

ANLT5020 – Unit 4

Assignment 1 Tutorial

SAS Studio



Instructions

Following are the assumptions:

- Claim information (that is, MOS, DOS, YOS, DRG, and Charges) is valid (as there is no way to verify otherwise at this moment).
- Assume that names that look or sound the same represent the same person.
- For this assignment, use the First_Name and Last_Name fields to check for data quality issues. You may need to do additional research about how to use the DO loop to fix the data quality issues.

Carry out the following tasks in SAS.

- Use PROC FREQ to check for invalid values for a character value.
- Use DATA Step to check for invalid values.
- Use IF-THEN statement to compute the value of a new variable.
- Debug your code using two types of records or data sets. (You can create these sets by taking a subset from the original data set to test for syntax errors, or you can create test data sets with the specific criteria you have coded for to test your logic.)
- Summarize how to use SAS to standardize character data and identify strategies for data cleansing.
- Explain how debugging techniques may be used as part of the ETL process and to assist in data cleansing activities.
- Use the activities you completed to support your explanations. Attach screenshots of your activity results with your assignment or as addendums to your assignment.

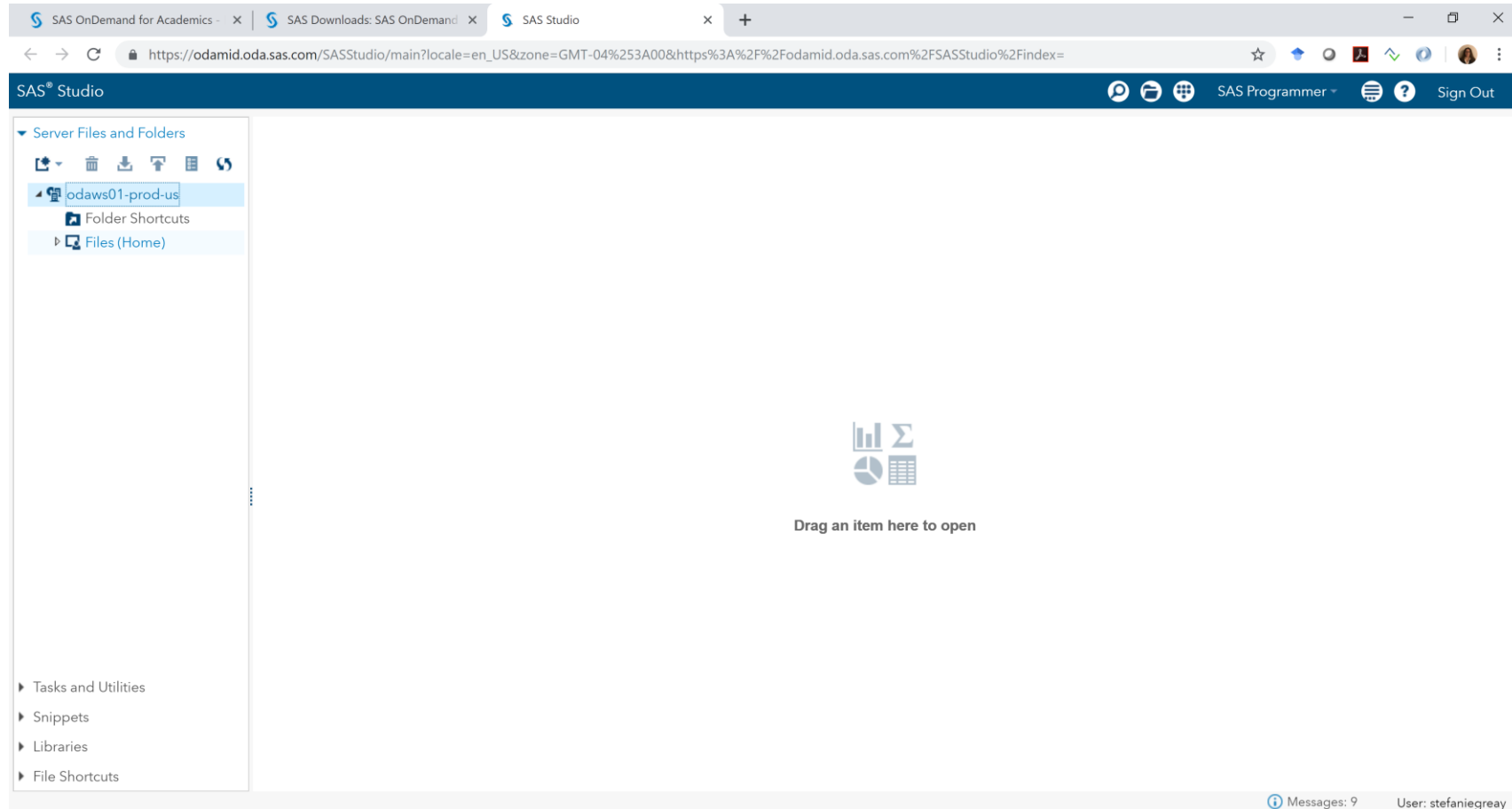


Dataset

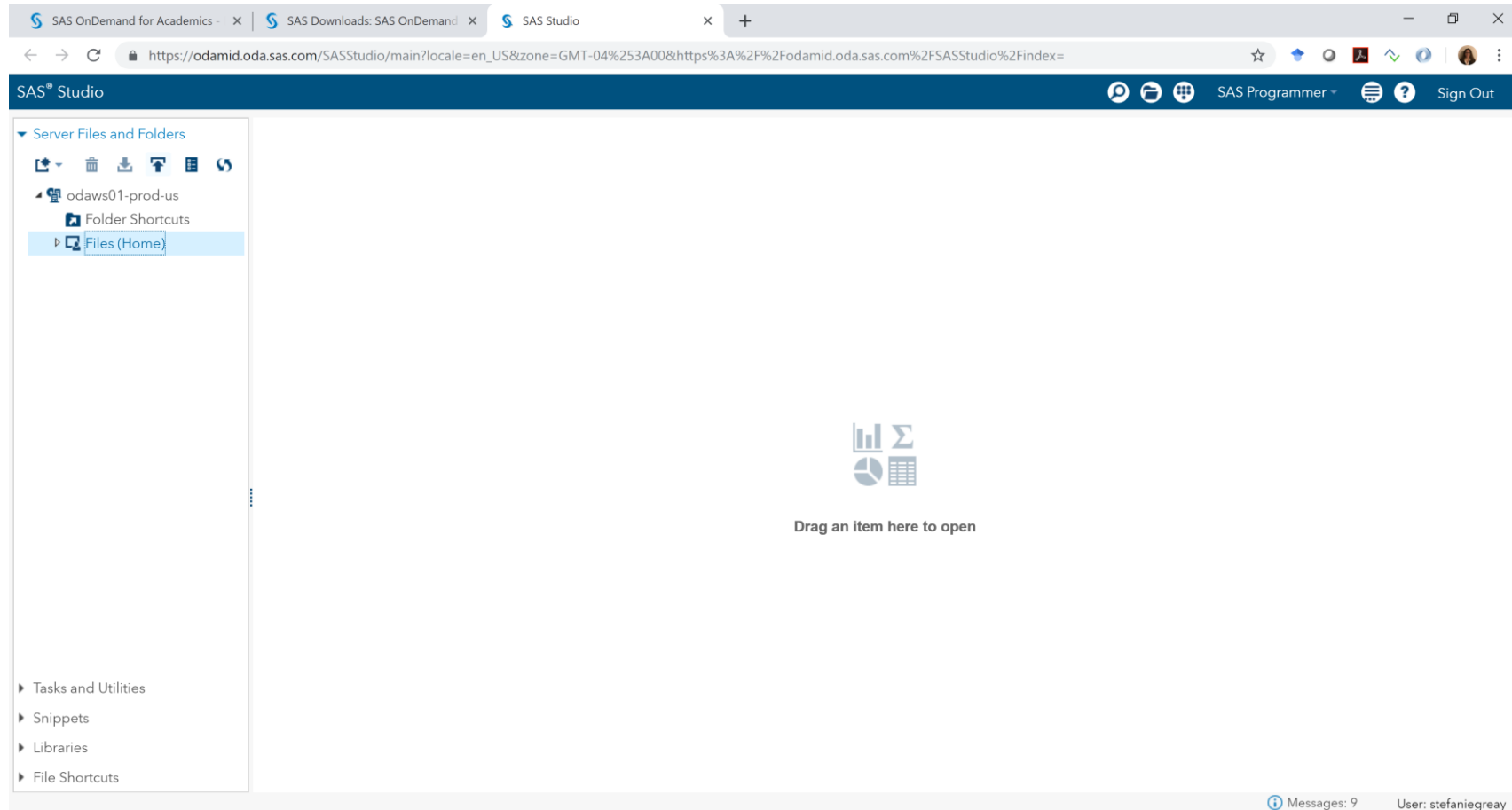
- Download the Claims.csv file from the course datasets zip file or from the Unit 4 Welcome announcement in the course announcements.



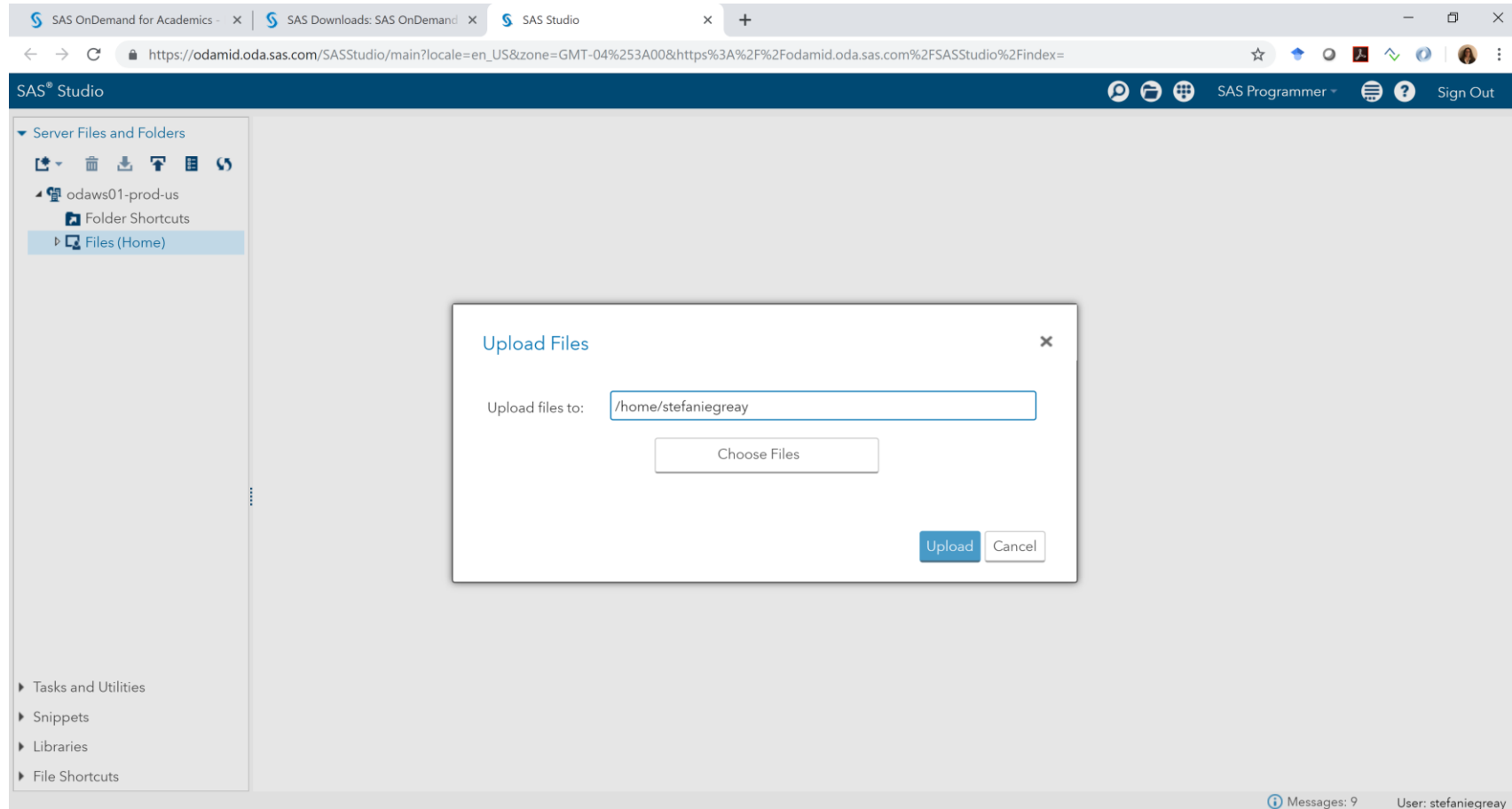
Click on Files(Home)



