StatR 301: Spring 2013

Homework 01

Rod Doe

Thursday, April 18, 2013

Parametric and non-parametric numerical p-values

Key point: What exactly do we mean by a p-value?

Definition (from Dobson 12.1.1):

A p-value it the probability of observing more extreme data (if the random process were repeated) given that the null hypothesis is correct.

The Challenger dataset records the count of eroded and intact O-ring seals (a total of six) as a function of temperature. Here is a sample of the relevant data:

```
> head(challenger)
```

Event	atrisk	Eroded	Intact	tempF
1	6	0	6	66
2	6	1	5	70

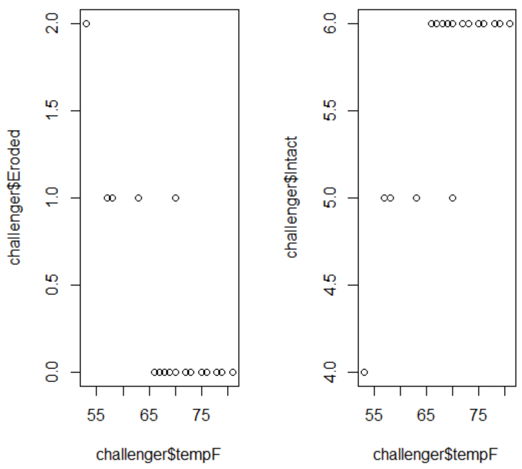
Here is my null hypothesis on the relationship between(Eroded,Intact) and temperature:

Null hypothesis: There is no relationship between (Eroded,Intact) and temperature.

Alternative hypothesis: There is a relationship between (Eroded,Intact) and temperature.

Plots of the data suggest a positive relationship between seal efficacy and temperature, in which the seals are more likely to be compromised by low temperature:

```
par(mfcol = c(1,2))
plot(challenger$tempF, challenger$Eroded)
plot(challenger$tempF, challenger$Intact)
```



Standard regression p-value for the effect of temperature in the Challenger data set.

```
> summary.lm(fit)

Call:
glm(formula = cbind(Eroded, Intact) ~ tempF, family = binomial,
    data = challenger)

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-0.5385 -0.3938 -0.2401 -0.1539  2.3454

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.81692    2.93149   3.008  0.00670 **
tempF       -0.17949    0.04732  -3.794  0.00106 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8127 on 21 degrees of freedom
Multiple R-squared: 0.005942,   Adjusted R-squared: -0.04139
F-statistic: 0.1255 on 1 and 21 DF,  p-value: 0.7266
```

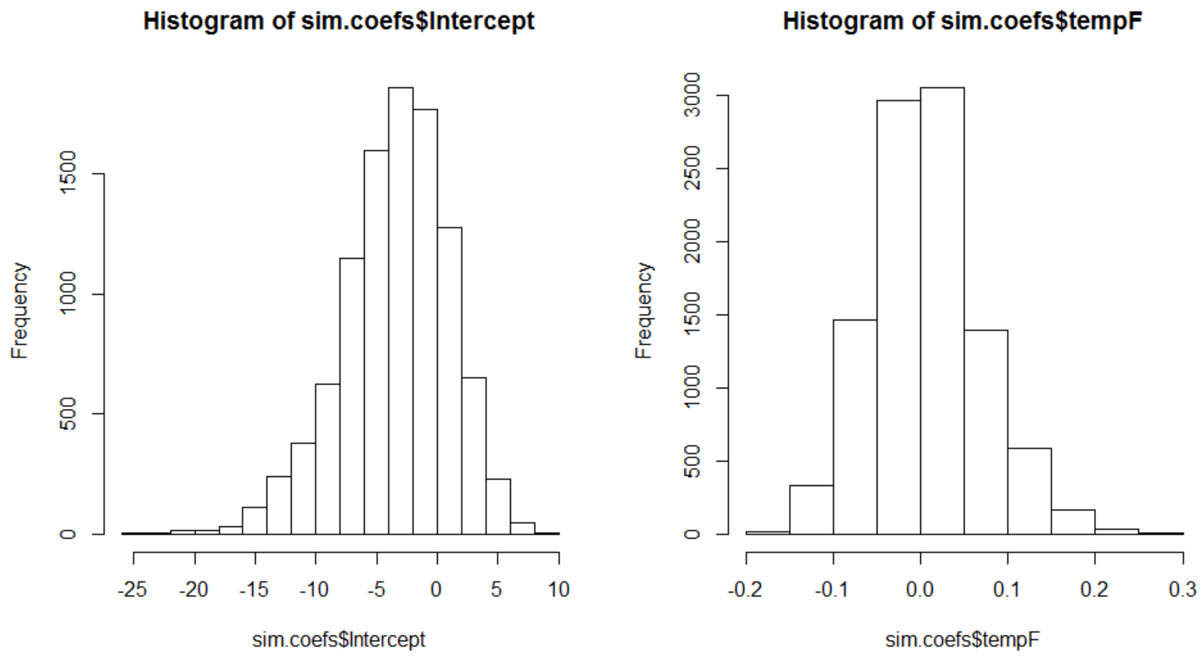
Numerical p-value for the effect of temperature in the Challenger data set

The null hypothesis is that there is no relationship between seal efficacy and temperature. To test this, model the seal efficacy as a function of random values of temperatures taken from the data set. To do this, exploit the fact that the R sample function simply scrambles data when invoked with no size parameter. This is from the help page for the sample function:

For `sample` the default for `size` is the number of items inferred from the first argument, so that `sample(x)` generates a random permutation of the elements of `x` (or `1:x`).

```
nRows = 10000
sim.coefs <- data.frame(Intercept = numeric(), tempF=numeric())

for (I in 1:nRows) {
  challenger$permT = sample(challenger$tempF)
  fit.loop = glm( cbind(Eroded,Intact) ~ permT, data=challenger,
family=binomial )
  coef.loop = coef(fit.loop)
  df.loop = data.frame(Intercept=coef.loop[1], tempF=coef.loop[2])
  sim.coefs = rbind(sim.coefs, df.loop)
}
```



The numeric p-value is the proportion of runs with estimates as extreme as the true one or more.

The parametric coefficient for temperature was -0.17949. In the simulated data, this only occurred once:

```
length(which(sim.coefs$tempF <= -0.17949))
[1]  1
```

As such, I conclude that my numerical p-value is 1/10000. Quite low…

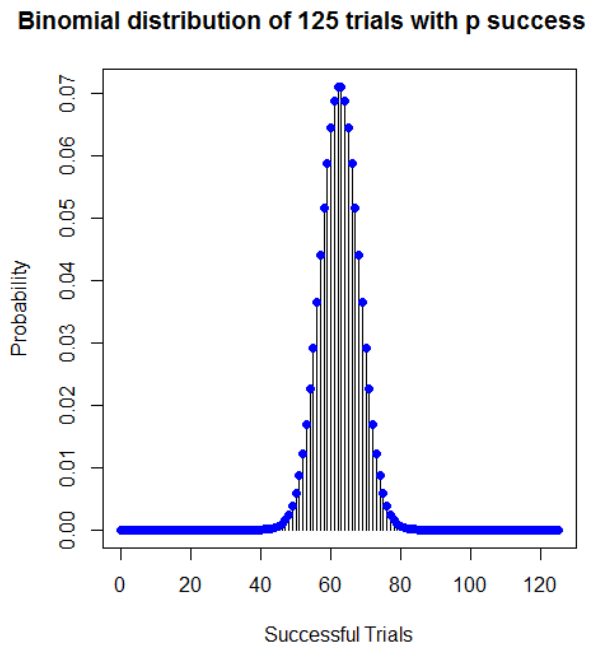
Estimator Performance Simulation – Bayesian versus Frequentist

