

SAS In The Classroom: Exploratory Data Analysis with SAS Studio

Jonathan W. Duggins, NC State University, Raleigh, North Carolina

Jim Blum, University of North Carolina Wilmington, Wilmington, North Carolina

ABSTRACT

SAS Studio®, and by extension SAS University Edition®, provide exciting opportunities for students early in their statistics education. By allowing students, and teachers, the ability to use a point-and-click, web-based interface students can calculate summary statistics, explore complex sampling distributions, and carry out inference procedures using an industry-standard analysis package but with minimal need for programming. We will discuss walking students through an activity intended to develop a graphics-based understanding of the sampling distribution for a mean.

At the conclusion of the session participants will know: the difference between tasks and snippets; how to create graphical and numerical summaries of a data set, how to draw samples from a population; and how to use macro variables to customize the code generated by a process flow.

INTRODUCTION

SAS Studio provides access to SAS via a web browser that connects to a SAS server. Since that server can be a local copy of SAS installed on your PC, a local network SAS server, or a cloud-based SAS server this provides a flexible and convenient way to interact with SAS. SAS University Edition allows users to access SAS via a virtual application that includes a SAS server in the installation. Regardless of the location of the SAS server, once it processes the user-submitted code the results are displayed via your web browser using the SAS Studio interface. As such, the term SAS Studio is used exclusively for the rest of this paper with the understanding that the end-user must select the best implementation for their classroom setting.

Regardless of implantation, SAS Studio provides an opportunity to introduce more high school students to statistical software to help them develop a deeper understanding of data analysis and help high school teachers implement portions of the newest Guidelines for Assessment and Instruction in Statistics Education (GAISE) Report. With the ability to provide drag-and-drop and point-and-click interfaces students can quickly begin exploring data using an intuitive interface. Attendees learn how to use SAS Studio to select random samples, explore data graphically, and generate a sampling distribution by creating a custom snippet.

SAS STUDIO BASICS

When opening SAS Studio users will notice the navigation pane, shown in Figure 1, on the left of the screen that provides access to files, libraries, and other useful resources such as tasks and snippets.

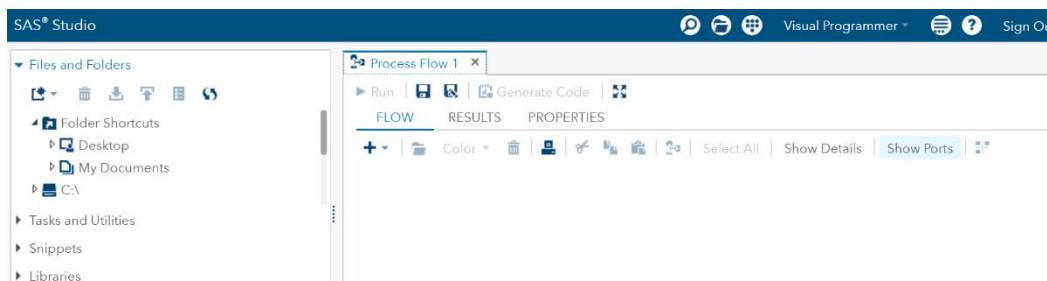


Figure 1. SAS Studio home screen. Navigation pane (left) and workflow pane (right).

The **Files and Folders** menu and **Libraries** menu allow navigation to access files and data sets for use during the current SAS session. The **Tasks and Utilities** menu gives users access to a wide variety of common programming elements (e.g. random sampling or data transposition) that allow output to be

generated via a point-and-click interface. Snippets are smaller programs designed to be inserted into the user's workflow. Unlike tasks, snippets are not editable via a point-and-click or menu-driven interface; instead their lines of code need to be edited manually. However, once edited the user has the option of saving the updated code as a custom snippet that will appear in the **Snippets** menu.

Users should also note they have the option to select the **Visual Programmer** or **SAS Programmer** perspective via a dropdown menu shown near the top right corner of the current session. (Top right, Figure 1.) The SAS programmer perspective allows users to create original programs from scratch or edit previously created programs. This perspective would be familiar to programmers with experience using SAS via the Interactive Windowing Environment but with the code, log, and results viewer windows now represented as tabs within the same web page. However, users are now able to add code to a program via a snippet by simply dragging it from the snippet menu into their program editor. When using the visual programmer perspective users will be able to build a process flow in the workspace pane, shown on right side of Figure 1. Process flows are built when the user drags an object, such as a task or snippet, from the navigation pane into the workspace. Each object creates a node that can be connected to other nodes; these connections then create the process flow. Participants will be walked through creating a process flow.

SCENARIO AND DATA SET

The National Center for Missing and Exploited Children (NCMEC) provides publicly available data that includes several variables related to open cases of missing children. The data set used here is from March, 2017. This data was made public as part of the Cloudera Child Finder Hackathon to prompt the development of new techniques for finding missing children. In this workshop participants will be guided through using SAS Studio to explore the data and use it to demonstrate some statistical principles appropriate for an AP Statistics course.

CREATING THE PROCESS FLOW

READING THE DATA

Participants have been provided with the NCMEC data and there are several methods for making it available for use. The most flexible is to create a new library to contain the input data and hold any output data sets that are created. To add a library, first go to the navigation pane and click on the **Libraries** menu. Next, click on the file drawer icon (indicated by the red circle in Figure 2) to bring up the New Library dialog box.

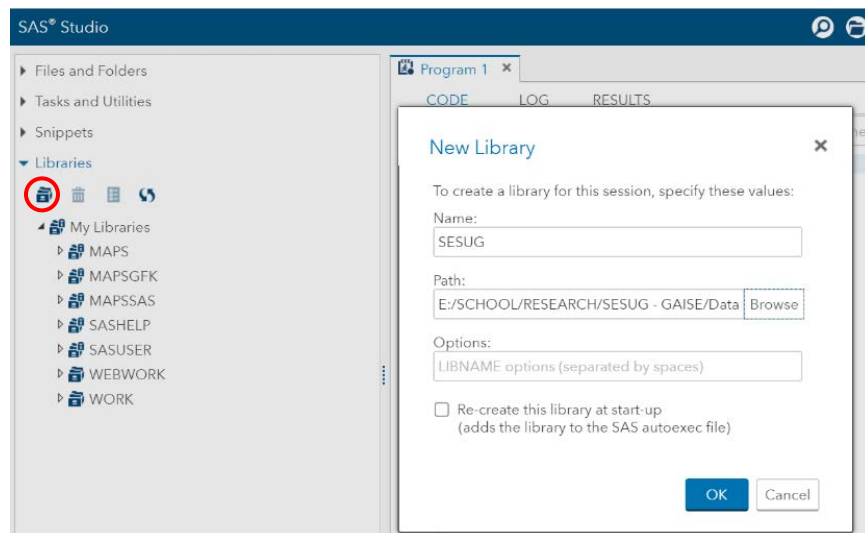


Figure 2. New Library dialog box.

In the dialog box include the information shown below to provide your library with a name and the physical location where your NCMEC data set is located. Once the information has been provided, click OK to add the library, which the author named SESUG, to the list of libraries shown in the Navigation Pane. Clicking on the newly added library will show the icon for the NCMEC data set.

TASKS

Now that the data has been made available participants can begin constructing the process flow. The first five nodes included in the solution to this activity come from the **Tasks and Utilities** menu. The first task is the histogram which is found by going to **Tasks and Utilities** → **Tasks** → **Graphs** → **Histogram**. Dragging the histogram task from this menu into the workspace adds the histogram as a node in the process flow as shown in Figure 3.

