Final_exam_RCode

Haichuan Du

7/5/2020

```r
library(clinfun)
library(jmuOutlier)
library(NSM3)
```

```
## Warning: package 'NSM3' was built under R version 3.6.2

## Loading required package: combinat

##
## Attaching package: 'combinat'

## The following object is masked from 'package:utils':
##
##     combn

## Loading required package: MASS

## Loading required package: partitions

## Loading required package: survival

## fANCOVA 0.5-1 loaded
```

```r
library(lsmeans)
```

```
## Loading required package: emmeans

## Warning: package 'emmeans' was built under R version 3.6.2

## The 'lsmeans' package is now basically a front end for 'emmeans'.
## Users are encouraged to switch the rest of the way.
## See help('transition') for more information, including how to
## convert old 'lsmeans' objects and scripts to work with 'emmeans'.
```

#4

```r
a=c(1,2,3,4,5) #Some data with <ties>.
b=c(-1,-2,-3,-4,-50)
cor.test(a,b,method='pearson')
```

```
##
##  Pearson's product-moment correlation
##
## data:  a and b
## t = -1.9245, df = 3, p-value = 0.15
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9817471  0.4037351
## sample estimates:
##        cor
## -0.7432941
```

```
cor.test(a,b,method='spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  a and b
## S = 40, p-value = 0.01667
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
##  -1
```

##5a)

```
data=c(42, 18, 14.0, 36.0, 11.6, 19.0,15.6,23.8,24.4, 24.0, 21.0, 21.2,35.3, 22.5,16.9, 25.0, 23.1, 26.
groups=rep(1:3,c(6,6,6)) #Specifying which values are for which samples.
perm.f.test(data,groups) #Permutation F test and ANOVA F test. (jmuOutlier)
```

```
## [[1]]
## [1] "One-way ANOVA"
##
## [[2]]
##              Df  Sum Sq Mean Sq F value Pr(>F)
## treatment     2   29.61  14.807  0.2182 0.8065
## Residuals    15 1018.09  67.872
##
## [[3]]
## [1] "The p-value from the permutation F-test based on"
##
## [[4]]
## [1] "20000 simulations is 0.80545"
```

Ho: All three population have the same distribution and has no impact on phosphorous content vs. Ha: All three population dot not have the same distribution and has impact on phosphorous content

By doing Permutation F test. We have p values = 0.807 > 0.1. Therefore, we falil to reject Ho and cannot conclude Ha, that is, we do not have enough evidence at 0.1 level to conclude that all three population dot not have the same distribution.

##5b)

```
kruskal.test(data,groups) #Kruskal-Wallis test.
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  data and groups
## Kruskal-Wallis chi-squared = 1.1345, df = 2, p-value = 0.5671
```

Instead of the doing the test we used in part 5a), we can also use Kruskal-Wallis rank sum test, We have p values = 0.5671 > 0.1. Therefore, we falil to reject Ho and cannot conclude Ha, that is, we do not have enough evidence at 0.1 level to conclude that all three population dot not have the same distribution.

##6 Rank-Sum test can be used to compare two independent samples without making distributional assumptions. It is useful when the sample sizes are small, variances are heterogeneous, and the data are ordered categorical. It is more robust to the outliers and consistent against any alternative with p(x<y)!=yi, it nearly as efficient as the t-test if the data are normally distributed. 60 words)

##7

```
set.seed(0)
n=10 #Sample size to use.
nruns=1000 #Number of runs to do.
nrej1=0 #For the rank sum test.
nrej2=0 #Ansari-Bradley test.
nrej3=0 #Kolmogorov-Smirnov test.

for (run in 1:nruns){
  samp1=rnorm(n, mean = 0, sd = 1) #Sample of size n.
  samp2=rnorm(n, mean = 1, sd = 2 ) #Second sample of size n.
  if (wilcox.test(samp1,samp2)$p.value<0.05){nrej1=nrej1+1}
  if (ansari.test(samp1,samp2)$p.value<0.05){nrej2=nrej2+1}
  if (ks.test(samp1,samp2)$p.value<0.05){nrej3=nrej3+1}}
print(paste("Estimated power for the Rank Sum test is:", nrej1/nruns))         #0.234
```

```
## [1] "Estimated power for the Rank Sum test is: 0.234"
```

```
print(paste("Estimated power for Ansari-Bradley test is:", nrej2/nruns))       #0.187
```

```
## [1] "Estimated power for Ansari-Bradley test is: 0.187"
```

```
print(paste("Estimated power for Kolmogorov-Smirnov test is:", nrej3/nruns))   #0.134
```

```
## [1] "Estimated power for Kolmogorov-Smirnov test is: 0.134"
```

comment: By comparing three different tests, the rank sum test has the best power(0.234) to detect the difference between the two populations.

Ansari-Bradley test is designed to be sensitive to differences in scale, while The Rank Sum test is designed to be sensitive to differences in location and it is also consistent against any alternative with p(x<y)!=yi. However, for Kolmogorov-Smirnov test, it is sensitive to any alternative to equality distribution, if the sample size goes to infinity, the power of the k-s test will goes to 1, here we have a relatively small sample size, location differences are easiest to pick up, but power is still relatively weak compared to Rank Sum test. Because we adjusted the both location and scale, the power of Rank Sum test and Ansari-Bradley is higher than kolmogorov-Smirnov test. From here, it is easy to see how there can be datasets where the tests will yield different results.