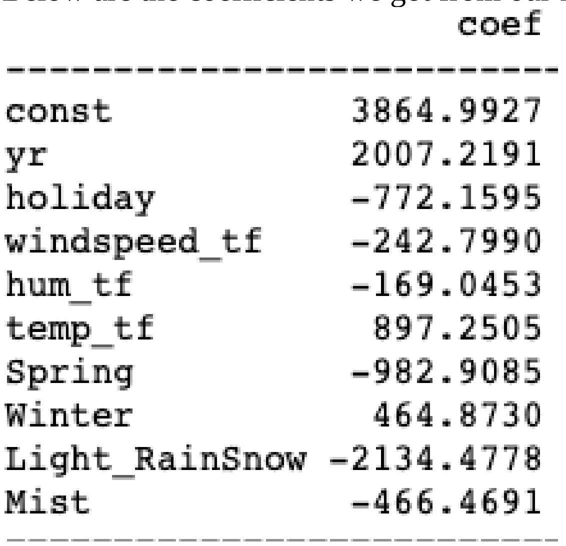
Assignment-based Subjective Questions

1. From your analysis of the categorical variables from the dataset, what could you infer about their effect on the dependent variable?

From our final model: we have **holiday, Spring, Winter, Light_RainSnow and Mist** as the categorical variables.
Below are the coefficients we got from our final model.



```
                        coef
--------------------------
const              3864.9927
yr                 2007.2191
holiday            -772.1595
windspeed_tf       -242.7990
hum_tf             -169.0453
temp_tf             897.2505
Spring             -982.9085
Winter              464.8730
Light_RainSnow    -2134.4778
Mist               -466.4691
--------------------------
```

Using this we can infer the following:

- **Winter** : A coefficient value of ‘464.873’ indicated that a unit increase in Winter variable increases the bike hire numbers by 464.873 units

- **Light_RainSnow** : A coefficient value of ‘-2134.48’ indicated that, a unit increase in Light_RainSnow variable, decreases the bike hire numbers by -2134.48 units

- **Spring**: A coefficient value of ‘-982.908490’ indicated that a unit increase in Spring variable decreases the bike hire numbers by 982.908490 units

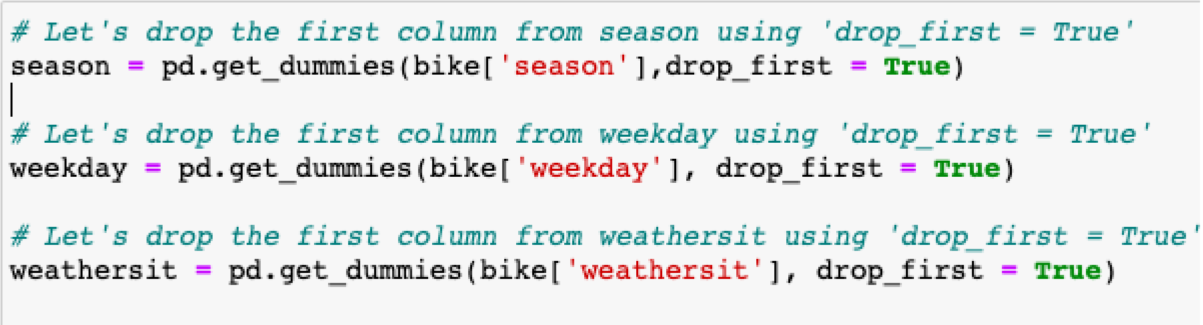
- **holiday** : A coefficient value of ‘-772.159535’ indicated that a unit increase in holiday variable decreases the bike hire numbers by 772.159535 units

- **Mist** : A coefficient value of ‘-466.469063’ indicated that a unit increase in Mist weather variable, decreases the bike hire numbers by 466.469063 units

So, we can conclude that **Winter** feature have **positive** impact on dependent feature ‘**cnt**’ whereas **Light_RainSnow, Spring, holiday and Mist** features have **negative** impact on dependent feature ‘**cnt’**.

2. Why is it important to use drop_first=True during dummy variable creation?

When model has variables that are qualitative (categorical) in nature. Example: season, yr, mnth, weekday, workingday, weathersit. In order to apply any model we need to convert these categorical variables into numeric. We can use get_dummies() to convert it into Dummy variables.



```
# Let's drop the first column from season using 'drop_first = True'
season = pd.get_dummies(bike['season'],drop_first = True)

# Let's drop the first column from weekday using 'drop_first = True'
weekday = pd.get_dummies(bike['weekday'], drop_first = True)

# Let's drop the first column from weathersit using 'drop_first = True'
weathersit = pd.get_dummies(bike['weathersit'], drop_first = True)
```

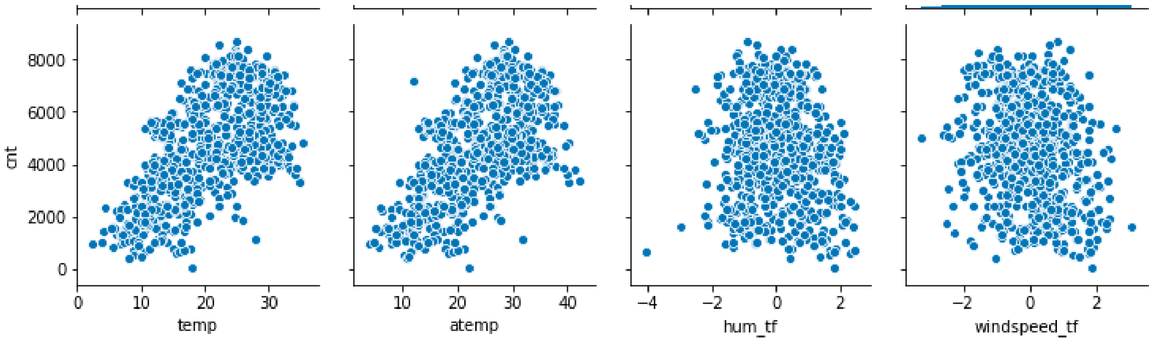
"A Dummy variable is created to represent an attribute with two or more distinct categories/levels."

The Dummy Variable trap is a scenario in which the independent variables are multicollinear — a scenario in which two or more variables are highly correlated; in simple terms one variable can be predicted from the others.

While creating dummies we can get the scenario of dummy variable trap. In order to avoid that, we always declare one less dummy variable(n-1) than the categorical values (n).
No of Dummy variables = categorical values(number of levels) -1

If you don't drop the first column then your dummy variables will be correlated. This may affect the model adversely and the effect is stronger when the cardinality is smaller. For example iterative models may have trouble converging and lists of variable importance may be distorted.

3. Looking at the pair-plot among the numerical variables, which one has the highest correlation with the target variable?

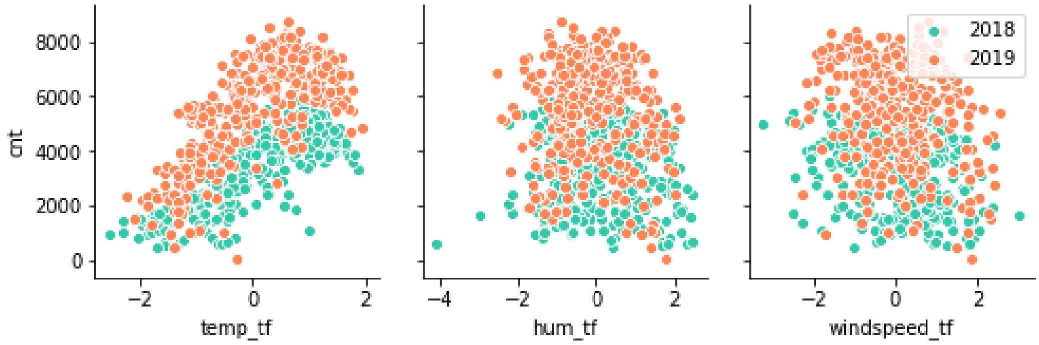


As we can observe from the above pair-plot, temp and atemp has the highest positive correlation with 'cnt' whereas hum_tf has the negative correlation with target variable i.e. cnt. Also, windspeed_tf doesn't show much correlation with target variable.

4. How did you validate the assumptions of Linear Regression after building the model on the training set?

Assumption 1 : Linear relationship between Independent and dependent variables.

As we can observe from the below plot, there is linear relationship between independent features temp_tf, hum_tf, windspeed_tf and dependent feature i.e. cnt



Assumption 2: No multi-collinearity in independent variables.

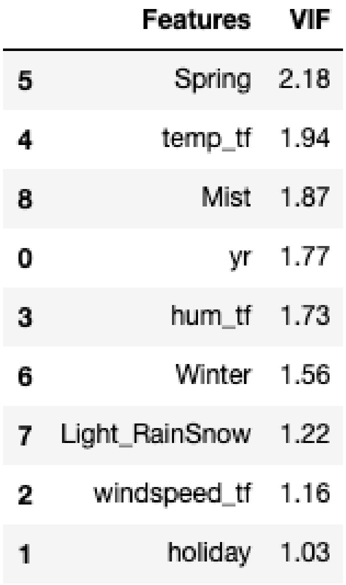
Two variables are said to be correlated when one variable changes, the other variable also changes in fixed proportions. They are said to be perfectly correlated when they have pearson correlation coefficient between them as +1 or -1.

Multicollinearity can be checked using Variance Inflation factor(VIF). VIF is calculated using the below formula.

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}$$

VIF > 5 is generally considered problematic and VIF > 10 suggests a definite presence of collinearity

In our final model we are getting vif<3, means there is no multicollinearity.

