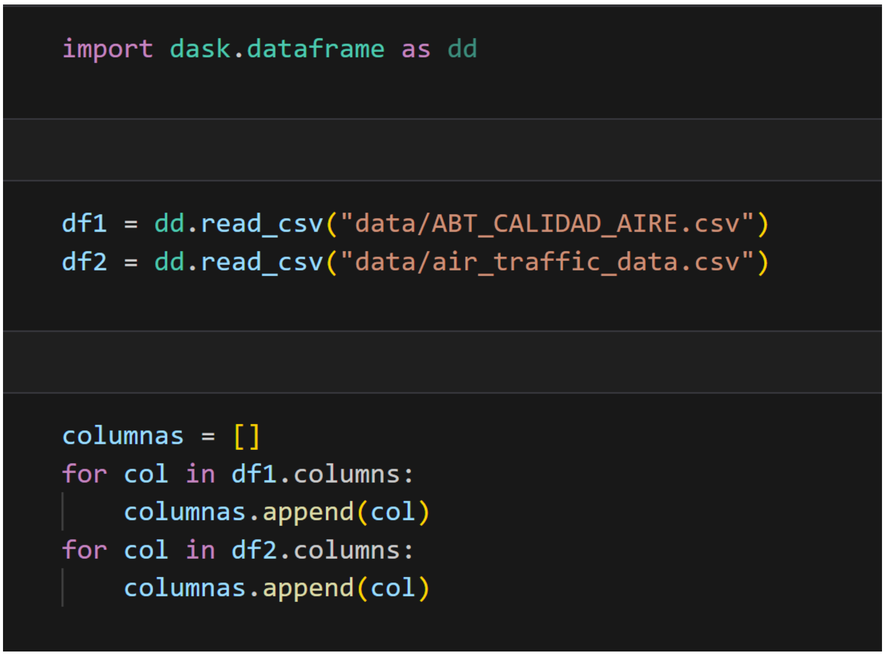
Código Fuente:

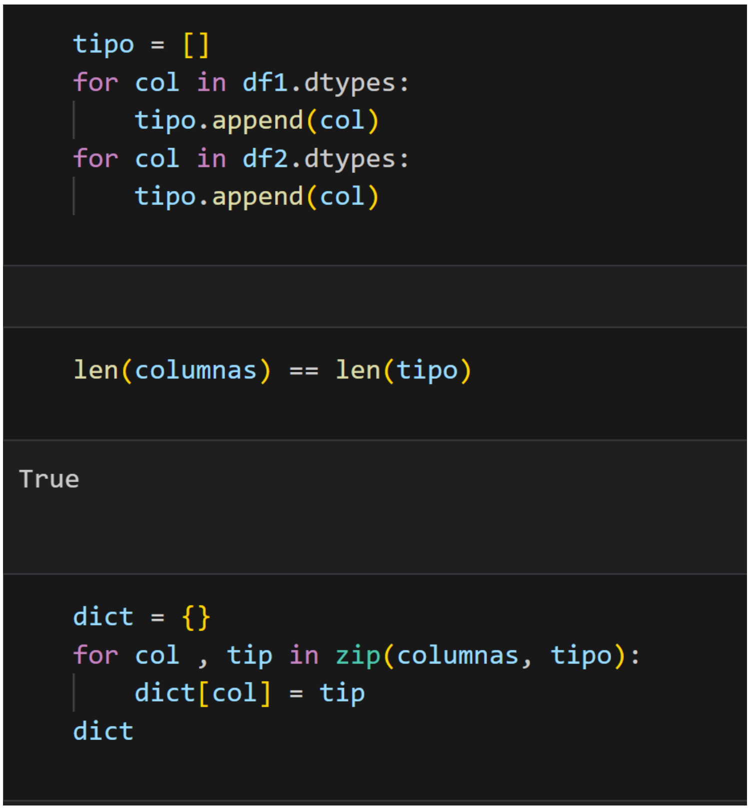
Desarrollo 1:



```python
import dask.dataframe as dd
```

```python
df1 = dd.read_csv("data/ABT_CALIDAD_AIRE.csv")
df2 = dd.read_csv("data/air_traffic_data.csv")
```

```python
columnas = []
for col in df1.columns:
    columnas.append(col)
for col in df2.columns:
    columnas.append(col)
```

