Problem Set 3

Yifan Li

Link to the GitHub

The link to my GitHub repository is https://github.com/FYlee39/Stats-506/tree/main/PS3.

Problem 1

a.

```
library(haven)
vix_d_data <- read_xpt("VIX_D.XPT")
demo_d_data <- read_xpt("DEMO_D.XPT")
# Merging two data frame based one SEQN
merged_df <- merge(vix_d_data, demo_d_data, by="SEQN", all=FALSE)
nrow(merged_df) == 6980
```

```
[1] TRUE
```

The total sample size is 6980.

b.

```
library(dplyr)
```

```
Attaching package: 'dplyr'
```

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

    filter, lag


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

    intersect, setdiff, setequal, union
```

```r
library(knitr)

# Create age brackets
merged_df <- merged_df %>%
  mutate(age_bracket = cut(RIDAGEYR, breaks=seq(0, 100, by=10), include.lowest=TRUE))

# Calculate proportions of wearing glasses/contact lenses for distance vision
glasses_prop <- merged_df %>%
  group_by(age_bracket) %>%
  summarize(prop_wearing_glasses = mean(VIQ220 == 1, na.rm = TRUE))

# build the table
kable(glasses_prop,
      caption = "Proportion of Respondents Wearing Glasses/Contacts by Age Bracket")
```

Table 1: Proportion of Respondents Wearing Glasses/Contacts by Age Bracket

age_bracket	prop_wearing_glasses
(10,20]	0.3165899
(20,30]	0.3404030
(30,40]	0.3503268
(40,50]	0.3890374
(50,60]	0.5625000
(60,70]	0.6337115
(70,80]	0.6768868
(80,90]	0.6544118

c.

```r
library(stargazer)
```

```
Please cite as:

 Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.

 R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```
# modified the glasses data so that it can be used in the logistics model
merged_df <- merged_df %>%
  mutate(VIQ220 = case_when(
    VIQ220 == 1 ~ 1,    # 1 means yes, keep as 1
    VIQ220 == 2 ~ 0,    # 2 means no, recode as 0
    VIQ220 == 9 ~ NA_real_,  # 9 means unknown/missing, treat as NA
    TRUE ~ NA_real_       # Handle any other cases as NA
  ))

# Model 1: age
model_1 <- glm(VIQ220 ~ RIDAGEYR,
               data=merged_df, family="binomial")

# Model 2: age, race, gender
model_2 <- glm(VIQ220 ~ RIDAGEYR + RIDRETH1 + RIAGENDR ,
               data=merged_df, family="binomial")

# Model 3: age, race, gender, poverty income ratio
model_3 <- glm(VIQ220 ~ RIDAGEYR + RIDRETH1 + RIAGENDR  + INDFMPIR ,
               data=merged_df, family="binomial")

#' Function to compute pseudo-R^2 values
#'
#' @param model the logistic regression model
#' @return p_r_value the pseudo R^2 value
pseudo_r_squared <- function(model) {
  null_dev <- model$null.deviance
  res_dev <- model$deviance
  p_r_value <- 1 - (res_dev / null_dev)
  return(p_r_value)
}

# Create a stargazer table with odds ratios
stargazer(model_1, model_2, model_3,
          type = "text",
```

```r
          coef = list(exp(coef(model_1)), exp(coef(model_2)), exp(coef(model_3))),
          p.auto = TRUE,
          apply.coef = exp,
          report = "vc*p", # shows estimates, standard errors, p-values
          ci = TRUE,
          dep.var.labels = "Wears Glasses/Contact Lenses for Distance Vision",
          covariate.labels = c("Age", "Race", "Gender", "Poverty Income Ratio"),
          add.lines = list(c("Sample Size", nrow(merged_df)),
                           c("Pseudo-R²", round(pseudo_r_squared(model_1), 2),
                               round(pseudo_r_squared(model_2), 2),
                               round(pseudo_r_squared(model_3), 2)),
                           c("AIC", AIC(model_1), AIC(model_2), AIC(model_3)))
          )
```

