MEMORIA

1. Business case & Data Collection

Datos

Los datos se encuentran en 3 archivos .csv flights.csv, hotels.csv y user.csv los cuales forman parte de viajes corporativos de 5 agencias de viajes en ciudades de Brasil.

Hipótesis

H1) Comportamiento de clientes:

-cuando prefieren viajar (dia, meses)

-en que clase prefieren viajar

-hacia que ciudad

-cuantos días

-analizar si los fines de semana y en temporada alta los ingresos aumentan

H2) Dependencia de Precios de Vuelos y Hoteles

Dependencia de Distancia y Precio de vuelo

2. Data Understanding

```
Importamos las librerias a utilizar
## PYTHON ##
import pandas as pd
import numpy as np
import datetime
import statistics
Para que me muestre todas las columnas
pd.options.display.max_colwidth = None  #con None es para todos los caracteres posibles

## VISUALIZACIONES ##
import matplotlib.pyplot as plt
import seaborn as sns
```

Exploratorio Inicial: para los 3 datasets realizamos los siguientes pasos

```
1°) Cargamos los 3 datasets con al siguiente función:
pd.read_csv(ruta)
2°) Analizamos como estan compuestos los datos y los tipos de variables
df.sample(5)
3°) De que tipo son los datos
df.info()
```

Tabla de Variables

```
1°) Analisis estadistico de las variables
df.describe(include="all")
2°) Descripción de las variables: de que tipo son, cuantitativas o cualitativas
3°) Analizamos desde que rango de fecha son los datos:
print(f'Inicio: {pd.to_datetime(df_flights["date"]).min()}')
print(f'Fin: {pd.to_datetime(df_flights["date"]).max()}')
4°) Observamos que variables tienen datos unicos:
.unique()
5°) Heatmap/Mapa de Correlación para saber de que variables podemos obtener relaciones
sns.heatmap(df.corr(), annot=True)
```

3. Data Cleaning

Missings/Valores nulos: observamos si hay valores nulos/Nan/None con los métodos:

```
df.isnull().sum()
df.isna().sum()
```

Outliers

Para saber si hay valores atipicos o anormales realizamos graficos Boxplots

- Histogramas: `sns.histplot(data=df, x='variable')`
- Densidad: `sns.distplot(df["variable"], hist = False);`
- Boxplots: `sns.boxplot(data=df[['variable1',..,'variable n']])`
- Barplot: `sns.barplot(data=df,x='variable',y='variable');`

Transformaciones

FLIGHTS: creamos las siguientes columnas

```
1°) Creamos columna year-month-day" para leer y manipular la fecha:
df_flights["year-month-day"] = pd.to_datetime(df_flights["date"])
2°)Para obtener el año y mes
df_flights['year-month'] = pd.to_datetime(df_flights['year-month-day']).dt.to_period('M')
3°) Para obtener el año
df_flights['year'] = pd.to_datetime(df_flights['year-month-day']).dt.year
4°) Para obtener el mes
df_flights['month'] = pd.to_datetime(df_flights['year-month-day']).dt.month
5°) Ordenamos el DF por fecha
df_flights.sort_values(by=['year-month-day'])
6°) Tomamos un rango de fechas para hacer el analisis: 2019-09-26 hasta 2021-09-26 ya que
para fechas posteriores no hay datos de los precios de vuelos
df_flights = df_flights.loc[(df_flights['year-month'] >= '2019-09') & (df_flights['year-
month'] <= '2021-09')]
7°) Ordenamos el DF por fecha: de menor a mayor
df_flights  = df_flights.sort_values(by=['year-month-day'])
8°) Creamos una columna con el nombre del dia de la semana
df_flights["day_of_week"] =df_flights["year-month-day"].dt.day_name()
```

