STATS 506 HW 3

Calder Moore

STATS 506 HW 3

Problem 1 -

a)

```
library(haven)

audio <- read_xpt("AUX_I.xpt")
demo <- read_xpt("DEMO_I.xpt")

audio_demo <- merge(audio, demo, by = "SEQN")
dim(audio_demo)
```

```
[1] 4582  119
```

b)

```
audio_demo$RIAGENDR[audio_demo$RIAGENDR == 1] <- 0 #male
audio_demo$RIAGENDR[audio_demo$RIAGENDR == 2] <- 1 #female
audio_demo$RIAGENDR <- factor(audio_demo$RIAGENDR)

audio_demo$DMDCITZN[audio_demo$DMDCITZN == 1] <- 0 #citizen by birth or naturalization
audio_demo$DMDCITZN[audio_demo$DMDCITZN == 2] <- 1 #non-citizen
audio_demo$DMDCITZN[audio_demo$DMDCITZN == 7] <- NA #refused: 9 observations
audio_demo$DMDCITZN[audio_demo$DMDCITZN == 9] <- NA #don't know: 7 observations
audio_demo$DMDCITZN <- factor(audio_demo$DMDCITZN)
```

```
#household size for kids under 5 seems to be okay

#drop groups 12 and 13 since they are redundant and don't have many observations
#(~450 out of ~10,000)
audio_demo <- subset(audio_demo, !(INDHHIN2 %in% c(12, 13)))
audio_demo$INDHHIN2[audio_demo$INDHHIN2 == 77] <- NA #refused: 220 obs
audio_demo$INDHHIN2[audio_demo$INDHHIN2 == 99] <- NA #don't know: 134 obs
```

c)

```
library(knitr)

R1 <- glm(AUXTWIDR ~ RIAGENDR, family = "poisson", data = audio_demo)
R2 <- glm(AUXTWIDR ~ RIAGENDR + DMDCITZN + DMDHHSZA + INDHHIN2, family = "poisson", data = au
L1 <- glm(AUXTWIDL ~ RIAGENDR, family = "poisson", data = audio_demo)
L2 <- glm(AUXTWIDL ~ RIAGENDR + DMDCITZN + DMDHHSZA + INDHHIN2, family = "poisson", data = au

#exponentiated coefficient is another way to calculate incidence ratio
GenderIRR <- exp(c(R1$coefficients[2], R2$coefficients[2], L1$coefficients[2], L2$coefficien
CitznIRR <- exp(c(NA, R2$coefficients[3], NA, L2$coefficients[3]))
HHSZIRR <- exp(c(NA, R2$coefficients[4], NA, L2$coefficients[4]))
HHINIRR <- exp(c(NA, R2$coefficients[5], NA, L2$coefficients[5]))

#use formula for pseudo-R^2
pseudoR2 <- c(1-R1$deviance/R1$null.deviance,
              1-R2$deviance/R2$null.deviance,
              1-L1$deviance/L1$null.deviance,
              1-L2$deviance/L2$null.deviance)

#sample size
R1n <- R1$df.residual + R1$rank
R2n <- R2$df.residual + R2$rank
L1n <- L1$df.residual + L1$rank
L2n <- L2$df.residual + L2$rank

sampsize <- c(R1n, R2n, L1n, L2n)

AIC <- c(R1$aic, R2$aic, L1$aic, L2$aic)

tymptable <- data.frame(Model = c("R1", "R2", "L1", "L2"),
                        `Gender IRR` = GenderIRR,
                        `Citizenship IRR` = CitznIRR,
                        `HHSize IRR` = HHSZIRR,
                        `HHIncome IRR` = HHSZIRR,
                        `Pseudo R^2` = pseudoR2,
                        `Sample Size` = sampsize,
                        AIC = AIC)

kable(tymptable, digits = 4)
```

Model	Gender.IRR	Citizenship.IRR	HHSize.IRR	HHIncome.IRR	Pseudo.R.2	Sample.Size	AIC
R1	1.0104	NA	NA	NA	0.0001	3967	91088.12
R2	1.0147	1.0455	0.9945	0.9945	0.0068	3705	84850.46
L1	1.0169	NA	NA	NA	0.0003	3920	93403.31
L2	1.0188	1.0218	0.9831	0.9831	0.0036	3665	86602.03

d)

