problem 3

Problem 1 - Vision

This problem will require you to do things in Stata we have not covered. Use the Stata help, or online resources, to figure out the appropriate command(s). Use citation as necessary.

a. Download the file VIX_D from this location, and determine how to read it into Stata. Then download the file DEMO_D from this location. Note that each page contains a link to a documentation file for that data set. Merge the two files to create a single Stata dataset, using the **SEQN** variable for merging. Keep only records which matched. Print our your total sample size, showing that it is now 6,980.

```
. import sasxport5 "C:\Users\dxixi\Downloads\506-hw 3\VIX_D.XPT"
save "C:\Users\dxixi\Downloads\506-hw 3\VIX_D.dta"
file C:\Users\dxixi\Downloads\506-hw 3\VIX_D.dta saved


. import sasxport5 "C:\Users\dxixi\Downloads\506-hw 3\DEMO_D.XPT"
. save "C:\Users\dxixi\Downloads\506-hw 3\DEMO_D.dta"
file C:\Users\dxixi\Downloads\506-hw 3\DEMO_D.dta saved


. use "C:\Users\dxixi\Downloads\506-hw 3\VIX_D.dta", clear
merge 1:1 seqn using "C:\Users\dxixi\Downloads\506-hw 3\DEMO_D.dta", keep(match) noge

    Result                      Number of obs
    -----------------------------------------
    Not matched                             0
    Matched                             6,980
    -----------------------------------------

.
save "C:\Users\dxixi\Downloads\506-hw 3\VIX_D.dta", replace
```

```
file C:\Users\dxixi\Downloads\506-hw 3\VIX_D.dta saved
```

b. Without fitting any models, estimate the proportion of respondents within each 10-year age bracket (e.g. 0-9, 10-19, 20-29, etc) who wear glasses/contact lenses for distance vision. Produce a nice table with the results.

(Hint: One approach might be to try and find a way to produce this table with a single command. Another might be to estimate each proportion separately and then combine the results somehow. Yet another approach might be to manually do the calculations in Mata. Or any other approach that produces a single nice table.)

```
* This table will show the proportion of each category within each age bracket.

. capture drop age_bracket

. gen age_bracket = ""
(6,980 missing values generated)

. replace age_bracket = "0-9" if ridageyr >= 0 & ridageyr <= 9
(0 real changes made)

.
. replace age_bracket = "10-19" if ridageyr >= 10 & ridageyr <= 19
variable age_bracket was str1 now str5
(2,207 real changes made)

.
. replace age_bracket = "20-29" if ridageyr >= 20 & ridageyr <= 29
(1,021 real changes made)

.
. replace age_bracket = "30-39" if ridageyr >= 30 & ridageyr <= 39
(818 real changes made)

.
. replace age_bracket = "40-49" if ridageyr >= 40 & ridageyr <= 49
(815 real changes made)

.
. replace age_bracket = "50-59" if ridageyr >= 50 & ridageyr <= 59
(631 real changes made)
```

```
.
. replace age_bracket = "60-69" if ridageyr >= 60 & ridageyr <= 69
(661 real changes made)

.
. replace age_bracket = "70-79" if ridageyr >= 70 & ridageyr <= 79
(469 real changes made)

.
. replace age_bracket = "80+" if ridageyr >= 80
(358 real changes made)

. * Recode viq220 for clarity in the output table

.
. gen glasses_distance = ""
variable glasses_distance already defined
r(110);

.
. replace glasses_distance = "Yes" if viq220 == 1
(0 real changes made)

.
. replace glasses_distance = "No" if viq220 == 2
(0 real changes made)

.
. replace glasses_distance = "Don't know" if viq220 == 9
(0 real changes made)
.
.
. * Step 2: Tabulate the proportions

.
. tabulate age_bracket glasses_distance, col nofreq

age_bracke |         glasses_distance
         t | Don't k..         No        Yes |     Total
-----------+---------------------------------+----------
     10-19 |      0.00      37.51      24.23 |     31.89
     20-29 |    100.00      16.69      11.07 |     14.34
```

```
     30-39 |      0.00      12.72       9.73 |     11.46
     40-49 |      0.00      12.88      10.34 |     11.81
     50-59 |      0.00       7.25      12.12 |      9.30
     60-69 |      0.00       6.30      14.18 |      9.62
     70-79 |      0.00       3.92      10.81 |      6.83
       80+ |      0.00       2.72       7.52 |      4.75
-----------+--------------------------------+----------
     Total |    100.00     100.00     100.00 |    100.00

. * This table will show the proportion of each category within each age bracket.
```

c. Fit three logistic regression models predicting whether a respondent wears glasses/contact lenses for distance vision. Predictors:

   1. age
   2. age, race, gender
   3. age, race, gender, Poverty Income ratio

   Produce a table presenting the estimated odds ratios for the coefficients in each model, along with the sample size for the model, the pseudo-R2, and AIC values.

```
. . ssc install estout, replace
checking estout consistency and verifying not already installed...
all files already exist and are up to date.

. eststo clear


. * Model 1: age

. logit wears_glasses ridageyr

Iteration 0:  Log likelihood = -4686.4561
Iteration 1:  Log likelihood = -4482.1322
Iteration 2:  Log likelihood = -4481.7871
Iteration 3:  Log likelihood = -4481.7871

Logistic regression                                     Number of obs =  6,980
                                                        LR chi2(1)    = 409.34
                                                        Prob > chi2   = 0.0000
```

```
Log likelihood = -4481.7871                                   Pseudo R2     = 0.0437

------------------------------------------------------------------------------
wears_glasses | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
--------------+---------------------------------------------------------------
     ridageyr |   .0228955    .001157    19.79   0.000     .0206278    .0251631
        _cons |   -1.30923   .0521054   -25.13   0.000    -1.411355   -1.207106
------------------------------------------------------------------------------

.
. eststo M1

. * Model 2: age, race, gender

.
. logit wears_glasses ridageyr ridreth1 riagendr

Iteration 0:  Log likelihood = -4686.4561
Iteration 1:  Log likelihood = -4432.2417
Iteration 2:  Log likelihood = -4431.2822
Iteration 3:  Log likelihood = -4431.2821

Logistic regression                                     Number of obs =  6,980
                                                        LR chi2(3)    = 510.35
                                                        Prob > chi2   = 0.0000
Log likelihood = -4431.2821                             Pseudo R2     = 0.0544

------------------------------------------------------------------------------
wears_glasses | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
--------------+---------------------------------------------------------------
     ridageyr |   .0230428   .0011715    19.67   0.000     .0207468    .0253388
     ridreth1 |   .1117481   .0218705     5.11   0.000     .0688826    .1546135
     riagendr |   .4392872   .0511169     8.59   0.000     .3390998    .5394745
        _cons |  -2.305327   .1152477   -20.00   0.000    -2.531209   -2.079446
------------------------------------------------------------------------------

.
. eststo M2

.
. * Model 3: age, race, gender, Poverty Income Ratio
```

```
.
. logit wears_glasses ridageyr ridreth1 riagendr indfmpir

Iteration 0:  Log likelihood = -4467.3703
Iteration 1:  Log likelihood = -4192.5804
Iteration 2:  Log likelihood = -4191.1253
Iteration 3:  Log likelihood = -4191.1249

Logistic regression                                     Number of obs =  6,638
                                                        LR chi2(4)    = 552.49
                                                        Prob > chi2   = 0.0000
Log likelihood = -4191.1249                             Pseudo R2     = 0.0618

------------------------------------------------------------------------------
wears_glasses | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
--------------+---------------------------------------------------------------
     ridageyr |   .0219737   .0012135    18.11   0.000     .0195953    .0243521
     ridreth1 |   .0785997   .0229914     3.42   0.001     .0335374     .123662
     riagendr |   .4610131   .0526539     8.76   0.000     .3578133    .5642129
     indfmpir |    .150954   .0165745     9.11   0.000     .1184686    .1834394
        _cons |  -2.570646   .1246284   -20.63   0.000    -2.814913   -2.326379
------------------------------------------------------------------------------

.
. eststo M3


. * Display the results

.
. esttab M1 M2 M3, eform cells(b(star) se(par)) stats(N ll r2_p AIC, labels("Sample S
> del 1" "Model 2" "Model 3")

------------------------------------------------------------
                      (1)             (2)             (3)
                  Model 1         Model 2         Model 3
                     b/se            b/se            b/se
------------------------------------------------------------
wears_glas~s
ridageyr          1.02316***      1.02331***     1.022217***
                (.0011838)      (.0011988)      (.0012405)
ridreth1                          1.118231***     1.081771***
```

```
                                (.0244563)      (.0248714)
riagendr                          1.551601***      1.58568***
                                (.0793131)      (.0834923)
indfmpir                                          1.162943***
                                                (.0192752)
------------------------------------------------------------
Sample Size           6980            6980            6638
Log Likeli~d    -4481.787       -4431.282       -4191.125
Pseudo R2         .0436724        .0544492        .0618362
AIC
------------------------------------------------------------
Exponentiated coefficients
```

d. From the third model from the previous part, discuss whether the *odds* of men and women being wears of glasess/contact lenses for distance vision differs. Test whether the *proportion* of wearers of glasses/contact lenses for distance vision differs between men and women. Include the results of the test and its interpretation.

```
 **Testing the Proportion**:
   For this, you'd typically conduct a chi-square test for independence between gende

. * Chi-square test for independence

.
. tabulate riagendr wears_glasses, chi2

           |     wears_glasses
    Gender |         0          1 |     Total
-----------+----------------------+----------
         1 |     2,202      1,181 |     3,383
         2 |     2,013      1,584 |     3,597
-----------+----------------------+----------
     Total |     4,215      2,765 |     6,980

          Pearson chi2(1) =  60.7082   Pr = 0.000
```

The p-value is highly significant ($p < 0.05$), indicating that the proportion of men and women who wear glasses/contact lenses for distance vision is statistically different. Specifically, a larger proportion of women (1,584 out of 3,597) wear glasses/contact lenses compared to men (1,181 out of 3,383).

Based on these results, we can conclude that there is a significant association between

gender and the likelihood of wearing glasses/contact lenses for distance vision. The odds ratio from the logistic regression model (Model 3) will provide a measure of the strength and direction of this association (i.e., whether men or women are more likely to wear glasses/contact lenses). If the odds ratio for gender from Model 3 was significantly different from 1, this would further confirm the observed difference in proportions between men and women.

Problem 2 - Sakila

Load the "sakila" database discussed in class into SQLite. It can be downloaded from https://github.com/bradleygrant/sakila-sqlite3.

```
library(RSQLite)
# Import the SQLite database of the sakila data
sakila <- dbConnect(RSQLite::SQLite(), "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"
```

a. Aside from English, what language is most common for films? Answer this with a single SQL query.

```
dbGetQuery(sakila,"
SELECT
    l.name AS Language,
    COUNT(f.film_id) AS NumberOfFilms
FROM
    film f
JOIN
    language l ON f.language_id = l.language_id
WHERE
    l.name != 'English'
GROUP BY
    l.name
ORDER BY
```

```
    NumberOfFilms DESC
LIMIT 1;
")
```

```
[1] Language      NumberOfFilms
<0 rows> (or 0-length row.names)
```

1. All films in the database are in English.
2. There might be some inconsistencies or errors in the data or in the foreign key relationships.
3. The representation of 'English' in the **language** table might not exactly match the string 'English'.

For each of the following questions, solve them in two ways: First, use SQL query or queries to extract the appropriate table(s), then use regular R to answer the question. Second, use a single SQL query to answer the question.

