**Simple Linear Regression:**

**Summary**  
  
Here's a brief summary of what you learned in this session:

1. Machine learning models can be classified into following two categories on the basis of learning algorithm:
   * **Supervised learning method:**Past data with labels is available to build the model
     + **Regression:**The output variable is continuous in nature
     + **Classification:**The output variable is categorical in nature
   * **Unsupervised learning method:**Past data with labels are not available
     + **Clustering:** No pre-defined notion of labels is there
2. Past data set is divided into two parts during supervised learning method:
   * **Training data**is used for the model to learn during modelling
   * **Testing data**is used by the trained model for prediction and model evaluation
3. Linear regression models can be classified into two types depending upon the number of independent variables:
   * **Simple linear regression:** When the number of independent variables is 1
   * **Multiple linear regression:** When the number of independent variables is more than 1
4. The equation of the best fit regression line Y = β₀ + β₁X can be found by minimising the cost function (RSS in this case, using the Ordinary Least Squares method) which is done using the following two methods:
   * **Differentiation**
   * **Gradient descent method**
5. The strength of a linear regression model is mainly explained by R²,  whereR² = 1 - (RSS / TSS)
   * **RSS:** Residual Sum of Squares
   * **TSS:** Total Sum of Squares

You are making inferences on the 'population' using a 'sample'. The assumption that variables are linearly dependent is not enough to generalise the results you obtain on a sample to the **population**, which is much larger in size than the sample. Thus, you need to have certain assumptions in place in order to make inferences.

Let's understand the importance of each assumption one by one:

**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.

A diagram of a graph

AI-generated content may be incorrect.

**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.

A comparison of a graph

AI-generated content may be incorrect.

**Image sources:**

* [Assumptions of Simple Linear Regression](https://reliawiki.org/index.php/Simple_Linear_Regression_Analysis)

You can also learn more about the simple linear regression in the above link.

**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).

A diagram of a graph

AI-generated content may be incorrect.

**Image sources:**

* [Assumptions of Simple Linear Regression](https://www.jmp.com/en_in/statistics-knowledge-portal/what-is-regression.html)

**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.

**ACPML - Linear Regression - Simple Linear Regression in Python - Hypothesis Testing in Linear Regression**

**Hypothesis Testing in Linear Regression**

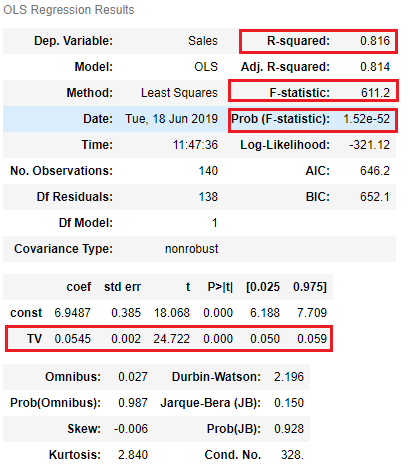
Before you move on to the model-building part, there is still one theoretical aspect left to be addressed: the significance of the derived beta coefficient. When you fit a straight line through the data, you'll obviously get the two parameters of the straight line, i.e. the intercept (β0) and the slope (β1). Now, while β0 is not of much importance right now, but there are a few aspects surrounding β1 which need to be checked and verified.

The first question we ask is, "Is the beta coefficient significant?" What does this mean?

Suppose you have a dataset for which the scatter plot looks like the following:

**OLS Summary Statistics:**

Now, let's take a look at the summary statistics that was outputted by the model again.



**F-statistic**

You were introduced to a new term named **F-statistic** and **Prob(F-statistic)**. Now, recall that in the last segment, you did a hypothesis test for beta to determine whether or not the coefficient β1 outputted by the model was significant or not. Now, F-statistic is similar in the sense that now instead of testing the significance of each of the betas, it tells you whether the overall model fit is significant or not. This parameter is examined because many a time it happens that even though all of your betas are significant, but your overall model fit might happen just by chance.

The heuristic is similar to what you learnt in the normal p-value calculation as well. If the '**Prob (F-statistic)'** is less than **0.05**, you can conclude that the overall model fit is significant. If it is greater than 0.05, you might need to review your model as the fit might be by chance, i.e. the line may have just luckily fit the data. In the image above, you can see that the p-value of the F-statistic is **1.52e-52** which is practically a zero value. This means that the model for which this was calculated is definitely significant since it is less than 0.05.

This will be more appreciable when you study multiple linear regression since there you have a lot of betas for the different predictor variables and thus it is very helpful in determining if all the predictor variables together as a whole are significant or not or simply put, it tells you whether the model fit as a whole is significant or not.

**R-squared**

Like you studied earlier as well, R-squared value tells you exactly how much variance in the data has been explained by the model. In our case, the R-squared is about 0.816 which means that the model is able to explain 81.6% of the variance which is pretty good.

**Coefficients and p-values:**

The p-values of the coefficients (in this case just one coefficient for TV) tell you whether the coefficient is significant or not. In this case, the coefficient of TV came out to be **0.0545** with a standard error of about **0.002**. Thus, you got a t-value of **24.722**which lead to a practically **zero p-value**. Hence, you can say that your coefficient is indeed significant.

Apart from this, the summary statistics outputs a few more metrics which are not of any use as of now. But you'll learn about some more of them in multiple linear regression.

**ACPML - Linear Regression - Simple Linear Regression in Python - Summary**

**Summary**  
  
In this session, you built a simple linear regression model in Python using the advertising dataset. You also saw some more theoretical aspects in between. Here's a brief of what you learnt in this session.

