MBTI_test

Zhilin Kong

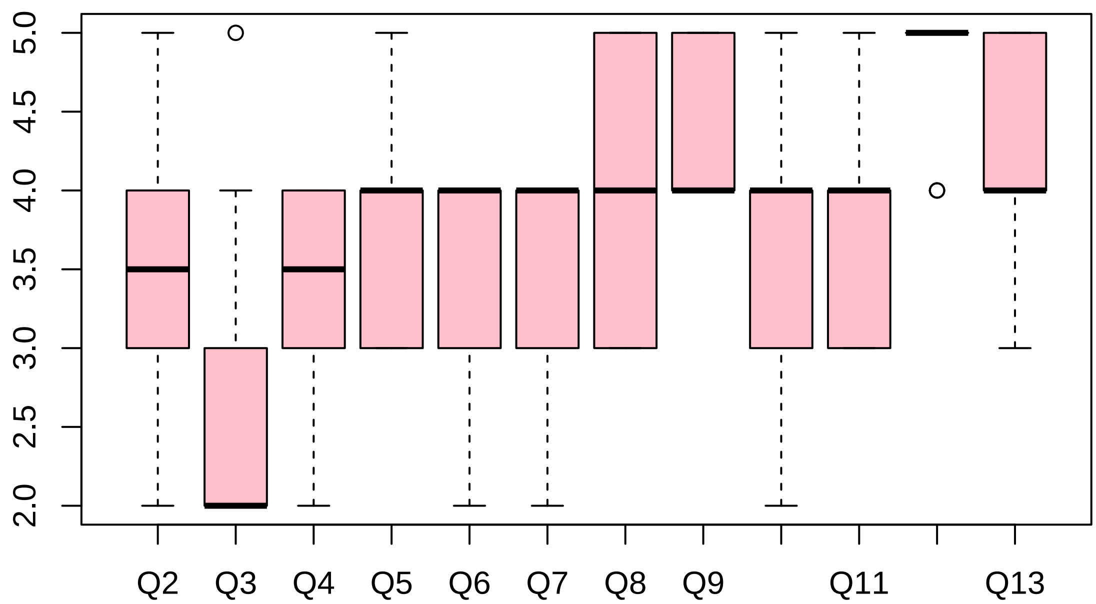
2022-06-12

Data

```
library(ggplot2)
Gender = c(rep("Male",5),rep("Female",5))
Q2 = c(5,4,4,3,3,4,4,3,3,2)
Q3 = c(5,4,2,3,2,3,2,2,2,2)
Q4 = c(4,4,3,3,3,3,4,4,2,4)
Q5 = c(4,3,3,4,3,4,4,4,3,5)
Q6 = c(3,4,4,4,2,4,4,4,4,2)
Q7 = c(4,4,4,4,2,3,4,3,4,4)
Q8 = c(5,5,3,4,3,5,4,3,3,4)
Q9 = c(4,4,5,4,4,4,4,4,5,5)
Q10 = c(4,4,2,4,3,4,4,4,5,3)
Q11 = c(4,4,4,4,3,3,4,3,5,3)
Q12 = c(5,5,5,4,5,5,5,4,5,5)
Q13 = c(4,5,4,4,5,4,4,3,5,4)
df = data.frame(Gender,factor(Q2),
                Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11,Q12,Q13);df
```

```
##    Gender factor.Q2. Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13
## 1    Male          5  5  4  4  3  4  5  4   4   4   5   4
## 2    Male          4  4  4  3  4  4  5  4   4   4   5   5
## 3    Male          4  2  3  3  4  4  3  5   2   4   5   4
## 4    Male          3  3  3  4  4  4  4  4   4   4   4   4
## 5    Male          3  2  3  3  2  2  3  4   3   3   5   5
## 6  Female          4  3  3  4  4  3  5  4   4   3   5   4
## 7  Female          4  2  4  4  4  4  4  4   4   4   5   4
## 8  Female          3  2  4  4  4  3  3  4   4   3   4   3
## 9  Female          3  2  2  3  4  4  3  5   5   5   5   5
## 10 Female          2  2  4  5  2  4  4  5   3   3   5   4
```

```
matrix = cbind(Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11,Q12,Q13)
boxplot(matrix, col = "pink")
```



Boostrapping (vs. MBTI resutls)

H0: The Survey's result agrees with the MBTI's result.

H1: The Survey's result doesn't agree with the MBTI's result.

```
library(boot)
```

```
## Warning:   'boot' R 4.1.3
```

```
#Observed Test Statistic
scale = c(1,2,3,4,5)
Mind = 3 - abs(3-quantile(scale,probs=0.67)[[1]]);Mind #vs. Q3
```

```
## [1] 2.32
```

```
Energy = quantile(scale,probs=0.59)[[1]];Energy #vs.Q5
```

```
## [1] 3.36
```

```r
Nature = quantile(scale,probs=0.75)[[1]];Nature #vs. Q4
```

```
## [1] 4
```

```r
Tactics = quantile(scale,probs=0.74)[[1]];Tactics #vs. Q6
```

```
## [1] 3.96
```

```r
Identity = quantile(scale,probs=0.67)[[1]]; Identity #vs.Q7
```

```
## [1] 3.68
```

```r
#Random Sampling Function.
# data: the data where the sample is from.
# i: index
#Return the mean of each sample.
function1 = function(data,i) {
  d = data[i]
  return (mean(d))
}

#Bootstrapping function to find confidence interval(CI) and give conclusion.
#vec: the bootstrapping vector.
#obs: the observed statistic.
#print: Boolean variable. True:print the bootstrap statistics and CI;   False:otherwise.
#question: a string. the question's number.
#return a string, a conclusion based on the test result.
boot_function = function(vec, obs, print,question){
  results = boot(data=vec,statistic=function1,R=1000)
  ci = boot.ci(results,type="basic")

  if(print){
    print(results)
    print(ci)
  }

  lower = ci$basic[4]
  upper = ci$basic[5]
  if (obs >= lower & obs<= upper){
    str = paste(question,"failed to reject the H0.")
  }
  else{
    str = paste(question, "rejects the H0.")
  }
  return (str)
}
#For readers' reference
boot_function(Q3,Mind,TRUE,'Q3')
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
```

```
##
##
## Call:
## boot(data = vec, statistic = function1, R = 1000)
##
##
## Bootstrap Statistics :
##     original  bias    std. error
## t1*      2.7  -0.007   0.3024276
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = results, type = "basic")
##
## Intervals :
## Level      Basic
## 95%   ( 2.1,  3.2 )
## Calculations and Intervals on Original Scale
```

```
## [1] "Q3 failed to reject the H0."
```

```r
boot_function(Q5,Energy,F,'Q5')
```

```
## [1] "Q5 failed to reject the H0."
```

```r
boot_function(Q4,Nature,F,'Q4')
```

```
## [1] "Q4 rejects the H0."
```

```r
boot_function(Q6,Tactics,F,'Q6')
```

```
## [1] "Q6 failed to reject the H0."
```

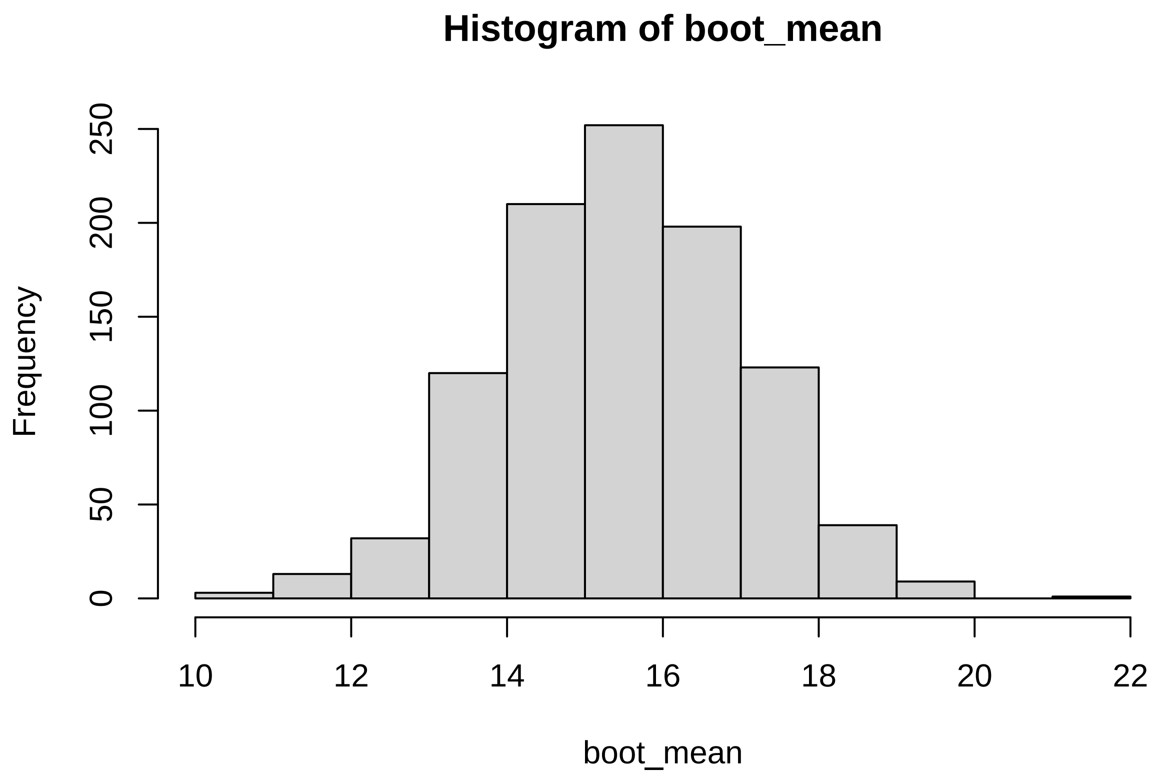
```r
boot_function(Q7,Identity,F,'Q7')
```

```
## [1] "Q7 failed to reject the H0."
```

Conclusion: Q3 Introverted vs. Extroverted -> Both agree: Introverted
Q5 Idealism vs. Pragmatism -> Both agree: Pragmatism
Q4 Emotional vs. Rational -> Disagree
Q6 Disorganized vs. Organized -> Both agree: Organized
Q7 Self-abased vs. Confident -> Both agree: Confident

Another way to use bootstrapping(for reader who is interested)

```r
set.seed(123)
x = 1:30
boot_mean = c()
for (i in 1:1000){
  resam = base::sample(x,size=length(x),replace=TRUE)
  mu = mean(resam)
  boot_mean = append(boot_mean,mu)
}
hist(boot_mean)
```




```r
quantile(boot_mean, probs=c(0.025,0.975))
```

```
##     2.5%    97.5%
## 12.46583 18.43417
```

Test whether there is a different criteria for two genders.

H0: They have the same criteria.

H1: They have different criterion.

