

# Relation

Gráficos de tipo relación, Seaborn maneja también el Scatterplot, entre otros.

```
In [ ]: import seaborn as sns
import matplotlib.pyplot as plt
```

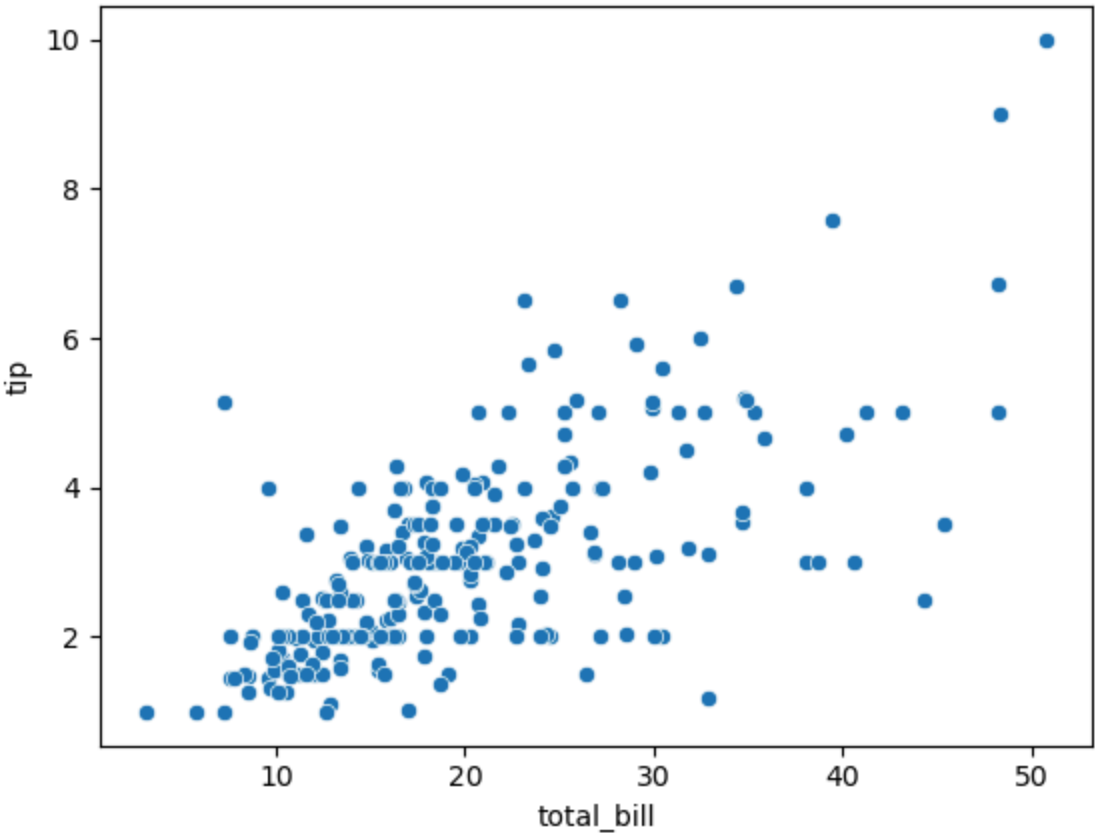
```
In [ ]: tips = sns.load_dataset('tips')
tips.head(3)
```

Out[ ]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3

## Scatterplot

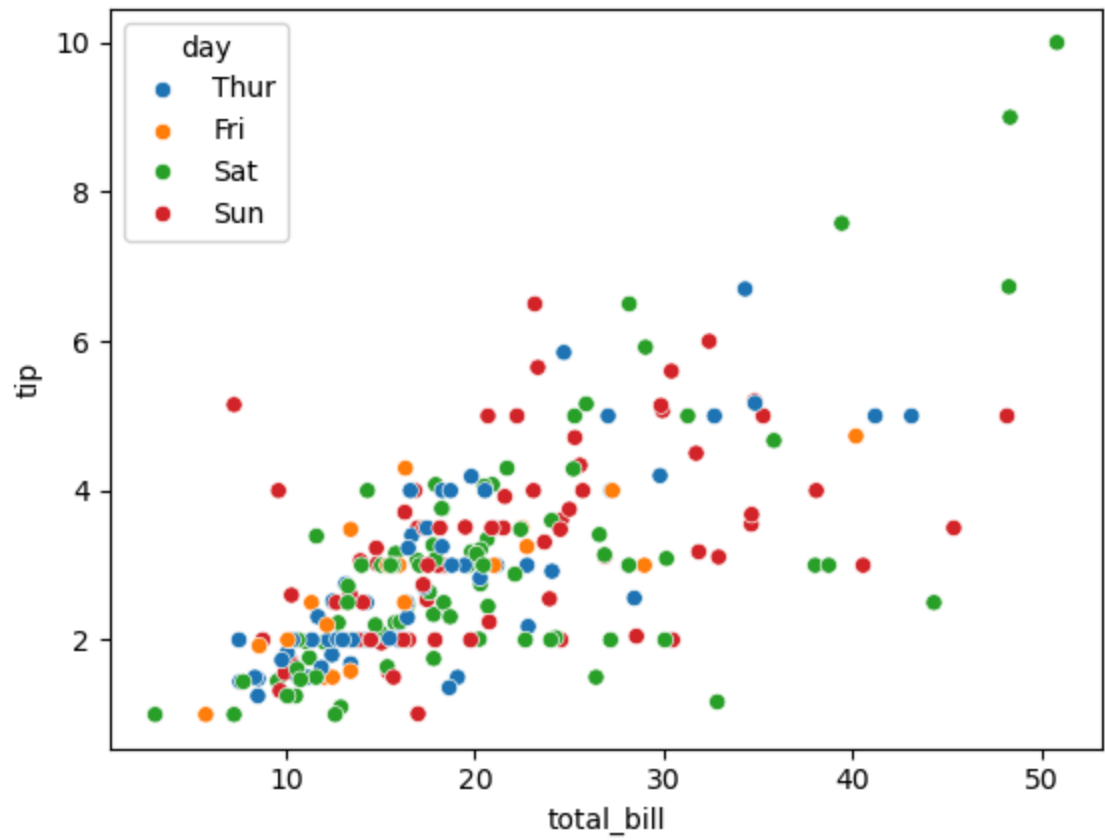
```
In [ ]: #Grafico de dispersión
sns.scatterplot(data=tips,x='total_bill',y='tip')
plt.show()
```



Vamos a analizar la relación que existe entre la cuenta y el total de las propinas. Se puede ver un comportamiento de relación - > Entre mas crece la cuenta, tiende a crecer la propina, también hay casos atípicos

