How to Compile your R Code in RMarkdown

Kenny Le

12/17/2020

Section 1: Background/Setting Up/Converting your R file to R-Markdown file

Background Rationale

RMarkdown is an application that can run and save R-code into a high-quality document like PDF, HTML, Word, to name a few. First, RMarkdown runs in the R-Studio's Integrated development environment (IDE), making it convenient to access. All you need to do is to download the package using the command "install.packages("rmarkdown"). RMarkdown can run code in different coding languages, including Python or SQL. RMarkdown saves the time to write the report within R-studio,instead of exporting graphs to word documents.

The R coding language can be used for DNA Sequence Statistics, Comparative Genomics, Identifying homologous genes between two species. R Markdown is capable of compiling all these data into a presentable format for any reports. This tutorial will help you set up your RMarkdown file; There will be various R language commands to go over the *Iris dataset* by illustrating the *Iris dataset* into a pairwise plot, scatterplot, linear regression, boxplot, histogram, and heat map in order to display this dataset into an HTML and PDF outputs while using RMarkdown.

Set up for Rmarkdown

Timestamp: starting in 3:18

1. Start RStudio and Open a "New Project"
2. Install RMarkdown

- Option 1: on the "console": code `install.packages("rmarkdown")`
- Option 2: on the upper tabs go to **Tools** next *Install packages* then search "rmarkdown" (default CRAN and install repositories)

3. Install TinyTex (to compile RMarkdown into a PDF document)

- Option 1: on the "console": code '`install.packages("tinytex") or tinytex::install_tinytex()`
- Option 2: on the upper tabs go to **Tools** next *Install packages* then search "tinytex" (default CRAN and install repositories)

4. Go to upper tabs **File**

- Start at "New File" next select *RMarkdown* then Name finally, select "HTML"

R file convert to RMarkdown file

1. Specify the code chunk with '#+'
2. Render the document as an RMarkdown file `knitr::spin("filename.R", knit = FALSE, format = "Rmd")`
3. This will generate an RMarkdown file called "filename.Rmd"
4. Knit this document, with your preferred output.

Text file (.txt) convert to Rmarkdown file

1. Save as ".Rmd" file
2. Now, you can use the various commands in RMarkdown

Section 2: Formatting and Knit your outputs

Timestamp: starting in 15:18

Formatting in RMarkdown:

1. For Headers, the amount of "#" characters corresponds to the size of Header it is.
2. Links, must contain "http://" prior to the web domain in order to operate for example: http://www.uml.edu
3. Images `![alt text here](path-to-image-here)` or `![alt text here](http://example.com/logo.png)` `![uml logo](https://www.uml.edu/Images/uml_vertical_logo_with_black_tagline_tcm18-288512.png)`
4. Also, commands can be found in the "Markdown Quick Reference" guide under the "Help" tab

Choose your outputs

replace the current output with either of the following:

1. output: html_document *html file (web page)*
2. output: pdf_document *pdf file(pdf)*
3. output: ioslides_presentation *ioslides slideshow (html)*
4. output: word_document *Microsoft Word (.docx)*
5. output: beamer_presentation *beamer slideshow (pdf)*

Rendering/Knitting

```
1. Option 1: ‘‘‘ rmarkdown::render("<insert your file path here>") ‘‘‘
2. Option 2: ‘‘‘ rmarkdown::render("<insert your file path here>","<insert output of your choice: pdf
3. Option 3: Knit Document button: File -> Knit Document‘‘‘
4. Option 4: To knit your document: for Mac Users ’Command + Shift + K’ and Windows/Linux Users ’Ctrl
```

Section 3: Shortcuts and Tips

Timestamp: starting in 22:30

Keyboard Shortcuts

1. We will create plenty of 'R Code Blocks' where R code will be evaluated and printed, a shortcut to insert a "R code Block" for Mac Users is 'Command + Option + I' and Windows/Linux Users is 'Ctrl + Alt + I'
2. To output your document: for Mac Users 'Command + Shift + K' and Windows/Linux Users 'Ctrl + Shift + K'
3. Run the current R code block: for Mac Users 'Command + shift + Enter' and Windows/Linux Users 'Ctrl + Shift + Enter'
4. Run all R code blocks in the .Rmd file: for Mac Users 'Command + Option + R' and and Windows/Linux Users 'Ctrl + Alt + R'

