

ANLT5020 – Unit 2

Assignment 1 Tutorial

SAS Studio



Instructions

- Get your Patient.xlsx data into SAS using an appropriate procedure.
- Record the code used to accomplish this.
- Use PROC CONTENTS to describe the variables in the dataset.
- Run SAS procedures to find errors and/or missing data in numerical (PROC UNIVARIATE) and character variables (PROC FREQ).
- Identify obvious errors and missing data using the output from SAS procedures.
- Use PROC MEANS to calculate before and after heart rate averages.
- Research and summarize several different methods analysts may use to resolve problems of missing data when it poses a threat to quality.

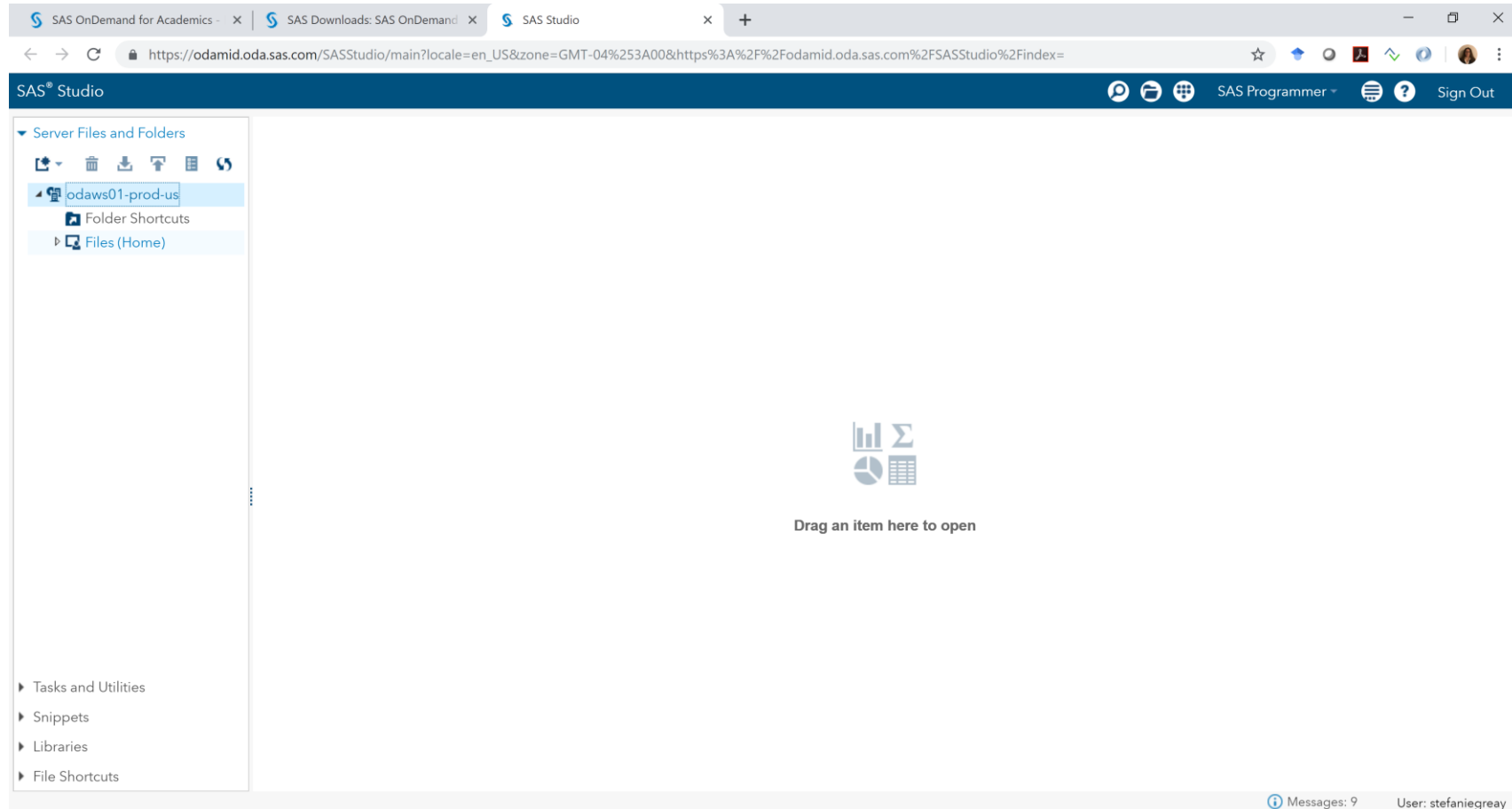


Dataset

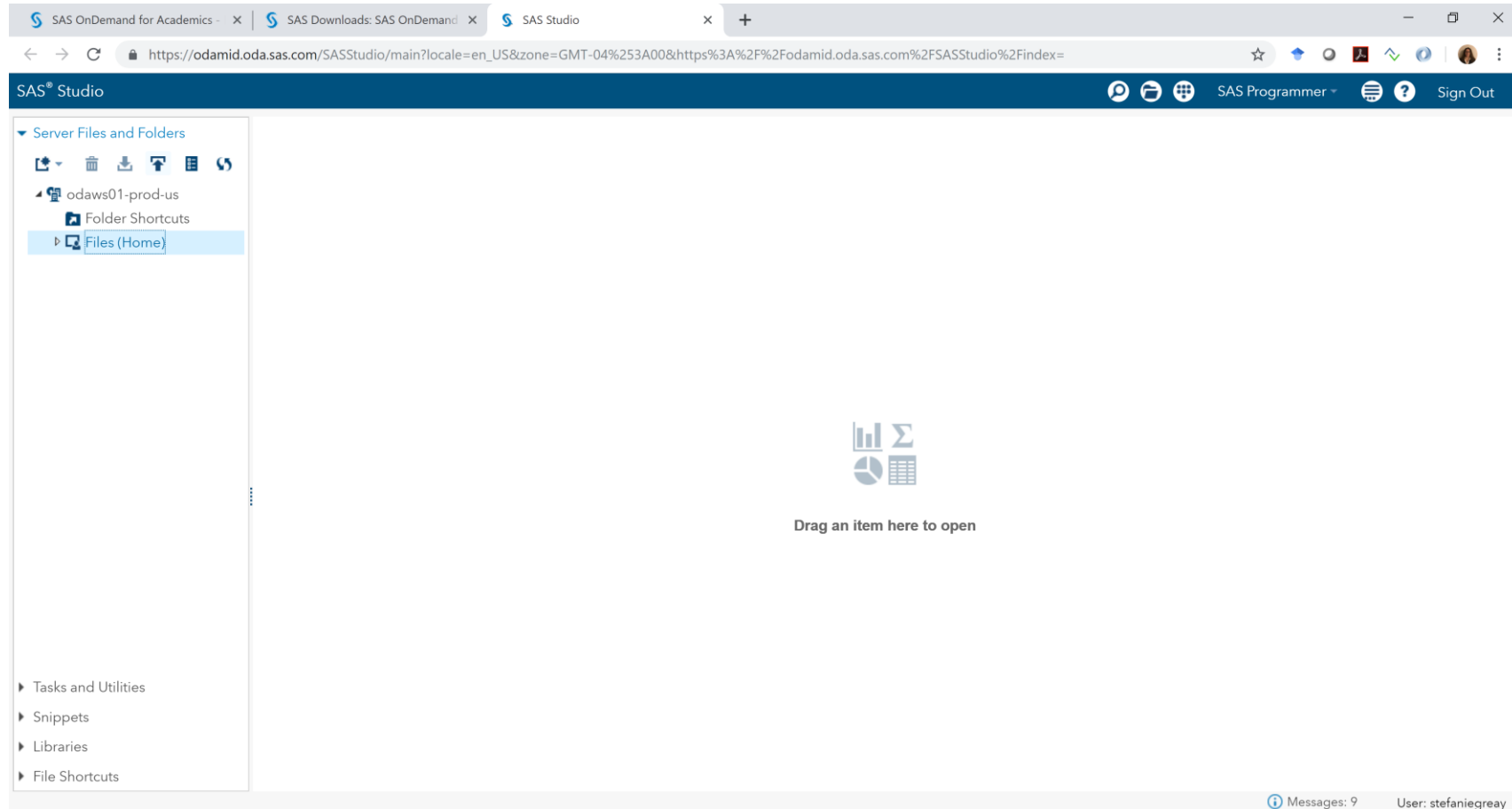
- Download the Patient.xlsx file from the course datasets zip file or from the Unit 2 Welcome announcement in the course announcements.



