G/B/U 00000 Stories

# Lecture 7: Storytelling, Functions, and Workshop

June 29, 2022

G/B/U 00000 Stories



**Course Administration** 

Good, Bad and Ugly

Telling a Story

Functions in  ${\sf R}$ 





#### **Course Administration**

- 1. July 20: presentations
- 2. July 27: policy briefs due
- 3. Anything else?

G/B/U ●0000

Stories Dooooooooo

#### Next Week's Assignment

Find a storytelling graphic. Post link to google sheet by Wednesday noon.

Finder	Commenter
Linsi G.	Richa
Brandon	Esnold
Dayo	Esther

itories 0000000000

# This Week's Good Bad and Ugly

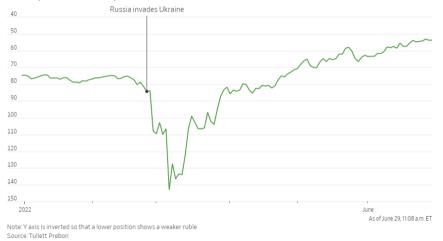
Finder	Commenter
Mary	David
Andres	Jarred
Ensold	Jack

・ロト・「「「・」」・ 「」・ 「」・ (「」・

Admır O Stories 0000000000

# David on Marys's graphic: "Ruble's Rally Stumbles"

How many Russian rubles \$1 buys



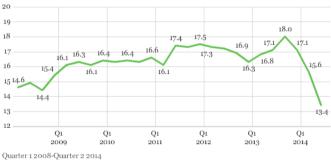
McCabe, Caitlin. "Ruble's Rally Stumbles After Interest Rate Cut," WSJ, May 27, 2022. [link].

#### Jarred on Andres's Uninsured Graphic

Percentage Uninsured in the U.S., by Quarter

Do you have health insurance coverage?

Among adults aged 18 and older



% Uninsured

Quarter 1 2008-Quarter 2 2014 Gallup-Healthways Well-Being Index

Levy, Jenna. "In US, Uninsured Rate Sinks to 13.4% in Second Quarter," Gallup, July 10, 2014. [link]

Jack on Esnold: "How the Trump Administration is Remaking the Courts"



Zengerle, Jason. "How the Trump Administration is Remaking the Courts," *NYT* August 22, 2018. [link].





- 1. Components of a story
- 2. Pulling apart a graph



• Act 1:

◆□ > ◆□ > ◆ Ξ > ◆ Ξ > → Ξ = の < @



# 1. Components of a Story

• Act 1: introduce characters, set up problem



# 1. Components of a Story

#### • Act 1: introduce characters, set up problem

• Act 2:





#### 1. Components of a Story

- Act 1: introduce characters, set up problem
- Act 2: working on the problem, main character changes as a result of problem



#### 1. Components of a Story

- Act 1: introduce characters, set up problem
- Act 2: working on the problem, main character changes as a result of problem
- Act 3:



#### 1. Components of a Story

- Act 1: introduce characters, set up problem
- Act 2: working on the problem, main character changes as a result of problem
- Act 3: climax and resolution of the problem



What Does this Mean for a Policy Brief?

◆□ > ◆□ > ◆ Ξ > ◆ Ξ > → Ξ → のへで



<□> <同> <同> < 目> < 目> < 目> < 目> < 目> □ ○ ○ ○

What Does this Mean for a Policy Brief?

- $1. \ \mbox{Pose the problem, showing its importance}$
- 2. Give evidence for the problem or magnitude
- 3. Propose resolutions



• Storyboard





- Storyboard
- Motivate: identify a problem/question/tension





<□> <同> <同> < 目> < 目> < 目> < 目> < 目> □ ○ ○ ○

- Storyboard
- Motivate: identify a problem/question/tension
- The evidence

- Storyboard
- Motivate: identify a problem/question/tension
- The evidence
  - In Knaflic's book this is the lead-up to a policy
  - In this work, it can be the lead-up to a conclusion

Stories

• Or an establishment of fact

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ の00

Which of Knaflic's Advice is Most Relevant for this Communication?

