

Package ‘geomnet’

December 8, 2016

Type Package

Title Network Visualization in the 'ggplot2' Framework

Version 0.2.0

Date 2016-11-14

Author Samantha Tyner, Heike Hofmann

Maintainer Samantha Tyner <sctyner@iastate.edu>

Description Network visualization in the 'ggplot2' framework. Network functionality is provided in a single 'ggplot2' layer by calling the geom 'net'. Layouts are calculated using the 'sna' package, example networks are included.

License GPL (>= 2)

URL <http://github.com/sctyner/geomnet>

BugReports <https://github.com/sctyner/geomnet/issues>

LazyData TRUE

Depends R (>= 2.14), ggplot2 (>= 2.2.0)

Imports sna, network, dplyr, tidyr, readr, plotly

Suggests ggrepel, knitr, igraph, ggmap

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-12-08 20:38:18

R topics documented:

as.adjmat	2
as.edgedf	2
bikes	3
blood	4
email	4
football	5

fortify.adjmat	7
fortify.edgedf	7
fortify.igraph	8
fortify.network	9
GeomCircle	9
geom_net	11
jtt	15
lesmis	16
madmen	17
metro_map	18
mm.directed	18
protein	19
soccer	20
theme_elements	21
theme_net	21

Index 22

as.adjmat	<i>Cast matrix to an adjacency matrix</i>
-----------	---

Description

Create and assign a new class for use of geomnet's [fortify.adjmat](#) function.

Usage

```
as.adjmat(dat)
```

Arguments

dat	A square adjacency matrix.
-----	----------------------------

as.edgedf	<i>Cast a data frame to an edgedf</i>
-----------	---------------------------------------

Description

Create and assign a new class for use of geomnet's [fortify.edgedf](#) function.

Usage

```
as.edgedf(dat)
```

Arguments

dat	A network edgelist of class "data.frame" for use with fortify.edgedf .
-----	--

bikes	<i>Bike sharing network (directed)</i>
-------	--

Description

This network is a summary of the bike trips taken by customers of the bike sharing company Capital Bikeshare (<https://secure.capitalbikeshare.com/>) during the second quarter of 2015. Only trips between stations in the vicinity of Rockville, MD, are included. The data is organized as a list of two datasets, vertices (stations) and edges (trips between stations), as follows:

Usage

bikes

Format

A list of two data frames:

- the trips data set consists of four variables of length 53:
 - Start.station: Station where bike trip starts
 - End.station: Station where bike trip ends
 - n: Number of trips between the two stations
 - minlength: Duration of shortest trip between the two stations (in seconds). Only those stations are included, if the shortest trip between them lasted not more than 15 minutes.
- the vertices data set consists of five variables with information on 21 stations:
 - id: Station ID number
 - name: Station name
 - lat: Latitude of station location
 - long: Longitude of station location
 - nbDocks: Number of bike docks at the station

References

<https://secure.capitalbikeshare.com/>

blood	<i>Network of blood types (directed)</i>
-------	--

Description

A list of two datasets, vertices and edges, containing information on blood type (see <http://www.redcrossblood.org/learn-about-blood/blood-types>). The variables are as follows:

Usage

blood

Format

A list of two data frames:

- the edges data set consists of three variables of length 27:
 - from, to: factor variables of blood types describing the relationship 'is compatible with'
 - group_to: factor variable with levels 'same' and 'diff' for same or different blood type group not regarding the rho-gam factor.
- the vertices data set consists of five variables and 32 rows:
 - label: factor variable of blood types,
 - type: factor variable of blood type, not regarding the rhesus factor,
 - rho: factor variable: 'pos' and 'neg' describing the rhesus factor,
 - Ethnicity: factor variable of four variables: 'Caucasians', 'African.American', 'Hispanic', and 'Asian',
 - Predominance: numeric variable consisting of the percentage points of each blood type within each ethnicity.

email	<i>Email network (directed)</i>
-------	---------------------------------

Description

A list of two datasets, vertices and edges, containing data on employees and (fictitious) email headers from two weeks of internal GASTech company email between employees made available as part of the VAST challenge 2014 (K. Cook, G. Grinstein, M. Whiting, see <http://hci12.cs.umd.edu/newwarepository/benchmarks.php>). The variables are as follows:

Usage

email

Format

A list of two data frames:

