

# Linear Regression

It all depends on line

# Linear Regression

---

- Introduction
- Linear Regression
- Linear Regression using stats model
- Interpret Regression Model
- Linear Regression (OLS) Assumptions
- Linear Regression in R



---

# Introduction

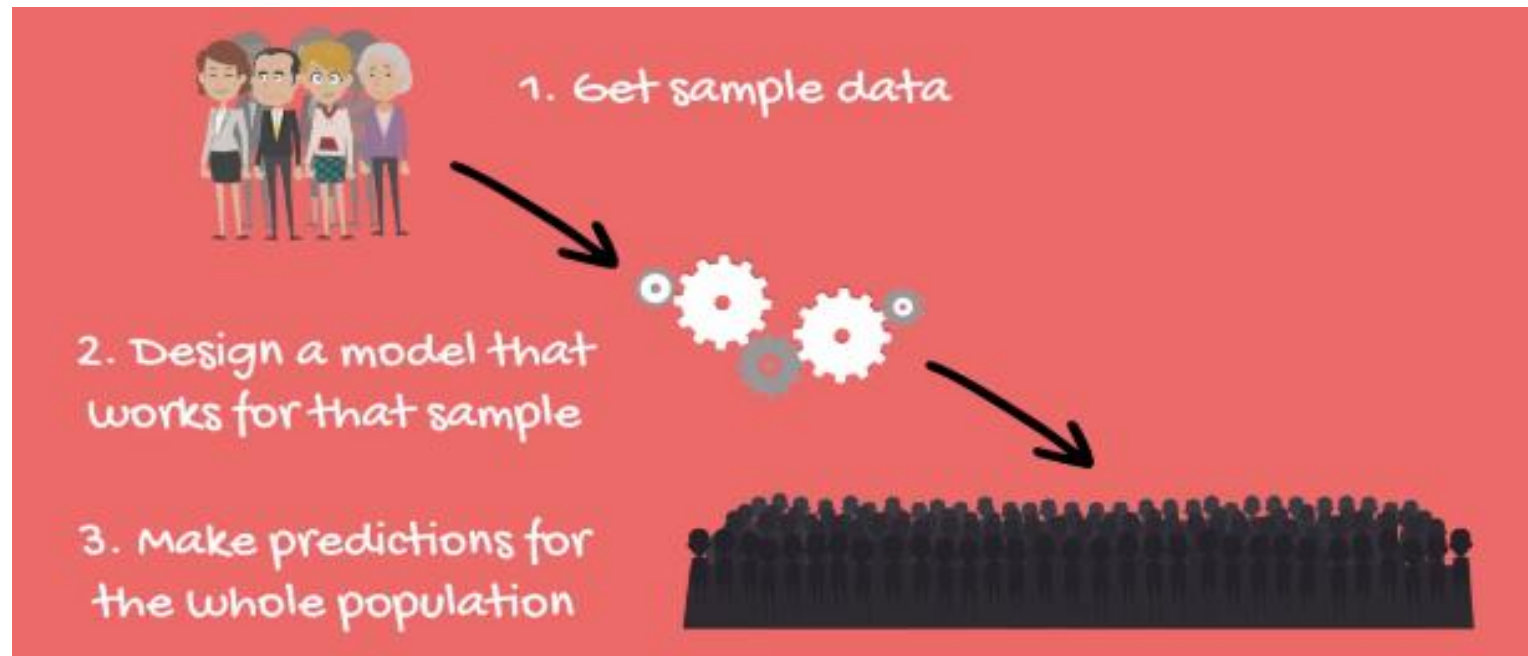


# Linear Regression

---

“A linear regression is a linear approximation of a causal relationship between two or more variables”

The process goes like this:



# Linear Regression

---



**DEPENDENT**

/predicted/

**INDEPENDENT**

/predictors/



$$Y = F(x_1, x_2, \dots, x_k)$$

The dependent variable  $y$  is a function of the independent variables  $x_1$  to  $x_k$



---

# Simple Linear Regression



# Simple Linear Regression

---



**DEPENDENT**

/predicted/

**INDEPENDENT**

/predictors/



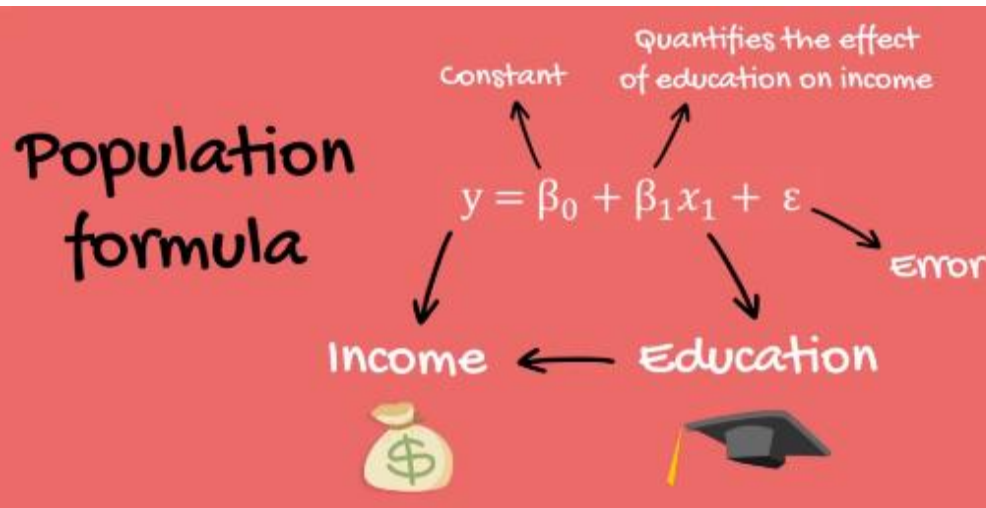
$$Y = F(x_1, x_2, \dots, x_k)$$

The dependent variable  $y$  is a function of the independent variables  $x_1$  to  $x_k$





↓  
Independent variable





# Simple Linear Regression Equation

---

The diagram illustrates the Simple Linear Regression Equation,  $\hat{y} = b_0 + b_1 x_1$ , with four annotations:

- Constant**: An arrow points from this label to the  $b_0$  term in the equation.
- Quantifies the effect the independent (x) on the dependent (y)**: An arrow points from this label to the  $b_1 x_1$  term in the equation.
- Estimated / Predicted value**: An arrow points from this label to the  $\hat{y}$  term in the equation.
- Sample data for independent variable**: An arrow points from this label to the  $x_1$  term in the equation.

# Simple Linear Regression Equation

There are 3 terms that we must define:

- Sum of Squares Total (SST / TSS – Total Sum of Squares)
- Sum of Squares Regression (SSR / ESS – Explained Sum of Squares)
- Sum of Squares Error (SSE/ RSS – Residual Sum of Squares)

Mathematically,

$$SST = SSR + SSE$$

$$\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^n e_i^2$$



# Simple Linear Regression Equation

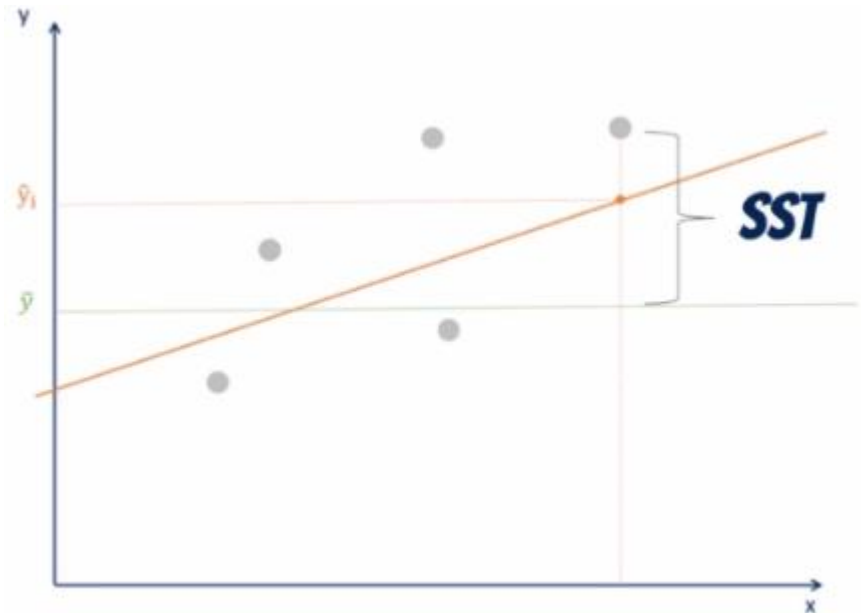
**Sum of Squares Total (SST) :**

**Also denoted as Total Sum of Squares (TSS)**

Sum of squared differences between the observed dependent variable and its means.

You can think of this as the dispersion of the observed variables around the mean. It is the measure of the total variability of the data set.

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



# Simple Linear Regression Equation

**Sum of Squares Regression (SSR) :**

**Also denoted as Explained Sum of Squares (ESS)**

Sum of squared differences between the predicted value and the mean of the dependent variable.

Think of this as a measure of how well your line fits the data.

**If SSR = SST, it mean your regression model captures all the variability and is perfect.**

$$SSR = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$$



# Simple Linear Regression Equation

**Sum of Squares Error (SSE)** or simply *error* or *unknown variability*:

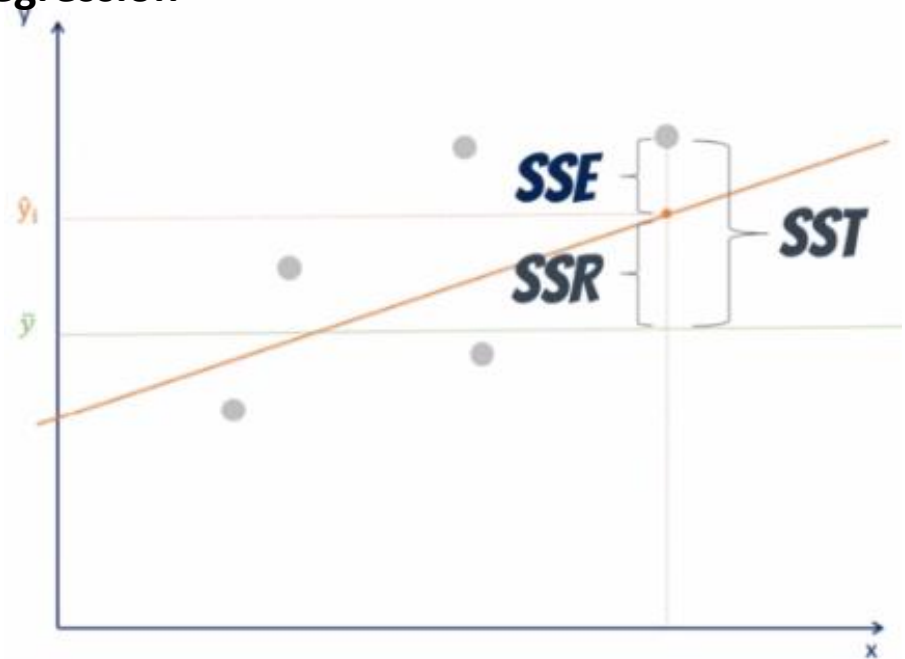
**Also denoted as Residual Sum of Squares (RSS)**

It is the difference between observed value and the predicted value.

We usually want to minimize it.

**The smaller the error (SSE) the better the estimation power of the regression**  
**i.e. a lower error will cause a better regression**

$$SSE = \sum_{i=1}^n e_i^2$$



---

# Correlation Vs Regression



# Linear Regression Vs Correlation

## CORRELATION

1

Relationship



2

Movement together



3

$\rho(x,y) = \rho(y,x)$



4

Single point



## REGRESSION

One variable affects the other



cause and effect



One way



Line



# Linear Regression Vs Correlation

---

Correlation is the measure of linear relationship between two variables.

Regression analysis is the tool to model the linear relationship between two variables.

**It tells to what degrees approximately the independent variable affects the dependent**

Regression analysis is useful when there is strong positive or negative correlation between two variables.

Poor correlation between two variables does not implies independency of two variables. There may be non-linear relationship between given two variables.





# When to use Linear Regression ?

Select A Statistical Test

■ Hypothesis tests to find relationships between project Y and potential X's

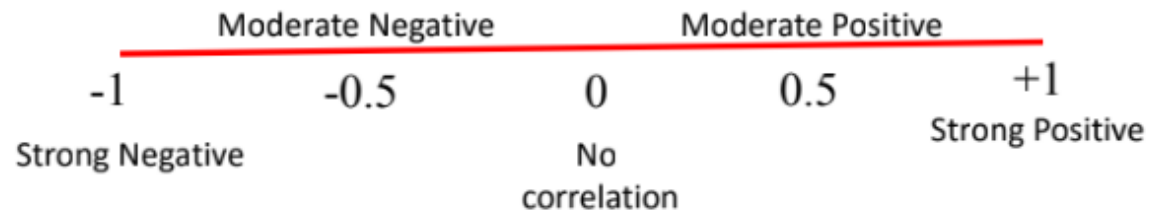
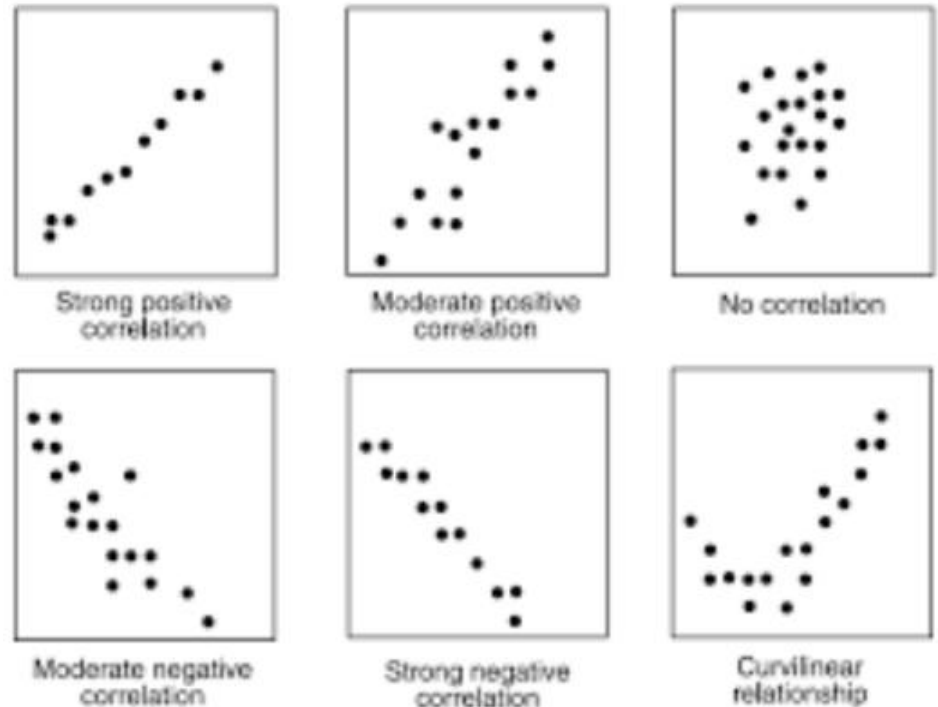
		Y	
		Continuous	Discrete
X	Continuous	Simple Linear Regression	Logistic Regression
	Discrete	2 Sample t-Test (Compare Means of two samples) ANOVA (Compare means of multiple samples) Homogeneity of Variance (Compare variances)	Chi-Square Test

E.g: We used ANOVA with the plant food example, where X was discrete with number of feedings per day (one or two) while Y was the plant height which was continuous

# When to use Correlation ?

When variables under consideration are continuous and you want to check whether there is any linear relationship between variables (more than 2 variables) under consideration.

When you want to check whether the observed correlation is significant or not.



# Correlation with R

---

```
> cor(x,y)
[1] -0.7761684
```

```
> cor.test(x,y)
Pearson's product-moment correlation
```

data: x and y

t = -6.7424, df = 30, p-value = 1.788e-07

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.8852686 -0.5860994

sample estimates:

cor

-0.7761684

---



---

# Simple Linear Regression with Python

statsmodels



# Linear Regression with Python - statsmodels

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

```
data = pd.read_csv('1.01. Simple linear regression.csv')
```

```
y = data['GPA']
x1 = data['SAT']
```

```
plt.scatter(x1,y)
plt.xlabel('SAT', fontsize=10)
plt.ylabel('GPA', fontsize=10)
```

```
x = sm.add_constant(x1)
```

```
x.head()
```

```
results = sm.OLS(y,x).fit()
```

```
results.summary()
```

```
plt.scatter(x1,y)
plt.xlabel('SAT', fontsize=10)
plt.ylabel('GPA', fontsize=10)
yhat = 0.275 + 0.0017 * x1
plt.plot(x1, yhat, lw=3, c='orange', label='regression line')
```

## OLS Regression Results

Dep. Variable:	GPA	R-squared:	0.408			
Model:	OLS	Adj. R-squared:	0.399			
Method:	Least Squares	F-statistic:	58.05			
Date:	Thu, 25 Apr 2019	Prob (F-statistic):	7.20e-11			
Time:	23:45:57	Log-Likelihood:	12.672			
No. Observations:	84	AIC:	-21.34			
Df Residuals:	82	BIC:	-16.48			
Df Model:	1					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	0.2750	0.409	0.673	0.503	-0.538	1.088
SAT	0.0017	0.000	7.487	0.000	0.001	0.002
Omnibus:	12.839	Durbin-Watson:	0.950			
Prob(Omnibus):	0.002	Jarque-Bera (JB):	16.155			
Skew:	-0.722	Prob(JB):	0.000310			
Kurtosis:	4.590	Cond. No.	3.29e+04			

# Linear Regression with Python - statsmodels

---

## Interpreting the Regression Table

1. Coefficients Table: To obtain the regression equation

Looking at the coefficients table:

- Coefficient of constant ( $b_0$ ) = 0.2750
- Coefficient of SAT ( $b_1$ ) = 0.0017
- Hence the equation of line:  $y = b_0 + b_1 * x_1$

$$y = 0.275 + 0.0017 * x_1 \quad \text{i.e.}$$

$$\text{GPA} = 0.275 + 0.0017 * \text{SAT}$$

**Std err:** The standard error shows the accuracy of prediction for each variable.  
**The lower the standard error better the estimate**

**t statistics and P value:** There is a hypothesis involved here

Null hypothesis is  $H_0: \beta = 0$  (Coefficient = zero, i.e. variable not significant)

Basically the hypothesis asks Is this a useful variable ?

***If  $p\text{-value} < 0.05$ , we reject null hypothesis – i.e. variable are significant***



# Linear Regression with Python - statsmodels

---

## Interpreting the Regression Table

### 2. Model Summary Table:

Looking at the model summary table:

- **Dep Variable:** Variable we are trying to predict
- **Model:** Which model/approach is used for regression. Here OLS or Ordinary Least Squares
  - OLS is the most common method for linear regression. Least Squares stands for the minimum squares error or SSE.
  - Hence this method aims to find the line, which minimizes the SSE.
  - Other Models which can be used are:
    - Generalized Least Squares
    - Maximum Likelihood Estimation
    - Bayesian Regression
    - Kernel Regression
    - Gaussian Process Regression
- **Method:** The method used within the Model. Here Least Squares



# Linear Regression with Python - statsmodels

---

## Interpreting the Regression Table

### 2. Model Summary Table:

Looking at the model summary table:

- **R-squared: measures the goodness of fit of your model. More factors (variables) we include in Regression, the higher the R-squared**
  - This is a relative measure and takes values ranging from 0 to 1
  - $R^2 = 0$  , Regression Line explains no variability of data
  - $R^2 = 1$  , Regression Line explains entire variability of data
  - Usually you get values between 0.2 – 0.9
  - This is the ratio of SSR to SSE ( $SSR / SSE$ )

There is no such thing called as a good value of  $R^2$ . It is relative.

In Science or Physics or Mathematics usually 0.7 or higher is a good value.

In Social Science a value of 0.2 or 0.3 could be fantastic.

It depends on the problem and number of variables involved.

e.g. your salary. It depends on education, your experience, place of living, languages you speak etc. However, it is quite possible that all of these in total contribute less than 40% of the variability of your salary.

---





# Linear Regression with Python - statsmodels

---

## Interpreting the Regression Table

### 2. Model Summary Table:

Looking at the model summary table:

- **R-squared:**

Here in our model, the R-squared value is  $0.406 \sim 0.41$ .

In other words, SAT explains the 41% variability of the college GPA for our sample

Now this value is neither good nor bad, but 41% denotes that probably we may be missing some information, say, like gender, attendance, student working or not, marital status etc.



# Linear Regression with Python - statsmodels

## Interpreting the Regression Table

### 2. Model Summary Table:

Looking at the model summary table:

- **Adj. R-squared:** This is smaller than the R-square as it penalizes the excessive use of additional variables added to the model which are not significant.

Lets, see how it penalizes the use of variables that are not significant

So we use a new dataset, which simply contains a new variable added to the college GPA dataset. The column added is a number assigned to each student at random which is not significant.

```
x3.head()
```

	SAT	Rand 1,2,3
0	1714	1
1	1664	3
2	1760	3
3	1685	3
4	1693	2

```
data1 = pd.read_csv('1.02. Multiple linear regression.csv')
```

```
y3 = data1['GPA']  
x3 = data1[['SAT', 'Rand 1,2,3']]
```

```
x3 = sm.add_constant(x3)  
results3 = sm.OLS(y3, x3).fit()
```

```
results3.summary()
```

# Linear Regression with Python - statsmodels

## Interpreting the Regression Table

### 2. Model Summary Table:

Looking at the model summary table:

- **Adj. R-squared:**

Without New Variable		With New Variable	
R-squared:	0.406	R-squared:	0.407
Adj. R-squared:	0.399	Adj. R-squared:	0.392
F-statistic:	56.05	F-statistic:	27.76

Although the R-squared value has gone up, which it should as it expects each new variable to increase the explanatory power of the Regression,

The Adjusted R – Squared value does down to **0.392**.

**The model has been penalized for adding additional variable that has no strong explanatory power. We have added information to the model, but have lost the value.**

# Linear Regression with Python - statsmodels

## Interpreting the Regression Table

### 2. Model Summary Table:

Looking at the model summary table:

- **Adj. R-squared:**

Also the Regression analysis in itself is smart. It does point out when we add an impractical variable. Check out the coefficients table p-value for the new variable.

**The p-value is greater than 0.05**, which simply means the null hypothesis upholds i.e. the variable is of no significant importance.

**Conclusion:** The variable **Rand 1,2,3** has not only worsen the explanatory power of the model (Reflected by lower Adjusted R-square) but is also insignificant. Hence it should be dropped altogether.

	coef	std err	t	P> t	[0.025	0.975]
const	0.2960	0.417	0.710	0.480	-0.533	1.125
SAT	0.0017	0.000	7.432	0.000	0.001	0.002
Rand 1,2,3	-0.0083	0.027	-0.304	0.762	-0.062	0.046

# Linear Regression with Python - statsmodels

---

## Interpreting the Regression Table

### 2. Model Summary Table:

Looking at the model summary table:

- **F-statistic:** Used to get the overall significance of the model
  - It follows the f-distribution
  - Lower the F-statistics, closer to non-significant model

The hypoth  $H_0: \beta_1 = \beta_2 = \dots = \beta_k = 0$  is:

$H_1$ : at least one  $\beta_i \neq 0$

If all betas are zero, then none of the x (variables) matter => our model has no merit  
Let's compare the F-statistics of our old model vs the new model with a new variable



# Linear Regression with Python - statsmodels

---

## Interpreting the Regression Table

### 2. Model Summary Table:

Looking at the model summary table:

- **F-statistic:** Used to get the overall significance of the model

Old Model	
<b>F-statistic:</b>	56.05
<b>Prob (F-statistic):</b>	7.20e-11

F-statistics = 56.05  
p-value = 7.20e-11 ~ **0.000**  
which says that the overall model  
is significant

New Model	
<b>F-statistic:</b>	27.76
<b>Prob (F-statistic):</b>	6.58e-10

F-statistics = 27.76  
p-value = 6.58e-10 ~ **0.000**  
F-statistics is lower, the model is  
still significant but less than the  
old one.

Lower the F-statistic, closer to a non-significant model



---

# Multiple Linear Regression with Python

## statsmodels



# Linear Regression with Python

**The multiple regression is not about the best fitting line anymore. It is about the best fitting model.**

It stops being two dimensional and when we have more than three dimensions (variables), there is no visual way to represent the data.

**As we saw from OLS, what we really want is the least SSE. And how do we decrease SSE ?**  
By increasing the explanatory power of the model, which can be done by including more variables.

More variables usually result in better fitting model. However, this is not always true. We shall see that case as well.

The diagram consists of three stacked panels, each showing the linear regression equation  $\hat{y} = b_0 + b_1x_1 + b_2x_2 + \dots + b_kx_k$  with arrows pointing to specific terms and their labels.

- Top panel:** An arrow points from the label "inferred value" to  $\hat{y}$ , and another arrow points from the label "intercept" to  $b_0$ .
- Middle panel:** Three arrows point from the labels "independent variable" to  $x_1$ ,  $x_2$ , and  $x_k$  respectively.
- Bottom panel:** Three arrows point from the labels "coefficient" to  $b_1$ ,  $b_2$ , and  $b_k$  respectively.



---

So Far...



# Summary so far...

---

Correlation

SST, SSR, SSE

Causation

OLS

Simple LR model

R-squared

Multivariate LR model

Adjusted R-squared

Geometrical representation

F-test



---

# Linear Regression (OLS) Assumptions



# Linear Regression (OLS) Assumptions

---

There are 5 assumptions for OLS:

1. **Linearity:** Linear Regression is the simplest regression and it assumes linearity among the variable(s)
2. **No endogeneity:** Omitted Variable bias
3. **Normality & Homoscedasticity of error term:** The error term is normally distributed . Homoscedasticity mean constant variance.
4. **No autocorrelation:** The co-variance of any two error terms is zero. This is the assumption that usually stops us from using a linear regression in our analysis.
5. **No multi-collinearity:** Multi-collinearity It is observed when two or more variables have a high correlation between each other.



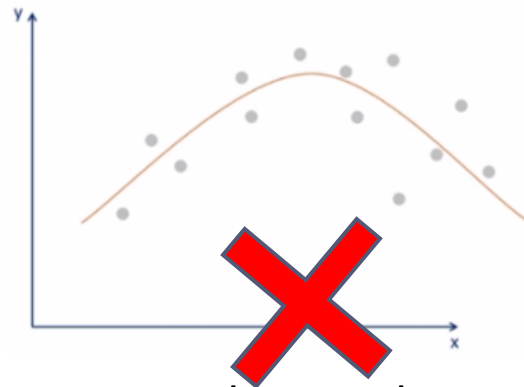
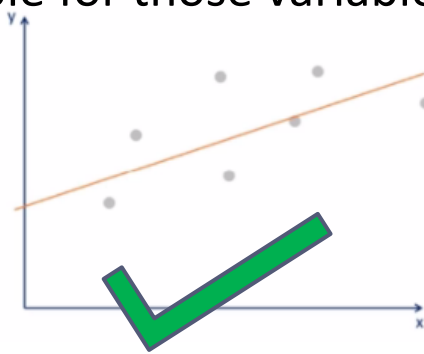
# Linear Regression (OLS) Assumptions

---

**Linearity:** Variables should be linearly dependent with each other

The easiest way is to take the independent variable(s)  $x_i$  one at a time and plot it against the dependent variable  $y$  on a scatter plot.

If the data points form any straight line pattern then the linear regression is suitable for those variables.



For Non-linear variables, either you can use the non-linear regression or you can transform your relationship to linear and then use it.

For transformation you can use:

- Exponential Transformation
- Log Transformation

# Linear Regression (OLS) Assumptions

---

**No endogeneity:** It is related to the problem of '**Omitted Variable bias**

Basically, it is very well possible that all your included variables do not explain the variability of the regression model. You might have missed to include the relevant variable in the model, which leads to incorrect and bias results.

Possibly, your dependent variable  $y$ , may be correlated with the missed variable and hence causing the pains with your model.

However, caution is advised to tackle Omitted variable bias as incorrect inclusion of variables leads to inefficient estimated/models.

So when in doubt, just include the variables and try your luck.



# Linear Regression (OLS) Assumptions

---

## **Normality & Homoscedasticity:**

It comprises of three parts (all for error term):

- Normality
- Zero Mean
- Homoscedasticity

**Normality:** Error term is normally distributed. Note that normal distribution is not required for creating the regression but for making inferences.

Remember the regression table with t-statistics and F-statistics, well all those works because we assume normality of the error term.

What should we do if error term is not normally distributed ?

**The Central Limit Theorem.** For large samples the CLT applies for error term too. Hence we can consider normality as a given for us.



# Linear Regression (OLS) Assumptions

---

## **Normality & Homoscedasticity:**

It comprises of three parts (all for error term):

- Normality
- Zero Mean
- Homoscedasticity

**Zero Mean:** If the mean is not expected to be 0 then the line is not the best fitting one. However, having an intercept solves that problem. Hence in real life, it is unusual to violate this part of the assumption.

**Homoscedasticity:** It means to have equal variance. Basically the error terms should have equal variance one with the other.

**If there is heteroscedasticity in the error term, then look for:**

- Omitted Variable Bias
- Outliers
- Log Transformation





# Linear Regression (OLS) Assumptions

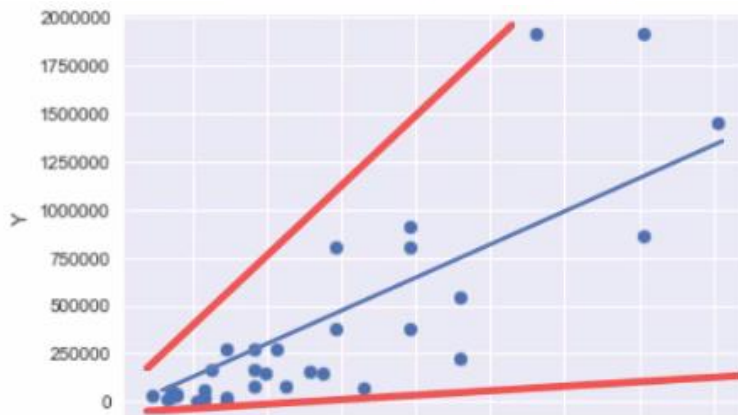
## Normality & Homoscedasticity:

### Log Transformation....

- Do natural log transformation of each dependent variable in Y
- Then create regression of that  $\log(y)$  and the independent Xs