1. A quick recap of simple linear regression
2. Assumptions of simple linear regression
   * Linear relationship between X and y.
   * Normal distribution of error terms.
   * Independence of error terms.
   * Constant variance of error terms.
3. Hypothesis testing in linear regression
   * To determine the significance of beta coefficients.
   * H0:β1=0;HA:β1≠0.
   * T-test on the beta coefficient.
   * t score=^βiSE(^βi).
4. Building a linear model
   * OLS (Ordinary Least Squares) method in statsmodels to fit a line.
   * Summary statistics
     + F-statistic, R-squared, coefficients and their p-values.
5. Residual Analysis
   * Histogram of the error terms to check normality.
   * Plot of the error terms with X or y to check independence.
6. Predictions
   * Making predictions on the test set using the 'predict()' function.
7. Linear Regression using SKLearn
   * A second package apart from statsmodels for linear regression.
   * A more hassle-free package to just fit a line without any inferences.

Rahim has also answered some common doubts surrounding linear regression. This part has also been included in the notebook provided to you at the beginning of the session.

**Multiple Linear Regressions:**

Most of the concepts in multiple linear regression are quite similar to those in simple linear regression. The formulation for predicting the response variable now becomes:

Y=β0+β1X1+β2X2+…+βpXp+ϵ  
 

Apart from the formulation, there are some other aspects that still remain the same:

1. The model now fits a hyperplane instead of a line
2. Coefficients are still obtained by minimising the sum of squared errors, the least squares criteria
3. For inference, the assumptions from simple linear regression still hold - zero-mean, independent and normally distributed error terms with constant variance

Coming up

But there are still a few considerations that need to be made when moving from simple to multiple linear regression. Let's discuss them in the next segment.

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

**Image used in the above question are from the following sources:**

In the link below, you can understand more about the overfitting concept.

* [Overfitting](https://elitedatascience.com/overfitting-in-machine-learning)

**Multicollinearity:**

Multicollinearity refers to the phenomenon of having related predictor variables in the input dataset. In simple terms, in a model which has been built using several independent variables, some of these variables might be interrelated, due to which the presence of that variable in the model is redundant. You drop some of these related independent variables as a way of dealing with 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

Multicollinearity is, thus, a big issue when you are trying to interpret the model. It is essential to detect and deal with the multicollinearity present in the model.

You saw two basic ways of dealing with multicollinearity

1. Looking at **pairwise correlations**
   * Looking at the correlation between different pairs of independent variables
2. 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 by:

                                                                            VIFi=11-Ri2

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.

But once you have detected the multicollinearity present in the dataset, how exactly do you deal with it?

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

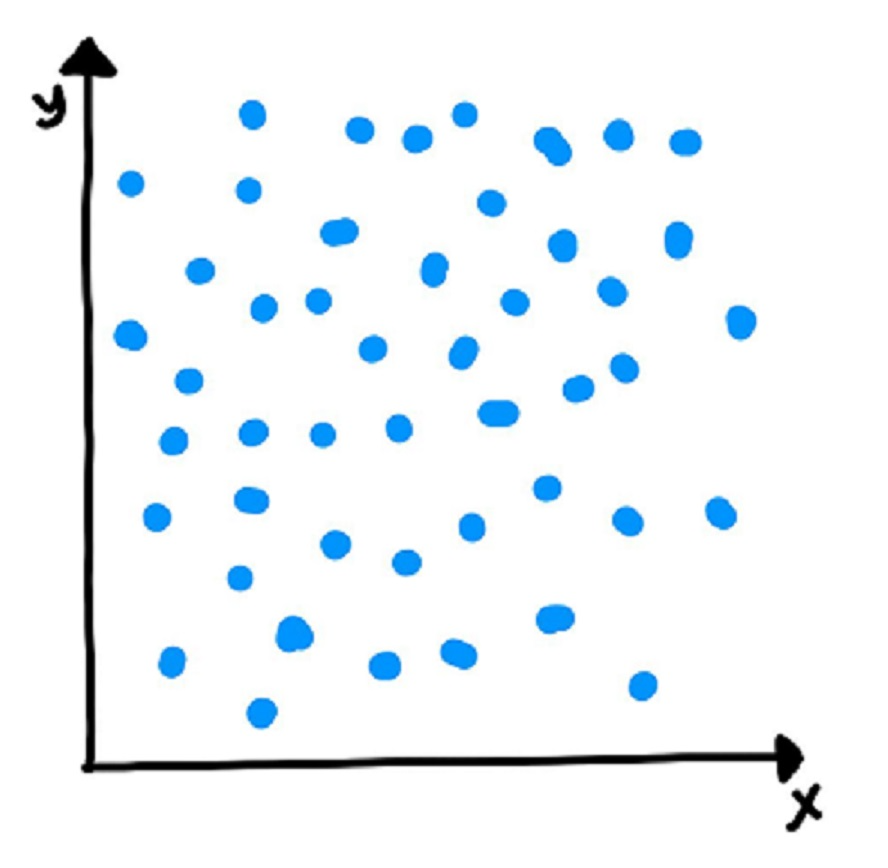
1. **Dropping variables**
   * Drop the variable which is highly correlated with others
   * Pick the business interpretable variable
2. **Create new variable**using the interactions of the older variables
   * Add interaction features, i.e. features derived using some of the original features
3. **Variable transformations**
   * Principal Component Analysis (covered in a later module)

