Package 'incursion'

September 5, 2015

Type Package

Title Functions for the analysis of infectious disease outbreaks in animal populations

Version 0.1-18

Date 2015-09-05

Author Mark Stevenson <mark.stevenson1@unimelb.edu.au>, Robert Sanson <robert.sanson@asurequality.com>, and Andrew McFadden <andrew.mcfadden@mpi.govt.nz>.

Maintainer Mark Stevenson <mark.stevenson1@unimelb.edu.au>

Description A collection of functions to assist in the analysis of infectious disease outbreaks in animal populations.

Imports maptools, rgdal, survival, plotKML, spacetime, splancs, igraph

License GPL (>= 2)

URL http://fvas.unimelb.edu.au/veam

R topics documented:

inc.about *The library incursion: summary information*

Description

Functions for the analysis of infectious disease outbreaks in animal populations.

Usage

```
inc.about()
```

Details

The most recent version of the incursion package can be obtained from: http://fvas.unimelb.edu.au/veam.

Author(s)

Mark Stevenson (<mark.stevenson1@unimelb.edu.au>), Faculty of Veterinary and Agricultural Sciences, University of Melbourne, Parkville, Victoria 3010, Australia.

Robert Sanson, AsureQuality New Zealand, PO Box 1654, Palmerston North, New Zealand.

Andrew McFadden, Investigation and Diagnostic Centre Biosecurity New Zealand, PO Box 40742, Upper Hutt, New Zealand.

inc.den *Compute farm and animal density in an area around a defined point location*

Description

Compute farm and animal density in an area around a defined point location.

Usage

```
inc.den(loc, par, dim, shape = "square")
```

Arguments

loc	a vector of length two defining the Cartesian coordinates of the point location of interest.
par	a three-column data frame comprised of three columns listing (in order) the easting coordinate of the premise location, the northing coordinate of the premise location, and the number of animals present at each premise location.
dim	scalar, defining the diameter of the circle constructed around the point location of interest (where `shape = "circle"`) or the length of each side of a square constructed around the point location of interest (where `shape = "square"`).
shape	character string defining the shape of the area constructed around the point location of interest. Options are `square` or `circle`.

Details

Point location details must be provided as Cartesian coordinates (not latitude and longitude).

Value

A list containing the following:

premise	the number of premises in the area, the size of the area, and premise density.
animal	the number of animals in the area, the size of the area, and animal density.

Examples

```
## Cartesian coordinates of the point location of interest:
loc <- c(1789917, 5924157)

## Data frame providing details of the point locations of the farm population
## at risk:
data(inc.outbreak)
par <- inc.outbreak$par[,3:4]

# Generate animal counts for each farm in data frame par:
par$lsu <- round(runif(n = nrow(par), min = 1, max = 1500), digits = 0)

## We want to know the density of farms and the density of animals in a 5 km
## by 5 km box drawn around the point location of interest:
rval <- inc.den(loc = loc, par = par, dim = 5000, shape = "square")
rval

## There are 9 farms in the 5 km by 5 km area. Density is 3.6e-07 farms per
## square metre (densities are reported in metres squared because Cartesian
## coordinates are specified in metres). How many farms per square kilometre?
rval$premise$n * 1000^2 / rval$premise$area

## Answer: 0.36 farms per square kilometre.
```

---

inc.edr *Compute estimated dissemination ratio*

Description

Computes estimated dissemination ratio on the basis of a vector of numbers (usually counts of incident cases identified on each day of an epidemic).

Usage

```
inc.edr(dat, n = 4, conf.level = 0.95, nsim = 99, na.zero = TRUE)
```

Arguments

dat	a numeric vector representing the count of incident cases for each day of an epidemic.
n	scalar, defining the number of days to be used when computing the estimated dissemination ratio.
conf.level	scalar, defining the magnitude of the returned confidence interval. Must be a single number between 0 and 1.
nsim	scalar, defining the number of simulations to be used for the confidence interval calculations.
na.zero	logical, replace NaN or Inf values with zeros?

Details

In infectious disease epidemics the n-day estimated dissemination ratio (EDR) at day i equals the total number of incident cases between day i and day [i - (n - 1)] (inclusive) divided by the total number of incident cases between day (i - n) and day (i - 2n) + 1 (inclusive). EDR values are often calculated for each day of an epidemic and presented as a time series analysis. If the EDR is consistently less than unity, the epidemic is said to be 'under control.'

A simulation approach is used to calculate confidence intervals around each daily EDR estimate. The numerator and denominator of the EDR estimate for each day is taken in turn and a random number drawn from a Poisson distribution, using the calculated numerator and denominator value as the mean. EDR is then calculated for these simulated values and the process repeated nsim times. Confidence intervals are then derived from the vector of simulated values for each day.

Value

Returns the point estimate of the EDR and the lower and upper bounds of the confidence interval of the EDR.

References

Miller W (1976). A state-transition model of epidemic foot-and-mouth disease. In: Proceedings of an International Symposium: New Techniques in Veterinary Epidemiology and Economics, University of Reading, Reading, 56 - 72.

Morris R, Sanson R, Stern M, Stevenson M, Wilesmith J (2002). Decision-support tools for foot and mouth disease control. Revue Scientifique et Technique de l'Office International des Epizooties 21, 557 - 567.

