Importing your data into SAS

1. The first step to get your data into SAS is to upload it to your cloud folder. Click on the folder Files (home), and then click the upload button (in blue) to upload your CSV file with your data. I have named my dataset SampleDatasetEdited. Once you have uploaded it, you should see it towards the left.

Graphical user interface, text, application

Description automatically generated

1. It is then time to begin programming. You want to start by importing your dataset from the cloud folder into the program itself. In order to do that you need to write some code but it is essentially copy/pate. Go to the right part of the screen (in the black box), and click in the program next to the line which is numbered 1

Graphical user interface, text, application

Description automatically generated

1. Copy/Paste the following info into that line:

*PROC IMPORT OUT= AnxietyPainDataset*

*DATAFILE= ""*

*DBMS=CSV*

*REPLACE;*

*RUN;*

1. Between the quotation marks in the second line, you need to insert the path to the file in the cloud. In order to do that, you have to find that path. To do that, you go to the left area of your screen and right click your data file as seen below and then click “properties”:

A screenshot of a computer

Description automatically generated

You then copy the text next to location in the “file properties” window as seen here:

Graphical user interface, application, Word

Description automatically generated

Once you copy that text, paste it between the quotes in the second line of the code. This is similar to what it should look like for you, though the path will be different if you are using a different account:

Text

Description automatically generated

1. Only other thing that you may change here is the name of your dataset. So here we are looking at anxiety and pain so I named it “AnxietyPainDataset” in the first line. You can change it to whatever you want, but it should be 1 word and should have no special characters like commas or slashes

Editing a Dataset: Why & How

Once you have your dataset in the program, you want to make sure your dataset is ready for analysis. To do this, you may want to edit your dataset and create new variables that may be a summary or combination of other variables. Think of it as advanced excel in terms of making a column that is a combination of the other columns. Here are some reasons why you want to edit your data:

1. **Put a certain explanatory variable into groups**: In this dataset, the variable age is a continuous variable. If you kept age as a continuous variable in a regression, the odds ratio for that variable would not be too large because it is unlikely that much changes when you go from 29 to 30 years old when predicting something. Instead, if you classify age into groups, and compare lets say those >60 years old to those <30 years old that may be more meaningful. In order to do that, you need to create a new variable (lets call it AgeGrp) and assign each of the patients to a value with that new variable
2. **Turn a categorical variable into a numerical variable for linear regression**: In a linear regression, the inputs must be numbers and they cannot be words. So let’s say we wanted to include gender in our linear regression, we can’t keep it as Male/Female. We must make it a 0/1 variable that would give us the same information when in the regression (the slope on that variable would tell us the risk factor of being a “1” for that gender. If we are classifying “1” as female, it would tell us if there is a positive or negative association with female gender for that variable.
3. **Classify an outcome variable**: Sometimes you want to make a discrete variable out of a continuous variable for your response variable. Here is an example: let’s say you have the KL grades of a bunch of knees with OA. And the goal of your study is to predict severe OA which you are defining as a KL grade of 3 or 4. You would need to create a new variable for SevereOA and assign each patient either a 1 if they have severe OA or a 0 if they do not. This 0-1 variable system is quite common as a logistic regression which predicts between 0 and 1 requires a response variable to either be 0 or 1.
4. **Create a new variable that is a combo of the other variables (e.g. BMI, CCI)**: You can think of this as an excel function. Let’s say you have someone’s height and weight and want to calculate BMI. You can do that here with code. You can also create a variable like Charleson Comorbidity Index.

So returning to our dataset. Let’s say we want to classify our age variable like I was mentioning above. We also want to classify our outcome variable – pain – into a binary variable SeverePain. And we also want to change our anxiety variable from a categorical to a binary numerical one. We do all this with if-then statements which are as easy as they sound. Simply say if “something” then “something.” Look below:

Text, application

Description automatically generated with medium confidence

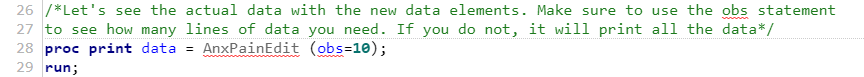
* First 2 lines (lines 15-16): look back at the first document on starting with SAS to understand this syntax. Essentially we are creating a new dataset in the program called *AnxPainEdit* which will be a manipulation of the original dataset *AnxietyPainDataset*.
* Next 3 lines (lines 17-20): we are defining that new variable AgeGrp and setting it as 0 for every single patient initially. We then say if a patient has an age >30, then they their value for AgeGrp is 1, and if their age >60 then their AgeGrp is 2. This makes it so anyone who has an age<30 has an AgeGrp of 0, anyone who has an age between 30 and 60 has an AgeGrp of 1, and anyone who has an age >60 has and AgeGrp of 2. It is important to note that the order matters here. If we swapped lines 18 and 19, anyone older than 30 would have an AgeGrp of 1 even if they are older than 60 because those older than 60 are also older than 30
* Next 2 lines (lines 20-21): essentially doing the same thing as above for our outcome variable pain. We are defining a new variable SeverePain and if patients have a pain score over 5, we are saying that they have severepain and assigning that variable as 1 for them
* Next 2 lines (lines 22-23): also essentially doing the same thing with the difference being here that you are changing a categorical variable into a numeric one. We are defining a new variable AnxTag as a numerical variable which has a value of 0 if they do not have anxiety and a value of 1 if they do.

After we do these things, it is useful to see if the new dataset has been made how you want it to be made. To do that we can see the new dataset in real time. To do this, we use the proc print command. You can copy paste the following at the end of your program.

proc print data = INSERTDATASETNAME (obs=10);  
run;

* Things to note here: the dataset that you created in the last dataset (so for me that was AnxPain Edit) is the one that you include where I have written “INSERTDATASETNAME”
* Obs is how many patients you want to see

This is what it looks like in the actual SAS ecosystem:



And to run your code and see this happen you click the picture of the person running in the top right of the screen. You then click on results.

This is what you should see:

Table

Description automatically generated

And if you see this, you can see if the data is doing what you want it to do with the new variables. And it seems to be working. For example:

* The fist patient has an age of 34 and is in AgeGrp 1. He has a pain score of 10 so has a SeverePain of 1. And he has anxiety so has an AnxTag of 1.