

# CSE P 590

## Building Data Analysis Pipelines

Fall 2024



Data visualization and reporting



# Today

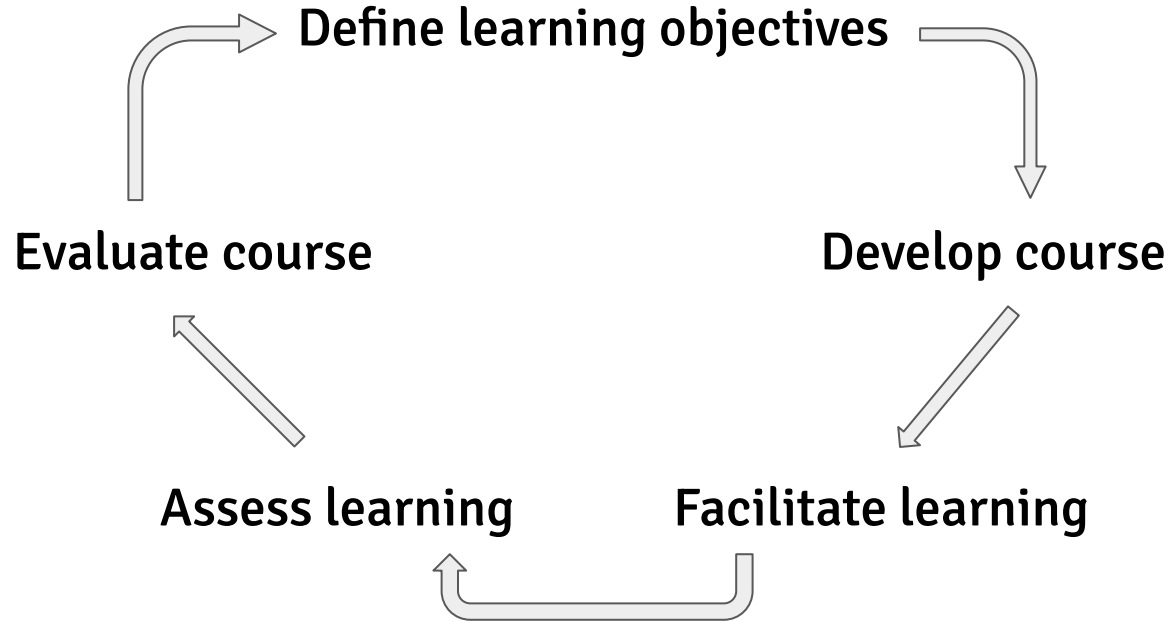
- Logistics and reflection
- Effective tables and visualizations
  - Tables vs. graphs
  - Effective tables
  - Effective visualizations (ggplot2)
- HW2: Overview
- Tutorial: Quarto

