Honda correction

Tae Kim

2/17/2019

```
set.seed(20180217)
```

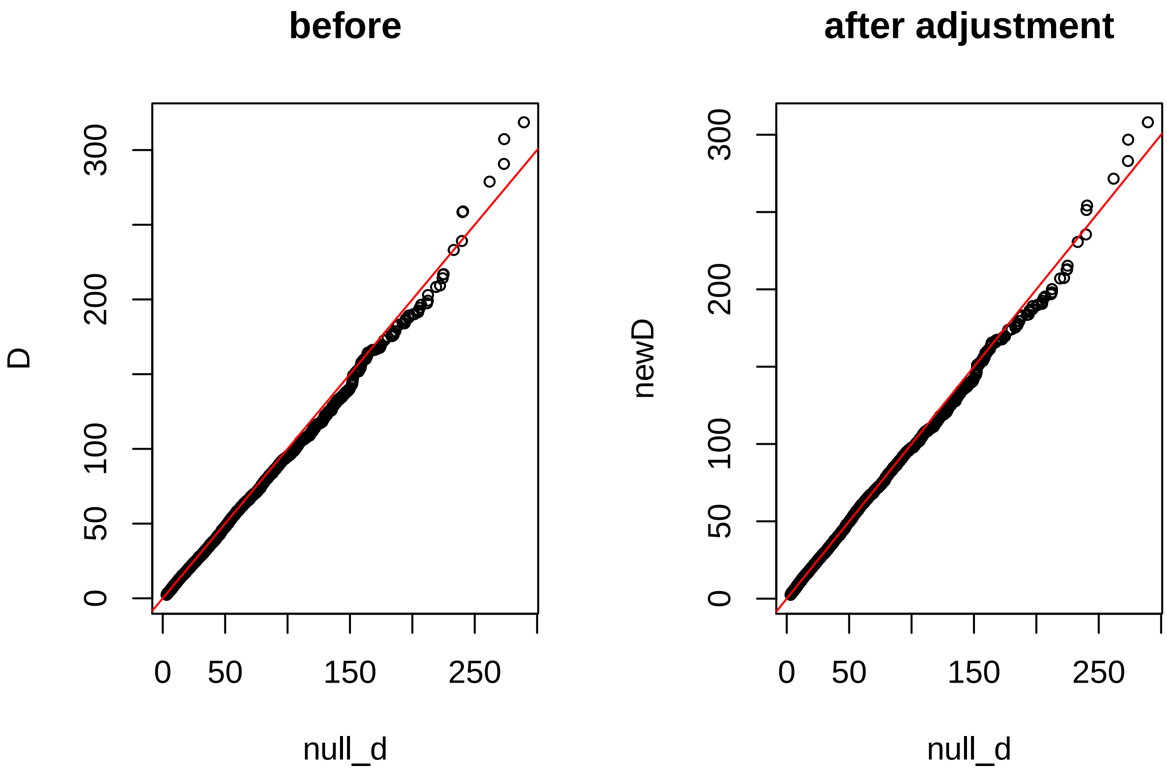
```
n = 100
K = 30
A = scale(rnorm(n)) * sqrt(n) / sqrt(n-1)
C = qchisq(0.95, 1)
p = 2
Sigma = matrix(0.5, K, K); diag(Sigma) = 1
```

