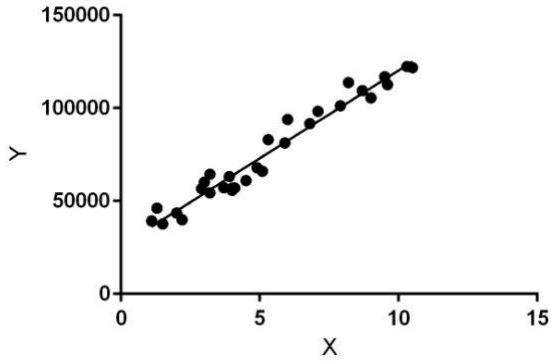
1.Explain the linear regression algorithm in detail

linear regression:

Linear Regression is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.



Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.
In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

Simple linear regression is a type of regression analysis where the number of independent variables is one and there is a linear relationship between the independent(x) and dependent(y) variable. The red line in the above graph is referred to as the best fit straight line. Based on the given data points, we try to plot a line that models the points the best. The line can be modelled based on the linear equation shown below.

For simple linear regression:

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

1.When one variable might not be enough:

A lot of variance isn't explained by just one feature

Inaccurate predictions

1.Formulation of MLR

$$Y=\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_p X_p$$

New considerations to be made when moving from SLR to MLR:

Overfitting:

Multicollinearity:

Multicollinearity affects

Interpretation:

Does “change in Y, when all others are held constant” apply?

Inference:

Coefficients swing wildly, signs can invert

p-values are, therefore, not reliable

1.Looking at **pairwise correlations**

Looking at the correlation between different pairs of independent variables

Checking the **Variance Inflation Factor** (VIF)

Sometimes pairwise correlations aren't enough

Instead of just one variable, the independent variable might depend upon a combination of other variables

VIF calculates how well one independent variable is explained by all the other independent variables combined

The VIF is given

$$VIF_i = \frac{1}{1 - R_i^2}$$

where *'i'* refers to the i-th variable which is being represented as a linear combination of rest of the independent variables. You'll see VIF in action during the Python demonstration on multiple linear regression.

The common heuristic we follow for the VIF values is:

> 10: Definitely high VIF value and the variable should be eliminated.

> 5: Can be okay, but it is worth inspecting.

< 5: Good VIF value. No need to eliminate this variable.

Some methods that can be used to deal with multicollinearity are:

Dropping variables

Drop the variable which is highly correlated with others

Pick the business interpretable variable

Create new variable using the interactions of the older variables

Add interaction features, i.e. features derived using some of the original features

Variable transformations

Principal Component Analysis (covered in a later module)

SCALING

It is important to note that **scaling just affects the coefficients** and none of the other parameters like t-statistic, F-statistic, p-values, R-squared, etc.

Standardisation: $x = \frac{x - mean(x)}{sd(x)}$

MinMax Scaling: $x = \frac{x - min(x)}{max(x) - min(x)}$

Adjusted $R^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$

$AIC = n \times \log(\frac{RSS}{n}) + 2p$

Feature selection:

Build the model with all the features

Drop the features that are least helpful in prediction (high p-value)

Drop the features that are redundant (using correlations and VIF)

Rebuild model and repeat

2.What are the assumptions of linear regression regarding residuals?

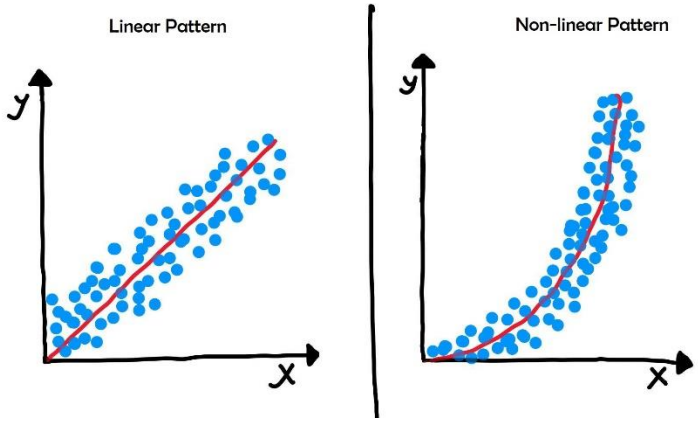
Assumptions of simple Linear Regression:

- Linear relationship between X and y
- Error terms are normally distributed(not x,y)
- Error terms are independent of each other
- Error terms have constant variance(homoscedasticity)

  These assumptions allow us to make inferences, no assumptions on the distribution of x and y

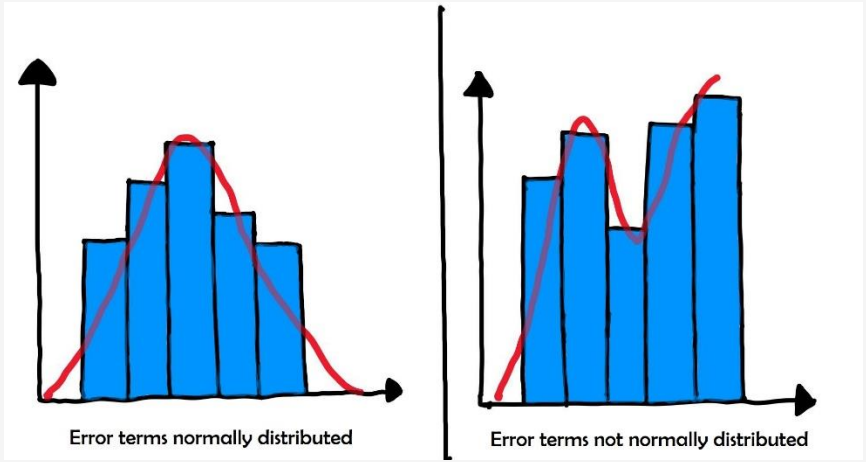
There is a *linear relationship* between X and Y:

X and Y should display some sort of a linear relationship; otherwise, there is no use of fitting a linear model between them.



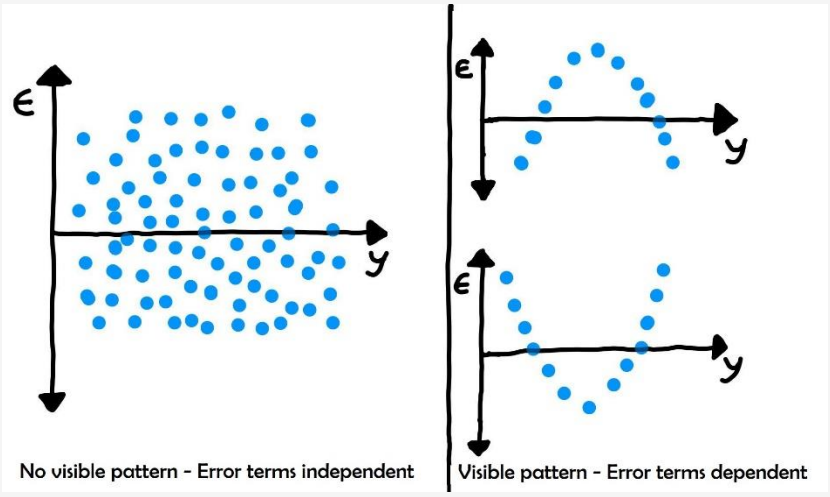
Error terms are *normally distributed* with mean zero(not X, Y):

- There is no problem if the error terms are not normally distributed if you just wish to fit a line and not make any further interpretations.
- But if you are willing to make some inferences on the model that you have built (you will see this in the coming segments), you need to have a notion of the distribution of the error terms. One particular repercussion of the error terms not being normally distributed is that the p-values obtained during the hypothesis test to determine the significance of the coefficients become unreliable. (You'll see this in a later segment.)
- The assumption of normality is made, as it has been observed that the error terms generally follow a **normal distribution with mean equal to zero** in most cases.



