Stats 506 PS3

Alyssa Yang

GitHub repo link: https://github.com/alyssawyang/stats506ps3

Problem 1 - Vision

1a

```
library(knitr)
library(haven)

# Read in files
vix <- read_xpt("VIX_D.XPT")
demo <- read_xpt("DEMO_D.XPT")

# Merge the two datasets
vision <- data.frame(merge(vix, demo, by = "SEQN"))

# Print total sample size - should be 6980
cat("Total sample size: ", nrow(vision))
```

```
## Total sample size:  6980
```

1b

```
# VIQ220 - Glasses/contact lenses worn for distance
# RIDAGEMN - Age in Years

# Rename variable columns
colnames(vision)[colnames(vision) == "VIQ220"] <- "glasses"
colnames(vision)[colnames(vision) == "RIDAGEYR"] <- "age"

# Replace all missing values with NA
vision[vision == "."] <- NA

# Redefine glasses variable
vision$glasses[vision$glasses == 1] <- 0
vision$glasses[vision$glasses == 2] <- 1
vision$glasses[vision$glasses == 9] <- NA

# Remove all NA values for these two variables
```

```r
vision <- vision[!is.na(vision$age), ]
vision <- vision[!is.na(vision$glasses), ]
```

```r
# Convert age in months into age brackets of 10 years
vision$age_brackets <- cut(vision$age, breaks = seq(0, 90, by = 10), right = FALSE,
                           labels = c("0-9", "10-19", "20-29", "30-39", "40-49",
                                      "50-59", "60-69", "70-79", "80-89"))

# Compute proportions for each age bracket
proportions <- aggregate(glasses ~ age_brackets, data = vision, FUN = mean)
colnames(proportions) <- c("Age Bracket", "Proportion Wearing Glasses/Contacts")

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

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

Age Bracket	Proportion Wearing Glasses/Contacts
10-19	0.6791188
20-29	0.6734258
30-39	0.6413333
40-49	0.6300129
50-59	0.4499179
60-69	0.3777778
70-79	0.3310962
80-89	0.3311897

1c

```r
# RIDRETH1 - Race/Ethnicity
# RIAGENDR - Gender
# INDFMPIR - Family Poverty Income Ratio

suppressMessages(library(pscl)) # For Pseudo-R^2 function

# Rename variable columns
colnames(vision)[colnames(vision) == "RIDRETH1"] <- "race"
colnames(vision)[colnames(vision) == "RIAGENDR"] <- "gender"
colnames(vision)[colnames(vision) == "INDFMPIR"] <- "pir"

# Convert race and gender to factors
vision$race <- as.factor(vision$race)
vision$gender <- as.factor(vision$gender)

# Rename levels of race and gender
levels(vision$race) <- c("Mexican American", "Other Hispanic", "Non-Hispanic White", "Non-Hispanic Blac
                         "Other Race - Including Multi-Racial")
levels(vision$gender) <- c("Male", "Female")
```

```r
# Model 1
model1 <- glm(glasses ~ age, data = vision, family = "binomial")

# Find values for the table
odds1 <- exp(coef(model1)) # Convert coefficients from log-odds to odds-ratios
size1 <- length(model1$fitted.values) # Find sample size
pseudo_r2_output <- capture.output(pseudo_r2_1 <- pR2(model1)) # Suppress output of pR2
pseudo_r2_1 <- pseudo_r2_1["McFadden"] # Find pseudo R^2 value
aic1 <- AIC(model1) # Find AIC
```

```r
# Remove all NA values for race and gender
vision <- vision[!is.na(vision$race), ]
vision <- vision[!is.na(vision$gender), ]

# Model 2
model2 <- glm(glasses ~ age + race + gender, data = vision, family = "binomial")

# Find values for the table
odds2 <- exp(coef(model2)) # Convert coefficients from log-odds to odds-ratios
size2 <- length(model2$fitted.values) # Find sample size
pseudo_r2_output <- capture.output(pseudo_r2_2 <- pR2(model2)) # Suppress output of pR2
pseudo_r2_2 <- pseudo_r2_2["McFadden"] # Find pseudo R^2 value
aic2 <- AIC(model2) # Find AIC
```

```r
# Remove NA values for pir
vision <- vision[!is.na(vision$pir), ]

