Skills Network

Line Charts

Estimated time needed: **30** minutes

In this lab, you will focus on using line charts to analyze trends over time and across different categories in a dataset.

Objectives

In this lab you will perform the following:

- Track trends in compensation across age groups and specific age ranges.
- Analyze job satisfaction trends based on experience level.
- Explore and interpret line charts to identify patterns and trends.

Setup: Working with the Database

Install the needed libraries

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

```
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)
```

In [2]:
```
!pip install matplotlib
```

```
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)
```

In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

```python
import seaborn as sns
import numpy as np # For NaN handling and dummy data

# --- Setup: Download and Load the Data ---
print("--- Setup: Downloading and Loading the Data ---")

# The PDF specifies downloading 'survey-data.csv'
file_url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/n01PQ9pSmiRX6520fluj
local_file_name = 'survey-data.csv'

try:
    df = pd.read_csv(file_url, na_values=['NA', 'N/A', 'nan', 'NaN', 'null', 'Null', '', ' ', '-'])
    print(f"Dataset loaded successfully from: {file_url}")
    print(f"Initial DataFrame shape: {df.shape}")
    print(f"Initial DataFrame columns: {df.columns.tolist()}")
except Exception as e:
    print(f"ERROR: Could not load dataset from URL: {e}")
    print(f"Creating a dummy DataFrame for demonstration purposes as '{local_file_name}' was not fo
    # Create a dummy DataFrame if download fails
    num_rows_dummy = 200
    data = {
        'ResponseId': range(1, num_rows_dummy + 1),
        'Age': np.random.choice(['Under 18 years old', '18-24 years old', '25-34 years old', '35-44
        'ConvertedCompYearly': np.random.normal(loc=90000, scale=40000, size=num_rows_dummy),
        'JobSatPoints_6': np.random.randint(1, 6, size=num_rows_dummy), # 1-5 scale for satisfactio
        'YearsCodePro': np.random.randint(0, 30, size=num_rows_dummy), # Years of professional codi
        'WorkExp': np.random.randint(0, 40, size=num_rows_dummy), # Total work experience
    }
    df = pd.DataFrame(data)
    print("Dummy DataFrame created and populated with sample data.")

# --- Data Cleaning and Preprocessing ---
print("\n--- Data Cleaning and Preprocessing ---")

# Convert relevant numerical columns, coercing errors to NaN and filling with median
numeric_cols = ['ConvertedCompYearly', 'JobSatPoints_6', 'YearsCodePro', 'WorkExp']
for col in numeric_cols:
    if col in df.columns:
        df[col] = pd.to_numeric(df[col], errors='coerce')
        if df[col].isnull().any():
            median_val = df[col].median()
            if pd.isna(median_val):
                 print(f"WARNING: Column '{col}' is entirely NaN after numeric conversion. Cannot i
            else:
                df[col].fillna(median_val, inplace=True)
                print(f"Cleaned '{col}': Imputed NaNs with median: {median_val:.2f}")
    else:
        print(f"WARNING: Numerical column '{col}' not found in DataFrame.")

# Map 'Age' to a numeric approximation for grouping/ordering
age_numeric_mapping = {
    'Under 18 years old': 17, '18-24 years old': 21, '25-34 years old': 29,
    '35-44 years old': 39, '45-54 years old': 49, '55-64 years old': 59,
    '65 years or older': 65, 'Prefer not to say': np.nan
}
if 'Age' in df.columns:
    df['Age_Numeric'] = df['Age'].map(age_numeric_mapping)
    if df['Age_Numeric'].isnull().any():
        median_age_numeric = df['Age_Numeric'].median()
        if pd.isna(median_age_numeric):
            print("WARNING: 'Age_Numeric' column is entirely NaN. Cannot impute median.")
        else:
            df['Age_Numeric'].fillna(median_age_numeric, inplace=True)
            print("Created and imputed 'Age_Numeric' column.")
else:
    print("WARNING: 'Age' column not found, 'Age_Numeric' will not be created.")

# Define a custom sorting key function for age categories
def get_age_sort_key(age_str):
