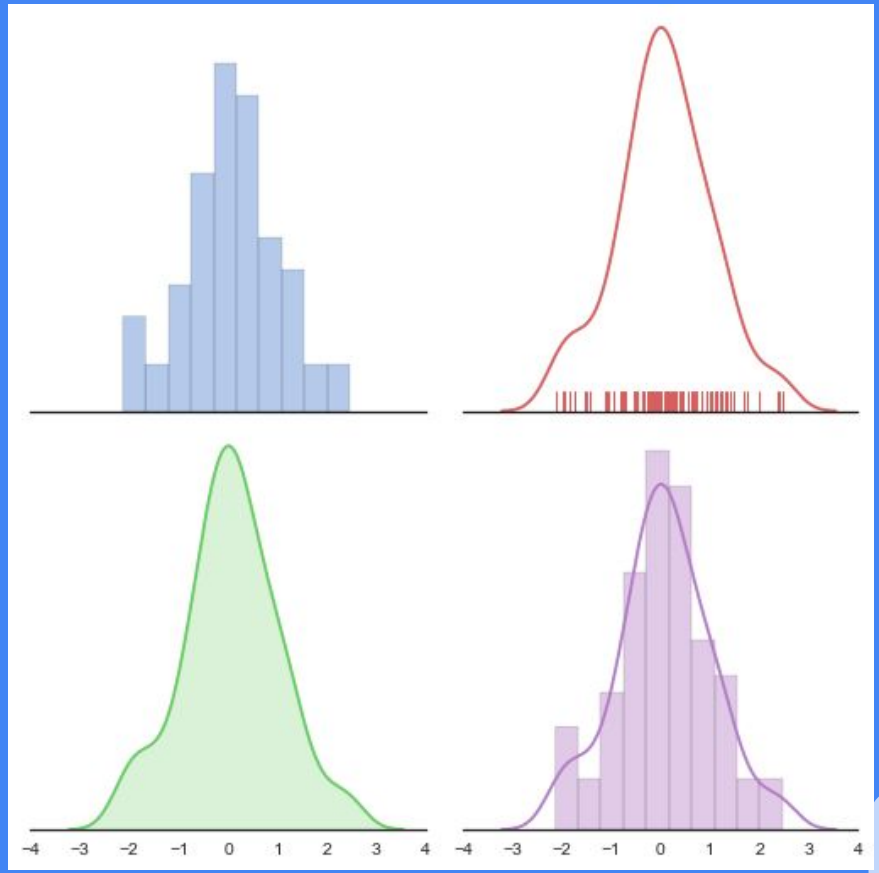
Seaborn

A library for data visualization



What is Seaborn?

- A high-level graphics interface for Python
- Complements matplotlib, not a replacement
- Tight integration with numpy and pandas
- Attractive graphics
- Statistical visualization

Where can I get it?

https://stanford.edu/~mwaskom/software/seaborn/

Installing in AFS space:

unc_anaconda

source activate astro

conda install seaborn

Dependencies:

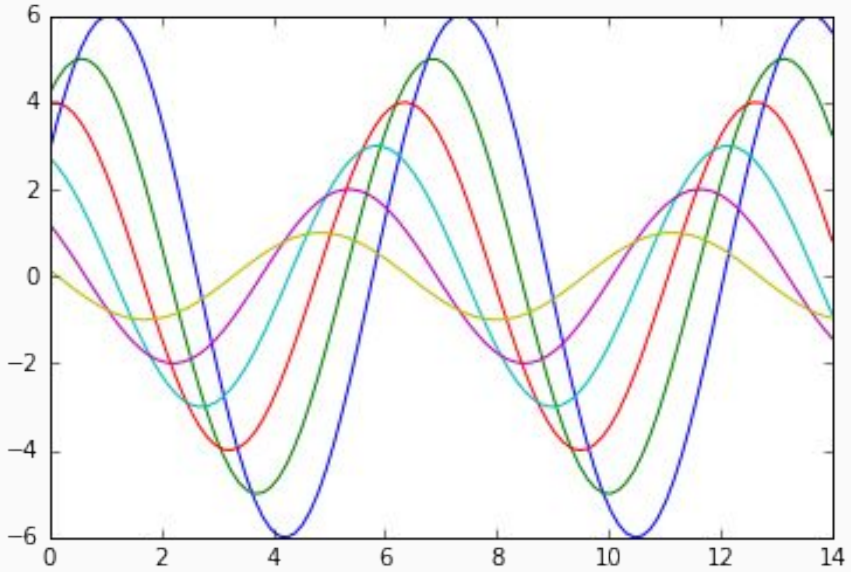
- numpy
- scipy
- matplotlib
- pandas

Recommended:

