UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
SECCIONAL TUNJA

VIGILADA MINEDUCACIÓN - SNIES 1732

ANÁLISIS Y VISUALIZACIÓN DE DATOS CON PYTHON

CIENCIA DE LOS DATOS

- Sabemos que estamos ante una época de una proliferación inimaginable de datos. Se producen datos por doquier.
- Los datos nos proporcionan respuestas a través de valores ocultos y patrones.
- Hay suficientes datos, pero está la dificultad de analizarlos y tomar provecho de ellos.
- Para ello se aplican los métodos que permiten el análisis masivo de datos y poder extraer **conocimiento** de ellos.

ANÁLISIS DE DATOS

El proceso de análisis de datos permite:

- Refinar datos sin procesar.
- Transformar datos a una mejor calidad y formato estándar.
- Visualizar datos para facilitar su comprensión.

ANÁLISIS DE DATOS

Tipos de análisis de datos:

- Análisis de texto.
- Análisis estadístico.
- Análisis de diagnóstico.
- Análisis predictivo.
- Análisis prescriptivo.

ANÁLISIS DE DATOS

1. Análisis de texto:

Comprende el proceso automatizado para extraer y clasificar información de un texto como por ejemplo emails, tweets, respuesta a encuestas, etc.

Dentro de las tareas más comunes se encuentran análisis de sentimientos, extracción de palabras clave y detección de temas.

ANÁLISIS DE DATOS

2. Análisis estadístico:

Este proceso permite recolectar, explorar y presentar grandes volúmenes de datos para identificar patrones y tendencias.

Mediante el uso de estadísticas, es posible agrupar y organizar los datos, para luego realizar algunas inferencias.

Por ejemplo: analizar el flujo de personas que frecuentan una sala de urgencias en un periodo de tiempo.

ANÁLISIS DE DATOS

3. Análisis de diagnóstico:

En este proceso se examinan los datos y se busca identificar la razón por la cual sucedió un evento particular.

A través del análisis de diagnóstico es posible hacer correlaciones de las variables que intervienen en el estudio y así poder determinar la razón de cierto comportamiento como el aumento de casos del COVID por ejemplo. En este caso se identificaron las variables que tenían incidencia en la exposición y contagio del virus.

ANÁLISIS DE DATOS

4. Análisis predictivo:

En este proceso se trabaja con datos históricos para encontrar patrones y poder predecir efectos similares a futuro.

En el ejemplo del COVID, se puede predecir un aumento de casos por cierta condición como una festividad que promueve las reuniones de varias personas en sitios cerrados.

ANÁLISIS DE DATOS

5. Análisis prescriptivo:

En este proceso se sugieren acciones ante la situación detectada. Por ejemplo, en el caso de un aumento de casos, se puede sugerir establecer políticas restrictiva para el ingreso a ciertos establecimientos o el aumento de personal médico que pueda atender la emergencia.

ANÁLISIS DE DATOS

Incluye diversos procesos, tales como:

- Recopilación de datos.
- Limpieza de datos.
- Procesamiento, configuración de datos.
- Visualización.
- Interpretación.
- Divulgación de resultados.

FORMATO DE LOS DATOS

Los conjuntos de datos pueden estar en diferentes formatos, pero los más comunes son:

- CSV (Comma Separated Value).
- Excel
- JSON (JavaScript Object Notation).
- Texto

REGRESIÓN LINEAL: Taller

1. Buscar en el siguiente link un conjunto de datos para regresión univariable

https://archive.ics.uci.edu/ml/datasets.php

Bike Sharing Dataset (data folder)

2. Agregar las librerías: pandas, numpy, matplotlib, seaborn y sklearn para usar el modelo lineal.
3. Importar el conjunto de datos (por días)

REGRESIÓN LINEAL: Taller

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model
import seaborn as sns
import warnings

from google.colab import files
datos = files.upload()
```

Elegir archivos day.csv

- **day.csv**(application/vnd.ms-excel) - 57569 bytes, last modified: 20/9/2021 - 100% done

```
Saving day.csv to day (1).csv
```

REGRESIÓN LINEAL: Taller

4. Convertir el conjunto de datos en un dataframe y visualizarlo.

REGRESIÓN LINEAL: Taller

4. Convertir el conjunto de datos en un dataframe y visualizarlo.

```
import io
datos_bici = pd.read_csv(io.BytesIO(datos['day.csv']))
datos_bici
```

5. Visualizar la información estadística

REGRESIÓN LINEAL: Taller

4. Convertir el conjunto de datos en un dataframe y visualizarlo.

```
import io
datos_bici = pd.read_csv(io.BytesIO(datos['day.csv']))
datos_bici
```

5. Visualizar la información estadística

```
datos_bici.describe()
```

REGRESIÓN LINEAL: Taller

6. Visualizar la información de los tipos de datos por campo

REGRESIÓN LINEAL: Taller

6. Visualizar la información de los tipos de datos por campo

```
datos_bici.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 731 entries, 0 to 730
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   instant     731 non-null    int64  
 1   dteday      731 non-null    object 
 2   season      731 non-null    int64  
 3   yr          731 non-null    int64  
 4   mnth        731 non-null    int64  
 5   holiday     731 non-null    int64  
 6   weekday     731 non-null    int64  
 7   workingday  731 non-null    int64  
 8   weathersit  731 non-null    int64  
 9   temp        731 non-null    float64
 10  atemp       731 non-null    float64
 11  hum         731 non-null    float64
 12  windspeed   731 non-null    float64
 13  casual      731 non-null    int64  
 14  registered  731 non-null    int64  
 15  cnt         731 non-null    int64  
dtypes: float64(4), int64(11), object(1)
memory usage: 91.5+ KB
```

REGRESIÓN LINEAL: Taller

7. Visualizar los valores únicos de todos los campos

```
datos_bici.apply(lambda x: len(x.unique()))
```

```
instant       731
dteday        731
season          4
yr              2
mnth           12
holiday         2
weekday         7
workingday      2
weathersit      3
temp          499
atemp         690
hum           595
windspeed     650
casual        606
registered    679
cnt           696
dtype: int64
```

REGRESIÓN LINEAL: Taller

8. Visualizar los valores vacíos

REGRESIÓN LINEAL: Taller

8. Visualizar los valores vacíos

```
datos_bici.isnull().sum()
```

```
instant       0
dteday        0
season        0
yr            0
mnth          0
holiday       0
weekday       0
workingday    0
weathersit    0
temp          0
atemp         0
hum           0
windspeed     0
casual        0
registered    0
cnt           0
dtype: int64
```

REGRESIÓN LINEAL: Taller

9. Si se quiere renombrar algunas columnas para entender mejor la información que contienen:

```
[14] datos_bici=datos_bici.rename(columns={'weathersit':'weather',
                                           'yr':'year',
                                           'mnth':'month',
                                           'hum': 'humidity',
                                           'cnt':'count'})
```

REGRESIÓN LINEAL: Taller

10. ¿Cuál información no es relevante para el análisis y predicción?

