Performance Evaluation

Regression Metrics

How do we decide if those predictions are any good?

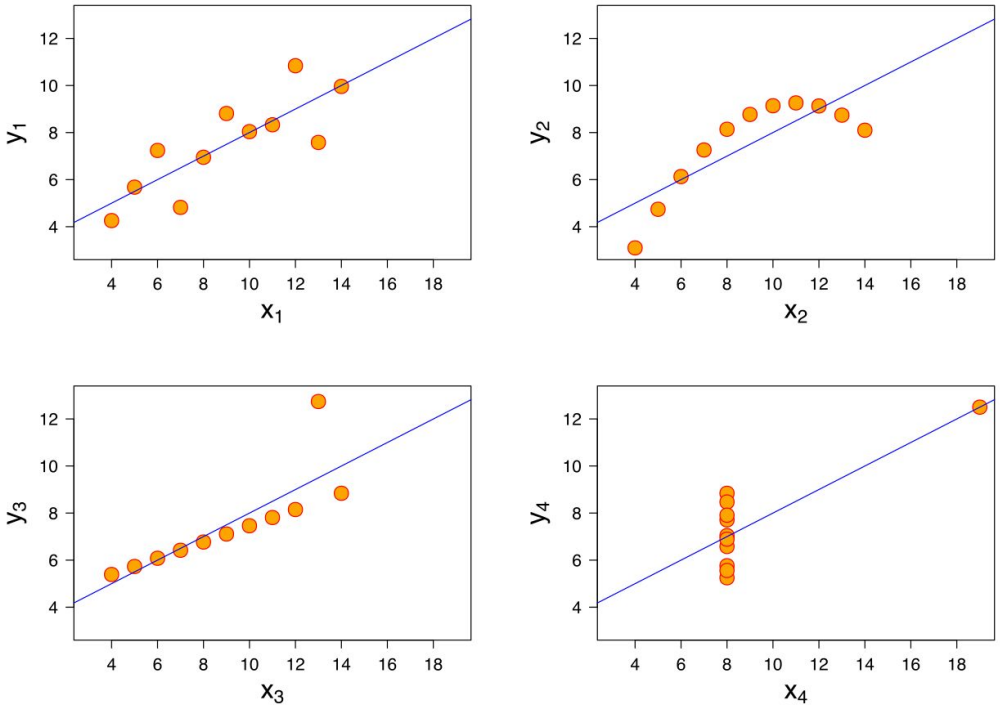
- Accuracy
- Recall.
- These sort of metrics aren’t useful for regression problems, we need metrics designed for **continuous** values!

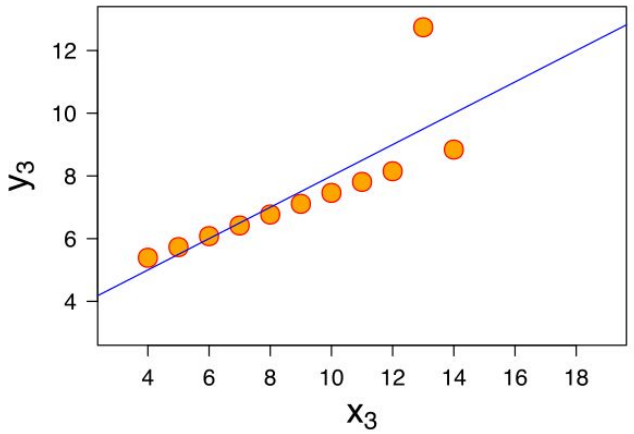
- Mean Absolute Error
- Mean Squared Error
- Root Mean Square Error

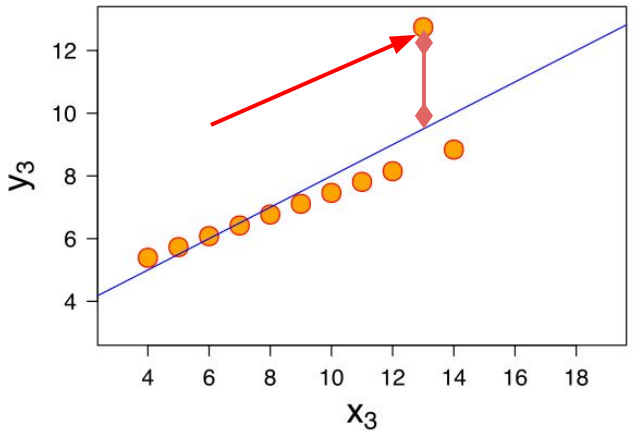
Mean Absolute Error (MAE)

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

MAE won't punish large errors however.







