

The Simple Guide to SAS[®] From Null to Novice

A large, colorful geometric pattern composed of many small, tilted rectangular blocks in various colors (including red, blue, green, yellow, orange, and grey) arranged in a grid-like structure that recedes into the distance, creating a sense of depth and perspective.

Kirby Thomas

The correct bibliographic citation for this manual is as follows: Thomas, Kirby. 2024. *The Simple Guide to SAS®: From Null to Novice*. Cary, NC: SAS Institute Inc.

The Simple Guide to SAS®: From Null to Novice

Copyright © 2024, SAS Institute Inc., Cary, NC, USA

ISBN 978-1-68580-009-3 (Paperback)

ISBN 978-1-68580-010-9 (Web PDF)

ISBN 978-1-68580-011-6 (EPUB)

ISBN 978-1-68580-012-3 (Kindle)

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

April 2024

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.

SAS software may be provided with certain third-party software, including but not limited to open-source software, which is licensed under its applicable third-party software license agreement. For license information about third-party software distributed with SAS software, refer to <http://support.sas.com/thirdpartylicenses>.

Contents

About The Author.....	v
Acknowledgments.....	vii
Part One: Getting Started with SAS	1
1 Introduction.....	3
1.1 Note of Encouragement.....	3
1.2 Purpose of this Guide.....	3
2 Getting Started	5
2.1 SAS Windows	6
2.2 SAS® OnDemand for Academics.....	9
2.3 SAS® Studio	11
2.3.1 Tasks	13
2.3.2 Snippets.....	16
2.4 Data Setup.....	17
3 How SAS® Works.....	19
3.1 DATA and PROC Steps.....	19
3.1.1 DATA Step	20
3.1.2 PROC Step.....	20
3.2 Data Structure	21
3.2.1 Columns/Variables.....	21
3.2.2 Values	22
3.2.3 PROC Contents.....	22
3.3 DATA Step Processing	24
3.4 OUTPUT Statements	26
3.5 SAS Libraries.....	29
3.5.1 Macro Libraries.....	31
3.5.2 Read-Only Access Libraries	32
4 Import/Export.....	35
4.1 Importing Data	35
4.2 Exporting Data	39
5 Viewing and Summarizing Data.....	41
5.1 Viewing Data	41
5.2 Summarizing Data	44

Part Two: Coding with SAS	51
6 Data Transformations	53
6.1 Sorting and De-duplicating Data	53
6.2 Calculating New Variables	55
6.3 Filtering	56
6.4 Conditional Logic.....	57
6.5 Manipulating Values.....	60
6.5.1 Character Functions.....	61
6.5.2 Numeric Functions.....	62
6.5.3 Date Functions.....	62
6.5.4 Manipulating Values Example.....	63
6.6 Formatting.....	65
6.6.1 User-Defined Formats.....	66
6.6.2 Storing User-Defined Formats	68
7 Combining and Aggregating Data	71
7.1 Combing Data Using the DATA Step	71
7.1.1 Appending Data	71
7.1.2 One-to-One Merge	73
7.1.3 One-to-Many Merge.....	74
7.2 PROC SQL	76
7.2.1 Types of Joins.....	76
7.2.2 PROC SQL Syntax	79
8 Comparing Data	87
8.1 DATA Step Compare Merge	87
8.2 PROC COMPARE	88
9 Reporting	89
9.1 PROC PRINT	89
9.2 PROC REPORT	91
9.3 Output Delivery System	96
9.3.1 ODS Destinations	97
9.3.2 ODS Graphics	97
10 Introduction to Selected Advanced Topics	105
10.1 Cloud Analytic Services (CAS).....	105
10.2 Macro Language.....	106
10.2.1 Macro Variables.....	106
10.2.2 Macro Programs	108
Appendix A: Resources	109
Appendix B: Create GSS SAS Data Set	110

About The Author



Kirby Thomas is a Technical Communicator at SAS who enjoys taking complex concepts and breaking them down into easy-to-understand steps. She has a PhD in Sociology from Florida State University, where her research focused on why women are less likely than men to major and persist in science, technology, engineering, and mathematics (STEM) fields. Her passion for creating more accessible and equitable learning environments was cultivated in her job at the Florida Department of Education and continues to motivate her work every day. She has over 9 years of experience programming in SAS and is excited to share her lessons learned with new programmers so that they can feel empowered to dive into coding without the hindrance of perfection paralysis.

Learn more about this author by visiting her author page at <https://support.sas.com/en/books/authors/kirby-thomas.html>. There you can download free book excerpts, access example code and data, read the latest reviews, get updates, and more.

2 Getting Started

The field of data analytics is fast-paced, ever-evolving, and often intimidating to newcomers, but learning how to leverage data analytics is a critical skill that can enrich your life in so many ways. Not only can it dramatically increase your marketability and financial security, as many companies compete to hire and retain talent in these positions, but your work can lead to real policy changes that can positively impact the world. For example, data analytics has been used to improve water quality, aid in natural disaster response, protect endangered species, fight human trafficking, detect fraud, track the spread of disease, and combat homelessness and food insecurity. To learn more about how analytics can help humanity, visit https://www.sas.com/en_us/data-for-good.html.

