Código primera parte

```
import dask.dataframe as dd
import dask.bag as db
import os

df = dd.read_csv(os.path.join('air_traffic_data1.csv')).compute()
df.head()
```

```
df.info()
```

```
#¿Cuántas compañías diferentes aparecen en el fichero?

a = df.groupby('Operating Airline IATA Code').count()
company =list(a.index)
print(f'Hay {len(company)} compañias')
```

```
#¿Cuántos pasajeros tienen de media los vuelos de cada compañía?
b = df.groupby('Operating Airline IATA Code')['Passenger Count'].mean()
print(b)
```

```
#Eliminaremos los registros duplicados por el campo "GEO Región", manteniendo únicamente aquel con mayor número de pasajeros.
df1 = df.copy()
c = df.groupby('GEO Region')['Passenger Count'].max()
l = list(c)
l1 = list(c.index)

borrar = []
for i in range (len(df['Passenger Count'])):
    g = l1.index(df['GEO Region'][i])
    if df['Passenger Count'][i] != l[g]:
      borrar.append(i)

df =df.drop(borrar)
```

```
df.reset_index(inplace = True, drop = True)
df.head()
```

```
#Volcaremos los resultados de los dos puntos anteriores a un CSV.

df.to_csv('Maximo_pasajeros.csv')
b.to_csv('Media_pasajeros.csv')
```

Código segunda parte:

```
import dask.dataframe as dd
import os
import warnings

warnings.filterwarnings('ignore')


df = dd.read_csv(os.path.join('air_traffic_data1.csv')).compute()
df.head()
```

```
df.info()
```

```
def val_unicos(dato):
  a = df[dato].unique()
  print(dato, df[dato].unique())
  return a
ba =val_unicos('Boarding Area')
pcc = val_unicos('Price Category Code')
atc =val_unicos('Activity Type Code')
t = val_unicos('Terminal')
gs = val_unicos('GEO Summary')
aatc = val_unicos('Adjusted Activity Type Code')
m = val_unicos('Month')
```

```
#Vamos a pasar algunos datos str a int
#price Category Code: 0 = low, 1 = other
#Geo Summary: 0 = Domestic, 1 = International
#Adjusted Activity Type Code: 0 = deplaned, 1 = Enplaned, 2 = Thru/Transit * 2
#Terminal: 0 = International, 1 =Terminal 1, 2 = Terminal 2, 3 = Terminal 3, 4 = Other

def cambio(a, b, c, d, e, dato):
  if dato != 'Month':
    for i in range(len(df)):
      if df[dato][i] == a:
        df[dato][i] = 0
      elif df[dato][i] == b:
        df[dato][i] = 1
      elif df[dato][i] == c:
        df[dato][i] = 2
      elif df[dato][i] == d:
        df[dato][i] = 3
      elif df[dato][i] == e:
        df[dato][i] = 4
  else:
    for i in range(len(df)):
      if df[dato][i] == 'January':
        df[dato][i] = 1
```

```
else:
  for i in range(len(df)):
    if df[dato][i] == 'January':
      df[dato][i] = 1
    elif df[dato][i] == 'February':
      df[dato][i] = 2
    elif df[dato][i] == 'March':
      df[dato][i] = 3
    elif df[dato][i] == 'April':
      df[dato][i] = 4
    elif df[dato][i] == 'May':
      df[dato][i] = 5
    elif df[dato][i] == 'June':
      df[dato][i] = 6
    elif df[dato][i] == 'July':
      df[dato][i] = 7
    elif df[dato][i] == 'August':
      df[dato][i] = 8
    elif df[dato][i] == 'September':
      df[dato][i] = 9
    elif df[dato][i] == 'October':
      df[dato][i] = 10
    elif df[dato][i] == 'November':
      df[dato][i] = 11
    elif df[dato][i] == 'December':
```

```
      elif df[dato][i] == 'August':
        df[dato][i] = 8
      elif df[dato][i] == 'September':
        df[dato][i] = 9
      elif df[dato][i] == 'October':
        df[dato][i] = 10
      elif df[dato][i] == 'November':
        df[dato][i] = 11
      elif df[dato][i] == 'December':
        df[dato][i] = 12
  df[dato] =df[dato].astype(int)
cambio(pcc[0], pcc[1], None, None,None, 'Price Category Code')
cambio(gs[0], gs[1],None, None, None, 'GEO Summary')
cambio(aatc[0], aatc[1], aatc[2], None, None,'Adjusted Activity Type Code')
cambio(t[0], t[1], t[2], t[3], t[4], 'Terminal')
cambio(None, None, None, None, None, 'Month')

print(df.head())
```

```
df.info()
```

```
#calcular la media y la desviación estándar
print(df.mean())
print(df.groupby('Boarding Area').mean())
print(df.groupby('GEO Region').mean())
print(df.groupby('Operating Airline').mean())
```

```
print(df.std())
print(df.groupby('Boarding Area').std())
print(df.groupby('GEO Region').std())
print(df.groupby('Operating Airline').std())
```

```
#análisis de la correlación cuyo resultado debe ser una matriz de correlación de datos que represente
#de qué manera están relacionadas las diferentes variables
df.corr()
```