**Coming up**

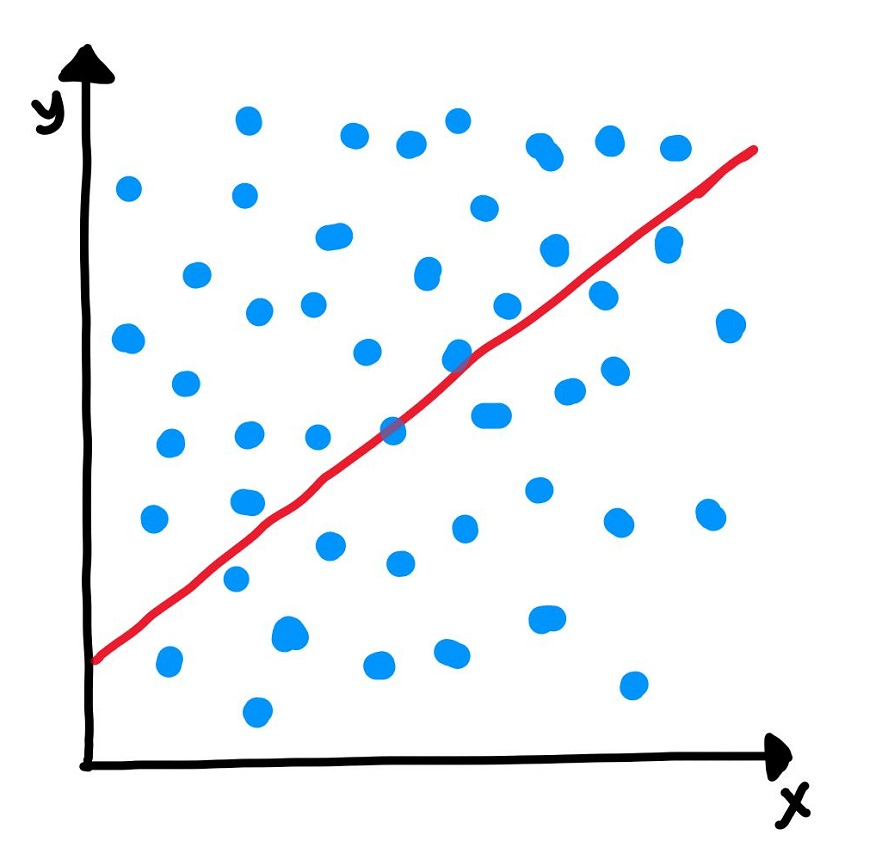
In the next segment, you will learn to handle the categorical variables present in the dataset.

**Additional Reading**

* [Partial Least Squares (PLS)](https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/regression/supporting-topics/partial-least-squares-regression/what-is-partial-least-squares-regression/)



Now, if you run a linear regression on this dataset in Python, it will fit a line on the data which, say, looks like the following:



Now, you can clearly see that the data is randomly scattered and doesn't seem to follow a linear trend or any trend, in general. But Python will anyway fit a line through the data using the least squared method. But you can see that the fitted line is of no use in this case.

Hence, every time you perform a linear regression, you need to test whether the fitted line is a significant one or not or to simply put it, you need to test whether β1 is significant or not. And in comes the idea of Hypothesis Testing on β1. **Please note** that the following text will assume the knowledge of hypothesis testing, which was covered in one of the earlier modules. Please revisit the module on hypothesis testing in case you need to brush up.

You start by saying that β1 is not significant, i.e. there is no relationship between X and y.

So in order to perform the hypothesis test, we first propose the null hypothesis that β1 is 0. And the alternative hypothesis thus becomes β1 is not zero.

* **Null Hypothesis (H0): β1=0**
* **Alternate Hypothesis (HA): β1≠0**

Let's first discuss the implications of this hypothesis test. If you fail to reject the null hypothesis that would mean that β1 is zero which would simply mean that β1 is insignificant and of no use in the model. Similarly, if you reject the null hypothesis, it would mean that β1 is not zero and the line fitted is a significant one.

Now, how do you perform the hypothesis test? Recall from your hypothesis testing module that you first used to compute the t-score (which is very similar to the Z-score) which is given by X-μs/n where μ is the population mean and s is the sample standard deviation which when divided by n is also known as standard error.

Using this, the t-score for β^1 comes out to be (since the null hypothesis is that β1 is equal to zero):

β^1-0SE(β^1)

Now, in order to perform the hypothesis test, you need to derive the p-value for the given beta. If you're hazy on what **p-value** is and how it is calculated, it is recommended that you revisit the segment on p-value. Please note that the formula of SE(β1) provided in the t-score above is out of scope of this course.

Let's do a quick recap of how do you calculate p-value anyway:

* Calculate the value of **t-score** for the mean point (in this case, zero, according to the Null hypothesis that we have stated) on the distribution
* Calculate the **p-value** from the cumulative probability for the given t-score using the t-table
* Make the decision on the basis of the p-value with respect to the given value of β (significance level)

Now, if the p-value turns out to be less than **0.05**,you can reject the null hypothesis and state that β1 is indeed significant.

Please note that all of the above steps will be performed automatically by the libraries we use Python which you'll learn in the very next segment.

Coming up

Now that you know how to determine whether your beta is significant or not, you'll start building the model in the next segment

**Additional Reading**

Why does the test statistic for β1 follow a t-distribution instead of a normal distribution? ([here](https://www.youtube.com/watch?v=78YNvrsRzVw&t=269s))  
  
Before you proceed further, spend some time answering the question next.

A diagram of a graph

AI-generated content may be incorrect.

