Conceptos Básicos de la fusión de datos

Edwin John Fredy Reyes Aguirre

2025-03-10

Contenido

Aprende a fusionar datos dispares mediante uniones internas. Combinando información de múltiples fuentes, descubrirás perspectivas convincentes que antes podían estar ocultas. También aprenderás cómo la relación entre esas fuentes, de uno a uno o de uno a muchos, puede afectar a tu resultado.

Unión interna

Qué columna elegiremos para fusionar?

Chicago proporciona una lista de propietarios de taxis y vehículos con licencia para operar en la ciudad, por seguridad pública. Tu objetivo es unir dos tablas. Una tabla se llama `taxi_owners` y contiene información sobre los propietarios de las empresas de taxis, mientras que la otra se llama `taxi_vehe` incluye información sobre cada vehículo de taxi.

```
import pandas as pd
```

```
taxi_owners = pd.read_pickle('../datasets/taxi_owners.p')
taxi_owners.head()
```

	rid	vid	owner	address	zip
0	T6285	6285	AGEAN TAXI LLC	4536 N. ELSTON AVE.	60630
1	T4862	4862	MANGIB CORP.	5717 N. WASHTENAW AVE.	60659
2	T1495	1495	FUNRIDE, INC.	3351 W. ADDISON ST.	60618
3	T4231	4231	ALQUSH CORP.	6611 N. CAMPBELL AVE.	60645
4	T5971	5971	EUNIFFORD INC.	3351 W. ADDISON ST.	60618

```
taxi_veh = pd.read_pickle('../datasets/taxi_vehicles.p')
taxi_veh.head()
```

	vid	make	model	year	fuel_type	owner
0	2767	TOYOTA	CAMRY	2013	HYBRID	SEYED M. BADRI
1	1411	TOYOTA	RAV4	2017	HYBRID	DESZY CORP.
2	6500	NISSAN	SENTRA	2019	GASOLINE	AGAPH CAB CORP
3	2746	TOYOTA	CAMRY	2013	HYBRID	MIDWEST CAB CO, INC
4	5922	TOYOTA	CAMRY	2013	HYBRID	SUMETTI CAB CO

Instrucciones:

Elige una columna que utilizarías para fusionar las dos tablas utilizando el método `.merge()`.

Respuestas posibles

- ☐ on='rid'
- ☒ on='vid'
- ☐ on='year'
- ☐ on='zip'

Tu primera unión interna

Te han encargado que averigües cuáles son los tipos de combustibles más utlizados en los taxis de Chicago. Para completar el análisis, tienes que fusionar las tablas `taxi_owners` y `taxi_veh` en la columna `vid`. A continuación, puedes utilizar la tabla combinada junto con el método `.values_counts()` para encontrar el `fuel_type` más común.

Instrucciones:

1. Fusiona `taxi_owners` con `taxi_veh` en la columna `vid` y guarda el resultado en `taxi_own_veh`.

```
# Merge the taxi_owners and taxi_veh tables
taxi_own_veh = taxi_owners.merge(taxi_veh, on='vid')
taxi_own_veh.head()

# Print the column names of taxi_own_veh
print(taxi_own_veh.columns)
```

```
Index(['rid', 'vid', 'owner_x', 'address', 'zip', 'make', 'model', 'year',
       'fuel_type', 'owner_y'],
      dtype='object')
```

2. Establece los sufijos izquierdo y derecho de la tabla para las columnas solapadas de la fusión en `_own` y `_veh`, respectivamente.

```
# Merge the taxi_owners and taxi_veh tables setting a suffix
taxi_own_veh = taxi_owners.merge(taxi_veh, on='vid', suffixes=('_own', '_veh'))
taxi_own_veh.head()

# Print the column names of taxi_own_veh
print(taxi_own_veh.columns)
```

```
Index(['rid', 'vid', 'owner_own', 'address', 'zip', 'make', 'model', 'year',
       'fuel_type', 'owner_veh'],
      dtype='object')
```

3. Selecciona la columna `fuel_type` de `taxi_own_veh` e imprime `value_counts()` para encontrar los `fuel_type` más utilizados.

```
# Merge the taxi_owners and taxi_veh tables setting a suffix
taxi_own_veh = taxi_owners.merge(taxi_veh, on='vid', suffixes=('_own', '_veh'))

# Print the value_counts to find the most popular fuel_type
print(taxi_own_veh['fuel_type'].value_counts())
```

```
fuel_type
HYBRID                     2792
GASOLINE                    611
FLEX FUEL                    89
COMPRESSED NATURAL GAS       27
Name: count, dtype: int64
```

