Data Analytics - Challenge

The goal of this challenge is to analyze a restaurant invoices. Some celles are already implemented, you just need to **run** them. Some other cells need you to write some code.

Start the challenge by running the two following cells:

In [1]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
tips_df = pd.read_csv("https://raw.githubusercontent.com/mwaskom/seaborn-da
```

❓ Display the 10 first rows of the dataset (no need to sort)
🙈 Reveal solution

In [3]:

```
# Your code here
tips_df.head(10)
```

Out[3]:

	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
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
5	25.29	4.71	Male	No	Sun	Dinner	4
6	8.77	2.00	Male	No	Sun	Dinner	2
7	26.88	3.12	Male	No	Sun	Dinner	4
8	15.04	1.96	Male	No	Sun	Dinner	2
9	14.78	3.23	Male	No	Sun	Dinner	2

❓ How many days per week is the restaurant open?
🙈 Reveal solution

In [4]:

```
# Your code here
day_work = tips_df['day'].unique().tolist()
print("il y a : ",len(day_work)," jours/semaine ou le restaurant est ouvert
```

```
il y a :  4  jours/semaine ou le restaurant est ouvert, qui sont :  ['Su
n', 'Sat', 'Thur', 'Fri']
```

❓ What day of the week is there more bills? Plot this with a Seaborn Countplot.
🙈 Reveal solution

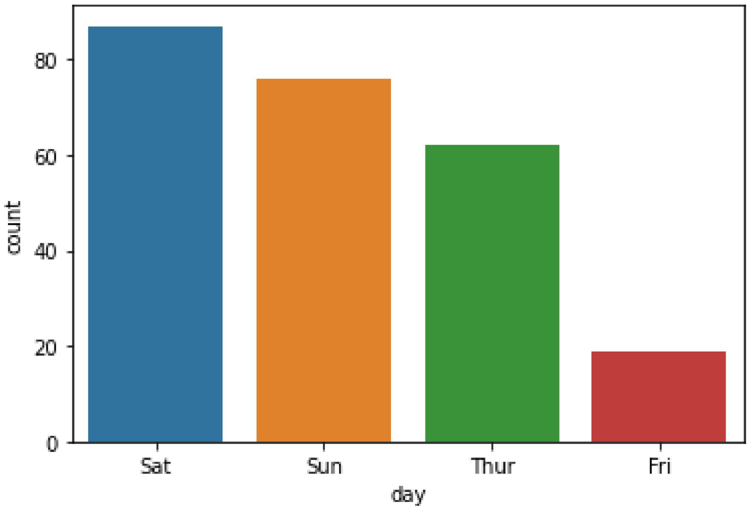
In [5]:
```
# Your code here
day_more_total_bill = tips_df['day'].value_counts()
day_more_total_bill
```

Out[5]:
```
Sat     87
Sun     76
Thur    62
Fri     19
Name: day, dtype: int64
```

c'est le samedi

In [6]:
```
# Your plot here
order = day_more_total_bill.index
sns.countplot(tips_df, x='day',order=order)
```

Out[6]:
```
<AxesSubplot:xlabel='day', ylabel='count'>
```

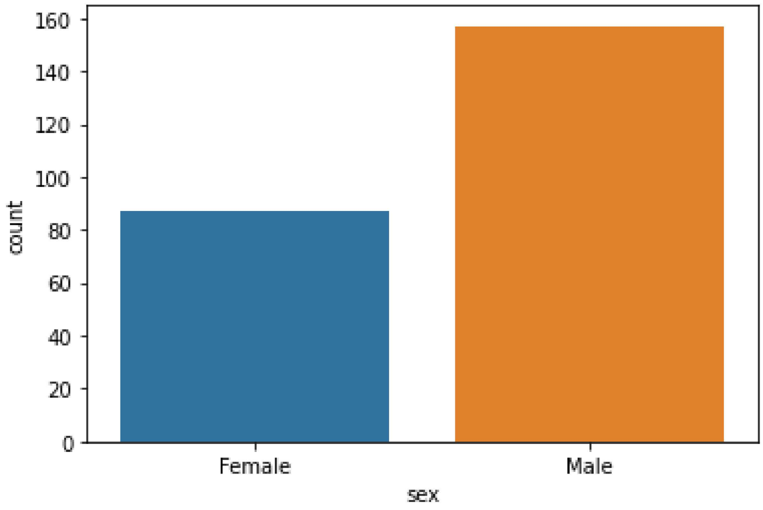


❓ Try to do some other countplots, varying `x` with one of the categorical column (`sex` , `smoker` , `time`)

In [7]:
```
# Your first plot here
# To add a cell, you can go in the menu and do Insert > Insert cell below
```

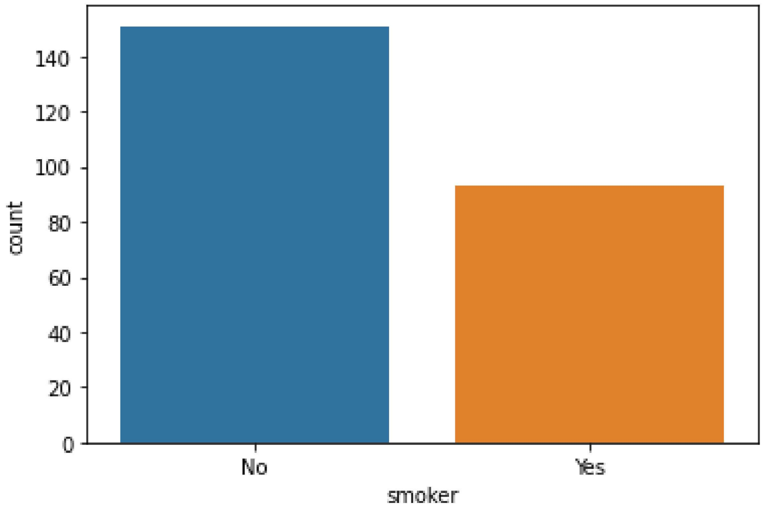
```
In [8]: sns.countplot(tips_df, x='sex')
```

