Combining Data With Pandas: Takeaways

Syntax

Concat() Function

Concatenate dataframes vertically (axis=0):

```
pd.concat([df1, df2])
```

Concatenate dataframes horizontally (axis=1):

```
pd.concat([df1, df2], axis=1)
```

Concatenate dataframes with an inner join:

```
pd.concat([df1, df2], join='inner')
```

Merge() Function

Join dataframes on index:

```
pd.merge(left=df1, right = df2, left_index=True, right_index=True)
```

Customize the suffix of columns contained in both dataframes:

```
pd.merge(left=df1, right=df2, left_index=True, right_index=True,
suffixes=('left_df_suffix', 'right_df_suffix'))
```

Change the join type to left, right, or outer:

```
pd.merge(left= df1, right=df2, how='join_type', left_index=True,
right_index=True))
```

Join dataframes on a specific column:

```
pd.merge(left=df1, right=df2, on='Column_Name')
```

Concepts

A key or join key is a shared index or column that is used to combine dataframes together.
There are four kinds of joins:

- Inner: Returns the intersection of keys, or common values.
- Outer: Returns the union of keys, or all values from each dataframe.
- Left: Includes all of the rows from the left dataframe, along with any rows from the right dataframe with a common key. The result retains all columns from both of the original dataframes.
- Right: Includes all of the rows from the right dataframe, along with any rows from the left dataframe with a common key. The result retains all columns from both of the original dataframes. This join type is rarely used.

The `pd.concat()` function can combine multiple dataframes at once and is commonly used to "stack" dataframes, or combine them vertically (axis=0). The `pd.merge()` function uses keys to perform database-style joins. It can only combine two dataframes at a time and can only merge dataframes horizontally (axis=1).

Resources

- Merge and Concatenate