```
datos_bici.head()
```

	instant	dteday	season	year	month	holiday	weekday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600

REGRESIÓN LINEAL: Taller

10. ¿Cuál información no es relevante para el análisis y predicción? Se deben borrar del conjunto de datos.

```
datos_bici.head()
```

	instant	dteday	season	year	month	holiday	weekday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600

REGRESIÓN LINEAL: Taller

10. ¿Cuál información no es relevante para el análisis y predicción? Se deben borrar del conjunto de datos.

```
datos_bici.head()
```

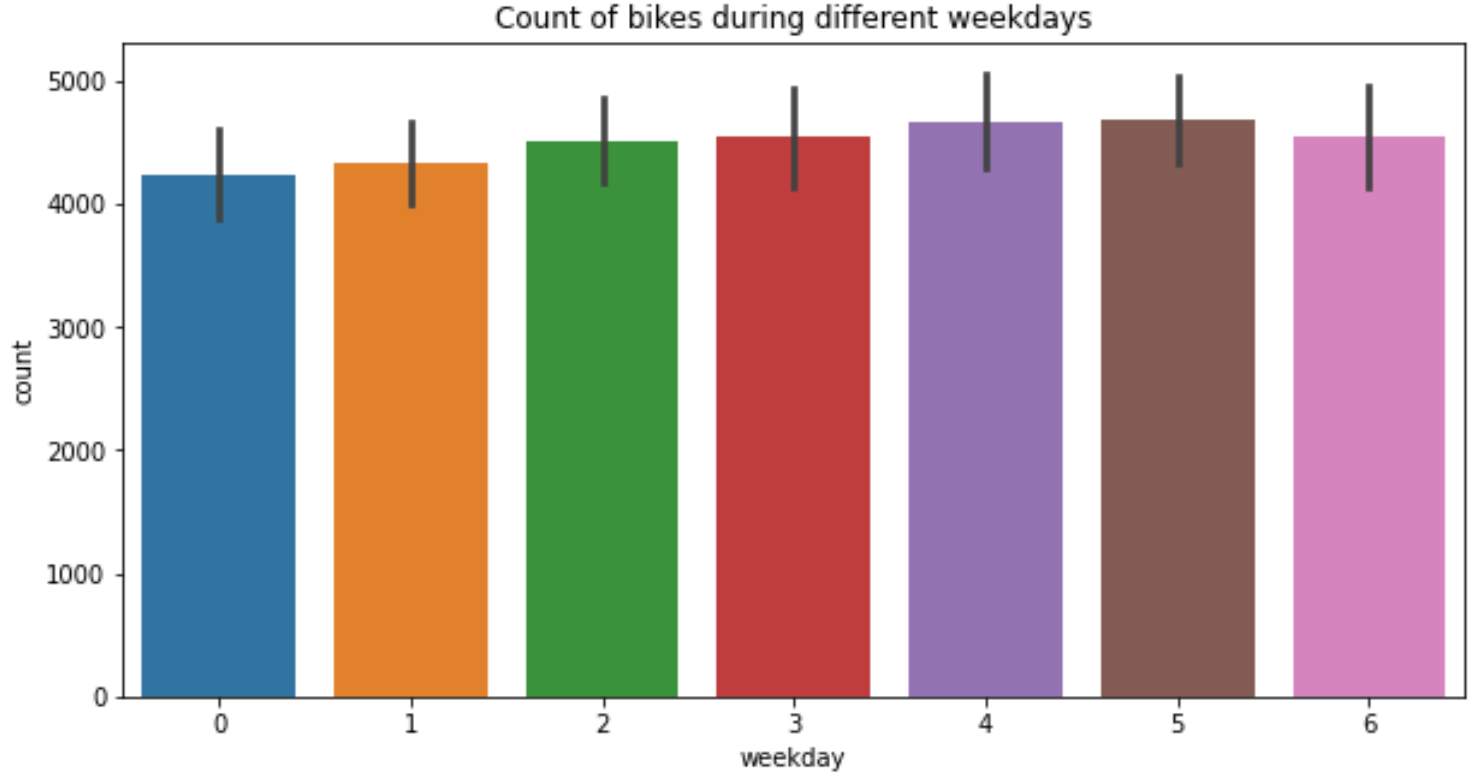
	instant	dteday	season	year	month	holiday	weekday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600

```
datos_bici=datos_bici.drop(columns=['instant','dteday','year'])
datos_bici.head()
```

REGRESIÓN LINEAL: Taller

