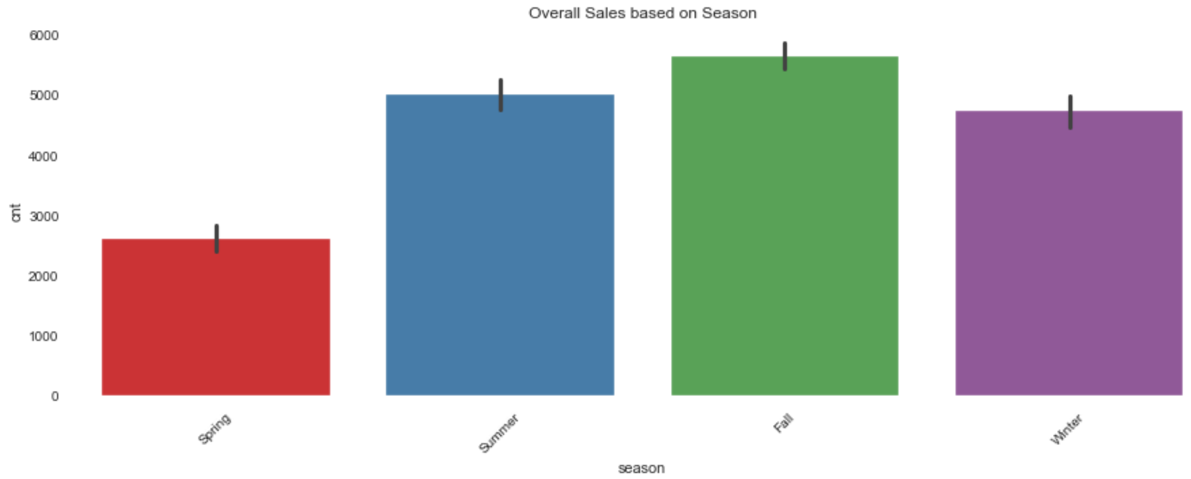
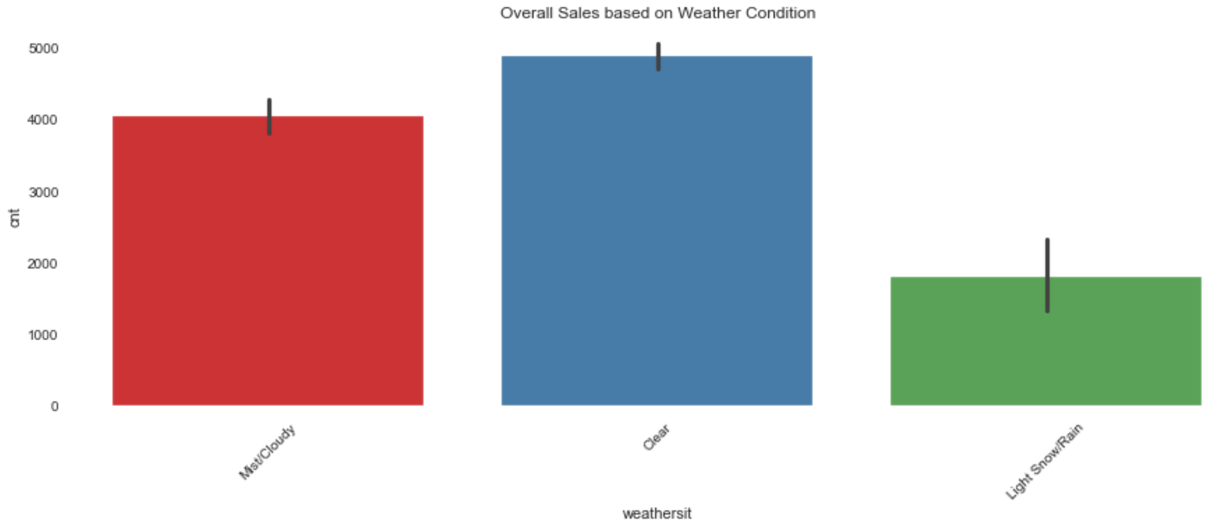
Assignment-based Subjective Questions