Uniones internas y número de filas devueltas

Todas las fusiones que has estudiado hasta ahora se llaman uniones internas. Es necesario comprender que las uniones internas solo devuelven las filas con valores coincidentes en ambas tablas. Explorarás esto más a fondo revisando la fusión entre las tablas `wards` y `census`, y comparándola después con fusiones de copias de estas tablas ligeramente alteradas, denominadas `wards_altered` y `census_altered`. La primera fila de la columna `wards` se ha modificado en las tablas alteradas. Examinarás cómo afecta esto a la fusión entre ellos.

```
wards = pd.read_pickle('../datasets/ward.p')
wards.head()
```

	ward	alderman	address	zip
0	1	Proco "Joe" Moreno	2058 NORTH WESTERN AVENUE	60647
1	2	Brian Hopkins	1400 NORTH ASHLAND AVENUE	60622
2	3	Pat Dowell	5046 SOUTH STATE STREET	60609
3	4	William D. Burns	435 EAST 35TH STREET, 1ST FLOOR	60616
4	5	Leslie A. Hairston	2325 EAST 71ST STREET	60649

```
census = pd.read_pickle('../datasets/census.p')
census.head()
```

	ward	pop_2000	pop_2010	change	address	zip
0	1	52951	56149	6%	2765 WEST SAINT MARY STREET	6064
1	2	54361	55805	3%	WM WASTE MANAGEMENT 1500	6062
2	3	40385	53039	31%	17 EAST 38TH STREET	6065
3	4	51953	54589	5%	31ST ST HARBOR BUILDING LAKEFRONT TRAIL	6065
4	5	55302	51455	-7%	JACKSON PARK LAGOON SOUTH CORNELL DRIVE	6063

```
wards_altered = wards.copy()
wards_altered.loc[0, 'ward'] = 61
wards_altered.head()
```

	ward	alderman	address	zip
0	61	Proco ”Joe” Moreno	2058 NORTH WESTERN AVENUE	60647
1	2	Brian Hopkins	1400 NORTH ASHLAND AVENUE	60622
2	3	Pat Dowell	5046 SOUTH STATE STREET	60609
3	4	William D. Burns	435 EAST 35TH STREET, 1ST FLOOR	60616
4	5	Leslie A. Hairston	2325 EAST 71ST STREET	60649

```
census_altered = census.copy()
census_altered.loc[0, 'ward'] = None
census_altered.head()
```

	ward	pop_2000	pop_2010	change	address	zip
0	None	52951	56149	6%	2765 WEST SAINT MARY STREET	6064
1	2	54361	55805	3%	WM WASTE MANAGEMENT 1500	6062
2	3	40385	53039	31%	17 EAST 38TH STREET	6065
3	4	51953	54589	5%	31ST ST HARBOR BUILDING LAKEFRONT TRAIL	6065
4	5	55302	51455	-7%	JACKSON PARK LAGOON SOUTH CORNELL DRIVE	6063

Instrucciones:

1. Fusiona `wards` y `census` en la columna `ward` y guarda el resultado en `ward_census`.

```
# Merge the wards and census tables on the ward column
ward_census = wards.merge(census, on='ward')

# Print the shape of wards_census
print(f'ward_census table shape: {ward_census.shape}')
```

```
ward_census table shape: (50, 9)
```

2. Fusiona las tablas `merge_altered` y `census` en la columna `ward` y observa la diferencia en las filas devueltas.

```
# Print the first few rows of the wards_altered table to view the change
print(wards_altered[['ward']].head())

# Merge the wards_altered and census tables on the ward column
wards_altered_census =  wards_altered.merge(census, on='ward')
```

```
# Print the shape of wards_altered_census
print(f'wards_altered_census table shape: {wards_altered_census.shape}')
```

```
  ward
0   61
1    2
2    3
3    4
4    5
wards_altered_census table shape: (49, 9)
```

3. Fusiona las tablas `wards` y `census_altered` en la columna `ward` y observa la diferencia en las filas devueltas.

```
# Print the first few rows of the wards_altered table to view the change
print(census_altered[['ward']].head())

# Merge the wards_altered and census tables on the ward column
wards_altered_census =  wards.merge(census_altered, on='ward')

# Print the shape of wards_altered_census
print(f'wards_altered_census table shape: {wards_altered_census.shape}')
```

```
   ward
0  None
1     2
2     3
3     4
4     5
wards_altered_census table shape: (49, 9)
```

En el paso 1, el `.merge()` devolvió una tabla con el mismo número de filas que la tabla original `wards`. Sin embargo, en los pasos 2 y 3, al usar las tablas alteradas con la primera fila alterada de la columna `ward`, el número de filas devueltas fue menor. No había un valor coincidente en la columna `ward` de la otra tabla. *Recuerda que `.merge()` solo devuelve filas donde los valores coinciden en ambas tablas.*

Relaciones de uno a muchos