

R You Ready for New Tools to Manage Research Resources?

ASA-CSA-SSSA Annual International Meeting

October 22, 2017 Tampa, FL



A. M. Poncet, T. Knappenberger

Auburn University



E. Van Santen, G. Ben-Chanoch

University of Florida



Preamble

Reproducibility is one of the most fundamental cornerstones of science. Research results are typically considered **replicable** if there is sufficient information available for independent researchers to obtain the same findings using the same methods with new data. Working to make your research reproducible surely requires extra upfront effort but beyond the direct benefit to science, succeeding can also benefit you personally by making your research process more effective and easier (Gandrud 2015; Xie 2015).

Cutting-edge statistical analysis often requires tools not yet available in commercial software-packages and the open-source programming language and development environment R has gained increased acceptance within the research community as it provides one solution to these problems. In this workshop you will learn how to use RStudio and R Markdown as one tool integrating data storage, analysis, graphing, and manuscript writing. You will first learn how to create your own R-package to store data, code, and share your research with other people. You will then learn how to use R Markdown to integrate data analysis, graphing, and manuscript writing.

To be able to complete this workshop, participants are required to install all the R-packages listed below. Windows users must also install the latest version of the RTool application.

- htmltools (RStudio and Inc. 2016),
- caTools(Tuszynski 2014),
- bitops (Steve Dutky initial R port, Martin Maechler; revised, and Steve Dutky 2013),
- base64enc (Urbanek 2015),
- rmarkdown (Allaire et al. 2016),
- devtools (Wickham and Chang 2017),
- roxygen2 (Wickham, Danenberg, and Eugster 2017),
- knitr (Xie 2015), and
- ggplot2 (Wickham 2009).
- plyr (Wickham 2011)

You can do so using the following script.

```
#"This script installs R packages required to complete the Workshop Tutorial"

# lists all packages already installed
packages.list <- rownames(as.data.frame(installed.packages()))

# lists all packages that must be installed
packages.required <- c("htmltools", "caTools", "bitops", "base64enc", "rmarkdown",
                       "devtools", "roxygen2", "knitr", "ggplot2", "plyr")

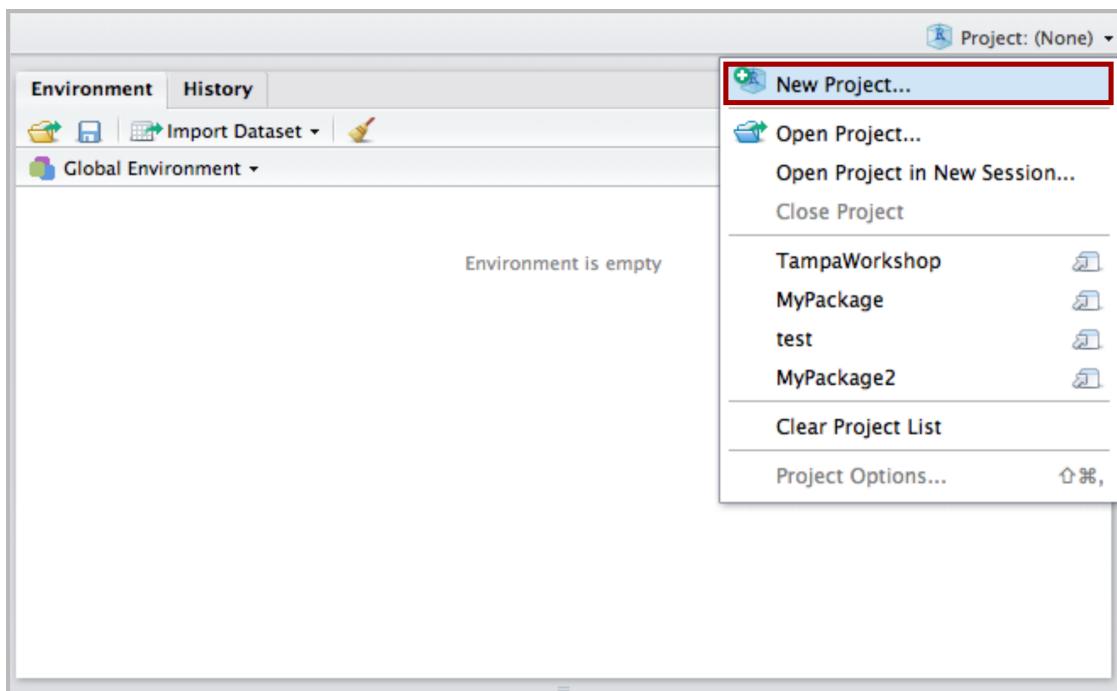
# install missing packages
for(package in packages.required){ # iteration over required packages
  if(package %in% packages.list == F){ # if package has not been installed yet
    install.packages(package)
  }
}
```

Section 1. Creating an R-Package

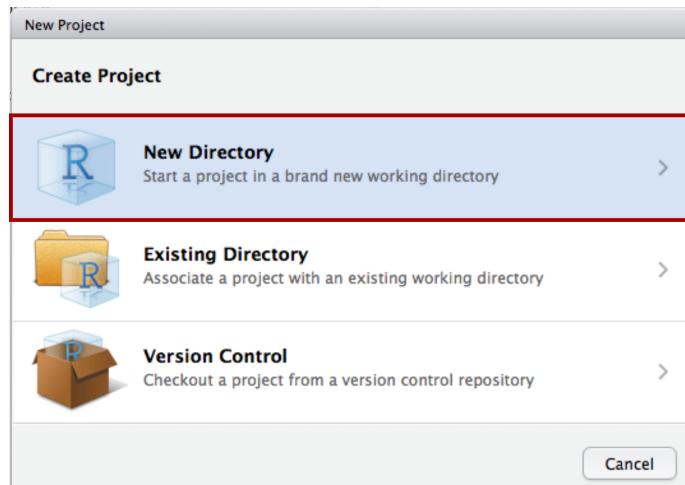
Are you looking for an efficient way to store your data, metadata, and code to share your research with others or simply to make it easier on yourself? Stop looking, the solution is simple: you need to create your own R-package! The simplest way to create an R-package is to use the `devtools` and `roxygen2` packages and this first section will get you started with creating your own R-packages using these two libraries. You will be guided through a step by step tutorial. The following topics will be covered: including and documenting datasets as well as functions within an R-package, file structure, and exporting the package in a format that you can share with other users

1. Create a New Project.

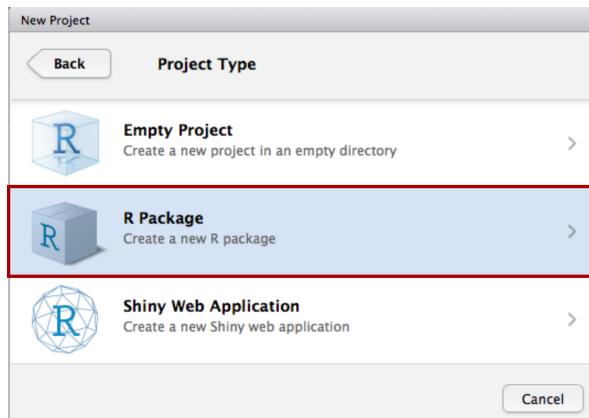
- a. Click on Project: (None), and select New Project.



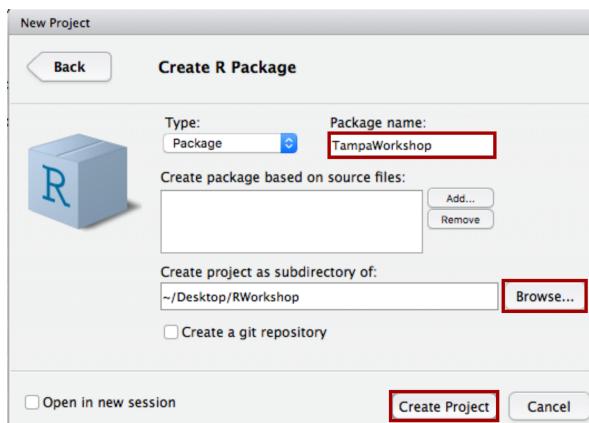
- b. Select New Directory.



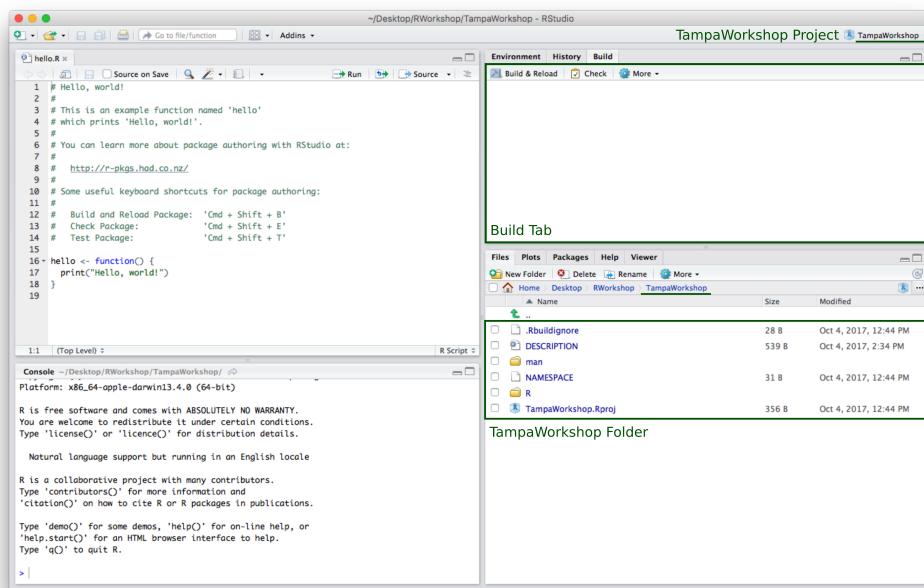
c. Select R Package.



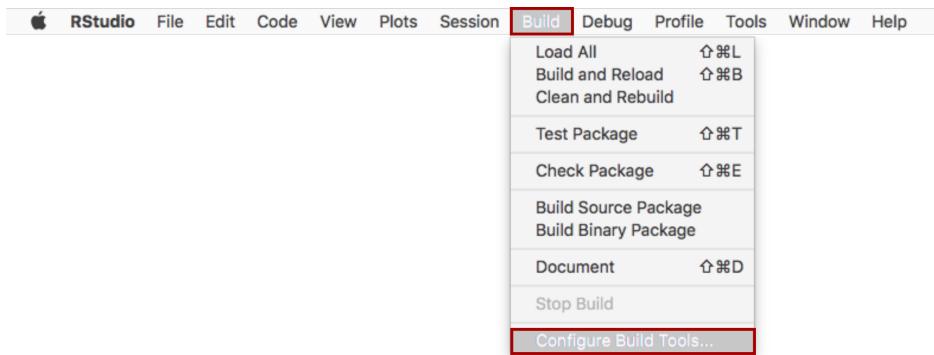
d. Name your package, select the directory in which your project/package will be located, and click on **Create Project**.



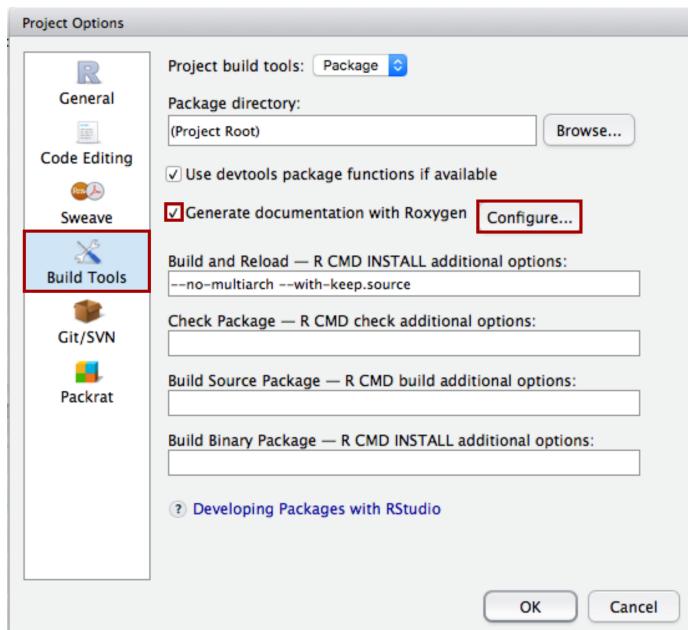
e. You should obtain the following screen.



