



STOR 320 Workflow in RMarkdown

Lecture 2

Yao Li

Department of Statistics and Operations Research

UNC Chapel Hill

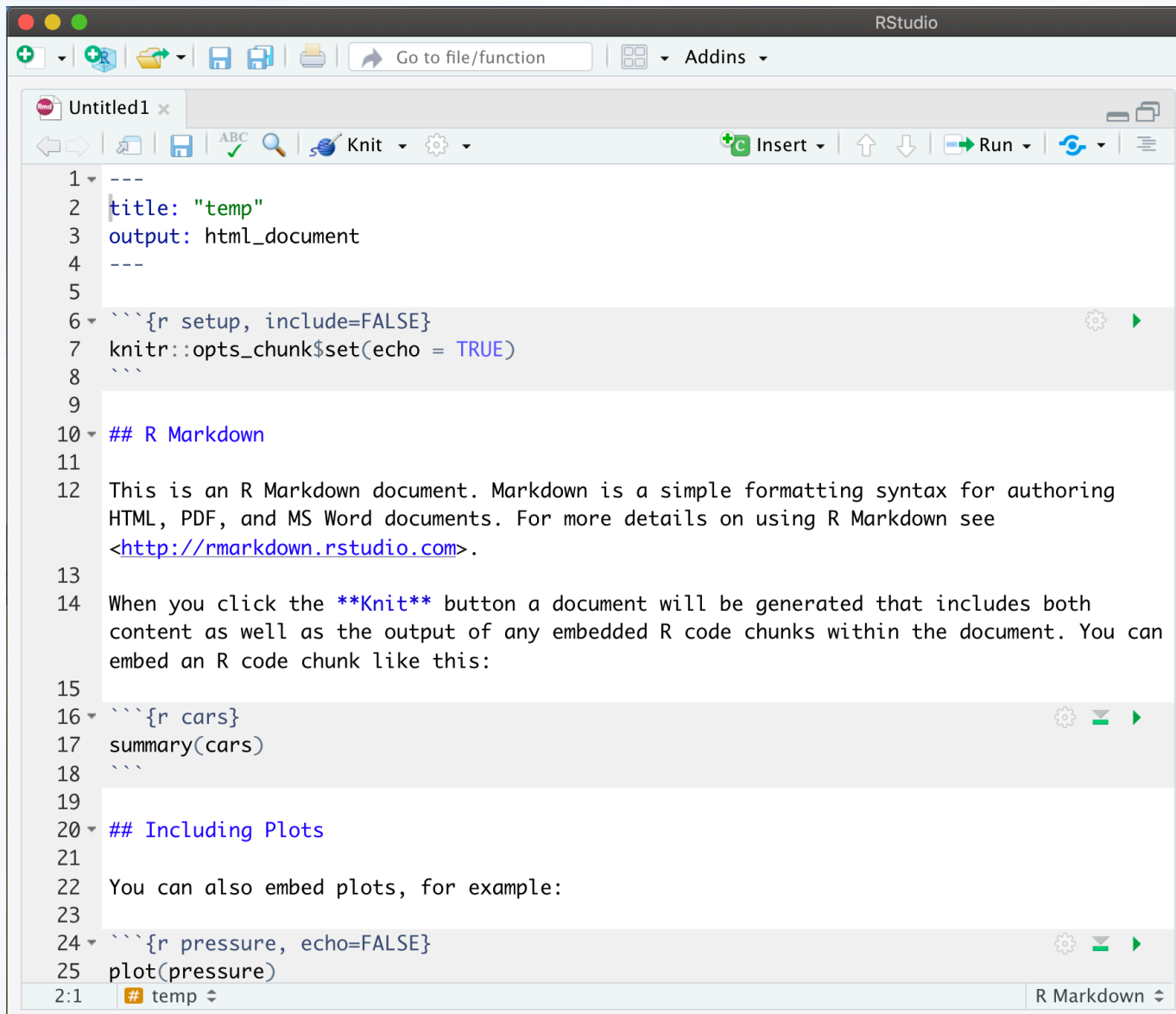
Workflow Information

- Chapters Discussing Workflow
 - Chapter 2: Basics
 - Chapter 4: Rscripts
 - Chapter 6: Projects
- Our Focus is on Workflow Within RMarkdown
- Today's Lecture on RMarkdown
 - Running R Code
 - Objects
 - Functions