- Storyboard
- Motivate: identify a problem/question/tension
- The evidence
  - In Knaflic's book this is the lead-up to a policy
  - In this work, it can be the lead-up to a conclusion

Stories

- Or an establishment of fact
- Call to action

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ の00

Which of Knaflic's Advice is Most Relevant for this Communication?

- Storyboard
- Motivate: identify a problem/question/tension
- The evidence
  - In Knaflic's book this is the lead-up to a policy
  - In this work, it can be the lead-up to a conclusion

Stories

- Or an establishment of fact
- Call to action
  - people want a resolution
  - make sure these relate to evidence

- Storyboard
- Motivate: identify a problem/question/tension
- The evidence
  - In Knaflic's book this is the lead-up to a policy
  - In this work, it can be the lead-up to a conclusion

Stories

- Or an establishment of fact
- Call to action
  - people want a resolution
  - make sure these relate to evidence
- All parts should be linked

G/B/

Stories

#### Helpful Tips You Can Apply

- Do slide headers read as a story? aka horizonal alignment
- Vertical alignment within slide agreement
- Use headers to work out your story, then build inside
- Be wary: things that work for a presentation don't always work for a written product



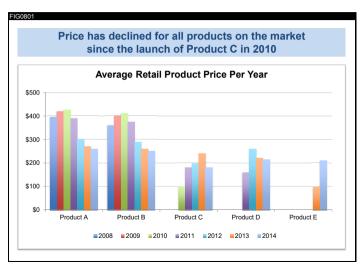
G/B/U 00000 Stories

#### **Common Pitfalls**

- Failure to motivate problem or issue
- Too little definition
- Too much information
- Conclusion without evidence

Stories

#### Telling a Story with Graphics



◆□ > ◆□ > ◆ Ξ > ◆ Ξ > → Ξ = の < @

FIG0811

Stories

# Telling a Story with Graphics

In the next 5 minutes...

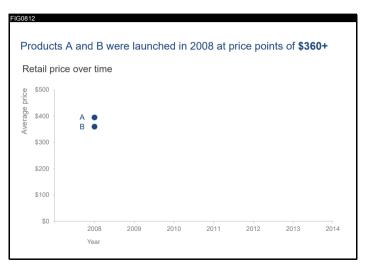
# OUR GOAL:

Understand how prices have changed over time in the competitive landscape.

2 Use this knowledge to inform the pricing of our product.

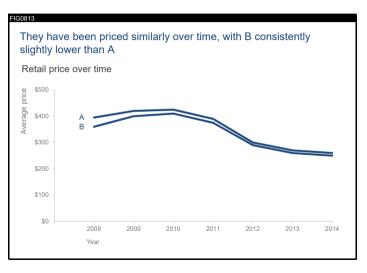
We will end with a specific recommendation.