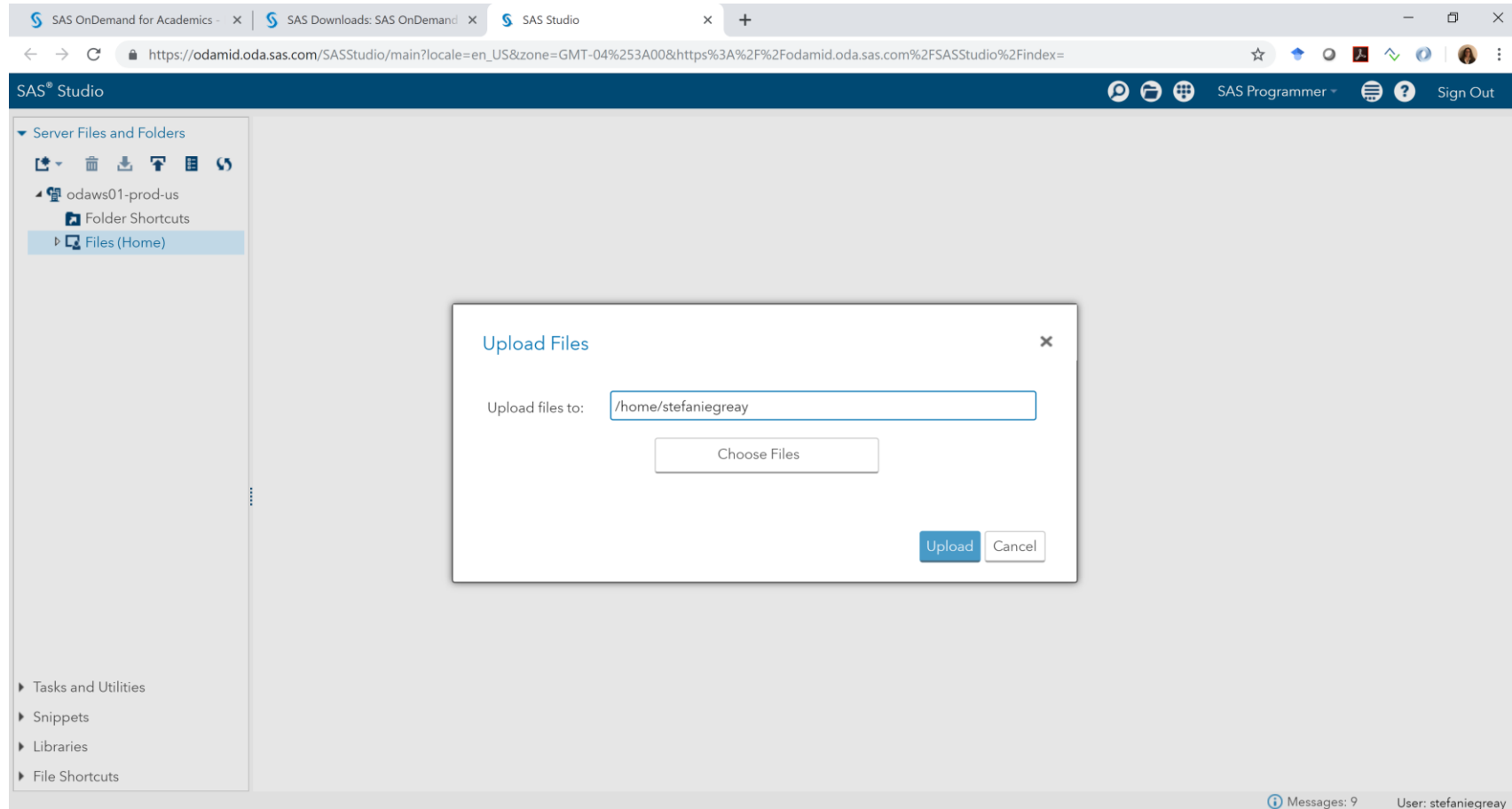
Click on Files(Home)



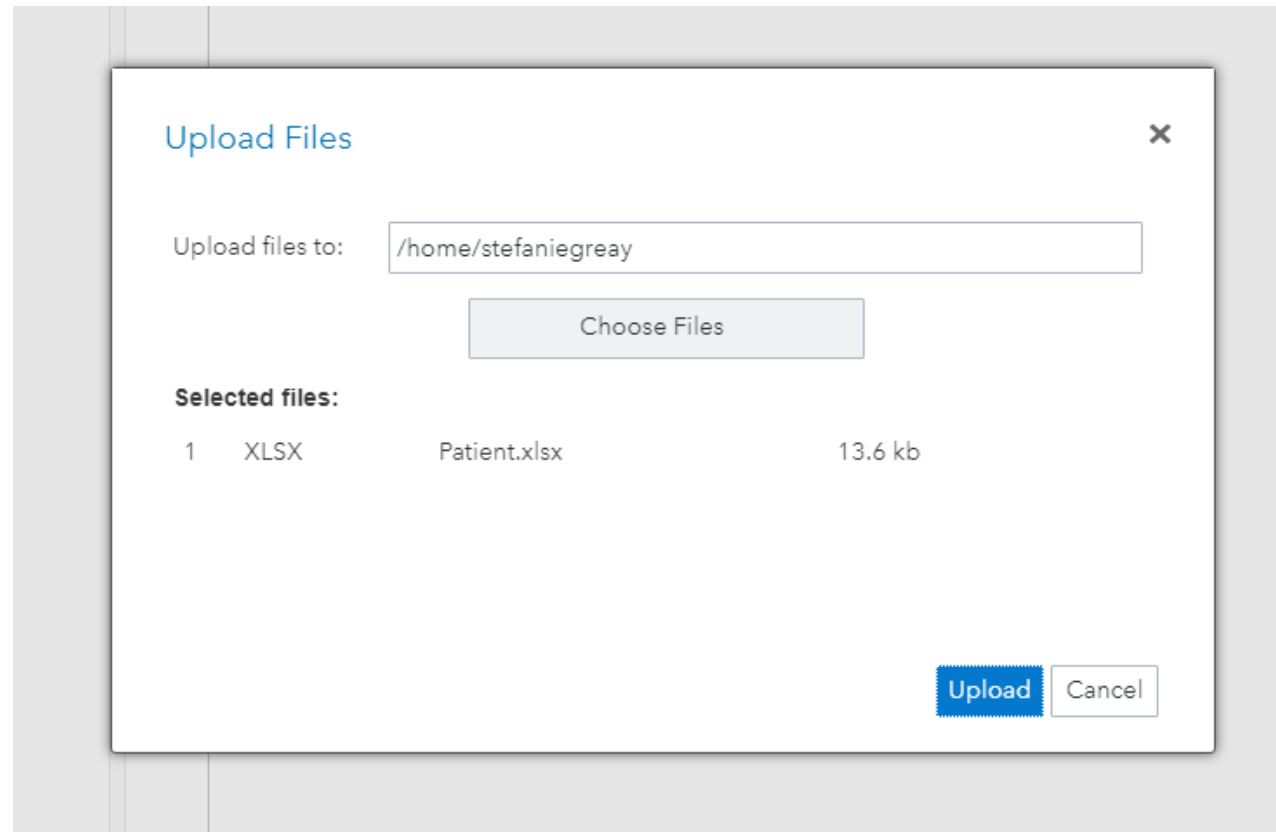
The Upload button will display in dark blue



