Seaborn

Seaborn

- Seaborn is a statistical plotting library that is specifically designed to interact well with Pandas DataFrames to create common statistical plot types.
- Seaborn is built directly off of Matplotlib but uses a simpler “one-line” syntax.

Seaborn

- When using seaborn, we trade-off customization for ease of use.
- However, since its built directly off of Matplotlib, we can actually still make plt method calls to directly affect the resulting seaborn plot.

Seaborn

- A typical seaborn plot uses one line of code, for example:
    - **sns.scatterplot(x='salary',y='sales',data=df)**
- Seaborn takes in a pandas DataFrame and then the user provides the corresponding string column names for x and y (depending on the plot type)

Seaborn

* In this section we focus on understanding the use cases for each plot and the seaborn syntax for them.
* Online Docs: https://seaborn.pydata.org/
* Common student question:
  * *"How do I choose which plot to use?"*

Seaborn

- It depends on what questions or relationships you are trying to understand.
- Google Image Searching “Choosing a plot visualization” will yield many useful flowcharts.
- At the end of this section, you will have a good intuition of which plots to use.

Seaborn

- Section Topics:
    - Scatter Plots
    - Distribution Plots
    - Categorical Plots
    - Comparison Plots
    - Seaborn Grids
    - Matrix Plots

Let’s get started!

Scatter Plots

Seaborn

- Scatter plots show the relationship between two continuous features.
- Recall that **continuous** features are numeric variables that can take any number of values between any two values.

Seaborn

* Continuous Feature Examples
    * Age
    * Height
    * Salary
    * Temperature
    * Prices

Seaborn

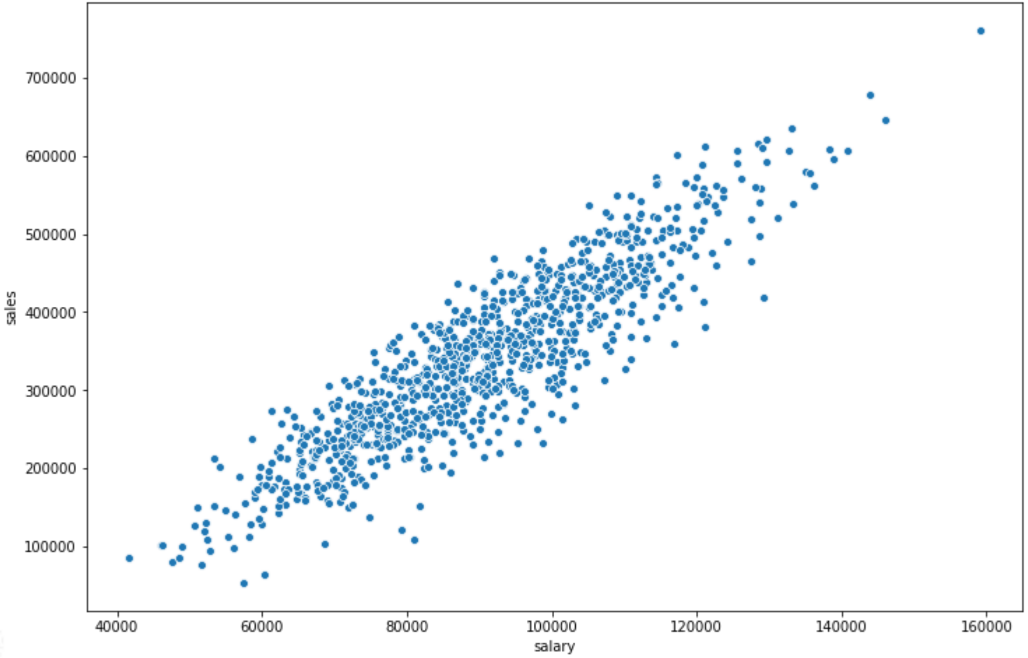
- A **continuous** feature allows for a value to always be between two values.
- Not to be confused with **categorical** features which represent distinct and unique categories:
  - Colors
  - Shapes
  - Names

Seaborn

- Scatter plots line up a set of two continuous features and plots them out as coordinates.
- For example, imagine employees with salaries who sell a certain dollar amount of items each year. We could explore the relationship between employee salaries and sales amount.

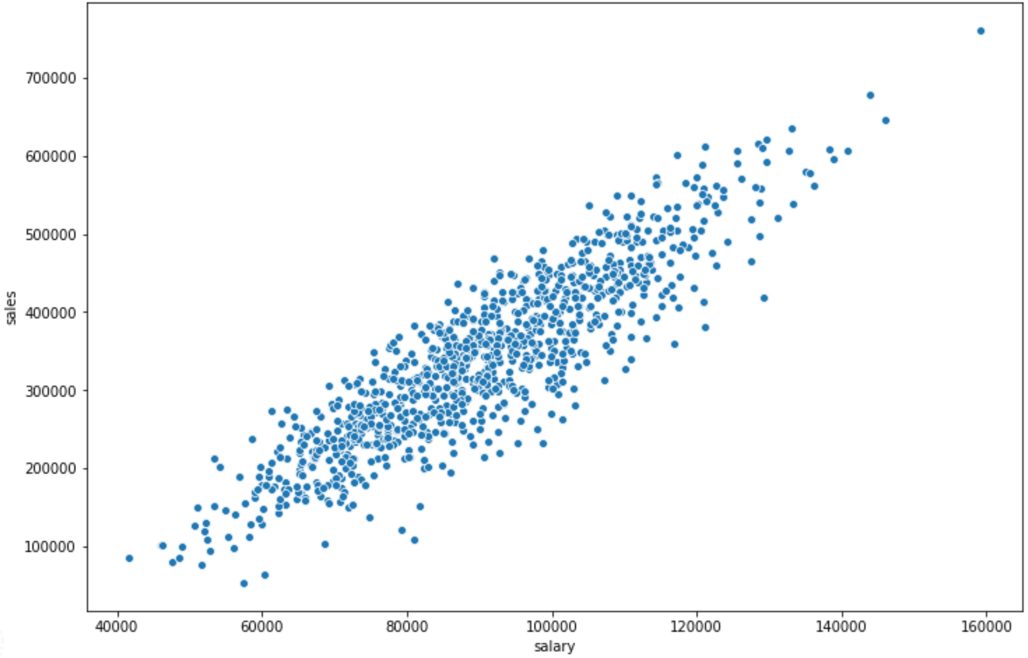
Seaborn

- Plot (x,y) coordinate points



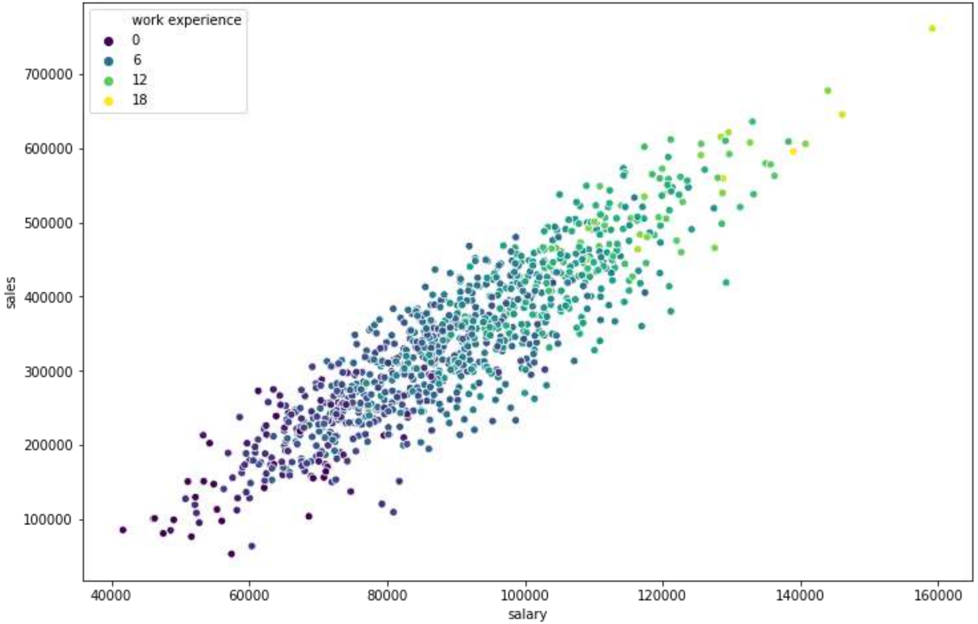
Seaborn

- Plot (salary,sales) coordinate points



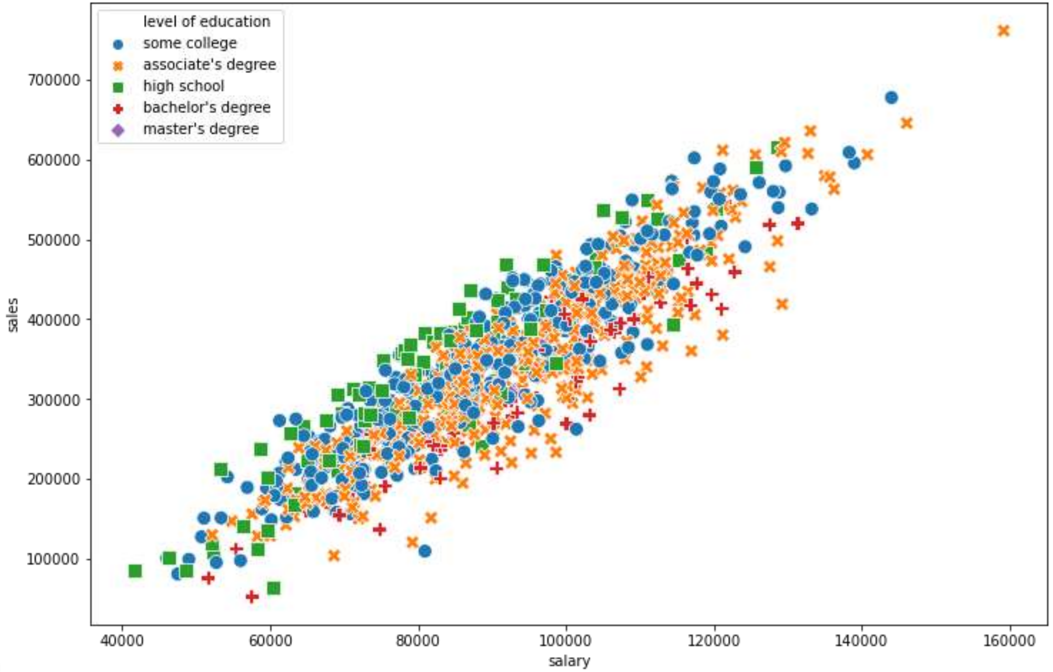
Seaborn

- Seaborn can then add coloring and styling



Seaborn

- Seaborn can then add coloring and styling



PIERIAN DATA

Seaborn

- Let’s explore Scatter Plots with seaborn!

Distribution Plots

PART ONE: UNDERSTANDING PLOT TYPES

Seaborn

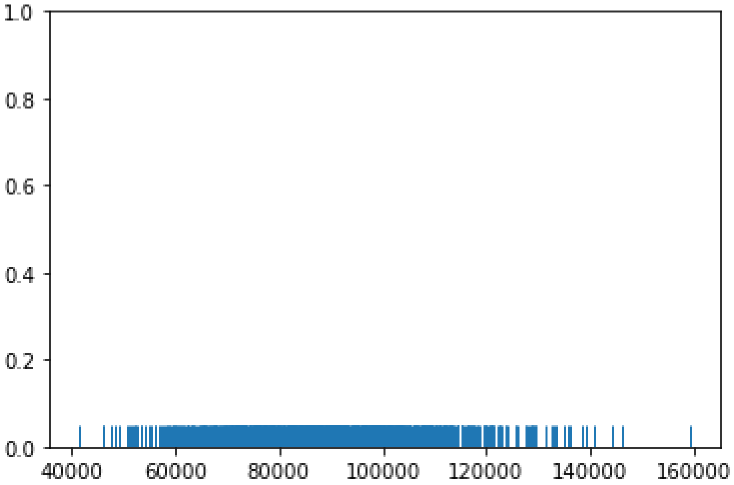
- Distribution plots display a single continuous feature and help visualize properties such as deviation and average values.
- There are 3 main distribution plot types:
    - Rug Plot
    - Histogram
    - KDE Plot

Seaborn

- Let’s explore the distribution of employee salaries.
- One way to do this is through a rug plot.
- A rug plot is the simplest distribution plot and merely adds a dash or tick line for every single value.
- The y-axis does not really have a meaning.

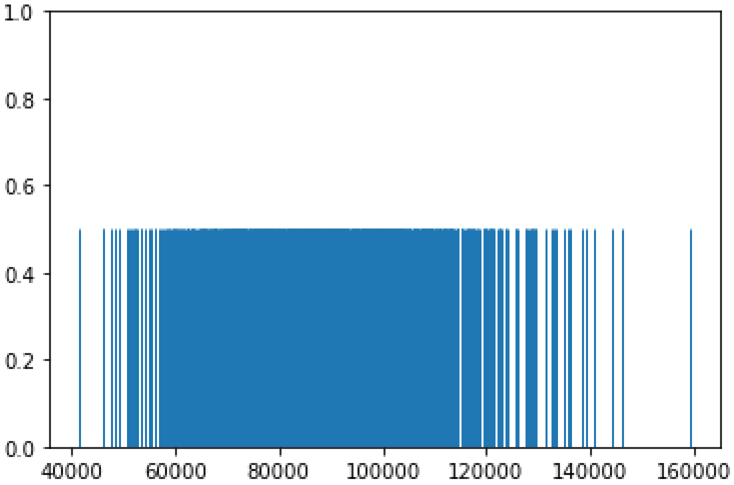
Seaborn

- Rug Plot of Salaries:
    - Adds a tick for every salary value



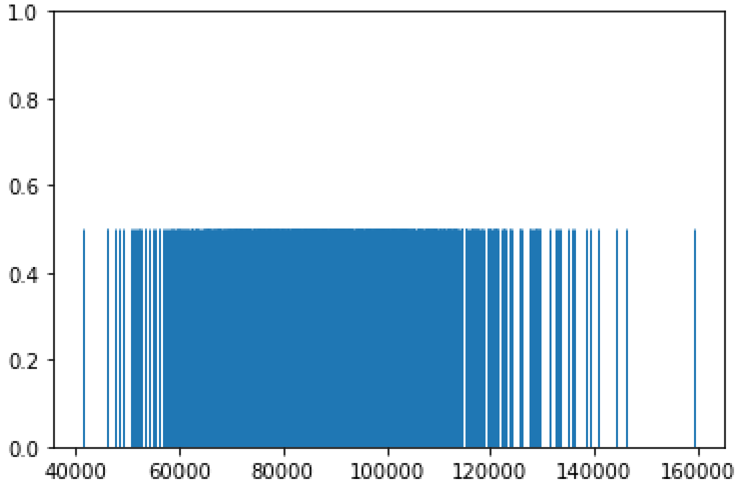
Seaborn

- Rug Plot of Salaries:
  - Optionally adjust height of ticks



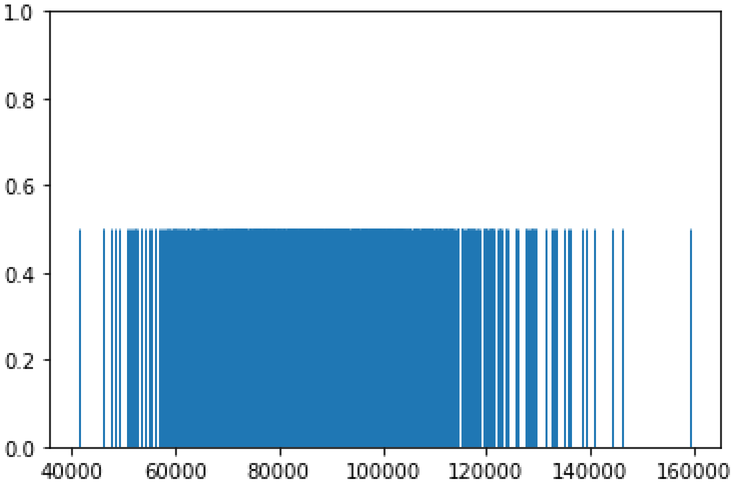
Seaborn

- Rug Plot of Salaries:
    - Y-axis not interpretable



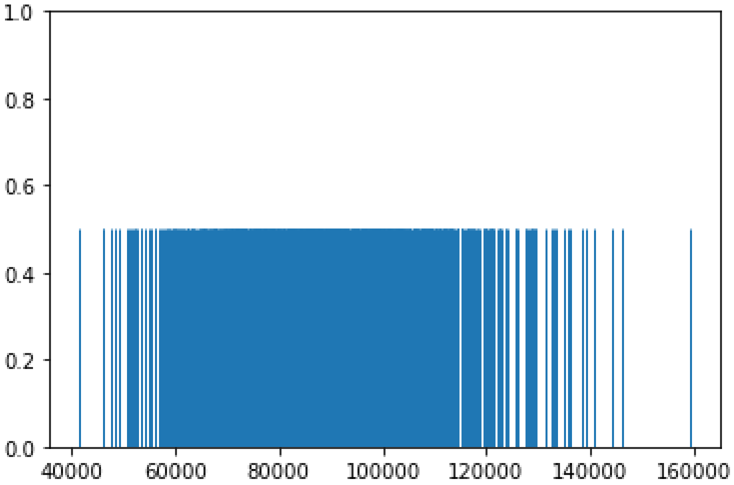
Seaborn

- Rug Plot of Salaries:
    - Highest salary near $160,000



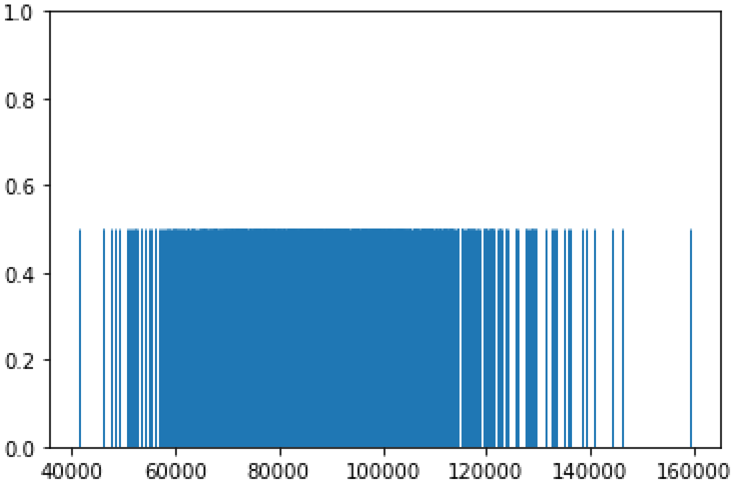
Seaborn

- Rug Plot of Salaries:
    - Many salaries between $60k - $120k



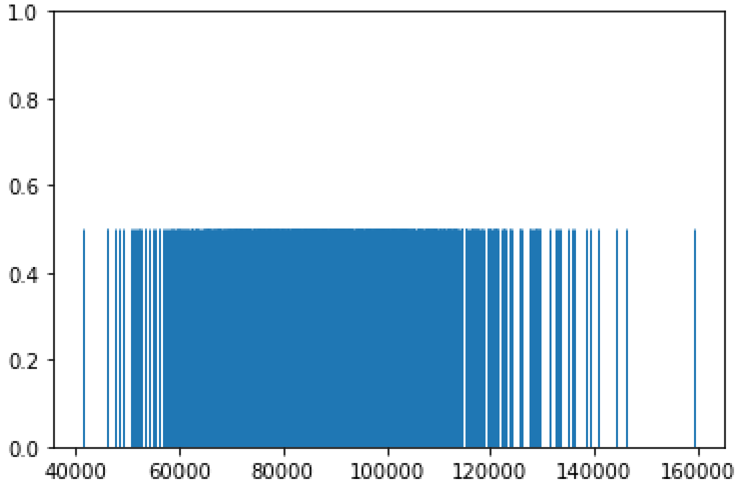
Seaborn

- Rug Plot of Salaries:
  - Many ticks could be right on top of eachother, we can't tell!



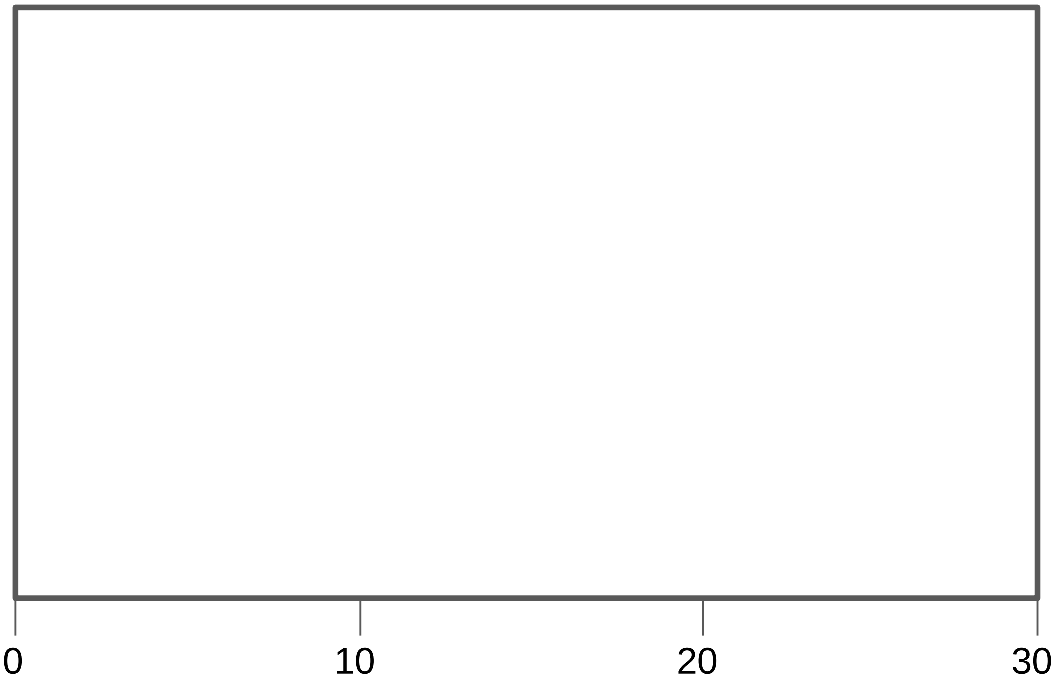
Seaborn

- If we **count** how many ticks there are per various x-ranges, we can create a **histogram**.



