# Session 3
# Linear Regression II

PSYC 560
Heungsun Hwang

# Linear Regression: Non-linear Relationship

- The linearity assumption states that the changes in the response Y due to a one-unit change in a predictor are constant, regardless of the value of the predictor.
- We consider a very simple way of accommodating non-linear relationships, using polynomial regression.

# Linear Regression: Non-linear Relationship

- To capture a quadratic shape, we may add $X^2$:

$$f(X) = \beta_0 + \beta_1 X + \beta_2 X^2$$

- Note that this is still a linear regression model with two predictors, so we can use standard linear regression software to estimate the coefficients $\beta_0, \beta_1$, and $\beta_2$.

- We can add higher-degree polynomials (e.g., $X^3$, $X^4$, $X^5$, etc.).

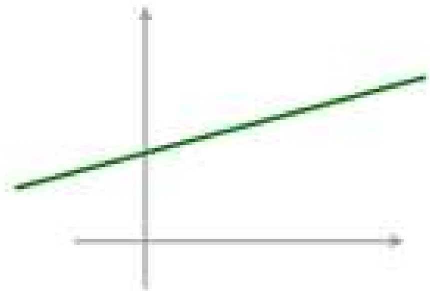
# Polynomial Regression

*1st degree polynomial*

$y = a + bx^1$

straight line with no peaks and no valleys

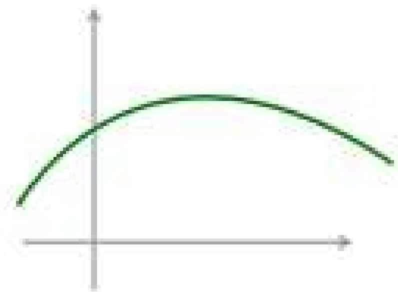
often written as $y = a + bx$



*2nd degree polynomial*

$y = a + bx^1 + cx^2$

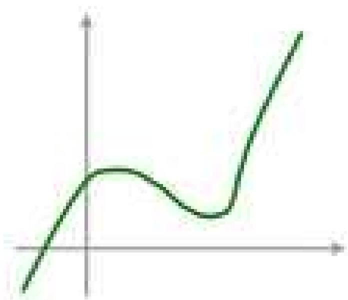
curved line with only one peak or one valley.



