STATS506-Problem Set 3

Shenyi Tang

2024-10-04

Shenyi Tang's GitHub Repo For STATS 506 FA 2024

https://github.com/shenyi-tang/stats506-computing-methods-and-tools.git

```
# for importing .xpt data
library("foreign")
library("dplyr")
library("emmeans")

# for creating nice tables
library("knitr")
library("kableExtra")

library("DBI")
library("ggplot2")
```

Problem 1 - Vision

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

```
# import two data sets
vix <- read.xport("vix_d.xpt")
demo <- read.xport("demo_d.xpt")

# merge 2 data sets using the key 'SEQN'
merge_d <- merge(vix, demo, by = "SEQN")

# sample size of the newly merged data
paste0("The sample size of the new data frame: ", dim(merge_d)[1])
```

```
[1] "The sample size of the new data frame: 6980"
```

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.

```
# create 'age_group' with 10-year age bracket
# the interval is closed on the right
merge_d$age_group <- cut(merge_d$RIDAGEYR, breaks = seq(0, 100, by = 10)
                ,right = TRUE
                ,labels = paste(seq(0,90,by=10),seq(9,99,by=10),sep='-'))

# group the date set by age group
# calculate the total number and NO.wearing glasses for distance vision in each group
# use the frequency to estimate the proportion with in each group
df_dv_gls <- merge_d %>%
  group_by(age_group) %>%
  summarize(total = n()
            ,wear_gls_dv = sum(VIQ220 == 1, na.rm = TRUE)
            ,prop = round(wear_gls_dv/total,2))

# create a nice table for the statistical result above
kable(df_dv_gls, format = "latex",
      col.names = c("Age Group", "Total", "# Wear Glasses for Distance Vision",
                    "Proption"),
      booktabs = TRUE,
      caption =
      "Proportion of Respondents Wearing Glasses for Distance Vision in Each Age
       ↪  Group",
      align = 'cccc'
      ) %>%
  kable_styling(latex_options = c("hold_position","striped"))
```

Table 1: Proportion of Respondents Wearing Glasses for Distance Vision in Each Age Group

Age Group	Total	# Wear Glasses for Distance Vision	Proption
10-19	2302	687	0.30
20-29	1019	321	0.32
30-39	828	268	0.32
40-49	791	291	0.37
50-59	632	342	0.54
60-69	646	391	0.61
70-79	446	287	0.64
80-89	316	178	0.56

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 ration**

Produce a table presenting the estimated odds ratios for the coefficients in each model, along with the sample size for the model, the pseudo-R^2, and AIC values.

```r
# drop data where viq220 equals to 9 and na
sub_merge_d <- merge_d %>% filter(VIQ220 == 1 | VIQ220 == 2)

# reassign value for VIQ220
sub_merge_d <- sub_merge_d %>%
  mutate(viq220 = ifelse(VIQ220 == 1, 1, ifelse(VIQ220 == 2, 0, VIQ220)))


# Logistic Regression wiz age
glm1 <- glm(viq220 ~ RIDAGEYR, data = sub_merge_d,
            family = binomial(link = "logit"))

# Logistic Regression wiz age, race and gender
glm2 <- glm(viq220 ~ RIDAGEYR + RIAGENDR + RIDRETH1, data = sub_merge_d,
            family = binomial(link = "logit"))

# Logistic Regression wiz age, race, gender, poverty income ratio
glm3 <- glm(viq220 ~ RIDAGEYR + RIAGENDR + RIDRETH1 + INDFMPIR,
            data = sub_merge_d, family = binomial(link = "logit"))

#' create data frame to summarize the information of glm model
#' @param model, the model from which to extract the information
#' @return data frame of the odds ratio, AIC, and pseudo R2
#'
model_info <- function(model) {
  data.frame(
    sample_size = nobs(model),
    odds_ratio = exp(coef(model)),
    pr2 = 1 - (model$deviance / model$null.deviance),
    aic = AIC(model)
  )
}

glm1.info <- model_info(glm1) %>%
  mutate(Model = "Model 1(Age)")
glm2.info <- model_info(glm2) %>%
  mutate(Model = "Model 2(Age, Race, Gender)")
glm3.info <- model_info(glm3) %>%
  mutate(Model = "Model 3(Age, Race, Gender, PIR)")

lm.info <- bind_rows(glm1.info, glm2.info, glm3.info) %>% select(Model, everything())

# display in a nice table
kable(lm.info, format = "latex", caption = "Information of 3 Logistic models") %>%
  kable_styling(latex_options = "striped", full_width = FALSE)
```