- the edges data set consists of ten variables of length 9063:
 - From: Email address of sender
 - eID: email ID. If an email was sent to multiple recipients, the email ID is the same.
 - Date: Date and time email was sent
 - Subject: Subject line of email
 - to: Email address of recipient
 - month: Month email was sent
 - day: Day of month email was sent
 - year: Year email was sent
 - nrecipients: Number of recipients of email
- the nodes data set consists of 18 variables with information on 55 employees:
 - label: Employee's email address
 - LastName: Employee's last name
 - FirstName: Employee's first name
 - BirthDate: Employee's first date
 - BirthCountry: Employee's (fictional) country of birth
 - Gender: Employee's gender
 - CitizenshipCountry: Employee's (fictional) country of citizenship
 - CitizenshipBasis: Is citizenship by birth or parents?
 - CitizenshipStartDate: When employee became a citizen
 - PassportCountry: (Fictional) country issuing employee's passport
 - PassportIssueDate: Date employee received passport
 - PassportExpirationDate: Date employee's passport expires
 - CurrentEmploymentType: Employee's department in the company
 - CurrentEmploymentTitle: Employee's title at the company
 - CurrentEmploymentStartDate: Date employee started at position
 - MilitaryServiceBranch: Branch of the (fictional) military in which the employee serves
 - MilitaryDischargeType: General or honorable discharge from military service?

football

College football games network (undirected)

Description

A list of two datasets, vertices and edges, containing data on Division I college football games in the Fall 2000 season. The variables are as follows:

Usage

football

Format

A list of two data frames:

- the edges data set consists of three variables of length 613:
 - from, to: Character variables describing the teams playing in the game
 - same.conf: An indicator variable that is 1 if the two teams are in the same conference and 0 otherwise.
- the vertices data set consists of two variables with information on 115 Division I schools:
 - label: Character variable containing the school names
 - value: Character variable containing the conference of the schools

References

M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. USA 99, 7821-7826 (2002).

Examples

```
# data step: merge vertices and edges
ftnet <- merge(
  football$edges, football$vertices,
  by.x = "from", by.y = "label", all = TRUE
)

# label independent schools
ftnet$schools <- ifelse(ftnet$value == "Independents", ftnet$from, "")

library(geomnet)
library(dplyr)
# create data plot
ggplot(data = ftnet,
  aes(from_id = from, to_id = to)) +
  geom_net(
    aes(
      colour = value, group = value,
      linetype = factor(1-same.conf),
      label = schools
    ),
    linewidth = 0.5,
    size = 5, vjust = -0.75, alpha = 0.3,
    layout.alg = 'fruchtermanreingold'
  ) +
  theme_net() +
  theme(legend.position = "bottom") +
  scale_colour_brewer("Conference", palette = "Paired")
```

fortify.adjmat	<i>Function for converting a network adjacency matrix into the correct format for use with geomnet</i>
----------------	--

Description

Function for converting a network adjacency matrix into the correct format for use with geomnet

Usage

```
## S3 method for class 'adjmat'  
fortify(model, data = NULL, ...)
```

Arguments

model	An adjacency matrix of class "adjmat".
data	not used in this function
...	not used in this function

Examples

```
data(emon, package = "network")  
adjmat <- as.adjmat(network::as.matrix.network.adjacency(emon$MtSi))  
str(adjmat)  
fortify(adjmat)
```

fortify.edgedf	<i>Function for converting a network edge list in data frame form into the correct format for use with geomnet</i>
----------------	--

Description

Function for converting a network edge list in data frame form into the correct format for use with geomnet

Usage

```
## S3 method for class 'edgedf'  
fortify(model, data, group = NULL, ...)
```

Arguments

model	A network edgelist of class "edgedf". See as.edgedf . Can contain edge variables as well.
data	Data frame containing network node list and other node information. First column should contain node ids.
group	character. Used for faceting. If you wish to facet on network variable provide the name of that variable here.
...	not used in this function

Examples

```
data(blood)
fortify(as.edgedf(blood$edges), blood$vertices)
fortify(as.edgedf(blood$edges), blood$vertices, group = "Ethnicity")
```

fortify.igraph	<i>Function for converting an igraph object into the correct format for use with geomnet</i>
----------------	--

Description

Function for converting an igraph object into the correct format for use with geomnet

Usage

```
## S3 method for class 'igraph'
fortify(model, data = NULL, group = NULL, ...)
```

