Prediction intervals with R

```
> kars = read.table("http://www.utstat.utoronto.ca/~brunner/data/legal/mcars4.data")
> head(kars); attach(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
> contrasts(Cntry)
      Japan US
Europ     0  0
Japan     1  0
US        0  1
> fullmodel = lm(lper100k ~ weight + length + Cntry)
> summary(fullmodel)

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

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) -5.789215   2.855736  -2.027 0.045441 *  
weight       0.005457   0.001472   3.707 0.000352 ***
length       2.345968   0.980329   2.393 0.018676 *  
CntryJapan   0.506517   0.660158   0.767 0.444826    
CntryUS     -1.487722   0.575633  -2.584 0.011274 *  
---
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 

> fullmodel$df.residual # n-k-1
[1] 95
```

```
>
> ###### 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
>
```

This handout was prepared by Jerry Brunner, Department of Statistical Sciences, University of Toronto. It is licensed under a Creative Commons Attribution - ShareAlike 3.0 Unported License. Use any part of it as you like and share the result freely. It is available in OpenOffice.org from the course website:

http://www.utstat.toronto.edu/~brunner/oldclass/302f17