```
Out[8]: <AxesSubplot:xlabel='sex', ylabel='count'>
```



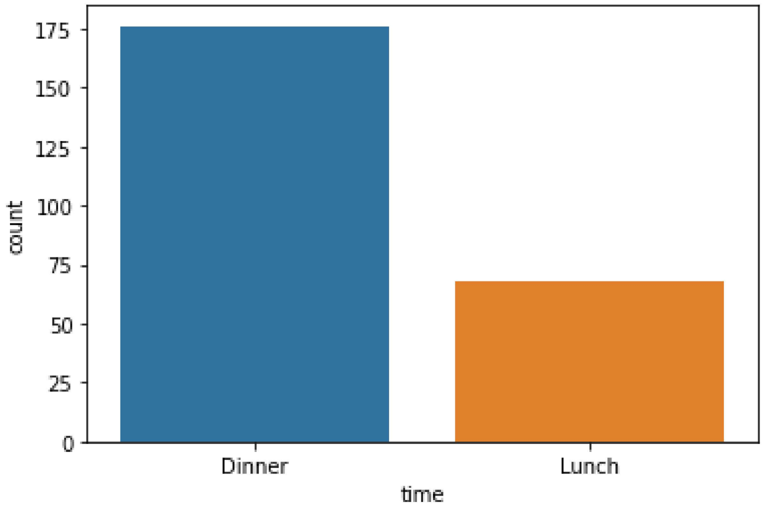
```
In [9]: sns.countplot(tips_df, x='smoker')
```

```
Out[9]: <AxesSubplot:xlabel='smoker', ylabel='count'>
```



In [10]:
```
sns.countplot(tips_df, x='time')
```

Out[10]:
```
<AxesSubplot:xlabel='time', ylabel='count'>
```



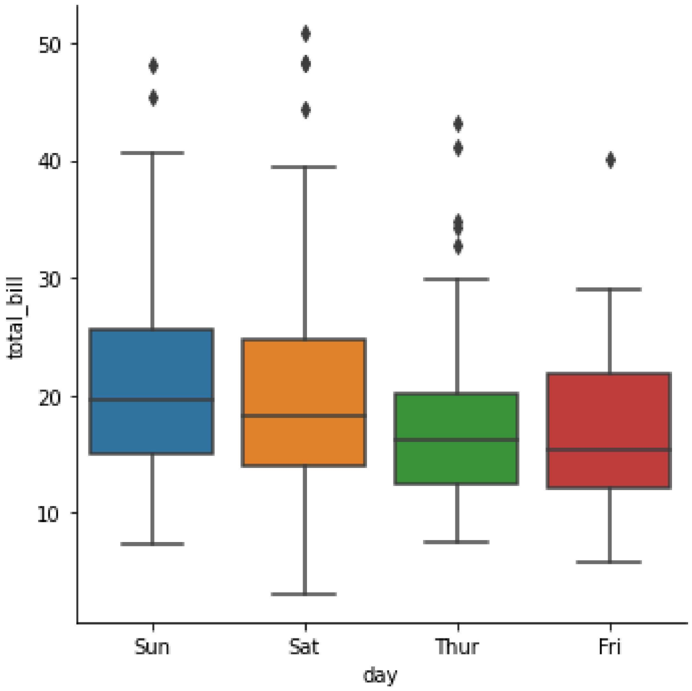
❓ Let's plot the distribution of `total_bill` based on a given category. Start with `day` :