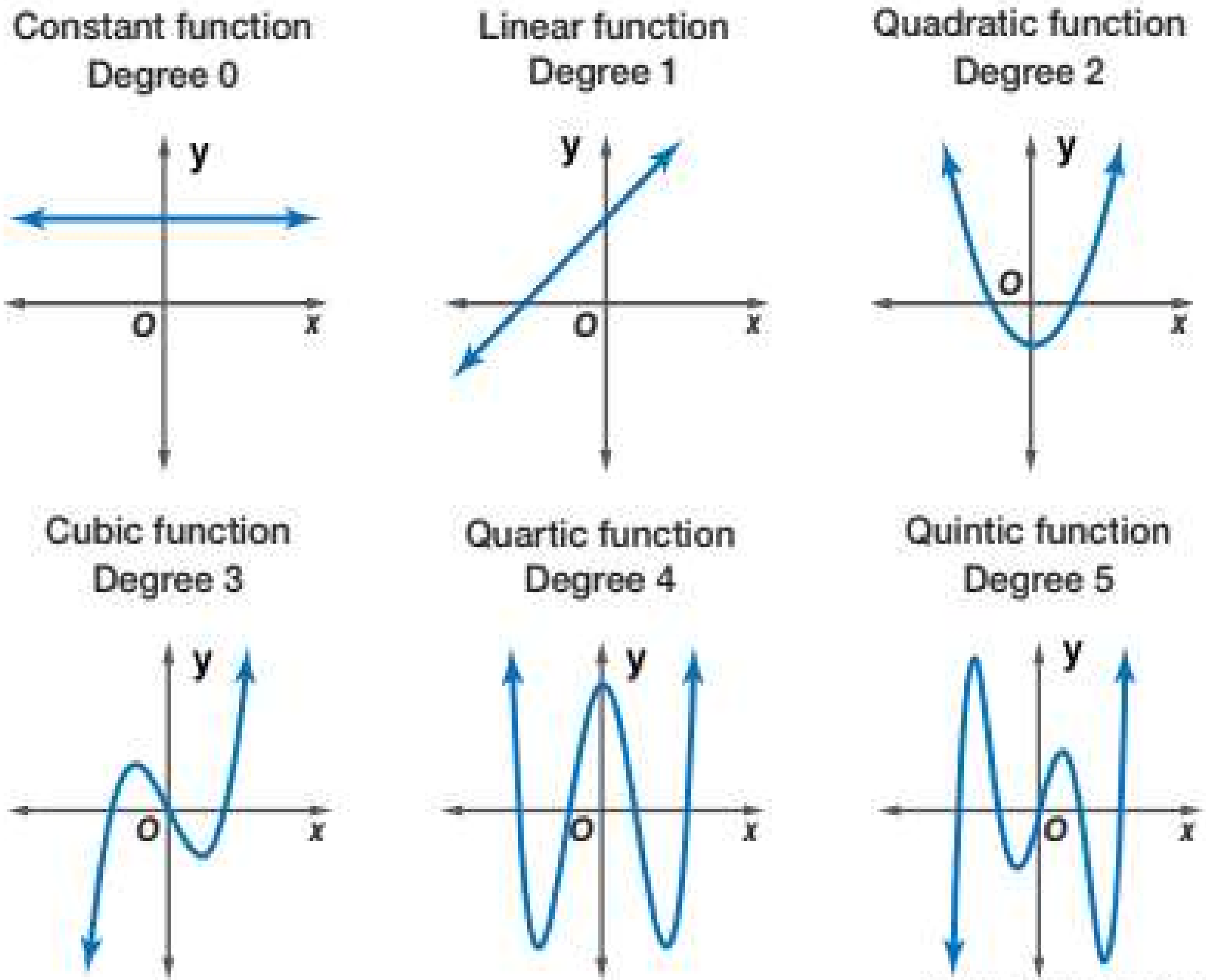
*3rd degree polynomial*

$y = a + bx^1 + cx^2 + dx^3$

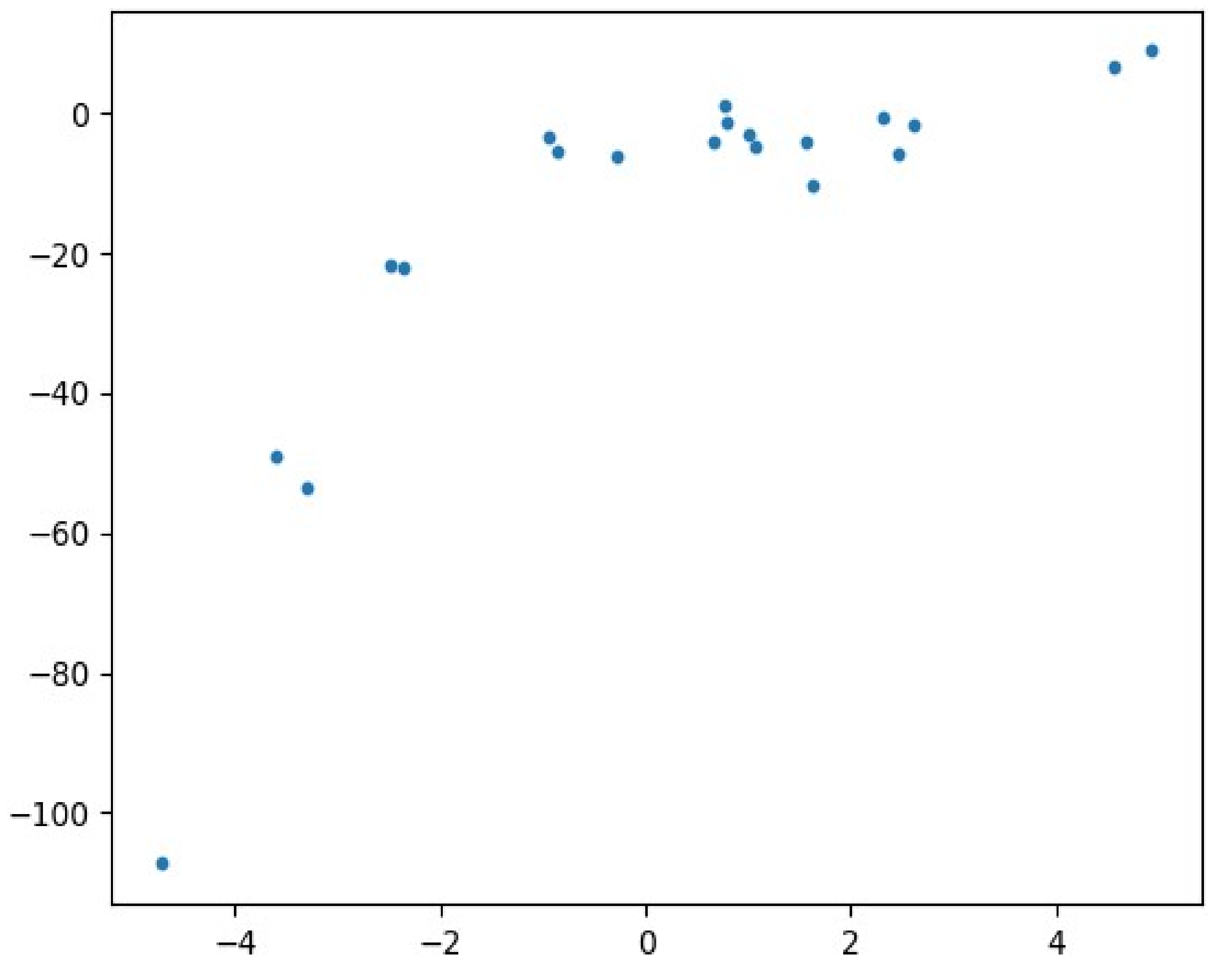
curved line with multiple peaks & valley.