```
========================================================================
                                       Dependent variable:
                     ---------------------------------------------------
                      Wears Glasses/Contact Lenses for Distance Vision
                            (1)               (2)               (3)
------------------------------------------------------------------------
Age                       2.787***          2.788***         2.784***
                          p = 0.000        p = 0.000        p = 0.000

Race                                        3.104***         2.996***
                                           p = 0.000        p = 0.000

Gender                                      5.184***         5.364***
                                           p = 0.000        p = 0.000

Poverty Income Ratio                                         3.169***
                                                            p = 0.000

Constant                  1.328***          1.096***         1.074***
                          p = 0.000        p = 0.000        p = 0.000

------------------------------------------------------------------------
Sample Size                 6980
Pseudo-R²                   0.05              0.06             0.07
AIC                  8475.88661639229  8358.4955583034  7940.7895500819
Observations                6,545             6,545            6,247
Log Likelihood           -4,235.943        -4,175.248       -3,965.395
```

```
Akaike Inf. Crit.        8,475.887        8,358.496        7,940.790
=======================================================================
Note:                                          *p<0.1; **p<0.05; ***p<0.01
```

d.

```r
library(aod)

# Gender comparison test
summary(model_3)
```

```
Call:
glm(formula = VIQ220 ~ RIDAGEYR + RIDRETH1 + RIAGENDR + INDFMPIR,
    family = "binomial", data = merged_df)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.634169   0.128457 -20.506  < 2e-16 ***
RIDAGEYR     0.023763   0.001262  18.829  < 2e-16 ***
RIDRETH1     0.092776   0.023564   3.937 8.24e-05 ***
RIAGENDR     0.518595   0.054121   9.582  < 2e-16 ***
INDFMPIR     0.142601   0.017011   8.383  < 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: 8519.1  on 6246  degrees of freedom
Residual deviance: 7930.8  on 6242  degrees of freedom
  (733 observations deleted due to missingness)
AIC: 7940.8

Number of Fisher Scoring iterations: 4
```

```r
# Wald test for gender coefficient
# Assuming gender is the 3rd coefficient
wald.test(b=coef(model_3), Sigma=vcov(model_3), Terms=3)
```

```
Wald test:
```

```
----------

Chi-squared test:
X2 = 15.5, df = 1, P(> X2) = 8.2e-05
```

Thus one can conclude that there is a statistically significant difference in the proportion of men and women who wear glasses/contact lenses for distance vision. Specifically, the odds of men wearing glasses differ from the odds of women wearing them.

```
# Proportion test between men and women
prop.test(table(merged_df$VIQ220, merged_df$VIQ220))
```

```

	2-sample test for equality of proportions with continuity correction

data:  table(merged_df$VIQ220, merged_df$VIQ220)
X-squared = 6540.9, df = 1, p-value < 2.2e-16
alternative hypothesis: two.sided
95 percent confidence interval:
 0.9996869 1.0000000
sample estimates:
prop 1 prop 2
     1      0
```

Thus one can conclude that there is a statistically significant difference in the proportions of glasses wearers between the two groups being compared. The extremely low p-value suggests that the observed difference is unlikely due to random chance.

Problem 2.

a.

```
library(DBI)
library(RSQLite)

sakila <- dbConnect(RSQLite::SQLite(), "sakila_master.db")
```

In the 'film' table, select the minimum value of 'release_year' and count the number of films that were released in that year.

```
query_a <- "SELECT MIN(release_year), COUNT(*)
          FROM film
          HAVING release_year = MIN(release_year)"
dbGetQuery(sakila, query_a)
```

```
  MIN(release_year) COUNT(*)
1              2006     1000
```

Thus from the table, the oldest releasing year is 2006 and there are 1000 movies released in that year.

b.

For combing SQL and R:

```
query_b_1 <- "SELECT f.film_id, fcc.name
              FROM film AS f
              INNER JOIN (
                SELECT fc.film_id, fc.category_id, c.name
                FROM film_category AS fc
                INNER JOIN category AS c ON fc.category_id == c.category_id
              ) AS fcc ON f.film_id == fcc.film_id
              "
table_b_1 <- dbGetQuery(sakila, query_b_1)