HOTELS

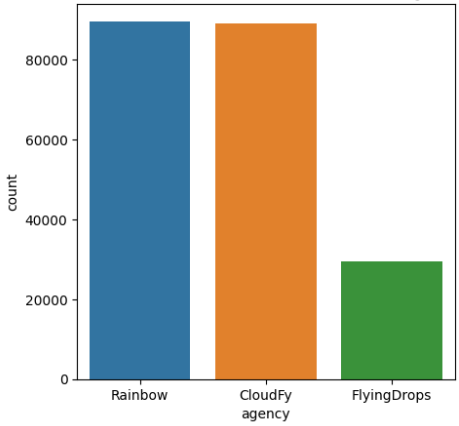
```
1°) Agregamos columna 'year-month-day' para leer y manipular la fecha:
df_hotels["year-month-day"] = pd.to_datetime(df_hotels["date"])
2°) Agregamos columna del el year-month
df_hotels['year-month'] = pd.to_datetime(df_hotels['year-month-day']).dt.to_period('M')
3°) Agregamos columna del el mes
df_hotels['month'] = pd.to_datetime(df_hotels['year-month-day']).dt.month
4°) Agregamos columna del año
df_hotels['year'] = pd.to_datetime(df_hotels['year-month-day']).dt.year
5°) Tomamos un rango de fechas para hacer el analisis: 2019-09-26 hasta 2021-09-26
df_hotels = df_hotels.loc[(df_hotels['year-month'] >= '2019-09') & (df_hotels['year-month']
<= '2021-09')]
6°) Ordenamos el DF por fecha: de menor a mayor
df_hotels  = df_hotels.sort_values(by=['year-month-day'])
```

USERS : no se agregaron columnas

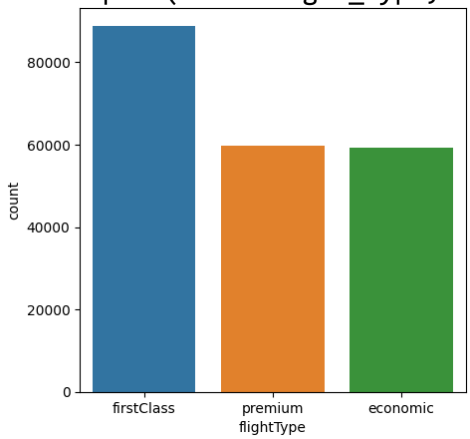
4. Analysis

Análisis Univariante

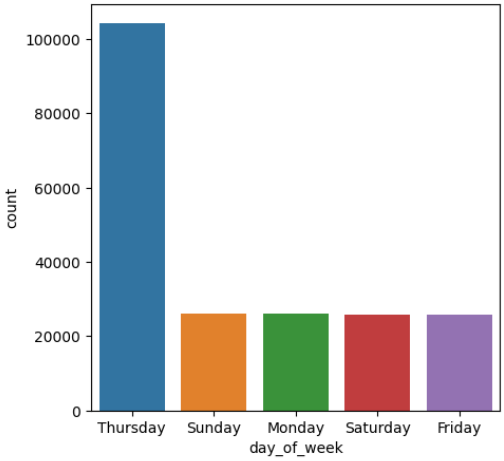
```
1°) Que aerolínea es la más demandada
vuelos_x_agencia =
pd.DataFrame(df_flights.groupby("agency").size(),columns=['count']).sort_values(by="count",as
cending=False)
sns.barplot(data=vuelos_x_agencia,x='agency',y='count');
```



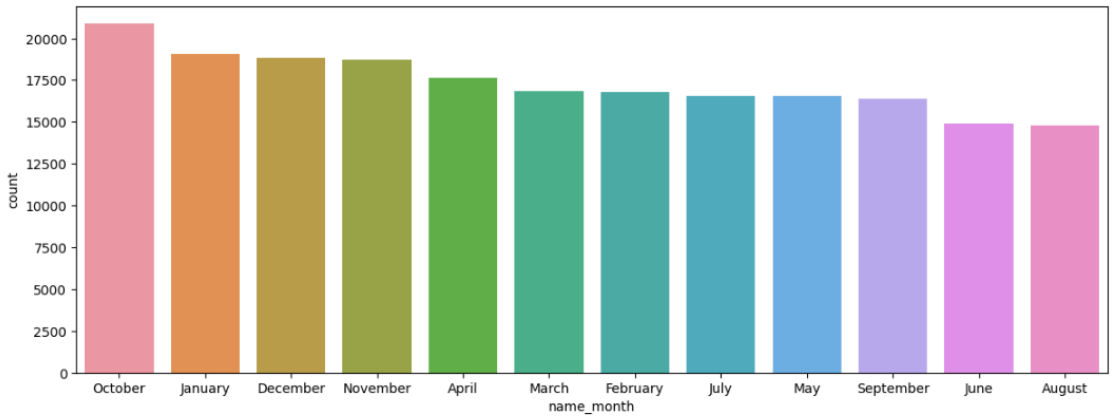

```
2°) Que flightType es el mas demandado
flight_type=pd.DataFrame(df_flights.groupby("flightType").size(),columns=['count']).sort_valu
es(by="count",ascending=False)
plt.figure(figsize=(5,5))
sns.barplot(data=flight_type,x='flightType',y='count');
```



