**Project 3**

**General Instructions**

The project is modeled after a data management project you could encounter as a research assistant. The instructions take you through several steps of managing and manipulating data files in preparation for analyses. Throughout the project, you will apply and build on some of the more advanced techniques covered in class.

**Honor Code and Authorized Forms of Assistance**

You will complete this project in an assigned group. You may discuss the project and any questions you have only with fellow members of your assigned group or the instructor. You may not discuss this project with any other person. You may not consult any materials from past semesters of EPID 700, but you may consult any other (non-human) resources (e.g., course materials, readings, internet sources).

Use good coding practices and style as modeled and discussed throughout the semester. Annotate your code clearly by project part.

**Project 3 Submission Guidelines**

There are five deliverables for Project 3: the Project 3 Questions on Sakai(**90 points**) and four files (**10 points**).

All deliverables are due to Sakai by **Tuesday, November 17 at 11:55pm.**

## *Project 3 Questions*

Once you have completed this project to your satisfaction, complete the Project 3 questions on Sakai. The Project 3 questions are also included in blue text in the Instructions for Data Management, below the relevant project parts.

Throughout the entirety of the Project 3 Questions on Sakai, be mindful of the following instructions:

* All responses must be reported with the correct number of significant digits to receive any credit.
  + Because you are using significant digits to round, you should expect some variation in the number of decimal places across your answers.
* Do not use commas as thousands separators.
  + For example: Enter 12345, and not 12,345.
* Do not enter any percent characters (%).
  + These characters are already included around the answer spaces on Sakai where relevant.
* Where multiple values are requested, enter them in the order specified.
  + For example: If asked for the mean and standard deviation, enter the mean first and the standard deviation second.

## *Project 3 Files*

You will also submit the following files to Sakai > Labs > Project 3 Files:

1. SAS program file (**EPID700\_Project3\_GroupName.sas**) *(log and Results file are not needed)* **(2 points)**
2. Project 3 Codebook (**Project3\_Codebook\_GroupName.pdf**). The instructions will guide you through creating this file. Instructions for this file are given in red text in the Instructions for Data Management. **(3 points)**
3. Project 3 Report (**Project3\_Report\_GroupName.pdf**), containing the 13 site-by-site reports you will generate at the end of this project. Instructions for this file are given in red text in the Instructions for Data Management. **(3 points)**
4. Encrypted zip file containing your final Project 3 Data file (**Project3\_Data\_GroupName.csv** *within* **Project3\_Data\_GroupName.zip**). Instructions for this file are given in red text in the Instructions for Data Management. **(2 points)**

**Data Source**

The data provided to you are a subset of data collected in the East Africa Cross-Border Integrated Health Study (CBIHS). A major component of the CBIHS was a bio-behavioral survey conducted between late 2016 and early 2017. The CBIHS assessed health status and service uptake among populations recruited at venues and events at several cross-border sites in Kenya, Rwanda, Tanzania, and Uganda near Lake Victoria.

## *Methods Summary*

The three-step Priorities for Local AIDS Control Efforts (PLACE) method[[1]](#footnote-1) was implemented at each cross-border site to separately sample and recruit participants.

In Step 1 of the PLACE method, study staff interviewed community informants (using data collection “Form A”) to ask people to identify venues or events (“spots”) within the cross-border site where people meet new sexual partners and/or where female sex workers, men who have sex with men, people who inject drugs, mobile populations, and other vulnerable groups socialize. Spots reported by multiple community informants were manually de-duplicated. Then, a list of unique spots was compiled (in “Form A Data Entry Template.xlsm”).

In Step 2 of PLACE, interviewers visited a census or a stratified random sample of spots (depending on the number of spots listed in Step 1) to verify the existence of these venues and events and to collect data on spot characteristics (using “Form B”).

In Step 3, interviewers visited spots to recruit patrons and workers at the spots for a bio-behavioral survey (administered using “Form C”). Participants completed a questionnaire on socio-demographic characteristics, health status, access to health services, and sexual and health-seeking behaviors. HIV counselors offered optional on-site rapid HIV tests with counseling, and those with a reactive test were linked to care and asked to provide blood spots (DBS) for HIV-1 RNA viral load testing.

***Project Materials***

Posted with these instructions, you’ll find several materials:

1. Four data Collection Tools and Templates:
   1. **Form A.pdf**: This is the questionnaire used to interview community informants in Step 1 of the PLACE method. Community informants completed one **Form A.pdf** questionnaire for each spot they reported.
   2. **Form A Data Entry Template.xlsm**: This is a data entry template that study staff used to enter the compiled list of unique (de-duplicated) spots reported each cross-border site, along with some basic information about each spot (as collected in **Form A.pdf)**.
      1. This template was completed separately for each cross-border site. The 13 site-specific files are not provided here, for brevity.
      2. In the template, one spot was listed per row, and spot characteristics (as reported by community informants) were summarized in columns G through X.
      3. **Form A Data Entry Template.xlsm** is the actual data entry template and spot sampling tool used for the study. For the purposes of this project, you may disregard: the “Instructions” worksheet and, in the “Spot Data Entry and Sampling” worksheet, Columns A through F and Y through AG. (These portions of the workbook were used to implement the stratified sampling protocol.)
   3. **Form B.pdf**: the questionnaire used to verify the existence of spots and collect spot characteristics in Step 2 of the PLACE method
   4. **Form C.pdf**: the questionnaire administered for the bio-behavioral survey in Step 3 of the PLACE method
2. Four raw data files containing a subset of variables collected with each questionnaire:
   1. **FormA.xls**
   2. **KirongweA.txt**: A data set containing re-keyed Form A data from one cross-border site (Kirongwe)
   3. **FormB.csv**
   4. **FormC.xlsx**
3. Three SAS program files:
   1. **CBIHSFormats.sas**: A program that includes variable formats
   2. **CBIHSApplyLF.sas**: A program that includes format and label statements
   3. **proc\_codebook.sas**: A macro you will use to generate a codebook

