



STOR 320 Workflow in RMarkdown

Lecture 3

Yao Li

Department of Statistics and Operations Research

UNC Chapel Hill



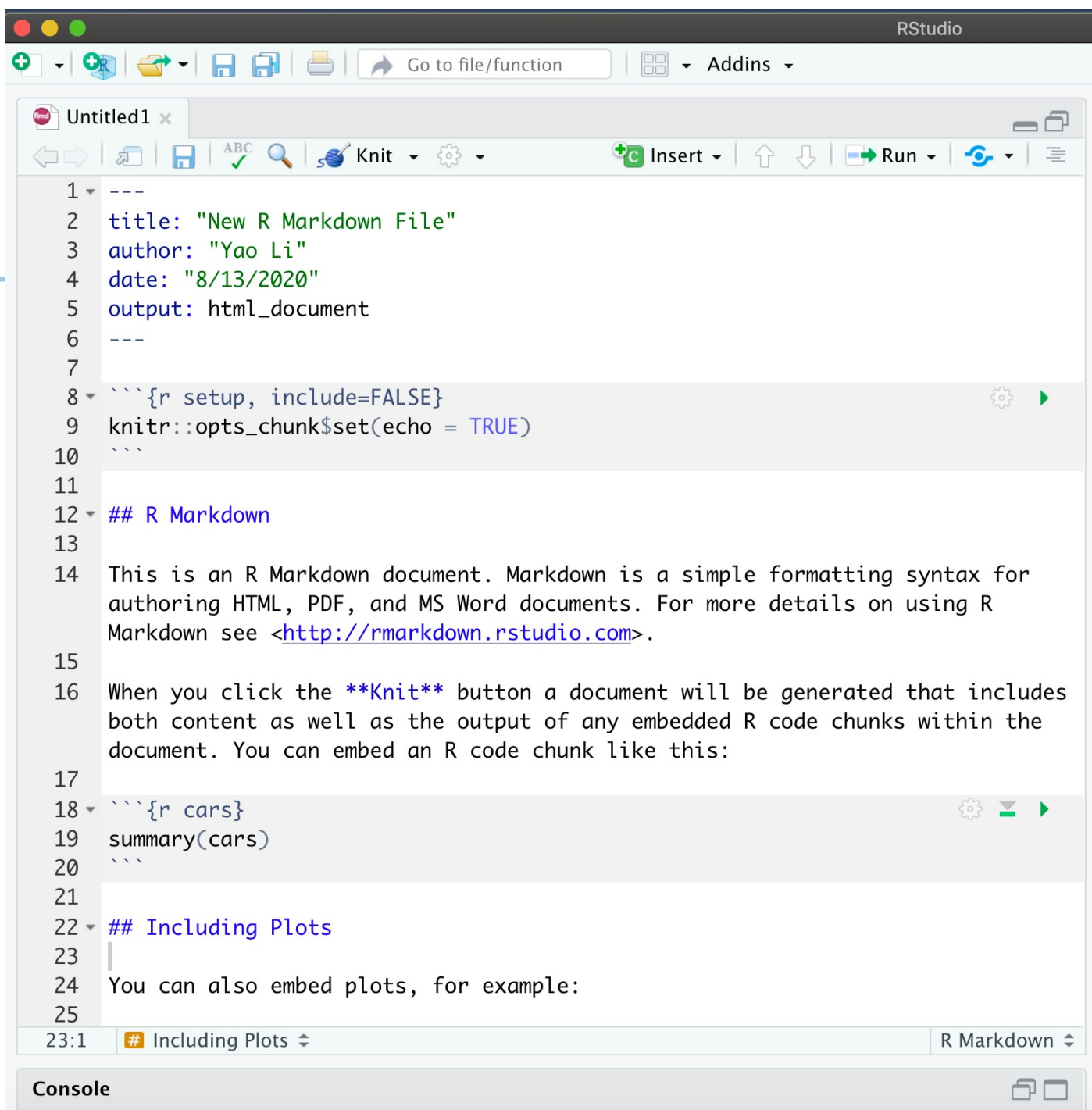
Workflow Information

- Chapters Discussing Workflow
 - Chapter 4: Basics (calculation, object, function, etc)
 - Chapter 6: Rscripts (R script, diagnostics)
 - Chapter 8: Projects
- Our Focus is on Workflow Within RMarkdown
- Today's Lecture on RMarkdown
 - Running R Code
 - Objects
 - Functions



