Problem Set 4, Solutions

Statistics 506, Fall 2017

Question 1

In this question you will use SAS to fit mixed models to the audiometry data from the 2009 NHANES survey previously used for problem set 1, question 3. As before you may treat this as a simple random sample. You may wish to refer to the previous solution when specifying the mixed models.

An executable sas script for this question can be found here (./PS4_Q1.sas).

a. *Determine how to load the audiometry and demographic data into SAS and then merge on the common identifier seqn. Drop all cases without audiometry data.*

Here is the sas code I used for this.

```
libname mylib './data/';

/* (a) Read and merge data.
 *     I also do part of (b) here.
 */
libname demod xport './data/DEMO_D.XPT';
libname auxd  xport './data/AUX_D.XPT';

data nhanes;
 merge demod.DEMO_D auxd.AUX_D;
  by seqn;
  keep seqn riagendr ridageyr auxu:;
  drop auxu1k2:;
  if auxu1k1r = . then delete;
```

b. *Produce a reduced data set in long format containing columns for:*

- *the unique identifier seqn*
- *[demographics] age and gender*
- *[hearing threshold tests] ear, frequency, and threshold or result*
- *An age group indicator for whether the person is older than 25 years of age.*

For each person use just the first test at each frequency for each ear.

```
/* (b) Reshape to long format.
 *  The approach used here is:
 *   1. separate demographics & AUXU*
 *   2. convert AUXU* to long
 *   3. Merge demographics back in.
 */

data nhanes_demo;
  set nhanes(rename=(riagendr=gender ridageyr=age));
    keep seqn gender age;

 data nhanes_aux;
   set nhanes(rename=(AUXU1K1R=AUXU1KR AUXU1K1L=AUXU1KL));
   keep seqn auxu:;

/* drop additional missing values 888 or 666 */

proc transpose data=nhanes_aux out=aux_long;
   by seqn;
   var auxu:;
 run;

data nhanes_long;
   merge aux_long(drop=_LABEL_ rename=(_NAME_=test col1=thresh))
         nhanes_demo;
   by seqn;

   ear = 'right';
   if prxmatch("/L/",test) then ear = 'left';
   right = 1;
   if ear = 'left' then right=0;

   freq = prxchange('s/AUXU(.)K(.)/$1/', 1, test);
   if prxmatch("/U500/", test) then freq='5';
   fr = input(freq, 1.);
   if fr ne 5 then fr=10*fr;
   drop freq;

   if thresh = 888 then delete;
   if thresh = 666 then delete;

   older = 0;
   if age ge 25 then older=1;
   older_age = age*older;

   female = 1;
   if gender = 1 then female = 0;
```

When creating `nhanes_long` we also create an explicit interaction `older_age` between the age group indicator `older` and `age` for later use. The code below will export this data to csv for use in answering question 2.

```
  proc export data=nhanes_long
    outfile = '~/nhanes_long.csv'
    dbms=dlm replace;
    delimiter = ',';
  run;
```

c. *Filter your data to contain only the 1000 Hz test for the right ear so that each unique id appears just once. Use proc reg to fit regression models for answering the following questions*

You can find additional discussion of parts c and d in the solution for question 2. The code for producing these estimates along with sas output is below.

i. At this frequency, is there a significant interaction between age group and gender in determining how well an individual hears form their right ear?

The data step below subsets the data and creates an explicit interaction between `older` and `female`.

```
data nhanes_1kr;
    set nhanes_long;
    older_female = older*female;
    where ear='right' & fr=10;
```

Here is the `proc reg` statement to carry out the regression.

```
proc reg data=nhanes_1kr outest=sum_1kr
          rsquare;
model thresh = older female older_female;
```

And here is the output from `proc print data=sum_1kr;`

```
                                          Parameter Estimates

                                        Parameter       Standard
                    Variable        DF   Estimate          Error    t Value     Pr
> |t|

                    Intercept        1    5.48000        0.37792      14.50
<.0001
                    older            1   23.17633        0.71545      32.39
<.0001
                    female           1   -0.14799        0.53326      -0.28
0.7814
                    older_female     1    1.38780        1.03789       1.34
0.1813
```

ii. After controlling for age group and gender, is age still important as a continuous variable?