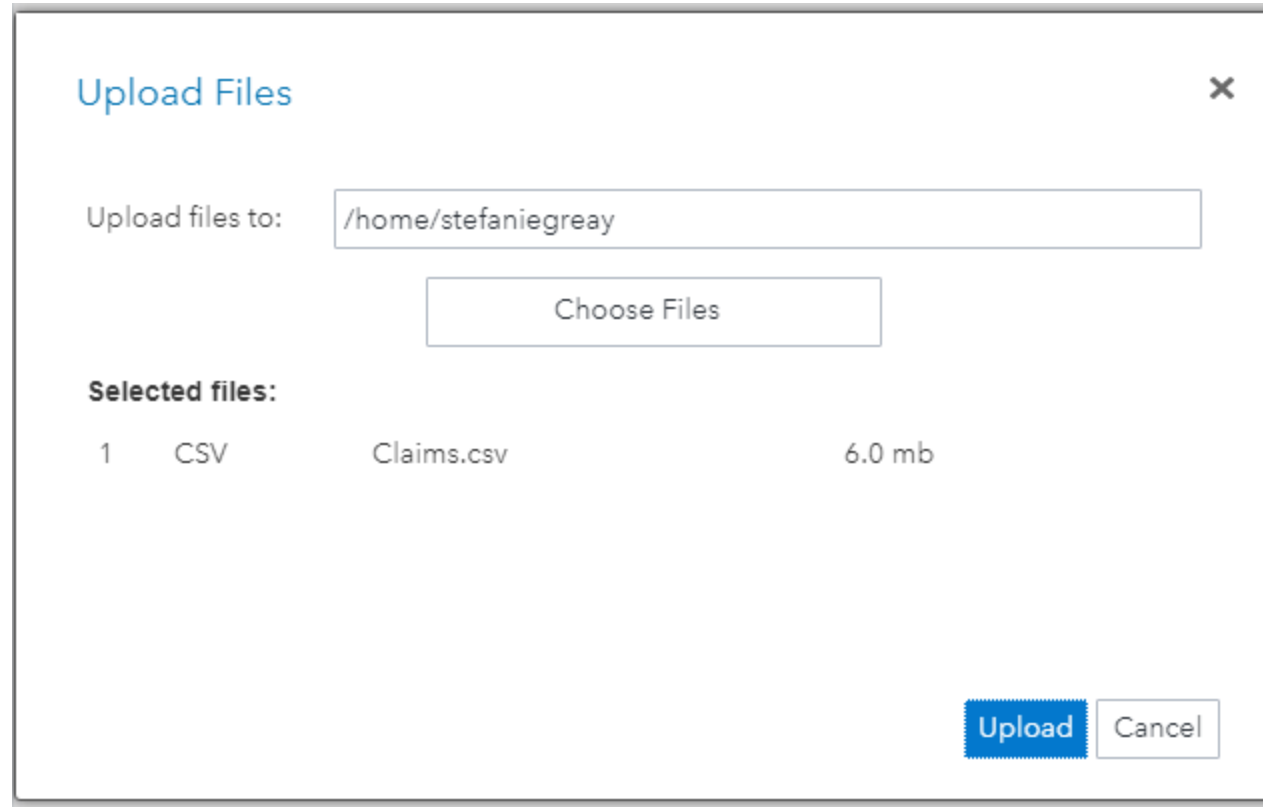
The Upload button will display in dark blue



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.”

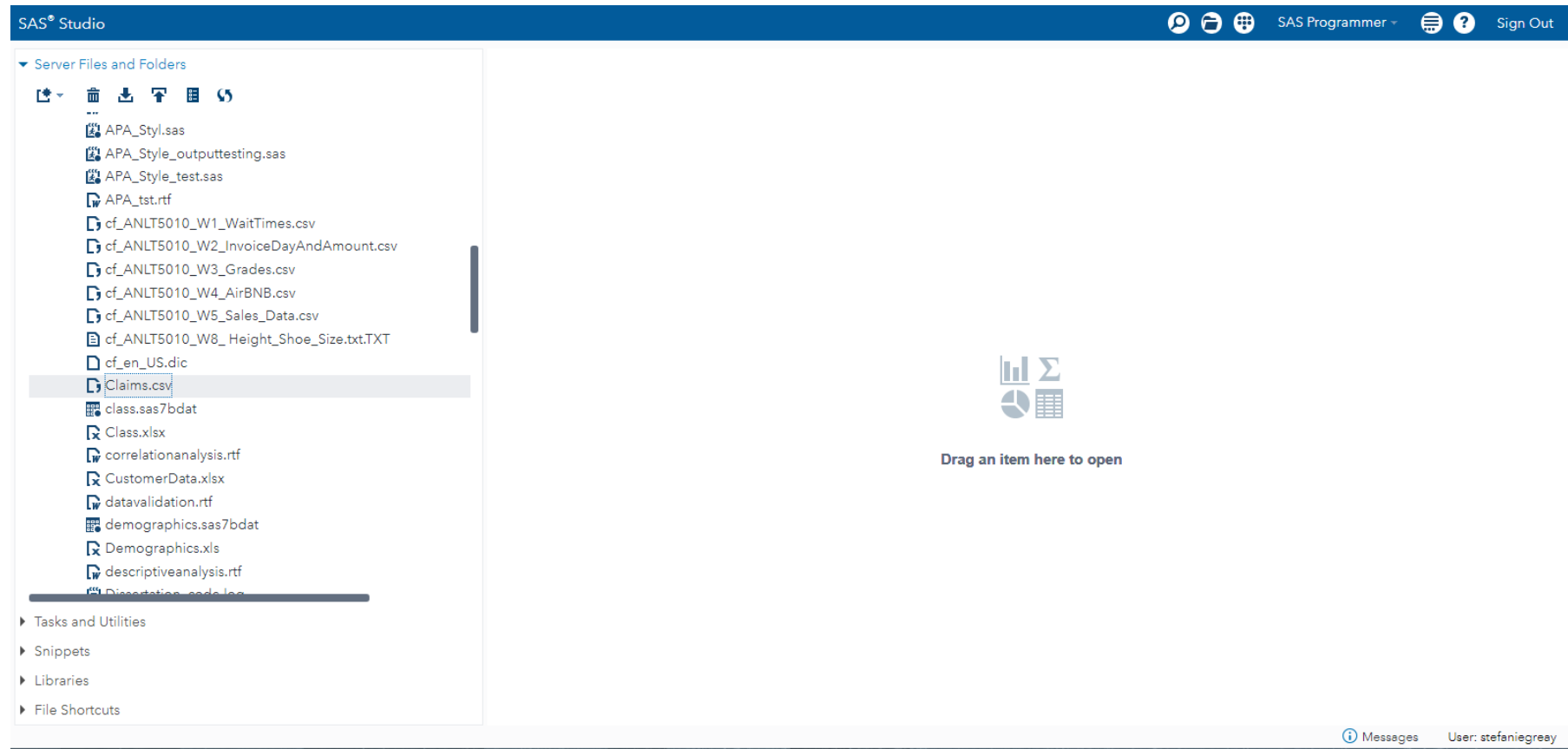


The screenshot shows a web-based 'Upload Files' dialog box. At the top left is the title 'Upload Files' in blue, and at the top right is a close button 'x'. Below the title, there is a label 'Upload files to:' followed by a text input field containing the path '/home/stefaniegreay'. Underneath the input field is a button labeled 'Choose Files'. Below this, there is a section titled 'Selected files:'. Under this title, there is a single file entry: '1 CSV Claims.csv 6.0 mb'. At the bottom right of the dialog box, there are two buttons: a blue 'Upload' button and a white 'Cancel' button.

