Outlier detection and treatment

✅ What Are Outliers?

An outlier is a data point that lies far outside the normal range of values in a dataset.
For example:

Example 1: Student Scores

Student Math Score

Student	**Math Score**
A	85
B	88
C	91
D	87
E	**20** (▲ Outlier)

⍰ Most students scored around 85–91,
but Student E got 20, which is very far from the others.
✅So 20 is an outlier.

✅Why Are Outliers Important?

Reason	**Explanation**
⚠ Biases the Mean	Outliers can shift the average drastically
📉 Affects Model Accuracy	Many ML models (like Linear Regression) are sensitive
✅ Sometimes Insightful	Can reveal fraud, system failures, or rare cases

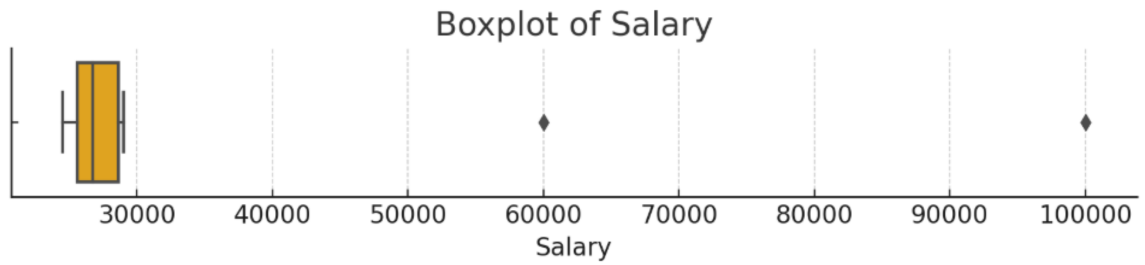
3)Visualize the Data

```
sns.boxplot(x=df['Salary'])

plt.title("Boxplot of Salary")

plt.show()
```

Explanation:

The box represents the middle 50% of the data.

Dots outside the “whiskers” are outliers.



As you can see:

The box shows the middle 50% of salaries (around 25k–29k).

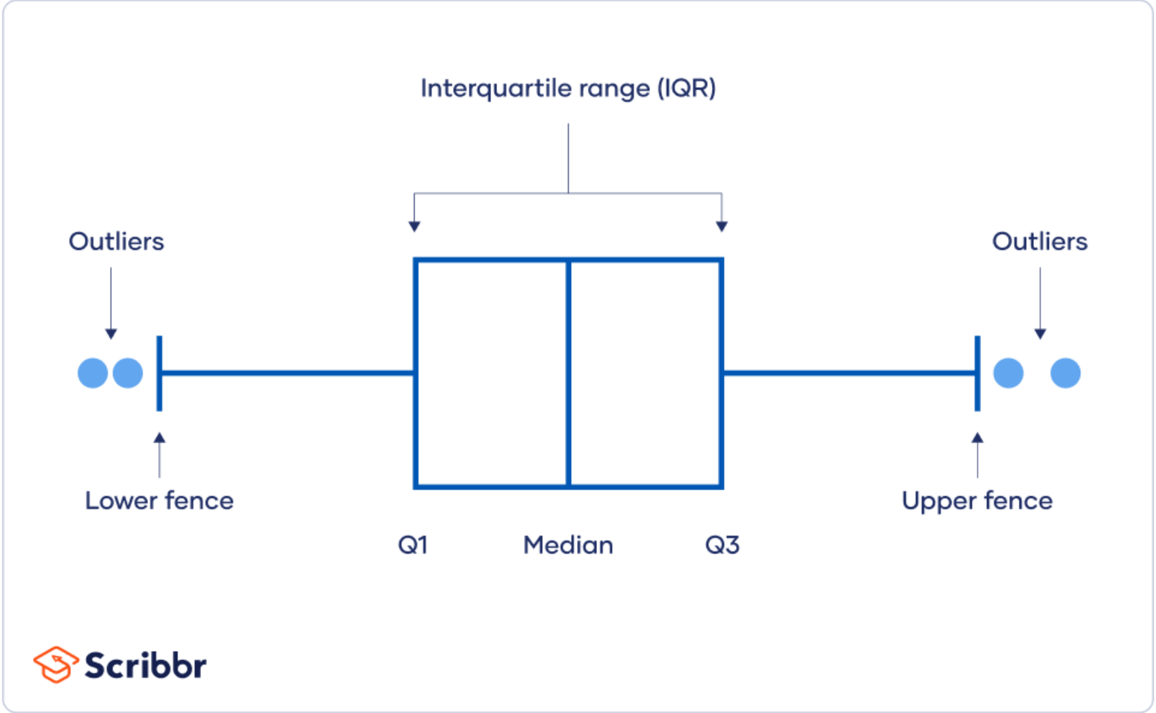
The dots to the far right (60,000 and 100,000) are outliers — values much higher than the rest.

✅Detect Outliers Using the IQR Method

We use the IQR method to define what’s “too far” from the normal range.

25% of the data falls below this value.

75% of the data falls below this value.



```
Q1 = df['Salary'].quantile(0.25)

Q3 = df['Salary'].quantile(0.75)

IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR

upper_bound = Q3 + 1.5 * IQR

outliers = df[(df['Salary'] < lower_bound) | (df['Salary'] > upper_bound)]

print(outliers)
```

```
Lower Bound: 937

Upper Bound: 1437
```

```
Salary
0    1000
1    1100
2    1150
3    1200
4    1300
5   20000
```

Output:

```
  Salary
5   20000
```

☑ Choose a Treatment Method

☑ Remove the Outlier

Python

Copy code

```
df_removed = df[(df['Salary'] >= lower_bound) & (df['Salary'] <= upper_bound)]
print(df_removed)
```

□ Use this when:

It represents <5% of the data

✅Cap the Outlier (Winsorize)

python

Copy code

```
df_capped = df.copy()

df_capped['Salary'] = df_capped['Salary'].clip(lower=lower_bound,
upper=upper_bound)

print(df_capped)
```

output

```
 5   1437
```

□ Use this when:

You want to keep all data but reduce the effect of outliers

You want to prevent the model from overfitting(*) to extreme values

The outliers are valid but too extreme

Don't cap if:

If you’re analyzing true extremes (e.g., VIP users, fraud)

✅Replace the Outlier with Median

□ When:

The outlier is likely an error or noise

You're working with skewed data (median works better than mean).

You want consistent scaling or normalization

Data is small

python

Copy code

```
median_salary = df['Salary'].median()  # 1200

df_replaced = df.copy()

df_replaced.loc[(df['Salary'] > upper_bound) | (df['Salary'] < lower_bound), 'Salary'] = median_salary

print(df_replaced)
```

Summary

Step	**Description**
Step 1: Understand Data	See the values and know what the data means
Step 2: Visualize	Use boxplot to spot obvious outliers
Step 3: Detect	Use IQR
Step 4: Treat	Remove / Cap / Replace