```
fig, ax=plt.subplots(figsize=(10,5))
sns.barplot(data=datos_bici, x='weekday', y='count', ax=ax)
ax.set(title='Count of bikes during different weekdays')
```

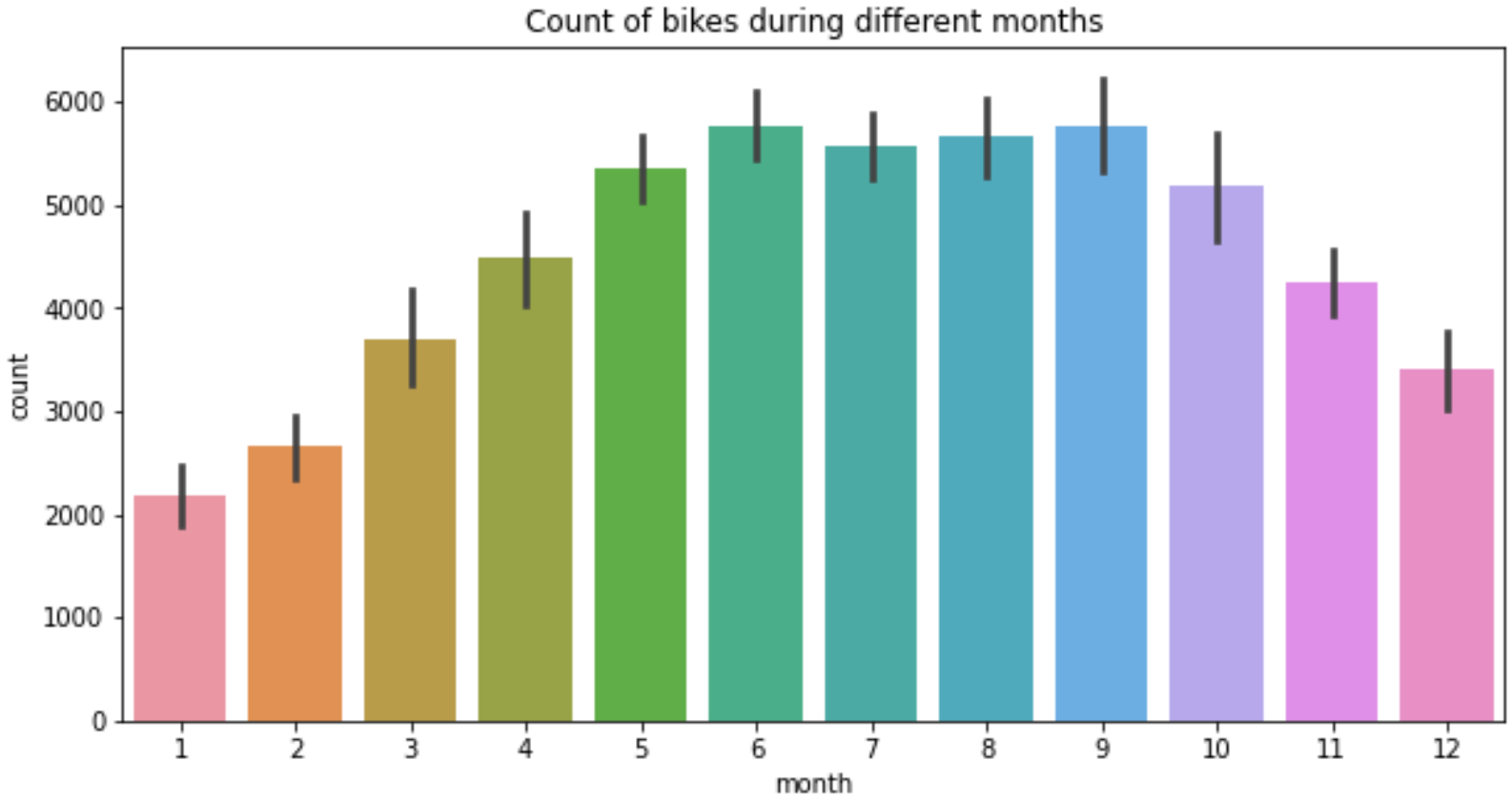
```
[Text(0.5, 1.0, 'Count of bikes during different weekdays')]
```



REGRESIÓN LINEAL: Taller

```
fig, ax=plt.subplots(figsize=(10,5))
sns.barplot(data=datos_bici, x='month', y='count', ax=ax)
ax.set(title='Count of bikes during different months')
```

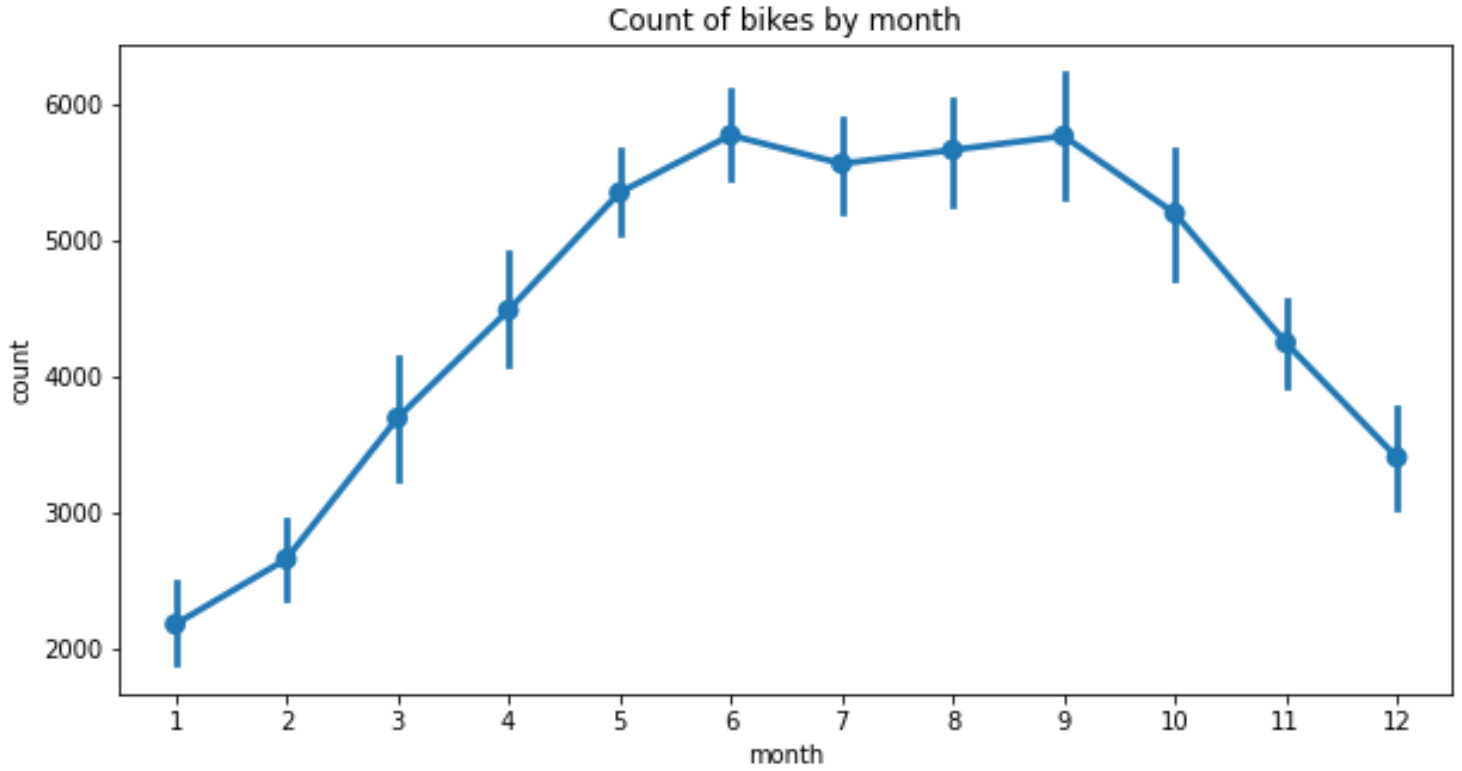
```
[Text(0.5, 1.0, 'Count of bikes during different months')]
```



REGRESIÓN LINEAL: Taller

