

```

/*****
/* Program Name: STAT 604 HW#14
/* Date Created: 11/28/2021
/* Author: Jack Rodoni
/* Purpose: STAT 604 HW#14
/* Date Modified: 12/02/2021
/* Location: /home/u59649056/Homeworks/JRodoni_Homework14.sas
*****/

/* This assignment will use the “All Texas” - permanent data set that was created in Homework 10 and */
/* used in Homework 11. If you had difficulty creating this data set, the professor’s version, named */
/* alltx.sas7bdat, is available on the Week 9 module in Canvas and in the Fall2021 folder on SoDA. You will */
/* also be using three quarterly employment data sets downloaded from the U.S. Bureau of Labor */
/* Statistics. Download these three data sets to your homework data folder for PC SAS or download them */
/* then upload them to your homework data folder on SoDA. Familiarize yourself with these data sets */
/* before you start writing your program code. */

/* 1.) Add a header comment section to the beginning of a new program in your SAS session. Be sure */
/* to include a comment line above each section of the program that identifies the associated */
/* assignment step and a brief description of what the section is doing. Include housekeeping */
/* statements to clear titles and footnotes and suppress the printing of procedure titles. */

title;
footnote;
ods noProctitle;

/* 2.) If you are using the professor’s data set, assign a libref to the folder where it is located and add */
/* access=readonly at the end of the libname statement, before the semicolon, to protect data */
/* sets in this folder from being accidentally overwritten. Assign a libref to the mylib folder */
/* containing your permanent data sets. Create a fileref to the pdf file for output. Ensure that */
/* your SAS session can locate any permanent user defined formats that you create. */

libname mylib "/home/u59649056/Homeworks/mylib";
filename HW14pdf "/home/u59649056/Homeworks/mylib/JRodoni_HW14_Output.pdf";
libname HWDATA "/home/u59649056/Homeworks/Homework Data" access = readonly;

/* 3.) Open a PDF destination to receive your output. */

ods pdf file=HW14pdf;

/* 4.) The FIPS code is the common value between the COVID data we have been working with and */
/* the Employment data. The employment data sets are already ordered by the column containing */
/* the FIPS code and do not need any other modifications prior to merging. However, they do not */
/* contain county names and the FIPS code is a character value. Use one PROC SORT and one */
/* DATA step to create an unduplicated list of county names and FIPS codes from the permanent */
/* “All Texas” data set without altering the original data set. The final result of these two SAS steps */
/* will be a permanent data set with two columns ready to be merged with the employment data */
/* sets. NOTE: You must deal with any extra blanks that are created in the conversion from */
/* numeric to character or you will not get any results from your match merge process. */

PROC SORT data=HWDATA.alltx out=alltx_sort; /*out = so we dont alter the original data set*/
    by COUNTY_FIPS_NUMBER;
run;

data mylib.alltx_sort;
    set alltx_sort;
    by COUNTY_FIPS_NUMBER;
    if first.COUNTY_FIPS_NUMBER;
    AREA_FIPS = PUT(COUNTY_FIPS_NUMBER,7. -L);
    keep area_fips COUNTY_NAME;
run;

/* 5.) Use the match merge process in a single DATA step to combine the three employment data sets */
/* with the list of counties to create a new permanent data set. Start with the county list then add */

```

```

/* the employment data sets from the newest. The data sets are named to indicate */
/* the year and quarter of the data they contain. The resulting data set should have 254 */
/* observations and 26 variables. NOTE: A significant amount of the code for this step will be in */
/* data set options. */

```

```

/* a. The output data set must only contain employment data for FIPS codes that are in the */
/* Texas county list. */

```

```

data mylib.merged_data;
  merge mylib.alltx_sort(IN=Texas)
        HWDATA.employ2020q1(RENAME=(qtrly_estabs = qtrly_estabs20q1
                                     month1_emplvl = emplvl20m1
                                     month2_emplvl = emplvl20m2
                                     month3_emplvl = emplvl20m3
                                     avg_wkly_wage = avg_wkly_wage20q1))
        HWDATA.employ2020q2(RENAME=(qtrly_estabs = qtrly_estabs20q2
                                     month1_emplvl = emplvl20m4
                                     month2_emplvl = emplvl20m5
                                     month3_emplvl = emplvl20m6
                                     avg_wkly_wage = avg_wkly_wage20q2))
        HWDATA.employ2021q1(RENAME=(qtrly_estabs = qtrly_estabs21q1
                                     month1_emplvl = emplvl21m1
                                     month2_emplvl = emplvl21m2
                                     month3_emplvl = emplvl21m3
                                     avg_wkly_wage = avg_wkly_wage21q1));

  by area_fips;
  if Texas = 1;