Essential Reads

- Highly Advised Reading
 - Chapter 27: RMarkdown
 - Basics
 - Text Formatting
 - Code Chunks
 - Chapter 28: More ggplot Info
 - Labeling
 - Annotating
 - Scaling
 - Zooming
 - Themes
 - Saving Graphics



The screenshot shows the RStudio interface with a file named 'Untitled1'. The editor contains the following R Markdown code:


```
1 ---
2 title: "New R Markdown File"
3 author: "Yao Li"
4 date: "8/13/2020"
5 output: html_document
6 ---
7
8 ```{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ```
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting syntax for
15 authoring HTML, PDF, and MS Word documents. For more details on using R
16 Markdown see <http://rmarkdown.rstudio.com>.
17
18 When you click the Knit button a document will be generated that includes
19 both content as well as the output of any embedded R code chunks within the
20 document. You can embed an R code chunk like this:
21
22 ```{r cars}
23 summary(cars)
24 ```
25
26 ## Including Plots
27
28 You can also embed plots, for example:
```

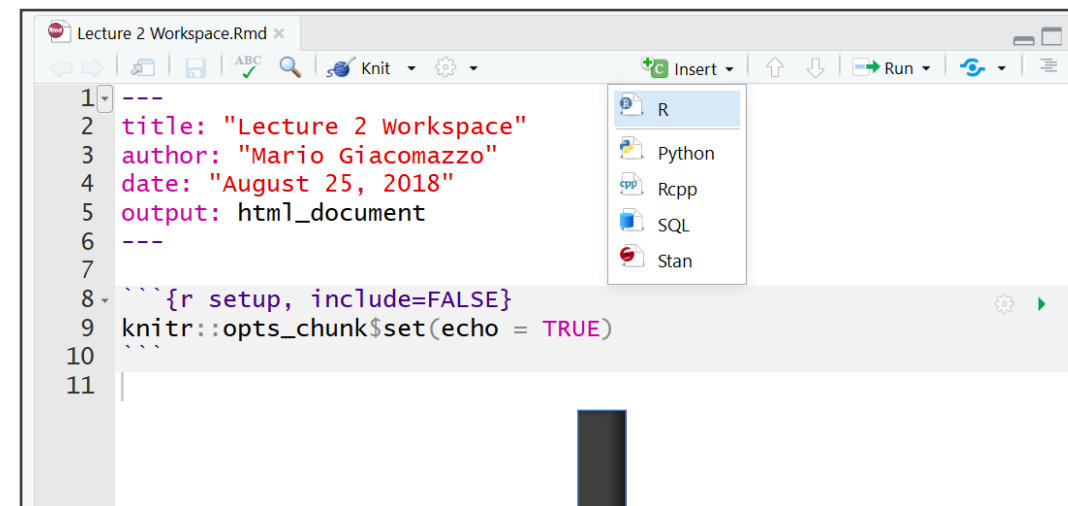


Rmarkdown File

[Cheat Sheet](#)


Placing Code in RMarkdown

- Code Chunks (Mini Rscripts)
 - R, Python, SQL, Rcpp (C++)
 - Inserting R Chunks
 - Method 1: 

