

VarTable

VarTable is a package to make variation table, in a simple way
This package is build on top of [Cetz](#)
(version: 0.1.0)

Contents

1 - Introduction	1
2 - tabvar function	3
2.1 - general description	3
tabvar	3
Parameters	3
variable	3
label	3
domain	4
contents	4
table-style	4
nocadre	4
arrow-mark	4
arrow-style	4
line-0	5
line-style	5
hatching-style	5
first-column-width	5
first-line-height	5
element-distance	5
values	6
add	6
2.2 - The contents parameter	7
2.2.1 - Sign rows format	7
2.2.1.1 - A classical sign array	7
2.2.1.2 - Custom separation bar	8
2.2.1.2.1 style of bar	8
2.2.1.2.2 type of bar	8
2.2.1.3 - Same sign for more than one value of the variable	9
2.2.2 - Variation table	10
2.2.2.1 - A classical variation array	10
2.2.2.2 - Undefined values	10
2.2.2.3 - Skip a value	12
3 - More complex examples	13
3.1 - Γ function on $[0; +\infty]$	13
3.2 - A Rational function	14
3.3 Hyperbolic function	15
3.3 A weird table for a simple polynomial function	16

1 - Introduction

This package is designed to simplify the creation of variation tables for functions. To do this, it gives

you a typst function, whose parameters are described in detail in this documentation.

A word of warning: it's quite normal that during the array creation process, The elements, such as the lines between the various elements, aren't created as they should be. For example, the line between the labels and the rest, which doesn't go all the way to the end.

If you encounter any bugs, please report them on my [GitHub](#).

2 - tabvar function

2.1 - general description

tabvar

Return a variation table

Parameters

```
tabvar(  
  variable: string,  
  label: string,  
  domain: array,  
  contents: array,  
  table-style: style,  
  nocadre,  
  arrow-mark: mark,  
  arrow-style: style,  
  line-0: bool,  
  line-style: style,  
  hatching-style: style,  
  first-column-width: length,  
  first-line-height: length,  
  element-distance: length,  
  values: array,  
  add: content  
)
```

variable string

variable is a content block which contains the table's variable name (like x or t)

Default: x

label string

label is an array which contains arrays of 2 arguments that contains in first position the line's label and in second position, if the line is a variation table or a sign table with this following keys : "v" for variation and "s" for sign

Example: for a variation table of a function f , you should write:

```
init: (  
  variable:  $x$ ,  
  label: (  
    ([sign of  $f$ ], "Sign"), // the first line is a sign table  
    ([variation of  $f$ ], "Variation") // the second line is a variation table  
  )  
)
```

Default: ()

domain array

values taken by the variable

for example if your function changes sign or reaches a max/min for $x \in \{0, 1, 2, 3\}$

you should write this :

domain: ($\$0$, $\$1$, $\$2$, $\$3$)

Default: ()

contents array

the content of the table

see below for more details

Default: ((),)

table-style style

Optional

The style of the table,

the style type is defined by Cetz, so I invite you to have a look at the [Cetz manual](#).

Caution : if you haven't entered the mark symbol as none, all lines in the table will have an arrowhead.

Default: (stroke: 1pt + black, mark: (symbol: none))

nocadre

to hide the external cadre

Default: false

arrow-mark mark

the style of the arrowhead, the type of which is defined by Cetz

Default: (end: "straight")

arrow-style style

Optional

the style of the arrow, as for the table-style parameter

Caution : the mark section is overwritten by the arrow-mark

Default: (stroke: black + 1pt)

line-0 `bool`

Optional

if you want to change the default bar sign to a bar with a 0

Default: `false`

line-style `style`

Optional

if you want to change the style of all separator lines between signs

Warning: this will only change the default lines, the `||`, `|` or `0` lines will not be changed.

Default: `(stroke: black + 1pt)`

hatching-style `style`

Optional

The style of the hatching in the sign and variation table

Default: `tiling(size: (30pt, 30pt))`

```
#place(line(start: (0%, 100%), end: (100%, 0%), stroke: 2pt))
#place(line(start: (-100%, 100%), end: (100%, -100%), stroke: 2pt))
#place(line(start: (0%, 200%), end: (200%, 0%), stroke: 2pt))
]
```

first-column-width `length`

Optional

change the width of the first column

Default: `none`

first-line-height `length`

Optional

change the height of the first line

Default: `none`

element-distance `length`

Optional

change the distance between two elements

Default: `none`

values array

Optional

To add values between to pre-defined values

Default: ((),)

add content

Optional

To add more stuff with Cetz

Default: ()

2.2 - The contents parameter

The contents parameter must be an array with one element per line (per label)

Each element is itself an array with one element per column, with a different format for either sign or variation rows

2.2.1 - Sign rows format

Should contain as much element as the domain less one (one per interval) + one optional end bar style element

Each element is in either of these form (can be mixed on a same line):

() – Empty : extend previous cell

body – Simple body such as $++$ or $--$

(body, bar_style) – to specify an optional style for the **previous** bar, with one of "|" (simple bar), "||" (double bar) or "0" (bar with a zero)

NB: the line-0 parameter change the default bar style to "|" "

The optional last element is "||"

2.2.1.1 - A classical sign array

A sign array must contain contents like $++$ or $--$, but you can put anything else.

Example:

A normal sign table:

```
#tabvar(  
  init: (  
    variable: $t$,  
    label: ([sign], "Sign"),  
  ),  
  domain: ($2$, $4$, $6$, $8$),  
  contents: (($+$, $-$, $+$),)  
)
```

t	2	4	6	8
sign	+	–	+	

More complex usage:

```
#tabvar(  
  init: (  
    variable: $t$,  
    label: ([sign], "Sign"),  
  ),  
  domain: ($2$, $4$, $6$, $8$),  
  contents: (  
    (  
      "Hello world !",  
      $-$,  
      $ 3 / 2 $  
    )  
  ),  
)
```

t	2	4	6	8
sign	Hello world!	–	$\frac{3}{2}$	

Note: on the second example the table is squeezed with the scale function)

2.2.1.2 - Custom separation bar

2.2.1.2.1 style of bar

you can modify the style of the bars (note that this modifies all the default ones, not the others, see 2.2.1.2.2).

the style of the bar is a dictionary, of the type "style" defined by Cetz as said earlier.

To make it simple if you want to modify only the stroke of the bar, you just have to put `stroke`: your `stroke` in brackets.

For more complex applications, please refer to the Cetz manual.

Example

```
#tabvar(  
  line-style: (  
    stroke: (paint: red, dash: "dashed")  
  ),  
  init: (  
    variable: $t$,  
    label: ([sign], "Sign"),  
  ),  
  domain: ($2$, $4$, $6$,),  
  contents: (  
    ($+$, $-$),  
  ),  
)
```

t	2	4	6
sign	+	—	—

2.2.1.2.2 type of bar

For all signs except the first one, instead of putting the sign directly, you can put a couple, whose first element defines the previous bar's type.

There are 3 different types of bar:

- "`|`" : a simple bar
- "`0`" : a bar with a 0 on the center
- "`||`" : a double bar, like for the undefined values

NB : the `marks-line` parameter has no effect on these bars

Example:

```
#tabvar(  
  init: (  
    variable: $t$,  
    label: ([sign], "Sign"),  
  ),  
  domain: ($2$, $4$, $6$, $8$, $10$),  
  contents: (  
    (  
      $+$,  
      ("|", $-$),  
      ("0", $-$),  
      ("||", $+$)  
    ),  
  ),  
)
```

t	2	4	6	8	10
sign	+	—	0	—	+

Note: The `line-0` parameter is to default lines to "`0`" type or "`|`" type

If you want a double line at the start of the table, you can use a double bar "||" on the very first sign. If you want it at the end, you can add this element "||" at the end of sign array

