Finding Correlation

Estimated time needed: **30** minutes

In this lab, you will work with a cleaned dataset to perform exploratory data analysis (EDA). You will examine the distribution of the data, identify outliers, and determine the correlation between different columns in the dataset.

Objectives

In this lab, you will perform the following:

- Identify the distribution of compensation data in the dataset.
- Remove outliers to refine the dataset.
- Identify correlations between various features in the dataset.

Hands on Lab

Step 1: Install and Import Required Libraries

In [1]:
```
# Install the necessary libraries
!pip install pandas
!pip install matplotlib
!pip install seaborn

# Import libraries
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 2: Load the Dataset

In [2]:

```python
# Load the dataset from the given URL
file_url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/n01PQ9pSmiRX6520fluj
df = pd.read_csv(file_url)

# Display the first few rows to understand the structure of the dataset
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

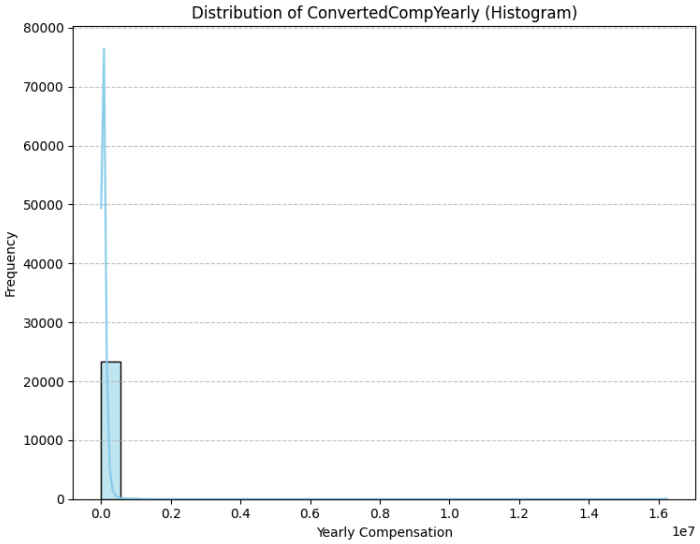
5 rows × 114 columns

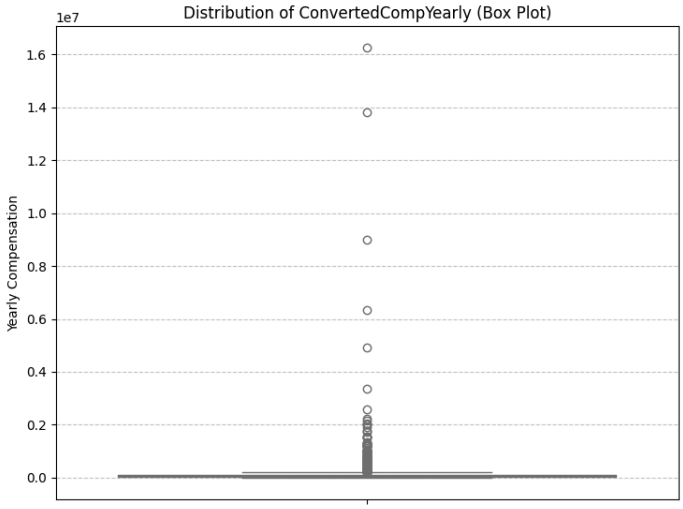
Step 3: Analyze and Visualize Compensation Distribution

Task: Plot the distribution and histogram for `ConvertedCompYearly` to examine the spread of yearly compensation among respondents.