## Segmentando por día

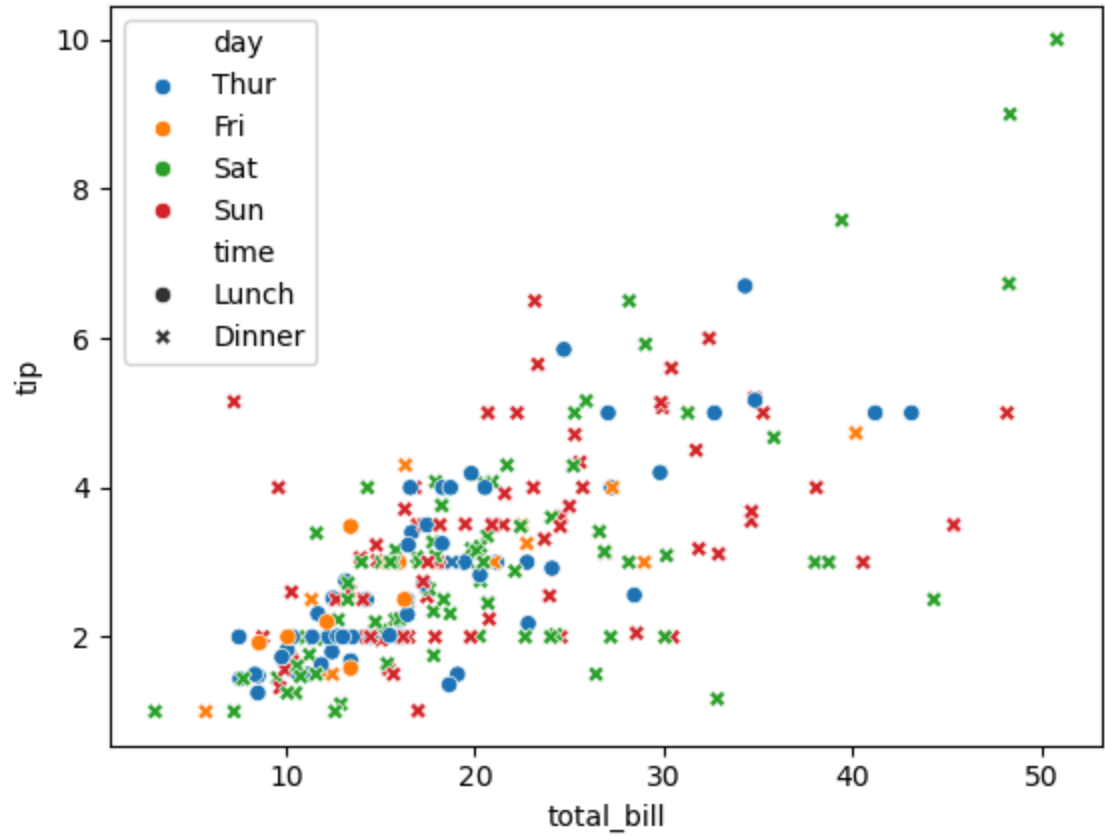
```
In [ ]: #Grafico de dispersión
sns.scatterplot(data=tips,x='total_bill',y='tip',hue='day')
plt.show()
```



Podemos ver que por color se hace la segmentación y por ciertos días hay un consumo mayor en la cuenta y a su vez una mayor propina.

### Estilo dependiente de 'time'

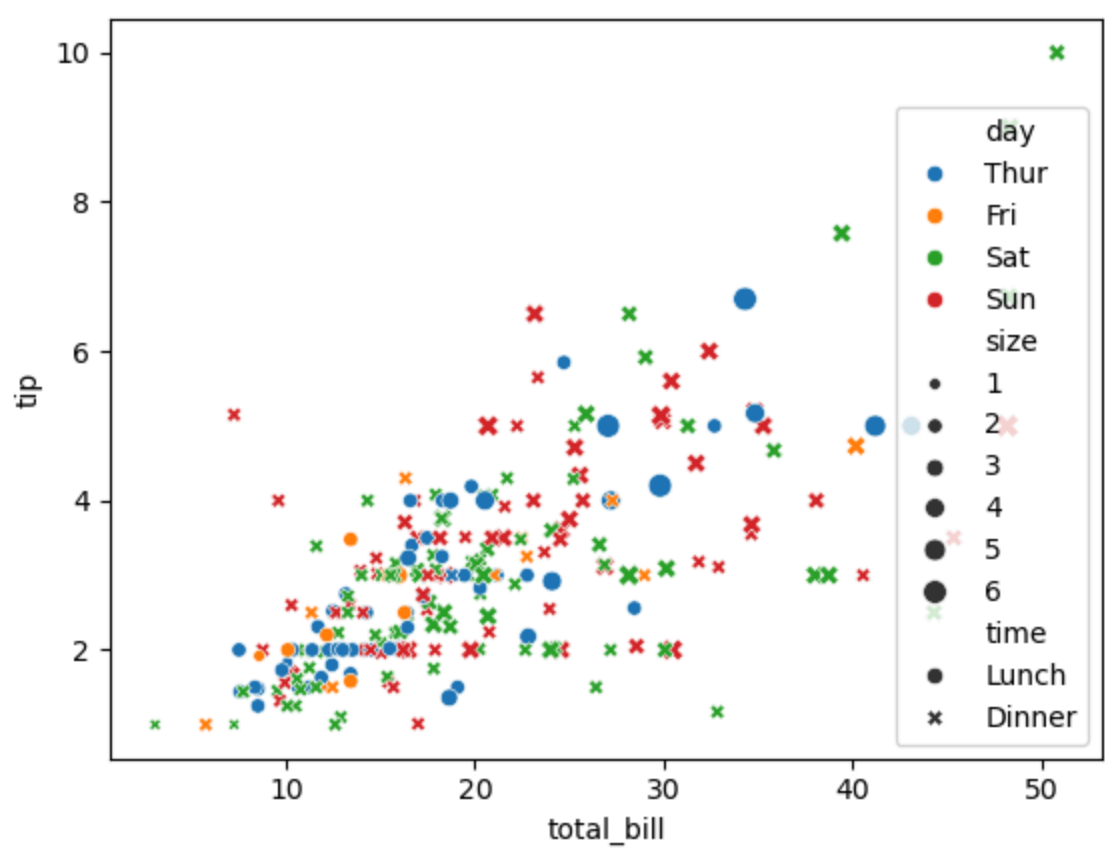
```
In [ ]: #Grafico de dispersión
sns.scatterplot(data=tips,x='total_bill',y='tip',hue='day',style='time')
plt.show()
```



Aquí hay otro tipo de segmentación en el que gráficamente se puede distinguir entre `lunch` y `dinner` , por medio de un marcador diferente. Entonces como se puede apreciar, parece que para `dinner` hay tanto un consumo mayor en `total_bill` y `tip`