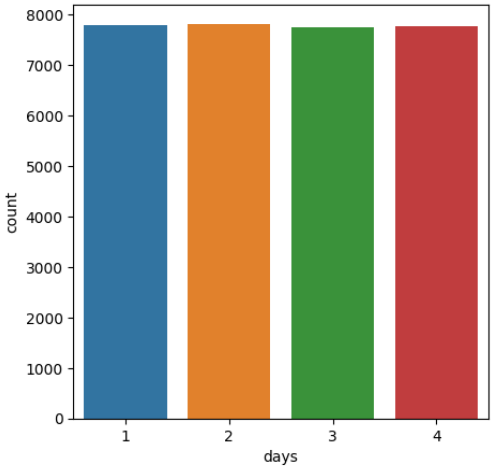

```
3°) Analizamos que dia de la semana es el mas elegido para viajar
day=pd.DataFrame(df_flights.groupby("day_of_week").size(),columns=['count']).sort_values(by="
count",ascending=False)
plt.figure(figsize=(5,5))
sns.barplot(data=day,x='day_of_week',y='count');
```




```
4°) Analizamos que meses son los más demandados
Creamos una columna month para obtener el mes
flights_copy = df_flights
flights_copy['month'] = pd.to_datetime(df_flights['year-month-day']).dt.month
flights_copy["name_month"] =df_flights["year-month-day"].dt.month_name()
months=pd.DataFrame(flights_copy.groupby("name_month").size(),columns=['count']).sort_values(
by="count",ascending=False)
plt.figure(figsize=(14,5))
sns.barplot(data=months,x='name_month',y='count');
```

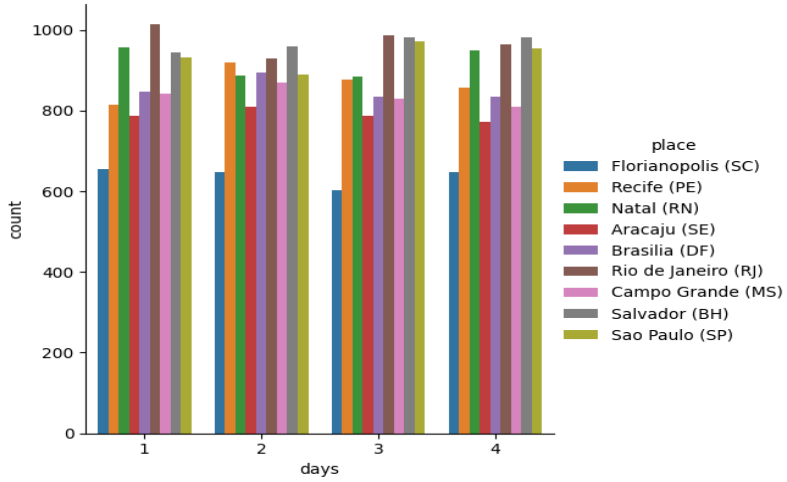



```
5°) Analizamos la cantidad de días que los clientes viajan para identificar posibles
preferencias
cant_dias=pd.DataFrame(df_hotels.groupby(['days']).size(),columns=['count']).sort_values(by="
count",ascending=False)
plt.figure(figsize=(5,5))
sns.barplot(data=cant_dias,x='days',y='count');
```



