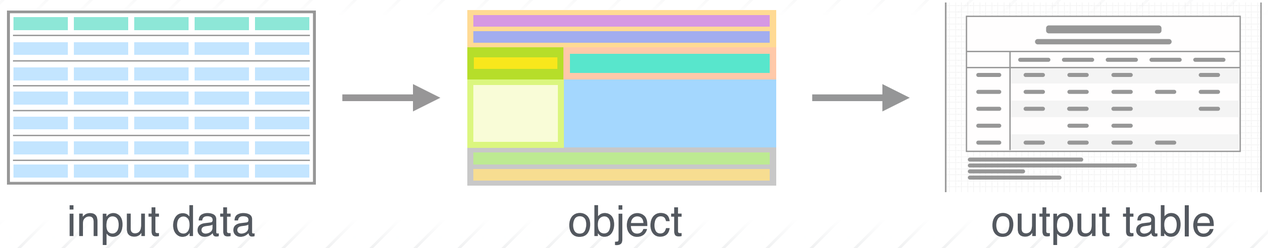
How to Use **Great Tables** for Really Nice Tables in **Quarto** documents



Display Tables... What Do We Want?

shape: (15, 5)

product	revenue_dollars	revenue_pct	margin_dollars	margin_pct
str	f64	f64	f64	f64
"Grinder"	904.5	0.03	567.96	0.04
"Moka pot"	2045.25	0.07	181.08	0.01
"Cold brew"	288.75	0.01	241.77	0.02
"Filter"	404.25	0.01	70.01	0.02
"Drip machine"	2632.0	0.1	1374.45	0.09
…	…	…	…	…
"Dripper"	575.75	0.02	139.02	0.01
"Scale"	3801.0	0.13	2910.29	0.19
"Kettle"	756.25	0.02	617.52	0.04
"Espresso Machi…	8406.0	0.28	3636.44	0.24
"Total"	30284.25	1.0	14932.16	1.0

Less of This

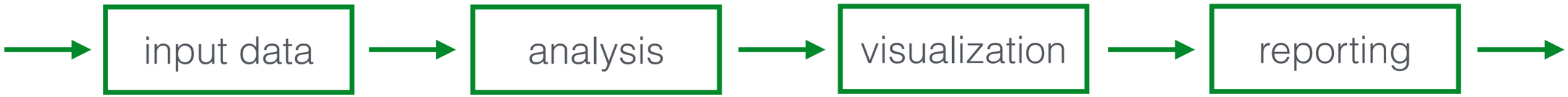
Sales of Coffee Equipment

		Revenue		Profit		
icon	product	Earnings	Fraction	Earnings	Fraction	monthly_sales
	Grinder	$904K	3%	$568K	4%	
	Moka pot	$2.05M	7%	$181K	1%	
	Cold brew	$289K	1%	$242K	2%	
	Filter	$404K	1%	$70.0K	0%	
	Drip machine	$2.63M	9%	$1.37M	9%	
	AeroPress	$2.60M	9%	$1.29M	9%	
	Pour over	$846K	3%	$365K	2%	
	French press	$1.11M	4%	$748K	5%	
	Cezve	$2.51M	9%	$1.97M	13%	
	Chemex	$3.14M	11%	$818K	6%	
	Scale	$3.80M	13%	$2.91M	20%	
	Kettle	$756K	3%	$618K	4%	
	Espresso Machine	$8.41M	29%	$3.64M	25%	
	Total	**$29.4M**	**102%**	**$14.8M**	**100%**	

More of This

Why Do We Want it?

We can all benefit from a reproducible workflow.



How Do You Make Tables Today?

Raw Table

shape: (15, 5)

product	revenue_dollars	revenue_pct	margin_dollars	margin_pct
str	f64	f64	f64	f64
"Grinder"	904.5	0.03	567.96	0.04
"Moka pot"	2045.25	0.07	181.08	0.01
"Cold brew"	288.75	0.01	241.77	0.02
"Filter"	404.25	0.01	70.01	0.02
"Drip machine"	2632.0	0.1	1374.45	0.09
…	…	…	…	…
"Dripper"	575.75	0.02	139.02	0.01
"Scale"	3801.0	0.13	2910.29	0.19
"Kettle"	756.25	0.02	617.52	0.04
"Espresso Machi…	8406.0	0.28	3636.44	0.24
"Total"	30284.25	1.0	14932.16	1.0

You could present this to others,
but it's not recommended.

