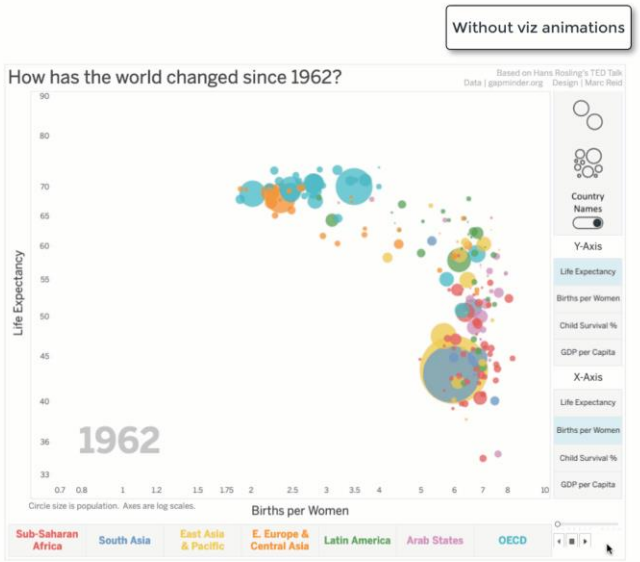
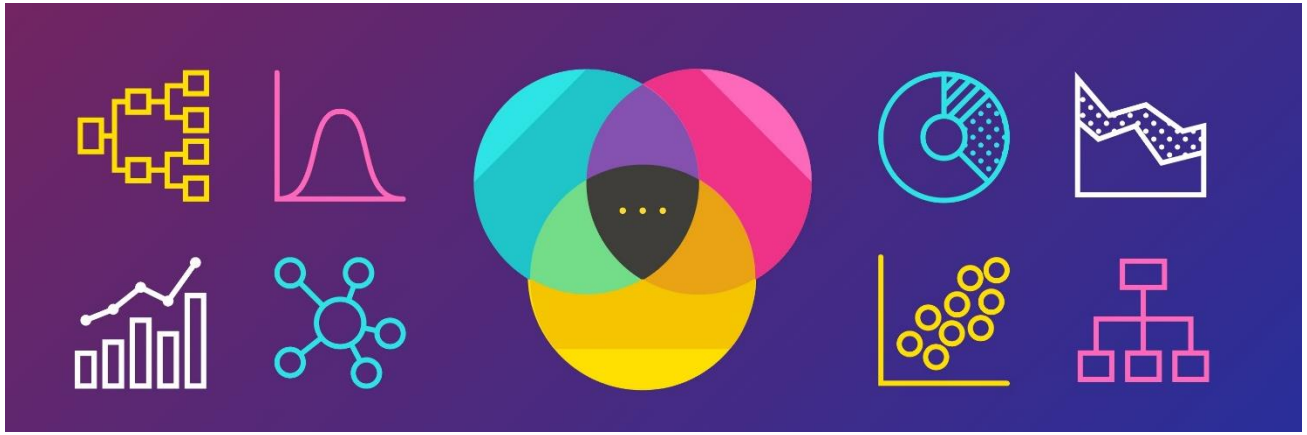
# Python-Data Visualization

