Types of relation between a variable and its probabilities

By Benoit Bruneau

Package: ‘bmisc’

Version: 0.2-12

Depends: car, lattice, zoo, robustbase, and methods

Author & Maintainer: Benoit Bruneau (benoit.bruneau1@gmail.com)

Description: These functions can be used to estimate probabilities `[0,1]` by specifying the inflection points of a relation. Described relations are of type ‘const’, ‘full’, ‘ramp’ and ‘logistic’.

Contents

1 Types ‘const’, ‘full’, and ‘plat.full’

These relations are the simplest that can be used. While ‘const’ stands for a constant probability of one for all values of `x` (Figure 1), the other two have ”all-or-nothing” types of probabilities. One or two thresholds (inflection points) need to be defined for types ‘full’ and ‘plat.full’. The main difference between ‘full’ (Figure 2 & 3) and ‘plat.full’ (Figure 4, 5 & 6) types are the number of thresholds. For all types, ‘plat’ stands for ”plateau”.

const.sel `(x)`

full.sel `(x, infl1, neg=FALSE, lv=0, uv=1)`

plat.full.sel `(x, infl1, infl2, neg=FALSE, lv=c(0,0), uv=c(1,1))`

where `infl1` and `infl2` are the inflection points, `x` is a numeric vector for which probabilities are estimated, `neg` indicates if the trend is negative (`TRUE`) or positive (`FALSE`), `lv` defines the lower probability values of the relation, and `uv` defines the upper probability values of the relation. By default, all fonctions have `neg=FALSE`, `lv=c(0,0)`, and `uv=c(1,1)`.

Here is an example for ‘const’ type:

```
> data = 0:3000
> const.sel(x = data)
```

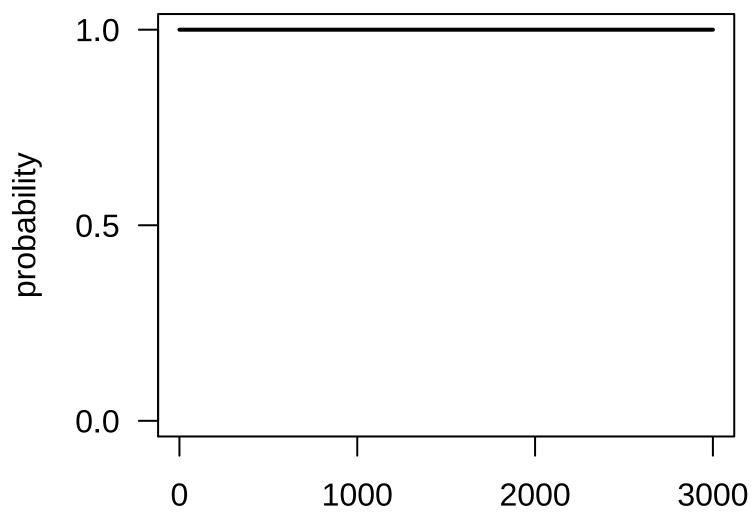

Figure 1: Type ‘const’ probabilities.

Here are examples for 'full' type:

```
> data = 0:3000
> full.sel(x = data, infl1 = 1500, neg = FALSE)
> full.sel(x = data, infl1 = 1500, neg = TRUE)
```

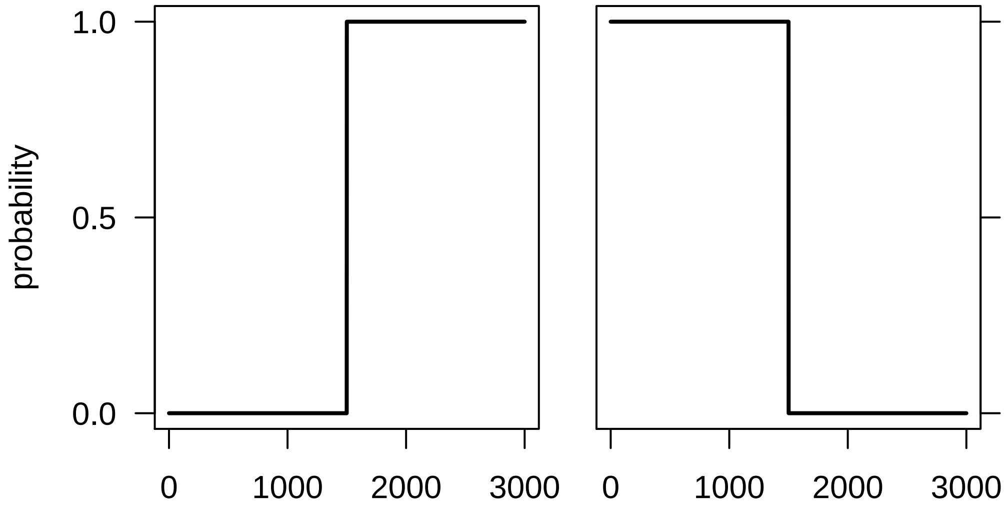

Figure 2: Type 'full' probabilities (left -> neg=FALSE | right -> neg=TRUE).

```
> data = 0:3000
> full.sel(x = data, infl1 = 1500, neg = FALSE, lv = 0.2, uv = 0.8)
> full.sel(x = data, infl1 = 1500, neg = TRUE, lv = 0.2, uv = 0.8)
```

