**Seaborn**

* Seaborn yra duomenų vizualizacijos biblioteka.
* Ji sukurta Matplotlib pagrindu.
* Stilistiškai kiek išbaigtesnė už Matplotlib.
* Labai gerai veikia su pandas DataFrame'ais.
* Labai plačiai naudojama duomenų analizėje.
* Dokumentaciją rasite [čia](https://seaborn.pydata.org/).

diegiasi *conda install seaborn*, arba *pip install seaborn*

In [2]:

**import** pandas **as** pd

**import** matplotlib **as** plt

**import** seaborn **as** sns

In [3]:

**%matplotlib** inline

Dažnai duomenų analizės bibliotekos turi integruotus dataset'us, demonstraciniams tikslams. Pasinaudokime vienu iš jų:

In [4]:

tips **=** sns**.**load\_dataset('tips')

In [5]:

tips**.**head()

Out[5]:

|  | **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 |

In [6]:

tips**.**shape

Out[6]:

(244, 7)

**Pasiskirstymo histogramos**

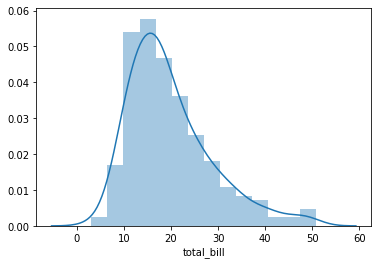
Jų pagalba matysime, koks yra kokio nors vieno rodiklio pasiskirstymas.

In [7]:

sns**.**distplot(tips['total\_bill'])

Out[7]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4e1f0f3898>



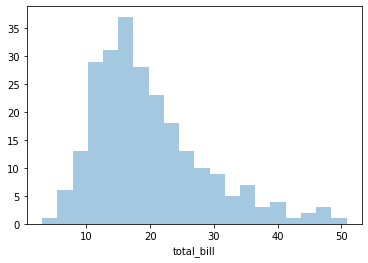
parametras **kde** gali nuimti lijiną, o parametras **bins** padidinti stulpelių skaičių.

In [8]:

sns**.**distplot(tips['total\_bill'], kde**=False**, bins**=**20)

Out[8]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4e1ed27518>



Matome kad dažniausiai sąskaitos išrašomos tarp 10 ir 20$.

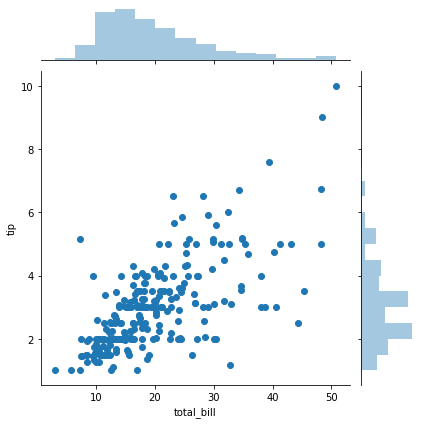
**.jointplot()**

In [9]:

sns**.**jointplot(x**=**'total\_bill', y**=**'tip', data**=**tips)

Out[9]:

<seaborn.axisgrid.JointGrid at 0x7f4e1ed60278>



Matome, koks yra sąntykio tarp 'total\_bill' ir 'tip' pasiskirstymas. Pagrindiniame lange yra sklaidos diagrama, o iš dešinės ir viršuje - tos pačios histogramos, iš kurių sukombinavome rezultatą.