***Data Sets Characteristics***

Variables names in the three provided data sets correspond to question numbers in their respective data collection/data entry tools:

|  |  |
| --- | --- |
| **Data set** | **Data collection/Data entry form** |
| FormA | Form A Data Entry Template.xlsx (see columns G-X) |
| FormB | Form B.pdf |
| FormC | Form C.pdf |

**FormA.xls**

This data set was created by a colleague who imported and cleaned Form A Data Entry Template.xlsm files for each of 13 cross-border sites. To avoid the need to repetitively write code to import the Form A data from each of the 13 cross-border sites, she wrote and executed a custom SAS macro to import the files, define consistent variable attributes, and add a variable indicating the cross-border site at which the data were collected. She concatenated the 13 site-specific SAS data sets to generate the Form A data set which has been provided to you (FormA.xls).

**Form B**

Data for the Form B survey were collected on tablets and aggregated as one data file on a secure server. As Form B data were received them, your colleague used another macro to identify data entry and protocol errors, monitor progress towards targets, and perform some simple data management tasks. She has provided you with the resulting Form B data set (FormB.csv), which contains one record for each spot visited for Form B data collection.

**Form C**

The protocol for compiling and cleaning the Form C data mirrored the process used for the Form B data. The Form C data set provided (FormC.xlsx) provided contains one record for each person who was approached for the bio-behavioral survey.

**Instructions for Data Management**

There are ten parts to this assignment. Throughout these instructions, variable names appear in **bold**; file names are in “quotations marks,” and SAS data set names appear in *italics.*

Throughout this assignment, to preserve the integrity of the raw data and ensure your process is well documented and reproducible, all data management tasks must be done using SAS code. (i.e., do not make any changes to the provided data files, in Excel or other non-SAS programs).

## Part I. Reading External Data Files into SAS

* 1. Read in the three raw data files (“FormA.xls,” “FormB.csv,” and “FormC.xlsx”) as SAS data sets.
* You may use any method(s) you would like to read the data files into SAS; however, your SAS program must include code that can be run to read the data in (i.e., I should be able to update the file paths to match the file locations on my computer, then run your code to import the data).
* Name each SAS data set according to the name of the corresponding external data file (e.g., give the name *FormA* to the SAS data set you read in from “FormA.xls”).
* In the *FormC* SAS data set, ensure that the DATE9. format is assigned to the variable **c2** and that the TIME. format is assigned to variable **c3**.
  1. Next, execute one – and only one – PROC step – which, when executed once (and without use of a macro) produces output showing the contents of your three SAS data sets: *FormA*, *FormB*, and *FormC*.
* Hint 1: PROC CONTENTS won’t accomplish this; it only produces output for one data set at a time.
* Hint 2: You will need to research the relevant procedure on your own.
* Your output should include at least the following information for each of the three data sets: the number of observations and variables, variable names, variable types, and formats.

**Question 1.** What procedure did you use to report the contents of all three data sets with just one PROC step? (Enter only one word.) PROC \_\_\_\_\_\_

**Question 2.** Using the output from the procedure you named in Question 1, report the number of observations in your *FormB* data set. \_\_\_

## Part II. Conducting Data Quality Checks

The Form A data entry protocol specified that once unique spots had all been entered into the “Form A Data Entry Template.xlsm” for a cross-border site, the data entrants were to assign Spot IDs sequentially, starting with a value of 1, up to the final unique spot identified.

For example, in the Form A Data Entry Template for the cross-border site Malaba, the first spot entered into the template should have been assigned a Spot ID of 1, the second spot should have been assigned Spot ID 2… up to the last spot on the list. Similarly, in the Form A Data Entry Template for the cross-border site Busia, the first spot entered should have been assigned a Spot ID of 1, the second spot should have the Spot ID 2, and so on. In the Form A data, cross-border sites are identified by the variable **a\_siteid**. The assigned Spot ID values in your *Form A* data set are in the variable **spotid**.

Though you know how the spot IDs were *supposed* to be assigned, you also know how important it is to verify that the assigned **spotID** values have been assigned according to protocol. Why? Here are a couple of important reasons:

* Later on, you’ll merge records that were collected at any one spot across the 3 data collection tools (Forms A, B, and C). To do this, you will use a merge variable that incorporates **spotID**. Duplicated Spot IDs can cause erroneous linkages. To understand why this would be problematic: if, e.g., Joe’s Bar has a Spot ID of 12 and the next spot, Sally’s Salon, is accidentally assigned a Spot ID of 12 also (instead of 13), any Form B or C records from Joe’s Bar would end up linked with the Form A data from Joe’s Bar *and*, erroneously, Sally’s Salon.
* Deviations from protocol in anywhere can be a red flag for deviations from protocol elsewhere. Verifying protocol adherence in places where you are familiar with the protocol can help focus other data quality checks. For instance, if you find several errors at a certain study site, you may want to follow-up with the data manager (or the data collection supervisors at that site, if the study is ongoing) to inquire whether there were systematic issues with the data entry process at that site that could affect your results.