# Count the number of genre
genre_counts <- table(table_b_1$name)
least_common_genre <- names(which.min(genre_counts))
least_common_genre
```

```
[1] "Music"
```

```
least_common_count <- min(genre_counts)
least_common_count
```

```
[1] 51
```

Thus the genre of movie with least common data is 'Music' and there is only 51 movies of that genre.

For SQL only:

```
query_b_2 <- "SELECT fcc.name, COUNT(f.film_id) AS number
              FROM film AS f
              INNER JOIN (
                SELECT fc.film_id, fc.category_id, c.name
                FROM film_category AS fc
                INNER JOIN category AS c ON fc.category_id == c.category_id
              ) AS fcc ON f.film_id == fcc.film_id
              GROUP BY fcc.name
              ORDER BY number
              LIMIT 1
              "
dbGetQuery(sakila, query_b_2)
```

```
   name number
1 Music     51
```

Thus the genre of movie with least common data is 'Music' and there is only 51 movies of that genre.

c.

```
For combing SQL and R:
```

```
query_c_1 <- "SELECT cuac.customer_id, cou.country
              FROM country AS cou
                INNER JOIN (
                  SELECT c.country_id, c.city_id, cua.customer_id
                  FROM city AS c
                  INNER JOIN (
                    SELECT cu.customer_id, cu.address_id, a.city_id
                    FROM customer AS cu
                    INNER JOIN
                      address as a ON cu.address_id == a.address_id
                  ) AS cua ON c.city_id == cua.city_id
                ) AS cuac ON cuac.country_id == cou.country_id
              "
```

```
table_c_1 <- dbGetQuery(sakila, query_c_1)

country_counts <- table(table_c_1$country)
target_countries <- country_counts[country_counts == 13]
target_countries
```

```
Argentina   Nigeria
       13        13
```

Thus there are two countries whose numbers of customers are 13, they are Argentina and Nigeria.

For SQL only:

```
query_c_2 <- "SELECT COUNT(cuac.customer_id) AS number , cou.country
              FROM country AS cou
                INNER JOIN (
                  SELECT c.country_id, c.city_id, cua.customer_id
                  FROM city AS c
                  INNER JOIN (
                    SELECT cu.customer_id, cu.address_id, a.city_id
                    FROM customer AS cu
                    INNER JOIN
                      address as a ON cu.address_id == a.address_id
                  ) AS cua ON c.city_id == cua.city_id
                ) AS cuac ON cuac.country_id == cou.country_id
                GROUP BY cou.country
                HAVING number == 13
              "
dbGetQuery(sakila, query_c_2)
```

```
  number   country
1     13 Argentina
2     13   Nigeria
```

Thus there are two countries whose numbers of customers are 13, they are Argentina and Nigeria.

Problem 3.

a.

```r
raw_data <- read.csv("us-500.csv")
email_info <- raw_data$email
target_proportion_a <- mean(grepl("\\.com$", email_info))
target_proportion_a
```

```
[1] 0.732
```

Thus the proportion of email addresses are hosted at a domain with TLD ".com" is 73.2%.

b.

```r
# First create a list that containing the email address excluding the required "@" and "."
new_email_info <- gsub('[@\\.]', '', email_info)
target_proportion_b <- mean(grepl("[^0-9a-zA-Z]+", new_email_info))
target_proportion_b
```

```
[1] 0.248
```

Thus the proportion of email addresses have at least one non alphanumeric character in them is 24.8%.

c.