We will first re-center age relative to the minimum age in each group.

```
data nhanes_1kr_ii;
   set nhanes_1kr;
   group_age = age - 12;
   if older = 1 then group_age = age - 70;
```

Then we can run the regression.

```
proc reg data=nhanes_1kr_ii outest=sum_1kr_ii rsquare;
    model thresh = older female group_age;
```

Examining the results below we can see that age within group remains important.

```
                                        Parameter Estimates

                                   Parameter       Standard
                Variable     DF     Estimate          Error    t Value    Pr > |t
|

                  Intercept     1       2.44549        0.41400        5.91       <.0
001
                  older         1      20.38474        0.57773       35.28       <.0
001
                  female        1       0.08467        0.44564        0.19       0.8
493
                  group_age     1       0.83558        0.06806       12.28       <.0
001
```

iii. Is the effect of age, as a continuous variable, significantly different among the older and/or younger age groups?

The SAS code below creates an explicit interaction between `group_age` and `older`, fits a model with this interaction, and then prints the results.

```
data nhanes_1kr_iii;
   set nhanes_1kr_ii;
   older_age = older*group_age;

proc reg data=nhanes_1kr_iii outest=sum_1kr_iii rsquare;
     model thresh = older group_age older_age female;

proc print data = sum_1kr_iii;
```

Based on the results below, we see that the slope for age in the older group is significantly higher than for the younger group.

```
                              Parameter Estimates

                                   Parameter       Standard
             Variable     DF        Estimate          Error    t Value    Pr > |t|

             Intercept     1         4.82370        0.51785       9.31      <.0001
             older         1        15.13856        0.90257      16.77      <.0001
             group_age     1         0.15745        0.11263       1.40      0.1623
             older_age     1         1.05596        0.14054       7.51      <.0001
             female        1         0.06412        0.44119       0.15      0.8845
```

d. Repeat part (c) using `proc mixed` and data from both ears and all frequencies.

In each of the models below, the *fixed effects* in our model statement will be the same as above with the exception that we also account for the frequency of each hearing test.

There are various ways to specify the *random effects* to account for subject level differences. The approach below includes random intercepts for each person and for each ear within each person. We can speed up the fitting process by leaving the person level id `seqn` as continuous and sorting on this.

First, we will add some derived variables to the full data set.

```
/* add derived variables to full data */
data nhanes_d;
 set nhanes_long;
  older_female = older*female;
  group_age = age - 12;
  if older = 1 then group_age = age - 70;
```

Next, we will sort on `seqn` and `ear`.

```
/* after sorting on SEQN we can use it as
 * continuous in specifying random effects
 */

 proc sort data=nhanes_d out=nd_sorted;
   by seqn ear;
```

In the models below, the option `subject = seqn` in the `ranodm` statement ensures the random effects across subjects are independent. In the model statement, the options `s cl` request a parameter summary table with confidence limits.

Here is the statement for (i).

```
proc mixed data=nd_sorted method=REML noclprint;
   class ear;
   model thresh = fr older female older_female /
         s cl;
   random intercept ear / subject=seqn;

run;
```

Here are the estimated fixed effects and variances for the random effects.

The Mixed Procedure

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
Intercept	0.9804	0.3219	2723	3.05	0.0023	0.05	0.3492	1.6115
fr	0.1902	0.002315	33E3	82.19	<.0001	0.05	0.1857	0.1948
older	43.7334	0.5902	33E3	74.10	<.0001	0.05	42.5766	44.8902
female	-0.8617	0.4393	33E3	-1.96	0.0498	0.05	-1.7229	-0.00060
older_female	-7.3679	0.8558	33E3	-8.61	<.0001	0.05	-9.0454	-5.6905

Covariance Parameter Estimates

Cov Parm	Subject	Estimate
Intercept	SEQN	81.0792
ear	SEQN	13.3125
Residual		129.25

This output tells us that yes, there is a significant interaction between gender and age group when considering all frequencies with older women having average hearing threshold 8.2 (7.37 + .86) db lower than older men. We see also that the subject level intercepts have standard deviation $\sqrt{81.1}$ = 9 db and that ear level effects have standard deviation $\sqrt{13.3} = 3.6$ db.

ii. Here is the `proc mixed` specification for part (ii).

```
 proc mixed data=nd_sorted method=REML noclprint;
   class ear;
   model thresh = fr older female group_age /
         s cl;
   random intercept ear / subject=seqn;

run;
```

Here are the estimated fixed effects and variances for the random effects.

The Mixed Procedure

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
Intercept	-0.7814	0.3521	2723	-2.22	0.0266	0.05	-1.4718	-0.09095
fr	0.1902	0.002315	33E3	82.18	<.0001	0.05	0.1857	0.1948
older	36.9165	0.4782	33E3	77.20	<.0001	0.05	35.9792	37.8537
female	-2.9295	0.3687	33E3	-7.95	<.0001	0.05	-3.6522	-2.2069
group_age	0.8024	0.05634	33E3	14.24	<.0001	0.05	0.6920	0.9128

Covariance Parameter Estimates

Cov Parm	Subject	Estimate
Intercept	SEQN	76.7765
ear	SEQN	13.3227
Residual		129.25

Based on these results we can answer that age is important over and above the differences between age groups with each additional year of age over the group minimum raising the expected hearing threshold by .8 db.

As an aside, notice the reduced variance for the random intercept on account of additional variance being explained by fixed effects.

iii. And finally, the call to part (iii).

```
proc mixed data=nd_sorted method=REML noclprint;
    class ear;
    model thresh = fr older female group_age older_age /
      s cl;
    random intercept ear / subject=seqn;

 run;
```

The results, shown below, indicate that among the older group each additional year of age raises the expected hearing threshold by 1.05 + .13 = 1.2 db, as compared to only .13 db per year among the younger group. The difference in slopes represented by `older_age` is highly significant.

The Mixed Procedure

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
Intercept	1.5873	0.4340	2724	3.66	0.0003	0.05	0.7363	2.4383
fr	0.1902	0.002315	33E3	82.19	<.0001	0.05	0.1857	0.1948
older	-42.0057	8.6899	33E3	-4.83	<.0001	0.05	-59.0382	-24.9731
female	-2.9500	0.3633	33E3	-8.12	<.0001	0.05	-3.6620	-2.2380
group_age	0.1268	0.09273	33E3	1.37	0.1714	0.05	-0.05492	0.3086
older_age	1.0527	0.1157	33E3	9.10	<.0001	0.05	0.8259	1.2796

Covariance Parameter Estimates

Cov Parm	Subject	Estimate
Intercept	SEQN	74.0563
ear	SEQN	13.3314
Residual		129.25

Question 2

For this question, we export the “long” data produced from parts a and b of question 1 for use in R. Here’s an approach to these questions using mixed models fit and specified by `lme4`.

First some preliminaries.

```
#load packages
library(lme4)

## read the data
aud_long = data.table::fread('./nhanes_long.csv')
```

In part (c) we use a subset of data and linear models with fixed effects only so we create that subset here.

```
# Filter
aud_subset = dplyr::filter(aud_long, ear=='right' & fr == 10)
```

i. At this frequency, is there a significant interaction between age group and gender in determining how well an individual hears form their right ear?

To answer this question, we fit a model with our age group variable “older” and our gender variable “female”.

```
fit_ci = lm(thresh~older*female,data=aud_subset)
summary(fit_ci)
```

```
##
## Call:
## lm(formula = thresh ~ older * female, data = aud_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.896  -5.480  -0.480   4.668  89.520
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.4800     0.3779  14.501   <2e-16 ***
## older         23.1763     0.7154  32.394   <2e-16 ***
## female        -0.1480     0.5333  -0.278    0.781
## older:female   1.3878     1.0379   1.337    0.181
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.95 on 2729 degrees of freedom
## Multiple R-squared:  0.4369, Adjusted R-squared:  0.4363
## F-statistic: 705.9 on 3 and 2729 DF,  p-value: < 2.2e-16
```

At a 5% significance level, there is no significant main effect for gender nor for the interaction between gender and age group.

ii. After controlling for age group and gender, is age still important as a continuous variable? To understand the "still" here consider the following model showing the importance of age after controlling for gender:

```
fit_cii0 = lm(thresh ~ age+gender, data=aud_subset)
summary(fit_cii0)
```

```
##
## Call:
## lm(formula = thresh ~ age + gender, data = aud_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.155  -5.973  -0.805   4.417  88.415
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.991605   0.759531  -1.306    0.192
## age          0.389990   0.008129  47.976   <2e-16 ***
## gender       0.167264   0.448913   0.373    0.709
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.73 on 2730 degrees of freedom
## Multiple R-squared:  0.4576, Adjusted R-squared:  0.4572
## F-statistic:  1151 on 2 and 2730 DF,  p-value: < 2.2e-16
```

