HW 3

Yoonseo Mok

Link to github: https://github.com/mokys1213/STATS-506-FA-2024/blob/main/HW3/HW3.pdf

```r
library(foreign)
library(kableExtra)
library(knitr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()     masks stats::filter()
## x dplyr::group_rows() masks kableExtra::group_rows()
## x dplyr::lag()        masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(gtsummary)
library(lmtest)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

Problem 1

A

```r
# Reading VIX_D and DEMO_D datasets
vix_d=read.xport('/Users/ymok/Desktop/Fall 2024/STATS-506-FA-2024/HW3/VIX_D.XPT')
demo_d=read.xport('/Users/ymok/Desktop/Fall 2024/STATS-506-FA-2024/HW3/DEMO_D.XPT')

# Merging VIX_D and DEMO_D using SEQN
```

```
p1dat=merge(vix_d,demo_d,by="SEQN")
nrow(p1dat)
```

```
## [1] 6980
```

Total sample size is 6980

B

```
# Creating 10 year age bracket variable
p1dat$age10year=NA
p1dat$age10year[p1dat$RIDAGEYR>=0 & p1dat$RIDAGEYR<=9]="0-9"
p1dat$age10year[p1dat$RIDAGEYR>=10 & p1dat$RIDAGEYR<=19]="10-19"
p1dat$age10year[p1dat$RIDAGEYR>=20 & p1dat$RIDAGEYR<=29]="20-29"
p1dat$age10year[p1dat$RIDAGEYR>=30 & p1dat$RIDAGEYR<=39]="30-39"
p1dat$age10year[p1dat$RIDAGEYR>=40 & p1dat$RIDAGEYR<=49]="40-49"
p1dat$age10year[p1dat$RIDAGEYR>=50 & p1dat$RIDAGEYR<=59]="50-59"
p1dat$age10year[p1dat$RIDAGEYR>=60 & p1dat$RIDAGEYR<=69]="60-69"
p1dat$age10year[p1dat$RIDAGEYR>=70 & p1dat$RIDAGEYR<=79]="70-79"
p1dat$age10year[p1dat$RIDAGEYR>=80 & p1dat$RIDAGEYR<=89]="80-89"

# Recoding glasses/contact lenses for distance vision variable
p1dat$VIQ220[p1dat$VIQ220 %in% 1]=1
p1dat$VIQ220[p1dat$VIQ220 %in% 2]=0
p1dat$VIQ220[p1dat$VIQ220 %in% 9]=NA

# Creating table for the proportion of respondents
kable(p1dat %>% group_by(age10year) %>% count(VIQ220) %>% mutate(Percentage = n / sum(n) * 100) %>%
 filter(VIQ220 ==1) %>% select(-n,-VIQ220) ,
  col.names = c('Age (10 years)', 'Percentage'))
```

Age (10 years)	Percentage
10-19	30.35795
20-29	29.97062
30-39	32.88509
40-49	35.09202
50-59	53.09033
60-69	59.30408
70-79	63.75267
80-89	58.10056

C

```
# Renaming age variable
p1dat$Age=p1dat$RIDAGEYR
```

```r
# Recoding race variable
p1dat$Race=NA
p1dat$Race[p1dat$RIDRETH1 %in% 1]="Mexican American"
p1dat$Race[p1dat$RIDRETH1 %in% 2]="Other Hispanic"
p1dat$Race[p1dat$RIDRETH1 %in% 3]="Non-Hispanic White"
p1dat$Race[p1dat$RIDRETH1 %in% 4]="Non-Hispanic Black"
p1dat$Race[p1dat$RIDRETH1 %in% 5]="Other Race"

# Recoding gender variable
p1dat$Gender=NA
p1dat$Gender[p1dat$RIAGENDR %in% 1]="Male"
p1dat$Gender[p1dat$RIAGENDR %in% 2]="Female"

# Recoding income variable
p1dat$Income_Ratio=p1dat$INDFMPIR

# Logistic regression with age
model1 <- glm(VIQ220 ~ Age, data = p1dat, family = binomial)
# Logistic regression with age, race, gender
model2 <- glm(VIQ220 ~ Age + Race + Gender, data = p1dat, family = binomial)
# Logistic regression with age, race, gender, Poverty Income ratio
model3 <- glm(VIQ220 ~ Age + Race + Gender + Income_Ratio, data = p1dat, family = binomial)

