', 'Employment_Employed, full-time;Stu
dent, full-time;Independent contractor, freelancer, or self-employed', 'Employment_Employed, full-ti
me;Student, full-time;Independent contractor, freelancer, or self-employed;Employed, part-time', 'Em
ployment_Employed, full-time;Student, full-time;Independent contractor, freelancer, or self-employe
d;Student, part-time;Employed, part-time', 'Employment_Employed, full-time;Student, full-time;Indepe
ndent contractor, freelancer, or self-employed;Student, part-time;Employed, part-time;Retired', 'Emp
loyment_Employed, full-time;Student, full-time;Independent contractor, freelancer, or self-employed;
Student, part-time;Retired', 'Employment_Employed, full-time;Student, full-time;Not employed, but lo
oking for work', 'Employment_Employed, full-time;Student, full-time;Not employed, but looking for wo
rk;Independent contractor, freelancer, or self-employed', 'Employment_Employed, full-time;Student, f
ull-time;Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Emp
loyed, part-time', 'Employment_Employed, full-time;Student, full-time;Not employed, but looking for 
work;Independent contractor, freelancer, or self-employed;Not employed, and not looking for work;Stu
dent, part-time;Employed, part-time', 'Employment_Employed, full-time;Student, full-time;Not employe
d, but looking for work;Independent contractor, freelancer, or self-employed;Not employed, and not l
ooking for work;Student, part-time;Employed, part-time;Retired', 'Employment_Employed, full-time;Stu
dent, full-time;Not employed, but looking for work;Independent contractor, freelancer, or self-emplo
yed;Student, part-time;Employed, part-time', 'Employment_Employed, full-time;Student, full-time;Not 
employed, but looking for work;Independent contractor, freelancer, or self-employed;Student, part-ti
me;Employed, part-time;Retired', 'Employment_Employed, full-time;Student, full-time;Not employed, bu
t looking for work;Student, part-time', 'Employment_Employed, full-time;Student, full-time;Not emplo
yed, but looking for work;Student, part-time;Employed, part-time', 'Employment_Employed, full-time;S
tudent, full-time;Student, part-time', 'Employment_Employed, full-time;Student, full-time;Student, p
art-time;Employed, part-time', 'Employment_Employed, full-time;Student, part-time', 'Employment_Empl
oyed, full-time;Student, part-time;Employed, part-time', 'Employment_Employed, part-time', 'Employme
nt_Employed, part-time;Retired', 'Employment_I prefer not to say', 'Employment_Independent contracto
r, freelancer, or self-employed', 'Employment_Independent contractor, freelancer, or self-employed;E
mployed, part-time', 'Employment_Independent contractor, freelancer, or self-employed;Employed, part
-time;Retired', 'Employment_Independent contractor, freelancer, or self-employed;Not employed, and n
ot looking for work', 'Employment_Independent contractor, freelancer, or self-employed;Not employed,
 and not looking for work;Employed, part-time', 'Employment_Independent contractor, freelancer, or se
lf-employed;Not employed, and not looking for work;Retired', 'Employment_Independent contractor, fre
elancer, or self-employed;Not employed, and not looking for work;Student, part-time', 'Employment_In
dependent contractor, freelancer, or self-employed;Not employed, and not looking for work;Student, p
art-time;Retired', 'Employment_Independent contractor, freelancer, or self-employed;Retired', 'Emplo
yment_Independent contractor, freelancer, or self-employed;Student, part-time', 'Employment_Independ
ent contractor, freelancer, or self-employed;Student, part-time;Employed, part-time', 'Employment_In
dependent contractor, freelancer, or self-employed;Student, part-time;Retired', 'Employment_Not empl
oyed, and not looking for work', 'Employment_Not employed, and not looking for work;Employed, part-t
ime', 'Employment_Not employed, and not looking for work;Retired', 'Employment_Not employed, and not
 looking for work;Student, part-time', 'Employment_Not employed, and not looking for work;Student, pa
rt-time;Employed, part-time', 'Employment_Not employed, but looking for work', 'Employment_Not emplo
yed, but looking for work;Employed, part-time', 'Employment_Not employed, but looking for work;Indep
endent contractor, freelancer, or self-employed', 'Employment_Not employed, but looking for work;Ind
ependent contractor, freelancer, or self-employed;Employed, part-time', 'Employment_Not employed, bu
t looking for work;Independent contractor, freelancer, or self-employed;Not employed, and not lookin
g for work', 'Employment_Not employed, but looking for work;Independent contractor, freelancer, or s
elf-employed;Not employed, and not looking for work;Employed, part-time', 'Employment_Not employed,
 but looking for work;Independent contractor, freelancer, or self-employed;Not employed, and not look
ing for work;Retired', 'Employment_Not employed, but looking for work;Independent contractor, freela
ncer, or self-employed;Retired', 'Employment_Not employed, but looking for work;Independent contract
or, freelancer, or self-employed;Student, part-time', 'Employment_Not employed, but looking for wor
k;Independent contractor, freelancer, or self-employed;Student, part-time;Employed, part-time', 'Emp
loyment_Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Stud
ent, part-time;Retired', 'Employment_Not employed, but looking for work;Not employed, and not lookin
g for work', 'Employment_Not employed, but looking for work;Not employed, and not looking for work;S
tudent, part-time', 'Employment_Not employed, but looking for work;Not employed, and not looking for
work;Student, part-time;Employed, part-time', 'Employment_Not employed, but looking for work;Retire
d', 'Employment_Not employed, but looking for work;Student, part-time', 'Employment_Not employed, bu
t looking for work;Student, part-time;Employed, part-time', 'Employment_Not employed, but looking fo
r work;Student, part-time;Retired', 'Employment_Retired', 'Employment_Student, full-time', 'Employme
nt_Student, full-time;Employed, part-time', 'Employment_Student, full-time;Independent contractor, f
reelancer, or self-employed', 'Employment_Student, full-time;Independent contractor, freelancer, or 
self-employed;Employed, part-time', 'Employment_Student, full-time;Independent contractor, freelance
r, or self-employed;Employed, part-time;Retired', 'Employment_Student, full-time;Independent contrac
tor, freelancer, or self-employed;Not employed, and not looking for work', 'Employment_Student, full
-time;Independent contractor, freelancer, or self-employed;Student, part-time', 'Employment_Student,
 full-time;Independent contractor, freelancer, or self-employed;Student, part-time;Employed, part-tim
e', 'Employment_Student, full-time;Not employed, and not looking for work', 'Employment_Student, ful
l-time;Not employed, and not looking for work;Employed, part-time', 'Employment_Student, full-time;N
ot employed, and not looking for work;Student, part-time', 'Employment_Student, full-time;Not employ
ed, but looking for work', 'Employment_Student, full-time;Not employed, but looking for work;Employe
d, part-time', 'Employment_Student, full-time;Not employed, but looking for work;Independent contrac
tor, freelancer, or self-employed', 'Employment_Student, full-time;Not employed, but looking for wor
```