6°) Analizamos la cantidad de dias x Ciudad

```
df1 = pd.DataFrame(df_hotels.groupby(['name', 'place','days']).size())
plt.figure(figsize=(40,15));
sns.catplot(x="days",hue="place",kind='count',data=df_hotels,ci=None);
```

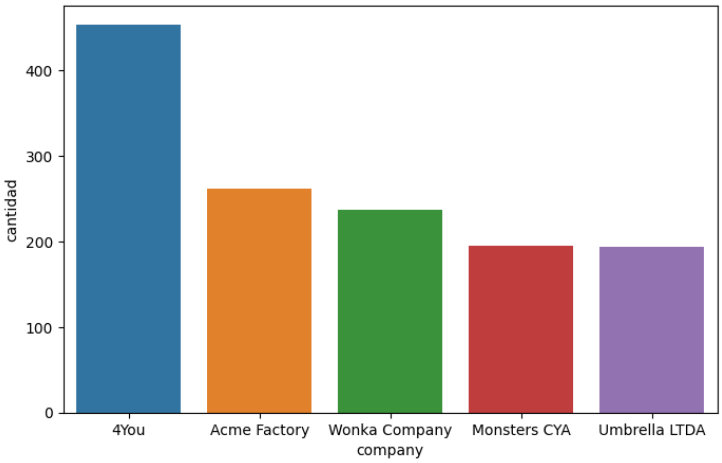


7°) Destinos más demandados (cada destino tiene un único hotel)

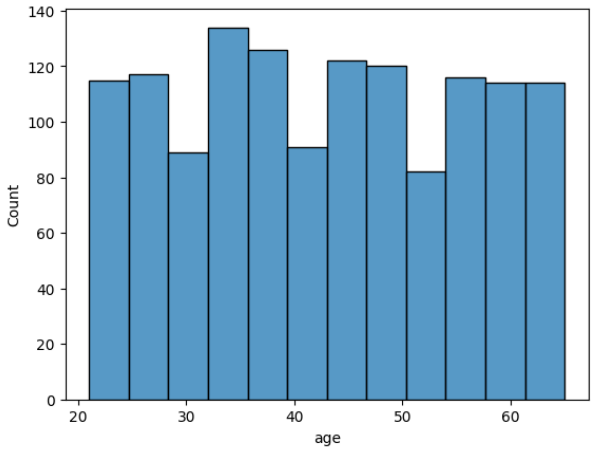
```
df_hotels.groupby(['name', 'place']).size()
# Graficamos Destinos más demandados
places=df_hotels['place'].value_counts()
fig, ax = plt.subplots(figsize=(15, 4))
ax.bar(x=places.index,height=places.values);
ax.set_title('Destinos mas demandados');
ax.set_ylabel('Cantidad')
ax.set_xlabel('Destinos')
```

8°) Company que mas clientes tiene

```
company=pd.DataFrame(df_users.groupby('company').size(),columns=['cantidad']).sort_values(by="cantidad",ascending=False)
plt.figure(figsize=(8,5))
sns.barplot(data=company,x='company',y='cantidad');
```

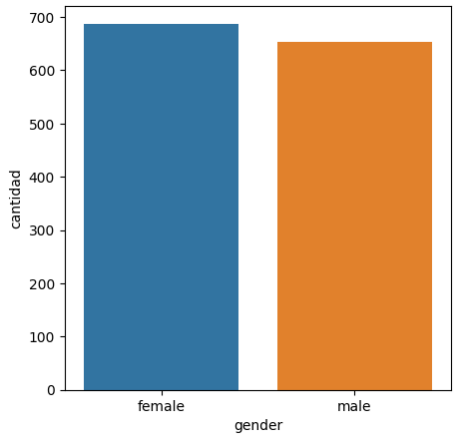


9°) Analizamos si hay una inclinación por cierto rango de edad para realizar los viajes



