STAT 506 HW 3

ZHAO Shengchun

Github URL:

https://github.com/ZHAOShengchun67734538/STAT-506-HW-3

Question 1

(a)

```
library(knitr)
library(Hmisc)
```

```
Attaching package: 'Hmisc'

The following objects are masked from 'package:base':

    format.pval, units
```

```
vix.data = sasxport.get("C:/Users/z1883/Desktop/VIX_D.XPT")
```

```
Processing SAS dataset VIX_D     ..
```

```
nrow(vix.data)
```

```
[1] 6980
```

```r
demo.data = sasxport.get("C:/Users/z1883/Desktop/DEMO_D.XPT")
```

```
Processing SAS dataset DEMO_D    ..
```

```r
nrow(demo.data)
```

```
[1] 10348
```

```r
# Using the SEQN variable for merging.
# Keep only records which matched.
mix.data = merge(vix.data, demo.data,by = "seqn", all = FALSE)
nrow(mix.data)
```

```
[1] 6980
```

(b)

```r
# Check the NA data
sum(is.na(mix.data$viq220))
```

```
[1] 433
```

```r
sum(is.na(mix.data$ridageyr))
```

```
[1] 0
```

```r
# From the output result, we need to do data clean
```

```r
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:Hmisc':

    src, summarize
```

```
The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
new.data = filter(mix.data, !is.na(mix.data$viq220))
nrow(new.data)
```

```
[1] 6547
```

```r
# find the range of age
min(new.data$ridageyr)
```

```
[1] 12
```

```r
max(new.data$ridageyr)
```

```
[1] 85
```

```r
age = c("10-19","20-29","30-39","40-49","50-59","60-69","70-79","80-89")
proportion = c(1:8)*0
lowerbound = 10
upperbound = 19

for(i in 1:8)
{
  d = new.data[which(new.data$ridageyr>=lowerbound &
                       new.data$ridageyr<=upperbound),]
  # in the data document, the 1 means yes
  n1 = nrow(d[which(d$viq220 == 1),])
  n = nrow(d)
  proportion[i] = n1/n
  lowerbound = lowerbound + 10
  upperbound = upperbound + 10
}
df = data.frame(age, proportion = proportion*100)
library(knitr)
kable(df)
```

age	proportion
10-19	32.08812
20-29	32.58786
30-39	35.86667
40-49	36.99871
50-59	55.00821
60-69	62.22222
70-79	66.89038
80-89	66.88103

