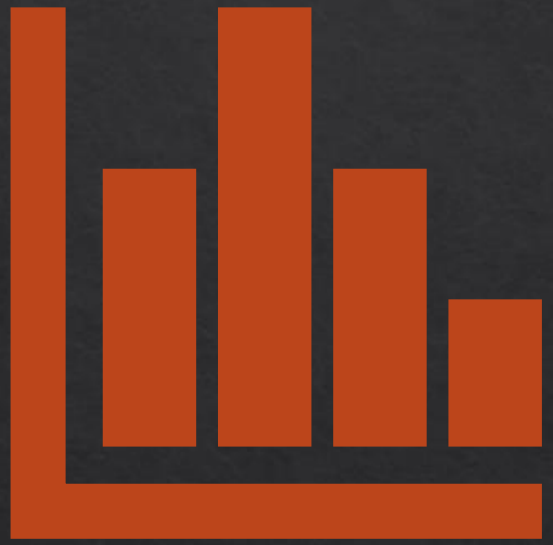
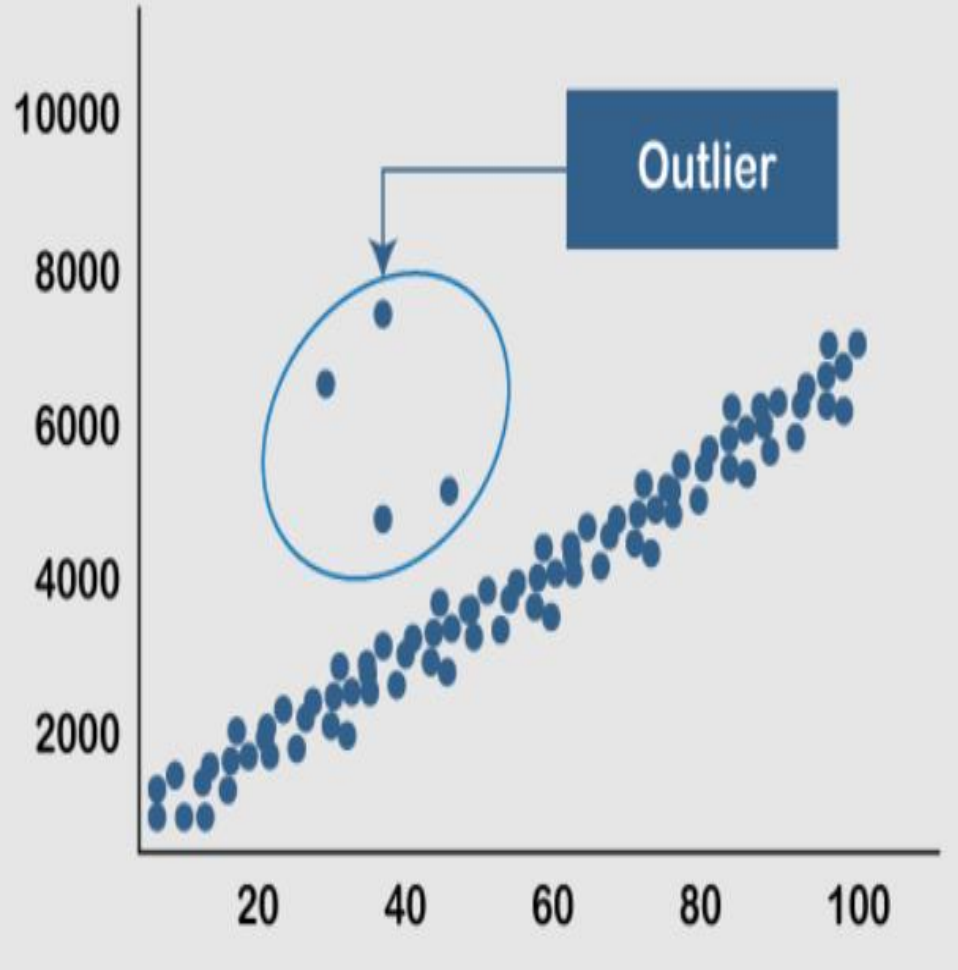


Understanding and Managing Outliers in Data Science

The Significance of Addressing Outliers

- Treating outliers ensures that statistical measures accurately reflect the central tendencies of the data without being skewed by extreme values.
- By addressing outliers, machine learning models can make more reliable predictions, as they are less likely to be influenced by irregular data points.
- Outliers can obscure trends and patterns in data visualization, so removing them leads to clearer and more accurate insights.
- Outliers can violate the assumptions of statistical tests, potentially compromising the validity of analyses. Treating outliers helps maintain the integrity of these assumptions and ensures reliable results.
- Addressing outliers helps identify and rectify errors or anomalies in data collection processes, promoting higher-quality and more trustworthy datasets for analysis.

Outliers are like the odd ones out in a group of data points. They're the numbers that don't fit in with the rest. These outliers can mess up our analysis by making things look different than they really are. We find outliers using tools like looking at how far a number is from the average or comparing it to the other numbers. Handling outliers means deciding if we should keep them, change them, or get rid of them. It's important to deal with outliers carefully so our data analysis is accurate and reliable.



