ACCESSING R WITHIN THE SAS SYSTEM

Alan Mitchell
Cancer Research and Biostatistics
WHY ARE YOU HERE?

• Many statisticians/analysts/programmers are familiar with both SAS and R.

• Some tasks are more easily accomplished in one compared to the other.
  • BY processing with SAS
  • median() function in R

• Pick your favorite features from each package and use what suits you best.

• Most of the examples will be related to visualizing data, but don’t let that limit your creativity when using this feature!
WHAT’S TO COME

- Getting Started
- The Interface
- A Few Examples
- Integration with SAS Macros
- R GUIs
- Final Comments
GETTING STARTED

• What you need

  • SAS/IML 9.22 or Later

  • A supported version of R
    • SAS/IML 9.22 use R 2.9.1-2.11.1
    • SAS/IML 12.1 (Base SAS 9.3) use R 2.12.1 or later

The following examples were run with SAS/IML 12.1 and R version 2.15.1
GETTING STARTED

• Point SAS to the R installation directory

  • Environment Variable
    • R_HOME = C:\Program Files\R 2.15.1\n    • This method is very helpful if you have multiple versions of R installed.

  • InstallPath (Windows Registry)
    • HKEY_LOCAL_MACHINE\SOFTWARE\R-core\R
GETTING STARTED

• To use the interface during a SAS session
  • Start SAS from the command line with the –RLANG option
  • Add –RLANG to SASV9.CFG

• Use PROC OPTIONS OPTION=RLANG to make sure your session supports access to R.
THE INTERFACE

• Four CALL routines perform the data exchange.

  • ExportDataSetToR(SAS Object, R Object)
    call ExportDataSetToR(‘sashelp.class’,’class’);
  • ExportMatrixToR
  • ImportMatrixFromR
  • ImportDataSetFromR

• SUBMIT/ENDSUBMIT statements define sections of code that will be executed in R.

  submit /R;
  <Insert R statements here>
  endsubmit;
proc iml;

x = {1 2 3, 4 5 6, 7 8 9};

call ExportDataSetToR("SASHELP.CLASS", "SASData");
call ExportMatrixToR(x,"xmat");
call ImportDataSetFromR("WORK.OLDFAITHFUL","faithful");
call ImportMatrixFromR(xR,"xmat");

SUBMIT /R;
plot(height~weight,data=SASData);
ENDSUBMIT;
THE INTERFACE

submit / R;
cat("\n SASData has class: ",class(SASData))
cat("\n\n xmat has class: ",class(xmat))
cat('\n\n\n');
endsubmit;

R Output:

SASData has class:  data.frame

xmat has class:  matrix
A SIMPLE PLOT

- submit /R;
- plot(x=SASData$Height, y=SASData$Weight,
  pch=19, cex=2);
- endsubmit;
Saving & Directing Output

%let basedir = H:\;

proc iml;

basedir = translate("&basedir","/","\");

print basedir;

submit basedir /R;
setwd("&basedir")
win.metafile('faithful.emf')
plot(eruptions~waiting,data=faithful)
dev.off();
endsubmit;

quit;
EXAMPLE 1

- Want to compare City vs Highway MPG in sashelp.cars

- Scatterplots can be deceiving since there are so many duplicate pairs (multiple cars with 20 MPG City & 20 MPG Highway)

- We can use color gradient to represent one of the variables

- Think heatmaps
EXAMPLE 1

```
proc iml;

call ExportDataSetToR('sashelp.cars','cars');

submit /R;
names(cars) = tolower(names(cars));
mpg = cars[order(cars$mpg_highway),c('mpg_city','mpg_highway')]

mpg$order = 1:length(mpg$mpg_highway)

mpgdist = ecdf(mpg$mpg_city - min(mpg$mpg_city))
mpg$col = mpgdist(mpg$mpg_city - min(mpg$mpg_city))

mpgcolfun = colorRamp(c('red','yellow','forestgreen'))

cols = sprintf('#%02X%02X%02X',
   floor(mpgcolfun(mpg$col)[,1]),
   floor(mpgcolfun(mpg$col)[,2]),
   floor(mpgcolfun(mpg$col)[,3]))
```
EXAMPLE 1

windows(height=4, width=8)
par(mar=c(1,4,4,1))

