R2

Coefficient of Determination

(Goodness-of-fit measure)

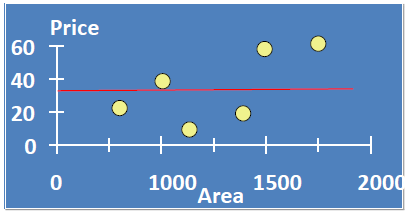
* It measures how well our model fits the data
* It is square of the correlation between observed values and predicted values
* The value of R2 can range between 0 and 1, and the higher its value the more accurate the regression model is

Base line model

* Suppose we build a model which contains only intercept
* Then ( Mean of the dependent Variable)
* Predicted value for every record used in training set is 

SST  (Sum of squares of errors in base line model)

SST: Sum of Square of Total / Total Sum of Square



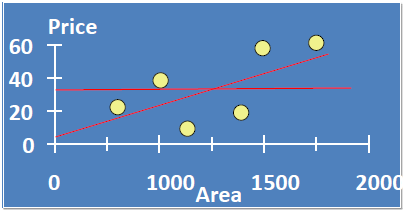
Regression model

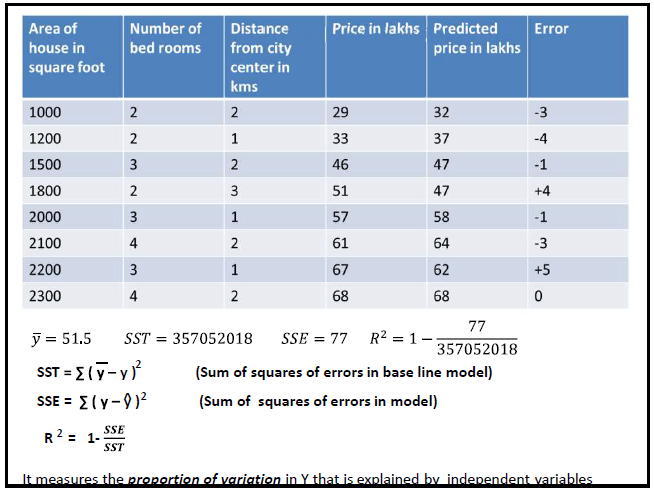
* Suppose we build a model which contains both intercept and angle
* SSE =  (Sum of Squares of Errors in model)





* It measures the proportion of variation in Y that is explained by independent variables/model





* SSR is the "regression sum of squares" and quantifies how far the estimated sloped regression line,, is from the horizontal ( mean or ).







SST = SSR + SSE

SSR=SST-SSE

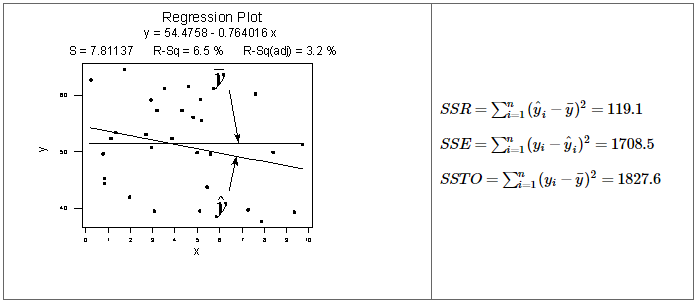




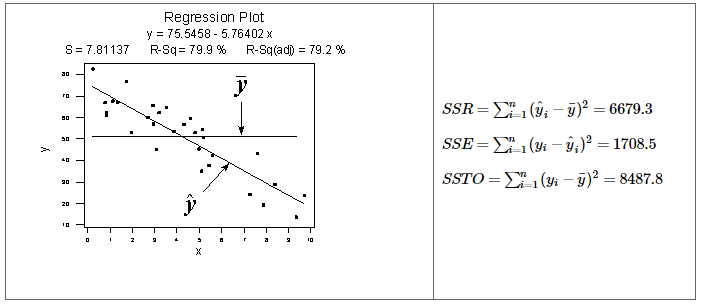
 

**Discussion with examples**

Here's a plot illustrating a very weak relationship between y and x. There are two lines on the plot, a horizontal line placed at the average response, y¯, and a shallow-sloped estimated regression line, y^. Note that the slope of the estimated regression line is not very steep, suggesting that as the predictor x increases, there is not much of a change in the average response y. Also, note that the data points do not "hug" the estimated regression line:



Contrast the above example with the following one in which the plot illustrates a fairly convincing relationship between y and x. The slope of the estimated regression line is much steeper, suggesting that as the predictor x increases, there is a fairly substantial change (decrease) in the response y. And, here, the data points do "hug" the estimated regression line:

****

**Basic Characteristics (R2)**

* Since *r*2 is a proportion, it is always a number between 0 and 1.
* If *r*2 = 1, all of the data points fall perfectly on the regression line.The predictor *x* accounts for *all* of the variation in *y*!

(100% indicates that the model explains all the variability of the response data around its mean.)

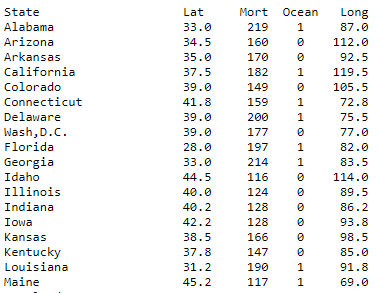
* If *r*2 = 0, the estimated regression line is perfectly horizontal. The predictor *x* accounts for *none* of the variation in *y*!

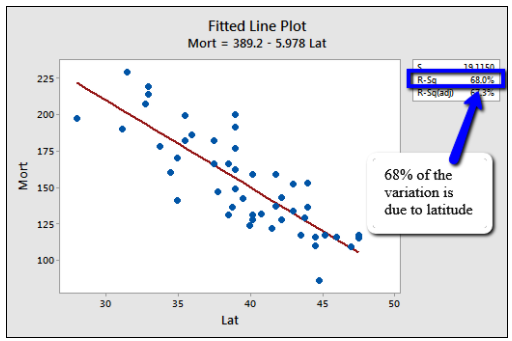
(0% indicates that the model explains none of the variability of the response data around its mean)

We've learned the interpretation for the two easy cases — when r2 = 0 or r2 = 1 — but, how do we interpret r2 when it is some number between 0 and 1, like 0.23 or 0.57, say?

"r2 ×100 percent of the variation in y is 'explained by' the variation in predictor x/model."

Let's revisit the skin cancer mortality example

****



We can say that that 68% of the variation in skin cancer mortality is 'due to' or is 'explained by' latitude.

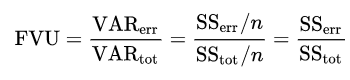
**Key Limitations of R-squared**

R-squared *cannot* determine whether the coefficient estimates and predictions are biased, which is why you must assess the residual plots.

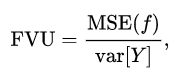
R-squared does not indicate whether a regression model is adequate. You can have a low R-squared value for a good model, or a high R-squared value for a model that does not fit the data!

**Fraction of Variance Unexplained (FVU)**

In [statistics](https://en.wikipedia.org/wiki/Statistics), the **fraction of variance unexplained (FVU)** in the context of a [regression task](https://en.wikipedia.org/wiki/Regression_analysis) is the fraction of variance of the [regressand](https://en.wikipedia.org/wiki/Regressand)(dependent variable) *Y* which cannot be explained, i.e., which is not correctly predicted, by the [explanatory variables](https://en.wikipedia.org/wiki/Explanatory_variable) *X*.

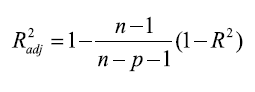


**R2 = 1- FAV**

****

adjusted R2

* Note that R2 is only a descriptive measure to give a quick assessment of the model.
* Adding explanatory variables to the model always increases R2 . Hence in practice, it is more usual to look at the adjusted R2
* The adjusted R2 is calculated as



|  |  |  |
| --- | --- | --- |
| p | r2 | ad-r2 |
| 1 | 0.2 | 0.111111 |
| 2 | 0.23 | 0.0375 |
| 3 | 0.5 | 0.285714 |
| 4 | 0.55 | 0.25 |
| 5 | 0.8 | 0.6 |
| 6 | 0.92 | 0.8 |