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

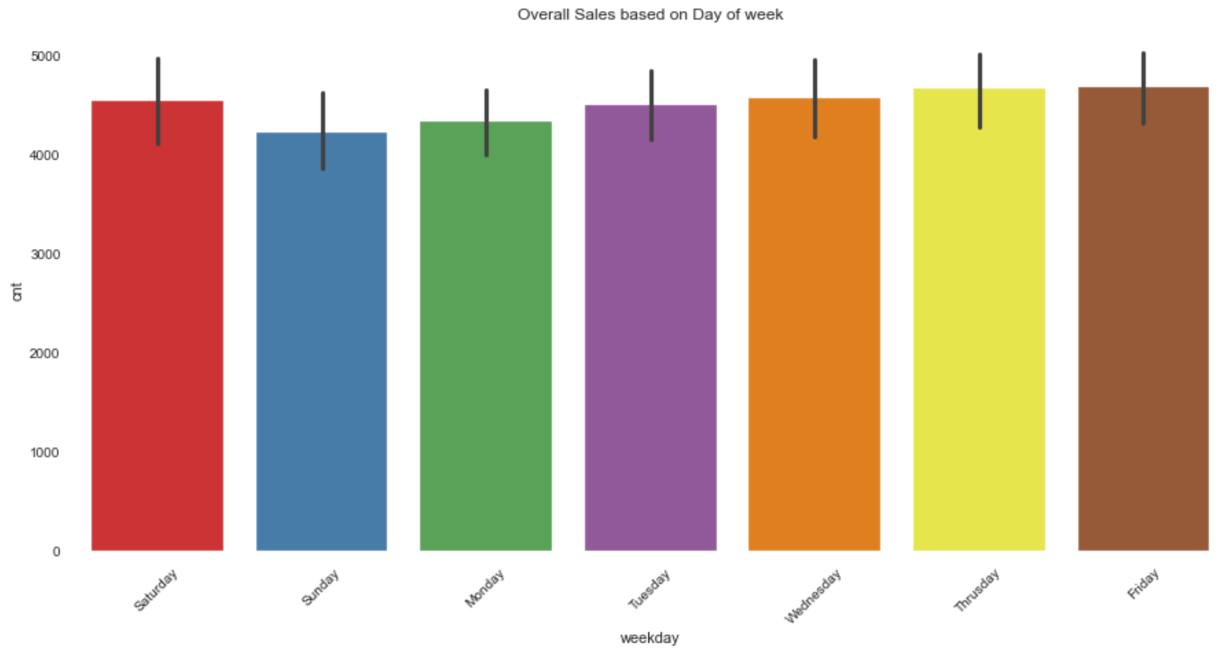
Ans. From the Dataset we saw that we had couple of categorical variables such as "season", " month", "weekday", "weathersit" We could see some close impact with the target variable and these were a deriving factor to find the target variable.



We could see a clear distribution of bike rentals across the season and largely contribution in the "Fall" Season.



We also a large volume of bike rentals during clear sky. And there was drop in rentals during Rains and Snow as expected.



We also almost uniform rental sales distribution over all the days in a week and a little higher during Saturdays.

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

Ans. Drop_first becomes important when lets say you have a variable called “Is_Active” then if you don’t use drop_first =True then it would create 2 Columns Is_Active0 and Is_Active1 which is not required so in that case it would be good to use drop_first =True.

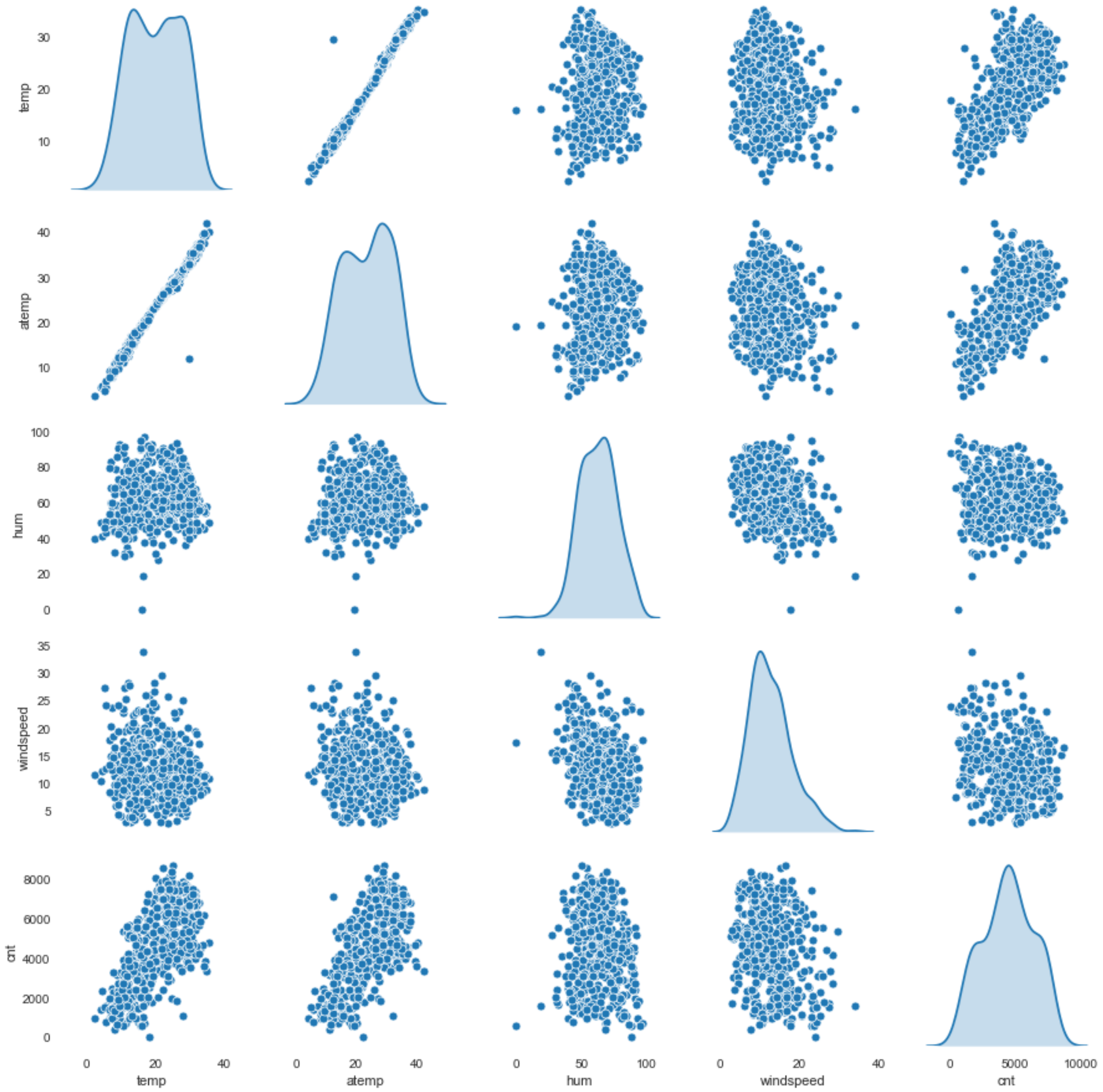
Using drop_first=True is more common in statistics and often referred to as "dummy encoding" while using drop_first=False gives you "one hot-encoding" which is more common in ML. For algorithmic approaches like Random Forests it does not make a difference.