In [3]:
```
## Write your code here
# --- Step 3: Analyze and Visualize Compensation Distribution ---
print("\n--- Step 3: Analyze and Visualize Compensation Distribution ---")
print("Task: Plot the distribution and histogram for ConvertedCompYearly to examine the spread of y

if 'ConvertedCompYearly' in df.columns:
    plt.figure(figsize=(15, 6))

    # Histogram
    plt.subplot(1, 2, 1) # 1 row, 2 columns, 1st plot
    sns.histplot(df['ConvertedCompYearly'], kde=True, bins=30, color='skyblue')
    plt.title('Distribution of ConvertedCompYearly (Histogram)')
    plt.xlabel('Yearly Compensation')
    plt.ylabel('Frequency')
    plt.grid(axis='y', linestyle='--', alpha=0.7)

    # Box Plot
    plt.subplot(1, 2, 2) # 1 row, 2 columns, 2nd plot
    sns.boxplot(y=df['ConvertedCompYearly'], color='lightcoral')
    plt.title('Distribution of ConvertedCompYearly (Box Plot)')
    plt.ylabel('Yearly Compensation')
    plt.grid(axis='y', linestyle='--', alpha=0.7)

    plt.tight_layout()
    plt.show()

    print("\nInterpretation of Compensation Distribution:")
    print("- The histogram shows the frequency of different compensation ranges, indicating its ove
    print("- The box plot visualizes the median, quartiles (spread of the middle 50%), and potentia
else:
    print("Column 'ConvertedCompYearly' not found or not properly prepared. Cannot analyze compensa
```

```
--- Step 3: Analyze and Visualize Compensation Distribution ---
Task: Plot the distribution and histogram for ConvertedCompYearly to examine the spread of yearly co
mpensation among respondents.
```






```
Interpretation of Compensation Distribution:
- The histogram shows the frequency of different compensation ranges, indicating its overall shape
(e.g., skewed).
- The box plot visualizes the median, quartiles (spread of the middle 50%), and potential outliers
(points beyond the whiskers).
```

Step 4: Calculate Median Compensation for Full-Time Employees

Task: Filter the data to calculate the median compensation for respondents whose employment status is "Employed, full-time."

In [4]:
```python
## Write your code here
# --- Step 4: Calculate Median Compensation for Full-Time Employees ---
print("\n--- Step 4: Calculate Median Compensation for Full-Time Employees ---")
print("Task: Filter the data to calculate the median compensation for respondents whose employment

if 'Employment' in df.columns and 'ConvertedCompYearly' in df.columns:
    # Filter for 'Employed, full-time'
    full_time_employees = df[df['Employment'] == 'Employed, full-time']

    if not full_time_employees.empty:
        median_comp_full_time = full_time_employees['ConvertedCompYearly'].median()
        print(f"\nMedian Yearly Compensation for 'Employed, full-time' respondents: {median_comp_fu
        print(f"Number of 'Employed, full-time' respondents: {len(full_time_employees)}")
    else:
        print("No 'Employed, full-time' respondents found in the dataset.")
else:
    print("Required columns ('Employment' or 'ConvertedCompYearly') not found or not properly prepa
```

```
--- Step 4: Calculate Median Compensation for Full-Time Employees ---
Task: Filter the data to calculate the median compensation for respondents whose employment status i
s 'Employed, full-time'.

Median Yearly Compensation for 'Employed, full-time' respondents: 69814.00
Number of 'Employed, full-time' respondents: 39041
```

