

Two documents on SAS On Demand for Academics (ODA) from two online sources.

Document 1: Copied from https://www.9to5sas.com/sas-on-demand/?expand_article=1 – Thanks to the author!

1. [Getting Started with SAS On-Demand for Academics](#)
 1. [Step 1: Create Your Sas Profile](#)
 2. [Step 2: Register for Sas On-demand for Academics](#)
2. [Logging in to SAS OnDemand for Academics and Accessing SAS Studio](#)
3. [The SAS Studio Environment \(ODA version\)](#)
4. [Uploading Data and Files to SAS On Demand for Academics](#)
5. [Accessing Uploaded Data with a LIBNAME Statement](#)

SAS On Demand for Academics (ODA) provides free access to SAS Studio for students and professors via a web browser. SAS on Demand's advantage is that it requires no installation and runs on the cloud over the internet, processing data as it connects to the cloud. However, you must create a SAS profile and register for SAS OnDemand for Academics.

It's an excellent choice for students who want to use SAS on a computer that doesn't support the desktop version. We'll begin this tutorial by discussing how to access SAS Studio using SAS ODA. Following that, we'll discuss the SAS Studio user interface and how to import data into your account.

Getting Started with SAS On-Demand for Academics

If you are an independent learner and not associated with a college or university, you can learn SAS and sharpen your analytics skills with free access to SAS On Demand for Academics: SAS Studio. You will also get access to

- Free video tutorials will teach you the basics of SAS programming and statistical analysis.
- Two free e-learning courses – [Programming 1](#) and [Statistics 1](#).
- An online [community](#) where you'll find forums, software support, instructional videos, and more.

Before you can begin using SAS On Demand for Academics, you must first create a SAS Profile and register to use SAS OnDemand for Academics.

Step 1: Create Your Sas Profile

You must first create a SAS Profile before registering with SAS ODA. If you are an SAS user, you may already have an account and can proceed to the next step. However, if you

are new to SAS and need to create a profile, go to the [SAS ODA](#) and select “Don’t have a SAS Profile.”

Step 2: Register for Sas On-demand for Academics

After you’ve entered all the required information to create a SAS profile, you’ll see the message below on the webpage. To activate your SAS Profile, follow the on-screen instructions. Next, you will see a confirmation page and receive a follow-up email from SAS to find your user ID for SAS ODA. You can use this user ID or your email address to sign in on the SAS ODA sign-in page.

Logging in to SAS OnDemand for Academics and Accessing SAS Studio

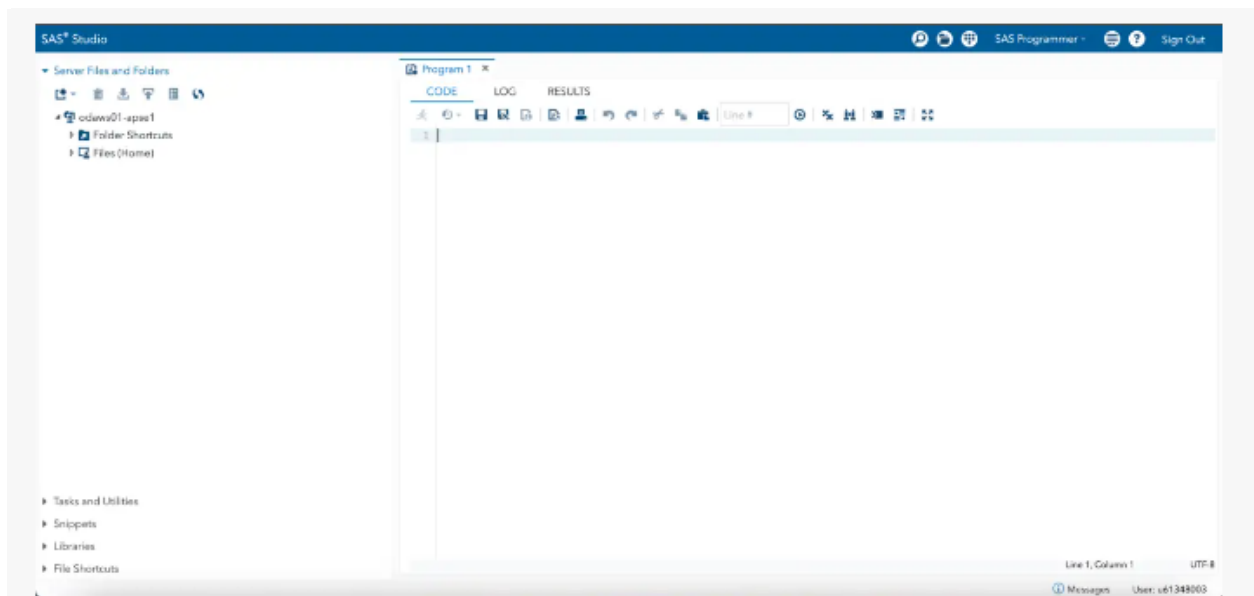
After successfully registering for SAS OnDemand for Academics, you can now log in to the software.

Enter your user ID or email address and password to sign in. Once logged in, you can access SAS Studio and the SAS ODA dashboard.

The free quota limit for the home directory is 5 GB; if you upload a dataset, it must be under 1 GB.

The SAS Studio Environment (ODA version)

SAS Studio refers to the actual SAS software that runs in the browser, and “SAS On Demand for Academics” is the service that allows users to access SAS Studio.

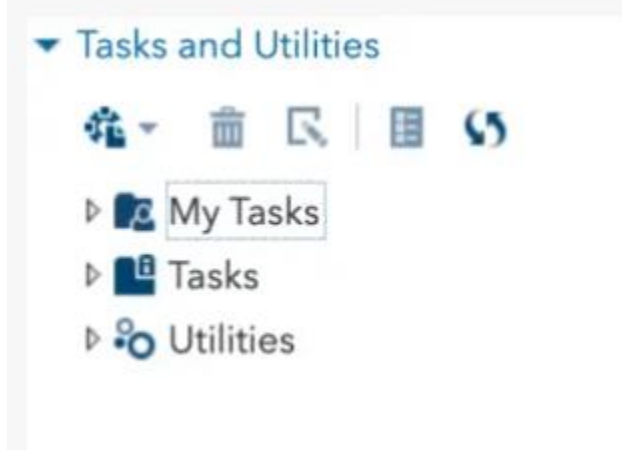


After launching SAS Studio, the *navigation pane* appears on the left, and the *program window* appears on the right. By default, the *program window* opens to the *Code* tab.

Server Files and Folders. The left-hand side of SAS Studio is dedicated to the *Navigation Pane*. The first immediately visible items are the server files and folders by default.

On SAS OnDemand for Academics, you can upload or download data and SAS files to or from your local computer.

