Lab 5

Creating and Using Our Own Statistical Functions for Correlational Testing
Psychology 310

Instructions. Work through the lab, and hand in your results as an Rmd file.

1 Introduction

This assignment involves creating and using your own statistical functions for performing correlational tests and confidence intervals. We have not discussed these procedures in class. They are discussed near the end of the *Unified Approach* handout starting on page 19, and in Cases 7,8,9 in the *Cases* handout. Each of these procedures is simple and straightforward, and is a special case of an asymptotically normal procedure (although one procedure is exactly distributed as Student's t. Your ability to study and construct statistical functions to execute procedures such as these on your own is an important final goal of your training in Psychology 310. I want you to create a small library of functions to perform the most basic standard correlational tests. These can be added to your library of R functions, and may very well prove useful on the last midterm!

For each of the 4 procedures discussed, write your own function, then demonstrate that, when applied to the sample function calls, you get the same output as I got from my functions, as shown below.

This *looks* at first glance like a long assignment. However, the 4 routines take a total of around 25 lines of code, and much of this handout is simply demonstration output.

2 One Sample Hypothesis Test that $\rho = 0$

The one sample test that $\rho = 0$ based on a sample of size n from a bivariate normal distribution uses a t-statistic with $n - 2$ degrees of freedom.

$$t_{n-2} = \sqrt{n-2}\ \frac{r}{\sqrt{(1-r^2)}} \tag{1}$$

Your function should be written so that a call to it will be in the following form:

`OneSampleCorrelationTest(r,n,digits = 4)`

Assume no missing data is this routine. `digits` is an optional parameter to specify the number of digits printed in the correlation and p-value. Default is 4.

Here are some example function calls

```
> OneSampleCorrelationTTest(r = 0.2, n = 100)

$correlation
```

```
[1] 0.2

$t.observed
[1] 2.021

$df
[1] 98

$p.value
[1] 0.046

> OneSampleCorrelationTTest(r = -0.196, n = 102)

$correlation
[1] -0.196

$t.observed
[1] -1.999

$df
[1] 100

$p.value
[1] 0.0483
```

Here is an example of how to use the function with raw data.

```
> x <- c(1, 3, 4, 5, 2)
> y <- c(1, 2, 3, 4, 5)
> OneSampleCorrelationTTest(cor(x, y), length(x))

$correlation
[1] 0.4

$t.observed
[1] 0.7559

$df
[1] 3

$p.value
[1] 0.5046
```

3 One Sample Test that $\rho = \rho_0$

The classic "Fisher Z test" uses the *Fisher transform*, a nonlinear transform that changes the rather skewed distribution of the sample correlation so that it is almost perfectly normal with a variance closely approximated by $1/(n-3)$. The Fisher transform was traditionally written in the form

$$\phi(r) = \ln \frac{1+r}{1-r} \tag{2}$$

where ln is the natural logarithm (to the base e).

The Fisher transform is actually the inverse hyperbolic tangent $\tanh^{-1}(r)$, and may be evaluated directly in R as `atanh(r)`. This turns out to be extremely convenient, because this function is invertible, and the inverse Fisher transform, needed to construct a confidence interval on an unknown population correlation ρ, is simply the hyperbolic tangent $\tanh(r)$, computed in R as `tanh(r)`.

The Fisher Z statistic for the hypothesis $\rho = \rho_0$ is simply

$$Z = \frac{\tanh^{-1}(r) - \tanh^{-1}(\rho_0)}{\sqrt{1/(n-3)}} \tag{3}$$

$$= \sqrt{n-3}\left(\tanh^{-1}(r) - \tanh^{-1}(\rho_0)\right) \tag{4}$$

Construct a routine for a one sample test that $\rho = \rho_0$. Your function should take all the same input as before, except that:

- There is an additional parameter ρ_0. For simplicity, we'll again assume your test is two-sided.
- If $\rho_0 = 0$, your routine should perform the Z test anyway, even though the t test is exact under normality assumptions and the Z test is not.