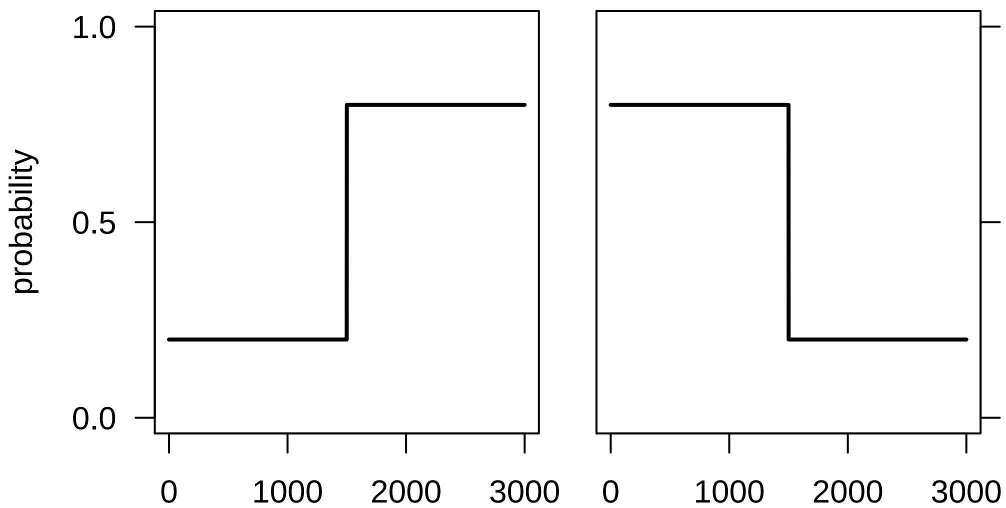

Figure 3: Type 'full' probabilities (left -> neg=FALSE | right -> neg=TRUE). In this example, minimum and maximum probabilities are respectively lv=0.2 and uv=0.8.

Here are examples for 'plat.full' type:

```
> data = 0:3000
> plat.full.sel(x = data, infl1 = 1000, infl2 = 2000, neg = FALSE)
> plat.full.sel(x = data, infl1 = 1000, infl2 = 2000, neg = TRUE)
```

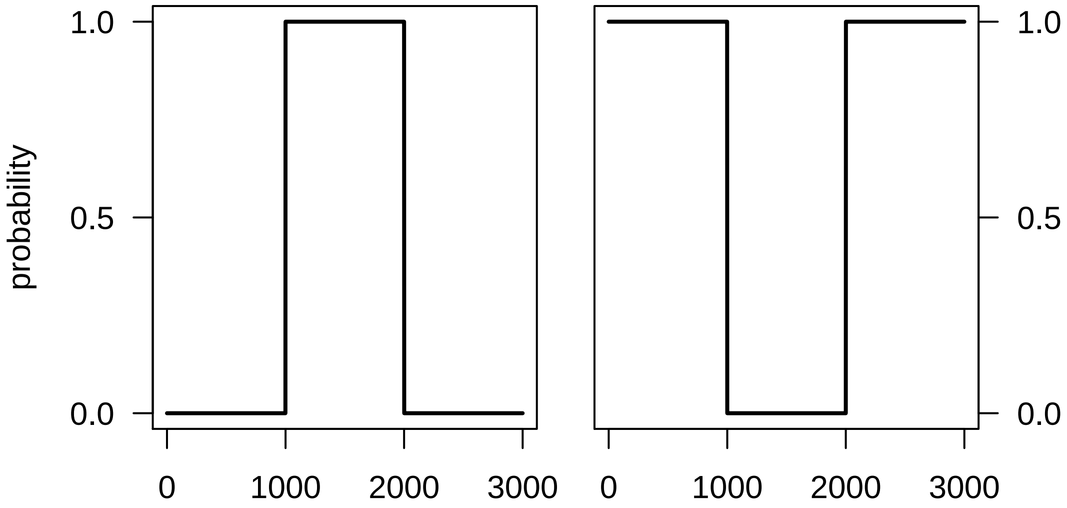

Figure 4: Type 'plat.full' probabilities (left -> neg=FALSE | right -> neg=TRUE).

```
> data = 0:3000
> plat.full.sel(x = data, infl1 = 1000, infl2 = 2000, neg = FALSE,
+     lv = 0.2, uv = 0.8)
> plat.full.sel(x = data, infl1 = 1000, infl2 = 2000, neg = TRUE,
+     lv = 0.2, uv = 0.8)
```