b. What genre of movie is the most common in the data, and how many movies are of this genre?

```
sql_query1 <- "
SELECT
    category_id,
    COUNT(film_id) AS film_count
FROM
    film_category
GROUP BY
    category_id;
"
result <- dbGetQuery(sakila, sql_query1)

# Find the genre with the maximum count in R
most_common_genre <- result[which.max(result$film_count), ]
category_id_of_most_common <- most_common_genre$category_id

# To fetch the name of the category
category_name_query <- paste0("SELECT name FROM category WHERE category_id = ",
    category_id_of_most_common)
category_name <- dbGetQuery(sakila, category_name_query)$name

dbGetQuery(sakila, "
```

```
    SELECT
    c.name AS Genre,
    COUNT(fc.film_id) AS NumberOfFilms
FROM
    film_category fc
JOIN
    category c ON fc.category_id = c.category_id
GROUP BY
    c.name
ORDER BY
    NumberOfFilms DESC
LIMIT 1;
")
```

```
   Genre NumberOfFilms
1 Sports            74
```

c. Identify which country or countries have exactly 9 customers.

```
sql_query2 <- "
SELECT
    country_id,
    COUNT(customer_id) AS customer_count
FROM
    customer cu
JOIN
    address a ON cu.address_id = a.address_id
JOIN
    city ci ON a.city_id = ci.city_id
GROUP BY
    ci.country_id;

"
result <- dbGetQuery(sakila, sql_query2)

# Find countries with exactly 9 customers
countries_with_9_customers <- result[result$customer_count ==
    9, ]
country_ids <- countries_with_9_customers$country_id

# Fetch names of these countries
country_name_query <- paste("SELECT country FROM country WHERE country_id IN (",
```

```
    paste(country_ids, collapse = ","), ")")
country_names <- dbGetQuery(sakila, country_name_query)$country

dbGetQuery(sakila, "
    SELECT
    co.country
FROM
    customer cu
JOIN
    address a ON cu.address_id = a.address_id
JOIN
    city ci ON a.city_id = ci.city_id
JOIN
    country co ON ci.country_id = co.country_id
GROUP BY
    co.country
HAVING
    COUNT(cu.customer_id) = 9;

")
```

```
        country
1 United Kingdom
```

Problem 3 - US Records

Download the "US - 500 Records" data from https://www.briandunning.com/sample-data/ and import it into R. This is entirely fake data - use it to answer the following questions.

```
data <- read.csv("us-500.csv")
```

a. What proportion of email addresses are hosted at a domain with TLD ".net"? (E.g. in the email, "angrycat@freemail.org", "freemail.org" is the domain, with TLD (top-level domain) ".org".)

```
# Extract TLD from email addresses
tlds <- sub(".*\\.(\\w+)$", "\\1", data$email)

# Calculate the proportion of email addresses with ".net" TLD
proportion_net <- mean(tlds == "net")
```

```r
# Print the proportion
cat("Proportion of email addresses with '.net' TLD:", proportion_net, "\n")
```

```
Proportion of email addresses with '.net' TLD: 0.14
```

b. What proportion of email addresses have at least one non alphanumeric character in them? (Excluding the required "`@`" and "`.`" found in every email address.)

```r
# Function to check if an email has at least one
# non-alphanumeric character
has_non_alphanumeric <- function(email) {
    !grepl("^[A-Za-z0-9@.]*$", email)
}

# Count the number of emails with at least one
# non-alphanumeric character
count_non_alphanumeric <- sum(sapply(data$email, has_non_alphanumeric))

# Calculate the proportion
proportion_non_alphanumeric <- count_non_alphanumeric/length(data$email)

# Print the proportion
cat("Proportion of email addresses with at least one non-alphanumeric character:",
    proportion_non_alphanumeric, "\n")
```

```
Proportion of email addresses with at least one non-alphanumeric character: 0.248
```

c. What is the most common area code amongst all phone numbers?

```r
# Extract area codes from both columns and combine them
area_codes_phone1 <- substr(data$phone1, 1, 3)
area_codes_phone2 <- substr(data$phone2, 1, 3)
all_area_codes <- c(area_codes_phone1, area_codes_phone2)

# Count occurrences of each area code
area_code_counts <- table(all_area_codes)

# Identify the most common area code
most_common_area_code <- as.character(names(which.max(area_code_counts)))
most_common_area_code
```