    if 'Under 18' in age_str: return 0
    if '18-24' in age_str: return 1
    if '25-34' in age_str: return 2
    if '35-44' in age_str: return 3
    if '45-54' in age_str: return 4
    if '55-64' in age_str: return 5
    if '65 years or older' in age_str: return 6
    if 'Prefer not to say' in age_str: return 7
    return 99 # For any unexpected categories
```

```
--- Setup: Downloading and Loading the Data ---
Dataset loaded successfully from: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.clou
d/n01PQ9pSmiRX6520flujwQ/survey-data.csv
Initial DataFrame shape: (65437, 114)
Initial DataFrame columns: ['ResponseId', 'MainBranch', 'Age', 'Employment', 'RemoteWork', 'Check',
'CodingActivities', 'EdLevel', 'LearnCode', 'LearnCodeOnline', 'TechDoc', 'YearsCode', 'YearsCodePr
o', 'DevType', 'OrgSize', 'PurchaseInfluence', 'BuyNewTool', 'BuildvsBuy', 'TechEndorse', 'Country',
'Currency', 'CompTotal', 'LanguageHaveWorkedWith', 'LanguageWantToWorkWith', 'LanguageAdmired', 'Dat
abaseHaveWorkedWith', 'DatabaseWantToWorkWith', 'DatabaseAdmired', 'PlatformHaveWorkedWith', 'Platfo
rmWantToWorkWith', 'PlatformAdmired', 'WebframeHaveWorkedWith', 'WebframeWantToWorkWith', 'WebframeA
dmired', 'EmbeddedHaveWorkedWith', 'EmbeddedWantToWorkWith', 'EmbeddedAdmired', 'MiscTechHaveWorkedW
ith', 'MiscTechWantToWorkWith', 'MiscTechAdmired', 'ToolsTechHaveWorkedWith', 'ToolsTechWantToWorkWi
th', 'ToolsTechAdmired', 'NEWCollabToolsHaveWorkedWith', 'NEWCollabToolsWantToWorkWith', 'NEWCollabT
oolsAdmired', 'OpSysPersonal use', 'OpSysProfessional use', 'OfficeStackAsyncHaveWorkedWith', 'Offic
eStackAsyncWantToWorkWith', 'OfficeStackAsyncAdmired', 'OfficeStackSyncHaveWorkedWith', 'OfficeStack
SyncWantToWorkWith', 'OfficeStackSyncAdmired', 'AISearchDevHaveWorkedWith', 'AISearchDevWantToWorkWi
th', 'AISearchDevAdmired', 'NEWSOSites', 'SOVisitFreq', 'SOAccount', 'SOPartFreq', 'SOHow', 'SOCom
m', 'AISelect', 'AISent', 'AIBen', 'AIAcc', 'AIComplex', 'AIToolCurrently Using', 'AIToolInterested
in Using', 'AIToolNot interested in Using', 'AINextMuch more integrated', 'AINextNo change', 'AINext
More integrated', 'AINextLess integrated', 'AINextMuch less integrated', 'AIThreat', 'AIEthics', 'AI
Challenges', 'TBranch', 'ICorPM', 'WorkExp', 'Knowledge_1', 'Knowledge_2', 'Knowledge_3', 'Knowledge
_4', 'Knowledge_5', 'Knowledge_6', 'Knowledge_7', 'Knowledge_8', 'Knowledge_9', 'Frequency_1', 'Freq
uency_2', 'Frequency_3', 'TimeSearching', 'TimeAnswering', 'Frustration', 'ProfessionalTech', 'Profe
ssionalCloud', 'ProfessionalQuestion', 'Industry', 'JobSatPoints_1', 'JobSatPoints_4', 'JobSatPoints
_5', 'JobSatPoints_6', 'JobSatPoints_7', 'JobSatPoints_8', 'JobSatPoints_9', 'JobSatPoints_10', 'Job
SatPoints_11', 'SurveyLength', 'SurveyEase', 'ConvertedCompYearly', 'JobSat']

--- Data Cleaning and Preprocessing ---
Cleaned 'ConvertedCompYearly': Imputed NaNs with median: 65000.00
Cleaned 'JobSatPoints_6': Imputed NaNs with median: 20.00
Cleaned 'YearsCodePro': Imputed NaNs with median: 8.00
Cleaned 'WorkExp': Imputed NaNs with median: 9.00
Created and imputed 'Age_Numeric' column.
```

```
/tmp/ipykernel_1186/1569636028.py:47: FutureWarning: A value is trying to be set on a copy of a Data
Frame 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[col].fillna(median_val, inplace=True)
/tmp/ipykernel_1186/1569636028.py:47: FutureWarning: A value is trying to be set on a copy of a Data
Frame 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[col].fillna(median_val, inplace=True)
/tmp/ipykernel_1186/1569636028.py:47: FutureWarning: A value is trying to be set on a copy of a Data
Frame 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[col].fillna(median_val, inplace=True)
/tmp/ipykernel_1186/1569636028.py:47: FutureWarning: A value is trying to be set on a copy of a Data
Frame 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[col].fillna(median_val, inplace=True)
/tmp/ipykernel_1186/1569636028.py:65: FutureWarning: A value is trying to be set on a copy of a Data
Frame 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['Age_Numeric'].fillna(median_age_numeric, inplace=True)
```