Altering Your R Code Blocks

1. `eval = FALSE    Show code, but do not evaluate it`
2. `fig.show = "hide" Hides plots`
3. `results = "hide" Hides printed output`
4. `include = FALSE Runs code, but suppresses all output. This is helpful for setup code.`
5. `message=TRUE Whether to display messages, FALSE does not display`

This is not the definitive list of tricks, only the tutorials demostration. For more information, please refer back to R Markdown tips, tricks, and shortcuts in the references section

Youtube Video

https://youtu.be/8ogEaPWMwCw

Section 4: Plotting/Modeling/Mapping with the "Iris" dataset

Timestamp: starting in 32:40

Selected Dataset

The **Iris dataset** which is one of many built-in datasets found in R. This dataset consists of 5 variables: sepal length, sepal width, petal length and petal width measured in centimeters of 50 samples each of 3 species of Iris: Iris setosa, versicolor, and virginica totaling 150 samples.

install packages for this dataset

```r
install.packages("RColorBrewer") # example of modfied R code block
install.packages("gplots") # show code only not evaluated, because these packages are already installed
```

Load libraries for graphs

```r
library(RColorBrewer) # for the colors of the graphs
library(gplots) # for heat map
```

Load Iris Dataset

```r
library(datasets)
data("iris") #"iris" is one of few built-in R datasets!
summary(iris) #output a summary data table of all 150 irises
```

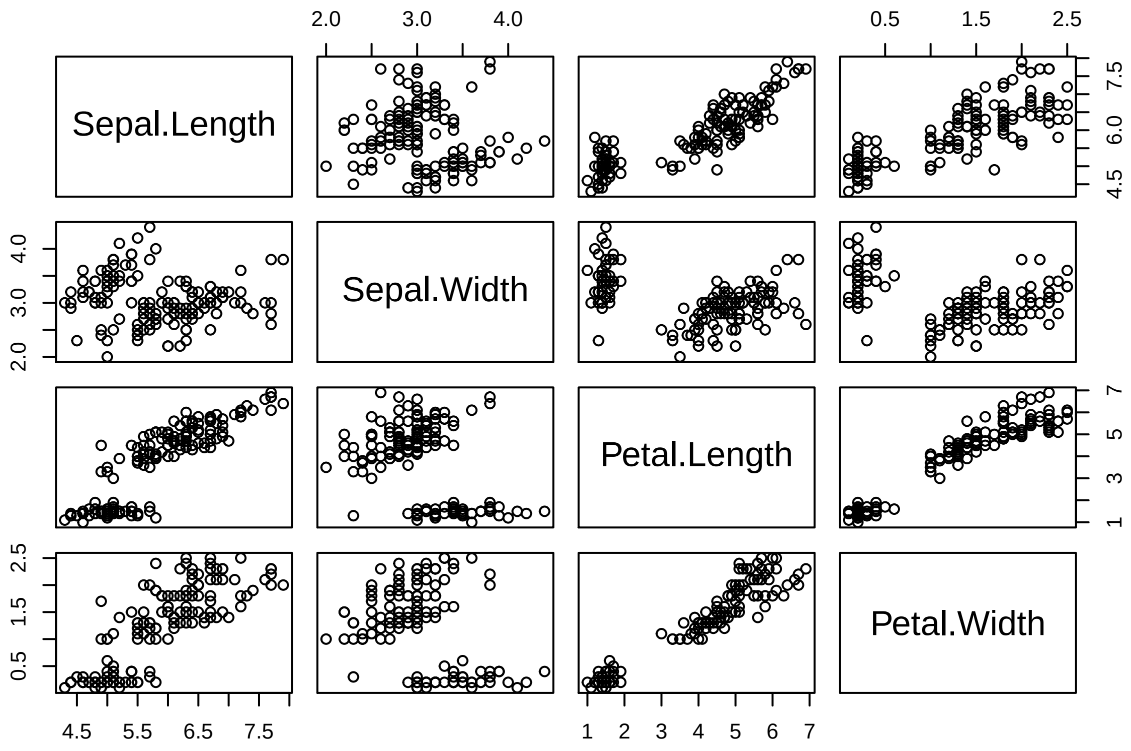
```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
## 
```

