Specifications:

- You are allowed at most 7 `DATA` steps and 12 `PROC` steps to complete this assignment.
- Set up your *librefs* and *filerefs* as usual.
- You'll need at least one custom format for this assignment, save any custom format you make to your HW library. For this assignment, name each custom format the same as the variable you apply it to. So, if you create a format to apply to Salary, then you should name your format Salary. [Note: this is not a great idea, in general, but it helps substantially with grading, so, here we are...]
- For this assignment, the data we need is split across six files (listed below): two sas data sets and four raw files. Now that we've covered all the DATA step-based methods for joining data sets, as well as some options for controlling data sets during a join, you should be able to identify the best way to join these into a single data set. Name your file as we usually do - e.g. HW6StarkIpums2005 for Tony Stark.
  - Cities.txt (2 variables, 181 records) contains demographic data that are not specific to an individual record.
  - States.txt (3 variables, 1,159,062 records) contains demographic data that are specific to an individual record.
  - The following four data sets contain information about households based on a classification. Since a person cannot be in two groups at once (e.g. you cannot be a renter and an owner of the same property!), you may assume these data sets are disjoint.
    - Contract.txt (6 variables, 9756 records)
    - Mortgaged.txt (6 variables, 545,615 records)
    - FreeClear.sas7bdat (6 variables, 300,349 records)
    - Renters.sas7bdat (6 variables, 303,342 records)
  - The FreeClear and Renters data set were built on a system using Japanese encoding which is (most likely!) different from your setup on the VTL. You'll see some notes about these data sets having different encoding – those are OK and there isn't anything you should do about them anyway.
- As always, I suggest you plan out your program before getting started. For example,
  - reviewing the data sets and what variables are present in each
  - Determine not only how you plan to read in the data, but how you plan to combine the data sets. (If it helps, review the difference between the three types of joins we could use: concatenate, interleave, and match-merge. Each one is used for a specific application and, in particular, ***horizontal and vertical joins are not interchangeable***!)
  - Also, we've previously talked about the importance of efficient coding; for example, doing all your data cleaning in one location. However, sometimes life throws you a curveball and you need to do some cleaning or deriving to even join the data sets. In those cases, you have no choice but to carry out those tasks when necessary. [Clearly I'm telling you this because this is one of those times.]
- As you might expect, you'll need to do some work on the combined data before it can be considered cleaned and to ensure some variables we need have been correctly derived. Here are some specifics to help you out:
  - You'll need a new variable that is based on which data set the household information comes from. The possible values are shown below. I trust everyone can deduce which records go with each value.
    - "Yes, contract to purchase"
    - "Yes, mortgaged/ deed of trust or similar debt"
    - "No, owned free and clear"
    - "N/A"
  - Home Value is set to $9,999,999 instead of missing for individuals who rent. For those records, use the special missing value .R to denote the values.
  - Ensure that if Home Value is missing for any other household, you would use .M for those generic missing values.
  - The Ownership variable takes the value 'Rented' for anyone from the renters data set and is 'Owned' for all other records.
  - MetroDesc is a variable you need to derive and it describes what each value of Metro means. ***Create the MetroDesc variable without using any conditional logic.***

    * 0 is *Indeterminable*, 1 is *Not in a Metro Area*, 2 is *In Central/Principal City*, 3 is *Not in Central/Principal City*, and 4 is *Central/Principal Indeterminable*
    * Yes, we've learned all the tools you need to do this.
    * It only takes one function to do this. Do not nest functions! Yes, we have already learned the function.
    * Hint, remember that scalability is important. What is a way to change a value of Metro (e.g., 0) into a different value (e.g., Indeterminable) and store that new value in a new variable?

- Electronically validate your descriptor portion against mine, then electronically validate your content portion against mine. Both my descriptor and content portions are available for you in the Results folder (HW6DugginsDesc and HW6DugginsIpums2005). For both validations, use 1E-15 as the criterion for the absolute difference between numeric values. As has become our habit, use a macro variable to store the constant options for the validation. Name your macro variable whatever you like.

- This week, our final results brings together several of the reporting tools we've seen so far. Produce the report you see in HW6 Duggins IPUMS Report.pdf. Do not send any results to any other destination.

  – All graphs that I create here have a width of 5.5 inches and 300 DPI,

  – Recall from a previous assignment, page breaks in the PDF can be controlled by using the option STARTPAGE = option in the ODS PDF statement. The value of NEVER will only move to the next page when the current page is full. The value of NOW requests a page break be inserted immediately.

  – Do *not* use PROC PRINT. Moving forward, all tables and listings are created with PROC REPORT.

    * Note that in my (and your) report, special missing values do not appear. We simply get . instead of .R or .M - even though they show up properly in the PROC UNIVARIATE summary!
    * We've briefly talked about PROC REPORT some already, so what is going on that is causing this to happen? You don't need to include an answer in your homework assignment, but you should definitely be curious about this and answer it if you can!

  – In the graph generated by UNIVARIATE, I used the default histogram and the default settings for a non-normal density. Your HISTOGRAM statement here should only have five tokens.

  – In the panel graph, use the NOAUTOLEGEND option in the SGPANEL statement to get rid of the legend entirely. Use NOVARNAME in the PANELBY statement to force it to only print the value of the panel variable.

  – I used all the default fonts, font sizes, etc. I only changed two aesthetic options: I chose the color of the kernel fits (Wolfpack red in both graphs) and increased the thickness of the kernel fit (only in the single-cell graph). You should change these options as well, but they don't have to match mine exactly.