Week 8 Extension Code

Oakleigh Wilson

2025-09-07

Table of Contents

[Automated Report Writing 1](#_Toc208161316)

[How to Open an R Markdown 1](#_Toc208161317)

[Case Study Data 2](#_Toc208161318)

[Embedding Code 2](#_Toc208161319)

[Embedding Plots 2](#_Toc208161320)

[Extracting the stats automatically 3](#_Toc208161321)

[Putting the stats into a table 4](#_Toc208161322)

[Automating extraction across multiple variables 5](#_Toc208161323)

[Embedding the results in sentences 6](#_Toc208161324)

[Final Comments 7](#_Toc208161325)

## Automated Report Writing

When I worked as a data analyst for an ecology consultancy, one of my main jobs was to explain clients’ data to them. They would send me giant awful sprawling excel workbooks of years’ worth of field annotations and I would wrangle, analyse, summarise, and visualise this as a streamlined report. After a hard fortnight’s work, however, the client would inevitably change some detail and want me to do it all over again… Rather than writing a new report every time (copying in new stats and graphs after every change), I could make a “report scaffold” that would largely self-populate and save me heaps of time and error. I did this using the R functionality called “R Markdown”.

R Markdown is what this extension code is currently written in. It is essentially a syntax for writing text documents with embedded code and code output. In this extension code, we will look at R Markdowns basic functionality.

## How to Open an R Markdown

To open a new R Markdown file to follow along with, go to the file+ icon and select R Markdown. It will open a blank version of this file that you can mess around with. Select whether you want your final document rendered as a PDF, html, or word. When you’ve written your code in the interactive version of the file, you can then “knit” it to the render type. There should be a button for that in your task bar.

To follow along with this tutorial and compare input with output, knit this current document.

## Case Study Data

For this example, we’ll use the lab 4 practice data again.

### Embedding Code

The writing here is just plain text. To make it code, you need to wrap it in notation as below. The curly parentheses section “{r}” is where we can define some setting for our “chunk”. The main options to be aware of are: - include = TRUE/FALSE -> whether the code and output appear in the final document. Defaults to true. - echo = TRUE/FALSE -> whether the code itself is shown (output still appears if include=TRUE). Defaults to true. - warning = TRUE/FALSE -> whether warnings from code execution are printed (error does the same for errors and message for messages)

For example, to show the code and the output:

data$BL[1]

## [1] 21.394

Or to show the output but not the code:

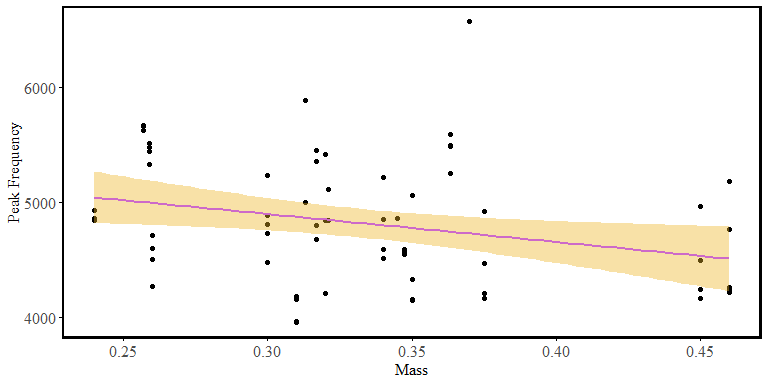
## [1] 21.394

And then neither (but whatever it is is still run in the background):

… obviously nothing will be displayed in the final render here.

### Embedding Plots

You can also embed graphs into your display and set echo = FALSE to prevent the source code from printing. Control the size of the graph in the options ({r}) section by specifying fig.width and/or fig.height. Remember to add your custom theme!



### Extracting the stats automatically

Thus far, to extract the stats from the linear models we have been generating, we have printed summary(model) and then visually inspected the output.

mass\_freq\_model <- lm(freq ~ Mass, data = data)  
summary(mass\_freq\_model)

##   
## Call:  
## lm(formula = freq ~ Mass, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -922.9 -291.6 -112.0 414.1 1846.0   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5626.3 339.2 16.588 <2e-16 \*\*\*  
## Mass -2430.5 997.8 -2.436 0.0175 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 520.3 on 67 degrees of freedom  
## Multiple R-squared: 0.08136, Adjusted R-squared: 0.06765   
## F-statistic: 5.934 on 1 and 67 DF, p-value: 0.01752

This method is time consuming and prone to human error. Instead, we could automatically extract these values.

mass\_freq\_model <- lm(freq ~ Mass, data = data)  
model\_summary <- summary(mass\_freq\_model)  
  