Mindful of the headaches that errors in **spotID** valuescould cause down the line, create a new data set called *CheckSpotIDs* that contains a new variable **flag\_spotid** coded as follows:

1. **flag\_spotid** = 1 wherever the **spotid** is not exactly 1 more than the prior value of **spotid**, when the **spotids** are ordered in ascending order (i.e., whenever the current **spotid** value is not sequential, relative to the **spotid** in the prior observation).
2. **flag\_spotid** = 0 when the **spotid** value *is* sequential, relative to the prior value.

**flag\_spotid** is, essentially, an indicator variable for **spotid** sequence interruptions. To help envision your coding goal, see the values below that **flag\_spotid** should take for a hypothetical site “30” that has 10 observations and 2 errors in the **spotid** sequence:

|  |  |  |
| --- | --- | --- |
| **a\_siteid** | **spotid** | **flag\_spotid** |
| 30 | 1 | 0 |
| 30 | 2 | 0 |
| 30 | 3 | 0 |
| 30 | 4 | 0 |
| 30 | 6 | 1 |
| 30 | 7 | 0 |
| 30 | 8 | 0 |
| 30 | 8 | 1 |
| 30 | 9 | 0 |
| 30 | 10 | 0 |

Keep in mind that the *FormA* data set contains records for multiple cross-border sites.

**Question 3.** How many records in your *CheckSpotIDs* data set have a **flag\_spotid** value of 1? \_\_\_

**Question 4**. How many cross-border sites have any records that have a **flag\_spotid** value of 1? \_\_\_

## Part III. Identifying Discrepancies in Double-entered Data

You determine that the error in the cross-border site Kirongwe (site ID: 20) will be problematic for future data processing, and so you request that the Kirongwe team re-key their Form A data. They re-enter the values and send them to you in the file “KirongweA.txt.”

Explore whether there are any discrepancies between the original data from Kirongwe and this revised data file.

1. First, read the “KirongweA.txt” data file in as a SAS data set. Name your new SAS data set *KirongweA.*
2. To facilitate the comparison of the data files, sort both the *KirongweA* and *FormA* data sets in order of ascending **spotid.**
3. Using one PROC (not a DATA) step, compare the *KirongweA* SAS data set to the subset of records for Kirongwe (only) in the *FormA* SAS data set. In your PROC step, create an output data set called *kcompare* that contains records from both the sorted *FormA* data for Kirongwe and the *KirongweA* data set. Include all records from both data sets in your *kcompare* data set (i.e., write observations to the output data set even when all values are judged equal). Be sure to examine your log after running the PROC. (As desired, feel free to explore any other options for the PROC you use to create this data set.)

**Question 5**. What procedure did you use to create the *kcompare* data set? (Enter only one word.) PROC \_\_\_\_\_

1. Print your *kcompare* data set. Take a close look at the data, noting which observations are from the original *FormA* data set and which are from the re-keyed *KirongweA* data. Based on what you see in *kcompare*, consider what might have gone wrong during assignment of the Spot IDs in the *Form A* data set.

**Question 6**. Other than the **spotid** variable, are there any other variables in the two data sets that have with inconsistent values between the *FormA* data vs. *KirongweA*? Assume that the KirongweA data set and the *FormA* data for Kirongwe contain data for the same set of spots (i.e., there are no missing or repeated records). Yes/No

At your request, the Kirongwe supervisor verifies all discrepant values against the paper-based records that had been keyed in. They confirm that the values in the new *Kirongwe* data set are the correct ones.

1. Delete the original Kirongwe records from the *FormA* data set, and replace them with the records from *KirongweA*. Name your new and improved *FormA* data set *FormA2*.

**Question 7**. How many observations are in your *FormA2* data set? \_\_\_\_

Use the *FormA2* data set moving forward.

## Part IV. Creating a Unique Identifier

The CBIHS used multi-stage sampling to implement the PLACE method. The spots listed in Step 1 of the PLACE method formed the sampling frame used to sample spots to be verified and characterized in Step 2, and the spots verified and characterized in Step 2 formed the sampling frame to select spots for Step 3.

Because of the study design, it is pertinent to merge, for each spot, the records collected across the three PLACE steps. To merge the *FormA2, FormB,* and *FormC* data, you will first need to construct an identifier that is unique to each spot. Thinking that **spotid** is that variable? Think again. Remember: **spotID** values are unique among spots in a single cross-border site, but **spotid** values are not unique among records from multiple cross-border sites. Recall that every cross-border site, the protocol was to assign the first spot on the list a **spotid** of 1, the second listed spot a **spotid** of 2, and so on.

1. First, write three DATA steps (one for each data set *FormA2, FormB,* and *FormC*), in which you create a new variable **sitespotid** to serve as your unique spot identifier.
   1. Assign **sitespotid** values according to the following structure:

<2-digit cross-border site ID><first letter of country where spot is located><hyphen><3-digit **spotid**>

For example:

* A spot in Malaba (site ID of 1) in Kenya that has the **spotid** 33 should be assigned the **sitespotid** value: 01K-033.
* A spot with **spoid** 122 in Katuna/Gatuna (site ID of 3) that is on the Rwandan side of the border should be assigned the **sitespotid** 03R-122.

