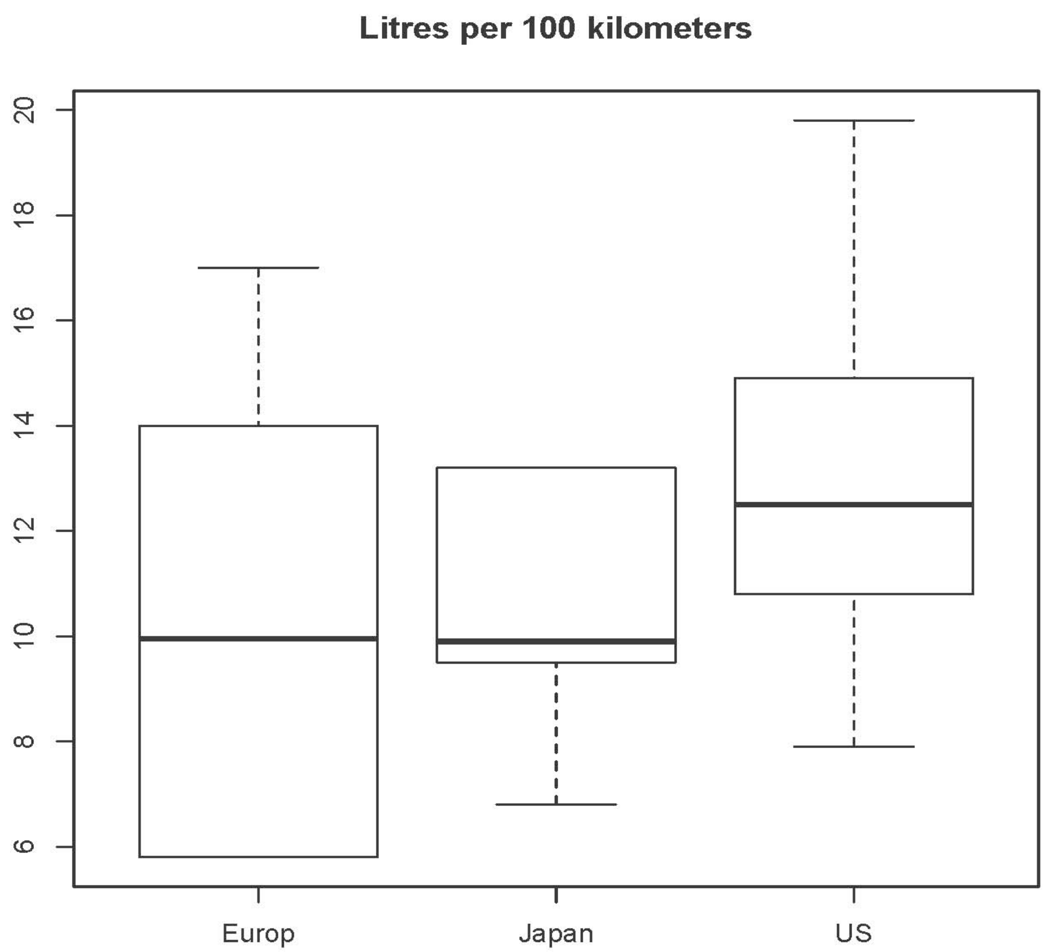
Ordinary Multiple Regression with R*

```
> kars = read.table("http://www.utstat.toronto.edu/~brunner/data/legal/mcars4.data.txt")
> kars[1:4,]
  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
>
> attach(kars) # Variables are now available by name
> # Boxplots by country
> boxplot(lper100k~Cntry); title("Litres per 10 kilometers")
```



* Copyright information is on the last page.

```
> n = length(length); n
[1] 100
> # Make indicator dummy variables for Cntry
> # 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
>
> # Take a look at mean fuel consumption per 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

```
> # One-factor ANOVA to compare means
> 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
```

```
>
> # Get nicer-looking ANOVA summary table
> is.factor(Cntry)
[1] TRUE
> jc2 = aov(lper100k~Cntry); summary(jc2)  # aov is a wrapper for lm
            Df Sum Sq Mean Sq F value  Pr(>F)
Cntry        2  129.1   64.55   6.634 0.00199 **
Residuals   97  943.8    9.73
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
> # Which means are different?
> TukeyHSD(jc2,ordered=T)
  Tukey multiple comparisons of means
    95% family-wise confidence level
    factor levels have been ordered

Fit: aov(formula = lper100k ~ Cntry)

$Cntry
                diff         lwr      upr     p adj
Japan-Europ 0.506044 -2.35364917 3.365737 0.9069443
US-Europ    2.785812  0.61956789 4.952056 0.0079628
US-Japan    2.279768  0.04470727 4.514829 0.0445191

>
> # 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
>
> summary(lm(lper100k~Cntry))
```

```
> summary(lm(lper100k~Cntry))

