Problem Set 5

Shane Mueller

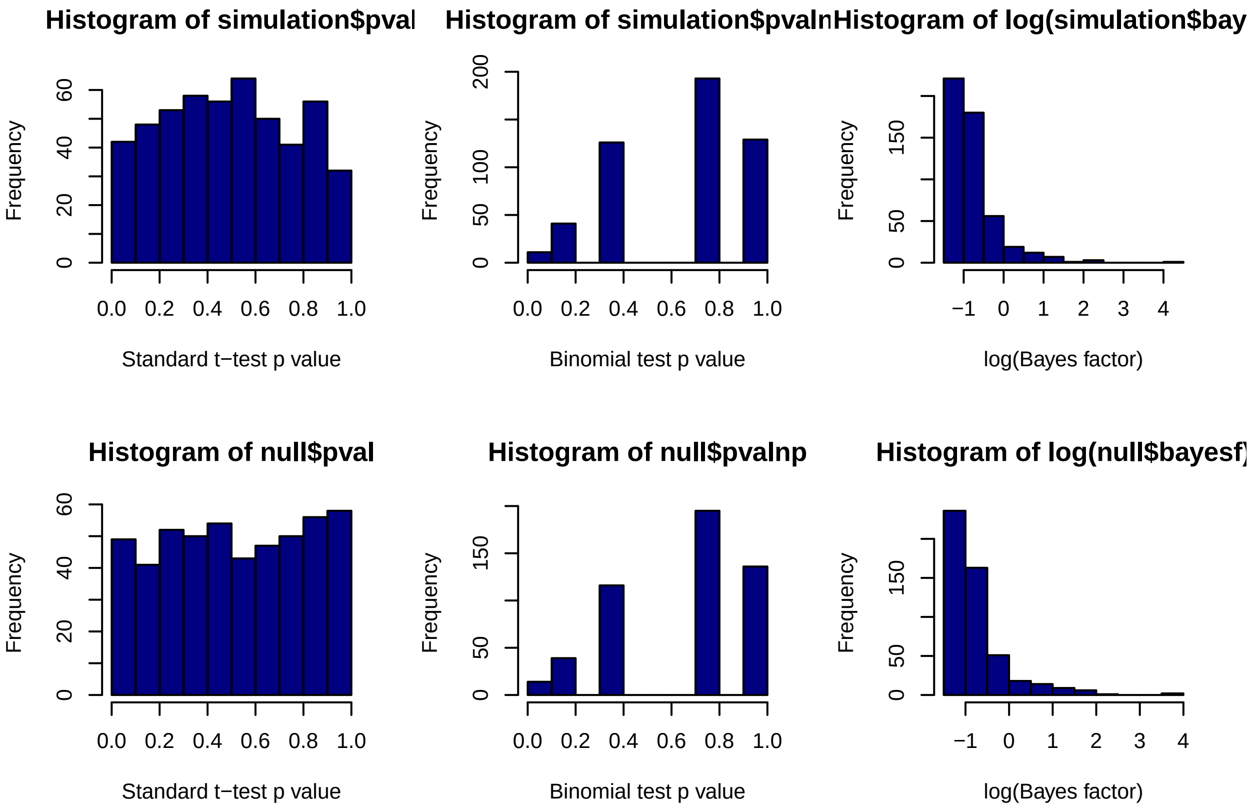
Problem 1. Understanding the power of a test.

When we do a NHST test, we look at the p-value–which is the chance that the results could have arisen if the null hypothesis were true. Let's suppose the null hypothesis is false. The data come from a normal distribution with sd = 1 and mean +.1, this has a relatively small effect size (.1), but the truth is that the mean is different from 0. In this problem, we want to test, via simulation, how likely you are to detect this difference using the three methods discussed in the book for one-sample tests (t-test, binomial test, and bayes factor test). The following code will run a 500 simulated studies with 10 trials per study. It does one study in which the null is true, and a second in which the alternative is true, and looks at the proportion of these that the statistical test determines are different. In this case, we will use p=.05, and bayes factor = 2, as reasonable criterion for calling a difference statistically significant. A more conservative criterion would use p=.01 and bf=3.

```
set.seed(100)
##this function generates n data points extracts the p/value
##bayes factor of the one-sample two-sided test for each:
simdata1 <- function(n,mean=.1)
{
  data <- rnorm(n,mean=mean)

  c( t.test(data)$p.value,
    binom.test(sum(data>0),n)$p.value,
    exp(ttestBF(data)@bayesFactor$bf))
  ##exponentiate because bf is stored as a log and so 0 is unbiased.

}

runs <- 500
##this simulates 1000 experiments:
null <- data.frame(pval=rep(NA,runs),
                           pvalnp=rep(NA,runs),
                           bayesf=rep(NA,runs))

##this simulates 1000 experiments:
simulation <- data.frame(pval=rep(NA,runs),
                           pvalnp=rep(NA,runs),
                           bayesf=rep(NA,runs))
n<-10   ##this is how many samples are drawn in each experiment
for(i in 1:runs)
{
  simulation[i,] <- simdata1(n)
  null[i,] <- simdata1(n,0)
}
```

```
## This determines how many of the cases produce a significant result.
##
par(mfrow=c(2,3))
hist(simulation$pval,col="navy",xlab="Standard t-test p value")
hist(simulation$pvalnp,col="navy",xlab="Binomial test p value")
```

```r
hist(log(simulation$bayesf),col="navy", xlab="log(Bayes factor)")
hist(null$pval,col="navy",xlab="Standard t-test p value")
hist(null$pvalnp,col="navy",xlab="Binomial test p value")
hist(log(null$bayesf),col="navy", xlab="log(Bayes factor)")
```