Assuming H is known

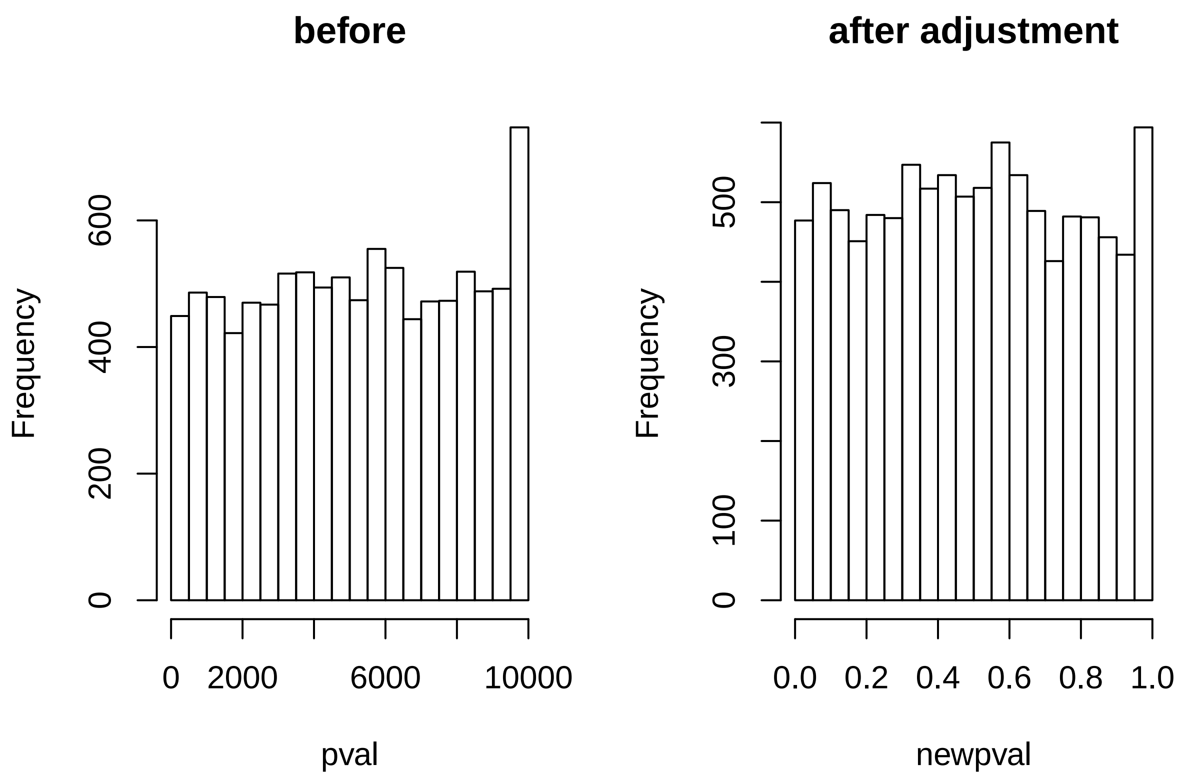
We compute the score statistic and the adjusted score statistic. We adjust the score statistics by numerically approximating the solution to the cubic function with polyroot function. We assume H is known and use the true matrix. We compare both qq-plots of the score statistics against the null and the histogram of p-values.

```
numsim = 10000
newscores = rep(NA, K-1)
newD2 = newD = D = rep(NA, numsim)
for (i in 1:numsim){
  tmp = simulate_c(A, Sigma, matrix(1))
  scores = tmp$scores
  coef = cubic_coeff_c(A,C,p,p)
  for (j in 1:(K-1)){
    roots = polyroot(c(-scores[j], coef))
    newscores[j] = Re(roots)[abs(Im(roots)) < 1e-6]
  }
  newD[i] = sum(newscores)
  D[i] = sum(scores)
}
H      = get_H_c(Sigma)
lambda = eigen(H)$values
null = matrix(NA, numsim, K-1)
for (k in 1:(K-1)){
  null[,k] = rgamma(numsim, 1/2, 1/(2*lambda[k]))
}
null_d = rowSums(null)

par(mfrow = c(1,2))
qqplot(null_d, D, cex=0.7, main="before"); abline(0,1,col='red')
qqplot(null_d, newD, cex = 0.7, main="after adjustment"); abline(0,1,col='red')
```




```
newpval2 = newpval = pval = rep(NA, numsim)
for (i in 1:numsim){
  newpval[i] = sum(null_d > newD[i])/length(D)
  newpval2[i] = sum(null_d > newD2[i])/length(D)
  pval[i] = sum(null_d > D[i])
}
hist(pval, 20, main="before")
hist(newpval, 20, main="after adjustment")
```




```
print(sum(pval < 0.05)/length(pval))
```

```
## [1] 3e-04
```

```
print(sum(newpval<0.05) / length(newpval))
```

```
## [1] 0.0477
```