When assigning the country letter, be mindful not to assume that the country is consistent across all spots in a site. Because of the cross-border focus of the study, some cross-border sites included spots in more than one country.

The table below summarizes the variables relevant to creating the **sitespotid** in each of the data sets.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data set** | **Relevant variable** | | |
| **Cross-border site** | **Country** | **Spot ID** |
| *FormA2* | a\_siteid | a11 | spoitid |
| *FormB* | b4 | b3 | b5 |
| *FormC* | c5 | c4 | c6 |

* 1. Output new Form A, B, and C data sets that contain the **sitespotid** variable. Name your output data sets *FormA3, FormB2,* and *FormC2* (based on the input data sets *FormA2*, *FormB*, and *FormC*, respectively).

1. Write one macro called idcheck that can be applied to any data set (*FormA3, FormB2,* or *FormC2)* to perform the following tasks:
   1. Verify coding of the **sitespotid** variable by separately comparing the assigned **sitespotid** values to the values of each of the variables that went into constructing it (in the data set for which the macro is called)
   2. Print all records with missing **sitespotid** values.

Ensure that the macro is sufficiently generalizable to be run on the *FormA3, FormB2,* or *FormC2* data sets.

1. Call the idcheck macro you wrote to check for missing **sitespotid** values in each of your new data sets: *FormA3, FormB2,* and *FormC2.*

**Question 8.** Across *FormA3, FormB2,* and *FormC2*, how many observations have a missing value for **sitespotid**? \_\_\_

Look through the output produced from each call of the idcheck macro.

* Did you code the **sitespotid** variable correctly? If not, go back and fix your coding in Step IV.1.
* Are there any missing values for **sitespotid**?

It is important to verify that we don’t have any missing values for **sitespotid** before merging by that variable; otherwise, records with a missing **sitespotid** may be merged together across the data sets, even if the records are not actually for the same spot. (To learn about this and other common merge mistakes, see [this SUGI](http://www2.sas.com/proceedings/sugi23/Advtutor/P47.pdf) and [this paper](https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/1746-2018.pdf).)

1. Investigate any missing **sitespotid** values. Consider the multistage design of this PLACE study and how you might leverage other data you have for a given spot to determine what any missing **sitespotid** should be.
   1. Consider the DATA steps you wrote in Step IV.1. Modify the DATA step(s) that created the *FormA3, FormB2,* and *FormC2* data sets as needed to assign an appropriate (non-missing) **sitespotid** value to any record where **sitespotid** was found to be missing**.**
   2. If you hard-code or overwrite any values, add comments to your code stating the original value(s) and providing your rationale for the new value assigned.
   3. Call the idcheck macro again for any updated data set(s). Look again at the log and/or output from the macro to verify that now, no **sitespotid** values are missing.

**Question 9.** Enter the revised (corrected, updated, non-missing) **sitespotid** for the spot from the *FormC2* data set that originally had a missing value for **sitespotid**. (Type carefully!) \_\_\_\_\_\_\_\_

Consider whether repetition of **sitespotid** values is appropriate for any of your new data sets, *FormA3, FormB2,* and *FormC2.*

**Question 10.** Considering the study design and data structure, in which data set(s) do you expect to see repetition of **sitespotid** values?

1. FormA3
2. FormB2
3. FormC2
4. For each data set where you do not expect repetition of **sitespotid** values, write and execute one PROC step that allows you to verify whether there is any duplication of **sitespotid** values.

**Hint:** One approach is to use the NODUPKEY option in a PROC that is very familiar to you (albeit for a different use).

**Question 11**. Consider the data set(s) that you determined should not have any repetition of values in the **sitespotid** variable. In those data sets, are there any erroneously repeats of **spotsiteid** values?

1. Yes
2. No

## Part V. Merging Data Sets

Now that you’ve created a common variable **sitespotid** uniquely identifying spots across the three data sets (*FormA3, FormB2,* and *FormC2*), you’re ready to merge all the records together, spot-by-spot.

1. Merge the *FormA3, FormB2,* and *FormC2* data together by **sitespotid** to create an output data set called *ABC* according to the following instructions:
   1. Keep all records from the *FormC2* data, and link, onto each of the FormC2 records, the corresponding spot-level records from the *FormA3* and *FormB2* data sets.
   2. Exclude, from the *ABC* data set, any observations that appear in *FormA3* or *FormB2* which do not have their **sitespotid** represented in the *FormC2* data set. (In other words, only include records for the spots where bio-behavioral interviews occurred. Exclude all spot data for spots where bio-behavioral interviews did not occur.)
   3. For spots where multiple bio-behavioral interviews were conducted, merge the *FormA3* and *FormB2* data sets for that spot onto every bio-behavioral interview record from that spot. (For example, if 6 people were interviewed at a spot with **sitespotid** 01K-033, you should have 6 records in the *ABC* data set – one per Form C interview – and each of those 6 records should include 01K-033’s spot-level data from *FormA3* and *FormB2*.)
   4. Output only the following variables to the *ABC* data set:

a11 a17 a18 a19 a20 a21 a22 a23 a24 a\_siteid a\_totalci b1 b3 b4 b5 b15 b16 b17 b18 b19 b20 b23 b24 b29 b30 b49 b50 b10a b14a b14b b14c b14d b14e b14f b52a b52b b52c b52d b9a b9b c1 c2 c3 c4 c5 c6 c12 c13 c14 c15 c16 c18 c19 c21 c22 c23 c24 c25 c32 c84 c85 c116 c10a c10b c10c c10d c117b c117c c11a c120a c120b c120c c120d c120e c120f c120g c120h c17a c20a c20b c20c c29a c29b c34a c34b\_v1 c34b\_v2 sitespotid spotid

