In [3]:
```
import numpy as np
import pandas as pd
from numpy.random import randn
```

In [4]:
```
# Stats
from scipy import stats
```

In [5]:
```
# Plotting
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
```

In [6]:
```
%matplotlib inline
```

In [7]:
```
# numpy.random.randn(d0, d1, ..., dn)¶
# Return a sample (or samples) from the "standard normal" distribution.
url = 'http://en.wikipedia.org/wiki/Histogram'
ds1 = randn(100)
```

In [8]:
```
ds1.ndim
```

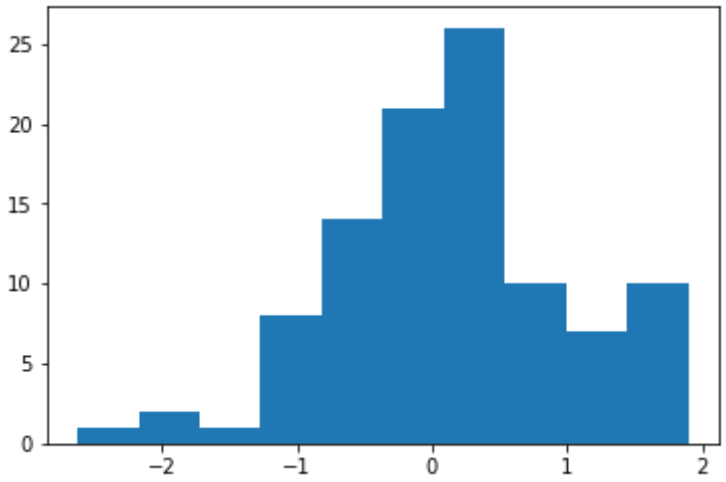
Out[8]:
```
1
```

In [9]:
```
ds1
```

Out[9]:
```
array([-5.26117773e-02,  2.17290036e-01,  6.86234520e-01, -2.62651373e+00,
        1.74321232e+00,  1.33464622e-01,  6.64370596e-01,  6.42556965e-01,
       -6.09041783e-01, -1.01021622e+00, -5.27721365e-01, -2.00630633e+00,
        1.13999473e+00, -7.30515213e-01,  9.90144334e-02, -1.13762101e+00,
       -1.41537200e-01,  3.25992644e-01, -7.95636916e-01,  7.78855589e-01,
       -3.60288563e-01, -7.95214122e-01,  4.73039446e-01, -2.36361239e-01,
       -6.90130518e-01, -2.37709749e-01, -6.20257307e-01,  1.56258589e+00,
       -1.15506197e+00, -3.74413625e-01,  8.69857520e-01,  4.54706543e-01,
        1.64853855e+00,  1.89525947e+00,  3.77660299e-01,  2.94155763e-01,
        7.92556196e-02,  1.08595181e-01, -2.45707244e-01,  1.68267524e-01,
        1.07004246e-01, -9.18220971e-01,  3.73845717e-01, -7.22022231e-01,
        1.34819700e+00,  1.65926838e+00,  4.03341541e-01,  8.33708890e-03,
       -1.33863337e-01, -5.37692261e-01, -5.09635931e-01,  1.47974730e+00,
       -1.33501231e+00,  8.08743834e-02,  1.53286730e+00, -1.00939995e+00,
        9.31174174e-01,  2.42484469e-03,  3.83342650e-01,  9.94009179e-01,
       -2.24559169e-02,  3.54894126e-01,  1.25671026e+00,  4.61263064e-01,
        5.20406045e-01, -1.13488546e+00,  1.38275081e-01,  1.36956701e+00,
        2.08454485e-01,  8.96267218e-01,  1.23539054e-01, -1.11987019e+00,
        7.84298161e-01, -3.02734990e-01, -1.79438019e+00,  6.54641391e-02,
        7.99954737e-01,  2.76702566e-02, -3.20303278e-01,  5.26088964e-01,
        1.01208331e+00,  1.45897480e+00,  1.12240139e+00, -1.64234628e-01,
        1.79289706e+00, -2.29063394e-01,  3.40994365e-01,  4.16632809e-01,
        2.03676307e-01, -5.71112942e-01,  1.44872649e+00,  4.40145240e-01,
       -4.65548445e-03, -9.88062874e-01,  1.68790366e-02,  4.78988393e-01,
       -7.89960783e-01, -2.70921681e-01, -6.19538509e-01,  9.85711097e-01])
```

In [10]:
```
plt.hist(ds1)
```

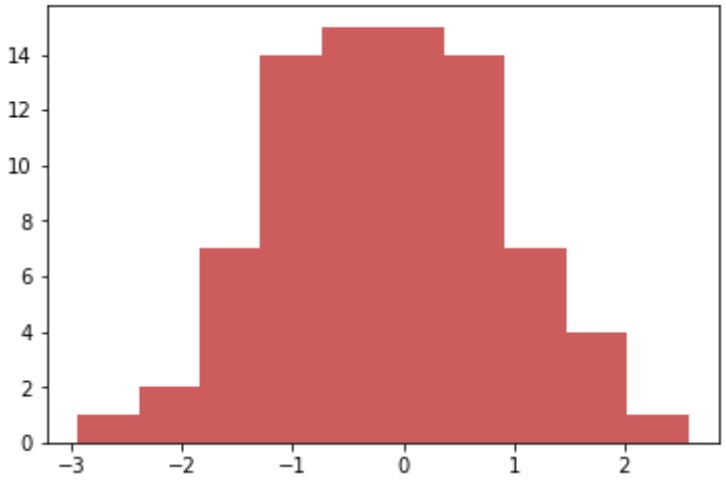
Out[10]:
```
(array([ 1.,  2.,  1.,  8., 14., 21., 26., 10.,  7., 10.]),
 array([-2.62651373, -2.17433641, -1.72215909, -1.26998177, -0.81780445,
        -0.36562713,  0.08655019,  0.53872751,  0.99090483,  1.44308215,
         1.89525947]),
 <a list of 10 Patch objects>)
```



In [11]:
```
ds2 = randn(80)
```

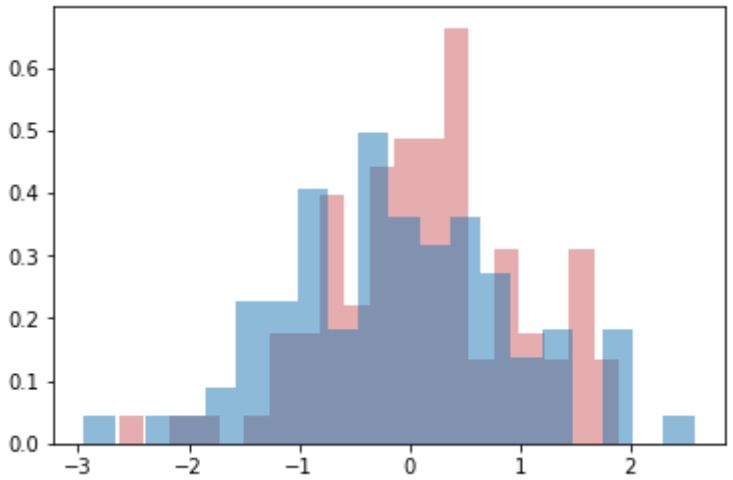
In [12]:
```
ds2
plt.hist(ds2, color="indianred")
```

Out[12]:
```
(array([ 1.,  2.,  7., 14., 15., 15., 14.,  7.,  4.,  1.]),
 array([-2.94884687, -2.39611276, -1.84337865, -1.29064454, -0.73791043,
        -0.18517632,  0.36755779,  0.9202919 ,  1.47302601,  2.02576012,
         2.57849423]),
 <a list of 10 Patch objects>)
```



In [16]:
```
plt.hist(ds1, normed=True, color="indianred", alpha=0.5, bins=20)
plt.hist(ds2, normed=True, alpha=0.5, bins=20)
```

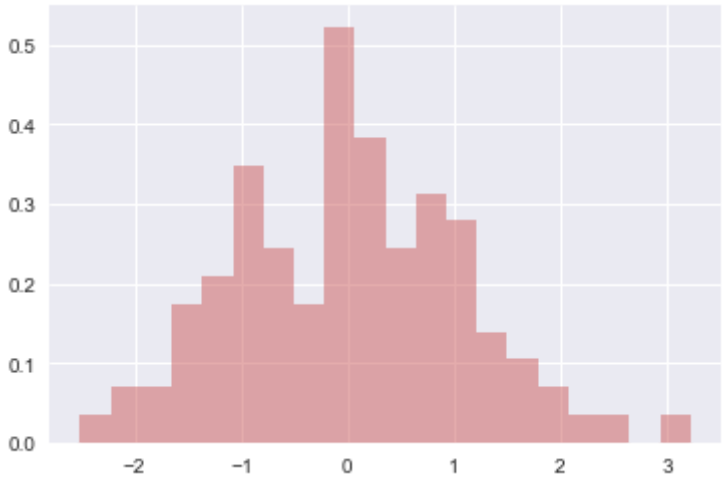
Out[16]:
```
(array([0.0452297 , 0.        , 0.0452297 , 0.0452297 , 0.09045941,
        0.22614852, 0.22614852, 0.40706733, 0.18091881, 0.49752674,
        0.36183763, 0.31660793, 0.36183763, 0.27137822, 0.13568911,
        0.18091881, 0.        , 0.18091881, 0.        , 0.0452297 ]),
 array([-2.94884687, -2.67247982, -2.39611276, -2.11974571, -1.84337865,
        -1.5670116 , -1.29064454, -1.01427749, -0.73791043, -0.46154338,
        -0.18517632,  0.09119073,  0.36755779,  0.64392484,  0.9202919 ,
         1.19665895,  1.47302601,  1.74939306,  2.02576012,  2.30212717,
         2.57849423]),
 <a list of 20 Patch objects>)
```