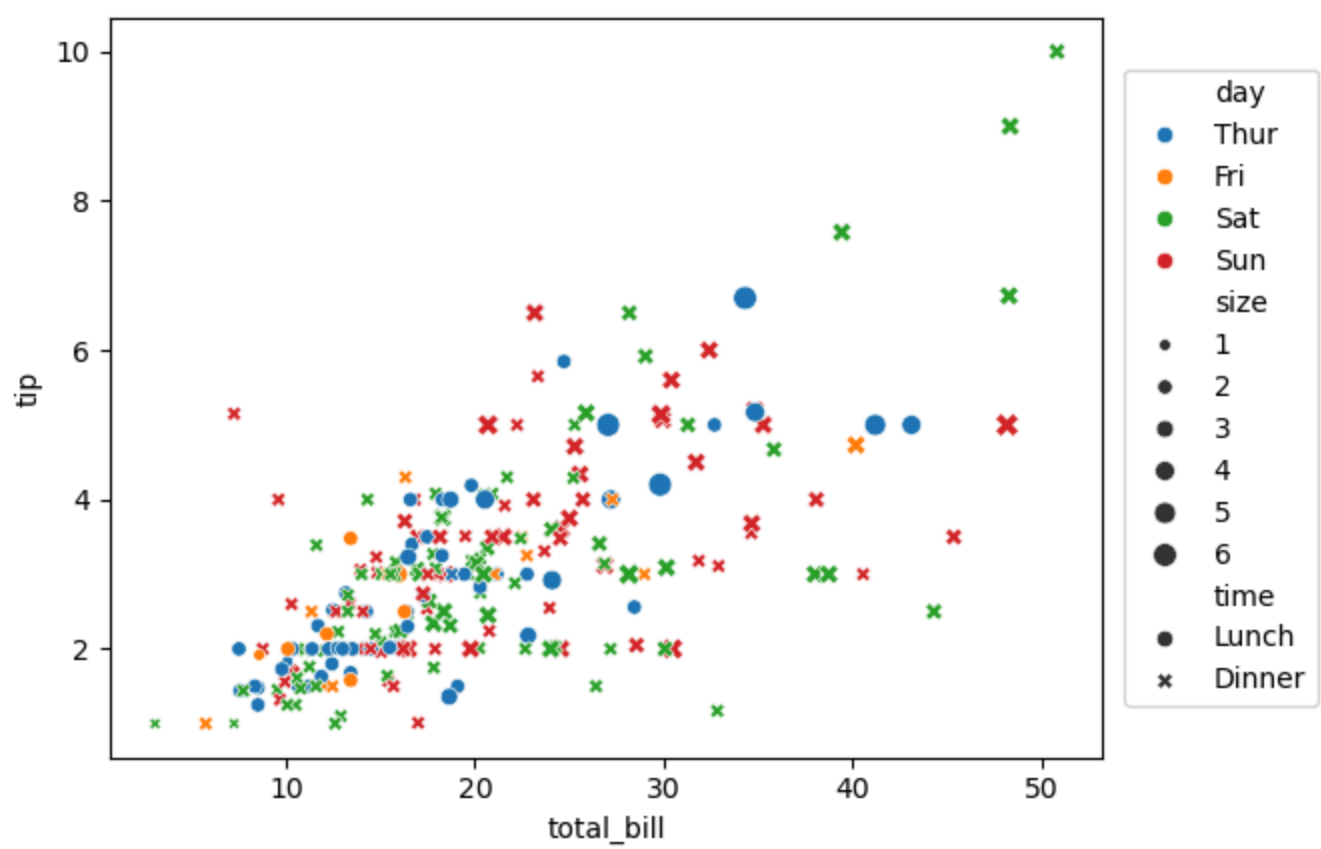
### Tamaño de marcador relacionado a size(número de comensales)

```
In [ ]: #Grafico de dispersión
sns.scatterplot(data=tips,x='total_bill',y='tip',hue='day',style='time',size='size')
plt.show()
```



La leyenda anterior me molesta la gráfica así que vamos a moverla.

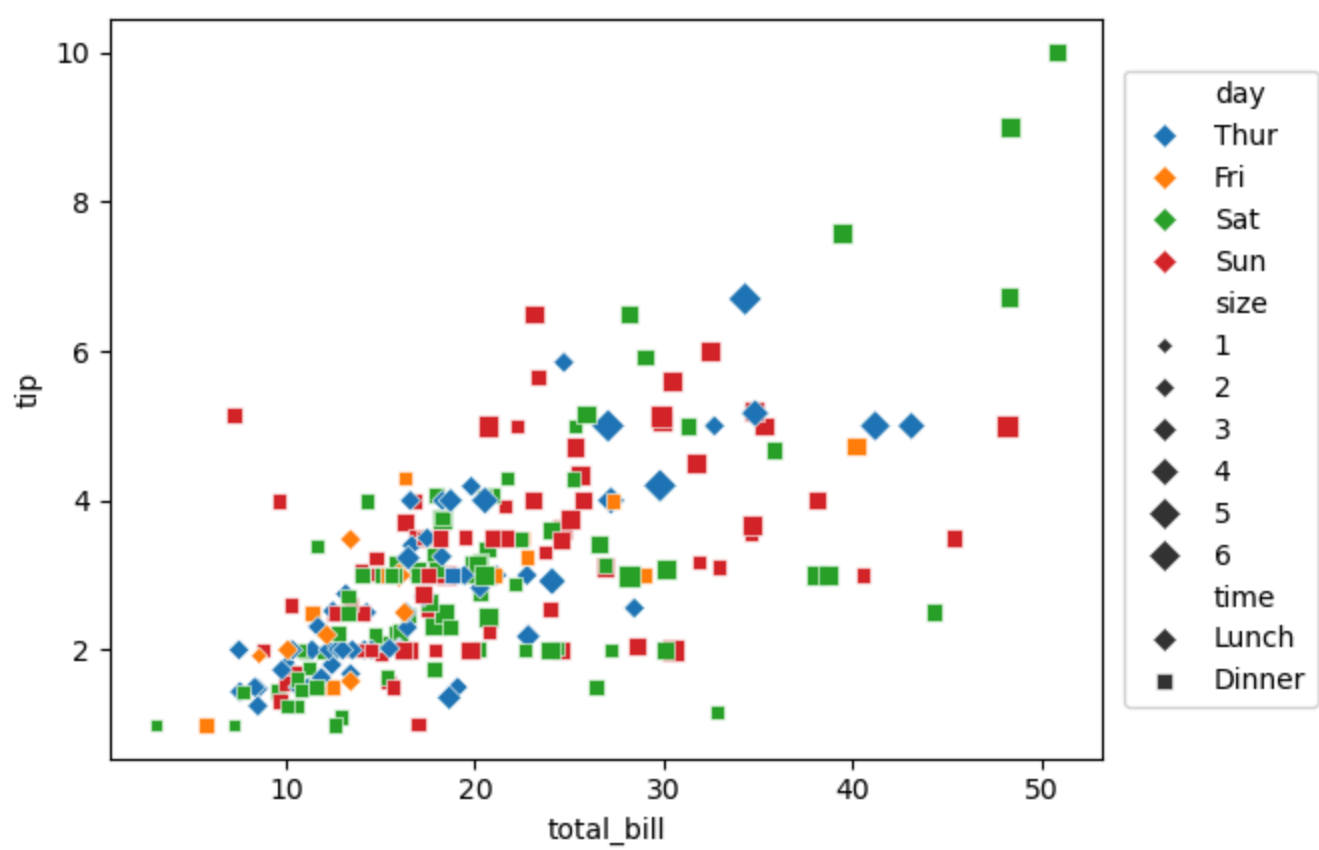
```
In [ ]: #Grafico de dispersión
sns.scatterplot(data=tips,x='total_bill',y='tip',hue='day',style='time',size='size')
plt.legend(loc='center',bbox_to_anchor=(1.12,0.5))
plt.show()
```



Así mi leyenda ya no me estorba y puedo realizar un analisis correctamente.