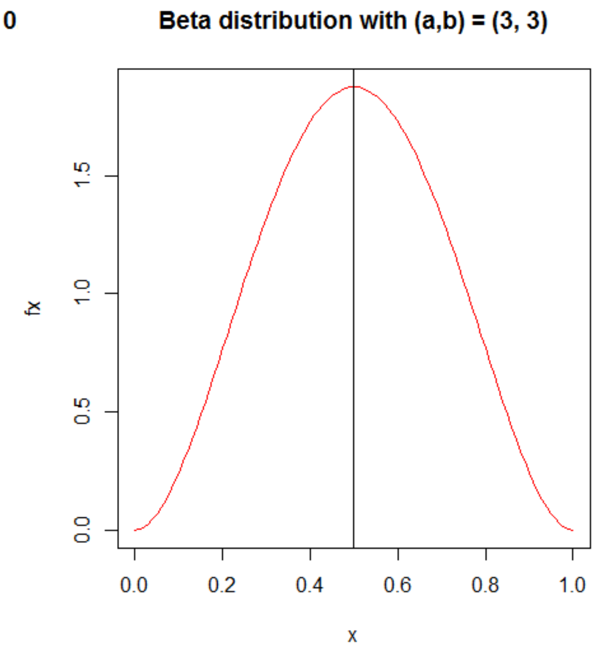
This raised more questions than answers. I get that the general idea here is to start with a prior, in this case a Beta distribution, and (somehow) apply an observed likelihood to it (in this case, a binomial distribution. We then sample? I'm not sure… Maybe we're not ready for this question yet.

```
# Beta prior symmetric around p=0.5, whose weight is equivalent to a+ß=6 observations
# Need a beta distribution symmetric around 0.5 with alpha + beta = 6.  So, what are alpha and beta?
# "symmetric about 0.5" may imply that they are equal...
x <- seq(0, 1, 0.01)
fx <- dbeta(x, 3, 3)
plot(x, fx, type="l", main="Beta distribution with (a,b) = (3,3)", col="red")
abline(v=0.5)
```

This is the distribution to be used for the prior and one of the likeliehoods:

I refactored a binomial distribution plotting function I had for depicting baseball batting averages. It helped to answer the question, "What is the chance that he can hit 300 given that he's historically been a 250 hitter?" Bit I digress…





Add some general functionality:

```
x <- seq(0, 1, 0.01)
fx <- dbeta(x, 3, 3)

plot(x, fx, type="l", main="Beta distribution with (a,b) = (3,3)",
col="red")
abline(v=0.5)  # good guess...

PlotBetaDist <- function(alpha, beta)
{
        x <- seq(0, 1, 0.01)
        fx <- dbeta(x, alpha, beta)
        plot(x, fx, type="l",
                main=sprintf("Beta distribution with (a,b) = (%d, %d)",
alpha, beta),
                col="red")
        abline(v=0.5)
}

# Will need to plot binomial dists.

PlotBinomDist <- function(size, p)
{
        plot(0:size, dbinom(0:size, size, p),
                xlab="Successful Trials",
           ylab="Probability", type="h",
           main=sprintf("Binomial distribution of %d trials with p
success = %0.2f", size, p) )

           points(0:size, dbinom(0:size, size, p), pch=19, col="blue")
}


PlotBinomDist(5, 0.5);
PlotBinomDist(25, 0.5);
PlotBinomDist(125, 0.5);

alpha = 3
beta = 3
# Create the prior
prior.x <- seq(0, 1, 0.01)
prior.fx <- dbeta(x, alpha, beta)

# The likelihood is the binomial dist.

# Create the posterior.
# Posterior ~ prior * likelihood
 PlotBetaDist(3,3)

# I don't understand what's going on here...
> pvec=seq(1/1001,1000/1001,1/1001) ### equally-spaced theta vector
> priorvec=dbeta(pvec,3,3)  ### prior density
> posteriorvec=priorvec*pvec^4  ### posterior ~ prior*likelihood  ##
Why ^4
> posteriorvec=1000*posteriorvec/sum(posteriorvec) # Normalizing   ##
Why 1000?
> plot(pvec,posteriorvec,main="4 Coin Posterior, Numerically",   ##
Why 4?
+      xlab='p',ylab='Posterior Density')
> lines(pvec,dbeta(pvec,7,3),col=2)  ## Why 7,3?
```