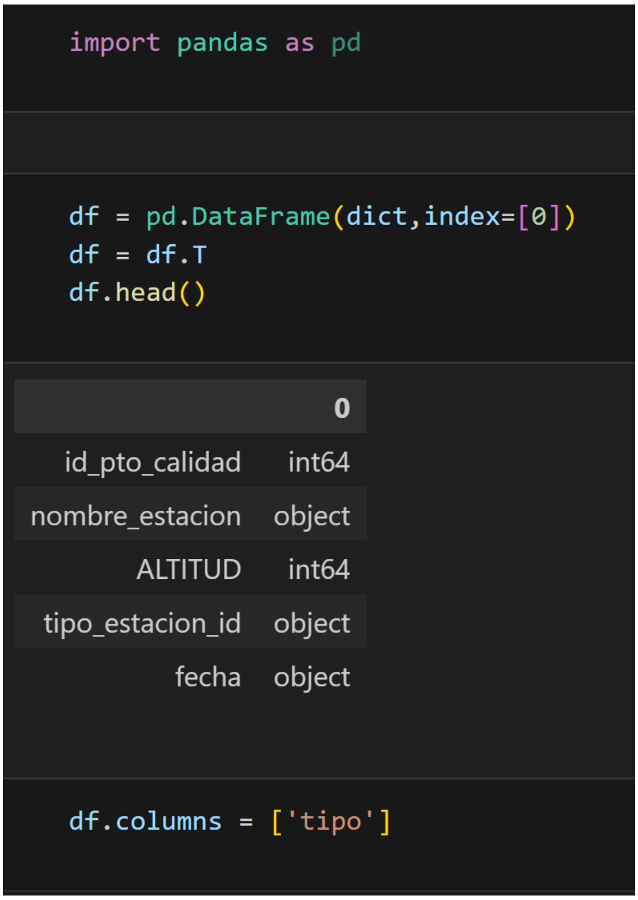


```python
tipo = []
for col in df1.dtypes:
    tipo.append(col)
for col in df2.dtypes:
    tipo.append(col)
```

```python
len(columnas) == len(tipo)
```

```
True
```

```python
dict = {}
for col , tip in zip(columnas, tipo):
    dict[col] = tip
dict
```

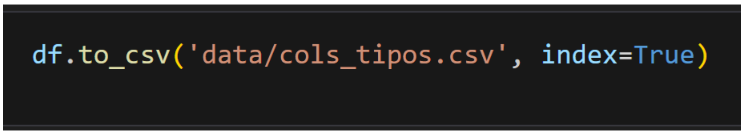


```
import pandas as pd
```

```
df = pd.DataFrame(dict,index=[0])
df = df.T
df.head()
```

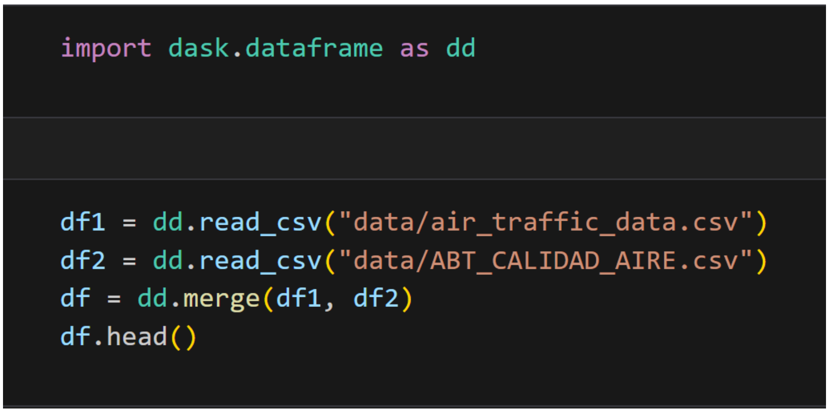
```
                        0
   id_pto_calidad   int64
nombre_estacion    object
         ALTITUD    int64
 tipo_estacion_id  object
           fecha   object
```

```
df.columns = ['tipo']
```



