KIBERNUM
IT Academy

Módulo 2 – Obtención y Preparación de Datos

Data Wrangling II

Ciencia de Datos

Objetivos

Data Wrangling II

- Concatenación de dataframes.
- Combinación de dataframes.

KIBERNUM

Concatenación de dataframes

Sea el siguiente set de datos extraído desde el servicio de impuestos internos que contiene el valor de la UF de un mes determinado. Como se puede observar, la lectura ha tenido algunas dificultades, por lo tanto, debemos volver a realizar la carga agregando algunos parámetros adicionales.

```
df = pd.read_csv('Valor-UF-SII.csv')
df.head()
```

	1;28.601	15;11;28.636	35;21;28.664	96
0	2;28.604	83;12;28.639	21;22;28.667	82.0
1	3;28.608	51;13;28.642	07;23;28.670	68.0
2	4;28.612	20;14;28.644	93;24;28.673	54.0
3	5;28.615	88;15;28.647	79;25;28.676	41.0
4	6;28.619	57;16;28.650	65;26;28.679	27.0

KIBERNUM

Concatenación de Dataframes

Como se puede observar, se especificó el carácter de separación. Se indicó también, que el set de datos no trae header y se asignaron nombres de columnas. Ahora el set de datos luce de mejor forma.

```
# Lectura con separador, sin header y asignación de nombres de columnas
df = pd.read_csv('Valor-UF-SII.csv', sep=';', header=None,
                 names=['dia.1', 'valor.1', 'dia.2', 'valor.2', 'dia.3', 'valor.3'])
df.head()
```

	dia.1	valor.1	dia.2	valor.2	dia.3	valor.3
0	1.0	28.601,15	11.0	28.636,35	21	28.664,96
1	2.0	28.604,83	12.0	28.639,21	22	28.667,82
2	3.0	28.608,51	13.0	28.642,07	23	28.670,68
3	4.0	28.612,20	14.0	28.644,93	24	28.673,54
4	5.0	28.615,88	15.0	28.647,79	25	28.676,41

Concatenación de Dataframes

Crearemos tres dataframes independientes con la información de día y valor. Nótese que hemos utilizado el método **copy()** para crear un objeto distinto y no trabajar con referencias a un mismo objeto.

```
# creación de 3 nuevos dataframes
# seleccionando las respecticas columnas
df1 = df.iloc[:,:2].copy()
df2 = df.iloc[:,2:4].copy()
df3 = df.iloc[:,4:].copy()
```

```
df1.head(2)
```

Out[22]:

	dia.1	valor.1
0	1.0	28.601,15
1	2.0	28.604,83

```
df2.head(2)
```

Out[23]:

	dia.2	valor.2
0	11.0	28.636,35
1	12.0	28.639,21

```
df3.head(2)
```

Out[24]:

	dia.3	valor.3
0	21	28.664,96
1	22	28.667,82

KIBERNUM

Concatenación de dataframes

Ahora renombraremos las columnas, para que los tres dataframes tengan los mismos nombres.

```
#renombramiento de columnas
df1.rename(columns={'dia.1':'Dia', 'valor.1':'Valor'}, inplace=True)
df2.rename(columns={'dia.2':'Dia', 'valor.2':'Valor'}, inplace=True)
df3.rename(columns={'dia.3':'Dia', 'valor.3':'Valor'}, inplace=True)
```

```
df1.head(2)
```

	Dia	Valor
0	1.0	28.601,15
1	2.0	28.604,83

KIBERNUM

Concatenación de dataframes

```
# Concatenación de los 3 dataframes anteriores
dft = pd.concat([df1,df2,df3])
dft
```

	Dia	Valor
0	1.0	28.601,15
1	2.0	28.604,83
2	3.0	28.608,51
3	4.0	28.612,20
4	5.0	28.615,88
5	6.0	28.619,57
6	7.0	28.623,25
7	8.0	28.626,94
8	9.0	28.630,63
9	10.0	28.633,49
10	NaN	NaN
0	11.0	28.636,35
1	12.0	28.639,21
2	13.0	28.642,07
3	14.0	28.644,93

- Utilizaremos la función **concat()** que posee la librería Pandas para unir los dataframes.
- Nótese que esta función recibe como parámetro un listado con los dataframes que se busca concatenar.
- Sin embargo, han quedado algunas filas con valores NaN y el índice repite algunos valores. Ese será el último paso que realizaremos para ajustar el dataframe final.