Essential Reads

- Highly Advised Reading
 - Chapter 21: RMarkdown
 - Basics
 - Text Formatting
 - Code Chunks
 - Chapter 22: More ggplot Info
 - Labeling
 - Annotating
 - Scaling
 - Zooming
 - Themes
 - Saving Graphics

Rmarkdown File




```
1 ---
2 title: "temp"
3 output: html_document
4 ---
5
6 ```{r setup, include=FALSE}
7 knitr::opts_chunk$set(echo = TRUE)
8 ```
9
10 ## R Markdown
11
12 This is an R Markdown document. Markdown is a simple formatting syntax for authoring
13 HTML, PDF, and MS Word documents. For more details on using R Markdown see
14 <http://rmarkdown.rstudio.com>.
15
16 When you click the Knit button a document will be generated that includes both
17 content as well as the output of any embedded R code chunks within the document. You can
18 embed an R code chunk like this:
19
20 ```{r cars}
21 summary(cars)
22 ```
23
24 ## Including Plots
25
26 You can also embed plots, for example:
27
28 ```{r pressure, echo=FALSE}
29 plot(pressure)
30 ```
```

2:1 # temp R Markdown

[Cheat Sheet](#)

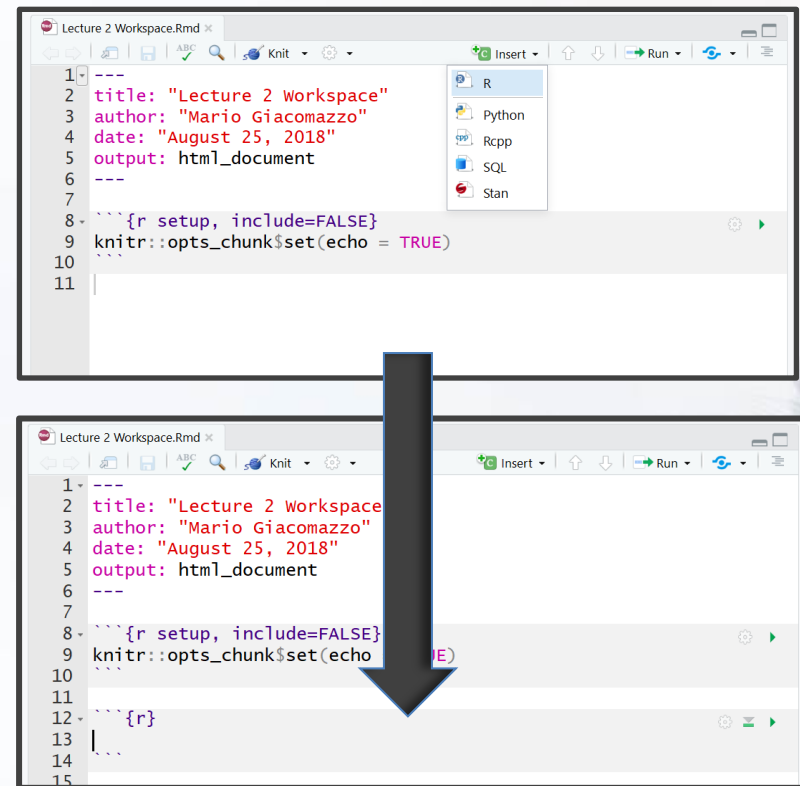
Placing Code in RMarkdown

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

- Method 2: Ctrl+Alt+I

- Method 3: Type ````${r}```` 

Put R code here



Inline Code in RMarkdown

```
`r`{r}  
a <- c(1,2,3)  
`r`
```

The sum of vector `a` is ``r sum(a)``.

