Review: Normal Regression with R*

```
> kars = read.table("http://www.utstat.utoronto.ca/~brunner/data/legal/mcars4.data.txt")
> head(kars)
  Cntry lper100k weight length
1    US     19.8   2178   5.92
2 Japan      9.9   1026   4.32
3    US     10.8   1188   4.27
4    US     12.5   1444   5.11
5    US     12.5   1485   5.03
6    US     12.5   1485   5.03
>
> attach(kars) # Variables are now available by name
> n = length(Cntry); n
[1] 100
> # Make indicator dummy variables for Cntry. Just use 2 for now.
> # U.S. will be the reference category
> c1 = numeric(n); c1[Cntry=='Europ'] = 1
> table(c1,Cntry)
   Cntry
c1  Europ Japan US
  0     0    13 73
  1    14     0  0
> c2 = numeric(n); c2[Cntry=='Japan'] = 1
> table(c2,Cntry)
   Cntry
c2  Europ Japan US
  0    14     0 73
  1     0    13  0
>
> c3 = numeric(n); c3[Cntry=='US'] = 1
> table(c3,Cntry)
   Cntry
c3  Europ Japan US
  0    14    13  0
  1     0     0 73
```

* Copyright information is on the last page.

```
> # Take a look at mean fuel consumption for each country
> aggregate(lper100k,by=list(Cntry),FUN=mean)
  Group.1        x
1   Europ 10.17857
2   Japan 10.68462
3      US 12.96438
> # Must specify a LIST of grouping factors
```

On average, the U.S. cars seem to be using more fuel. Back it up with a hypothesis test.

Origin	c1	c2	$E(Y\|\mathbf{X}=\mathbf{x}) = \beta_0 + \beta_1 C_1 + \beta_2 C_2$
Europe	1	0	$\beta_0 + \beta_1$
Japan	0	1	$\beta_0 + \beta_2$
U.S.	0	0	β_0

```
> # H0: mu1=mu2=mu3
> justcountry = lm(lper100k ~ c1+c2)
> summary(justcountry)

Call:
lm(formula = lper100k ~ c1 + c2)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.0644 -2.1644 -0.4644  2.5154  6.8356 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  12.9644     0.3651  35.511  < 2e-16 ***
c1           -2.7858     0.9101  -3.061  0.00285 ** 
c2           -2.2798     0.9390  -2.428  0.01703 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

Residual standard error: 3.119 on 97 degrees of freedom
Multiple R-squared: 0.1203,	Adjusted R-squared: 0.1022 
F-statistic: 6.634 on 2 and 97 DF,  p-value: 0.001993
```

```
>
> # Which means are different?
> Have t-tests. What about Europe vs. Japan?
> # Test H0: beta1 = beta2
> # A cheap way is to use a different reference category.

> # R can make the dummy variables for you
> is.factor(Cntry)
[1] TRUE
> # The factor Cntry has dummy vars built in. What are they?
> contrasts(Cntry) # Note alphabetical order
      Japan US
Europ     0  0
Japan     1  0
US        0  1
>
> jc2 = lm(lper100k~Cntry); summary(jc2)

Call:
lm(formula = lper100k ~ Cntry)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.0644 -2.1644 -0.4644  2.5154  6.8356 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  10.1786     0.8337  12.209  < 2e-16 ***
CntryJapan    0.5060     1.2014   0.421  0.67454    
CntryUS       2.7858     0.9101   3.061  0.00285 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.119 on 97 degrees of freedom
Multiple R-squared: 0.1203,	Adjusted R-squared: 0.1022 
F-statistic: 6.634 on 2 and 97 DF,  p-value: 0.001993
```

Conclusion: American cars are getting fewer kilometers per litre on average than Japanese and European cars. There is no evidence of different average fuel efficiency for European and Japanese cars.

```
> # You can select the dummy variable coding scheme.
> contr.treatment(3,base=2) # Category 2 is the reference category
  1 3
1 1 0
2 0 0
3 0 1

> # U.S. as reference category again
> Country = Cntry
> contrasts(Country) = contr.treatment(3,base=3)
> summary(lm(lper100k~Country))

Call:
lm(formula = lper100k ~ Country)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.0644 -2.1644 -0.4644  2.5154  6.8356 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  12.9644     0.3651  35.511  < 2e-16 ***
Country1     -2.7858     0.9101  -3.061  0.00285 ** 
Country2     -2.2798     0.9390  -2.428  0.01703 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

Residual standard error: 3.119 on 97 degrees of freedom
Multiple R-squared: 0.1203,	Adjusted R-squared: 0.1022 
F-statistic: 6.634 on 2 and 97 DF,  p-value: 0.001993 

> # Names of dummy variables 1=Europe, 2=Japan could be nicer
> colnames(contrasts(Country)) = c("Europe","Japan")
> contrasts(Country)
      Europe Japan
Europ      1     0
Japan      0     1
US         0     0
```