1. Verify that your output data set *ABC* contains the correct number of variables (it should have 88).

**Question 12.** How many observations are in your *ABC* data set? (Be sure you’ve followed the Step V.1 instructions carefully!) \_\_\_

**Question 13.** Would SAS have produced an output data set if you had excluded the BY statement in the DATA step you wrote in Part V.1?

1. Yes
2. No

## Part VI. Applying Labels and Formats

Creating labels and user-defined formats for a large data set can be cumbersome. Fortunately, a colleague has shared two SAS program files for these tasks: “CBIHSApplyLF.sas” and “CBIHSFormats.sas.”

1. Open these two SAS programs just to see what they contain. Exit the files once you have an understanding of their contents. (Do not run any code while you have these program files open in SAS.)
2. Return to your SAS program for Project 3. Write and execute exactly one statement in your SAS program that will execute all the code in the “CBIHSFormats.sas” file.

**Question 14.** What statement did you use to execute the code in the “CBIHSFormats.sas” file? Enter the SAS keyword for the statement (enter one word, including any relevant special characters): \_\_\_\_

1. In a new DATA step, use one statement to execute the contents of “CBIHSApplyLF.sas.” (Do not copy/paste the formats and labels into your Project 3 code.)
   1. Name the output data set *ABC2*.
   2. Do not worry if the log alerts you to variables that are not included in your subset of the CBIHS data.

**Question 15.** How many statements did you write in your Project 3 SAS program to accomplish Step VI.3? (Be sure to include the DATA and RUN statements in your count.) \_\_\_\_

1. Inspect the variables attributes in your new data set, *ABC2*.

**Question 16.** How many of the 88 variables in *ABC2* do not have a label? (In your count, include only variables for which the label attribute is completely empty/blank.) \_\_\_\_

1. In a new DATA step, create an output data set named *ABC3* in which you:
   1. Assign informative variable labels to any yet-unlabeled variables (as identified in Question 16). Remember: you have been given the questionnaires that show the questions associated with each variable.
   2. Assign the format CBSITE\_REVISED. (you loaded this format into the WORK library in Step VI.2) to the variable **a\_siteid**.

## Part VII. De-identifying Data

Another important task in managing a data set can data de-identification. As you can see by looking at the data collection forms (“FormA.pdf,” “FormB.pdf,” and “FormC.pdf”) the full CBIHS data sets contain sensitive data. In addition to HIV test results and viral load data, the data sets include indications of behaviors that can be stigmatizing and illegal.

When de-identifying data, you’ll need to consider which variables will need to be dropped or masked. You’ll need to exclude direct identifiers, of course, but you’ll also need take precautions to prevent deductive disclosure of participants. The study design and setting should be considered when making plans to de-identify data. In a venue-based survey such as this, for example, the combination of a spot name or location data along with age, sex, and an indicator of employment at the spot could be sufficient to identify a respondent. A strategy for de-identifying a data set will typically weigh, among other factors, the risks of identification, the data safeguards to be used, and data use agreements.

If your data set contains identifiers, you could de-identify your data in a variety of ways. You could exclude the relevant variables or delete sensitive values contained in them; you might recode values into coarser categories; or you may use offsets to mask the values, preferably in a way that maintains the statistical properties of the original data. In the data set you received, for example, I applied a random offset to the date variable **c2**.

If the data collection tool for your study contains open-ended fields, your de-identification process should include manual checks for sensitive values in these fields. Be sure to check even fields were not intended to contain sensitive data. Participants or study staff could have entered more information than requested, or they may have recorded sensitive information in an incorrect field.

One way to efficiently check for sensitive values in open-ended questions is to generate frequency tables listing all the values of character variables in your data set.

**Question 17.** Fill in the blank with just one term (and not a macro variable) that will provide one-way frequency tables for all character variables in the *ABC3* data set:

PROC FREQ DATA=ABC3;

TABLES \_\_\_\_\_ / LIST MISSPRINT;

RUN;

1. Run the PROC step you completed in Question 17.
2. Consult [this list](https://research.unc.edu/files/2012/11/ccm3_018987.pdf) of direct identifiers from UNC’s IRB office. Look through the values from your Step VII.1 output and note whether the data values contains any direct identifiers that are included on that list.

**Question 18.** Does *ABC3* contain any direct identifiers listed on the UNC IRB document provided?

1. Yes
2. No
3. In a new DATA step, create a new data set named *ABC4*, based on the following instructions:
   1. If you determined that there were no direct identifiers in the data set, output *ABC3* to *ABC4* without any changes.
   2. If you did find identifiers in *ABC3*, suppress any identifying data values by replacing them with a SAS missing value appropriate for the variable type. (Consider referencing a participant’s values for **c13** and/or **sitespotid** to identify individuals whose sensitive values are to be overwritten). Be sure to add a comment in your code explaining what you did.