Arguments

model	A network object of class "igraph".
data	NULL - not used in this function
group	character. Used for faceting. If you wish to facet on a network variable provide the name of that variable here.
...	not used in this function

Examples

```
# class igraph (igraph, igraphdata packages)
library(igraph)
data(blood, package = "geomnet")
blood.igraph <- graph_from_data_frame(d = blood$edges,
  directed = TRUE, vertices = unique(blood$vertices[, 1:3]))
fortify(blood.igraph)
```

fortify.network	<i>Function for converting a network object into the correct format for use with geomnet</i>
-----------------	--

Description

Function for converting a network object into the correct format for use with geomnet

Usage

```
## S3 method for class 'network'  
fortify(model, data = NULL, group = NULL, ...)
```

Arguments

model	object of class "network"
data	NULL - not used in this function
group	character. Used for faceting. If you wish to facet on a network variable provide the name of that variable here.
...	not used in this function

Examples

```
# class network (sna, network, statnet packages)  
  
library(network)  
data(emon, package = "network")  
fortify(emon$Cheyenne)
```

GeomCircle	<i>Geom for drawing circles in the ggplot2 framework</i>
------------	--

Description

Circles are drawn with a specified radius centered at (x, y). This geom is very much exploratory - we are using it for drawing edges for self references. It is not explored for any more general use, so use with caution!

Usage

```
GeomCircle  
  
geom_circle(mapping = NULL, data = NULL, stat = "identity",  
            position = "identity", na.rm = FALSE, show.legend = NA,  
            inherit.aes = TRUE, radius = 0.05, ...)
```

Arguments

mapping	Set of aesthetic mappings created by aes or aes_ . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code> , and will be used as the layer data.
stat	The statistical transformation to use on the data for this layer, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders .
radius	numeric value giving the radius of the circle to be drawn (0-1 normalized scale)
...	other arguments passed on to layer . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>color = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.

Format

An object of class `GeomCircle` (inherits from `Geom`, `ggproto`) of length 5.

Examples

```
# circles are drawn centered at x and y
library(ggplot2)
data(mpg)
ggplot(mpg, aes(displ, hwy)) + geom_circle(radius=0.1) + geom_point()
ggplot(mpg, aes(displ, hwy)) + geom_circle(linetype=2, radius=0.05, alpha=0.5)
ggplot(mpg, aes(displ, hwy)) + geom_circle(aes(linetype=factor(cyl)), radius=0.05, alpha=0.5)
```

geom_net

Networks

Description

The net geom is used visualize networks within the **ggplot2** framework. `geom_net` combines the many parts of a network visualization into a single layer in **ggplot2**. It makes use of various other geoms, including but not limited to, `geom_point`, `geom_segment`, and `geom_text`.

Usage

```
geom_net(mapping = NULL, data = NULL, stat = "net",
         position = "identity", show.legend = NA, na.rm = FALSE,
         inherit.aes = TRUE, layout.alg = "kamadakawai", layout.par = list(),
         directed = FALSE, fiteach = FALSE, selfloops = FALSE,
         singletons = TRUE, alpha = 0.25, ecolour = NULL, ealpha = NULL,
         arrow = NULL, arrowgap = 0.01, arrowsize = 1, labelon = FALSE,
         labelcolour = NULL, labelgeom = "text", repel = FALSE,
         vertices = NULL, ...)
```

GeomNet

StatNet

```
stat_net(mapping = NULL, data = NULL, geom = "net",
         position = "identity", show.legend = NA, inherit.aes = TRUE,
         layout.alg = "kamadakawai", layout.par = list(), fiteach = FALSE,
         vertices = NULL, singletons = TRUE, na.rm = FALSE, ...)
```

Arguments

mapping	Set of aesthetic mappings created by <code>aes</code> or <code>aes_</code> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to <code>ggplot</code> . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify</code> for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code> ., and will be used as the layer data.
stat	The statistical transformation to use on the data for this layer, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.

