1. **Does OLS necessarily produce the lowest mean square error (MSE)?**

**No. The mean squared error is made up of bias, variance and error:**

MSE = Variance + Bias2 + Irreducible error

Put I the diagram

But OLS minimizes bias first and then variance. There are times when we can give up a little bias to greatly reduce variance.

1. When would it happen that we would increase bias to lower variance?
   1. Imagine for a moment you have to bet on the accuracy of two great archers. One is very drunk – on average he hits the bullseye but soemtimes misses the target entirely. The other has astigmatism, on average they land in the same spot near the bullseye but sometimes hit the bullseye and are nevey too far off.
   2. Skim vs memorize
2. When might we prefer the blind archer to the drunk one?
   1. The drunk archer might be
      1. Overfit
      2. Multicollinearity
3. How can we move from drunk to blind?
   1. Regularization – restrict the size of the coefficients. They add a penalty to how big your beta vector can get, each in a different way.We can do this with:

Lasso Lasso puts a penalty on the l1-norm of your Beta vector. The l1-norm of a vector is the sum of the absolute values in that vector.

**Lasso and ridge – regularization**

# How do these models work?

Lasso and Ridge are both Linear Regression models but with a penalty (also called a regularization). They add a penalty to how big your beta vector can get, each in a different way.

## Lasso regression

Lasso puts a penalty on the l1-norm of your Beta vector. The l1-norm of a vector is the sum of the absolute values in that vector.

This makes Lasso zero out some coefficients in your Beta vector.

To summarise it simply, using Lasso is like saying: “Try to achieve the best performance possible but if you find that some coefficients are useless, drop them”.

Ridge puts a penalty on the l2-norm of your Beta vector. The 2-norm of a vector is the square root of the sum of the squared values in your vector.

This makes Ridge prevent the coefficients of your Beta vector to reach extreme values (which often happens when overfitting).

To summarise it simply, using Ridge is like saying: “Try to achieve the best performance possible but none of the coefficients should have extreme values”.

Both of these models have a regularisation parameter called lambda, which controls how large the penalty is. At λ=0, both Lasso and Ridge become Linear Regression models (we simply do not put any penalties). By increasing lambda, we increase the constraint on the size of the beta vector. This is where each model optimises differently and tries to find the best set of coefficients given its own constraints.

Therefore, you can see the link with what we discussed earlier. We “told” Lasso to find the best model given the constraint on how much weight could be put on each coefficient (i.e. the “budget”) and it “decided” to put a large amount of that “budget” on the number of rooms to figure out the price of the properties.

Lasso is good when you have a few features with high predicting power while the others are useless: it will zero out the useless ones and keep only a subset of the variables.

Ridge is good when the predicting power of your dataset is spread out over the different features: it will not zero out features that could be helpful when making predictions but will simply reduce the weight of most variables in the model.

In practice, this is often hard to determine. Thus, the best way is to simply do what I coded above and see what is the best MSE you can get on the test set using different values of lambda.

One problem that often occurs in practice with multiple linear regression is [multicollinearity](https://www.statology.org/multicollinearity-regression/) – when two or more predictor variables are highly correlated to each other, such that they do not provide unique or independent information in the regression model.

This can cause the coefficient estimates of the model to be unreliable and have high variance. That is, when the model is applied to a new set of data it hasn’t seen before, it’s likely to perform poorly.

MSE = Var(f̂(x0)) + [Bias(f̂(x0))]2 + Var(ε) ([When to Use Ridge & Lasso Regression - Statology](https://www.statology.org/when-to-use-ridge-lasso-regression/))

MSE = Variance + Bias2 + Irreducible error

Ordinary Least Squares (OLS) regression is known to give unbiased results with low variance as compared to non linear models.**Ridge** (OLS with L2 penalty) and **Lasso** (OLS with L1 penalty) give biased results with a much lower variance as compared to OLS. The degree of penalization is controlled by the regularization coefficient, λ.

The basic idea of both ridge and lasso regression is to introduce a little bias so that the variance can be substantially reduced, which leads to a lower overall MSE.

This means the model fit by ridge and lasso regression can potentially produce smaller test errors than the model fit by least squares regression.

The **drawback** of ridge and lasso regression is that it becomes difficult to interpret the coefficients in the final model since they get shrunk towards zero.

Thus, ridge and lasso regression should be used when you’re interested in optimizing for predictive ability rather than inference.

In cases where only a small number of predictor variables are significant, **lasso regression** tends to perform better because it’s able to shrink insignificant variables completely to zero and remove them from the model.

However, when many predictor variables are significant in the model and their coefficients are roughly equal then **ridge regression** tends to perform better because it keeps all of the predictors in the model.

To determine which model is better at making predictions, we typically perform [k-fold cross-validation](https://www.statology.org/k-fold-cross-validation/) and choose whichever model produces the lowest test mean squared error.

**Errors due to Bias**

An error due to ***Bias*** is the ***distance between the predictions*** of a model and the ***true values***. In this type of error, the model pays little attention to training data and ***oversimplifies***the model and doesn't learn the patterns. The model **learns the wrong relations** by **not taking in account all the features**

**Skimming -underfit**

## ****Errors due to Variance****

Variability of model prediction for a given data point or a value that **tells us the spread of our data**. In this type of error, the model pays a l**ot of attention in training data**, to the point to memorize it instead of learning from it. A model with a high error of variance is not flexible to generalize on the data which it hasn’t seen before.

Memorizing=overfit