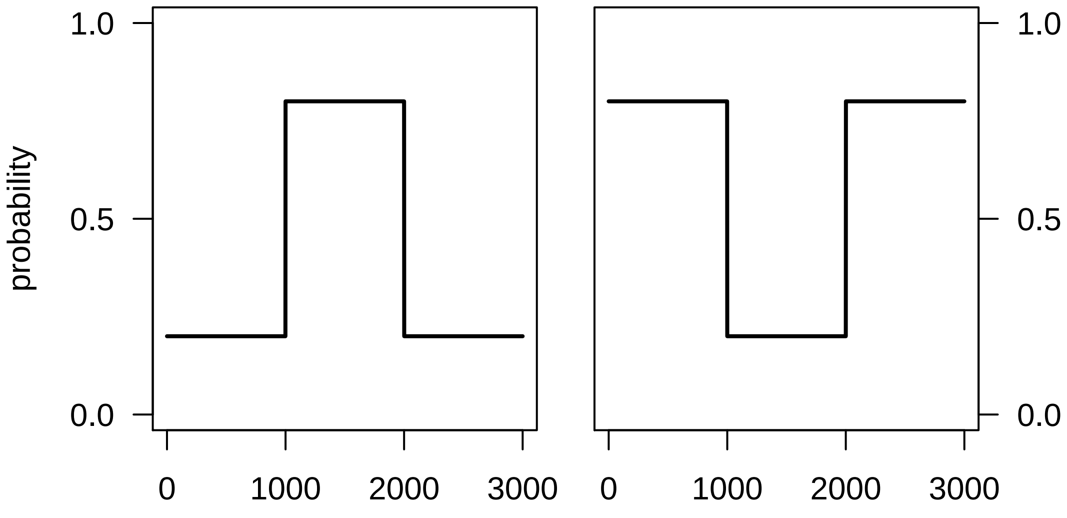

Figure 5: Type 'plat.full' probabilities (left -> neg=FALSE | right -> neg=TRUE). In this example, minimum and maximum probabilities are respectively lv=0.2 and uv=0.8.

```
> data = 0:3000
> plat.full.sel(x = data, infl1 = 1000, infl2 = 2000, neg = FALSE,
+     lv = c(0.2, 0.4), uv = 0.8)
> plat.full.sel(x = data, infl1 = 1000, infl2 = 2000, neg = TRUE,
+     lv = 0.2, uv = c(0.8, 0.6))
```

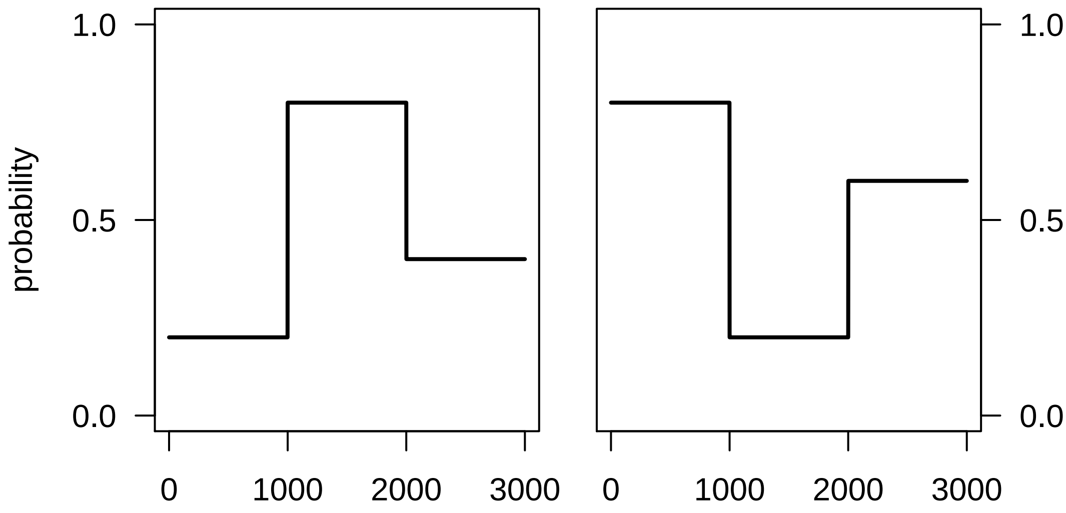

Figure 6: Type 'plat.full' probabilities (left -> neg=FALSE, lv=c(0.2,0.4) , uv=0.8 | right -> neg=TRUE, lv=0.2, uv=c(0.8,0.6)).

2 Types ‘ramp’ and ‘plat.ramp’

These relations involve adding a gradual increase (or decrease) of probabitily between two inflection points. They are an ’upgraded’ version of ‘full’ and ‘plat.full’. Two or four inflection points are needed. The main difference between ‘ramp’(Figure 7 & 8) and ‘plat.ramp’ (Figure 9, 10 & 11) types are the number inflection points.

ramp.sel `(x, infl1, infl2, neg=FALSE, lv=0, uv=1)`

plat.ramp.sel `(x, infl1, infl2, infl3, infl4, neg=FALSE, lv=c(0,0), uv=c(1,1))`

where `infl1` to `infl4` are the inflection points, `x` is a numeric vector for which probabilities are estimated, `neg` indicates if the trend is negative (`TRUE`) or positive (`FALSE`), `lv` defines the lower probability values of the relation, and `uv` defines the upper probability values of the relation. By default, all fonctions have `neg=FALSE`, `lv=c(0,0)`, and `uv=c(1,1)`.

Here are examples for ‘ramp’ type:

```
> ramp.sel(x = data, infl1 = 1000, infl2 = 2000, neg = FALSE)
> ramp.sel(x = data, infl1 = 1000, infl2 = 2000, neg = TRUE)
```

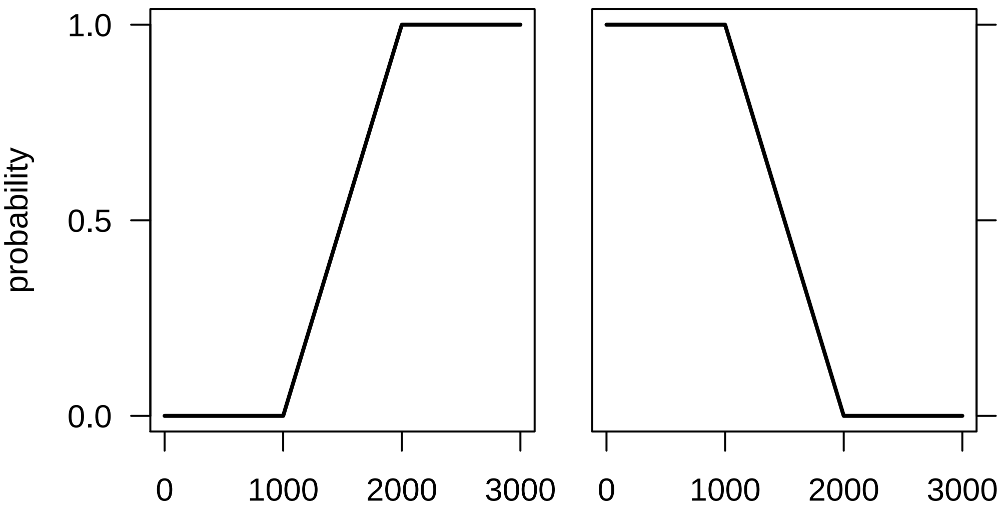

Figure 7: Type ’ramp’ probabilities (left -> neg=FALSE | right -> neg=TRUE).

```
> ramp.sel(x = data, infl1 = 1000, infl2 = 2000, neg = FALSE, lv = 0.2,
+     uv = 0.8)
> ramp.sel(x = data, infl1 = 1000, infl2 = 2000, neg = TRUE, lv = 0.2,
+     uv = 0.8)
```

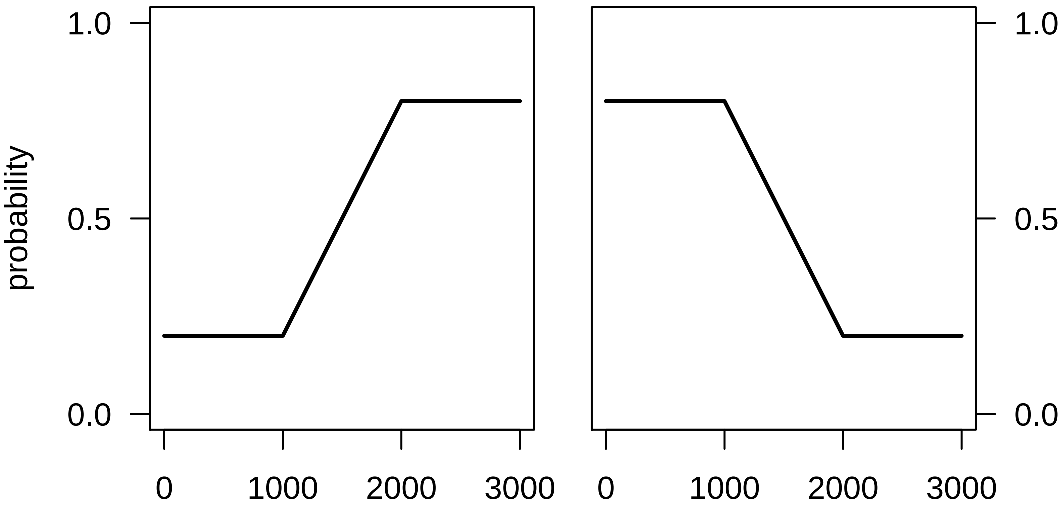

Figure 8: Type 'ramp' probabilities (left -> neg=FALSE | right -> neg=TRUE). In this example, minimum and maximum probabilities are respectively lv=0.2 and uv=0.8.

