1. **Explain the linear regression algorithm in detail.**

Single Linear Regression -

This depicts the relationship between a dependent variable and one independent variable using a straight line. The straight line is plotted on the scatter plot of these two points.

The standard equation of the regression line is given by the following expression: Y = β 0 + β 1X

The strength of the linear regression model can be assessed using 2 metrics:

1. R2 or Coefficient of Determination
2. Residual Standard Error (RSE)

Multiple Linear Regression -

Multiple linear regression is a statistical technique to understand the relationship between one dependent variable and several independent variables. The objective of multiple regression is to find a linear equation that can best determine the value of dependent variable Y for different values independent variables in X.

The standard equation of the regression line is given by the following expression: Y = β 0 + β1 X + β2 X + β3 X + e

1. Model now fits a hyperplane instead of a line
2. Coefficients still obtained by minimizing sum of squared error (Least squares criterion)
3. Zero mean, independent, normally distributed error terms that have constant

- Need to take care of Overfitting, Multicollinear and Feature section using RFE
- Also, Variation Inflation Factor for collinearity, Minmax Scaling of feature for standardizing variables between 0 to 1, Substituting Dummy variables for categorical variables
- Measures Include Adjusted R2 and AIC

2. **What are the assumptions of linear regression regarding residuals?**
   - Normality assumption: It is assumed that the error terms, ε(i), are normally distributed.
   - 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.
   - 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.
   - Independent error assumption: It is assumed that the residual terms are independent of each other, i.e. their pair-wise covariance is zero.

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

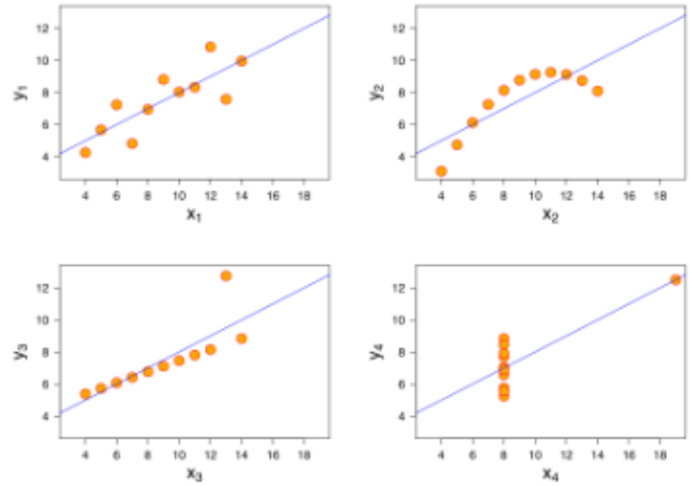
The **correlation coefficient** is a statistical measure that calculates the strength of the relationship between the relative movements of two variables. The values range between -1.0 and 1.0. A calculated number greater than 1.0 or less than -1.0 means that there was an error in the correlation measurement. A correlation of -1.0 shows a perfect -ve Correlation, while a correlation of 1.0 shows a perfect +ve correlation. A correlation of 0.0 shows no relationship between the movement of the two variables

The **coefficient of determination** is a measure used in statistical analysis that assesses how well a model explains and predicts future outcomes. It is indicative of the level of explained variability in the data set. The coefficient of determination, also commonly known as R-Squared is used as a guideline to measure the accuracy of the model.

4. Explain the Anscombe's quartet in detail.

The below 3 short comings of linear regression can be explained by Anscombe's quartet

- It is sensitive to outliers
- It models the linear relationships only
- A few assumptions are required to make the inference



-

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.

5. What is Pearson's R?

It is a measure of the linear correlation between two variables X and Y.
Pearson's correlation coefficient is the covariance of the two variables divided by the product of their standard deviations. The form of the definition involves a "product moment", that is, the

mean of the product of the mean-adjusted random variables; hence the modifier product-moment in the name.

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

Feature Scaling is used when there are multiple independent variables in a model, a lot of them might be on very different scales which will lead a model with very weird coefficients that might be difficult to interpret. So we need to scale features because of two reasons:
1. Ease of interpretation 2. Faster convergence for gradient descent methods
there are two ways

- **Standardizing**: The variables are scaled in such a way that their mean is zero and standard deviation is one.
- **MinMax Scaling**: The variables are scaled in such a way that all the values lie between zero and one using the maximum and the minimum values in the data.

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

7. **You might have observed that sometimes the value of VIF is infinite. Why does this happen?**
When there is very high collinearity of a variable with all other variables (means perfect correlation)

8. **What is the Gauss-Markov theorem?**
The Gauss Markov theorem tells us that if a certain assumption are met, the ordinary least square estimate for regression coefficients gives you the best linear unbiased estimate possible.
Gauss Markov Assumptions
There are five Gauss Markov assumptions (also called conditions):
1. Linearity: the parameters we are estimating using the OLS method must be themselves linear.
2. Random: our data must have been randomly samples from the population
3. Non-Collinearity: the regressors being calculated aren't perfectly correlated with each other.
4. Exogeneity: the regressors aren't correlated with the error term
5. Homoscedasticity: no matter what the values of our regressors might be, the error of the variance is constant

9. **Explain the gradient descent algorithm in detail.**
Gradient Descent is an a first-order iterative optimization algorithm used for minimizing the cost function in various machine learning algorithms. It is basically used for updating the parameters of the learning model.