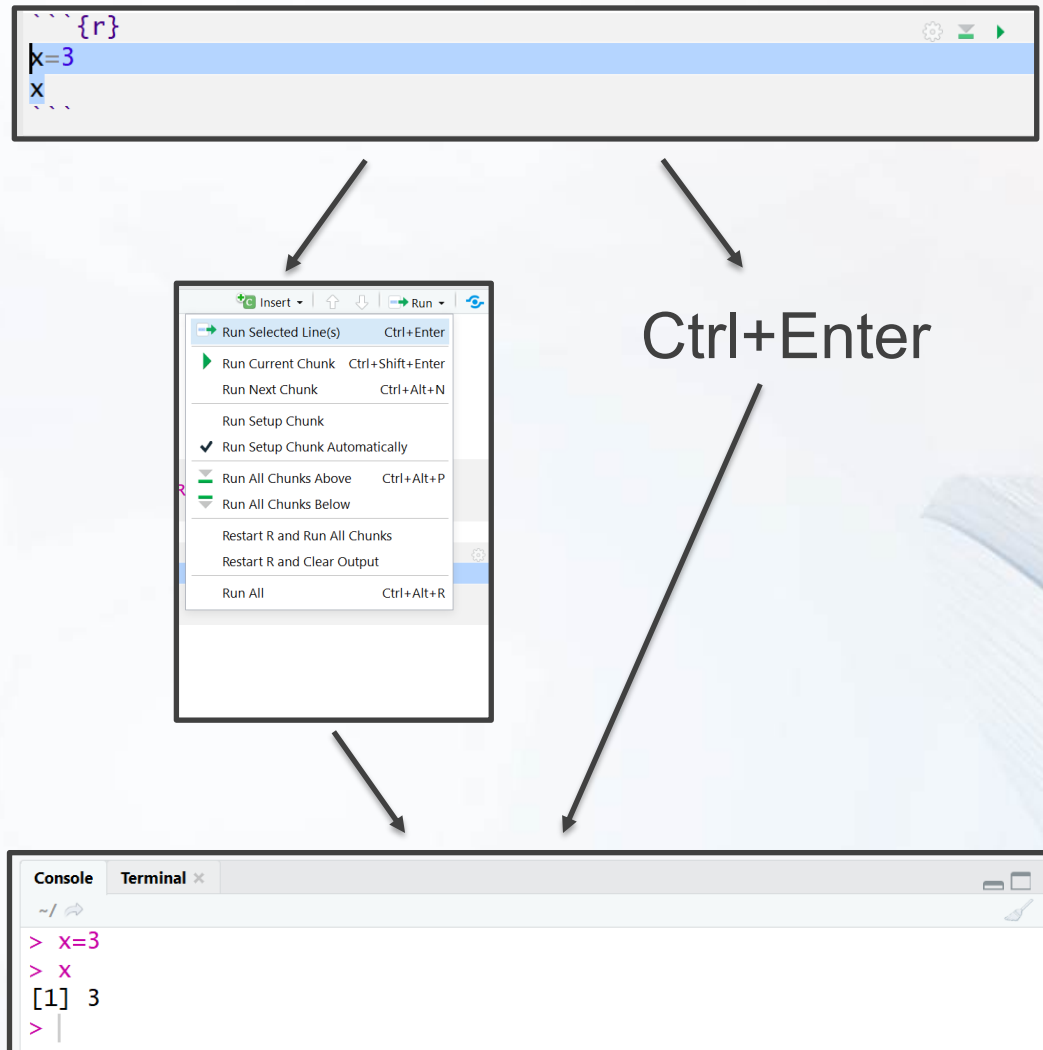
Knit to HTML

```
a <- c(1,2,3)
```

The sum of vector *a* is 6.

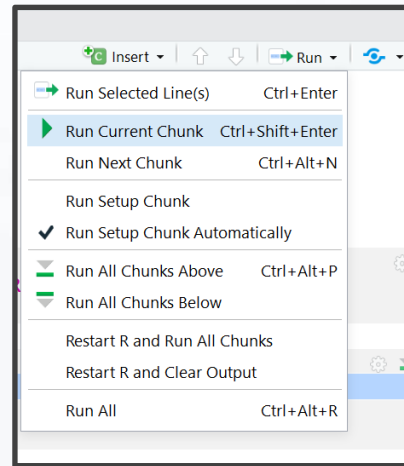
Running Code in RMarkdown

- Various Ways
 - Highlighted Code

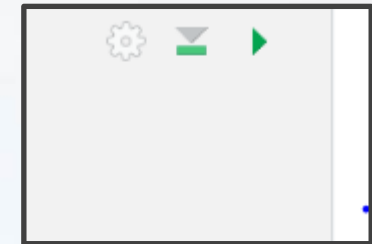


Running Code in RMarkdown

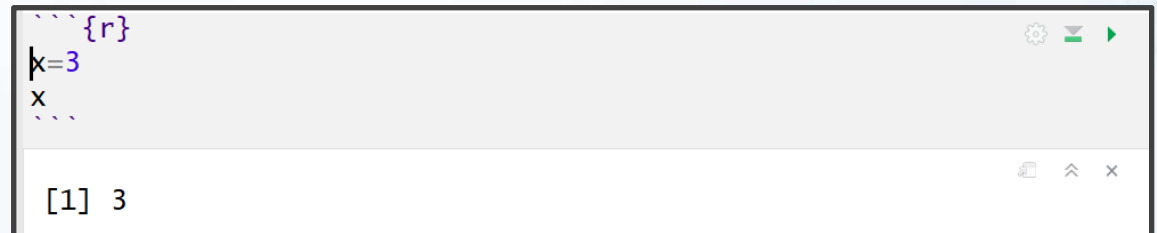
- Various Ways (Cont.)
 - Chunking It (Recommended)



