```
/*      Josemari Feliciano
        HW 6, Fall 2018         */



/*  creates pathname that will be used throughout the code.
    creation of this assumes two things:
        1. that data files will be in the same folder as this file
        2. that Feliciano_Josemari_Format.sas file will be in the same folder as this file  */
%let pathname = C:\Users\seri\Desktop\advancedprogramming\assignment6\;



*   uses proc import to import 4 relevant datasets for analysis     ;
proc import datafile="&pathname.CohortCrosswalk.csv" out=cohort_data dbms=csv replace;
    getnames=yes;
run;

proc import datafile="&pathname.MainPatientFile.csv" out=main_data dbms=csv replace;
    getnames=yes;
run;

proc import datafile="&pathname.ODiagnosisCrosswalk.csv" out=diagnosis_data dbms=csv replace;
    getnames=yes;
run;

proc import datafile="&pathname.OutpatientVisits.csv" out=outpatient_data dbms=csv replace;
    getnames=yes;
run;



/*  both include, options statements are to ensure that seperate
    format file is ran and utilized for later printing of 'demo'd dataset   */
%include "&pathname.FELICIANO_JOSEMARI_FORMATS_final.sas";
options fmtsearch=(work.hw6format);



/*  creates:    'filtered_diagnosis' data
    purpose:    full joins diagnosis_data and outpatient_data on their diagnosis_code variables
                only keeps data if:
                    1. IF visitdate is before 2015:
                            calculated in WHERE statement via simple arithmetic by comparing
                            SAS unformatted dates of visitdate and 01/01/2015
                    2. AND IF diagnosis is one of the following:  'A1', 'A1.1', 'A1.2', 'A1.3', 'A1.4'
                also assigns each filtered data with count = 1 variable that I can exploit and sum up
                less important: order data by siteID for aesthetic purposes and for possible debugging
proc sql;
    create table filtered_diagnosis as
    select o.siteID, visitdate, d.diagnosis, 1 as count
    from outpatient_data as o full join diagnosis_data as d
    on o.diagnosis_code=d.diagnosis_code
    WHERE input("01/01/2015", mmddyy10.) - visitdate > 0
```

```
    AND d.Diagnosis in ('A1', 'A1.1', 'A1.2', 'A1.3', 'A1.4', 'A1.5', 'A1.6')
    order by siteID;
quit;



/*  creates:    'sum_a1_data' data
    purpose:    calculates sum of a1-related diagnosis by siteID grouping
                only keeps data that has 2 or more a1-related diagnosis     */
proc sql;
    create table sum_a1_data as
    select siteID, sum(count) as sum_a1
    from filtered_diagnosis
    group by siteID
    having sum(count) >= 2;
quit;



/*  creates:    'a1_labeled_cohort' data
    purpose:    full join datasets of 'cohort_data' and 'sum_a1_data' from the previous proc sql I ran
                MAIN PURPOSE is to label every data in cohort data with their chronic a1 status:
                    either 0 (non-a1 chronic) or 1 (a1 chronic) using sum_a1 variable from 'sum_a1_data
proc sql;
    create table a1_labeled_cohort as
    select uniqueID, c.siteID, CASE WHEN sum_a1 >= 2 THEN 1
            ELSE 0
            END as A1_dx
    from cohort_data as c full join sum_a1_data as s
    on c.siteID = s.siteID;
quit;



/*  creates:    'combined_data' data
    purpose:    merges a1-labeled cohort data with the main data file which includes demographic data
                minor:  creates sas unformatted date for both birthday and record date */
proc sql;
    create table combined_data as
    select uniqueid, a.siteid, sex, mdy(month_birth, day_birth, year_birth) as birthday, race, marital
    from a1_labeled_cohort as a full join main_data as m
    on a.siteid = m.siteid
    order by uniqueID;
quit;



/*  creates:    'compressed_combined' data
    purpose:    each uniqueid can have multiple data stemming from multiple siteids (from visiting oth
                so this reduces/combines data into a single row per uniqueid
                calculates age by using the latest record date
    point:      uses unique since max() method can keep multiple copies of same uniqueID if
                    a patient visited multiple clinics at the same day
                    uniqueid 9818594 falls into this category for instance                    */
proc sql;
    create table compressed_combined as
```

```
    select unique uniqueid, sex, birthday, race, marital_status, income_bracket, floor(yrdif(birthday,
    from combined_data
    group by uniqueID
    having record_day = max(record_day);
quit;




/*  creates:    'demo' data REQUESTED in homework6 handout
    purpose:    creates the 'demo' data requested for this project
                uses exact data from compressed_combined BUT labels everything accordingly in one step
    alternate:  could have labeled and formatted in multiple proc sql steps above but this is neater t
data demo; set compressed_combined;
    label uniqueID = "Unique ID" sex = "Gender" birthday = "Birthday"
        race = "Race" marital_status = "Marital Status" income_bracket = "Income Bracket" age = "Age"
        A1_dx = "Has A1 Chronic Condition Before 2015?";
    format sex gender_format. race race_format. marital_status marital_format. A1_dx A1dx_format.
        birthday mmddyy10. income_bracket income_format.;
run;



* FINAL OUTPUT 1:   prints first 10 observations of data sorted by unique id;
title "First 10 Observations in Demo Data Sorted by Unique ID";
proc print data=demo (obs=10) label;
run;
title;



* FINAL OUTPUT 2:   prints 3 seperate plots, comparing a1 distribution by sex race marital status;
title "Plot distribution of Chronic A1 Condition by Gender";
proc sgplot data=demo noborder;
  vbar sex /
         group=A1_dx groupdisplay=cluster
         dataskin=pressed;
run;
title;
title "Plot distribution of Chronic A1 Condition by Race";
proc sgplot data=demo noborder;
  vbar race /
         group=A1_dx groupdisplay=cluster
         dataskin=pressed;
run;
title;
title "Plot distribution of Chronic A1 Condition by Marital Status";
proc sgplot data=demo noborder;
  vbar marital_status /
         group=A1_dx groupdisplay=cluster
        dataskin=pressed;
run;
title;
```