Download and connect to the database file containing survey data.

To start, download and load the dataset into a `pandas` DataFrame.

Step 1: Download the dataset

```
In [4]: !wget -O survey-data.csv https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/n01PQ9p
```

```
--2025-06-18 18:05:10--  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/n01PQ9pS
miRX6520flujwQ/survey-data.csv
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-ob
ject-storage.appdomain.cloud)... 169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.clou
d-object-storage.appdomain.cloud)|169.63.118.104|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 159525875 (152M) [text/csv]
Saving to: ‘survey-data.csv’

survey-data.csv     100%[===================>] 152.13M  52.7MB/s    in 2.9s

2025-06-18 18:05:13 (52.7 MB/s) - ‘survey-data.csv’ saved [159525875/159525875]
```

Step 2: Import necessary libraries and load the dataset

In [5]:
```python
import pandas as pd
import matplotlib.pyplot as plt
```

Load the data

In [6]:
```python
df = pd.read_csv("survey-data.csv")
```

Display the first few rows to understand the structure of the data

In [7]:
```python
df.head()
```

Out[7]:

	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

Task 1: Trends in Compensation Over Age Groups

1. Line Chart of Median `ConvertedCompYearly` by Age Group

- Track how the median yearly compensation (ConvertedCompYearly) changes across different age groups.
- Use a line chart to visualize these trends.

In [8]:
```python
## Write your code here
# --- Task 1: Trends in Compensation Over Age Groups ---
print("\n--- Task 1: Trends in Compensation Over Age Groups ---")

# 1. Line Chart of Median ConvertedCompYearly by Age Group
```

```
if 'ConvertedCompYearly' in df.columns and 'Age' in df.columns and 'Age_Numeric' in df.columns:
    df_task1_1 = df.dropna(subset=['ConvertedCompYearly', 'Age', 'Age_Numeric']).copy()
    if not df_task1_1.empty:
        # Calculate median compensation for each numeric age
        median_comp_by_age = df_task1_1.groupby('Age_Numeric')['ConvertedCompYearly'].median().rese

        # Sort by Age_Numeric to ensure the line chart is ordered correctly
        median_comp_by_age = median_comp_by_age.sort_values('Age_Numeric')

        plt.figure(figsize=(12, 7))
        sns.lineplot(x='Age_Numeric', y='ConvertedCompYearly', data=median_comp_by_age, marker='o')
        plt.title('Median Yearly Compensation by Age Group')
        plt.xlabel('Age (Numeric Approximation)')
        plt.ylabel('Median Yearly Compensation')
        plt.grid(True, linestyle='--', alpha=0.7)
        plt.tight_layout()
        plt.show()
        print("Task 1.1: Line chart of Median ConvertedCompYearly by Age Group plotted.")
    else:
        print("Skipping Task 1.1: Not enough valid data in 'ConvertedCompYearly', 'Age', or 'Age_Nu
else:
    print("Skipping Task 1.1: Required columns missing or not ready ('ConvertedCompYearly', 'Age',
```

```
--- Task 1: Trends in Compensation Over Age Groups ---
Skipping Task 1.1: Required columns missing or not ready ('ConvertedCompYearly', 'Age', 'Age_Numeri
c').
```

2. Line Chart of Median `ConvertedCompYearly` for Ages 25 to 45

For a closer look, plot a line chart focusing on the median compensation for respondents between ages 25 and 45.