```
fig, ax=plt.subplots(figsize=(10,5))
sns.pointplot(data=datos_bici, x='month', y='count',ax=ax)
ax.set(title='Count of bikes by month')
```

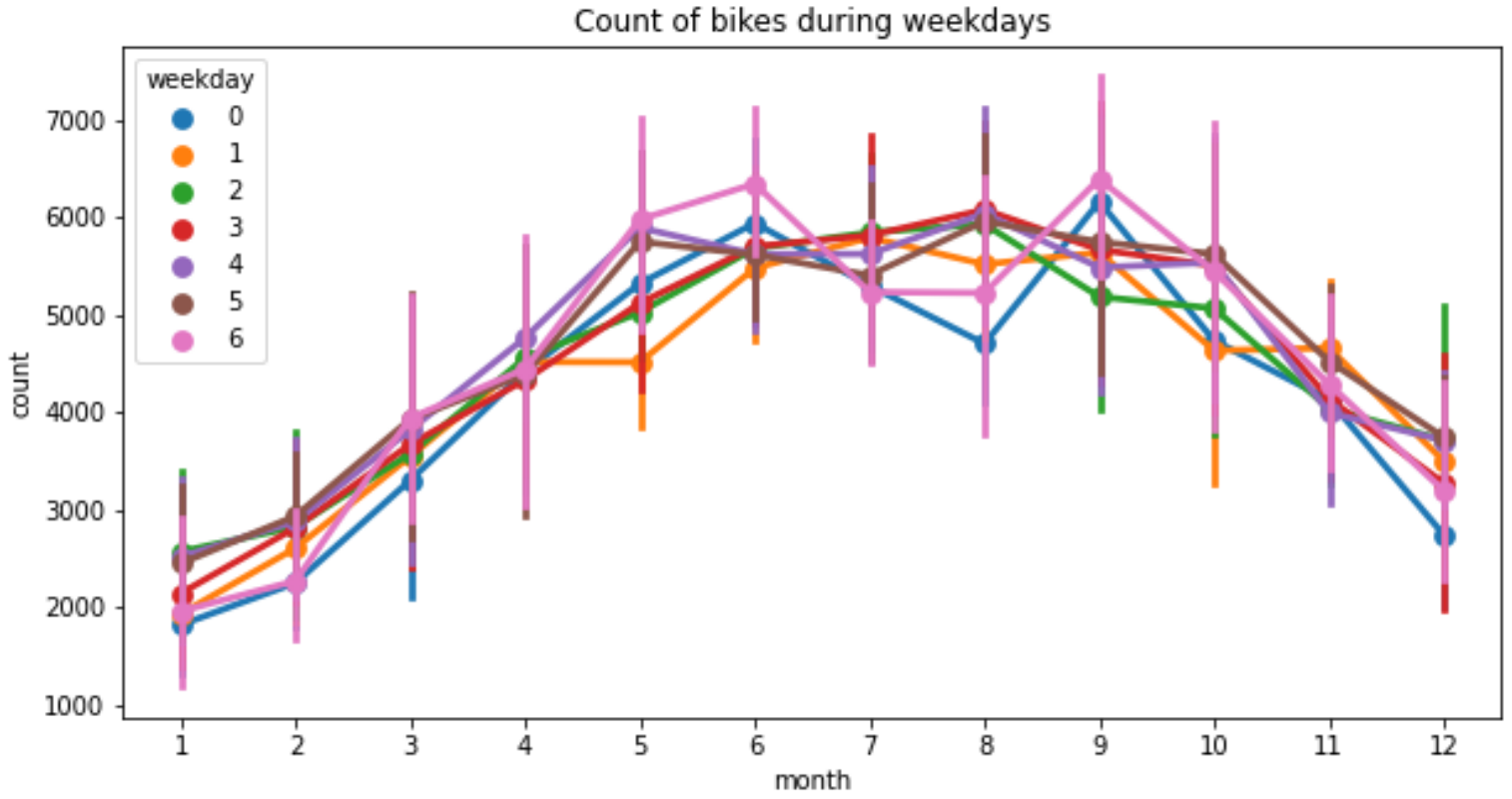
```
[Text(0.5, 1.0, 'Count of bikes by month')]
```



REGRESIÓN LINEAL: Taller

```
fig, ax=plt.subplots(figsize=(10,5))
sns.pointplot(data=datos_bici, x='month', y='count',hue='weekday', ax=ax)
ax.set(title='Count of bikes during weekdays')
```