https://www.davidlongstreet.com/polynomials.html

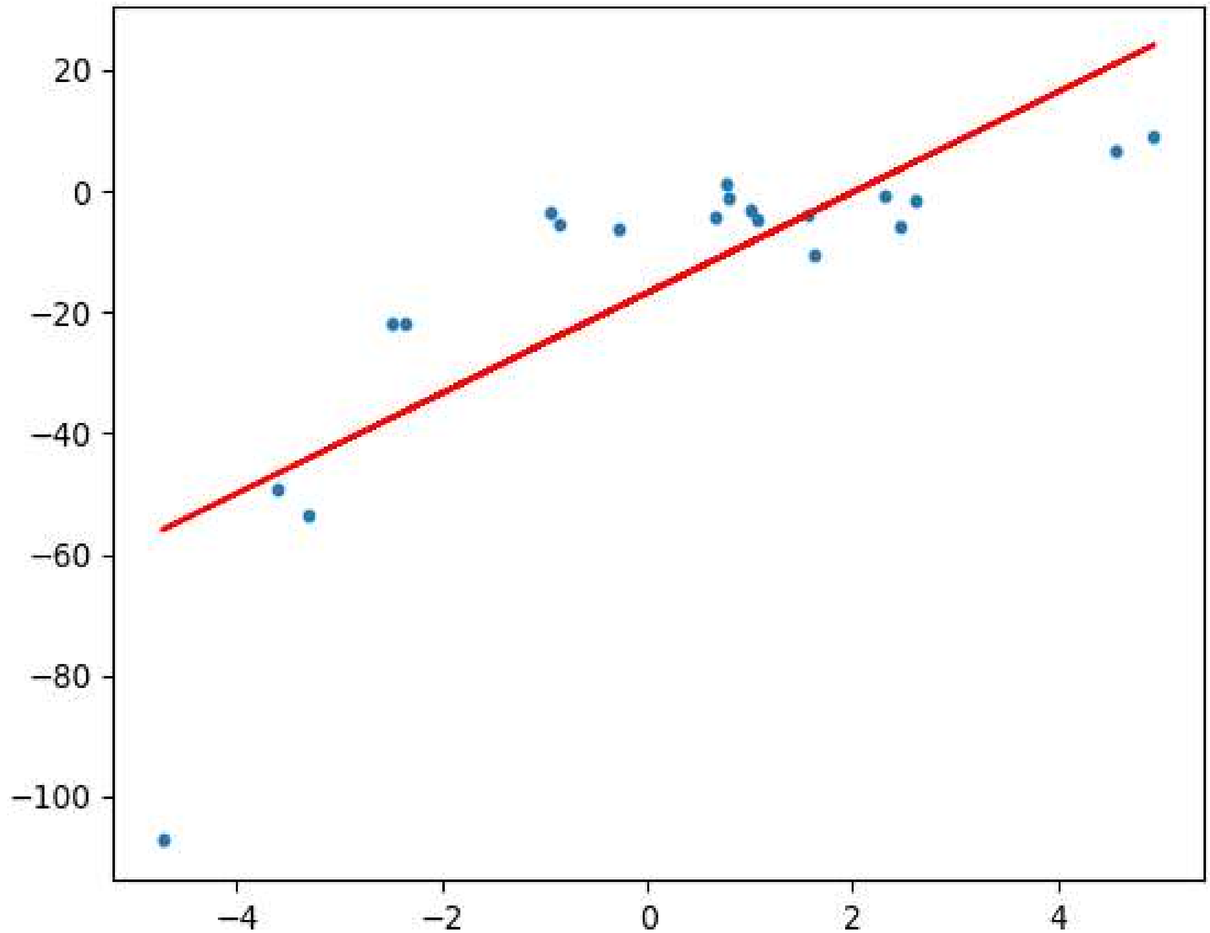



http://www.math.glencoe.com/




https://towardsdatascience.com/polynomial-regression-bbe8b9d97491

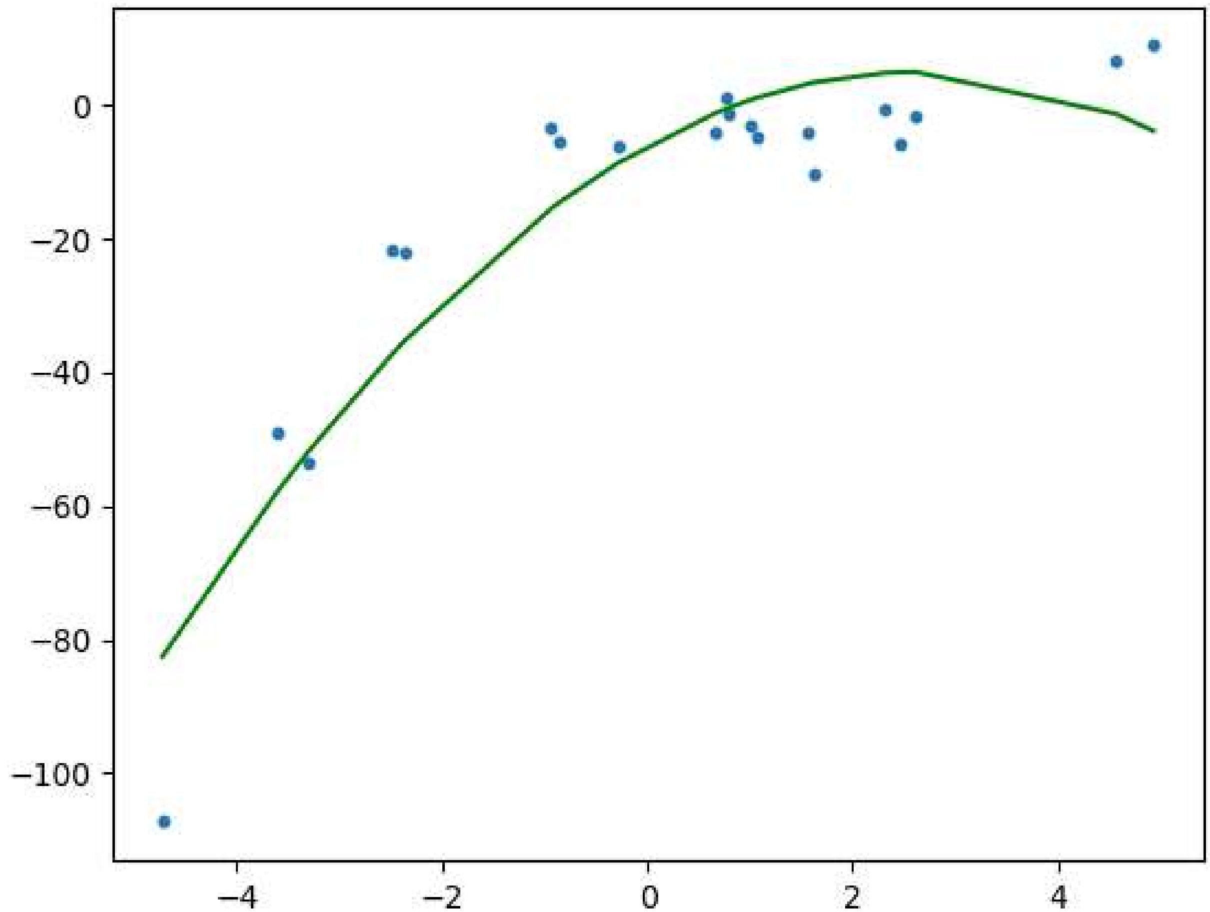
# Linear regression




$R^2 = 0.639$

https://towardsdatascience.com/polynomial-regression-bbe8b9d97491

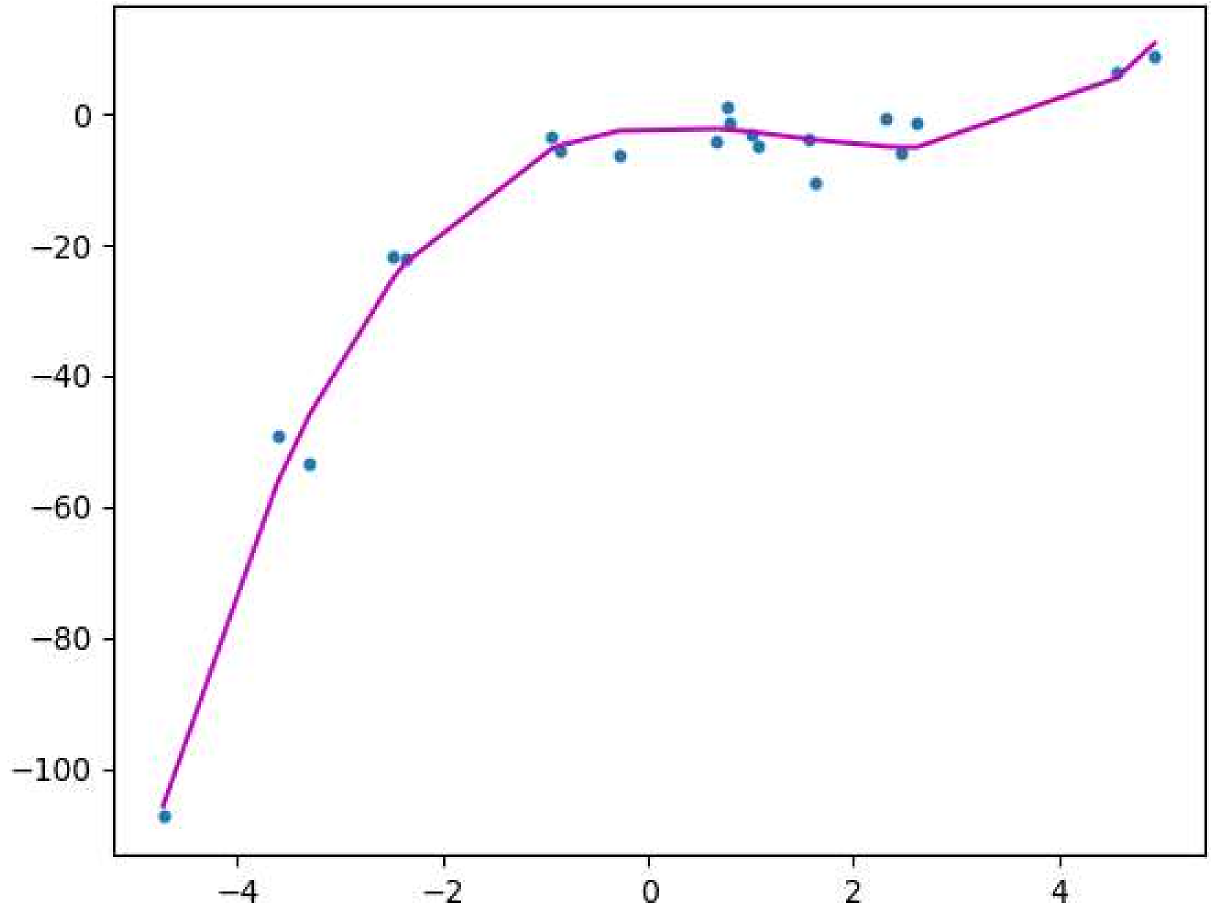
# Polynomial regression (degree = 2)




$R^2 = 0.854$

https://towardsdatascience.com/polynomial-regression-bbe8b9d97491

# Polynomial regression (degree = 3)




$R^2 = 0.983$

https://towardsdatascience.com/polynomial-regression-bbe8b9d97491

# The Bias–Variance Tradeoff

- The bias of a statistical method/model refers to the error that is introduced by approximating a real-life problem, which may be extremely complicated, by much simpler models/methods.

  - For example, linear regression assumes that there is a linear relationship between Y and Xs, which is often unlikely in a real-life situation.

# The Bias–Variance Tradeoff

- The variance of a statistical method/model is the amount by which $\hat{f}(X)$ would change if it is estimated using a different training set.

- Different training datasets will result in a different $\hat{f}(X)$. But ideally the estimate for $f(X)$ should not vary too much between training sets. However, if a method has high variance, then small changes in the training data can result in large changes in $\hat{f}(X)$.

- In general, more flexible statistical methods/models have higher variance.

# The Bias–Variance Tradeoff

- A statistical method's expected generalization error beyond the training sample is a sum of three terms, the bias, variance, and a quantity called the irreducible error, resulting from noise in the problem itself.