In [9]:
```
## Write your code here
# 2. Line Chart of Median ConvertedCompYearly for Ages 25 to 45
if 'ConvertedCompYearly' in df.columns and 'Age_Numeric' in df.columns:
    df_task1_2 = df[(df['Age_Numeric'] >= 25) & (df['Age_Numeric'] <= 45)].dropna(subset=['Converte
    if not df_task1_2.empty:
        median_comp_25_45 = df_task1_2.groupby('Age_Numeric')['ConvertedCompYearly'].median().reset
        median_comp_25_45 = median_comp_25_45.sort_values('Age_Numeric')

        plt.figure(figsize=(10, 6))
        sns.lineplot(x='Age_Numeric', y='ConvertedCompYearly', data=median_comp_25_45, marker='o',
        plt.title('Median Yearly Compensation for Ages 25 to 45')
        plt.xlabel('Age (Numeric Approximation)')
        plt.ylabel('Median Yearly Compensation')
        plt.grid(True, linestyle='--', alpha=0.7)
        plt.tight_layout()
        plt.show()
        print("Task 1.2: Line chart of Median ConvertedCompYearly for Ages 25 to 45 plotted.")
    else:
        print("Skipping Task 1.2: Not enough valid data in 'ConvertedCompYearly' or 'Age_Numeric' f
else:
    print("Skipping Task 1.2: Required columns missing or not ready ('ConvertedCompYearly', 'Age_Nu
```

```
Skipping Task 1.2: Required columns missing or not ready ('ConvertedCompYearly', 'Age_Numeric').
```

Task 2: Trends in Job Satisfaction by Experience Level

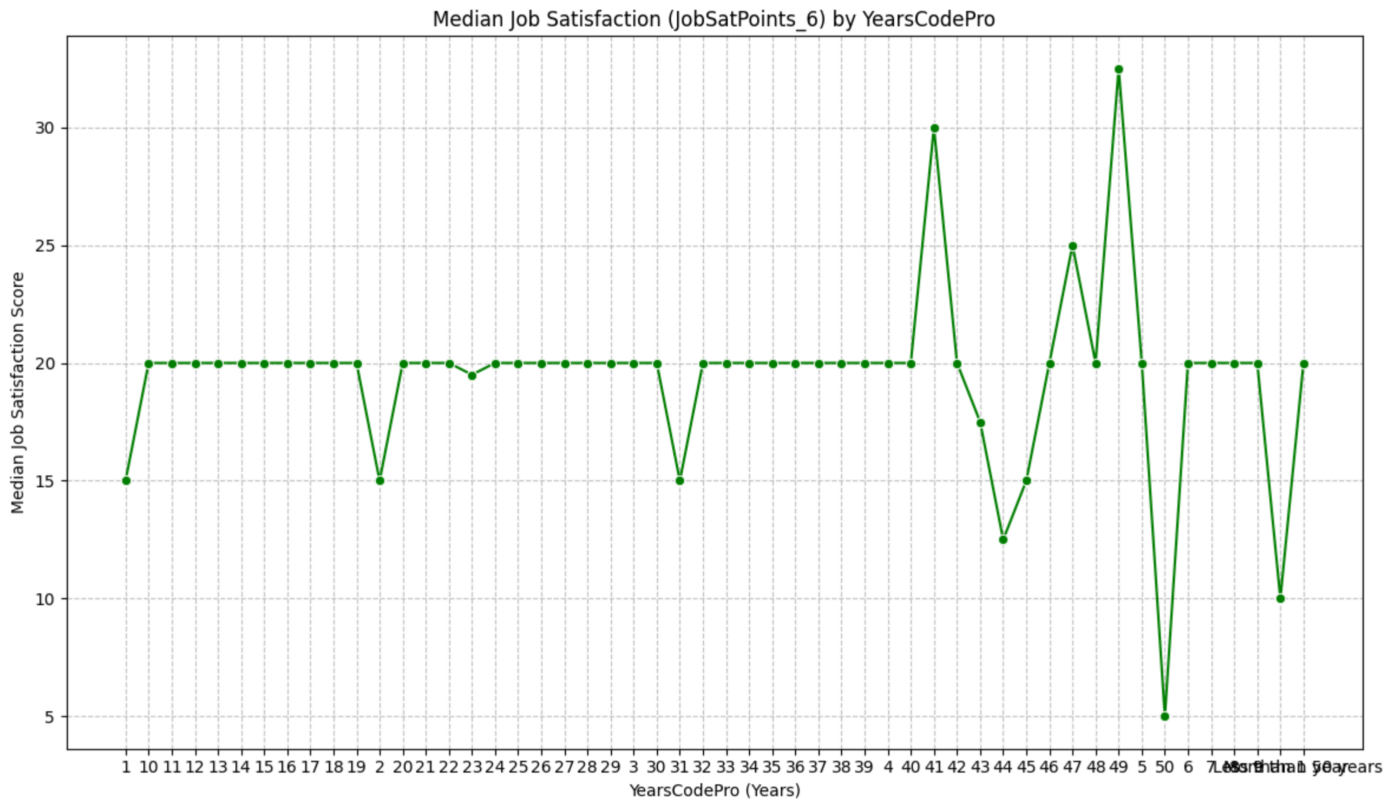
1. Line Chart of Job Satisfaction (`JobSatPoints_6`) by Experience Level