(Note: Any identifiers in the *ABC3* data set are simulated for this project; they are not true values.)

1. Your study participants have trusted you with their personal information, likely with little direct benefit to them. We owe it to our participants to be scrupulous when de-identifying data. If you made any changes to the *ABC4* data set, re-check the character variables to ensure that your data set is now properly de-identified.

## Part VIII. Creating a Codebook

Knowing that you have not received a codebook for this data set (you only have questionnaires), you are dreading the task of creating one from scratch. Luckily, a colleague has shared a macro that promises to make this task easier.

1. Open the program “codebook.sas.” There are detailed comments near the top of the program explaining various parameters you can specify for the macro.
2. Before you can call the codebook macro, you’ll first need to define this macro in your SAS session. Within your Project 3 SAS program, write one statement that will execute the contents of the “codebook.sas” program.
3. Call the macro, specifying the data set name and increasing the widths of the label and format fields to ensure the labels and formats are not truncated in the output (consult the “codebook.sas” program for instructions on how to specify these and other parameters for the macro). You may specify additional parameters for the macro to further customize the codebook if you would like, but you are not required to.
4. Use an ODS destination statement to output the codebook to a PDF file. Name your codebook file “Project3\_Codebook\_GroupName.pdf.”

## Part IX. Repetitively generating reports

The implementing partner for the CBIHS has requested some preliminary results briefs to support their local data dissemination meetings at each cross-border site. To provide a consistent depth of information and reduce production time, you and the local PI agree on a standard set of results to be reported for each site.

One result the PI is keen to explore is cross-border utilization of health facilities. Anticipating this interest, the bio-behavioral questionnaire (Form C) included the prompt: “In the past 12 months, have you received any services at a health facility in…” and asked whether services were received in Burundi, Rwanda, Tanzania, Uganda, Kenya, the Democratic Republic of the Congo (DRC), or any other countries.

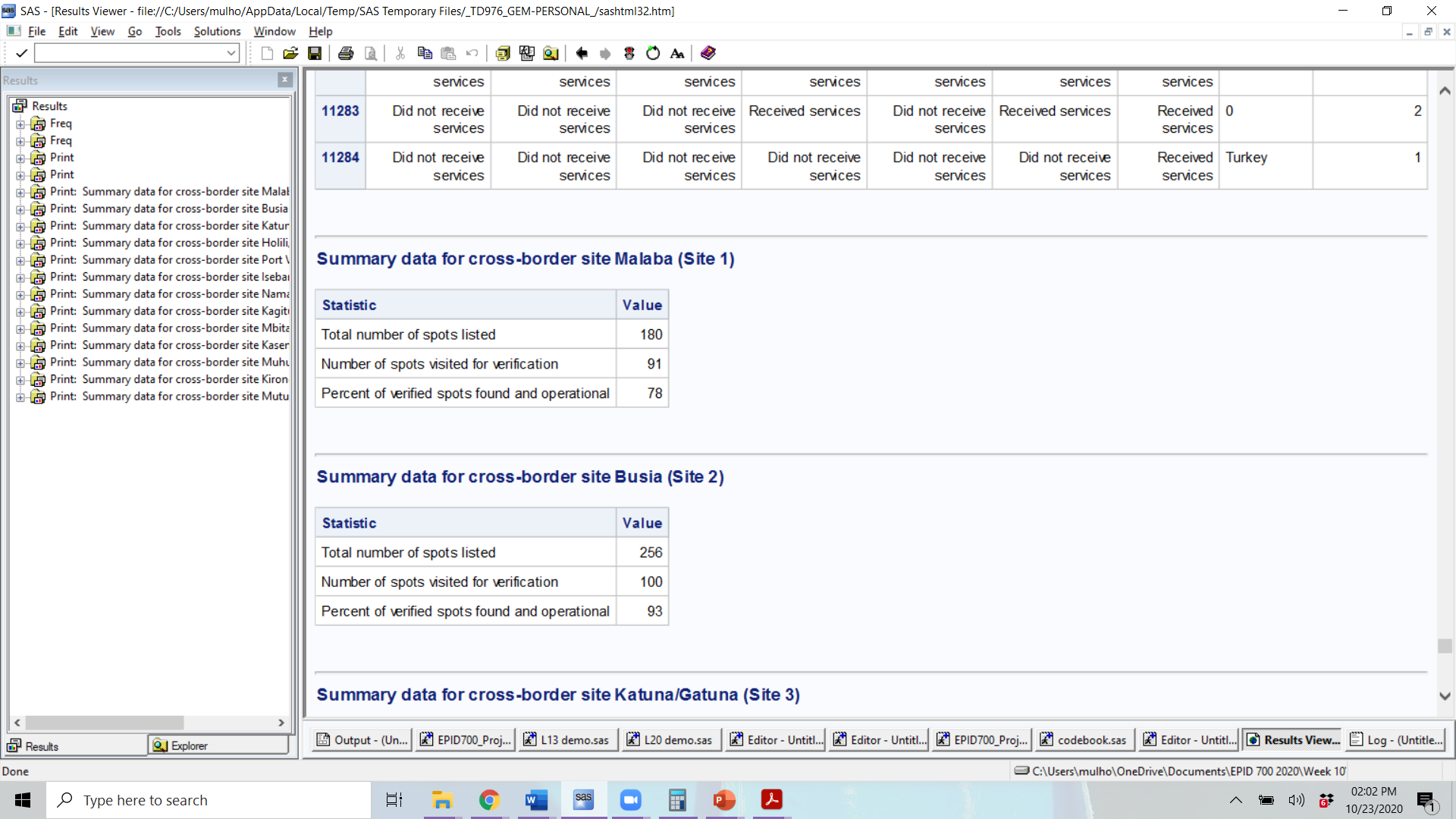
1. Among only those who consented to answering the bio-behavioral survey questions (in Form C), create a new variable, **hfcountrycount**, which is equal to the number of countries in which the respondent reported receiving health services in the preceding 12 months. Follow these instructions:
   1. Use a DATA step, and name your output data set *ABC5*.
   2. Use an ARRAY and a DO loop (and any other statements you need) in your DATA step.
   3. If *any* value among **c120a, c120b, c120c, c120d, c120e, c120f,** or **c120g** is 999 (refused) or missing: **hfcountrycount** should be assigned a missing value
   4. If no value among **c120a, c120b, c120c, c120d, c120e, c120f,** or **c120g** is 999 (refused) or missing: **hfcountrycount** should be set equal to the number of countries in which the participant reported receiving health services.
      1. For each variable **c120a** through **c120f** that has a value of 1 (indicating that service was received in the country inquired about it that variable), increase the value in **hfcountrycount** by 1.
      2. **c120g** indicates if services were received in a country other than those specifically asked about in **c120a** through **c120f**. Respondents who said that they received services in any additional countries were asked to provide the names of those countries in **c120h**. For these respondents, we need to review the values in **c120h** and manually update **hfcountrycount** according to the number of countries listed. To do this, perform the following steps for any participants where **c120g**=1:
         1. Use a conditional PUT statement to print, to your log, the values in **sitespotid**, **c13** (participant ID) and **c120h** (the other countries in which services were received) for each participant.
         2. Based on the notes you find in the log, use the **sitespotid** and **c13** values to (conditionally, for the appropriate records) add the number of additional countries named in c120h to the value in **hfcountrycount**.
         3. Assume that any numeric responses in **c120h** indicate the number of additional countries in which services were received. If **c120h**=0 (or “O,” potentially intended as a 0), assume that the interviewer had erroneously selected “yes” to **c120g**, and do add anything to the value in **hfcountrycount**.
   5. Given that you manually checked the values in **c120** and hard-coded changes to the **hfcountrycount** variable, add a comment to your code with a prominent reminder that values in **c120** and **hfcountrycount** must be re-checked and possibly updated if the source data files change.

