

Reproducible Research with knitr+R+L^AT_EX

Mohammad R. Arbabshirani

October 4, 2013

Contents

1	Introduction	1
2	Reproducible Research	1
3	Changing the default setting in ggplot	2
4	Final Comments	6

1 Introduction

There are many situations where a report contains text, code, code results and figures. Preparing each element of the report separately and combining them in a word processing software can be cumbersome. Moreover, if for any reason, one need the change the data or code, most steps should be repeated. The purpose of this document is 2-fold. First I want to describe a unified framework with R, knitr and L^AT_EX that makes reproducible research much easier by enabling you to write text and code in a single documents. Second, I want to discuss figure setting manipulation in the ggplot package. Often, one is not satisfied with the default figure settings such as font and label size. It is necessary to adjust those setting to better communicate with the reader and/or meet the poster/paper requirements.

2 Reproducible Research

Hopefully all of you have R studio installed on your computers. The next steps are as follows:

- Download and install L^AT_EX. You can download it from: <http://miktex.org/download>
- Install R package: "knitr". You can install it manually by clicking on Packages → Packages or by running `install.packages("knitr")` in the console.
- In R studio, go to Tools → Options → Sweave and make sure that in "Weave Rnw files using", *knitr* is selected. You can use any pdfviewer but some systems have problems with some pdfviewers so you can change it if you face a problem later.

You are all set! You can now download this document in .rnw format. Paste it or open it in a new file in your R Studio and save it as ".rnw" not ".r". For example you can save it as "rep_research.rnw". Now click on the Compile PDF on top of the editor. Hopefully everything works just fine and a pdf file is created.

3 Changing the default setting in ggplot

Most of the time, the default setting for figures are not pleasant. This happens in almost every scientific computing software I have seen so far. A good software should give you control over different elements of the figure. These options are available in ggplot. Let's start by an example. since I want to write code now, I need to create a new *chunk*. Every time you want to write code, you need a *chunk* which starts with "<<" and ends with "@". This way, R+knitr+L^AT_EX understand that this part is code not text. After "||" you need a *unique name*. There are many options for chunk which can be found here: <http://yihui.name/knitr/options> If you want you code become visible in the final document (for example in your homeworks), make sure that "echo=TRUE" is in the chunk header line. Ok, lets take a look at the data in problem 1 of homework 2:

```

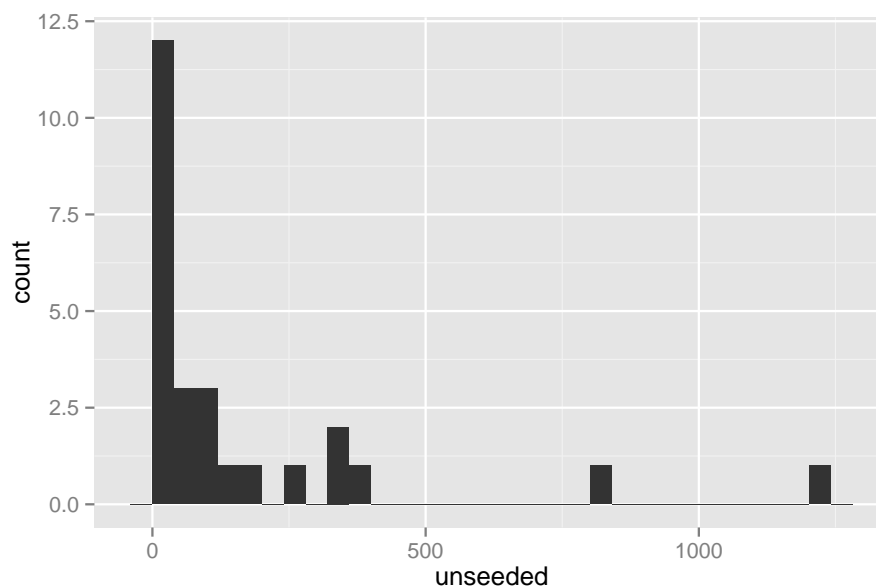
d1 <- read.csv("http://statacumen.com/teach/ADA1/ADA1_HW_02_F13-1.csv")
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 2.15.3

p <- ggplot(d1, aes(x = unseeded))
p <- p + geom_histogram()
print(p)

## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust
this.

