Extending R with Compiled C code

Michael Anderson, PhD

Department of Biostatistics and Epidemiology, OU Health Sciences Center

Nov 19, 2012

The Wonders of R

- R has features that have attracted users over the years.
  - calculator to programming language all in one.
  - resampling schemes.
  - total plotting control.
  - infinitely expandable.
- It also has “features” that have puzzled users over the years.
  - Avoid loops in programming language like the plague.
  - Cumbersome data frame specifications.
  - Elementwise vs matrix/vector calculations.

Programming Language

Compiled Language

Commands are reduced to machine level language before the program is run (pre compilied).

Interpreted Language

Commands are reduced to machine level language at runtime (compilied line by line).

R is an interpreted language which affords some pros and cons

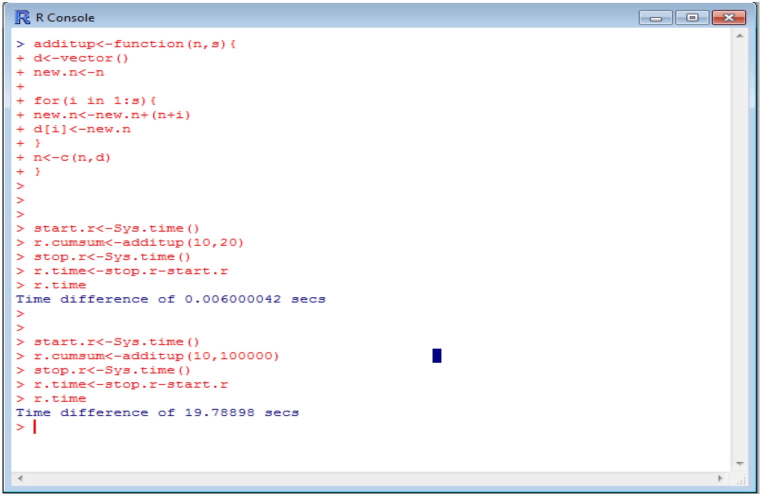
- Pro: Flexibilty of adding or changing functions at runtime.
- Con: Each line of a loop must be recompiled on every iteration

R Interpreting Code

```
additup<-function(n,s){

        new.n<-n

        for(i in 1:s){
            new.n<-new.n+(n+i)
            d[i]<-new.n
        }
        n<-c(n,d)
}
```

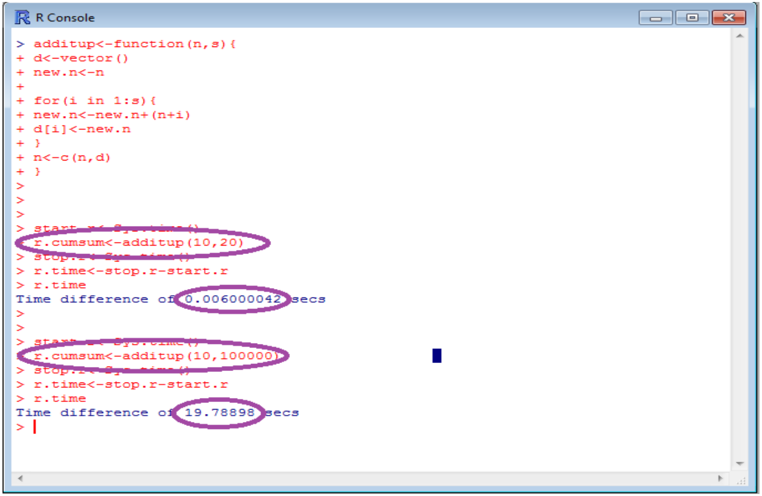
As the second argument, s, increases, so does the time required to perform the calculation.

R Interpreting Code



```
> additup<-function(n,s){
+ d<-vector()
+ new.n<-n
+
+ for(i in 1:s){
+ new.n<-new.n+(n+i)
+ d[i]<-new.n
+ }
+ n<-c(n,d)
+ }
>
>
>
> start.r<-Sys.time()
> r.cumsum<-additup(10,20)
> stop.r<-Sys.time()
> r.time<-stop.r-start.r
> r.time
Time difference of 0.006000042 secs
>
>
> start.r<-Sys.time()
> r.cumsum<-additup(10,100000)
> stop.r<-Sys.time()
> r.time<-stop.r-start.r
> r.time
Time difference of 19.78898 secs
>
```

