AN7914 Week 06 Python

March 6, 2024

1 Week 6 Python

1.1 Introduction to Data Visualisation

There are lots of libraries for doing data visualisation in Python.

1. `Matplotlib`- the most versatile and customizable
2. `Seaborn`- easy for beginners
3. `Plotnine`- library that is very close R's ggplot2 (probably not updated regularly)
4. `Lets-plot` - Another library that is very close to ggplot2 (it is maintained regularly)

I would suggest you to learn Matplotlib later at some point. But in this class we will do `Seaborn`

[1]:
```
import pandas as pd
#import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
#Apply default theme
sns.set_theme()
```

[2]:
```
#df_tips = pd.read_csv('https://raw.githubusercontent.com/mwaskom/seaborn-data/
 ↪master/tips.csv')
df_tips=sns.load_dataset('tips')
```

[3]:
```
df_tips
```

[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
..	…	…	…	…	…	…	…
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

```
[244 rows x 7 columns]
```

[4]:
```
df_tips.head()
```

[4]:

	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

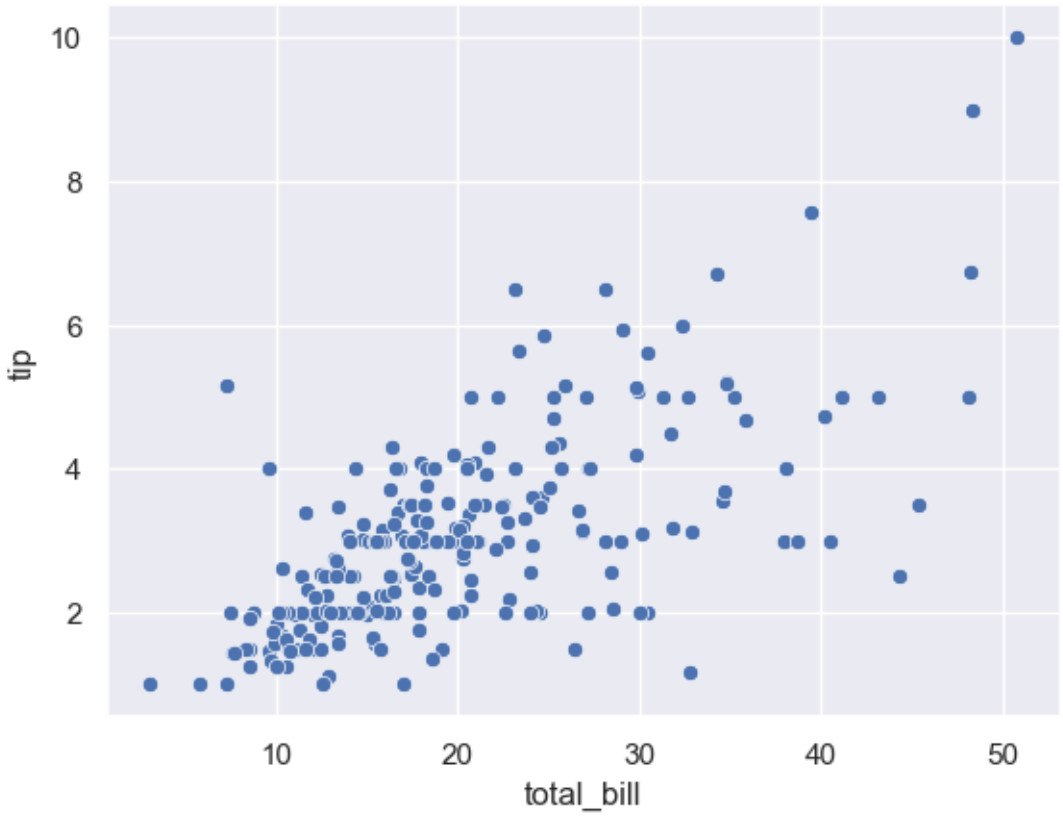
[5]:
```
df_tips.describe()
```

[5]:

	total_bill	tip	size
count	244.000000	244.000000	244.000000
mean	19.785943	2.998279	2.569672
std	8.902412	1.383638	0.951100
min	3.070000	1.000000	1.000000
25%	13.347500	2.000000	2.000000
50%	17.795000	2.900000	2.000000
75%	24.127500	3.562500	3.000000
max	50.810000	10.000000	6.000000