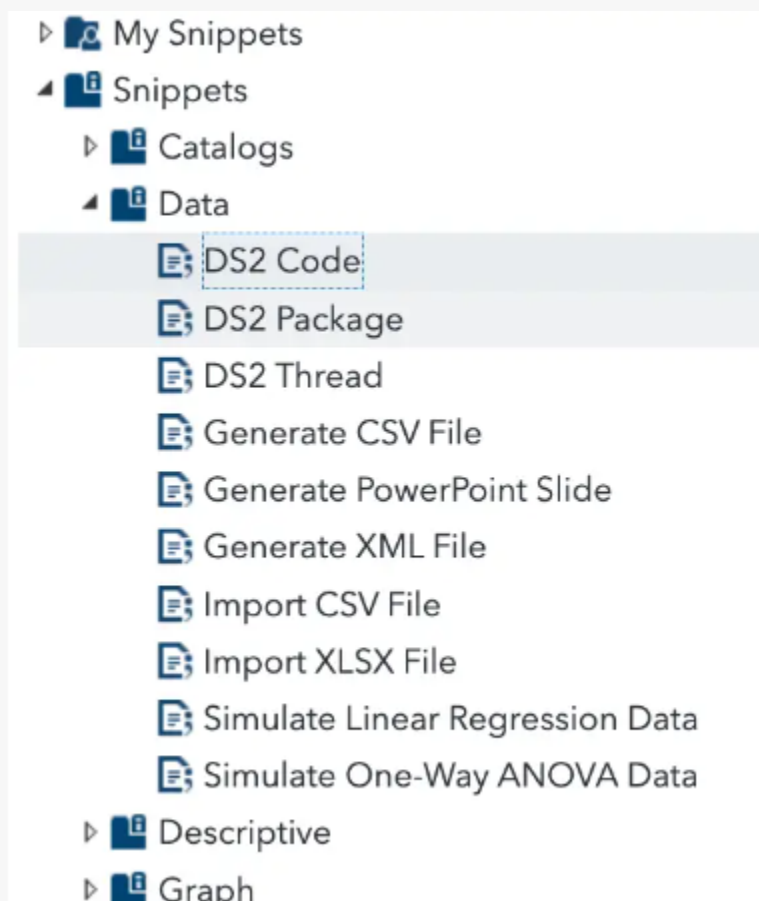
Tasks and utilities: There are a few other dropdowns within the navigation pane. First is **Tasks and Utilities**. If you click on the arrow next to **Tasks and Utilities**, SAS Studio will show you more options:



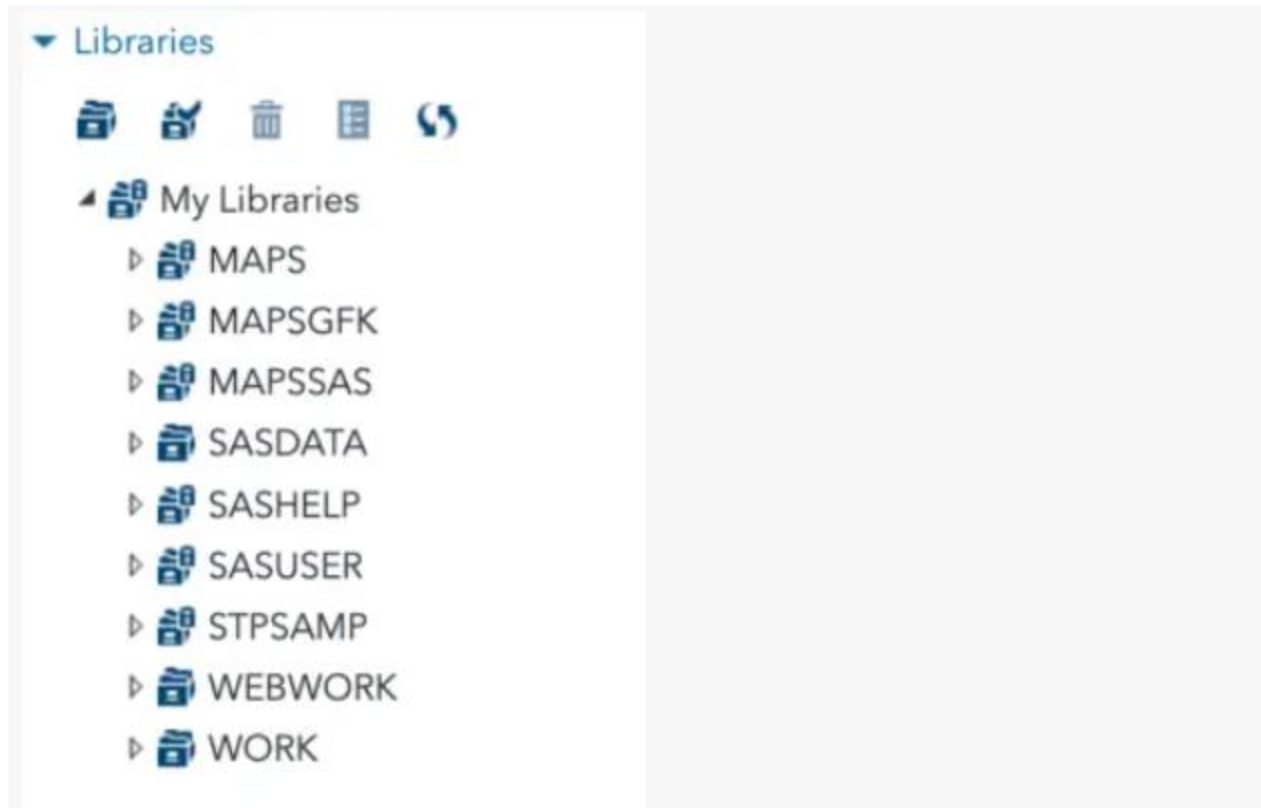
Tasks are SAS Studio's version of a point-and-click interface. This is a viable option if you are not comfortable writing code for SAS programs.

SAS Studio has predefined tasks that generate SAS code and format results. In addition, you can import data within Tasks and Utilities using Import Data or code in a new Program window.

Snippets: Code snippets are lines of SAS code that you may copy and paste into your program. It allows you to quickly insert SAS code into your program and modify it to your requirements. SAS Studio comes with several code snippets. You may also make your snippets and save them to your favourites list. Below are some of the built-in snippets that are available in SAS studio.



Libraries: The second drop-down within the *Navigation Pane* is **Libraries**. If you click on the arrow next to "Libraries," you will see the following icons:



The four default libraries – SASHELP, SASUSER, WEBWORK, and WORK – are visible when you first open SAS Studio.

If you define other SAS libraries during your session, those will also appear in this window. Clicking on any of these icons will open that library, where you can browse the stored data files.