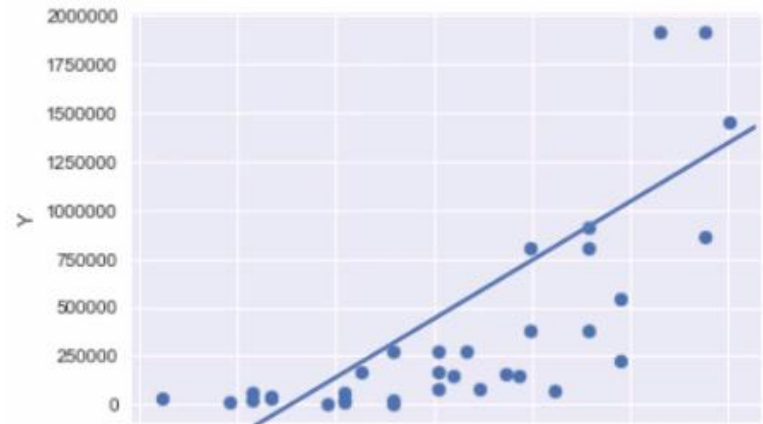
Conversely, you can take log of independent X that is causing the trouble and do the same.

$$\hat{y} = b_0 + b_1 x_1$$



Heteroscedasticity increases  
from left to right

$$\hat{y} = b_0 + b_1 (\log x_1)$$



Heteroscedasticity controlled  
Semi-log model

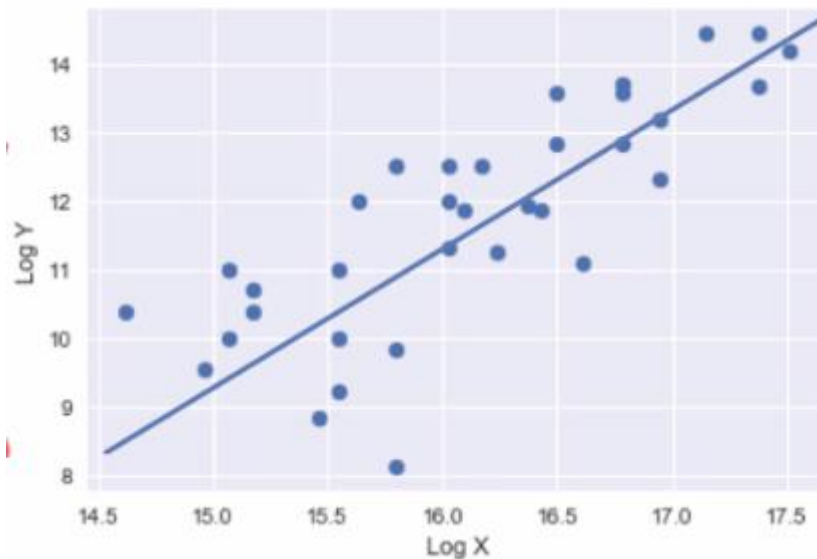
# Linear Regression (OLS) Assumptions

---

**Normality & Homoscedasticity:**

**Log Transformation....**

Or you can do the log transformation for both dependent and independent variables and create regression on that.



**Log-log model**

$$\log \hat{y} = b_0 + b_1(\log x_1)$$

**As X increases by 1 percent,  
Y increases by  $b_1$  percent**

This relationship is known as elasticity

# Linear Regression (OLS) Assumptions

---

**No auto-correlation:** also known as no serial correlation  
This cannot be relaxed.

Errors are assumed to be un-correlated.  
i.e. it assumes the errors should be randomly spread around the regression line.

It is highly unlikely to find serial correlation between errors take at one moment of time, known as cross-sectional data.

However, it is very common in time series data. Think about the stock prices. Every day you have a new quote price for the same stock. Which shows that the errors are not auto-correlated that is errors are randomly spread across the and hence you cannot predict the stock prices.



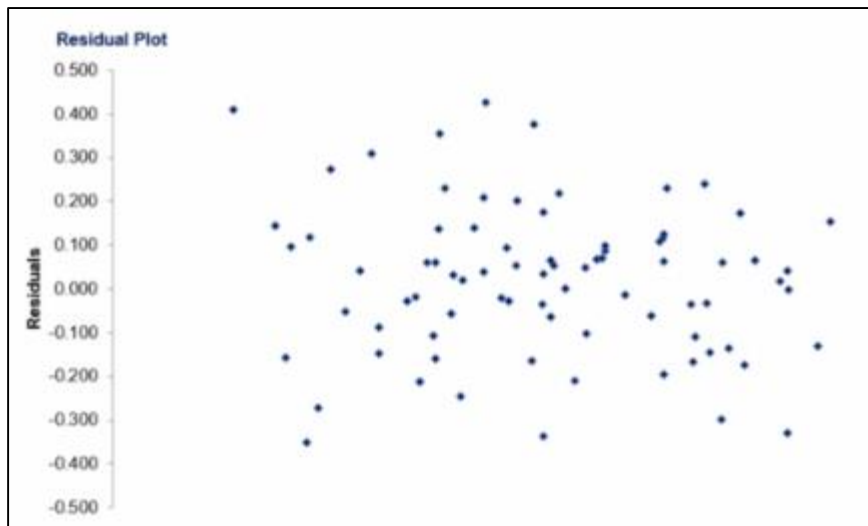
# Linear Regression (OLS) Assumptions

---

**No auto-correlation:** also known as no serial correlation  
This cannot be relaxed.

How to look for auto-correlation ?