# Logistics and reflection

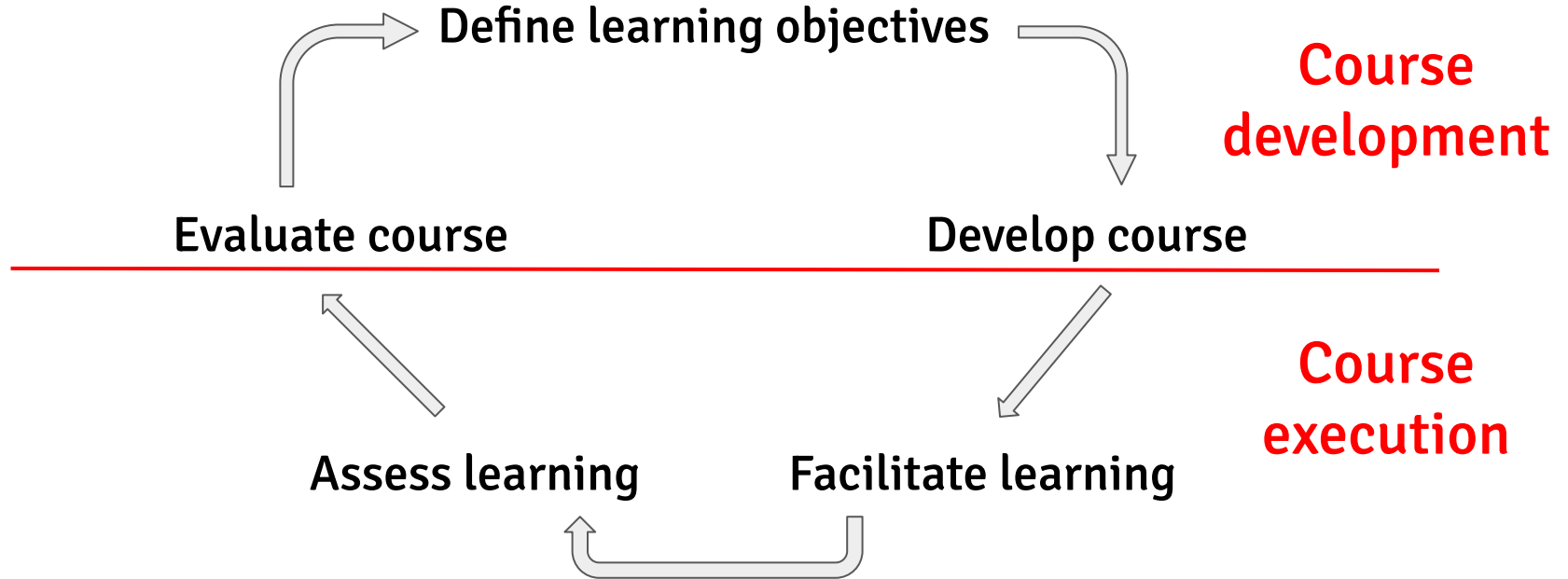
# Reflections on HW1



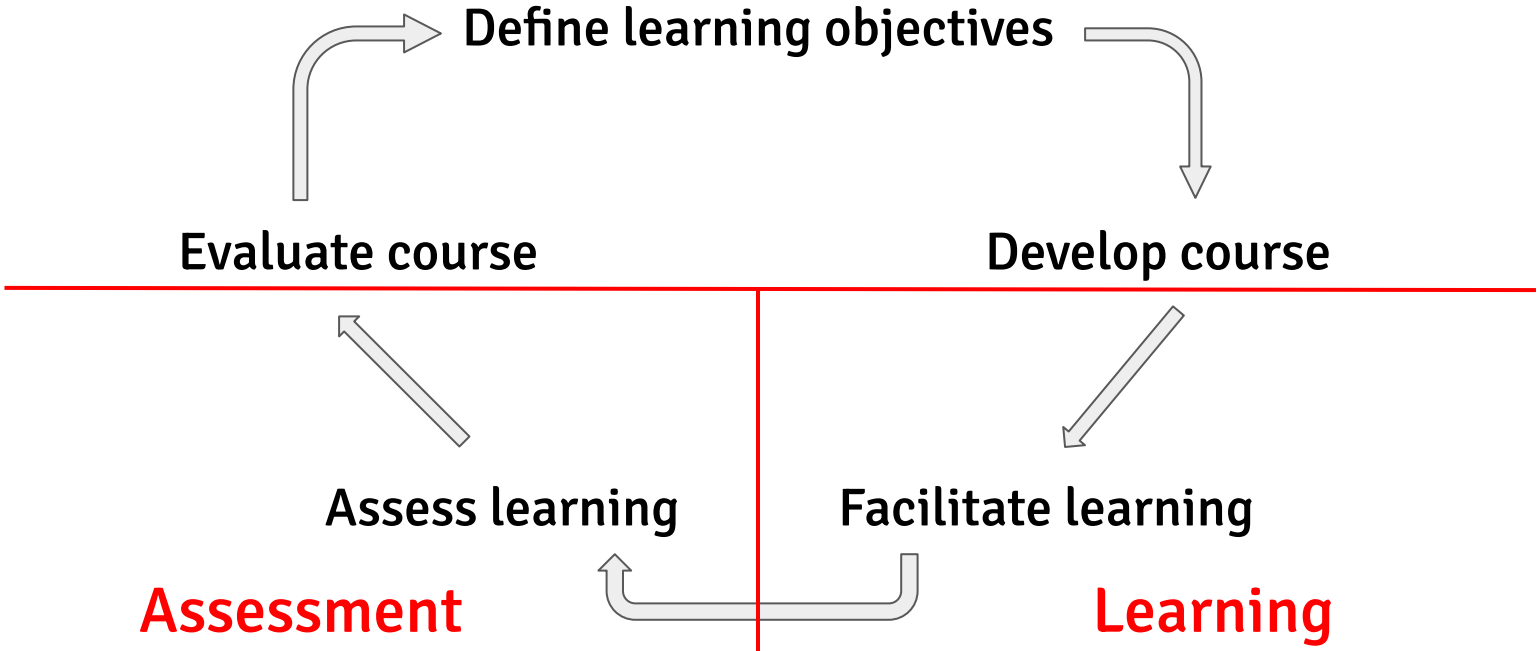
# HW1 in the teaching, learning, assessment cycle



# HW1 in the teaching, learning, assessment cycle



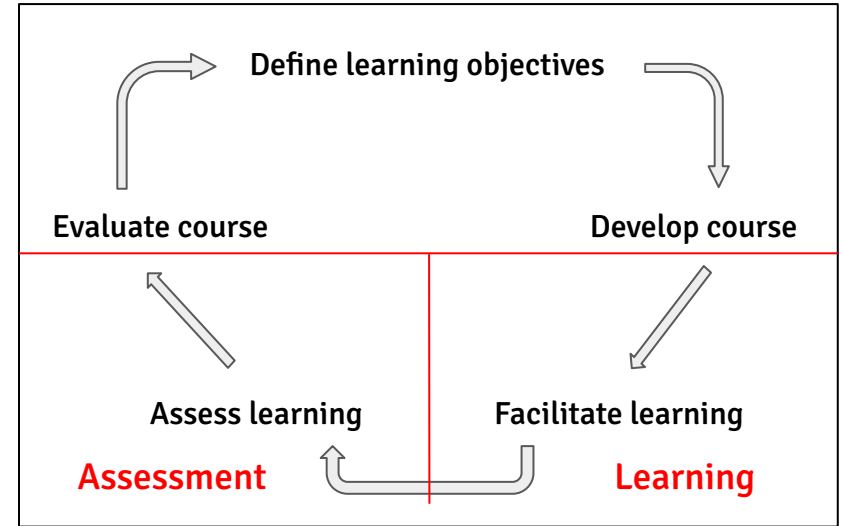
# HW1 in the teaching, learning, assessment cycle



# HW1 in the teaching, learning, assessment cycle

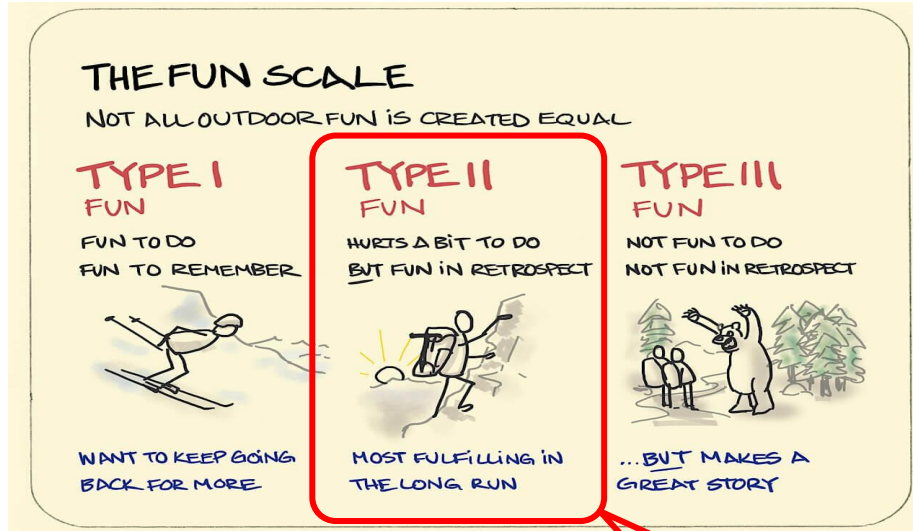
## HW 1

- Designed to facilitate learning
- Two key focus areas:
  - Analysis design and validity
  - Reasoning under uncertainty
- Primer for HW2
- Address HW1 grading feedback (and get HW1 points back)





# HW1 in one picture: mostly type II fun



Sweet spot for teaching

# Course overview: the big picture

- 09/30: Course introduction
- 10/07: Analysis design and validity
- 10/14: Data wrangling
- 10/21: Statistical modeling
- 10/28: Statistical significance and power
- 11/04: Advanced statistical modeling
- 11/11: *No class*
- **11/18: Data visualization and reporting**
- **11/25: Big data**
- **12/02: Big data**



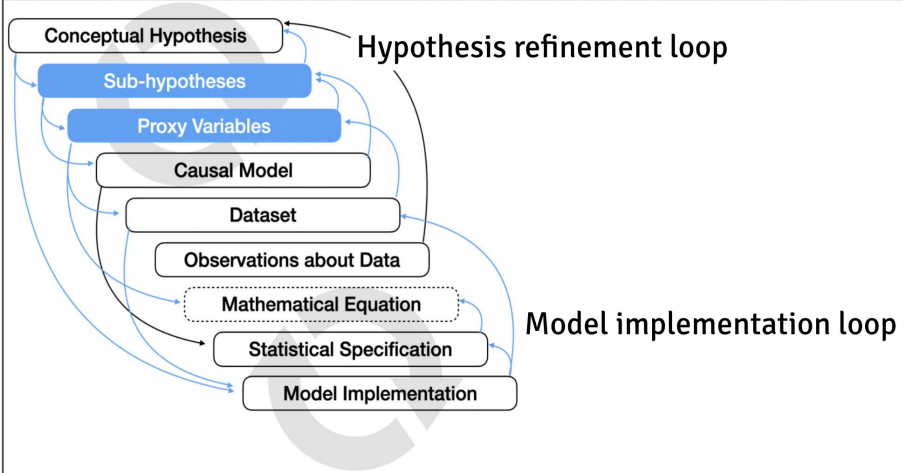
# Course overview: the big picture

- 09/30: Course introduction
- 10/07: Analysis design and validity In-class exercise
- 10/14: Data wrangling In-class exercise
- 10/21: Statistical modeling In-class exercise
- 10/28: Statistical significance and power In-class exercise
- 11/04: Advanced statistical modeling HW 1
- 11/11: *No class*
- 11/18: Data visualization and reporting HW 2
- 11/25: Big data In-class exercise
- 12/02: Big data

