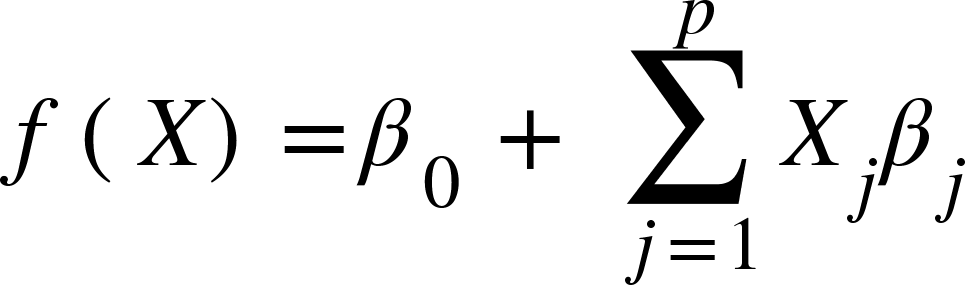
Regression Analysis

Linear regression models can sometimes outperform non-linear models, especially in situations with a small number of training cases, low signal-to-noise ratio or sparse data.

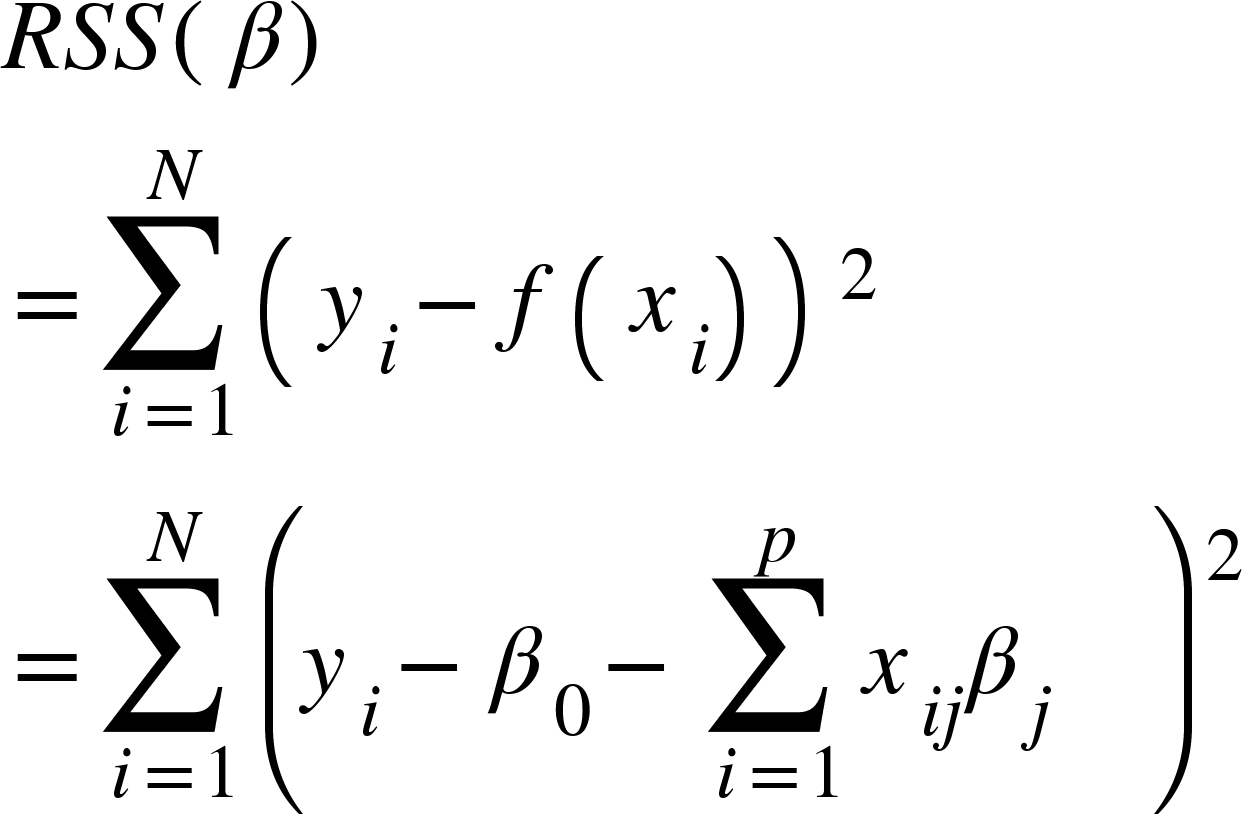
**Linear Regression and Least Squares**

Considering an input vector XT = {X1, X2,....Xp} and we want to predict output Y. The linear regression model as the form



It assumes either that the regression function E(Y|X) is linear, or the linear model is a reasonable approximation.

We pick the coefficients beta such that we minimize the residual sum of squares.