Based on the IRR from the gender variable for model L2, we can say that women have a higher incidence than men of about 1.8%.

```
summary(L2)
```

```
Call:
glm(formula = AUXTWIDL ~ RIAGENDR + DMDCITZN + DMDHHSZA + INDHHIN2,
    family = "poisson", data = audio_demo)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.4767710  0.0047890 934.810  < 2e-16 ***
RIAGENDR1    0.0186387  0.0035994   5.178 2.24e-07 ***
DMDCITZN1    0.0215476  0.0046464   4.637 3.53e-06 ***
DMDHHSZA    -0.0170192  0.0027274  -6.240 4.37e-10 ***
INDHHIN2    -0.0047513  0.0004118 -11.538  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 64149  on 3664  degrees of freedom
Residual deviance: 63920  on 3660  degrees of freedom
  (719 observations deleted due to missingness)
AIC: 86602

Number of Fisher Scoring iterations: 4
```

The coefficient on the gender variable in the L2 model is significant at the 99.9% level, suggesting that there is a difference between men and women.

Problem 2 - Sakila

a)

```
library(DBI)

sakila <- dbConnect(RSQLite::SQLite(), "sakila_master.db")
Rsakilacustomer <- dbGetQuery(sakila, "SELECT * FROM customer")

#SQL
#multiply the SUM(active) by 100.0 to make a percent and to get around integer division
SQLcustomer <- dbGetQuery(sakila, "SELECT store_id, COUNT(customer_id) AS customercount,
                                      100.0 * SUM(active)/COUNT(customer_id) as percentactive
                    FROM customer
                    GROUP BY store_id")

SQLcustomer
```

```
  store_id customercount percentactive
1        1           326      97.54601
2        2           273      97.43590
```

```
#R
store1 <- subset(Rsakilacustomer, store_id == 1)
store2 <- subset(Rsakilacustomer, store_id == 2)

customers <- matrix(c(1, 2, nrow(store1), nrow(store2),
                      100*sum(as.numeric(store1$active))/nrow(store1),
                      100*sum(as.numeric(store2$active))/nrow(store2)), nrow = 2)
colnames(customers) <- c("Store", "Customer Count", "Percent Active")

customers
```

```
     Store Customer Count Percent Active
[1,]     1            326       97.54601
[2,]     2            273       97.43590
```

```
library(microbenchmark)

microbenchmark(SQLcustomer)
```

```
Warning in microbenchmark(SQLcustomer): Could not measure a positive execution
time for 35 evaluations.
```

```
Unit: nanoseconds
        expr min lq mean median uq  max neval
 SQLcustomer   0  0   22      0  0 2200   100
```

```
microbenchmark(customers)
```

```
Warning in microbenchmark(customers): Could not measure a positive execution
time for 10 evaluations.
```

```
Unit: nanoseconds
      expr min lq mean median uq  max neval
 customers   0  0   20      0  0 2000   100
```

They are both quite fast in this case.

b)

```
countries <- dbGetQuery(sakila,
                        "SELECT CONCAT(s.first_name, ' ', s.last_name) AS name, co.country
                         FROM staff AS s
                            INNER JOIN address AS a ON a.address_id = s.address_id
                            INNER JOIN city AS c ON a.city_id = c.city_id
                            INNER JOIN country AS co ON co.country_id = c.country_id")

countries
```

```
          name   country
1 Mike Hillyer    Canada
2 Jon Stephens Australia
```

```
sakstaff <- dbGetQuery(sakila, "SELECT * FROM staff")
sakadd <- dbGetQuery(sakila, "SELECT * FROM address")
sakcity <- dbGetQuery(sakila, "SELECT * FROM city")
sakcountry <- dbGetQuery(sakila, "SELECT * FROM country")

Rsak <- merge(sakcountry, merge(sakcity, merge(sakadd, sakstaff, by = "address_id"),
                                by = "city_id"), by = "country_id")
```

```
Warning in merge.data.frame(sakcountry, merge(sakcity, merge(sakadd, sakstaff,
: column names 'last_update.x', 'last_update.y' are duplicated in the result
```

```
staffmat <- matrix(c(paste(Rsak$first_name, Rsak$last_name), Rsak$country), nrow = 2)
colnames(staffmat) <- c("Name", "Country")

staffmat
```

```
     Name           Country
[1,] "Jon Stephens" "Australia"
[2,] "Mike Hillyer" "Canada"
```

```
microbenchmark(countries)
```

```
Warning in microbenchmark(countries): Could not measure a positive execution
time for 7 evaluations.

Unit: nanoseconds
      expr min lq mean median uq  max neval
 countries   0  0   18      0  0 1800   100
```

```
microbenchmark(Rsak)
```

```
Warning in microbenchmark(Rsak): Could not measure a positive execution time
for 8 evaluations.
```

```
Unit: nanoseconds
 expr min lq mean median uq  max neval
 Rsak   0  0   19      0  0 1900   100
```

