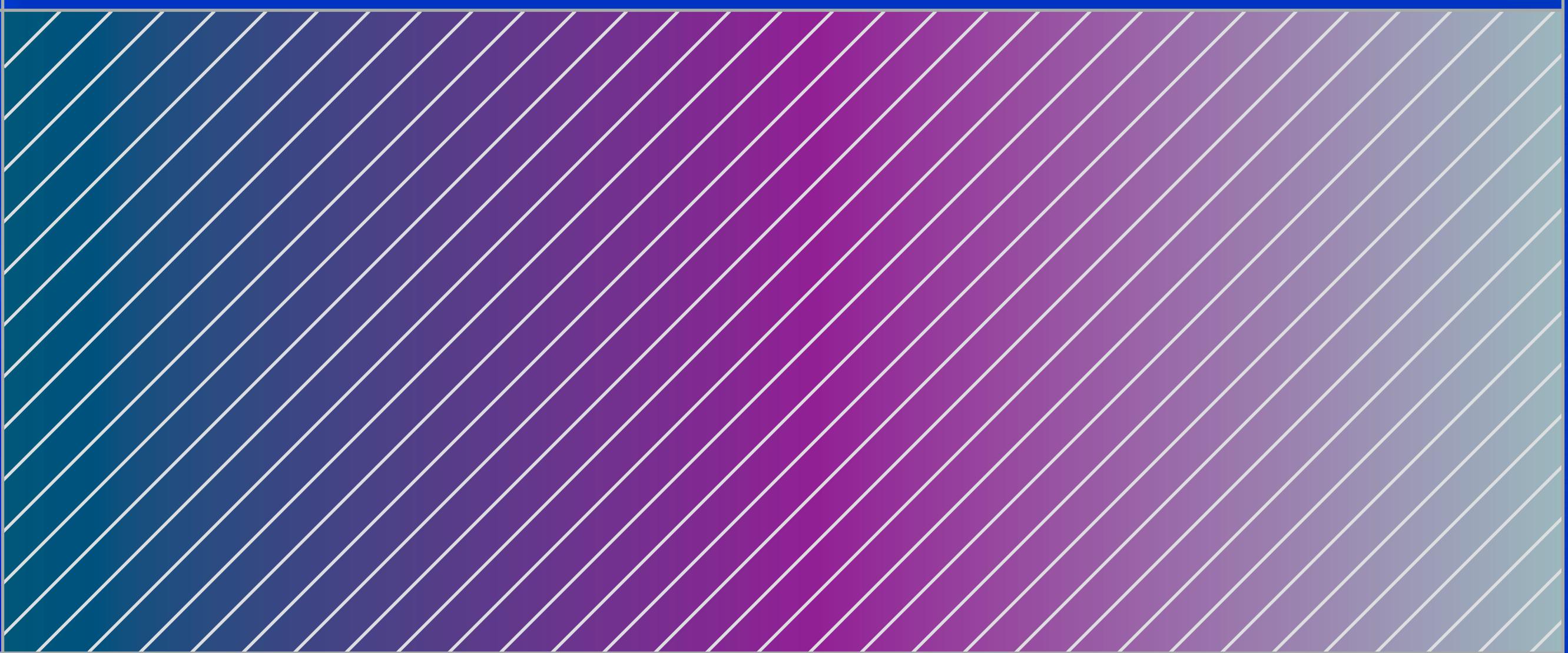


Adequate Tables? No, We Want Great Tables



 rich-iannone
 rich@posit.co

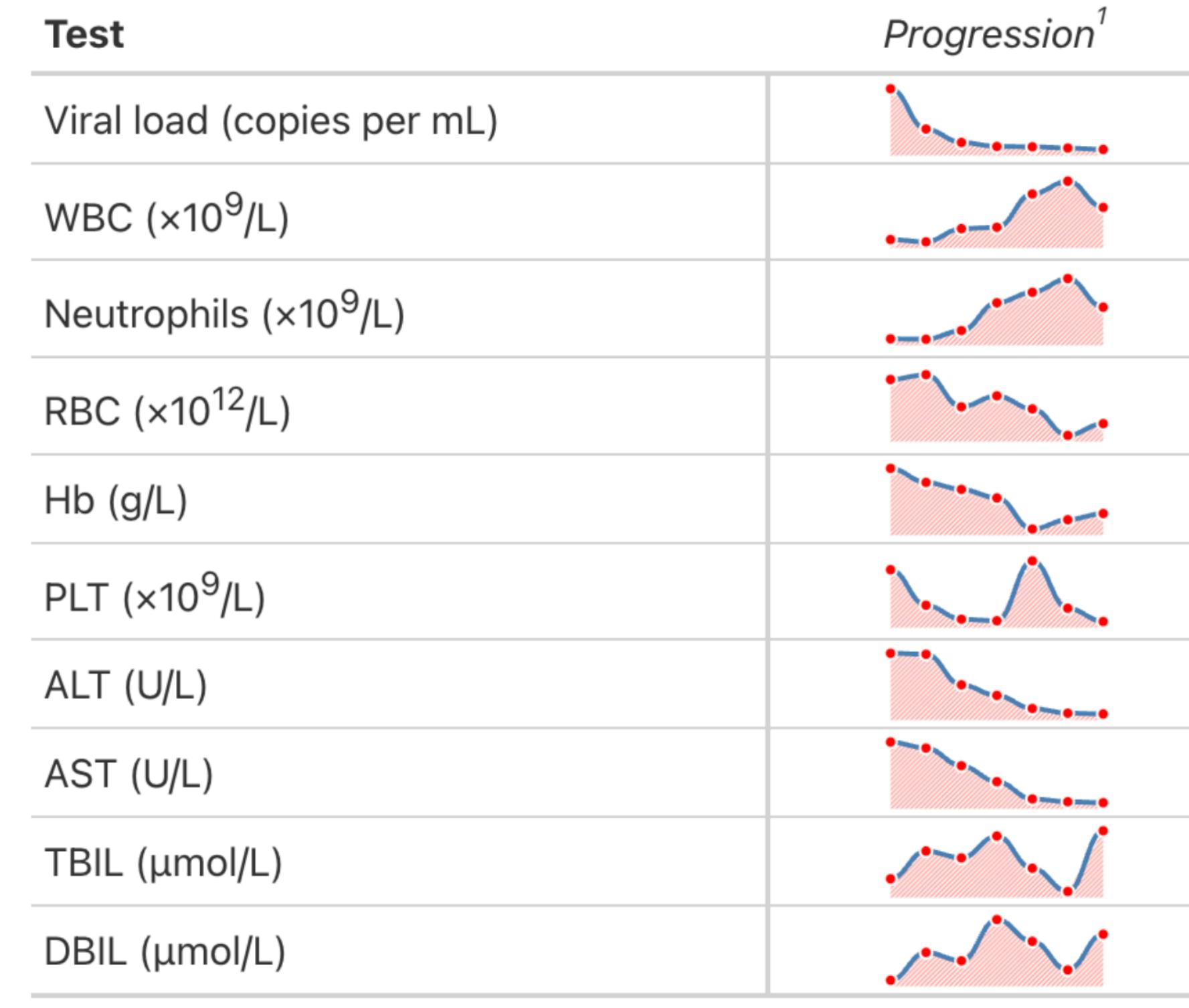
Tables Are Great for Displaying Information

Tables are underrated. They can be awesome. Here's a few I made:

Gas-phase reactions of selected mercaptan compounds

	CH ₄ S	Reaction Rate Constant (298 K), cm ³ molecules ⁻¹ s ⁻¹		
		OH	Cl	NO ₃
methanethiol	CH ₄ S	3.50 × 10 ⁻¹¹	2.00 × 10 ⁻¹⁰	9.20 × 10 ⁻¹³
ethanethiol	C ₂ H ₆ S	4.50 × 10 ⁻¹¹	1.75 × 10 ⁻¹⁰	1.21 × 10 ⁻¹²
propanethiol	C ₃ H ₈ S	5.30 × 10 ⁻¹¹	2.14 × 10 ⁻¹⁰	—
2-propanethiol	C ₃ H ₈ S	3.90 × 10 ⁻¹¹	2.70 × 10 ⁻¹⁰	—
1-butanethiol	C ₄ H ₁₀ S	5.60 × 10 ⁻¹¹	—	—
2-methyl-1-propanethiol	C ₄ H ₁₀ S	4.60 × 10 ⁻¹¹	—	—
2-butanethiol	C ₄ H ₁₀ S	3.80 × 10 ⁻¹¹	1.65 × 10 ⁻¹⁰	—
t-butylsulfide	C ₄ H ₁₀ S	2.90 × 10 ⁻¹¹	—	—
2-methylbutanethiol	C ₅ H ₁₂ S	5.20 × 10 ⁻¹¹	—	—
n-pantanethiol	C ₅ H ₁₂ S	—	1.97 × 10 ⁻¹⁰	—
1,2-ethanedithiol	C ₂ H ₆ S ₂	3.80 × 10 ⁻¹¹	—	—

Partial summary of daily tests performed on YF patient



¹ Measurements from Day 3 through to Day 8.

Tables Can Also Look Really, Really Good.

Here's some *really nice* tables made by someone else:

Performance against top 100 teams over the past five seasons				
Average T-Rank game score vs. then-top 100 opponents from 2020-2024; min. total 25 games				
1.  89.6 (85)	21.  80.0 (100)	41.  74.4 (100)	61.  71.1 (83)	81.  65.1 (42)
2.  88.0 (118)	22.  80.0 (112)	42.  74.4 (107)	62.  70.9 (92)	82.  65.0 (64)
3.  87.4 (103)	23.  79.0 (89)	43.  74.4 (97)	63.  70.1 (102)	83.  65.0 (72)
4.  86.7 (111)	24.  79.0 (135)	44.  74.3 (110)	64.  69.5 (49)	84.  64.9 (61)
5.  85.8 (132)	25.  78.8 (119)	45.  74.3 (80)	65.  68.9 (59)	85.  64.8 (95)
6.  85.5 (129)	26.  78.7 (70)	46.  74.2 (93)	66.  68.6 (71)	86.  64.5 (98)
7.  84.4 (99)	27.  78.4 (116)	47.  73.2 (106)	67.  68.4 (99)	87.  64.5 (67)
8.  83.6 (130)	28.  78.2 (91)	48.  73.0 (82)	68.  68.4 (108)	88.  64.0 (110)
9.  83.1 (107)	29.  78.1 (103)	49.  73.0 (111)	69.  68.2 (116)	89.  63.1 (72)
10.  82.5 (122)	30.  77.8 (105)	50.  72.8 (117)	70.  67.4 (56)	90.  63.1 (72)
11.  82.4 (120)	31.  77.4 (82)	51.  72.7 (38)	71.  67.3 (94)	91.  62.8 (26)
12.  82.4 (92)	32.  76.6 (118)	52.  72.5 (44)	72.  67.0 (98)	92.  62.8 (114)
13.  82.1 (114)	33.  76.4 (95)	53.  72.4 (116)	73.  66.5 (89)	93.  62.8 (27)
14.  82.0 (106)	34.  76.3 (121)	54.  72.3 (113)	74.  66.2 (81)	94.  62.0 (44)
15.  81.2 (118)	35.  76.1 (113)	55.  72.1 (104)	75.  66.1 (47)	95.  61.9 (111)
16.  81.1 (107)	36.  75.6 (124)	56.  71.7 (105)	76.  66.1 (86)	96.  61.5 (93)
17.  80.8 (111)	37.  75.5 (71)	57.  71.6 (95)	77.  65.9 (74)	97.  60.8 (58)
18.  80.5 (125)	38.  75.3 (118)	58.  71.4 (106)	78.  65.5 (62)	98.  60.6 (112)
19.  80.2 (133)	39.  74.7 (107)	59.  71.3 (126)	79.  65.4 (98)	99.  60.5 (108)
20.  80.1 (120)	40.  74.6 (103)	60.  71.1 (80)	80.  65.3 (90)	100.  60.5 (46)

Opponent rank is determined by Barttorvik T-Rank on game date. Game score is game-level Barthag performance (the probability that you beat an average team on a neutral floor). Total games played is next to mean score.