# Model 3
model3 <- glm(glasses ~ age + race + gender + pir, data = vision, family = "binomial")

# Find values for the table
odds3 <- exp(coef(model3)) # Convert coefficients from log-odds to odds-ratios
size3 <- length(model3$fitted.values) # Find sample size
pseudo_r2_output <- capture.output(pseudo_r2_3 <- pR2(model3)) # Suppress output of pR2
pseudo_r2_3 <- pseudo_r2_3["McFadden"] # Find pseudo R^2 value
aic3 <- AIC(model3) # Find AIC
```

```r
# Find the maximum length of the variable names from all models
max_length <- max(length(names(odds1)), length(names(odds2)), length(names(odds3)))

# Create list of variable names
var_names <- c("(Intercept)", "Age", "Other Hispanic", "Non-Hispanic White", "Non-Hispanic Black",
               "Other Race - Including Multi-racial", "Female", "Poverty Income Ratio")

# Create a data frame with the odds ratios
odds <- data.frame(
  Variable = var_names,
  Model1 = c(odds1, rep(NA, max_length - length(odds1))),
  Model2 = c(odds2, rep(NA, max_length - length(odds2))),
  Model3 = odds3
)

# Create a data frame with the statistics
```

```r
stats <- data.frame(
  Variable = c("Sample Size", "Pseudo R²", "AIC"),
  Model1 = c(size1, pseudo_r2_1, aic1),
  Model2 = c(size2, pseudo_r2_2, aic2),
  Model3 = c(size3, pseudo_r2_3, aic3)
)

# Create a separator data frame for easier readability
separator_row <- data.frame(
  Variable = "-",
  Model1 = "-",
  Model2 = "-",
  Model3 = "-"
)

# Combine data frames together
results <- rbind(odds, separator_row, stats)

# Create the table using kable
kable(results, caption = "Summary of Logistic Regression Models", row.names = FALSE,
      col.names = c("Variable", "Model 1", "Model 2", "Model 3"))
```

Table 2: Summary of Logistic Regression Models

Variable	Model 1	Model 2	Model 3
(Intercept)	3.52884283250537	6.27557877055865	7.50943111677432
Age	0.97562897271348	0.977678717174749	0.978055999935046
Other Hispanic	NA	0.855283698504547	0.890455172276266
Non-Hispanic White	NA	0.512255998722561	0.605604012440809
Non-Hispanic Black	NA	0.76960959108977	0.812707057627795
Other Race - Including Multi-racial	NA	0.521528191890914	0.587001983390827
Female	NA	0.605264585170351	0.596741475791057
Poverty Income Ratio	NA	NA	0.892616931108457
-	-	-	-
Sample Size	6545	6545	6247
Pseudo R^2	0.0497312271885485	0.0719544547407939	0.0733995246302627
AIC	8475.88661639229	8287.76091821125	7909.8082207605

1d

```r
suppressMessages(library(multcomp))

# Extract the coefficient for Females in model 3
print(summary(model3))
```

```
##
## Call:
## glm(formula = glasses ~ age + race + gender + pir, family = "binomial",
##     data = vision)
##
```

```
## 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 ***
## raceOther Hispanic                      -0.116023   0.168265  -0.690 0.490495
## raceNon-Hispanic White                  -0.501529   0.075149  -6.674 2.49e-11 ***
## raceNon-Hispanic Black                  -0.207385   0.079217  -2.618 0.008847 **
## raceOther Race - Including Multi-Racial -0.532727   0.140152  -3.801 0.000144 ***
## genderFemale                            -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
```

```
cat("Odds-Ratio for Females: ", odds3["genderFemale"])
```

```
## Odds-Ratio for Females:  0.5967415
```

The odds-ratio for females is around 0.5967, and from the summary, we can see that the p-value for the genderFemale coefficient is $<$ 2e-16 which is less than level $\alpha = 0.05$. Thus, this result is statistically significant, and we can conclude that the odds of women being wearers of glasses/contact lenses for distance vision is lower than the odds for males.

