506 Problem Set 4

Xiaohan Liu

Table of contents

Problem 1 2
- (a) 2
- (b) 6

Problem 2 6

Problem 3 8
- (a) 9
- (b) 10
- (c) 10
- (d) 10
- (e) 11

Problem 4 11
- (a) 11
- (b) 12
- (c) 13
- (d) 14
- (e) 15
- (f) 15
- (g) 17
- (h) 17

GitHub repository: https://github.com/EmiiilyLiu/STATS_506

```r
setwd("F:/Desktop/STATS 506/STATS_506")
```

Problem 1

```r
## Install and load package
## install.packages("nycflights13")
#install.packages("tidyverse")

library(nycflights13)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.3     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
#data(package = "nycflights13")
```

(a)

```r
## Departure table
departure_delay <- flights %>%
  group_by(origin) %>%
  summarise(
    mean_delay = mean(dep_delay, na.rm = TRUE),
    median_delay = median(dep_delay, na.rm = TRUE)
  ) %>%
  arrange(-mean_delay) %>%  ## Order in descending mean delay
  ## use the airport names not the airport codes
  left_join(airports, by = c("origin" = "faa")) %>%
  select(name, mean_delay, median_delay) %>%
  print(n = Inf)  ## print all rows
```

```
# A tibble: 3 x 3
```

```
  name                mean_delay median_delay
  <chr>                    <dbl>        <dbl>
1 Newark Liberty Intl       15.1           -1
2 John F Kennedy Intl       12.1           -1
3 La Guardia                10.3           -3
```

```r
## Arrival table
arrival_delay <- flights %>%
  group_by(dest) %>%
  summarise(
    total_flights = n(),
    mean_delay = mean(arr_delay, na.rm = TRUE),
    median_delay = median(arr_delay, na.rm = TRUE)
  ) %>%
  filter(total_flights >= 10) %>%
  arrange(-mean_delay) %>%  ## Order in descending mean delay
  ## use the airport names not the airport codes
  left_join(airports, by = c("dest" = "faa")) %>%
  select(name, mean_delay, median_delay) %>%
  ## several airports exist in table "flights" with more than 10 records
  ## but not in table "airports", ignore them
  filter(!is.na(name)) %>%
  print(n = Inf) ## print all rows
```

```
# A tibble: 98 x 3
   name                                   mean_delay median_delay
   <chr>                                       <dbl>        <dbl>
 1 "Columbia Metropolitan"                       41.8            28
 2 "Tulsa Intl"                                  33.7            14
 3 "Will Rogers World"                           30.6            16
 4 "Jackson Hole Airport"                        28.1            15
 5 "Mc Ghee Tyson"                               24.1             2
 6 "Dane Co Rgnl Truax Fld"                      20.2             1
 7 "Richmond Intl"                               20.1             1
 8 "Akron Canton Regional Airport"               19.7             3
 9 "Des Moines Intl"                             19.0             0
10 "Gerald R Ford Intl"                          18.2             1
11 "Birmingham Intl"                             16.9            -2
12 "Theodore Francis Green State"                16.2             1
13 "Greenville-Spartanburg International"        15.9            -0.5
14 "Cincinnati Northern Kentucky Intl"           15.4            -3
15 "Savannah Hilton Head Intl"                   15.1            -1
```

```
16 "Manchester Regional Airport"             14.8                    -3
17 "Eppley Afld"                             14.7                    -2
18 "Yeager"                                  14.7                    -1.5
19 "Kansas City Intl"                        14.5                     0
20 "Albany Intl"                             14.4                    -4
21 "General Mitchell Intl"                   14.2                     0
22 "Piedmont Triad"                          14.1                    -2
23 "Washington Dulles Intl"                  13.9                    -3
24 "Cherry Capital Airport"                  13.0                   -10
25 "James M Cox Dayton Intl"                 12.7                    -3
26 "Louisville International Airport"        12.7                    -2
27 "Chicago Midway Intl"                     12.4                    -1
28 "Sacramento Intl"                         12.1                     4
29 "Jacksonville Intl"                       11.8                    -2
30 "Nashville Intl"                          11.8                    -2
31 "Portland Intl Jetport"                   11.7                    -4
32 "Greater Rochester Intl"                  11.6                    -5
33 "Hartsfield Jackson Atlanta Intl"         11.3                    -1
34 "Lambert St Louis Intl"                   11.1                    -3
35 "Norfolk Intl"                            10.9                    -4
36 "Baltimore Washington Intl"               10.7                    -5
37 "Memphis Intl"                            10.6                    -2.5
38 "Port Columbus Intl"                      10.6                    -3
39 "Charleston Afb Intl"                     10.6                    -4
40 "Philadelphia Intl"                       10.1                    -3
41 "Raleigh Durham Intl"                     10.1                    -3
42 "Indianapolis Intl"                        9.94                   -3
43 "Charlottesville-Albemarle"                9.5                    -5
44 "Cleveland Hopkins Intl"                   9.18                   -5
45 "Ronald Reagan Washington Natl"            9.07                   -2
46 "Burlington Intl"                          8.95                   -4
47 "Buffalo Niagara Intl"                     8.95                   -5
48 "Syracuse Hancock Intl"                    8.90                   -5
49 "Denver Intl"                              8.61                   -2
50 "Palm Beach Intl"                          8.56                   -3
51 "Bob Hope"                                 8.18                   -3
52 "Fort Lauderdale Hollywood Intl"           8.08                   -3
53 "Bangor Intl"                              8.03                   -9
54 "Asheville Regional Airport"               8.00                   -1
55 "Pittsburgh Intl"                          7.68                   -5
56 "Gallatin Field"                           7.6                    -2
57 "NW Arkansas Regional"                     7.47                   -2
58 "Tampa Intl"                               7.41                   -4
```

```
59 "Charlotte Douglas Intl"                   7.36          -3
60 "Minneapolis St Paul Intl"                 7.27          -5
61 "William P Hobby"                          7.18          -4
62 "Bradley Intl"                             7.05         -10
63 "San Antonio Intl"                         6.95          -9
64 "South Bend Rgnl"                          6.5           -3.5
65 "Louis Armstrong New Orleans Intl"         6.49          -6
66 "Key West Intl"                            6.35           7
67 "Eagle Co Rgnl"                            6.30          -4
68 "Austin Bergstrom Intl"                    6.02          -5
69 "Chicago Ohare Intl"                       5.88          -8
70 "Orlando Intl"                             5.45          -5
71 "Detroit Metro Wayne Co"                   5.43          -7
72 "Portland Intl"                            5.14          -5
73 "Nantucket Mem"                            4.85          -3
74 "Wilmington Intl"                          4.64          -7
75 "Myrtle Beach Intl"                        4.60         -13
76 "Albuquerque International Sunport"        4.38          -5.5
77 "George Bush Intercontinental"             4.24          -5
78 "Norman Y Mineta San Jose Intl"            3.45          -7
79 "Southwest Florida Intl"                   3.24          -5
80 "San Diego Intl"                           3.14          -5
81 "Sarasota Bradenton Intl"                  3.08          -5
82 "Metropolitan Oakland Intl"                3.08          -9
83 "General Edward Lawrence Logan Intl"       2.91          -9
84 "San Francisco Intl"                       2.67          -8
85 "Yampa Valley"                             2.14           2
86 "Phoenix Sky Harbor Intl"                  2.10          -6
87 "Montrose Regional Airport"                1.79         -10.5
88 "Los Angeles Intl"                         0.547         -7
89 "Dallas Fort Worth Intl"                   0.322         -9
90 "Miami Intl"                               0.299         -9
91 "Mc Carran Intl"                           0.258         -8
92 "Salt Lake City Intl"                      0.176         -8
93 "Long Beach"                              -0.0620       -10
94 "Martha\\\\'s Vineyard"                   -0.286        -11
95 "Seattle Tacoma Intl"                     -1.10         -11
96 "Honolulu Intl"                           -1.37          -7
97 "John Wayne Arpt Orange Co"               -7.87         -11
98 "Palm Springs Intl"                      -12.7          -13.5
```

(b)

Although Table *planes* has a "speed" column, most of the values in this column are *NA*. Therefore, in this problem, I choose to calculate the average flights using information from Table *flights*.

```
fastest_model <- flights %>%
  ## Calculate the average speed for each model
  mutate(speed_mph = distance/(air_time/60)) %>%
  inner_join(planes, by = "tailnum") %>%
  group_by(model) %>%
  summarise(
    avg_speed_mph = mean(speed_mph, na.rm = TRUE),
    total_flights = n()
  ) %>%
  arrange(-avg_speed_mph) %>%
  slice(1) %>%
  print()
```

```
# A tibble: 1 x 3
  model   avg_speed_mph total_flights
  <chr>           <dbl>         <int>
1 777-222          483.             4
```

Problem 2

```
nnmaps <- read.csv("chicago-nmmaps.csv")
#' get_temp - Compute the average temperature for a given month
#' using a specified averaging function
#' @param month either a numeric 1-12 or a string
#' @param year numeric year
#' @param data the data set to obtain data from
#' @param celsius logically indicating whether the results
#' should be in celsius. Default FALSE
#' @param average_fn a function with which to compute the mean. Default is mean
#' @return numeric vector of length 1 indicating
#' the average temperature for a given month
get_temp <- function(month, year, data, celsius = FALSE, average_fn = mean){
  ## Rename input parameters to avoid
  ## potential conflicts with column names in 'data'
  input_month <- month
  input_year <- year

  # Convert string months to numeric format
  # Convert string months to numeric format
  if (is.character(input_month)) {
    input_month <- case_when(
      input_month %in% c("Jan", "January")   ~ 1,
      input_month %in% c("Feb", "February")  ~ 2,
      input_month %in% c("Mar", "March")     ~ 3,
      input_month %in% c("Apr", "April")     ~ 4,
      input_month %in% c("May")              ~ 5,
      input_month %in% c("Jun", "June")      ~ 6,
      input_month %in% c("Jul", "July")      ~ 7,
      input_month %in% c("Aug", "August")    ~ 8,
      input_month %in% c("Sep", "September") ~ 9,
      input_month %in% c("Oct", "October")   ~ 10,
      input_month %in% c("Nov", "November")  ~ 11,
      input_month %in% c("Dec", "December")  ~ 12,
      TRUE                                    ~ -1 # Default case
    )
  }

  # Ensure month is between 1-12
  if (input_month < 1 || input_month > 12) {
    stop("Invalid month input. Please provide a valid month")
  }

  # Extract the relevant data
  temp_data <- data %>%
    filter(month_numeric == input_month, year == input_year) %>%
    pull(temp)

  # Convert to Celsius if requested
  if (celsius) {
    temp_data <- (temp_data - 32) * (5/9)
  }

  # Return the average temperature using the provided function
  return(average_fn(temp_data))
}
```

```
get_temp("Apr", 1999, data = nnmaps)
```

```
[1] 49.8
```

```
get_temp("Apr", 1999, data = nnmaps, celsius = TRUE)
```

```
[1] 9.888889
```

```
get_temp(10, 1998, data = nnmaps, average_fn = median)
```

```
[1] 55
```

```
# get_temp(13, 1998, data = nnmaps)
# Error message for this case:
# Error in get_temp(13, 1998, data = nnmaps) :
#  Invalid month input. Please provide a valid month
get_temp(2, 2005, data = nnmaps)
```

```
[1] NaN
```

```
get_temp("November", 1999, data =nnmaps, celsius = TRUE,
         average_fn = function(x) {
           x %>% sort -> x
           x[2:(length(x) - 1)] %>% mean %>% return
         })
```

```
[1] 7.301587
```

Problem 3

```
/* import csv file */
PROC IMPORT DATAFILE='/home/u63651235/sasuser.v94/recs2020_public_v5.csv'
            OUT=recs2020
            DBMS=CSV;
RUN;
```

```
/* get some information about the data */
/* I comment for the submitted version since the output is too long.
proc print data=recs2020 (obs=5);
run;

proc contents data=recs2020;
run;
*/
```

(a)

```
/* (a) */
proc sql noprint;
   /* weight of each state */
   create table state_weights as
   select state_name, sum(NWEIGHT) as total_weight
   from recs2020
   group by state_name;

   /* sum of weights of all states */
   select sum(NWEIGHT) into :totalWeight from recs2020;

   /* percentage of each state and sort */
   create table state_percent as
   select state_name,
          total_weight / &totalWeight as percentage
   from state_weights
   order by percentage desc;
run;

/* print the state with the highest percentage */
proc print data=state_percent(obs=1);
run;

/* print the percentage of Michigan */
proc print data=state_percent;
   where state_name = "Michigan";
run;
```

(b)

```
/* (b) */
/* get records with strictly positive value of total electricity cost */
data positive_costs;
    set recs2020;
    if DOLLAREL > 0;
run;

/* plot the histogram */
proc sgplot data=positive_costs;
    histogram DOLLAREL;
    title "Histogram of Total Electricity Cost with Strictly Positive Values";
run;
```

(c)

```
/* (c) */
/* calculate the log of total elevtricity cost */
data positive_costs;
    set positive_costs;
    log_DOLLAREL = log(DOLLAREL);
run;

/* plot the histogram */
proc sgplot data=positive_costs;
    histogram log_DOLLAREL;
    title "Histogram of Log of Total Electricity Cost";
run;
```

(d)

```
/* (d) */
/* Use the sum of NCOMBATH, NHAFBATH and TOTROOMS as the number of rooms */
data positive_costs;
    set positive_costs;
    TOTAL_ROOM = NCOMBATH + NHAFBATH + TOTROOMS;
run;
```

```
/* linear regression */
proc glm data=positive_costs;
    where PRKGPLC1 ne -2; /* Exclude observations where PRKGPLC1 is -2(missing values) */
    class PRKGPLC1;
    model log_DOLLAREL = TOTAL_ROOM PRKGPLC1;  /* response and predictors */
    weight NWEIGHT;  /* weights */
    title "Linear Regression on Log of Total Electricity Cost";
    output out=predicted_data p=predicted_log; /* get predicted data for (e) */
run;
```

(e)

```
/* (e) */
data predicted_data;
    set predicted_data;
    predicted_DOLLAREL = exp(predicted_log);
run;

proc sgplot data=predicted_data;
    scatter x=DOLLAREL y=predicted_DOLLAREL;
    xaxis label="Actual Total Electricity Cost";
    yaxis label="Predicted Total Electricity Cost";
    title "Scatterplot of Predicted vs Actual Total Electricity Cost";
run;
```

Problem 4

(a)

The codebook contains the variable names, labels, and tabulations of responses for the questions asked in the survey.

(b)

```
/* (b) */
/* import csv file */
PROC IMPORT DATAFILE='/home/u63651235/sasuser.v94/public2022.csv'
            OUT=fulldata
            DBMS=CSV;
RUN;

/* get some information about the data */
proc print data=fulldata (obs=5);
run;

/* select needed variables */
proc sql;
    create table subsetData as
    select CaseID, weight_pop, B3, ND2, B7_b, GH1, race_5cat, educ_4cat
    from fullData;
run;

/* transformation */
proc format;
    value $B3fmt
    'Much worse off' = 1
    'Somewhat worse off' = 2
    'About the same' = 3
    'Somewhat better off' = 4
    'Much better off' = 5;

    value $ND2fmt
    'Much higher' = 1
    'Somewhat higher' = 2
    'About the same' = 3
    'Somewhat lower' = 4
    'Much lower' = 5;

    value $B7_bfmt
    'Poor' = 1
    'Only fair' = 2
    'Good' = 3
    'Excellent' = 4;
    value $GH1fmt
    'Own your home with a mortgage or loan' = 1
    'Own your home free and clear (without a mortgage or loan)' = 2
    'Pay rent' = 3
    'Neither own nor pay rent' = 4;

    value $race_fmt
    'White' = 1
    'Black' = 2
    'Hispanic' = 3
    'Asian' = 4
    'Other' = 5;

    value $edu_fmt
    'Less than a high school degree' = 1
    'High school degree or GED' = 2
    'Some college/technical or  associates degree' = 3
    "Bachelor's degree or more" = 4;
run;

data Data_trans;
    set subsetData;

    B3_trans = put(B3, B3fmt.);
    ND2_trans = put(ND2, ND2fmt.);
    B7_b_trans = put(B7_b, B7_bfmt.);
    GH1_trans = put(GH1, GH1fmt.);
    race_trans = put(race_5cat, race_fmt.);
    educ_trans = put(educ_4cat, edu_fmt.);
run;
```

(c)

```
/* (c) */
/* export the transformed data into Stata format */
proc sql;
    create table outputData as
    select CaseID, weight_pop, B3_trans, ND2_trans, B7_b_trans, GH1_trans, race_trans, edu
    from Data_trans;
```

```
run;
proc export data=outputData
    outfile="/home/u63651235/sasuser.v94/public2022.dta"
    dbms=stata replace;
run;

proc print data=outputData (obs=5);
run;

proc contents data=outputData;
run;
```

(d)

```
. do "C:\Users\LOCAL_~3\Temp\STD25c4_000000.tmp"

. * (d) import data
. use "K:\public2022.dta", clear

. count
  11,667

. describe

Contains data from K:\public2022.dta
 Observations:        11,667
    Variables:             8
--------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
--------------------------------------------------------------------------------
CaseID          double  %12.0g
weight_pop      double  %12.0g
B3_trans        str2    %2s
ND2_trans       str2    %2s
B7_b_trans      str2    %2s
GH1_trans       str2    %2s
race_trans      str2    %2s
educ_trans      str2    %2s
--------------------------------------------------------------------------------
```

```
Sorted by:
```

There are total 11,667 observations, aligning with that in the codebook and 8 selected variables, aligning with output from SAS.

(e)

```
. * (e) transform response to a binary variable
. gen B3_bin = cond(inlist(B3_trans, "1", "2"), 0, 1)
```

(f)

```
. * (f) logisitic regression model
. svyset CaseID [pw=weight_pop]

Sampling weights: weight_pop
             VCE: linearized
     Single unit: missing
        Strata 1: <one>
 Sampling unit 1: CaseID
           FPC 1: <zero>


.
. * convert string variable to numeric
. encode ND2_trans, gen(ND2_num)

. encode B7_b_trans, gen(B7_b_num)

. encode GH1_trans, gen(GH1_num)

. encode race_trans, gen(race_num)

. encode educ_trans, gen(educ_num)

.
. svy: logistic B3_bin i.ND2_num i.B7_b_num i.GH1_num i.race_num i.educ_num
(running logistic on estimation sample)

Survey: Logistic regression
```

```
Number of strata =      1                        Number of obs   =      11,667
Number of PSUs   = 11,667                        Population size = 255,114,223
                                                 Design df       =      11,666
                                                 F(17, 11650)    =       57.08
                                                 Prob > F        =      0.0000

------------------------------------------------------------------------------
             |             Linearized
      B3_bin | Odds ratio   std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
     ND2_num |
          2  |   1.085264   .1004103     0.88   0.377     .9052586    1.301062
          3  |   1.068285   .0911387     0.77   0.439     .9037764    1.262738
          4  |   1.298803   .2656084     1.28   0.201     .8698664    1.939252
          5  |   1.268982    .211473     1.43   0.153     .9153563    1.759224
             |
    B7_b_num |
          2  |   3.023388   .1478851    22.62   0.000     2.746971    3.327619
          3  |   6.053997   .4822353    22.61   0.000     5.178836    7.077051
          4  |   11.91989   4.109229     7.19   0.000     6.064588    23.42845
             |
     GH1_num |
          2  |    .938628   .0530243    -1.12   0.262     .8402394    1.048538
          3  |   1.025434   .0602294     0.43   0.669     .9139174    1.150559
          4  |   1.422547    .140974     3.56   0.000     1.171397    1.727545
             |
    race_num |
          2  |   2.031276   .1647816     8.74   0.000     1.732648    2.381373
          3  |   1.183977   .0847211     2.36   0.018      1.02903    1.362255
          4  |   1.562447   .1970896     3.54   0.000     1.220175    2.000729
          5  |   .9875135    .163249    -0.08   0.939     .7141897     1.36544
             |
    educ_num |
          2  |   1.119783   .1303859     0.97   0.331     .8912739     1.40688
          4  |   1.360402     .15108     2.77   0.006     1.094276    1.691249
          S  |   1.178032   .1311927     1.47   0.141     .9470055    1.465419
             |
       _cons |   .5763645   .0791819    -4.01   0.000     .4402969    .7544819
------------------------------------------------------------------------------
Note: _cons estimates baseline odds.
```

```
.
```

All of the $p-value$ of all categories of variable *ND2_NUM* larger than 0.05, meaning statistically insignificant. Therefore, we have no enough evidence to conclude that long-term concerns about climate change impact current day concerns about financial stability, controlling other predictors.

(g)

```
. * (g) output data
. export delimited "K:\output.csv"
file K:\output.csv saved

.
end of do-file
```

(h)

```
library(survey)
```

```
Loading required package: grid

Loading required package: Matrix

Attaching package: 'Matrix'

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

    expand, pack, unpack

Loading required package: survival

Attaching package: 'survey'

The following object is masked from 'package:graphics':

    dotchart
```

```r
public2022 <- read.csv("output.csv")

des <- svydesign(id = ~ CaseID, weight = ~ weight_pop, data = public2022)

fit <- svyglm(B3_bin ~ factor(ND2_num) + factor(B7_b_num) + factor(GH1_num) +
                factor(race_num) + factor(educ_num), design = des, family = quasibinomial(

summary(fit)
```

```

Call:
svyglm(formula = B3_bin ~ factor(ND2_num) + factor(B7_b_num) + 
    factor(GH1_num) + factor(race_num) + factor(educ_num), design = des, 
    family = quasibinomial())

Survey design:
svydesign(id = ~CaseID, weight = ~weight_pop, data = public2022)

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -0.55101    0.13738  -4.011 6.09e-05 ***
factor(ND2_num)2   0.08182    0.09252   0.884 0.376515    
factor(ND2_num)3   0.06605    0.08531   0.774 0.438791    
factor(ND2_num)4   0.26144    0.20450   1.278 0.201119    
factor(ND2_num)5   0.23822    0.16665   1.429 0.152899    
factor(B7_b_num)2  1.10638    0.04891  22.619  < 2e-16 ***
factor(B7_b_num)3  1.80072    0.07966  22.606  < 2e-16 ***
factor(B7_b_num)4  2.47821    0.34473   7.189 6.94e-13 ***
factor(GH1_num)2  -0.06334    0.05649  -1.121 0.262241    
factor(GH1_num)3   0.02512    0.05874   0.428 0.668936    
factor(GH1_num)4   0.35245    0.09910   3.557 0.000377 ***
factor(race_num)2  0.70866    0.08112   8.736  < 2e-16 ***
factor(race_num)3  0.16888    0.07156   2.360 0.018287 *  
factor(race_num)4  0.44625    0.12614   3.538 0.000405 ***
factor(race_num)5 -0.01257    0.16531  -0.076 0.939414    
factor(educ_num)2  0.11314    0.11644   0.972 0.331254    
factor(educ_num)4  0.30778    0.11106   2.771 0.005590 ** 
factor(educ_num)S  0.16385    0.11137   1.471 0.141255    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 1.002519)
```

```
Number of Fisher Scoring iterations: 4
```

```
pseudo_R2 <- 1 - (fit$deviance / fit$null.deviance)
pseudo_R2
```

```
[1] 0.09010785
```