10°) Que sexo es el que mas viaja

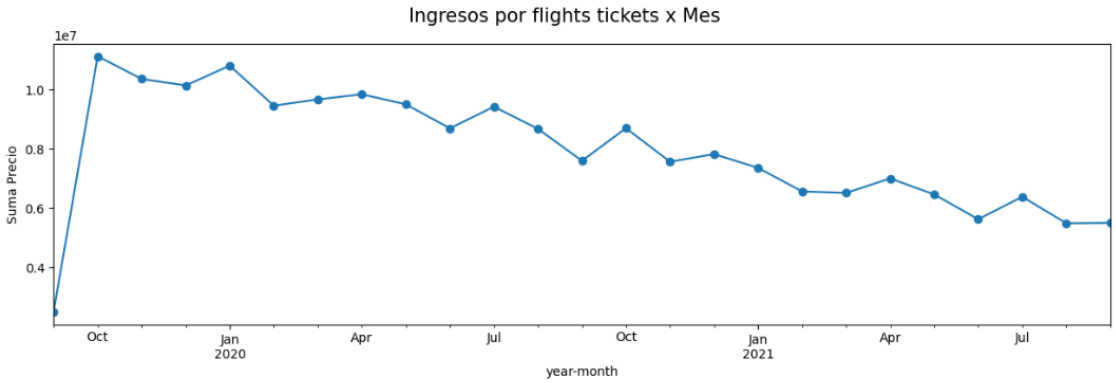
```
gender=pd.DataFrame(df_users.groupby('gender').size(),columns=['cantidad']).sort_values(by="cantidad",ascending=False)
plt.figure(figsize=(5,5))
sns.barplot(data=gender,x='gender',y='cantidad')
```



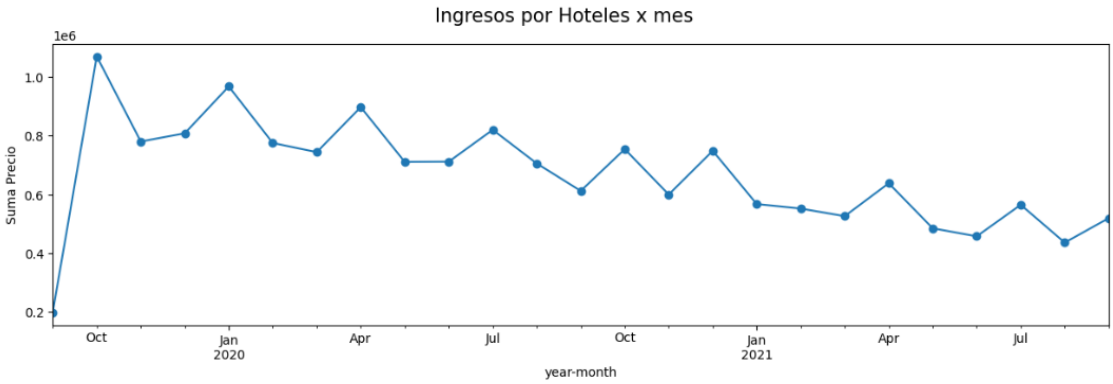
Análisis Bivariante

H2) 1. Relación de Ingresos por Vuelos-Ingresos por Hoteles

```
1°) Ingresos por Vuelos x mes
ingresos_x_vuelos = df_flights.groupby('year-month')["price"].sum()

2°) Graficamos los ingresos por los vuelos x mes
fig, ax = plt.subplots(figsize=(15, 4))
fig.suptitle('Ingresos por flights tickets x Mes', fontsize = '15')
ax.set(xlabel='year-month', ylabel='Suma Precio')
ingresos_x_vuelos.plot(kind='line',marker='o')
```




```
3°) Ingresos por mes de todos los hoteles
ingresos_hoteles = df_hotels.groupby('year-month')["total"].sum()

4°) Graficamos los ingresos por mes de todos los hoteles
fig, ax = plt.subplots(figsize=(15, 4))
fig.suptitle('Ingresos por Hoteles x mes', fontsize = '15')
ax.set(xlabel='year-month', ylabel='Suma Precio')
ingresos_hoteles.plot(kind='line',marker='o')
```



5°) Unimos los datasets Flight y Hotels para analizar como se relacionan los ingresos de cada uno con respecto al otro

