

Recap III

The R Bootcamp
Twitter: [@therbootcamp](https://twitter.com/therbootcamp)
September 2017

Tidying

In this introduction you will learn...

...how to write clean,
documented code.

...to understand errors
(and warnings).

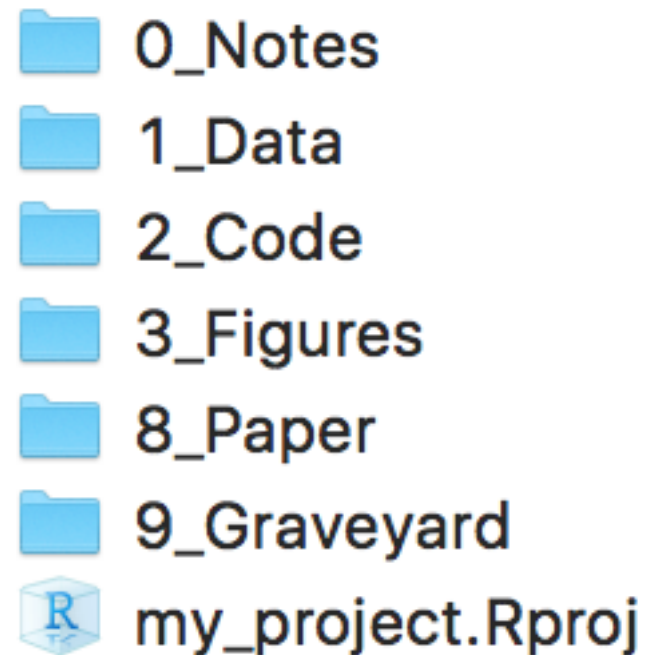
...how to deal with missing
values.



source <https://build2be.com/>

Project structure

Good, clean, documented code begins with a **project** and a **folder structure**.



7 most frequent errors

According to stackoverflow.com

Error	Example	Description
'could not find function'	<code>lenth(my_vec)</code>	There is a typo in the function name or that a package has not been loaded.
'error in if'	<code>if(NA == 2) 2 + 2</code>	The object in the <code>if</code> clause is non-logical or NA.
'error in eval'	<code>lm(fefq~wzfe)</code>	An object is used that does not exist.
'cannot open()'	<code>read_csv('hjht.txt')</code>	The file does not exist. Could be a typo or a missing filepath.
'no applicable method'	<code>predict('efwe')</code>	A 'generic function' has not been defined for this type/class
'subscript out of bounds'	<code>a <- matrix(c(1,2)); a[2,2]</code>	R tried to access an element (or variable) that does not exist
package errors		Occur when R is unable to install, compile, or load a package. Often this means that some software background is missing.

Assignments & Comments

For **assignments** use `<-`, not `=`. However, to specify arguments in functions use `=`.

```
# Good
x <- 24324

# Bad
x = 24324
```

Comment each line of your code. To break up your code in chunks use `-` or `=`.

```
# Plot data -----

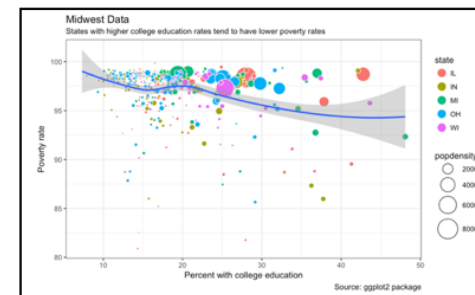
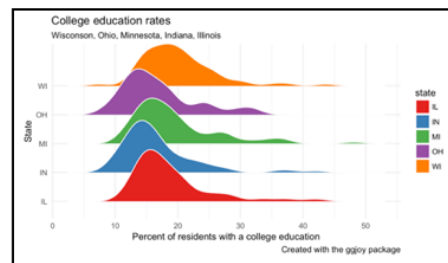
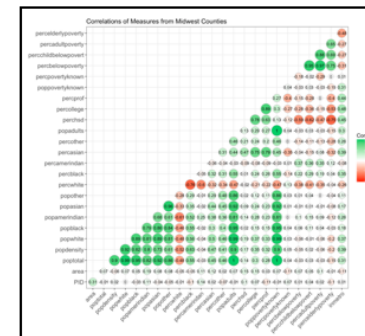
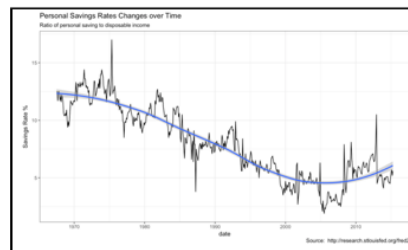
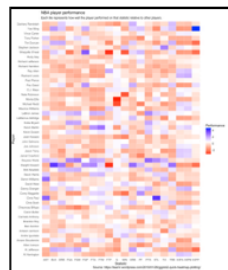
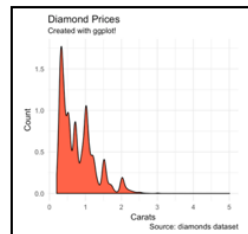
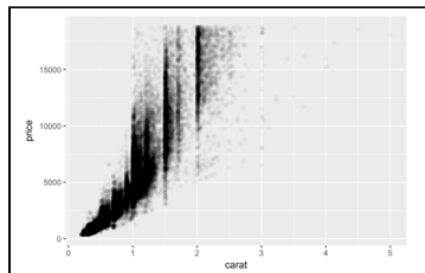
this_is <- "pseduo_code"
my_function(arg1 = x,
            arg2 = y)

# Plot data =====

this_is <- "pseduo_code"
```

You can do amazing plots in R!