Press
Play

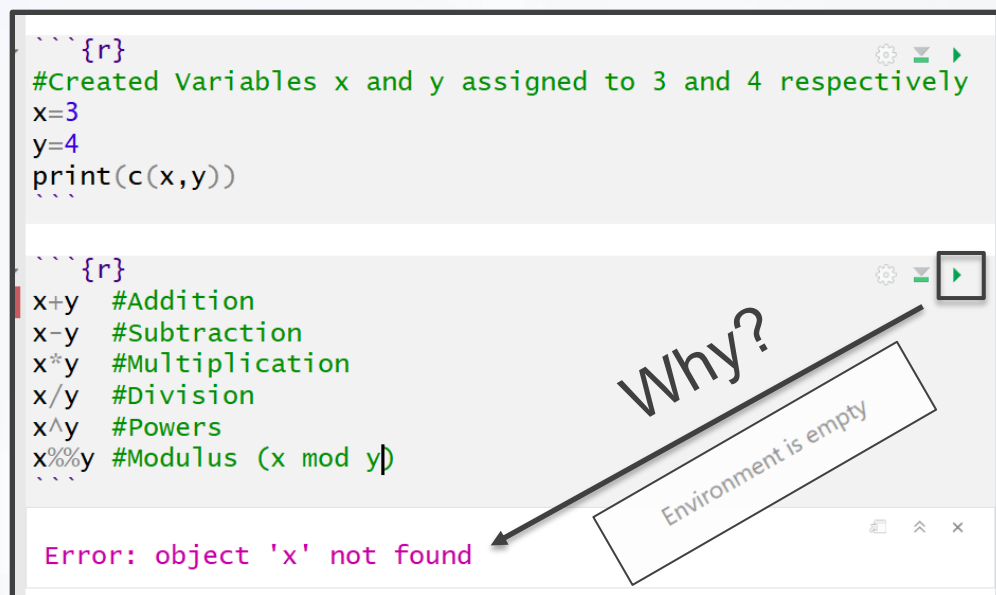


Ctrl+Shift+Enter



Running Code in RMarkdown

- Order Matters



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

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

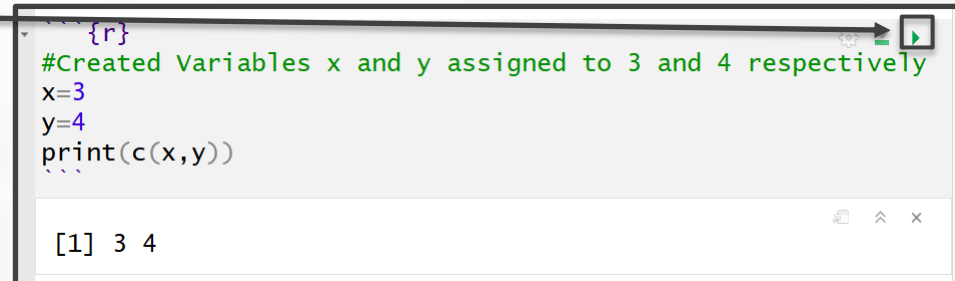
Why?

Environment is empty

Error: object 'x' not found

Running Code in RMarkdown

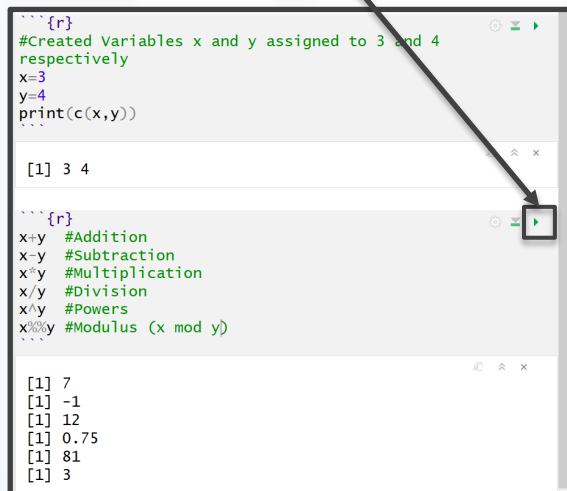
- Order Matters (Cont.)
 - Run First Chunk