```
sns.catplot(data=tips_df, x='day', y='total_bill', kind="box")
```

1. Change the value of `x` with one of the categorical column of the dataset and the value of `kind` (`"bar"` , `"box"` , `"violin"` , `"boxen"`)
2. Change the value of `y` with one of the numerical column of the dataset

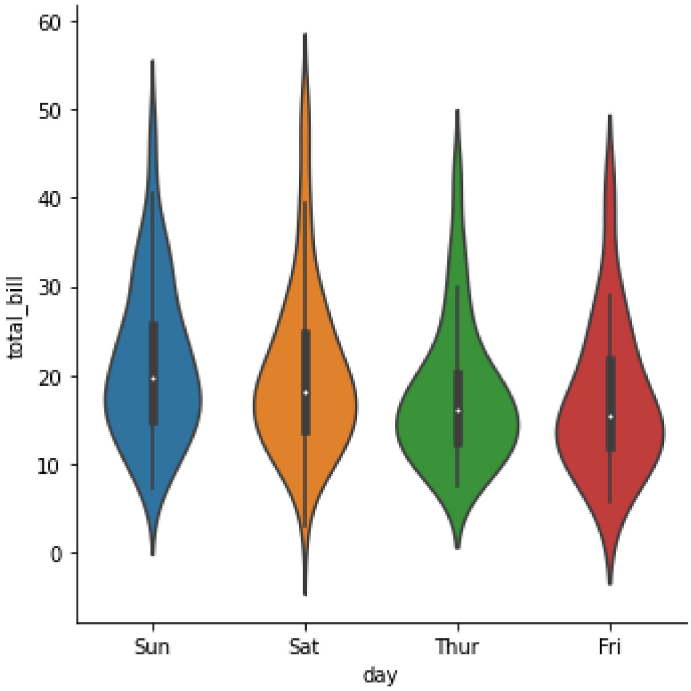
In [11]:
```
# Your experiments here
sns.catplot(data=tips_df, x='day', y='total_bill', kind="box")
```

Out[11]:
```
<seaborn.axisgrid.FacetGrid at 0x7f41907128d0>
```



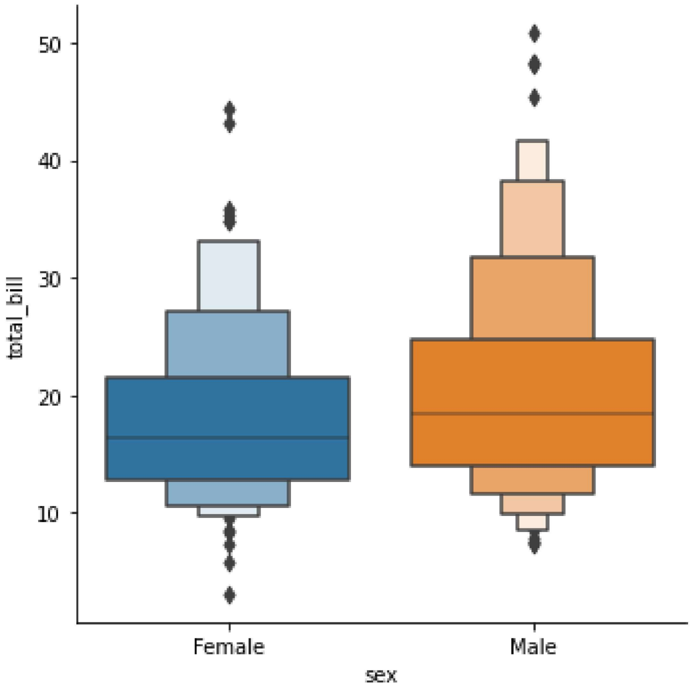
```
In [12]: sns.catplot(data=tips_df, x='day', y='total_bill', kind="violin")
```

```
Out[12]: <seaborn.axisgrid.FacetGrid at 0x7f41906e5a90>
```



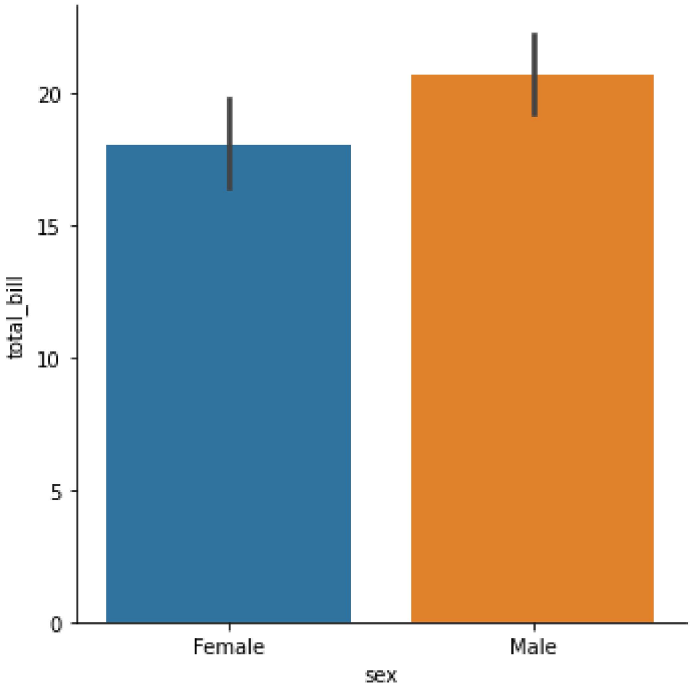
```
In [13]: sns.catplot(data=tips_df, x='sex', y='total_bill', kind="boxen")
```