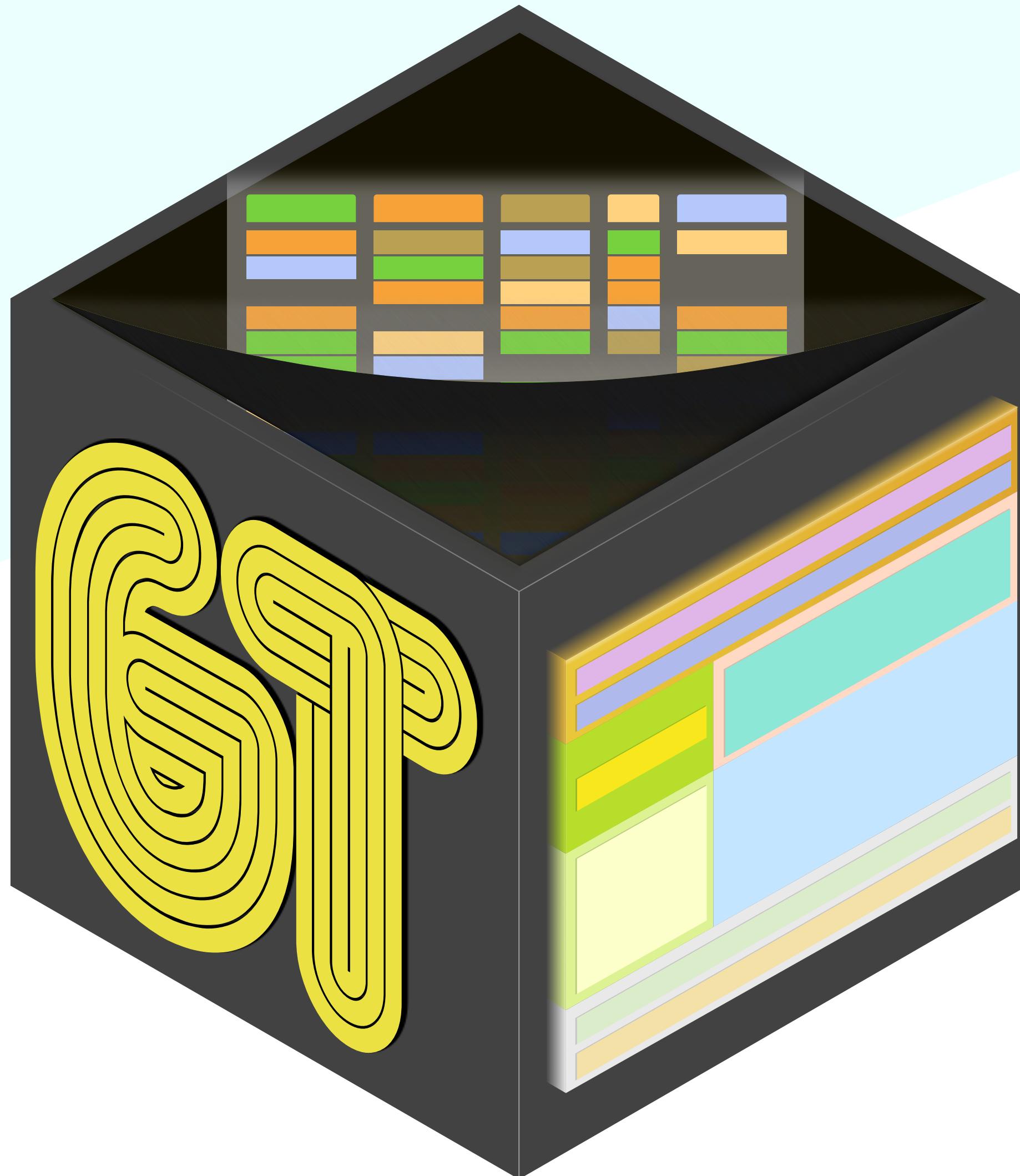
Data by Barttorvik + cbbdata || Viz. + Analysis by @andrewetherman

Who are the best road teams in college basketball?

10 highest T-Rank ratings in true road performances (D-1 vs. D-1)

AVG ¹	TEAM (RECORD)	Road Efficiency			Quadrant Records				
		ADJ. O	ADJ. D	T-RANK ²	Q1	Q2	Q3	Q4	BEST ³
#17	 Saint Mary's (9-0)	119.2	89.9	96.2%	3-0	1-0	2-0	3-0	
#3	 Purdue (7-3)	123.3	96.4	94.4%	3-3	4-0	0-0	0-0	
#2	 Connecticut (9-3)	129.5	101.5	94.2%	7-3	0-0	1-0	1-0	
#9	 North Carolina (8-2)	113.0	88.8	94.1%	5-0	3-2	0-0	0-0	
#8	 Duke (7-4)	123.6	98.4	93.2%	3-2	3-2	1-0	0-0	
#12	 Illinois (6-5)	129.2	103.0	93.1%	3-4	3-1	0-0	0-0	
#10	 Creighton (8-5)	116.6	93.4	92.8%	5-4	1-1	1-0	1-0	
#6	 Arizona (7-4)	123.8	99.4	92.6%	4-1	3-2	0-1	0-0	
#14	 Gonzaga (8-2)	122.6	99.0	92.1%	3-1	0-1	2-0	3-0	
#2	 Houston (7-3)	118.3	95.5	92.1%	7-3	0-0	0-0		

Those Were Made with **gt**. A Package for Making Tables.

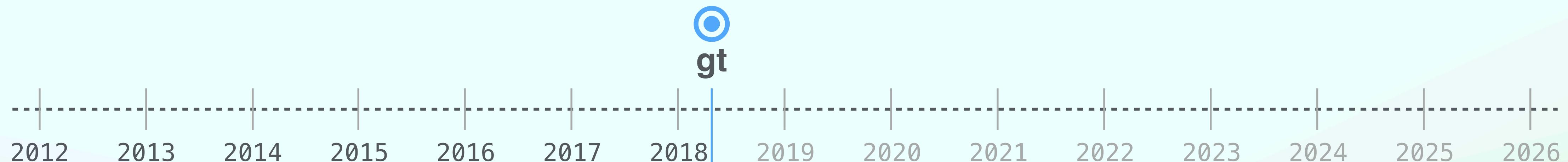


With **gt** you can make useful tables for publication. And you can make them look good!

GOAL OF THE TALK

We're going to look at the story of **gt** and find out where all this table work is headed.

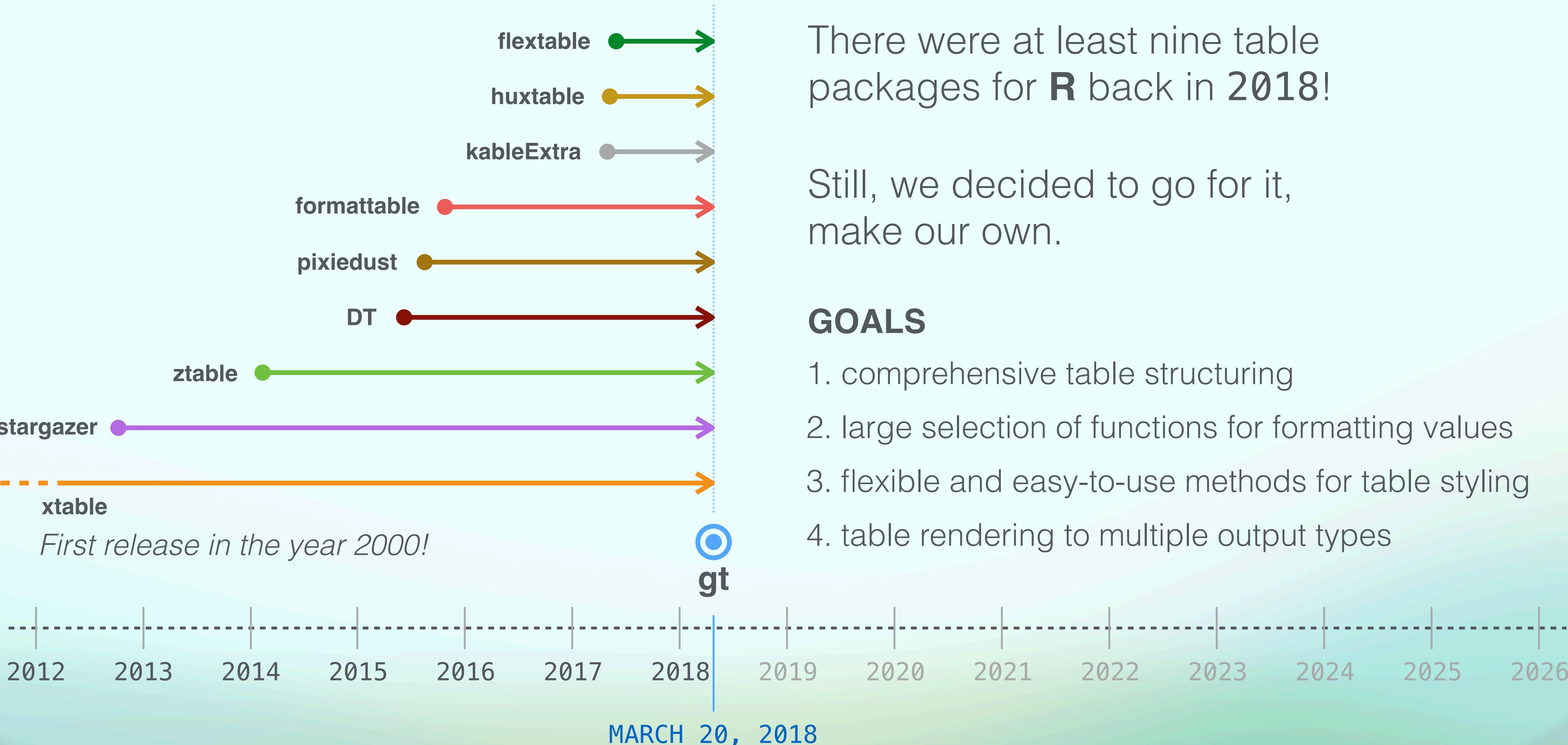
When gt Started



MARCH 20, 2018

Started a week after joining the company.

