



Conversion of a SAS Macro to R PHUSE US CONNECT 2022



| Data Science



SAS Macro



- ▶ Opaque.
- ▶ Difficult to replicate.
- ▶ Written by programmers who are no longer available.
- ▶ Look at output and work backwards.

R Packages



- ▶ Foreign:
 - ▶ Reads .xpt but not sas7bdat
 - ▶ No writing capabilities
- ▶ Hmisc:
 - ▶ Reads .xpt and sas7bdat
 - ▶ No writing capabilities
- ▶ Haven:
 - ▶ Writes .xpt but labels not sticking
 - ▶ Can't write sas7bdat

SASXport



- ▶ Unable to write sas7bdat files.
- ▶ Writes .xpt files.
- ▶ Capabilities for adding labels, formats and changing attributes.
- ▶ Unable to alter length of columns.

SASxport - Functions



- ▶ SASformat - Used for adding SAS formats to .xpt datasets.
- ▶ SAStype - Used to add the type of column to .xpt datasets e.g Numeric, Character.
- ▶ toSAS - Controls how R objects are represented when stored in a SAS .xpt file.

Variable Lengths

```
# get attribute information before any transformations!"
varLabel <- attr(var, "label")
varFormat <- attr(var, "SASformat")
varIFormat <- attr(var, "SASiFormat")
varLength <- attr(var, "SASlength")

# Convert R object to SAS object
df[[i]] <- var <- toSAS(var, format.info=formats)

# compute variable length
if(is.character(var))
  varLen <- max(c(1,nchar(var, "bytes", keepNA=FALSE) ))
varLen <- ifelse(is.null(varLength), max(c(1,nchar(var, "bytes", keepNA=FALSE))), varLength)

# fill in variable offset and length information
offsetTable[i, "len"] <- varLen
offsetTable[i, "offset"] <- lenIndex
```

```
# get attribute information before any transformations!"
varLabel <- attr(var, "label")
varFormat <- attr(var, "SASformat")
varIFormat <- attr(var, "SASiFormat")

# Convert R object to SAS object
df[[i]] <- var <- toSAS(var, format.info=formats)

# compute variable length
if(is.character(var))
  varLen <- max(c(1,nchar(var, "bytes", keepNA=FALSE) ))
else
  varLen <- 8

# fill in variable offset and length information
offsetTable[i, "len"] <- varLen
offsetTable[i, "offset"] <- lenIndex

# parse format and iformat
formatInfo <- parseFormat(varFormat)
iFormatInfo <- parseFormat(varIFormat)

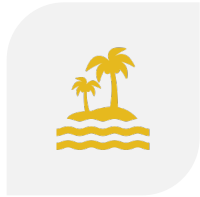
# write the entry
out(
  xport.namestr(
    var=var,
    varName = i,
    varNum = varIndex,
    varPos = lenIndex,
    varLength = varLen,
    varLabel = varLabel,
    fName = formatInfo$name,
    fLength = formatInfo$len,
    fLabel = formatInfo$label,
```

- ▶ Unable to be updated in original SASxport package.
- ▶ SasLength function created to update lengths.
- ▶ Ifelse inserted to check for length update.

Conclusion



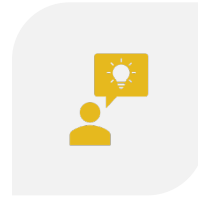
INITIALLY DIFFICULT
TO INTERPRET.



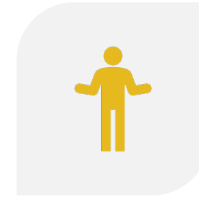
START WITH THE
END IN MIND.



RECREATING SAS
MACROS IN R IS
POSSIBLE.



SASXPORT, GREAT
PACKAGE WITH A
LOT OF DEPTH.



SMALL
CUSTOMIZATIONS
NEEDED.

Closing and Questions

Contact Information



Contact

Vincent Allibone

YPrime

Sandwich Discovery Park

Sandwich, Kent

vallibone@yprime.com

<https://www.yprime.com>

Closing and Questions

Contact Information



Contact

Frank Menius

YPrime

3301 Benson Dr #300

Raleigh, NC 27609

267-536-3268 fmenius@yprime.com

<https://www.yprime.com>

<https://www.linkedin.com/in/frank-menius/>