Examples

```
data(inc.outbreak)
dat <- inc.outbreak$ips

dat$sgndate <- as.Date(dat$sgndate, format = "%Y-%m-%d")
range(dat$sgndate)
dat$days <- as.numeric(dat$sgndate - min(dat$sgndate))

## EDR plot for "beef" and "other" premises:
id <- dat$placetype == "BEF"
bef <- dat[id,]

id <- dat$placetype != "BEF"
```

```
oth <- dat[id,]

## Counts of infected premises:
bef.n <- hist(bef$days, breaks = seq(from = 0, to = 40, by = 1),
   plot = FALSE)
bef.n <- bef.n$counts

oth.n <- hist(oth$days, breaks = seq(from = 0, to = 40, by = 1),
   plot = FALSE)
oth.n <- oth.n$counts

## Four-day EDR:
bef.edr <- inc.edr(bef.n, n = 4, conf.level = 0.95, nsim = 99, na.zero = TRUE)
oth.edr <- inc.edr(oth.n, n = 4, conf.level = 0.95, nsim = 99, na.zero = TRUE)

## Plot:
par(pty = "s", mfrow = c(1,2))
plot(1:40, 1:40, xlim = c(0,25), ylim = c(0, 10), xlab = "Days",
   ylab = "Estimated dissemination ratio", type = "n", main = "Beef")
lines(1:40, bef.edr[,1], type = "l", lwd = 2, lty = 1, col = "blue")
lines(1:40, bef.edr[,2], type = "l", lwd = 1, lty = 2, col = "blue")
lines(1:40, bef.edr[,3], type = "l", lwd = 1, lty = 2, col = "blue")

plot(1:40, 1:40, xlim = c(0,25), ylim = c(0, 10), xlab = "Days",
   ylab = "Estimated dissemination ratio", type = "n", main = "Other")
lines(1:40, oth.edr[,1], type = "l", lwd = 2, lty = 1, col = "red")
lines(1:40, oth.edr[,2], type = "l", lwd = 1, lty = 2, col = "red")
lines(1:40, oth.edr[,3], type = "l", lwd = 1, lty = 2, col = "red")
```

`inc.fdi` *First day incidence*

Description

Returns a data frame of the within-place frequency of disease, expressed as first day incidence.

Usage

```
inc.fdi(ips, use = "species", conf.level = 0.95)
```

Arguments

`ips`	a 21 column data frame listing listing details of infected places. Columns (in order): unique place identifier, place type, infected place number, and: (1) number of animals present, (2) number of animals initially affected, and (3) estimated age of clinical signs at time of examination for pigs, dairy cattle, beef cattle, sheep, deer, and goats (respectively). There should be one row for each infected place.
`use`	a character string specifying which group to use as the denominator for the incidence risk calculations. Options are `species`: select the species with the oldest lesions and express first day incidence was the number of that species with signs divided by the total number of that species present on the infected place, `all`:

select the species with the oldest lesions and express first day incidence was the number of that species with signs divided by the total number of susceptible species present on the infected place

`conf.level` scalar, defining the magnitude of the returned confidence interval. Must be a single number between 0 and 1.

Details

First day incidence is a term coined by Hutber and Kitching (1996) to denote the number of animals showing clinical signs on the first day of a herd outbreak. It provides an indication of the number of animals initially infected by each particular exposure pathway that introduced infection onto the farm and indicates the infectiousness of the farm in terms of forward risk potential during the period from infection to diagnosis.

Confidence intervals for the incidence risk estimates are calculated using Wilson's approximation (see Rothman 2002, page 132).

The function takes each infected place in turn and chooses the species with the oldest clinical signs at the time of examination. First day incidence equals the number of that species with signs divided by the total number of that species present on the infected place. Where there are two or more animal species with the oldest clinical signs the function chooses one of them at random.

Value

A data frame containing the following:

`placeid`	the unique place identifier (taken directly from the input data frame).
`ipnumber`	the infected premises number (taken directly from the input data frame).
`est`	the point estimate of first day incidence risk.
`lower`	the lower bound of first day incidence risk.
`upper`	the upper bound of first day incidence risk.

References

Hutber AM, Kitching RP (1996). The use of vector transition in modelling of intra-herd foot-and-mouth disease. Environmental and Ecological Statistics 3: 245 - 255.

Rothman KJ (2002). Epidemiology An Introduction. Oxford University Press, London, pp. 130 - 143.

Examples

```
data(inc.outbreak)
ips <- inc.outbreak$ips[,c(1:3,11:28)]

res <- inc.fdi(ips, conf.level = 0.05)
rank <- rank(res$est)

## Not run:
library(Hmisc)
errbar(x = rank, y = res$est, yplus = res$upper, yminus = res$lower,
   xlab = "Rank", ylab = "First day incidence",
   pch = 16, lty = 1, lwd = 1, cap = 0.015)

## End(Not run)
```

`inc.ffi` *First fortnight incidence*

Description

Returns a data frame of first fortnight incidence (risk) for farm premises within a defined area.

Usage

```
inc.ffi(ips, par, start, period = 14, conf.level = 0.95)
```

Arguments

`ips`	an 8 column data frame listing details of infected places. Columns (in order): unique place identifier, place type, infected place number, easting coordinate of the place centroid, northing coordinate of the place centroid, infection date, date of onset of clinical signs, and visit date. There should be one row for each infected place.
`par`	a 5 column data frame listing: unique place identifier, place type, easting coordinate of the place centroid, northing coordinate of the place centroid, date of slaughter, date cleaning and disinfection completed.
`start`	the start date for the analysis.
`period`	the length of time (in days) to be analysed.
`conf.level`	scalar, defining the magnitude of the returned confidence interval. Must be a single number between 0 and 1.

Details

First fortnight incidence is a term coined by Hutber et al. (2006) as a predictor for regional prevalence zonal disease duration. The numerator is the number of infected places where the onset of clinical signs is from `start` to `start + period`. If `start` equals 22 Feb 2001 and `period` equals 14 days then selected interval is from 22 Feb 2001 to 7 Mar 2001 (inclusive). The denominator is the total number of susceptible places in the area of interest. Although 'fortnight' is used as the name for this function it should be noted that incidence risk can be calculated for any time interval following `start` date.

Confidence intervals for the incidence risk estimates are calculated using Wilson's approximation (see Rothman 2002, page 132).

Value

A data frame containing the following:

`est`	the point estimate of incidence risk for the prescribed interval.
`lower`	the lower bound of incidence risk for the prescribed interval.
`upper`	the upper bound of incidence risk for the prescribed interval.

References

Hutber AM, Kitching RP, Pilipcinec E (2006). Predictions for the timing and use of culling or vaccination during a foot-and-mouth disease epidemic. Research in Veterinary Science 81: 31 - 36.

Rothman KJ (2002). Epidemiology An Introduction. Oxford University Press, London, pp. 130 - 143.

Examples

```
data(inc.outbreak)
ips <- inc.outbreak$ips[,c(1:5,8:10)]
par <- inc.outbreak$par[,c(1:5)]

## Incidence risk for the 14 days following 10 Mar 2001:
res <- inc.ffi(ips = ips, par = par, start = as.Date("2001-03-10",
   format = "%Y-%m-%d"), period = 14, conf.level = 0.95)
round(res * 100, digits = 2)

## The incidence risk of disease for the 14 days following 10 Mar 2001 was
## 29 cases (95% CI 23 to 36) cases per 100 places at risk.
```

---

inc.genint *Estimated generation intervals*

Description

This function returns the generation interval, the number of days from the onset of clinical signs on a source place to the onset of clinical signs on one or more identified destination place(s).

Usage

```
inc.genint(tra, use = "all")
```

Arguments

`tra`	a 7 column data frame listing details of tracing events. Columns (in order): source place unique identifier, date of onset of signs on the source place, destination place unique identifier, date of onset of signs on the destination place, episode start date (the date that contact between the two places commenced), and episode end date (the date that contact between the two places ceased).
`use`	a character string, indicating the method to be used. Options are `all` calculate all generation intervals, or `first` calculate the first generation interval for each source place (if there are more than one). Option `first` should be used if you want to calculate the earliest generation interval.

Details

This function uses the unique numeric identifiers for places (as opposed to infected place number).

It is assumed that the input data for this function is a complete record of contact events between infected places. That is, both source and destination places are both infected (and therefore have an onset of clinical signs date).

Generation intervals that are negative (i.e. cases where the onset of clinical signs date on the destination place occurs before the onset of clinical signs on the source place) are ignored.

Value

A data frame containing the following:

src	the source place identifier (taken directly from the input data frame).
des	the destination place identifier (taken directly from the input data frame).
gen	the generation interval.

Examples

```
data(inc.outbreak)
tra <- inc.outbreak$tra

gen01 <- inc.genint(tra, use = "first")
hist(gen01$gen)
summary(gen01$gen)

## When all generation intervals are included, there may be some
## outliers:
gen02 <- inc.genint(tra, use = "all")
hist(gen02$gen)
summary(gen02$gen)
```

---

inc.incubation *Estimated infection dates and incubation periods*

Description

Returns a data frame of infection dates and incubation periods. Where data is missing or unavailable an incubation period is estimated by drawing a random number from a log normal distribution (the parameters of which may be specified by the user).

Usage

```
inc.incubation(ips, meanlog = log(7), sdlog = log(1.5))
```

Arguments

ips	a 12 column data frame listing details of infected places. Columns (in order): unique place identifier, place type, infected place number, infection date, date of onset of clinical signs, visit date, estimated age of clinical signs at time of examination for pigs, dairy cattle, beef cattle, sheep, deer, and goats. There should be one row for each infected premises.
meanlog	mean of the log normal distribution on the log scale.
sdlog	standard deviation of the log normal distribution on the log scale.

Details

In the input data frame the variables defining infection date and date of onset of signs are recorded at the place level. These values are used in preference to the species age of signs variables for the incubation period calculations.

The arguments `meanlog` and `sdlog` are used to define a log normal distribution which is used to estimate incubation period.

If (infection date present, signs date present, species age of signs absent) a random number is drawn from the log normal distribution to estimate incubation period.

If (infection date absent, signs date present, species age of signs absent) a random number is drawn from the log normal distribution to estimate incubation period. Infection date is estimated as signs date minus the estimated incubation period.

If (infection date absent, signs date absent, species age of signs present) a random number is drawn from the log normal distribution to estimate incubation period. Infection date is estimated as visit date minus the largest value of age of signs (across all species) minus the estimated incubation period.

If (infection date present, signs date present, species age of signs absent) incubation period equals signs date minus infection date.

If (infection date absent, signs date present, species age of signs present) a random number is drawn from the log normal distribution to estimate incubation period. Infection date is estimated as signs date minus the estimated incubation period.

