R to C: The Comprehensive Guide

Setup:

Installation

Run the makefile as appropriate. Make sure that R.h and Rinternals.h are in your path. You can get these by installing the R binary. However, the R IDE, RStudio, is an **extremely** helpful addition.

```
# run me!
make interface
```

Linking the library and running c code

1) Start R in terminal using the command:

```
sudo R
```

note: sudo permissions are important for using the pcap library in c.

2) While in R, it requires us to link the library in R using:

```
dyn.load("interface.so")
```

3) Test it out! Call a C function from R using `.Call("my_c_function", ...)`, where the elipses supply the parameters, if necessary. Try it using:

```
.Call("testThread")
```

4) In order to use capture.c, we should follow these steps:

```
# This will startup capture.c, which will reroute standard out to /dev/null. This ensures # that our te
.Call("startSession")

# Use this to end the session of capture.c, which will stop the thread running in the backround. A succ
.Call("endSession")
```

Running RShiny

1) Install RShiny using either bash or R

```
# bash
R -e "install.packages(\"shiny\", repos=\"http://cran.us.r-project.org\")"
```

```
# R
install.packages("shiny", repos = "http://cran.us.r-project.org")
```

2) Run the Server!

```bash
# bash
# Warning: dont run this inside the RMarkdown Document. It may hide the port number.
R -e "shiny::runApp('./Viewer')"
```

```r
# R
shiny::runApp('./Viewer')
```