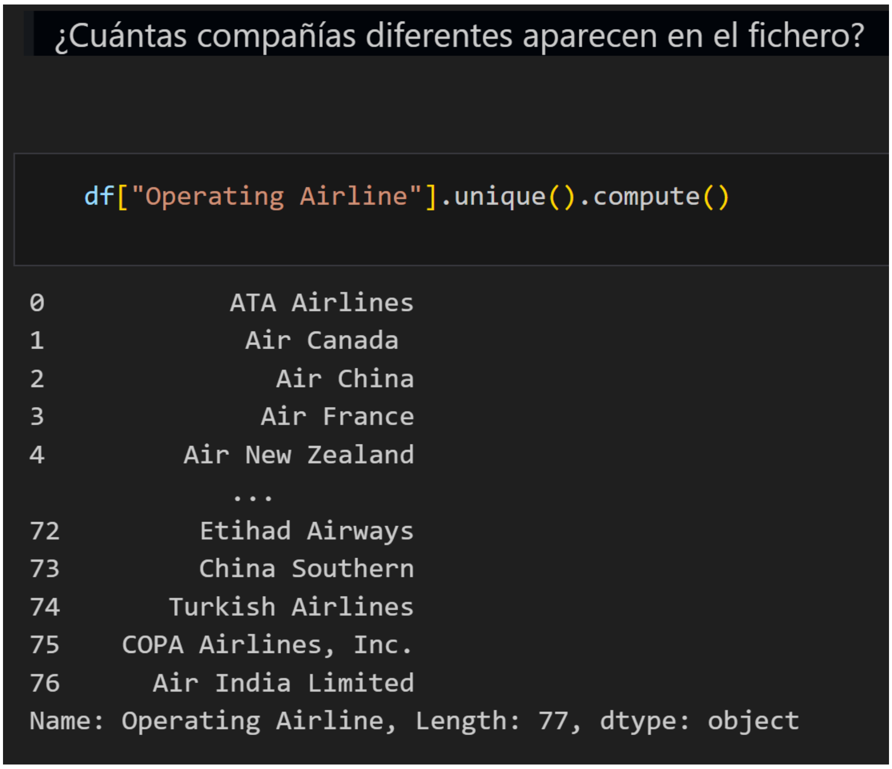
```
df.to_csv('data/cols_tipos.csv', index=True)
```

Desarrollo 2:



```
import dask.dataframe as dd
```

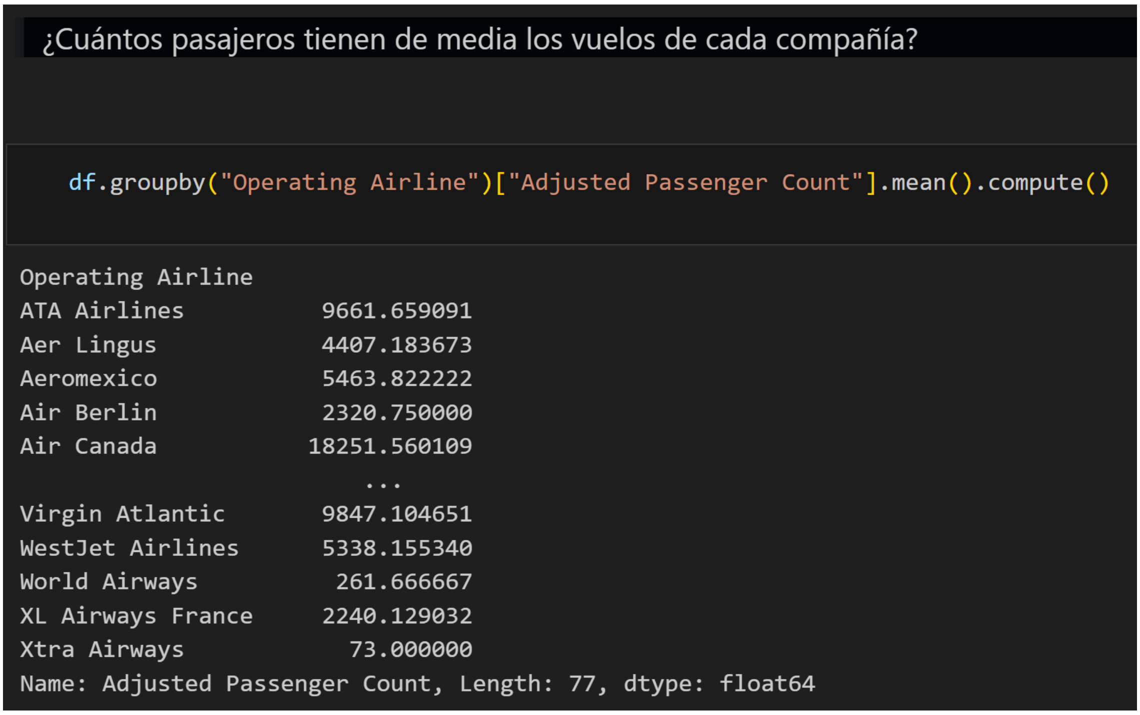
```
df1 = dd.read_csv("data/air_traffic_data.csv")
df2 = dd.read_csv("data/ABT_CALIDAD_AIRE.csv")
df = dd.merge(df1, df2)
df.head()
```