Table 2: Information of 3 Logistic models

	Model	sample_size	odds_ratio	pr2	aic
(Intercept)...1	Model 1(Age)	6545	0.2833790	0.0497312	8475.887
RIDAGEYR...2	Model 1(Age)	6545	1.0249798	0.0497312	8475.887
(Intercept)...3	Model 2(Age, Race, Gender)	6545	0.0918421	0.0633473	8358.496
RIDAGEYR...4	Model 2(Age, Race, Gender)	6545	1.0253239	0.0633473	8358.496
RIAGENDR...5	Model 2(Age, Race, Gender)	6545	1.6456280	0.0633473	8358.496
RIDRETH1...6	Model 2(Age, Race, Gender)	6545	1.1327501	0.0633473	8358.496
(Intercept)...7	Model 3(Age, Race, Gender, PIR)	6247	0.0717786	0.0690585	7940.790
RIDAGEYR...8	Model 3(Age, Race, Gender, PIR)	6247	1.0240472	0.0690585	7940.790
RIAGENDR...9	Model 3(Age, Race, Gender, PIR)	6247	1.6796667	0.0690585	7940.790
RIDRETH1...10	Model 3(Age, Race, Gender, PIR)	6247	1.0972155	0.0690585	7940.790
INDFMPIR	Model 3(Age, Race, Gender, PIR)	6247	1.1532697	0.0690585	7940.790

d. From the third model from the previous part, test 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 each test and their interpretation.

```
# odds test
odd_ratios_test <- anova(glm3)
odd_ratios_test_p <- odd_ratios_test$'Pr(>Chi)'[3]
paste0("The p-value of odds test: ", odd_ratios_test_p)
```

```
[1] "The p-value of odds test: 8.81080177466263e-21"
```

```
# proportion test
emm <- emmeans(glm3, ~ RIAGENDR)
gender_contrast <- contrast(emm, method = "pairwise", type = "response")
summary(gender_contrast)
```

```
 contrast              odds.ratio     SE  df null z.ratio p.value
 RIAGENDR1 / RIAGENDR2      0.595 0.0322 Inf    1  -9.582  <.0001

Tests are performed on the log odds ratio scale
```

- For the odds test, we could reject the null hypothesis as the p-value $< 5\%$, and conclude that there's significant difference in odds in wearing glasses between men and women.
- For the proportion test, we could reject the null hypothesis as the p-value < 0.0001, and conclude that there's significant difference in proportion in wearing glasses between men and women.

Problem 2 - Sakila

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

#' to simplify the sql query
```

```
#' @param query, the sql query sentence
#' @return the result of the sql query
rs <- function(query) {
  dbGetQuery(sakila,query)
}
```

a. What year is the oldest movie from, and how many movies were released in that year? Answer this with a single SQL query.

```
rs("
   select a.release_year
    ,count(a.title) as movie_cnt
   from (
    -- subquery to find out the earliest year
    select *
      , dense_rank() over(order by release_year) as rk
    from film
    ) as a
   where 1=1
    and a.rk = 1
   ")
```

```
  release_year movie_cnt
1         2006      1000
```

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

```
# R operations on data.frame

# extract table to data.frame
film <- rs("select * from film")
category <- rs("select * from category")
film_cat <- rs("select * from film_category")

#merge 3 tables
df2b <- merge(film, film_cat, by = "film_id", all.x = TRUE)
df2b <- merge(df2b, category, by = "category_id", all.x = TRUE)

# group by category name to count the number of films in each
# find the lease common genre and its film numbers
cate_cnt <- tapply(df2b$title, df2b$name, length)

genre_name <- names(cate_cnt)[which.min(cate_cnt)]
genre_no <- min(cate_cnt)

paste(genre_name,
      "is the lease common in data, there are",
```

```
      genre_no,
      "movies in this genre")
```

```
[1] "Music is the lease common in data, there are 51 movies in this genre"
```

```
# SQL Query
rs("
  select a.name
    , a.movie_cnt
  from (
    -- left join 3 tables
    select c.name
      , count(f.title) as movie_cnt -- cnt by group
    from film_category as fc
      left join film as f
        on fc.film_id = f.film_id
      left join category as c
        on fc.category_id = c.category_id
    group by c.name
  ) as a
  order by a.movie_cnt
  limit 1
   ")
```

```
   name movie_cnt
1 Music        51
```

c. Identify which country or countries have exactly 13 customers.

```
# R operations on data.frame

# extract tables to date.frames
customer <- rs("
                select customer_id
                  , address_id
                  , email
                from customer
               ")
address <- rs("
                select address_id
                  , city_id
                from address
              ")
city <- rs("
            select city_id
              , country_id
            from city
           ")
country <- rs("
```

```
                select country_id
                  ,country
                from country
               ")

# merge 4 data sets
df2c <- merge(customer, address, by = "address_id", all.x = TRUE)
df2c <- merge(df2c, city, by = "city_id", all.x = TRUE)
df2c <- merge(df2c, country, by = "country_id", all.x = TRUE)

# calculate the customer numbers from different countries
no_cust <- tapply(df2c$email, df2c$country, length)

# find countries having exactly 13 customers
wiz_13_cust_cntry <- names(no_cust)[no_cust == 13]
paste(
  "The countries with exactly 13 customers are:",
  paste(wiz_13_cust_cntry, collapse = " & ")
)
```

```
[1] "The countries with exactly 13 customers are: Argentina & Nigeria"
```

```
# SQL Query
rs("
   select ss.country
   from (
    select cry.country
      , count(c.customer_id) as cust_cnt
    from customer as c
      left join address as a
        on c.address_id = a.address_id
      left join city as ct
        on a.city_id = ct.city_id
      left join country as cry
        on ct.country_id = cry.country_id
    group by cry.country
   ) as ss
   where 1=1
    and ss.cust_cnt = 13
   ")
```

```
    country
1 Argentina
2   Nigeria
```

```
dbDisconnect(sakila)
```

Problem 3 - US Records

```
us <- read.csv("us-500.csv", sep = ",")
```

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

```
# extract data with a ".com" TLD email
df3a <- us[grepl("\\.com", us$email), ]

# calculate the proportion
prop <- dim(df3a)[1] / dim(us)[1]

paste0("The proportion of email addresses hosted at a \".com\" TLD is: ", prop)
```

```
[1] "The proportion of email addresses hosted at a \".com\" TLD is: 0.732"
```

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

```
# extract the part before "@" of each email
df3b <- as.data.frame(sub("@.*$", "", us$email))
colnames(df3b) <- "eadd"

# non-alphanumeric: character excluding a-z A-Z 0-9
df3b2 <- df3b[grep("[^a-zA-Z0-9]", df3b$eadd), ]


# calculate the proportion
prop3b <- length(df3b2) / dim(us)[1]

paste0(
  "The proportion of email addresses having at least one non-alphanumeric character in
   ↪  them is: ", prop3b
)
```

```
[1] "The proportion of email addresses having at least one non-alphanumeric character
 ↪  in them is: 0.506"
```

c. **What are the top 5 most common area codes amongst all phone numbers? (The area code is the first three digits of a standard 10-digit telephone number.)**

```
# split the area code from phone 1
us$area_code1 <- substr(us$phone1, 1, 3)
us$area_code2 <- substr(us$phone2, 1, 3)
```

```r
# count by the area code
df3c <- table(cbind(us$area_code1, us$area_code2))
# sort to filter the top 5 most common area codes
df3c <- as.data.frame(sort(df3c, decreasing = TRUE)[1:5])
names(df3c) <- c("area_code", "cnt")

paste("The top 5 common area codes:", paste(df3c$area_code, collapse = ","))
```

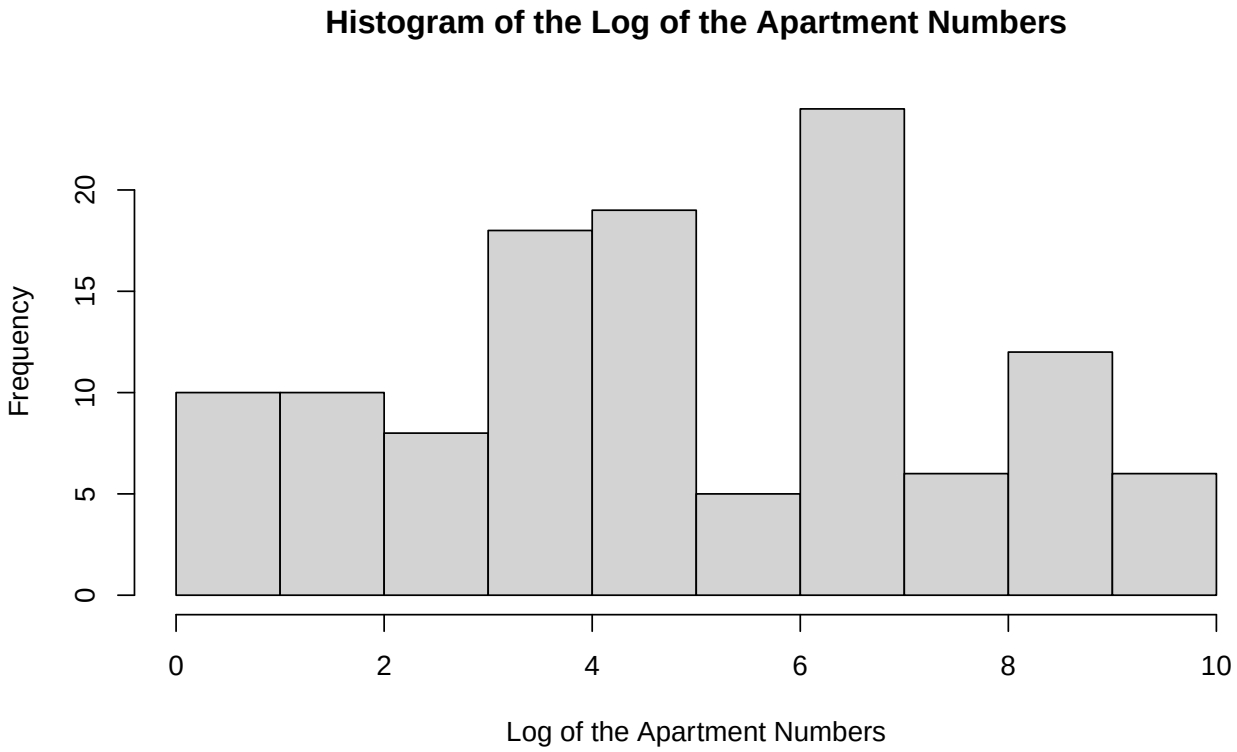
```
[1] "The top 5 common area codes: 973,212,215,410,201"
```

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