**Image sources:**

* [Assumptions of Simple Linear Regression](https://www.jmp.com/en_in/statistics-knowledge-portal/what-is-regression.html)

You will look at each of these assumptions in more detail later and validate these while building the model.

You can also go through the following [link](http://people.duke.edu/~rnau/testing.htm)to see what happens when the assumptions are violated. But things will anyway get clearer once we keep moving ahead.

In the next segment you will learn how to read data and understand data which is a core part of machine learning.

**Categorical Variables**:

So far, you have worked with numerical variables. But many times, you will have non-numeric variables in the data sets. These variables are also known as categorical variables. Obviously, these variables can't be used directly in the model since they are non-numeric.

When you have a categorical variable with say 'n' levels, the idea of dummy variable creation is to build 'n-1' variables, indicating the levels. For a variable say, 'Relationship' with three levels namely, 'Single', 'In a relationship', and 'Married', you would create a dummy table like the following:

|  |  |  |  |
| --- | --- | --- | --- |
| Relationship Status | Single | In a relationship | Married |
| Single | 1 | 0 | 0 |
| In a relationship | 0 | 1 | 0 |
| Married | 0 | 0 | 1 |

But you can clearly see that there is no need of defining **three** different levels. If you drop a level, say 'Single', you would still be able to explain the three levels.

Let's drop the dummy variable 'Single' from the columns and see what the table looks like:

|  |  |  |
| --- | --- | --- |
| Relationship Status | In a relationship | Married |
| Single | 0 | 0 |
| In a relationship | 1 | 0 |
| Married | 0 | 1 |

If both the dummy variables namely 'In a relationship' and 'Married' are equal to zero, that means that the person is single. If 'In a relationship' is one and 'Married' is zero, that means that the person is in a relationship and finally, if 'In a relationship' is zero and 'Married' is 1, that means that the person is married.

**Feature 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.

There are two major methods to scale the variables, i.e. standardisation and MinMax scaling. Standardisation basically brings all of the data into a standard normal distribution with mean zero and standard deviation one. MinMax scaling, on the other hand, brings all of the data in the range of 0 and 1. The formulae in the background used for each of these methods are as given below:

* Standardisation: x=x-mean(x)sd(x)
* MinMax Scaling: x=x-min(x)max(x)-min(x)

Coming up

In the next segment, you will learn to assess and compare models. The concept of comparing models is a very important part as for a data set we can use many models but to use a correct mapped model to the problem statement is a core part of machine learning.

**Additional Reading**

* To know more about dummy variables ([here](https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faqwhat-is-dummy-coding/))
* Why it's necessary to create dummy variables ([here](https://stats.stackexchange.com/questions/89533/convert-a-categorical-variable-to-a-numerical-variable-prior-to-regression))
* When to Normalise data and when to standardise? ([here](https://stackoverflow.com/questions/32108179/linear-regression-normalization-vs-standardization))
* Various scaling techniques ([here](https://en.wikipedia.org/wiki/Feature_scaling))

**Model Assessment and Comparison**

Once the model is built, you would want to assess it in terms of its predictive powers. For multiple linear regression, you may build more than one model, with different combinations of the independent variables. In such a case, you would also need to compare these models with one another to check which one yields optimal results.

Now, for the assessment, you have a lot of new considerations to make. Besides, selecting the best model to obtain decent predictions becomes quite subjective. You need to maintain a balance between **keeping the model simple** and **explaining the highest variance** (which means that you would want to keep as many variables as possible). This can be done using the key idea that a model can be penalised for keeping a large number of predictor variables.

Hence, there are two new parameters that come into picture:

                                                              Adjusted R2=1-(1-R2)(N-1)N-p-1

Aℂ=n×logRSSn+2p

Here, n is the sample size meaning the number of rows you'd have in the dataset and p is the number of predictor variables.

Coming up

Adjusted R2 adjusts the value of R2 such that a model with a larger number of variables is penalized. In the next segment, Rahim will talk about feature selection.

**Additional Reading :**

The following links provide a detail study on AIC and other parameters used in automatic feature selection :

* [AIC](https://en.wikipedia.org/wiki/Akaike_information_criterion)
* [BIC](https://en.wikipedia.org/wiki/Bayesian_information_criterion)
* [Mallows' CP](https://en.wikipedia.org/wiki/Mallows%27s_Cp)

**Feature Selection - Part 1**

The one crucial aspect of multiple linear regression that remains to be discussed is feature selection. When building a multiple linear regression model, you might have quite a few potential predictor variables; selecting just the right ones becomes an extremely important exercise.

Let’s see how you can **select the optimal features** for building a good model.

To get the optimal model, you can always try all the possible combinations of independent variables and see which model fits the best. But this method is obviously, time-consuming and infeasible. Hence, you need some other method to get a decent model. This is where manual feature elimination comes in, where you:

1. Build the model with all the features
2. Drop the features that are least helpful in prediction (high p-value)
3. Drop the features that are redundant (using correlations and VIF)
4. Rebuild model and repeat

Note that, the second and third steps go hand in hand and the choice of which features to eliminate first is very subjective. You'll see this during the hands-on demonstration of multiple linear regression in Python in the next session.

Now, manual feature elimination might work when you have a relatively low number of potential predictor variables, say, ten or even twenty. But it is not a practical approach once you have a large number of features, say 100. In such a case, you automate the feature selection (or elimination) process. Let's see how.