Step 5: Analyzing Compensation Range and Distribution by Country

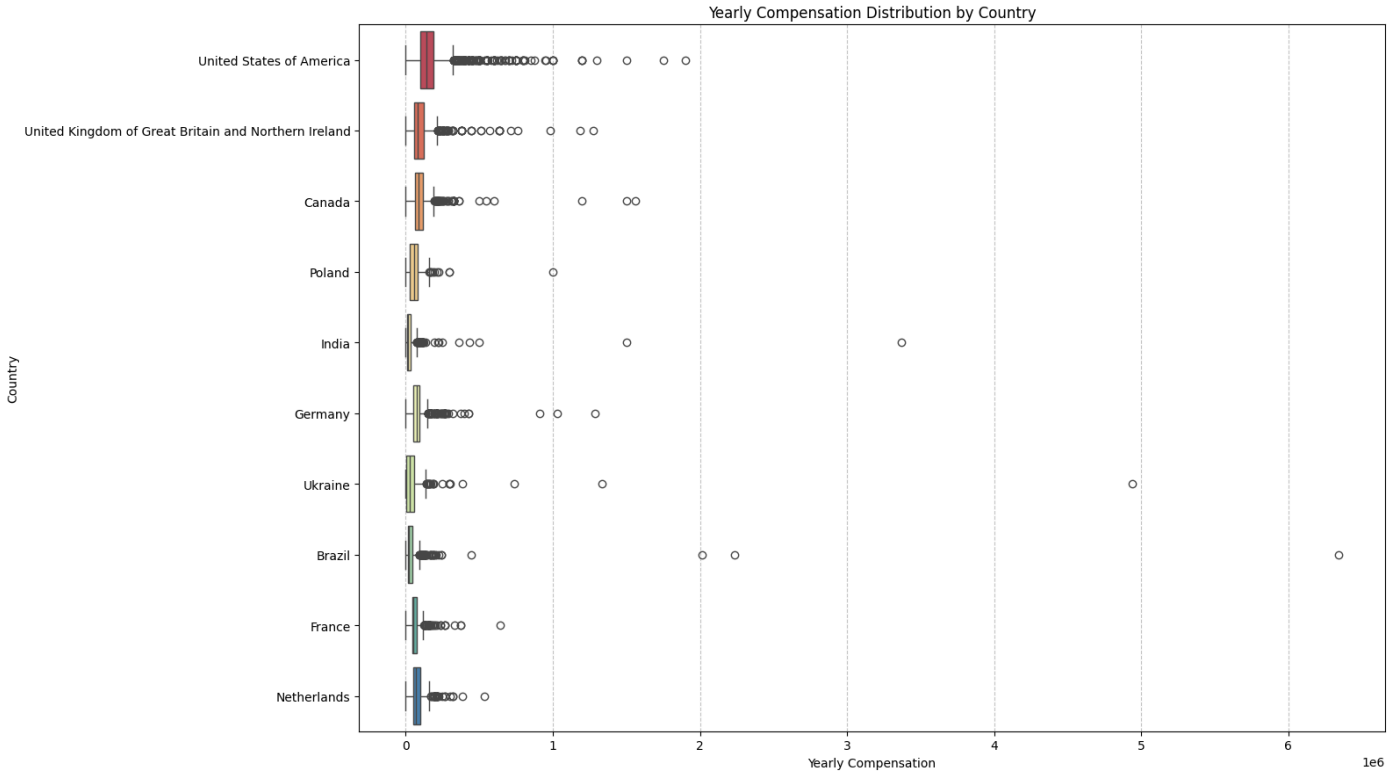
Explore the range of compensation in the ConvertedCompYearly column by analyzing differences across countries. Use box plots to compare the compensation distributions for each country to identify variations and anomalies within each region, providing insights into global compensation trends.