¿Cuántas compañías diferentes aparecen en el fichero?

```
df["Operating Airline"].unique().compute()
```

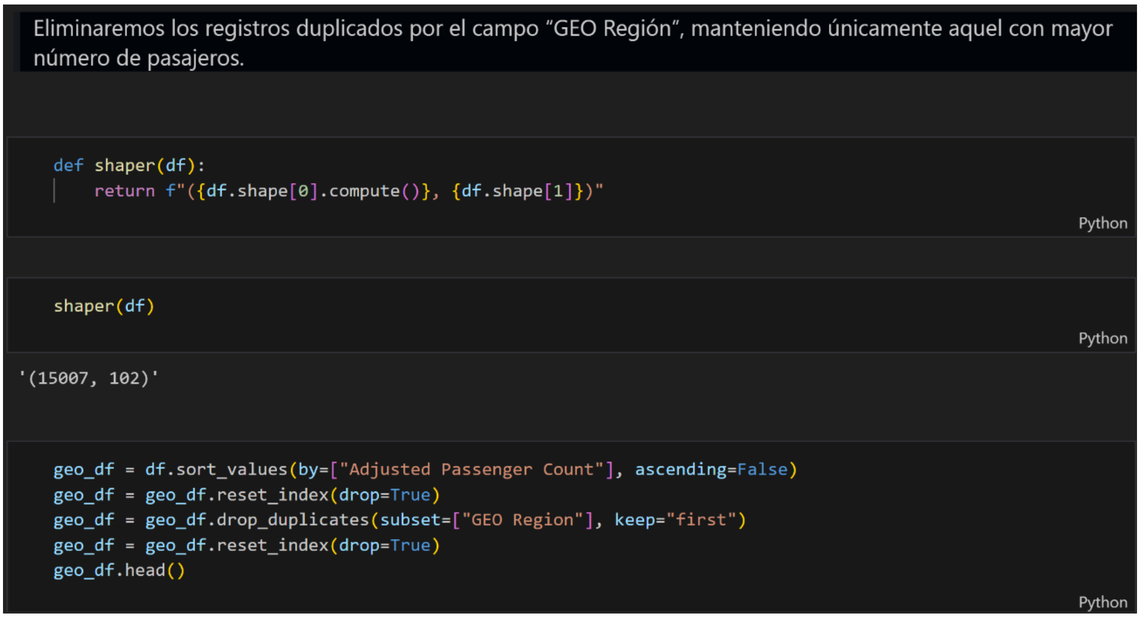
```
0             ATA Airlines
1              Air Canada
2                Air China
3               Air France
4          Air New Zealand
             ...
72          Etihad Airways
73          China Southern
74        Turkish Airlines
75    COPA Airlines, Inc.
76       Air India Limited
Name: Operating Airline, Length: 77, dtype: object
```



