统计计算与软件包第三次作业

姓名：王韦力　　学号：2019302010009

2022 年 4 月 10 日

6.3

分析过程 仿照例 6.9 即可. 为了方便 R 语言的实现，可以把例 6.9 中的代码写成关于 n 的函数，这样可以更简单地绘制 n 不同的多条曲线.

Rcode

```
f.power <- function(n=10, mu=seq(450,650,10)){
mu0 <- 500
sigma <- 100
m <- 1000
M <- length(mu)
power <- numeric(M)
for(i in 1:M){
mu1 <- mu[i]
pvalues <- replicate(m, expr = {
x <- rnorm(n,mean = mu1,sd = sigma)
ttest <- t.test(x, alternative = "greater", mu = mu0)
ttest$p.value
})
power[i] <- mean(pvalues<=0.05)
}
return(power)
}
power <- matrix(NA, nrow = 5, ncol = 21)
plot(x=NULL,xlim = c(450,650), ylim = c(0,1), ylab="power",xlab=bquote(theta))
legend(x='bottomright',lty = 1,col = c(1:5), cex = 0.8, legend = c(expression(n
== 10),
expression(n == 20),
expression(n == 30),
expression(n == 40),
expression(n == 50)))
for(i in 1:5) {
power[i,] <- f.power(n=10*i)
lines(mu,power[i,],col=i)
}
```

运行结果 R 输出的图像如图 1所示.

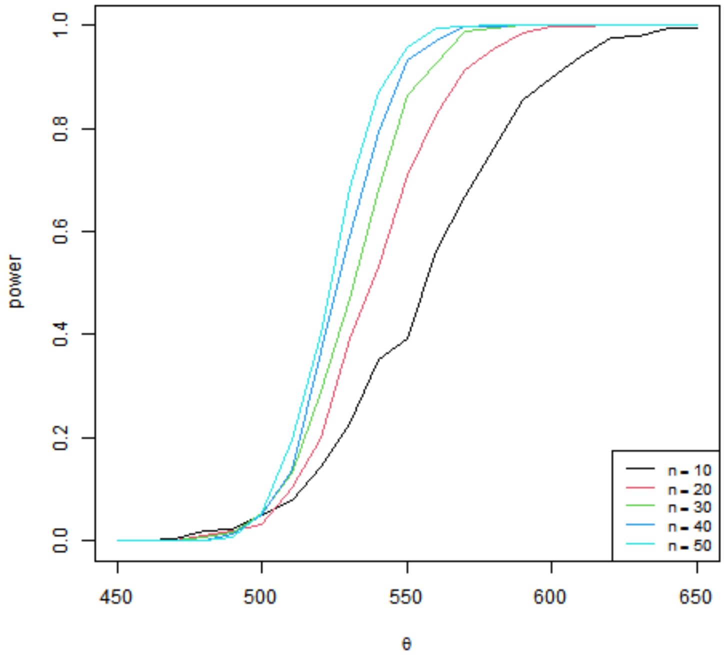

图 1: 练习 6.3 中样本大小 $n = 10, 20, 30, 40$ 和 50 时 t 检验的经验功效曲线

结论 由图 1可见，当 $\theta \le 500$ 时，各样本大小的经验功效 $\pi(\theta)$ 十分接近（都约等于 0）. 当 $\theta > 500$ 时，明显可以看出，经验功效 $\pi(\theta)$ 随着 n 的增大而增大. 当 $\theta \to +\infty$ 时，各样本大小的经验功效都趋向于 1.