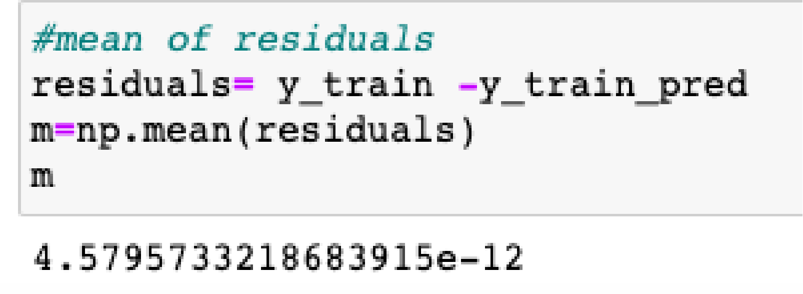


	Features	VIF
5	Spring	2.18
4	temp_tf	1.94
8	Mist	1.87
0	yr	1.77
3	hum_tf	1.73
6	Winter	1.56
7	Light_RainSnow	1.22
2	windspeed_tf	1.16
1	holiday	1.03

Assumption 3 : Mean of the residuals should be Zero

Residuals refer to the difference between actual value and predicted value

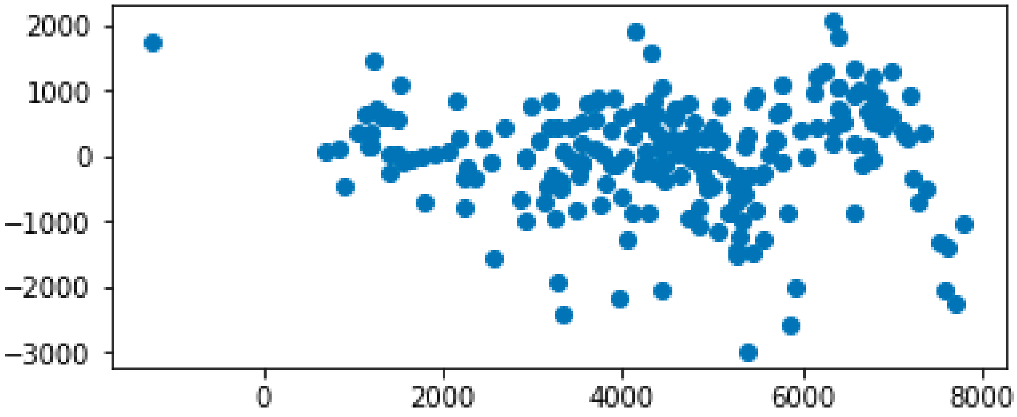
Based on the final model, we are getting residual mean as 0



```
#mean of residuals
residuals= y_train -y_train_pred
m=np.mean(residuals)
m
```

```
4.5795733218683915e-12
```

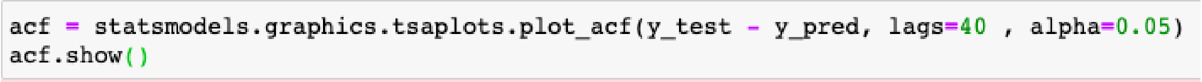
Assumption 4 : Standard Deviation of the residuals should be constant (Homoscedasticity)
The variance of the errors should be consistent for all the observations. This condition is known as homoscedasticity



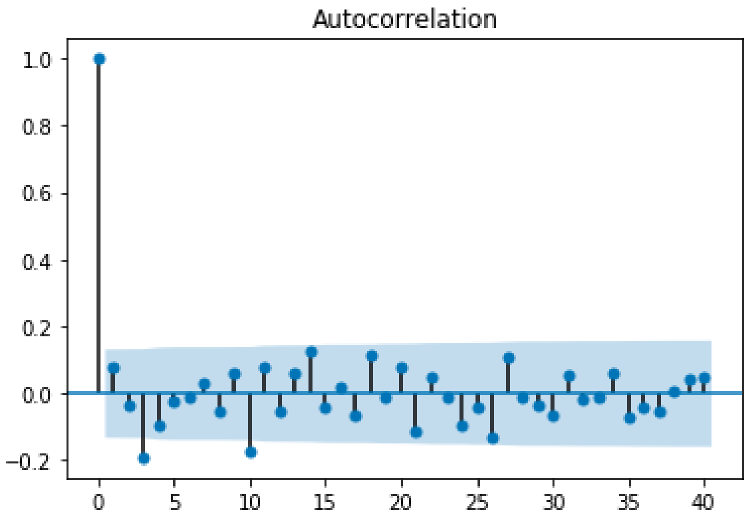
There is no visible pattern in the residual values, thus homoscedasticity is preserved.

Assumption 5 : No auto-correlation between the residuals.

Residual of one observation should not predict the next observation. This problem is also known as auto correlation.
As it can be observed from the below plot that there is no auto correlation between the residuals.

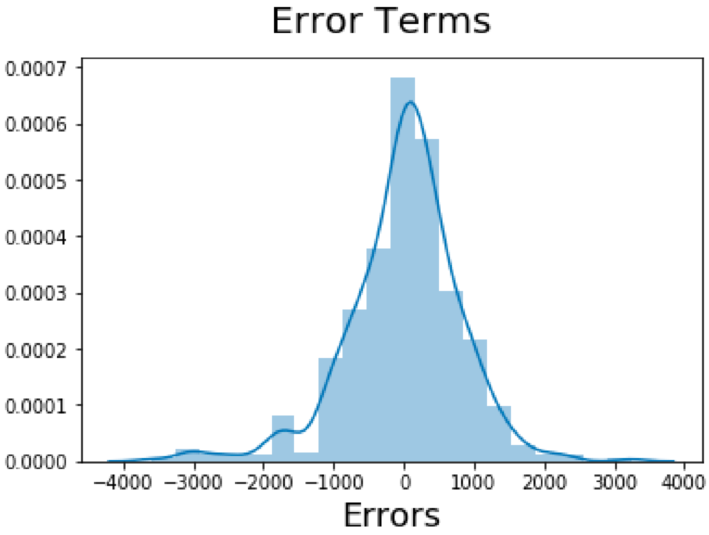


```
acf = statsmodels.graphics.tsaplots.plot_acf(y_test - y_pred, lags=40 , alpha=0.05)
acf.show()
```



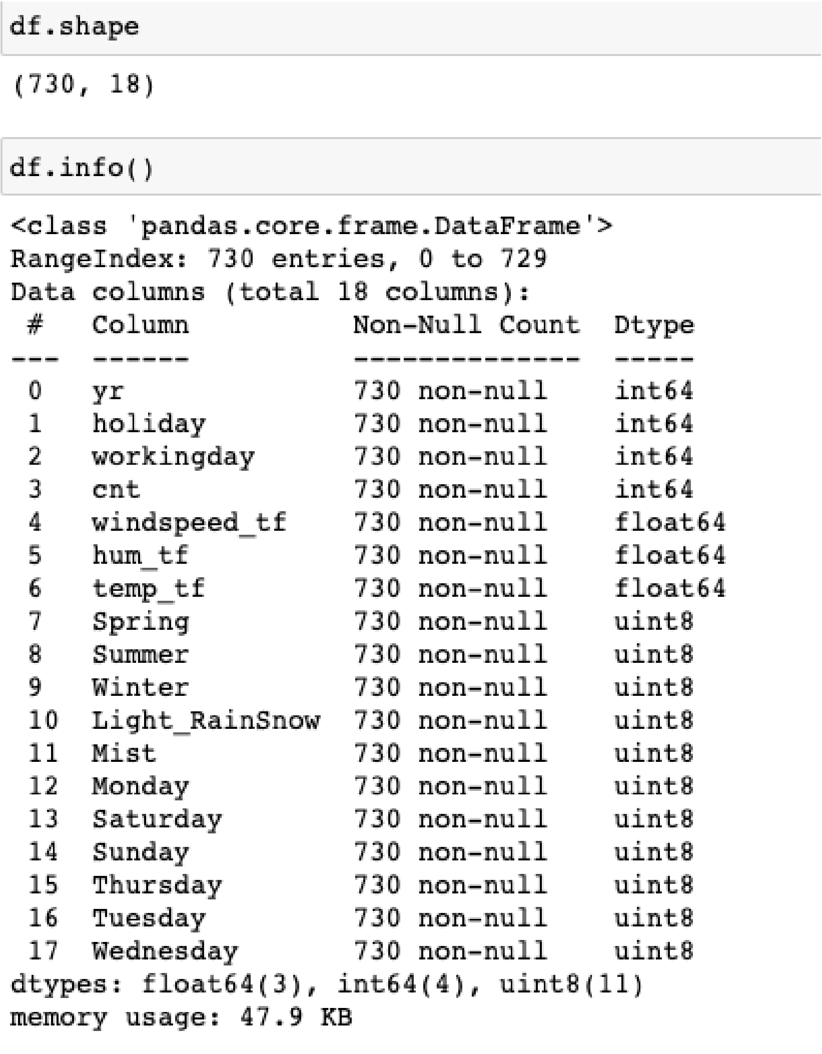
Assumption 6 : Residuals should be normally distributed.

Residuals are normally distributed in our model as shown below in the fig



Assumption 7: Number of observations should be greater than number of independent variables.

We are having 730 rows and 17 independent features in our dataset as ‘cnt’ is dependent feature. So the assumptions holds true



```
df.shape
```

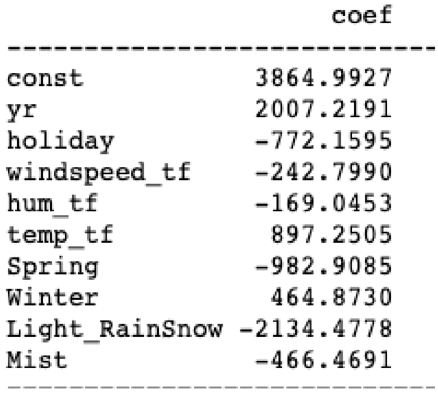
```
(730, 18)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 730 entries, 0 to 729
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   yr               730 non-null    int64  
 1   holiday          730 non-null    int64  
 2   workingday       730 non-null    int64  
 3   cnt              730 non-null    int64  
 4   windspeed_tf     730 non-null    float64
 5   hum_tf           730 non-null    float64
 6   temp_tf          730 non-null    float64
 7   Spring           730 non-null    uint8  
 8   Summer           730 non-null    uint8  
 9   Winter           730 non-null    uint8  
 10  Light_RainSnow   730 non-null    uint8  
 11  Mist             730 non-null    uint8  
 12  Monday           730 non-null    uint8  
 13  Saturday         730 non-null    uint8  
 14  Sunday           730 non-null    uint8  
 15  Thursday         730 non-null    uint8  
 16  Tuesday          730 non-null    uint8  
 17  Wednesday        730 non-null    uint8  
dtypes: float64(3), int64(4), uint8(11)
memory usage: 47.9 KB
```

5. Based on the final model, which are the top 3 features contributing significantly towards explaining the demand of the shared bikes?

