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: Δ = 0,0.25,0.5,1,1.5,2 (in units of standard deviations)
- The significance threshold: α = 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 = -10.659, df = 1997.6, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5792844 -0.3992387
## sample estimates:
## mean of x mean of y
## 0.5020302 0.9912917
```

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?

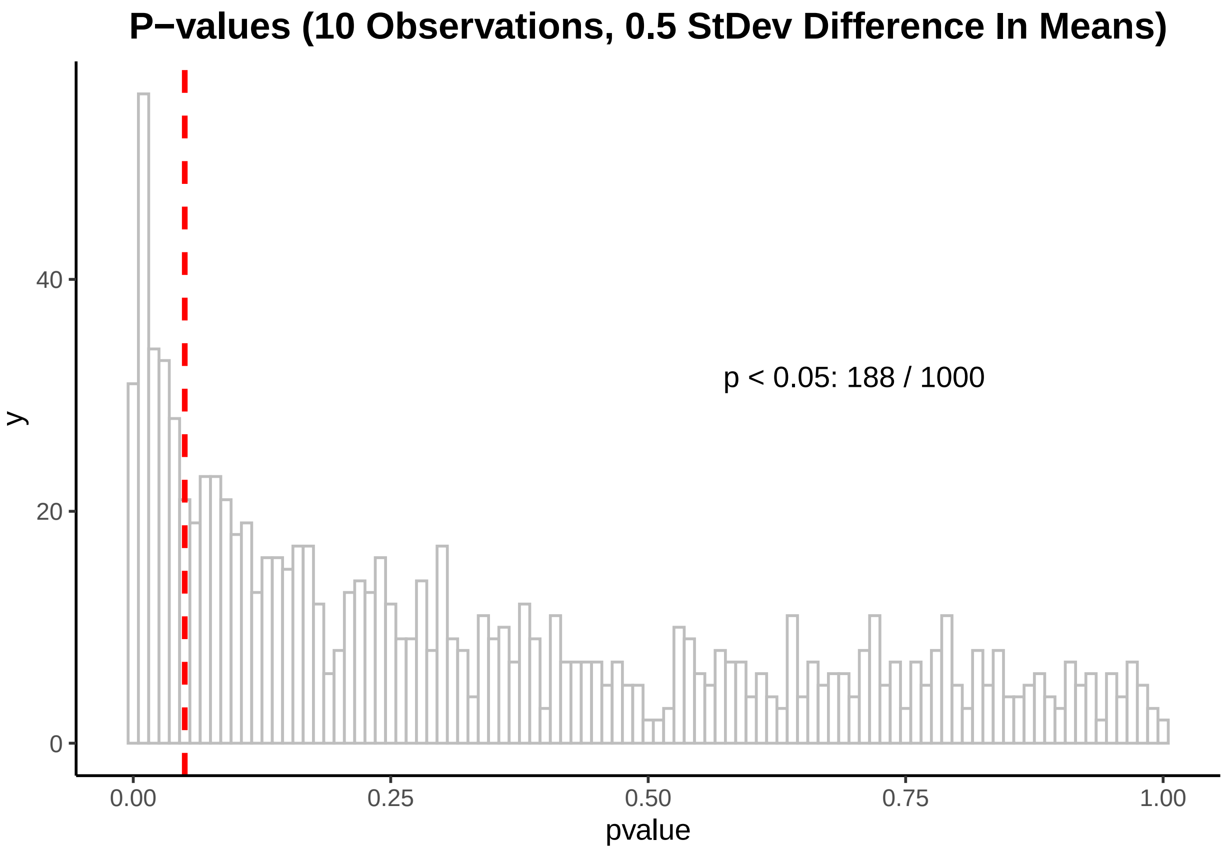
```
test.vals <- t.test(data.set.1, data.set.2)
test.vals
```

```
##
##  Welch Two Sample t-test
##
## data:  data.set.1 and data.set.2
## t = -1.9091, df = 13.426, p-value = 0.07786
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.7396311  0.1046345
## sample estimates:
## mean of x mean of y
## 0.5371502 1.3546485
```

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?