In [14]:
```
# plt.hist(ds1, normed=True, color="indianred", alpha=0.5, bins=20)
# matplotlib.pyplot.hist(x, bins=None, normed=None, color=None, alpha=0.5, bins=20)
# x : (n,) array or sequence of (n,) arrays
# bins : integer or sequence or 'auto', optional
# alpha : float or None ( float (0.0 transparent through 1.0 opaque))
# normed
```

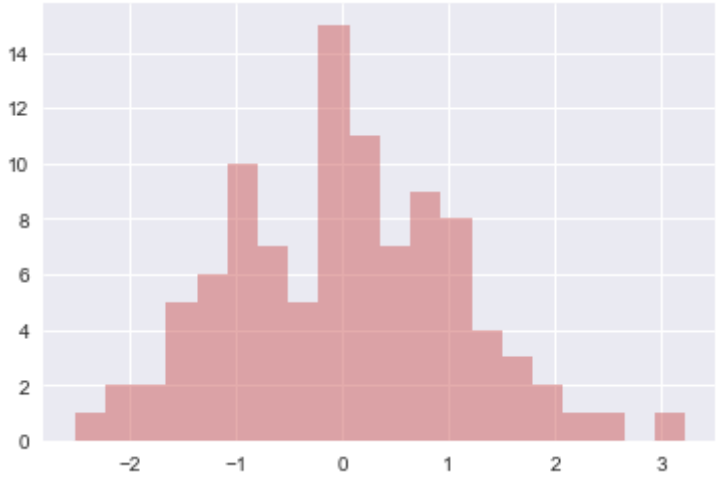
In [15]:
```
a = plt.hist(ds1, normed=True, color="indianred", alpha=0.5, bins=20)
a
```

Out[15]:
```
(array([ 0.03487884,  0.06975768,  0.06975768,  0.17439419,  0.20927303,
         0.34878839,  0.24415187,  0.17439419,  0.52318258,  0.38366723,
         0.24415187,  0.31390955,  0.27903071,  0.13951536,  0.10463652,
         0.06975768,  0.03487884,  0.03487884,  0.        ,  0.03487884]),
 array([-2.5132266 , -2.22651981, -1.93981302, -1.65310622, -1.36639943,
        -1.07969264, -0.79298585, -0.50627906, -0.21957227,  0.06713452,
         0.35384132,  0.64054811,  0.9272549 ,  1.21396169,  1.50066848,
         1.78737527,  2.07408207,  2.36078886,  2.64749565,  2.93420244,
         3.22090923]),
 <a list of 20 Patch objects>)
```



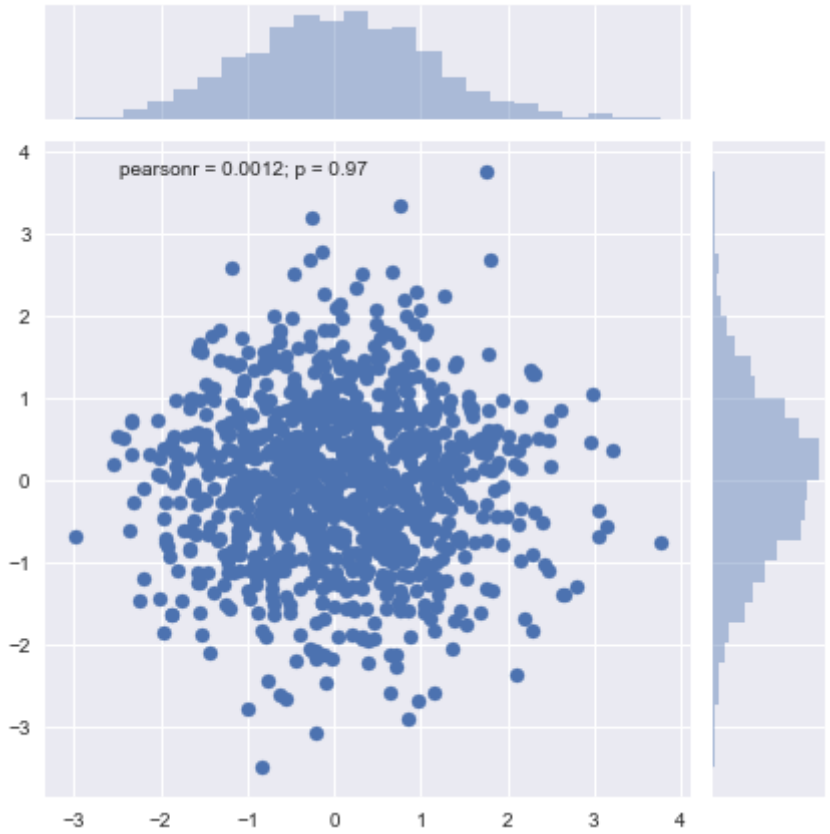
In [16]:
```
a = plt.hist(ds1, normed=False, color="indianred", alpha=0.5, bins=20)
a
```

Out[16]:
```
(array([  1.,   2.,   2.,   5.,   6.,  10.,   7.,   5.,  15.,  11.,   7.,
         9.,   8.,   4.,   3.,   2.,   1.,   1.,   0.,   1.]),
 array([-2.5132266 , -2.22651981, -1.93981302, -1.65310622, -1.36639943,
        -1.07969264, -0.79298585, -0.50627906, -0.21957227,  0.06713452,
         0.35384132,  0.64054811,  0.9272549 ,  1.21396169,  1.50066848,
         1.78737527,  2.07408207,  2.36078886,  2.64749565,  2.93420244,
         3.22090923]),
 <a list of 20 Patch objects>)
```



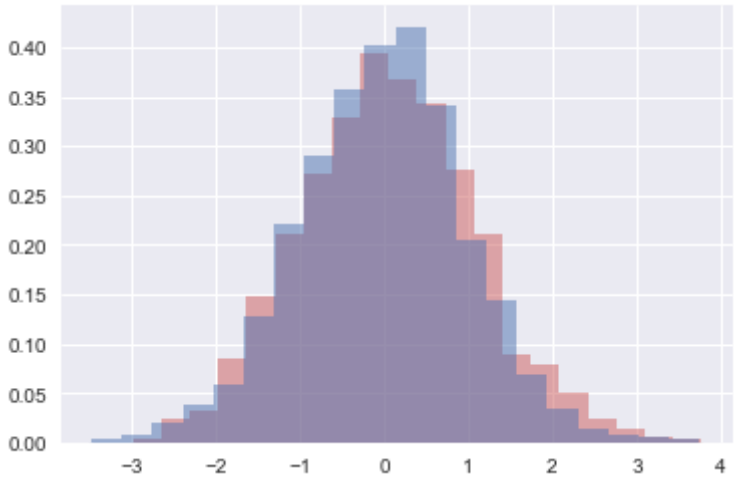
In [17]:
```
# p-value 와 분포도를 보여준다.
data1 = randn(1000)
data2 = randn(1000)
sns.jointplot(data1,data2)
```

