20230330_Wirtschaftsinformatik_MPW2

March 30, 2023

[1]:

```
# graphische Darstellung von Daten
```

[2]:

```
!pip install seaborn # %pip install seaborn
```

```
Requirement already satisfied: seaborn in
/Users/h4/anaconda3/lib/python3.9/site-packages (0.11.2)
Requirement already satisfied: numpy>=1.15 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from seaborn) (1.23.2)
Requirement already satisfied: pandas>=0.23 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from seaborn) (1.4.2)
Requirement already satisfied: matplotlib>=2.2 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from seaborn) (3.5.1)
Requirement already satisfied: scipy>=1.0 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from seaborn) (1.9.1)
Requirement already satisfied: pillow>=6.2.0 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from matplotlib>=2.2->seaborn)
(9.0.1)
Requirement already satisfied: python-dateutil>=2.7 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from matplotlib>=2.2->seaborn)
(2.8.2)
Requirement already satisfied: cycler>=0.10 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from matplotlib>=2.2->seaborn)
(0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from matplotlib>=2.2->seaborn)
(1.4.2)
Requirement already satisfied: pyparsing>=2.2.1 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from matplotlib>=2.2->seaborn)
(3.0.4)
Requirement already satisfied: packaging>=20.0 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from matplotlib>=2.2->seaborn)
(21.3)
Requirement already satisfied: fonttools>=4.22.0 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from matplotlib>=2.2->seaborn)
(4.25.0)
Requirement already satisfied: pytz>=2020.1 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from pandas>=0.23->seaborn)
(2021.3)
```

```
Requirement already satisfied: six>=1.5 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from python-
dateutil>=2.7->matplotlib>=2.2->seaborn) (1.16.0)
```

```
[3]: import seaborn as sns # alternative zu matplotlib
```

```
[4]: !pip install pandas #pandas brauchen wir um Daten zu steuern
```

```
Requirement already satisfied: pandas in /Users/h4/anaconda3/lib/python3.9/site-
packages (1.4.2)
Requirement already satisfied: python-dateutil>=2.8.1 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from pandas) (2021.3)
Requirement already satisfied: numpy>=1.20.0 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from pandas) (1.23.2)
Requirement already satisfied: six>=1.5 in
/Users/h4/anaconda3/lib/python3.9/site-packages (from python-
dateutil>=2.8.1->pandas) (1.16.0)
```

```
[5]: import pandas as pd
```

```
[6]: import matplotlib.pyplot as plt
```

```
[8]: # Datenbänke

     print(sns.get_dataset_names())
```

```
['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes',
'diamonds', 'dots', 'dowjones', 'exercise', 'flights', 'fmri', 'geyser', 'glue',
'healthexp', 'iris', 'mpg', 'penguins', 'planets', 'seaice', 'taxis', 'tips',
'titanic']
```

```
[9]: # df (dataframe = dataset) wird geladen
     # wir definieren eine Variable "df" aus dem Datasets von seaborn

     df = sns.load_dataset('car_crashes')
```

```
[10]: df.head()
```

```
[10]:    total  speeding  alcohol  not_distracted  no_previous  ins_premium  \
      0   18.8     7.332    5.640          18.048       15.040       784.55
      1   18.1     7.421    4.525          16.290       17.014      1053.48
      2   18.6     6.510    5.208          15.624       17.856       899.47
      3   22.4     4.032    5.824          21.056       21.280       827.34
      4   12.0     4.200    3.360          10.920       10.680       878.41

         ins_losses abbrev
```

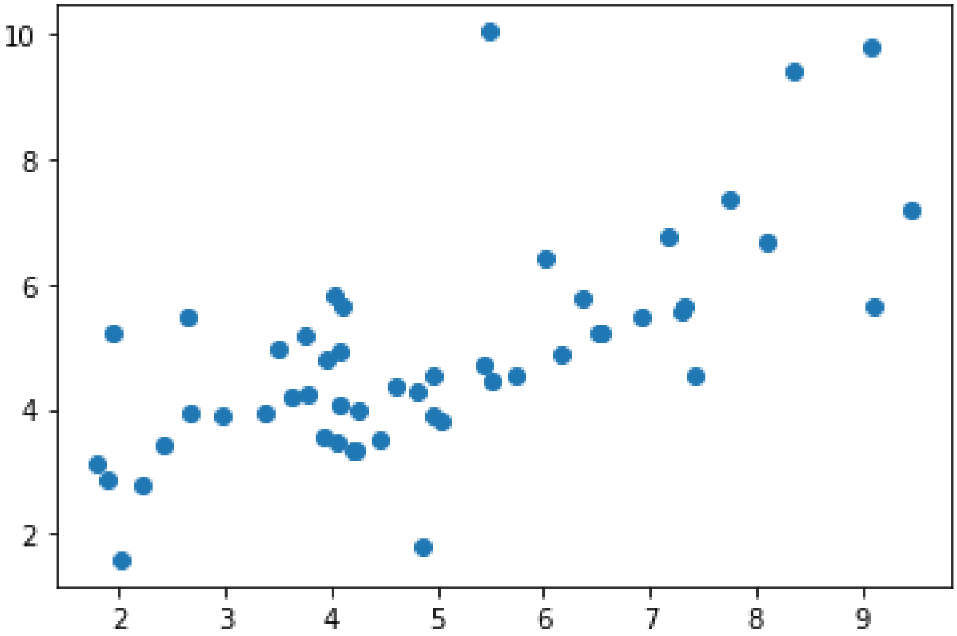
```
0      145.08     AL
1      133.93     AK
2      110.35     AZ
3      142.39     AR
4      165.63     CA
```