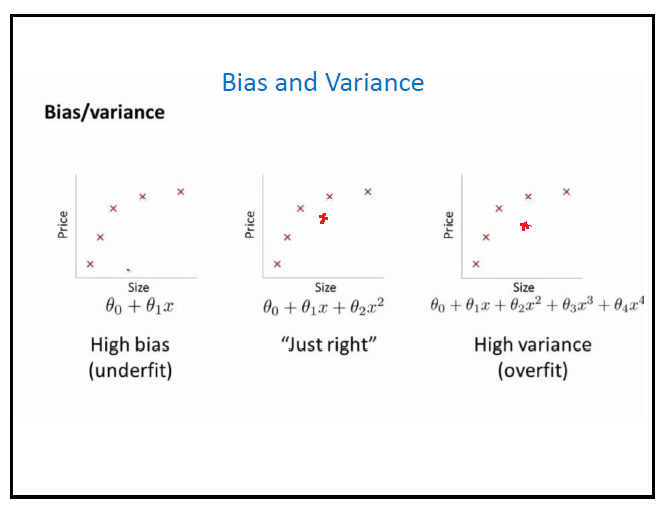
* ‘n’ is the number of points in your data sample.
* ‘p’ is the number of independent regressors (predictor), i.e. the number of variables in your model, excluding the constant.

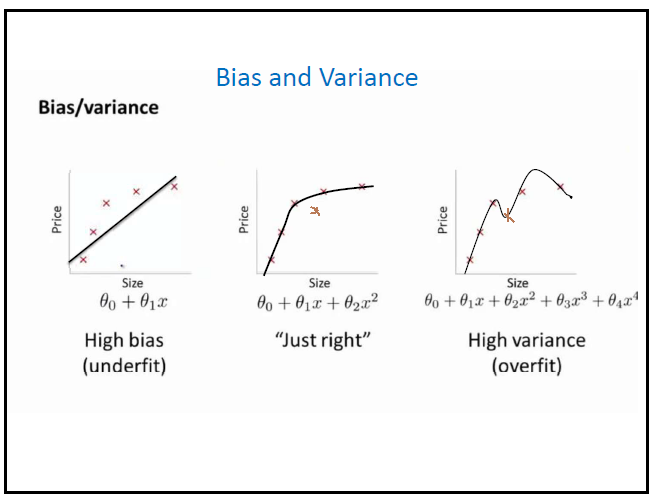
**Main differences** between R2 and the adjusted R2: R2 assumes that every single variable explains the *variation in the*[dependent variable](http://www.statisticshowto.com/dependent-variable-definition/). The adjusted R2 tells you the percentage of *variation explained by only the*[independent variables](http://www.statisticshowto.com/independent-variable-definition/)*that* actually *affect the dependent variable*. adjusted R2 can be negative. Adjusted R2 is always less than or equal to R2

Problems with R2 that are corrected with an adjusted R2

1. R2 increases with every predictor added to a model. As R2 always increases and never decreases, it can appear to be a better fit with the more terms you add to the model. This can be completely misleading.
2. Similarly, if your model has too many terms and too many high-order polynomials you can run into the problem of over-fitting the data. When you over-fit data, a misleadingly high R2 value can lead to misleading projections.

Correlation coefficient(r) is the measure of strength and direction of a linear relationship between to random variables x and y and it varies between (-1 to 1). Whereas (<|0.3|)- weak, (|0.3|<->|o.6|)- moderate and (>|0.6|)- strong. On the other hand regression is a statistical model or better to say predictive model. It tells how much variation is expected on dependent variable against variation on independent variables. It is always positive.

