NYC DATA SCIENCE ACADEMY

Python Machine Learning

Simple Linear Regression

NYC Data Science Academy

Outline

- What is Machine Learning
- Introduction to Scikit-Learn
- Simple Linear Regression
  - Estimating Coefficients
  - Coefficient of Determination

What is Machine Learning?

- **Task**: To recognize a tree
- **Problem**:
  - It's hard to write a program to do this.
- **Solution**:
  - Learn from data (observations).
- ML-based tree recognition systems can be much more effective than hand-programmed systems.

What is Machine Learning?

- Machine learning is a subfield of computer science that provides computers with the ability to learn without being explicitly programmed.
- The machine learning paradigm can be viewed as “programming by example”.
  - The learning is always based on some sort of observations or data.
  - The goal is to devise learning algorithms that do the learning automatically without human assistance.

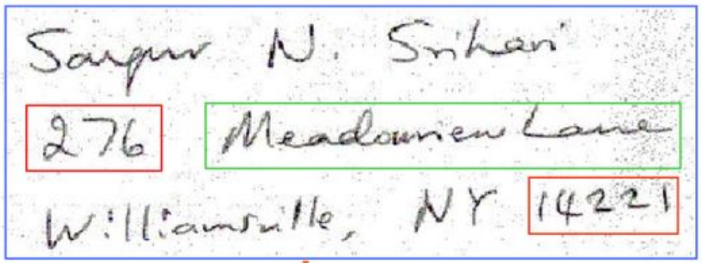
What can Machine Learning do?

- Applications that can’t be programmed by hand:
  - Autonomous helicopters, handwriting recognition, most Natural Language Processing (NLP), Computer Vision.
- Self-customizing programs:
  - Amazon, Netflix product recommendations
- Database mining:
  - Large data sets from growth of automation/web. Web click data, medical records, biology, engineering...

What can Machine Learning do?

- Handwriting Recognition: address recognition

Street address



Database query

ZIP Code: 14221
Primary number: 276

Records Retrieved

Lexicon entry (Street name)	ZIP+4 add-on
AMHERSTON DR	7006
BELVOIR RD	
CADMAN DR	
CLEARFIELD DR	
FORESTVIEW DR	
HARDING RD	7111
HUNTERS LN	3330
MCNAIR RD	3718
MEADOWVIEW LN	3557
OLD LYME DR	2250
RANCH TRL	2340
RANCH TRL W	2246
SHERBROOKE AVE	3421
SUNDOWN TRL	2242
TENNYSON TER	5916

Recognizer choice (after lex. expansion)

Address encoding

ZIP+4: 142213557

What can Machine Learning do?

- Amazon Recommendation

Machine Learning Terminology

- Variables that are considered to be measurable or present, and have some influence on other variables, are referred as **inputs**, **predictors**, **features**, or **independent variables**.
- Variables that are assumed to be influenced by others and that we want to predict, are called **outputs**, **responses**, or **dependent variables**.
- For example, to predict how smoking, together with other variables, would affect lung cancer rate, smoking is the *predictor* and lung cancer rate is the *response*.

Machine Learning Problems

- There are two main types of machine learning problems:
  - ***Supervised Learning***:
    - Inferential and predictive tasks
    - Understand the relationship between different variables.
    - Predict outcome measurement Y using predictor measurements X.
    - In a *regression* problem, Y is quantitative (e.g., price).
    - In a *classification* problem, Y is categorical (e.g., digit 0-9).

Machine Learning Problems

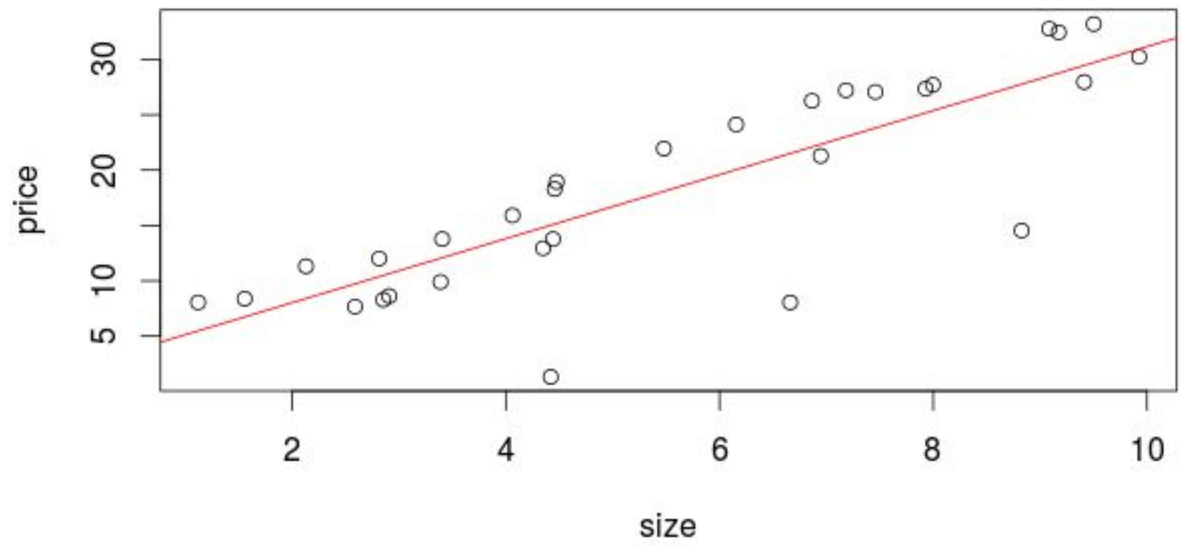
- ***Unsupervised Learning***:
  - No outcome variables, inputs are often called *features*.
  - Objective is more fuzzy - e.g., find groups of people that behave similarly.

Machine Learning Problems

- There are other kinds of learning problems, like reinforcement learning and recommendation systems. We will not cover those.
- The next few slides give examples of the main learning problems.

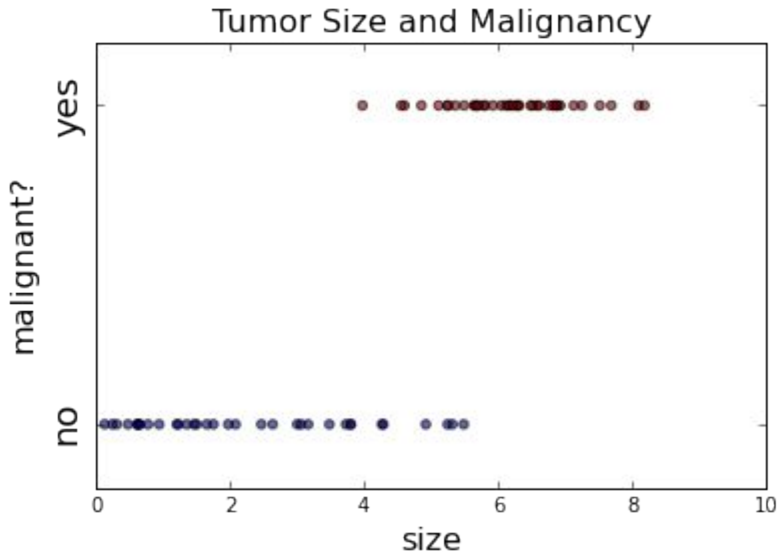
Supervised Learning: regression

- Regression problem: predict continuous output.
- Example: You have a data set of house sizes and prices; given a new house's size, predict its price.



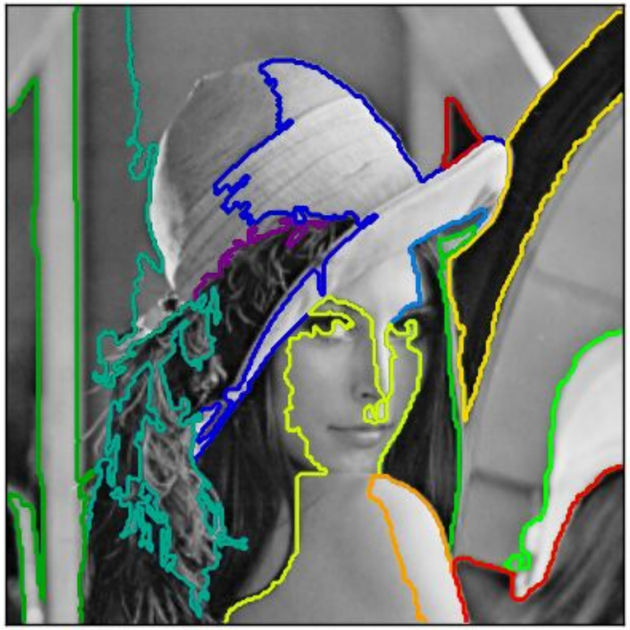
Supervised Learning: classification

- Classification problem: predict discrete output.
- Example: classify whether a tumor is malignant or not by its size.



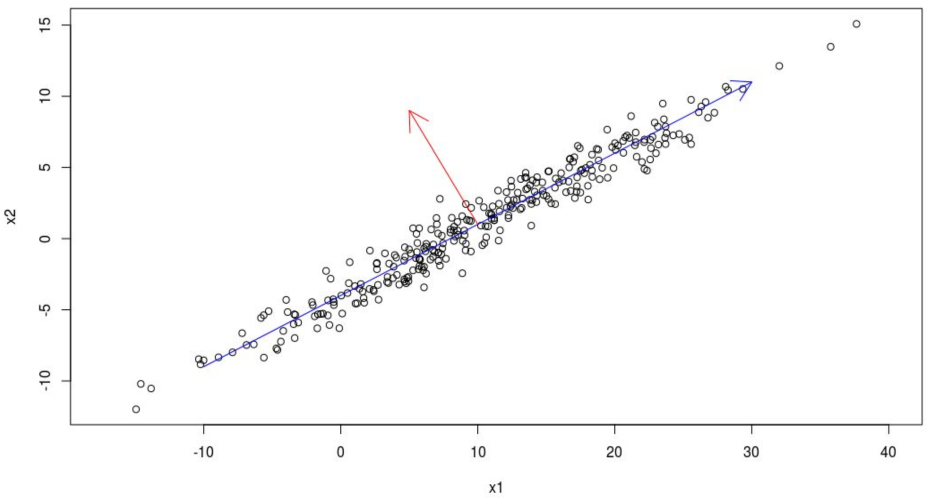
Unsupervised Learning: clustering

- Given some features, cluster the observations into groups.
- Example: segmenting an image in regions based on voxel-to-voxel similarity using hierarchical clustering.



