llorban / **stats101**

Branch: master ▾ stats101 / live / **10-t-test.r** Find file Copy path

llorban Create 10-t-test.r 133fa13 19 days ago

1 contributor

48 lines (40 sloc) 1.5 KB

```r
exams=read.csv(url("http://llorban.net/psyc2300/1100_exam_grades.csv"))

### Appropriate t-test is the related-sample t-test because one student contribution two scores (Midterm 2 and Midterm 3)

### practical issues: finding out the variable names, creating an easy variable name
### for columns used in the analysis
head(exams)
exam2=exams$exam_2
exam3=exams$exam_3

### Assumption checking
### Check for normality
hist(exam2)
hist(exam3)

### Indendent-samples t (just for practice)
n2 = length(exam2)
n3 = length(exam3)
df2 = n2-1
df3 = n3-1
df_total = df2+df3
mean2 = sum(exam2)/n2
mean3 = sum(exam3)/n3
mean_diff = mean2-mean3
centered2 = exam2 - mean2
centered3 = exam3 - mean3
ss2 = sum(centered2^2)
ss3 = sum(centered3^2)
v2 = ss2/df2
v3 = ss3/df3
vp = ss2/df2 + ss3/df3  ### this was used in class, but it's wrong
vp = (ss2+ss3)/(df2+df3)  ### this should be used instead
se_p = sqrt(vp/n2 + vp/n3)
t = mean_diff/se_p
t

### Find proportion associated with t-value
### the pt() function looks up t-values and returns exact proportion values (p-values)
### the pt() function returns a one-tailed value so you need to multiple it by 2 to get the two-tailed result
### specify lower=FALSE to look for the upper tail of the distribution
2*pt(tt,df_total,lower=FALSE)

### Optional: Look up the critical t-value associated with the proportion at 95% two-tailed point, given your df total
qt(.025, df_total, lower=FALSE)

### Double check your work using the built-in example
t.test(exam2,exam3,paired=FALSE,var.equal=TRUE)
```