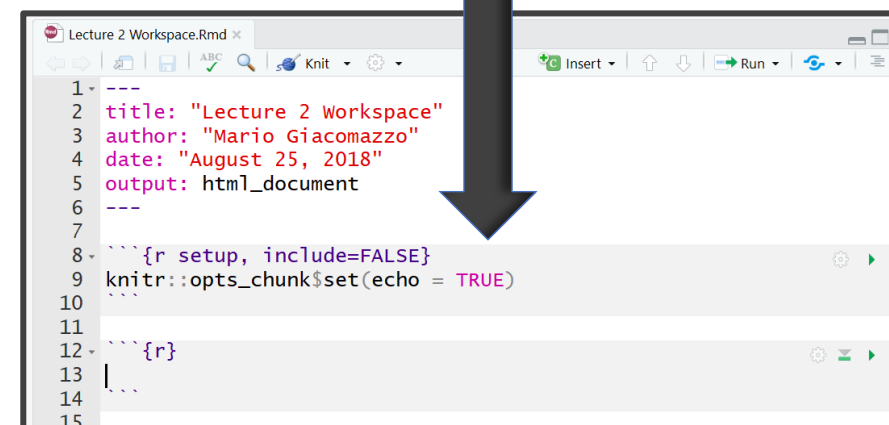


A screenshot of the RStudio interface. The 'Insert' menu is open, showing options for R, Python, Rcpp, SQL, and Stan. The R option is highlighted. The background shows a code chunk with the following content:

```
1 ---  
2 title: "Lecture 2 Workspace"  
3 author: "Mario Giacomazzo"  
4 date: "August 25, 2018"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ```  
11
```

- Method 2: Ctrl+Alt+I
- Method 3: Type ````${r}```` 

Put R code here



A screenshot of the RStudio interface showing the same code chunk as above, but with a new R code chunk inserted at the bottom. A large black arrow points from the 'R' option in the menu of the top screenshot to this new chunk. The new chunk is:

```
12 ```{r}  
13 |  
14 |  
15 |
```



Inline Code in RMarkdown

```
```\r\na <- c(1,2,3)\r\n```\n\nThe sum of vector $a$ is `r sum(a)`.
```

Knit to HTML

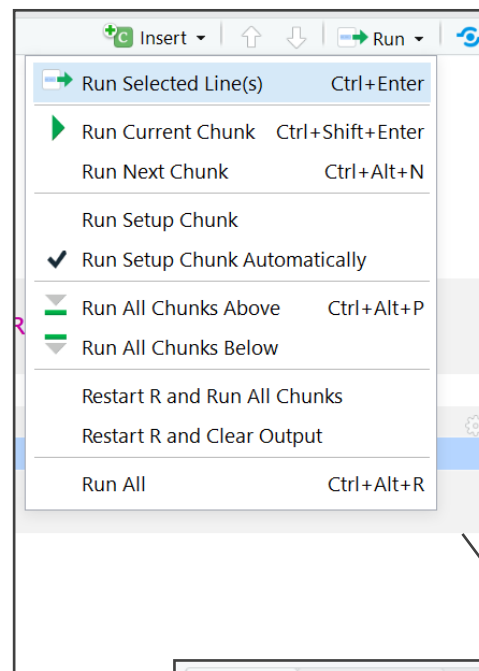
```
a <- c(1,2,3)\n\nThe sum of vector a is 6.
```

Evaluate R expression in  
the Markdown part

# Running Code in RMarkdown

- Various Ways
  - Highlighted Code

```
{r}
x=3
x
}
```

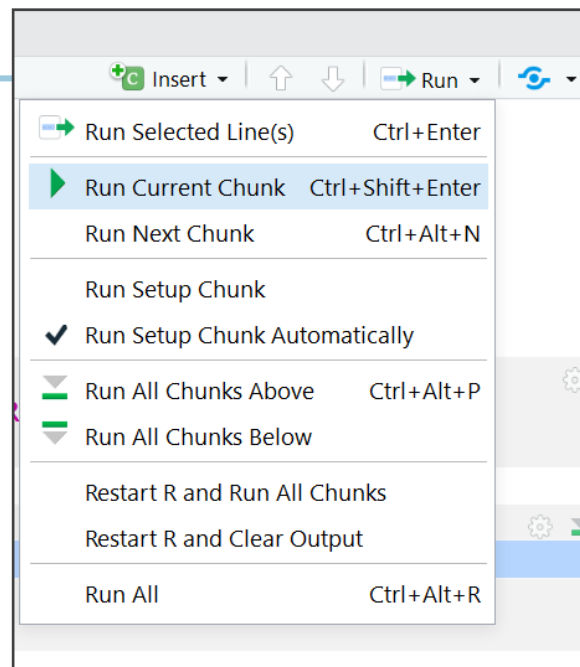


Ctrl+Enter

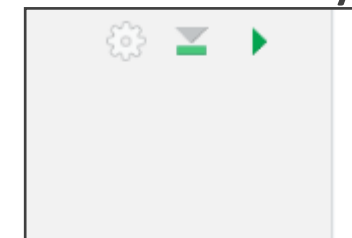
```
Console Terminal x
~/
> x=3
> x
[1] 3
> |
```

# Running Code in RMarkdown

- Various Ways (Cont.)
  - Chunking It (Recommended)



Press Play



Ctrl+Shift+Enter



A screenshot of the RStudio code editor. The code is enclosed in a code chunk with curly braces. The code inside the chunk is: `x=3` followed by a blank line and `x`. Below the code, the output is displayed as `[1] 3`. The RStudio interface elements like the gear icon and the Run button are visible in the top right corner of the editor window.





# Order

- Order Matters