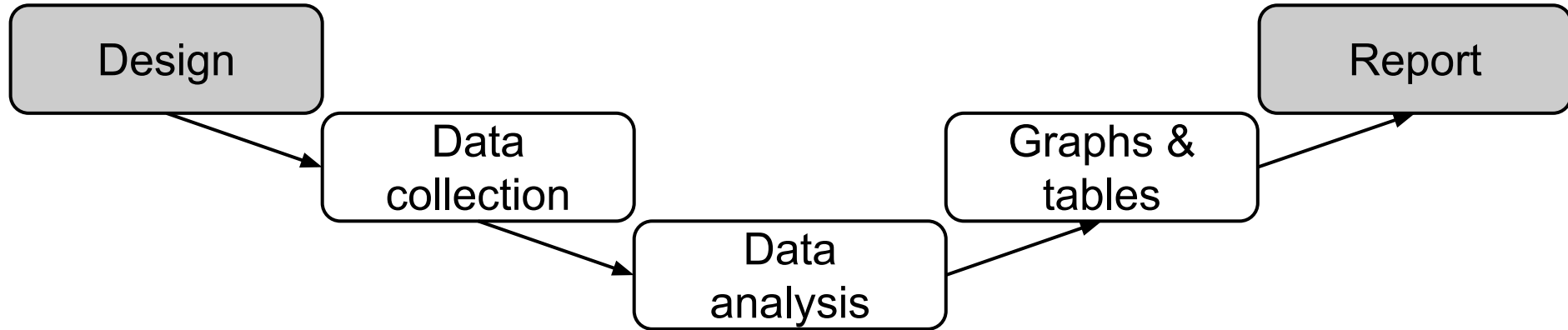
**Extended due date for HW2 (12/04) and more time for in-class 5!**

# Tables vs. graphs

# From analysis design to report

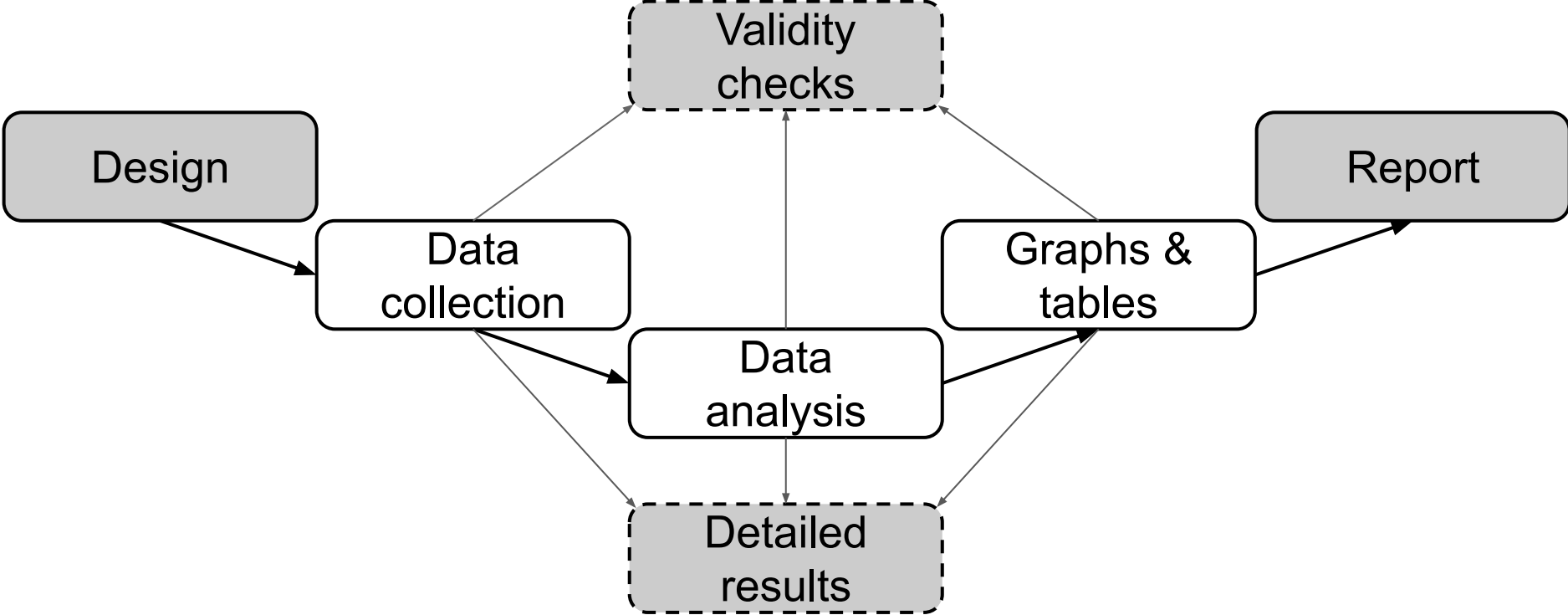


# From analysis design to report



**Do all analysis results go into the final report?**

# From analysis design to report



# Tables vs. graphs

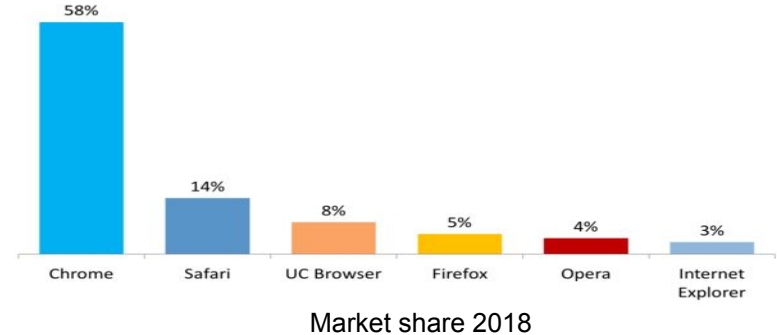
- When are tables useful?

- Compare individual values
- Values involve multiple units
- Precise values are important

	Browser	Market share (%)	
		June 08	July 09
All users	Internet Explorer	75.4	67.7
	Firefox	18.9	22.5
	Safari	2.8	4.1
	Chrome	—	2.6
	Opera	2.1	2.0
	Netscape	0.5	0.7
	Other	0.2	0.5

- When are graphs useful?