R Interpreting Code



```
> additup<-function(n,s){
+ d<-vector()
+ new.n<-n
+
+ for(i in 1:s){
+ new.n<-new.n+(n+i)
+ d[i]<-new.n
+ }
+ n<-c(n,d)
+ }
>
>
>
> start.r<-Sys.time()
> r.cumsum<-additup(10,20)
> stop.r<-Sys.time()
> r.time<-stop.r-start.r
> r.time
Time difference of 0.006000042 secs
>
>
> start.r<-Sys.time()
> r.cumsum<-additup(10,100000)
> stop.r<-Sys.time()
> r.time<-stop.r-start.r
> r.time
Time difference of 19.78898 secs
> 
```

Another wonder of R

R can be extended using

- C for example `.C()`, `.Call()`.
- Fortran for example `.Fortran()`.

Some nice references for doing this are:

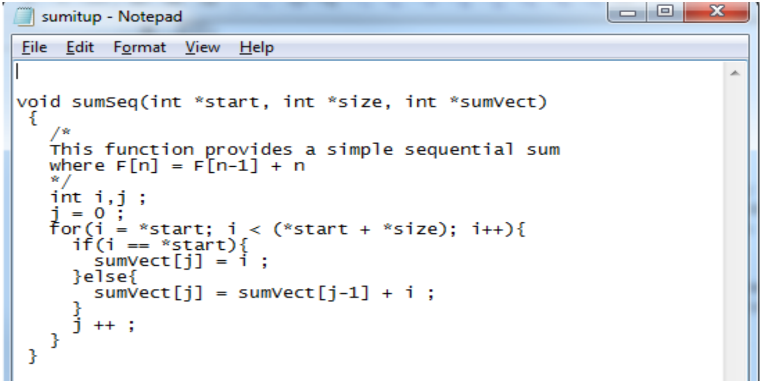
- **Writing R Extensions**
- **R Installation and Administration** (Appendix D)

Both are available at http://www.r-project.org/ (Documentation Manuals).

What to do?

The general idea:

1. Write a C function in C to do the heavy lifting.



```
void sumSeq(int *start, int *size, int *sumVect)
 {
   /*
   This function provides a simple sequential sum
   where F[n] = F[n-1] + n
   */
   int i,j ;
   j = 0 ;
   for(i = *start; i < (*start + *size); i++){
     if(i == *start){
       sumVect[j] = i ;
     }else{
       sumVect[j] = sumVect[j-1] + i ;
     }
     j ++ ;
   }
 }
```

What to do?

The general idea:

1. Write a C function in C to do the heavy lifting.
2. In a current R session, “outsource” the heavy lifting to C .
3. Retrieve the results from C for use in the current R session.

R becomes both a compiled and interpreted language.
Many R functions already do this (i.e. `cumsum()`).

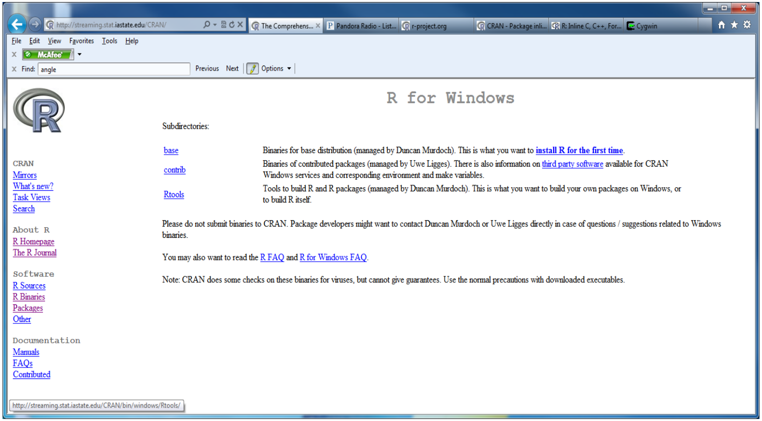
Necessary Background Work

Suppose saved C code as *filename.c*

- Unix/Linux EASY(already has compilers).
    1. Issue the command: R CMD SHLIB filename.c
    2. This creates a shared object file (filename.so).
- Windows COMPLEX (need to install compilers).
    1. Download and install “Rtools” from http://www.r-project.org.
    2. Define a new environment variable.
    3. Issue the command: R CMD SHLIB filename.c
    4. This creates a dynamic loadable library (filname.dll).