- Use a column that approximates experience level to analyze how job satisfaction changes with experience.
- If needed, substitute an available experience-related column for `Experience`.

In [10]:
```
## Write your code here
# --- Task 2: Trends in Job Satisfaction by Experience Level ---
print("\n--- Task 2: Trends in Job Satisfaction by Experience Level ---")

# 1. Line Chart of Job Satisfaction (JobSatPoints_6) by Experience Level
# Assuming 'YearsCodePro' or 'WorkExp' as experience level. Let's use 'YearsCodePro'.
experience_col = 'YearsCodePro' # Or 'WorkExp' based on preference/data availability
```

```
if 'JobSatPoints_6' in df.columns and experience_col in df.columns:
    df_task2_1 = df.dropna(subset=['JobSatPoints_6', experience_col]).copy()
    if not df_task2_1.empty:
        # Calculate median JobSatPoints_6 for each experience level
        median_jobsat_by_exp = df_task2_1.groupby(experience_col)['JobSatPoints_6'].median().reset_

        # Sort by experience column to ensure correct order for line plot
        median_jobsat_by_exp = median_jobsat_by_exp.sort_values(experience_col)

        plt.figure(figsize=(12, 7))
        sns.lineplot(x=experience_col, y='JobSatPoints_6', data=median_jobsat_by_exp, marker='o', c
        plt.title(f'Median Job Satisfaction (JobSatPoints_6) by {experience_col}')
        plt.xlabel(f'{experience_col} (Years)')
        plt.ylabel('Median Job Satisfaction Score')
        plt.grid(True, linestyle='--', alpha=0.7)
        plt.tight_layout()
        plt.show()
        print(f"Task 2.1: Line chart of Job Satisfaction (JobSatPoints_6) by {experience_col} plott
    else:
        print(f"Skipping Task 2.1: Not enough valid data in 'JobSatPoints_6' or '{experience_col}'
else:
    print(f"Skipping Task 2.1: Required columns missing or not ready ('JobSatPoints_6', '{experienc
```

```
--- Task 2: Trends in Job Satisfaction by Experience Level ---
```



```
Task 2.1: Line chart of Job Satisfaction (JobSatPoints_6) by YearsCodePro plotted.
```