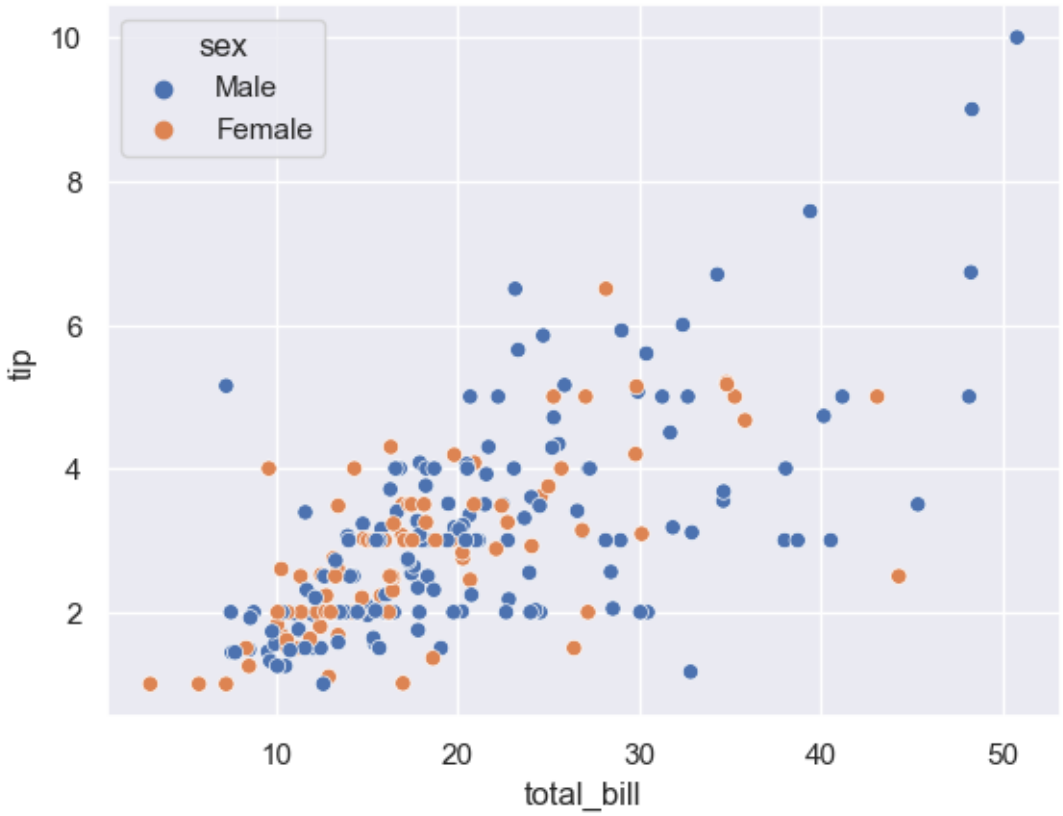
[6]:
```
sns.scatterplot(data=df_tips, x=df_tips['total_bill'], y=df_tips['tip'])
```

[6]:
```
<AxesSubplot: xlabel='total_bill', ylabel='tip'>
```



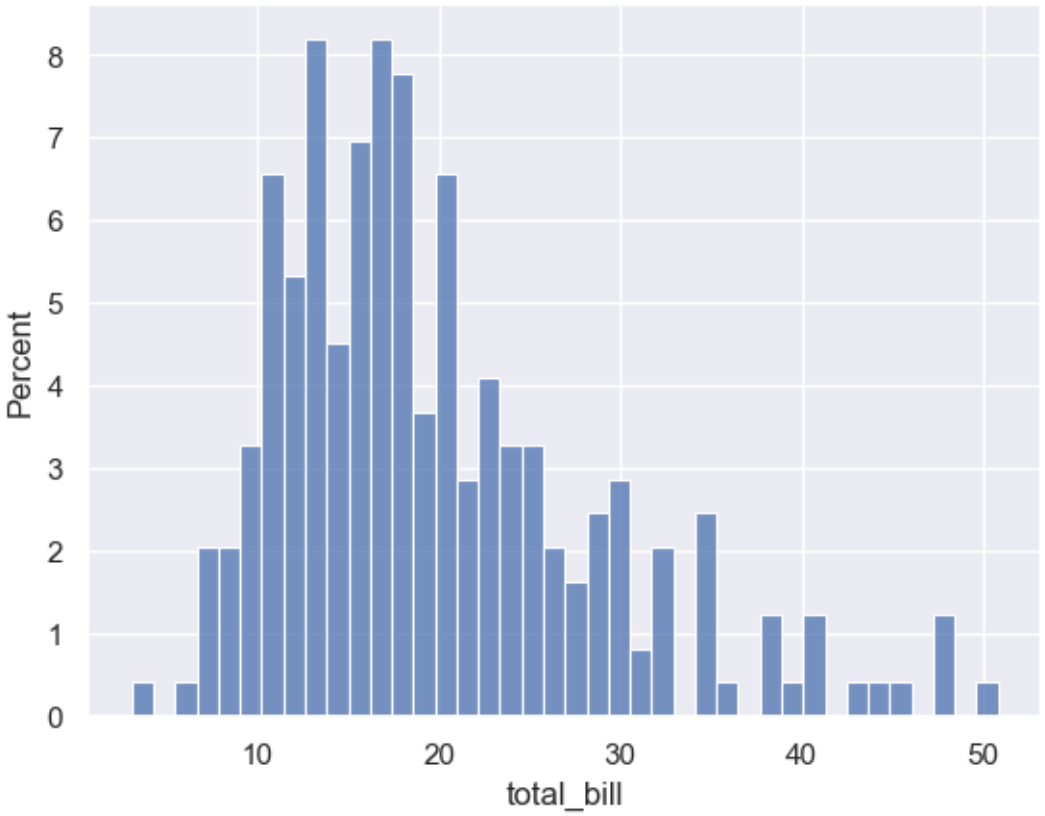
[7]:
```
sns.scatterplot(data=df_tips, x=df_tips['total_bill'], y=df_tips['tip'],␣
↪hue=df_tips['sex'])
```