The grade's mean for each question

```
mu_each_question = c(colMeans(matrix));mu_each_question
```

```
##  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9 Q10 Q11 Q12 Q13
## 3.5 2.7 3.4 3.7 3.5 3.6 3.9 4.3 3.7 3.7 4.8 4.2
```

```
#sort in ascending order.
sort(mu_each_question)
```

```
##  Q3  Q4  Q2  Q6  Q7  Q5 Q10 Q11  Q8 Q13  Q9 Q12
## 2.7 3.4 3.5 3.5 3.6 3.7 3.7 3.7 3.9 4.2 4.3 4.8
```

calculate the means of two genders.

```
#Get rid of column "Q2"
mat = matrix[,2:12] #or using mat=matrix[,colnames(matrix)!="Q2"]
Female_mean = colMeans(mat[6:10,]);Female_mean
```

```
##  Q3  Q4  Q5  Q6  Q7  Q8  Q9 Q10 Q11 Q12 Q13
## 2.2 3.4 4.0 3.6 3.6 3.8 4.4 4.0 3.6 4.8 4.0
```

```
Male_mean = colMeans(mat[1:5,]);Male_mean
```

```
##  Q3  Q4  Q5  Q6  Q7  Q8  Q9 Q10 Q11 Q12 Q13
## 3.2 3.4 3.4 3.4 3.6 4.0 4.2 3.4 3.8 4.8 4.4
```

A more general way to calculate the means is by using group_by method.

```
library(dplyr)
```

```
##
##    'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
Qlist = c(Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,Q11,Q12,Q13)
df1 = df %>% group_by(Gender)%>%summarise(mean_Q3 = mean(Q3),
                                          mean_Q4 = mean(Q4),
                                          mean_Q5 = mean(Q5),
                                          mean_Q6 = mean(Q6),
                                          mean_Q7 = mean(Q7),
                                          mean_Q8 = mean(Q8),
                                          mean_Q9 = mean(Q9),
                                          mean_Q10 = mean(Q10),
                                          mean_Q11 = mean(Q11),
                                          mean_Q12 = mean(Q12),
                                          mean_Q13 = mean(Q13),.groups = 'drop')
df1
```

```
## # A tibble: 2 x 12
##   Gender mean_Q3 mean_Q4 mean_Q5 mean_Q6 mean_Q7 mean_Q8 mean_Q9 mean_Q10
##   <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>    <dbl>
## 1 Female     2.2     3.4     4       3.6     3.6     3.8     4.4      4
## 2 Male       3.2     3.4     3.4     3.4     3.6     4       4.2      3.4
## # ... with 3 more variables: mean_Q11 <dbl>, mean_Q12 <dbl>, mean_Q13 <dbl>
```

Applying t.test

```r
library(ggstatsplot)
```

```
## You can cite this package as:
##      Patil, I. (2021). Visualizations with statistical details: The 'ggstatsplot' approach.
##      Journal of Open Source Software, 6(61), 3167, doi:10.21105/joss.03167
```

```r
matrix.t = cbind(Male_mean,Female_mean);matrix.t
```

```
##     Male_mean Female_mean
## Q3        3.2         2.2
## Q4        3.4         3.4
## Q5        3.4         4.0
## Q6        3.4         3.6
## Q7        3.6         3.6
## Q8        4.0         3.8
## Q9        4.2         4.4
## Q10       3.4         4.0
## Q11       3.8         3.6
## Q12       4.8         4.8
## Q13       4.4         4.0
```

```r
t.test(Male_mean,Female_mean,paired=TRUE)
```

```
##
##  Paired t-test
##
```

```
## data:  Male_mean and Female_mean
## t = 0.13363, df = 10, p-value = 0.8963
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2849794  0.3213430
## sample estimates:
## mean of the differences
##              0.01818182
```

p.value $= 0.8963 > \alpha = 0.05$. Therefore, I failed to reject the H_0.

Or applying the non-parametric test:Wilcoxon signed-rank test

```
wilcox.test(Male_mean,Female_mean,paired=TRUE,exact=F,correct=F)
```

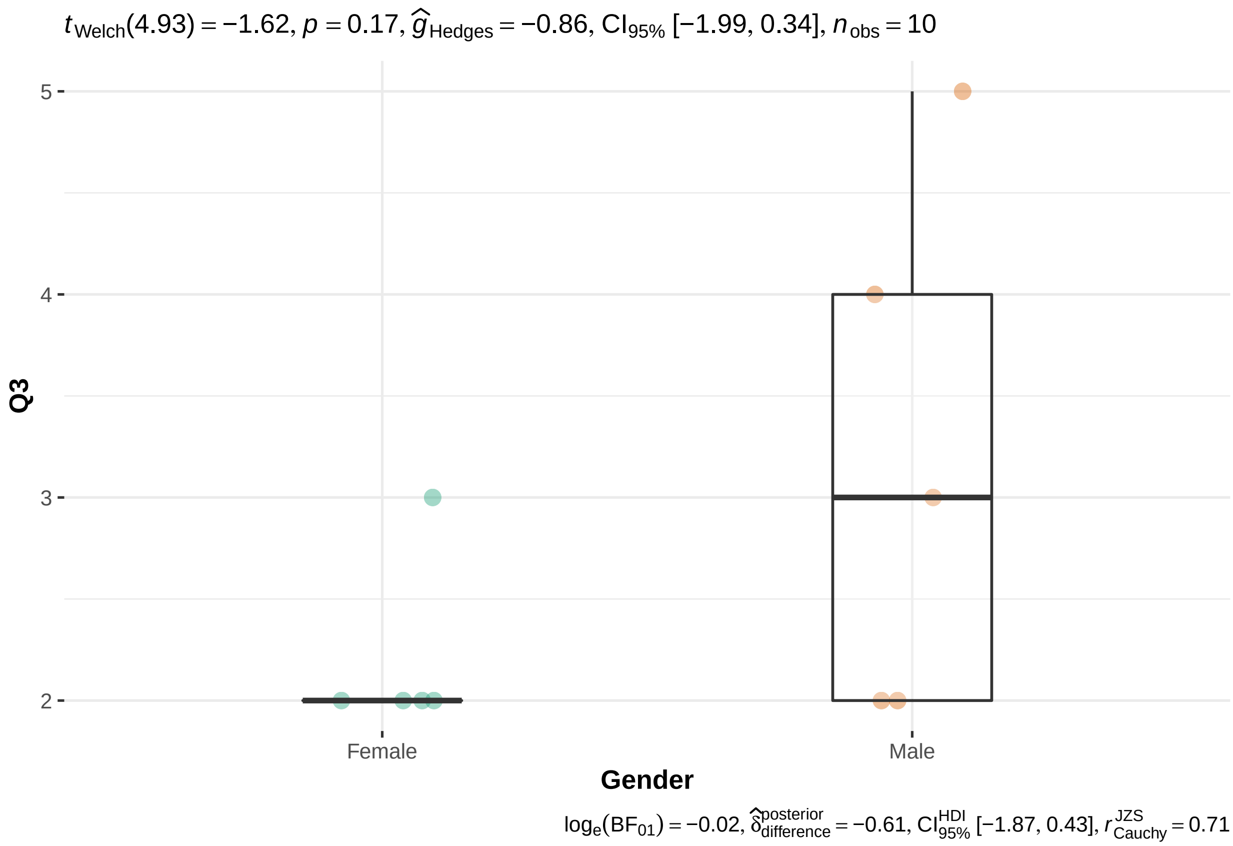
```
##
##  Wilcoxon signed rank test
##
## data:  Male_mean and Female_mean
## V = 17, p-value = 0.888
## alternative hypothesis: true location shift is not equal to 0
```

p.value $= 0.888 > \alpha = 0.05$. Therefore, again, I failed to reject the H_0.

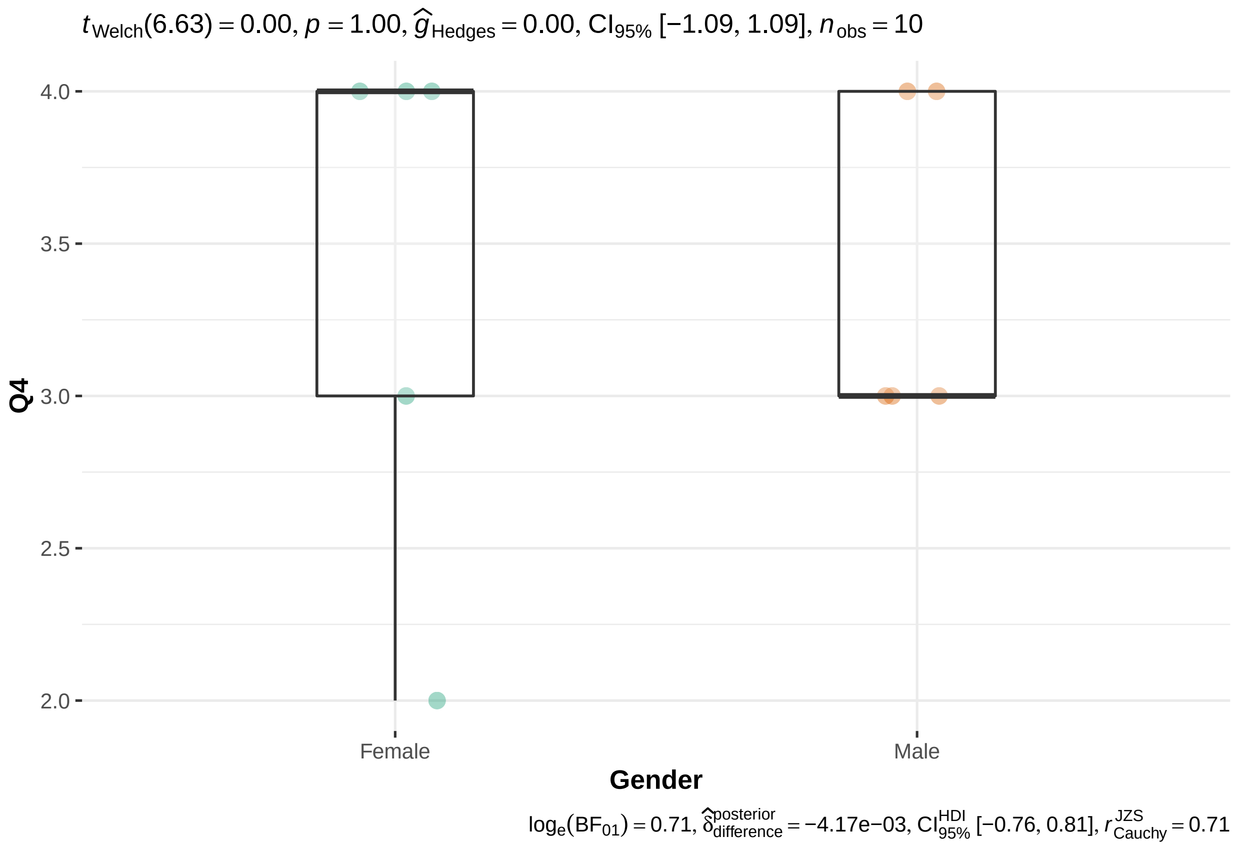
Conclusion: Based on this sample, I failed to reject the H0: males and females have the same judgement.

parametric version (for your reference)