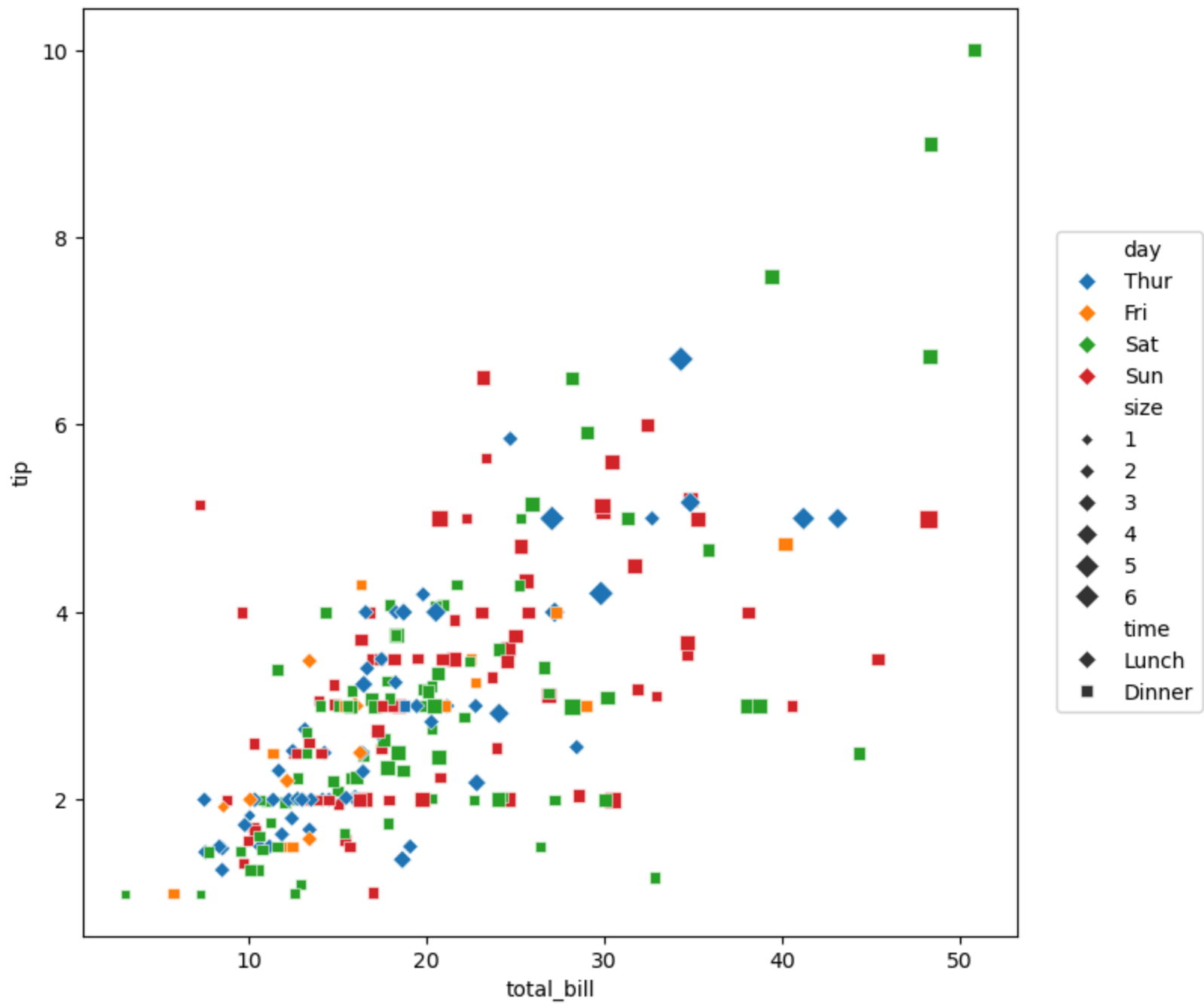
### Definiendo markers

```
In [ ]: #Grafico de dispersión
marker = {'Lunch':'D','Dinner':'s'}
sns.scatterplot(data=tips,x='total_bill',y='tip',hue='day',style='time',size='size',markers=marker)
plt.legend(loc='center',bbox_to_anchor=(1.12,0.5))
plt.show()
```



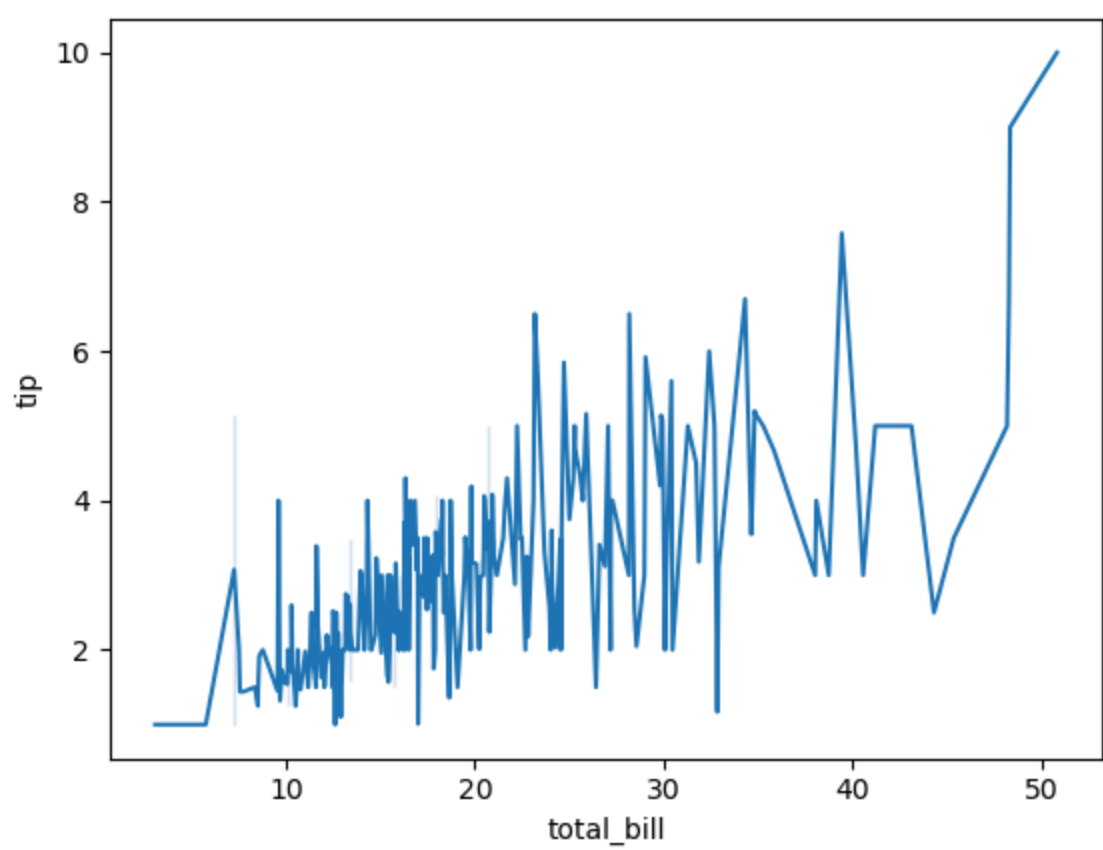
### Ajustando tamaño

```
In [ ]: #Grafico de dispersión
marker = {'Lunch':'D','Dinner':'s'}
plt.figure(figsize=(8,8))
sns.scatterplot(data=tips,x='total_bill',y='tip',hue='day',style='time',size='size',markers=marker)
plt.legend(loc='center',bbox_to_anchor=(1.12,0.5))
plt.show()
```