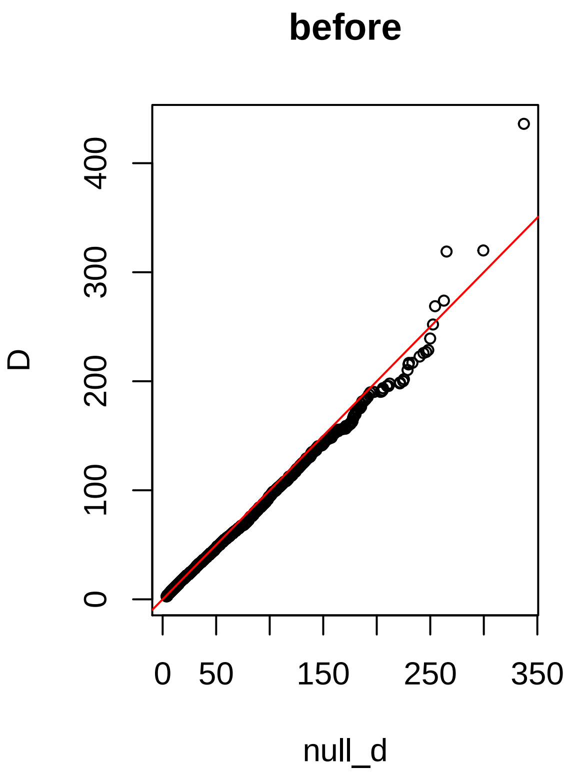
Assuming H is unknown

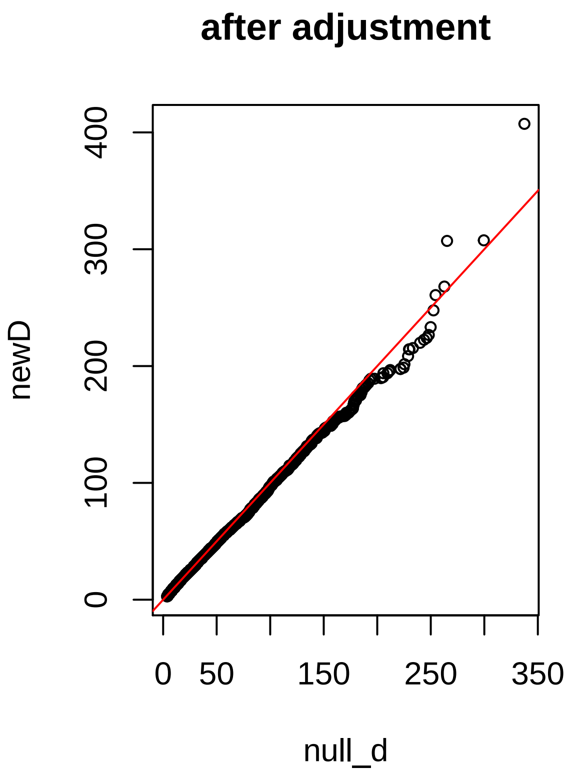
We repeat the same procedure except we assume H is unknown. We estimate the empirical covariance matrix of the data Y, and compute H from the estimated covariance matrix. Again, we compare both qq-plots of the score statistics against the null and the histogram of p-values. Due to computational burden, we repeat the procedure 5,000 times.

```
numsim = 10000
newscores = rep(NA, K-1)
newD = D = rep(NA, numsim)
newpval = pval = rep(NA, numsim)
for (i in 1:numsim){
  tmp = simulate_c(A, Sigma, matrix(1))
  scores = tmp$scores
  Hhat = tmp$Hhat
  coef = cubic_coeff_c(A,C,p,p)
  for (j in 1:(K-1)){
    roots = polyroot(c(-scores[j], coef))
    newscores[j] = Re(roots)[abs(Im(roots)) < 1e-6]
  }
  newD[i] = sum(newscores)
  D[i] = sum(scores)
```