[11]:

```
# darstellung im Scatterplot mit Matplotlib
```

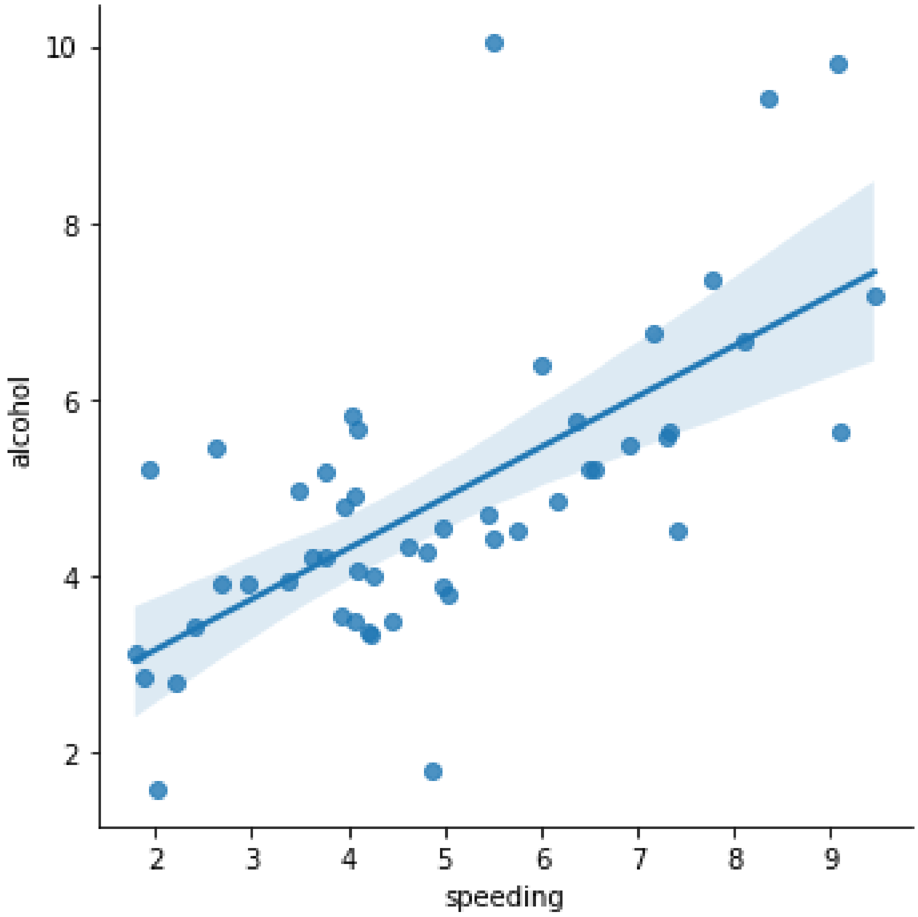
[12]:

```
plt.scatter(df.speeding, df.alcohol)
plt.show()
```

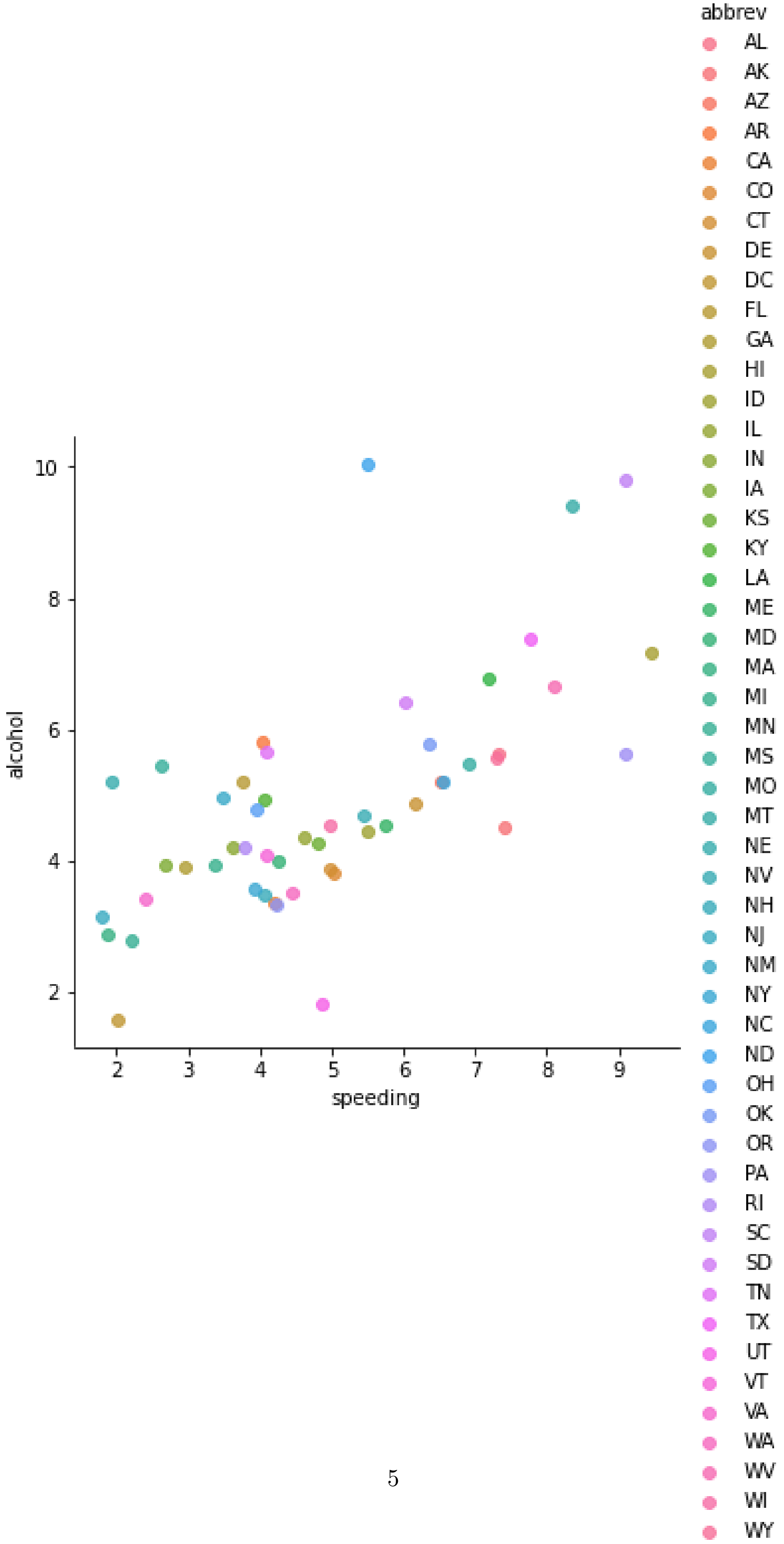


[13]:

```
sns.lmplot(x = 'speeding', y = 'alcohol', data = df)
plt.show()
```



[14]:
```
# mit "hue" erzeugen wir eine Gruppierung

sns.lmplot(x = 'speeding', y = 'alcohol', data = df, hue = "abbrev")
plt.show()
```


