Using the R Function scan\_var\_napp() To Debug Your Code Before Sending It to the DAC

# Intro

Your sas, stata, or r code is a text file.

Use this to check if your code uses any ‘not (yet) approved’ variables of your project specific master variable list.

## Background

This function helps you debug your code before sending it to us.

This is step 6 of the DAC workflow described in the link

<https://ucla.app.box.com/v/chis-dir-template/file/1163645005033>

Prior to this, we assume you have followed step 2 where you have:

* 2 cut-subset your dataset to only the approved variables of your MVL
  + Variables marked with X in /Variable List/MasterVariableList.xlsx
  + Save your cut data ‘your\_cut\_data.dta\_sas7bdat\_sav’ into the higher directory /Data/
* Please (re-)visit <https://healthpolicy.ucla.edu/chis/data/confidential-data/Pages/dummy.aspx>
  + Ctrl+f to the section “How can I Prepare the Dummy Data for Analysis?”
  + It has a link to download sas and stata starter scripts by clicking “Instructions and Code for Preparing Dummy Data Files”.

More Details of Step 2 “Dummy Data Prep Instructions” are in the Appendix.

If you successfully followed step 2, then you should expect to pass this step 6 debugging check. That is, if your code does not use any not approved MVL variables, then the scan\_var\_napp() function described below will not complain (as desired).

# Too Long Did Not Read Version

Below is a brief example

# file path to MVL with selected / approved variables  
fp\_mvl = paste0(getwd(),"/Copy of MasterVariableList.xlsx")  
  
# file path to code  
fp\_code\_client = paste0(getwd(),"/your\_ex\_stata.do")  
# fp\_code\_client = '/.../your\_ex\_stata.do'  
  
con = file(fp\_code\_client)  
code\_client = readLines(con) # vector of text  
close(con)  
  
# pass in vector of text 'code\_client'   
# pass in file path to mvl 'fp\_mvl'  
  
scan\_var\_napp(code\_client=code\_client,  
 # lgl\_code\_tolower = TRUE,  
 fp\_mvl=fp\_mvl,  
 sheet="ADULT")

# Detailed Usage

## Get the scan\_var\_napp() function

Download or copy+paste the helper function scan\_var\_napp.R at

<https://github.com/mikejacktzen/chismisc/blob/main/scan_var_napp/scan_var_napp.R>.

source(paste0(getwd(),"/scan\_var\_napp.R"))  
args(scan\_var\_napp)

function (fp\_mvl, sheet = "ADULT", code\_client, lgl\_code\_tolower = TRUE)   
NULL

The input arguments and output return of the function are

#' @param fp\_mvl a character string of the file path of the users approved '.xlsx' master variable list  
#' @param sheet a character string of the "ADULT" "CHILD" or "TEEN" tab in the MVL  
#' @param code\_client a character vector whose elements are each line of the users code. Result of ?readLines()  
#' @param lgl\_code\_tolower a logical TRUE/FALSE whether the variable names in the user's code are lowercase or not  
#'  
#'  
#' @return a 2 column dataframe with column 'var\_notapp' for the not approved variables and the associated column 'lines' for the line number(s) of the user supplied code where the variable was found

Adapt the below examples to your context

## read in code

# file path to client code to inspect  
# program files .do .sas or .r are all text files with lines of code  
  
fp\_code\_client = paste0(getwd(),"/your\_ex\_stata.do")  
# fp\_code\_client = '/.../your\_ex\_stata.do'  
  
con = file(fp\_code\_client)  
code\_client = readLines(con)  
close(con)  
  
head(code\_client)

[1] "fake line1"   
[2] "fake line2"   
[3] "regress ADLTCNT CHLDCNT HH\_SIZE FAMCNT"   
[4] ""   
[5] "/\* this is a comment with mention of var FAMCNT not approved in mvl \*/"

tail(code\_client)

[1] "fake line1"   
[2] "fake line2"   
[3] "regress ADLTCNT CHLDCNT HH\_SIZE FAMCNT"   
[4] ""   
[5] "/\* this is a comment with mention of var FAMCNT not approved in mvl \*/"

str(code\_client)

chr [1:5] "fake line1" "fake line2" ...

Outside of this R sesion, your raw code file probably looks like

writeLines(code\_client)

fake line1  
fake line2  
regress ADLTCNT CHLDCNT HH\_SIZE FAMCNT  
  
/\* this is a comment with mention of var FAMCNT not approved in mvl \*/

## filepath to your MVL

# file path mvl with selected / approved variables  
  
fp\_mvl = paste0(getwd(),"/Copy of MasterVariableList.xlsx")  
  
# toy example MVL  
# 'ADLTCNT' 'CHLDCNT' are approved with "X" in the cell  
# 'HH\_SIZE' 'FAMCNT' not approved

## run the function with defaults

scan\_var\_napp(code\_client=code\_client,  
 # lgl\_code\_tolower = TRUE,  
 fp\_mvl=fp\_mvl,  
 sheet="ADULT")

[1] var\_notapp  
<0 rows> (or 0-length row.names)

With the default argument lgl\_code\_tolower = TRUE , the function did not flag any variables since the example program used upper case variable names.

## toggle to uppercase

Switching it to lgl\_code\_tolower = FALSE we flag

scan\_var\_napp(code\_client=code\_client,  
 lgl\_code\_tolower = FALSE,  
 fp\_mvl=fp\_mvl,  
 sheet="ADULT"  
)

This may flag comments or unrelated partial matches.  
It is up to the user to review and distinguish.

This function expected you used UPPERCASE variable names in your code. If not the case, switch the 'lgl\_code\_tolower' argument to TRUE

var\_notapp lines  
1 HH\_SIZE 3  
2 FAMCNT 3,5

Here, the function flagged HH\_SIZE in line 3, and FAMCNT in lines 3 and 5.

FAMCNT was used in a regress command in line 3.  
However,  
FAMCNT was merely referenced in a comment in line 5.

Above was an example of your MVL for **ADULT** data, where the sheet="ADULT" was the default.

## TEEN tab of MVL

If you requested variables in the **TEEN** dataset, then toggle the sheet="TEEN" option below.

scan\_var\_napp(code\_client=code\_client,  
 lgl\_code\_tolower = FALSE,  
 fp\_mvl=fp\_mvl,  
 sheet="TEEN")

This may flag comments or unrelated partial matches.  
It is up to the user to review and distinguish.

This function expected you used UPPERCASE variable names in your code. If not the case, switch the 'lgl\_code\_tolower' argument to TRUE

var\_notapp lines  
1 ADLTCNT 3  
2 CHLDCNT 3  
3 HH\_SIZE 3  
4 FAMCNT 3,5

no teen variables were approved in this example, but the same code made reference to them

## CHILD tab of MVL

If you requested variables in the **CHILD** dataset, then toggle the sheet="CHILD" option below.

scan\_var\_napp(code\_client=code\_client,  
 lgl\_code\_tolower = FALSE,  
 fp\_mvl=fp\_mvl,  
 sheet="CHILD")

This may flag comments or unrelated partial matches.  
It is up to the user to review and distinguish.

This function expected you used UPPERCASE variable names in your code. If not the case, switch the 'lgl\_code\_tolower' argument to TRUE

var\_notapp lines  
1 ADLTCNT 3  
2 CHLDCNT 3  
3 HH\_SIZE 3  
4 FAMCNT 3,5

no child variables were approved in this example, but the same code made reference to them

# Appendix

## Dummy Data Prep Instructions

Below are the “Dummy Data Prep Instructions” (Step 2), which you can obtain from the links below. As mentioned in the Background Section, prior to this, we assume you have followed step 2 where you have:

* 2 cut-subset your dataset to only the approved variables of your MVL
  + Variables marked with X in /Variable List/MasterVariableList.xlsx
  + Save your cut data ‘your\_cut\_data.dta\_sas7bdat\_sav’ into the higher directory /Data/
* Please (re-)visit <https://healthpolicy.ucla.edu/chis/data/confidential-data/Pages/dummy.aspx>
  + Ctrl+f to the section “How can I Prepare the Dummy Data for Analysis?”
  + It has a link to download sas and stata starter scripts by clicking “Instructions and Code for Preparing Dummy Data Files”.

### How can I Prepare the Dummy Data for Analysis?

Dummy data code can be used by researchers to analyze dummy data that contain approved CHIS variables from the Master Variable List. Researchers can run this code to remove variables that are not approved for use in DAC projects before writing code for dummy data and submitting DAC data requests. The researchers can simply change the file paths and run the code after downloading dummy data files. This code is available in SAS and Stata formats.

<https://healthpolicy.ucla.edu/chis/data/confidential-data/Documents/Dummy_Data_Prep.zip>

### Code to Prepare Dummy Data for Analysis

This code is meant for researchers to utilize dummy data with variables available to them, as indicated by corresponding selected CHIS variables in the master variable list. Researchers can run this code to remove variables not currently available to them before writing code for dummy data and submitting DAC requests. The researchers can simply change the file paths and run the code after downloading dummy data from the CHPR website. This code is available in SAS and Stata formats.

### Steps for SAS

1. Change datafile location to the file path where you have saved dummy data (either single year or pooled years of data)
2. Change file path for MVL Excel table
3. Change dataset name from “Combined” to the dummy dataset name that you have saved in your directory. (Also remove the “Year” variable from the “Keep=” option if you are using single year data and do not have a year variable.)
4. Change file path to output final dataset that is ready for analysis, either in SAS (.sas7bdat) or Stata (.dta)
5. Remove lines of code if needed—if you do not want to condense a combined adult, teen, and child dummy data set, remove the line for the age group that is not part of the dummy data set you are condensing.
   1. i.e. if you only you are analyzing the adult data set separately, keep %xlimp(ADULT); and remove %xlimp(TEEN); and %xlimp(CHILD); This will import only the “Adult” sheet from the MVL Excel table
6. Remove data sets that were not imported in the previous step from the data step.

### Steps for Stata

1. Change file path to folder containing dummy datasets (either individual years or, if working with multiple years, already-combined dummy data).
2. Change file path to import Master Variable List (MVL) with approved variables selected for each age group.
3. Change file path to dummy data file.
   1. Remove “year” in keep statement if working with individual years, or if “year” variable was not created upon merging data

NOTE: After preparing code that runs successfully with dummy data, please ensure that all variables are changed from PUF to CHIS variable names. Please refer to “PUF RECODES” tab in MVL to ensure the variable names are changed from those in the “VARIABLE NAME” column (indicating PUF variable names) to the corresponding names in the “ORIGINAL” column (indicating CHIS variable names).