6.4

分析过程 设 $X_1, \dots, X_n$ 是来自对数正态总体 X 的样本，记 $Y = \log X$，则 $Y \sim N(\mu, \sigma^2)$. 由题意，μ 和 σ 未知. 把 $Y_i = \log X_i$ 看作来自总体 Y 的样本，则 $T = \dfrac{\bar{Y} - \mu}{S_Y/\sqrt{n}} \sim t(n-1)$，于是

$$P(|T| > t_{1-\frac{\alpha}{2}}(n-1)) = P\left(\left|\bar{Y} - \mu\right| \le \frac{S_Y t_{1-\frac{\alpha}{2}}(n-1)}{\sqrt{n}}\right) = 1 - \alpha.$$

由此可以得到置信水平的蒙特卡罗估计（取 $\alpha = 0.05$），对应的置信区间为

$$\left[\bar{Y} - S_Y t_{1-\frac{\alpha}{2}}(n-1)\sqrt{n}, \bar{Y} - S_Y t_{1+\frac{\alpha}{2}}(n-1)\sqrt{n}\right].$$

我们在 Rcode 部分中令 $\mu = 0$，$\sigma = 2$，$n = 20$，$m = 1000$（重复次数）来进行验证.

Rcode

```
n <- 20
alpha <- 0.05
UCL <- replicate(1000,expr = {
x <- rlnorm(n, meanlog = 0, sdlog = 2)
abs(mean(log(x)-0))<=sd(log(x))*qt(1-alpha/2, df=n-1)/sqrt(n) })
```

```
sum(UCL)
mean(UCL)
```

运行结果 Rcode 最后两行的运行结果为

```
> sum(UCL)
[1] 948
> mean(UCL)
[1] 0.948
```

结论 由运行结果可见，经验置信水平与理论置信水平很接近.

6.5

分析过程 本题的思路与练习 6.4 的类似，只是把总体 X 服从的分布改为 $\chi^2(2)$，置信区间为关于样本 $X_i(i=1,2,\dots,n)$ 的 t 区间

$$\left[\bar{X}-S_X t_{1-\frac{\alpha}{2}}(n-1)\sqrt{n}, \bar{X}-S_X t_{1+\frac{\alpha}{2}}(n-1)\sqrt{n}\right].$$

Rcode

```
n <- 20
alpha <- 0.05
UCL <- replicate(1000,expr = {
x <- rchisq(n,df=2)
abs(mean(x)-2)<=sd(x)*qt(1-alpha/2, df=n-1)/sqrt(n)
})
sum(UCL)
mean(UCL)
```

运行结果 Rcode 最后两行的运行结果为

```
> sum(UCL)
[1] 909
> mean(UCL)
[1] 0.909
```

结论 由运行结果可见此时 t 区间对均值的覆盖率为 90.9%，略低于 95%. 但在样本服从 $\chi^2(2)$ 时，方差区间的覆盖率只有约 77%（参考例 6.6），所以在把样本的分布改为 $\chi^2(2)$ 的情形下，t 区间比方差区间的表现得更加稳健（robust）.

6.8

分析过程 依例 6.16 重复模拟过程，并对样本进行等方差 F 检验. 令两组抽样的样本大小均为 n，再分别令 $n=10,20,100$，比较 Count Five 检验和 F 检验的功效.

Rcode

```
x1 <- rnorm(20,0,sd=1)
x2 <- rnorm(20,0,sd=1.5)
```

```
y <- c(x1,x2)
group <- rep(1:2,each=length(x1))
boxplot(y~group,boxwex=0.3,xlim=c(0.5,2.5),main="")
points(group,y)
range(x1)
range(x2)
# In x1, pick out the points that are out of the range of x2:
i <- which(x1<min(x2) | x1>max(x2))
# In x2, pick out the points that are out of the range of x1:
j <- which(x2>max(x1) | x2<min(x1))
x1[i]
x2[j]
# figure out the total number
out1 <- sum(x1>max(x2))+sum(x1<min(x2))
out2 <- sum(x2>max(x1))+sum(x2<min(x1))
max(c(out1,out2))
# F-test
alpha <- 0.055
var.test(x,y,alternative = "two.sided")$p.value<1-alpha
# a function of Count Five test
count5test <- function(x,y){
X <- x-mean(x)
Y <- y-mean(y)
outx <- sum(X>max(Y))+sum(X<min(Y))
outy <- sum(Y>max(X))+sum(Y<min(X))
# return 1 (reject H_0) or 0 (do not reject H_0)
return(as.integer(max(c(outx,outy))>5))
}
compare.5.F <- function(n1,n2,sigma1,sigma2){
power <- numeric(2)
power <- apply(replicate(m, expr={
x <- rnorm(n1,0,sigma1)
y <- rnorm(n2,0,sigma2)
c(count5test(x,y),var.test(x,y,alternative = "two.sided")$p.value<1-alpha)
}),1,mean)
return(power)
}
sigma1 <- 1
sigma2 <- 1.5
m <- 1000
compare.5.F(10,10,sigma1,sigma2)
```

```
compare.5.F(20,20,sigma1,sigma2)
compare.5.F(100,100,sigma1,sigma2)
```