Concatenación de dataframes

- Realizamos algunos ajustes finales, tales como eliminar los valores perdidos y reseteamos el índice.
- La opción drop=True en el método de reseteo del índice sirve para indicar que no deseamos que el índice sea promovido a columna, puesto que no es de utilidad en este caso.

```
# eliminamos valores nulos
dft.dropna(inplace=True)

# reseteamos el indice
dft.reset_index(inplace=True, drop=True)
```

```
dft
```

	Dia	Valor
0	1.0	28.601,15
1	2.0	28.604,83
2	3.0	28.608,51
3	4.0	28.612,20
4	5.0	28.615,88
5	6.0	28.619,57
6	7.0	28.623,25
7	8.0	28.626,94
8	9.0	28.630,63
9	10.0	28.633,49
10	11.0	28.636,35
11	12.0	28.639,21
12	13.0	28.642,07
13	14.0	28.644,93
14	15.0	28.647,79
15	16.0	28.650,65

Merge

Sea el siguiente set de datos que contiene información de los clientes, el cual incluye a su comuna y su segmento de cliente en un banco determinado.

```
df1 = pd.read_excel('riesgo-segmento.xlsx', sheet_name='Clientes')
df1
```

	ID_CLI	NOMBRE	COD_COMUNA	COD_SEGMENTO	SALDO
0	1	Cecilia Del Carmen Ayala Cabrera	C20	S01	1803344
1	2	Jesús Ignacio Contreras Vivar	C20	S01	236489
2	3	Carolina Andrea Estay Pangue	C01	S03	664647
3	4	Jorge Eduardo García Lagos	C10	S02	1279353
4	5	Carolina Lissett Gómez Morales	C02	S03	255036
5	6	Eugenio Leonardo Hidalgo Araya	C03	S04	315293
6	7	Evelyn Francesca Leiva Gajardo	C06	S03	287562
7	8	Andrea Mena Barrientos	C05	S02	843623
8	9	Vanessa Teresa Montenegro Robles	C06	S02	559817
9	10	Carol Muñoz Belmar	C08	S01	735440

Merge

Sean los siguientes sets de datos que contienen el maestro de comunas y los factores de riesgo aplicados a cada par segmento/comuna.

```
df2 = pd.read_excel('riesgo-segmento.xlsx',
                                    sheet_name='Comunas')
df2
```

	COD_COMUNA	COMUNA
0	C01	Santiago
1	C02	Rancagua
2	C03	Viña del Mar
3	C04	Concepción
4	C05	La Serena
5	C06	Iquique
6	C07	Puerto Montt
7	C08	Temuco
8	C09	Punta Arenas
9	C10	Antofagasta
10	C11	Villarrica
11	C12	Los Angeles

```
df2 = pd.read_excel('riesgo-segmento.xlsx',
                            sheet_name='Factores Riesgo')
df2
```

Out[45]:

	COD_SEGMENTO	COD_COMUNA	Factor Riesgo
0	S01	C01	0.10
1	S01	C02	0.20
2	S01	C03	0.20
3	S02	C10	0.10
4	S02	C05	0.30
5	S03	C02	0.25
6	S04	C04	0.10
7	S05	C12	0.40

KIBERNUM

Merge

Left dataframe

Rigth dataframe

Método de unión

Columna para hacer unión

Primero, realizaremos un merge entre el dataframe de clientes y de comunas. El método de unión es análogo a los joins de bases de datos. Un inner join corresponde a un producto cruz entre el dataframe left y right con todas las filas que hicieron match con la columna declarada en el parámetro **on.**
Sin embargo, se aprecia que ya no figuran algunos registros de cliente a causa de que no hubo match en el campo de la comuna. Esto es perjudicial puesto que se han perdido registros de clientes durante el merge.

```
# merge clientes comuna
pd.merge(df1,df2, how='inner', on='COD_COMUNA')
```

	ID_CLI	NOMBRE	COD_COMUNA	COD_SEGMENTO	SALDO	COMUNA
0	3	Carolina Andrea Estay Pangue	C01	S03	664647	Santiago
1	4	Jorge Eduardo García Lagos	C10	S02	1279353	Antofagasta
2	5	Carolina Lissett Gómez Morales	C02	S03	255036	Rancagua
3	6	Eugenio Leonardo Hidalgo Araya	C03	S04	315293	Viña del Mar
4	7	Evelyn Francesca Leiva Gajardo	C06	S03	287562	Iquique
5	9	Vanessa Teresa Montenegro Robles	C06	S02	559817	Iquique
6	8	Andrea Mena Barrientos	C05	S02	843623	La Serena
7	10	Carol Muñoz Belmar	C08	S01	735440	Temuco

Inner significa que se combinan las filas cundo la llave está presente en ambos dataframes.

KIBERNUM

Merge

En este caso, es más conveniente aplicar un "**left merge**", puesto que así no desaparecería registros de clientes que no hicieron match con la comuna.

```
# left merge
pd.merge(df1,df2, how='left', on='COD_COMUNA')
```

	ID_CLI	NOMBRE	COD_COMUNA	COD_SEGMENTO	SALDO	COMUNA
0	1	Cecilia Del Carmen Ayala Cabrera	C20	S01	1803344	NaN
1	2	Jesús Ignacio Contreras Vivar	C20	S01	236489	NaN
2	3	Carolina Andrea Estay Pangue	C01	S03	664647	Santiago
3	4	Jorge Eduardo García Lagos	C10	S02	1279353	Antofagasta
4	5	Carolina Lissett Gómez Morales	C02	S03	255036	Rancagua
5	6	Eugenio Leonardo Hidalgo Araya	C03	S04	315293	Viña del Mar
6	7	Evelyn Francesca Leiva Gajardo	C06	S03	287562	Iquique
7	8	Andrea Mena Barrientos	C05	S02	843623	La Serena
8	9	Vanessa Teresa Montenegro Robles	C06	S02	559817	Iquique
9	10	Carol Muñoz Belmar	C08	S01	735440	Temuco

En este caso, los registros del dataframe derecho son completados con NaN, pero no se pierde el registro de cliente.