Unsupervised Learning: dimensionality reduction

- Reduce the number of features without losing much information.
  - This is usually a required preprocessing step for supervised learning when the number of features is large or they are highly correlated.
- Example: These two variables are highly correlated. They can be converted into one combined variable without losing much information using Principal component analysis (PCA).



Questions

- Which of the following problems are best suited to machine learning?
  - (a) Classifying numbers into primes and non-primes.
  - (b) Detecting potential fraud in credit card charges.
  - (c) Determining the time it would take a falling object to hit the ground.
  - (d) Determining the optimal cycle for traffic lights in a busy intersection.
  - (e) Determining the age at which a particular medical test is recommended.

Questions

- Which of the following problems are best suited for machine learning?
  - (a) Classifying numbers into primes and non-primes
  - **(b) Detecting potential fraud in credit card charges**
  - (c) Determining the time it would take a falling object to hit the ground
  - **(d) Determining the optimal cycle for traffic lights in a busy intersection**
  - **(e) Determining the age at which a particular medical test is recommended**

Questions

- Identify which type of learning can be used to solve each task below:
  - Categorize books into groups, knowing only what books were bought by each customer.
  - Decide the maximum allowed debt for a bank customer.

Questions

- Identify which type of learning can be used to solve each task below:
  - Categorize books into groups, knowing only what books were bought by every customer.
  - Deciding the maximum allowed debt for each bank customer.
- The first is unsupervised learning. We have no "correct answers" to "supervise" our learning. More precisely, it is a clustering problem.
- The second is a supervised learning problem, more precisely, a regression problem.

Outline

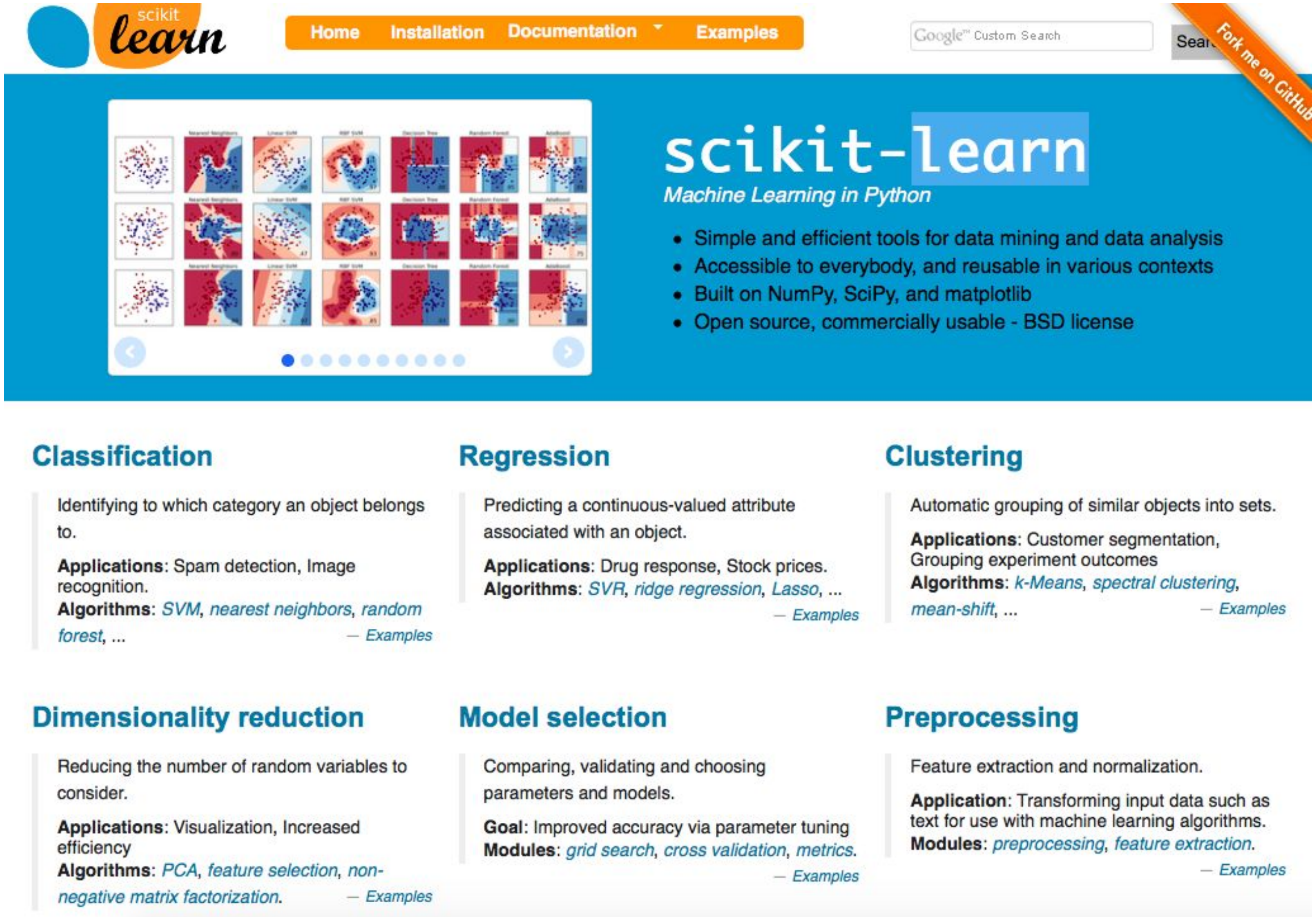
- What is Machine Learning
- **Introduction to Scikit-Learn**
- Simple Linear Regression
  - Estimating Coefficients
  - Coefficient of Determination

Overview

- scikit-learn is an open source machine learning library for the Python programming language. It is built on Numpy, Scipy and matplotlib. It is designed to be an efficient tool box for machine learning and data mining. It provides user friendly functions to facilitate:
  - Supervised learning, including regression and classification.
  - Unsupervised learning.
  - Functions to help test your results, choose the correct algorithm and parameters, etc. *cross validation, feature and model selection.* We will cover those topics in later classes.

Introduction to scikit-learn

- http://scikit-learn.org



Introduction to scikit-learn

- The learning algorithms that scikit-learn provides:
  - Supervised learning
    - Regression
    - Classification
  - Unsupervised learning
    - Clustering
    - Dimension reduction
- We will focus on linear regression in this lecture.

Outline

- What is Machine Learning
- Introduction to Scikit-Learn
- **Simple Linear Regression**
  - Estimating Coefficients
  - Coefficient of Determination

Linear Regression

- Linear regression is a basic supervised machine learning method that aims to uncover the relationship between continuous variables:
  - One or more **explanatory/independent/input** variables: $X_1, X_2, \dots X_p$
  - The **response/dependent/output** variable Y.
- Linear regression is the foundation of many nonlinear regression techniques like Spline regression, Kernel regression, Support Vector Regression and Gauss Process Regression, etc.

Simple Linear Regression

❖ **Simple linear regression** in a special case when there is only one explanatory variable X . Then the relation can be represented quantitatively by:

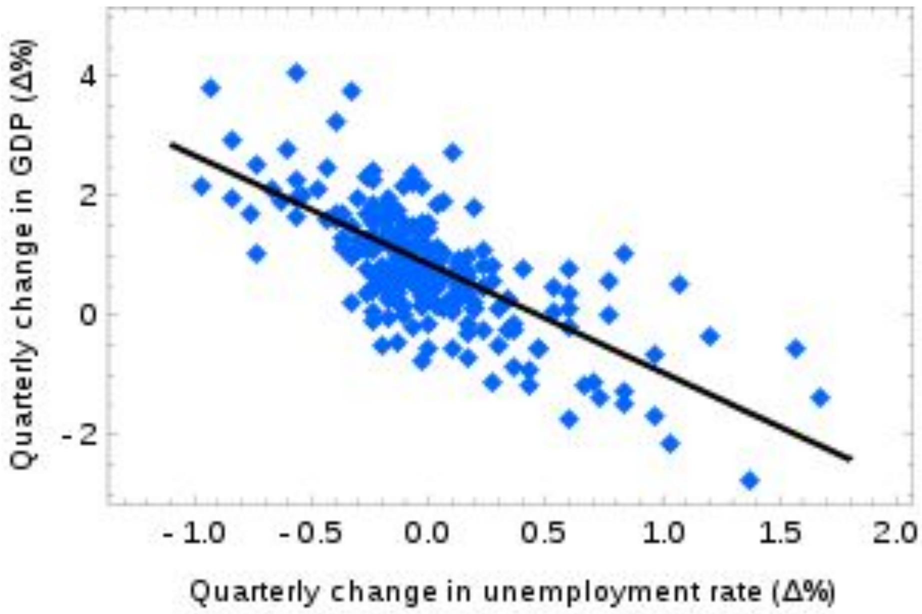
$$Y = \beta_0 + \beta_1 X + \epsilon$$

➢ β_0 and β_1 are two unknown constants that represent the intercept and slope.

➢ ϵ is called the **error term**. This represents the deviation of the value from the linearity.

Simple Linear Regression

- For example, Okun's law in macroeconomics can be modeled by simple linear regression. Here the GDP growth is presumed to be in a linear relationship with the changes in the unemployment rate.