```

```

/* b. There could be up to 7 rows per FIPS depending on the types of business owners in the */
/* county as indicated by the own_code column. The row with an own_code value of 0 is a */
/* summary of all owner types in the county. This is the only row we want to use for each */
/* county in our data step. The own_code column is not to be included in the output data */
/* set. */

```

```

  if own_code='0';

```

```

/* c. The only employment statistics we want are qtrly_estabs, the three columns that begin */
/* with month (monthly jobs), and the avg_weekly_wage. However, we do not want any */
/* of the new data to overwrite the older data due to same named columns. Use naming */
/* patterns so that the qtrly_estabs can be accessed as a group using a variable list. */
/* Include the two-digit year and quarter number at the end of the column name. */
/* Similarly, all of the "month" columns should be named as a group with the year and */
/* month at the end of the name. However, use the true month number instead of the */
/* month within the quarter. For example, Month1 in quarter 2 is June so its name should */
/* end with the number 6. Finally, the avg_weekly_wage columns should be named as a */
/* third distinct group with the year and quarter number at the end of each name. */

```

```

/* renaming done with the rename statements*/

```

```

/* keep statement done at the end*/

```

```

/* d. Define an array that can be used to access the monthly jobs columns incrementally. */
/* Make the array definition dynamic such that it would not need to be changed should we */
/* add another quarterly data set to the merge list. */

```

```

  array monthly{*} emplvl;;

```

```

/* e. Use a second array definition that will create 8 numeric variables to store the difference */
/* in numbers of jobs from one month to the next. The variable should get their name */
/* from the array name, and it should be distinct enough to not be confused with any of */
/* the other variable lists we have used so far. */

```

```

  array diffs{8};

```

```

/* f. Use a loop to populate the monthly difference variables. Base the stop value of the loop */
/* on the size of the array instead of hard coding the number. The difference will be */

```

```

/* calculated by subtracting the month1 value from month2 and so on. NOTE: Even */
/* though there is a 6-month gap in the reported data, we still want to compute the */
/* difference between January 2021 and June of 2020 in sequence. The index variable */
/* must not be in the output data set. */

    n = dim(diffs);
    do i=1 to n;
        diffs{i}=monthly{i+1}-monthly{i};
/*        output; */
    end;

/* g. Use the array in the argument of a function to compute the mean of all the monthly */
/* differences for the county. */

    MEAN_DIFS = mean(of diffs{*});
    keep qtrly_estabs: emplvl: avg_wkly_wage: COUNTY_NAME area_fips diffs: MEAN_DIFS;

run;

/* 6. Report the descriptor portion of the permanent data set of county names. Supply an */
/* appropriate title. */

title "My titles never make sense anyways";
proc contents data = mylib.alltx_sort;
run;

/* 7. Report the descriptor portion of the permanent data set of merged data. The variables must be */
/* listed in creation order instead of alphabetically. Supply an appropriate title. */

title "Do you even check this?";
proc contents data = mylib.merged_data varnum;
run;

/* 8. Print the changes in monthly jobs for each county from the last data set created. Show the */
/* county name, the 8 monthly difference variables, and the mean of the monthly differences. Use */
/* a variable list to specify the variables when appropriate. Do not show column labels or */
/* observation numbers in the output. Supply an appropriate title. */

title "Seriously, do you?";
proc print data=mylib.merged_data NOOBS;
var COUNTY_NAME diff: MEAN_DIFS;
run;

/* 9. Close the PDF destination. */

ods pdf close;

/* 10. View the data sets, log, and report information contained in your PDF output document to find */
/* the answers to the questions below and include the answers in a comment section at the */
/* bottom of your program file: */

/* (a) What county name is in the data set of counties but not in the match merge output data */
/* set? Why is it not included? */

/* Unknown, b/c there is no FIPS number to work with */

/* (b) How many observations were read from each of the employment data sets? */

/* 19248 */

/* (c) How does the average monthly difference from Brazos County compare to McLennan */
/* County? */

/* Brazos County: -157.5 McLennan: -299.5, brazos is losing less jobs on average */

/* (d) How do the extreme monthly difference values in Brazos County compare to McLennan */

```

```
/* County? */
```

```
/* -14113 is the largest absolute monthly difference in Brazos, 5542 is the largest monthly gain in brazos */  
/* -9820 is the largest absolute monthly differnce in McLennan, 4212 is the largest monthly gain in McLennan */
```

```
/* 11.) Save the final version of the program and convert it to a PDF file. Convert the log to PDF */
```