Lab 15 Box Plot

April 7, 2025

1 Box Plots

Estimated time needed: **45** minutes

In this lab, you will focus on the visualization of data. The dataset will be provided through an RDBMS, and you will need to use SQL queries to extract the required data.

1.1 Objectives

In this lab you will perform the following:

- Visualize the distribution of data.
- Visualize the relationship between two features.
- Visualize data composition and comparisons using box plots.

1.1.1 Setup: Connecting to the Database

1. Download the Database File

[1]:
```
!wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
↪QR9YeprUYhOoLafzlLspAw/survey-results-public.sqlite
```

```
--2025-04-07 11:30:28--  https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/QR9YeprUYhOoLafzlLspAw/survey-results-public.sqlite
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-
courses-data.s3.us.cloud-object-storage.appdomain.cloud)… 169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-
courses-data.s3.us.cloud-object-storage.appdomain.cloud)|169.63.118.104|:443…
connected.
200 OKequest sent, awaiting response…
Length: 211415040 (202M) [application/octet-stream]
Saving to: ‘survey-results-public.sqlite.2’

survey-results-publ 100%[===================>] 201.62M  69.1MB/s    in 2.9s

2025-04-07 11:30:34 (69.1 MB/s) - ‘survey-results-public.sqlite.2’ saved
[211415040/211415040]
```

2. Connect to the Database Install the needed libraries

[2]:
```
!pip install pandas
```

```
Requirement already satisfied: pandas in /opt/conda/lib/python3.12/site-packages
(2.2.3)
Requirement already satisfied: numpy>=1.26.0 in /opt/conda/lib/python3.12/site-
packages (from pandas) (2.2.4)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/conda/lib/python3.12/site-packages (from 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 pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-
packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

[3]:
```
!pip install matplotlib
!pip install seaborn
```

```
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.12/site-
packages (3.10.1)
Requirement already satisfied: contourpy>=1.0.1 in
/opt/conda/lib/python3.12/site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.12/site-
packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/conda/lib/python3.12/site-packages (from matplotlib) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/opt/conda/lib/python3.12/site-packages (from matplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in /opt/conda/lib/python3.12/site-
packages (from matplotlib) (2.2.4)
Requirement already satisfied: packaging>=20.0 in
/opt/conda/lib/python3.12/site-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /opt/conda/lib/python3.12/site-
packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/opt/conda/lib/python3.12/site-packages (from matplotlib) (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-dateutil>=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) (2.2.4)
Requirement already satisfied: pandas>=1.2 in /opt/conda/lib/python3.12/site-
packages (from seaborn) (2.2.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in
```

```
/opt/conda/lib/python3.12/site-packages (from seaborn) (3.10.1)
Requirement already satisfied: contourpy>=1.0.1 in
/opt/conda/lib/python3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(1.3.1)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.12/site-
packages (from matplotlib!=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 matplotlib!=3.6.1,>=3.4->seaborn)
(4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/opt/conda/lib/python3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(1.4.8)
Requirement already satisfied: packaging>=20.0 in
/opt/conda/lib/python3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(24.2)
Requirement already satisfied: pillow>=8 in /opt/conda/lib/python3.12/site-
packages (from matplotlib!=3.6.1,>=3.4->seaborn) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/opt/conda/lib/python3.12/site-packages (from matplotlib!=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 pandas>=1.2->seaborn) (2025.2)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-
packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)
```

[12]:
```python
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Connect to the SQLite database
conn = sqlite3.connect('survey-results-public.sqlite')
```

1.2 Demo: Basic SQL Queries

Demo 1: Count the Number of Rows in the Table

[5]:
```python
QUERY = "SELECT COUNT(*) FROM main"
df = pd.read_sql_query(QUERY, conn)
print(df)
```

```
   COUNT(*)
0     65437
```

Demo 2: List All Tables

[6]:
```
QUERY = """
SELECT name as Table_Name
FROM sqlite_master
WHERE type = 'table'
"""
pd.read_sql_query(QUERY, conn)
```

[6]:

	Table_Name
0	main

Demo 3: Group Data by Age

[7]:
```
QUERY = """
SELECT Age, COUNT(*) as count
FROM main
GROUP BY Age
ORDER BY Age
"""
df_age = pd.read_sql_query(QUERY, conn)
print(df_age)
```