[7]: <AxesSubplot: xlabel='total_bill', ylabel='tip'>

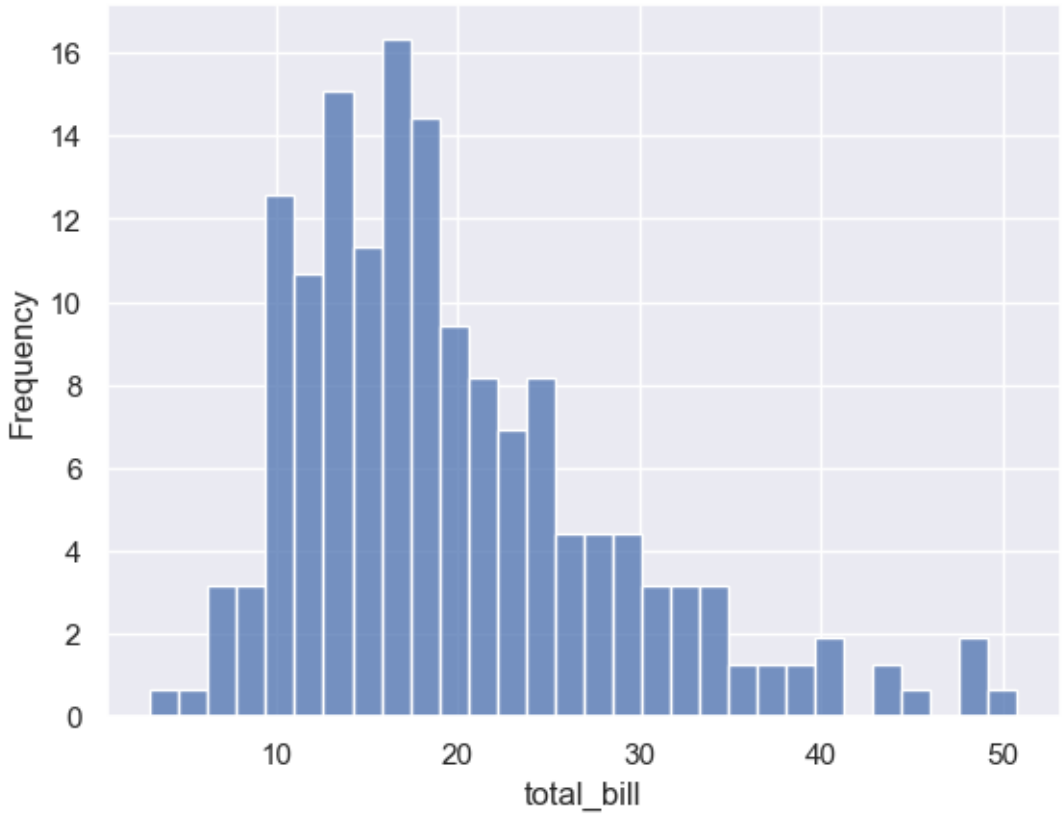


[8]:
```
#import matplotlib as mpl
#import matplotlib.pyplot as plt
#%matplotlib inline
```

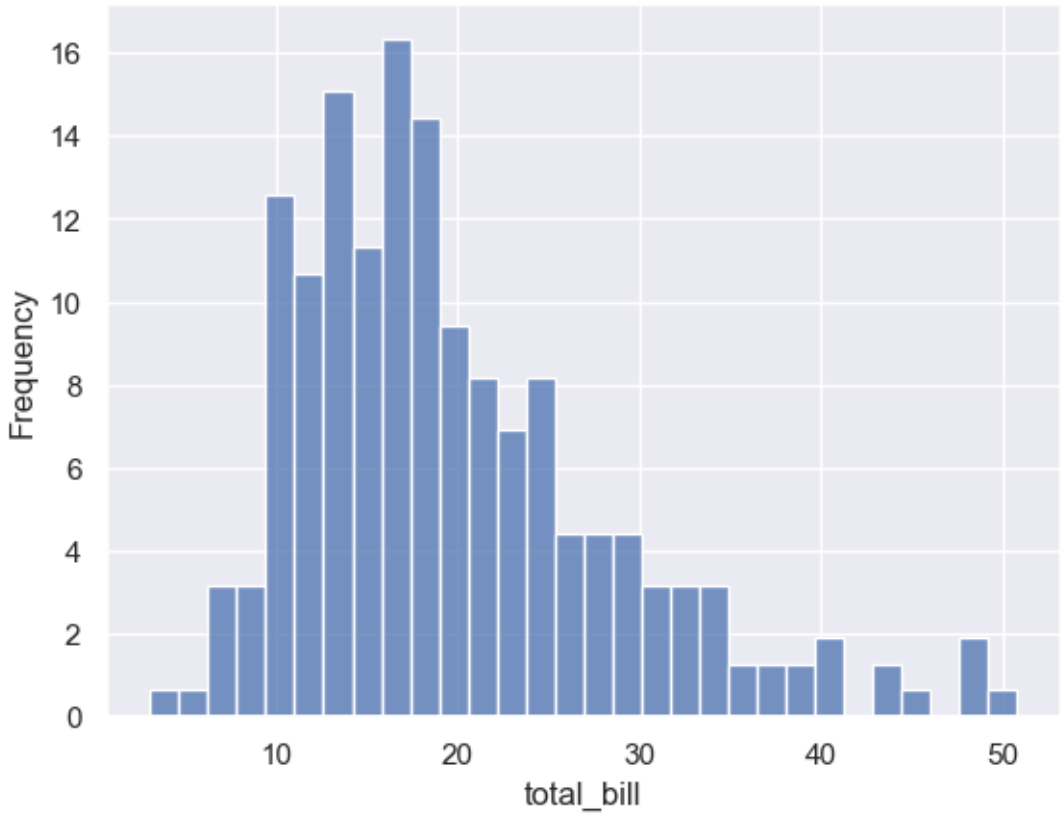
[22]:
```
sns.histplot(data=df_tips, x=df_tips['total_bill'], bins=40, stat="percent")
#plt.savefig("hist_example.png")
plt.show()
```