When gt Started

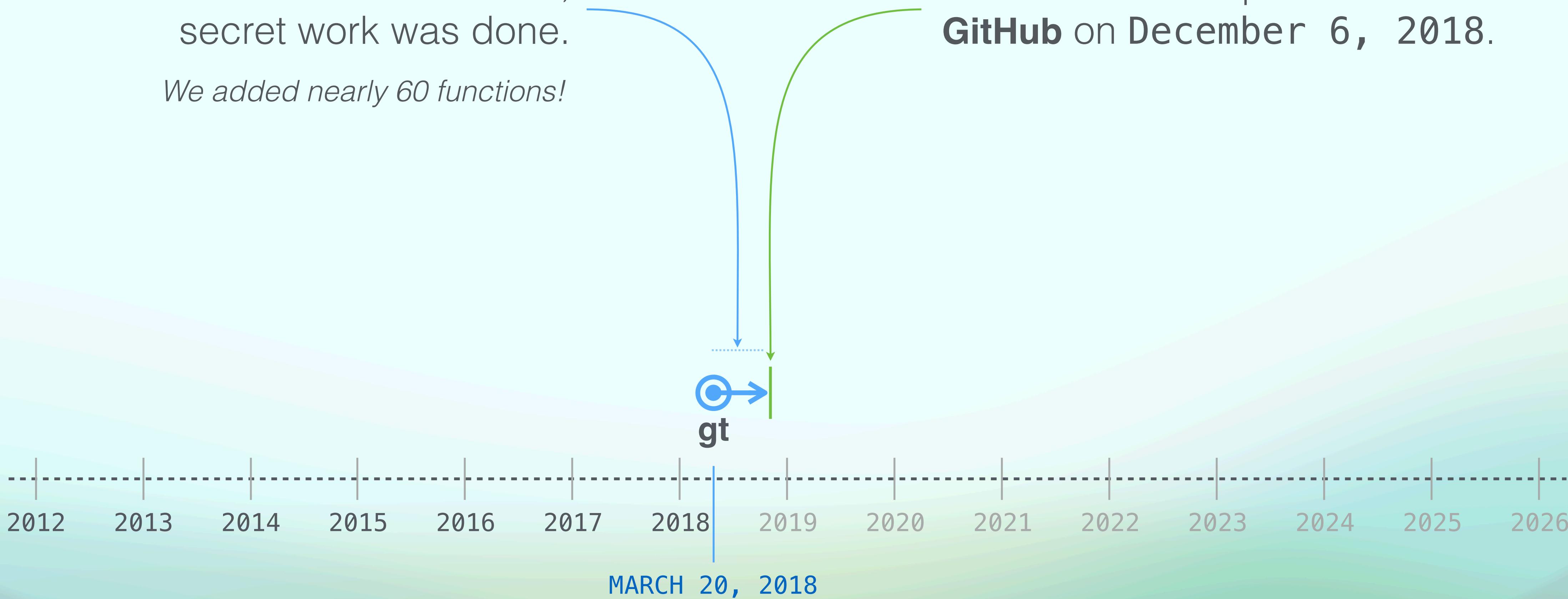


When gt Was No Longer a Secret Project

For a few months in 2018,
secret work was done.

We added nearly 60 functions!

Then we made it public on
GitHub on December 6, 2018.



When gt Was No Longer a Secret Project

The API

Create Table	
<code>gt()</code>	Create a gt table object
<code>gt_preview()</code>	Preview a gt table object
Create/Modify Parts	
<code>tab_header()</code>	Add a table header
<code>tab_spanner()</code>	Add a spanner column label
<code>tab_row_group()</code>	Add a row group
<code>tab_stubhead_label()</code>	Add label text to the stubhead
<code>tab_footnote()</code>	Add a footnote
<code>tab_source_note()</code>	Add a source note citation
<code>tab_options()</code>	Modify the table output options
<code>tab_style()</code>	Add custom styles to one or more cells
Format Data	
<code>fmt()</code>	Set a column format with a formatter function
<code>fmt_number()</code>	Format numeric values
<code>fmt_scientific()</code>	Format values to scientific notation
<code>fmt_percent()</code>	Format values as a percentage
<code>fmt_currency()</code>	Format values as currencies
<code>fmt_date()</code>	Format values as dates
<code>fmt_time()</code>	Format values as times
<code>fmt_datetime()</code>	Format values as date-times
<code>fmt_missing()</code>	Format missing values
<code>fmt_passthrough()</code>	Format by simply passing data through
<code>text_transform()</code>	Perform targeted text transformation with a function
<code>data_color()</code>	Set data cell colors using a palette or a color function
Modify Columns	
<code>cols_align()</code>	Set the alignment of columns
<code>cols_hide()</code>	Hide one or more columns
<code>cols_label()</code>	Relabel one or more columns
<code>cols_merge()</code>	Merge two columns to a single column
<code>cols_merge_range()</code>	Merge two columns to a value range column
<code>cols_merge_uncert()</code>	Merge two columns to a value & uncertainty column
<code>cols_move()</code>	Move one or more columns
<code>cols_move_to_end()</code>	Move one or more columns to the end
<code>cols_move_to_start()</code>	Move one or more columns to the start
<code>cols_split_delim()</code>	Create group names and column labels via delimited names
Modify Rows	
<code>row_group_order()</code>	Modify the ordering of any row groups
Add Rows	
<code>summary_rows()</code>	Add summary rows using aggregation functions
Export Table	
<code>as_raw_html()</code>	Get the HTML content of a gt table
<code>as_latex()</code>	Output a gt object as LaTeX
<code>as_rtf()</code>	Save a gt object as an RTF file
<code>extract_summary()</code>	Extract a summary list from a gt object
Shiny	
<code>render_gt()</code>	gt table render function for use in Shiny
<code>gt_output()</code>	Create a gt table output element for Shiny
Information	
<code>info_date_style()</code>	View a table with info on date styles
<code>info_time_style()</code>	View a table with info on time styles
Datasets	
<code>countrypops</code>	Yearly populations of countries from 1960 to 2017
<code>sza</code>	Twice hourly solar zenith angles by month & latitude
<code>gtcars</code>	Deluxe automobiles from the 2014–2017 period
<code>sp500</code>	Daily S&P 500 Index data from 1950 to 2015
<code>pizzaplace</code>	A year of pizza sales from a pizza place
<code>exibble</code>	A toy example tibble for testing with gt: exibble
Location Helpers	
	Helpers for targeting multiple cells in different locations
<code>cells_title()</code>	
<code>cells_column_labels()</code>	
<code>cells_group()</code>	
<code>cells_stub()</code>	
<code>cells_summary()</code>	
<code>cells_data()</code>	

Then we made it public on
GitHub on December 6, 2018.

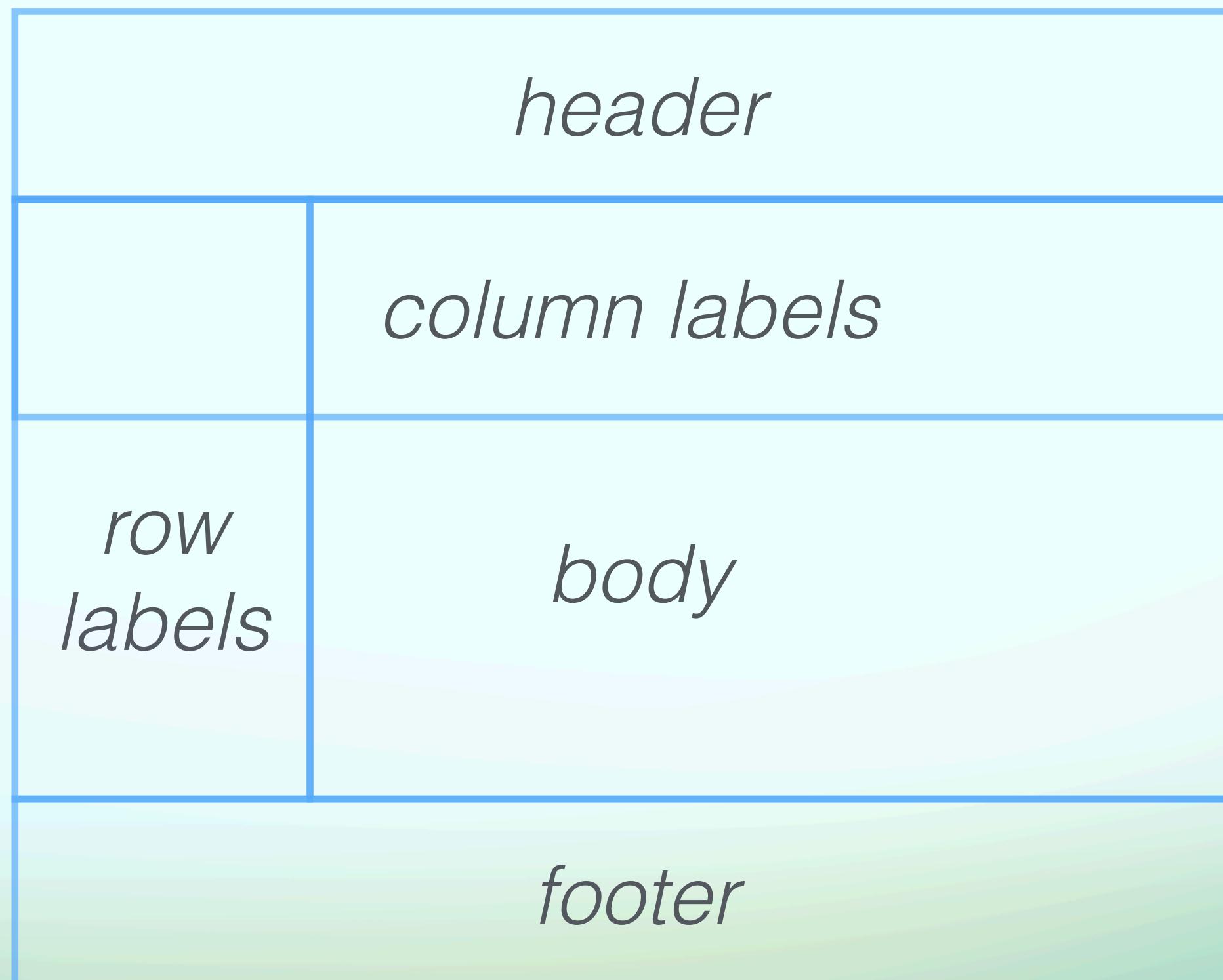


Our First Goal for gt

By this public release we were able to realize our first goal:
comprehensive table structuring

This was our basic blueprint for a table.