Example:

```
#tabvar(
  init: (
    variable: $t$,
    label: ([sign], "Sign"),
  ),
  domain: ($2$, $4$, $6$, $8$),
  contents: (
    (
      ("||", $+$),
      $-$,
      "||"
    ),
  ),
)
```

t	2	4	6
sign	+	—	

2.2.1.3 - Same sign for more than one value of the variable

For this, it is pretty easy, instead of putting the sign directly, you can put an empty couple

Example:

```
#tabvar(
  line-0: true,
  init: (
    variable: $t$,
    label: (
      [sign], "Sign",
    ),
  ),
  domain: ($2$, $4$, $6$, $8$),
  contents: (
    ($+$, (), $-$),
  ),
)
```

t	2	4	6	8
sign	+		0	—

2.2.2 - Variation table

Should contains as much elements as the domain

Each element is in etheir of these forms :

- () to extend the previous arrow
- (position,body) with position being one of top, center or bottom
- (pos1, pos2, "||", body1, body2) to put in 2 value separated by an undefined value (double bar)
- (pos, "||", body) shorthand for (pos, pos, "||", body, body) (see previous format)

2.2.2.1 - A classical variation array

A variation array must contain couple with in first position, the element position, and in second position, whatever you want as long as it's of the contents type.

The position can be top, center or bottom, but no other type of alignment

Example :

```
#tabvar(
  init: (
    variable: $t$,
    label: ([[variation], "Variation"],),
  ),
  domain: ($2$, $4$, $6$, $8$),
  contents: (
    (
      (top, $3$),
      (bottom, $0$),
      (center, $1$),
      (top, $4$),
    ),
  ),
)
```

t	2	4	6	8
variation	3		1	4

2.2.2.2 - Undefined values

If your function is not defined on some values like $f(x) = \frac{1}{x}$ for $x = 0$, you certainly want to put a double line meaning that the function is undefined on this value, and you can !

★ For each values of domain except the start and the end.



The array of one value should look like (pos1, pos2, "||", contents1, contents2)

where :

- pos1 and 2 is top, center, bottom and pos1 is for the placement of contents1 similary for pos2
- "||" is to precise the value is undefine
- contents1 and 2 is type of contents and contents1 one is for before the double bar and contents2 for after

Example :

```
#tabvar(
  init: (
    variable: $t$,
    label: ([variation], "Variation"),),
),
domain: ($2$, $4$, $6$, $8$),
contents: (
  (
    (top, $3$),
    (bottom, top, "||", $0$, $2$),
    (bottom, $1$),
  ),
),
)
```

t	2	4	6
variation	3  0	2  1	

If pos1 and pos2 is same, you can just fill in one instead of two,
In the same way if contents1 and contents2 is same, you can also enter just one

Example:

Instead of (top, top, "||" , \$0\$, \$0\$) you can use (top, "||" , \$0\$)

```
#tabvar(
  init: (
    variable: $t$,
    label: ([variation], "Variation"),),
),
domain: ($ 2 $, $4$, $6$, $8$),
contents: (
  (
    (top, $3$),
    (bottom, "||", $0$, $1$),
    (top, center, "||", $2$),
    (top, "||", $3$),
    (bottom, $1$),
  ),
),
)
```

t	2	4	6	8	9
variation	3 ↘ 0	1 ↗ 2	2 ↗ 3	3 ↗ 3	1 ↘ 1

★ For the first and the end values

It a basic array but with "||" this parameter at the array's center

For example (top, "||", \$3\$)

```

#tabvar(
  init: (
    variable: $t$,
    label: ([variation], "Variation"),),
),
domain: ($2$, $4$, $6$, $8$),
contents: (
  (
    (top, "||", $3$),
    (bottom, $1$),
    (top, "||", $2$),
  ),
),
)

```

t	2	4	6
variation	3	1	2

2.2.2.3 - Skip a value

When you want to use several functions in the same table, you will probably want to skip some values, to do this, as with sign arrays, you must create an empty array

Example:

```

#tabvar(
  arrow-mark: (end: ">", stroke: red),
  init: (
    variable: $t$,
    label: ([variation], "Variation"),),
),
domain: ($2$, $4$, $6$),
contents: (
  (
    (top, "||", $3$),
    (),
    (bottom, $2$),
  ),
),
)

```

t	2	4	6
variation	3		2

3 - More complex examples

Here is a little bundle of what you can do with the package.

