> # Bootstrap regression example
> 
> > kars = read.table("http://www.utstat.toronto.edu/~brunner/appliedf12/data/mcars3.data", + header=T)
> > kars[1:4,]
>   Cntry kpl weight length
> 1   US  5.04 2.1780 5.9182
> 2 Japan 10.08 1.0260 4.3180
> 3   US  9.24 1.1880 4.2672
> 4   US  7.98 1.4445 5.1054
> > attach(kars) # Variables are now available by name
> >
> > # Before regression, a garden variety univariate bootstrap
> > hist(kpl) # Right skewed
>

![Histogram of kpl](image)

> # Small example for demonstration of R syntax
> > set.seed(3244)
> > x = kpl[1:10]; x
> > n = length(x)
> > # Sample of size n from the numbers 1:n, with replacement.
> > choices = sample(1:n,size=n,replace=T); choices
> [1]  2  7  5  1  4  6  9  8  4 10
> > x[choices]
> [1] 10.08  9.66  7.98  5.04  7.98  7.98  5.88  7.56  7.98 10.92
> # Now bootstrap the mean of kpl
> n = length(kpl); B = 1000
> mstar = NULL # mstar will contain bootstrap mean values
>
> for(draw in 1:B) mstar = c(mstar,mean(kpl[sample(1:n,size=n,replace=T)]))
> hist(mstar)
> # Look at a normal qq plot. That's a plot of the order statistics against
> # the corresponding quantiles of the (standard) normal. Should be roughly linear
> # if the data are from a normal distribution.

> qnorm(mstar); qqline(mstar)

![Normal Q-Q Plot](image)

> # Quantile bootstrap CI for mu. Use ONLY if the bootstrap distribution is symmetric.
> sort(mstar)[25]; sort(mstar)[975]
> [1] 8.3034
> [1] 9.3492
> # Compare the usual CI
> t.test(kpl)

One Sample t-test

data:  kpl
t = 32.9363, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
  8.264966 9.324634
sample estimates:
  mean of x
  8.7948
> # Now regression
> # Compute some polynomial terms
> wsq = weight^2; lsq = length^2; wl = weight*length
> # Bind it into a nice data frame
> datta = cbind(kpl,weight,length,wsq,lsq,wl)
> datta = as.data.frame(datta)
> 
> modell = lm(kpl ~ weight + length + wsq + lsq + wl, data=datta)
> summary(modell)

Call:
  lm(formula = kpl ~ weight + length + wsq + lsq + wl, data = datta)

Residuals:
     Min      1Q  Median      3Q     Max
  -4.0861 -0.8702  0.0490  0.6898  4.4006

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)     79.124214   29.12112  2.717   0.00784 **
weight          24.336109   26.57008  0.916  0.36204
length         -33.763782   19.35035 -1.745  0.08427 .
wsq             11.376646    8.53130  1.334  0.18556
lsq              5.140000    3.41003  1.507  0.13508
wl            -12.442449  10.174080 -1.223  0.22442

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.577 on 94 degrees of freedom
Multiple R-squared:  0.6689,    Adjusted R-squared:  0.6513
F-statistic: 37.98 on 5 and 94 DF,  p-value: < 2.2e-16

> betahat = coef(modell); betahat

(Intercept)     weight      length       wsq       lsq         wl
  79.124214   24.336109  -33.763782  11.376646  5.140000  -12.442449

> set.seed(3244)
> bstar = NULL # Rows of bstar will be bootstrap vectors of regression coefficients.
> n = length(kpl); B = 1000
> for(draw in 1:B)
+  {
+    # Randomly sample from the rows of kars, with replacement
+    Dstar = datta[sample(1:n,size=n,replace=T),]
+    model = lm(kpl ~ weight + length + wsq + lsq + wl, data=Dstar)
+    bstar = rbind( bstar,coef(model) )
+  } # Next draw
>
> bstar[1:5,]

     (Intercept) weight length  wsq  lsq      wl
[1,]    64.73852  15.322187 -25.549389 14.376646  5.139649  -12.442449
[2,]  270.35328 158.868074 -149.690584 26.298031 21.834487  47.728186
[3,]  30.97446  -0.156246  10.815623  1.853871  2.162598   1.070870
[5,]  74.31620  -4.632140 -23.726771  1.497523  2.162598  1.070870