# Computing psuedo-R^2
model1_r2 = 1 - (model1$deviance / model1$null.deviance)
model2_r2 = 1 - (model2$deviance / model2$null.deviance)
model3_r2 = 1 - (model3$deviance / model3$null.deviance)

# Generate table
model1 %>% tbl_regression(exponentiate = TRUE) %>%
  add_glance_table(include = c(AIC,nobs))%>%
  modify_table_body(~add_row(.,label = "pseudo R-squared",estimate = model1_r2, row_type = "label" ))
```

Characteristic	**OR**[1]	**95% CI**[1]	**p-value**
Age	1.02	1.02, 1.03	<0.001
AIC	8,476		
No. Obs.	6,545		
pseudo R-squared	0.05		

[1] OR = Odds Ratio, CI = Confidence Interval

```r
model2 %>% tbl_regression(exponentiate = TRUE) %>%
  add_glance_table(include = c(AIC,nobs))%>%
  modify_table_body(~add_row(.,label = "pseudo R-squared",estimate = model2_r2, row_type = "label" ))
```

Characteristic	**OR**[1]	**95% CI**[1]	**p-value**
Age	1.02	1.02, 1.03	<0.001
Race			

Mexican American	—	—	
Non-Hispanic Black	1.30	1.12, 1.51	<0.001
Non-Hispanic White	1.95	1.70, 2.24	<0.001
Other Hispanic	1.17	0.84, 1.61	0.3
Other Race	1.92	1.47, 2.50	<0.001
Gender			
Female	—	—	
Male	0.61	0.55, 0.67	<0.001
AIC	8,288		
No. Obs.	6,545		
pseudo R-squared	0.07		

[1] OR = Odds Ratio, CI = Confidence Interval

```
model3 %>% tbl_regression(exponentiate = TRUE) %>%
  add_glance_table(include = c(AIC,nobs))%>%
  modify_table_body(~add_row(.,label = "pseudo R-squared",estimate = model3_r2, row_type = "label" ))
```

Characteristic	OR[1]	95% CI[1]	p-value
Age	1.02	1.02, 1.03	<0.001
Race			
Mexican American	—	—	
Non-Hispanic Black	1.23	1.05, 1.44	0.009
Non-Hispanic White	1.65	1.43, 1.91	<0.001
Other Hispanic	1.12	0.80, 1.56	0.5
Other Race	1.70	1.29, 2.24	<0.001
Gender			
Female	—	—	
Male	0.60	0.54, 0.66	<0.001
Income_Ratio	1.12	1.08, 1.16	<0.001
AIC	7,910		
No. Obs.	6,247		
pseudo R-squared	0.07		

[1] OR = Odds Ratio, CI = Confidence Interval

D-1

```
# Testing whether the odds of men and women being wears of glasess/contact lenses for distance
# vision differs from model 3
summary(model3)$coefficients["GenderMale",]
```

```
##      Estimate    Std. Error       z value      Pr(>|z|)
## -5.162713e-01  5.430496e-02 -9.506891e+00  1.964460e-21
```

p-value is less than 0.05. We can conclude that the odds of men and women being wears of glasess/contact lenses for distance vision differs.

D-2

```
# Testing whether the proportion of wearers of glasses/contact lenses for distance vision differs
# between men and women from model 3

# Performing Chi-squared test
chisq.test(table(p1dat$Gender, p1dat$VIQ220))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(p1dat$Gender, p1dat$VIQ220)
## X-squared = 70.955, df = 1, p-value < 2.2e-16
```

Since p-value is less than 0.05, we can conclude that the proportion of wearers of glasses/contact lenses for distance vision differs between men and women.

Problem 2

```
# Loading "sakila" database
library(RSQLite)
p2dat <- dbConnect(RSQLite::SQLite(), "sakila_master.db")
```

A

```
# Using single SQL query to calculate oldest year and count

#dbListTables(p2dat)
#dbListFields(p2dat, "film")
dbGetQuery(p2dat, "SELECT release_year, COUNT(*) AS movie_count
           FROM film")
```

```
##   release_year movie_count
## 1         2006        1000
```

The oldest movie is from 2006 and 1000 movies were released in that year

B-1