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

> # You can select the dummy variable coding scheme.
> contr.sum(3) # Effect coding
  [,1] [,2]
1    1    0
2    0    1
3   -1   -1
> 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
```

Include covariates

Origin	c1	c2	$E(Y\|\mathbf{X}=\mathbf{x}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 C_1 + \beta_4 C_2$
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 *  
Country1     1.487722   0.575633   2.584 0.011274 *  
Country2     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 
> # Mean fuel Consumption "adjusted" for the covariates is Y-hat
> # for each country, with covariates set to overall sample mean values.
> xbar1=mean(weight); xbar2= mean(length)
> betahat = fullmodel$coefficients; betahat
 (Intercept)       weight       length     Country1     Country2 
-7.276936526  0.005456609  2.345968436  1.487721833  1.994238863 
> adjEurope = sum(betahat*c(1,xbar1,xbar2,1,0)); adjEurope
[1] 13.29819
> adjJapan = sum(betahat*c(1,xbar1,xbar2,0,1)); adjJapan
[1] 13.80471
> adjUS = sum(betahat*c(1,xbar1,xbar2,0,0)); adjUS
[1] 11.81047
```

```
> # Test car size controlling for country
> anova(justcountry,fullmodel) # Full vs reduced
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. It's the main question.
> 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
```

```
> # How strong is the relationship? Proportion of remaining variation.
```

$$a = \frac{R_F^2 - R_R^2}{1 - R_R^2} = \frac{rF}{n - p + rF}$$

```
> # R^2 for full model is 0.7431 and R^2 for reduced model is 0.7058
> (0.7431-0.7058)/(1-0.7058)
[1] 0.1267845
> # Or equivalently,
> anova(justsize,fullmodel) # Repeating
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
> 2*6.8999/(95+2*6.8999)
[1] 0.1268366

> # How about confidence intervals? Based on output from full model,

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 .

>
> critval = qt(0.975,95)
> # Margin of error is critical value times standard error
> me1 = critval*0.575633; me2 = critval*0.584995
> # Europe vs US
> LO1 = betahat[4]-me1; HI1 = betahat[4]+me1; cbind(LO1, HI1)
               LO1      HI1
Country1 0.3449458 2.630498
> # Japan vs US
> LO2 = betahat[5]-me2; HI2 = betahat[5]+me2; cbind(LO2, HI2)
              LO2      HI2
Country2 0.832877 3.155601
```

```
>
> # General linear test: Compare F = 6.8999
```

$$F = \frac{(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{h})^\top (\mathbf{L}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{L}^\top)^{-1} (\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{h})}{r\, MSE}$$

```
> # First do it the hard way
> fullmod = lm(lper100k ~ weight+length+Country,x=T) # To get X matrix, x=T
> L = rbind(c(0,0,0,1,0),
+           c(0,0,0,0,1))
> r = dim(L)[1]
> X = fullmod$x
> xtxinv = solve(t(X)%*%X)
> MSE = deviance(fullmod)/df.residual(fullmod)  # MSE = SSE/(n-p)
> Fstat = ( t(L%*%betahat) %*% solve(L%*%xtxinv%*%t(L)) %*% L%*%betahat ) /
(r*MSE)
> Fstat = as.numeric(Fstat); Fstat
[1] 6.89995

> # The easy way
> source("http://www.utstat.utoronto.ca/~brunner/Rfunctions/ftest.txt")
> ftest # See function definition
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)
    }
> ftest(fullmodel,LL) # Again compare  F = 6.8999 from full-reduced
           F          df1          df2      p-value
 6.899949667  2.000000000 95.000000000  0.001592274
```

```
> ###### Predictions and prediction intervals ######
>
> # Predict litres per 100 km for a Japanese car weighing 1295kg, 4.52m
long
> # (1990 Toyota Camry)
> betahat = fullmodel$coefficients; betahat
 (Intercept)       weight       length     Country1     Country2 
-7.276936526  0.005456609  2.345968436  1.487721833  1.994238863 
> contrasts(Country)
      1 2
Europ 1 0
Japan 0 1
US    0 0
> x1 = c(1,1295,4.52,0,1)
> sum(x1*betahat)
[1] 12.38739
>
> # Use the predict function
> # help(predict.lm)
>
> camry1990 = data.frame(weight=1295,length=4.52,Country='Japan');
camry1990
  weight length Country
1   1295   4.52   Japan
> predict(fullmodel,newdata=camry1990)
       1 
12.38739 
> # With 95 percent prediction interval (default)
> 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,Country='US')
> volvo1990 = data.frame(weight=1371,length=4.823,Country='Europ')
> newcars = rbind(camry1990,cadillac1990,volvo1990); newcars
  weight length Country
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
>
> # It's not a bad idea to "predict" the observed data
> # Just look at the first 10 rows, for example
> cbind(lper100k,predict(fullmodel,interval='prediction'))[1:10,]
   lper100k       fit       lwr      upr
1      19.8 18.495691 15.012790 21.97859
2       9.9 10.450367  6.940724 13.96001
3      10.8  9.222800  5.747978 12.69762
4      12.5 12.590305  9.162660 16.01795
5      12.5 12.626349  9.219284 16.03341
6      12.5 12.626349  9.219284 16.03341
7      10.4  9.766491  6.236956 13.29603
8      13.2 14.570386 11.135730 18.00504
9      17.0 14.056832 10.515850 17.59782
10      9.2  8.330626  4.870298 11.79095
Warning message:
In predict.lm(fullmodel, interval = "prediction") :
  Predictions on current data refer to _future_ responses
```