However, using dummy encoding on a binary variable does not mean that a 0 has no relevance.
If gender_male has high importance that does not generally say anything about the importance of gender_male==0 vs gender_male==1. It is **variable** importance and accordingly calculated per variable. If you, for example, use impurity based estimates in Trees it only gives you the average reduction in impurity achieved by splitting on this very variable.
Moreover, if your gender variable is binary, gender_male==1 is equivalent to gender_female==0. Therefore from a high variable importance of gender_male you cannot infer that being female (or not) is not relevant.

In this case gender_male==0 AND gender_female==0 means Transgender is true.
For algorithmic approaches in ML there is no statistical disadvantage using one-hot-encoding. (as pointed out in the comments it might even be advantageous since tree-based models can directly split on all features when none is being dropped)

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

Ans. Looking at the Pair plot we see a high correlation between “temp” and “atemp” wrt. to the target variable “cnt”.

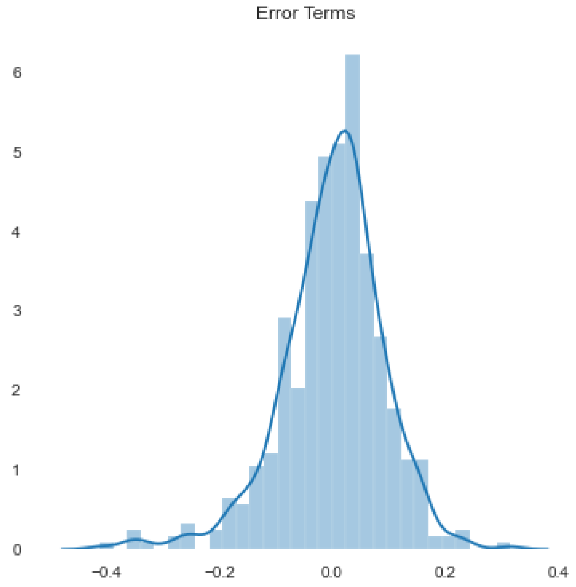


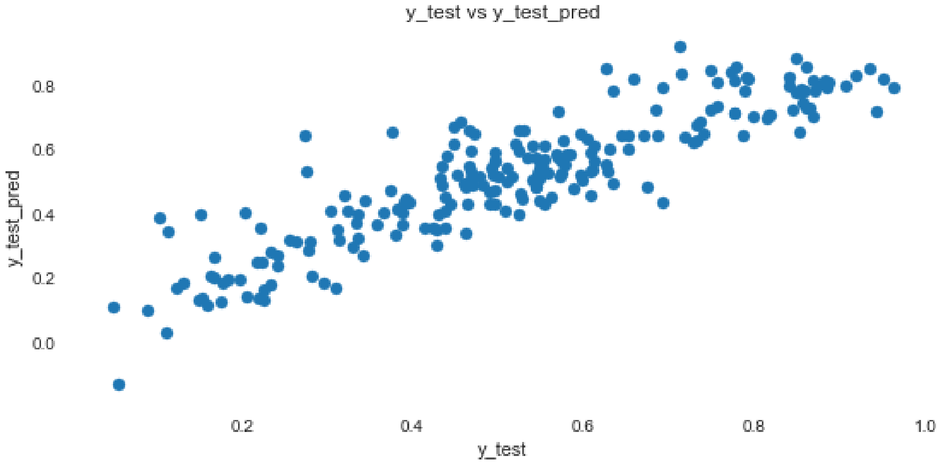
4. How did you validate the assumptions of Linear Regression after building the model on the training set?

Ans. After building the model we predicted the target variable with the help of the model and then with the residual (Error terms) we plotted a graph which clearly indicates a normal distribution.

From the VIF calculation we also saw that there is no multicollinearity existing between the predictor variables, as all the values are within permissible range. We also saw that p-value was 0 for all the variables and more that 80%(R2 value) of the model was predicted by those variables.

We also plotted the actual vs derived target variable “cnt” to identify and show the linear distribution.