Source: https://en.wikipedia.org/wiki/Simple_linear_regression

Simple Linear Regression

- The Okun's law from previous slide can then be modeled as

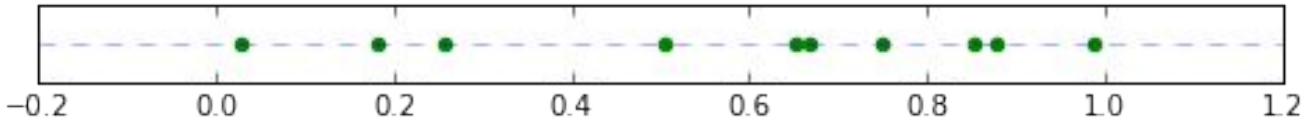
$$\Delta GDP = \beta_0 + \beta_1 \Delta unemployment\_rate + \epsilon$$

The Coefficients

- Below we visualize our simple linear model with an example:

$$Y = 1 + 0.5X + \epsilon. \ (\beta_0 = 1 \text{ and } \beta_1 = 0.5)$$

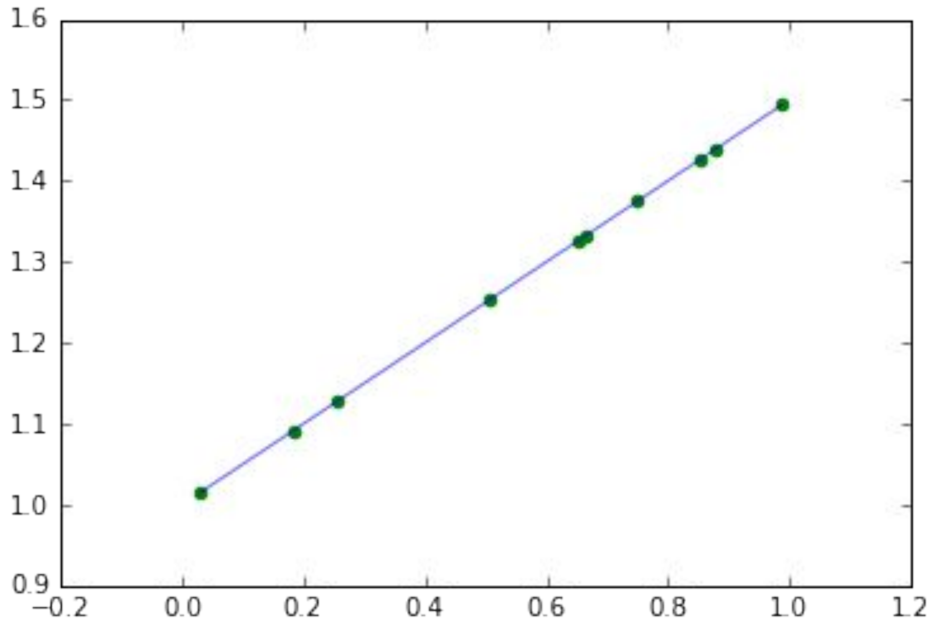
  - β_0 and β_1 define the linear relation. That says, if we observe a set of n observations $X = (x_1, x_2, ..., x_n)$:



The Coefficients

- The linear relation indicates that the outcome $Y = (y_1, y_2, \ldots, y_n)$

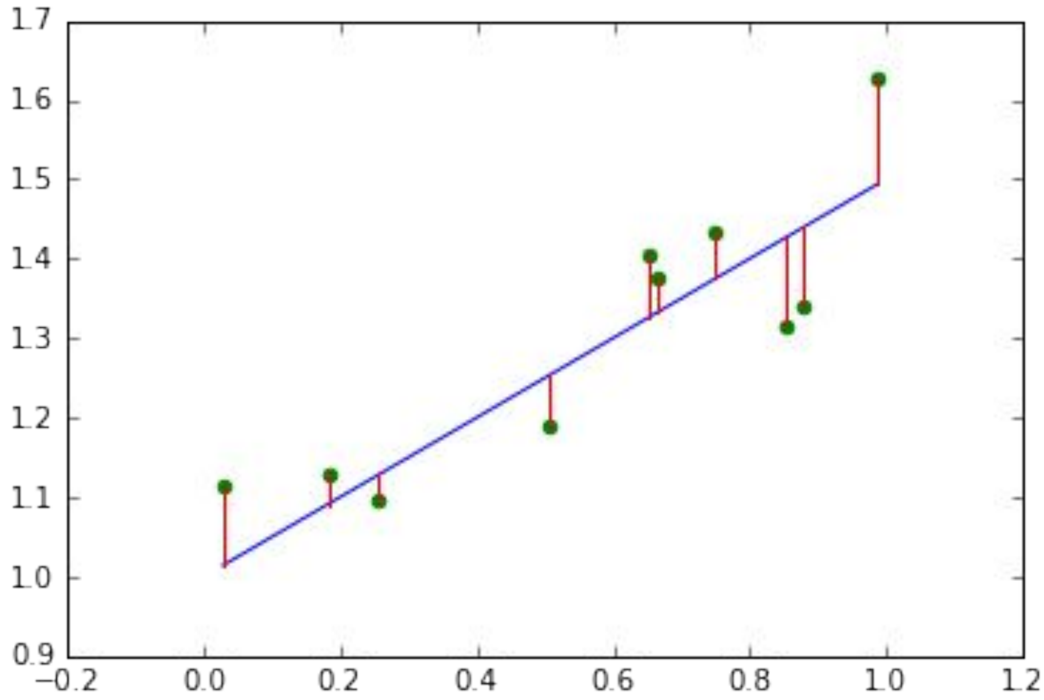
  Should be:



Note that there is **NO** randomness involved.

The Errors

- The randomness is attributed to ϵ.
  - The relation $Y = 1 + 0.5X + \epsilon$ becomes



The Basic Assumptions on Linear Regression

- The basic assumptions of a simple linear model are:
  - Linearity
  - Normality
  - Constant Variance
  - Independent Errors
- Violating any of these assumptions would lead to an inaccurate model.

  We will reflect upon the issue in the advanced linear regression lecture.

The Basic Assumptions on Linear Regression

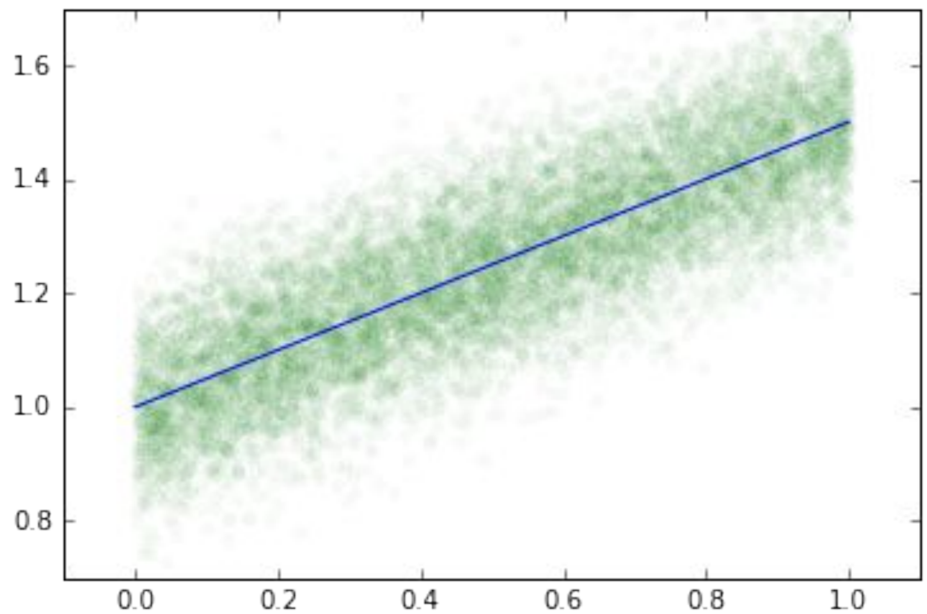
- Linearity:
  - Linearity defines the relation between *X* and *Y*. As we saw in the previous plot, it is represented by β_0 and β_1.
- We will discuss how these two constants are estimated.

The Basic Assumptions on Linear Regression

- Assumptions on the Errors:
  - We **cannot** estimate ϵ mainly because it is random. However, we can still study some properties of the randomness. The last three (except the linearity) assumptions on linear model describe what kind of randomness ϵ should be.

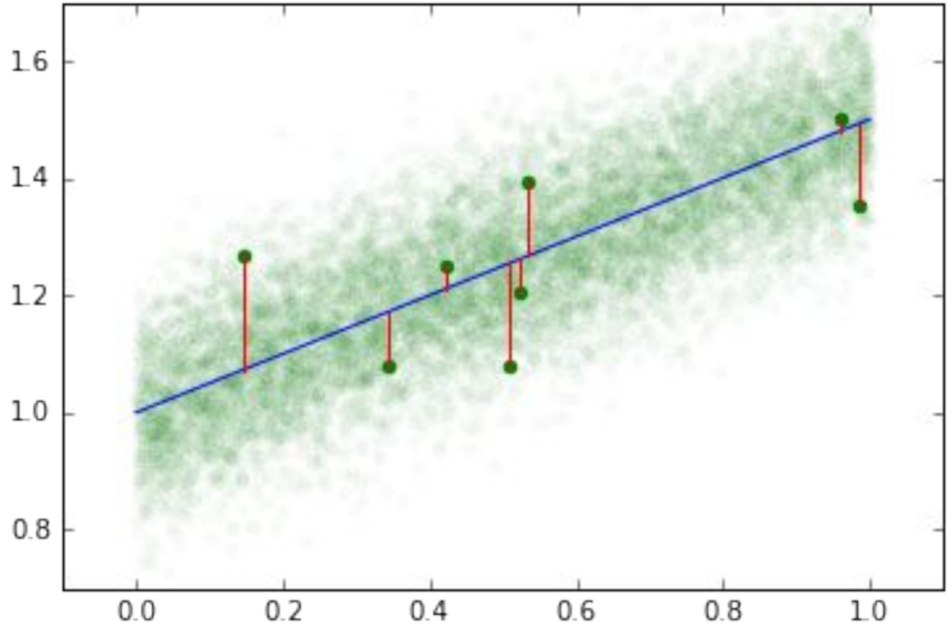
The Basic Assumptions on Linear Regression

