

Seidenberg School Of Computer Sciences | Pace University

Introduction to Linear Regression

Predictions Using Linear Regression Models

Descriptive, Predictive, Prescriptive Analytics

Prof. Tassos H. Sarbanes

CS675 --- Spring 2022

What is Regression Analysis? (In Business)

Suppose you are a Sales Manager and trying to **predict** sales for next month.

There are most likely many **factors (variables)** that could determine that!

Regression analysis is a way **of mathematically** sorting out which of those **variables** does indeed **have an impact**.

It answers the questions:

Which factors matter most?

Which can we ignore?

How do those factors interact with each other?

And, perhaps most importantly, how certain are we about all of these factors?

We have the **dependent variable** [**$Y: (y_1, y_2, \dots, y_k)$**]

— the main factor that you're trying to understand or predict, in our case above, the dependent variable is monthly sales.

And then we have the **independent variable(s)** [**$X: (x_{i1}, x_{i2}, \dots, x_{ik})$**]

— the factors you suspect have an impact on the dependent variable.

How does it work?

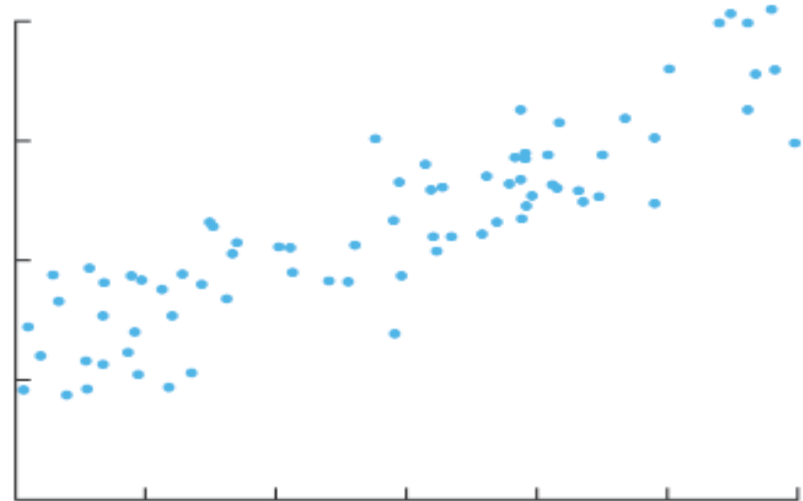
In order to **conduct a regression analysis**, you **gather the data** on the variables in question.

You take all of your monthly sales numbers for, say, the past three years and any data on the independent variables you're interested in. So, in this case, let's say you find out the average monthly rainfall for the past three years as well.

Then you plot all of that information on a chart that looks like this:

Is There a Relationship Between These Two Variables?

Plotting your data is the first step in figuring that out.



Build the Model – Find the Relationship

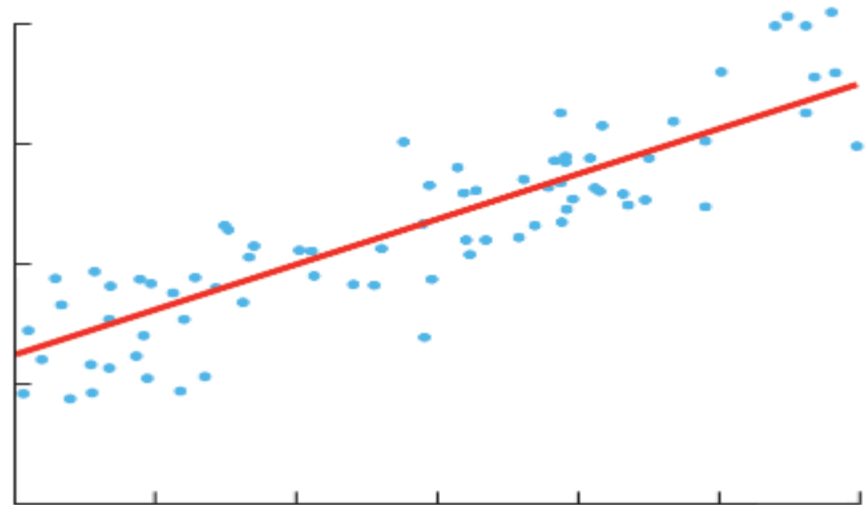
The **y-axis** is the amount of sales (the **dependent variable**, the thing you're interested in, is always on the y-axis) and the x-axis is the total rainfall. Each blue dot represents one month's data—how much it rained that month and how many sales you made that same month.

Glancing at this data, you probably notice that sales are higher on days when it rains a lot. That's interesting to know, but by how much? If it rains 3 inches, do you know how much you'll sell? What about if it rains 4 inches?

Now imagine drawing a line through the chart, one that runs roughly through the middle of all the data points. This line will help you answer, with some degree of certainty, how much you typically sell when it rains a certain amount.

Building a Regression Model

The line summarizes the relationship between x and y.



A Note about Correlation

Most companies use regression analysis to:

- **Explain a phenomenon** they want to understand (e.g. why did customer service calls drop last month?); [[Descriptive Analytics](#)]
- **Predict things** about the future (e.g. what will sales look like over the next six months?); [[Predictive Analytics](#)]
- **Decide what to do** (e.g. should we go with this promotion or a different one?). [[Prescriptive Analytics](#)]

“Correlation is not Causation”

Whenever you work with regression analysis or any other analysis that tries to **explain the impact of one factor on another**, you need to remember the important adage: **Correlation is not causation**.

This is **critical** and here's why: It's easy to say that there is a correlation between rain and monthly sales. The regression shows that they are indeed related. But it's an entirely different thing to say that rain *caused* the sales. Unless you're selling umbrellas, it might be difficult to prove that there is **cause and effect**.

Descriptive / Inferential Statistics (Definitions)

Descriptive Statistics

- Statistical procedures used to summarize, organize, and simplify data

Inferential Statistics

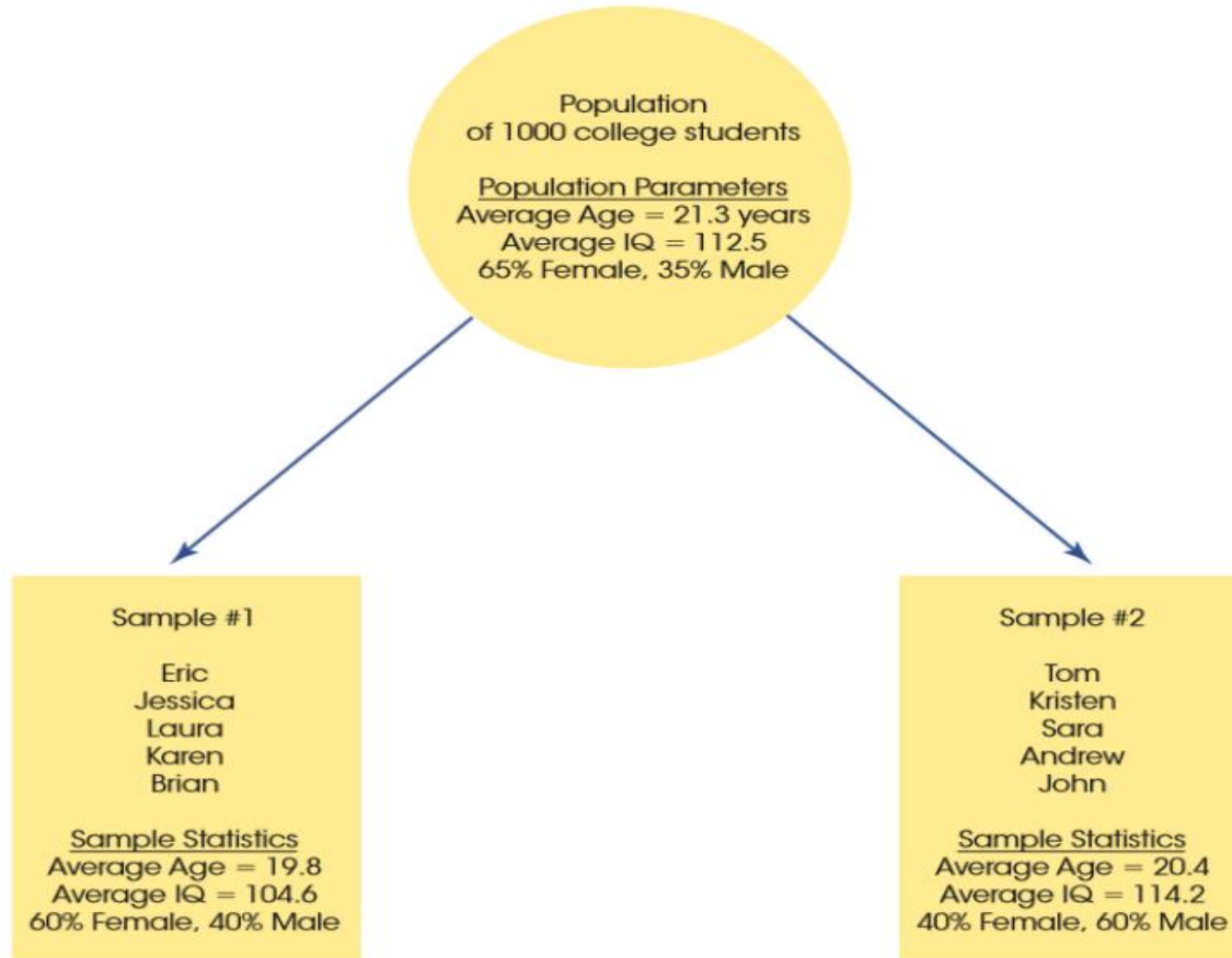
- Techniques allow to study samples and make generalizations about the populations from which they were selected.

Sampling Error

- Naturally occurring discrepancy, or error, that exists between a **sample** statistic (*) and corresponding **population** parameter

(*) A value, usually numeric, that describes a sample. Usually derived from measurements of the individuals in the sample.

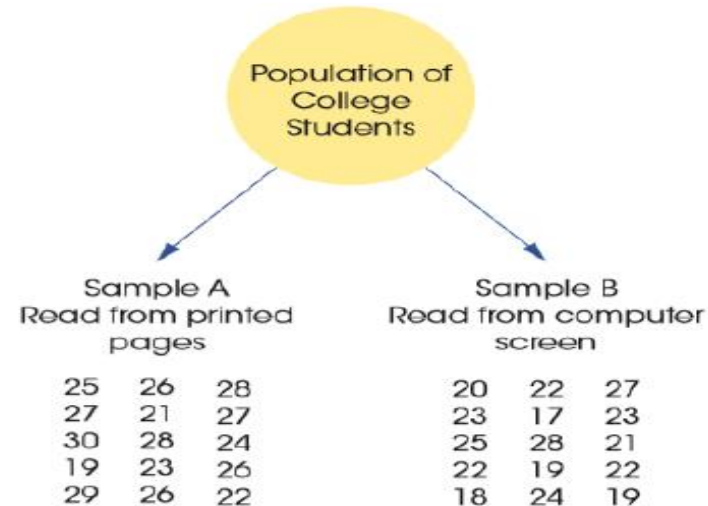
Example of Sampling Error



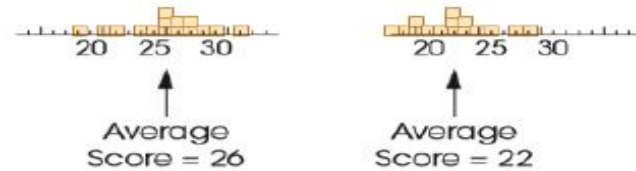
The Role of Statistics in Experimental Research

Step 1
Experiment:
Compare two
studying methods

Data
Test scores for the
students in each
sample



Step 2
Descriptive statistics:
Organize and simplify



Step 3
Inferential statistics:
Interpret results

The sample data show a 4-point difference between the two methods of studying. However, there are two ways to interpret the results.

