Project 4 - Macros and Opiods

# Overview

At this stage in the game, we should be comfortable with PROC IMPORT, EXPORT, SORT, MEANS, basic DATA step, SGPLOT and some SQL. This week we’ll dig in and make sure we are comfortable with these core procedures of SAS. We’ll also follow our framework Stage, Structure, Cleanse, Transform and Explore as we analyze our opioid data. Finally, we’ll dig a little deeper into PROC MEANS and FREQ and learn how we can automate common analysis with SAS macro.

# TASKS

## TASK 1 – STAGE

Import the following CSV files into SAS using PROC IMPORT – yes just like PROJECT 3!

* NC\_Physician.csv
* NC\_Physician\_Education.csv
* NC\_Physician\_Prescriptions.csv
* NC\_Zip.csv
* OPIOD\_LIST.CSV

## TASK 2 – STRUCTURE

### Task 2a – Physician Master

Create a PHYSICIAN\_MASTER table and create a couple of new variables. To do this you’ll want to join Physician to Physician Education on NPI and to NC Zip on zip\_code;

1. Take all of the fields from NC\_PHYSICIAN and INNER JOIN this to NC\_Physician\_Education on NPI number, don’t worry about the field overlap.
2. join to NC\_ZIP buy you’ll notice that NC\_SIP has 5 digit Zipcodes while NC\_physician has both 5 and 9 digit Zips. You’ll need to resolve this in order to join correctly! To do this you need to use PUT to convert numbers to character, in this case 9. Or a 9 character, character field. You’ll use LEFT to left justify the zip characters. Finally, you’ll use substring to chop the first 5 digits of a zip code. LIKE THIS:

SUBSTR(left(PUT(zip\_code,9.)),1,5);

* Also, note you must join by LIKE Data types so IF NC\_ZIP’s zipcode column is numeric guess what we’ll need to convert that too. What does that? PUT converts numeric to character! You tell it a format to PUT it to. In this case 5.
  + put(B.ZIP\_CODE,5.);

1. Next create a couple of useful variables.

|  |  |  |
| --- | --- | --- |
| ***New Fields*** | ***Formula*** |  |
| Years\_in\_Practice | 2018 – a.Graduation\_Year | Num |
| Full\_Name | compbl(a.first\_name ||' '|| a.middle\_initial ||' '|| a.last\_name) | Char |

* Years\_In\_Practice like this:

1. - Graduation\_Year as years\_in\_practice,

Full name like this, COMPBL function removes multiple blanks in a character **string** by translating each occurrence of two or more consecutive blanks into a single blank.

* compbl(first\_name ||' '|| middle\_initial ||' '|| last\_name) as full\_name

My advice is to break it up into two or more steps (unless you are comfortable with SQL then just do it in one). All we are doing is joining three tables together.

### TASK 2b – Top Opioid / non-opiod Prescriptions

#### PART 1. Now you are going to create a N\_SCRIPTS table.

Your final N\_SCRIPTS table will contain the Sum of Total Day Supply for the top 5 opioids and top 5 non-opioids. Hint: WHERE DRUG NAME IN ()

The list of opiate DRUG\_NAMES:

* HYDROCODONE-ACETAMINOPHEN
* TRAMADOL HCL
* OXYCODONE-ACETAMINOPHEN (AKA Percocet)
* OXYCODONE HCL
* FENTANYL

The list of non-opiate DRUG\_NAMES:

* LEVOTHYROXINE SODIUM
* POTASSIUM CHLORIDE
* METFORMIN HCL
* DILTIAZEM HCL
* GABAPENTIN

When we are done the Structure of your **n\_scripts** table should look like this.

|  |  |  |
| --- | --- | --- |
| Column | Desc | Type |
| **NPI** | national Provider ID – this is what we’ll join to | Num |
| Drug\_Name | Drug Name | Num |
| SUM\_TOT\_DAY\_SUPPLY | Sum of Total Day Supply | Num |

#### PART 2. TRANSPOSE your N\_SCRIPTS into TOPN\_SCRIPTS

proc transpose data=N\_SCRIPTS out=TOPN\_SCRIPTS;

var sum\_total\_day\_supply;

by NPI;

id drug\_name;

run;

your result should look something like this

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **npi** | **NAME OF FORMER VARIABLE** | **TRAMADOL\_HCL** | **HYDROCODONE\_ACETAMINOPHEN** | **FENTANYL** | **OXYCODONE\_HCL** | **…** | **..** | **.** GABAPENTIN |
| 1003005679 | sum\_total\_day\_supply | 291 | . | . | . |  |  |  |
| 1003006149 | sum\_total\_day\_supply | . | 102 | . | . |  |  |  |
| 1003007964 | sum\_total\_day\_supply | . | 83 | . | . |  |  |  |
| 1003013756 | sum\_total\_day\_supply | 660 | 472 | . | . |  |  |  |
| 1003023201 | sum\_total\_day\_supply | 487 | 441 | . | . |  |  |  |
| 1003026550 | sum\_total\_day\_supply | 300 | 3619 | 990 | 17807 |  |  |  |
| 1003037359 | sum\_total\_day\_supply | . | 59 | . | . |  |  |  |
| 1003041534 | sum\_total\_day\_supply | 1566 | 1871 | . | 589 |  |  |  |
| 1003042268 | sum\_total\_day\_supply | . | 47 | . | . |  |  |  |
| 1003042409 | sum\_total\_day\_supply | . | 84 | . | . |  |  |  |

Notice all those missing values? That is why I prefer to do CASE WHEN but we’ll skip that this semester – for now.