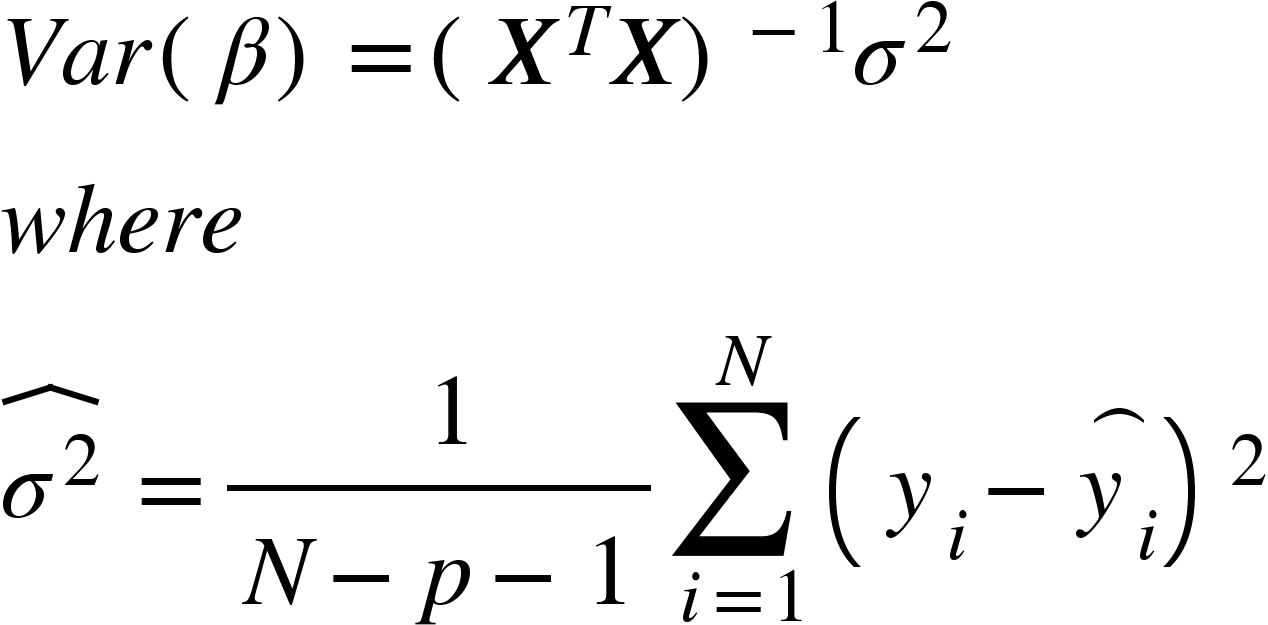
To minimize this,

<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>R</mi><mi>S</mi><mi>S</mi><mfenced><mi>&#x3B2;</mi></mfenced><mo>=</mo><msup><mfenced><mrow><mi mathvariant="bold-italic">y</mi><mo mathvariant="bold">&#xA0;</mo><mo mathvariant="bold">-</mo><mo mathvariant="bold">&#xA0;</mo><mi mathvariant="bold-italic">X</mi><mi>&#x3B2;</mi></mrow></mfenced><mi>T</mi></msup><mfenced><mrow><mi mathvariant="bold-italic">y</mi><mo mathvariant="bold">&#xA0;</mo><mo mathvariant="bold">-</mo><mo mathvariant="bold">&#xA0;</mo><mi mathvariant="bold-italic">X</mi><mi>&#x3B2;</mi></mrow></mfenced><mspace linebreak="newline"/><mfrac><mrow><mo>&#x2202;</mo><mi>R</mi><mi>S</mi><mi>S</mi><mfenced><mi>&#x3B2;</mi></mfenced></mrow><mrow><mo>&#x2202;</mo><mi>&#x3B2;</mi></mrow></mfrac><mo>=</mo><mo>-</mo><mn>2</mn><msup><mi mathvariant="bold-italic">X</mi><mi mathvariant="bold">T</mi></msup><mi mathvariant="bold-italic">X</mi><mspace