Plots for Iris Dataset

```r
names(iris) # list variables
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

```r
pairs(iris[1:4]) # illustrate how variables compare to eachother. Excluding "Species"
```



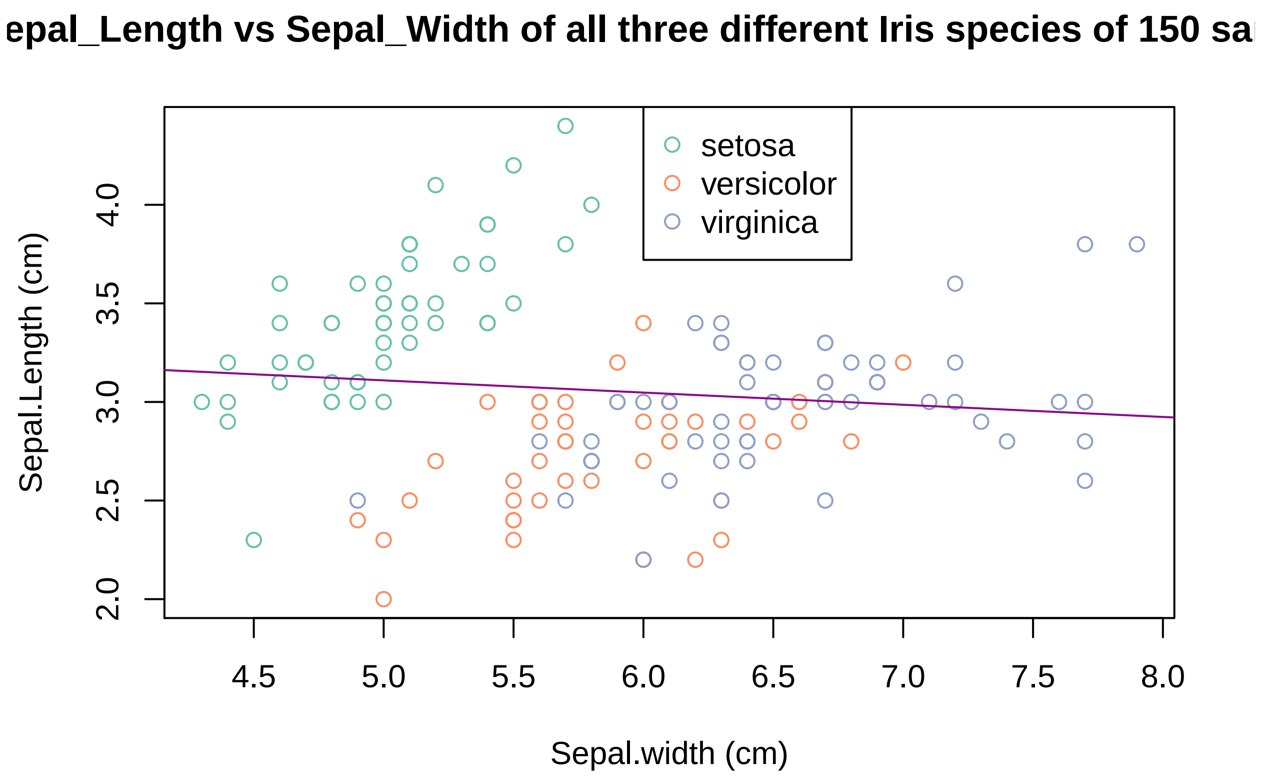
```
#this function provides a list of scatterplots of each variable being "mapped" to eachother as both ind
```

Exploring Rcolor Brewer

```
library(RColorBrewer)
display.brewer.all(n=3)
brewer.pal.info # provide information of all possible palette names
```

Regression model of Sepal_Length vs Sepal_Width of all three different Iris species of 150 samples

```
# Plotting the main and both axis titles
# RColorBrewer (n, name). n = number of different colors (3), name = selected from "brewer.pal.info". M

plot(Sepal.Width ~ Sepal.Length, data = iris, main = "Sepal_Length vs Sepal_Width of all three differen