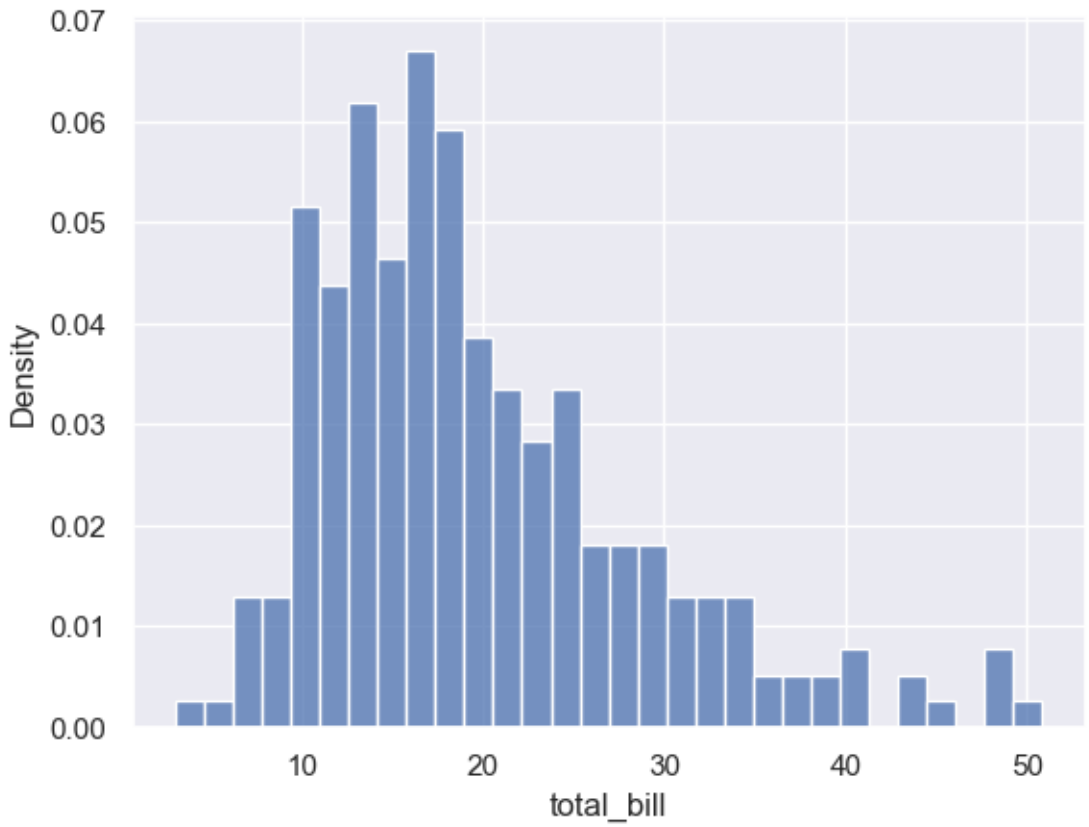
[10]:
```
sns.histplot(data=df_tips, x=df_tips['total_bill'], bins=30, stat='frequency')
plt.savefig("hist_example1.png")
plt.show()
```



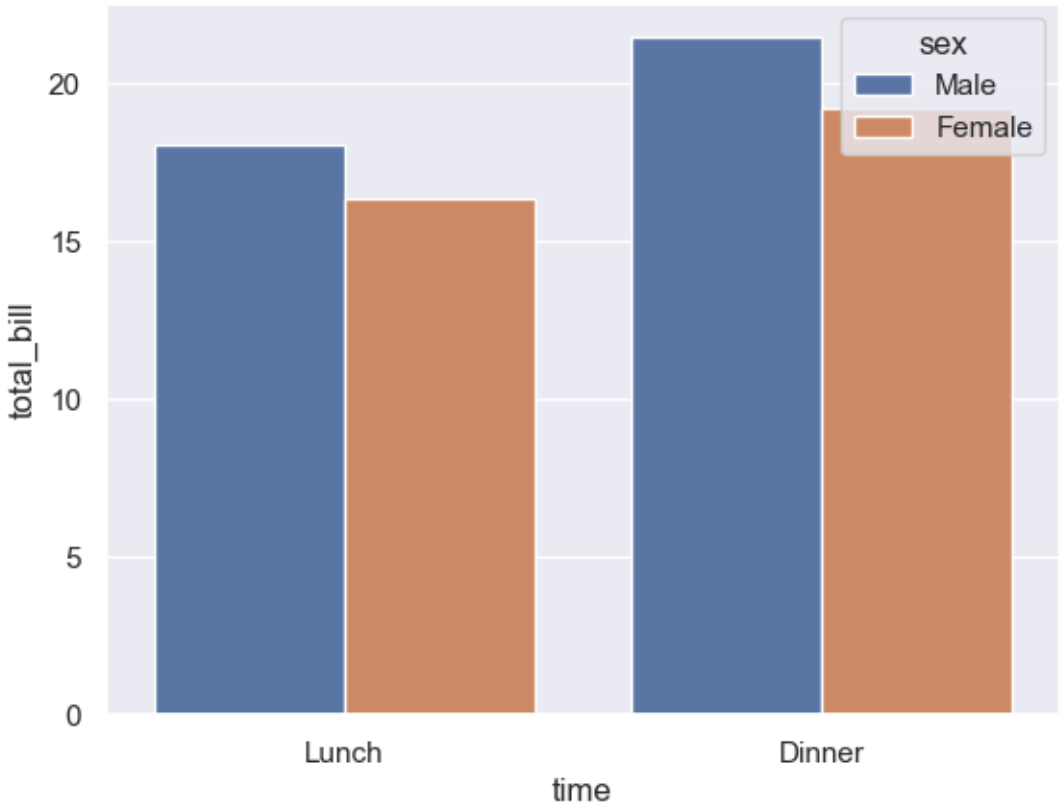
[11]:
```
sns.histplot(data=df_tips, x=df_tips['total_bill'], bins=30, stat='frequency')
plt.savefig("hist_example.png")
plt.show()
```