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

Ans. With the final model we saw that **“temp”, “yr” and “weathersit_Light Snow/Rain”** were the main driving factors. Where in **“weathersit_Light Snow/Rain”** had a **negative** correlation to the predictive variable (“**cnt**”).

cnt = 0.075009+(yr×0.233139) + (workingday×0.056117) + (temp×0.549892) − (windspeed×0.155203) + (season_Summer×0.088621) + (season_Winter × 0.130655) + (mnth_Sep×0.097365) + (weekday_Saturday×0.067500) − (weathersit_Light Snow/Rain×0.287090) − (weathersit_Mist/Cloudy×0.080022)

General Subjective Questions

1. Explain the linear regression algorithm in detail.

Linear regression is one of the very basic forms of machine learning where we train a model to predict the behavior of your data based on some variables. In the case of linear regression as you can see the name suggests linear that means the two variables which are on the x-axis and y-axis should be linearly correlated.

Example for that can be let’s say you are running a sales promotion and expecting a certain number of count of customers to be increased now what you can do is you can look the previous promotions and plot if over on the chart when you run it and then try to see whether there is an increment into the number of customers whenever you rate the promotions and with the help of the previous historical data you try to figure it out or you try to estimate what will be the count or what will be the estimated count for my current promotion this will give you an idea to do the planning in a much better way about how many numbers of stalls maybe you need or how many increase number of employees you need to serve the customer. Here the idea is to estimate the future value based on the historical data by learning the behavior or patterns from the historical data.

In some cases, the value will be linearly upward that means whenever X is increasing Y is also increasing or vice versa that means they have a correlation or there will be a linear downward relationship.

One example for that could be that the police department is running a campaign to reduce the number of robberies, in this case, the graph will be linearly downward.

Linear regression is used to predict a quantitative response Y from the predictor variable X.

Mathematically, we can write a linear regression equation as:

$$y = a + bx$$

Where a and b given by the formulas:

$$b\,(slope) = \frac{n\sum xy - (\sum x)\,(\sum y)}{n\sum x^2 - (\sum x)^2}$$

$$a\,(intercept) = \frac{n\sum y - b\,(\sum x)}{n}$$

Here, x and y are two variables on the regression line.

b = Slope of the line.

a = y-intercept of the line.

x = Independent variable from dataset

y = Dependent variable from dataset

Use Cases of Linear Regression:

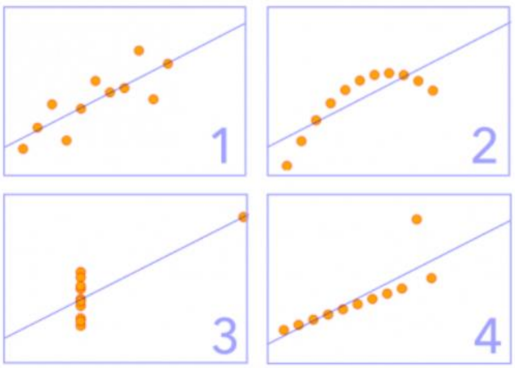
1. Prediction of trends and Sales targets – To predict how industry is performing or how many sales targets industry may achieve in the future.
2. Price Prediction – Using regression to predict the change in price of stock or product.
3. Risk Management- Using regression to the analysis of Risk Management in the financial and insurance sector.

However, Linear Regression is a very vast algorithm and it will be difficult to cover all of it. You can improve the model in various ways could be by detecting collinearity and by transforming predictors to fit nonlinear relationships. This article is to get you started with simple linear regression. Let’s quickly see the advantage and disadvantage of linear regression algorithm:

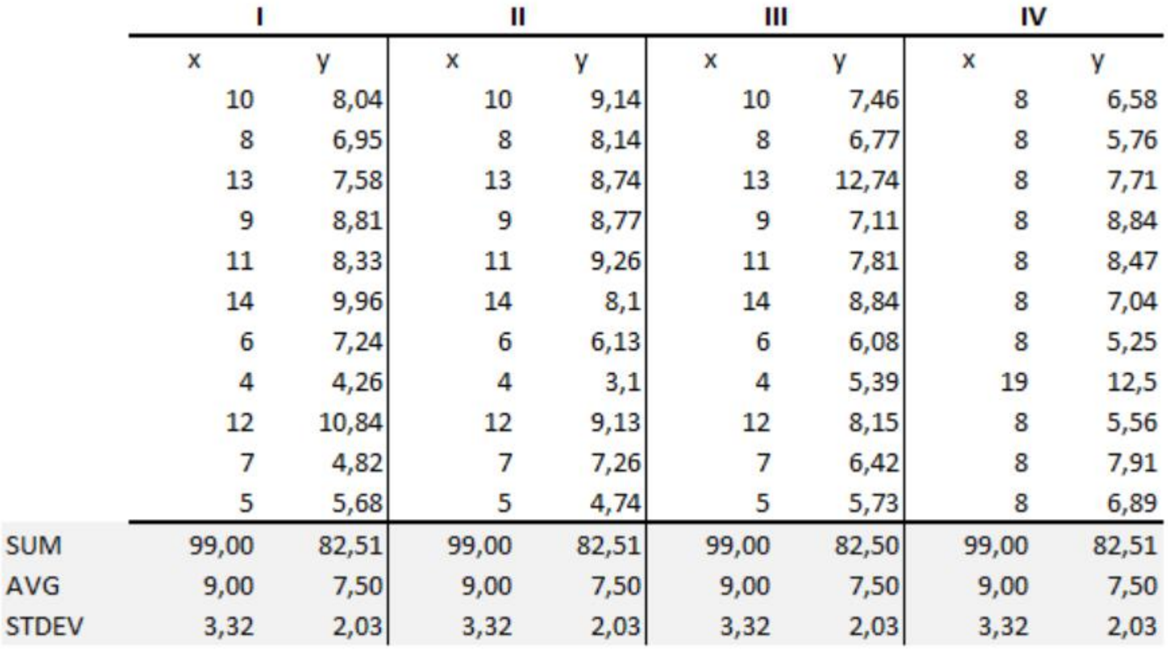
1. Linear regression provides a powerful statistical method to find the relationship between variables. It hardly needs further tuning. However, it’s only limited to linear relationships.
2. Linear regression produces the best predictive accuracy for linear relationship whereas its little sensitive to outliers and only looks at the mean of the dependent variable.