show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders .
layout.alg	character. Value specifying the layout algorithm to use. Defaults to "kamadakawai". See ?sna::gplot.layout documentation for more choices.
layout.par	list. Parameters detailing algorithmic specs. Default parameters from sna are used initially. See ?sna::gplot.layout documentation for all options corresponding to all layouts.
directed	logical value. Should an arrow be drawn pointing to the to_id node? Default is FALSE.
fiteach	logical. Should the network be fit in each of the panels separately, or is there going to be one fit for all?
selfloops	logical value. Should loops (self-referencing edges) be shown (by drawing a circle adjacent to the corresponding node)? Default is FALSE.
singletons	logical. Should singletons (nodes with no incoming or outgoing ties) be plotted? Default is TRUE.
alpha	numeric. Value from 0-1 of alpha blending of nodes.
ecolour	character. Colour for edges.
ealpha	numeric. Value from 0-1 of alpha blending of edges.
arrow	what kind of arrow should be drawn? See ?grid::arrow for more.
arrowgap	numeric value between 0 and 1 specifying how much (as a proportion of the line length) earlier the line segment should be stopped drawing before reaching the target node. This parameters is only regarded in directed networks.
arrowsize	numeric. How big should the arrow be drawn? Multiplicative of the default, 10 points.
labelon	logical. Include labels for all nodes. Labels are taken from the from_id variable, unless a label aesthetic is provided.
labelcolour	character. Colour for the labels. If this argument is not specified, labels have the same colour as the nodes.
labelgeom	character. Which ggplot2 geom should be used to draw the labels? Either "text" or "label". Default is "text"
repel	logical. If TRUE, uses the ggrepel package geoms to draw the node labels instead of the ggplot2 geoms.
vertices	data frame. Dataset containing vertex information. Usage is a bit awkward, because every variable in this data set can only be used with the ggplot2 double dot representation <code>..varname..</code> . Better: use the fortify.edgedf method
...	other arguments passed on to layer . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>color = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.
geom	<code>geom_net</code> , the geom attached to <code>stat_net</code> is called "net".

Format

An object of class `GeomNet` (inherits from `Geom`, `ggproto`) of length 6.

Value

A data frame with additional columns:

<code>x</code> , <code>y</code>	coordinates of the nodes, beginning of edges,
<code>xend</code> , <code>yend</code>	coordinates end points of edges.

Aesthetics

`geom_net` understands the following aesthetics (required aesthetics are in bold):

- `from_id`
- `to_id`
- `x`
- `y`
- `alpha`
- `colour`
- `fontsize`
- `group`
- `label`
- `linetype`
- `linewidth`
- `shape`
- `size`

Examples

```
## Not run:
library(geomnet)
data(blood)
p <- ggplot(data = blood$edges, aes(from_id = from, to_id = to))
p + geom_net(vertices=blood$vertices, aes(colour=..type..)) + theme_net()

bloodnet <- fortify(as.edgdf(blood$edges), blood$vertices)
p <- ggplot(data = bloodnet, aes(from_id = from_id, to_id = to_id))
p + geom_net()
p + geom_net(aes(colour=rho)) + theme_net()
p + geom_net(aes(colour=rho), labelon=TRUE, vjust = -0.5)
p + geom_net(aes(colour=rho, linetype = group_to, label = from_id),
             vjust=-0.5, labelcolour="black", directed=TRUE) +
  theme_net()
p + geom_net(colour = "orange", layout.alg = 'circle', size = 6)
p + geom_net(colour = "orange", layout.alg = 'circle', size = 6, linewidth=.75)
p + geom_net(colour = "orange", layout.alg = 'circle', size = 0, linewidth=.75,
```

```

    directed = TRUE)
p + geom_net(aes(size=Predominance, colour=rho, shape=rho, linetype=group_to),
  linewidth=0.75, labelon =TRUE, labelcolour="black") +
  facet_wrap(~Ethnicity) +
  scale_colour_brewer(palette="Set2")
gg <- ggplot(data = blood$edges, aes(from_id = from, to_id = to)) +
  geom_net(colour = "darkred", layout.alg = "circle", labelon=TRUE, size = 15,
    directed = TRUE, vjust = 0.5, labelcolour = "grey80",
    arrowsize = 1.5, linewidth = 0.5, arrowgap = 0.05,
    selfloops = TRUE, ecolour = "grey40") +
  theme_net()
gg
dframe <- ggplot_build(gg)$data[[1]] # contains calculated node and edge values

#Madmen Relationships
data(madmen)
MMnet <- fortify(as.edgedf(madmen$edges), madmen$vertices)
p <- ggplot(data = MMnet, aes(from_id = from_id, to_id = to_id))
p + geom_net(labelon=TRUE)
p + geom_net(aes(colour=Gender), size=6, linewidth=1, labelon=TRUE, fontsize=3, labelcolour="black")
p + geom_net(aes(colour=Gender), size=6, linewidth=1, labelon=TRUE, labelcolour="black") +
  scale_colour_manual(values=c("#FF69B4", "#0099ff")) + xlim(c(-.05,1.05))
p + geom_net(aes(colour=Gender), size=6, linewidth=1, directed=TRUE, labelon=TRUE,
  arrowgap=0.01, labelcolour="black") +
  scale_colour_manual(values=c("#FF69B4", "#0099ff")) + xlim(c(-.05,1.05))

p <- ggplot(data = MMnet, aes(from_id = from_id, to_id = to_id))
# alternative labelling: specify label aesthetic.
p + geom_net(aes(colour=Gender, label=Gender), size=6, linewidth=1, fontsize=3,
  labelcolour="black")

## visualizing ggplot2 theme elements
data(theme_elements)
TENet <- fortify(as.edgedf(theme_elements$edges[,c(2,1)]), theme_elements$vertices)
ggplot(data = TENet, aes(from_id = from_id, to_id = to_id)) +
  geom_net(labelon=TRUE, vjust=-0.5)

## emails example from VastChallenge 2014
# care has to be taken to make sure that for each panel all nodes are included with
# the necessary information.
# Otherwise line segments show on the plot without nodes.
emailedges <- as.edgedf(subset(email$edges, nrecipients < 54))
emailnet <- fortify(emailedges, email$nodes)
#no facets
ggplot(data = emailnet, aes(from_id = from_id, to_id = to_id)) +
  geom_net(aes(colour= CurrentEmploymentType), linewidth=0.5) +
  scale_colour_brewer(palette="Set2")
#facet by day

emailnet <- fortify(emailedges, email$nodes, group = "day")
ggplot(data = emailnet, aes(from_id = from, to_id = to_id)) +
  geom_net(aes(colour= CurrentEmploymentType), linewidth=0.5, fiteach=TRUE) +
  scale_colour_brewer(palette="Set2") +

```

```

    facet_wrap(~day, nrow=2) + theme(legend.position="bottom")
  ggplot(data = emailnet, aes(from_id = from, to_id = to_id)) +
    geom_net(aes(colour= CitizenshipCountry), linewidth=0.5, fiteach=TRUE) +
    scale_colour_brewer(palette="Set2") +
    facet_wrap(~day, nrow=2) + theme(legend.position="bottom")
  ggplot(data = emailnet, aes(from_id = from, to_id = to_id)) +
    geom_net(aes(colour= CurrentEmploymentType), linewidth=0.5, fiteach=FALSE) +
    scale_colour_brewer(palette="Set2") +
    facet_wrap(~day, nrow=2) + theme(legend.position="bottom")

## Les Miserables example
data(lesmis)
lesmisnet <- fortify(as.edgedf(lesmis$edges), lesmis$vertices[, c(2,1)])
p <- ggplot(data=lesmisnet, aes(from_id=from_id, to_id=to_id))
p + geom_net(layout.alg="fruchtermanreingold")
p + geom_net(layout.alg="fruchtermanreingold", labelon=TRUE, vjust=-0.5)
p + geom_net(layout.alg="fruchtermanreingold", labelon=TRUE, vjust=-0.5,
  aes(linewidth=degree/5))

## College Football Games in the Fall 2000 regular season
# Source: http://www-/personal.umich.edu/~mejn/netdata/
data(football)
ftnet <- fortify(as.edgedf(football$edges), football$vertices)
p <- ggplot(data=ftnet, aes(from_id=from_id, to_id=to_id))
p + geom_net(aes(colour=value), linewidth=0.75, size=4.5, ecolour="grey80") +
  scale_colour_brewer("Conference", palette="Paired") + theme_net() +
  theme(legend.position="bottom")

## End(Not run)

```

jtt

Number of JTTs in a graph

Description

Number of jumping transitive triplets (JTT) in a graph. A JTT between three nodes i, j , and k is defined as the situation that when there is a (directed) edge from i to j and an edge from j to k there is also a direct edge from i to k . In an undirected situation we can think of any undirected edge as two directed edges between the two nodes involved.