```r
# Genre of movie is the least common in the data and count using R

# Using SQL query to extract film_category table
film_cat=as.data.frame(dbGetQuery(p2dat, "SELECT film_ID, category_ID FROM film_category"))
# Using SQL query to extract category table
cat=as.data.frame(dbGetQuery(p2dat, "SELECT category_ID, name FROM category"))

# Merging film_cat and cat dataframe
b1=merge(film_cat,cat,by="category_id")

# Generating counts
b1 %>% group_by(category_id,name) %>% summarize(n=n()) %>% arrange(n)
```

```
## `summarise()` has grouped output by 'category_id'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 16 x 3
## # Groups:   category_id [16]
##    category_id name            n
##          <int> <chr>       <int>
##  1          12 Music          51
##  2          11 Horror         56
##  3           4 Classics       57
##  4          16 Travel         57
##  5           5 Comedy         58
##  6           3 Children       60
##  7          10 Games          61
##  8          14 Sci-Fi         61
##  9           7 Drama          62
## 10          13 New            63
## 11           1 Action         64
## 12           2 Animation      66
## 13           6 Documentary    68
## 14           8 Family         69
## 15           9 Foreign        73
## 16          15 Sports         74
```

Music genre is the least common in the data with 51 of music genre.

B-2

```r
# Genre of movie is the least common in the data and count using SQL query
#dbGetQuery(p2dat,"SELECT * FROM film_category")

dbGetQuery(p2dat,"SELECT film_category.category_ID AS category_ID, film_category.film_id AS film_id, na
                  FROM film_category
                        LEFT JOIN
                        (SELECT *
                          FROM category) AS c on c.category_ID =film_category.category_ID
```

```
                GROUP BY genre
                ORDER by count
           ")
```

```
##    category_ID film_id       genre count
## 1           12      12       Music    51
## 2           11       2      Horror    56
## 3            4      14    Classics    57
## 4           16      41      Travel    57
## 5            5       7      Comedy    58
## 6            3      48    Children    60
## 7           10      46       Games    61
## 8           14      26      Sci-Fi    61
## 9            7      33       Drama    62
## 10          13      22         New    63
## 11           1      19      Action    64
## 12           2      18   Animation    66
## 13           6       1 Documentary    68
## 14           8       5      Family    69
## 15           9       6     Foreign    73
## 16          15      10      Sports    74
```

Music is the least common genre in the data, with 51 counts in music genre.

C-1

```
# Identifying which country or countries have exactly 13 customers using R

# Using SQL query to extract customer table
customer=as.data.frame(dbGetQuery(p2dat, "SELECT customer_ID, address_ID FROM customer"))
# Using SQL query to extract address table
address=as.data.frame(dbGetQuery(p2dat, "SELECT address_ID, city_ID FROM address"))
# Using SQL query to extract city table
city=as.data.frame(dbGetQuery(p2dat, "SELECT city_ID, city, country_ID FROM city"))
# Using SQL query to extract country table
country=as.data.frame(dbGetQuery(p2dat, "SELECT country_ID, country FROM country"))

tmp=merge(customer,address,by="address_id")
tmp2=merge(country,city,by="country_id")
c1=merge(tmp,tmp2,by="city_id")
c1 %>% group_by(country) %>% summarize(n=n()) %>% filter(n %in% 13)
```

```
## # A tibble: 2 x 2
##   country       n
##   <chr>     <int>
## 1 Argentina    13
## 2 Nigeria      13
```

Argentina and Nigeria has 13 customers

C-2

```
# Identifying which country or countries have exactly 13 customers using SQL query

dbGetQuery(p2dat,"SELECT country.country, COUNT(customer.customer_id) AS count
                  FROM customer
                    JOIN address on customer.address_id= address.address_id
                    JOIN city on address.city_id =city.city_id
                    JOIN country on city.country_id = country.country_id
                  GROUP BY country.country
                  HAVING count=13")
```

```
##     country count
## 1 Argentina    13
## 2   Nigeria    13
```

Argentina and Nigeria has 13 customers.

Problem 3

```
# Importing the "US - 500 Records" data
us500=read.csv("us-500.csv")
```

A

```
# Calculating the proportion of email addresses hosted at a domain with TLD ".com"
p3a=us500[substr(us500$email,nchar(us500$email)-3,nchar(us500$email)) %in% ".com",]
nrow(p3a)/nrow(us500)*100
```

```
## [1] 73.2
```

73.2% of email addresses are hosted at a domain with TLD ".com"

B