# Telling a Story with Graphics

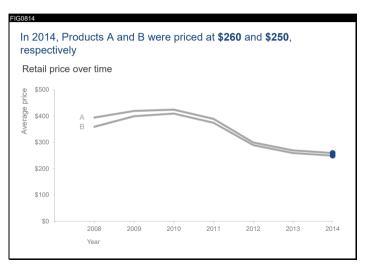


◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

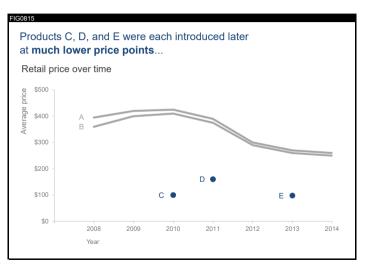
# Telling a Story with Graphics



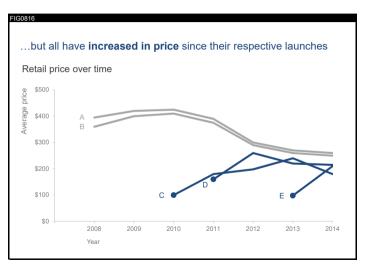
◆□▶ ◆□▶ ◆三▶ ◆三▶ ○□ ● ○○○

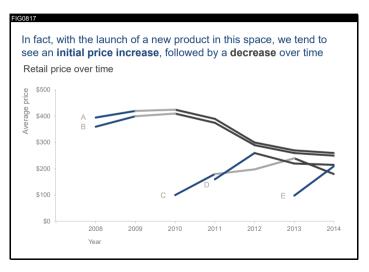


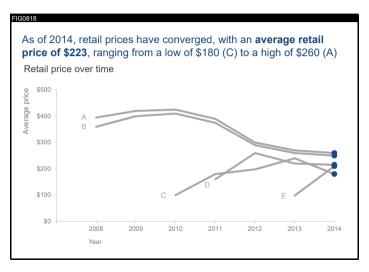
◆□▶ ◆□▶ ◆三▶ ◆三▶ ◆□▶



◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで







# Telling a Story with Graphics





Be Aware that the Presentation Version is Not the Print Version

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ の00

- The final graph of the sequence just before is not a good explanation
- You may need to limit the points along the way
- And make sure you highlight the finding

Admin O Stories

# Telling a Story with Post-its

- Goal today is brainstorming
- Write down your key points
- One per post-it note
- Re-organize and delete as needed
- Tell your story to your group
- I'll wander around to see if you want input



From Knaflic's webpage

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ の00

Admin

G/B/U 00000

Stories

R

To last slide

## Today in R

- 1. Why Functions?
- 2. Defining a function
- 3. Getting things out of a function

・ロト ・日ト ・ヨト ・ヨト ・ヨー うへで

- 4. Modifying a dataframe
- 5. Functions and ggplot

## Why Functions?

Many times, you need to repeat very similar code

- You can copy and paste, but ...
  - Subject to error when you make your small changes

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三 のへぐ

- A real bother when you need to change things
- For example

### Why Functions?

Many times, you need to repeat very similar code

- You can copy and paste, but ...
  - Subject to error when you make your small changes

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ ● のへで

- A real bother when you need to change things
- For example
  - Make many similar graphs
  - Load multiple files with similar names
  - Create summary stats with different subsets

#### Good Functions

- 1. Make code more readable
- 2. Avoid coding errors
- 3. Make you more productive

From "Nice R Code' ' on github.

・ロト ・日ト ・ヨト ・ヨト ・ヨー うへで

## However: Never Start Writing a Function by Writing a Function

- Get one version of your code working first
- Then build the function
- When you've been programming for two years, try the function first

#### What We Cover About Functions

- 1. Defining a function
- 2. Calling a function
- 3. Getting things out of a function

- 4. Modifying a dataframe
- 5. Functions and ggplot

## 1. Defining a Function

```
function.name <- function(arg1, arg2){
    # stuff your function does
}</pre>
```

- function.name: what you call the function
- function: needed to tell R this is a function
- arg1: first argument of the function
- arg2: second argument of the function
- inside the curly braces: what you want the function to do

(日) (四) (王) (王) (王) (王)

## Simple Function Example

```
summer <- function(x,y){
    x^y
}</pre>
```

- function name?
- arguments?
- body of the function?

summer <- function(x,y){
 x^y
}</pre>

◆□▶ ◆□▶ ◆□▶ ◆□▶ □ のへで

summer(x = 2, y = 3)

summer <- function(x,y){
 x^y
}
summer(x = 2,y = 3)</pre>

## [1] 8

summer <- function(x,y){
 x^y
}
summer(x = 2,y = 3)
## [1] 8
summer(x = 3,y = 2)</pre>

▲□▶ ▲圖▶ ▲匡▶ ▲匡▶ ― 匡 … のへで

summer <- function(x,y){
 x^y
}
summer(x = 2,y = 3)
## [1] 8
summer(x = 3,y = 2)
## [1] 9</pre>

▲□▶ ▲圖▶ ▲匡▶ ▲匡▶ ― 匡 … のへで

#### 3. Getting things out of a function

Suppose you want to use the output of summer elsewhere in your program

- Functions "return' the last line
- "Return' means makes a value that exists outside of the function
- Best explained via example

## 3. Getting things out of a function

Suppose you want to use the output of summer elsewhere in your program

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

- Functions "return' the last line
- "Return' means makes a value that exists outside of the function
- Best explained via example

