Ejercicios de introducción a Pandas

Dado que muchos usuarios potenciales de pandas tienen cierta familiaridad con `SQL`, estos ejercicios pretenden proporcionar algunos ejemplos de cómo se realizarían varias operaciones de SQL usando pandas.

In [1]:
```
import pandas as pd
pd.__version__
```

Out[1]:
```
'0.25.1'
```

Paso inicial: cargar el dataset **tips** (`'../data/teoria/tips.csv'`) y mostrar sus 5 primeras filas:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [3]:
```
pd.options.display.max_rows = 10
tips = pd.read_csv('../data/teoria/tips.csv')
tips.head()
```

Out[3]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

```
SELECT total_bill, tip, smoker, time
FROM tips
LIMIT 7;
```

In [7]:
```
# tips[['total_bill', 'tip', 'smoker', 'time']].head(7)
# tips[['total_bill', 'tip', 'smoker', 'time']][:7]
# tips.loc[:7, ['total_bill', 'tip', 'smoker', 'time']]
tips.iloc[:7, [0, 1, 3, 5]]
```

Out[7]:

	total_bill	tip	smoker	time
0	16.99	1.01	No	Dinner
1	10.34	1.66	No	Dinner
2	21.01	3.50	No	Dinner
3	23.68	3.31	No	Dinner
4	24.59	3.61	No	Dinner
5	25.29	4.71	No	Dinner
6	8.77	2.00	No	Dinner

```
SELECT *
FROM tips
WHERE time = 'Dinner'
LIMIT 5;
```

In [9]:
```
# tips.where(tips.time == 'Dinner').head()
# tips.query("time == 'Dinner'").head()
# tips[lambda x : x.time == 'Dinner'].head()
is_dinner = tips['time'] == 'Dinner'
tips[is_dinner].head()
```

Out[9]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

```
SELECT count(*)
FROM tips
WHERE time = 'Dinner';
```

In [17]:
```
# tips.where(tips.time == 'Dinner')['time'].count()
# tips.where(tips.time == 'Dinner').time.count()
# tips[is_dinner].shape[0]
# tips[is_dinner].tip.count()
# len(tips[is_dinner])
is_dinner.value_counts()[True]
```

Out[17]: 176

```
SELECT *
FROM tips
WHERE time = 'Dinner' AND tip > 5.00;
```

In [18]:
```
# tips.where((tips.time == 'Dinner') & (tips.tip > 5.00)).dropna()
# tips.query("time == 'Dinner' & tip > 5.00")
tip_is_greater5 = tips['tip'] > 5.00
tips[is_dinner & tip_is_greater5]
```

Out[18]:

	total_bill	tip	sex	smoker	day	time	size
23	39.42	7.58	Male	No	Sat	Dinner	4
44	30.40	5.60	Male	No	Sun	Dinner	4
47	32.40	6.00	Male	No	Sun	Dinner	4
52	34.81	5.20	Female	No	Sun	Dinner	4
59	48.27	6.73	Male	No	Sat	Dinner	4
...	...	...	...	...	...	...	...
183	23.17	6.50	Male	Yes	Sun	Dinner	4
211	25.89	5.16	Male	Yes	Sat	Dinner	4
212	48.33	9.00	Male	No	Sat	Dinner	4
214	28.17	6.50	Female	Yes	Sat	Dinner	3
239	29.03	5.92	Male	No	Sat	Dinner	3

15 rows × 7 columns

```
SELECT sex, count(*)
FROM tips
GROUP BY sex;
```

In [21]:
```
# tips.groupby('sex').tip.count()
# tips.groupby('sex').size()
tips.sex.value_counts()
```

Out[21]:
```
sex
Female     87
Male      157
Name: tip, dtype: int64
```

```
SELECT day, AVG(tip), COUNT(tip)
FROM tips
GROUP BY day;
```

In [28]:
```
# tips.groupby('day').tip.mean()
# tips.groupby('day').tip.agg(['mean','max'])
# tips.groupby('day').agg({'tip': ['mean','size','max']})
# tips.groupby('day').agg({'tip': ['mean','size']}).max()
tips.groupby('day').agg({'tip': ['mean','size']})
```

Out[28]:

	tip	
	mean	size
day		
Fri	2.734737	19
Sat	2.993103	87
Sun	3.255132	76
Thur	2.771452	62

```
SELECT day, AVG(tip), COUNT(tip), MAX(total_bill), MIN(total_bill)
FROM tips
GROUP BY day;
```

In [27]:
```
tips.groupby('day').agg({'tip': ['mean','size'], 'total_bill': ['mi
n', 'max']})
```