- Consider an entire set of values
- Visualize trends and patterns
- Relationships are more important than precise values





# Effective tables

# Effective tables: the run-time data set

```
variant, naive, caching, forking, run, subject
11,      309.8, 157.6, 144.8, 1, "tax"
12,      379.5, 237.4, 254.5, 1, "tax"
13,      415.9, 225.9, 225.9, 1, "tax"
...
```

- Recall the run-time data set
  - 3 subjects (tax, tictactoe, triangle)
  - 3 strategies (naive, caching, forking)
  - 5 runs to account for the variation in run time

**Goal: show run times and relative improvements in a table**

# Effective tables: layout

TABLE I  
RUN TIMES AND IMPROVEMENTS.

Subject	RT-naive	RT-cache	RT-fork	I-cache	I-fork
tax	504.11	247.01	195.42	51.02%	61.31%
tictactoe	17.44	16.32	15.43	6.31%	11.49%
triangle	3.13	2.79	1.67	10.91%	46.62%

- Recall the run-time data set
  - 3 subjects (tax, tictactoe, triangle)
  - 3 strategies (naive, caching, forking)
  - 5 runs to account for the variation in run time

What are the pros/cons of Table I?

How would you improve it?

# Effective tables: layout

TABLE I  
RUN TIMES AND IMPROVEMENTS.

Subject	RT-naive	RT-cache	RT-fork	I-cache	I-fork
tax	504.11	247.01	195.42	51.02%	61.31%
tictactoe	17.44	16.32	15.43	6.31%	11.49%
triangle	3.13	2.79	1.67	10.91%	46.62%

Compare the two  
w.r.t. readability,  
clarity, and  
interpretability

TABLE II  
RUN TIMES AND IMPROVEMENTS FOR THE NAIVE, CACHING (CACHE),  
AND FORKING (FORK) STRATEGIES. RUN TIMES ARE GIVEN IN SECONDS  
AND AVERAGED OVER FIVE RUNS.

Subject	Run times			Improvements	
	naive	cache	fork	cache (vs. naive)	fork (vs. naive)
Tax	504	247	195	51.0%	61.3%
TicTacToe	17.4	16.3	15.4	6.31%	11.5%
Triangle	3.13	2.79	1.67	10.9%	46.6%

# Effective tables: content

## Keep it simple

- Avoid mixing higher-is-better and lower-is-better numbers
- Allow for easy comparisons, primarily by row
- Be consistent about precision vs. significant digits
- Summarize the table (what is the bottom line?)

TABLE II  
RUN TIMES AND IMPROVEMENTS FOR THE NAIVE, CACHING (CACHE),  
AND FORKING (FORK) STRATEGIES. RUN TIMES ARE GIVEN IN SECONDS  
AND AVERAGED OVER FIVE RUNS.

Subject	Run times			Improvements	
	naive	cache	fork	cache (vs. naive)	fork (vs. naive)
Tax	504	247	195	51.0%	61.3%
TicTacToe	17.4	16.3	15.4	6.31%	11.5%
Triangle	3.13	2.79	1.67	10.9%	46.6%

# Effective tables: summaries



**Total**

**vs.**

**Average**

Subject	LOC	Speed up
Tax	8900	10.2%
TicTacToe	120	54.2%
Triangle	80	60.9%
Average	3393	41.8%

What are the downsides of these summaries?

# Effective tables: best practices

## Do

- Make each table self-contained (content and caption)
- Use descriptive (hierarchical) headers
- Right align numbers
- Use meaningful totals or weighted averages
- Be consistent about precision vs. significant digits

## Don't

- Don't use horizontal lines between related rows
- Don't use vertical lines between related columns

# Effective graphs

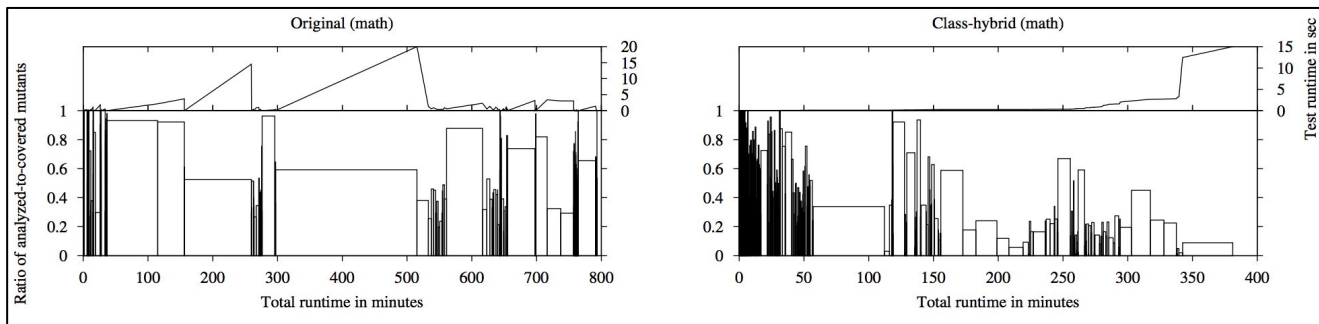
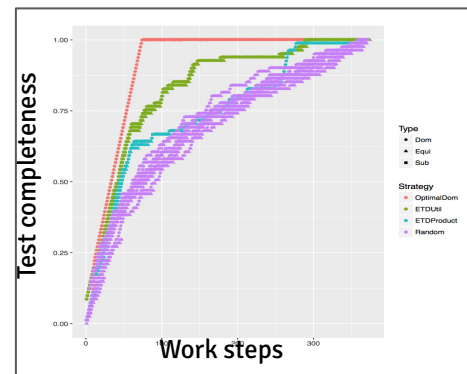
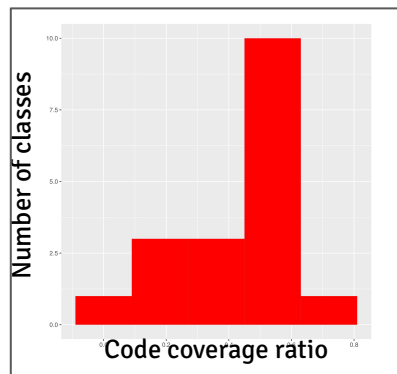
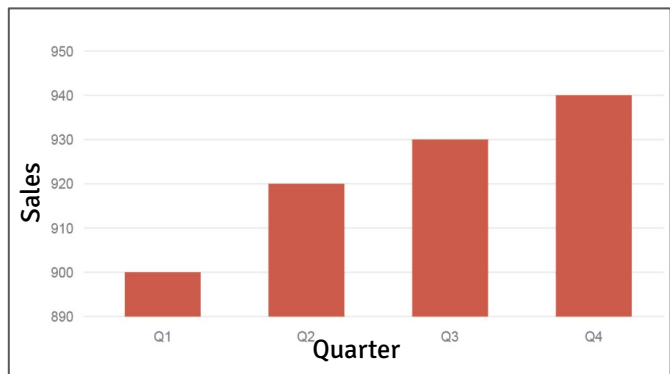