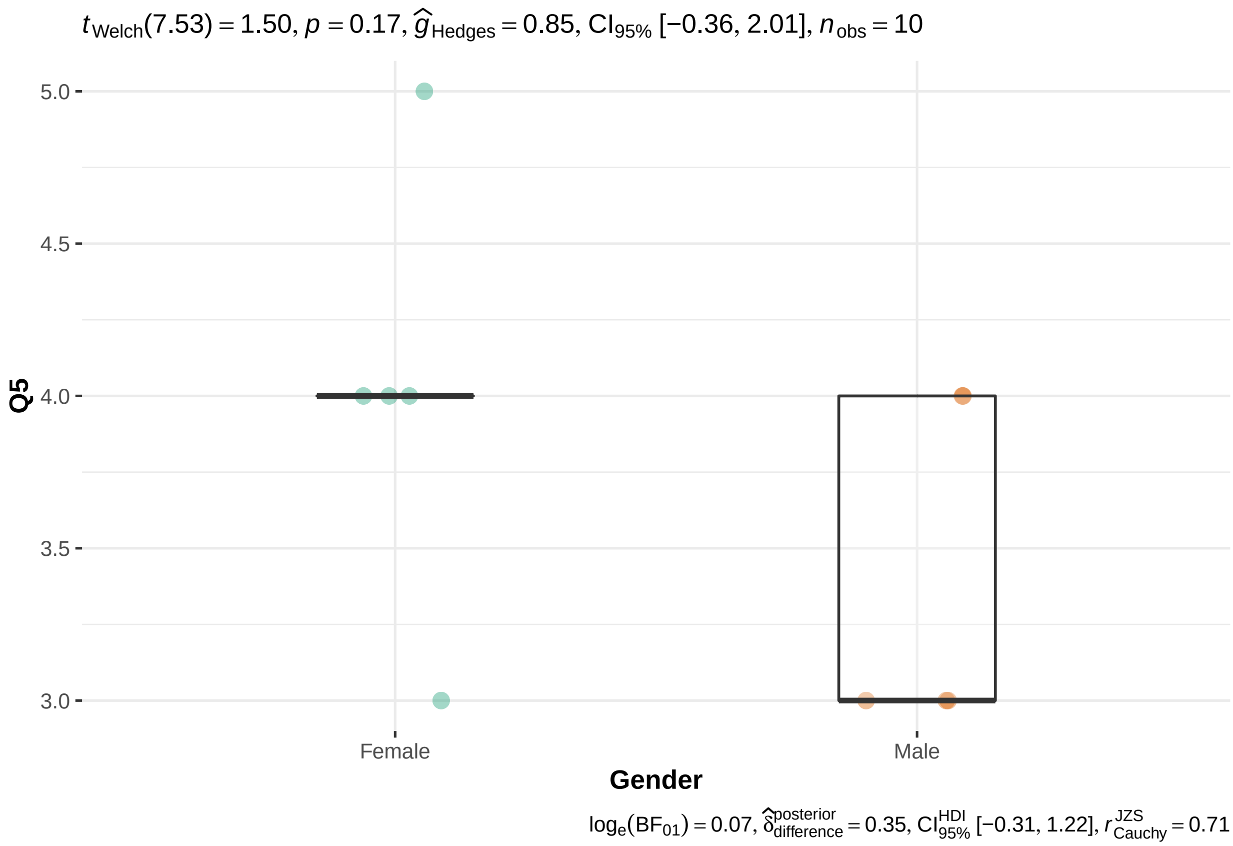
```
  ggbetweenstats(data = df,x = Gender,y = Q3,
                 plot.type = "box",type = "parametric",centrality.plotting = FALSE)
```



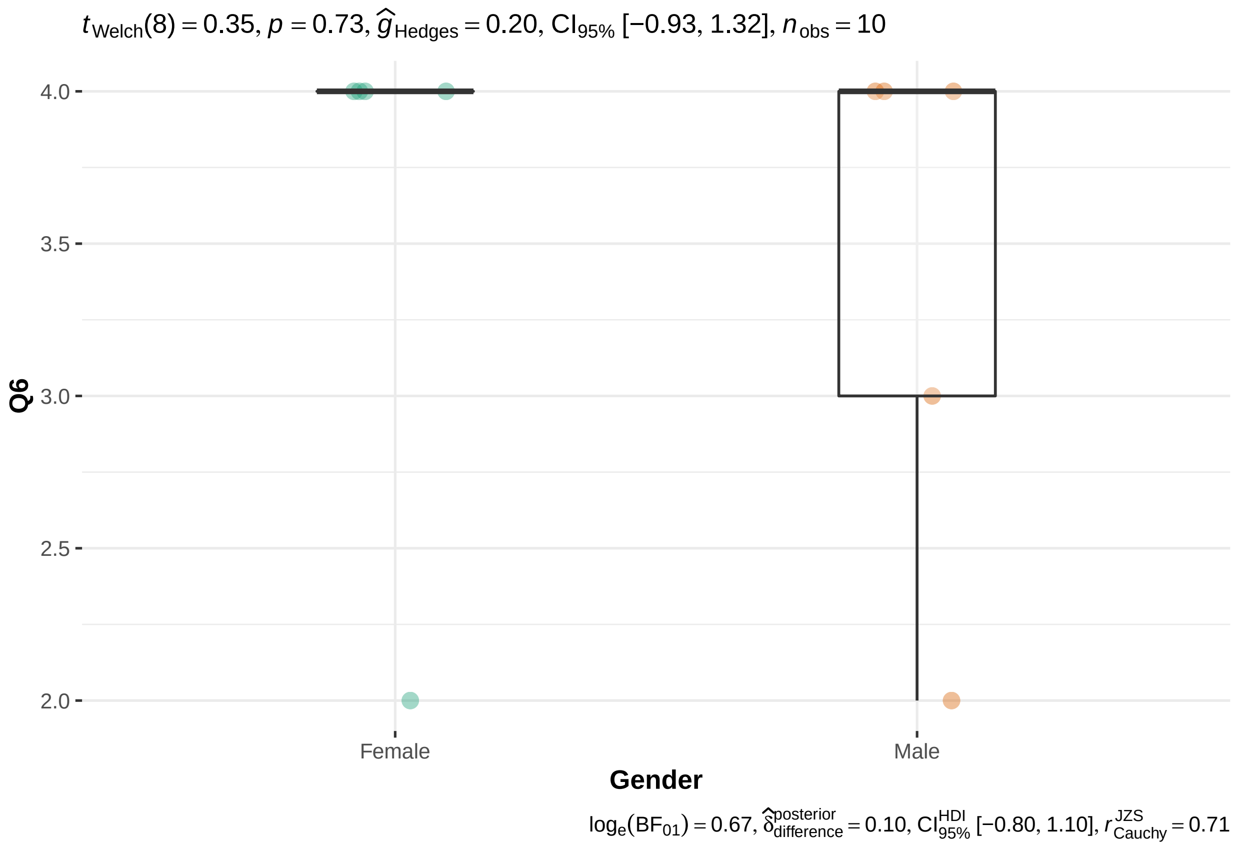

```
ggbetweenstats(data = df,x = Gender,y = Q4,
               plot.type = "box",type = "parametric",centrality.plotting = FALSE)
```



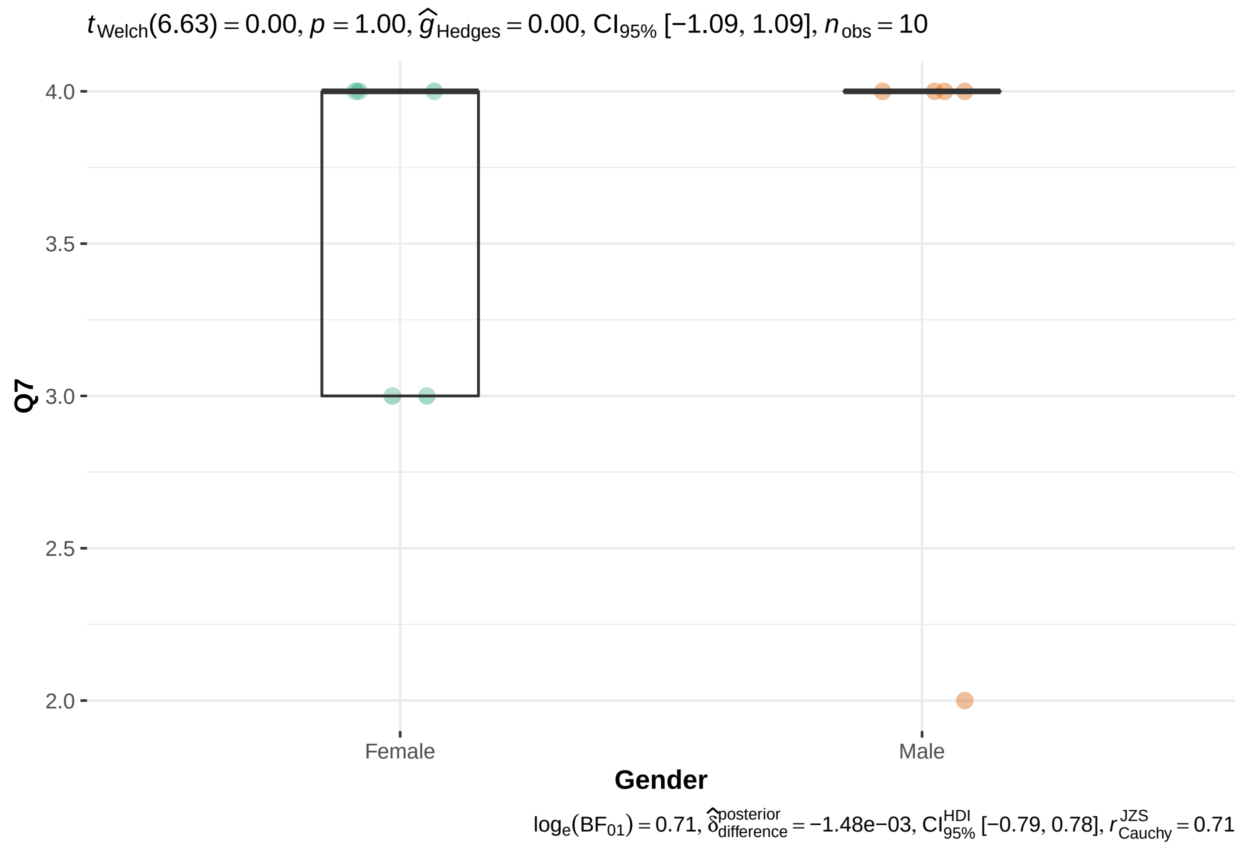

```
ggbetweenstats(data = df,x = Gender,y = Q5,
               plot.type = "box",type = "parametric",centrality.plotting = FALSE)
```



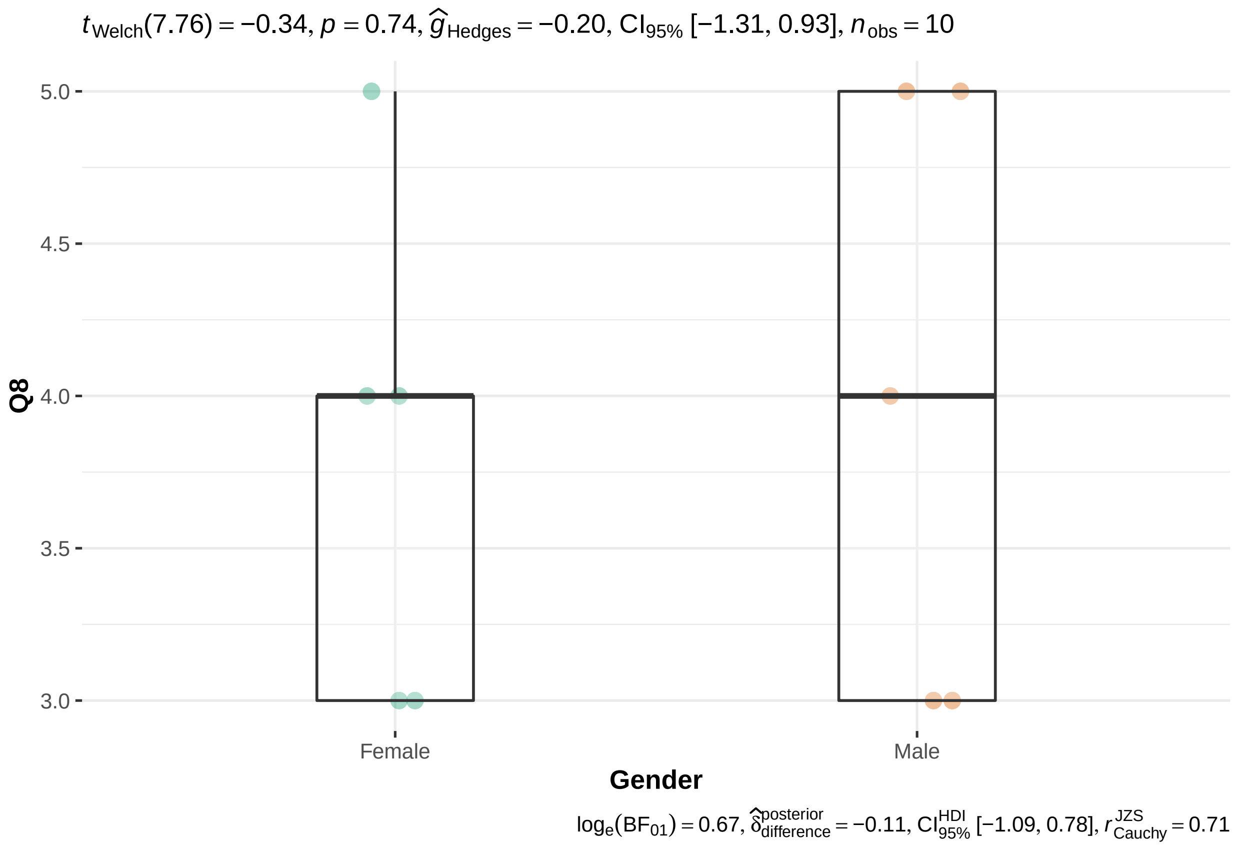

```
ggbetweenstats(data = df,x = Gender,y = Q6,
               plot.type = "box",type = "parametric",centrality.plotting = FALSE)
```



