STSCI4110 MIDTERM

Nick Gembs

11/9/2022

df = read.csv("C:/Users/Nick/Downloads/policing2.csv")

df

df = df[(df$race == "White") | (df$race == "Black"),]  
df = df[(df$sex == "Male") | (df$sex == "Female"),]  
df = df[(df$held == 0) | (df$held == 1),]  
df = df[(df$employed == "Yes") | (df$employed == "No"),]  
df = df[(df$citizen == "Yes") | (df$citizen == "No"),]  
df = df[(df$prior.traffic == 0) | (df$prior.traffic == 1)| (df$prior.traffic == 2),]  
df = df[(df$year > 2000) & (df$year < 2007),]  
df = df[(df$databases > -1) & (df$databases < 7),]  
df = df[(df$age > 3) & (df$age < 110),]  
df = df[(df$region == "North") | (df$region == "South")| (df$region == "East")| (df$region == "West"),]

heldbinary = df$held  
df = cbind(df, heldbinary)  
  
for(i in 1:length(df$held)){  
 if(df$held[i]==1){  
 df$held[i] = "harsh"  
 } else {  
 df$held[i] = "lenient"  
 }  
}

#Descriptive Stats Numeric  
#Numeric = prior.traffic, databases, age, year  
  
#min  
  
minprior.traffic = min(df$prior.traffic)  
minprior.traffic

## [1] 0

mindatabases = min(df$databases)  
mindatabases

## [1] 0

minyear = min(df$year)  
minyear

## [1] 2001

minage = min(df$age)  
minage

## [1] 13

#max  
  
maxprior.traffic = max(df$prior.traffic)  
maxprior.traffic

## [1] 2

maxdatabases = max(df$databases)  
maxdatabases

## [1] 6

maxyear = max(df$year)  
maxyear

## [1] 2006

maxage = max(df$age)  
maxage

## [1] 67

#median  
  
medianprior.traffic = median(df$prior.traffic)  
medianprior.traffic

## [1] 1

mediandatabases = median(df$databases)  
mediandatabases

## [1] 1

medianyear = median(df$year)  
medianyear

## [1] 2004

medianage = median(df$age)  
medianage

## [1] 22

#min  
  
sdprior.traffic = sd(df$prior.traffic)  
sdprior.traffic

## [1] 0.801067

sddatabases = sd(df$databases)  
sddatabases

## [1] 1.540232

sdyear = sd(df$year)  
sdyear

## [1] 1.389671

sdage = sd(df$age)  
sdage

## [1] 8.330865

min = c(minage,mindatabases,minprior.traffic,minyear)  
max = c(maxage,maxdatabases,maxprior.traffic,maxyear)  
median = c(medianage,mediandatabases,medianprior.traffic,medianyear)  
Standard.Dev = c(sdage,sddatabases,sdprior.traffic,sdyear)  
  
numerics = data.frame(min,max,median,Standard.Dev)  
row.names(numerics) = c("Age", "Databases", "Prior Traffic", "Year")  
numerics

## min max median Standard.Dev  
## Age 13 67 22 8.330865  
## Databases 0 6 1 1.540232  
## Prior Traffic 0 2 1 0.801067  
## Year 2001 2006 2004 1.389671

# Descriptive Statistics: Categorical  
# Categorical: Outcome(held), race, sex, prior.traffic, region, employed, citizen, databases  
  
# held  
harsh = length(df[df$heldbinary == 1,]$held)/length(df$held)  
harsh

## [1] 0.1702708

lenient = 1-harsh  
lenient

## [1] 0.8297292

held = data.frame(harsh,lenient, row.names = "held")  
held

## harsh lenient  
## held 0.1702708 0.8297292

# race  
  
black = length(df[df$race == "Black",]$race)/length(df$race)  
black

## [1] 0.2472238

white = 1-black  
white

## [1] 0.7527762

race = data.frame(black,white, row.names = "race")  
race

## black white  
## race 0.2472238 0.7527762

# sex  
  
Male = length(df[df$sex == "Male",]$sex)/length(df$sex)  
Male

## [1] 0.914475

Female = 1-Male  
Female

## [1] 0.08552503

sex = data.frame(Male,Female, row.names = "sex")  
sex

## Male Female  
## sex 0.914475 0.08552503

# Citizen  
  
Citizen = length(df[df$citizen == "Yes",]$citizen)/length(df$citizen)  
Citizen

## [1] 0.8540814

NonCitizen = 1-Citizen  
NonCitizen

## [1] 0.1459186

Citizenship = data.frame(Citizen,NonCitizen, row.names = "Citizenship")  
Citizenship