In [5]:
```python
## Write your code here
# --- Step 5: Analyzing Compensation Range and Distribution by Country ---
print("\n--- Step 5: Analyzing Compensation Range and Distribution by Country ---")
print("Explore the range of compensation in the ConvertedCompYearly column by analyzing differences

if 'Country' in df.columns and 'ConvertedCompYearly' in df.columns:
    # Get the top N countries for visualization to avoid clutter
    top_countries = df['Country'].value_counts().head(10).index.tolist()
    print(f"\nAnalyzing compensation distribution for top {len(top_countries)} countries: {top_coun

    # Filter the DataFrame to include only these top countries
```

```
    df_top_countries = df[df['Country'].isin(top_countries)]

    if not df_top_countries.empty:
        plt.figure(figsize=(16, 9))
        sns.boxplot(x='ConvertedCompYearly', y='Country', data=df_top_countries, palette='Spectral'
        plt.title('Yearly Compensation Distribution by Country')
        plt.xlabel('Yearly Compensation')
        plt.ylabel('Country')
        plt.grid(axis='x', linestyle='--', alpha=0.7)
        plt.tight_layout()
        plt.show()

        print("\nInterpretation of Compensation Distribution by Country:")
        print("- Box plots allow for visual comparison of median, spread, and outliers across diffe
        print("- You can observe if certain countries have generally higher or lower compensation r
    else:
        print("No data available for selected top countries to visualize compensation distribution.
else:
    print("Required columns ('Country' or 'ConvertedCompYearly') not found or not properly prepared
```

```
--- Step 5: Analyzing Compensation Range and Distribution by Country ---
Explore the range of compensation in the ConvertedCompYearly column by analyzing differences across
countries.

Analyzing compensation distribution for top 10 countries: ['United States of America', 'Germany', 'I
ndia', 'United Kingdom of Great Britain and Northern Ireland', 'Ukraine', 'France', 'Canada', 'Polan
d', 'Netherlands', 'Brazil']
```

```
/tmp/ipykernel_1100/750062032.py:16: 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.boxplot(x='ConvertedCompYearly', y='Country', data=df_top_countries, palette='Spectral')
```




```
Interpretation of Compensation Distribution by Country:
- Box plots allow for visual comparison of median, spread, and outliers across different countries.
- You can observe if certain countries have generally higher or lower compensation ranges, or more v
ariance.
```

Step 6: Removing Outliers from the Dataset

Task: Create a new DataFrame by removing outliers from the `ConvertedCompYearly` column to get a refined dataset for correlation analysis.

In [6]:
```
## Write your code here
# --- Step 6: Removing Outliers from the Dataset ---
print("\n--- Step 6: Removing Outliers from the Dataset ---")
print("Task: Create a new DataFrame by removing outliers from the ConvertedCompYearly column to get
```

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

    # Determine upper and lower bounds for outliers
    upper_bound = Q3 + (1.5 * IQR)
    lower_bound = Q1 - (1.5 * IQR)

    print(f"\nIQR Method Bounds for 'ConvertedCompYearly':")
    print(f"Q1 (25th percentile): {Q1:.2f}")
    print(f"Q3 (75th percentile): {Q3:.2f}")
    print(f"IQR: {IQR:.2f}")
    print(f"Lower Bound: {lower_bound:.2f}")
    print(f"Upper Bound: {upper_bound:.2f}")

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

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

    print("\nFirst 5 rows of the new DataFrame without outliers:")
    print(df_cleaned_outliers.head())

    # Update df to df_cleaned_outliers for subsequent steps as instructed by the PDF (implied)
    df = df_cleaned_outliers
    print("\nDataFrame 'df' has been updated to exclude outliers from 'ConvertedCompYearly'.")

else:
    print("'ConvertedCompYearly' column not found or is not numeric. Cannot perform outlier removal
```

```
--- Step 6: Removing Outliers from the Dataset ---
Task: Create a new DataFrame by removing outliers from the ConvertedCompYearly column to get a refin
ed dataset for correlation analysis.

IQR Method Bounds for 'ConvertedCompYearly':
Q1 (25th percentile): 32712.00
Q3 (75th percentile): 107971.50
IQR: 75259.50
Lower Bound: -80177.25
Upper Bound: 220860.75

