

Step 1: Installing the package

[emayili](#) is now on CRAN so you can install it directly from there.

If you want to get the development version, you can download it using `devtools::install_github()`

```
# Option 1: Download the `emayili` package from CRAN
```

```
install.packages("emayili")
```

```
# Option 2: Download the emayili package from Github with the `devtools` package
```

```
install.packages("devtools")
```

```
devtools::install_github("datawookie/emayili")
```

```
# Or to get the package from the development branch
```

```
devtools::install_github("datawookie/emayili", ref="dev")
```

Step 2: writing your email

Because there are many customizable options for writing an email with `emayili`, I have put together a function called `write_email()` which contains the arguments which I wanted. You could just pipe (`%>%`) everything together and not have a function, but for our script – having a pre-built function makes things easier IMHO.

There are definitely better ways to do this, but here is what worked for me!

The function:

```
library(emayili)
```

```
write_email<-function(sender,
                      reciever,
                      reciever_name,
                      subject_line,
                      opening_line= "Dear",
                      body_text){

  all_text<-paste(opening_line,reciever_name,",",",","\n",body_text)

  #Using emayili

  envelope() %>%
    from(sender) %>%
    to(reciever) %>%
    subject(subject_line) %>%
    text(content =all_text)

}
```

The first four arguments are self explanatory. The `opening_line` argument I have set as default as "Dear" which is essentially a introduction line that you can have to address your recipient. You can change it to "Hello" or whatever you like.

The `body_text` argument is the standard text you would like to include. This is all put together into the `all_text` variable embedded in the function and passed into the `text()` function which was chained to the rest of the `emayili` functions.

Using `mapply()` to implement the merge:

```
mapply(function(email,name){
  email<-write_email(sender="myemail@hotmail.com",
    reciever=email,
    reciever_name = name,
    subject_line='Personal from R',
    body_text=
      "\nThanks for letting me spam your email with my robot
\n\nBest,\n\nBenjamin")

  # Instalize STMP server
  smtp <- server(host = "smtp.live.com",
    port = 587,
    username = "myemail@hotmail.com",
    password = '*****')

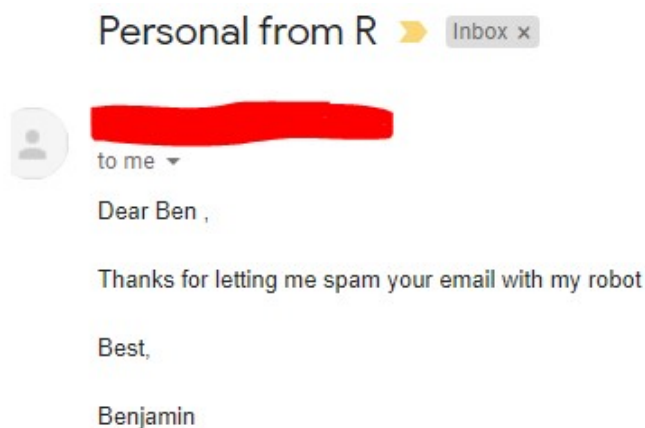
  smtp(email, verbose = TRUE)

  # Added System Sleep to Not slam the server with requests and to
  # avoid an ip-ban.

  Sys.Sleep(runif(1, max=5))
},res,nams)
```

To send the email we need to instalize a STMP server. Based on the host you will be using your port will vary. Some quick googling can help you figure out your appropriate host and port (or you can just check [here](#)).

After running this script your blast *should be* sucessful!



And there you have it!

Conclusion

Breaking into tech can be tough. Just because you are not in tech doesn't mean you can't use a programming language like R to help with your work- even if you work in sales or admin.

I hope this blog opens your mind to using R beyond statistical analysis.

Let me know what you think!