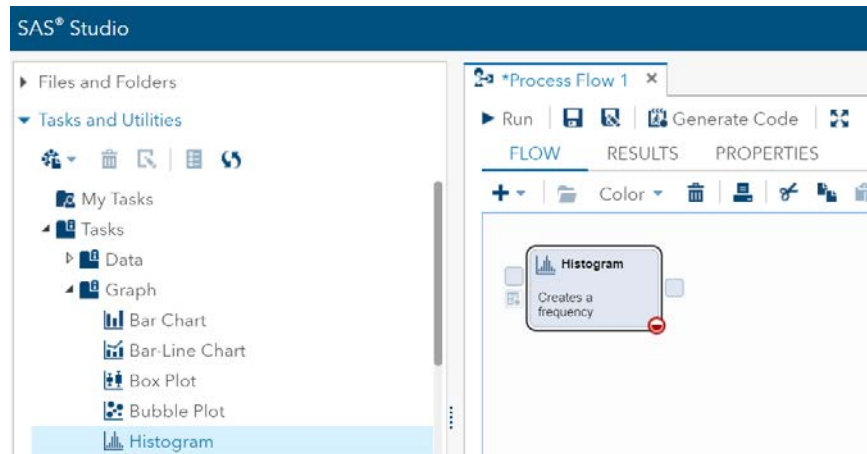


Figure 3. Histogram task as a process flow node

The red triangle in the circle located in the lower right corner of the node indicates it is not completed and still requires user input. Three smaller boxes on the edges of the node – two on the left and one on the right – appear as well. The lower left-hand box has a grayed out data set icon indicating that no data set has been assigned to the node yet. The remaining boxes are control ports. Control ports are used to indicate the order in which nodes should be included in the process flow. With only a single node in the process flow these ports are not needed yet.

To access the histogram node, double-click it (or right-click and select **Open** from the menu) and the workspace pane will be further split: on the left are the settings for this node and on the right is where SAS will display the code and results generated based on the user's choices in the settings. Figure 4 shows the panes from the author's SAS Studio session.

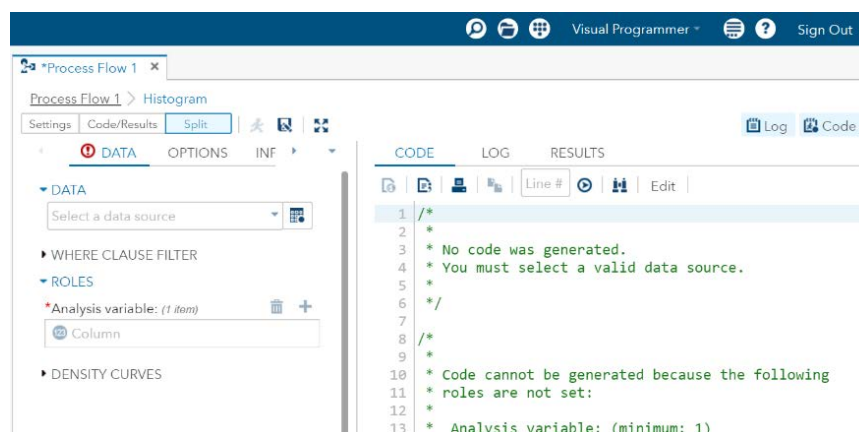


Figure 4. Histogram task with Settings (left) and Code/Results (right) displayed.

Users can select to see only the **Settings** or **Code/Results** pane by selecting the appropriate button at the top left of the session; **Split** is chosen by default to allow users to see both panes. The top of the settings pane includes four tabs: **Data**, **Options**, **Information**, and **Node**. The **Data** tab is selected with two settings expanded by default: **Data** and **Roles**. The Code/Results pane initially contains three tabs: **Code**, **Log**, and **Results**; the **Code** tab is active by default. Looking again at the **Data** tab users will also notice the red exclamation point indicating that SAS can't generate a histogram yet. The histogram task requires both a data set and an analysis variable be selected. The **Code** tab indicates that no code could be generated as a result; users will need to specify a data set and an analysis variable to proceed.

Under the **Data** setting the drop-down allows the selection of a recently-used data set and the icon to the right of the drop-down opens a dialog box where a user can navigate to any desired data set. SAS Studio will populate this field automatically with a recently used data set so it is important to ensure the correct data set is selected for each node in the process flow. Attendees should use the NCMEC data set in the SESUG library added above to begin the process flow.

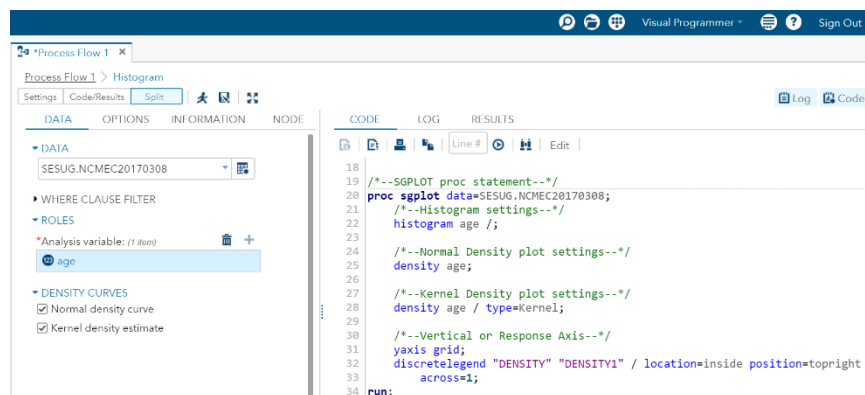


Figure 5. Histogram task showing Data tab selections and resulting code.

Users can now select a single response variable for the histogram via the plus sign next to **Analysis variable** under the **Roles** setting. This displays a list of the numeric variables available in the data set from the **Data** setting. (If the wrong variable is chosen the trashcan icon can be used to remove it.) Expanding the **Density Curves** setting shows two check boxes that can be used to request normal and kernel density estimates be overlain on the histogram. Users should note that as settings are customized in the settings pane the code/results pane is being edited automatically. In fact, the code in this pane is read-only.

At the top of the settings pane the user can now select the **Options** tab to customize the histogram. Customization options are shown in Figure 6 and include titles, footnotes, axis labels, and other aesthetic changes to the graph. Two settings have been changed here: the **Title and Footnote** setting is used to apply the title to the graph while the **Vertical Axis** setting is used to change the scale to proportion (the default is percent).

DATA **OPTIONS** INFORMATION NODE

• TITLE AND FOOTNOTE

Title:

Set title font size

*Font size:

Footnote:

Set footnote font size

*Font size:

• BIN DETAILS

• HORIZONTAL AXIS

• VERTICAL AXIS

Specify axis scaling:

Show grid

Show label

• LEGEND DETAILS

• GRAPH SIZE

Figure 6. Histogram task showing Options tab selections.

As shown in Figure 7, the **Information** tab is useful for reference if users are looking for further reading on SAS Studio or on the code used to support the node. Finally, the **Node** tab can be used to include information for the user's reference. A name, description, and notes can be associated with the node while the creation and last modified date of the node are recorded. For reference, Figure 7 shows this node is named HIST00. Now that this task has been edited to the user's liking the generated code can be submitted for compilation and execution. This can be achieved by using the F3 key or by clicking the "running person" icon above the settings pane. The histogram will now be generated in the **Results** tab under the code/results pane. Before proceeding any further it is good practice to view the **Log** tab first to ensure no errors, warnings, or notes occurred that would impact the validity of the results.

The figure consists of two side-by-side screenshots of the SAS Studio interface for a Histogram task. Both screenshots show the top navigation bar with 'DATA', 'OPTIONS', 'INFORMATION', and 'NODE' tabs. The left screenshot has 'INFORMATION' selected, showing a 'Properties' section with fields for Name (Histogram), Description (Creates a frequency distribution of a numeric variable), Category (Graph), Procedures (SGPlot), and Version (3.6). Below this is a 'Resources' section with links to SAS Studio Task Reference Guide, PROC SGPLOT Documentation, PROC SGPLOT Papers, and PROC SGPLOT Samples and SAS Notes. The right screenshot has 'NODE' selected, showing an 'Identification' section with fields for Name (HIST00), Description (Creates a frequency distribution of a numeric variable), Created date (8/27/2017, 2:12:39 PM), and Modified date (8/27/2017, 3:07:44 PM). Below this is a 'Notes' section with an empty text area.

Figure 7. Histogram task showing Information and Node tabs.

The **Log** tab displays a summary of the log file at the top and shows there were no errors, no warnings, and 3 notes. Users can click on the triangle icon to the left of the **Errors**, **Warnings**, and **Notes** subheadings to get a summary of any included items. Figure 8 shows a summary of the three notes – none of which provide a reason to distrust the results. Below the header the submitted code is displayed and the three notes are shown with the code that generated the note.