If (infection date present, signs date present, species age of signs present) incubation period equals signs date minus infection date.

Value

A data frame containing the following:

`ip`	the infected place number (taken directly from the input data frame).
`inf.date`	actual or estimated infection date.
`inf.est`	indicator for status of infection date: 0 = actual, 1 = estimated.
`inc`	actual or estimated incubation period.
`inc.est`	indicator for status of incubation period: 0 = actual, 1 = estimated.

Note

Incubation period refers to the period from infection to the development of symptomatic disease. Latent period refers to the period between disease initiation and development of symptomatic disease. The term incubation period is equivalent to latent period: incubation period is the term specifically used for infectious diseases.

The antilog of one standard deviation from the mean log incubation period has been called the dispersion factor (Sartwell 1950). The dispersion factor multiplied by the mean log of the incubation period will define an interval above which 16% of the intervals will fall, and the mean divided by the dispersion factor will define the period below which 16% will occur.

References

Sartwell PE (1950). The distribution of incubation of disease. American Journal of Epidemiology 51: 310 - 318.

Examples

```
data(inc.outbreak)
ips <- inc.outbreak$ips[,c(1:3,8:10,13,16,19,22,25,28)]

fmd.inc <- inc.incubation(ips, meanlog = log(7), sdlog = log(1.5))

## Frequency histogram of incubation periods:
hist(fmd.inc$inc, xlab = "Incubation period (days)", main = "")

## Compare the distribution of actual incubation periods with those that
## have been estimated:
par(pty = "s", mfrow = c(1,2))
hist(fmd.inc$inc[fmd.inc$inc.est == 0], breaks = seq(from = 0, to = 50, by = 1),
   xlim = c(0, 30), ylim = c(0,50), xlab = "Days", main = "Actual")
hist(fmd.inc$inc[fmd.inc$inc.est == 1], breaks = seq(from = 0, to = 50, by = 1),
   xlim = c(0, 30), ylim = c(0,50), xlab = "Days", main = "Estimated")

## Alternative, using a stacked bar graph:
obs <- hist(fmd.inc$inc[fmd.inc$inc.est == 0],
   breaks = seq(from = 0, to = 50, by = 2), plot = FALSE)
est <- hist(fmd.inc$inc[fmd.inc$inc.est == 1],
   breaks = seq(from = 0, to = 50, by = 2), plot = FALSE)

rval <- matrix(rbind(obs$counts, est$counts), nrow = 2)
colnames(rval) <- obs$mids
rownames(rval) <- c("Observed", "Estimated")

barplot(rval, ylim = c(0, 100),
   xlab = "Incubation period (days)", ylab = "Frequency",
   col = c("red", "dark blue"), border = "gray")
legend("topright", legend = c("Observed", "Estimated"),
fill = c("red", "dark blue"), c("gray", "gray"), bty = "n")

## Epidemic curve computed using the augmented data:
fmd.inc <- inc.incubation(ips, meanlog = log(7), sdlog = log(1.5))
inf.date <- fmd.inc$inf.date
date.bins <- seq(from = as.Date("2001-02-01", format = "%Y-%m-%d"),
   to = as.Date("2001-03-31", format = "%Y-%m-%d"), by = "1 day")
hist(inf.date, breaks = date.bins, freq = TRUE)
```

`inc.local` *Local spread probabilities*

Description

This function returns data that can be used to estimate the probability of local spread as a function of time and distance from an infected source.

Usage

```
inc.local(ips, par, src.des, neighbours, relative.to = "signs",
   offset = -3)
```

Arguments

ips	a 7 column data frame listing listing details of infected places. Columns (in order): unique place identifier, place type, infected place number, easting coordinate of place centroid, northing coordinate of place centroid, infection date, onset of clinical signs date. There should be one row for each infected place.
par	a 6 column data frame listing details of the place population at risk. Columns (in order): unique place identifier, place type, easting coordinate of the place centroid, northing coordinate of the place centroid, date of slaughter, date cleaning and disinfection completed.
src.des	a 5 column data frame listing details where transmission of infection has occurred from a source place to a destination place by local spread. Columns (in order): unique identifier of source place, date of onset of signs on the source place, unique identifier of destination place, cause identifier, date of onset of signs on the destination place. See details for further information.
neighbours	a list of neighbours within a defined distance band of each place listed in par.
relative.to	a character string, indicating when infectivity starts on the source place. Options are signs: infectivity starts from the date of onset of clinical signs, infection: infectivity starts from the date of infection.
offset	scalar, indicating the offset to be used for defining the risk period for each place acting as an infected source. See details for further information.

Details

In the src.des data frame there is a one to many relationship between source and destination places. An infected source can have many destinations. A destination can receive infection from only one source.

Use the dnearneigh function in the spdep package to generate data for the neighbours list.

Setting relative.to to signs and offset to -3 sets the start of the risk period at three days before the date of onset of clinical signs on the source place. Infectivity of a source place ends on the date that animals on the source place were slaughtered.

Value

A data frame containing the following:

time	the number of time periods from the date of onset of infectivity of the source place.
n.risk	the number of places at risk of infection.
n.event	the number of places infected.

References

Sanson R, Stevenson M, Moles-Benfell N (2006). Quantifying local spread probabilities for foot-and-mouth disease. In: Proceedings of the 11th International Symposium on Veterinary Epidemiology and Economics. Cairns Convention Centre, Cairns, Australia.

