



Experts in R conversion

A Division of Boston Decision, LLC

# SAS to R: It's Not Just Syntax

*Common Pitfalls in SAS to R Conversions*



# About Rconvert.com

- ➔ Division of Boston Decision, LLC
- ➔ Founded 2010 - Cambridge, MA
- ➔ Finance, Marketing, Technology



[www.BostonDecision.com](http://www.BostonDecision.com)

# SAS & R - Frameworks

---

SAS and R are not just different languages.

- They are different frameworks.

# SAS Components

- Data Step
  - Functions
- Procedure Step
- Macro Language
- SAS ODS
- Component Product Languages
  - SAS/IML

# R Components

---

- Functions

# A Paradigm Shift

- In R, all work is performed by functions
  - Data steps = expressions with functions
  - Procedures = expressions with functions
  - Macros = expressions with functions
  - SAS functions = R functions

# Many Approaches

- In both SAS & R, there are usually multiple approaches to achieving the same result.

# SAS to R Mental Shift

- However, about 9 times out of 10:
  - The R approach most obvious to convert a SAS program works...
  - ...but is not the best.



# Performance Impact

- In practice, we've found:
  - It's very common for early R code written by a SAS programmer to show sluggish performance.
  - Matlab programmers generally write R code slightly more efficient at first pass.

# Conversion Planning

- Planning must ensure appropriate vectorization to fit the R framework.

# A Simple SAS to R Example

# A SAS to R Conversion Example

```
data squares (drop = i);  
  do i = 1 to 1000000;  
    y = i^2;  
    output;  
  end;  
run;
```

# Procedural Conversion Technique

```
data squares (drop = i);  
  do i = 1 to 1000000;  
    y = i^2;  
    output;  
  end;  
run;
```



```
y <- vector()  
  
for(i in 1:1000000)  
  y[i] <- i^2
```

# Procedural Performance

```
data squares (drop = i);  
  do i = 1 to 1000000;  
    y = i^2;  
    output;  
  end;  
run;
```



```
y <- vector()  
  
for(i in 1:1000000)  
  y[i] <- i^2
```

user	system	elapsed
1044.97	1015.47	2082.64

# Procedural with Allocation

```
data squares (drop = i);  
  do i = 1 to 1000000;  
    y = i^2;  
    output;  
  end;  
run;
```



```
y <- vector()  
  
length(y) <- 1000000  
  
for(i in 1:1000000)  
  y[i] <- i^2
```

# Procedural with Allocation

```
data squares (drop = i);  
  do i = 1 to 1000000;  
    y = i^2;  
    output;  
  end;  
run;
```



```
y <- vector()  
length(y) <- 1000000  
for(i in 1:1000000)  
  y[i] <- i^2
```



# Procedural Allocation Performance

```
data squares (drop = i);  
  do i = 1 to 1000000;  
    y = i^2;  
    output;  
  end;  
run;
```



```
y <- vector()  
  
length(y) <- 1000000  
  
for(i in 1:1000000)  
  y[i] <- i^2
```

user	system	elapsed
2.45	0.00	2.45

# Functional Conversion Technique

```
data squares (drop = i);  
  do i = 1 to 1000000;  
    y = i^2;  
    output;  
  end;  
run;
```



```
y <- (1:1000000)^2
```

# Functional Performance

```
data squares (drop = i);  
  do i = 1 to 1000000;  
    y = i^2;  
    output;  
  end;  
run;
```



```
y <- (1:1000000)^2
```

user	system	elapsed
0.02	0.00	0.02

# Performance Comparison

Conversion Technique	Elapsed Time (in seconds)	Performance Improvement
Procedural (“SAS-like” R programming)	2082.64	-
Procedural with Memory Allocation	2.45	850x
Functional	0.02	104,132x

# Summary

- SAS and R “think” differently.
- Both typically allow for multiple approaches.
- In R, think functional for best performance.



Experts in R conversion

A Division of Boston Decision, LLC

Thank you

*Timothy D'Auria*

# Disclaimers

- SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.
- R is available as Free Software under the terms of the [Free Software Foundation's GNU General Public License](#). R is hosted at [www.r-project.org](http://www.r-project.org).
- Boston Decision LLC and Rconvert.com are not affiliated with SAS Institute nor the core R development team.