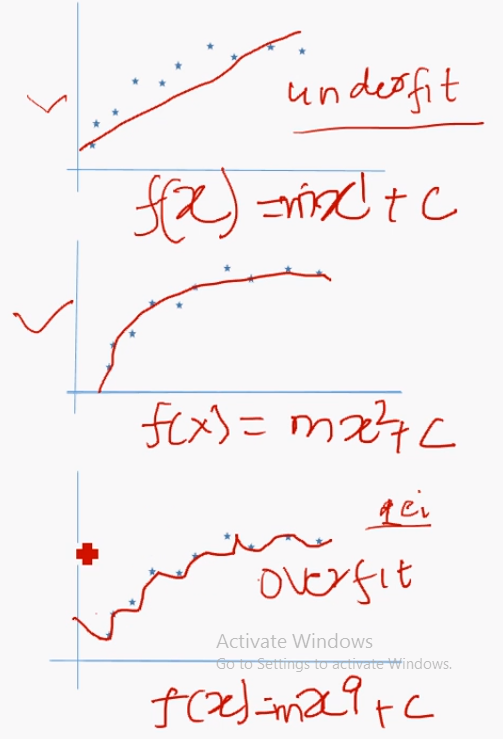


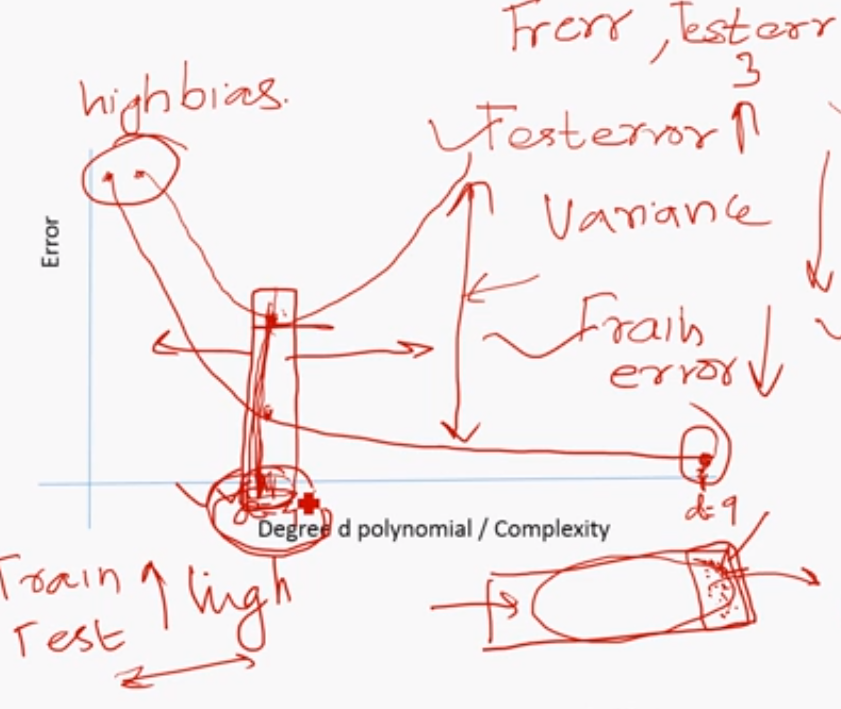


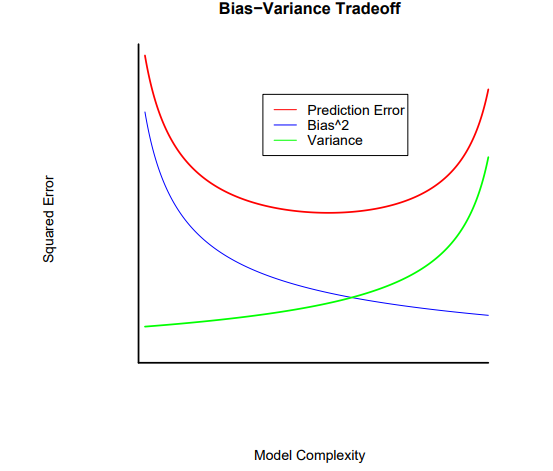
* Bias Problem : R2 is very low (Both Training and Testing errors are high)
* Variance Problem: Training error is low and testing error is high

**Solutions:**

* Get more training examples- Fix high variance
* Try smaller set of feature- Fix high variance
* Try getting additional features- Fix high bias
* Try additional polynomial feature- Fix high bias
* Ridge regression- Fix high variance
* Take number of training examples should be at least 10 time the number of variables- Fix high variance