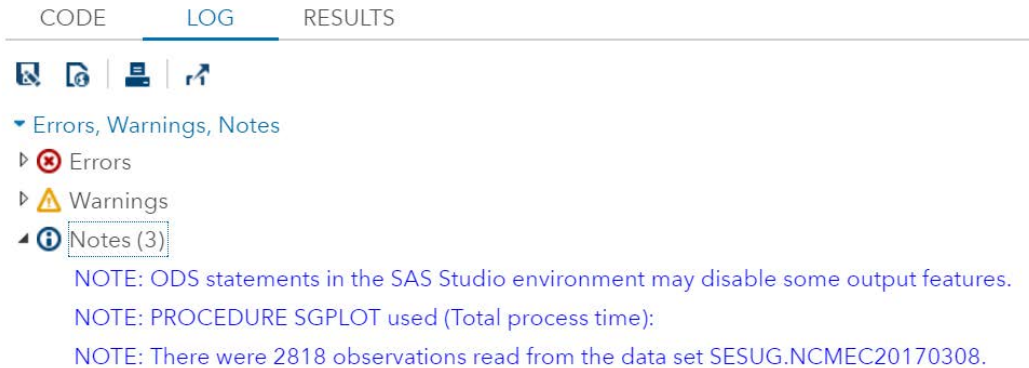


Figure 8. Header provided in Log tab.

Figure 9 shows the histogram for the data along with the two density estimates added in Figure 5. At the top of the **Results** tab users have the option to export the results in HTML, PDF, or RTF. The table of contents provides a quick way to navigate output when submitting code that produces a large number of results.

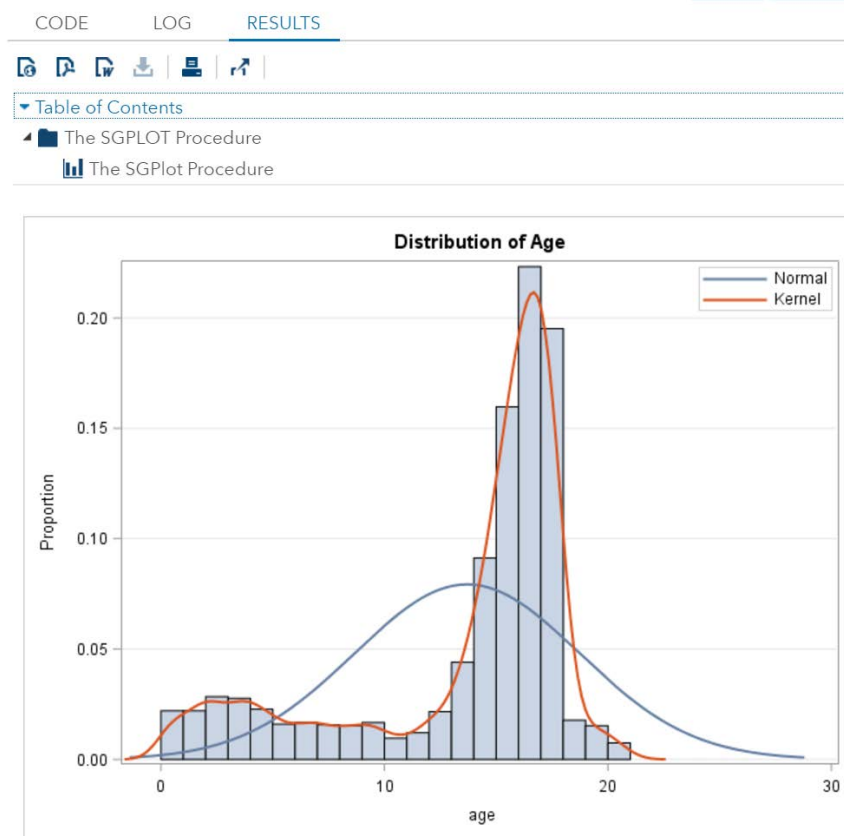


Figure 9. Histogram showing distribution of age with normal and kernel density estimates.

To return to the process flow in the workspace simply click on the **Process Flow 1** link above the settings pane. (Note that if you save your process flow the file name replaces the default **Process Flow 1**.) As shown in Figure 10, the HIST00 node will appear with updated graphics. The data set icon is no longer grayed out but instead is shown in blue since a data set was selected as part of the previous process. Additionally, the red circle containing a triangle is now a green circle with a check mark indicating that the node has been successfully submitted. Figure 10 also shows the addition of a new node.

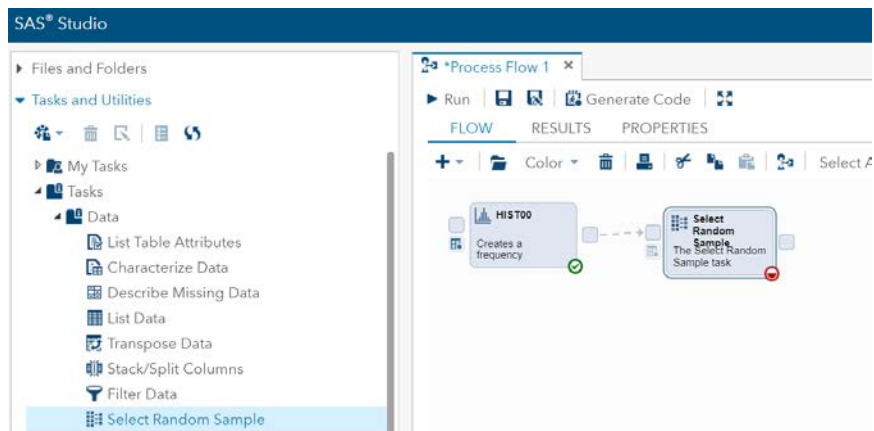


Figure 10. Process flow showing updated HIST00 node and attached Select Random Sample node.

The next task needed to complete the activity is Select Random Sample which is located under **Tasks and Utilities** → **Tasks** → **Data**. As before, the node is added by dragging the task into the work space. Now that two nodes are present they can be connected to indicate the order in which they should be executed. To connect them the user needs only to left-click (and hold) on the right control port in the HIST00 node and drag the cursor to the left control port on the select random sample node. A dotted gray arrow will appear showing the order in which the nodes would be executed.

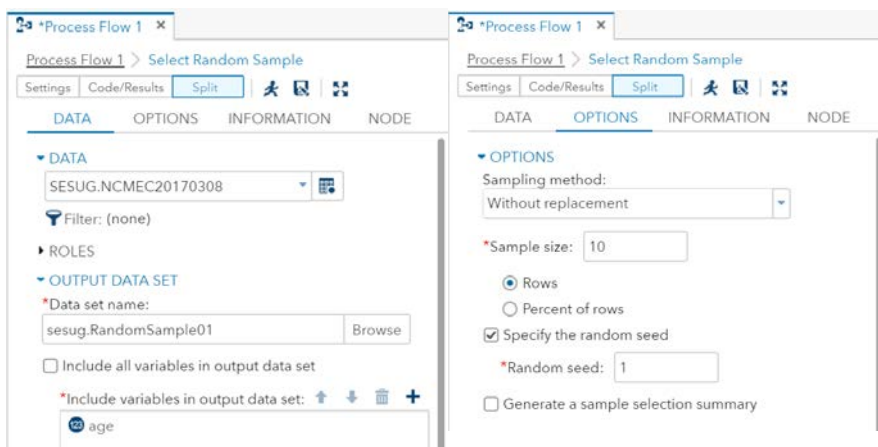


Figure 11. Data tab (left) and Options tab (right) for the Select Random Sample Node.

As before the Simple Random Sample node can be edited by double-clicking to access the settings pane. This pane has the same tabs as before but with different settings available. Selecting the data set in the **Data** setting gives users access to the data set's columns in the **Roles** and **Output Data Set** settings. To obtain a simple random sample the **Roles** setting is not needed since it allows for stratification during sampling. Users can specify a data set name, SESUG.RANDOMSAMPLE01 here, via the **Output Data Set** setting. Including the SESUG prefix stores this data set in the SESUG library for future use.