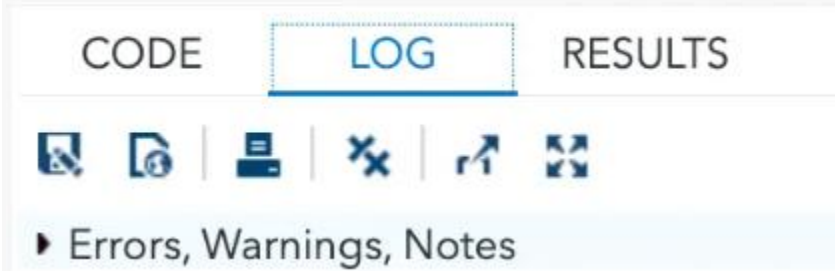
SAS libraries are essential for working with data in SAS, so we recommend checking out our tutorial on [SAS Libraries](#).

Code: The *Code* tab within the *Program* window is where you create and edit your SAS program.

The *Code* tab has a variety of icons in the toolbar. Most options available on the toolbar are similar to those in other programs.

LOG: The Log tab is one of the essential tools you can use to debug problems in SAS! This is where error messages and warnings are displayed, which might help troubleshoot a program that isn't working correctly.

There are several icons available on the Log tab's toolbar. These can be used to save or print the log, download it as an HTML file, open a new browser tab, clear the log, or maximise the window view. These icons are as follows:



Results: The *Results* tab displays the printable results of any analysis you submitted in your program.

The *Results* tab also has a variety of icons. These allow you to download the results as HTML/PDF/RTF files, download or print the generated data, or open in a new browser tab:



Uploading Data and Files to SAS On Demand for Academics

To upload files in SAS Studio, select **Files(Home)**. Then right-click on the *my_content* folder and select **Upload Files..** from the dropdown. You may also create a folder here. All the files and folders will be uploaded to the server's Home directory.

Local data such as SAS datasets or .csv files can be uploaded within the SAS studio and other SAS On Demand for Academics applications.

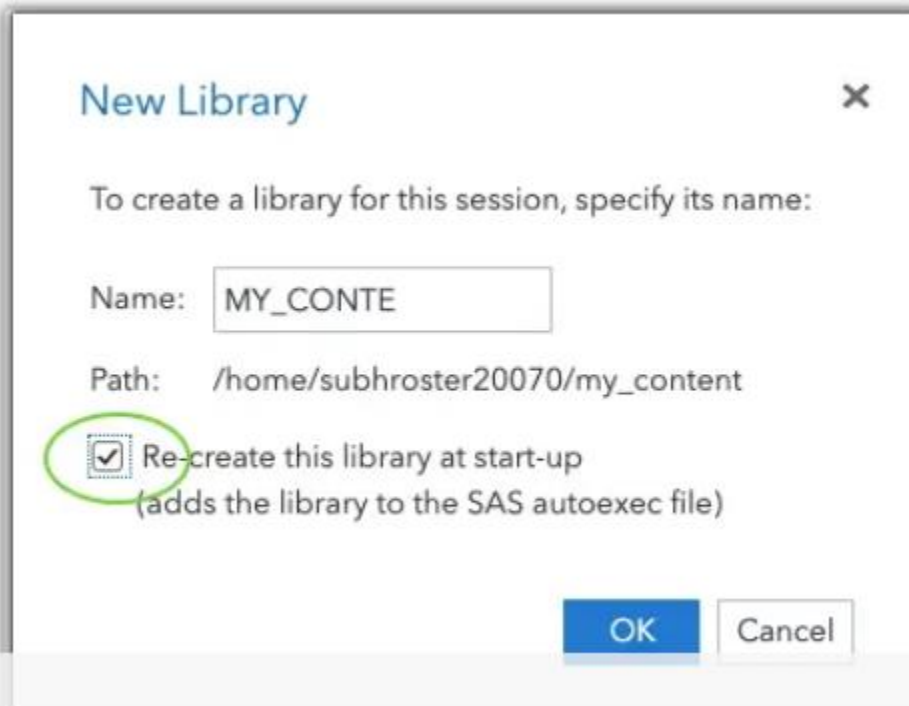
Accessing Uploaded Data with a LIBNAME Statement

The uploaded data can be used within SAS studio after you have uploaded it. You can either use a `LIBNAME` statement in your code or create a Library by following the below steps.

1. From within SAS Studio, expand the Folders panel and locate the folder to which you uploaded data.
2. Right-click on the folder and select “Create Library.”

Accessing Uploaded Data with a LIBNAME Statement

3. Provide a name for the library. The library will be created and will be available from the Libraries panel. Enabling Re-create this Library at start-up will automatically assign the library when you start the SAS session again.



New Library

To create a library for this session, specify its name:

Name: MY_CONTE

Path: /home/subhroster20070/my_content

☒ Re-create this library at start-up
(adds the library to the SAS autoexec file)

OK Cancel

If you have uploaded a SAS data set, you and your students can access the data using the LIBNAME statement below.

You can include the `access=readonly` option to prevent your students from modifying the data.

```
libname example "/path" access=readonly;
```

So, this was our side on the steps to access SAS Studio on the cloud. We hope this article helped you to access SAS Studio using SAS on Demand for Academics.

Moreover, if you have any other suggestions, suggest them below the comment section. We would take those lists in our further blog post.

Read the documentation below for a detailed description of the section, **Accessing Uploaded Data** ... in the previous documentation.

DOCUMENT 2: SAS ODA DOCUMENTATION (AUTHORED BY KENT STATE UNIVERSITY LIBRARY)
<https://libguides.library.kent.edu/SAS/OnDemandImportData> (copied here part of it) – Thanks to the author!

SAS TUTORIALS: IMPORTING DATA INTO SAS ONDEMAND FOR ACADEMICS

If you want to work with your own data in SAS OnDemand for Academics, you'll first need to upload it to your account. This tutorial shows how to upload and import SAS and non-SAS data files (such as Excel and CSV) into SAS ODA for analysis.

How to Get Your Own Data into SAS OnDemand for Academics

When starting a data analysis project in SAS OnDemand for Academics, your data files may or may not be in the SAS data file format (*.sas7bdat). Your data might be an Excel spreadsheet, an SPSS dataset, or a text or CSV file.

Regardless of what format your data is in, the process of getting your data into your SAS OnDemand for Academics account requires two steps. First, you must upload the file to your SAS OnDemand for Academics account -- that is, your data file must actually be uploaded to the cloud. Then, you must "import" the uploaded file into SAS Studio (using a process that is similar to what you would do with "desktop" SAS). The first step is unique to those using SAS OnDemand for Academics (if you were using "desktop" SAS, you could simply import the file directly from your computer), but fortunately only needs to be done once.