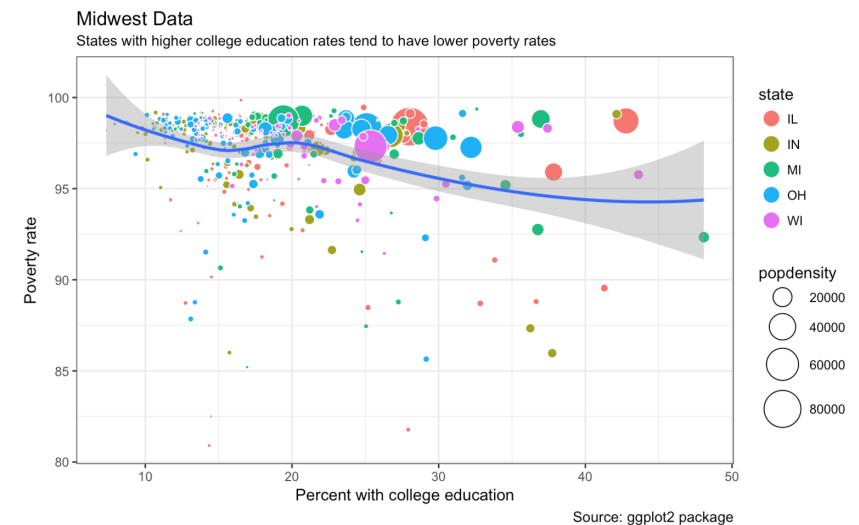
- As good as R is for statistics, it's as good if not better for plots.



Grammar of Graphics

The Grammar of graphics breaks down plots into several key pieces:

aesthetics	Description
Data	What dataframe contains the data?
Aesthetics	What does the x-axis, y-axis, color (etc) represent?
Geometries	What kind of geometric object do you want to plot?
Facets	Should there be groups of plots?
Statistics	What statistic summaries / transformations should be done?
Coordinates	What is the scale of the axes?
Theme	What should the overall plot look like?



ggplot2

How do we make elegant, easy to program plots according to the grammar of graphics in R?

Answer: ggplot2

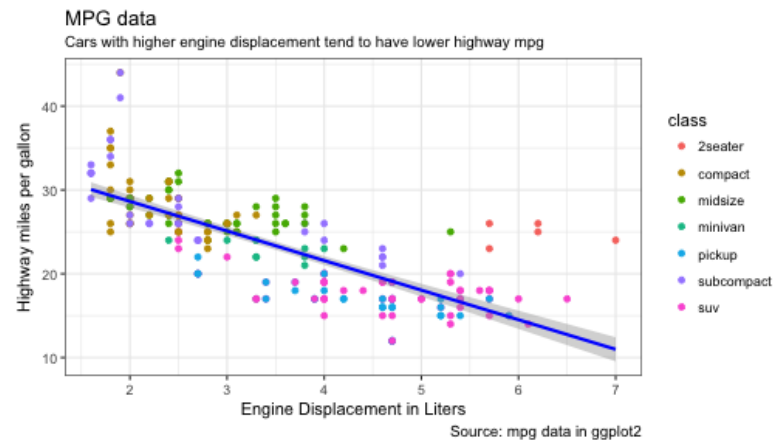
By far one of the most popular R packages, used to generate the vast majority of plots from R.

Hadley Wickham



Final result!

```
ggplot(data = mpg,
       mapping = aes(x = displ, y = hwy, col = class)) +
  geom_point() +
  geom_smooth(col = "blue", method = "lm")+
  labs(x = "Engine Displacement in Liters",
       y = "Highway miles per gallon",
       title = "MPG data",
       subtitle = "Cars with higher engine displacement tend to have lower highway mpg",
       caption = "Source: mpg data in ggplot2") +
  theme_bw()
```



What is Shiny?

A **Shiny** app is a web page (**UI**) connected to a computer running a live R session (**Server**)



Users can manipulate the UI, which will cause the server to update the UI's displays (by running R code).

Source: <http://shiny.rstudio.com/images/shiny-cheatsheet.pdf>

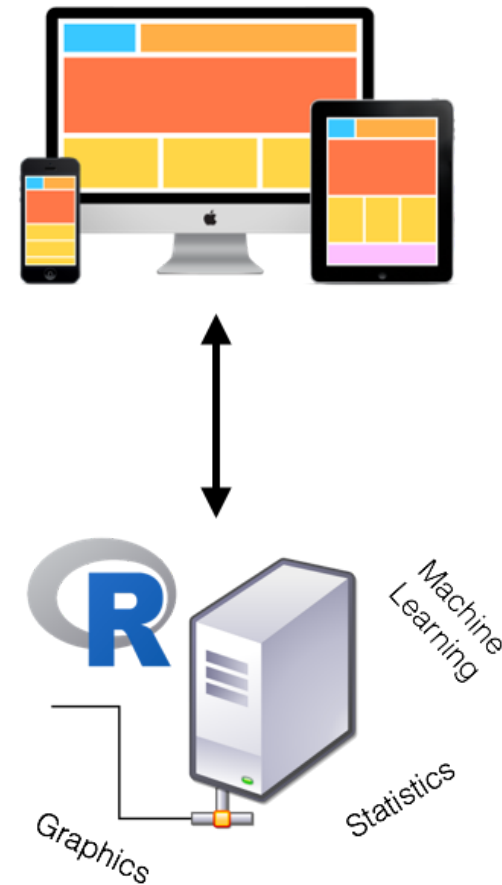
Structure of a Shiny App

User Interface: **ui()**

```
ui <- (  
# Set overall layout  
# Add widgets,  
# Display plots  
)
```

R Server: **server()**

```
server <- function(input, output) {  
# Access Data  
# Run analyses  
# Create and render plots  
}  
}
```



User Interface