## Citizen NonCitizen  
## Citizenship 0.8540814 0.1459186

# Employed  
  
Employed = length(df[df$employed == "Yes",]$employed)/length(df$employed)  
Employed

## [1] 0.7858952

Unemployed = 1-Employed  
Unemployed

## [1] 0.2141048

Employment = data.frame(Employed,Unemployed, row.names = "Employment")  
Employment

## Employed Unemployed  
## Employment 0.7858952 0.2141048

# Region  
  
North = length(df[df$region == "North",]$region)/length(df$region)  
North

## [1] 0.2955387

South = length(df[df$region == "South",]$region)/length(df$region)  
South

## [1] 0.2018313

East = length(df[df$region == "East",]$region)/length(df$region)  
East

## [1] 0.205338

West = length(df[df$region == "West",]$region)/length(df$region)  
West

## [1] 0.297292

Region = data.frame(North,South,East,West, row.names = "Region")  
  
# Prior.traffic  
  
pt0 = length(df[df$prior.traffic == 0,]$prior.traffic)/length(df$prior.traffic)  
pt0

## [1] 0.3970388

pt1 = length(df[df$prior.traffic == 1,]$prior.traffic)/length(df$prior.traffic)  
pt1

## [1] 0.3403468

pt2 = 1-(pt1+pt0)  
pt2

## [1] 0.2626145

Prior.Traffic = data.frame(pt0,pt1,pt2 )  
colnames(Prior.Traffic) = (c("No Prior Violations", "1 Prior Violation", "2 or more Prior Violations"))  
Prior.Traffic

## No Prior Violations 1 Prior Violation 2 or more Prior Violations  
## 1 0.3970388 0.3403468 0.2626145

# Databases  
  
db0 = length(df[df$databases == 0,]$databases)/length(df$databases)  
db0

## [1] 0.3526203

db1 = length(df[df$databases == 1,]$databases)/length(df$databases)  
db1

## [1] 0.1622833

db2 = length(df[df$databases == 2,]$databases)/length(df$databases)  
db2

## [1] 0.1511786

db3 = length(df[df$databases == 3,]$databases)/length(df$databases)  
db3

## [1] 0.183908

db4 = length(df[df$databases == 4,]$databases)/length(df$databases)  
db4

## [1] 0.124099

db5 = length(df[df$databases == 5,]$databases)/length(df$databases)  
db5

## [1] 0.02415741

db6 = length(df[df$databases == 6,]$databases)/length(df$databases)  
db6

## [1] 0.001753361

Databases = data.frame(db0,db1,db2,db3,db4,db5,db6, row.names = "Databases")  
colnames(Databases) = c("0","1","2","3","4","5","6")  
Databases

## 0 1 2 3 4 5  
## Databases 0.3526203 0.1622833 0.1511786 0.183908 0.124099 0.02415741  
## 6  
## Databases 0.001753361

mat = table(df$held, df$sex)  
  
mat

##   
## Female Male  
## harsh 61 813  
## lenient 378 3881

bl = mat[1,1] + mat[2,1]  
br = mat[1,2] + mat[2,2]  
rt = mat[1,1] + mat[1,2]  
rb = mat[2,1] + mat[2,2]  
total = mat[2,1] + mat[2,2]+mat[1,1] + mat[1,2]  
  