Include covariates

Origin	c1	c2	$E(Y\mid\mathbf{X}=\mathbf{x}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 C_1 + \beta_4 C_2$
Europe	1	0	$(\beta_0 + \beta_3) + \beta_1 X_1 + \beta_2 X_2$
Japan	0	1	$(\beta_0 + \beta_4) + \beta_1 X_1 + \beta_2 X_2$
U.S.	0	0	$\beta_0 \quad + \beta_1 X_1 + \beta_2 X_2$

```
> # Include covariates
> fullmodel = lm(lper100k ~ weight+length+Country)
> summary(fullmodel) # Look carefully at the signs!


Call:
lm(formula = lper100k ~ weight + length + Country)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.5063 -0.8813  0.0147  1.3043  2.9432 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -7.276937   3.006354  -2.421 0.017399 *  
weight         0.005457   0.001472   3.707 0.000352 ***
length         2.345968   0.980329   2.393 0.018676 *  
CountryEurope  1.487722   0.575633   2.584 0.011274 *  
CountryJapan   1.994239   0.584995   3.409 0.000958 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.703 on 95 degrees of freedom
Multiple R-squared:  0.7431,	Adjusted R-squared:  0.7323 
F-statistic: 68.71 on 4 and 95 DF,  p-value: < 2.2e-16
```

```
> # Test car size controlling for country
> anova(justcountry,fullmodel) # Full vs restricted
Analysis of Variance Table

Model 1: lper100k ~ c1 + c2
Model 2: lper100k ~ weight + length + Country
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     97 943.81
2     95 275.61  2     668.2 115.16 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> # I advise using anova ONLY to compare full and reduced models
>
> # Test country controlling for size too.
> justsize = lm(lper100k ~ weight+length); summary(justsize)

Call:
lm(formula = lper100k ~ weight + length)

Residuals:
    Min      1Q  Median      3Q     Max
-4.3857 -1.0684 -0.0556  1.3077  4.0429

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.617472   2.958472  -1.223  0.22439
weight       0.004949   0.001546   3.202  0.00185 **
length       1.835625   1.017349   1.804  0.07428 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.804 on 97 degrees of freedom
Multiple R-squared: 0.7058,   Adjusted R-squared: 0.6997
F-statistic: 116.4 on 2 and 97 DF,  p-value: < 2.2e-16
> anova(justsize,fullmodel)
Analysis of Variance Table

Model 1: lper100k ~ weight + length
Model 2: lper100k ~ weight + length + Country
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     97 315.64
2     95 275.61  2    40.035 6.8999 0.001592 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> # General linear test of L beta = h
```