2. Explain the Anscombe’s quartet in detail.

Anscombe's quartet comprises four data sets that have nearly identical simple descriptive statistics, yet have very different distributions and appear very different when graphed. Each dataset consists of eleven (x,y) points.



Anscombe’s Quartet was developed by statistician Francis Anscombe. It comprises four datasets, each containing eleven (x,y) pairs. The essential thing to note about these datasets is that they share the same descriptive statistics. But things change completely, and I must emphasize **COMPLETELY,** when they are graphed. Each graph tells a different story irrespective of their similar summary statistics.



	I		II		III		IV	
	x	y	x	y	x	y	x	y
	10	8,04	10	9,14	10	7,46	8	6,58
	8	6,95	8	8,14	8	6,77	8	5,76
	13	7,58	13	8,74	13	12,74	8	7,71
	9	8,81	9	8,77	9	7,11	8	8,84
	11	8,33	11	9,26	11	7,81	8	8,47
	14	9,96	14	8,1	14	8,84	8	7,04
	6	7,24	6	6,13	6	6,08	8	5,25
	4	4,26	4	3,1	4	5,39	19	12,5
	12	10,84	12	9,13	12	8,15	8	5,56
	7	4,82	7	7,26	7	6,42	8	7,91
	5	5,68	5	4,74	5	5,73	8	6,89
SUM	99,00	82,51	99,00	82,51	99,00	82,50	99,00	82,51
AVG	9,00	7,50	9,00	7,50	9,00	7,50	9,00	7,50
STDEV	3,32	2,03	3,32	2,03	3,32	2,03	3,32	2,03

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

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 :

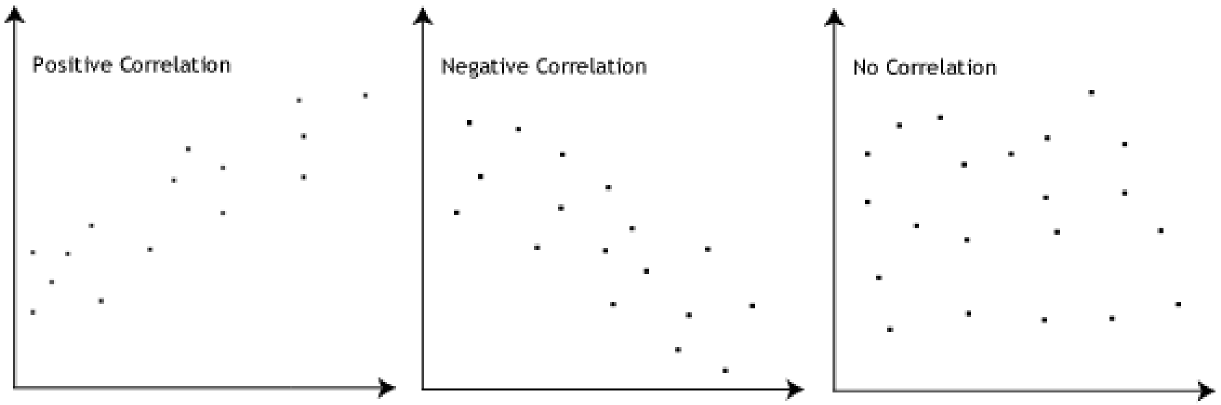
- Dataset I appears to have clean and well-fitting linear models.
- Dataset II is not distributed normally.
- In Dataset III the distribution is linear, but the calculated regression is thrown off by an outlier.
- Dataset IV shows that one outlier is enough to produce a high correlation coefficient.

This quartet emphasizes the importance of visualization in Data Analysis. Looking at the data reveals a lot of the structure and a clear picture of the dataset.

3. What is Pearson's R?

The Pearson product-moment correlation coefficient (or Pearson correlation coefficient, for short) is a measure of the strength of a linear association between two variables and is denoted by *r*. Basically, a Pearson product-moment correlation attempts to draw a line of best fit through the data of two variables, and the Pearson correlation coefficient, *r*, indicates how far away all these data points are to this line of best fit (i.e., how well the data points fit this new model/line of best fit).