expected1 = (bl\*rt)/total  
expected1

## [1] 74.74888

expected2 = (bl\*rb)/total  
expected2

## [1] 364.2511

expected3 = (rt\*br)/total  
expected3

## [1] 799.2511

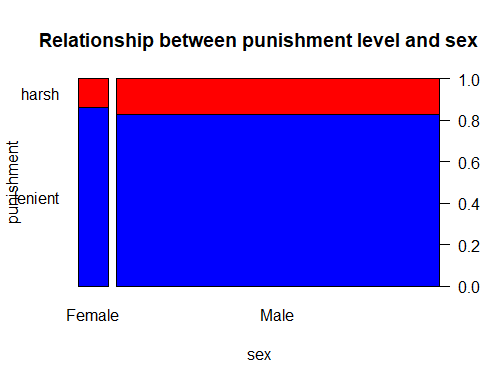
expected4 = (rb\*br)/total  
expected4

## [1] 3894.749

cat("All expected counts greater than 5")

## All expected counts greater than 5

par(las=1)  
spineplot(t(mat), col = c("blue","red"), main = ("Relationship between punishment level and sex"), xlab = "sex", ylab = "punishment")



chisq.test(mat, correct = FALSE)

##   
## Pearson's Chi-squared test  
##   
## data: mat  
## X-squared = 3.3329, df = 1, p-value = 0.06791

mat = table(df$held, df$race)  
  
mat

##   
## Black White  
## harsh 325 549  
## lenient 944 3315

bl = mat[1,1] + mat[2,1]  
br = mat[1,2] + mat[2,2]  
rt = mat[1,1] + mat[1,2]  
rb = mat[2,1] + mat[2,2]  
total = mat[2,1] + mat[2,2]+mat[1,1] + mat[1,2]  
  
expected1 = (bl\*rt)/total  
expected1

## [1] 216.0736

expected2 = (bl\*rb)/total  
expected2

## [1] 1052.926

expected3 = (rt\*br)/total  
expected3

## [1] 657.9264

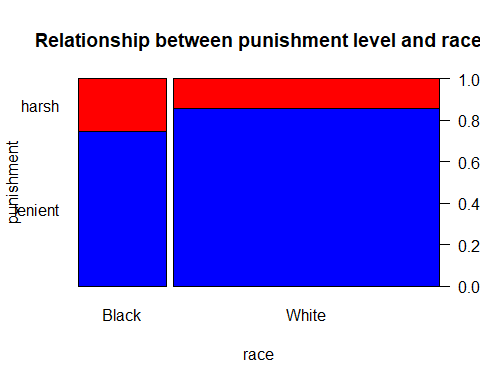
expected4 = (rb\*br)/total  
expected4

## [1] 3206.074

cat("All expected counts greater than 5")

## All expected counts greater than 5

par(las=1)  
spineplot(t(mat), col = c("blue","red"), main = ("Relationship between punishment level and race"), xlab = "race", ylab = "punishment")



chisq.test(mat, correct = FALSE)

##   
## Pearson's Chi-squared test  
##   
## data: mat  
## X-squared = 87.915, df = 1, p-value < 2.2e-16

mat = table(df$held, df$employed)  
  
mat

##   
## No Yes  
## harsh 343 531  
## lenient 756 3503

bl = mat[1,1] + mat[2,1]  
br = mat[1,2] + mat[2,2]  
rt = mat[1,1] + mat[1,2]  
rb = mat[2,1] + mat[2,2]  
total = mat[2,1] + mat[2,2]+mat[1,1] + mat[1,2]  
  
expected1 = (bl\*rt)/total  
expected1

## [1] 187.1276

expected2 = (bl\*rb)/total  
expected2

## [1] 911.8724

expected3 = (rt\*br)/total  
expected3

## [1] 686.8724

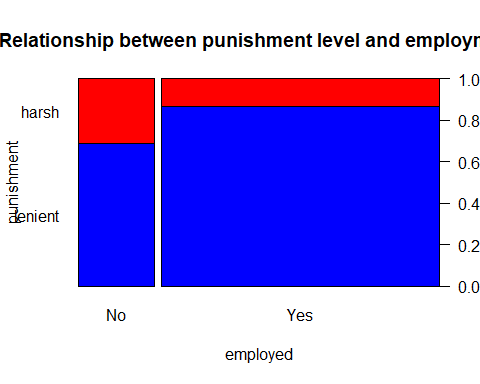
expected4 = (rb\*br)/total  
expected4

## [1] 3347.128

cat("All expected counts greater than 5")