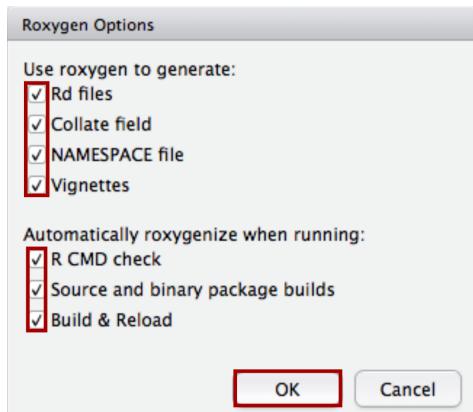
f. In the main drop-down menu click on Build and select Configure Build Tools.



g. In the Build Tools Tab, check the option Generate documentation with Roxygen. If this was already checked, click on Configure.

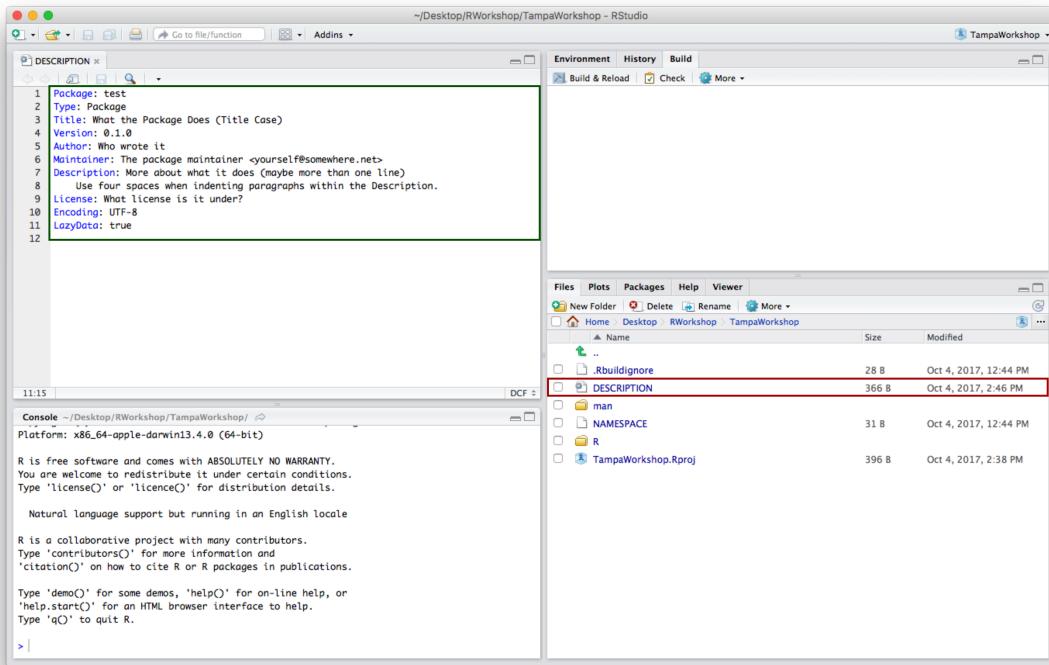


h. In the Roxygen Options window, check all options and click OK. Click OK again to exit the Project Options window.

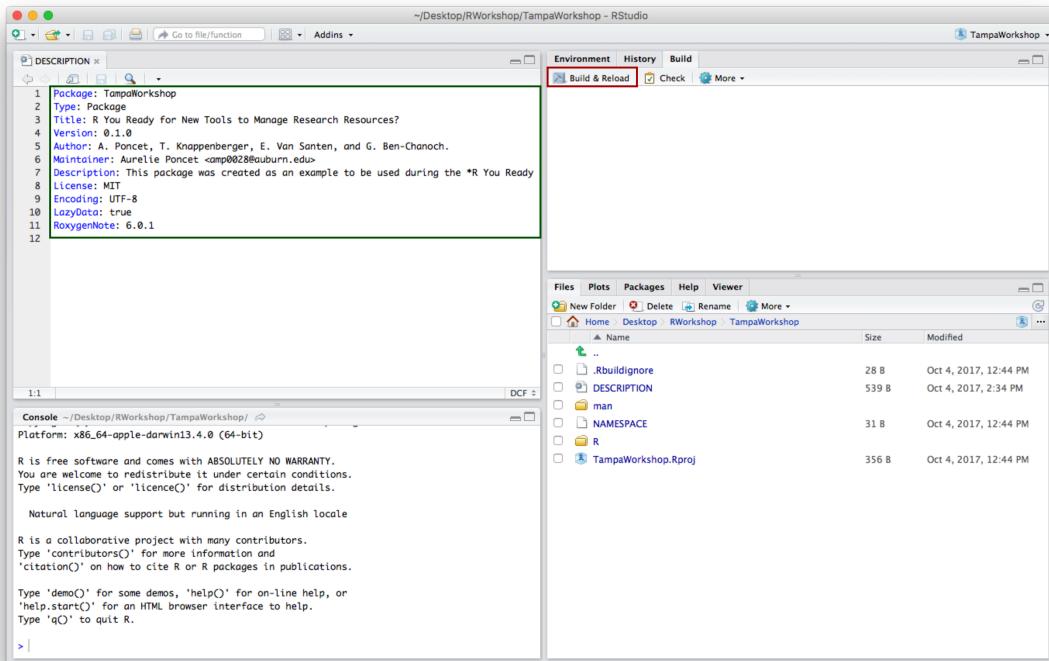


2. Complete the DESCRIPTION File.

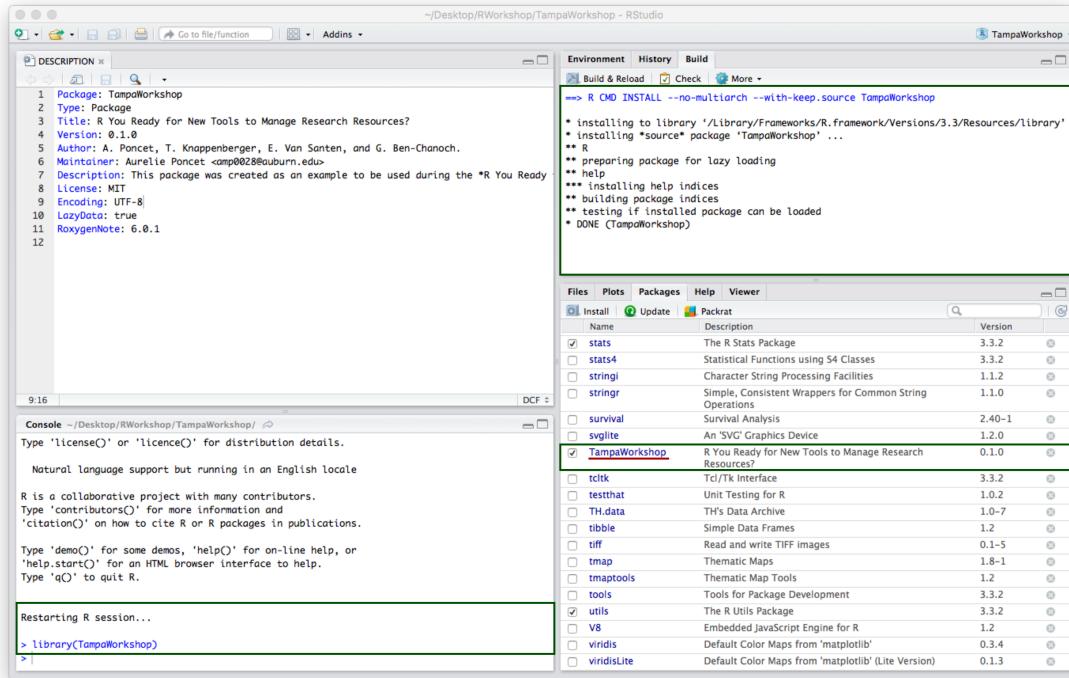
- Open the DESCRIPTION file from the file window in RStudio.



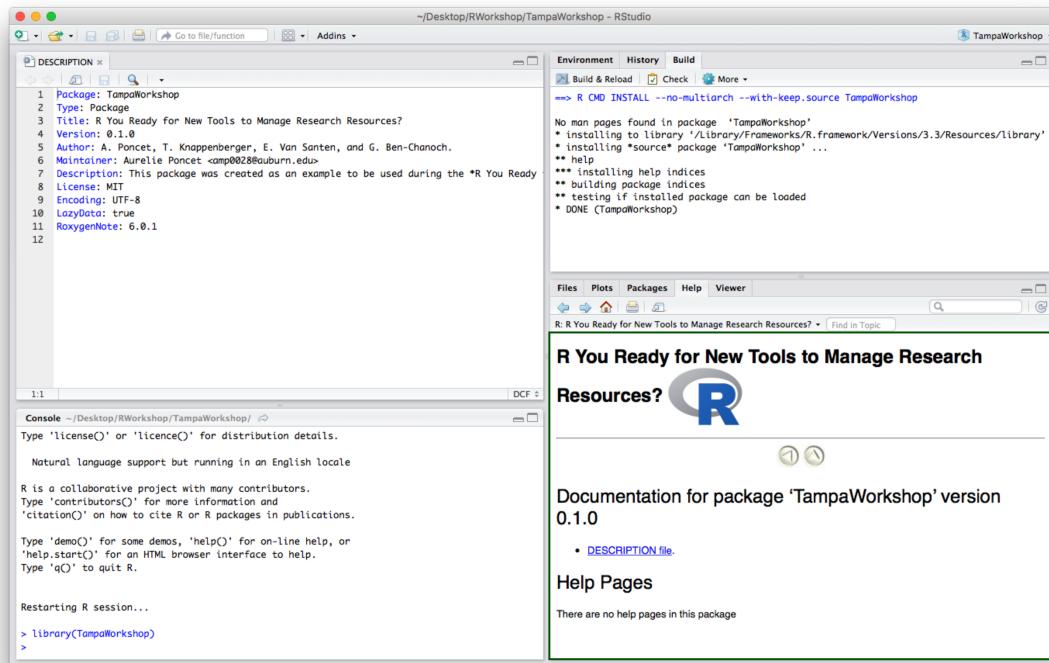
- Edit the file and click on Build & Reload or Install & Restart depending on which R and RStudio versions are installed on your computer.



- c. After clicking on Build & Reload (or Install & Restart), your package has been loaded to the Packages tab.

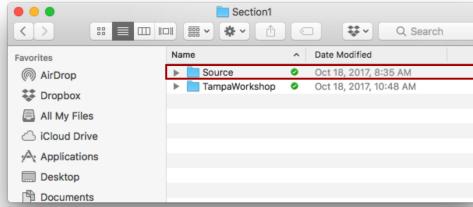


- d. Click on your package within the help screen to access its information.