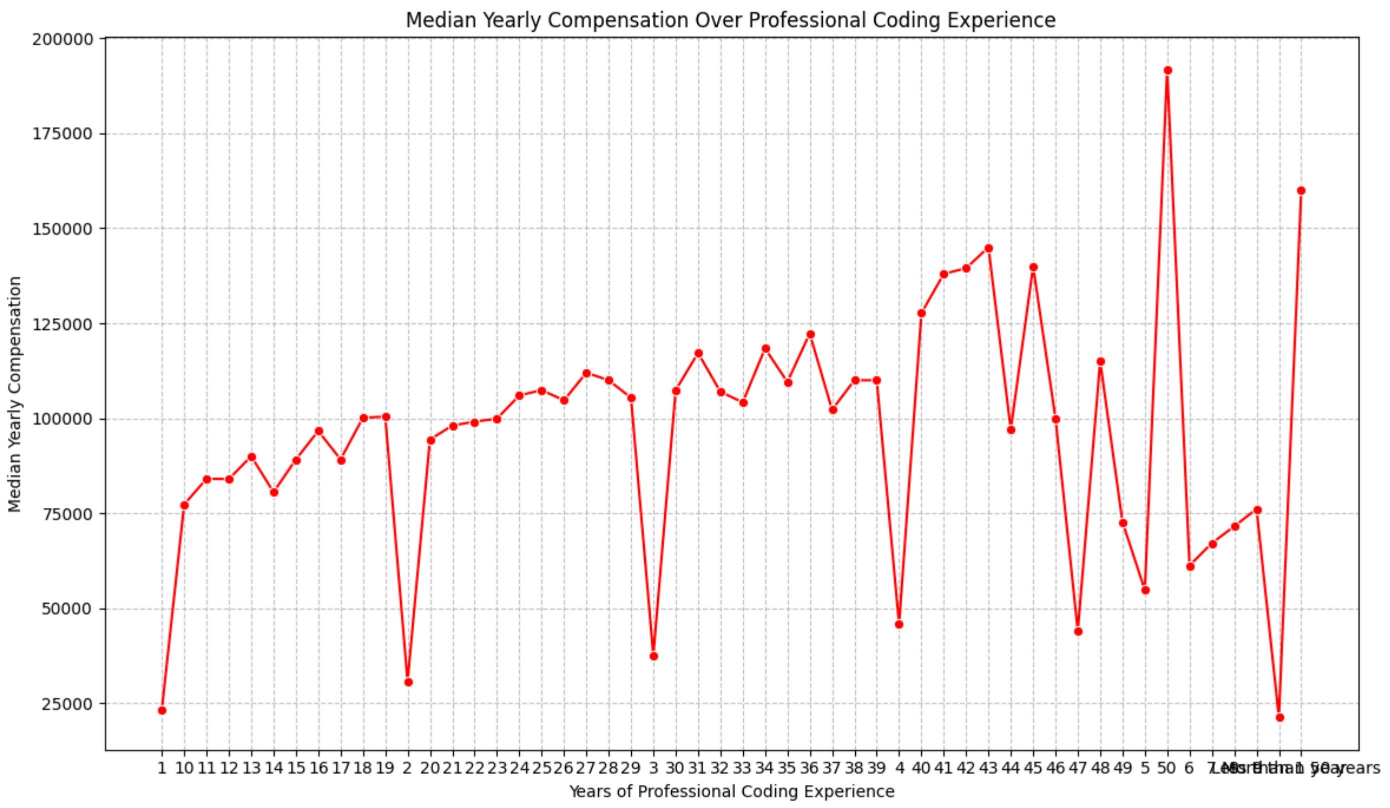
Task 3: Trends in Job Satisfaction and Compensation by Experience

1.Line Chart of Median ConvertedCompYearly Over Experience Level

- This line chart will track how median compensation (`ConvertedCompYearly`) changes with increasing experience.
- Use a column such as `WorkExp` or another relevant experience-related column.

