2019302030053-胡哲-第二次作业

第六章习题

6-3

利用example6.9的代码，先封装好成一个函数，之后对不同n的取值获得数据贴上标签，最后利用ggplot2中颜色和形状映射的概念即可在一幅图上产生不同形状、颜色的线条，并自动绘制图例。

具体代码如下

```
#### 6-3 ####
example_7_9 <- function(n){
m <- 1000
mu0 <- 500
sigma <- 100
mu <- c(seq(450, 650, 10)) #alternatives
M <- length(mu)
power <- numeric(M)
for (i in 1:M) {
  mu1 <- mu[i]
  pvalues <- replicate(m, expr = {
    #simulate under alternative mu1
    x <- rnorm(n, mean = mu1, sd = sigma)
    ttest <- t.test(x,
                    alternative = "greater", mu = mu0)
    ttest$p.value } )
```

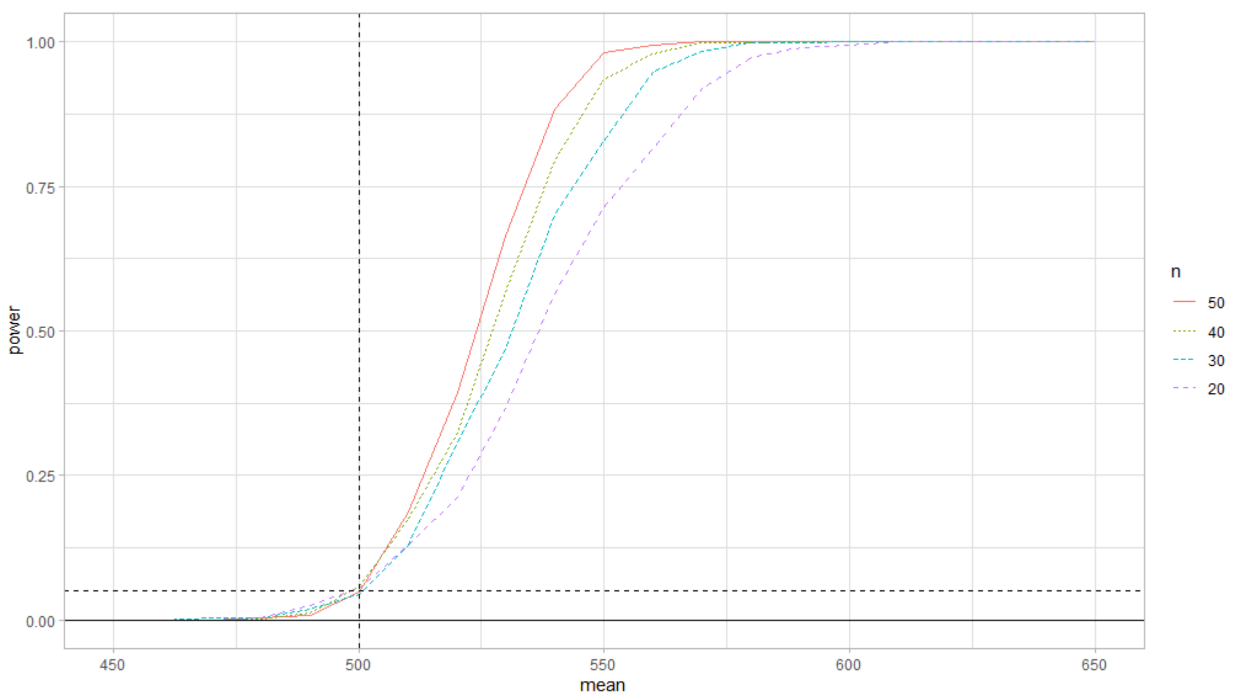
```r
  power[i] <- mean(pvalues <= .05)
}
se <- sqrt(power * (1-power) / m)
df <- data.frame(mean=mu, power=power,
                 upper=power+2*se, lower=power-2*se,n=factor(n))
return(df)
}
# 封装成一个函数


data <- data.frame()
for (n in seq(20,50,10)) {
  df <- example_7_9(n)
  data <- rbind(df,data)
}
# 将不同取值n的df合并成一个data

library(ggplot2)

ggplot(data, aes(x=mean, y=power,col=n,linetype=n)) +
  geom_line() +
  geom_vline(xintercept=500, lty=2) +
  geom_hline(yintercept=c(0,.05), lty=1:2) +
  theme_light() # 主题可以自己改我喜欢用这个
 # 利用ggplot2的颜色和形状映射概念自动生成图例(没学过ggplot2包的可以看看R数据科学)
```