In addition to using the **Output Data Set** setting to name the data set users can also decide which variables to include. To simplify the output for this activity, only the response variable is included in the output data set. To achieve this, first uncheck the box for "Include all variables in output data set." Doing so introduces an additional setting where users can specify variables using the plus sign. The author

used the dialog box it produced to select only the variable of interest here. Figure 11 shows the selections made in the **Data** tab.

Figure 11 also shows the **Options** tab where the first setting, also named **Options**, selects the sample style. Sampling without replacement is used here and the default sample size of 10 is also kept. For the purposes of the paper the random seed is specified so users can replicate the results using SEED = 1. Unchecking the box for **Generate a sample selection summary** prevents this node from producing output in the **Results** tab of the Code/Results pane. As before, the **Information** tab contains useful resources and the **Node** tab can be used to customize the node's properties. For concreteness this node is named SRS01 but for economic use of space the **Information** and **Node** tabs are not shown. Running this node (running person icon or F3 key) adds a new tab, **Output Data**, to the Code/Results pane.

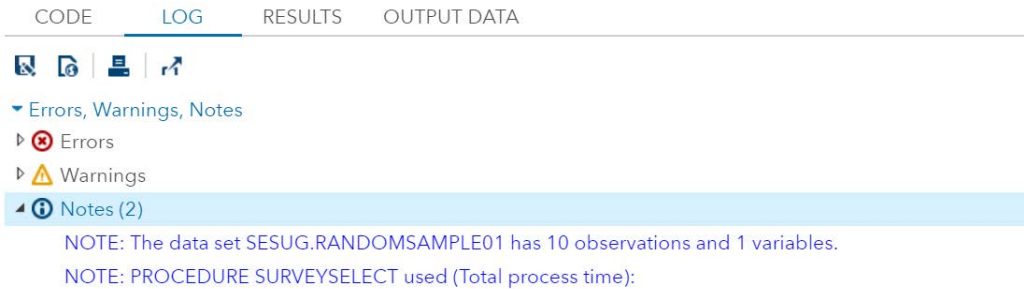


Figure 12. Log tab for Select Random Sample.

Users can verify in Figure 12 that the **Log** tab shows no errors, warnings, or worrisome notes. Users can also verify (using Figure 13) that the same 10 observations shown are obtained any time this node is submitted. (Be sure **not** to specify the seed for an in-class activity though to ensure students get different samples.)

	age
1	14.82
2	17.00
3	15.89
4	17.19
5	12.84
6	15.89
7	3.97
8	15.06
9	14.95
10	13.80

Figure 13. Output Data tab shows data generated by the SRS.

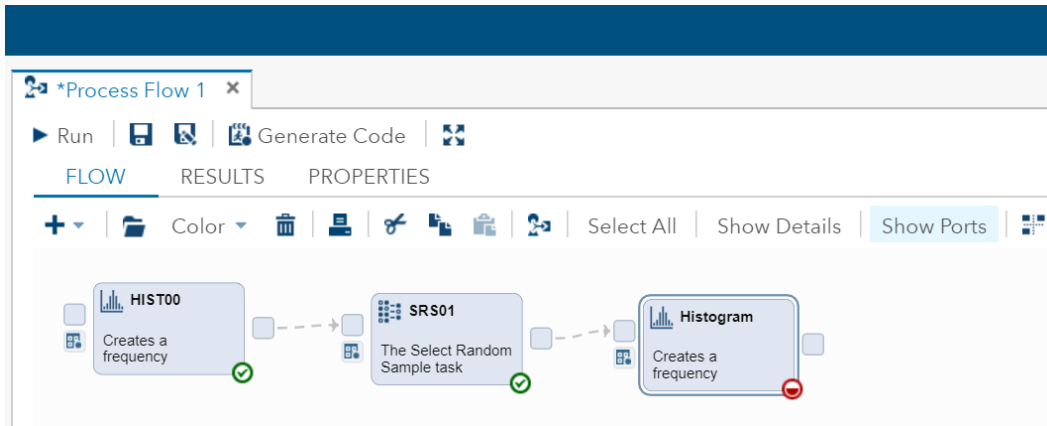


Figure 14. Process Flow workspace updated with new Histogram node.

Returning to the workspace another histogram node can be added as before and connected to the SRS01 node to ensure the nodes are executed in the proper order. Entering into the settings for the second histogram node, HIST01, the **Data**, **Roles**, and **Density Curves** settings can all be adjusted in the **Data** tab. Note that while the **Roles** and **Density Curves** settings are the same as in HIST00 the **Data** setting should reference the data set generated by the SRS01 node (SESUG.RandomSample01). For this reason it is important to run the SRS01 node before editing this histogram node; the data set must exist in order to select it and its columns for use in the current node. As with the HIST00 node the **Options** tab is used to apply a title and set the vertical axis to use proportions. The **Node** tab was used to name this histogram HIST01.

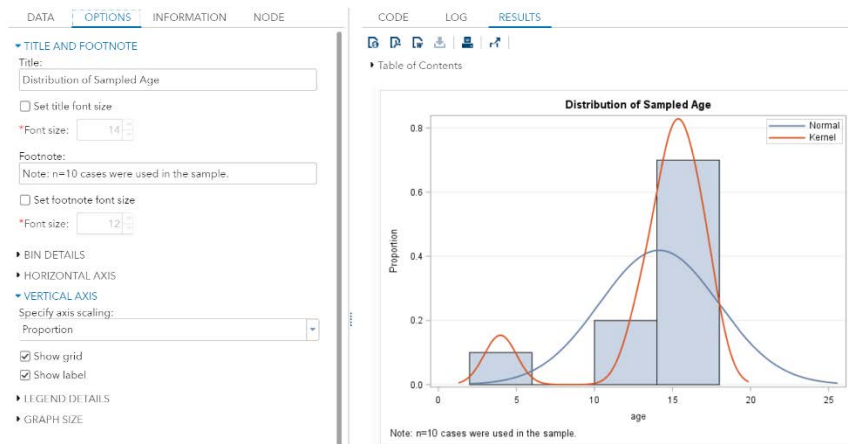


Figure 15. Options and Results tabs for HIST01 node.

Note that a footnote has been added to indicate the sample size used in the SRS01 node. Running this node generates a log almost identical to the one from the HIST00 node and a histogram with similar graphical elements such as density estimates.

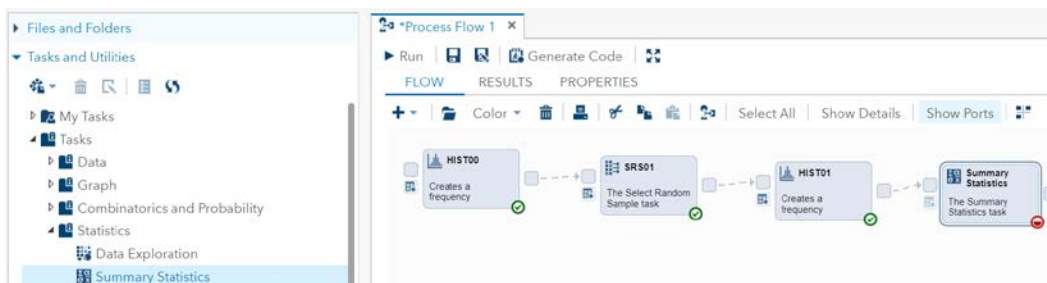


Figure 16. Summary Statistics Node from Tasks added to Process Flow.

Again, return to the workspace to add the fourth task, **Summary Statistics**, which is located under **Tasks and Utilities** → **Tasks** → **Statistics**. Connect this node, which will be named SUMM01, to ensure it runs after the HIST01 node.