6°) Filtramos por el rango de fecha que decidimos analizar

```
flights = df_flights.loc[(df_flights['year-month'] >= '2019-9') & (df_flights['year-month'] <= '2022-09')]
hotels = df_hotels.loc[(df_hotels['year-month'] >= '2019-9') & (df_hotels['year-month'] <= '2022-09')]
```

7°) Sumamos todos los vuelos x mes

```
flights = flights.groupby('year-month')['price'].sum()
flights
hotels = hotels.groupby('year-month')['price'].sum()
hotels
```

8°) Unimos por fecha

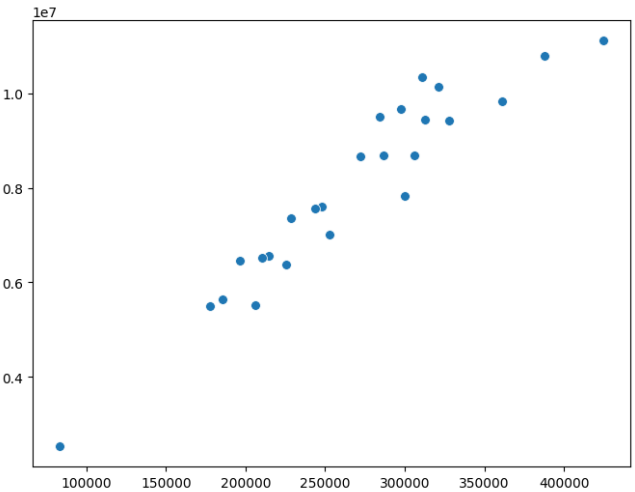
```
flights_hotels = pd.merge(flights,hotels, on=['year-month'], how='left')
flights_hotels.reset_index(inplace=True)
flights_hotels.rename(columns={'price_x':'flight_price','price_y':'hotels_price'})
flights_hotels
```

9°) Matriz de Correlacion para ver si son variables dependientes o no

```
flights_hotels.corr()
```

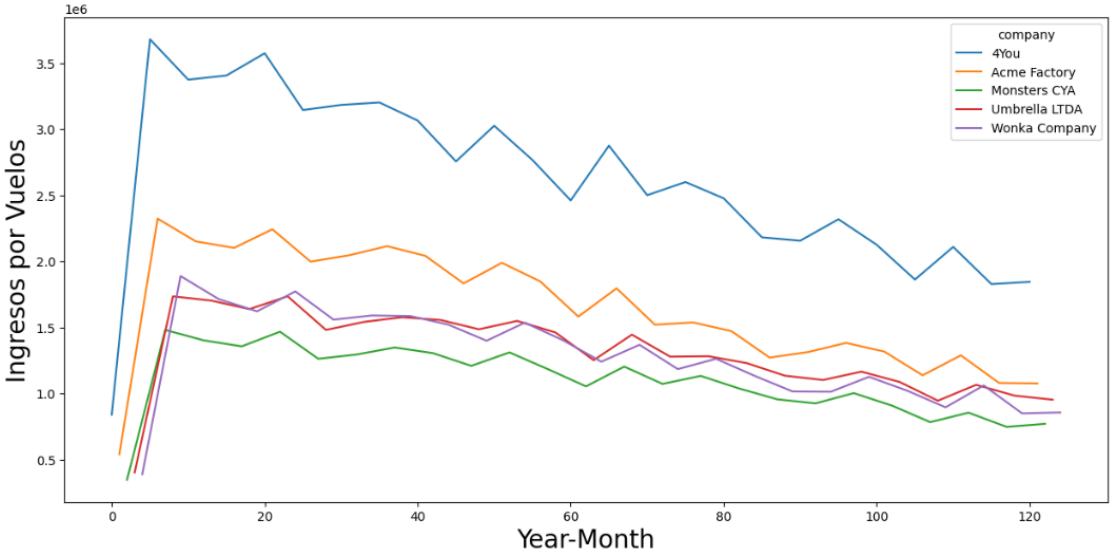
10°) Diagrama de dispersion

```
plt.figure(figsize=(8,6))
sns.scatterplot(x=hotels.values,y=flights.values,s=50);
```