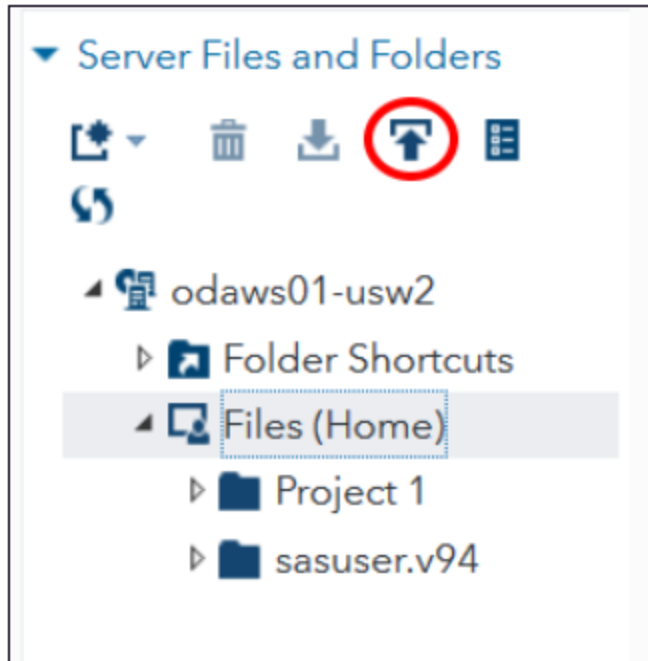
In this tutorial, we'll cover how to get data files from your computer into your SAS OnDemand for Academics account so that you can analyze them.

Uploading Data Files to Your SAS OnDemand for Academics Account

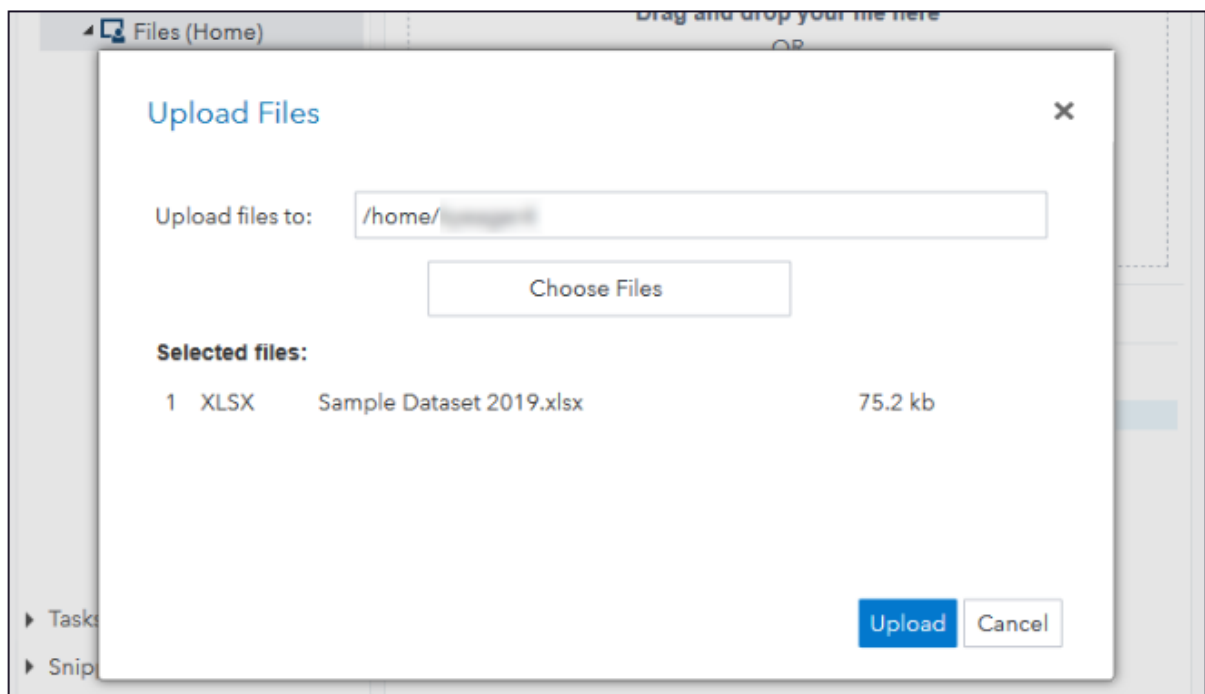
If you are using SAS OnDemand for Academics and have data files you want to work with in the program, you will need to upload data to your account. This goes for ANY dataset you want to work with in SAS ODA, including "native" SAS datasets (*.sas7bdat) or other formats such as Excel and CSV.

To upload a file to SAS ODA:

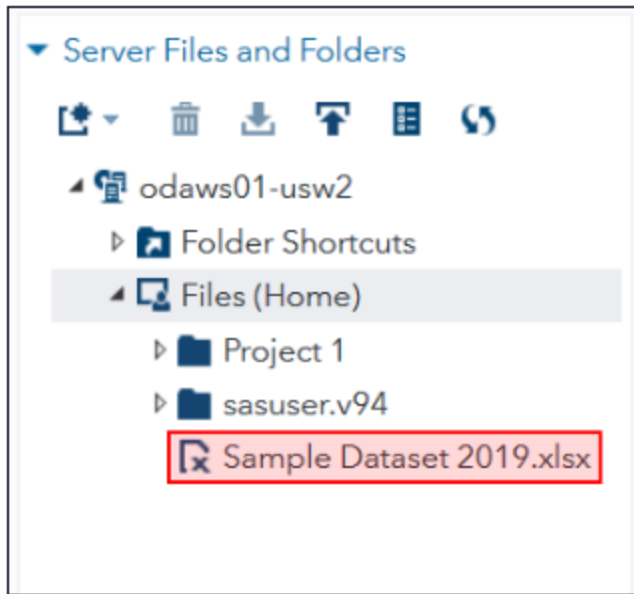
1. In the Navigation pane, click Server Files and Folders.
2. Left-click on the folder you want to upload the data file to (usually "Files (Home)" or a folder within that directory). Then click **Upload**. (Note that the Upload button will not be clickable until a folder location is selected, so if it's greyed out, make sure you've selected a folder.)



3. The Upload Files window will open. Click **Choose Files**, then choose the data file on your computer to upload.



4. SAS will show the name(s) and size(s) of the file(s) you selected. To complete the upload, click **Upload**.
5. If successful, you should see the data file(s) appear in the directory you selected in step 2:



At this stage, if you have uploaded a SAS dataset file (*.sas7bdat) and have already [mapped a library to the folder where the data file is located](#), you can skip the next part of the tutorial and start working with the data right away! However, if you have uploaded a text, CSV, Excel, SPSS, or other non-SAS format, you'll need to proceed to the next step: importing your uploaded data.