linebreak="newline"/><mi>A</mi><mi>s</mi><mi>s</mi><mi>u</mi><mi>m</mi><mi>i</mi><mi>n</mi><mi>g</mi><mo>&#xA0;</mo><mi>X</mi><mo>&#xA0;</mo><mi>h</mi><mi>a</mi><mi>s</mi><mo>&#xA0;</mo><mi>f</mi><mi>u</mi><mi>l</mi><mi>l</mi><mo>&#xA0;</mo><mi>c</mi><mi>o</mi><mi>l</mi><mi>u</mi><mi>m</mi><mi>n</mi><mo>&#xA0;</mo><mi>r</mi><mi>a</mi><mi>n</mi><mi>k</mi><mspace linebreak="newline"/><msup><mi mathvariant="bold-italic">X</mi><mi>T</mi></msup><mstyle mathvariant="bold"><mrow><mo>(</mo><mi>y</mi><mo>-</mo><mi>X</mi><mi mathvariant="normal">&#x3B2;</mi><mo>)</mo></mrow></mstyle><mo mathvariant="bold">&#xA0;</mo><mo mathvariant="bold">=</mo><mo mathvariant="bold">&#xA0;</mo><mn mathvariant="bold">0</mn><mspace linebreak="newline"/><mover><mi mathvariant="bold">&#x3B2;</mi><mo mathvariant="bold">^</mo></mover><mo mathvariant="bold">&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><msup><mfenced><mrow><msup><mi mathvariant="bold-italic">X</mi><mi mathvariant="bold">T</mi></msup><mi mathvariant="bold-italic">X</mi></mrow></mfenced><mrow><mo>-</mo><mn>1</mn></mrow></msup><msup><mi mathvariant="bold-italic">X</mi><mi>T</mi></msup><mi mathvariant="bold-italic">y</mi></math>