Now that we have discussed some of the benefits of data analytics, let's talk about why SAS is a great tool to help you on your learning journey. To stay competitive in the field of analytics, SAS has released a slew of products over the years to assist users with their data management, forecasting, modeling, machine learning, text analytics, and visual analytics needs, to name a few. SAS® Viya® was specifically designed as a one-stop shop software platform that allows users, businesses, and organizations to access, manage, analyze, and visualize their data all in one place. The SAS Viya® platform houses several applications like SAS® Drive to share and collaborate between users, SAS® Data Studio to prepare data, SAS® Studio to develop SAS code, SAS® Model Studio to build models, and SAS® Visual Analytics to explore and visualize data.

With all of the products SAS has to offer, it is hard to know where to start. And in my experience, starting is always the most challenging part of any new venture. Rather than trying to teach you everything there is to know about SAS, which would be overwhelming and, quite frankly, outside the scope of my knowledge, my goal in creating this guide is to create a roadmap for new coders on what topics to start with to quickly learn the basics of programming in SAS. To really understand SAS and build a strong foundation, the best place to start, in my opinion, is learning Base SAS®.

Base SAS® is a programming language at the heart of all SAS Software, making it an essential step in learning SAS. Base SAS® is updated with new functionality every so often, resulting in new versions. I am using SAS 9.4 (maintenance release TS1M7) for all examples in this guide, so minor modifications might be required if you are working in a different version. SAS® Studio (web-based), SAS® Enterprise Guide® (Windows client application), and the SAS Windowing Environment (locally installed) are all Graphical User Interfaces (GUIs) that can be used to develop and run Base SAS® code. Most of the code discussed in this guide will work on any of these interfaces, and although each GUI operates a little differently, Base SAS® is the common language used for all of them.

2.1 SAS Windows

There are four universal components to know about when getting started, no matter which SAS programming GUI you use: the Editor window, Log window, Output Data window, and Results window. They usually appear as tabs (Code, Log, Output Data, and Results) under each SAS program, and you can customize the layout of these windows in many ways. Try out different layout views and find what works best for you.

1. The **Editor window** (Code tab) in [Figure 1](#) is where you do most of your coding. This is where you enter and save the syntax— or properly structured code— for your program. There are line numbers on the left-hand side of the window for easy reference. You can run entire programs or highlight a few lines of code within a program and only run the portion of code that you selected. This is helpful when you are building a new program or troubleshooting a bug in your code. You might want to run one step at a time and view your results. Be sure to *save your programs often* as you work so that you don't lose work if your computer dies, restarts unexpectedly, you time out, or get kicked off of your online session. Every data analyst has at least one horror story about losing several hours of code work because they didn't save — or at least that's what I tell myself to sleep better at night.

Figure 1: Editor Window

The screenshot shows the SAS Editor window for a file named *Book Examples.sas. The window has three tabs: CODE, LOG, and RESULTS. The CODE tab is active and highlighted with a red box. Below the tabs is a toolbar with various icons for file operations and execution. The main area contains SAS code with line numbers on the left. The code is as follows:

```

1 Libname input "/home/u1093828/Data";
2
3 data donations;
4   set input.Donations_JUL;
5   Total_Spent=Donation + Merchandise;
6 run;
7
8 proc print data=donations; run;
9
10 proc means data=input.student_demographic;
11   var age;
12 run;
13
14

```

2. The **Log window**, displayed in [Figure 2](#), provides information about how your code ran. It lists notes during program execution, warnings, and error messages if encountered. If there is an error in your code, a message will appear in red in the

log under the statement that caused the error. Notes appear in blue, and warning messages in green. Use the log messages to help you troubleshoot any issues you encounter when executing your code. **Always check the log** after running any code to ensure it ran the way you expected it to. The log also provides additional useful information like the number of variables and observations in each data set and how long it took to process each DATA step. This can be invaluable information for you if you need to know which part of the code was taking the longest to run. There are statements and options that allow users to customize their logs by adjusting the appearance of the log, suppressing certain contents of the log, or writing additional information to the log. More information about these statements and options can be found at: https://go.documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/lepg/p119kau8rt2ebgn1bzaipafu6jp3.htm.

Figure 2: Log Window

The screenshot shows the SAS Log Window for a session named '*Book Examples.sas'. The 'LOG' tab is selected and highlighted with a red box. The window displays the following content:

```

CODE LOG RESULTS OUTPUT DATA
Errors, Warnings, Notes
  Errors
  Warnings
  Notes (8)

1      OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
68
69      Libname input "/home/u1093828/Data";
NOTE: Libref INPUT was successfully assigned as follows:
      Engine:          V9
      Physical Name:  /home/u1093828/Data
70
71      data donations;
72      set input.Donations_JUL;
73      Total_Spent=Donation + Merchandise;
74      run;

NOTE: There were 3 observations read from the data set INPUT.DONATIONS_JUL.
NOTE: The data set WORK.DONATIONS has 3 observations and 8 variables.
NOTE: DATA statement used (Total process time):
      real time          0.00 seconds
      user cpu time      0.00 seconds
      system cpu time    0.00 seconds
      memory             950.03k
      OS Memory         21160.00k
      Timestamp          09/19/2023 08:02:06 PM
      Step Count         35  Switch Count  2
      Page Faults        1
      Page Reclaims     203
      Page Swaps         0
      Voluntary Context Switches 24
      Involuntary Context Switches 0
      Block Input Operations 480
      Block Output Operations 264