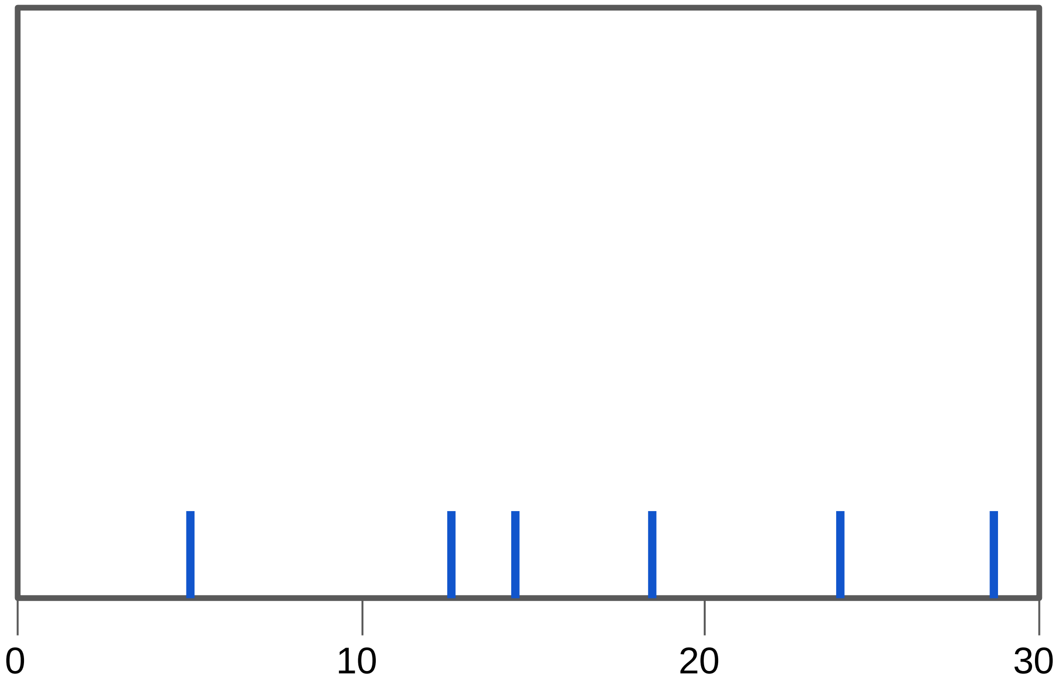
Seaborn

- Let's explore a simple example



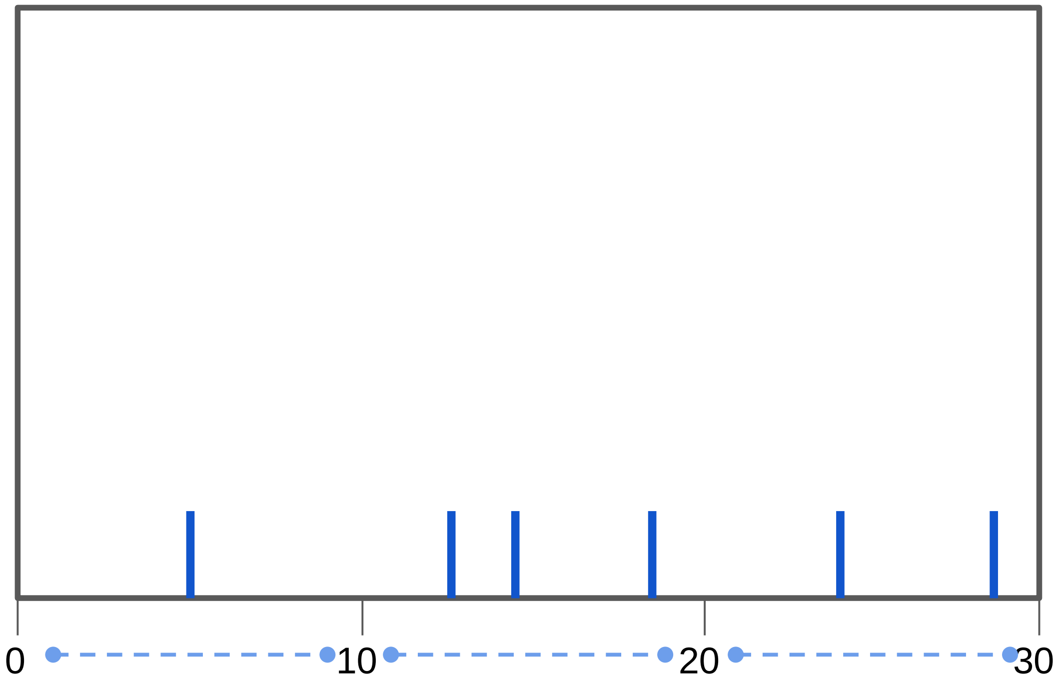
Seaborn

- We place the rug plot ticks



Seaborn

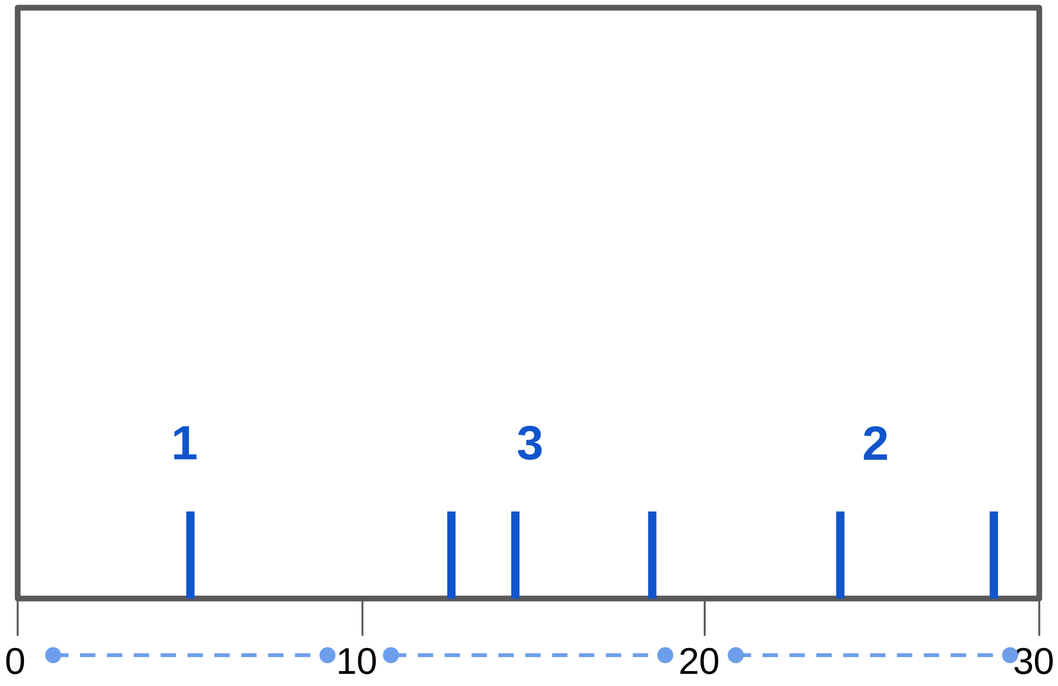
- Choose a number of “bins”, we’ll pick 3



PIERIAN DATA

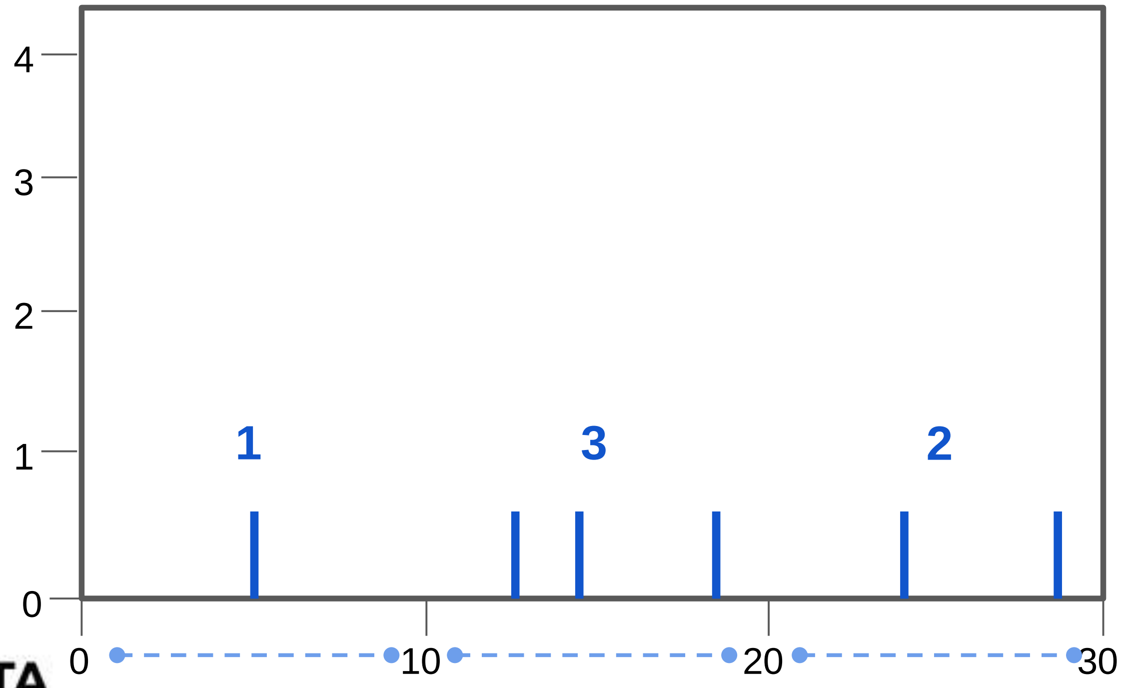
Seaborn

- Count ticks per bin



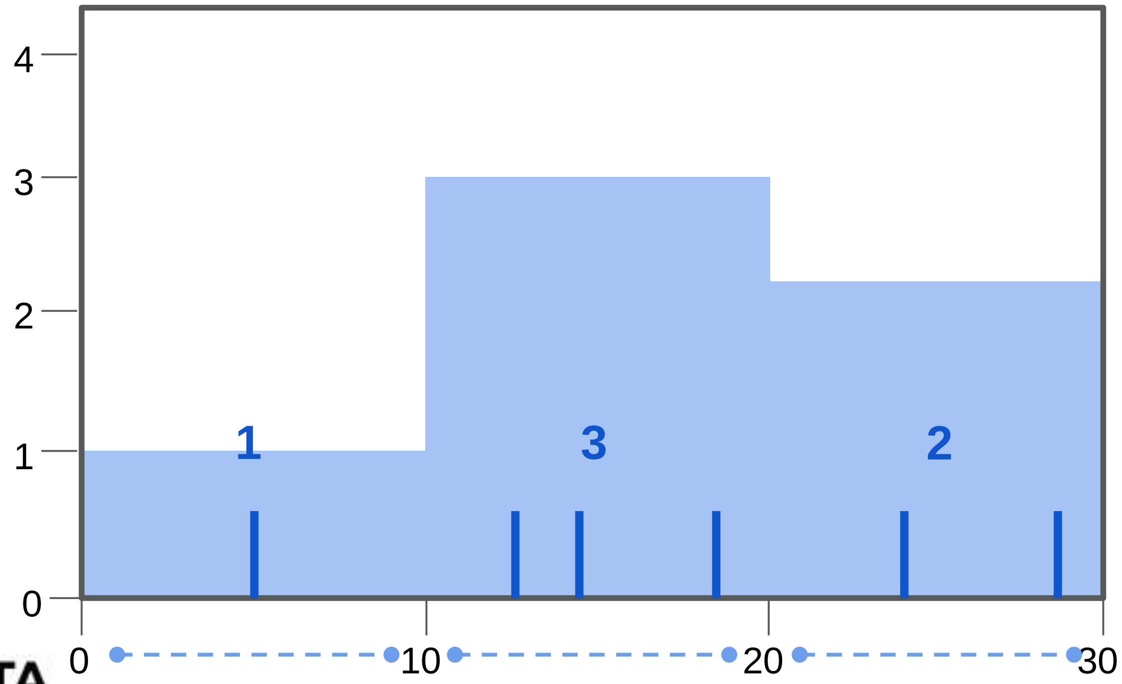
Seaborn

- Create a bar as high as count



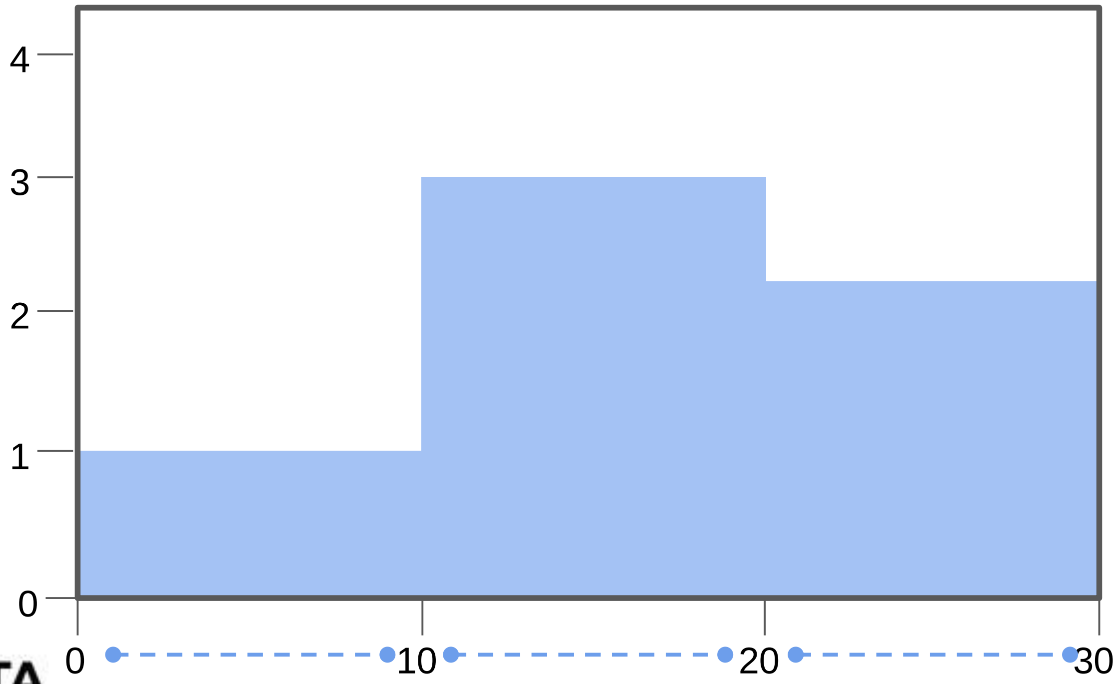
Seaborn

* Create a bar as high as count



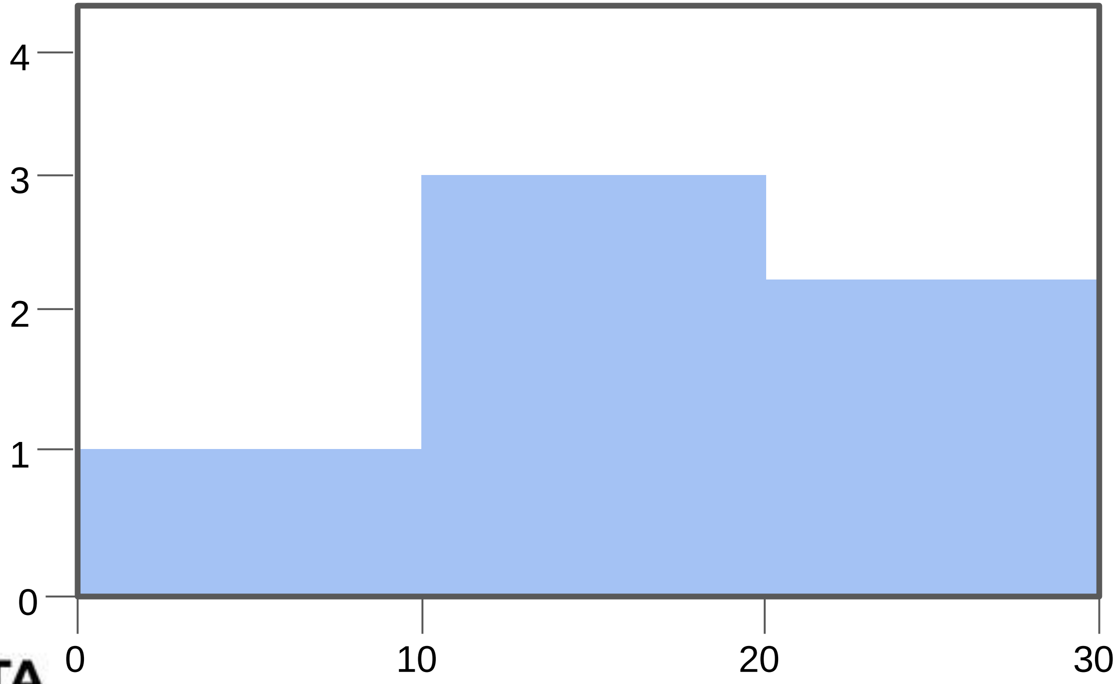
Seaborn

- Create a bar as high as count



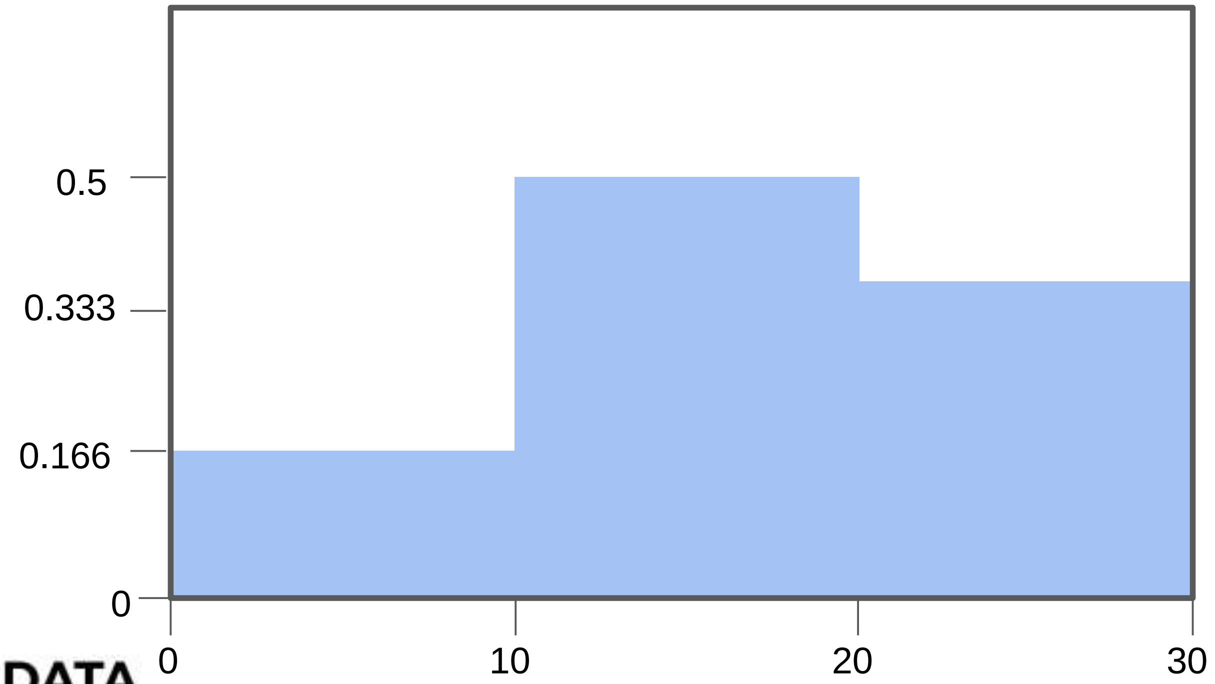
Seaborn

- Histogram is complete



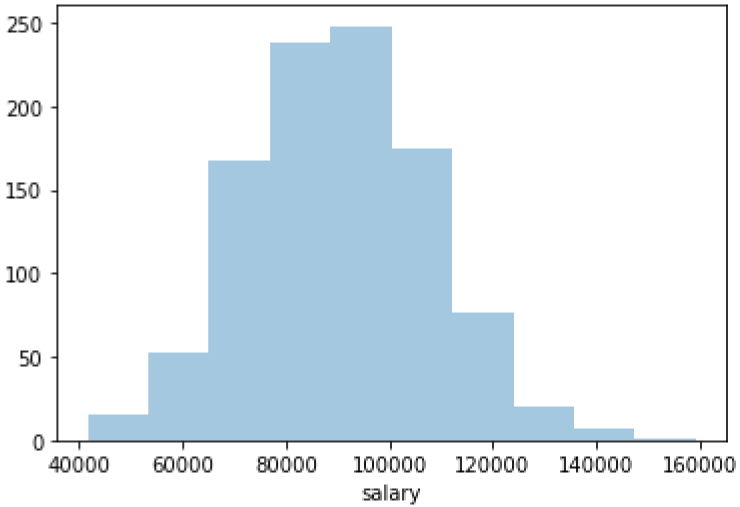
Seaborn

- Y-axis can also be normalized as percent



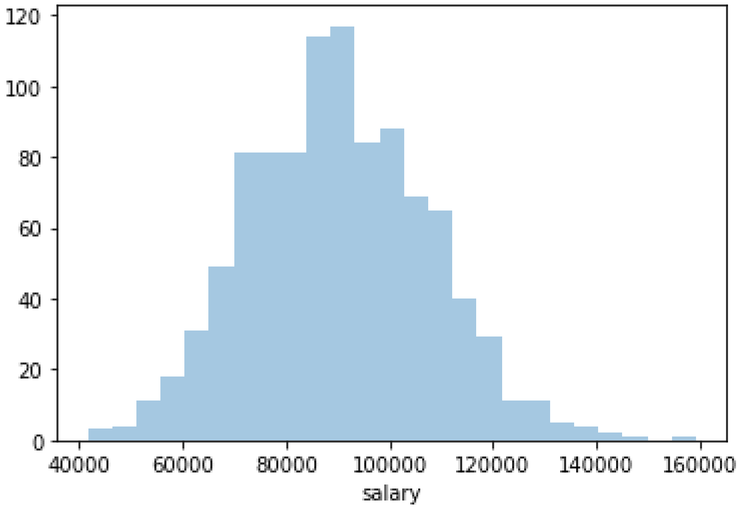
Seaborn

- Changing number of bins shows more detail instead of general trends.