Out[17]: <seaborn.axisgrid.JointGrid at 0x2a808662160>



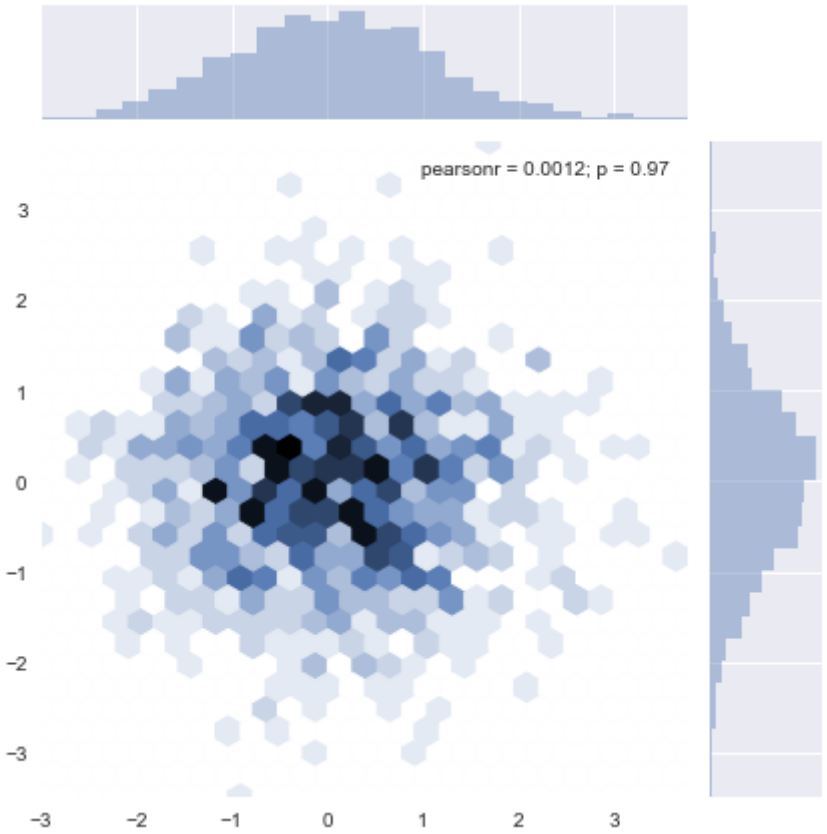
In [18]:
```
plt.hist(data1, normed=True, color="indianred", alpha=0.5, bins=20)
plt.hist(data2, normed=True, alpha=0.5, bins=20)
```

Out[18]:
```
(array([ 0.00276803,  0.00830409,  0.0193762 ,  0.0387524 ,  0.0581286 ,
         0.12732932,  0.22144229,  0.29064301,  0.3570757 ,  0.40136416,
         0.42074036,  0.34046753,  0.20483412,  0.14393749,  0.06920072,
         0.03321634,  0.01384014,  0.00830409,  0.00553606,  0.00276803]),
 array([-3.47871023, -3.1174423 , -2.75617436, -2.39490643, -2.0336385 ,
        -1.67237056, -1.31110263, -0.9498347 , -0.58856677, -0.22729883,
         0.1339691 ,  0.49523703,  0.85650497,  1.2177729 ,  1.57904083,
         1.94030876,  2.3015767 ,  2.66284463,  3.02411256,  3.3853805 ,
         3.74664843]),
 <a list of 20 Patch objects>)
```



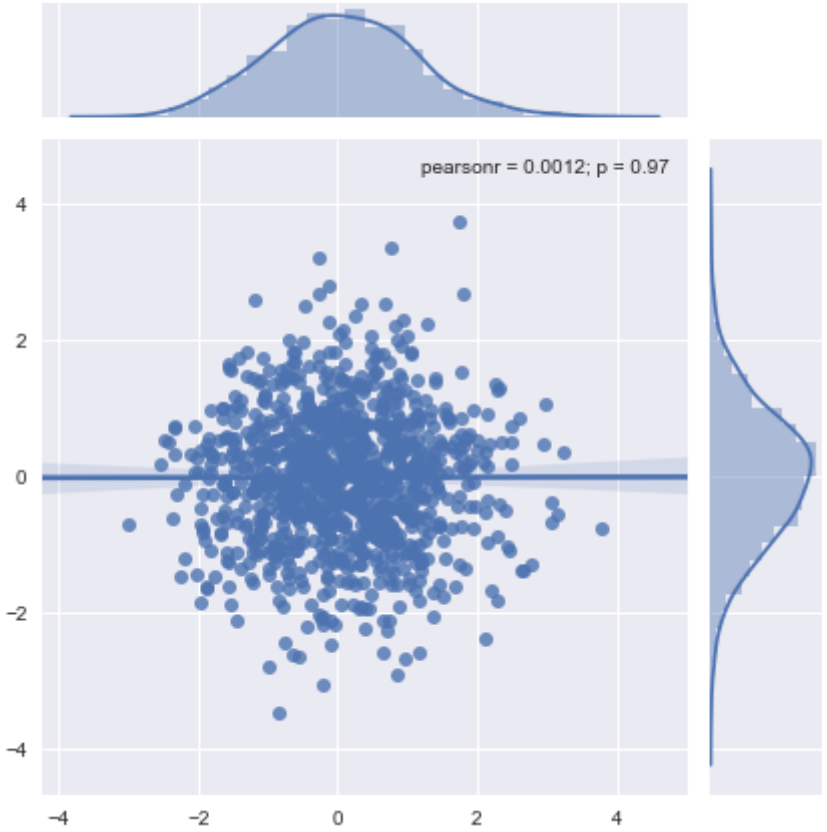
In [19]:
```
# 중앙 부분이 진하게 보이는 것을 확인해 볼 수 있다.
sns.jointplot(data1, data2,kind='hex')
```

Out[19]: <seaborn.axisgrid.JointGrid at 0x2a808685f28>



In [20]:
```
# 중앙 부분이 진하게 보이는 것을 확인해 볼 수 있다.
sns.jointplot(data1, data2,kind='reg')
```

Out[20]: <seaborn.axisgrid.JointGrid at 0x2a808d32780>



In [21]:
```
from scipy.stats import spearmanr
tips = sns.load_dataset("tips")
print(tips.shape)
print(tips.info())
print(tips.describe())
tips.head(10)
tips['sex'].head(10)
```

```
(244, 7)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 7 columns):
total_bill    244 non-null float64
tip           244 non-null float64
sex           244 non-null category
smoker        244 non-null category
day           244 non-null category
time          244 non-null category
size          244 non-null int64
dtypes: category(4), float64(2), int64(1)
memory usage: 7.2 KB
None
       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
```

Out[21]:
```
0    Female
1      Male
2      Male
3      Male
4    Female
5      Male
6      Male
7      Male
8      Male
```

```
9      Male
Name: sex, dtype: category
Categories (2, object): [Male, Female]
```

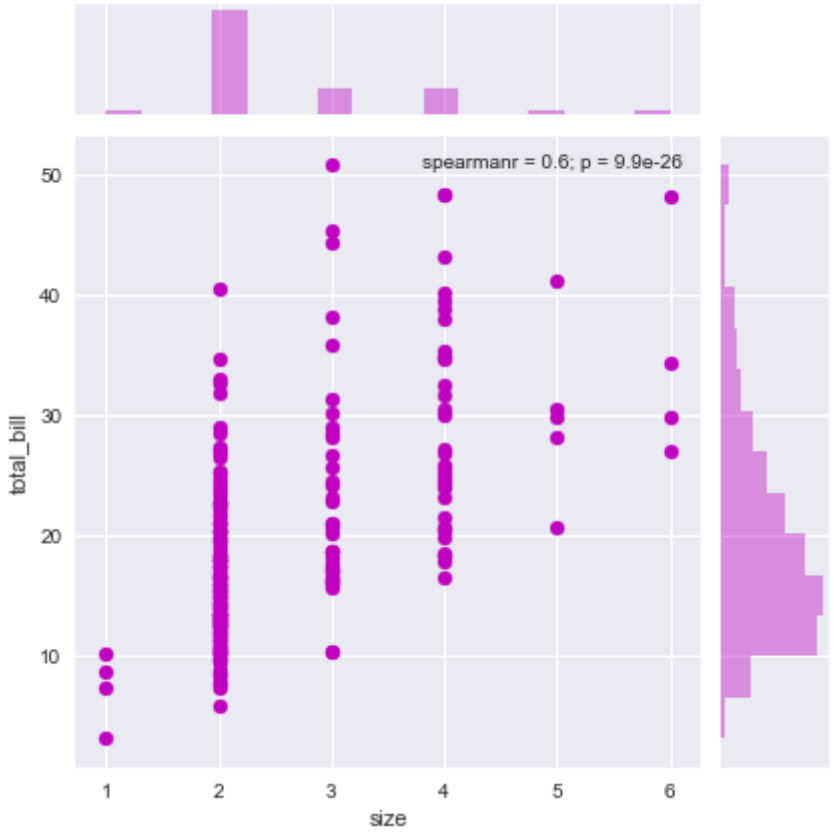
In [22]:
```
tips[['total_bill','size']].head()
```

Out[22]:

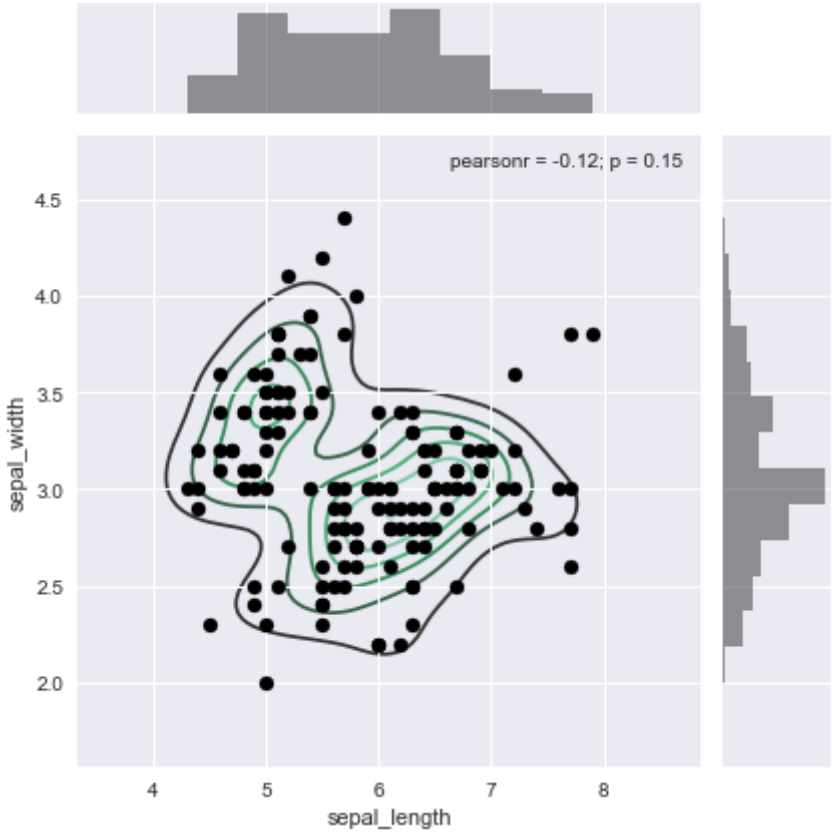
	total_bill	size
0	16.99	2
1	10.34	3
2	21.01	3
3	23.68	2
4	24.59	4

In [23]:

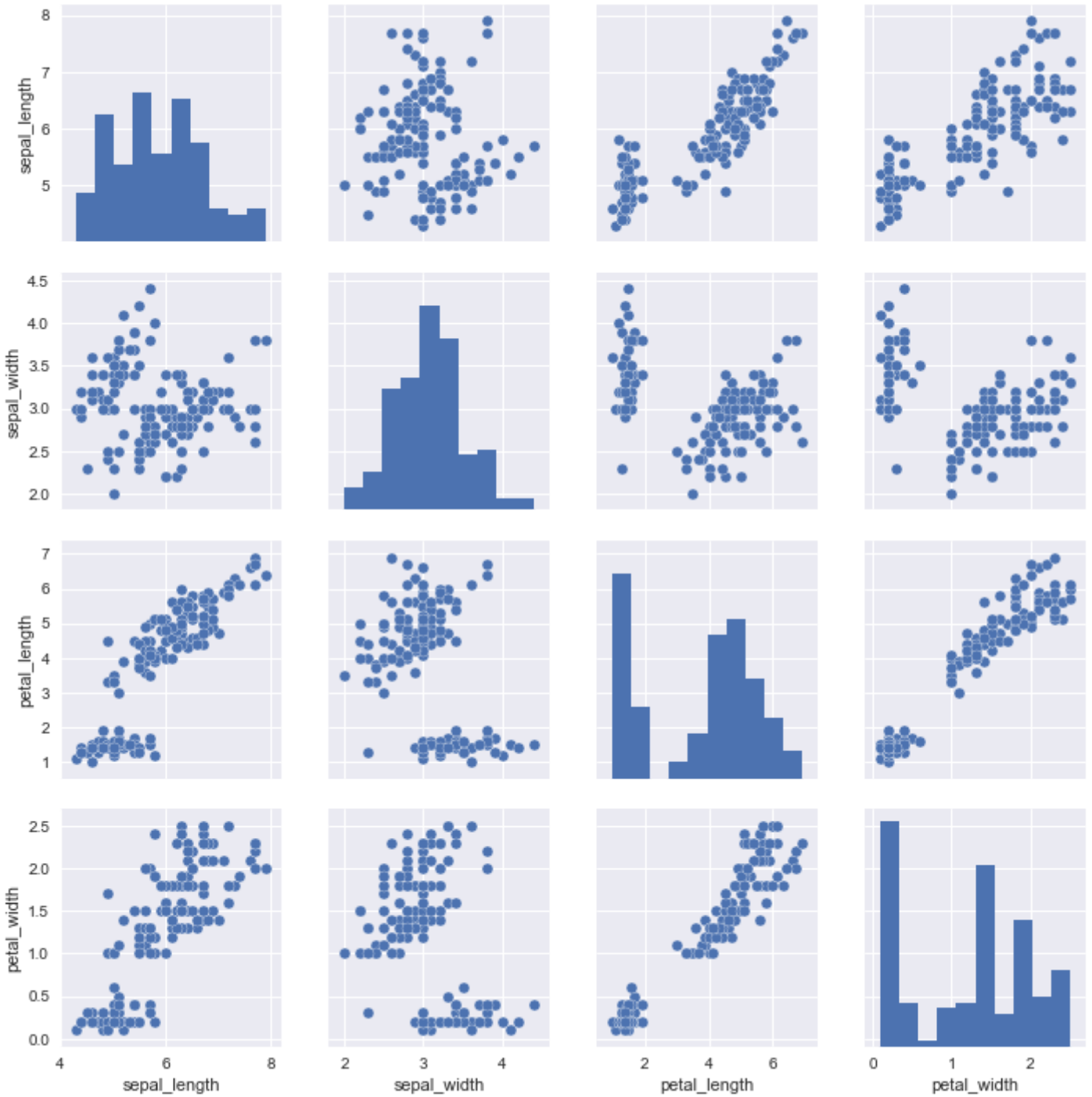
```
from scipy.stats import spearmanr
g = sns.jointplot("size", "total_bill", data=tips,
                  stat_func=spearmanr, color="m")
```



In [24]:
```
iris = sns.load_dataset("iris")
g = (sns.jointplot("sepal_length", "sepal_width",
                   data=iris, color="k")
                  .plot_joint(sns.kdeplot, zorder=0, n_levels=6))
```

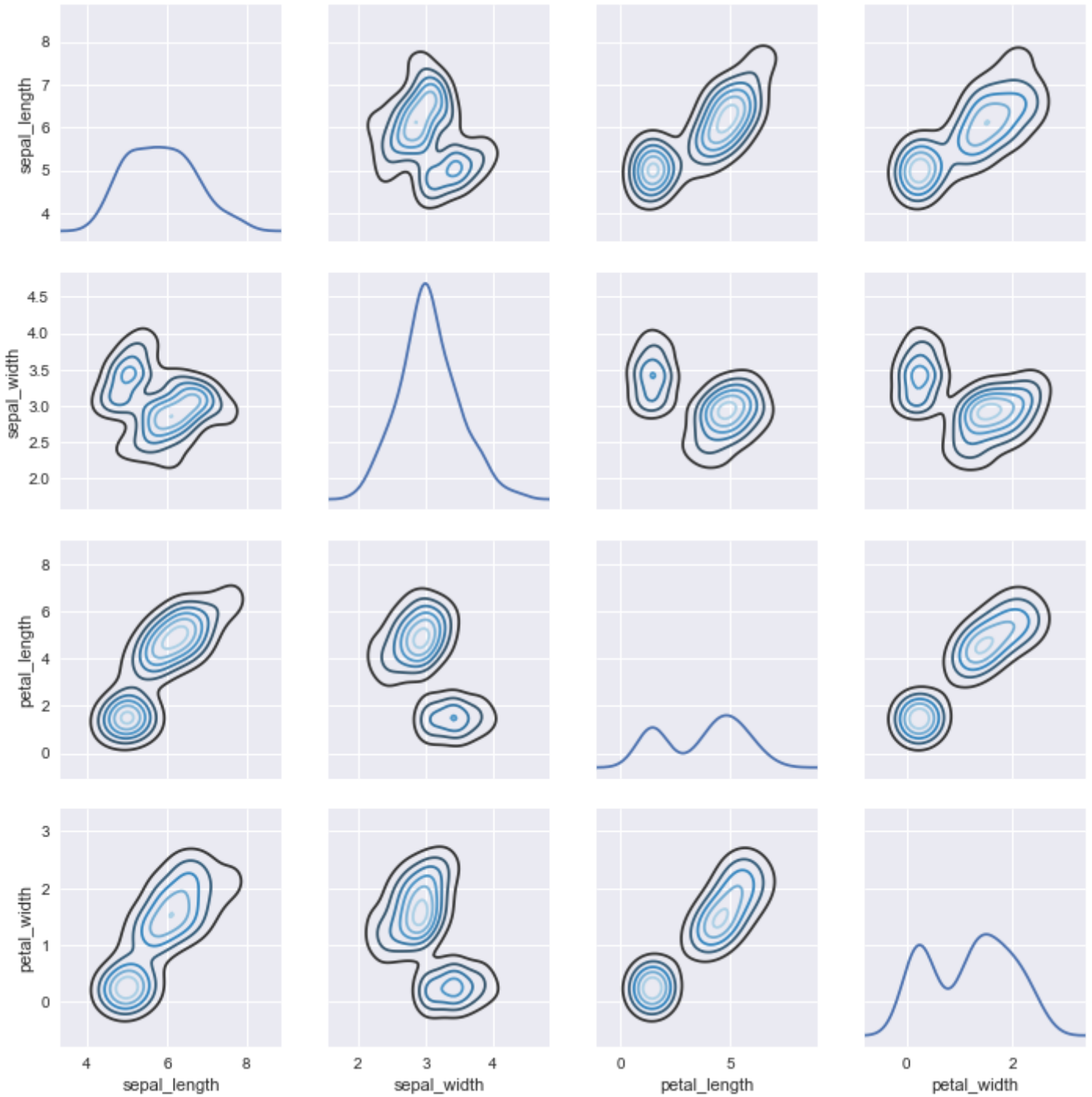


In [25]:
```
sns.pairplot(iris);
```


sepal_length
sepal_width
petal_length
petal_width
sepal_length
sepal_width
petal_length
petal_width