Here are some sample function calls:

```
> FisherZTest(r = 0.196, n = 102, rho0 = 0)

$correlation
[1] 0.196

$null.hypothesized.correlation
[1] 0

$z.statistic
[1] 1.976

$p.value
[1] 0.0482

> FisherZTest(r = -0.196, n = 100, rho0 = 0)

$correlation
[1] -0.196

$null.hypothesized.correlation
[1] 0

$z.statistic
[1] -1.956

$p.value
[1] 0.0505

> x <- c(1, 3, 4, 5, 2)
> y <- c(1, 2, 3, 4, 5)
> FisherZTest(r = cor(x, y), n = length(x), rho0 = 0)
```

```
$correlation
[1] 0.4

$null.hypothesized.correlation
[1] 0

$z.statistic
[1] 0.5991

$p.value
[1] 0.5491

> FisherZTest(r = cor(x, y), n = length(x), rho0 = 0.3)

$correlation
[1] 0.4

$null.hypothesized.correlation
[1] 0.3

$z.statistic
[1] 0.1614

$p.value
[1] 0.8718
```

4 Confidence Interval on a Single Correlation

Case 7 handout describes a two step procedure for computing a confidence interval on a single correlation. It proceeds in 3 steps.

1. Compute a critical Z value for your level of confidence. This is easy, because for confidence level C, $\alpha = 1 - C$, and the critical value is always `qnorm(1 - alpha/2)`, or, more directly `qnorm(1 - (1-C)/2)`.
2. Compute a confidence interval for the Fisher transform of ρ.
3. Transform the endpoints of this confidence interval using the inverse Fisher transform.

I want you to write a function that computes the confidence interval on a single correlation. The input should include the correlation, the sample size, the confidence level, and, optionally, the number of digits for rounding the output. Call the confidence level `conf`, and give it a default value of 0.95.

Here are some sample function calls:

```
> FisherCI(r = 0.196, n = 102, conf = 0.95)

$lower.limit
[1] 0.0016
```

```
$upper.limit
[1] 0.3761

$confidence.level
[1] 0.95

> FisherCI(r = -0.196, n = 100)

$lower.limit
[1] -0.3779

$upper.limit
[1] 4e-04

$confidence.level
[1] 0.95

> x <- c(1, 3, 4, 5, 2)
> y <- c(1, 2, 3, 4, 5)
> FisherCI(r = cor(x, y), n = length(x))

$lower.limit
[1] -0.7453

$upper.limit
[1] 0.9478

$confidence.level
[1] 0.95

> FisherCI(r = cor(x, y), n = length(x), conf = 0.9)

$lower.limit
[1] -0.6288

$upper.limit
[1] 0.9196

$confidence.level
[1] 0.9
```

5 Two-Sample Fisher Z for Comparing Two Independent Correlations

This is discussed in the *Unified Approach* and *Case 9* handouts. For two sample correlations r_1, r_2 based on sample sizes n_1 n_2, the test statistic is computed as

$$Z = \frac{\tanh^{-1}(r_1) - \tanh^{-1}(r_2)}{\sqrt{1/(n_1 - 3) + 1/(n_2 - 3)}} \tag{5}$$

Here are some sample function calls. The first one is the example from Case 9, and demonstrates that there is some minor rounding error in the Case 9 calculation due to its truncation at 3 decimal places.

Both these examples demonstrate, indirectly, the low power of correlational significance tests when performed with the kind of sample sizes used in the sample calculations.

```
> TwoSampleFisherZ(r1 = 0.59, n1 = 48, r2 = 0.36, n2 = 92)

$z.statistic
[1] 1.644

$p.value
[1] 0.1001

> TwoSampleFisherZ(r1 = 0.35, n1 = 40, r2 = 0.1, n2 = 40)

$z.statistic
[1] 1.14

$p.value
[1] 0.2542
```