```
Out[13]: <seaborn.axisgrid.FacetGrid at 0x7f4190663b38>
```



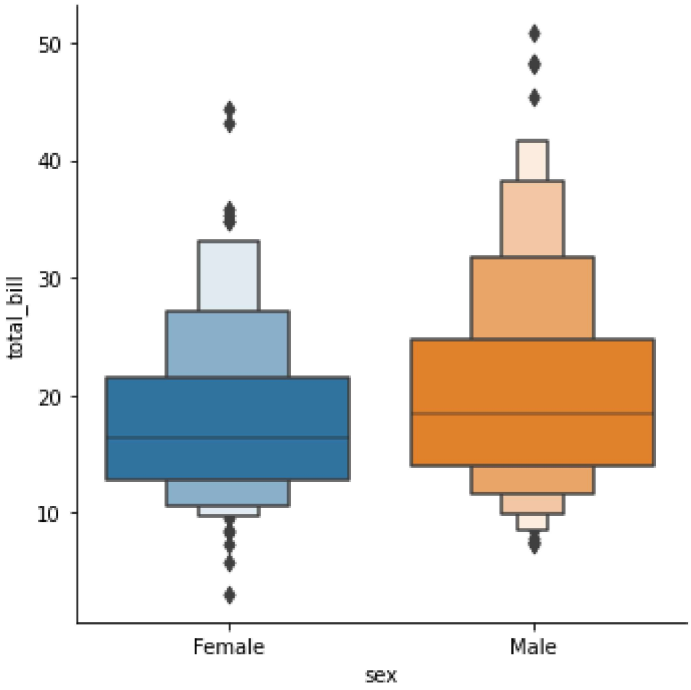
```
In [14]: sns.catplot(data=tips_df, x='sex', y='total_bill', kind="bar")
```

```
Out[14]: <seaborn.axisgrid.FacetGrid at 0x7f4190789898>
```



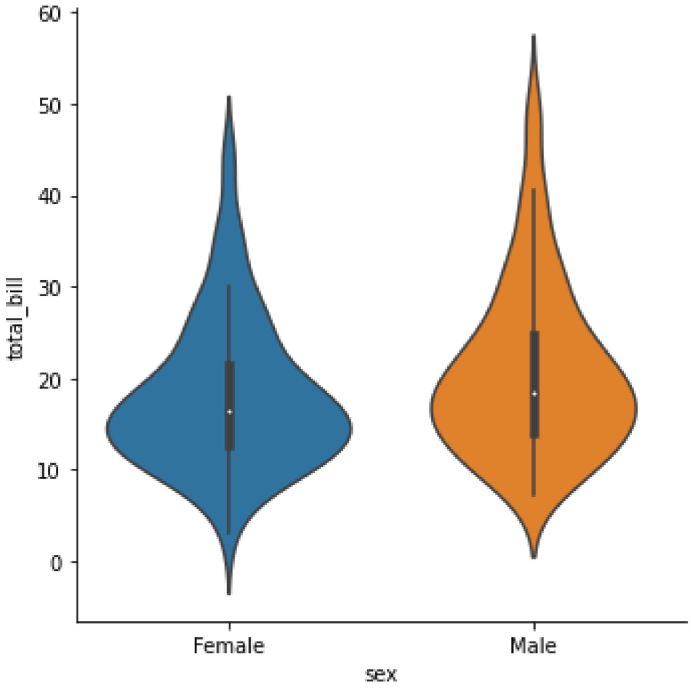
```
In [15]: sns.catplot(data=tips_df, x='sex', y='total_bill', kind="boxen")
```

```
Out[15]: <seaborn.axisgrid.FacetGrid at 0x7f419055d4a8>
```



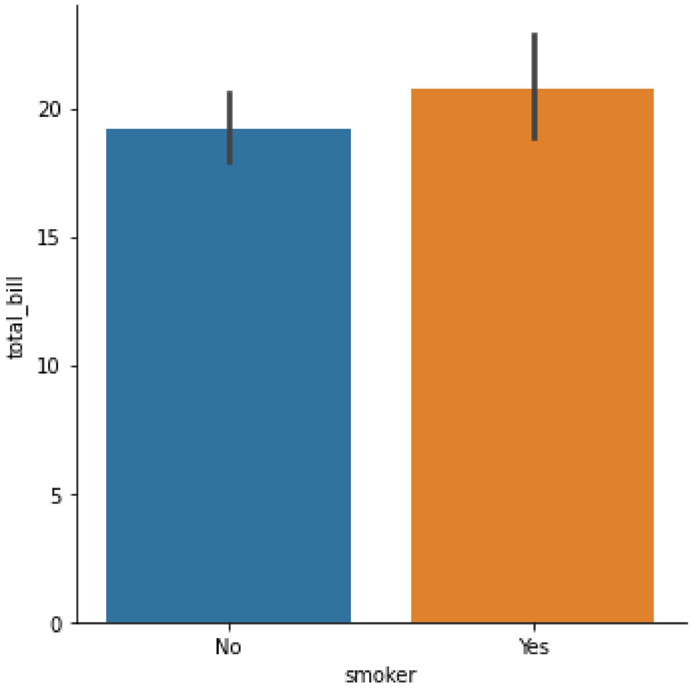
```
In [16]: sns.catplot(data=tips_df, x='sex', y='total_bill', kind="violin")
```