绘制图像如下:



根据图像可知，当样本量更大时，经验功效会更高，随$\hat{\theta}$增加趋近1的速度更快。

6-4

X服从对数正态分布，Y=ln(x)~N(μ,σ^2)，所以可以直接利用Y进行估计经验置信区间,经验置信区间为:

$$[\hat{\theta}-\frac{SE\times t_{\alpha/2}}{\sqrt{n}},\hat{\theta}+\frac{SE\times t_{\alpha/2}}{\sqrt{n}}]$$

具体代码如下:

```
exercise_6_4 <- function(seed=123){
set.seed(seed)
n <- 20 # X服从对数正态分布，Y=ln(x)~N(μ,σ^2)，所以可以直接利用Y进行估计，再代入X即可
alpha <- .05
m <- 1000
cv.t<-sapply(1:m,FUN= function(o){
  y<-rnorm(n)
  c<-qt(0.975,n-1) # 0.975 quantile of t-distribution
  m<-mean(y) # estimate of mean
  se<-sqrt(var(y)) # estimate of standard error
  as.numeric((m-c*se/sqrt(n)<0)&(m+c*se/sqrt(n)>0)) # ci
})
level <- mean(cv.t) # mean of  Monte Carlo experiment

return(data.frame(level=level))
}

exercise_6_4()
```

置信水平估计如下:

```
  level
  0.946
```

和0.95的真实值非常接近

6-6

t方法的估计类似上题，不同的是该题x从卡方分布中取样，为了做对比，将example6.4中方法做对比。

具体代码如下:

```
exercise_6_5 <- function(seed=123){
set.seed(seed)
n<-20
c<-qt(0.975,n-1) # 0.975 quantile of t-distribution
m <- 1000
cv.t<-sapply(1:m,FUN= function(o){
x<-rchisq(n,2)  # 注意这里的x是从卡方分布取样
m<-mean(x) # estimate of mean
se<-sqrt(var(x)) # estimate of standard error
as.numeric((m-c*se/sqrt(n)<2)&(m+c*se/sqrt(n)>2)) # ci
})
level1 <- mean(cv.t) # mean of  Monte Carlo experiment


# 我们可以得出概率小于0.95,example6.4使用卡方分布来估计方差（真值为4)
alpha <- .05
UCL <- replicate(1000, expr = {
x <- rchisq(n,2)
(n-1) * var(x) / qchisq(alpha, df = n-1)
} )
#计算包含sigma^2=4的区间数
level2 <- sum(UCL > 4)/m
return(data.frame(level1,level2))
}

exercise_6_5(1012)
# 我们可以看到结果远小于0.95，因此t-区间更稳健
```

输出结果如下:

```
  level1 level2
   0.908  0.794
```

两者显著水平的估计都小于0.95，但t方法的估计距离0.95更接近，说明t方法更稳健。

6-8

count-5检验利用书上所给代码，F检验利用var.test函数，对不同样本量(20,200,1000)比较功效。

具体代码如下：

```
exercise_6_8 <- function(){
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(as.integer(max(c(outx,outy)) > 5))
}
n <- c(20,200,1000)#分别对应小样本、中样本和大样本
mu1 <- mu2 <- 0
sigma1 <- 1
sigma2 <- 1.5
m <- 10000
power1 <- power2 <- numeric(length(n))
set.seed(1234)
for(i in 1:length(n)){
  power1[i] <- mean(replicate(m,expr = {
    x <- rnorm(n[i],mu1,sigma1)
    y <- rnorm(n[i],mu2,sigma2)
    x <- x - mean(x)
    y <- y - mean(y)
    count5test(x,y)
  }))
  pvalues <- replicate(m,expr={
    x <- rnorm(n[i],mu1,sigma1)
    y <- rnorm(n[i],mu2,sigma2)
    Ftest <- var.test(x, y, ratio = 1,
                      alternative = c("two.sided"),
                      conf.level = 0.945, ...)
    Ftest$p.value})
  power2[i] <- mean(pvalues<=0.055)
}
return(data.frame(power1,power2))
}
exercise_6_8()
```

