

DATA HANDLING WITH SASMARKDOWN IN RSTUDIO

Nandukrishnan

Data Analyst I, Genpro Research

nandukrishnan.h@genproresearch.com

Senju Mani

SAS Programmer III, Genpro Research

senju.mani@genproresearch.com

Vikash Jain

Head of Statistical Programming US, Genpro Research

vikash.jain@genproresearch.com

R Markdown

- File format to generate dynamic documents from R.
- Introduced with the knitr package to embed R chunks in Markdown documents.
- `knitr` runs the R code in the chunk and append results of code in the final document.
- Uses `pandoc` to transform files into HTML, PDF or Word.

`SASmarkdown` PACKAGE

- R package with functions to extend the 'knitr' SAS engine.
- Available in CRAN: Current Version - 0.8.0

INSTALLATI ON

Directly installed from CRAN

- Run in console: `install.packages("SASmarkdown")`

HOW TO INCLUDE SAS CODES IN R MARKDOWN

- Included as part of code chunks in R markdown.
- When Knitted, `knitr` writes the code into a file and then submits to SAS.
- SAS runs the file in batch mode and produce log and listing output.
- Listing output is read back into `knitr` and attached in the Markdown document.

Limitations

- Each code chunk runs separately as a SAS batch job. A dataset prepared using a data step cannot be reused in a different chunk unless the chunk option `collectcode` is set to `TRUE`.
- To generate outputs in different formats(listing/HTML/ LaTeX) corresponding SAS engines must be used.
- Knitr returns two blocks of text to your document - by default, the SAS code and the listing output. To include sas log, ``saslog`` engine must be used.

EXAMPLE

SAS chunk:

```
```{sas}  
proc means data=sashelp.class;
run;
```
```

Final output:

```
proc means data=sashelp.class;  
run;
```

The MEANS Procedure

| Variable | N | Mean | Std Dev | Minimum | Maximum |
|----------|----|-------------|------------|------------|-------------|
| Age | 19 | 13.3157895 | 1.4926722 | 11.0000000 | 16.0000000 |
| Height | 19 | 62.3368421 | 5.1270752 | 51.3000000 | 72.0000000 |
| Weight | 19 | 100.0263158 | 22.7739335 | 50.5000000 | 150.0000000 |

SAS Engines

- **sas** - this returns the SAS code and the listing output
- **saslog** - this returns the SAS log instead of your code, and ordinary SAS output
- **sashtml** - this returns SAS code and SAS HTML output
- **sashtmllog** - this returns the SAS log and SAS HTML output
- **sashtml5** - this returns SAS code and SAS HTML output with inline graphics
- **sashtml5log** - this returns the SAS log and SAS HTML output
- **saspdf** - this returns SAS code and SAS LaTeX output with PDF graphics
- **saspdflog** - this returns the SAS log and SAS LaTeX output

ADDING SAS LOG

SAS chunk:

```
```{sas log}  
proc means data=sashelp.class;
run;
```
```

Final output:

```
2      proc means data=sashelp.class;  
3      run;
```

NOTE: There were 19 observations read from the data set SASHELP.CLASS.

NOTE: The PROCEDURE MEANS printed page 1.

NOTE: PROCEDURE MEANS used (Total process time):

real time 0.05 seconds

cpu time 0.04 seconds

The MEANS Procedure

| Variable | N | Mean | Std Dev | Minimum | Maximum |
|----------|----|-------------|------------|------------|-------------|
| Age | 19 | 13.3157895 | 1.4926722 | 11.0000000 | 16.0000000 |
| Height | 19 | 62.3368421 | 5.1270752 | 51.3000000 | 72.0000000 |
| Weight | 19 | 100.0263158 | 22.7739335 | 50.5000000 | 150.0000000 |

ADDING SAS HTML OUTPUT

SAS chunk:

```
```{sashtml}  
proc sql;
create table cars1 as
select make, model, type, invoice, horsepower, length, weight
from
sashelp.cars
where make in ('Audi','BMW')
;
run;