¿Cuántos pasajeros tienen de media los vuelos de cada compañía?

```python
df.groupby("Operating Airline")["Adjusted Passenger Count"].mean().compute()
```

```
Operating Airline
ATA Airlines           9661.659091
Aer Lingus             4407.183673
Aeromexico             5463.822222
Air Berlin             2320.750000
Air Canada            18251.560109
                          ...
Virgin Atlantic        9847.104651
WestJet Airlines       5338.155340
World Airways           261.666667
XL Airways France      2240.129032
Xtra Airways             73.000000
Name: Adjusted Passenger Count, Length: 77, dtype: float64
```



Eliminaremos los registros duplicados por el campo “GEO Región”, manteniendo únicamente aquel con mayor número de pasajeros.

```python
def shaper(df):
    return f"({df.shape[0].compute()}, {df.shape[1]})"
```

Python

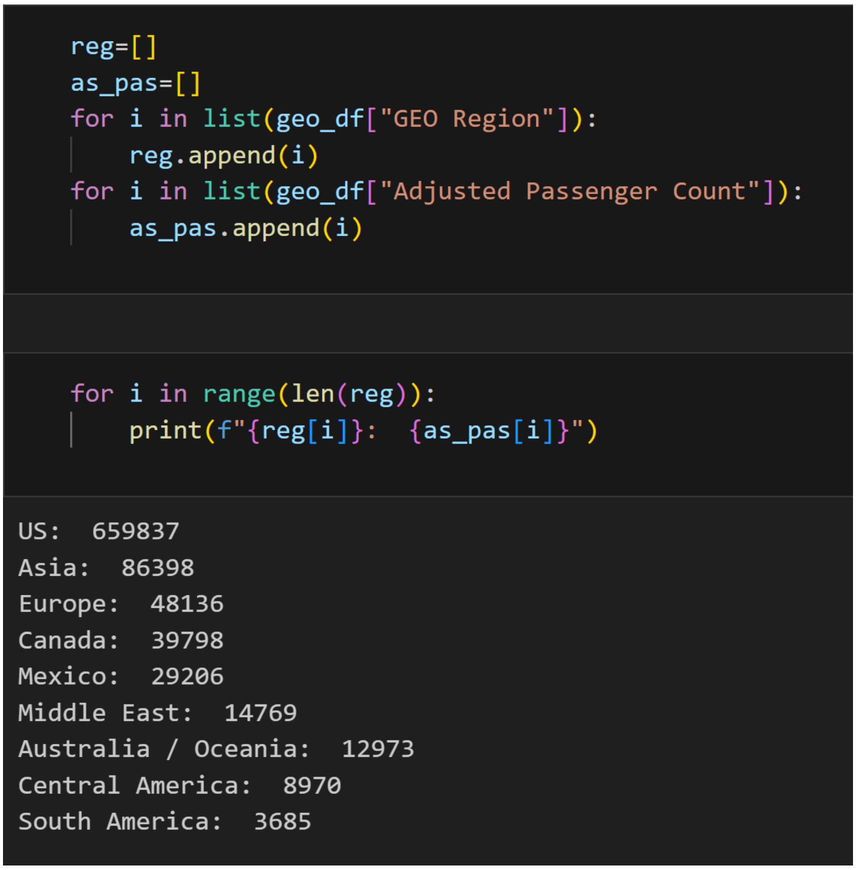
```python
shaper(df)
```

Python

```
'(15007, 102)'
```

```python
geo_df = df.sort_values(by=["Adjusted Passenger Count"], ascending=False)
geo_df = geo_df.reset_index(drop=True)
geo_df = geo_df.drop_duplicates(subset=["GEO Region"], keep="first")
geo_df = geo_df.reset_index(drop=True)
geo_df.head()
```

