stats506-ps3-JiabaoLi

Jiabao Li

git:

lijiabao203/stats506_rwork (github.com)

Problem 1 - Vision

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

For the "nice tables", use a function such as kable from knitr, or the stargazer package (or find another approach) to generate HTML/LaTeX tables for inclusion. The results should be clearly labeled, rounded appropriately, and easily readable.

Import first:

```r
library(knitr)
```

a. Download the file VIX_D from this location, and determine how to read it into R. 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 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.

Because it is not a usual format, we need to use the foregin package and use the function: 'read.xport()'. Here is the code:

```r
library(foreign)
vixd = read.xport('VIX_D.XPT')
demod = read.xport('DEMO_D.XPT')
```

And here is the process to merge these files, use the argument 'by' of the function merge to use the SEQN variable.

```r
df_vd = merge(vixd, demod, by = 'SEQN')
# this function return a data frame, so no need to create again
nrow(df_vd)
```

```
[1] 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.

First, from the document of the data set, VIQ220 is “Glasses/contact lenses worn for distance”. And it has:

```r
df_tb = data.frame(
  name = c('value', 'means', 'count'),
  val_1 = c(1, 'Yes', 2765),
  val_2 = c(2, 'No', 3780),
  val_3 = c(9, 'Don\'t know', 2),
  val_4 = c(NA, 'Missing', 433)
)
kable(df_tb, caption = 'Meaning of VIQ220')
```

Table 1: Meaning of VIQ220

name	val_1	val_2	val_3	val_4
value	1	2	9	NA
means	Yes	No	Don’t know	Missing
count	2765	3780	2	433

And RIDAGEYR is “Age at Screening Adjudicated - Recode”, which has:

```r
df_tb = data.frame(
  name = c('value', 'means', 'count'),
  val_1 = c('0 to 84', 'Range of Values', 10178),
  val_2 = c(85, 'bigger than 85', 170),
  val_3 = c(NA, 'Missing', 0)
)
kable(df_tb, caption = 'Meaning of RIDAGEYR')
```

Table 2: Meaning of RIDAGEYR

name	val_1	val_2	val_3
value	0 to 84	85	NA
means	Range of Values	bigger than 85	Missing
count	10178	170	0

Because NA and 9 in VIQ220 has no contribution to our analysis. we use the percent to show wear glasses/contact lenses for distance vision or not. Now, it's time to get the summary of each parts:

```r
# use python to generate these code because it's too long
# from table(df_vd$RIDAGEYR), we should start from age of 12
# 10-20 to 80-90 (any people with age bigger than 85 are in this group)
df_12 = df_vd[which(df_vd$RIDAGEYR > 10 & df_vd$RIDAGEYR < 21), ]$VIQ220
df_23 = df_vd[which(df_vd$RIDAGEYR > 20 & df_vd$RIDAGEYR < 31), ]$VIQ220
df_34 = df_vd[which(df_vd$RIDAGEYR > 30 & df_vd$RIDAGEYR < 41), ]$VIQ220
df_45 = df_vd[which(df_vd$RIDAGEYR > 40 & df_vd$RIDAGEYR < 51), ]$VIQ220
df_56 = df_vd[which(df_vd$RIDAGEYR > 50 & df_vd$RIDAGEYR < 61), ]$VIQ220
df_67 = df_vd[which(df_vd$RIDAGEYR > 60 & df_vd$RIDAGEYR < 71), ]$VIQ220
df_78 = df_vd[which(df_vd$RIDAGEYR > 70 & df_vd$RIDAGEYR < 81), ]$VIQ220
df_89 = df_vd[which(df_vd$RIDAGEYR > 80 & df_vd$RIDAGEYR < 91), ]$VIQ220

df_tb = data.frame(
  "Age Group" = c('10-20', '21-30', '31-40', '41-50', '51-60',
                  '61-70', '71-80', '81-90'),
  "percent of wear" = c(
    paste(round(table(df_12)[1]/(table(df_12)[1]+table(df_12)[2])*100, 2), '%'),
    paste(round(table(df_23)[1]/(table(df_23)[1]+table(df_23)[2])*100, 2), '%'),
    paste(round(table(df_34)[1]/(table(df_34)[1]+table(df_34)[2])*100, 2), '%'),
    paste(round(table(df_45)[1]/(table(df_45)[1]+table(df_45)[2])*100, 2), '%'),
    paste(round(table(df_56)[1]/(table(df_56)[1]+table(df_56)[2])*100, 2), '%'),
    paste(round(table(df_67)[1]/(table(df_67)[1]+table(df_67)[2])*100, 2), '%'),
    paste(round(table(df_78)[1]/(table(df_78)[1]+table(df_78)[2])*100, 2), '%'),
    paste(round(table(df_89)[1]/(table(df_89)[1]+table(df_89)[2])*100, 2), '%')),
  "percent of not wear" = c(
    paste(round(table(df_12)[2]/(table(df_12)[1]+table(df_12)[2])*100, 2), '%'),
    paste(round(table(df_23)[2]/(table(df_23)[1]+table(df_23)[2])*100, 2), '%'),
    paste(round(table(df_34)[2]/(table(df_34)[1]+table(df_34)[2])*100, 2), '%'),
    paste(round(table(df_45)[2]/(table(df_45)[1]+table(df_45)[2])*100, 2), '%'),
    paste(round(table(df_56)[2]/(table(df_56)[1]+table(df_56)[2])*100, 2), '%'),
    paste(round(table(df_67)[2]/(table(df_67)[1]+table(df_67)[2])*100, 2), '%'),
    paste(round(table(df_78)[2]/(table(df_78)[1]+table(df_78)[2])*100, 2), '%'),
    paste(round(table(df_89)[2]/(table(df_89)[1]+table(df_89)[2])*100, 2), '%'))

)
kable(df_tb, caption = 'Portion of wear or not in different age range')
```

Table 3: Portion of wear or not in different age range

Age.Group	percent.of.wear	percent.of.not.wear
10-20	31.66 %	68.34 %
21-30	34.11 %	65.89 %
31-40	35.03 %	64.97 %
41-50	38.9 %	61.1 %
51-60	56.25 %	43.75 %
61-70	63.37 %	36.63 %
71-80	67.69 %	32.31 %
81-90	65.44 %	34.56 %

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 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-R^2, and AIC values.

For each model, construct an odds ratios table first:

```
model1 = glm(VIQ220 ~ RIDAGEYR, data = df_vd)
summary_or_table1 = data.frame(
  "estimated coefficients" = c("(Intercept)", "RIDAGEYR"),
  "odds ratios" = setNames(exp(model1$coefficients), NULL)
)
kable(summary_or_table1,
      caption = "Estimated Odds Ratios for the Coefficients in Model1")
```

Table 4: Estimated Odds Ratios for the Coefficients in Model1

estimated.coefficients	odds.ratios
(Intercept)	6.0710200
RIDAGEYR	0.9941005

```r
model2 = glm(VIQ220 ~ RIDAGEYR + RIDRETH1 + RIAGENDR, data = df_vd)
summary_or_table2 = data.frame(
  "estimated coefficients" = c("(Intercept)", "RIDAGEYR", "RIDRETH1", "RIAGENDR"),
  "odds ratios" = setNames(exp(model2$coefficients), NULL)
)
kable(summary_or_table2,
      caption = "Estimated Odds Ratios for the Coefficients in Model2")
```

Table 5: Estimated Odds Ratios for the Coefficients in Model2

estimated.coefficients	odds.ratios
(Intercept)	7.7332483
RIDAGEYR	0.9941465
RIDRETH1	0.9716107
RIAGENDR	0.8983367

```r
model3 = glm(VIQ220 ~ RIDAGEYR + RIDRETH1 + RIAGENDR + INDFMPIR, data = df_vd)
summary_or_table3 = data.frame(
  "estimated coefficients" = c("(Intercept)", "RIDAGEYR",
                               "RIDRETH1", "RIAGENDR", "INDFMPIR"),
  "odds ratios" = setNames(exp(model3$coefficients), NULL)
)
kable(summary_or_table3,
      caption = "Estimated Odds Ratios for the Coefficients in Model3")
```

Table 6: Estimated Odds Ratios for the Coefficients in Model3

estimated.coefficients	odds.ratios
(Intercept)	8.1302270
RIDAGEYR	0.9944587
RIDRETH1	0.9784215
RIAGENDR	0.8950932
INDFMPIR	0.9692335

estimated.coefficients	odds.ratios

And construct a table include the sample size for the models, the pseudo-R^2, and AIC values:

```r
loglik_null = logLik(glm(VIQ220 ~ 1, data = df_vd))
summary_table = data.frame(
  Model = c("Age", "Age + Race + Gender",
            "Age + Race + Gender + Poverty Income Ratio"),
  SampleSize = c(nrow(model1$model), nrow(model2$model), nrow(model3$model)),
  PseudoR2 = c(1-(logLik(model1)/loglik_null), 1-(logLik(model2)/loglik_null),
               1-(logLik(model3)/loglik_null)),
  AIC = c(model1$aic, model2$aic, model3$aic)
)
kable(summary_table, caption = "Logistic Regression Model Summary")
```

Table 7: Logistic Regression Model Summary

Model	SampleSize	PseudoR2	AIC
Age	6547	0.0441621	9352.988
Age + Race + Gender	6547	0.0552942	9248.129
Age + Race + Gender + Poverty Income Ratio	6249	0.0994401	8818.434

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.

The gender only has values 1 and 2, which means male and female. Since we don't consider NA values and 9 values which has no attribution to our analysis, the table is:

```r
df_m = df_vd[which(df_vd$RIAGENDR == 1), ]$VIQ220
df_f = df_vd[which(df_vd$RIAGENDR == 2), ]$VIQ220
model2 = glm(VIQ220 ~ RIDAGEYR + RIDRETH1 + RIAGENDR, data = df_vd)
summary_or_table2 = data.frame(
  "Gender" = c("Male", "Female"),
  "percent of wear" = c(paste(round(table(df_m)[1]/(table(df_m)[1]+
    table(df_m)[2])*100, 2), '%'), paste(round(table(df_f)[1]/
    (table(df_f)[1]+table(df_f)[2])*100, 2), '%')),
  "percent of not wear" = c(paste(round(table(df_m)[2]/(table(df_m)[1]+
    table(df_m)[2])*100, 2), '%'), paste(round(table(df_f)[2]/
    (table(df_f)[1]+table(df_f)[2])*100, 2), '%'))
)
kable(summary_or_table2,
      caption = "Percent of Wear or Not in Different Gender")
```

Table 8: Percent of Wear or Not in Different Gender

Gender	percent.of.wear	percent.of.not.wear
Male	36.96 %	63.04 %
Female	47.28 %	52.72 %

Problem 2 - Sakila

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

For these problems, do not use any of the tables whose names end in _list.

Read db:

```r
library(DBI)
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"
```

```r
# use this function to simplify the dbquery
qe <- function(stri){
  return(dbGetQuery(sakila, stri))
}
```

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

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 operations on those data.frames to answer the question. Second, use a single SQL query to answer the question.

SQL:

```
qe("select count(film_id) as film_number from film
   group by release_year
   having release_year = (
    select release_year from film
    order by release_year
    limit 1
   )")
```

```
  film_number
1        1000
```

R:

```
# extract table
table_film = qe("select * from film")
nrow(table_film[which(table_film$release_year == min(table_film$release_year)), ])
```

```
[1] 1000
```

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

SQL:

```
qe("select count(film_id) as film_number, name as genre_name
   from film_category left join category on
    film_category.category_id = category.category_id
   group by film_category.category_id
   having film_category.category_id = (
    select category_id from film_category
    group by category_id
    order by count(film_id)
    limit 1
   )")
```

```
  film_number genre_name
1          51      Music
```

R: warning does not affect the answer.

```
# extract table
table_filc = qe("select * from film_category")
table_ca = qe("select* from category")
# find the least common category id
counts = table(table_filc$category_id)
lc_cid = names(counts)[which.min(counts)]
movies_number = min(counts)
genre_name = setNames(table_ca[which(table_ca$category_id == lc_cid),]["name"],
                      NULL)
c("genre name"=genre_name, "movies number"=movies_number)
```

```
$`genre name`
[1] "Music"

$`movies number`
[1] 51
```

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

SQL:

```
qe("select country from country
   where country_id in (
    select country.country_id from customer left join address on
      customer.address_id = address.address_id
      left join city on city.city_id = address.city_id
      left join country on city.country_id = country.country_id
    group by country.country_id
    having count(customer.customer_id) == 13
   )")
```

```
    country
1 Argentina
2   Nigeria
```

R:

```
# extract table
table_cust = qe("select * from customer")
table_addr = qe("select * from address")
table_city = qe("select * from city")
table_coun = qe("select * from country")
# construct a merge table like in sql
table_cio = merge(table_city, table_coun, by = "country_id")
table_aio = merge(table_addr, table_cio, by = "city_id")
table_all = merge(table_cust, table_aio, by="address_id")
```

```
Warning in merge.data.frame(table_cust, table_aio, by = "address_id"): column
names 'last_update.x', 'last_update.y' are duplicated in the result
```

```
# extract countries with exactly 13 coustmers
counts = table(table_all$country_id)
cid = names(counts)[which(counts==13)]
coun_name = setNames(table_coun[which(table_coun$country_id %in% cid),]["country"],
                     NULL)
coun_name
```

```
6   Argentina
69   Nigeria
```

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.

```
usr = read.csv("us-500.csv")
```

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).)

```r
email = usr$email
# select those end with".org"
email_org = email[grep("\\.org$", email)]
portion = length(email_org)/length(email)
portion
```

```
[1] 0.128
```

From the output, only 12.8% of all email address have a TLD.

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
# That is to find email address with at least 3 non alphanumeric
emails3 = email[grep("[^a-zA-Z0-9].*[^a-zA-Z0-9].*[^a-zA-Z0-9]", email)]
portion = length(emails3)/length(email)
portion
```

```
[1] 0.506
```

The proportion is 50.6%. Or 50.6% of email addresses have at least one non alphanumeric character in them. (Excluding the required “@” and “.” found in every email address.)

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.)

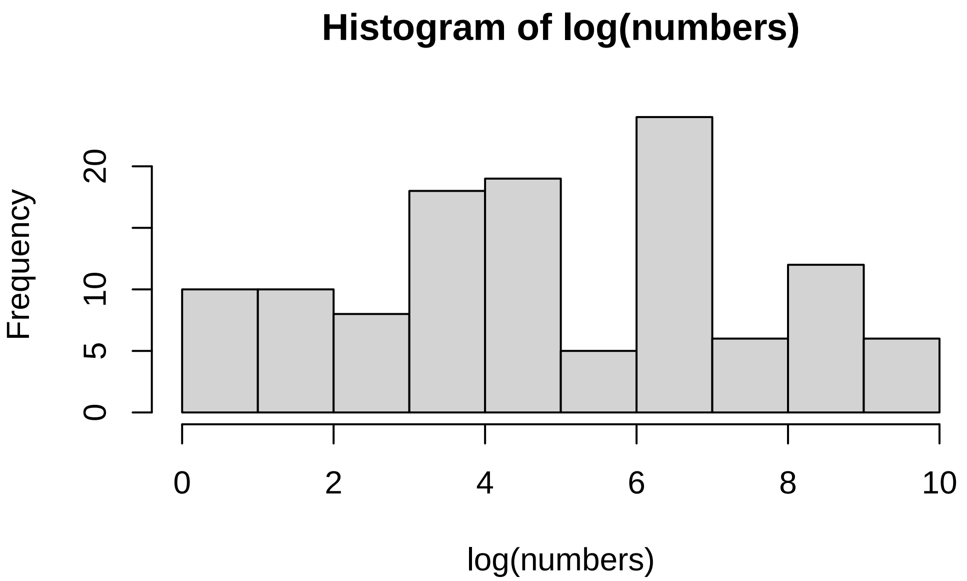
These codes are: 973, 212, 215, 410, 201.

```r
phone = rbind(usr$phone1, usr$phone2)
phone_first3 = substr(phone,0,3)
teb = table(phone_first3)
names(teb[order(-teb)][1:5])
```

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

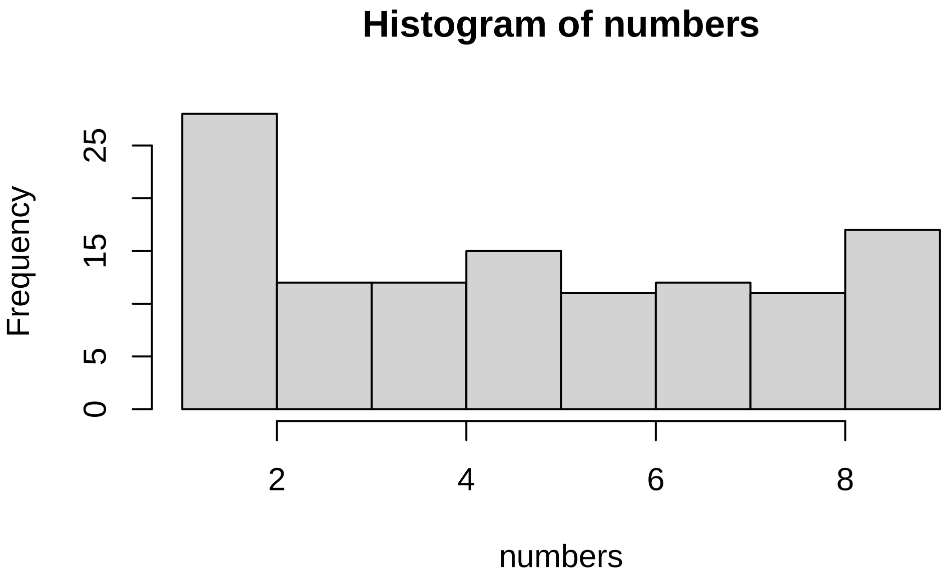
d. Produce a histogram of the log of the apartment numbers for all addresses. (You may assume any number at the end of the an address is an apartment number.)

```
address = usr$address
# extract numbers in the end
st_n = regmatches(address, regexec("\\d+$", address))
numbers = as.numeric(st_n)
hist(log(numbers))
```



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 number here
st_n_first = regmatches(st_n, regexec("^\\d", st_n))
numbers = as.numeric(st_n_first)
hist(numbers)
```



Overall, there are more numbers smaller. Or the smaller side is “heavier” that the bigger side.

So the apartment numbers pass Benford’s law as real data.