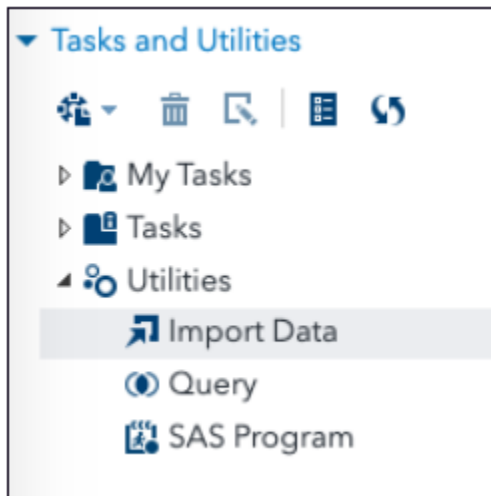
Tip

If you have SAS code files (*.sas) on your computer that you want to use in SAS OnDemand for Academics, you can upload them to your account using these same steps. After uploading *.sas files to your account, you can open and edit them just like a SAS script created directly in ODA.

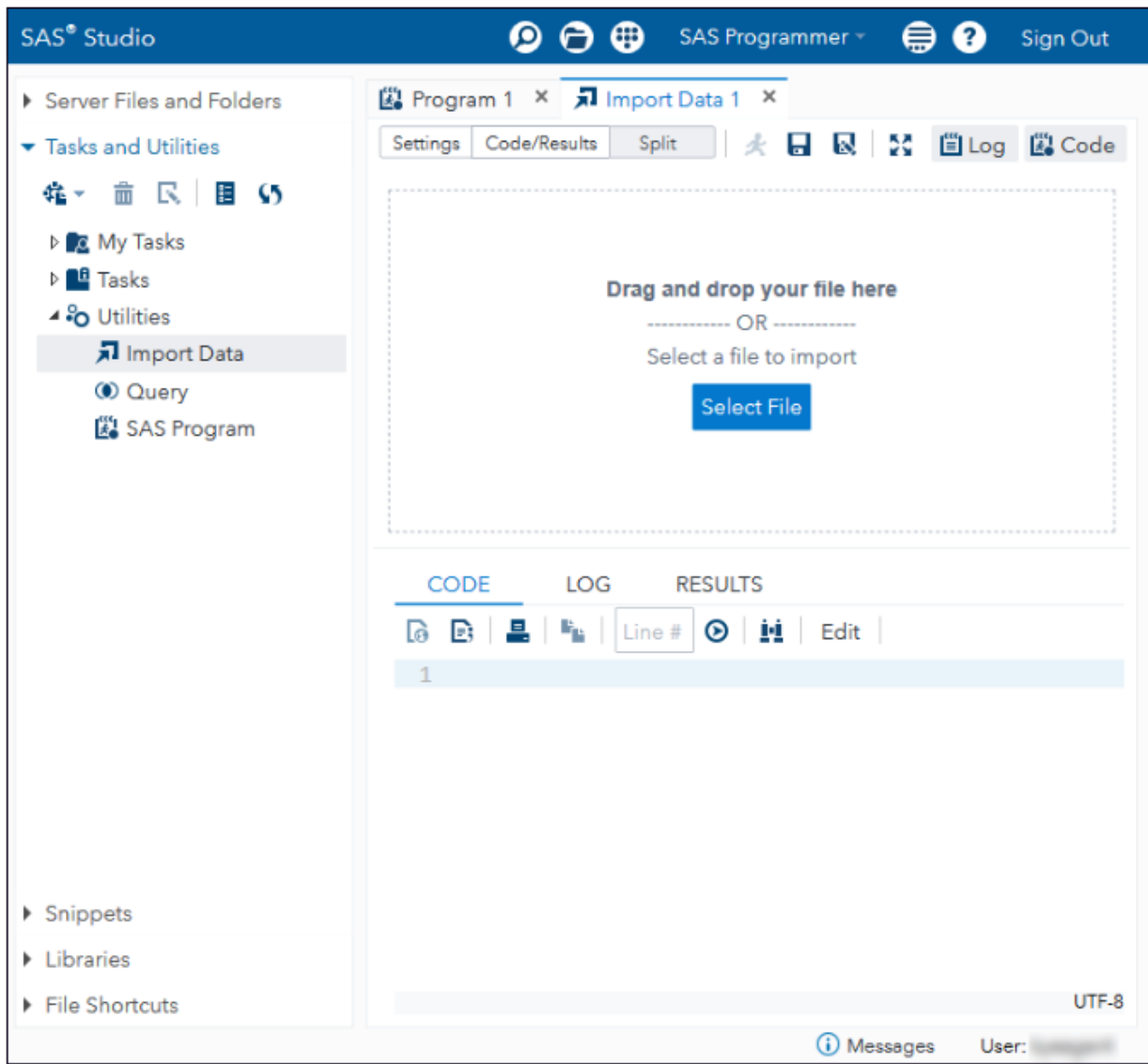
Importing Uploaded Non-SAS Format Data (Excel and CSV)

After your datafiles have been uploaded to your SAS OnDemand for Academics account, you can import the data into a SAS data step. The easiest way to do so is to use the Import Data task:

1. Click **Tasks and Utilities** in the Navigation pane, then click **Import Data**.



This will create a new tab in the work area, titled Import Data 1.



2. You can decide to **drag and drop your files** or **select a file** to import. In our case, the Excel file with the sample data is located in our *Sample data* folder. (Remember: when using SAS OnDemand for Academics, your data must be uploaded into the *Server Files and Folders* tab. If you do not see any data files in this window, you most likely need to upload them; see the previous section on uploading data to SAS ODA.)
3. Once you've chosen the Excel file with the data, click **Open**. SAS will populate the file information and generate code for importing.

*Sample Dataset x

Settings

Code/Results

Split

FILE INFORMATION

SOURCE FILE

File name:

Sample Dataset.xlsx

Source location:

/home/ /Sample data

Worksheet name:

User ID

First worksheet

OUTPUT DATA

SAS server:

SASApp

Data set name:

IMPORT

Library:

WORK

Change

OPTIONS

File type:

DEFAULT (Based on file extension)

☒

Generate SAS variable names

4. To run the code, click the **Run** button (running man icon) to execute the code.
5. Once you've executed the code, check the *Log* to see if any errors have occurred. If successful, the Log window should tell us how many observations and variables the new dataset contains.

```
NOTE: The import data set has 435 observations and 32 variables.
NOTE: WORK.IMPORT data set was successfully created.
NOTE: PROCEDURE IMPORT used (Total process time):
      real time           0.33 seconds
      cpu time            0.24 seconds
```

Additionally, you can preview the dataset in the *Output Data* tab.