Error terms are *independent* of each other:

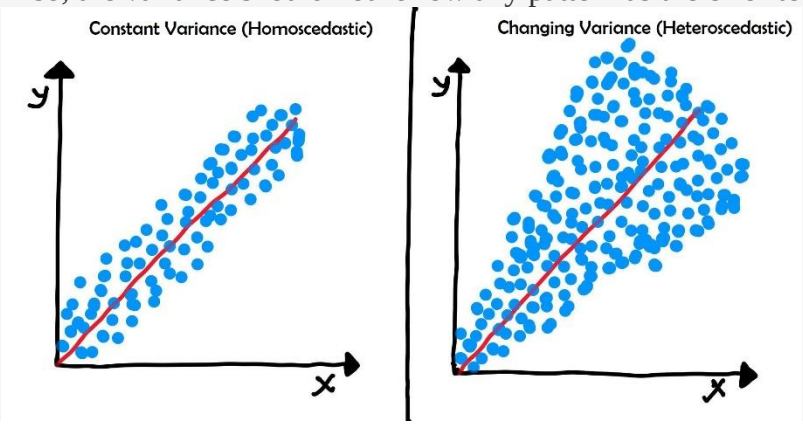
- The error terms should not be dependent on one another (like in a time-series data wherein the next value is dependent on the previous one).



Error terms have *constant variance* (homoscedasticity):

- The variance should not increase (or decrease) as the error values change.

- Also, the variance should not follow any pattern as the error terms change

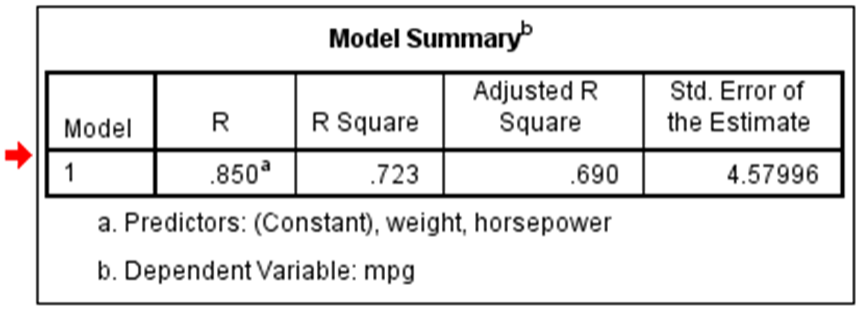


The new aspects to consider when moving from simple to multiple linear regression are:

1. **Overfitting**
    - As you keep adding the variables, the model may become far too complex
    - It may end up memorising the training data and will fail to generalise
    - A model is generally said to overfit when the training accuracy is high while the test accuracy is very low
2. **Multicollinearity**
    - Associations between predictor variables, which you will study later
3. **Feature selection**
    - Selecting the optimal set from a pool of given features, many of which might be redundant becomes an important task

3.What is the coefficient of correlation and the coefficient of determination?

1. Coefficient of correlation is “R” value which is given in the summary table in the Regression output. R square is also called coefficient of determination. Multiply R times R to get the R square value. In other words Coefficient of Determination is the square of Coefficient of Correlation.
2. R square or coeff. of determination shows percentage variation in y which is explained by all the x variables together. Higher the better. It is always between 0 and 1. It can never be negative – since it is a squared value.
3. It is easy to explain the R square in terms of regression. It is not so easy to explain the R in terms of regression.



Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.850[a]	.723	.690	4.57996

a. Predictors: (Constant), weight, horsepower

b. Dependent Variable: mpg

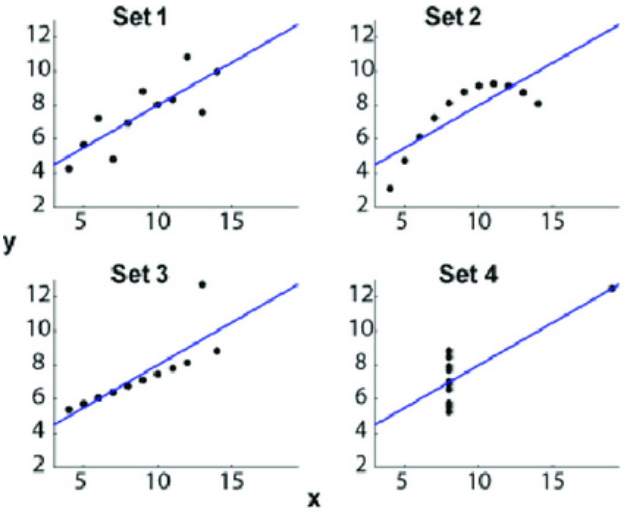
4. Coefficient of Correlation is the R value i.e. .850 (or 85%). Coefficient of Determination is the R square value i.e. .723 (or 72.3%). R square is simply square of R i.e. R times R.
5. Coefficient of Correlation: is the degree of relationship between two variables say x and y. It can go between -1 and 1. 1 indicates that the two variables are moving in unison. They rise and fall together and have perfect correlation. -1 means that the two variables are in perfect opposites. One goes up and other goes down, in perfect negative way. Any two variables in this universe can be argued to have a correlation value. If they are not correlated then the correlation value can still be computed which would be 0. The correlation value always lies between -1 and 1 (going thru 0 – which means no correlation at all – perfectly not related). Correlation can be rightfully explained for simple linear regression – because you only have one x and one y variable. For multiple linear regression R is computed, but then it is difficult to explain because we have multiple variables involved here. That’s why R square is a better term. You can explain R square for both simple linear regressions and also for multiple linear regressions.

4.Explain the Anscombe’s quartet in detail.

Statistics have long been used to describe data in general terms. For example, things like variance and standard deviation allow us to understand how much variation there was in some data without having to look at every data point individually. They give us a rough idea as to how consistent data is. However, knowing variance alone does not give you the full picture as to what the data truly is in its native form.

Statistics are great for describing general trends and aspects of data, but statistics alone can't fully depict any data set. Francis Anscombe realized this in 1973 and created several data sets, all with several identical statistical properties, to illustrate it. These data sets, collectively known as “Anscombe's Quartet,” are shown below.

All four sets are identical when examined using simple summary statistics, but vary considerably when graphed



Mean of Y	**7.50 in all 4 XY plots**
Sample variance of Y	**4.122 or 4.127 in all 4 XY plots**
Correlation (r)	**0.816 in all 4 XY plots**
Linear regression	**y = 3.00 + (0.500 x) in all 4 XY plots**

Data sets for the 4 XY plots

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

- The first scatter plot (top left) appears to be a simple linear relationship, corresponding to two variables correlated where y could be modelled as gaussian with mean linearly dependent on x.
- The second graph (top right) is not distributed normally; while a relationship between the two variables is obvious, it is not linear, and the Pearson correlation coefficient is not relevant. A more general regression and the corresponding coefficient of determination would be more appropriate.
- In the third graph (bottom left), the distribution is linear, but should have a different regression line (a robust regression would have been called for). The calculated regression is offset by the one outlier which exerts enough influence to lower the correlation coefficient from 1 to 0.816.
- Finally, the fourth graph (bottom right) shows an example when one high-leverage point is enough to produce a high correlation coefficient, even though the other data points do not indicate any relationship between the variables.