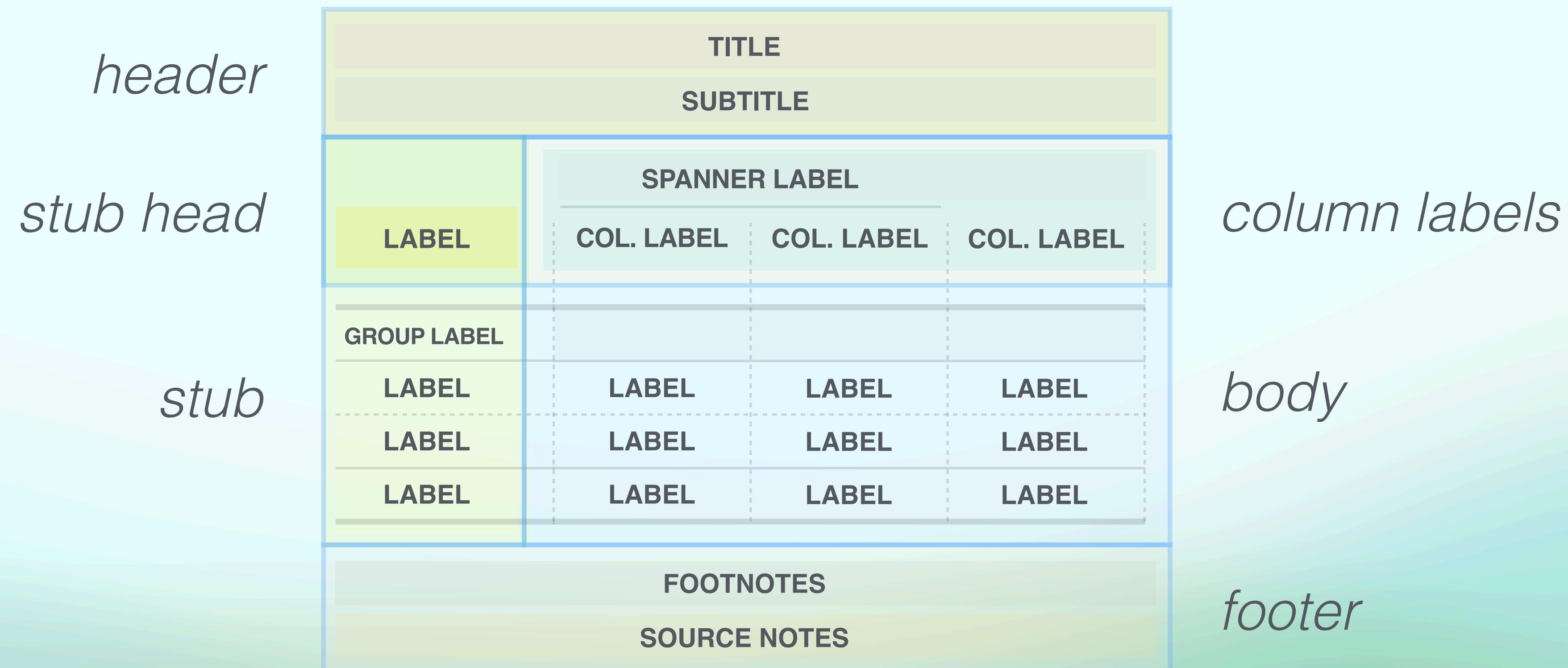
It has composable and configurable parts.



We worked on formalizing the components to make everything easier to understand.

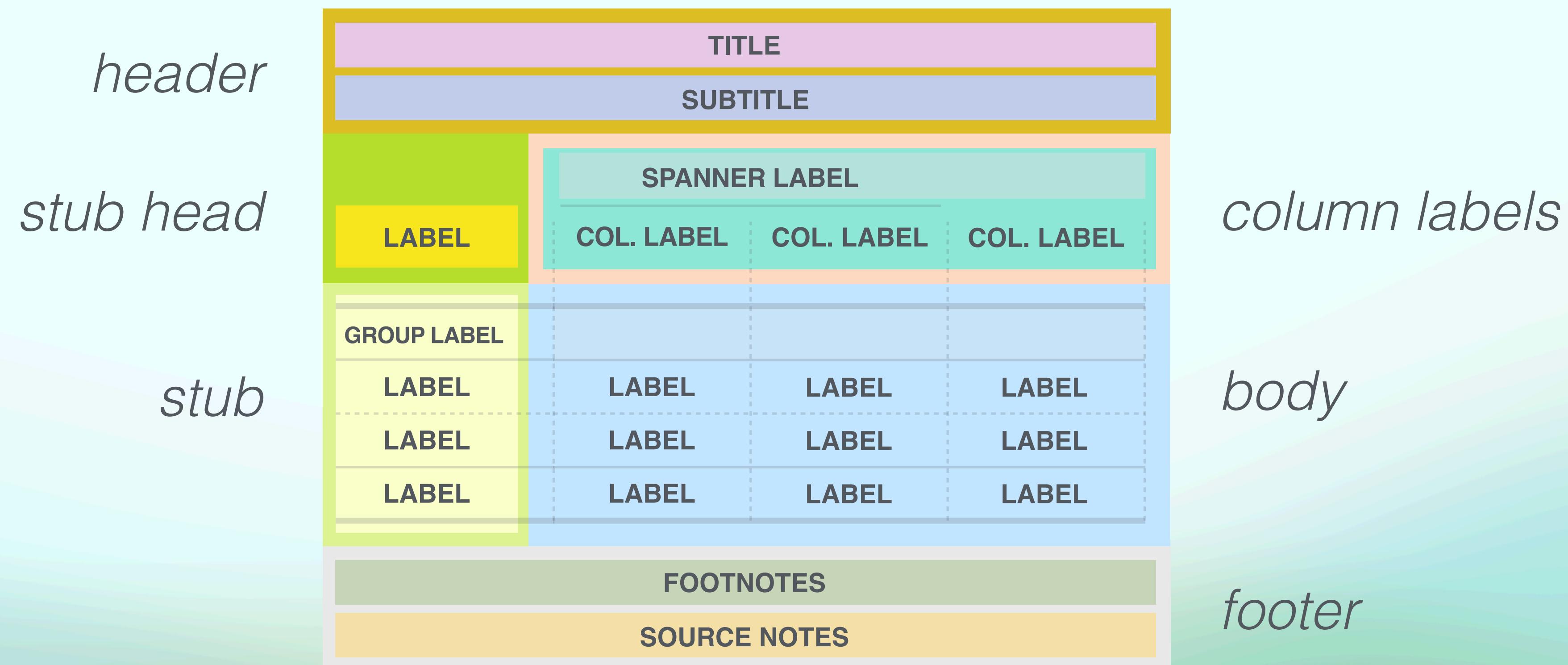
Our First Goal for gt comprehensive table structuring

We eventually solidified our table framework...



Our First Goal for gt comprehensive table structuring

...and the nomenclature stuck. All this helps with explaining how **gt** works.



Our Second Goal for gt

The next stage was to get the package onto **CRAN**.

In doing so we had to ensure we met the second goal.

large selection of functions for formatting values

`fmt_number()`

`fmt_scientific()`

`fmt_percent()`

`fmt_currency()`

`fmt_date()`

`fmt_time()`

`fmt_datetime()`

`fmt_markdown()`

`fmt_passthrough()`

`fmt()`

*This is a pretty reasonable
starter pack of formatters.*



Our Second Goal for gt

The next stage was to get the package onto **CRAN**.

In doing so we had to ensure we met the second goal.

large selection of functions for formatting values

`fmt_number()`

`fmt_scientific()`

`fmt_percent()`

`fmt_currency()`

`fmt_date()`

`fmt_time()`

`fmt_datetime()`

`fmt_markdown()`

`fmt_passthrough()`

`fmt()`



Taken altogether, it took a while to get to the first CRAN release.