```

```

75
76      proc print data=donations; run;

NOTE: There were 3 observations read from the data set WORK.DONATIONS.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.01 seconds
      user cpu time       0.01 seconds
      system cpu time     0.00 seconds
      memory              1246.59k
      OS Memory           21156.00k
      Timestamp           09/19/2023 08:02:06 PM
      Step Count          36   Switch Count  0
      Page Faults         0
      Page Reclaims       180
      Page Swaps           0
      Voluntary Context Switches 0
      Involuntary Context Switches 0
      Block Input Operations 0

```

3. The **Output Data window** shows your output data tables (see [Figure 3](#)). It is similar to an Excel spreadsheet, as it shows the rows and columns of your data table. I often spot-check my code using the Output Data window. For example, if I calculate a new variable, I make sure that variable shows up in my output table after I run my code and that the first few rows have the appropriate calculated values in the new variable column. Remember, SAS will not always give you an error if you make a mistake. For example, suppose you type the wrong variable name in a calculation. SAS might not give you an error in the log because the wrong variable does exist in your data. However, the calculation itself is wrong because you meant to use a different variable. The most important lesson about programming in SAS (and programming in general) is that programs run exactly what you type/tell them to run, *not what you meant to run*. So always spot-check your output for accuracy.

Figure 3: Output Data Window

The screenshot shows the SAS Output Data window for a table named WORK.DONATIONS. The window displays a table with 3 rows and 8 columns. The columns are: First_Name, Last_Name, Address, DOB, Donation, Donation_Date, Merchandise, and Total_Spent. The rows contain data for three individuals: John Brown, Kelsey Green, and Sherlock Holmes. The table is displayed in a grid format with a column list on the left and a table view on the right.

First_Name	Last_Name	Address	DOB	Donation	Donation_Date	Merchandise	Total_Spent
1 John	Brown	6523 E Cherry St	05/22/1957	\$500	07/12/2022	\$122.68	622.68
2 Kelsey	Green	3111 Orange Ave	03/16/1989	\$25	07/13/2022	\$57.23	82.23
3 Sherlock	Holmes	221B Baker Street	01/06/1977	\$221	07/08/2022	\$0.00	221

4. Finally, the **Results window** in [Figure 4](#) displays any written output such as descriptions, calculations from SAS procedures, or sample data. Throughout this guide, the word

print is used to indicate outputting information to the results window. For example, suppose you would like to see a data table description. You can run PROC CONTENTS on this table, and the results of this procedure are printed to the Results window for you to view. If you want to know the mean value of one of the variables in your data set, you can run PROC MEANS on the variable and view the results in this window. It takes SAS time to print results for you, so avoid printing large data sets (PROC PRINT) or frequencies (PROC FREQ) for variables with over 100 value categories. It will take forever to run or warn you that you are trying to print out too much information to the results window. If you need to look at a large data set, it is easier to view it in your Output Data window or to export the data to Excel.

Figure 4: Results Window

The screenshot shows the SAS Results Window for a file named *Book Examples.sas. The window has tabs for CODE, LOG, RESULTS (which is selected and highlighted with a red box), and OUTPUT DATA. Below the tabs is a toolbar with icons for home, back, forward, print, save, refresh, and help. A 'Table of Contents' section is visible, followed by two data tables.

Obs	First_Name	Last_Name	Address	DOB	Donation	Donation_Date	Merchandise	Total_Spent
1	John	Brown	6523 E Cherry St	05/22/1957	\$500	07/12/2022	\$122.68	622.68
2	Kelsey	Green	3111 Orange Ave	03/16/1989	\$25	07/13/2022	\$57.23	82.23
3	Sherlock	Holmes	221B Baker Street	01/06/1977	\$221	07/08/2022	\$0.00	221.00

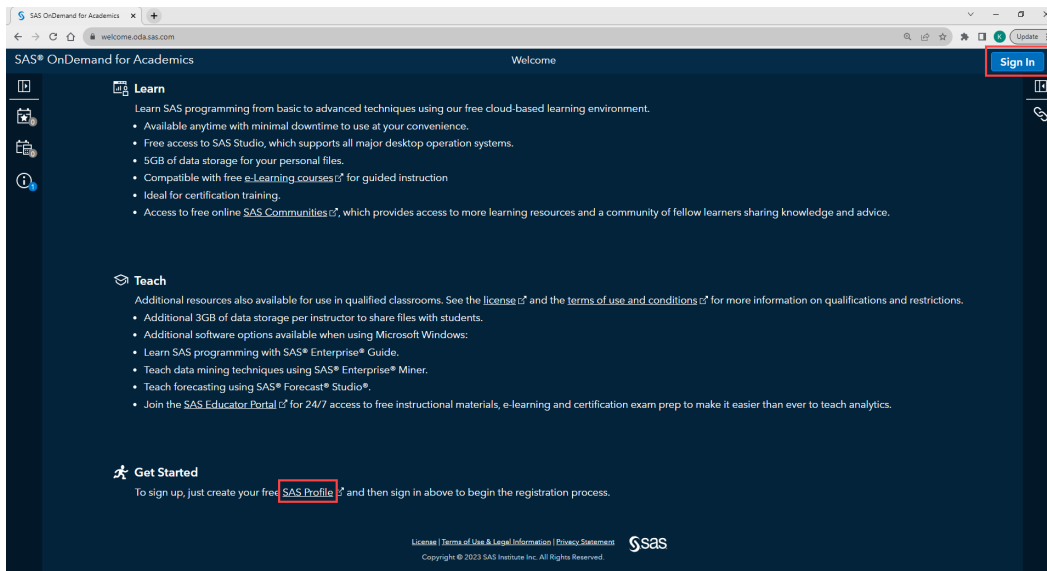
The MEANS Procedure				
Analysis Variable : Age Age				
N	Mean	Std Dev	Minimum	Maximum
4	20.2500000	0.9574271	19.0000000	21.0000000