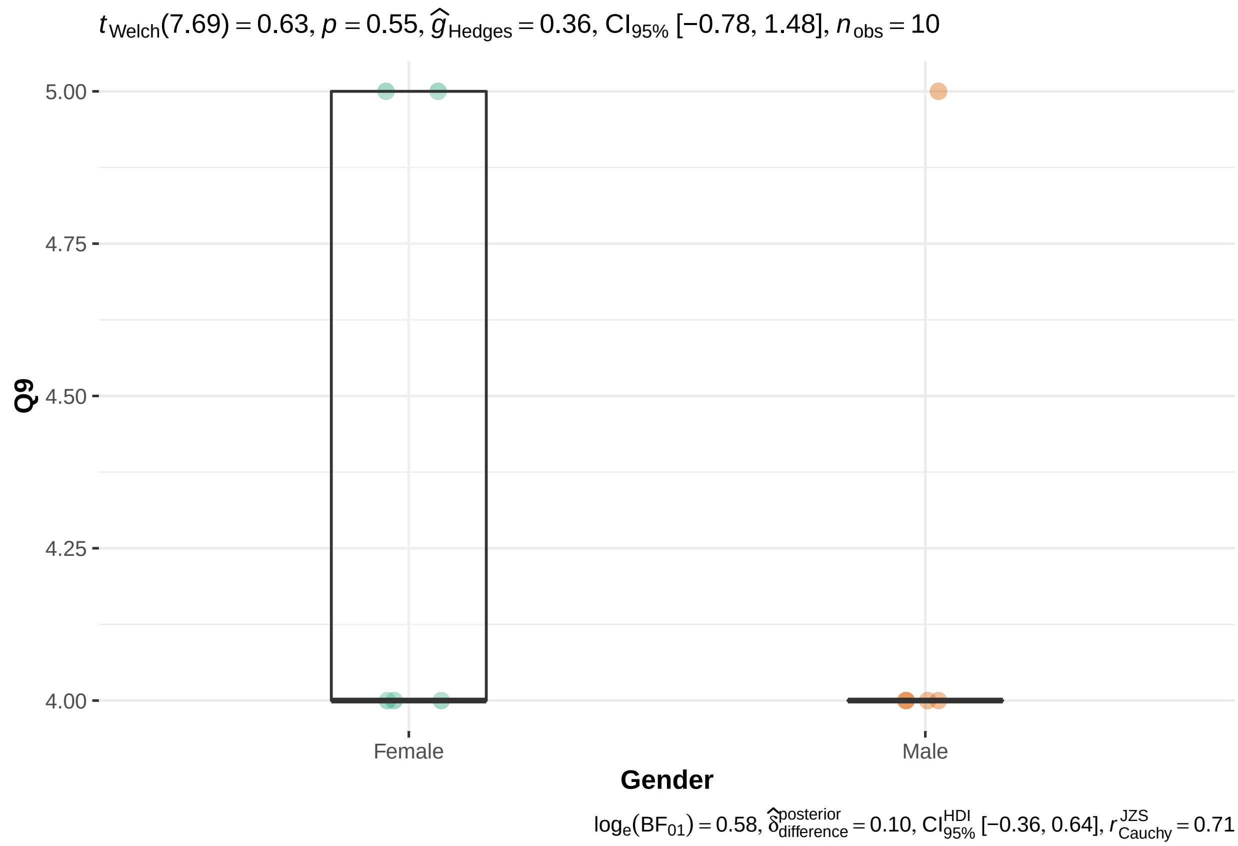

```
ggbetweenstats(data = df,x = Gender,y = Q7,
               plot.type = "box",type = "parametric",centrality.plotting = FALSE)
```



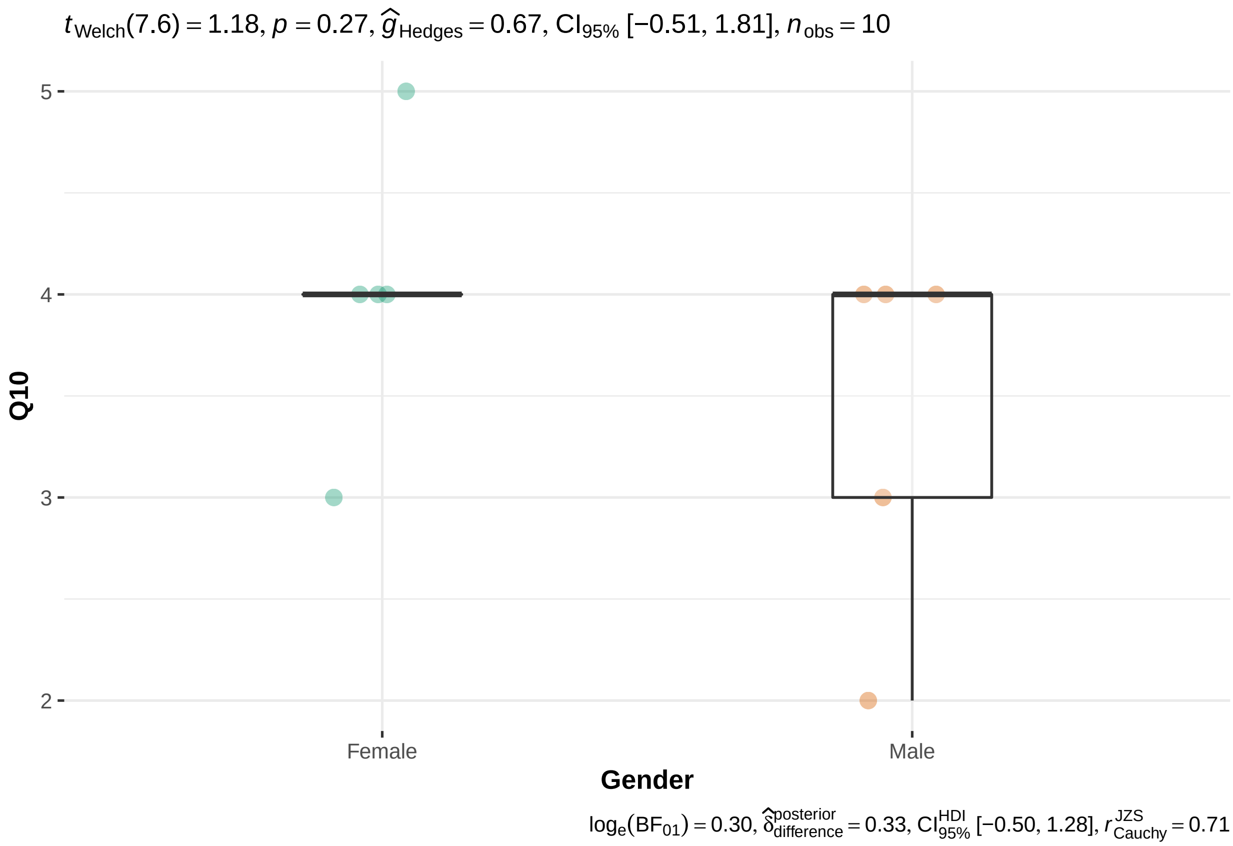

```
ggbetweenstats(data = df,x = Gender,y = Q8,
               plot.type = "box",type = "parametric",centrality.plotting = FALSE)
```



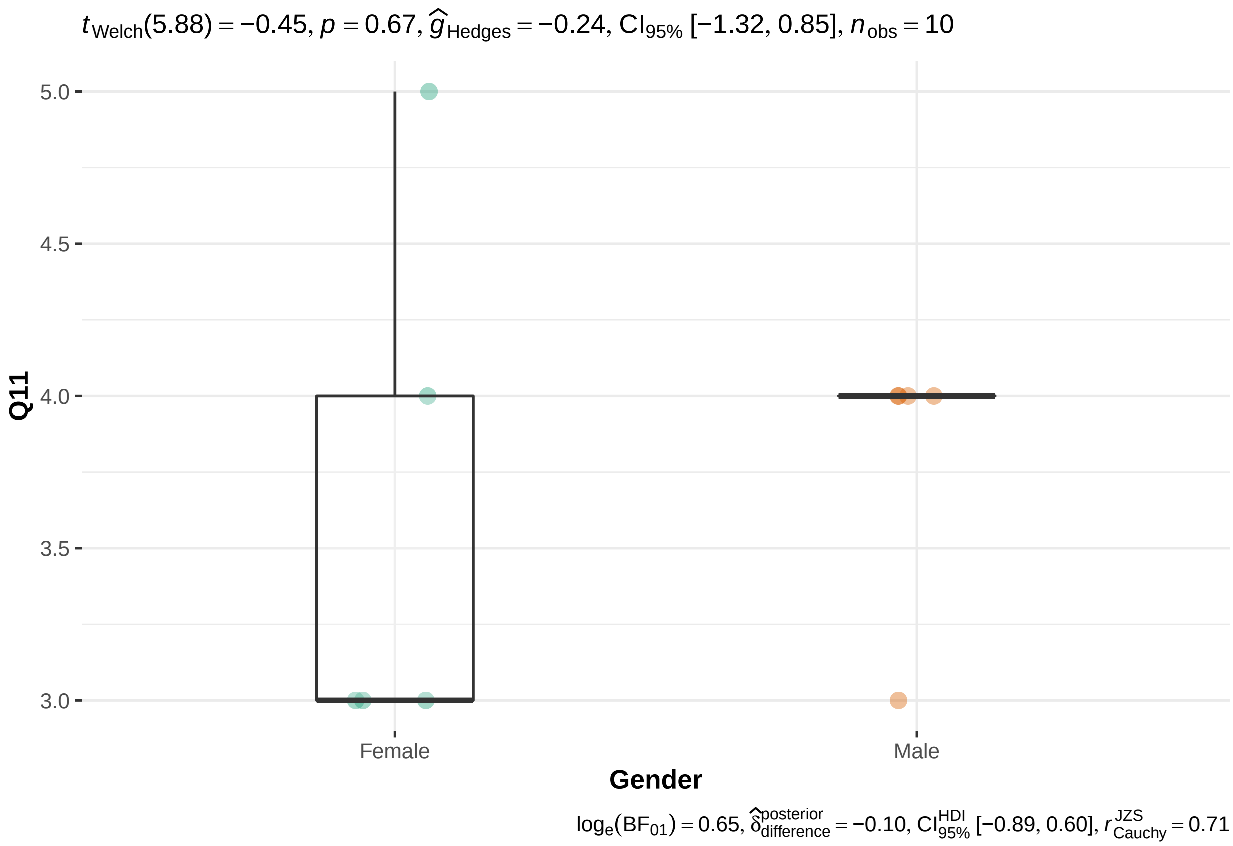

```
ggbetweenstats(data = df,x = Gender,y = Q9,
               plot.type = "box",type = "parametric",centrality.plotting = FALSE)
```