- statsmodels

Improving matplotlib plots

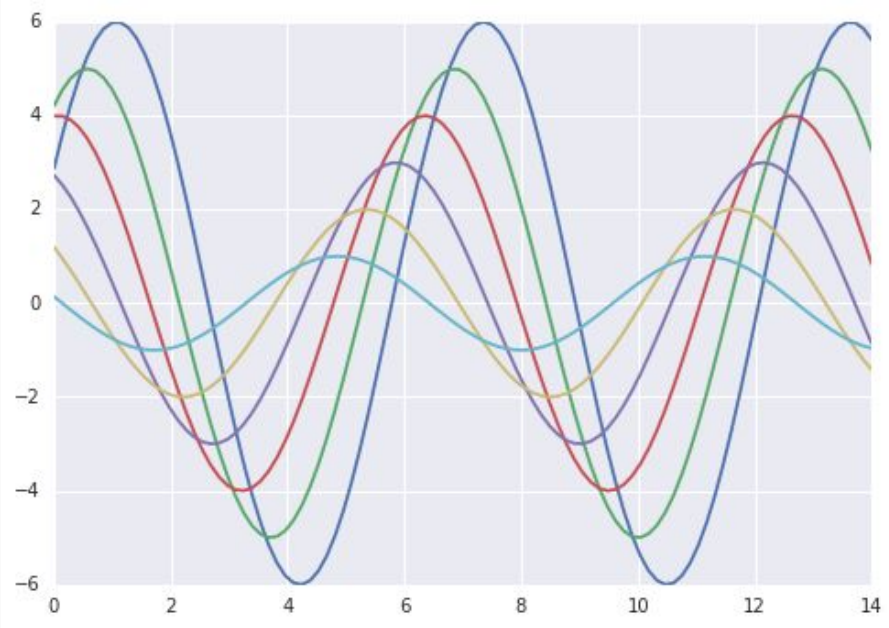
Matplotlib default



Improving matplotlib plots

Seaborn default

Themes available to control background color, grid, and axis appearance



Improving matplotlib plots

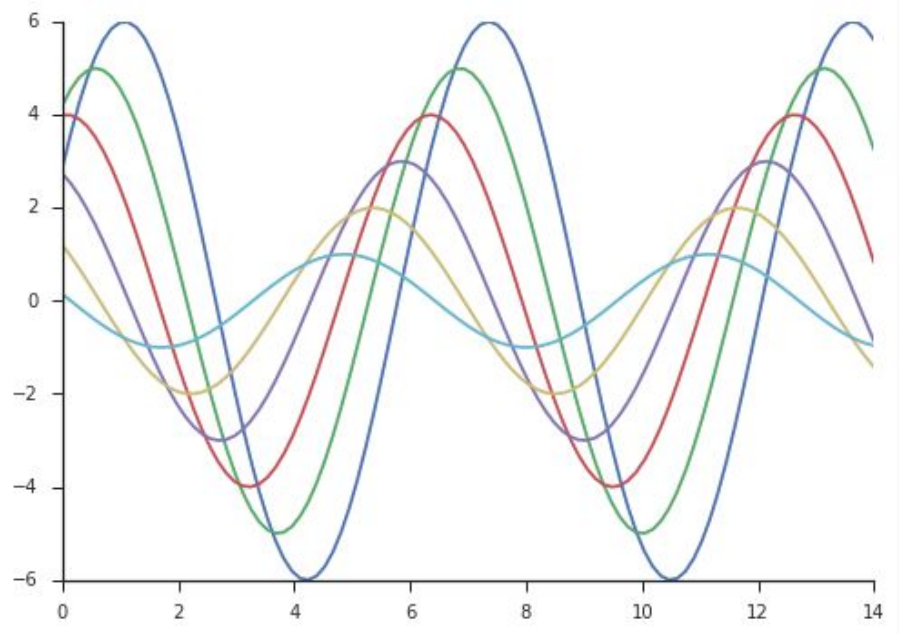
Control appearance of axes

import seaborn as sns

sns.set_style(“white”)

sns.set_style(“ticks”)

sns.despine()