1. There actually is no difference between the two studying methods, and the sample difference is due to chance (sampling error).
2. There really is a difference between the two methods, and the sample data accurately reflect this difference.

The goal of inferential statistics is to help researchers decide between the two interpretations.

FIGURE 1.3

The role of statistics in experimental research.

Statistical Approaches / Methods

Some research studies are conducted simply to describe **individual variables** as they exist naturally.

However, most research is intended to examine **relationships between two (2) or more variables**.

In order to determine whether a relationship exists between two (2) variables, we must observe/measure the variables by implementing **statistical methods**.

The Correlation Method

The **Correlation Method** observes two (2) different variables to determine whether there is a relationship between them

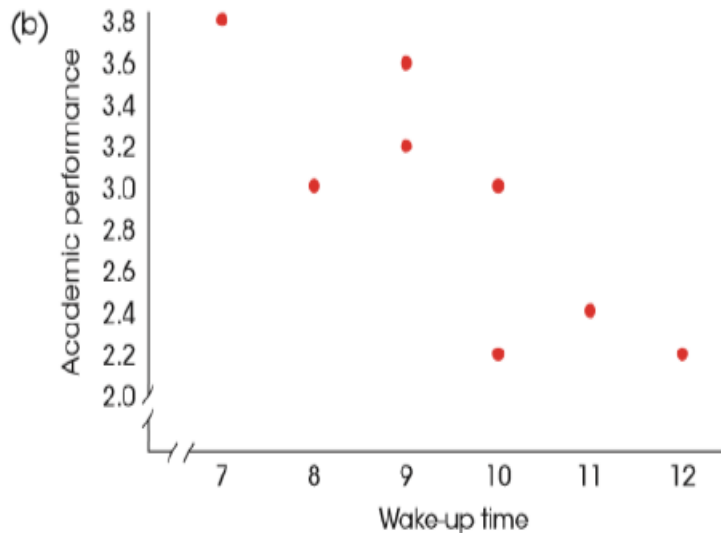
- When the data consists of numerical scores, the relationship between the variables is usually measured and described using a statistics called a **correlation**
- If the measurement process simply classifies individual elements into categories that do not correspond to numerical values, the **chi-square (*) test** is used

(*) One of the methods for Decision Tree Splitting Algos

Correlations – Examples

(a)

Student	Wake-up Time	Academic Performance
A	11	2.4
B	9	3.6
C	9	3.2
D	12	2.2
E	7	3.8
F	10	2.2
G	10	3.0
H	8	3.0



Cell Phone Preference		
	Text	Talk
Males	30	20
Females	25	25

50

50

Limitation of Correlation

The results from a correlational study can demonstrate the existence of a relationship between two variables, but they do not provide an explanation for the relationship.

i.e. **CORRELATION IS NOT CAUSATION**

<Q> What is the 'solution' for it?

<A> See next slide...

The Experimental Method

Examines the relationship between variables by using one of the variables to define the groups, and then measuring the second variable to obtain scores for each group.

- The results from an experiment allow a **cause-and-effect explanation**
- A **nonexperimental** study does not permit a **cause-and-effect explanation**

What is Regression Analysis? (In Statistics)

Regression is arguably the **workhorse** of statistics.

Despite its popularity, however, it may also be the most misunderstood. Why?

There is no such thing as Regression.

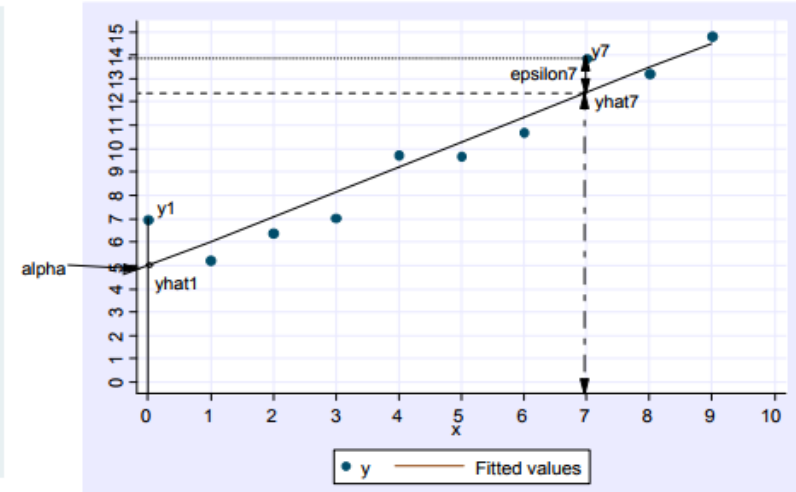
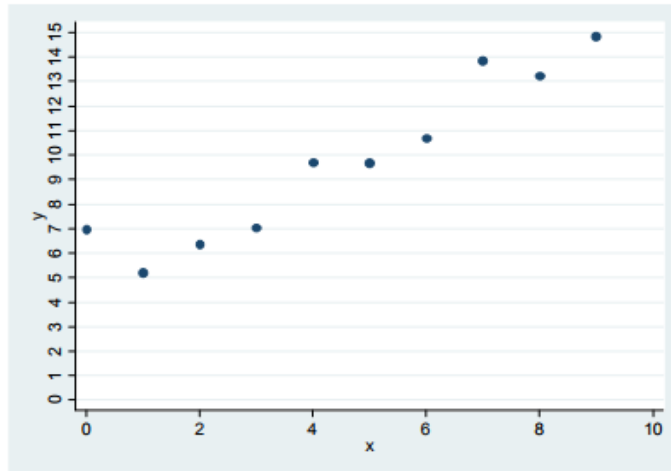
Rather, there are a **large number of statistical methods** that are called Regression or grounded on its **fundamental idea**

Dependent Variable = Constant + Slope*Independent Variable + Error

$$y_i = \alpha + \beta * x_i + \varepsilon_i \quad \hat{y}_i = \hat{\alpha} + \hat{\beta} * x_i + \hat{\varepsilon}_i \Rightarrow \hat{\varepsilon}_i = y_i - (\hat{\alpha} + \hat{\beta} * x_i)$$

Epsilon

Random component (noise) of the linear relationship between x and y.



Regression Analysis Formula: $Y = b + mX + e$

The **Dependent Variable (Y)** is something you want to predict or explain. For example the price of a house.

The **Independent Variable (X)** is what you use to explain or predict the Dependent Variable. For instance the # of rooms in a house.

The **Constant (b)** term in the equation above may be more familiar to you as the Y intercept; by convention, Y is used to represent the Dependent Variable and X the Independent Variable.

The **Slope (m)** is amount Y changes when X changes by a certain amount.

The **Error(e)** term is very important for many reasons, one being that it reminds us that we can seldom, if ever, predict Y from X exactly. When we can, it suggests that Y and X may be the same thing or that there is an error in our data! The pattern of errors is also a tip-off as to how trustworthy our model is and how to make it better.

Multiple Regression Analysis -- ANOVA

The formula at the previous slide was for **Simple Regression**.

When we have **more than one Independent Variable** - sometimes also called a Predictor or a Covariate - it becomes **Multiple Regression**.

Multiple Regression is more widely used than Simple Regression since a single Independent Variable can usually only show us part of the picture!

Analysis of Variance (ANOVA): statistical method to test differences between two or more group means and their associated procedures (such as “variation”), it was developed by Sir Ronald Fisher. Seem odd that technique is called "Analysis of Variance" rather than "Analysis of Means." Name is appropriate because **inferences about means are made by analyzing variance**. In its simplest form, ANOVA provides a statistical test of whether or not the means of several groups are equal.

ANOVA: statistical tool used in ways to develop, confirm explanation for observed data.

Experiments where the **effects** of more than **one factor (variable)** considered together are called '**factorial experiments**' and are analyzed with the use of **factorial ANOVA**. One-Way ANOVA, Two-Way ANOVA [there are one, two factors (ind. Variable), etc...]

Research Question such as the below could be answered by Factorial ANOVA:

Is there a **statistically significant difference** on dependent variable by independent var-1 and var-2?

ANalysis Of VAriance -> ANOVA

Variance is a statistical term introduced by Sir **Ronald Fisher**. It is the expectation of the squared deviation of a random variable from its mean. It measures how far a set of (random) numbers are spread out from their average value. Variance has a central role in statistics, it is often σ^2 , s^2 , or $\text{Var}(X)$. by

Variance of random variable X is the **expected value of the squared deviation from the mean** of X

$$\mu = E[X] \longrightarrow \text{Var}(X) = E[(X - \mu)^2]$$

Variance can also be thought of as the **covariance** of a random variable with itself:

$$\text{Var}(X) = \text{Cov}(X, X).$$

ANOVA synthesis of several ideas, used for multiple purposes, it does 3 things at once:

- 1) As exploratory data analysis (**EDA**), an ANOVA is an organization of an additive data decomposition, and its sums of squares indicate the variance of each component of the decomposition (or, equivalently, each set of terms of a linear model).
- 2) **Comparisons of mean squares, along with an F-test ... (in honor of Ronald Fisher).**
- 3) **Closely related to ANOVA is a linear model fit** w/ coefficient estimates & standard errors

Additionally:

- 4) It is computationally elegant and relatively robust against violations of its assumptions.
- 5) ANOVA provides industrial strength (multiple sample comparison) statistical analysis.
- 6) It has been adapted on the analysis of a variety of experimental designs.

