Plotting and Curve Fitting in Python

And why we need to be able to see and show our data

Quantitative Methods Workshop 2025
January 3rd 2025
9:00am – 11:00am
Georgina Woo

The Plan!!

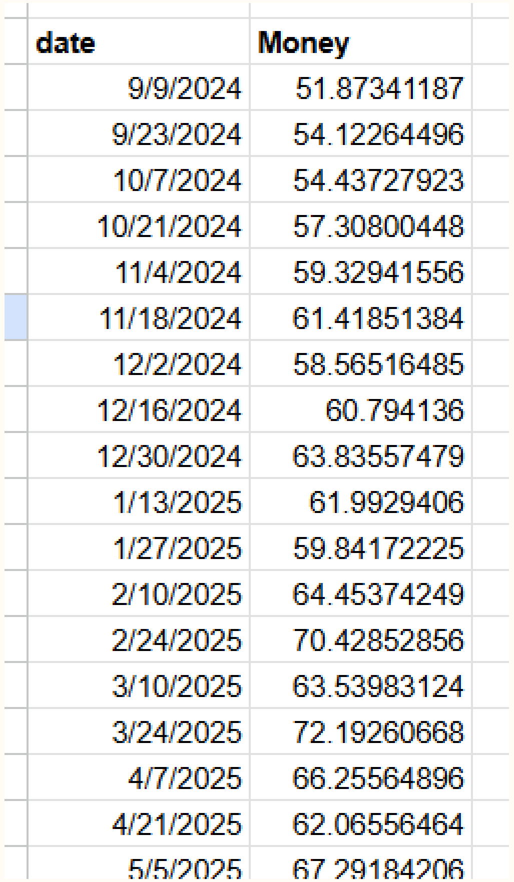
10am – 12pm

5m	Why Visualization Matters
10m	Matplotlib Basics
10m	Seaborn Basics
20m	Curve Fitting with Python
1h	Programming time!
10m	The final kahoot

Why Visualization Matters

A picture is worth a thousand words…

We see a thousand numbers…and not much else

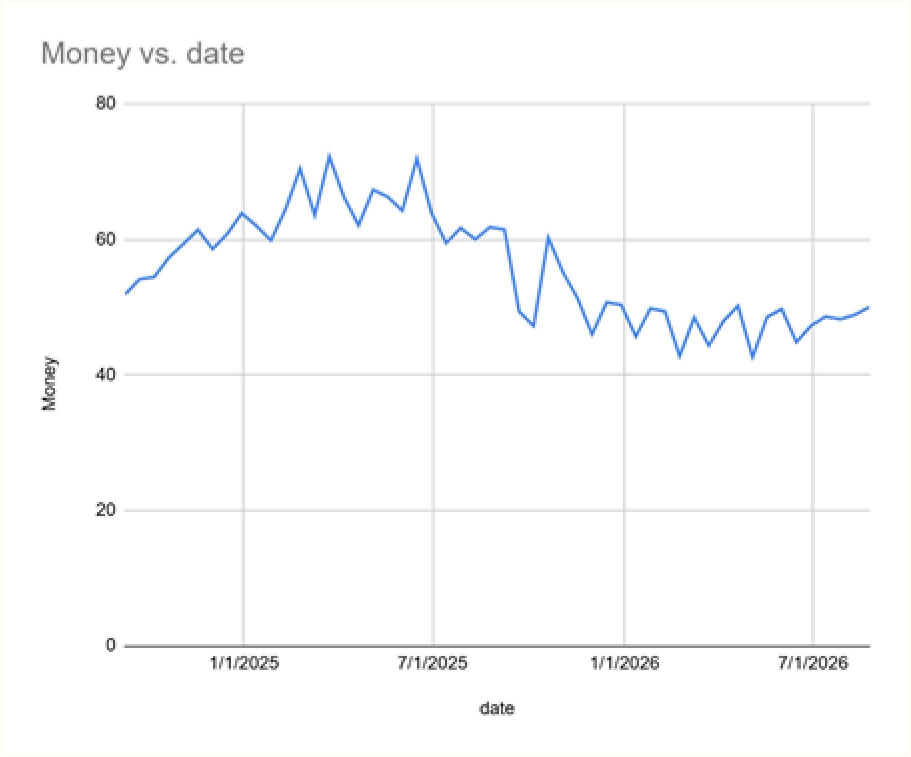


date	Money
9/9/2024	51.87341187
9/23/2024	54.12264496
10/7/2024	54.43727923
10/21/2024	57.30800448
11/4/2024	59.32941556
11/18/2024	61.41851384
12/2/2024	58.56516485
12/16/2024	60.794136
12/30/2024	63.83557479
1/13/2025	61.9929406
1/27/2025	59.84172225
2/10/2025	64.45374249
2/24/2025	70.42852856
3/10/2025	63.53983124
3/24/2025	72.19260668
4/7/2025	66.25564896
4/21/2025	62.06556464
5/5/2025	67.29184206

Why Visualization Matters

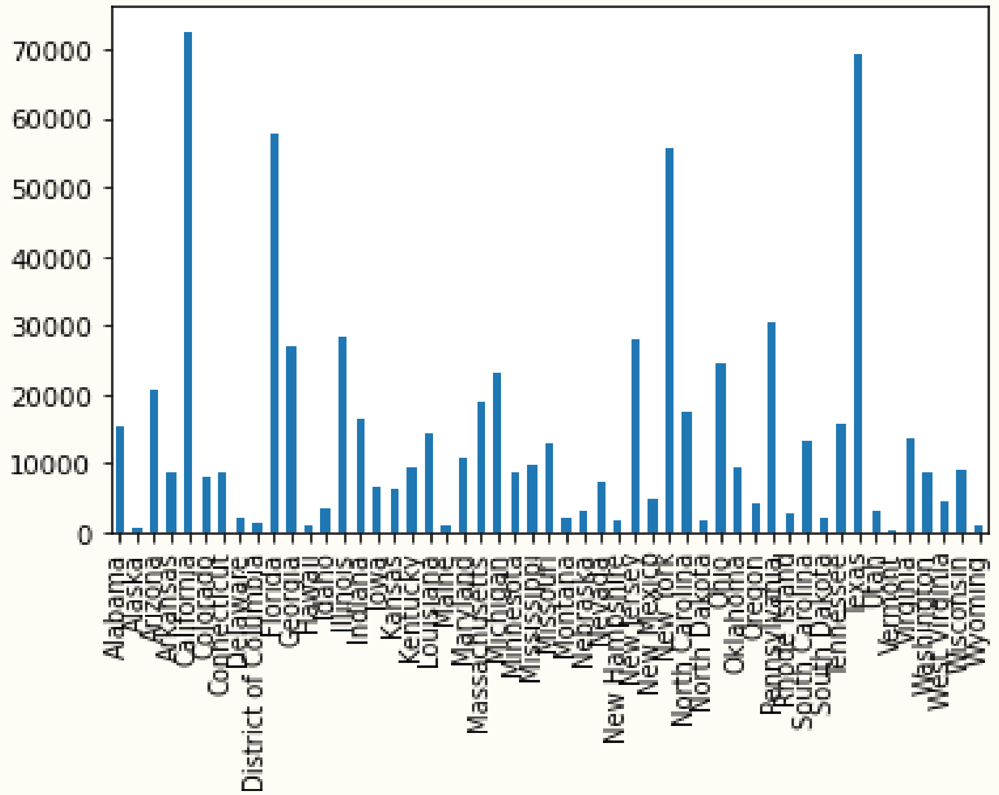
A picture is worth a thousand words…

Same information, but now we see the story!
(A sad one)