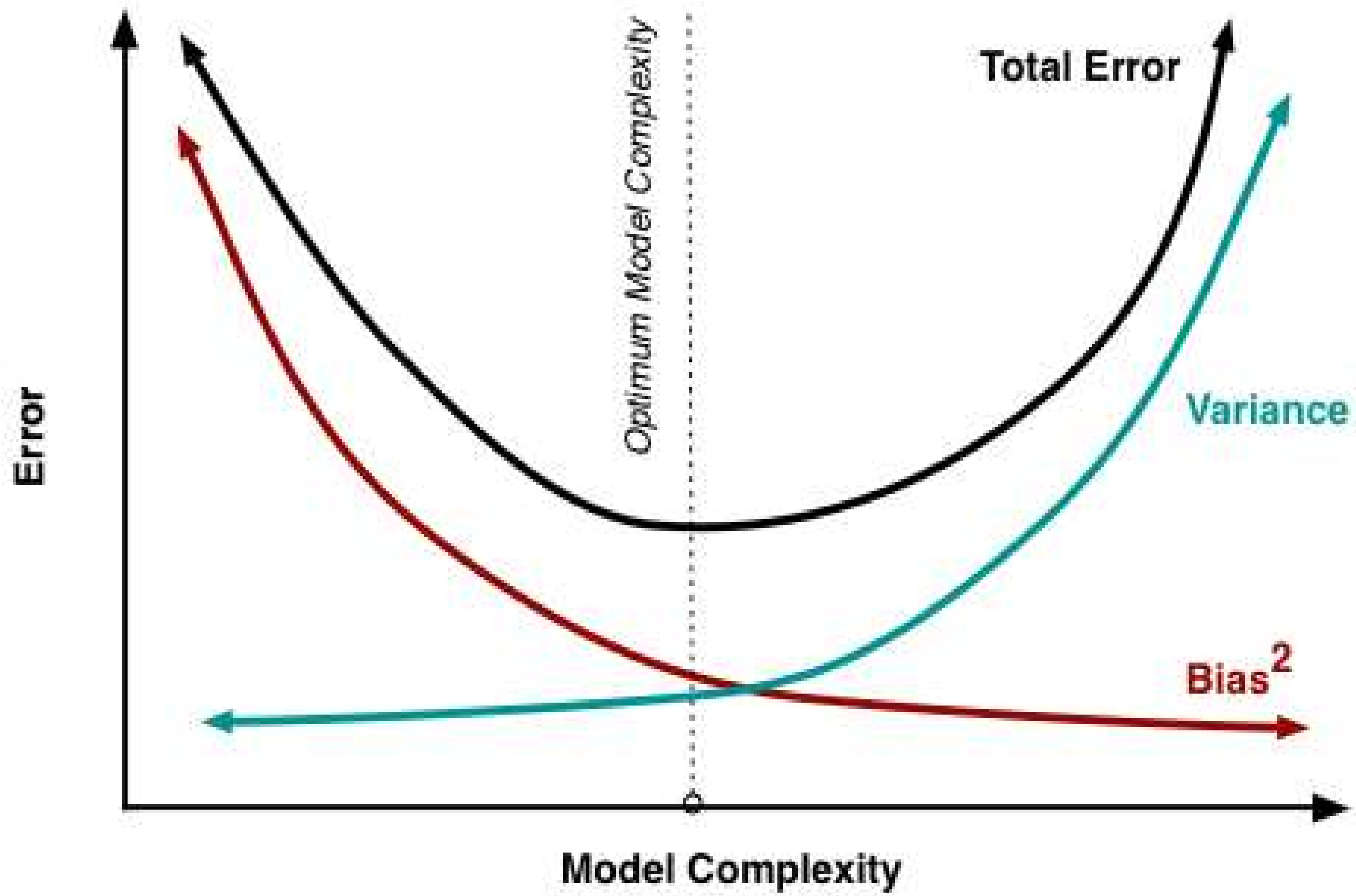
- That is, the expected test mean square error (MSE) is:

$$E(y_0 - \hat{f}(x_0))^2 = \mathrm{Var}\left(\hat{f}(x_0)\right) + [\mathrm{Bias}(\hat{f}(x_0)]^2 + \mathrm{Var}(e)$$

# The Bias–Variance Tradeoff

- This indicates that good test set performance of a statistical learning method requires low variance as well as low (squared) bias.
  - It is not easy to simultaneously minimize these two sources of error that prevent learning methods from generalizing beyond their training set.
- This is referred to as a trade-off because it is easy to obtain a method with extremely low bias but high variance or a method with a very low variance but high bias.
- The challenge lies in finding a method for which both the variance and squared bias are low.

# The Bias–Variance Tradeoff




http://scott.fortmann-roe.com/docs/BiasVariance.html

# Generalizations of Linear Regression

- We will discuss methods that expand the scope of linear models:
  - Classification problems: logistic regression, support vector machines
  - Non-linearity: K-nearest neighbors regression
  - Interactions: Tree-based methods, bagging, random forests, boosting
  - Regularized methods: Ridge/lasso regression

# K-Nearest Neighbors Regression

- K-nearest neighbors regression (KNN regression) is one of the simplest and best-known non-parametric (nonlinear) regression methods.

- Given K and a prediction point $x_0$, KNN regression identifies the K training observations that are closest to $x_0$, represented by Q. It then estimates $f(x_0)$ using the average of all the training responses, as follows.

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in Q} y_i$$

# K-Nearest Neighbors Regression

| Age X (month) | Height Y (cm) |
|---|---|
| 18 | 76.1 |
| 19 | 77.0 |
| 20 | 78.1 |
| 21 | 78.2 |
| 22 | 78.8 |
| 23 | 79.7 |
| 24 | 79.9 |
| 25 | 81.1 |
| 26 | 81.2 |
| 27 | 81.8 |
| 28 | 82.8 |
| 29 | 83.5 |

Moore & McCabe (1998, p.136)

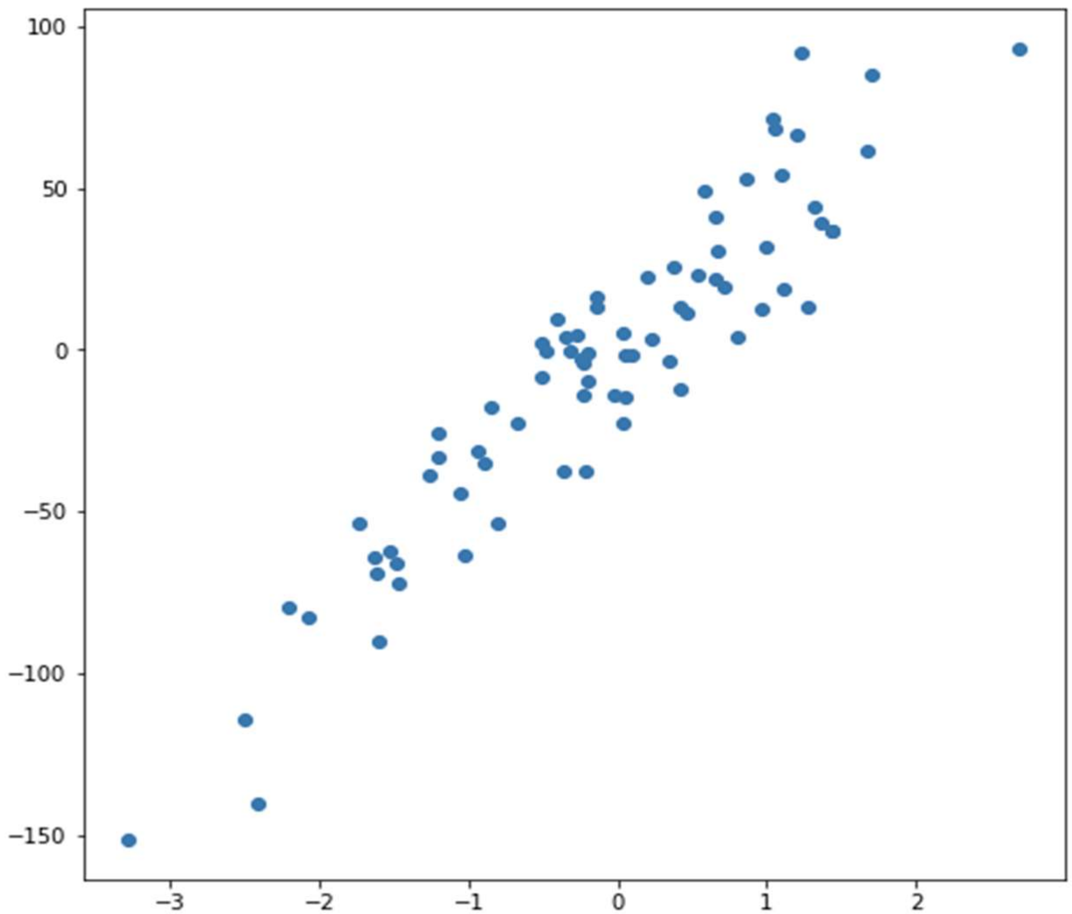
If age = 30 ($X_0$), then height?