```
```\r}\n#Created Variables x and y assigned to 3 and 4 respectively\nx=3\ny=4\nprint(c(x,y))\n```\n\n```\r}\nx+y #Addition\nx-y #Subtraction\nx*y #Multiplication\nx/y #Division\nx^y #Powers\nx%%y #Modulus (x mod y)\n```\n\nError: object 'x' not found
```

Why?
Environment is empty



Order

- Order Matters (Cont.)
 - Run First Chunk
 - Then, Run Second Chunk

```
{r}
#Created Variables x and y assigned to 3 and 4
respectively
x=3
y=4
print(c(x,y))

[1] 3 4

{xr}
x+y #Addition
x-y #Subtraction
x*y #Multiplication
x/y #Division
x^y #Powers
x%%y #Modulus (x mod y)

[1] 7
[1] -1
[1] 12
[1] 0.75
[1] 81
[1] 3
```

```
{r}
#Created Variables x and y assigned to 3 and 4 respectively
x=3
y=4
print(c(x,y))

[1] 3 4
```

Environment History Connections

Import Dataset

Global Environment

Values

x	3
y	4



Run All Previous Chunks

```
{r}
#Created Variables x and y assigned to 3 and 4 respectively
x=3
y=4
print(c(x,y))

[1] 3 4

{r}
x+y #Addition
x-y #Subtraction
x*y #Multiplication
x/y #Division
x^y #Powers
x%%y #Modulus (x mod y)

[1] 7
[1] -1
[1] 12
[1] 0.75
[1] 81
[1] 3

{r}
log(x) #Logarithm of x
abs(x-y) #Absolute value of x-y
exp(x) #e^x|
```

Runs All Previous Chunks

- Order Matters (Cont.)



Run All Previous Chunks

```
{r}
#Created Variables x and y assigned to 3 and 4 respectively
x=3
y=4
print(c(x,y))

[1] 3 4

{xr}
x+y #Addition
x-y #Subtraction
x*y #Multiplication
x/y #Division
x^y #Powers
x%%y #Modulus (x mod y)

[1] 7
[1] -1
[1] 12
[1] 0.75
[1] 81
[1] 3

{xr}
log(x) #Logarithm of x
abs(x-y) #Absolute value of x-y
exp(x) #e^x

[1] 1.098612
[1] 1
[1] 20.08554
```

Then, Run Current Chunk

- Order Matters (Cont.)



Chunk Options

```
`` `{r,eval=F}  
p3<-p2+geom_smooth(COMPLETE_INSIDE)  
p3  
``
```



Option	Run code	Show code	Output	Plots	Messages	Warnings
<code>eval = FALSE</code>	-		-	-	-	-
<code>include = FALSE</code>		-	-	-	-	-
<code>echo = FALSE</code>		-				
<code>results = "hide"</code>			-			
<code>fig.show = "hide"</code>				-		
<code>message = FALSE</code>					-	
<code>warning = FALSE</code>						-

[Chunk Options](#)