```
k;Independent contractor, freelancer, or self-employed;Employed, part-time', 'Employment_Student, fu
ll-time;Not employed, but looking for work;Independent contractor, freelancer, or self-employed;Not 
employed, and not looking for work', 'Employment_Student, full-time;Not employed, but looking for wo
rk;Independent contractor, freelancer, or self-employed;Not employed, and not looking for work;Stude
nt, part-time', 'Employment_Student, full-time;Not employed, but looking for work;Independent contra
ctor, freelancer, or self-employed;Student, part-time', 'Employment_Student, full-time;Not employed, 
but looking for work;Independent contractor, freelancer, or self-employed;Student, part-time;Employe
d, part-time;Retired', 'Employment_Student, full-time;Not employed, but looking for work;Not employe
d, and not looking for work', 'Employment_Student, full-time;Not employed, but looking for work;Not 
employed, and not looking for work;Student, part-time', 'Employment_Student, full-time;Not employed, 
but looking for work;Retired', 'Employment_Student, full-time;Not employed, but looking for work;Stu
dent, part-time', 'Employment_Student, full-time;Retired', 'Employment_Student, full-time;Student, p
art-time', 'Employment_Student, full-time;Student, part-time;Employed, part-time', 'Employment_Stude
nt, full-time;Student, part-time;Retired', 'Employment_Student, part-time', 'Employment_Student, par
t-time;Employed, part-time', 'Employment_Student, part-time;Retired']

Shape of DataFrame after one-hot encoding: (65437, 225)
One-hot encoding of 'Employment' column complete. This converts categorical data into a numerical fo
rmat suitable for machine learning models.
```

5. Handling Missing Values

5.1 Identify columns with the highest number of missing values.

In [14]:

```python
## Write your code here
# 5.1 Identify columns with the highest number of missing values.
print("\n5.1 Identify columns with the highest number of missing values.")
missing_values_count = df.isnull().sum()
# Filter to show only columns with missing values and sort them
columns_with_missing = missing_values_count[missing_values_count > 0].sort_values(ascending=False)

if not columns_with_missing.empty:
    print("\nColumns with missing values (highest first):")
    print(columns_with_missing)
    # Get the column with the absolute highest number of missing values
    highest_missing_column = columns_with_missing.index[0]
    print(f"\nColumn with the highest number of missing values: '{highest_missing_column}' ({column
else:
    print("\nNo missing values found in the dataset.")
```

```
5.1 Identify columns with the highest number of missing values.

