

## 15.6: Testing the Significance of a Correlation

### 15.6.1 Hypothesis tests for a single correlation

I don't want to spend too much time on this, but it's worth very briefly returning to the point I made earlier, that Pearson correlations are basically the same thing as linear regressions with only a single predictor added to the model. What this means is that the hypothesis tests that I just described in a regression context can also be applied to correlation coefficients. To see this, let's take a `summary()` of the `regression.1` model:

```
summary( regression.1 )
```

```
##
## Call:
## lm(formula = dan.grump ~ dan.sleep, data = parenthood)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.025  -2.213  -0.399   2.681  11.750
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 125.9563     3.0161  41.76  <2e-16 ***
## dan.sleep    -8.9368     0.4285 -20.85  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.332 on 98 degrees of freedom
## Multiple R-squared:  0.8161, Adjusted R-squared:  0.8142
## F-statistic: 434.9 on 1 and 98 DF,  p-value: < 2.2e-16
```

The important thing to note here is the t test associated with the predictor, in which we get a result of  $t(98) = -20.85$ ,  $p < .001$ . Now let's compare this to the output of a different function, which goes by the name of `cor.test()`. As you might expect, this function runs a hypothesis test to see if the observed correlation between two variables is significantly different from 0. Let's have a look:

```
cor.test( x = parenthood$dan.sleep, y = parenthood$dan.grump )
```

```
##
## Pearson's product-moment correlation
##
## data:  parenthood$dan.sleep and parenthood$dan.grump
## t = -20.854, df = 98, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9340614 -0.8594714
## sample estimates:
##           cor
## -0.903384
```

Again, the key thing to note is the line that reports the hypothesis test itself, which seems to be saying that  $t(98) = -20.85$ ,  $p < .001$ . Hm. Looks like it's exactly the same test, doesn't it? And that's exactly what it is. The test for the significance of a correlation is

identical to the t test that we run on a coefficient in a regression model.

### 15.6.2 Hypothesis tests for all pairwise correlations

Okay, one more digression before I return to regression properly. In the previous section I talked about the `cor.test()` function, which lets you run a hypothesis test on a single correlation. The `cor.test()` function is (obviously) an extension of the `cor()` function, which we talked about in Section 5.7. However, the `cor()` function isn't restricted to computing a single correlation: you can use it to compute *all* pairwise correlations among the variables in your data set. This leads people to the natural question: can the `cor.test()` function do the same thing? Can we use `cor.test()` to run hypothesis tests for all possible pairwise correlations among the variables in a data frame?

The answer is no, and there's a very good reason for this. Testing a single correlation is fine: if you've got some reason to be asking "is A related to B?", then you should absolutely run a test to see if there's a significant correlation. But if you've got variables A, B, C, D and E and you're thinking about testing the correlations among all possible pairs of these, a statistician would want to ask: what's your hypothesis? If you're in the position of wanting to test all possible pairs of variables, then you're pretty clearly on a fishing expedition, hunting around in search of significant effects when you don't actually have a clear research hypothesis in mind. This is *dangerous*, and the authors of `cor.test()` obviously felt that they didn't want to support that kind of behaviour.

On the other hand... a somewhat less hardline view might be to argue we've encountered this situation before, back in Section 14.5 when we talked about *post hoc* tests in ANOVA. When running post hoc tests, we didn't have any specific comparisons in mind, so what we did was apply a correction (e.g., Bonferroni, Holm, etc) in order to avoid the possibility of an inflated Type I error rate. From this perspective, it's okay to run hypothesis tests on all your pairwise correlations, but you must treat them as post hoc analyses, and if so you need to apply a correction for multiple comparisons. That's what the `correlate()` function in the `lsr` package does. When we use the `correlate()` function in Section 5.7 all it did was print out the correlation matrix. But you can get it to output the results of all the pairwise tests as well by specifying `test=TRUE`. Here's what happens with the `parenthood` data:

```
library(lsr)
```

```
## Warning: package 'lsr' was built under R version 3.5.2
```

```
correlate(parenthood, test=TRUE)
```

```
##
## CORRELATIONS
## =====
## - correlation type: pearson
## - correlations shown only when both variables are numeric
##
##          dan.sleep  baby.sleep  dan.grump  day
## dan.sleep      .          0.628***   -0.903*** -0.098
## baby.sleep     0.628***      .         -0.566*** -0.010
## dan.grump     -0.903***   -0.566***      .         0.076
## day           -0.098     -0.010      0.076      .
##
## ---
## Signif. codes: . = p < .1, * = p<.05, ** = p<.01, *** = p<.001
##
##
## p-VALUES
## =====
## - total number of tests run: 6
## - correction for multiple testing: holm
##
##          dan.sleep baby.sleep dan.grump  day
## dan.sleep      .          0.000      0.000 0.990
## baby.sleep     0.000      .          0.000 0.990
## dan.grump     0.000     0.000      . 0.990
## day           0.990     0.990     0.990 .
##
##
## SAMPLE SIZES
## =====
##
##          dan.sleep baby.sleep dan.grump day
## dan.sleep      100      100      100 100
## baby.sleep     100      100      100 100
## dan.grump     100      100      100 100
## day           100      100      100 100
```

The output here contains three matrices. First it prints out the correlation matrix. Second it prints out a matrix of p-values, using the Holm method<sup>218</sup> to correct for multiple comparisons. Finally, it prints out a matrix indicating the sample size (number of pairwise complete cases) that contributed to each correlation.

So there you have it. If you really desperately want to do pairwise hypothesis tests on your correlations, the `correlate()` function will let you do it. But please, **please** be careful. I can't count the number of times I've had a student panicking in my office because they've run these pairwise correlation tests, and they get one or two significant results that don't make any sense. For some reason, the moment people see those little significance stars appear, they feel compelled to throw away all common sense and assume that the results must correspond to something real that requires an explanation. In most such cases, my experience has been that the right answer is "it's a Type I error".

This page titled [15.6: Testing the Significance of a Correlation](#) is shared under a [CC BY-SA 4.0](#) license and was authored, remixed, and/or curated by [Danielle Navarro](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.