From our final model 7 we are getting top 3 features as **yr, temp_tf and Light_RainSnow**



```
                      coef
 ----------------------------
 const            3864.9927
 yr               2007.2191
 holiday          -772.1595
 windspeed_tf     -242.7990
 hum_tf           -169.0453
 temp_tf           897.2505
 Spring           -982.9085
 Winter            464.8730
 Light_RainSnow  -2134.4778
 Mist             -466.4691
 ----------------------------
```

- Year (yr)

A coefficient value of ‘2007.219125’ indicated that a unit increase in yr variable, increases the bike hire numbers by 2007.219125 units

- Temperature (temp_tf)
A coefficient value of ‘897.250477’ indicated that a unit increase in temp variable, increases the bike hire numbers by 897.250477 units

- Light Rain & Snow (weathersit =3)
A coefficient value of ‘-2134.477790’ indicated that, a unit increase in Light_RainSnow variable, decreases the bike hire numbers by -2134.477790 units

It is recommended to give importance to these three variables while planning to achieve maximum bike rental booking.
As high temperature and good weather positively impacts bike rentals, it is recommended that bike availability and promotions to be increased during these months to further increase bike rentals.

General Subjective Questions

1. Explain the linear regression algorithm in detail.

Regression is the term is defined as a measure of the relation between an output variable and the input variable(s). Hence, the Linear Regression assumes a linear relationship between the former and the latter.
Depending upon the number of input variables, Linear Regression can be classified into two categories:
1. Simple Linear Regression (Single Input Variable)
2. Multiple Linear Regression (Multiple Input Variables)

In linear regression the we explore the relation between input and target with a linear equation. For a simple linear regression model with only one feature the Equation becomes:

$$Y=W1*X+b$$

- Y=Predicted value/Target Value
- X=Input
- W1=Gradient/slope/Weight
- b=Bias

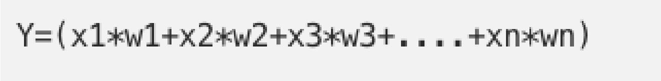
I.e. straight line (Y=m X + c where c is the intercept and m is the slope)

***W1 and b** are the parameters to adjust the straight line to get the best fit. By adjusting the W1 and b we get the algorithm to get the most optimized results.*

Multiple Regression:

This method uses more than one independent variable to predict a dependent variable by fitting a best linear relationship.
In case of Multiple Regression, the parameters can be found in the same way as that in the case of simple linear regression, by minimising the cost function using:

Now we have a set of input features X={x1,x2,x3,….,xn} and weights associated with it W={w1,w2,w3,….wn}.
Thus, the equation becomes:



```
Y=(x1*w1+x2*w2+x3*w3+....+xn*wn)
```

Or

$$Y = \sum_{1}^{n}(w1 * x1)$$

With bias consideration:

Y= (x1*w1+x2*w2+x3*w3+….+xn*wn)+b

$$Y = \sum_{1}^{n}(w1 * x1) + b$$

Or

In order to determine the weights and bias, we can use MSE
Because weights are measured by MSE (Mean Squared Error) and adjusting them to get a best possible Linear line.

MSE = average of ((predicted value — actual value of i th value of y)^2)

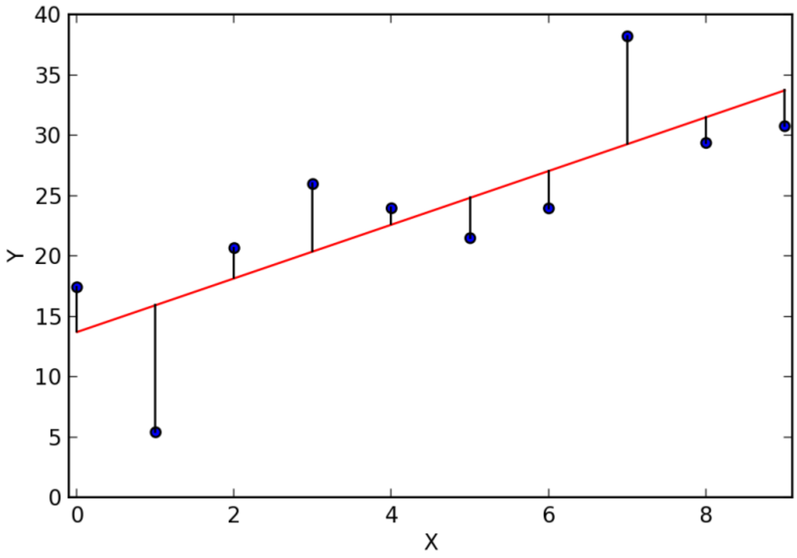


Image1 from(https://cdn-images-1.medium.com/max/1200/0*FjKhbw6Va8O8bCkF.png)

In the above image, the ‘**red line**’ is our linear regression line or our **predicted value(y′).** And the ‘**blue**’ points are our given data or **actual value**. The average of square of distance from the blue points (actual value) to the red line (predicted value) must be minimum to get the best fit regression line.
Thus, can be represented as

$$1/n \sum (y' - y_i)^2$$

MSE where y’ is predicted value yi is actual value

<u>Goal: To gain optimal result we need to minimize MSE</u>

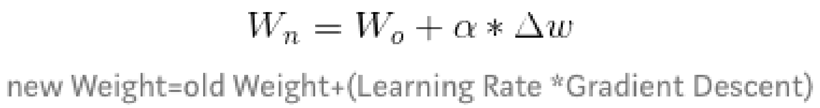
To minimize MSE we use **<u>Gradient descent</u>** to find the weights after MSE or error rate calculation.

Gradient Descent: Given a function defined by a set of parameters, Gradient Descent starts with an initial set of parameter values and iteratively moves towards a set of values that minimise the function. This iterative minimisation is done using calculus, taking steps in the *negative direction of the function gradient*.

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Now after we get the Gradient descent we need to update the weight every time until we get the best fitted value

new Weight=old Weight + (Learning Rate *Gradient Descent)

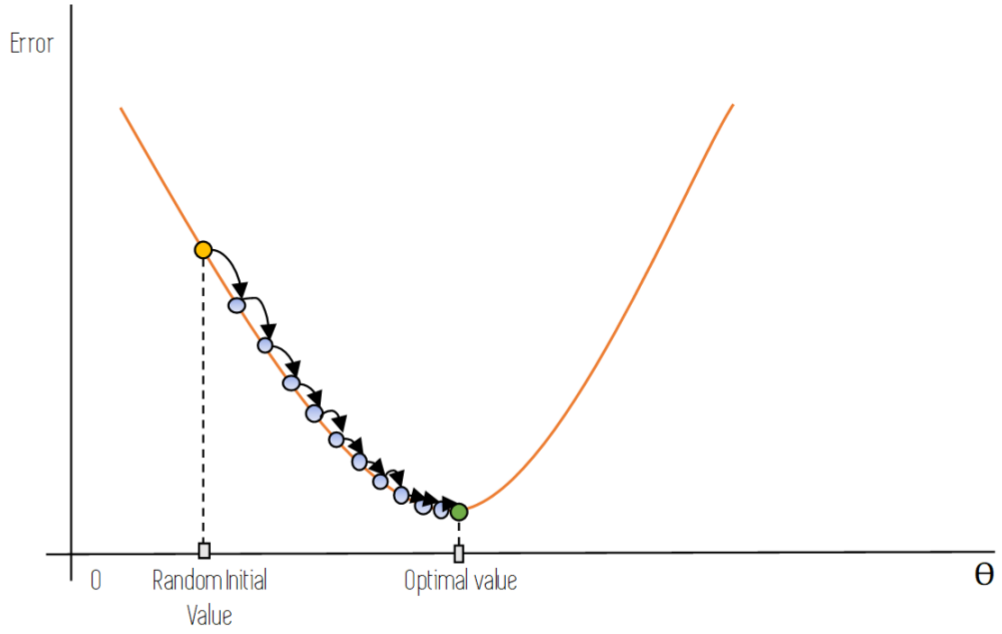


$$W_n = W_o + \alpha * \Delta w$$

new Weight=old Weight+(Learning Rate *Gradient Descent)

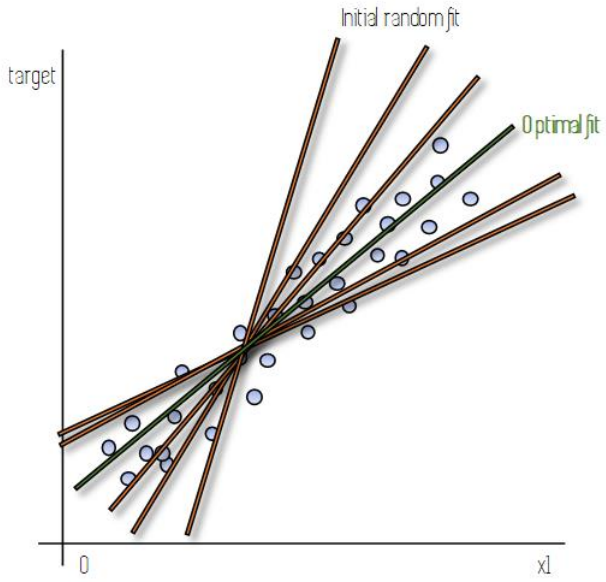
Whereas learning rate is fixed value ranging between 0–1.

Training with Gradient Descent
The following figure shows graphically how this is done: we start at the orange point, which is the initial random value of the model parameters. After one iteration of gradient descent, we move to the blue point which is directly right and down from the initial orange point: we have gone in the direction of descending gradient.



Gradient descent
Iteration after iteration, we travel along the orange error curve, until we reach the optimal value, located at the bottom of the curve and represented in the figure by the green point.

If we had a linear model with only one feature (x1) just so that we can plot it easily. In the following figure the blue points represent our data instances, for which we have the value of the target and the value of the one feature.