Mean Squared Error (MSE)

- Issue with MSE:
  - Different units than y.
  - It reports units of y squared!

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

Root Mean Square Error (RMSE)

- root of the mean of the squared errors.
- Most popular (has same units as y)

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}$$

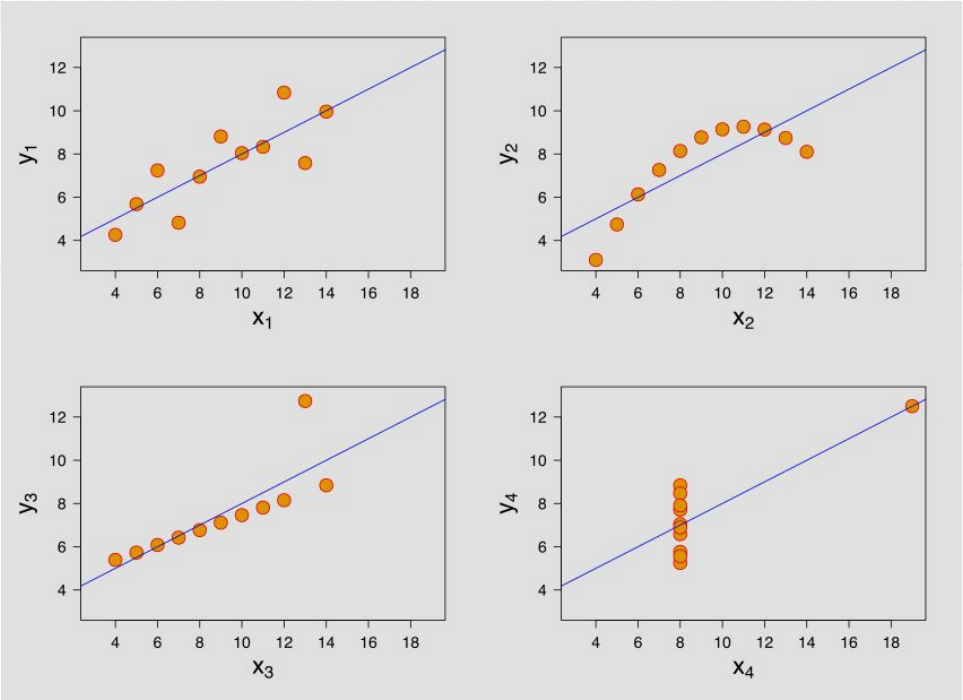
Compare your error metric to the average value of the label in your data set to try to get an intuition of its overall performance.

Domain knowledge also plays an important role here!

Context of importance is also necessary to consider.
We may create a model to predict how much medication to give, in which case small fluctuations in RMSE may actually be very significant.

Evaluating Residuals

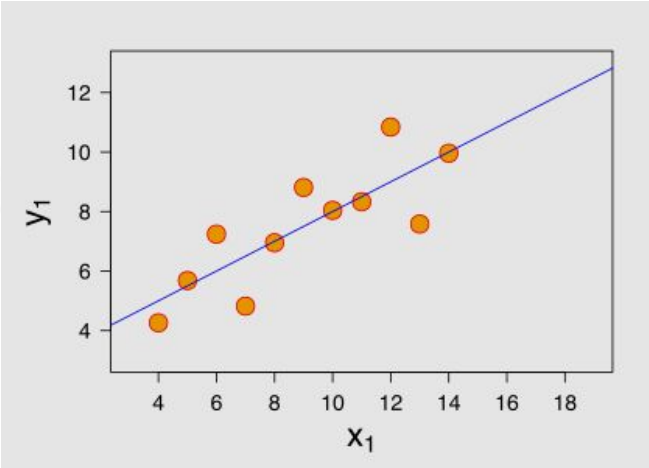
Anscombe's quartet:



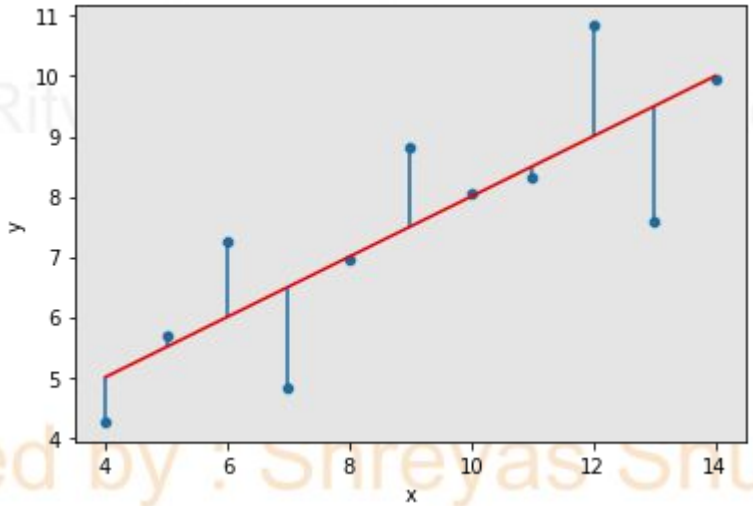
Clearly Linear Regression is not suitable

But how can we tell if we're dealing with more than one x feature?

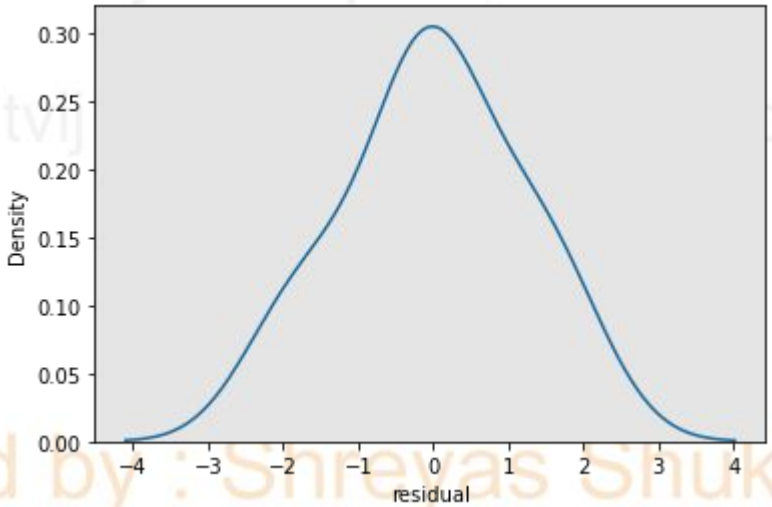
Consider an appropriate data set:



The residual errors.

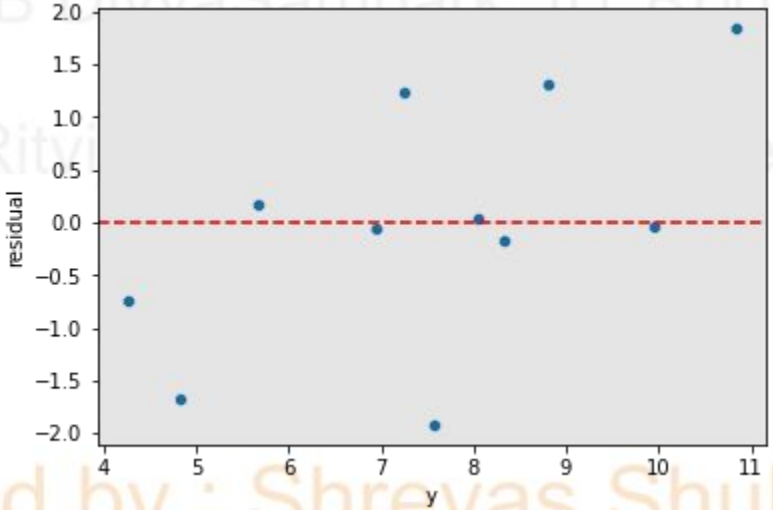


The residual errors should be random and close to a normal distribution.