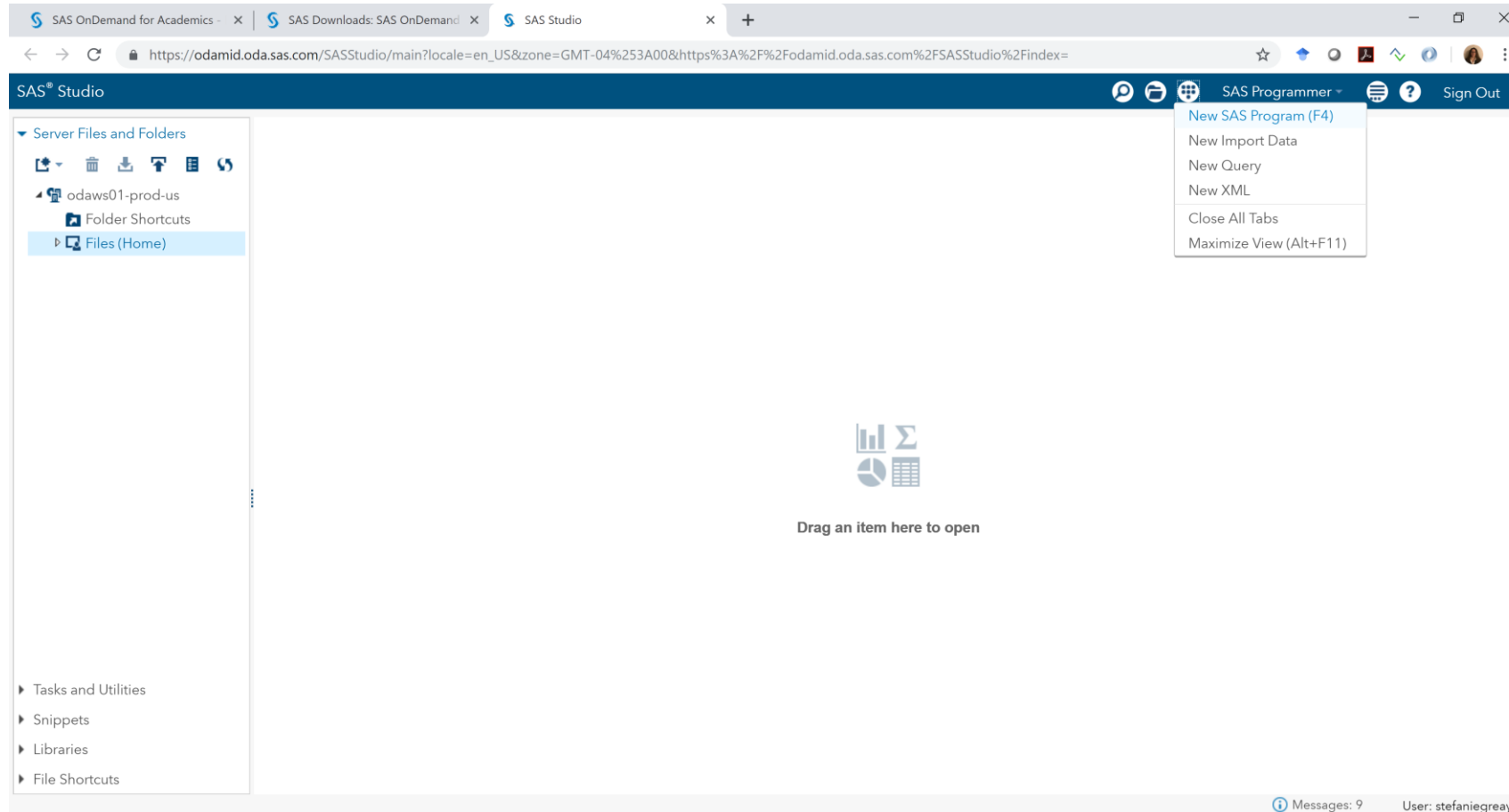
Selected files:			
1	CSV	Claims.csv	6.0 mb



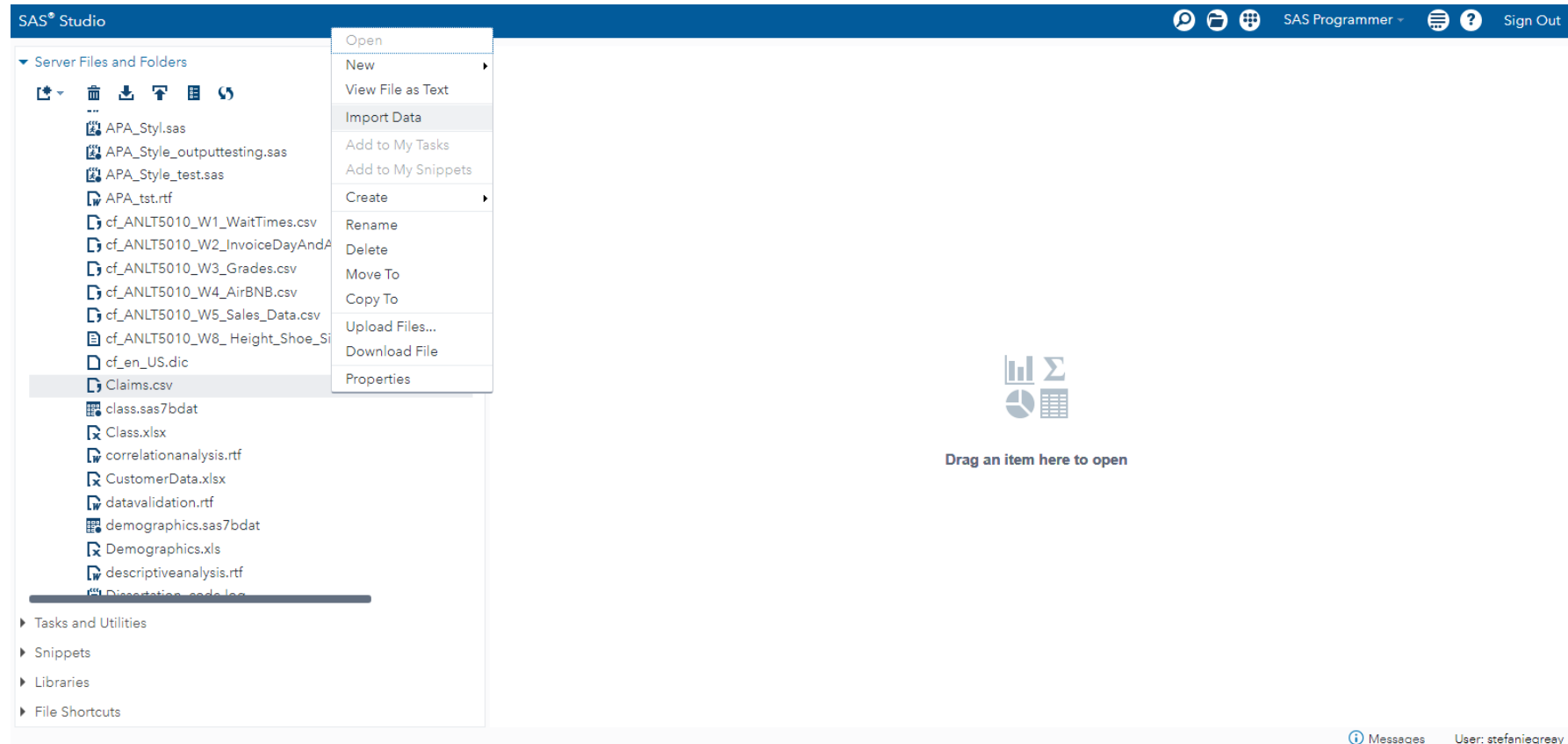
You will be able to view your files by clicking on “Files(Home)” to verify that your file successfully uploaded.