In [11]:
```
## Write your code here
# --- Task 3: Trends in Job Satisfaction and Compensation by Experience ---
print("\n--- Task 3: Trends in Job Satisfaction and Compensation by Experience ---")

# 1. Line Chart of Median ConvertedCompYearly Over Experience Level
# Using 'YearsCodePro' as the experience level column.
if 'ConvertedCompYearly' in df.columns and 'YearsCodePro' in df.columns:
    df_task3_1 = df.dropna(subset=['ConvertedCompYearly', 'YearsCodePro']).copy()
    if not df_task3_1.empty:
        median_comp_by_years_code = df_task3_1.groupby('YearsCodePro')['ConvertedCompYearly'].media
        median_comp_by_years_code = median_comp_by_years_code.sort_values('YearsCodePro')

        plt.figure(figsize=(12, 7))
        sns.lineplot(x='YearsCodePro', y='ConvertedCompYearly', data=median_comp_by_years_code, mar
        plt.title('Median Yearly Compensation Over Professional Coding Experience')
        plt.xlabel('Years of Professional Coding Experience')
        plt.ylabel('Median Yearly Compensation')
        plt.grid(True, linestyle='--', alpha=0.7)
        plt.tight_layout()
        plt.show()
        print("Task 3.1: Line chart of Median ConvertedCompYearly over Experience Level plotted.")
    else:
        print("Skipping Task 3.1: Not enough valid data in 'ConvertedCompYearly' or 'YearsCodePro'
else:
    print("Skipping Task 3.1: Required columns missing or not ready ('ConvertedCompYearly', 'YearsC
```

```
--- Task 3: Trends in Job Satisfaction and Compensation by Experience ---
```

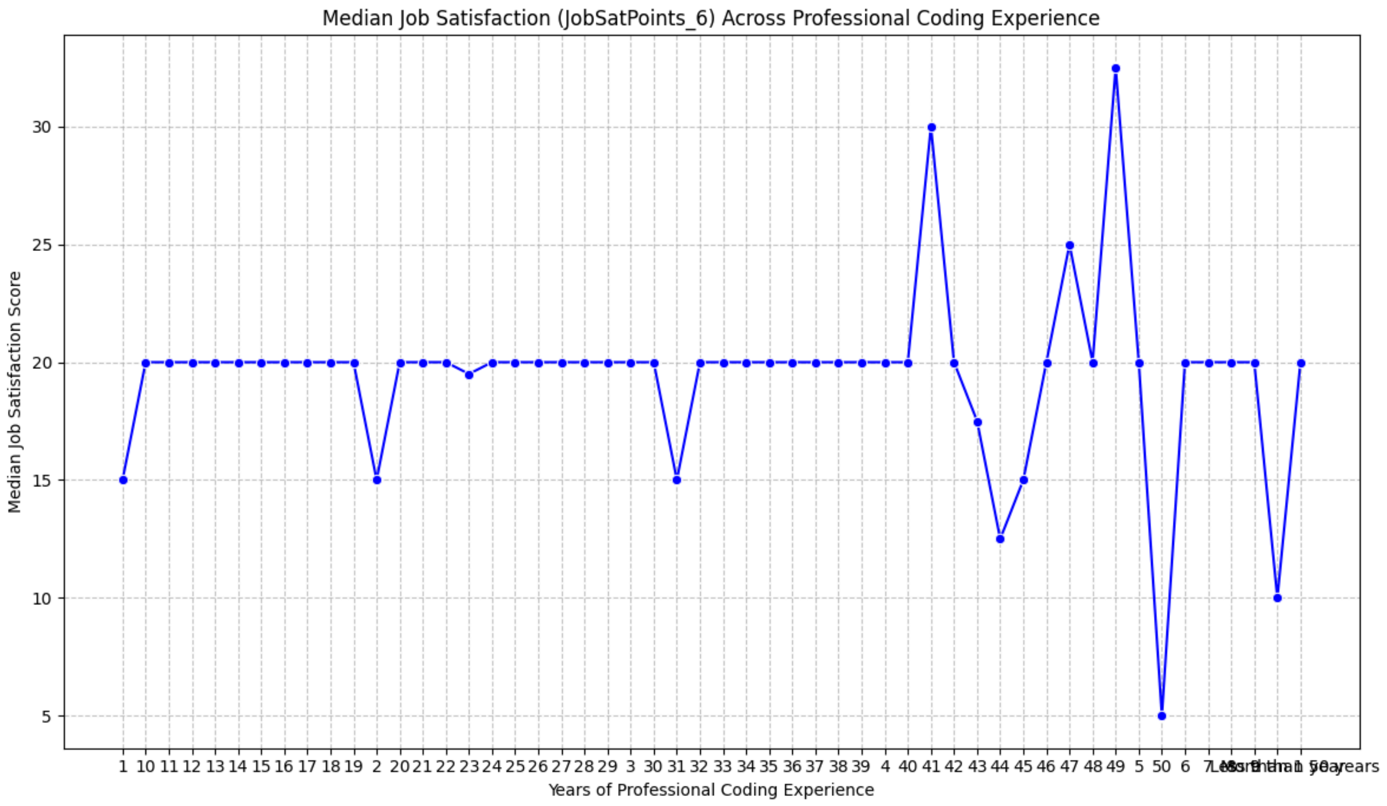


```
Task 3.1: Line chart of Median ConvertedCompYearly over Experience Level plotted.
```