- Normality:
  - Randomness is often described by a **probability distribution**, which can be seen only when we have a lot of samples. So let's create a much larger sample set:



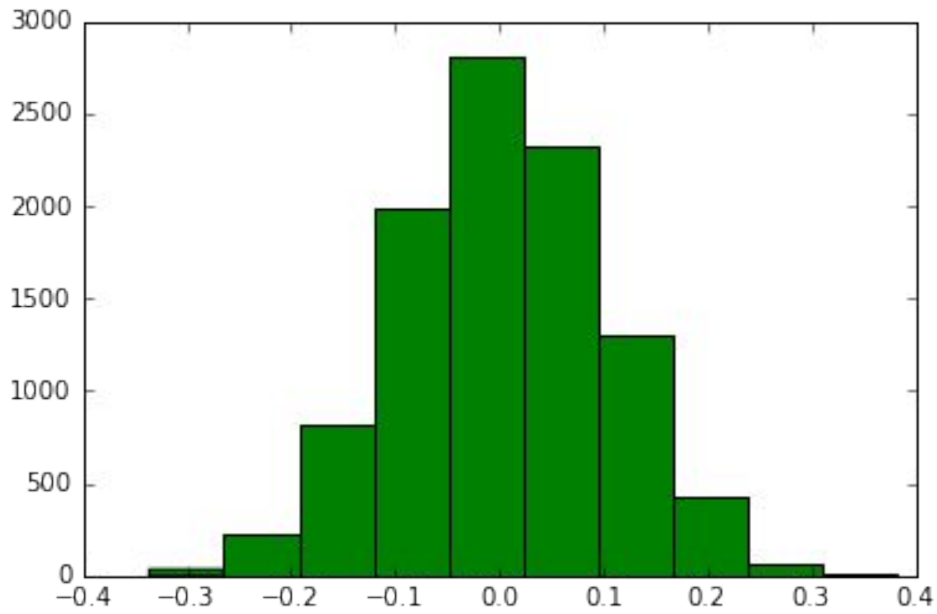
The Basic Assumptions on Linear Regression

- Normality:
  - And again we visualize the error with some of the *X*:



The Basic Assumptions on Linear Regression

- Normality:
  - Roughly speaking the normality assumption means that if we sketch the histogram of the errors, it looks like:

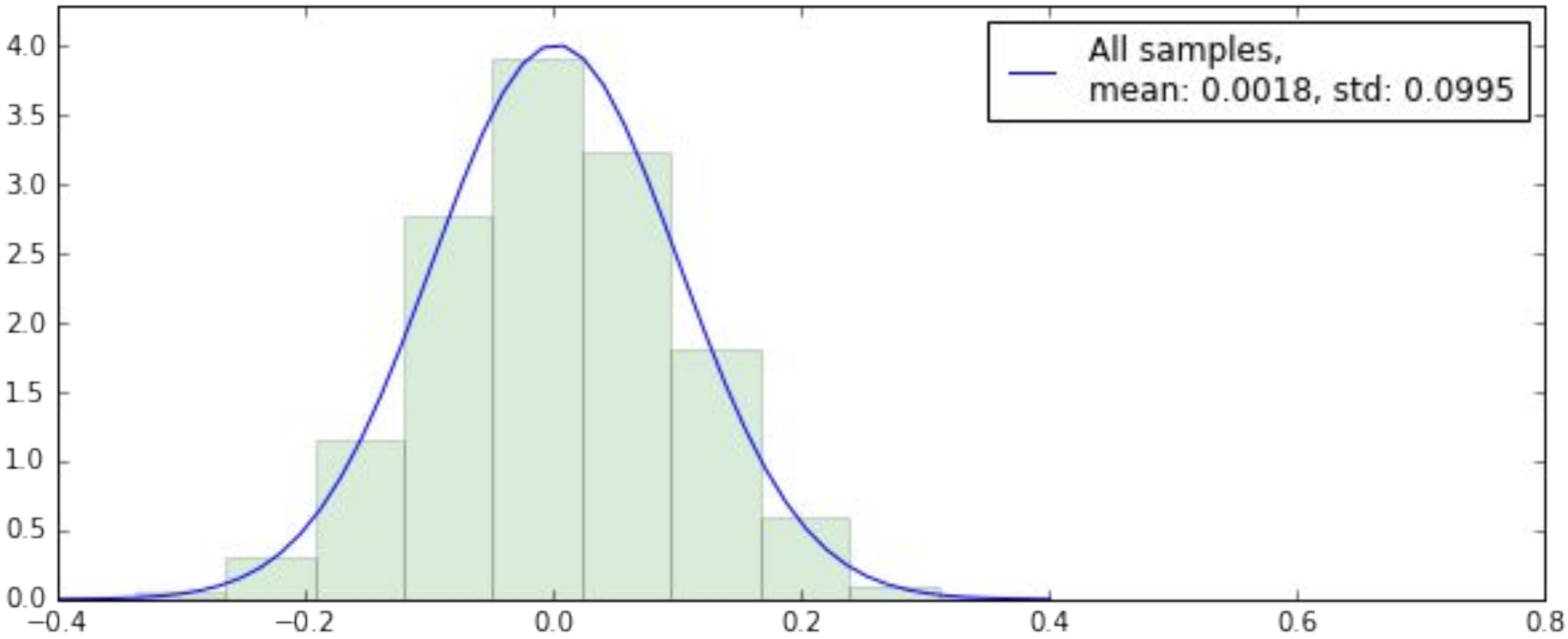


The Basic Assumptions on Linear Regression

- Normality:
  - Note that our errors have:
    - sample mean equals to 0.0018
    - sample standard deviation equals to 0.0995
  - Then we can compare the **normalized** histogram and the pdf curve of a normal distribution.
    - Note the difference between the ***y* axes** of the plot below and of the previous one.

The Basic Assumptions on Linear Regression

- Normality:



The Basic Assumptions on Linear Regression

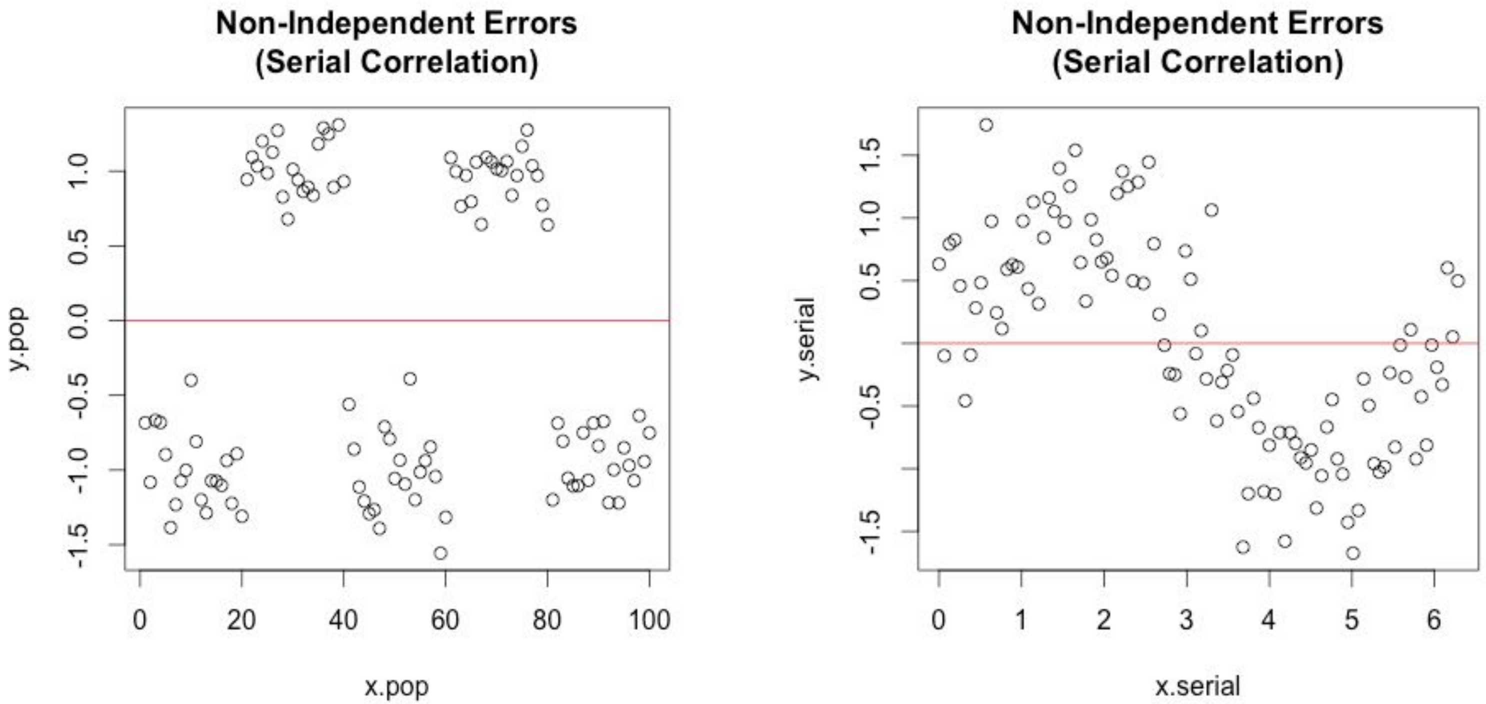
- Constant Variance and Independent Errors
- Both of the other two conditions:
  - **constant variance**
  - **independent errors**

  indicate that the error of each observation obeys the same distribution.

WHY?

The Basic Assumptions on Linear Regression

- Some examples of violating independent errors:



- There exist subgroups of errors distributing differently.