Dr. Sarwan Singh

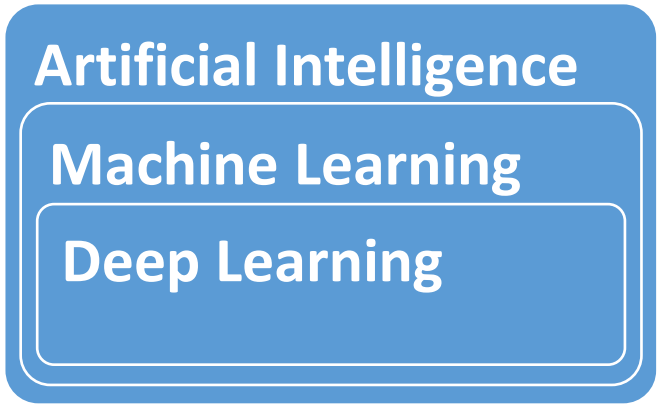



Data Visualizations with Tableau Public

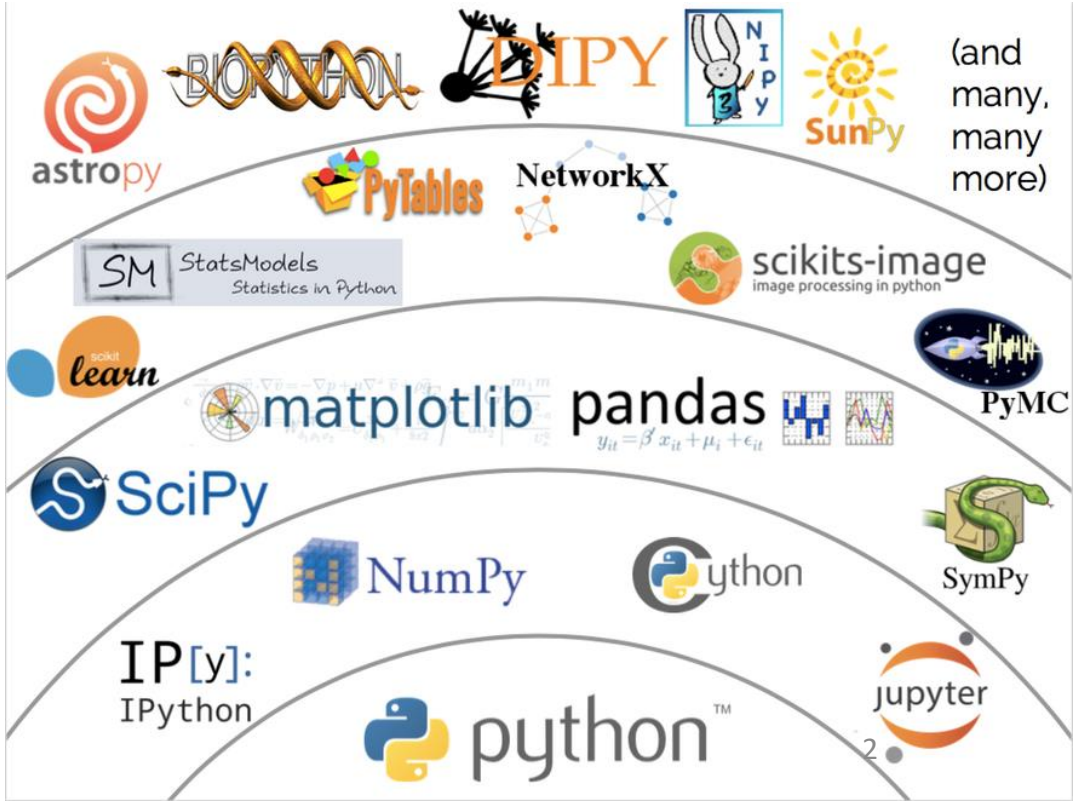


# Agenda

- Seaborn – Introduction




- References - seaborn.pydata.org, tutorialpoint.com, geeksforgeeks.com, kaggle.com