```
                  Age  count
0     18-24 years old  14098
1     25-34 years old  23911
2     35-44 years old  14942
3     45-54 years old   6249
4     55-64 years old   2575
5   65 years or older    772
6   Prefer not to say    322
7  Under 18 years old   2568
```

1.3 Visualizing Data

1.3.1 Task 1: Visualizing the Distribution of Data

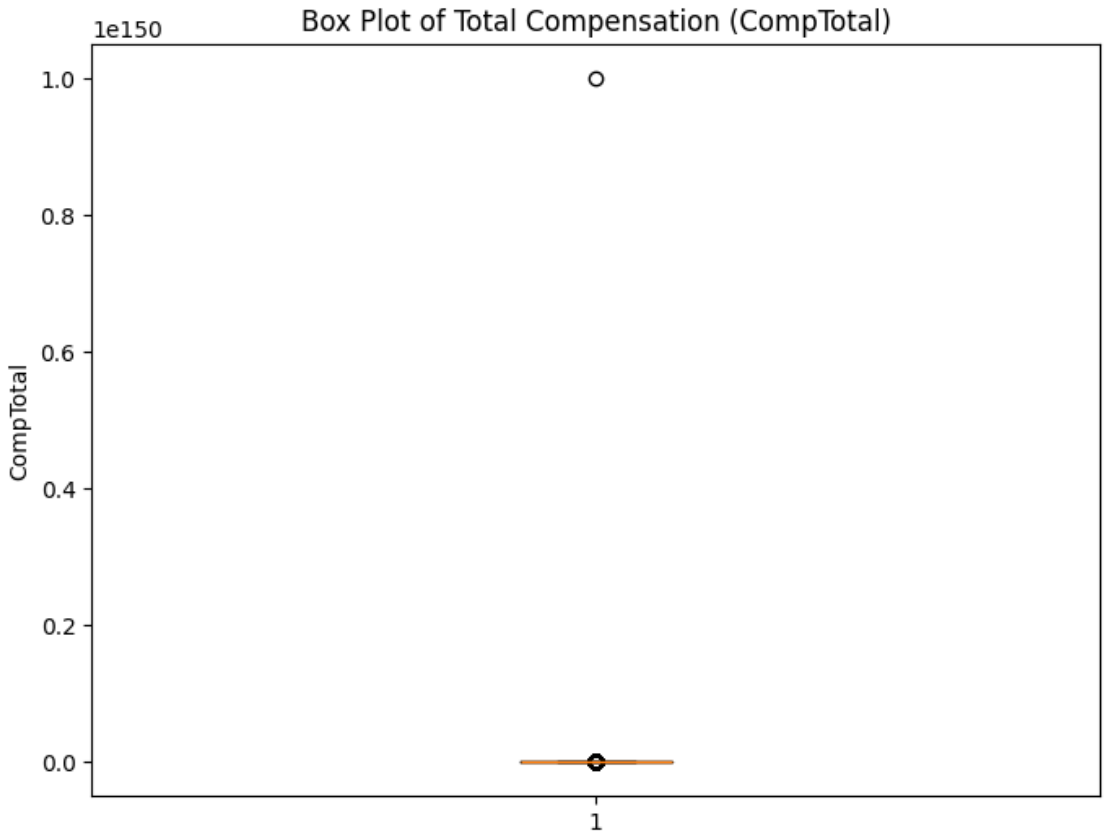
1. Box Plot of `CompTotal` (Total Compensation)

Use a box plot to analyze the distribution and outliers in total compensation.

[8]:
```
query_check = "SELECT CompTotal FROM main WHERE CompTotal IS NOT NULL LIMIT 5"
df_check = pd.read_sql(query_check, conn)
print(df_check)


if not df_check.empty:
    query = "SELECT CompTotal FROM main WHERE CompTotal IS NOT NULL"
    df = pd.read_sql(query, conn)


    plt.figure(figsize=(8, 6))
```

```
    plt.boxplot(df['CompTotal'])
    plt.title('Box Plot of Total Compensation (CompTotal)')
    plt.ylabel('CompTotal')
    plt.show()
else:
    print("No data found for CompTotal.")
```