2.2 SAS® OnDemand for Academics

Now that we've covered the main components of Base SAS® GUIs, let's walk through one of these interfaces together. Since SAS® Studio is the newest of these GUIs, is offered through SAS Viya®, and is available for free on the SAS® OnDemand for Academics website, I will give a brief overview of its layout and features before diving into Base SAS® syntax.

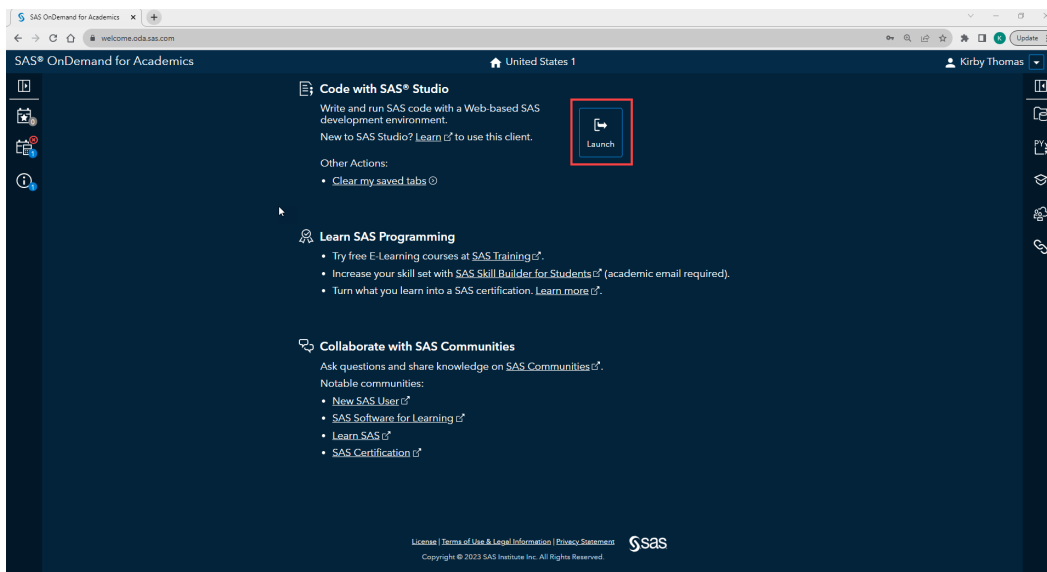
To start using SAS® Studio for free, visit: <https://welcome.oda.sas.com/>. If you have not done so already, under Get Started, click **SAS Profile** to create an account ([Figure 5](#)). Fill out the required fields and click **Create Profile**.

Figure 5: SAS® OnDemand for Academics Welcome Page



Once you have a registered account, from the Welcome page, click **Sign In** in the upper right hand corner of the screen (Figure 5) and type in your login information. Read and then check the box next to Accept the terms of the license and the terms of use and conditions. Then, click **Sign In**. Click the **Launch** button next to Code with SAS® Studio (Figure 6).

Figure 6: Launch SAS® Studio

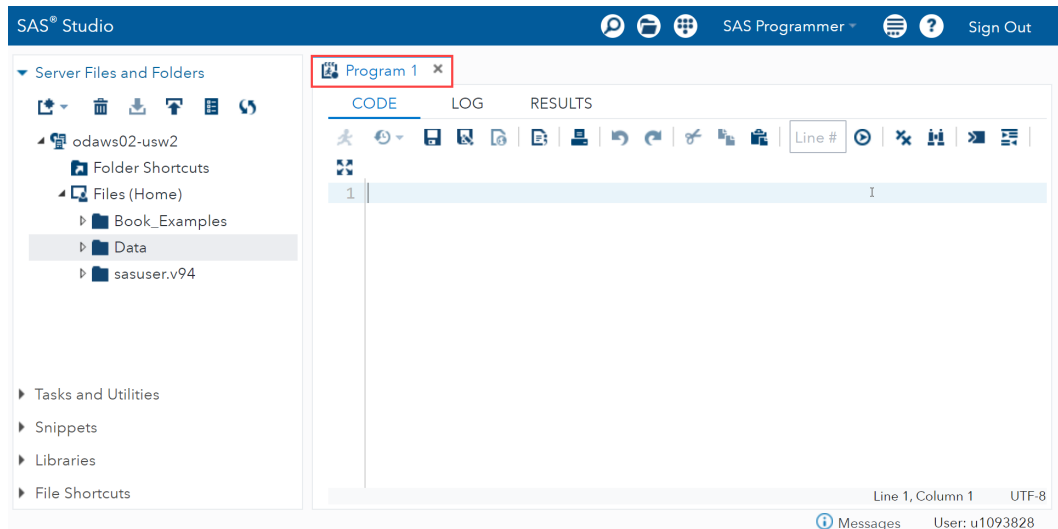


SAS® Studio opens in a new tab in your browser.

