# Package 'edgefx'

March 8, 2010

**Type** Package

**Title** edgefx: Edge effects on an ecological landscape.

**Version** 0.2.2

**Date** 2010-03-08

**Author** Emma E. Goldberg, Leslie Ries

**Maintainer** Emma E. Goldberg <eeg@uic.edu>

**Depends** R (>= 2.4)

**Suggests** MASS, MCMCpack

**Description** Allow all edges on a landscape to exert effects on observed values at any location. Original inspiration is Malcolm (1994) Ecology 75:2438-2445.

**License** GPL (>=2)

**LazyLoad** yes

## R topics documented:

---

`edgefx-package` *Edge Effects on an Ecological Landscape*

---

## Description

Allow all edges on a landscape to exert effects on observed values at any location. Original inspiration is Malcolm (1994) Ecology 75:2438-2445.

## Details

| | |
|---|---|
| Package: | edgefx |
| Type: | Package |
| Version: | 0.2.2 |
| Date: | 2010-03-08 |
| License: | GPL (>=2) |
| LazyLoad: | yes |

Please see `vignette("edgefx")` for basic usage.

## Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

## References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

---

`distance` *Compute the Distance between Two Points*

---

## Description

Computes the Euclidean distance between two points.

## Usage

```
distance(pt1, pt2)
```

### Arguments

| | |
|---|---|
| pt1 | A vector of the coordinates of the first point. |
| pt2 | A vector of the coordinates of the second point. |

### Details

Any number of dimensions is okay.

### Value

A scalar

### Note

See `vignette("edgefx")` for an example.

### Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

### References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

### Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (pt1, pt2)
{
    sqrt(sum((pt1 - pt2)^2))
}
```

---

## `draw.edges` *Plot of Focal Point and Edges*

---

### Description

Make a plot showing the focal point and all its associated edges.

### Usage

```
draw.edges(edgedat, ...)
```

## Arguments

| | |
|---|---|
| edgedat | A dataframe |
| ... | Additional arguments to be passed to plot. |

## Details

The first four columns of edgedat are used. The first row is for the focal point, giving its x and y coordinates in the first two columns. Each additional row is for an edge segment, giving the coordinates of one end (first two columns) and the other end (second two columns).

## Value

Produces a plot.

## Note

See vignette("edgefx") for an example.

## Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

## References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

## Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (edgedat, ...)
{
    xs <- c(edgedat[, 1], edgedat[, 3])
    ys <- c(edgedat[, 2], edgedat[, 4])

    plot(c(xs, edgedat[1, 1]), c(ys, edgedat[1, 2]), type="n", ...)
    points(edgedat[1, 1], edgedat[1, 2], col = "#D55E00", pch = 16)
    for (n in seq(length(edgedat[, 1]) - 1)) {
        lines(edgedat[n + 1, c(1, 3)], edgedat[n + 1, c(2, 4)],
            col = "#0072B2", lwd = 3)
    }
  }
```

`edge.lnL` *Likelihood Function for Edge Data*

## Description

Compute the log-likelihood (to within an additive constant) of observing the data, under the multiple-edge model.

## Usage

```
edge.lnL(params, edgecoord, observed, family="gaussian", neg=FALSE)
```

## Arguments

| | |
|---|---|
| `params` | The parameter values. |
| `edgecoord` | A dataframe of edges relocated to standard coordinates. |
| `observed` | A vector of observed response values, one for each focal point. |
| `family` | The error distribution of the observations. |
| `neg` | Whether to return the negative of the log-likelihood. |

## Details

`params` must be given as a vector. If it has names, they should be those given here. If it has no names, the elements should be in this order.

**e0** The maximum effect (at distance 0)

**Dmax** The maximum distance at which any effect is felt

**k** The baseline value (at large distance)

**D0** The distance out to which the maximum effect is felt. If omitted, its value will be fixed to 0.

The input `edgecoord` can be produced with `relocate.edge.df`. It is a dataframe with one row for each focal point–edge pair in the map. The columns are:

**map** The name of the map containing the focal point–edge pair

**x0** The x coordinate of the focal point (its y value is 0)

**y1** The y coordinate of one end of the edge segment (its x value is 0)

**y2** The y coordinate of the other end of the edge segment (its x value is 0)

There are two options for `family`:

**"gaussian"** Gaussian errors

**"poisson"** Poisson errors

## Value

A floating point value that is proportional to the log-likelihood of the data under the multiple-edge model with the specified parameter values.