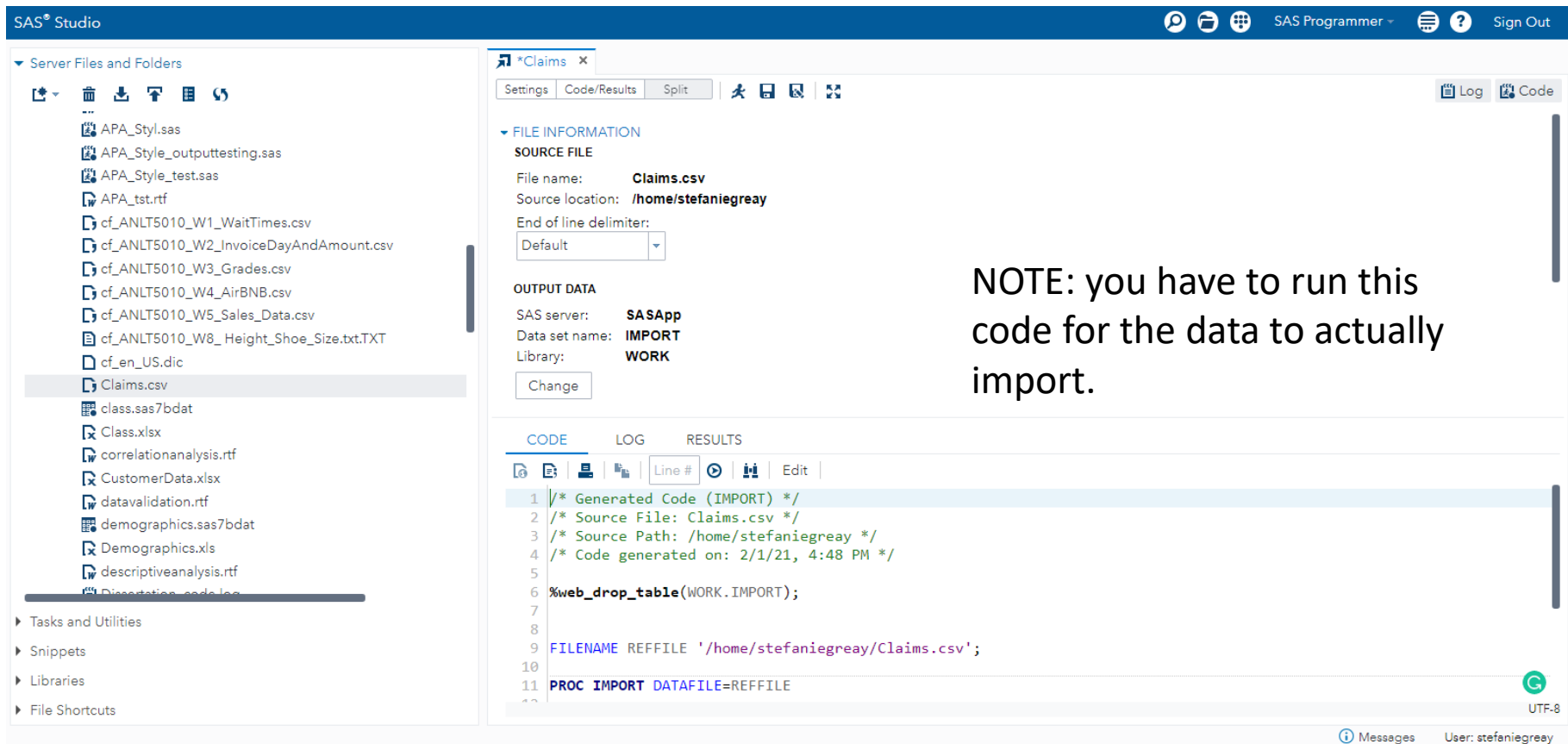
To get started with the SAS portion of the assignment, start a new SAS program.



Import the dataset into a SAS dataset format (from the current csv 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, the 'Server Files and Folders' pane lists various files, with 'Claims.csv' selected. The main window is titled '*Claims' and shows the 'FILE INFORMATION' tab. It displays the source file 'Claims.csv' located at '/home/stefaniegreay'. The 'OUTPUT DATA' section shows the SAS server 'SASApp', data set name 'IMPORT', and library 'WORK'. The 'CODE' tab is active, showing the generated Proc Import code. A note on the right states: 'NOTE: you have to run this code for the data to actually import.'

FILE INFORMATION

SOURCE FILE

File name: **Claims.csv**
Source location: **/home/stefaniegreay**
End of line delimiter: **Default**