Common way is to plot all the residuals on a graph and try to find a pattern. If you cannot find any, you are safe.



# Linear Regression (OLS) Assumptions

---

**No auto-correlation:** also known as no serial correlation  
This cannot be relaxed.

How to look for auto-correlation ?

Another way is **Durbin-Watson test**, which is there in the regression table summary

Generally its values lie between **0 and 4**

**Value = 2 : no auto-correlation**

**Value < 1 or value >3 : Cause for alarm, looks like auto-correlation**

**There is no remedy for this problem. The only thing you can do here is you can avoid using the linear regression in this case.**

You may use other models such as Autoregressive model, Moving average model, Autoregressive Moving Average Model, Autoregressive Integrated Moving Average Model

---



# Linear Regression (OLS) Assumptions

---

**No multi-collinearity:** We observe multi-collinearity when two or more variables have a high correlation

The reason for this assumption is, in multi-collinearity, the relationship is symmetric. Hence the variability of one can be represented by the other. In such cases, there is no need to include both the variables, we can keep any one of them.

**Fixes:** There are 3 ways to do this

1. You can drop one of the two variables
2. You can transform them into one variable
3. You can keep them both and treat them with extreme caution



# Linear Regression with R

---

```
> Z <- lm(y ~ x)
```

```
> Z
```

Call:

lm(formula = y ~ x)

Coefficients:

(Intercept)	x
324.08	-8.83

```
> summary(Z)
```

Call:

lm(formula = y ~ x)

Residuals:

Min	1Q	Median	3Q	Max
-59.26	-28.93	-13.45	25.65	143.36

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	324.08	27.43	11.813	8.25e-13 ***
x	-8.83	1.31	-6.742	1.79e-07 ***

---

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

Residual standard error: 43.95 on 30 degrees of freedom

Multiple R-squared: 0.6024, Adjusted R-squared: 0.5892

F-statistic: 45.46 on 1 and 30 DF, p-value: 1.788e-07



---

# Approaches of Linear Regression





# Approaches of Linear Regression

---

## Popular methods of building models:

1. All in
2. Backward Elimination
3. Forward Selection
4. Bidirectional Elimination
5. Score Comparison

**Note:** Backward, Forward and Bi-direction are called as **Stepwise Regression**. However, sometimes, if we refer Stepwise Regression, generally **Bi-directional Elimination** is considered



# Approaches of Linear Regression

---

## **“All – in” – cases:**

- Prior knowledge of features that all features are important
- Or you have to put all features into modelling
- Or you are preparing for Backward Elimination



# Approaches of Linear Regression

---

## **Backward Elimination:**

**Step 1:** Select a significance level to stay in the model (e.g.  $SL = 0.05$ )

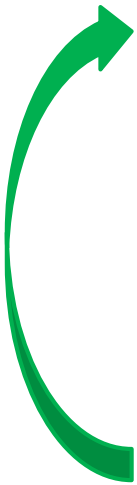
**Step 2:** Fit the full model with all possible predictors (All-in approach)

**Step 3:** Consider the predictor with highest p-value (Basically check for non-significant features/predictors)

**If  $p\text{-value} > SL$ , proceed to Step 4, else FINISH**

**Step 4:** Remove the predictor/feature/variable with highest p-value

**Step 5:** Fit model without this variable(s).



# Approaches of Linear Regression


---

## Forward Selection:

**Step 1:** Select a significance level to stay in the model (e.g.  $SL = 0.05$ )

**Step 2:** Fit all simple regression models ( $y \sim x_n$ ).

Select the one with the lowest p-value



**Step 3:** Keep this variable and fit all possible models with one extra predictor added to the one(s) you already have.

**Step 4:** Consider the predictor with the lowest p-value.

**If  $p < SL$ , go to Step 3, otherwise FINISH.**




# Approaches of Linear Regression

---

## **Bi-direction Elimination: (combines Forward and Backward)**

**Step 1:** Select a significance level to enter and to stay in the model (e.g. SL-Enter = 0.05, SL-Stay = 0.05)



**Step 2:** Perform the next step of forward selection (new variables must have  $p\text{-value} < \text{SL-Enter}$  to enter). Basically add a new variable based on forward selection method.

**Step 3:** Perform all steps of backward elimination (old variables must have  $p\text{-values} < \text{SL-Stay}$  to stay)

**Step 4:** No new variables can enter and variables can exit, then **FINISH**



# Approaches of Linear Regression

---

## Score Comparison:

**Step 1:** Select a criterion (score) of goodness of fit (e.g.  $R^2$  or adjusted  $R^2$ )

**Step 2:** Construct All Possible Regression Models  $2^N - 1$  total combinations (if you have  $n$  features, you create  $2^N - 1$  models)

**Step 3:** Select the one with the best criterion/score

**Step 4: FINISH**

This sounds easy but even if you have 10 columns in your dataset, you create 1023 models. That is very resource intensive. This is usually not preferred.



---

Thank You