2.3 SAS® Studio

SAS® Studio is a web-based programming interface that connects to a local or hosted SAS server. When you log on, a new session initiates, and a navigation pane appears on the left side of the screen while a work area appears on the right side. You might see a Start Page in the work area where you can select **New SAS Program**, or SAS® Studio might have already opened a new program for you. By default, a new Program tab reads Program 1 until you save it ([Figure 7](#)).

Figure 7: Screenshot of a New SAS Program in SAS® Studio



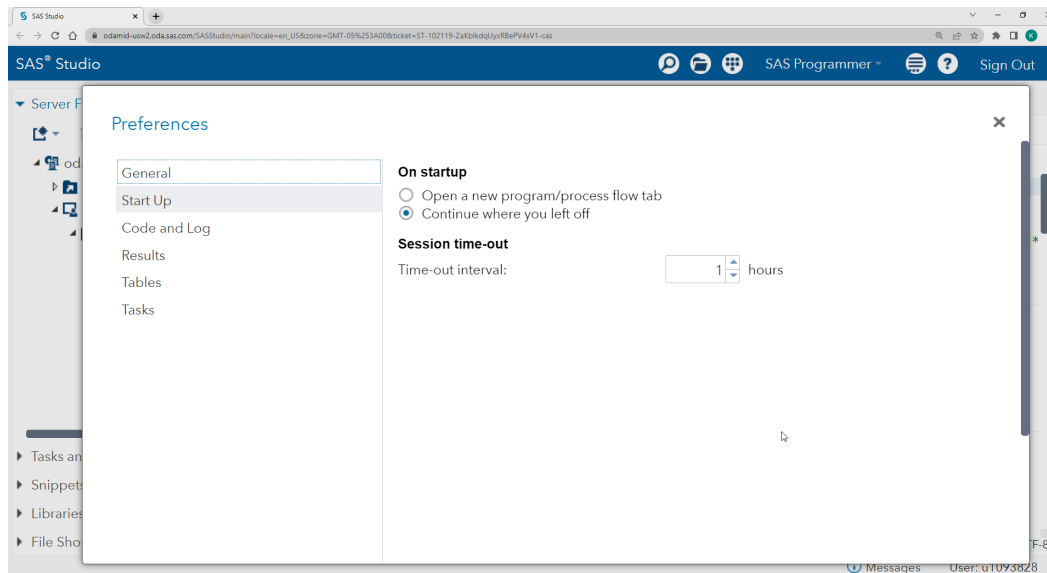
Once saved, the Program tab will display the program name that you chose. SAS programs are saved with the .sas file extension and SAS tables are saved with the .sas7bdat file extension. The Program tab has subtabs that are discussed in detail in the [SAS Windows](#) section. At a high level, the Code subtab is where you write your SAS code, the Log displays information and errors when you run your code, the Output Data subtab displays the SAS tables that you are working with, and the Results subtab provides the results of data analysis and anything you tell SAS to print out for you. When you save your program, it saves the syntax in the Code subtab so that you can open and rerun the code later.

Note: The Output Data subtab does not appear until you read in or create a data table.

If you want SAS® Studio to automatically open the program(s) that you were working on and saved but did not close before exiting your previous SAS® Studio session (I highly recommend this), click the **hamburger icon** (three stacked lines) in the upper-right hand corner of the screen

called More Application Options, then **Preferences > Start Up > Continue where you left off**¹ ([Figure 8](#)).

Figure 8: SAS Studio Preferences



The Navigation Pane is typically located on the left-hand side of the screen ([Figure 7](#)). It has five sections: Server Files and Folders, Tasks and Utilities, Snippets, Libraries, and File Shortcuts.

1. The **Server Files and Folders** section (labeled Explorer in SAS Viya®) provides access to your computer and server folders that contain files, programs, and data tables.
2. The **Tasks and Utilities** section provides a point-and-click feature that allows users to explore, visualize, and analyze data.
3. The **Snippets** section lets users save syntax that they use frequently so that they can easily paste and edit it in new programs.
4. The **Libraries** section displays the assigned SAS libraries for your session and the data tables contained in each library. You will learn more about assigning SAS libraries in the [SAS Libraries](#) section of the guide.
5. The **File Shortcuts** section enables you to create shortcuts to files on the SAS server or found at a URL and quickly access them by double-clicking or dragging the shortcut to the work area.