Objects in R: Vector and Matrix

```
````{r}
#Numeric Vector Named x
x=c(3,2,1,5,7,8)
#Prints x
x
#Third Element of x
x[3]
#Character Vector Named y
y=c("H","T","H","T","H","T")
#Fifth Element of y
y[5]
#3x2 Matrix Named z
z=matrix(c(3,2,1,5,7,8),
 nrow=2,ncol=3,byrow=T)
#Prints z
z
#First Row of z
z[1,]
#1st and 3rd Column of z
z[,c(1,3)]
```

- Many Types of Objects
  - Vector and Matrix

```
[1] 3 2 1 5 7 8
[1] 1
[1] "H"
 [,1] [,2] [,3]
[1,] 3 2 1
[2,] 5 7 8
[1] 3 2 1
 [,1] [,2]
[1,] 3 1
[2,] 5 8
```



# Objects in R: Dataframe

- Many Types of Objects (Cont.)
  - Tibble/Dataframe

```
```{r}
#Create Tibble named tbl
tbl<-tibble(x=x,y=y)
#Print tbl
tbl
```
```

| <b>x</b> | <b>y</b> |
|----------|----------|
| <dbl>    | <chr>    |
| 3        | H        |
| 2        | T        |
| 1        | H        |
| 5        | T        |
| 7        | H        |
| 8        | T        |

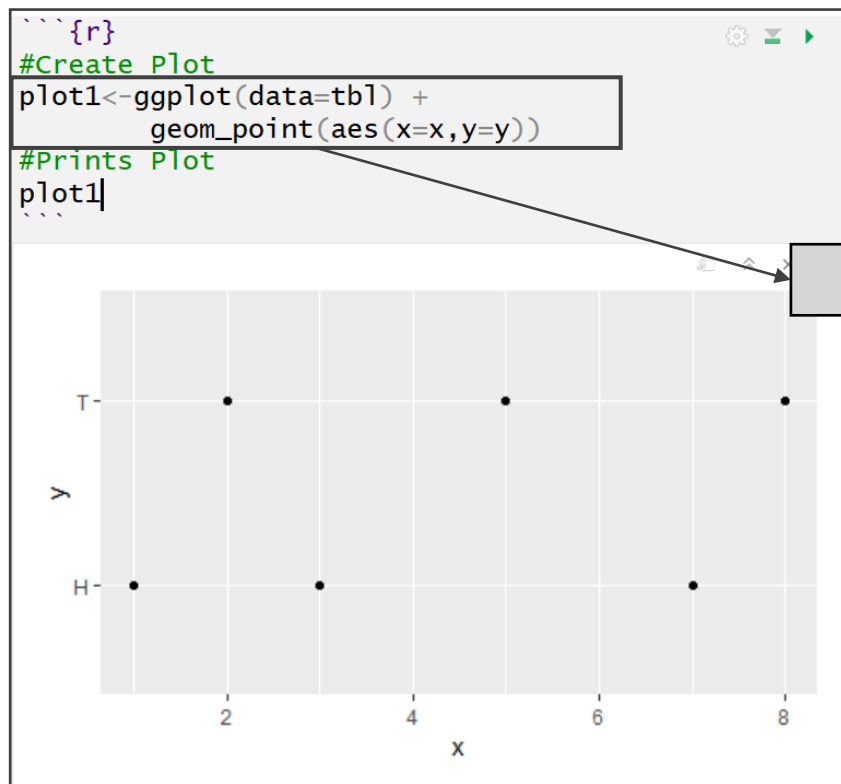
6 rows

```
```{r}
#Create Dataframe named df
df<-data.frame(x=x,y=y)
#Print df
df
```
```

| <b>x</b> | <b>y</b> |
|----------|----------|
| <dbl>    | <fctr>   |
| 3        | H        |
| 2        | T        |
| 1        | H        |
| 5        | T        |
| 7        | H        |
| 8        | T        |

6 rows

# Objects in R: Lists



- Many Types of Objects (Cont.)
  - Lists (Combines Different Objects)

