

Parcialito Pandas

```
In [17]: import pandas as pd
import numpy as np

In [21]: event_log = pd.read_csv('...path/event_log.csv')
events_types = pd.read_csv('..path/event_types.csv')
```

Parte a

Top 5 de Consumers que han tenido la mayor cantidad de eventos que resultaron en un event_status de ERROR.

```
In [56]: #primero filtro aquellos que tengan event_status ERROR, me quedo con la columna que me interesa para mergear despues
eventos_error = event_log.loc[event_log['event_status'] == 'ERROR', ['event_id','event_type_id']]

In [23]: #ahora hago un merge con events_type que tiene info de los consumidores, puede ser inner o left en este caso
merged_df = eventos_error.merge(events_types,how = 'inner', on = 'event_type_id')

In [28]: #creo una nueva columna y cuenta la cantidad de eventos para cada event_consumer_id
#me quedo con una columna que la utilizo para contar
merged_df['total_events'] = merged_df.groupby('event_consumer_id')['event_type_name'].transform('count')

In [ ]: resultado_A = merged_df.loc[:, ['event_consumer_id','total_events']]
#hago un drop_duplicates pq sino con el n_largest puede que me muestre un solo event_consumer_id repetido
resultado_A.drop_duplicates('event_consumer_id').nlargest(5,'total_events')
```

Parte B

De los eventos ocurridos para el country_id: BR indicar la cantidad de eventos totales por cada evento ocurridos por event_consumer_target.

```
In [37]: #primero filtro para el country_id pedido
eventos_BR = event_log.loc[event_log['country_id'] == 'BR', ['event_type_id','event_status']]

In [38]: #mergeo con el otro df, puede ser tanto inner como left en este caso

df_merged = eventos_BR.merge(events_types, how = 'inner', on = 'event_type_id')

In [39]: pseudo_resultado_b = df_merged.loc[:, ['event_type_id','event_status','event_consumer_target']]
#event_type_id la voy a usar para contar cuando haga pivot_table

In [40]: resultado_b = pseudo_resultado_b.pivot_table(index = 'event_status', columns = 'event_consumer_target',\
values = 'event_type_id', aggfunc = 'count')

In [ ]: resultado_b.rename_axis(None, axis=1,inplace= True)
resultado_b.reset_index().rename(columns = {'event_status':' '})
```