

Agricultural Data Analysis & Visualization in R



Lily Northcutt
Data Scientist

Chile Pepper Conference 2026



New Mexico
February 5, 2026

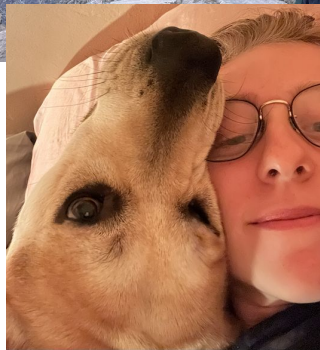
Hi 🙋, I'm Lily

I'm a data scientist with an academic background in Mathematics and Machine Learning.

I've worked with NMSU's Chile Breeding Program & CPI, as well as NMSU's Entomology, Plant Pathology, and Weed Science group.

I primarily work in python now but I started programming with R and I still love the language.

When I'm not working can find me trying to pet dogs on a hike, singing karaoke (poorly), learning to knit, and trying out new restaurants!



Working With Data

Part 1

1. Setting up the environment
2. Reading in data
3. Tidy and Transform our data
4. Summarize the data

- This is an **interactive** and **hands-on** workshop!
- Not a lecture - be vocal, ask questions
- Need help?
 - Your Resources: Talk to your neighbor, **google**, ask me, [R for Data Science Book](#), [R Cheatsheets](#)
- **There are stickers**



This slide means you
have work to do!



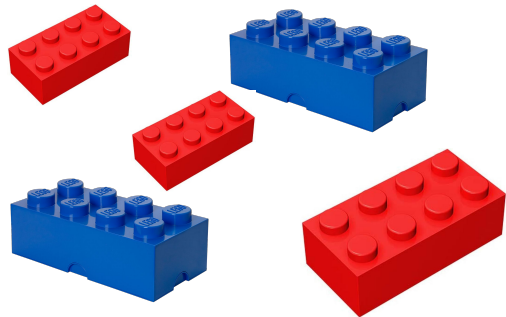
(practice run)

Introduce yourself to your
neighbors

My Goal

My goal is to introduce some of (what I consider to be) the **essential** building blocks for working in R.

Through practice and creativity, you will be able to use the blocks to make something spectacular



What R provides

you



What
you will
build

Section A: Intro

R is a programming language

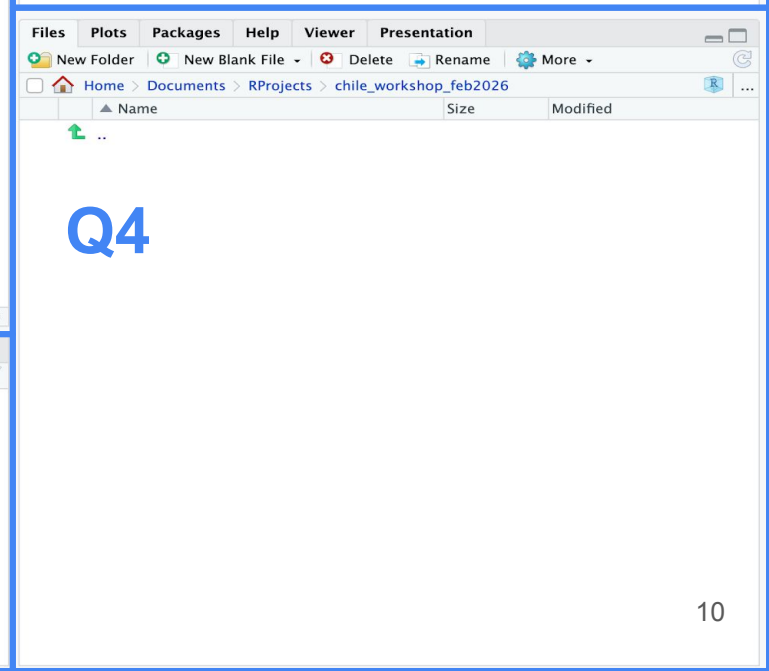
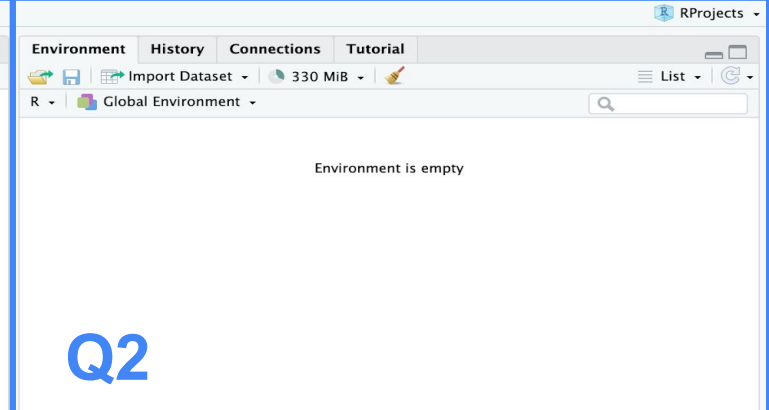
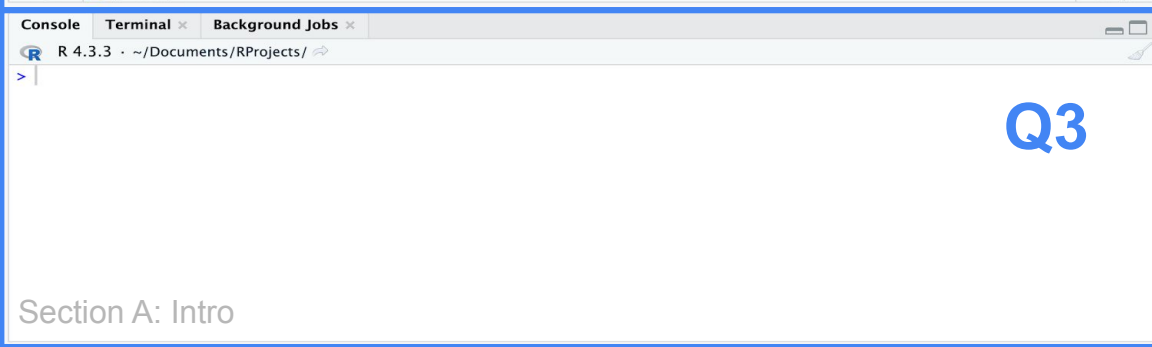
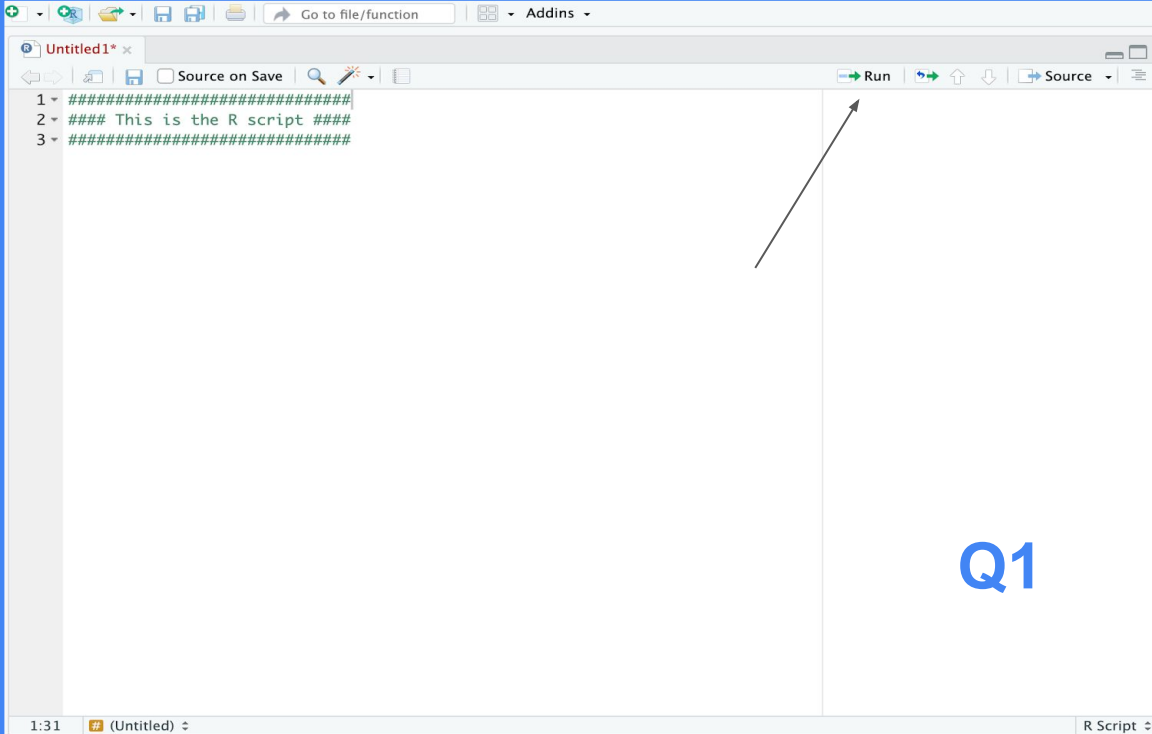
- Free, open-source programming language for statistics
- Widely used in agricultural research, data analysis, modeling
- Great for reproducibility - your code is your methods section!



RStudio is an Integrated Development Environment (IDE)

- The user-friendly interface to using R!





Quick Demo: Ways to run code in RStudio

1. Console code

- a. Type `2 + 2` in console → Enter

2. Create a new .R file

- a. File → New File → R Script
- b. Type `2 + 2` in script, highlight, click "Run" (or Ctrl/Cmd + Enter)
- c. Now press Source

Scripts are saved and reproducible, console is typically for a quick line of code!

Section B: Project Set-Up

Github Repo 🌶️

github.com/lilynorthcutt/chile_workshop_feb2026

Step 1: Navigate to my personal github repo for this workshop

(use link above or use QR or use bit.ly link)

Step 2: Go into the `working_with_data` folder and find the `problems.R` file



bit.ly/4kgUqqV

Quick Demo: Project Set-Up

1. Create a project

- a. File → New Project → New Directory →
New Project

2. (Install your Packages)

- a. To install the “dplyr” package you
would run:
 - i. `install.packages("dplyr")`

3. Open an .R file

What kind of data can **R** use?

- Built-in datasets & Downloaded Package datasets
- CSV files (.csv)
- Excel files (.xlsx, .xls)
- Text files (.txt)
- Database connections (SQL)
- Web data (APIs, web scraping)

TODAY: We're using package datasets (agridat)



Follow Along Demo in RStudio

1. Load the Package
2. Explore the `agridat` Documentation in the Help Tab

```
# TODO: Follow along reading in data:  
# 1. Load Required Package  
library("agridat")  
  
# Explore the agridat documentation  
?agridat
```




Follow Along Demo in RStudio

1. Read in the `yates.oats` data
2. Check out the quick ways to view our data

Q: If you were a researcher and this was your experiment, what questions would you want to know?

Section C: Working with Data Basics

Tidy data is a way to organize tabular data in a consistent data structure across packages. A table is tidy if:



Each **variable** is in its own **column**

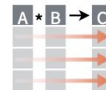
&



Each **observation**, or **case**, is in its own row

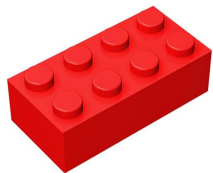


Access **variables** as **vectors**



Preserve **cases** in vectorized operations

Accessing Columns and Basic Functions



1. Access/Select our Columns

Concept: Data frames are like spreadsheets with named columns

Two ways to access columns:

- dollar sign: `oats$yield`
- bracket notation: `oats[, "yield"]`



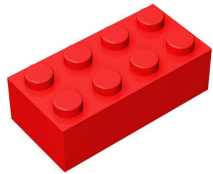
2. Basic Functions

- `mean()`
- `median()`
- `sum()`
- `sd()` - standard deviation
- `length()` - count of values

Demo in RStudio



PROBLEM 1 in the code



filter()

Concept: Selects specific ROWS based on the conditions you set

In order to specify your conditions, you will use:

Comparison Operators

- == means "equals" (not =)
- ! means NOT
- > < >= <= for comparisons

Combining Operators

- & means "and"
- | means "or"

```
nitro > 0.2 & gen == "Marvelous"
```

Translates to:

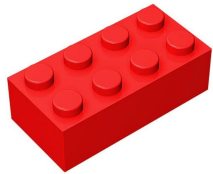
```
gen != "Marvelous" &  
(grain > 30 | straw > 30)
```

Translates to:

Demo in RStudio



PROBLEM 2 in the code



mutate()

Concept: Creates NEW columns or modifies existing ones in your dataset

- We will use our previous blocks we acquired inside of mutate to build something bigger and better



case_when()

Concept: We can specify different 'outs' based on different 'ins'

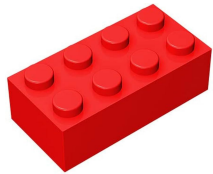
```
case_when(  
  condition1 ~ "Category 1",  
  condition2 ~ "Category 2",  
  condition3 ~ "Category 3",  
  TRUE ~ "Everything else"  
)
```

```
case_when(  
  nitro == 0 ~ "Control",  
  nitro <= 0.2 ~ "Low",  
  nitro <= 0.4 ~ "Mid",  
  TRUE ~ "High"  
)
```

Translate to:



PROBLEM 3 in the code



merge ()

Concept: Combining datasets by matching rows on common column(s)

Think of it like adding additional information from a different table based on something in common - will commonly be metadata

Example: We may merge a key with full variety or treatment names into a large datasets from a trial

Important! The matching column must have *THE SAME NAME* in both datasets



`data.frame()`

Instead of only reading in data, we can always custom create it in R



v_3	n_3 156	n_2 118		n_2 109	n_3 99	v_3
	n_1 140	105		n_0	70	
	n_0	130	B4	n_0	94	
v_1	n_3	157		n_3 126	n_1 82	v_2
v_2	n_0 117	n_1 114		n_1 90	n_2 100	v_1
	n_2 161	n_3 141		n_3 116	n_0 62	

v_3	n_2 104	n_0 70		n_3 96	n_0 60	v_2
	n_1 89	n_3 117		n_2 89	n_1 102	
	n_3 1	74	B5	n_2	86	
v_1	n_1	81		n_0	64	v_1
v_2	n_1 103	n_0 64		n_2 132	n_3 124	v_3
	n_2 132	n_3 133		n_1 129	n_0 89	

v_2	n_1 108	n_2 126		n_2 118	n_0 53	v_1
	n_3 149	n_0 70		n_3	74	
	n_3 1	24	B6	n_3	86	
v_3	n_2 1	96		n_0	82	v_2
v_1	n_0 61	n_3 100		n_0 97	n_1 99	v_3
	n_1 91	n_2 97		n_2 119	n_3 121	

← rows →

Area of each plot : 1/80 acre. (28.4 links \times 44 link rows.)

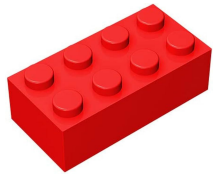
FIG. 2.

*Oats Variety and Manuring Experiment.
Plan and yields in quarter lb.*



PROBLEM 4 in the code

Section D: Summarizing Data



`summarize()`

Concept: Collapses your dataset down to whatever type of summary you specify

This will commonly be things like `mean()`, `min()`, `max()`, `nrow()`



`group_by()`

`summarize()` really SHINES when used with `group_by()`


Concept: Calculate summaries FOR EACH GROUP instead of the whole dataset



PROBLEM 5



BONUS CHALLENGE!

Your Turn  30

Exploring & Visualizing Data

Part 2

1. Using %>% operator
2. Intro to ggplot2
 - a. Common plots
 - b. Customizing
 - c. Error Bars
 - d. Facet Plots
3. Field Layout Visualization

Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813.

Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite. Paris, le 20 Novembre 1869.

Les nombres d'hommes présents sont représentés par les largeurs des zones colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zones. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M. M. Chiers, de Légar, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.

Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davout, qui avaient été détachés sur Minsk et Mohilow et qui rejoignent Orscha et Witebsk, avaient toujours marché avec l'armée.

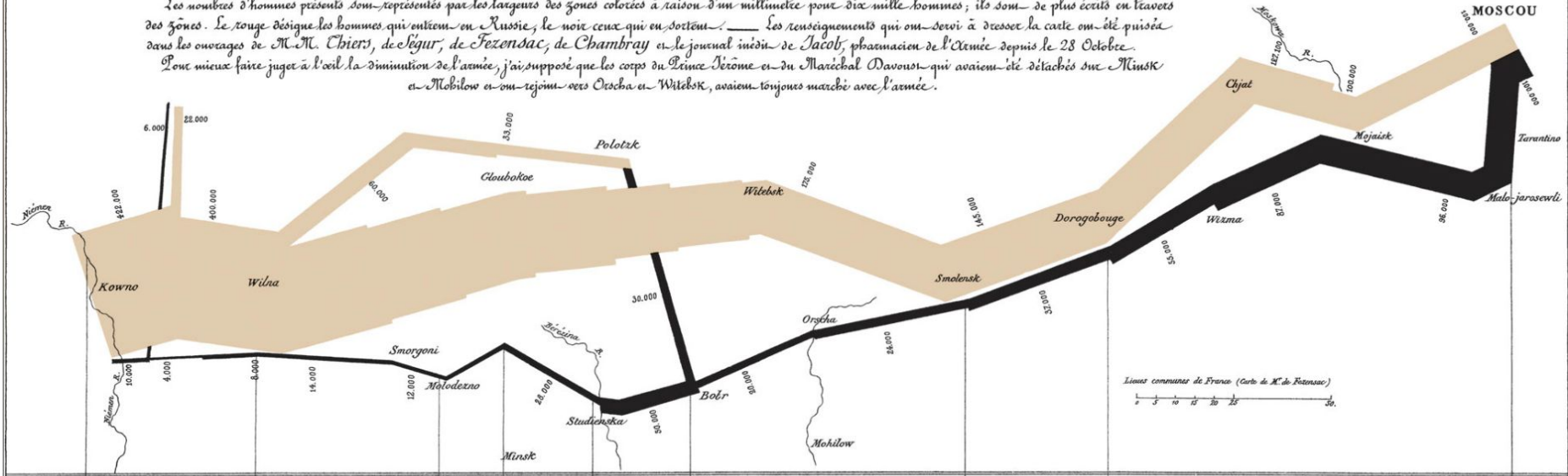
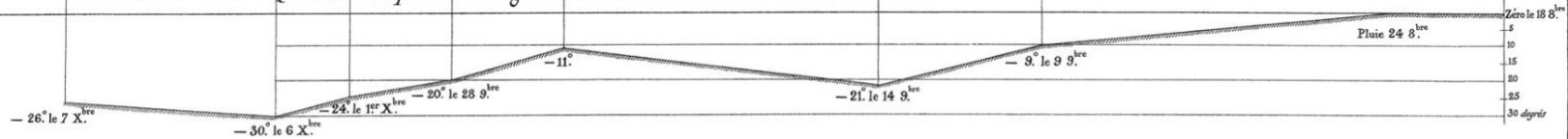


TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.



Autog. par Regnier, 8. Par. 5^{me} Marie 5^{me} 0^{me} à Paris.

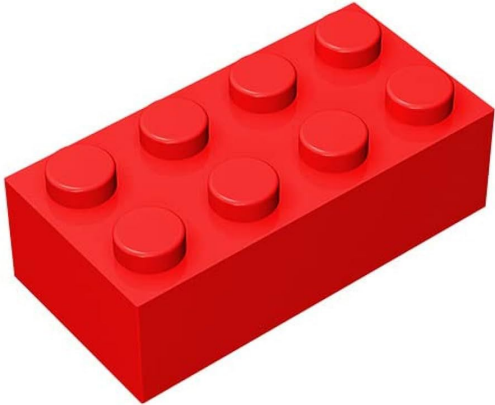
Imp. Lit. Regnier et Bourdat.

Map by Charles Joseph Minard

<https://ageofrevolution.org/200-object/flow-map-of-napoleons-invasion-of-russia/>

- This is an **interactive** and **hands-on** workshop!
- Will be faster paced than last section
- Need Help?
 - Your Resources: Talk to your neighbor, google, ask me, [R for Data Science Book](#), [R Cheatsheets](#)
- There are still stickers!

Section A: Pipe Operator $\%>\%$



Pipe operator: `%>%`

Problem: Nested code is hard to read, constantly assigning new variables is not sustainable

Solution: `%>%`

Concept: A way to chain operations together so code reads like a sentence

“... and then ...”

Basic Syntax:

```
data %>%  
  function1() %>%  
  function2() %>%  
  function3()
```



PROBLEM 1

Section B: Intro to ggplot2

<https://ggplot2-book.org/facet.html>

<https://rstudio.github.io/cheatsheets/data-visualization.pdf>



What is `ggplot2`?

Library of for creating graphics
based on "The Grammar of
Graphics"

`ggplot2` Building Beautiful Plots

We build plots in 3 main layers:

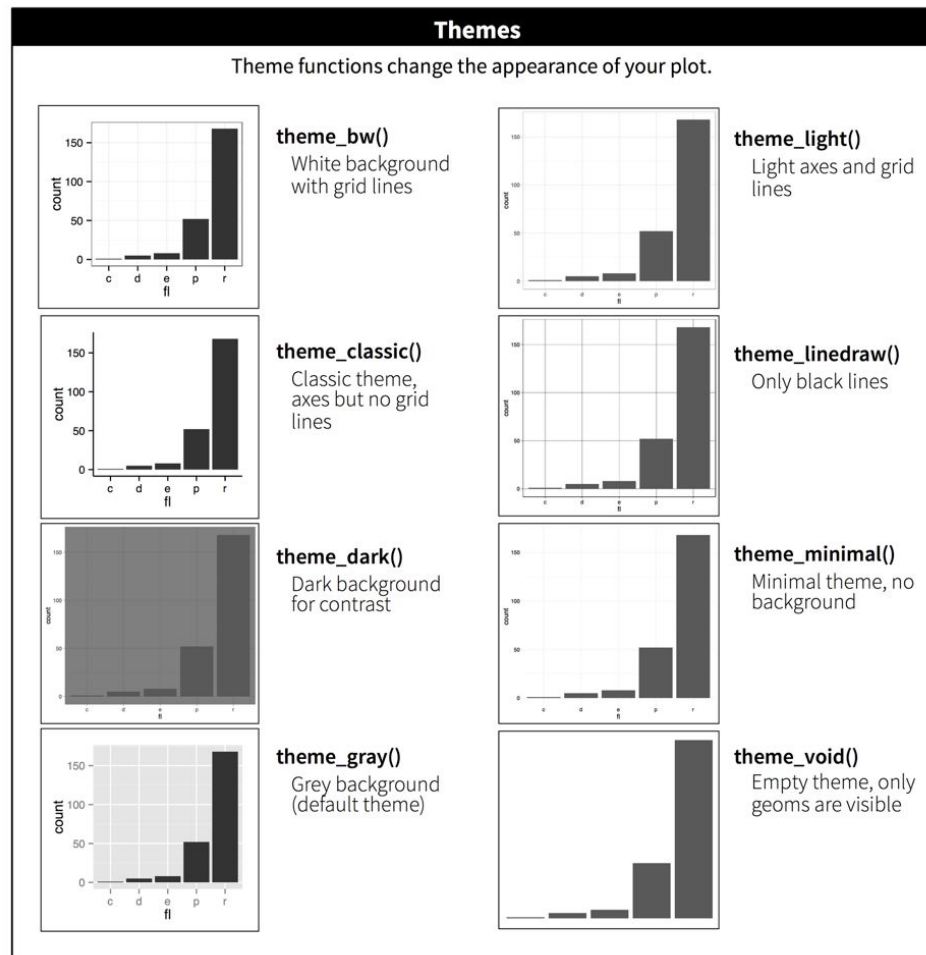
1. Data
 - a. What data are you plotting?
2. Aesthetics (`aes`)
 - a. x-axis & y-axis
 - b. shape
 - c. color
 - d. size
 - e. fill
3. Geoms (how the data will be plotted)
 - a. Scatter plot `geom_point()`
 - b. Boxplot `geom_boxplot()`
 - c. Barchart `geom_bar()`
 - d. Heatmap `geom_tile()`

Unfortunately I can't go through every plot.....

But I can show the iterative process for improving one single plot, which can be applied to all other ggplots!