# 4 beautiful graphs

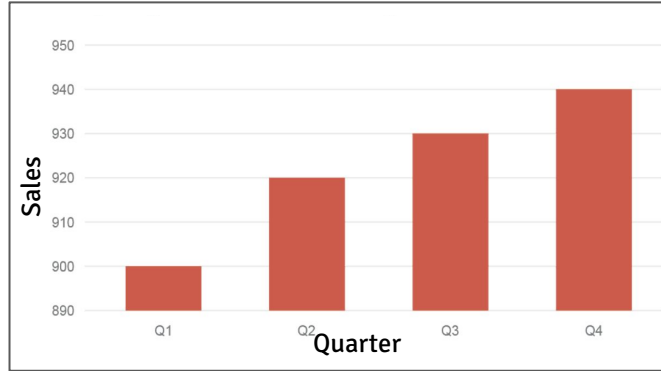
- Small groups of 4-6 students
- 4 example graphs
- For each graph
  - Discuss pros and cons
  - Propose improvements



# 4 beautiful graphs

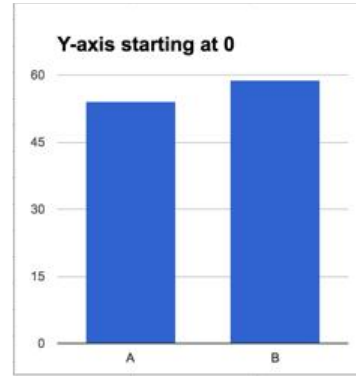
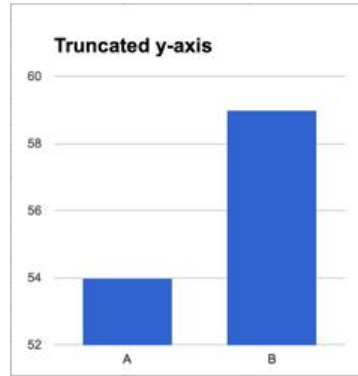


# Example 1: bar charts



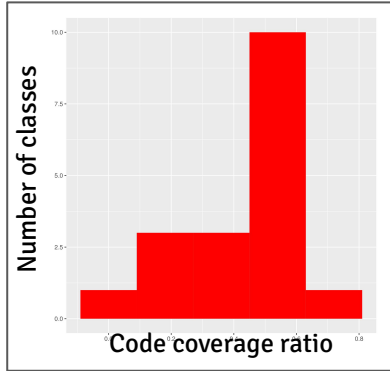
**Truncated axes are misleading and not a proper way to “demonstrate” effect size!**

# Example 1: bar charts



**Truncated axes are misleading and not a proper way to “demonstrate” effect size!**

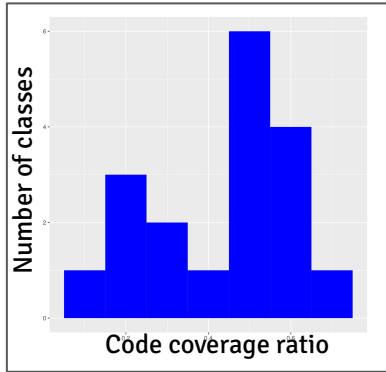
# Example 2: histogram



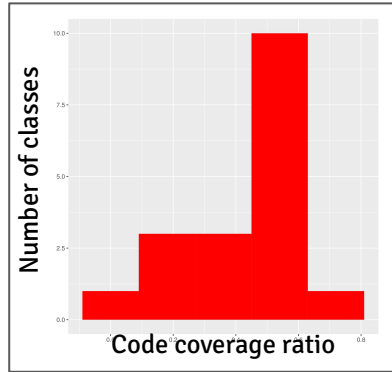
Good visual summary of count data, but binning may be misleading.

Kernel density overlay can provide information about adequate binning.

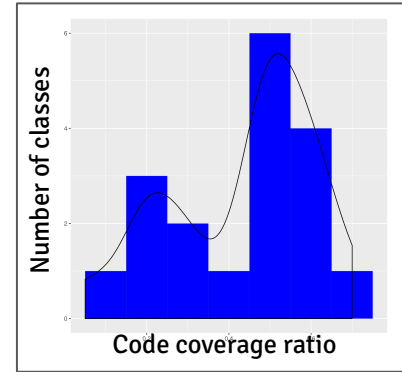
# Example 2: histogram vs. density plot



Adequate binning



Changed binning

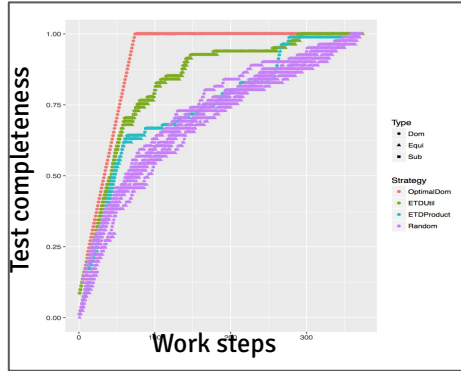


Kernel density overlay

Good visual summary of count data, but binning may be misleading.

Kernel density overlay can provide information about adequate binning.

# Example 3: scatter plot

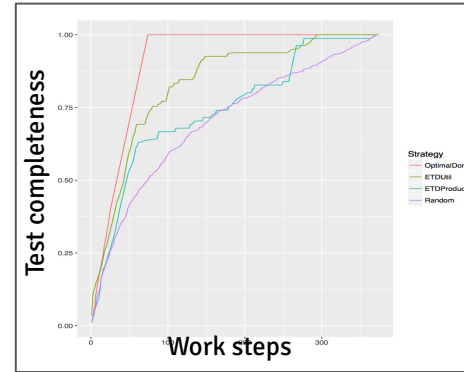
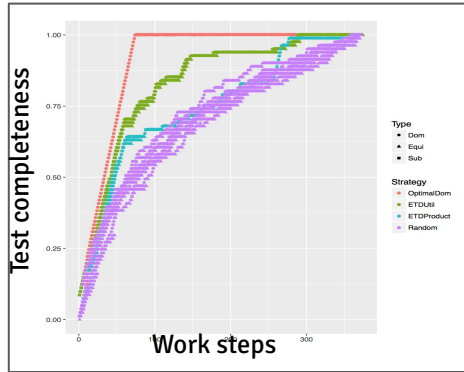


Good visual summary of point clouds, trends, and relationships.

May obscure relevant trends (overlapping points).

Hard to reason about density (without adding transparency).