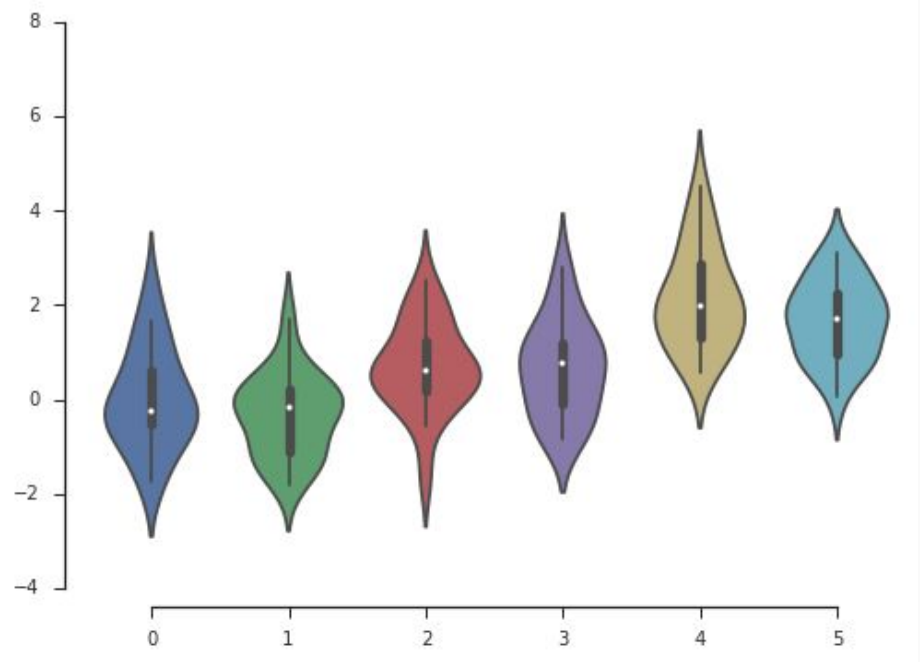


Improving matplotlib plots

Control appearance of axes

Add offset

sns.despine(offset=10, trim=True)



Improving matplotlib plots

Control scale of plot elements for publication

Allows use of same code for plots suited for different settings & sizes

Four preset contexts provided: notebook, paper, talk, poster (default: notebook)

```
sns.set_context("poster")
```

Color

Can reveal or hide patterns in data depending on how well it’s used

Seaborn provides powerful interface to color palettes for different types of data

sns.color_palette()

Color

Circular palette

For distinguishing categories without inherent ordering

sns.color_palette(“hls”,8)

Color

Sequential palette

Useful for data ranging from low/uninteresting to high/interesting values, ex: contour plots

sns.color_palette("Blues")

Color

Sequential palette

Cubehelix palette - linear change in brightness & hue, preserves information when converted to black & white

sns.cubehelix_palette(8)

Color

Diverging palette

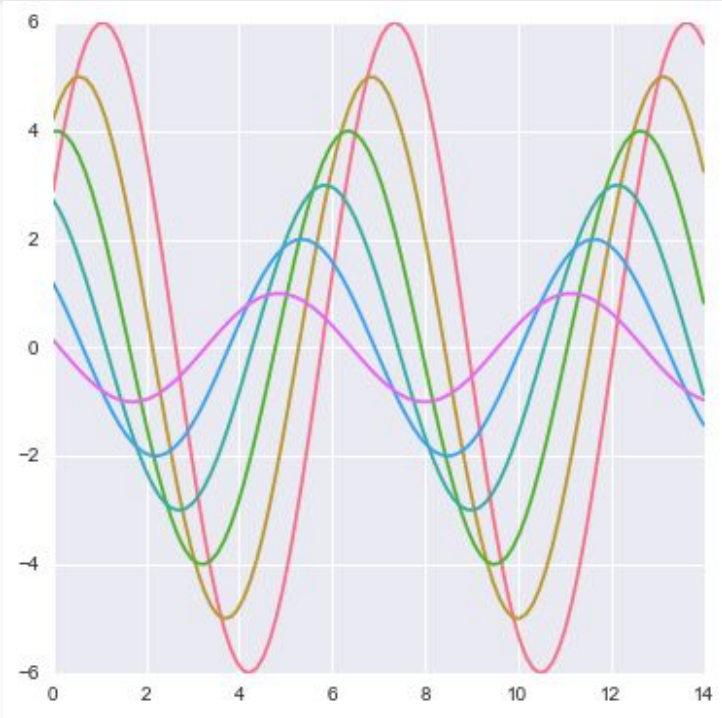
Useful when very low & very high values are interesting

sns.color_palette(“RdBu_r”, 7)

Color

Using palettes in Seaborn

sns.set_palette("husl")