# Add regression line
abline(lm(Sepal.Width ~ Sepal.Length, data = iris), col = 'darkmagenta')

# legend of of the three different species
legend(x=6.0, y=4.5, legend=levels(iris$Species), col=brewer.pal(3, "Set2"), pch=1)
```



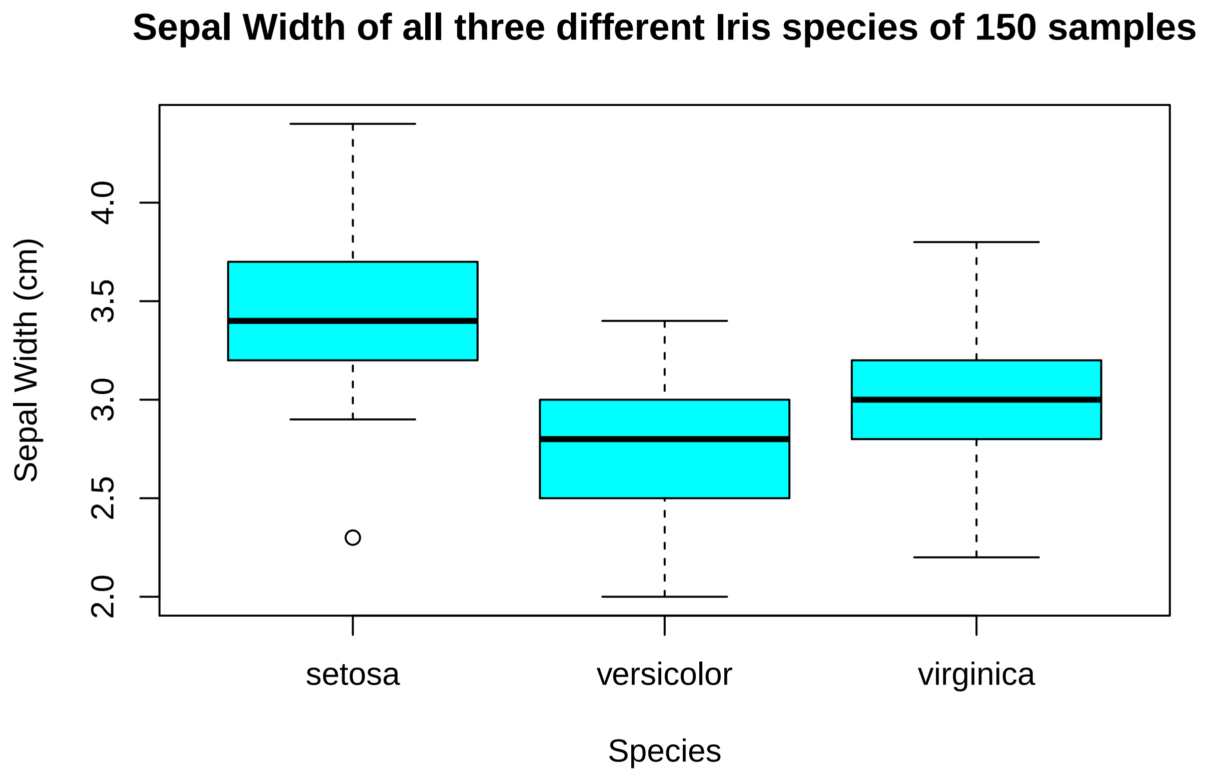
```r
#Summary table of regression data
summary(lm(Sepal.Width ~ Sepal.Length, data = iris))
```

```
##
## Call:
## lm(formula = Sepal.Width ~ Sepal.Length, data = iris)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1095 -0.2454 -0.0167  0.2763  1.3338
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.41895    0.25356   13.48   <2e-16 ***
## Sepal.Length -0.06188    0.04297   -1.44    0.152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4343 on 148 degrees of freedom
## Multiple R-squared:  0.01382,    Adjusted R-squared:  0.007159
## F-statistic: 2.074 on 1 and 148 DF,  p-value: 0.1519
```