Again, similarly speedy.

c)

```
films <- dbGetQuery(sakila, "SELECT f.title, p.amount
                             FROM film AS f
                                INNER JOIN inventory AS i ON f.film_id = i.film_id
                                INNER JOIN rental AS r ON i.inventory_id = r.inventory_id
                                INNER JOIN payment AS p ON r.rental_id = p.rental_id
                                WHERE p.amount = (
                                  SELECT MAX(amount)
                                  FROM payment
                                ) ORDER BY f.title")
films
```

```
                   title amount
1  FLINTSTONES HAPPINESS  11.99
2    MIDSUMMER GROUNDHOG  11.99
3            MINE TITANS  11.99
4        SCORPION APOLLO  11.99
5        SCORPION APOLLO  11.99
6              SHOW LORD  11.99
7         STING PERSONAL  11.99
8            TIES HUNGER  11.99
9              TRAP GUYS  11.99
10      VIRTUAL SPOILERS  11.99
```

```r
sakfilm <- dbGetQuery(sakila, "SELECT * FROM film")
sakin <- dbGetQuery(sakila, "SELECT * FROM inventory")
sakrent <- dbGetQuery(sakila, "SELECT * FROM rental")
sakpay <- dbGetQuery(sakila, "SELECT * FROM payment")

Rsakpay <- merge(sakpay, merge(sakrent, merge(sakin, sakfilm, by = "film_id"),
                              by = "inventory_id"), by = "rental_id")
```

```
Warning in merge.data.frame(sakpay, merge(sakrent, merge(sakin, sakfilm, :
column names 'last_update.x', 'last_update.y' are duplicated in the result
```

```r
filmmat <- matrix(c(Rsakpay$title[Rsakpay$amount == max(Rsakpay$amount)],
                    Rsakpay$amount[Rsakpay$amount == max(Rsakpay$amount)]), ncol = 2)
colnames(filmmat) <- c("Film", "Value")

filmmat
```

```
      Film                    Value  
 [1,] "SHOW LORD"             "11.99"
 [2,] "VIRTUAL SPOILERS"      "11.99"
 [3,] "MIDSUMMER GROUNDHOG"   "11.99"
 [4,] "MINE TITANS"           "11.99"
 [5,] "SCORPION APOLLO"       "11.99"
 [6,] "TIES HUNGER"           "11.99"
 [7,] "STING PERSONAL"        "11.99"
 [8,] "FLINTSTONES HAPPINESS" "11.99"
 [9,] "TRAP GUYS"             "11.99"
[10,] "SCORPION APOLLO"       "11.99"
```

```r
microbenchmark(films)
```

```
Warning in microbenchmark(films): Could not measure a positive execution time
for 41 evaluations.
```

```
Unit: nanoseconds
  expr min lq mean median uq  max neval
 films   0  0   23      0  0 2200   100
```

```
microbenchmark(Rsakpay)
```

```
Warning in microbenchmark(Rsakpay): Could not measure a positive execution time
for 25 evaluations.
```

```
Unit: nanoseconds
    expr min lq mean median uq  max neval
 Rsakpay   0  0   33      0  0 3100   100
```

Still similar run times.