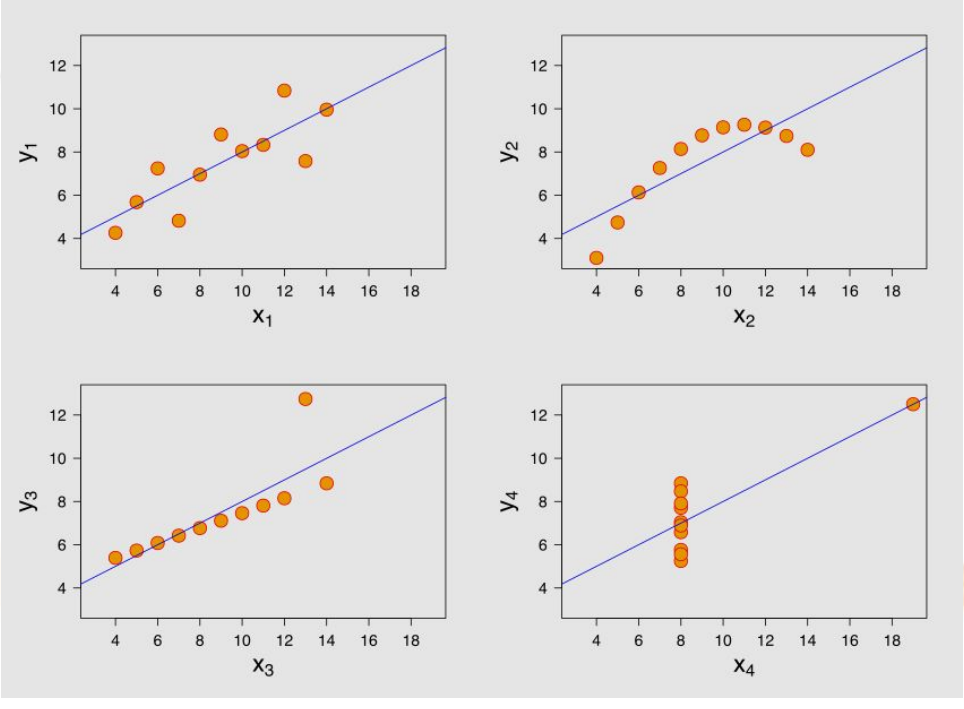


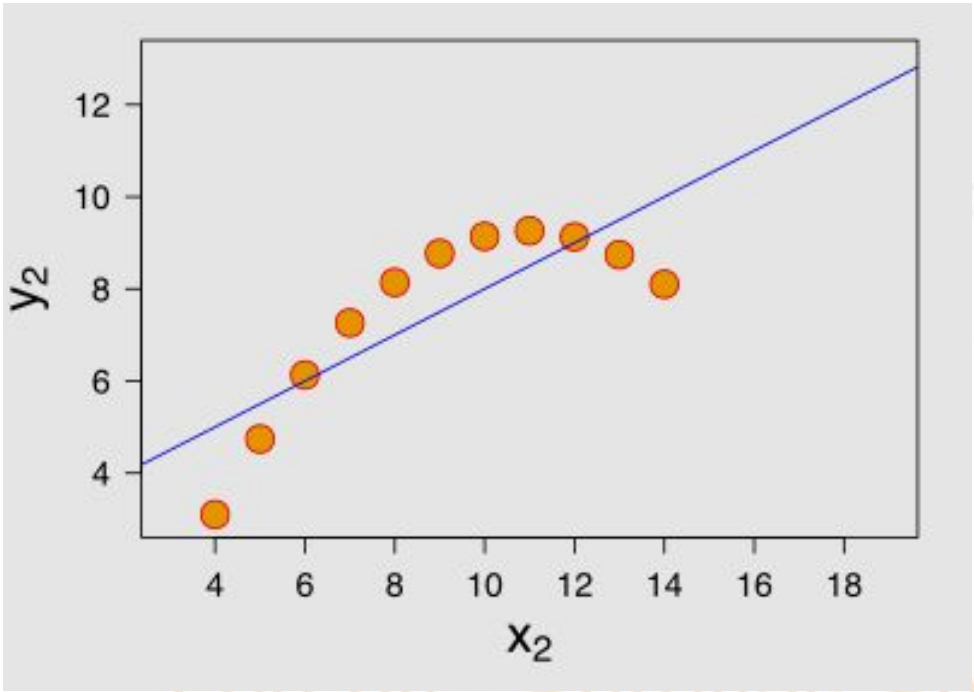
Residual plot shows residual error vs. true y value.

There should be no clear line or curve.



