** This file provides example SAS code to analyze simulated data that was generated to mimic data arising from the ADHD SMART study (PI: William Pelham). An accompanying handout "ADHD Handout.pdf" describes the variables in the data.

** SAS code used in the Summer Institute on adaptive interventions; ** in Ann Arbor, June 19-20, 2014

********************
*** READ IN DATA ***
********************

* create a SAS library that points to the folder that contains ADHD_Simulated_Data.sas7bdat; *** LINE 20 IS THE NEXT LINE ***;
libname libdat "???/SAS Files/";

data adhddat;
  set libdat.adhd_simulated_data;
  label a1 = "initial txt: A1=-1=MED; A1=1=BMOD"
    a2 = "second txt: A2=-1=ADD; A2=1=INTSFY"
    r = "R=0=early non-response; R=1=early response"
    o11 = "oppositional defiant disorder dx at baseline: 1=yes; 0=no"
    o12 = "ADHD score at baseline: hi is better"
    o13 = "received med prior to txt and found acceptable: 1=yes; 0=no"
    o14 = "race: 1=white; 0=non-white"
    o21 = "number of months until non-response: missing for responders"
    o22 = "adherence to stage 1 intervention: 1=yes; 0=no"
    y = "school performance at end of school year"
    ReRand = "did child get re-randomized? note: ReRand = 1 - R"
; run;

** brief description of the data; proc means data=adhddat; run;
** take a look at a frequency table of the initial treatment, early response by week 8, and second stage treatments; proc sort data=adhddat; by a1 a2 r;
proc freq data=adhddat;
  table a1;
  table r;
  table a2*r;
run;

* center baseline covariates at the means ; * this will facilitate interpretation of the regression parameters; proc means data=adhddat; run;
data dat1;
  set adhddat;
  o11c = o11 - 0.3533333;
  o12c = o12 - -0.1205948;
  o13c = o13 - 0.3133333;
  o14c = o14 - 0.8066667;
  o21c = o21 - 4.5858586;
  o22c = o22 - 0.4533333;
run;
proc means data=dat1; run;
proc print data=dat1 (firstobs=15 obs = 35) heading=h; run;
** code to examine the main effect of initial treatment.;

* regression to compare first-line treatments on end of study outcome Y = school performance;
proc genmod data = dat1;
   model y = al o1c o12c o13c o14c;
   estimate 'Mean Y under BMOD' intercept 1 al 1;
   estimate 'Mean Y under MED' intercept 1 al -1;
   estimate 'Between groups diff' al 2;
run;
the following implements a t-test which is an alternative to the regression model:

```sas
data dat2;
  set dat1;
  if al=1 then altmp="BMOD";
  if al=-1 then altmp="MED";
run;
proc ttest data=dat2;
  class altmp; var y;
run;
```

* the analysis above is for illustrative purposes only;
* in an actual SMART, there will be longitudinal measures for y;
* therefore, the analysis above can be carried out using a longitudinal;
* data analysis method such as a growth curve model (linear mixed model).

**************************************************************************************
** code to examine impact of BMOD and MED on short-term response rate
** this is not an useful analysis in terms of building an adaptive txt strategy;
* because this outcome does not incorporate the effects of future treatment decisions;
* nonetheless, is still interesting in its own right;
* and we analyze it here for completeness.

```sas
proc freq data=dat2;
  table al*r / chisq nocol nopercent;
run;
```

* again for simplicity, here we used a simple frequency table with a chi-squared test;
* to compare the proportions responding to initial treatment;
* in practice, we can also use a logistic regression to adjust for initial covariates.;
** code to compare the best second-line tactic. ;
** this code allows investigators to examine whether, among non-responders to initial;
** treatment, it is better to intensify treatment (a2=1) vs augment txt (a2=-1). ;

* use only the non-responders;
data dat3;
  set dat1; if R=0;
run;
* run the regression;
proc genmod data = dat3;
  model y = a2 o11c o12c o13c o14c o21c o22c;
  estimate 'Mean Y w/INTENSIFY tactic' intercept 1 a2 1;
  estimate 'Mean Y w/ADD tactic'      intercept 1 a2 -1;
  estimate 'Between groups difference'             a2 2;
run;

* the analysis above are for illustrative purposes only. ;
* in an actual SMART, there will be longitudinal measures for y ;
* therefore, the analysis above can be carried out using a longitudinal ;
* data analysis method such as a growth curve model (linear mixed model).;
* code to estimate the mean outcome under the (MED, Add BMOD) AI
* this is AI#1 in the workshop slides

* this is the adaptive intervention that begins with medication, then continues with
  * medication given early response, but augments medication with BMOD given early
  * non-response.

* create indicator for the (MED, Add BMOD) AI;
* and assign weights;
data dat5; set dat1;
  * create indicator Z1 = -1/1 = didn't/did follow AI#1;
  Z1=-1;
  if A1=-1 and R=1 then Z1=1;
  if A1=-1 and R=0 and A2=-1 then Z1=1;
  * define weights: responders get a 2, non-responders get a 4;
  W=2*R + 4*(1-R);
run;
* run the W-weighted regression;
proc genmod data = dat5;
  class id;
  model y = z1 ;
  scwgt w;
  repeated subject = id / type = ind; * this lines requests robust standard errors;
  estimate 'Mean Y under AI #1 (MED, Add BMOD)' intercept 1 z1 1;
run;}
** code to estimate the mean outcome under the (BMOD, Add MED) AI;
** this is AI#2 in the workshop slides;

* create indicator and assign weights;
data dat6; set dat2;
  * create indicator Z2 = -1/1 = didn't/did follow AI#2 (BMOD, Add MED);
    Z2=-1;
    if A1=1 and R=1 then Z2=1;
    if A1=1 and R=0 and A2=-1 then Z2=1;
  * define weights: responders get a 2, non-responders get a 4;
    W=2*R + 4*(1-R);
run;
* run the W-weighted regression;
proc genmod data = dat6;
  class id;
  model y = z2 ;
  scwgt w;
  repeated subject = id / type = ind; * this lines requests robust standard errors;
  estimate 'Mean Y under AI#2 (BMOD, Add MED)' intercept 1 z2 1;
run;

** an exercise for the student is to estimate the mean outcome under  ;
** remaining two AIs using code similar to the above. ;
** primary data 
** analyses 
** part II 

** PII(a) **

** code to compare the mean under the 2 AIs beginning with different treatments **

** the following code is used to estimate the mean outcomes under the following AIs **
** AI#1: (MED, Add BMOD), vs **
** AI#2: (BMOD, Add MED) **
** the goal is to do this simultaneously. this also facilitates making inferences **
** about the difference in the mean of the AIs. **

* first, create indicator variables and define the weights; 
data dat7; set dat1; 
* first create indicator Z1 to identify those following (MED, Add BMOD) AI#1; 
* here we show an alternate way to define this indicator in SAS, different from above; 
Z1=-1; if A1*R=-1 then Z1=1; if (1-A1)*(1-R)*A2=-2 then Z1=1; 
* second, create indicator Z2 to identify those following (BMOD, Add MED) AI#2; 
* here we show an alternate way to define this indicator in SAS, different from above; 
Z2=-1; if A1*R= 1 then Z2=1; if (1+A1)*(1-R)*A2=-2 then Z2=1; 
* define the weights; 
W=2*R + 4*(1-R); 
r

** next, limit the data set only to those following either AI#1 or AI#2; 
data dat8; set dat7; if Z1=1 or Z2=1; run; 
** finally, conduct a regression analysis to compare mean outcomes under AI#1 vs AI#2; 
proc genmod data = dat8; 
class id; 
model y = z1 ; 
scwgt w; 
repeated subject = id / type = ind; 
estimate 'Mean Y under AI#1 (MED, ADD BMOD)' intercept 1 z1 1; 
estimate 'Mean Y under AI#2 (BMOD, ADD MED)' intercept 1 z1 -1; 
estimate '   Diff: AI#1 - AI#2' z1 2; 
run;
** code to estimate and compare the mean outcome under all 4 of the embedded AIs
** using 1 regression analysis

** this analysis differs from the above in that it requires both weighting and replication. see the slides for intuition concerning replication

* first, replicate responders and define weights;

```sas
data dat9; set dat1;
  * define weights and create replicates of responders
  * (with equal "probability of getting A2");
  if R=1 then do;
    ob = 1; A2 = -1; weight = 2; output;
    ob = 2; A2 = 1 ; weight = 2; output;
  end;
  else if R=0 then do;
    ob = 1; weight = 4; output;
  end;
run;
proc freq data=dat1;
  table r*a2;
run;
proc freq data=dat9;
  table r*a2;
run;
proc print data=dat9 (firstobs=45 obs=65) heading=h ; run;
```
proc genmod data = dat9;
class id;
model y = a1 a2 a1*a2 ;
scwgt weight;
repeated subject = id / type = ind;
* these statements get the mean under each AI;
estimate 'Mean Y: AI#1(MED, Add BMOD)' int 1 a1 -1 a2 -1 a1*a2  1;
estimate 'Mean Y: AI#2(BMOD, Add MED)' int 1 a1  1 a2 -1 a1*a2 -1;
estimate 'Mean Y: AI#3(MED, INTENSFY)' int 1 a1 -1 a2  1 a1*a2 -1;
estimate 'Mean Y: AI#4(BMOD, INTNSFY)' int 1 a1  1 a2  1 a1*a2  1;
* these statements are to get all pairwise comparisons;
estimate '       Diff: AI#1 – AI#2   ' int 0 a1 -2 a2  0 a1*a2  2;
estimate '       Diff: AI#1 – AI#3   ' int 0 a1  0 a2 -2 a1*a2  2;
estimate '       Diff: AI#1 – AI#4   ' int 0 a1 -2 a2 -2 a1*a2  0;
estimate '       Diff: AI#2 – AI#3   ' int 0 a1  2 a2 -2 a1*a2  0;
estimate '       Diff: AI#2 – AI#4   ' int 0 a1  0 a2 -2 a1*a2 -2;
estimate '       Diff: AI#3 – AI#4   ' int 0 a1 -2 a2  0 a1*a2 -2;
run;
** a statistical advantage about estimating the means under all 4 AI;**
** simultaneously is that we can increase statistical efficiency;**
** (lower standard errors) in the estimation of the differences in means;**
** we can do this by adjusting for baseline covariates that might**
** explain variability in Y. however, we must be careful not to adjust for;**
**post-baseline/time-varying covariates or intermediate outcomes.**

```plaintext
proc genmod data = dat9;
  class id;
  model y = a1 a2 a1*a2 o12c o14c;
  scwgt weight;
  repeated subject = id / type = ind;
  * these statements get the mean under each AI;
  estimate 'Mean Y: AI#1(MED, Add BMOD)' int 1 a1 -1 a2 -1 a1*a2 1;
  estimate 'Mean Y: AI#2(BMOD, Add MED)' int 1 a1 1 a2 -1 a1*a2 -1;
  estimate 'Mean Y: AI#3(MED, INTENSFY)' int 1 a1 -1 a2 1 a1*a2 -1;
  estimate 'Mean Y: AI#4(BMOD, INTNSFY)' int 1 a1 1 a2 1 a1*a2 1;
  * these statements are to get all pairwise comparisons;
  estimate 'Diff: AI#1 – AI#2' int 0 a1 -2 a2 0 a1*a2 2;
  estimate 'Diff: AI#1 – AI#3' int 0 a1 0 a2 -2 a1*a2 2;
  estimate 'Diff: AI#1 – AI#4' int 0 a1 -2 a2 -2 a1*a2 0;
  estimate 'Diff: AI#2 – AI#3' int 0 a1 2 a2 -2 a1*a2 0;
  estimate 'Diff: AI#2 – AI#4' int 0 a1 0 a2 -2 a1*a2 -2;
  estimate 'Diff: AI#3 – AI#4' int 0 a1 -2 a2 0 a1*a2 -2;
run;
```
** Secondary data analyses;**

** Q-learning;**

** Step (1) of Q-learning is to examine intermediate moderators of the effect of add vs intensify;**

** this will help us tailor second stage treatment for non-responders based on the status of the participants up to the point of non-response. **

* we only use non-responders to answer this question;

```plaintext
* first, fit a full model to investigate whether candidate tailoring variables are moderators;
proc genmod data = dat10;
  model y = o11c o12c o13c o14c o21c a1 o22c a2 a2*a1 a2*o22c;
run;
* Both a2 and o22 appear to moderate the effect;
```

* Second, fit a model with appropriate comparisons;

```plaintext
* given stage 1 option a1 = MED = -1;
  estimate 'NR MED ADH INT' intercept 1 a1 -1 o22 1 a2 1 a2*a1 -1 a2*o22 1 ;
  estimate 'NR MED ADH ADD' intercept 1 a1 -1 o22 1 a2 -1 a2*a1 1 a2*o22 -1 ;
  estimate 'INT vs ADD for NR MED ADH' a2 2 a2*a1 -2 a2*o22 2 ;
  estimate 'NR MED Non-ADH INT' intercept 1 a1 -1 o22 0 a2 1 a2*a1 -1 a2*o22 0 ;
  estimate 'NR MED Non-ADH ADD' intercept 1 a1 -1 o22 0 a2 -1 a2*a1 1 a2*o22 0 ;
  estimate 'INT vs ADD for NR MED Non-ADH' a2 2 a2*a1 -2 a2*o22 0 ;
* now given stage 1 option a1 = BMOD = 1;
  estimate 'NR BMOD ADH INT' intercept 1 a1 1 o22 1 a2 1 a2*a1 1 a2*o22 1 ;
  estimate 'NR BMOD ADH ADD' intercept 1 a1 1 o22 1 a2 -1 a2*a1 -1 a2*o22 -1 ;
  estimate 'INT vs ADD for NR BMOD ADH' a2 2 a2*a1 2 a2*o22 2 ;
  estimate 'NR BMOD Non-ADH INT' intercept 1 a1 1 o22 0 a2 1 a2*a1 1 a2*o22 0 ;
  estimate 'NR BMOD Non-ADH ADD' intercept 1 a1 1 o22 0 a2 -1 a2*a1 -1 a2*o22 0 ;
  estimate 'INT vs ADD for NR BMOD Non-ADH' a2 2 a2*a1 2 a2*o22 0 ;
run;
```
Step (2+3) manually; this "manual" code is optional, for statistical connoisseurs!

** here we do the step 2 and Step 3 manually
** then we describe software PROC QLEARN that does these two steps automatically
** it can be skipped

** Step 2 of Q-learning is to assign non-responders their estimated optimal outcome they would have if they were to have been assigned the best stage 2 treatment based on adherence. The other part of Step 2 is to assign all responders their observed outcome.

** Step 3 of Q-learning is to use this new "future-set-to-optimal" outcome and use it to examine baseline moderators of the effect of MED vs BMOD.
** these baseline moderators help us understand how to best assign/tailor first stage treatment given that the future treatment has already been chosen optimally.

* we first show how to do these two steps manually--later, we describe software that does Steps 2 and 3 automatically.

* Step 2 uses the results from the Step 1 model above to construct Yhat
* the new outcome will be called yhat;
  data dat11;
  set dat1;
  * first, everyone gets their observed outcome;
  yhat = y;
  * second, re-assign the outcome for non-responders;
  if R=0 then
    yhat = 3.0039 - 0.2462*o11c - 0.2961*o12c + 0.0391*o13c + 0.4868*o14c + 0.0758*a1 - 0.0097*o21c - 0.0980*o22 + abs(-0.8640*a2 - 0.1934*a1*a2 + 1.1826*o22*a2);
  run;

proc means data=dat11 n mean std ; var y yhat; run;

* Step 3 regression using the new outcome;
  proc genmod data=dat11;
    model yhat = o11c o12c o14c o13 a1*o13 a1;
    estimate 'BMOD vs MED given MED prior yr ' a1*o13 2 a1 2;
    estimate 'BMOD vs MED given NO MED prior yr' a1*o13 0 a1 2;
  run;

* however, statistical inferences (pvalues, confidence intervals) cannot be based on this output. this is because the output in the above regression does not account for the fact that the yhat is estimated for the non-responders. This (and other statistical issues related to the fact that yhat is a function of an absolute value) required the development of new methodology to calculate standard errors (laber and murphy, 2012).
* the software below, PROC QLEARN, does steps 2 and 3 for the user and it implements the methods in laber and murphy (2012) to calculate the correct asymptotic standard errors for the stage 1 regression.
**** Steps 2 + 3 **********;
**** using software ********;
**** PROC QLEARN **********;
*****************************************************************************;

** the code above helps understand how to use intermediate outcomes to make
** stage 2 decisions about intensifying vs augmenting-- this is Step 1 of the
** Q-learning algorithm to develop a more deeply-tailored adaptive
** intervention. Step 2+3 are about using baseline covariates to make stage 1
** decisions about medication vs behavioral modification, assuming that the best
** treatment was provided in stage 2. The code below implements these steps.

* define an indicator variable for the subsample of the data that will be used in;
* the stage 2 regression. just as above, these are non-responders to initial txt
* this variable is used in the STG2SAMPLE option in PROC QLEARN.

data dat11;
  set dat1; s = 1 - r;  ** S=1 (non-responders) are used in the stage 2 reg.;
run;

* second, set up some contrast matrices to understand how the effects of initial;
* med vs initial bmod differ depending on baseline covariates

data contrasts1;
  input M1 M2 M3 M4 M5 M6 M7;    *no. of matrix columns = no. of stage 1 parameters   
  * the columns correspond to the parameters in the following stage 1 model:
  * b1 + b2 o11c + b3 o12c + b4 o14c + b5 o13 + b6 a1 o13 + b7 a1
  * each row corresponds to a different linear combination of the b's. some linear;
  * combinations can be used to obtain mean outcomes for different children under
  * initial med vs initial bmod. whereas other linear combinations can be used to;
  * compare mean outcomes between med vs bmod for different types of children.
  datalines;
  1 0 0 0 1 1 1
  1 0 0 0 1 -1 -1
  0 0 0 0 0 2 2
  1 0 0 0 0 0 1
  1 0 0 0 0 0 -1
  0 0 0 0 0 0 2
  ;
  * taking into account optimal future decisions (to intensify vs augment)   
  * contrast 1 = mean outcome under bmod for children w/yes med in prior year;
  * contrast 2 = mean outcome under med for children w/yes med in prior year
  * contrast 3 = mean diff (bmod - med) for children w/yes med in prior year
  * contrast 4 = mean outcome under bmod for children w/no med in prior year
  * contrast 5 = mean outcome under med for children w/no med in prior year
  * contrast 6 = mean diff (bmod - med) for children w/no med in prior year
run;

* third, conduct the q-learning algorithm;
* PROC QLEARN provides appropriate confidence intervals;

proc qlearn data=dat11 contrast1=contrasts1 deriveci;
  main1 o11c o12c o14c;
  tailor1 o13;
  main2 o11c o12c o13c o14c o21c;
  tailor2 a1 o22;
  response y;
  stg2sample s;
  stg1trt a1;
  stg2trt a2;
run;
** the following is optional, for statistical connoisseurs! **

The following code is used to estimate the mean outcome under the more deeply-tailored AI identified using q-learning.

```
data dat12;
set dat2;
yopt = 3.4497 + -0.4556*o11c + -0.3458*o12c + -0.0236*o13 + abs( -0.5254*o13*a1 + 0.2934*a1 );
run;
```

```
proc means data=dat12 mean;
var yopt;
run;
```

** that is, given the optimal AI identified using q-learning, this code is to obtain the estimated mean outcome had the entire population followed this optimal AI found by qlearning.

The mean of yopt2 is the estimated mean under the optimal AI found via q-learning. In the future, PROC QLEARN will automatically report this value with an estimated confidence interval. For now, the user must calculate it him/herself.

The estimated mean school performance under the optimal AI found under Q-learning is 3.7165; this is larger (as expected) than the Mean Y under the best design-embedded AI (BMOD, AUGMENT), which was 3.5067

*** eof;
*** eof;