Problem 3 - Australian Records

a)

```
aus <- read.csv("au-500.csv")

#use grepl since it returns T/F
100*length(aus$email[grepl(".com", aus$email) & !grepl(".au$", aus$email)])/nrow(aus)
```

```
[1] 60
```

b)

```
#replace everything up to @ with blanks so we're left with the domain name
domains <- sub(".*@", "", aus$email)
sort(table(domains), decreasing = TRUE)
```

```
domains
        hotmail.com           gmail.com           yahoo.com         agar.net.au 
                114                 102                  84                   1 
       agney.net.au      ahlborn.com.au     albrough.com.au       alerte.com.au 
                  1                   1                   1                   1 
      amedro.net.au      andrion.com.au andrzejewski.com.au      angeron.net.au 
                  1                   1                   1                   1 
   arellanes.net.au      badgero.com.au        baird.net.au        bakey.com.au 
```

```
                  1                   1                   1                   1
      barras.com.au        biasi.net.au        biler.net.au       binnie.net.au
                  1                   1                   1                   1
     boudrie.net.au     brackett.net.au breckenstein.com.au       brueck.net.au
                  1                   1                   1                   1
    buchauer.net.au        bumby.com.au       burket.com.au   burnsworth.net.au
                  1                   1                   1                   1
     capelli.com.au    carabajal.com.au       catton.com.au      charney.net.au
                  1                   1                   1                   1
   chrusciel.net.au       chudej.net.au       connon.com.au     conquest.net.au
                  1                   1                   1                   1
    costeira.com.au      couzens.com.au        daleo.net.au      davoren.net.au
                  1                   1                   1                   1
    decelles.net.au      dejarme.net.au     delacruz.net.au       dellen.com.au
                  1                   1                   1                   1
     deritis.net.au   desjardiws.com.au        devol.net.au      diciano.com.au
                  1                   1                   1                   1
       didio.com.au   digregorio.net.au        druck.net.au      eilbeck.net.au
                  1                   1                   1                   1
         elm.net.au        entzi.net.au        fajen.net.au      farnham.com.au
                  1                   1                   1                   1
   fellhauer.com.au     fernades.com.au     figueras.net.au        filan.net.au
                  1                   1                   1                   1
      fraize.net.au      francis.net.au      freiman.net.au       fritch.com.au
                  1                   1                   1                   1
       fults.net.au     galagher.com.au       gedman.net.au         gene.com.au
                  1                   1                   1                   1
    gephardt.com.au        ghera.com.au         gish.net.au     glockner.com.au
                  1                   1                   1                   1
        gong.com.au     goodness.net.au       gordis.com.au       gudgel.com.au
                  1                   1                   1                   1
      helger.com.au      hermens.net.au      herrera.net.au hessenthaler.net.au
                  1                   1                   1                   1
     hinkson.net.au     hollimon.com.au        hoyne.com.au        hulme.com.au
                  1                   1                   1                   1
 huntsberger.net.au      hutchin.com.au         iida.net.au        jarva.com.au
                  1                   1                   1                   1
        jebb.net.au       kazeck.com.au       kazemi.net.au    kellebrew.com.au
                  1                   1                   1                   1
     kellman.net.au     kenfield.com.au       kinney.com.au        kloos.com.au
                  1                   1                   1                   1
      kloska.net.au      koerner.com.au        kopet.com.au        koury.net.au
                  1                   1                   1                   1
```