> Vb = var(bstar)  # Approximate asymptotic covariance matrix of betahat
> Vb

<table>
<thead>
<tr>
<th></th>
<th>weight</th>
<th>length</th>
<th>wsq</th>
<th>lsq</th>
<th>wl</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>4009.6</td>
<td>2805.7</td>
<td>2432.9</td>
<td>403.6</td>
<td>359.9</td>
</tr>
<tr>
<td>weight</td>
<td>2805.7</td>
<td>1816.7</td>
<td>1511.5</td>
<td>434.7</td>
<td>359.9</td>
</tr>
<tr>
<td>length</td>
<td>2432.9</td>
<td>1816.7</td>
<td>292.3</td>
<td>52.9</td>
<td>158.0</td>
</tr>
<tr>
<td>wsq</td>
<td>403.6</td>
<td>434.7</td>
<td>117.9</td>
<td>52.9</td>
<td>158.0</td>
</tr>
<tr>
<td>lsq</td>
<td>359.9</td>
<td>288.9</td>
<td>292.3</td>
<td>52.9</td>
<td>89.1</td>
</tr>
<tr>
<td>wl</td>
<td>795.8</td>
<td>724.6</td>
<td>229.9</td>
<td>534.8</td>
<td>239.4</td>
</tr>
</tbody>
</table>

> # Test individual coefficients.  H0: betaj=0
> se = sqrt(diag(Vb)); Z = betahat/se
> rbind(betahat,se,Z)

<table>
<thead>
<tr>
<th></th>
<th>weight</th>
<th>length</th>
<th>wsq</th>
<th>lsq</th>
<th>wl</th>
</tr>
</thead>
<tbody>
<tr>
<td>betahat</td>
<td>79.12</td>
<td>24.34</td>
<td>-33.8</td>
<td>11.3</td>
<td>5.14</td>
</tr>
<tr>
<td>se</td>
<td>63.32</td>
<td>48.34</td>
<td>38.8</td>
<td>10.8</td>
<td>6.02</td>
</tr>
<tr>
<td>Z</td>
<td>1.25</td>
<td>0.50</td>
<td>-0.87</td>
<td>1.05</td>
<td>0.85</td>
</tr>
</tbody>
</table>

> # Now test the product terms all at once
>
> WaldTest = function(L,thetahat,Vn,h=0)  # H0: L theta = h
+ # Note Vn is the asymptotic covariance matrix, so it's the
+ # Consistent estimator divided by n. For true Wald tests
+ # based on numerical MLEs, just use the inverse of the Hessian.
+ {
+   WaldTest = numeric(3)
+   names(WaldTest) = c("W","df","p-value")
+   r = dim(L)[1]
+   W = t(L%*%thetahat-h) %*% solve(L%*%Vn%*%t(L)) %*% (L%*%thetahat-h)
+   W = as.numeric(W)
+   pval = 1-pchisq(W,r)
+   WaldTest
+ } # End function WaldTest
>
> Lprod = rbind( c(0,0,0,1,0,0),
+                c(0,0,0,0,1,0),
+                c(0,0,0,0,0,1) )
> WaldTest(Lprod,betahat,Vb)

  W     df  p-value
  9.46  3.00 0.02372

> # Normal test for comparison
> model0 = lm(kpl ~ weight + length)  # No product terms
>anova(model0,model1)  # p = 0.0133

Analysis of Variance Table

Model 1: kpl ~ weight + length
Model 2: kpl ~ weight + length + wsq + lsq + wl

<table>
<thead>
<tr>
<th></th>
<th>Res.Df</th>
<th>RSS</th>
<th>Sum of Sq</th>
<th>F</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>97</td>
<td>261.81</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>94</td>
<td>233.72</td>
<td>28.095</td>
<td>3</td>
<td>0.0133</td>
</tr>
</tbody>
</table>

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Final comment: This is not a typical bootstrap regression. It's more common to bootstrap the residuals. But that applies to a conditional model in which the values of the explanatory variables are fixed constants.