All four of these data sets have the same variance in x, variance in y, mean of x, mean of y, and linear regression. But, as you can clearly tell, they are all quite different from one another. So, what does this mean to you as a statistician?

Well, to start, Anscombe's Quartet is a great demonstration of the importance of graphing data to analyze it. Given simply variance values, means, and even linear regressions can not accurately portray data in its native form. Anscombe's Quartet shows that multiple data sets with many similar statistical properties can still be vastly different from one another when graphed.

Additionally, Anscombe's Quartet warns of the dangers of outliers in data sets. Think about it: if the bottom two graphs didn't have that one point that strayed so far from all the other points, their statistical properties would no longer be identical to the two top graphs. In fact, their statistical properties would more accurately resemble the lines that the graphs seem to depict.

how to analyze your data. For example, while all four data sets have the same linear regression, it is obvious that the top right graph really shouldn't be analyzed with a linear regression at all because it's a curvature. Conversely, the top left graph probably *should* be analyzed with a linear regression because it's a scatter plot that moves in a roughly linear manner. These observations demonstrate the value in graphing your data before analyzing it.

Anscombe's Quartet reminds us that graphing data prior to analysis is good practice, outliers should be removed when analyzing data, and statistics about a data set do not fully depict the data set in its entirety.

5.What is Pearson's R?

Correlatin(Pearsons) is also called " r" or "pearsons's R

Correlation:

Correlation is a bi-variate analysis that measures the strength of association between two variables and the direction of the relationship. In terms of the strength of relationship, the value of the correlation coefficient varies between +1 and -1. A value of ± 1 indicates a perfect degree of association between the two variables. As the correlation coefficient value goes towards 0, the relationship between the two variables will be weaker. The direction of the relationship is indicated by the sign of the coefficient; a + sign indicates a positive relationship and a - sign indicates a negative relationship.

The Pearson's correlation coefficient varies between -1 and +1 where:

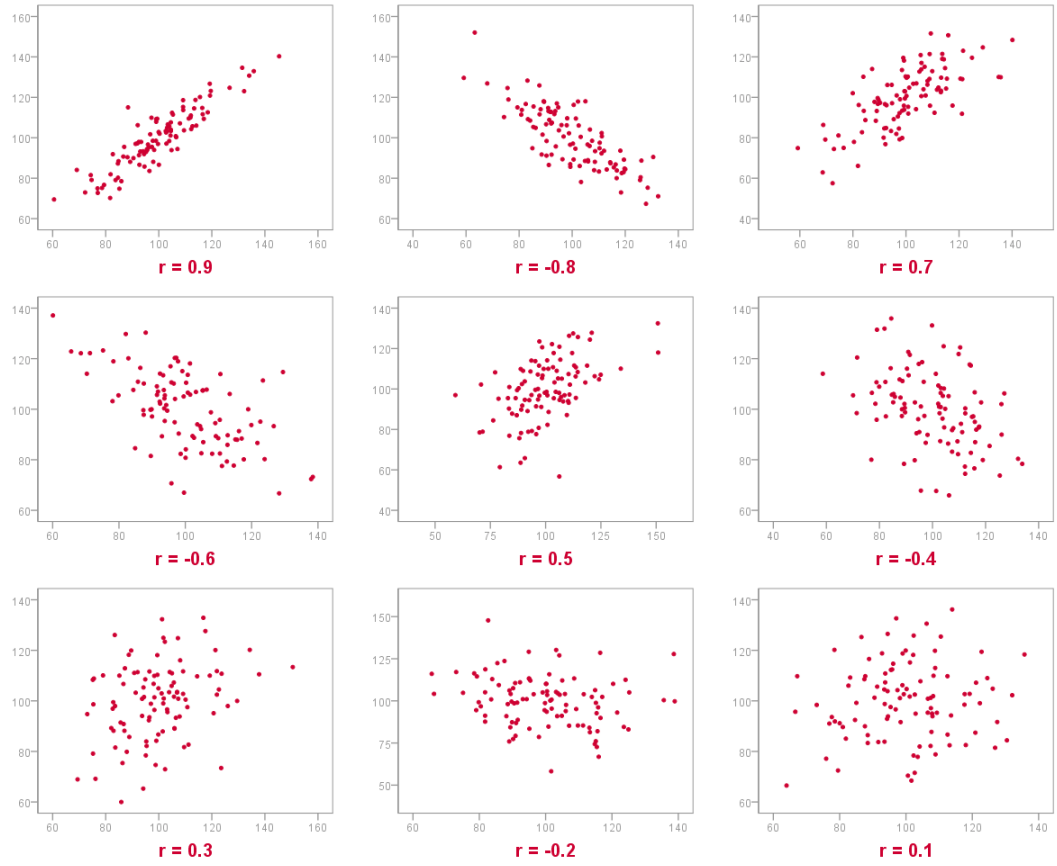
r = 1 means the data is perfectly linear with a positive slope (i.e., both variables tend to change in the same direction)
r = -1 means the data is perfectly linear with a negative slope (i.e., both variables tend to change in different directions)
r = 0 means there is no linear association
r > 0 < 5 means there is a weak association
r > 5 < 8 means there is a moderate association
r > 8 means there is a strong association

Questions a Pearson correlation answers

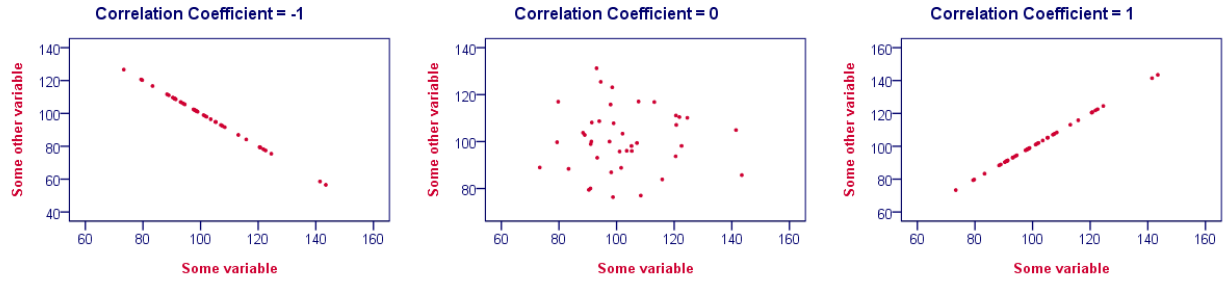
- Is there a statistically significant relationship between age and height?
- Is there a relationship between temperature and ice cream sales?
- Is there a relationship among job satisfaction, productivity, and income?
- Which two variable have the strongest co-relation between age, height, weight, size of family and family income?

Correlation Coefficients and Scatterplots:

A correlation coefficient indicates the extent to which dots in a scatterplot lie on a straight line. This implies that we can usually estimate correlations pretty accurately from nothing more than scatterplots. The figure below nicely illustrates this point.