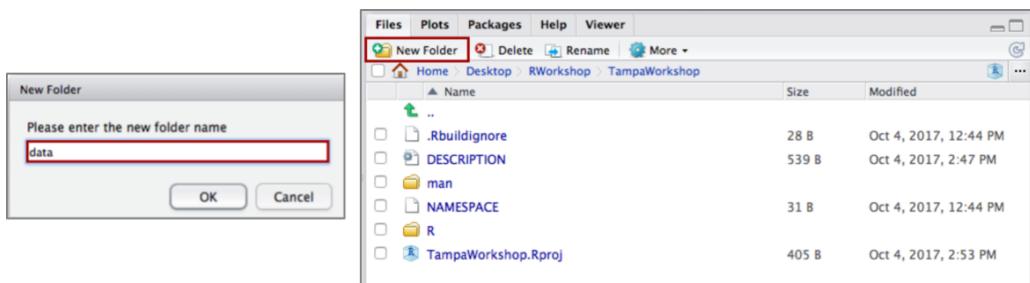


3. Add and Document a Dataset.

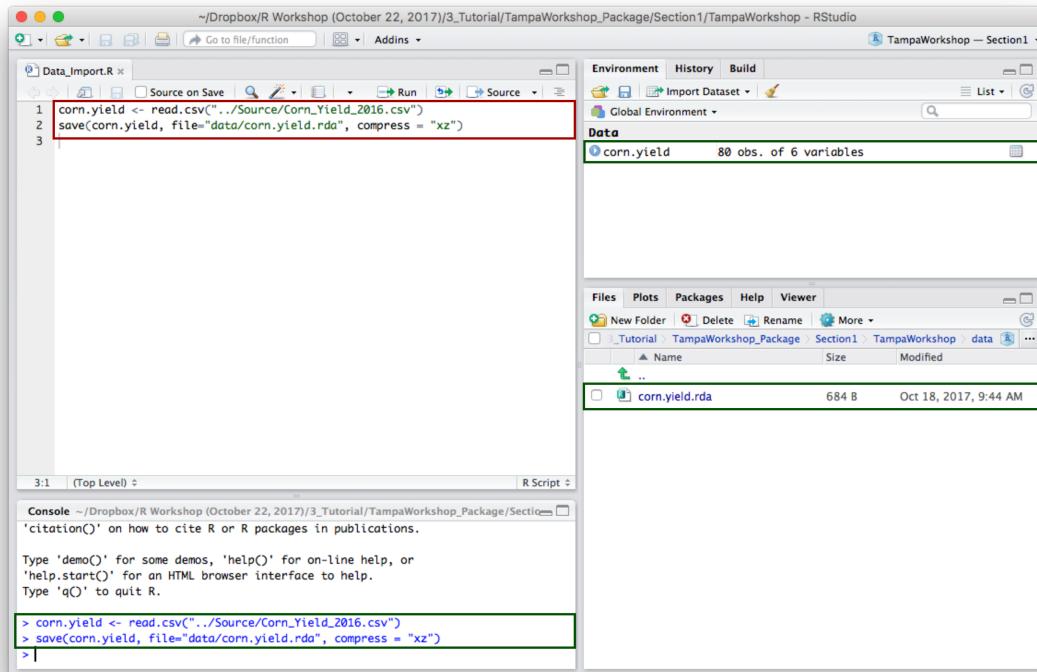
- Save the original data files (here .csv files) within a **Source** folder located outside of the **TampaWorkshop** package folder.



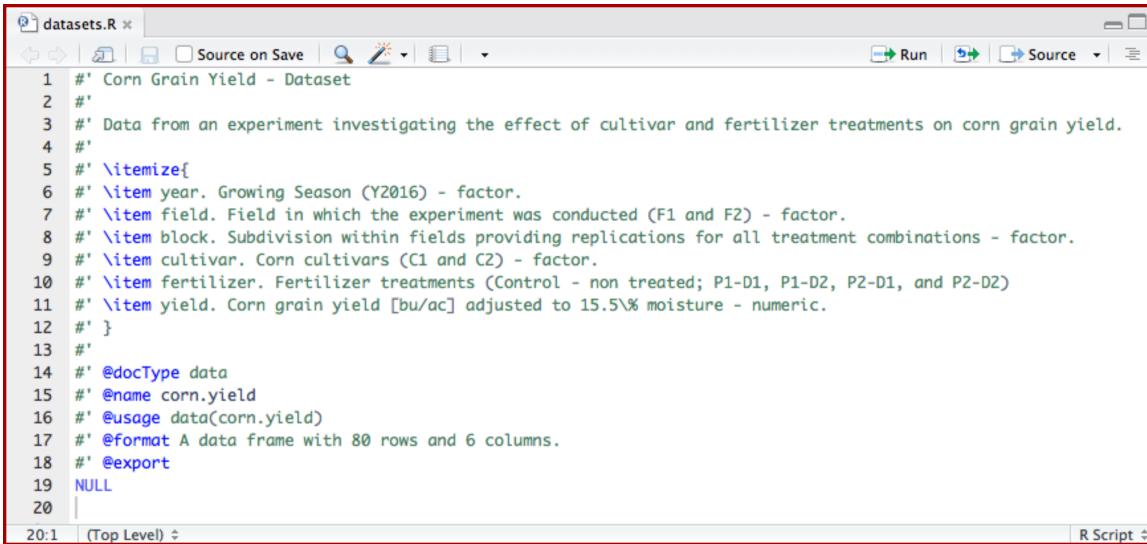
- Data within a R-package must be stored within a **data** folder. Hence, within the **TampaWorkshop** folder create a new folder named **data**.



- Import your data into RStudio. Save data as **.rda** into the data folder created step 3b. Save the script created to import the data within the **Source** package created step 3a.



d. Document the dataset. Start a new script and save within the package R folder. Metadata with `roxygen2` are added following `#'` (pound plus single quote). The first line must state the title of the dataset. The line below remains blank. The third line provides a short description of the dataset. Dataset individual columns are defined using `\itemize` and `\item`. The name of the dataset and how the dataset is imported in R are defined using respectively `\@name` and `\@usage`. `\@export` is required to ensure that the dataset will be available after exportation.

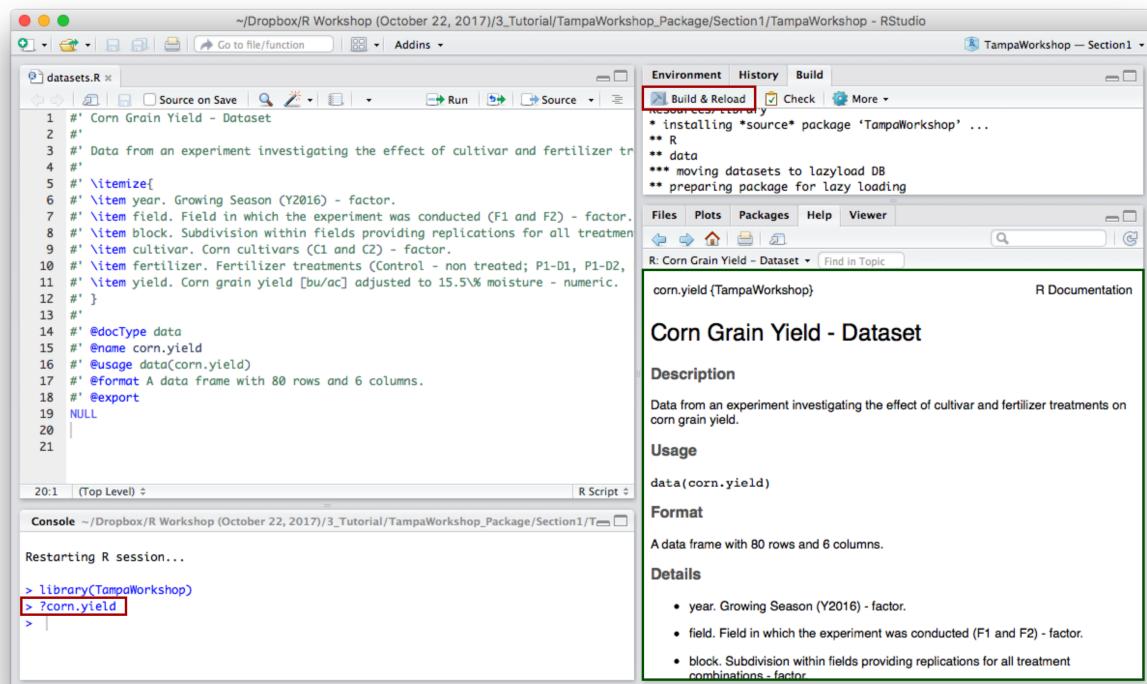


```

datasets.R ✘
Source on Save | Run | Source | ▾
1 #' Corn Grain Yield - Dataset
2 #
3 #' Data from an experiment investigating the effect of cultivar and fertilizer treatments on corn grain yield.
4 #
5 #' \itemize{
6 #'   \item year. Growing Season (Y2016) - factor.
7 #'   \item field. Field in which the experiment was conducted (F1 and F2) - factor.
8 #'   \item block. Subdivision within fields providing replications for all treatment combinations - factor.
9 #'   \item cultivar. Corn cultivars (C1 and C2) - factor.
10 #'   \item fertilizer. Fertilizer treatments (Control - non treated; P1-D1, P1-D2, P2-D1, and P2-D2)
11 #'   \item yield. Corn grain yield [bu/ac] adjusted to 15.5% moisture - numeric.
12 #' }
13 #
14 #' @docType data
15 #' @name corn.yield
16 #' @usage data(corn.yield)
17 #' @format A data frame with 80 rows and 6 columns.
18 #' @export
19 NULL
20
20:1 (Top Level) ▾ R Script

```

e. Click on Build & Reload - or Install & Restart - to update your package in the Packages tab. Use `?dataset.name` (here `?corn.yield`) to see the dataset metadata in the help screen.



The screenshot shows the RStudio interface with the `datasets.R` script open in the left pane. In the right pane, the help documentation for the `corn.yield` dataset is displayed. The `Build` tab in the top navigation bar is selected, showing the build process: "Installing *source* package 'TampaWorkshop' ...". The help page itself has the following content:

Corn Grain Yield - Dataset

Description
Data from an experiment investigating the effect of cultivar and fertilizer treatments on corn grain yield.

Usage
`data(corn.yield)`

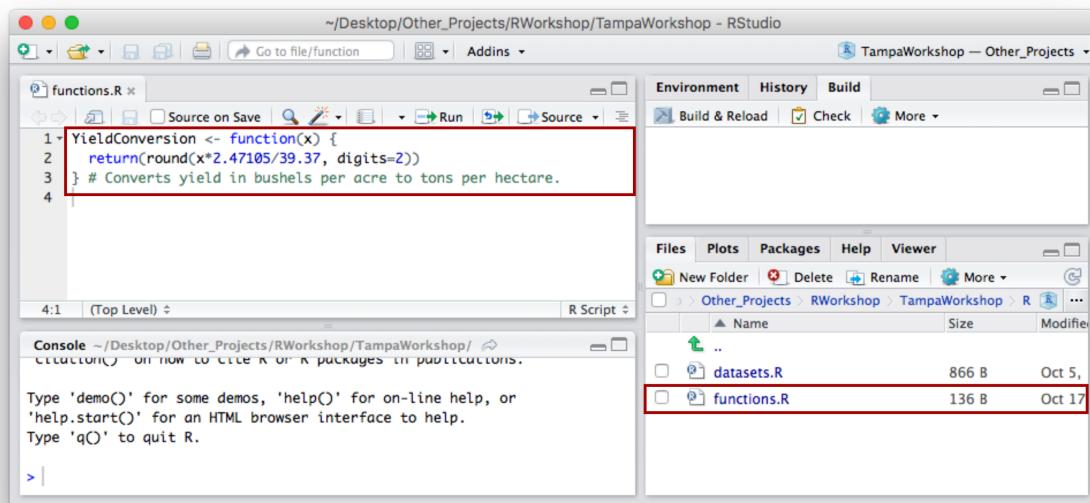
Format
A data frame with 80 rows and 6 columns.