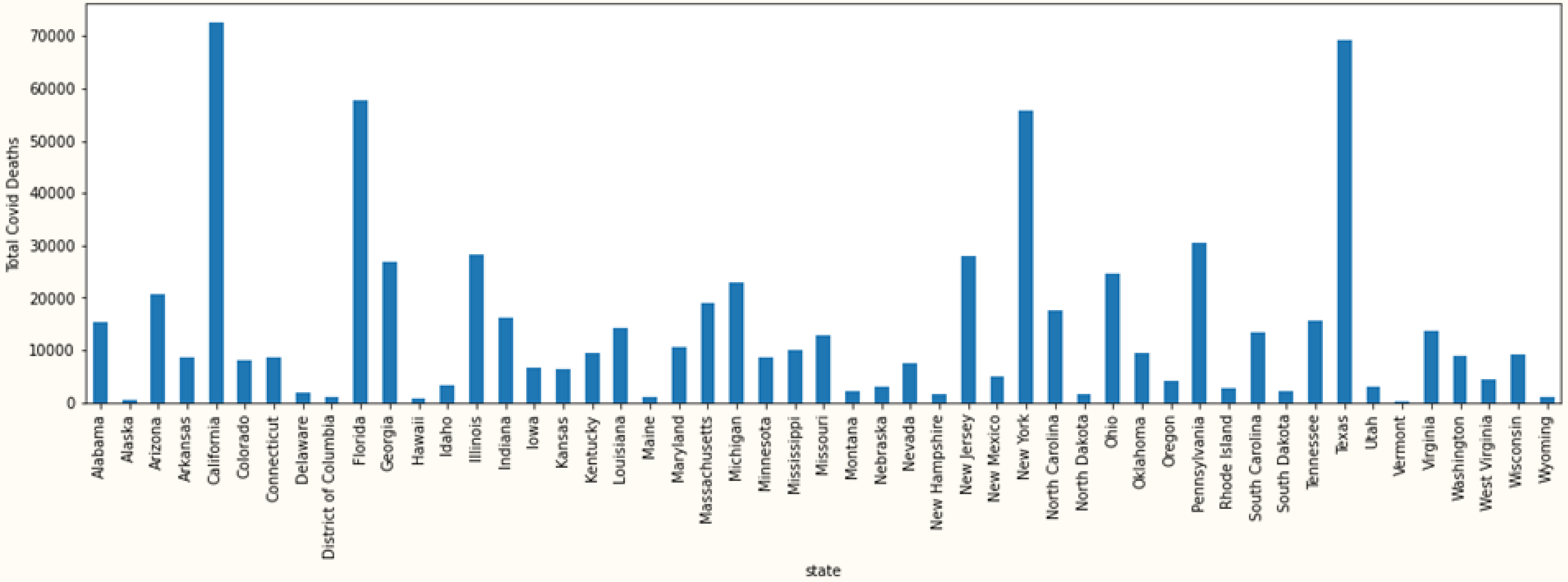


“Where was COVID the worst in the US?”

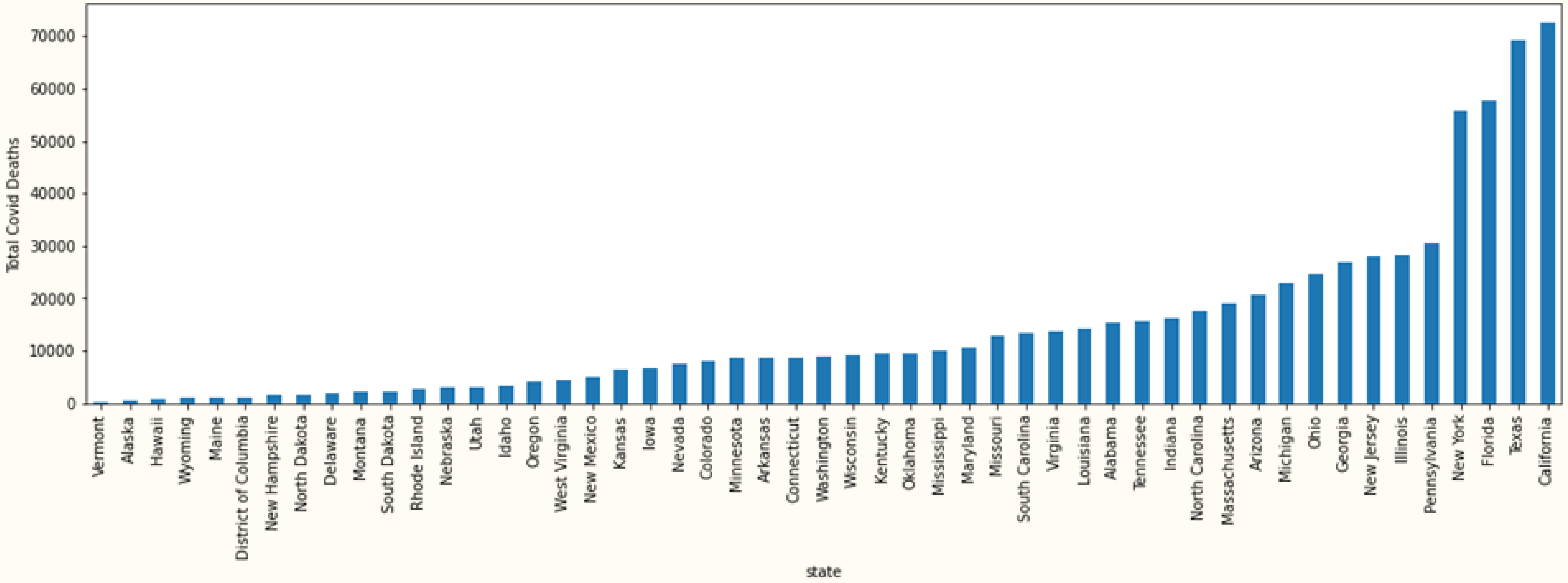
Why is this not giving?



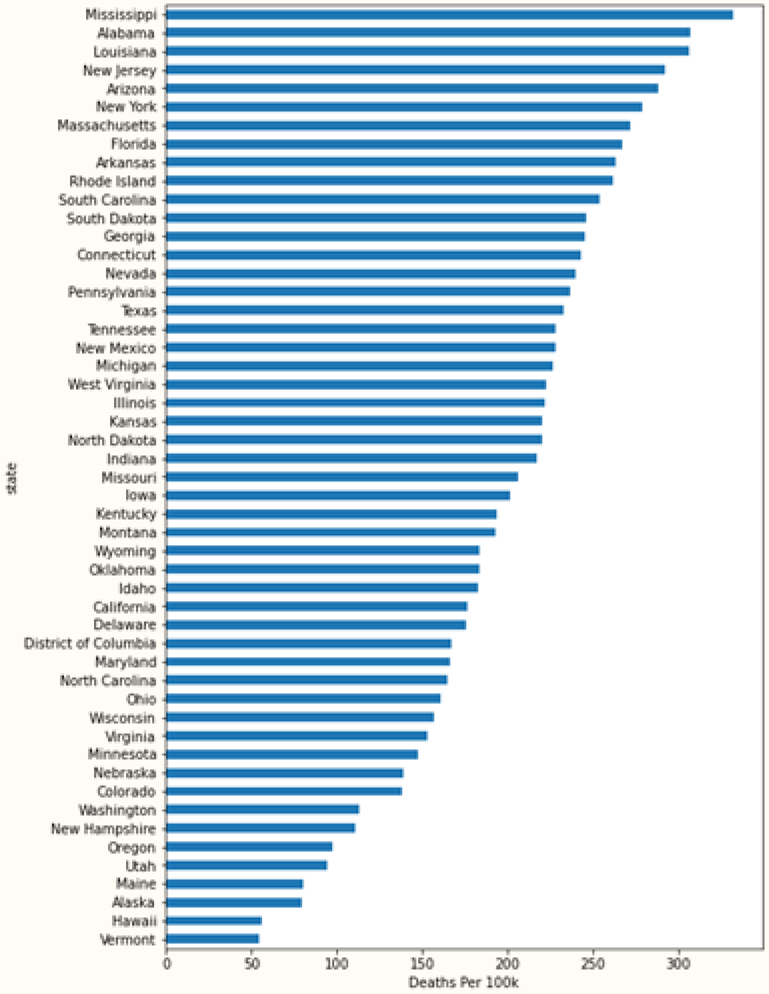
It’s readable, but still troublesome



Sorting helps, but what's the story?