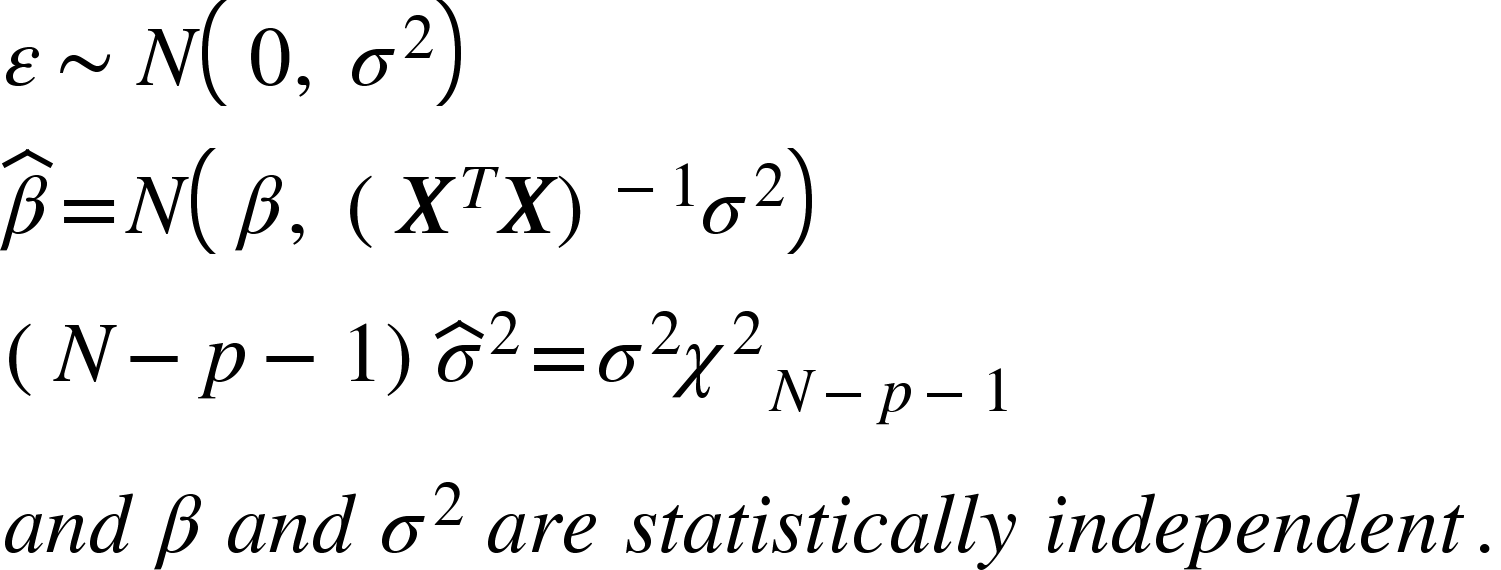
And then we can substitute it in y = XB to get the regression line. The y we get is said to be the orthogonal projection of the actual values onto the subspace.

If X does not have full rank, then the XTX is singular, and the coefficients are not uniquely defined. However, the fit value is still an orthogonal projection of y onto the column space of X.

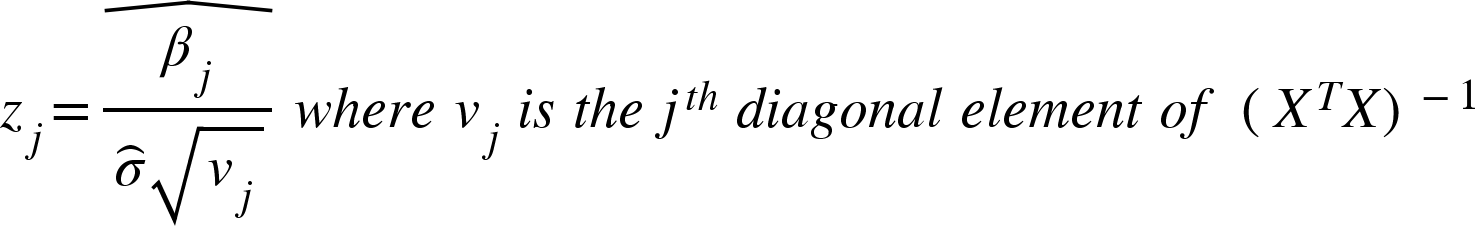
To pin down the sampling properties of beta, we now assume that the observations yi are uncorrelated and have constant variance, and the xi are fixed.