Arun is interested in losing weight through exercise. He is comparing two workout plans, A and B, to decide which one to follow. Plan A has an average weight loss of 4.65 kg, while plan B has an average weight loss of 3.16 kg. Based on this information, Arun decides to go with workout plan A because it has a higher average weight loss. To determine if Arun made the right decision, we need to conduct a more comprehensive analysis to ascertain which plan is truly effective?

Workout Plan A

```
Workout_Plan_A = {'Person': ['Alice', 'Bob', 'Charlie', 'David', 'Emma',
                  'Frank', 'Grace', 'Hannah', 'Ian', 'Jack'],
        'Total Weight Loss (kg)': [3.0, 30.5, 0.5, 2.3, 1.0,
                                   2.2, 3.0, 1.0, 1.0, 2.0]}

# Extracting total weight loss values
Workout_Plan_A_values = Workout_Plan_A['Total Weight Loss (kg)']

# Calculating mean
Workout_Plan_A_Mean = np.mean(Workout_Plan_A_values)
print("Average Weigtloss for gym A is :", np.round((Workout_Plan_A_Mean),2))
```

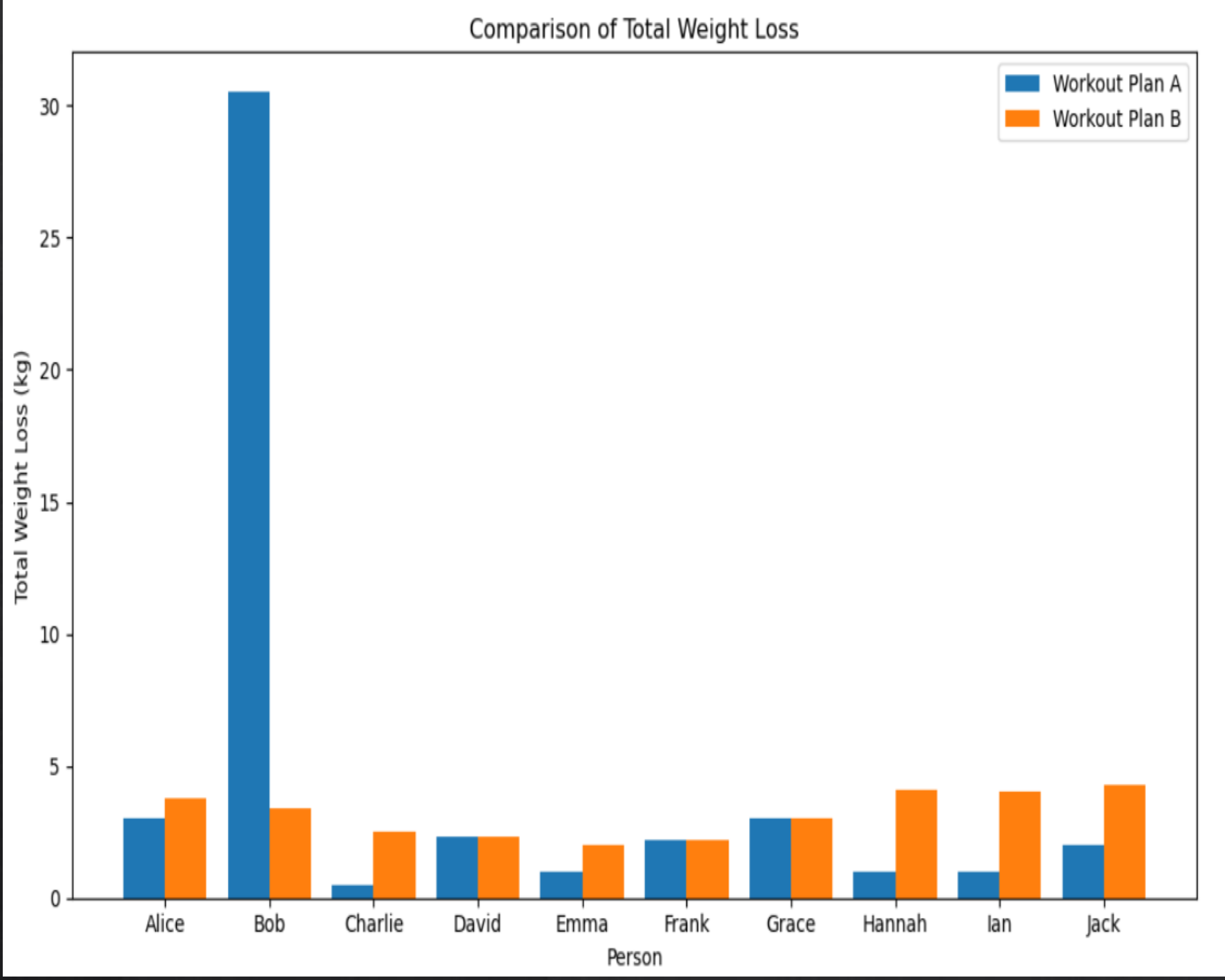
```
Average Weigtloss for gym A is : 4.65
```

