Graphing.R

riserate

Fri Jun 30 20:49:04 2017

#Network graphing

#need packages  
library(GGally)  
library(network)

## network: Classes for Relational Data  
## Version 1.13.0 created on 2015-08-31.  
## copyright (c) 2005, Carter T. Butts, University of California-Irvine  
## Mark S. Handcock, University of California -- Los Angeles  
## David R. Hunter, Penn State University  
## Martina Morris, University of Washington  
## Skye Bender-deMoll, University of Washington  
## For citation information, type citation("network").  
## Type help("network-package") to get started.

library(sna)

## Loading required package: statnet.common

## sna: Tools for Social Network Analysis  
## Version 2.4 created on 2016-07-23.  
## copyright (c) 2005, Carter T. Butts, University of California-Irvine  
## For citation information, type citation("sna").  
## Type help(package="sna") to get started.

library(ggplot2)  
library(igraph)

##   
## Attaching package: 'igraph'

## The following objects are masked from 'package:sna':  
##   
## betweenness, bonpow, closeness, components, degree,  
## dyad.census, evcent, hierarchy, is.connected, neighborhood,  
## triad.census

## The following objects are masked from 'package:network':  
##   
## %c%, %s%, add.edges, add.vertices, delete.edges,  
## delete.vertices, get.edge.attribute, get.edges,  
## get.vertex.attribute, is.bipartite, is.directed,  
## list.edge.attributes, list.vertex.attributes,  
## set.edge.attribute, set.vertex.attribute

## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

## The following object is masked from 'package:base':  
##   
## union

library(statnet)

## Loading required package: tergm

## Loading required package: ergm

##   
## ergm: version 3.7.1, created on 2017-03-20  
## Copyright (c) 2017, Mark S. Handcock, University of California -- Los Angeles  
## David R. Hunter, Penn State University  
## Carter T. Butts, University of California -- Irvine  
## Steven M. Goodreau, University of Washington  
## Pavel N. Krivitsky, University of Wollongong  
## Martina Morris, University of Washington  
## with contributions from  
## Li Wang  
## Kirk Li, University of Washington  
## Skye Bender-deMoll, University of Washington  
## Based on "statnet" project software (statnet.org).  
## For license and citation information see statnet.org/attribution  
## or type citation("ergm").

## NOTE: Versions before 3.6.1 had a bug in the implementation of the  
## bd() constriant which distorted the sampled distribution somewhat.  
## In addition, Sampson's Monks datasets had mislabeled verteces. See  
## the NEWS and the documentation for more details.

## Loading required package: networkDynamic

##   
## networkDynamic: version 0.9.0, created on 2016-01-12  
## Copyright (c) 2016, Carter T. Butts, University of California -- Irvine  
## Ayn Leslie-Cook, University of Washington  
## Pavel N. Krivitsky, University of Wollongong  
## Skye Bender-deMoll, University of Washington  
## with contributions from  
## Zack Almquist, University of California -- Irvine  
## David R. Hunter, Penn State University  
## Li Wang  
## Kirk Li, University of Washington  
## Steven M. Goodreau, University of Washington  
## Jeffrey Horner  
## Martina Morris, University of Washington  
## Based on "statnet" project software (statnet.org).  
## For license and citation information see statnet.org/attribution  
## or type citation("networkDynamic").

##   
## tergm: version 3.4.0, created on 2016-03-28  
## Copyright (c) 2016, Pavel N. Krivitsky, University of Wollongong  
## Mark S. Handcock, University of California -- Los Angeles  
## with contributions from  
## David R. Hunter, Penn State University  
## Steven M. Goodreau, University of Washington  
## Martina Morris, University of Washington  
## Nicole Bohme Carnegie, New York University  
## Carter T. Butts, University of California -- Irvine  
## Ayn Leslie-Cook, University of Washington  
## Skye Bender-deMoll  
## Li Wang  
## Kirk Li, University of Washington  
## Based on "statnet" project software (statnet.org).  
## For license and citation information see statnet.org/attribution  
## or type citation("tergm").