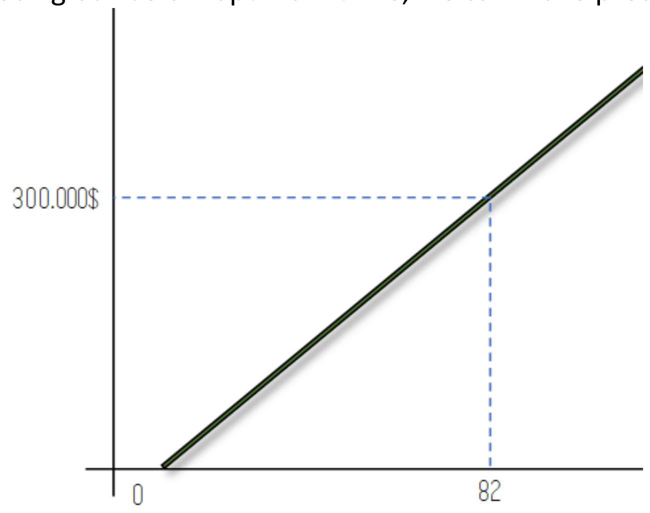


Graphical representation of the different iterations of a linear regression model with one feature (x1)

When we train a model using **gradient descent** is that we start by fitting a line to our data (the Initial random fit line) that is not a very good representation of it. After each iteration of gradient descent, as the parameters get updated, this line changes its slope and where it cuts the y axis. This process is repeated until we reach a set of parameter values that are good enough (these are not always the optimal values) or until we complete a certain number of iterations.
These parameters are represented by the green Optimal fit line.

$$\overset{\text{target}}{\downarrow}\atop{\hat{y}} = \underset{\text{y-cut}}{\underset{\uparrow}{\Theta_0}} + \overset{\text{slope}}{\overset{\downarrow}{\Theta_1}}\underset{\text{feature}}{\underset{\uparrow}{x}}1$$

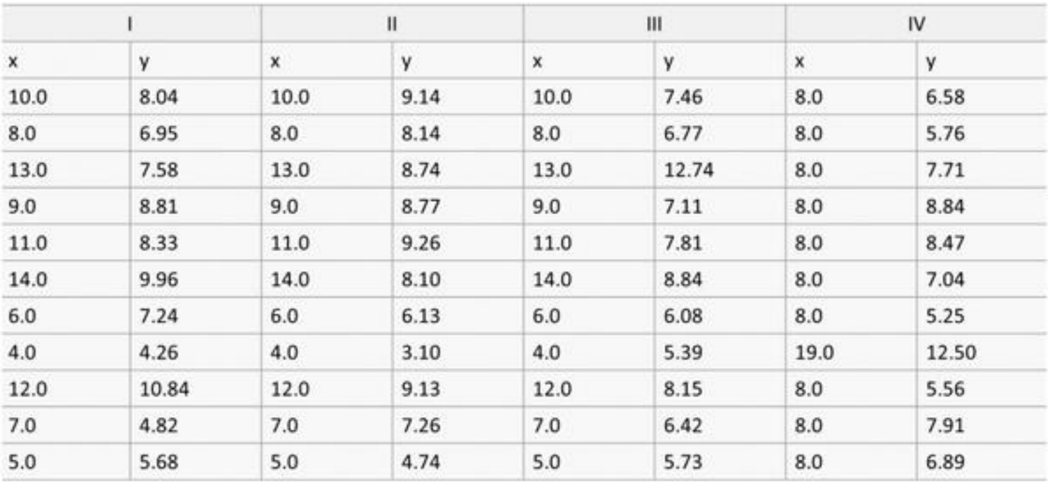
Equation of linear regression model with only one feature.
Using our below optimal fit line, we can make prediction.



2. Explain the Anscombe’s quartet in detail.

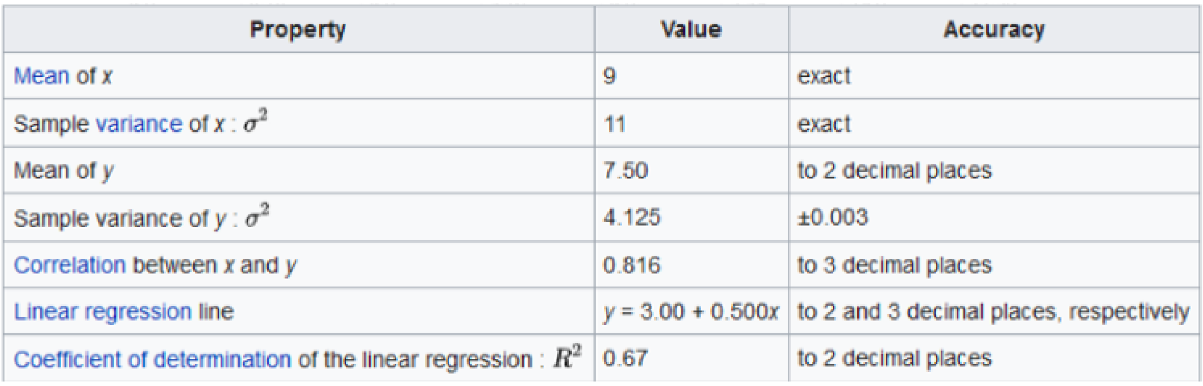
Anscombe’s Quartet can be defined as a group of four data sets which are nearly identical in simple descriptive statistics, but there are some peculiarities in the dataset that fools the regression model if built. They have very different distributions and appear differently when plotted on scatter plots. It was constructed by statistician Francis Anscombe to illustrate the **importance of plotting the graphs before analysing and model building**, and the effect of other observations on statistical properties.

It comprises of 4 datasets that have nearly identical descriptive statistics, yet have very different distributions and appear very different when visualized.
Anscombe’s Quartet was developed to highlight the **importance of data visualization**. Each data set consists of 11 (x, y) points as shown below.



I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Data Sets



Property	Value	Accuracy
Mean of *x*	9	exact
Sample variance of *x* : σ^2	11	exact
Mean of *y*	7.50	to 2 decimal places
Sample variance of *y* : σ^2	4.125	±0.003
Correlation between *x* and *y*	0.816	to 3 decimal places
Linear regression line	*y* = 3.00 + 0.500*x*	to 2 and 3 decimal places, respectively
Coefficient of determination of the linear regression : R^2	0.67	to 2 decimal places

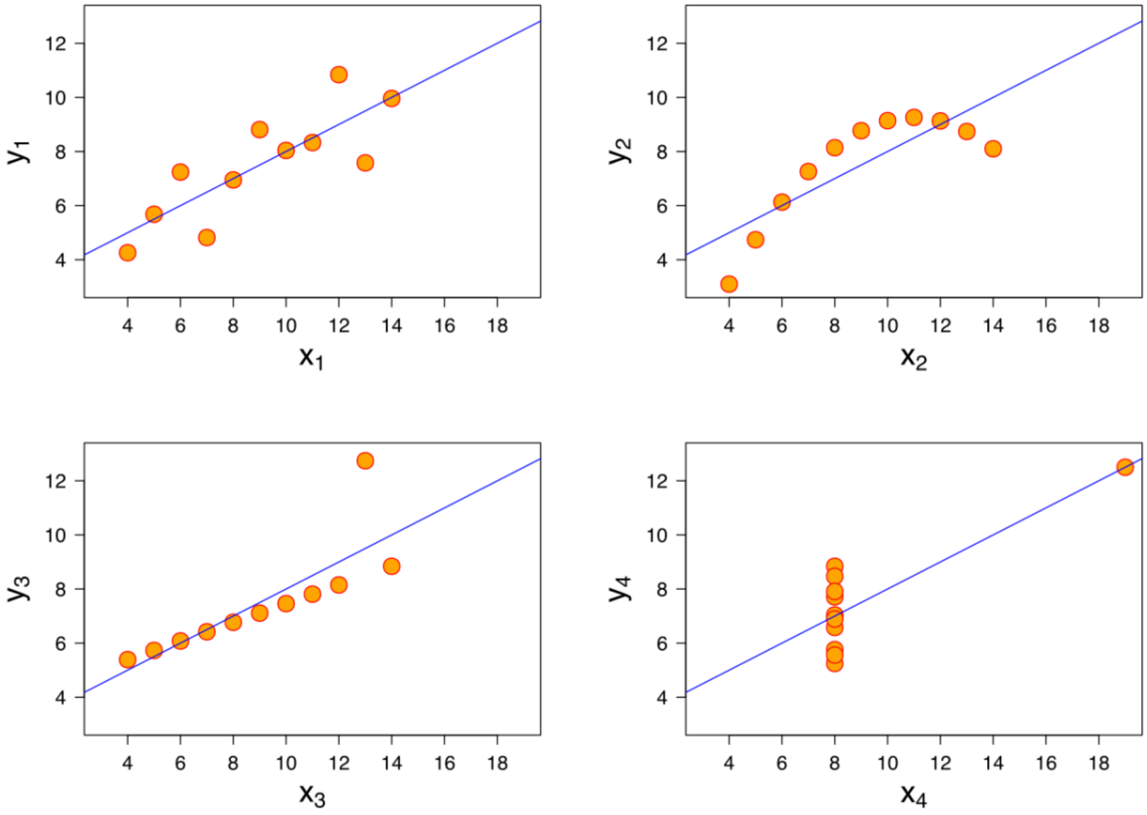
Quartet’s Summary Stats

The summary statistics show that the means and the variances were identical for x and y across the groups :
Mean of x is 9 and mean of y is 7.50 for each dataset.
Similarly, the variance of x is 11 and variance of y is 4.13 for each dataset

The correlation coefficient (how strong a relationship is between two variables) between x and y is 0.816 for each dataset

Visualization of the above dataset:

When we plot these four datasets on an x/y coordinate plane, we can observe that they show the same regression lines as well but each dataset is telling a different story :