Ingresos por Vuelos y Hoteles x compañía

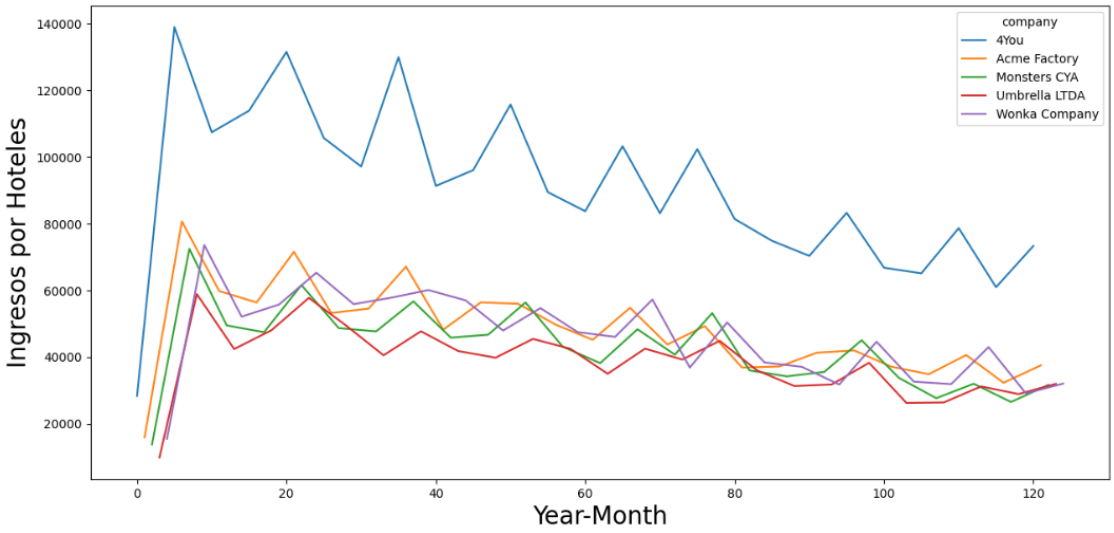
```
1°) Obtenemos de c dataframe las columnas que queremos
Flights
vuelos = df_flights[['userCode','price','year-month','year']]

Hotels
hoteles = df_hotels[['userCode','price','year-month','year']]

Users
usuarios = df_users[['code','company']]
Renombramos la columna user con userCode para poder hacer la union
usuarios = usuarios.rename(columns={'code': 'userCode'})

Unimos vuelos_usuarios
vuelos_usuarios = pd.merge(vuelos,usuarios, on=['userCode'], how='left')
vuelos_usuarios

Unimos hoteles_usuarios
hoteles_usuarios = pd.merge(hoteles,usuarios, on=['userCode'], how='left')
hoteles_usuarios

2°) Agrupamos por year-month y company los ingresos de vuelos
suma_precios_vuelos = pd.DataFrame(vuelos_usuarios.groupby(['year-
month','company','year'])['price'].sum())

3°) Graficamos Lineplot
fig, axs = plt.subplots(figsize=(15,8))
sns.lineplot(data=suma_precios_vuelos,x=suma_precios_vuelos.index,y='price',hue='company');
axs.set_xlabel("Year-Month", fontsize = 20)
axs.set_ylabel("Ingresos por Vuelos", fontsize = 20)
```




```
4°) Agrupamos por year-month y company los ingresos de hoteles
suma_precios_hoteles = pd.DataFrame(hoteles_usuarios.groupby(['year-
month','company'])['price'].sum())
suma_precios_hoteles
```