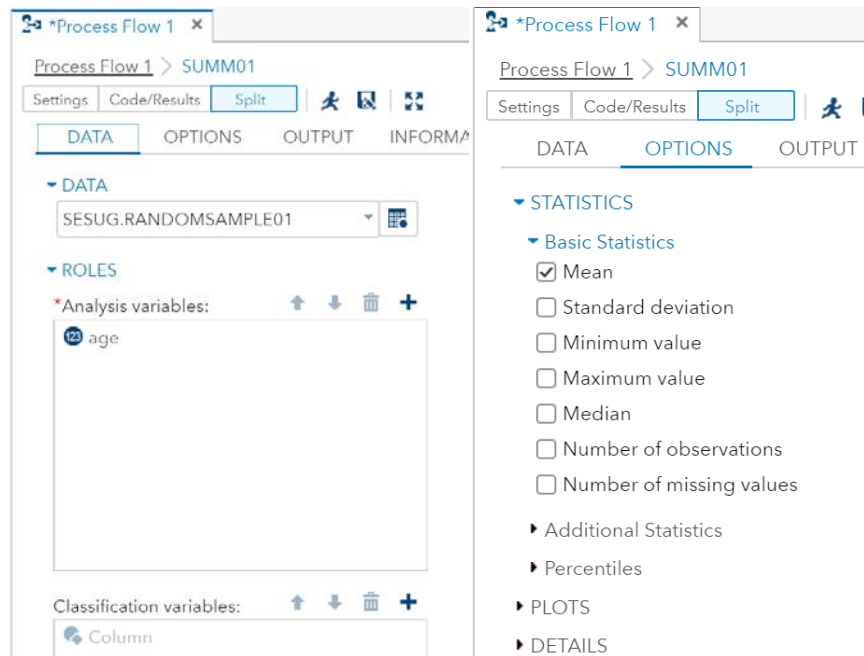


Figure 17. Data and Options tabs for the summary statistics node.

Opening this task for editing shows the expected settings: **Data** and **Roles**. Under the **Data** tab (Figure 17, left) use the **Data** setting to select the SESUG.RandomSample01 data set. Because no classification, grouping, or weighting is needed for this analysis the only setting to be changed under **Roles** is to select the response variable (AGE) as the analysis variable. As before, the plus icon allows users to select variables from a dialog box. Moving to the **Options** tab (Figure 17, right) the statistics of interest can be selected. For the purpose of this activity only the mean is needed; generalizations using other statistics are discussed below.

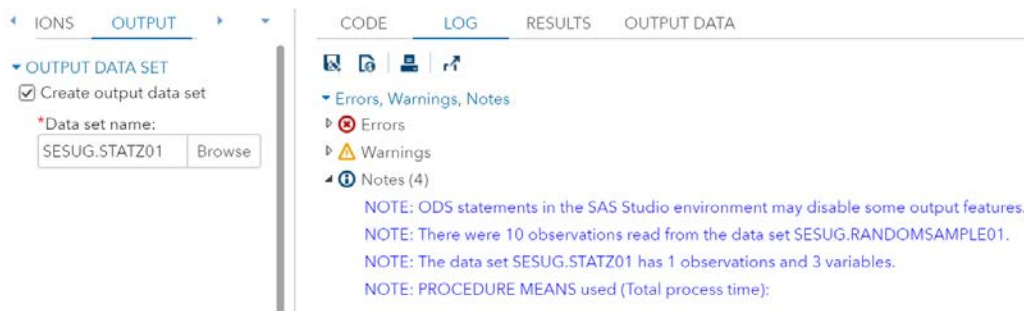


Figure 18. Output and Log tabs for Summary Statistics node.

The **Output** tab has not been available in previous tasks but can be used here to ensure the mean is not only displayed in **Results** tab under the Code/Results pane but that it is also saved to a data set. Thus the **Output Data Set** option should be used by checking the box and entering a data set name. The author named the data set STATZ01 and stored it in the SESUG library. After changing the desired options in the **Node** tab to name the task SUMM01 this task can be submitted. The **Log** tab indicates no errors or warnings and four notes.

Moving to either the **Results** tab or the **Output Data** tab the single mean of 14.14 is shown. This output data set is crucial as it is used as the basis for the next node. Without saving the mean to a data set participants won't have access to the value for programming purposes.

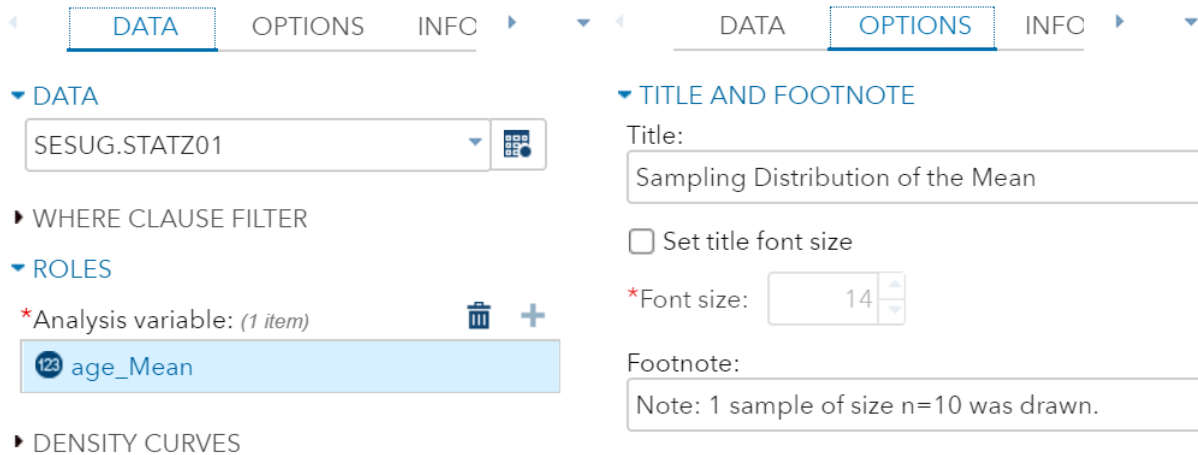


Figure 19. Data and Options tabs for HIST02 node.

Again add a histogram node, named HIST02 here, to create a histogram for the sampling distribution. As with the previous histograms Figure 18 (above, left) shows how to select a data set (SESUG.STATZ01) and an analysis variable (age_Mean). Do not apply the density curves to this histogram as that will result in an error. (Only one sample mean has been created so far and SAS can't fit a density estimate to a single value.) Use the **Options** tab (Figure 18, right) to apply the title, footnote, and to scale the vertical axis as before (proportion).

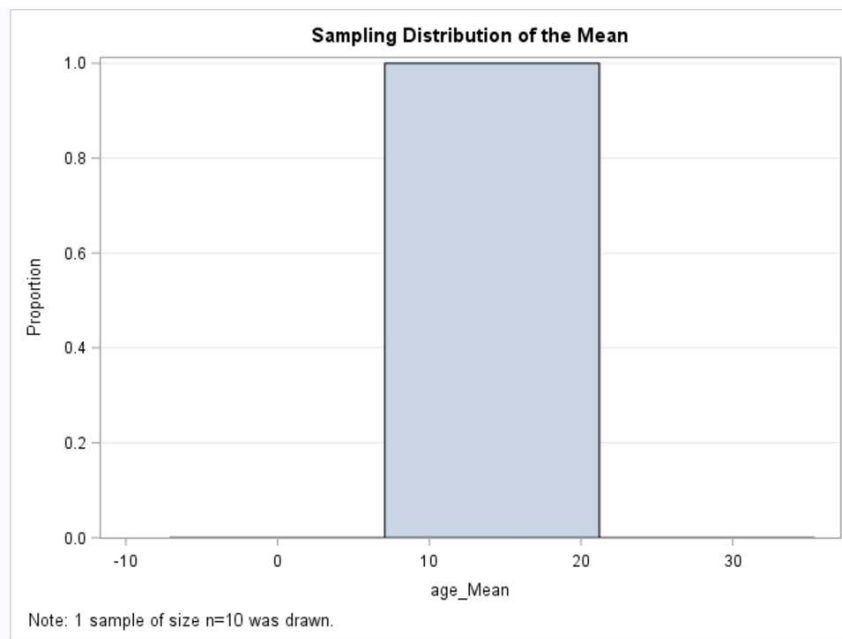


Figure 20. Histogram of the sampling distribution generated from a single sample.

The **Results** tab shows the resulting histogram. As expected the histogram of a single sample mean does not provide any insight into the sampling distribution. To truly produce a histogram that estimates the sampling distribution many samples would need to be taken. Unfortunately, the SRS01 node did not have an option that allows repeated sampling to be performed. However, there are several solutions of varying complexity which can be implemented by editing the code used by the SRS01 node.