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

[1] 3 4

- Then, Run Second Chunk



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

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

[1] 3 4

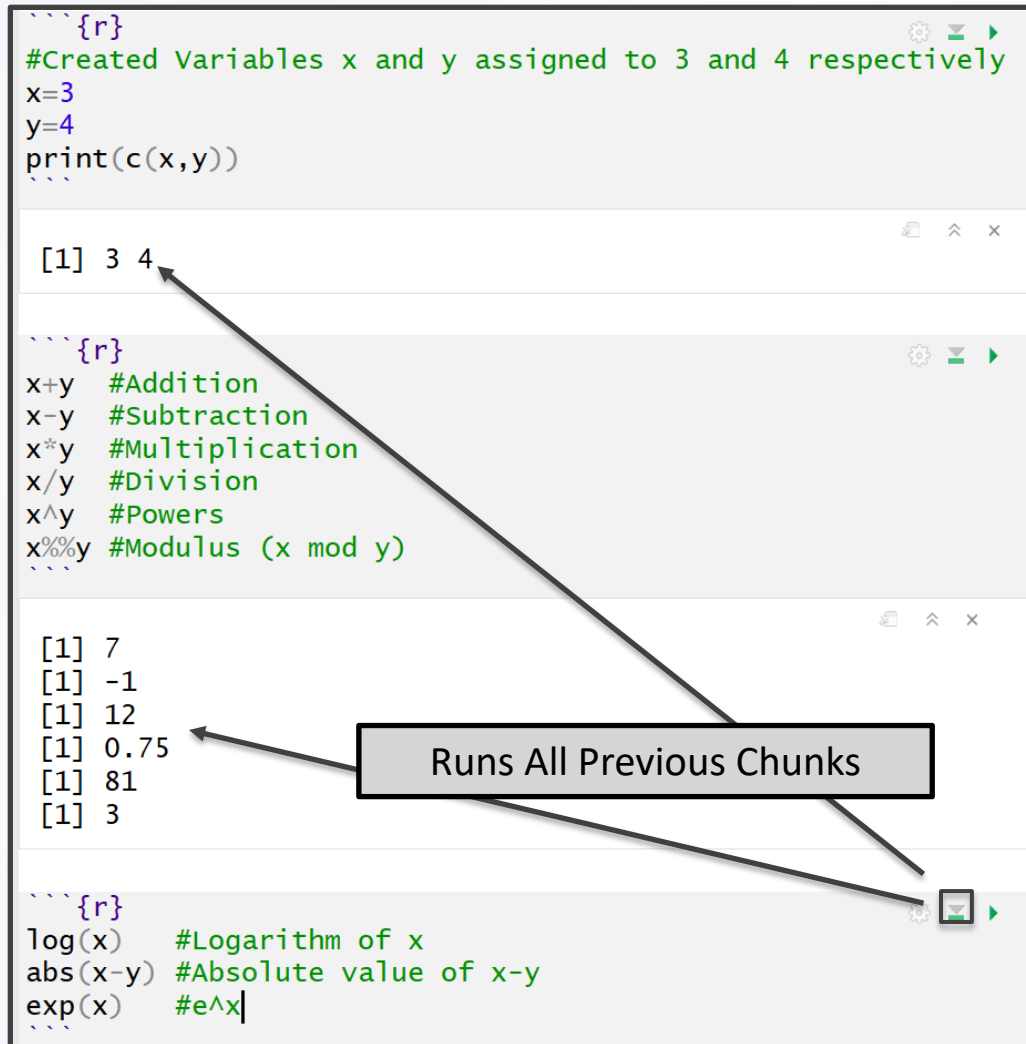
[1] 7
[1] -1
[1] 12
[1] 0.75
[1] 81
[1] 3



Values	
x	3
y	4

Running Code in RMarkdown

- Order Matters (Cont.)
 - Super Chunky



```
{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

Running Code in RMarkdown

- Order Matters (Cont.)
 - Super Chunky (Cont.)