***Import Data 1** x

Settings Code/Results Split Log Code

CODE LOG RESULTS **OUTPUT DATA**

Table: WORK.IMPORT View: Column names Filter: (none)

Columns Total rows: 435 Total columns: 32 Rows 1-100

	ids	bday	Gender	Age
1	20055	08/06/1995	0	24
2	20075	.	0	
3	20087	.	0	
4	20088	01/08/2001	1	18
5	20135	12/26/1998	0	20
6	20161	12/02/2000	1	18
7	20188	08/06/2001	0	18
8	20215	08/05/1997	0	22
9	20250	12/14/1994	.	24
10	20354	08/05/1999	1	20
11	20640	05/31/2000	1	19
12	20739	06/23/1999	1	20
13	20990	06/08/1998	1	21

Property Value

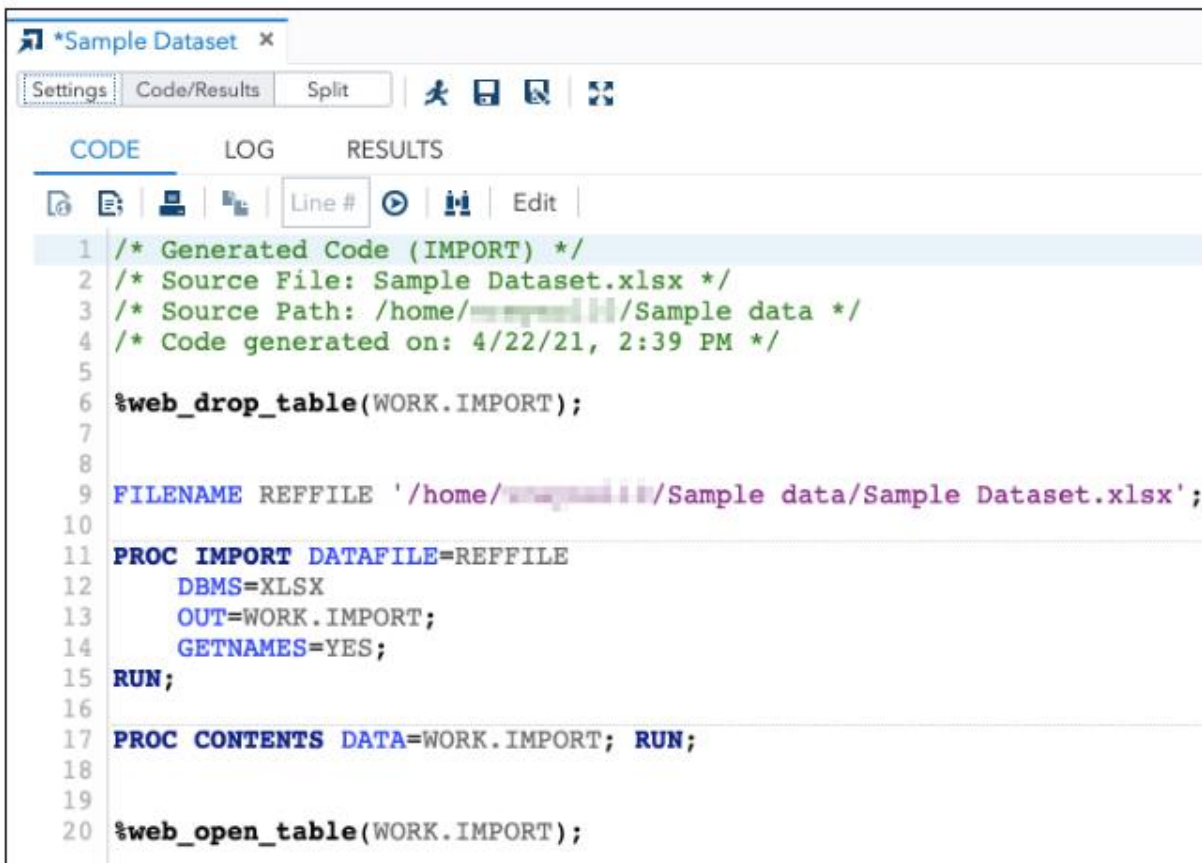
Label

Name

- If you are satisfied with the result of the import process, you have the option to save the Import task by clicking the **Save** button (floppy disk icon). This will create a *.ctl file in your SAS ODA account, which you can use to repeat/re-create the import process in the future. We highly recommend doing this so you have a record of how you imported the data.

UNDERSTANDING THE IMPORT SYNTAX

Let's take a closer look at the syntax that SAS generates when using the Import tool and understand what each line is doing:



The screenshot shows the SAS IDE interface with a window titled '*Sample Dataset'. The 'CODE' tab is active, displaying the following SAS code:

```
1 /* Generated Code (IMPORT) */
2 /* Source File: Sample Dataset.xlsx */
3 /* Source Path: /home/---/Sample data */
4 /* Code generated on: 4/22/21, 2:39 PM */
5
6 %web_drop_table(WORK.IMPORT);
7
8
9 FILENAME REFFILE '/home/---/Sample data/Sample Dataset.xlsx';
10
11 PROC IMPORT DATAFILE=REFFILE
12     DBMS=XLSX
13     OUT=WORK.IMPORT;
14     GETNAMES=YES;
15 RUN;
16
17 PROC CONTENTS DATA=WORK.IMPORT; RUN;
18
19
20 %web_open_table(WORK.IMPORT);
```

The first four lines are comments -- they do not contain executable code, and exist only as notes for human readers. The notes contain the name and location of the imported file, and the date that the import code was generated.

