

XDAS Week 6 Recitation

Chris Jackson

10/9/2019

Hypothesis Testing with random sampling and t -tests

This exercise tests what percentage of times a t -test returns a p -value less than the critical threshold.

We take $n = 1000$ sets of random samples taken from two normal distributions: a standard normal, $\mathcal{N} \sim (0, 1)$, and a second normal distribution also with standard deviation $\sigma = 1$.

We examine what happens when we vary:

- The sample size: $N = 3, 10, 20, 30, 50, 100$
- The (true) separation between parent distributions: $\Delta = 0, 0.25, 0.5, 1, 1.5, 2$ (in units of standard deviations)
- The significance threshold: $\alpha = 0.05, 0.01$

The results demonstrate our **power** to detect true differences between distributions, given varying **sample size** and **effect size**.

Let's go ahead and simulate two data sets using `rnorm`.

```
data.set.1 <- rnorm(1000, mean=0.5, sd=1)
data.set.2 <- rnorm(1000, mean=1, sd=1)
```

Are the means of these data sets different by t -test?

```
t.test(data.set.1, data.set.2)

##
##  Welch Two Sample t-test
##
## data:  data.set.1 and data.set.2
## t = -12.743, df = 1997.5, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   -0.6400221 -0.4693009
## sample estimates:
## mean of x mean of y
##  0.4750837  1.0297452
```

Awesome - the thing we defined as being different is in fact, different. But this is a lot of observations (1000). What if we have fewer?

```
data.set.1 <- rnorm(10, mean=0.5, sd=1)
data.set.2 <- rnorm(10, mean=1, sd=1)
```

Are the means of these data sets different by t -test?

```
test.vals <- t.test(data.set.1, data.set.2)
test.vals
```

```
##
##  Welch Two Sample t-test
```

```
##
## data: data.set.1 and data.set.2
## t = 0.19631, df = 17.964, p-value = 0.8466
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.7806204 0.9415157
## sample estimates:
## mean of x mean of y
## 0.9649719 0.8845243
```

Well that's not nearly as awesome. Even though we KNOW that the means of these two populations is different (we defined it that way!), the test has failed to demonstrate a statistically significant difference. How often does that occur?

```
# Make a function that tests normal data num.t times
# The number of observations per sampling is num.o
# The means of these normal distributions are different
# The standard deviation is the same though

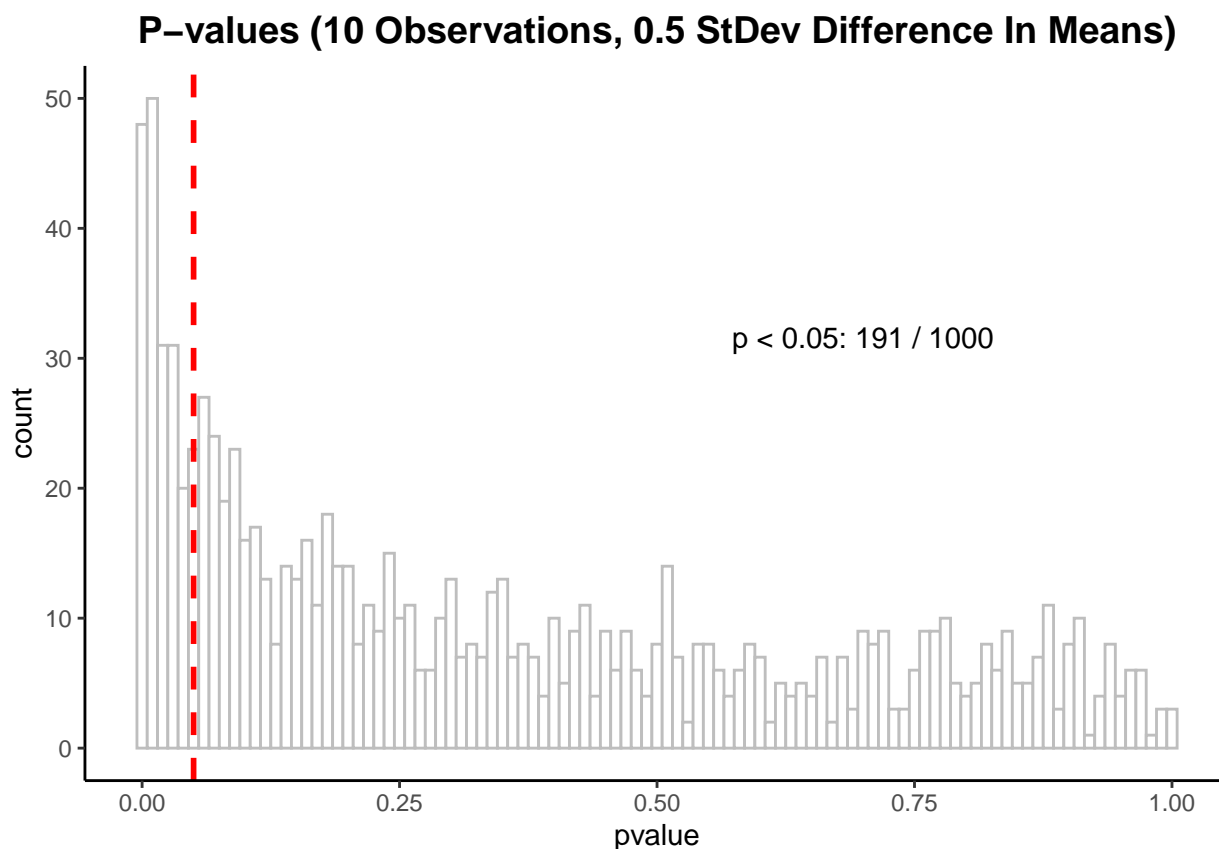
run.ttest.function <- function(num.t, num.o, sd.diff) {
  test.vals <- NULL
  for (i in 1:num.t) {
    loop.set.1 <- rnorm(num.o, mean=0, sd=1)
    loop.set.2 <- rnorm(num.o, mean=sd.diff, sd=1)
    loop.vals <- t.test(loop.set.1, loop.set.2)
    test.vals <- rbind(test.vals,
                      data.frame(pvalue=loop.vals$p.value,
                                diff=0.5, n=10))
  }
  return(test.vals)
}

n.tests <- 1000
n.observations <- 10
stdev.difference <- 0.5

n10.diff0.5 <- run.ttest.function(n.tests, n.observations, stdev.difference)

plot.title <- paste0("P-values (", toString(n.observations),
                    " Observations, ", toString(stdev.difference),
                    " StDev Difference In Means)")
sig.by.test <- sum(n10.diff0.5$pvalue < 0.05)

ggplot(n10.diff0.5, aes(x=pvalue)) +
  theme_classic() +
  labs(title=plot.title) +
  theme(plot.title = element_text(hjust=0.5, size=14, face="bold")) +
  geom_histogram(binwidth = 0.01, color="grey", fill="white") +
  geom_vline(xintercept = 0.05, color = "red", linetype = 'dashed', size=1) +
  annotate("text", label=paste("p < 0.05:", toString(sig.by.test), "/",
                              toString(n.tests)), x=0.7, y=sqrt(n.tests))
```