$$F = \frac{(\mathbf{L}\widehat{\boldsymbol{\beta}}-\mathbf{h})^\top (\mathbf{L}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{L}^\top)^{-1}(\mathbf{L}\widehat{\boldsymbol{\beta}}-\mathbf{h})}{r\, MSE_F} \stackrel{H_0}{\sim} F(r, n-p)$$

```
> source("http://www.utstat.utoronto.ca/~brunner/Rfunctions/ftest.txt")
> ftest

function(model,L,h=0)
# General linear test of H0: L beta = h
    {
    BetaHat = model$coefficients
    dimL = dim(L)
    if(length(BetaHat) != dimL[2]) stop("Sizes of L and Beta are incompatible")
    r = dimL[1]
    if(qr(L)$rank != r) stop("Rows of L must be linearly independent.")
    out = numeric(4)
    names(out) = c("F","df1","df2","p-value")
    dfe = df.residual(model)
    diff = L%*%BetaHat-h
    fstat = t(diff) %*% solve(L%*%vcov(model)%*%t(L)) %*% diff / r
    # Note vcov = MSE * XtXinv
    fstat = as.numeric(fstat)
    out[1] = fstat; out[2]=r; out[3]=dfe
    out[4] = 1-pf(fstat,r,dfe)
    return(out)
    }
```

```
> # Test country controlling for size
>
> summary(fullmodel) # Full model again

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -7.276937   3.006354  -2.421 0.017399 *
weight         0.005457   0.001472   3.707 0.000352 ***
length         2.345968   0.980329   2.393 0.018676 *
CountryEurope  1.487722   0.575633   2.584 0.011274 *
CountryJapan   1.994239   0.584995   3.409 0.000958 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


> # Now the F-test of couuntry controlling for size: : F = 6.8999
> L0 = rbind(c(0,0,0,1,0),
+            c(0,0,0,0,1))
> L0
     [,1] [,2] [,3] [,4] [,5]
[1,]    0    0    0    1    0
[2,]    0    0    0    0    1
>
> ftest(fullmodel,L0)
          F          df1          df2      p-value
 6.899949667  2.000000000 95.000000000  0.001592274

> # As before, t-tests give comparison of U.S with Europe an Japan.
> # Test Europe vs. Japan controlling for size.
> L1 = cbind(0,0,0,1,-1) # One row, 5 columns
> ftest(fullmodel,L1)
        F        df1        df2    p-value
 0.5886970  1.0000000 95.0000000  0.4448261
```

```
>
> ###### Predictions, confidence intervals and prediction intervals ######
>
> # Predict litres per 100 km for a Japanese car weighing
> # 1295kg, 4.52m long (1990 Toyota Camry)
>
> b = fullmodel$coefficients; b
 (Intercept)       weight       length   CntryJapan      CntryUS
-5.789214693  0.005456609  2.345968436  0.506517030 -1.487721833
> ell = c(1,1295,4.52,1,0)
> yhat = sum(ell*b); # ell-prime b
> yhat
[1] 12.38739
>
> # Confidence interval for E(ell-prime beta)
> # First the hard way
```

$$\ell' \mathbf{b} \pm t_{\alpha/2} \sqrt{\ell' s^2 (X'X)^{-1} \ell}$$

```
>
> tcrit = qt(0.975,df=fullmodel$df.residual) # t_alpha/2
> MSE.XpXinv = vcov(fullmodel)
> ell = as.matrix(ell) # Now it's a column vector
> me95 = tcrit * sqrt( as.numeric(t(ell) %*% MSE.XpXinv %*% ell) )
> lower95 = yhat - me95; upper95 = yhat + me95
> c(lower95, upper95) # 95% Confidence interval for ell-prime beta
[1] 11.37128 13.40349
>
> # Use the predict function
> # help(predict.lm)
>
> camry1990 = data.frame(weight=1295,length=4.52,Cntry='Japan')
> camry1990
  weight length Cntry
1   1295   4.52 Japan
> predict(fullmodel,newdata=camry1990) # Compare yhat = 12.38739
       1
12.38739
> predict(fullmodel,newdata=camry1990, interval='confidence')
       fit      lwr      upr
1 12.38739 11.37128 13.40349




>
```

```
> # With 95 percent prediction interval (95 is default)
```

$$\boldsymbol{\ell}'\mathbf{b} \pm t_{\alpha/2}\sqrt{s^2\left(1+\boldsymbol{\ell}'(X'X)^{-1}\boldsymbol{\ell}\right)}$$

```
> predict(fullmodel,newdata=camry1990, interval='prediction')
       fit      lwr      upr
1 12.38739 8.856608 15.91817
>

>  # Multiple predictions
> cadillac1990 = data.frame(weight=1800,length=5.22,Cntry='US')
> volvo1990 = data.frame(weight=1371,length=4.823,Cntry='Europ')
> newcars = rbind(camry1990,cadillac1990,volvo1990); newcars
  weight length Cntry
1   1295  4.520 Japan
2   1800  5.220    US
3   1371  4.823 Europ
>
> is.data.frame(newcars)
[1] TRUE
>
> predict(fullmodel,newdata=newcars, interval='prediction')
       fit       lwr      upr
1 12.38739  8.856608 15.91817
2 14.79091 11.354379 18.22745
3 13.00640  9.481598 16.53121
>
```