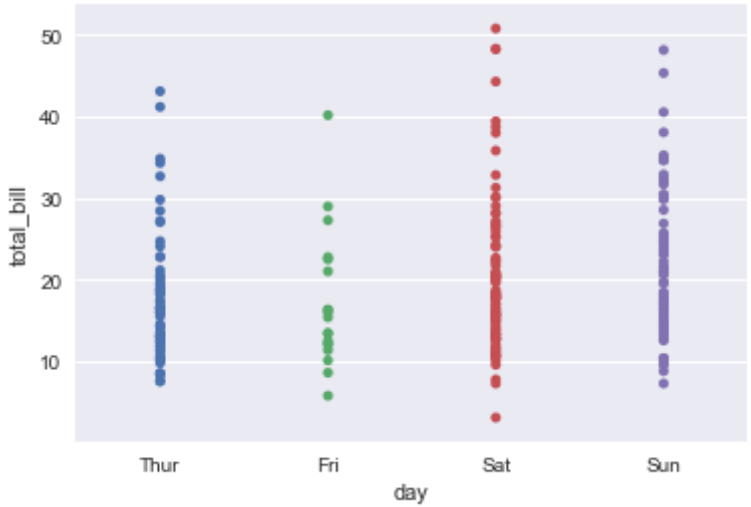
In [26]:
```
g = sns.PairGrid(iris)
g.map_diag(sns.kdeplot)
g.map_offdiag(sns.kdeplot, cmap="Blues_d", n_levels=6);
```

```
C:₩Anaconda3₩lib₩site-packages₩matplotlib₩axes₩_axes.py:545: UserWarning: No labelled objects found. Use label='...' kwarg on indi
vidual plots.
  warnings.warn("No labelled objects found. "
```

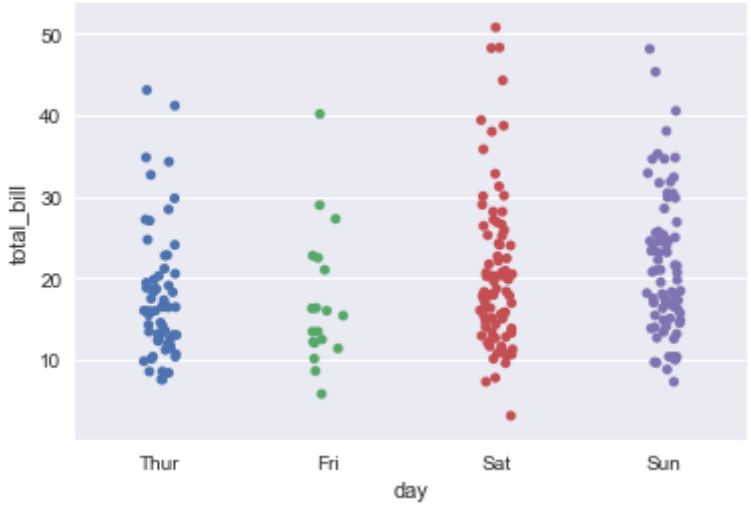

sepal_length
sepal_width
petal_length
petal_width
4
6
8
2
3
4
0
5
0
2

In [27]:
```
titanic = sns.load_dataset("titanic")
tips = sns.load_dataset("tips")
iris = sns.load_dataset("iris")
```

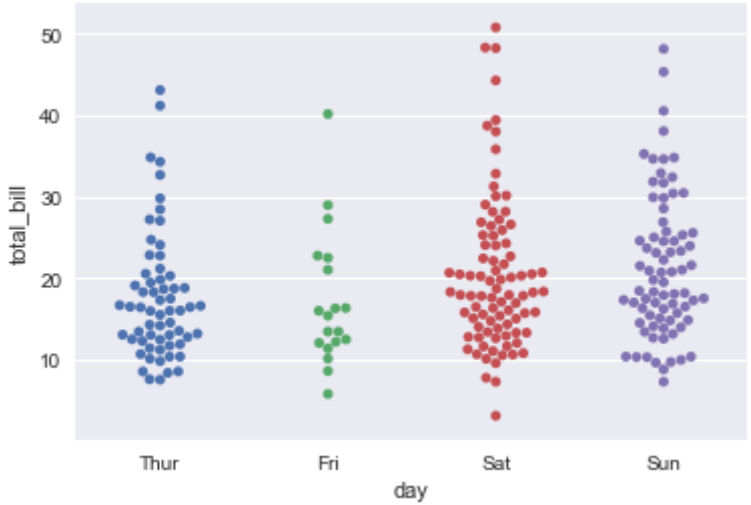
In [28]:
```
sns.stripplot(x="day", y="total_bill", data=tips);
```



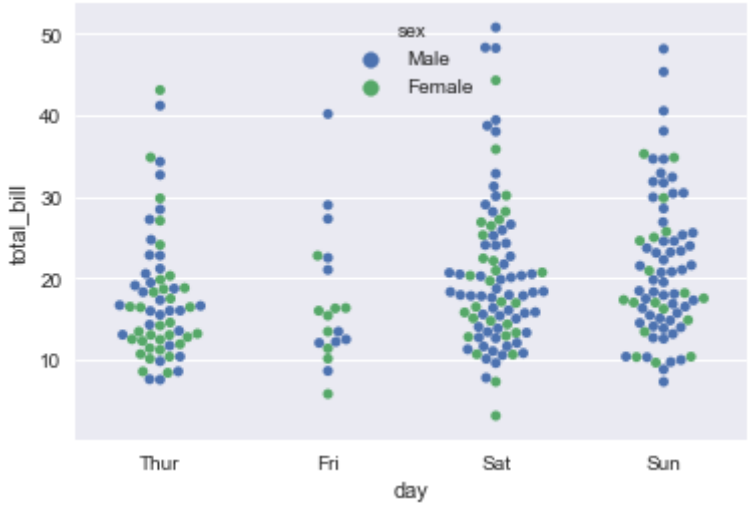
In [29]:
```
sns.stripplot(x="day", y="total_bill", data=tips, jitter=True);
```



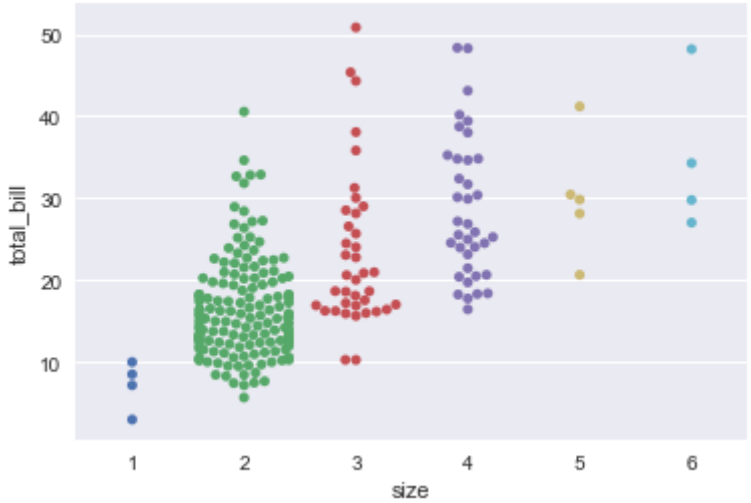
In [30]:
```
sns.swarmplot(x="day", y="total_bill", data=tips);
```



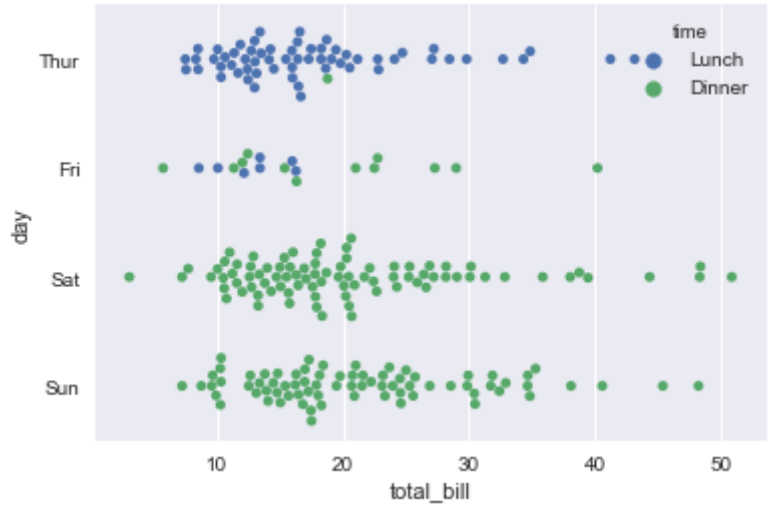
In [31]:
```
sns.swarmplot(x="day", y="total_bill", hue="sex", data=tips);
```



In [32]:
```
sns.swarmplot(x="size", y="total_bill", data=tips);
```



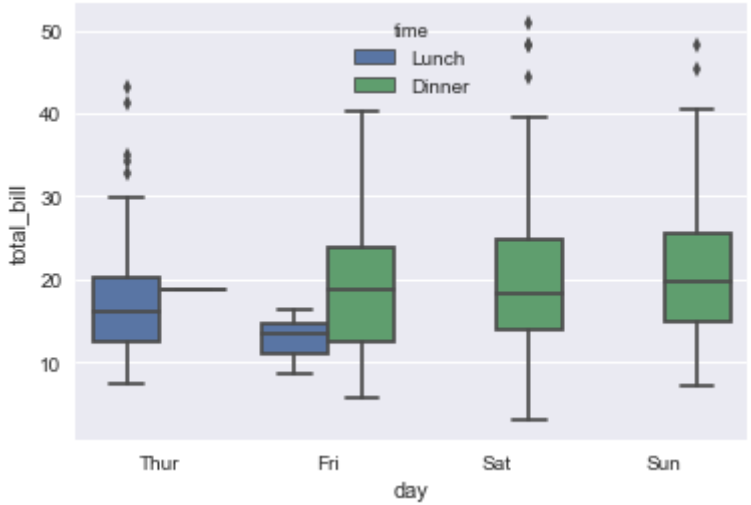
In [33]:
```
sns.swarmplot(x="total_bill", y="day", hue="time", data=tips);
```



Distributions of observations within categories¶

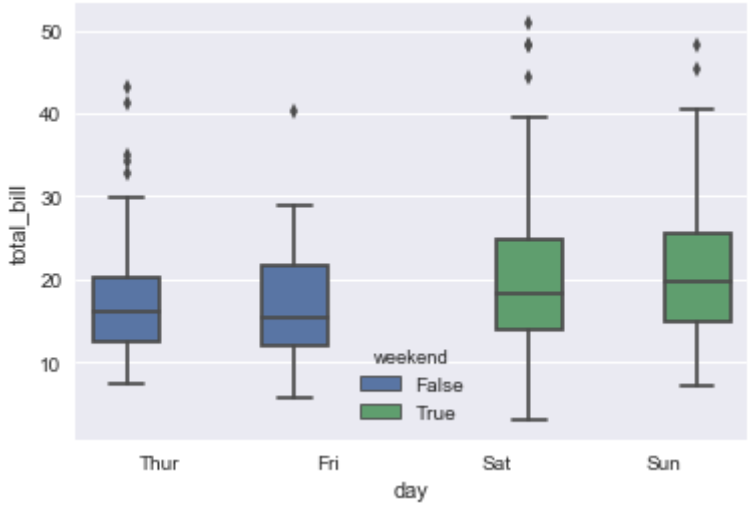
https://seaborn.pydata.org/tutorial/categorical.html (https://seaborn.pydata.org/tutorial/categorical.html)

In [34]:
```
sns.boxplot(x="day", y="total_bill", hue="time", data=tips);
```



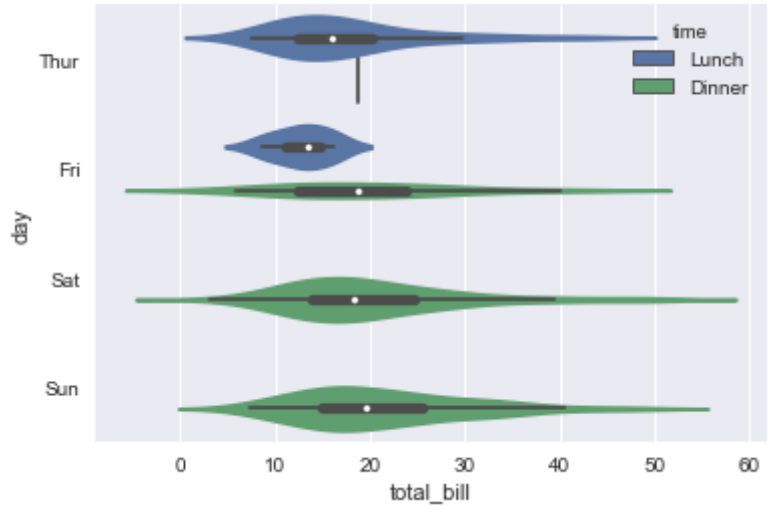
In [35]:
```
tips["weekend"] = tips["day"].isin(["Sat", "Sun"])
sns.boxplot(x="day", y="total_bill", hue="weekend", data=tips)
```

Out[35]: <matplotlib.axes._subplots.AxesSubplot at 0x2a80995d710>

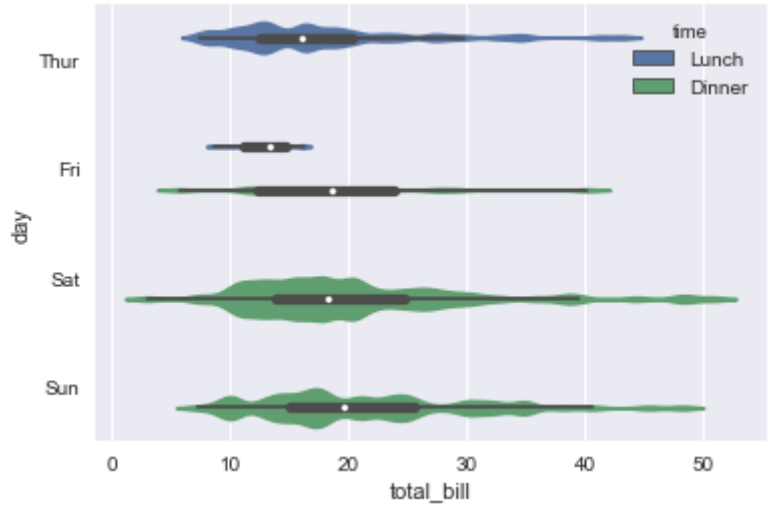


Violinplots

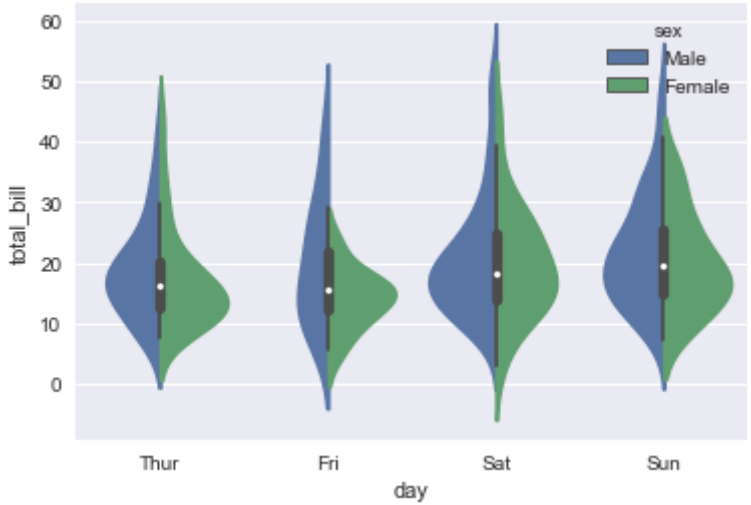
In [36]:
```
sns.violinplot(x="total_bill", y="day", hue="time", data=tips);
```



In [37]:
```
sns.violinplot(x="total_bill", y="day", hue="time", data=tips,
               bw=.1, scale="count", scale_hue=False);
```

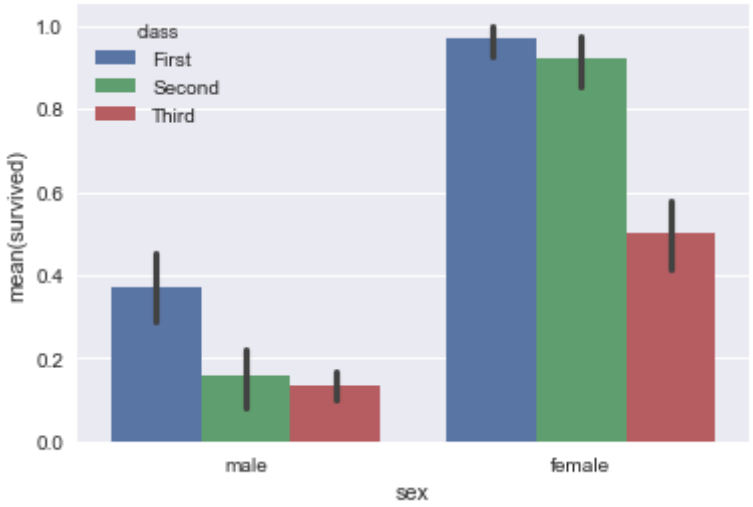


In [38]:
```
sns.violinplot(x="day", y="total_bill", hue="sex", data=tips, split=True);
```

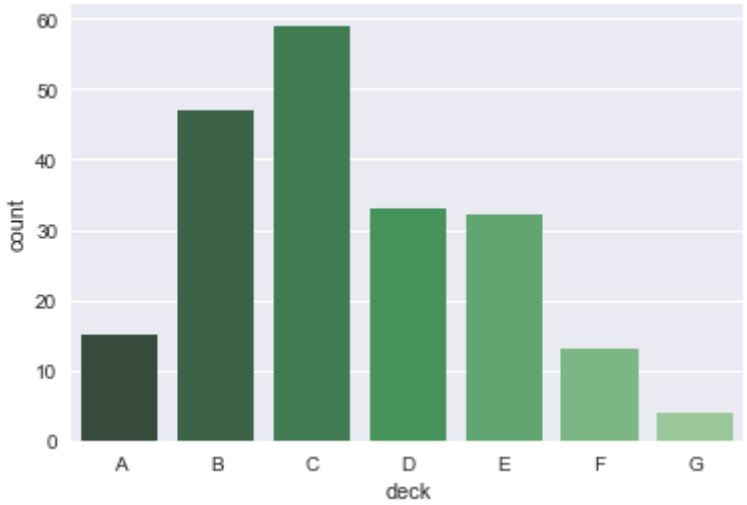


Bar plots

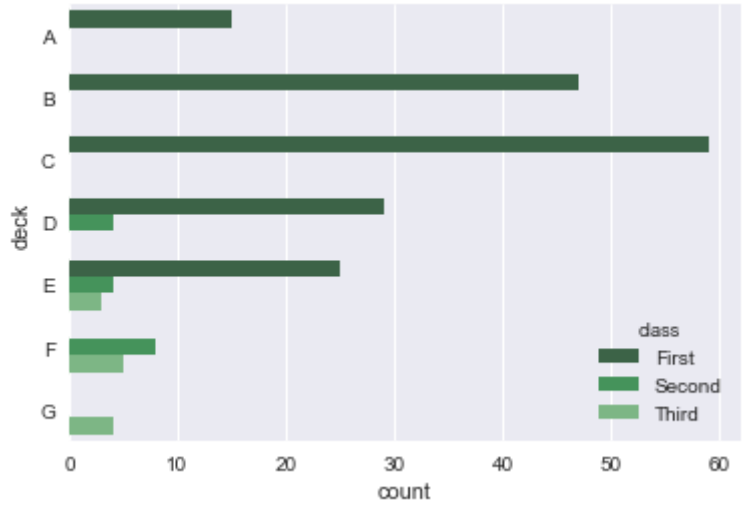
In [39]:
```
sns.barplot(x="sex", y="survived", hue="class", data=titanic);
```



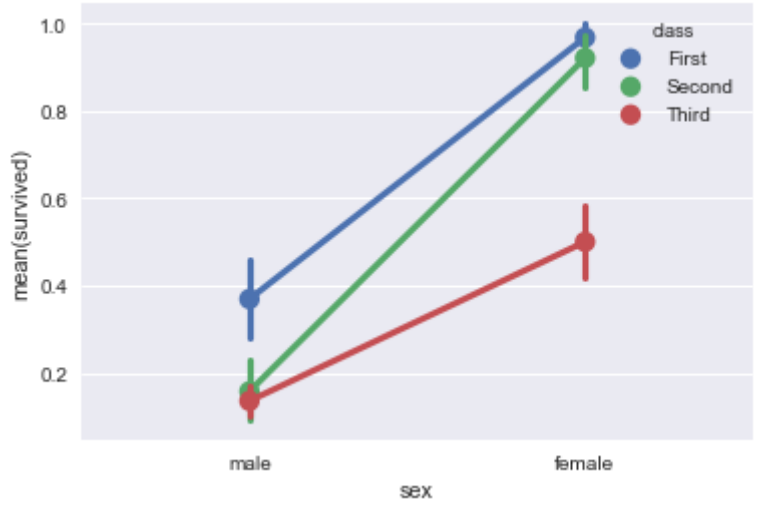
In [40]:
```
sns.countplot(x="deck", data=titanic, palette="Greens_d");
```



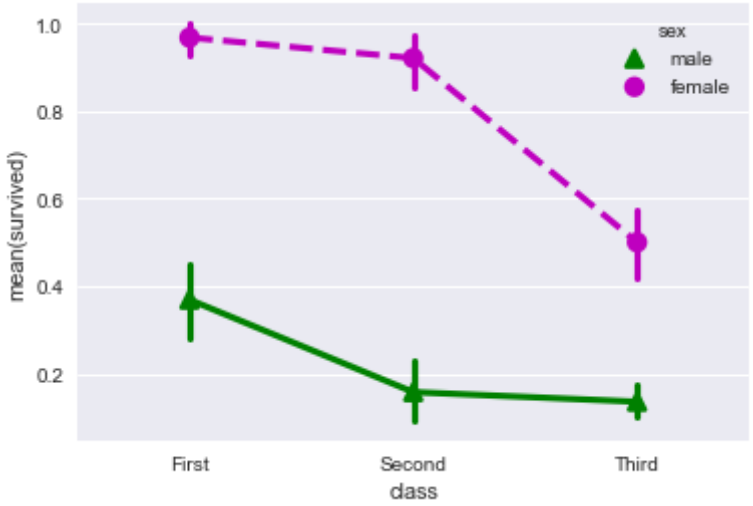
In [41]:
```
sns.countplot(y="deck", hue="class", data=titanic, palette="Greens_d");
```



In [42]:
```
### Point plots
sns.pointplot(x="sex", y="survived", hue="class", data=titanic);
```

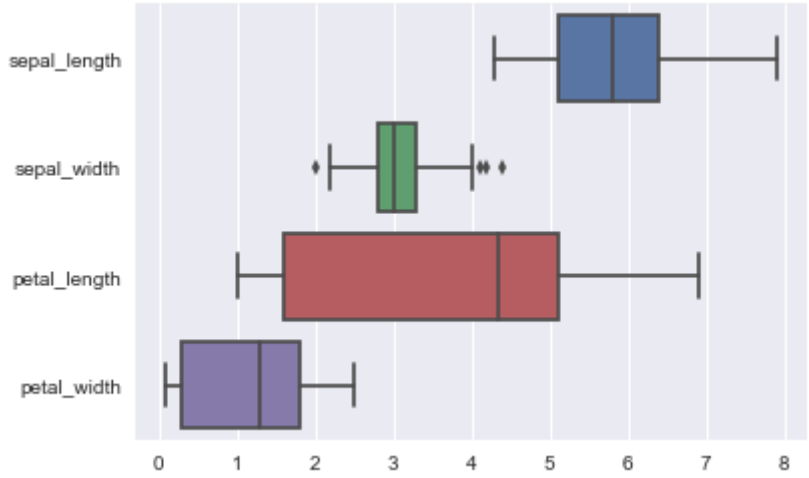


In [43]:
```
sns.pointplot(x="class", y="survived", hue="sex", data=titanic,
              palette={"male": "g", "female": "m"},
              markers=["^", "o"], linestyles=["-", "--"]);
```

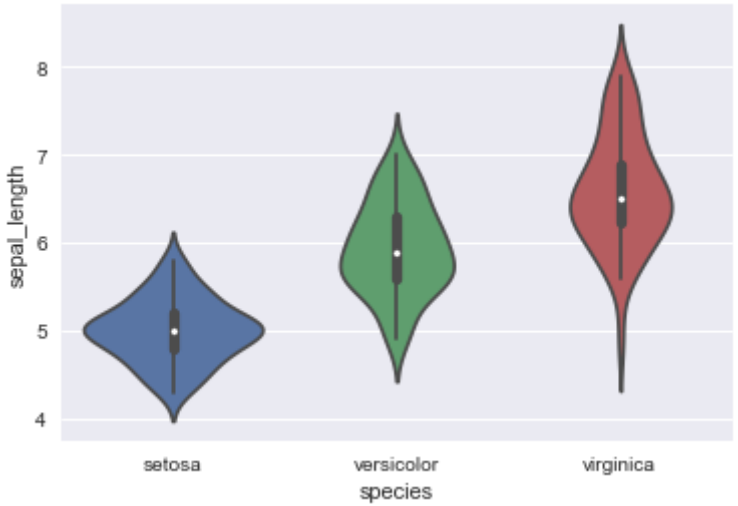


Plotting “wide-form” data

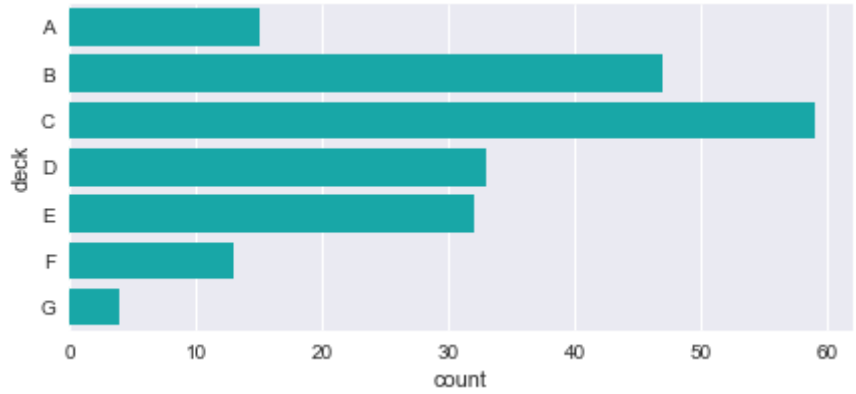
In [44]:
```
sns.boxplot(data=iris, orient="h");
```



In [45]:
```
sns.violinplot(x=iris.species, y=iris.sepal_length);
```

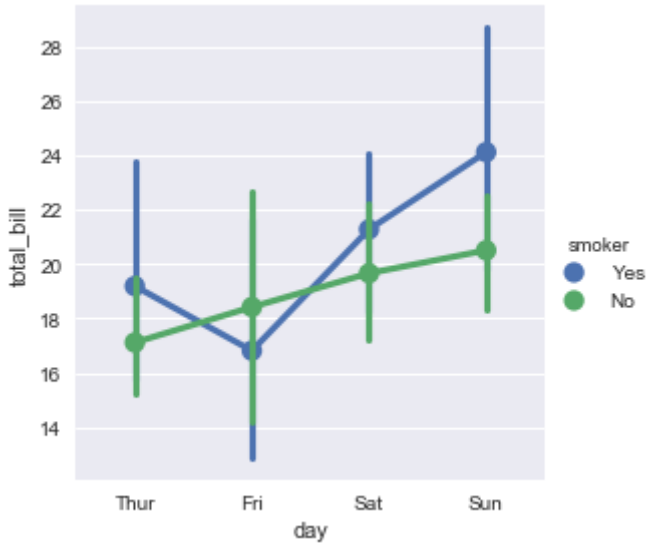


In [46]:
```
f, ax = plt.subplots(figsize=(7, 3))
sns.countplot(y="deck", data=titanic, color="c");
```

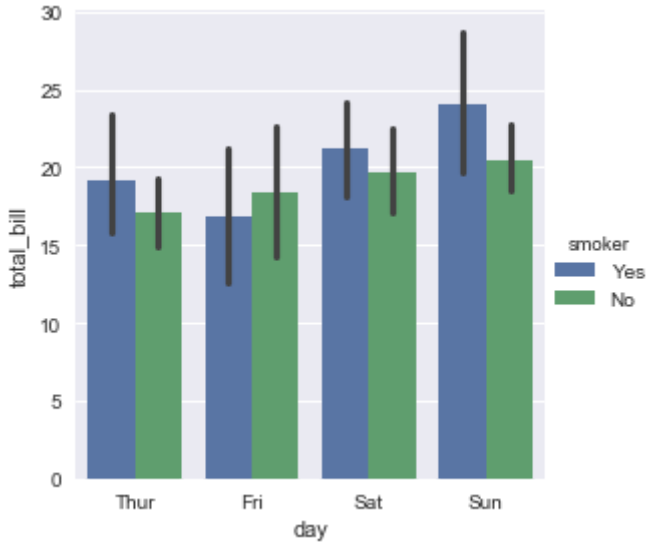


Drawing multi-panel categorical plots¶

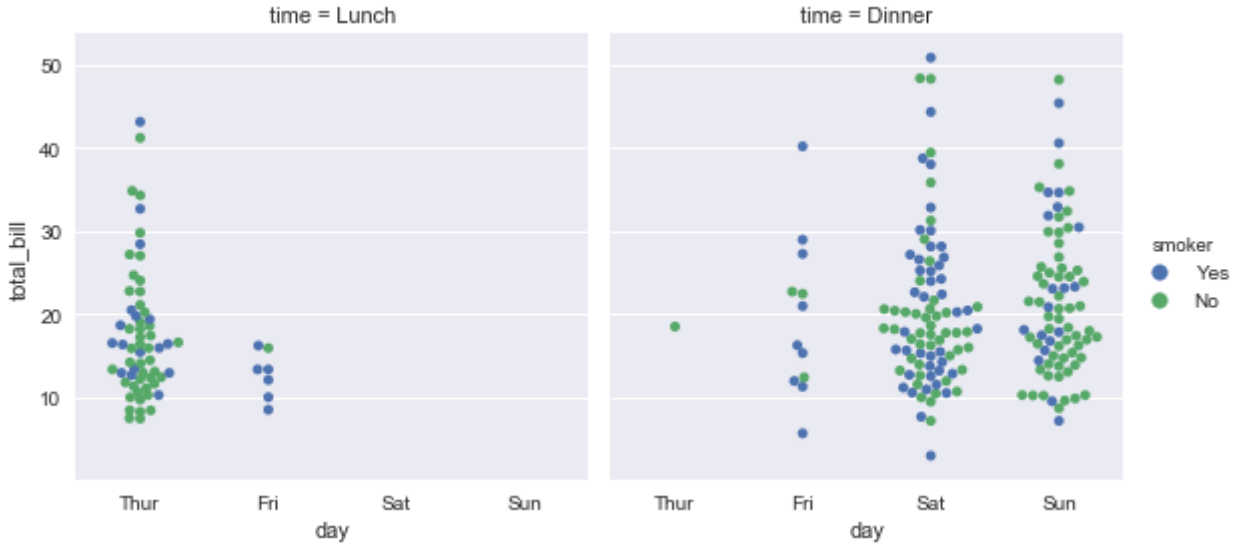
In [47]:
```
sns.factorplot(x="day", y="total_bill", hue="smoker", data=tips);
```



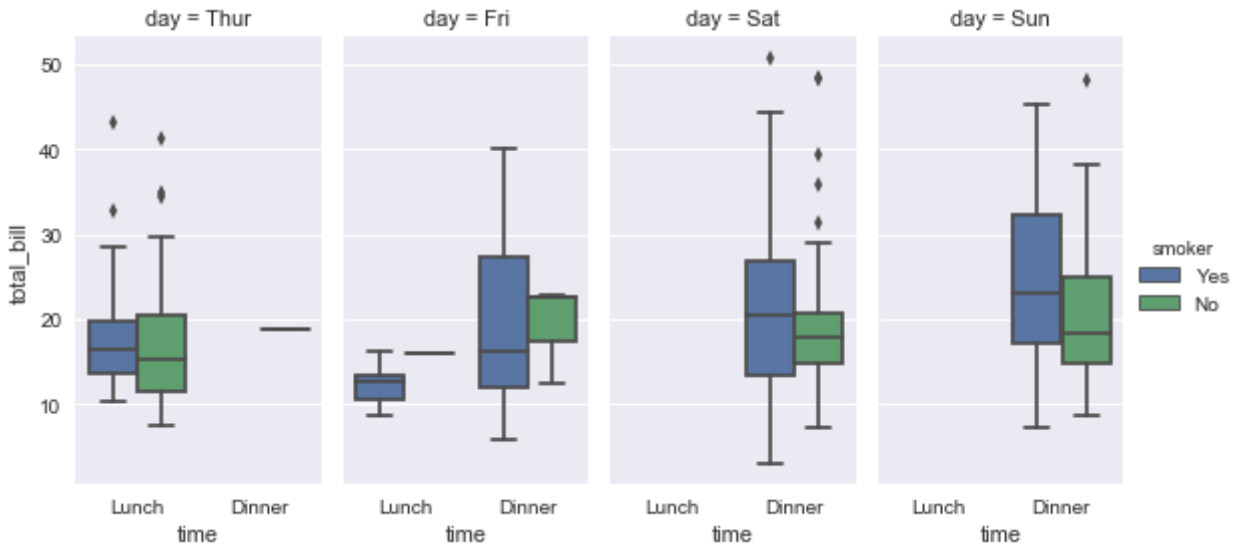
In [48]:
```
sns.factorplot(x="day", y="total_bill", hue="smoker", data=tips, kind="bar");
```



In [49]:
```
sns.factorplot(x="day", y="total_bill", hue="smoker",
               col="time", data=tips, kind="swarm");
```



In [50]:
```
sns.factorplot(x="time", y="total_bill", hue="smoker",
               col="day", data=tips, kind="box", size=4, aspect=.5);
```



In [ ]:
```
```