The Pearson correlation coefficient, *r*, can take a range of values from +1 to -1. A value of 0 indicates that there is no association between the two variables. A value greater than 0 indicates a positive association; that is, as the value of one variable increases, so does the value of the other variable. A value less than 0 indicates a negative association; that is, as the value of one variable increases, the value of the other variable decreases. This is shown in the diagram below:



The first and **most important** step before analysing your data using Pearson’s correlation is to check whether it is appropriate to use this statistical test. After all, Pearson’s correlation will only give you **valid/accurate results** if your study design and data "**pass/meet**" **seven assumptions** that underpin Pearson’s correlation.

In many cases, Pearson’s correlation will be the **incorrect** statistical test to use because your data "**violates/does not meet**" one or more of these assumptions. This is not uncommon when working with real-world data, which is often "messy", as opposed to textbook examples. However, there is often a solution, whether this involves using a **different statistical test**, or making **adjustments** to your data so that you can continue to use Pearson’s correlation.

- Assumption #1: Your two variables should be measured on a continuous scale (i.e., they are measured at the interval or ratio level). Examples of continuous variables include revision time (measured in hours), intelligence (measured using IQ score), exam performance (measured from 0 to 100), weight (measured in kg), driving speed (measured in km/h) and so forth.
- Assumption #2: Your two continuous variables should be paired, which means that each case (e.g., each participant) has two values: one for each variable. These "values" are also referred to as "data points".

  For example, imagine that you had collected the revision times (measured in hours) and exam results (measured from 0 to 100) from 100 randomly sampled students at a university (i.e., you have two continuous variables: "revision time" and "exam performance"). Each of the 100 students would have a value for revision time (e.g., "student #1" studied for "23 hours") and an exam result (e.g., "student #1" scored "81 out of 100"). Therefore, you would have 100 paired values.
- Assumption #3: There should be independence of cases, which means that the two observations for one case (e.g., the scores for revision time and exam performance for "student #1") should be independent of the two observations for any other case (e.g., the scores for revision time and exam

performance for "student #2", or "student #3", or "student #50", for example). If observations are not independent, they are related, and Pearson's correlation is not an appropriate statistical test (although there are other measures of association that can be used when you have observations that are not independent).

For example, if some of the 100 students were in a revision study group, we might expect the relationship between revision time and exam performance for those students to be more similar when compared to other students, violating the independence of cases assumption. Alternatively, if some of the 100 students included siblings (e.g., two sisters), you might expect the relationship between revision time and exam performance of those two sisters to be more similar compared to the other students, again violating the independence of cases assumption.

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

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

Example: If an algorithm is not using feature scaling method then it can consider the value 3000 meter to be greater than 5 km but that's actually not true and in this case, the algorithm will give wrong predictions. So, we use Feature Scaling to bring all values to same magnitudes and thus, tackle this issue.

Techniques to perform Feature Scaling
Consider the two most important ones:

- Min-Max Normalization: This technique re-scales a feature or observation value with distribution value between 0 and 1.

$$X_{new} = \frac{X_i - \min(X)}{\max(x) - \min(X)}$$

- Standardization: It is a very effective technique which re-scales a feature value so that it has distribution with 0 mean value and variance equals to 1.

$$X_{new} = \frac{X_i - X_{mean}}{\text{Standard Deviation}}$$

The terms normalization and standardization are sometimes used interchangeably, but they usually refer to different things. Normalization usually means to scale a variable to have a values between 0 and 1, while standardization transforms data to have a mean of zero and a standard deviation of 1.

Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.

Here, Xmax and Xmin are the maximum and the minimum values of the feature respectively.

- When the value of X is the minimum value in the column, the numerator will be 0, and hence X’ is 0
- On the other hand, when the value of X is the maximum value in the column, the numerator is equal to the denominator and thus the value of X’ is 1
- If the value of X is between the minimum and the maximum value, then the value of X’ is between 0 and 1
- **Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.**

- Here’s the formula for standardization:

- $$X' = \frac{X - \mu}{\sigma}$$

- μ is the mean of the feature values and σ is the standard deviation of the feature values. Note that in this case, the values are not restricted to a particular range.

- Now, the big question in your mind must be when should we use normalization and when should we use standardization? Let’s find out!

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

An infinite VIF value indicates that the corresponding variable may be expressed exactly by a linear combination of other variables (which show an infinite VIF as well).

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.

For any predictor orthogonal (independent) to all other predictors, the variance inflation factor is 1.0. VIFi thus provides us with a measure of how many times larger the variance of the ith regression coefficient will be for multicollinear data than for orthogonal data (where each VIF is 1.0). If the VIF's are not unusually larger than 1.0, multicollinearity is not a problem. An advantage of knowing the VIF for each variable is that it gives a tangible idea of how much of the variances of the estimated coefficients are degraded by the multicollinearity. VIF's may be printed using the VI=Y option.