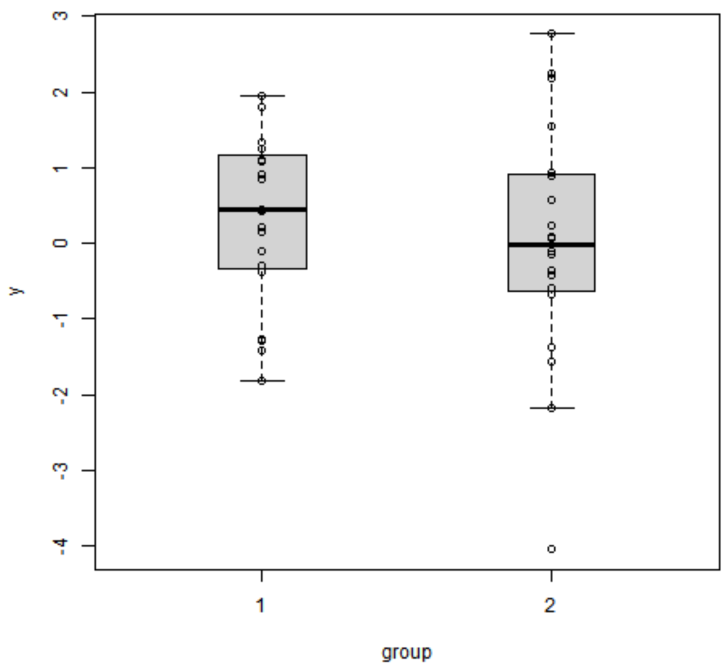

图 2: 练习 6.8 中 Count Five 统计量端点的箱型图

运行结果与结论 Rcode 输出的箱型图如图 2所示. 结合图 2以及如下结果

```
> max(c(out1,out2))
[1] 5
```

可知 Count Five 检验刚好拒绝等方差假设. 而由

```
> var.test(x,y,alternative = "two.sided")$p.value<1-alpha
[1] TRUE
```

可知在显著水平为 $\alpha = 0.055$ 的情形下, F 检验也拒绝等方差假设.

Rcode 最后三行运行的结果如下:

```
> compare.5.F(10,10,sigma1,sigma2)
[1] 0.111 0.962
> compare.5.F(20,20,sigma1,sigma2)
[1] 0.319 0.985
> compare.5.F(100,100,sigma1,sigma2)
[1] 0.848 1.000
```

可见随着样本大小的增加, 两种检验的功效都有所提高, Count Five 检验的功效提升的尤为明显, 但是还是不如 F 检验.

6.9

分析过程 利用

$$\hat{G} = \frac{1}{n^2 \bar{x}} \sum_{i=1}^{n} (2i - n - 1) x_{(i)} \tag{1}$$

来计算 $\hat{G}$，其中顺序统计量可以用 sort() 函数来获得. 对服从标准对数正态的总体 X，取样本大小为 $n = 20$ 的样本，用 1式计算 $\hat{G}$. 重复上述过程 $m = 1000$ 次，估计 $\hat{G}$ 的均值、中位数和十分位数.

对均匀分布（这里取 Uniform(0,1)）和 Bernoulli(0.1) 分布也重复上述过程，并对每种情况绘制重复试验的密度直方图.