(c)

```
### (1) ###
# in respond, if it = 9, treat as missing value and delete
data = new.data[-which(new.data$viq220 == 9),]
```

```
d1 = cbind(data$viq220, data$ridageyr)
subdata1 = as.data.frame(d1)
colnames(subdata1) = c("viq220","age")
# Check whether there exist NA values
sum(is.na(subdata1$viq220))
```

```
[1] 0
```

```
sum(is.na(subdata1$age))
```

```
[1] 0
```

```
# Construct the model
subdata1$viq220 = as.factor(subdata1$viq220)
model1 = glm(viq220 ~ age, data = subdata1, family = binomial)
summary(model1)
```

```
Call:
glm(formula = viq220 ~ age, family = binomial, data = subdata1)

Coefficients:
```

```
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.260970   0.053448   23.59   <2e-16 ***
age         -0.024673   0.001206  -20.47   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8915.3  on 6544  degrees of freedom
Residual deviance: 8471.9  on 6543  degrees of freedom
AIC: 8475.9

Number of Fisher Scoring iterations: 4
```

```
### (2) ###
d2 = cbind(data$viq220, data$ridageyr, data$ridreth1,data$riagendr)
subdata2 = as.data.frame(d2)
colnames(subdata2) = c("viq220","age","race", "gender")
# Check whether there exist NA values
sum(is.na(subdata2$viq220))
```

```
[1] 0
```

```
sum(is.na(subdata2$age))
```

```
[1] 0
```

```
sum(is.na(subdata2$race))
```

```
[1] 0
```

```
sum(is.na(subdata2$gender))
```

```
[1] 0
```

```
# Construct the model
subdata2$viq220 = as.factor(subdata2$viq220)
subdata2$race = as.factor(subdata2$race)
subdata2$gender = as.factor(subdata2$gender)
model2 = glm(viq220 ~ age+race+gender, data = subdata2, family = binomial)
summary(model2)
```

```
Call:
glm(formula = viq220 ~ age + race + gender, family = binomial,
    data = subdata2)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.836666   0.077923  23.570  < 2e-16 ***
age         -0.022574   0.001262 -17.882  < 2e-16 ***
race2       -0.156322   0.164284  -0.952 0.341332
race3       -0.668931   0.070023  -9.553  < 2e-16 ***
race4       -0.261872   0.076580  -3.420 0.000627 ***
race5       -0.650992   0.135407  -4.808 1.53e-06 ***
gender2     -0.502090   0.053011  -9.471  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8915.3  on 6544  degrees of freedom
Residual deviance: 8273.8  on 6538  degrees of freedom
AIC: 8287.8

Number of Fisher Scoring iterations: 4
```

```
### (3) ###
d3 = cbind(data$viq220,data$ridageyr,data$ridreth1,
           data$riagendr,data$indfmpir)
subdata3 = as.data.frame(d3)
colnames(subdata3) = c("viq220","age","race", "gender", "pir")
# Check whether there exist NA values
sum(is.na(subdata3$viq220))
```

```
[1] 0
```

```
sum(is.na(subdata3$age))
```

```
[1] 0
```

```r
sum(is.na(subdata3$race))
```

```
[1] 0
```

```r
sum(is.na(subdata3$gender))
```

```
[1] 0
```

```r
sum(is.na(subdata3$pir))
```

```
[1] 298
```

```r
# so, we need to do the data clean for variable pir
```

```r
library(dplyr)
new.subdata3 = filter(subdata3, !is.na(subdata3$pir))
# Construct the model
new.subdata3$viq220 = as.factor(new.subdata3$viq220)
new.subdata3$race = as.factor(new.subdata3$race)
new.subdata3$gender = as.factor(new.subdata3$gender)
model3 = glm(viq220~age+race+gender+pir,data=new.subdata3,family = binomial)
summary(model3)
```

```
Call:
glm(formula = viq220 ~ age + race + gender + pir, family = binomial,
    data = new.subdata3)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.016160   0.087788  22.966  < 2e-16 ***
age         -0.022188   0.001295 -17.135  < 2e-16 ***
race2       -0.116023   0.168265  -0.690 0.490495
race3       -0.501529   0.075149  -6.674 2.49e-11 ***
race4       -0.207385   0.079217  -2.618 0.008847 **
race5       -0.532727   0.140152  -3.801 0.000144 ***
gender2     -0.516271   0.054305  -9.507  < 2e-16 ***
pir         -0.113598   0.017707  -6.415 1.41e-10 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8519.1  on 6246  degrees of freedom
Residual deviance: 7893.8  on 6239  degrees of freedom
AIC: 7909.8

Number of Fisher Scoring iterations: 4
```

```r
# combine the result and output
odds.ratios = cbind(
  m1 = round(exp(coef(model1)),6),
  m2 = round(exp(coef(model2)),6),
  m3 = round(exp(coef(model3)),6)
)
```

```
Warning in cbind(m1 = round(exp(coef(model1)), 6), m2 =
round(exp(coef(model2)), : number of rows of result is not a multiple of vector
length (arg 2)
```

```r
odds.ratios
```

```
                  m1       m2       m3
(Intercept) 3.528843 6.275579 7.509431
age         0.975629 0.977679 0.978056
race2       3.528843 0.855284 0.890455
race3       0.975629 0.512256 0.605604
race4       3.528843 0.769610 0.812707
race5       0.975629 0.521528 0.587002
gender2     3.528843 0.605265 0.596741
pir         0.975629 6.275579 0.892617
```

```r
odds.ratios[3:8,1] = 0
odds.ratios[8,2] = 0
odds.ratios
```

```
                  m1       m2       m3
(Intercept) 3.528843 6.275579 7.509431
age         0.975629 0.977679 0.978056
race2       0.000000 0.855284 0.890455
```

```
race3       0.000000 0.512256 0.605604
race4       0.000000 0.769610 0.812707
race5       0.000000 0.521528 0.587002
gender2     0.000000 0.605265 0.596741
pir         0.000000 0.000000 0.892617
```

```
sample.size = c(nobs(model1), nobs(model2), nobs(model3))

# calculate the pseudo R^2
null.model1 = glm(viq220 ~ 1, data = subdata1, family = binomial)
pr2.m1 = 1-(as.numeric(logLik(model1))/as.numeric(logLik(null.model1)))

null.model2 = glm(viq220 ~ 1, data = subdata2, family = binomial)
pr2.m2 = 1-(as.numeric(logLik(model2))/as.numeric(logLik(null.model2)))

null.model3 = glm(viq220 ~ 1, data = new.subdata3, family = binomial)
pr2.m3 = 1-(as.numeric(logLik(model3))/as.numeric(logLik(null.model3)))
pr2 = c(pr2.m1, pr2.m2, pr2.m3)

AIC = c(model1$aic,model2$aic,model3$aic)

result = rbind(odds.ratios,sample.size, AIC, pr2)
result = as.data.frame(result)
result = round(result, 6)
result
```

```
                    m1          m2          m3
(Intercept)    3.528843    6.275579    7.509431
age            0.975629    0.977679    0.978056
race2          0.000000    0.855284    0.890455
race3          0.000000    0.512256    0.605604
race4          0.000000    0.769610    0.812707
race5          0.000000    0.521528    0.587002
gender2        0.000000    0.605265    0.596741
pir            0.000000    0.000000    0.892617
sample.size 6545.000000 6545.000000 6247.000000
AIC         8475.886616 8287.760918 7909.808221
pr2            0.049731    0.071954    0.073400
```

```
rownames(result) = c("Intercept","Age","Other Hispanic",
                     "Non-Hispanic White","Non-Hispanic Black",
                     "Other Race-Including Multi-Racial",
```

```
                    "Female","PIR","Sample Size","AIC","Pseudo R^2")
library(knitr)
kable(result)
```