Here are examples for ‘plat.ramp’ type:

```
> data = 0:3000
> plat.ramp.sel(x = data, infl1 = 500, infl2 = 1000, infl3 = 2000,
+      infl4 = 2500, neg = FALSE)
> plat.ramp.sel(x = data, infl1 = 500, infl2 = 1000, infl3 = 2000,
+      infl4 = 2500, neg = TRUE)
```

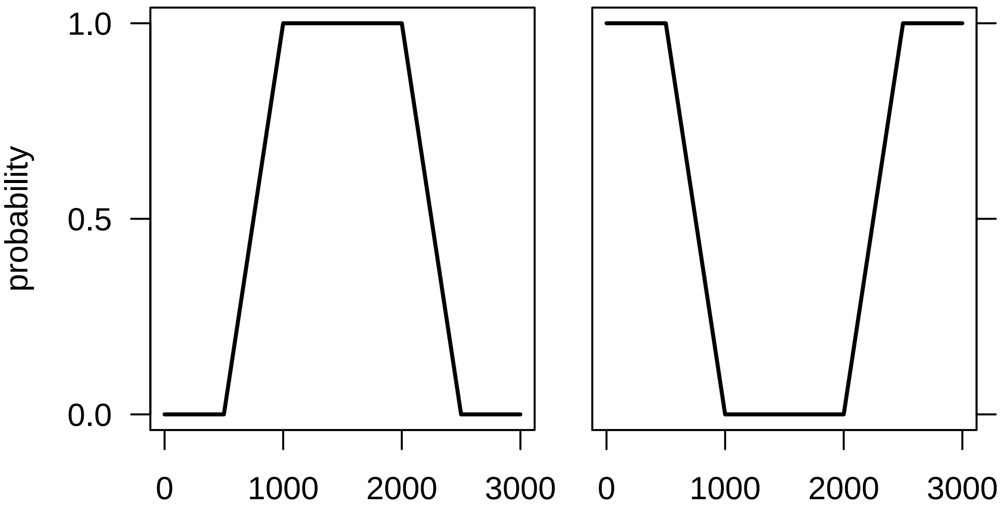

Figure 9: Type ’plat.ramp’ probabilities (left -> neg=FALSE | right -> neg=TRUE).

```
> data = 0:3000
> plat.ramp.sel(x = data, infl1 = 500, infl2 = 1000, infl3 = 2000,
+      infl4 = 2500, neg = FALSE, lv = 0.2, uv = 0.8)
> plat.ramp.sel(x = data, infl1 = 500, infl2 = 1000, infl3 = 2000,
+      infl4 = 2500, neg = TRUE, lv = 0.2, uv = 0.8)
```

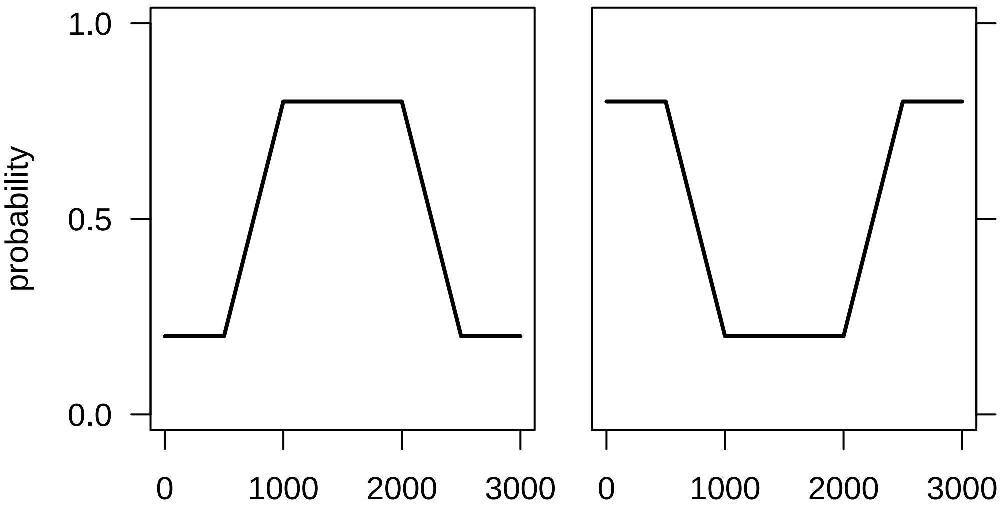

Figure 10: Type ’plat.ramp’ probabilities (left -> neg=FALSE | right -> neg=TRUE). In this example, minimum and maximum probabilities are respectively lv=0.2 and uv=0.8.

```
> data = 0:3000
> plat.ramp.sel(x = data, infl1 = 500, infl2 = 1000, infl3 = 2000,
+     infl4 = 2500, neg = FALSE, lv = c(0.2, 0.4), uv = 0.8)
> plat.ramp.sel(x = data, infl1 = 500, infl2 = 1000, infl3 = 2000,
+     infl4 = 2500, neg = TRUE, lv = 0.2, uv = c(0.8, 0.6))
```