结果如下：

```
power1 power2
0.3128 0.4118
0.9475 0.9999
0.9980 1.0000
```

power1为count5功效，power2为F-test功效，可见在小、中、大样本下，F检验的功效都要更高。

6-9

按照书上所给公式计算Gini系数，重复多次绘制密度直方图，为了表示从不同分布中抽样，构建函数时将分布作为形参传进去。

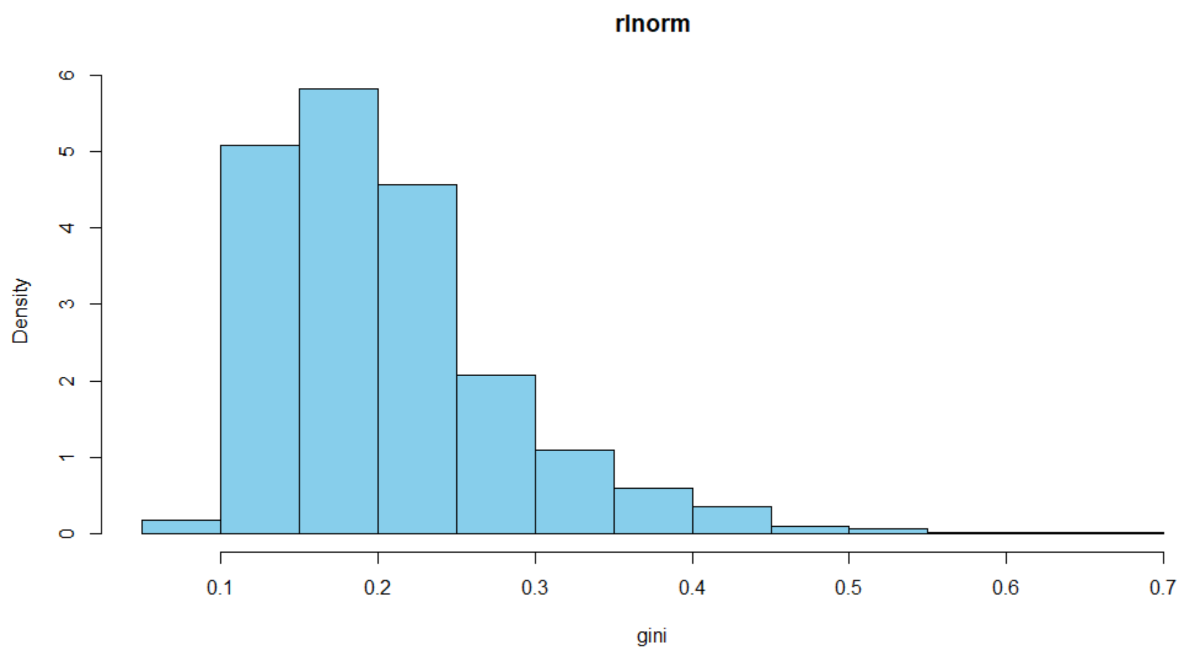
具体代码如下：

```
exercise_6_9 <- function(distribution=c('rlnorm','uniform','Bernoulli')){
n <- 20 # 样本量
size <- 1000 # 重复次数
ginifun <- function()
{
  if (distribution == 'rlnorm')x <- sort(rlnorm(n))
  else if(distribution == 'uniform') x <- sort(runif(n,0,1))
  else x <- sort(rbinom(n,size = 100,prob = .1))
  m=mean(x)
  sum=0
  for (k in n) {
    t=(2*k-n-1)*x[k]
    sum=sum+t
  }
  gini=sum/(n^2*m)
}
res <- replicate(size,expr = ginifun())
hist(as.numeric(res), prob = TRUE, main = distribution)
return(data.frame(mean = mean(res),median = median(res),quantile =
quantile(res,seq(.1,.9,.1))))
```

```
}
exercise_6_9(distribution = 'rlnorm')
exercise_6_9(distribution = 'uniform')
exercise_6_9(distribution = 'Bernoulli')
```