See Also

inc.incubation

Examples

```
data(inc.outbreak)
ips <- inc.outbreak$ips[,c(1:5,8,9)]
par <- inc.outbreak$par
src.des <- inc.outbreak$tra
n0001 <- inc.outbreak$n0001

## Estimate missing infection dates:
tmp <- inc.incubation(inc.outbreak$ips[,c(1:3,8:10,13,16,19,22,25,28)],
   meanlog = log(7), sdlog = log(1.5))
ips$infdate <- tmp$inf.date

inc.local(ips = ips, par = par, src.des = src.des, neighbours = n0001,
   relative.to = "signs", offset = -3)
```

---

inc.network *Data from a hypothetical tracing exercise*

Description

Data from a hypothetical tracing exercise.

Usage

```
data(inc.network)
```

Format

A list comprised of objects `edg.dat` and `vert.dat`.

Object `edg.dat` provides details of transactions (that is, edges) between vertices in a network. Variables are: `srcherd` the identifier of the vertex acting as the source of the transaction, `desherd` the identifier of the vertex acting as the destination of the transaction, `mdate` the date on which the transaction occurred, and `n` an estimated risk score for the transaction.

Object `vert.dat` provides details of individual vertices participating in the network. Variables are `herd`: a unique vertex identifier, `xcoord`: the easting coordinate of the vertex centroid, and `ycoord`: the northing coordinate of the vertex centroid.

Details

Variable `mdate` in the `edg.dat` table is in `character` format. Easting and northing coordinates are in NZTM format.

Examples

```
data(inc.network)

plot(x = inc.network$vert.dat$xcoord / 1000,
   y = inc.network$vert.dat$ycoord / 1000,
   xlab = "Easting (km)", ylab = "No
   rthing (km)", pch = 16)
```

`inc.outbreak` *Data from a hypothetical outbreak of foot-and-mouth disease*

Description

Data from a hypothetical outbreak of foot-and-mouth disease in New Zealand.

Usage

```
data(inc.outbreak)
```

Format

A list comprised of objects `ips`, `par`, `tra`, and `n0001`. These data provide details of 122 farms infected with foot-and-mouth from a population of 28,291 farms.

Object `ips` provides summary information for the 122 infected places. Variables are `placeid`: unique place identifier, `placetype`: place type, `ipnumber`: infected place number, `x`: easting co-ordinate of the place centroid, `y`: northing coordinate of the place centroid, `pubrept`: identifies whether or not the infected place was reported by a member of the public (1 = public report, 0 = other), `cause`: estimated cause of infection, `infdate`: date of infection, `sgndate`: date of onset of clinical signs, and `visdate`: visit date. The next 18 variables provide the following details for pigs, dairy cattle, beef cattle, sheep, deer, and goats present on each infected place: the number of animals present (e.g. `nopig`), the number of animals initially infected (e.g. `inipig`), and the estimated age of lesions at time of examination (e.g. `agepig`).

Object `par` provides details of the premises at risk. Variables are `placeid`: unique place identifier, `placetype`: place type, `x`: easting coordinate of the place centroid, `y`: northing coordinate of the place centroid, `slgtdate` date of slaughter, `disdate` date cleaning and disinfection completed.

Object `tra` provides details of tracing events. Variables are `srcplaceid` unique place identifier of the source, `srcsgndate` date of onset of clinical signs on the source place, `desplaceid` unique place identifier of the destination, `causeid` cause identifier, `dessgndate` date of onset of clinical signs on the destination place, `startdate` date event started, and `enddate` date event ended.

Object `n0001` lists places that lie within 0 – 1000 metres of each of the 28,291 farms included in `par`. This object was produced using the `dnearneigh` in the `spdep` package.

Details

Dates are in `Date` format. Easting and northing coordinates are in NZTM format.

The NZMG projection (EPSG code 27200) is based on the New Zealand geodetic datum 1949 which uses the 1924 international ellipsoid. The NZTM projection (EPSG code 2193) is based on the New Zealand geodetic datum 2000 which uses the GRS80 ellipsoid (as used by WGS84). NZTM is the current standard for Biosecurity New Zealand.