SNIPPETS

As mentioned previously, tasks are prepackaged modules that allow the user to adjust settings in a menu-driven manner.

```
proc surveysselect data=SESUG.NCMEC20170308 out=sesug.RandomSample01 method=srs
    sampsize=10 seed=1 noprint;
    id age;
run;
```

Figure 21. Code from the SRS01 node that was automatically generated.

When tasks are completed their code is viewable in the **Code** tab of the Code/Results pane. A node in the process flow is simply an icon that represents those lines of code. Figure 21 above shows an example of the code generated as a result of a task, specifically the SRS01 node that used the Select Random Sample task as shown in Figure 11. Recall this task-generated code was read-only. One way to edit this code is via snippets.

Snippets also are used to create nodes, i.e. icons representing lines of code, in the process flow. The difference is that snippets can either be provided by SAS, provided by a third party (such as a teacher, classmate, or colleague), or written by the user.

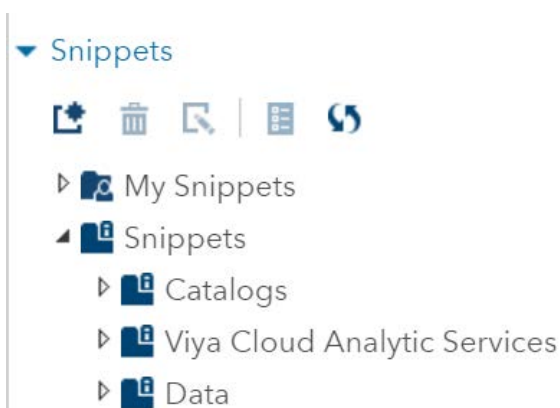


Figure 22. SAS-provide Snippet categories as shown in the Navigation Pane.

To view the ones provided by SAS simply click on the **Snippets** line in the menu on the left-hand side of the screen. As shown in Figure 22, users will notice two subheadings: **My Snippets** and **Snippets**. Third-party and user-created snippets would be stored under the **My Snippets** subheading while those provided by SAS are under the **Snippets** subheading. Exploring further shows that SAS includes snippets to read data, provide descriptive information, and even make graphs (including histograms). The histogram snippet requires users to edit code manually, rather than by interacting with menus, which is why the histogram task was used earlier.

Of interest here is the ability for users to create custom snippets. To begin making a custom snippet as part of this demonstration, return to the SRS01 node as if to edit it as shown in Figure 11 and select the **Code** tab in the Code/Results pane. In the menu just above the code a row of shortcuts is included and the final menu item simply says **Edit**. Clicking this menu item opens a new window that only has the **Code**, **Log**, and **Results** tabs; however, the code is no longer read-only. Users now have the ability to edit the code as needed; creating multiple samples, instead of one, is now possible.

```

1 proc surveyselect data=SESUG.NCMEC20170308 out=sesug.RandomSample02
2     method=srs reps=1000 sampsize=10 seed=1 noprint;
3     id age;
4 run;

```

Figure 23. Edited code from SRS01 node. Circled icon allows the code to be saved as a snippet.

Note the editable code has been opened in its own tab which SAS has named **Program 1**. To request multiple samples simply edit the code to include the option REPS=1000 as shown above. (This procedure option can be placed before or after any of the other options but individual programmers tend to have a preferred order for options.) To ensure the data set from the SRS01 node isn't overwritten make sure to take note of the updated data set name in the code (RandomSample02). As users edit the code they should notice autocomplete menus appearing that can provide guidance on what syntax is appropriate and the purpose of each option. (Users who wish to disable autocomplete may do so via the **Application Options** icon at the top of the SAS Studio session.)

Add to My Snippets

Name:

Location:

Figure 24. Dialog box for saving a custom snippet.

Once the code has been edited users need to save it as a snippet. SAS has provided an icon specifically for this purpose – indicated by the red circle in Figure 23. Clicking this icon brings up a dialog box, shown in Figure 24, where users can name their snippet and see where it will be located. For the purposes of this activity the snippet used here is named SRS1000. After saving, the snippet is opened in an additional window and the previous window is left unchanged (and unsaved). The SRS1000 window and Program 1 windows can now be closed; this returns the user to the SRS01 code that was used to generate the snippet. (Closing **Program 1** will prompt a confirmation box since that file is unsaved. Don't worry – this is an exact copy of the code that got saved in the snippet.)

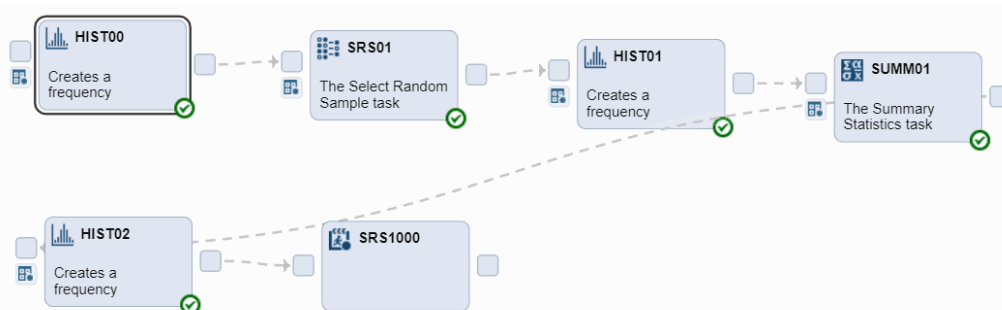


Figure 25. Updated process flow after dragging over (and connecting) the SRS1000 snippet.

Returning to the Process Flow users can now navigate to **Snippets** → **My Snippets** → **SRS1000** and drag that to the workspace to create a new node and then connect it to the HIST02 node. The icon for snippets is slightly different than for tasks. There are still control ports so users can dictate in what order the snippet should be included in the Process Flow but there is no data icon nor is there the red circle and triangle. Running the SRS1000 node does produce a green check and circle indicating that the results were successfully generated. Recall nodes can be submitted by right-clicking the node and selecting **Run** or by double-clicking and using the running person icon.

Columns		Total rows: 10000 Total columns: 2	
<input checked="" type="checkbox"/> Select all <input checked="" type="checkbox"/> Replicate <input checked="" type="checkbox"/> age		Replicate	age
1		1	14.82
2		1	17.00
3		1	15.89
4		1	17.19
5		1	12.84
6		1	15.89
7		1	3.97
8		1	15.06
9		1	14.95
10		1	13.80
11		2	6.74
12		2	17.60

Figure 26. First 12 of the 10,000 generated records. Each set of 10 has a unique value of replicate.

After running the **Log** tab shows nothing of concern and the **Output Data** tab (Figure 26) shows that many samples have now been generated, each with a unique value of the replicate variable that has been included in the data set (RandomSample02). Again, users can verify their data set matches that given here but when carrying out the activity ensure the SEED= option is not used unless the intent is for all students to have the exact same realization of the sampling distribution.

TASKS: PART 2

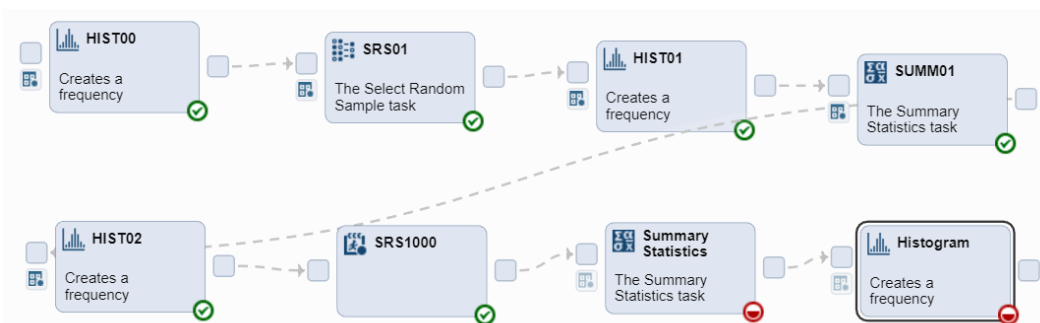


Figure 27. Summary statistics and histogram nodes added to complete the process flow.

