Final Project

In addition to the usual requirements for our programming assignments, please note the following:

1. Allowable Resources: You are allowed to use any materials provided to you by me during the current semester - this includes lecture notes/slides, videos, your class notes, previous discussion board posts, homework assignments, homework solutions, quizzes, and quiz solutions. You are also allowed to use any of the SAS help documentation located at documentation.sas.com - however, if you do use the SAS documentation you must include a note in your code that explains what you used from the help files and the URL for the help file you referenced. This will effectively serve as a citation for your help in the event that a check for plagiarism flags that section of your code.

   No other resources are allowed. This includes, but is not limited to, cheating sites such as Chegg, other SAS resources such as SAS Communities, other programming help sites such as Stack Overflow or Stack Exchange, and generative AI tools like ChatGPT. You are not allowed to discuss any portion of the exam with any other individual, including but not limited to, current classmates, significant others, private tutors, missionaries who happened to stop by your house, siblings, etc. If you are unsure of whether a resource is allowed, it probably isn't - in that case, ask for clarification in advance. ***If the idea doesn't come out of your head, then it shouldn't be part of your graded work.***

   Notice that I am not included in either list. That is because I will not be as forthcoming with help on the final as I usually am on homework assignments. That said, I am the only other individual you can talk to about the exam - and **I certainly encourage you to ask me questions if you get stuck or need clarification!** You may ask questions in the Slack channel for the final, but I will be the only person allowed to respond. (Slack will still let you, but I will delete the post immediately and you will get a sternly worded email. If it happens twice, you'll lose 10% of your grade on the exam.) The channel, as always, can be helpful so that we are all working from the same information. If I feel the question is better answered one-on-one, I will reach out to you via e-mail to set up a time for us to talk.

2. Exam Setup and Grading:

   (a) Section 1 includes the specifications that are primarily about data creation and management – these are the ones that are due at the first deadline of this assignment. Section 2 is for the more analysis-focused questions that you'll turn in at the second deadline.

   (b) Section 1 and Section 2 will be graded separately. Our usual SAS rubric will be used to grade both sections of the final project.

   (c) **The sum of your scores on Section 1 and Section 2 will be your grade on the Final Exam - Part A.** Thus, for example, losing 5 points on Section 1 is equal to losing 5 points on Section 2. The two sections are roughly equal in terms of points available, so they contribute roughly equally to your the overall project grade.

3. Storage: All data, formats, files, or other objects *you store* must be saved in the `Final` library. All formats you create must be saved in this library and all non-SAS files should be saved in the same folder. The only **four** SAS data sets you need to save permanently are described in detail below.

4. Commenting:

   (a) Include your usual header at the top.

   (b) DATA steps only *need* one comment prior to each step that includes a brief statement of what you're using that step to do.

   (c) PROC steps only *need* a comment before them if they are used to create one of the pieces of output – in that case, the comment should just include the name of the output. For example: *Output 6;

   (d) You do not need to provide additional comments for each piece of code that you write unless you want to.

Overview:
Congratulations [EMPLOYEE NAME GOES HERE], you have been promoted to Lead Statistical Programmer! This promotion is the result of your hard work during your 15-week stint as a Validation Programmer. As a reward for this momentous achievement, you have been assigned to a project for a new client we just landed – the Washington State Department of Licensing. They are interested in analyzing their newly-updated (20OCT2023) database of electric vehicles and you've been chosen to lead the charge on this analysis. (Get it, the *charge*, hah! Because they are *electric* cars. You know, it's not funny if you have to explain it...)

Unfortunately, we landed this project because they are not great at data management and that is one of your specialties. Your first task will be to prepare the data to meet their specifications and to support the generation of the initial report you will be preparing. The client specifications for each phase of this project – Data Management (Phase 1) and Presentation of Results (Phase 2) – are laid out below.

Congratulations again on your promotion, [EMPLOYEE NAME GOES HERE]. You truly are one of the best employees we've got!

Sincerely,
[MANAGER'S NAME GOES HERE]

- Remember, this project is divided into two phases: data management and presentation of results.
  - Data Management: The code for the first section of the project is due at the first due date. The purpose of this phase of the project is to build the four data sets you need for phase two. *No output is due at the end of phase 1.*
  - Presentation of Results: The code for the second section of the project is due at the second due date. The purpose of this phase of the project is to create all the analyses - so, the tables and graphs.
  - *To ensure that everyone can earn full credit on Phase 2, you must use* ***my*** *data sets from Phase 1 when you work on Phase 2.* My data sets will automatically populate in the under `/Results/FinalProjectPhase1`.
- Client-Provided Resources:
  - The client has already provided the source data sets and IT has loaded them into `/Data/WashingtonState`
  - Raw files are located in the `RawData` folder. Each file contains a substantial header provided by the client. Much of the detail about the content and structure of the files can be found here. Pay special attention to the Unknown data set – they have not been able to correctly export it, so it needs some advanced data handling skills.
  - They have some look up tables they regularly use and they are stored in Microsoft Access and Excel formats. These are located in `StructuredData`. The Access database is kept up to date, so we will exclusively use it and *we will not use the Excel format.* We know you are new to the role of Lead Statistical Programmer, so remember that when you use an Access database no one else can – so get in, get what you need, and get out. ***In particular, do not wait to the last minute to use this data as other programmers might be doing the same thing...***
  - They have used SAS in the past and have kept several of their formats and informats over the years. The relavent ones are located in `FormatCatalogs`.
  - *No other resources are needed to start the project. Any other resources you need are to be created by you from the client-provided files.*
- Results:
  - The client had one of their employees mock up a PDF that shows a sample of what they want produced. The PDF has their name in the title and can be found in `Results/Duggins Washington State Electric Vehicle Study.pdf`.
  - When you produce your results, use the same file name but replace their name with your name.
  - Unless otherwise specified, your output should be identical to those in the mocked-up PDF. Color choices do not need to match exactly unless the colors are provided to you.
  - As with most output objects, the titles and footnotes contain valuable information about what the client wants displayed in the report. To help support you dividing this work into phases, I (as your amazing and duteous manager) have provided you some hints below on how to prepare some of the data sets you will need and how to achieve some of the tabular and graphical output.

Section 1 – Due November 28 @ 11:00 PM

Data Management:

1. Budget: 7 DATA steps and 8 PROC steps
2. The client is expecting four data sets to be generated by Phase 1. They have provided a summary of the metadata they expect their final data to have in each of these data sets. I suggest you refer to it early and often. It is located in `Results`.
3. You have three raw data sets to read in.
   - The ‘Yes’ and ‘No’ data sets are relatively straightforward for programmers with an understanding of basic data reading principles.
   - No data cleaning should be done to the ‘Yes’ and ‘No’ data sets individually.
   - The ‘Unk’ data set will take more advanced skills. In particular, it is based on groups of variables so you will need to use a technique for handling SAS variables in groups.
   - The variables that are grouped are placed in groups of 250 – I strongly suggest using a macro variable to set this quantity globally when reading in this data set.
   - Due to the grouped nature of the data, some data cleaning is necessary here in order to prepare a data set that can be combined with the other two. (1) You’ll need to transform the wordy dates they provided to SAS dates and (2) you need to ensure records that are missing all seven of the “core” variables are not sent to the data set – these are an artifact of the grouping done to these variables.
   - Hint: You’ll want a custom format to handle those wordy month names to turn them into something more number-y for you to use.
4. For context, VIN is an identifier that uniquely identifies a particular vehicle. DOLID is an identifier that uniquely identifies a driver. Because a vehicle can be bought/sold or otherwise need to be reregistered, the same VIN may occur multiple times in the data set. (In the actual database, some VIN occur nearly 1,000 times. For your first analysis, we’ve limited you to VIN that appear no more than 250 times.) Due to the way the database is built, each DOLID can only appear in the database once. Thus, DOLID is not only unique to a person, it is a unique row identifier in the database while VIN is not – it is unique only to a vehicle. Information that is specific to the car (and thus, not unique) is included in the lookup table named “Demographics.”
5. In terms of BEV vs PHEV, BEV refers to a fully-electric (i.e., battery powered only) vehicle while PHEV refers to a plug-in hybrid (i.e., battery and combustion engine together). As such, we expect to see differences in electric ranges between PHEV and BEV and likely differences in which vehicles qualify for the Clean Alternative Fuel Vehicle (CAFV) designation.
6. Once you read in all the CAFV data sets, they need to be combined in a single data set. I suggest naming this one `AllCAFV`. (Remember, SAS is not case-sensitive for name tokens.)
   - Use an appropriate technique to combine the data sets based on their structure and contents.
   - You’ll need to derive new variables at this point since otherwise this is the last time you’ll know which data set the records came from.
     (a) ZipN (num): numeric version of Zip
     (b) CAFVCode (num): records from ‘Yes’ are 1, ‘No’ are 2, and ‘Unk’ are 3.
     (c) CAFV (char): description of the CAFV categories – ‘Clean Alternative Fuel Vehicle Eligible’, ‘Not eligible due to low battery range’, and ‘Eligibility unknown as battery range has not been researched’
7. After creating a data set of all CAFV records, the remaining information needs to be loaded into each record:
   - The lookup table “Non-Domestic Registrations” contains the driver’s license ID (DOLID) and State Postal Code (ST) for all registered vehicles in this study with a non-US postal code. This table is guaranteed to be in ascending order of DOLID because Microsoft Access will guarantee it for this table. AP refers to “Armed Forces Pacific” and BC refers to “British Columbia”.
   - The lookup table “Demographics” contains information about *all* electric cars in the Washington State database and not just the ones for our study. When you grab information from this table, make sure you only pull in records if there is a match.
   - SAS provides a data set `Sashelp.ZipCode` that includes geographic information that is necessary for this study. Use this as a lookup table to grab the remaining information to complete your records. Obviously, this is for every zip code in the United States, so don’t grab all of these records either or we’re taking away your promotion to Lead Statistical Programmer.

8. Regardless of the order in which you decide to combine the AllCAFV data set with the lookup tables, in your last DATA step you will need to set the attributes of your permanent variables and derive some new variables based on the information found in the raw data or lookup tables. In particular, you need to derive the following as evidenced by the provided metadata. Some variables may already exist and just need some data cleaning. This final DATA step to combine and clean your data will produce your primary data set. Name it Final¡yourname¿EV so, for example, Tony Stark would create a file named FinalStarkEV.

   - Latitude (num): second number in the parentheses in Location
   - Longitude (num): first number in the parentheses in Location
   - StateCode (char): comes from Sashelp.ZipCodes for domestic records. For non-US records, use the information provided above where that lookup table is described.
   - ElecUtil (char): comes from raw data for domestic records. For non-US records use the value ‘NON WASHINGTON STATE ELECTRIC UTILITY’
   - StateName (char): comes from Sashelp.ZipCodes for domestic records. For non-US records, use the information provided above where that lookup table is described.
   - PrimaryUtl (char): first value in ElecUtil which is a pipe-delimited set of all electric utilities that service an address
   - MaskVin (char): derived from VIN by replacing the first 7 characters with asterisks. Use the `CATS` function. For example, `CATS(‘Jon’,’athan’) = ‘Jonathan’`, and it can, of course, accept both literal and name tokens as arguments.
   - EVTypeShort (char): PHEV or BEV are the only possible values. Don’t you dare use conditional logic on this one – derive it directly from EVType.
   - Make (char): derived from MakeCat without using conditional logic because the client maintains a central list of all vehicle make values
   - Model (char): derived from ModelCat without using conditional logic because the client maintains a central list of all vehicle model values
   - BaseMSRP (num): original values are poorly reported – to improve, set any currently missing values to .M (for missing), any negative values to .I (for invalid), and any 0 values to .Z (for zero).

9. From your Final¡yourname¿EV data set, you need to produce three additional data sets:

   (a) Final¡yourname¿UniqueVinMask contains all variables in Final¡yourname¿EV but only one record per VIN. It will be used to help investigate how useful this masking strategy is.

   (b) Final¡yourname¿Models contains one record per Make. The reocord for each Make should include a column for every value of Model.

   (c) Final¡yourname¿CafvCrossEV contains all percentages that result from a two-way frequency analysis of CAFV and EvType. You’ll see from the client metadata they used the CAFV code rather than the description and similarly they used the short version of EV type. You don’t have to do that, but it is probably less of a hassle to look at.

10. That’s it – those four data sets can be used to create every output object the client wants.

    - Outputs 4 and 12 uses the VinMask data
    - Outputs 9 and 10 use the CafvCrossEv data
    - Output 13 uses the Models data
    - All other Output use the primary EV data

    That concludes what is due in Phase 1. Of course, you probably want to at least *try* to produce the output in Phase 2 to ensure that your data sets can be used as intended. Just don’t worry about making that code pretty or delivering it to the client – they only want to see the data management piece for Phase 1.

Presenting Results:

1. Budget: 0 DATA step and 19 PROC steps
2. You are being asked to use some macro variables:
   - IdStamp: this is the text that appears in the first footnote of every output object. It depends on three automatic macro variables: `SysUserID`, `SysDate9`, and `SysVLong`.
   - TitleOpts: this contains all the options for the headers. This makes preparing handouts or presentations easy by changing all title settings at once. It should set 14pt bold font.
   - SubTitleOpts: this contains all the options for the "Output #" title that always comes first and for all subtitles. It should set 10pt bold font.
   - FootOpts: this contains all the options for the footnotes. It should set left justified, 8pt italic font.
3. Output:
   - The client only wants a PDF.
   - The PDF should use the sapphire style and all images in the PDF should be at 300 dpi and 6 inches.
   - No standalone graphics files should be created.
   - They company has some employees who often use HTML output and others who use listing ouptut. Your program should ensure that no matter which employee runs your code, it only produces the PDF. You do not need to worry about restoring default ODS settings when your program concludes.
4. Formats:
   - You are going to need at least five custom formats
   - All formats should be saved to the project library (Final) so they can be provided to the client for reference.
   - When categorizing the electric ranges, the value of 0 should always be in a category by itself. (Recall, these are electric vehicles – it is impossible for them to have an electric range of 0, that indicates a data entry issue.) Missing or negative values should be grouped together. Otherwise, the ranges are provided in the mocked up PDF.
5. For Output 1, that's two separate listings placed side-by-side to ensure we see relevant records.
6. For Output 5, the client wanted the missing values included in the table rather than as the default footnote.
7. For Output 6 (and other similar output objects), note the client wants the frequencies to appear with commas for easier reading whenever possible.
8. For Outputs 9 and 10, the client wants them on the same page so they are easy to compare.
9. For Outputs 9 and 10, note that at most 5 colors are going to be used but they will always refer to 5 distinct groups. The client wants to ensure that different colors are used and that the colors "go together" as they put it. I suggest using `colorbrewer2.org` and choosing a pleasing color palette to get your 5 colors easily.
10. For Output 9, set the bars to be 50% of the available width and keep them from getting outlined. Use the VALUESDISPLAY= option to control the labels used for your VALUES= entries.
11. For Output 10, again ensure there is no outlining. The labels at the end of the bar should be the plotted percentages in 10pt, gray font. (Or grey font if you're British.)
12. For Output 11, do not attempt to set colors manually unless you have about 40 favorite colors handy...
13. For Output 13, note that some of the values of Make have so many Model columns that the table is too wide to fit onto a page.
    - Apply the ID option to the Make report item to ensure it is used as a row header.
    - Don't try to write a separate DEFINE statement for each Model – they all need the same attributes, so put them all in one DEFINE statement. (Yep, it's legal... sometimes...)
    - You may need to play with the titles a bit to get them aligned properly...
14. Output 14 is the first of a series of REPORT results that increase in complexity. Outputs 14 through 17 all use the same basic report definition, you are making slight changes so the client can pick out which one they like best for the full project.
    - Output 14: All the statistical analysis should be done in PROC REPORT. No styling (e.g., colors, fonts, etc.) is necessary beyond the defaults. Formats and labels are necessary.

  - Output 15: As the footnote says, this alternative display carries down the value of EV Type within each set of Model Years. *For this report item, you are only being evaluated on the code that changed from Output 14 – if your Output 14 is not correct, the client has ensured us they won't hold that against us in each of these reports.*
  - Output 16: Like Output 15, but including gray background on the summary rows, a black line with white text as a table footer to describe the coloring, and a data-driven set of colors for the Electric Range means. The range cutoffs use the BEV levels shown in earlier output objects. The client wants a single-color ramp from light (low means) to dark (high means). They used purple in the mock-up, but you can use any color you like. If you need inspiration, head back to `colorbrewer2.org`. The 0 cells should be flagged with a shade of red though as they are not truly zero.
  - Output 17: The BEV and PHEV cars have substantially different ranges, so they should use different color cutoffs. The updated cutoffs are in the table footer and a description of the coding is included as a footnote. You should use the same colors for both ramps – e.g, the first level should be the same color for both BEV and PHEV even though the cutoffs are different.

15. Unless otherwise specified, your output should be effectively identical to those in the mocked-up PDF.
16. Use the ODS option NOPROCTITLE to ensure that only the headers we want are appearing in the output.

Closing Comments:

Note that, as in most practical situations, you do not have an already-completed data set to use to guide you to what the correct data set. Though, through the magic that is "this is still a project that has to work for like 200 people" you are getting a PDF of the actual results you should produce.(Also, there is no validation here - so, for example, column position does not need to match exactly.) You also have the data set metadata which is very helpful.

Also, note that the majority of the work on this project is on the front end – getting the data in, cleaned, and prepared. As is the case in practice, data handling is the vast majority of every project – analysis cannot begin until the data is well-prepped!

If you have spent time this semester working on writing out your code by hand and planning how to achieve each output, this is the time to shine - that skill is crucial on larger projects. If you've put that off all semester, then I suggest you start planning out your attack for this project immediately. Similarly, if you've spent the semester having to look back to your notes or the book on early-semester topics like reading in raw data or using ODS, then this project may be time-consuming because you are expected to have that content internalized by now.

**In a project this size, there is sure to be something you or I have overlooked, so let me re-emphasize that I encourage everyone to ask me questions early and often. The worst that can happen is that I just tell you I cannot help you. The best that can (realistically) happen is that I realize I forgot to give everyone a crucial detail and I provide information that helps you keep from spinning your wheels!

Good luck!