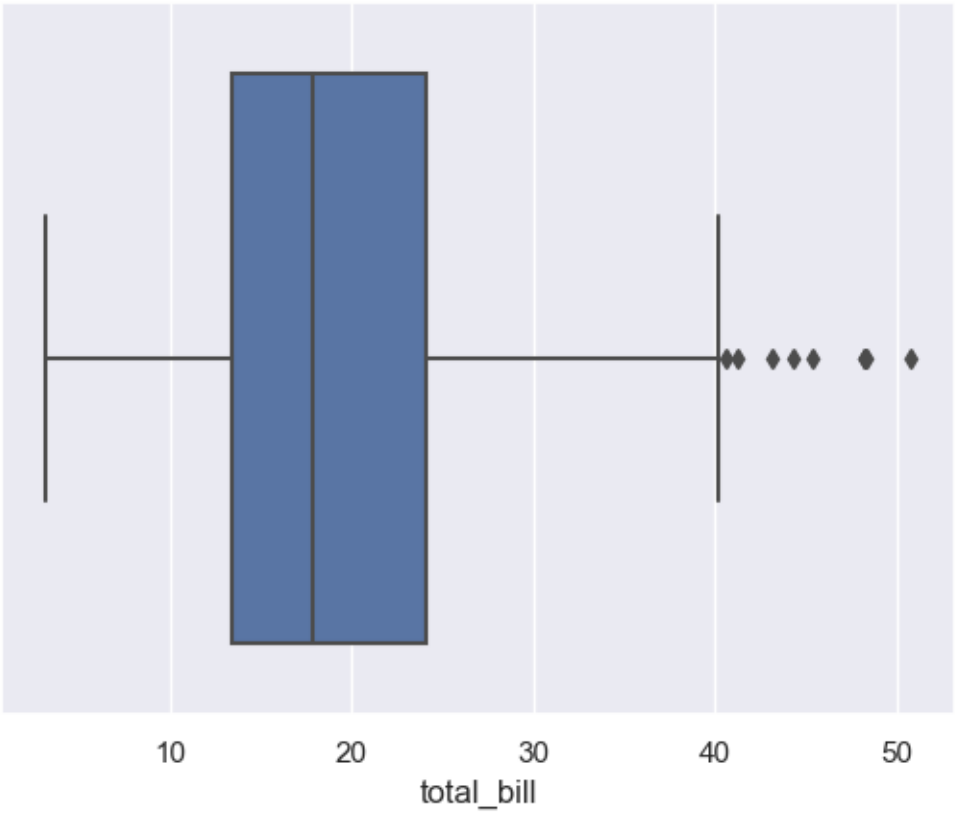
[12]:
```
sns.histplot(data=df_tips, x=df_tips['total_bill'], bins=30, stat='density')
plt.savefig("hist_example.png")
plt.show()
```



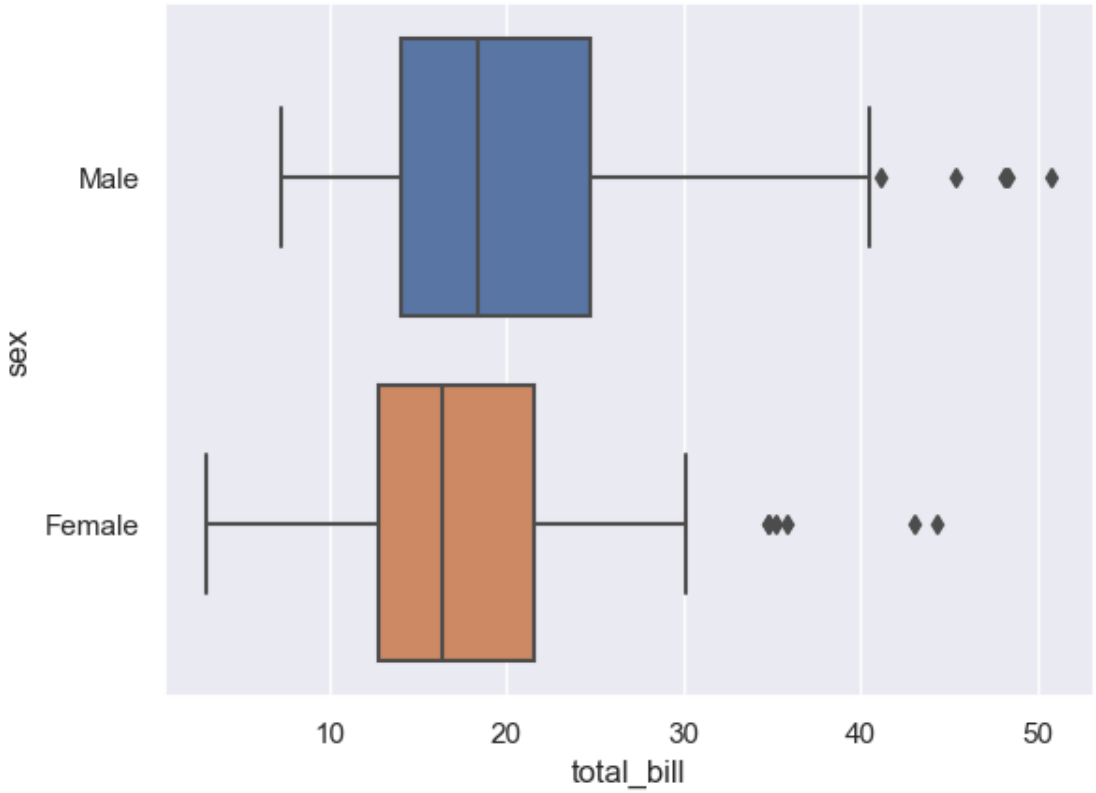
[13]:
```
sns.barplot(data=df_tips, x=df_tips['time'],␣
 ↪y=df_tips['total_bill'],hue=df_tips['sex'], errorbar=None)
plt.show()
```



