

# Package: fca (via r-universe)

August 20, 2024

**Title** Floating Catchment Area (FCA) Methods to Calculate Spatial Accessibility

**Version** 0.1.0

**Description** Perform various floating catchment area methods to calculate a spatial accessibility index (SPAI) for demand point data. The distance matrix used for weighting is normalized in a preprocessing step using common functions (gaussian, gravity, exponential or logistic).

**License** GPL (>= 3)

**URL** <https://egrueebler.github.io/fca/>,  
<https://github.com/egrueebler/fca/>

**BugReports** <https://github.com/egrueebler/fca/issues/>

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.2

**Suggests** covr, knitr, rmarkdown, testthat

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Repository** <https://egrueebler.r-universe.dev>

**RemoteUrl** <https://github.com/egrueebler/fca>

**RemoteRef** HEAD

**RemoteSha** f813216e2c52c2d480ff540b96656f412120bcd2

## Contents

dist_normalize . . . . .	2
spai_2sfca . . . . .	2
spai_3sfca . . . . .	3
spai_mh3sfca . . . . .	4

**Index****5**


---

dist_normalize	<i>Distance weight methods</i>
----------------	--------------------------------

---

**Description**

Distance weight methods

**Usage**

```
dist_normalize(D, d_max, imp_function, function_d_max = 0.01)
```

**Arguments**

D	numeric matrix, distance or time values
d_max	numeric, threshold for max distance
imp_function	character, type of distance weights method
function_d_max	numeric, condition for the result of the function(d_max) used to calculate beta (default = 0.01, is considered optimal for the Gaussian function)

**Value**

matrix, normalized distance or time values

**Examples**

```
dist_normalize(matrix(10), 10, "gaussian")
```

---

spai_2sfca	<i>Two-Step Floating Catchment Area method</i>
------------	--

---

**Description**

Two-Step Floating Catchment Area method

**Usage**

```
spai_2sfca(p, s, W, step = 2)
```

**Arguments**

p	numeric vector, number of population at origin locations
s	numeric vector, capacity of services at supply locations
W	numeric matrix, distance or time matrix
step	numeric, number of the steps of the method to perform

**Value**

data.frame, depending on selected step

**Examples**

```
p <- 1:4
s <- 1:6
W <- matrix(1:24, ncol = 4, nrow = 6)
spai <- spai_2sfca(p, s, W, step = 2)
```

---

spai\_3sfca

*Three-Step Floating Catchment Area method*

---

**Description**

Three-Step Floating Catchment Area method

**Usage**

```
spai_3sfca(p, s, W, step = 3)
```

**Arguments**

p	numeric vector, number of population at origin locations
s	numeric vector, capacity of services at supply locations
W	numeric matrix, distance or time matrix
step	numeric, number of the steps of the method to perform

**Value**

data.frame, depending on selected step

**Examples**

```
p <- 1:4
s <- 1:6
W <- matrix(1:24, ncol = 4, nrow = 6)
spai <- spai_3sfca(p, s, W, step = 3)
```

---

`spai_mh3sfca`*Modified-Huff-Three-Step Floating Catchment Area method*

---

**Description**

Modified-Huff-Three-Step Floating Catchment Area method

**Usage**

```
spai_mh3sfca(p, s, W, step = 3)
```

**Arguments**

<code>p</code>	numeric vector, number of population at origin locations
<code>s</code>	numeric vector, capacity of services at supply locations
<code>W</code>	numeric matrix, distance or time matrix
<code>step</code>	numeric, number of the steps of the method to perform

**Value**

data.frame, depending on selected step

**Examples**

```
p <- 1:4
s <- 1:6
W <- matrix(1:24, ncol = 4, nrow = 6)
spai <- spai_mh3sfca(p, s, W, step = 3)
```

# Index

`dist_normalize`, [2](#)

`spai_2sfca`, [2](#)

`spai_3sfca`, [3](#)

`spai_mh3sfca`, [4](#)