	m1	m2	m3
Intercept	3.528843	6.275579	7.509431
Age	0.975629	0.977679	0.978056
Other Hispanic	0.000000	0.855284	0.890455
Non-Hispanic White	0.000000	0.512256	0.605604
Non-Hispanic Black	0.000000	0.769610	0.812707
Other Race-Including Multi-Racial	0.000000	0.521528	0.587002
Female	0.000000	0.605265	0.596741
PIR	0.000000	0.000000	0.892617
Sample Size	6545.000000	6545.000000	6247.000000
AIC	8475.886616	8287.760918	7909.808221
Pseudo R^2	0.049731	0.071954	0.073400

(d)

```
### (1) ###
# H0: the odds of men and women being wears of glasess/contact lenses
# for distance vision is not differ
# H1: H0 is not true
# alpha = 0.05
summary(model3)
```

```
Call:
glm(formula = viq220 ~ age + race + gender + pir, family = binomial,
    data = new.subdata3)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.016160   0.087788  22.966  < 2e-16 ***
age         -0.022188   0.001295 -17.135  < 2e-16 ***
race2       -0.116023   0.168265  -0.690 0.490495
race3       -0.501529   0.075149  -6.674 2.49e-11 ***
race4       -0.207385   0.079217  -2.618 0.008847 **
race5       -0.532727   0.140152  -3.801 0.000144 ***
gender2     -0.516271   0.054305  -9.507  < 2e-16 ***
```

```
pir         -0.113598   0.017707  -6.415 1.41e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8519.1  on 6246  degrees of freedom
Residual deviance: 7893.8  on 6239  degrees of freedom
AIC: 7909.8

Number of Fisher Scoring iterations: 4
```

The summary result, the coefficient of female is log(odds) and its corresponding standard error, but they will give us the same test result. From the summary, we can find the log odds ratio is significant, which means odds of females wearing glasses/contacts for distance vision is statistically significantly lower than the odds for males.

```r
# (2) ###
# H0: the the proportion of wearers of glasses/contact lenses
# for distance vision is not differs between men and women
# H1: H0 is not true
# alpha = 0.05
table.gender = table(gender=new.subdata3$gender,viq220=new.subdata3$viq220)
table.gender
```

```
      viq220
gender    1    2
     1 1134 1919
     2 1521 1673
```

```r
wear = table.gender[,"1"]
total = rowSums(table.gender)
test = prop.test(wear, total)
test
```

```
    2-sample test for equality of proportions with continuity correction

data:  wear out of total
X-squared = 69.683, df = 1, p-value < 2.2e-16
alternative hypothesis: two.sided
```

```
95 percent confidence interval:
 -0.12945505 -0.08007986
sample estimates:
   prop 1    prop 2
0.3714379 0.4762054
```

From the result, we can find the p-value is less than 0.05, so, we should reject H0 and conclude that we have confidence to say the the proportion of wearers of glasses/contact lenses for distance vision differs between men and women.