Install Rtools

Download and install latest version of “Rtools” from http://www.r-project.org.



R for Windows

Subdirectories:

base	Binaries for base distribution (managed by Duncan Murdoch). This is what you want to install R for the first time.
contrib	Binaries of contributed packages (managed by Uwe Ligges). There is also information on third party software available for CRAN Windows services and corresponding environment and make variables.
Rtools	Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

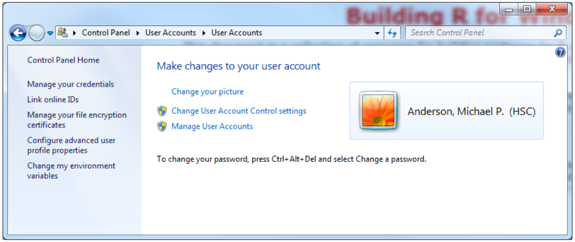
Environment Variable

Define a new environment variable to specify the path Windows is to use to find the compiler.

For Windows Vista and Windows 7, Enviroment variables can be modified by:

- Control Panel
- User Accounts
- Change Environment Variables

Environment Variable



Control Panel ▸ User Accounts ▸ User Accounts

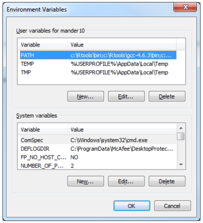
Control Panel Home
Manage your credentials
Link online IDs
Manage your file encryption certificates
Configure advanced user profile properties
Change my environment variables

Make changes to your user account

Change your picture
Change User Account Control settings
Manage User Accounts

Anderson, Michael P. (HSC)

To change your password, press Ctrl+Alt+Del and select Change a password.



Environment Variables

User variables for mander10

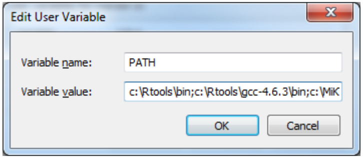
Variable	Value
PATH	c:\Rtools\bin;c:\Rtools\gcc-4.6.3\bin;c:...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DEFLOGDIR	C:\ProgramData\McAfee\DesktopProtec...
FP_NO_HOST_C...	NO
NUMBER_OF_P...	2

Environment Variable

Define a new variable "PATH" with the following pathway


Edit User Variable

Variable name: PATH

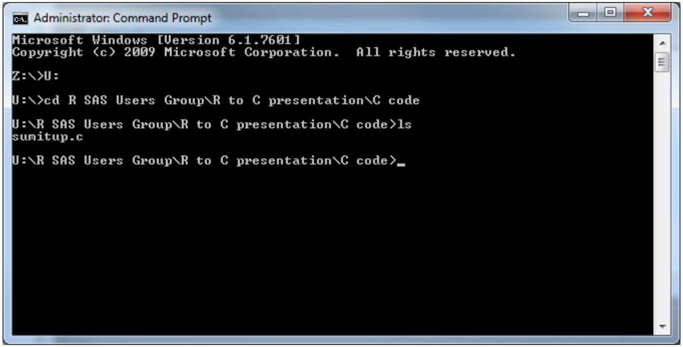
Variable value: c:\Rtools\bin;c:\Rtools\gcc-4.6.3\bin;c:\MiK

For example for a 32-bit build, all on one line,

```
    PATH=c:\Rtools\bin;c:\Rtools\gcc-4.6.3\bin;c:\MiKTeX\miktex\bin;
         c:\R\R-2.15\bin\i386;c:\windows;c:\windows\system32
```

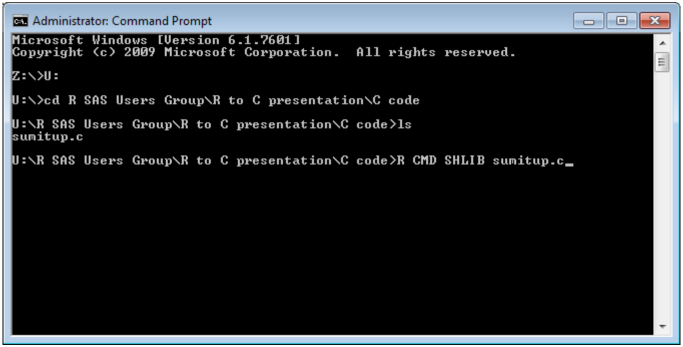
It is essential that the directory containing the command line tools comes first or second in the path: there are typically like-named tools[45] in other directories, and they will **not** work. The ordering of the other directories is less important, but if in doubt, use the order above.