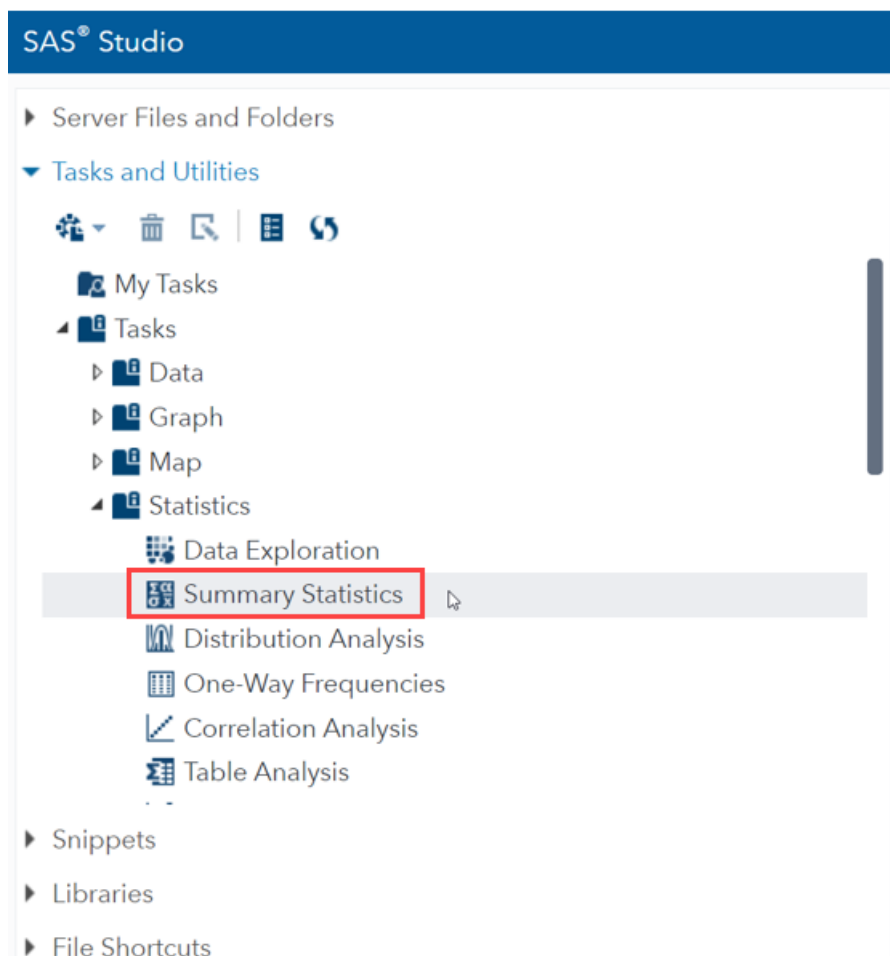
¹ In SAS® Studio V in SAS Viya®, check the bubble next to **Options > Preferences > Start Up > Restore tabs listed in Open Files pane**.

2.3.1 Tasks

While this guide focuses on writing your own code or syntax, Tasks can be a useful tool when you are unsure of the syntax needed for a procedure. You can use this point-and-click feature in SAS® Studio or SAS® Enterprise Guide® to generate the code that you need, copy the syntax, and save it to your program for future use. This syntax can then be run on any SAS GUI.

Let's walk through an example. If you want to know summary statistics about a data set but are unsure of the syntax needed to create it, click on the **Tasks** section, expand **Tasks** and then **Statistics** and double-click on **Summary Statistics** (Figure 9). A new tab called Summary Statistics opens that lets you browse and select a data table, select variables of interest, and run the generated code.

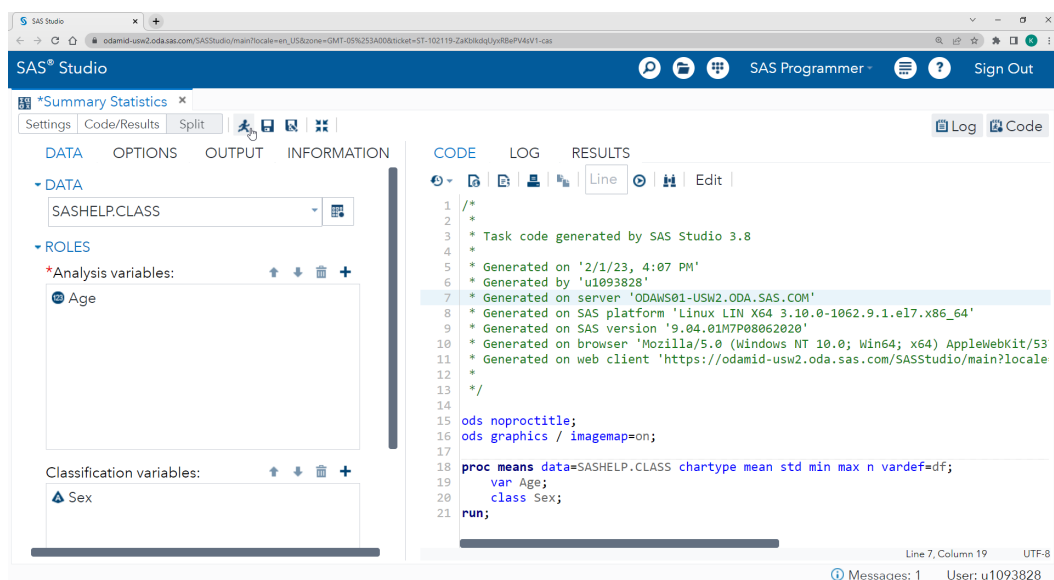
Figure 9: SAS Studio Tasks in Navigation Pane



Note: For those just getting started, SAS provides several preloaded data sets to practice with. When browsing for a data table, go to the **Libraries** section in the navigation pane, expand **My Libraries** and **SASHELP**.