Rcode

```
gini <- function(x,mu="NULL"){ # compute Gini ratio
n <- length(x)
x.sort=sort(x)
coe <- 2*c(1:n)-n-1
if(mu=="NULL")
mu <- mean(x)
G <- sum(coe*x.sort)/(mu*n^2)
return(G)
}
mmq.list <- function(x){
x.mean <- mean(x,na.rm = TRUE)
x.median <- median(x,na.rm = TRUE)
x.deciles <- quantile(x,probs = c(1:9)/10,na.rm = TRUE)
result <- list(x.mean,x.median,x.deciles)
names(result) <- c("mean","median","deciles")
return(result)
}
n <- 20
m <- 1000
norm.gini <- unif.gini <- binom.gini <- numeric(m)
for(i in 1:m){
x <- rlnorm(n, meanlog = 0, sdlog = 1)
y <- runif(n)
z <- rbinom(n,size = 1,prob = 0.1)
norm.gini[i] <- gini(x)
unif.gini[i] <- gini(y)
binom.gini[i] <- gini(z)
}
mmq.list(norm.gini)
mmq.list(unif.gini)
mmq.list(binom.gini)
```

```
draw.hist <- function(x){
hist(x,prob = TRUE, xlim = c(max(0.75*min(x),0),min(1.25*max(x),1)), main="")
}
draw.hist(norm.gini)
draw.hist(unif.gini)
draw.hist(binom.gini[which(binom.gini>0)])
```

运行结果与结论 对数标准正态样本、均匀分布样本和 Bernoulli(0.1) 分布样本的基尼系数的均值、中位数与十分位数分别由 mmq.list(norm.gini)，mmq.list(unif.gini) 和 mmq.list(binom.gini) 给出.

```
> mmq.list(norm.gini)
$mean
[1] 0.4767836
$median
[1] 0.4728739
$deciles
10%        20%        30%        40%        50%        60%
70%        80%
0.3768939 0.4098800 0.4317879 0.4526885 0.4728739 0.4931606 0.5140061 0.5458180
90%
0.5817892
> mmq.list(unif.gini)
$mean
[1] 0.3226978
$median
[1] 0.3235838
$deciles
10%        20%        30%        40%        50%        60%
70%        80%
0.2534111 0.2770055 0.2927798 0.3086732 0.3235838 0.3373053 0.3504720 0.3664799
90%
0.3926358
> mmq.list(binom.gini)
$mean
[1] 0.8851429
$median
[1] 0.9
$deciles
10%  20%  30%  40%  50%  60%  70%  80%  90%
0.80 0.85 0.85 0.90 0.90 0.90 0.95 0.95 0.95
```

它们的密度直方图分别见图 3、图 4和图 5.