## Loading required package: ergm.count

##   
## ergm.count: version 3.2.2, created on 2016-03-29  
## Copyright (c) 2016, Pavel N. Krivitsky, University of Wollongong  
## with contributions from  
## Mark S. Handcock, University of California -- Los Angeles  
## David R. Hunter, Penn State University  
## Based on "statnet" project software (statnet.org).  
## For license and citation information see statnet.org/attribution  
## or type citation("ergm.count").

## NOTE: The form of the term 'CMP' has been changed in version 3.2  
## of 'ergm.count'. See the news or help('CMP') for more information.

##   
## statnet: version 2016.9, created on 2016-08-29  
## Copyright (c) 2016, Mark S. Handcock, University of California -- Los Angeles  
## David R. Hunter, Penn State University  
## Carter T. Butts, University of California -- Irvine  
## Steven M. Goodreau, University of Washington  
## Pavel N. Krivitsky, University of Wollongong  
## Skye Bender-deMoll  
## Martina Morris, University of Washington  
## Based on "statnet" project software (statnet.org).  
## For license and citation information see statnet.org/attribution  
## or type citation("statnet").

## unable to reach CRAN

pairs <- read.table('pairs.txt')  
g <- graph.data.frame(pairs, directed = TRUE)  
  
#vertices count  
vcount(g)

## [1] 15

#edge count  
ecount(g)

## [1] 47

E(g)

## + 47/47 edges (vertex names):  
## [1] AR ->EN AR ->FR ZHS->EN ZHS->JA ZHT->EN NL ->EN NL ->FR   
## [8] NL ->DE NL ->ES EN ->AR EN ->ZHS EN ->ZHT EN ->NL EN ->FR   
## [15] EN ->DE EN ->HE EN ->IT EN ->JA EN ->PL EN ->PT EN ->RU   
## [22] EN ->ES EN ->RO FR ->AR FR ->ZHS FR ->NL FR ->EN FR ->DE   
## [29] FR ->HE FR ->IT FR ->JA FR ->PL FR ->PT FR ->RU FR ->ES   
## [36] DE ->AR DE ->NL DE ->EN DE ->JA DE ->PT HE ->FR IT ->EN   
## [43] IT ->JA IT ->PT JA ->ZHS JA ->EN ES ->PT

#get edge list  
get.edgelist(g)

## [,1] [,2]   
## [1,] "AR" "EN"   
## [2,] "AR" "FR"   
## [3,] "ZHS" "EN"   
## [4,] "ZHS" "JA"   
## [5,] "ZHT" "EN"   
## [6,] "NL" "EN"   
## [7,] "NL" "FR"   
## [8,] "NL" "DE"   
## [9,] "NL" "ES"   
## [10,] "EN" "AR"   
## [11,] "EN" "ZHS"  
## [12,] "EN" "ZHT"  
## [13,] "EN" "NL"   
## [14,] "EN" "FR"   
## [15,] "EN" "DE"   
## [16,] "EN" "HE"   
## [17,] "EN" "IT"   
## [18,] "EN" "JA"   
## [19,] "EN" "PL"   
## [20,] "EN" "PT"   
## [21,] "EN" "RU"   
## [22,] "EN" "ES"   
## [23,] "EN" "RO"   
## [24,] "FR" "AR"   
## [25,] "FR" "ZHS"  
## [26,] "FR" "NL"   
## [27,] "FR" "EN"   
## [28,] "FR" "DE"   
## [29,] "FR" "HE"   
## [30,] "FR" "IT"   
## [31,] "FR" "JA"   
## [32,] "FR" "PL"   
## [33,] "FR" "PT"   
## [34,] "FR" "RU"   
## [35,] "FR" "ES"   
## [36,] "DE" "AR"   
## [37,] "DE" "NL"   
## [38,] "DE" "EN"   
## [39,] "DE" "JA"   
## [40,] "DE" "PT"   
## [41,] "HE" "FR"   
## [42,] "IT" "EN"   
## [43,] "IT" "JA"   
## [44,] "IT" "PT"   
## [45,] "JA" "ZHS"  
## [46,] "JA" "EN"   
## [47,] "ES" "PT"

#sample vertices  
neighbors(g, 1)

## + 2/15 vertices, named:  
## [1] EN FR

#Longest geodesic distance  
diameter(g, directed=F)

