


Análisis y Manipulación de Datos 2

Data BootCamp
Cámara Comercio Bilbao

Germán Alonso Lascurain
germanalonso@opendeusto.es

CÁMARABILBAO 

 C2B
campus to business

 ai
ai power

 tigloo

Índice de contenidos

1. Unión de DataFrames
2. Pivotaje de DataFrames
3. Cálculos sobre Series Temporales

1. Unión de DataFrames

Unión de DataFrames - join

Invocación:

```
import pandas as pd
df1 = pd.DataFrame(datos1, index=indice)
df2 = pd.DataFrame(datos2, index=indice)
df1.join(df2)
```

Parámetros de interés:

- on: columnas o índices sobre los que unir
- how: forma de la unión

Unión de DataFrames - merge

Invocación:

```
import pandas as pd
df1 = pd.DataFrame(datos1, index=indice)
df2 = pd.DataFrame(datos2, index=indice)
df1.merge(df2)
```

Parámetros de interés:

- on: columnas o índices sobre los que unir
- how: forma de la unión

Diferencias frente a Join:

- Permite unir de forma independiente cada DataFrame
- Podemos comprobar el resultado de la unión
- No podemos unir varios DataFrames

Unión de DataFrames - join vs merge

Join: Funciona mejor uniendo dataframes a través de sus índices.

Merge: Es más flexible que Join y nos permite especificar las columnas a través de las cuales generar la unión.

Unión de DataFrames - append

Invocación:

```
import pandas as pd
df1 = pd.DataFrame(datos1, index=indice)
df2 = pd.DataFrame(datos2, index=indice)
df1.append(df2)
```

Parámetros de interés:

- `ignore_index`: reinicia el índice
- `sort`: ordena el eje sobre el que se une antes de hacer la operación

Unión de DataFrames - concat

Invocación:

```
import pandas as pd
df1 = pd.DataFrame(datos1, index=indice)
df2 = pd.DataFrame(datos2, index=indice)
pd.concat([df1,df2])
```

Parámetros de interés:

- axis: eje sobre el que concatenar
- ignore_index: reinicia el índice
- sort: ordena el eje sobre el que se une antes de hacer la operación

2. Pivotaje de DataFrames

Pivotaje de DataFrames - transpose

Invocación:

```
import pandas as pd
df = pd.DataFrame()
df.transpose()
```

Parámetros de interés:

- `copy`: realiza una copia del dataframe resultante

Pivotaje de DataFrames - pivot

Invocación:

```
import pandas as pd
df = pd.DataFrame()
df.pivot(index=indice,
          columns=etiquetas_nuevas_columnas,
          values=valores_nuevas_columnas)
```

Pivotaje de DataFrames - melt

Invocación:

```
import pandas as pd
df = pd.DataFrame()
df.melt(id_vars=columnas_indice,
        value_vars=columnas a desvincular)
```

Parámetros de interés:

- `var_name`: nombre para la columna de etiquetas
- `value_name`: nombre para la columna de valores

3. Cálculos sobre Series Temporales

Cálculos sobre Series Temporales - rolling

Invocación:

```
import pandas as pd
df = pd.DataFrame(datos, index=datetime_index)
df_rolling = df.rolling(ventana)
df_rolling.mean()
```

Parámetros de interés:

- `min_periods`: Número mínimo de observaciones
- `win_type`: función de enventanado

Cálculos sobre Series Temporales - shift

Invocación:

```
import pandas as pd
df = pd.DataFrame(datos, index=datetime_index)
df_shifted = df.shift()
```

Parámetros de interés:

- periods: Número de desplazamientos
- freq: Offset de los periodos
- fill_value: Valor por defecto para nulos

Cálculos sobre Series Temporales - cumsum, cumprod, cummax, cummin

Invocación:

```
import pandas as pd
df = pd.DataFrame(datos, index=datetime_index)
df_cumsum = df.cumsum()
```

Parámetros de interés:

- skipna: Ignorar los nulos

Cálculos sobre Series Temporales - diff

Invocación:

```
import pandas as pd
df = pd.DataFrame(datos, index=datetime_index)
df_diff = df.diff()
```

Parámetros de interés:

- **periods:** offset de posición con la que calcular la diferencia

Cálculos sobre Series Temporales - pct_change

Invocación:

```
import pandas as pd
df = pd.DataFrame(datos, index=datetime_index)
df_pct_change = df.pct_change()
```

Parámetros de interés:

- periods: offset de posición con la que calcular el porcentaje
- fill_method: como manejar los nulos antes de calcular el porcentaje
- freq: Incremento temporal a utilizar

Copyright (c) 2021 Germán Alonso Lascurain

This work (but the quoted images, whose rights are reserved to their owners*) is licensed under the

Creative Commons “Attribution-ShareAlike” License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>



Germán Alonso Lascurain
germanalonso@opendeusto.es