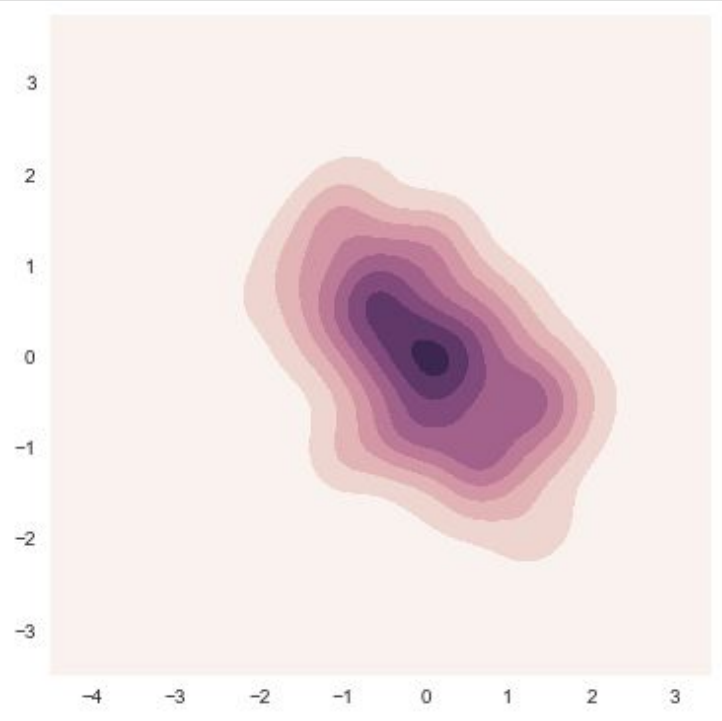
Color

Using palettes in Seaborn

as colormap object

```
x, y = np.random.multivariate_normal([0, 0],
[[1, -.5], [-.5, 1]], size=300).T


cmap = sns.cubehelix_palette(light=1,
as_cmap=True)
sns.kdeplot(x, y, cmap=cmap, shade=True);
```



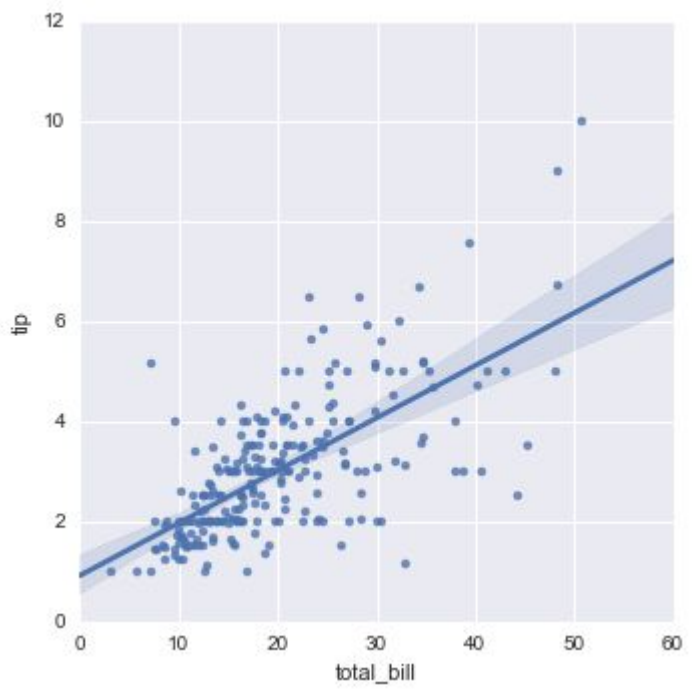
Data & Statistics Visualization

Line fitting

2 functions provided - regplot & lmplot

By default both plot a scatter plot, linear fit, & 95% confidence interval

```
sns.lmplot(x="total_bill", y="tip", data=tips)
```



Data & Statistics Visualization

Line Fitting

Differences between regplot & lmplot

- regplot
  - More input flexibility than lmplot, but fewer features
- lmplot
  - Requires “tidy data” http://vita.had.co.nz/papers/tidy-data.pdf

Data & Statistics Visualization

Line fitting

Both regplot & lmplot can

- Polynomial, logistic & LOWESS fit
- Check linear regression residuals with residplot()
- Tune appearance of scatter plot with binning, jitter

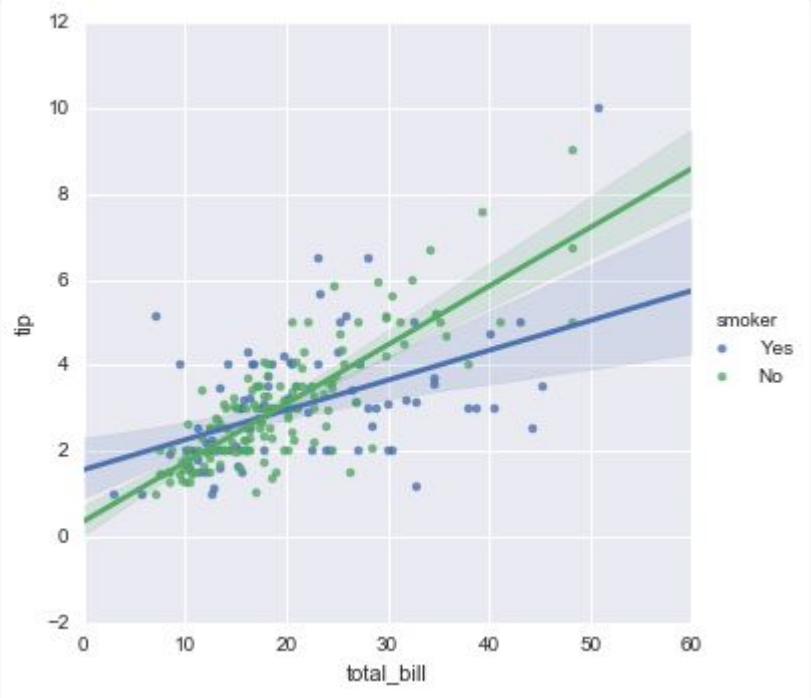
Data & Statistics Visualization

Line Fitting

When to use lmplot

Exploring how relationships change as function of a 3rd variable

```
sns.lmplot(x="total_bill", y="tip", hue="smoker", data=tips)
```