How Do You Make Tables Today?

Raw Table

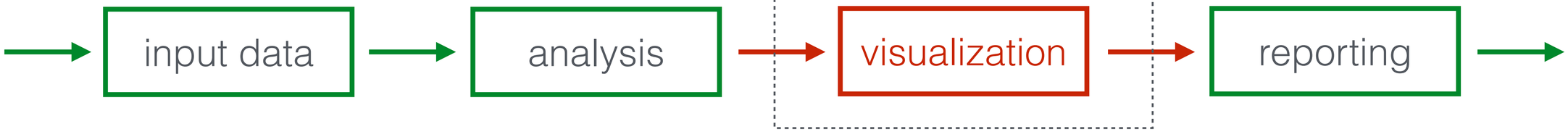
shape: (15, 5)

product	revenue_dollars	revenue_pct	margin_dollars	margin_pct
str	f64	f64	f64	f64
"Grinder"	904.5	0.03	567.96	0.04
"Moka pot"	2045.25	0.07	181.08	0.01
"Cold brew"	288.75	0.01	241.77	0.02
"Filter"	404.25	0.01	70.01	0.02
"Drip machine"	2632.0	0.1	1374.45	0.09
…	…	…	…	…
"Dripper"	575.75	0.02	139.02	0.01
"Scale"	3801.0	0.13	2910.29	0.19
"Kettle"	756.25	0.02	617.52	0.04
"Espresso Machi…	8406.0	0.28	3636.44	0.24
"Total"	30284.25	1.0	14932.16	1.0

Excel

Product	Revenue $ (000's)	Revenue %	Margin $ (000's)	Margin %
Grinder	$904.50	3%	$567.96	4%
Moka pot	$2,045.25	7%	$181.08	1%
Cold brew	$288.75	1%	$241.77	2%
Filter	$404.25	1%	$70.01	0%
Drip machine	$2,632.00	10%	$1,374.45	9%
AeroPress	$2,601.50	9%	$1,293.78	9%
Pour over	$846.00	3%	$364.53	2%
French press	$1,113.25	4%	$748.12	5%
Cezve	$2,512.50	8%	$1,969.52	13%
Chemex	$3,137.25	10%	$817.68	5%
Dripper	$575.75	2%	$139.02	1%
Scale	$3,801.00	13%	$2,910.29	19%
Kettle	$756.25	2%	$617.52	4%
Espresso Machine	$8,406.00	28%	$3,636.44	24%
Total	**$30,284.25**	**100%**	**$14,932.16**	**100%**

You could instead make a nice display table with Excel. But your reproducible workflow is now broken.



How Do You Make Tables Today?

Excel

Product	Revenue $ (000's)	Revenue %	Margin $ (000's)	Margin %
Grinder	$904.50	3%	$567.96	4%
Moka pot	$2,045.25	7%	$181.08	1%
Cold brew	$288.75	1%	$241.77	2%
Filter	$404.25	1%	$70.01	0%
Drip machine	$2,632.00	10%	$1,374.45	9%
AeroPress	$2,601.50	9%	$1,293.78	9%
Pour over	$846.00	3%	$364.53	2%
French press	$1,113.25	4%	$748.12	5%
Cezve	$2,512.50	8%	$1,969.52	13%
Chemex	$3,137.25	10%	$817.68	5%
Dripper	$575.75	2%	$139.02	1%
Scale	$3,801.00	13%	$2,910.29	19%
Kettle	$756.25	2%	$617.52	4%
Espresso Machine	$8,406.00	28%	$3,636.44	24%
Total	**$30,284.25**	**100%**	**$14,932.16**	**100%**

Great Tables

Sales of Coffee Equipment

		Revenue		Profit		
	Product	Amount	Percent	Amount	Percent	Monthly Revenue
	Grinder	$904,500	3%	$567,960	4%	
	Moka pot	$2,045,250	7%	$181,080	1%	
	Cold brew	$288,750	1%	$241,770	2%	
	Filter	$404,250	1%	$70,010	0%	
	Drip machine	$2,632,000	9%	$1,374,450	9%	
	AeroPress	$2,601,500	9%	$1,293,780	9%	
	Pour over	$846,000	3%	$364,530	2%	
	French press	$1,113,250	4%	$748,120	5%	
	Cezve	$2,512,500	9%	$1,969,520	13%	
	Chemex	$3,137,250	11%	$817,680	6%	
	Scale	$3,801,000	13%	$2,910,290	20%	
	Kettle	$756,250	3%	$617,520	4%	
	Espresso Machine	$8,406,000	29%	$3,636,440	25%	
	Total	**$29,448,500**	**100%**	**$14,793,150**	**100%**	

Using **Great Tables**, you work entirely in Python!