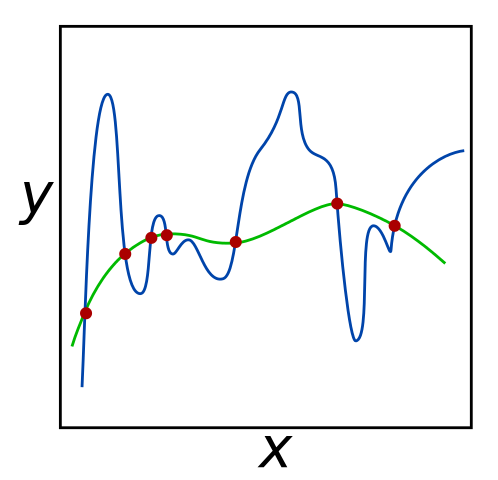
**Regularization** is a *technique*used in an attempt to solve the **overfitting.** Regularization refers to the method of preventing overfitting, by explicitly controlling the model complexity.

It leads to smoothening of the regression line and thus prevents overfitting. It does so by penalizing the bent of the regression line that tries to closely match the noisy data points.

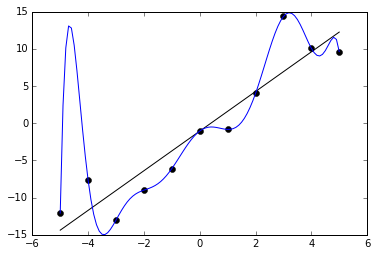
1. **Ridge Regression:**
   * Performs L2 regularization, i.e. adds penalty equivalent to **square of the magnitude** of coefficients
   * Minimization objective = LS Obj + α \* (sum of square of coefficients)
2. **Lasso Regression:**
   * Performs L1 regularization, i.e. adds penalty equivalent to **absolute value of the magnitude** of coefficients
   * Minimization objective = LS Obj + α \* (sum of absolute value of coefficients)

Note that here ‘LS Obj’ refers to ‘least squares objective’, i.e. the linear regression objective without regularization.

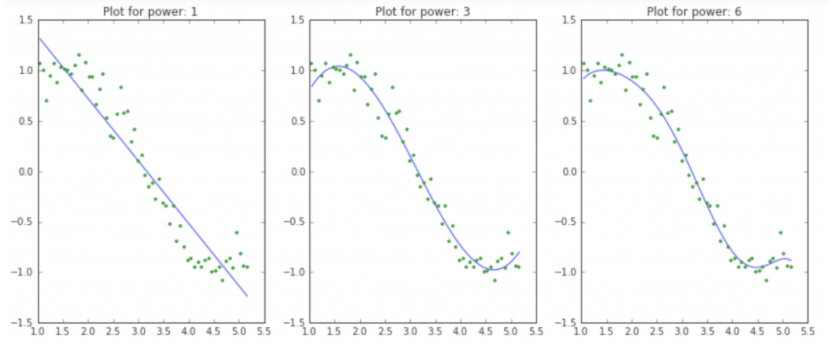
**Ridge Regression**

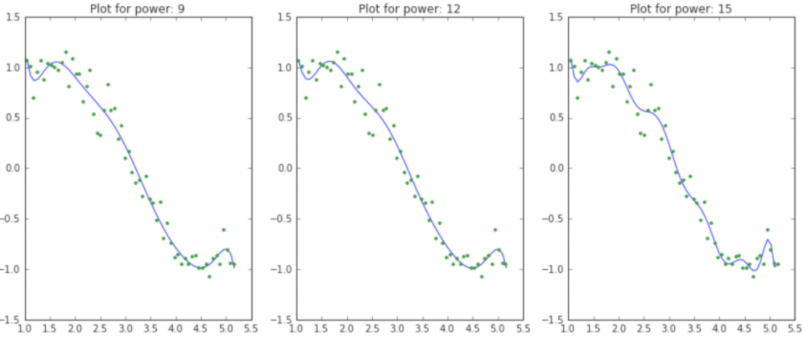


*For the given set of red input points, both the green and blue lines minimize error to 0. However, the green line may be more successful at predicting the coordinates of unknown data points, since it seems to*[*generalize*](https://brilliant.org/wiki/generalization/?wiki_title=generalize)*the data better. Ridge regression can be used to prefer the green line over the blue line by penalizing large*[*coefficients*](https://brilliant.org/wiki/coefficients/?wiki_title=coefficients)*.*