K = 1, $\hat{Y} = 83.5$

K = 2, $\hat{Y} = (83.5+82.8)/2$

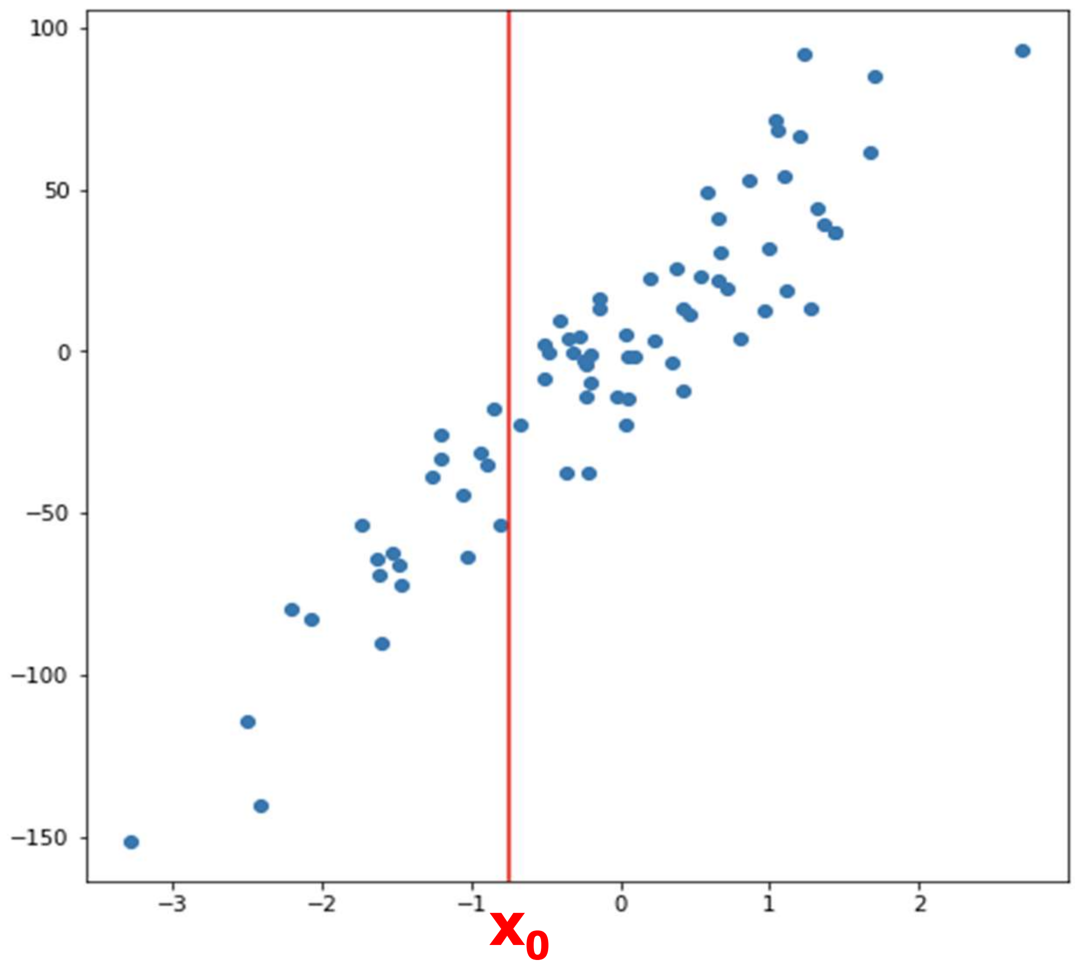
K = 3, $\hat{Y} = (83.5+82.8+81.8)/3$

# K-Nearest Neighbors Regression



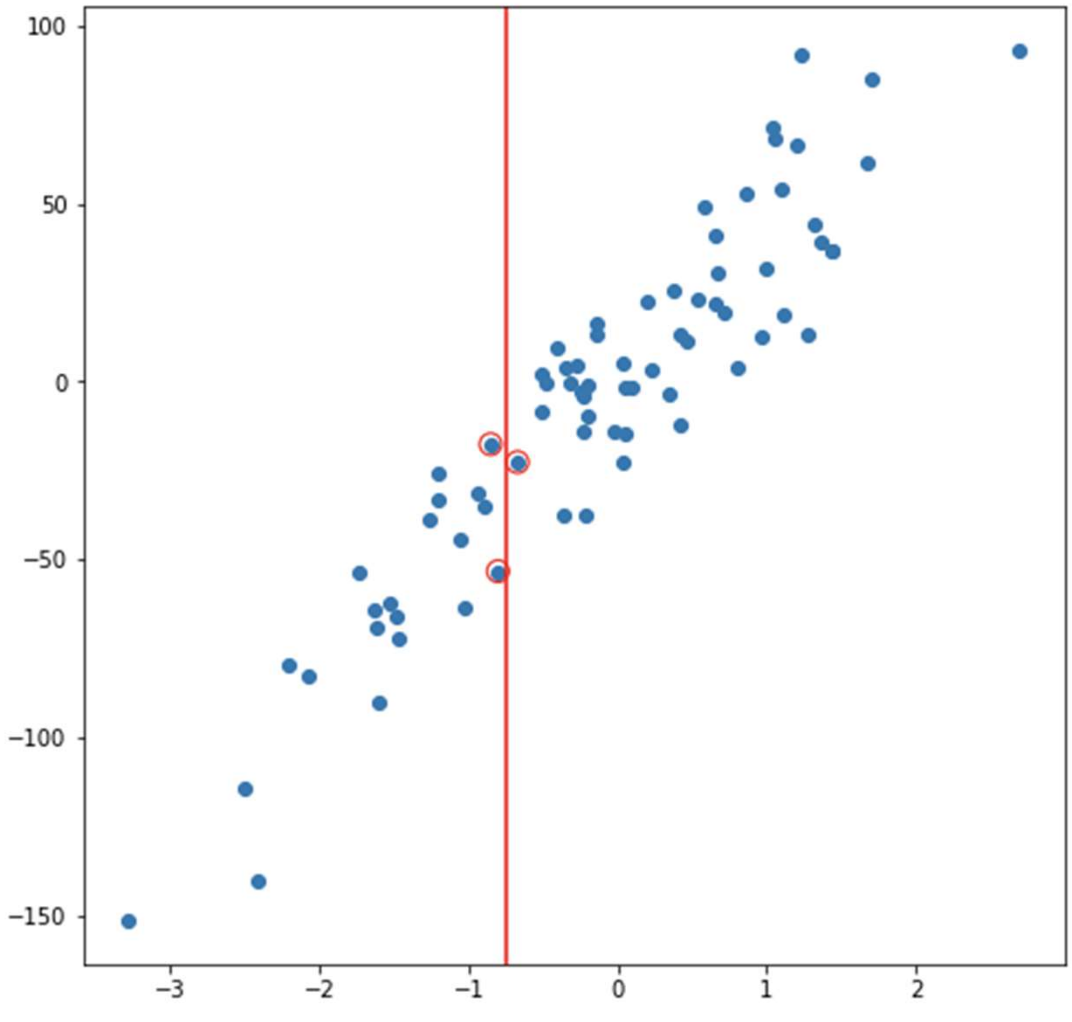

https://towardsdatascience.com/the-basics-knn-for-classification-and-regression-c1e8a6c955