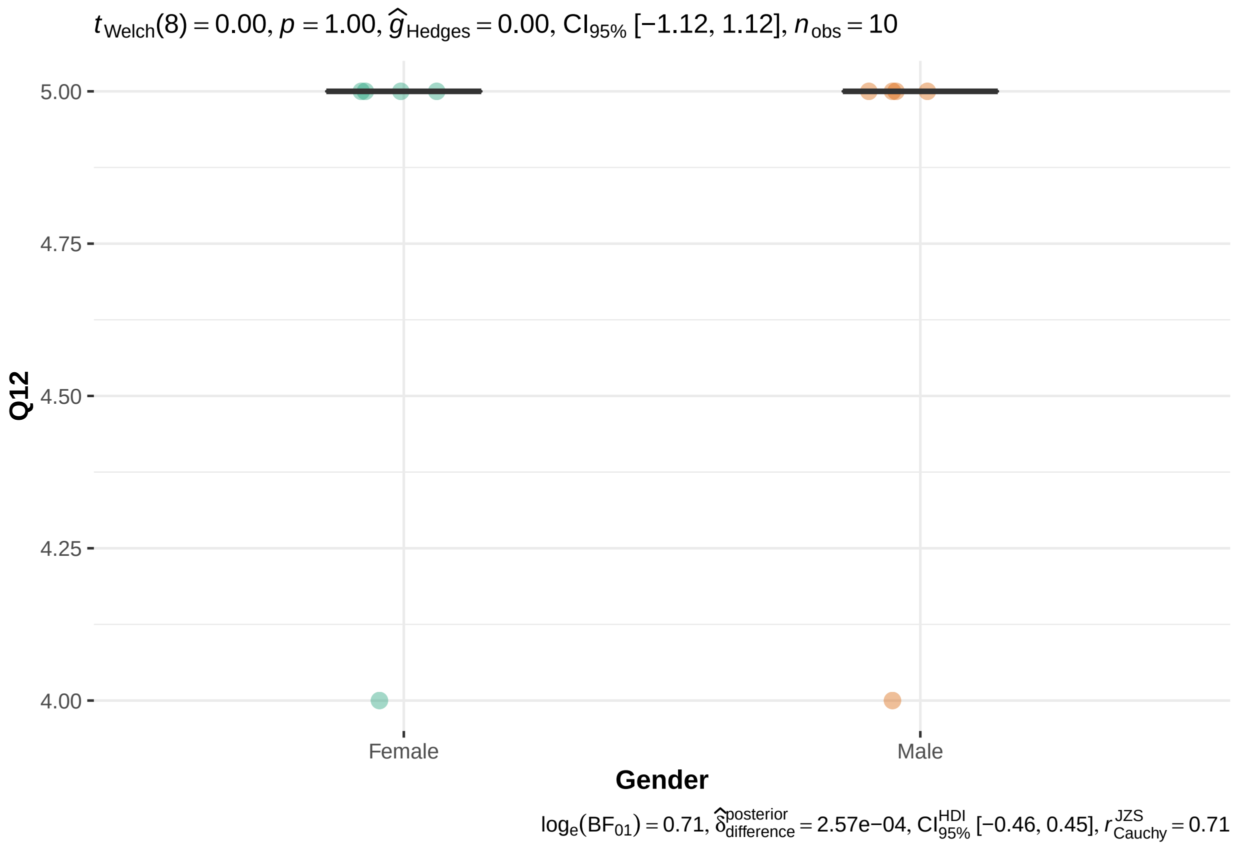

```
ggbetweenstats(data = df,x = Gender,y = Q10,
               plot.type = "box",type = "parametric",centrality.plotting = FALSE)
```



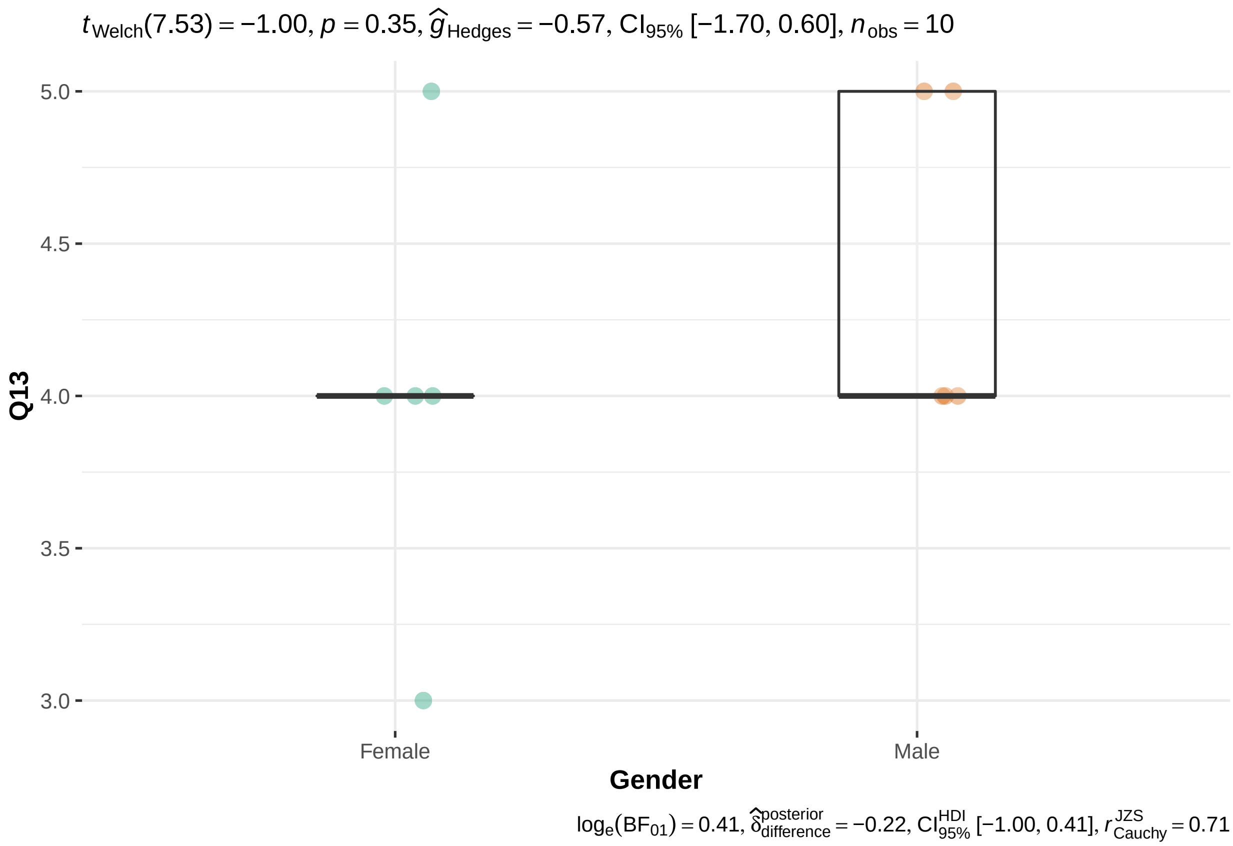

```
ggbetweenstats(data = df,x = Gender,y = Q11,
               plot.type = "box",type = "parametric",centrality.plotting = FALSE)
```




```
ggbetweenstats(data = df,x = Gender,y = Q12,
               plot.type = "box",type = "parametric",centrality.plotting = FALSE)
```

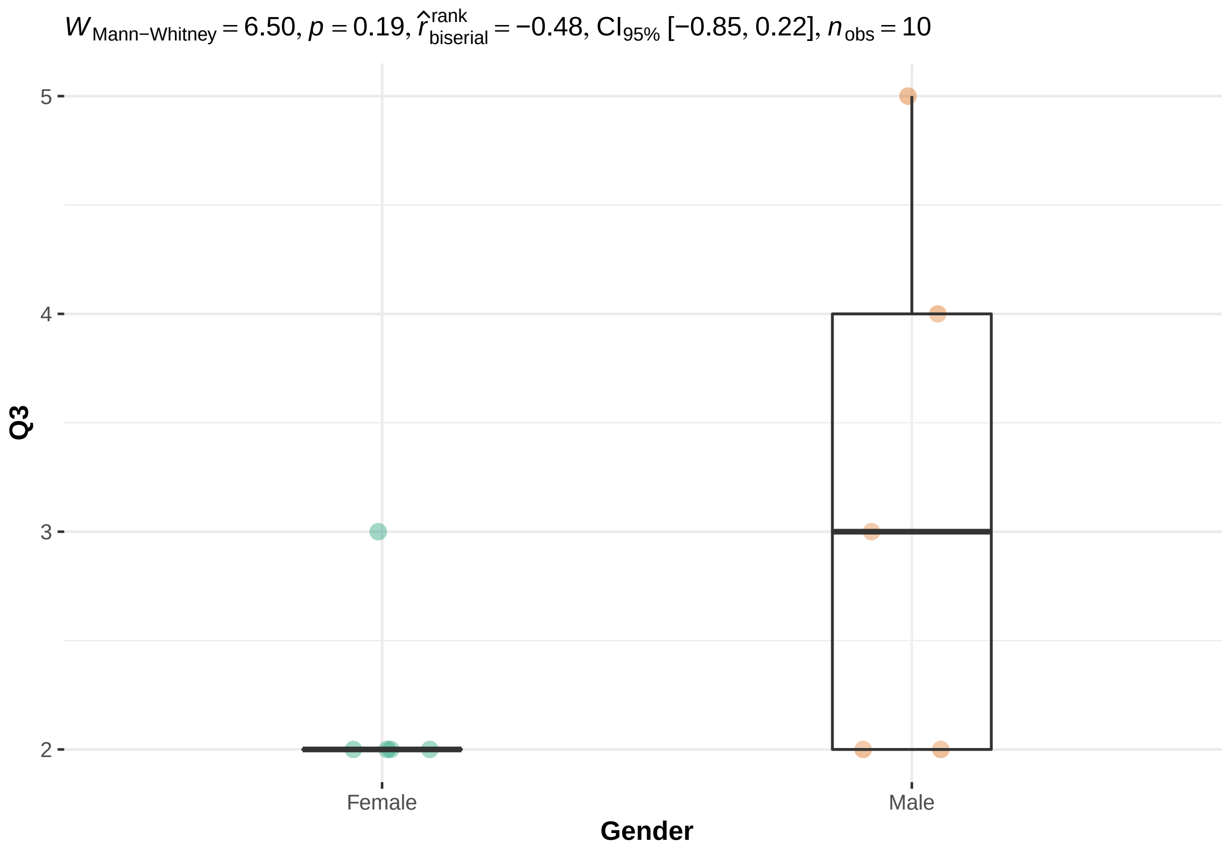



```
ggbetweenstats(data = df,x = Gender,y = Q13,
               plot.type = "box",type = "parametric",centrality.plotting = FALSE)
```