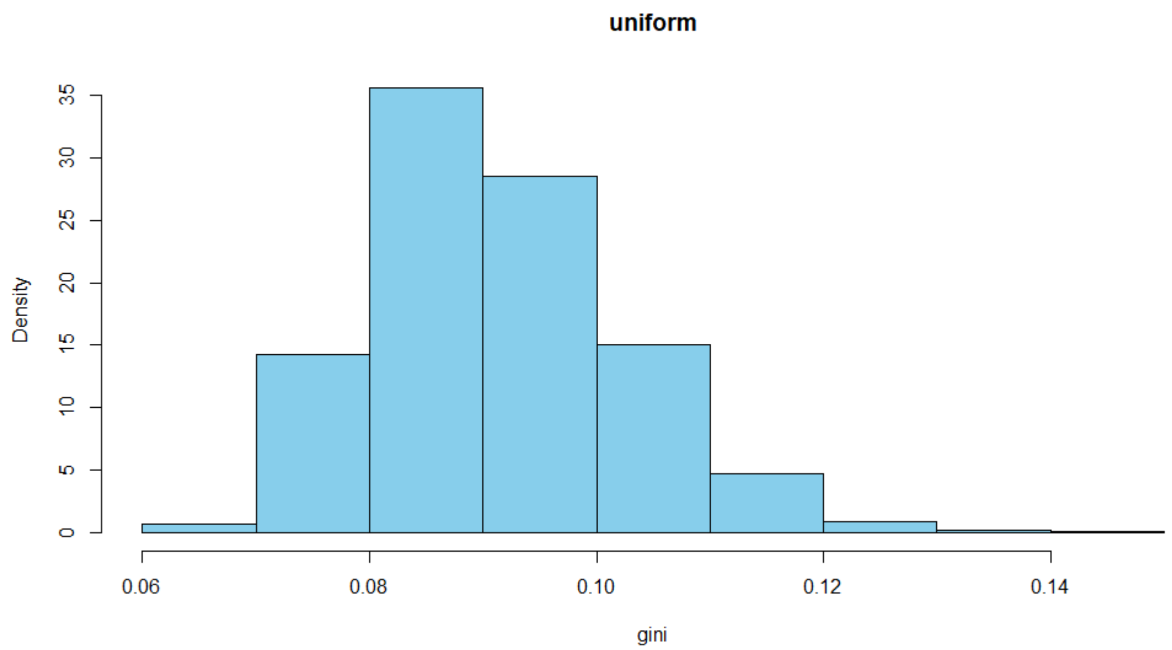
对数正态分布均值，中位数，十分位数和图象：

```
         mean    median  quantile
10% 0.2098996 0.1916062 0.1301766
20% 0.2098996 0.1916062 0.1476594
30% 0.2098996 0.1916062 0.1620432
40% 0.2098996 0.1916062 0.1748635
50% 0.2098996 0.1916062 0.1916062
60% 0.2098996 0.1916062 0.2089830
70% 0.2098996 0.1916062 0.2306749
80% 0.2098996 0.1916062 0.2619465
90% 0.2098996 0.1916062 0.3119318
```



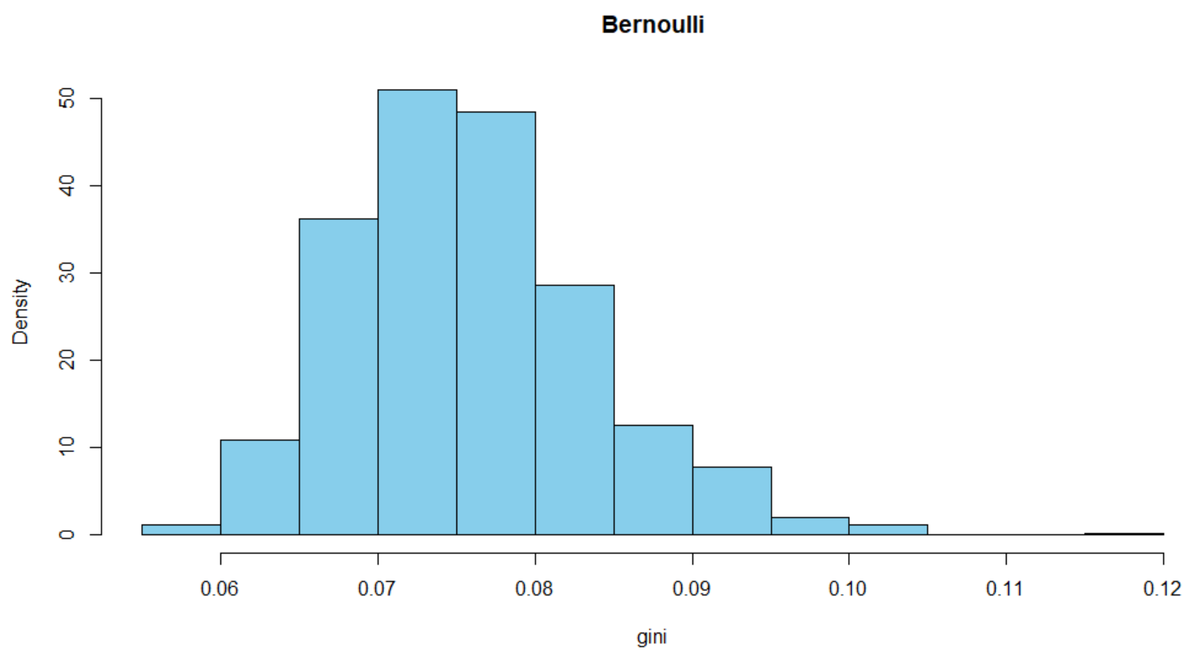
均匀分布均值，中位数，十分位数和图象：

```
         mean     median   quantile
10% 0.09150293 0.08972052 0.07859503
20% 0.09150293 0.08972052 0.08208537
30% 0.09150293 0.08972052 0.08477673
40% 0.09150293 0.08972052 0.08760314
50% 0.09150293 0.08972052 0.08972052
60% 0.09150293 0.08972052 0.09356417
70% 0.09150293 0.08972052 0.09688078
80% 0.09150293 0.08972052 0.10019635
90% 0.09150293 0.08972052 0.10605506
```