Some basic points regarding correlation coefficients are nicely illustrated by the previous figure. The least you should know is that



There are several types of correlation coefficient formulas.

One of the most commonly used formulas in stats is Pearson's correlation coefficient formula. If you're taking a basic stats class, this is the one you'll probably use:

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\, n\Sigma x^2 - (\Sigma x)^2 \,][\, n\Sigma y^2 - (\Sigma y)^2 \,]}}$$

Two other formulas are commonly used: the sample correlation coefficient and the population correlation coefficient.

Sample correlation coefficient

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

S_x and s_y are the sample standard deviation, and s_{xy} is the sample covariance

Population correlation coefficient

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

The population correlation coefficient uses σ_x and σ_y as the population standard deviations, and σ_{xy} as the population covariance.

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

Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing,, it is also known as data normalization and is generally performed during the data pre-processing step.

Since the range of values of raw data varies widely, in some machine learning machine algorithms, objective functions will not work properly without normalization. For example, many classifiers 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. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance.

Another reason why feature scaling is applied is that gradient descent converges much faster with feature scaling than without it.

Need of feature scaling:

1. Ease of interpretation
2. Faster convergence for gradient descent methods

Scaling methods:

1. Standardization
2. Minmax scaling

Difference between normalized scaling and standardized scaling

About standardization

The result of **standardization** (or **Z-score normalization**) is that the features will be rescaled so that they'll have the properties of a standard normal distribution with

$\mu=0$ and $\sigma=1$
where $\mu\mu$ is the mean (average) and $\sigma\sigma$ is the standard deviation from the mean; standard scores (also called ***z*** scores) of the samples are calculated as follows:
$z=x-\mu\sigma z$

Standardizing the features so that they are centered around 0 with a standard deviation of 1 is not only important if we are comparing measurements that have different units, but it is also a general requirement for many machine learning algorithms. Intuitively, we can think of gradient descent as a prominent example (an optimization algorithm often used in logistic regression, SVMs, perceptrons, neural networks etc.); with features being on different scales, certain weights may update faster than others since the feature values xjxj play a role in the weight updates
$\Delta wj = -\eta \partial J \partial wj = \eta \sum i(t(i) - o(i))x(i)j$

so that

wj:=wj+Δwj,wj where ηη is the learning rate, tt the target class label, and oo the actual output. Other intuitive examples include K-Nearest Neighbor algorithms and clustering algorithms that use, for example, Euclidean distance measures – in fact, tree-based classifier are probably the only classifiers where feature scaling doesn't make a difference.
In fact, the only family of algorithms that I could think of being scale-invariant are tree-based methods. Let's take the general CART decision tree algorithm. Without going into much depth regarding information gain and impurity measures, we can think of the decision as "is feature x_i >= some_val?" Intuitively, we can see that it really doesn't matter on which scale this feature is (centimeters, Fahrenheit, a standardized scale – it really doesn't matter).

Some examples of algorithms where feature scaling matters are:

- k-nearest neighbors with an Euclidean distance measure if want all features to contribute equally
- k-means (see k-nearest neighbors)
- logistic regression, SVMs, perceptrons, neural networks etc. if you are using gradient descent/ascent-based optimization, otherwise some weights will update much faster than others
- linear discriminant analysis, principal component analysis, kernel principal component analysis since you want to find directions of maximizing the variance (under the constraints that those directions/eigenvectors/principal components are orthogonal); you want to have features on the same scale since you'd emphasize variables on "larger measurement scales" more. There are many more cases than I can possibly list here … I always recommend you to think about the algorithm and what it's doing, and then it typically becomes obvious whether we want to scale your features or not.

In addition, we'd also want to think about whether we want to "standardize" or "normalize" (here: scaling to [0, 1] range) our data. Some algorithms assume that our data is centered at 0. For example, if we initialize the weights of a small multi-layer perceptron with tanh activation units to 0 or small random values centered around zero, we want to update the model weights "equally." As a rule of thumb I'd say: When in doubt, just standardize the data, it shouldn't hurt.

About Min-Max scaling

An alternative approach to Z-score normalization (or standardization) is the so-called **Min-Max scaling** (often also simply called “normalization” - a common cause for ambiguities).
In this approach, the data is scaled to a fixed range - usually 0 to 1.
The cost of having this bounded range - in contrast to standardization - is that we will end up with smaller standard deviations, which can suppress the effect of outliers.

A Min-Max scaling is typically done via the following equation:

Xnorm=X−Xmin/Xmax−Xmin

Z-score standardization or Min-Max scaling?

“Standardization or Min-Max scaling?” - There is no obvious answer to this question: it really depends on the application.

For example, in clustering analyses, standardization may be especially crucial in order to compare similarities between features based on certain distance measures. Another prominent example is the Principal Component Analysis, where we usually prefer standardization over Min-Max scaling, since we are interested in the components that maximize the variance (depending on the question and if the PCA computes the components via the correlation matrix instead of the covariance

However, this doesn’t mean that Min-Max scaling is not useful at all! A popular application is image processing, where pixel intensities have to be normalized to fit within a certain range (i.e., 0 to 255 for the RGB color range). Also, typical neural network algorithm require data that on a 0-1 scale.

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

VIF:

The ***variance inflation factor*** *(VIF)* quantifies the extent of correlation between one predictor and the other predictors in a model. It is used for diagnosing *collinearity/multicollinearity*. Higher values signify that it is difficult to impossible to assess accurately the contribution of predictors to a model.

How the VIF is computed

The *standard error* of an *estimate* in a *linear regression* is determined by four things:

- The overall amount of noise (error). The more noise in the data, the higher the standard error.

- The variance of the associated predictor variable. The greater the variance of a predictor, the smaller the standard error (this is a *scale* effect).
- The sampling mechanism used to obtain the data. For example, the smaller the sample size with a simple random sample, the bigger the standard error.

- The extent to which a predictor is correlated with the other predictors in a model.

  The extent to which a predictor is correlated with the other predictor variables in a linear regression can be quantified as the *R-squared* statistic of the regression where the predictor of interest is predicted by all the other predictor variables ( ). The *variance inflation* for a variable is then computed as:

$$VIF = \frac{1}{1 - R^2}$$

  Some statistical software use *tolerance* instead of VIF, where tolerance is:

$$1 - R^2 = \frac{1}{VIF}.$$

  The VIF can be applied to any type of predictive model (e.g., CART, or deep learning). A generalized version of the VIF, called the *GVIF*, exists for testing sets of predictor variables and generalized linear models.

Reasons why the VIF is infinitive:

- A VIF can be computed for each predictor in a predictive model. A value of 1 means that the predictor is not correlated with other variables. The higher the value, the greater the correlation of the variable with other variables. Values of more than 4 or 5 are sometimes regarded as being moderate to high, with values of 10 or more being regarded as very high.

- If one variable has a high VIF it means that other variables must also have high VIFs. In the simplest case, two variables will be highly correlated, and each will have the same high VIF.

- Where a VIF is high, it makes it difficult to disentangle the relative importance of predictors in a model, particularly if the standard errors are regarded as being large. This is particularly problematic in two scenarios, where:

- Because more data, so the increase in the standard errors.