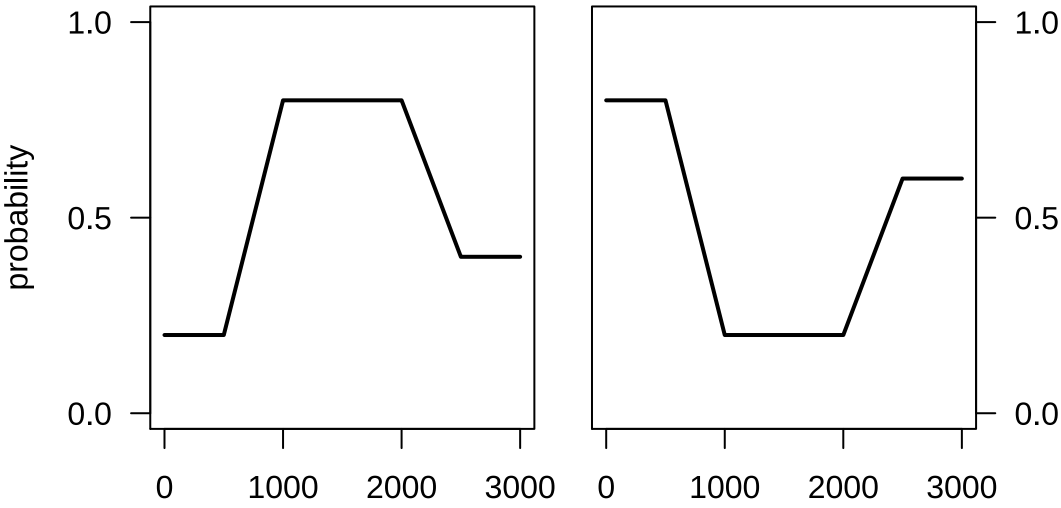

Figure 11: Type 'plat.ramp' probabilities (left -> neg=FALSE, lv=c(0.2,0.4) , uv=0.8 | right -> neg=TRUE, lv=0.2, uv=c(0.8,0.6)).

3 Types ‘logit’ and ‘plat.logit’

These relations use logistic curves. Inflection points are defined as points where the intantaneous splope is a proportion (`prop`) of the intantenuous slope at x_{50}. These types make use of the function `find.beta()` of `package::bmisc`. Default value of `prop` is `0.1`. The end result is a logistic curve with x_{50} being the midpoint between the inflection points. Two or four inflection points are needed. The main difference between ‘logit’(Figure 12 & 13) and ‘plat.logit’ (Figure 14, 15 & 16) types are the number inflection points.

logit.sel `(x, infl1, infl2, neg=FALSE, lv=0, uv=1, ...)`

plat.logit.sel `(x, infl1, infl2, infl3, infl4, neg=FALSE,`
`lv=c(0,0), uv=c(1,1), ...)`

where `infl1` to `infl4` are the inflection points, `x` is a numeric vector for which probabilities are estimated, `neg` indicates if the trend is negative (`TRUE`) or positive (`FALSE`), `lv` defines the lower probability values of the relation, and `uv` defines the upper probability values of the relation. By default, all fonctions have `neg=FALSE`, `lv=c(0,0)`, and `uv=c(1,1)`. Additionnal options of `find.beta()` can be added. Default values are `prob=NULL`, `prop=0.1`,`beta=0.2`, and `fast=TRUE`.

Here are examples for ‘logit’ type:

```
> logit.sel(x = data, infl1 = 1000, infl2 = 2000, neg = FALSE)
> logit.sel(x = data, infl1 = 1000, infl2 = 2000, neg = TRUE)
```

