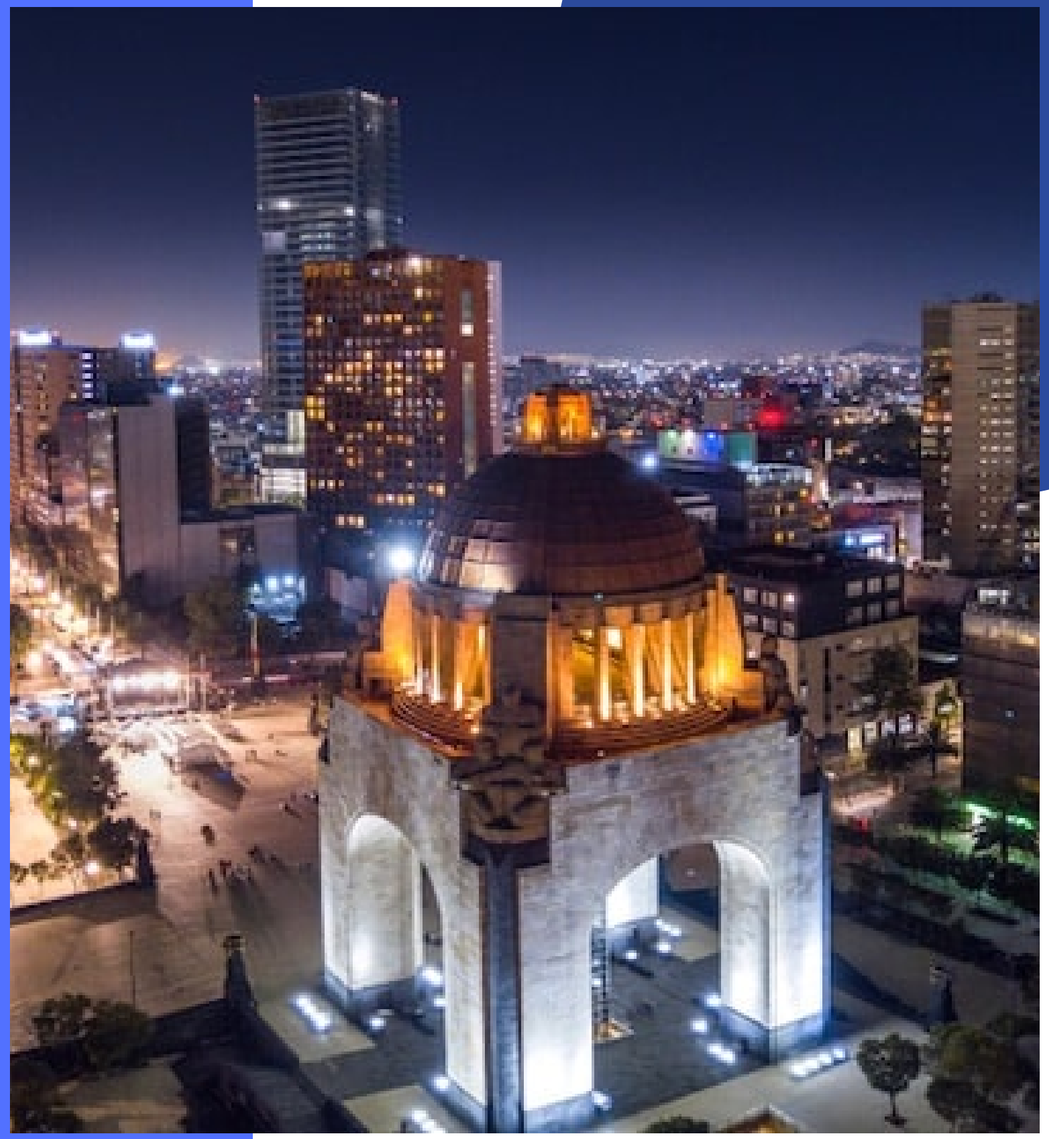
# ACT. 5
# REGRESIÓN LINEAL SIMPLE

ANDRÉS SALMERÓN GARCÍA
JULEN UGARTECHEA REPETTO
YAEL MOJICA PÉREZ

# INTRODUCCIÓN

AIRBNB EN CIUDAD DE MÉXICO
AIRBNB EN CALIFORNIA



# PROCEDIMIENTO

```python
#Importamos librerias requeridas
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```python
#carga archivo
from google.colab import files
files.upload()
```

```
Elegir archivos No se eligió ningún archivo Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving DF_Mexico.csv to DF_Mexico.csv
```