- where the predictors are highly correlated

- increased correlations

8.What is the Gauss-Markov theorem:

Gauss Markov theorem

The Gauss Markov theorem says that, under certain conditions, the ordinary least squares (OLS) estimator of the coefficients of a linear regression model is the best linear unbiased estimator (BLUE), that is, the estimator that has the smallest variance among those that are unbiased and linear in the observed output variables.

Assumptions

The regression model iswhere:

- is an vector of observations of the output variable (is the sample size);
- is an matrix of inputs (is the number of inputs for each observation);
- is a vector of regression coefficients;
- is an vector of errors.

The OLS estimator of is

We assume that:

1. has full-rank (as a consequence, is invertible, and is well-defined);
2. ;
3. , where is the identity matrix and is a positive constant.

OLS is linear and unbiased

First of all, note that is linear . In fact, is the product between the matrix ,and matrix multiplication is a linear operation.

It can easily be proved that is unbiased, both conditional on , and unconditionally, that is,
Proof

What it means to be best

Now that we have shown that the OLS estimator is linear and unbiased, we need to prove that it is also the **best** linear unbiased estimator.

What exactly do we mean by best?

When is a scalar (i.e., there is only one regressor), we consider to be the best among those we are considering (i.e., among all the linear unbiased estimators) if and only if it has the smallest possible variance, that is, if its deviations from the true value tend to be the smallest on average. Thus, is the best linear unbiased estimator (BLUE) if and only iffor any other linear unbiased estimator .

Since we often deal with more than one regressor, we have to extend this definition to a multivariate context. We do this by requiring thatfor any constant vector , any other linear unbiased estimator .

In other words, OLS is BLUE if and only if any linear combination of the regression coefficients is estimated more precisely by OLS than by any other linear unbiased estimator.

Condition (1) is satisfied if and only ifis a positive semi-definite matrix.
Proof

In the next two sections we will derive (the covariance matrix of the OLS estimator), and then we will prove that (2) is positive-semidefinite, so that OLS is BLUE.

The covariance matrix of the OLS estimator

The conditional covariance matrix of the OLS estimator is
Proof

OLS is BLUE

Since we are considering the set of linear estimators, we can write any estimator in this set as where is a matrix.

Furthermore, if we definethen we can write

It is possible to prove that if is unbiased.
Proof

By using this result, we can also prove that $\mathrm{Var}\left[\tilde{\beta}|X\right] = \mathrm{Var}\left[\hat{\beta}|X\right] + \sigma^2 DD^\top$
Proof

As a consequence, $\mathrm{Var}\left[\tilde{\beta}|X\right] - \mathrm{Var}\left[\hat{\beta}|X\right] = \sigma^2 DD^\top$ is positive semi-definite because is positive semi-definite. This is true for any unbiased linear estimator . Therefore, the OLS estimator is BLUE.

9.Explain the gradient descent algorithm in detail.

It is an optimization algorithm used in training a model. In simple words, Gradient Descent finds the parameters that minimize the cost function (error in prediction). Gradient Descent does this by iteratively moves toward a set of parameter values that minimize the function, taking steps in the opposite direction of the gradient.

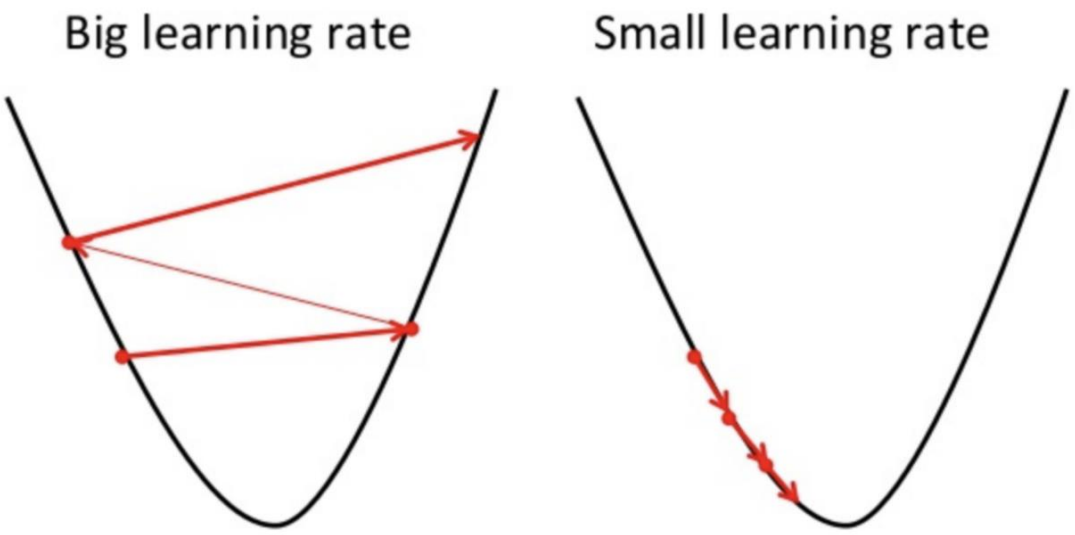
Gradient

A gradient is a vector-valued function that represents the slope of the tangent of the graph of the function, pointing the direction of the greatest rate of increase of the function. It is a derivative that indicates the incline or the slope of the cost function.

What is the Learning rate?

Like we said, the gradient is a vector-valued function, and as a vector, it has both a direction and a magnitude. The Gradient descent algorithm multiplies the gradient by a number (Learning rate or Step size) to determine the next point.

For example: having a gradient with a magnitude of 4.2 and a learning rate of 0.01, then the gradient descent algorithm will pick the next point 0.042 away from the previous point.



The learning rate is a randomly chosen number that tells how far to move the weights. A small learning rate and you will take forever to reach the minima, a large learning rate and you will skip the minima.

Typically, the value of the learning rate is chosen manually, starting with 0.1, 0.01 or 0.001 as the common values, and then adapt it whether the gradient descent is taking too long to calculate (you need to increase the learning rate), or is exploding or being erratic (you need to decrease the learning rate).

Although the learning rate is usually chosen manually, there are several methods to automatically choose a fitting learning rate, such as AdaGram and RMSProp methods, but we will talk about them later.

Steps of gradient descent

Following the mountain example, we will call Agent as the blinded person. In each position, the agent only knows two things: the gradient (for that position, or parameters) and the width of the step to take (learning rate). With that information, the current value of each parameter is updated. With the new parameter values, the gradient is re-calculated and the process is repeated until reach *convergence* or local minima.

Let's revise how the gradient descent algorithm works at each step:

Repeat until hit convergence:

1. Given the gradient, calculate the change in the parameters with the learning rate.
2. Re-calculate the new gradient with the new value of the parameter.
3. Repeat step 1.

Here is the formula of gradient descent algorithm:

Repeat until convergence {

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

Image 2: Gradient Descent algorithm. (Source: Samuel Trendler)[/caption]Convergence

Convergence is a name given to the situation where the loss function does not improve significantly, and we are stuck in a point near to the minima.

Gradient Descent variants

There are three variants of gradient descent based on the amount of data used to calculate the gradient:

- Batch gradient descent
- Stochastic gradient descent
- Mini-batch gradient descent

Batch Gradient Descent:

Batch Gradient Descent, aka Vanilla gradient descent, calculates the error for each observation in the dataset but performs an update only after all observations have been evaluated.