3.1 - Γ function on $[0; +\infty]$

Where it takes a minimum on $[0; +\infty[$ for $x = \alpha$

```
#tabvar(
  init: (
    variable: $t$,
    label: (
      ([sign of #sym.Gamma], "Sign"),
      ([variation of #sym.Gamma], "v"),
    ),
  ),
  domain: ($0$, $ \alpha $, $ +\infty $),
  contents: (
    ($-$, $+$),
    (
      (top, "||", $+\infty$),
      (bottom, $Gamma(\alpha)$),
      (top, $+\infty$),
    ),
  ),
)
```

t	0	α	$+\infty$
sign of Γ'	-		+
variation of Γ	$+\infty \searrow \Gamma(\alpha) \nearrow +\infty$		

Take $f(x) = \frac{4x^2+12x+29}{4(x^2+3x+2)}$

And finally, we get:

```
#tabvar(
  init: (
    variable: $t$,
    label: (
      ([sign of $f'$], "Sign"),
      ([variation of $f$], "Variation"),
    ),
  ),
  domain: ($ -\infty $, $ -2 $, $ -3 / 2 $, $ -1 $, $ +\infty $),
  contents: (
    ($+$, ("||", $+$), $-$, ("||", $-$)),
    (
      (bottom, $1$),
      (top, bottom, "||", $+\infty$, $-\infty$),
      (top, $-20$),
      (bottom, top, "||", $-\infty$, $+\infty$),
      (bottom, $1$),
    ),
  ),
)
```

- Result:

t	$-\infty$	-2	$-\frac{3}{2}$	-1	$+\infty$
sign of f'	$+$	$+$	$-$	$-$	
variation of f	\nearrow $1 \rightarrow 4$	\nearrow $1 \rightarrow -20$	\searrow $-20 \rightarrow -\infty$	\searrow $+\infty \rightarrow 1$	

3.3 Hyperbolic function

Code:

```
#tabvar(
  arrow: "|-harpoon",
  stroke-arrow:
gradient.linear(..

```

Result:

t	$-\infty$	0	$+\infty$
sign of cosh	$-$	$+$	
variation of cosh	$+\infty$	1	$+\infty$
sign of sinh and tanh		$+$	
variation of sinh	$-\infty$		$+\infty$
variation of tanh		1	$-$

3.3 A weird table for a simple polynomial function

Take $g(t) = t^2 - t^3$

So, we have $g'(t) = 2t - 3t^2$

And has local extrema for $x = 0$ and $x = \frac{2}{3}$

Code:

```
#tabvar(

stroke: 5pt + red,
arrow: "X-*-<>",
arrow-style: (stroke :purple + 1.4pt),
marks-line: "<->",
init: (
  variable: $t$,
  label: (
    ([sign of $g'$], "Sign"),
    ([variation of $g$], "Variation"),
  ),
),
domain: ($ -\infty $, $ 0 $, $ 2 / 3 $, $ +\infty $),
contents: (
  ($-$, ("|", $+$), $-$),
  (
    (top, $+\infty$),
    (bottom, $0$),
    (center, $ 4 / 27 $),
    (bottom, $-\infty$),
  ),
),
),
)
```

Result:

t	$-\infty$	0	$\frac{2}{3}$	$+\infty$
sign of g'	$-$	$+$	0	$-$
variation of g	$+\infty$	0	$\frac{4}{27}$	$-\infty$