- NULOS
- OUTLIERS
- CORRELACIONES

- MODELO
- DETERMINACIÓN
- COEFICIENTES

- COMPARACIÓN CON CALIFORNIA

# NULOS

**Empieza el filtro de nulos para el grupo privado**

```
[ ] filtro_privado = df2[df2['room_type'] == 'Private room']
    filtro_privado.info
```

```
[ ] #Se corroboran los valores nulos de las columnas con valores cuantitativos
    valores_nulos = filtro_privado.isnull().sum()
    valores_nulos
```

```
[ ] #Primero se elimina el signo % para aplicar el remplazo de nulos
    #filtro_privado['host_acceptance_rate']=filtro_privado['host_acceptance_rate'].astype(str)
    filtro_privado['price']=filtro_privado['price'].str.replace('$', '')
    filtro_privado['price']=filtro_privado['price'].str.replace(',', '')
    filtro_privado['price']=filtro_privado['price'].str.replace('.', '')
    filtro_privado['host_acceptance_rate']=filtro_privado['host_acceptance_rate'].str.replace('%', '')
    filtro_privado['host_response_rate']=filtro_privado['host_response_rate'].str.replace('%', '')
```

```
[ ] #Pasamos los valores de la columna a tipo numerico
    filtro_privado['price']=filtro_privado['price'].astype(float)
    filtro_privado['host_acceptance_rate']=filtro_privado['host_acceptance_rate'].astype(float)
    filtro_privado['host_response_rate']=filtro_privado['host_response_rate'].astype(float)
    filtro_privado['number_of_reviews']=filtro_privado['number_of_reviews'].astype(float)
    filtro_privado['availability_365']=filtro_privado['availability_365'].astype(float)
    filtro_privado['number_of_reviews']=filtro_privado['number_of_reviews'].astype(float)
```

```
#Reemplazando nulos de columna host_acceptance_rate
#filtro_private['host_acceptance_rate'] = filtro_private['host_acceptance_rate'].str.rstrip('%').astype(float
mean_acceptance_rate = filtro_privado['host_acceptance_rate'].mean()
filtro_privado['host_acceptance_rate'].fillna(mean_acceptance_rate, inplace=True)
print('La media de la columna acceptance_rate es: ', mean_acceptance_rate)
```

```
] #Reemplazando nulos de columna host_acceptance_rate
#filtro_private['host_response_rate'] = filtro_private['host_response_rate'].str.rstrip('%').astype(float)
mean_response_rate = filtro_privado['host_response_rate'].mean()
filtro_privado['host_response_rate'].fillna(mean_response_rate, inplace=True)
print('La media de la columna response_rate es: ', mean_response_rate)
```

```
] valores_nulos = filtro_privado.isnull().sum()
valores_nulos
```

```
] #Reemplazando nulos de las columnas review_scores_location, review_scores_cleanliness,
#reviews_per_month y review_scores_communication
mean_review_scores_location = filtro_privado['review_scores_location'].mean()
filtro_privado['review_scores_location'].fillna(mean_review_scores_location, inplace=True)

mean_review_scores_cleanliness = filtro_privado['review_scores_cleanliness'].mean()
filtro_privado['review_scores_cleanliness'].fillna(mean_review_scores_cleanliness, inplace=True)

mean_reviews_per_month = filtro_privado['reviews_per_month'].mean()
filtro_privado['reviews_per_month'].fillna(mean_reviews_per_month, inplace=True)
```

# OUTLIERS

**Empieza eliminación de outliers para el grupo compartido**

```
[199] #Realizamos diagrama de caja o bigote de cada columna dek dataframe
      fig = plt.figure(figsize = (15, 8))
      filtro_compartido.plot(kind='box', vert=False)
      plt.title('Valores atipicos del dataframe')
      plt.show() #Dibujamos el histograma
```

```
[200] #Metodo aplicando desviación estandar. Encuentro los valores extremos
      y=filtro_compartido
      Limite_Superior_iqr= y.mean() + 3*y.std()
      Limite_Inferior_iqr= y.mean() - 3*y.std()
      print('Limite superior permitido', Limite_Superior_iqr)
      print('Limite inferior permitido', Limite_Inferior_iqr)
```

```
[201] #Encontramos outliers del DataFrame
      outliers=filtro_compartido[(y>Limite_Superior_iqr)|(y<Limite_Inferior_iqr)]
      outliers
```