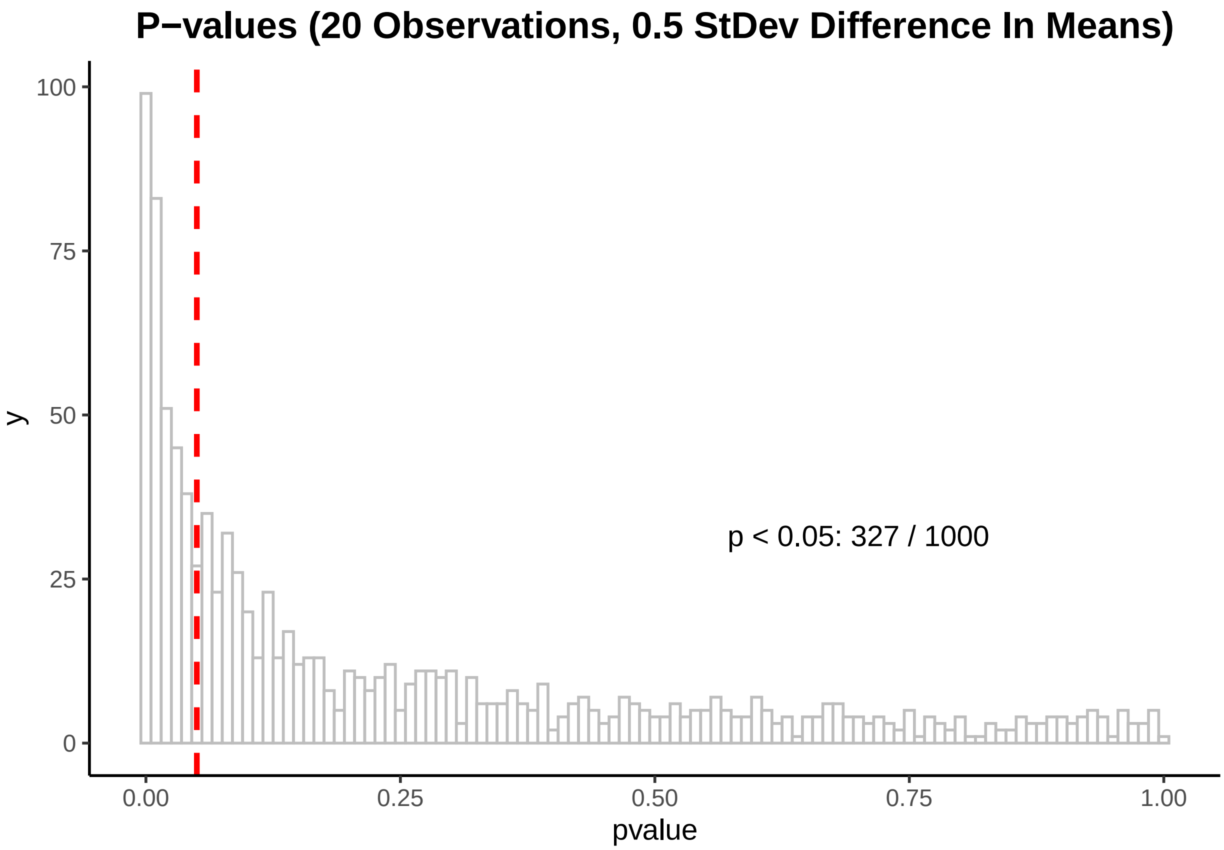
```
# Make a function that tests normal data num.t times
# The number of observations per sampling is num.0
# The means of these normal distributions are different
# The standard deviation is the same though

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))
```



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  0.042 0.075 0.182 0.560 0.908 0.992
## 20  0.047 0.121 0.359 0.881 0.997 1.000
```

```
## 30  0.057 0.167 0.485 0.970 1.000 1.000
## 50  0.036 0.253 0.688 0.998 1.000 1.000
## 100 0.050 0.399 0.939 1.000 1.000 1.000
```

```r
## 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"
```

```r
results
```

```
##         0  0.25   0.5     1   1.5     2
## 3   0.003 0.004 0.007 0.021 0.050 0.105
## 10  0.008 0.010 0.065 0.298 0.642 0.914
## 20  0.011 0.037 0.150 0.680 0.981 1.000
## 30  0.008 0.042 0.248 0.881 0.999 1.000
## 50  0.010 0.095 0.452 0.990 1.000 1.000
## 100 0.010 0.200 0.808 1.000 1.000 1.000
```