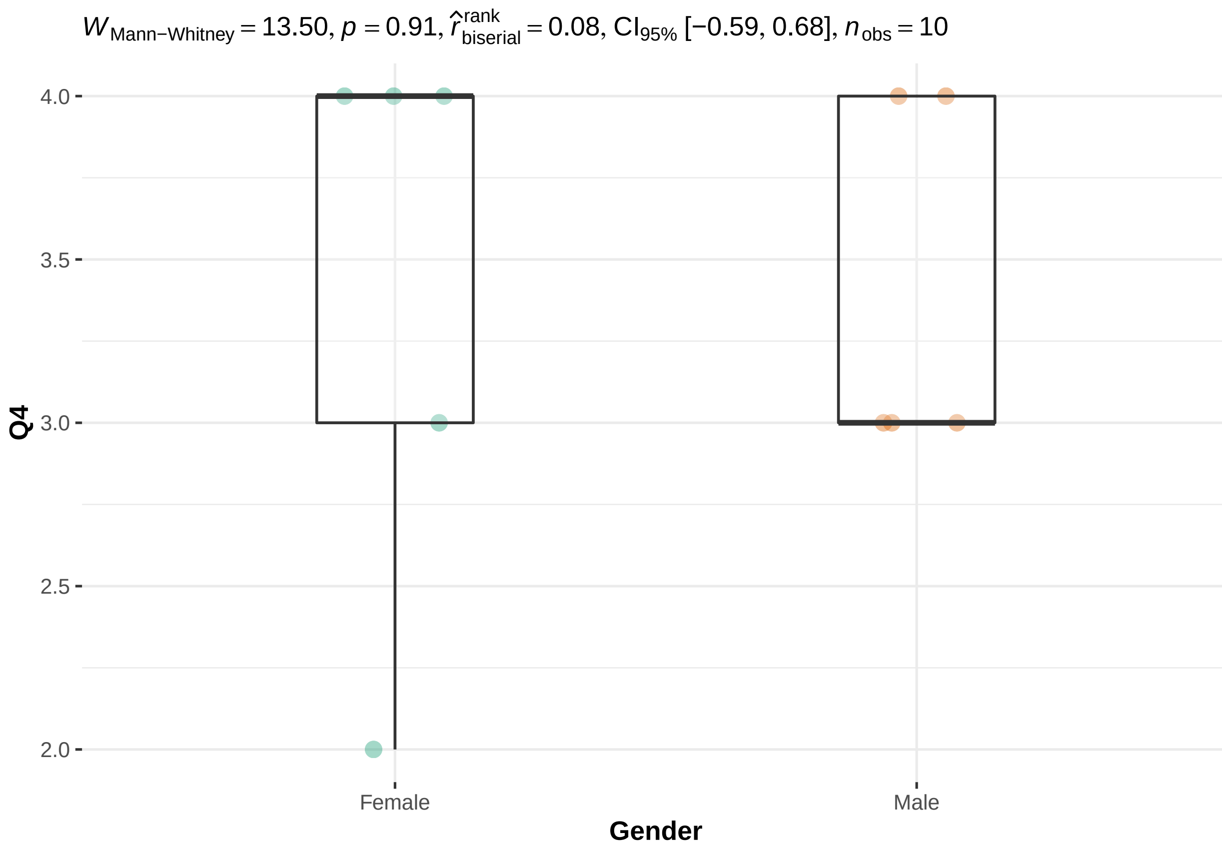


The non-parametric version

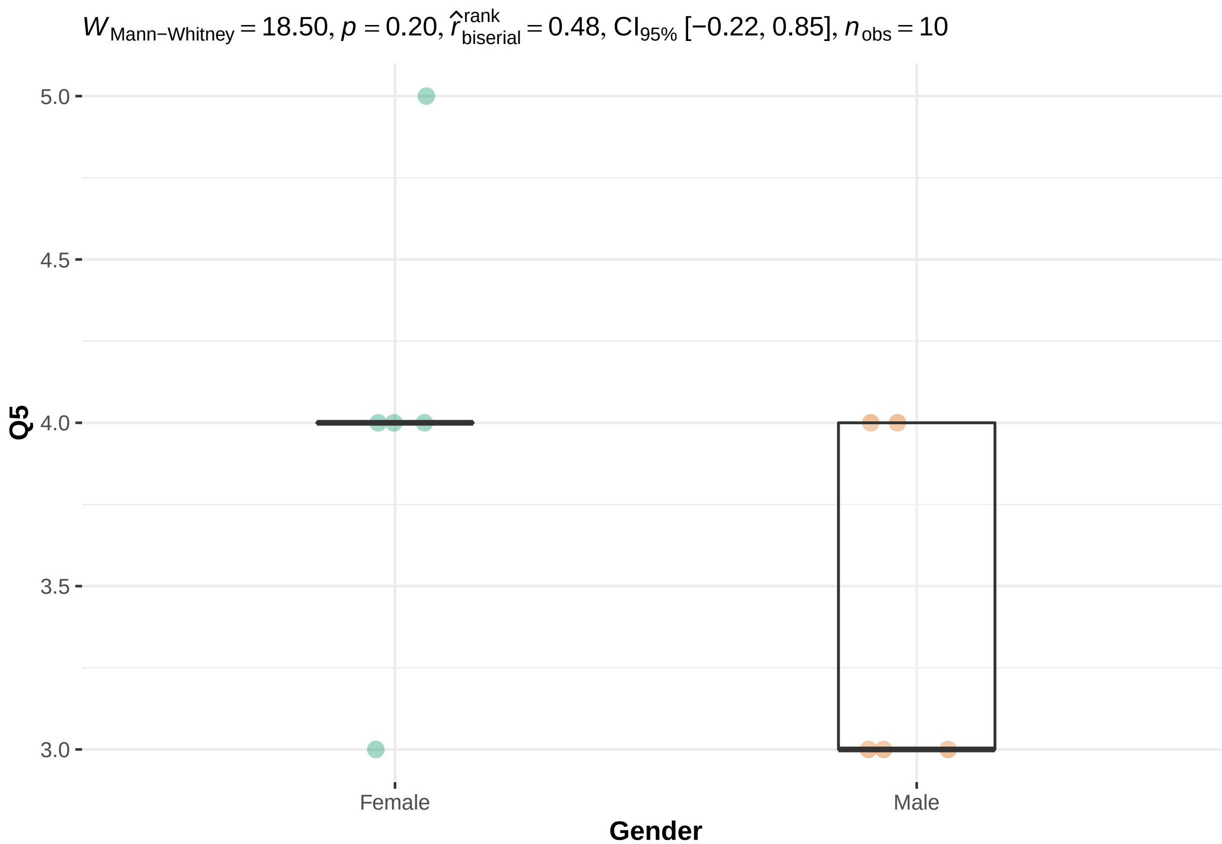
```
ggbetweenstats(data = df,x = Gender,y = Q3,
               plot.type = "box",type = "nonparametric",centrality.plotting = FALSE)
```



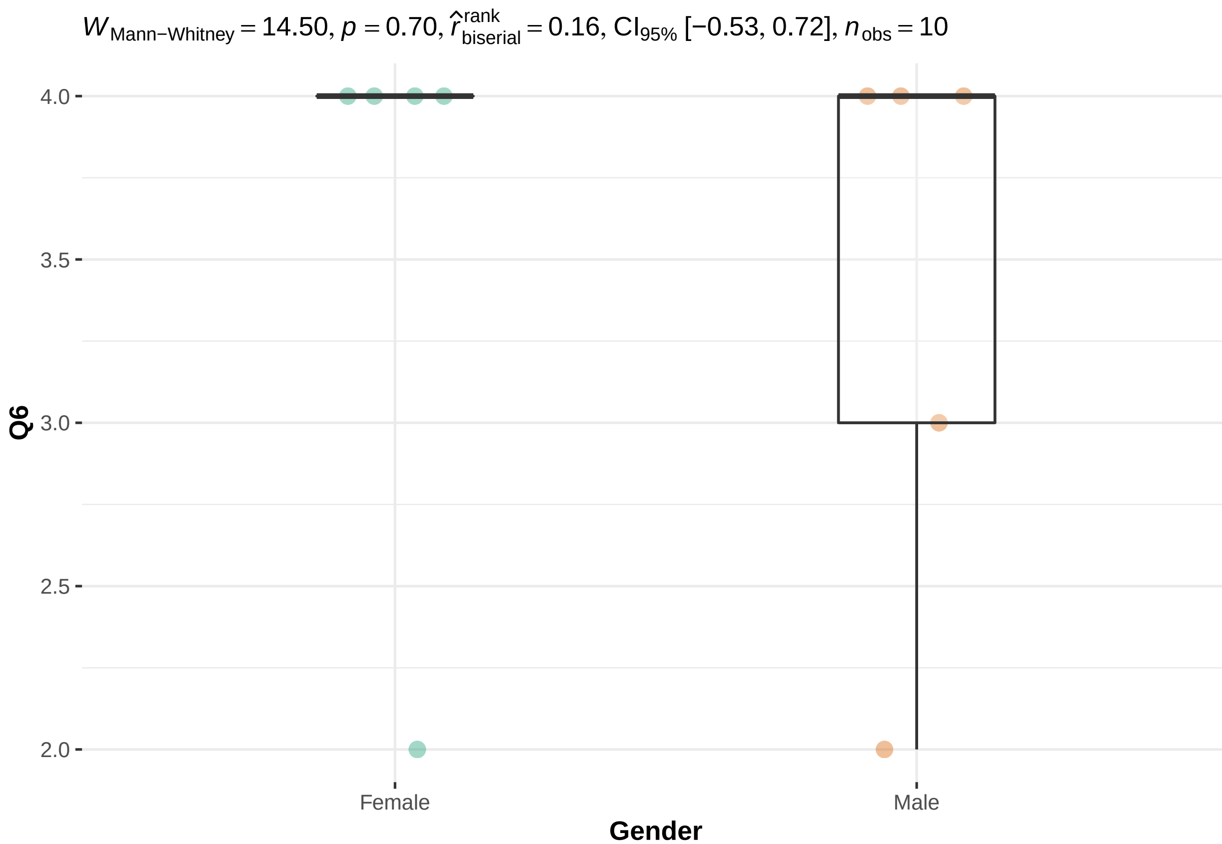

```
ggbetweenstats(data = df,x = Gender,y = Q4,
               plot.type = "box",type = "nonparametric",centrality.plotting = FALSE)
```