```
   CompTotal
0  2040000.0
1    28000.0
2    85000.0
3    50000.0
4   110000.0
```



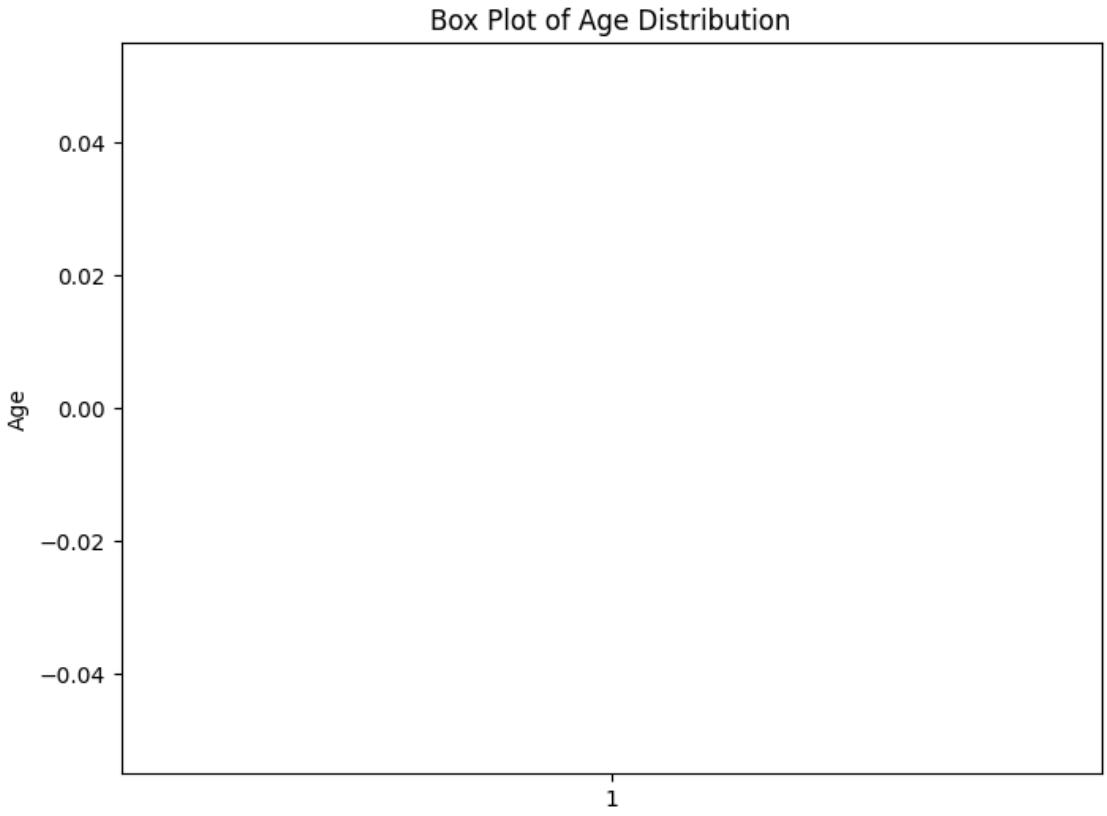
2. Box Plot of Age (converted to numeric values)

Convert the `Age` column into numerical values and visualize the distribution.

[13]:
```
# Check the first few rows to inspect the Age column
query_check = "SELECT Age FROM main LIMIT 5"
df_check = pd.read_sql(query_check, conn)
```

```
print(df_check)

# Convert 'Age' to numeric (errors='coerce' will convert non-numeric values to␣
 ↪NaN)
query = "SELECT Age FROM main WHERE Age IS NOT NULL"
df = pd.read_sql(query, conn)

# Convert the Age column to numeric, coercing errors to NaN (non-numeric␣
 ↪entries)
df['Age'] = pd.to_numeric(df['Age'], errors='coerce')

# Drop rows where Age is NaN (optional, depending on your preference)
df_clean = df.dropna(subset=['Age'])

# Plot the boxplot
plt.figure(figsize=(8, 6))
plt.boxplot(df_clean['Age'])
plt.title('Box Plot of Age Distribution')
plt.ylabel('Age')
plt.show()
```