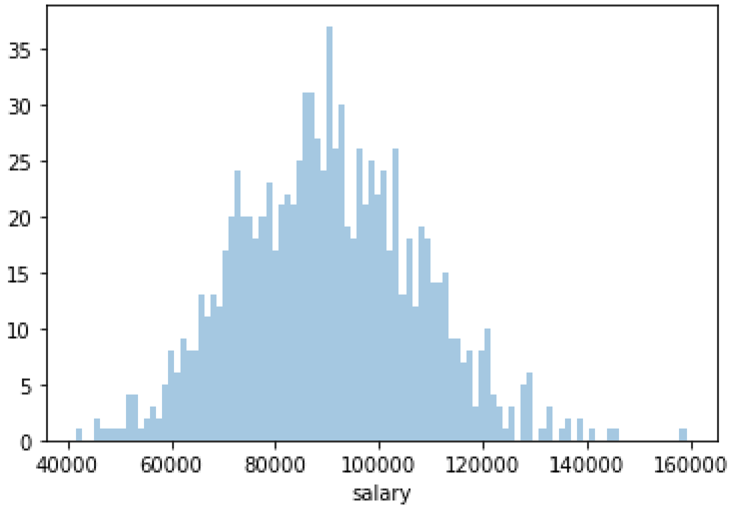
Seaborn

- Changing number of bins shows more detail instead of general trends.



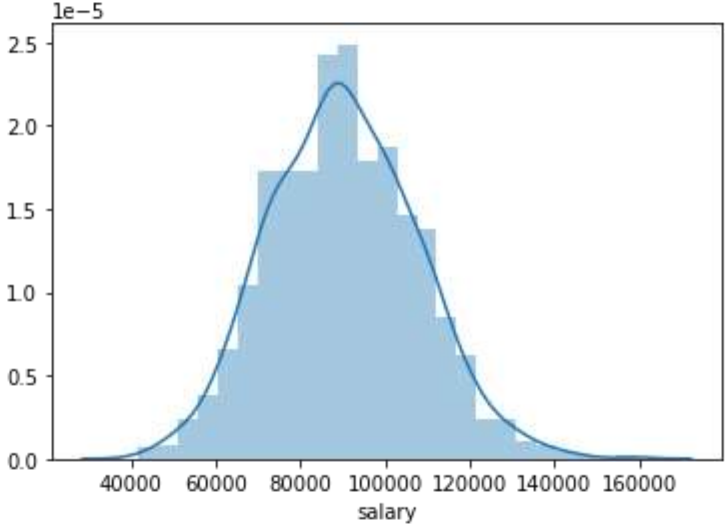
Seaborn

- Changing number of bins shows more detail instead of general trends.



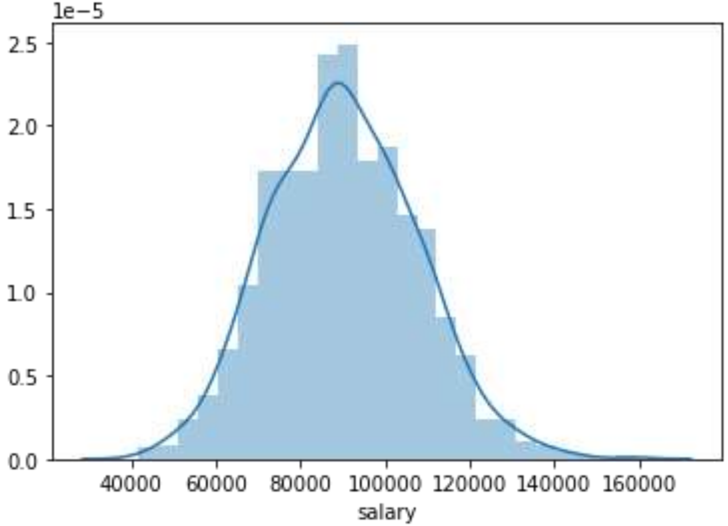
Seaborn

- Seaborn also allows us to add on a KDE plot curve on top of a histogram.



Seaborn

- Let's explore what a KDE plot is and how it is constructed.



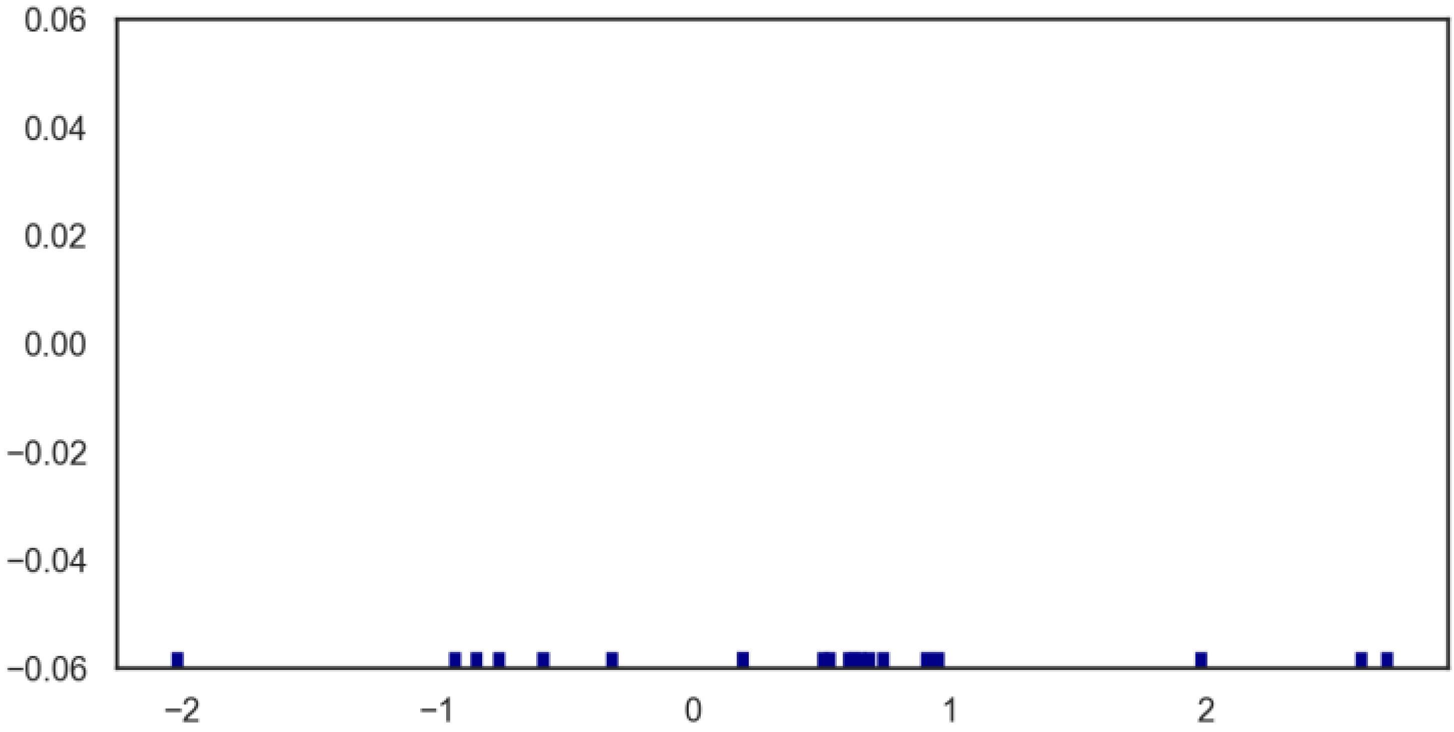
Seaborn

* KDE stands for Kernel Density Estimation.
* It is a method of **estimating** a probability density function of a random variable.
* In simpler terms, it is a way of estimating a continuous probability curve for a finite data sample.

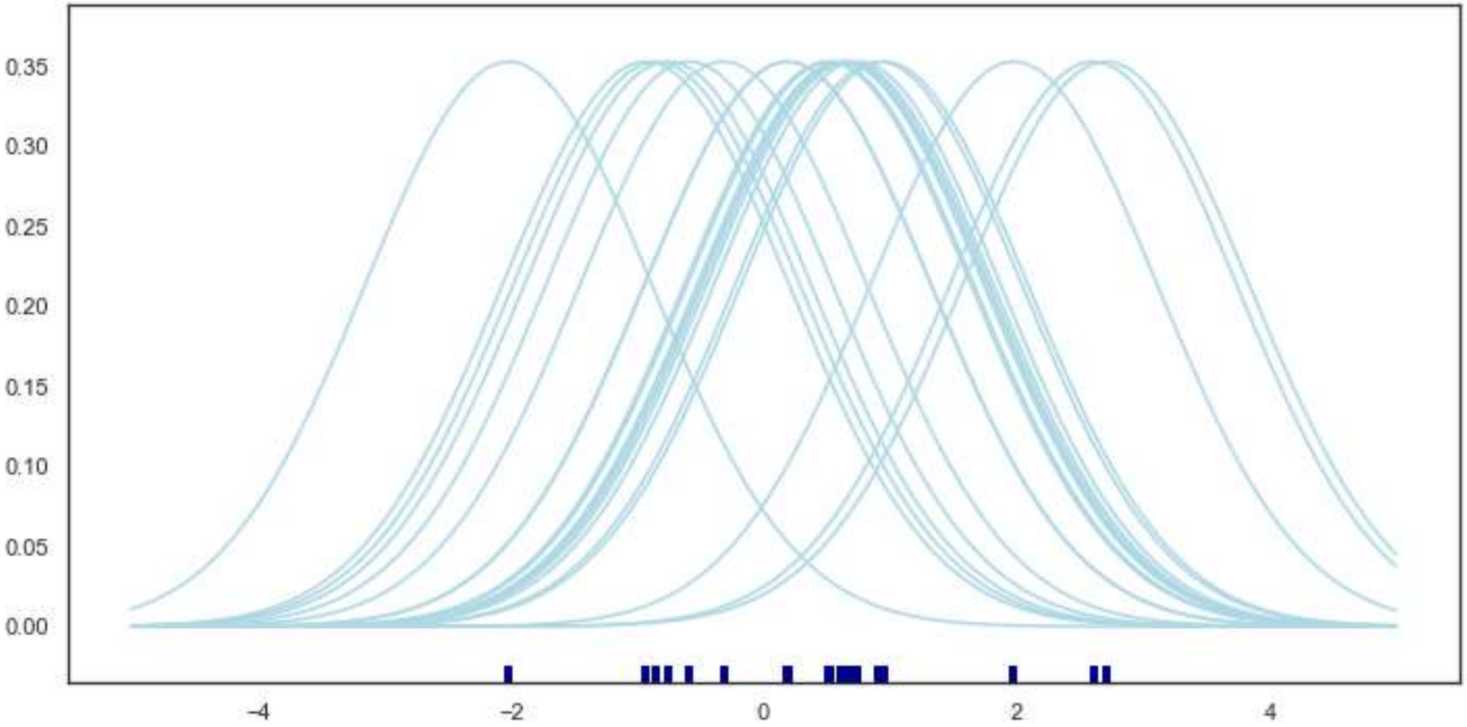
Seaborn

- KDE plots are best understood by visualizing their “construction”.
- Let’s start with a rug plot….

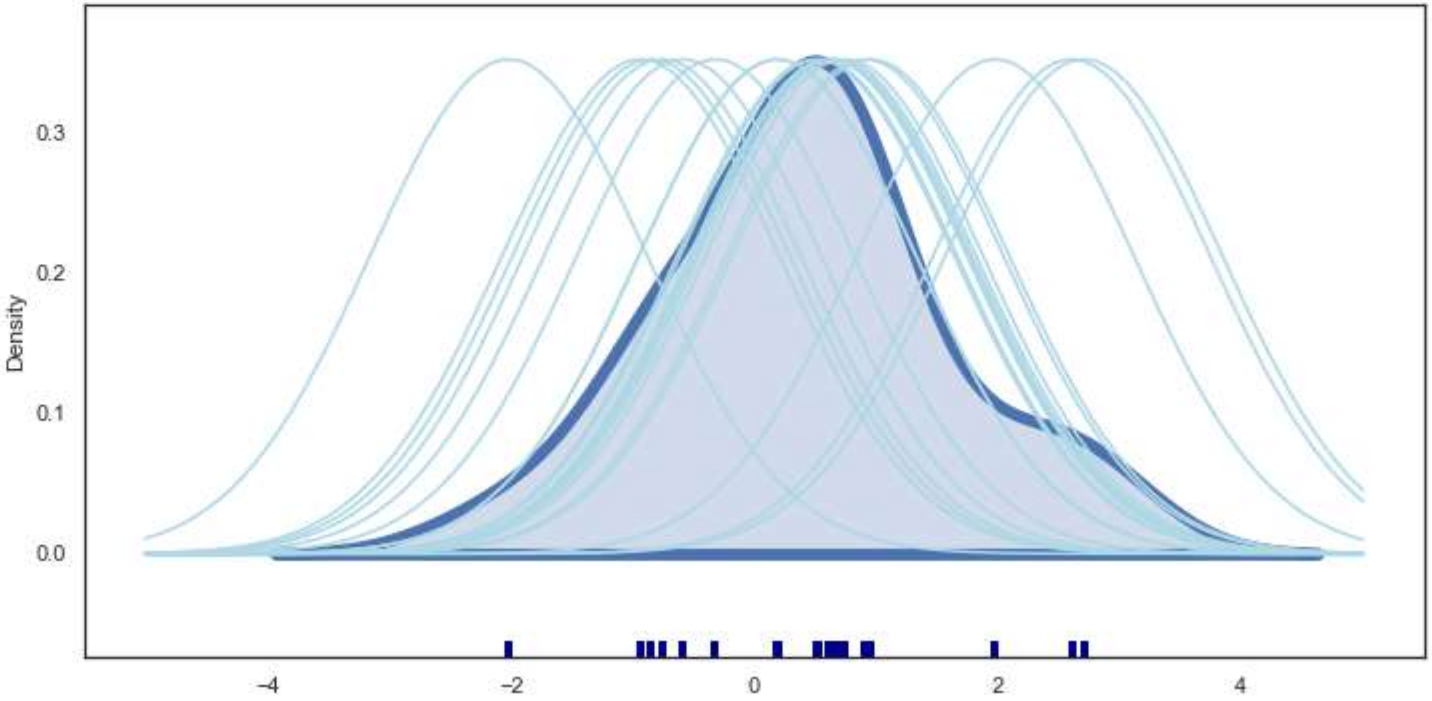
Seaborn



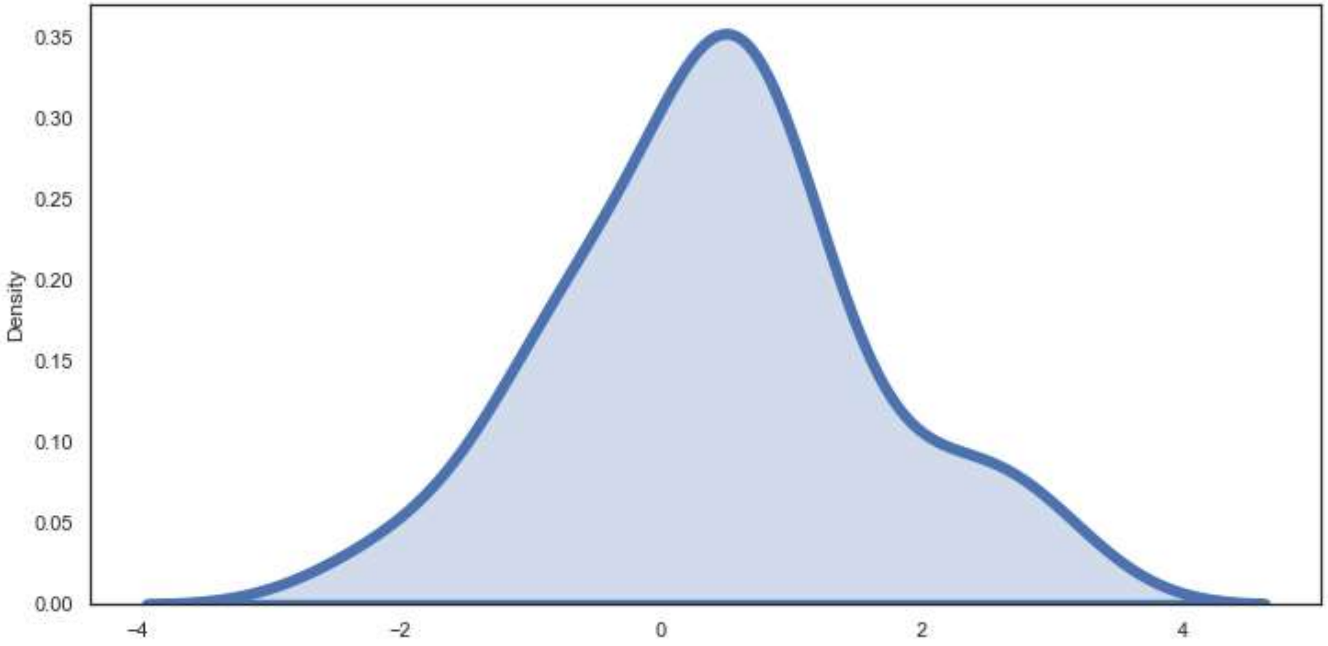
Seaborn



Seaborn



Seaborn



Seaborn

- You can change the kernel and bandwidth used which can make your KDE show more or less of the variance contained in the data.
- In the next lecture we will explore how to create these plots with python and seaborn!

Distribution Plots

PART TWO: CODING WITH SEABORN

Categorical Plots

Statistical Estimation within Categories
Part One: Understanding the Plots

PIERIAN DATA

Seaborn

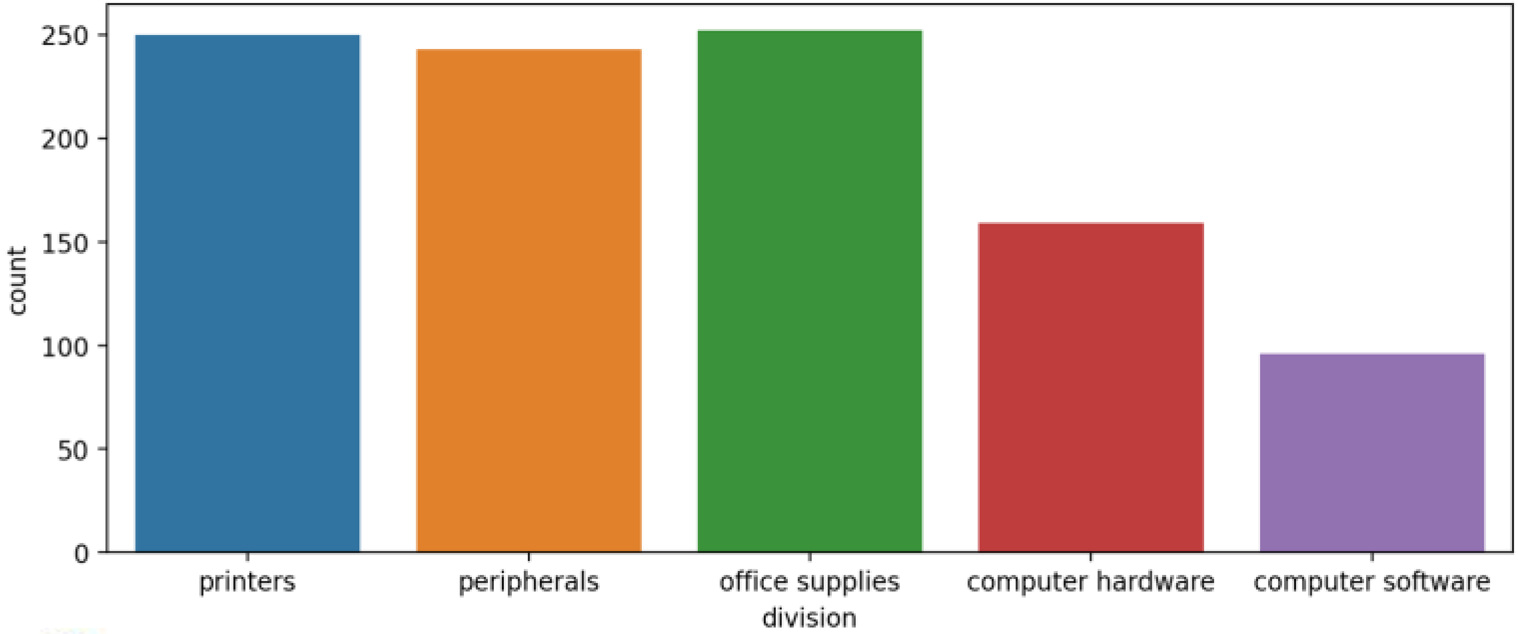
- The categorical plots discussed here will display a statistical metrics **per** a category.
- For example mean value per category or a count of the number of rows **per** category.
- It is the visualization equivalent of a groupby() call.

Seaborn

- The two main types of plots for this are:
  - countplot()
    - Counts number of rows per category.
  - barplot()
    - General form of displaying any chosen metric per category.

