data_visualization_using_seaborn

April 22, 2018

0.0.1 Data Visualization using Seaborn (a Python library)

Tutorial by: Navie Narula, Digital Centers Teaching Intern

Created for: Research Data Services at Columbia University Libraries Resources used to create tutorial: DataCamp's Introductory Tutorial Pandey's Visualization Examples Seaborn PyData Swarm Plots Seaborn PyData Heat Maps List of Colors in Python

```python
In [3]: # import libraries
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn

        %matplotlib inline
```

The seaborn library has many in-house datasets. You may find them here. We'll be starting off with the tips dataset.

```python
In [4]: # load in data and save to a variable
        df = seaborn.load_dataset("tips")
```

```python
In [5]: # first five rows of dataset
        df.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
```

```python
In [6]: # last five rows of dataset
        df.tail()
```

```
Out[6]:      total_bill   tip     sex smoker   day    time  size
        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
```
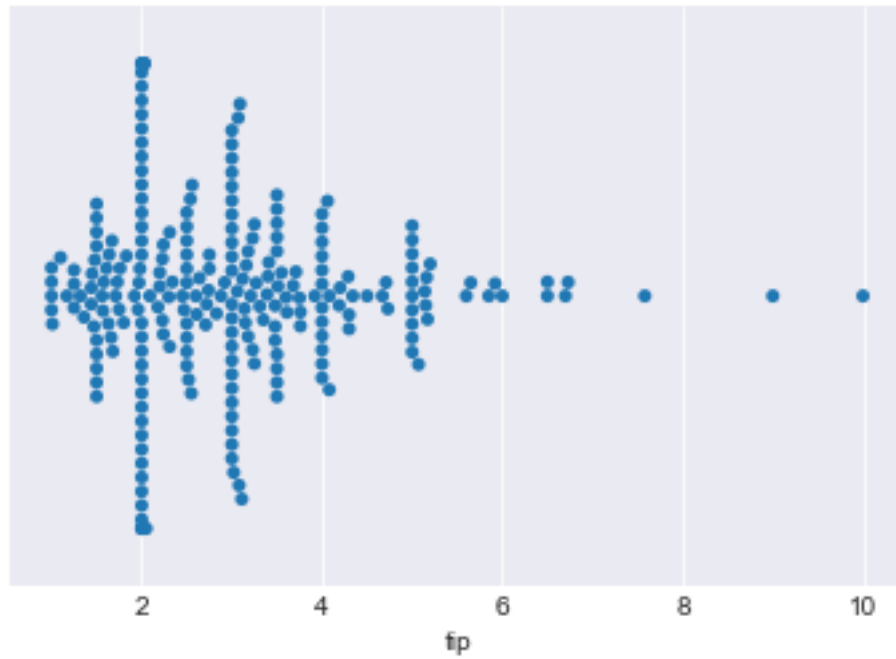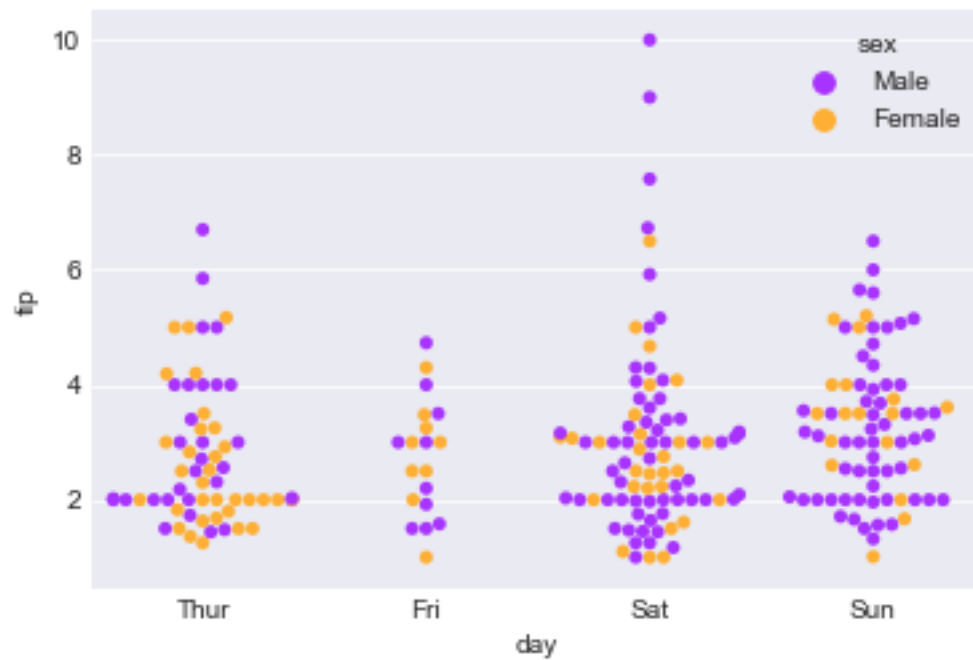
Swarm Plots

