

# Tests and Confidence Intervals with R\*

## The Math Data (introduced in the first R lecture)

Before the beginning of the Fall term, students in a first-year Calculus class took a diagnostic test with two parts: Pre-calculus and Calculus. Their High School Calculus marks and their marks in University Calculus were also available.

```
> math = read.table("http://www.utstat.toronto.edu/~brunner/data/legal/mathtest.txt")
> colnames(math) = c("ID", "HScalc", "PreCalcScore", "CalcScore", "UnivCalc")
> head(math)
  ID HScalc PreCalcScore CalcScore UnivCalc
1  1     65           2         0        39
2  2     54           6         2        57
3  3     77           4         4        62
4  4     80           5         2        76
5  5     87           4         4        86
6  6     53           3         1        60

> attach(math) # Make variable names available
> # PreCalc score is out of 9 and Calc score is out of 11. Convert to percentages.
> PreCalcScore = 100 * PreCalcScore/9
> CalcScore = 100 * CalcScore/11

> ##### Fit the full regression model #####
> fullmodel = lm(UnivCalc ~ HScalc+PreCalcScore+CalcScore)
> sumfull = summary(fullmodel); sumfull
```

Call:

```
lm(formula = UnivCalc ~ HScalc + PreCalcScore + CalcScore)
```

Residuals:

Min	1Q	Median	3Q	Max
-48.699	-7.954	1.603	9.242	30.260

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-6.32155	6.01019	-1.052	0.29376
HScalc	0.70097	0.08133	8.619	4.4e-16 ***
PreCalcScore	0.16849	0.05182	3.252	0.00128 **
CalcScore	0.08722	0.04282	2.037	0.04257 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.16 on 291 degrees of freedom

Multiple R-squared: 0.3583, Adjusted R-squared: 0.3517

F-statistic: 54.17 on 3 and 291 DF, p-value: < 2.2e-16

---

\* Copyright information is on the last page.

$$t = \frac{\mathbf{a}'\hat{\boldsymbol{\beta}} - \mathbf{a}'\boldsymbol{\beta}}{\sqrt{MSE \mathbf{a}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{a}}} \sim t(n - k - 1)$$

```
> # Reproduce the test for H0: beta1=0
> a = as.matrix(c(0,1,0,0)); a
      [,1]
[1,]    0
[2,]    1
[3,]    0
[4,]    0
> betahat = coefficients(fullmodel)
> MSE.XpXinv = vcov(fullmodel) # Very handy
> dfe = df.residual(fullmodel) # n-k-1
> tstat = t(a) %*% betahat / sqrt( t(a)%*%MSE.XpXinv%*%a ) # tstat is a 1x1 matrix
> tstat = as.numeric(tstat) # Make it a number
> pvalue = 2 * ( 1-pt(abs(tstat),dfe) ) # 2-sided p-value
> c(tstat,pvalue) # Compare t = 8.619, p = 4.4e-16
[1] 8.618832e+00 4.440892e-16

> # Where do the "standard errors" come from?
> sumfull$coefficients
      Estimate Std. Error  t value    Pr(>|t|)
(Intercept) -6.32155443  6.01018690 -1.051807 2.937609e-01
HScalc       0.70097203  0.08133028  8.618832 4.400339e-16
PreCalcScore 0.16848750  0.05181513  3.251704 1.282019e-03
CalcScore    0.08721742  0.04281987  2.036845 4.257051e-02
> sqrt(diag(MSE.XpXinv))
      (Intercept)      HScalc PreCalcScore      CalcScore
      6.01018690    0.08133028    0.05181513    0.04281987
```

```
> ##### Confidence intervals #####
```

$$\mathbf{a}'\hat{\boldsymbol{\beta}} \pm t_{\alpha/2} \sqrt{MSE \mathbf{a}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{a}}$$

```
> ##### Confidence intervals #####
> # Confidence interval for beta_j is just betahat_j plus or minus t_{alpha/2} * SE
> t.025 = qt(0.975,dfe); t.025
[1] 1.96815
> # If High School calculus is 10 marks higher, expected university calculus increases by
> # 10*beta1. Estimate the increase and give a 95% confidence interval.
> # 0.95 = P(A < beta1 < B) = P(10*A < 10*beta1 < 10*B)
> me95 = 10 * t.025*SE[2] # Number copied from the printout.
> cat("University calculus mark is expected to be",10*betahat[2],"points higher, plus or
minus",me95,"\n")
University calculus mark is expected to be 7.00972 points higher, plus or minus 1.600702

> # Confidence interval for beta3-beta2
> a = as.matrix(c(0,0,-1,1)) # Now a is different
> se = sqrt( t(a)%%MSE.XpXinv%%a ) # Standard error of the difference
> me95 = as.numeric( t.025*se ) # Now me95 is different
> estdiff = as.numeric( t(a) %% betahat ); estdiff
[1] -0.08127008
> Lower95 = estdiff - me95; Upper95 = estdiff + me95
> c(Lower95, Upper95)
[1] -0.23598624 0.07344609

> ##### General linear test #####
```