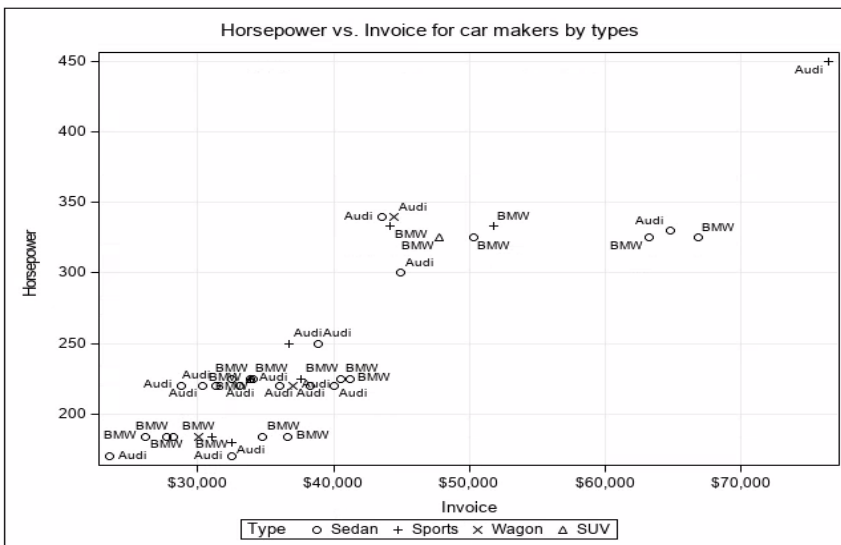
title 'Scatterplot - two variables';
proc sgscatter data = cars1;
plot horsepower*invoice
/ datalabel = make group = type grid;
title 'Horsepower vs. Invoice for car makers by types';
run;
```
```

ADDING SAS HTML OUTPUT II

Final output:

```
proc sql;
create table cars1 as
select make, model, type, invoice, horsepower, length, weight
from
sasHELP.cars
where make in ('Audi', 'BMW');
;
run;

title 'Scatterplot - two variables';
proc sgscatter data = cars1;
plot horsepower*invoice
/ datalabel = make group = type grid;
title 'Horsepower vs. Invoice for car makers by types';
run;
```



DATA PREPARATION IN SAS + PLOTS IN

R

Data Used

Anonymized data with the following variables:

- SUBJID - Subject Identifier.
- AVAL - Duration of progression free survival.
- PARAMCD - Progression free survival is selected. Duration from first dose date to death or PD.
- DXCLASS - Disease classification (Contains different indications of cancer).
- CNSR - Censoring status, 0 for event and greater than 0 for censored subjects.

Preparing plot data in sas

SAS chunk to prepare data for KM plot:

```
```{sas}
data inp.km_data;
 set inp.adtte;
 where PARAMCD eq "PFSPNV";
 CNSR = ifn(CNSR eq 0,1,0);
 keep SUBJID AVAL PARAMCD DXCLASS CNSR;
run;
```
```

Generating plot from R

```
```{r, echo=FALSE, message=FALSE}
library(haven); library(survival)
library(survminer); library(tidyverse)

dat <- read_sas(file.path(getwd(), "plot_data", "km_data.sas7bdat"))

fit survival model
surv_object <- Surv(time = dat$AVAL, event = dat$CNSR)
fit <- survfit(surv_object ~ DXCLASS, data = dat, conf.type = "log-log")

km plot
p <- ggsurvplot(fit, data = dat, pval = FALSE, risk.table = TRUE, font.size = 2,
 conf.int = F, axes.offset = FALSE, legend.title = "",
 legend.labs = c("MELANOMA", "NSCLC", "TCC"), break.x.by = 3,
 xlab = "(Months)", ylab = c("Progression-free Survival (%)"),
 ggtheme = theme_bw(), xlim = c(0, 34))

p$plot <- p$plot +
 scale_y_continuous(breaks = seq(0,1,by=0.1), labels = seq(0,100,by=10)) +
 theme(legend.position = "bottom", panel.grid.major = element_blank(),
 panel.grid.minor = element_blank())