It's reproducible, less effort, and the tables look great!

Great Tables

The **Great Tables** package is focused purely on the display of tables.

Great Tables

The **Great Tables** package is focused purely on the display of tables.

It's not the only approach in their respective language, but it is:

— comprehensive
— actively-developed
— attentive to all table-related problems

Great Tables

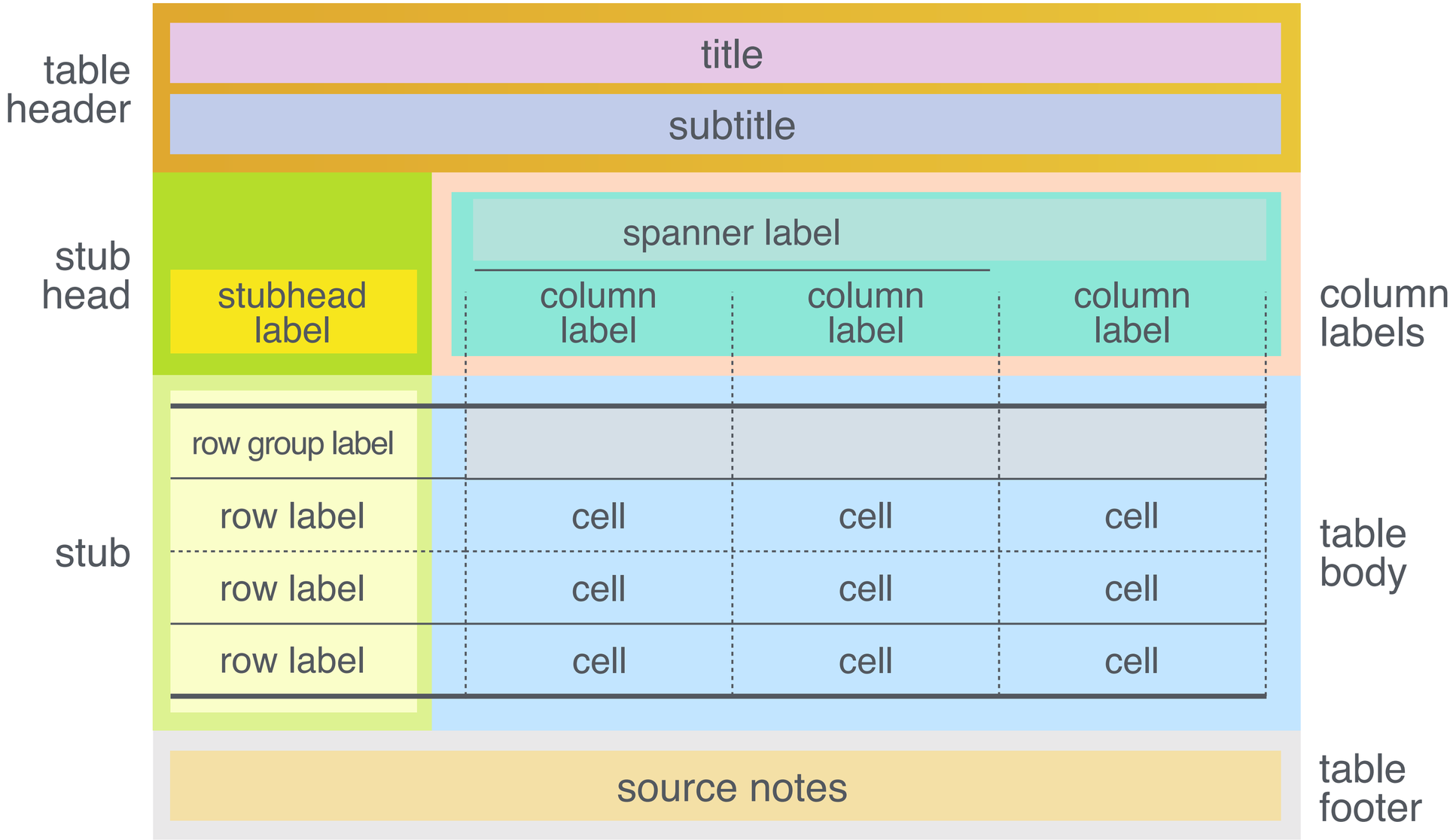
The **Great Tables** package is focused purely on the display of tables.