The blue curve minimizes the error of the data points. However, it does not generalize well (it overfits the data). Introducing a  term can result in a curve like the black one, which does not minimize errors, but fits the data well,





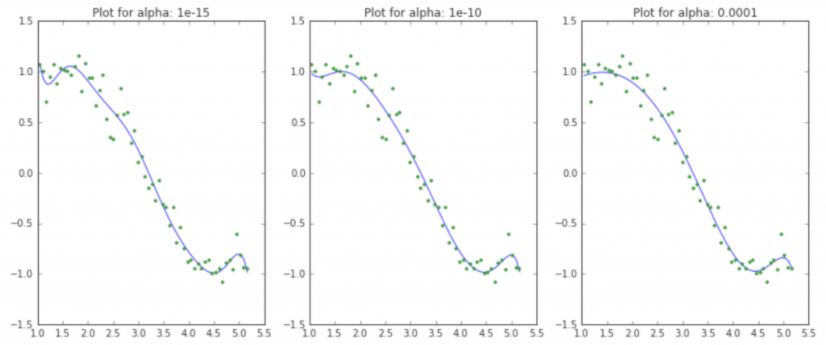
**size of coefficients increase exponentially with increase in model complexity**. I hope this gives some intuition into why putting a constraint on the magnitude of coefficients can be a good idea to reduce model complexity.

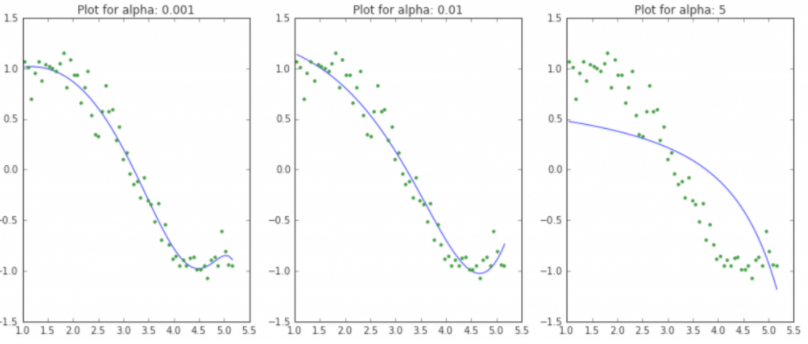
What does a large coefficient signify? It means that we’re putting a lot of emphasis on that feature, i.e. the particular feature is a good predictor for the outcome. When it becomes too large, the algorithm starts modelling intricate relations to estimate the output and ends up overfitting to the particular training data.

Objective = RSS + α \* (sum of square of coefficients)

Here, α (alpha) is the parameter which balances the amount of emphasis given to minimizing RSS vs minimizing sum of square of coefficients. α can take various values:

1. **α = 0:**
   * The objective becomes same as simple linear regression.
   * We’ll get the same coefficients as simple linear regression.
2. **α = ∞:**
   * The coefficients will be zero. Why? Because of infinite weightage on square of coefficients, anything less than zero will make the objective infinite.
3. **0 < α < ∞:**
   * The magnitude of α will decide the weightage given to different parts of objective.





**As the value of alpha increases, the model complexity reduces**. Though higher values of alpha reduce overfitting, significantly high values can cause underfitting as well (eg. alpha = 5). Thus alpha should be chosen wisely. A widely accept technique is cross-validation, i.e. the value of alpha is iterated over a range of values and the one giving higher cross-validation score is chosen.

This straight away gives us the following inferences (Ridge Regression):

1. The RSS (SSE) increases with increase in alpha, this model complexity reduces
2. High alpha values can lead to significant underfitting.
3. Though the coefficients are **very very small**, they are **NOT zero**.

**k-fold cross-validation**

In k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data.

The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data.

The k results from the folds can then be averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once. 10-fold cross-validation is commonly used.