```
[Text(0.5, 1.0, 'Count of bikes during weekdays')]
```

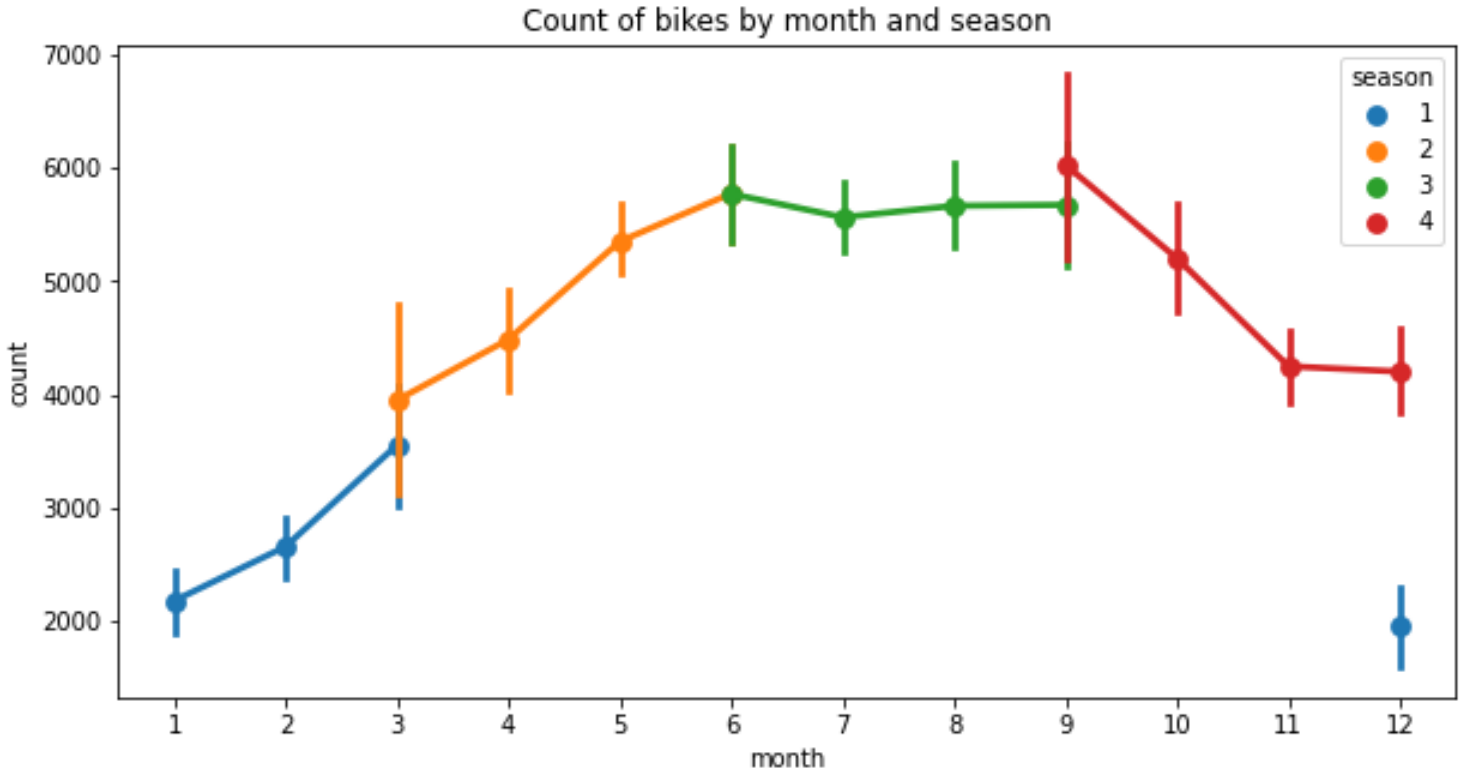


Hue: es la serie

REGRESIÓN LINEAL: Taller

```
fig, ax=plt.subplots(figsize=(10,5))
sns.pointplot(data=datos_bici, x='month', y='count',hue='season', ax=ax)
ax.set(title='Count of bikes by month and season')
```

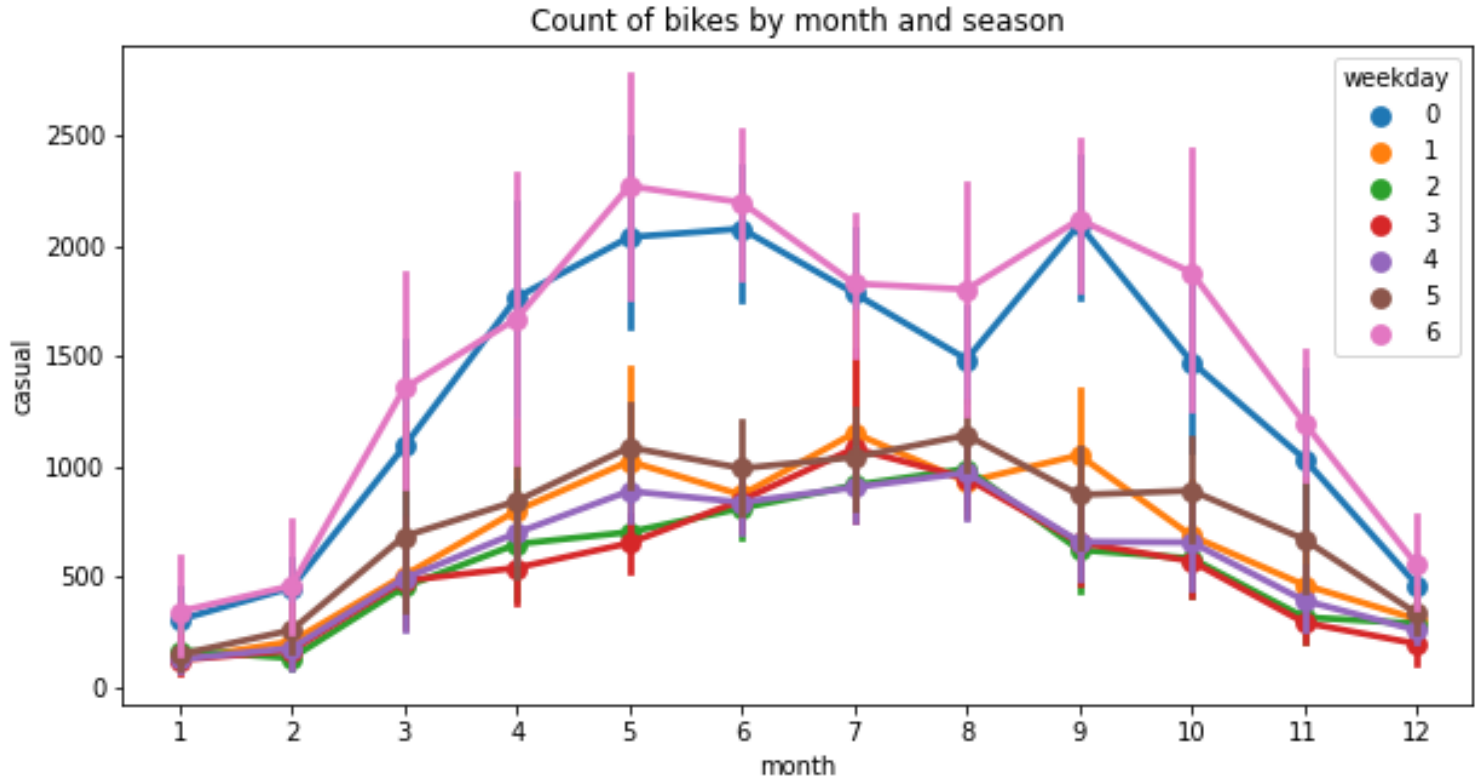
```
[Text(0.5, 1.0, 'Count of bikes by month and season')]
```



REGRESIÓN LINEAL: Taller