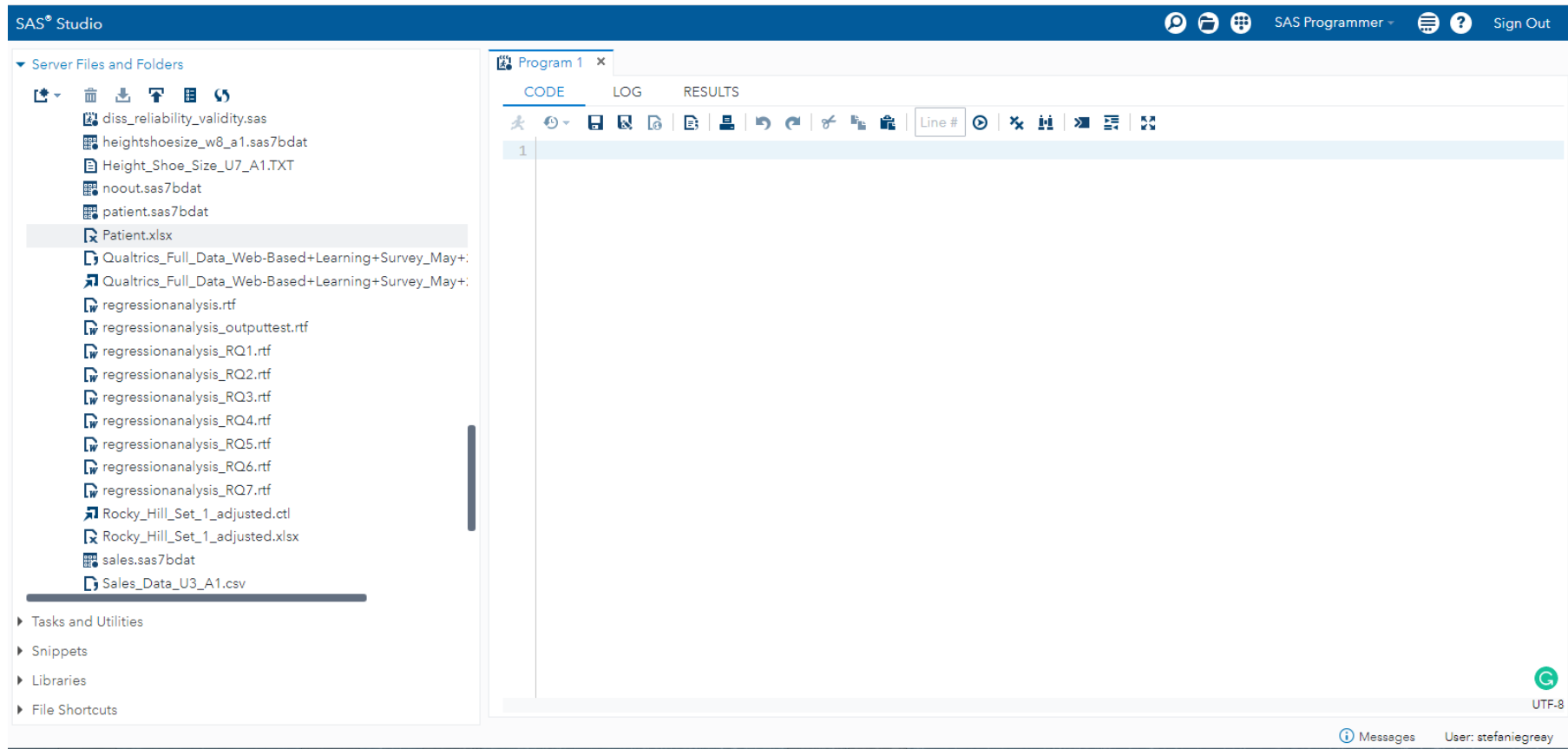
You can create a folder at this point, if you wish, or simply upload to your home directory.



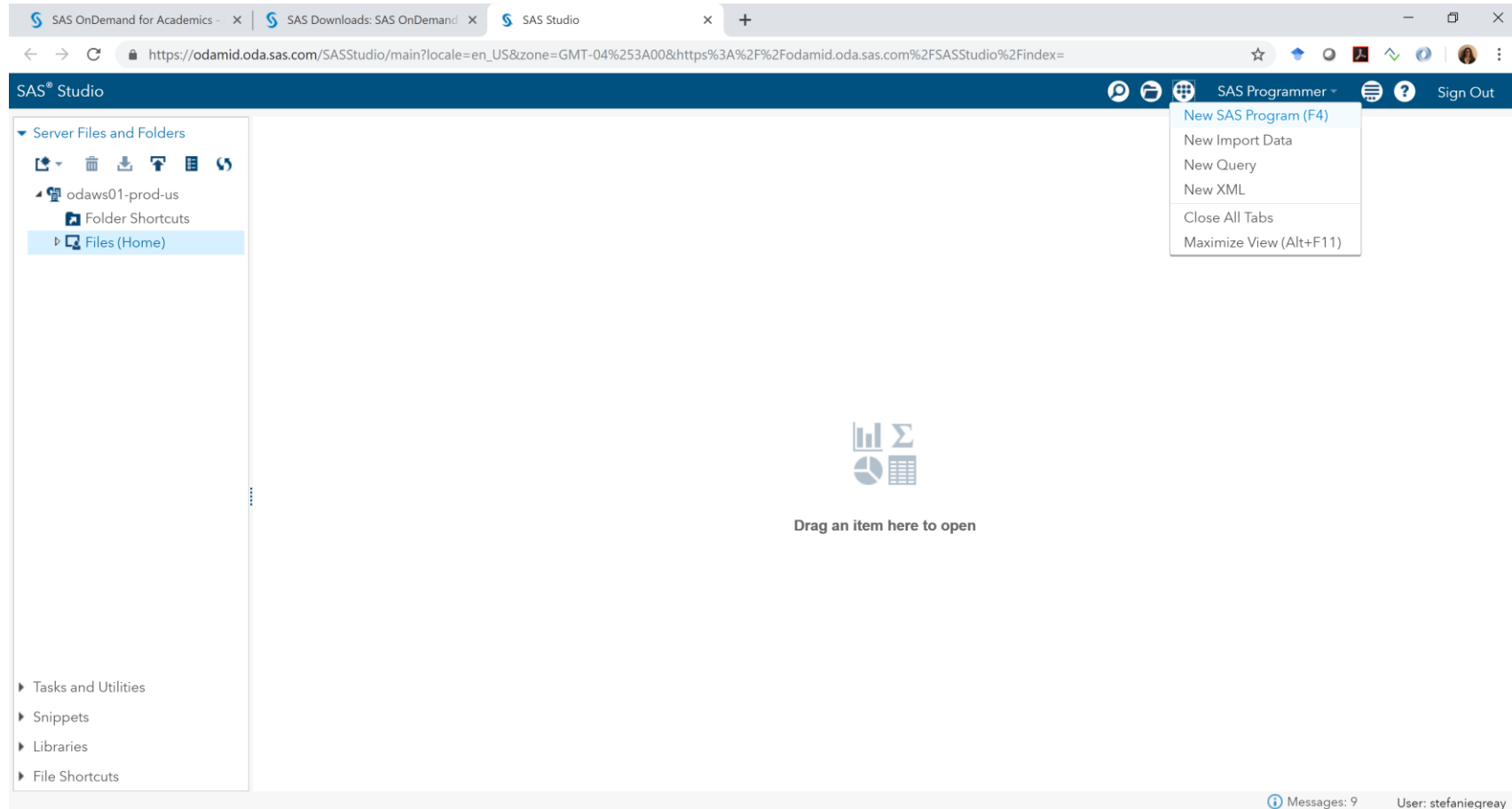
Select “Choose Files” to browse your computer for the dataset you want to upload. Once the dataset has been selected, click “Upload.”