Rezultatą galime koreguoti su parametru **kind**:

In [10]:

sns**.**jointplot(x**=**'total\_bill', y**=**'tip', data**=**tips, kind**=**'hex')

Out[10]:

<seaborn.axisgrid.JointGrid at 0x7f4e1eb9af28>

Paveikslėlis, kuriame yra žinutė

Automatiškai sugeneruotas aprašymas

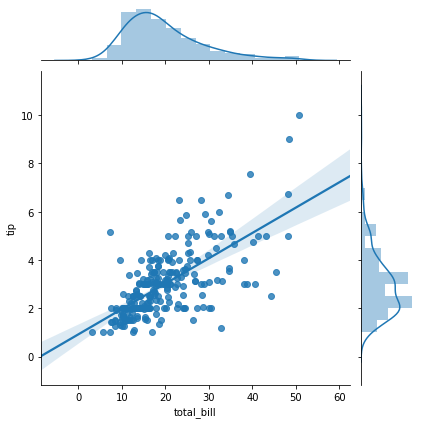
reikšmė 'reg' stengiasi nubrėžti mums liniją, kuri atspindėtų tendenciją.

In [11]:

sns**.**jointplot(x**=**'total\_bill', y**=**'tip', data**=**tips, kind**=**'reg')

Out[11]:

<seaborn.axisgrid.JointGrid at 0x7f4e1ea19828>



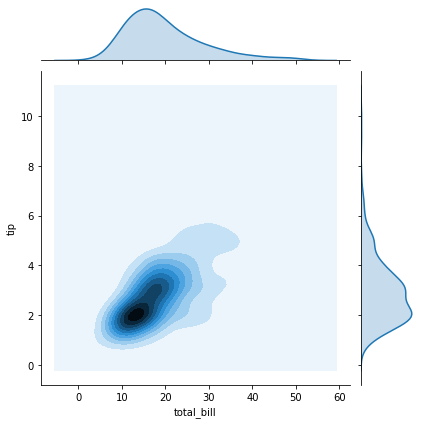
taip pat galime naudoti **kind** su 'kde' reikšme. Tokiu atveju matysime didžiausio tankio zonas:

In [12]:

sns**.**jointplot(x**=**'total\_bill', y**=**'tip', data**=**tips, kind**=**'kde')

Out[12]:

<seaborn.axisgrid.JointGrid at 0x7f4e1e02a160>



**.scatterplot()**

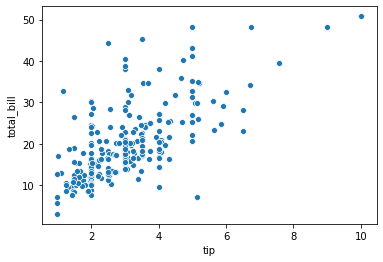
parametruose nurodžius ašis ir šaltinį, nubrėžia mums sklaidos diagramą.

In [13]:

sns**.**scatterplot('tip', 'total\_bill', data**=**tips)

Out[13]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4e1d1c39e8>



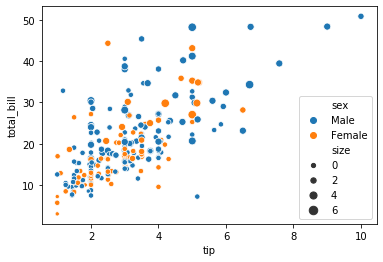
Parametruose taip pat galima nurodyti **hue** ir **size** reikšmes:

In [16]:

sns**.**scatterplot('tip', 'total\_bill', data**=**tips, hue**=**'sex', size**=**'size')

Out[16]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4e1cfe1898>



**.pairplot()**

duoda mums visas įmanomas pasiskirstymo kombinacijas lentelėje:

In [20]:

sns**.**pairplot(tips)

Out[20]:

<seaborn.axisgrid.PairGrid at 0x7f2c0808f470>

Paveikslėlis, kuriame yra žinutė, elektroniniai prietaisai

Automatiškai sugeneruotas aprašymas

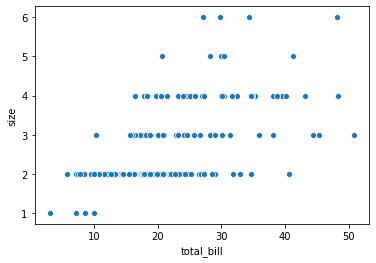
pvz, dešinys-apatinis grafikas yra ne kas kita, kaip:

In [17]:

sns**.**scatterplot(x**=**'total\_bill', y**=**'size', data**=**tips)

Out[17]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f4e1d0067f0>



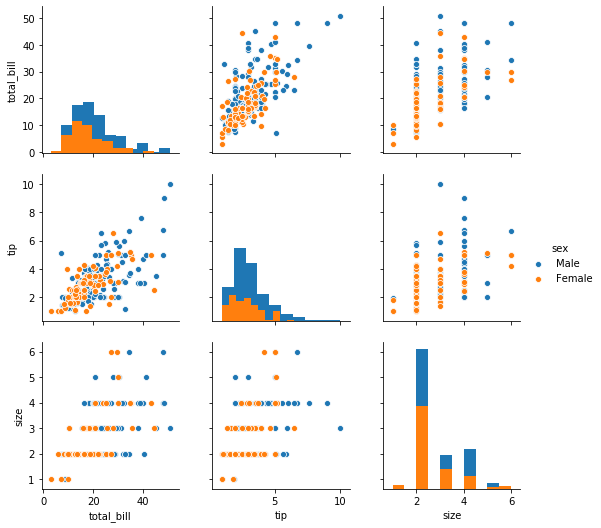
parametras **hue** išskiria kategorines reikšmes, o **diag\_kind** šiuo atveju nurodo, kad įstrižainėje norime matyti ne linijas, o histogramas pvz.:

In [21]:

sns**.**pairplot(tips, hue**=**'sex', diag\_kind**=**'hist')

Out[21]:

<seaborn.axisgrid.PairGrid at 0x7f4e1c238b70>



Turime skirtingų spalvų taškus, kur oranžinė spalva reprezentuoja moteris.

**Kategorizavimo histogramos**

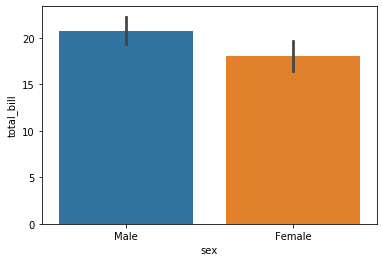
**.barplot()** išskirsto kategorijas pagal kurį nors rodiklį ir leidžia tam rodikliui taikyi kokią nors funkciją:

In [36]:

sns**.**barplot(x**=**'sex', y**=**'total\_bill', data**=**tips)

Out[36]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f2c0106e630>



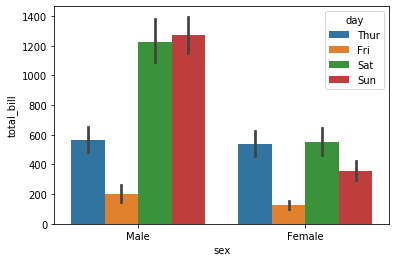
Jeigu mes nenurodome, kokia ta funkcija, numatyta reikšmė yra vidurkis. Taigi šiame pavyzdyje matome sąskaitos vidurkių pasiskirstymą tarp lyčių. Jeigu norime nurodyti vidurkį, turime naudoti **estimator** parametrą, pvz.:

In [43]:

sns**.**barplot(x**=**'sex', y**=**'total\_bill', data**=**tips, hue**=**'day', estimator**=**sum)

Out[43]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f2c00e42198>



Šiuo atveju matome bendras sumas. Taip pat panaudojome *hue*, tokiu būdu rezultatą išskirstydami savaitės dienomis. *Nekreipkite dėmesio į juodas linijas, jos yra error bars, rodo statistinį skaičiavimo patikimumą, ir nėra mums aktualios. Norint jas išjungti, parametruose nurodykite ci=False*

**.countplot() tiesiog suskaičiuoja kategorijas:**

In [45]:

sns**.**countplot(x**=**'smoker', data**=**tips)