Question 2

First, import the data

```
library(DBI)
sakila = dbConnect(RSQLite::SQLite(),
                   "C:/Users/z1883/Desktop/sakila_master.db")
dbListTables(sakila)
```

```
 [1] "actor"                "address"              "category"
 [4] "city"                 "country"              "customer"
 [7] "customer_list"        "film"                 "film_actor"
[10] "film_category"        "film_list"            "film_text"
[13] "inventory"            "language"             "payment"
[16] "rental"               "sales_by_film_category" "sales_by_store"
[19] "staff"                "staff_list"           "store"
```

```
dbListFields(sakila, "film")
```

```
 [1] "film_id"              "title"                "description"
 [4] "release_year"         "language_id"          "original_language_id"
 [7] "rental_duration"      "rental_rate"          "length"
[10] "replacement_cost"     "rating"               "special_features"
[13] "last_update"
```

(a)

```
dbGetQuery(sakila, "SELECT release_year, count(release_year) FROM film
                    GROUP BY release_year")
```

```
  release_year count(release_year)
1         2006                1000
```

(b)

```
# using R
category = dbGetQuery(sakila, "SELECT * FROM category ")
film.category = dbGetQuery(sakila, "SELECT * FROM film_category ")
# do the right join
category.table = merge(category, film.category, by = "category_id",all.y = TRUE)
# convert the table to data frame
t1 = as.data.frame(table(category.table$name))
t1$Freq = as.numeric(as.character(t1$Freq))
t1[which(t1$Freq == min(t1$Freq)),]
```

```
    Var1 Freq
12 Music   51
```

```
# using SQL
dbListFields(sakila, "film_category")
```

```
[1] "film_id"     "category_id" "last_update"
```

```
dbListFields(sakila, "category")
```

```
[1] "category_id" "name"        "last_update"
```

```
dbGetQuery(sakila,"SELECT fc.category_id, c.name,
                   count(fc.category_id) AS totalNumber
                   FROM film_category AS fc
                   LEFT JOIN category AS c ON
                   fc.category_id = c.category_id
                   GROUP BY fc.category_id
                   ORDER BY totalNumber
                   LIMIT 1")
```

```
  category_id  name totalNumber
1          12 Music          51
```

(c)

```
# using R
customer = dbGetQuery(sakila, "SELECT * FROM customer")
address = dbGetQuery(sakila, "SELECT * FROM address")
city = dbGetQuery(sakila, "SELECT * FROM city")
country = dbGetQuery(sakila, "SELECT * FROM country")
m1 = merge(customer, address, by="address_id",all.x=TRUE)
m2 = merge(m1, city, by="city_id", all.x=TRUE)
m3 = merge(m2, country, by="country_id", all.x=TRUE)
```

```
Warning in merge.data.frame(m2, country, by = "country_id", all.x = TRUE):
column names 'last_update.x', 'last_update.y' are duplicated in the result
```

```
t2 = as.data.frame(table(m3$country))
t2$Freq = as.numeric(as.character(t2$Freq))
t2[which(t2$Freq == 13),]
```

```
        Var1 Freq
6  Argentina   13
68   Nigeria   13
```

```
dbGetQuery(sakila,
                   "SELECT co.country, count(co.country) AS Freq
                    FROM country AS co
                    RIGHT JOIN
                    (SELECT country_id
                    FROM city AS ci
                    RIGHT JOIN
                    (SELECT city_id
                    FROM customer AS cu
                    LEFT JOIN address AS ad
                    ON cu.address_id = ad.address_id
                    )AS ca ON ca.city_id = ci.city_id
                    )AS ci1 ON ci1.country_id = co.country_id
                    GROUP BY co.country
                    HAVING Freq == 13")
```

```
    country Freq
1 Argentina   13
2   Nigeria   13
```

Question 3

(a)

```
data = read.csv("C:/Users/z1883/Desktop/us-500/us-500.csv",header = TRUE)
nrow(data)
```

```
[1] 500
```

```
sum(grepl(".com$", data$email))/nrow(data)
```

```
[1] 0.732
```

(b)

```
# there are only one necessary "@" and one "." should be excluded
# the other should remain, so, if the address have more than one "."
# or "@", we just need to exclude once.
d = gsub(pattern = "@", replacement = "", data$email)
email = sub(pattern = "\\.", replacement = "", d)
head(email)
```

```
[1] "jbuttgmailcom"              "josephine_darakjydarakjyorg"
[3] "artvenereorg"               "lpaprockihotmailcom"
[5] "donettefollercox.net"       "simonamorascacom"
```

```
result = grepl("[^a-zA-Z0-9]", email)
mean(result)
```

```
[1] 0.506
```