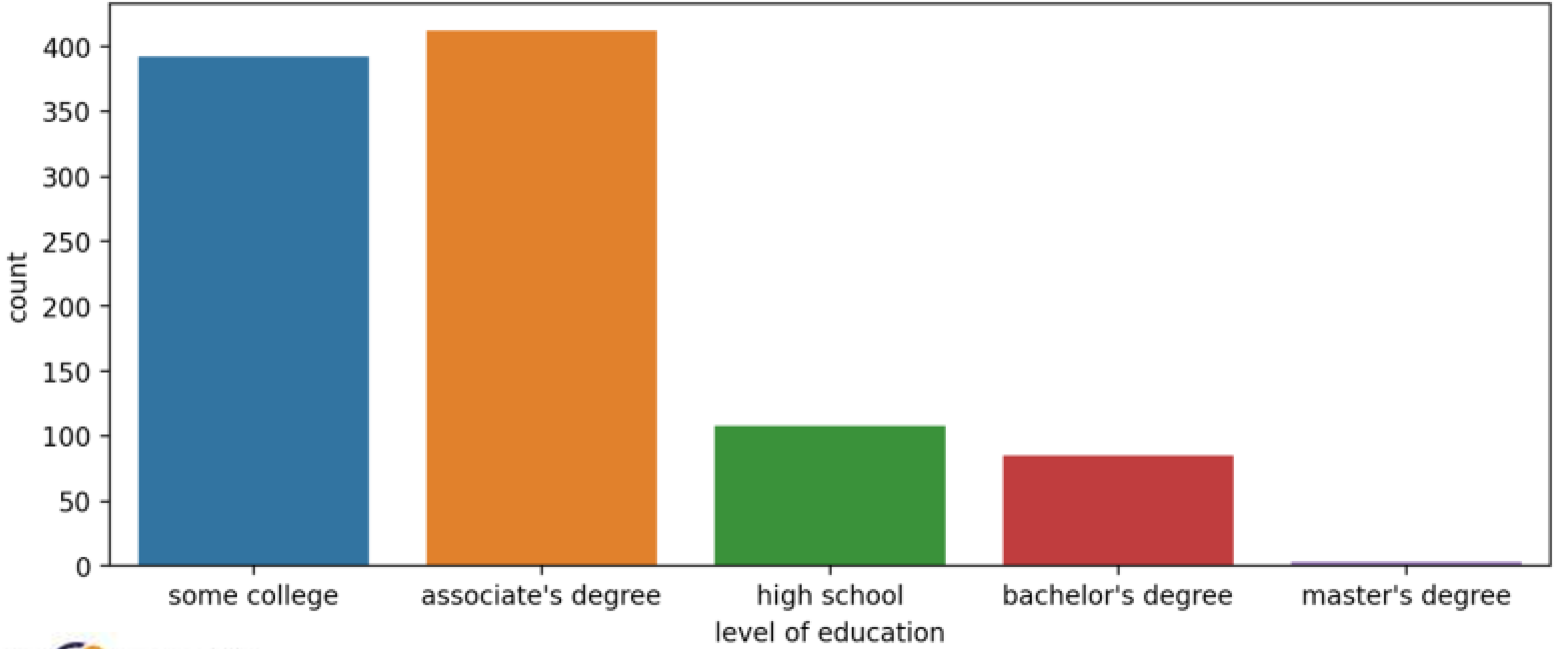
Seaborn

- Countplot for corporate divisions



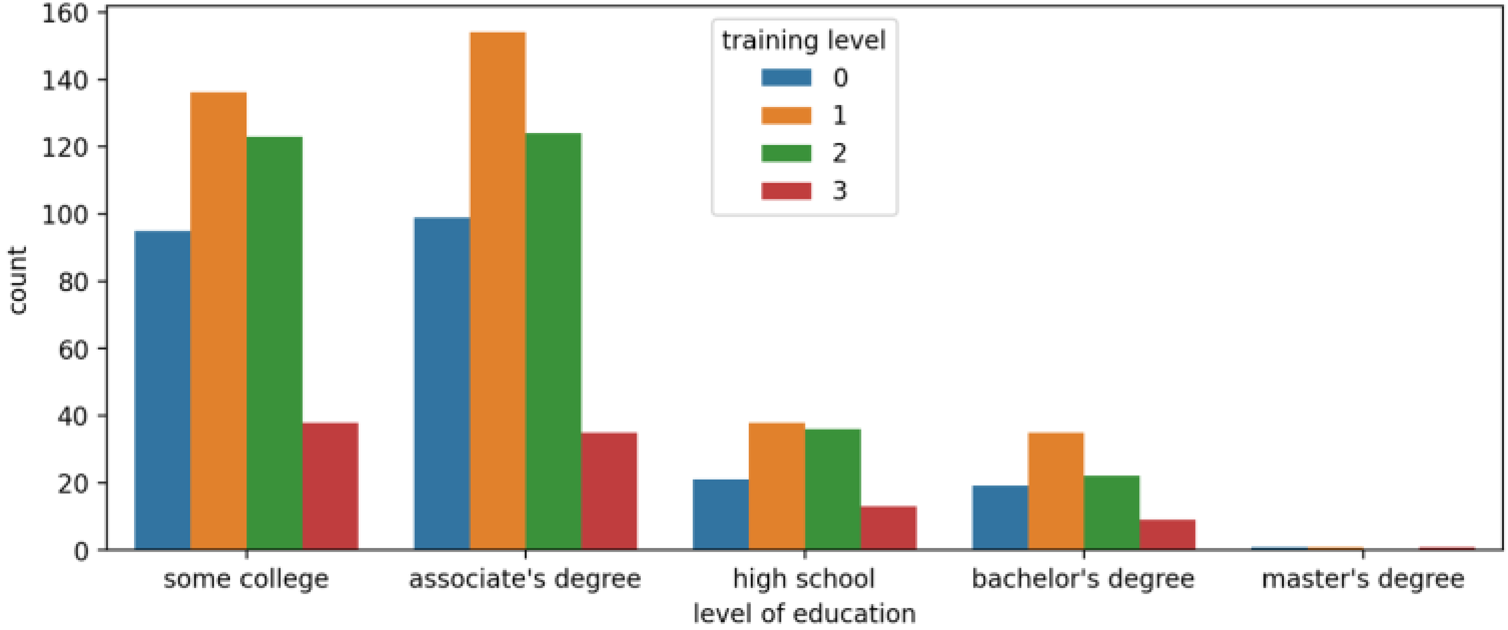
Seaborn

* Countplot for education level



Seaborn

* Countplot with additional hue separation



Seaborn

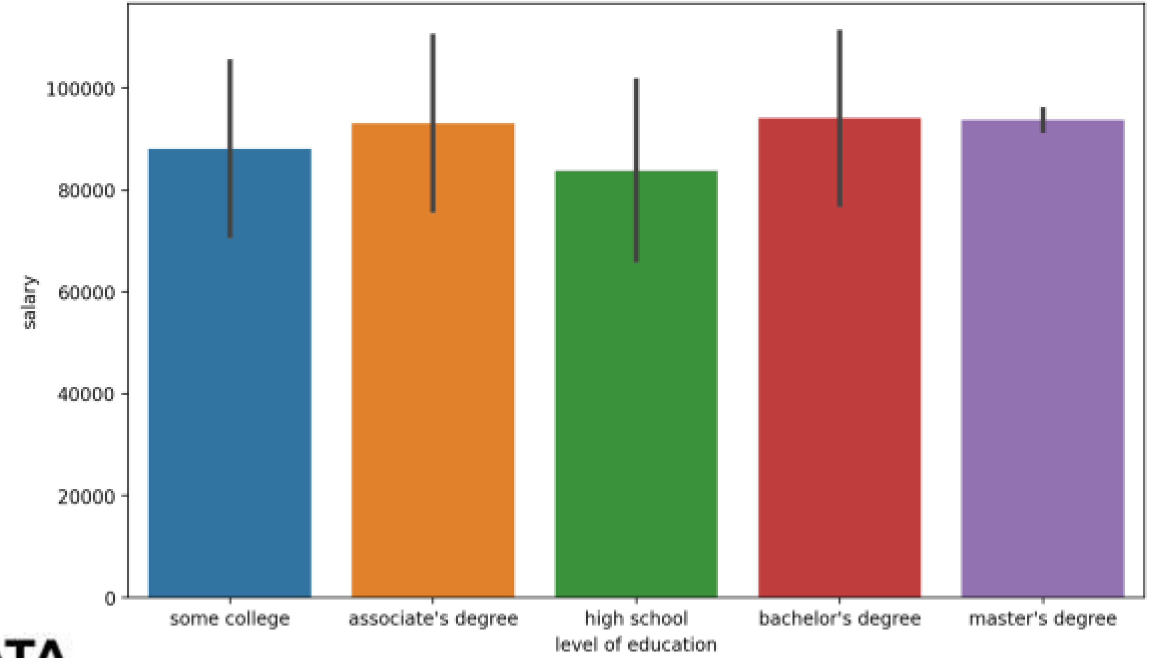
- The barplot is the general form that allows you to choose any measure or estimator for the y axis.
- We could plot the mean value and standard deviation per category instead.

Seaborn

* **Important Note!**
    * Be very careful with these plots, since the bar is filled and continuous, a viewer may interpret continuity along the y-axis which may be incorrect!
    * Always make sure to add additional labeling and explanation for these plots!

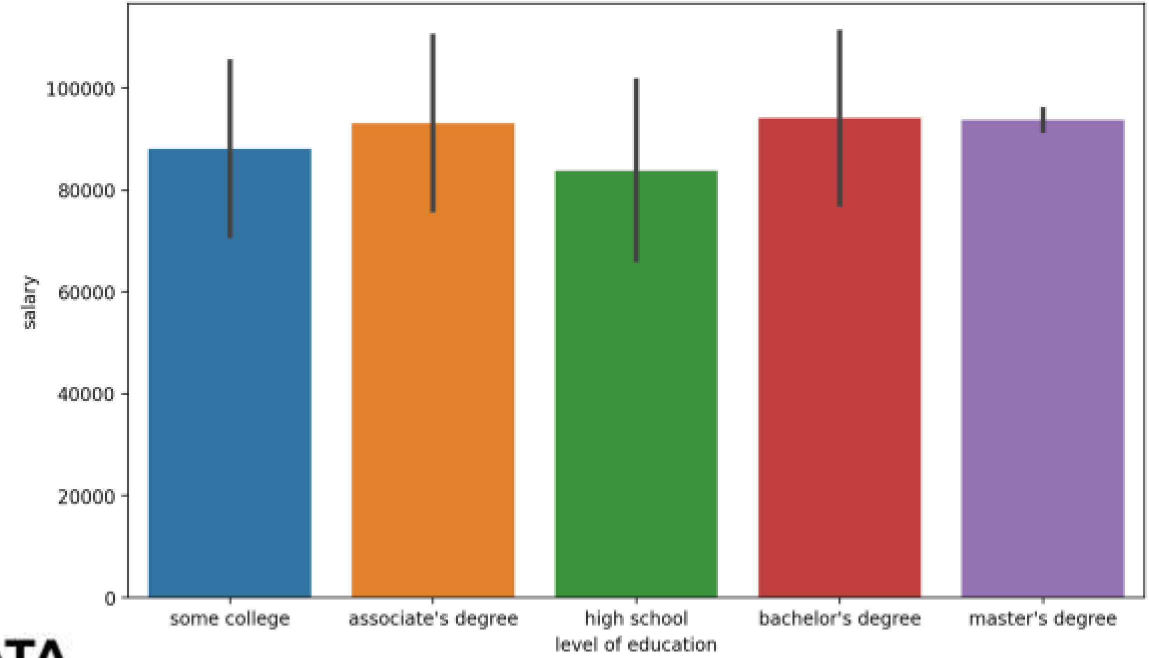
Seaborn

- Barplot showing mean and SD bar



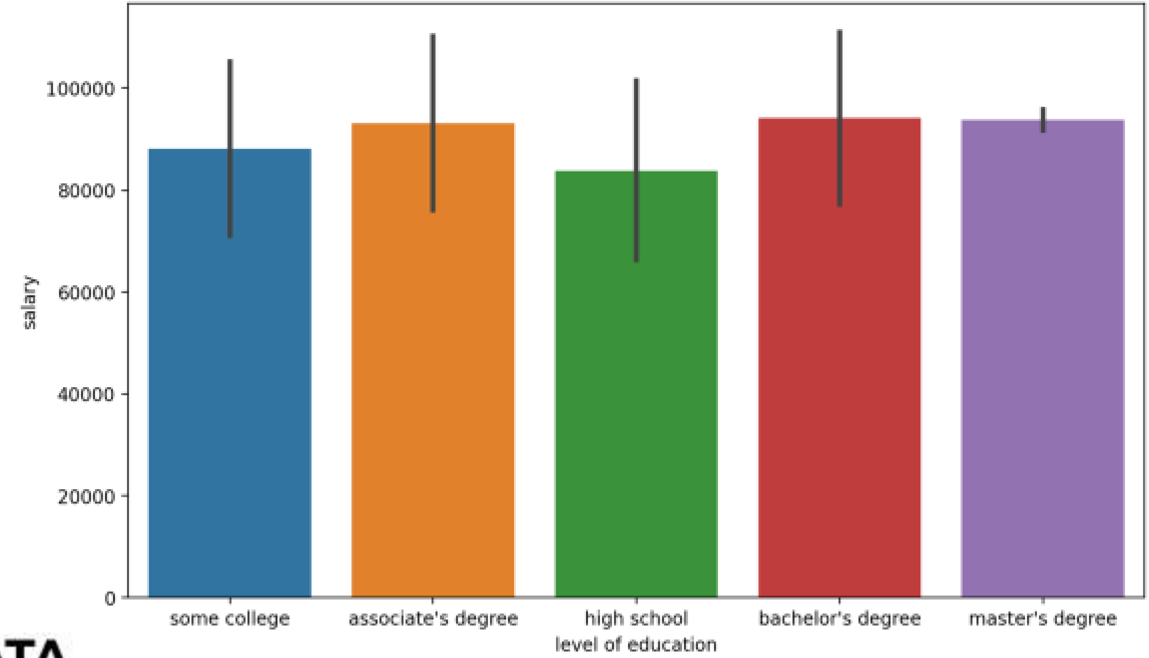
Seaborn

- Is this best shown with a barplot?



Seaborn

* Probably not! These are just single values!



Seaborn

- A simple table is probably better.

level of education	mean	std
associate's degree	93156.41	17066.06
bachelor's degree	94133.76	17007.09
high school	83887.35	17674.44
master's degree	93718.00	2497.63
some college	88115.84	17076.28

Seaborn

- Let’s explore coding out these plots with seaborn in the next lecture!

Categorical Plots

Statistical Estimation within Categories
Part Two: Coding the Plots

Categorical Plots

Distribution within Categories
Part One: Understanding the Plots

PIERIAN DATA

Seaborn

- We’ve explored distribution plots for a single feature, but what if we want to compare distributions across categories?
- For example, instead of the distribution of everyone’s salary, we can compare the distributions of salaries **per** level of education.

Seaborn

- We will first separate out each category, then create the distribution visualization.
- Let’s explore what plot types we have available….

Seaborn

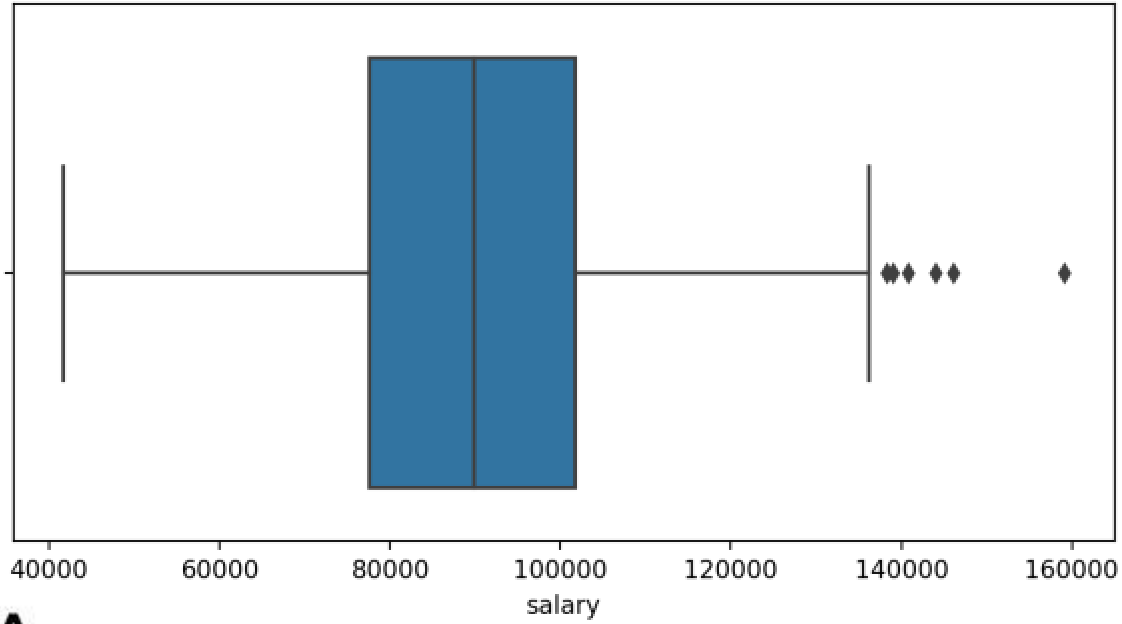
- Distribution within Categories
  - Boxplot
  - Violinplot
  - Swarmplot
  - Boxenplot (Letter-Value Plot)
- Let’s explore understanding these plots on the previous salary dataset.

Seaborn

- The Boxplot displays the distribution of a continuous variable.
- It does this through the use of quartiles.
- Quartiles separate out the data into 4 equal number of data points :
    - 25% of data points are in bottom quartile.
    - 50th percentile (Q2) is the median.

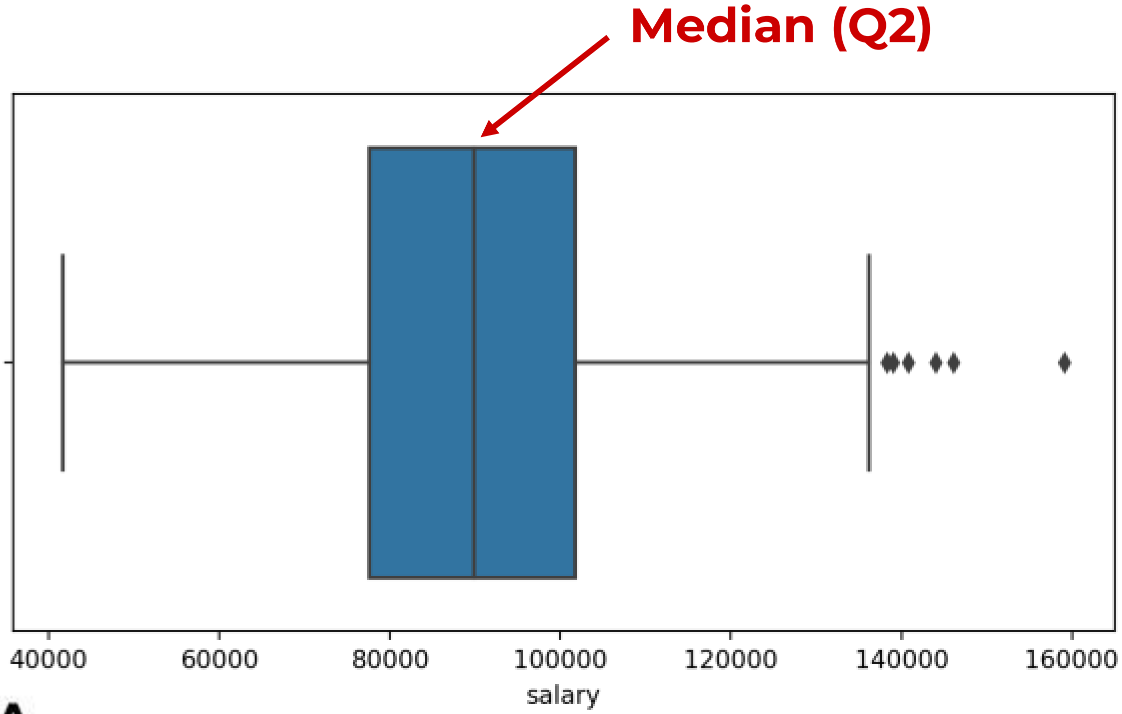
Seaborn

* Boxplot on single feature:



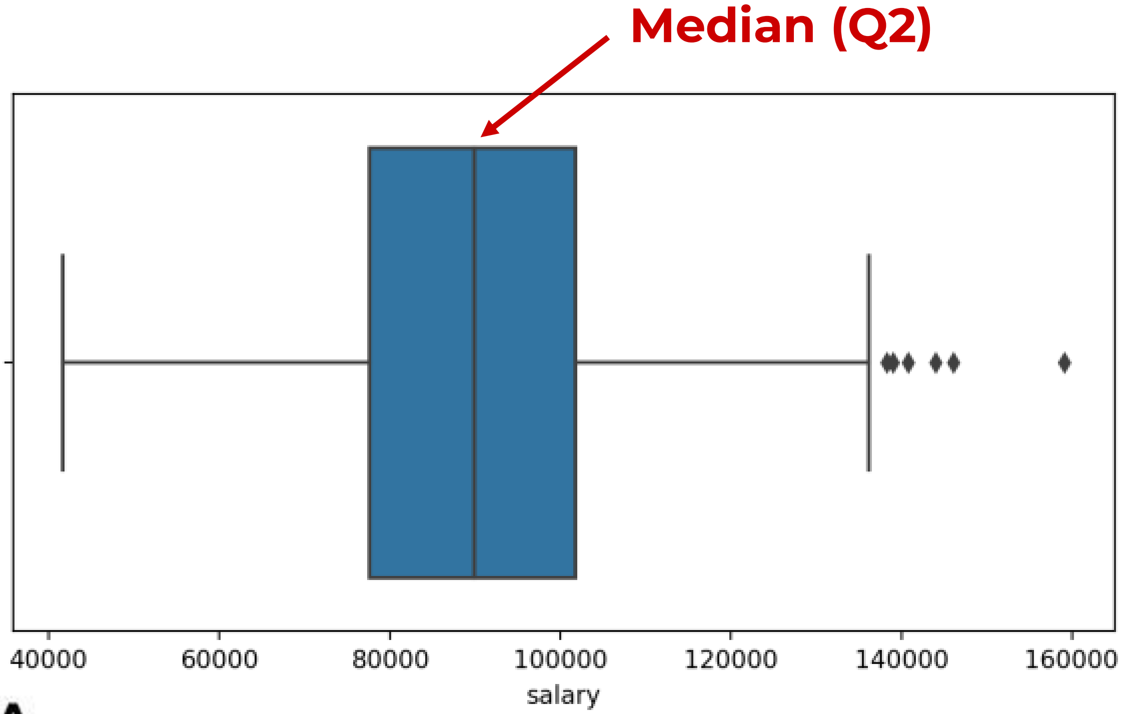
Seaborn

- Median is 50th percentile



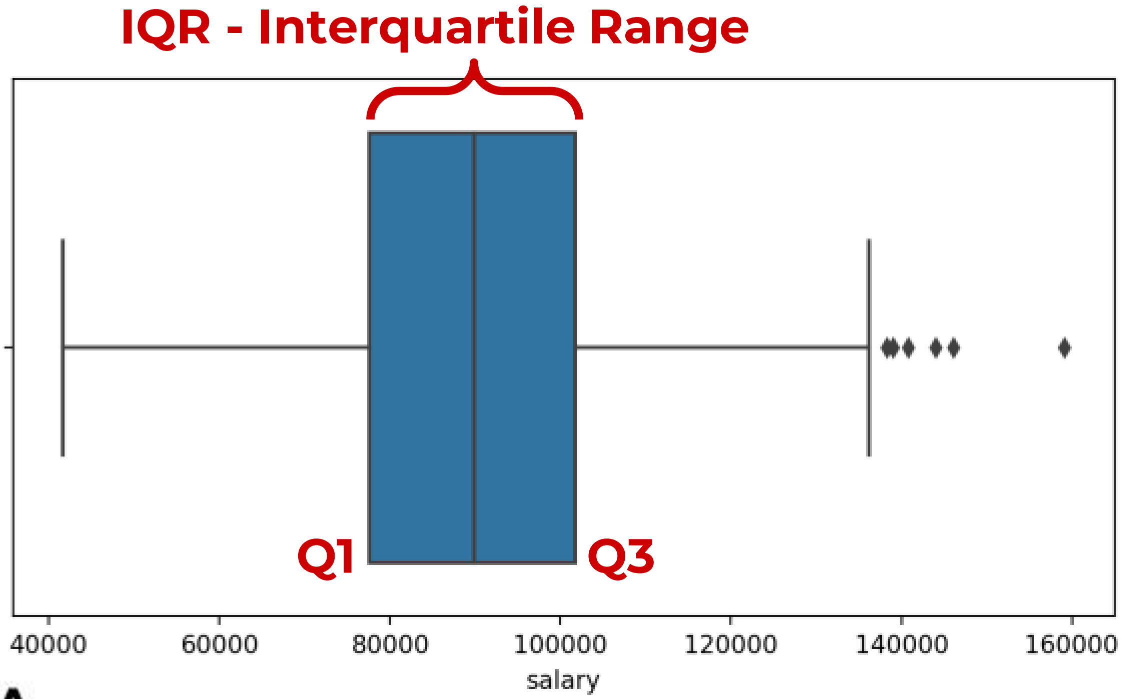
Seaborn

- Median splits data in half



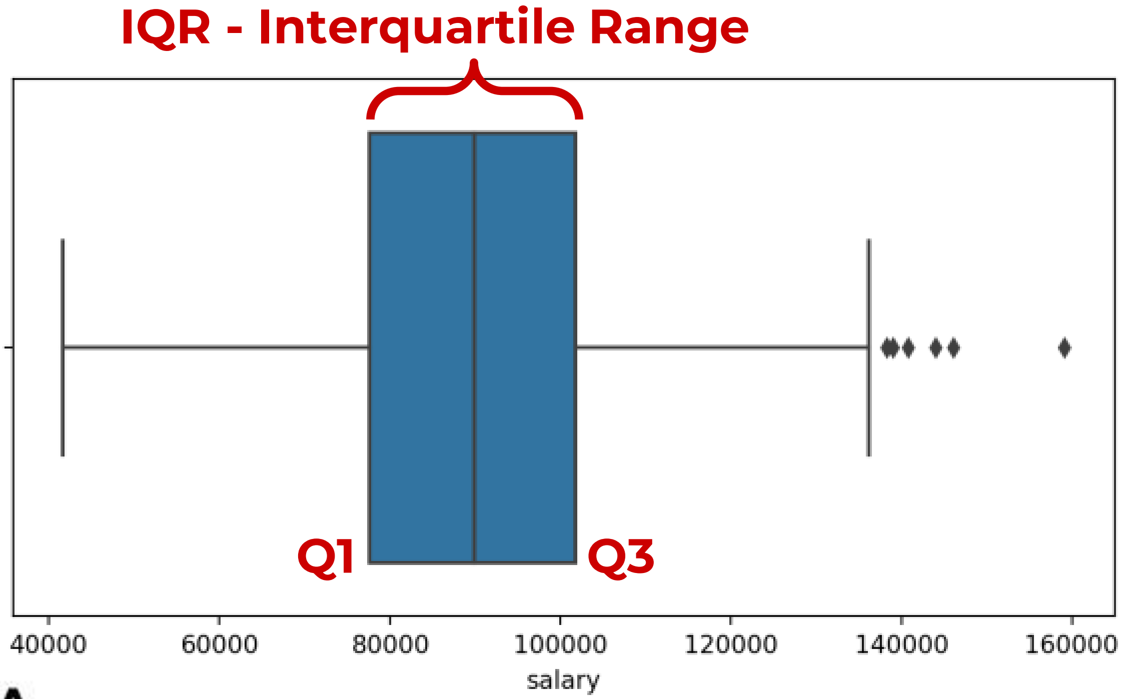
Seaborn

- IQR defines the box width



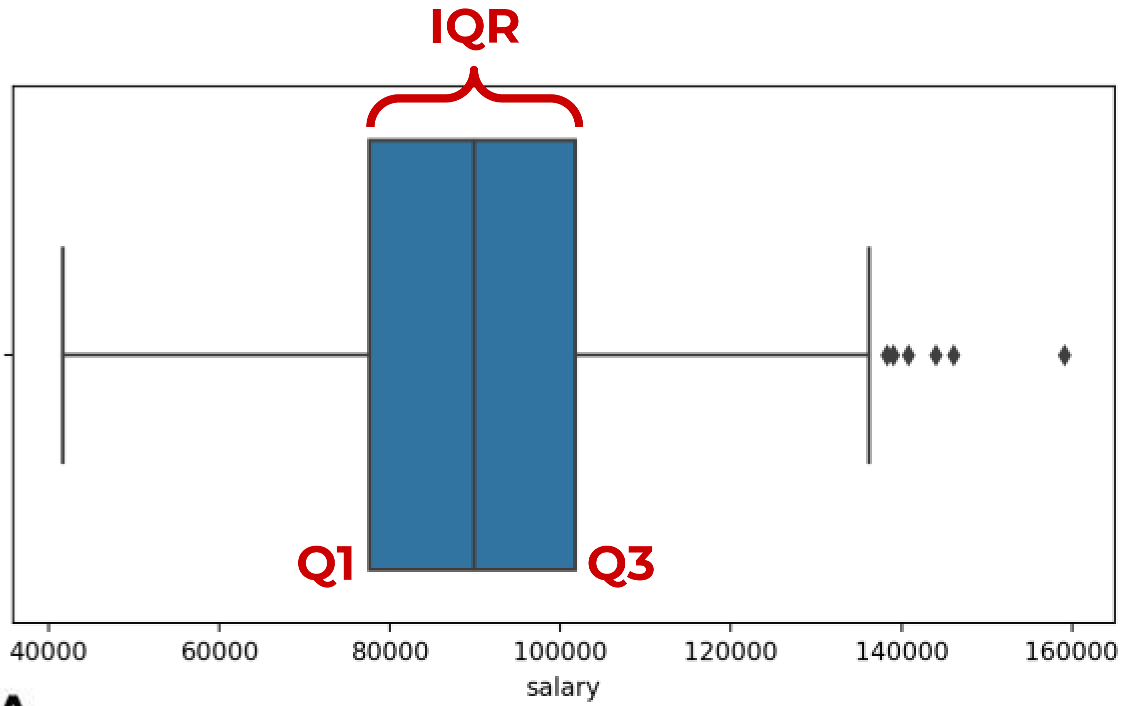
Seaborn

- 50% of all data points are inside the box



Seaborn

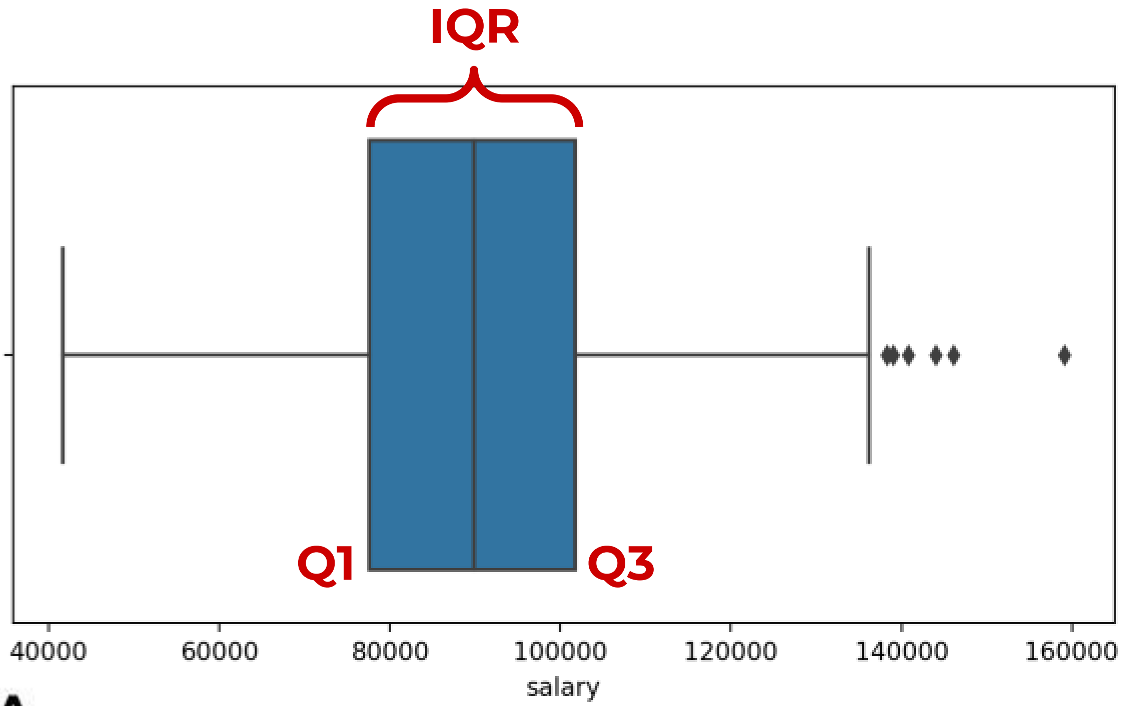
- 50% of all data points are inside the box