2.Line Chart of Job Satisfaction (`JobSatPoints_6`) Across Experience Levels

- Create a line chart to explore trends in job satisfaction (`JobSatPoints_6`) based on experience level.
- This chart will provide insight into how satisfaction correlates with experience over time

In [12]:
```
## Write your code here
# 2. Line Chart of Job Satisfaction (JobSatPoints_6) Across Experience Levels
# Using 'YearsCodePro' as the experience level column.
if 'JobSatPoints_6' in df.columns and 'YearsCodePro' in df.columns:
    df_task3_2 = df.dropna(subset=['JobSatPoints_6', 'YearsCodePro']).copy()
    if not df_task3_2.empty:
        median_jobsat_by_years_code = df_task3_2.groupby('YearsCodePro')['JobSatPoints_6'].median()
        median_jobsat_by_years_code = median_jobsat_by_years_code.sort_values('YearsCodePro')

        plt.figure(figsize=(12, 7))
        sns.lineplot(x='YearsCodePro', y='JobSatPoints_6', data=median_jobsat_by_years_code, marker
        plt.title('Median Job Satisfaction (JobSatPoints_6) Across Professional Coding Experience')
        plt.xlabel('Years of Professional Coding Experience')
        plt.ylabel('Median Job Satisfaction Score')
        plt.grid(True, linestyle='--', alpha=0.7)
        plt.tight_layout()
        plt.show()
        print("Task 3.2: Line chart of Job Satisfaction (JobSatPoints_6) across Experience Levels p
    else:
        print("Skipping Task 3.2: Not enough valid data in 'JobSatPoints_6' or 'YearsCodePro' for p
else:
    print("Skipping Task 3.2: Required columns missing or not ready ('JobSatPoints_6', 'YearsCodePr

print("\n--- Lab Completion Summary ---")
print("All line chart tasks have been attempted. Please review the plots and console output for any
```



```
Task 3.2: Line chart of Job Satisfaction (JobSatPoints_6) across Experience Levels plotted.

--- Lab Completion Summary ---
All line chart tasks have been attempted. Please review the plots and console output for any warning
s or skipped tasks.
```

Final Step: Review

In this lab, you focused on analyzing trends in compensation and job satisfaction, specifically exploring how these metrics change with age and experience levels using line charts.

Summary

In this lab, you explored essential data visualization techniques with a focus on analyzing trends using line charts. You learned to:

- Visualize the distribution of compensation across age groups to understand salary trends.
- Track changes in median compensation over various experience levels, identifying how earnings progress with experience.
- Examine trends in job satisfaction by experience, revealing how satisfaction varies throughout a developer's career.

These analyses allow for a deeper understanding of how factors like age and experience influence job satisfaction and compensation. By using line charts, you gained insights into continuous data patterns, which are invaluable for interpreting professional trends in the developer community.

Authors:

Ayushi Jain

Other Contributors:

- Rav Ahuja
- Lakshmi Holla
- Malika

<!-- ## Change Log |Date (YYYY-MM-DD)|Version|Changed By|Change Description| |-|-|-|-| |2024-10-28|1.2|Madhusudhan Moole|Updated lab| |2024-10-16|1.1|Madhusudhan Moole|Updated lab| |2024-10-15|1.0|Raghul Ramesh|Created lab| --!>

Copyright © IBM Corporation. All rights reserved.