## All expected counts greater than 5

par(las=1)  
spineplot(t(mat), col = c("blue","red"), main = ("Relationship between punishment level and employment"), xlab = "employed", ylab = "punishment")



chisq.test(mat, correct = FALSE)

##   
## Pearson's Chi-squared test  
##   
## data: mat  
## X-squared = 199.11, df = 1, p-value < 2.2e-16

mat = table(df$held, df$citizen)  
  
mat

##   
## No Yes  
## harsh 203 671  
## lenient 546 3713

bl = mat[1,1] + mat[2,1]  
br = mat[1,2] + mat[2,2]  
rt = mat[1,1] + mat[1,2]  
rb = mat[2,1] + mat[2,2]  
total = mat[2,1] + mat[2,2]+mat[1,1] + mat[1,2]  
  
expected1 = (bl\*rt)/total  
expected1

## [1] 127.5328

expected2 = (bl\*rb)/total  
expected2

## [1] 621.4672

expected3 = (rt\*br)/total  
expected3

## [1] 746.4672

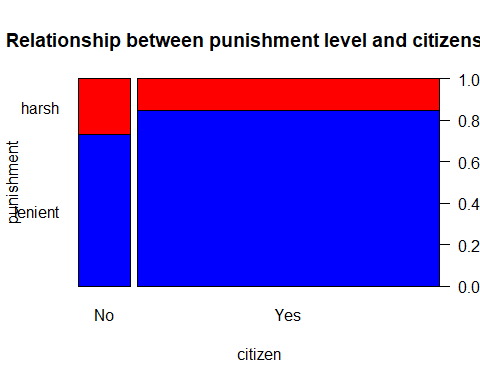
expected4 = (rb\*br)/total  
expected4

## [1] 3637.533

cat("All expected counts greater than 5")

## All expected counts greater than 5

par(las=1)  
spineplot(t(mat), col = c("blue","red"), main = ("Relationship between punishment level and citizenship"), xlab = "citizen", ylab = "punishment")



chisq.test(mat, correct = FALSE)

##   
## Pearson's Chi-squared test  
##   
## data: mat  
## X-squared = 63.017, df = 1, p-value = 2.049e-15

mat = table(df$held, df$region)  
  
mat

##   
## East North South West  
## harsh 169 279 172 254  
## lenient 885 1238 864 1272

bl = mat[1,1] + mat[2,1]  
br = mat[1,2] + mat[2,2]  
rt = mat[1,1] + mat[1,2] + mat[1,3] + mat[1,4]  
rb = mat[2,1] + mat[2,2] + mat[2,3] + mat[2,4]  
bl1 = mat[1,3] + mat[2,3]  
br1 = mat[1,4] + mat[2,4]  
  
total = mat[2,1] + mat[2,2]+mat[1,1] + mat[1,2] + mat[1,3] + mat[2,3] +mat[1,4] + mat[2,4]  
  
expected1 = (bl\*rt)/total  
expected1

## [1] 179.4654

expected2 = (bl\*rb)/total  
expected2

## [1] 874.5346

expected3 = (br\*rt)/total  
expected3

## [1] 258.3008

expected4 = (br\*rb)/total  
expected4

## [1] 1258.699

expected5 = (bl1\*rt)/total  
expected5

## [1] 176.4005

expected6 = (bl1\*rb)/total  
expected6

## [1] 859.5995

expected7 = (br1\*rt)/total  
expected7

## [1] 259.8332

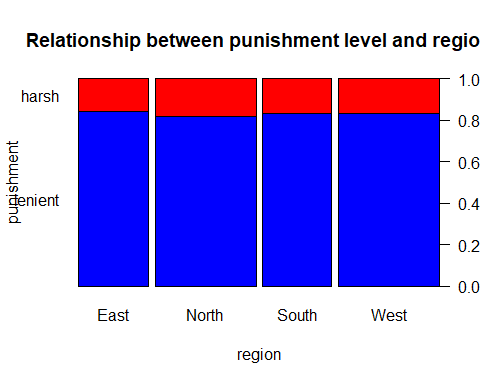
expected8 = (br1\*rb)/total  
expected8

## [1] 1266.167

cat("All expected counts greater than 5")