It's not the only approach in their respective language, but it is:

— comprehensive
— actively-developed
— attentive to all table-related problems

We put together this workshop to help you learn the process and design behind making presentation-quality tables.

Our Modern Take on a Table Display Framework



Part 1: **Coffee Table**

One example table: **Coffee Table**

Sales of Coffee Equipment				
	Revenue		Profit	
Product	Amount	Percent	Amount	Percent
Grinder	$904,500	3%	$567,960	4%
Moka pot	$2,045,250	7%	$181,080	1%
Cold brew	$288,750	1%	$241,770	2%
Filter	$404,250	1%	$70,010	0%
Drip machine	$2,632,000	9%	$1,374,450	9%
AeroPress	$2,601,500	9%	$1,293,780	9%
Pour over	$846,000	3%	$364,530	2%
French press	$1,113,250	4%	$748,120	5%
Cezve	$2,512,500	9%	$1,969,520	13%
Chemex	$3,137,250	11%	$817,680	6%
Scale	$3,801,000	13%	$2,910,290	20%
Kettle	$756,250	3%	$617,520	4%
Espresso Machine	$8,406,000	29%	$3,636,440	25%
Total	**$29,448,500**	**100%**	**$14,793,150**	**100%**

Key Ingredients: Structure, Format, Style

Sales of Coffee Equipment				
	Revenue		Profit	
Product	Amount	Percent	Amount	Percent
Grinder	$904,500	3%	$567,960	4%
Moka pot	$2,045,250	7%	$181,080	1%
Cold brew	$288,750	1%	$241,770	2%
Filter	$404,250	1%	$70,010	0%
Drip machine	$2,632,000	9%	$1,374,450	9%
AeroPress	$2,601,500	9%	$1,293,780	9%
Pour over	$846,000	3%	$364,530	2%
French press	$1,113,250	4%	$748,120	5%
Cezve	$2,512,500	9%	$1,969,520	13%
Chemex	$3,137,250	11%	$817,680	6%
Scale	$3,801,000	13%	$2,910,290	20%
Kettle	$756,250	3%	$617,520	4%
Espresso Machine	$8,406,000	29%	$3,636,440	25%
Total	**$29,448,500**	**100%**	**$14,793,150**	**100%**

Title.

Column spanners.

Nice column labels.

STRUCTURE

Key Ingredients: Structure, Format, Style

Sales of Coffee Equipment				
	Revenue		Profit	
Product	Amount	Percent	Amount	Percent
Grinder	$904,500	3%	$567,960	4%
Moka pot	$2,045,250	7%	$181,080	1%
Cold brew	$288,750	1%	$241,770	2%
Filter	$404,250	1%	$70,010	0%
Drip machine	$2,632,000	9%	$1,374,450	9%
AeroPress	$2,601,500	9%	$1,293,780	9%
Pour over	$846,000	3%	$364,530	2%
French press	$1,113,250	4%	$748,120	5%
Cezve	$2,512,500	9%	$1,969,520	13%
Chemex	$3,137,250	11%	$817,680	6%
Scale	$3,801,000	13%	$2,910,290	20%
Kettle	$756,250	3%	$617,520	4%
Espresso Machine	$8,406,000	29%	$3,636,440	25%
Total	**$29,448,500**	**100%**	**$14,793,150**	**100%**

Title.

Column spanners.

Nice column labels.

STRUCTURE

Currency values.

Percentages.

FORMAT

Key Ingredients: Structure, Format, Style

Sales of Coffee Equipment				
	Revenue		Profit	
Product	Amount	Percent	Amount	Percent
Grinder	$904,500	3%	$567,960	4%
Moka pot	$2,045,250	7%	$181,080	1%
Cold brew	$288,750	1%	$241,770	2%
Filter	$404,250	1%	$70,010	0%
Drip machine	$2,632,000	9%	$1,374,450	9%
AeroPress	$2,601,500	9%	$1,293,780	9%
Pour over	$846,000	3%	$364,530	2%
French press	$1,113,250	4%	$748,120	5%
Cezve	$2,512,500	9%	$1,969,520	13%
Chemex	$3,137,250	11%	$817,680	6%
Scale	$3,801,000	13%	$2,910,290	20%
Kettle	$756,250	3%	$617,520	4%
Espresso Machine	$8,406,000	29%	$3,636,440	25%
Total	**$29,448,500**	**100%**	**$14,793,150**	**100%**

Title.