OUTPUT DATA

SAS server: **SASApp**
Data set name: **IMPORT**
Library: **WORK**

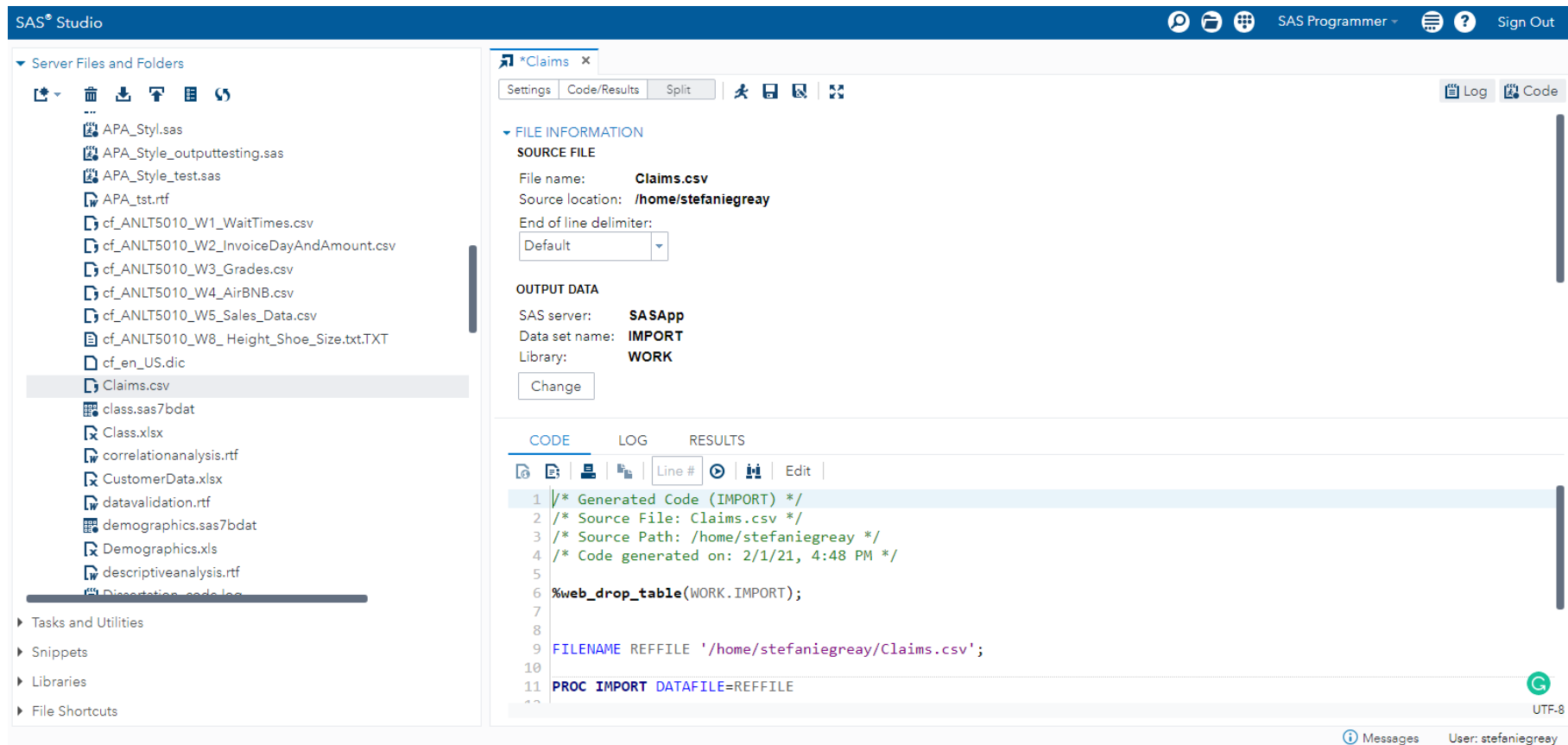
CODE LOG RESULTS

```
1 /* Generated Code (IMPORT) */  
2 /* Source File: Claims.csv */  
3 /* Source Path: /home/stefaniegreay */  
4 /* Code generated on: 2/1/21, 4:48 PM */  
5  
6 %web_drop_table(WORK.IMPORT);  
7  
8  
9 FILENAME REFFILE '/home/stefaniegreay/Claims.csv';  
10  
11 PROC IMPORT DATAFILE=REFFILE
```

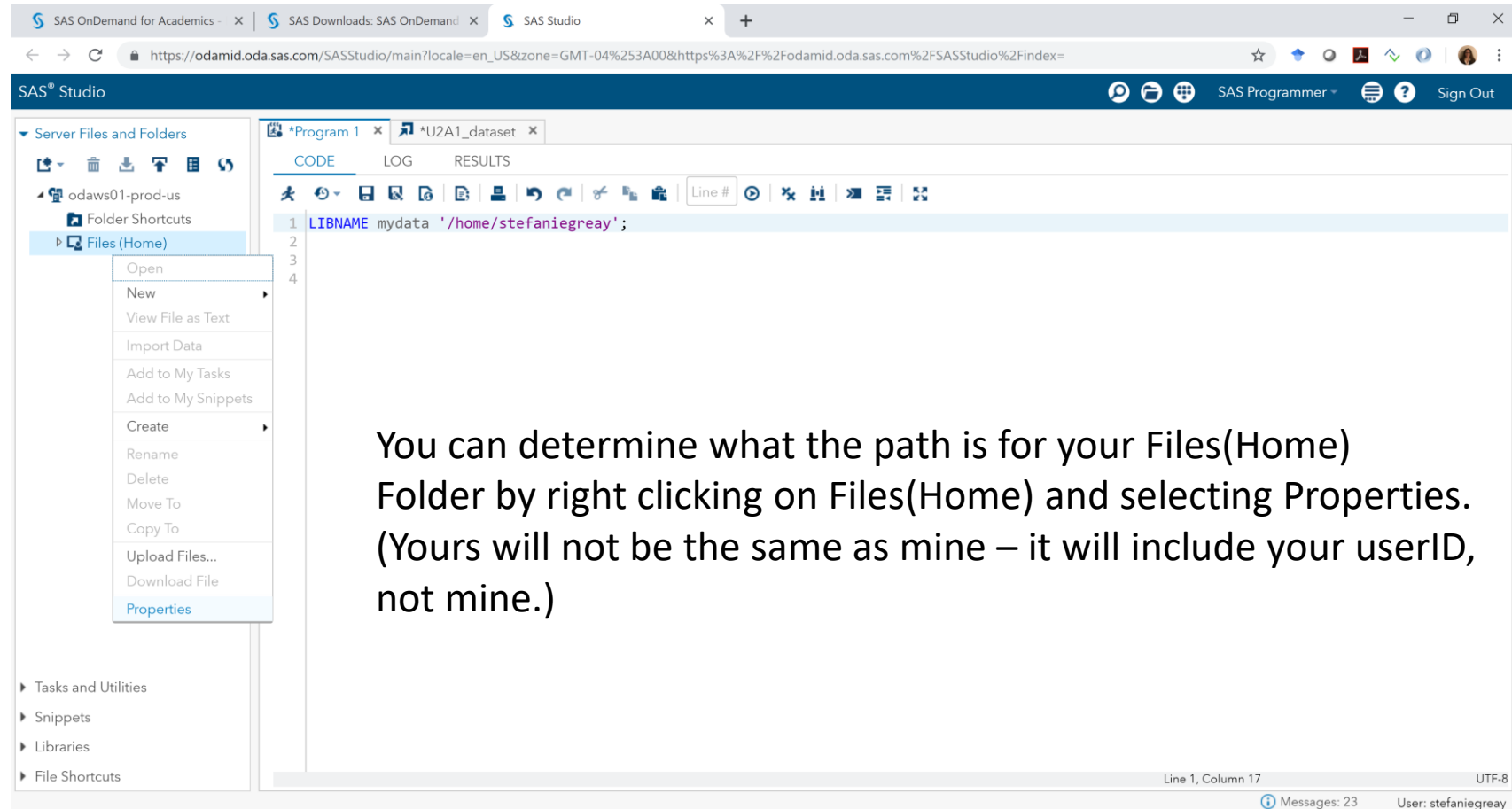
NOTE: you have to run this code for the data to actually import.