Columns with missing values (highest first):
AINextMuch less integrated    64289
AINextLess integrated         63082
AINextNo change               52939
AINextMuch more integrated    51999
EmbeddedAdmired               48704
                              ...  
LanguageHaveWorkedWith         5692
YearsCode                      5568
NEWSOSites                     5151
LearnCode                      4949
AISelect                       4530
Length: 107, dtype: int64

Column with the highest number of missing values: 'AINextMuch less integrated' (64289 missing)
```

5.2 Impute missing values in numerical columns (e.g., `ConvertedCompYearly`) with the mean or median.

In [15]:

```python
## Write your code here
# 5.2 Impute missing values in numerical columns (e.g., ConvertedCompYearly) with the mean or media
print("\n5.2 Impute missing values in numerical columns with mean or median.")
# For 'ConvertedCompYearly', median is generally preferred for skewed data like income
# to be less affected by extreme outliers.
if 'ConvertedCompYearly' in df.columns and df['ConvertedCompYearly'].isnull().any():
    median_comp = df['ConvertedCompYearly'].median()
    df['ConvertedCompYearly'].fillna(median_comp, inplace=True)
    print(f"Missing values in 'ConvertedCompYearly' imputed with median: {median_comp:.2f}")
else:
    print("'ConvertedCompYearly' column not found or has no missing values. No imputation performed
```

```
5.2 Impute missing values in numerical columns with mean or median.
Missing values in 'ConvertedCompYearly' imputed with median: 65000.00
```

```
/tmp/ipykernel_2452/3683777211.py:8: 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['ConvertedCompYearly'].fillna(median_comp, inplace=True)
```

5.3 Impute missing values in categorical columns (e.g., `RemoteWork`) with the most frequent value.

In [16]:

```
## Write your code here
# 5.3 Impute missing values in categorical columns (e.g., RemoteWork) with the most frequent value.
print("\n5.3 Impute missing values in categorical columns with the most frequent value.")
if 'RemoteWork' in df.columns and df['RemoteWork'].isnull().any():
    most_frequent_remotework = df['RemoteWork'].mode()[0]
    df['RemoteWork'].fillna(most_frequent_remotework, inplace=True)
    print(f"Missing values in 'RemoteWork' imputed with most frequent value: '{most_frequent_remote
else:
    print("'RemoteWork' column not found or has no missing values. No imputation performed.")

# Verify missing values after handling
print("\nMissing values after Section 5 imputation:")
print(df.isnull().sum()[df.isnull().sum() > 0]) # Show only columns that still have NaNs
```

```
5.3 Impute missing values in categorical columns with the most frequent value.
Missing values in 'RemoteWork' imputed with most frequent value: 'Hybrid (some remote, some in-perso
n)'

Missing values after Section 5 imputation:
```

```
/tmp/ipykernel_2452/2099079762.py:6: 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['RemoteWork'].fillna(most_frequent_remotework, inplace=True)
```

```
CodingActivities    10971
LearnCode            4949
LearnCodeOnline     16200
TechDoc             24540
YearsCode            5568
                    ...  
JobSatPoints_10     35987
JobSatPoints_11     35992
SurveyLength         9255
SurveyEase           9199
JobSat              36311
Length: 105, dtype: int64
```

6. Feature Scaling and Transformation

6.1 Apply Min-Max Scaling to normalize the `ConvertedCompYearly` column.

In [24]:

```
## Write your code here

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from scipy.stats import skew

#6.1 Apply Min-Max Scaling to normalize the ConvertedCompYearly column.
print("\n6.1 Apply Min-Max Scaling to normalize the 'ConvertedCompYearly' column.")
if 'ConvertedCompYearly' in df.columns and pd.api.types.is_numeric_dtype(df['ConvertedCompYearly'])
```