# K-Nearest Neighbors Regression

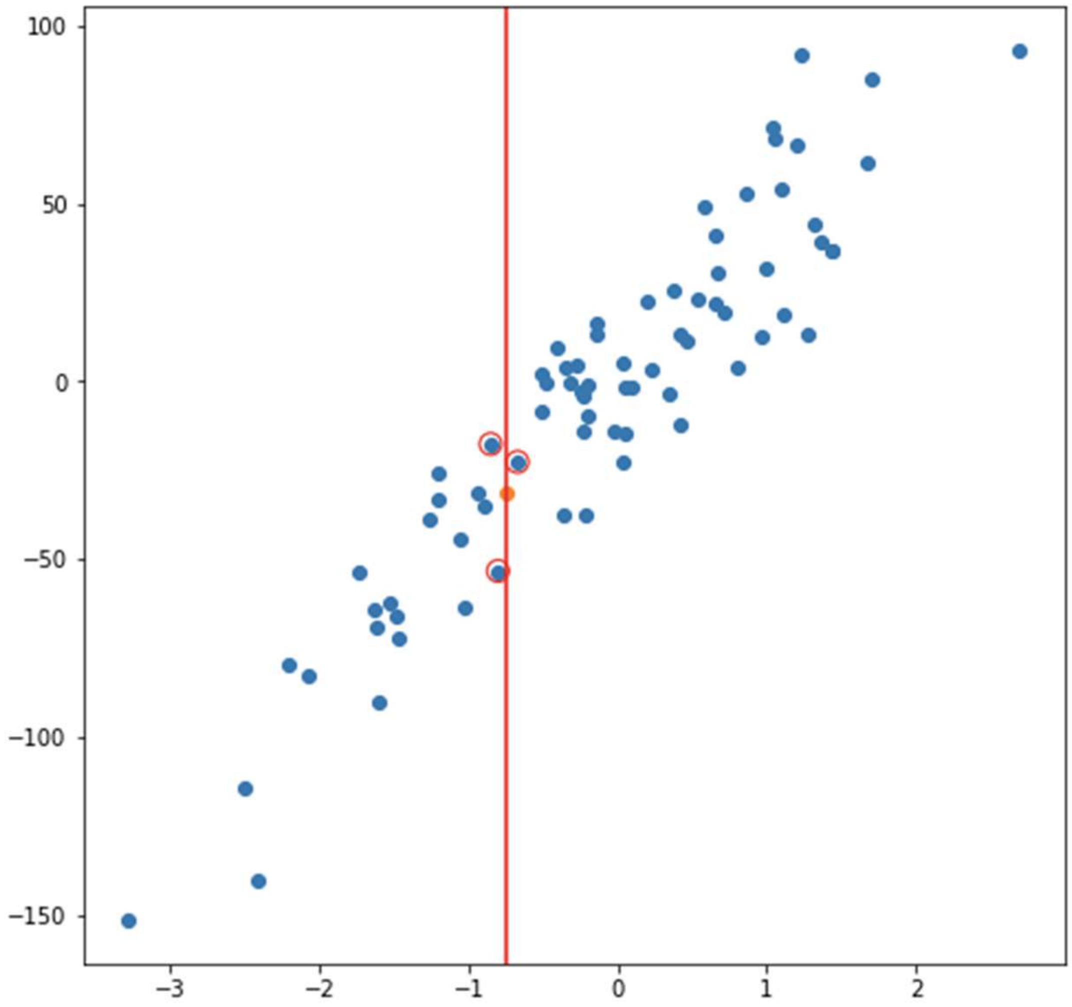


# K-Nearest Neighbors Regression

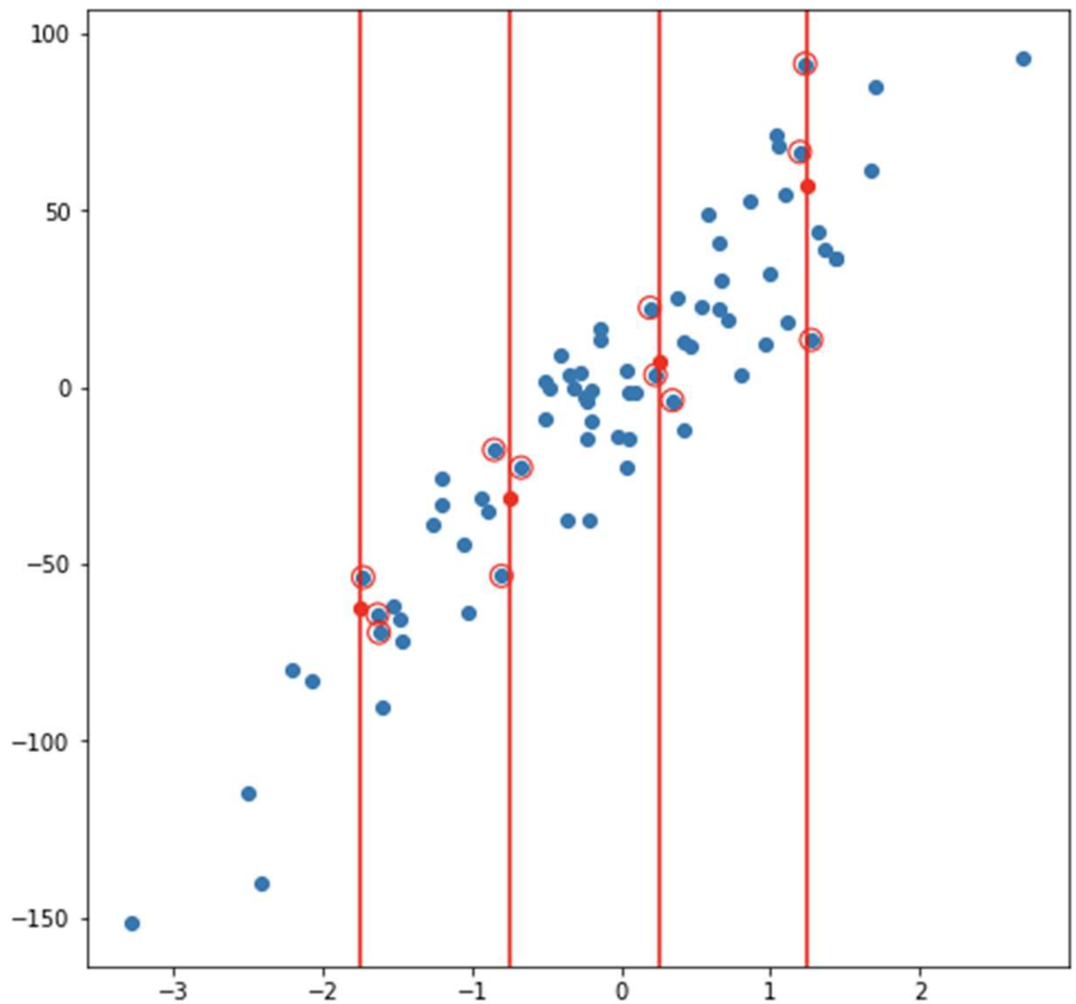
K = 3



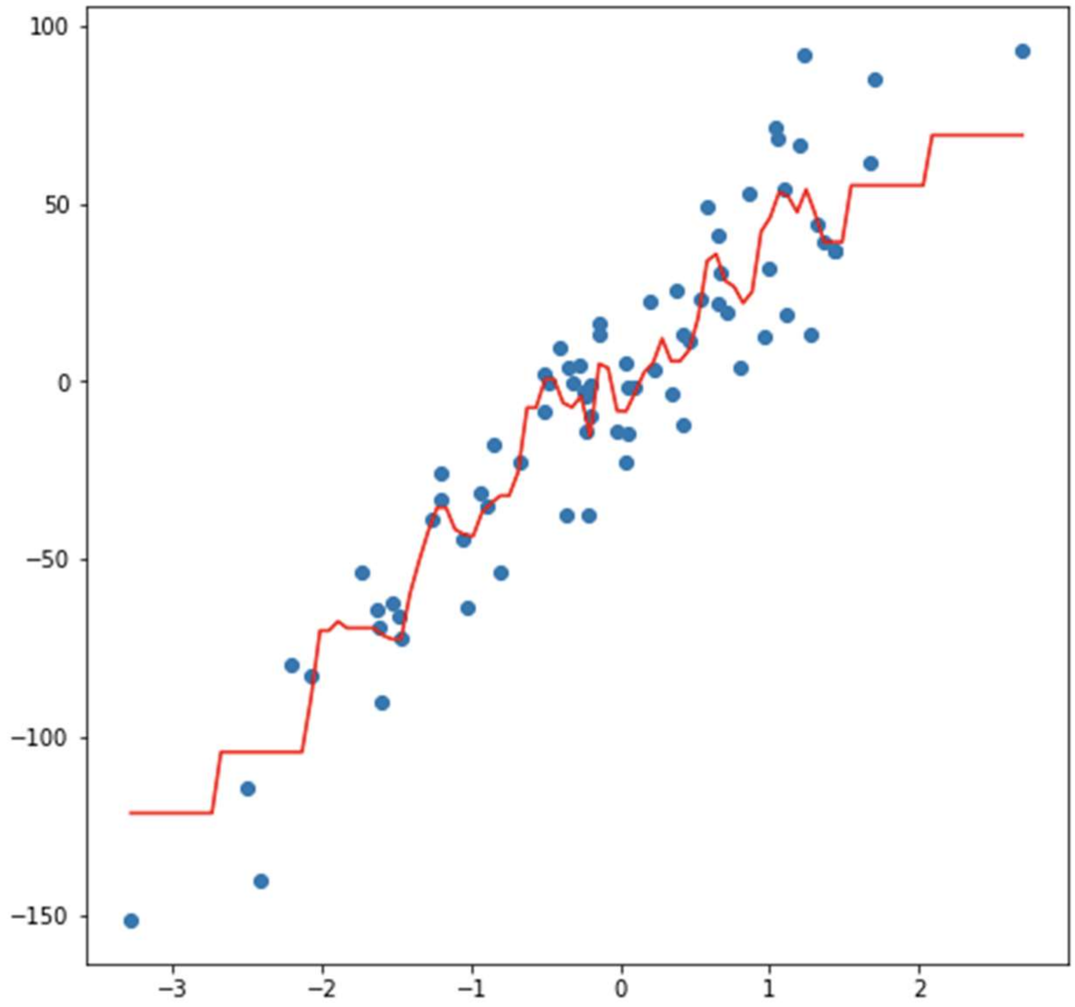
# K-Nearest Neighbors Regression



# K-Nearest Neighbors Regression



# K-Nearest Neighbors Regression

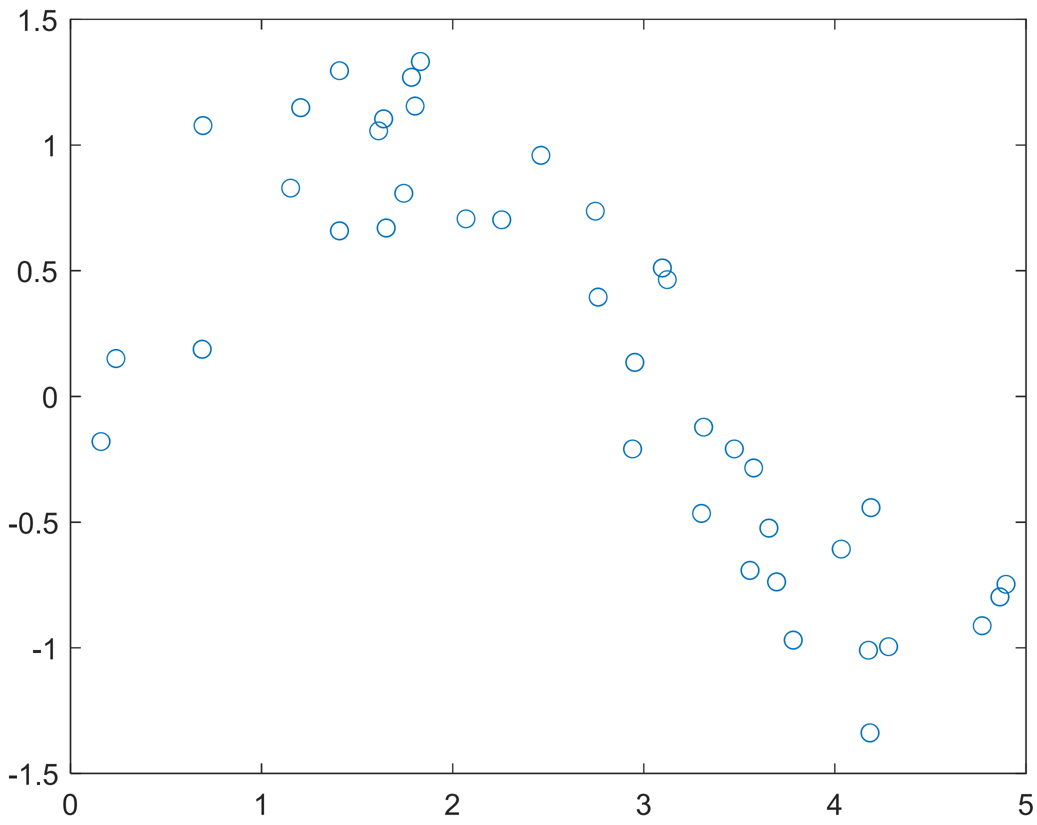


# K-Nearest Neighbors Regression

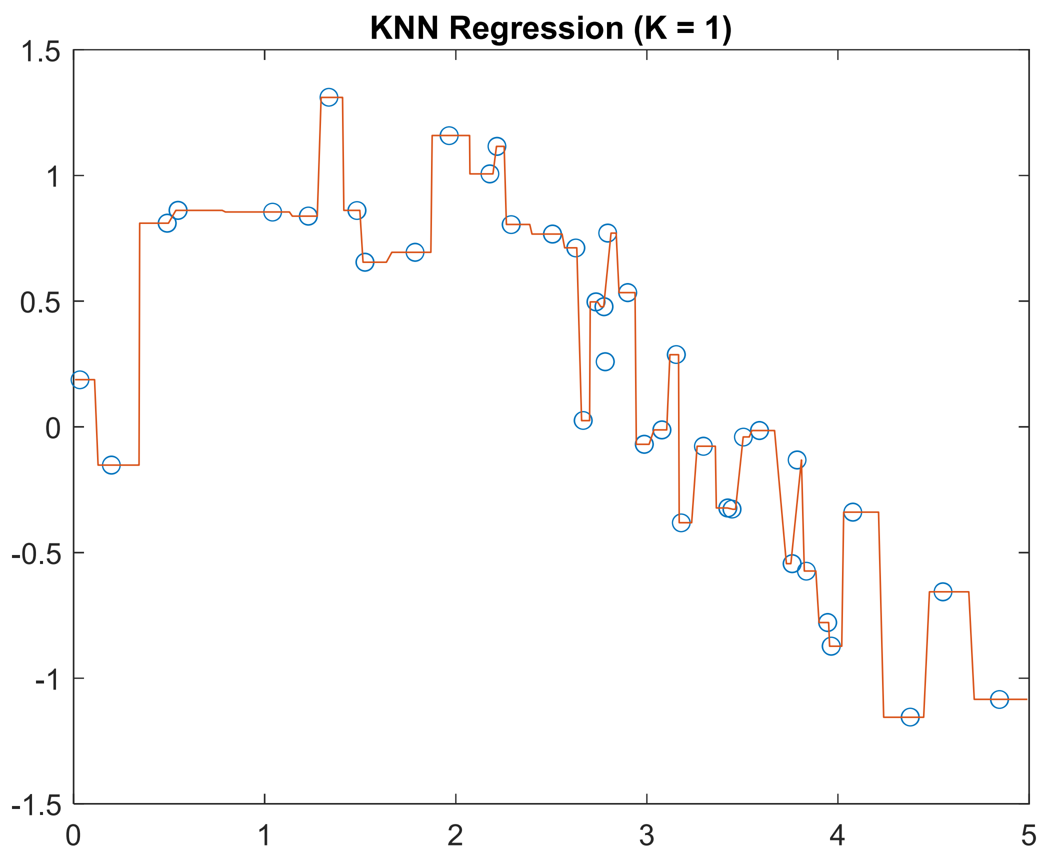
- K (the number of neighbors): When K = 1, KNN regression interpolates the training observations and takes the form of a step function. Increasing K will tend to give a smooth or less wiggly function. Yet, there is no single value of K that will work for every case. The optimal value for K will depend on the bias-variance tradeoff. Cross validation can be used for deciding K.

