



Summarization

- **Pandas** is a powerful library in Python used for data manipulation and analysis. Summarization refers to the process of obtaining a concise description or representation of the dataset.

1-Descriptive Statistics: Pandas provides various methods to compute descriptive statistics for numerical data, such as mean, median, standard deviation, minimum, maximum, etc. These methods include `mean()`, `median()`, `std()`, `min()`, `max()`, `describe()`, etc.

2-Aggregation: You can aggregate data using functions like `groupby()` followed by an aggregation function such as `sum()`, `mean()`, `count()`, etc.

- operators and methods commonly used in pandas for data manipulation

<: Less than

Used to check if one value is less than another.

!=: Not equal to

Used to check if two values are not equal.

>: Greater than

Used to check if one value is greater than another

&: Logical AND

|: Logical OR

~: Logical NOT

Make New Columns:

In pandas, you can create new columns in a DataFrame based on existing columns or computations. This is often done using the assignment operator (=) with the new column name. Here's an example:

```
df['new_column'] = df['existing_column'] * 2
```

Rows and Columns:

- In pandas, you can access rows and columns of a DataFrame using various methods. You can use indexing and slicing for selecting rows or columns based on their position or label. Here are some examples:

```
# Select a single column
```

```
column_data = df['column_name']
```

```
# Select multiple columns
```

```
subset = df[['column1', 'column2']]
```

```
# Select rows by index label
```

```
row = df.loc['index_label']
```

```
# Select rows by integer index
```

```
row = df.iloc[integer_index]
```

Tidy Data:

Tidy data complements pandas's vectorized operations. pandas will automatically preserve observations as you manipulate variables. No other format works as intuitively with pandas

Method Chaining

Most pandas methods return a DataFrame so that another pandas method can be applied to the result. This improves readability of code.

```
df = (pd.melt(df)
      .rename(columns={
'variable':'var', 'value':'val'}) .query('val >= 200')
)
```