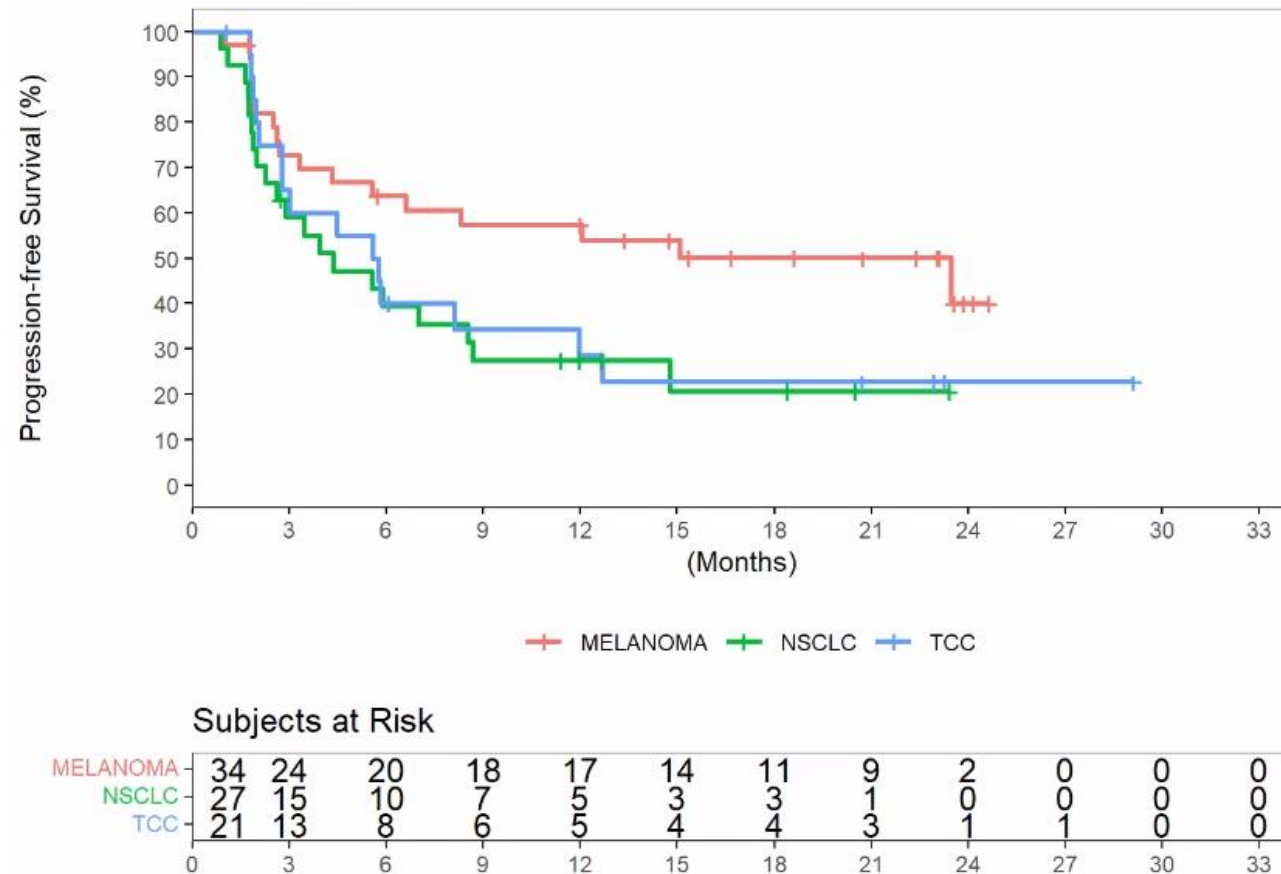
p$table <- p$table +
 ggtitle("Subjects at Risk") +
 theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
 xlab("")

p
```
```



R chunk to
generate KM plot

Generating plot from R



SOURCING SAS FILES INTO R MARKDOWN

- SAS files can be sourced into R markdown by using `%include` statement in a SAS chunk.
- Enables reusability of existing sas codes to produce outputs, saves time and efforts.

To include a waterfall chart by sourcing the program waterfall.sas, use:

```
```{sashtml5}  
%include "...\\code\\waterfall.sas";
```
```

CONCLUSION

- SAS markdown can provide an interface for interaction of SAS and R codes in a single markdown document.
- SAS programmers can easily plug in their existing SAS code blocks within R markdown to generate reports using the various engines provided by SASmarkdown package
- Can be made useful in scenarios where reports are generated repeatedly upon minor data updates and SAS codes are already available to generate the specific outputs.

THANK YOU

Any Questions ?