Merge

Esto es lo que ocurriría con un “**right merge**”.

```
# rigth merge
pd.merge(df1,df2, how='right', on='COD_COMUNA')
```

	ID_CLI	NOMBRE	COD_COMUNA	COD_SEGMENTO	SALDO	COMUNA
0	3.0	Carolina Andrea Estay Pangue	C01	S03	664647.0	Santiago
1	4.0	Jorge Eduardo García Lagos	C10	S02	1279353.0	Antofagasta
2	5.0	Carolina Lissett Gómez Morales	C02	S03	255036.0	Rancagua
3	6.0	Eugenio Leonardo Hidalgo Araya	C03	S04	315293.0	Viña del Mar
4	7.0	Evelyn Francesca Leiva Gajardo	C06	S03	287562.0	Iquique
5	9.0	Vanessa Teresa Montenegro Robles	C06	S02	559817.0	Iquique
6	8.0	Andrea Mena Barrientos	C05	S02	843623.0	La Serena
7	10.0	Carol Muñoz Belmar	C08	S01	735440.0	Temuco
8	NaN	NaN	C04	NaN	NaN	Concepción
9	NaN	NaN	C07	NaN	NaN	Puerto Montt
10	NaN	NaN	C09	NaN	NaN	Punta Arenas
11	NaN	NaN	C11	NaN	NaN	Villarrica
12	NaN	NaN	C12	NaN	NaN	Los Angeles

KIBERNUM

Merge

Esto es lo que ocurriría con un “**outer merge**”.

```
# outer merge
pd.merge(df1,df2, how='outer', on='COD_COMUNA')
```

	ID_CLI	NOMBRE	COD_COMUNA	COD_SEGMENTO	SALDO	COMUNA
0	1.0	Cecilia Del Carmen Ayala Cabrera	C20	S01	1803344.0	NaN
1	2.0	Jesús Ignacio Contreras Vivar	C20	S01	236489.0	NaN
2	3.0	Carolina Andrea Estay Pangue	C01	S03	664647.0	Santiago
3	4.0	Jorge Eduardo García Lagos	C10	S02	1279353.0	Antofagasta
4	5.0	Carolina Lissett Gómez Morales	C02	S03	255036.0	Rancagua
5	6.0	Eugenio Leonardo Hidalgo Araya	C03	S04	315293.0	Viña del Mar
6	7.0	Evelyn Francesca Leiva Gajardo	C06	S03	287562.0	Iquique
7	9.0	Vanessa Teresa Montenegro Robles	C06	S02	559817.0	Iquique
8	8.0	Andrea Mena Barrientos	C05	S02	843623.0	La Serena
9	10.0	Carol Muñoz Belmar	C08	S01	735440.0	Temuco
10	NaN	NaN	C04	NaN	NaN	Concepción
11	NaN	NaN	C07	NaN	NaN	Puerto Montt
12	NaN	NaN	C09	NaN	NaN	Punta Arenas
13	NaN	NaN	C11	NaN	NaN	Villarrica
14	NaN	NaN	C12	NaN	NaN	Los Angeles

Merge

Esto es lo que ocurriría con un “**left merge**”.

```
pd.merge(df1,df3, how='left', on=['COD_SEGMENTO','COD_COMUNA'])
```

	ID_CLI	NOMBRE	COD_COMUNA	COD_SEGMENTO	SALDO	Factor Riesgo
0	1	Cecilia Del Carmen Ayala Cabrera	C20	S01	1803344	NaN
1	2	Jesús Ignacio Contreras Vivar	C20	S01	236489	NaN
2	3	Carolina Andrea Estay Pangue	C01	S03	664647	NaN
3	4	Jorge Eduardo García Lagos	C10	S02	1279353	0.10
4	5	Carolina Lissett Gómez Morales	C02	S03	255036	0.25
5	6	Eugenio Leonardo Hidalgo Araya	C03	S04	315293	NaN
6	7	Evelyn Francesca Leiva Gajardo	C06	S03	287562	NaN
7	8	Andrea Mena Barrientos	C05	S02	843623	0.30
8	9	Vanessa Teresa Montenegro Robles	C06	S02	559817	NaN
9	10	Carol Muñoz Belmar	C08	S01	735440	NaN

En estos casos, la doble llave no hizo match.

Dudas y consultas

KIBERNUM

Fin de la presentación

KIBERNUM