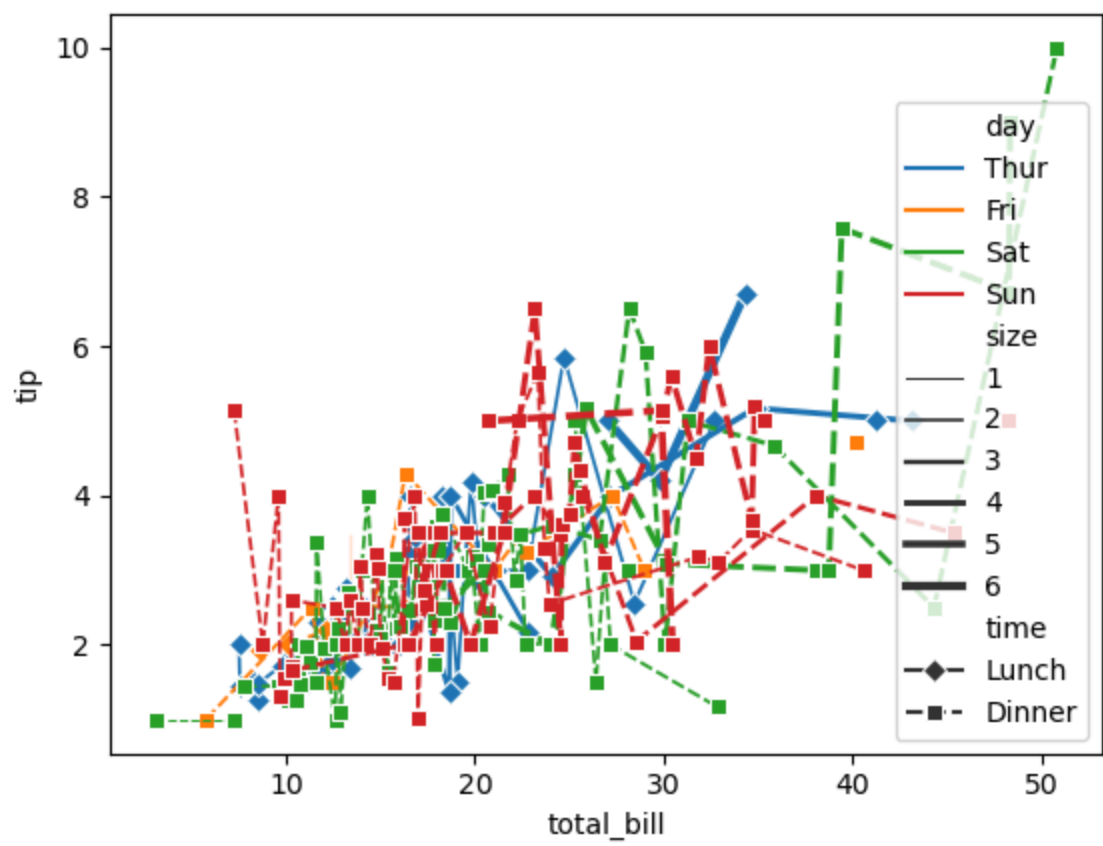


### Lineplot

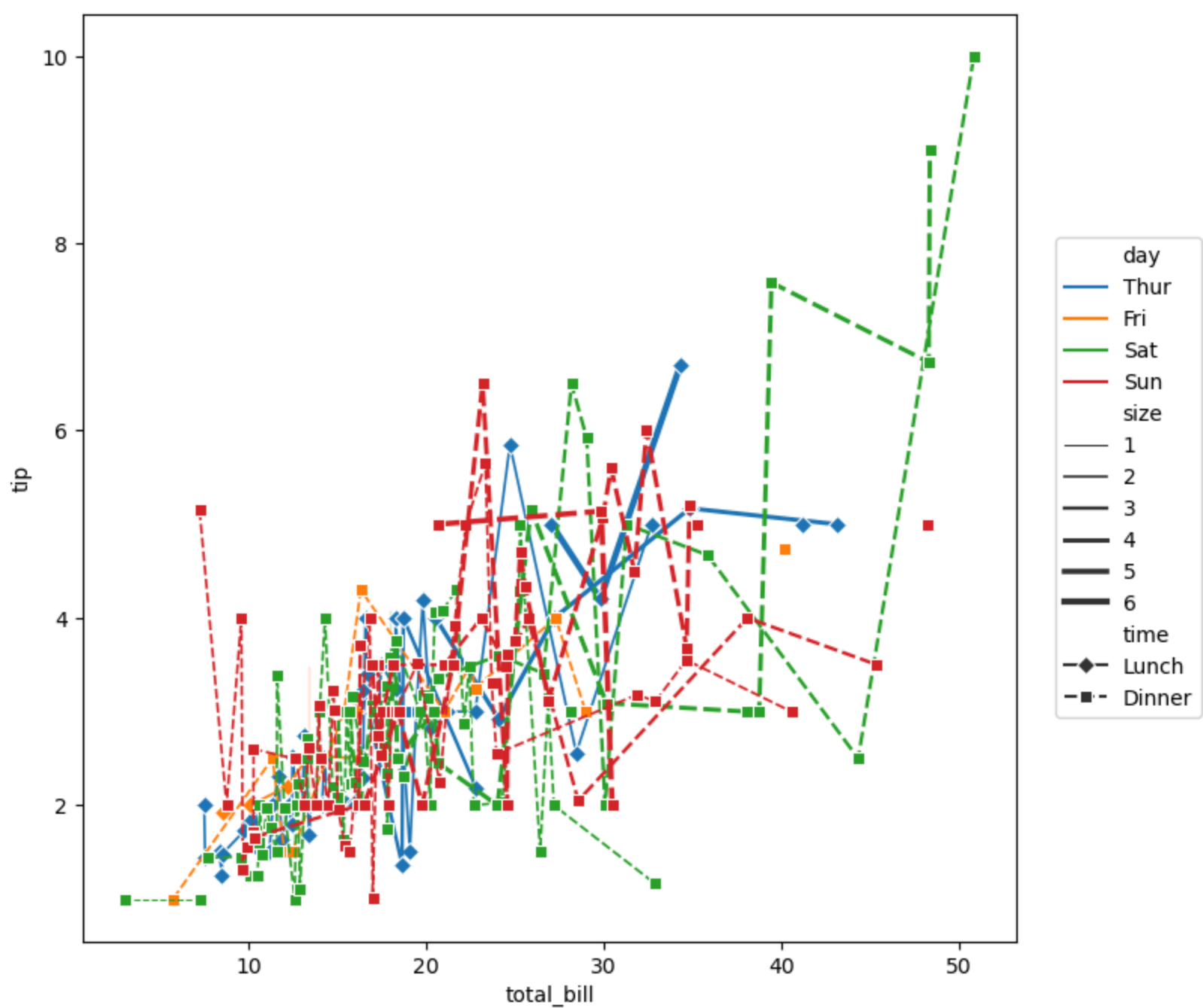
```
In [ ]: #Grafico lineplot
sns.lineplot(data=tips,x='total_bill',y='tip')
plt.show()
```



```
In [ ]: #Grafico lineplot
sns.lineplot(data=tips,x='total_bill',y='tip',hue='day',style='time',size='size',markers=marker)
plt.show()
```