**Question 19.** For many observations did SAS print a note to the log (as per Step 1.d.ii.2)? (You may add another note to your log in which you calculate this value, if you feel so inclined.)\_\_\_\_

1. Moving forward with the report, use any number of SAS procedures and DATA steps to create and output the following summary table, with no other output, for the subset of data collected in Malaba (cross-border site ID = 1):

|  |  |
| --- | --- |
| **Statistic** | **Value** |
| Total number of spots listed |  |
| Number of spots visited for verification |  |
| Percent of visited spots found and operational |  |

* 1. Consider the study design and structure of your data sets carefully to determine which data set (among *Form A3, FormB2*, *FormC2,* or *ABC5*) is most appropriate for calculating the requested statistics.
  2. For added clarity:
     1. “Total number of spots listed” = The total number of unique spots in the cross-border site listed by community informants in Step 1 of the PLACE method
     2. “Number of spots visited for verification” = The total number of spots visited for spot verification in Step 2 of the PLACE method, regardless of the outcome of verification (i.e., regardless of whether the spot was ultimately found, closed, a duplicate spot, etc.)
     3. “Percent of verified spots found and operational” = Among all spots visited for spot verification in Step 2 of the PLACE method, the percent of spots that were classified as “found and operational.” **Hint:** see **b9a**.
  3. In your SAS code, round all numeric values to be output to this table to the nearest integer. (Do not round by hand.)
  4. Your output should contain only the summary table above (i.e., the headers and three rows of information). For example:



**Hint:** You may wish to produce the relevant statistics, output these to output data sets, clean up the output data sets and then merge them together as needed. Then, you could specify the NOPRINT option in all PROC steps except your final PROC PRINT step, to suppress the output from intermediate PROC steps (i.e., to suppress the output from the PROC steps where you’re actually doing the calculations for the summary table).