```r
#Notice how Setosa have noticably short Sepal.width. However length its the longest compared to Versico
#The regression line tells that future outcome will range around the line's value
```

Boxplot of Sepal Width of all three different Iris species of 150 samples

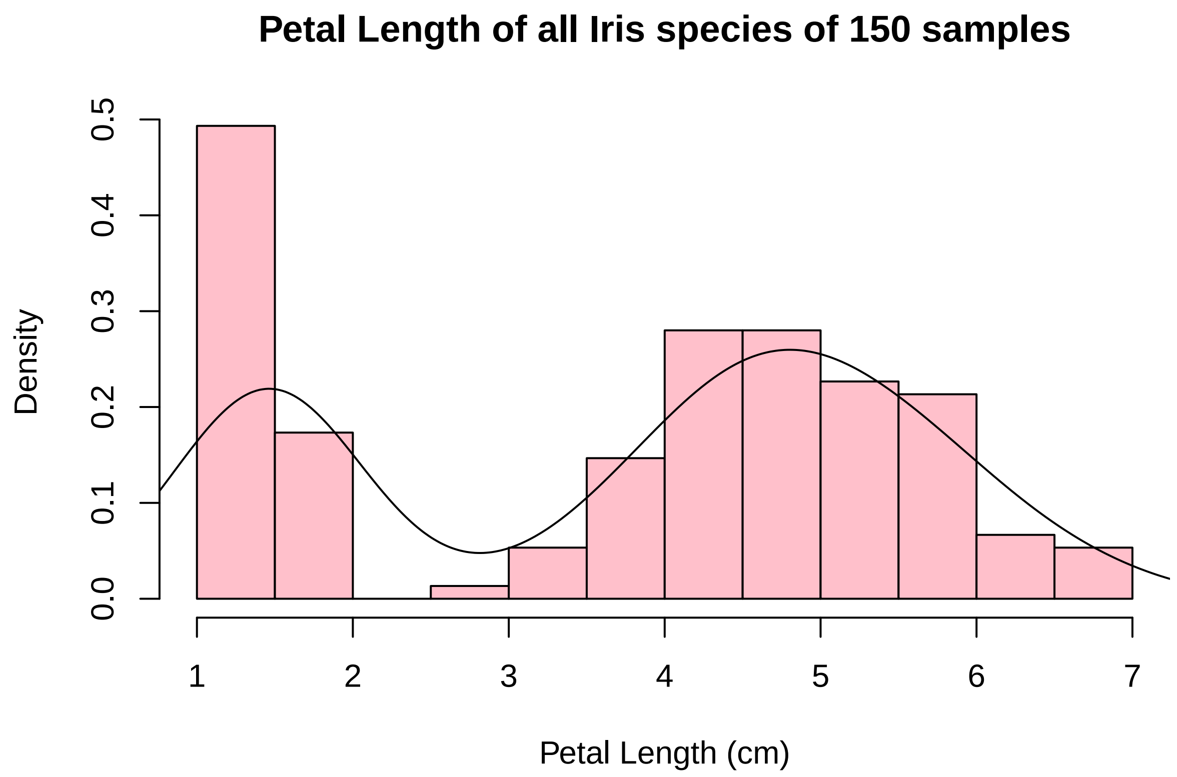
```
boxplot(Sepal.Width ~ Species, data = iris, main = "Sepal Width of all three different Iris species of
```



```
#Y-variable is mapped to X-variable
#Setosa have the miniumin sepal width as it’s outlier however,it also have the greatest sepal width and
```

Histogram of Petal Length of all three different Iris species of 150 samples

```
Petal_Length <- iris$Petal.Length #collect all petal length samples in all 150 Iris samples
hist(Petal_Length, main = "Petal Length of all Iris species of 150 samples", xlab = "Petal Length (cm)"
# freq = FALSE, check for population density instead of freq = TRUE which counts frequency