Analysis of Variance -- Continuous & Discrete Variables

Continuous random variable

If the random variable X represents samples generated by a **continuous distribution** with **probability density distribution** $f(x)$, then the population variance is given by:

$$\begin{aligned}\text{Var}(X) = \sigma^2 &= \int (x - \mu)^2 f(x) dx \\ &= \int x^2 f(x) dx - 2\mu \int x f(x) dx + \int \mu^2 f(x) dx \\ &= \int x^2 f(x) dx - \mu^2,\end{aligned}$$

where μ is the expected value of X given by

$$\mu = \int x f(x) dx,$$

Discrete random variable

If the generator of random variable X is **discrete (categorical)** with **probability mass function**: $x_1 \rightarrow p_1, x_2 \rightarrow p_2, \dots, x_n \rightarrow p_n$ { $\text{Var}(X) = \sum_{i=1}^n p_i \cdot (x_i - \mu)^2$ or equivalently

$$\text{Var}(X) = \sum_{i=1}^n p_i x_i^2 - \mu^2,$$

where μ is the average value, i.e.

$$\mu = \sum_{i=1}^n p_i \cdot x_i.$$

Note: The variance of a set of n **equally likely values** can be written as

$$\text{Var}(X) = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2,$$

where μ is the expected value, i.e.,

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i.$$

Is ANOVA equivalent to Linear Regression?

Analysis of Variance consists of calculations that provide information about **levels of variability within a regression model** and form a basis for **tests of significance**.

ANOVA and linear regression are equivalent when the two models test against the same hypotheses and use an identical encoding. One can **describe ANOVA as a regression with dummy variables**. Regression (OLS: Ordinary Least Squares) with categorical (*dummy-coded*) 'regressors' are equivalent to the *factors* in ANOVA. In both cases there are levels (or groups in the case of ANOVA).

Regression & ANOVA Example --- in R

Motor Trend Car Road Tests

Format

A data frame with 32 observations on 11 variables.

- [, 1] mpg Miles/(US) gallon
- [, 2] cyl Number of cylinders
- [, 3] disp Displacement (cu.in.)
- [, 4] hp Gross horsepower
- [, 5] drat Rear axle ratio
- [, 6] wt Weight (lb/1000)
- [, 7] qsec 1/4 mile time
- [, 8] vs V/S
- [, 9] am Transmission (0 = automatic, 1 = manual)
- [, 10] gear Number of forward gears
- [, 11] carb Number of carburetors

R Console

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4

ANOVA Equivalent to Linear Regression

Run the Linear Regression model, for mtcars, in R:

```
> lm(mpg ~ wt + as.factor(cyl), data = mtcars)

Call:
lm(formula = mpg ~ wt + as.factor(cyl), data = mtcars)

Coefficients:
(Intercept)          wt  as.factor(cyl)6  as.factor(cyl)8
      33.991        -3.206        -4.256        -6.071
```

```
> mod <- lm(mpg ~ wt + as.factor(cyl), data = mtcars)
```

```
> summary(mod)
```

```
Call:
lm(formula = mpg ~ wt + as.factor(cyl), data = mtcars)
```

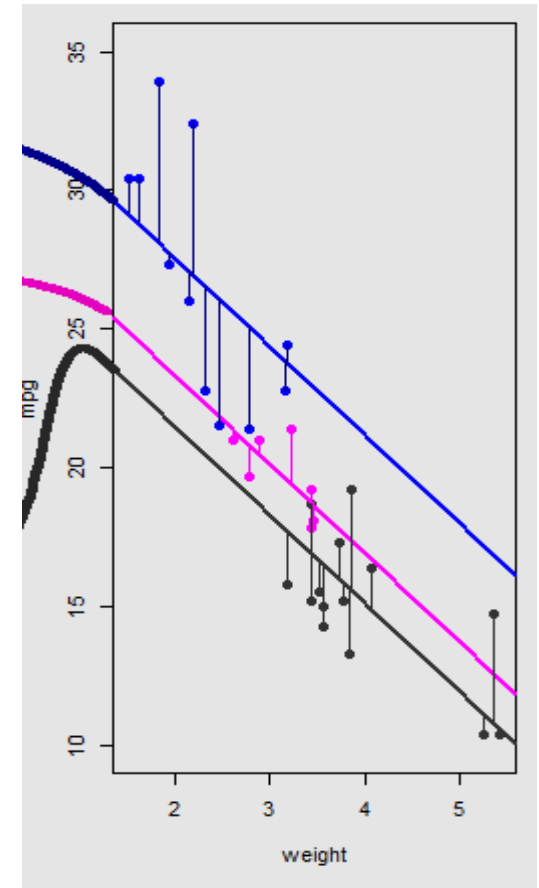
```
Residuals:
    Min       1Q   Median       3Q      Max
-4.5890 -1.2357 -0.5159  1.3845  5.7915
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   33.9908     1.8878  18.006 < 2e-16 ***
wt            -3.2056     0.7539  -4.252 0.000213 ***
as.factor(cyl)6 -4.2556     1.3861  -3.070 0.004718 **
as.factor(cyl)8 -6.0709     1.6523  -3.674 0.000999 ***
---