## All expected counts greater than 5

par(las=1)  
spineplot(t(mat), col = c("blue","red"), main = ("Relationship between punishment level and region"), xlab = "region", ylab = "punishment")



chisq.test(mat, correct = FALSE)

##   
## Pearson's Chi-squared test  
##   
## data: mat  
## X-squared = 3.0248, df = 3, p-value = 0.3878

mat = table(df$held, df$prior.traffic)  
  
mat

##   
## 0 1 2  
## harsh 347 299 228  
## lenient 1691 1448 1120

bl = mat[1,1] + mat[2,1]  
br = mat[1,2] + mat[2,2]  
rt = mat[1,1] + mat[1,2] + mat[1,3]   
rb = mat[2,1] + mat[2,2] + mat[2,3]   
bl1 = mat[1,3] + mat[2,3]  
  
total = mat[2,1] + mat[2,2]+mat[1,1] + mat[1,2]+mat[1,3] + mat[2,3]  
  
expected1 = (bl\*rt)/total  
expected1

## [1] 347.0119

expected2 = (bl\*rb)/total  
expected2

## [1] 1690.988

expected3 = (br\*rt)/total  
expected3

## [1] 297.4631

expected4 = (br\*rb)/total  
expected4

## [1] 1449.537

expected5 = (bl1\*rt)/total  
expected5

## [1] 229.525

expected6 = (bl1\*rb)/total  
expected6

## [1] 1118.475

cat("All expected counts greater than 5")

## All expected counts greater than 5

par(las=1)  
spineplot(t(mat), col = c("blue","red"), main = ("Relationship between punishment level and prior traffic violations"), xlab = "prior traffic", ylab = "punishment")  
  
  
library("vcdExtra")

## Warning: package 'vcdExtra' was built under R version 4.2.2

## Loading required package: vcd

## Warning: package 'vcd' was built under R version 4.2.2

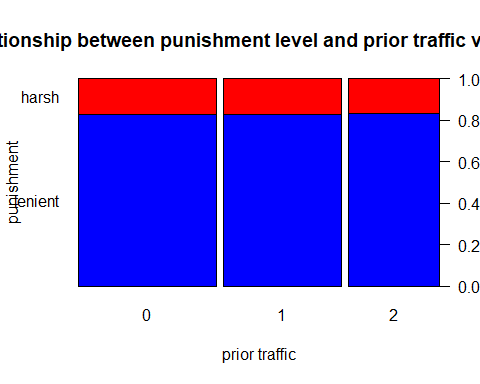
## Loading required package: grid

##   
## Attaching package: 'vcd'

## The following object is masked \_by\_ '.GlobalEnv':  
##   
## Employment

## Loading required package: gnm

## Warning: package 'gnm' was built under R version 4.2.2



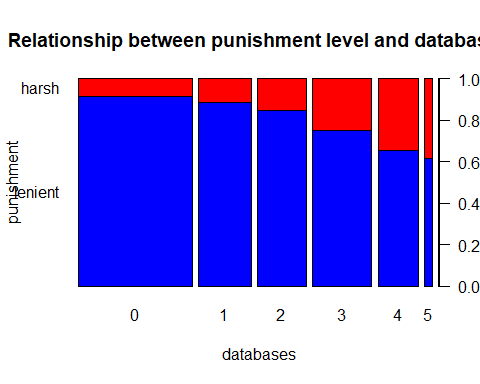
CMHtest(mat)