```
fig, ax=plt.subplots(figsize=(10,5))
sns.pointplot(data=datos_bici, x='month', y='casual',hue='weekday', ax=ax)
ax.set(title='Count of bikes during weekdays  - unregistered users')
```

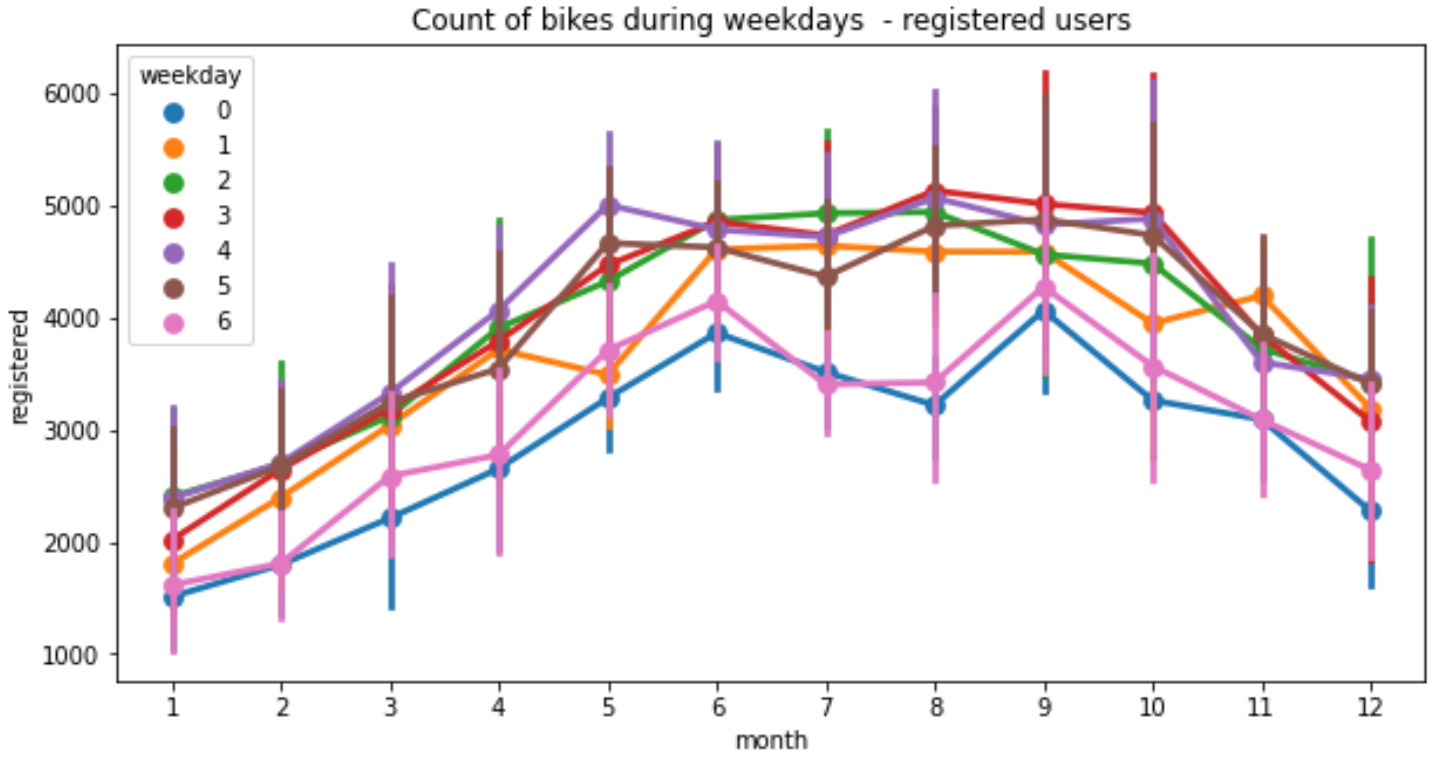
```
[Text(0.5, 1.0, 'Count of bikes by month and season')]
```



REGRESIÓN LINEAL: Taller