## [1] 2

diam <- get\_diameter(g, directed=T)  
diam

## + 4/15 vertices, named:  
## [1] HE FR EN ZHT

class(diam)

## [1] "igraph.vs"

as.vector(diam)

## [1] 8 6 5 3

distances(g, v = V(g), to = V(g), mode = c("all", "out",  
 "in"), weights = NULL, algorithm = c("automatic"))

## AR ZHS ZHT NL EN FR DE HE IT JA ES PL PT RU RO  
## AR 0 2 2 2 1 1 1 2 2 2 2 2 2 2 2  
## ZHS 2 0 2 2 1 1 2 2 2 1 2 2 2 2 2  
## ZHT 2 2 0 2 1 2 2 2 2 2 2 2 2 2 2  
## NL 2 2 2 0 1 1 1 2 2 2 1 2 2 2 2  
## EN 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1  
## FR 1 1 2 1 1 0 1 1 1 1 1 1 1 1 2  
## DE 1 2 2 1 1 1 0 2 2 1 2 2 1 2 2  
## HE 2 2 2 2 1 1 2 0 2 2 2 2 2 2 2  
## IT 2 2 2 2 1 1 2 2 0 1 2 2 1 2 2  
## JA 2 1 2 2 1 1 1 2 1 0 2 2 2 2 2  
## ES 2 2 2 1 1 1 2 2 2 2 0 2 1 2 2  
## PL 2 2 2 2 1 1 2 2 2 2 2 0 2 2 2  
## PT 2 2 2 2 1 1 1 2 1 2 1 2 0 2 2  
## RU 2 2 2 2 1 1 2 2 2 2 2 2 2 0 2  
## RO 2 2 2 2 1 2 2 2 2 2 2 2 2 2 0

#Average path distance  
mean\_distance(g, directed = TRUE, unconnected = TRUE)

## [1] 1.680851

closeness(g, mode="all")

## AR ZHS ZHT NL EN FR   
## 0.04000000 0.04000000 0.03703704 0.04166667 0.07142857 0.06250000   
## DE HE IT JA ES PL   
## 0.04545455 0.03846154 0.04166667 0.04347826 0.04166667 0.03846154   
## PT RU RO   
## 0.04347826 0.03846154 0.03703704

#Edges going into or out of a vertex  
inc.edges <- incident(g, V(g), mode="all")  
class(inc.edges)

## [1] "igraph.es"

inc.edges

## + 5/47 edges (vertex names):  
## [1] AR->EN AR->FR EN->AR FR->AR DE->AR

neighbors(g, V(g), mode="in")

## + 3/15 vertices, named:  
## [1] EN FR DE

neighbors(g, V(g), mode="out")

## + 2/15 vertices, named:  
## [1] EN FR

neighbors(g, V(g), mode="all")

## + 5/15 vertices, named:  
## [1] EN EN FR FR DE

#Number of paths  
#Number of geodesics that pass through the node or the edge.  
routes <- edge\_betweenness(g, directed=T, weights=NA)  
print(routes)

## [1] 8.000000 6.333333 13.000000 1.000000 14.000000 6.416667 4.750000  
## [8] 1.916667 1.750000 5.333333 4.000000 9.000000 5.500000 5.333333  
## [15] 5.500000 7.000000 6.000000 2.833333 7.000000 4.750000 7.000000  
## [22] 6.000000 10.000000 2.333333 3.000000 2.500000 6.000000 2.500000  
## [29] 2.000000 3.000000 2.833333 3.000000 2.750000 3.000000 2.500000  
## [36] 1.666667 1.833333 8.333333 1.833333 1.250000 14.000000 11.500000  
## [43] 1.500000 1.000000 2.000000 13.000000 1.250000

summary(routes)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.167 4.000 5.043 6.708 14.000

sum(routes)

## [1] 237

#Cocitation counts  
cocitation(g, v = V(g))