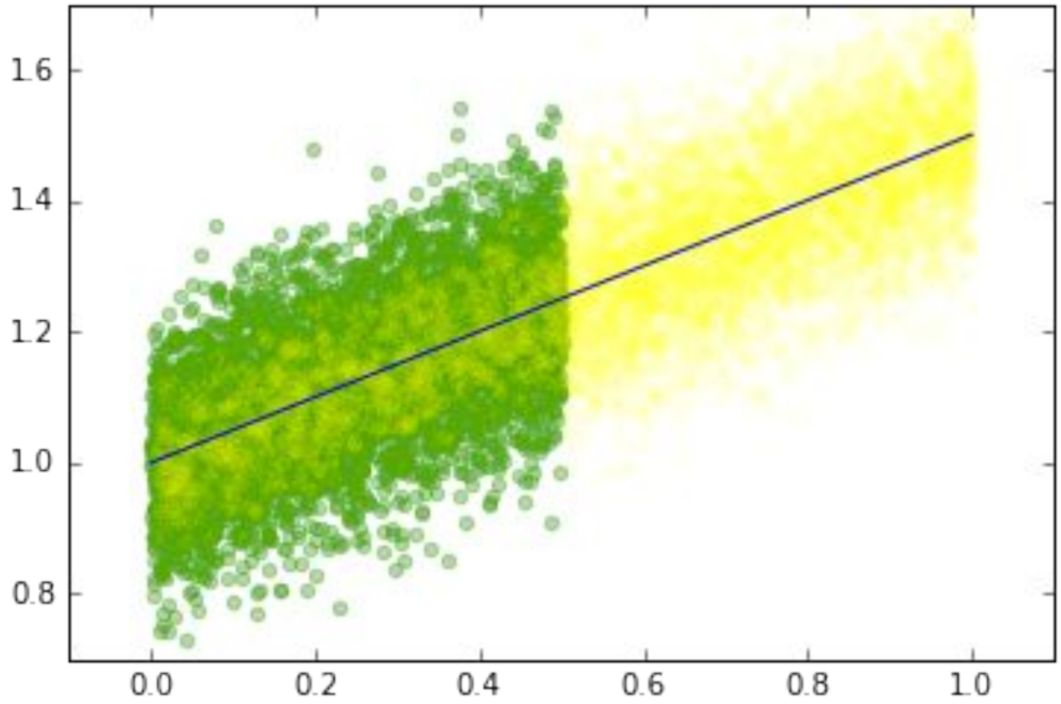
The Basic Assumptions on Linear Regression

- How do we check that?
  - There is no way to talk about the distribution of one observation. However, if all the errors obey the same distribution, we should obtain the same (or very similar) pdf (probability density function) plot when we randomly choose a (large enough) subset from the observations.

- For example, let's pick the observations with X less than 0.5 and compare the pdf curve obtained from the sub-samples with the one obtained from all the observations.

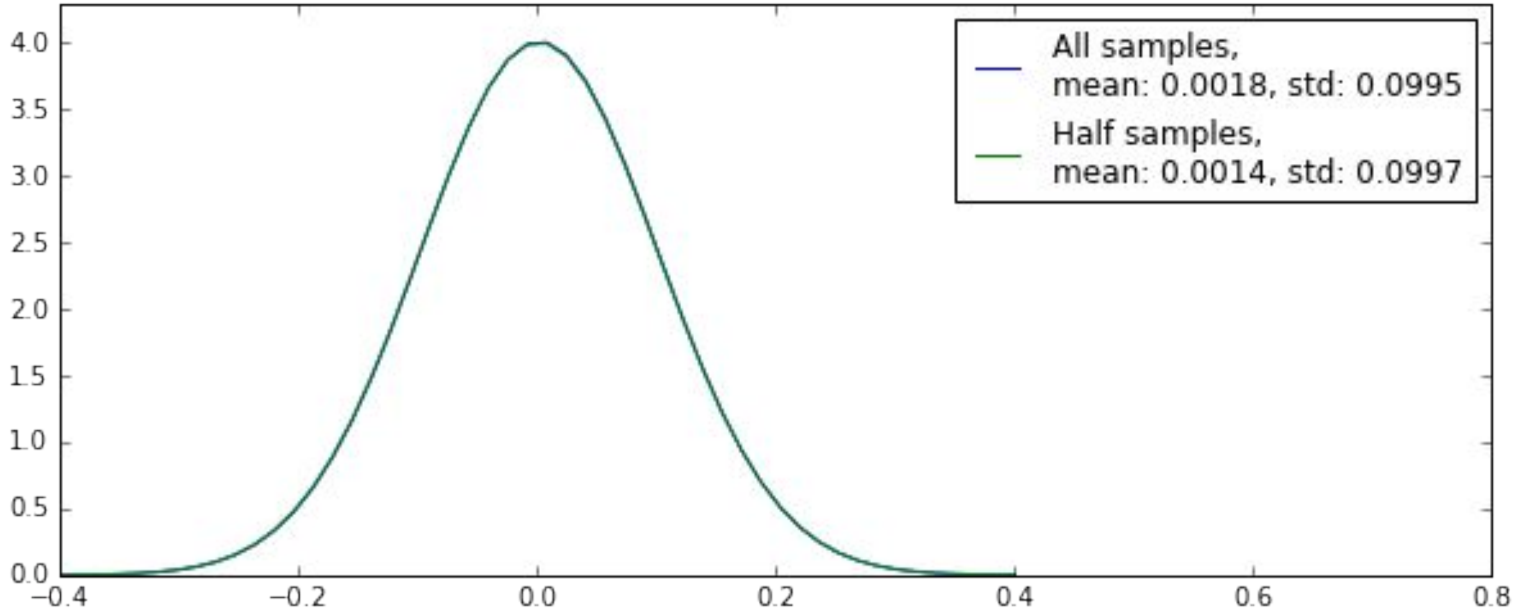
The Basic Assumptions on Linear Regression

- These are the samples we select:



The Basic Assumptions on Linear Regression

- Then we compare the pdfs of the normal distributions obtained from different groups:



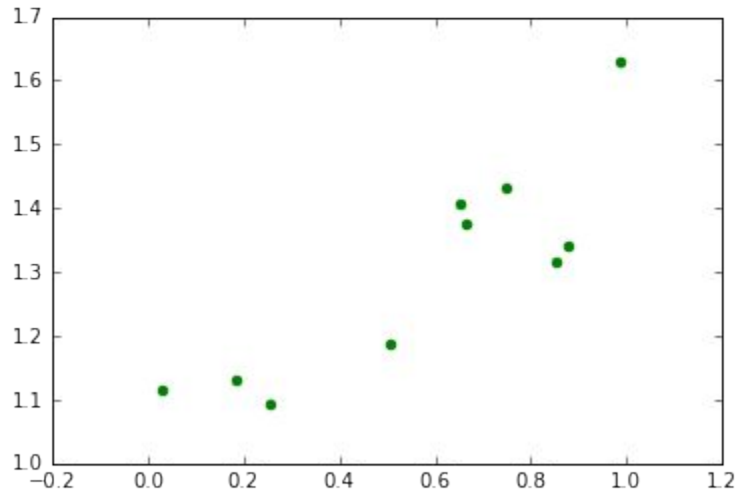
- You can go to the lecture code, change the range selected and compare the normal curves obtained.

Outline

- What is Machine Learning
- Introduction to Scikit-Learn
- Simple Linear Regression
  - **Estimating Coefficients**
  - Coefficient of Determination

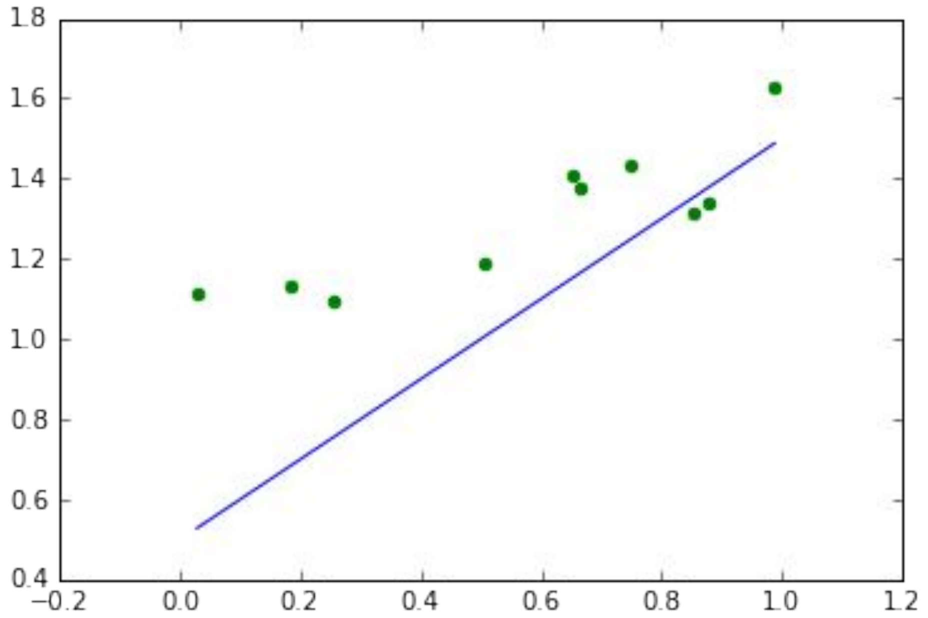
The Basic Assumptions on Linear Regression

- In general, β_0 and β_1 are unknown, what we have is a set of observations X and Y. Essentially what we do is to **try** all the possible pairs of β_0 and β_1, and find the one defining the linear model most similar to the observations.
  - We again illustrate the process with visualization.