```
# Create contingency table for gender vs glasses
table_gender_glasses <- table(vision$gender, vision$glasses)

# Find proportions of glasses-wearers for men and women
prop_male <- table_gender_glasses["Male", "1"] / sum(table_gender_glasses["Male", ])
prop_female <- table_gender_glasses["Female", "1"] / sum(table_gender_glasses["Female", ])

cat('Proportion of male glasses-wearers: ', prop_male, '\n')
```

```
## Proportion of male glasses-wearers:  0.6285621
```

```
cat('Proportion of female glasses-wearers: ', prop_female, '\n')
```

```
## Proportion of female glasses-wearers:  0.5237946
```

```
# Perform Chi-Square test
chi_test <- chisq.test(table_gender_glasses)
print(chi_test)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
```

```
##
## data:  table_gender_glasses
## X-squared = 69.683, df = 1, p-value < 2.2e-16
```

The p-value from the chi-square test is $<$ 2.2e-16, which is less than significant level $\alpha = 0.05$, thus we can conclude that the proportion of glasses wearers is significantly different between men and women. From the contingency table, we can see that the proportion of women being wearers of glasses/contact lenses for distance vision is lower than the proportion for males.

Problem 2 - Sakila

```
library(DBI)
library(RSQLite)

# Connect to Sakila database
sakila <- dbConnect(SQLite(), "sakila_master.db")

# Shortcut for queries
gg <- function(query) {
  dbGetQuery(sakila, query)
}
```

2a

```
gg("
SELECT release_year AS year, count(film_id) AS num_movies
  FROM film
 ORDER BY release_year
 LIMIT 1
")
```

```
##   year num_movies
## 1 2006       1000
```

2b

```
# Extract tables using SQL queries
film_category <- gg("SELECT * FROM film_category")
category <- gg("SELECT * FROM category")

# Find the genre with the least amount of movies
genre_counts <- table(film_category["category_id"])
min_genre <- which.min(genre_counts)

# Match the category_ids to find the name of the genre
genre <- category$name[category["category_id"] == min_genre]
```

```r
# Find number of movies with this genre
movies <- film_category$film_id[film_category["category_id"] == min_genre]
num_movies <- length(movies)

cat("genre: ", genre, "\nnum_movies: ", num_movies)
```

```
## genre:  Music
## num_movies:  51
```

```r
# Single SQL query
gg("
SELECT c.name AS genre, COUNT(f.film_id) AS num_movies
  FROM film_category AS f
 INNER JOIN category AS c ON f.category_id = c.category_id
 GROUP BY f.category_id
 ORDER BY num_movies
 LIMIT 1
")
```

```
##   genre num_movies
## 1 Music         51
```

2c

```r
# Extract tables using SQL queries
country <- gg("SELECT * FROM country")
city <- gg("SELECT * FROM city")
address <- gg("SELECT * FROM address")
customer <- gg("SELECT * FROM customer")

# Find cities that customers are from
# I used chatGPT to help me come up with the match() function
cities <- address$city_id[match(customer$address_id, address$address_id)]

# Find the countries of each city that customers are from
countries <- city$country_id[match(cities, city$city_id)]

# Create table with frequencies of how many customers are from each country
table <- table(country$country[match(countries, country$country_id)])

# Find countries where frequency = 13
table[table == 13]
```

```
##
## Argentina   Nigeria
##        13        13
```

```r
# Single SQL query
gg("
SELECT co.country
```

```
 FROM country AS co
INNER JOIN city AS ci ON co.country_id = ci.country_id
INNER JOIN address AS a ON a.city_id = ci.city_id
INNER JOIN customer AS cu ON cu.address_id = a.address_id
GROUP BY co.country_id
HAVING COUNT(cu.customer_id) = 13
")
```

```
##     country
## 1 Argentina
## 2   Nigeria
```

Problem 3 - US Records

```
# Load in records csv file
records <- read.csv("us-500.csv")
View(records)
```

3a

```
# Find number of emails that end in 'com'
num_com <- length(records$email[grepl("com$", records$email)])

# Divide by total number of records to find proportion
prop <- num_com / nrow(records)

cat('Proportion of emails with TLD ".com": ', prop)
```

```
## Proportion of emails with TLD ".com":  0.732
```

3b

```
# Split emails by @
email_split <- strsplit(records$email, "@")
email <- sapply(email_split, function(x) x[1])
address <- sapply(email_split, function(x) x[2])

# Split addresses by .
address_split <- strsplit(address, "\\.")
domain <- sapply(address_split, function(x) x[1])
tld <- sapply(address_split, function(x) x[2])

# Find emails with non-alphanumeric characters
non_email <- grepl("[^A-Za-z0-9]", email)
non_domain <- grepl("[^A-Za-z0-9]", domain)
non_tld <- grepl("[^A-Za-z0-9]", tld)
```