Examine the number of significant tests when there is a true difference

```r
mean(simulation$pval <.05)
```

```
## [1] 0.048
```

```r
mean(simulation$pvalnp < .05)
```

```
## [1] 0.022
```

```r
mean(simulation$bayesf > 2)
```

```
## [1] 0.038
```

```r
mean(simulation$bayesf< (-2))
```

```
## [1] 0
```

Examine number of significant tests under the null hypothesis:

```r
mean(null$pval <.05)
```

```
## [1] 0.064
```

```r
mean(null$pvalnp < .05)
```

```
## [1] 0.028
```

```
mean(null$bayesf   >2 )  #evidence for the alternative
```

```
## [1] 0.056
```

```
mean(null$bayesf < (-2)) ##evidence for the null?
```

```
## [1] 0
```

For n=10, we can see that about 5% of the cases are found to be significant in the standard t-test. In this case, it is almost identical to the number that are found to be significant in the null distribution (we'd expect 5% to occur just by chance if there were no difference.). But the number we find in the binomial test is lower in both cases, and the number of bayes factors > 2 is about the same as well. Also note that we can test the evidence against the alternative (b>2) and the evidence in support of the null (b<(-2)), and we see that although the Bayes factor test only finds a few cases that are significant, it is otherwise typically in an ambivalent state–it never finds support for the null hypothesis either.

The probability of finding an effect when it exists is the power of the test. For a sample size of 10 and an effect size of .1, we have almost no power, with any of the tests. An acceptable power is usually 0.8, and we could gain power by accepting more false alarms (using a less stringent criterion), or by collecting larger numbers of samples, or by obtaining better experimental control and measuring an effect that has a larger effect size.

For this problem, repeat the above simulation for a range of values of n (at least 5 values). Create a table with a row for each n, and the power for each of the three tests as you change n (i.e., the proportion of true differences the test detected), as well as the false alarm rate for each test. A power of .8 is considered acceptably large when planning a study. Identify roughly the n needed for a power of 0.8 in each of the three tests. Discuss the relative advantages and disadvantages of the three tests with respect to power and false alarm (incorrectly rejecting the null).

Problem 2: t-tests

For the data below, compute a one-tailed t-test by hand to determine whether the average is reliably greater than 100. That is, compute the mean, standard deviation, standard error, t value, and the corresponding p value. To compute the t value for this comparison, you need to compute the mean and subtract 100, and then divide that by the standard error. Verify you did this correctly by using the t.test() function to produce the same values. Then, re-run your analysis by removing the 245 value, which might seem like it is the strongest piece of evidence for your data being greater than 100.

```
data <- c(101,103,99,92,110,105,103,102,104,106,101,
          101,101,102,103,101,99,104,105,102,102,103,
          245,103,107,101,103,108,104,101,101,102)
```

Problem 3. Wilcox test

Generate a set of 20 random normal numbers, and compare them to their sorted values with both an independent samples wilcox test, and a paired wilcox test. What value do you get for W for the independent test? Why do the p values differ?

```
x <- rnorm(20)
y <- sort(x)
```

Problem 4. Comparing t-test, wilcox, and Bayes factor t-test

Generate a set of 150 normal random numbers whose means differ by .3 standard deviation units from another set, but that are uncorrelated to the first set, as in the example below. Run one-sided tests compariing these, including standard t-test, a wilcox test, and a BayesFactor t-test. Do the results differ for the different tests? Repeat several times to determine whether there is a pattern.

```r
library(BayesFactor)
x <- rnorm(150)
y <- rnorm(150) + .3
```

Problem 5. Robustness to transforms

For each of the data sets you created in the previous problem, exponen- tiate your samples using the exp function (e.g., exp(x) and exp(y)), and plot the distributions using hist. Then do the same three tests you did originally (t-test, wilcox test, and bayes factor test) on these two exponentiated data sets. How do this test compare to the previous test? Discuss why you see the differences and similarities with Problem 4.

Problem 6. Comparison to paired tests.

For the original data (not the exponentiated data), run a paired t-test, as well as a paired wilcox test, and a paired BayesFactor test (even though the data were not paired).

Discuss what the findings were and how they compare to the unpaired versions of the tests.

Problem 7: Correlations

Suppose you have a true correlation of .3 between two variables, created by mixing one uniform with a second like this:

```r
x <- runif(100)
y <- runif(100)

z <- x + .5*y
z1 <- z
z2 <- z+10
z3 <- log(z)
z4 <- z*10
z5 <- z + runif(100)
```

Compute pearson and spearman correlations between x and each of the z variables z1 through z5. Identify the statistical significance of the correlations, and report a Bayes factor for each correlation as well. Examine the pearson and spearman correlations, and discuss why for some transforms, these are unchanged, but for other transforms they are changed.