```
/*********************************************************************************
 Program Name: jblubau1_hw11_script
 Date Created: 10/30/2016
 Author: Joseph Blubaugh
 Purpose: Homework Assignment 11
********************************************************************************* */

libname datadb 'C:\Users\Joseph\Projects\learning\Statistics\STAT_604\Materials' access=readonly;
libname output 'C:\Users\Joseph\Projects\learning\Statistics\STAT_604\Data';

filename outpdf 'C:\Users\Joseph\Projects\learning\Statistics\STAT_604\Homework\jblubau1_hw11_output.pdf';

* 2) Read in an clean up zip codes data;
data output.zips;
   * change the length of the county field;
            length county $ 31;
            set datadb.zip_codes (rename = (estimated_population = EstPopChar));
            * Remove decommissioned zips;
            where decommissioned = '0';
            * Move redundent name info;
            if scan(county, -1) = 'County' then do
                        county = tranwrd(county, 'County', '');
            end;
            if scan(county, -1) = 'Parish' then do
                        county = tranwrd(county, 'Parish', '');
            end;
            if scan(county, -1) = 'Borough' then do
                        county = tranwrd(county, 'Borough', '');
            end;
            * Convert population to numeric);
            estimated_population = input(EstPopChar, 10.);
            * Replace underscore with blank;
            if find(timezone, '_') > 0 then do;
                        substr(timezone, find(timezone, '_'), 1) = ' ';
            end;
            * Keep needed variables only;
            keep county estimated_population primary_city state timezone zip;
            * Relabel variables;
            label county = 'County'
                        estimated_population = 'Est. Population'
                        primary_city = 'City'
                  state = 'State'
                  timezone = 'Time Zone'
                  zip = 'Zip Code';
run;

* 3) Summarise the data by State and City;
* a) Sort the clean data set by State and City;
proc sort data=output.zips;
            by state primary_city;
run;

* c) Aggregate population to city;
data zipsagg;
            * Make sure zips length is long enough;
            length zips $ 1700;
            set output.zips;
```

```
        * Reset count every time a new city is encountered;
        by state primary_city;
                if first.primary_city then do;
                        total = 0;
                        zips = '';
                        retain total zips;
                end;
        * Create a running sum of the population;
        total = sum(total, estimated_population);
        * Create a list of all of the zip codes in a city;
        zips = catx(',', zips, zip);
        if last.primary_city;
        label zips = 'Zip Codes'
                 total = 'Est. City Population';
        format total comma10.0;
        keep primary_city state county zips total;
        if total > 0;
run;

* 4) Open pdf, turn off bookmarks;
ods pdf file=outpdf bookmarkgen=no;

* 5) Print descriptor portion and subset of both data sets;
proc contents data=output.zips;
run;

proc print data=output.zips label;
        where primary_city in ('Albany', 'Center', 'Reno', 'Rome', 'Paris', 'San Juan', 'Juneau', 'Washington');
        var zip primary_city state timezone county estimated_population;
run;

proc contents data=zipsagg;
run;

proc print data=zipsagg label;
        where primary_city in ('Albany', 'Center', 'Reno', 'Rome', 'Paris', 'San Juan', 'Juneau', 'Washington');
        var primary_city state county zips total;
run;

ods pdf close;
```