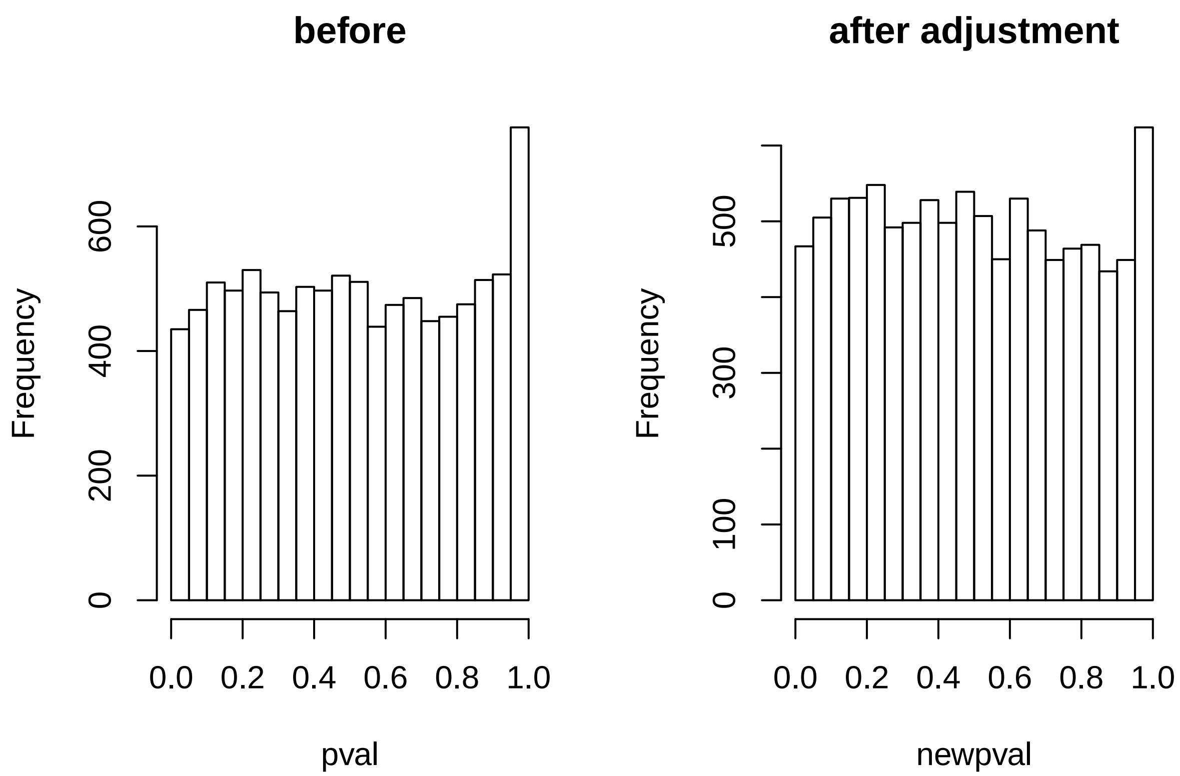
```
  lambda = eigen(H)$values
  null = matrix(NA, numsim, K-1)
  for (k in 1:(K-1)){
    null[,k] = rgamma(numsim, 1/2, 1/(2*lambda[k]))
  }
  null_d = rowSums(null)
  pval[i] = sum(null_d > D[i])/length(D)
  newpval[i] = sum(null_d > newD[i])/length(newD)
}

par(mfrow = c(1,2))
qqplot(null_d, D, cex=0.7, main="before"); abline(0,1,col='red')
qqplot(null_d, newD, cex = 0.7, main="after adjustment"); abline(0,1,col='red')
```






```
hist(pval, 20, main="before")
hist(newpval, 20, main="after adjustment")
```




```r
print(sum(pval < 0.05)/length(pval))
```

```
## [1] 0.0435
```

```r
print(sum(newpval<0.05) / length(newpval))
```

```
## [1] 0.0465
```