```
Out[16]: <seaborn.axisgrid.FacetGrid at 0x7f41904c6f60>
```



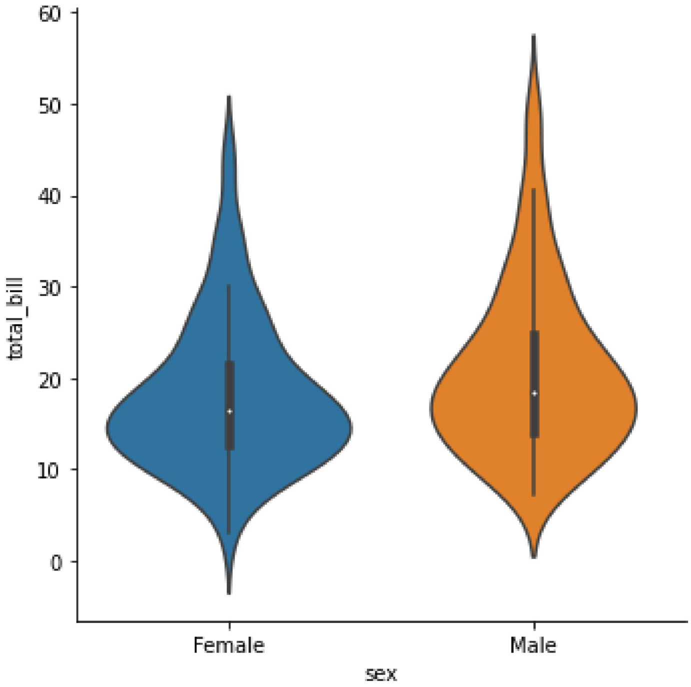
```
In [17]: sns.catplot(data=tips_df, x='smoker', y='total_bill', kind="bar")
```

```
Out[17]: <seaborn.axisgrid.FacetGrid at 0x7f41904035f8>
```



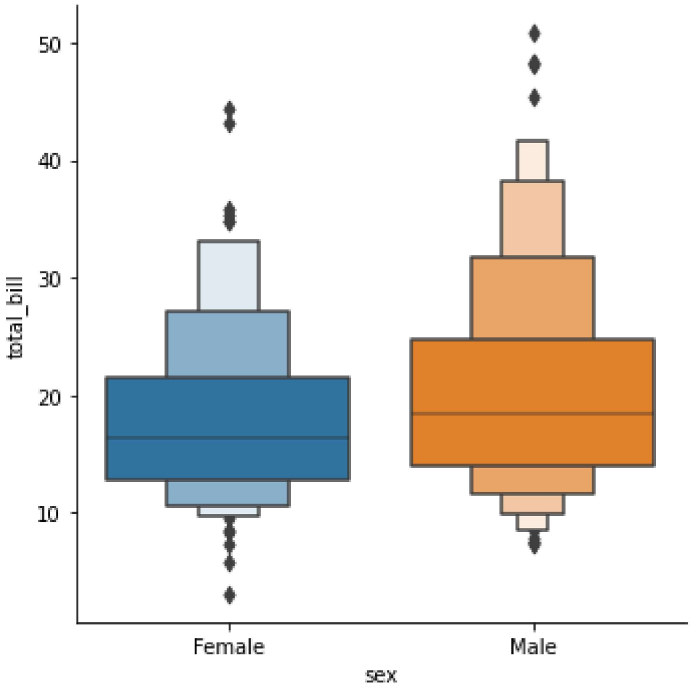
In [18]:
```
sns.catplot(data=tips_df, x='sex', y='total_bill', kind="violin")
```

Out[18]:
```
<seaborn.axisgrid.FacetGrid at 0x7f419040c550>
```



In [19]:
```
sns.catplot(data=tips_df, x='sex', y='total_bill', kind="boxen")
```

Out[19]:
```
<seaborn.axisgrid.FacetGrid at 0x7f4190375ef0>
```

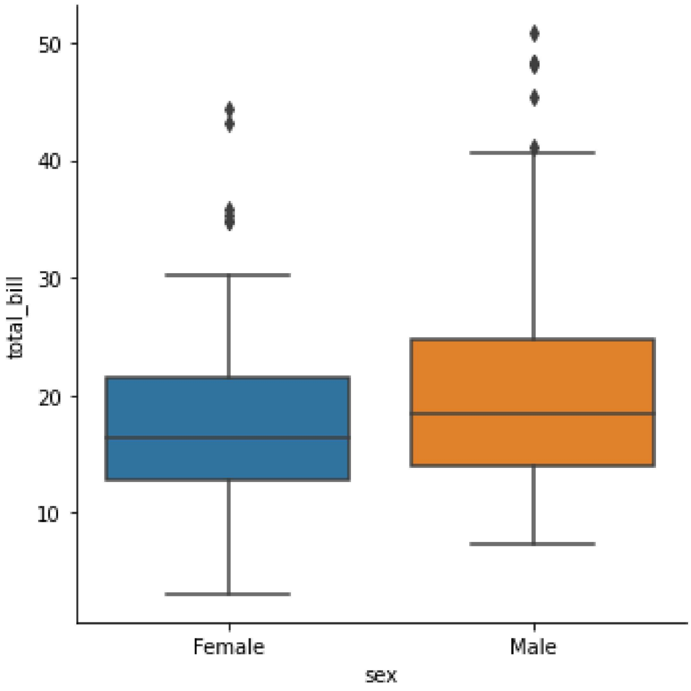


In [20]:

```
sns.catplot(data=tips_df, x='sex', y='total_bill', kind="box")
```

Out[20]:

```
<seaborn.axisgrid.FacetGrid at 0x7f4190309ef0>
```



❓ Let's use `seaborn.FacetGrid` [(https://seaborn.pydata.org/generated/seaborn.FacetGrid.html)](https://seaborn.pydata.org/generated/seaborn.FacetGrid.html)

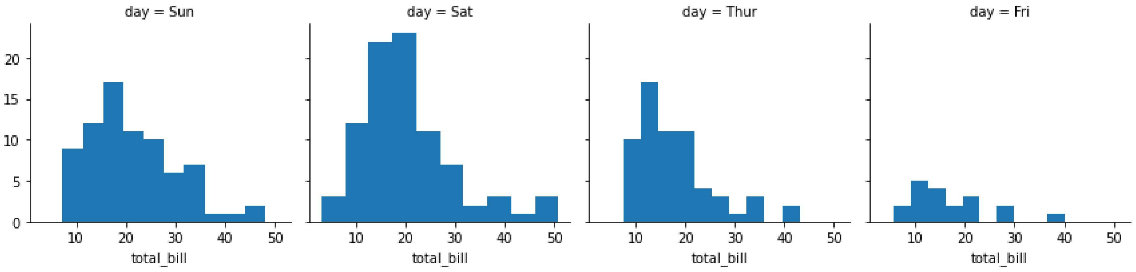
1. Run the cell below. What do you observe?
2. Change `col` in the first line with another column (e.g. `"time"`). Run the cell again. What do you observe?

In [21]:

```
g = sns.FacetGrid(tips_df, col="day")
g.map(plt.hist, "total_bill")
```

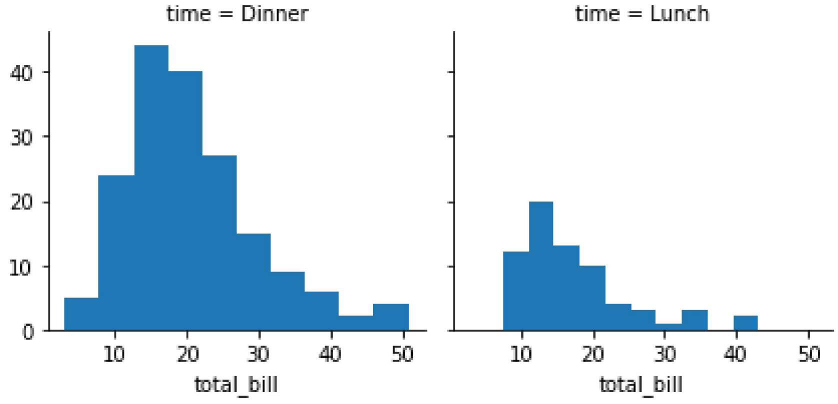
Out[21]:

```
<seaborn.axisgrid.FacetGrid at 0x7f41902f5e80>
```



In [22]:
```
g = sns.FacetGrid(tips_df, col="time")
g.map(plt.hist, "total_bill")
```

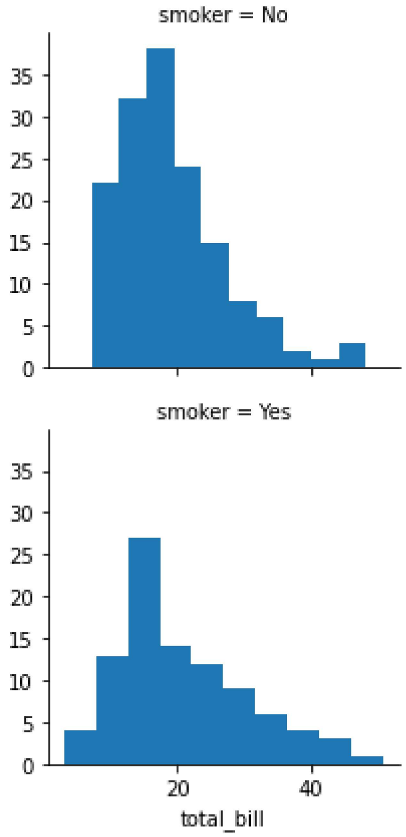
Out[22]:
```
<seaborn.axisgrid.FacetGrid at 0x7f418ffd7860>
```



❓ Let's continue with FacetGrid and add a `row="smoker"` parameter. How many cells do you get in the plot?
🙈 Reveal solution

In [23]:
```
# Your code here
g = sns.FacetGrid(tips_df, row="smoker")
g.map(plt.hist, "total_bill")
```

Out[23]:
```
<seaborn.axisgrid.FacetGrid at 0x7f418ffd7780>
```



Correlation

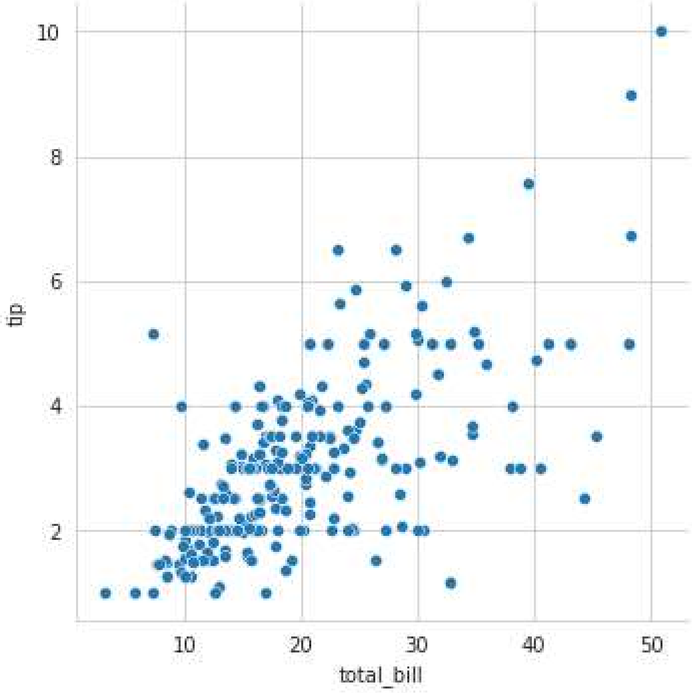
Let's start looking for correlation between columns in the dataset.

❓ What is your intuition about the relationship between the columns `tip` and `total_bill` ?

Je pense qu'il y a une correlation positive entre les deux variables

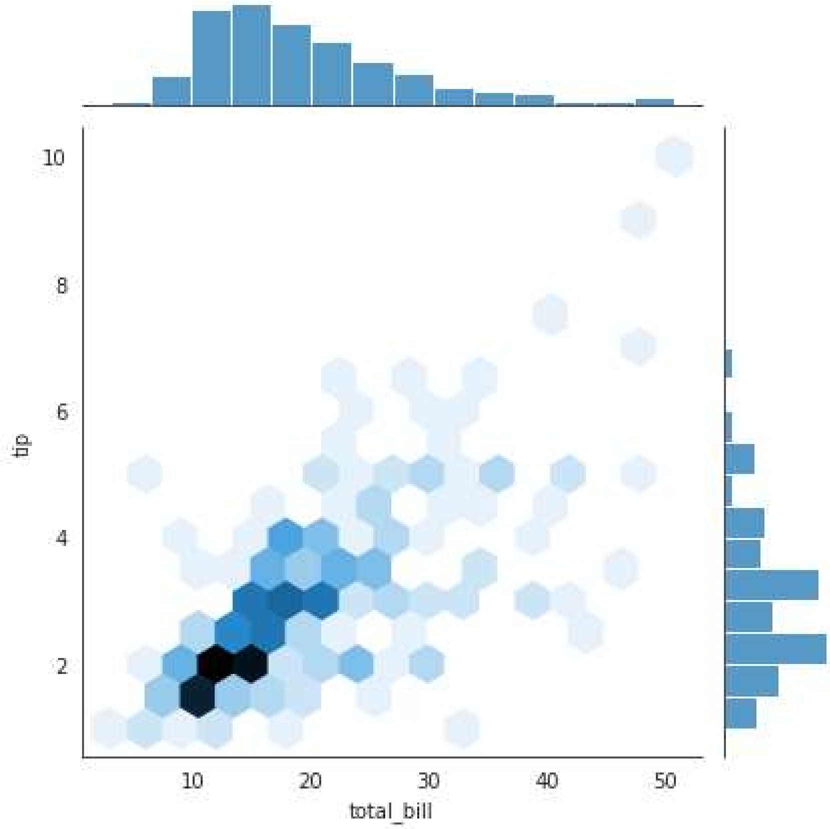
❓ Let's look at the data to see if our intuition is correct. We will do a **scatterplot** with `x` being `total_bill` and `y` the tip.

In [24]:
```
with sns.axes_style(style="whitegrid"):
    sns.relplot(x="total_bill", y="tip", data=tips_df)
```



❓ Another way of looking at this data is to use a `seaborn.jointplot` [(https://seaborn.pydata.org/generated/seaborn.jointplot.html)](https://seaborn.pydata.org/generated/seaborn.jointplot.html).

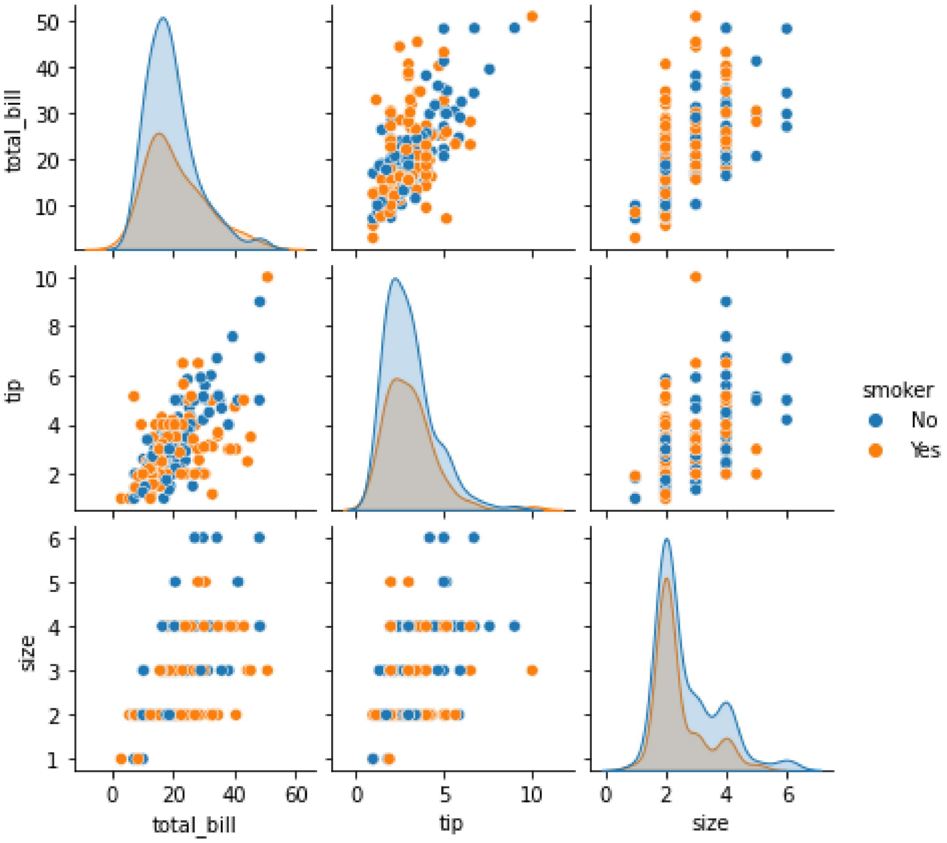
In [25]:
```
with sns.axes_style("white"):
    sns.jointplot(x="total_bill", y="tip", kind="hex", data=tips_df)
```



❓ A very useful tool to **identify** correlations is the `seaborn.pairplot` [(https://seaborn.pydata.org/generated/seaborn.pairplot.html)](https://seaborn.pydata.org/generated/seaborn.pairplot.html):