```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 2.557 on 28 degrees of freedom
Multiple R-squared:  0.8374,    Adjusted R-squared:  0.82
F-statistic: 48.08 on 3 and 28 DF,  p-value: 3.594e-11
```



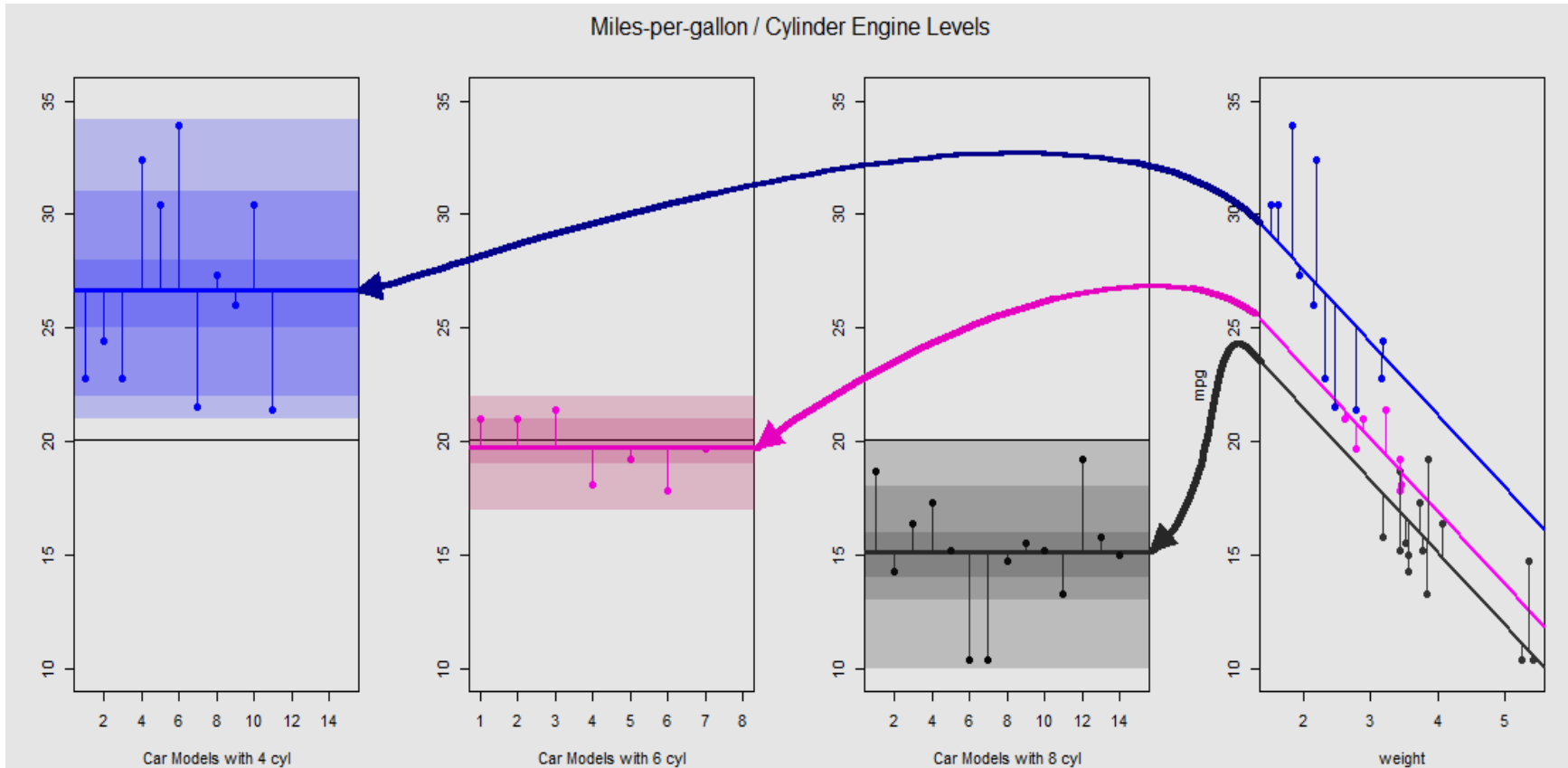
4 cylinders

6 cylinders

8 cylinders

ANOVA Equivalent to Linear Regression

If we suppress the effect of weight (wt variable/factor) by straightening these lines and returning them to the horizontal line, we'll end up with the ANOVA plot (**One-Way ANOVA with three levels**) of the model `aov(mtcars$mpg ~ as.factor(mtcars$cyl))` on the three subplots to the left.



ANOVA for Regression

The **basic regression line** concept, **DATA = FIT + RESIDUAL**, could be rewritten as follows:

$$(y_i - \bar{y}) = (\hat{y}_i - \bar{y}) + (y_i - \hat{y}_i)$$

The **1st term** is the **total variation** in the **response/dependent variable y**,
The **2nd term** is the **variation in mean response/dependent variable**, and
The **3rd term** is the **residual value**.

Squaring each of terms and adding over all of the n observations gives the equation

$$\sum (y_i - \bar{y})^2 = \sum (\hat{y}_i - \bar{y})^2 + \sum (y_i - \hat{y}_i)^2.$$

Equation may be written as **SST = SSM + SSE**, where SS is notation for **sum of squares** and T, M, and E are notation for **total**, **model**, and **error**, respectively.

The square of the sample correlation is equal to the ratio of the model sum of squares to the total sum of squares: **$r^2 = \text{SSM}/\text{SST}$** . (see next slide)

This formalizes the **interpretation of r^2 as explaining the fraction of variability in the data explained by the regression model**.

The sample **variance** s_y^2 is equal to $\rightarrow \sum (y_i - \bar{y})^2 / (n - 1) = \text{SST} / \text{DFT}$

Total Sum of Squares
divided by the Total Degrees
of Freedom (DFT).

For simple linear regression, MSM (mean square model) = $\sum (\hat{y}_i - \bar{y})^2 / (1) = \text{SSM} / \text{DFM}$.

The corresponding MSE (mean square error) = $\sum (y_i - \hat{y}_i)^2 / (n - 2) = \text{SSE} / \text{DFE}$

Estimate of the **variance** about the population **regression line** (σ^2)

Correlation in Linear Regression (r^2)

Correlation --- Karl Pearson Correlation Coefficient

Strength of linear association between 2 variables is quantified by the **correlation coefficient**. Given observations $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, formula for computing correlation coefficient is given

$$r = \frac{1}{n-1} \sum \left(\frac{x - \bar{x}}{s_x} \right) \left(\frac{y - \bar{y}}{s_y} \right) \quad , \quad s_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \text{ (the sample standard deviation)}$$

Correlation coefficient takes a value between -1 and 1, with 1 or -1 indicating perfect correlation. A **positive correlation** indicates a positive association between the variables (increasing values in one variable correspond to increasing values in the other variable), a **negative correlation** indicates a negative association between the variables (increasing values in one variable correspond to decreasing values in the other variable). A correlation value close to 0 indicates no association between the variables. Since formula for calculating correlation coefficient standardizes the variables, changes in scale or units of measurement will not affect its value. For this reason, the correlation coefficient is often more useful than a graphical depiction in determining the strength of the association between two variables.

Correlation in Linear Regression

Square of the correlation coefficient, r^2 , is a useful value in linear regression. This value represents the fraction of the variation in one variable that may be explained by the other variable. If correlation of 0.8 is observed between two variables (say, height and weight), then a linear regression model attempting to explain either variable in terms of the other variable will account for **64% of the variability in the data**.

Correlation coefficient also relates directly to the regression line $Y = a + bX$ for any two variables, where

Note: Because least-squares regression line always pass through the means of x and y , the regression line may be entirely described by the means, standard deviations, and correlation of the two variables under investigation.

$$b = r \frac{s_y}{s_x}$$

Analysis of Variance - Why it is more important than ever.

An interesting academic paper on the matter is Gelman's 2005 paper titled:

Analysis of Variance - Why it is more important than ever.

Although, I am not fully supportive of the paper, it can be a constructive read, since some important points are raised.

<http://www.stat.columbia.edu/~gelman/research/published/AOS259.pdf>

ANALYSIS OF VARIANCE—WHY IT IS MORE IMPORTANT THAN EVER¹

BY ANDREW GELMAN

Columbia University

Analysis of variance (ANOVA) is an extremely important method in exploratory and confirmatory data analysis. Unfortunately, in complex problems (e.g., split-plot designs), it is not always easy to set up an appropriate ANOVA. We propose a hierarchical analysis that automatically gives the correct ANOVA comparisons even in complex scenarios. The inferences for all means and variances are performed under a model with a separate batch of effects for each row of the ANOVA table.

We connect to classical ANOVA by working with finite-sample variance components: fixed and random effects models are characterized by inferences about existing levels of a factor and new levels, respectively. We also introduce a new graphical display showing inferences about the standard deviations of each batch of effects.

We illustrate with two examples from our applied data analysis, first illustrating the usefulness of our hierarchical computations and displays, and second showing how the ideas of ANOVA are helpful in understanding a previously fit hierarchical model.

Linear Regression Analysis

Linear Regression is one of the most widely used statistical model.

If we have Y variable which in continuous i.e. can take decimal values, and is expected to have linear relation with X variables, this relation could be modeled as linear regression.

(**Linear relation** is any equation that, when graphed, gives you a straight line)

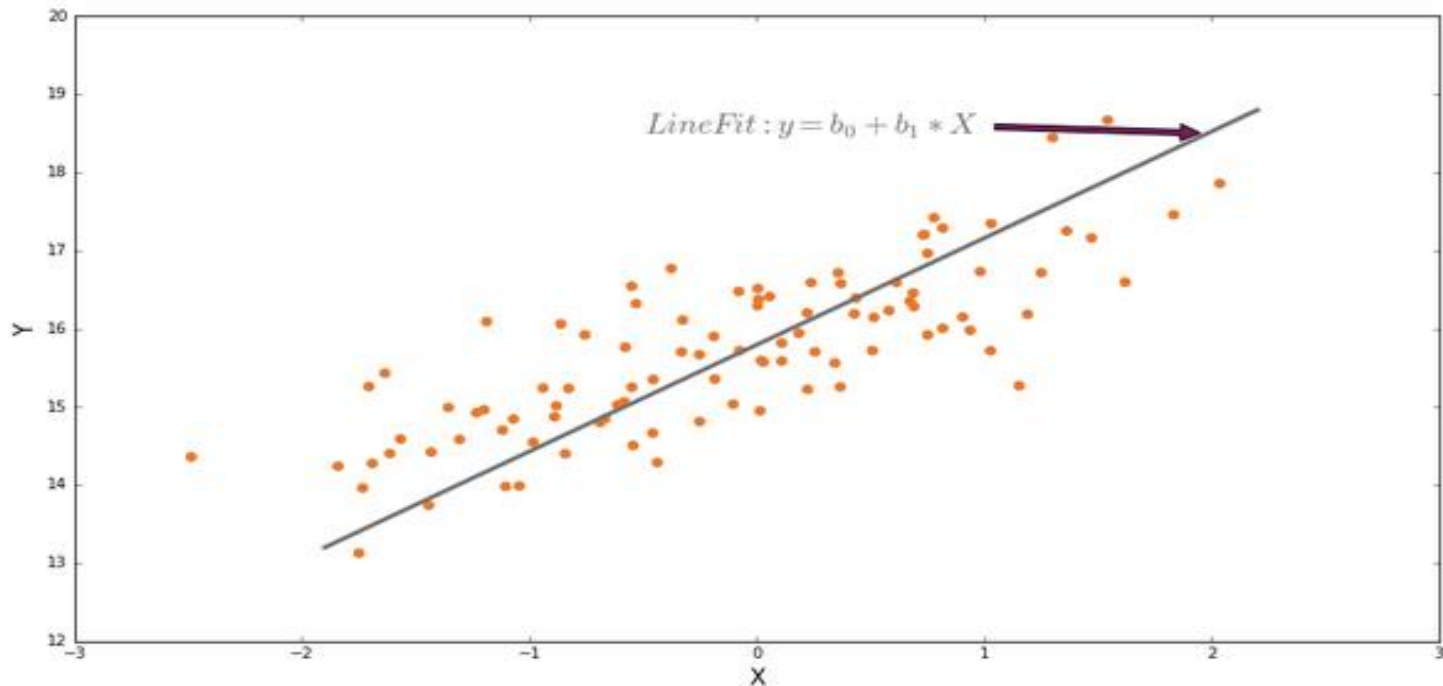
The **objective of the linear regression** is to express a dependent variable in terms of linear function of independent variables, if we have one dependent variable, (some call it uni-variate) **linear regression** or **single variable linear regression** and when we have many, we call it **multiple linear regression**.

Note: This is NOT **multi-variate**, the multi-variate linear regression refers to when we have more than one dependent variables.

Linear Regression – Scatter Plot

Without much of math and symbols, the idea of linear regression could be explored through a scatter plot. We generate a sample of X and Y from normal distribution.

Linear Regression is about fitting a straight line from the scatter plot, key challenge here is **what constitutes a best fit line**, in other words what would be the **best values** of b_0 and b_1 **coefficients**.



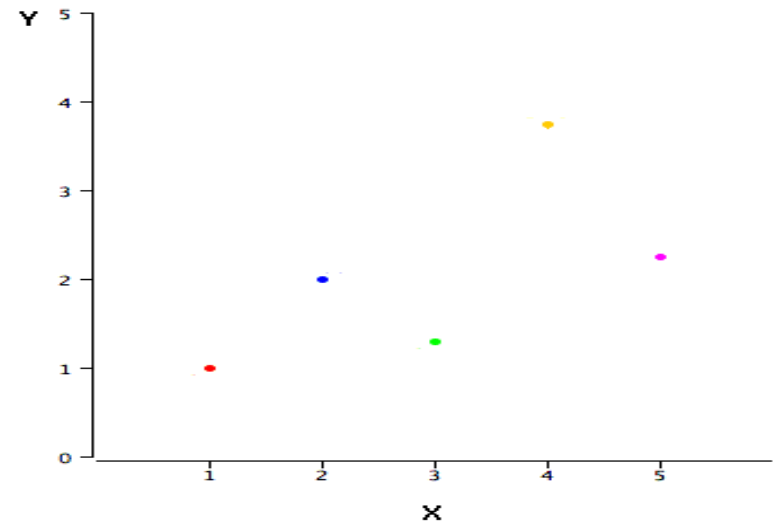
Simple Linear Regression - Example

When there is **one predictor** variable, the prediction method is called **simple regression**.

Example Data

X	Y
1.00	1.00
2.00	2.00
3.00	1.30
4.00	3.75
5.00	2.25

Scatter Plot of
Example Data



<Q> What's the corresponding Regression Line?

Linear regression consists of finding the best-fitting straight line through the points. The **best-fitting line** is called a **regression line**

The **formula for a regression line** is $Y' = bX + A$

where **Y'** is the predicted score, **b** is the slope of the line, and **A** is the Y-intercept.

Computing the Regression Line

The calculations are based on the statistics shown below.

The **slope (b)** can be calculated as follows:

☐ $b = r s_Y / s_X$

and the **intercept (A)** can be calculated as

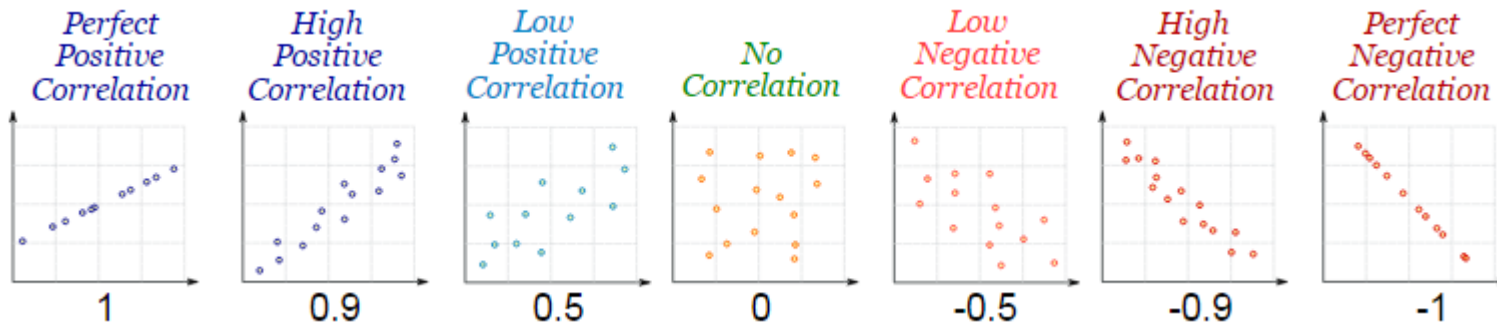
☐ $A = M_Y - bM_X$.

$$r = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{\left(\sum X^2 - \frac{(\sum X)^2}{N}\right) \left(\sum Y^2 - \frac{(\sum Y)^2}{N}\right)}}$$

M_X is the mean of X, M_Y is the mean of Y, s_X is the standard deviation of X, s_Y is the standard deviation of Y, and r is the correlation between X and Y.

The **standard deviation** is the square root of the **variance**. Formula for the **variance** is:

$$\sigma^2 = \frac{\sum (X - \mu)^2}{N} \quad \text{where } \sigma^2 \text{ is the variance, } \mu \text{ is the mean, and } N \text{ is the number of elements}$$



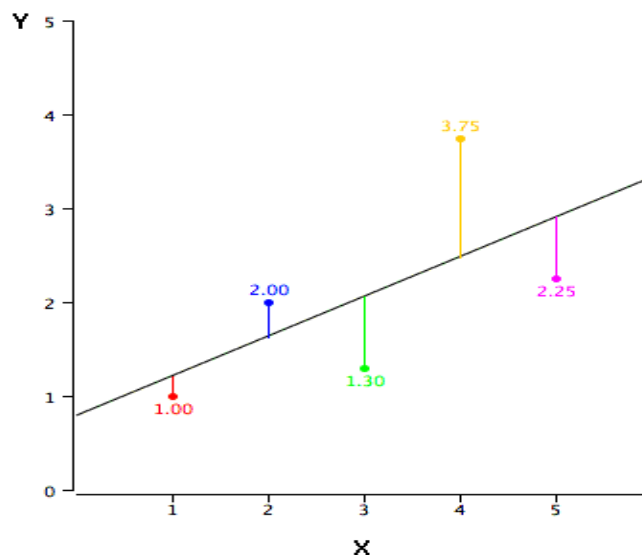
When 2 sets of data are strongly linked together, have **High Correlation**.

Plotting the Regression Line

By substituting the regression line formula ($Y' = bX + A$) the equation of the line is :
 $Y' = 0.425X + 0.785$

X	Y	Y'
1.00	1.00	1.210
2.00	2.00	1.635
3.00	1.30	2.060
4.00	3.75	2.485
5.00	2.25	2.910

→
The **black line** consists of the predictions, the points are the actual data, and the **vertical lines** between the points and the black line represent **errors of prediction**



M_X	M_Y	S_X	S_Y	r
3	2.06	1.581	1.072	0.627

The slope (b) can be calculated as follows:

$$b = r s_Y / s_X$$

and the intercept (A) can be calculated as

$$A = M_Y - bM_X.$$

For these data,

$$b = (0.627)(1.072)/1.581 = 0.425$$

$$A = 2.06 - (0.425)(3) = 0.785$$

Plotting the Regression Line in R

```
> x <- c(1.00, 2.00, 3.00, 4.00, 5.00)
> y <- c(1.00, 2.00, 1.30, 3.75, 2.25)
> plot(x,y,
+      main="Relationship between X and Y",
+      sub="A scatter plot of the example
data.")
> cor(x,y)
[1] 0.6268327
> fit <- lm(y ~ x)
> fit
Call:
lm(formula = y ~ x)
```

Coefficients:

(Intercept)	x
0.785	0.425

```
> abline(fit)
> summary(fit)
```

```
> summary(fit)
Call:
lm(formula = y ~ x)

Residuals:
    1     2     3     4     5 
-0.210  0.365 -0.760  1.265 -0.660 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.785     1.012   0.776  0.494
x              0.425     0.305   1.393  0.258

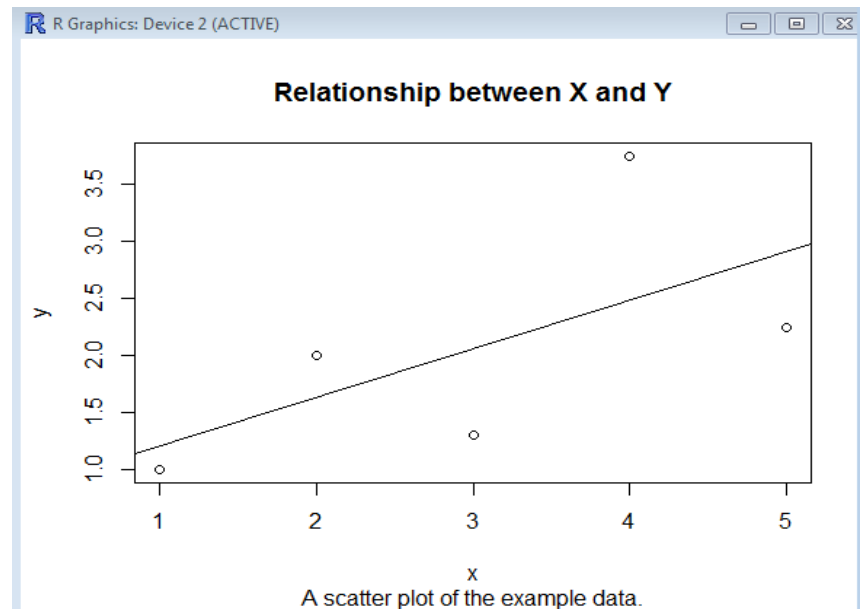
Residual standard error: 0.9645 on 3 degrees of freedom
Multiple R-squared:  0.3929,    Adjusted R-squared:  0.1906 
F-statistic: 1.942 on 1 and 3 DF,  p-value: 0.2578
```

```
> x <- c(1.00, 2.00, 3.00, 4.00, 5.00)
> y <- c(1.00, 2.00, 1.30, 3.75, 2.25)
> plot(x,y,
+      main="Relationship between X and Y",
+      sub="A scatter plot of the example data.")
> cor(x,y)
[1] 0.6268327
> fit <- lm(y ~ x)
> fit

Call:
lm(formula = y ~ x)

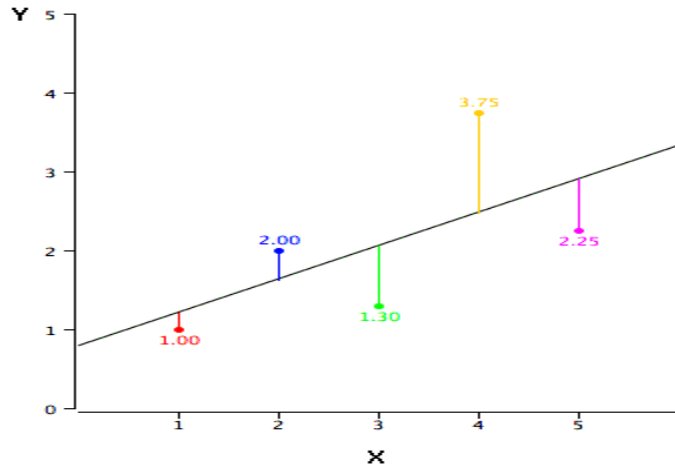
Coefficients:
(Intercept)          x 
         0.785         0.425 

> abline(fit)
```



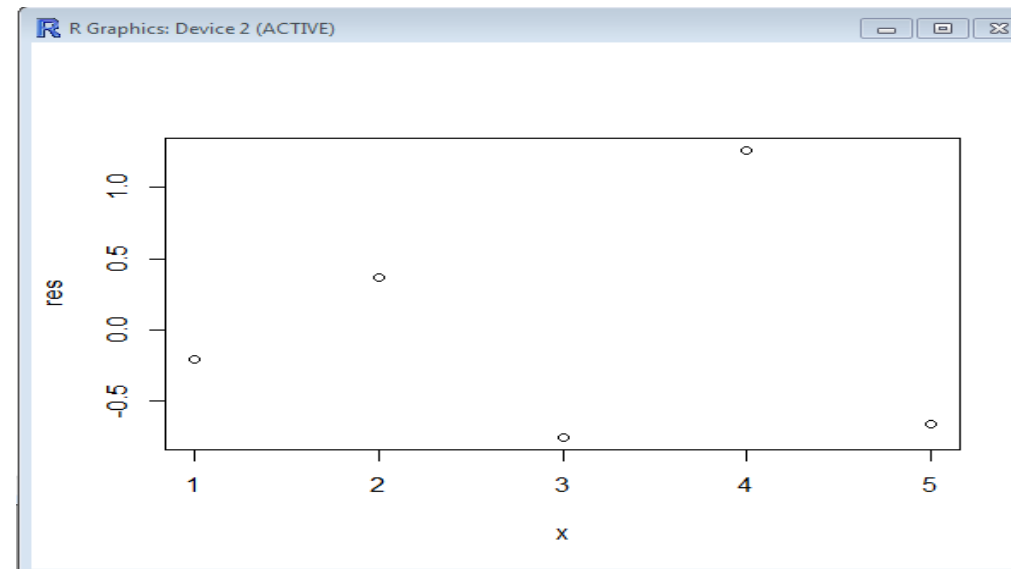
Linear Least Squares Regression

The vertical lines from the points to the regression line represent the **errors of prediction**. As you can see, the **red** point is very near the regression line; its error of prediction is small. By contrast, the **yellow** point is much higher than the regression line and therefore its error of prediction is large.



```
> res <- y - fit$coefficients[[2]]*x+fit$coefficients[[1]]
> res
[1] 1.360 1.935 0.810 2.835 0.910
> res <- y - (fit$coefficients[[2]]*x+fit$coefficients[[1]])
> res
[1] -0.210 0.365 -0.760 1.265 -0.660
> fit$residuals
      1      2      3      4      5
-0.210 0.365 -0.760 1.265 -0.660
> residuals(fit)
      1      2      3      4      5
-0.210 0.365 -0.760 1.265 -0.660
> plot(x,res)
```