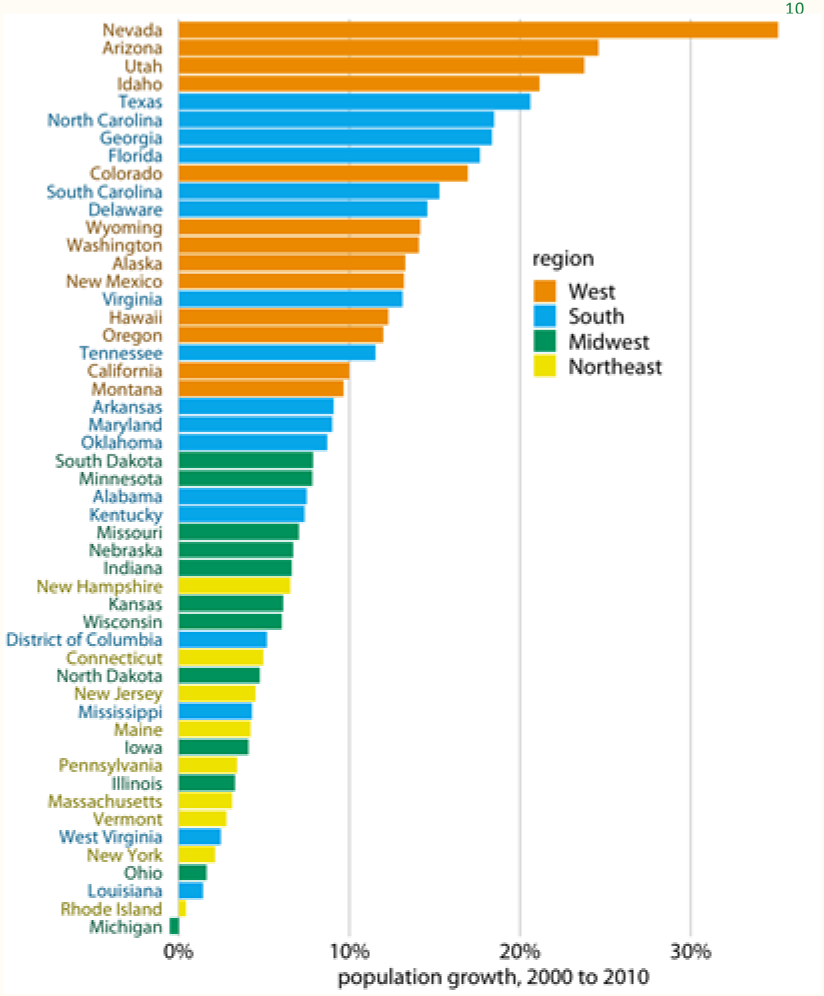


We love not having to tilt our heads to read graphs!



Use color to draw focus!

Just don’t over do it.



When to use what chart

Pie charts

Line charts

Scatter Plots

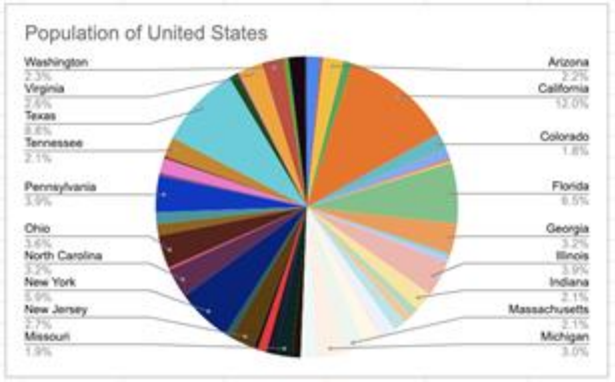
Bar Graphs

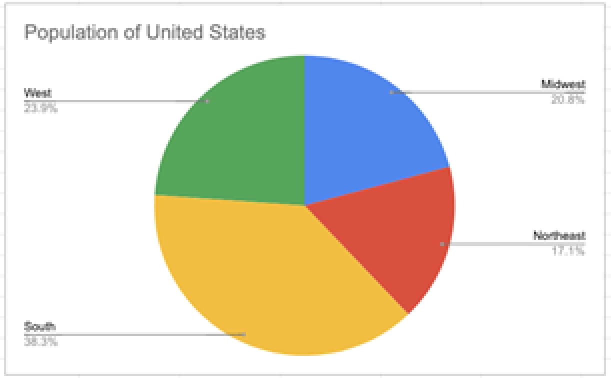
Histograms

Heatmaps

Pie Charts

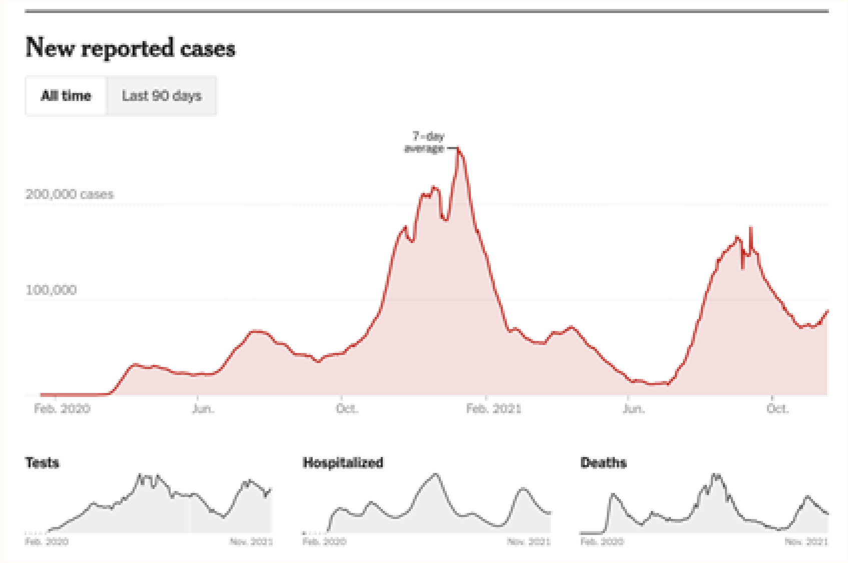
- When you show relative proportions and ***percentages of a whole*** dataset.
- When comparing the effect of ONE factor on different categories.
- Have no more than 6 categories.





Line Charts

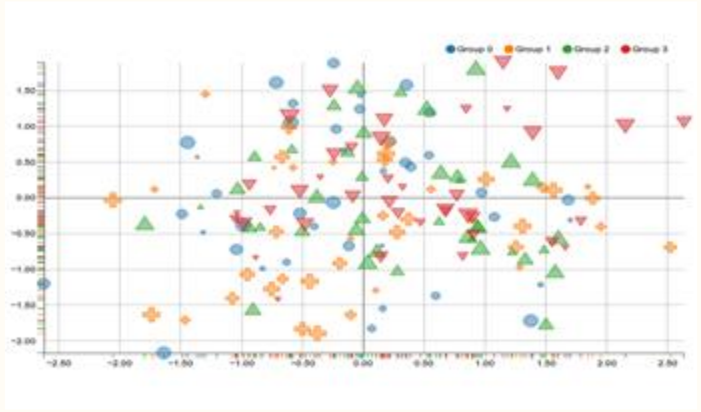
- When you have a continuous dataset that changes over time.
- When your dataset is too big for a bar chart.
- When you want to visualize trends instead of exact values.