START

MARCH 20, 2018

CRAN

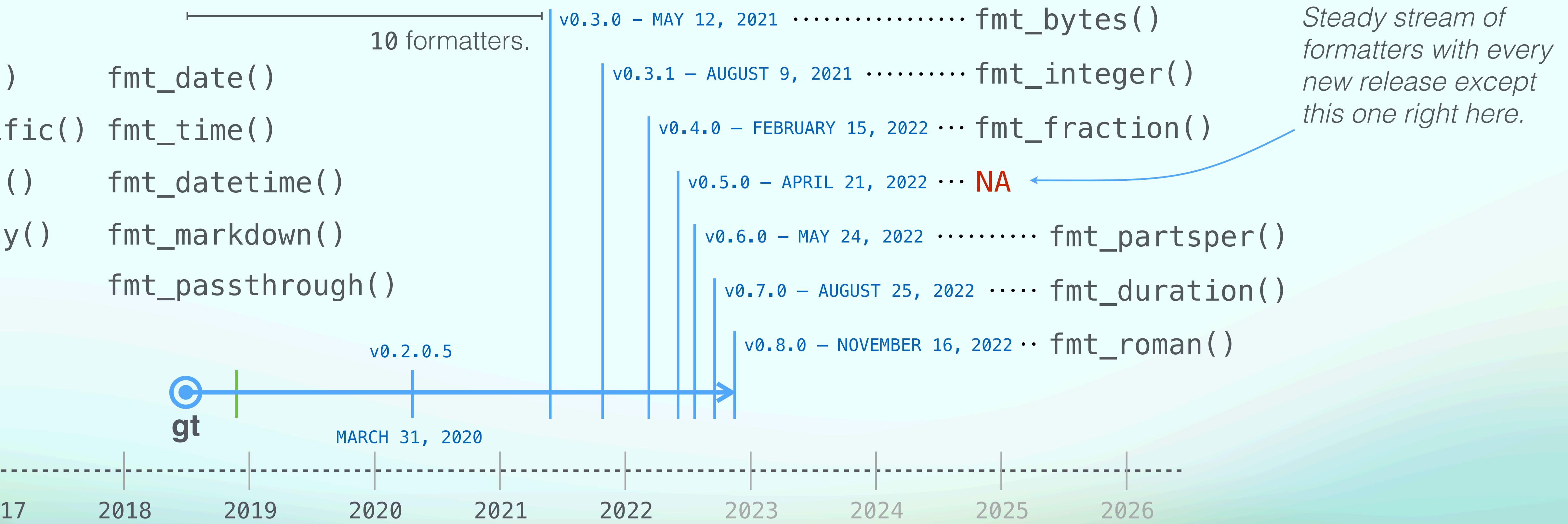
v0.2.0.5 – MARCH 31, 2020

DIFF

APPROX 2 YEARS

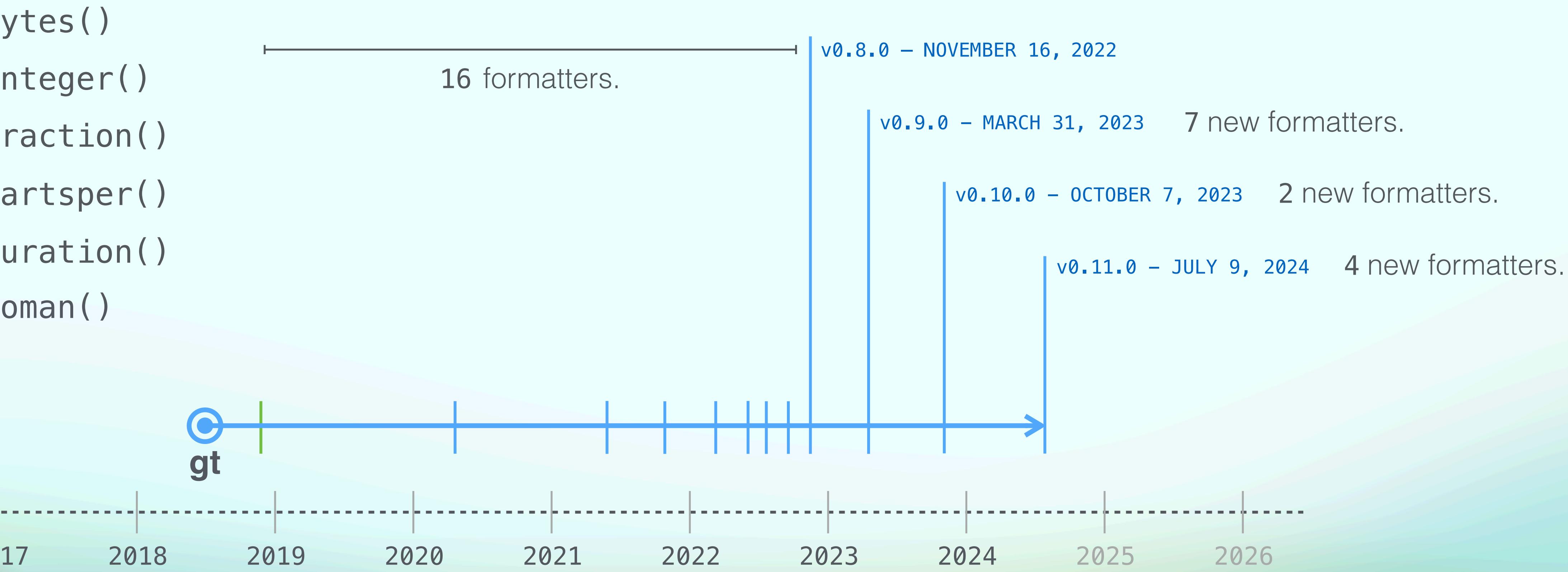
Our Second Goal for gt large selection of functions for formatting values

More formatting functions added in 2021 and 2022.



Our Second Goal for gt large selection of functions for formatting values

Then a large amount of new formatters were added in the last 3 releases.



Our Second Goal for gt large selection of functions for formatting values

The current set of formatters covers a lot of formatting tasks.

TOTAL: 29 formatters. (16 + 7 + 2 + 4)

fmt_number()	fmt_date()
fmt_scientific()	fmt_time()
fmt_percent()	fmt_datetime()
fmt_currency()	fmt_markdown()
fmt()	fmt_passthrough()

fmt_bytes()

fmt_integer()

fmt_fraction()

fmt_partsper()

fmt_duration()

fmt_roman()

fmt_spelled_num()

fmt_index()

fmt_url()

fmt_image()

fmt_flag()

fmt_bins()

fmt_auto()

fmt_icon()

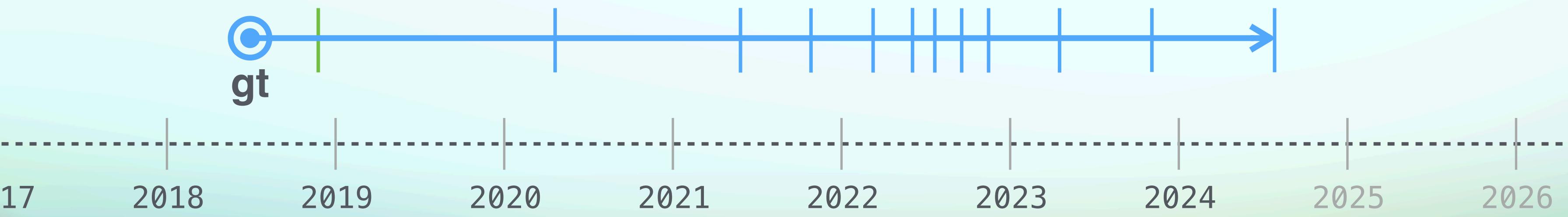
fmt_units()

fmt_chem()

fmt_email()

fmt_tf()

fmt_country()



Our Third Goal for gt flexible and easy-to-use methods for table styling

THREE METHODS

One way of table styling is with:

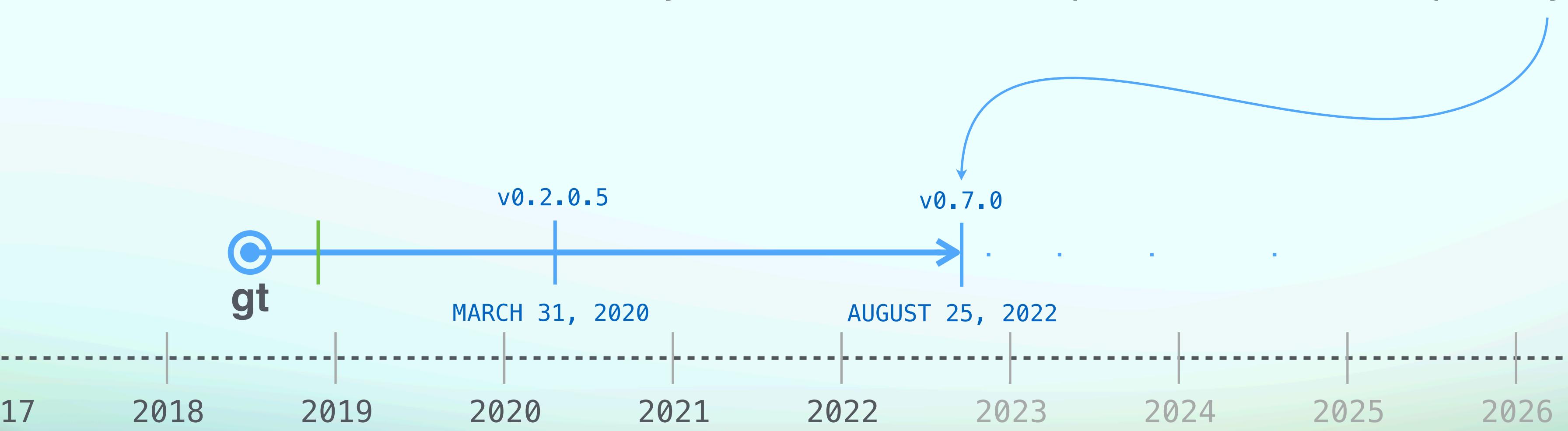
`tab_style()`

Another way is to use this function:

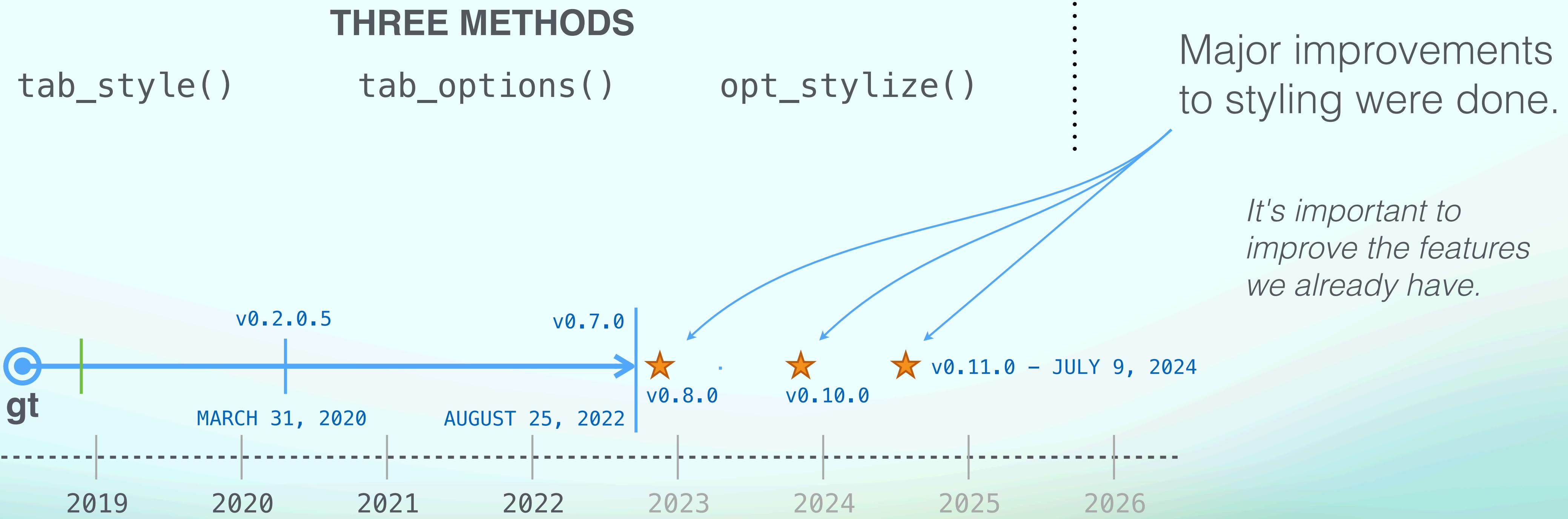
`tab_options()`

We eventually put in this one:

`opt_stylize()`



Our Third Goal for gt flexible and easy-to-use methods for table styling



Our Third Goal for gt Examples of Really Good Styling in Tables



liga endesa

¿Ganan Campeonatos Las Defensas o Los Ataques?