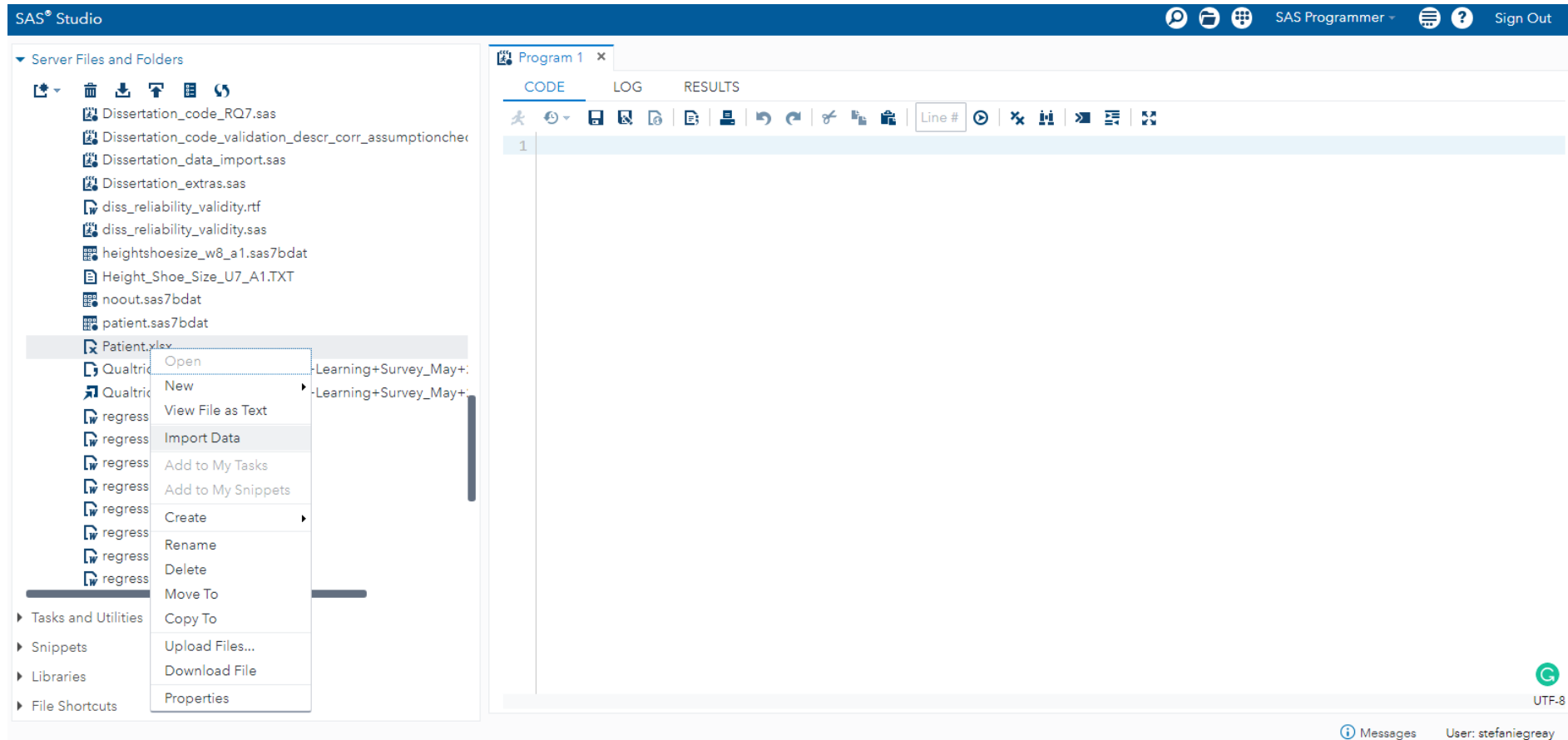
You will be able to view your files by clicking on “Files(Home)” to verify that your file successfully uploaded.



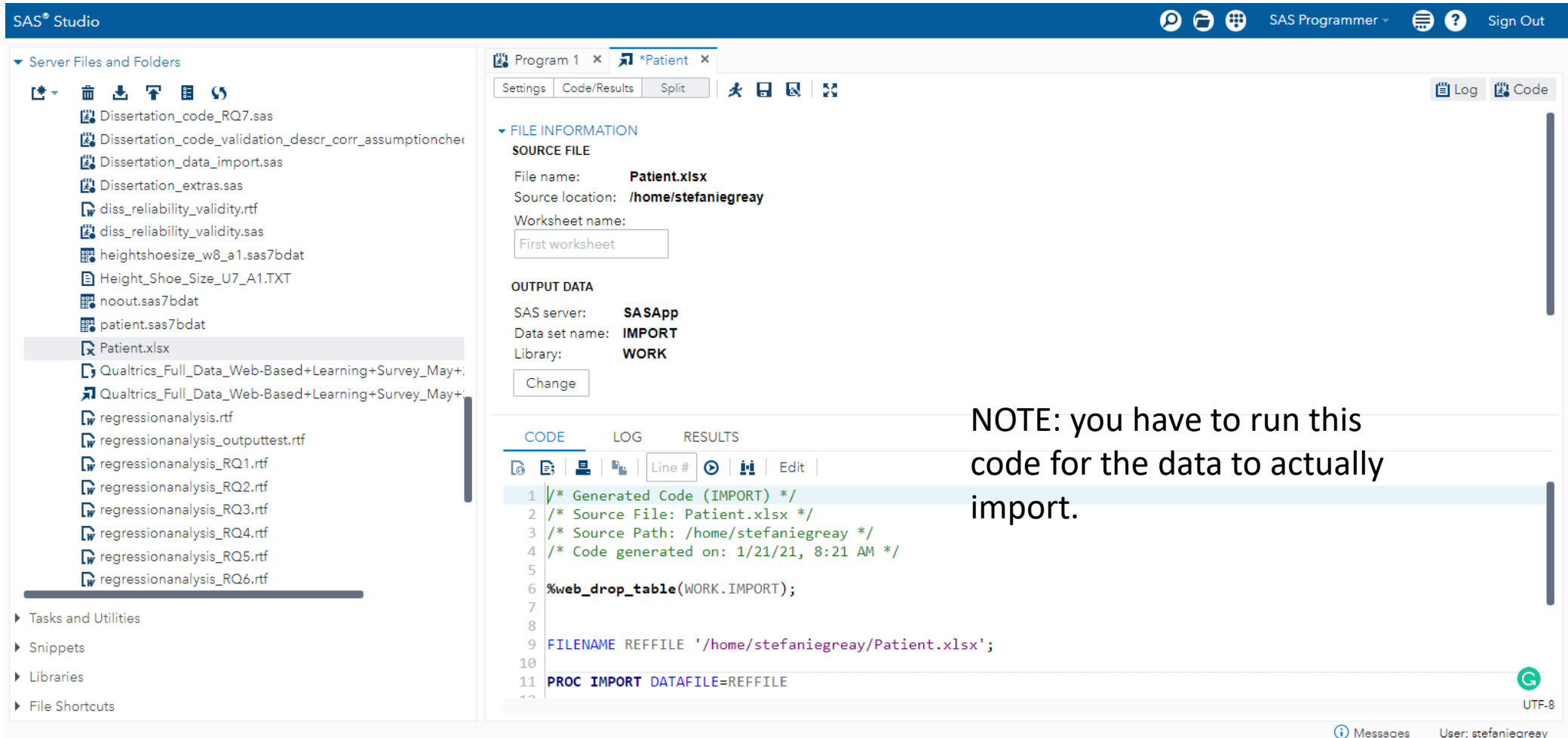
To get started with the SAS portion of the Unit 2 Assignment 1 assignment, start a new SAS program.