Workout Plan B

```
Workout_Plan_B = {'Person': ['Alice', 'Bob', 'Charlie', 'David', 'Emma',
                  'Frank', 'Grace', 'Hannah', 'Ian', 'Jack'],
        'Total Weight Loss (kg)': [3.8,3.4, 2.5, 2.3, 2.0,
                                   2.2, 3.0, 4.1, 4.0, 4.3]}

# Extracting total weight loss values
Workout_Plan_B_values = Workout_Plan_B['Total Weight Loss (kg)']

# Calculating mean and median
Workout_Plan_A_Mean = np.mean(Workout_Plan_B_values)

print("Average Weigtloss for Workout B is :", np.round((Workout_Plan_A_Mean),2))
```

```
Average Weigtloss for Workout B is : 3.16
```

Bob's weight loss data is noticeably different from the others, making it an outlier in the dataset. Outliers like Bob's can skew the results of our analysis, potentially leading to inaccurate conclusions. Therefore, it's essential to address these outliers to ensure the accuracy of our findings. By correcting outliers like Bob's weight loss, we aim to improve the precision and reliability of our analysis.



While the IQR Method and Imputing Median Method are effective strategies for handling outliers, there exist various other techniques tailored to specific dataset characteristics and analytical goals. The choice of method depends on factors such as the nature of the data, the analysis objectives, and the desired level of data reliability. By employing appropriate outlier treatment methods, analysts can ensure the integrity and accuracy of their data analysis results.

IQR Method

```
# Step 1: Calculate the IQR
Q1 = np.percentile(weight_loss_values, 25)
Q3 = np.percentile(weight_loss_values, 75)
IQR = Q3 - Q1
# Step 2: Define the lower and upper bounds for outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
# Step 3: Identify and remove outliers
outliers_removed = weight_loss_values[(weight_loss_values >= lower_bound) & (weight_loss_values <= upper_bound)]
# Calculate the corrected mean after removing outliers
corrected_mean_iqr = np.mean(outliers_removed)
print("Method 1: Outlier Reduction using IQR Method")
print("Corrected Mean after Outlier Removal (IQR Method):", np.round((corrected_mean_iqr),2))
```

```
Method 1: Outlier Reduction using IQR Method
Corrected Mean after Outlier Removal (IQR Method): 1.78
```

Imputing Median Value

```
# Step 1: Calculate the median of the dataset
median = np.median(weight_loss_values)

# Step 2: Replace outliers with the median value
imputed_values = np.where((weight_loss_values < np.percentile(weight_loss_values, 25)) |
                          (weight_loss_values > np.percentile(weight_loss_values, 75)),
                          median, weight_loss_values)

# Calculate the corrected mean after imputing median value for outliers
corrected_mean_imputation = np.mean(imputed_values)
print("Method 2: Outlier Reduction using Imputing Median Value")
print("Corrected Mean after Imputing Median Value for Outliers:", np.round((corrected_mean_imputation),2))
```

```
Method 2: Outlier Reduction using Imputing Median Value
Corrected Mean after Imputing Median Value for Outliers: 1.79
```

Conclusion:

After addressing outliers, we proceed to compare the accurate mean values for Workout Plan A and Workout Plan B. Arun opted for Workout Plan A, unaware of its lower efficiency, without delving into further analysis. It's evident that Workout Plan B yields more efficient weight loss results. This instance underscores the significant impact outliers have on model accuracy. Therefore, it's imperative to effectively handle outliers before making predictions.

Workout Plan A

```
# Step 1: Calculate the IQR
Q1 = np.percentile(weight_loss_values, 25)
Q3 = np.percentile(weight_loss_values, 75)
IQR = Q3 - Q1
# Step 2: Define the lower and upper bounds for outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
# Step 3: Identify and remove outliers
outliers_removed = weight_loss_values[(weight_loss_values >= lower_bound) & (weight_loss_values <= upper_bound)]
# Calculate the corrected mean after removing outliers
corrected_mean_iqr = np.mean(outliers_removed)
print("Method 1: Outlier Reduction using IQR Method")
print("Corrected Mean after Outlier Removal (IQR Method):", np.round((corrected_mean_iqr),2))
```

```
Method 1: Outlier Reduction using IQR Method
Corrected Mean after Outlier Removal (IQR Method): 1.78
```

Workout Plan B

```
Workout_Plan_B = {'Person': ['Alice', 'Bob', 'Charlie', 'David', 'Emma',
                             'Frank', 'Grace', 'Hannah', 'Ian', 'Jack'],
                  'Total Weight Loss (kg)': [3.8,3.4, 2.5, 2.3, 2.0,
                                             2.2, 3.0, 4.1, 4.0, 4.3]}

# Extracting total weight loss values
Workout_Plan_B_values = Workout_Plan_B['Total Weight Loss (kg)']

# Calculating mean and median
Workout_Plan_A_Mean = np.mean(Workout_Plan_B_values)

print("Average Weigtloss for Workout B is :", np.round((Workout_Plan_A_Mean),2))
```

```
Average Weigtloss for Workout B is : 3.16
```