## Lasso Regression

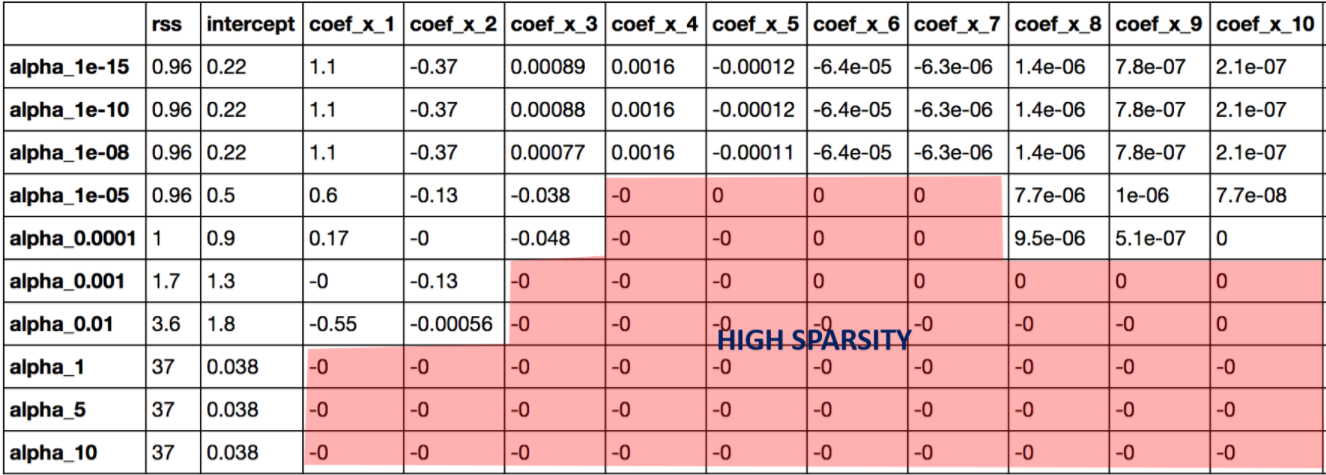
## LASSO stands for Least Absolute Shrinkage and Selection Operator. Lasso regression performs **L1 regularization**, i.e. it adds a factor of sum of absolute value of coefficients in the optimization objective. Thus, lasso regression optimizes the following:

Observation

1. For the same values of alpha, the coefficients of lasso regression are much smaller as compared to that of ridge regression.
2. For the same alpha, lasso has higher RSS (poorer fit) as compared to ridge regression.
3. Many of the coefficients are zero even for very small values of alpha

Discussion:

**Even for a small value of alpha, a significant number of coefficients are zero**. This phenomenon of most of the coefficients being zero is called ‘**sparsity**‘. Although lasso performs feature selection.



### Key Difference

* **Ridge:** It includes all (**α!=0**) of the features in the model. Thus, the major advantage of ridge regression is coefficient shrinkage and reducing model complexity.
* **Lasso:** Along with shrinking coefficients, lasso performs feature selection as well. (Remember the ‘*selection*‘ in the lasso full-form?), some of the coefficients become exactly zero, which is equivalent to the particular feature being excluded from the model.

Traditionally, techniques like **stepwise regression** were used to perform feature selection and make parsimonious models. But with advancements in Machine Learning, ridge and lasso regression provide very good alternatives as they give much**better output**, require **fewer tuning parameters** and can be **automated** to a large extend.

Typical Use Cases

* **Ridge:** It is majorly used to *prevent overfitting*. Since it includes all the features, it is not very useful in case of high #features, as it will pose computational challenges.
* **Lasso:** Since it provides *sparse solutions*, it is generally the model of choice (or some variant of this concept) for modelling cases where the #features are more. In such a case, getting a sparse solution is great computational advantage as the features with zero coefficients can simply be ignored.

Presence of Highly Correlated Features

* **Ridge:** It generally works well even in presence of highly correlated features as it will include all of them in the model but the coefficients will be distributed among them depending on the correlation.
* **Lasso:** It arbitrarily selects any one feature among the highly correlated ones and reduced the coefficients of the rest to zero. Also, the chosen variable changes randomly with change in model parameters. This generally doesn’t work that well as compared to ridge regression.