Out[27]:

	tip		total_bill	
	mean	size	min	max
day				
Fri	2.734737	19	5.75	40.17
Sat	2.993103	87	3.07	50.81
Sun	3.255132	76	7.25	48.17
Thur	2.771452	62	7.51	43.11

```sql
SELECT smoker, day, COUNT(tip), AVG(tip)
FROM tips
GROUP BY smoker, day;
```

In [24]:
```
#tips.groupby(['smoker', 'day']).tip.agg(['size', 'mean'])
tips.groupby(['smoker', 'day']).agg({'tip': ['size', 'mean']})
```

Out[24]:

		tip	
		size	mean
smoker	day		
No	Fri	4	2.812500
	Sat	45	3.102889
	Sun	57	3.167895
	Thur	45	2.673778
Yes	Fri	15	2.714000
	Sat	42	2.875476
	Sun	19	3.516842
	Thur	17	3.030000

Sean los siguientes DataFrame:

In [29]:
```
from numpy.random import randn
df1 = pd.DataFrame({'key': ['A', 'B', 'C', 'D'],
                    'value': randn(4)})

df2 = pd.DataFrame({'key': ['B', 'D', 'D', 'E'],
                    'value': randn(4)})
```

```sql
SELECT *
FROM df1
INNER JOIN df2
  ON df1.key = df2.key;
```

In [30]:
```
#df1.merge(df2, on='key')
pd.merge(df1, df2, on='key', suffixes=['_df1', '_df2'])
```

Out[30]:

	key	value_df1	value_df2
0	B	-0.380054	-0.347772
1	D	-0.615095	-0.605540
2	D	-0.615095	-0.662378

```
SELECT *
FROM df1
LEFT OUTER JOIN df2
  ON df1.key = df2.key;
```

In [53]:
```
#df1.merge(df2, on='key', how='left')
pd.merge(df1, df2, on='key', how='left')
```

Out[53]:

	key	value_x	value_y
0	A	-1.881358	NaN
1	B	-0.042679	-0.508523
2	C	-1.225004	NaN
3	D	0.529961	2.033017
4	D	0.529961	-0.098683

```
SELECT *
FROM df1
RIGHT OUTER JOIN df2
  ON df1.key = df2.key;
```

In [54]:
```
#df1.merge(df2, on='key', how='right')
pd.merge(df1, df2, on='key', how='right')
```

Out[54]:

	key	value_x	value_y
0	B	-0.042679	-0.508523
1	D	0.529961	2.033017
2	D	0.529961	-0.098683
3	E	NaN	1.057076

```
SELECT *
FROM df1
FULL OUTER JOIN df2
  ON df1.key = df2.key;
```

In [55]:
```
#df1.merge(df2, on='key', how='outer')
pd.merge(df1, df2, on='key', how='outer')
```

Out[55]:

	key	value_x	value_y
0	A	-1.881358	NaN
1	B	-0.042679	-0.508523
2	C	-1.225004	NaN
3	D	0.529961	2.033017
4	D	0.529961	-0.098683
5	E	NaN	1.057076

Sean los siguientes DataFrame:

In [31]:
```
df1 = pd.DataFrame({'city': ['Chicago', 'San Francisco', 'New York City'],
                    'rank': range(1, 4)})

df2 = pd.DataFrame({'city': ['Chicago', 'Boston', 'Los Angeles'],
                    'rank': [1, 4, 5]})
```

```
SELECT city, rank
FROM df1
UNION ALL
SELECT city, rank
FROM df2;
```

In [57]:
```
pd.concat([df1, df2])
```

Out[57]:

	city	rank
0	Chicago	1
1	San Francisco	2
2	New York City	3
0	Chicago	1
1	Boston	4
2	Los Angeles	5

```sql
SELECT city, rank
FROM df1
UNION
SELECT city, rank
FROM df2;
```

In [33]:
```
pd.concat([df1, df2]).drop_duplicates()
```

Out[33]:

	city	rank
0	Chicago	1
1	San Francisco	2
2	New York City	3
1	Boston	4
2	Los Angeles	5

```sql
UPDATE tips
SET tip = tip*2
WHERE tip < 2;
```

In [54]:
```
# tips.loc[tips['tip'] < 2, 'tip'] *= 2
# tips[tips['tip'] < 2].tip *= 2
tips.tip.where(tips.tip >= 2, tips.tip*2, inplace=True)
```

Out[54]:
```
0      2.02
1      3.32
2      3.50
3      3.31
4      3.61
       ...
239    5.92
240    2.00
241    2.00
242    3.50
243    3.00
Name: tip, Length: 244, dtype: float64
```

```sql
DELETE FROM tips
WHERE tip > 9;
```

In [60]:
```
# tips = tips.loc[tips.tip <= 9]
tips.where(tips.tip <= 9).dropna(inplace=True)
tips
```

Out[60]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns