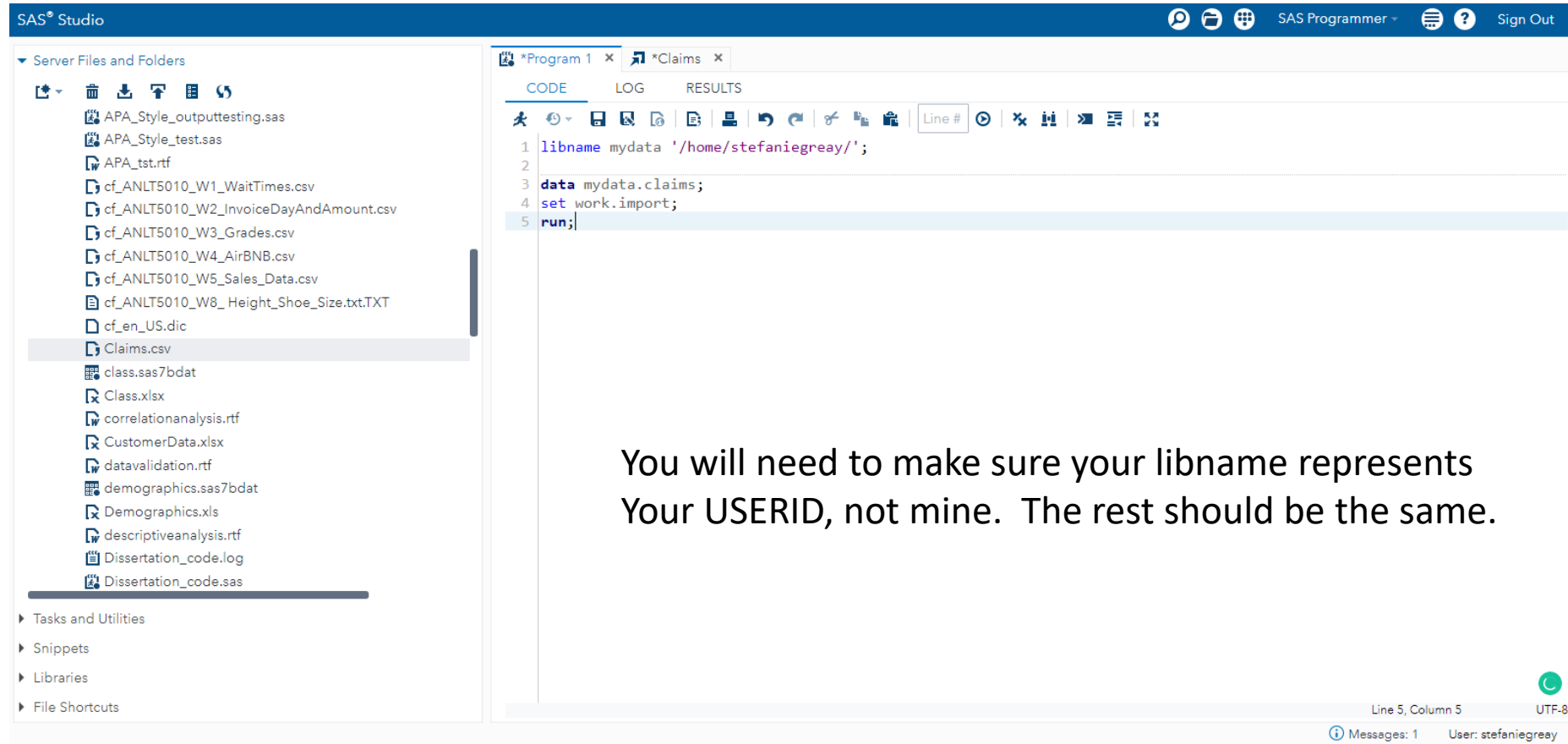
To run the code, click the icon that looks like a guy running.



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.



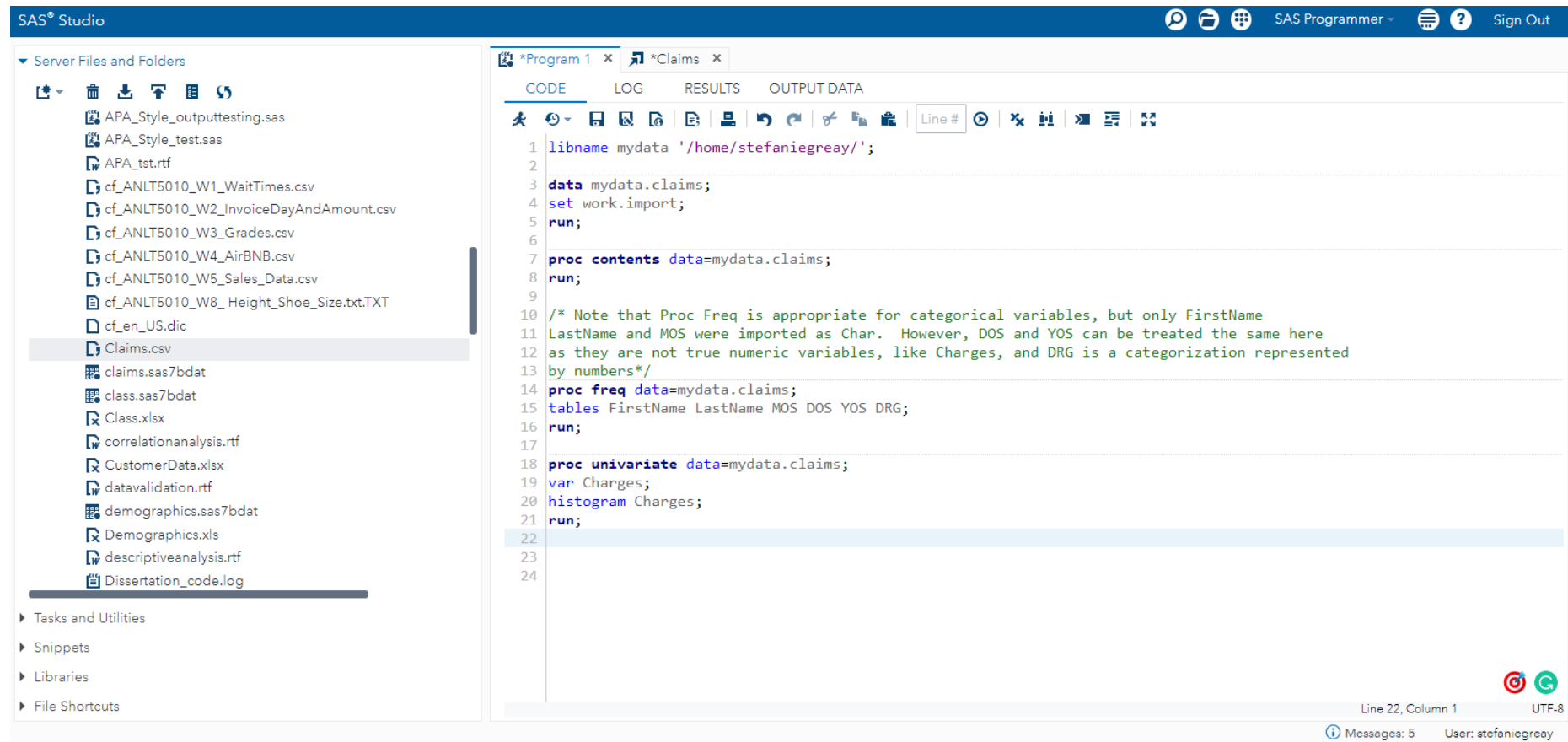
When you run the code, you will see the dataset in the output data window and can verify its success.