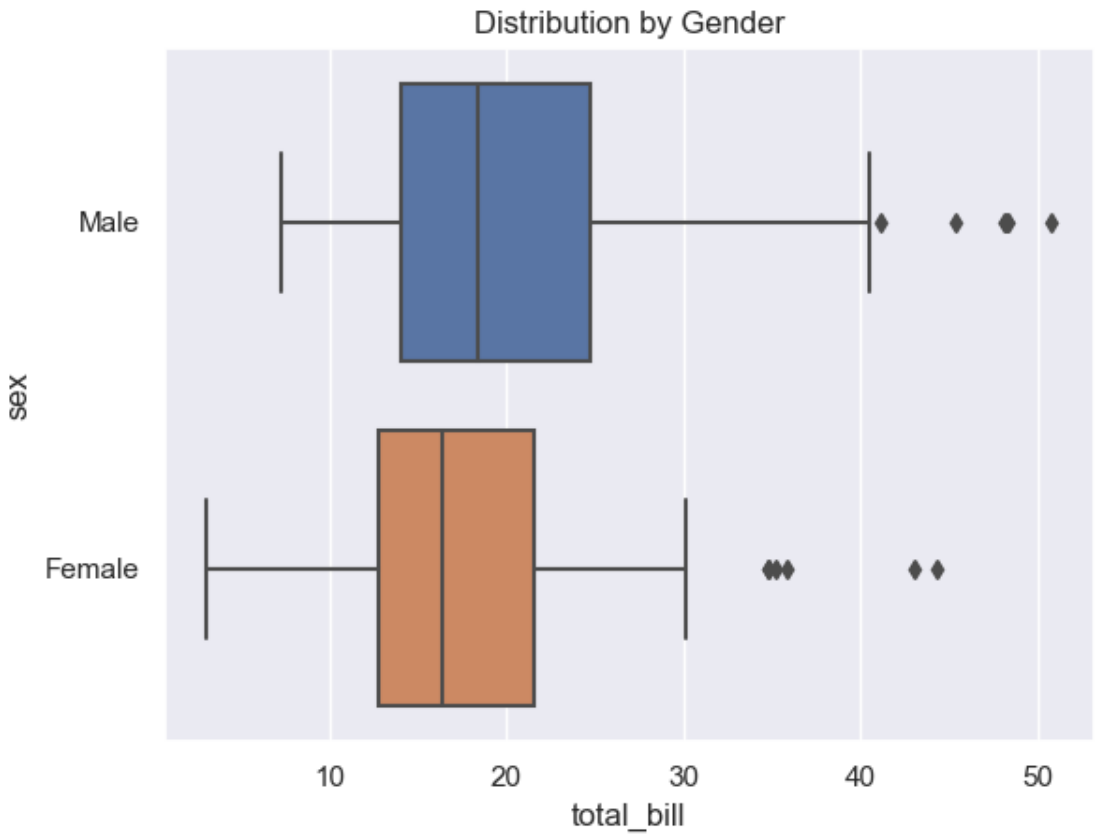
[14]:
```
sns.boxplot(data=df_tips, x=df_tips['total_bill'])
plt.show()
```