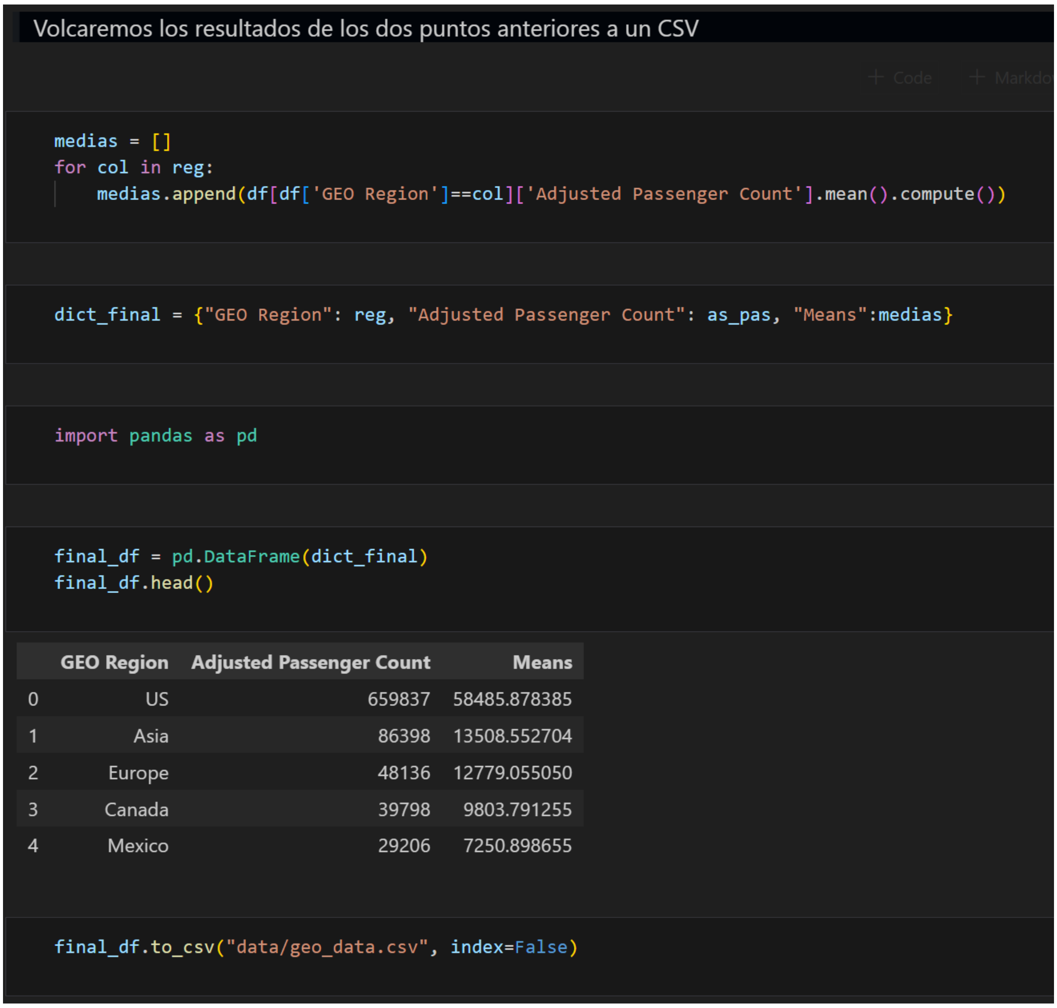
Python



```
reg=[]
as_pas=[]
for i in list(geo_df["GEO Region"]):
    reg.append(i)
for i in list(geo_df["Adjusted Passenger Count"]):
    as_pas.append(i)
```

```
for i in range(len(reg)):
    print(f"{reg[i]}:  {as_pas[i]}")
```

```
US:  659837
Asia:  86398
Europe:  48136
Canada:  39798
Mexico:  29206
Middle East:  14769
Australia / Oceania:  12973
Central America:  8970
South America:  3685
```



Volcaremos los resultados de los dos puntos anteriores a un CSV

```
medias = []
for col in reg:
    medias.append(df[df['GEO Region']==col]['Adjusted Passenger Count'].mean().compute())
```

```
dict_final = {"GEO Region": reg, "Adjusted Passenger Count": as_pas, "Means":medias}
```

