STAT 455 Homework 04 - R Code

Martha Eichlersmith
2019-11-07

Problem 3.3

```
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))}
```

```
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"
```

```
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

```r
names.r <- c("Treatment", "Control")
names.c <- c("Normal.Yes", "Normal.No")
r <-length(names.r)
c <- length(names.c)
count.vec <- count <- c(7, 8, 0, 15)
Count<- matrix(count.vec, nrow=r, ncol=c, byrow=TRUE)
rownames(Count) <- names.r
colnames(Count) <- names.c
kable(Count, booktabs=TRUE)
```

	Normal.Yes	Normal.No
Treatment	7	8
Control	0	15

```r
n <- sum(count)
OR <- (Count[1,1]*Count[2,2]) / (Count[1,2]*Count[2,1])

#WOOLF (WALD)
A <- oddsratio(Count, conf.level=0.95, method="wald")$measure[2,2:3]

#CORNFIELD EXACT
B <- Cornfieldexact.CI(Count[1,1], sum(Count[1,]), Count[2,1], sum(Count[2,]))

#PROFILE LIKELIHOOD
y.data <- c(rep(1, Count[1,1]), rep(0, Count[1,2]), rep(1, Count[2,1]), rep(0, Count[2,2]))
x.data <- c(rep(1, Count[1,1]), rep(1, Count[1,2]), rep(0, Count[2,1]), rep(0, Count[2,2]))
df.data <- data.frame("Y"=y.data, "X"=x.data)
logit <- glm(Y ~ X, data=df.data, family=binomial)
C1 <- exp(confint(logit)[2,])

#PROFILE LIKELIHOOD adj
Count.adj <- Count+1
y.data.adj <- c(rep(1, Count.adj[1,1]), rep(0, Count.adj[1,2]), rep(1, Count.adj[2,1]), rep(0, Count.adj[2,2
x.data.adj <- c(rep(1, Count.adj[1,1]), rep(1, Count.adj[1,2]), rep(0, Count.adj[2,1]), rep(0, Count.adj[2,2
df.data.adj <- data.frame("Y"=y.data.adj, "X"=x.data.adj)
logit.adj <- glm(Y ~ X, data=df.data.adj, family=binomial)
C2 <- exp(confint(logit.adj)[2,])



names <- c("Woolf (Wald)", "Cornfield Exact", "Profile Likelihood", "Profe Likelihood, counts+1")
results<- cbind( names, decimal(rbind(A, B, C1, C2), dec))
rownames(results) <- c("A", "B", "C(1)", "C(2)")
colnames(results) <- c("", "CI Lower", "CI Uppder")

kable(results, align='lrrr', booktabs=TRUE)
```

		CI Lower	CI Uppder
A	Woolf (Wald)	NaN	Inf
B	Cornfield Exact	1.9784	Inf
C(1)	Profile Likelihood	0.0000	NA
C(2)	Profe Likelihood, counts+1	2.1242	286.7235

Problem 3.31

```
func_Xsqcells.iid <- function(matrix, i, j){
n.ij <- matrix[i, j]
n <- sum(matrix)
p.1plus <- sum(matrix[1,])/n
p.plus1 <- sum(matrix[,1])/n
theta <- (p.1plus + p.plus1) / 2
if (i==1 & j==1) {mu.hat <- n * theta^2}
if (i==1 & j==2) {mu.hat <- n * theta * (1 - theta)}
if (i==2 & j==1) {mu.hat <- n * theta * (1 - theta)}
if (i==2 & j==2) {mu.hat <- n * (1 - theta)^2 }
(n.ij - mu.hat)^2 / mu.hat }

func_chisqtest.iid <- 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 <- sum(dim(matrix)) - 2
pval <- 1- pchisq(Xsq, df)
paste("X-squared=", round(Xsq, 5), ", df=", df, ", p-value=", decimal(pval, 4))}
```

```
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"
```

```
func_chisqtest.iid(shots)
```

```
## [1] "X-squared= 0.27274 , df= 2 , p-value= 0.8725"
```