lines(density(iris$Petal.Length)) #create density plot line
```



```
summary(Petal_Length, data = iris)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.600   4.350   3.758   5.100   6.900
```

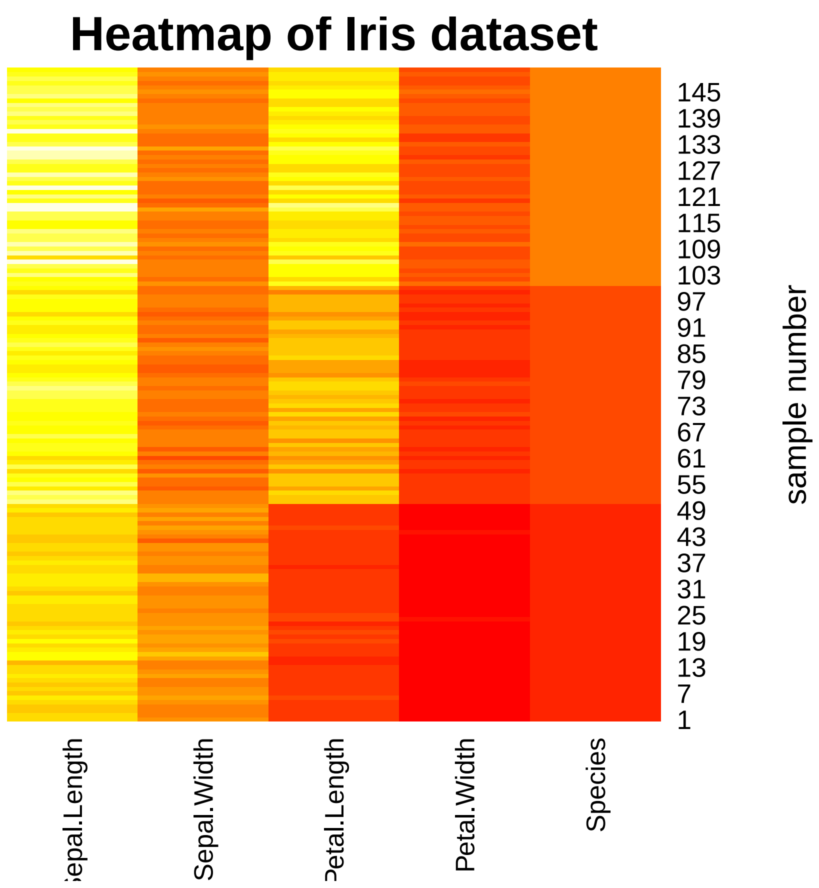
```
#this density plot indicates the likeliness based on frequency data where the Petal length is most like
```

Heat Maps of Iris Dataset

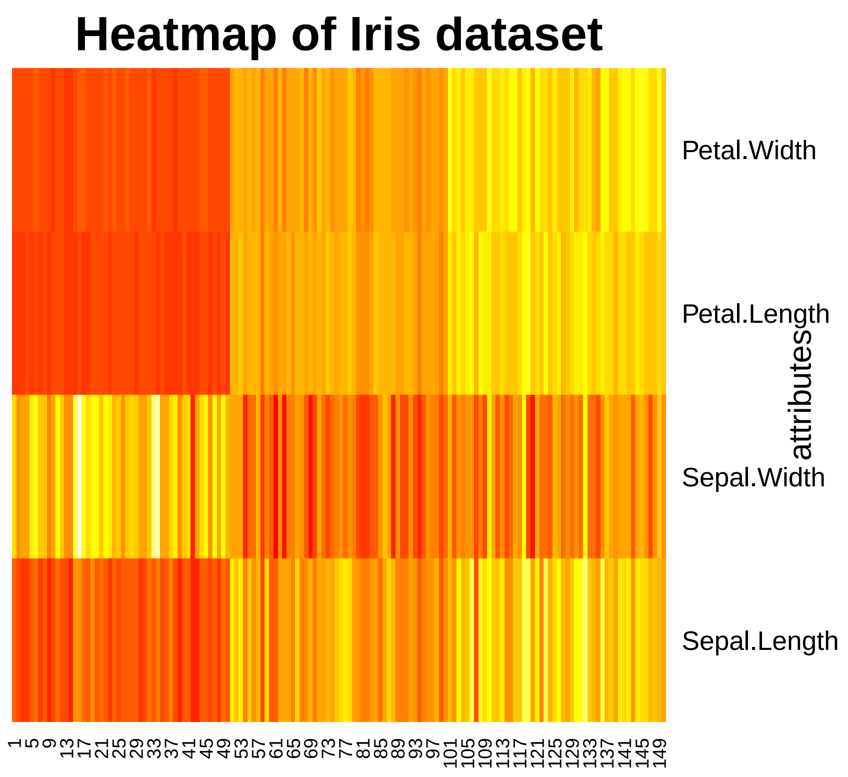
```
#heatmap() found in default R package
?heatmap # for quick reference guide