Import the dataset into a SAS dataset format (from the current xlsx format)



The Proc Import code will be written for you (save this as a template to use for future imports!)



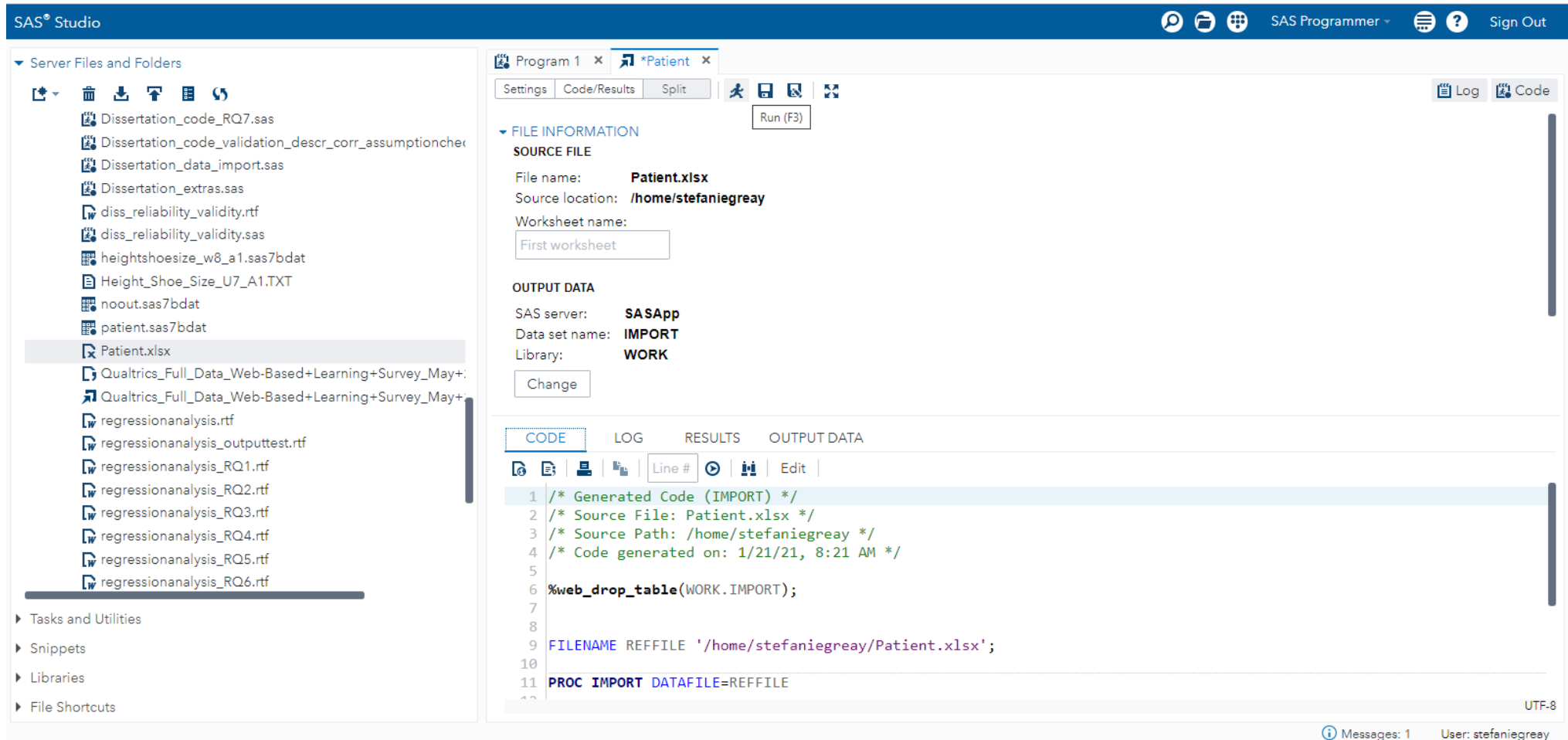
The screenshot shows the SAS Studio interface. On the left is the 'Server Files and Folders' pane with a list of files, including 'Patient.xlsx'. The main window is titled 'Program 1' and 'Patient'. It has tabs for 'Settings', 'Code/Results', and 'Split'. Below the tabs is the 'FILE INFORMATION' section, which shows the source file 'Patient.xlsx' at location '/home/stefaniegreay' with the first worksheet selected. The 'OUTPUT DATA' section shows the SAS server as 'SASApp', data set name as 'IMPORT', and library as 'WORK'. Below this is the 'CODE' tab, which displays the generated Proc Import code. A note on the right side of the code area states: 'NOTE: you have to run this code for the data to actually import.'

NOTE: you have to run this code for the data to actually import.

```
1 /* Generated Code (IMPORT) */
2 /* Source File: Patient.xlsx */
3 /* Source Path: /home/stefaniegreay */
4 /* Code generated on: 1/21/21, 8:21 AM */
5
6 %web_drop_table(WORK.IMPORT);
7
8
9 FILENAME REFFILE '/home/stefaniegreay/Patient.xlsx';
10
11 PROC IMPORT DATAFILE=REFFILE
```



To run the code, click the icon that looks like a guy running.