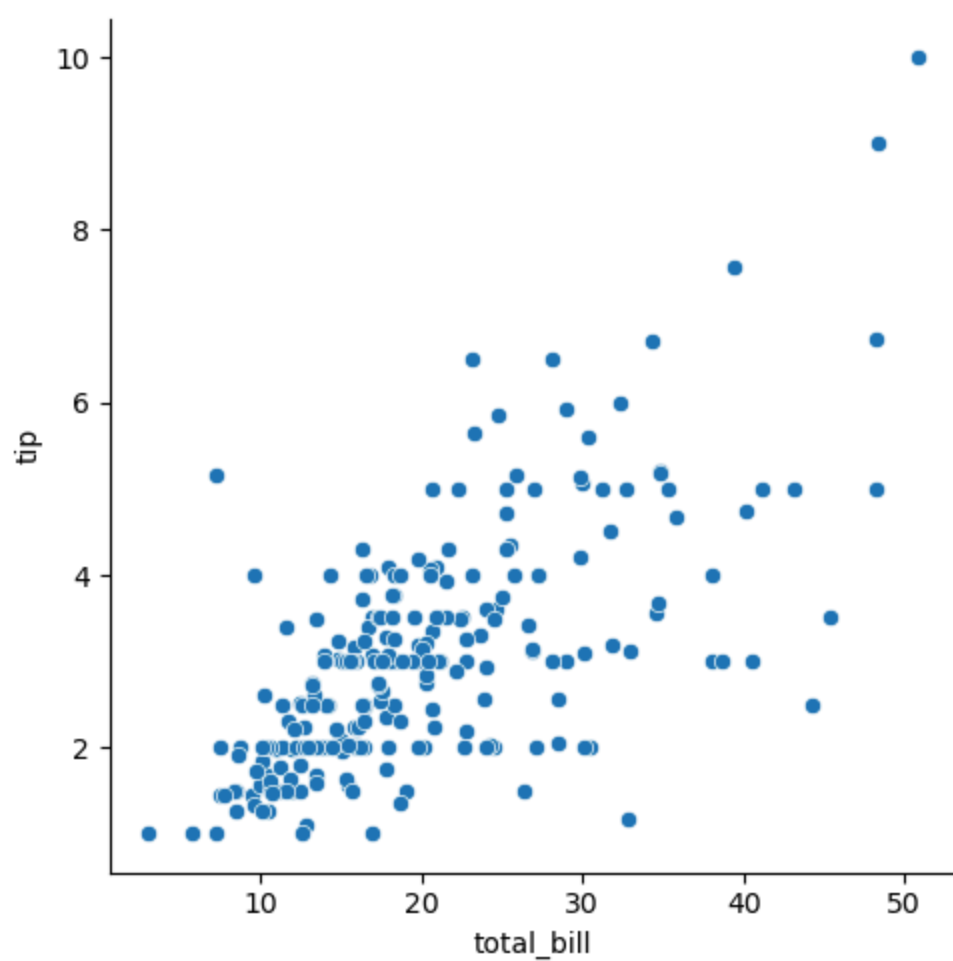


```
In [ ]: #Grafico lineplot
marker = {'Lunch':'D','Dinner':'s'}
plt.figure(figsize=(8,8))
sns.lineplot(data=tips,x='total_bill',y='tip',hue='day',style='time',size='size',markers=marker)
plt.legend(loc='center',bbox_to_anchor=(1.12,0.5))
plt.show()
```