### Note

Useful with `optim` or a general MCMC routine. See `vignette("edgefx")` for an example.

### Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

### References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

### Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function(params, edgecoord, observed, family="gaussian", neg=FALSE)
{
    if (!is.null(names(params)))
    {
        params <- c(params["e0"], params["Dmax"], params["k"], params["D0"])
    }
    # if the params vector doesn't have column names, assume it's in the right order
    if (length(params) == 3 || is.na(params[4]))
    {
        params[4] <- 0
    }

    if (params[2] < 0 || params[4] < 0 || params[4] > params[2])
    {
        lnL <- -Inf
    } else
    {
        xvals <- edgecoord[, 1]
        predicted <- as.vector(by(edgecoord[, 2:4], xvals, map.edge.effect, params[1], param
        if (family == "gaussian")
        {
            lnL <- -sum((observed - predicted)^2)
            # could easily include weights:  diff / (2*var)
        } else if (family == "poisson")
        {
            if (any(predicted <= 0))
            {
                lnL <- -Inf
            } else
            {
                lnL <- sum(dpois(observed, predicted, log=T))
```

```
                # identical to:
                #    sum(-log(factorial(observed)) + observed*log(predicted) - predicted)
            }
        } else
        {
            stop(paste("ERROR: family", family, "not available; use gaussian or poisson."))
        }
    }
    if (neg)
    {
        return(-lnL)
    } else
    {
        return(lnL)
    }

}
```

---

`edge.nls` *Non-Linear Least Squares for Edge Data*

---

## Description

Fit a set of edges and observations at focal points to estimate the parameters of the point edge effect function.

## Usage

```
edge.nls(edgelist, observed, guess, ...)
```

## Arguments

| | |
|---|---|
| `edgelist` | A list of dataframes, each giving focal point and edge coordinates. Or a dataframe with equivalent information, as produced by, e.g., `relocate.edge.df`. |
| `observed` | A vector of observed response values, one for each focal point. |
| `guess` | A list containing the initial guess at the parameter values. |
| `...` | Additional arguments to be passed to `nls`. |

## Details

Each element of `edgelist` gives coordinates of the edges relevant to a focal point. Each of these elements is a dataframe with these features:

- The first four columns are used.
- The first row is for the focal point, giving its x and y coordinates in the first two columns.
- Each additional row is for an edge segment, giving the coordinates of one end (first two columns) and the other end (second two columns).

The components of `guess` are:

**e0** The maximum effect (at distance 0)

**k** The baseline value (at large distance)

**Dmax** The maximum distance at which any effect is felt

**D0** The distance out to which the maximum effect is felt. If omitted, its value will be fixed to 0.

The call to `nls` uses `algorithm="port"` and `lower=list(e0=-Inf, Dmax=0, k=-Inf, D0=0)` to ensure that Dmax and D0 are non-negative.

## Value

An `nls` object

## Note

See `vignette("edgefx")` for an example.

## Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

## References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

## Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (edgelist, observed, guess, ...)
{
    if (class(edgelist) == "list")
    {
        edges <- relocate.edge.df(edgelist)
    } else
    {
        edges <- edgelist
    }

    xvals <- edges[,1]

    if (is.null(guess$D0))
    {
        edge.formula <- formula(observed ~ by(edges[,2:4], xvals,
```

```
                                map.edge.effect, e0, Dmax, k)[xvals])
    } else
    {
        edge.formula <- formula(observed ~ by(edges[,2:4], xvals,
                                map.edge.effect, e0, Dmax, k, D0)[xvals])
    }

    fit <- nls(edge.formula, start=guess, algorithm="port",
              lower=list(e0=-Inf, Dmax=0, k=-Inf, D0=0), ...)
    return(fit)

}
```

---

`find.edges` *Identify Edge Cells*

---

## Description

Find the coordinates of edge cells in a habitat grid. Edges are non-habitat cells that border at least one habitat cell.

## Usage

```
find.edges(map)
```

## Arguments

| | |
|---|---|
| `map` | A matrix with one number per cell. 0 is non-habitat, numbers greater than 0 are habitat. |

## Value

A dataframe with one row per edge cell found. The columns `row` and `col` give the coordinates.

## Note

See `vignette("edgefx")` for an example.

## Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

## References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

## Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (map)
{
    nonhabitat <- data.frame(which(map == 0, arr.ind = T))
    edges <- nonhabitat[apply(nonhabitat, 1, is.edge, map),]
    return(edges)
  }
```

---

`grid.edge.effect` *Compute Edge Effect a Specified Grid Cells*

---

## Description

For each cell in a list of focal cells, compute its response value, incorporating the effects of all edge cells.

## Usage

```
grid.edge.effect(focal, edges, map, param)
```

## Arguments

| | |
|---|---|
| `focal` | A matrix or dataframe of focal cell coordinates. Each row is a focal cell, the first column is its row coordinate and the second column is its column coordinate. |
| `edges` | A dataframe with one row per edge cell. The columns `row` and `col` give the coordinates. Can be produced with `find.edges`. |
| `map` | A matrix with one number per cell. 0 is non-habitat, numbers greater than 0 are habitat. |
| `param` | A list of parameter values for the plateau point edge effect function. |

## Details

The components of `param` are:

**e0** The maximum effect (at distance 0)

**k** The baseline value (at large distance) (not actually required here)

**Dmax** The maximum distance at which any effect is felt

**D0** The distance out to which the maximum effect is felt

## Value

A vector with one item per focal cell, in the same order as the focal cell matrix. Non-habitat cells get `NA`.

## Note

This step can be slow for a large grid and large values of `Dmax`. See `vignette("edgefx")` for an example.

## Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

## References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

## Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (focal, edges, map, param)
{
    if (class(focal) != "data.frame") {
        focal <- data.frame(row = focal[1], col = focal[2])
    }
    if (map[focal$row, focal$col] == 0) {
        return(NA)
    }
    else {
        dx <- param$Dmax
        near.edges <- edges[edges$row %in% seq(focal$row - dx,
            focal$row + dx) & edges$col %in% seq(focal$col -
            dx, focal$col + dx), ]
        distances <- apply(near.edges, 1, distance, focal)
        distances <- distances[distances < param$Dmax]
        if (length(distances) == 0) {
            return(param$k)
        }
        else {
            edge.effects <- sapply(distances, cell.edge.effect,
                param)
            return(sum(edge.effects) + param$k)
        }
    }
  }
```

`grid.effects` *Compute Edge Effect at Each Grid Cell*

## Description

Treat each habitat cell in turn as the focal cell and compute its response value, incorporating the effects of all edge cells.

## Usage

```
grid.effects(map, edges, param)
```

## Arguments

| | |
|---|---|
| `map` | A matrix with one number per cell. 0 is non-habitat, numbers greater than 0 are habitat. |
| `edges` | A dataframe with one row per edge cell. The columns `row` and `col` give the coordinates. Can be produced with `find.edges`. |
| `param` | A list of parameter values for the plateau point edge effect function. |

## Details

The components of `param` are:

**e0** The maximum effect (at distance 0)

**k** The baseline value (at large distance) (not actually required here)

**Dmax** The maximum distance at which any effect is felt

**D0** The distance out to which the maximum effect is felt

## Value

A vector with one item per cell, ordered by column (read down column 1, then read down column 2, etc.). Non-habitat cells get `NA`.

## Note

This step can be slow for a large grid and large values of `Dmax`. See `vignette("edgefx")` for an example.

## Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

## References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

## Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
grid.effects <- function(map, edges, params)
{
    focals <- data.frame(which(map>=0, arr.ind=T))
    edgeff <- apply(focals, 1, grid.edge.effect, edges, map, params)

    return(edgeff)
}
```

---

# infinite.edge.effect

*Point Edge Effect Function*

---

## Description

Compute the effect of an infinitely long edge at another location (the focal point).

## Usage