Estimating the Coefficients

- We then start trying out some pair of $(\tilde{\beta}_0, \tilde{\beta}_1)$, say, (0.5, 1):



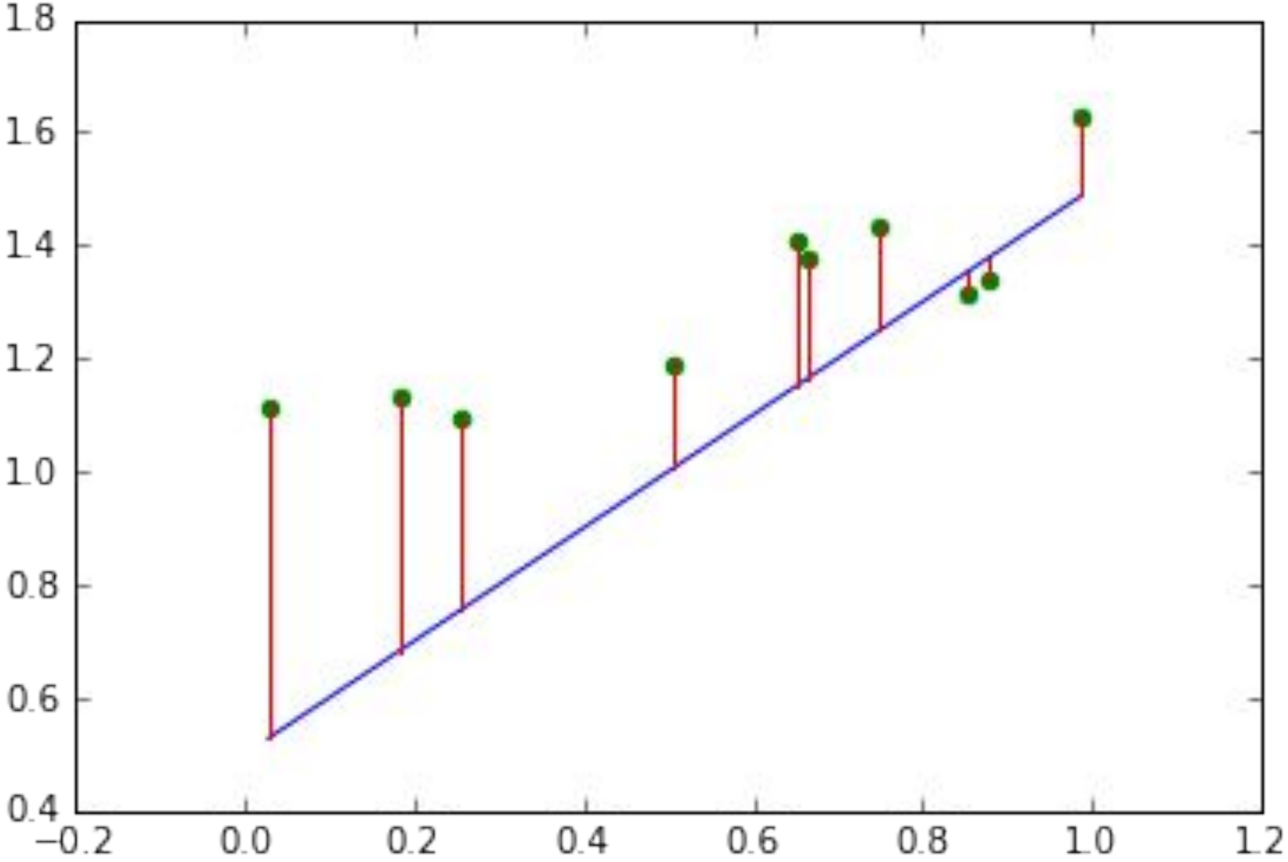
Estimating the Coefficients

- How different is the model from the observations?
  - We may again consider the difference between the observation and the model predicted value:

$$e = Y - (\tilde{\beta}_0 + \tilde{\beta}_1 X)$$

- This difference vector (in the space of samples) is called the residual.

Estimating the Coefficients



Estimating the Coefficients

- To quantify the difference between the model and the observations, we use the residual sum of squares, or RSS. It is defined by:

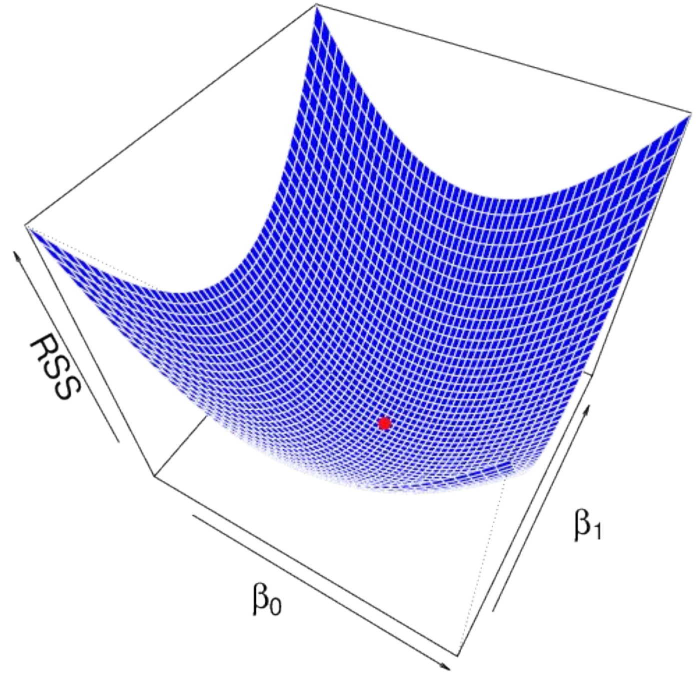
$$\text{Denote } e = (e_1, e_2, e_3, \ldots, e_n)$$

$$\begin{aligned} RSS(\tilde{\beta}_0, \tilde{\beta}_1) &= e_1^2 + e_2^2 + \ldots + e_n^2 \\ &= \sum_{i=1}^{n} (y_i - \tilde{\beta}_0 - \tilde{\beta}_1 x_i)^2 \end{aligned}$$

- Therefore RSS depends explicitly on $(\tilde{\beta}_0, \tilde{\beta}_1)$

Simple Linear Regression - RSS and Least Squares

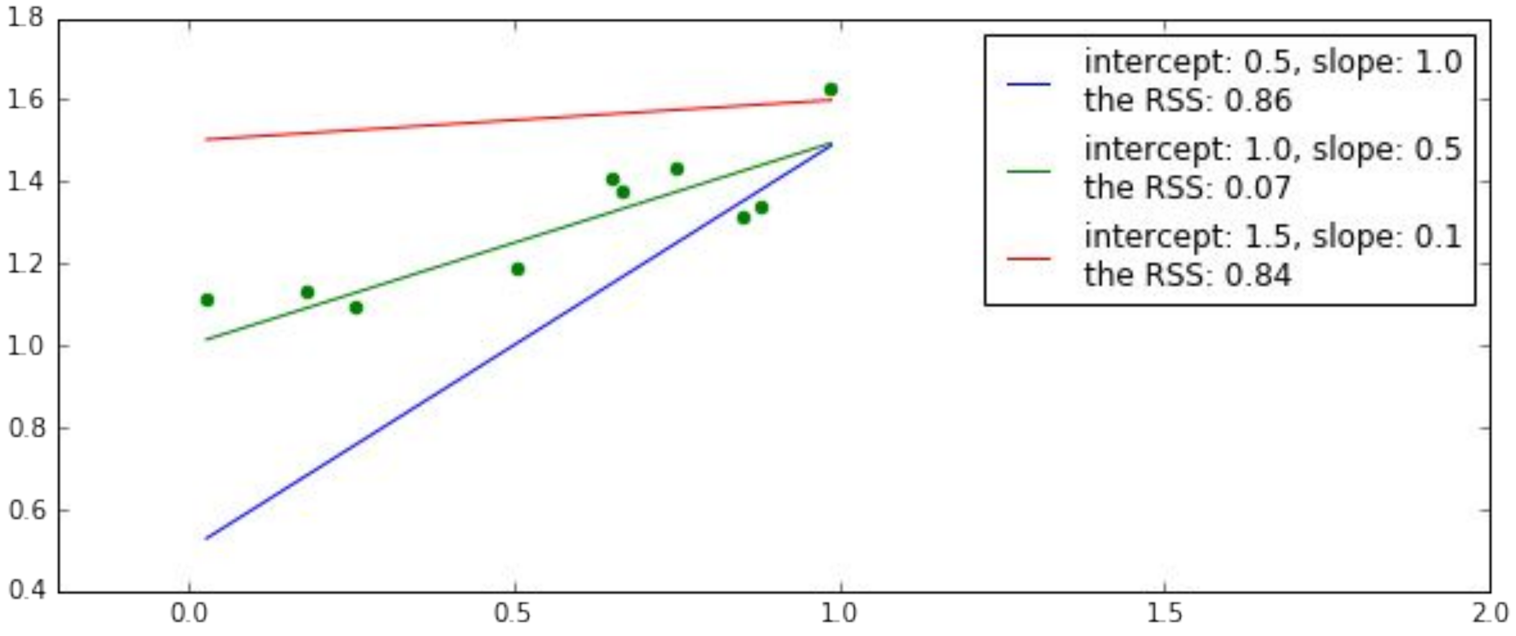
- As you observed, the smaller the **RSS** is, the better the fit is.
- **RSS** is a quadratic function of parameters β_0 and β_1.



- Our goal now becomes to find the minimum point of **RSS**(β).

Estimating the Coefficients

❖ Below we see that indeed the model with least **RSS** is most similar (visually) to the observations.



Estimating the Coefficients

QUESTION

- $(\tilde{\beta}_0, \tilde{\beta}_1) = (1, 0.5)$ is the best among the three. Is it actually the best possible pair among all?

- The coefficients that really minimize RSS is denoted by $(\hat{\beta}_0, \hat{\beta}_1)$, And the model is denoted by:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$$

- The symbol ^ denotes an estimated value based on the given samples. The coefficients are called the **ordinary least square estimator** (**OLS**). Once we have the estimators and a newly observed X, the Y can be predicted by passing X into the formula above.

Estimating the Coefficients

- Minimizing the RSS characterizes the coefficients $(\hat{\beta}_0, \hat{\beta}_1)$. We will not discuss how they can be obtained, though it is actually the standard optimization problem with differentiation. $(\hat{\beta}_0, \hat{\beta}_1)$ actually admits a closed form:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

where $\bar{x}$ and $\bar{y}$ are the sample means of x and y, respectively.

- Of course, if you don't care about math, **scikit-learn** will compute the coefficients for us.