Column spanners.

Nice column labels.

STRUCTURE

Currency values.

Percentages.

FORMAT

Fill color.

Bold text.

STYLE

Three Last Things: Nanoplots, Images, and Missing Values

After we do all these things, we'll get this table:

adding icons
`fmt_image()`

`sub_missing()`
addressing a missing val

Sales of Coffee Equipment

	Product	Revenue Amount	Revenue Percent	Profit Amount	Profit Percent	Monthly Revenue
	Grinder	$904,500	3%	$567,960	4%	
	Moka pot	$2,045,250	7%	$181,080	1%	
	Cold brew	$288,750	1%	$241,770	2%	
	Filter	$404,250	1%	$70,010	0%	
	Drip machine	$2,632,000	9%	$1,374,450	9%	
	AeroPress	$2,601,500	9%	$1,293,780	9%	
	Pour over	$846,000	3%	$364,530	2%	
	French press	$1,113,250	4%	$748,120	5%	
	Cezve	$2,512,500	9%	$1,969,520	13%	
	Chemex	$3,137,250	11%	$817,680	6%	
	Scale	$3,801,000	13%	$2,910,290	20%	
	Kettle	$756,250	3%	$617,520	4%	
	Espresso Machine	$8,406,000	29%	$3,636,440	25%	
	Total	**$29,448,500**	**100%**	**$14,793,150**	**100%**	

adding nanoplots
`fmt_nanoplot()`

Structure Basics

Structure: `GT()` — How to Begin with the GT API

We first need to introduce our data to **Great Tables**.

This could all start with either a Pandas or a Polars DataFrame.

Code

```
GT(<data>)
```

Structure: `GT()` — How to Begin with the GT API

Let's use an example dataset, `exibble`, to make a **Great Tables** table:

Code

```
from great_tables import GT, exibble

GT(exibble)
```

num	char	fctr	date	time	datetime	currency	row	group
1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	49.950	row_1	grp_a
2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	17.950	row_2	grp_a
3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	1.390	row_3	grp_a
4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	65100.000	row_4	grp_a
5.550e+03	NA	five	2015-05-15	17:55	2018-05-05 04:00	1325.810	row_5	grp_b
NA	fig	six	2015-06-15	NA	2018-06-06 16:11	13.255	row_6	grp_b
7.770e+05	grapefruit	seven	NA	19:10	2018-07-07 05:22	NA	row_7	grp_b
8.880e+06	honeydew	eight	2015-08-15	20:20	NA	0.440	row_8	grp_b

Structure: `GT()` — Making a Stub with Row Labels

See this column called `row`? It contains row labels and we could structure this table with a stub (holds row labels).

num	char	fctr	date	time	datetime	currency	row	group
1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	49.950	row_1	grp_a
2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	17.950	row_2	grp_a
3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	1.390	row_3	grp_a
4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	65100.000	row_4	grp_a
5.550e+03	NA	five	2015-05-15	17:55	2018-05-05 04:00	1325.810	row_5	grp_b
NA	fig	six	2015-06-15	NA	2018-06-06 16:11	13.255	row_6	grp_b
7.770e+05	grapefruit	seven	NA	19:10	2018-07-07 05:22	NA	row_7	grp_b
8.880e+06	honeydew	eight	2015-08-15	20:20	NA	0.440	row_8	grp_b

Structure: `GT()` — Making a Stub with Row Labels

`GT()` has the `rowname_col=` arg. Supply the column name there and it moves into the stub.

Code

```
from great_tables import GT, exibble

GT(exibble, rowname_col="row")
```

	num	char	fctr	date	time	datetime	currency	group
row_1	1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	49.950	grp_a
row_2	2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	17.950	grp_a
row_3	3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	1.390	grp_a
row_4	4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	65100.000	grp_a
row_5	5.550e+03	NA	five	2015-05-15	17:55	2018-05-05 04:00	1325.810	grp_b
row_6	NA	fig	six	2015-06-15	NA	2018-06-06 16:11	13.255	grp_b
row_7	7.770e+05	grapefruit	seven	NA	19:10	2018-07-07 05:22	NA	grp_b
row_8	8.880e+06	honeydew	eight	2015-08-15	20:20	NA	0.440	grp_b

Structure: `GT()` — Making Row Groups

See this column called `group`? It contains categories for grouping rows, so and we could add row groups to the table.

