```
* 2) Create a LIBNAME statement that points to 'XXX';

LIBNAME dclaims "D:\SASProgrammingPractice\Hands-on-session-3";

*3.Create a new data step and SET the permanent SAS data set named
"demptclaims;

DATA dptclaims;
SET dclaims.demptclaims;
RUN;

* 4) Sort the data set by the variable studyid and dos_from (the identifier
and date of service for the claim);

PROC SORT DATA=dptclaims;
BY studyid dos_from;
RUN;

*5.Create a new data step.  SET the previous data set with the BY statement
and keep the firstobservation;

DATA dptc;
SET dptclaims;
BY studyid dos_from;

*When using BY statement in DATA step, the First.variable will give the first
occurrence of a new value for the
variable studyid a value of 1, and 0 for others. The Last.variable will give
the last occurrence of a new value
for that variable a value of 1, and 0 for others.;

* 6) Create the following new variables.
      - Age from january 1, 1998 using the variable dob and the MDY function
        - Age groups of 45-64, 65-84, >=85 using IF-THEN-ELSE IF statements
      - Create a delirium diagnosis from the 5 diagnosis variables, dx_1-
dx_5, using
        IF-THEN-ELSE IF statements and the SUBSTR function. The first three
characters
        of the 5 character ICD-9 codes used for delirium are: 290, 203, 293,
291, 292.;

IF first.studyid;
*PROC PRINT;
*RUN;

*creating new variables: studyd, age, age1;
      studyd = MDY(1,1,1998);
      age = (dos_from - dob)/365.25;
      age1 = (studyd - dob)/365.25;

IF 45<=age<65 THEN agegroup=1;
ELSE IF 65<=age<84 THEN agegroup=2;
ELSE IF age>=85 THEN agegroup=3;

IF substr(dx 1,1,3) IN ('290','203','293','291','292') OR
substr(dx_2,1,3) IN ('290','203','293','291','292') OR
substr(dx_3,1,3) IN ('290','203','293','291','292') OR
```

```
substr(dx_4,1,3) IN ('290','203','293','291','292') OR
substr(dx_5,1,3) IN ('290','203','293','291','292') THEN delirium=1;
ELSE delirium=0;

FORMAT studyd dob dos_from mmddyy10.;
KEEP studyid studygrp dos_from studyd dob age age1 agegroup provspec provtype
dx 1 dx_2 dx_3 dx_4 dx_5 delirium;
RUN;

* 7) Print all the variables from the above data. (100 observations only). IS
your
     calculation correct?;

PROC PRINT DATA=dptc (obs=100);
VAR studyid studygrp dos_from studyd dob age age1 agegroup dx_1 dx_2 dx_3
dx 4 dx_5 delirium;
RUN;

* 8) PROC the FREQ of delirium with agegroup;

PROC FREQ DATA=dptc;
TABLES delirium agegroup;
RUN;


* import the excel file "Cord" in the handson folder into a temporary SAS
file Cord.
Using SET to create another SAS file, and do the following edit to the data;
* The variable names have to be edited before import;

PROC IMPORT OUT=WORK.cord
DATAFILE= "D:\SASProgrammingPractice\Hands-on-session-3\cord.xls"
DBMS=EXCEL REPLACE;
RANGE="Sheet2$";
GETNAMES=YES;
MIXED=NO;
SCANTEXT=YES;
USEDATE=YES;
SCANTIME=YES;
RUN;

DATA cord1;
set cord;
*systolic = substr(maternalBP,1,3);
*diastolic = substr(maternalBP,5,2);

* 1) calculate the age of each subject at collection date;


* 2) Use SAS function to seperate character variable "maternalBP" into the
numeric
variable "systolic" and "diastolic";

*scan will scan through the maternalBP from the left of the first character
and stop at the delimiter / and store it
in new variable systolic;
```

```
*scan will scan through the maternalBP from the right of the last character
and stop at the delimiter / and store it
in new variable diastolic;

systolic = SCAN(maternalBP,1,'/');
diastolic = SCAN(maternalBP,-1,'/');

* 3) Use SAS function to seperate character variable "placentalshape" into
the numeric
variable "P_long", "P_wide", and "P_thin";

p_long = SCAN(placentalshape, 1, '*');
p_wide = SCAN(placentalshape, 2, '*');
p_thin = SCAN(placentalshape, 3, '*');

* 4) Use SAS function to change the units of the character variable
"CordLength" into cm
and make it as a numeric variable;

cordlength_n=SCAN(cordlength,1,' ');
if SCAN(cordLength,-1, ' ') = 'cm' then c_Length_cm = CordLength_n*1;
else if SCAN(CordLength,-1,' ')= 'm' then c_Length_cm=CordLength_n*100;
else if SCAN(CordLength,-1,' ')= 'mm' then c_Length_cm=CordLength_n/10;
*proc contents;
*run;
PROC PRINT;
RUN;
```