ph1855_hw3_ygu5

Yue Gu

2024-02-03

Hollander et al. Nonparametric Statistical Methods-Chapter 4

P133 Q1

```
qnorm(0.95) #1.645
```

```
## [1] 1.644854
```

P135 Q15

```
1-pnorm(2.27)
```

```
## [1] 0.01160379
```

P136 Q16

```
# read data
noninjured = c(.19, .14, .02, .44, .37)
iSCI = c(.89, .76, .63, .69, .58, .79, .02, .79)

wilcox.test(c(noninjured, iSCI)~factor(c(0,0,0,0,0, 1,1,1,1,1,1,1,1)))
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  c(noninjured, iSCI) by factor(c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1))
## W = 4.5, p-value = 0.02768
## alternative hypothesis: true location shift is not equal to 0
```

P136 Q17

```r
noninjured = c(.19, .14, .02, .44, .37, 0, 0, 0)
iSCI =       c(.89, .76, .63, .69, .58, .79, .02, .79)

#install.packages("agricolae")
library(agricolae)
results = waerden.test(iSCI, noninjured)
results
```

```
## $statistics
##      Chisq Df   p.chisq
##   3.593444  5 0.6092979
##
## $parameters
##      test     name.t ntr alpha
##   Waerden noninjured   6  0.05
##
## $means
##           iSCI normalScore       std r  Min  Max   Q25  Q50  Q75
## 0    0.5333333 -0.01454693 0.4445597 3 0.02 0.79 0.405 0.79 0.79
## 0.02 0.6300000 -0.43164424        NA 1 0.63 0.63 0.630 0.63 0.63
## 0.14 0.7600000  0.14083537        NA 1 0.76 0.76 0.760 0.76 0.76
## 0.19 0.8900000  1.22122722        NA 1 0.89 0.89 0.890 0.89 0.89
## 0.37 0.5800000 -0.76545610        NA 1 0.58 0.58 0.580 0.58 0.58
## 0.44 0.6900000 -0.14083537        NA 1 0.69 0.69 0.690 0.69 0.69
##
## $comparison
## NULL
##
## $groups
##            score groups
## 0.19  1.22122722      a
## 0.14  0.14083537      a
## 0    -0.01454693      a
## 0.44 -0.14083537      a
## 0.02 -0.43164424      a
## 0.37 -0.76545610      a
##
## attr(,"class")
## [1] "group"
```

P141 Q18

```r
# input nonallergics (baseline) vs allergics
X =  c(48.1, 48.0, 45.5, 41.7, 35.4, 34.3, 32.4, 29.1, 27.3, 18.9, 6.6, 5.2, 4.7)
Y =  c(1651.0, 1112.0, 102.4, 100.0, 67.6, 65.9, 64.7, 39.6, 31.0)
# store the difference
n = 9
m = 13
```

```
diff = NULL

for (i in 1:m){
  for (j in 1:n){
    diff = c(diff, X[i]-Y[j])
}
  }
# sort the difference by increasing order
sort(diff)[59]
```

```
## [1] -54.3
```

```
# check output from the test results
wilcox.test(X, Y, conf.int = T)
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  X and Y
## W = 11, p-value = 0.000772
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -95.8 -22.1
## sample estimates:
## difference in location
##                  -54.3
```

P144 Q30

```
X =  c(48.1, 48.0, 45.5, 41.7, 35.4, 34.3, 32.4, 29.1, 27.3, 18.9, 6.6, 5.2, 4.7)
Y =  c(1651.0, 1112.0, 102.4, 100.0, 67.6, 65.9, 64.7, 39.6, 31.0)
wilcox.test(X, Y, conf.int = T, conf.level = 0.95)
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  X and Y
## W = 11, p-value = 0.000772
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -95.8 -22.1
## sample estimates:
## difference in location
##                  -54.3
```

P145 Q35

```r
X =  c(48.1, 48.0, 45.5, 41.7, 35.4, 34.3, 32.4, 29.1, 27.3, 18.9, 6.6, 5.2, 4.7)
Y =  c(1651.0, 1112.0, 102.4, 100.0, 67.6, 65.9, 64.7, 39.6, 31.0)
# store the difference
n = 9
m = 13
diff = NULL

for (i in 1:m){
  for (j in 1:n){
    diff = c(diff, X[i]-Y[j])
}
  }

# find the number of position 29 and 89
sort(diff)[29]
```

```
## [1] -95.8
```

```r
sort(diff)[89]
```

```
## [1] -22.1
```

```r
# compare with Q20
wilcox.test(X, Y, conf.int = T, conf.level = 0.95)
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  X and Y
## W = 11, p-value = 0.000772
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -95.8 -22.1
## sample estimates:
## difference in location
##                  -54.3
```

P149 Q43

```r
X =  c(48.1, 48.0, 45.5, 41.7, 35.4, 34.3, 32.4, 29.1, 27.3, 18.9, 6.6, 5.2, 4.7)
Y =  c(1651.0, 1112.0, 102.4, 100.0, 67.6, 65.9, 64.7, 39.6, 31.0)

library(NSM3)
pFligPoli(X, Y)
```

```
## Number of X values: 13 Number of Y values: 9
## Fligner-Policello U Statistic: 5.6287
## Monte Carlo (Using 10000 Iterations) upper-tail probability: 4e-04
## Monte Carlo (Using 10000 Iterations) two-sided p-value: 8e-04
##
```