However, if you write to disk, that will exist outside the function

- save using ggplot or
- write using write\_csv

```
summer2 <- function(x,y){
    o1 <- x^y
    o1
    print(paste0("o1 is ", o1))
    o2 <- x + y
    print(paste0("o2 is ", o2))
}</pre>
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

summer2(x = 1, y = 2)

```
summer2 <- function(x,y){
    o1 <- x^y
    o1
    print(paste0("o1 is ", o1))
    o2 <- x + y
    print(paste0("o2 is ", o2))
}</pre>
```

```
summer2(x = 1,y = 2)
## [1] "o1 is 1"
## [1] "o2 is 3"
```

```
summer2 <- function(x,y){
    o1 <- x^y
    o1
    print(paste0("o1 is ", o1))
    o2 <- x + y
    print(paste0("o2 is ", o2))
}</pre>
```

```
summer2(x = 1,y = 2)
## [1] "o1 is 1"
## [1] "o2 is 3"
What if I write o2?
```

```
summer2 <- function(x,y){
    o1 <- x^y
    o1
    print(paste0("o1 is ", o1))
    o2 <- x + y
    print(paste0("o2 is ", o2))
}</pre>
```

```
summer2(x = 1,y = 2)
## [1] "o1 is 1"
## [1] "o2 is 3"
What if I write o2?
o2
```

```
summer2 <- function(x,y){
    o1 <- x^y
    o1
    print(paste0("o1 is ", o1))
    o2 <- x + y
    print(paste0("o2 is ", o2))
}</pre>
```

```
summer2(x = 1, y = 2)
```

## [1] "o1 is 1" ## [1] "o2 is 3"

What if I write o2?

o2

## Error in eval(expr, envir, enclos): object 'o2' not found

```
summer2 <- function(x,y){
    o1 <- x^y
    print(paste0("o1 is ", o1))
    o2 <- x + y
    print(paste0("o2 is ", o2))
}</pre>
```

o3 <- summer2(x = 1, y = 2)

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

## [1] "o1 is 1" ## [1] "o2 is 3"

```
summer2 <- function(x,y){
    o1 <- x^y
    print(paste0("o1 is ", o1))
    o2 <- x + y
    print(paste0("o2 is ", o2))
}</pre>
```

```
o3 <- summer2(x = 1, y = 2)
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

## [1] "o1 is 1"
## [1] "o2 is 3"
What if | call o3?

```
summer2 <- function(x,y){
    o1 <- x^y
    print(paste0("o1 is ", o1))
    o2 <- x + y
    print(paste0("o2 is ", o2))
}</pre>
```

```
o3 <- summer2(x = 1, y = 2)
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

## [1] "o1 is 1"
## [1] "o2 is 3"
What if I call o3?
o3

## [1] "o2 is 3"

```
summer2 <- function(x,y){
    o1 <- x^y
    print(paste0("o1 is ", o1))
    o2 <- x + y
    #print(paste0("o2 is ", o2))
}</pre>
```

```
o3 <- summer2(x = 1, y = 2)
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

## [1] "o1 is 1"

```
summer2 <- function(x,y){
    o1 <- x^y
    print(paste0("o1 is ", o1))
    o2 <- x + y
    #print(paste0("o2 is ", o2))
}</pre>
```

o3 <- summer2(x = 1, y = 2)

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

## [1] "o1 is 1"

What if I call o3?

```
summer2 <- function(x,y){
    o1 <- x^y
    print(paste0("o1 is ", o1))
    o2 <- x + y
    #print(paste0("o2 is ", o2))
}</pre>
```

```
o3 <- summer2(x = 1, y = 2)
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

## [1] "o1 is 1"

What if I call o3?

oЗ

## [1] 3

```
summer2 <- function(x,y){
    o1 <- x^y
    print(paste0("o1 is ", o1))
    o2 <- x + y
    print(paste0("o2 is ", o2))
    return(o2)
}</pre>
```

```
o3 <- summer2(x = 1, y = 2)
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

## [1] "o1 is 1" ## [1] "o2 is 3"

```
summer2 <- function(x,y){
    o1 <- x^y
    print(paste0("o1 is ", o1))
    o2 <- x + y
    print(paste0("o2 is ", o2))
    return(o2)
}</pre>
```

```
o3 <- summer2(x = 1, y = 2)
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

## [1] "o1 is 1"
## [1] "o2 is 3"
What if | call o3?