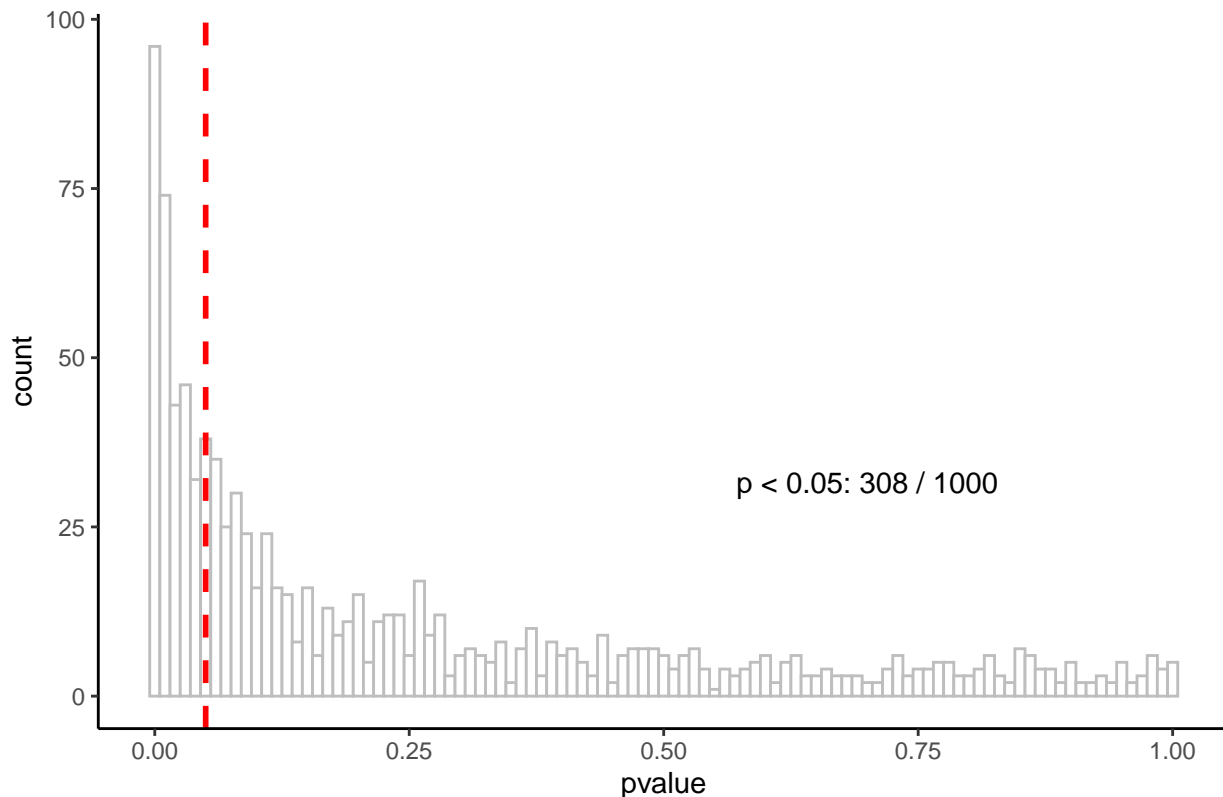
Interesting. About 15% of the time we decide on statistical significance for when $n=10$ observations and a half standard deviation. How does this change with increasing the number of observations?

```
n.observations <- 20
n20.diff0.5 <- run.ttest.function(n.tests, n.observations, stdev.difference)

plot.title <- paste0("P-values (", toString(n.observations),
                     " Observations, ", toString(stdev.difference),
                     " StDev Difference In Means)")
sig.by.test <- sum(n20.diff0.5$pvalue < 0.05)

ggplot(n20.diff0.5, aes(x=pvalue)) +
  theme_classic() +
  labs(title=plot.title) +
  theme(plot.title = element_text(hjust=0.5, size=14, face="bold")) +
  geom_histogram(binwidth = 0.01, color="grey", fill="white") +
  geom_vline(xintercept = 0.05, color = "red", linetype = 'dashed', size=1) +
  annotate("text", label=paste("p < 0.05:", toString(sig.by.test), "/",
                              toString(n.tests)), x=0.7, y=sqrt(n.tests))
```

