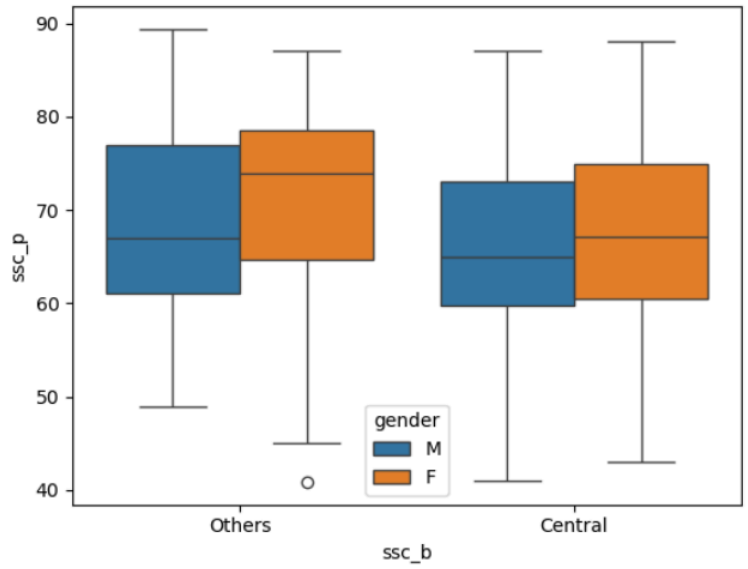
Seaborn - BoxPlot:

Box Plot

```
[29]: sns.boxplot(x='ssc_b',y='ssc_p',data=dataset,hue='gender')
      plt.show()
```



A box plot (also called a box-and-whisker plot) is a standardized way of displaying the distribution of data based on a five-number summary: minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum. It's super useful for spotting outliers and understanding the spread and skewness of the data.

In Seaborn, a popular Python data visualization library built on top of Matplotlib, you can easily create box plots with the `seaborn.boxplot()` function.

🔹 What a Box Plot Shows:

- **Box: Represents the interquartile range (IQR), from Q1 to Q3.**
- **Line inside the box: The median (Q2).**
- **Whiskers: Extend to the minimum and maximum values within 1.5 * IQR.**
- **Dots outside the whiskers: Outliers, values that are unusually far from the rest.**

🔹 What Are Whiskers in a Box Plot?

Whiskers are the lines that extend from the box in a box plot. They show the range of the data outside the interquartile range (IQR), but not including outliers.

🔹 How Are They Calculated?

By default (in Seaborn and most box plot implementations), whiskers extend to:

📍 Lowest data point within 1.5 × IQR below Q1
📍 Highest data point within 1.5 × IQR above Q3

Where:

- **Q1 = 25th percentile (the start of the box)**
- **Q3 = 75th percentile (the end of the box)**
- **IQR = Q3 - Q1**

So:

- **Lower whisker = Q1 - 1.5 × IQR**
- **Upper whisker = Q3 + 1.5 × IQR**

Any data point outside this range is plotted individually as an outlier (usually as dots or circles).

🔹 Why Are Whiskers Important?

They give you a sense of:

- **Spread of most of the data**
- **Where the extremes are (without being distorted by outliers)**

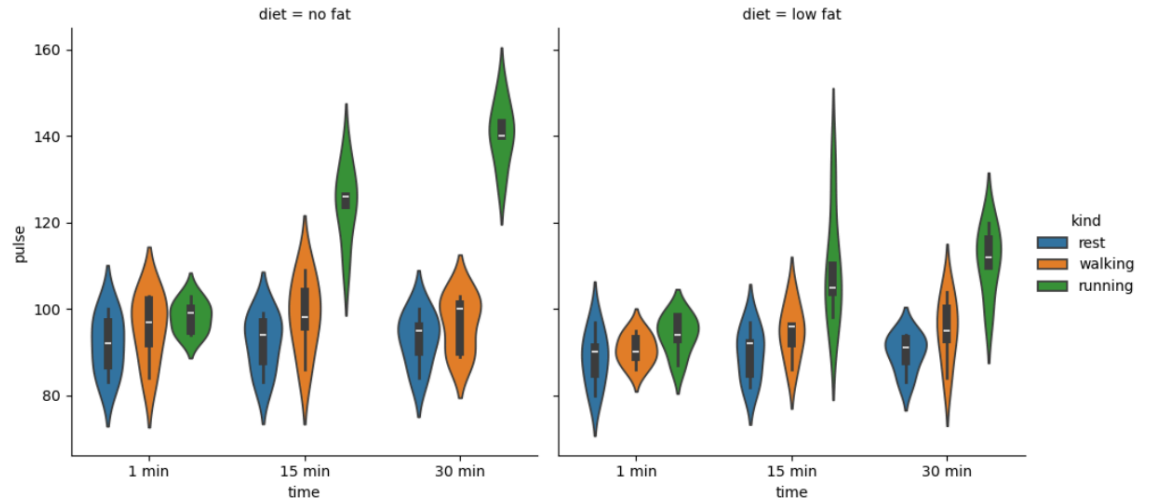
🔹 When to Use a Box Plot:

- **Comparing distributions across categories**
- **Detecting outliers**

- **Getting a quick sense of data spread and central tendency**

Violin Plot :

```
[38]: import pandas as pd
      import seaborn as sns
      from matplotlib import pyplot as plt
      df=sns.load_dataset('exercise')
      sns.catplot(x='time',y='pulse',hue='kind',kind='violin',col='diet',data=df)
      plt.show()
```



Dataset: `'exercise'`

This is a built-in dataset from Seaborn that contains pulse (heart rate) data based on:

- **`time`: The time of exercise (like 1 min, 15 min, etc.)**
- **`pulse`: Heart rate**
- **`kind`: Type of activity (e.g., rest, walking, running)**
- **`diet`: Type of diet (e.g., low fat, no fat)**

Violin Plot — What It Is

A violin plot combines a box plot and a KDE plot (smoothed distribution). It helps you see:

- **Distribution of the data (like a mirrored density plot)**

- **Median and quartiles (like a box plot)**
- **Spread and shape of the data**

Creating multiple violin plots showing:

- `x='time'`: **Time of exercise (like 1 min, 15 min, 30 min) on the x-axis**
- `y='pulse'`: **Pulse (heart rate) on the y-axis**
- `hue='kind'`: **Activity type shown in different colors (rest, walking, running)**
- `col='diet'`: **Creates separate plots (columns) for each diet type (low fat and no fat)**
- `kind='violin'`: **The actual chart type used**

The Plot Shows:

- **For each time point, it shows how pulse varies for different activities.**
- **You'll see separate violins for each `kind` (activity) inside each time group.**
- **Since `col='diet'`, the figure is split into two subplots (one for each diet type).**

What You Can Interpret:

- **How heart rate (`pulse`) changes over time and exercise type**
- **How different activities (rest, walk, run) affect pulse**
- **Whether the diet makes a difference in pulse trends**
- **The distribution shape — e.g., skewed, multimodal, etc.**
- **Presence of outliers or variation**

Why it's useful:

It makes comparison easier between groups — especially when you want to see how one variable (like pulse) behaves under different categories (like diet types) without overlapping everything into one chart.