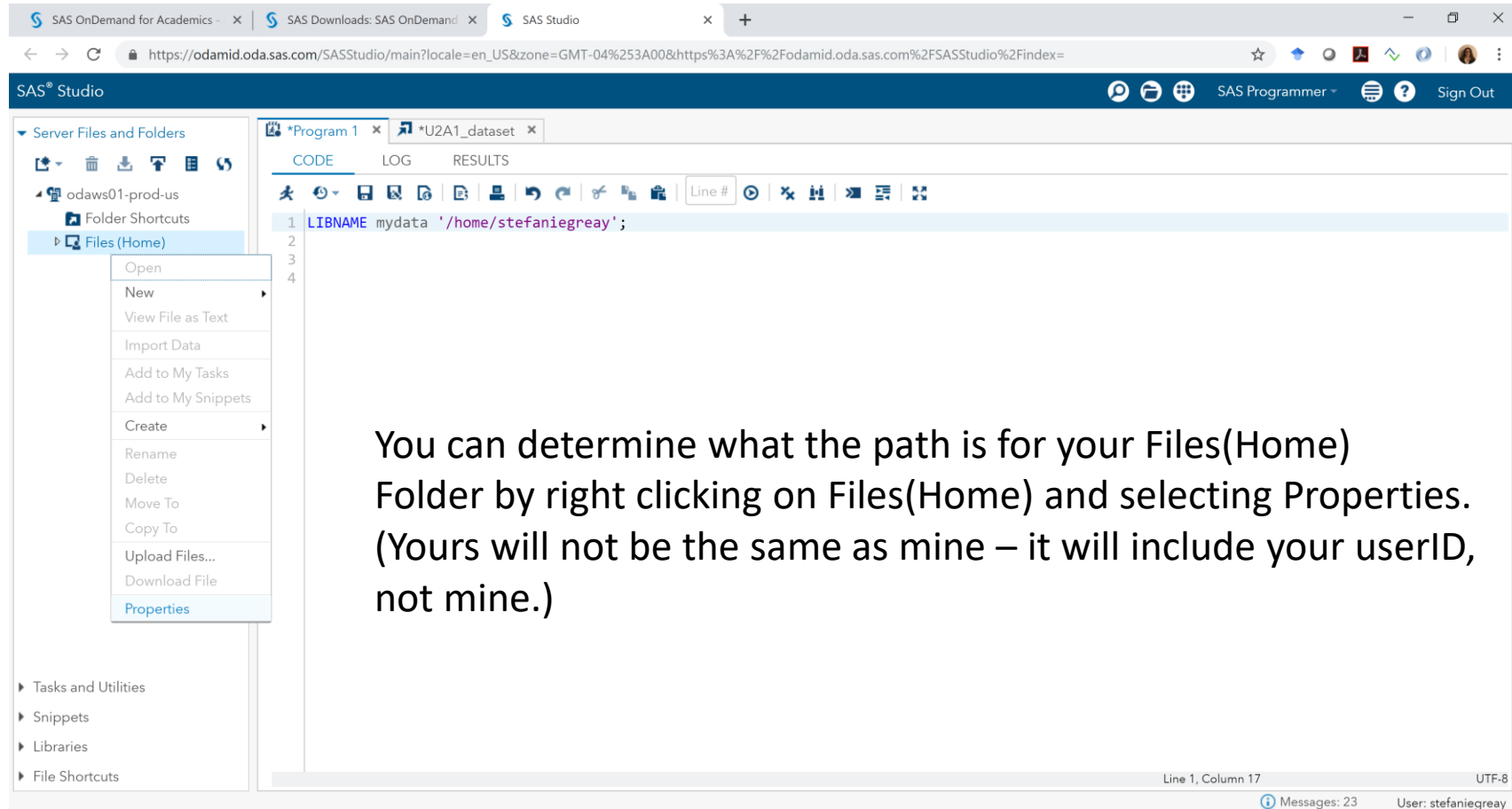
For this particular dataset, because there is a . for missing values, two columns imported as character instead of numeric. We will have to fix this.

The screenshot shows the SAS Studio interface. On the left is the 'Server Files and Folders' pane with a list of files including 'Patient.xlsx'. The main window is titled 'Program 1' and 'Patient'. It has tabs for 'Settings', 'Code/Results', and 'Split'. Below the tabs is the 'FILE INFORMATION' section, which shows 'SOURCE FILE' details: File name: Patient.xlsx, Source location: /home/stefaniegreay, and Worksheet name: First worksheet. Below this is the 'OUTPUT DATA' section, showing 'SAS server: SASApp', 'Data set name: IMPORT', and 'Library: WORK'. At the bottom of the main window is the 'RESULTS' tab, which displays a 'Table of Contents' and an 'Alphabetic List of Variables and Attributes' table.

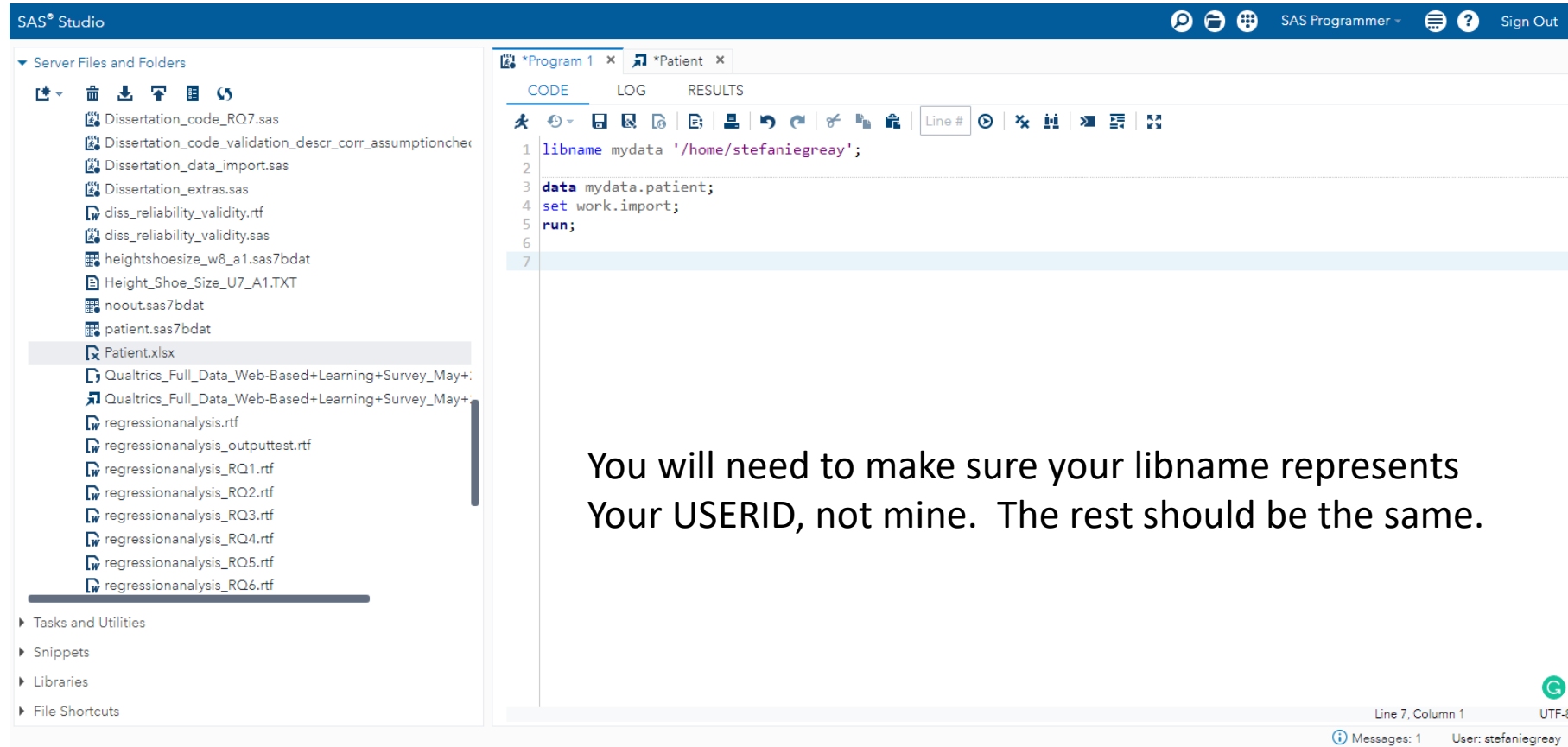
#	Variable	Type	Len	Format	Informat	Label
6	After	Char	3	\$3.	\$3.	After
5	Before	Char	3	\$3.	\$3.	Before
4	Date	Num	8	MMDDYY8.		Date
7	Event	Char	1	\$1.	\$1.	Event
1	FirstName	Char	9	\$9.	\$9.	FirstName
3	Gender	Char	1	\$1.	\$1.	Gender
2	LastName	Char	10	\$10.	\$10.	LastName



To create a SAS Library for your Files(Home) folder, you need to use a libname statement



Save the temporary SAS dataset created by the import to your library using the following sample code.