P-values (20 Observations, 0.5 StDev Difference In Means)



Doubling the number of observations doubles the number of ‘successful’ tests. How well does that hold up?

```
## KCG solution (p=0.05 exercise)
n.tests <- 1000
n.obs <- c(10,20,30,50,100)
sd.diff <- c(0,0.25,0.5,1,1.5,2)
pval <- 0.05

results = matrix(nrow = length(n.obs),
                 ncol = length(sd.diff))

for (i in 1:length(n.obs)) {
  for (j in 1:length(sd.diff)) {
    n.diff <- run.ttest.function(n.tests, n.obs[i], sd.diff[j])
    results[i,j] = sum(n.diff$pvalue < pval) / n.tests
  }
}
dimnames(results) = list(as.character(n.obs),
                        as.character(sd.diff))

paste0("Results for p-value=",pval)

## [1] "Results for p-value=0.05"

results

##      0  0.25  0.5   1  1.5   2
## 10  0.052 0.086 0.190 0.566 0.890 0.985
## 20  0.043 0.117 0.373 0.868 0.996 1.000
```

```
## 30  0.057 0.148 0.492 0.958 1.000 1.000
## 50  0.043 0.245 0.707 1.000 1.000 1.000
## 100 0.055 0.411 0.943 1.000 1.000 1.000

## Yingzhen's solution (p=0.01 exercise)
n.tests <- 1000
n.observations <- c(3,10,20,30,50,100)
stdev.difference <- c(0,0.25,0.5,1,1.5,2)
p_gating = 0.01
results = matrix(1:length(n.observations)*length(stdev.difference),
                 nrow = length(n.observations),
                 ncol = length(stdev.difference))

for(i in 1:length(n.observations)){
  for(j in 1:length(stdev.difference)){
    temp_result = run.ttest.function(n.tests, n.observations[i], stdev.difference[j])
    results[i,j] = sum(temp_result$pvalue < p_gating) / n.tests
  }
}
rownames(results) = as.character(n.observations)
colnames(results) = as.character(stdev.difference)

paste0("Results for p-value=",p_gating)

## [1] "Results for p-value=0.01"

results

##           0  0.25  0.5    1   1.5    2
## 3    0.007 0.008 0.016 0.018 0.044 0.090
## 10   0.008 0.030 0.061 0.269 0.695 0.924
## 20   0.014 0.039 0.147 0.662 0.962 1.000
## 30   0.015 0.042 0.244 0.887 0.999 1.000
## 50   0.012 0.085 0.439 0.994 1.000 1.000
## 100  0.008 0.208 0.827 1.000 1.000 1.000
```