kueter.com.au	kunich.net.au	kushnir.net.au	ladeau.net.au
1	1	1	1
langanke.net.au	laprade.net.au	laroia.net.au	lary.net.au
1	1	1	1
leicht.com.au	leja.com.au	lek.net.au	levay.net.au
1	1	1	1
limberg.com.au	lofts.com.au	lolley.net.au	luening.com.au
1	1	1	1
lymaster.net.au	magnotta.net.au	mahmud.com.au	maker.net.au
1	1	1	1
malboeuf.com.au	mckale.net.au	menez.net.au	merkt.net.au
1	1	1	1
metevelis.net.au	mikel.net.au	mikovec.com.au	milbrandt.com.au
1	1	1	1
milsap.com.au	mishkin.com.au	moehring.net.au	mohrmann.net.au
1	1	1	1
mongolo.net.au	morguson.com.au	muhlbauer.net.au	nicley.com.au
1	1	1	1
novosel.net.au	nybo.net.au	oakland.com.au	ocken.net.au
1	1	1	1
okojie.com.au	orlinski.com.au	osmer.com.au	oto.com.au
1	1	1	1
overbough.com.au	paavola.com.au	pacleb.net.au	palaspas.net.au
1	1	1	1
pata.net.au	pawell.net.au	phay.com.au	ploszaj.net.au
1	1	1	1
polek.net.au	poncio.com.au	prez.com.au	prosienski.net.au
1	1	1	1
quintero.com.au	raddle.com.au	radel.net.au	rael.com.au
1	1	1	1
ramero.net.au	rathmann.com.au	rebich.net.au	remillard.net.au
1	1	1	1
roches.net.au	sanzenbacher.com.au	schimke.com.au	schmale.net.au
1	1	1	1
schoenleber.com.au	servantes.com.au	shiflett.com.au	silverstone.net.au
1	1	1	1
skursky.net.au	stavely.com.au	stitely.com.au	strawbridge.com.au
1	1	1	1
suffern.net.au	sumera.net.au	svoboda.net.au	taghon.net.au
1	1	1	1
taketa.net.au	telch.net.au	tepley.net.au	thro.net.au
1	1	1	1
tokich.net.au	tolbent.net.au	tovmasyan.net.au	vandermeer.com.au

```
                  1                   1                   1                   1
      vaughn.net.au    vollstedt.com.au       vrieze.net.au    vugteveen.net.au
                  1                   1                   1                   1
     waganer.net.au         ware.net.au         wasp.net.au   weissbrodt.com.au
                  1                   1                   1                   1
      weyman.com.au         whal.net.au    wildeboer.com.au   wisenbaker.net.au
                  1                   1                   1                   1
     wodicka.net.au     woodhams.com.au        yuasa.net.au
                  1                   1                   1
```

hotmail.com is the most common

c)

```
ampersands <- aus$company_name[grepl("[0123456789!@#$%^&*()<>?]", aus$company_name)]

noampersands <- aus$company_name[grepl("[0123456789!@#$%^*()<>?]", aus$company_name)]

100*length(ampersands)/nrow(aus)
```

```
[1] 8.8
```

```
100*length(noampersands)/nrow(aus)
```

```
[1] 0.6
```

8.8% with special characters including ampersands, 0.6% excluding them.

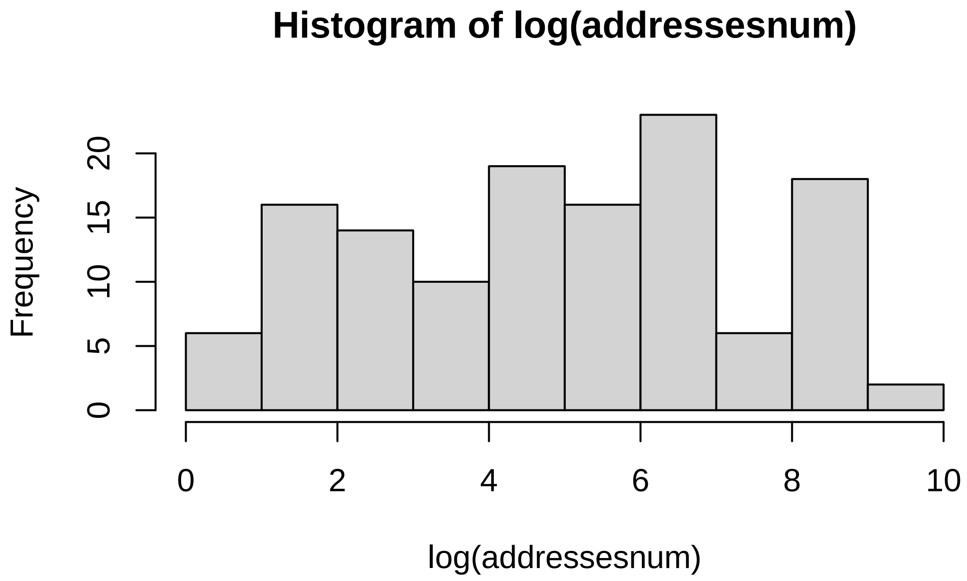
d)

```
#split along "-" to get the parts easier to paste and substr
phoneparts <- strsplit(aus$phone1, "-")

#basically a big paste function for each of the components of the number
newphones <- c()
for (i in 1:length(phoneparts)){
  newphones <- c(newphones, paste(phoneparts[[i]][1], substr(phoneparts[[i]][2], 1, 2), "-",
                                  substr(phoneparts[[i]][2], 3, 4), substr(phoneparts[[i]][3
                                  substr(phoneparts[[i]][3], 2, 4), sep = ""))
}

aus$phone3 <- newphones

#compare against values of phone1 and structure of phone 2. I put phone3 in the
#middle to compare

data.frame(aus$phone1[1:10], aus$phone3[1:10], aus$phone2[1:10])
```

```
   aus.phone1.1.10. aus.phone3.1.10. aus.phone2.1.10.
1      03-8174-9123     0381-749-123     0458-665-290
2      07-9997-3366     0799-973-366     0497-622-620
3      08-5558-9019     0855-589-019     0427-885-282
4      02-6044-4682     0260-444-682     0443-795-912
5      02-1455-6085     0214-556-085     0453-666-885
6      08-7868-1355     0878-681-355     0451-966-921
7      08-6522-8931     0865-228-931     0427-991-688
8      02-5226-9402     0252-269-402     0415-961-606
9      07-3184-9989     0731-849-989     0411-732-965
10     08-6890-4661     0868-904-661     0461-862-457
```