```
In [26]: sns.pairplot(tips_df, height=2, hue="smoker")
```

```
Out[26]: <seaborn.axisgrid.PairGrid at 0x7f418fc5cd30>
```

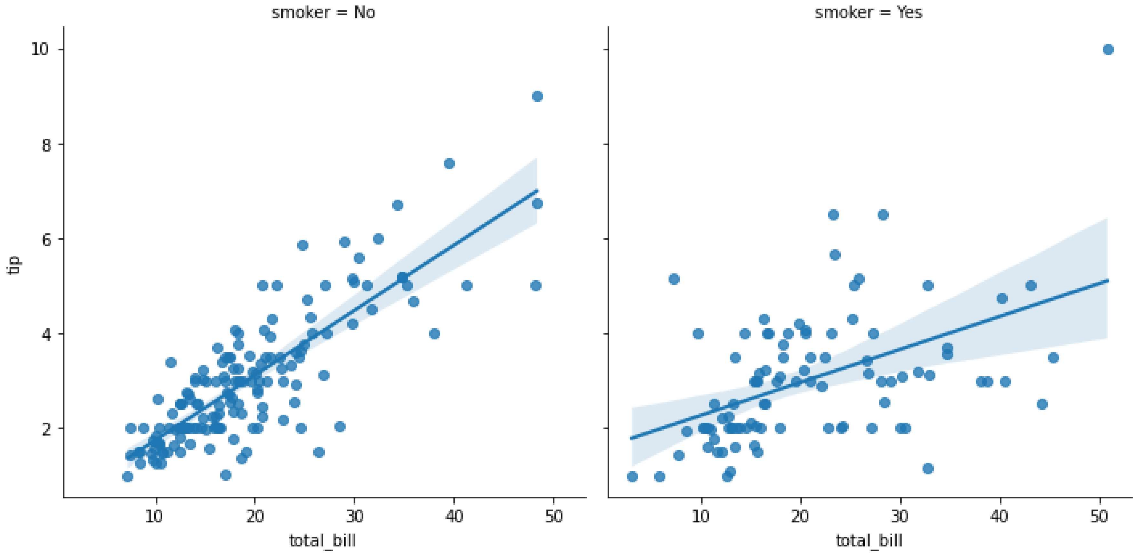


Regression

We are not doing Machine Learning yet but we can use `seaborn.lmplot` [(https://seaborn.pydata.org/generated/seaborn.lmplot.html)](https://seaborn.pydata.org/generated/seaborn.lmplot.html) to graphically read a linear correlation between two columns:

```
In [27]: sns.lmplot(x="total_bill", y="tip", col="smoker", data=tips_df)
```

```
Out[27]: <seaborn.axisgrid.FacetGrid at 0x7f4190541b70>
```



Good job!

Save your notebook, go back to the **Le Wagon - Learn** platform to upload your progress. A quiz awaits you!