1. Done

2. 60

3.

Chi-squared test for given probabilities

data: mat

X-squared = 144.3667, df = 4, p-value < 2.2e-16

4.

Pearson's Chi-squared test

data: fuckit

X-squared = 32.3686, df = 12, p-value = 0.001214

p-value is very small so we accept H0

5.

School 1 2 3 4

Grade

1 9 8 7 5

2 12 23 11 15

3 22 36 27 55

4 8 6 5 11

5 19 7 10 4

We’re not completely sure how to get the data asked, but this is what we have.

6. ctmcp(mat, 3)

Rowi Rowj pi pj pi-pj

vec 1 2 0.3142857 0.4500000 -0.1357143

vec 1 3 0.3142857 0.4500000 -0.1357143

vec 1 4 0.3142857 0.6111111 -0.2968254

vec 2 3 0.4500000 0.4500000 0.0000000

vec 2 4 0.4500000 0.6111111 -0.1611111

vec 3 4 0.4500000 0.6111111 -0.1611111

7.

THIS DOES NOT WORK, BECAUSE WE DON’T KNOW HOW TO DO BONFERONI. OTHERWISE, EVERYTHING ELSE HAS BEEN IMPLEMENTED

ctmcp <- function(ctmat,colk,alpha =.05)

{  
  
## the CT is in the array (matrix) ctmat. It is  
## c-table for testing of homogenity. The range  
## of the distributions is across the columns  
## the rows correspond to the distributions (populations).  
  
## This functions only gets the difference in estimates  
## the colk range value. It can be extended to CI's

tab <- c()  
a <- length(ctmat[,1])

b <- length(ctmat[1,])

alpha = 0.5

numc <- a\*(a-1)/2

zc <- abs(qnorm(alpha/(2\*numc)))

for(i in 1:(a-1))

{  
 ni <- sum(ctmat[i,])  
 pi <- ctmat[i,colk]/ni  
 for(j in (i+1):a)

{  
 nj <- sum(ctmat[j,])  
 pj <- ctmat[j,colk]/nj

conf<-

#get the lower bound and the upper bound

if(lowerBound\*UpperBound > 0)

{

sig<-1

}

else

{

sig<-0

}  
 vec <- c(i,j,pi,pj,pi-pj,conf,sig)  
 tab <- rbind(tab,vec)  
 }  
}  
colnames(tab) <- c("Rowi","Rowj","pi","pj","pi-pj",”CI”,”SIG”)  
return(tab)  
}