```
summer2 <- function(x,y){
    o1 <- x^y
    print(paste0("o1 is ", o1))
    o2 <- x + y
    print(paste0("o2 is ", o2))
    return(o2)
}</pre>
```

```
o3 <- summer2(x = 1, y = 2)
```

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

## [1] "o1 is 1"
## [1] "o2 is 3"
What if | call o3?

oЗ

## [1] 3

## 4. What About Modifying a Dataframe?

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○□ のへで

# What About Modifying a Dataframe? nkd

##		year	defectors
##	1	2000	0
##	2	2001	0
##	3	2002	1
##	4	2003	0
##	5	2004	0
##	6	2005	0
##	7	2006	0
##	8	2007	0
##	9	2008	2
##	10	2009	0
##	11	2010	1
##	12	2011	0
##	13	2012	3
##	14	2013	0
μц		0044	^

#### First Try

#### First Try

▲□▶ ▲圖▶ ▲필▶ ▲필▶ - 亘 - のへで

How do you call this?

#### First Try

```
How do you call this?
```

```
addone(fixyear = 2002)
addone(fixyear = 2005)
nkd
```

##		year	defectors
##	1	2000	0
##	2	2001	0
##	3	2002	1
##	4	2003	0
##	5	2004	0

##		year	defectors
##	1	2000	0
##	2	2001	0
##	3	2002	100
##	4	2003	0
##	5	2004	0
##	6	2005	100
	-	~~~~	^

Create a new variable as a function of an old variable

```
This doesn't work:
```

```
multiplypls <- function(varo){
    nkd$new_varo <- nkd$varo * 5
    return(nkd)
}
nkd2 <- multiplypls(var = defectors)</pre>
```

R doesn't know to plug in defectors for varo in the dataframe\$variable construction

Create a new variable as a function of an old variable

Instead

```
multiplypls <- function(varo){
   nkd[[paste0("new_",varo)]] <- nkd[[varo]] * 5
   return(nkd)
}
nkd2 <- multiplypls(var = "defectors")
head(nkd2)</pre>
```

▲□▶ ▲圖▶ ▲필▶ ▲필▶ - 亘 - のへで

##		year	defectors	new_defectors
##	1	2000	0	0
##	2	2001	0	0
##	3	2002	100	500
##	4	2003	0	0
##	5	2004	0	0
##	6	2005	100	500

## 5. And a Word of Warning About ggplot()

- many tidyverse commands, including ggplot() use non-standard evaluation
- for your purposes, that means that these command don't always work in expected ways in functions

◆□▶ ◆□▶ ◆三▶ ◆三▶ 三三 - のへで

BUT there are work-arounds – see tutorial

#### Bottom Line

- Use functions!
- Write a non-function example first

◆□▶ ◆□▶ ◆□▶ ◆□▶ □ のへで

- Test
- Write the function
- Check output



# Now See If I Can Program in Real Time!

- 1. What would you like a function to do?
- 2. Up for more or less anything that can involve very sample data we make



G/B/U 00000 Stories 0000000000 

# Next Lecture

- Next week: Scatter plots and color!
- Read
  - Skim origins of scatterplot
  - Some of Few Ch. 10, and Few Ch. 11
  - Chang Chapter 5



G/B/U 00000 Stories 0000000000 

# Today's Workshop

- 1. Do the Post-it challenge for yourself
- 2. Share your post-it output
- 3. Share your graphics ideas

Admin 0 tories 0000000000 

# Telling a Story with Post-its

- Goal today is brainstorming
- Write down your key points
- One per post-it note
- Re-organize and delete as needed
- Tell your story to your group
- I'll wander around to see if you want input



From Knaflic's webpage

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ の00