**Question 20.** According to your summary table, what was the total number of spots listed in **Busia**? N = \_\_\_

**Question 21.** According to your summary table, how many spots were visited for verification in **Busia**? N = \_\_\_

**Question 22**. According to your summary table, what percentage of spots visited for verification in **Busia** were found and operational? Report the percentage to the nearest integer, as shown in your table. (Do not enter the percent sign.) \_\_%

1. Run the following code to create some additional analysis variables and then generate a well-formatted table using PROC TABULATE:

**DATA** ABC6;

SET ABC5;

IF c10c=**1** THEN count=**1**;

IF c11a=**1** THEN consentHIV=**1**;

IF c117b=**1** THEN HIVpos=**1**;

IF **4**<=c15<=**6** THEN secormore=**1**;

IF **1**<=c15<=**6** THEN edudata=**1**;

IF c18=**1** THEN fisher=**1**;

IF c18 IN(**1**,**2**) THEN fisherdata=**1**;

**RUN**;

**PROC** **TABULATE** data=ABC6 STYLE=[JUST=CENTER];

WHERE c10c=**1** AND c5=**1**;

CLASS c10c c14 c11a;

VAR count consentHIV HIVpos secormore edudata fisher fisherdata c12 hfcountrycount;

TABLE (count="Total respondents")\*(N='N'\*f=**8.**)

(HIVpos="Reactive HIV test")\*(N='N'\*f=**8.** PCTN<consentHIV>='%'\*f=**12.2**)

(secormore="Secondary education or higher")\*(N='N'\*f=**8.** PCTN<edudata>='%'\*f=**12.2**)

(fisher="Works in fishing industry")\*(N='N'\*f=**8.** PCTN<fisherdata>='%'\*f=**12.2**)

(c12="Age")\*(MEAN='Mean'\*f=**12.2** STD='SD'\*f=**12.2**)

(hfcountrycount="Countries received services in\*")\*(MEAN='Mean'\*f=**12.2** STD='SD'\*f=**12.2**)

,

(c14 all) / STYLE=[JUST=center];

TITLE "East Africa Cross-Border Integrated Health Study (2016-2017)";

TITLE2 "Unweighted results from Malaba (Site 1)";

FOOTNOTE "\*Number of countries in which health services were received in preceding 12 months";

**RUN**;

**Question 23**. According to the table you just produced, among **female** respondents in **Malaba**, what was the mean number of countries in which health services were received in the preceding 12 months? Report the mean to 2 decimal places. Mean = \_\_\_

1. Convert your code from Step IX.2 and Step IX.3 into a macro called sitereport that will output both the summary table and PROC TABULATE table for any specified site.
   1. In your macro, make sure to use some TITLE statements that state the site name and site ID.
   2. You can copy/paste your code from Step IX.2 and Step IX.3 or just go back and update it so that the code from these steps can all be run together as a macro. Be sure to change all references to site number and site name variables to a macro variable (e.g., you might change **a\_siteid**=1 to something like **a\_siteid**=&siteid), whose value you can assign when the macro is called for a certain site.

**Hint:** The PROC TABULATE code I gave in Part IX.3 has two references to site ID (in the CLASS statement and the TITLE2 statement) and one reference to the site name (in the TITLE2 statement). You’ll need to replace the value given for **c5** (i.e., 1, the site ID for Malaba) with a macro variable for the site ID, and you’ll need to replace the site name “Malaba” with a macro variable for the site name.

1. Run your sitereport macro code, %MACRO to %MEND, to define the sitereport macro.
2. Call your site-by-site report macro for all 13 sites, exporting the output to a single, well-formatted PDF file with the output centered and the results for each site on a new page. To do this:
   1. Execute the statement: OPTIONS CENTER; in open code to center your output.
   2. Write an ODS destination statement to write your output to a PDF. Apply the ODS style JOURNAL in that ODS statement.
   3. Your goal is to have one page per site, with both tables for a single site appearing on one page To do this, include the STARTPAGE=NEVER option in the ODS statement. Then, add the following ODS statement in your sitereport macro (it’ll run in open code, like other ODS statements) to tell SAS where to begin printing to a new page: ODS STARTPAGE=NOW;
   4. After the ODS PDF statement, call your sitereport macro for all 13 sites.
   5. Immediately after the macro calls, close the ODS PDF destination.

Ensure that each page of output in your PDF includes the site name and site ID corresponding to the two tables shown on that page.

**Question 24**. According to your report, among the spots visited for verification in Kirongwe, what percent of visited spots were found and operational? Report the percentage to the nearest integer. (Do not enter the percent sign.) \_\_%

**Question 25**. According to your report, which cross-border site had the highest overall prevalence of HIV among bio-behavioral survey participants? Provide the site ID number for that cross-border site. (Interpretation note: I am requesting the prevalence only among participants because we have not yet weighted these data to account for refusals of the HIV test or for differential sampling probabilities across participants.) Site ID number: \_\_\_\_

1. Save the report you just generated as a PDF file named “Project3\_Report\_GroupName.pdf.”

**Part X. Export Data**

Now it’s time to pass the data set on to your collaborators. Your collaborators use STATA and Excel, so to keep things simple, you’ll export the data to a CSV file that can be read into either program.

1. Export your data set as a CSV file with formatted values (using the formats already applied – no need to add more). Name the data file “Project3\_Data\_GroupName.csv.” Check the output data set to be sure you’ve written the formatted values to the CSV file.
2. Next, encrypt the data file in a zip file for added security and faster upload/download times. Save your csv file to a zip file named “Project3\_Data\_GroupName.zip” and encrypt the file with the password: 2+2...isfive!(be sure to include the “!”). Click [here](https://www.northeastern.edu/securenu/sensitive-information-2/how-to-use-7-zip-to-encrypt-files-and-folders/) more detailed instructions on preparing zip files.

1. For more information on the PLACE method, see: https://www.measureevaluation.org/resources/tools/hiv-aids/place/place-files-1 [↑](#footnote-ref-1)