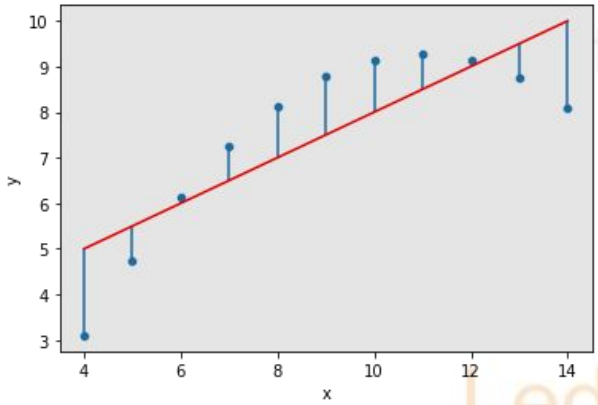
Let's see 2nd one

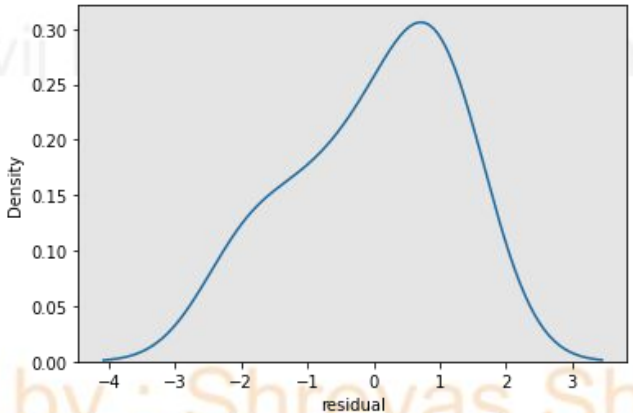


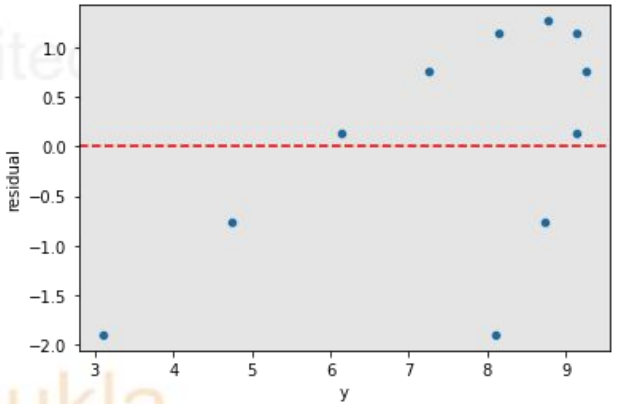


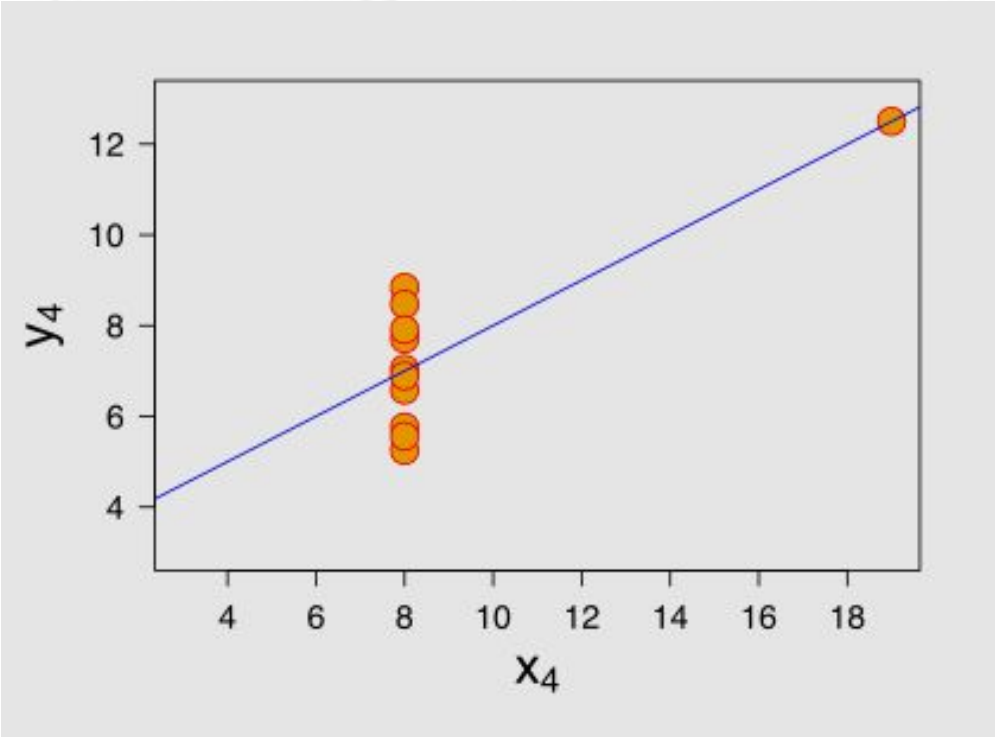
Residual plot showing a clear pattern, indicating Linear Regression no valid!