In the Summary Statistics tab and Data subtab, click the **data set icon** (select a table) under Data, expand **My Libraries** and **SASHELP**, and double-click **CLASS** to load the Class table. Under Roles, click the **plus sign** over the Analysis variables box, click on **Age** and then **OK**. Next, click the **plus sign** over the Classification variables box, click on **Sex** and then **OK**. There is also an Options subtab that lets you choose which summary statistics you would like included in your output. Keep the default selections checked for now. Notice that once the required roles are filled in on the Data tab, code is generated on the right-hand side of the screen in the Code tab ([Figure 10](#)). When you click **Run** (the little running person next to the save button) or **F3** on your keyboard, the code executes, and the results are shown in the Results subtab.

Figure 10: Summary Statistics Task Generated Code



Note: The green text in the Code subtab displays comments that are not part of the SAS syntax itself but that provide helpful information about when and where the code was generated. SAS ignores comments during code execution. `/*` denotes the beginning of a comment, while `*/` denotes the end of the comment². It is always good practice to include

² Pro tip: If you highlight a block of text in your code editor and then hold **Ctrl** and click **/**, SAS will comment out the entire block of text you have selected. Use this keystroke combination again to uncomment a highlighted block of code. In the SAS® Enterprise Guide®, click **Ctrl, Shift, /** to uncomment.

comments in your programs to help remind yourself or other coders that might inherit your programs of why certain decisions were made or what certain code blocks are doing.

We can see from the log that no errors occurred when running the program ([Figure 11](#)).

Figure 11: Summary Statistics Task Log

```

CODE    LOG    RESULTS
-----
[Icons]
▼ Errors, Warnings, Notes
  ▸ Errors
  ▸ Warnings
  ▸ Notes (3)

1      OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
NOTE: ODS statements in the SAS Studio environment may disable some output features.
69
70      /*
71      *
72      * Task code generated by SAS Studio 3.8
73      *
74      * Generated on '9/18/23, 5:10 PM'
75      * Generated by 'u1093828'
76      * Generated on server 'ODAMS01-USW2.ODA.SAS.COM'
77      * Generated on SAS platform 'Linux LIN X64 3.10.0-1062.9.1.el7.x86_64'
78      * Generated on SAS version '9.04.01M7P08062020'
79      * Generated on browser 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
79      ! Chrome/116.0.0.0 Safari/537.36'
80      * Generated on web client
80      ! 'https://odamid-usw2.oda.sas.com/SASStudio/main?locale=en_US&zone=GMT-04%253A00&ticket=ST-157946-0cha4ejteSF9QJzkrT5q-cas
80      ! '
81      *
82      */
83
84      ods noproctitle;
85      ods graphics / imagemap=on;
86
87      proc means data=SASHELP.CLASS chartype mean std min max n vardef=df;
88      var Age;
89      class Sex;
90      run;

NOTE: There were 19 observations read from the data set SASHELP.CLASS.
NOTE: PROCEDURE MEANS used (Total process time):
real time          0.01 seconds
user cpu time      0.01 seconds
system cpu time    0.01 seconds
memory            9505.96k
OS Memory         30900.00k
Timestamp         09/18/2023 09:10:46 PM
Step Count                103  Switch Count  1
Page Faults                0
Page Reclaims            2109
Page Swaps                0
Voluntary Context Switches 10
Involuntary Context Switches 0
Block Input Operations     0
Block Output Operations    8

```

The results in [Table 1](#) show the number of observations, the mean, standard deviation, minimum, and maximum ages for females (F) and males (M). There were 9 females in the data set with a mean age of 13.2222222, a standard deviation of 1.3944334, a minimum Age of 11, and a maximum Age of 15.

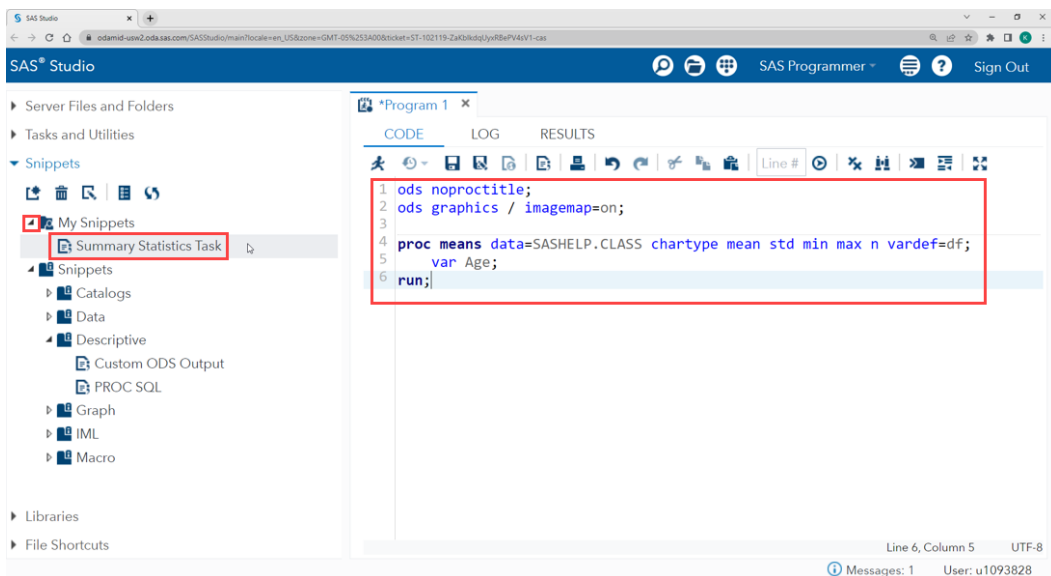
Table 1: Summary Statistics Task Results (PROC MEANS of Age by Sex)