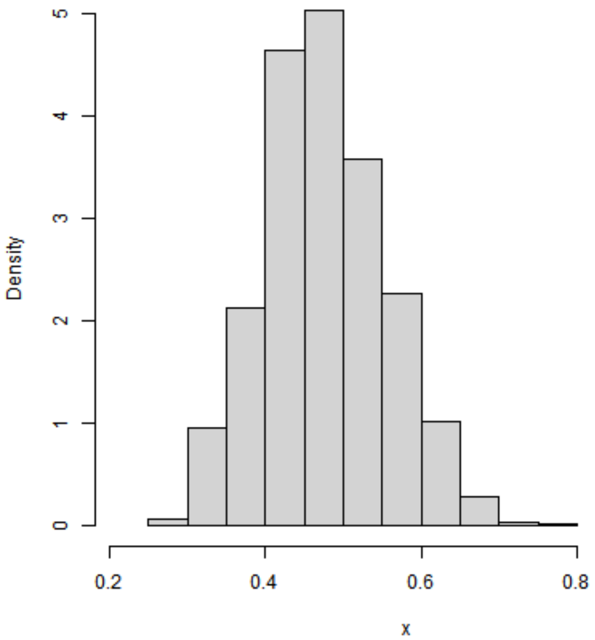

图 3: 练习 6.9 中标准正态分布样本的基尼系数的密度直方图

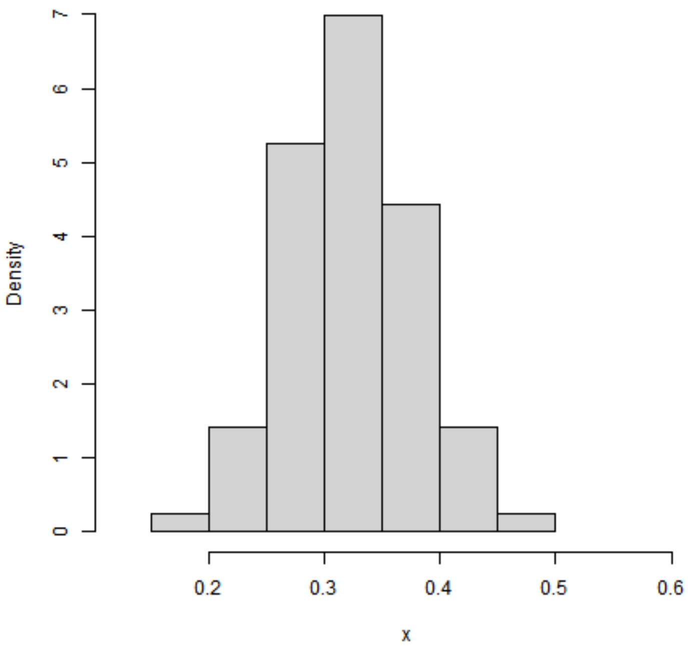

图 4: 练习 6.9 中均匀分布样本的基尼系数的密度直方图

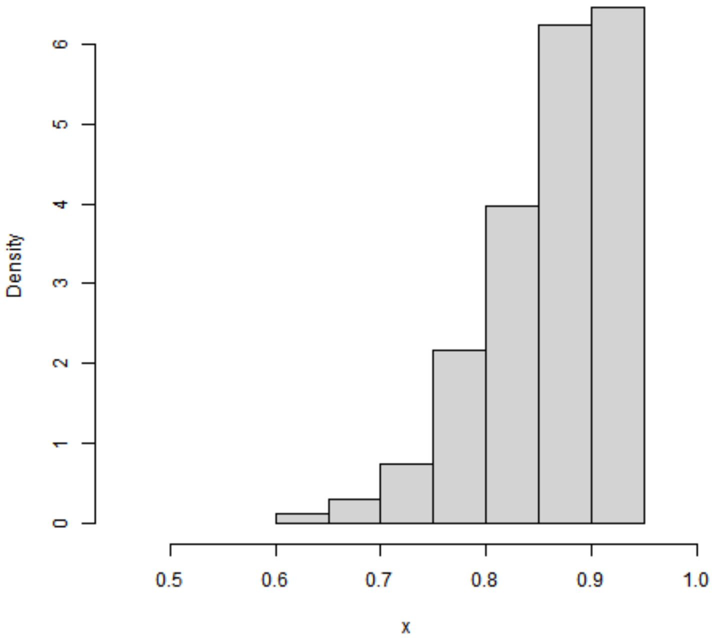

图 5: 练习 6.9 中 Bernoulli(0.1) 分布样本的基尼系数的密度直方图

6.A

分析过程 参考课本例 6.7 研究 t 检验的经验第一类错误率. 注意 (i)(ii)(iii) 样本总体均值均为 1.