$$F^* = \frac{(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{t})'(\mathbf{C}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{C}')^{-1}(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{t})}{q \text{ MSE}} \stackrel{H_0}{\sim} F(q, n - k - 1)$$

```
> # Reproduce the overall test: F = 54.17
> # First do it the hard way
> C1 = rbind( c(0,1,0,0),
+             c(0,0,1,0),
+             c(0,0,0,1) )
> q = dim(C1)[1] # Number of rows
> F1 = t(C1%%betahat-0) %% solve(C1 %% MSE.XpXinv %% t(C1)) %%
+      (C1%%betahat-0) / q
> F1 # It's a 1x1 matrix
      [,1]
[1,] 54.16969
```

```

> # The easy way: Use this function if you like.
> source("http://www.utstat.utoronto.ca/~brunner/Rfunctions/ftest.txt")
> ftest # See the 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,C1) # Compare F = 54.17
      F      df1      df2      p-value
54.16969   3.00000 291.00000   0.00000

> # Test the two subtests simultaneously, controlling for HS Calculus mark
> C2 = rbind( c(0,0,1,0),
+             c(0,0,0,1) )
> ftest(fullmodel,C2)
      F      df1      df2      p-value
1.144444e+01 2.000000e+00 2.910000e+02 1.642604e-05

> round(ftest(fullmodel,C2),4)
      F      df1      df2      p-value
11.4444   2.0000 291.0000   0.0000

> # Do it with the full-reduced approach

```

$$F^* = \frac{SSR(full) - SSR(reduced)}{q \text{ MSE}}$$

```

> JustHScalc = lm(UnivCalc ~ HScalc) # The reduced (restricted) model
> anova(JustHScalc,fullmodel)

```

Analysis of Variance Table

Model 1: UnivCalc ~ HScalc  
 Model 2: UnivCalc ~ HScalc + PreCalcScore + CalcScore

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	293	62967				
2	291	58375	2	4591.5	11.444	1.643e-05 ***

---  
 Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## The SAT Data

In the United States, admission to university is based partly on high school marks and recommendations, and partly on applicants' performance on a standardized multiple choice test called the Scholastic Aptitude Test (SAT). The SAT has two sub-tests, Verbal and Math. A university administrator selected a random sample of 200 applicants, and recorded the Verbal SAT, the Math SAT and first-year university Grade Point Average (GPA) for each student. We seek to predict GPA from the two test scores. Throughout, we will use the  $\alpha = 0.05$  significance level.

1. First, fit a model using just the Math score as a predictor. “Fit” means estimate the model parameters.

```
> rm(list=ls())
> sat = read.table("http://www.utstat.toronto.edu/~brunner/data/legal/openSAT.data.txt")
> head(sat)
  VERBAL  MATH  GPA
1    578   567 2.68
2    474   653 2.51
3    546   657 1.95
4    664   686 2.81
5    600   619 2.79
6    488   738 2.36
> # Q1
> justmath = lm(GPA ~ MATH, data=sat); summary(justmath)
```

Call:

```
lm(formula = GPA ~ MATH, data = sat)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.92390	-0.38854	-0.00325	0.38448	1.43043

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.5272240	0.3981544	3.836	0.000168 ***
MATH	0.0016979	0.0006098	2.784	0.005885 **

---

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: 0.03768, Adjusted R-squared: 0.03282

F-statistic: 7.752 on 1 and 198 DF, p-value: 0.005885

Does there appear to be a relationship between Math score and grade point average?

- Answer Yes or No.
- Fill in the blank. Students who did better on the Math test tended to have \_\_\_\_\_ first-year grade point average.
- Do you reject  $H_0: \beta_1 = 0$ ? Answer Yes or No.
- Are the results statistically significant? Answer Yes or No.
- What is the  $p$ -value? The answer can be found in *two* places on your printout.
- What proportion of the variation in first-year grade point average is explained by score on the SAT Math test? The answer is a number from your printout.

g. Give a predicted first-year grade point average for a student who got 700 on the Math SAT. The answer is a number you could get with a calculator from the output of `summary`, but do it with R.

```
> betahat = coefficients(justmath); betahat
(Intercept)      MATH
1.527224026 0.001697934
> betahat[1]+700*betahat[2]
(Intercept)
2.715778
```

2. Now fit a model with both the Math and Verbal sub-tests.