abbrev
AL
AK
AZ
AR
CA
CO
CT
DE
DC
FL
GA
HI
ID
IL
IN
IA
KS
KY
LA
ME
MD
MA
MI
MN
MS
MO
MT
NE
NV
NH
NJ
NM
NY
NC
ND
OH
OK
OR
PA
RI
SC
SD
TN
TX
UT
VT
VA
WA
WV
WI
WY
10
8
6
4
2
alcohol
2
3
4
5
6
7
8
9
speeding

[15]:

```
# Übung (ähnlich wie die Prüfung): bitte stellen Sie einen Scatterplot mit␣
 ↪Seaborn
# aus dem car_crashes dataset, welche auf der X-Achse ins_premium und auf der␣
 ↪Y-Achse alcohol hat, mit "hue" Staaten
```

[16]:

```
# beispiel mit einem anderen Dataset "tips" (truinkgeld)
```

[17]:

```
tips = sns.load_dataset('tips')
```

[18]:

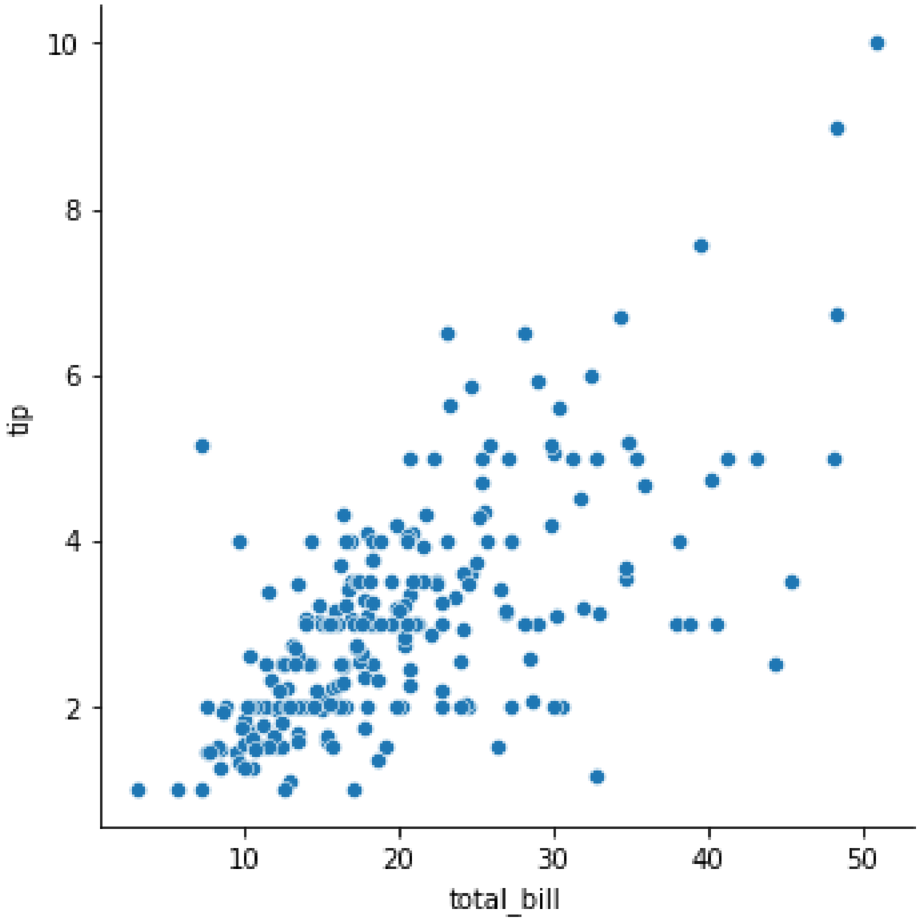
```
tips.head()
```

[18]:

```
   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
```

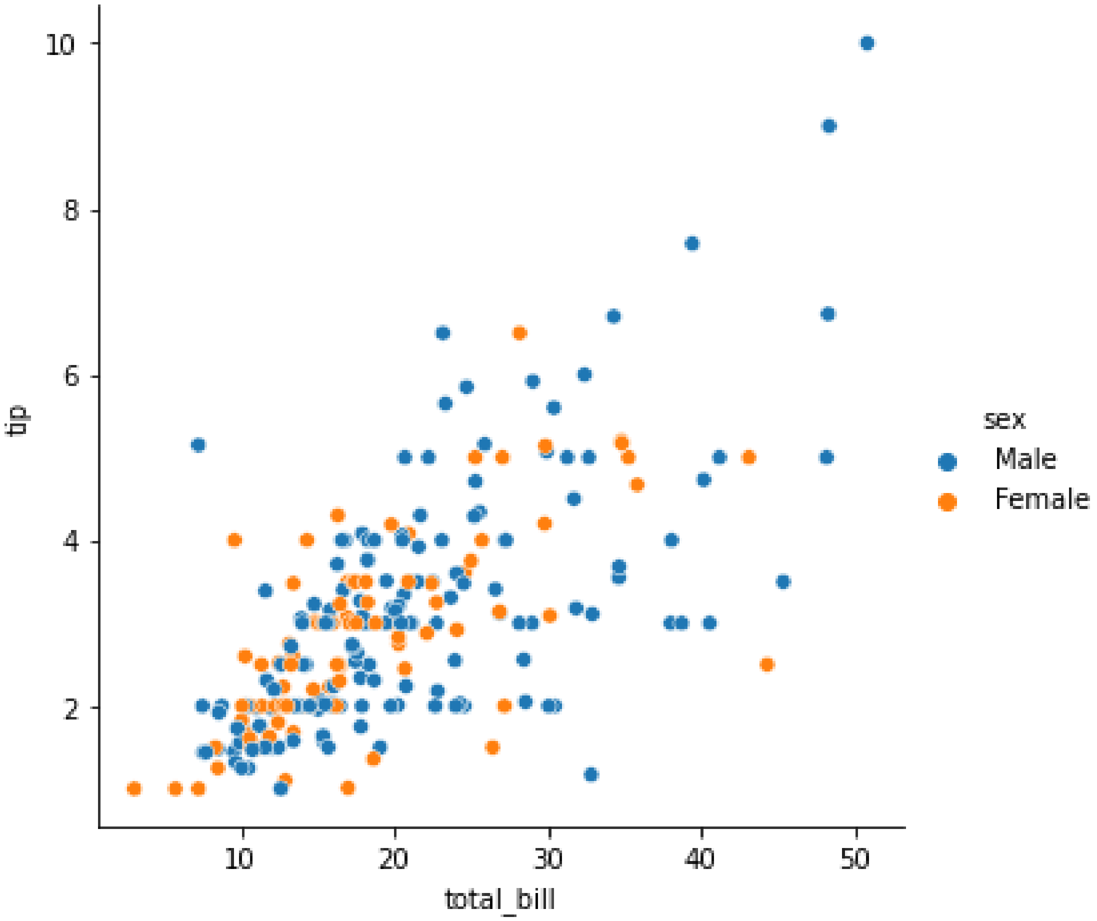
[19]:

```
# relations plot "relplot"

sns.relplot(data=tips, x = 'total_bill', y='tip')
plt.show()
```

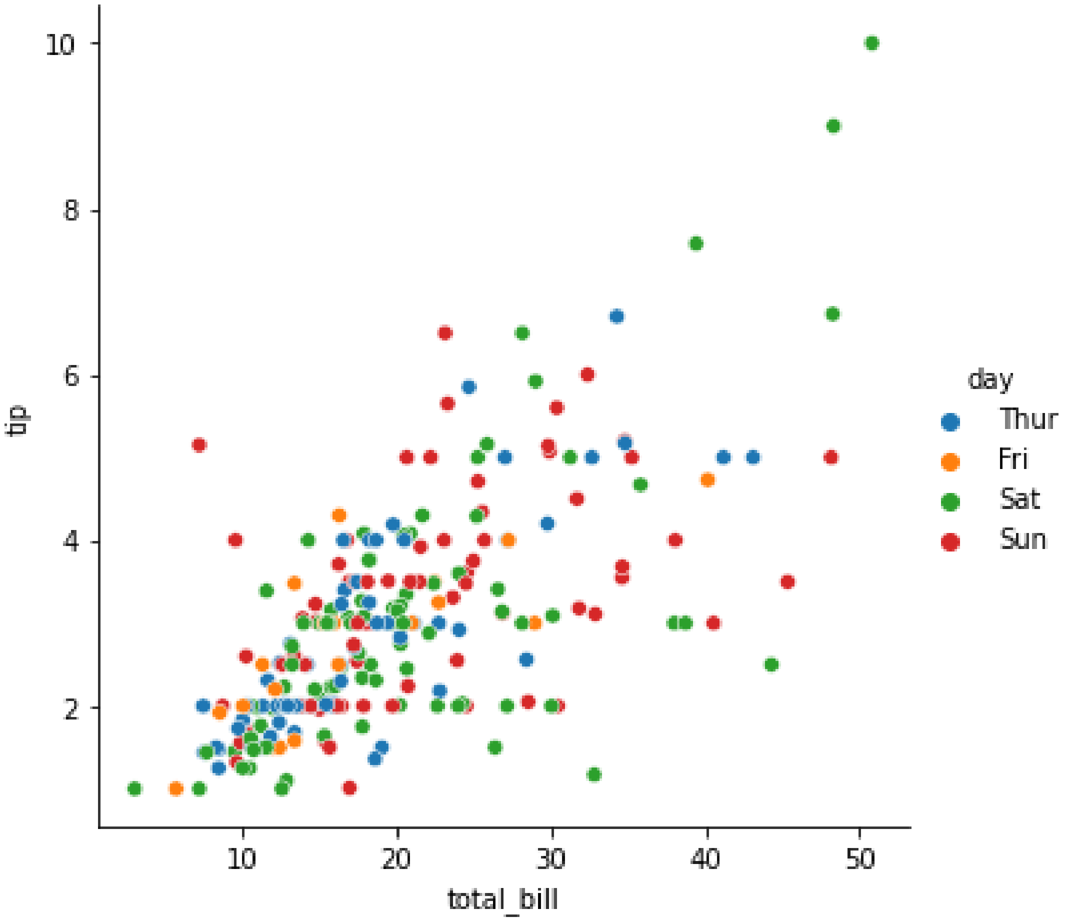


[20]:

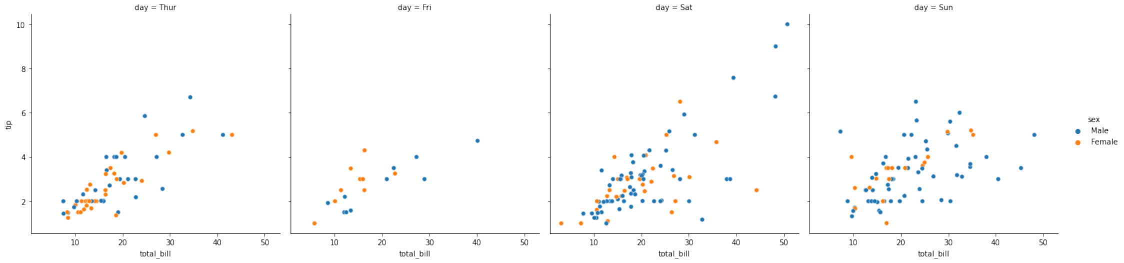
```
# Aufteilung nach Geschlecht

sns.relplot(data=tips, x = 'total_bill', y='tip', hue = "sex")
plt.show()
```