The first executable line of code is

```
%web_drop_table(WORK.IMPORT);
```

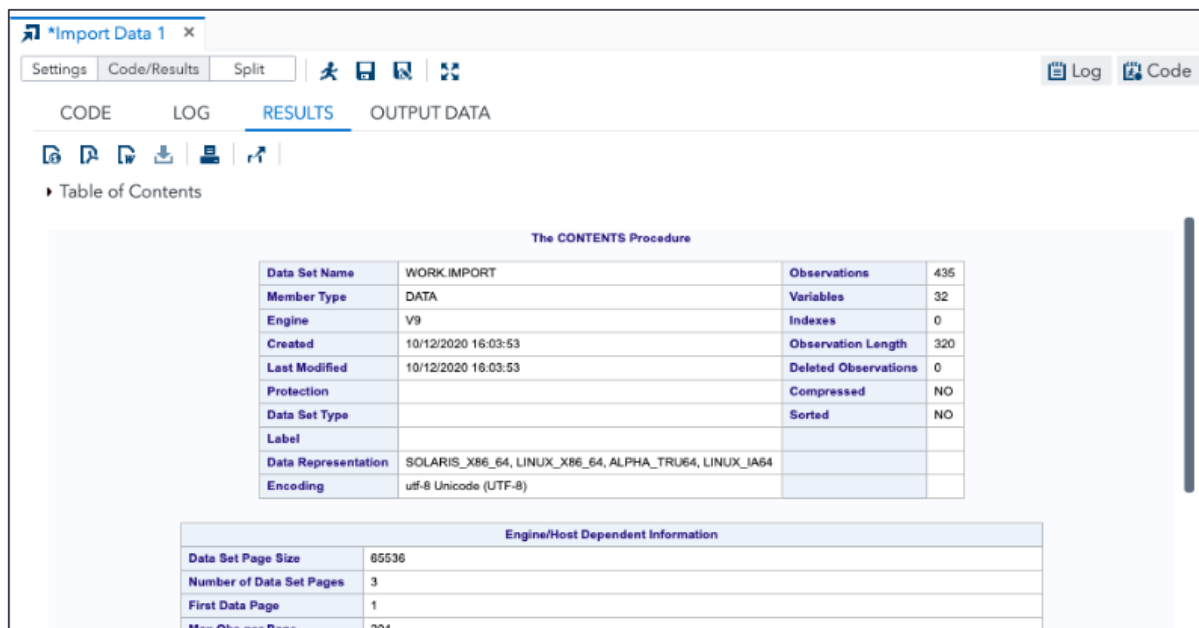
This particular command "clears out" any data in the system saved under the WORK.IMPORT name. You might notice that this command looks very different than the PROC IMPORT and PROC CONTENTS commands. Notice the percent symbol (%) at the start of the line: this tells us that **%web_drop_table** is a user-defined macro command. In this case, it's a function that SAS themselves defined to quickly delete a particular dataset. **If you've imported another dataset into SAS using the Import task under the name WORK.IMPORT and want to keep it, make sure you save it under a name other than WORK.IMPORT, or else you will lose it!**

The first pertinent line starts with the **FILENAME** statement, which specifies the path to the file containing the data to import. In particular, it assigns the nickname **REFFILE** to the full file path where our data is located. This means that we can simply type the nickname *REFFILE* in the **PROC IMPORT** step instead of having to write out the full file path.

The next block of code is the **PROC IMPORT** statement, which triggers the data import action. Note the first semicolon is not until the third line. This is because the syntax includes quite a few options associated with the **PROC IMPORT** statement:

- The **OUT** option tells SAS where to put the new SAS dataset it is creating – in this case we told it to put the new SAS dataset “import” in the Work library.
- The **DATAFILE** option points to the file directory of the dataset you are importing.
- The **DBMS** option tells SAS what *engine* to use to read the data (in this case, the XLSX engine). This value will depend on what type of file you're importing.
- **GETNAMES=YES** instructs SAS to use the first row of the file as variable names. The PROC IMPORT code in this step is very similar to the PROC IMPORT code one would use for "desktop" SAS, so for more information about PROC IMPORT, see our [Importing Excel Files into SAS tutorial](#).

Following the **PROC IMPORT** command, SAS generates code for **PROC CONTENTS**, which displays information about the newly-imported dataset (WORK.IMPORT).



The screenshot shows the SAS 'Import Data 1' window with the 'RESULTS' tab selected. It displays the output of the PROC CONTENTS procedure for the dataset WORK.IMPORT. The output is presented in a table format with two main sections.

The CONTENTS Procedure			
Data Set Name	WORK.IMPORT	Observations	435
Member Type	DATA	Variables	32
Engine	V9	Indexes	0
Created	10/12/2020 16:03:53	Observation Length	320
Last Modified	10/12/2020 16:03:53	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Engine/Host Dependent Information	
Data Set Page Size	65536
Number of Data Set Pages	3
First Data Page	1
Max Obs per Page	204

PROC CONTENTS is a standard SAS procedure; for more information about how to use and understand it, see the [Summarizing Dataset Contents with PROC CONTENTS tutorial](#).

Importing Special Character Delimited Files

Data saved in Excel or *.csv formats should work with the Import Wizard's default choices, but data saved in *.txt format with a special character separator requires a bit of extra work. The process starts the same,

but if your data is saved as a .txt file, SAS requires you to explicitly say you're importing delimited data, and specify what character is used as the *delimiter*.



"Delimiters" are the character used to separate columns (i.e., variables) in the data file. In a CSV file, the delimiter is a comma; other common delimiters include tabs and spaces, but any character could be used. (The only way to know what character is used as the delimiter in a given *.txt file is to look at the data's accompanying documentation (if available), or look at the data file itself using a text editor.)





If the delimiter is not a white-space character, the character can be entered into the **Delimiter** field. If the delimiter is a white-space character, like a tab or a space, then the character's ASCII code must be used instead.

To import delimited data in a *.txt file, take the following steps:

1. Click **Tasks and Utilities** in the Navigation pane, then click **Import Data**.
2. Choose which uploaded file you want to import using the same steps described above. You can **drag and drop your files** or **select a file** to import. Once you've chosen the *.txt file you want to import, SAS will populate the file information and generate code for importing - but *don't run the code yet!*
3. Under **Options**:
 1. In the **File Type** list, select **DLM (Delimited File)**.
 2. In the **Delimiter** field, enter the character that is used as the delimiter:
 1. If the delimiter is not a white-space character (such as a comma or period), type the character surrounded by quotes (make sure the **Quote delimiter value** checkbox is cleared/empty).
For example, if the data file was a comma delimited file, in the **Delimiter** field, add **" , "**
 2. For white-space characters like tab or space, the character's ASCII code can be used to designate the delimiter. If an ASCII code is used, **do not** surround the ASCII code with quotation marks, and make sure the **Quote delimiter value** check box is cleared/empty. (The straight single quotes you see below are a literal part of the ASCII codes for tabs and spaces, and are not considered "quotation marks"; they should be left as-is.)
For tab separated files, use: **'09'x**
For space separated files use: **'20'x**
 3. In the **Start reading data at row** field, enter the row number corresponding to the first row containing your observations. In most *.txt files, your first row of observations will begin on line 2 (with line 1 containing the variable names), but this is not universally true. Some *.txt files may have multiple rows associated with variable metadata, while other *.txt files may not contain any variable metadata at all (i.e., the first line of the file will contain the observations).
4. When finished, click the **Run** button.