Command Prompt



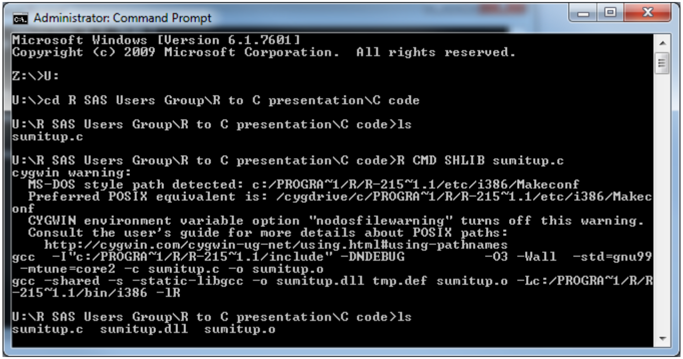
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

Z:\>U:

U:\>cd R SAS Users Group\R to C presentation\C code

U:\R SAS Users Group\R to C presentation\C code>ls
sumitup.c

U:\R SAS Users Group\R to C presentation\C code>_
```

Command Prompt



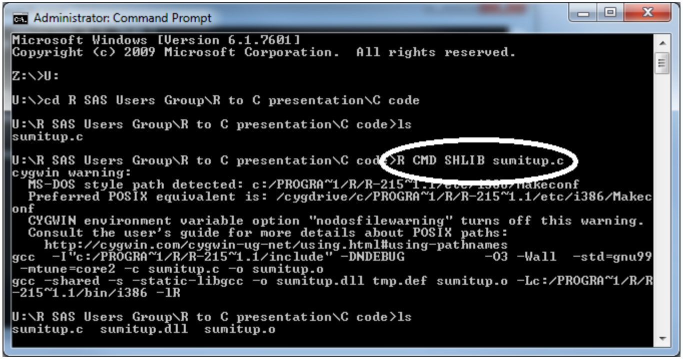
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

Z:\>U:

U:\>cd R SAS Users Group\R to C presentation\C code

U:\R SAS Users Group\R to C presentation\C code>ls
sumitup.c

U:\R SAS Users Group\R to C presentation\C code>R CMD SHLIB sumitup.c_
```

Command Prompt



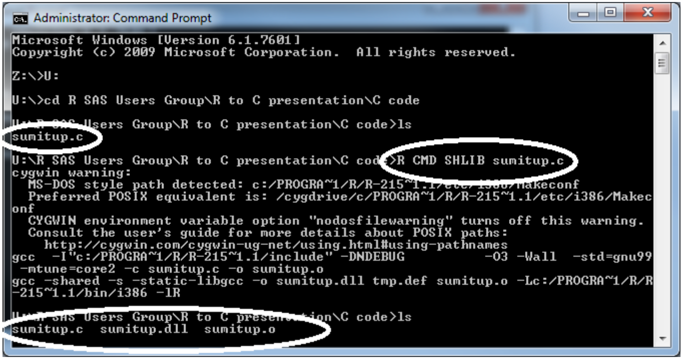
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

Z:\>U:

U:\>cd R SAS Users Group\R to C presentation\C code

U:\R SAS Users Group\R to C presentation\C code>ls
sumitup.c

U:\R SAS Users Group\R to C presentation\C code>R CMD SHLIB sumitup.c
cygwin warning:
  MS-DOS style path detected: c:/PROGRA~1/R/R-215~1.1/etc/i386/Makeconf
  Preferred POSIX equivalent is: /cygdrive/c/PROGRA~1/R/R-215~1.1/etc/i386/Makec
onf
  CYGWIN environment variable option "nodosfilewarning" turns off this warning.
  Consult the user's guide for more details about POSIX paths:
    http://cygwin.com/cygwin-ug-net/using.html#using-pathnames
gcc  -I"c:/PROGRA~1/R/R-215~1.1/include" -DNDEBUG          -O3 -Wall  -std=gnu99
 -mtune=core2 -c sumitup.c -o sumitup.o
gcc -shared -s -static-libgcc -o sumitup.dll tmp.def sumitup.o -Lc:/PROGRA~1/R/R
-215~1.1/bin/i386 -lR