```
```{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|
```
```

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

Then, Run Current Chunk

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

```
{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)]
```

```
[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
```

- Many Types of Objects
 - Vector and Matrix

Objects in R

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

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

6 rows

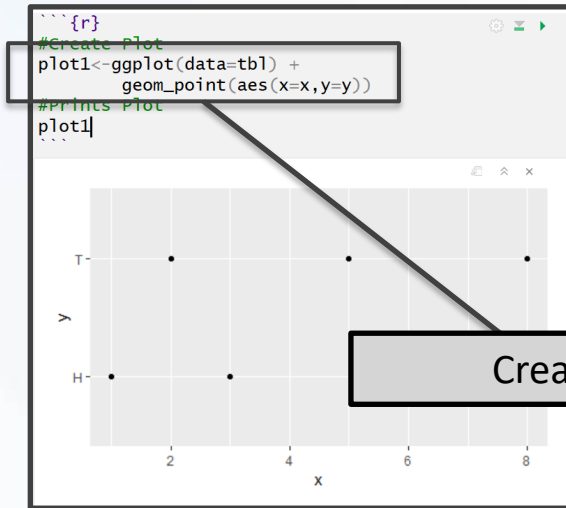
```
{r}
#Create Dataframe named df
df<-data.frame(x=x,y=y)
#Print df
df
```

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

6 rows

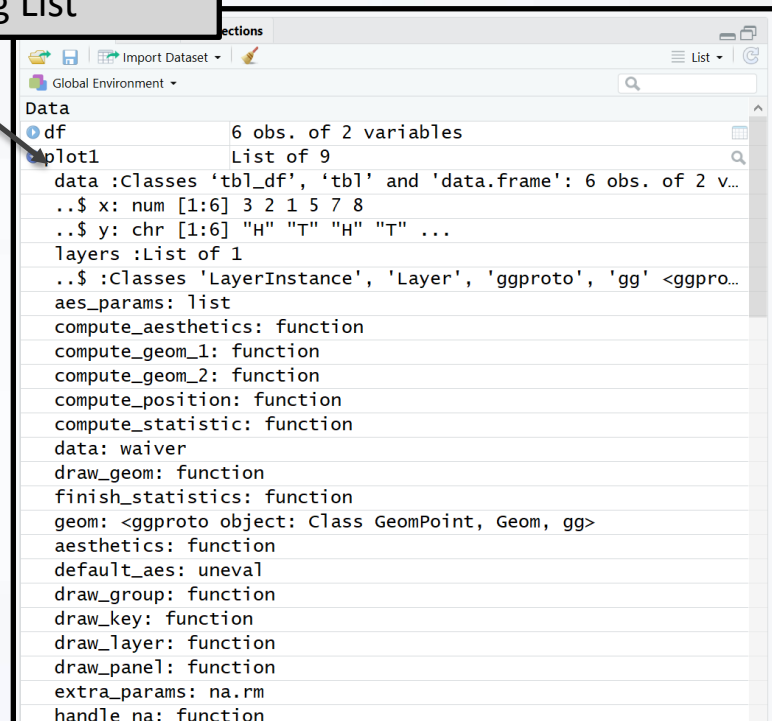
- Many Types of Objects (Cont.)
 - Tibble/Dataframe

Objects in R



Creates Long List

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

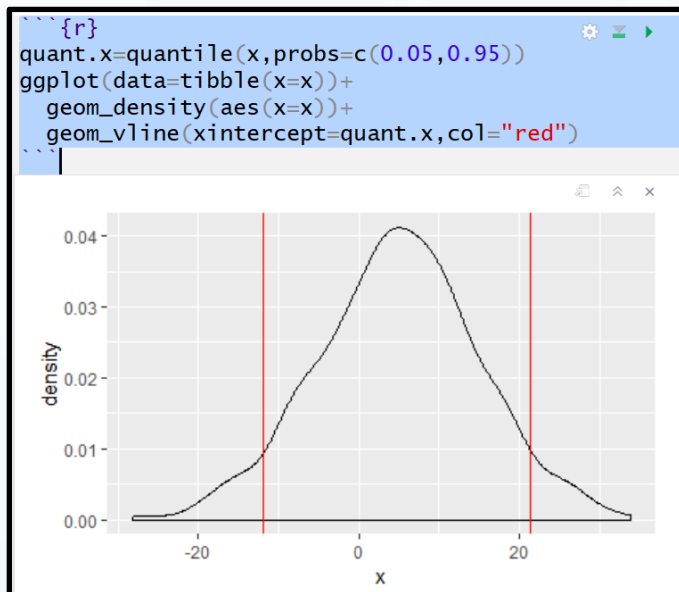


Functions in R

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

Functions in R

```
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
```



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

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