```r
library(stringr)

phone1_info <- raw_data$phone1
phone2_info <- raw_data$phone2

area_codes_1 <- unlist(sapply(phone1_info,
                              function(x) regmatches(x, gregexpr("^\\d{3}", x))))
area_codes_2 <- unlist(sapply(phone2_info,
                              function(x) regmatches(x, gregexpr("^\\d{3}", x))))
```

```r
area_codes_1_table <- table(area_codes_1)
area_codes_2_table <- table(area_codes_2)

top_5_area_code_1 <- as.numeric(names(sort(area_codes_1_table, decreasing=TRUE)[1:5]))
top_5_area_code_2 <- as.numeric(names(sort(area_codes_2_table, decreasing=TRUE)[1:5]))

top_5_area_code_1
```

```
[1] 973 212 215 410 201
```

```r
top_5_area_code_2
```

```
[1] 973 212 215 410 201
```

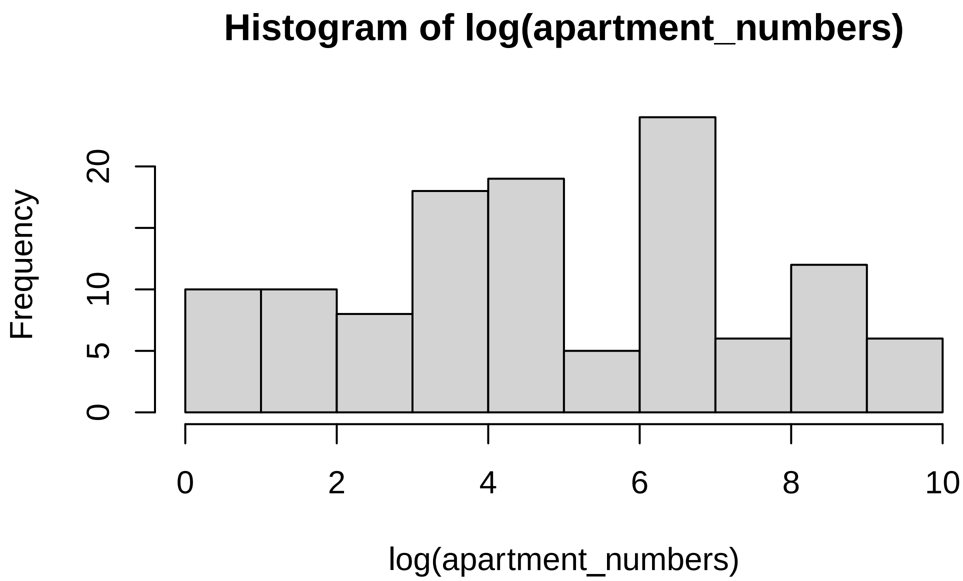
For phione1, the top 5 most common area codes among all phone numbers are $973, 212, 215, 410, 201$, For phione2, the top 5 most common area codes among all phone numbers are $973, 212, 215, 410, 201$

d.

```r
address_info <- raw_data$address

apartment_numbers <- unlist(sapply(address_info, function(x) {
  matches <- regmatches(x, regexpr("\\d+$", x))
  as.numeric(matches)}
  ))

# Produce a histogram of the log of the apartment numbers
hist(log(apartment_numbers))
```



e.

```
#get the leading digits
leading_digits <- sapply(apartment_numbers, function(x) {
  as.numeric(substr(as.character(x), 1, 1))}
  )

# Get the frequency of each leading digit
leading_digit_freq <- table(leading_digits)

# Compare with Benford's law distribution
benford_dist <- c(0.301, 0.176, 0.125, 0.097, 0.079, 0.067, 0.058, 0.051, 0.046)

# Normalize the frequencies
leading_digit_prop <- prop.table(leading_digit_freq)

# the comparison table
data.frame(
  Leading_Digit = 1:9,
  Observed_Proportion = leading_digit_prop,
```

```
  Benford_Proportion = benford_dist
)
```

```
  Leading_Digit Observed_Proportion.leading_digits Observed_Proportion.Freq
1             1                                  1               0.12711864
2             2                                  2               0.11016949
3             3                                  3               0.10169492
4             4                                  4               0.10169492
5             5                                  5               0.12711864
6             6                                  6               0.09322034
7             7                                  7               0.10169492
8             8                                  8               0.09322034
9             9                                  9               0.14406780
  Benford_Proportion
1              0.301
2              0.176
3              0.125
4              0.097
5              0.079
6              0.067
7              0.058
8              0.051
9              0.046
```

Judging from the table, the observed frequency differ the frequency of Benford distribution a lot, thus one can argue that the apartment numbers would not pass as real data.