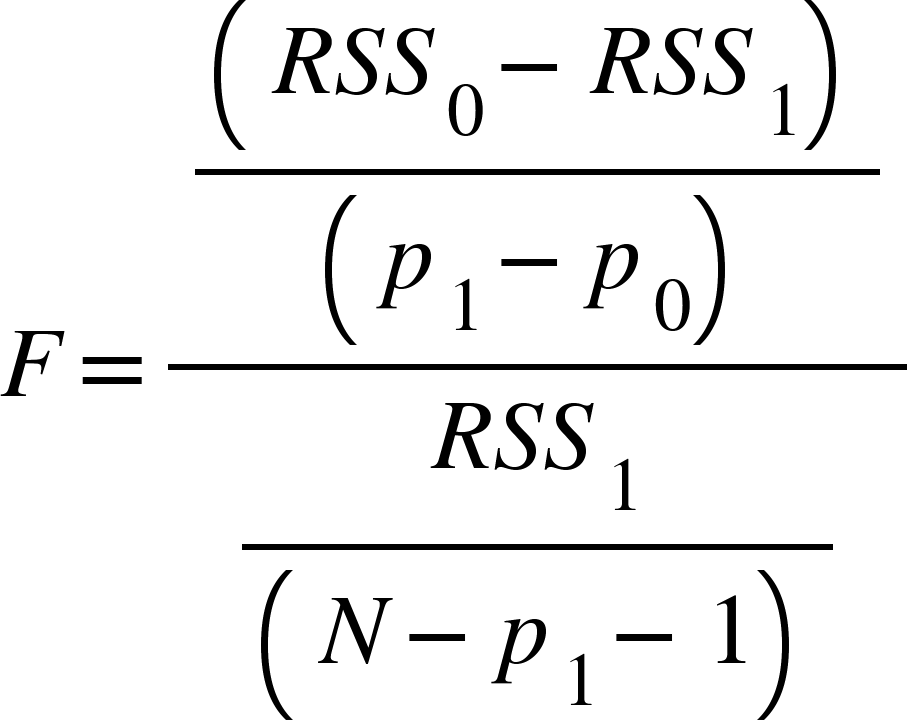
Assuming the correct model is taken, we also assume the deviations around Y follow a Gaussian distribution with variance given above.



To test the hypothesis of a particular coefficient is 0, we take the Z score, and zj is distributed as a tN-p-1 distribution, where



The significance of a group of coefficients is taken simultaneously. This is done by the F statistic as



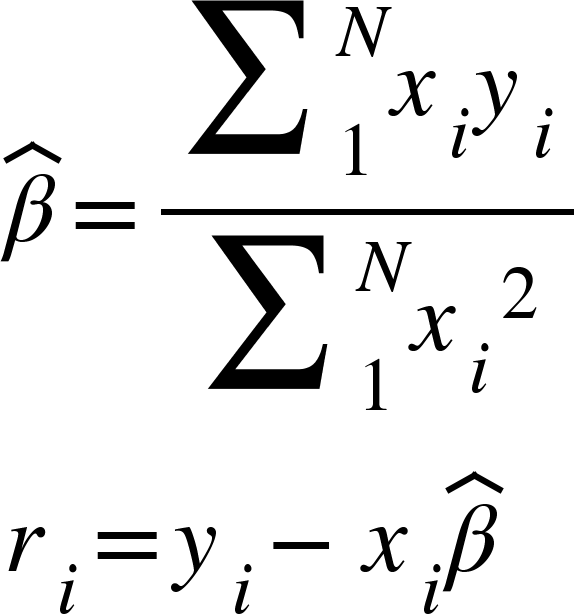
Where RSS1 is the RSS for the bigger model with p1+1 parameters and RSS0 for the smaller model with p0+1 parameters, having p1-p0 parameters constrained to 0. This hence measures the impact of additional parameters in the model.

**Multiple Regression from Simple Univariate Regression**

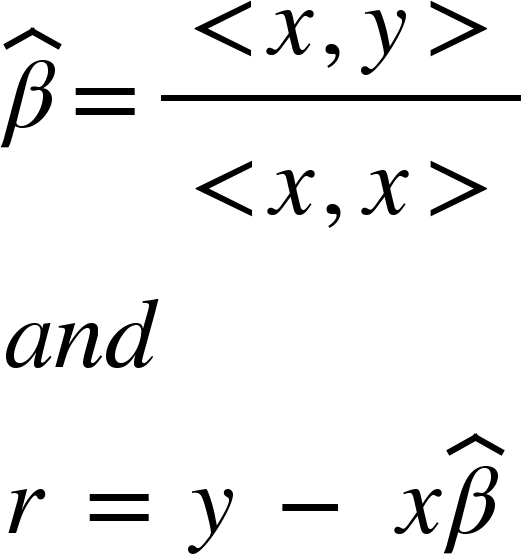
A linear model with p>1 is called a multiple regression model.