For tab-delimited data where the first line contains variable names and data begins on line 2, your inputs should look like the following:

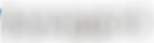
 *Import Data 1 

SettingsCode/ResultsSplit


FILE INFORMATION

SOURCE FILE

File name: **Sample Dataset 2019.txt**

Source location: **/home/**

End of line delimiter:

Default

OUTPUT DATA

SAS server: **SASApp**


Data set name: **IMPORT**

Library: **WORK**

Change

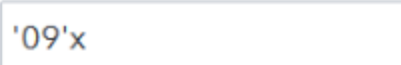
OPTIONS

File type:

DLM (Delimited file)

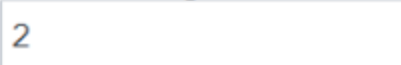
☒ Generate SAS variable names

Delimiter:


'09'x

☐ Quote delimiter value

Start reading data at row:

2

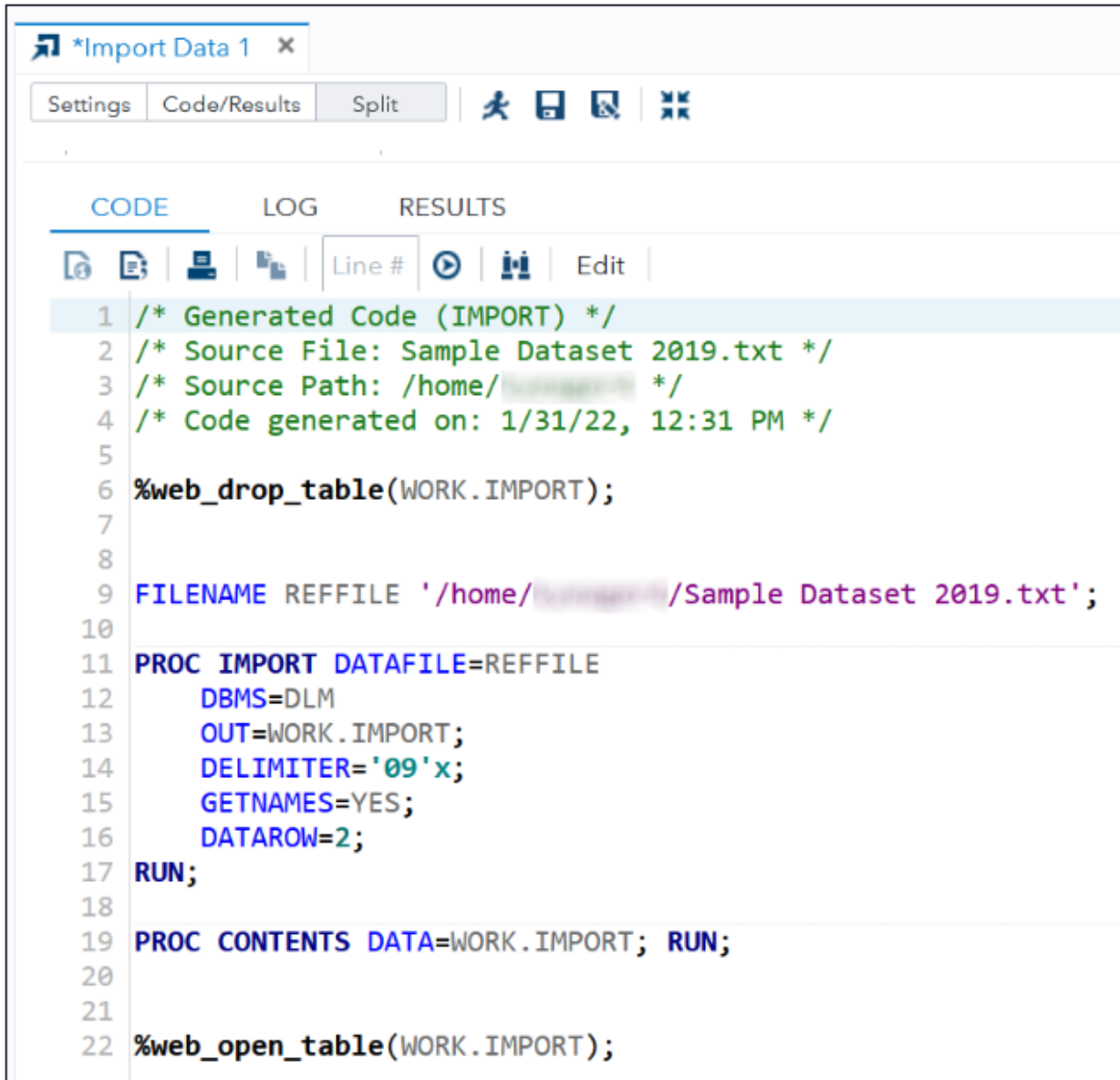
Guessing rows:

Default

18

UNDERSTANDING THE IMPORT SYNTAX (FOR DELIMITED FILES)

If you used the above setup, your generated import syntax should look like this:



The screenshot shows the SAS Studio interface with a window titled '*Import Data 1'. The 'Code/Results' tab is active, and the 'CODE' sub-tab is selected. The code editor displays the following SAS code:

```
1 /* Generated Code (IMPORT) */
2 /* Source File: Sample Dataset 2019.txt */
3 /* Source Path: /home/username */
4 /* Code generated on: 1/31/22, 12:31 PM */
5
6 %web_drop_table(WORK.IMPORT);
7
8
9 FILENAME REFFILE '/home/username/Sample Dataset 2019.txt';
10
11 PROC IMPORT DATAFILE=REFFILE
12     DBMS=DLM
13     OUT=WORK.IMPORT;
14     DELIMITER='09'x;
15     GETNAMES=YES;
16     DATAROW=2;
17 RUN;
18
19 PROC CONTENTS DATA=WORK.IMPORT; RUN;
20
21
22 %web_open_table(WORK.IMPORT);
```

The import syntax generated for delimited data is very similar to the syntax generated for Excel files described earlier, but there are several notable differences. For the Excel file, PROC IMPORT syntax contained four special parameters: **DATAFILE**, **DBMS**, **OUT**, and **GETNAMES**. The generated PROC IMPORT syntax for delimited files has these same four parameters, plus two new ones: **DELIMITER** and **DATAROW**.

Let's go over each one:

- The **PROC IMPORT** statement is associated with three options, all of which appear before the first semicolon: **DATAFILE**, **DBMS**, and **OUT**.
- **DATAFILE=REFFILE** option points to the file directory of the dataset you are importing. The term **REFFILE** is a "nickname" for the file path where the data is actually located, and is defined on line 9 of the script.
- The **DBMS=DLM** option tells SAS what *engine* to use to read the data, and differs based on what type of data you're importing. Because we are reading delimited data, we are using the DLM engine (instead of the XLSX engine).
- The **OUT=WORK.IMPORT** option tells SAS what to call and where to put the new SAS dataset. The name WORK.IMPORT means that the dataset will be named IMPORT, and it will be placed in the WORK library.
- The **DELIMITER='09'x** statement declares that tabs are the delimiter used in the file ('09'x is the ASCII code for the tab character).
- The **GETNAMES=YES** statement tells SAS to use the first row of the file as the variable names.
- The **DATAROW=2** statement says to begin reading data/observations on line 2.