```
    scaler_minmax = MinMaxScaler()
    # Assign to a NEW column to keep original, and match print statement's intention
    df['ConvertedCompYearly_MinMax'] = scaler_minmax.fit_transform(df[['ConvertedCompYearly']])
    print("Min-Max Scaling applied. New column 'ConvertedCompYearly_MinMax' created.")
    print("Descriptive stats for 'ConvertedCompYearly_MinMax':")
    # Corrected column name in .describe()
    print(df['ConvertedCompYearly_MinMax'].describe())
else:
    print("'ConvertedCompYearly' not suitable for Min-Max Scaling (not found or not numeric).")
```

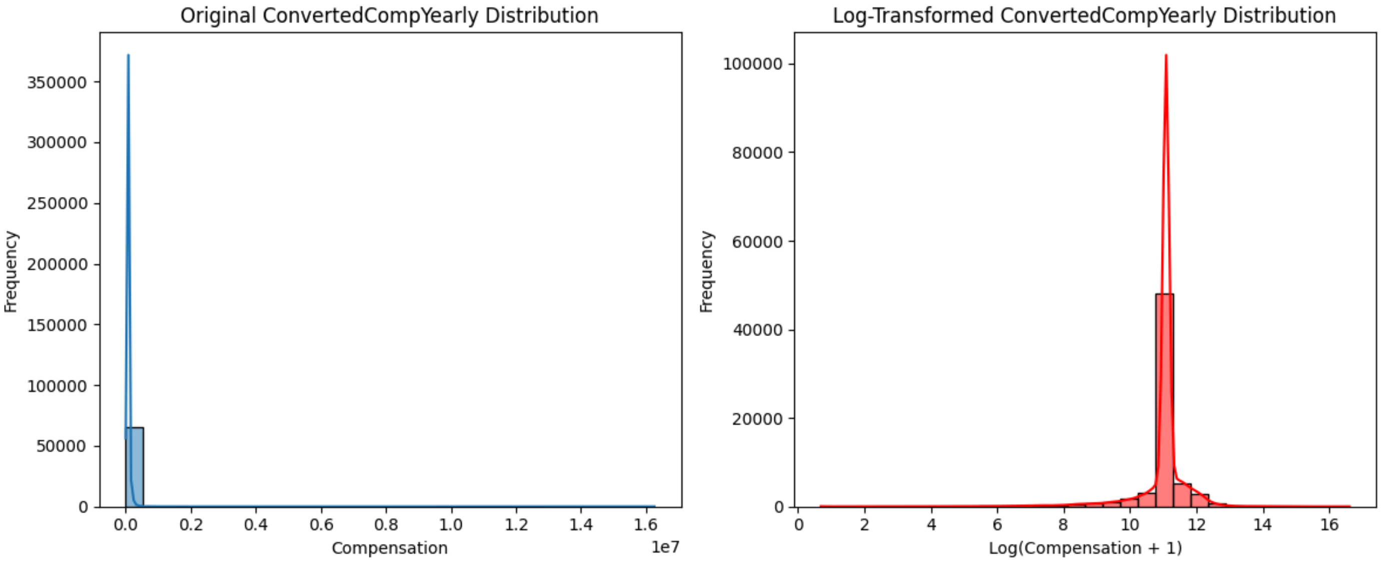
```
6.1 Apply Min-Max Scaling to normalize the 'ConvertedCompYearly' column.
Min-Max Scaling applied. New column 'ConvertedCompYearly_MinMax' created.
Descriptive stats for 'ConvertedCompYearly_MinMax':
count    65437.000000
mean         0.004464
std          0.006903
min          0.000000
25%          0.003998
50%          0.003998
75%          0.003998
max          1.000000
Name: ConvertedCompYearly_MinMax, dtype: float64
```

6.2 Log-transform the ConvertedCompYearly column to reduce skewness.

In [19]:

```
## Write your code here
print("\n6.2 Log-transform the ConvertedCompYearly column to reduce skewness.")
if 'ConvertedCompYearly' in df.columns and pd.api.types.is_numeric_dtype(df['ConvertedCompYearly'])
    # Check skewness before transformation (optional)
    original_skew = skew(df['ConvertedCompYearly'].dropna())
    print(f"Original skewness of 'ConvertedCompYearly': {original_skew:.2f}")

    # Apply log transformation. Add 1 to handle zero or negative values if they exist, to avoid log
    df['ConvertedCompYearly_Log'] = np.log1p(df['ConvertedCompYearly'])
    print("Log transformation applied. New column 'ConvertedCompYearly_Log' created.")
    print("Descriptive stats for 'ConvertedCompYearly_Log':")
    print(df['ConvertedCompYearly_Log'].describe())