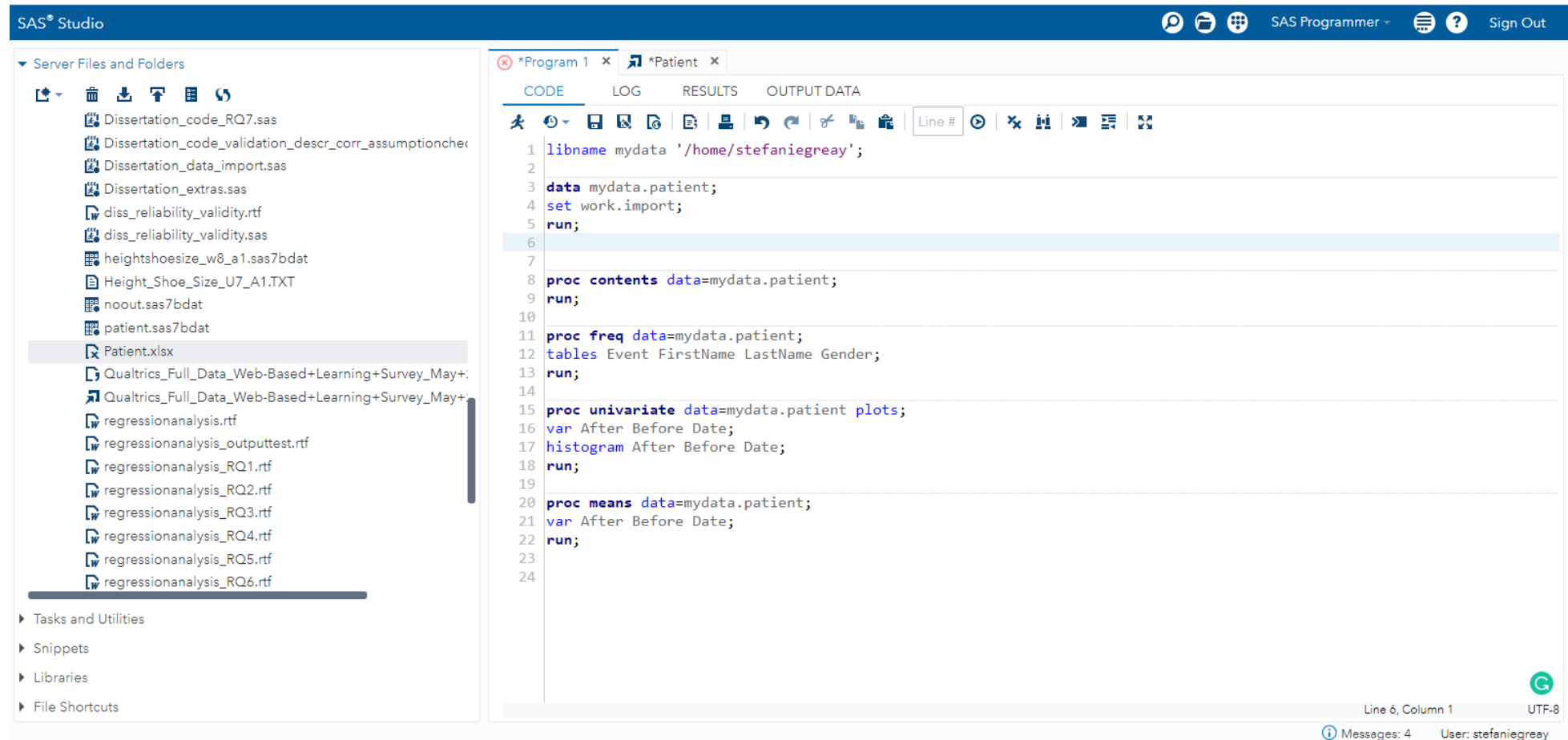
When you run the code, you will see the dataset in the output data window and can verify its success.

The screenshot displays the SAS Studio interface. On the left, the 'Server Files and Folders' pane shows a list of files, with 'Patient.xlsx' selected. The main window is divided into several tabs: 'CODE', 'LOG', 'RESULTS', and 'OUTPUT DATA'. The 'OUTPUT DATA' tab is active, showing a table named 'MYDATA.PATIENT'. The table has 99 rows and 7 columns: 'FirstName', 'LastName', 'Gender', 'Date', 'Before', 'After', and 'Event'. The first 18 rows are visible in the screenshot. Below the table, there is a 'Property Value' section with fields for Label, Name, Length, Type, Format, and Informat.

	FirstName	LastName	Gender	Date	Before	After	Event
1	Nancy	Garcia	F	08/21/15	72	168	0
2	Gail	Davis	F	03/05/15	68	165	1
3	Joan	Jones	F	04/19/15	88	165	0
4	Jim	Brown	M	04/03/15	70	848	1
5	Bob	Williams	1	03/31/15	71	154	0
6	Gail	Brown	F	04/24/15	82	144	0
7	Tom	Garcia	M	05/09/15	73	150	0
8	Thomas	Hernandez	M	06/28/15	70	158	0
9	Saly	Brown	X	05/03/15	81	149	0
10	Tom	Garcia	1	12/08/15	78	144	1
11	Jack	Hernadnez	M	01/03/15	80	156	1
12	Bob	Brown	m	08/16/15	90	145	0
13	Jack	Brown	M	03/31/15	68	163	0
14	Joan	Hernandez	F	07/10/15	83	169	1
15	Jim	Miller	M	10/08/15	83	163	1
16	Gail	Garcia	F	07/31/15	74	157	1
17	Jonathon	Hernadnez	m	08/09/15	.	.	1
18	William	Jones	f	06/19/15	79	160	0



You can now run any procedures against that dataset via the code window.