(c)

```
# first let us check the digits of phone numbers
table(nchar(data$phone1))
```

```
 12
500
```

```
table(nchar(data$phone2))
```

```
 12
500
```

```
# The phone number are all 12 digits,
# So, the first 3 digits will be the are code
p1.area = as.numeric(substr(data$phone1, 1,3))
p2.area = as.numeric(substr(data$phone2, 1,3))
p.area = c(p1.area,p2.area)
area = as.data.frame(table(p.area))
area$Freq = as.numeric(as.character(area$Freq))
order.area = area[order(area$Freq, decreasing = TRUE),]
order.area[1:5,]
```

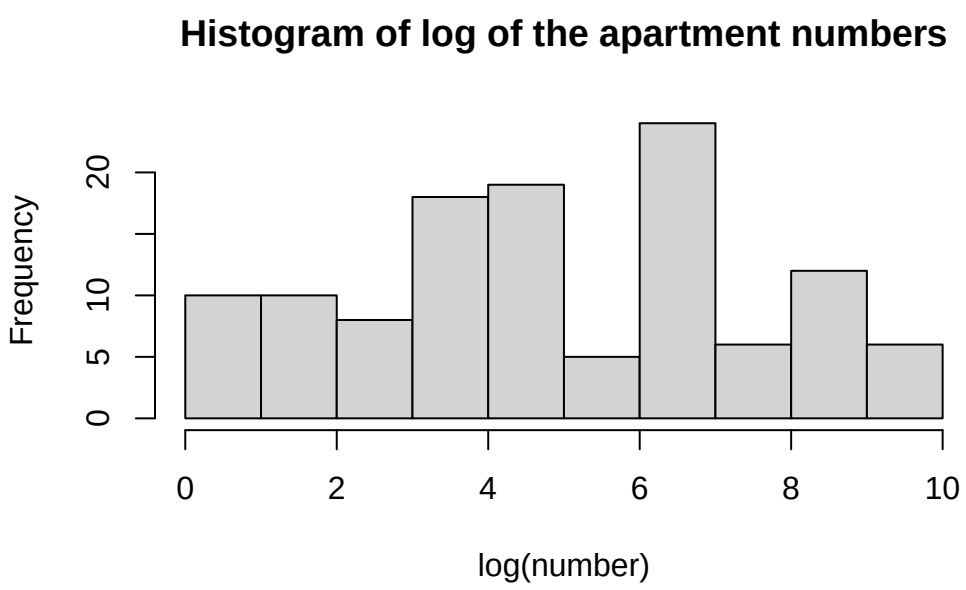
```
    p.area Freq
158    973   36
9      212   28
12     215   28
47     410   28
1      201   24
```

(d)

```
# we assume any number at the end of the an address is an apartment number.
apa.num = data$address[grepl("[0-9]+$", data$address)]
head(apa.num)
```

```
[1] "8 W Cerritos Ave #54"  "5 Boston Ave #88"      "228 Runamuck Pl #2808"
[4] "25 E 75th St #69"      "1 State Route 27"      "86 Nw 66th St #8673"
```

```
num = gsub(".*(?:#|\\s)(\\d+)$", "\\1", apa.num)
number = as.numeric(num)
hist(log(number),main="Histogram of log of the apartment numbers")
```



(e)

```
leading = substr(num, 1,1)
lead.num = as.numeric(leading)
lead.table = as.data.frame(table(lead.num))
lead.table$Freq = as.numeric(as.character(lead.table$Freq))
lead.table$lead.num = as.numeric(as.character(lead.table$lead.num))


expected.prob = log10(1 + 1/(1:9))
expected.freq = sum(lead.table$Freq)*expected.prob
expected.freq
```

```
[1] 35.521539 20.778769 14.742771 11.435382  9.343387  7.899721  6.843050
[8]  6.035998  5.399384
```

```
cbind(lead.table, expected.freq)
```

```
  lead.num Freq expected.freq
1        1   15     35.521539
2        2   13     20.778769
3        3   12     14.742771
```

```
4         4    12      11.435382
5         5    15       9.343387
6         6    11       7.899721
7         7    12       6.843050
8         8    11       6.035998
9         9    17       5.399384
```

From the result, we can find the distribution of leading digit of real numerical data does not follows the Benford's Law. It is more likely follows a uniform distribution. So, I think this apartment numbers would not pass as real data.