Especially the 3rd plot

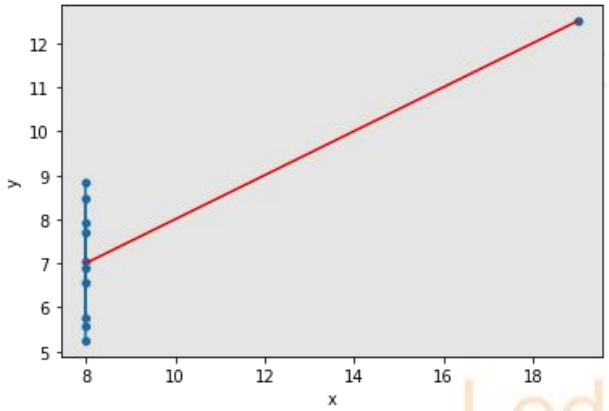


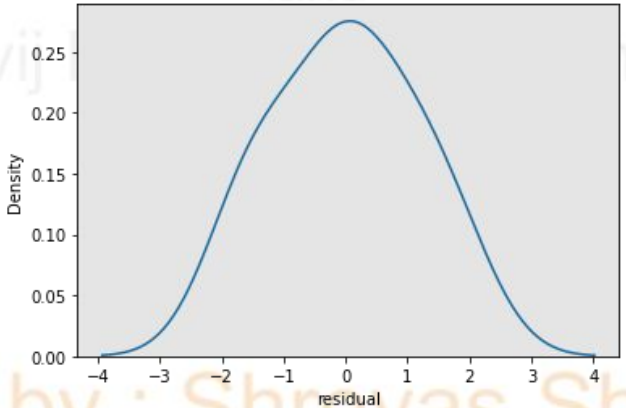


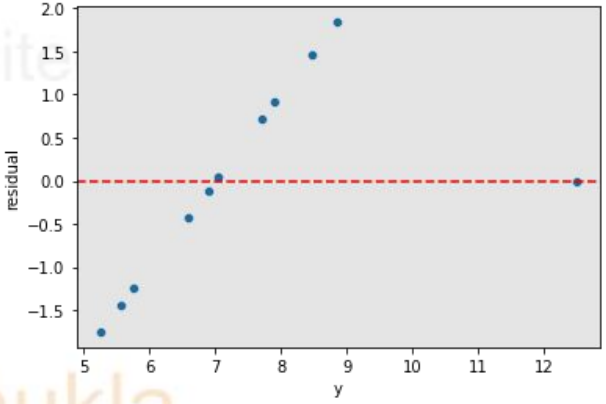




Residual plot showing a clear pattern, indicating Linear Regression not valid!