```
infinite.edge.effect(d, e0, Dmax=NULL, k=NULL, D0=0)
```

## Arguments

| | |
|---|---|
| d | The (perpendicular) distance between the edge and the focal point. |
| e0 | A scalar value for the parameter e0, or a vector or list of all parameters. See Details. |
| Dmax | A scalar value for the parameter Dmax; see Details. |
| k | A scalar value for the parameter k; see Details. |
| D0 | A scalar value for the parameter D0; see Details. |

## Details

If e0 is a scalar value, the remaining three arguments are used. But it can also be a list or a vector containing elements e0, Dmax, k, and optionally D0. If the vector elements are unnamed, they are assumed to be in that order. If a list or a vector is given as the second argument, the last three arguments are ignored.

## Value

A scalar.

## Note

See `vignette("edgefx")` for examples.

## Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

## References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

## Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (d, e0, Dmax=NULL, k=NULL, D0=0)
{
    if (length(d) != 1)
    {
        stop("distance argument must be a scalar")
    }

    if (length(e0) == 1)
    {
        if (is.null(Dmax) || is.null(k))
            stop("invalid parameter specification")
    } else   # "e0" could actually be a list or vector of all parameters
    {
        params <- unlist(e0)
        if (is.null(names(params)))
            names(params) <- c("e0", "Dmax", "k", "D0")[1:length(params)]

        e0 <- params["e0"]
        Dmax <- params["Dmax"]
        k <- params["k"]
        D0 <- params["D0"]
        if (is.na(D0))  D0 <- 0
        names(e0) <- names(Dmax) <- names(k) <- names(D0) <- NULL
    }

    p <- rbind(c(d, 0), c(0, -Inf), c(0, Inf))
    return(segment.edge.effect(p, e0, Dmax, D0) + k)
```

```
}
```

---

`is.edge` *Test if a Cell is an Edge*

---

## Description

In a gridded landscape, test if a cell is an edge.

## Usage

```
is.edge(cell, map)
```

## Arguments

| | |
|---|---|
| `cell` | A vector or dataframe with two components, the (x, y) coordinates of the cell. |
| `map` | The gridded habitat map. |

## Value

TRUE or FALSE

## Note

Used by `find.edges`.

## Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

## References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

## Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (cell, map)
{
    if (class(cell) != "data.frame") {
        cell <- data.frame(row = cell[1], col = cell[2])
    }
```

```
    if (map[cell$row, cell$col] > 0) {
        return(FALSE)
    }
    else {
        neighbors <- matrix(c(rep(cell$row, 2), c(-1, 1) + cell$row,
            c(-1, 1) + cell$col, rep(cell$col, 2)), nrow = 4,
            dimnames = list(seq(4), c("row", "col")))
        neighbors <- neighbors[neighbors[, 1] > 0 & neighbors[,
            1] <= nrow(map) & neighbors[, 2] > 0 & neighbors[,
            2] <= ncol(map), ]
        return(sum(map[neighbors]) != 0)
    }
}
```

---

`map.edge.effect` *Compute Effect of all Edges in a Map*

---

## Description

Compute the edge effect at the focal point, incorporating all its relevant edges.

## Usage

```
map.edge.effect(edgecoord, e0, Dmax, k, D0 = 0)
```

## Arguments

| | |
|---|---|
| `edgecoord` | A dataframe |
| `e0` | The maximum effect (at distance 0) |
| `Dmax` | The maximum distance at which any effect is felt |
| `k` | The baseline value (at large distance) |
| `D0` | The distance out to which the maximum effect is felt |

## Details

The input `edgecoord` can be produced with `relocate.edge.df`. It is a dataframe with one row for each focal point–edge pair in the map. The columns are:

**map** The name of the map containing the focal point–edge pair

**x0** The x coordinate of the focal point (its y value is 0)

**y1** The y coordinate of one end of the edge segment (its x value is 0)

**y2** The y coordinate of the other end of the edge segment (its x value is 0)

## Value

A scalar

### Note

This function is intended to be used with `edge.nls`. See `vignette("edgefx")` for an example.

### Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

### References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

### Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (edgecoord, e0, Dmax, k, D0 = 0)
{
    ans = apply(edgecoord, 1, segment.effect.wrapper, e0, Dmax, D0)
    return(sum(ans) + k)
  }
```

---

## `point.edge.effect` *Point Edge Effect Function*

---

### Description

Compute the effect of a very small amount of edge (a point source) at another location (the focal point).

### Usage

```
point.edge.effect(d, param)
```

### Arguments

| | |
|---|---|
| `d` | The distance between the edge point source and the focal point |
| `param` | A list of parameter values for the plateau point edge effect function (see below) |

### Details

The components of `param` are:

**e0** The maximum effect (at distance 0)

**k** The baseline value (at large distance) (not actually used here)

**Dmax** The maximum distance at which any effect is felt

**D0** The distance out to which the maximum effect is felt

### Value

A scalar.

### Note

See `vignette("edgefx")` for the actual function.

### Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

### References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

### Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (d, param)
{
    if (length(d) != 1) {
        stop("distance argument must be a scalar")
    }
    if (d < 0) {
        warning("negative distance; using absolute value")
        d <- -d
    }
    if (d <= param$D0) {
        return(p$e0)
    }
    else if (d <= param$Dmax) {
        return(param$e0 * (1 + (param$D0 - d)/(param$Dmax - param$D0)))
    }
    else {
        return(0)
```

```
    }
  }
```

---

`relocate.edge.df` *Relocate Edges to Standard Coordinates*

---

## Description

For each focal point–edge pair, rotate and translate them so that the focal point is on the positive x axis and the edge lies along the y axis.

## Usage

```
relocate.edge.df(edgelist)
```

## Arguments

| | |
|---|---|
| `edgelist` | A list of dataframes, each giving focal point and edge coordinates. |

## Details

Each element of `edgelist` gives coordinates of the edges relevant to a focal point. Each of these elements is a dataframe with these features:

- The first four columns are used.
- The first row is for the focal point, giving its x and y coordinates in the first two columns.
- Each additional row is for an edge segment, giving the coordinates of one end (first two columns) and the other end (second two columns).

## Value

A dataframe with one row for each focal point–edge pair. The columns are:

| | |
|---|---|
| `map` | The name of the map containing the focal point–edge pair |
| `x0` | The x coordinate of the focal point (its y value is 0) |
| `y1` | The y coordinate of one end of the edge segment (its x value is 0) |
| `y2` | The y coordinate of the other end of the edge segment (its x value is 0) |

## Note

See `vignette("edgefx")` for an example.

## Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

## References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

## Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (edgelist)
{
    edges.per.map = sapply(edgelist, nrow) - 1
    mapnames = rep(names(edgelist), edges.per.map)
    num.sets = length(mapnames)
    edges1 = matrix(NA, nrow = num.sets, ncol = 3, dimnames = list(c(),
        c("x0", "y1", "y2")))
    atrow = 0
    for (j in seq(length(edgelist))) {
        dat = edgelist[[j]]
        focal = as.matrix(dat[1, 1:2])
        edges = as.matrix(dat[-1, 1:4])
        ans = apply(edges, 1, relocate.wrapper, focal)
        for (i in seq(ncol(ans))) {
            atrow = atrow + 1
            edges1[atrow, 1:3] = ans[, i]
        }
    }
    edges2 = cbind(mapnames, data.frame(edges1))
    return(edges2)
  }
```

---

# segment.edge.effect

*Segment Edge Effect Function*

---

## Description

Compute the effect of a finite edge at another location (the focal point).

## Usage

```
segment.edge.effect(p, e0, Dmax, D0)
```

### Arguments

| | |
|---|---|
| p | A matrix or dataframe with two columns ((x, y) coordinates) and three rows (focal point, edge end, other edge end). The coordinates must be transformed (via e.g. `relocate.edge`) so that p[1][2] = p[2][1] = p[3][1] = 0. |
| e0 | The maximum effect (at distance 0) |
| Dmax | The maximum distance at which any effect is felt |
| D0 | The distance out to which the maximum effect is felt |

### Value

A scalar.

### Note

Rather than calling this function directly, you will probably use it through `point.edge.effect`, `infinite.edge.effect`, or `map.edge.effect`.

### Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

### References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

### Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (p, e0, Dmax, D0)
{
    p0 = p[1, ]
    p1 = p[2, ]
    p2 = p[3, ]
    if (p0[1] < Dmax) {
        yDmax <- sqrt(Dmax^2 - p0[1]^2)
    }
    if (p0[1] >= Dmax || p1[2] == p2[2] || -yDmax > max(p1[2],
        p2[2]) || yDmax < min(p1[2], p2[2])) {
        answer <- 0
    }
    else if (p0[1] >= D0) {
        yDmax <- sqrt(Dmax^2 - p0[1]^2)
        amin <- max(min(p1[2], p2[2]), -yDmax)
```

```
        amax <- min(max(p1[2], p2[2]), yDmax)
        answer <- def.integral(amin, amax, p0[1], e0, D0, Dmax)
    }
    else if (p0[1] < D0) {
        yD0 <- sqrt(D0^2 - p0[1]^2)
        a1min <- max(min(p1[2], p2[2]), -yD0)
        a1max <- min(max(p1[2], p2[2]), yD0)
        yDmax <- sqrt(Dmax^2 - p0[1]^2)
        a2min <- max(min(p1[2], p2[2]), -yDmax)
        a2max <- min(max(p1[2], p2[2]), yDmax)
        answer <- e0 * (a1max - a1min)
        if (a2min < a1min) {
            answer <- answer + def.integral(a2min, a1min, p0[1],
                e0, D0, Dmax)
        }
        if (a2max > a1max) {
            answer <- answer + def.integral(a1max, a2max, p0[1],
                e0, D0, Dmax)
        }
    }
    return(answer)
}
```

---

`vecmap.edge.effect` *Point Edge Effect Function*

---

## Description

Compute the effect of a vectorized map of edges at its focal point.

## Usage