Details

- year. Growing Season (Y2016) - factor.
- field. Field in which the experiment was conducted (F1 and F2) - factor.
- block. Subdivision within fields providing replications for all treatment combinations - factor.

4. Add and Document a Function.

- Create a new script and save into the TampaWorkshop package R folder. Write your function.



- Add meta data to the function. As we have seen before, metadata are specified following `#'`. The first line must state the title of the function. The line below remains blank. The third line provides a short description of the function. The function parameters and output are defined using `\@param` and `\@return`. Examples are provided below `\@example`. `\@export` is required to ensure that the dataset will be available after exportation.

The screenshot shows the RStudio interface with the same project and file. The code in "functions.R" now includes documentation and metadata:1 #' Convert Corn Yield Data from US to SI Units
2 #'
3 #' Converts corn yield data from bushels per acre to tons per hectare.
4 #' @param x unique numeric value or list of values to be converted into.
5 #' @return Converted numeric value or list of values.
6 #' @examples
7 #' YieldConversion(150)
8 #' YieldConversion(seq(from = 100, to = 200, by = 10))
9 #' YieldConversion(data\$yield)
10 #' @export
11 #'
12 YieldConversion <- function(x) {
13 return(round(x*2.47105/39.37, digits=2))
14 } # Converts yield in bushels per acre to tons per hectare.
15

- c. Click on Build & Reload - or Install & Restart - to update your package in the Packages tab. Use ?FunctionName (here YieldConversion) to see the function metadata in the help screen.

The screenshot shows the RStudio interface with the following components:

- Top Panel:** Shows the file path: ~/Dropbox/R Workshop (October 22, 2017)/TampaWorkshop - RStudio.
- Code Editor:** Displays the contents of functions.R. A red box highlights the "Build & Reload" button in the toolbar.
- Console:** Shows the command `library(TampaWorkshop)` followed by `?YieldConversion`.
- Output:** Shows the results of the R CMD INSTALL command, indicating the package is being built and installed.
- Help Viewer:** Shows the help documentation for YieldConversion, including its description, usage, arguments, value, and examples.

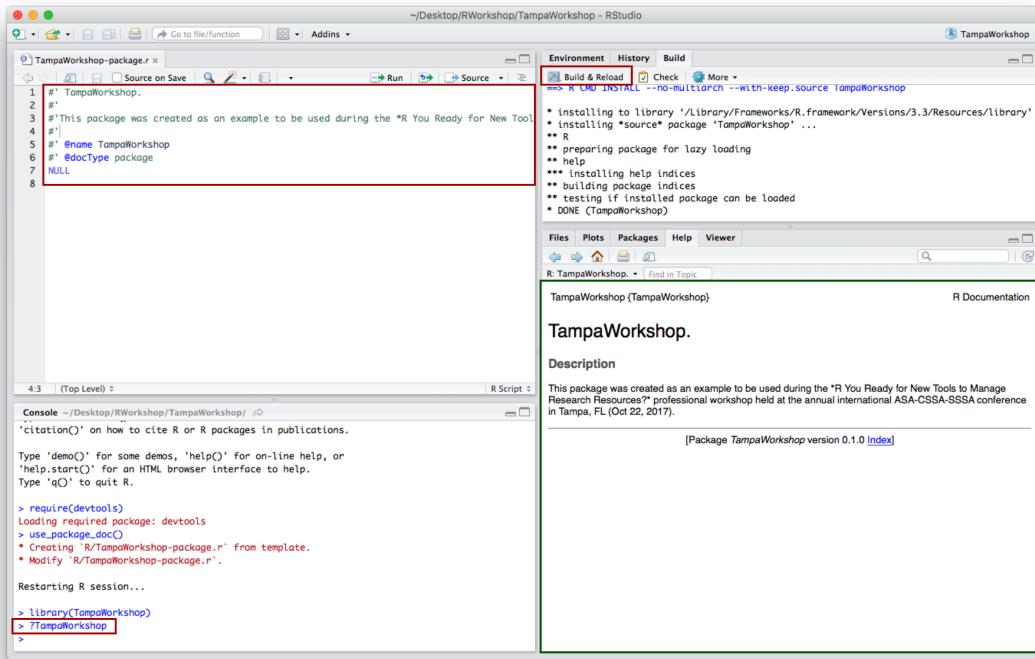
5. Document and Build the Package.

- a. In console, import `devtools` and execute `use_package_doc()`. This creates a new file in the package R directory.

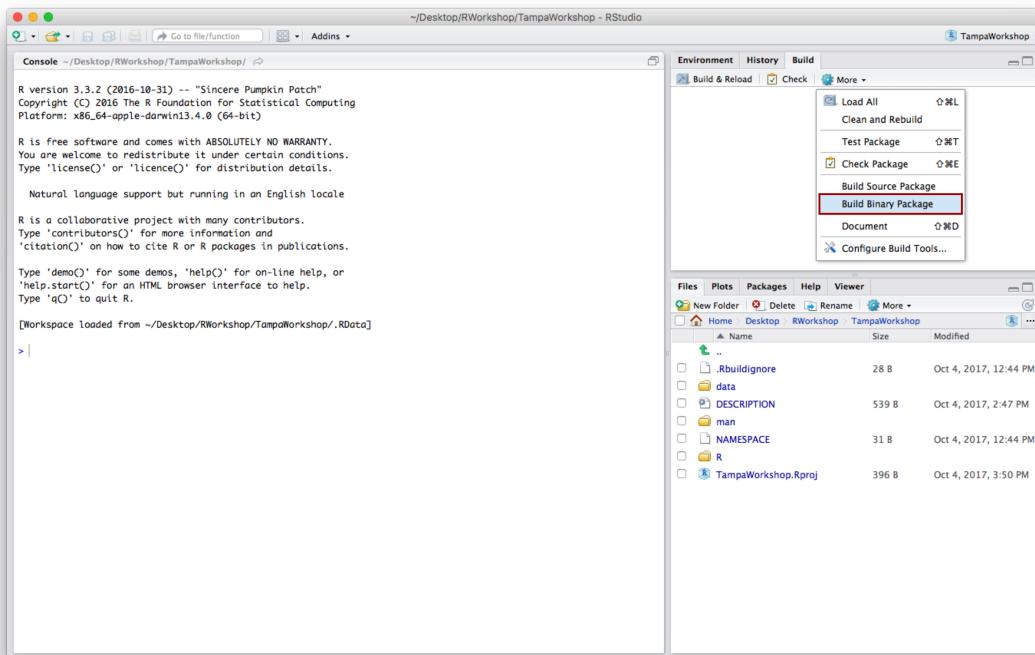
The screenshot shows the RStudio interface with the following components:

- Top Panel:** Shows the file path: ~/Desktop/RWorkshop/TampaWorkshop - RStudio.
- Code Editor:** Displays the contents of TampaWorkshop-package.R.
- Console:** Shows the command `require(devtools)` followed by `use_package_doc()`. A red box highlights the command.
- Output:** Shows the results of the command, including the creation of `R/TampaWorkshop-package.r` and modifications to `R/TampaWorkshop-package.r`.
- File Explorer:** Shows the package structure with files: datasets.R, functions.R, and TampaWorkshop-package.r.

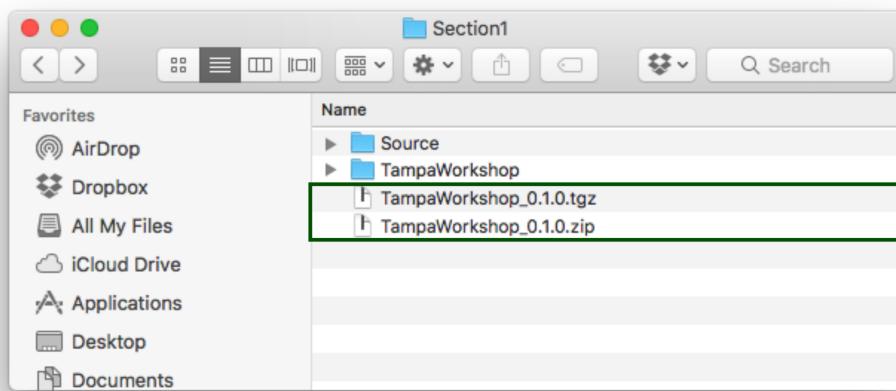
- b. You can edit the created file to document the package as a whole. Click on Build \& Reload - or Install & Restart - to update your package in the Packages tab. Use ?PackageName (here ?TampaWorkshop) to see the package metadata in the help screen.



- c. Once you have included and adequately documented all datasets and functions in your package, you can export your package as follow. In the Build tab, click on More and select Build Binary Package



- d. Your package has now been exported into a .zip package if you are using Windows - or into a .tgz file if you are using Mac - and can be found in the same directory than your original package folder.

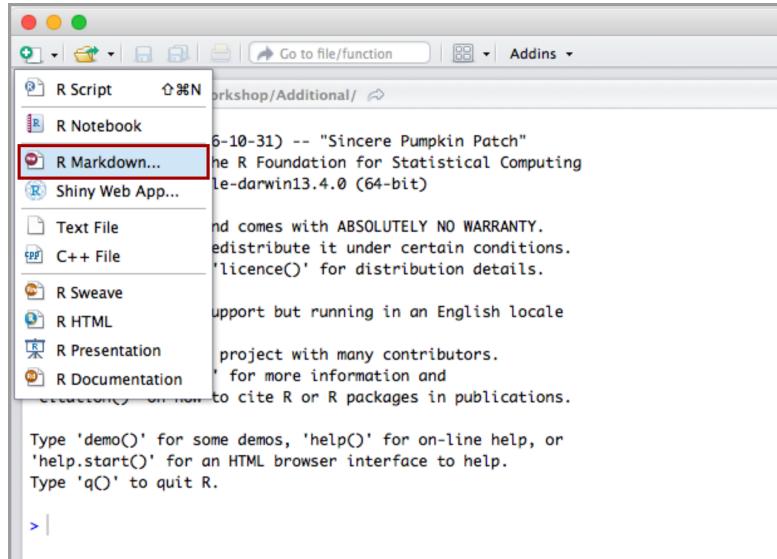


Section 2. Producing a Manuscript with R Markdown

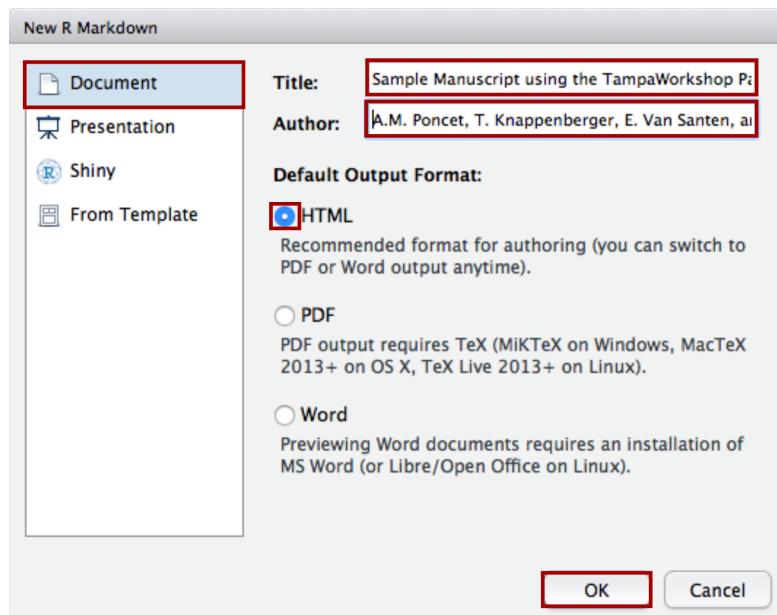
In this second section you will learn how to use R Markdown to integrate data visualization, data analysis, and manuscript/report writing into a single software application. You will be guided through a step by step tutorial that uses the R-package created in Section 1.