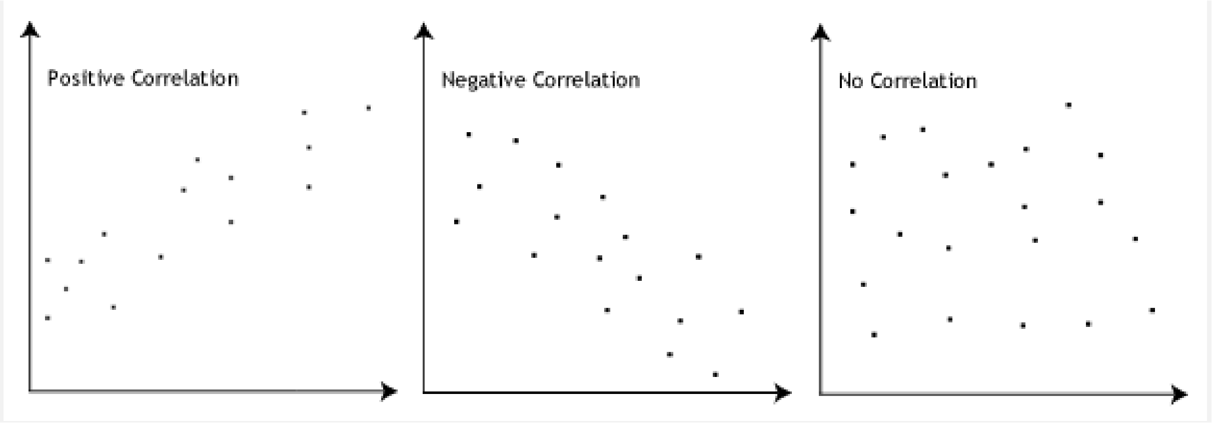
- The first scatter plot is a plot for first data set , it appears to have a simple linear relationship.
- The second scatter plot is a plot for the second data set , it is evident that the data set is not linear. Perhaps, using a polynomial function will serve as a good fit.
- The third plot is a plot for the third data set , The linear curve is influenced by a single outlier .The calculated regression is offset by the one outlier.
- In the fourth plot , the variables don’t seem to have any relationship. However, the regression line is heavily influenced by one single data point , and is enough to produce high correlation coefficient

This quartet tell us about the importance of visualising the data before applying various algorithms to build models out of them which suggests that the data features must be plotted in order to see the distribution of the samples that can help you identify the various anomalies present in the data like outliers, diversity of the data, linear separability of the data, etc. Also, the Linear Regression can be only be considered a fit for the data with linear

3. What is Pearson’s R?

Correlation is a measure of how well two variables are related to each other. There are positive as well as negative correlation.

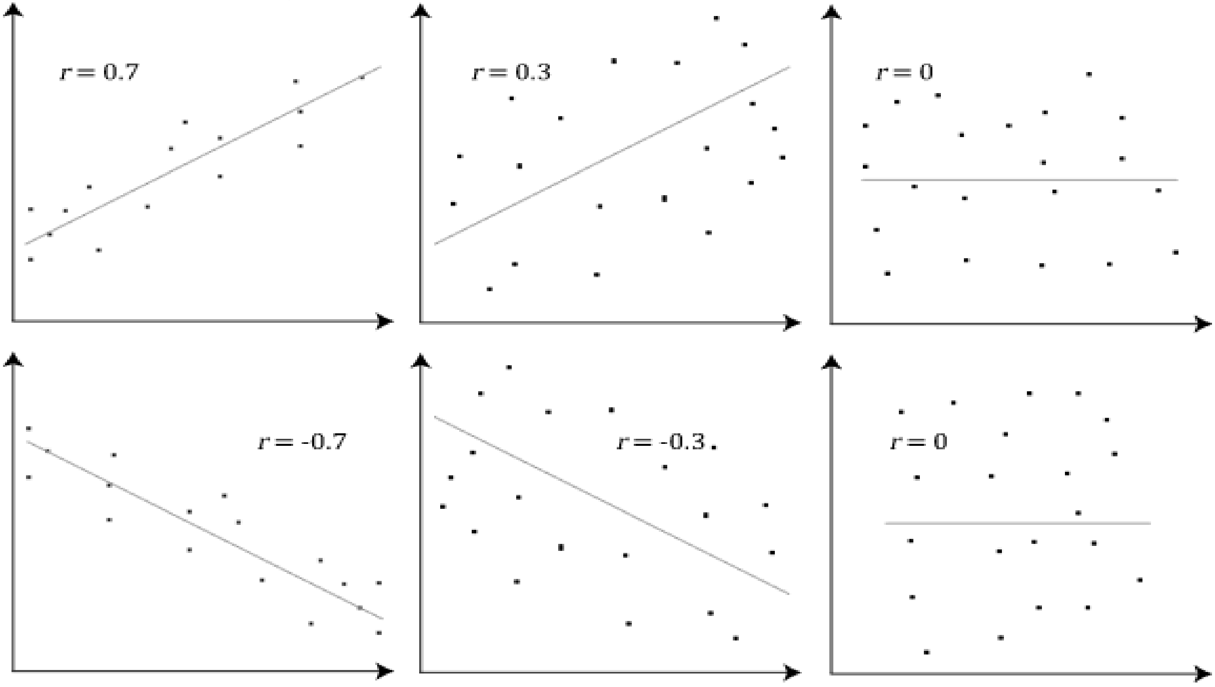
- **Positive Correlation**: It refers to the extent to which the two variables increases or decreases in parallel (i.e. directly proportional, one increases other will increase, one decreases other will follow the same).
- **Negative Correlation**: It refers to the extent to which one of the two variables increases as the other decreases (i.e. inversely proportional, one increases other will decrease or if one decreases other will increase).



Pearson Correlation also known as Pearson Product Moment Correlation (PPMC), attempts to draw a line to best fit through the data of the given two variables, and the Pearson correlation coefficient “r” indicates how far away all these data points are from the line of best fit.

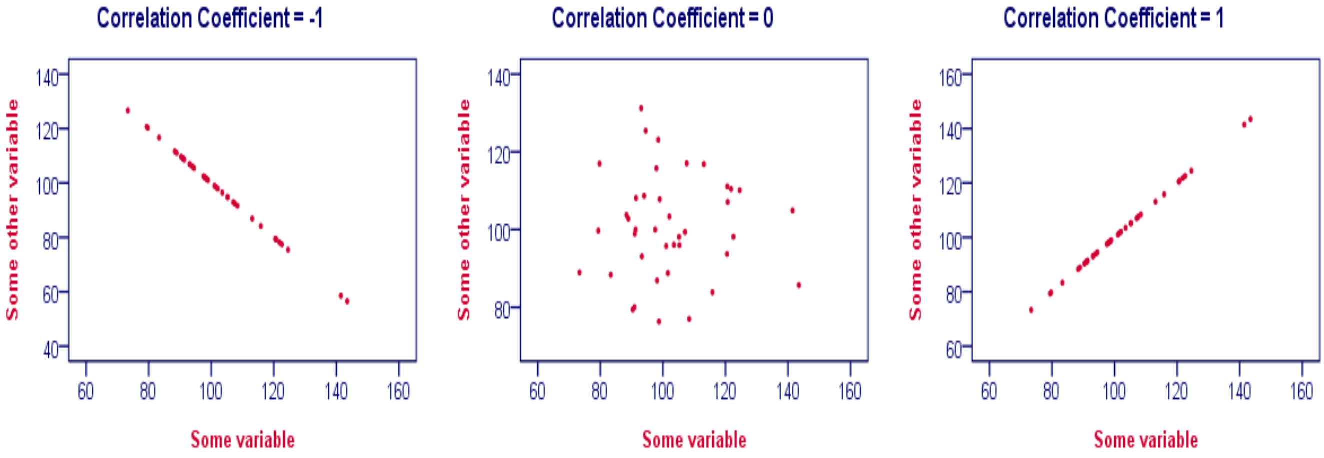
The value of “r” ranges from +1 to -1 where:

r= +1/-1 represents that al our data points lie on the line of best fit only i.e there is no data point which shows any variation from the line of best fit.



Pearson correlations are only suitable for quantitative variables(including dichotomous variables).
For **ordinal variables**, use the Spearman correlation
For **nominal variables**, use Cramér's V.
Correlation Coefficients lies between -1 and 1.

Scenarios:
Correlations Coefficient = -1, indicates that the data points in a scatter plot lie exactly on a straight descending line; the two variables are perfectly negatively linearly related.
Correlation of 0 means that two variables don't have any linear relation whatsoever. However, some non-linear relation may exist between the two variables.
Correlation coefficient= 1, means that two variables are *perfectly* positively linearly related; the dots in a scatter plot lie exactly on a straight ascending line.



Pearson Correlation – Formula

Pearson correlation between variables X and Y is calculated by

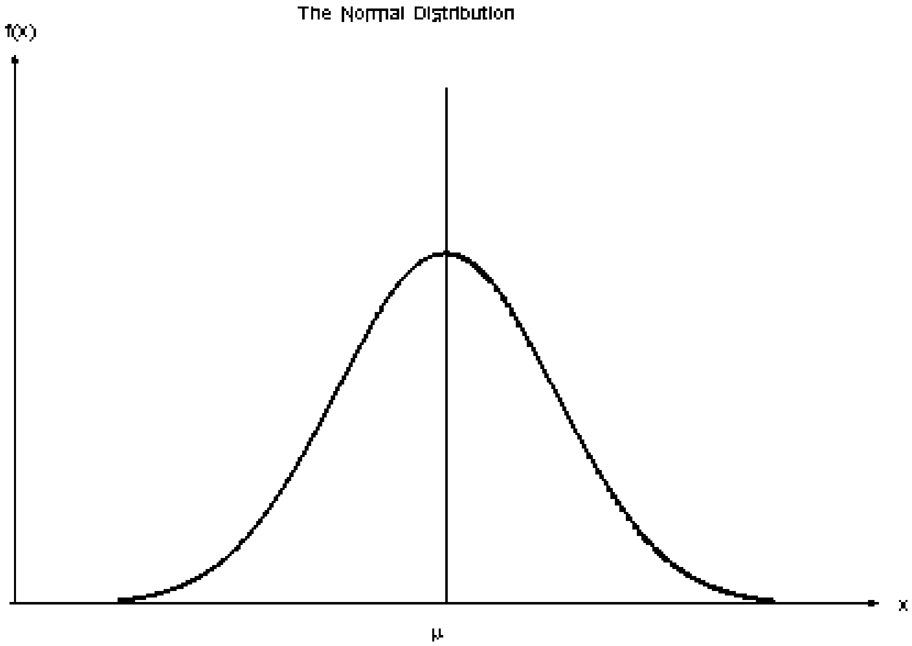
$$r_{XY} = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}}$$

The formula basically comes down to dividing the **covariance** by the product of the **standard deviations**. Since a coefficient is a number divided by some other number our formula shows why we speak of a correlation *coefficient*.