	num	char	fctr	date	time	datetime	currency	group
row_1	1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	49.950	grp_a
row_2	2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	17.950	grp_a
row_3	3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	1.390	grp_a
row_4	4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	65100.000	grp_a
row_5	5.550e+03	NA	five	2015-05-15	17:55	2018-05-05 04:00	1325.810	grp_b
row_6	NA	fig	six	2015-06-15	NA	2018-06-06 16:11	13.255	grp_b
row_7	7.770e+05	grapefruit	seven	NA	19:10	2018-07-07 05:22	NA	grp_b
row_8	8.880e+06	honeydew	eight	2015-08-15	20:20	NA	0.440	grp_b

Structure: `GT()` — Making Row Groups

Use the `groupname_col=` arg to define row groups with a column.

Code

```
GT(
  exibble,
  rowname_col="row",
  groupname_col="group"
)
```

	num	char	fctr	date	time	datetime	currency
grp_a							
row_1	1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	49.950
row_2	2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	17.950
row_3	3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	1.390
row_4	4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	65100.000
grp_b							
row_5	5.550e+03	NA	five	2015-05-15	17:55	2018-05-05 04:00	1325.810
row_6	NA	fig	six	2015-06-15	NA	2018-06-06 16:11	13.255
row_7	7.770e+05	grapefruit	seven	NA	19:10	2018-07-07 05:22	NA
row_8	8.880e+06	honeydew	eight	2015-08-15	20:20	NA	0.440

Structure: Comparing a Basic and a More Structured Table

Basic GT Table

num	char	fctr	date	time	row	group
1.111e-01	apricot	one	2015-01-15	13:35	row_1	grp_a
2.222e+00	banana	two	2015-02-15	14:40	row_2	grp_a
3.333e+01	coconut	three	2015-03-15	15:45	row_3	grp_a
4.444e+02	durian	four	2015-04-15	16:50	row_4	grp_a
5.550e+03	NA	five	2015-05-15	17:55	row_5	grp_b
NA	fig	six	2015-06-15	NA	row_6	grp_b
7.770e+05	grapefruit	seven	NA	19:10	row_7	grp_b
8.880e+06	honeydew	eight	2015-08-15	20:20	row_8	grp_b

Table w/ Stub & Row Groups

	num	char	fctr	date
grp_a				
row_1	1.111e-01	apricot	one	2015-01-15
row_2	2.222e+00	banana	two	2015-02-15
row_3	3.333e+01	coconut	three	2015-03-15
row_4	4.444e+02	durian	four	2015-04-15
grp_b				
row_5	5.550e+03	NA	five	2015-05-15
row_6	NA	fig	six	2015-06-15
row_7	7.770e+05	grapefruit	seven	NA
row_8	8.880e+06	honeydew	eight	2015-08-15

Structure: `tab_header()` — Adding a Title to Your Table

Adding a title to the GT table in a header component can be good for presentation. We do that with `tab_header()`.

Code

```
GT(exibble) \
.tab_header(title="Table Title")
```

Table Title

num	char	fctr	date	time	datetime	currency	row	group
1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	49.950	row_1	grp_a
2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	17.950	row_2	grp_a
3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	1.390	row_3	grp_a
4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	65100.000	row_4	grp_a
5.550e+03	NA	five	2015-05-15	17:55	2018-05-05 04:00	1325.810	row_5	grp_b

Structure: `tab_header()` — You Can Also Add a Subtitle

Having a subtitle is also an option here.

Code

```
GT(exibble) \
.tab_header(
  title="Table Title",
  subtitle="The Subtitle"
)
```

Table Title

The Subtitle

num	char	fctr	date	time	datetime	currency	row	group
1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	49.950	row_1	grp_a
2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	17.950	row_2	grp_a
3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	1.390	row_3	grp_a
4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	65100.000	row_4	grp_a
5.550e+03	NA	five	2015-05-15	17:55	2018-05-05 04:00	1325.810	row_5	grp_b

Structure: `tab_spanner()` — Spanners Above Column Labels

Another way to add structure is to add spanner labels above sets of column labels. We do this with `tab_spanner()`.