Batch gradient descent is not often used, because it represents a huge consumption of computational resources, as the entire dataset needs to remain in memory.

Stochastic Gradient Descent:

Stochastic gradient descent (SGD) performs a parameter update for each observation. So instead of looping over each observation, *it just needs one* to perform the parameter update. SGD is usually faster than batch gradient descent, but its frequent updates cause a higher variance in the error rate, that can sometimes jump around instead of decreasing.

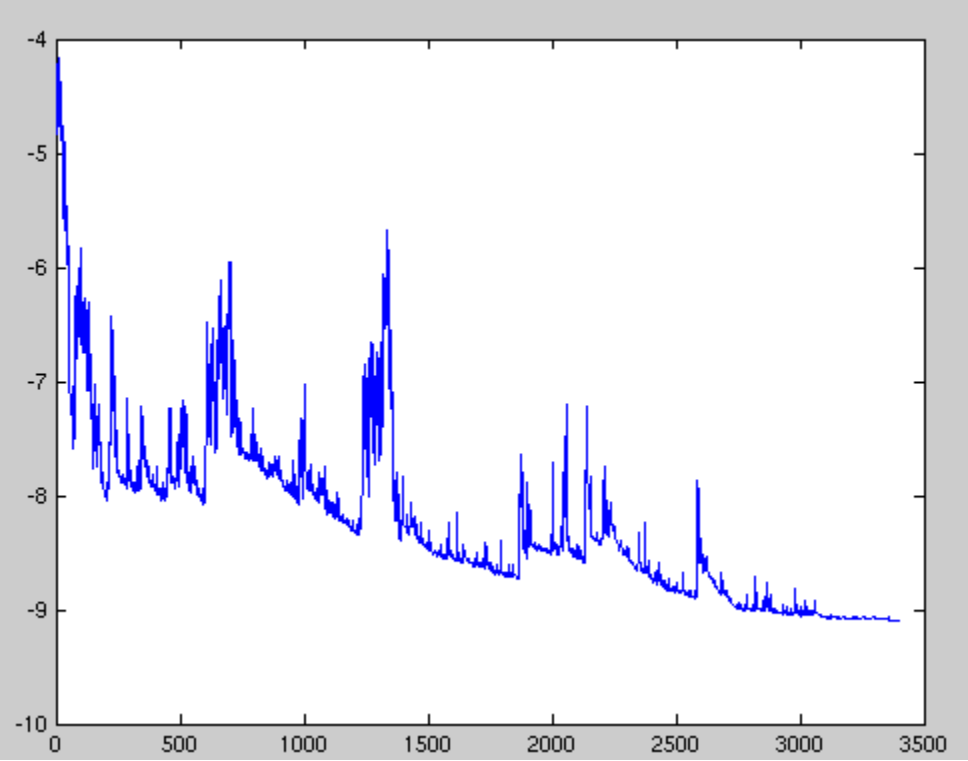


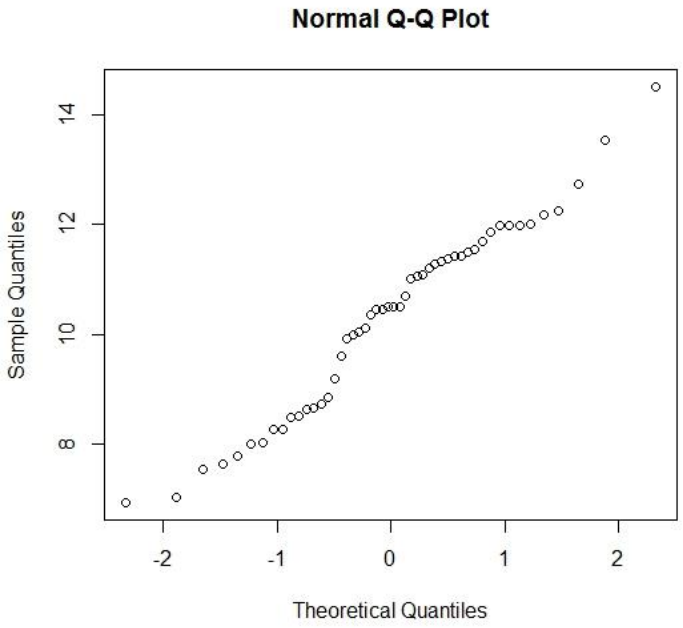
Image 3: SDG fluctuation (Source: Wikipedia)

Mini-Batch Gradient Descent:

It is a combination of both bath gradient descent and stochastic gradient descent. Mini-batch gradient descent performs an update for a batch of observations. It is the algorithm of choice for neural networks, and the batch sizes are usually from 50 to 256.

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

The quantile-quantile or q-q plot is an exploratory graphical device used to check the validity of a distributional assumption for a data set. In general, the basic idea is to compute the theoretically expected value for each data point based on the distribution in question. If the data indeed follow the assumed distribution, then the points on the q-q plot will fall approximately on a straight line.Before delving into the details of q-q plots, we first describe two related graphical methods for assessing distributional assumptions: the histogram and the cumulative distribution function (CDF). As will be seen, q-q plots are more general than these alternatives.



If the spinner is fair, then these numbers should follow a uniform distribution. To investigate whether the spinner is fair, spin the arrow n times, and record the measurements by $\{\mu_1, \mu_2, ..., \mu_n\}$. In this example, we collect n = 100 samples. The histogram provides a useful visualization of these data. In Figure 2, we display three different histograms on a probability scale. The histogram should be flat for a uniform sample, but the visual perception varies depending on whether the histogram has 10, 5, or 3 bins. The last histogram looks flat, but the other two histograms are not obviously flat. It is not clear which histogram we should base our conclusion on.

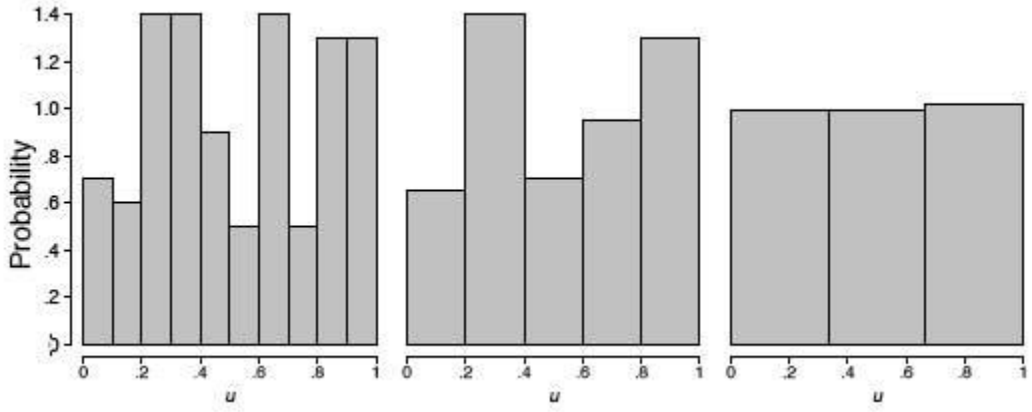


Figure 2. Three histograms of a sample of 100 uniform points.