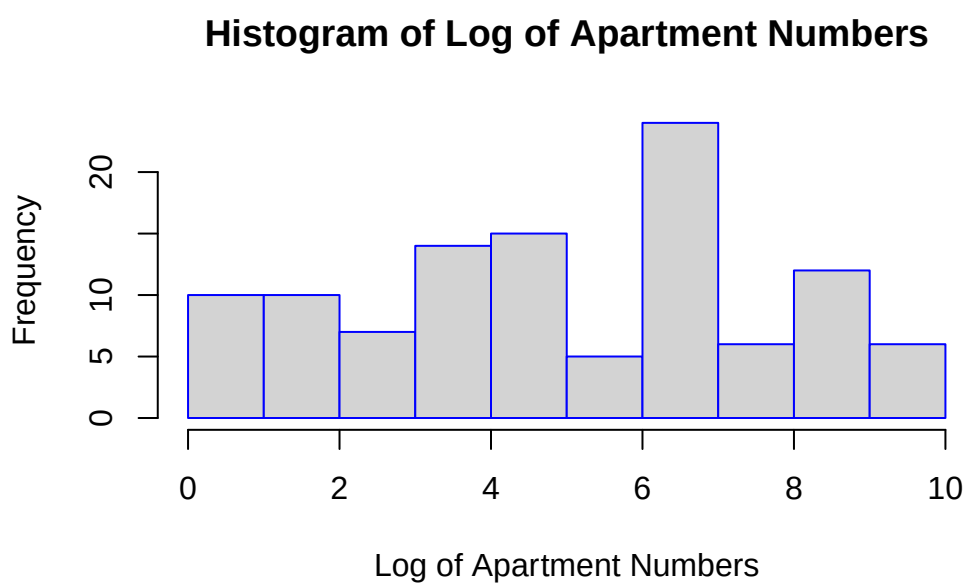
```
[1] "973"
```

d. Produce a histogram of the log of the apartment numbers for all addresses. (You may assume any number after the street is an apartment number.)

```r
# Extract apartment numbers using regex
data$apartment_numbers <- gsub(".*#", "", data$address)
data$apartment_numbers <- ifelse(grepl("^[0-9]+$", data$apartment_numbers),
    as.numeric(data$apartment_numbers), NA)
```

```
Warning in ifelse(grepl("^[0-9]+$", data$apartment_numbers),
as.numeric(data$apartment_numbers), : NAs introduced by coercion
```

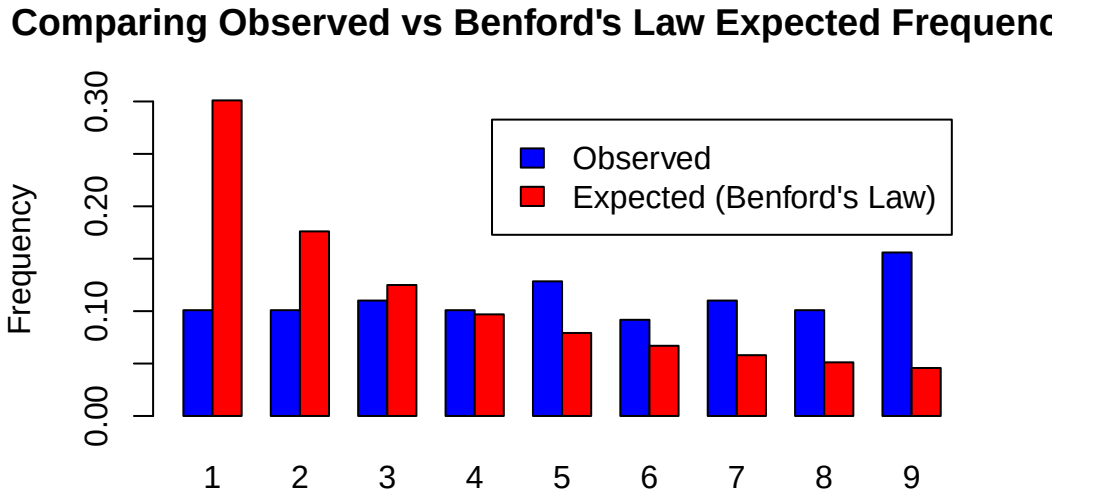
```r
# Removing NA values
apartment_numbers_clean <- na.omit(data$apartment_numbers)

# Compute log values
log_values <- log(apartment_numbers_clean)

# Plot a histogram
hist(log_values, main = "Histogram of Log of Apartment Numbers",
    xlab = "Log of Apartment Numbers", border = "blue", col = "lightgray")
```



e. Benford's law is an observation about the distribution of the leading digit of real numerical data. Examine whether the apartment numbers appear to follow Benford's law. Do

you think the apartment numbers would pass as real data?