The screenshot displays the SAS Studio interface. On the left, the 'Server Files and Folders' pane shows a list of files, with 'Claims.csv' highlighted. The main window is titled '*Program 1' and '*Claims'. The 'OUTPUT DATA' tab is active, showing a table named 'MYDATA.CLAIMS'. The table has 7 columns: FirstName, LastName, MOS, DOS, YOS, DRG, and Charges. The first 18 rows are visible, showing data for various individuals. The bottom status bar indicates 'Messages: 2' and 'User: stefaniegreay'.

	FirstName	LastName	MOS	DOS	YOS	
1	Nancy	Garcia	September	8	2015	
2	Gail	Davis	October	21	2015	
3	Joan	Jones	January	22	2015	
4	Jim	Brown	June	21	2015	
5	Bob	Williams	December	18	2015	
6	Gail	Brown	July	19	2015	
7	Tom	Garcia	January	28	2015	
8	Thomas	Hernandez	June	22	2015	
9	Saly	Brown	January	11	2015	
10	Tom	Garcia	February	28	2015	
11	Jack	Hernadnez	February	9	2015	
12	Bob	Brown	August	26	2015	
13	Thomas	Brown	May	9	2015	
14	Joan	Hernandez	April	12	2015	
15	Jim	Miller	September	14	2015	
16	Gail	Garcia	December	23	2015	
17	Johnathon	Hernadnez	April	18	2015	
18	Sally	Jones	May	18	2015	



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



Sample Code for identifying issues

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

data mydata.claims;
set work.import;
run;

proc contents data=mydata.claims;
run;

/* Note that Proc Freq is appropriate for categorical variables, but only FirstName
LastName and MOS were imported as Char. However, DOS and YOS can be treated the same here
as they are not true numeric variables, like Charges, and DRG is a categorization represented
by numbers*/
proc freq data=mydata.claims;
tables FirstName LastName MOS DOS YOS DRG;
run;

proc univariate data=mydata.claims;
var Charges;
histogram Charges;
run;
```



Once you run the code, you can review the output to identify data issues, like those in the FirstName and LastName variables.

The screenshot shows the SAS Studio interface. On the left is the 'Server Files and Folders' pane with a list of files including 'Claims.csv'. The main window displays the 'RESULTS' tab for a program named '*Claims'. It shows the output of 'The FREQ Procedure' with two tables: one for 'FirstName' and one for 'LastName'. Both tables show frequency, percent, cumulative frequency, and cumulative percent for various names. The 'FirstName' table lists names like Bob, Gail, Jack, Jim, Joan, John, Johnathon, Nancy, Ruth, Sally, Saly, Thomas, and Tom. The 'LastName' table lists names like Brown, Davis, Garcia, Hernandez, Hernandez, Johnson, Jones, Jones, Miller, and Simth. The interface also includes a top navigation bar with 'SAS Programmer' and 'Sign Out' options, and a bottom status bar showing 'Messages: 5' and 'User: stefaniecreay'.

FirstName	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Bob	21758	13.30	21758	13.30
Gail	21834	13.34	43592	26.64
Jack	10964	6.70	54556	33.34
Jim	11027	6.74	65583	40.07
Joan	10823	6.61	76406	46.69
John	10598	6.48	87004	53.16
Johnathon	10977	6.71	97981	59.87
Nancy	10956	6.69	108937	66.57
Ruth	10899	6.66	119836	73.23
Sally	10971	6.70	130807	79.93
Saly	10781	6.59	141588	86.52
Thomas	11098	6.78	152686	93.30
Tom	10988	6.70	163674	100.00

LastName	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Brown	21785	13.31	21785	13.31
Davis	10915	6.67	32700	19.98
Garcia	10934	6.68	43634	26.66
Hernadnez	10969	6.70	54603	33.36
Hernandez	21577	13.18	76180	46.55
Johnson	10896	6.66	87076	53.21
Jone s	10836	6.62	97912	59.83
Jones	10912	6.67	108824	66.50
Miller	10959	6.70	119783	73.19
Simth	11319	6.92	131102	80.11



Sample Code

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

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

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

```
/* Note that Proc Freq is appropriate for  
categorical variables, but only FirstName  
LastName and MOS were imported as Char. However,  
DOS and YOS can be treated the same here  
as they are not true numeric variables, like  
Charges, and DRG is a categorization represented  
by numbers*/
```

```
proc freq data=mydata.claims;  
tables FirstName LastName MOS DOS YOS DRG;  
run;
```

```
proc univariate data=mydata.claims;  
var Charges;  
histogram Charges;  
run;
```

```
data mydata.claimsnew;  
set mydata.claims;  
if FirstName='[ENTER INCORRECT FIRSTNAME HERE]' then  
  FirstName_c='[ENTER CORRECTED FIRSTNAME HERE]';  
else FirstName_c=COMPRESS(FirstName);  
if LastName='[ENTER INCORRECT FIRSTNAME HERE]' then  
  LastName_c='[ENTER CORRECTED FIRSTNAME HERE]';  
else LastName_c=COMPRESS(LastName);  
run;run;
```



Testing Options with Smaller Samples

```
proc surveyselect samsize=100  
data=mydata.claims  
out=mydata.claimssamp;  
run;
```

```
options obs=100;
```



Options for if-then logic

```
data mydata.claimsnew;  
set mydata.claims;  
if LastName='Hernadnez' then LastName_c='Hernandez';  
else Lastname_c=COMPRESS(LastName);  
run;
```

```
data mydata.claimsnew2;  
set mydata.claims;  
if LastName='Simth' then do;  
  LastName_c='Smith';  
end;  
else Lastname_c=COMPRESS(LastName);  
run;
```

```
data mydata.claimsnew3;  
set mydata.claims;  
if LastName='Hernadnez' then LastName_c='Hernandez';  
else if LastName='Simth' then LastName_c='Smith';  
else Lastname_c=COMPRESS(LastName);  
run;
```

