

HW1 Emma Yu MY4389

Sep 9 2014

## Problem one

Build up the two sample t statistic:

```
twosample.tstatistic = function(x1, x2){  
  # sample sizes  
  n1 = length(x1)  
  n2 = length(x2)  
  # standard deviations  
  s1 = sd(x1)  
  s2 = sd(x2)  
  # test statistic  
  sig = ((n1-1)*s1**2 + (n2-1)*s2**2)/((n1-1)+(n2-1))  
  ttstat = (mean(x1) - mean(x2))/(1/n1 + 1/n2)**0.5/sig  
  pval = (1-pt(abs(ttstat), df = n1+n2-2))*2  
  return(c(ttstat, pval))  
}
```

### Part a.

Test the code with samples drawn from two normal distributions.

Let's try it first with the same distribution for two samples. The true p-value is roughly our expected value.

```

alpha = 0.05
# Sample size
m = 15
# number of MC samples
N = 100000
n.reject = 0
for (i in 1:N){
  x1 = rnorm(m)
  x2 = rnorm(m)
  twosample.tstat = twosample.tstatistic(x1,x2)[1]
  if (abs(twosample.tstat) > qt(1-alpha/2, df = m-1)){
    n.reject = n.reject + 1
  }
}

true.sig.level = n.reject/N
true.sig.level

```

```
## [1] 0.05877
```

Let's try to change the variance of the two samples. The true p-value becomes very small as it should be. The larger the ratio of the standard deviation, the smaller the p-value is.

```

alpha = 0.05
# Sample size
m = 15
# number of MC samples
N = 100000
n.reject = 0
for (i in 1:N){
  x1 = rnorm(m, sd = 1)
  x2 = rnorm(m, sd = 2)
  twosample.tstat = twosample.tstatistic(x1,x2)[1]
  if (abs(twosample.tstat) > qt(1-alpha/2, df = m-1)){
    n.reject = n.reject + 1
  }
}

true.sig.level = n.reject/N
true.sig.level

```

```
## [1] 0.01134
```

```

alpha = 0.05
# Sample size
m = 15
# number of MC samples
N = 100000
n.reject = 0
for (i in 1:N){
  x1 = rnorm(m, sd = 1)
  x2 = rnorm(m, sd = 3)
  twosample.tstat = twosample.tstatistic(x1,x2)[1]
  if (abs(twosample.tstat) > qt(1-alpha/2, df = m-1)){
    n.reject = n.reject + 1
  }
}

true.sig.level = n.reject/N
true.sig.level

```

```
## [1] 0.00281
```

## Part b.

Test against non-normal distributions. (1) Heavy tailed distribution

Test the code with samples drawn from a student t distribution with 3 degrees of freedom. The true p-value smaller than what we expected. The algorithm assuming a normal distributin is very conservative for the t distribution with heavier tails.

```
alpha = 0.05
# Sample size
m = 15
# number of MC samples
N = 100000
n.reject = 0
for (i in 1:N){
  x1 = rt(m, df = 3)
  x2 = rt(m, df = 3)
  twosample.tstat = twosample.tstatistic(x1,x2)[1]
  if (abs(twosample.tstat) > qt(1-alpha/2, df = m-1)){
    n.reject = n.reject + 1
  }
}

true.sig.level = n.reject/N
true.sig.level
```

```
## [1] 0.01205
```

(2) Skewed distribution. Test the code with samples drawn from a Gamma distribution with  $k = 2$ . The true p-value is smaller than what we expected. The algorithm assuming a normal distribution is very conservative for the positive skewed Gamma distribution.

```
alpha = 0.05
# Sample size
m = 15
# number of MC samples
N = 100000
n.reject = 0
for (i in 1:N){
  x1 = rgamma(m, shape = 2)
  x2 = rgamma(m, shape = 2)
  twosample.tstat = twosample.tstatistic(x1, x2)[1]
  if (abs(twosample.tstat) > qt(1-alpha/2, df = m-1)){
    n.reject = n.reject + 1
  }
}

true.sig.level = n.reject/N
true.sig.level
```

```
## [1] 0.01691
```