PIERIAN DATA

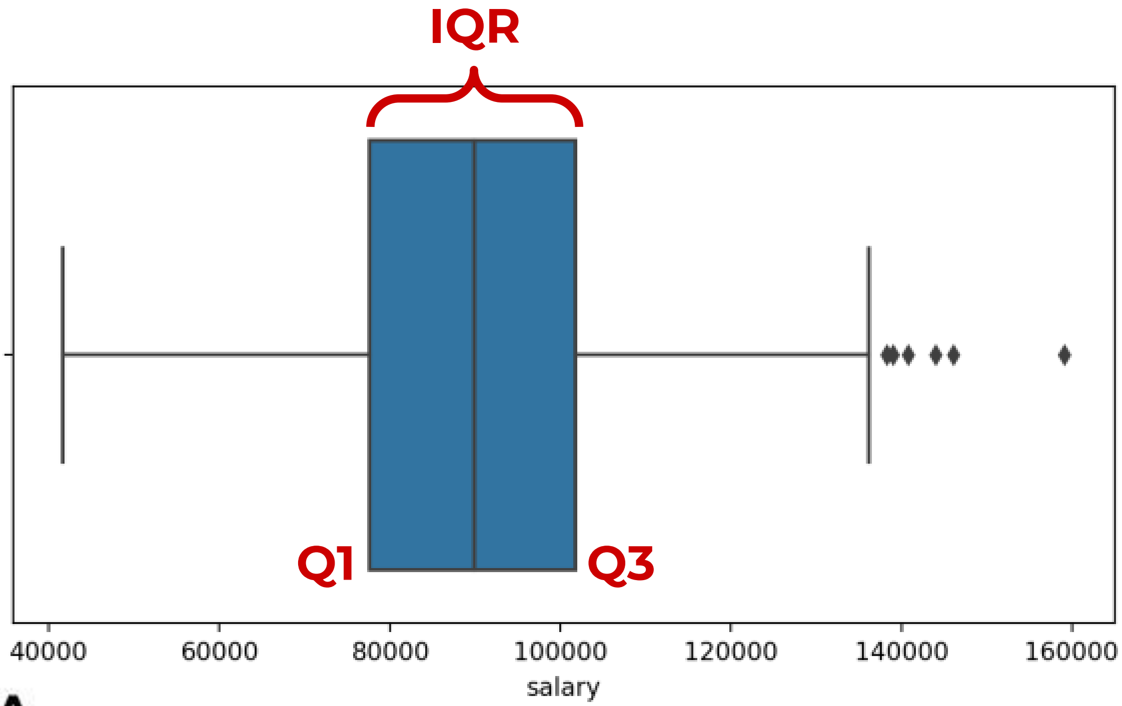
Seaborn

- Q1 is the 25th percentile



Seaborn

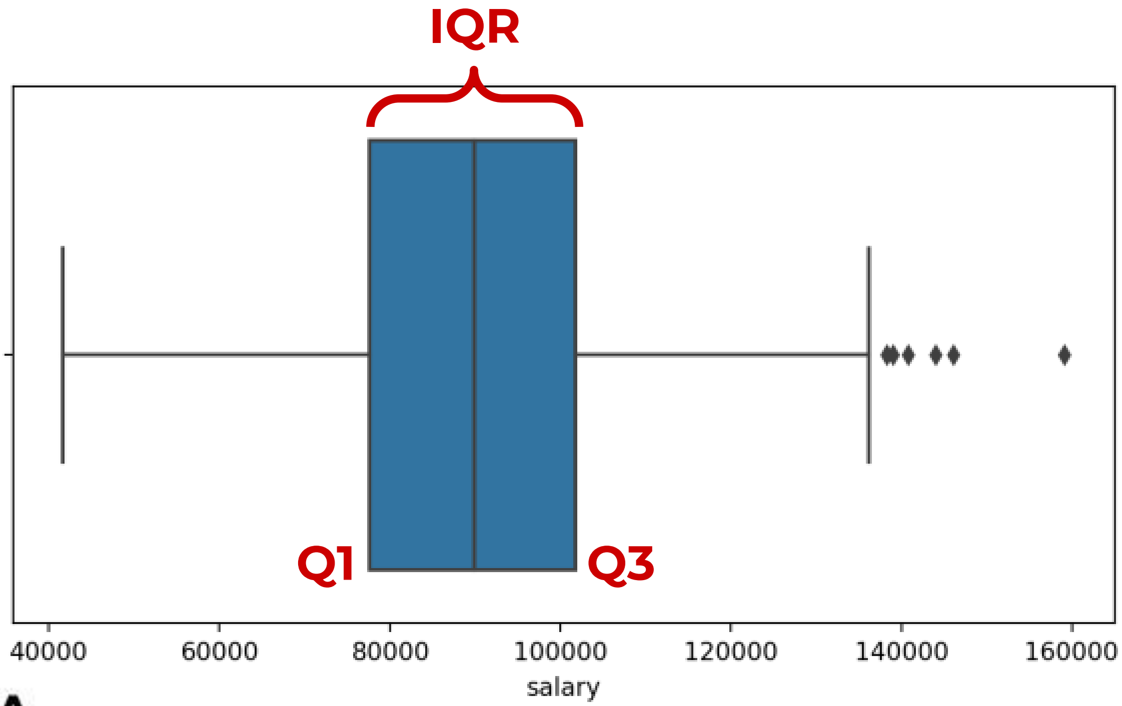
- 25% of data points are below Q1



PIERIAN DATA

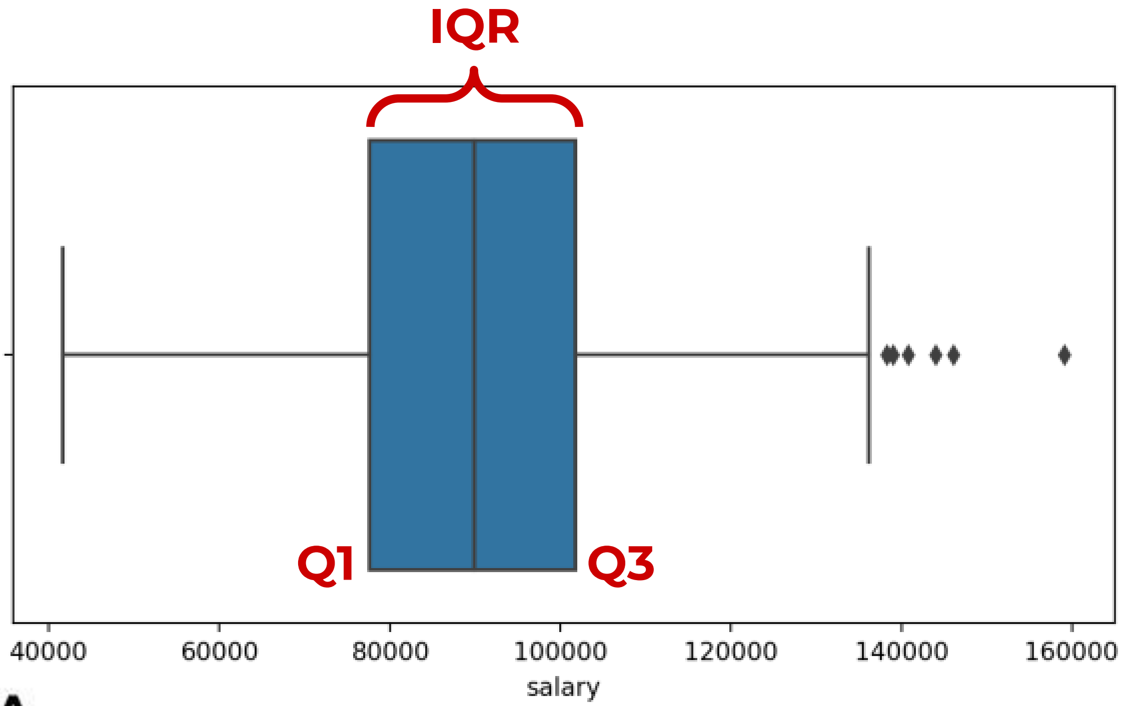
Seaborn

- Q3 is the 75th percentile



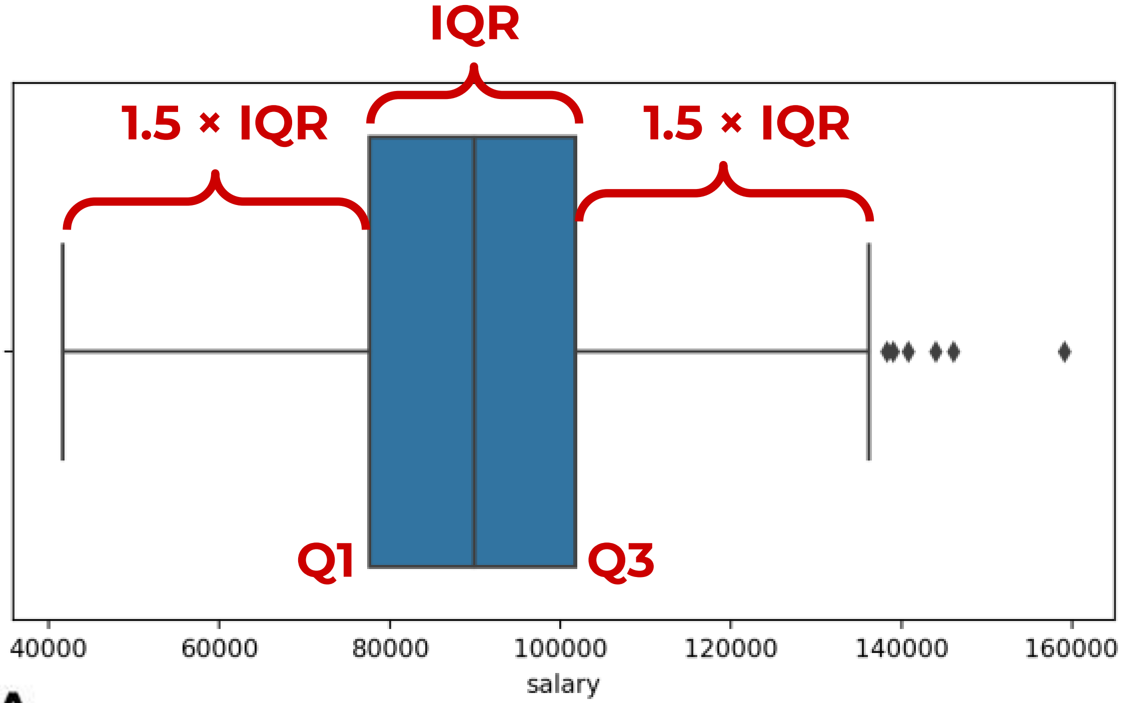
Seaborn

- 25% of all points are above Q3



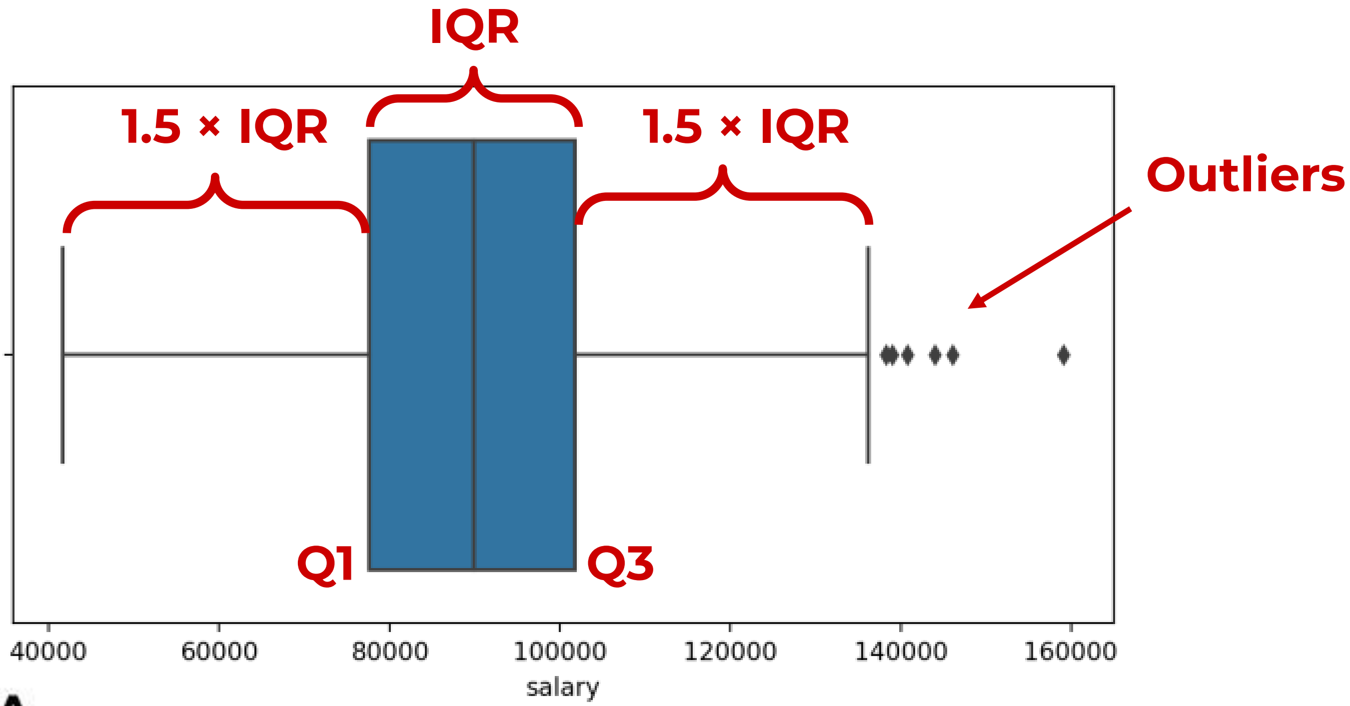
Seaborn

- The "whiskers" are defined by 1.5 × IQR



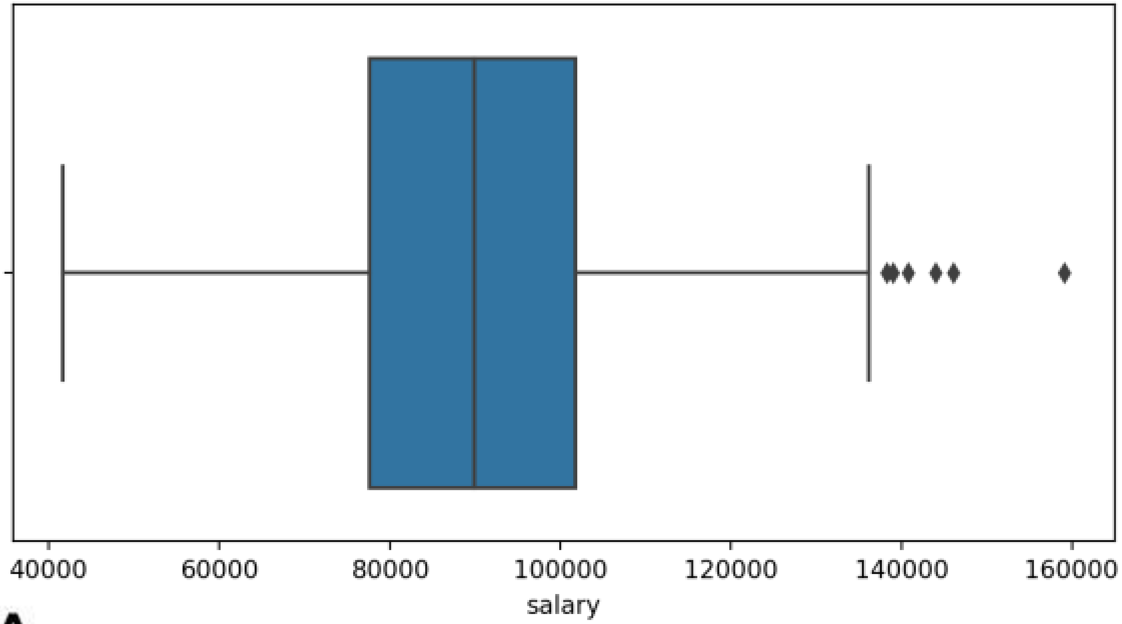
Seaborn

- Outside of the whiskers are outliers



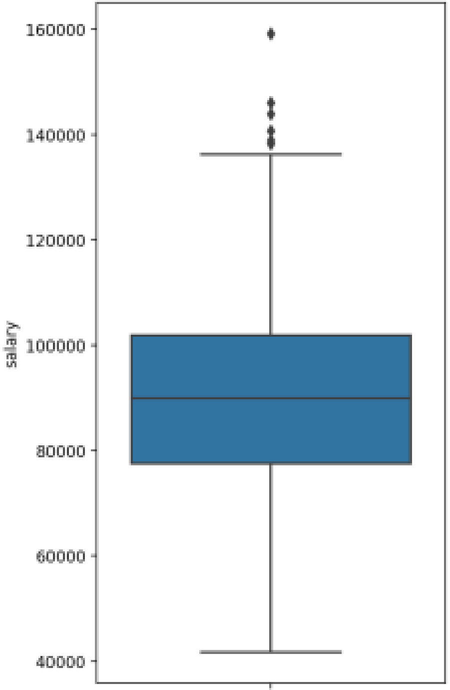
Seaborn

- Boxplot quickly gives statistical distribution information in a visual format:



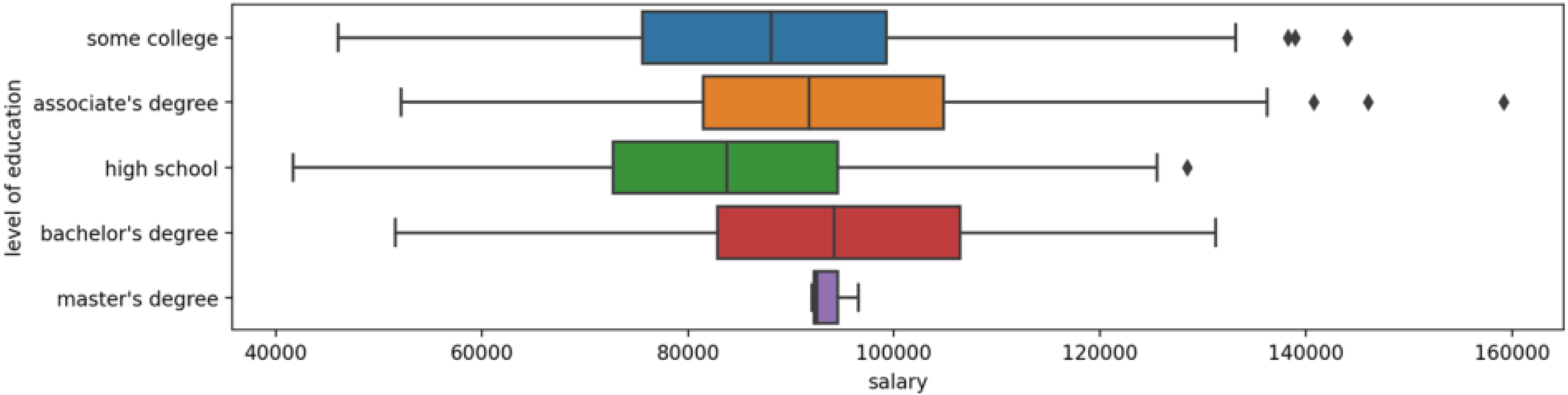
Seaborn

- Boxplot can be oriented vertically or horizontally.



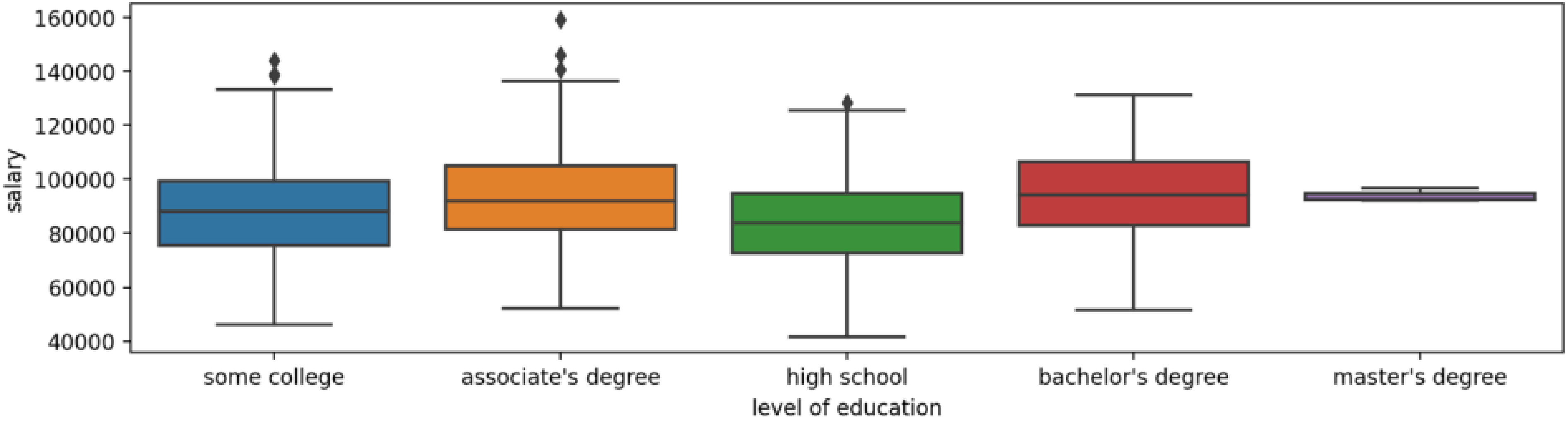
Seaborn

- We can create a box plot **per** category!



Seaborn

- We can create a box plot **per** category!



Seaborn

- The violin plot plays a similar role as the box plot.
- It displays the probability density across the data using a KDE.
- We can imagine it as a mirrored KDE plot.

Seaborn

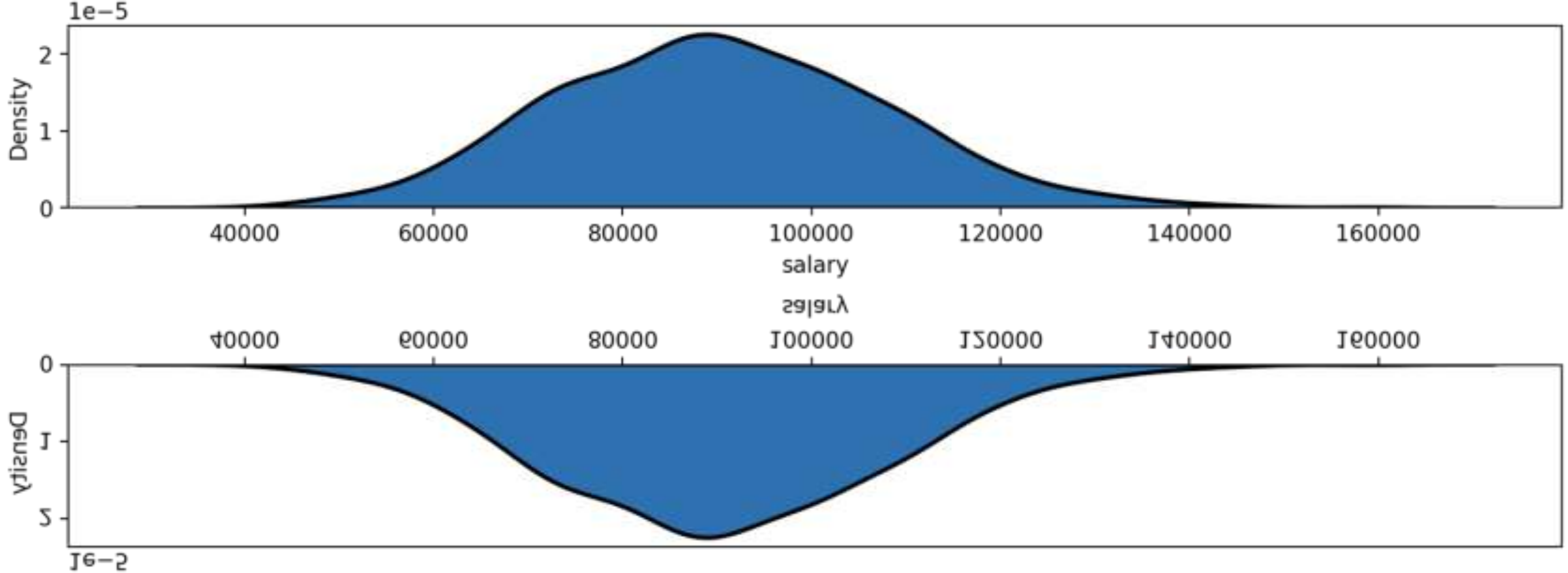
* The violin plot plays a similar role as the box plot.
* It displays the probability density across the data using a KDE.
* We can imagine it as a mirrored KDE plot.

Seaborn

- We take the KDE of a single feature:

Seaborn

- We could then “mirror” it:

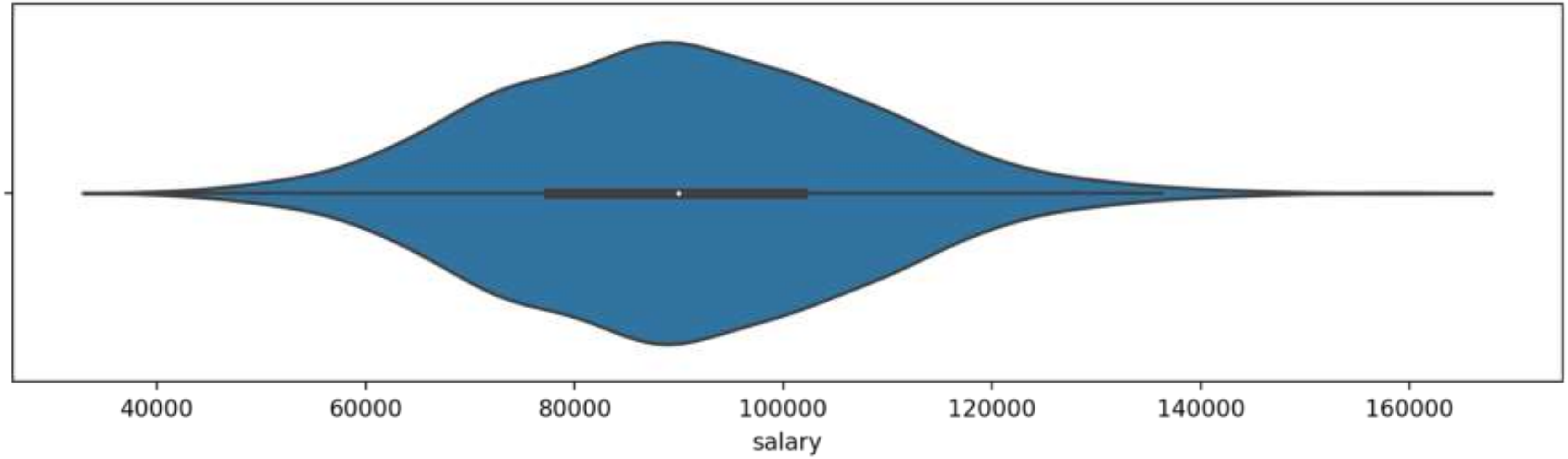


Seaborn

- Then combine it to get the violin plot:

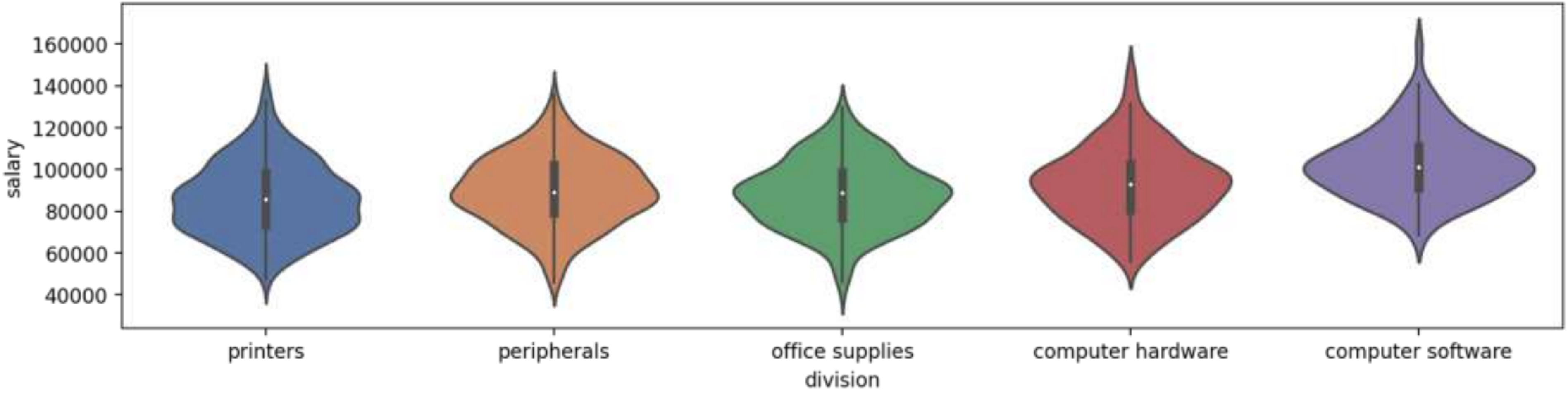
Seaborn

- Then combine it to get the violin plot:



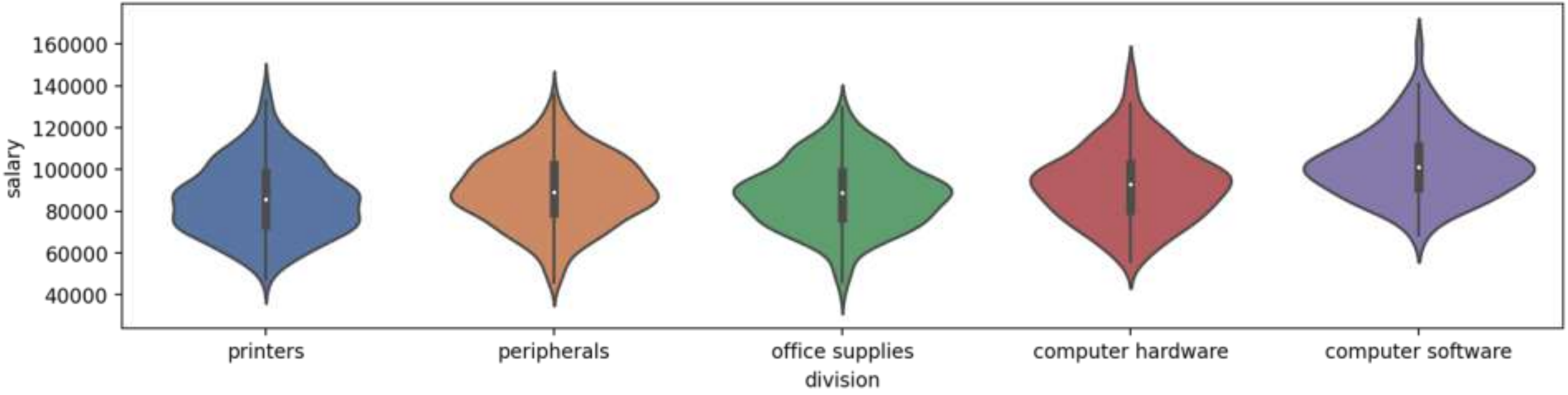
Seaborn

* The violin plots can then be created **per** category:



Seaborn

- The violin plots can then be created **per** category:



Seaborn

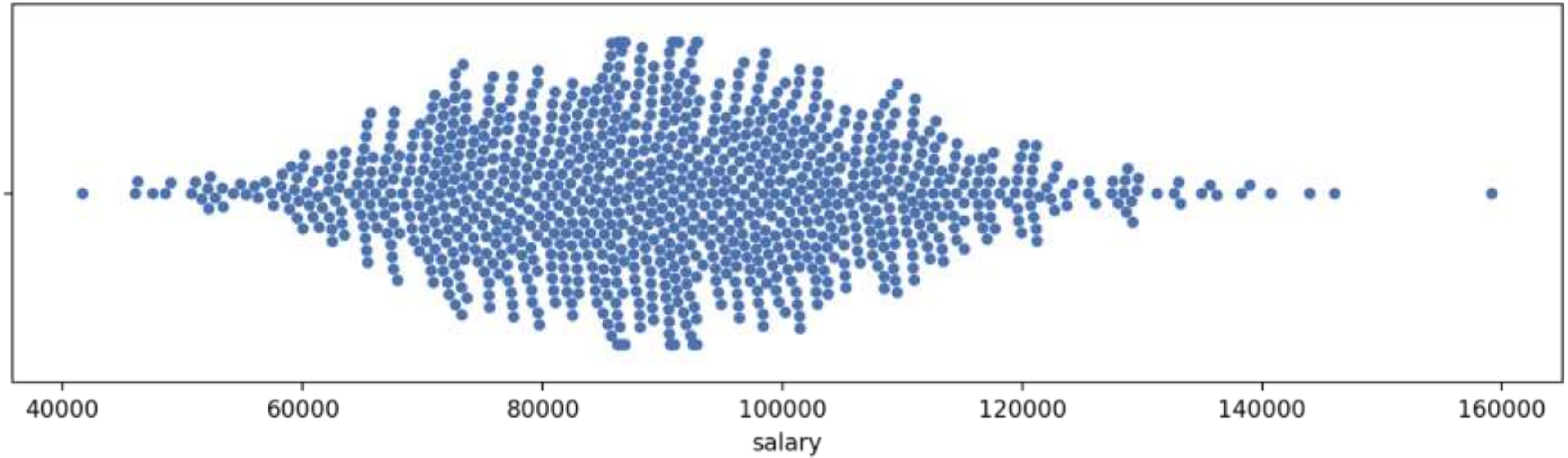
- A few more less common categorical distribution plots are the swarmplot and the boxenplot.
- Let’s quickly explore these plot types...