Y=X\beta + \epsilon

The least squares estimate and residuals are

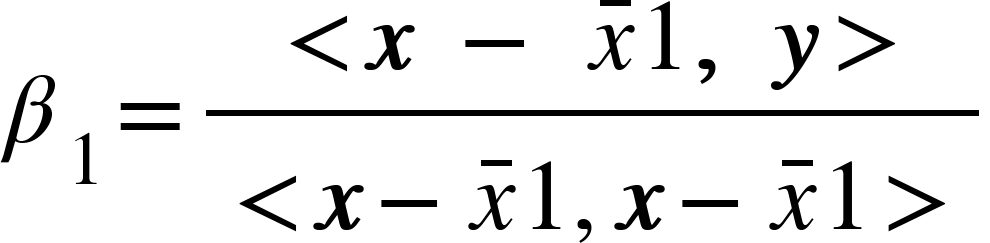


In convenient vector notation, this is given by



Hence, this gives the basis for multiple linear regression. When the inputs are orthogonal, or each <xi,xj> pair is 0, the univariate estimates are obtained.

Since inputs are not usually orthogonal, we orthogonalize them by subtracting the mean from each value of x.



The estimate can be just viewed as two steps

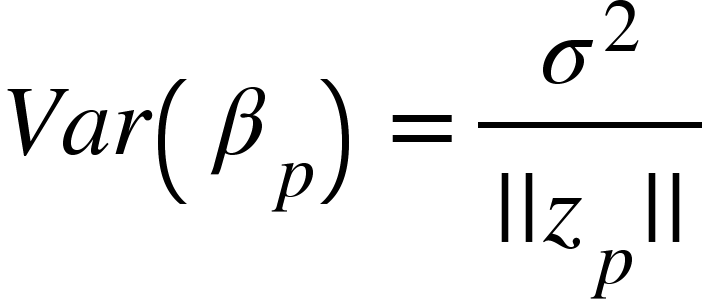
1. Regress x on 1 to produce residual z = x - x’1
2. Regress y on residual z to give the coefficient

So for all the attributes, carry out this process to find all z values by regressing on each x and then regress the final value on y.

Since all the values of z are orthogonal, they form the basis for the column space of X, and hence the least squares projection of y.

The multiple regression coefficient represents the additional contribution of xj on y, after xj has been adjusted for all the values of x for a set of parameters p.

If the x values are highly correlated, the z score will come down, resulting in unstable beta coefficients. So, we can define an alternative variance as



In other words, the precision of the estimate depends on the length of the residual vector zp, representing how much of xp is unexplained by the other x values for a parameter p.

This procedure is called the Gram Schmidt procedure.

The entire algorithm is given by

1. Initialise z0 = x0 = 1
2. For j = 1, 2, …..p
   1. Regress xj on z0,z1,...zj-1 to produce coefficients \gamma$_l$_j = <zl,xj>/<zl,zl>, l = 0...j-1 and residual vector zj = xj - sum(\gamma$_l$_jzk)
3. Regress y on residual zp to give estimate \beta$_p

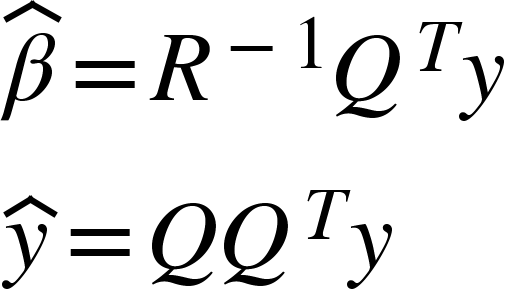
We can represent this is matrix form as **X=ZT** where Z has the columns zj and T is the upper triangular matrix with entries \gamma$_l$_j.