```
> res <- y -
(fit$coefficients[[2]]*x+fit$coefficients[[1]])
> res
> fit$residuals
> residuals(fit)
> plot(x,res)
> plot(x,fit$residuals)
```



What is OLS (Ordinary Least Squares)?

OLS – Ordinary Least Squares regression: A technique for estimating coefficients

Discovered by Legendre (1805) and Gauss (1809) to solve problems in astronomy.

The model is of the form:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \dots + \beta_p x_p$$

β_0 – the intercept term

$\beta_1, \beta_2, \beta_3 \dots$ – coefficient estimates

$x_1, x_2, x_3, \dots x_p$ – predictor variables (i.e. columns in the dataset)

Example: $\text{Income} = 20,000 + 2,500 * \text{WorkExperience} + 1,000 * \text{EducationYears}$

Common Issues in Regression

1_ Missing Values

- Requires **imputation** (replacing missing data with substituted values)
- Results in record deletion

2_ Nonlinearities and Local Effects

- Example: $Y = 10 + 3x_1 + x_2 - .3x_1^2$
- Modeled via manual **transformations** or they are automatically added and then selected via forward, backward, stepwise, or **regularization**
- Ignores local effects unless specified by the analyst, but this is very difficult/impossible in practice without subject matter expertise or prior knowledge

3_ Interactions

- Example: $Y = 10 + 3x_1 - 2x_2 + .25x_1x_2$
- Manually added to the model (or through some automated procedure)
- Add interactions then use variable selection (i.e. regularized regression or forward, backward, or stepwise selection)

4_ Variable Selection

- Usually accomplished manually on in combination w/ automated procedures

Solutions to OLS (Ordinary Least Squares) Problems

Two methods that do not suffer from the drawbacks of linear regression are **CART (Classification And Regression Trees)** and **Gradient Tree Boosting (*)**

These methods automatically

- ☐ Handle **Missing Values**
- ☐ Model **Nonlinear** Relationships and **Local Effects**
- ☐ **Select Variables**
- ☐ Model **Variable Interactions**

(*) The term '**Boosting**' refers to a family of algorithms which converts '**weak learners**' to '**strong learners**' in ML. If classifier misclassifies some data, train another copy of it mainly on this misclassified part with hope that it will discover something subtle. And then **iterate**. Caution: Boosting too much may lead to **overfitting**!

Bagging & **Boosting** are 2 powerful **ensemble techniques** where a set of **weak learners** are combined to create a **strong learner**.

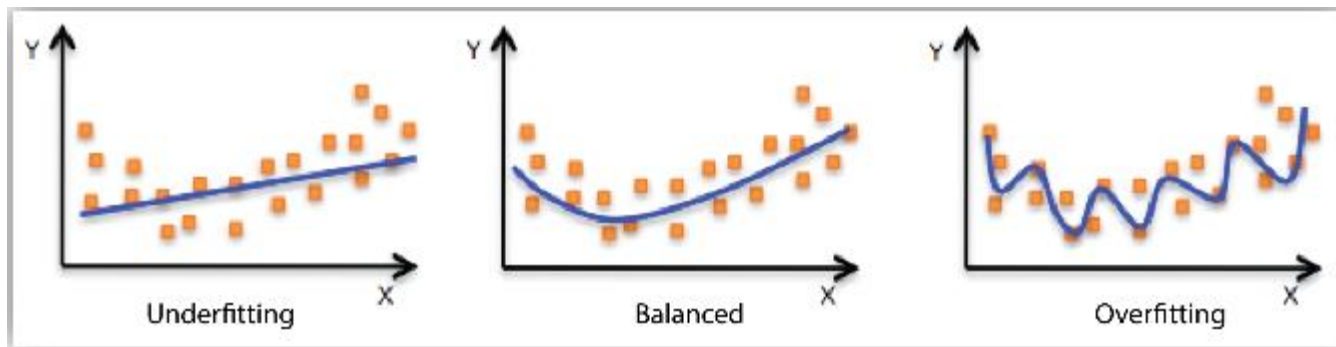
Bagging at training stage is done in **parallel**, whereas Boosting builds the learner **sequential**. In **Bagging element selection has same probability**, in **Boosting observations are weighted**.

Overfitting – Underfitting ML Models

Understanding model fit is important for understanding the root cause for poor model accuracy. This understanding will guide you to take corrective steps. We can **determine whether a predictive model is Underfitting or Overfitting** the training data by looking at the **Prediction Error** on the training data and the evaluation data.

A **model is Underfitting** the training data when the model **performs poorly on the training data**. This is because the **model is unable to capture the relationship between the input** examples (often called X) and the **target** values (often called Y).

A **model is Overfitting** the training data when you see that the model performs well on the training data but does not perform well on the evaluation data. This is because the **model is memorizing the data it has seen** and is unable to generalize to unseen examples.

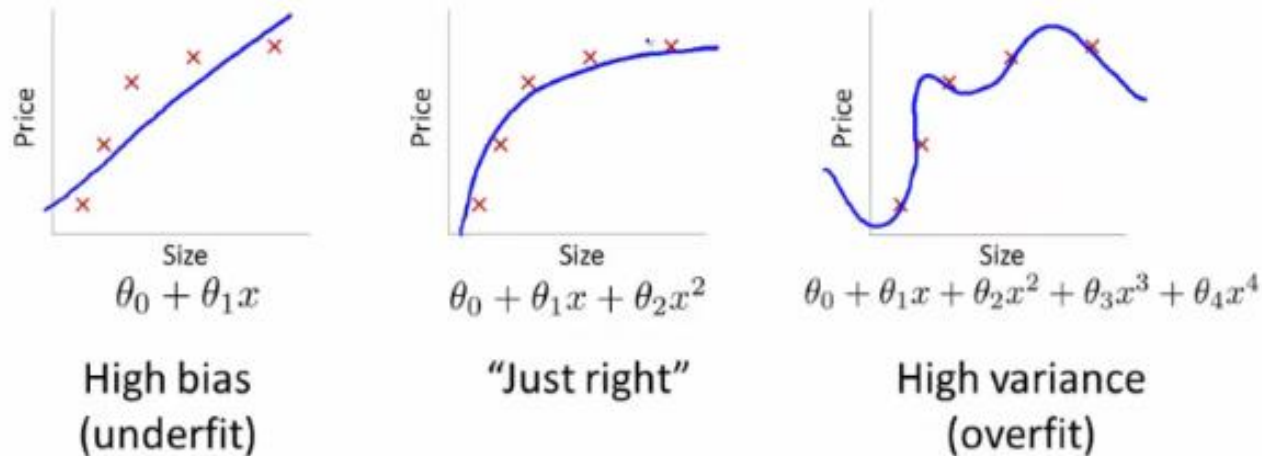


Overfitting (High Variance): tendency to learn random things irrespective of real signal.
Underfitting (High Bias): learner's tendency to consistently learn the same wrong thing.

ML Training & Validation Data Sets

Possibly, the **most important idea** in Machine Learning is that having **training** & **validation** data sets.

As motivation, suppose you don't divide the data, instead you use all of it. At the same time you have lots of parameters. End-result -> **Overfitting**.



The error for the pictured data points is lowest for the model on the far right (the blue curve passes through the red points perfectly), yet it's not the best choice. Why is that?? If you were to gather some new data points, they most likely would not be on that curve in the graph on the right, but would be to the curve in the middle graph.

Improving Model Accuracy

Poor performance on the training data could be because the model is too simple (the input features are not expressive enough) to describe the target well. Performance can be improved by **increasing model flexibility**.

To **increase model flexibility**, try the following:

- Add new domain-specific features and more feature Cartesian products, and change the types of feature processing used (e.g., increasing n-grams size)
- Decrease the amount of regularization used (see L1 & L2 Regularization)

If model overfits training data, take actions that **reduce model flexibility**.

To **reduce model flexibility**, try the following:

- **Feature selection**: consider using fewer feature combinations, decrease n-grams size, and decrease the number of numeric attribute bins.
- Increase the amount of regularization used.

Accuracy on training and test data could be poor because the learning algorithm did not have enough data to learn from.

You could **improve performance** by doing the following:

- **Increase the amount of training data** examples.
- **Increase the number of passes** on the existing training data (epochs).

Overfitting ML Models

In ML we fit a model to a set of training data so to make reliable predictions on untrained data.

In **overfitting**, a **model describes random error or noise instead of the underlying relationship**.

Overfitting occurs when a model is excessively complex, such as having too many parameters relative to the number of observations. A **model that has been overfit has poor predictive performance**, as it overreacts to minor fluctuations in the training data.

Overfitting occurs because the criterion used for training the model is not the same as the criterion used to judge the efficacy of a model.

In **Overfitting**, a model is trained by **maximizing performance on a set of training data**.

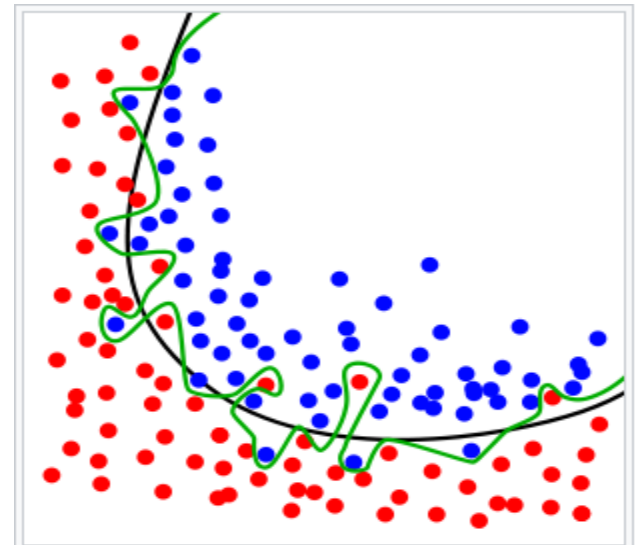
Efficacy is determined but by its ability to **perform well on unseen data**.

Overfitting occurs when a model begins to **"memorize" training data rather than "learning" to generalize from trend**

For example:

Green line represents an **'overfitting' model** and **black** line represents a **'regularized' model**.

While green line best follows training data, it is too dependent on it, likely to have higher error rate on new unseen data, compared to black line.



Overfitting – Validation Error

The potential for **overfitting depends** not only on the number of parameters and data but also the conformability of the model structure with the data shape, and the **magnitude of model error compared to the expected level of noise or error in the data**.

To **avoid overfitting** use additional techniques (e.g. [cross-validation](#), [regularization](#), [early stopping](#), [pruning](#), [Bayesian priors](#) on parameters or [model comparison](#)), that can indicate when further training is not resulting in better generalization.

For example:

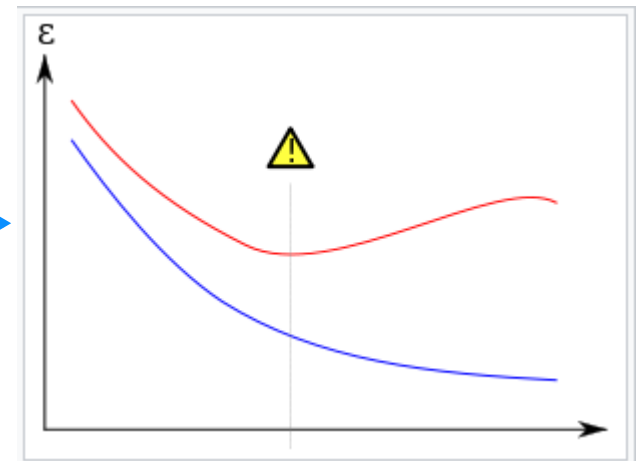
Overfitting/overtraining in supervised learning (e.g., neural network).

Training error is shown in **blue**.

Validation error in **red**, both as a function of the number of training cycles.

If validation error increases (**positive slope**) while the training error steadily decreases (**negative slope**) then a situation of overfitting may have occurred.

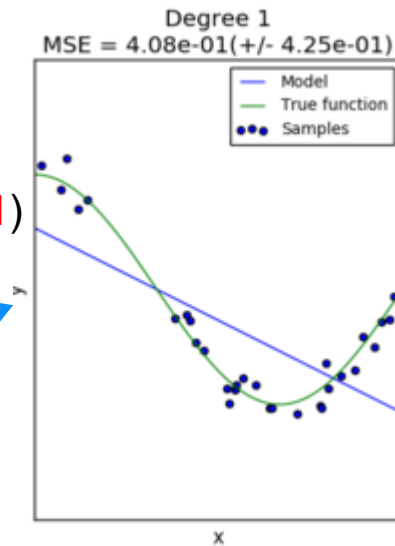
The **best predictive and fitted model** would be where the **validation error** has its **global minimum**.



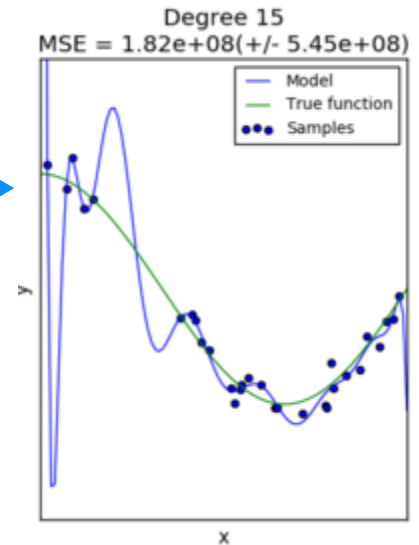
Underfitting vs. Overfitting – Calculating Error

Example: Use linear regression w/ polynomial features to approximate nonlinear (part of cosine) function.

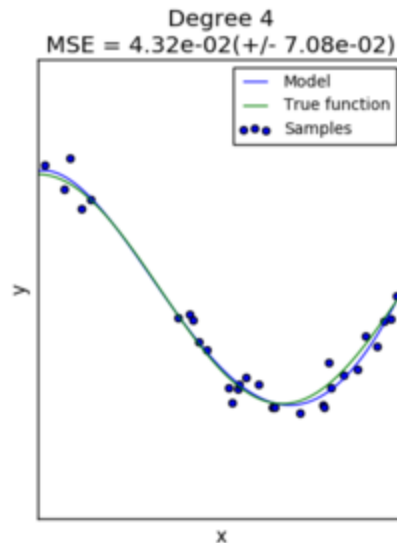
A linear function (polynomial with **degree 1**) is not sufficient to fit the training samples. This is **underfitting**.



Higher **degrees** (of **15**) the model will **overfit** the training data, i.e. it learns the noise of the training data



A polynomial of **degree 4** approximates the true function almost perfectly



We evaluate quantitatively ****overfitting**** / ****underfitting**** by using **cross-validation**. We calculate the **Mean Squared Error (MSE)** on the validation set.

The **higher MSE**, the **less likely** the model **generalizes** correctly from training data. (see next slide...)

Evaluating Model // Cross-Validation Score (scikit-learn)

To **evaluate** quantitatively **overfitting** / **underfitting** by using **cross-validation**.

We calculate the **Mean Squared Error (MSE)** on the validation set, the higher, the less likely the model generalizes correctly from the training data.

