STAT 455 Homework 04 - R Code

Martha Eichlersmith
2019-11-07

Problem 3.3

```r
func_Xsqcells <- function(matrix, i, j){
Y.ij <- matrix[i, j]
Y.iplus <- sum(matrix[i,])
Y.plusj <- sum(matrix[,j])
n <- sum(matrix)
m.ij <- (Y.iplus*Y.plusj)/n
(Y.ij - m.ij)^2 / m.ij}

func_chisqtest <- function(matrix){
I <- dim(matrix)[1]
J <- dim(matrix)[2]
row <- c()
for (i in 1:I){
  vec <- rep(i, J)
  row <- c(row, vec)}
col <- c(rep(c(1:J), I))
Xsq <- sum(mapply(func_Xsqcells, matrix=rep(list(matrix), length(row)), i=row, j=col))
df <- (I - 1)*(J-1)
pval <- 1- pchisq(Xsq, df)
paste("X-squared=", round(Xsq, 5), ", df=", df, ", p-value=", decimal(pval, 4))}

func_Lsqcells <- function(matrix, i, j){
Y.ij <- matrix[i, j]
Y.iplus <- sum(matrix[i,])
Y.plusj <- sum(matrix[,j])
n <- sum(matrix)
m.ij <- (Y.iplus*Y.plusj)/n
2*Y.ij*log(Y.ij/m.ij)}

func_liketest <- function(matrix){
I <- dim(matrix)[1]
J <- dim(matrix)[2]
row <- c()
for (i in 1:I){
  vec <- rep(i, J)
  row <- c(row, vec)}
col <- c(rep(c(1:J), I))
Lsq <- sum(mapply(func_Lsqcells, matrix=rep(list(matrix), length(row)), i=row, j=col))
df <- (I - 1)*(J-1)
pval <- 1- pchisq(Lsq, df)
paste("L-squared=", round(Lsq, 5), ", df=", df, ", p-value=", decimal(pval, 4))}
```

```r
shots <- matrix(c(251, 48, 34, 5), nrow=2, ncol=2)
func_chisqtest(shots)
```

```
## [1] "X-squared= 0.27274 , df= 1 , p-value= 0.6015"
```

```r
func_liketest(shots)
```

```
## [1] "L-squared= 0.28575 , df= 1 , p-value= 0.5930"
```

Problem 3.9a

```r
names.r <- c("Schizophrenia",
             "Affective disorder",
             "Neurosis",
             "Personality disorder",
             "Special Systems")
names.c <- c("Drugs", "No Drugs")
count.vec <- c(105, 12, 18, 47,  0,
             8,  2, 19, 52, 13)

r <-length(names.r)
c <- length(names.c)
row <- c()
for (i in 1:r){
  vec <- rep(i, c)
  row <- c(row, vec)
}
column <- c(rep(c(1:c), r))

Count<- matrix(count.vec, nrow=r, ncol=c)
rownames(Count) <- names.r
colnames(Count) <- names.c
stdreschi <- chisq.test(Count)$stdres
t1 <- kable(Count,    format="latex", booktabs=TRUE)
t2 <- kable(stdreschi, format="latex" , booktabs=TRUE)
```

Counts

	Drugs	No Drugs
Schizophrenia	105	8
Affective disorder	12	2
Neurosis	18	19
Personality disorder	47	52
Special Systems	0	13

Perason Standard Residuals

	Drugs	No Drugs
Schizophrenia	7.874526	-7.874526
Affective disorder	1.602262	-1.602262
Neurosis	-2.385315	2.385315
Personality disorder	-4.841701	4.841701
Special Systems	-5.139491	5.139491

Problem 3.12

```r
names.r <- c("Less than highschool",
             "Highschool",
             "More than highschool")
names.c <- c("Disapprove", "Middle", "Approve")
count.vec <- c(209, 151, 16,
```

```
               101, 126, 21,
               237, 426, 138)

r <-length(names.r)
c <- length(names.c)
row <- c()
for (i in 1:r){
  vec <- rep(i, c)
  row <- c(row, vec)
}
column <- c(rep(c(1:c), r))

Count<- matrix(count.vec, nrow=r, ncol=c)

gamma <- GKgamma(Count)$gamma
gammaCI <- GKgamma(Count)$CI
```

```
## [1] "Gamma: 0.3873"
```

```
## [1] "CI: ( 0.3156 , 0.4591 )"
```

Problem 3.15

```
count <- c(7, 8, 0, 15)
n <- sum(count)
Count <- matrix(count, nrow=2, ncol=2, byrow=TRUE)
OR <- (Count[1,1]*Count[2,2]) / (Count[1,2]*Count[2,1])

#WOOLF'S APPROXIMATE CI
se_log <- sqrt(  sum(1/count))
z.star <- qnorm(0.975, 0, 1)
CI.up_log <- log(OR) + z.star*se_log
CI.low_log <- log(OR) - z.star*se_log
CI.up <- exp(CI.up_log)
CI.low <- exp(CI.low_log)
Woolf_CI <- c(CI.low, CI.up)
```

```
## [1] "OR: Inf"
```

```
## [1] "Woolf CI Interval: ( NaN , Inf )"
```