#set up class for iris dataset
class(iris)
```

```
## [1] "data.frame"
```

```
#convert iris_heat dataframe into a matrix
iris_heat <- data.matrix(iris)
heatmap(iris_heat,  scale="none", cexRow=1, cexCol = 1, col= heat.colors(n=20), Rowv = NA, Colv = NA, y
```



```
#Transpose dataset (flip x and y axis)
heatmap(t(iris[, 1:4]), scale="row",  cexRow=1, col = heat.colors(n=20), Rowv = NA, Colv = NA, ylab =
```



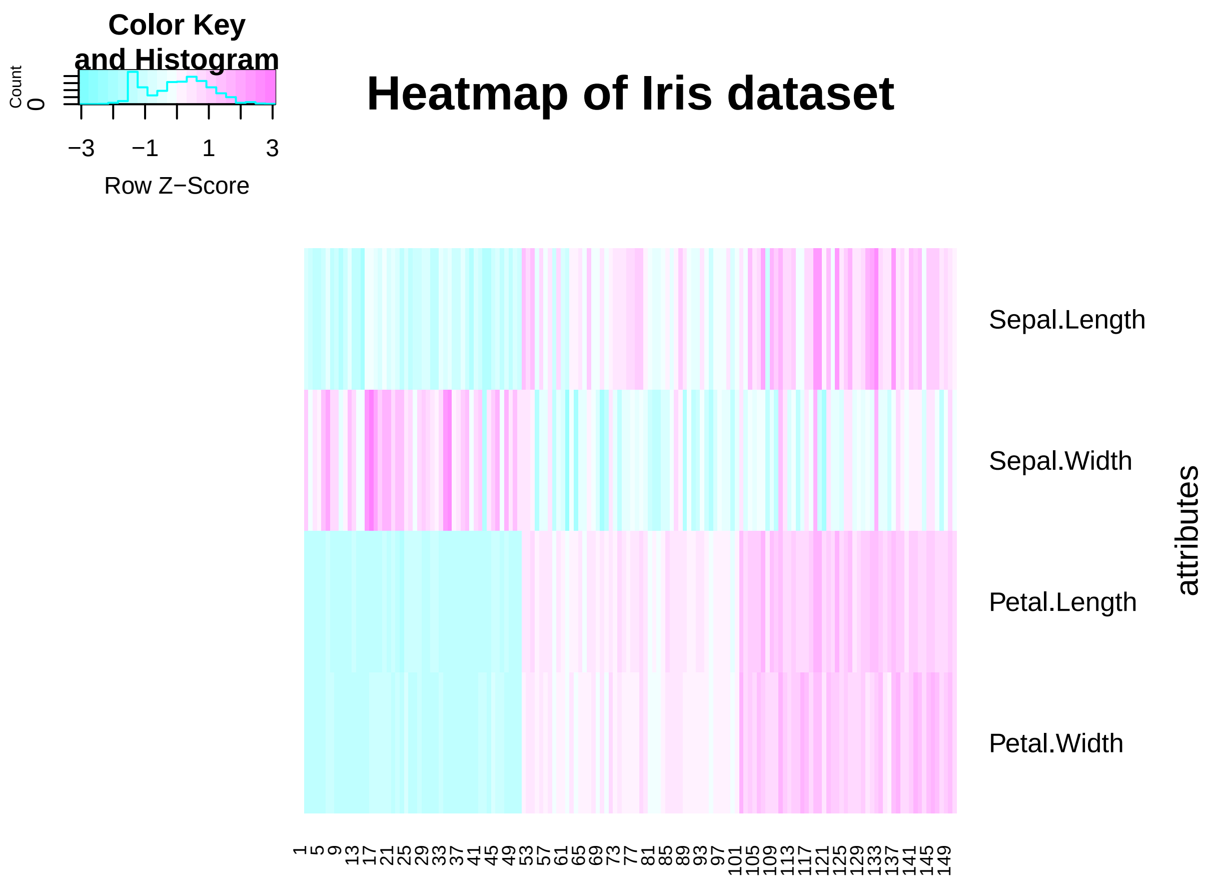
```
#"scale" indicates if the values in either the row direction or column direction
# heat.colors() heat map default colors, n = number of colors. hcl.pals() to list avaliable color palet
#"Colv" for the column dendrogram and "RowV" for Row dendrogram. 'NA' does not render dendrograms in pl
#"cexRow" and "cexCol" are for scaling of the x and y axis