```
>
> # Q2
> fullmodel = lm(GPA ~ VERBAL + MATH, data=sat); summary(fullmodel)

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

Residuals:
    Min       1Q   Median       3Q      Max
-1.70296 -0.36750  0.02644  0.38869  1.24830

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.6080747   0.4413074   1.378   0.170
VERBAL      0.0023070   0.0005521   4.178 4.41e-05 ***
MATH        0.0009974   0.0006095   1.636   0.103
---
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:  0.116,    Adjusted R-squared:  0.107
F-statistic: 12.93 on 2 and 197 DF,  p-value: 5.305e-06
```

a. Give the test statistic, the degrees of freedom and the  $p$ -value for each of the following null hypotheses. The answers are numbers from your printout.

- i)  $H_0: \beta_1 = \beta_2 = 0$
- ii)  $H_0: \beta_1 = 0$
- iii)  $H_0: \beta_2 = 0$
- iv)  $H_0: \beta_0 = 0$

b. Controlling for Verbal score, is Math score related to first-year grade point average?

- i) Give the value of the test statistic. The answer is a number from your printout.
- ii) Give the  $p$ -value. The answer is a number from your printout.
- iii) Do you reject the null hypothesis? Answer Yes or No.
- iv) Are the results statistically significant? Answer Yes or No.
- v) In plain, non-statistical language, what do you conclude? The answer is something about test scores and grade point average.

Repeating ...

```
> # Q2
>
> fullmodel = lm(GPA ~ VERBAL + MATH, data=sat); summary(fullmodel)
```

Call:

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

Residuals:

Min	1Q	Median	3Q	Max
-1.70296	-0.36750	0.02644	0.38869	1.24830

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.6080747	0.4413074	1.378	0.170
VERBAL	0.0023070	0.0005521	4.178	4.41e-05 ***
MATH	0.0009974	0.0006095	1.636	0.103

---  
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: 0.116, Adjusted R-squared: 0.107

F-statistic: 12.93 on 2 and 197 DF, p-value: 5.305e-06

c. Allowing for Math score, is Verbal score related to first-year grade point average?

i) Give the value of the test statistic. The answer is a number from your printout.

ii) Give the  $p$ -value. The answer is a number from your printout.

iii) Do you reject the null hypothesis? Answer Yes or No.

iv) Are the results statistically significant? Answer Yes or No.

v) In plain, non-statistical language, what do you conclude? The answer is something about test scores and grade point average.

vi) Do the test with the full-reduced model approach. Does  $F = t^2$ ? Compare the  $p$ -values.

```
> anova(justmath,fullmodel)
```

Analysis of Variance Table

Model 1: GPA ~ MATH

Model 2: GPA ~ VERBAL + MATH

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	198	64.486				
2	197	59.236	1	5.2496	17.458	4.411e-05 ***

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

d. Give a predicted first-year grade point average for a student who got 650 on the Verbal and 700 on the Math SAT.

```
> betahat = coefficients(fullmodel); betahat
(Intercept)      VERBAL      MATH
0.6080747411 0.0023070007 0.0009973607
> betahat[1] + 650*betahat[2] + 700*betahat[3]
(Intercept)
2.805778
```

e. Let's do one more test. We want to know whether expected GPA increases faster as a function of the Verbal SAT, or the Math SAT. That is, we want to compare the regression coefficients, by testing  $H_0: \beta_1 = \beta_2$ .

i) Do it with the general linear test. Feel free to use my `fctest` function.

```
> source("http://www.utstat.toronto.edu/~brunner/Rfunctions/fctest.txt")
> # To test H0: C beta = t. Input us model, C matrix, optional t (default zero)
> C = cbind(0,1,-1); C # One row
      [,1] [,2] [,3]
[1,]    0    1   -1
> fctest(fullmodel,C)
      F          df1          df2      p-value
1.9909873  1.0000000 197.0000000  0.1598149
```

ii) Do the same thing with the full-reduced model approach. Show your work.

Setting  $\beta_2 = \beta_1$ ,

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \epsilon_i \\ &= \beta_0 + \beta_1 x_{i,1} + \beta_1 x_{i,2} + \epsilon_i \\ &= \beta_0 + \beta_1 (x_{i,1} + x_{i,2}) + \epsilon_i \end{aligned}$$

So the restricted model is one with total score instead of separate scores for Math and Verbal.

```
> total = sat$VERBAL + sat$MATH
> sat = cbind(sat,total)
> reducedmodel = lm(GPA ~ total, data=sat)
>
> anova(reducedmodel,fullmodel)
```

Analysis of Variance Table

```
Model 1: GPA ~ total
Model 2: GPA ~ VERBAL + MATH
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1     198 59.835
2     197 59.236   1    0.59867 1.991 0.1598
```

iii) State your conclusion in plain, non-technical language. It's something about first-year grade point average.

---

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. The OpenOffice.org document is available from the course website:

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