Now that multiple random samples have been selected new summary statistics can be calculated and a new histogram can be constructed that gives better insight into the sampling distribution. First drag another instance of the summary statistics task into the workspace and connect it to the SRS1000 node. Next, add an additional histogram task to create the final node. Ensure it is connected to the newly added summary statistics node. Figure 27 shows the inclusion and connection of the additional nodes.

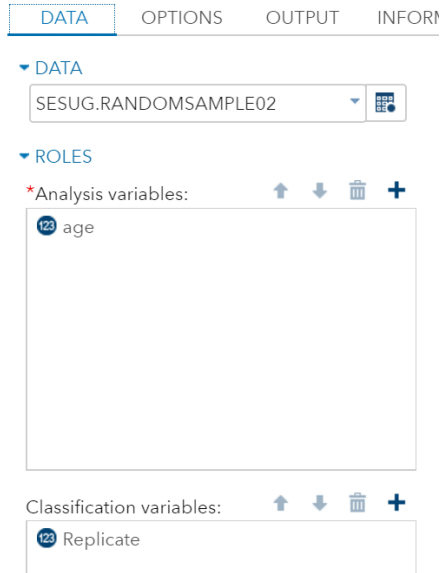


Figure 28. Data tab for the newly added summary statistics task (SUMM02).

Prepare the new summary statistics node (SUMM02) by ensuring it uses the RandomSample02 data set from the SESUG library and uses age as the analysis variable. Unlike the SUMM01 node this node, SUMM02, needs a classification variable to ensure the sample mean is calculated separately for each replicate. Figure 28 also shows use of the **Classification Variable** section of the **Roles** setting add replicate via the plus icon. As before use the **Output** tab to create a data set (STATZ02) to hold these summary statistics. After naming the node via the **Node** tab submit the code for execution.

Analysis Variable : age		
Sample Replicate Number	N Obs	Mean
1	10	14.1388090
2	10	15.3322382
3	10	15.2301164
4	10	13.1101985
5	10	12.5051335
6	10	12.4046543
7	10	12.6787132
8	10	15.7637235
9	10	14.6666667
10	10	12.4210815

Figure 29. RESULTS tab for the second Summary Statistics node (SUMM02).

While the **Log** tab should look very similar to that generated by SUMM01 the **Results** and **Output Data** tabs are noticeably different. Rather than a single mean, the sample mean for each of the 1000 random samples is shown as a table (shown in Figure 29) and as a data set. Notice that as expected the first mean, 14.14, matches the one generated by the SUMM01 node since the SEED= option was used to select a specific seed. Now that the data set contains sample means from multiple samples the final histogram can be constructed.

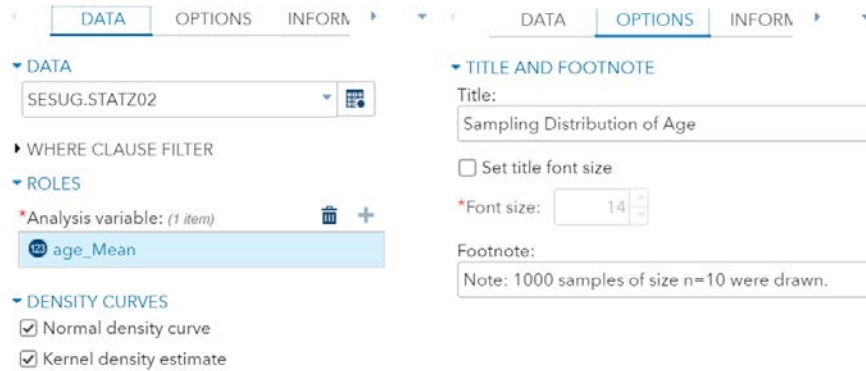


Figure 30. Data and Options tabs for final histogram (HIST03).

Return to the workspace and double-click the final histogram to edit it. This node is very similar to HIST02 in that it uses STATZ02 for data and age_Mean for the response variable. However ensure the normal and kernel densities are added now that more than one sample has been taken (Figure 30, left). The **Options** tab can be used to update the title, footnote, and vertical scaling (Figure 30, right).

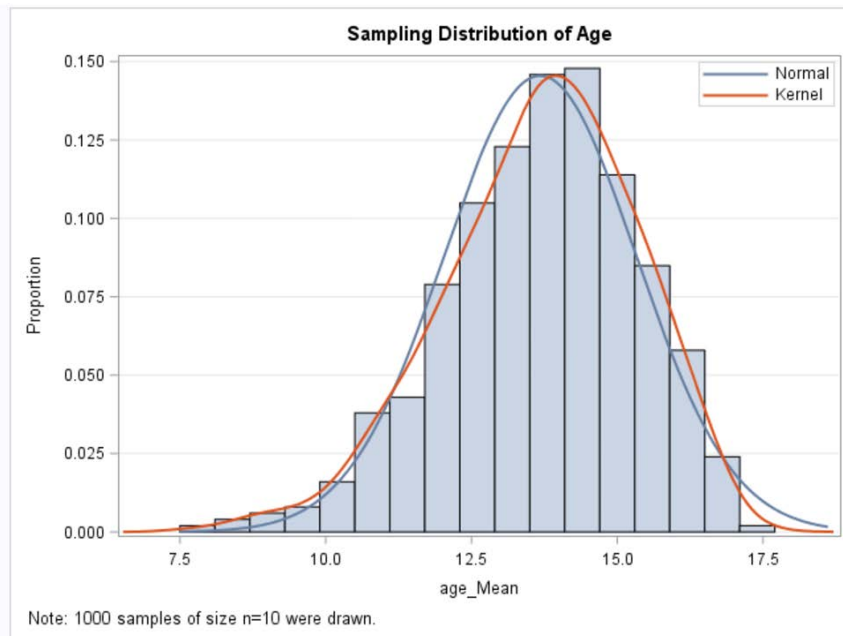


Figure 31. Sampling distribution of 1000 samples of size 10 including normal and kernel density estimates.

Running this node produces a trouble-free log and displays the histogram shown in Figure 31. The histogram is still somewhat skewed due to the skewness of the population and the small sample size. This can be seen either by the poor fit of the normal density to the histogram or by the difference between the kernel and normal density estimates.

At this stage the process flow, as shown in Figure 25, would only require minimal edits to investigate the sampling distribution of a single variable's mean in other scenarios. Editing the data set and variables used in the tasks, and editing the code for the SRS1000 custom snippet, provides a quick investigative tool. However, further customizations would allow much greater flexibility in how it could be implemented.

CUSTOMIZING THE CODE

At this point a discussion with students about the assumptions for inference, and what to do when they're not met, might be appropriate. To motivate the remainder of the activity one question that hasn't been

answered is: for this population, how large a sample size is needed to feel the assumption of a normal sampling distribution has been met? To answer this question using the process Flow created during the activity students would need to manually edit each of the nodes that used $n=10$ and update to a new sample size. This would certainly require updates to the SRS01 and SRS1000 nodes since they generated the samples. However, since each of HIST01, HIST02, and HIST03 included footnotes with the sample size they would need to be updated as well. Clicking back through each node, editing the options, and rerunning is somewhat monotonous even on such a small process flow. Furthermore, recall SRS1000 was a snippet and not a task meaning the code needs to be edited and saved, either as a new snippet or by overwriting the SRS1000 snippet. If students (or the teacher) wish to examine more than the effect of one or two sample sizes on the sampling distribution this becomes frustratingly tedious.

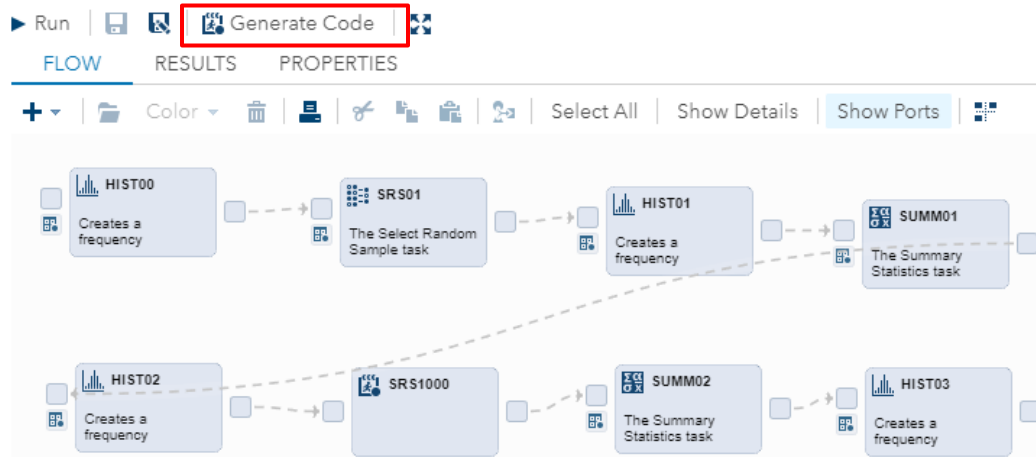


Figure 32. Using the process flow to generate all the code.