For the diagonal matrix Djj = ||zj||, we get

**X = ZD-1DT**

**X = QR**

Then, the least squares solution is given by



**Multiple Outputs**

Suppose there are outputs Y1,Y2,... desired to be found from inputs X. We assume a linear model for each output as

**Y = XB + E**

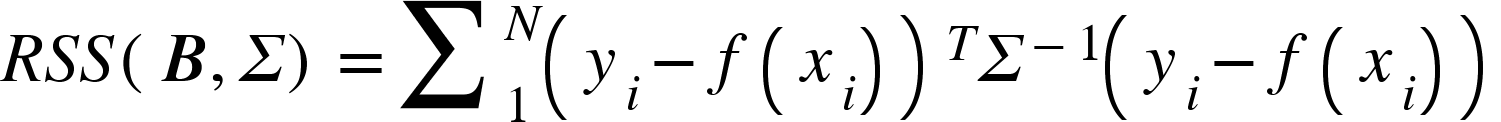
This gives

RSS(**B**) = transpose[(**Y - XB**)T(**Y** - **XB**)]

The least squares estimate is thus obtained as

**B =** (**X**T**X**)-1**X**T**Y**

If the errors are correlated, this equation needs to be modified as



**Subset Selection**

The main drawbacks of the least squares estimate are

1. Prediction accuracy usually has low bias and high variance, and can be improved by disregarding some coefficients or shrinking them.
2. Interpreting a large number of predictors is hard

So we select subsets of parameters and choose only them for our regression model.

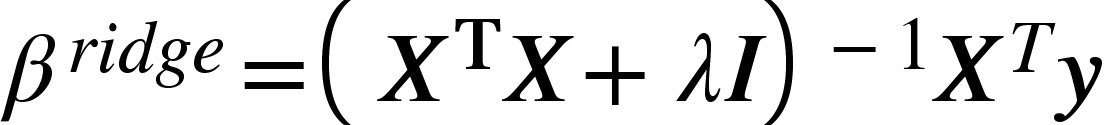
1. Best subset selection
   1. Finds a subset of size k that gives the smallest RSS value
   2. Leaps and bounds is efficient for parameters upto 30 to 40.
2. Forward-Backward Stepwise selection
   1. In the forward stepwise selection method, we start with an empty set and add predictors only if they increase the accuracy
   2. In backward stepwise, we eliminate the worst predictor if it does not change the accuracy of the model
3. Forward stagewise regression
   1. It starts with forward stepwise regression, with an intercept of the y\_mean, and centered predictors with coefficients 0.
   2. At each step, find the variable most correlated with the residual and compute the simple linear regression coefficient on that variable, and adds it to the coefficient of that variable
   3. This is continued until none of the variables have coefficients with the residuals.
   4. This is inefficient due to slow fitting.

**Shrinkage Methods**

Subset selection exhibits high variance because it is a discrete process, and does not reduce the prediction error of the full model. Shrinkage methods are more continuous, and do not suffer from high variability.

1. **Ridge Regression**
   1. This shrinks the coefficients by imposing a penalty on their size