You need to combine the manual and the automated approaches in order to get an optimal model relevant to the business. Hence, you first do an automated elimination (coarse tuning), and when you have a small set of potential variables left to work with, you can use your expertise and subjectivity to eliminate a few other features (fine tuning)

**ACPML - Linear Regression - Multiple Linear Regression - Summary**

**Summary**  
  
Here’s a brief summary of what you learned in this session:

1. When one variable might not be enough
   * A lot of variance isn’t explained by just one feature
   * Inaccurate predictions
2. Formulation of MLR: MLR helps us to understand how much will the dependent variable change when we change the independent variables.
3. New considerations to be made when moving from SLR to MLR
   * Overfitting - When the model becomes complex and gives very good results in training data and fails in the testing data.
   * Multicollinearity - To identify if there is any dependency within the pool of independent variables to remove redundancy.
   * Feature selection - Out of the pool of many features what features are considered to be most important. We drop the redundant features and those features that are not helpful in prediction.
4. Dealing with categorical variables
   * Dummy variables - USed when there are fewer levels. You learnt about it using the marital status example.
5. Feature Scaling
   * Standardisation - Method used to make sure that **data** is internally consistent.
   * MinMax scaling - Method used to make sure that **data** is internally consistent.
   * Scaling for categorical variables - Categorical variables cannot used as they are, so they are converted to numeric format.
6. Model Assessment and Comparison
   * Adjusted R-squared - The **adjusted R**-**squared** value increases only if the new term improves the model more than would be expected by chance.
   * AIC, BIC - Various types of criteria used for automatic feature selection
7. Feature Selection
   * Manual feature selection - A very tedious task in order to select the correct set of features.
   * Automated feature selection - The three step process is involved.
     + Select top 'n' features
     + Forward/backward/Stepwise selection based on AIC
     + Regularization
   * Finding a balance between the two - A balance of both manual and automatic feature selection is required to attain the features.

**Linear Regression - Industry Relevance of Linear Regression - Linear Regression:**

When you wish to predict the Y for X1, which lies between a and b, it is called interpolation. On the other hand, extrapolation would be extending the line to predict Y for X2 which lies outside the range on which the linear model was trained.

For now, you only need to understand what these terms mean. You will learn more about this in the upcoming lectures.

Ujjyaini also mentioned that linear regression is a parametric model, as opposed to non-parametric ones.

A detailed discussion on parametric and non-parametric models is beyond the discussion of this module, though a simple explanation is given below. You may also read up the additional resources given below.

In simple terms, a parametric model can be described using a finite number of parameters. For e.g., a linear regression model built using n independent variables will have exactly n ‘parameters’ (i.e. the n coefficients). The entire model can be described using these n parameters.

In the upcoming modules, you will learn some ‘non-parametric’ models as well, such as decision trees.

They do not have a finite set of parameters which completely describe the model.

To read up on this topic further, here are some useful links:

* [Parametric v/s non parametric models in short](https://stats.stackexchange.com/questions/268638/confusion-about-parametric-and-non-parametric-model)
* [A detailed explanation on parametric and non-parametric models](http://machinelearningmastery.com/parametric-and-nonparametric-machine-learning-algorithms/)

It is very crucial to understand when to apply linear regression modelling.

A good model tells a good story. It is not important that you base your story entirely on a single model. Drawing insights from the current model (model 9) and previous models, Ujjyaini identified **Ad Impressions** and **Character A** as the driver variables that could be used to increase the viewership of the show. **Ad impressions are directly proportional to the marketing budget**. Thus, by increasing the marketing budget, a better viewership could be achieved. Similarly, Character A’s absence and presence create a significant change in show viewership. Character A’s presence brings viewers to the show. Thus, these two variables can be acted upon to improve the show viewership.

Additional reading

Read more about the bootstrap method [here](http://rstudio-pubs-static.s3.amazonaws.com/24365_2803ab8299934e888a60e7b16113f619.html).

In the next segment, you will interpret the results that you have obtained.

**Subjective Questions - I**

It is a common practice to test data science aspirants on linear regression as it is the first algorithm that almost everyone studies in Data Science/Machine Learning. Aspirants are expected to possess an in-depth knowledge of these algorithms. We consulted hiring managers and data scientists from various organisations to know about the typical Linear Regression questions which they ask in an interview. Based on their extensive feedback a set of question and answers were prepared to help students in their conversations.

**Q1. What is linear regression?**

In simple terms, linear regression is a method of finding the best straight line fitting to the given data, i.e. finding the best linear relationship between the independent and dependent variables.

In technical terms, linear regression is a machine learning algorithm that finds the best linear-fit relationship on any given data, between independent and dependent variables. It is mostly done by the Sum of Squared Residuals Method.

**Q2. What are assumptions in a linear regression model?**

The assumptions of linear regression are:

1. The assumption about the form of the model: It is assumed that there is a linear relationship between the dependent and independent variables. It is known as the ‘linearity assumption’.
2. Assumptions about the residuals:
   1. Normality assumption: It is assumed that the error terms, ε(i), are normally distributed.
   2. Zero mean assumption: It is assumed that the residuals have a mean value of zero, i.e., the error terms are normally distributed around zero.
   3. Constant variance assumption: It is assumed that the residual terms have the same (but unknown) variance, σ2 . This assumption is also known as the assumption of homogeneity or homoscedasticity.
   4. Independent error assumption: It is assumed that the residual terms are independent of each other, i.e. their pair-wise covariance is zero.
3. Assumptions about the estimators:
   1. The independent variables are measured without error.
   2. The independent variables are linearly independent of each other, i.e. there is no multicollinearity in the data.