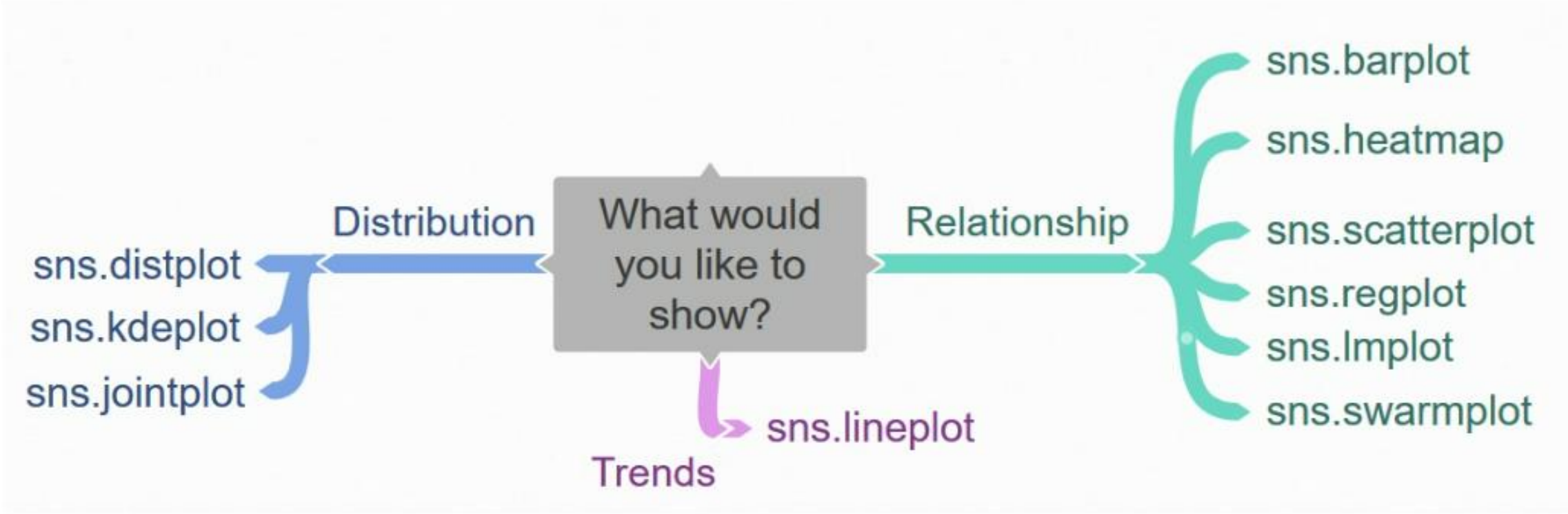


# Seaborn

*"a powerful but easy-to-use data visualization tool"*

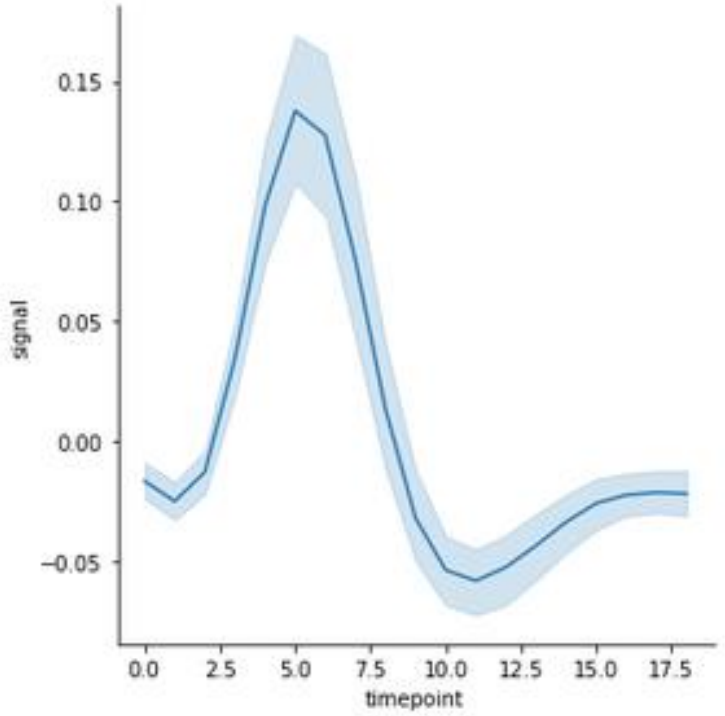
- Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.
- Seaborn helps you explore and understand your data.
- Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.
- Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