# extract everything from the model summary  
intercept <- model\_summary$coefficients[1,1]  
slope <- model\_summary$coefficients[2,1]  
  
adj\_r2 <- model\_summary$adj.r.squared  
rounded\_adj\_r2 <- round(adj\_r2, 3)  
  
# F-statistic and p-value  
f\_value <- model\_summary$fstatistic[1]  
p\_value <- model\_summary$coefficients[2,4]   
  
# print them out  
intercept

## [1] 5626.316

slope

## [1] -2430.518

rounded\_adj\_r2

## [1] 0.068

f\_value

## value   
## 5.933781

p\_value

## [1] 0.01751917

### Putting the stats into a table

Make it neater by coercing all the stats into a table.

mass\_freq\_stats <- data.frame( # make a data.frame  
 Intercept = intercept, # column name and the variable  
 Slope = slope,  
 R2 = rounded\_adj\_r2,  
 F\_value = f\_value,  
 P\_value = p\_value  
)  
mass\_freq\_stats

## Intercept Slope R2 F\_value P\_value  
## value 5626.316 -2430.518 0.068 5.933781 0.01751917

#### Bonus Task 1

The table above is pretty ugly and not appropriate for inclusion in a professional report. R Markdown uses the package ‘kable’ from package ‘knitr’ to make nice tables. You could google it and find some tutorials or something to make your table nicer.

### Automating extraction across multiple variables

The next step is to automate these analyses across multiple variables. Before we do it together, can you think of how you would do it yourself?

There are many different ways to solve the same problem, but an intuitive way to do it might be with a very simple ‘for loop’. A ‘for loop’ is a loop which iterates through a list of options. It follows this format:

for (i in 1:3){ # for iterator in list of things to iterate across  
 print(i) # the task you want it to do  
}

## [1] 1  
## [1] 2  
## [1] 3

For loops are slow and clunky when it comes to big data tasks, but for most small data things, they will be your friend. In this case, lets iterate across the different variables in our data and extract the slope from each.

dependent\_variables <- c("chirp", "freq")  
  
# make an empty data.frame to save your results in  
results <- data.frame(  
 Variable = character(),  
 Slope = numeric()  
)  
  
for (variable in dependent\_variables) {  
   
 # We can't just say "variable" in our formula because R will take it literally  
 # instead we can try building it as a formula and then feeding it in  
 # there are many other ways to do it too, but this one works  
 formula <- reformulate("Mass", response = variable)  
   
 # Fit model  
 model <- lm(formula, data = data)  
 summary\_model <- summary(model)  
   
 # Extract statistics like we did above  
 slope <- coef(model)[2]  
   
 # Add row to results using 'rbind' (literally row bind -- bind the rows together)  
 results <- rbind(results, data.frame(  
 Variable = variable,  
 Slope = slope  
 ))  
}  
results

## Variable Slope  
## Mass chirp -0.86743  
## Mass1 freq -2430.51782

#### Bonus Task 2

Expand the system above to calculate more possible combinations of variables. You might want a system to test Mass, BL, FWW, and FWL against chirp and freq… how would these independent variables fit into the above formula reformulation? Could you nest multiple for loops inside each other?

### Embedding the results in sentences

At the end of the day, clients don’t just want graphs and tables, they want plain English sentences. Your job as an ecologist is to take all the information from the data and summarise it to something the client can understand… That being said… if you’re a lazy feral like me you can also partially automate this too lol. (In my defence, it eliminates human error!)

for (i in 1:nrow(results)){  
 output <- results[i,]  
 variable <- output$Variable  
 valence <- ifelse(output$Slope>0, "positive", "negative")  
  
 print(paste0("The relationship between Mass and ", variable, " is ", valence, "."))  
}

## [1] "The relationship between Mass and chirp is negative."  
## [1] "The relationship between Mass and freq is negative."

And so on and so forth. In this particular example, it isn’t very useful but I hope you can imagine the possible utility of this idea.

#### Bonus Task 3

Can you make print statements reporting the relationship and significance of each of the linear models we investigated? Include the stats in a print statement or a table.

#### Bonus Tasks

Why not try producing your Lab 4 report using R markdown?

## Final Comments

While this version has been rendered to a word document appropriate for inclusion in an ecology report, the sky is the limit with this tool. Instead of a word document you can render your files to a html (like a web page) and embed significantly more complex things - such as interactive graphs and dashboards. Check out this link for examples: <https://rmarkdown.rstudio.com/gallery.html>

What you do with an R Markdown is limited only by your imagination and I hope you can think of many creative ways to use this fun tool :)