*Explanations:*

1. This is self-explanatory.
2. If the residuals are not normally distributed, their randomness is lost, which implies that the model is not able to explain the relation in the data.  
   Also, the mean of the residuals should be zero.  
   Y(i)i= β0+ β1x(i) + ε(i)  
   This is the assumed linear model, where ε is the residual term.  
   E(Y) = E(β0+ β1x(i) + ε(i))  
   = E(β0+ β1x(i) + ε(i))  
   If the expectation(mean) of residuals, E(ε(i)), is zero, the expectations of the target variable and the model become the same, which is one of the targets of the model.  
   The residuals (also known as error terms) should be independent. This means that there is no correlation between the residuals and the predicted values, or among the residuals themselves. If some correlation is present, it implies that there is some relation that the regression model is not able to identify.
3. If the independent variables are not linearly independent of each other, the uniqueness of the least squares solution (or normal equation solution) is lost.

**Q3. What is heteroscedasticity? What are the consequences, and how can you overcome it?**

A random variable is said to be heteroscedastic when different subpopulations have different variabilities (standard deviation).

The existence of heteroscedasticity gives rise to certain problems in the regression analysis as the assumption says that error terms are uncorrelated and, hence, the variance is constant. The presence of heteroscedasticity can often be seen in the form of a cone-like scatter plot for residual vs fitted values.

One of the basic assumptions of linear regression is that the data should be homoscedastic, i.e., heteroscedasticity is not present in the data. Due to the violation of assumptions, the Ordinary Least Squares (OLS) estimators are not the [Best Linear Unbiased Estimators (BLUE)](https://en.wikipedia.org/wiki/Gauss%E2%80%93Markov_theorem). Hence, they do not give the least variance than other Linear Unbiased Estimators (LUEs).

There is no fixed procedure to overcome heteroscedasticity. However, there are some ways that may lead to a reduction of heteroscedasticity. They are —

1. Logarithmising the data: A series that is increasing exponentially often results in increased variability. This can be overcome using the log transformation.
2. Using weighted linear regression: Here, the OLS method is applied to the weighted values of X and Y. One way is to attach weights directly related to the magnitude of the dependent variable.

**Q4. How do you know that linear regression is suitable for any given data?**  
To see if linear regression is suitable for any given data, a scatter plot can be used. If the relationship looks linear, we can go for a linear model. But if it is not the case, we have to apply some transformations to make the relationship linear. Plotting the scatter plots is easy in case of simple or univariate linear regression. But in the case of multivariate linear regression, two-dimensional pairwise scatter plots, rotating plots, and dynamic graphs can be plotted.

**Q5. How is hypothesis testing used in linear regression?**

Hypothesis testing can be carried out in linear regression for the following purposes:

1. To check whether a predictor is significant for the prediction of the target variable. Two common methods for this are —
   1. By the use of p-values:  
      If the p-value of a variable is greater than a certain limit (usually 0.05), the variable is insignificant in the prediction of the target variable.
   2. By checking the values of the regression coefficient:  
      If the value of the regression coefficient corresponding to a predictor is zero, that variable is insignificant in the prediction of the target variable and has no linear relationship with it.
2. To check whether the calculated regression coefficients are good estimators of the actual coefficients.

The Null and Alternate Hypothesis used in the case of linear regression, respectively, are:

β1=0

β1≠0

Thus, if we reject the Null hypothesis, we can say that the coefficient β1 is not equal to zero and hence, is significant for the model. On the other hand, if we fail to reject the Null hypothesis, it is concluded that the coefficient is insignificant and should be dropped from the model.

**Q6. How do you interpret a linear regression model?**

A linear regression model is quite easy to interpret. The model is of the following form:

                                               y=β0+β1X1+β2X2+...+βnXn

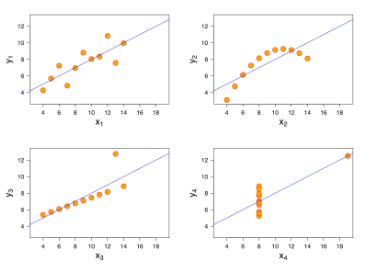
The significance of this model lies in the fact that one can easily interpret and understand the marginal changes and their consequences. For example, if the value of *x0* increases by 1 unit, keeping other variables constant, the total increase in the value of *y* will be *βi*. Mathematically, the intercept term (*β0*) is the response when all the predictor terms are set to zero or not considered.

**Q7. What are the shortcomings of linear regression?**

You should never just run a regression without having a good look at your data because simple linear regression has quite a few shortcomings:

1. It is sensitive to outliers
2. It models the linear relationships only
3. A few assumptions are required to make the inference

These **phenomena** can be best explained by the Anscombe's Quartet, shown below:



As we can see, all the four linear regression are exactly the same. But there are some peculiarities in the datasets which have fooled the regression line. While the first one seems to be doing a decent job, the second one clearly shows that linear regression can only model linear relationships and is incapable of handling any other kind of data. The third and fourth images showcase the linear regression model's sensitivity to outliers. Had the outlier not been present, we could have gotten a great line fitted through the data points. So we should never ever run a regression without having a good look at our data.

**Q8. What parameters are used to check the significance of the model and the goodness of fit?**

To check if the overall model fit is significant or not, the primary parameter to be looked at is the **F-statistic**. While the t-test along with the p-values for betas test if each coefficient is significant or not individually, the F-statistic is a measure that can determine whether the overall model fit with all the coefficients is significant or not.