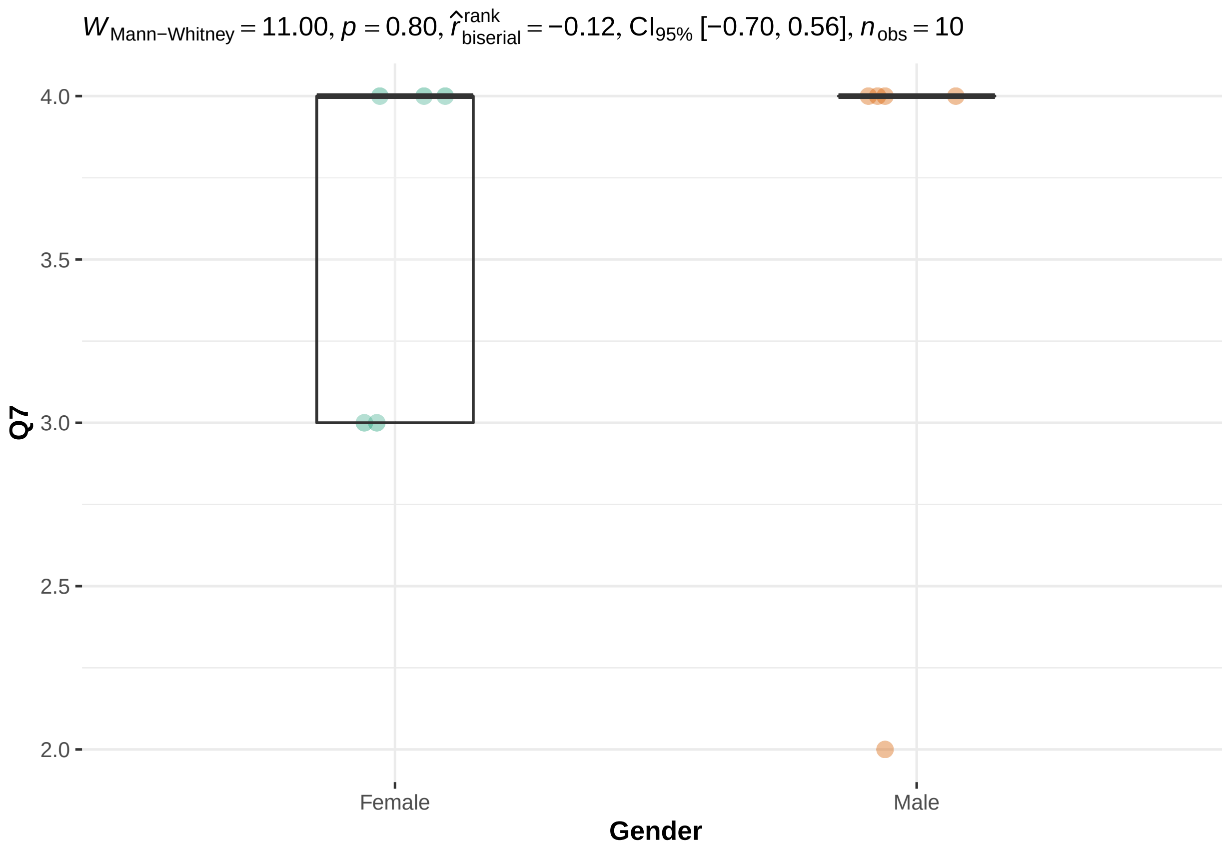

```
ggbetweenstats(data = df,x = Gender,y = Q5,
               plot.type = "box",type = "nonparametric",centrality.plotting = FALSE)
```



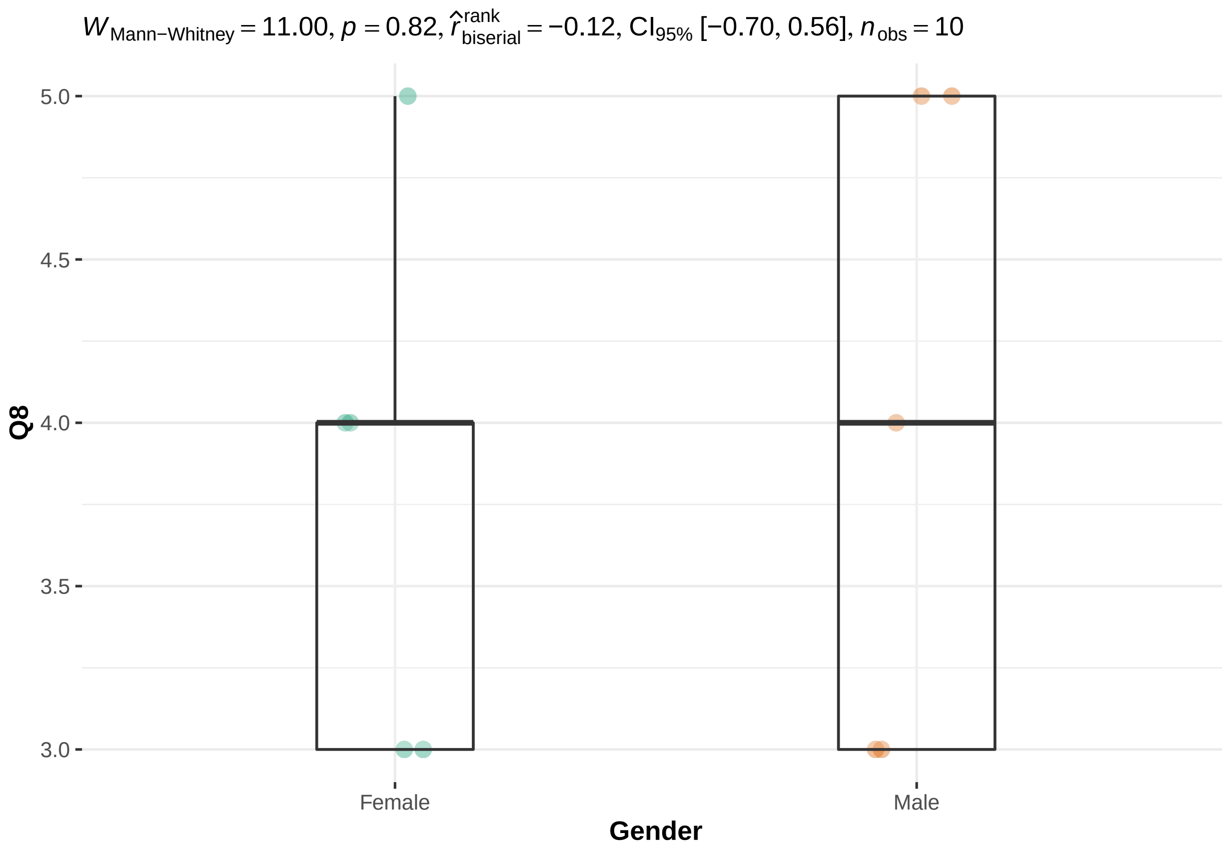

```
ggbetweenstats(data = df,x = Gender,y = Q6,
               plot.type = "box",type = "nonparametric",centrality.plotting = FALSE)
```



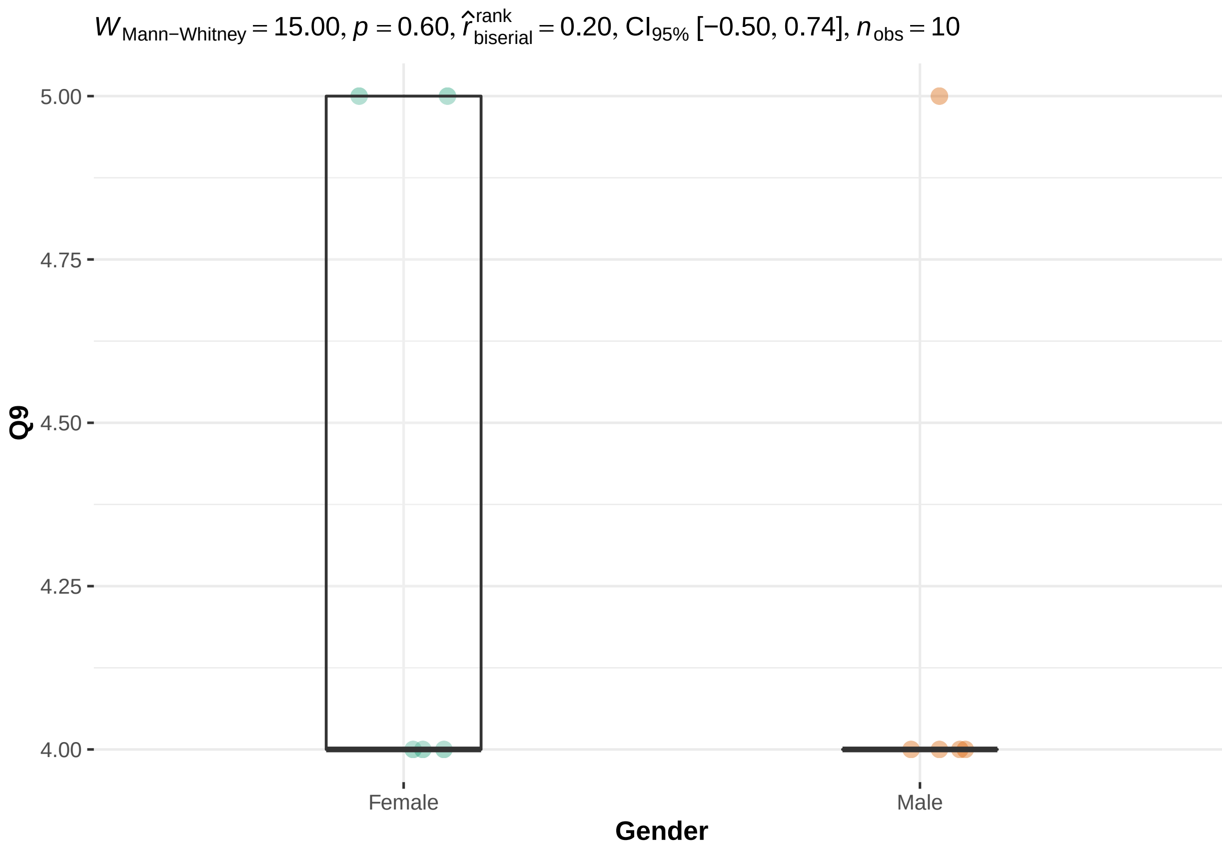

```
ggbetweenstats(data = df,x = Gender,y = Q7,
               plot.type = "box",type = "nonparametric",centrality.plotting = FALSE)
```



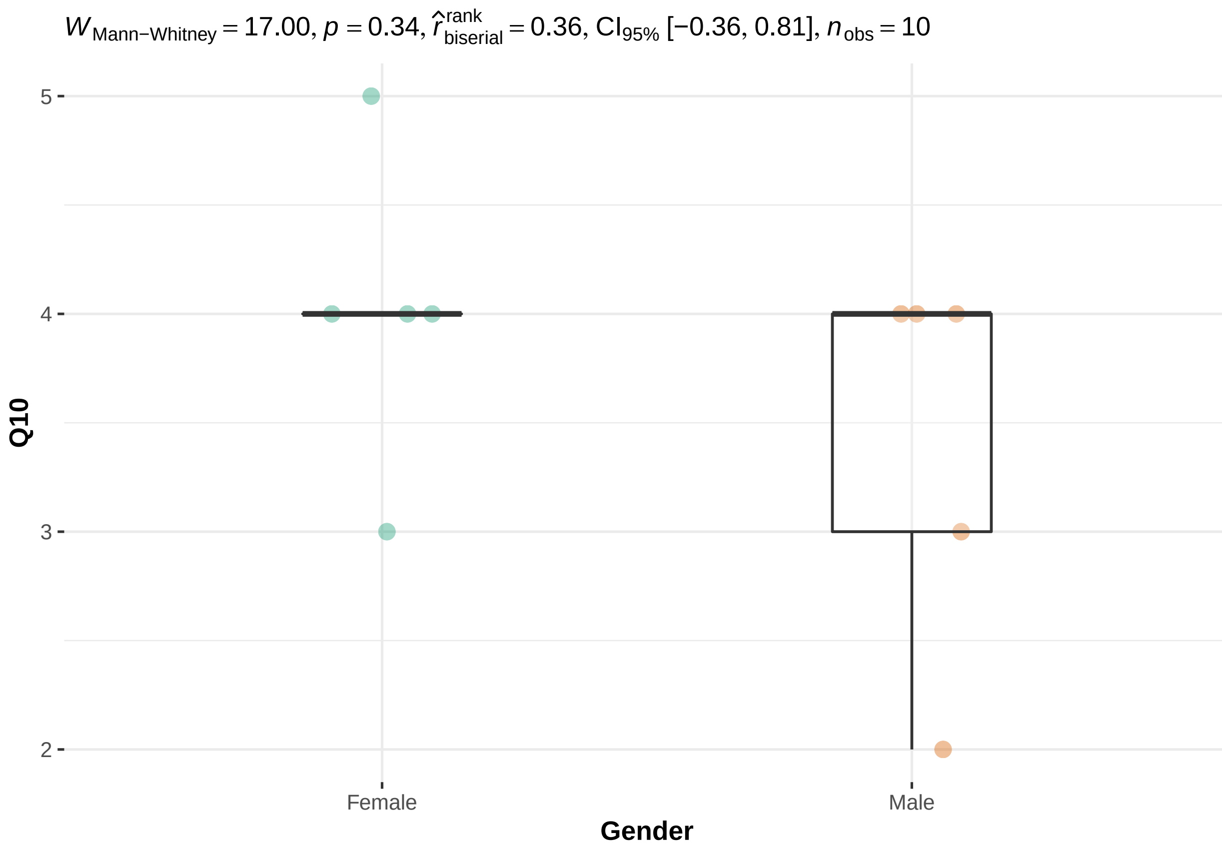

```
ggbetweenstats(data = df,x = Gender,y = Q8,
               plot.type = "box",type = "nonparametric",centrality.plotting = FALSE)
```