Alternatively, we might use the cumulative distribution function (CDF), which is denoted by F(μ). The CDF gives the probability that the spinner gives a value less than or equal to μ, that is, the probability that the red arrow lands in the interval [0, μ]. By simple arithmetic, F(μ) = μ, which is the diagonal straight line y = x. The CDF based upon the sample data is called the empirical CDF (ECDF), is denoted by $\hat{F}_n(u)$, and is defined to be the fraction of the data less than or equal to μ; that is,

$$\hat{F}_n(u) = \frac{\#\ u_i \leq u}{n}.$$

In general, the ECDF takes on a ragged staircase appearance.
For the spinner sample analyzed in Figure 2, we computed the ECDF and CDF, which are displayed in Figure 3. In the left frame, the ECDF appears close to the line y = x, shown in the middle frame. In the right frame, we overlay these two curves and verify that they are indeed quite close to each other. Observe that we do not need to specify the number of bins as with the histogram.

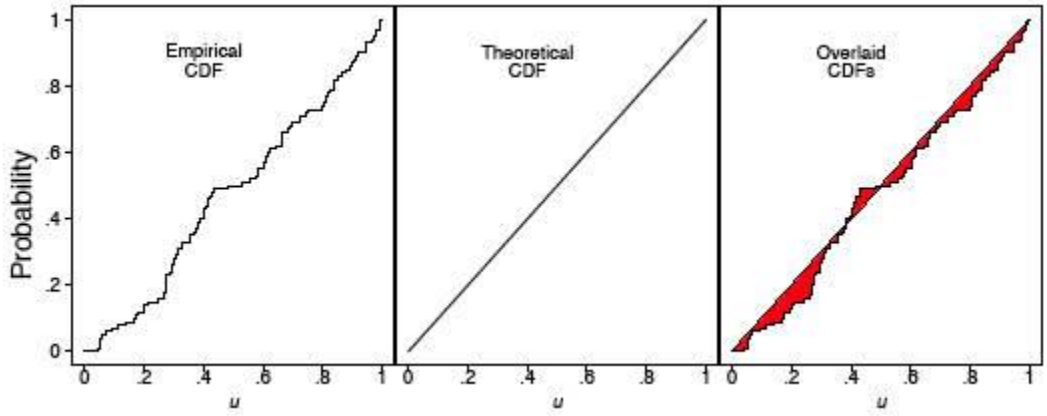


Figure 3. The empirical and theoretical cumulative distribution functions of a sample of 100 uniform points.

Q-Q PLOT FOR UNIFORM DATA

The q-q plot for uniform data is very similar to the empirical CDF graphic, except with the axes reversed. The q-q plot provides a visual comparison of the sample quantiles to the corresponding theoretical quantiles. In general, if the points in a q-q plot depart from a straight line, then the assumed distribution is called into question.

Here we define the qth quantile of a batch of n numbers as a number ξ_q such that a fraction q x n of the sample is less than ξ_q, while a fraction (1 - q) x n of the sample is greater than ξ_q. The best known quantile is the median, $\xi_{0.5}$, which is located in the middle of the sample.

Consider a small sample of 5 numbers from the spinner:

$\mu_1 = 0.41$, $\mu_2 = 0.24$, $\mu_3 = 0.59$, $\mu_4 = 0.03$, and $\mu_5 = 0.67$.

Based upon our description of the spinner, we expect a uniform distribution to model these data. If the sample data were "perfect," then on average there would be an observation in the middle of each of the 5 intervals: 0 to .2, .2 to .4, .4 to .6, and so on. Table 1 shows the 5 data points (sorted in ascending order) and the theoretically expected value of each based on the assumption that the distribution is uniform (the middle of the interval).

Table 1. Computing the Expected Quantile Values.

Data (μ)	**Rank (*i*)**	**Middle of the *i*th Interval**
.03	1	.1
.24	2	.3
.41	3	.5
.59	4	.7
.67	5	.9

The theoretical and empirical CDFs are shown in Figure 4 and the q-q plot is shown in the left frame of Figure 5.

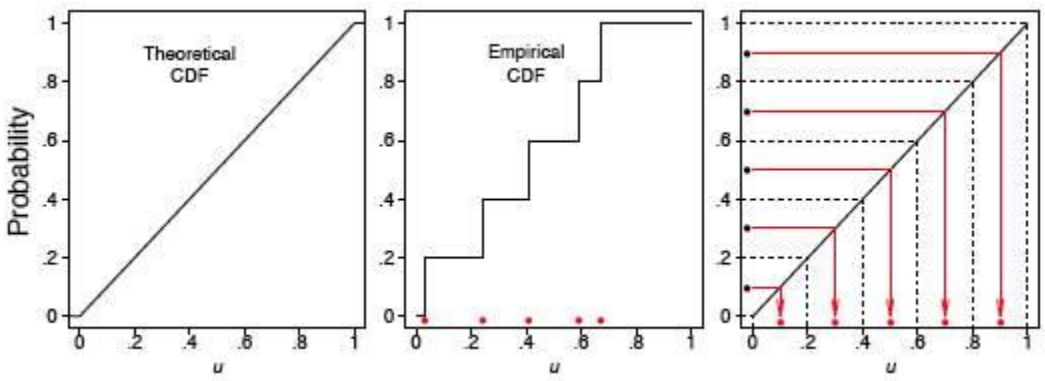


Figure 4. The theoretical and empirical CDFs of a small sample of 5 uniform points, together with the expected values of the 5 points (red dots in the right frame).

In general, we consider the full set of sample quantiles to be the sorted data values

$$\mu_{(1)} < \mu_{(2)} < \mu_{(3)} < \cdots < \mu_{(n-1)} < \mu_{(n)} ,$$

where the parentheses in the subscript indicate the data have been ordered. Roughly speaking, we expect the first ordered value to be in the middle of the interval (0, 1/n), the second to be in the middle of the interval (1/n, 2/n), and the last to be in the middle of the interval ((n - 1)/n, 1). Thus, we take as the theoretical quantile the value

$$\xi_q = q \approx \frac{i - 0.5}{n},$$

where q corresponds to the ith ordered sample value. We subtract the quantity 0.5 so that we are exactly in the middle of the interval ((i - 1)/n, i/n). These ideas are depicted in the right frame of Figure 4 for our small sample of size n = 5.

We are now prepared to define the q-q plot precisely. First, we compute the n expected values of the data, which we pair with the n data points sorted in ascending order. For the uniform density, the q-q plot is composed of the n ordered pairs

$$\left(\frac{i - 0.5}{n}, u_{(i)} \right) , \qquad \text{for } i = 1, 2, \ldots, n .$$

This definition is slightly different from the ECDF, which includes the points ($u_{(i)}$,

i/n). In the left frame of Figure 5, we display the q-q plot of the 5 points in Table 1. In the right two frames of Figure 5, we display the q-q plot of the same batch of numbers used in Figure 2. In the final frame, we add the diagonal line y = x as a point of reference.