```r
# Find proportion of emails with non-alphanumeric characters
num_non_alphanumeric <- sum(non_email | non_domain | non_tld)
prop <- num_non_alphanumeric / nrow(records)

cat('Proportion of emails with non-alphanumeric characters: ', prop)
```

```
## Proportion of emails with non-alphanumeric characters:  0.506
```

3c

```r
# Extract area codes
phone1ac <- substr(records$phone1, 1, 3)
phone2ac <- substr(records$phone2, 1, 3)

# Find the 5 most common ones
most_common <- sort(table(c(phone1ac, phone2ac)), decreasing = TRUE)[1:5]

cat('The 5 most common area codes are: ', names(most_common))
```

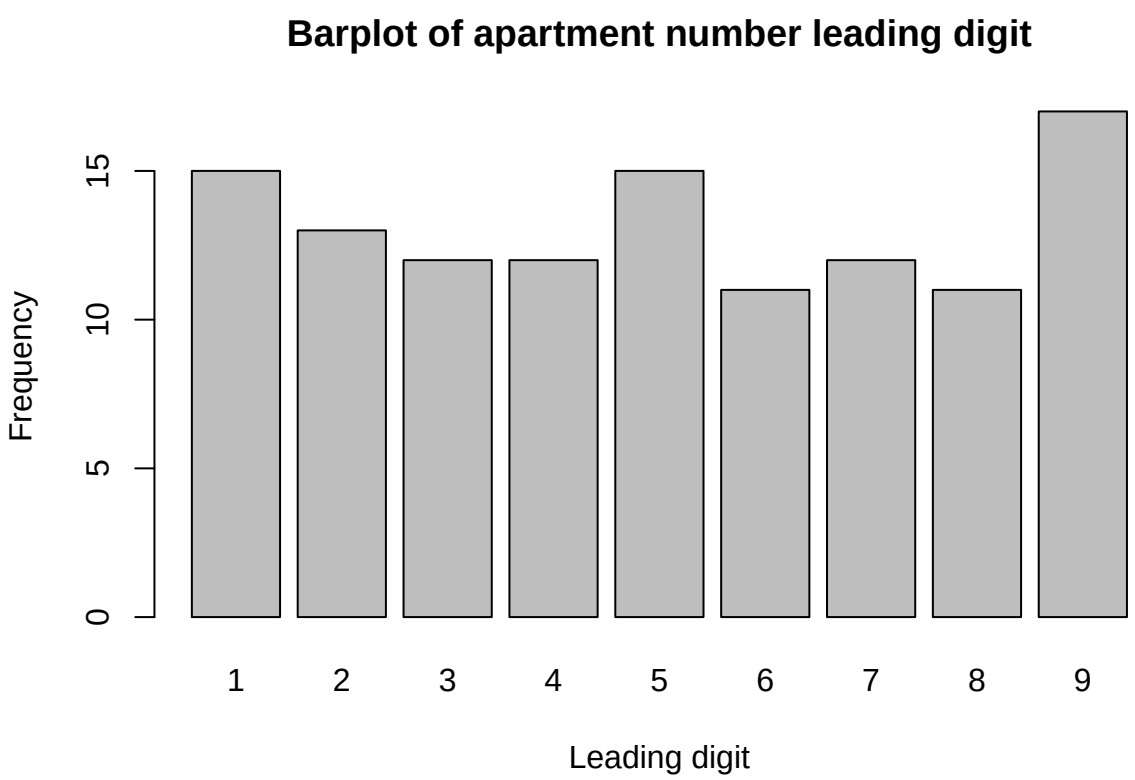
```
## The 5 most common area codes are:  973 212 215 410 201
```

3d

```r
# Find all addresses with at least one number at the end (apt number)
apartments <- records$address[grepl("[0-9]+$", records$address)]

# Extract apartment number
apartment_split <- strsplit(apartments, " ")
apt_numbers <- sapply(apartment_split, function(x) x[length(x)])
apt_numbers <- as.numeric(sub("#", "", apt_numbers))

# Graph histogram of the log of apartment numbers
hist(log(apt_numbers), main = "Histogram of the log of apartment numbers", xlab = "Log of apartment numb
```

3e

```
# Extract the first number of each apartment number
first_nums <- as.numeric(substr(apt_numbers, 1, 1))

# Plot histogram of first numbers
barplot(table(first_nums), main = "Barplot of apartment number leading digit", xlab = "Leading digit", y
```



This distribution looks closer to a Uniform distribution rather than Benford's law which states that the probability of a higher leading-numbers is smaller than lower ones (the distribution would be decreasing as the first number increases). Thus, I do not think that the apartment numbers would pass as real data.