1. Create a New Markdown Document

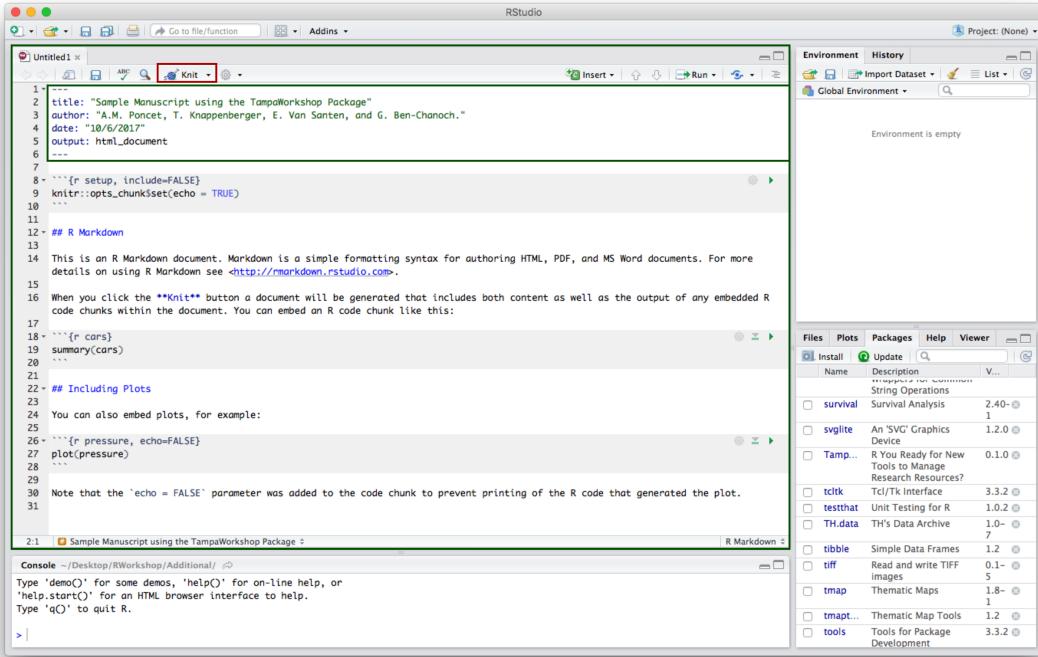
- a. Click on File >> New File >> R Markdown to open a new Markdown document.



- b. In the New R Markdown window select the type of document you want to create -here Document- , specify a title - optional, can be changed later -, the author(s), and an output format - for now select HTML. Click ok to continue.



- c. You obtain the following screen. The first four lines define the title of the document, authors, creation date, and output format. You can generate the **HTML** document by clicking on **Knit** button.



- d. You obtain the following result.

Sample_Manuscript.html | Open in Browser | Find | Publish

Sample Manuscript using the TampaWorkshop Package

A.M. Poncet, T. Knappenberger, E. Van Santen, and G. Ben-Chanoch.

10/6/2017

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:



2. Write the Manuscript.

- a. You can structure the document as presented in the figure below. Use # for first level headers and ## for second level headers.

The screenshot shows the RStudio interface with a file named "Sample_Manuscript.Rmd". The code editor contains the following R Markdown code:

```
9 knitr::opts_chunk$set(echo = TRUE)
10 ``
11 
12 # Abstract
13 You can write your abstract here.
14 
15 # Introduction
16 You can write your introduction here.
17 
18 
19 # Material and Methods
20 
21 ## Site Description
22 This is the first subsection within Material and Methods.
23 
24 ## Experimental Design
25 This is your second subsection within Material and Methods.
26 
27 ## Data Analysis
28 This is your third subsection within Material and Methods.
29 
30 # Results and Discussion
31 You can discuss your results here.
32 
33 # Conclusions
34 You can state your conclusions here.
35 
36 # References
37 Your references would go here.
38
```

The code is highlighted in blue, and the RStudio toolbar at the top includes icons for file operations, search, and knit.

- b. Leave a blank line to start a new paragraph. Use * and *** to italicize and bold text, respectively. Use \ as an escape character.

The screenshot shows the RStudio interface with the same "Sample_Manuscript.Rmd" file. A red box highlights the following text in the code:

```
11 
12 # Abstract
13 You can write your abstract here.
14 
15 # Introduction
16 
17 Leave a blank line to start a new paragraph.
18 
19 *You can italicize text this way.* **You can bold text this way**
20 
21 Use \\ as escape character. An escape character is placed in front of characters that have a different meaning for the programming language. For instance, \\ would be used here in front of \\\\", \*, \%.
22 
23 # Material and Methods
24 
25 ## Site Description
26 This is the first subsection within Material and Methods.
27 
28 ## Experimental Design
29 This is your second subsection within Material and Methods.
30 
31 ## Data Analysis
32 This is your third subsection within Material and Methods.
33 
34 # Results and Discussion
35 You can discuss your results here.
36 
37 # Conclusions
38 You can state your conclusions here.
39
```

The highlighted text includes a blank line, italicized text (*You can italicize text this way.*), bolded text (**You can bold text this way**), and an escape character example (Use \\ as escape character).

c. You can create non-ordered and order lists as presented in the figure below.

```
Sample_Manuscript.Rmd x
34 # Results and Discussion
35 You can discuss your results here.
36
37 # Conclusions
38
39 This is how you create a non-ordered list in markdown.
40
41 * first item in first order list.
42   + First item in second order list below the first item.
43   + Second item in second order list below the first item.
44 * second item in first order list.
45
46 This is how you create an ordered list in markdown.
47
48 1. first item in first order list.
49   a. First item in second order list below the first item.
50   b. Second item in second order list below the first item.
51 2. second item in first order list.
52
53 Make sure that:
54
55 * You have an empty line between the text and your list.
56 * First order lists are indented with 1 tab or 4 spaces.
57   + Second order lists are indented with 2 tabs ...
58   + ... or 8 spaces.
59
60
61 # References
62 Your references would go here.
63
```

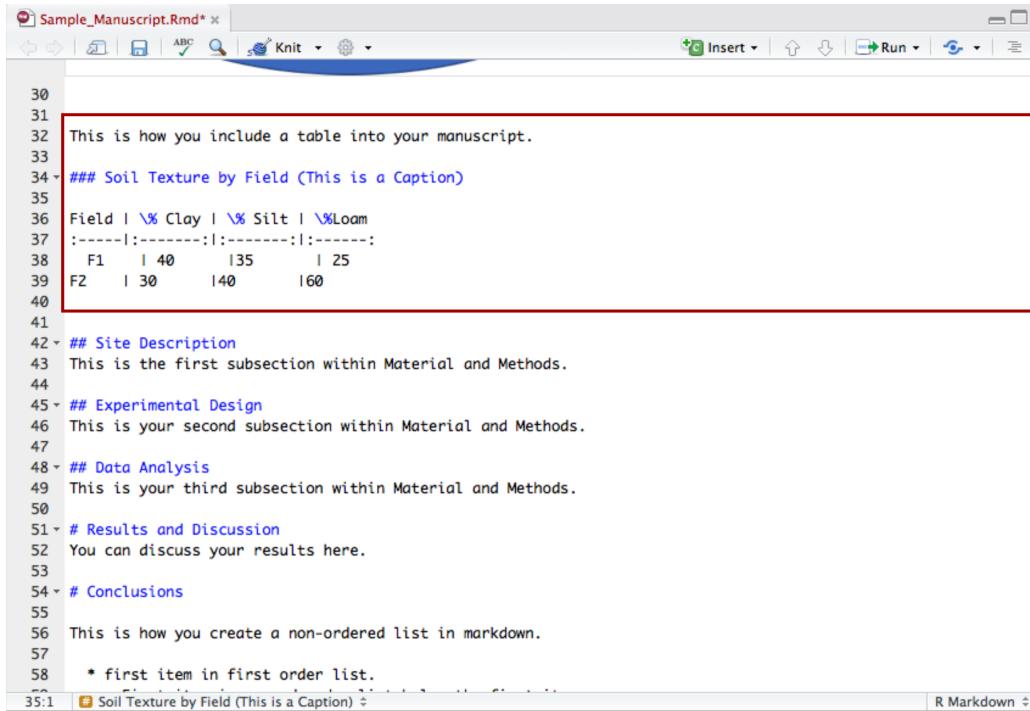
d. You can add figures using `![caption]{Figure Path and Name}`

```
Sample_Manuscript.Rmd x
25 # Material and Methods
26
27 This is how you insert a figure.
28
29 !["Sample Figure (This is a Caption)"]("Sample_Figure.png")
```



```
30
31
32 ## Site Description
33 This is the first subsection within Material and Methods.
34
35 ## Experimental Design
36 This is your second subsection within Material and Methods.
37
```

- e. You can also add tables as presented in the figure below. Use \ | to separate columns. Use - to separate header from rows below. Use : to determine text adjustment within individual columns. Use :--- to left-justify, :---: to center, and ---: to right-justify.



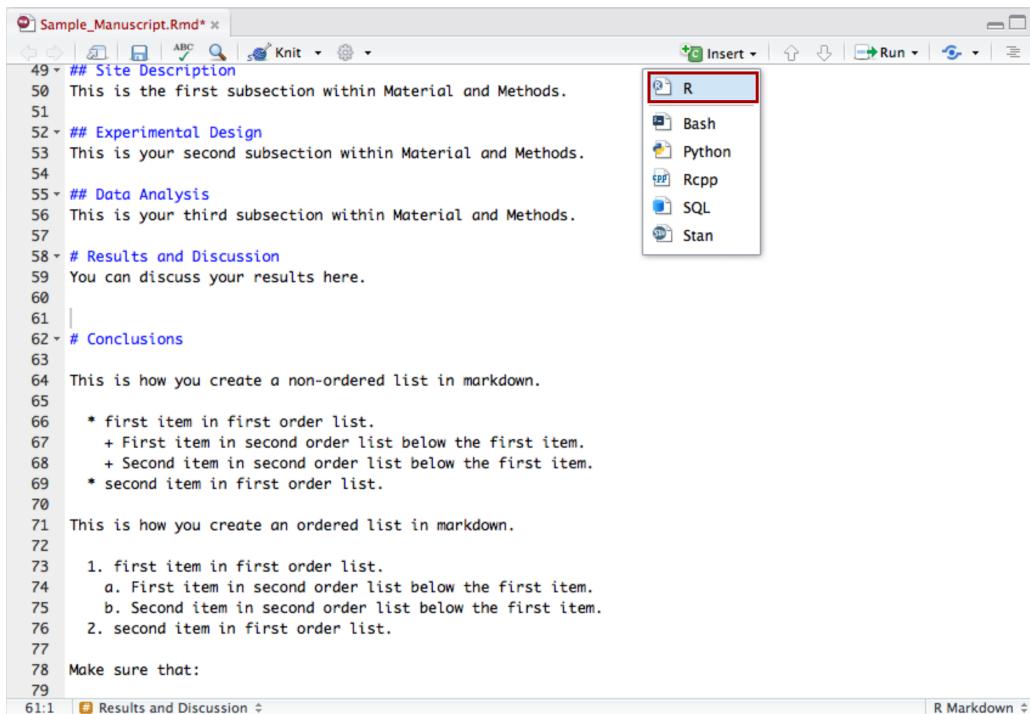
```

30
31
32 This is how you include a table into your manuscript.
33
34 ### Soil Texture by Field (This is a Caption)
35
36 Field | % Clay | % Silt | %Loam
37 :-----|:-----|:-----|
38   F1   | 40      135      | 25
39   F2   | 30      140      | 160
40
41
42 ## Site Description
43 This is the first subsection within Material and Methods.
44
45 ## Experimental Design
46 This is your second subsection within Material and Methods.
47
48 ## Data Analysis
49 This is your third subsection within Material and Methods.
50
51 # Results and Discussion
52 You can discuss your results here.
53
54 # Conclusions
55
56 This is how you create a non-ordered list in markdown.
57
58 * first item in first order list.