```



As you see, the font are small. Lets make some changes to the defaults. Remember that most setting are accessible via *theme* command:

```

d1 <- read.csv("http://statacumen.com/teach/ADA1/ADA1_HW_02_F13-1.csv")
library(ggplot2)

# Plotting Parameters
# -----
plot.size = 5 #(height,width)
main.font.size = 18
legend.font.size = 16

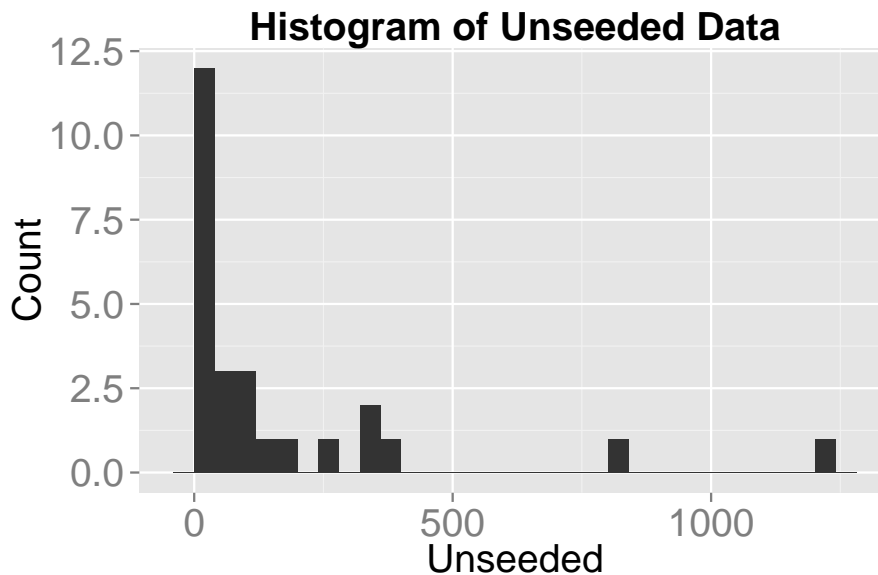
```

```

p <- ggplot(d1, aes(x = unseeded))
p <- p + geom_histogram() + xlab("Unseeded") + ylab("Count")
p <- p + ggtitle("Histogram of Unseeded Data")
p <- p + theme(axis.text.y = element_text(size = main.font.size))
p <- p + theme(axis.text.x = element_text(size = main.font.size))
p <- p + theme(axis.title.y = element_text(size = main.font.size))
p <- p + theme(axis.title.x = element_text(size = main.font.size))
p <- p + theme(plot.title = element_text(size = main.font.size, face = "bold"))
p <- p + theme(legend.text = element_text(size = legend.font.size, face = "bold"))
print(p)

## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust
this.

```



The axis and tick labels are readable now. We can also change the color and background a little bit:

```

d1 <- read.csv("http://statacumen.com/teach/ADA1/ADA1_HW_02_F13-1.csv")
library(ggplot2)

# Plotting Parameters
# -----
plot.size = 5 #(height,width)
main.font.size = 18

```

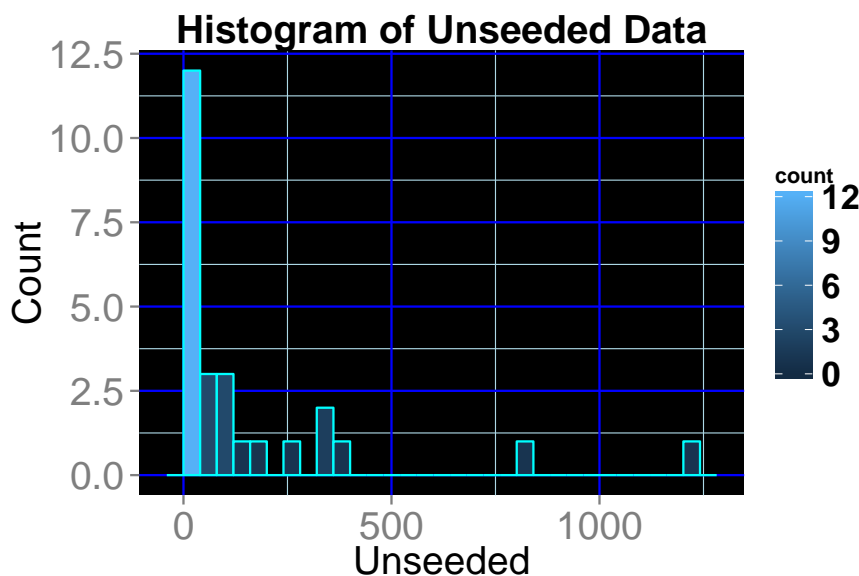
```

legend.font.size = 16

p <- ggplot(d1, aes(x = unseeded))
p <- p + geom_histogram(aes(y = ..count.., fill = ..count..), colour = "cyan")
p <- p + ggtitle("Histogram of Unseeded Data")
p <- p + xlab("Unseeded") + ylab("Count")
p <- p + theme(axis.text.y = element_text(size = main.font.size))
p <- p + theme(axis.text.x = element_text(size = main.font.size))
p <- p + theme(axis.title.y = element_text(size = main.font.size))
p <- p + theme(axis.title.x = element_text(size = main.font.size))
p <- p + theme(plot.title = element_text(size = main.font.size, face = "bold"))
p <- p + theme(legend.text = element_text(size = legend.font.size, face = "bold"))
p <- p + theme(panel.background = element_rect(fill = "black"))
p <- p + theme(panel.grid.major = element_line(colour = "blue"))
p <- p + theme(panel.grid.minor = element_line(colour = "light blue"))
print(p)

## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust
this.