**Aqui se convierten los outliers en nulos**

```
[202] #Obtenemos datos y los outliers se convierten en nulos en el dataframe
      data3=filtro_compartido[(y<=Limite_Superior_iqr)&(y>=Limite_Inferior_iqr)]
      data3
```

```
[203] #Corroboramos valores nulos
      valores_nulos=data2.isnull().sum()
      valores_nulos
```

```
[204] #Reemplazamos valores atipicos (nulos) del dataframe con 'mean'
      #Realizamos una copia del dataframe
      data_clean=data2.copy()
      data_clean=data_clean.fillna(round(data2.mean(), 1))
      data_clean
```

```
[205] #Corroboramos valores nulos
      valores_nulos=data_clean.isnull().sum()
      valores_nulos
```

```
[206] #Unimos la columna 'DF_2020' con el dataframe
      Datos_limpios=pd.concat([data_clean, filtro_compartido], axis=1)
      Datos_limpios
```

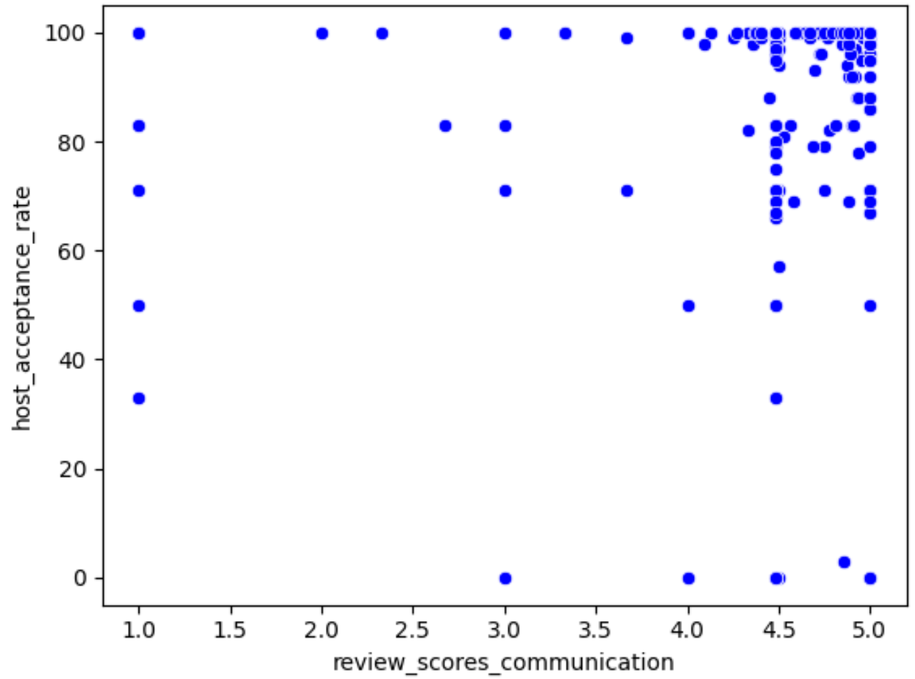
```
[207] #Realizamos diagrama de caja o bigote de cada columna dek dataframe
      fig = plt.figure(figsize = (15, 8))
      filtro_compartido.plot(kind='box', vert=False)
      plt.title('Valores atipicos del dataframe')
      plt.show() #Dibujamos el histograma
```

# CORRELACIONES

```
[139] import seaborn as sns
      from turtle import color

      sns.scatterplot(x = 'review_scores_communication', y = 'host_acceptance_rate', color = 'blue', data = filtro_compartido)
```

```
<Axes: xlabel='review_scores_communication', ylabel='host_acceptance_rate'>
```




```
[ ]  #Declaramos las variables dependientes e independientes
     Vars_Indep = filtro_compartido[['review_scores_communication']]
     Var_Dep = filtro_compartido['host_acceptance_rate']
```

# MODELO

```
[ ]  #Se define model como la función de regresión lineal
     from sklearn.linear_model import LinearRegression
     model = LinearRegression()
```

```
[ ]  #Verificamos la función relacionada al modelo
     type(model)
```

```
sklearn.linear_model._base.LinearRegression
```

```
[ ]  #Ajustamos el modelo con los variables antes declaradas
     model.fit(X = Vars_Indep, y = Var_Dep)
```

```
▾ LinearRegression
LinearRegression()
```