Let’s create in Python

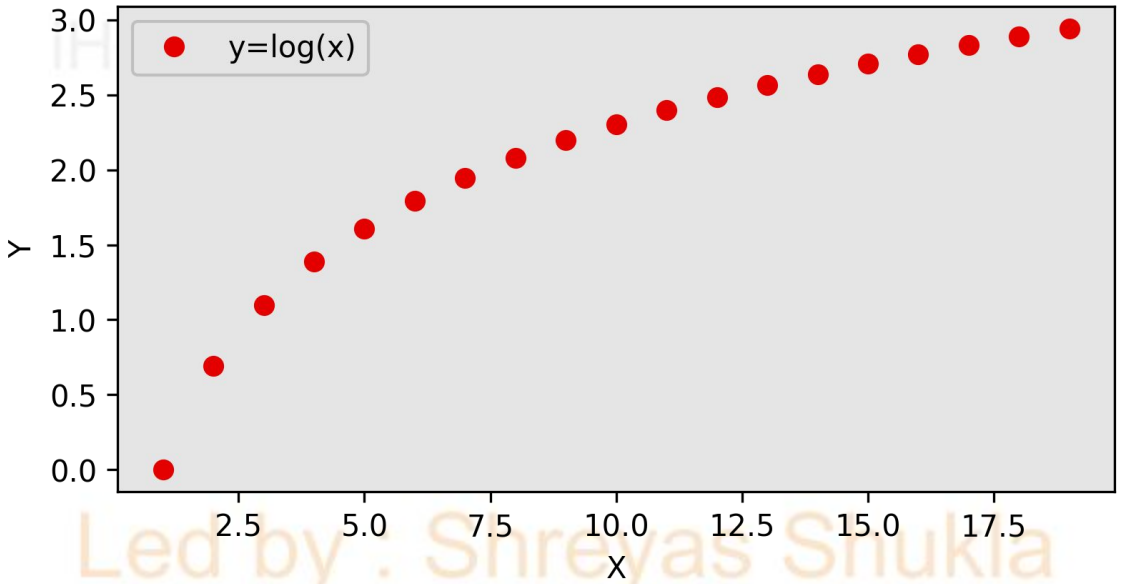
Polynomial Regression

Theory and Motivation

Two issues polynomial regression will address for us:

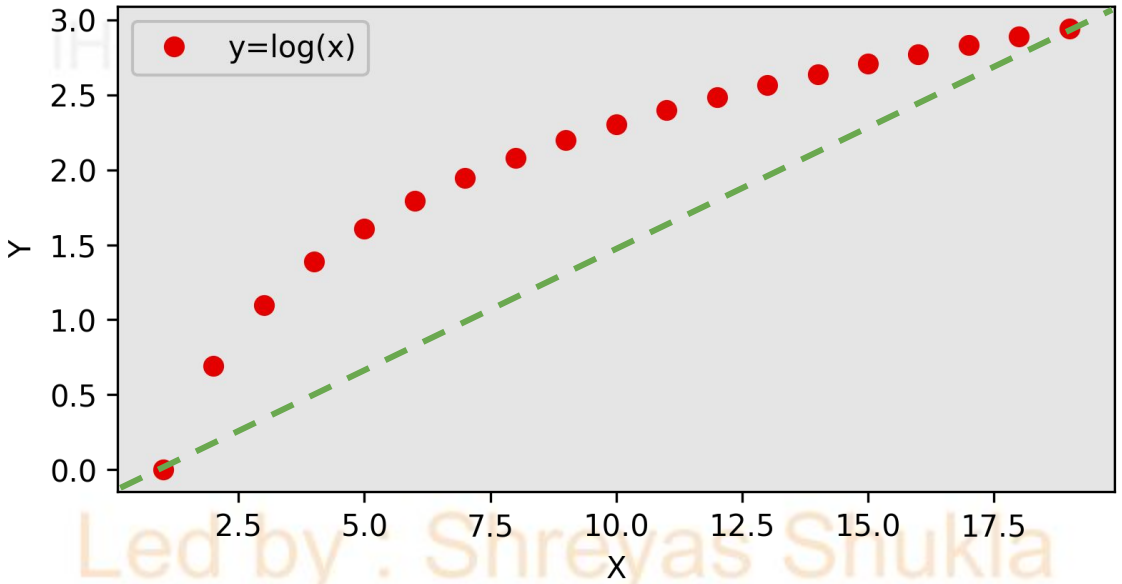
- Non-linear feature relationships to label
- Interaction terms between features

Non- linear:



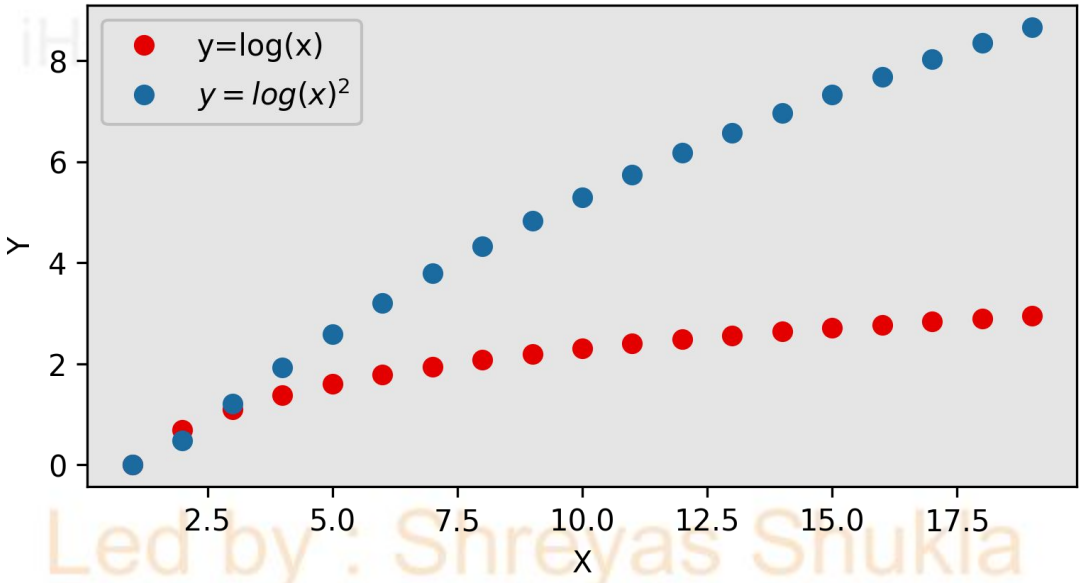
Caution: x inside log(x) is not feature

Not a linear relationship

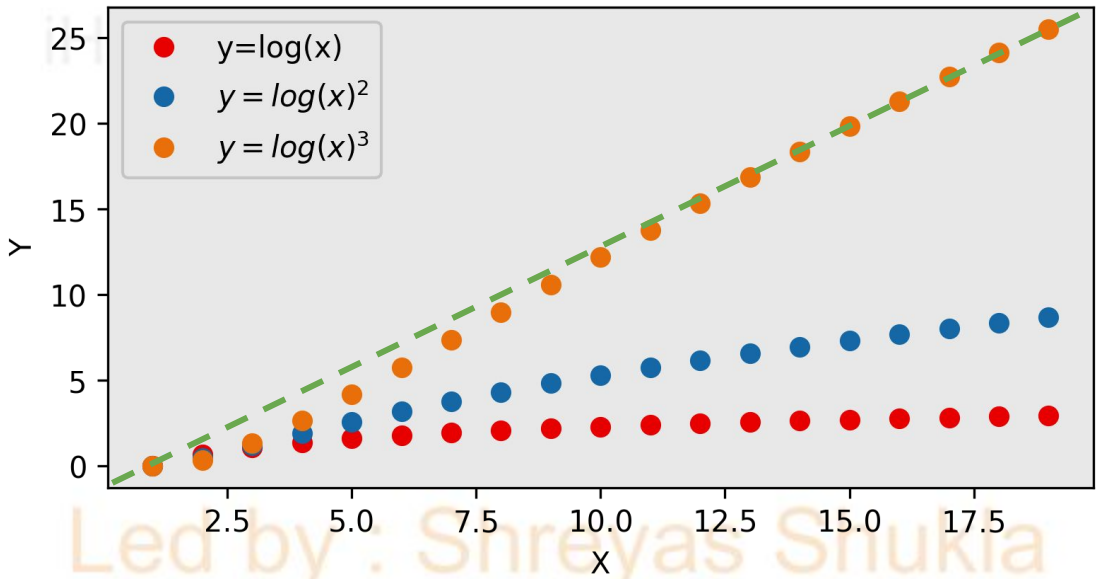


What about the square?

Seems moving to linearity



Even more so for higher orders!



- **Interaction terms.**
- What if features are only significant when in sync with one another?
  - Perhaps newspaper advertising spend by itself is not effective, but greatly increases effectiveness if added to a TV advertising campaign. How to check?
  - create a new feature that multiplies two existing features together to create an interaction term

TV, Newspaper, and also TV*Newspaper

The features created include:

- **The bias (the value of 1.0)**
- **Values raised to a power for each degree (e.g. x^1, x^2, x^3, …)**
- **Interactions between all pairs of features (e.g. x1 * x2, x1 * x3, …)**

Datasets **A** and **B**

- **$1, A, B, A^2, AB, B^2$**

Generalized terms of features X_1 and X_2

- **$1, X_1, X_2, X_1^2, X_1X_2, X_2^2$**

Let us jump to Jupyter Notebook