The basic idea behind the F-test is that it is a relative comparison between the model that you've built and the model without any of the coefficients except for β0. If the value of the F-statistic is high, it would mean that the Prob(F) would be low and hence, you can conclude that the model is significant. On the other hand, if the value of F-statistic is low, it might lead to the value of Prob(F) being higher than the significance level (taken 0.05, usually) which in turn would conclude that the overall model fit is insignificant and the intercept-only model can provide a better fit.

Apart from that, to test the goodness or the extent of fit, we look at a parameter called **R-squared** (for simple linear regression models) or **Adjusted R-squared** (for multiple linear regression models). If your overall model fit is deemed to be significant by the F-test, you can go ahead and look at the value of R-squared. This value lies between 0 and 1, with 1 meaning a perfect fit. A higher value of R-squared is indicative of the model being good with much of the variance in the data being explained by the straight line fitted. For example, an R-squared value of 0.75 means that 75% of the variance in the data is being explained by the model. But it is important to remember than R-squared only tells the extent of the fit and should not be used to determine whether the model fit is significant or not.

**Q9. If two variables are correlated, is it necessary that they have a linear relationship?**

No, not necessarily. If two variables are correlated, it is very much possible that they have some other sort of relationship and not just a linear one.

But the important point to note here is that there are two correlation coefficients that are widely used in regression. One is the Pearson's R correlation coefficient which is the correlation coefficient you've studied in the linear regression model. This correlation coefficient is designed for linear relationships and it might not be a good measure for if the relationship between the variables is non-linear. The other correlation coefficient is Spearman's R which is used to determine the correlation if the relationship between the variables is not linear. So even though, Pearson's R might give a correlation coefficient for non-linear relationships, it might not be reliable. For example, the correlation coefficients as given by both the techniques for the relationship y=X3 for 100 equally separated values between 1 and 100 were found out to be:

                                                                  Pearson′s R≈0.91

                                                                   Spearman′s R≈1

And as we keep on increasing the power, the Pearson's R value consistently drop whereas the Spearman's R remains robust at 1. For example, for the relationship y=X10 for the same data points, the coefficients were:

                                                                  Pearson′s R≈0.66

                                                                    Spearman′s R≈1

So the takeaway here is that if you have some sense of the relationship being non-linear, you should look at Spearman's R instead of Pearson's R. It might happen that even for a non-linear relationship, the Pearson's R value might be high, but it is simply not reliable.

**Q10. What is the difference between Least Square Error and Mean Square Error?**

Least Square Error is the method used to find the best-fit line through a set of data points. It is. The idea behind the least squared error method is to minimize the square of errors between the actual data points and the line fitted.

Mean Square Error, on the other hand, is used once you have fitted the model and want to evaluate it. So the mean squared error finds out the average of the difference between the actual and predicted values and hence, is a good parameter to compare various models on the same data set.

Thus, LSE is a method used to minimise the sum of squares and is used during model fitting, and MSE is a metric used to evaluate the model after fitting based on the average squared errors.

**Subjective Questions - II**  
  
In the previous segment, you saw some common interview questions asked on linear regression. The questions in that segment were mostly related to the essence of linear regression and focused on general concepts related to linear regression. This section extensively covers the common interview questions asked related to the concepts learnt in multiple linear regression.

**Q1. What is Multicollinearity? How does it affect the linear regression? How can you deal with it?**

Multicollinearity occurs when some of the independent variables are highly correlated (positively or negatively) with each other. This multicollinearity causes a problem as it is against the basic assumption of linear regression. The presence of multicollinearity does not affect the predictive capability of the model. So, if you just want predictions, the presence of multicollinearity does not affect your output. However, if you want to draw some insights from the model and apply them in, let’s say, some business model, it may cause problems.

One of the major problems caused by multicollinearity is that it leads to incorrect interpretations and provides wrong insights. The coefficients of linear regression suggest the mean change in the target value if a feature is changed by one unit. So, if multicollinearity exists, this does not hold true as changing one feature will lead to changes in the correlated variable and consequent changes in the target variable. This leads to wrong insights and can produce hazardous results for a business.

A highly effective way of dealing with multicollinearity is the use of VIF (Variance Inflation Factor). Higher the value of VIF for a feature, more linearly correlated is that feature. Simply remove the feature with very high VIF value and re-train the model on the remaining dataset.

**Q2. How can you handle categorical variables present in the dataset?**

Many a time it might happen that your dataset has categorical variables that might be a potentially good predictor for the response variable. So handling them right is quite crucial.

One of the ways to handle categorical data with just two levels is to do a binary mapping of the variables wherein one of the levels will correspond to zero and the other to 1.

Another way of handling categorical variables with few levels is to perform a dummy encoding. The key idea behind dummy encoding is that for a variable with, say, 'N' levels, you create 'N-1' new indicator variables for each of these levels. So for a variable say, 'Relationship' with three levels namely, 'Single', 'In a relationship', and 'Married', you would create a dummy table like the following:

|  |  |  |  |
| --- | --- | --- | --- |
| Relationship Status | Single | In a relationship | Married |
| Single | 1 | 0 | 0 |
| In a relationship | 0 | 1 | 0 |
| Married | 0 | 0 | 1 |

But you can clearly see that there is no need of defining **three** different levels. If you drop a level, say 'Single', you would still be able to explain the three levels.

Let's drop the dummy variable 'Single' from the columns and see what the table looks like:

|  |  |  |
| --- | --- | --- |
| Relationship Status | In a relationship | Married |
| Single | 0 | 0 |
| In a relationship | 1 | 0 |
| Married | 0 | 1 |

If both the dummy variables namely 'In a relationship' and 'Married' are equal to zero, that means that the person is single. If 'In a relationship' is one and 'Married' is zero, that means that the person is in a relationship and finally, if 'In a relationship' is zero and 'Married' is 1, that means that the person is married.