Out[45]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f2c00ddcf98>



In [49]:

tips**.**head()

Out[49]:

|  | **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 |

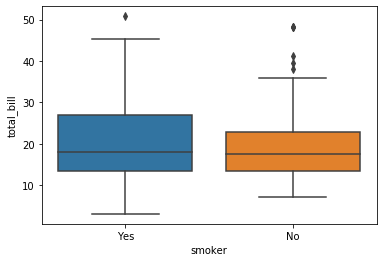
**.boxplot() - stačiakampės diagramos**

In [53]:

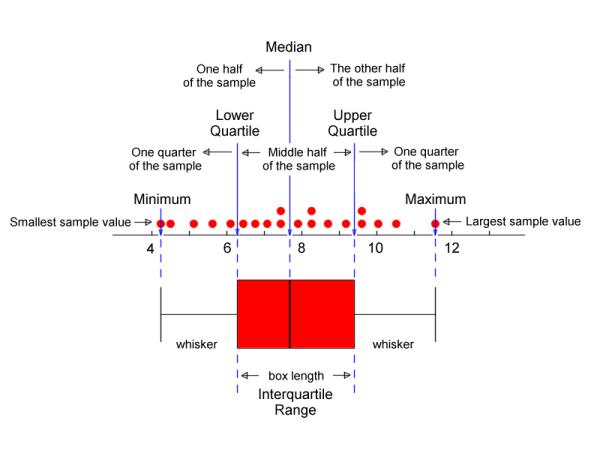
sns**.**boxplot(x**=**'smoker', y**=**'total\_bill', data**=**tips)

Out[53]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f2c00a17860>



Kas yra *boxplot* (stačiakampė diagrama)? Iš esmės tai į ketvirtadalius sugrupuota informacija. Mūsų pavyzdyje matosi, kad rūkalių viršutinis ketvirtadalis maistui išleido nuo ~28 iki ~46, kai tuo tarpu nerūkančiųjų viršutinis ketvirtadalis išleido nuo ~24 iki ~37. Taškai viršuje yra išimtys. Jie pasitaiko per retai, kad būtų įtraukti į ketvirtadalius.



Šis paveikslėlis paaiškina boxplot atvaizdavimą.

**.heatmap()**

pandas turi metodą, kuris parodo skaičių koreliacijas lentelėje:

In [56]:

koreliacijos **=** tips**.**corr()

In [57]:

koreliacijos

Out[57]:

|  | **total\_bill** | **tip** | **size** |
| --- | --- | --- | --- |
| **total\_bill** | 1.000000 | 0.675734 | 0.598315 |
| **tip** | 0.675734 | 1.000000 | 0.489299 |
| **size** | 0.598315 | 0.489299 | 1.000000 |

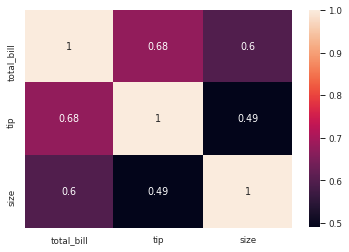
čia turime koreliacijų matricą. Ji rodo, kaip skirtingi kintamieji tarpusavyje koreliuoja. Kad būtų lengviau skaityti, atsispausdinkime vizualizaciją (mozaikinę diagramą):

In [81]:

sns**.**heatmap(koreliacijos, annot**=True**)

Out[81]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f2bf027e278>



**.FacetGrid()**

**.FacetGrid()** leidžia susikurti tinklelį, kurį vėliau reikės užpildytyi diagramomis. Pvz.:

In [22]:

sns**.**FacetGrid(data**=**tips, col**=**'time', row **=**'smoker')

Out[22]:

<seaborn.axisgrid.FacetGrid at 0x7f4e1657e668>

Paveikslėlis, kuriame yra pastatas, siluetas

Automatiškai sugeneruotas aprašymas

O pats užpildymas atrodo taip:

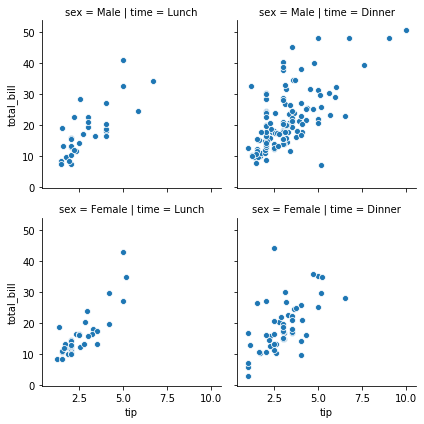
In [23]:

g **=** sns**.**FacetGrid(data **=** tips, col**=**'time', row**=**'sex')

g**.**map(sns**.**scatterplot, 'tip', 'total\_bill')

Out[23]:

<seaborn.axisgrid.FacetGrid at 0x7f4e1657e400>



Taip išskirstėme atvaizdavimą į keturias diagramas. Vienoje pavaizduotos moterys per pietus, kitoje moterys per vakarienę ir t.t.

**Stilius ir spalvos**

**.set\_style()**

Seaborn leidžia nustatyti stilių su **.set\_style()** metodu. Į parametrus reikia įkelti vieną iš šių reikšmių - darkgrid, whitegrid, dark, white, ticks.

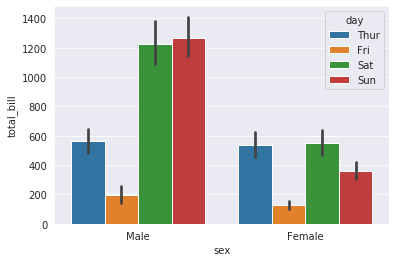
In [65]:

sns**.**set\_style('darkgrid')

sns**.**barplot(x**=**'sex', y**=**'total\_bill', data**=**tips, hue**=**'day', estimator**=**sum)

Out[65]:

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f2c001622e8>



**.despine()**

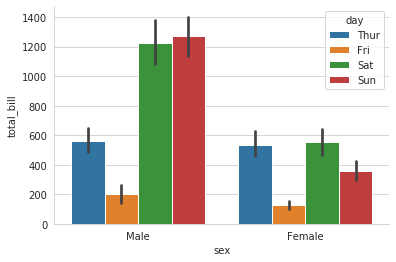
Nuima viršutinį ir dešinį rėmą.

In [24]:

sns**.**set\_style('whitegrid')

sns**.**barplot(x**=**'sex', y**=**'total\_bill', data**=**tips, hue**=**'day', estimator**=**sum)

sns**.**despine()



**palette parametras**

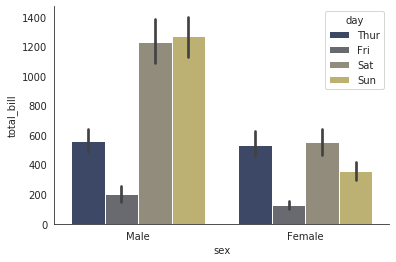
leidžia pasirinkti vieną iš paruoštų naudoti spalvų palečių. Jas rasite [čia](https://matplotlib.org/3.1.0/tutorials/colors/colormaps.html#classes-of-colormaps). Kai kurie metodai nepriima parametro palette, tuomet reikia bandyti tas pačias reikšmes nurodyti į **cmap** parametrą

In [25]:

sns**.**set\_style('white')

sns**.**barplot(x**=**'sex', y**=**'total\_bill', data**=**tips, hue**=**'day', estimator**=**sum, palette**=**'cividis')

sns**.**despine()



Rekomendacija: pasitikrinti galimus įvairių metodų parametrus galite kursoriui mirksint skliaustelių viduje, spaudžiant *Shift+Tab*. Nevenkite eksperimentuoti :)