## Cochran-Mantel-Haenszel Statistics for by   
##   
## AltHypothesis Chisq Df Prob  
## cor Nonzero correlation 0.0049202 1 0.94408  
## rmeans Row mean scores differ 0.0049202 1 0.94408  
## cmeans Col mean scores differ 0.0217789 2 0.98917  
## general General association 0.0217789 2 0.98917

mat = table(df$held, df$databases)  
  
mat

##   
## 0 1 2 3 4 5 6  
## harsh 153 95 119 237 219 48 3  
## lenient 1657 738 657 707 418 76 6

par(las=1)  
spineplot(t(mat), col = c("blue","red"), main = ("Relationship between punishment level and databases"), xlab = "databases", ylab = "punishment")



CMHtest(mat)

## Cochran-Mantel-Haenszel Statistics for by   
##   
## AltHypothesis Chisq Df Prob  
## cor Nonzero correlation 319.24 1 2.1216e-71  
## rmeans Row mean scores differ 319.24 1 2.1216e-71  
## cmeans Col mean scores differ 336.67 6 1.1221e-69  
## general General association 336.67 6 1.1221e-69

mat = table(df$held, df$year)  
  
mat

##   
## 2001 2002 2003 2004 2005 2006  
## harsh 111 152 182 173 202 54  
## lenient 369 714 901 1071 984 220

CMHtest(mat)

## Cochran-Mantel-Haenszel Statistics for by   
##   
## AltHypothesis Chisq Df Prob  
## cor Nonzero correlation 4.5411 1 0.03309048  
## rmeans Row mean scores differ 4.5411 1 0.03309048  
## cmeans Col mean scores differ 22.8030 5 0.00036813  
## general General association 22.8030 5 0.00036813

lm.fit = glm(df$heldbinary~df$year, family = "binomial")  
summary(lm.fit)

##   
## Call:  
## glm(formula = df$heldbinary ~ df$year, family = "binomial")  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.6511 -0.6185 -0.6027 -0.5872 1.9446   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 112.12896 53.38327 2.10 0.0357 \*  
## df$year -0.05676 0.02665 -2.13 0.0332 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 4684.5 on 5132 degrees of freedom  
## Residual deviance: 4680.0 on 5131 degrees of freedom  
## AIC: 4684  
##   
## Number of Fisher Scoring iterations: 4

mat = table(df$held, df$age)  
  
mat

##   
## 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29  
## harsh 1 5 7 37 57 63 67 81 55 61 40 38 42 20 26 27 24  
## lenient 3 13 73 162 246 376 400 379 337 316 243 196 174 131 111 89 84  
##   
## 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46  
## harsh 16 11 5 14 21 14 15 9 17 8 13 14 14 10 6 3 10  
## lenient 70 84 64 57 67 68 48 70 49 42 46 38 31 25 22 26 23  
##   
## 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63  
## harsh 5 4 6 1 1 1 0 1 2 0 0 0 0 0 1 0 0  
## lenient 10 12 12 11 5 9 6 7 6 3 1 1 1 4 1 2 1  
##   
## 65 67  
## harsh 0 1  
## lenient 3 1

CMHtest(mat)

## Cochran-Mantel-Haenszel Statistics for by   
##   
## AltHypothesis Chisq Df Prob  
## cor Nonzero correlation 10.269 1 0.0013529  
## rmeans Row mean scores differ 10.269 1 0.0013529  
## cmeans Col mean scores differ 76.978 52 0.0137917  
## general General association 76.978 52 0.0137917

lm.fit = glm(df$heldbinary~df$age, family = "binomial")  
summary(lm.fit)

##   
## Call:  
## glm(formula = df$heldbinary ~ df$age, family = "binomial")  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.7862 -0.6141 -0.5954 -0.5807 1.9542   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.925500 0.114401 -16.831 < 2e-16 \*\*\*  
## df$age 0.013579 0.004245 3.199 0.00138 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 4684.5 on 5132 degrees of freedom  
## Residual deviance: 4674.6 on 5131 degrees of freedom  
## AIC: 4678.6  
##   
## Number of Fisher Scoring iterations: 4