```
print(__doc__)

import numpy as np
import matplotlib.pyplot as plt
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score

np.random.seed(0)

n_samples = 30
degrees = [1, 4, 15]

true_fun = lambda X: np.cos(1.5 * np.pi * X)
X = np.sort(np.random.rand(n_samples))
y = true_fun(X) + np.random.randn(n_samples) * 0.1

plt.figure(figsize=(14, 5))

for i in range(len(degrees)):
    ax = plt.subplot(1, len(degrees), i + 1)
    plt.setp(ax, xticks=(), yticks=())

    polynomial_features = PolynomialFeatures(degree=degrees[i],
                                             include_bias=False)
    linear_regression = LinearRegression()
    pipeline = Pipeline([("polynomial_features", polynomial_features),
                          ("linear_regression", linear_regression)])
    pipeline.fit(X[:, np.newaxis], y)

    # Evaluate the models using crossvalidation
    scores = cross_val_score(pipeline, X[:, np.newaxis], y,
                             scoring="neg_mean_squared_error", cv=10)

    X_test = np.linspace(0, 1, 100)
    plt.plot(X_test, pipeline.predict(X_test[:, np.newaxis]), label="Model")
    plt.plot(X_test, true_fun(X_test), label="True function")
    plt.scatter(X, y, label="Samples")
    plt.xlabel("x")
    plt.ylabel("y")
    plt.xlim((0, 1))
    plt.ylim((-2, 2))
    plt.legend(loc="best")
    plt.title("Degree {} \n MSE = {:.2e} (+/- {:.2e})".format(
        degrees[i], -scores.mean(), scores.std()))
plt.show()
```


Linear Regression – Scatter Plot – Total Error

The general idea is to find a line (its coefficients) such that **total error** is at the minimum. There is a standard explanation that we need to minimize the **total square error**, which means we have to **solve a minimization problem** to solve optimal values of the coefficients.

This method involves a lot of mathematics or calculus.

See below the formula to calculate the **Error Sum of Squares (ESS)**

$$\text{ESS} = \sum_{i=1}^n x_i^2 - \frac{1}{n} \left(\sum_{i=1}^n x_i \right)^2$$

You may also hear the **Sum of Least Squared** residuals (a residual being: the difference between an observed value, and the fitted value provided by a model). It is also called **Ordinary Least Squares (OLS)** technique.

Let \hat{y}_i be the vertical coordinate of the best-fit line with x -coordinate x_i , so

$$\hat{y}_i \equiv a + b x_i,$$

then the error between the actual vertical point y_i and the fitted point is given by

$$e_i \equiv y_i - \hat{y}_i.$$

Linear Regression Problem as System of Equations

Let us assume we have 100 data points, let us assume that our solution has to satisfy all the data points, which means ...

$$y_0 = b_0 + b_1 * x_0$$

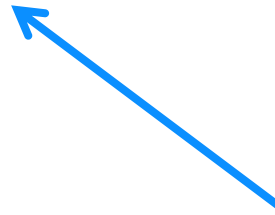
$$y_1 = b_0 + b_1 * x_1$$

.....

.....

$$Y_{99} = b_0 + b_1 * x_{99}$$

$$Y = Xb$$



We can write the above system of equations in **matrix** notation, where X and Y are matrices. See below a general matrix representation from Linear Algebra.

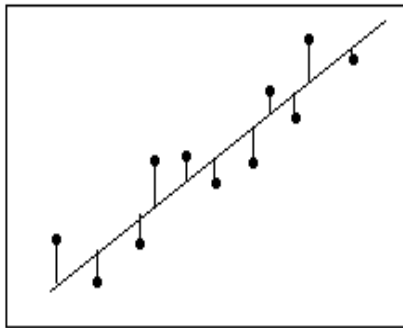
$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_1 \\ a_{21} & a_{22} & \dots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1p} \\ b_{21} & b_{22} & \dots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mp} \end{bmatrix}$$

What Is Goodness-of-Fit for a Linear Model?

Linear regression calculates an equation (model) that **minimizes** the distance between the fitted line and all of the data points. Technically, **ordinary least squares (OLS) regression minimizes the sum of the squared residuals**.

In general, a model fits the data well if the differences between the **observed values** and the **model's predicted values** are **small** and **unbiased**.

Before you look at the statistical measures for goodness-of-fit, you should check the residuals plots. **Residual plots** can reveal unwanted residual patterns that indicate biased results more effectively than numbers. When your residual plots pass muster, you can trust your numerical results and check the goodness-of-fit statistics.



Definition: Residual = Observed value - Fitted value

Goodness-of-Fit in Statistics

Goodness of fit of a statistical model describes how well it fits a set of observations. Measures of goodness of fit typically summarize the discrepancy between observed values and the values expected under the model in question.

Pearson's chi-squared test uses a measure of goodness of fit which is the sum of differences between observed and expected outcome frequencies (that is, counts of observations), each squared and divided by the expectation

$$\chi^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i}$$

OLS (Ordinary Least Squares) in Python

OLS is the most commonly used technique in [Statistical Learning](#).

It is the oldest, dating back 18th century Carl Friedrich Gauss & Adrien-Marie Legendre.

It is one of the easier and more intuitive techniques to understand.

The [statsmodels](#) package provides several different classes that provide different options for linear regression. Getting started with linear regression is quite straightforward with the [OLS module](#).

```
pip install numpy
pip install pandas
pip install statsmodels
```

```
# load numpy and pandas for data manipulation
import numpy as np
import pandas as pd

# load statsmodels as alias ``sm``
import statsmodels.api as sm
```

```
# load the longley dataset into a pandas data frame - first column (year) used as
row labels
```

```
df =
```

```
pd.read\_csv('http://vincentarelbundock.github.io/Rdatasets/csv/datasets/longley.csv', index_col=0)
```

```
df.head\(\)
```

R-squared (R^2) aka Coefficient of Determination

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the **coefficient of determination**, or the **coefficient of multiple determination** for multiple regression.

(Note: Not to be confused with Coefficient of Variation or Correlation)

The definition of R-squared is fairly straight-forward; it is the **percentage of the response variable variation that is explained by a linear model**. Or:

R-squared = Explained variation / Total variation

R-squared is always between 0 and 100%:

- **0%** -→ the model explains **none of the variability** of the response data around its mean.
- **100%** -→ the model explains **all the variability** of the response data around its mean.

In general, **the higher the R-squared, the better the model fits your data**. However, there are important conditions for this guideline!

The most general definition of the Coefficient of Determination is:

$$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

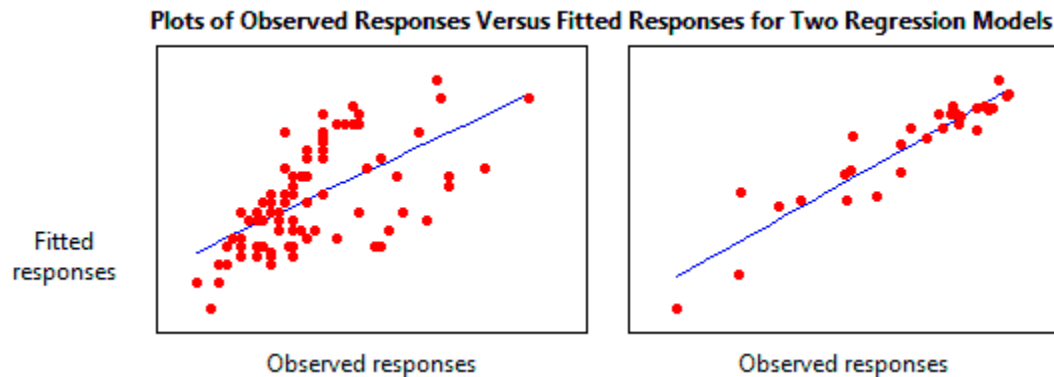
Where **SS_{res}** (**residual sum of squares**) and **SS_{tot}** (**tot sum of squares**) are

$$SS_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$$

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

Graphical Representation of R-squared (R^2)

Plotting fitted values by observed values graphically illustrates different **R-squared values for regression models**.



Regression model on the left accounts for 38.0% of the variance while the one on the right accounts for 87.4%. **The more variance that is accounted for by the regression model the closer the data points will fall to fitted regression line.**

Theoretically, if a model could explain 100% of the variance, the fitted values would always equal the observed values and, therefore, all the data points would fall on the fitted regression line.

R (Correlation Coefficient) vs R² (Determination Coefficient)

In statistics, the **correlation coefficient r** measures the **strength and direction of a linear relationship between two variables** on a scatterplot. The value of r is always between **+1 and -1**.

To interpret its value, see which of the following values your correlation r is closest to:

- | | |
|---|--|
| <input type="checkbox"/> Exactly -1. A perfect downhill (negative) linear relationship | +0.30. A weak uphill (positive) linear relationship |
| <input type="checkbox"/> -0.70. A strong downhill (negative) linear relationship | +0.50. A moderate uphill (positive) relationship |
| <input type="checkbox"/> -0.50. A moderate downhill (negative) relationship | +0.70. A strong uphill (positive) linear relationship |
| <input type="checkbox"/> -0.30. A weak downhill (negative) linear relationship | Exactly +1. A perfect uphill (positive) linear relationship |
| <input type="checkbox"/> 0. No linear relationship | |

The **correlation coefficient** of two variables in a data set equals to their **covariance** divided by the product of their individual **standard deviations**.

$$r_{xy} = \frac{s_{xy}}{s_x s_y}$$

Coefficient of Determination R² is a statistical measure of how close the data are to the fitted regression line.

The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model. Or:

R-squared = Explained variation / Total variation

R-squared is always between 0 and 100%:

- 0% indicates that the model explains none of the variability of the response data around its mean.
- 100% indicates that the model explains all the variability of the response data around its mean.


In general, the higher the R-squared, the better the model fits your data.

R² (Determination Coefficient) Calculation

Ways to calculate R² in Python:

- ☐ numpy (polyfitm, corrcoef)
- ☐ scipy (scipy.stats.linregress)
- ☐ scikit-learn (sklearn.metrics.r2_score)
- ☐ statsmodels (statsmodels.api, statsmodels.formula.api)

Ways to calculate R² in R:

- ☐ Apply the **lm function** 
> help(summary.lm)
- ☐ Manually, see next slide