#### PArt 3 – pull everyhthing together - create NC\_opioid\_analysis

Finally, you’ll INNER join PHYSICAN\_MASTER to TOPN\_SCRIPTS on NPI number – this produces ONE BIG Table table that we’ll use for analysis.

create table NC\_OPIOID\_ANALYSIS

as

select

a.\*, b.\*

FROM PHYSICAN\_MASTER a INNER JOIN TOPN\_SCRIPTS b ON (a.npi=b.npi);

## TASK 3 – Analysis

### Task 3b – Profile Character / Nominal Values

**Profile Character / Nominal Values ~ charProfile**

You are going to create a macro to profile character and nominal fields in the NC\_Opioid\_Analysis table. To do this I’ve provided a sample program which performs frequency analysis, identifying the top 10 and bottom 20ish results. Your job is to finish it off, and turn it into a macro. For example:

%charProfile(inDat= NC\_Physician\_Opioid\_Analysis, charField=Gender);

* inData is an input Data set,
* charField is the character field you want to profile.

#### Step 1 – get your SAS code running

I would start with getting your SAS code running for “Credentials” analysis

Step 1. Use PROC FREQ to get a frequency count of the variable and OUTPUT a \_FREQ table we aren’t necessarily interested in the results just the frequency table, suppress RESULTS with the NOPRINT option!.

PROC FREQ data=nc\_opioid\_analysis **noprint;**

table credentials / **out=freq\_out;**

run;

Step 2. Use PROC SORT the FREQ\_OUT table descending by COUNT

PROC SORT DATA=freq\_out;

By descending count;

Run;

Step 3. Use DATA Step to Fetch first 20 records

DATA TOP\_20;

SET freq\_out;

IF \_N\_ LE 20 THEN OUTPUT;

RUN;

Step 3. Use PROC PRINT to print the top 20 records

proc PRINT DATA=TOP\_20;

RUN;

Step 5. Use PROC SGPLOT to create a bar chart with the top 20 frequencies

Finally plot the frequencies

proc sgplot data=TOP\_20;

hbar **credentials** / missing response=count categoryorder=respdesc;

run;

#### Step 2 - wrapper your code in a macro

Once you have the SAS program running simply wrapper it in a macro, and replace the field you just analyzed with the charField macro variable.

%MACRO charProfile(inDat = , charField=);

< Your SAS Code with macro variables for example: >

PROC FREQ data=&inDat **noprint;**

table &charField / **out=freq\_out;**

run;

PROC SORT…

DATA …

PROC PRINT..

PROC SGPLOT …

%MEND;

You should be able to call your macro like this.

%charProfile(inDat= NC\_Physician\_Opioid\_Analysis, charField=Gender);

#### Step 3 – Frequency Profile the Following

Once it is running, simply profile the following fields – dump the output in your word report

* Gender
* Credentials
* Medical School
* Organization legal name
* Graduation Year
* Provider Type

### Task 3C - Profile Numeric / Continuous Values ~ numProfile

You are going to create a macro to profile the numeric fields in the NC \_Opioid\_Analysis table. To do this use the samples program below to descriptive statistics analysis of numeric variables and generates a plot of the variable. Your job is to finish it off and turn it into a macro. You’ll follow the same recipe as Task 3b – get your SAS code running first then wrapper it in a macro, substitute the data set and the column variable.

#### Step 1 – get your SAS code Running

Step 1. Use PROC MEANS to create a table of Summary Statistics for the input field. Your proc means statement should produce an output file with the following statistics N, NMISS, MEAN, Q1, Q3 MEDIAN, MIN, MAX and STDEV.

PROC MEANS data= WORK.TOPN\_SCRIPTS n nmiss mean median min max;

var HYDROCODONE\_ACETAMINOPHEN;

run;

Step 2. Use PROC SGPLOT to produce a histogram

proc sgplot data=WORK.TOPN\_SCRIPTS;

histogram HYDROCODONE\_ACETAMINOPHEN /;

yaxis grid;

run;

#### Step 2 – Wraper your code in a Macro

Once you have the SAS program running simply wrapper it in a macro, and replace the field you just analyzed with the numField macro variable.

%MACRO numProfile(inDat=, numField=);

PROC MEANS data=&inData n nmiss mean median min max q1 q3 stdev;

Var &numField;

proc sgplot data= =&inData;

histogram &numField /;

yaxis grid;

run;

%MEND;

The call to your macro should look something like this.

%numProfile(inDat= NC\_Physician\_Opioid\_Analysis, numField= Percocet\_day\_count);

Once it is running, simply profile the following 10 variables , and catch the output in your report.

1. Years\_in\_practice
2. Average\_Age\_of\_Beneficiaries
3. 'TRAMADOL\_HCL',
4. 'OXYCODONE\_ACETAMINOPHEN ',
5. 'OXYCODONE HCL',
6. 'FENTANYL'
7. LEVOTHYROXINE\_SODIUM
8. POTASSIUM\_CHLORIDE
9. METFORMIN\_HCL
10. GABAPENTIN

**Rubric**

This one is relatively easy; the goal is to be able to use SQL to structure data and use SAS PROCS to analyze/profile your data and MACROS to automate our code. The main idea is that your analysis is typically an iterative process and once you’ve done it once, you’ll probably find yourself doing it again and again i.e. “write once-ish use many”. Macros are especially useful for automating and doing repeatable discrete tasks, and to pass data from step to step. SQL is the universal language of data, regardless of what programing language there is typically a SQL interface that you can leverage.

**Turn in**

Project\_4\_your-name.docx – all of your profiling analysis both output for character and numeric data…

Project\_4\_code.sas – for your stage, structure and correlation analysis

Project\_4\_charProfile.sas – macro and calls to perform character profile analysis

Project\_4\_numProfile.sas – macro and calls to perform numeric profile analysis

* 20 pts for stage and structure
* 80 pts for your Macro’s and results
  + 40pts %charProfile
  + 40pts %numProfile