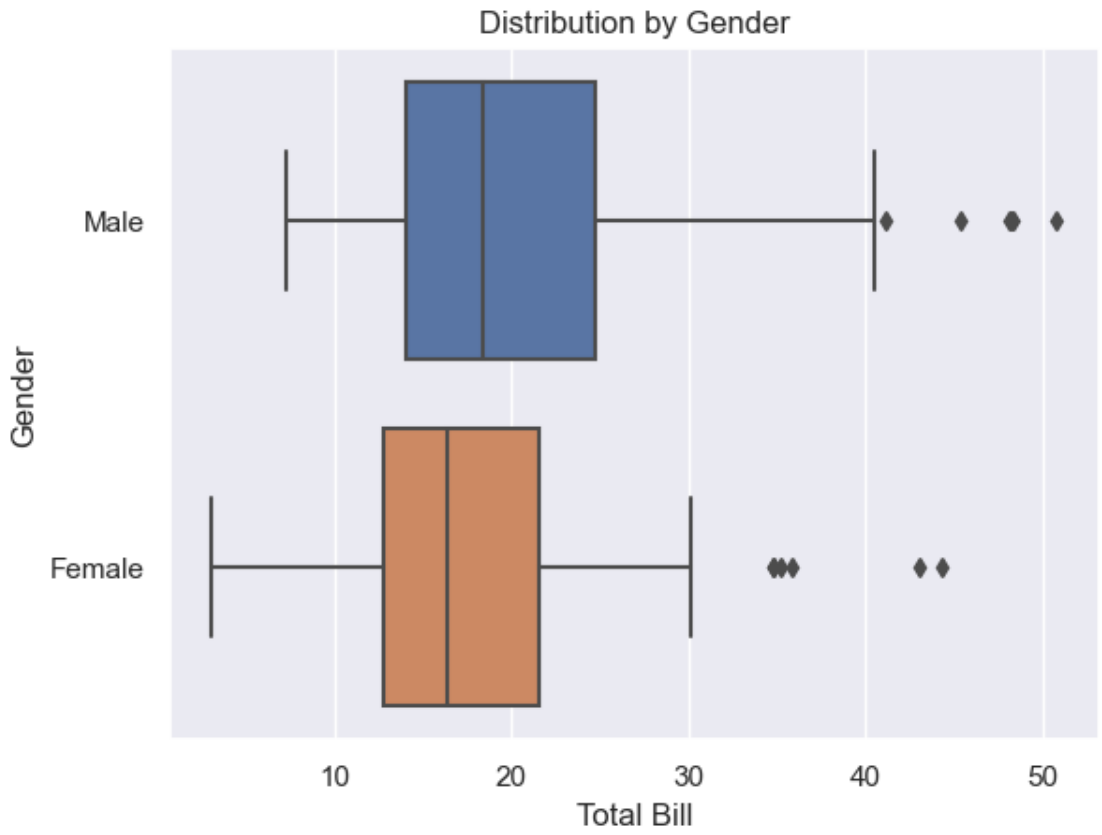
[15]:
```
sns.boxplot(data=df_tips, x=df_tips['total_bill'],y=df_tips['sex'])
plt.show()
```



[16]:
```
sns.boxplot(data=df_tips, x=df_tips['total_bill'],y=df_tips['sex']).
  ↪set_title('Distribution by Gender')
plt.show()
```



[28]:
```
sns.boxplot(data=df_tips, x=df_tips['total_bill'],y=df_tips['sex']).
  ↪set(title='Distribution by Gender',xlabel='Total Bill', ylabel='Gender')
plt.show()
```



[17]:
```
df_flights = sns.load_dataset("flights")
df_flights
```

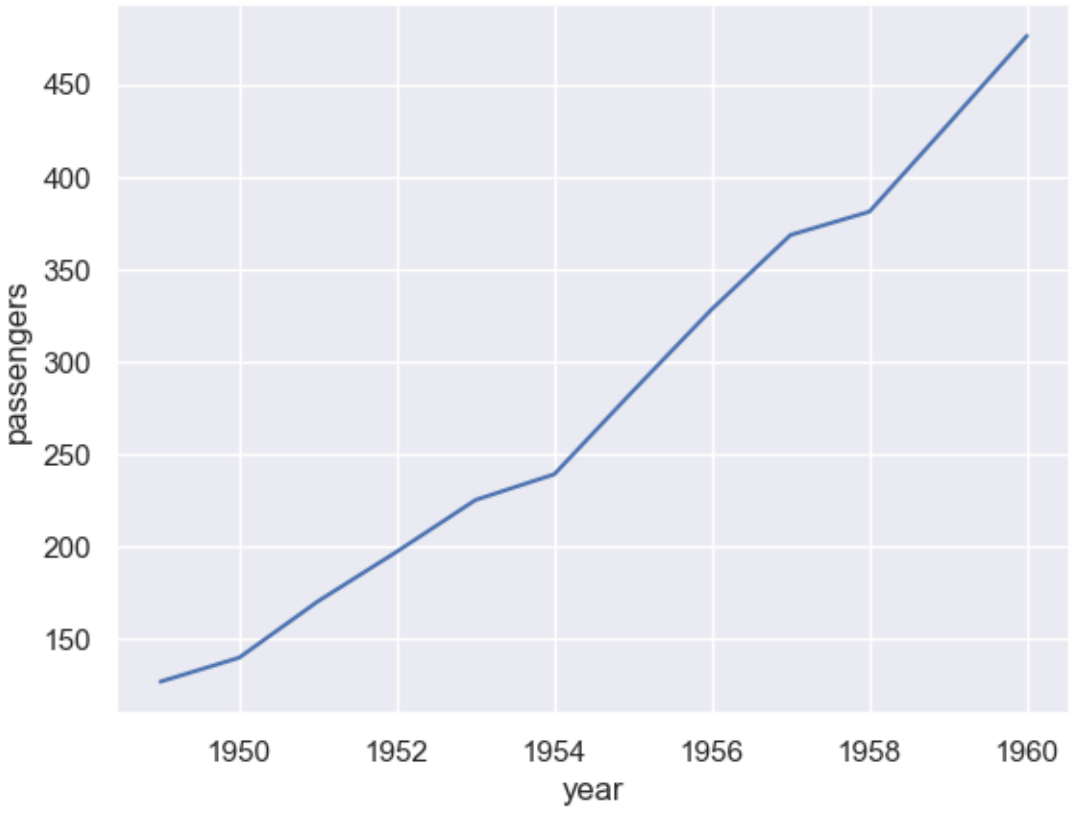
[17]:

	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121
..	…	…	…
139	1960	Aug	606
140	1960	Sep	508
141	1960	Oct	461
142	1960	Nov	390
143	1960	Dec	432