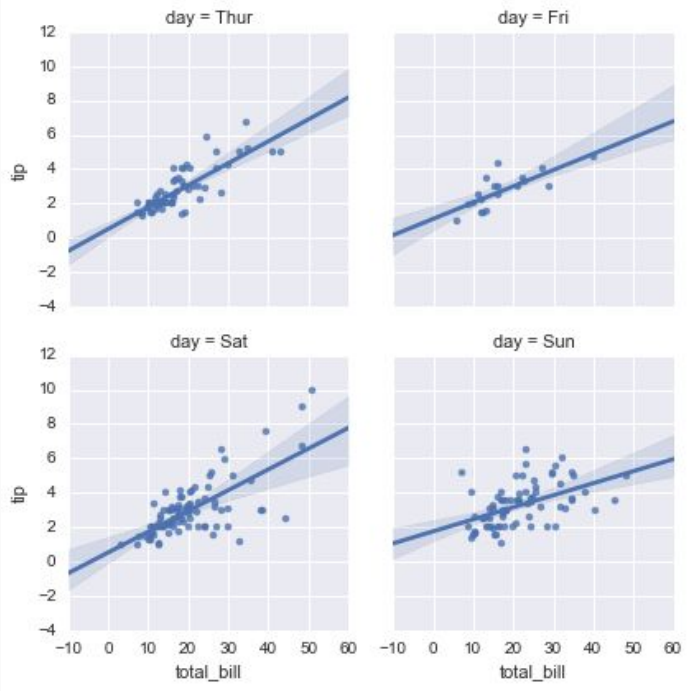
Data & Statistics Visualization

Line Fitting

When to use lmplot

Exploring how relationships change as function of a 3rd variable

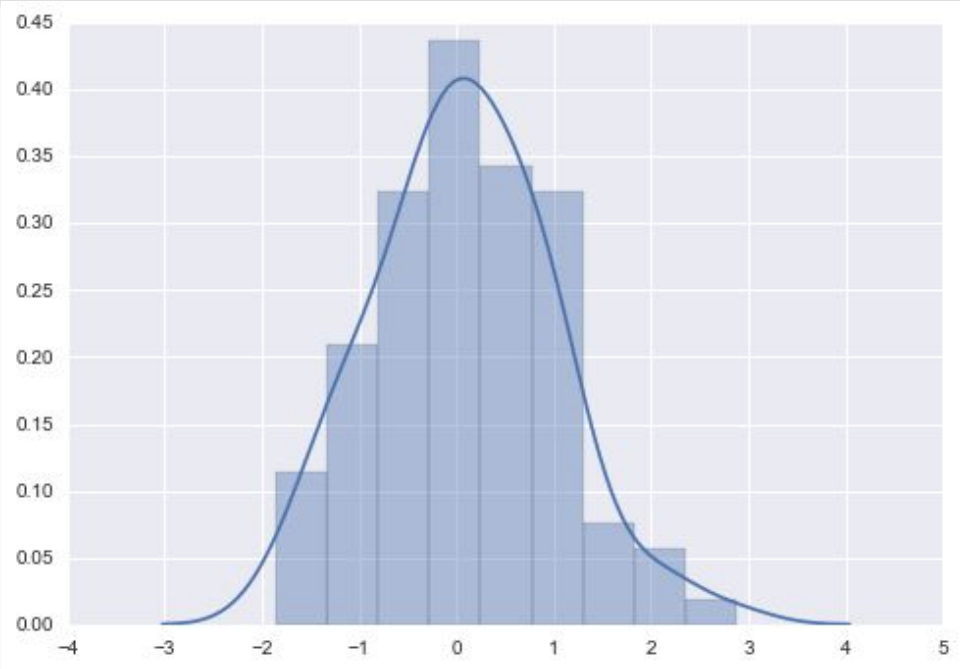
```
sns.lmplot(x="total_bill", y="tip", col="day", data=tips,
           col_wrap=2, size=3);
```



Data & Statistics Visualization

Univariate distributions

sns.distplot(x)