# Example 3: scatter plot vs. line plot



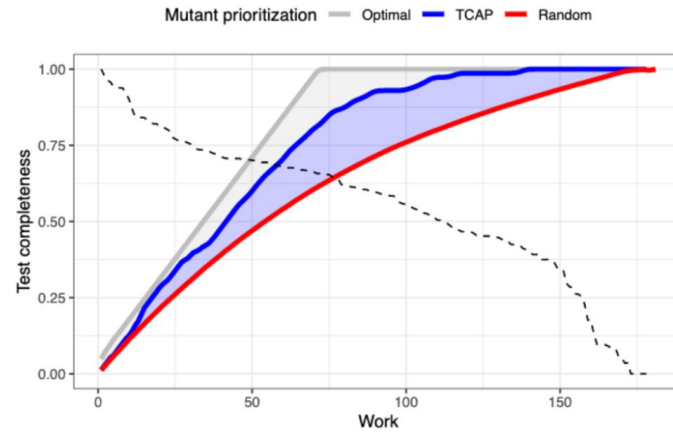
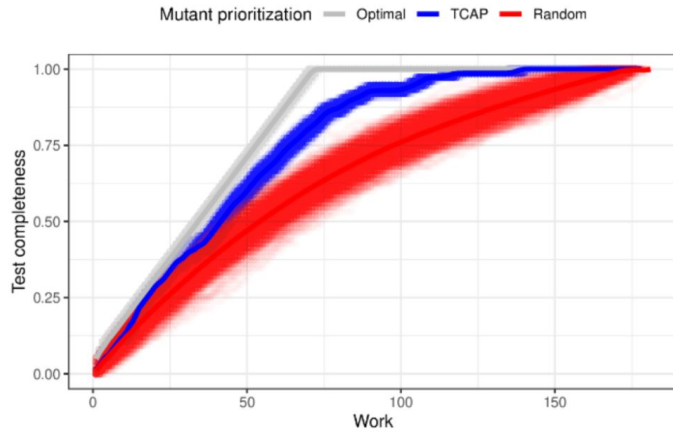
Good visual summary of point clouds, trends, and relationships.

May obscure relevant trends (overlapping points).

Hard to reason about density (without adding transparency).



# Example 3: scatter plot vs. line plot

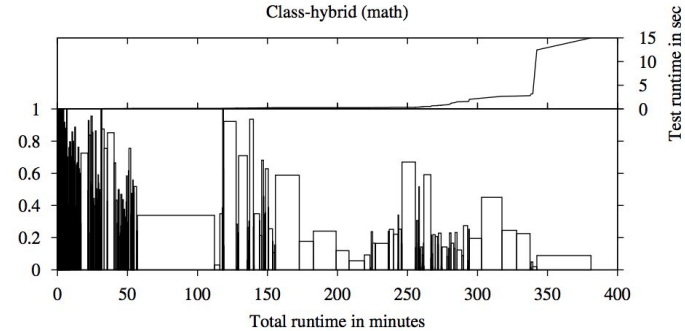
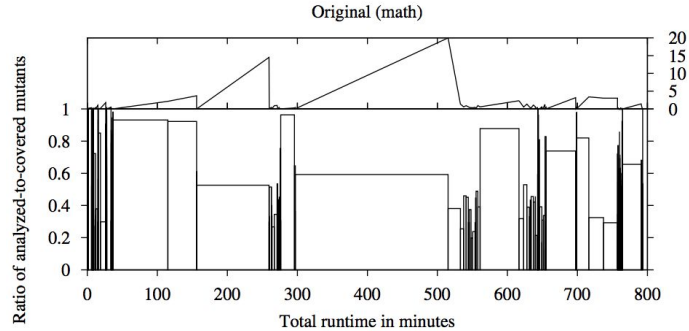


Good visual summary of point clouds, trends, and relationships.

May obscure relevant trends (overlapping points).

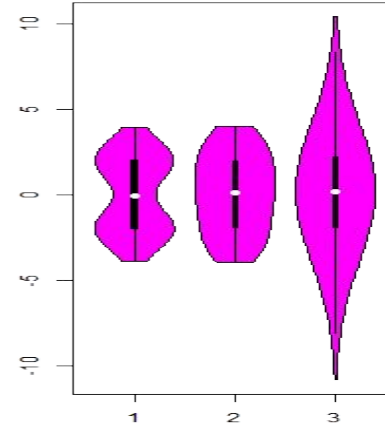
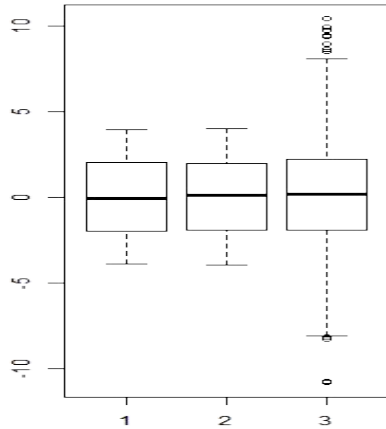
Hard to reason about density (without adding transparency).

# Example 4: multi-plot visualization



**Way too many details!**  
**The key trends and takeaways are obscured.**  
**Good for detailed results but not a final report.**

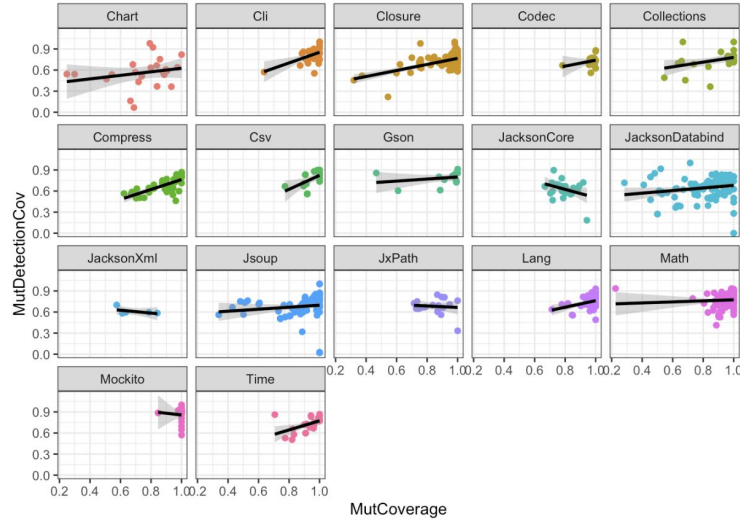
# Effective graphs: box plots vs. violin plots



## Box plots:

- Good visual data summary
- Nicely complements hypothesis tests
- May be misleading for multimodal data
- May be misleading for small samples

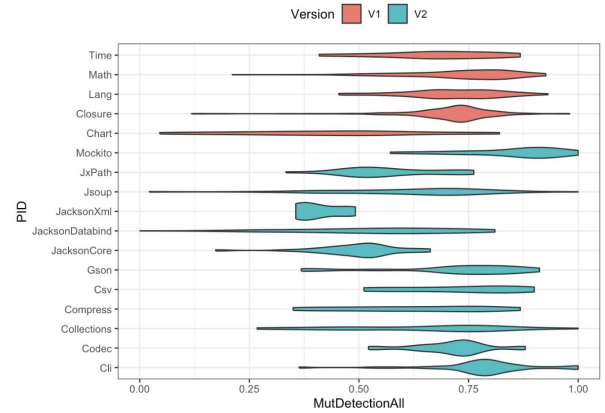
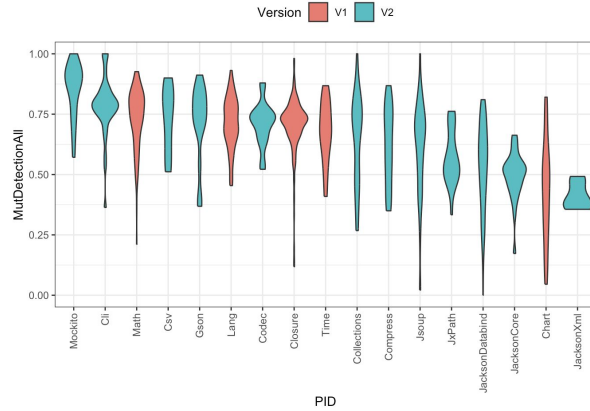
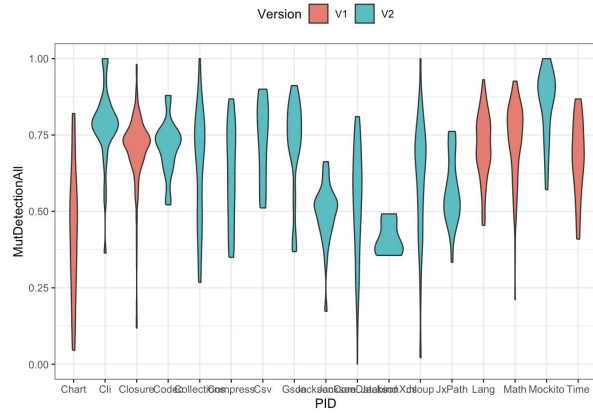
# Effective graphs: facet plots