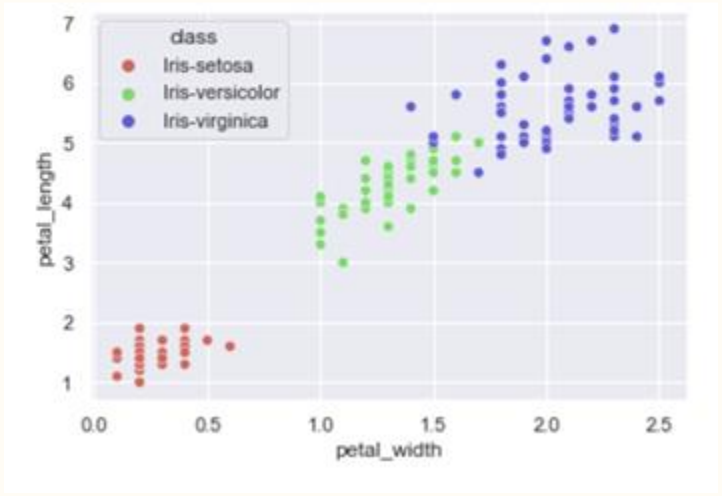
Scatter Plots

- Show the relationship of two variables.
- Show correlation and clustering in big datasets.
- When ordering is not important (ie; not time series data)

✗



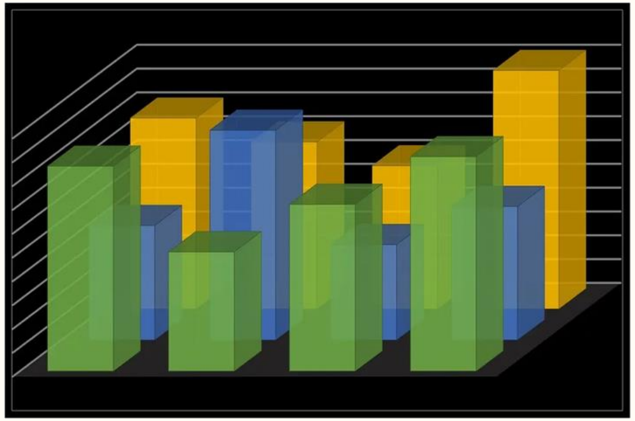
✓



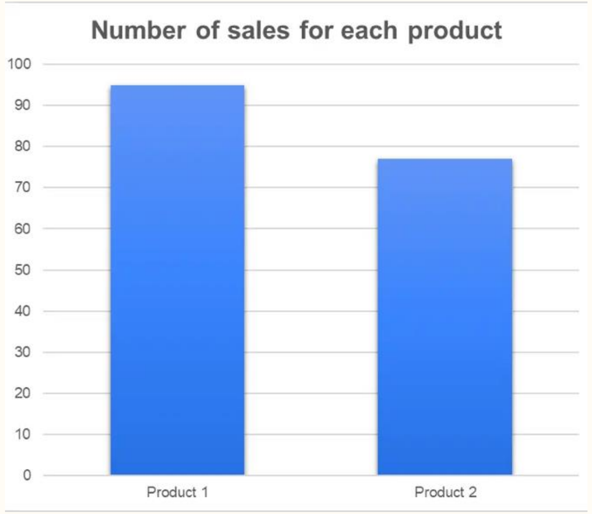
Bar Graphs

- Compare quantities across categories.
- Show trends or comparisons over time (use clustered or stacked bars for multi-variable data).
- Categories are discrete (e.g., "Apple vs. Banana sales").

✗



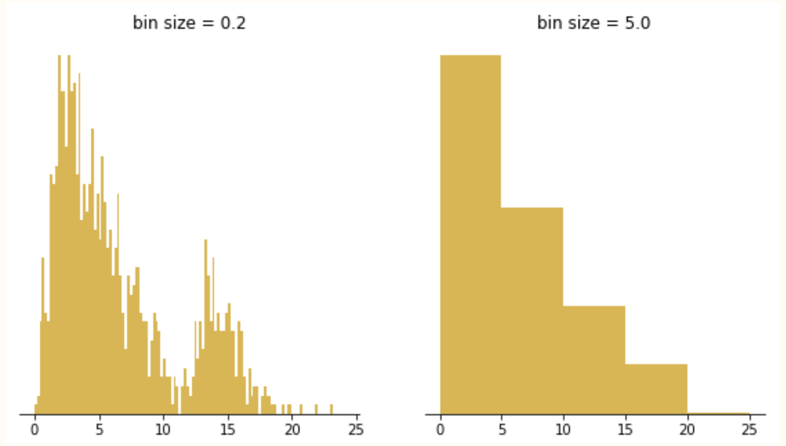
✓



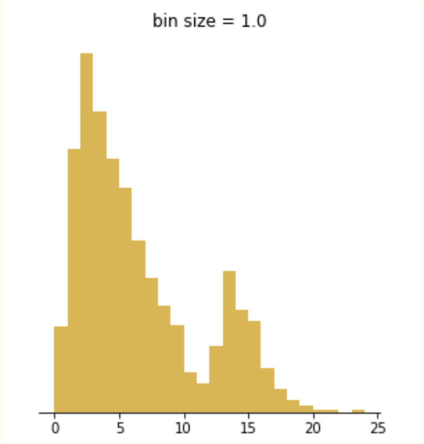
Histograms

- Show the distribution of a continuous variable (e.g., "Exam scores").
- Identify the shape of the data (e.g. normal, skewed).
- Use for large datasets, with values grouped into "bins."

✗



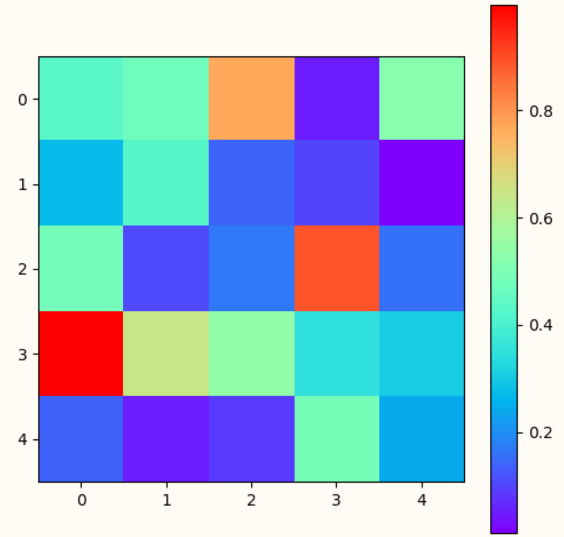
✓



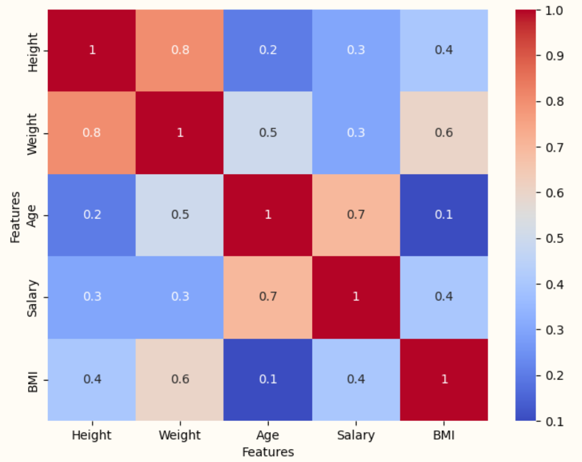
Heatmaps

- Visualize relationships between two variables in matrix/tabular form.
- Highlight areas of high or low intensity (e.g. "Correlation matrix", "Population density").

✗



✓



	Height	Weight	Age	Salary	BMI
Height	1	0.8	0.2	0.3	0.4
Weight	0.8	1	0.5	0.3	0.6
Age	0.2	0.5	1	0.7	0.1
Salary	0.3	0.3	0.7	1	0.4
BMI	0.4	0.6	0.1	0.4	1

The Do’s

- Use graphs to tell a message, story, or insight.
- Use well labeled axes and titles. (axes that start at 0)
- Use appropriate color that helps convey your message
- Use the least amount of ink to get your message across.
- Use the appropriate graph for what you’re trying to communicate.