```



You can also change the legend and main background settings:

```

d1 <- read.csv("http://statacumen.com/teach/ADA1/ADA1_HW_02_F13-1.csv")
library(ggplot2)

```

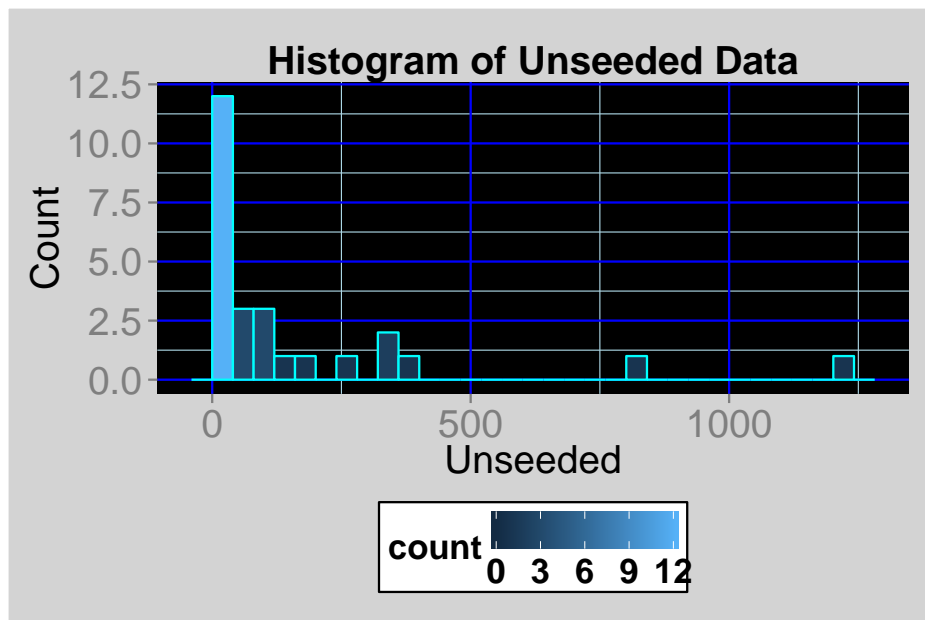
```

# Plotting Parameters
# -----
plot.size = 5 #(height,width)
main.font.size = 18
legend.font.size = 16

p <- ggplot(d1, aes(x = unseeded))
p <- p + geom_histogram(aes(y = ..count.., fill = ..count..), colour = "cyan")
p <- p + ggtitle("Histogram of Unseeded Data")
p <- p + xlab("Unseeded") + ylab("Count")
p <- p + theme(axis.text.y = element_text(size = main.font.size))
p <- p + theme(axis.text.x = element_text(size = main.font.size))
p <- p + theme(axis.title.y = element_text(size = main.font.size))
p <- p + theme(axis.title.x = element_text(size = main.font.size))
p <- p + theme(plot.title = element_text(size = main.font.size, face = "bold"))
p <- p + theme(legend.text = element_text(size = legend.font.size, face = "bold"))
p <- p + theme(panel.background = element_rect(fill = "black"))
p <- p + theme(panel.grid.major = element_line(colour = "blue"))
p <- p + theme(panel.grid.minor = element_line(colour = "light blue"))
p <- p + theme(legend.position = "bottom")
p <- p + theme(legend.background = element_rect(colour = "black"))
p <- p + theme(legend.title = element_text(size = legend.font.size))
p <- p + theme(plot.background = element_rect(fill = "light gray"))
print(p)

## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust
this.

```



For more setting please visit: <http://docs.ggplot2.org/0.9.2.1/theme.html>. Please note that **many of the demonstrated changes are not necessary nor interesting**. You should always try to increase information to ink ratio to the level that is suitable for the reader. Figures should be in black and white unless color is necessary.

4 Final Comments

- Using knitr+R+ \LaTeX requires writing in \LaTeX . This might take time for newbies to \LaTeX . Moreover, debugging \LaTeX is sometime confusing.
- Please note that when you compile the .rnw file, first it is converted to .tex and then from .tex to the .pdf. You can open the .tex file in the tex editor and run in there if you want.
- There are 2 types of errors that you may get when compiling the .rnw file. First is the error in your R code. You will see the line number corresponding to your R code that caused the problem. The second type is \LaTeX error. The error gives you a line number which is **NOT** the line number in your .rnw file. This is the line in the .tex file which

cased the problem. Sometimes it is better to open the .tex file and figure out the problem there.

- Always choose unique chunk names (without space)
- The header of the chunk starts with "`<<`" and ends with "`>>=`".
- The chunk ends with "`@`".
- Sometimes you get strange errors but the pdf file is built! One of the common errors of this type is: "Issues: 1 badbox".
- For more information on knitr+R+L^AT_EX you can refer to: http://statacumen.com/teach/SC1/SC1_01_LaTeX_R.pdf