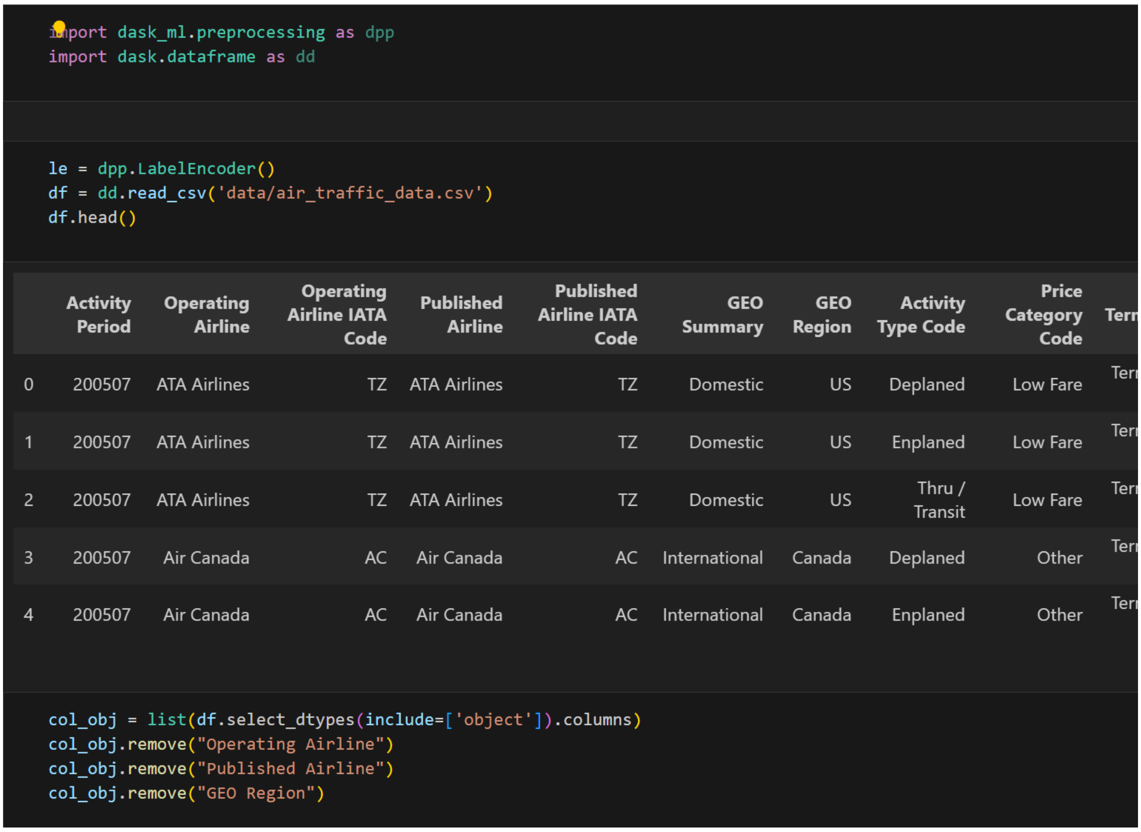
```
import pandas as pd
```

```
final_df = pd.DataFrame(dict_final)
final_df.head()
```

	GEO Region	Adjusted Passenger Count	Means
0	US	659837	58485.878385
1	Asia	86398	13508.552704
2	Europe	48136	12779.055050
3	Canada	39798	9803.791255
4	Mexico	29206	7250.898655

```
final_df.to_csv("data/geo_data.csv", index=False)
```

Desarrollo 3:

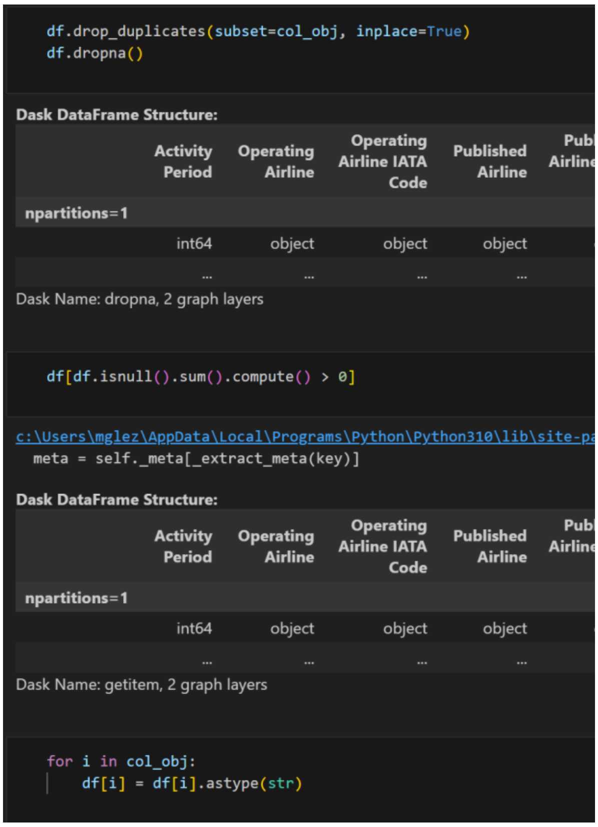


```
import dask_ml.preprocessing as dpp
import dask.dataframe as dd
```

```
le = dpp.LabelEncoder()
df = dd.read_csv('data/air_traffic_data.csv')
df.head()
```