The Do not’s

- Don’t use visualizations that don’t deliver a message, story, or insight.
- Don’t (un)intentionally misrepresent data.
- Don’t use ink that doesn’t add to the story

Python Plotting Tools

Matplotlib

Seaborn

What is Matplotlib?

- A widely-used Python library for creating static, animated, and interactive visualizations.

Basic syntax:

import matplotlib.pyplot as plt

plt.plot() for line plots.

plt.scatter() for scatter plots.

plt.bar() for bar charts.

What is Matplotlib?

- A widely-used Python library for creating static, animated, and interactive visualizations.

Basic syntax:

import matplotlib.pyplot ~~as plt~~

matplotlib.pyplot.plot() for line plots.

matplotlib.pyplot.scatter() for scatter plots.

matplotlib.pyplot.bar() for bar charts.

What is Matplotlib?

- A widely-used Python library for creating static, animated, and interactive visualizations.

Basic syntax:

```
import matplotlib.pyplot as plt
```

plt.plot() for line plots.

plt.scatter() for scatter plots.

plt.bar() for bar charts.

Try it in Colab!

- In a new cell:
- import matplotlib.pyplot as plt
- Set up 2 lists of 5 heights and weights
- Make a line plot with the heights, weights, using the color blue
- Title the plot and the x and y labels
- Show the graph

Syntax:

```
plt.plot(x, y)

plt.title(“Plot title”)

plt.xlabel(“x-axis”)

plt.ylabel(“y-axis”)

plt.show()
```

Try it in Colab!

```
plt.bar(x, y, color = “teal”)

plt.title(“Plot title”)

plt.xlabel(“x-axis”)

plt.ylabel(“y-axis”)

plt.show()

plt.savefig(“barplot.png”, transparent = True)
```

- In a new cell:
- Set up 2 lists of 5 cities and rainfall levels
- Make a bar plot with the heights, weights, using the color teal, alpha 0.7
- Title the plot and the x and y labels
- Show the graph
- Save the graph
- Save the graph with a transparent background

What is Seaborn?

- Built on Matplotlib, offering easier syntax and more aesthetic default styles.

Basic syntax:

import seaborn as sns

sns.lineplot() for line plots.

sns.scatterplot() for scatter plots.

sns.barplot() for bar charts.

sns.heatmap() for heat maps.

Try it in Colab!

Syntax:

```
df = pd.DataFrame({”Col1”:[1,2,3], “Col2”:[10,20,30]’})

sns.lineplot(x=“Col1”, y=”Col2”, data=df, marker=“o”, color=“green)
```

- In a new cell:
- import seaborn and pandas
- Set up a pandas dataframe storing COVID cases over 5 days
- Make a line plot with Days and Cases, using the color green
- Title the plot and the x and y labels using plt
- Show the graph

Try it in Colab!

```
sns.histplot(ages, bins=5, kde=True, color = “purple”)

plt.title(“Plot title”)

plt.xlabel(“x-axis”)

plt.ylabel(“y-axis”)

plt.show()

plt.savefig(“barplot.png”, transparent = True)
```

- In a new cell:
- Set up a list of 10 ages
- Make a seaborn histogram with the ages, 5 bins, kde=True, and the color purple
- Title the plot and the x and y labels with plt
- Show the graph

What is Curve Fitting?

What is Curve Fitting?

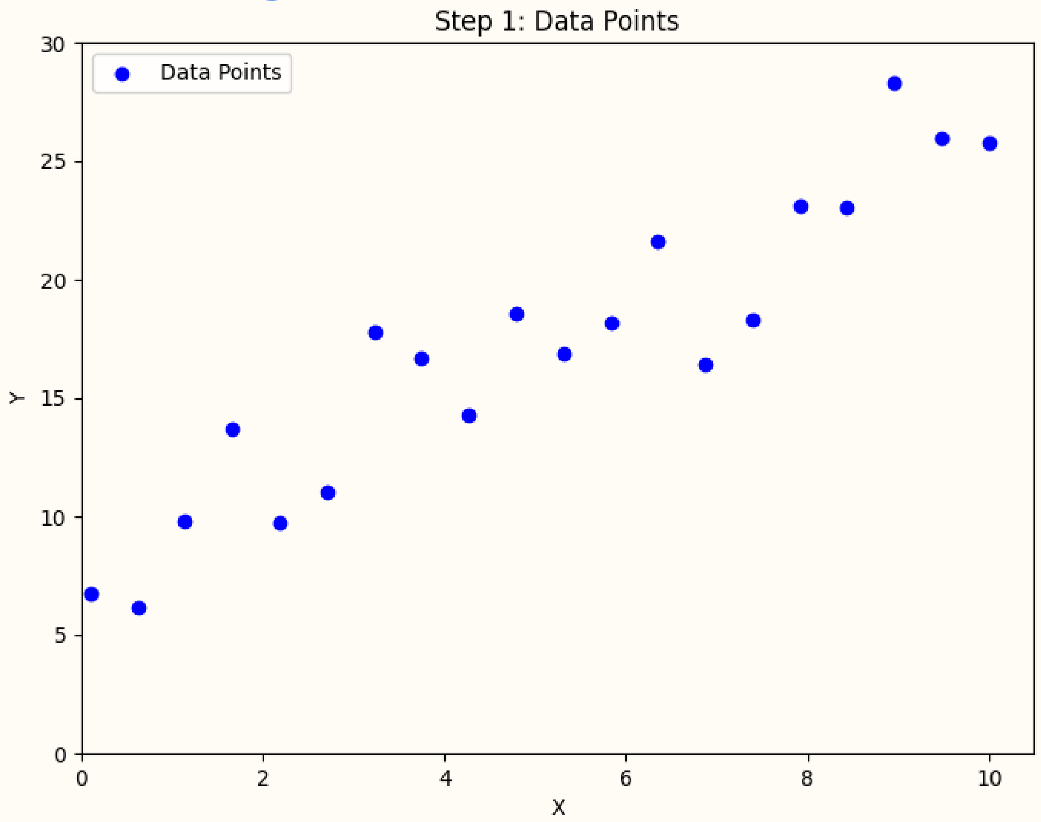
- The process of finding a mathematical function that best fits data points.
- A linear growth function can be represented by $y = mx+b$
- To properly analyze and predict trends, we can also consider other forms

Bacterial growth (exponential).

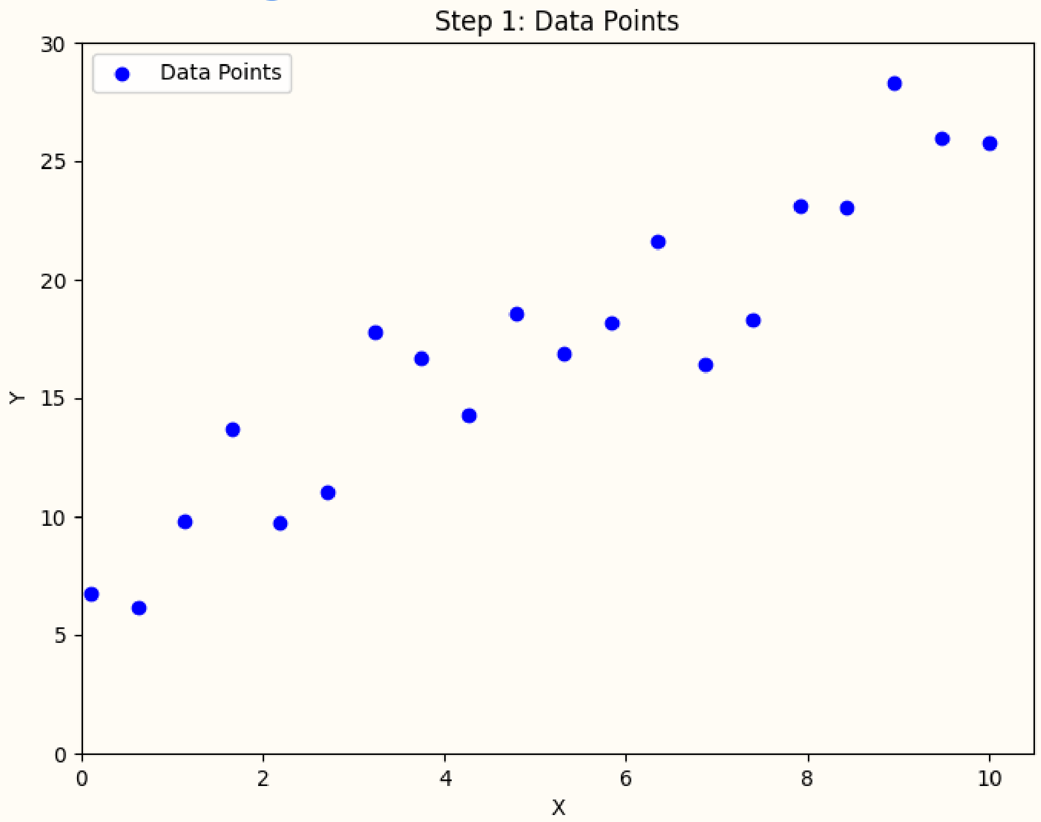
Housing prices (linear or polynomial).

