Debugging_Tools

yob

3/31/2019

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

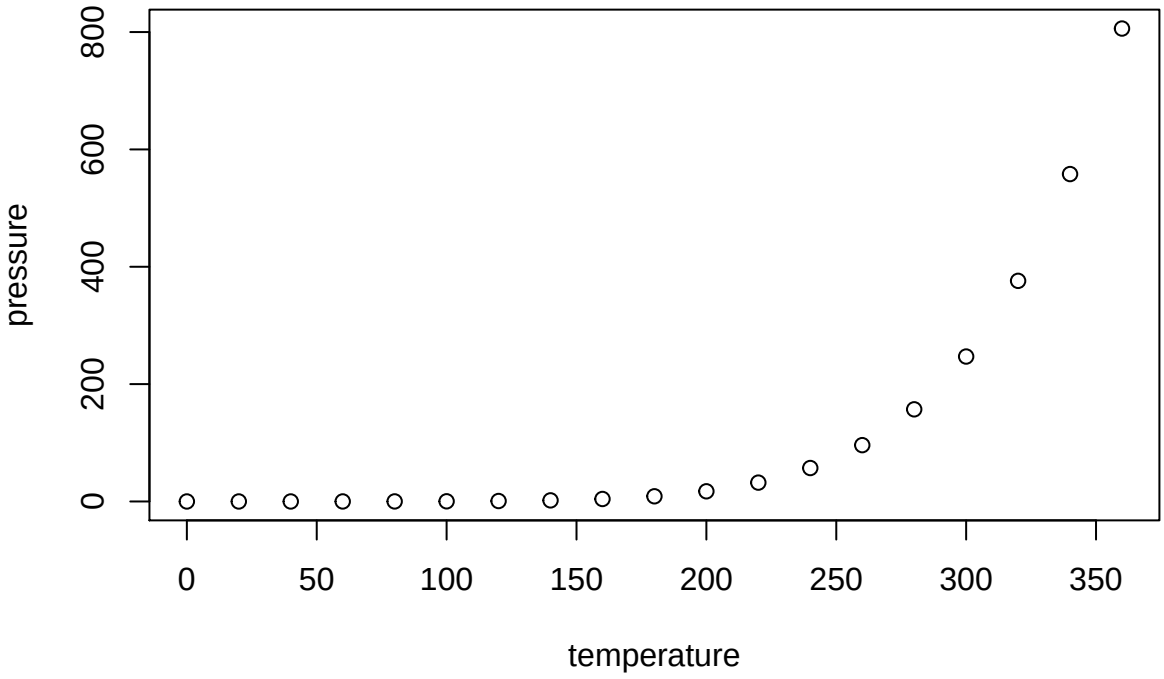
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

*** DEBUGGIN TOOLS ***

indications that something's not right

message : tame notification, diagnostic message, it could be nothing won't stop the function from executing

message function

warning : indication that something unexpected happened sometimes you may want to ignore warnings. won't stop executing generated by warning function

error : last stop. fatal problem. stops execution.

error messages are produced. produced by stop function

condition : a generic concept for indication that something unexpected can occur

log(-1) #[1] NaN #Warning message: # In log(-1) : NaNs produced

typical warning. maybe you don't care. you don't want the function's behaviour to stop

printmessage <- function(x) { if(x > 0) print("x is greater than zero") else print("x is less than or equal to zero") invisible(x) #stops or prevents auto printing the # lastly used element- in this case the numeric vector. # if you call invisible, it will still return the same object, # but will not autoprint to console }

printmessage(1) # works fine

printmessage(NA) #error here. NA > 0 is not defined. it can't move out # so it has to error out. # it was expecting TRUE/FALSE value

printmessage2 <- function(x) { if(is.na(x)) print("x is a missing value!") else if(x > 0) print("x is greater than zero") else print("x is less than or equal to zero") invisible(x) #stops or prevents auto printing the

}

x<- log(-1)

printmessage2(x)

how do you know that something is wrong with your function?

what was your input? how did you call the function?

what were you expecting? output, messages, other results?

what did you get?

how does what you get differ from what you were expecting?

were your expectations correct in the first place?

can you reproduce the problem(exactly)? ********!!!!!!!!!

Debugging Tools

primary tools for debugging functions in R are

traceback

prints out the function call stack after an error occurs; does nothing if there's no error

debug

flags a function for "debug" mode which allows you to step through execution of a function one line at a time

browser

suspends the execution of a function wherever it is called and puts the function in debug more

trace

allows you to insert debugging code into a function a specific places handy if you are debugging someone else's code

recover

allows you to modify the error behavior so that you can browse the function call stack