See Also

`inc.incubation`, `inc.ffi`, `inc.fdi`, `inc.genint`

Examples

```
## Not run:
library(maptools); library(rgdal)
data(inc.outbreak)
ips <- inc.outbreak$ips[,1:9]

## Breakdown of infection explanations for identified infected places:
round(table(ips$cause) / sum(table(ips$cause)), digits = 2)
## Local spread accounted for 74% of all infected places.

## Map of New Zealand (NZTM2000) with infected place locations:
data(inc.outbreak)
ips <- inc.outbreak$ips[,1:5]

shapefile <- paste(.libPaths(),
   "/incursion/extdata/NZcoast-NZTM2000.shp", sep = "")
nzcoast.nztm2000 <- readShapePoly(shapefile, proj4string=CRS("+init=epsg:2193"))

shapefile <- paste(.libPaths(),
   "/incursion/extdata/NZTLA-NZTM2000.shp", sep = "")
nztla.nztm2000 <- readShapePoly(shapefile, proj4string=CRS("+init=epsg:2193"))

xylims <- attr(nzcoast.nztm2000, "bbox")
ratio <- (xylims[2,2] - xylims[2,1]) / (xylims[1,2] - xylims[1,1])

x.points <- seq(from = 1000000, to = 2000000, by = 2e05)
x.lab <- x.points / 1000
y.points <- seq(from = 4000000, to = 7000000, by = 2e05)
y.lab <- y.points / 1000

par(pin = c(3.5, ratio * 3.5), omi = c(0,0,0,0))
plot(x = xylims[1,], y = xylims[2,], xlab = "Easting (km)", type = "n",
   ylab = "Northing (km)", xaxt = "n", yaxt = "n", xlim = xylims[1,],
   ylim = xylims[2,], cex.lab = 1.00)
plot(nztla.nztm2000, border = "gray", add = TRUE)
points(ips$x, ips$y, type = "p", pch = 16, col = "red")
plot(nzcoast.nztm2000, border = "black", add = TRUE)
axis(side = 1, at = x.points, labels = x.lab, tick = TRUE, cex.axis = 0.80)
axis(side = 2, at = y.points, labels = y.lab, tick = TRUE, cex.axis = 0.80)

## Detailed map of infected place locations (NZTM2000):
id <- nztla.nztm2000$TA_NAME == "Auckland City"
tmp.nztm2000 <- nztla.nztm2000[id,]

xylims <- attr(tmp.nztm2000, "bbox")
ratio <- (xylims[2,2] - xylims[2,1]) / (xylims[1,2] - xylims[1,1])

x.points <- seq(from = 1000000, to = 2000000, by = 2e05)
x.lab <- x.points / 1000
y.points <- seq(from = 4000000, to = 7000000, by = 2e05)
y.lab <- y.points / 1000

par(pin = c(3.5, ratio * 3.5), omi = c(0,0,0,0))
plot(x = xylims[1,], y = xylims[2,], xlab = "Easting (m)", type = "n",
   ylab = "Northing (m)", xlim = xylims[1,], ylim = xylims[2,], cex.lab = 1.00)
plot(tmp.nztm2000, border = "black", add = TRUE)
```

```
points(ips$x, ips$y, type = "p", pch = 16, col = "red")

# Convert IP locations to a SpatialPoints object and set projection as NZTM2000:
ips.nztm2000 <- SpatialPoints(coords = cbind(ips$x, ips$y))
proj4string(ips.nztm2000) <- CRS("+init=epsg:2193")

# Re-project points to NZMG1949:
ips.nzmg1949 <- spTransform(ips.nztm2000, CRS("+init=epsg:27200"))

## Map of New Zealand (NZMG1949):
shapefile <- paste(.libPaths(),
   "/incursion/extdata/NZcoast-NZMG1949.shp", sep = "")
nzcoast.nzmg1949 <- readShapePoly(shapefile,
   proj4string=CRS("+init=epsg:27200"))

shapefile <- paste(.libPaths(),
   "/incursion/extdata/NZTLA-NZMG1949.shp", sep = "")
nztla.nzmg1949 <- readShapePoly(shapefile,
   proj4string=CRS("+init=epsg:27200"))

xylims <- attr(nzcoast.nzmg1949, "bbox")
ratio <- (xylims[2,2] - xylims[2,1]) / (xylims[1,2] - xylims[1,1])

x.lab <- seq(from = 2000000/1000, to = 3000000/1000, by = 1e05/1000)
x.points <- seq(from = 2000000, to = 3000000, by = 1e05)
y.lab <- seq(from = 5500000/1000, to = 7500000/1000, by = 1e05/1000)
y.points <- seq(from = 5500000, to = 7500000, by = 1e05)

par(pin = c(3.5, ratio * 3.5), omi = c(0,0,0,0))
plot(x = xylims[1,], y = xylims[2,], xlab = "Easting (km)", type = "n",
   ylab = "Northing (km)", xaxt = "n", yaxt = "n", xlim = xylims[1,],
   ylim = xylims[2,], cex.lab = 1.00)
plot(nztla.nzmg1949, border = "gray", add = TRUE)
plot(ips.nzmg1949, pch = 16, col = "red", add = TRUE)
plot(nzcoast.nzmg1949, border = "black", add = TRUE)
axis(side = 1, at = x.points, labels = x.lab, tick = TRUE, cex.axis = 0.80)
axis(side = 2, at = y.points, labels = y.lab, tick = TRUE, cex.axis = 0.80)

## End(Not run)
```

`inc.ripratio` *At-risk infected place ratio*

Description

Ratio of the number of places exposed to movements off identified infected places to the number of identified infected places for a given time frame.

Usage

```
inc.ripratio(ips, tra, start, period = 14, conf.level = 0.95)
```

Arguments

ips	a 4 column data frame listing details of infected places. Columns (in order): unique place identifier, place type, infected place number, and onset of clinical signs date. There should be one row for each infected place.
tra	a 7 column data frame listing details of tracing events. Columns (in order): source place unique identifier, date of onset of signs on the source place, destination place unique identifier, date of onset of signs on the destination place, episode start date (the date that contact between the two places commenced), and episode end date (the date that contact between the two places ceased).
start	the start date for the analysis.
period	the length of time (in days) to be analysed.
conf.level	scalar, defining the magnitude of the returned confidence interval. Must be a single number between 0 and 1.

Details

When an epidemic commences there are no controls in place and the disease has the potential to spread extensively by the time the index premises is identified. The number of premises with direct or indirect contact with the index premise(s) will have direct bearing on the number of newly infected premises and hence the estimated dissemination ratio.