Types of gradient Descent:

Batch Gradient Descent: This is a type of gradient descent which processes all the training examples for each iteration of gradient descent. But if the number of training examples is large, then batch gradient descent is computationally very expensive. Hence if the number of training examples is large, then batch gradient descent is not preferred.

Stochastic Gradient Descent: This is a type of gradient descent which processes 1 training example per iteration. Hence, the parameters are being updated even after one iteration in which only a single example has been processed. Hence this is quite faster than batch gradient descent. But again, when the number of training examples is large, even then it processes only one example which can be additional overhead for the system as the number of iterations will be quite large.
Mini Batch gradient descent: This is a type of gradient descent which works faster than both batch gradient descent and stochastic gradient descent. Here b examples where $b<m$ are processed per iteration. So even if the number of training examples is large, it is processed in batches of b training examples in one go. Thus, it works for larger training examples and that too with lesser number of iterations.

Variables used:
Let m be the number of training examples.
Let n be the number of features.

Note: if $b == m$, then mini batch gradient descent will behave similarly to batch gradient descent.

Algorithm for batch gradient descent:
Let $h\theta(x)$ be the hypothesis for linear regression. Then, the cost function is given by:
Let Σ represents the sum of all training examples from $i=1$ to m.

$$Jtrain(\theta) = (1/2m)\ \Sigma(\ h\theta(x(i)) - y(i))2$$

```
Repeat {
 $\theta j = \theta j - (\text{learning rate}/m) * \Sigma(\ h\theta(x(i)) - y(i))xj(i)$
    For every $j = 0 \ldots n$
}
```

Where $xj(i)$ Represents the jth feature of the ith training example. So if m is very large, then the derivative term fails to converge at the global minimum.

Algorithm for stochastic gradient descent:
1) Randomly shuffle the data set so that the parameters can be trained evenly for each type of data.
2) As mentioned above, it takes into consideration one example per iteration.

Hence,
Let $(x(i),y(i))$ be the training example
$Cost(\theta, (x(i),y(i))) = (1/2)\ \Sigma(\ h\theta(x(i)) - y(i))2$

$Jtrain(\theta) = (1/m)\ \Sigma\ Cost(\theta, (x(i),y(i)))$

```
Repeat {

For i=1 to m{

     $\theta j = \theta j - (\text{learning rate}) * \Sigma( h\theta(x(i)) - y(i))xj(i)$
    For every j =0 …n

         }
}
```

Algorithm for mini batch gradient descent:
Say b be the no of examples in one batch, where b < m. Assume b = 10, m = 100; Note: However we can adjust the batch size. It is generally kept as power of 2. The reason behind it is because some hardware such as GPUs achieve better run time with common batch sizes such as power of 2.

```
Repeat {
 For i=1,11, 21,…..,91

  Let $\Sigma$ be the summation from i to i+9 represented by k.

  $\theta j = \theta j - (\text{learning rate/size of (b)}) * \Sigma( h\theta(x(k)) - y(k))xj(k)$
    For every j =0 …n

}
```

Convergence trends in different variants of Gradient Descents:
In case of **Batch Gradient Descent**, the algorithm follows a straight path towards the minimum. If the cost function is convex, then it converges to a global minimum and if the cost function is not convex, then it converges to a local minimum. Here the learning rate is typically held constant.
In case of **stochastic gradient Descent and mini-batch gradient descent**, the algorithm does not converge but keeps on fluctuating around the global minimum. Therefore, in order to make it converge, we have to slowly change the learning rate. However, the convergence of Stochastic gradient descent is much noisier as in one iteration, it processes only one training example.

10. **What is a Q-Q plot? Explain the use and importance of a Q-Q plot in linear regression.**
    A Q–Q (quantile-quantile) plot is a probability plot, which is a graphical method for comparing two probability distributions by plotting their quantiles against each other.
    First, the set of intervals for the quantiles is chosen. A point (x, y) on the plot corresponds to one of the quantiles of the second distribution (y-coordinate) plotted against the same quantile of the first distribution (x-coordinate).

Thus, the line is a parametric curve with the parameter which is the number of the interval for the quantile.

If the two distributions being compared are similar, the points in the Q–Q plot will approximately lie on the line y = x.
If the distributions are linearly related, the points in the Q–Q plot will approximately lie on a line, but not necessarily on the line y = x.
Q–Q plots can also be used as a graphical means of estimating parameters in a location-scale family of distributions.

Importance of Q-Q Plots

- A Q–Q plot is used to compare the shapes of distributions, providing a graphical view of how properties such as location, scale, and skewness are similar or different in the two distributions.
- Q–Q plots can be used to compare collections of data, or theoretical distributions.
- The use of Q–Q plots to compare two samples of data can be viewed as a non-parametric approach to comparing their underlying distributions.
- This can provide an assessment of "goodness of fit" that is graphical, rather than reducing to a numerical summary.
- Q–Q plots are also used to compare two theoretical distributions to each other. Since Q–Q plots compare distributions, there is no need for the values to be observed as pairs, as in a scatter plot, or even for the numbers of values in the two groups being compared to be equal.