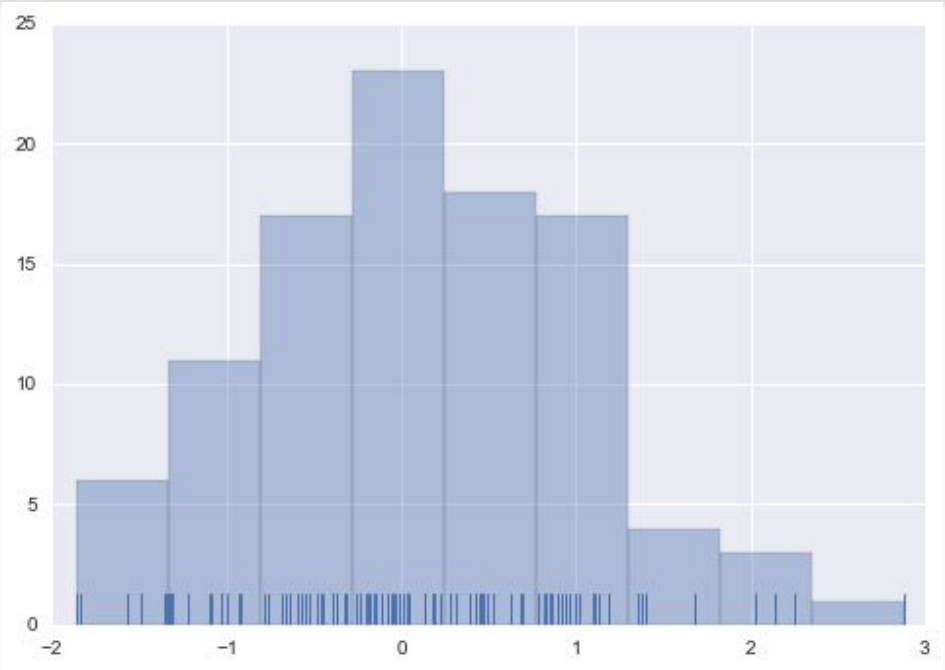


Data & Statistics Visualization

Univariate distributions

Rug plot - adds tick for each observation

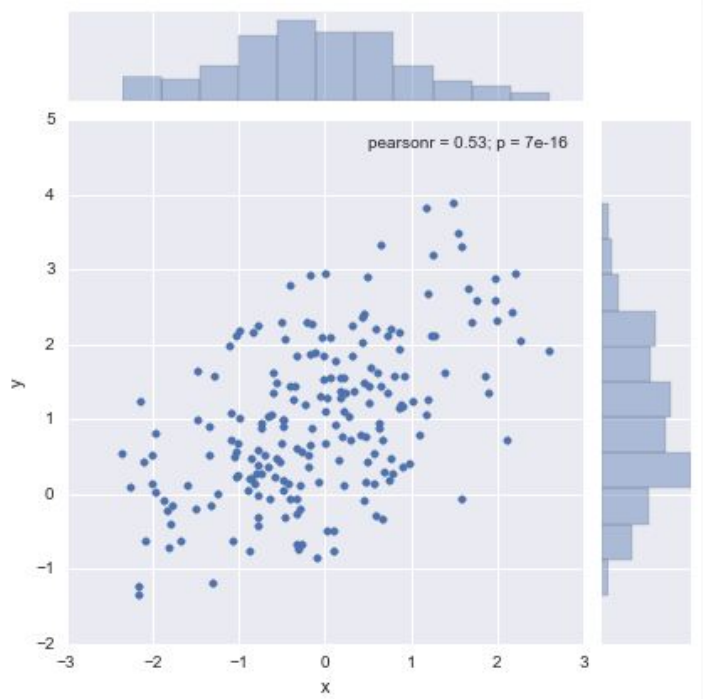
sns.distplot(x, kde=False, rug=True)



Data & Statistics Visualization

Bivariate distributions

sns.jointplot(x="x", y="y", data=df)