Seaborn

- The swarmplot is very simple and simply shows all the data points in the distribution.
- For very large data sets, it won’t show all the points, but will display the general distribution of them.

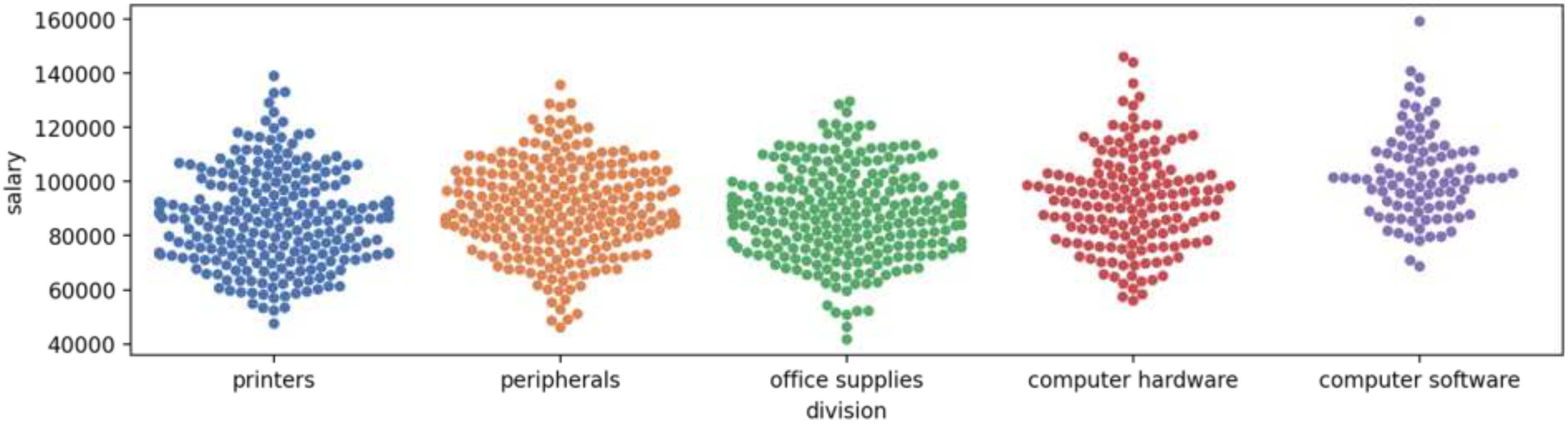
Seaborn

- Swarmplot



Seaborn

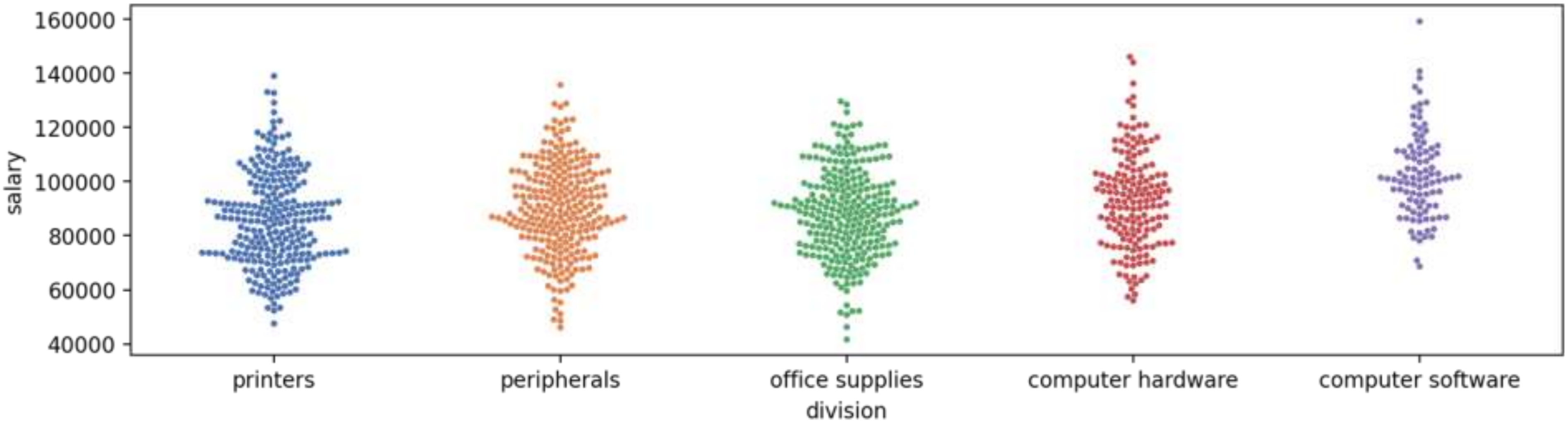
- Swarmplot per Category



PIERIAN DATA

Seaborn

- Change size of points to show more



Seaborn

- The boxenplot (Letter-value plot) is a relatively new plot developed in 2011 by Heike Hofmann, Karen Kafadar, and Hadley Wickham.
- Its mainly designed as an expansion upon the normal box plot.
- Make sure to read the linked paper in the notebook if you end up using this plot!

Seaborn

- Note that the boxenplot is currently very uncommon, in fact a Google search will often auto-correct this to a “boxplot” call.
- Only use this plot type if you know your audience is familiar with it.
- Let’s briefly explore the boxenplot and its benefits.

Seaborn

- Using a system of letter-values we can use multiple quantiles instead of strictly quartiles.

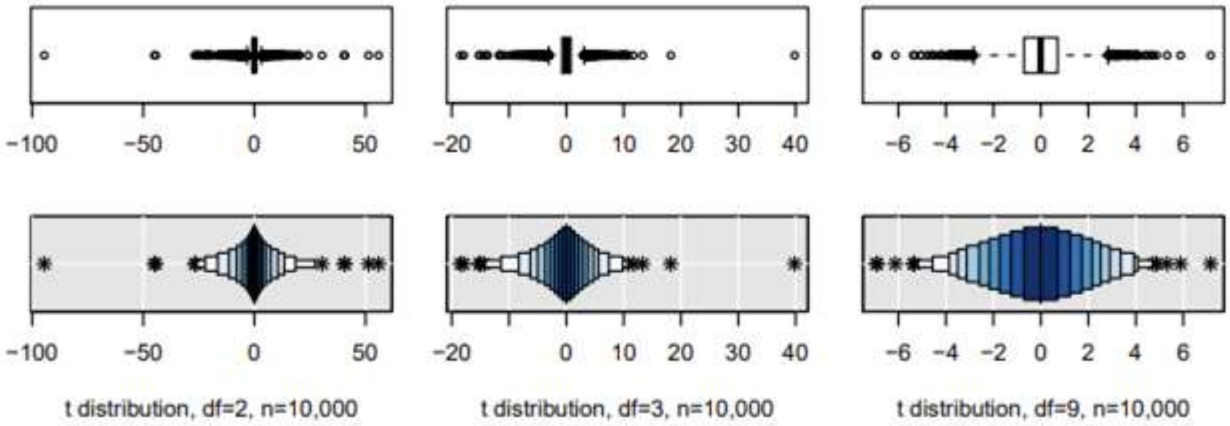
LV	ideal tail area	rough %	odds (2^i)	SEfactor	n-equiv*
M	.50	50.0%	2	1.253314	
F	.25	25.0%	4	1.36	1.0
E	.125	12.5%	8	1.60	1.4
D	.0625	6.25%	16	1.96	2.1
C	.03125	3.13%	32	2.47	3.3
B	.015625	1.56%	64	3.16	5.4
A	.0078125	0.8%	128	4.10	9.1
Z	.00390625	0.4%	256	5.37	15.6
Y	.001953125	0.2%	512	7.11	27.3
X	.0009765625	0.1%	1,024	9.48	48.4
W	.00048828125	0.05%	2,048	12.70	87.0
V	.000244140625	0.024%	4,096	17.11	157.7
U	.0001220703125	0.012%	8,192	23.14	288.5
T	.00006103515625	0.006%	16,384	31.40	531.3
S	.000030517578125	0.003%	32,768	42.75	984.4
R	.0000152587890625	0.0015%	65,536	58.34	1833.5
Q	.00000762939453125	0.0008%	131,072	79.80	3430.5
P	.000003814697265625	0.0004%	252,144	109.38	6444.3
O	.0000019073486328125	0.0002%	504,288	150.19	12149.2
N	.00000095367431640625	0.0001%	1,008,576	206.55	22977.6

Table 1: First 20 letter values. Ideal `tail area` is 2^{-i}, $i = 1, \ldots, 20$. `rough%` rounds $2^{-i} \times 100\%$ to the first 1 or 2 nonzero digits. `odds` expresses tail area as `1 in` 2^i. `SEfactor` gives the factor for the asymptotic standard error of the order statistic (from a Gaussian population, variance σ^2) corresponding to `tail area`, i.e., $SE(LV) \approx$ `SEfactor` $\times \sigma/\sqrt{n}$, where `SEfactor` $= \sqrt{p_i(1-p_i)}/\phi(\Phi^{-1}(p_i))$, p_i = `tail area` $= 2^{-i}$. `n-equiv` = $(\texttt{SEfactor}/1.362633)^2$ which gives the factor of increase in sample size for the uncertainty in that letter value to be the same as that for the fourth; e.g., need $1.4n$ (respectively, $2.1n$) observations for the eighth (respectively, sixteenth) to have the same uncertainty as that of a fourth from a sample of size n.

PIERIAN DATA

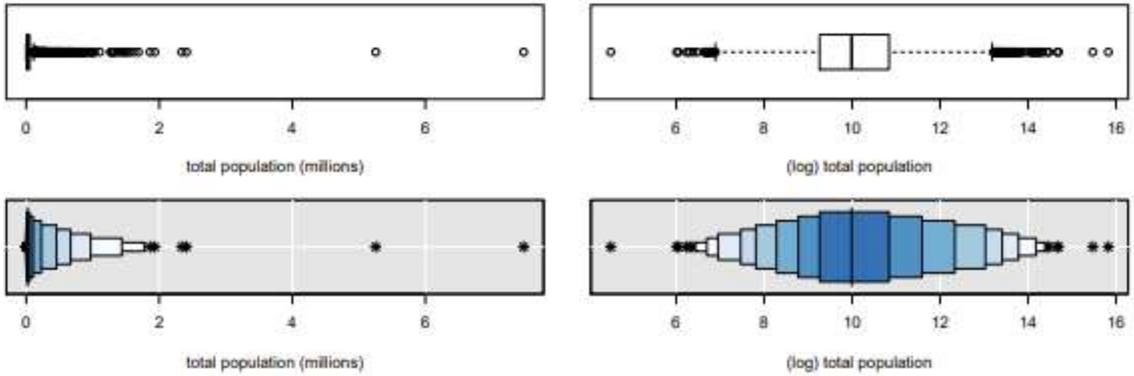
Seaborn

- Boxenplot showing letter-value quantiles to display against a standard boxplot:



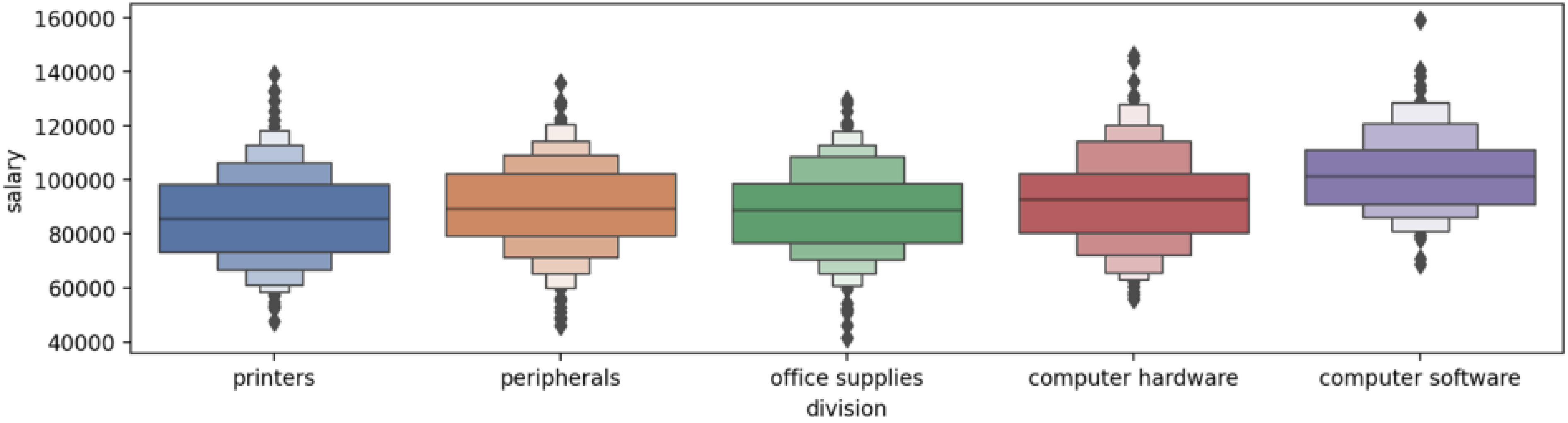
Seaborn

- Boxenplot showing letter-value quantiles to display against a standard boxplot:



Seaborn

- Boxenplot in seaborn:



Seaborn

- Keep in mind the main purpose of data visualizations is to inform, not confuse or show-off various esoteric plots!
- In the next lecture we will explore coding out these plot types.

Categorical Plots

Distribution within Categories
Part Two: Creating the Plots

Comparison Plots

Part One: Understanding the Plots

Seaborn

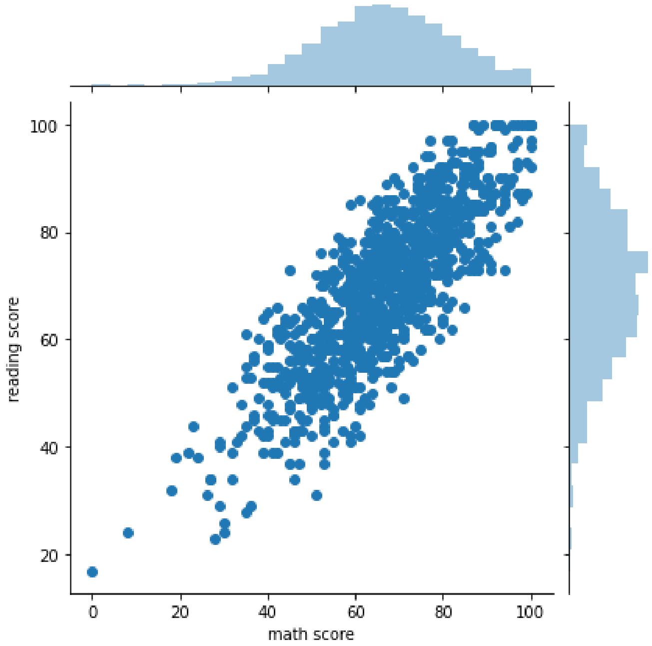
- Comparison plots are essentially 2-dimensional versions of the plots we've learned about so far.
- The two main plots types discussed here:
  - jointplot()
  - pairplot()

Seaborn

- jointplot()
  - We can map histograms to each feature of a scatterplot to clarify the distributions within each feature.
  - We can also adjust the scatterplot to be a hex plot or a 2D KDE plot.

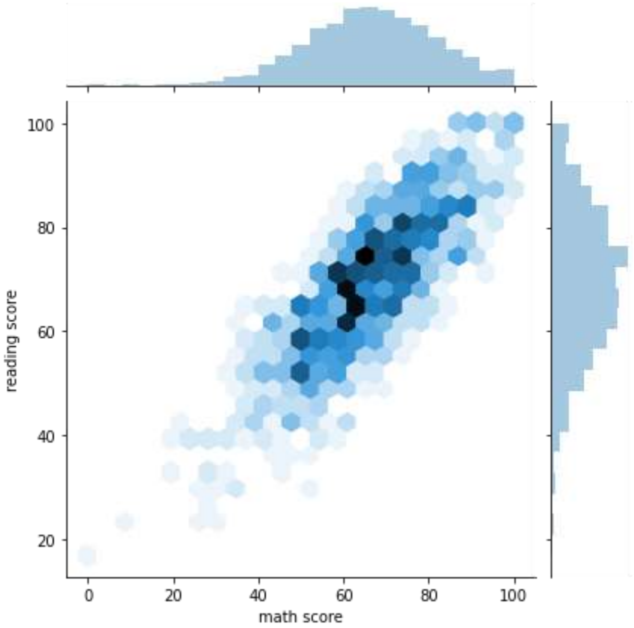
Seaborn

- Histograms with Scatterplot:



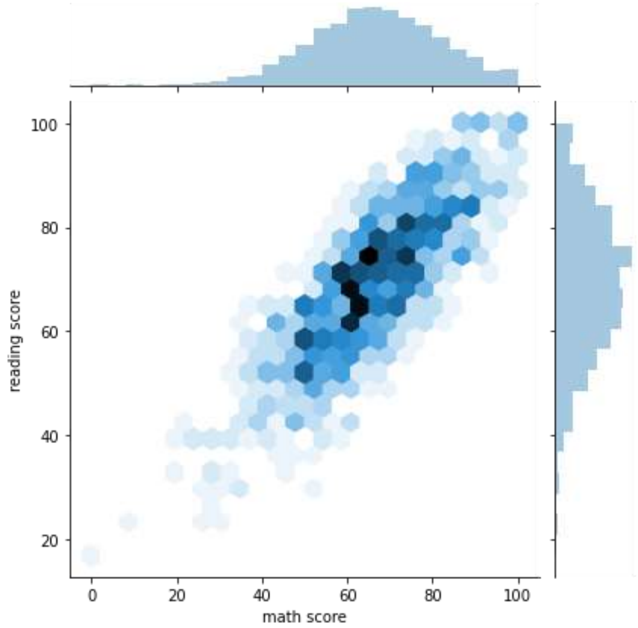
Seaborn

- Histograms with hexagons:



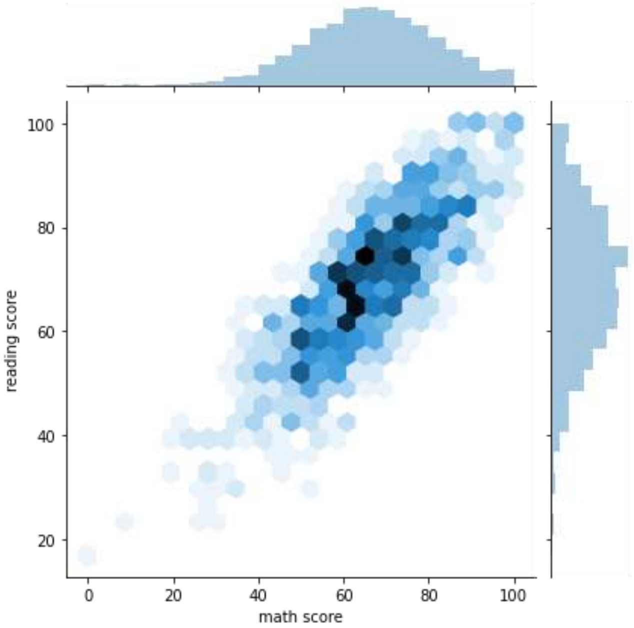
Seaborn

* Hexagons are dark the more points fall into their area.



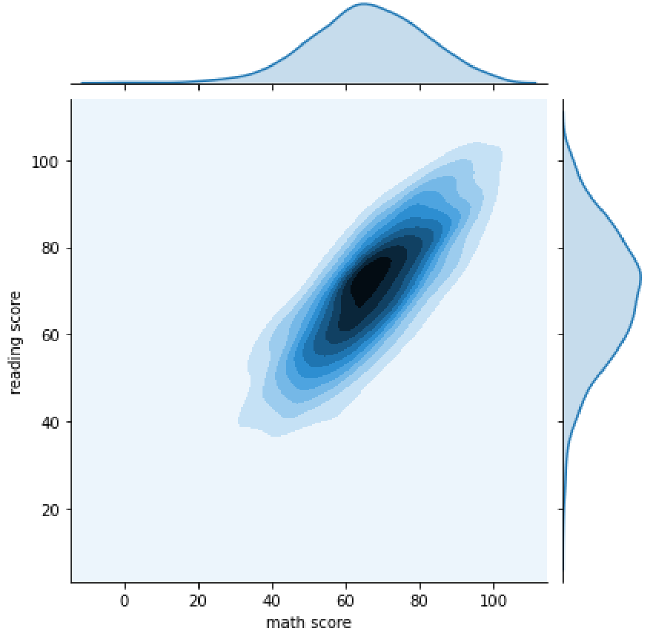
Seaborn

- Hexagons are useful when many points overlap.



Seaborn

- 2D KDE plots show shaded distribution between both KDEs:



Seaborn

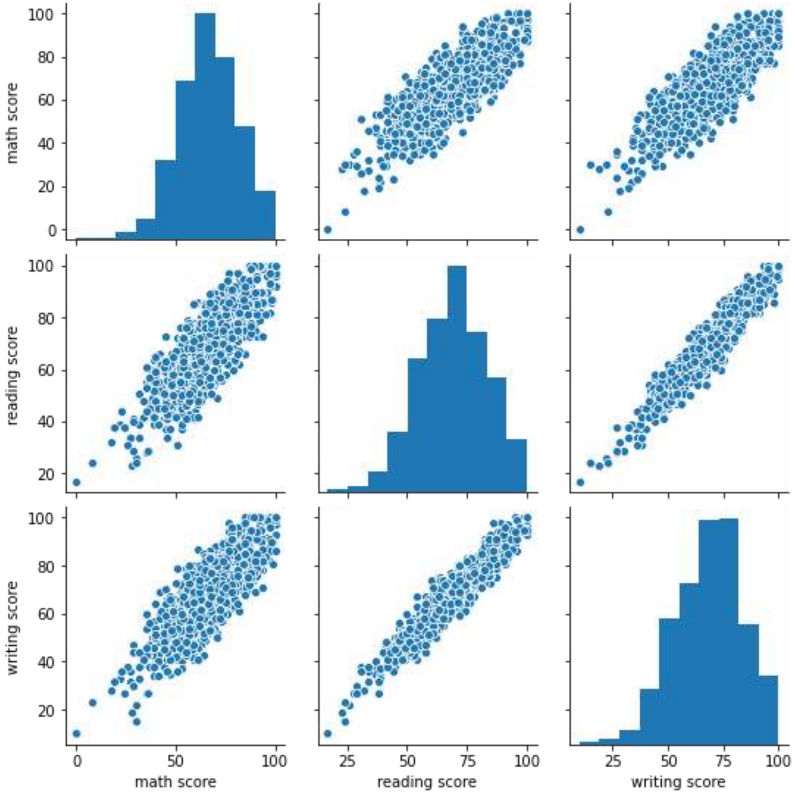
* pairplot()
    * The pairplot() is a quick way to compare all numerical columns in a DataFrame.
    * It automatically creates a histogram for each column and a scatterplot comparison between all possible combinations of columns.