	Activity Period	Operating Airline	Operating Airline IATA Code	Published Airline	Published Airline IATA Code	GEO Summary	GEO Region	Activity Type Code	Price Category Code	Terr
0	200507	ATA Airlines	TZ	ATA Airlines	TZ	Domestic	US	Deplaned	Low Fare	Ter
1	200507	ATA Airlines	TZ	ATA Airlines	TZ	Domestic	US	Enplaned	Low Fare	Ter
2	200507	ATA Airlines	TZ	ATA Airlines	TZ	Domestic	US	Thru / Transit	Low Fare	Ter
3	200507	Air Canada	AC	Air Canada	AC	International	Canada	Deplaned	Other	Ter
4	200507	Air Canada	AC	Air Canada	AC	International	Canada	Enplaned	Other	Ter

```
col_obj = list(df.select_dtypes(include=['object']).columns)
col_obj.remove("Operating Airline")
col_obj.remove("Published Airline")
col_obj.remove("GEO Region")
```



```
df.drop_duplicates(subset=col_obj, inplace=True)
df.dropna()
```

Dask DataFrame Structure:

	Activity Period	Operating Airline	Operating Airline IATA Code	Published Airline	Publ Airline
npartitions=1					
	int64	object	object	object	
	...	...	...	...	

Dask Name: dropna, 2 graph layers

```
df[df.isnull().sum().compute() > 0]
```

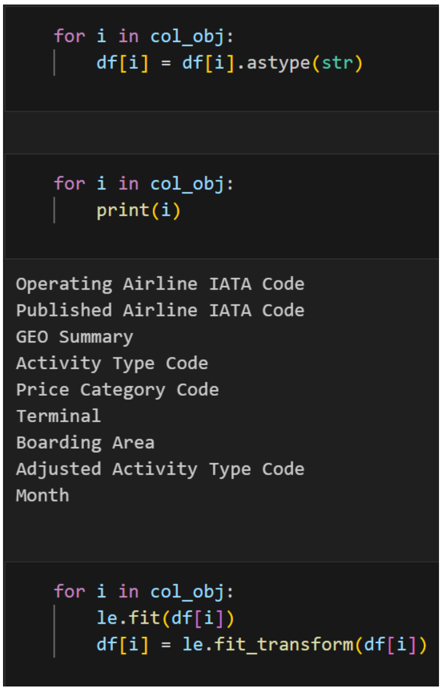
```
c:\Users\mglez\AppData\Local\Programs\Python\Python310\lib\site-pa
  meta = self._meta[_extract_meta(key)]
```

Dask DataFrame Structure:

	Activity Period	Operating Airline	Operating Airline IATA Code	Published Airline	Publ Airline
npartitions=1					
	int64	object	object	object	
	...	...	...	...	

Dask Name: getitem, 2 graph layers

```
for i in col_obj:
    df[i] = df[i].astype(str)
```



```
for i in col_obj:
    df[i] = df[i].astype(str)
```

```
for i in col_obj:
    print(i)
```

```
Operating Airline IATA Code
Published Airline IATA Code
GEO Summary
Activity Type Code
Price Category Code
Terminal
Boarding Area
Adjusted Activity Type Code
Month
```

```
for i in col_obj:
    le.fit(df[i])
    df[i] = le.fit_transform(df[i])
```