Temperature trends over years (sinusoidal).

What is Curve Fitting?



What is Curve Fitting?

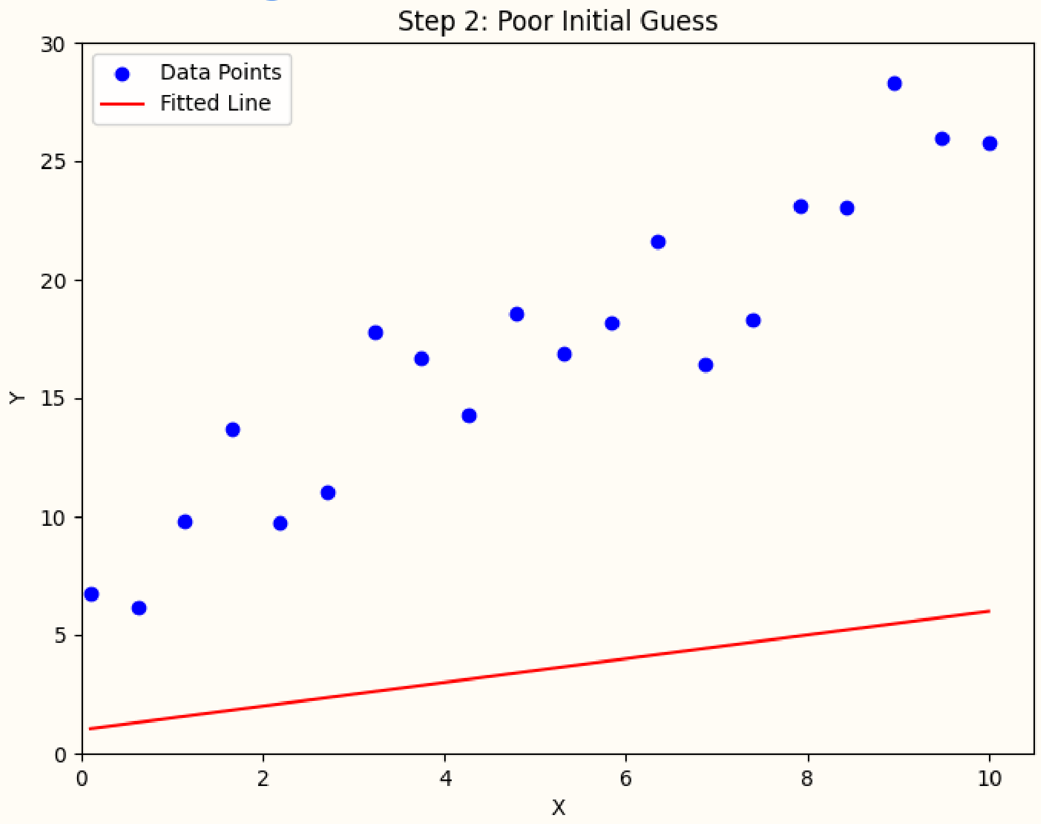


$y=mx+b$

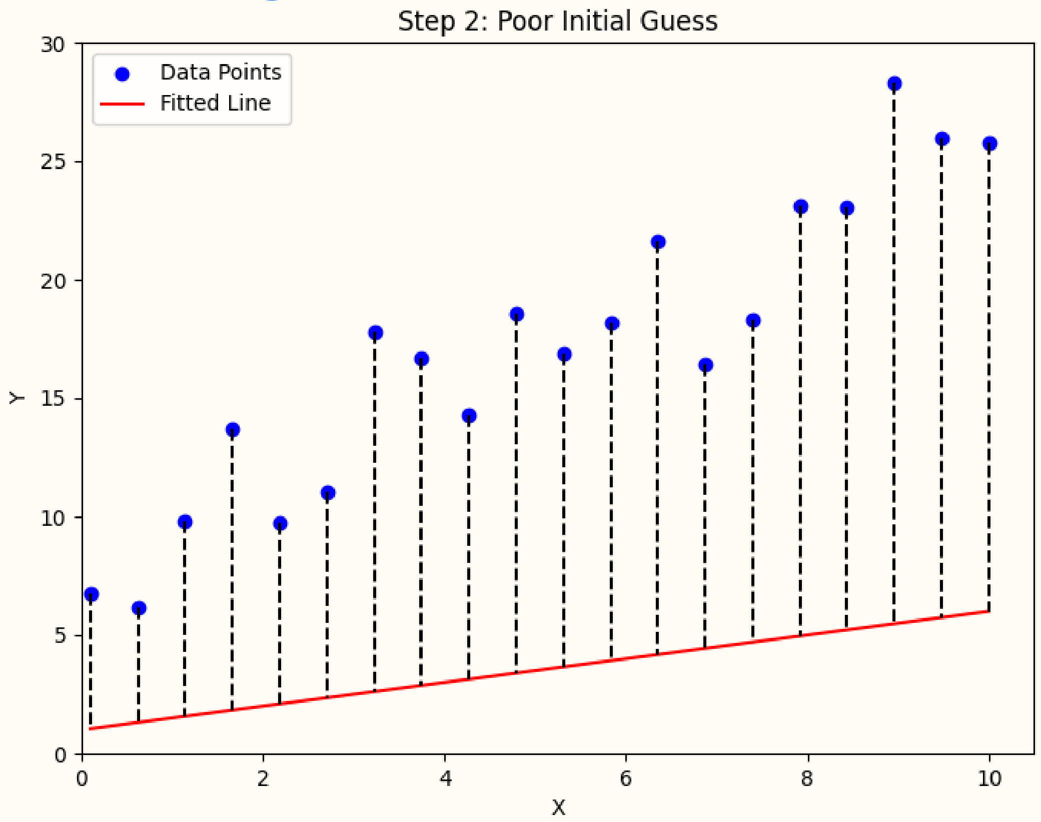
$m=?$

$b=?$