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

First 5 rows of the new DataFrame without outliers:
     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
```

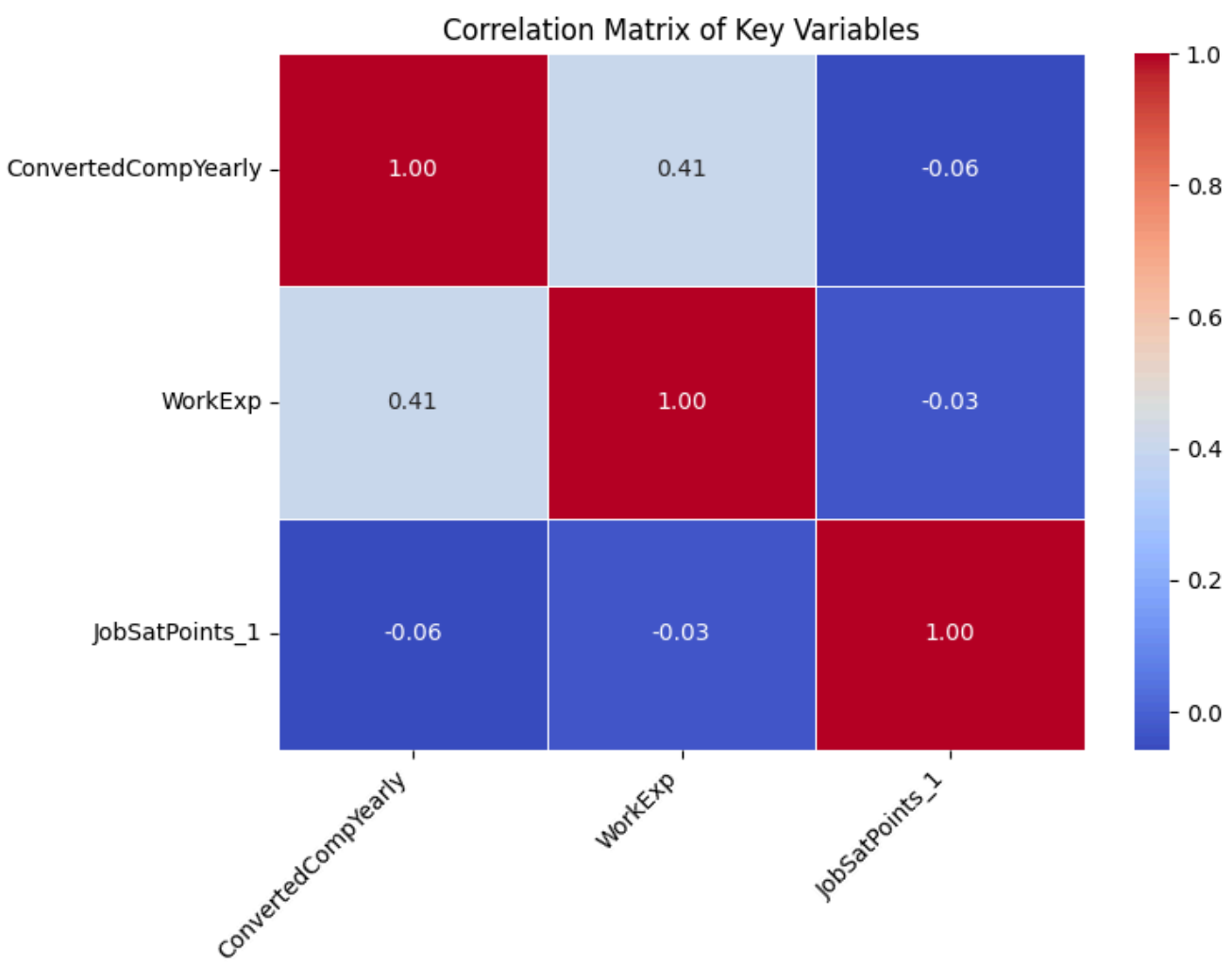
```
379             0.0              Too long                  Difficult
385             NaN             Too short                       Easy
389             5.0              Too long                       Easy

    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]

DataFrame 'df' has been updated to exclude outliers from 'ConvertedCompYearly'.
```

Step 7: Finding Correlations Between Key Variables

Task: Calculate correlations between `ConvertedCompYearly`, `WorkExp`, and `JobSatPoints_1`. Visualize these correlations with a heatmap.

In [7]:
```python
## Write your code here
# --- Step 7: Finding Correlations Between Key Variables ---
print("\n--- Step 7: Finding Correlations Between Key Variables ---")
print("Task: Calculate correlations between ConvertedCompYearly, WorkExp, and JobSatPoints_1. Visua

