第9章 | Pandasの応用

Author: sharo

9.1 DataFrameの連結・結合の概観

9.1.1 連結・結合について

PandasではDataFrameに対して**連結**、**結合**という操作ができます。DataFrame同士を一定の方向についてそのままつなげる操作を**連結**、特定の**Key**を参照してつなげる操作を**結合**と言います。

9.2 DataFrameの連結

9.2.1 インデックス、カラムが一致しているDataFrame同士の連結

```
pandas.concat(<DataFrameのリスト>, [軸])
```

```
# default: axis = 0
pandas.concat([df1, df2], axis = 0) # 縦連結
pandas.concat([df1, df2], axis = 1) # 横連結
```

9.2.2 インデックス、カラムが一致していないDataFrame同士の連結

共通のインデックスやカラムでない行や列に**NaN**を持つセルが作成されます。

9.2.3 連結する際のラベルの指定

```
pandas.concat([df1, df2], axis = 1, keys = ['X', 'Y'])
```

```
      X                   Y
   apple orange banana apple orange banana
1     45     68     37    38     76     17
2     48     10     88    13      6      2
3     65     84     71    73     80     77
4     68     22     89    10     65     72
```

9.3 DataFrameの結合

9.3.1 結合の種類

結合のことを**マージ**とも呼びます。結合は、**Key**と呼ばれる列を指定し、2つのデータベースの**Key**内の値が一致する行を横につなげる操作です。

結合には大きく分けて**内部結合**と**外部結合**の2つの方法があります。

内部結合

Key列に共通の値がない行は破棄されます。

外部結合

Key列に共通の値がない行も残ります。共通でない列については**NaN**で埋められたセルが作成されます。

9.3.2 内部結合の基本

```
pandas.merge(df1, df2, on = key, how = 'inner')
```

```
pandas.merge(df1, df2, on = 'fruits', how = 'inner')
```

9.3.3 外部結合の基本

```
pandas.merge(df1, df2, on = key, how = 'outer')
```

```
pandas.merge(df1, df2, on = 'fruits', how = 'outer')
```

9.3.4 同名でない列をKeyにして結合する

```
pandas.merge(df1, df2, left_on = '左側DFのカラム', right_on = '右側DFのカラム', how = '結合方法')
```

9.3.5 インデックスをKeyにして結合する

9.3.4の `left_on` 、 `right_on` の値を `True` に設定するとインデックスを**Key**として結合します。

P280参照

9.4 DataFrameを用いたデータ分析

9.4.1 一部の行を得る

```
df.head() # 冒頭5行のみを含むDataFrameを返す
df.tail() # 末尾5行のみを含むDataFrameを返す
df.tail(3) # 末尾3行のみを含むDataFrameを返す
```

9.4.2 計算処理を適用する

```
double_df = df * 2 # double_df = df + df
square_df = df * df # square_df = df ** 2
sqrt_df = np.sqrt(df)
```

9.4.3 要約統計量を得る

列ごとの平均値、最大値、最小値などの統計的情報をまとめたものを**要約統計量**と呼びます。DataFrame型変数`df`に対して、`df.describe()`は`df`の列ごとの**個数**、**平均値**、**標準偏差**、**最小値**、**四分位数**、**最大値**を含むDataFrameを返します。得られたDataFrameのインデックスは統計量の名前になります。

```
# dfの要約統計量のうち、'mean', 'max', 'min'を取り出してdf_desに代入
df_des = df.describe().loc[['mean', 'max', 'min']]
```

9.4.4 DataFrameの行間または列間の差を求める

```
df.diff(<行または列の間隔>, [軸])
```

```
# 第1引数が正の場合前の行との差、負の場合は後の行との差
# axis = 0 -> 行方向, axis = 1 -> 列方向
df.diff(-2, axis = 0) # 2行後の行との差を計算したDataFrameを生成
```

9.4.5 グループ化

データベースやDataFrameに対して、ある特定の列について同じ値を持つ行を集約することを**グループ化**と呼びます。GroupByオブジェクトに対して、各グループの平均値を求める`mean()`や和を求める`sum()`などの演算を行うことができます。

```
grouped_region = prefecture_df.groupby('Regioin')
mean_df = grouped_region.mean()
```