- Distance metric: There are different ways to measure how close two points are to each other, and the differences between these methods can become significant in higher dimensions. Most commonly used is Euclidean distance.
  - $d(x, x_0) = \sqrt{\sum_{p=1}^{P} (x_p - x_{0p})^2}$

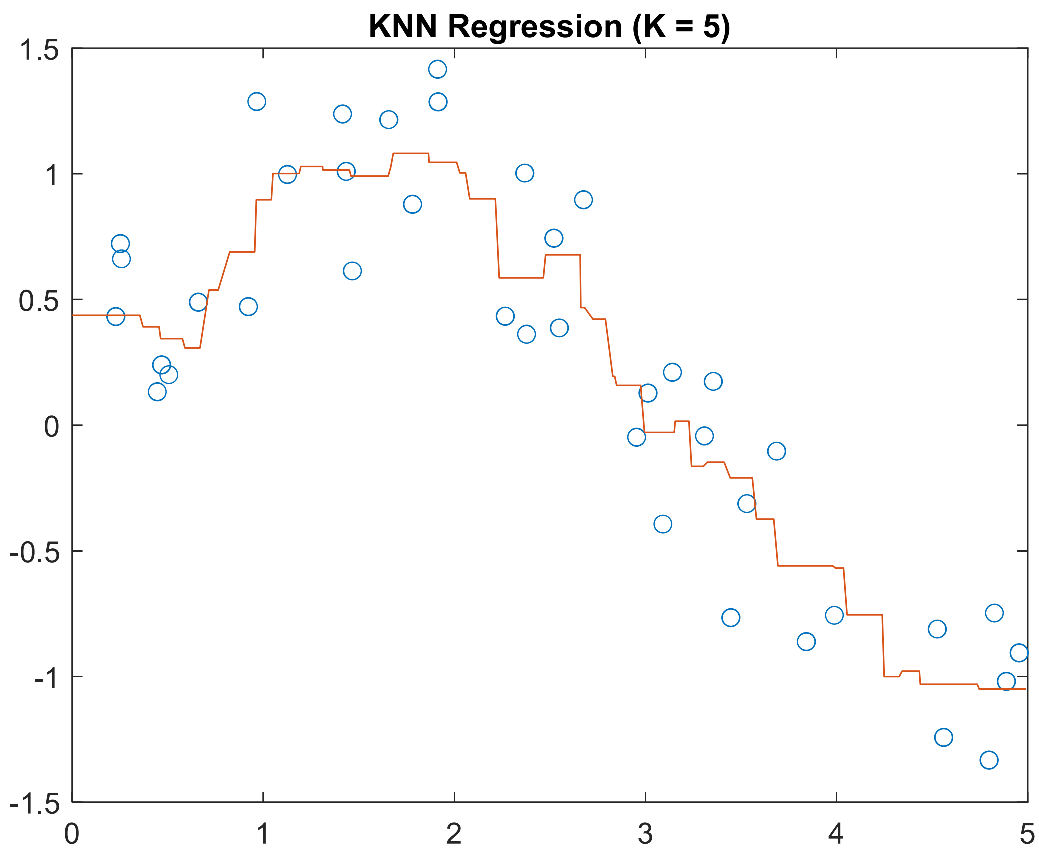
# K-Nearest Neighbors Regression



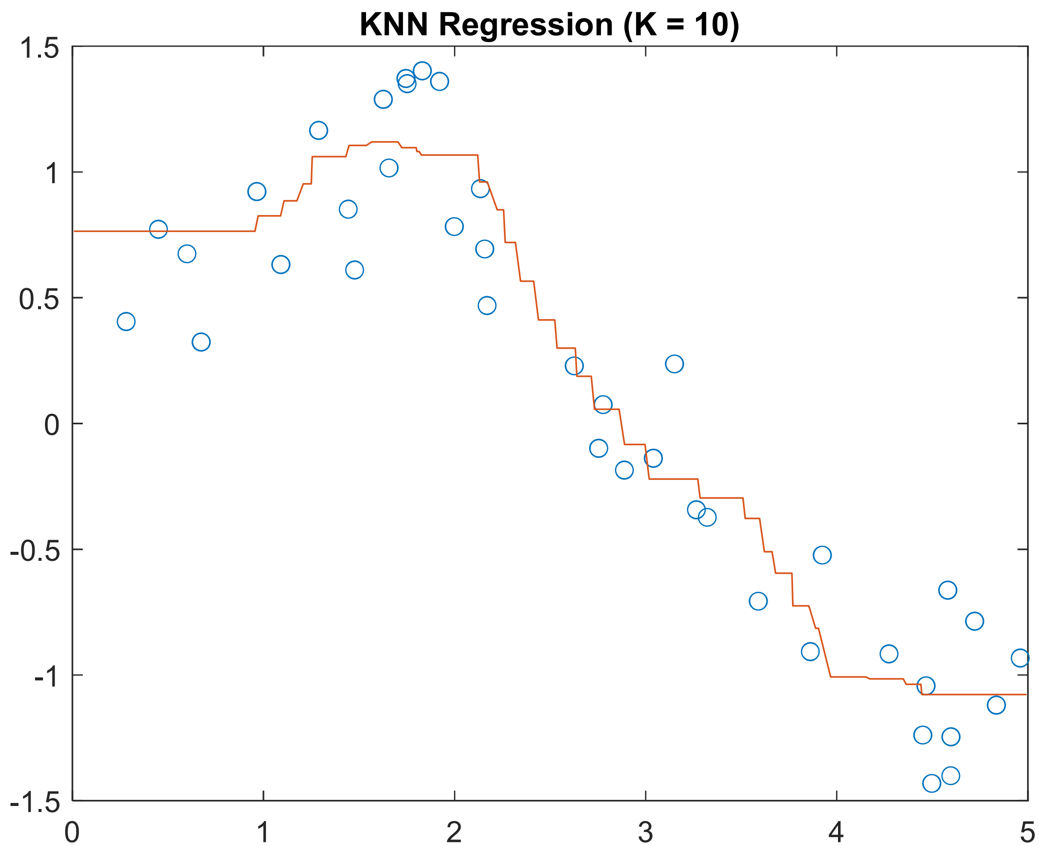
# K-Nearest Neighbors Regression



# K-Nearest Neighbors Regression



# K-Nearest Neighbors Regression



# K-Nearest Neighbors Regression

- KNN regression is easy to implement and will outperform linear regression when the true relationship between X and Y is substantially nonlinear.
- KNN regression can be chosen over linear regression when the true relationship between X and Y is unknown.
- But, in high dimensions (the number of predictors is large), KNN regression often performs worse than linear regression – the curse of dimensionality.
- Another issue with KNN regression is that it lacks interpretability.

# Lab: Linear Regression II

1. Multiple linear regression with dummy variables – GlastonburyDummy.csv
2. Polynomial regression – curran_training.csv
3. KNN regression - curran_training.csv
4. Performance of linear regression models and KNN regression for test data (in MSE) – curran_training.csv & curran_test.csv