# Ensure df contains the necessary columns after outlier removal
required_corr_cols = ['ConvertedCompYearly', 'WorkExp', 'JobSatPoints_1']
available_corr_cols = [col for col in required_corr_cols if col in df.columns]

if len(available_corr_cols) >= 2: # Need at least two columns for correlation
    # Calculate the correlation matrix
    correlation_matrix = df[available_corr_cols].corr()

    print("\nCorrelation Matrix:")
    print(correlation_matrix)

    # Visualize the correlation matrix with a heatmap
    plt.figure(figsize=(8, 6))
    sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
    plt.title('Correlation Matrix of Key Variables')
    plt.xticks(rotation=45, ha='right')
    plt.yticks(rotation=0)
    plt.tight_layout()
    plt.show()

    print("\nInterpretation of Correlation Matrix:")
    print("- Values close to 1 indicate a strong positive linear relationship.")
    print("- Values close to -1 indicate a strong negative linear relationship.")
    print("- Values close to 0 indicate little to no linear relationship.")
else:
    print(f"Insufficient or missing columns for correlation analysis. Required: {required_corr_cols
```

```
--- Step 7: Finding Correlations Between Key Variables ---
Task: Calculate correlations between ConvertedCompYearly, WorkExp, and JobSatPoints_1. Visualize the
se correlations with a heatmap.

Correlation Matrix:
                    ConvertedCompYearly   WorkExp  JobSatPoints_1
ConvertedCompYearly             1.000000  0.408438       -0.058170
WorkExp                         0.408438  1.000000       -0.032388
JobSatPoints_1                 -0.058170 -0.032388        1.000000
```




```
Interpretation of Correlation Matrix:
- Values close to 1 indicate a strong positive linear relationship.
- Values close to -1 indicate a strong negative linear relationship.
- Values close to 0 indicate little to no linear relationship.
```

Step 8: Scatter Plot for Correlations

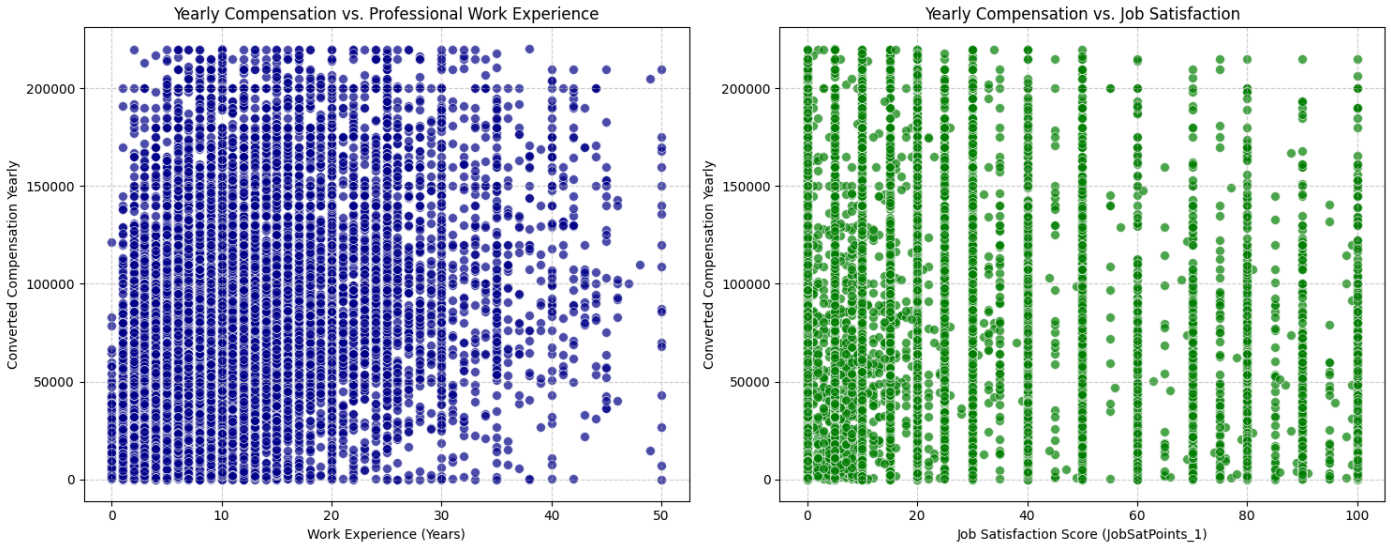
Task: Create scatter plots to examine specific correlations between `ConvertedCompYearly` and `WorkExp`, as well as between `ConvertedCompYearly` and `JobSatPoints_1`.

In [8]:

```python
## Write your code here
# --- Step 8: Scatter Plot for Correlations ---
print("\n--- Step 8: Scatter Plot for Correlations ---")
print("Task: Create scatter plots to examine specific correlations between ConvertedCompYearly and

# Ensure df contains the necessary columns after outlier removal
if 'ConvertedCompYearly' in df.columns and 'WorkExp' in df.columns and 'JobSatPoints_1' in df.colum
    plt.figure(figsize=(15, 6))

    # Scatter plot: ConvertedCompYearly vs. WorkExp
    plt.subplot(1, 2, 1) # 1 row, 2 columns, 1st plot
    sns.scatterplot(x='WorkExp', y='ConvertedCompYearly', data=df, alpha=0.7, s=50, color='darkblue
    plt.title('Yearly Compensation vs. Professional Work Experience')
    plt.xlabel('Work Experience (Years)')
    plt.ylabel('Converted Compensation Yearly')
    plt.grid(True, linestyle='--', alpha=0.6)

    # Scatter plot: ConvertedCompYearly vs. JobSatPoints_1
    plt.subplot(1, 2, 2) # 1 row, 2 columns, 2nd plot
    sns.scatterplot(x='JobSatPoints_1', y='ConvertedCompYearly', data=df, alpha=0.7, s=50, color='g
    plt.title('Yearly Compensation vs. Job Satisfaction')
    plt.xlabel('Job Satisfaction Score (JobSatPoints_1)')
    plt.ylabel('Converted Compensation Yearly')
    plt.grid(True, linestyle='--', alpha=0.6)

    plt.tight_layout()
    plt.show()
```

```
    print("\nInterpretation of Scatter Plots:")
    print("- Each scatter plot visually represents the relationship between two variables.")
    print("- You can observe patterns like positive/negative trends, clusters, or lack of clear rel
    print("- High density of points indicates common combinations of values.")

else:
    print("Required columns for scatter plots ('ConvertedCompYearly', 'WorkExp', or 'JobSatPoints_1
```

```
--- Step 8: Scatter Plot for Correlations ---
Task: Create scatter plots to examine specific correlations between ConvertedCompYearly and WorkExp,
as well as between ConvertedCompYearly and JobSatPoints_1.
```




```
Interpretation of Scatter Plots:
- Each scatter plot visually represents the relationship between two variables.
- You can observe patterns like positive/negative trends, clusters, or lack of clear relationship.
- High density of points indicates common combinations of values.
```

Summary

In this lab, you practiced essential skills in correlation analysis by:

- Examining the distribution of yearly compensation with histograms and box plots.
- Detecting and removing outliers from compensation data.
- Calculating correlations between key variables such as compensation, work experience, and job satisfaction.
- Visualizing relationships with scatter plots and heatmaps to gain insights into the associations between these features.

By following these steps, you have developed a solid foundation for analyzing relationships within the dataset.

Authors:

Ayushi Jain

Other Contributors:

- Rav Ahuja
- Lakshmi Holla
- Malika