```r
# extract address with a apartment number (end with number)
df3d <- as.data.frame(us$address[grepl(".*\\d$", us$address)])
colnames(df3d) <- "naprt"



# extract the apartment number
# apartment number is the end of an address
# substitute the apartment number of apartment address
# using \\1 to catch the content in the previous bracket
df3d$aprt <- as.numeric(sub(".*?(\\d+)$", "\\1", df3d$naprt))
df3d$log_aprt <- log(df3d$aprt)
hist(df3d$log_aprt,
     main = "Histogram of the Log of the Apartment Numbers",
     xlab = "Log of the Apartment Numbers",
     ylab = "Frequency")
```



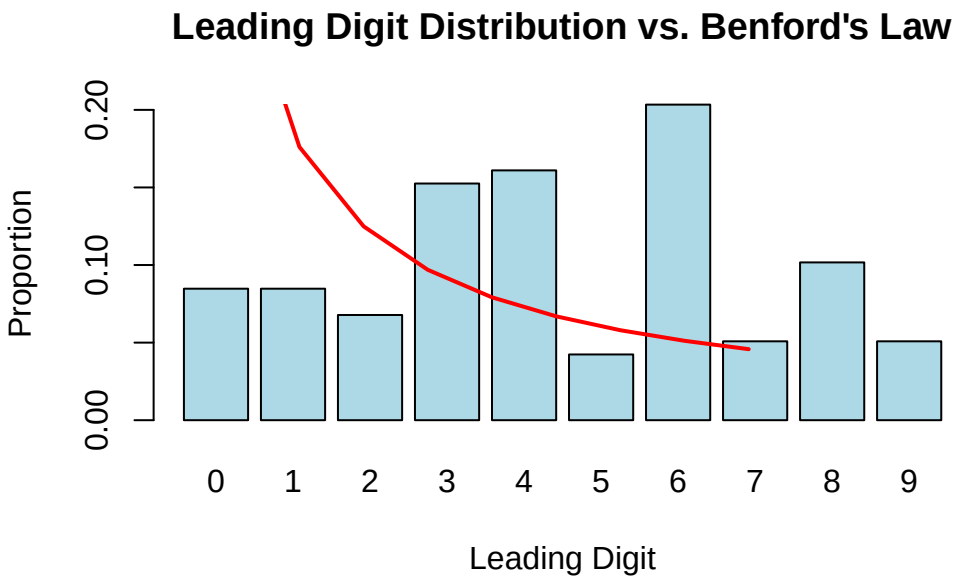
e. Benford's law is an observation about the distribution of the leading digit of real numerical data. Examine whether the apartment numbers appear to follow Benford's law. Do you think the apartment numbers would pass as real data?

```
# extract the first digit of the apartment number
df3d$aprt_1 <- as.numeric(substr(df3d$log_aprt, 1, 1))

# calculate the frequency
df3e <- as.data.frame(table(df3d$aprt_1))
names(df3e) <- c("1st digit", "cnt")

# Ben Ford's law
bf <- log10(1 + 1/(1:9))

barplot(df3e$cnt / sum(df3e$cnt), names.arg = df3e$`1st digit`,
        main = "Leading Digit Distribution vs. Benford's Law",
        xlab = "Leading Digit", ylab = "Proportion", col = "lightblue")
lines(1:9, bf, col = "red", lwd = 2)
```



- According to the plot, I don't think the apartment numbers would pass as real data