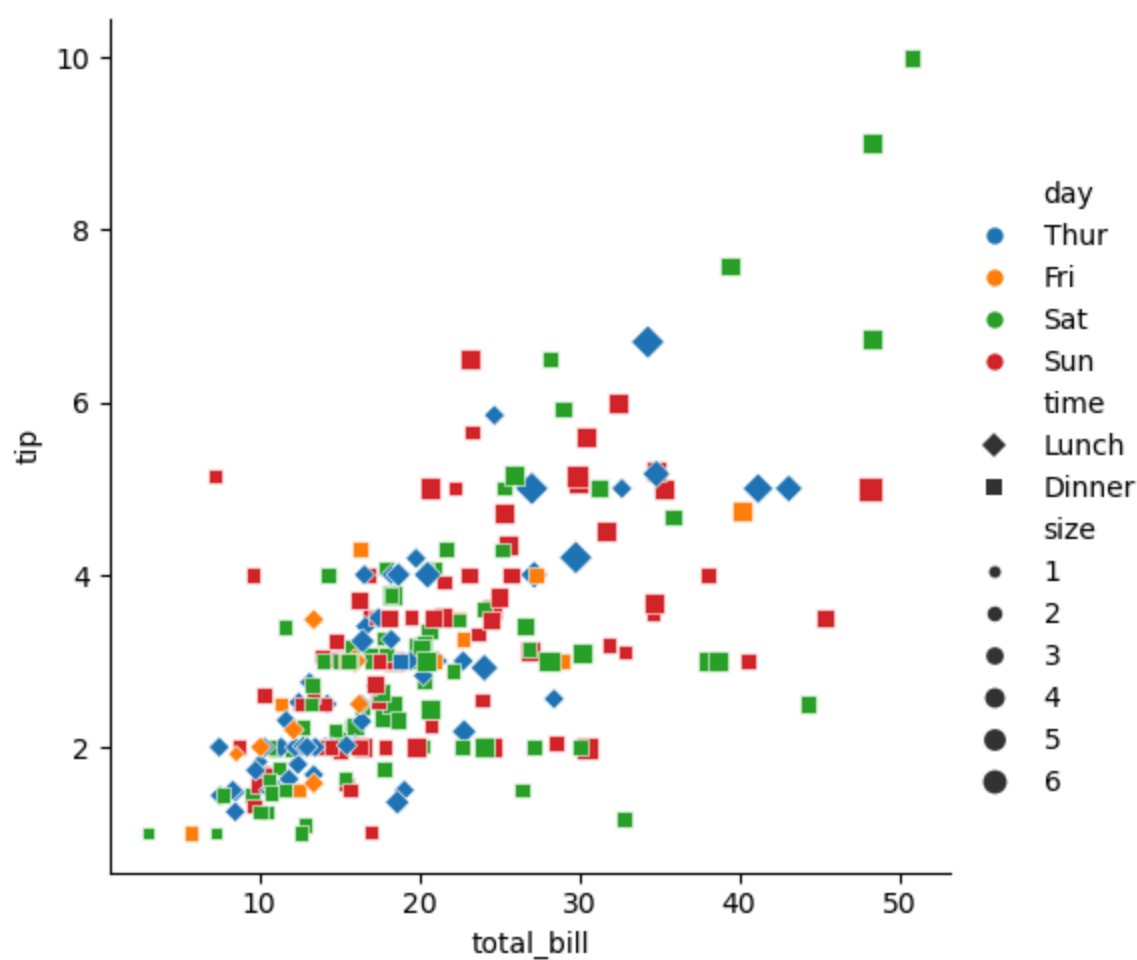


## Relplot

```
In [ ]: #Grafico relplot
sns.relplot(data=tips,x='total_bill',y='tip')
plt.show()
```

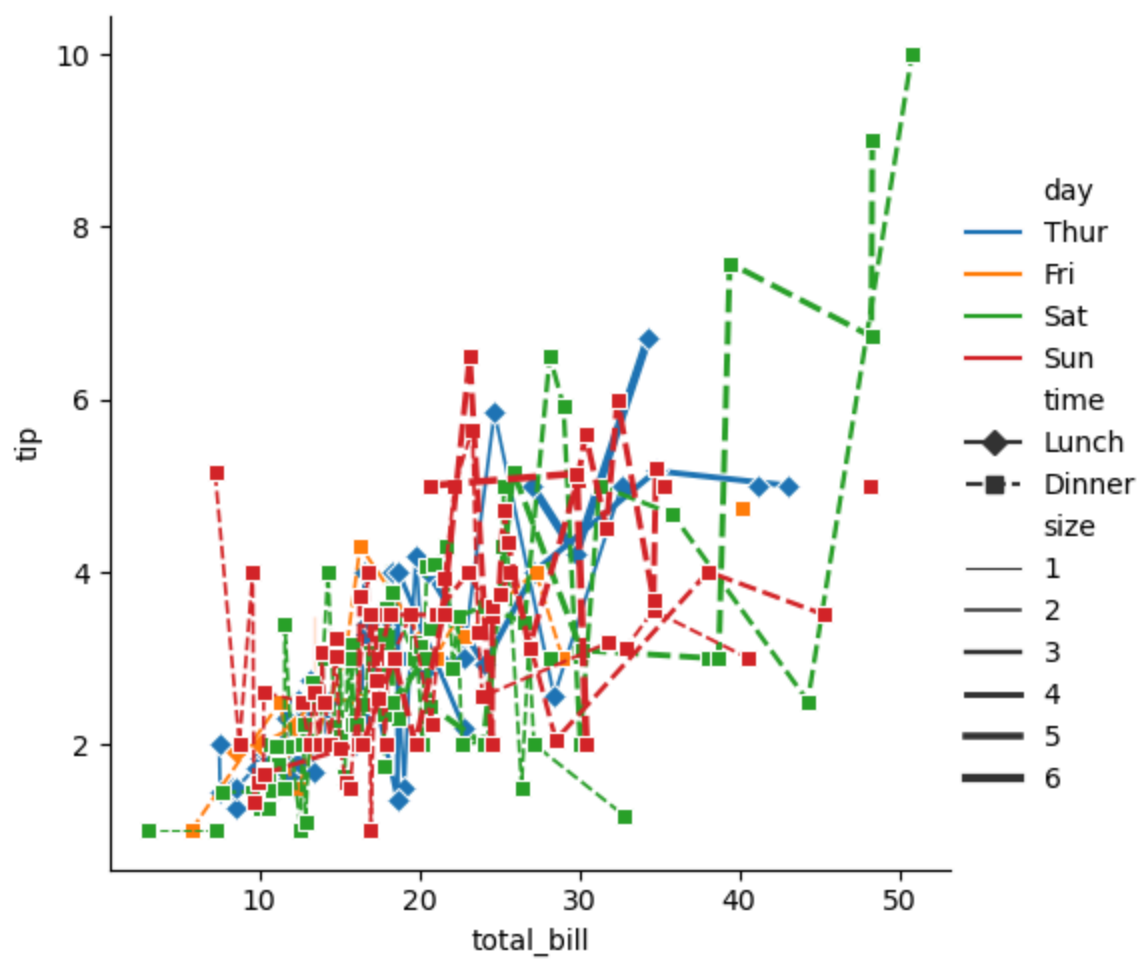


```
In [ ]: #Grafico relplot
sns.relplot(data=tips,x='total_bill',y='tip',hue='day',style='time',size='size',markers=marker)
plt.show()
```



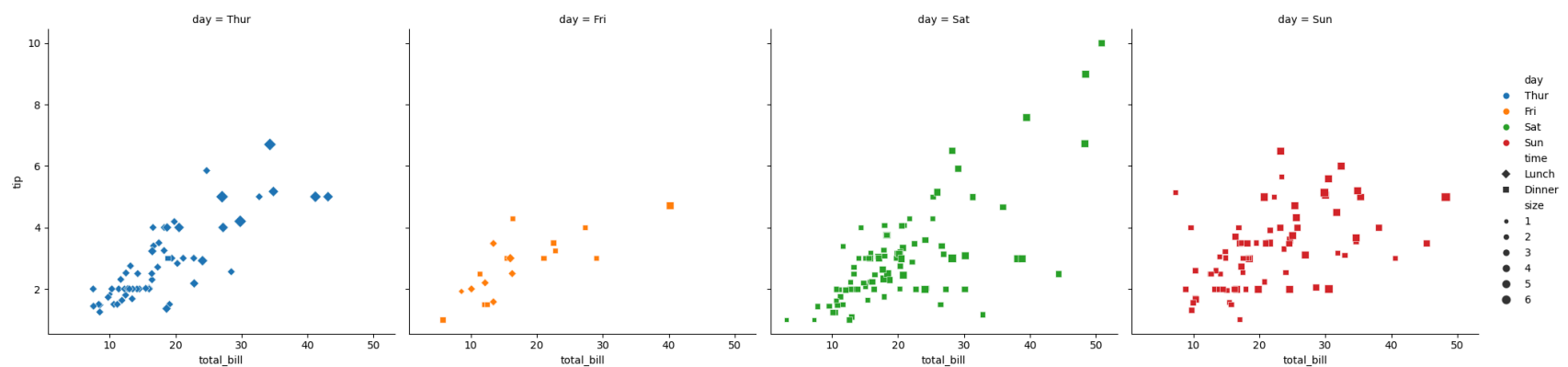
## Usando Kind

```
In [ ]: #Grafico replot
sns.relplot(data=tips,x='total_bill',y='tip',hue='day',style='time',size='size',markers=marker,kind='line')
plt.show()
```