Seaborn

- pairplot()
  - *Warning!*
    - pairplot() can be CPU and RAM intensive for large DataFrames with many columns.
    - It is a good idea to first filter down to only the columns you are interested in.

Seaborn

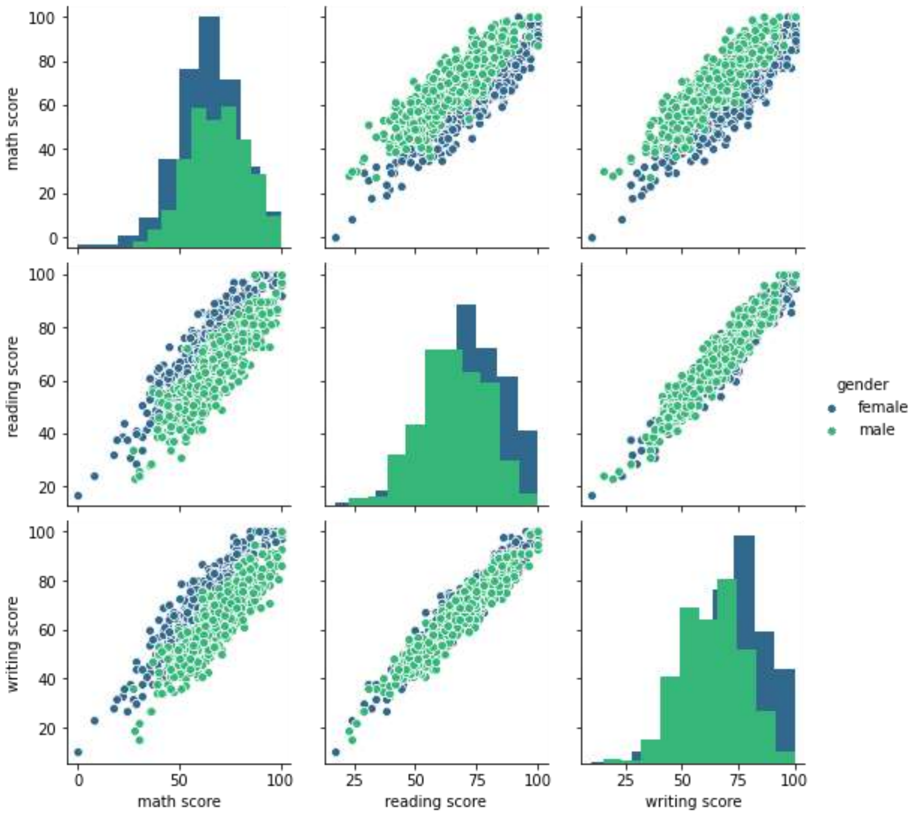
- pairplot()



PIERIAN DATA

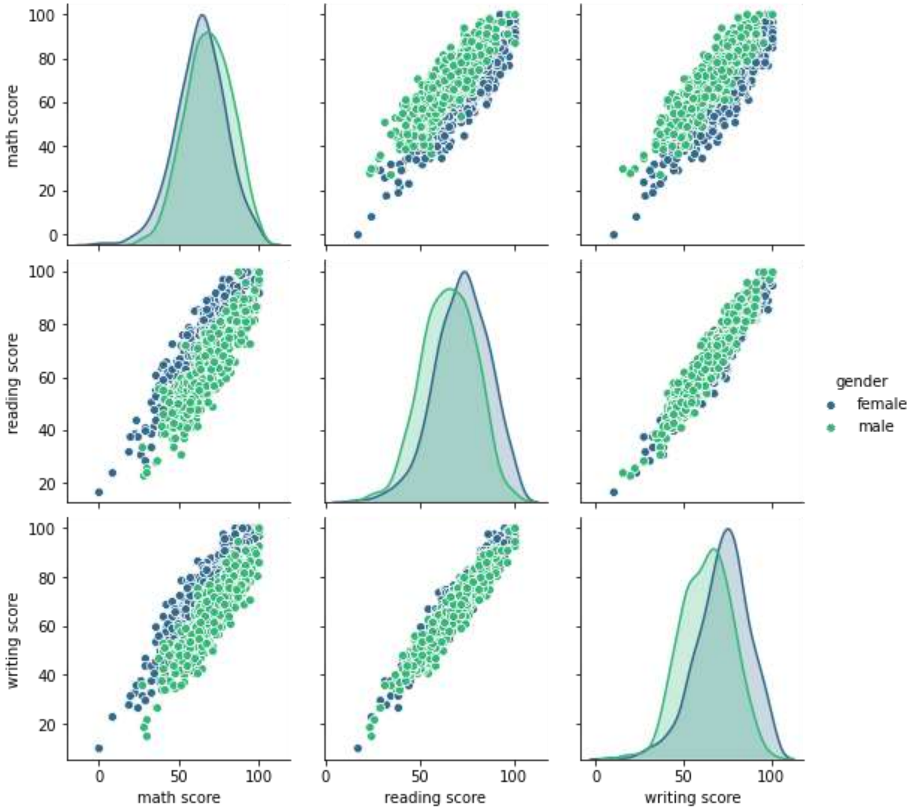
Seaborn

- pairplot()



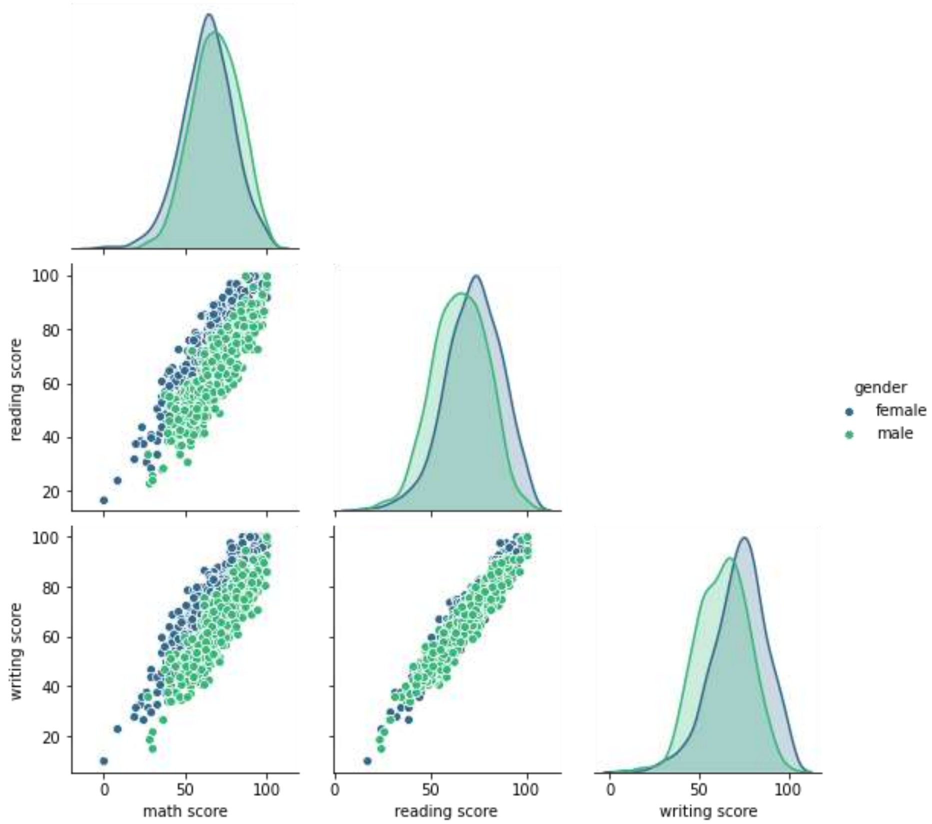
Seaborn

- pairplot()



Seaborn

- pairplot()



Seaborn

- Let’s code out these comparison plots in the next lecture!

Comparison Plots

Part Two: Coding the Plots

Seaborn Grids

Seaborn

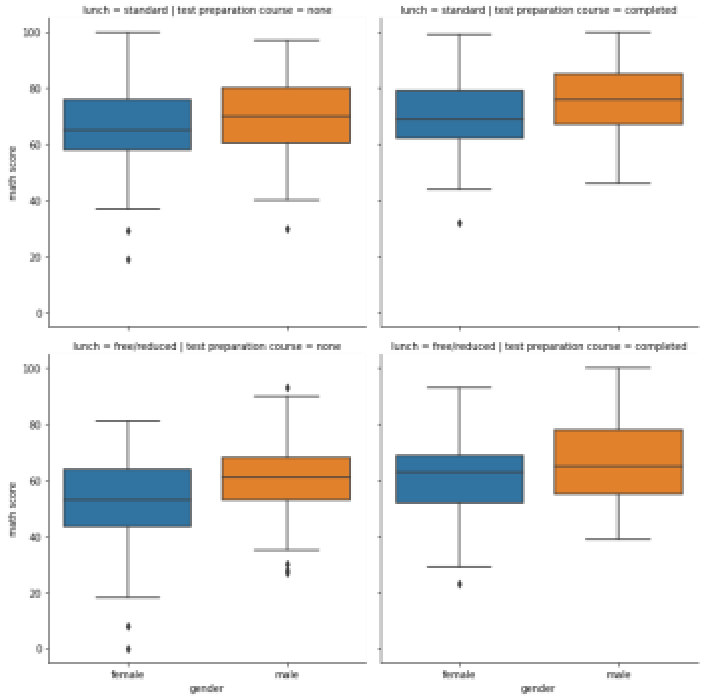
* Seaborn grid calls use Matplotlib subplots() to automatically create a grid based off a categorical column.
* Instead of passing in a specific number of cols or rows for the subplots, we can simply pass in the name of the column and seaborn will automatically map the subplots grid.

Seaborn

- Many of seaborn's built-in plot calls are running on top of this grid system.
- Directly calling the grid system allows users to heavily customize plots.

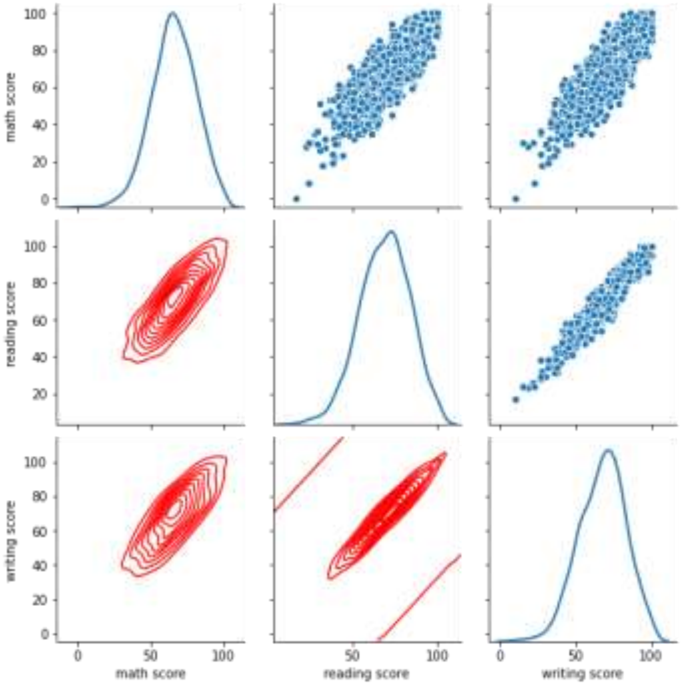
Seaborn

- Creating subplots based on grids:



Seaborn

- Map plots based on pairplot() grid:



Seaborn

- This is best understood through code, so let’s jump to the notebook!

Matrix Plots

Seaborn

- Matrix plots are the visual equivalent of displaying a pivot table.
- The matrix plot displays all the data passed in, visualizing all the numeric values in a DataFrame.
- Note!
    - Not every DataFrame is a valid choice for a matrix plot such as a heatmap.

Seaborn

- The two main matrix plot types are:
    - heatmap()
        - Visually displays the distribution of cell values with a color mapping.
    - clustermap()
        - Same visual as heatmap, but first conducts hierarchical clustering to reorganize data into groups.

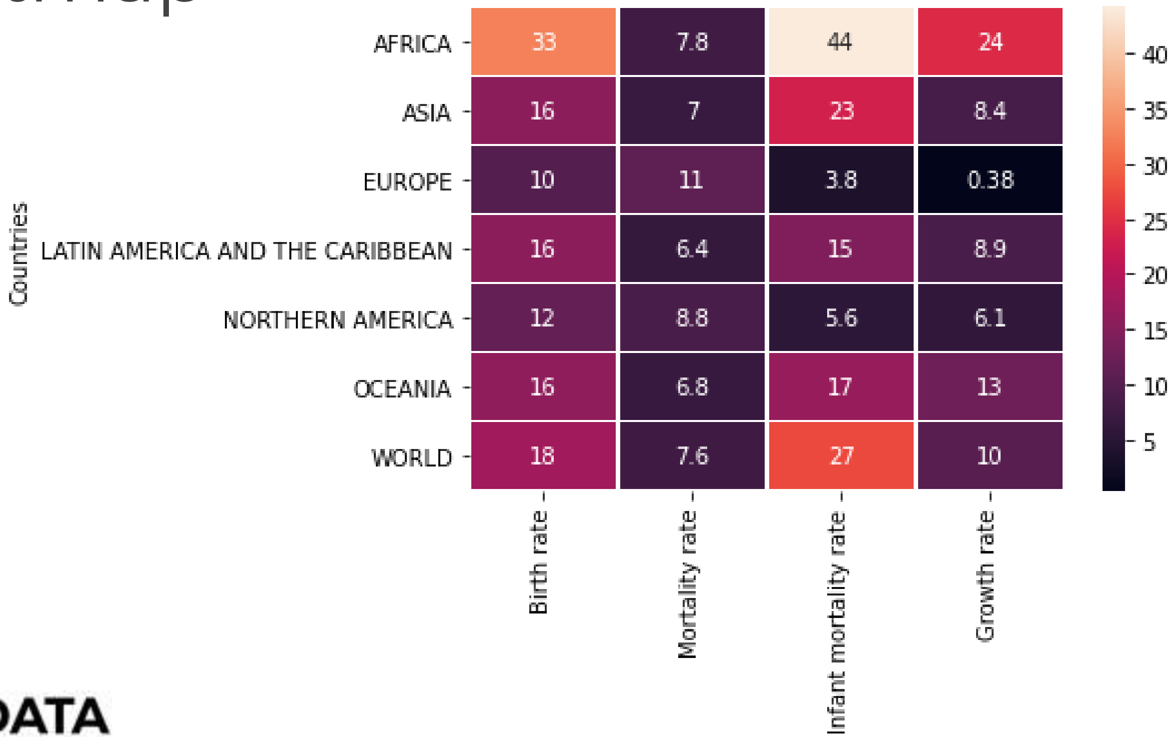
Seaborn

- Heatmap

Countries	Birth rate	Mortality rate	Life expectancy	Infant mortality rate	Growth rate
AFRICA	32.577	7.837	63.472	44.215	24.40
ASIA	15.796	7.030	73.787	23.185	8.44
EUROPE	10.118	11.163	78.740	3.750	0.38
LATIN AMERICA AND THE CARIBBEAN	15.886	6.444	75.649	14.570	8.89
NORTHERN AMERICA	11.780	8.833	79.269	5.563	6.11
OCEANIA	16.235	6.788	78.880	16.939	12.79
WORLD	17.963	7.601	72.766	27.492	10.36

Seaborn

- Heatmap

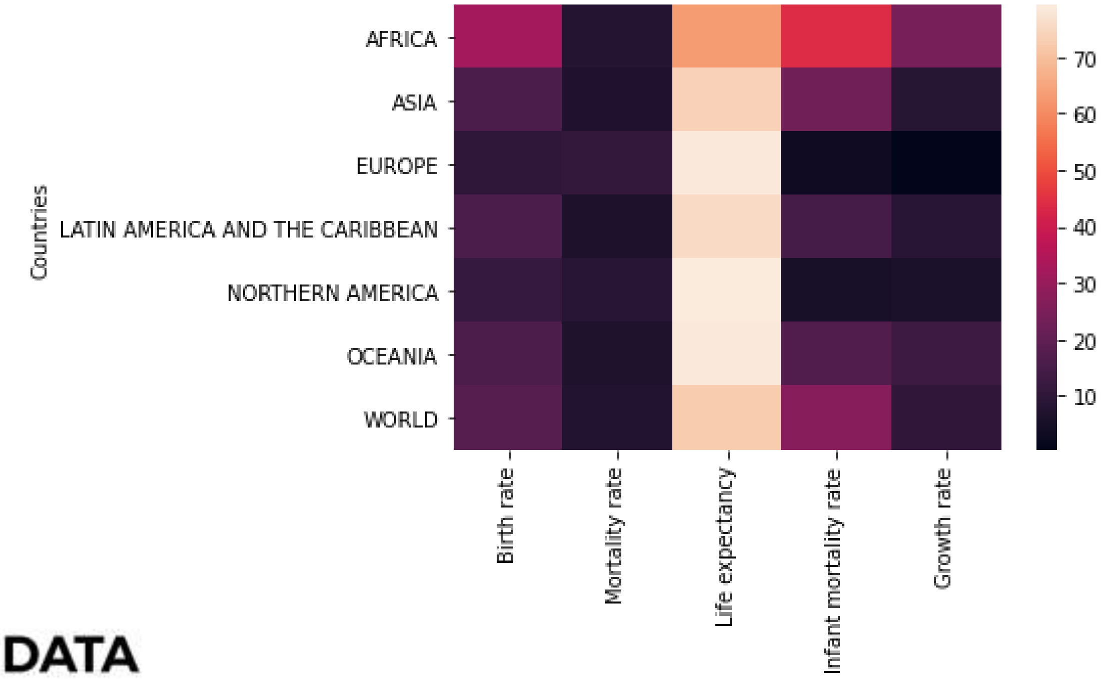


Seaborn

* Note that a heatmap should ideally have all cells be in the same units, so the color mapping makes sense across the entire DataFrame.
* In this particular case, all values were “rates” of percentage growth or change were in the heatmap.

Seaborn

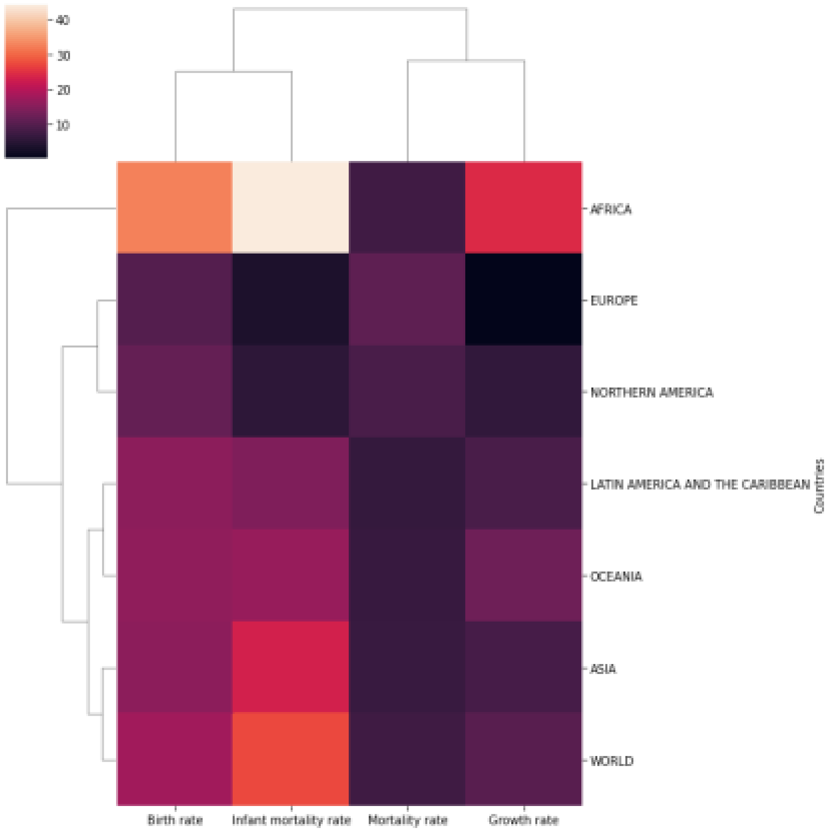
- If we included age:



Seaborn

- Seaborn also comes with the ability to automatically cluster similar groupings.
- Later on we will discuss how this clustering is done when we learn about Machine Learning clustering techniques.

Seaborn



PIERIAN DATA

Seaborn

- Let’s get to coding out these matrix plots!

Seaborn Exercises

PIERIAN DATA

Seaborn

- Main goal of seaborn is to be able to use its simpler syntax to quickly create informative plots.
- In general its difficult to test on seaborn skills since most plots are simply passing in the data and choosing x and y.

Seaborn

- For these exercises we’ve inserted jpg images of seaborn plots we want you to replicate.
- Don’t worry if you don’t get coloring or dimensions exactly the same as ours, focus on the general plots and relationships visualized.

Seaborn

- Read the plot descriptions **carefully**!
- Most of these plots have filtering and adjustments with pandas on the DataFrame **before** being passed into the seaborn call.

Seaborn Exercises Solutions