Correlation Test – Assumptions

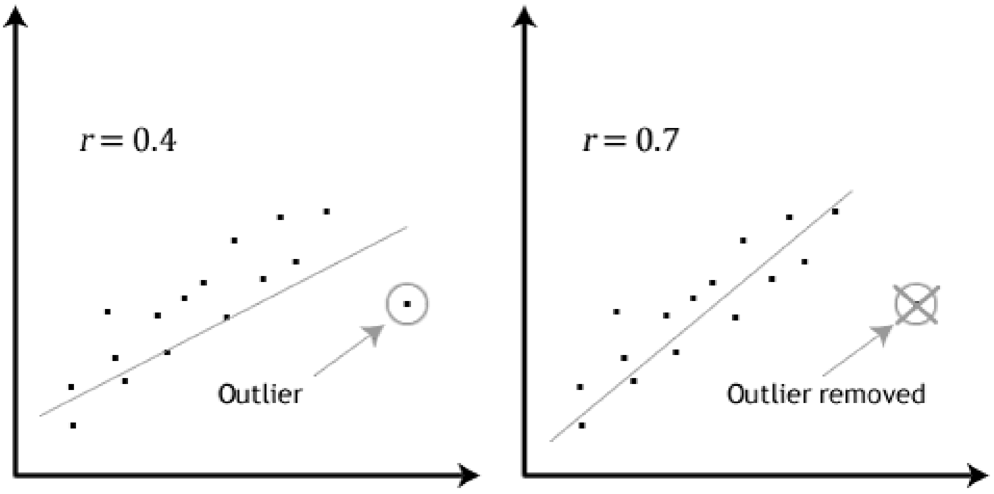
The statistical significance test for a Pearson correlation requires these assumptions:
1. **Normal Distribution**

For the Pearson r correlation, both variables should be normally distributed. i.e. the normal distribution describes how the values of a variable are distributed. This is sometimes called the ‘Bell Curve’ or the ‘Gaussian Curve’. A simple way to do this is to determine the normality of each variable separately using the Shapiro-Wilk Test.



2. **Outliers**

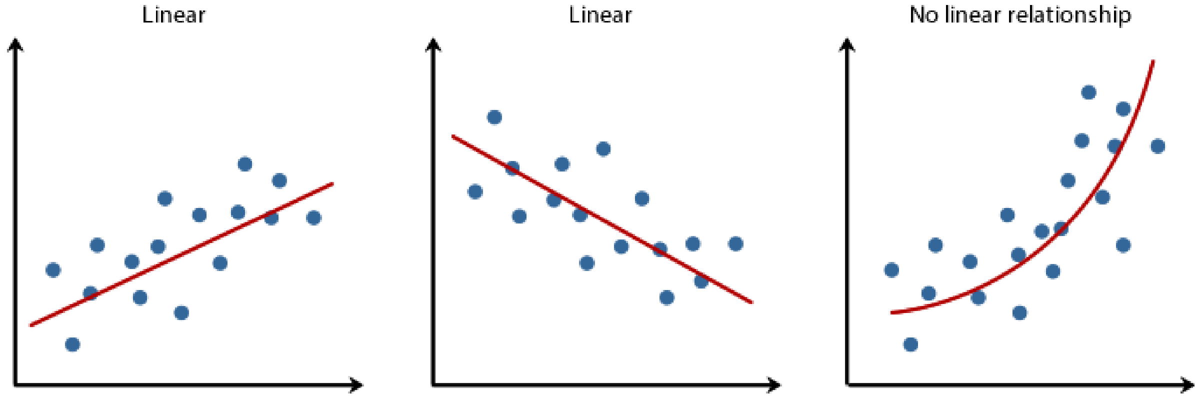
There should be no significant outliers. We all know what outliers are but we don’t know the effect of outliers on Pearson’s correlation coefficient, r. Pearson’s correlation coefficient, r, is very sensitive to outliers, which can have a very large effect on the line of best fit and the Pearson correlation coefficient. This means — including outliers in your analysis can lead to misleading results.



3. Each variable should be continuous i.e. interval or ratios for example weight, time, height, age etc. If one or both of the variables are ordinal in measurement, then a Spearman correlation could be conducted instead.

4. **Linear and non-Linear Relationships**

The two variables have a linear relationship. Scatter plots will help you tell whether the variables have a linear relationship. If the data points have a straight line (and not a curve), then the data satisfies the linearity assumption. If the data you have is not linearly related you might have to run a non-parametric .



Copyright 2014. Laerd Statistics.

5. The observations are paired observations. That is, for every observation of the independent variable, there must be a corresponding observation of the dependent variable. For example if you're calculating the correlation between age and weight. If there are 12 observations of weight, you should have 12 observations of age. i.e. no blanks.

6. **Homoscedascity**.
Homoscedascity simply refers to 'equal variances'. A scatter-plot makes it easy to check for this. If the points lie equally on both sides of the line of best fit, then the data is homoscedastic

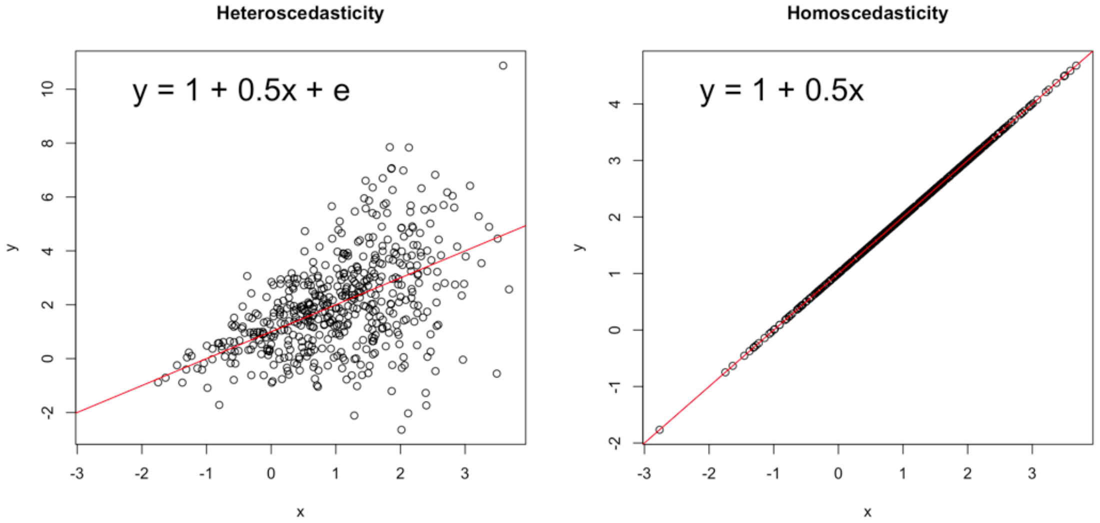
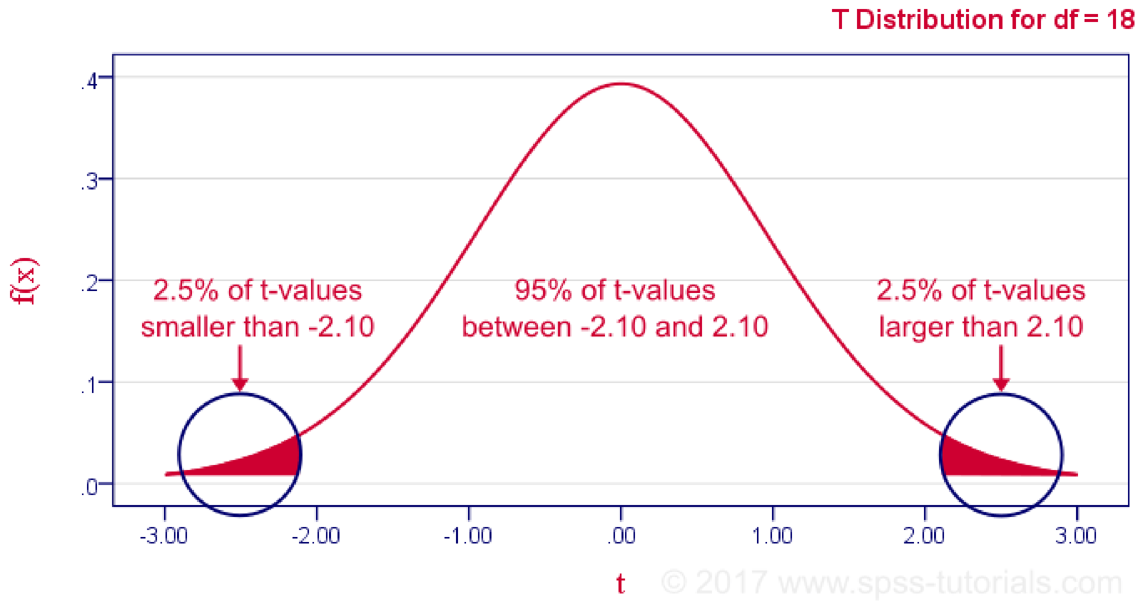


Fig. Heteroscedasticity vs homoscedasticity

Pearson Correlation - Sampling Distribution
In our example, the sample size N was 20. So if we meet our assumptions, T follows a t-distribution with df = 18 as shown below.



This distribution tells us that there's a 95% probability that -2.1 < t < 2.1, corresponding to -0.44 < r < 0.44.

Conclusion :if N = 20, there's a 95% probability of finding -0.44 < r < 0.44.There's only a 5% probability of finding a correlation outside this range. That is, such correlations are statistically significant at α = 0.05 or lower: they are (highly) unlikely and thus refute the null hypothesis of a zero population correlation.

4. What is scaling? Why is scaling performed? What is the difference between normalized scaling and standardized scaling?

Scaling is a technique to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information.