We will take 1 plot and improve it over 7 iterations

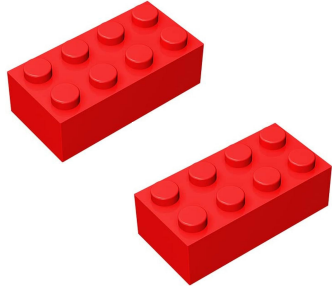
Themes



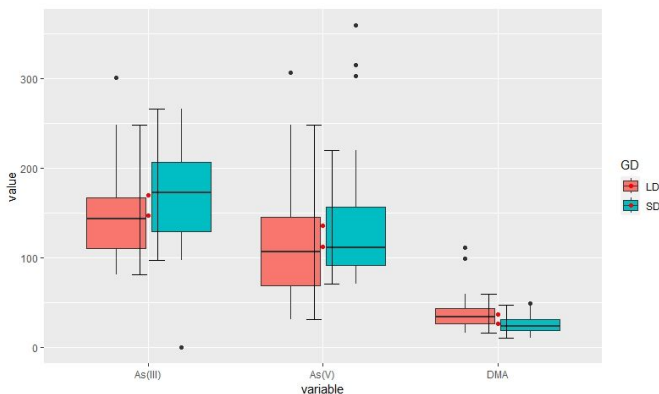


PROBLEM 2

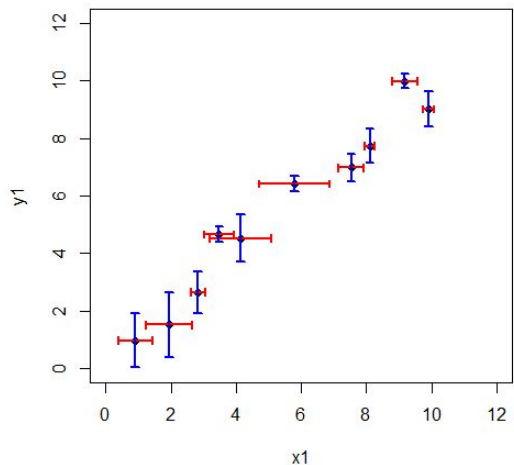
Section C: More Plots



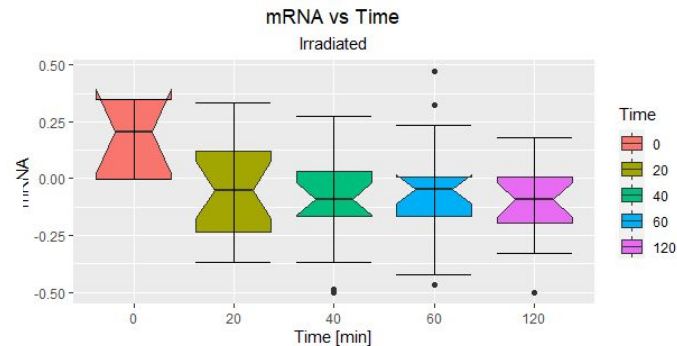
Boxplots and Error Bars (examples)



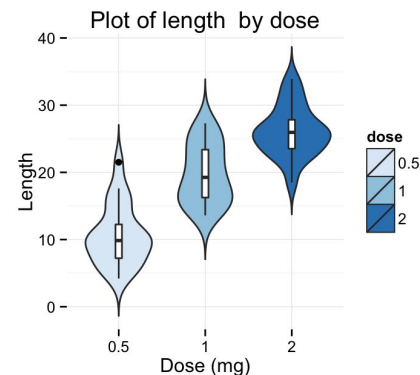
<https://forum.posit.co/t/adding-error-bar-and-mean-value-in-box-plot-with-multiple-variables/174908>



<https://chitchat.wordpress.com/2013/06/25/add-error-bars-to-a-plot-in-r/>



<https://stackoverflow.com/questions/63860118/why-doesnt-ggplot-show-the-error-bar-of-a-boxplot>



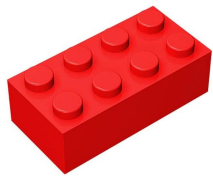
https://www.sthda.com/english/wiki/ggplot2-violin-plot-quick-start-guide-r-software-and-data-visualization#google_vignette

Demo in RStudio



PROBLEM 3

Section D: Faceting



Faceting

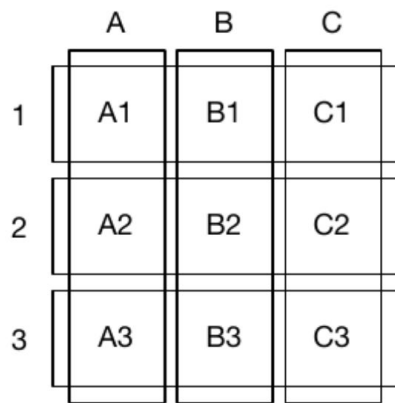
Concept: Plot subplots by the group(s) of your choosing

Why it's so great:

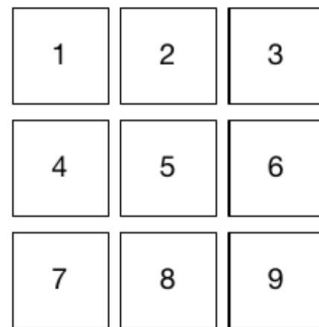
- Separate panels for each group
- Compare patterns side-by-side
- Professional, publication-quality

There are three types of faceting:

- `facet_null()`: a single plot, the default.
- `facet_wrap()`: “wraps” a 1d ribbon of panels into 2d.
- `facet_grid()`: produces a 2d grid of panels defined by variables which form the rows and columns.



facet_grid



facet_wrap



PROBLEM 4



Lily's 2 cents

1. Learn to use .Rdata - super helpful and can save time
2. Never use a pie chart
3. Marker Understanding: Color > Size > Shape
4. Good figures + graphics can take as much time as the analysis
5. Take time to explain