Rcode

```
n <- 20
alpha <- 0.05
mu0 <- 1
m <- 10000
#(i)
p.i <- numeric(m)
for (j in 1:m) {
x <- rchisq(n,df=1)
ttest.i <- t.test(x,alternative = "two.sided",mu=mu0)
p.i[j] <- ttest.i$p.value
}
p.i.hat <- mean(p.i<alpha)
se.i.hat <- sqrt(p.i.hat*(1-p.i.hat)/m)
result.i <- c(p.i.hat,se.i.hat)
#(ii)
p.ii <- numeric(m)
for (j in 1:m) {
y <- runif(n,min = 0,max = 2)
```

```
ttest.ii <- t.test(y,alternative = "two.sided",mu=mu0)
p.ii[j] <- ttest.ii$p.value
}
p.ii.hat <- mean(p.ii<alpha)
se.ii.hat <- sqrt(p.ii.hat*(1-p.ii.hat)/m)
result.ii <- c(p.ii.hat,se.ii.hat)
#(iii)
p.iii <- numeric(m)
for (j in 1:m) {
z <- rexp(n,rate = 1)
ttest.iii <- t.test(y,alternative = "two.sided",mu=mu0)
p.iii[j] <- ttest.iii$p.value
}
p.iii.hat <- mean(p.iii<alpha)
se.iii.hat <- sqrt(p.iii.hat*(1-p.iii.hat)/m)
result.iii <- c(p.iii.hat,se.iii.hat)
print(result.i)
print(result.ii)
print(result.iii)
```

运行结果 Rcode 最后三行的运行结果为

```
> print(result.i)
[1] 0.109100000 0.003117646
> print(result.ii)
[1] 0.053300000 0.002246311
> print(result.iii)
[1] 0.080500000 0.002720657
```

结论 由运行结果可见三种情形下犯第一类错误的概率都比较接近 $\alpha = 0.05$.

6.B

分析过程 设 X 和 Y 为两个样本总体，我们要检验

$$H_0 : X \text{ 与 } Y \text{ 不相关} \quad \text{vs.} \quad H_1 : X \text{ 与 } Y \text{ 相关}.$$

当 (X, Y) 服从二元正态分布且 X 与 Y 独立（不相关）时，不妨令此分布为 $N(\boldsymbol{\mu}, \Sigma)$，其中

$$\boldsymbol{\mu} = (0,0)', \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$

程序设计思路可以参考课本的例 6.9，程序的循环改变点为 $Cov(X, Y)$. 令 $Z = e^Y$，则 X 也与 Z 独立，从而不相关. 此时检验变为

$$H_0 : X \text{ 与 } Z \text{ 不相关} \quad \text{vs.} \quad H_1 : X \text{ 与 } Z \text{ 相关}.$$

我们将在下文指出此时基于 ρ_s 和 τ 的非参数检验的功效都要比相关性检验的功效高.

Rcode

```
library(MASS)
n <- 20
m <- 1000
mu <- rep(0,2)
co <- seq(-1.8,1.8,0.1)
alpha <- 0.05
M <- length(co)
Sigma <- vector("list",M)
for (i in 1:M) {
Sigma[[i]] <- matrix(c(1,co[i],co[i],4),ncol = 2)
}
p <- function(method="pearson"){
power <- numeric(M)
for (j in 1:M) {
Sigma1<- Sigma[[j]]
pvalues <- replicate(m,expr = {
X<- mvrnorm(n,mu,Sigma1)
x <- X[,1]
y <- X[,2]
cortest <- cor.test(x,y,method=method)
cortest$p.value
})
power[j] <- mean(pvalues<=alpha)
}
return(power)
}
power <- matrix(rep(0,3*M),nrow = 3)
power[1,] <- p()
power[2,] <- p("spearman")
power[3,] <- p("kendall")
plot(NULL,NULL,xlim=range(co),ylim=c(0,1),xlab="Cov(X,Y)",ylab="power")
for (i in 1:3) {
lines(co,power[i,],lty=i)
}
legend(x='top',lty = 1:3, cex = 0.8,legend = c("Pearson","Spearman","Kendall"))
p1 <- function(method="pearson"){
power <- numeric(M)
for (j in 1:M) {
Sigma1<- Sigma[[j]]
```

```
pvalues <- replicate(m,expr = {
X<- mvrnorm(n,mu,Sigma1)
x <- X[,1]
y <- exp(X[,2]) # here is the change
cortest <- cor.test(x,y,method=method)
cortest$p.value
})
power[j] <- mean(pvalues<=alpha)
}
return(power)
}
power1 <- matrix(rep(0,3*M),nrow = 3)
power1[1,] <- p1()
power1[2,] <- p1("spearman")
power1[3,] <- p1("kendall")
plot(NULL,NULL,xlim=range(co),ylim=c(0,1),xlab="Cov(X,Z)",ylab="power")
for (i in 1:3) {
lines(co,power1[i,],lty=i)
}
legend(x='top',lty = 1:3, cex = 0.8,legend = c("Pearson","Spearman","Kendall"))
```