<math xmlns="http://www.w3.org/1998/Math/MathML"><msup><mover><mi>&#x3B2;</mi><mo>^</mo></mover><mrow><mi>r</mi><mi>i</mi><mi>d</mi><mi>g</mi><mi>e</mi></mrow></msup><mo>=</mo><munder><mrow><mi>a</mi><mi>r</mi><mi>g</mi><mi>m</mi><mi>i</mi><mi>n</mi></mrow><mi>&#x3B2;</mi></munder><mfenced open="{" close="}"><mrow><munderover><mo>&#x2211;</mo><mrow><mi>i</mi><mo>=</mo><mn>1</mn></mrow><mi>N</mi></munderover><msup><mfenced><mrow><msub><mi>y</mi><mi>i</mi></msub><mo>-</mo><msub><mi>&#x3B2;</mi><mn>0</mn></msub><mo>-</mo><munderover><mo>&#x2211;</mo><mrow><mi>i</mi><mo>=</mo><mn>1</mn></mrow><mi>p</mi></munderover><msub><mi>x</mi><mrow><mi>i</mi><mi>j</mi></mrow></msub><msub><mi>&#x3B2;</mi><mi>j</mi></msub></mrow></mfenced><mn>2</mn></msup><mo>+</mo><mi>&#x3BB;</mi><munderover><mo>&#x2211;</mo><mrow><mi>j</mi><mo>=</mo><mn>1</mn></mrow><mi>p</mi></munderover><msup><msub><mi>&#x3B2;</mi><mi>j</mi></msub><mn>2</mn></msup></mrow></mfenced><mspace linebreak="newline"/><mi>w</mi><mi>h</mi><mi>i</mi><mi>c</mi><mi>h</mi><mo>&#xA0;</mo><mi>c</mi><mi>a</mi><mi>n</mi><mo>&#xA0;</mo><mi>b</mi><mi>e</mi><mo>&#xA0;</mo><mi>w</mi><mi>r</mi><mi>i</mi><mi>t</mi><mi>t</mi><mi>e</mi><mi>n</mi><mo>&#xA0;</mo><mi>a</mi><mi>s</mi><mspace linebreak="newline"/><msup><mover><mi>&#x3B2;</mi><mo>^</mo></mover><mrow><mi>r</mi><mi>i</mi><mi>d</mi><mi>g</mi><mi>e</mi></mrow></msup><mo>=</mo><munder><mrow><mi>a</mi><mi>r</mi><mi>g</mi><mi>m</mi><mi>i</mi><mi>n</mi></mrow><mi>&#x3B2;</mi></munder><mfenced open="{" close="}"><mrow><munderover><mo>&#x2211;</mo><mrow><mi>i</mi><mo>=</mo><mn>1</mn></mrow><mi>N</mi></munderover><msup><mfenced><mrow><msub><mi>y</mi><mi>i</mi></msub><mo>-</mo><msub><mi>&#x3B2;</mi><mn>0</mn></msub><mo>-</mo><munderover><mo>&#x2211;</mo><mrow><mi>i</mi><mo>=</mo><mn>1</mn></mrow><mi>p</mi></munderover><msub><mi>x</mi><mrow><mi>i</mi><mi>j</mi></mrow></msub><msub><mi>&#x3B2;</mi><mi>j</mi></msub></mrow></mfenced><mn>2</mn></msup></mrow></mfenced><mspace linebreak="newline"/><mi>s</mi><mi>u</mi><mi>b</mi><mi>j</mi><mi>e</mi><mi>c</mi><mi>t</mi><mo>&#xA0;</mo><mi>t</mi><mi>o</mi><mo>&#xA0;</mo><munderover><mo>&#x2211;</mo><mrow><mi>j</mi><mo>=</mo><mn>1</mn></mrow><mi>p</mi></munderover><msub><msup><mi>&#x3B2;</mi><mn>2</mn></msup><mi>j</mi></msub><mo>&#xA0;</mo><mo>&#x2264;</mo><mo>&#xA0;</mo><mi>t</mi></math>

* 1. The intercept B0 has been left out, because leaving it in would make the other coefficients be heavily dependent on the origin, and hence is just taken as the y\_mean.
  2. This hence gives 
  3. The ridge regression is a linear function of y.
  4. It adds a positive constant to XTX’s diagonal, making the problem non singular

1. **Lasso Regression**
   1. The lasso equation is the same as ridge, but instead of adding B2, we add |B|.
   2. This makes the solution non linear in y, and there exists no closed form expression.
   3. This is called soft thresholding, as the coefficient is translated by a constant factor, truncating at 0.