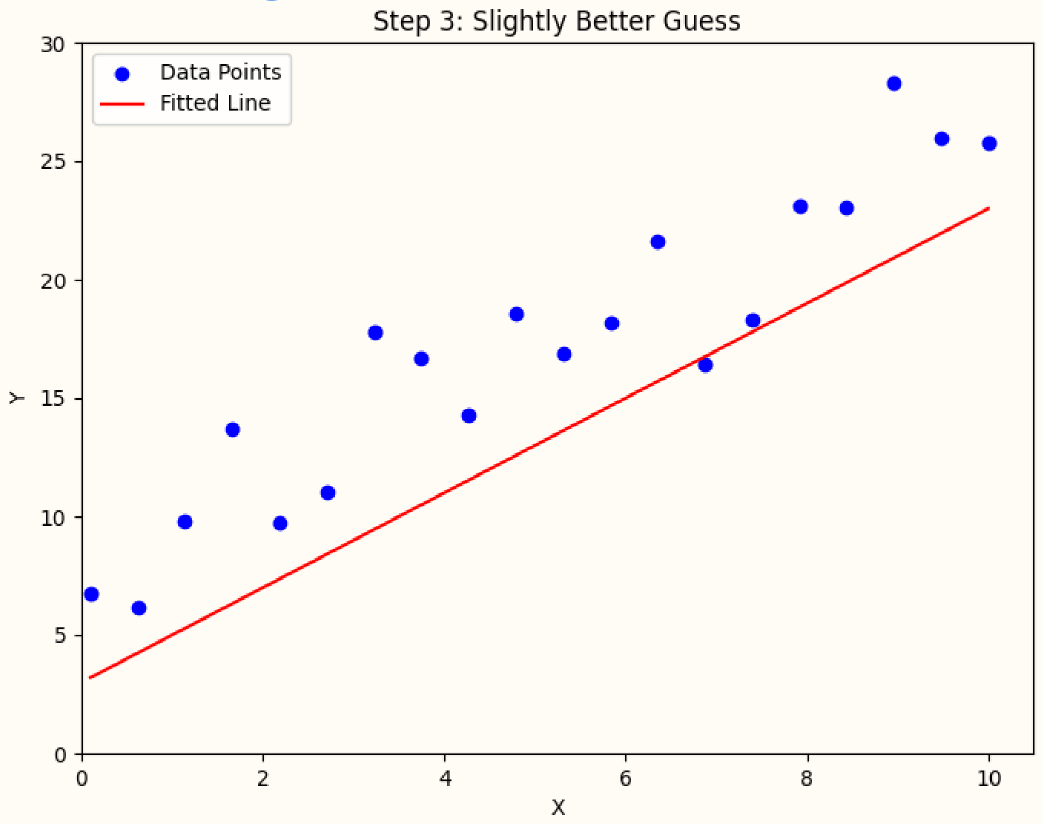
What is Curve Fitting?



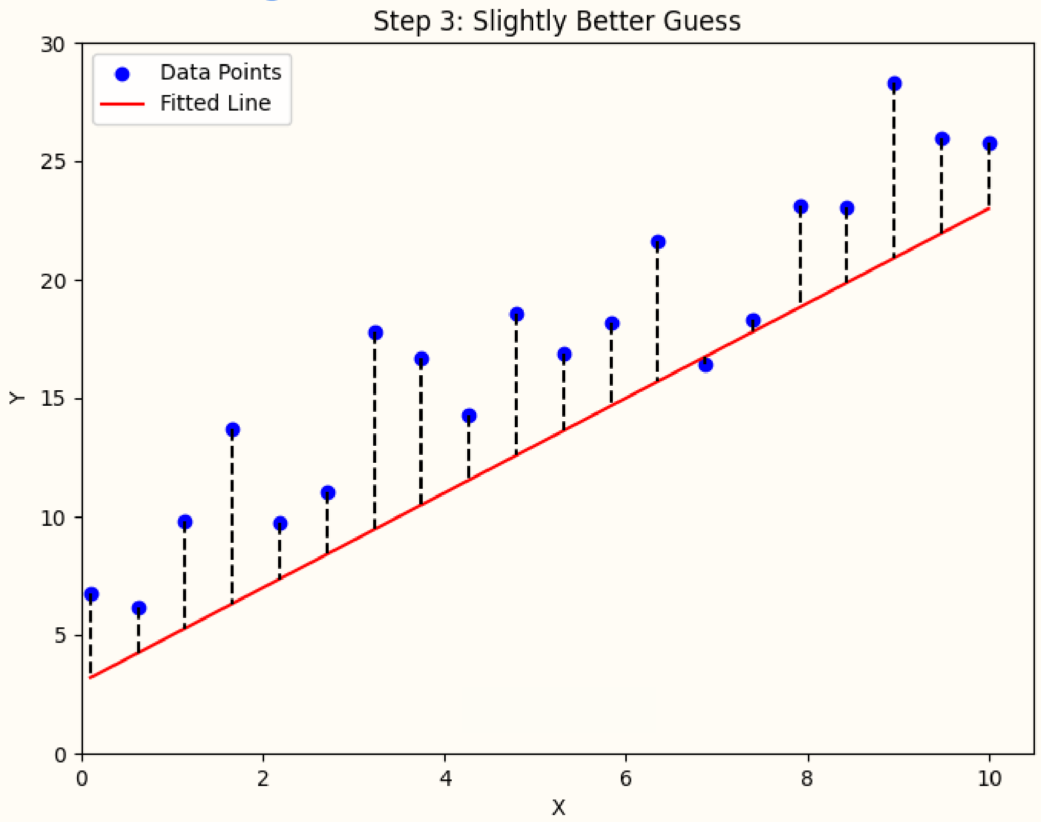
What is Curve Fitting?



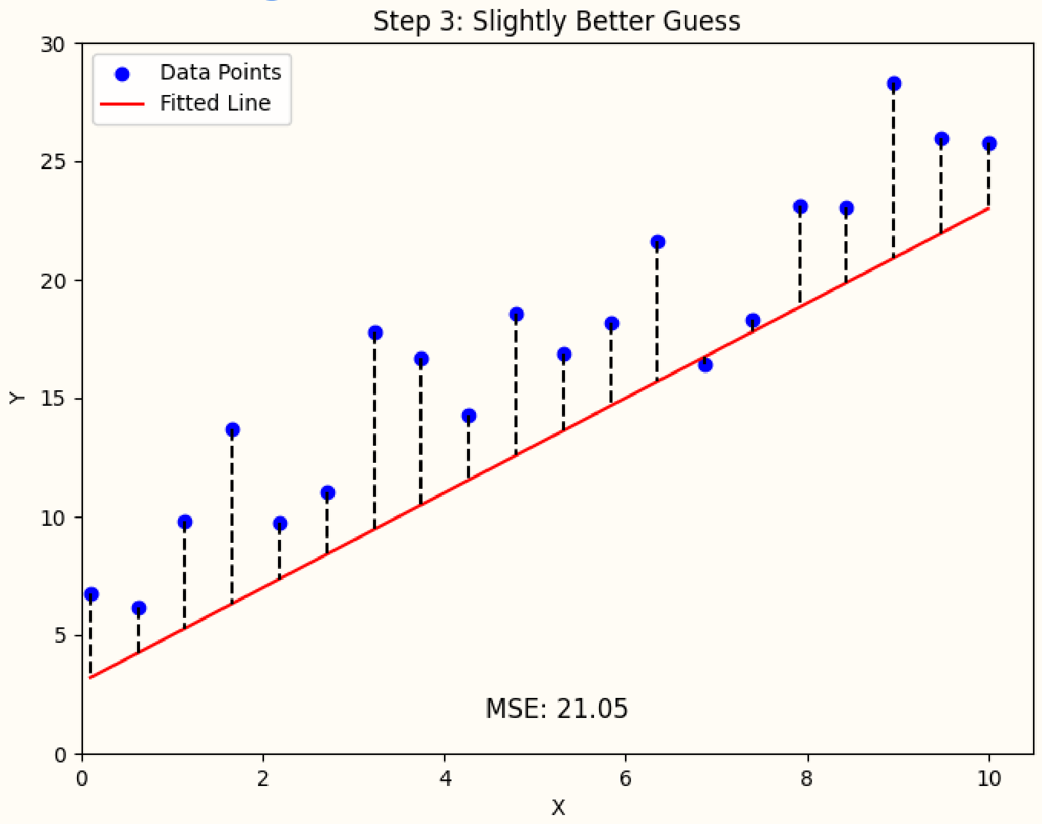
What is Curve Fitting?



What is Curve Fitting?