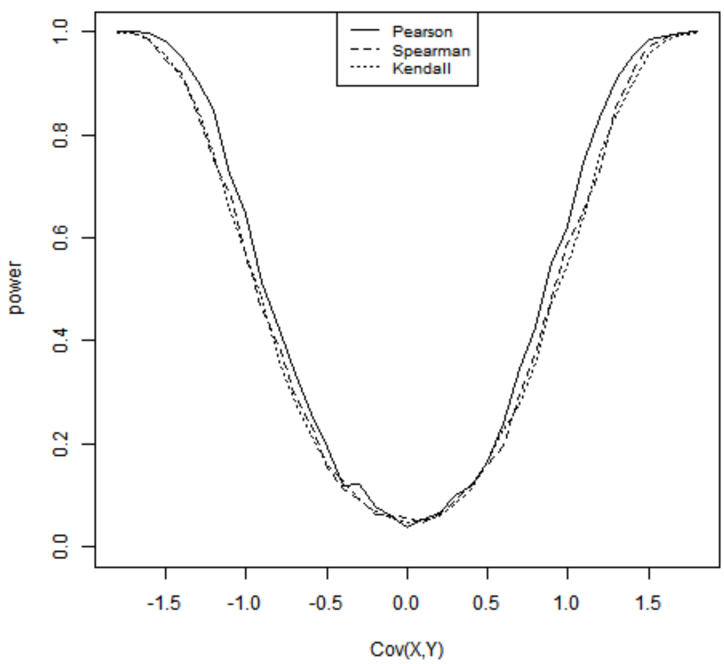

图 6: 习题 6.B 中检验经验功效的示意图 1

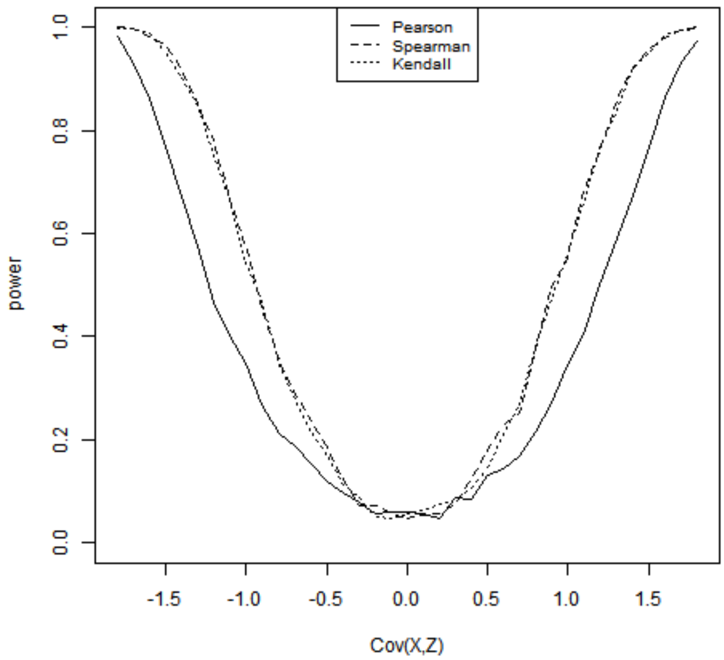

图 7: 习题 6.B 中检验经验功效的示意图 2

运行结果与结论 Rcode 输出的图像如图 6和图 7所示. 从图 6中可以看出，相关性检验的功效曲线（实线）几乎始终在非参数检验的功效曲线（两条虚线）的上方，而在图 7中情况则恰恰相反. 这就从经验上说明了，当抽样分布是二元正态分布时，非参数检验的功效要比相关性检验的功效低，而对于我们选择的二元分布 (X, Z)，非参数检验的功效要更高.