# DETERMINACIÓN

```
[144] #Verificamos los coeficientes obtenidos para el modelo ajustado
      model.__dict__
```

```
[145] filtro_compartido.info()
```

```
[146] #Predecimos los valores
      y_pred=model.predict(X=filtro_compartido[['review_scores_communication']])
      y_pred
```

```
      #Visualizamos la grafica comparativa entre el total real y el total predecido

      sns.scatterplot(x='review_scores_communication', y='host_acceptance_rate', color='blue', data=filtro_compartido)
      sns.scatterplot(x='review_scores_communication', y='review_scores_communication', color='red', data=filtro_compartido)
      sns.lineplot(x='review_scores_communication', y='review_scores_communication', color='red', data=filtro_compartido)
```

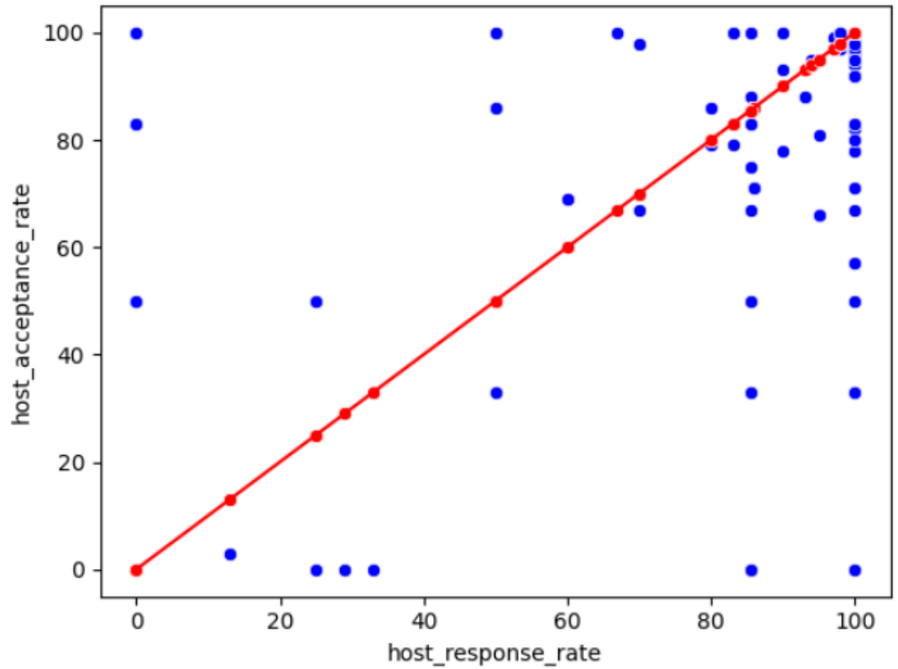
```
[148] #Corroboramos cual es el coeficiente de Determinación de nuestro modelo
      coef_Deter=model.score(X=Vars_Indep, y=Var_Dep)
      coef_Deter
```

# COEFICIENTES

```
#Visualizamos la grafica comparativa entre el total real y el total predecido

sns.scatterplot(x='host_response_rate', y='host_acceptance_rate', color='blue', data=filtro_compartido)
sns.scatterplot(x='host_response_rate', y='host_response_rate', color='red', data=filtro_compartido)
sns.lineplot(x='host_response_rate', y='host_response_rate', color='red', data=filtro_compartido)
```

```
<Axes: xlabel='host_response_rate', ylabel='host_acceptance_rate'>
```




```
#Corroboramos cual es el coeficiente de Determinación de nuestro modelo
coef_Deter=model.score(X=Vars_Indep, y=Var_Dep)
coef_Deter
```

```
0.18955749550448142
```

# COMPARACIÓN

| | Base de datos de DF | Base de datos de California |
|---|---|---|
| Tipo compartido | 0.158% | 0.102% |
| | 0.118% | 1.019% |
| | 0.043% | 0.019% |
| | 18.956% | 34.533% |
| | 0.093% | 6.307% |
| | 1.646% | 0.002% |
| Tipo privado | 0.030% | 0.211% |
| | 0.103% | 0.039% |
| | 0.036% | 0.491% |
| | 14.115% | 2.485% |
| | 0.468% | 0.966% |
| | 1.417% | 2.556% |

FIN