# Data Visualization with Seaborn

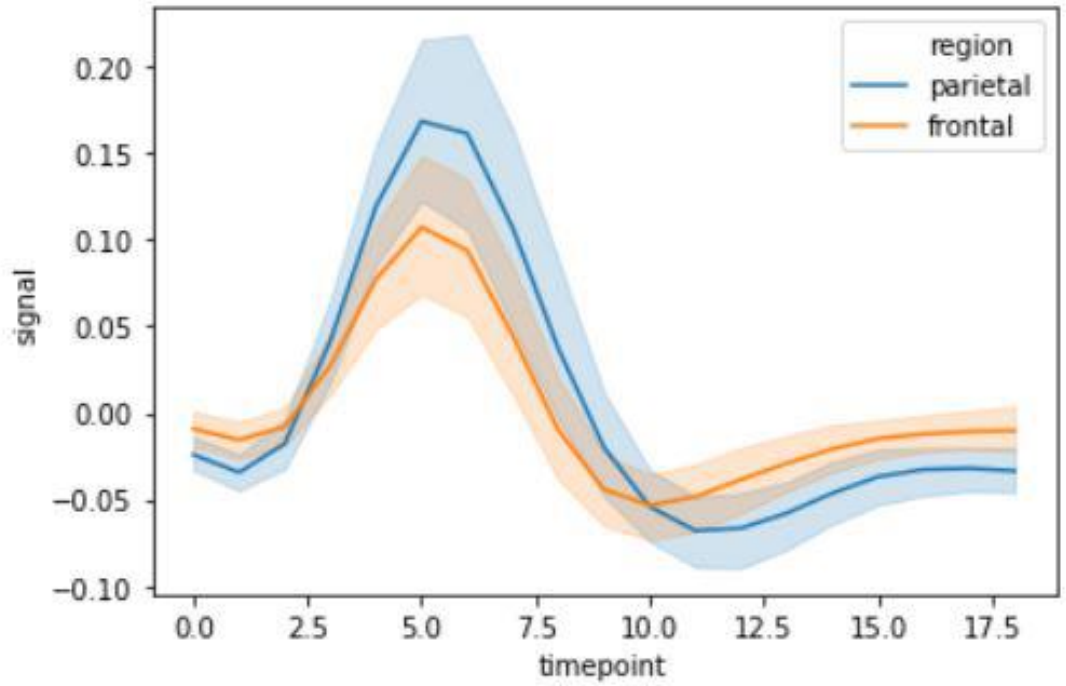


# Trends

- A trend is defined as a pattern of change.
- sns.lineplot - **Line charts** are best to show trends over a period of time, and multiple lines can be used to show trends in more than one group.




```
sns.lineplot( x = "timepoint",
              y = "signal", data = fmri);
```




```
sns.lineplot( x = "timepoint",  y = "signal",
              hue = "region",data = fmri );
```

# Lineplot - Syntax

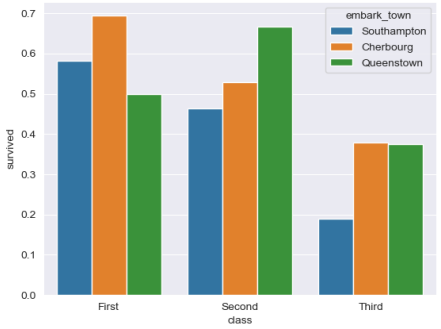
- seaborn.lineplot(x=None, y=None, hue=None, size=None, style=None, data=None, palette=None, hue_order=None, hue_norm=None, sizes=None, size_order=None, size_norm=None, dashes=True, markers=None, style_order=None, units=None, estimator='mean', ci=95, n_boot=1000, seed=None, sort=True, err_style='band', err_kws=None, legend='brief', ax=None, **kwargs)

**Parameters:**

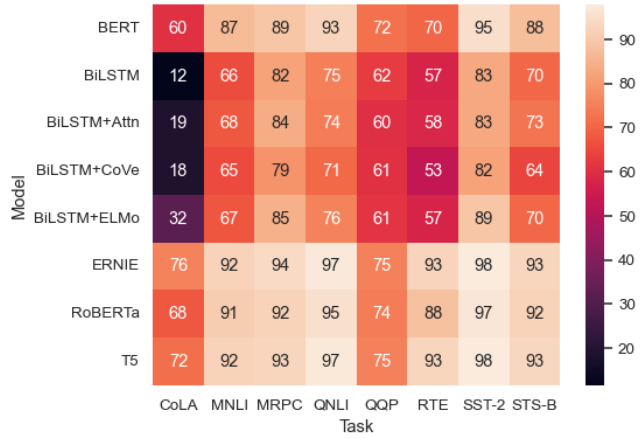
- **x, y:** Input data variables; must be numeric.
- **data:** Dataframe where each column is a variable and each row is an observation.
- **size:** Grouping variable that will produce lines with different widths.
- **style:** Grouping variable that will produce lines with different dashes and/or markers.

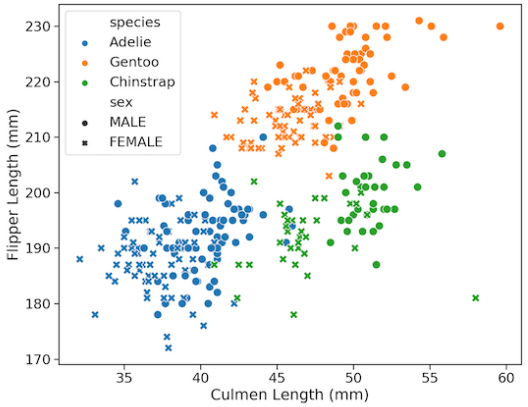
# Relationship

*"use to understand relationships between variables in your data"*




- sns.barplot - **Bar charts** are useful for comparing quantities corresponding to different groups.
- sns.heatmap - **Heatmaps** can be used to find color-coded patterns in tables of numbers.
- sns.scatterplot - **Scatter plots** show the relationship between two continuous variables; if color-coded, we can also show the relationship with a third categorical variable.