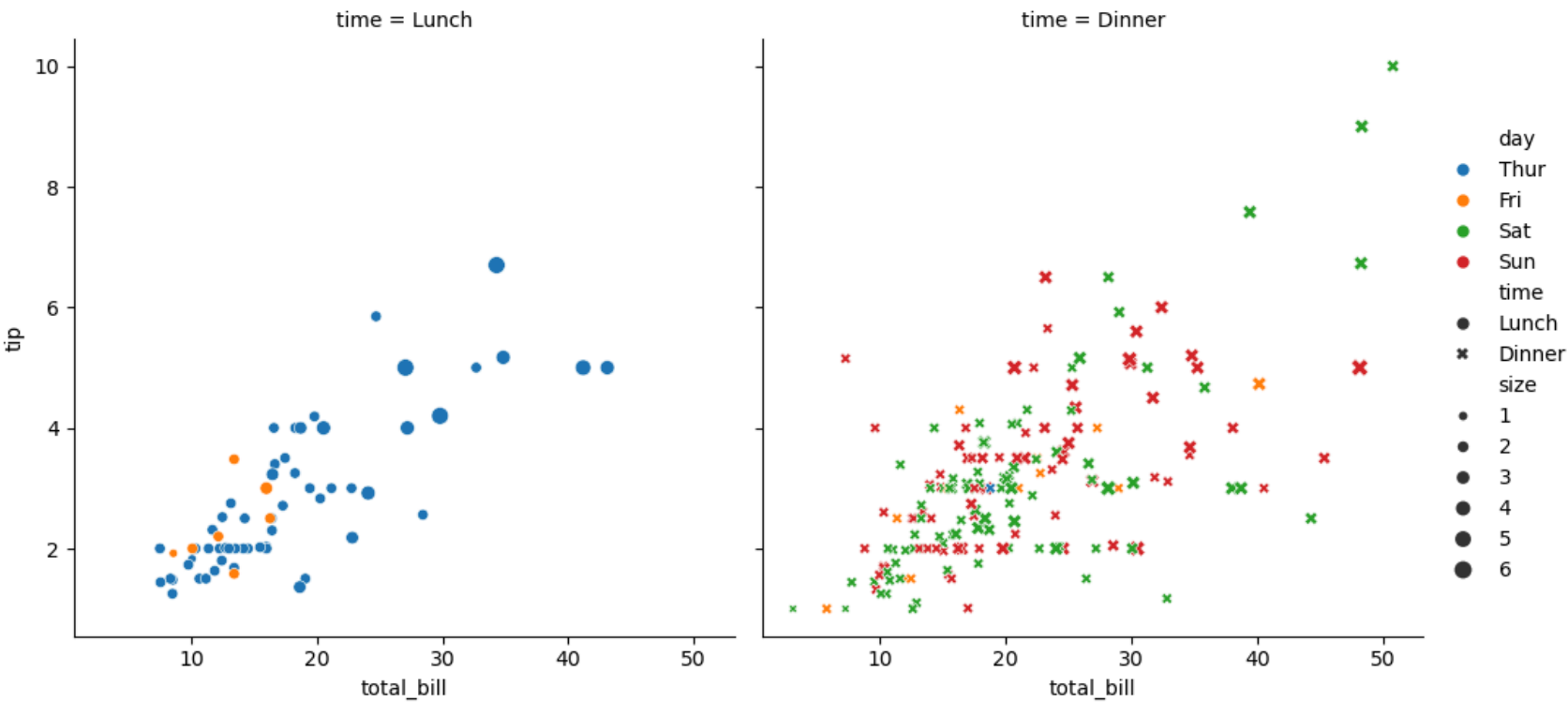


## Usando col = 'day' & kind = 'scatter'

```
In [ ]: #Grafico replot
sns.relplot(data=tips,x='total_bill',y='tip',hue='day',style='time',size='size',markers=marker,col='day',kind='scatter')
plt.show()
```



```
In [ ]: #Grafico replot
sns.relplot(data=tips,x='total_bill',y='tip',hue='day',style='time',size='size',col='time',kind='scatter')
plt.show()
```



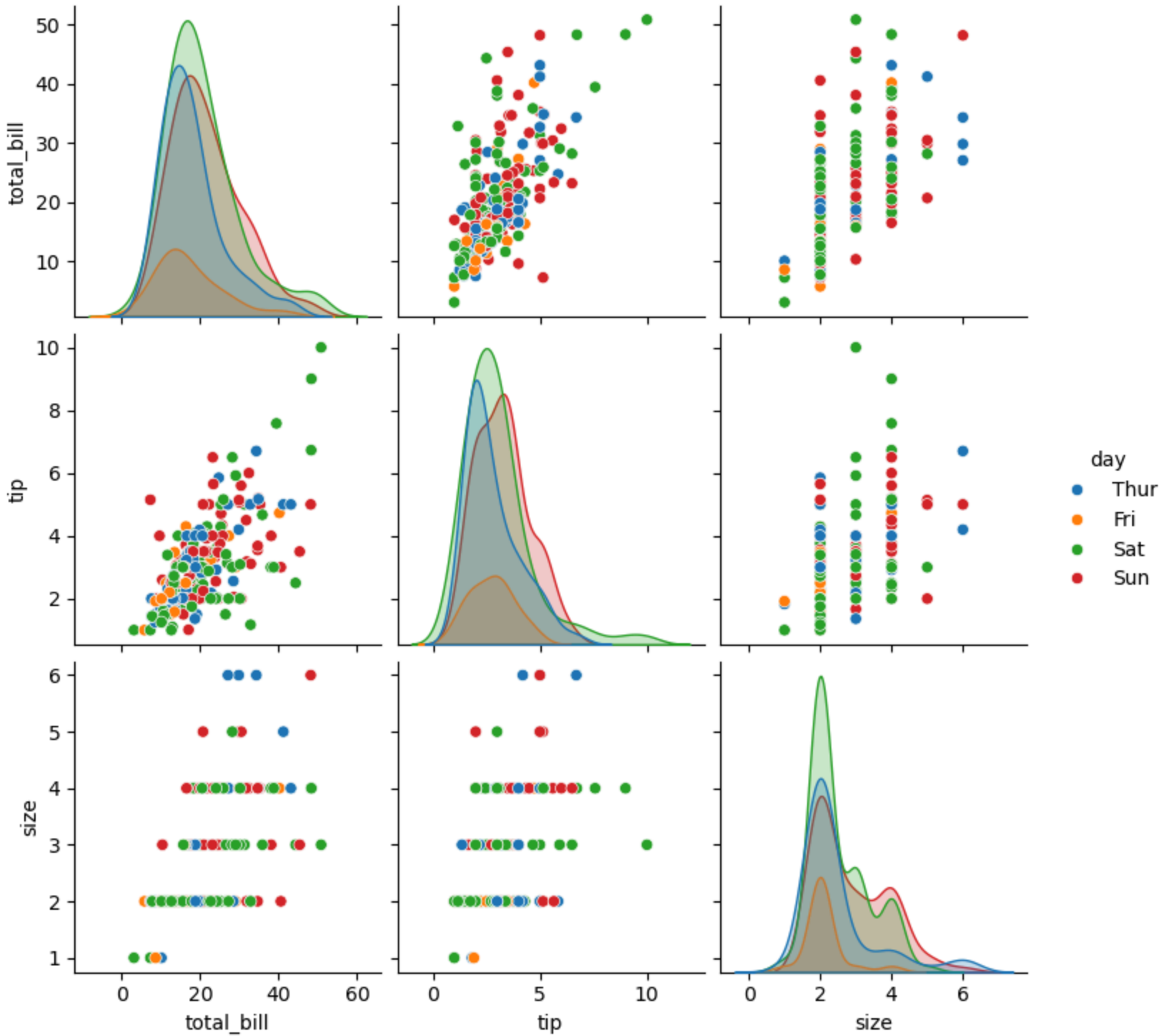
Extra:

Aporte de Carlos Felipe Saldarriaga Bejarano

También podemos usar pairplot, te permite ver comparativas de pares de variables y analizar su comportamiento en un solo gráfico.

```
sns.pairplot(data=tips,hue='day')
plt.show()
```

```
In [ ]: sns.pairplot(data=tips,hue='day')
plt.show()
```



Aporte de Carlos Castillo

No se porque hasta ahora no se menciona pero hay un parametro llamado "ax" para poder usar seaborn en subplots.





## Referencias:

- [Relplot](#)