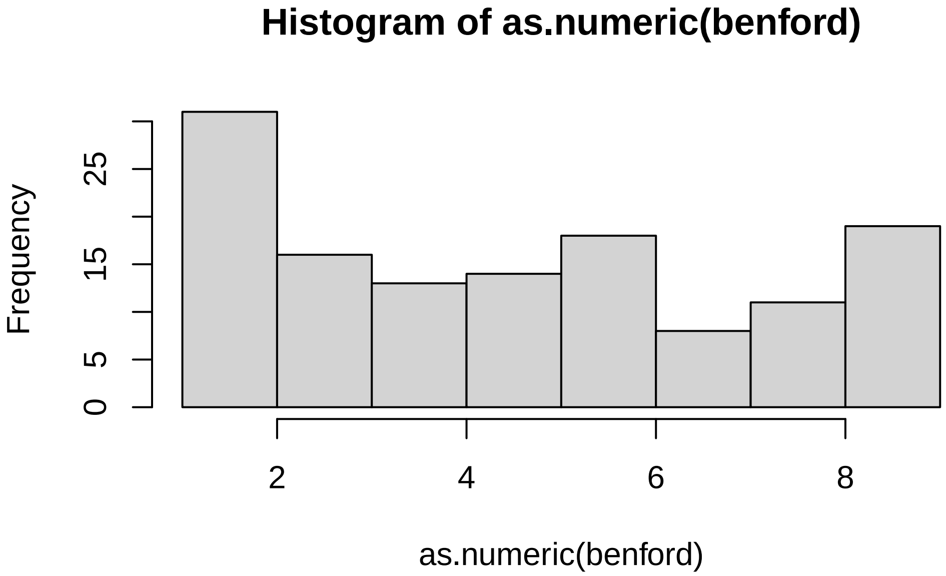
e)

```
#check for numbers at the end of the address, get rid of the rest of the address if they
#aren't there get rid of an address if no numbers at the end. Need *? so that not only
#the last number is included. Otherwise it cuts off all numbers but the last

addresses <- ifelse(grepl("[0-9]+$", aus$address),
                    sub(".*?([0-9]+)$", "\\1", aus$address),
                    "")


addressesnum <- as.numeric(addresses)

hist(log(addressesnum))
```



f)

```
benford <- substr(addresses, 1, 1)

hist(as.numeric(benford))
```



This could plausibly be real data since 1 occurs with the most frequency, although the pattern of larger digits being less frequent doesn't strictly hold. 9 for example is much more frequent than 6 or 7. But overall the pattern though is not so extreme as to be abnormal, so it could maybe pass depending on how strict one is with the criteria.