Now, creating dummy variables might be useful when the number of levels in a categorical variable is small, but if a categorical variable has a hundred levels, it is clearly impossible to create 99 new variables. In such cases, grouping the variables might be useful. For example, for the variable “Cities in India”, you can use a geographical grouping, i.e.:

* Keep the 'n' largest cities, group the rest
* Geographical hierarchy
  + City < District < State < Zone
* Group cities with the similar value for the outcome variable
* Cluster cities with similar values for the predictor variables

Another way to deal with categorical variables is that you can perform a [One-hot encoding](https://hackernoon.com/what-is-one-hot-encoding-why-and-when-do-you-have-to-use-it-e3c6186d008f) which hasn't been covered in our syllabus.

**Q3. What is the major difference between R-squared and Adjusted R-squared/Why is it advised to use Adjusted R-squared in case of multiple linear regression?**

The major difference between R-squared and Adjusted R-squared is that R-squared doesn't penalise the model for having more number of variables. Thus, if you keep on adding variables to the model, the R-squared will always increase (or remain the same in the case when the value of correlation between that variable and the dependent variable is zero). Thus, R-squared assumes that any variable added to the model will increase the predictive power.

Adjusted R-squared on the other hand, penalises models based on the number of variables present in it. Its formula is given as:

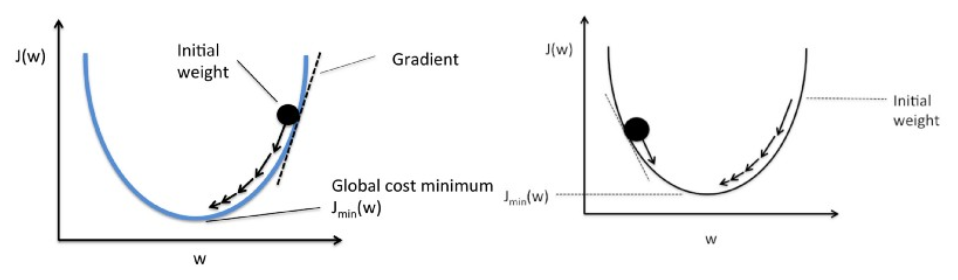
                                                             Adj.\spaceR2=1-(1-R2)(N-1)N-k-1

*where 'N' is the number of data points and 'k' is the number of features*

So if you add a variable and the Adjusted R-squared drops, you can be certain that that variable is insignificant to the model and shouldn't be used. So in the case of multiple linear regression, you should always look at the adjusted R-squared value in order to keep redundant variables out from your regression model.

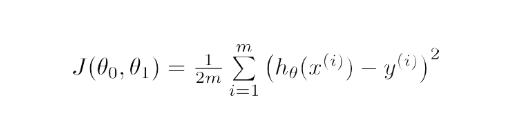
**Q4. Explain gradient descent with respect to linear regression.**

Gradient descent is an optimisation algorithm. In linear regression, it is used to optimise the cost function and find the values of the βs (estimators) corresponding to the optimised value of the cost function.

Gradient descent works like a ball rolling down a graph (ignoring the inertia). The ball moves along the direction of the greatest gradient and comes to rest at the flat surface (minima).  
  


Mathematically, the aim of gradient descent for linear regression is to find the solution of

ArgMin J(Θ*0*,Θ*1*), where J(Θ*0*,Θ*1*) is the cost function of the linear regression. It is given by —

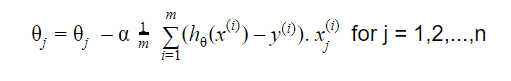


Here, *h* is the linear hypothesis model, h=Θ*0* + Θ1x, *y* is the true output, and *m* is the number of data points in the training set.

Gradient Descent starts with a random solution, and then based on the direction of the gradient, the solution is updated to the new value where the cost function has a lower value.

The update is:

Repeat until convergence



To read more about gradient descent, refer to the additional resources on linear regression here ([Link](https://learn.upgrad.com/course/3094/segment/26812/160582/493508/2548547)).

**Q4. What is VIF? How do you calculate it?**

Variance Inflation Factor (VIF) is used to check the presence of multicollinearity in a dataset. It is calculated as—

                                                                                    v|FF;11-Ri2

Here, VIFi  is the value of VIF for the ith variable,  Ri2  is the R2 value of the model when that variable is regressed against all the other independent variables.

If the value of VIF is high for a variable, it implies that the R2 value of the corresponding model is high, i.e. other independent variables are able to explain that variable. In simple terms, the variable is linearly dependent on some other variables.

**Q5. Explain the bias-variance trade-off.**

Bias refers to the difference between the values predicted by the model and the real values. It is an error. One of the goals of an ML algorithm is to have a low bias.

Variance refers to the sensitivity of the model to small fluctuations in the training dataset. Another goal of an ML algorithm is to have low variance.

For a dataset that is not exactly linear, it is not possible to have both bias and variance low at the same time. A straight line model will have low variance but high bias, whereas a high-degree polynomial will have low bias but high variance.

There is no escaping the relationship between bias and variance in machine learning.

1. Decreasing the bias increases the variance.
2. Decreasing the variance increases the bias.

So, there is a trade-off between the two; the ML specialist has to decide, based on the assigned problem, how much bias and variance can be tolerated. Based on this, the final model is built.