In order to determine VIF, we fit a regression model between the independent variables. For example, we would fit the following models to estimate the coefficient of determination R1 and use this value to estimate the VIF:

$X_1=C+ \alpha_2 X_2+\alpha_3 X_3+\cdots$

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

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

$X_2=C+ \alpha_1 X_1+\alpha_3 X_3+\cdots$

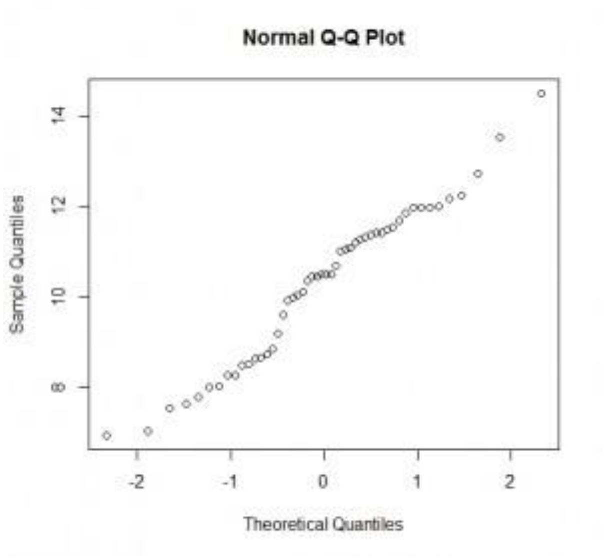
$\text{VIF}_2=1/(1-R_2^2)$

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. If the VIF is 4, this means that the variance of the model coefficient is inflated by a factor of 4 due to the presence of multicollinearity. This would mean that that standard error of this coefficient is inflated by a factor of 2 (square root of variance is the standard deviation). The standard error of the coefficient determines the confidence interval of the model coefficients. If the standard error is large, then the confidence intervals may be large, and the model coefficient may come out to be non-significant due to the presence of multicollinearity. 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.

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

The Q-Q plot, or quantile-quantile plot, is a graphical tool to help us assess if a set of data plausibly came from some theoretical distribution such as a Normal or exponential. For example, if we run a statistical analysis that assumes our dependent variable is Normally distributed, we can use a Normal Q-Q plot to check that assumption. It’s just a visual check, not an air-tight proof, so it is somewhat subjective. But it allows us to see at-a-glance if our assumption is plausible, and if not, how the assumption is violated and what data points contribute to the violation.

A Q-Q plot is a scatterplot created by plotting two sets of quantiles against one another. If both sets of quantiles came from the same distribution, we should see the points forming a line that’s roughly straight. Here’s an example of a Normal Q-Q plot when both sets of quantiles truly come from Normal distributions.



Now what are “quantiles”? These are often referred to as “percentiles”. These are points in your data below which a certain proportion of your data fall. For example, imagine the classic bell-curve standard Normal distribution with a mean of 0. The 0.5 quantile, or 50th percentile, is 0. Half the data lie below 0. That’s the peak of the hump in the curve. The 0.95 quantile, or 95th percentile, is about 1.64. 95 percent of the data lie below 1.64. The following R code generates the quantiles for a standard Normal distribution from 0.01 to 0.99 by increments of 0.01:

```
qnorm(seq(0.01,0.99,0.01))
```

We can also randomly generate data from a standard Normal distribution and then find the quantiles. Here we generate a sample of size 200 and find the quantiles for 0.01 to 0.99 using the quantile function:

```
quantile(rnorm(200),probs = seq(0.01,0.99,0.01))
```

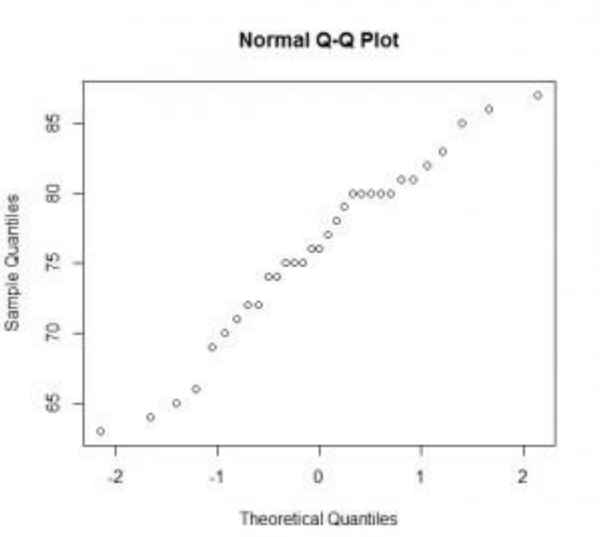
So we see that quantiles are basically just your data sorted in ascending order, with various data points labelled as being the point below which a certain proportion of the data fall. However it’s worth noting there are many ways to calculate quantiles. In fact, the quantile function in R offers 9 different quantile algorithms! See help(quantile) for more information.

Q-Q plots take your sample data, sort it in ascending order, and then plot them versus quantiles calculated from a theoretical distribution. The number of quantiles is selected to match the size of your sample data. While Normal Q-Q Plots are the ones most often used in practice due to so many statistical methods assuming normality, Q-Q Plots can actually be created for any distribution.