```
fig, ax=plt.subplots(figsize=(10,5))
sns.pointplot(data=datos_bici, x='month', y='registered',hue='weekday', ax=ax)
ax.set(title='Count of bikes during weekdays  - registered users')
```

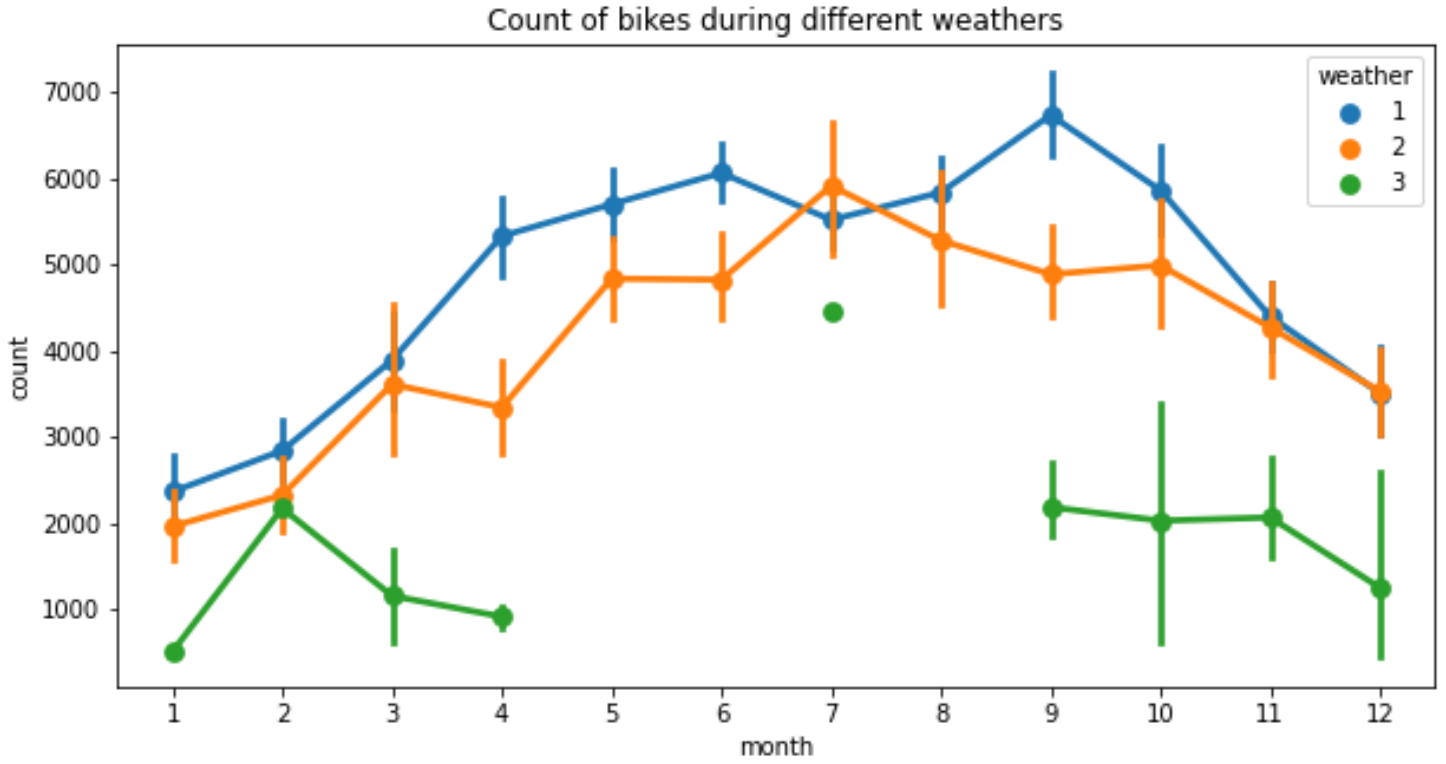
```
[Text(0.5, 1.0, 'Count of bikes during weekdays  - registered users')]
```



REGRESIÓN LINEAL: Taller

```
fig, ax=plt.subplots(figsize=(10,5))
sns.pointplot(data=datos_bici, x='month', y='count',hue='weather', ax=ax)
ax.set(title='Count of bikes during different weathers')
```

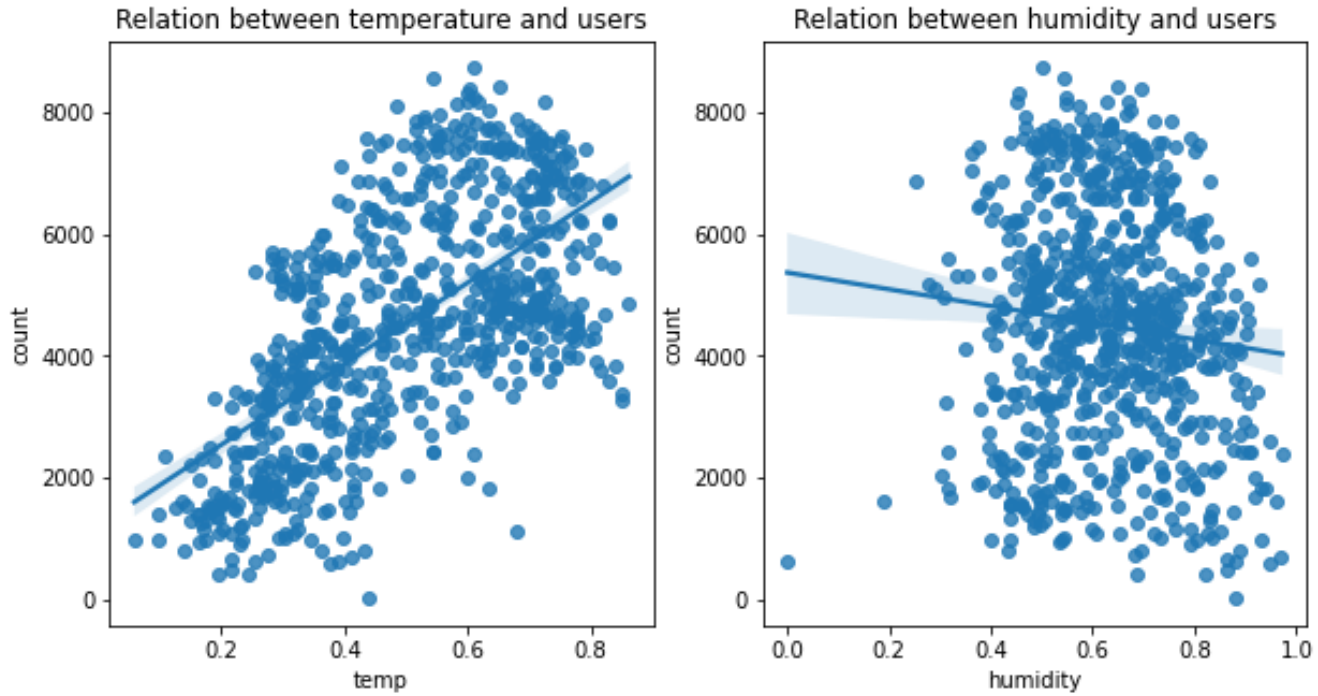
```
[Text(0.5, 1.0, 'Count of bikes during different weathers')]
```



REGRESIÓN LINEAL: Taller