Usage

```
jtt(data, from_id, to_id)
```

Arguments

data	data set
from_id	name of the variable of edge tails
to_id	name of the variable of edge heads

Examples

```
data(blood)
ggplot(data= blood$edges) + geom_net(aes(from_id=from, to_id=to), directed=TRUE) + theme_net()
jtt(blood$edges, "from", "to")
# this number is very high compared to the overall number of edges that are not self-loops
nrow(subset(blood$edges, from != to))
```

lesmis

Coappearance network of characters in Les Miserables (undirected)

Description

A list of two datasets, vertices and edges, containing data on characters and their coappearance in chapters in Victor Hugo's Les Miserables. The variables are as follows:

Usage

```
lesmis
```

Format

A list of two data frames:

- the edges data set consists of three variables of length 254:
 - from: Character 1
 - to: Character 2
 - degree: number of times they appear together in a chapter of Les Miserables
- the vertices data set consists of two variables with information on 77 characters:
 - id: Character ID number
 - label: Character name

References

D. E. Knuth, The Stanford GraphBase: A Platform for Combinatorial Computing, Addison-Wesley, Reading, MA (1993).

Examples

```
# prep the data
lesmisnet <- merge(lesmis$edges, lesmis$vertices, by.x = "from",
                  by.y = "label", all = TRUE)
lesmisnet$degree[is.na(lesmisnet$degree)] <- 0

# create plot
library(geomnet)
library(dplyr)
```



```
ggplot(data = lesmisnet, aes(from_id = from, to_id = to,
                             linewidth = degree / 5 + 0.1 )) +
  geom_net(aes(size = degree, alpha = degree),
           colour = "grey30", ecolour = "grey60",
           layout.alg = "fruchtermanreingold", labelon = TRUE, vjust = -0.75) +
  scale_alpha(range = c(0.3, 1)) +
  theme_net()
```

madmen	<i>Network of romantic relationships in the TV show Mad Men (undirected)</i>
--------	--

Description

A list of two datasets, vertices and edges, containing information on sexual relations in the TV show Mad Men. This data set was first compiled by Winston Chang for the package gcookbook (under the same name) and was extended here to include the gender of each of the characters. The variables are as follows:

Usage

```
madmen
```

Format

A list of two data frames:

- the edges data set consists of two variables of length 39:
 - Name1, Name2: Factor variables containing names of characters with a sexual relationship
- the vertices data set consists of three variables with information on 45 characters of the show:
 - label: Factor variable with name of the character,
 - Gender: Factor variable of the gender of the character.

References

Winston Chang. 2013. R Graphics Cookbook. O'Reilly Media, Inc..

`metro_map`*Map of Washington DC Metro area*

Description

A dataset containing information to draw a map of Rockville, MD, and vicinity using ggmap. This information was pulled from Google Maps using the **ggmap** package.

Usage`metro_map`**Format**

An object of class "ggmap" and "raster" containing a map of the Rockville, MD, area.

References

D. Kahle and H. Wickham. ggmap: Spatial Visualization with ggplot2. The R Journal, 5(1), 144-161. <https://journal.r-project.org/archive/2013-1/kahle-wickham.pdf>

Examples

```
## Not run:  
library(ggmap)  
data(metro_map)  
ggmap(metro_map)  
  
## End(Not run)
```

`mm.directed`*A directed network of Mad Men relationships*

Description

A list of two datasets, vertices and edges, containing information on sexual advances made in the TV show Mad Men. This data set was first compiled by Winston Chang for the package gcookbook (under the name madmen2) and was extended here to include the gender of each of the characters. The variables are as follows:

Usage`mm.directed`

Format

A list of two data frames:

- the edges data set consists of two variables of length 87:
 - Name1: Character variable with name of the character who made a sexual advance toward the character in Name2
 - Name2: Character variable with name of the character receiving, not necessarily reciprocating, Name1's advance.
- the vertices data set consists of two variables with information on 52 characters:
 - label: Factor variable with name of the character
 - Gender: Factor variable with gender of the character

References

Winston Chang (2012). `gcookbook`: Data for "R Graphics Cookbook". R package version 1.0. <https://CRAN.R-project.org/package=gcookbook>

protein	<i>A protein interaction network (undirected)</i>
---------	---

Description

A list of two datasets, vertices and edges, containing data on the complete protein-protein interaction network in the yeast species *S. cerevisiae* (<http://www3.nd.edu/~networks/resources.htm>). The variables are as follows:

Usage

```
protein
```

Format

A list of two data frames:

- the edges data set consists of two variables of length 4480:
 - from, to: Integer variables describing interactions between proteins, which are identified integers
- the vertices vector consists of the corresponding IDs of 2113 proteins in the edges data set

References

H. Jeong, S. Mason, A.L. Barabasi and Z.N. Oltvai, Centrality and lethality of protein networks Nature 411, 41 (2001)

 soccer

High school boys' soccer games in Iowa (undirected)

Description

A list of two datasets, vertices and edges, containing data on boys' soccer games at Iowa high schools in the 2011-2014 seasons. This dataset was compiled by Danny Bero (<bero.danny@gmail.com>). The variables are as follows:

Usage

```
soccer
```

Format

A list of two data frames:

- the edges data set consists of 11 variables of length 4484:
 - sub: Integer variable with the subregion of the state the school is in (1-8)
 - home: Factor variable containing the home team school
 - season: Integer variable with the year the game was played
 - week: Integer variable with the week of the season the game was played
 - date: Factor variable containing the date the game was played
 - away: Factor variable containing the away team school
 - ha: Factor variable stating if the first team is the home team, away team, or neutral?
 - result: Factor variable stating if the home team won, lost, or tied?
 - score: Factor variable with game final score, home team first
 - diff: Integer variable with home team score minus away team score
 - same_div: An indicator variable that is 1 if the schools are in the same division and 0 otherwise
- the vertices data set consists of two variables with information on 157 schools:
 - div: Factor variable with division school is in (1, 2, or 3A)
 - label: Factor variable with school name

Examples

```
# prep the data
soccernet <- merge(soccer$edges, soccer$vertices, by.x = "home",
                  by.y = "label", all = TRUE)

library(geomnet)
library(dplyr)
# create plot
ggplot(data = soccernet, aes(from_id = home, to_id = away)) +
  geom_net(aes(colour = div, group = div), ealpha = .25,
          layout.alg = 'fruchtermanreingold') +
  facet_wrap(~season) +
  theme_net()
```

theme_elements	ggplot2 theme attribute inheritance network (directed)
----------------	---

Description

A list of two datasets, vertices and edges, containing data on the inheritance structure of theme elements in **ggplot2** (see <http://docs.ggplot2.org/current/theme.html>) The variables are as follows:

Usage

```
theme_elements
```

Format

A list of two data frames:

- the edges data frame consists of two variables of length 48:
 - child: Theme element that inherits its properties from the corresponding parent element
 - parent: Theme element that passes its properties to its children
- the vertices data frame consists of one variable with information on 53 theme elements:
 - name: Name of the theme element

theme_net	<i>Theme for network visualization</i>
-----------	--

Description

Themes set the general aspect of the plot such as the colour of the background, gridlines, the size and colour of fonts. `theme_net` provides access to the regular `ggplot2` theme, but removes any background, axes, and ensures an aspect ratio of 1 for better viewing of networks and graphs.

Arguments

<code>base_size</code>	base font size
<code>base_family</code>	base font family

Examples

```
library(ggplot2)
data(blood)
p <- ggplot(data = blood$edges, aes(from_id = from, to_id = to))
p + geom_net()
p + geom_net() + theme_net()
```

Index

*Topic **datasets**

- bikes, 3
- blood, 4
- email, 4
- football, 5
- geom_net, 11
- GeomCircle, 9
- lesmis, 16
- madmen, 17
- metro_map, 18
- mm.directed, 18
- protein, 19
- soccer, 20
- theme_elements, 21

- aes, 10, 11
- aes_, 10, 11
- arrow, 12
- as.adjmat, 2
- as.edgedf, 2, 8

- bikes, 3
- blood, 4
- borders, 10, 12

- email, 4

- football, 5
- fortify, 10, 11
- fortify.adjmat, 2, 7
- fortify.edgedf, 2, 7, 12
- fortify.igraph, 8
- fortify.network, 9

- geom_circle (GeomCircle), 9
- geom_net, 11
- geom_point, 11
- geom_segment, 11
- geom_text, 11
- GeomCircle, 9
- GeomNet (geom_net), 11

- ggplot, 10, 11
- gplot.layout, 12

- jtt, 15

- layer, 10, 12
- lesmis, 16

- madmen, 17
- metro_map, 18
- mm.directed, 18

- protein, 19

- soccer, 20
- stat_net (geom_net), 11
- StatNet (geom_net), 11

- theme_elements, 21
- theme_net, 21