```
Global Environment
Data
df 6 obs. of 2 variables
plot1 List of 9
 data :Classes 'tbl_df', 'tbl' and 'data.frame': 6 obs. of 2 v..
 ..$ x: num [1:6] 3 2 1 5 7 8
 ..$ y: chr [1:6] "H" "T" "H" "T" ...
 layers :List of 1
 ..$:Classes 'LayerInstance', 'Layer', 'ggproto', 'gg' <ggpro..
 aes_params: list
 compute_aesthetics: function
 compute_geom_1: function
 compute_geom_2: function
 compute_position: function
 compute_statistic: function
 data: waiver
 draw_geom: function
 finish_statistics: function
 geom: <ggproto object: Class GeomPoint, Geom, gg>
 aesthetics: function
 default_aes: uneval
 draw_group: function
 draw_key: function
 draw_layer: function
 draw_panel: function
 extra_params: na.rm
 handle_na: function
```





# Functions in R

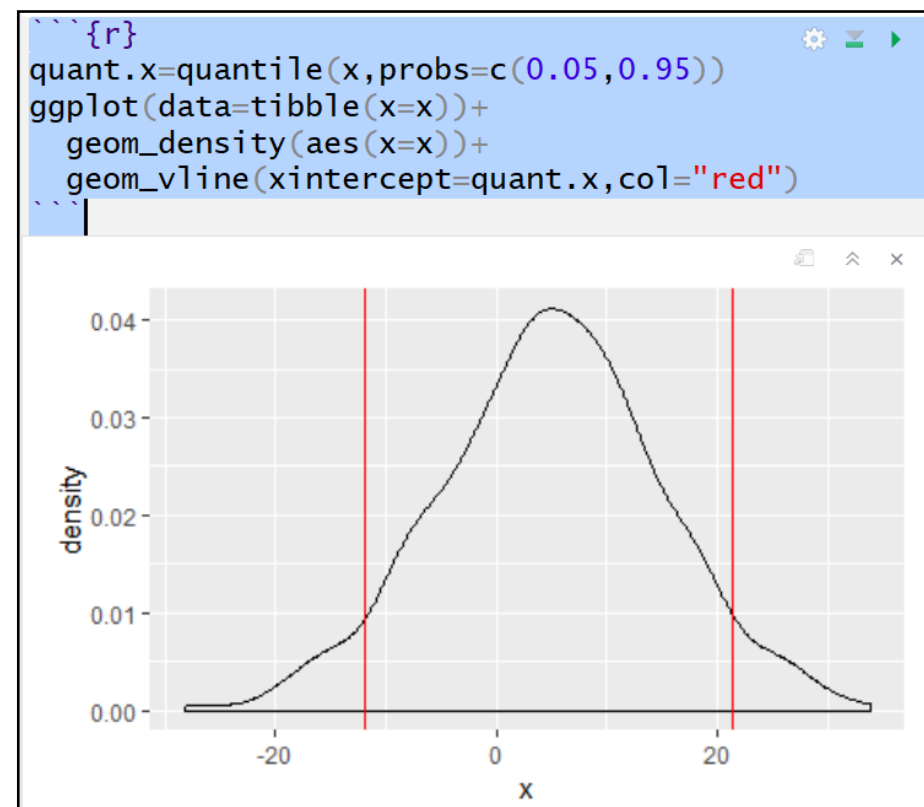
- Many Types of Functions
  - You: Input Objects and Specify Arguments (Defaults Exist)
  - Function: Outputs Objects
  - Example `> quantile()`
    - Input: Vector and Specified Probabilities
    - Output: Desired Percentiles
    - For online help, `> ?quantile`



# Functions in R

- Many Types of Functions (Cont.)
  - Example (Cont.)

```
Console Terminal x
~/
> #Randomly Draw 1000 Samples from
> #Normal Distribution with Mean=5 and SD=10
> x=rnorm(1000,mean=5,sd=10)
> mean(x) #Prints Sample Mean
[1] 4.905269
> sd(x) #Prints Sample SD
[1] 10.01766
> quantile(x) #Default Quantiles (Min,Quartiles,Max)
 0% 25% 50% 75% 100%
-28.232597 -1.480456 5.022031 11.433746 33.929228
> quantile(x,probs=c(0.05,0.95)) #Middle 90%
 5% 95%
-11.98847 21.30757
```





# Rmarkdown Training

---

**Now, let us**

**PRACTICE**

**Download the Rmd for Tutorial 2 to Your Computer from the Course Website and open the file in RStudio**