```
summary.lm {stats}
```

Description

```
summary method for class "lm".
```

Usage

```
## S3 method for class 'lm'  
summary(object, correlation = FALSE, symbolic.cor = FALSE, ...)  
  
## S3 method for class 'summary.lm'  
print(x, digits = max(3, getOption("digits") - 3),  
      symbolic.cor = x$symbolic.cor,  
      signif.stars = getOption("show.signif.stars"), ...)
```


R² Calculation – Benchmarking (in Python)

```
import numpy as np
from scipy import stats
import statsmodels.api as sm
import math

n=1000
x = np.random.rand(1000)*10
x.sort()
y = 10 * x + (5*np.random.randn(1000)*10-5)

x_list = list(x)
y_list = list(y)

def get_r2_numpy(x, y):
    slope, intercept = np.polyfit(x, y, 1)
    r_squared = 1 - (sum((y - (slope * x + intercept))**2) / ((len(y) - 1) * np.var(y, ddof=1)))
    return r_squared

def get_r2_scipy(x, y):
    _, _, r_value, _, _ = stats.linregress(x, y)
    return r_value**2

def get_r2_statsmodels(x, y):
    return sm.OLS(y, sm.add_constant(x)).fit().rsquared

def get_r2_python(x_list, y_list):
    n = len(x)
    x_bar = sum(x_list)/n
    y_bar = sum(y_list)/n
    x_std = math.sqrt(sum([(xi-x_bar)**2 for xi in x_list])/(n-1))
    y_std = math.sqrt(sum([(yi-y_bar)**2 for yi in y_list])/(n-1))
    zx = [(xi-x_bar)/x_std for xi in x_list]
    zy = [(yi-y_bar)/y_std for yi in y_list]
    r = sum(zxi*zyi for zxi, zy_i in zip(zx, zy))/(n-1)
    return r**2
```

```
def get_r2_numpy_manual(x, y):
    zx = (x-np.mean(x))/np.std(x, ddof=1)
    zy = (y-np.mean(y))/np.std(y, ddof=1)
    r = np.sum(zx*zy)/(len(x)-1)
    return r**2
```

```
def get_r2_numpy_corrcoef(x, y):
    return np.corrcoef(x, y)[0, 1]**2
```

```
print('Python')
%timeit get_r2_python(x_list, y_list)
print('Numpy polyfit')
%timeit get_r2_numpy(x, y)
print('Numpy Manual')
%timeit get_r2_numpy_manual(x, y)
print('Numpy corrcoef')
%timeit get_r2_numpy_corrcoef(x, y)
print('Scipy')
%timeit get_r2_scipy(x, y)
print('Statsmodels')
%timeit get_r2_statsmodels(x, y)
```

R² (Determination Coefficient) Calculation in R

```
r.square = function(object=NULL, y, fitted.y)
```

```
{  
  # get y and fitted.y from a lm or glm object  
  if (!is.null(object)) {  
    fitted.y = fitted(object)  
    if (class(fitted.y) == "numeric")  
      y = object$model[[1]]  
    else  
      y = object$model[,1]  
  }  
  
  # compute coefficient of determination  
  if (class(fitted.y) == "numeric") {  
    return(cor(y, fitted.y)^2)  
  } else {  
    R2 = double(ncol(y))  
    for (ic in 1:ncol(y))  
      R2[ic] = cor(y[,ic], fitted.y[,ic])^2  
    return(R2)  
  }  
}
```

```
#  
# Compute coefficient of determination (R-squared)  
#  
# + object is the returned object of lm(..), glm(..) or rlm(..)  
#   if a regress model object is given, r.square(..) will get y  
#   and fitted.y from this model  
# + user can also pass their y and fitted.y instead of a regress  
#   model object  
#  
# for example:  
#  
# r.square(y, fitted.y)  
# r.square(lm(y1~x1+x2+x3, data=dfm))  
# r.square(rlm(cbind(y1, y2, y3)~x1+x2+x3, data=dfm))  
#
```

```
> head(cars)  
  speed dist  
1     4    2  
2     4   10  
3     7    4  
4     7   22  
5     8   16  
6     9   10
```

```
> r.square(lm(dist ~ speed, data=cars))  
[1] 0.6510794  
> |
```

```
> help(cars)  
> scatter.smooth(x=cars$speed, y=cars$dist, main="Dist ~ Speed")  
> linearMod <- lm(dist ~ speed, data=cars)  
> print(linearMod)
```

Limitations of $R^2 \Rightarrow$ Adjusted R^2 & Predicted R^2

- ❑ R-squared *cannot* determine whether the **coefficient estimates and predictions are biased**, which is why you must assess the residual plots.
- ❑ R-squared does not indicate whether a **regression model is adequate**. You can have a low R-squared value for a good model, or a high R-squared value for a model that does not fit the data!
- ❑ Every time you **add a predictor to a model, the R-squared increases**, even if due to chance alone. It never decreases. Consequently, a model with more terms may appear to have a better fit simply because it has more terms.
- ❑ If a model has **too many predictors** and **higher order polynomials**, it begins to model the **random noise** in the data. This condition is known as **overfitting** the model and it produces misleadingly high R-squared values and a lessened ability to make predictions.

Adjusted R-squared, Predicted R-squared are designed to address these issues

Adjusted R^2 is a modified version of R^2 that has been **adjusted for the number of predictors in the model**. The adjusted R-squared increases only if the new term improves the model more than would be expected by chance. It decreases when a predictor improves the model by less than expected by chance. The adjusted R-squared can be negative, but it's usually not. It is always lower than the R-squared.

The **Predicted R^2** indicates **how well a regression model predicts responses for new observations**. This statistic helps determine when model fits original data but is less capable of providing valid predictions for new observations.

The meaning of “p” value

- 1_ Probability of obtaining test results at least as extreme as the results actually observed, under the assumption H_0 is correct,
- 2_ Used to help you support or reject H_0
- 3_ Evidence against H_0

		<u>True state of affairs</u>	
		H_0	H_A
<u>What we claim</u>	H_0	Correct Non-Rejection [of H_0]	Miss (Type II Error)
	H_A	False Alarm (Type I Error)	Correct Rejection [of H_0]

Inference Statistics

p = Probability obtaining ‘test-statistic’ this extreme, given that H_0 is true

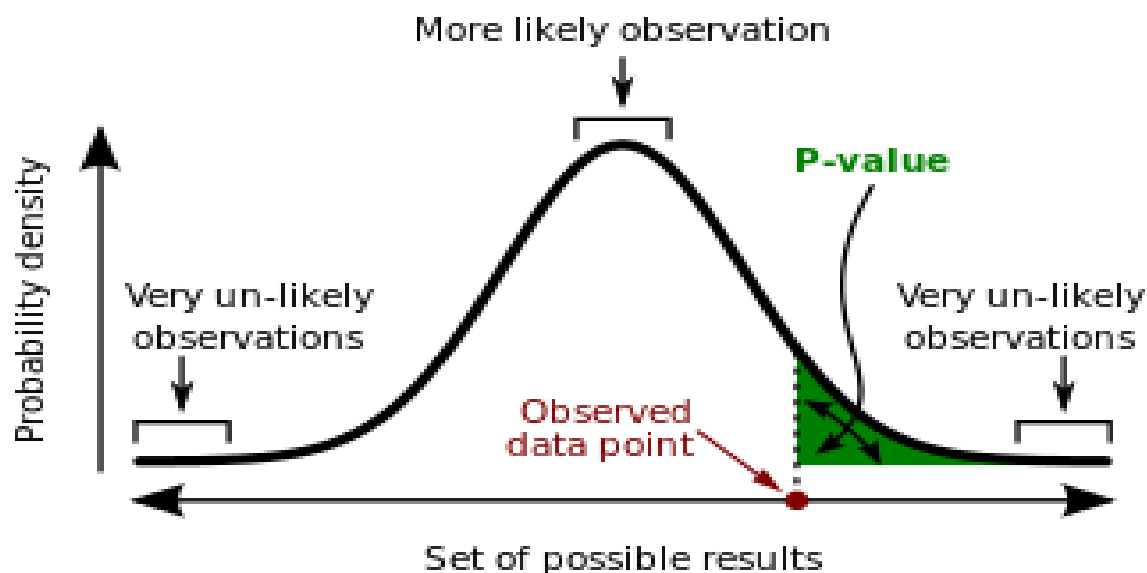
Inference Rule: Reject H_0 if test-statistic is "large" enough.

THEREFORE: If we choose to reject H_0 when **p** < 0.05, then the chance of having a False Alarm is no more than 0.05 (1 out of 20).

When **p** < 0.05, we commonly say that the effect is statistically significant (in the case of a regression coefficient, we say it is significantly different from zero).

P-value Calculation

Example of a **p-value computation**. The vertical coordinate is the probability density of each outcome, computed under the null hypothesis. The **p-value is the area under the curve past the observed**



A **p-value** (shaded green area) is the probability of an observed (or more extreme) result assuming that the null hypothesis is true.

Questions?