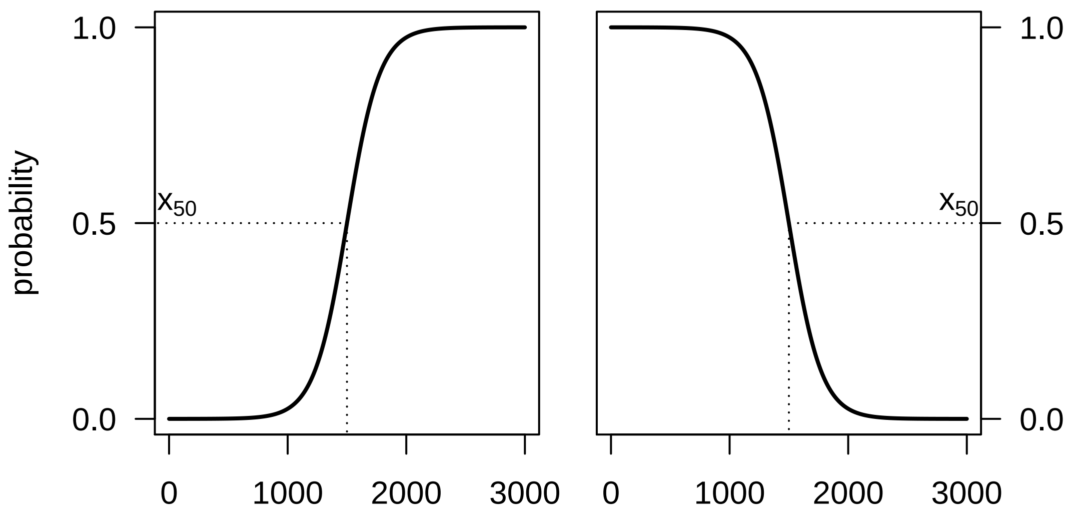

Figure 12: Type ’logit’ probabilities (left -> neg=FALSE | right -> neg=TRUE).

```
> logit.sel(x = data, infl1 = 1000, infl2 = 2000, neg = FALSE,
+     lv = 0.2, uv = 0.8)
> logit.sel(x = data, infl1 = 1000, infl2 = 2000, neg = TRUE, lv = 0.2,
+     uv = 0.8)
```

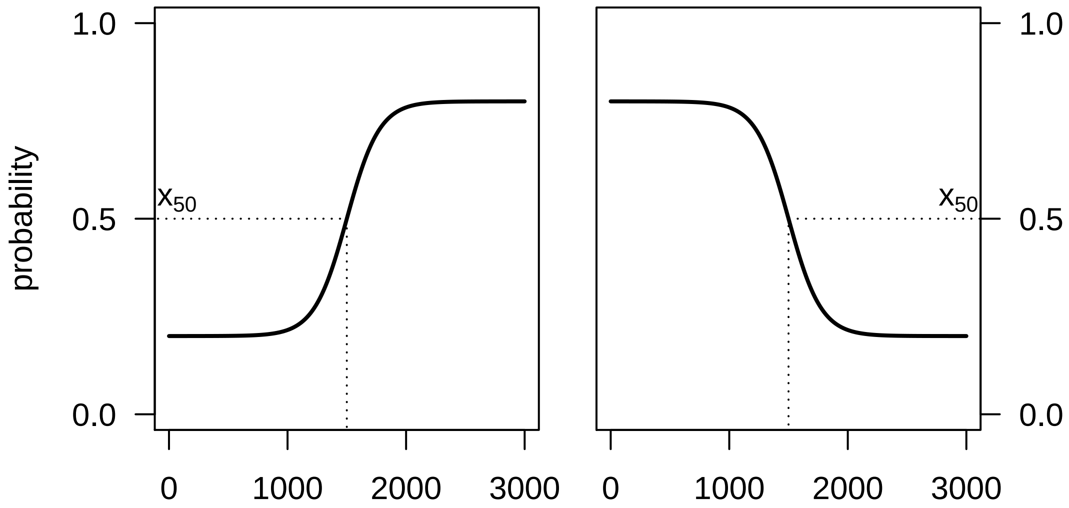

Figure 13: Type 'logit' probabilities (left -> neg=FALSE | right -> neg=TRUE). In this example, minimum and maximum probabilities are respectively lv=0.2 and uv=0.8.

Here are examples for 'plat.logit' type:

```
> data = 0:3000
> plat.logit.sel(x = data, infl1 = 500, infl2 = 1000, infl3 = 2000,
+     infl4 = 2500, neg = FALSE)
> plat.logit.sel(x = data, infl1 = 500, infl2 = 1000, infl3 = 2000,
+     infl4 = 2500, neg = TRUE)
```

