

VarTable

VarTable est un paquet pour rendre la réalisation
des tableaux de signe plus simple

Ce paquet est construit sur [Cetz](#)

(version: 0.2.1)

Table des Matières

1 - Introduction	2
2 - Tabvar	3
2.1 - description générale	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 Le paramètre de contenus	7

1 - Introduction

Ce paquet a été réalisé pour rendre la création de tableau de signe plus simple. Pour cela, ce paquet fournis la fonction « `tabvar` », dont les arguments sont décrit dans cette documentation.

Si vous rencontrez un bug, merci de me prévenir via mon [GitHub](#).

2 - Tabvar

2.1 - description générale

tabvar

Retourne un tableau de variation

Parameters

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

variable content

variable est la variable qui contient la variable du tableau (comme x ou t)

Exemple: si la variable de la fonction est t , alors:

variable : \$ t \$

Default: \$ x \$

label array

label est un array qui contient des array de longueur 2, une pour chaque ligne du tableau, dont le premier élément est le titre de la ligne et le second est le type de la ligne: signe (s) ou variation (v)

Exemple: pour le tableau de variation de la fonction f , vous devriez écrire:

```
label : (  
  ([Signe de $f$], "s"), // la première ligne est un tableau de signe  
  ([Variation de $f$], "v") // la seconde ligne est un tableaux de variation  
)
```

Default: ()

domain array

les valeurs prises par la variable

par exemple, si votre fonction change de signe ou atteint un extremum pour $x \in \{0, 1, 2, 3\}$ vous devriez écrire:

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

Default: ()

contents array

le contenu de la table

voir 2.2 pour plus de détaille

Default: ((),)

table-style style

Optionelle

Le style de la table

le type style est définis par Cetz, ainsi je vous recommande de vous référer au [manuelle de Cetz](#).

Attention : Si vous ne mettez pas le paramètre de style: mark a none, alors toute les lignes du tableau aurons une tête en flèche

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

nocadre bool

Pour cacher le cadre externe du tableau

Default: false

arrow-mark mark

Le style de la tête de flèche.

N.B. le type mark est définis par Cetz

Default: (end: "straight")

arrow-style style

Optionelle :

Le style des flèches.

Attention : le paramètre mark est supplenté par le paramètre arrow-mark

Default: (stroke: black + 1pt)

line-0 `bool`

Optionelle

si vous voulez changer la bar par défaut dans les tableaux de signe, pour une bar avec un zéro en son centre

Default: `false`

line-style `style`

Optionelle

Si vous voulez le style de toutes les bars de séparation entre les signes

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

hatching-style `tiling`

Optionelle

le style des hachures s'il y a des zones hachurées

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`

Optionelle

change la largeur de la première colonne

Default: `none`

first-line-height `length`

change la hauteur de la première ligne (celle du domaine et de la variable)

Default: `none`

element-distance `length`

Optionelle

change la distance entre deux éléments

Default: `none`

values `array`

pour ajouter des valeurs entre deux valeurs prè-définis

Default: `((,)`

add `content`

Pour ajouter plus d'éléments via Cetz

Default: `()`

2.2 Le paramètre de contenus