伯努利分布均值，中位数，十分位数和图象：

```
         mean     median   quantile
10% 0.07585646 0.07511628 0.06658879
20% 0.07585646 0.07511628 0.06909091
30% 0.07585646 0.07511628 0.07136150
40% 0.07585646 0.07511628 0.07342161
50% 0.07585646 0.07511628 0.07511628
60% 0.07585646 0.07511628 0.07702703
70% 0.07585646 0.07511628 0.07916667
80% 0.07585646 0.07511628 0.08181818
90% 0.07585646 0.07511628 0.08636364
```



6-A

利用$T=\frac{\bar{X}-\mu}{S/\sqrt{n}}\sim t(n-1)$，该题中X从非正态分布中抽样，并取五种不同的样本量进行测试

具体代码如下：

```
exercise_6_A <- function(seed){
set.seed(123)
num<-c(50,100,200,500,1000) # Estimate the Type-I error for different sizes.
m<-10000

er<-NULL
for (n in num){
  cv<-qt(0.975,n-1)
  er1<-mean(sapply(1:m,FUN = function(o){
  x<-rchisq(n,1)
  m<-mean(x)
  se<-sqrt(var(x))
  abs((m-1)*sqrt(n)/se)>=cv
})) # 估计卡方分布的第一类错误
```

```
  er2<-mean(sapply(1:m,FUN = function(o){
  x<-runif(n,0,2)
  m<-mean(x)
  se<-sqrt(var(x))
  abs((m-1)*sqrt(n)/se)>=cv
}))   # 估计均匀分布的第一类错误
  er3<-mean(sapply(1:m,FUN = function(o){
  x<-rexp(n,1)
  m<-mean(x)
  se<-sqrt(var(x))
  abs((m-1)*sqrt(n)/se)>=cv
}))  # 估计指数分布的第一类错误
er<-cbind(er,c(er1,er2,er3))
}
colnames(er)<-num
rownames(er)<-c("chi(1)","U(0,2)","exp(1)")
return(er)
}
exercise_6_A(1012)
```

结果如下：

```
           50    100    200    500   1000
chi(1) 0.0783 0.0657 0.0584 0.0496 0.0535
U(0,2) 0.0492 0.0495 0.0460 0.0499 0.0493
exp(1) 0.0655 0.0644 0.0515 0.0492 0.0518
```

设计的$\alpha=0.05$，实验可见当样本量较大时,t检验具有较好的稳健性

6-B

从总体$X\sim N(2,10)$取样本$X_1,X_2\cdots X_n$，令$Y_n=3*X_n+\sigma,\quad \sigma\sim N(5,50)$，则$(X,Y)$为有依赖关系的二元联合正态分布，使用三种方法比较检验显著性。

具体代码如下：

```
#### 6_B ####
exercise_6_B <- function(seed){
seed <- set.seed(123)
```

```
x <- rnorm(20,2,10)
sigma <- rnorm(20,5,50)
y <- 3*x+sigma
cor(x,y)
pearson <- cor.test(x,y)
kendall <- cor.test(x,y,method = 'kendall')
spearman <- cor.test(x,y,method = 'spearman')
data.frame(x,y)
return(list(pearson=pearson,kendall=kendall,spearman=spearman))
}
exercise_6_B()
```

输出结果如下：

```
$pearson

  Pearson's product-moment correlation

data:  x and y
t = 2.6052, df = 18, p-value = 0.0179
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1050841 0.7842035
sample estimates:
      cor
0.5232718


$kendall

  Kendall's rank correlation tau

data:  x and y
T = 136, p-value = 0.007346
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.4315789


$spearman

  Spearman's rank correlation rho

data:  x and y
S = 532, p-value = 0.00608
```

```
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.6
```

可见后两种方法具有更好的检验功效(p-value更小)。