In this function the numerator is the total number of premises exposed to direct or indirect movements off these identified infected premises. The denominator is the total number of infected premises identified in the specified time frame. If `start` equals 22 Feb 2001 and `period` equals 14 days then the time frame is from 22 Feb 2001 to 7 Mar 2001 (inclusive).

Confidence intervals are calculated using the method of Dobson et al. (1991).

Value

Returns the point estimate and confidence interval of the at-risk infected place ratio.

References

Dobson AJ, Kuulasmaa K, Eberle E, Scherer J (1991). Confidence intervals for weighted sums of Poisson parameters. Statistics in Medicine 10: 457 - 462.

Examples

```
data(inc.outbreak)
ips <- inc.outbreak$ips[,c(1:3,9)]
tra <- inc.outbreak$tra

start <- as.Date("2001-02-22", format = "%Y-%m-%d")
inc.ripratio(ips, tra, start, period = 14, conf.level = 0.95)

## The at-risk infected place ratio for the period 22 February 2001 to
## 7 March 2001 (inclusive) was 0.16 (95% CI 0.09 -- 0.27).
```

`inc.selfsurv` *Ratio of public reported diagnoses to surveillance activity diagnoses*

Description

Ratio of public reported diagnoses to surveillance activity diagnoses for a given time frame.

Usage

```
inc.selfsurv(ips, start, period = 14, conf.level = 0.95)
```

Arguments

`ips`	a 5 column data frame listing details of infected places. Columns (in order): unique place identifier, place type, infected place number, public report flag (0 or 1), onset of clinical signs date. There should be one row for each infected place.
`start`	the start date for the analysis.
`period`	the length of time (in days) to be analysed.
`conf.level`	scalar, defining the magnitude of the returned confidence interval. Must be a single number between 0 and 1.

Details

In a well managed outbreak most infected places will be identified by surveillance activities conducted by the state veterinary service. The ratio of infected places brought to the attention of authorities by members of the public (i.e. 'surprise' infections) to those known to be at risk provides a measure of the effectiveness of epidemic control and eradication measures. Ideally, this ratio should be close to zero at all times.

If `start` equals 22 Feb 2001 and `period` equals 14 days then the time frame is from 22 Feb 2001 to 7 Mar 2001 (inclusive).

Confidence intervals are calculated using the method of Dobson et al. (1991).

Value

Returns the point estimate and confidence interval of ratio of public reported diagnoses to surveillance activity diagnoses.

References

Dobson AJ, Kuulasmaa K, Eberle E, Scherer J (1991). Confidence intervals for weighted sums of Poisson parameters. Statistics in Medicine 10: 457 - 462.

Examples

```
data(inc.outbreak)
ips <- inc.outbreak$ips[,c(1:3,6,9)]

start <- as.Date("2001-02-22", format = "%Y-%m-%d")
inc.selfsurv(ips, start, period = 14, conf.level = 0.95)
```

```
## For the period 22 February 2001 to 7 March 2001 (inclusive) the ratio of
## public reported diagnoses to surveillance activity diagnoses was 0.14
## (95% CI 0.07 -- 0.24).
```

---

inc.sna *Manipulate network data in edgelist format*

Description

Convert network data in edgelist format to matrix format, ready to be read by programs such as Ucinet, SocNetV and a geographic information system.

Usage

```
inc.sna(edge, vertex, fn = "SNA", shapefile = TRUE, socnetv = TRUE,
   ucinet = TRUE, dichotomise = TRUE)
```

Arguments

edge	a 4 column data frame listing details of movment events. Columns (in order): unique identifier of source, unique identifier of destination, date of transaction, estimated risk score of transaction.
vertex	a data frame of vertex attributes. There must be at least three columns: unique identifier of vertex, easting coordinate of vertex, and northing coordinate of vertex.
fn	character string, prefix for output file names.
shapefile	logical, is function to return an ESRI shapefile?
socnetv	logical, is function to return a matrix file suitable for eading into SocNetV?
ucinet	logical, is function to return a matrix file suitable for eading into Ucinet?
dichotomise	logical, dichotomise the matrix?

Details

Unique vertex identifiers can be either character strings or numbers, or combinations of strings and numbers.

Value

Returns one or more of three file formats: an ESRI shapefile (`*.shp, *.shx, *.dbf`), a `*.csv` suitable for reading into SocNetV, a `*.csv` file suitable for reading into Ucinet.

When `shapefile = TRUE` a vertex (point) and edge (linear) shape file are returned. The vertex shape file lists the identity of each vertex as well as its easting and northing coordinate. The edge shape file lists the identity of the source and destination vertices as well as the date of the recorded transaction.

When `ucinet = TRUE` the first column of the returned attribute file is `nid` specifying the 1 to `n` identifier of the vertices of the network. The original identifiers of the network vertices are listed in the second column. These details are written out to the attribute table of the ESRI shapefile providing the vertex details.

Examples

```
data(inc.network)

inc.sna(edge = inc.network$edg.dat, vertex = inc.network$vert.dat,
   fn = "SNA", shapefile = TRUE, socnetv = TRUE, ucinet = TRUE,
   dichotomise = TRUE)
```

`inc.stkml` *Export point locations and dates of events to a space-time KML file*

Description

Export point locations and dates of events to a space-time KML (keyhole markup language) file.