Code

```
GT(exibble) \
.tab_spanner(
  columns=["date", "time", "datetime"],
  label="A Spanner"
)
```

			A Spanner					
num	char	fctr	date	time	datetime	currency	row	group
1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	49.950	row_1	grp_a
2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	17.950	row_2	grp_a
3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	1.390	row_3	grp_a
4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	65100.000	row_4	grp_a

Structure: `cols_label()` — Making Column Labels Much Nicer

The column labels are derived from the column names. We usually want to make them more presentable and it's done with `cols_label()`.

Code

```
GT(exibble) \
.cols_label(
  num="Numbers",
  char="Fruits"
)
```

Numbers	Fruits	fctr	date	time	datetime	currency	row	group
1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	49.950	row_1	grp_a
2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	17.950	row_2	grp_a
3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	1.390	row_3	grp_a
4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	65100.000	row_4	grp_a

Formatting Basics

Format: the family of `fmt_*()` Methods

There is a huge number of formatting methods!

They all begin with `fmt_` and they format values in the table body.

They they typically operate on whole columns of data but
you can subset the columns' rows.

Here's a sampling of what is available:

```
fmt_number()
fmt_integer()
fmt_scientific()
fmt_engineering()
fmt_percent()
fmt_currency()
```

```
fmt_date()
fmt_time()
fmt_datetime()
fmt_markdown()
fmt_image()
fmt()
```

+ many more.

Format: `fmt_currency()` — Formatting Monetary Values

Let's again use the example dataset, **`exibble`**, and see what we're starting with.

Code

```
from great_tables import GT, exibble

GT(exibble)
```

num	char	fctr	date	time	datetime	currency	row	group
1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	49.950	row_1	grp_a
2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	17.950	row_2	grp_a
3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	1.390	row_3	grp_a
4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	65100.000	row_4	grp_a
5.550e+03	NA	five	2015-05-15	17:55	2018-05-05 04:00	1325.810	row_5	grp_b
NA	fig	six	2015-06-15	NA	2018-06-06 16:11	13.255	row_6	grp_b
7.770e+05	grapefruit	seven	NA	19:10	2018-07-07 05:22	NA	row_7	grp_b
8.880e+06	honeydew	eight	2015-08-15	20:20	NA	0.440	row_8	grp_b

Format: `fmt_currency()` — Formatting Monetary Values

There's a column called `currency` here. Let's format that with `fmt_currency()`

num	char	fctr	date	time	datetime	currency	row	group
1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	49.950	row_1	grp_a
2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	17.950	row_2	grp_a
3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	1.390	row_3	grp_a
4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	65100.000	row_4	grp_a
5.550e+03	NA	five	2015-05-15	17:55	2018-05-05 04:00	1325.810	row_5	grp_b
NA	fig	six	2015-06-15	NA	2018-06-06 16:11	13.255	row_6	grp_b
7.770e+05	grapefruit	seven	NA	19:10	2018-07-07 05:22	NA	row_7	grp_b
8.880e+06	honeydew	eight	2015-08-15	20:20	NA	0.440	row_8	grp_b

Format: `fmt_currency()` — Formatting Monetary Values

We've got to specify the columns here when using any formatter. In this case it is the one column called `currency`.

Code

```
GT(exibble) \
.fmt_currency(columns="currency")
```

num	char	fctr	date	time	datetime	currency	row	group
1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	$49.95	row_1	grp_a
2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	$17.95	row_2	grp_a
3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	$1.39	row_3	grp_a
4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	$65,100.00	row_4	grp_a
5.550e+03	NA	five	2015-05-15	17:55	2018-05-05 04:00	$1,325.81	row_5	grp_b
NA	fig	six	2015-06-15	NA	2018-06-06 16:11	$13.26	row_6	grp_b
7.770e+05	grapefruit	seven	NA	19:10	2018-07-07 05:22	NA	row_7	grp_b
8.880e+06	honeydew	eight	2015-08-15	20:20	NA	$0.44	row_8	grp_b

Format: `fmt_currency()` — Formatting Monetary Values

Before Formatting

currency
49.950
17.950
1.390
65100.000
1325.810
13.255
NA
0.440

After Formatting

currency
$49.95
$17.95
$1.39
$65,100.00
$1,325.81
$13.26
NA
$0.44

What changes can you see in the *before* and *after* of this formatting?

Format: `fmt_percent()` — Formatting as Percentages

There's not really a great column to demonstrate percentage formatting in `exibble`.
Let's make up a new table with five rows:

row	value
1	0.0200
2	0.4345
3	0.0520
4	0.7530
5	1.0234

Format: `fmt_percent()` — Formatting as Percentages

We will format that called **`value`** column with **`fmt_percent()`**. Let's use the defaults.