```
fig, (ax1,ax2)=plt.subplots(ncols=2, figsize=(10,5))
sns.regplot(data=datos_bici, x='temp', y='count', ax=ax1)
ax1.set(title='Relation between temperature and users')
sns.regplot(data=datos_bici, x='humidity', y='count', ax=ax2)
ax2.set(title='Relation between humidity and users')
```

```
[Text(0.5, 1.0, 'Relation between humidity and users')]
```

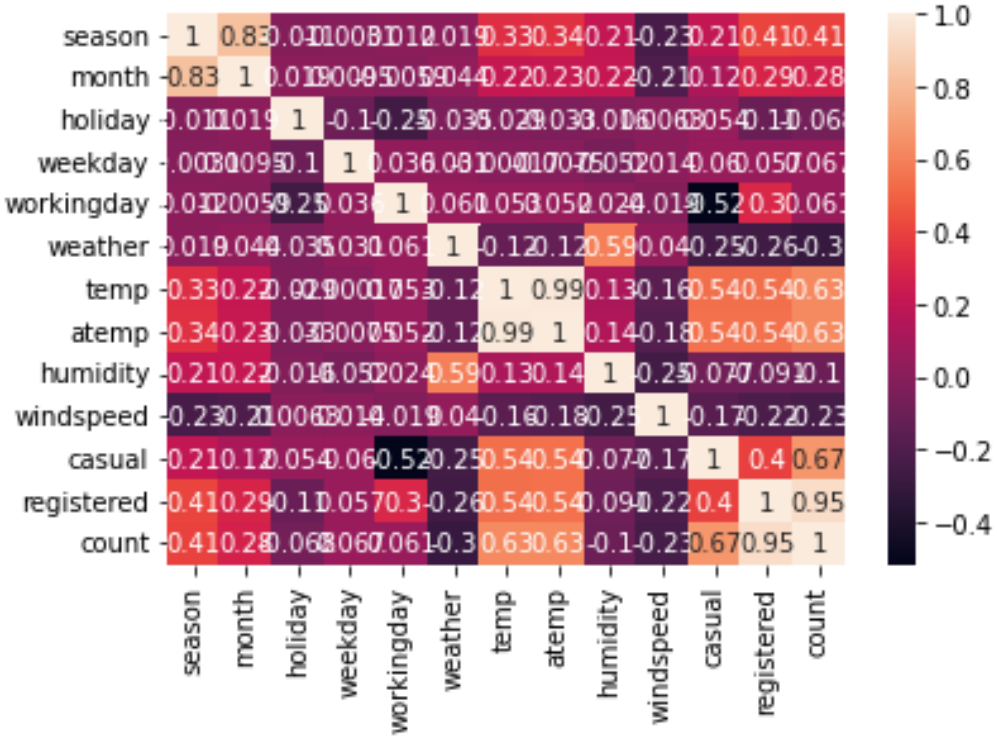


REGRESIÓN LINEAL: Taller

MATRIZ DE CORRELACIÓN

```
correlacion = datos_bici.corr()
sns.heatmap(correlacion, annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f33a1475610>
```



No se aprecia correctamente la información

¡Siempre hacia lo alto!

REGRESIÓN LINEAL: Taller

MATRIZ DE CORRELACIÓN

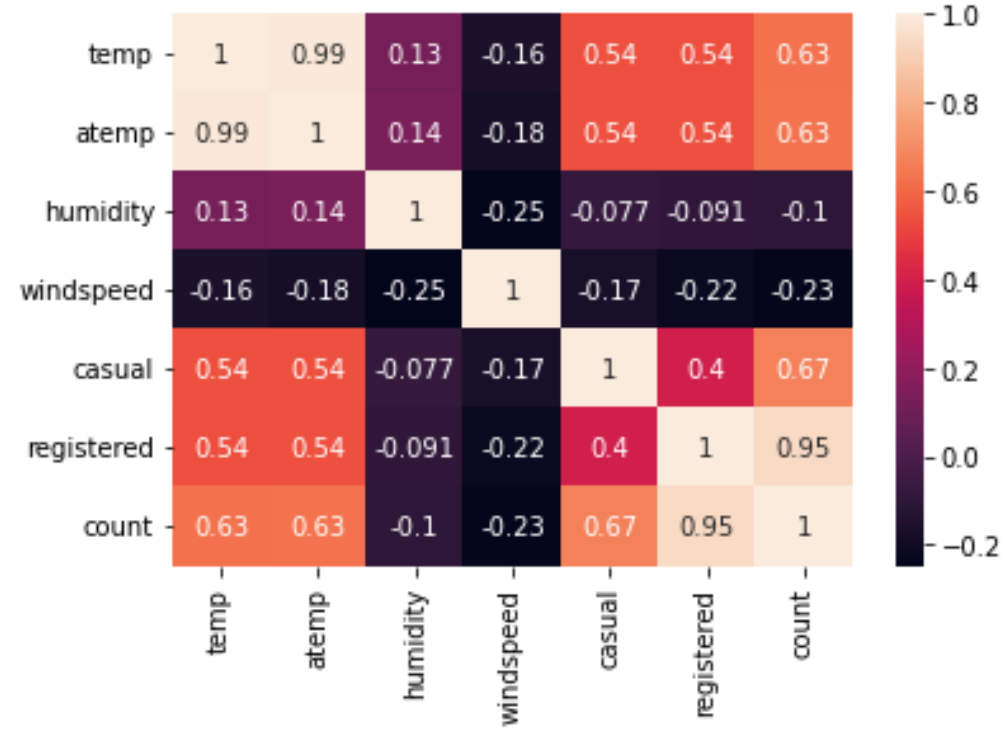
```
cols =['season', 'month', 'holiday','weekday','workingday','weather']
for col in cols:
  datos_bici[col]=datos_bici[col].astype('category')
datos_bici.info()
```

REGRESIÓN LINEAL: Taller

MATRIZ DE CORRELACIÓN

```
correlacion = datos_bici.corr()
sns.heatmap(correlacion, annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f33a1b2bc50>
```



REFERENCIAS

https://youtu.be/3Ua6lT7Ye0A

¡Siempre hacia lo alto!