U:\R SAS Users Group\R to C presentation\C code>ls
sumitup.c  sumitup.dll  sumitup.o
```

Command Prompt



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

Z:\>U:

U:\>cd R SAS Users Group\R to C presentation\C code

U:\R SAS Users Group\R to C presentation\C code>ls
sumitup.c

U:\R SAS Users Group\R to C presentation\C code>R CMD SHLIB sumitup.c
cygwin warning:
  MS-DOS style path detected: c:/PROGRA~1/R/R-215~1.1/etc/i386/Makeconf
  Preferred POSIX equivalent is: /cygdrive/c/PROGRA~1/R/R-215~1.1/etc/i386/Makec
onf
  CYGWIN environment variable option "nodosfilewarning" turns off this warning.
  Consult the user's guide for more details about POSIX paths:
    http://cygwin.com/cygwin-ug-net/using.html#using-pathnames
gcc  -I"c:/PROGRA~1/R/R-215~1.1/include" -DNDEBUG          -O3 -Wall  -std=gnu99
 -mtune=core2 -c sumitup.c -o sumitup.o
gcc -shared -s -static-libgcc -o sumitup.dll tmp.def sumitup.o -Lc:/PROGRA~1/R/R
-215~1.1/bin/i386 -lR

U:\R SAS Users Group\R to C presentation\C code>ls
sumitup.c  sumitup.dll  sumitup.o
```

Command Prompt



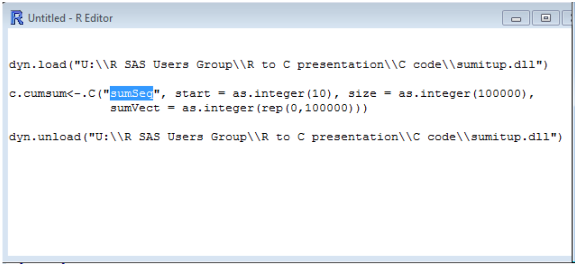
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

Z:\>U:

U:\>cd R SAS Users Group\R to C presentation\C code

U:\R SAS Users Group\R to C presentation\C code>ls
sumitup.c

U:\R SAS Users Group\R to C presentation\C code>R CMD SHLIB sumitup.c
cygwin warning:
  MS-DOS style path detected: c:/PROGRA~1/R/R-215~1.1/etc/i386/Makeconf
  Preferred POSIX equivalent is: /cygdrive/c/PROGRA~1/R/R-215~1.1/etc/i386/Makec
onf
  CYGWIN environment variable option "nodosfilewarning" turns off this warning.
  Consult the user's guide for more details about POSIX paths:
    http://cygwin.com/cygwin-ug-net/using.html#using-pathnames
gcc  -I"c:/PROGRA~1/R/R-215~1.1/include" -DNDEBUG          -O3 -Wall  -std=gnu99
 -mtune=core2 -c sumitup.c -o sumitup.o
gcc -shared -s -static-libgcc -o sumitup.dll tmp.def sumitup.o -Lc:/PROGRA~1/R/R
-215~1.1/bin/i386 -lR

U:\R SAS Users Group\R to C presentation\C code>ls
sumitup.c  sumitup.dll  sumitup.o
```

Now What?

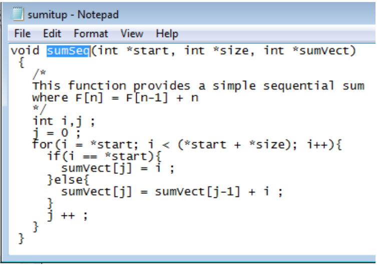
Now that we have the DLL file, we can use it within R:

- `dyn.load(‘‘path/to/filename.dll’’)`
- `.C(‘‘C function name’’,arguments to send to C, arguments C will return)`
  - First argument needs to match the name of the C function in the filname.c
  - arguments sent/returned must be numeric.
  - arguments sent/returned must be vectors or scalars.
  - `.Call()` allows strings, and arrays to be passed.
- `dyn.unload(‘‘path/to/filename.dll’’)`

R objects to C objects

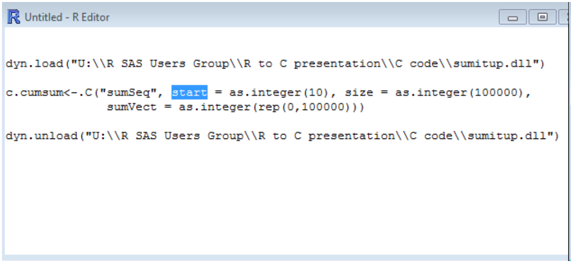