    # Check skewness after transformation (optional)
    transformed_skew = skew(df['ConvertedCompYearly_Log'].dropna())
    print(f"Skewness of 'ConvertedCompYearly_Log' after transformation: {transformed_skew:.2f}")

    # Visualize original vs log-transformed distribution to show effect on skewness
    plt.figure(figsize=(12, 5))
    plt.subplot(1, 2, 1)
    sns.histplot(df['ConvertedCompYearly'], kde=True, bins=30)
    plt.title('Original ConvertedCompYearly Distribution')
    plt.xlabel('Compensation')
    plt.ylabel('Frequency')

    plt.subplot(1, 2, 2)
    sns.histplot(df['ConvertedCompYearly_Log'], kde=True, bins=30, color='red')
    plt.title('Log-Transformed ConvertedCompYearly Distribution')
    plt.xlabel('Log(Compensation + 1)')
    plt.ylabel('Frequency')
    plt.tight_layout()
    plt.show()

else:
    print("'ConvertedCompYearly' not suitable for log transformation (not found or not numeric).")
```

```
6.2 Log-transform the ConvertedCompYearly column to reduce skewness.
Original skewness of 'ConvertedCompYearly': 87.71
Log transformation applied. New column 'ConvertedCompYearly_Log' created.
Descriptive stats for 'ConvertedCompYearly_Log':
count    65437.000000
mean        10.976053
std          0.851456
min          0.693147
25%         11.082158
50%         11.082158
75%         11.082158
max         16.604010
Name: ConvertedCompYearly_Log, dtype: float64
Skewness of 'ConvertedCompYearly_Log' after transformation: -4.28
```



7. Feature Engineering

7.1 Create a new column `ExperienceLevel` based on the `YearsCodePro` column:

In [20]:

```python
print("\n--- 7. Feature Engineering ---")

# 7.1 Create a new column ExperienceLevel based on the YearsCodePro column:
print("\n7.1 Create a new column 'ExperienceLevel' based on the 'YearsCodePro' column.")

# Ensure 'YearsCodePro' is numeric and handle NaNs (from 5.2 or earlier)
if 'YearsCodePro' in df.columns and pd.api.types.is_numeric_dtype(df['YearsCodePro']):
    # Define conditions for different experience levels
    conditions = [
        (df['YearsCodePro'] < 3),
        (df['YearsCodePro'] >= 3) & (df['YearsCodePro'] < 10),
        (df['YearsCodePro'] >= 10) & (df['YearsCodePro'] < 20),
        (df['YearsCodePro'] >= 20)
    ]
    # Define corresponding experience level labels
    choices = ['Junior', 'Mid-level', 'Senior', 'Lead/Principal']

    # Use numpy.select to apply conditions and assign labels
    df['ExperienceLevel'] = np.select(conditions, choices, default='Unknown')
    print("New column 'ExperienceLevel' created based on 'YearsCodePro'.")
    print("\nValue counts for 'ExperienceLevel':")
    print(df['ExperienceLevel'].value_counts(dropna=False)) # dropna=False to see 'Unknown' or actu

    print("\nSample of 'YearsCodePro' and 'ExperienceLevel':")
    print(df[['YearsCodePro', 'ExperienceLevel']].head(10))

else:
    print("'YearsCodePro' column not found or is not numeric. Cannot create 'ExperienceLevel'.")

# Summary statement as per PDF (no code needed here)
print("\n--- Summary ---")
print("In this lab, you:")
print("• Explored the dataset to identify inconsistencies and missing values.")
print("• Encoded categorical variables for analysis.")
print("• Handled missing values using imputation techniques.")
print("• Normalized and transformed numerical data to prepare it for analysis.")
print("• Engineered a new feature to enhance data interpretation.")
```

```
--- 7. Feature Engineering ---

7.1 Create a new column 'ExperienceLevel' based on the 'YearsCodePro' column.
'YearsCodePro' column not found or is not numeric. Cannot create 'ExperienceLevel'.

--- Summary ---
In this lab, you:
• Explored the dataset to identify inconsistencies and missing values.
• Encoded categorical variables for analysis.
• Handled missing values using imputation techniques.
• Normalized and transformed numerical data to prepare it for analysis.
• Engineered a new feature to enhance data interpretation.
```

Summary

In this lab, you:

- Explored the dataset to identify inconsistencies and missing values.
- Encoded categorical variables for analysis.
- Handled missing values using imputation techniques.
- Normalized and transformed numerical data to prepare it for analysis.
- Engineered a new feature to enhance data interpretation.