In [12]: `# use swarmplot to visualize tip observations and amounts`
`# by day of the week`
`seaborn.swarmplot(x="day", y="tip", data=df)`
`seaborn.set_style("whitegrid")`
`plt.show()`
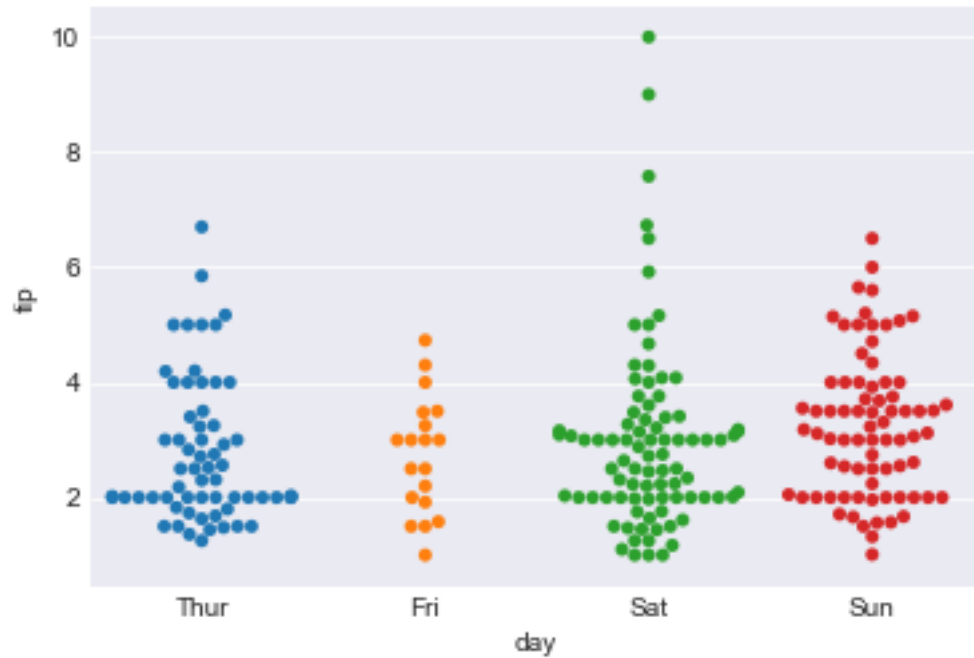
In [16]: `# visualize tip observations`
`seaborn.swarmplot(x=df["tip"])`
`seaborn.set_style("darkgrid")`
`plt.show()`

```
# color points by category
# create customized palette
gender_palette = ["#A833FF", "#FFAF33"]
seaborn.swarmplot(x="day", y="tip", hue="sex", palette=gender_palette, data=df)
plt.show()
```
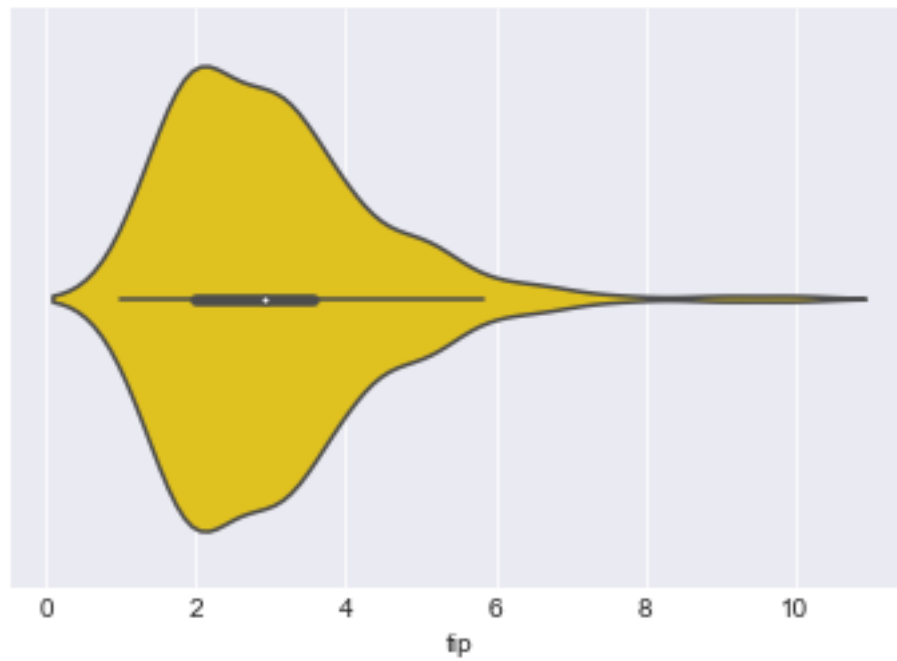
```
In [41]:  # control plot order on x-axis
          seaborn.swarmplot(x="smoker", y="total_bill", data=df, palette="husl", order=["Yes", "N
          plt.show()
```
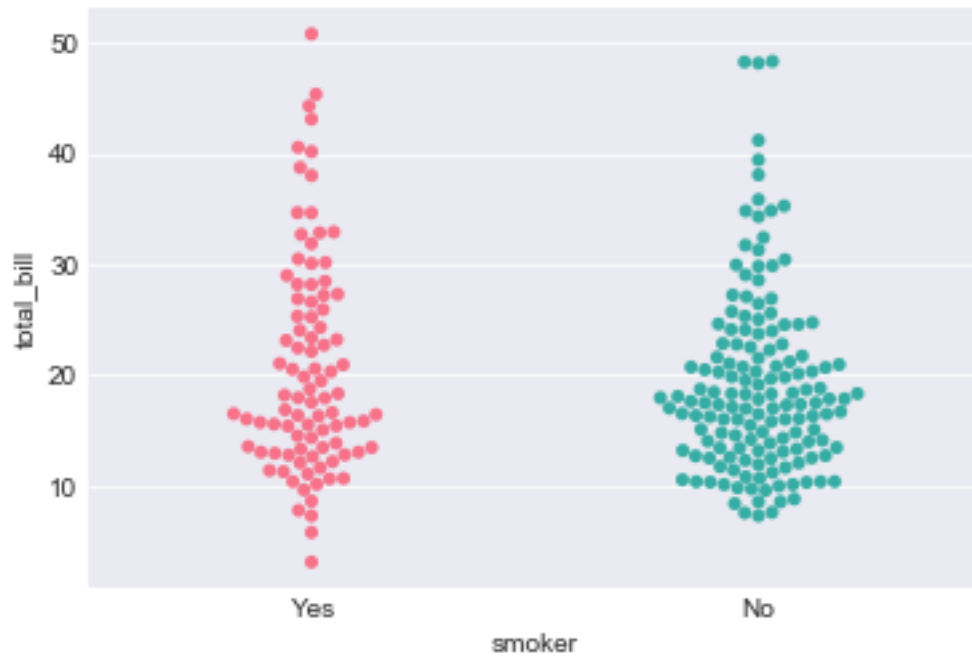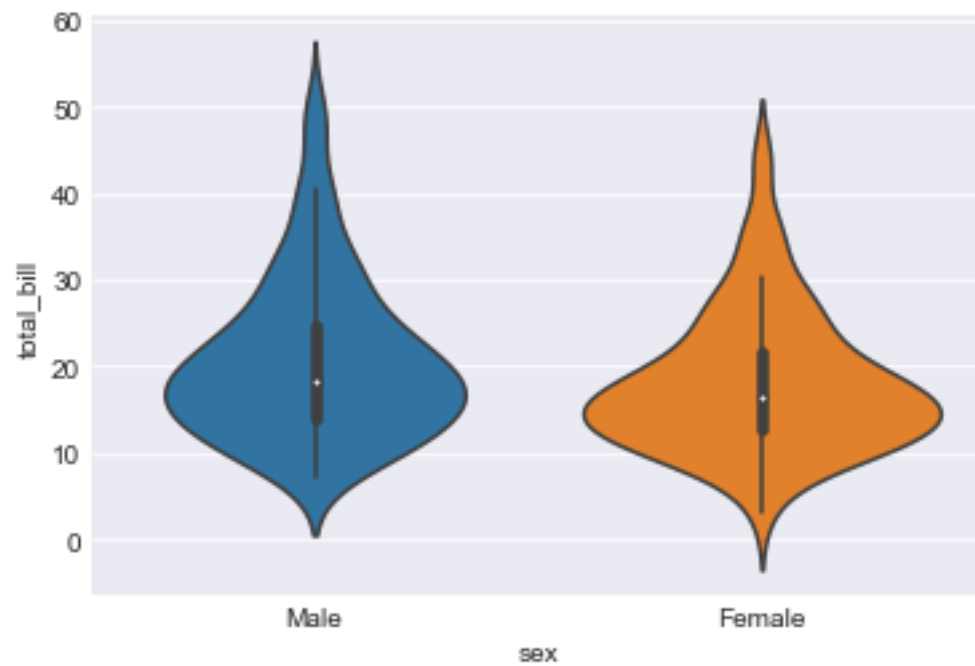

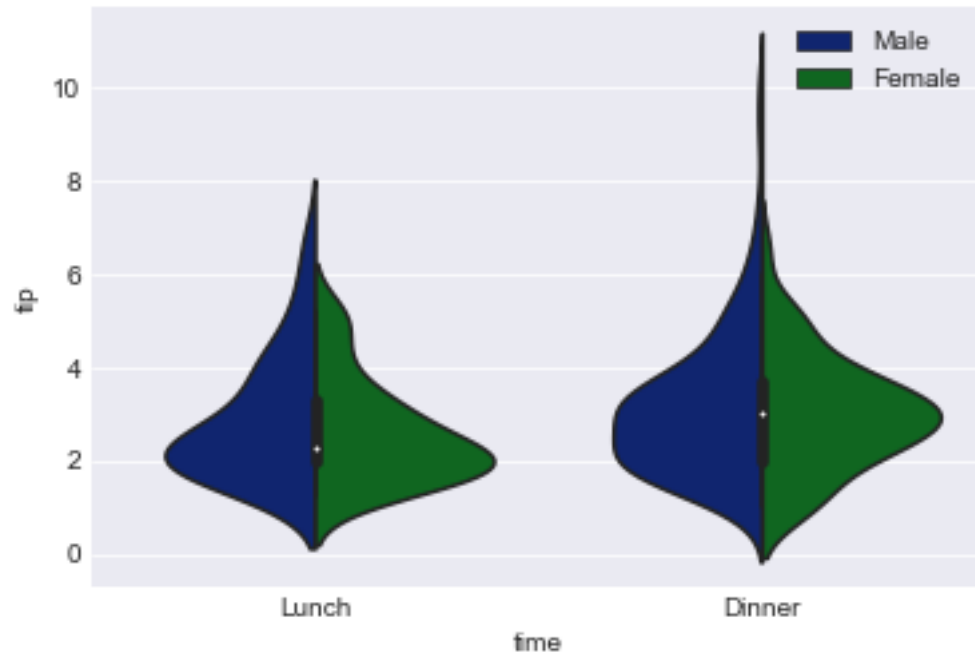
Violin Plots

```
In [50]:  # plot tips
          seaborn.violinplot(x = df["tip"], color="gold")
          plt.show()
```

tip

In [54]: # draw plot based on variable
seaborn.violinplot(x = "sex",y ="total_bill",data=df)
plt.show()

sex

5

```
In [57]: # Split drawings to compare with hue/legend variables
         seaborn.violinplot(x = "time",y ="tip",data=df, hue ="sex",palette ="dark",split = True
         plt.legend()
         plt.show()
```
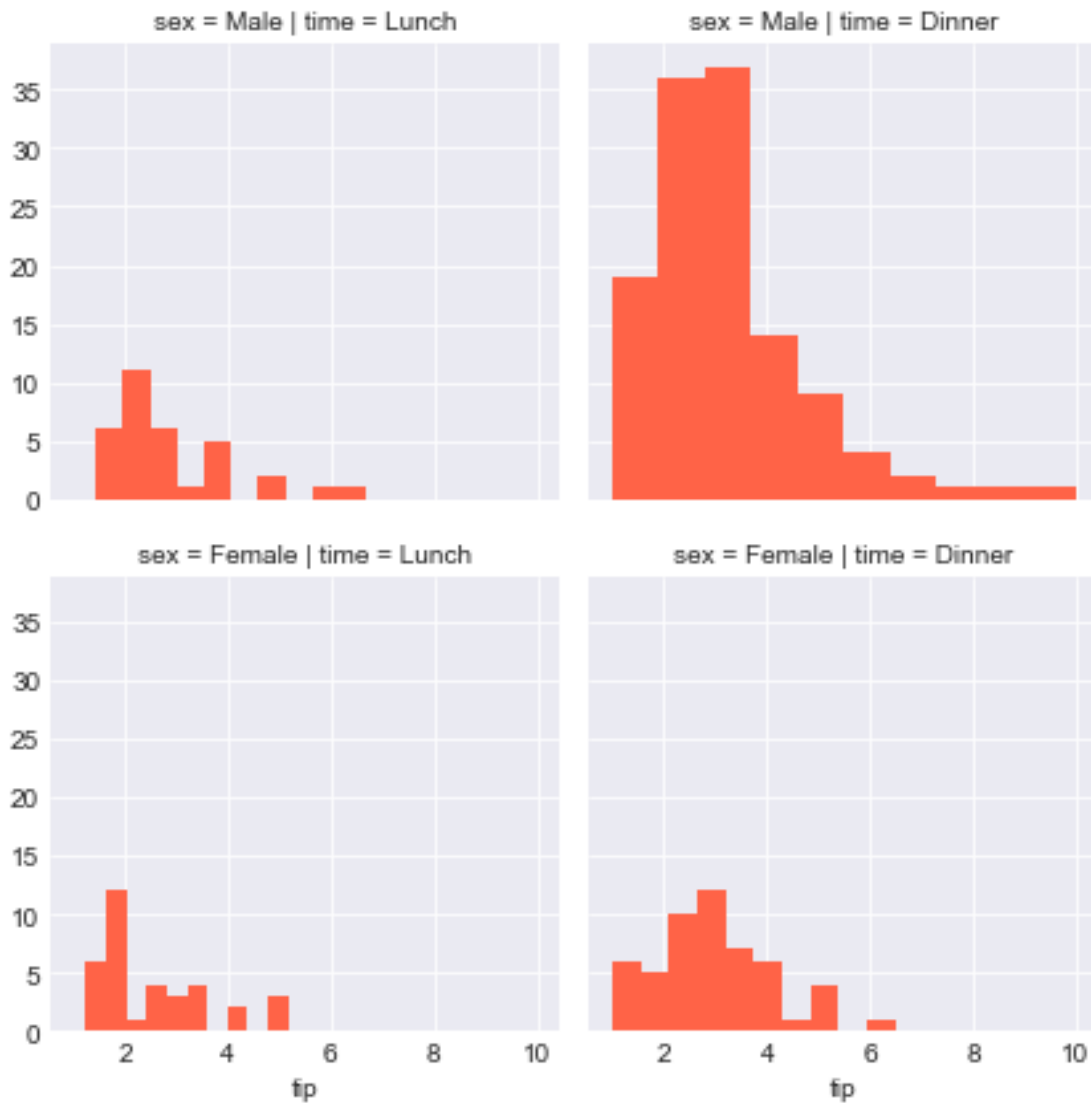

Facet Grids

```
In [72]: # draw facet grid based on tip variable
         fg = seaborn.FacetGrid(df,col = "time",row = "sex")
         fg = fg.map(plt.hist,"tip", color ="tomato")
```

In [73]: # we can also change the type of plot
         # ...and the colors around the points
         fg = seaborn.FacetGrid(df, col="time", row="sex")
         fg = fg.map(plt.scatter, "total_bill", "tip", color="floralwhite", edgecolor="hotpink")
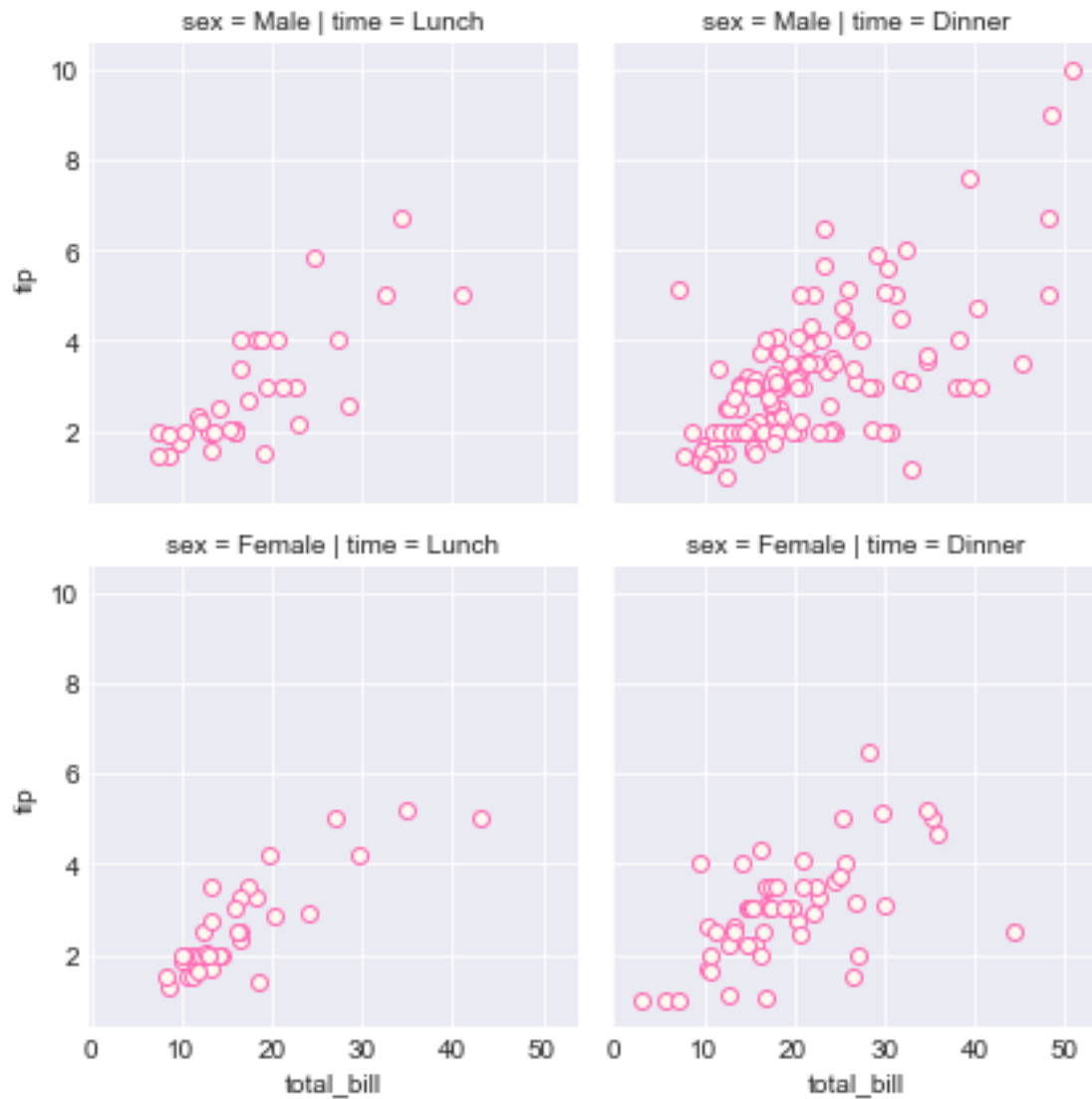
```
In [84]: # plot by category
         x = seaborn.FacetGrid(df, col="time",  hue="sex")
         x = x.map(plt.scatter,"total_bill","tip")
         x =x.add_legend()
```
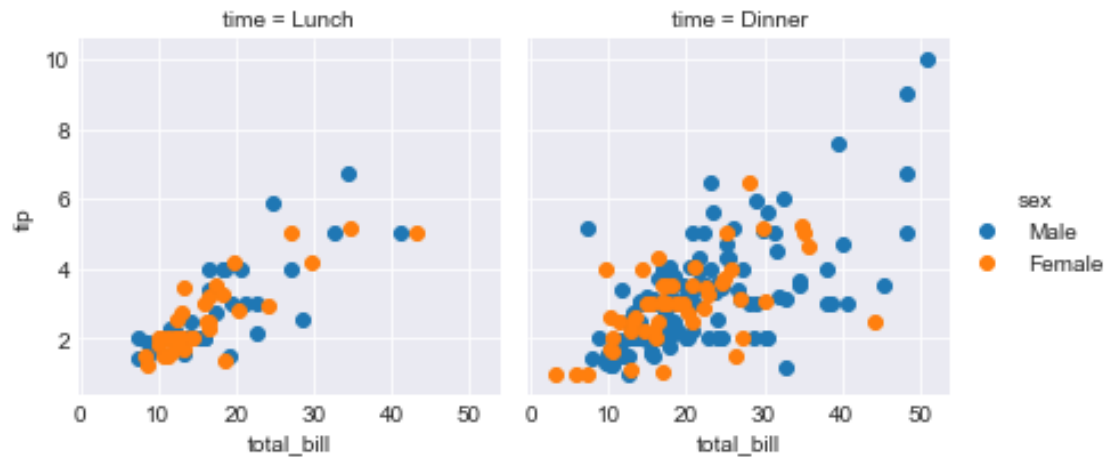
Heat Maps

```
In [146]: # create random data
          uniform_data = np.random.rand(5, 3) # five rows, 3 columns
          print(uniform_data)
          seaborn.heatmap(uniform_data)
          plt.show()
```

```
[[0.39376482 0.61566449 0.94105178]
 [0.57360765 0.66858876 0.03326495]
 [0.35962929 0.46553437 0.28784689]
 [0.32919801 0.02822342 0.38018925]
 [0.69303348 0.559752   0.61115946]]
```
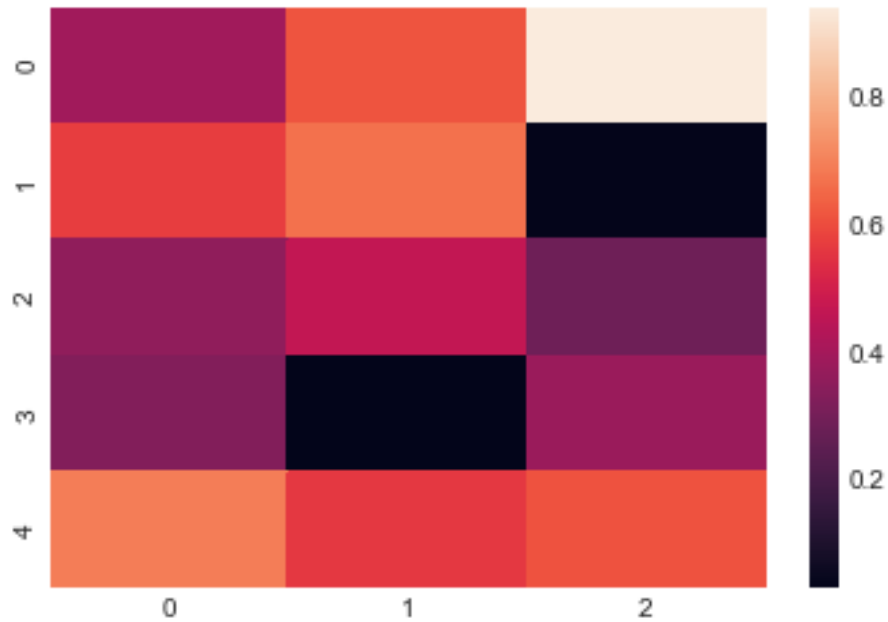
```
In [147]:  # load in flights dataset
           flights = seaborn.load_dataset("flights")

In [148]:  # print first five rows
           flights.head()

Out[148]:     year     month  passengers
           0  1949   January         112
           1  1949  February         118
           2  1949     March         132
           3  1949     April         129
           4  1949       May         121

In [149]:  # print last five rows
           flights.tail()

Out[149]:       year      month  passengers
           139  1960     August         606
           140  1960  September         508
           141  1960    October         461
           142  1960   November         390
           143  1960   December         432

In [150]:  flights = flights.pivot("month", "year", "passengers")
           # draw border
           x = seaborn.heatmap(flights, linewidths=0.3)
```
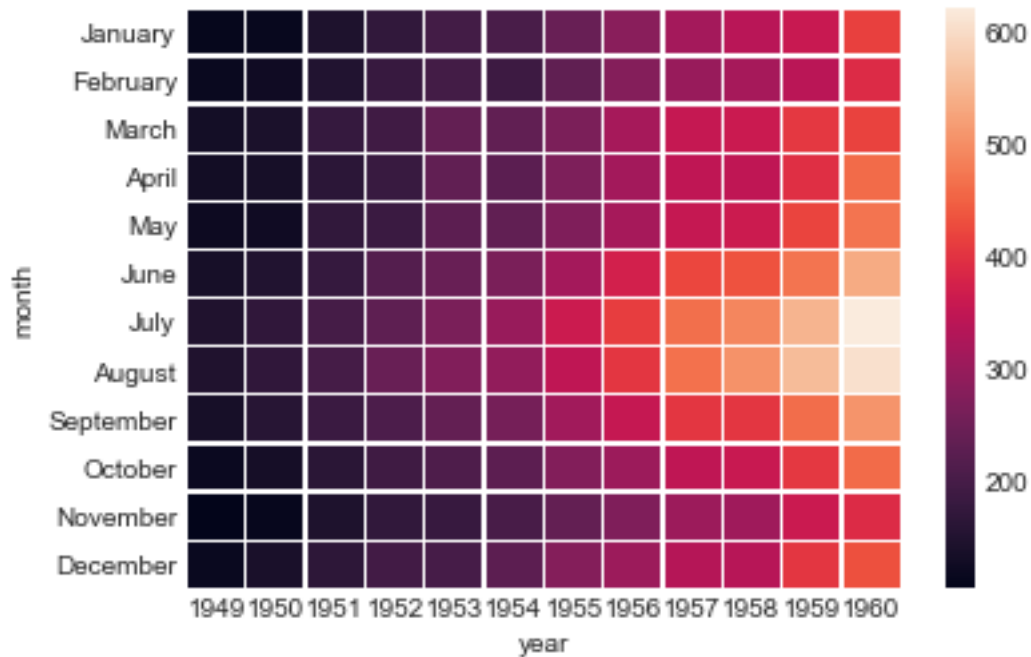
`# change color and add value`
`x = seaborn.heatmap(flights, annot=True, fmt="d", cmap="YlGnBu")`

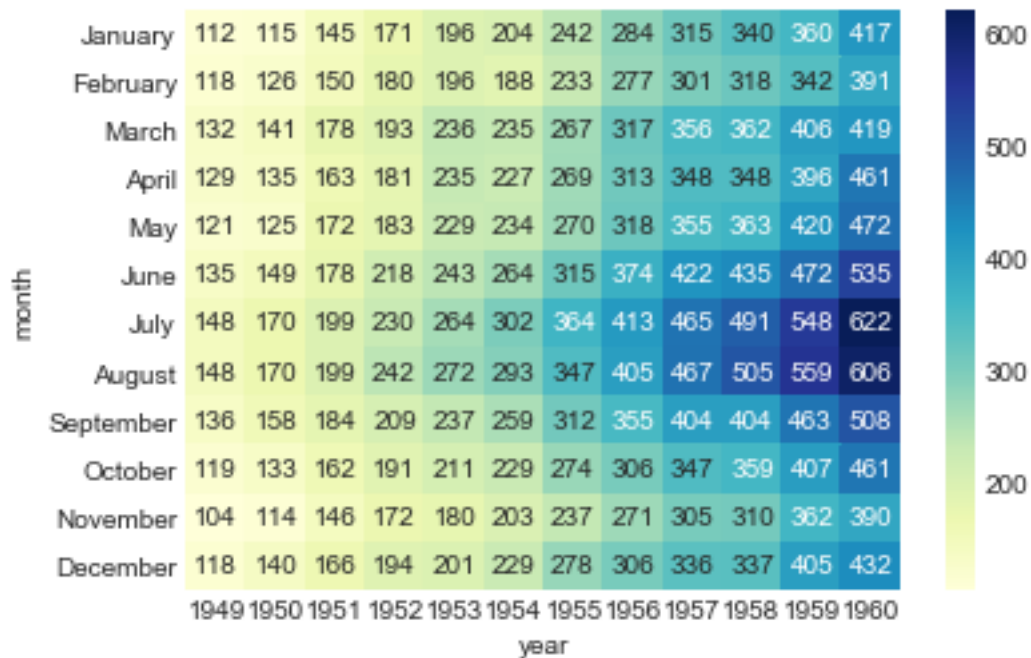|	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
January	112	115	145	171	196	204	242	284	315	340	360	417
February	118	126	150	180	196	188	233	277	301	318	342	391
March	132	141	178	193	236	235	267	317	356	362	406	419
April	129	135	163	181	235	227	269	313	348	348	396	461
May	121	125	172	183	229	234	270	318	355	363	420	472
June	135	149	178	218	243	264	315	374	422	435	472	535
July	148	170	199	230	264	302	364	413	465	491	548	622
August	148	170	199	242	272	293	347	405	467	505	559	606
September	136	158	184	209	237	259	312	355	404	404	463	508
October	119	133	162	191	211	229	274	306	347	359	407	461
November	104	114	146	172	180	203	237	271	305	310	362	390
December	118	140	166	194	201	229	278	306	336	337	405	432

Now, it's time for you to start working with your own data of choice and produce the visualizations you like! You can use one of seaborn's in-house datasets or load in your own. If you'd like to use in your own .csv file, you can load that into a dataframe by doing something like this:

```
import pandas as pd
df = pd.read_csv("<filename>", sep=",")
```

Source: https://blogs.cul.columbia.edu/spotlights/files/apr_23_2018.pdf