```
dyn.load("U:\\R SAS Users Group\\R to C presentation\\C code\\sumitup.dll")

c.cumsum<-.C("sumSeq", start = as.integer(10), size = as.integer(100000),
              sumVect = as.integer(rep(0,100000)))

dyn.unload("U:\\R SAS Users Group\\R to C presentation\\C code\\sumitup.dll")
```

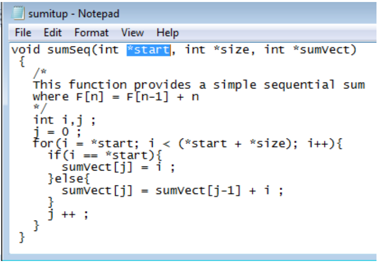


```
void sumSeq(int *start, int *size, int *sumVect)
 {
   /*
   This function provides a simple sequential sum
   where F[n] = F[n-1] + n
   */
   int i,j ;
   j = 0 ;
   for(i = *start; i < (*start + *size); i++){
     if(i == *start){
       sumVect[j] = i ;
     }else{
       sumVect[j] = sumVect[j-1] + i ;
     }
     j ++ ;
   }
 }
```

R objects to C objects

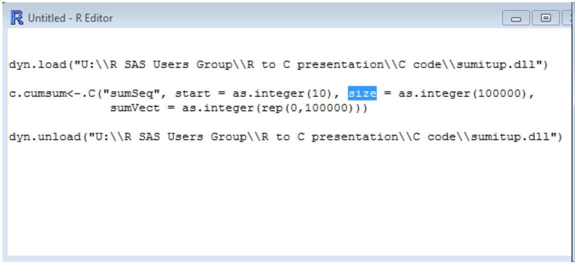


```
dyn.load("U:\\R SAS Users Group\\R to C presentation\\C code\\sumitup.dll")

c.cumsum<-.C("sumSeq", start = as.integer(10), size = as.integer(100000),
              sumVect = as.integer(rep(0,100000)))

dyn.unload("U:\\R SAS Users Group\\R to C presentation\\C code\\sumitup.dll")
```



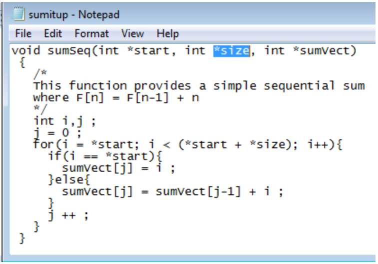
```
void sumSeq(int *start, int *size, int *sumVect)
 {
   /*
   This function provides a simple sequential sum
   where F[n] = F[n-1] + n
   */
   int i,j ;
   j = 0 ;
   for(i = *start; i < (*start + *size); i++){
     if(i == *start){
       sumVect[j] = i ;
     }else{
       sumVect[j] = sumVect[j-1] + i ;
     }
     j ++ ;
   }
 }
```

R objects to C objects



Untitled - R Editor

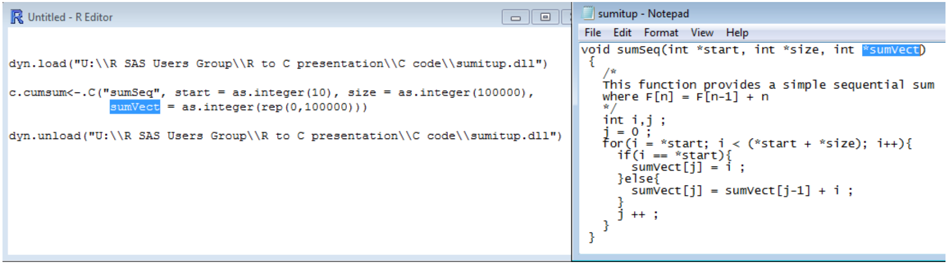
```
dyn.load("U:\\R SAS Users Group\\R to C presentation\\C code\\sumitup.dll")

c.cumsum<-.C("sumSeq", start = as.integer(10), size = as.integer(100000),
             sumVect = as.integer(rep(0,100000)))

dyn.unload("U:\\R SAS Users Group\\R to C presentation\\C code\\sumitup.dll")
```



sumitup - Notepad