#red equates high and yellow equates to low value. More matching, means more similiar and related the d
# Notice samples 1-49 have lower sepal width compare to other samples, but greater value in the other a
# Notice how consistent petal.width, petal.length. sepal.length is pattern wise based on the color howe
```

Iris Enhance Heat Map

```
?heatmap.2 #reference guide

library(gplots) #enchanced heatmap
heatmap.2(t(iris[, 1:4]), trace="none", scale="row", key=TRUE, margins=c(2, 8), cexRow=1, col = cm.colo
```

```
## Warning in heatmap.2(t(iris[, 1:4]), trace = "none", scale = "row", key =
## TRUE, : Discrepancy: Rowv is FALSE, while dendrogram is 'both'. Omitting row
## dendogram.
```

```
## Warning in heatmap.2(t(iris[, 1:4]), trace = "none", scale = "row", key =
## TRUE, : Discrepancy: Colv is FALSE, while dendrogram is 'column'. Omitting
## column dendogram.
```



```
# "t(iris[,1:4])" matrix transpose that reconstruct the order of attributes and sample number
# "trace" shows you where on the range the cells are positions. options are column","row","both","none"
# "key=TRUE"    logical indicating whether a color-key should be shown.
# "margins = c(2,8)" gives less space to x axis labels, but more space to y axis labels


#if error of graphic error occurs debug with dev.off()

#about the data, it's very similar inference compare the to heatmap above however the presentation is a
```

Section 5: Saving your Rmarkdown file and References

Timestamp: starting in 40:57

Assuming that you have Adobe Acrobat Reader installed in your computer 1. Go to the top right corner of your PDF file 2. Go to “Downloads” 3. It’ll open up in the Adobe Acrobat Reader Application/Client 4. Go to “File”, Save As, rename and there you have your final PDF file of your report

Congratulations! you have learned the overall basics of how to use RMarkdown and hopefully you’ll be able to incorporate these lessons of the tutorial into your assignments, projects, and overall productivity involving R and RStudio.

References

Bates, C. (2020, June 23). *R Markdown tips, tricks, and shortcuts* – Dataquest. Dataquest. Retrieved December 18, 2020, from https://www.dataquest.io/blog/r-markdown-tips-tricks-and-shortcuts/

Dataviscatalouge. (2018, January 16). A Guide to Heatmaps [Video]. YouTube. https://www.youtube.com/watch?v=lOowWU1t6q4

Moerane, S. M. (2016, August 29). *An introduction to R using Iris.* RPubs. https://rpubs.com/moeransm/intro-iris

Rdrr.io. (2014, December 7). *ColorBrewer: ColorBrewer palettes in RColorBrewer: ColorBrewer palettes.* R Package Documentation. https://rdrr.io/cran/RColorBrewer/man/ColorBrewer.html

RStudio. (2016). *Compiling reports from R scripts. R Markdown.* Retrieved December 17, 2020, from https://rmarkdown.rstudio.com/articles_report_from_r_script.html

UQ Library. (2019, June 13). R data visualisation with RStudio: heatmaps [Video]. YouTube. https://www.youtube.com/watch?v=V-IRkO4NIHU

Waldon, L. (n.d.). *Introduction to R graphics. RPubs.* https://rpubs.com/lwaldron/iris

Xie, Y., Allaire, J. J., & Grolemund, G. (2020, December 14). *Chapter 1 installation | R Markdown: The definitive guide.* Bookdown. Retrieved December 17, 2020, from https://bookdown.org/yihui/rmarkdown/installation.html