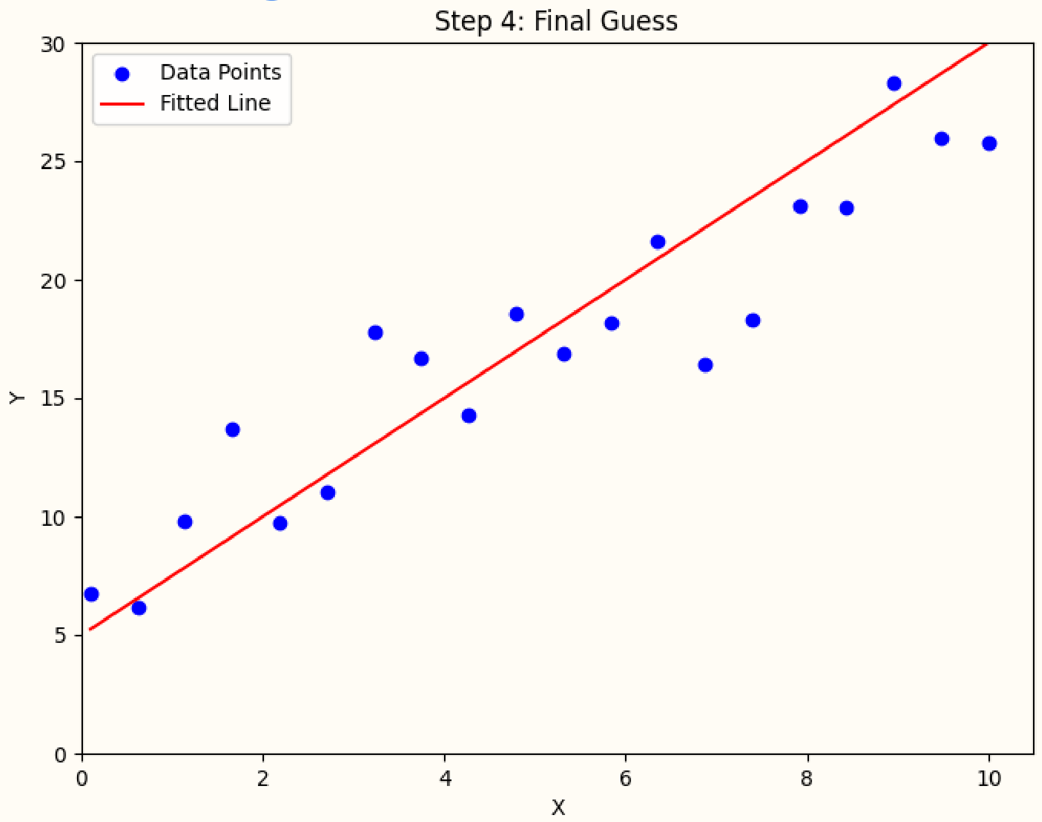


What is Curve Fitting?

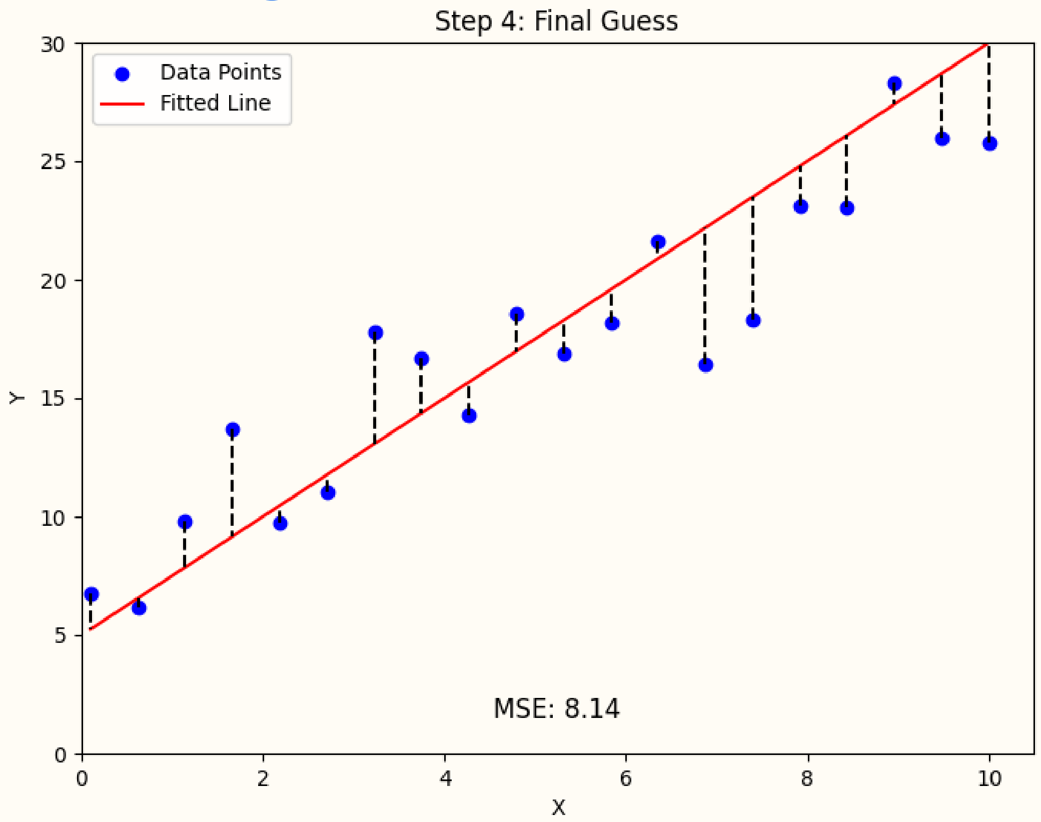


MSE: Mean Squared Error

What is Curve Fitting?



What is Curve Fitting?



Curve Fitting with Python

scipy.optimize

scipy.optimize

- Scipy is a scientific library in Python that provides tools for optimization, interpolation, and mathematical functions.
- scipy.optimize provides the curve_fit() function

How it works:

Define a function that represents the curve you expect in your data
Provide the data points (x, y) and let curve_fit optimize the parameters of the function to best fit the data.

```
from scipy.optimize import curve_fit

def model_function(x, a, b):
    return a * np.exp(b * x)

params = curve_fit(model_function, x_data, y_data)
```

Try it in Colab!

In a new cell:

- Import numpy, curve_fit from scipy.optimize, and matplotlib.pyplot
- Define a function y=mx+b

```
def linear_func(x, m, b):
    return m * x + b
```

- Create arrays x (independent variable) and y (dependent variable) with some random noise.

```
x = np.linspace(0, 10, 10)
y = 3 * x + 5 + np.random.normal(0, 2, len(x))
```

- Use curve_fit to estimate the parameters a and b that minimize the error between the defined function and the data.

```
params, _ = curve_fit(linear_func, x, y)
```

- Plot the original data and the fitted curve for comparison.

```
plt.scatter(x, y, label="Data Points", color="blue")
```

```
plt.plot(x, linear_func(x, est_m, est_b), label=f"Fitted Line")
```

The Final Kahoot

if you_have_questions:

AMA fr

else:

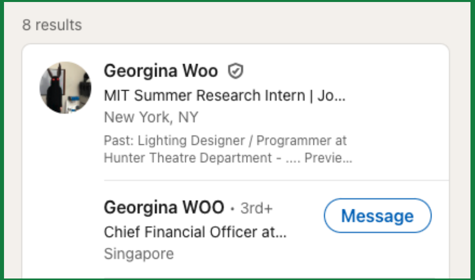
Programming Problem Set 2 is on MITx, have fun!

Thank you!

Keep in touch:

gwoo@mit.edu

I am also on LinkedIn (Georgina Woo)



8 results

Georgina Woo
MIT Summer Research Intern | Jo...
New York, NY
Past: Lighting Designer / Programmer at Hunter Theatre Department - .... Previe...

Georgina WOO · 3rd+
Chief Financial Officer at...
Singapore