Usage

```
inc.stkml(dat, file, shape, colour, labels, kmz = FALSE)
```

Arguments

`dat`	a six-column data frame listing (in order): a unique identifier for the event, the event date, the latitude of the event (in decimal degrees), the longitude of the event (in decimal degrees), a text label to defining the type of event, and a text label defining the event subtype (see details for further information).
`file`	a character string specifying the name of the output KML file.
`shape`	a character string specifying the type of icon to be used in the output KML file.
`colour`	a vector of character strings defining the colour (in KML format — see examples below) of the icon to be used at each point location.
`labels`	a vector of character strings defining the text label to appear adjacent to each point location.
`kmz`	logical; if yes the output KML file will be compressed.

Details

Typically this function would be used used to create a space-time KML showing the date and location of foot-and-mouth disease outbreaks. Used in this way the fifth column of the input data frame specifying the event type would be a text string with `FMD` (or something similar) and the sixth column (specifying the event subtype) would be a text string listing one of `A`, `Asia 1`, `O` or `UNK` to represent the FMD subtype identified at the specified location on the specified date.

Value

A keyhole markup language (KML) file that can be loaded into Google Earth.

Examples

```
## Not run:
## Load the plotKML and spacetime packages:
library(plotKML); library(spacetime)

data(inc.outbreak)
dat <- data.frame(id = inc.outbreak$ips$placeid,
   infdate = inc.outbreak$ips$infdate, x = inc.outbreak$ips$x,
   y = inc.outbreak$ips$y, type = "FMD", subtype = "O")

# Make sure the infection date is in as.Date format:
dat$infdate <- as.Date(dat$infdate, format = "%Y-%m-%d")
class(dat$infdate)

# Drop observations with null dates:
id <- !is.na(dat$infdate)
dat <- dat[id,]

# Assign colours for KML file:
dat$kcol <- col2kml("yellow")

## Re-project the easting and northing coordinates (in NZTM projection) in the
## original data file to lat-lon:
dat.nzsp <- SpatialPoints(dat[,c(3,4)])
proj4string(dat.nzsp) <- CRS("+init=epsg:2193")
dat.llsp <- spTransform(dat.nzsp, CRS("+proj=longlat +datum=WGS84"))

ndat <- data.frame(id = dat$id, infdate = dat$infdate,
   lat = coordinates(dat.llsp)[,2], lon = coordinates(dat.llsp)[,1],
   type = dat$type, subtype = dat$subtype, kcol = dat$kcol)

## Write the FMD data set out to a space-time KML file.
shape <- "http://maps.google.com/mapfiles/kml/pal2/icon18.png"

inc.stkml(dat = ndat, file = "NZ_FMD.kml", shape = shape,
   colour = ndat$kcol, labels = ndat$subtype, kmz = FALSE)

## End(Not run)
```

---

inc.trace *Subset network data in edgelist format*

Description

Subset network data in edgelist format to include a specified vertex of interest and the vertices connected to it.

Usage

```
inc.trace(graph, nodes, order = 2, mode = "in")
```

Arguments

graph	an object of class igraph defining the vertex connections.
nodes	character scalar or character vector, specifying the individual vertex (or vertices) of interest.
order	numeric scalar defining the search distance. See details.
mode	character scalar. Options are in, out or all. Use out to return only the outgoing edges from the vertex (or vertices) of interest. Use in to return the edges incoming to the vertex (or vertices) of interest. Use all to return outgoing and incoming edges to the vertex (or vertices) of interest. This argument is ignored for undirected graphs.

Details

This function would typically be used for tracing movement events onto and off livestock premises in the event of an infectious disease incursion. The user would load movement event data into R and create an igraph object from that data. They would then specify a herd of interest and use the inc.trace function to take a subset of the entire network that includes the herd of interest and all herds connected to it. The order argument controls the number of connections to the herd of interest. Setting order = 2 includes the first and second order connections, that is herd directly connected to the herd of interest as well as the herds connected to those herds. Care is advised when setting the order argument. Setting it too high (for example, greater than 4) can result in a very large subset graph. The tkplot option, as shown in the example below, provides a convenient way to re-arrange the position of vertices on a network plot for clear display.

Value

Returns an igraph object which is a subset of the object graph.

Examples

```
data(inc.network)

index.herd <- "N094650321-1"
inc.network$vert.dat$col <- "light blue"
inc.network$vert.dat$col[inc.network$vert.dat$herd == index.herd] <- "red"

## Create a graph object from using the edge and vertex data:
tmp.edg <- graph.data.frame(d = inc.network$edg.dat, directed = TRUE,
   vertices = inc.network$vert.dat)
trace.edg <- inc.trace(graph = tmp.edg, nodes = index.herd, order = 2,
   mode = "in")

## Plot the subsetted network. The vertex of interest is coloured red. Date
## and direction of transactions between vertices are labelled:
plot(trace.edg, vertex.size = 10, vertex.label.cex = 0.75,
   vertex.color = get.vertex.attribute(graph = trace.edg, name = "col"),
   vertex.frame.color = "transparent",
   edge.arrow.size = 0.50,
   edge.color = "black",
   edge.label = get.edge.attribute(graph = trace.edg, name = "mdate"),
   edge.label.cex = 0.60)

## The same, using using the tkplot function in package igraph:
tkplot(trace.edg,
```

```
edge.label = get.edge.attribute(graph = trace.edg, name = "mdate"),
vertex.color = get.vertex.attribute(graph = trace.edg, name = "col"))
```

Index