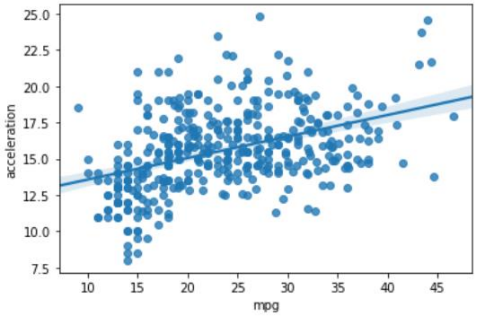


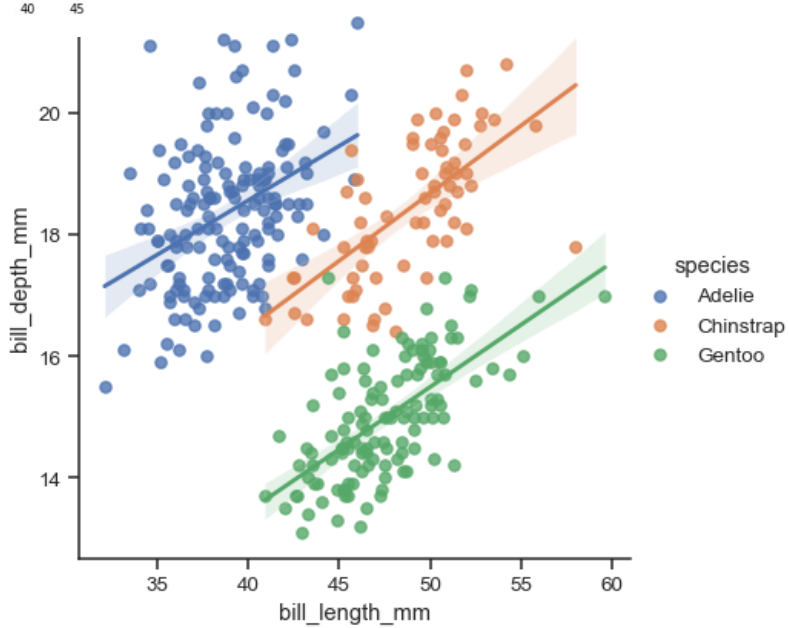



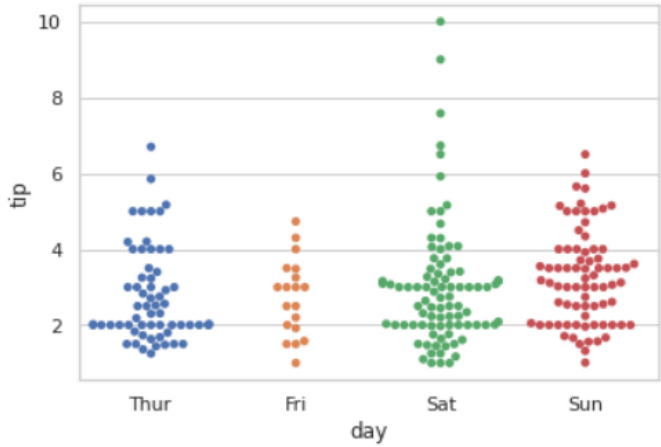
# Relationship

*"use to understand relationships between variables in your data"*

- sns.regplot - Including a **regression line** in the scatter plot makes it easier to see any linear relationship between two variables.
- sns.lmplot - This command is useful for drawing **multiple regression lines**, if the scatter plot contains multiple, color-coded groups.
- sns.swarmplot - **Categorical scatter plots** show the relationship between a continuous variable and a categorical variable.



# Distribution

*"visualize distributions to show the possible values that we can expect to see in a variable, along with how likely they are"*

- sns.distplot - **Histograms** show the distribution of a single numerical variable.
- sns.kdeplot - **KDE plots** (or **2D KDE plots**) show an estimated, smooth distribution of a single numerical variable (or two numerical variables).
- sns.jointplot - This command is useful for simultaneously displaying a 2D KDE plot with the corresponding KDE plots for each individual variable.

# Categorical Graphs

**Categorical scatterplots:**

- sns.stripplot() (with kind="strip"; the default)
- sns.swarmplot() (with kind="swarm")

**Categorical distribution plots:**

- sns.boxplot() (with kind="box")
- sns.violinplot() (with kind="violin")
- sns.boxenplot() (with kind="boxen")

**Categorical estimate plots:**

- sns.pointplot() (with kind="point")
- sns.barplot() (with kind="bar")
- sns.countplot() (with kind="count")