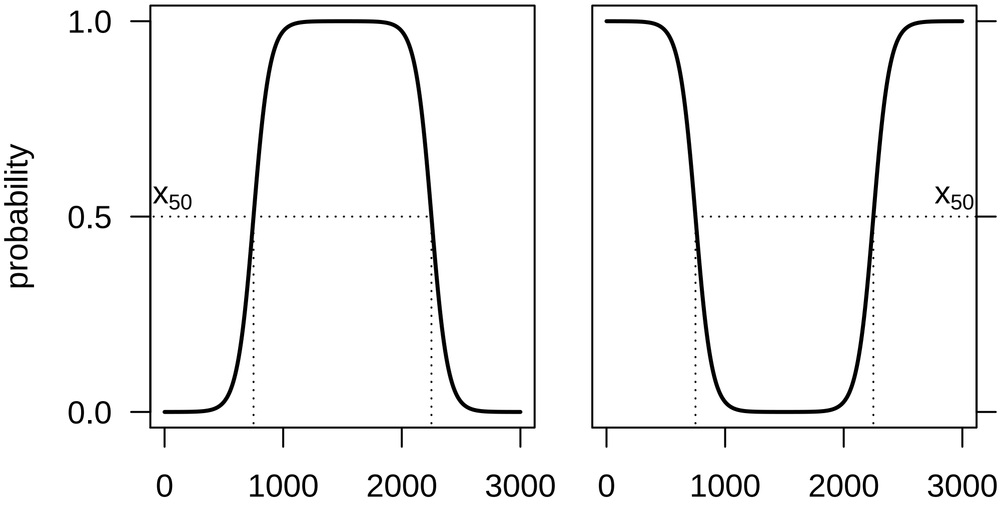

Figure 14: Type 'plat.logit' probabilities (left -> neg=FALSE | right -> neg=TRUE).

```
> data = 0:3000
> plat.logit.sel(x = data, infl1 = 500, infl2 = 1000, infl3 = 2000,
+     infl4 = 2500, neg = FALSE, lv = 0.2, uv = 0.8)
> plat.logit.sel(x = data, infl1 = 500, infl2 = 1000, infl3 = 2000,
+     infl4 = 2500, neg = TRUE, lv = 0.2, uv = 0.8)
```

```
> data = 0:3000
> plat.logit.sel(x = data, infl1 = 500, infl2 = 1000, infl3 = 2000,
+     infl4 = 2500, neg = FALSE, lv = c(0.2, 0.4), uv = 0.8)
> plat.logit.sel(x = data, infl1 = 500, infl2 = 1000, infl3 = 2000,
+     infl4 = 2500, neg = TRUE, lv = 0.2, uv = c(0.8, 0.6))
```

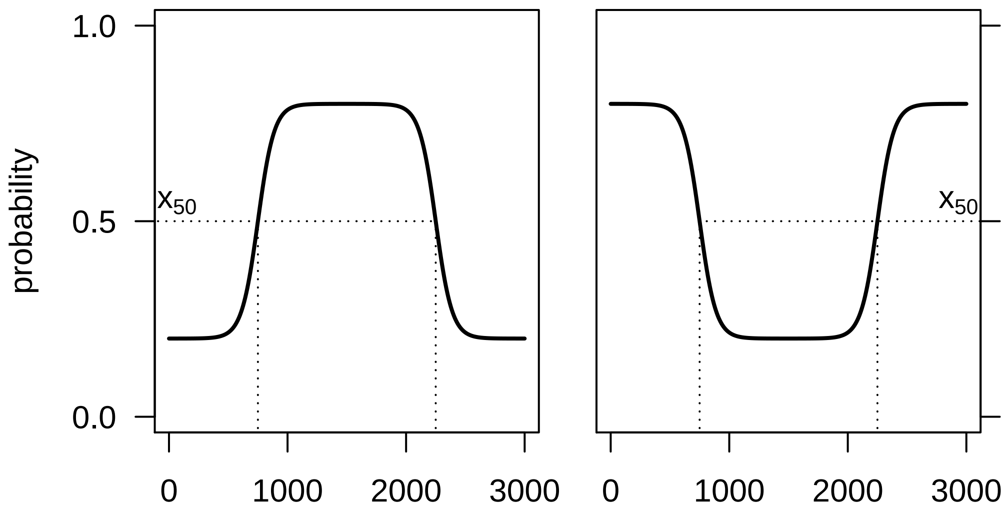

Figure 15: Type 'plat.logit' probabilities (left -> neg=FALSE | right -> neg=TRUE). In this example, minimum and maximum probabilities are respectively lv=0.2 and uv=0.8.

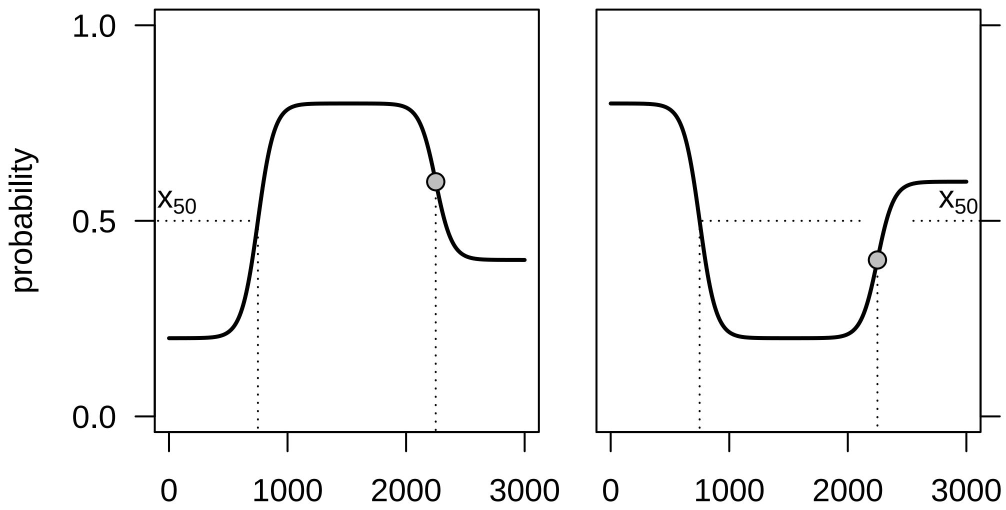

Figure 16: Type 'plat.logit' probabilities (left -> neg=FALSE, lv=c(0.2,0.4) , uv=0.8 | right -> neg=TRUE, lv=0.2, uv=c(0.8,0.6)).