Analysis Variable : Age						
Sex	N Obs	Mean	Std Dev	Minimum	Maximum	N
F	9	13.2222222	1.3944334	11.0000000	15.0000000	9
M	10	13.4000000	1.6465452	11.0000000	16.0000000	10

Now you know what syntax to run the next time you need to calculate summary statistics on a data set. You can copy the code generated by the task into another program or save it to your code snippets for future use.

2.3.2 Snippets

To add the code generated in the [Tasks](#) section to your snippets, highlight the code in the **Code** subtab, **right-click** the highlighted code, and then click **Add to My Snippets**. Give your code snippet a name like Summary Statistics Task, and then click **Save**. Now you can find this code snippet by expanding **My Snippets** under the **Snippets** section of the navigation pane. When you double-click the snippet, it will paste the code into the Code subtab of the program that you are currently working in ([Figure 12](#)). You will need to edit the data set name and variable names to reflect any new data set or variables that you would like to run summary statistics on.

Figure 12: Insert Code Snippet into Program

2.4 Data Setup

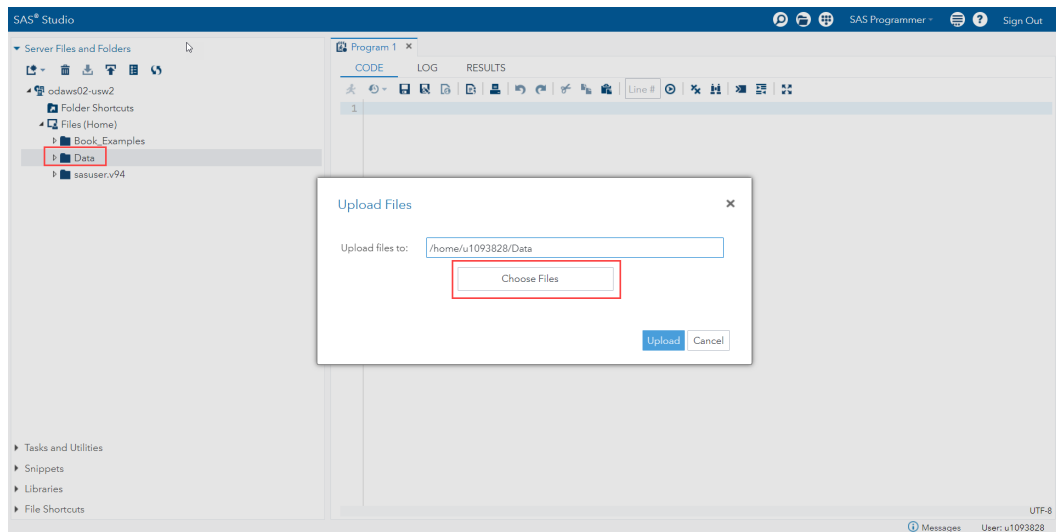
The data used in this book is available for download from the SAS Author Page found at <https://support.sas.com/en/books/authors/kirby-thomas.html>. Once the data has been downloaded, save it to your computer. I created a folder called Book_Data on my desktop and saved the files in a subfolder called Input_Data. View the properties of the folder that you save the book data to and make note of the location. In my case, the location is: C:\Users\Kirby\Desktop\Book_Data\Input_Data. When using SAS® Enterprise Guide® or the SAS Windowing environment, create a LIBREF called INPUT that references the location of your saved data files. In my case, I would begin my program with the following statement:

```
LIBNAME input 'C:\Users\Kirby\Desktop\Book_Data\Input_Data';
```

Update the location in single quotation marks to reflect where your data is saved.

If you are using SAS® Studio, you will need to upload the data files to the SAS server before you can work with them. First, create a folder to hold your book data. Right-click **Files** in the navigation pane and select **New > Folder**. I named my folder Data. Clever. I know. Now, right-click the folder you just created, and click **Upload Files** ([Figure 13](#)).

Figure 13: Upload Files



When you click **Choose Files**, the file explorer opens and enables you to navigate to where you saved your data files for this book. You can select one data file or hold down the **Ctrl** button and select several files to upload them all at once. Next, click **Open**. A list of selected files appears in the Upload Files window, and you can click the **Upload** button to upload the files to the SAS server into your Data folder.

Right-click the **Data** folder that you uploaded the files to, and then click **Properties** to copy this location. In my case, the location is: /home/u1093828/Data. Create a LIBREF called INPUT at the beginning of your SAS® Studio programs to reference any files found in this folder location:

```
LIBNAME input '/home/u1093828/Data';
```

The INPUT LIBREF will be used throughout the code examples to reference the location where the book data files are saved. More information about creating LIBNAMES can be found in the [SAS Libraries](#) section.