```
# Calculating proportion of email addresses that have at least one non alphanumeric character in them
length(grep("[^a-zA-Z0-9@.]", us500$email))/nrow(us500)*100
```

```
## [1] 24.8
```

24.8% of email addresses have at least one non alphanumeric character in them

C

```r
# Top 5 most common area codes amongst all phone numbers
# First 3 digits of phone 1
us500$areacode1=substr(us500$phone1,1,3)
# First 3 digits of phone 2
us500$areacode2=substr(us500$phone2,1,3)
# Top 5 of phone 1
sort(table(us500$areacode1),decreasing = T)[1:5]
```

```
##
## 973 212 215 410 201
##  18  14  14  14  12
```

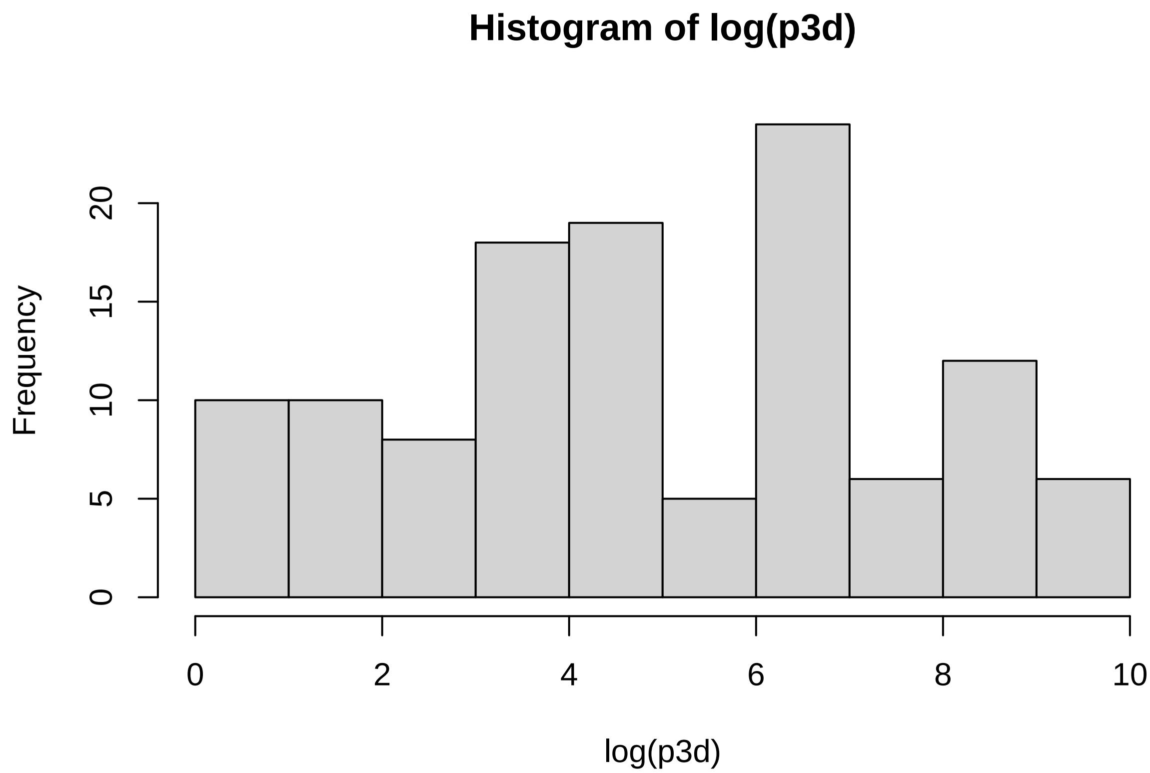
```r
# Top 5 of phone 2
sort(table(us500$areacode2),decreasing = T)[1:5]
```

```
##
## 973 212 215 410 201
##  18  14  14  14  12
```

The top 5 most common area codes amongst all phone numbers are 973,212,215,410,201.

D

```r
# Producing a histogram of the log of the apartment numbers for all addresses.
# Extracting apartment numbers
p3d=as.numeric(regmatches(us500$address, regexpr("[0-9]+$", us500$address)))
# Generating histogram
hist(log(p3d))
```



E

The apartment numbers does not appear to follow Benford's law. I don't think the apartment numbers would pass as real data. We have more frequency on 7, not on the 1.