Morado es bueno y naranja es malo | Cómo ha afectado la defensa o el ataque de Barça y Real Madrid en el resto de la liga | Ordenado por NetRating | Temporada regular 2021/2022

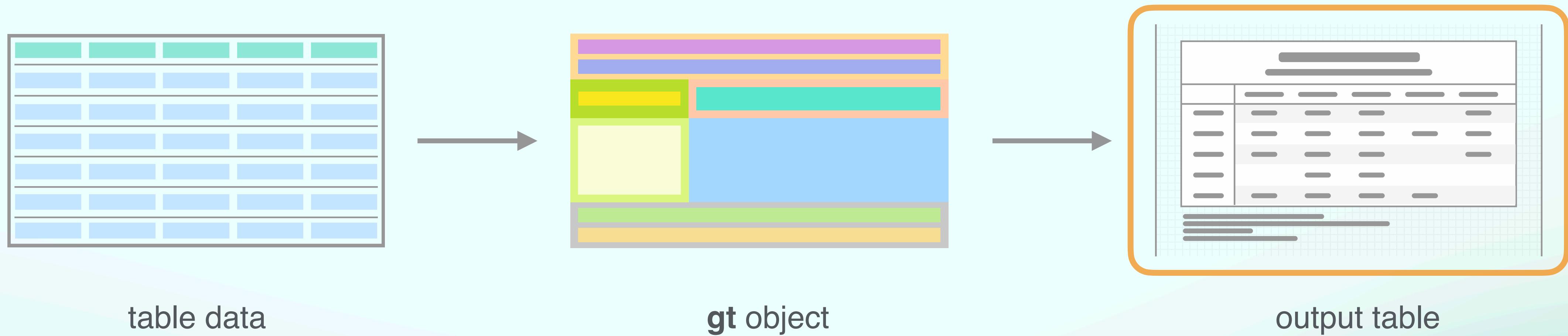
RK EQUIPO	MEDIA TOTAL			MEDIA VS BAR & MAD			DIFERENCIA EN %		WINS	LOST
	OFF RT	DEF RT	NET RT	OFF RT	DEF RT	NET RT	OFF%	DEF%		
1 BARÇA 27-7	111.89	99.33	12.57	125.34	109.45	15.89	12.02%	10.20%		
2 REAL MADRID 25-9	112.27	103.02	9.25	109.45	125.34	-15.89	-2.51%	21.66%		
5 LENOVO TENERIFE 21-13	114.53	108.48	6.04	97.81	101.52	-3.71	-14.60%	-6.42%		
3 VALENCIA BASKET 23-11	112.18	107.01	5.17	113.57	111.51	2.07	1.24%	4.20%		
6 BITCI BASKONIA 20-14	110.42	105.77	4.65	108.91	113.73	-4.81	-1.36%	7.52%		
4 JOVENTUT BADALONA 22-12	111.35	107.55	3.80	114.80	126.83	-12.03	3.10%	17.93%		
7 BAXI MANRESA 20-14	108.43	105.74	2.69	99.08	102.62	-3.54	-8.62%	-2.95%		
10 UCAM MURCIA 16-18	107.88	106.03	1.84	99.18	114.07	-14.89	-8.06%	7.58%		
12 UNICAJA 13-21	108.25	107.38	0.87	106.13	104.04	2.09	-1.95%	-3.11%		
11 RÍO BREOGÁN 16-18	110.01	111.17	-1.17	95.56	113.35	-17.78	-13.13%	1.96%		
8 GRAN CANARIA 17-17	106.68	109.15	-2.48	101.30	115.28	-13.98	-5.04%	5.61%		
9 SURÉ BILBAO BASKET 16-18	106.95	111.25	-4.30	94.14	112.18	-18.04	-11.98%	0.84%		
14 URBES FUENLABRADA 12-22	108.56	113.12	-4.56	102.26	115.91	-13.66	-5.80%	2.47%		
15 MONBUS OBRADOIRO 12-22	108.90	114.10	-5.20	101.30	115.44	-14.14	-6.98%	1.18%		
17 MORA BANC ANDORRA 11-23	103.98	109.42	-5.45	103.17	119.18	-16.01	-0.77%	8.92%		
18 HEREDA SAN PABLO BURGOS 10-24	105.38	112.21	-6.82	92.81	110.27	-17.46	-11.93%	-1.73%		
16 CASADEMONT ZARAGOZA 12-22	101.02	109.52	-8.50	92.07	112.31	-20.23	-8.86%	2.54%		
13 COOSUR REAL BETIS 13-21	103.87	112.69	-8.83	84.89	107.27	-22.38	-18.27%	-4.82%		

Datos: PlayByPlay_Acb.com
Fórmula Posesiones: tiros de campo intentados + tiros libres intentados/2 + balones perdidos - rebotes ofensivos
Fórmula Ofrt y Dfrt: (puntos/posesiones) x 100
Gráfico: Ivo Villanueva •

Our Fourth Goal for gt

In the general **gt** workflow, the last step is rendering and we wanted...

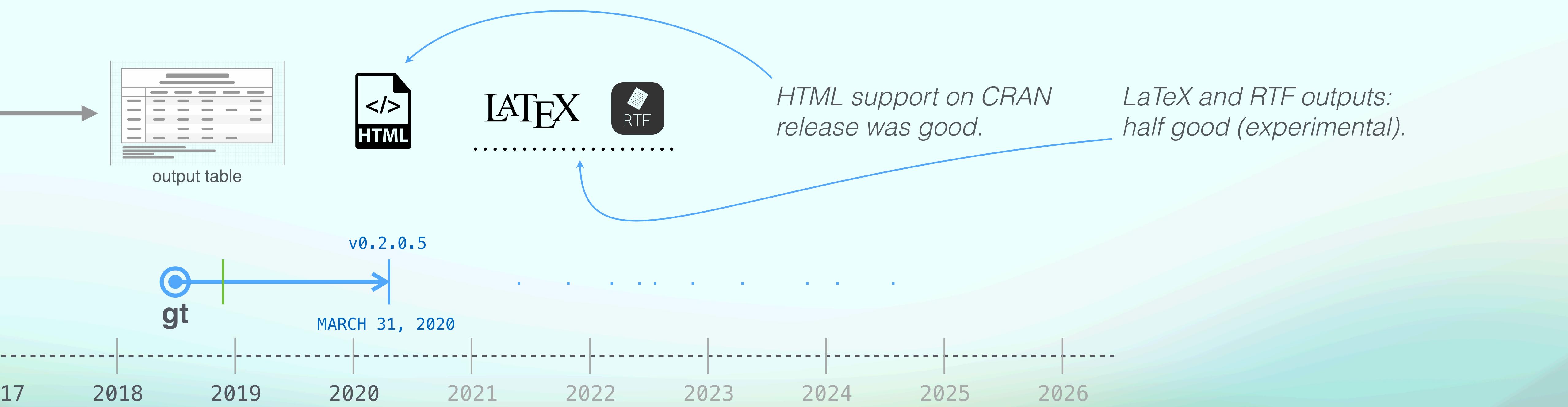
table rendering to multiple output types



Also, you needn't to change your table code depending on the output type.

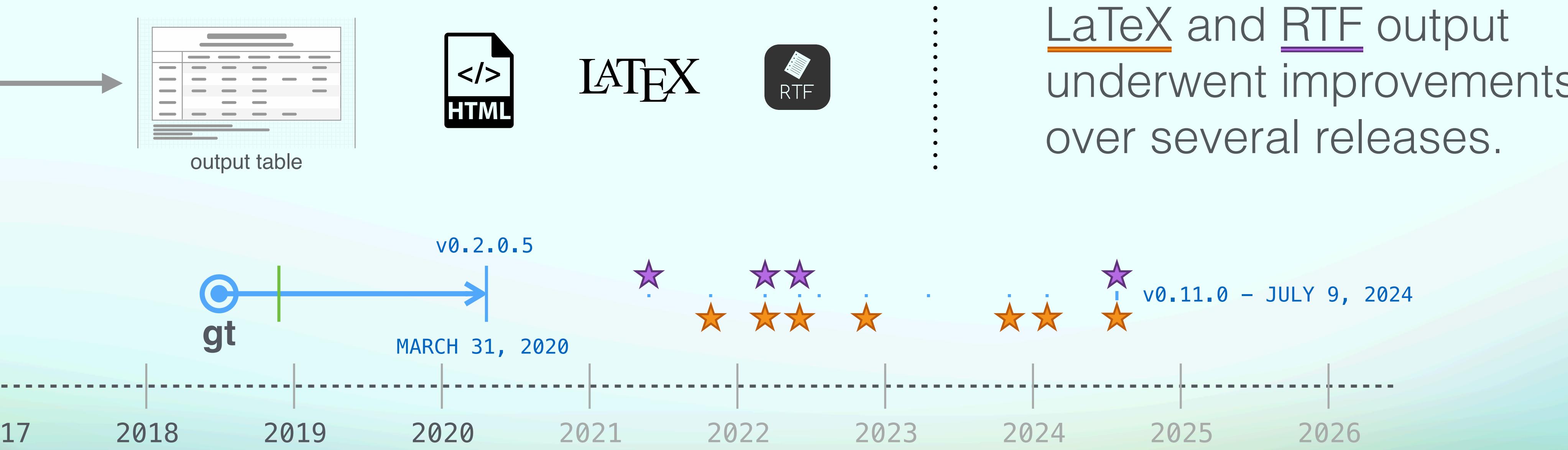
Our Fourth Goal for gt table rendering to multiple output types

Three output types were targeted: HTML, LaTeX, and RTF.