Scaling is essential for machine learning algorithms that calculate distances between data. It is performed during the data pre-processing to handle highly varying magnitudes or values or units.
Most of the times, collected data set contains features highly varying in magnitudes, units and range. If scaling is not done then algorithm only takes magnitude in account and not units hence incorrect modelling. To solve this issue, we have to do scaling to bring all the variables to the same level of magnitude.

Scaling just affects the coefficients and none of the other parameters like t-statistic, F-statistic, p-values, R-squared, etc.

Why use Scaling?

Gradient descent converges much faster with feature scaling than without it.
Many classifiers (like KNN, K-means) calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be

governed by this particular feature. So, the range of features should be scaled so that each feature contributes approximately proportionately to the final distance.
However, every dataset does not require features scaling. It is required only when features have **different ranges.**
This can be achieved using two widely used techniques.
Normalization
Standardization

Normalization:

Normalization is not required for every dataset, you have to sift through it and make sure if your data requires it and only then continue to incorporate this step in your procedure.

Apply Normalization if you are not very sure if the data distribution is Gaussian/ Normal/ bell-curve in nature. **Normalization will help in reducing the impact of non-gaussian attributes on your model.**

Normalization (also called, **Min-Max normalization**) is a scaling technique such that when it is applied the features will be **rescaled so that the data will fall in the *range of [0,1]***
Normalized form of each feature can be calculated as follows:

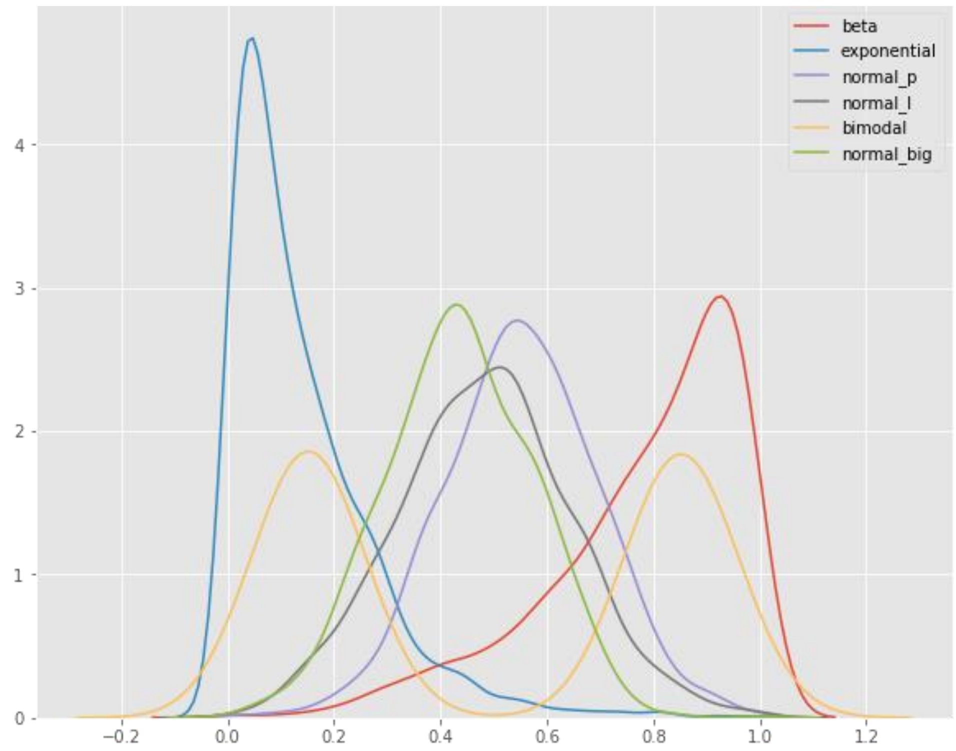
$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

where **x** is the original value and **x`** is the normalized value.
For each value in a feature, Min-Max scaler subtracts the minimum value in the feature and then divides by the range. The range is the difference between the original maximum and original minimum.

MinMaxScaler preserves the shape of the original distribution. It doesn’t meaningfully change the information embedded in the original data.

sklearn.preprocessing.MinMaxScaler helps to implement normalization in python

**MinMaxScaler doesn’t reduce the importance of outliers.
The default range for the feature returned by MinMaxScaler is 0 to 1.
Here’s the kdeplot after MinMaxScaler has been applied.



It can be observed that the features are all on the same relative scale. The relative spaces between each feature's values have been maintained.

Standardization Scaling:

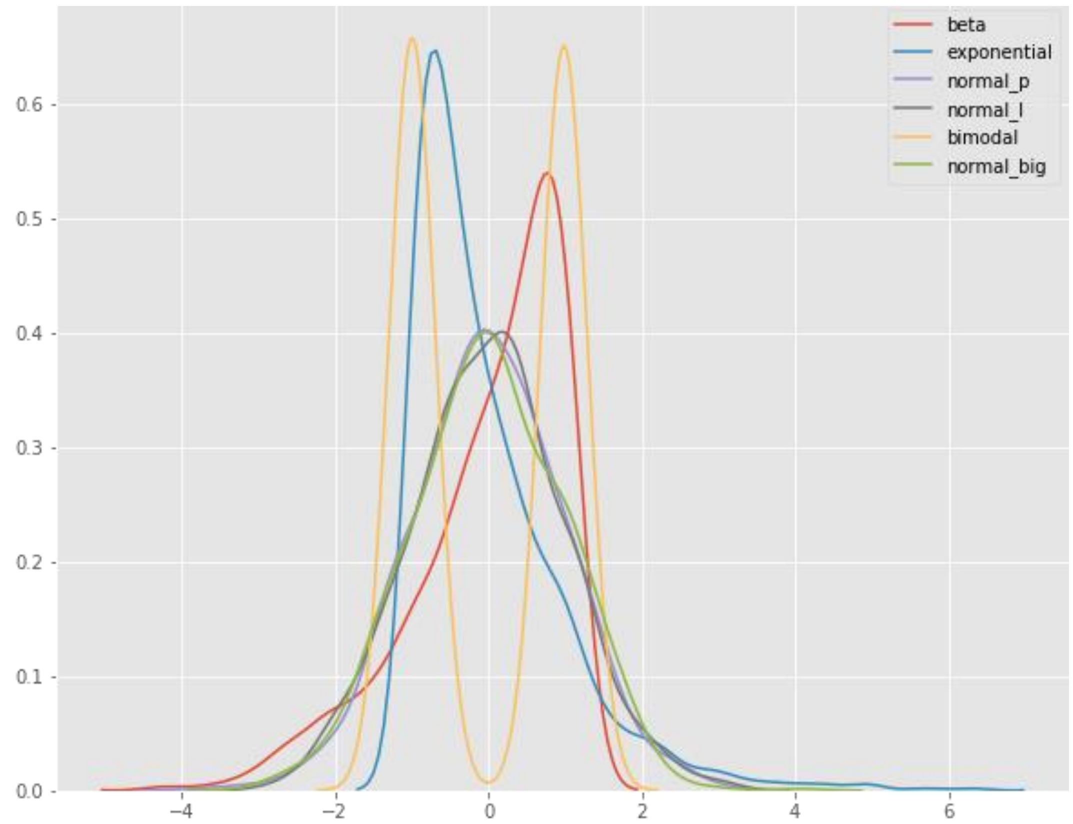
Standardization (also called, Z-score normalization) is a scaling technique such that when it is applied the features will be rescaled so that they'll have the properties of a standard normal distribution with mean, **μ=0** and standard deviation, **σ=1**; where μ is the mean (average) and σ is the standard deviation from the mean.
Standard scores (also called z scores) of the samples are calculated as follows:

$$z = \frac{x - \mu}{\sigma}$$

This scales the features in a way that they range between [-1,1]

sklearn.preprocessing.scale -helps to implement standardization in python.

StandardScaler results in a distribution with a standard deviation equal to 1. The variance is equal to 1 also, because *variance = (standard deviation)^2*.
StandardScaler makes the mean of the distribution 0. About 68% of the values will lie between -1 and 1.



In the plot above, you can see that all four distributions have a mean close to zero and unit variance.
StandardScaler does distort the relative distances between the feature values, so it's generally my second choice in this family of transformations.

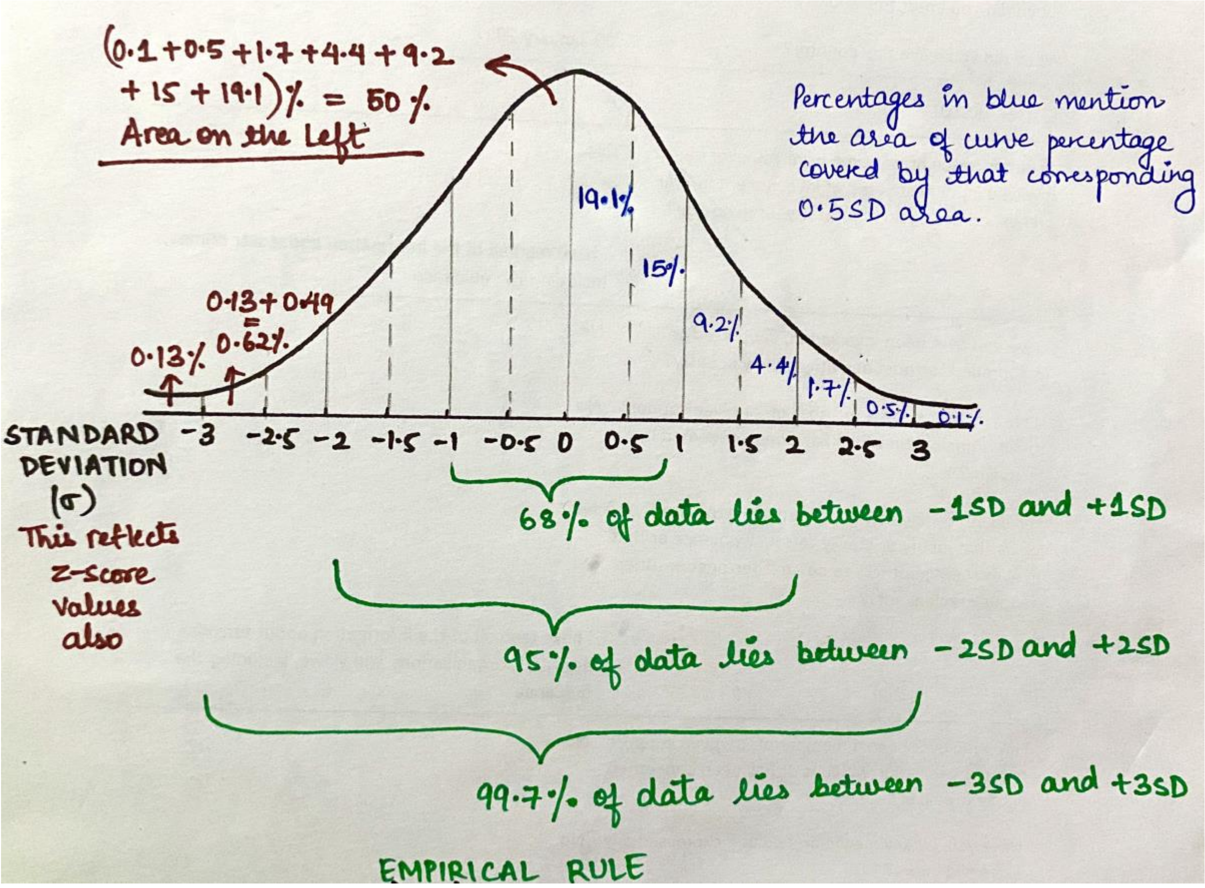
The Z-Score tells us how many standard deviations away from the mean your score is.
For example —