However, as with the creation of the SRS1000 snippet based on the SRS01 node which only required small edits to the code from one task, this update can be done with small changes to the code for all tasks. Unlike editing each individual node the edits needed now should be applied to all nodes. To facilitate this first use the menu item named **Generate Code** just above the **Properties** tab (outlined in red in Figure 32). This button creates a single program (named **Program 1** by default) that contains the code for all nodes in the process flow. Generating the proper program requires that the nodes have been sequenced properly in the process flow though; otherwise the program won't execute as intended.

There are two basic edits that allow students a quick and easy way to generate various sampling distributions.

1. Edit the summary statistics nodes to streamline the output
2. Edit the sample size used to allow quick customizations

SUMMARY STATISTICS

The procedure SAS Studio uses in the summary statistics task, PROC MEANS, generates output in the results window by default. As shown in Figure 29 this may useful for performing quick checks on the results during the developing of the program but isn't that useful for the students once the code has been finalized. Furthermore, as more code is developed and submitted the amount of output can become distracting for students.

```
proc means data=SESUG.RANDOMSAMPLE01 chartype mean vardef=df noprint;
var age;
output out=SESUG.STATZ01 mean= / autoname;
run;
```

Figure 33. Add the NOPRINT option to the keep PROC MEANS from printing the results.

To suppress this the NOPRINT option can be added to each of the PROC MEANS steps as shown in Figure 33. (It is only shown for SUMM01 but the same edit should be applied to the code generated by SUMM02). The effect of this change is most evident in that it prevents the table of 1000 sample means from being printed by the SRS1000 snippet. (Note: the NOPRINT option is not the only way to suppress the output, but it is the method that requires the least code editing given the basic code provided by the summary statistics task.)

CHANGING SAMPLE SIZES

The SAS macro language is powerful in that it allows for easy text substitution within a program. While even a cursory discussion of the macro language is outside the scope of this paper, creating and using a macro variable is simple in practice.

```
/*Macro variable definition*/  
%let n = 20;
```

Figure 34. Demonstration of the %LET statement to define a sample size.

To make this edit return to the top of the program and include the %LET statement shown in Figure 34. This creates a constant known as a macro variable which is now available for use in any future line of code. Using the variable requires only a slight change to each location where the previous sample size (10) was mentioned.

```
proc surveyselect data=SESUG.NCMEC20170308 out=sesug.RandomSample01 method=srs  
    sampsize=&n seed=1 noprint;  
    id age;  
run;
```

Figure 35. Changing the local sample size definition, 10, to the global definition, 20.

The first encounter of the sample size being defined is in the SRS01 node where the SAMPSIZE=10 option is used. To edit this code users simply need to replace the 10 with &n as shown in Figure 35. The ampersand instructs SAS to access a macro variable named n. SAS will then look up the current value, 20, and use that when constructing the SRS. It is important that the update occur throughout the program so ensure it is changed in the following locations: HIST01, HIST02, SRS1000, and HIST03. Figure 36 shows the change for HIST01 as an additional example.

```
/*--TITLE and FOOTNOTE--*/  
title "Distribution of Sampled Age";  
footnote2 j=1 "Note: n=&n cases were used in the sample.";
```

Figure 36. Including the macro variable, n, in the footnote of a histogram.

After the updated program is submitted (F3 or running person icon to submit the whole program) the results will be produced for a sample size of 20 instead of 10. Once the update is in place, rerunning for a different sample size is as simple as changing the value in the %LET statement at the top of the program. Figure 37 shows the final histogram of the sampling distribution for 1000 draws of size 20.

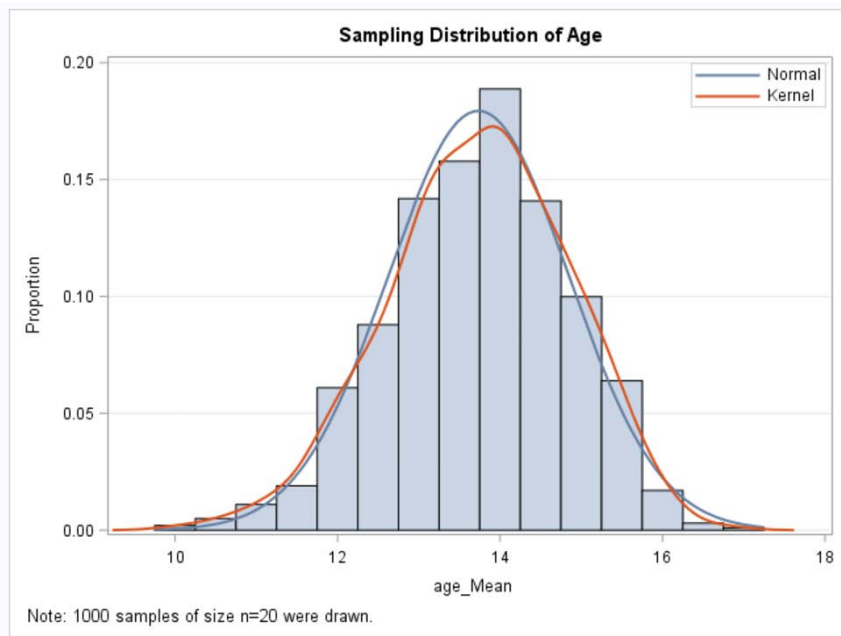


Figure 37. Sampling distribution of the mean age when 1000 draws of size 20 are used.

SUMMARY

The flexibility offered by SAS Studio allows for the expansion of one or more aspects of this activity. For example, students could be asked to investigate the Central Limit Theorem by creating a sequence of sampling distributions for a single population using several different sample sizes. For further investigations students could use the same set of sample sizes on multiple populations to see that while $n > 30$ is often sufficient it is not always enough to guarantee approximate normality of the sampling distribution of the sample mean. The activity presented here and both of these expansions could be applied to the sampling distribution of statistics other than the sample mean as well. Since the summary statistics task allows for the computation of many different statistics, the sampling distributions could be compared. Of particular interest might be showing the sampling distributions of the standard deviation and/or variance which are skewed versus those for the mean and median which are symmetric.

Another extension, particularly of interest for students who may go on to take computer science or major in STEM fields that require programming, is to develop further customizations by editing the code manually. Looping through different numbers of samples, sample sizes within each sample, and graphics customizations are just a few of the possibilities that would give students additional exposure to programming in general, but specifically programming in an industry-standard language such as SAS. The ability to build scaffolding into the activities as the semester progresses, by beginning with all menu-driven nodes then working in more user-programmed snippets, makes SAS Studio a natural tool for the statistics classroom. Leveraging the power of SAS Studio by incorporating activities into the classroom that reinforce all levels of the GAISE recommendations is made easier for every educator.

REFERENCES

Cody, Ron. 2015. *An Introduction to SAS University Edition*. Cary, NC: SAS Institute Inc.

SAS Institute Inc. *SAS Studio 3.6: User's Guide*. Available at <http://documentation.sas.com/?cdcid=webeditorcdc&cdcVersion=3.6&docsetId=webeditorug&docsetTarget=titlepage.htm&locale=en>. Last accessed on 2017-09-17.

Cloudera Child Finder Hackathon. Available at <https://childfinder.hackerearth.com/>. Last accessed on 2017-08-27.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jonathan W. Duggins
North Carolina State University
jwduggin@ncsu.edu
<http://www4.stat.ncsu.edu/~duggins/>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.