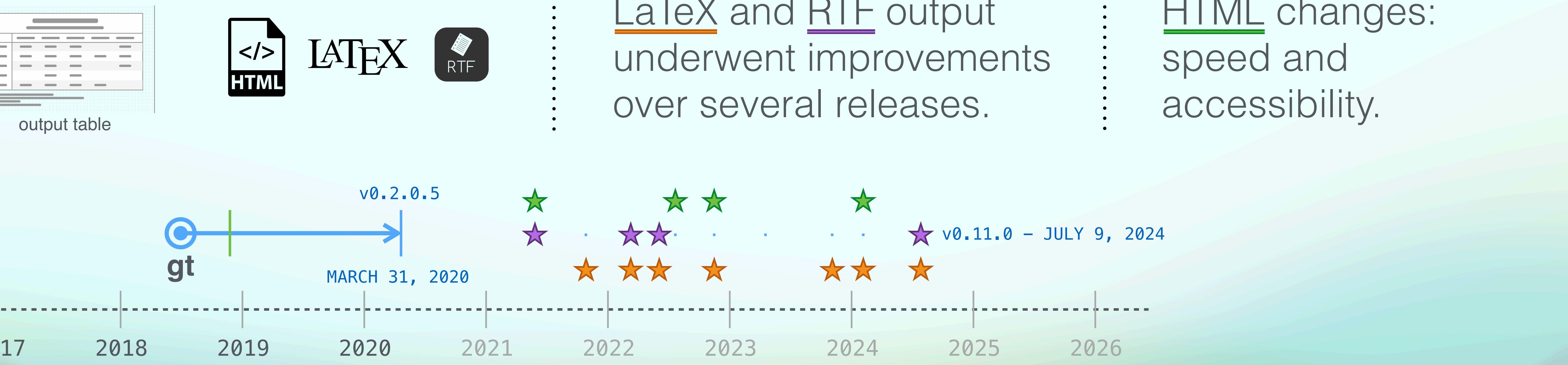
Our Fourth Goal for gt table rendering to multiple output types

Three output types were targeted: HTML, LaTeX, and RTF.



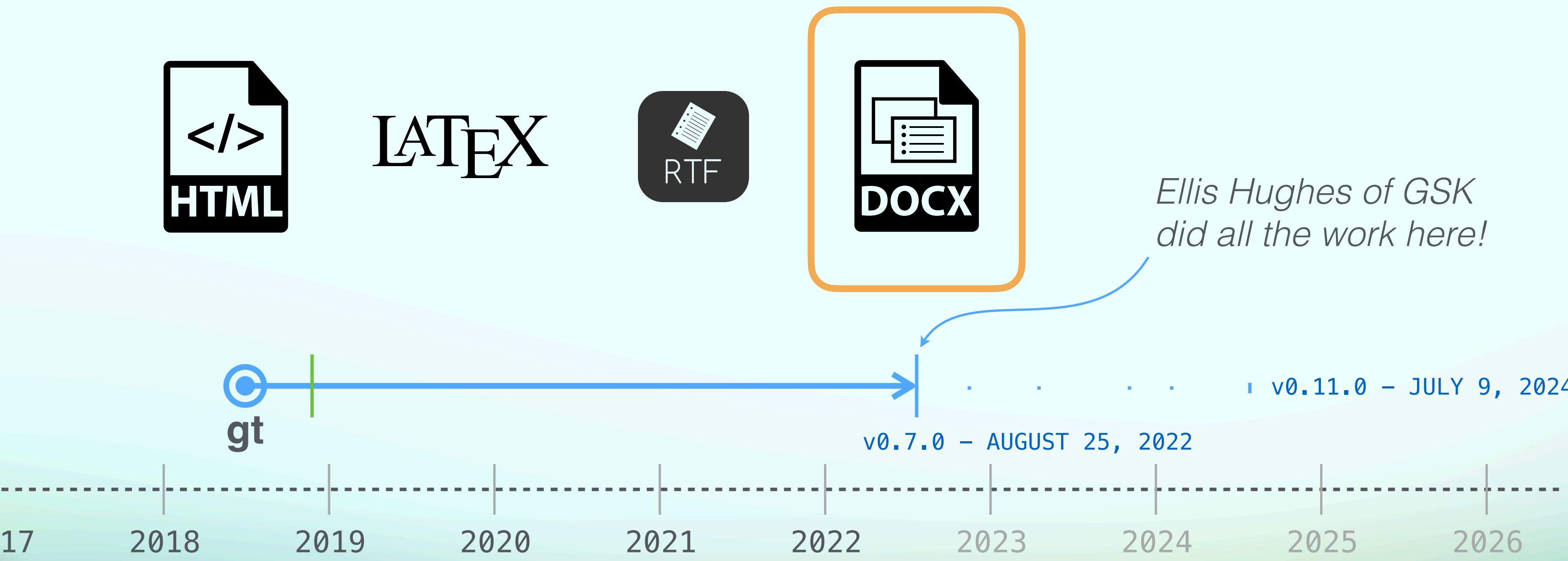
Our Fourth Goal for gt table rendering to multiple output types

Three output types were targeted: HTML, LaTeX, and RTF.



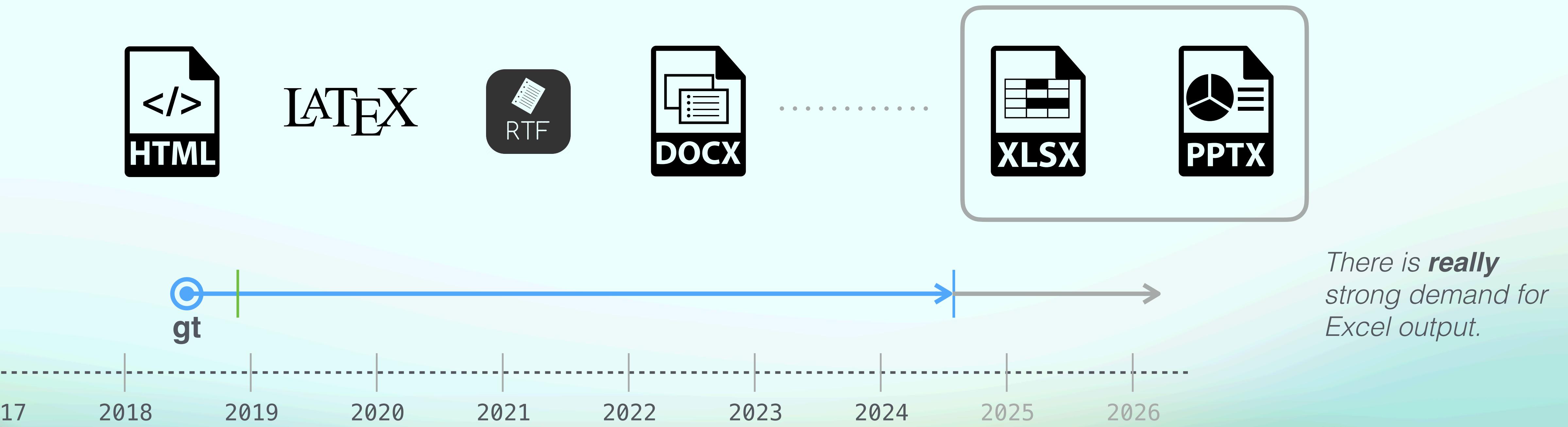
Our Fourth Goal for gt table rendering to multiple output types

Later, in August 2022, the .docx/Word output type was implemented.



Our Fourth Goal for gt table rendering to multiple output types

Though not possible now, we are looking at adding **Excel** and **PowerPoint** output.



We Met Our Goals But We Had Even More

.....

INITIAL GOALS

- ✓ 1. comprehensive table structuring
 - ✓ 2. large selection of functions for formatting values
 - ✓ 3. flexible and easy-to-use methods for table styling
 - ✓ 4. table rendering to multiple output types
-

We had a lot of goals.
We did pretty good on
the initial set.

As we were developing
the package. We also
developed new goals.

We Met Our Goals But We Had Even More



INITIAL GOALS

LATER GOALS

1. useful across many disciplines and use cases
2. localization options for users all over the world
3. good documentation to get you building quickly
4. useful for Pharma's specific table-making needs

There were even more things but we have to stop somewhere.

We Met Our Goals But We Had Even More



INITIAL GOALS

LATER GOALS

1. useful across many disciplines and use cases
2. localization options for users all over the world
3. good documentation to get you building quickly
4. useful for Pharma's specific table-making needs



We added formatters both general and domain-specific:

`fmt_scientific()`
`fmt_partsper()`
`fmt_chem()`

We've also included formatter options that target certain conventions.

Which disciplines targeted?

Physics · Biology · History · Mathematics Chemistry · Oceanography
Optics · Astronomy · Botany Geomorphology · Hydrology · Ecology
Fluid Dynamics · Tectonics · Geography · Biophysics · Zoology · Geology
Climatology · Marine Biology · Geophysics · Paleontology · Journalism
Education · Law · Engineering · Computer Science · Medicine · Pharmacy
Psychiatry · Sports Medicine · Business · Economics · Sociology
Dentistry · Pediatrics · Nutrition · Political Science

We Met Our Goals But We Had Even More



INITIAL GOALS

LATER GOALS

1. useful across many disciplines and use cases
2. localization options for users all over the world
3. good documentation to get you building quickly
4. useful for Pharma's specific table-making needs



Most formatting functions have a **locale** option.

This ensures that numbers, dates/times, and even words fit the language and region.

We now support 574 locales and update our code with **CLDR** guidance and data.

We Met Our Goals But We Had Even More



INITIAL GOALS

LATER GOALS

1. useful across many disciplines and use cases
2. localization options for users all over the world
3. good documentation to get you building quickly
4. useful for Pharma's specific table-making needs



Our documentation is something we constantly try to improve.

We found it important to have many examples for every function in the package.

A big thing is to make sure wherever you land in the docs, you find something useful fast.

We Met Our Goals But We Had Even More



INITIAL GOALS

LATER GOALS

1. useful across many disciplines and use cases
2. localization options for users all over the world
3. good documentation to get you building quickly
4. useful for Pharma's specific table-making needs



We know that Pharma has specific needs for building tables.

So we added RTF output format for Pharma-specific tables.

We have also introduced functionality for splitting tables across pages for paginated output formats.

The Future of gt

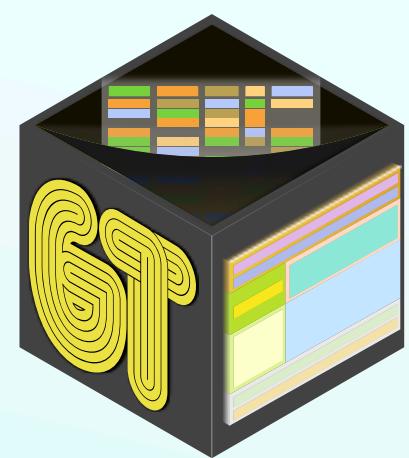
Here's a shortlist of things we need for a future **gt**:

- Row ordering functionality — so you don't have to order rows beforehand
- Improvements to footnotes — flexibility for affixing the marks
- **Excel** output tables — it's a popular file format
- Overall better table-splitting — better options, more dependable
- Integration with database tables — so it can be used as input data
- Ways to better style text — e.g., styling spans of text in a cell
- Methods for merging cells — right now: no way to merge adjacent cells



A Relatively New Initiative: Great Tables

We have been working super hard on the **Great Tables** Python package.