```
void sumSeq(int *start, int *size, int *sumVect)
 {
   /*
   This function provides a simple sequential sum
   where F[n] = F[n-1] + n
   */
   int i,j ;
   j = 0 ;
   for(i = *start; i < (*start + *size); i++){
     if(i == *start){
       sumVect[j] = i ;
     }else{
       sumVect[j] = sumVect[j-1] + i ;
     }
     j ++ ;
   }
 }
```

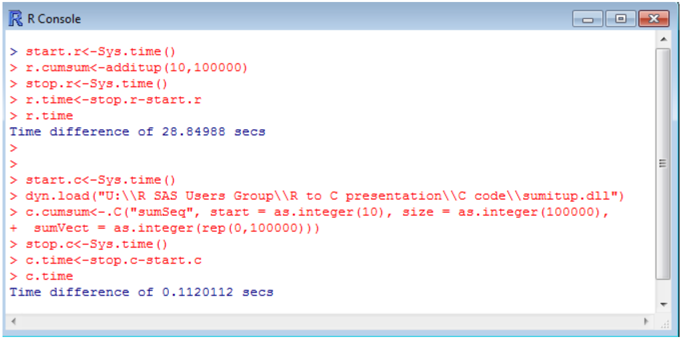
R objects to C objects



```
dyn.load("U:\\R SAS Users Group\\R to C presentation\\C code\\sumitup.dll")

c.cumsum<-.C("sumSeq", start = as.integer(10), size = as.integer(100000),
             sumVect = as.integer(rep(0,100000)))

dyn.unload("U:\\R SAS Users Group\\R to C presentation\\C code\\sumitup.dll")
```

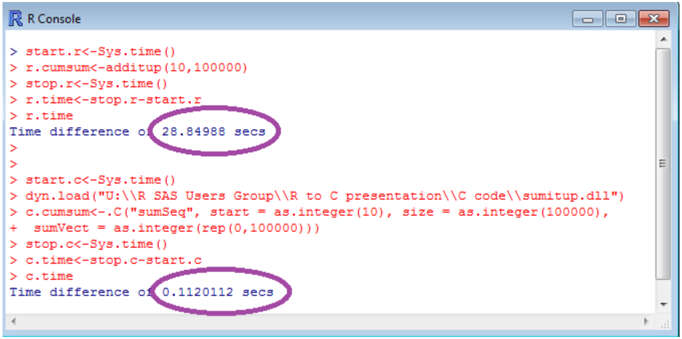
```
void sumSeq(int *start, int *size, int *sumVect)
 {
   /*
   This function provides a simple sequential sum
   where F[n] = F[n-1] + n
   */
   int i,j ;
   j = 0 ;
   for(i = *start; i < (*start + *size); i++){
     if(i == *start){
       sumVect[j] = i ;
     }else{
       sumVect[j] = sumVect[j-1] + i ;
     }
     j ++ ;
   }
 }
```

Time Comparison



```
> start.r<-Sys.time()
> r.cumsum<-additup(10,100000)
> stop.r<-Sys.time()
> r.time<-stop.r-start.r
> r.time
Time difference of 28.84988 secs
>
>
> start.c<-Sys.time()
> dyn.load("U:\\R SAS Users Group\\R to C presentation\\C code\\sumitup.dll")
> c.cumsum<-.C("sumSeq", start = as.integer(10), size = as.integer(100000),
+  sumVect = as.integer(rep(0,100000)))
> stop.c<-Sys.time()
> c.time<-stop.c-start.c
> c.time
Time difference of 0.1120112 secs
```

Time Comparison



```
> start.r<-Sys.time()
> r.cumsum<-additup(10,100000)
> stop.r<-Sys.time()
> r.time<-stop.r-start.r
> r.time
Time difference of 28.84988 secs
>
>
> start.c<-Sys.time()
> dyn.load("U:\\R SAS Users Group\\R to C presentation\\C code\\sumitup.dll")
> c.cumsum<-.C("sumSeq", start = as.integer(10), size = as.integer(100000),
+  sumVect = as.integer(rep(0,100000)))
> stop.c<-Sys.time()
> c.time<-stop.c-start.c
> c.time
Time difference of 0.1120112 secs
```

THANK YOU