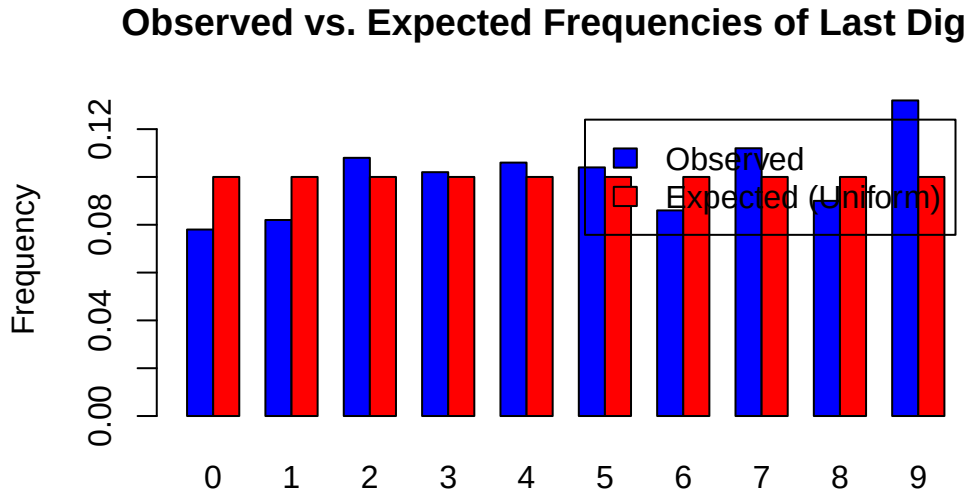
```
# Extract the leading digits
leading_digits <- as.numeric(substr(apartment_numbers_clean,
    1, 1))

# Compute observed frequency
# distribution
observed_frequencies <- table(leading_digits)/length(leading_digits)

# Benford's law expected
# frequencies
benford_frequencies <- log10(1 +
    1/(1:9))

# Plot the observed vs.
# expected frequencies
barplot(rbind(observed_frequencies,
    benford_frequencies), beside = TRUE,
    col = c("blue", "red"), names.arg = 1:9,
    legend.text = c("Observed",
        "Expected (Benford's Law)"),
    main = "Comparing Observed vs Benford's Law Expected Frequencies",
    ylab = "Frequency")
```



However, there's a noticeable deviation, it indicate that the numbers could be manufactured or artificially created, apartment numbers would not pass as real data

f. Repeat your analysis of Benford's law on the *last* digit of the street number. (E.g. if your address is "123 Main St #25", your street number is "123".)

```
# Extract street numbers using regex
street_numbers <- gsub(" .*", "", data$address)
street_numbers <- ifelse(grepl("^[0-9]+$", street_numbers), as.numeric(street_numbers),
    NA)

# Removing NA values
street_numbers_clean <- na.omit(street_numbers)

# Extract the last digits
last_digits <- as.numeric(substr(street_numbers_clean, nchar(street_numbers_clean),
    nchar(street_numbers_clean)))

# Compute observed frequency distribution
observed_frequencies <- table(last_digits)/length(last_digits)

# Expected uniform distribution frequencies for last digits
expected_frequencies <- rep(1/10, 10)  # 1/10 for each digit from 0 through 9

# Combining observed and expected frequencies for plotting
combined_frequencies <- rbind(as.numeric(observed_frequencies),
    expected_frequencies)

# Plotting side-by-side bar plot of observed vs. expected
# frequencies
barplot(combined_frequencies, beside = TRUE, names.arg = 0:9,
    col = c("blue", "red"), legend.text = c("Observed", "Expected (Uniform)"),
    main = "Observed vs. Expected Frequencies of Last Digits",
    ylab = "Frequency")
```



The distribution of the last digits of the street numbers aligns well with a uniform distribution, this would be consistent with what we'd expect for genuine real-world data.