Great Tables

gt



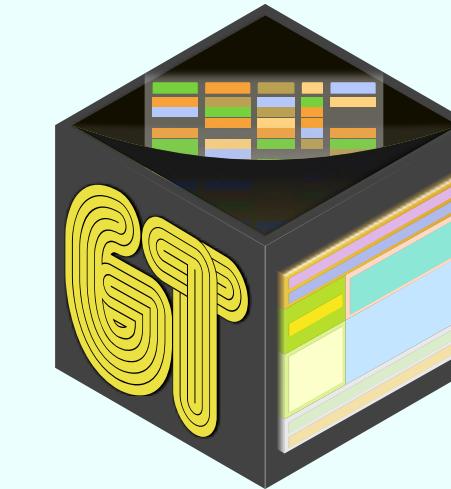
Goal is to bring all that **gt** goodness over to the Python language.

We are in the process of porting over all features while making it fit in well within the Python ecosystem.

Great Tables Is Actually Very Capable Today!

Tables made with
Great Tables look
indistinguishable from
their **gt** counterparts.

We are porting over
the most important
features first, with
secondary features
coming later.



	Gas-phase reactions of selected mercaptan compounds			
	Reaction Rate Constant (298 K), cm ³ molecules ⁻¹ s ⁻¹			
	OH	NO ₃	Cl	
methanethiol	CH ₄ S	3.50 × 10 ⁻¹¹	9.20 × 10 ⁻¹³	2.00 × 10 ⁻¹⁰
ethanethiol	C ₂ H ₆ S	4.50 × 10 ⁻¹¹	1.21 × 10 ⁻¹²	1.75 × 10 ⁻¹⁰
propanethiol	C ₃ H ₈ S	5.30 × 10 ⁻¹¹	—	2.14 × 10 ⁻¹⁰
2-propanethiol	C ₃ H ₈ S	3.90 × 10 ⁻¹¹	—	2.70 × 10 ⁻¹⁰
1-butanethiol	C ₄ H ₁₀ S	5.60 × 10 ⁻¹¹	—	—
2-methyl-1-propanethiol	C ₄ H ₁₀ S	4.60 × 10 ⁻¹¹	—	—
2-butanethiol	C ₄ H ₁₀ S	3.80 × 10 ⁻¹¹	—	1.65 × 10 ⁻¹⁰
t-butylsulfide	C ₄ H ₁₀ S	2.90 × 10 ⁻¹¹	—	—
2-methylbutanethiol	C ₅ H ₁₂ S	5.20 × 10 ⁻¹¹	—	—
n-pantanethiol	C ₅ H ₁₂ S	—	—	1.97 × 10 ⁻¹⁰
1,2-ethanedithiol	C ₂ H ₆ S ₂	3.80 × 10 ⁻¹¹	—	—

	Gas-phase reactions of selected mercaptan compounds			
	Reaction Rate Constant (298 K), cm ³ molecules ⁻¹ s ⁻¹			
	OH	NO ₃	Cl	
methanethiol	CH ₄ S	3.50 × 10 ⁻¹¹	9.20 × 10 ⁻¹³	2.00 × 10 ⁻¹⁰
ethanethiol	C ₂ H ₆ S	4.50 × 10 ⁻¹¹	1.21 × 10 ⁻¹²	1.75 × 10 ⁻¹⁰
propanethiol	C ₃ H ₈ S	5.30 × 10 ⁻¹¹	—	2.14 × 10 ⁻¹⁰
2-propanethiol	C ₃ H ₈ S	3.90 × 10 ⁻¹¹	—	2.70 × 10 ⁻¹⁰
1-butanethiol	C ₄ H ₁₀ S	5.60 × 10 ⁻¹¹	—	—
2-methyl-1-propanethiol	C ₄ H ₁₀ S	4.60 × 10 ⁻¹¹	—	—
2-butanethiol	C ₄ H ₁₀ S	3.80 × 10 ⁻¹¹	—	1.65 × 10 ⁻¹⁰
t-butylsulfide	C ₄ H ₁₀ S	2.90 × 10 ⁻¹¹	—	—
2-methylbutanethiol	C ₅ H ₁₂ S	5.20 × 10 ⁻¹¹	—	—
n-pantanethiol	C ₅ H ₁₂ S	—	—	1.97 × 10 ⁻¹⁰
1,2-ethanedithiol	C ₂ H ₆ S ₂	3.80 × 10 ⁻¹¹	—	—

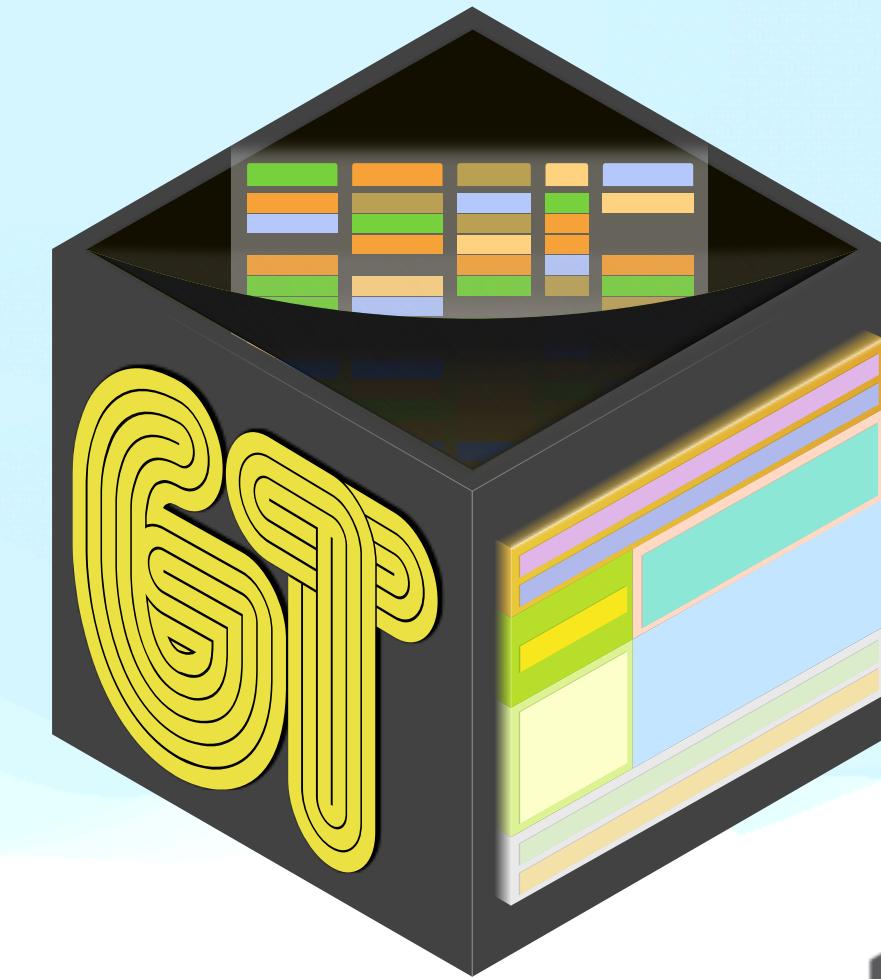
The Future Is Full of Great Tables



Great Tables

You deserve more-than-adequate tables in your life!

Try out one or more of these packages, and...



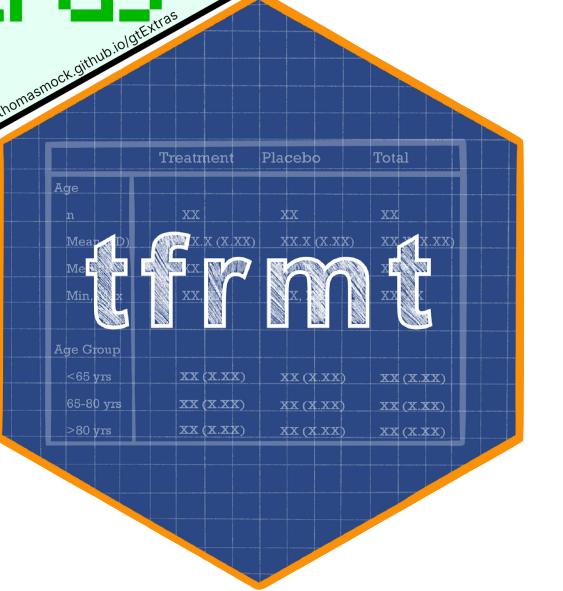
gt



gtsummary



gt
Extras



related table packages

The brand-new, Python version of **reactable**!



Order	Created	Status
2300	2019-04-01	Pending
2301	2019-04-02	Paid
2302	2019-04-03	Canceled
2303	2019-04-04	Pending
2304	2019-04-05	Paid

Order	Created	Status
2300	2019-04-01	Pending
2301	2019-04-02	Paid
2302	2019-04-03	Canceled
2303	2019-04-04	Pending
2304	2019-04-05	Paid

make	MPG (city)	MPG (highway)
Acura Integra	25	31
Acura Legend	18	25
Audi 90	20	26
Audi 100	19	26
BMW 535i	22	30

Animal	Body (kg)	Brain (g)
beaver	1.35	8.1
cow	465	423
wolf	36.33	119.5
goat	27.66	115

Movie	Rating	Votes
Silent Serpent	★★★★★	115
Nowhere to Hyde	★★★★★	37
The Ape-Man Goes to Mars	★★★★★	60
A Menace in Venice	★★★★★	99

Character	height	mass	gender
Luke Skywalker	172	77	masculine
Human			
C-3PO	167	75	masculine
Droid			
R2-D2	96	32	masculine
Droid			
Darth Vader	202	136	masculine
Human			

1–4 of 87 rows

Previous 1 2 3 4 5 ... 22 Next

manufacturer	model	type	price
Acura	Integra	Small	\$15,000
Acura	Legend	Midsize	\$33,000
Audi	90	Compact	\$21,000
Audi	100	Midsize	\$37,000
BMW	535i	Midsize	\$42,000

Total \$1814.4

1–5 of 93 rows

Previous 1 2 3 4 5 ... 19 Next

<https://github.com/machow/reactable-py>

<https://machow.github.io/reactable-py/demos/>

Thank You!

Adequate Tables? No, We Want Great Tables

<https://github.com/rich-iannone/presentations>

<https://github.com/rstudio/gt>

<https://github.com/posit-dev/great-tables>



rich-iannone



rich@posit.co