```
vecmap.edge.effect(edgedat, e0, Dmax=NULL, k=NULL, D0=0)
```

## Arguments

| | |
|---|---|
| `edgedat` | A data.frame containing coordinates of the focal point and edges. See specifications in `draw.edges`. |
| `e0` | A scalar value for the parameter `e0`, or a vector or list of all parameters. See Details. |
| `Dmax` | A scalar value for the parameter `Dmax`; see Details. |
| `k` | A scalar value for the parameter `k`; see Details. |
| `D0` | A scalar value for the parameter `D0`; see Details. |

## Details

If `e0` is a scalar value, the remaining three arguments are used. But it can also be a list or a vector containing elements `e0`, `Dmax`, `k`, and optionally `D0`. If the vector elements are unnamed, they are assumed to be in that order. If a list or a vector is given as the second argument, the last three arguments are ignored.

## Value

A scalar.

## Note

See `vignette("edgefx")` for examples.

## Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

## References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

## Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (edgedat, e0, Dmax=NULL, k=NULL, D0=0)
{
    if (class(edgedat) != "data.frame")
    {
        stop("Must present map as a data.frame.")
    }

    if (length(e0) == 1)
    {
        if (is.null(Dmax) || is.null(k))
            stop("invalid parameter specification")
    } else   # "e0" could actually be a list or vector of all parameters
    {
        params <- unlist(e0)
        if (is.null(names(params)))
            names(params) <- c("e0", "Dmax", "k", "D0")[1:length(params)]

        e0 <- params["e0"]
        Dmax <- params["Dmax"]
        k <- params["k"]
        D0 <- params["D0"]
        if (is.na(D0))  D0 <- 0
        names(e0) <- names(Dmax) <- names(k) <- names(D0) <- NULL
    }

    edgecoord <- relocate.edge.df(list("map"=edgedat))
    return(map.edge.effect(edgecoord[,2:4], e0, Dmax, k, D0))
```

```
}
```

---

`write.edges` *Prettily Write Edges in a Habitat Grid to a File*

---

## Description

Writes a gridded map to a file, showing edge cells with an x other cells with a space.

## Usage

```
write.edges(map, edges, filename = "edges.txt")
```

## Arguments

| | |
|---|---|
| `map` | A matrix, in which habitat cells contain a positive number and non-habitat cells contain 0. Actually, it can be any matrix the size of the map. |
| `edges` | A dataframe with one row per edge cell found. The columns `row` and `col` give the coordinates. Can be produced with `find.edges`. |
| `filename` | The destination filename |

## Value

Creates a text file.

## Note

See `vignette("edgefx")` for an example.

## Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

## References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

### Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (map, edges, filename = "edges.txt")
{
    edge.grid <- matrix(NA, nrow = nrow(map), ncol = ncol(map))
    edge.grid[as.matrix(edges)] <- 1
    write.table(edge.grid, file = filename, na = " ", quote = F,
        sep = "", row.names = F, col.names = F)
  }
```

---

## write.grid *Prettily Write a Habitat Grid to a File*

---

### Description

Writes a gridded map to a file, showing habitat with a dot and non habitat with a space.

### Usage

```
write.grid(map, filename = "map.txt")
```

### Arguments

| | |
|---|---|
| map | A matrix, in which habitat cells contain a positive number and non-habitat cells contain 0. |
| filename | The destination filename |

### Value

Creates a text file.

### Note

See `vignette("edgefx")` for an example.

### Author(s)

Goldberg, E.E. and Ries, L.

Maintainer: Emma E. Goldberg <eeg@uic.edu>

### References

Malcolm (1994) Ecology 75:2438–2445

Ries etc.

## Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--    or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (map, filename = "map.txt")
{
    map[map > 0] <- "."
    map[map == 0] <- " "
    write.table(map, file = filename, quote = F, sep = "",
        row.names = F, col.names = F)
  }
```

# Index