A simple approach to the question as posed indicates that continuous age remains important. However, the intercept and the coefficient on "older" are difficult to interpret.

```
fit_cii1 = lm(thresh ~ age + older + gender, data=aud_subset)
summary(fit_cii1)
```

```
##
## Call:
## lm(formula = thresh ~ age + older + gender, data = aud_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.857  -6.623  -0.872   4.212  86.705
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7.66613    1.26211  -6.074 1.42e-09 ***
## age           0.83558    0.06806  12.276  < 2e-16 ***
## older       -28.07887    4.25887  -6.593 5.15e-11 ***
## gender        0.08467    0.44564   0.190    0.849
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.64 on 2729 degrees of freedom
## Multiple R-squared:  0.4661, Adjusted R-squared:  0.4655
## F-statistic:   794 on 3 and 2729 DF,  p-value: < 2.2e-16
```

We can make these easier to interpret by creating a new variable "group_age" giving the age above the minimum age in each group. This way the intercept represents the threshold for a hypothetical 12 year old and the coefficient on "older" represents the difference between hypothetical 12 and 70 year olds.

```
aud_subset$group_age = with(aud_subset, ifelse(older==1,age-70,age-12))

fit_cii2 = lm(thresh ~ older + group_age + gender, data=aud_subset)
summary(fit_cii2)
```

```
##
## Call:
## lm(formula = thresh ~ older + group_age + gender, data = aud_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.857  -6.623  -0.872   4.212  86.705
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.36082    0.75119   3.143  0.00169 **
## older       20.38474    0.57773  35.284  < 2e-16 ***
## group_age    0.83558    0.06806  12.276  < 2e-16 ***
## gender       0.08467    0.44564   0.190  0.84933
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.64 on 2729 degrees of freedom
## Multiple R-squared:  0.4661, Adjusted R-squared:  0.4655
## F-statistic:   794 on 3 and 2729 DF,  p-value: < 2.2e-16
```

Based on this model, we can say that the older group has a hearing threshold about 20 db higher than the younger group at 1000 kHz. Each additional year of age above the group minimums, 12 and 70, respectively, increases the expected threshold by about .8 db.

iii. Is the effect of age, as a continuous variable, significantly different among the older and/or younger age groups?

To answer this, we include an interaction between `group_age` and `older` . You could also include the interaction explicitly by creating separate age variables for each group.

```
fit_ciii = lm(thresh ~ older*group_age + female, data=aud_subset)
summary(fit_ciii)
```

```
##
## Call:
## lm(formula = thresh ~ older * group_age + female, data = aud_subset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.523  -5.675  -0.518   4.547  89.074
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.82370    0.51785   9.315  < 2e-16 ***
## older           15.13856    0.90257  16.773  < 2e-16 ***
## group_age        0.15745    0.11263   1.398    0.162
## female           0.06412    0.44119   0.145    0.884
## older:group_age  1.05596    0.14054   7.514 7.75e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.52 on 2728 degrees of freedom
## Multiple R-squared:  0.4769, Adjusted R-squared:  0.4761
## F-statistic: 621.7 on 4 and 2728 DF,  p-value: < 2.2e-16
```

This output tells us that each year of age among the older group increases the expected hearing threshold by about 1.2 db (1.06 + .16). In contrast the expected increase for each year of age in the younger group is only around .16 and not significant at the 5% level.

d. Now we answer the three questions above using data from both ears and at all frequencies. The key here is to account for the nesting measurements within people and within ears. This question asks you use a mixed model for this and we will specify a random intercept for each ear within each person. We will allow the random effects for each ear to be correlated within a person.

Below is the model with just frequency, age group, and gender along with a random intercept for each person and ear.

```
fit_di0 = lmer(thresh ~ (1 | ear / SEQN) + older + fr + older + female, data=aud_long
)
summary(fit_di0)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: thresh ~ (1 | ear/SEQN) + older + fr + older + female
##    Data: aud_long
##
## REML criterion at convergence: 303330.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.5101 -0.5300  0.0063  0.5257  5.1068
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  SEQN:ear (Intercept)  97.02    9.85
##  ear      (Intercept)   0.00    0.00
##  Residual             129.15   11.36
## Number of obs: 38100, groups:  SEQN:ear, 5458; ear, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  1.941084   0.238255    8.15
## older       40.191992   0.330247  121.70
## fr           0.190270   0.002314   82.22
## female      -2.778436   0.291179   -9.54
##
## Correlation of Fixed Effects:
##        (Intr) older  fr
## older  -0.386
## fr     -0.340  0.002
## female -0.615  0.032  0.000
```

Based on the table summarizing the *random effects* we see that, as one might expect, there is more variance between individuals (`SEQN`) than between ears (`ear`) within an individual.

In the call to `lmer` observe that fitting via REML is the default. The following links may be helpful in understanding the syntax further:

- This answer (https://stats.stackexchange.com/questions/228800/crossed-vs-nested-random-effects-how-do-they-differ-and-how-are-they-specified) to a question on stack exchange.
- This book chapter (http://lme4.r-forge.r-project.org/book/Ch2.pdf).

Below are models for answering the actual questions.

i. Is there an interaction between age group and gender?

```
fit_di = lmer(thresh ~ (1 | SEQN / ear) + older*female + fr, data=aud_long)
summary(fit_di)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: thresh ~ (1 | SEQN/ear) + older * female + fr
##    Data: aud_long
##
## REML criterion at convergence: 301253
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7456 -0.5358  0.0047  0.5281  4.8618
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ear:SEQN (Intercept)  13.31    3.649
##  SEQN     (Intercept)  81.08    9.005
##  Residual             129.25   11.369
## Number of obs: 38100, groups:  ear:SEQN, 5458; SEQN, 2735
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)   0.980370   0.321876    3.05
## older        43.733411   0.590198   74.10
## female       -0.861730   0.439354   -1.96
## fr            0.190235   0.002315   82.19
## older:female -7.367949   0.855824   -8.61
##
## Correlation of Fixed Effects:
##              (Intr) older  female fr
## older        -0.511
## female       -0.686  0.374
## fr           -0.252  0.002  0.000
## older:femal  0.352 -0.690 -0.513 -0.001
```

When incorporating data for all frequencies we find that there is a meaningful interaction between age group and gender and that the audible thresholds for older women are are roughly 7 db lower than for older men on average.

ii. After controlling for age group and gender, is age still important as a continuous variable?

```
aud_long$group_age = with(aud_long, ifelse(older==1,age-70,age-12))
fit_dii =  lmer(thresh ~ (1 | SEQN / ear) + older + group_age + female + fr, data=aud
_long)
#fit_cii2 = lm(thresh ~ older + group_age + gender, data=aud_subset)
summary(fit_dii)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: thresh ~ (1 | SEQN/ear) + older + group_age + female + fr
##    Data: aud_long
## 
## REML criterion at convergence: 301136
## 
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -4.7089 -0.5380  0.0020  0.5254  4.8902 
## 
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ear:SEQN (Intercept)  13.32    3.650  
##  SEQN     (Intercept)  76.78    8.762  
##  Residual             129.25   11.369  
## Number of obs: 38100, groups:  ear:SEQN, 5458; SEQN, 2735
## 
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) -0.781370   0.352114   -2.22
## older       36.916476   0.478180   77.20
## group_age    0.802399   0.056343   14.24
## female      -2.929512   0.368704   -7.95
## fr           0.190231   0.002315   82.18
## 
## Correlation of Fixed Effects:
##           (Intr) older  grop_g female
## older     -0.024                     
## group_age -0.546 -0.486              
## female    -0.513  0.040 -0.024       
## fr        -0.230  0.001  0.001  0.000
```

As before age remains a significant factor even after controlling for differences between these two age groups and genders.

iii. Is the effect of age, as a continuous variable, significantly different among the older and/or younger age groups?

```
fit_diii = lmer(thresh ~ (1 | SEQN / ear) + older*group_age + female + fr, data=aud_l
ong)
summary(fit_diii)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: thresh ~ (1 | SEQN/ear) + older * group_age + female + fr
##    Data: aud_long
##
## REML criterion at convergence: 301056.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6988 -0.5377  0.0029  0.5266  4.8946
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ear:SEQN (Intercept)  13.33    3.651
##  SEQN     (Intercept)  74.06    8.606
##  Residual             129.25   11.369
## Number of obs: 38100, groups:  ear:SEQN, 5458; SEQN, 2735
##
## Fixed effects:
##                  Estimate Std. Error t value
## (Intercept)      1.587259   0.434010    3.66
## older           31.685314   0.743421   42.62
## group_age        0.126829   0.092729    1.37
## female          -2.949964   0.363273   -8.12
## fr               0.190233   0.002315   82.19
## older:group_age  1.052728   0.115747    9.10
##
## Correlation of Fixed Effects:
##             (Intr) older  grop_g female fr
## older       -0.476
## group_age   -0.742  0.435
## female      -0.414  0.030 -0.009
## fr          -0.186  0.000  0.000  0.000
## older:grp_g  0.600 -0.774 -0.801 -0.006  0.001
```

As in the previous model, we find that age is more impact among the older group than among the younger group.

Question 3

You can view the SAS files for this exercise at the links below:

- read_data.sas (./read_data.sas) To read the data into SAS and save as a `.sas7bdat` .
- PS4Q3.sas (./PS4Q3.sas) Performs the computations to answer questions in parts b and c.

For part a, use the file provided and edit the `INFILE` statement to point to the text file. I also added a `libname` statement and a data step to save the file for subsequent use.

```
libname mylib '/home/jbhender/ps4q3/data/';

/* INFILE data step provided with download */

/* edit this line to match linux file name (TXT to txt and correct path) */
INFILE './data/Medicare_Provider_Util_Payment_PUF_CY2014.txt'

/* Save as sas7bdat */
DATA mylib.Medicare_PS_PUF_2014;
  set Medicare_PS_PUF;
```

The results shown here are for the 2014 data.

For (b) read in the data and create `totpay`:

```
data da;
  set mylib.medicare_ps_puf_2014;
  totpay = line_srvc_cnt*average_medicare_payment_amt;
```

For part (c.i), use `proc sql` to compute the average cost for each procedure and order to answer the questions.

```
proc sql;
 create table dax as
   select sum(totpay) as s, sum(line_srvc_cnt) as n, sum(totpay)/sum(line_srvc_cnt) a
s avg,
          hcpcs_code as hcpcs, min(hcpcs_description) as service
   from da
   group by hcpcs_code
   order by -avg;

 create table daz as
   select *
   from dax
   where n > 1e5;

quit;

/* Highest avg cost */
proc print data=dax(obs=1);

/* Highest avg cost among services with n > 1e5 */
proc print data=daz(obs=1);
```

For part (c.ii) begin by limiting the data to individual providers,

```
proc sql;

  /* Limit the data to individual providers and summarize */
  create table dai as
    select npi, sum(totpay) as s, min(provider_type) as type
      from da
     where nppes_entity_code = "I"
    group by npi;

quit;
```

You can then create a table with provider counts by type among those with more than $1 million in total charges.

```
proc sql;

create table hcf as
    select count(npi) as n, type
      from dai
     where s > 1e6
     group by type
     order by -n;

quit;

proc print data=high_cost_freq(obs=10);
```

The top ten are:

Rank	N	Provider Type
1	1185	Ophthalmology
2	651	Hematology/Oncology
3	369	Radiation Oncology
4	293	Rheumatology
5	239	Dermatology
6	237	Cardiology
7	212	Medical Oncology
8	133	Internal Medicine
9	77	Diagnostic Radiology
10	75	Nephrology

Likewise, create a table with average total payments for each provider type.

```
create table prov_avg as
    select avg(s) as avg, type
      from dai
    group by type
    order by -avg;

proc print data=prov_avg(obs=2);

/* Create table with last two rows */
data low_prov_avg;
  set prov_avg nobs=nobs;
  if _n_ ge (nobs-2);

proc print data=low_prov_avg(obs=2);
```

The provider types with the two highest average total charges among individual providers in 2014 were Ophthalmology ($343,777) and Radiation Oncology ($338,148).

The provider types with the two lowest average total charges in 2014 were Mass Immunization Roster Biller ($4,047) and Anesthesiologist Assistants ($6,577).

Question 4

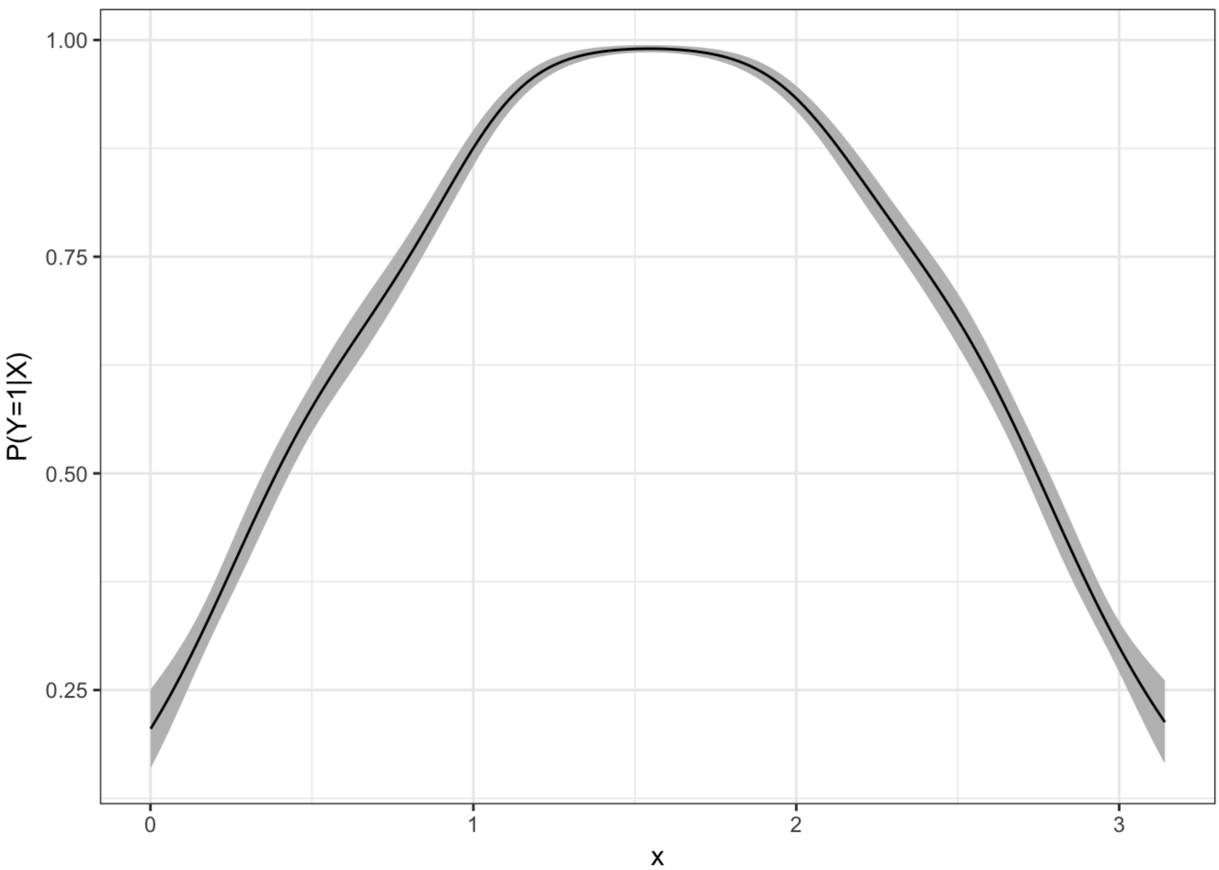
In this question you were asked to use logistic regression to predict `y` as a smooth function of `x` from the provided data. Below, I show to to do this using the default 'thin plate' splines for the smooth function. I also provide a plot showing the estimated probabilities conditional on `x`.

```
# Load libraries
library(mgcv); library(tidyverse); library(doParallel)
foo = load('~/ps506/PS4/ps4_q4.RData')

## Part a
fit = gam(y~s(x), family=binomial(link='logit'), data=get(foo))

# New data for plotting
new_data = with(get(foo), tibble(x=seq(min(x),max(x),length.out=1e3)))
pred = predict(fit,new_data, type='response', se.fit=TRUE)
new_data = new_data %>%
  mutate(y = pred$fit, u = y + 2*pred$se.fit, l=y - 2*pred$se.fit)

new_data %>%
  ggplot(aes(x=x,y=y)) +
  geom_ribbon(aes(ymin=l,ymax=u),fill='grey') +
  geom_line() +
  theme_bw() +
  ylab('P(Y=1|X)')
```



Estimated probability that Y=1 given X from a logistic model using a cubic regression spline to create a smooth function of X. The shaded area shows an approximate pointwise 95% confidence region using plus or minus two standard errors of the fit.

```
cap = 'Estimated probability that Y=1 given X from a logistic model using
a cubic regression spline to create a smooth function of X.  The shaded area
shows an approximate pointwise 95% confidence region using plus or minus
two standard errors of the fit.'
```

Though you weren't asked to we can check the in-sample error:

```
mean(sample_data$y != 1*{fit$fitted.values>.5})
```

```
## [1] 0.22
```

Here is a cross-validation function for part b making use of the `predict.gam` method.

```
xvalidate_seq = function(df,folds=10){
  # df - a data frame
   n = nrow(df)
   start = seq(0,n,length.out={folds+1})

   n_right = 0
   for(k in 1:folds){
     ind = {start[k]+1}:{start[k+1]}
     df_in = df[-ind,]
     df_out = df[ind,]

     fit = gam(y~s(x), data=df_in, family=binomial(link='logit'))
     y_hat = {predict(fit,df_out,type='response') > .5}
     n_right = n_right + sum(y_hat == df_out$y)
   }
   c('ErrorRate'=1- n_right / n)
}
xvalidate_seq(sample_data,folds=10)
```

```
## ErrorRate
##    0.2204
```

c. Now we modify to run in parallel when "cores > 1".

```
xvalidate = function(df,folds=10,cores=1){
  # df - a data frame
   n = nrow(df)
   start = seq(0,n,length.out={folds+1})

   do_fold = function(k){
     ind = {start[k]+1}:{start[k+1]}
     df_in = df[-ind,]
     df_out = df[ind,]

     fit = gam(y~s(x), data=df_in, family=binomial(link='logit'))
     y_hat = {predict(fit,df_out,type='response') > .5}
     sum(y_hat == df_out$y)
   }
   ## Compute serially or in parallel
   if(cores == 1){
     n_right = 0
     for(k in 1:folds) n_right = n_right + do_fold(k)
   } else{
     #cat('Running in parallel...')
      n_right = foreach(k=1:folds, .packages='mgcv', .combine='+') %dopar% {
        do_fold(k)
      }
   }

   c('ErrorRate'= 1 - n_right/n)
}
```

Below is a quick (local) test that this works.

```
ncores = 2
cl = makeCluster(ncores)
registerDoParallel(cl)

xvalidate(sample_data,folds=5,cores=ncores)
```

```
## ErrorRate
##    0.2209
```

```
stopCluster(cl)
```

d. To do this, I first saved the `xvalidate` function from part c in a script (./xvalidate.R) `xvalidate.R`. I then copied this script and the data to a folder in my Flux home directory:

```
scp xvalidate.R ps4_q4.RData flux-xfer.arc-ts.umich.edu:/home/jbhender/ps4q4/
```

The script below (shown without headers) will carry out the execution and is saved as `P4Q4d.R` (./P4Q4d.R):

```
library(mgcv); library(doParallel)
load('/home/jbhender/ps4q4/ps4_q4.RData')
source('/home/jbhender/ps4q4/xvalidate.R')

ncores = 2
cl = makeCluster(ncores)
registerDoParallel(cl)

xvalidate(sample_data,folds=10,cores=ncores)

stopCluster(cl)
```

Here (./P4Q4d_pbs_jbhender.txt) is a pbs script used to execute the code.

e. Here (./P4Q4e.R) is a modified script which accepts command line arguments for the number of cores and folds.

```
# Libraries, data, source files
library(mgcv); library(doParallel)
load('/home/jbhender/ps4q4/ps4_q4.RData')
source('/home/jbhender/ps4q4/xvalidate.R')

# Get command line arguments and assign as global variables
# Use to assign "cores" and "folds"
ca = commandArgs()
ca = ca[grep('=',ca)]
ca = strsplit(ca,'=')
lapply(ca,function(x) assign(x[1],as.numeric(x[2]), envir=.GlobalEnv))

# Warn and quit if problem.
if(sum(c('cores','folds') %in% ls())<2) stop("Please specify 'cores' and 'folds'.")

# Setup cluster
cl = makeCluster(cores)
registerDoParallel(cl)

# Computation
xvalidate(sample_data,folds=folds,cores=cores)

# Close cluser
stopCluster(cl)
```

There are a number of ways you can pass the folds using `$PBS_ARRAYID` . The way I show here uses a string in exponential notation. In the sh shell, we need to use double quotes “” to expand shell variables within a string:

```
R CMD BATCH --vanilla "--args cores=8 folds=1e${PBS_ARRAYID}" \
  /home/jbhender/ps4q4/P4Q4e.R \
  /home/jbhender/ps4q4/P4Q4e_Rout_${PBS_ARRAYID}_jbhender.txt
```

Here `\` is used to split a long command over multiple lines for the purpose of display. Here (./run_e.pbs) is a copy of the pbs script.

I ran these jobs using 8 cores. The table below shows the walltime, CPU time, and reported prediction error for each number of folds.

Folds	Wall time	CPU time	Error
10	11s	27s	22.04%
100	27s	2m 36s	22.07%
1000	3m 12s	24m 31s	22.04%
10000	31m 8s	4h 7m 7s	22.02%

The out-of-sample accuracy is nearly constant, while the estimation time is approximately linear taking ~1.5s CPU time per fold.