Sample Code

```
libname mydata '/home/stefaniegreay';
```

```
data mydata.patient;  
set work.import;  
run;
```

```
proc contents data=mydata.patient;  
run;
```

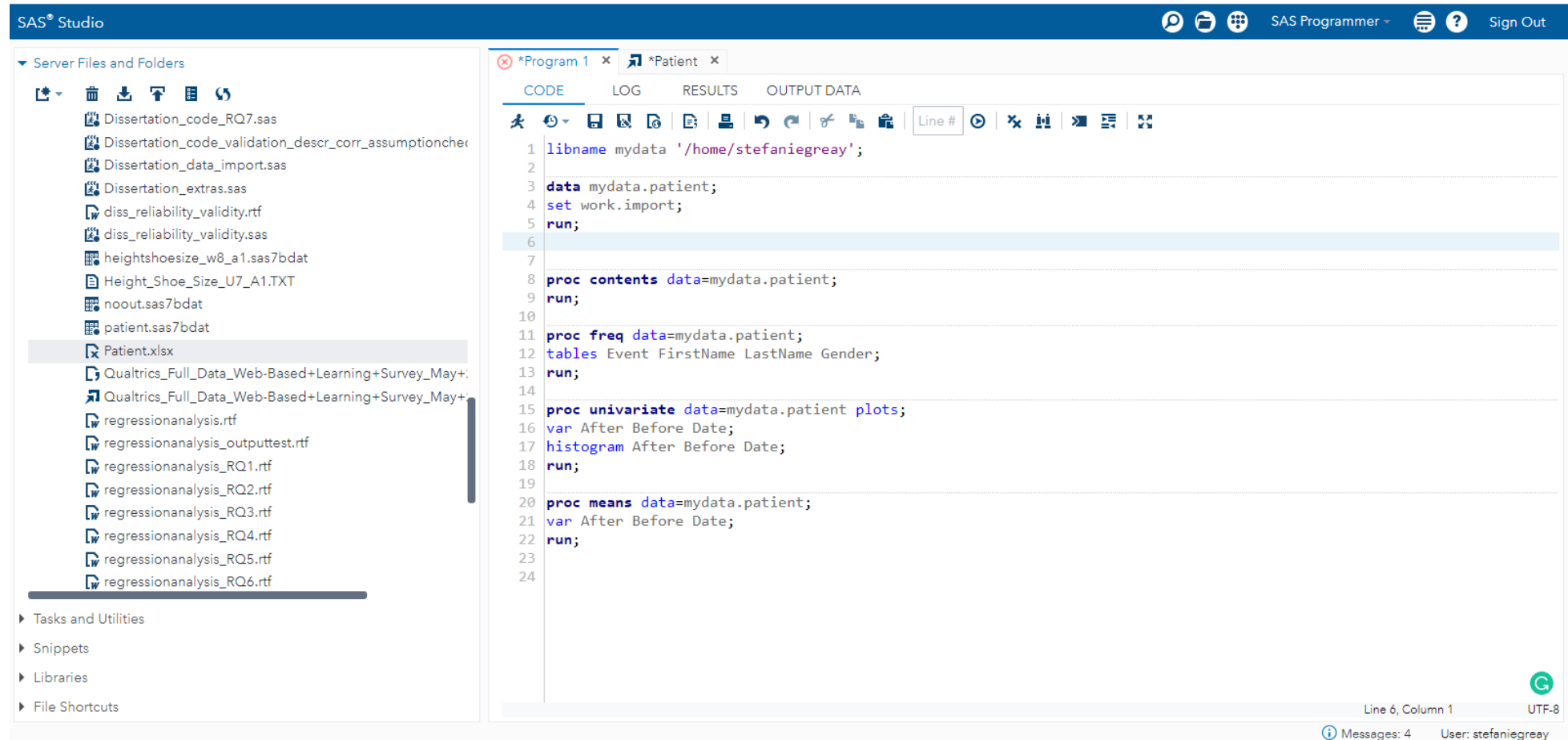
```
proc freq data=mydata.patient;  
tables Event FirstName LastName Gender;  
run;
```

```
proc univariate data=mydata.patient plots;  
var After Before Date;  
histogram After Before Date;  
run;
```

```
proc means data=mydata.patient;  
var After Before Date;  
run;
```



Once you run the code, you will notice a red x next to the program. This means we need to check the logfile for the errors.



Once you run the code, you will notice a red x next to the program. This means we need to check the logfile for the errors. All of the errors are shown in red text.

The screenshot displays the SAS Studio interface. On the left, the 'Server Files and Folders' pane shows a list of files, with 'Patient.xlsx' selected. The main window is divided into two tabs: '*Program 1' and '*Patient'. The '*Program 1' tab is active, showing the 'LOG' pane. The 'LOG' pane has a red 'x' icon next to the 'Errors (6)' section, indicating that there are errors. The errors are listed in red text:

- ERROR: Variable After in list does not match type prescribed for this list.
- ERROR: Variable Before in list does not match type prescribed for this list.
- ERROR: Variable After in list does not match type prescribed for this list.
- ERROR: Variable Before in list does not match type prescribed for this list.
- ERROR: Variable After in list does not match type prescribed for this list.
- ERROR: Variable Before in list does not match type prescribed for this list.

Below the errors, there is a 'Warnings' section with a yellow triangle icon. It shows a 'Block Output Operations' warning with a count of 304. The code in the background is as follows:

```
84  
85     proc univariate data=mydata.patient plots;  
86         var After Before Date;  
87         histogram After Before Date;  
88     run;
```

At the bottom of the log, there is a note: 'NOTE: The SAS System stopped processing this step because of errors.' and another note: 'NOTE: PROCEDURE UNIVARIATE used (Total process time):' followed by a table of resource usage:

real time	0.00 seconds
user cpu time	0.00 seconds
system cpu time	0.00 seconds
memory	505.25k
OS Memory	30120.00k

The bottom status bar shows 'Messages: 5' and 'User: stefaniegreay'.



ERRORS

- The errors here are indicating that our data was imported in the wrong format. (I pointed this out in previous slides/screens, but we haven't fixed it yet. Not fixing it causes this type of error when we go to use those variables that should be numeric, but came in as character.)
- The solution is to convert the data type of those variables to numeric. We can do that using a data step in our code in SAS.
- We can write a new data step or do it in the same data step we already have to make our dataset permanent.



Sample Code

```
data mydata.patient;  
set work.import;  
Aftern = input(After, 3.);  
Beforen = input(Before, 3.);  
run;
```

- We are creating two new variables Aftern and Beforen that will be numeric. We are doing so using the “input” function in SAS with a format of 3.
- We will need to change the name of the variables used in the proc univariate and proc means to use these two new numeric variables.



We can now run our updated program and review the results.

The screenshot displays the SAS Studio interface. On the left, the 'Server Files and Folders' pane shows a list of files, with 'Patient.xlsx' selected. The main window is divided into tabs: 'CODE', 'LOG', 'RESULTS', and 'OUTPUT DATA'. The 'RESULTS' tab is active, showing a 'Table of Contents' section. Below this, the 'The CONTENTS Procedure' results are displayed in a table format.

The CONTENTS Procedure			
Data Set Name	MYDATA.PATIENT	Observations	99
Member Type	DATA	Variables	9
Engine	V9	Indexes	0
Created	01/21/2021 08:49:39	Observation Length	56
Last Modified	01/21/2021 08:49:39	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	131072
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	2334
Obs in First Data Page	99
Number of Data Set Repairs	0
Filename	/home/stefaniegreay/patient.sas7bdat
Release Created	9.0401M6
Host Created	Linux
Inode Number	15054872
Access Permission	rw-r--r--
Owner Name	stefaniegreay
File Size	250KB
File Size (bytes)	262144

At the bottom right of the interface, it shows 'Messages: 8' and 'User: stefaniegreay'.



Sample Code

```
libname mydata '/home/stefaniegreay';
```

```
data mydata.patient;  
set work.import;  
Aftern = input(After, 3.);  
Beforen = input(Before, 3.);  
run;
```

```
proc contents data=mydata.patient;  
run;
```

```
proc freq data=mydata.patient;  
tables Event FirstName LastName Gender;  
run;
```

```
proc univariate data=mydata.patient plots;  
var Aftern Beforen Date;  
histogram Aftern Beforen Date;  
run;
```

```
proc means data=mydata.patient;  
var Aftern Beforen Date;  
run;
```