Code

```
GT(<data>) \
.fmt_percent(columns="value")
```

row	value
1	2.00%
2	43.45%
3	5.20%
4	75.30%
5	102.34%

Format: `fmt_percent()` — Formatting as Percentages

Before Formatting

row	value
1	0.0200
2	0.4345
3	0.0520
4	0.7530
5	1.0234

After Formatting

row	value
1	2.00%
2	43.45%
3	5.20%
4	75.30%
5	102.34%

What changes can you see in the *before* and *after* of this formatting?

Styling with `tab_style()`

Style: `tab_style()` — Styling the Table Cells

The `tab_style()` method is a bit more complicated than the previous ones we tried out.

Code

```
from great_tables import GT

GT(<data>) \
.tab_style(
  style=...,
  locations=...
)
```

The additional complication is in the `...` parts. They require the use of helper functions.

Style: `tab_style()` — Styling the Table Cells

There are two arguments in `tab_style()`: `style` and `locations`.

Code

```
from great_tables import GT

GT(<data>) \
.tab_style(
  style=...,
  locations=...
)
```

what

where

Guide to args:

`style`: what is the styling we are going to use?

`locations`: where is the styling going to be used? Or, which cells receive the styles?

Style: `tab_style()` — Styling the Table Cells

Helpers used to define the `style` and `locations`.

`style` use `style` class

- `style.fill()`
- `style.text()`
- `style.borders()`

`locations` use `loc` class

- `loc.body()`
- `loc.stub()`
- `loc.column_labels()`
- `...`

Before Styling

num	char
1.111e-01	apricot
2.222e+00	banana
3.333e+01	coconut
4.444e+02	durian
5.550e+03	NA
NA	fig
7.770e+05	grapefruit
8.880e+06	honeydew

After Styling ×2

num	char
1.111e-01	apricot
2.222e+00	banana
3.333e+01	coconut
4.444e+02	durian
5.550e+03	NA
NA	fig
7.770e+05	grapefruit
8.880e+06	honeydew

Style: `tab_style()` — Styling the Table Cells

IMPORTANT: we need two extra imports if using `tab_style()`.

Code

```
from great_tables import GT, style, loc

GT(<data>) \
.tab_style(
  style = style...,
  locations = loc...
)
```

Importing `style` and `loc` is important here. Otherwise, you can't use `tab_style()`.

Style: `tab_style()` — Styling the Table Cells

We will go back to the baseline table (using **`exibble`** for this).

Code

```
GT(exibble)
```

num	char	fctr	date	time	datetime	currency	row	group
1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	49.950	row_1	grp_a
2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	17.950	row_2	grp_a
3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	1.390	row_3	grp_a
4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	65100.000	row_4	grp_a
5.550e+03	NA	five	2015-05-15	17:55	2018-05-05 04:00	1325.810	row_5	grp_b
NA	fig	six	2015-06-15	NA	2018-06-06 16:11	13.255	row_6	grp_b
7.770e+05	grapefruit	seven	NA	19:10	2018-07-07 05:22	NA	row_7	grp_b
8.880e+06	honeydew	eight	2015-08-15	20:20	NA	0.440	row_8	grp_b

Style: `tab_style()` — Styling the Table Cells

Let's style the entire **`num`** column with a light blue background color.

Code

```
GT(exibble) \
.tab_style(
  style=style.fill(color="lightblue"),
  locations=loc.body(columns="num")
)
```

num	char	fctr	date	time	datetime	currency	row	group
1.111e-01	apricot	one	2015-01-15	13:35	2018-01-01 02:22	49.950	row_1	grp_a
2.222e+00	banana	two	2015-02-15	14:40	2018-02-02 14:33	17.950	row_2	grp_a
3.333e+01	coconut	three	2015-03-15	15:45	2018-03-03 03:44	1.390	row_3	grp_a
4.444e+02	durian	four	2015-04-15	16:50	2018-04-04 15:55	65100.000	row_4	grp_a
5.550e+03	NA	five	2015-05-15	17:55	2018-05-05 04:00	1325.810	row_5	grp_b
NA	fig	six	2015-06-15	NA	2018-06-06 16:11	13.255	row_6	grp_b
7.770e+05	grapefruit	seven	NA	19:10	2018-07-07 05:22	NA	row_7	grp_b
8.880e+06	honeydew	eight	2015-08-15	20:20	NA	0.440	row_8	grp_b