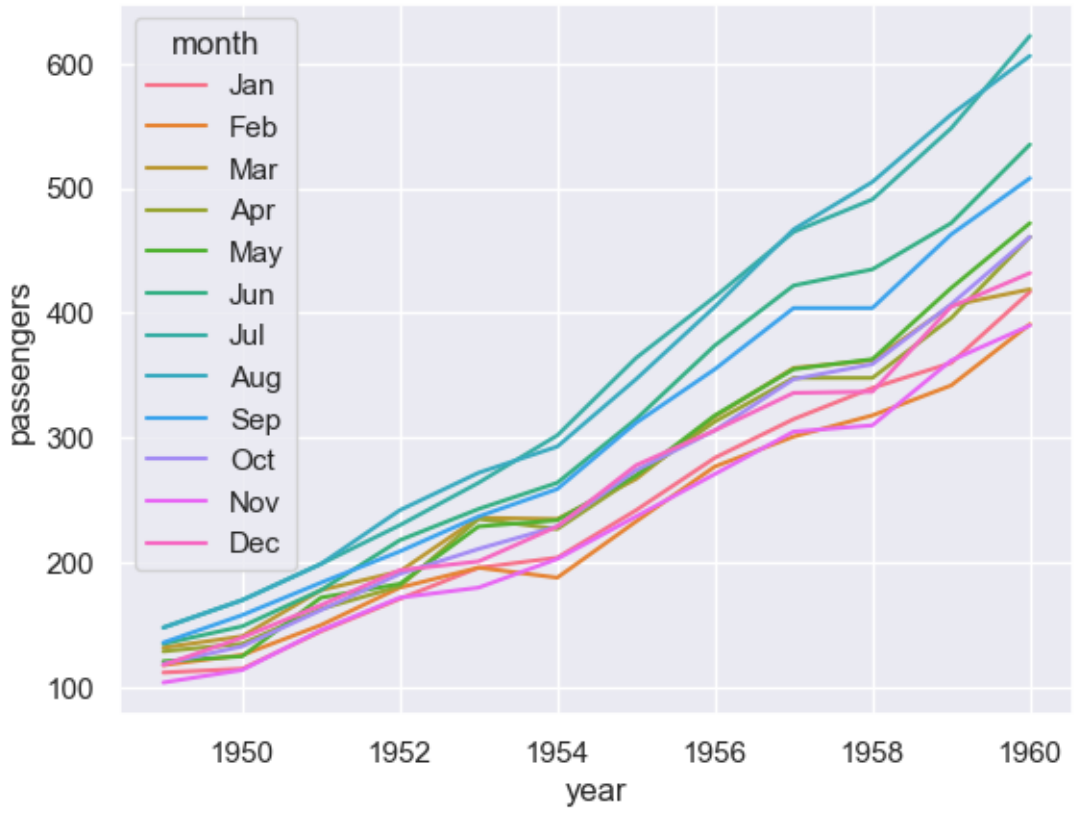
[144 rows x 3 columns]

Passing the entire dataset in long-form mode will aggregate over repeated values (each year) to show the mean and 95% confidence interval:

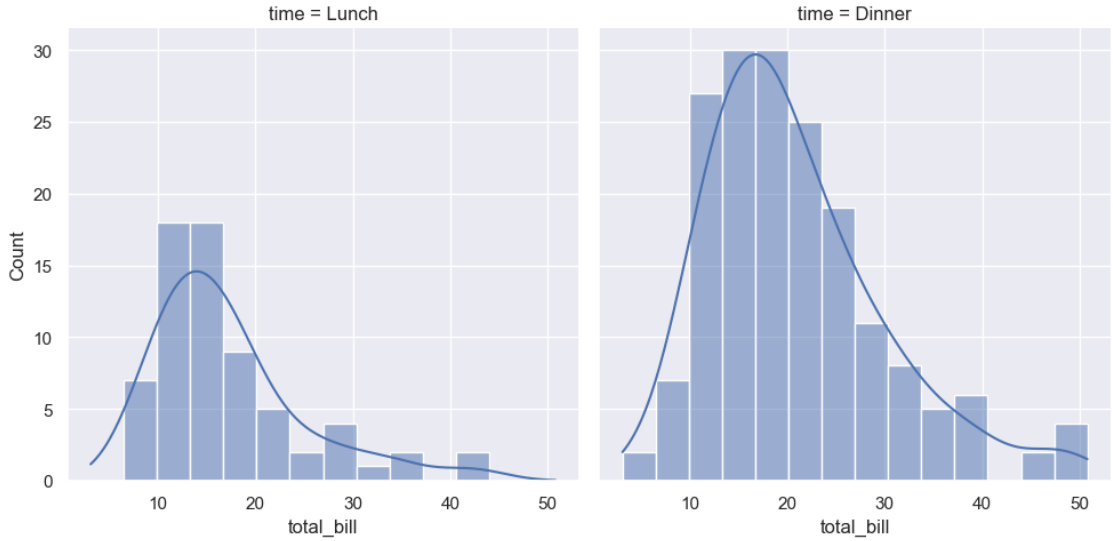
[29]:
```
sns.lineplot(data=df_flights, x="year", y="passengers",errorbar=None)
plt.show()
```



[19]:
```
sns.lineplot(data=df_flights, x="year", y="passengers", hue="month")
plt.show()
```



[20]:
```
sns.displot(data=df_tips, x="total_bill", col="time", kde=True)
plt.show()
```



[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: