```
/*************************************************************************
*******************/
/* Program Name:     pulkit.jain_HW11.sas
                                          */
/* Program Location: C:\Users\Pulkit
Jain\Documents\sasuniversityedition\myfolders\assign11 */
/* Date Created:     10/24/2017
                                          */
/* Author:              Pulkit Jain
                                       */
/* Purpose:             Assignment 11, practice creating variables
conditionally                  */
/*************************************************************************
*******************/

/* 1 Create two libname statements;                                    */
/* Assign library to locaion of hw data with access only; */
/* Assign another library with read and write access;     */

libname hw_data '/folders/myfolders/hw_data' access=readonly;
libname pulkit11 '/folders/myfolders/assign11';

/* Specify a fileref to designate output of pdf */

filename HW11 '/folders/myfolders/assign11/pulkit.jain_HW11_output.pdf';

/* 2 Use Jobs2017 data as input      */
/* Create temporary dataset "narrow" */
/* Only contain the variables Sector, state, month, year, and jobs */


data work.narrow;
   set hw_data.jobs2017;
      length month $ 12;
       * Make sure month doesn't truncates;
      if sector = 'PROFESSIONAL AND BUSINESS SERVICES' then
         sector = 'PROFESSIONAL/BUSINESS SERVICES';             * 2b
Convert the entry in sector;
        sector = propcase(sector);                                      *
2c Change variable sector to propercase;
             month = 'August';
            * 2d Hard code block of code for month info;
             year = '2016';
             if aug__2016 ne '' then jobs = aug__2016;              *
Delete observation if no jobs in month;
             else delete;
        output;
             month = 'September';
             year = '2016';
             jobs = sept__2016;
        output;
             month = 'October';
             year = '2016';
            jobs = oct__2016;
```

```
        output;
              month = 'November';
              year = '2016';
            jobs = nov__2016;
        output;
              month = 'December';
              year = '2016';
            jobs = dec__2016;
        output;
              month = 'January';
              year = '2017';
            jobs = jan__2017;
        output;
              month = 'February';
              year = '2017';
            jobs = feb__2017;
        output;
              month = 'March';
              year = '2017';
            jobs = mar__2017;
        output;
              month = 'April';
              year = '2017';
            jobs = apr__2017;
        output;
              month = 'May';
              year = '2017';
            jobs = may_2017;
        output;
              month = 'June';
              year = '2017';
            jobs = june_2017;
        output;
            month = 'July';
              year = '2017';
            jobs = july_2017;
        output;
            month = 'August';
              year = '2017';
            jobs = aug__2017;
        output;
        keep sector state month year jobs;                              * 2a
Specify which variables to retain;
        format jobs 6.1;
            * Decimal format for jobs;
run;

/* 3 Create 6 new datasets from monthly_jobs1617 data set */

data work.mrkt_small (KEEP = sector state avg_jobs)
       work.mrkt_med (KEEP = sector state avg_jobs)
       work.mrkt_large (KEEP = sector state avg_jobs)
       work.government (KEEP = state avg_jobs market_size)
       work.goods (KEEP = sector state avg_jobs market_size)
```

```
       work.services (KEEP = sector state avg_jobs market_size);


                  * specify names & variables of all 6 data outputs;
      set hw_data.monthly_jobs1617;
      drop rep_date

             ann_chg;
            * 3a Drop repeat_date & annual change;
      avg_jobs = sum(of aug__2016 -- aug__2017)/13;            * 3b
calculate avg_jobs of all months;
      label avg_jobs = 'Average Jobs';
      format avg_jobs 8.1;
      if avg_jobs EQ '' then delete;
      * 3c remove observations when avg_jobs is missing;
     if avg_jobs >900 then do
         market_size ='Large';
      * 3d classify jobs in market segments;
         output work.mrkt_large;
      * create first data output;
      end;
      else if avg_jobs >100 then do
         market_size ='Med.';
         output work.mrkt_med;
      * create second data output;
      end;
      else do
         market_size ='Small';
         output work.mrkt_small;
      * create third data output;
      end;
      keep sector state avg_jobs;

      select (sector);
      * 3e use select statement to filter;
            when ('GOVERNMENT') do;
            * filter when sector = govt.;
                  keep state
                        avg_jobs
                        market_size;
/*                drop sector;       */
                  output work.government;
            * create fourth data output;
            end;

            when('CONSTRUCTION', 'MANUFACTURING') do;              *
when sector = const or manuf;
                  keep sector
                        state
                        avg_jobs
                        market_size;
                  output work.goods;
            * create fifth data output;
            end;
```

```
           when ('FINANCIAL ACTIVITIES', 'PROFESSIONAL AND BUSINESS
SERVICES',
                   'EDUCATION AND HEALTH SERVICES', 'LEISURE AND
HOSPITALITY') do;
                 keep sector
                       state
                       avg_jobs
                       market_size;
                 output work.services;
     * create sixth data output;
           end;
           otherwise;
           * send rest of output to Null;
     end;
     label market_size = 'Market Size';                                  *
change label of variable market_size;
run;

/* 4 PDF output file so that bookmarks are created but not shown by
default*/

ods pdf file = HW11 bookmarkgen= yes bookmarklist = hide;

/*  5 Print first 50 and last 50 observations of data in step 2*/

title '5.1 - First 50 Observations from Monthly Jobs Data Set';
PROC Print data = work.narrow(obs = 50) label noobs;
RUN;

title '5.2 - Last 50 Observations from Monthly Jobs Data Set';
PROC Print data = work.narrow(firstobs=5385 obs=5434 ) label noobs;
RUN;

title '5.3 - Fifty Observations from Monthly Jobs Data Set Beginning with
#2800';
PROC Print data = work.narrow(firstobs = 2800 obs = 2849) label noobs;
RUN;

/* 6 Print the 6 datasets created in step 3 above */

title '6a - First 30 Observations of Small Markets';
PROC Print data = work.mrkt_small(obs = 30) label;
RUN;

title '6b - First 30 Observations of Medium Markets';
PROC Print data = work.mrkt_med(obs = 30) label;
RUN;

title '6c - Large Markets';
PROC Print data = work.mrkt_large label;
RUN;

title '6d - Selected Observations from Goods sector';
```

```
PROC Print data = work.goods(firstobs = 75 obs = 104) noobs label;
RUN;

title '6e - Small Markets in the Services sector';
PROC Print data = work.services(obs = 30) label;
  where upcase(market_size) = 'SMALL';
RUN;

title '6f - Government sector';
PROC Print data = work.government label;
RUN;

/* 7 Print specific contents is SAShelp vtable */

title '7 - Data Sets in the WORK Library';
PROC Print data = sashelp.vtable label noobs;
 where upcase(libname) = 'WORK';
 var libname memname crdate nobs nvar;
RUN;



ods pdf close;
ods listing;
```