Inference with R: Part One

```
> sat = read.table("http://www.utstat.utoronto.ca/~brunner/302f14/code_n_data/
                    lecture/sat.data")
> head(sat)
  VERBAL MATH GPA
1    623  509 2.6
2    454  471 2.3
3    643  700 2.4
4    585  719 3.0
5    719  710 3.1
6    693  643 2.9

> lm(GPA ~ VERBAL+MATH, data=sat)

Call:
lm(formula = GPA ~ VERBAL + MATH, data = sat)

Coefficients:
(Intercept)       VERBAL         MATH
  0.6062975    0.0023072    0.0009999

> mod1 = lm(GPA ~ VERBAL+MATH, data=sat); summary(mod1)

Call:
lm(formula = GPA ~ VERBAL + MATH, data = sat)

Residuals:
     Min       1Q   Median       3Q      Max
-2.24875 -0.35113  0.04659  0.38745  1.03527

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.6062975  0.4414062   1.374    0.171
VERBAL      0.0023072  0.0005522   4.178 4.42e-05 ***
MATH        0.0009999  0.0006093   1.641    0.102
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5484 on 197 degrees of freedom
Multiple R-squared:  0.1161,  Adjusted R-squared:  0.1071
F-statistic: 12.93 on 2 and 197 DF,  p-value: 5.284e-06

> summary(lm(GPA~MATH,data=sat)) # Math not controlling for Verbal

Call:
lm(formula = GPA ~ MATH, data = sat)

Residuals:
     Min       1Q   Median       3Q      Max
-2.26114 -0.35543  0.01944  0.36817  1.15075

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.5264336  0.3981176   3.834 0.000169 ***
MATH        0.0016990  0.0006098   2.786 0.005850 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5707 on 198 degrees of freedom
Multiple R-squared: 0.03773,    Adjusted R-squared: 0.03287
F-statistic: 7.764 on 1 and 198 DF,  p-value: 0.00585
```

```
> # mod1 is a list of lists -- see help(lm)
> mod1$coefficients
 (Intercept)       VERBAL         MATH
0.6062974824 0.0023071729 0.0009998537
> mod1$residuals[1:8] # Just the first 8 -- There are 200
          1           2           3           4           5           6
 0.04740826  0.17531492 -0.38970725  0.32511156  0.12494907  0.05192576
          7           8
 0.51140408  0.48437401
> # Summary object is a list of lists too.
> summary(mod1)[1] # First element
$call
lm(formula = GPA ~ VERBAL + MATH, data = sat)

> summary(mod1)[6] # 6th element
$sigma
[1] 0.5483806

> sqrt( sum((mod1$residuals^2))/197 ) # Cross-checking
[1] 0.5483806
> summary(mod1)[10] # 10th element
$fstatistic
   value    numdf    dendf
 12.9321   2.0000 197.0000

> is.list( summary(mod1)[10] )  # It's a list with one element
[1] TRUE
> summary(mod1)[[10]] # 10th element as a numeric vector
   value    numdf    dendf
 12.9321   2.0000 197.0000
> summary(mod1)[[10]][1] # Just the F statistic – first element
  value
12.9321
>
> V = vcov(mod1); V    # Estimated cov(betahat)

              (Intercept)        VERBAL          MATH
(Intercept)  0.1948393932 -1.216189e-04 -1.861246e-04
VERBAL      -0.0001216189  3.049503e-07 -9.241260e-08
MATH        -0.0001861246 -9.241260e-08  3.712995e-07

> # Get it another way
> mod1 = lm(GPA ~ VERBAL+MATH, x=T, data=sat) # mod1 now has the X matrix
> X = mod1$x
> G = solve(t(X) %*% X)  # X-Prime-X Inverse
> s2 = summary(mod1)$sigma^2  # MSE
> s2 * G
              (Intercept)        VERBAL          MATH
(Intercept)  0.1948393932 -1.216189e-04 -1.861246e-04
VERBAL      -0.0001216189  3.049503e-07 -9.241260e-08
MATH        -0.0001861246 -9.241260e-08  3.712995e-07
```

```
> # Confidence interval for betahat1
> betahat1 = mod1$coefficients[2]; betahat1

     VERBAL
0.002307173
> dfe = summary(mod1)[[10]][3]; dfe
dendf
  197
> tcrit = qt(0.975,dfe); tcrit
[1] 1.972079
> se1 = sqrt(V[2,2]); se1
[1] 0.000552223
> low1 = betahat1 - se1*tcrit; up1 = betahat1 + se1*tcrit
> low1; up1
     VERBAL
0.001218145
   VERBAL
0.0033962
```

$$T = \frac{\mathbf{a}'\widehat{\boldsymbol{\beta}} - \mathbf{a}'\boldsymbol{\beta}}{\sqrt{MSE\,\mathbf{a}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{a}}}$$

```
>
> # Test equal regression coefficients for  Math and Verbal
> a = rbind(0,1,-1); a
     [,1]
[1,]    0
[2,]    1
[3,]   -1
> sediff = sqrt(t(a) %*% V %*% a )
> T = t(a) %*% mod1$coefficients / sediff;   T
         [,1]
[1,] 1.408838
> T = as.numeric(T); T    # Make T numeric rather than matrix
[1] 1.408838
> T > tcrit
[1] FALSE
> 2 * (1-pt(T,dfe)) # 2-sided p-value
[1] 0.1604594
```