```

3. Include R Code and Outputs to the Manuscript.

- a. To insert R chunks within Markdown, click on Insert and select R.

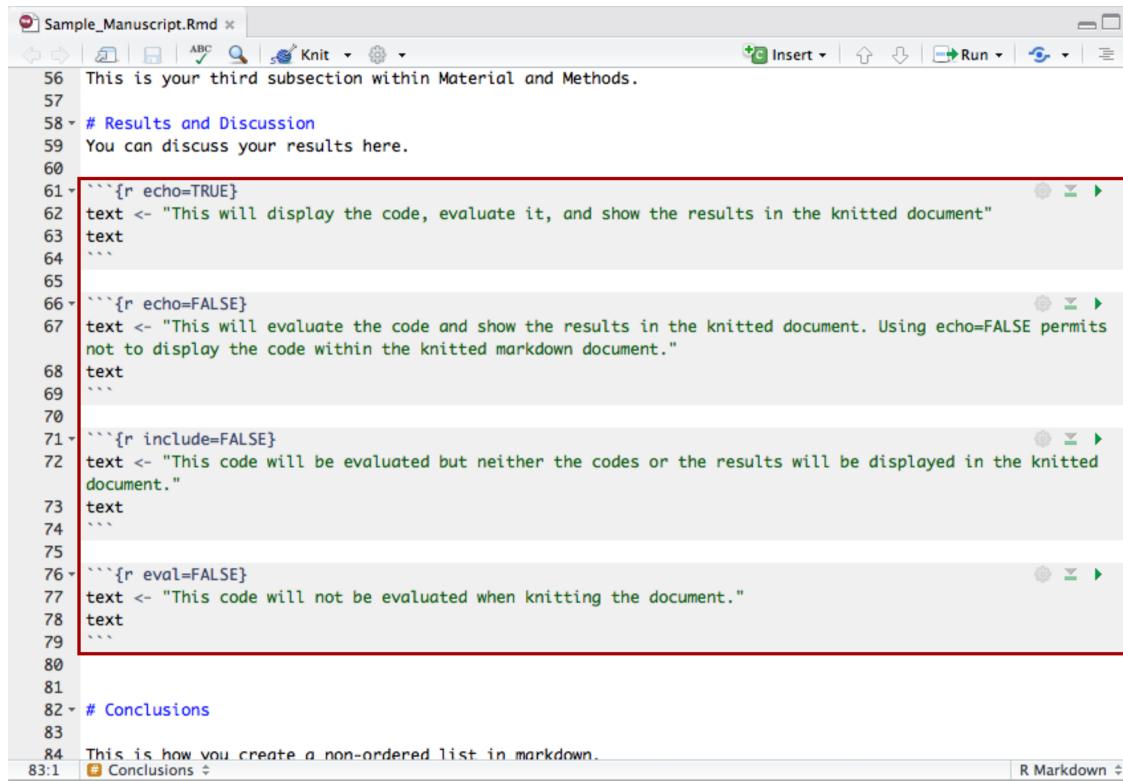


```

49 ## Site Description
50 This is the first subsection within Material and Methods.
51
52 ## Experimental Design
53 This is your second subsection within Material and Methods.
54
55 ## Data Analysis
56 This is your third subsection within Material and Methods.
57
58 # Results and Discussion
59 You can discuss your results here.
60
61
62 # Conclusions
63
64 This is how you create a non-ordered list in markdown.
65
66 * first item in first order list.
67   + First item in second order list below the first item.
68   + Second item in second order list below the first item.
69 * second item in first order list.
70
71 This is how you create an ordered list in markdown.
72
73 1. first item in first order list.
74    a. First item in second order list below the first item.
75    b. Second item in second order list below the first item.
76 2. second item in first order list.
77
78 Make sure that:

```

- b. You can specify if the code should be displayed, evaluated, and the results displayed within the R document using the `echo = T/F`, `include = T/F`, and `eval = T/F` parameters.

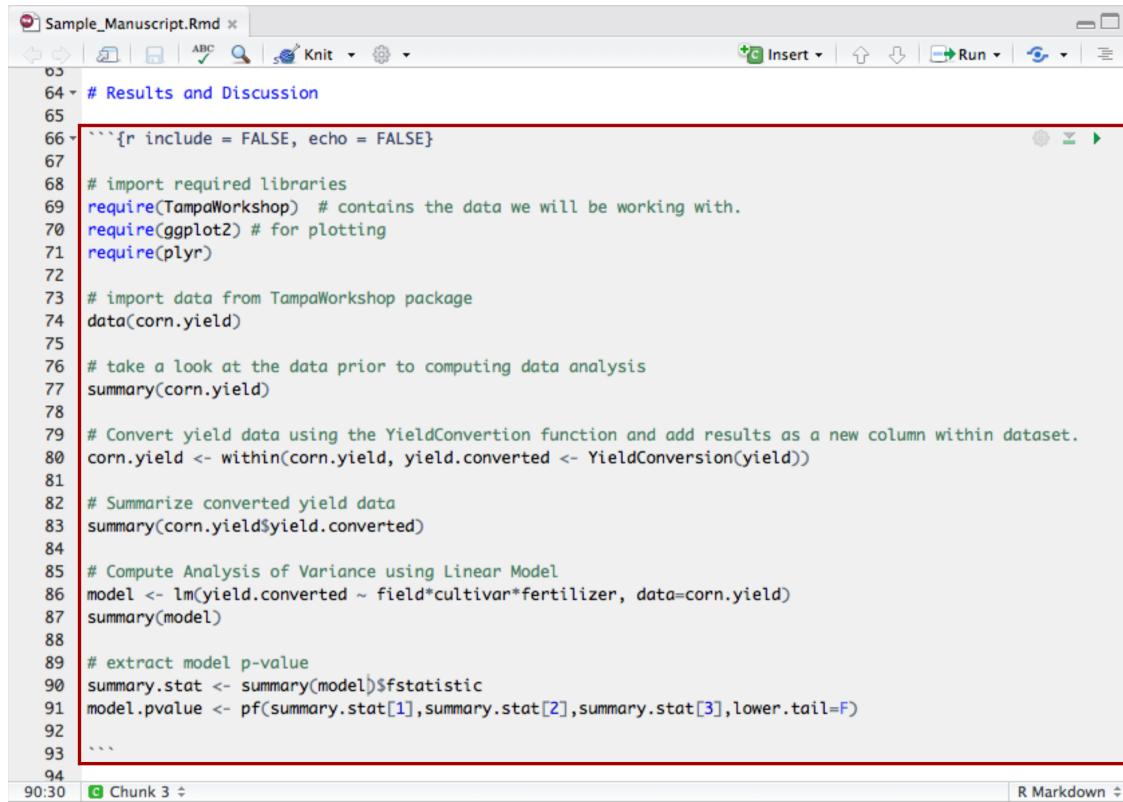


```

56 This is your third subsection within Material and Methods.
57
58 # Results and Discussion
59 You can discuss your results here.
60
61 ```{r echo=TRUE}
62 text <- "This will display the code, evaluate it, and show the results in the knitted document"
63 text
64 ```
65
66 ```{r echo=FALSE}
67 text <- "This will evaluate the code and show the results in the knitted document. Using echo=FALSE permits
not to display the code within the knitted markdown document."
68 text
69 ```
70
71 ```{r include=FALSE}
72 text <- "This code will be evaluated but neither the codes or the results will be displayed in the knitted
document."
73 text
74 ```
75
76 ```{r eval=FALSE}
77 text <- "This code will not be evaluated when knitting the document."
78 text
79 ```
80
81
82 # Conclusions
83
84 This is how you create a non-ordered list in markdown.
85
86 Conclusions

```

- c. You can then type your code as you would in a R script.

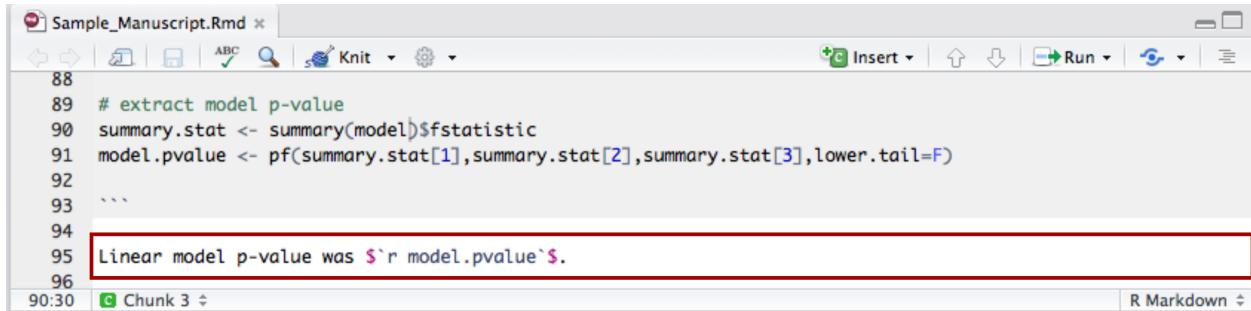


```

63
64 # Results and Discussion
65
66 ```{r include = FALSE, echo = FALSE}
67
68 # import required libraries
69 require(TampaWorkshop) # contains the data we will be working with.
70 require(ggplot2) # for plotting
71 require(plyr)
72
73 # import data from TampaWorkshop package
74 data(corn.yield)
75
76 # take a look at the data prior to computing data analysis
77 summary(corn.yield)
78
79 # Convert yield data using the YieldConversion function and add results as a new column within dataset.
80 corn.yield <- within(corn.yield, yield.converted <- YieldConversion(yield))
81
82 # Summarize converted yield data
83 summary(corn.yield$yield.converted)
84
85 # Compute Analysis of Variance using Linear Model
86 model <- lm(yield.converted ~ field*cultivar*fertilizer, data=corn.yield)
87 summary(model)
88
89 # extract model p-value
90 summary.stat <- summary(model)$fstatistic
91 model.pvalue <- pf(summary.stat[1],summary.stat[2],summary.stat[3],lower.tail=F)
92
93
94

```

d. It is also possible to inline some R code using `\(R code here)` ...