## Facet plots:

- Clean visualizations across multiple dimensions of interest
- Allows for comparisons within groups and across groups
- Complementary to other ggplot2 aesthetics (color, shape, etc.)
- Use ggplot's `facet_grid` for cross-product visualizations (formula syntax)

# Effective graphs: reorder and/or flip axes



## Reorder and/or flip axes:

- Reorder by mean/median or by groups of interest etc.
- Flip axes for readability if appropriate
- Favor short labels over rotated labels

# Effective graphs: best practices

## Do

- **Use ggplot2!**
- Make each plot self-contained (content and caption)
- Relate tables and graphs to tell a consistent story
- By default put the DV on the vertical axis
- Reduce complexity with facet plots

## Don't

- Don't use multiple, unrelated axes
- Don't connect unrelated data points  
(choose an appropriate graph instead)

**A real-world example**

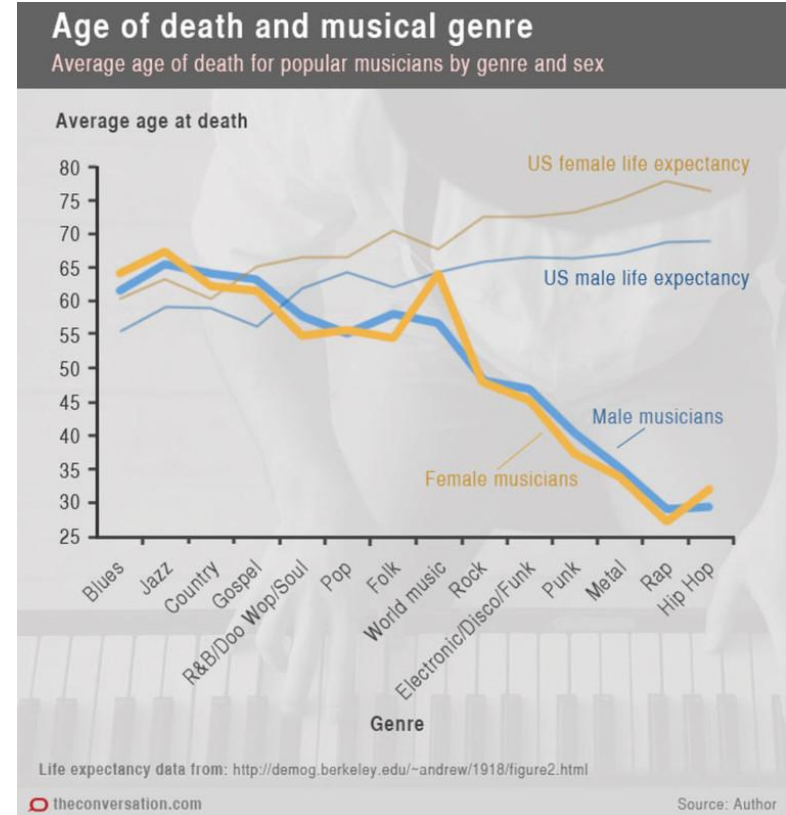
# Let's consider the following reporting

Cause of death by genre					
Various causes of death for musicians of different genres					
	Accidental	Suicide	Homicide	Heart-related	Cancer
% deaths per cause	19.5%	6.8%	6.0%	17.4%	23.4%
Blues	9.2%	2.0%	3.5%	28.0%	24.2%
Jazz	10.6%	2.7%	1.9%	20.7%	30.6%
Country	15.8%	4.7%	1.6%	23.5%	25.1%
Gospel	13.3%	0.9%	3.6%	18.5%	23.0%
R&B	11.5%	1.6%	5.0%	23.2%	26.8%
Pop	19.0%	6.4%	2.9%	16.4%	26.7%
Folk	15.9%	5.5%	4.4%	15.3%	32.3%
World music	12.7%	3.4%	9.6%	17.8%	19.9%
Rock	24.4%	7.2%	3.6%	15.4%	24.7%
Electronic	16.7%	5.0%	10.0%	15.0%	25.0%
Punk	30.0%	11.0%	8.2%	12.6%	18.3%
Metal	36.2%	19.3%	5.9%	11.0%	14.1%
Rap	15.9%	6.2%	51.0%	6.9%	7.6%
Hip Hop	18.3%	7.4%	51.5%	6.1%	6.1%

Red: significantly above the overall average rate for cause of death  
Blue: above the overall average rate for cause of death  
Green: significantly below the overall average rate for cause of death