```
                  Age
0  Under 18 years old
1     35-44 years old
2     45-54 years old
3     18-24 years old
4     18-24 years old
```



1.3.2 Task 2: Visualizing Relationships in Data

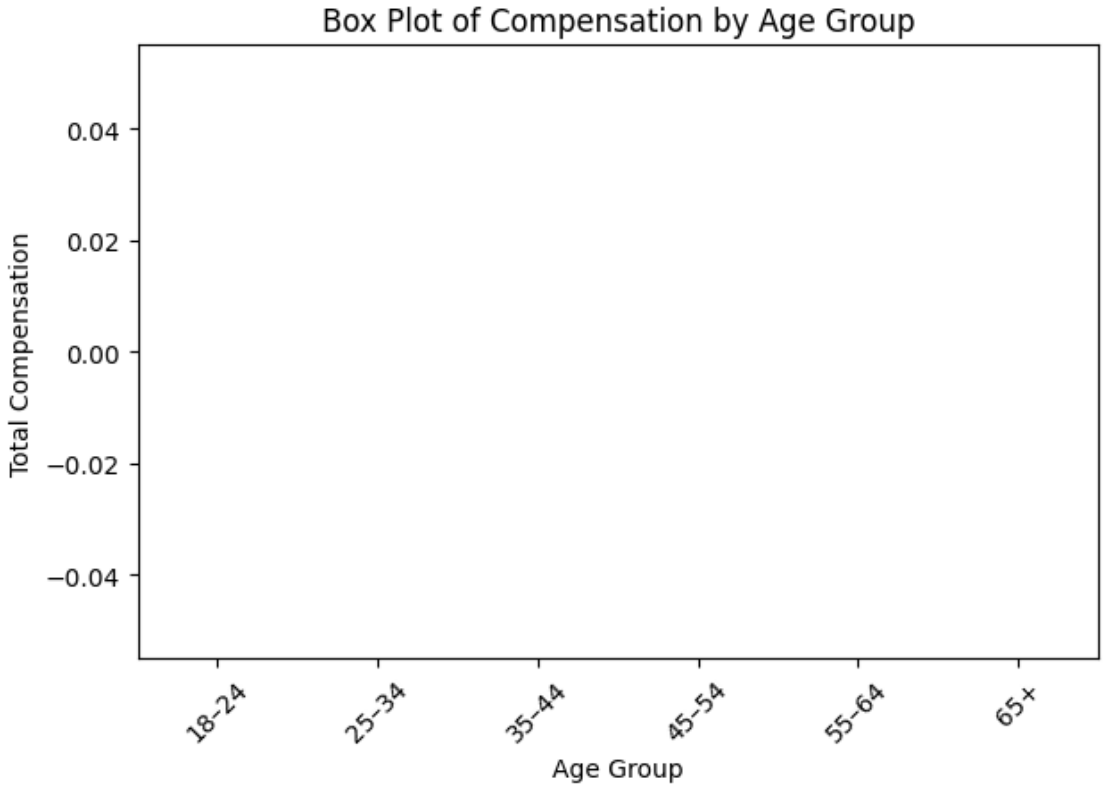
1. Box Plot of `CompTotal` Grouped by Age Groups:

Visualize the distribution of compensation across different age groups.

[14]:
```
# Load Age and CompTotal columns
query = "SELECT Age, CompTotal FROM main"
df = pd.read_sql(query, conn)

# Close the connection
conn.close()

# Convert 'Age' to numeric and drop rows with NaNs
df['Age'] = pd.to_numeric(df['Age'], errors='coerce')
df['CompTotal'] = pd.to_numeric(df['CompTotal'], errors='coerce')
df = df.dropna(subset=['Age', 'CompTotal'])

# Define age groups
bins = [18, 25, 35, 45, 55, 65, 100]
labels = ['18-24', '25-34', '35-44', '45-54', '55-64', '65+']
```

```
df['AgeGroup'] = pd.cut(df['Age'], bins=bins, labels=labels, right=False)

# Drop rows with undefined age groups (if any)
df = df.dropna(subset=['AgeGroup'])

# Plot box plot
plt.figure(figsize=(10, 6))
df.boxplot(column='CompTotal', by='AgeGroup', grid=False)
plt.title('Box Plot of Compensation by Age Group')
plt.suptitle('')
plt.xlabel('Age Group')
plt.ylabel('Total Compensation')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```



2. Box Plot of `CompTotal` Grouped by Job Satisfaction (`JobSatPoints_6`):

Examine how compensation varies based on job satisfaction levels.

[ ]:
```
# your code goes here
```

1.3.3 Task 3: Visualizing the Composition of Data

1. Box Plot of `ConvertedCompYearly` for the Top 5 Developer Types:

Analyze compensation across the top 5 developer roles.

[ ]:
```
# your code goes here
```

2. Box Plot of `CompTotal` for the Top 5 Countries:

Analyze compensation across respondents from the top 5 countries.

[ ]:
```
# your code goes here
```

1.3.4 Task 4: Visualizing Comparison of Data

1. Box Plot of CompTotal Across Employment Types:

Analyze compensation for different employment types.

[ ]:
```
# your code goes here
```

2. Box Plot of `YearsCodePro` by Job Satisfaction (`JobSatPoints_6`):

Examine the distribution of professional coding years by job satisfaction levels.

[ ]:
```
# your code goes here
```

1.3.5 Final Step: Close the Database Connection

After completing the lab, close the connection to the SQLite database:

[ ]:
```
conn.close()
```

1.4 Summary

In this lab, you used box plots to visualize various aspects of the dataset, focusing on:

- Visualize distributions of compensation and age.
- Explore relationships between compensation, job satisfaction, and professional coding experience.
- Analyze data composition across developer roles and countries.
- Compare compensation across employment types and satisfaction levels.

Box plots provided clear insights into the spread, outliers, and central tendencies of various features in the dataset.

1.5 Authors:

Ayushi Jain

1.5.1 Other Contributors:

- Rav Ahuja
- Lakshmi Holla
- Malika