Estimating the Coefficients

- The second equation indicates that $(\overline{x}, \overline{y})$ always lies on the regression line

- The first formula on $\hat{\beta}_1$ means $\hat{\beta}_1$ is equal to $\frac{cov(x,y)}{var(x)}$.

Simple Linear Regression: Mathematically

- Task: find the estimates of β_0 and β_1 that reduce the sum of the squared vertical distances from the observations to the regression line (i.e., the **RSS**) as much as possible.

- Procedure: derive formulas for these estimates using basic calculus.

Simple Linear Regression: Mathematically

- Procedure for the intercept coefficient estimate:

$$RSS = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

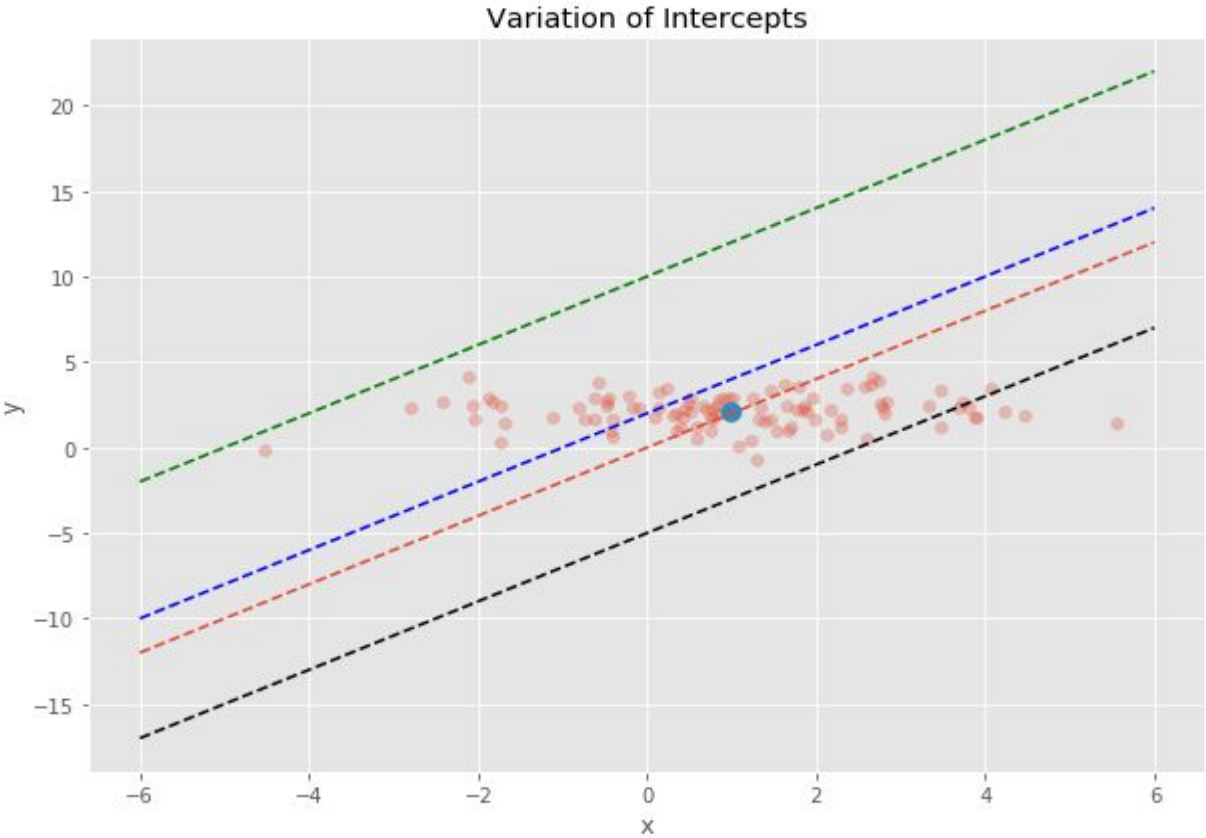
$$\frac{\partial RSS}{\partial \hat{\beta}_0} = \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)(2)(-1) \overset{!}{=} 0$$

$$\Rightarrow n\hat{\beta}_0 = \sum_{i=1}^{n} (y_i - \hat{\beta}_1 x_i)$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

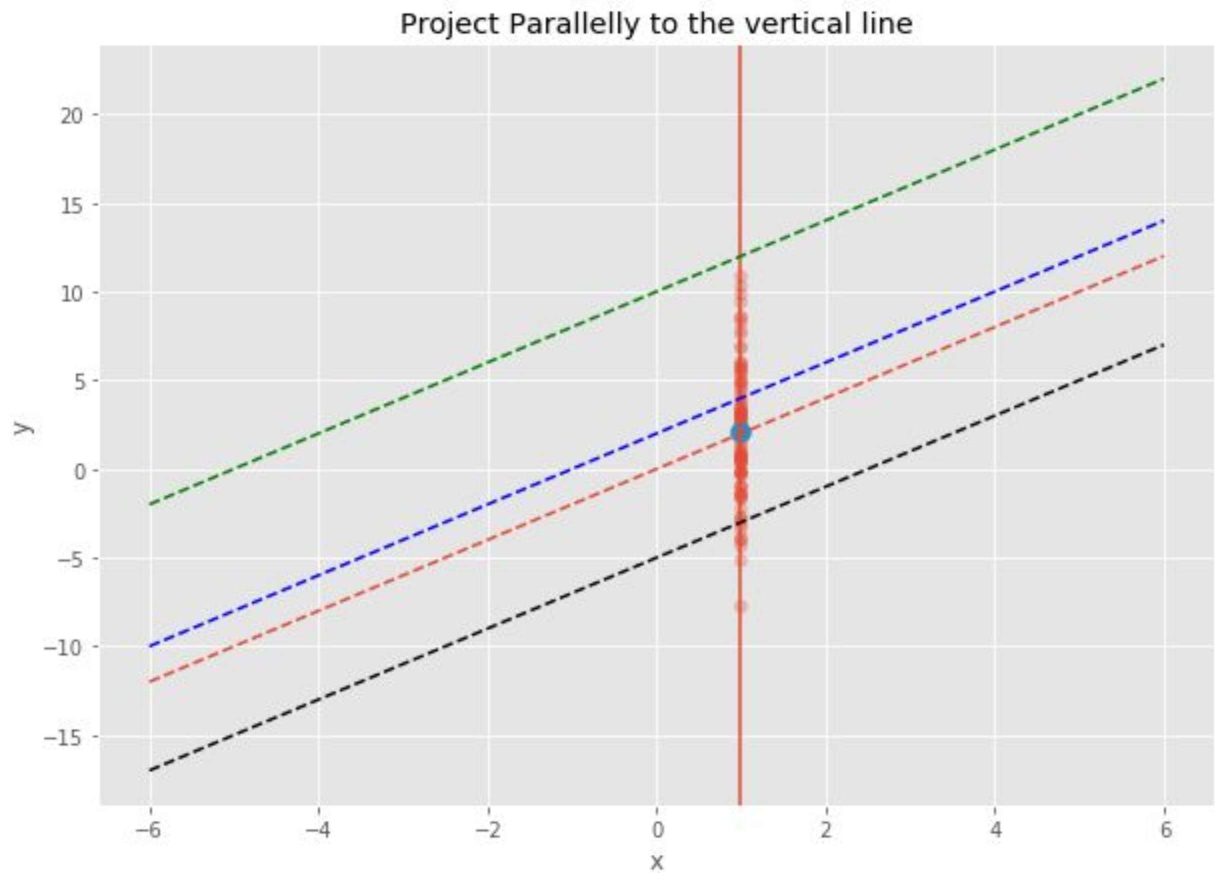
The Formula of the Intercept: Geometry

- Geometrically we vary the intercepts of the line fixing their common slope
- The line which minimizes **RSS** has to pass through the center of the data



Why Choosing the Center?

- We notice that the **RSS** computation involves a sum of vertical distances
- We can parallel-project the scatter plot to the red vertical line without affecting the answer



Simple Linear Regression: Mathematically

- Procedure for the intercept coefficient estimate (continued):

$$\frac{\partial^2 RSS}{\partial \hat{\beta}_0^2} = \sum_{i=1}^{n} (-1)(2)(-1)$$
$$= 2n$$

- Because the second derivative is necessarily positive, we know that this estimate is a minimum.

Simple Linear Regression: Mathematically

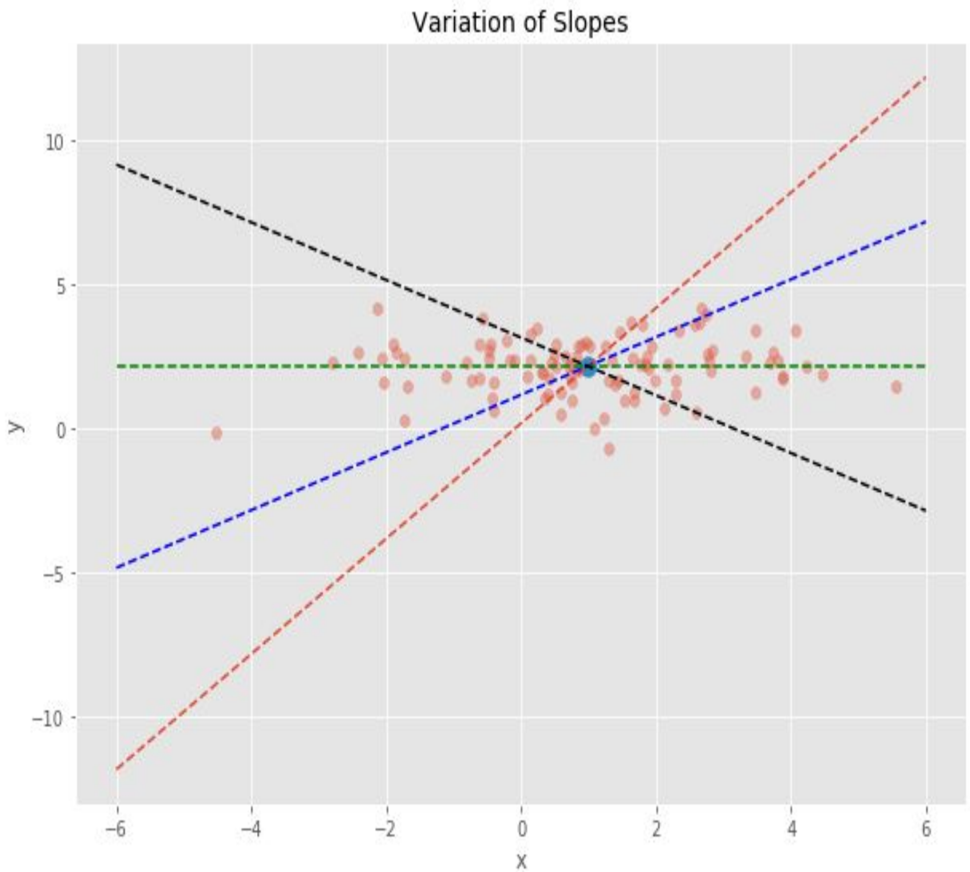
- Procedure for the slope coefficient estimate:

$$RSS = \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$
$$= \sum_{i=1}^{n}(y_i - (\bar{y} - \hat{\beta}_1 \bar{x}) - \hat{\beta}_1 x_i)^2$$
$$= \sum_{i=1}^{n}(y_i - \bar{y} - \hat{\beta}_1 (x_i - \bar{x}))^2$$
$$\frac{\partial RSS}{\partial \hat{\beta}_1} = \sum_{i=1}^{n}(y_i - \bar{y} - \hat{\beta}_1 (x_i - \bar{x}))(2)(-1)(x_i - \bar{x}) \stackrel{!}{=} 0$$
$$\Rightarrow \hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- The above formula is an estimation of $\frac{cov(x,y)}{var(x)}$

The Formula of the Slope: Geometry

- Knowing that the optimal line has to pass through the center of the data, we vary the slopes. It turns out that after suitable normalization of both x and y, the best slope is nothing but the sample correlation!



Simple Linear Regression: Mathematically

- Procedure for the slope coefficient estimate (continued):

$$\frac{\partial RSS}{\partial \hat{\beta}_1} = -2\sum_{i=1}^{n}(y_i - \bar{y} - \hat{\beta}_1(x_i - \bar{x}))(x_i - \bar{x})$$

$$\frac{\partial^2 RSS}{\partial \hat{\beta}_1^2} = -2\sum_{i=1}^{n}(x_i - \bar{x})(-1(x_i - \bar{x}))$$

$$= 2\sum_{i=1}^{n}(x_i - \bar{x})^2$$

- Because the second derivative is necessarily positive, we know that this estimate is a minimum.
- The second derivative measures whether a function is concave up or down!

Accuracy of the Coefficient Estimates

- Under the simple linear regression model, β_0 and β_1 exist in the universe as theoretical, true parameter values; however, we can only calculate an estimate based on our data. How can we quantify the accuracy of our estimates?

- We investigate their standard errors, which yield an approximation of how much our estimates vary from the true parameter values:

$$\widehat{SE}(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right]$$

$$\widehat{SE}(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- Notice that the estimate of the standard error for the slope gets smaller as the spread of our observations increases; this allows us to better gauge its leverage.

The Meaning of $\widehat{SE}$

- Both of the $\widehat{SE}$ estimate the plausible ranges in which the true regression coefficients should lie within
- By ‘true regression coefficients’, we mean either
  - The **hidden** coefficients which generate the linear related data
  - The model coefficients when the sample size approaches infinity
- Later we refer to the sum of $\widehat{SE}^2$ as the **model variance** of the simple linear model
- The concept of **model variance** captures the sensitivity of the model on the chosen sub-dataset (viewing all the possible finite samples as particular samplings from the true population)
- A tight **model variance** indicates that the model trained on a dataset is reasonably close to its theoretical limit of infinite samples
- On the other hand, a very wide $\widehat{SE}$ indicates that the current estimation is not trustworthy

Accuracy of the Coefficient Estimates

- But wait...we don't know the value of σ^2! Use the residual standard error, which serves as our best estimate for σ:

$$\hat{\sigma} = RSE = \sqrt{\frac{RSS}{n-2}}$$

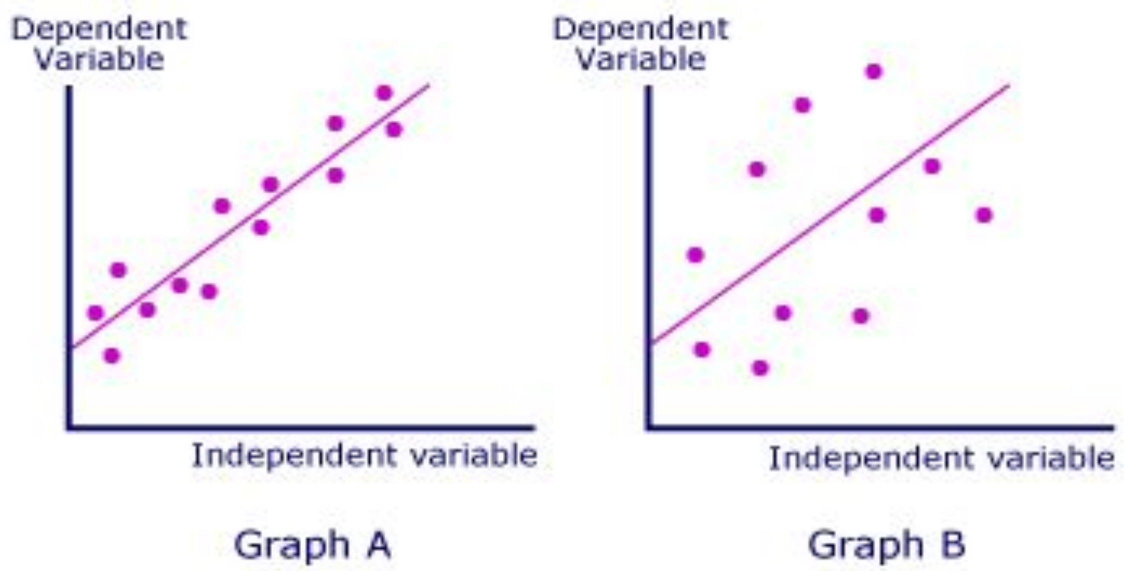
- Now that we can calculate the estimates of the standard errors, we can use them to assess the accuracy of our coefficient estimates by:
  - Performing hypothesis tests.
  - Constructing confidence intervals.

Outline

- What is Machine Learning
- Introduction to Scikit-Learn
- Simple Linear Regression
  - Estimating Coefficients
  - **Coefficient of Determination**

Coefficient of Determination

- Once we fit a linear model, how should we assess the overall accuracy of the model?
- Think about the two graphs shown below. They have the same fitted parameters, but the left graph shows a higher descriptive quality than the right one.



Coefficient of Determination

- The usual way to measure the overall accuracy of a simple linear model is to use the ***coefficient of determination***.
- The coefficient of determination, denoted R^2, measures how well data fits a model (how well the model describes the data).
- R^2 is defined as

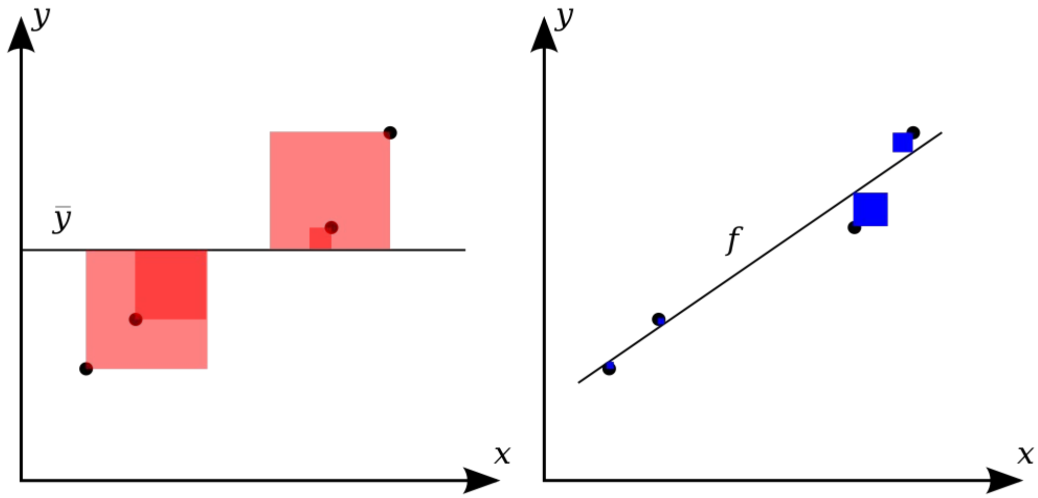
$$R^2 = 1 - \frac{RSS}{TSS}$$

where *TSS* is the total sum of squares, which measures the total variance of the output data y:

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Coefficient of Determination

- ***RSS*** (the areas of the blue squares) represents the squared residuals with respect to the linear regression.
- ***TSS*** (the areas of the red squares) represents the squared residuals with respect to the average value and is fixed if data is known. (Can you tell why?)



- ***RSS*** cannot exceed ***TSS***

Source: https://en.wikipedia.org/wiki/Coefficient_of_determination.

Coefficient of Determination

- Given a dataset, ***TSS*** is completely determined (i.e. model independent) and the fitted linear model has the minimum ***RSS***. Therefore:
  - R^2 = 1 indicates that the regression line perfectly fits the data.
  - R^2 = 0 indicates that the line does not fit the data at all.
  - **R^2 always lies between 0 and 1**
  - In general, the better the linear regression fits the data in comparison with the simple average (the null model), the closer the value of R^2 is to 1.

Hands-on Session

- Please go to the **"Linear Regression in Scikit-Learn"** in the lecture code.