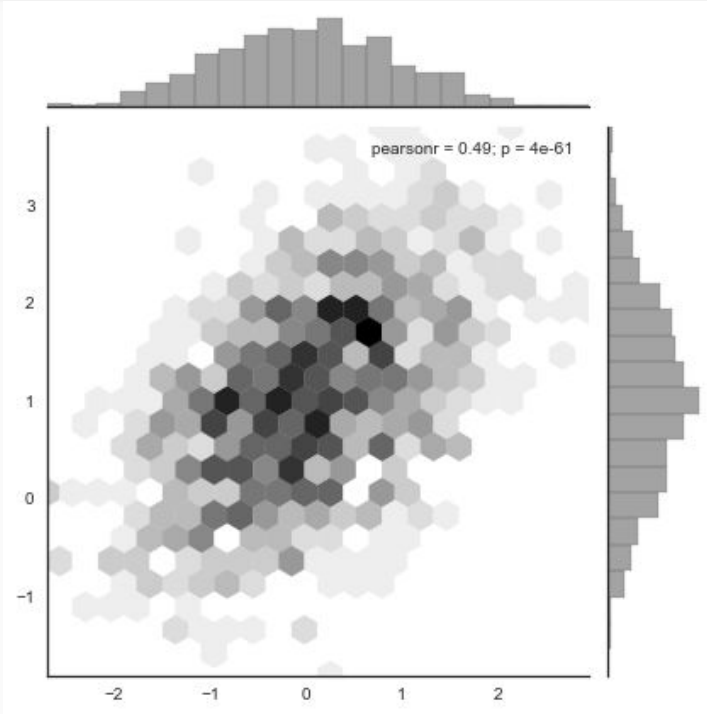


Data & Statistics Visualization

Bivariate Distributions

Hexbin - for larger data sets

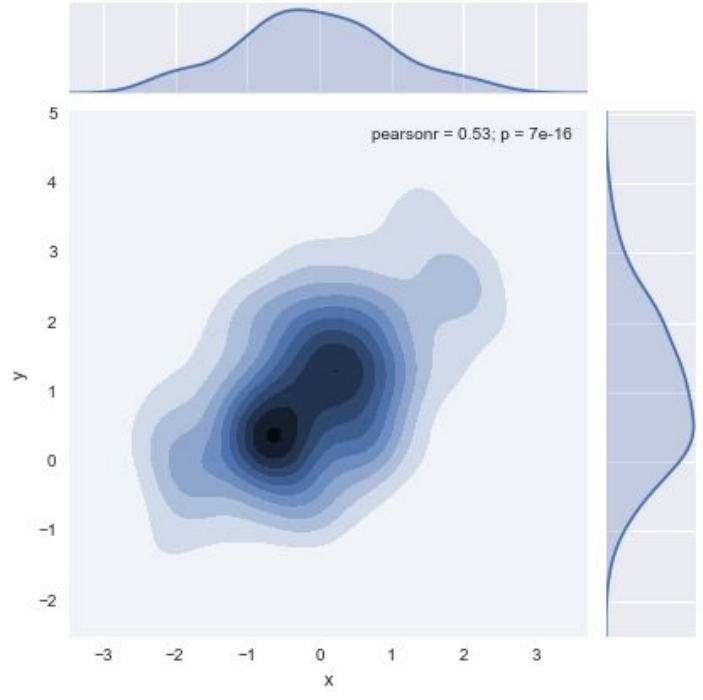
sns.jointplot(x=x, y=y, kind="hex", color="k")



Data & Statistics Visualization

Contour plot with KDE

sns.jointplot(x="x", y="y", data=df, kind="kde")



Data & Statistics Visualization

Heatmap

```
flights_long = sns.load_dataset("flights")
flights = flights_long.pivot("month", "year",
"passengers")

sns.heatmap(flights, annot=True, fmt="d",
linewidths=.5)
```

Data & Statistics Visualization

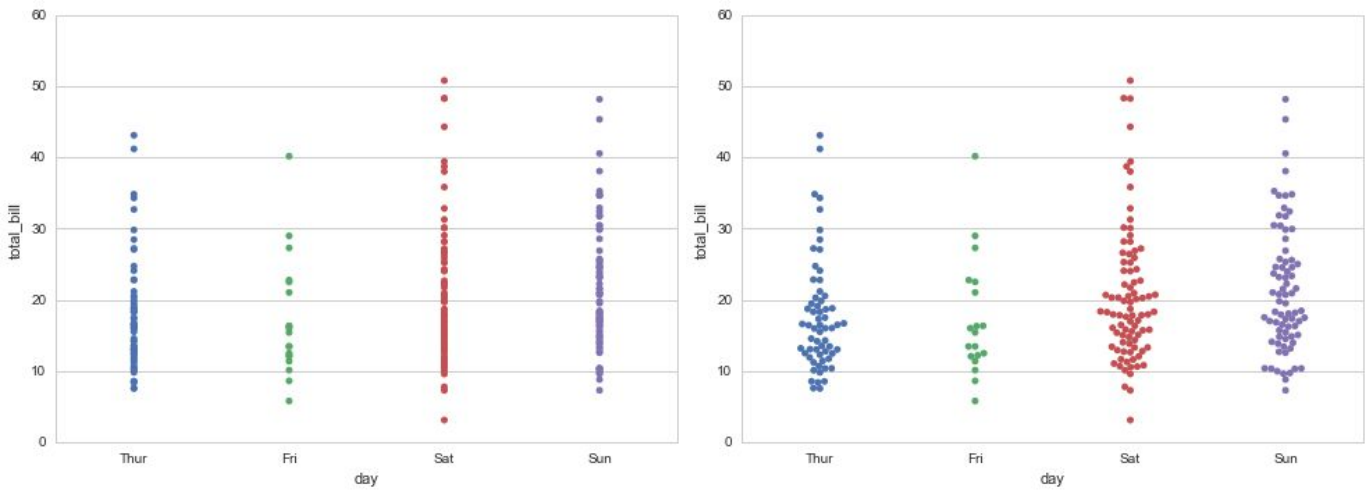
Categorical Data

Works best with “tidy format” DataFrames, low-level functions can handle “untidy” data & vectors