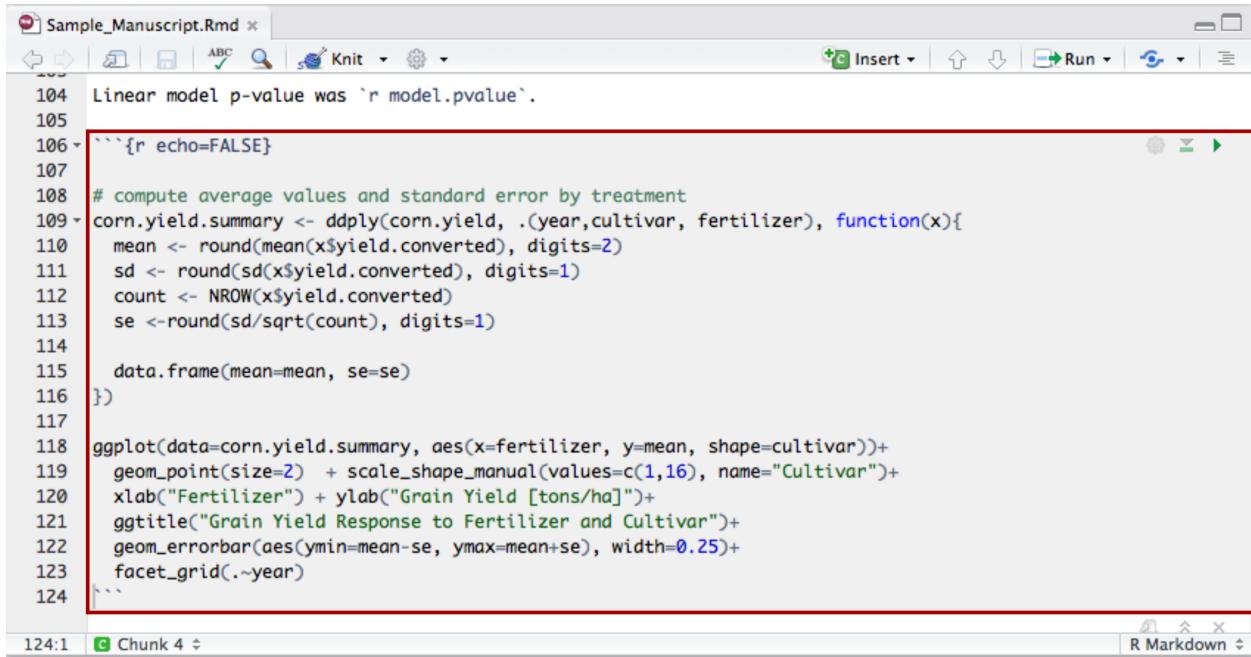


The screenshot shows the RStudio interface with a file named "Sample_Manuscript.Rmd". The code editor displays the following R code:

```
88 # extract model p-value
89 summary.stat <- summary(model)$fstatistic
90 model.pvalue <- pf(summary.stat[1],summary.stat[2],summary.stat[3],lower.tail=F)
91
92 ``
93
94
95 Linear model p-value was `r model.pvalue`.
96
```

The output "Linear model p-value was `r model.pvalue`" is highlighted with a red box. The status bar at the bottom indicates "90:30" and "Chunk 3".

e. ... and insert graphics to the knitted document.



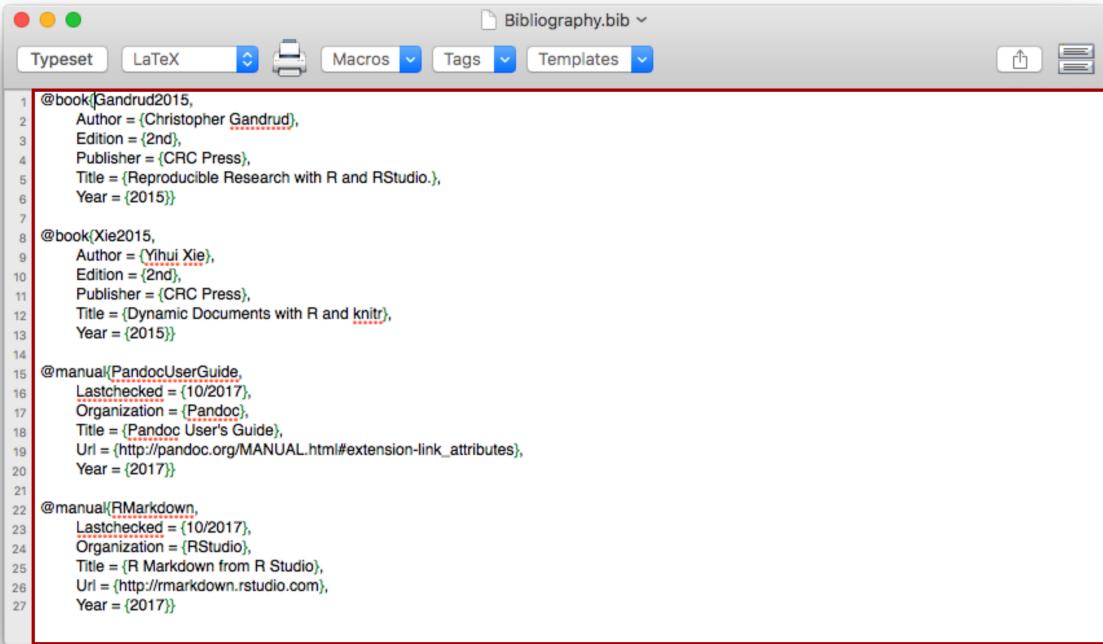
The screenshot shows the RStudio interface with a file named "Sample_Manuscript.Rmd". The code editor displays the following R code:

```
104 Linear model p-value was `r model.pvalue`.
105
106 ```{r echo=FALSE}
107
108 # compute average values and standard error by treatment
109 corn.yield.summary <- ddply(corn.yield, .(year,cultivar, fertilizer), function(x){
110   mean <- round(mean(x$yield.converted), digits=2)
111   sd <- round(sd(x$yield.converted), digits=1)
112   count <- NROW(x$yield.converted)
113   se <- round(sd/sqrt(count), digits=1)
114
115   data.frame(mean=mean, se=se)
116 })
117
118 ggplot(data=corn.yield.summary, aes(x=fertilizer, y=mean, shape=cultivar))+ 
119   geom_point(size=2) + scale_shape_manual(values=c(1,16), name="Cultivar")+
120   xlab("Fertilizer") + ylab("Grain Yield [tons/ha]")+
121   ggtitle("Grain Yield Response to Fertilizer and Cultivar")+
122   geom_errorbar(aes(ymin=mean-se, ymax=mean+se), width=0.25)+ 
123   facet_grid(.~year)
124 ```

The code chunk is highlighted with a red box. The status bar at the bottom indicates "124:1" and "Chunk 4".
```

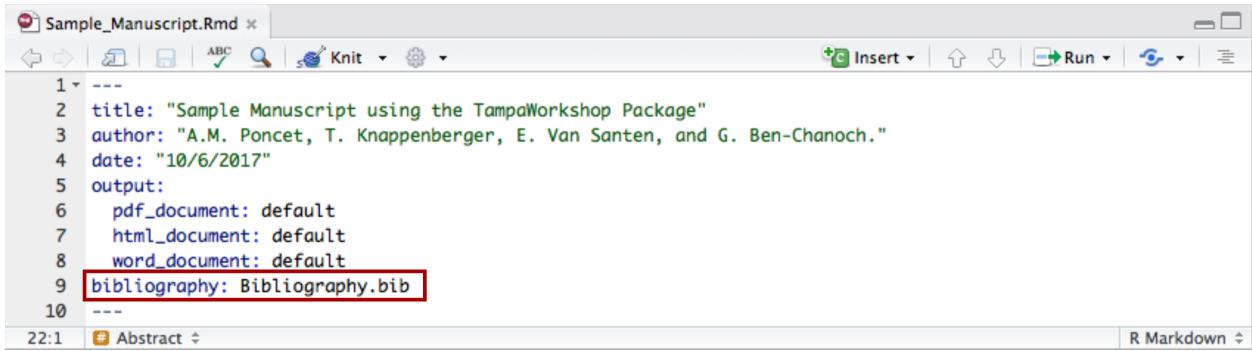
4. Insert References.

- List your references within a bibtex or biblatex file (.bib)



```
Bibliography.bib
Typeset LaTeX Macros Tags Templates
1 @book{Gandrud2015,
2   Author = {Christopher Gandrud},
3   Edition = {2nd},
4   Publisher = {CRC Press},
5   Title = {Reproducible Research with R and RStudio.},
6   Year = {2015}}
7
8 @book{Xie2015,
9   Author = {Yihui Xie},
10  Edition = {2nd},
11  Publisher = {CRC Press},
12  Title = {Dynamic Documents with R and knitr},
13  Year = {2015}}
14
15 @manual{PandocUserGuide,
16   Lastchecked = {10/2017},
17   Organization = {Pandoc},
18   Title = {Pandoc User's Guide},
19   Url = {http://pandoc.org/MANUAL.html#extension-link_attributes},
20   Year = {2017}}
21
22 @manual{RMarkdown,
23   Lastchecked = {10/2017},
24   Organization = {RStudio},
25   Title = {R Markdown from R Studio},
26   Url = {http://rmarkdown.rstudio.com},
27   Year = {2017}}
```

- Save that file and specify the file name and path within the Markdown file header.



```
Sample_Manuscript.Rmd
ABC Knit Insert Run Abstract R Markdown
1 ---
2 title: "Sample Manuscript using the TampaWorkshop Package"
3 author: "A.M. Poncet, T. Knappenberger, E. Van Santen, and G. Ben-Chanoch."
4 date: "10/6/2017"
5 output:
6   pdf_document: default
7   html_document: default
8   word_document: default
9 bibliography: Bibliography.bib
10 ---
```

- c. In-text citation go inside square brackets and are separated by a semicolon. Each citation must have a key composed of \@ plus the citation identifier from the database. A minus sign before the \@ will suppress mention of the author in the citation. This is useful when the author is already mentioned in the text.

```

20 # Abstract
21
22 This tutorial was written using the following resources.
23
24 - Books: [@Gandrud2015; @Xie2015]
25 - Websites: [@PandocUserGuide; @RMarkdown]
26
27 Use minus sign before the *@\* to remove the author from the citation. For instance the authors used
28 Gandrud's [-@Gandrud2015] book as a reference to write this manuscript.
29
30 # Introduction
32:1 | Abstract | R Markdown

```

- d. This provides the following result after knitting the document.

In text:

Abstract

This tutorial was written using the following resources.

- Books: (Gandrud 2015; Xie 2015)
- Websites: (*Pandoc User's Guide* 2017; *R Markdown from R Studio* 2017)

Use minus sign before the @ to remove the author from the citation. For instance the authors used Gandrud's (2015) book as a reference to write this manuscript.

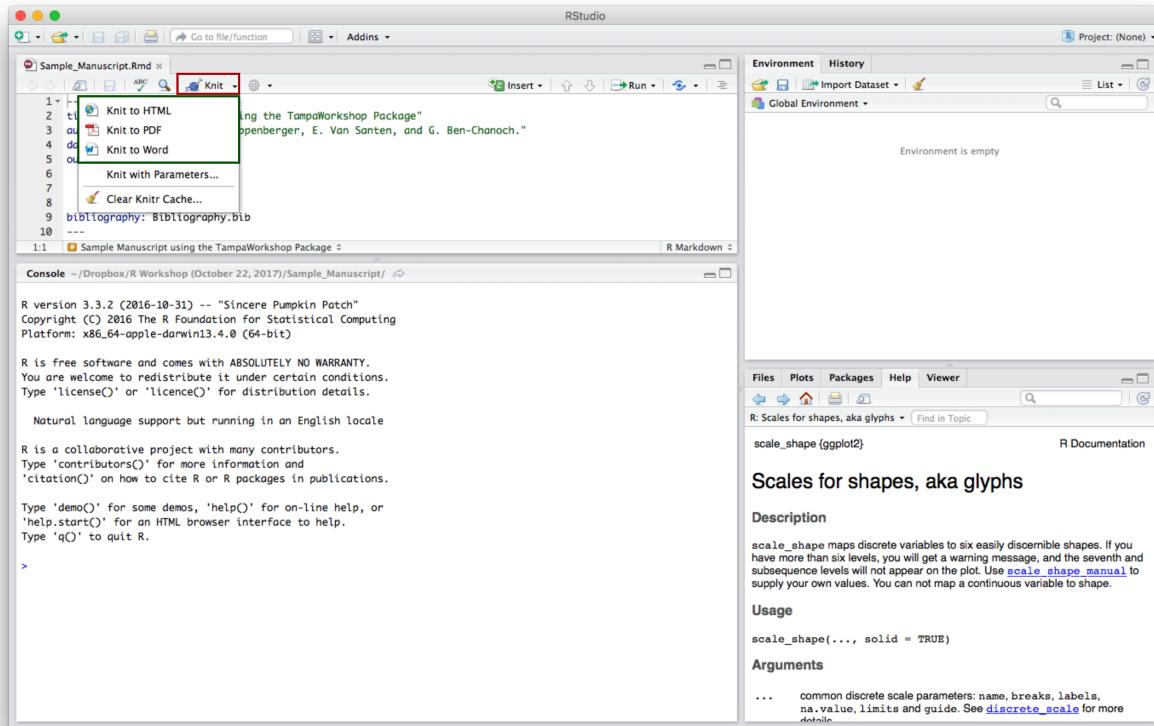
References Section:

References

- Gandrud, Christopher. 2015. *Reproducible Research with R and Rstudio*. 2nd ed. CRC Press.
Pandoc User's Guide. 2017. Pandoc. http://pandoc.org/MANUAL.html#extension-link_attributes.
R Markdown from R Studio. 2017. RStudio. <http://rmarkdown.rstudio.com>.
Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. CRC Press.

