Appendix0_10

```r
# NDNS analysis, data management ------------------------------------------


# Change the data path accordingly ----------------------------------------

setwd(".../UKDA-6533-stata11_se/stata11_se/")
setwd("~/Downloads/UKDA-6533-stata11_se/stata11_se/")

library(epiDisplay)
library(plyr)
library(tidyverse)


# Read the data into memory -----------------------------------------------

library(haven)
data <- read_dta("ndns_rp_yr1-4a_foodleveldietarydata_uk.dta")
data56 <- read_dta("ndns_rp_yr5-6a_foodleveldietarydata.dta")
data78 <- read_dta("ndns_rp_yr7-8a_foodleveldietarydata.dta")

names(data)
names(data56)
names(data78)
names(data)[names(data)=="seriali"] <- "id"
names(data56)[names(data56)=="seriali"] <- "id"
names(data78)[names(data78)=="seriali"] <- "id"


# Extract the variables needed -------------------------------------------

df14d <- data[,c(113,1,2,3,5,6,7,8,9,21, 22, 23, 24, 53, 55,
                 57,58,59,60,61,62,63,64,65)]
var <- names(df14d)
df56d <- data56 %>%
  select(var)
df78d <- data78 %>%
  select(var)
dfs1 <- rbind(df14d, df56d, df78d)

dfs2 <- dfs1[dfs1$Age>=19,] # keep participants who aged 19 or older
rm(data, data56, data78) # remove the unneeded big dataset

dfs2


# Calculate the time (minute and hour) when they eat ----------------------

dfs2$MealTime_chr <- as.character(dfs2$MealTime)
dfs2$MealTime_hm <- unlist(strsplit(dfs2$MealTime_chr," "))[c(FALSE,
                                                              TRUE)]
```

```r
dfs2$MealHourN <- as.numeric(unlist(strsplit(dfs2$MealTime_hm,
                                              ":"))[c(TRUE, FALSE, FALSE)])
dfs2$MealMinN <- as.numeric(unlist(strsplit(dfs2$MealTime_hm,
                                             ":"))[c(FALSE, TRUE, FALSE)])

dfs2$MealMinN0 <- (60*dfs2$MealHourN)+dfs2$MealMinN

dfs3 <- dfs2[order(dfs2$id,dfs2$DayNo,dfs2$MealMinN0),]

length(unique(dfs3$id)) ## number of participants = 6155


# Create a subset data with only the first observation of each participant --------

NDNS <- dfs3[!duplicated(dfs3$id), ]
with(NDNS, tab1(SurveyYear, graph = FALSE, decimal = 2))

# #SurveyYear :
#              Frequency Percent Cum. percent
# NDNS Year 1       801   13.01        13.01
# NDNS Year 2       812   13.19        26.21
# NDNS Year 3       782   12.71        38.91
# NDNS Year 4      1055   17.14        56.05
# NDNS Year 5       625   10.15        66.21
# NDNS Year 6       663   10.77        76.98
# NDNS Year 7       703   11.42        88.40
# NDNS Year 8       714   11.60       100.00
#   Total          6155  100.00       100.00
# how many men and women ---------------------------------------------------

with(NDNS, tab1(Sex, graph = FALSE, decimal = 2))

# Sex :
#           Frequency Percent Cum. percent
# 1              2537   41.22        41.22 Men
# 2              3618   58.78       100.00 Women
#   Total        6155  100.00       100.00
# create a variable combine id and day No ---------------------------------


dfs3 <- dfs3 %>%
  mutate(id_dy = paste(id, DayNo, sep = "D"))

# For each subject, the total energy/carbohydrate intake for each
# eating time can be calculated --------

old<-Sys.time()
Energy <- ddply(dfs3, .(id_dy, id, SurveyYear, DayNo, Age, Sex,
                        DiaryDaysCompleted, MealHourN, DayofWeek),
                summarise, Tot_Energ = sum(EnergykJ),
                Tot_Carb = sum(Carbohydrateg),
                Tot_Sugar = sum(Totalsugarsg),
                Tot_Starch = sum(Starchg),
```

```r
                Tot_Fibre = sum(Englystfibreg),
                Tot_Fat   = sum(Fatg),
                Tot_Prot  = sum(Proteing),
                Tot_Alc   = sum(Alcoholg),
                Tot_NMES  = sum(Nonmilkextrinsicsugarsg))
new<-Sys.time()-old
print(new)
# Time difference of 6.429822 min


# reset the time intervals into time slots --------------------------------

### Breakfast:       6am to 9am
### morning snack:   9am to 12noon
### lunch:           12noon to 2pm
### afternoon snack: 2pm to 5pm
### dinner:          5pm to 8pm
### night snack:     8pm to 10pm
### midnight:        10pm to 6am


Energy <- Energy %>%
  mutate(TimeSlot = cut(MealHourN, breaks = c(6, 9, 12, 14, 17, 20, 22),
                        right = FALSE))


levels(Energy$TimeSlot) <- c(levels(Energy$TimeSlot), "[22, 6)")

Energy$TimeSlot[is.na(Energy$TimeSlot)] <- "[22, 6)"

tab1(Energy$TimeSlot)


# For each subject, the total energy/carbohydrate intake for each
# time slot can be calculated --------
old<-Sys.time()
Energy <- ddply(Energy, .(id_dy, id, SurveyYear, DayNo, Age, Sex,
                          DiaryDaysCompleted, TimeSlot, DayofWeek),
                summarise,
                Tot_Energ = sum(Tot_Energ),
                Tot_Carb = sum(Tot_Carb),
                Tot_Sugar = sum(Tot_Sugar),
                Tot_Starch = sum(Tot_Starch),
                Tot_Fibre = sum(Tot_Fibre),
                Tot_Fat   = sum(Tot_Fat),
                Tot_Prot  = sum(Tot_Prot),
                Tot_Alc   = sum(Tot_Alc),
                Tot_NMES  = sum(Tot_NMES))
new<-Sys.time()-old
print(new)
# Time difference of 3.74195 mins
```

```r
# Calculate the energy from total carbohydrates ---------------------------

Energy <- Energy %>%
  mutate(KJcarbo = Tot_Carb*16) %>%
  mutate(CarKJpercentage = KJcarbo/Tot_Energ) %>%
  mutate(Carbo = cut(CarKJpercentage, breaks = c(0, 0.50, 2), right = FALSE))

Energy0 <- Energy[!(Energy$Tot_Energ == 0), ] # discard those eating occassion
# with 0 energy intake
write.csv(Energy0, file = "Energy_slots.csv") # for later analysis


Energy0$Carbo <- factor(Energy0$Carbo, labels = c("< 50%", ">= 50%"))

vecid<-unique(Energy0$id)

# Filter the data by observation day------------------------------------------

dta_day1 <- Energy0 %>%
  filter(DayNo == 1) %>%
  select(c("id", "id_dy", "Age", "Sex",
           "DayofWeek",  "TimeSlot", "Carbo")) %>%
  mutate(DayofWeek = factor(DayofWeek,
                            levels = c("Monday", "Tuesday",
                                       "Wednesday", "Thursday",
                                       "Friday", "Saturday", "Sunday")))

dta_day2 <- Energy0 %>%
  filter(DayNo == 2) %>%
  select(c("id", "id_dy", "Age", "Sex",
           "DayofWeek",  "TimeSlot", "Carbo")) %>%
  mutate(DayofWeek = factor(DayofWeek,
                            levels = c("Monday", "Tuesday",
                                       "Wednesday", "Thursday",
                                       "Friday", "Saturday", "Sunday")))

dta_day3 <- Energy0 %>%
  filter(DayNo == 3) %>%
  select(c("id", "id_dy", "Age", "Sex",
           "DayofWeek",  "TimeSlot", "Carbo")) %>%
  mutate(DayofWeek = factor(DayofWeek,
                            levels = c("Monday", "Tuesday",
                                       "Wednesday", "Thursday",
                                       "Friday", "Saturday", "Sunday")))

dta_day4 <- Energy0 %>%
  filter(DayNo == 4) %>%
  select(c("id", "id_dy", "Age", "Sex",
           "DayofWeek",  "TimeSlot", "Carbo")) %>%
  mutate(DayofWeek = factor(DayofWeek,
                            levels = c("Monday", "Tuesday",
                                       "Wednesday", "Thursday",
                                       "Friday", "Saturday", "Sunday")))
```

```r
vecid1<-unique(dta_day1$id) # n = 6153
vecid2<-unique(dta_day2$id) # n = 6153
vecid3<-unique(dta_day3$id) # n = 6151
vecid4<-unique(dta_day4$id) # n = 6026


Noday1 <- setdiff(vecid, vecid1) # two subjects did not have day 1 data

Noday2 <- setdiff(vecid, vecid2) # two subjects did not have day 2 data

Noday3 <- setdiff(vecid, vecid3) # four subjects did not have day 3 data

Noday4 <- setdiff(vecid, vecid4) # 129 subjects did not have day 4 data

# Long to wide data -------------------------------------------------------

dta_d1_wide <- dta_day1 %>%
  spread(key = TimeSlot,
         value = Carbo)

head(dta_d1_wide)
names(dta_d1_wide)[6:12] <- c("H6_9", "H9_12", "H12_14", "H14_17",
                              "H17_20", "H20_22", "H22_6")
names(dta_d1_wide)


dta_d2_wide <- dta_day2 %>%
  spread(key = TimeSlot,
         value = Carbo)

head(dta_d2_wide)
names(dta_d2_wide)[6:12] <- c("H6_9", "H9_12", "H12_14", "H14_17",
                              "H17_20", "H20_22", "H22_6")
names(dta_d2_wide)


dta_d3_wide <- dta_day3 %>%
  spread(key = TimeSlot,
         value = Carbo)

head(dta_d3_wide)
names(dta_d3_wide)[6:12] <- c("H6_9", "H9_12", "H12_14", "H14_17",
                              "H17_20", "H20_22", "H22_6")
names(dta_d3_wide)


dta_d4_wide <- dta_day4 %>%
  spread(key = TimeSlot,
         value = Carbo)

head(dta_d4_wide)
names(dta_d4_wide)[6:12] <- c("H6_9", "H9_12", "H12_14", "H14_17",
                              "H17_20", "H20_22", "H22_6")
```

```
names(dta_d4_wide)



# recode NA to not eating -------------------------------------------------

for (i in 6:ncol(dta_d1_wide))
  if(is.factor(dta_d1_wide[,i]))
    levels(dta_d1_wide[,i]) <- c("1", "2", "0")

dta_d1_wide[is.na(dta_d1_wide)] <- "0"


for (i in 6:ncol(dta_d2_wide))
  if(is.factor(dta_d2_wide[,i]))
    levels(dta_d2_wide[,i]) <- c("1", "2",  "0")


dta_d2_wide[is.na(dta_d2_wide)] <- "0"


for (i in 6:ncol(dta_d3_wide))
  if(is.factor(dta_d3_wide[,i]))
    levels(dta_d3_wide[,i]) <-  c("1", "2", "0")

dta_d3_wide[is.na(dta_d3_wide)] <- "0"


for (i in 6:ncol(dta_d4_wide))
  if(is.factor(dta_d4_wide[,i]))
    levels(dta_d4_wide[,i]) <-  c("1", "2", "0")

dta_d4_wide[is.na(dta_d4_wide)] <- "0"


dta_all <- rbind(dta_d1_wide, dta_d2_wide, dta_d3_wide, dta_d4_wide)

dta_all <- dta_all[order(dta_all$id,dta_all$id_dy),]

# Export the data for Mplus -----------------------------------------------

write_csv(dta_all, path = "dta_NDNS_Tslots.csv")
write_delim(dta_all, "dta_NDNS_Tslots.dat", na = ".", delim = " ")
```

```
Mplus VERSION 7.4
MUTHEN & MUTHEN
07/28/2018   9:55 AM

INPUT INSTRUCTIONS

  TITLE:     3-class at level 1 (CW), 3-classes at level 2 (CB) random effects model - non-pa
             ordered polytomous variables for carb intake at each time slot over four
             days of NDNS survey 2008/09 - 2015/16
             variable 0 = not eating
```

```
                   1 = eating & carb provided < 50% calorie
                   2 = eating & carb provided >= 50% calorie

DATA:     File is H:\summer_project\Mplus\TimeSlots\NDNS_Tslots.dat;


VARIABLE: NAMES = id id_dy Age Sex H6_9 H9_12 H12_14 H14_17 H17_20
                  H20_22 H22_6;

          USEVAR = H6_9 H9_12 H12_14 H14_17 H17_20
                  H20_22 H22_6;

          auxiliary = Age Sex;

          CATEGORICAL =  H6_9 H9_12 H12_14 H14_17 H17_20
                  H20_22 H22_6;

          CLUSTER = id;

          IDVARIABLE = id_dy;

          BETWEEN = CB;

          WITHIN = H6_9 H9_12 H12_14 H14_17 H17_20
                  H20_22 H22_6;

          CLASSES = CB(3) CW(3);

          MISSING are .;



ANALYSIS:
type = mixture twolevel;
starts = 50 25;
process = 8(starts);


MODEL:
%within%
%overall%
%between%
%overall%
CW ON CB;




Savedata:
  file is H:\summer_project\Mplus\TimeSlots\Multilevel\NDNSslot_CW3CB3.txt;
  save is cprob;
  format is free;
```

```
3-class at level 1 (CW), 3-classes at level 2 (CB) random effects model - non-par
ordered polytomous variables for carb intake at each time slot over four
days of NDNS survey 2008/09 - 2015/16
variable 0 = not eating
1 = eating & carb provided < 50% calorie
2 = eating & carb provided >= 50% calorie

SUMMARY OF ANALYSIS

Number of groups                                                 1
Number of observations                                       24483

Number of dependent variables                                    7
Number of independent variables                                  0
Number of continuous latent variables                            0
Number of categorical latent variables                           2

Observed dependent variables

  Binary and ordered categorical (ordinal)
   H6_9        H9_12       H12_14      H14_17      H17_20      H20_22
   H22_6

Observed auxiliary variables
   AGE         SEX

Categorical latent variables
   CB          CW

Variables with special functions

  Cluster variable      ID
  ID variable          ID_DY

  Within variables
   H6_9        H9_12       H12_14      H14_17      H17_20      H20_22
   H22_6


Estimator                                                      MLR
Information matrix                                        OBSERVED
Optimization Specifications for the Quasi-Newton Algorithm for
Continuous Outcomes
  Maximum number of iterations                                 100
  Convergence criterion                                  0.100D-05
Optimization Specifications for the EM Algorithm
  Maximum number of iterations                                 500
  Convergence criteria
    Loglikelihood change                                 0.100D-02
    Relative loglikelihood change                        0.100D-05
    Derivative                                           0.100D-02
Optimization Specifications for the M step of the EM Algorithm for
Categorical Latent variables
  Number of M step iterations                                    1
```

```
  M step convergence criterion                             0.100D-02
  Basis for M step termination                             ITERATION
Optimization Specifications for the M step of the EM Algorithm for
Censored, Binary or Ordered Categorical (Ordinal), Unordered
Categorical (Nominal) and Count Outcomes
  Number of M step iterations                                      1
  M step convergence criterion                             0.100D-02
  Basis for M step termination                             ITERATION
  Maximum value for logit thresholds                              15
  Minimum value for logit thresholds                             -15
  Minimum expected cell size for chi-square                0.100D-01
Maximum number of iterations for H1                             2000
Convergence criterion for H1                               0.100D-03
Optimization algorithm                                           EMA
Integration Specifications
  Type                                                      STANDARD
  Number of integration points                                    15
  Dimensions of numerical integration                              0
  Adaptive quadrature                                             ON
Random Starts Specifications
  Number of initial stage random starts                           50
  Number of final stage optimizations                             25
  Number of initial stage iterations                              10
  Initial stage convergence criterion                      0.100D+01
  Random starts scale                                      0.500D+01
  Random seed for generating random starts                         0
Parameterization                                               LOGIT
Link                                                           LOGIT
Cholesky                                                         OFF

Input data file(s)
  H:\summer_project\Mplus\TimeSlots\NDNS_Tslots.dat
Input data format  FREE


SUMMARY OF DATA

     Number of missing data patterns             1
     Number of y missing data patterns           0
     Number of u missing data patterns           1
     Number of clusters                       6155



COVARIANCE COVERAGE OF DATA

Minimum covariance coverage value   0.100


UNIVARIATE PROPORTIONS AND COUNTS FOR CATEGORICAL VARIABLES

    H6_9
      Category 1    0.313         7655.000
      Category 2    0.184         4500.000
```

```
      Category 3    0.504        12328.000
    H9_12
      Category 1    0.222         5447.000
      Category 2    0.295         7227.000
      Category 3    0.482        11809.000
    H12_14
      Category 1    0.195         4783.000
      Category 2    0.454        11112.000
      Category 3    0.351         8588.000
    H14_17
      Category 1    0.283         6926.000
      Category 2    0.338         8277.000
      Category 3    0.379         9280.000
    H17_20
      Category 1    0.124         3043.000
      Category 2    0.582        14240.000
      Category 3    0.294         7200.000
    H20_22
      Category 1    0.356         8722.000
      Category 2    0.363         8898.000
      Category 3    0.280         6863.000
    H22_6
      Category 1    0.666        16295.000
      Category 2    0.169         4144.000
      Category 3    0.165         4044.000


RANDOM STARTS RESULTS RANKED FROM THE BEST TO THE WORST LOGLIKELIHOOD VALUES

Final stage loglikelihood values at local maxima, seeds, and initial stage start numbers:

         -166348.815  153942           31
         -166348.815  573096           20
         -166348.815  253358           2
         -166348.816  318230           46
         -166348.816  246261           38
         -166348.873  285380           1
         -166348.908  903420           5
         -166349.394  120506           45
         -166349.394  966014           37
         -166349.394  207896           25
         -166349.395  195873           6
         -166349.513  68985            17
         -166349.514  366706           29
         -166352.737  76974            16
         -166357.057  127215           9
         -166482.723  533738           11
         -166495.844  645664           39
         -166668.918  372176           23


THE BEST LOGLIKELIHOOD VALUE HAS BEEN REPLICATED.  RERUN WITH AT LEAST TWICE THE
RANDOM STARTS TO CHECK THAT THE BEST LOGLIKELIHOOD IS STILL OBTAINED AND REPLICATED.
```

```
THE MODEL ESTIMATION TERMINATED NORMALLY



MODEL FIT INFORMATION

Number of Free Parameters                      134

Loglikelihood

          H0 Value                       -166348.815
          H0 Scaling Correction Factor      1.8182
            for MLR

Information Criteria

          Akaike (AIC)                    332965.630
          Bayesian (BIC)                  334051.799
          Sample-Size Adjusted BIC        333625.950
            (n* = (n + 2) / 24)



MODEL RESULTS USE THE LATENT CLASS VARIABLE ORDER

   CB  CW

  Latent Class Variable Patterns

        CB        CW
      Class     Class

         1         1
         1         2
         1         3
         2         1
         2         2
         2         3
         3         1
         3         2
         3         3


FINAL CLASS COUNTS AND PROPORTIONS FOR THE LATENT CLASS PATTERNS
BASED ON ESTIMATED POSTERIOR PROBABILITIES

  Latent Class
    Pattern

    1  1       4050.97975          0.16546
    1  2       1561.55249          0.06378
    1  3       1286.46696          0.05255
    2  1       2746.94031          0.11220
```

2 2	3011.00217	0.12298
2 3	1341.59686	0.05480
3 1	2748.25320	0.11225
3 2	4770.55950	0.19485
3 3	2965.64876	0.12113

FINAL CLASS COUNTS AND PROPORTIONS FOR EACH LATENT CLASS VARIABLE
BASED ON ESTIMATED POSTERIOR PROBABILITIES

Latent Class Variable	Class		
CB	1	6898.99902	0.28179
	2	7099.53906	0.28998
	3	10484.46094	0.42823
CW	1	9546.17285	0.38991
	2	9343.11426	0.38162
	3	5593.71240	0.22847

FINAL CLASS COUNTS AND PROPORTIONS FOR THE LATENT CLASS PATTERNS
BASED ON THEIR MOST LIKELY LATENT CLASS PATTERN

Class Counts and Proportions

Latent Class Pattern		
1 1	4262	0.17408
1 2	1406	0.05743
1 3	1178	0.04812
2 1	2807	0.11465
2 2	2946	0.12033
2 3	1260	0.05146
3 1	2745	0.11212
3 2	5315	0.21709
3 3	2564	0.10473

FINAL CLASS COUNTS AND PROPORTIONS FOR EACH LATENT CLASS VARIABLE
BASED ON THEIR MOST LIKELY LATENT CLASS PATTERN

Latent Class Variable	Class		
CB	1	6846	0.27962
	2	7013	0.28644
	3	10624	0.43393
CW	1	9814	0.40085
	2	9667	0.39485
	3	5002	0.20431

CLASSIFICATION QUALITY

Entropy 0.630

Average Latent Class Probabilities for Most Likely Latent Class Pattern (Row)
by Latent Class Pattern (Column)

Latent Class Variable Patterns

Latent Class Pattern No.	CB Class	CW Class
1	1	1
2	1	2
3	1	3
4	2	1
5	2	2
6	2	3
7	3	1
8	3	2
9	3	3

	1	2	3	4	5	6	7	8	9
1	0.720	0.091	0.073	0.016	0.032	0.004	0.005	0.033	0.025
2	0.183	0.609	0.098	0.005	0.002	0.030	0.040	0.005	0.027
3	0.211	0.084	0.629	0.008	0.005	0.007	0.011	0.036	0.009
4	0.019	0.004	0.002	0.692	0.184	0.051	0.011	0.034	0.003
5	0.042	0.001	0.001	0.158	0.709	0.045	0.001	0.035	0.009
6	0.012	0.037	0.013	0.065	0.084	0.702	0.042	0.003	0.042
7	0.011	0.029	0.004	0.012	0.002	0.022	0.641	0.126	0.153
8	0.026	0.003	0.009	0.025	0.024	0.001	0.115	0.675	0.123
9	0.046	0.024	0.004	0.003	0.010	0.018	0.079	0.174	0.642

MODEL RESULTS

Within Level

Latent Class Pattern 1 1

	Estimate	S.E.	Est./S.E.	Two-Tailed P-Value
Thresholds				
H6_9$1	-0.718	0.218	-3.294	0.001
H6_9$2	0.973	0.299	3.258	0.001
H9_12$1	-2.516	0.463	-5.433	0.000
H9_12$2	0.675	0.132	5.118	0.000
H12_14$1	-1.025	0.145	-7.057	0.000
H12_14$2	1.240	0.116	10.725	0.000
H14_17$1	-1.566	0.149	-10.520	0.000

H14_17$2	1.090	0.100	10.909	0.000
H17_20$1	-1.998	0.125	-16.000	0.000
H17_20$2	1.549	0.100	15.556	0.000
H20_22$1	-0.933	0.085	-10.914	0.000
H20_22$2	1.829	0.103	17.770	0.000
H22_6$1	0.253	0.083	3.046	0.002
H22_6$2	2.308	0.117	19.691	0.000

Latent Class Pattern 1 2

Thresholds				
H6_9$1	-4.021	1.788	-2.249	0.025
H6_9$2	-0.115	0.259	-0.445	0.656
H9_12$1	0.167	0.373	0.448	0.654
H9_12$2	2.142	0.586	3.657	0.000
H12_14$1	-3.210	1.518	-2.115	0.034
H12_14$2	0.858	0.167	5.124	0.000
H14_17$1	0.044	0.384	0.114	0.909
H14_17$2	1.617	0.293	5.509	0.000
H17_20$1	-2.109	0.390	-5.409	0.000
H17_20$2	1.399	0.196	7.126	0.000
H20_22$1	-0.367	0.174	-2.109	0.035
H20_22$2	2.347	0.382	6.151	0.000
H22_6$1	0.754	0.259	2.912	0.004
H22_6$2	2.542	0.264	9.646	0.000

Latent Class Pattern 1 3

Thresholds				
H6_9$1	-15.000	0.000	999.000	999.000
H6_9$2	2.357	0.783	3.011	0.003
H9_12$1	-1.433	0.372	-3.850	0.000
H9_12$2	-0.604	0.279	-2.166	0.030
H12_14$1	-1.988	0.257	-7.749	0.000
H12_14$2	0.524	0.125	4.209	0.000
H14_17$1	-1.027	0.232	-4.436	0.000
H14_17$2	0.274	0.131	2.087	0.037
H17_20$1	-2.665	0.310	-8.605	0.000
H17_20$2	0.707	0.112	6.322	0.000
H20_22$1	-0.527	0.152	-3.462	0.001
H20_22$2	0.702	0.138	5.102	0.000
H22_6$1	1.119	0.185	6.062	0.000
H22_6$2	1.748	0.183	9.544	0.000

Latent Class Pattern 2 1

Thresholds				
H6_9$1	1.663	0.199	8.370	0.000
H6_9$2	1.839	0.198	9.274	0.000
H9_12$1	-2.150	0.281	-7.643	0.000
H9_12$2	-0.869	0.140	-6.190	0.000
H12_14$1	-1.978	0.191	-10.349	0.000
H12_14$2	0.323	0.078	4.139	0.000
H14_17$1	0.237	0.183	1.293	0.196

```
    H14_17$2           0.782      0.123      6.352      0.000
    H17_20$1          -2.936      0.428     -6.853      0.000
    H17_20$2           0.632      0.081      7.807      0.000
    H20_22$1           0.028      0.142      0.194      0.846
    H20_22$2           0.868      0.086     10.145      0.000
    H22_6$1            0.658      0.109      6.010      0.000
    H22_6$2            1.326      0.100     13.215      0.000

Latent Class Pattern 2 2

 Thresholds
    H6_9$1             1.640      0.171      9.619      0.000
    H6_9$2             1.906      0.179     10.678      0.000
    H9_12$1           -1.954      0.347     -5.636      0.000
    H9_12$2           -0.360      0.127     -2.842      0.004
    H12_14$1          -0.016      0.189     -0.084      0.933
    H12_14$2           0.948      0.135      7.029      0.000
    H14_17$1          -1.906      0.301     -6.327      0.000
    H14_17$2           0.371      0.080      4.614      0.000
    H17_20$1          -0.812      0.116     -7.030      0.000
    H17_20$2           0.910      0.089     10.259      0.000
    H20_22$1          -0.742      0.089     -8.318      0.000
    H20_22$2           0.998      0.085     11.705      0.000
    H22_6$1            0.298      0.083      3.608      0.000
    H22_6$2            1.337      0.099     13.475      0.000

Latent Class Pattern 2 3

 Thresholds
    H6_9$1            -1.072      0.500     -2.144      0.032
    H6_9$2            -0.309      0.346     -0.892      0.372
    H9_12$1            2.441      1.044      2.339      0.019
    H9_12$2            3.599      1.983      1.815      0.069
    H12_14$1          -1.029      0.211     -4.880      0.000
    H12_14$2           0.603      0.123      4.913      0.000
    H14_17$1          -0.010      0.243     -0.041      0.967
    H14_17$2           0.784      0.157      4.977      0.000
    H17_20$1          -0.953      0.203     -4.684      0.000
    H17_20$2           0.779      0.135      5.784      0.000
    H20_22$1          -0.105      0.210     -0.500      0.617
    H20_22$2           1.203      0.135      8.914      0.000
    H22_6$1            0.582      0.299      1.950      0.051
    H22_6$2            1.370      0.206      6.653      0.000

Latent Class Pattern 3 1

 Thresholds
    H6_9$1            -4.593      1.699     -2.703      0.007
    H6_9$2            -2.975      0.428     -6.957      0.000
    H9_12$1           -0.322      0.207     -1.553      0.120
    H9_12$2            0.398      0.363      1.095      0.274
    H12_14$1          -5.060      3.668     -1.380      0.168
    H12_14$2           0.307      0.100      3.080      0.002
    H14_17$1           0.186      0.530      0.351      0.726
```

H14_17$2	0.317	0.245	1.295	0.195
H17_20$1	-4.019	0.957	-4.199	0.000
H17_20$2	0.747	0.093	7.987	0.000
H20_22$1	-0.233	0.132	-1.767	0.077
H20_22$2	0.607	0.109	5.571	0.000
H22_6$1	1.304	0.146	8.918	0.000
H22_6$2	1.850	0.160	11.579	0.000

Latent Class Pattern 3 2

Thresholds				
H6_9$1	-1.232	0.195	-6.305	0.000
H6_9$2	-0.858	0.169	-5.068	0.000
H9_12$1	-4.377	1.937	-2.260	0.024
H9_12$2	-1.488	0.316	-4.717	0.000
H12_14$1	-1.727	0.227	-7.611	0.000
H12_14$2	0.302	0.082	3.666	0.000
H14_17$1	-1.834	0.237	-7.730	0.000
H14_17$2	-0.294	0.186	-1.582	0.114
H17_20$1	-2.588	0.487	-5.313	0.000
H17_20$2	0.631	0.062	10.187	0.000
H20_22$1	-0.920	0.078	-11.852	0.000
H20_22$2	0.462	0.073	6.308	0.000
H22_6$1	0.640	0.119	5.361	0.000
H22_6$2	1.162	0.129	9.039	0.000

Latent Class Pattern 3 3

Thresholds				
H6_9$1	-4.941	5.813	-0.850	0.395
H6_9$2	-2.680	0.887	-3.024	0.002
H9_12$1	-0.765	0.640	-1.195	0.232
H9_12$2	1.164	0.920	1.265	0.206
H12_14$1	-1.415	0.439	-3.226	0.001
H12_14$2	0.566	0.085	6.626	0.000
H14_17$1	-2.052	0.650	-3.158	0.002
H14_17$2	0.612	0.210	2.909	0.004
H17_20$1	-1.627	0.427	-3.810	0.000
H17_20$2	0.713	0.103	6.935	0.000
H20_22$1	-0.850	0.329	-2.585	0.010
H20_22$2	0.685	0.134	5.104	0.000
H22_6$1	1.237	0.195	6.349	0.000
H22_6$2	1.893	0.179	10.582	0.000

Between Level

Categorical Latent Variables

Within Level

Intercepts				
CW#1	-0.076	0.366	-0.208	0.835
CW#2	0.475	0.309	1.539	0.124

```
Between Level

 CW#1       ON
    CB#1              1.223      0.473      2.585      0.010
    CB#2              0.793      0.441      1.796      0.073

 CW#2       ON
    CB#1             -0.282      0.535     -0.526      0.599
    CB#2              0.333      0.455      0.733      0.464

 Means
    CB#1             -0.417      0.100     -4.178      0.000
    CB#2             -0.386      0.067     -5.770      0.000


QUALITY OF NUMERICAL RESULTS

     Condition Number for the Information Matrix              0.428E-04
       (ratio of smallest to largest eigenvalue)


SAVEDATA INFORMATION


  Save file
    H:\summer_project\Mplus\TimeSlots\Multilevel\NDNSslot_CW3CB3.txt

  Order of variables

    H6_9
    H9_12
    H12_14
    H14_17
    H17_20
    H20_22
    H22_6
    ID_DY
    AGE
    SEX
    CPROB1
    CPROB2
    CPROB3
    CPROB4
    CPROB5
    CPROB6
    CPROB7
    CPROB8
    CPROB9
    CB
    CW
    MLCJOINT
    ID

  Save file format           Free
```

```
  Save file record length    10000


DIAGRAM INFORMATION

  Mplus diagrams are currently not available for Mixture analysis.
  No diagram output was produced.


     Beginning Time:  09:55:10
        Ending Time:  10:02:01
       Elapsed Time:  00:06:51



MUTHEN & MUTHEN
3463 Stoner Ave.
Los Angeles, CA  90066

Tel: (310) 391-9971
Fax: (310) 391-8971
Web: www.StatModel.com
Support: Support@StatModel.com

Copyright (c) 1998-2015 Muthen & Muthen
```

```r
library(plyr)
library(epiDisplay)
library(tidyverse)
library(dplyr)
library(readr)
library(haven)
library(naniar)



# read the individual level data sets ----------------------------------------

blood78 <- read_dta("ndns_rp_yr7-8a_indiv.dta")
blood56 <- read_dta("ndns_rp_yr5-6a_indiv.dta")
blood14 <- read_dta("ndns_rp_yr1-4a_indiv_uk.dta")

food14 <- read_dta("ndns_rp_yr1-4a_personleveldietarydata_uk.dta")

food56 <- read_dta("ndns_rp_yr5-6a_personleveldietarydata.dta")

food78 <- read_dta("ndns_rp_yr7-8a_personleveldietarydata.dta")

names(blood78)[names(blood78)=="seriali"] <- "ID"
names(blood56)[names(blood56)=="seriali"] <- "ID"
names(blood14)[names(blood14)=="seriali"] <- "ID"
names(food78)[names(food78)=="seriali"] <- "ID"
names(food56)[names(food56)=="seriali"] <- "ID"
```

```r
names(food14)[names(food14)=="seriali"] <- "ID"


# Loading the data from Mplus output
CW3CB3 <- read_table2("../NDNSslot_CW3CB3.txt",  # change the path accordingly
                      col_names = FALSE)

names(CW3CB3) <- c("Breakfast", "Morning.snack", "Lunch",
                   "Afternoon.snack", "Dinner", "Before.bedtime.snack",
                   "Midnight.food", "ID_DAY", "AGE", "SEX", "CPROB1",
                   "CPROB2", "CPROB3", "CPROB4", "CPROB5", "CPROB6",
                   "CPROB7", "CPROB8", "CPROB9",
                   "CB", "CW", "MLCJOINT", "ID")

CW3idday <- CW3CB3 %>%
  select(ID, ID_DAY, CW, CB, MLCJOINT, AGE, SEX)

Energy_slots <- read_csv("../Energy_slots.csv") # change the path accordingly
Energy_slots <- Energy_slots %>%
  rename(ID = id)

# Recode day level classification to keep consistency
CW3idday$CW_new <- 0
CW3idday$CW_new[CW3idday$CW == 1] <- 3
CW3idday$CW_new[CW3idday$CW == 2] <- 1
CW3idday$CW_new[CW3idday$CW == 3] <- 2

Energy_slots <- Energy_slots %>%
  left_join(CW3idday, by = c("ID", "DayofWeek"))


Energy_slots$TimeSlot <- factor(Energy_slots$TimeSlot,
                                levels = c("[6,9)", "[9,12)",
                                           "[12,14)", "[14,17)",
                                           "[17,20)", "[20,22)", "[22, 6)"))

FoodbyCB_7slots <- Energy_slots %>%
  group_by(ID, TimeSlot) %>%
  summarise(sumEnergy = sum(Tot_Energ),
            sumCarb = sum(Tot_Carb),
            sumSugar = sum(Tot_Sugar),
            sumStarch = sum(Tot_Starch),
            sumFibre = sum(Tot_Fibre),
            sumNMES = sum(Tot_NMES),
            sumFat = sum(Tot_Fat),
            sumProt = sum(Tot_Prot),
            sumAlc = sum(Tot_Alc))

# Calculate the sum of each nutrients for each time slots
Carbsum <- FoodbyCB_7slots %>%
  select(ID, TimeSlot, sumCarb) %>%
  spread(key = TimeSlot,
         value = sumCarb)
```

```r
Carbsum[is.na(Carbsum)] <- 0

names(Carbsum) <- c("ID", "Carb6_9", "Carb9_12", "Carb12_14", "Carb14_17",
                     "Carb17_20", "Carb20_22", "Carb22_6")


Energysum <- FoodbyCB_7slots %>%
  select(ID, TimeSlot, sumEnergy) %>%
  spread(key = TimeSlot,
         value = sumEnergy)
Energysum[is.na(Energysum)] <- 0


names(Energysum) <- c("ID", "Energy6_9", "Energy9_12", "Energy12_14", "Energy14_17",
                     "Energy17_20", "Energy20_22", "Energy22_6")

Starchsum <- FoodbyCB_7slots %>%
  select(ID, TimeSlot, sumStarch) %>%
  spread(key = TimeSlot,
         value = sumStarch)
Starchsum[is.na(Starchsum)] <- 0


names(Starchsum) <- c("ID", "Starch6_9", "Starch9_12", "Starch12_14", "Starch14_17",
                       "Starch17_20", "Starch20_22", "Starch22_6")


Sugarsum <- FoodbyCB_7slots %>%
  select(ID, TimeSlot, sumSugar) %>%
  spread(key = TimeSlot,
         value = sumSugar)
Sugarsum[is.na(Sugarsum)] <- 0


names(Sugarsum) <- c("ID", "Sugar6_9", "Sugar9_12", "Sugar12_14", "Sugar14_17",
                      "Sugar17_20", "Sugar20_22", "Sugar22_6")


Fibresum <- FoodbyCB_7slots %>%
  select(ID, TimeSlot, sumFibre) %>%
  spread(key = TimeSlot,
         value = sumFibre)
Fibresum[is.na(Fibresum)] <- 0


names(Fibresum) <- c("ID", "Fibre6_9", "Fibre9_12", "Fibre12_14", "Fibre14_17",
                      "Fibre17_20", "Fibre20_22", "Fibre22_6")


NMESsum <- FoodbyCB_7slots %>%
  select(ID, TimeSlot, sumNMES) %>%
  spread(key = TimeSlot,
         value = sumNMES)
```

```r
NMESsum[is.na(NMESsum)] <- 0


names(NMESsum) <- c("ID", "NMES6_9", "NMES9_12", "NMES12_14", "NMES14_17",
                     "NMES17_20", "NMES20_22", "NMES22_6")

Fatsum <- FoodbyCB_7slots %>%
  select(ID, TimeSlot, sumFat) %>%
  spread(key = TimeSlot,
         value = sumFat)
Fatsum[is.na(Fatsum)] <- 0


names(Fatsum) <- c("ID", "Fat6_9", "Fat9_12", "Fat12_14", "Fat14_17",
                    "Fat17_20", "Fat20_22", "Fat22_6")


Protsum <- FoodbyCB_7slots %>%
  select(ID, TimeSlot, sumProt) %>%
  spread(key = TimeSlot,
         value = sumProt)
Protsum[is.na(Protsum)] <- 0


names(Protsum) <- c("ID", "Prot6_9", "Prot9_12", "Prot12_14", "Prot14_17",
                   "Prot17_20", "Prot20_22", "Prot22_6")


Alcsum <- FoodbyCB_7slots %>%
  select(ID, TimeSlot, sumAlc) %>%
  spread(key = TimeSlot,
         value = sumAlc)
Alcsum[is.na(Alcsum)] <- 0


names(Alcsum) <- c("ID", "Alc6_9", "Alc9_12", "Alc12_14", "Alc14_17",
                    "Alc17_20", "Alc20_22", "Alc22_6")

# Extract number of days of diary completed

blood14 <- blood14 %>%
  select(ID, Ndays)

blood56 <- blood56 %>%
  select(ID, Ndays)

blood78 <- blood78 %>%
  select(ID, NDays) %>%
  rename(Ndays = NDays)

NDAYS <- rbind(blood14, blood56, blood78)
NDAYS$ID <- as.numeric(NDAYS$ID)
```

```r
IntakeSlots <- Energysum %>%
  left_join(Carbsum, by = "ID") %>%
  left_join(Sugarsum, by = "ID") %>%
  left_join(Starchsum, by = "ID") %>%
  left_join(Fibresum, by = "ID") %>%
  left_join(Fatsum, by = "ID") %>%
  left_join(Protsum, by = "ID") %>%
  left_join(NMESsum, by = "ID") %>%
  left_join(Alcsum, by = "ID") %>%
  left_join(NDAYS, by = "ID")


IntakeSlots$ID <- as.numeric(IntakeSlots$ID)

IntakeSlots$Energy6_9 <- IntakeSlots$Energy6_9/(IntakeSlots$Ndays)

for (i in 3:57){
  IntakeSlots[, i] <- IntakeSlots[, i]/(IntakeSlots$Ndays)
}



# select the variables needed, recoding
# NAs, renaming to the same ---------------------------------------

BMI78 <- blood78 %>%
  select(ID, Sex, age, bmival, wstval, Diabetes, bpmedc2, bpmedd2,
         hyper140_2, hibp140_2,  Glucose, A1C, cigsta3, dnoft3,
         dnnow, wti_Y78, wtn_Y78, wtb_Y78, ethgrp5, ethgrp2,
         cluster1, cluster2, cluster3, nssec8, paidemployment,
         qual7, eqvinc, MarSt2, cluster4, cluster5, area, gor,
         LDL, HDL, Chol, Trig) %>%
  rename(wti = wti_Y78, wtn = wtn_Y78, wtb = wtb_Y78,
         drink = dnoft3) %>%
  mutate(Years = "7-8", MVPAtime = NA, MarStat = NA) %>%
  replace_with_na(replace = list(bmival = -1, qual7 = -8,
                                 wstval = -1, eqvinc = -1,
                                 bpmedd2 = -1, MarSt2 = -1,
                                 bpmedc2 = -1, hyper140_2 = -7,
                                 hibp140_2 = -7, Glucose = -1,
                                 A1C  = -1,  LDL = -1, HDL = -1,
                                 Chol = -1, Trig = -1, dnnow = -1,
                                 drink = -1, ethgrp5 = -4,
                                 ethgrp2 = -4, cigsta3 = -1,
                                 nssec8 = -9,  paidemployment = -1)) %>%
  replace_with_na(replace = list(hyper140_2 = -1, hibp140_2 = -1,
                                 nssec8 = 99, qual7 = -1, drink = -8,
                                 ethgrp2 = -9, ethgrp5 = -9)) %>%
  replace_with_na(replace = list(drink = -9, ethgrp2 = -8, ethgrp5 = -8,
                                 cigsta3 = -8)) %>%
  replace_with_na(replace = list(ethgrp2 = -1, ethgrp5 = -1))
```

```r
BMI56 <- blood56 %>%
  select(ID, Sex, age, area, bmival, wstval, Diabetes, bpmedc2, bpmedd2,
         hyper140_2, hibp140_2, MVPAtime, Glucose, A1C, cigsta3, dnoft3,
         dnnow, wti_Y56, wtn_Y56, wtb_Y56, ethgrp5, ethgrp2, qual7,
         cluster1, cluster2, cluster3, nssec8, paidemployment, eqvinc,
         MarSt2, cluster4, cluster5, area, gor, LDL, HDL, Chol, Trig) %>%
  mutate(Years = "5-6", MarStat = NA) %>%
  rename(wti = wti_Y56, wtn = wtn_Y56, wtb = wtb_Y56, drink = dnoft3) %>%
  replace_with_na(replace = list(bmival = -1, paidemployment = -9,
                                 wstval = -1, qual7 = -8,
                                 bpmedd2 = -1, eqvinc = -1,
                                 bpmedc2 = -1, MarSt2 = -1,
                                 hyper140_2 = -7, hibp140_2 = -7,
                                 Glucose = -1,  A1C  = -1,
                                 dnnow = -1, drink = -1, ethgrp5 = -4,
                                 ethgrp2 = -4, LDL = -1, HDL = -1,
                                 Chol = -1, Trig = -1, cigsta3 = -1,
                                 MVPAtime = -1, nssec8 = -9)) %>%
  replace_with_na(replace = list(hyper140_2 = -1, hibp140_2 = -1,
                                 paidemployment = -8,
                                 drink = -8, nssec8 = 99)) %>%
    replace_with_na(replace = list(drink = -9, paidemployment = -1,
                                   qual7 = -1, cigsta3 = -8))

BMI14 <- blood14 %>%
  select(ID, Sex, age, bmival, wstval, Diabetes, bpmedc, bpmedd, hyper140,
         hibp140, MVPAtime, Glucose, A1C, cigsta3, dnoft3, dnnow,
         wti_CY1234, wtn_CY1234, wtb_CY1234, ethgr5, ethgr2, cluster,
         area, gor, nssec8, paidemployment, qual7, eqvinc, MarSt2,
         MarStat, LDL, HDL, Chol, Trig) %>%
  rename(hyper140_2 = hyper140, hibp140_2 = hibp140, bpmedd2 = bpmedd,
         bpmedc2 = bpmedc, cluster1 = cluster, ethgrp5 = ethgr5,
         ethgrp2 = ethgr2, wti = wti_CY1234, wtn = wtn_CY1234,
         wtb =  wtb_CY1234, drink = dnoft3) %>%
  mutate(cluster2 = NA, cluster3 = NA, cluster4 = NA, cluster5 = NA,
         Years = "1-4") %>%
  replace_with_na(replace = list(bmival = -1, paidemployment = -9,
                                 wstval = -1, qual7 = -8,
                                 bpmedd2 = -1, eqvinc = -1,
                                 bpmedc2 = -1, MarSt2 = -4,
                                 hyper140_2 = -7, MarStat = -4,
                                 hibp140_2 = -7,  Glucose = -1,
                                 A1C  = -1,  dnnow = -1, LDL = -1,
                                 HDL = -1, Chol = -1, Trig = -1,
                                 drink = -1, ethgrp5 = -4, ethgrp2 = -4,
                                 cigsta3 = -1, MVPAtime = -4,
                                 nssec8 = -8)) %>%
  replace_with_na(replace = list(hyper140_2 = -1, hibp140_2 = -1,
                                 MVPAtime = -1, paidemployment = -8,
                                 drink = -8, nssec8 = -1, qual7 = -1,
                                 MarSt2 = -1)) %>%
  replace_with_na(replace = list(drink = -9, paidemployment = -4,
```

```r
                                  cigsta3 = -8, nssec8 = 99)) %>%
  replace_with_na(replace = list(paidemployment = -1))


BMI <- bind_rows(BMI14, BMI56, BMI78)



Energy14 <- food14 %>%
  select(ID, Country, SurveyYear, EnergykJ, Carbohydrateg,
         CHOpctotE, Proteing, ProteinpctotE, Alcoholg, AlcoholpctotE,
         Fatg, FatpctotE)
Energy14$Country[Energy14$Country == "Northern Ireland"] <- "NI"

Energy56 <- food56 %>%
  select(ID, Country, Surveyyear, EnergykJ, Carbohydrateg,
         CHOpctotE, Proteing, ProteinpctotE, Alcoholg, AlcoholpctotE,
         Fatg, FatpctotE) %>%
  rename(SurveyYear = Surveyyear)

Energy78 <- food78 %>%
  select(ID, Country, SurveyYear, EnergykJ, Carbohydrateg,
         CHOpctotE, Proteing, ProteinpctotE, Alcoholg, AlcoholpctotE,
         Fatg, FatpctotE)

Energy <- bind_rows(Energy14, Energy56, Energy78)



dta_NDNS <- read_csv("dta_NDNS_Tslots.csv") # extract the day of week data


CW3idday$DayNo <- ave(CW3idday$ID_DAY, CW3idday$ID,
                      FUN = seq_along) # adding the day no

dta_DayofWeek <- dta_NDNS %>%
  select(id, id_dy, DayofWeek)

dta_DayofWeek$DayNo <- unlist(strsplit(dta_DayofWeek$id_dy,
                                       "D"))[c(FALSE, TRUE)] # creating a day No
names(dta_DayofWeek)[1] <- "ID"
dta_DayofWeek$DayNo <- as.numeric(dta_DayofWeek$DayNo)


CW3idday <-   CW3idday %>%
  left_join(dta_DayofWeek, by= c("ID", "DayNo"))

## Manually check the day of week
CW3idday$DayofWeek[CW3idday$ID_DAY == 40714261000] <- "Saturday"
CW3idday$DayofWeek[CW3idday$ID_DAY == 112050710000] <- "Tuesday"
CW3idday$DayofWeek[CW3idday$ID_DAY == 310122510000] <- "Wednesday"
CW3idday$DayofWeek[CW3idday$ID_DAY == 5050616100] <- "Monday"
CW3idday$DayofWeek[CW3idday$ID_DAY == 50506161000] <- "Tuesday"
```

```r
CW3idday$DayofWeek[CW3idday$ID_DAY == 505061610000] <- "Wednesday"
CW3idday$DayofWeek[CW3idday$ID_DAY == 7090824100] <- "Monday"
CW3idday$DayofWeek[CW3idday$ID_DAY == 70908241000] <- "Tuesday"
CW3idday$DayofWeek[CW3idday$ID_DAY == 709082410000] <- "Wednesday"
CW3idday$DayofWeek[CW3idday$ID_DAY == 807021910000] <- "Monday"
CW3idday$DayofWeek[CW3idday$ID_DAY == 812101310000] <- "Sunday"


CW3CB3_reg <- CW3CB3[!duplicated(CW3CB3$ID), ]

# extract only the CB variable (Between individual classes == 1 or 2)
CW3CB3_reg <- CW3CB3_reg %>%
  select(ID, AGE, SEX, CB)

tab1(CW3CB3_reg$CB, graph = FALSE)


## dataset for 3by3 multilevel latent classes
CW3CB3_regss <- CW3CB3_reg %>%
  left_join(BMI, by = "ID") %>%
  left_join(Energy, by = "ID")

# Rescale the weighting values
# individual weighting

a <- sum(CW3CB3_regss[CW3CB3_regss$Years == "1-4",]$wti)
b <- sum(CW3CB3_regss[CW3CB3_regss$Years == "5-6",]$wti)
c <- sum(CW3CB3_regss[CW3CB3_regss$Years == "7-8",]$wti)

CW3CB3_regss$wti1to8 <- CW3CB3_regss$wti

CW3CB3_regss[CW3CB3_regss$Years == "1-4",]$wti1to8 <-
  CW3CB3_regss[CW3CB3_regss$Years == "1-4",]$wti*(a+b+c)*(1/2)/a
CW3CB3_regss[CW3CB3_regss$Years == "5-6",]$wti1to8 <-
  CW3CB3_regss[CW3CB3_regss$Years == "5-6",]$wti*(a+b+c)*(1/4)/b
CW3CB3_regss[CW3CB3_regss$Years == "7-8",]$wti1to8 <-
  CW3CB3_regss[CW3CB3_regss$Years == "7-8",]$wti*(a+b+c)*(1/4)/c
mean(CW3CB3_regss$wti1to8)

CW3CB3_regss$wti1to8 <- CW3CB3_regss$wti1to8/1.209816814
summ(CW3CB3_regss$wti1to8, graph = FALSE)
#Check if the weighting sum up to the sample size we have
sum(CW3CB3_regss$wti1to8, graph = FALSE)

# Nurse weights

a <- sum(CW3CB3_regss[CW3CB3_regss$Years == "1-4",]$wtn)
b <- sum(CW3CB3_regss[CW3CB3_regss$Years == "5-6",]$wtn)
c <- sum(CW3CB3_regss[CW3CB3_regss$Years == "7-8",]$wtn)

CW3CB3_regss$wtn1to8 <- CW3CB3_regss$wtn

CW3CB3_regss[CW3CB3_regss$Years == "1-4",]$wtn1to8 <-
```

```
  CW3CB3_regss[CW3CB3_regss$Years == "1-4",]$wtn*(a+b+c)*(1/2)/a
CW3CB3_regss[CW3CB3_regss$Years == "5-6",]$wtn1to8 <-
  CW3CB3_regss[CW3CB3_regss$Years == "5-6",]$wtn*(a+b+c)*(1/4)/b
CW3CB3_regss[CW3CB3_regss$Years == "7-8",]$wtn1to8 <-
  CW3CB3_regss[CW3CB3_regss$Years == "7-8",]$wtn*(a+b+c)*(1/4)/c
mean(CW3CB3_regss$wtn1to8)

CW3CB3_regss$wtn1to8 <- CW3CB3_regss$wtn1to8/0.907003577
summ(CW3CB3_regss$wtn1to8, graph = FALSE)
#Check if the weighting sum up to the sample size we have
sum(CW3CB3_regss$wtn1to8, graph = FALSE)

# Blood weights
a <- sum(CW3CB3_regss[CW3CB3_regss$Years == "1-4",]$wtb)
b <- sum(CW3CB3_regss[CW3CB3_regss$Years == "5-6",]$wtb)
c <- sum(CW3CB3_regss[CW3CB3_regss$Years == "7-8",]$wtb)

CW3CB3_regss$wtb1to8 <- CW3CB3_regss$wtb

CW3CB3_regss[CW3CB3_regss$Years == "1-4",]$wtb1to8 <-
  CW3CB3_regss[CW3CB3_regss$Years == "1-4",]$wtb*(a+b+c)*(1/2)/a
CW3CB3_regss[CW3CB3_regss$Years == "5-6",]$wtb1to8 <-
  CW3CB3_regss[CW3CB3_regss$Years == "5-6",]$wtb*(a+b+c)*(1/4)/b
CW3CB3_regss[CW3CB3_regss$Years == "7-8",]$wtb1to8 <-
  CW3CB3_regss[CW3CB3_regss$Years == "7-8",]$wtb*(a+b+c)*(1/4)/c
mean(CW3CB3_regss$wtb1to8)

CW3CB3_regss$wtb1to8 <- CW3CB3_regss$wtb1to8/0.4817444505
summ(CW3CB3_regss$wtb1to8, graph = FALSE)
#Check if the weighting sum up to the sample size we have
sum(CW3CB3_regss$wtb1to8, graph = FALSE)


weightings <- CW3CB3_regss %>% select(ID, wti1to8, wtn1to8, wtb1to8)


# Combine the data from nutrient intake at each time slot
CW3CB3_7regss <- CW3CB3_7regss %>%
  left_join(IntakeSlots, by = "ID")

## dta ready to be analysed in STATA
 # change the path accordingly
write_dta(CW3CB3_regss, "../CW3CB3_7regss.dta")

************************************************
// Analysing NDNS survey data in stata
// for CW3CB3 survey data analysis
// date created: 2018-08-01
// manipulation of the data was done in R
// import data from CW3CB3_7sregss.dta
// change the path accordingly
************************************************
```

```
use "../CW3CB3_7regss.dta", clear

label define smoking 1 "current" 2 "ex-smoker" 3 "Never"
label values cigsta3 smoking
label define gender 1 "Men" 2 "Women"
label values Sex gender
label define paid 1 "No" 2 "Yes"
label values paidemployment paid
label define ethnicity5 1 "White" 2 "Mixed" 3 "Black" 4 "Asian" 5 "Other"
label values ethgrp5 ethnicity5
label define ethnicity2 1 "White" 2 "non-White"
label values ethgrp2 ethnicity2

gen Married = 1 if MarStat == 2 | MarSt2 == 2
replace Married = 1 if MarSt2 == 3
replace Married = 0 if Married !=1
tab Married
tab MarSt2
tab MarStat

label define Partner 0 "No" 1 "Yes"
label values Married Partner

gen Education = qual7 == 1
label define Ed 0 "lower than Degree" 1 "Degree or higher"
label values Education Ed

replace Education = . if qual7 >100
tab Educ


egen BMIcat = cut(bmival), at(10, 25, 30, 40, 60)
tab BMIcat

**********************************************************
// variables need to be log transfomred                 //
**********************************************************

gen logalc = ln(Alcoholg+1)
summ logalc, detail
gen logMVP = ln(MVPAtime+1)
summ logMVP, detail
gen logGlu = ln(Glucose)
summ logGlu, detail
gen logA1C = ln(A1C)
summ logA1C, detail
gen logChol = ln(Chol)
summ logChol, detail
gen logLDL = ln(LDL)
gen logHDL = ln(HDL)
gen logTG = ln(Trig)
```

```
*******************************************************
//  weighting use wti to see the individual results    //
//                                                     //
*******************************************************


// weighting with individual weights, area is primary sampling unit,
// gor is the cluster variable

svyset area [pweight = wti1to8], strata(gor)

svydescribe wti // describe the weighted data set



svy: tabulate Sex CB, row se ci format(%7.3f)
svy: tabulate Sex CB, col se ci format(%7.3f)
svy: tabulate Country CB, col se ci format(%7.3f)
svy: tabulate Country CB, row se ci format(%7.3f)
svy: tabulate SurveyYear CB, col se ci format(%7.3f)
svy: tabulate SurveyYear CB, row se ci format(%7.3f)
svy: tabulate paid CB, col se ci format(%7.3f)

svy: tabulate MarSt2 CB
svy: tabulate MarStat CB
svy: tabulate Married CB, row se ci format(%7.3f)
svy: tabulate Married CB, col se ci format(%7.3f)
svy: mean eqvinc, over(CB)
test [eqvinc]1 = [eqvinc]2 = [eqvinc]3, mtest(b)
// bonferroni-adjusted p-values for multiple groups using the mtest(b) option

svy: tabulate ethgrp2 CB, row se ci format(%7.3f)
svy: tabulate ethgrp2 CB, col se ci format(%7.3f)
svy: tabulate Education CB, row se ci format(%7.3f)
svy: tabulate Education CB, col se ci format(%7.3f)

*******************************************************
//  nutritional distribution                           //
//                                                     //
*******************************************************


svy: mean EnergykJ, over(CB)
test [EnergykJ]1 = [EnergykJ]2 = [EnergykJ]3, mtest(b)

svy: mean Energy6, over(CB)
test [Energy6_9]1 = [Energy6_9]2 = [Energy6_9]3, mtest(b)

svy: mean Energy9, over(CB)
test [Energy9_12]1 = [Energy9_12]2 = [Energy9_12]3, mtest(b)

svy: mean Energy12, over(CB)
test [Energy12_14]1 = [Energy12_14]2 = [Energy12_14]3, mtest(b)
```

```
svy: mean Energy14, over(CB)
test [Energy14_17]1 = [Energy14_17]2 = [Energy14_17]3, mtest(b)

svy: mean Energy17, over(CB)
test [Energy17_20]1 = [Energy17_20]2 = [Energy17_20]3, mtest(b)

svy: mean Energy20, over(CB)
test [Energy20_22]1 = [Energy20_22]2 = [Energy20_22]3, mtest(b)

svy: mean Energy22, over(CB)
test [Energy22_6]1 = [Energy22_6]2 = [Energy22_6]3, mtest(b)


svy: mean Carbohydrateg, over(CB)
test [Carbohydrateg]1 = [Carbohydrateg]2 = [Carbohydrateg]3, mtest(b)

svy: mean Carb6, over(CB)
test [Carb6_9]1 = [Carb6_9]2 = [Carb6_9]3, mtest(b)

svy: mean Carb9, over(CB)
test [Carb9_12]1 = [Carb9_12]2 = [Carb9_12]3, mtest(b)

svy: mean Carb12, over(CB)
test [Carb12_14]1 = [Carb12_14]2 = [Carb12_14]3, mtest(b)

svy: mean Carb14, over(CB)
test [Carb14_17]1 = [Carb14_17]2 = [Carb14_17]3, mtest(b)

svy: mean Carb17, over(CB)
test [Carb17_20]1 = [Carb17_20]2 = [Carb17_20]3, mtest(b)


svy: mean Carb20, over(CB)
test [Carb20_22]1 = [Carb20_22]2 = [Carb20_22]3, mtest(b)

svy: mean Carb22, over(CB)
test [Carb22_6]1 = [Carb22_6]2 = [Carb22_6]3, mtest(b)

svy: mean Sugar6, over(CB)
test [Sugar6_9]1 = [Sugar6_9]2 = [Sugar6_9]3, mtest(b)



svy: mean Sugar9, over(CB)
test [Sugar9_12]1 = [Sugar9_12]2 = [Sugar9_12]3, mtest(b)

svy: mean Sugar12, over(CB)
test [Sugar9_12]1 = [Sugar9_12]2 = [Sugar9_12]3, mtest(b)

svy: mean Sugar14, over(CB)
test [Sugar14_17]1 = [Sugar14_17]2 = [Sugar14_17]3, mtest(b)

svy: mean Sugar17, over(CB)
test [Sugar17_20]1 = [Sugar17_20]2 = [Sugar17_20]3, mtest(b)
```

```
svy: mean Sugar20, over(CB)
test [Sugar20_22]1 = [Sugar20_22]2 = [Sugar20_22]3, mtest(b)

svy: mean Sugar22, over(CB)
test [Sugar22_6]1 = [Sugar22_6]2 = [Sugar22_6]3, mtest(b)

svy: mean Starch6, over(CB)
test [Starch6_9]1 = [Starch6_9]2 = [Starch6_9]3, mtest(b)

svy: mean Starch9, over(CB)
test [Sugar12_14]1 = [Sugar12_14]2 = [Sugar12_14]3, mtest(b)

svy: mean Starch12, over(CB)
test [Starch12_14]1 = [Starch12_14]2 = [Starch12_14]3, mtest(b)

svy: mean Starch14, over(CB)
test [Starch14_17]1 = [Starch14_17]2 = [Starch14_17]3, mtest(b)

svy: mean Starch17, over(CB)
test [Starch17_20]1 = [Starch17_20]2 = [Starch17_20]3, mtest(b)


svy: mean Starch20, over(CB)
test [Starch20_22]1 = [Starch20_22]2 = [Starch20_22]3, mtest(b)

svy: mean Starch22, over(CB)
test [Starch20_22]1 = [Starch20_22]2 = [Starch20_22]3, mtest(b)

svy: mean Fibre6, over(CB)
test [Starch22_6]1 = [Starch22_6]2 = [Starch22_6]3, mtest(b)

gen Fibreg = Fibre6 + Fibre9 + Fibre12 + Fibre14 + Fibre17 + Fibre20 + Fibre22
svy: mean Fibreg, over(CB)
test [Fibreg]1 = [Fibreg]2 = [Fibreg]3, mtest(b)


svy: mean Fibre9, over(CB)
test [Fibre9_12]1 = [Fibre9_12]2 = [Fibre9_12]3, mtest(b)


svy: mean Fibre12, over(CB)
test [Fibre12_14]1 = [Fibre12_14]2 = [Fibre12_14]3, mtest(b)

svy: mean Fibre14, over(CB)
test [Fibre14_17]1 = [Fibre14_17]2 = [Fibre14_17]3, mtest(b)

svy: mean Fibre17, over(CB)
test [Fibre17_20]1 = [Fibre17_20]2 = [Fibre17_20]3, mtest(b)

svy: mean Fibre20, over(CB)
test [Fibre20_22]1 = [Fibre20_22]2 = [Fibre20_22]3, mtest(b)

svy: mean Fibre22, over(CB)
```

```
test [Fibre22_6]1 = [Fibre22_6]2 = [Fibre22_6]3, mtest(b)

svy: mean NMES6, over(CB)
test [NMES6_9]1 = [NMES6_9]2 = [NMES6_9]3, mtest(b)

svy: mean NMES9, over(CB)
test [NMES9_12]1 = [NMES9_12]2 = [NMES9_12]3, mtest(b)


svy: mean NMES12, over(CB)
test [NMES12_14]1 = [NMES12_14]2 = [NMES12_14]3, mtest(b)

svy: mean NMES14, over(CB)
test [NMES14_17]1 = [NMES14_17]2 = [NMES14_17]3, mtest(b)

svy: mean NMES17, over(CB)
test [NMES17_20]1 = [NMES17_20]2 = [NMES17_20]3, mtest(b)


svy: mean NMES20, over(CB)
test [NMES20_22]1 = [NMES20_22]2 = [NMES20_22]3, mtest(b)

svy: mean NMES22, over(CB)


svy: mean CHO, over(CB)
test [CHOpctotE]1 = [CHOpctotE]2 = [CHOpctotE]3, mtest(b)


svy: mean Proteing, over(CB)
test [Proteing]1 = [Proteing]2 = [Proteing]3, mtest(b)

svy: mean Prot6, over(CB)
test [Prot6_9]1 = [Prot6_9]2 = [Prot6_9]3, mtest(b)

svy: mean Prot9, over(CB)
test [Prot9_12]1 = [Prot9_12]2 = [Prot9_12]3, mtest(b)

svy: mean Prot12, over(CB)
test [Prot12_14]1 = [Prot12_14]2 = [Prot12_14]3, mtest(b)

svy: mean Prot14, over(CB)
test [Prot14_17]1 = [Prot14_17]2 = [Prot14_17]3, mtest(b)

svy: mean Prot17, over(CB)
test [Prot17_20]1 = [Prot17_20]2 = [Prot17_20]3, mtest(b)

svy: mean Prot20, over(CB)
test [Prot20_22]1 = [Prot20_22]2 = [Prot20_22]3, mtest(b)

svy: mean Prot22, over(CB)
test [Prot22_6]1 = [Prot22_6]2 = [Prot22_6]3, mtest(b)
```

```
svy: mean Proteinp, over(CB)
test [ProteinpctotE]1 = [ProteinpctotE]2 = [ProteinpctotE]3, mtest(b)

svy: mean Fatg, over(CB)
test [Fatg]1 = [Fatg]2 = [Fatg]3, mtest(b)


svy: mean Fat6, over(CB)
test [Fat6_9]1 = [Fat6_9]2 = [Fat6_9]3, mtest(b)

svy: mean Fat9, over(CB)
test [Fat9_12]1 = [Fat9_12]2 = [Fat9_12]3, mtest(b)


svy: mean Fat12, over(CB)
test [Fat12_14]1 = [Fat12_14]2 = [Fat12_14]3, mtest(b)

svy: mean Fat14, over(CB)
test [Fat14_17]1 = [Fat14_17]2 = [Fat14_17]3, mtest(b)

svy: mean Fat17, over(CB)
test [Fat17_20]1 = [Fat17_20]2 = [Fat17_20]3, mtest(b)

svy: mean Fat20, over(CB)
test [Fat20_22]1 = [Fat20_22]2 = [Fat20_22]3, mtest(b)

svy: mean Fat22, over(CB)
test [Fat22_6]1 = [Fat22_6]2 = [Fat22_6]3, mtest(b)



svy: mean Fatp, over(CB)
test [FatpctotE]1 = [FatpctotE]2 = [FatpctotE]3, mtest(b)

svy: mean Alcoholg, over(CB)
test [Alcoholg]1 = [Alcoholg]2 = [Alcoholg]3, mtest(b)


svy: mean Alc6, over(CB)
test [Alc6_9]1 = [Alc6_9]2 = [Alc6_9]3, mtest(b)


svy: mean Alc9, over(CB)
test [Alc9_12]1 = [Alc9_12]2 = [Alc9_12]3, mtest(b)

svy: mean Alc12, over(CB)
test [Alc12_14]1 = [Alc12_14]2 = [Alc12_14]3, mtest(b)

svy: mean Alc14, over(CB)
test [Alc14_17]1 = [Alc14_17]2 = [Alc14_17]3, mtest(b)

svy: mean Alc17, over(CB)
test [Alc14_17]1 = [Alc14_17]2 = [Alc14_17]3, mtest(b)
```

```
svy: mean Alc20, over(CB)
test [Alc14_17]1 = [Alc14_17]2 = [Alc14_17]3, mtest(b)


svy: mean Alcoholp, over(CB)
test [AlcoholpctotE]1 = [AlcoholpctotE]2 = [AlcoholpctotE]3, mtest(b)


svy: tabulate cigsta3 CB, col se ci format(%7.3f)
svy: tabulate dnnow CB, col se ci format(%7.3f)
svy: tabulate hibp CB, col se ci format(%7.3f)


sum MVP [weight=wti1to8] if CB ==1 , det
sum MVP [weight=wti1to8] if CB ==2 , det
sum MVP [weight=wti1to8] if CB ==3 , det
svy: mean MVP, over(CB)

svy: mean logMVP, over(CB) eform
test [logMVP]1 = [logMVP]2 = [logMVP]3, mtest(b)


disp exp(.731059) - 1
dis exp(.6768489) -1
dis exp(.7852691) -1

disp exp(.6239265) - 1
dis exp(.571165) -1
dis exp( .6766879) -1

disp exp(.7273621) - 1
dis exp(.684545) -1
dis exp(.7701791) -1

svy: mean logalc, over(CB)


disp exp( 2.035795) - 1
dis exp(1.933326) -1
dis exp(2.138264) -1



*********************************************************
//  re-weighting use wtn to see the BMI,wc measurements //
//                                                      //
*********************************************************

svyset area [pweight = wtn1to8], strata(gor)
svy: mean wst, over(CB)
test [wstval]1 = [wstval]2 = [wstval]3, mtest(b)

gen Men = Sex == 1
svy, subpop(Men): mean wst, over(CB)
```

```
test [wstval]1 = [wstval]2 = [wstval]3, mtest(b)

gen Women = Sex == 2
svy, subpop(Women): mean wst, over(CB)
test [wstval]1 = [wstval]2 = [wstval]3, mtest(b)



svy: mean bmi, over(CB)

test [bmival]1 = [bmival]2 = [bmival]3, mtest(b)


*******************************************************
//  re-weighting use wtb to see the blood test results //
//                                                     //
*******************************************************

svyset area [pweight = wtb1to8], strata(gor)


svy: mean HDL, over(CB)
test [HDL]1 = [HDL]2 = [HDL]3, mtest(b)

svy: mean Chol, over(CB)
test [Chol]1 = [Chol]2 = [Chol]3, mtest(b)

svy: mean LDL, over(CB)
test [LDL]1 = [LDL]2 = [LDL]3, mtest(b)

svy: mean Trig, over(CB)
test [Trig]1 = [Trig]2 = [Trig]3, mtest(b)


gen DM = A1C <= 6.5 if !missing(A1C)


svy, subpop(DM): mean Glucose, over(CB)
test [Glucose]1 = [Glucose]2
test [Glucose]1 = [Glucose]2 = [Glucose]3, mtest(b)

svy, subpop(DM): mean A1C, over(CB)
test [A1C]1 = [A1C]2
test [A1C]1 = [A1C]2 = [A1C]3, mtest(b)

svy: tabulate DM CB, col se ci format(%7.3f)




svy, subpop(DM): mean Glucose, over(CB)
test [Glucose]1 = [Glucose]2
test [Glucose]1 = [Glucose]2 = [Glucose]3, mtest(b)
svy, subpop(DM): mean A1C, over(C)
```

```
test [A1C]1 = [A1C]2
test [A1C]1 = [A1C]2 = [A1C]3, mtest(b)

svy: tabulate DM C, col se ci format(%7.3f)



svy, subpop(DM): mean logGlu, over(CB)
test [logGlu]1 = [logGlu]2 = [logGlu]3, mtest(b)


dis exp(1.642848)
dis exp(1.632226)
dis exp(1.653471)

dis exp(1.620347)
dis exp(1.606447)
dis exp(1.634246)


dis exp(1.629356)
dis exp(1.620271)
dis exp(1.63844)


svy, subpop(DM): mean logA1C, over(CB)
test [logA1C]1 = [logA1C]2 = [logA1C]3, mtest(b)

dis exp(1.699581)
dis exp(1.69296)
dis exp(1.706203)


dis exp(1.691608)
dis exp(1.683897)
dis exp(1.699318)


dis exp(1.705623)
dis exp(1.700665)
dis exp(1.710581)


svy: mean logChol, over(CB)
dis exp(1.598818)
dis exp(1.577698)
dis exp(1.619939)


dis exp(1.55251)
dis exp(1.530408)
dis exp(1.574613)

dis exp(1.599389)
```

```
dis exp(1.583391)
dis exp(1.615388)


test [logChol]1 = [logChol]2 = [logChol]3, mtest(b)




svy: mean logHDL, over(CB)
dis exp(.3293169)
dis exp(.3026793)
dis exp(.3559545)


dis exp(.2749379)
dis exp(.2476816)
dis exp(.3021941)

dis exp(.3269002)
dis exp(.3062623)
dis exp(.3475381)


test [logHDL]1 = [logHDL]2 = [logHDL]3, mtest(b)



svy: mean logLDL, over(CB)
dis exp(1.058635)
dis exp(1.028391)
dis exp(1.08888)


dis exp(1.018181)
dis exp(.984431)
dis exp(1.051931)

dis exp(1.075229)
dis exp(1.051369)
dis exp(1.09909)


test [logLDL]1 = [logLDL]2 = [logLDL]3, mtest(b)


svy: mean logTG, over(CB)
dis exp(.1273876)
dis exp(.0777152)
dis exp(.17706)


dis exp(.1012169)
dis exp(.0460972)
```

```
dis exp(.1563366)

dis exp(.0983298)
dis exp(.0607423)
dis exp(.1359172)


test [logTG]1 = [logTG]2 = [logTG]3, mtest(b)

***************************************************
// Analysing NDNS survey data in stata
// for CW3CB3 survey data analysis on hypertension
// date created: 2018-08-06
// manipulation of the data was done in R
// import data from CW3CB3_7sregss.dta
// change the path accordingly
**************************************************


use "../CW3CB3_7regss.dta", clear


label define smoking 1 "current" 2 "ex-smoker" 3 "Never"
label values cigsta3 smoking
label define gender 1 "Men" 2 "Women"
label values Sex gender
label define paid 1 "No" 2 "Yes"
label values paidemployment paid
label define ethnicity5 1 "White" 2 "Mixed" 3 "Black" 4 "Asian" 5 "Other"
label values ethgrp5 ethnicity5
label define ethnicity2 1 "White" 2 "non-White"
label values ethgrp2 ethnicity2

gen Married = 1 if MarStat == 2 | MarSt2 == 2
replace Married = 1 if MarSt2 == 3
replace Married = 0 if Married !=1
tab Married
tab MarSt2
tab MarStat

label define Partner 0 "No" 1 "Yes"
label values Married Partner

gen Education = qual7 == 1
label define Ed 0 "lower than Degree" 1 "Degree or higher"
label values Education Ed

replace Education = . if qual7 >100
tab Educ


egen BMIcat = cut(bmival), at(10, 25, 30, 40, 60)
tab BMIcat
```

```
********************************************************
// variables need to be log transfomred                //
********************************************************

gen logalc = ln(Alcoholg+1)
summ logalc, detail
gen logMVP = ln(MVPAtime+1)
summ logMVP, detail
gen logGlu = ln(Glucose)
summ logGlu, detail
gen logA1C = ln(A1C)
summ logA1C, detail
gen logChol = ln(Chol)
summ logChol, detail
gen logLDL = ln(LDL)
gen logHDL = ln(HDL)
gen logTG = ln(Trig)




*******************************************************
//  weighting use wti to see the individual results    //
//                                                     //
*******************************************************


// weighting with individual weights, area is primary sampling unit,
// gor is the cluster variable
svyset area [pweight = wti1to8], strata(gor)

svydescribe wti // describe the weighted data set


*******************************************************
//  re-weighting use wtn to see the BMI,wc measurements //
//                                                     //
*******************************************************

svyset area [pweight = wtn1to8], strata(gor)


gen Men = Sex == 1  //  n of men = 2537
gen Women = Sex == 2 // n of women = 3618

svy, subpop(Men): tab hibp, se ci format(%7.3f)
svy, subpop(Women): tab hibp, se ci format(%7.3f)

svy, subpop(Men): mean age, over(hibp)
test [age]1 = [age]0
svy, subpop(Women): mean age, over(hibp)
test [age]1 = [age]0
```

```
svy, subpop(Men): mean wst, over(hibp)
test [wstval]1 = [wstval]0
svy, subpop(Women): mean wst, over(hibp)
test [wstval]1 = [wstval]0

svy, subpop(Men): tabulate CB hibp, col se ci format(%7.3f)

svy, subpop(Women): tabulate CB hibp, col se ci format(%7.3f)


svy, subpop(Men): tabulate Country hibp, col se ci format(%7.3f)

svy, subpop(Women): tabulate Country hibp, col se ci format(%7.3f)

svy, subpop(Men): tabulate SurveyYear hibp, col se ci format(%7.3f)

svy, subpop(Women): tabulate SurveyYear hibp, col se ci format(%7.3f)

svy, subpop(Men): tabulate ethgrp2 hibp, col se ci format(%7.3f)

svy, subpop(Women): tabulate ethgrp2 hibp, col se ci format(%7.3f)


svy, subpop(Men): tabulate Edu hibp, col se ci format(%7.3f)

svy, subpop(Women): tabulate Edu hibp, col se ci format(%7.3f)

svy, subpop(Men): tabulate cigsta3 hibp, col se ci format(%7.3f)

svy, subpop(Women): tabulate cigsta3 hibp, col se ci format(%7.3f)


svy, subpop(Men): tabulate Married hibp, col  se ci format(%7.3f)

svy, subpop(Men): mean logMVP, over(hibp)
test [logMVP]1 = [logMVP]0


disp exp(.9234363 ) - 1
dis exp(.8457101) -1
dis exp(1.001163) -1

disp exp(.828635) - 1
dis exp(.730244) -1
dis exp( .9270261) -1


svy, subpop(Women): mean logMVP, over(hibp)
test [logMVP]1 = [logMVP]0


disp exp(.5916676) - 1
dis exp(.5473043) -1
dis exp(.6360309) -1
```

```
disp exp(.4231103) - 1
dis exp(.3536885) -1
dis exp(.4925322) -1




svy: mean logalc, over(CB)


disp exp( 2.035795) - 1
dis exp(1.933326) -1
dis exp(2.138264) -1




svy, subpop(Men): mean bmi, over(hibp)
test [bmival]1 = [bmival]0
svy, subpop(Women): mean bmi, over(hibp)
test [bmival]1 = [bmival]0


svy, subpop(Men): mean EnergykJ, over(hibp)
test [EnergykJkJ]1 = [EnergykJkJ]0
svy, subpop(Women): mean EnergykJ, over(hibp)
test [EnergykJkJ]1 = [EnergykJkJ]0


svy, subpop(Men): mean Carbo, over(hibp)
test [Carbohydrateg]1 = [Carbohydrateg]0

svy, subpop(Women): mean Carbohydrateg, over(hibp)
test [Carbohydrateg]1 = [Carbohydrateg]0


svy, subpop(Men): mean Proteing, over(hibp)
test [Proteing]1 = [Proteing]0

svy, subpop(Women): mean Carbohydrateg, over(hibp)
test [Carbohydrateg]1 = [Carbohydrateg]0



svy: tabulate Sex hibp, col se ci format(%7.3f)

svy: logistic hibp i.CB
svy: logistic hibp i.CB#i.Sex

test 2.CB#2.Sex

svy: tabulate paid hibp, col se ci format(%7.3f)
```

```
gen DM = A1C > 6.5 if !missing(A1C)
svyset area [pweight = wtb1to8], strata(gor)

svy, subpop(Men): tabulate DM hibp, col se ci format(%7.3f)
svy, subpop(Women): tabulate DM hibp, col se ci format(%7.3f)


svy, subpop(Men): tabulate Married hibp, col se ci format(%7.3f)
svy, subpop(Women): tabulate Married hibp, col se ci format(%7.3f)

svy, subpop(Men): mean eqvinc, over(hibp)
test [eqvinc]1 = [eqvinc]0

svy, subpop(Women): mean eqvinc, over(hibp)
test [eqvinc]1 = [eqvinc]0


********************************************************
**   Building the GLM model
**   date: 07/08/2018
**
**
********************************************************

svyset area [pweight = wtn1to8], strata(gor)


// crude association between CB and hypertension

svy, subpop(Men): logistic hibp i.CB

svy, subpop(Women): logistic hibp i.CB

// in non DM
svy, subpop(Men if DM != 1): logistic hibp i.CB
svy, subpop(Women if DM != 1): logistic hibp i.CB


// looking for confounders one by one
// Age: -> confounder
svy, subpop(Men): logistic hibp i.CB age
test age
svy, subpop(Women): logistic hibp i.CB age
test age

svy, subpop(Men): logistic hibp i.CB##c.age
svy, subpop(Women): logistic hibp i.CB##c.age
test 2.CB#c.age 3.CB#c.age // no interaction

// Partner -> confounder
svy, subpop(Men): logistic hibp i.CB i.Married
test 1.Married
```

```
svy, subpop(Women): logistic hibp i.CB i.Married
test 1.Married

svy, subpop(Men): logistic hibp i.CB##i.Married
svy, subpop(Women): logistic hibp i.CB##i.Married

test 2.CB#1.Married 3.CB#1.Married // -> no interaction


// Income -> not confounder for men but confounder for women
svy, subpop(Men): logistic hibp i.CB eqvinc
test eqvinc
svy, subpop(Women): logistic hibp i.CB eqvinc
test eqvinc

svy, subpop(Men): logistic hibp i.CB##c.eqvinc
svy, subpop(Women): logistic hibp i.CB##c.eqvinc

test 2.CB#c.eqvinc 3.CB#c.eqvinc // -> (probably) no interaction


// Education -> confounder
svy, subpop(Men): logistic hibp i.CB i.Edu
test 1.Edu
svy, subpop(Women): logistic hibp i.CB i.Edu
test 1.Edu

svy, subpop(Men): logistic hibp i.CB##i.Edu
svy, subpop(Women): logistic hibp i.CB##i.Edu

test 2.CB#1.Edu 3.CB#1.Edu // no interaction

// BMI -> confounder
svy, subpop(Men): logistic hibp i.CB bmi
test bmi
svy, subpop(Women): logistic hibp i.CB bmi
test bmi

svy, subpop(Men): logistic hibp i.CB##c.bmi
svy, subpop(Women): logistic hibp i.CB##c.bmi

test 2.CB#c.bmival 3.CB#c.bmival // no ineraction

// paid employment -> not confounder
svy, subpop(Men): logistic hibp i.CB i.paid
test 2.paid
svy, subpop(Women): logistic hibp i.CB i.paid
test 2.paid

svy, subpop(Men): logistic hibp i.CB##i.paid
svy, subpop(Women): logistic hibp i.CB##i.paid

test 2.CB#2.paid 3.CB#2.paid
```

```
// Smoking -> confounder
svy, subpop(Men): logistic hibp i.CB i.cigsta3
test 2.cigsta3 3.cigsta3
svy, subpop(Women): logistic hibp i.CB i.cigsta3
test 2.cigsta3 3.cigsta3

svy, subpop(Men): logistic hibp i.CB##i.cigsta3
svy, subpop(Women): logistic hibp i.CB##i.cigsta3

test 2.CB#2.cigsta3 2.CB#3.cigsta3 3.CB#2.cigsta3 3.CB#3.cigsta3 // no interaction


// Total energy intake -> confounder
svy, subpop(Men): logistic hibp i.CB EnergykJ
test EnergykJ
svy, subpop(Women): logistic hibp i.CB EnergykJ
test EnergykJ

svy, subpop(Men): logistic hibp i.CB##c.EnergykJ
svy, subpop(Women): logistic hibp i.CB##c.EnergykJ
test 2.CB#c.EnergykJkJ 3.CB#c.EnergykJkJ // no interaction


// ethnicity -> not confounder
svy, subpop(Men): logistic hibp i.CB i.ethgrp2
test 2.eth
svy, subpop(Women): logistic hibp i.CB i.ethgrp2
test 2.eth

svy, subpop(Men): logistic hibp i.CB##i.ethgrp2
svy, subpop(Women): logistic hibp i.CB##i.ethgrp2
test 2.CB#2.ethgrp2 // no interaction


// Alcohol -> not confounder for men but confounder for women
svy, subpop(Men): logistic hibp i.CB Alcoholg
test Alcoholg
svy, subpop(Women): logistic hibp i.CB Alcoholg
test Alcoholg

svy, subpop(Men): logistic hibp i.CB##c.Alcoholg
svy, subpop(Women): logistic hibp i.CB##c.Alcoholg
test 2.CB#c.Alcoholg 3.CB#c.Alcoholg // no interaction

// logMVP -> not confounder
svy, subpop(Men): logistic hibp i.CB logMVP
test logMVP
svy, subpop(Women): logistic hibp i.CB logMVP
test logMVP

svy, subpop(Men): logistic hibp i.CB##c.logMVP
svy, subpop(Women): logistic hibp i.CB##c.logMVP
test 2.CB#c.logMVP 3.CB#c.logMVP // no interaction
```

```
//  Preliminary model includes all possible confounders in Men

gen age2 = age^2

svy, subpop(Men): logistic hibp i.CB age i.Married i.Edu bmi i.cig EnergykJ
linktest
svy, subpop(Men): logistic hibp i.CB age i.Married i.Edu wst i.cig EnergykJ
linktest
svy, subpop(if Men & DM != 1): logistic hibp i.CB age i.Married i.Edu bmi i.cig EnergykJ
linktest
svy, subpop(if Men & DM != 1): logistic hibp i.CB age i.Married i.Edu wst i.cig EnergykJ
linktest

//  Preliminary model includes all possible confounders in Women

svy, subpop(Women): logistic hibp i.CB age i.Married eqvinc i.Edu bmi i.cig EnergykJ Alcoholg
linktest
svy, subpop(Women): logistic hibp i.CB age i.Married eqvinc i.Edu wst i.cig EnergykJ Alcoholg
linktest

svy, subpop(if Women & DM != 1): logistic hibp i.CB age i.Married eqvinc i.Edu bmi i.cig EnergykJ Alcoh
linktest

svy, subpop(if Women & DM != 1): logistic hibp i.CB age i.Married eqvinc i.Edu wst i.cig EnergykJ Alcoh
linktest
```