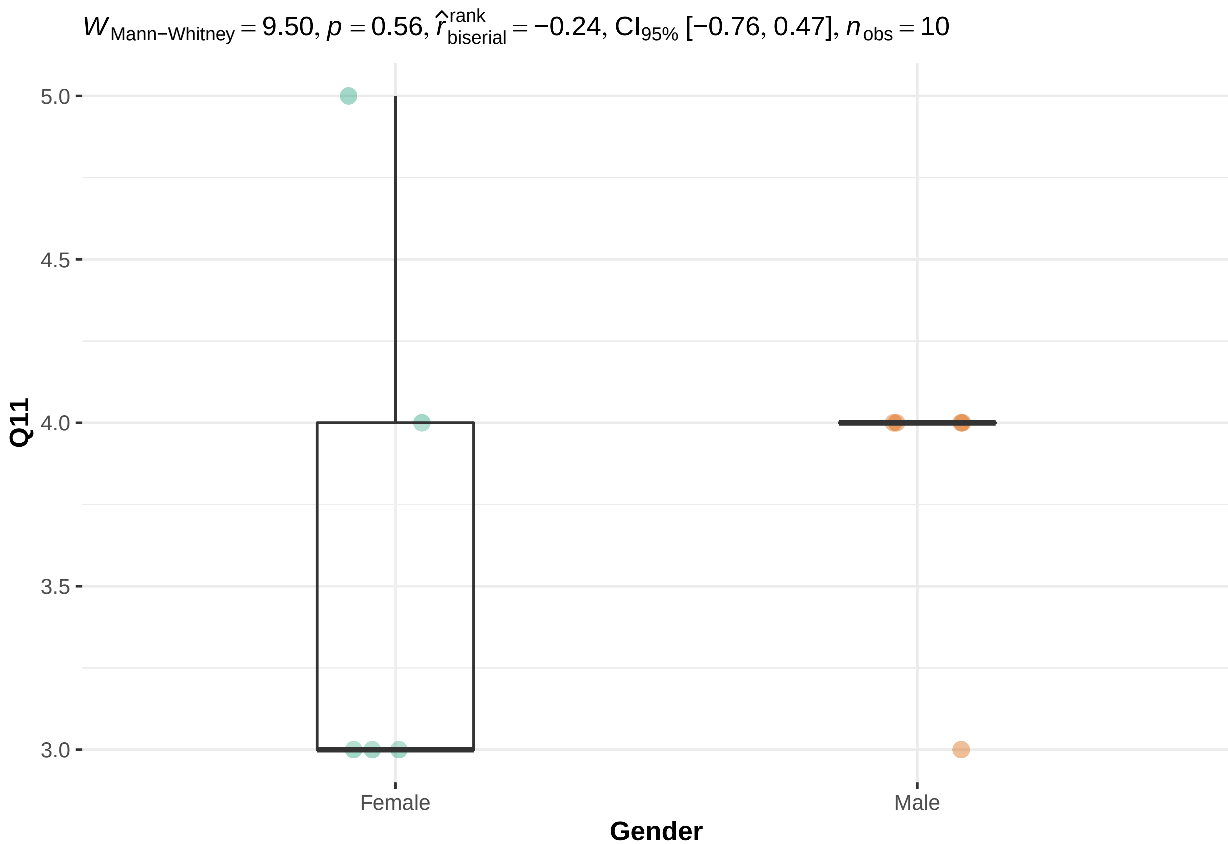

```
ggbetweenstats(data = df,x = Gender,y = Q9,
               plot.type = "box",type = "nonparametric",centrality.plotting = FALSE)
```



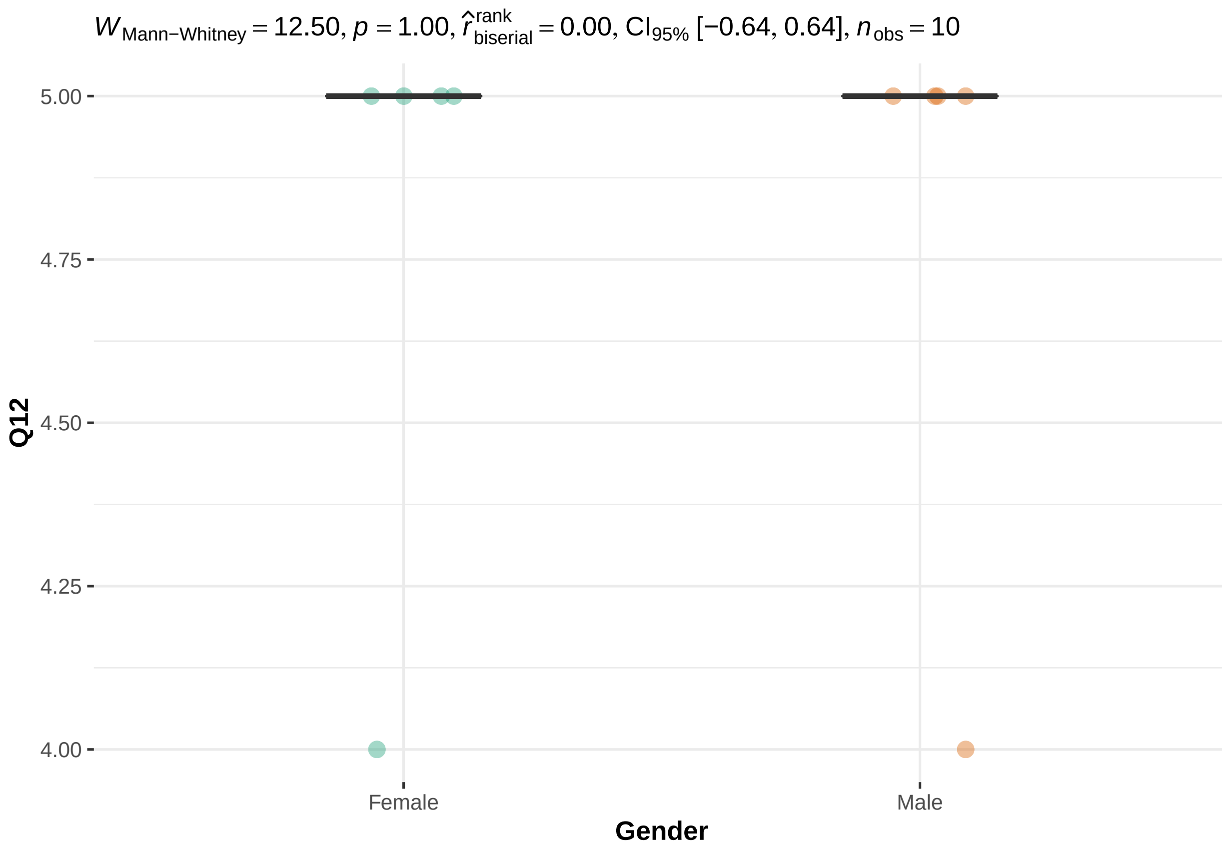

```r
ggbetweenstats(data = df,x = Gender,y = Q10,
               plot.type = "box",type = "nonparametric",centrality.plotting = FALSE)
```



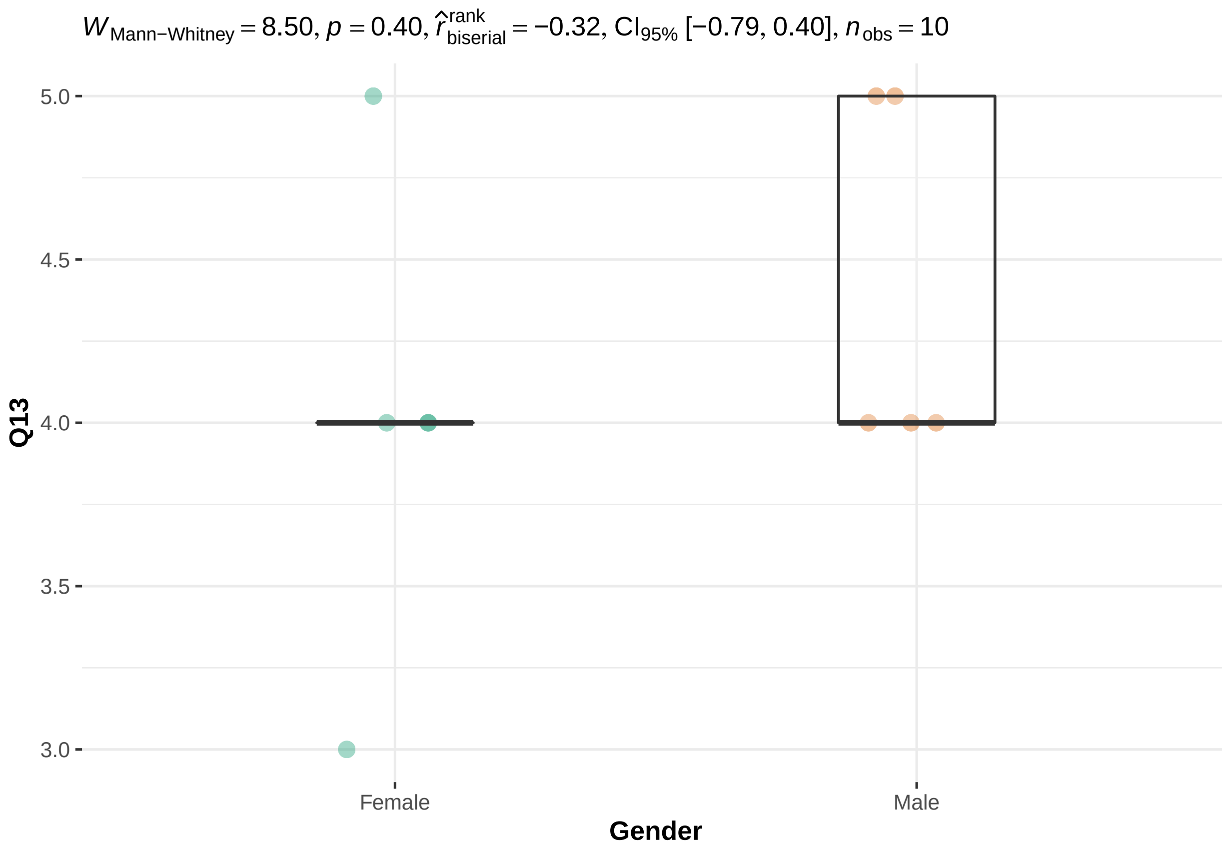

```
ggbetweenstats(data = df,x = Gender,y = Q11,
               plot.type = "box",type = "nonparametric",centrality.plotting = FALSE)
```




```
ggbetweenstats(data = df,x = Gender,y = Q12,
               plot.type = "box",type = "nonparametric",centrality.plotting = FALSE)
```




```
ggbetweenstats(data = df,x = Gender,y = Q13,
               plot.type = "box",type = "nonparametric",centrality.plotting = FALSE)
```



Conclusion: Based on this sample, for every question, I failed to reject the H0 again: the males and the females have the same criteria.