plot(0, type='n', ylim=c(0,70), xlim=c(0,429), xaxt='n', xlab='',
     ylab='MPG Highway', main='MPG - Highway vs City')

segments(y0 = 0, y1 = mpg$mpg_highway, x0 = mpg$order, col = cols, lwd=2)

legend('topleft', inset=1/100, bty='n', col=c('red','forestgreen'), lty=1, lwd=2,
       legend=c(sprintf('Min. MPG City (%2i)',min(mpg$mpg_city)),
                sprintf('Max. MPG City (%2i)', max(mpg$mpg_city))))

savePlot('cars.emf', type='emf')

endsubmit;

*quit;
EXAMPLE 1

MPG - Highway vs City

Min. MPG City (10)
Max. MPG City (60)
EXAMPLE 2

- Consider the map by John Snow regarding the 1854 Broad Street cholera outbreak in London.

- Specialized data require specialized tools
  - R library RGoogleMaps

- Given the locations of the nearby pumps and the locations and frequencies of deaths, this map can be recreated using satellite imagery available from Google Maps.

- Many thanks to Robin Wilson for compiling the data.
  - www.rtwilson.com/academic
http://nl.wikibooks.org/wiki/Bestand:Snow-cholera-map.jpg
EXAMPLE 2

data pumps;
input lat long best12.;
datalines;
51.51343153   -0.136462137
51.51396652   -0.13938074
51.51499622   -0.139465561
51.51244425   -0.131423584
51.51222911   -0.133388372
51.51163252   -0.135713397
51.51010934   -0.133755825
51.51138543   -0.137993287
;
run;
**EXAMPLE 2**

```sas
data deaths;
input lat long deaths best12.;
psize = .8 + log(deaths);
datalines;
51.51150816  -0.135724505  3
51.51145184  -0.135677478  2
51.51140748  -0.135647307  1
...  
51.51040124  -0.136268051  1
51.51008875  -0.135917641  1
51.50994598  -0.135556594  1
; run;
```
EXAMPLE 2
proc iml;

call ExportDataSetToR('pumps','pumps');
call ExportDataSetToR('deaths','deaths');

submit /R;

names(pumps) = tolower(names(pumps));
names(deaths) = tolower(names(deaths));

library('RgoogleMaps')
setwd("C:/SnowMap")

mymap = GetMap(center=c(51.513,-0.135),
    zoom =16, destfile = "map1.png",
    format="PNG", maptype = "satellite");
Example 2

```r
png('SnowMap1.png', width=1200, height=1200, pointsize=18)

PlotOnStaticMap(mymap)

PlotOnStaticMap(mymap, FUN=points, add=TRUE, 
                   lat=deaths$lat, 
                   lon=deaths$long, 
                   col='red', pch=19, cex=deaths$psize)

PlotOnStaticMap(mymap, FUN=points, add=TRUE, 
                   lat=pumps$lat, 
                   lon=pumps$long, 
                   col='green', pch=17, cex=2)

dev.off();
endsubmit;
```
Example 2
Not-so-Practical Application
Integration with SAS Macros

%MACRO PLOT(ds=, x=, y= ,Plotparms="");
PROC IML;

CALL EXPORTDATASETTO("&ds","rdata");
ds="&ds";
x="&x";
y="&y";
plotparms = &PlotParms;
print x y;

%INCLUDE "C:\PlotMacro\PlotR.SAS";

%MEND PLOT;
Integration with SAS Macros

SUBMIT ds x y PlotParms /R;

names(rdata) = tolower(names(rdata)) # Convert all column names to lowercase

attach(rdata)

plot(x=&x, y=&y,main=toupper("&ds"),&plotparms)

detach(rdata);

ENDSUBMIT;
Integration with SAS Macros

%PLOT(ds=sashelp.class, x=height, y=weight);
*quit;
R GUIs

DEBUGGING YOUR R CODE

submit /R;
library(Rcmdr);
endsubmit;
R GUILs
FINAL COMMENTS

- R can be a flexible system for the visualization of SAS data
- R package system
- Easy to integrate
- Extend these tools to more complex modeling scenarios
QUESTIONS?

For further questions, feel free to contact me by e-mail.

alanm@crab.org