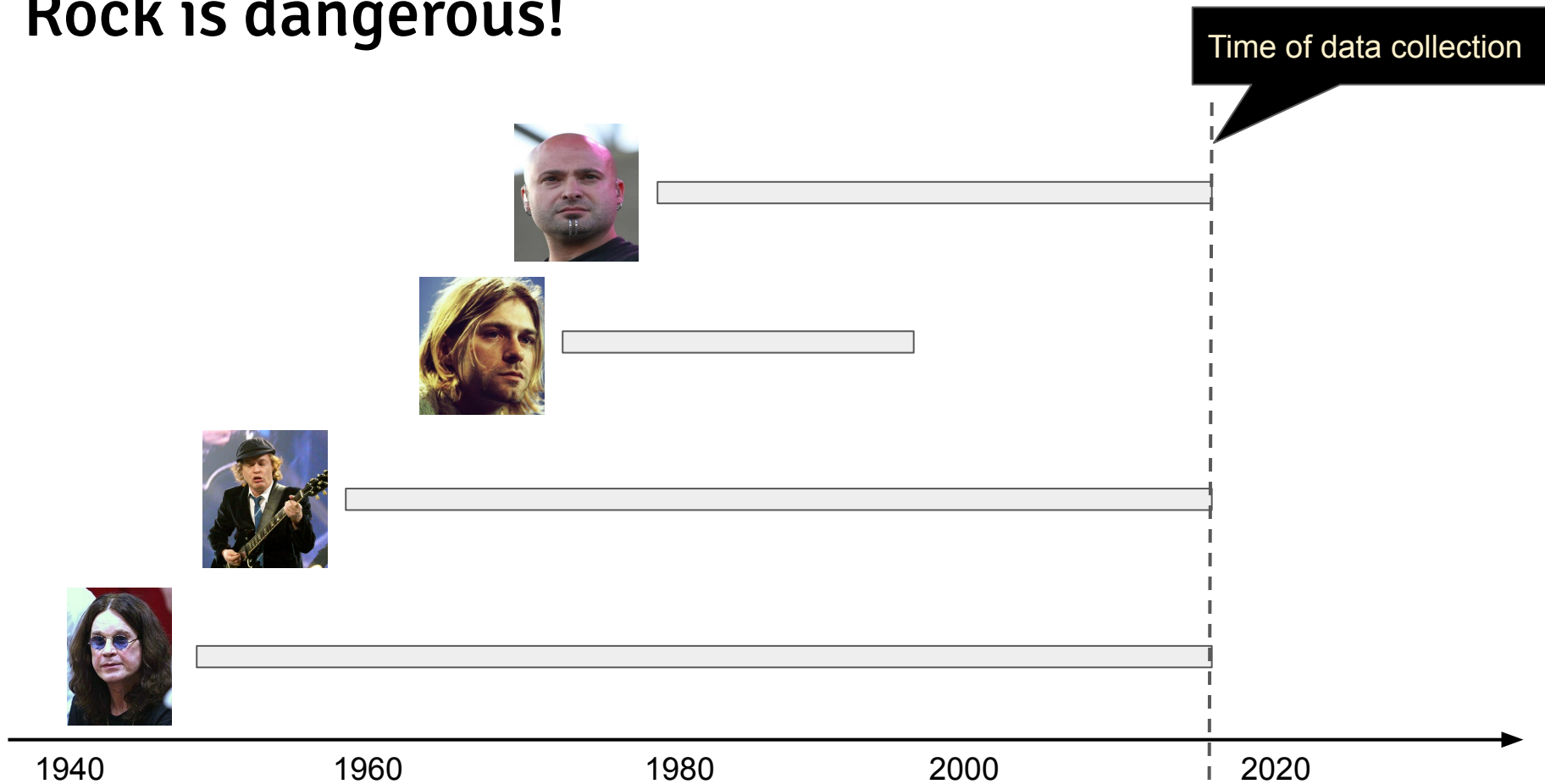
Note: not all causes shown

Source: Author

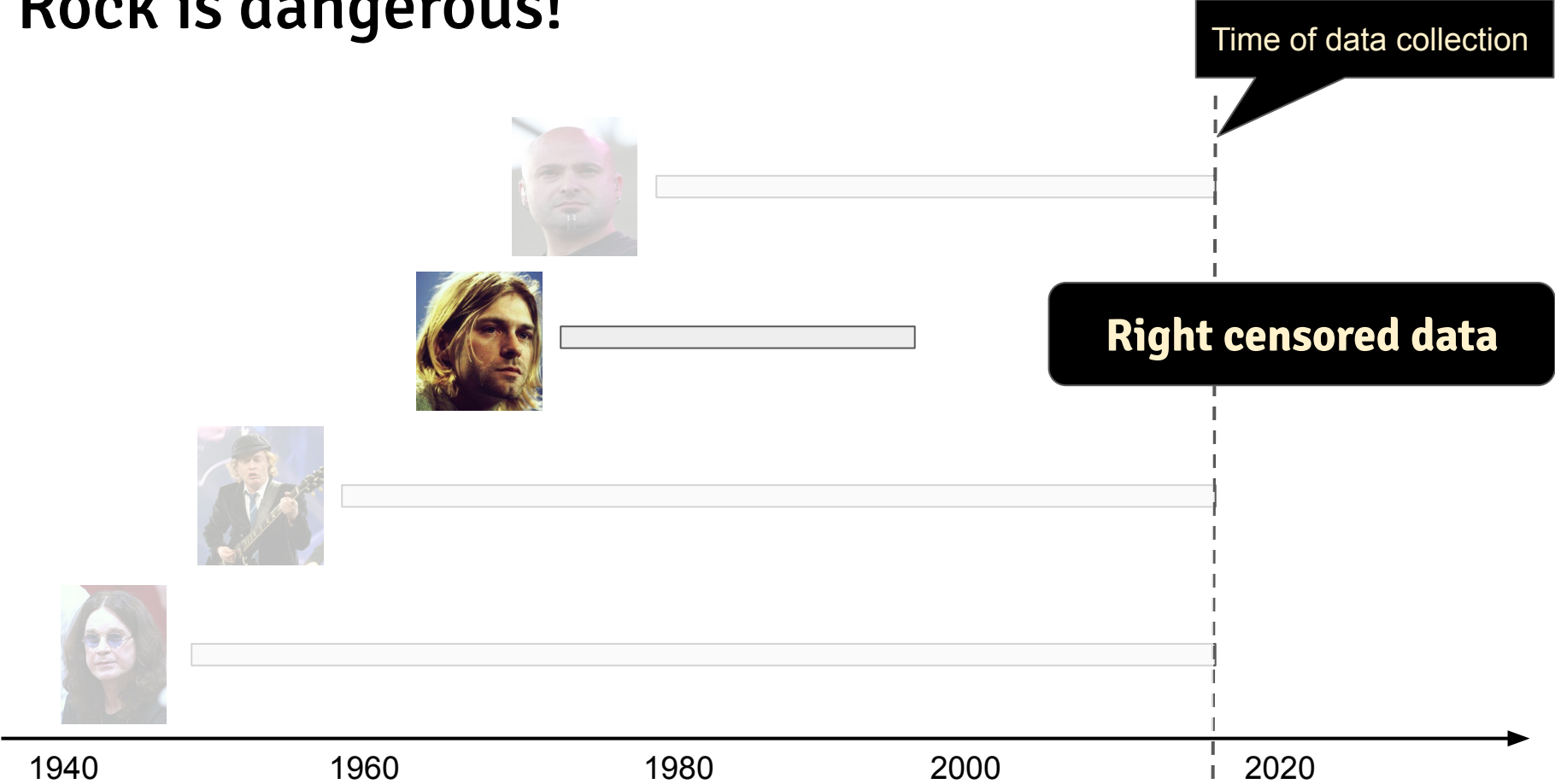




# Rock is dangerous!

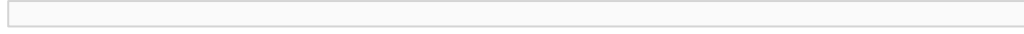


# Rock is dangerous!



# Rock is dangerous!

Use survival analysis or (if appropriate) assume the event happens immediately after data collection (e.g., overestimate baseline performance)



Time of data collection

**Right censored data**

1940

1960

1980

2000

2020

# HW2: Overview

# HW2: Revisit and extend your HW1 solution

## 4 Parts

1. Produce two Quarto reports
  - a. Detailed analysis report
  - b. Summary report or presentation
2. Use different visualizations
3. Address grading feedback from HW1
4. Use distributed computing with Spark(lyr)
  - a. Distributed data consolidation
  - b. Distributed computation

## Today

- Render your HW1 notebook with Quarto

# Tutorial: Quarto