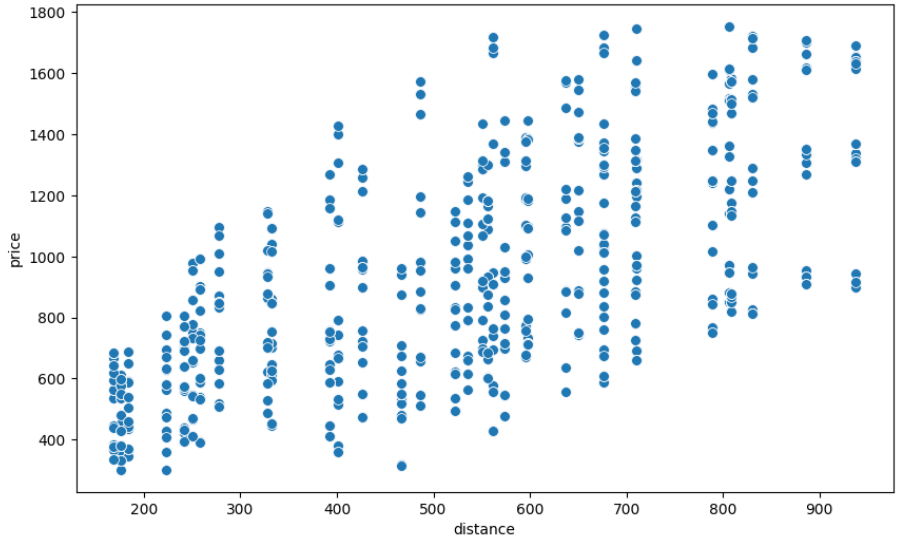
```
suma_precios_hoteles.reset_index(inplace=True)
suma_precios_hoteles
```

```
5°) Graficamos Lineplot
fig, axs = plt.subplots(figsize=(15,8))
sns.lineplot(data=suma_precios_hoteles,x=suma_precios_hoteles.index,y='price',hue='company');
axs.set_xlabel("Year-Month", fontsize = 20)
axs.set_ylabel("Ingresos por Hoteles", fontsize = 20)
```



2. Relacion de Precios de Vuelos-Distancia

```
1°) Filtramos por price y distance
price_distance = df_flights[['price','distance']]

2°)Coeficiente de Correlacion
price_distance.corr()

3°)Grafico Dispersion distance-price
plt.figure(figsize=(10,6))
sns.scatterplot(data=price_distance,x='distance',y='price',s=50);
```



5. Resultados

- En Brasil la temporada alta de turismo se centra en los meses de Julio-Diciembre y Enero, lo cuáles según el análisis obtenido coinciden con los más demandados por los clientes para viajar.
- Los jueves es el día de mayor demanda (50,15%). En general, de Lunes a Viernes los costos de vuelos son menores Viernes-Sábado-Domingos aumentan.
- Las empresas consideran que la calidad de los viajes aumenta la satisfacción y la productividad de los viajeros lo que implica mejores comodidades en las clases de vuelo, es por eso que eligen viajar en FirstClass (42,69%)
- Los mayores y más importantes centros de negocios de Brasil son las megaciudades de Sao Paulo y Río de Janeiro. Más allá, la mayoría de las capitales de los estados cumplen plenamente su papel de otros centros económicos principales del país: Belo Horizonte, Salvador, Recife, Fortaleza, Curitiba y Porto Alegre. Estas son las ciudades que dominan el desarrollo económico de Brasil.
- No se encontró preferencia por una determinada cantidad de días, por lo general los viajes tienen una duración de entre 1-4 días, lo cual puede deberse a que depende el tipo de viaje que realicen los clientes, ya sea para asistir a congresos, conferencias, capacitaciones o reuniones de negocios
- Aceptamos la hipótesis de la depencia de que el aumento/disminución en la compra de vuelos implica aumento/disminución en la reserva de hoteles.
- Aceptamos la hipótesis de que la distancia influye notablemente en el precio de los vuelos (proporcionales).

Como conclusión del EDA realizado, este datasets es muy particular y acotado ya que el análisis se realizó en un rango de fechas en la cual había covid y estaba presente la pandemia, lo cual nos muestra como el turismo corporativo se vio fuertemente afectado. Tras los decretos de cierre, varias empresas adoptaron modelos de home office y empezaron a realizar sus reuniones en línea, asi como también se prohibieron todo tipo de vuelos incluyendo los nacionales.
Un dato importante que se vio reflejado en el análisis fue que los datos de la Organización Mundial del Turismo (OMT) señalan que el turismo mundial cayó un 70% en los ocho primeros meses de 2020 respecto al año anterior.