Section 3. Exploring Different Markdown Outlets

Once completed, you can export your Markdown document into different formats including HTML, Word, and PDF (through Latex). To do so, click on the Knit button and select the desired outlet. In order to export your Markdown document to PDF you need to have already installed one of Latex distributions.

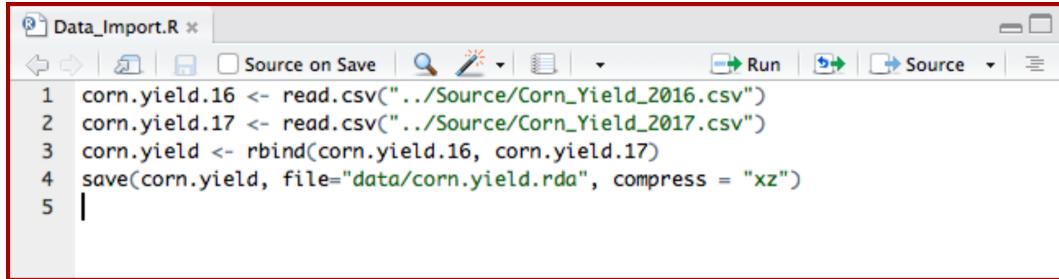


Section 4. Update R-Package and Manuscript.

This fourth and last section will demonstrate the benefits of using R and R Markdown as an integrated tool for research. You will first be guided to update the dataset included within the R-package created in Section 1. You will then knit again the manuscript produced in Section 2 and compare the outputs.

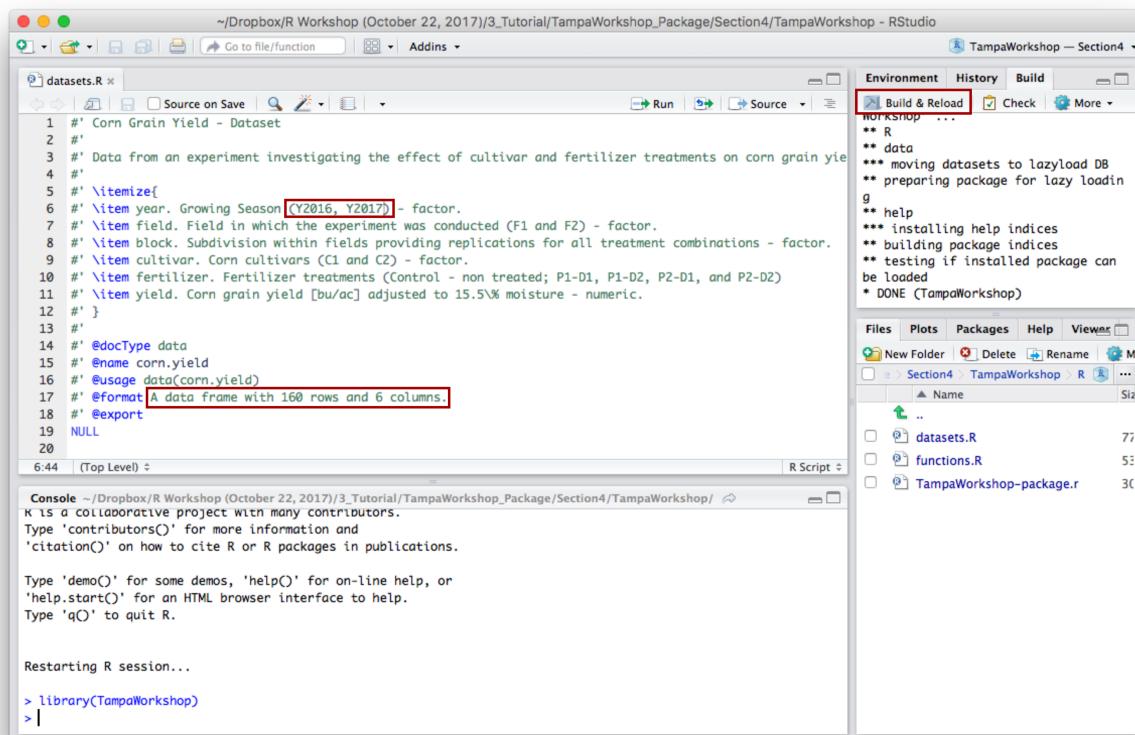
1. Update the R-Package

- Update the `corn.yield` dataset by adding the 2017 corn yield data.



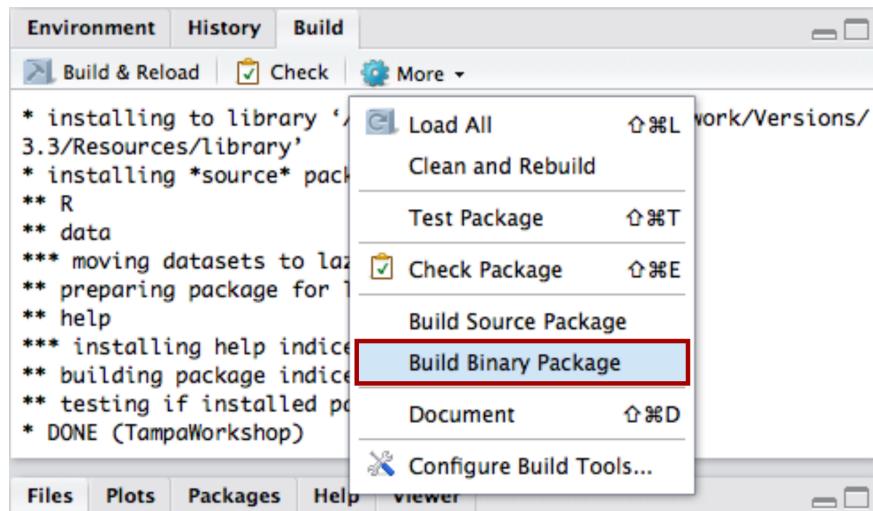
```
1 corn.yield.16 <- read.csv("../Source/Corn_Yield_2016.csv")
2 corn.yield.17 <- read.csv("../Source/Corn_Yield_2017.csv")
3 corn.yield <- rbind(corn.yield.16, corn.yield.17)
4 save(corn.yield, file="data/corn.yield.rda", compress = "xz")
5 |
```

- Open the `TampaWorkshop` package created in Section 1. Update the information provided in the `dataset.R` file and click on Build \& Reload - or Install & Restart.



The screenshot shows the RStudio interface with the `dataset.R` file open in the left pane. The code defines a dataset with various variables and their descriptions. In the middle pane, the `Build & Reload` button is highlighted. The right pane shows the `TampaWorkshop` package structure with files like `datasets.R`, `functions.R`, and `TampaWorkshop-package.r`. The console at the bottom shows the R startup message and the command `> library(TampaWorkshop)`.

c. In the Build tab, click on More and select Build Binary Package



d. Make sure the corn.yield dataset has been correctly updated.

```
Console ~ /Dropbox/R Workshop (October 22, 2017)/RPackage_Section4/TampaWorkshop/ ↵
> library(TampaWorkshop)
> data(corn.yield)
> unique(corn.yield$year)
[1] Y2016 Y2017
Levels: Y2016 Y2017
> |
```

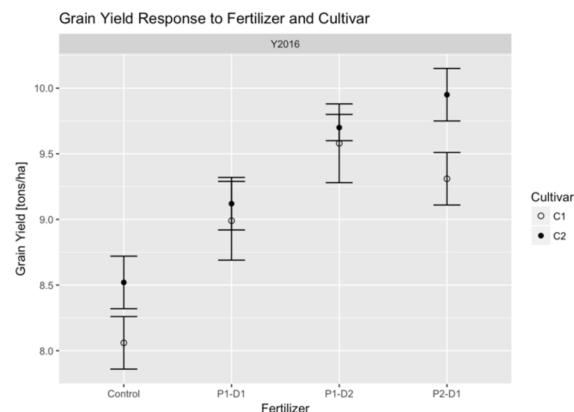
2. Update Manuscript

Knit the Markdown document created in section 2 and compare with previous version

Before Updating corn.yield.

Results and Discussion

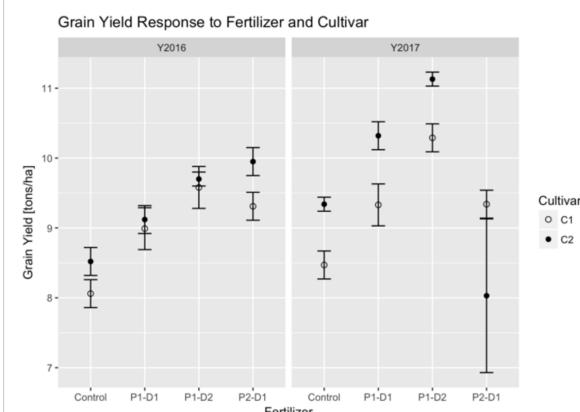
Linear model p-value was $2.2016117 \times 10^{-20}$.



After Updating corn.yield.

Results and Discussion

Linear model p-value was 5.5814812×10^{-9} .



References

- Allaire, JJ, Joe Cheng, Yihui Xie, Jonathan McPherson, Winston Chang, Jeff Allen, Hadley Wickham, Aron Atkins, and Rob Hyndman. 2016. *Rmarkdown: Dynamic Documents for R*. <https://CRAN.R-project.org/package=rmarkdown>.
- Gandrud, Christopher. 2015. *Reproducible Research with R and Rstudio*. 2nd ed. CRC Press.
- RStudio, and Inc. 2016. *Htmltools: Tools for Html*. <https://CRAN.R-project.org/package=htmltools>.
- Steve Dutky initial R port, S original by, extensions by Martin Maechler; revised, and modified by Steve Dutky. 2013. *Bitops: Bitwise Operations*. <https://CRAN.R-project.org/package=bitops>.
- Tuszynski, Jarek. 2014. *CaTools: Tools: Moving Window Statistics, Gif, Base64, Roc Auc, Etc*. <https://CRAN.R-project.org/package=caTools>.
- Urbanek, Simon. 2015. *Base64enc: Tools for Base64 Encoding*. <https://CRAN.R-project.org/package=base64enc>.
- Wickham, Hadley. 2009. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <http://ggplot2.org>.
- . 2011. “The Split-Apply-Combine Strategy for Data Analysis.” *Journal of Statistical Software* 40 (1): 1–29. <http://www.jstatsoft.org/v40/i01/>.
- Wickham, Hadley, and Winston Chang. 2017. *Devtools: Tools to Make Developing R Packages Easier*. <https://CRAN.R-project.org/package=devtools>.
- Wickham, Hadley, Peter Danenberg, and Manuel Eugster. 2017. *Roxygen2: In-Line Documentation for R*. <https://CRAN.R-project.org/package=roxygen2>.
- Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. CRC Press.