[21]:
```
# Aufteilung nach Tagen

sns.relplot(data=tips, x = 'total_bill', y='tip', hue = "day")
plt.show()
```

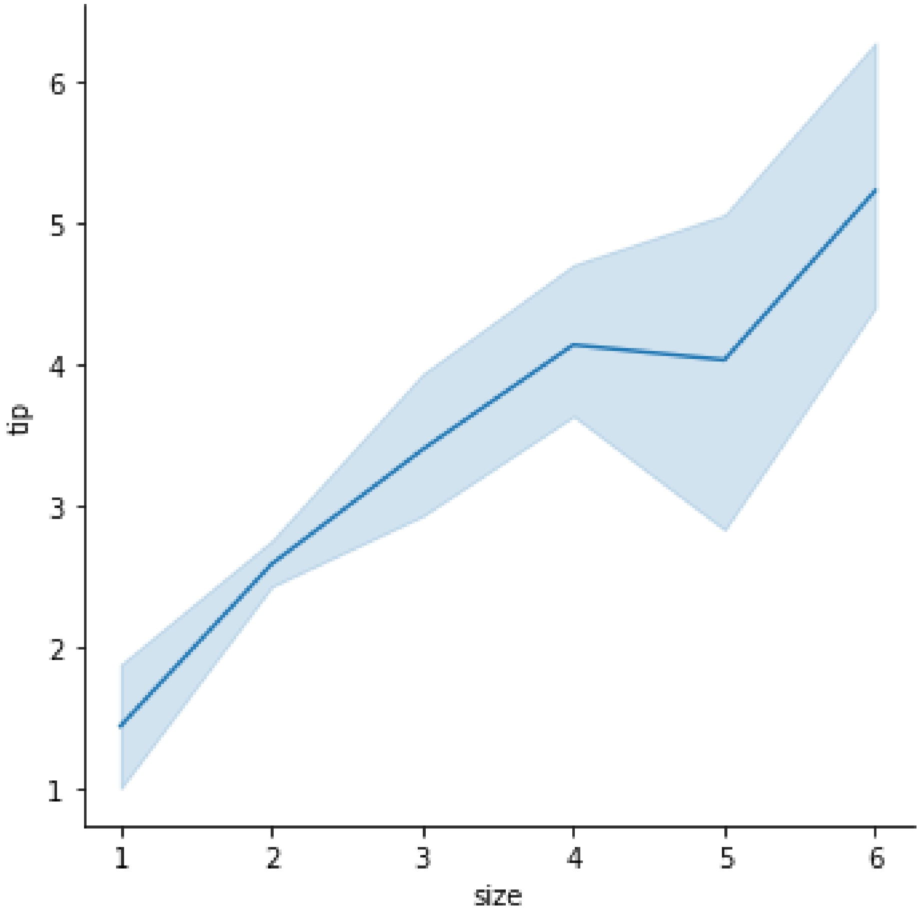


[22]:
```
# verschiedne Tage an verschiedenen Graphen darstellen

sns.relplot(data=tips, x='total_bill', y='tip', hue='sex', col='day')
plt.show()
```



[23]:
```
# daten in form von einer Linie darstellen mit Konfidenzinterval
```

```
sns.relplot(data=tips, x='size', y='tip', kind='line')
plt.show()
```

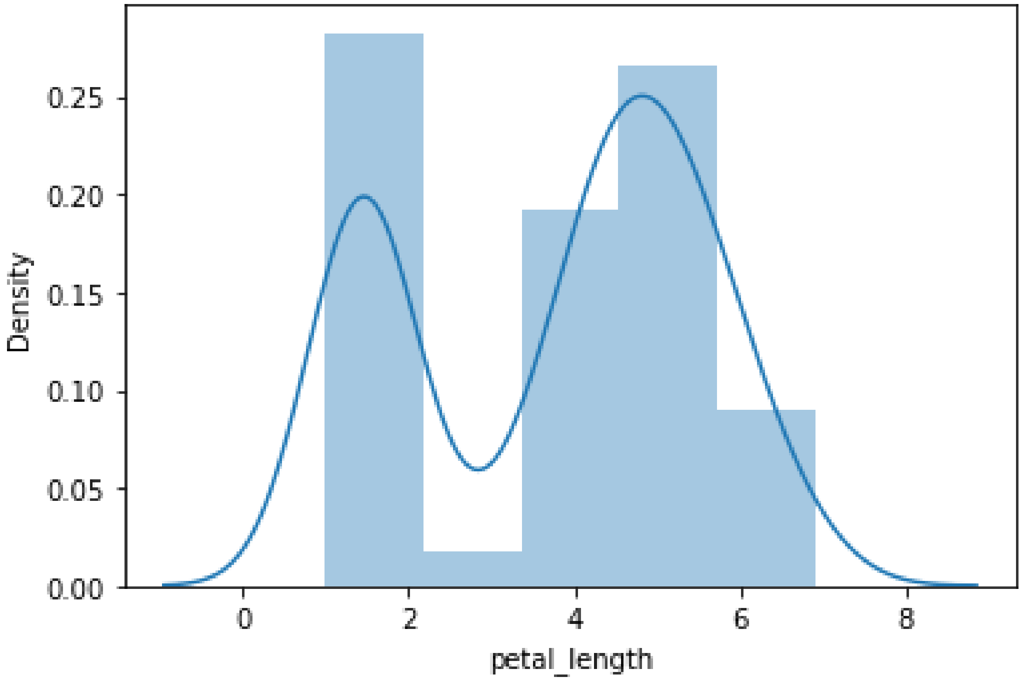


[24]:
```
# histogram

df = sns.load_dataset('iris')
df.head()
```

[24]:
```
   sepal_length  sepal_width  petal_length  petal_width species
0           5.1          3.5           1.4          0.2  setosa
1           4.9          3.0           1.4          0.2  setosa
2           4.7          3.2           1.3          0.2  setosa
3           4.6          3.1           1.5          0.2  setosa
4           5.0          3.6           1.4          0.2  setosa
```