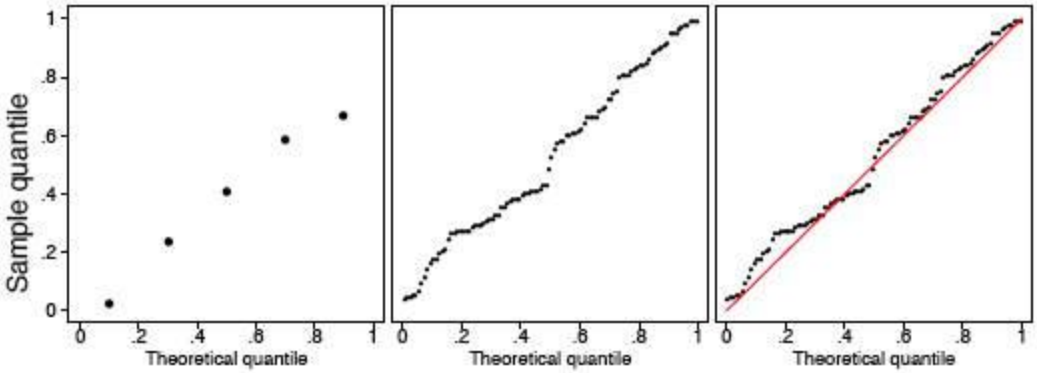


Figure 5. (Left) q-q plot of the 5 uniform points. (Right) q-q plot of a sample of 100 uniform points.

The sample size should be taken into account when judging how close the q-q plot is to the straight line. We show two other uniform samples of size n = 10 and n = 1000 in Figure 6. Observe that the q-q plot when n = 1000 is almost identical to the line y = x, while such is not the case when the sample size is only n = 10.

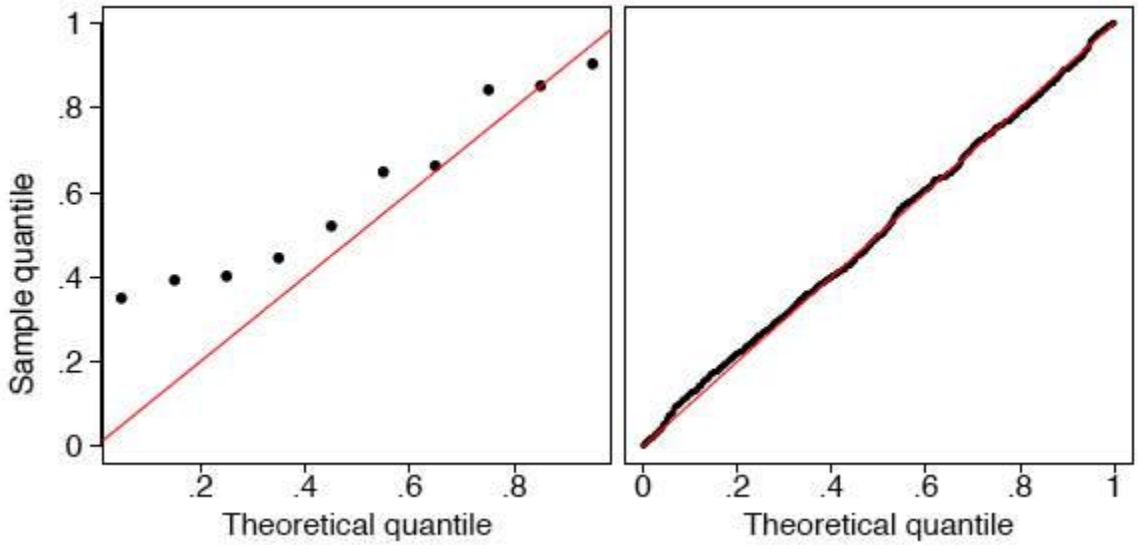


Figure 6. q-q plots of a sample of 10 and 1000 uniform points.

In Figure 7, we show the q-q plots of two random samples that are not uniform. In both examples, the sample quantiles match the theoretical quantiles only at the median and at the extremes. Both samples seem to be symmetric around the median. But the data in the left frame are closer to the median than would be expected if the data were uniform. The data in the right frame are further from the median than would be expected if the data were uniform.

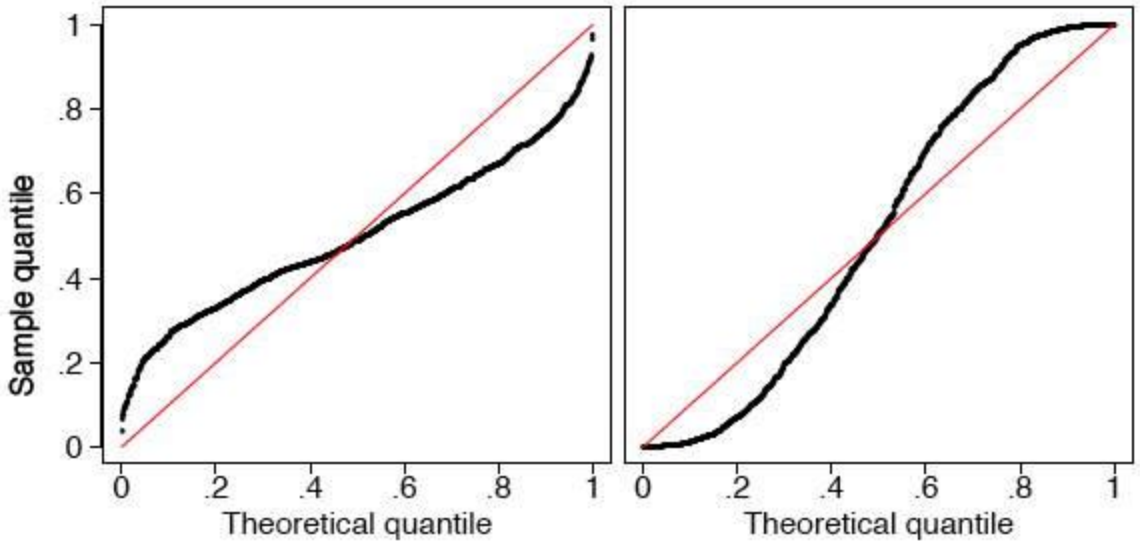


Figure 7. q-q plots of two samples of size 1000 that are not uniform.

In fact, the data were generated in the R language from beta distributions with parameters a = b = 3 on the left and a = b =0.4 on the right. In Figure 8 we display histograms of these two data sets, which serve to clarify the true shapes of the densities. These are clearly non-uniform.

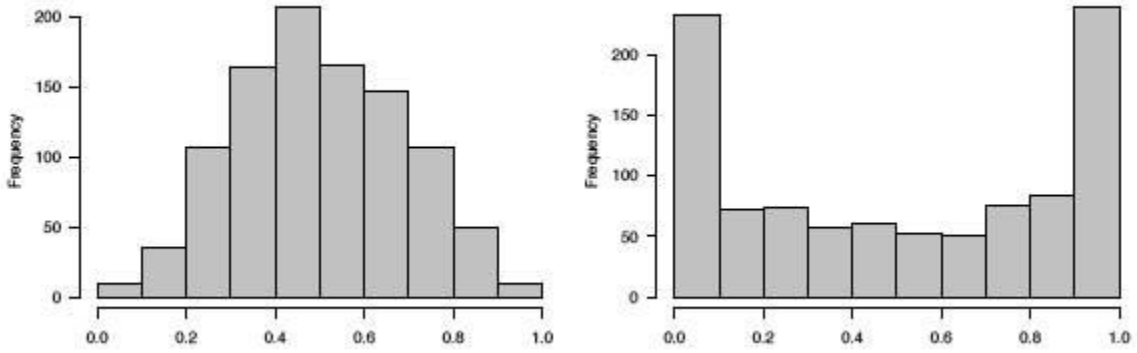


Figure 8. Histograms of the two non-uniform data sets.