In R, there are two functions to create Q-Q plots: qqnorm and qqplot.

qqnorm creates a Normal Q-Q plot. You give it a vector of data and R plots the data in sorted order versus quantiles from a standard Normal distribution. For example, consider the trees data set that comes with R. It provides measurements of the girth, height and volume of timber in 31 felled black cherry trees. One of the variables is Height. Can we assume our sample of Heights comes from a population that is Normally distributed?

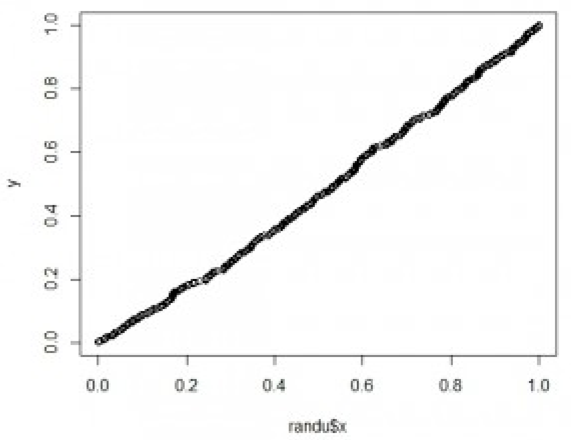
```
qqnorm(trees$Height)
```



That appears to be a fairly safe assumption. The points seem to fall about a straight line. Notice the x-axis plots the theoretical quantiles. Those are the quantiles from the standard Normal distribution with mean 0 and standard deviation 1.

The qqplot function allows you to create a Q-Q plot for any distribution. Unlike the qqnorm function, you have to provide two arguments: the first set of data and the second set of data. Let's look at the randu data that come with R. It's a data frame that contains 3 columns of random numbers on the interval (0,1). Random numbers should be uniformly distributed. Therefore we can check this assumption by creating a Q-Q plot of the sorted random numbers versus quantiles from a theoretical uniform (0,1) distribution. Here we create a Q-Q plot for the first column numbers, called x:

```
y <- qunif(ppoints(length(randu$x)))
qqplot(randu$x,y)
```

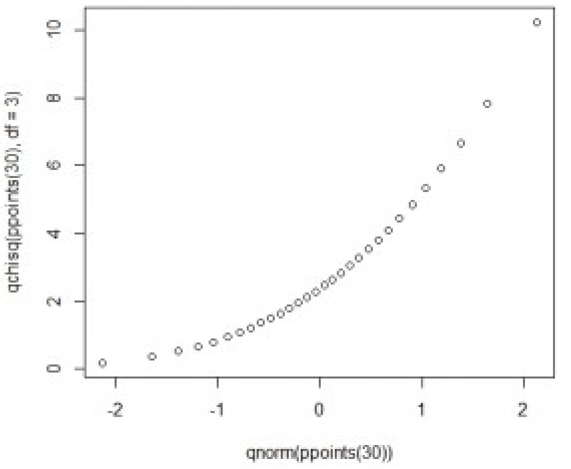


The ppoints function generates a given number of probabilities or proportions. I wanted the same number of values in randu$x, so I gave it the argument length(randu$x), which returns 400.

The qunif function then returns 400 quantiles from a uniform distribution for the 400 proportions. I save that to y and then plot y versus randu$x in the qqplot function. Again, we see points falling along a straight line in the Q-Q plot, which provide strong evidence that these numbers truly did come from a uniform distribution.

What about when points don’t fall on a straight line? What can we infer about our data? To help us answer this, let’s generate data from one distribution and plot against the quantiles of another. First we plot a distribution that’s skewed right, a Chi-square distribution with 3 degrees of freedom, against a Normal distribution.

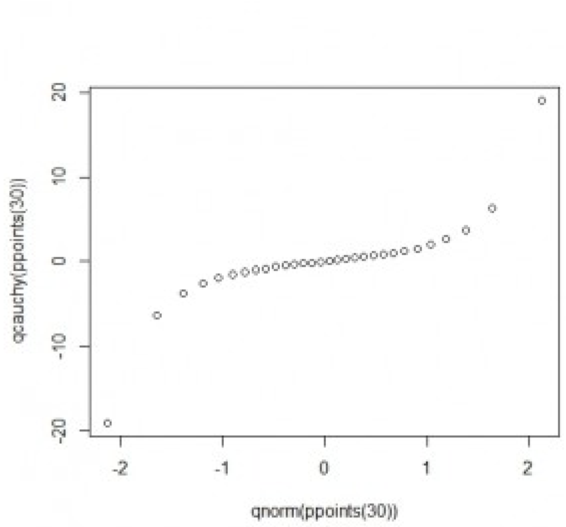
```
qqplot(qnorm(ppoints(30)), qchisq(ppoints(30),df=3))
```



Notice the points form a curve instead of a straight line. Normal Q-Q plots that look like this usually mean your sample data are skewed.

Next we plot a distribution with “heavy tails” versus a Normal distribution:

```
qqplot(qnorm(ppoints(30)), qcauchy(ppoints(30)))
```



Notice the points fall along a line in the middle of the graph, but curve off in the extremities. Normal Q-Q plots that exhibit this behavior usually mean your data have more extreme values than would be expected if they truly came from a Normal distribution.