- Z-score of 1.5 then it implies it's 1.5 standard deviations *above* the mean.
- Z-score of -0.8 indicates our value is 0.8 standard deviations *below* the mean.

As explained above, the z-score tells us where the score lies on a normal distribution curve. A Z-score of **zero** tells us the value is **exactly the mean/ average** while a Z-score of +3 tells you that the value is much higher than average (probably an outlier)

Z-score is converting our distribution to a Standard Normal Distribution with a mean of 0 and a Standard deviation of 1.

Interpretation of Z-Score


(0·1+0·5+1·7+4·4+9·2 + 15 + 19·1)% = 50%
Area on the Left
Percentages in blue mention the area of curve percentage covered by that corresponding 0·5SD area.
19·1%
15%
9·2%
4·4%
1·7%
0·5%
0·1%
0·13+0·49 = 0·62%
0·13%
STANDARD DEVIATION (σ)
This reflects z-score values also
-3 -2·5 -2 -1·5 -1 -0·5 0 0·5 1 1·5 2 2·5 3
68% of data lies between -1SD and +1SD
95% of data lies between -2SD and +2SD
99·7% of data lies between -3SD and +3SD
EMPIRICAL RULE

According to the Empirical rule, discussed in detail in the article on Normal distributions linked above and stated at the end of this post too, it's stated that:

- 68% of the data lies between +1SD and -1SD
- 99.5% of the data lies between +2SD and -2SD
- 99.7% of the data lies between +3SD and -3SD

One disadvantage of normalization over standardization is that it loses some information in the data, especially about outliers.

5. You might have observed that sometimes the value of VIF is infinite. Why does this happen?

When we change an independent variable and expect a change in a dependent variable, we see that another independent variable has also changed. These two independent variables are now co-dependent, or collinear of each other. **Adding more features that are collinear to each other and we get multicollinearity.**

Multicollinearity occurs when independent variables in a regression model are correlated. There are two main types of multicollinearity.
1. Structural (independent variable x is squared), which is simply a by-product and since more often than not that you will create it using an existing independent variable, you will be able to track it. Suppose, in a data set we decide to use log to either scale all features or normalize them. That would be an example of structural multicollinearity.

2. Data Multicollinearity.

Multicollinearity can reduce our overall coefficient as well as our p-value (known as the significance value) and cause unpredictable variance. This will lead to overfitting where the model may do great on known training set but will fail at unknown testing set. As this leads to higher standard error with lower statistical significance value, multicollinearity makes it difficult to ascertain how important a feature is to the target variable. And with a lower significance value, we will fail to reject the null, which leads to type II error for our hypothesis testing.

The **variance inflation factor (VIF)** identifies correlation between independent variables and the strength of that correlation.
VIF is an index that provides a measure of how much the variance of an estimated regression coefficient increases due to collinearity. In order to determine VIF, we fit a regression model between the independent variables.
Using Variance Inflation Factor- VIF- we can determine if two independent variables are collinear with each other.
When measuring, if the two features have a VIF of 1, then they are not collinear to each other (i.e. there are no correlation between these two features). However, as the numbers increases, the higher they are correlated with each other.

For example, we would fit the following models to estimate the coefficient of determination R1 and use this value to estimate the VIF:

$$X_1=C+a_2X_2+a_3X_3+\cdots$$

$$\mathrm{VIF}_1=1/(1-R_1^2)$$

Next, we fit the model between X2 and the other independent variables to estimate the coefficient of determination R2:

$$X_2=C+a_1X_1+a_3X_3+\cdots$$

$$\mathrm{VIF}_2=1/(1-R_2^2)$$

If all the independent variables are orthogonal to each other, then VIF = 1.0.

If there is perfect correlation, then VIF = infinity.
A large value of VIF indicates that there is a correlation between the variables.

A general rule of thumb is that if VIF > 10 then there is multicollinearity. Note that this is a rough rule of thumb, in some cases we might choose to live with high VIF values if it does not affect our model results such as when we are fitting a quadratic or cubic model or depending on the sample size a large value of VIF may not necessarily indicate a poor model.

VIF	Conclusion
1	No multicollinearity
4 - 5	Moderate
10 or greater	Severe

If **VIF is large and multicollinearity affects** your analysis results, then you need to take some corrective actions before you can use multiple regression.

Here are the various options:
1. Review your independent variables and eliminate terms that are duplicates or not adding value to explain the variation in the model.
For example, if your inputs are measuring the weight in kgs and lbs then just keep one of these variables in the model and drop the other one.
Dropping the term with a large value of VIF will hopefully, fix the VIF for the remaining terms and now all the VIF factors are within the threshold limits. If dropping one term is not enough, then you may need to drop more terms as required.

2. Use principal component analysis and determine the optimal set of principal components that best describe your independent variables. Using this approach will get rid of your multicollinearity problem but it may be hard for you to interpret the meaning of these “new” independent variables.

3. Increase the sample size
By adding more data points to our model, the confidence intervals for the model coefficients are narrower to overcome the problems associated with multicollinearity.

4. Transform the data to a different space like using a log transformation so that the independent variables are no longer correlated as strongly with each other.

Finally, you can use a different type of model called ridge regression that better handles multicollinearity.
In conclusion, when you are building a multiple regression model, always check your VIF values for your independent variables and determine if you need to take any corrective action before building the model.

6. What is a Q-Q plot? Explain the use and importance of a Q-Q plot in linear regression.

Quantile-Quantile (Q-Q) plot, is a graphical tool to help us assess if a set of data plausibly came from some theoretical distribution such as a Normal, exponential or Uniform distribution. Also, it helps to determine if two data sets come from populations with a common distribution.

This helps in a scenario of linear regression when we have training and test data set received separately and then we can confirm using Q-Q plot that both the data sets are from populations with same distributions.

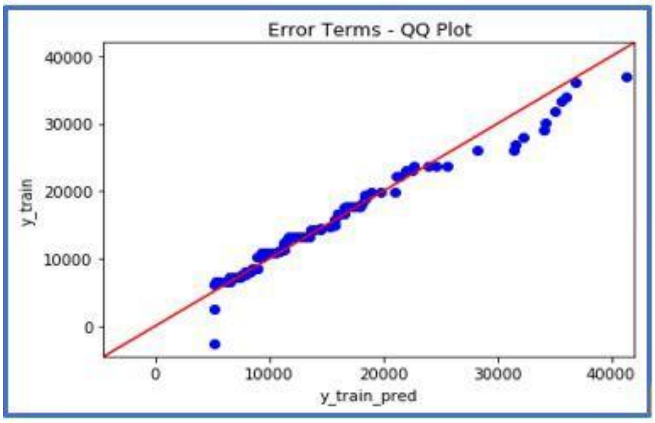
Advantages:
a) It can be used with sample sizes also
b) Many distributional aspects like shifts in location, shifts in scale, changes in symmetry and the presence of outliers can all be detected from this plot.

It is used to check following scenarios:
If two data sets:

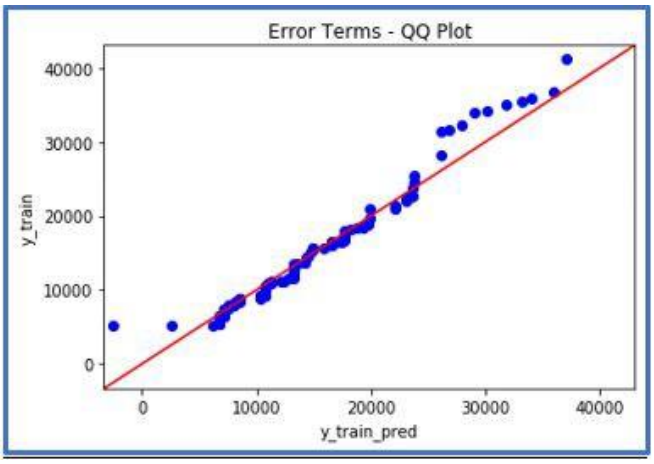
- come from populations with a common distribution
- have common location and scale
- have similar distributional shapes
- have similar tail behaviour

Interpretation:
A q-q plot is a plot of the quantiles of the first data set against the quantiles of the second data set.
Below are the possible interpretations for two data sets:

1. Similar distribution: If all point of quantiles lies on or close to straight line at an angle of 45 degree from x -axis
2. Y-values < X-values: If y-quantiles are lower than the x-quantiles.



3. X-values < Y-values: If x-quantiles are lower than the y-quantiles.



4. Different distribution: If all point of quantiles lies away from the straight line at an angle of 45 degree from x -axis

Python:
statsmodels.api provide qqplot and qqplot_2samples to plot Q-Q graph for single and two different data sets respectively.