Data & Statistics Visualization

Categorical data

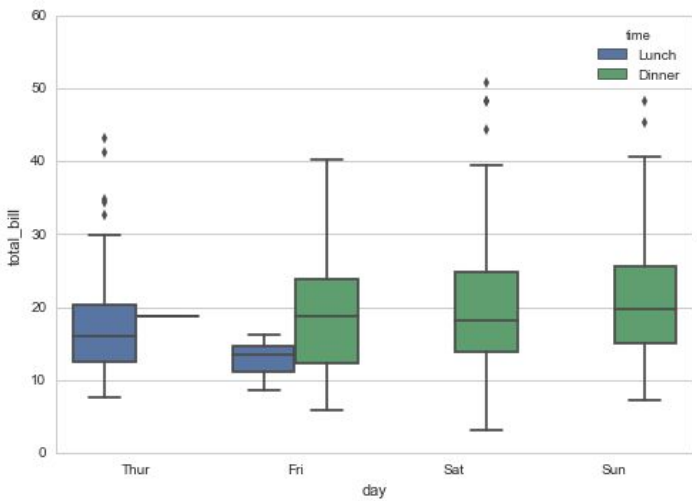
Strip/Swarm plots

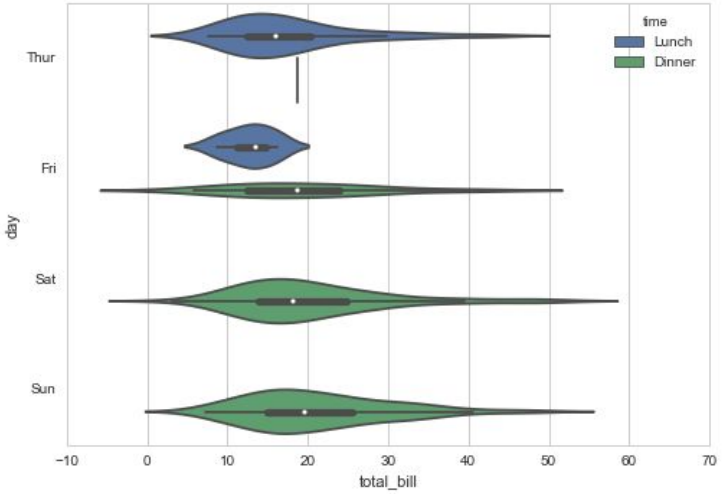


Data & Statistics Visualization

Categorical Data

Box & Violin plots



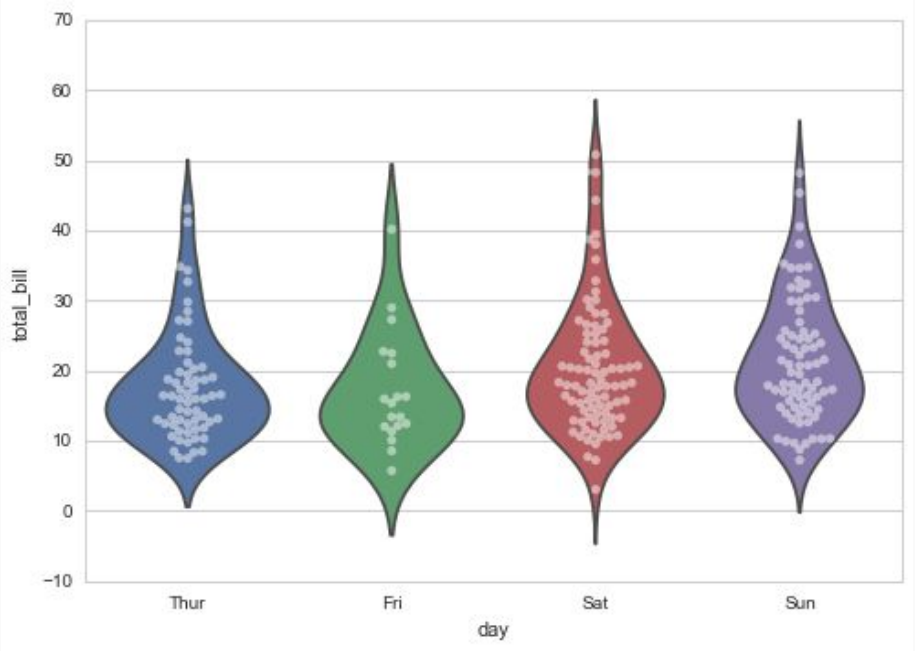


Data & Statistics Visualization

Categorical Data

Plots can be tuned & combined

```
sns.violinplot(x="day", y="total_bill", data=tips,
inner=None)
sns.swarmplot(x="day", y="total_bill", data=tips,
color="w", alpha=.5)
```



Resources

- Main web site: https://stanford.edu/~mwaskom/software/seaborn/
- Tutorial: https://stanford.edu/~mwaskom/software/seaborn/tutorial.html
- API reference: https://stanford.edu/~mwaskom/software/seaborn/api.html
- Tidy data: http://vita.had.co.nz/papers/tidy-data.pdf