[26]:
```
sns.distplot(df['petal_length'])
plt.show()
```

```
/Users/h4/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

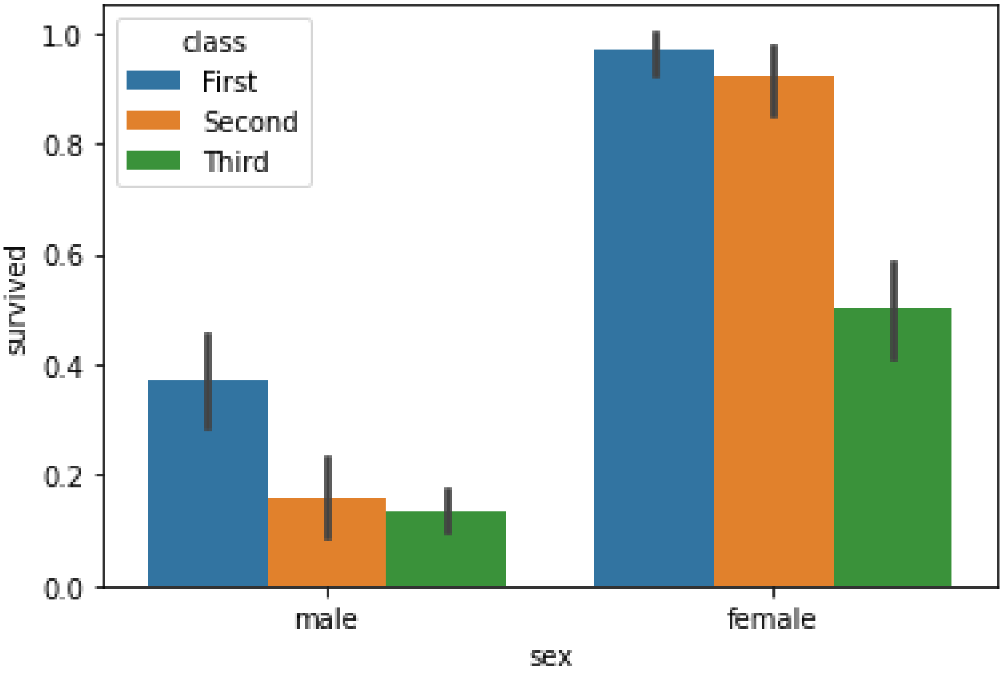


[27]:
```
# vertikaler barplot

titanic = sns.load_dataset('titanic')
titanic.head()
```

[27]:
```
   survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
0         0       3    male  22.0      1      0   7.2500        S  Third
1         1       1  female  38.0      1      0  71.2833        C  First
2         1       3  female  26.0      0      0   7.9250        S  Third
3         1       1  female  35.0      1      0  53.1000        S  First
4         0       3    male  35.0      0      0   8.0500        S  Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True
```

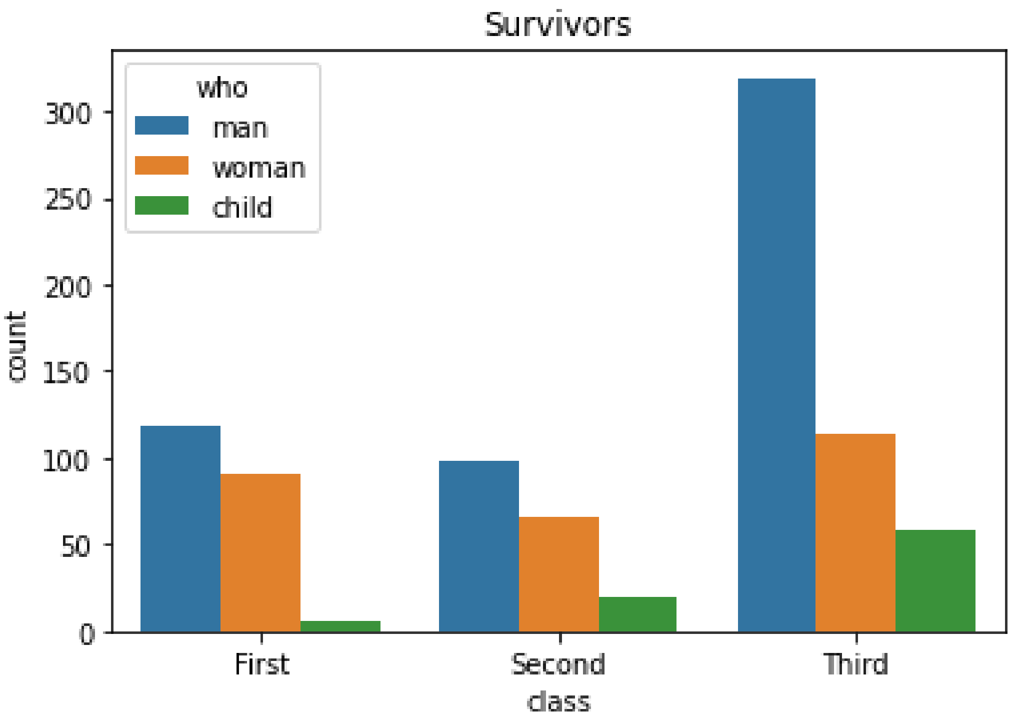
[28]:

```
sns.barplot(x = 'sex', y= 'survived', hue = 'class', data = titanic)
plt.show()
```



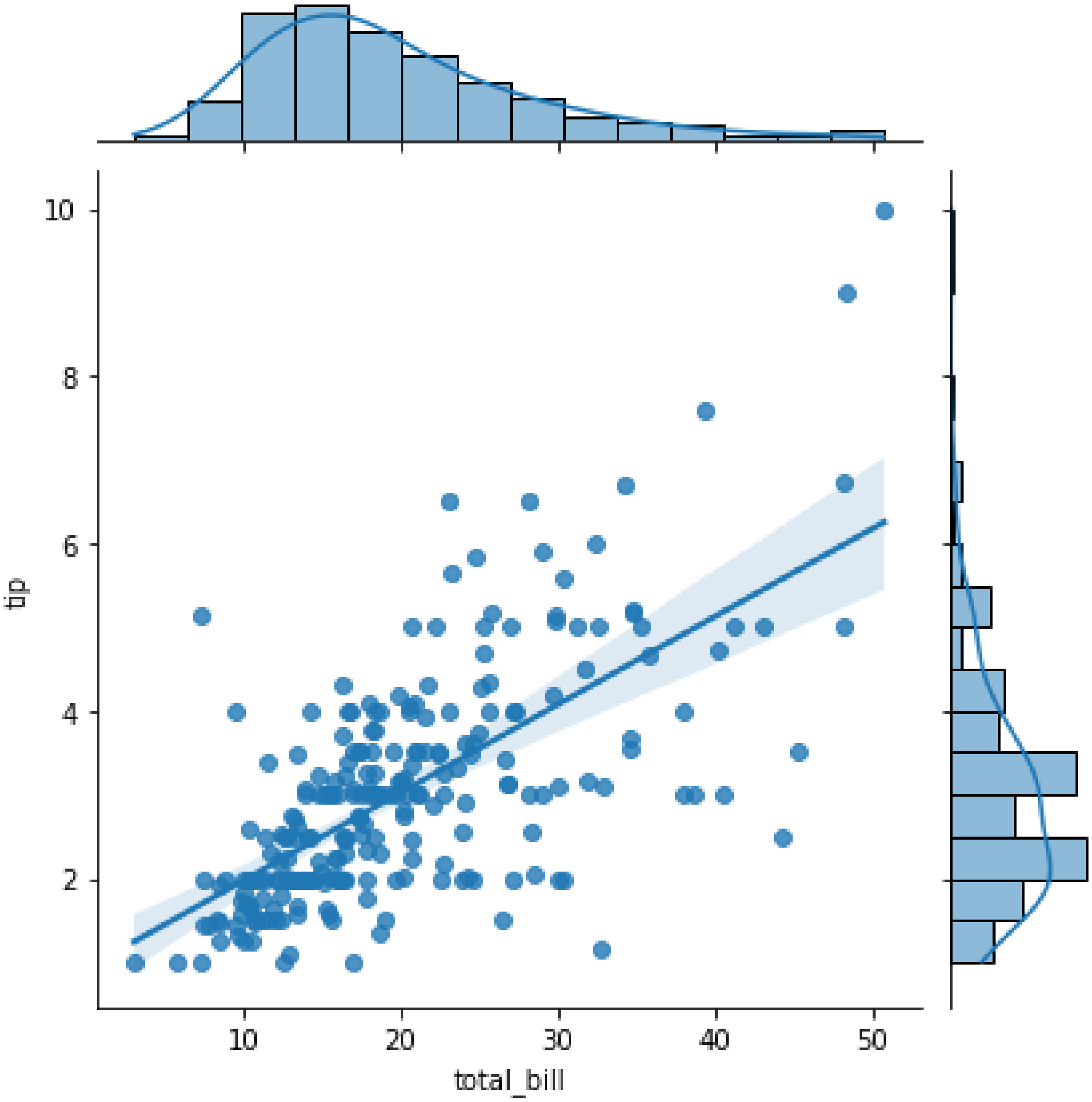
[29]:

```
# countplot

sns.countplot(x = 'class', hue = 'who', data = titanic)
plt.title('Survivors')
plt.show()
```

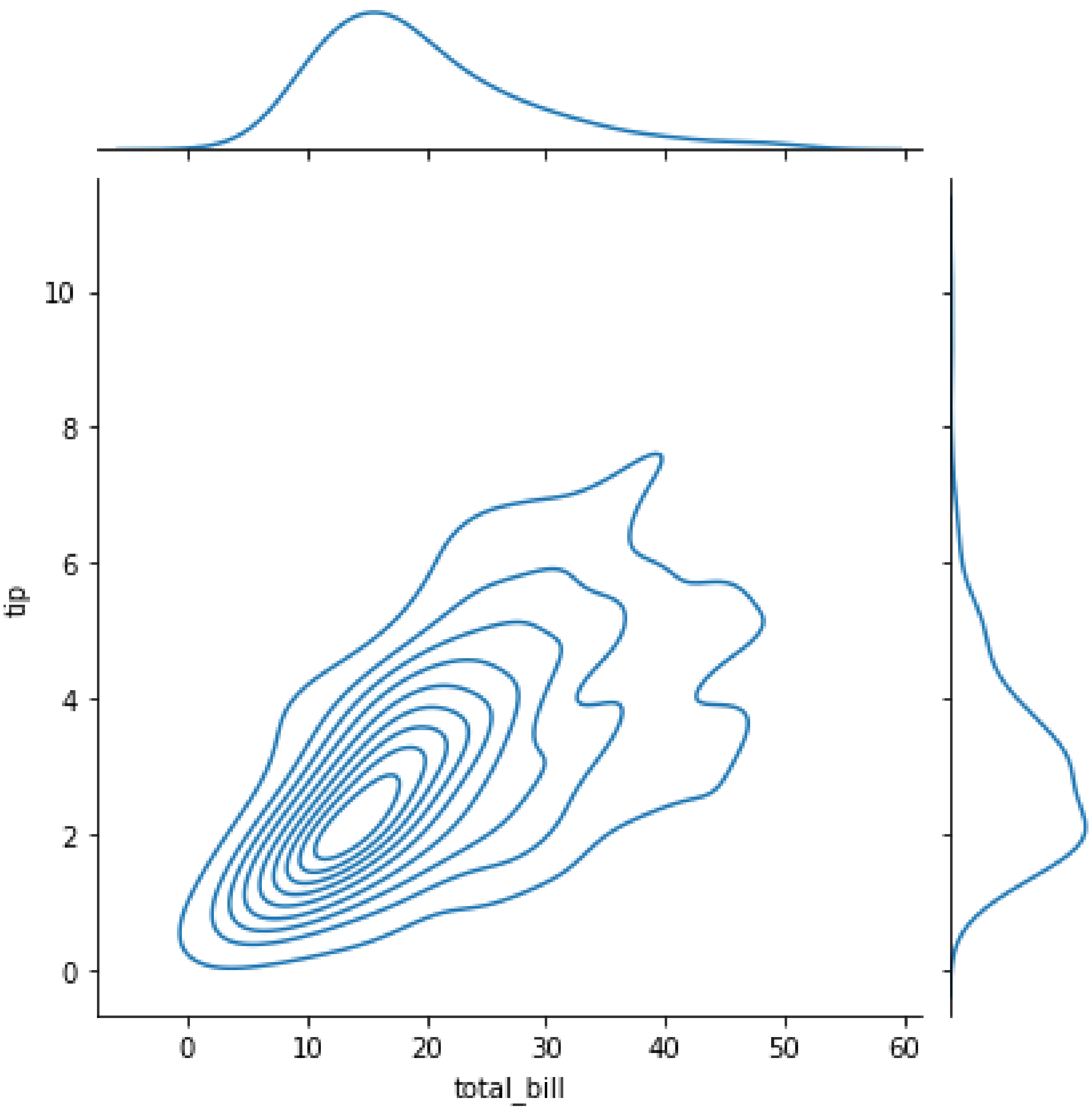


[30]:

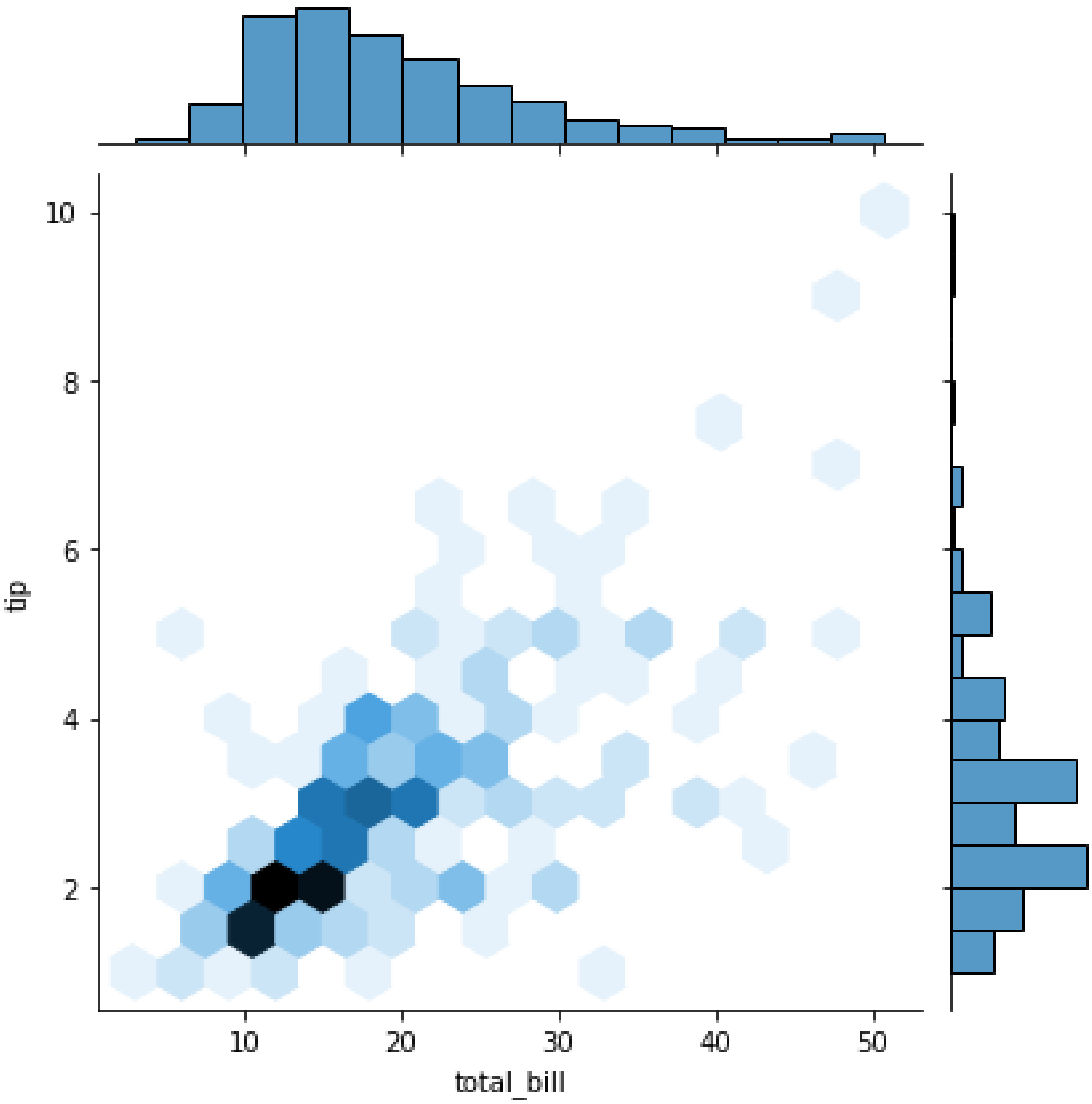
```
# jointplot

tips = sns.load_dataset('tips')
tips.head()

sns.jointplot(x='total_bill', y='tip', data=tips, kind='reg')
#regression
plt.show()
```



[32]:
```
# kde distribution

sns.jointplot(x='total_bill', y='tip', data=tips, kind='kde')
#distribution
plt.show()
```



[33]:
```
sns.jointplot(x='total_bill', y='tip', data=tips, kind='hex')
#hexagon
plt.show()
```


10
8
6
4
2
tip
10
20
30
40
50
total_bill

```
[ ]:
```