## AR ZHS ZHT NL EN FR DE HE IT JA ES PL PT RU RO  
## AR 0 2 1 3 2 1 2 2 2 3 2 2 3 2 1  
## ZHS 2 0 1 2 2 1 2 2 2 2 2 2 2 2 1  
## ZHT 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1  
## NL 3 2 1 0 2 1 2 2 2 3 2 2 3 2 1  
## EN 2 2 0 2 0 2 2 1 1 4 2 1 3 1 0  
## FR 1 1 1 1 2 0 2 1 1 1 2 1 1 1 1  
## DE 2 2 1 2 2 2 0 2 2 2 3 2 2 2 1  
## HE 2 2 1 2 1 1 2 0 2 2 2 2 2 2 1  
## IT 2 2 1 2 1 1 2 2 0 2 2 2 2 2 1  
## JA 3 2 1 3 4 1 2 2 2 0 2 2 4 2 1  
## ES 2 2 1 2 2 2 3 2 2 2 0 2 2 2 1  
## PL 2 2 1 2 1 1 2 2 2 2 2 0 2 2 1  
## PT 3 2 1 3 3 1 2 2 2 4 2 2 0 2 1  
## RU 2 2 1 2 1 1 2 2 2 2 2 2 2 0 1  
## RO 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0

#Edge sequences  
E(g)[ V(g)[pairs$V1] %->% V(g)[pairs$V2] ] #to

## + 47/47 edges (vertex names):  
## [1] AR ->EN AR ->FR ZHS->EN ZHS->JA ZHT->EN NL ->EN NL ->FR   
## [8] NL ->DE NL ->ES EN ->AR EN ->ZHS EN ->ZHT EN ->NL EN ->FR   
## [15] EN ->DE EN ->HE EN ->IT EN ->JA EN ->PL EN ->PT EN ->RU   
## [22] EN ->ES EN ->RO FR ->AR FR ->ZHS FR ->NL FR ->EN FR ->DE   
## [29] FR ->HE FR ->IT FR ->JA FR ->PL FR ->PT FR ->RU FR ->ES   
## [36] DE ->AR DE ->NL DE ->EN DE ->JA DE ->PT HE ->FR IT ->EN   
## [43] IT ->JA IT ->PT JA ->ZHS JA ->EN ES ->PT

E(g)[ V(g)[pairs$V1] %<-% V(g)[pairs$V2] ] #from

## + 37/47 edges (vertex names):  
## [1] AR ->EN AR ->FR ZHS->EN ZHS->JA ZHT->EN NL ->EN NL ->FR   
## [8] NL ->DE NL ->ES EN ->AR EN ->ZHS EN ->ZHT EN ->NL EN ->FR   
## [15] EN ->DE EN ->HE EN ->IT EN ->JA EN ->ES FR ->AR FR ->ZHS  
## [22] FR ->NL FR ->EN FR ->DE FR ->HE FR ->IT FR ->JA FR ->ES   
## [29] DE ->AR DE ->NL DE ->EN DE ->JA HE ->FR IT ->EN IT ->JA   
## [36] JA ->ZHS JA ->EN

E(g)$weight <- runif(ecount(g))  
get.adjacency(g, attr="weight")

## 15 x 15 sparse Matrix of class "dgCMatrix"

## [[ suppressing 15 column names 'AR', 'ZHS', 'ZHT' ... ]]

##   
## AR . . . . 0.2815954 0.5760622 .   
## ZHS . . . . 0.9558547 . .   
## ZHT . . . . 0.3673601 . .   
## NL . . . . 0.7895790 0.7938135 0.43833263  
## EN 0.1433979 0.6946875 0.9371556 0.1297684 . 0.4914945 0.27621878  
## FR 0.4447705 0.3549780 . 0.3579089 0.9568124 . 0.08370281  
## DE 0.6140744 . . 0.1910073 0.3087388 . .   
## HE . . . . . 0.6463192 .   
## IT . . . . 0.5546460 . .   
## JA . 0.3080918 . . 0.1637722 . .   
## ES . . . . . . .   
## PL . . . . . . .   
## PT . . . . . . .   
## RU . . . . . . .   
## RO . . . . . . .   
##   
## AR . . . . . . .   
## ZHS . . 0.8904188 . . . .   
## ZHT . . . . . . .   
## NL . . . 0.8910039 . . .   
## EN 0.9625784 0.5157586 0.4194800 0.5691678 0.13387998 0.7445893 0.2712097  
## FR 0.6897021 0.3182743 0.6260653 0.9942191 0.08101332 0.3249126 0.4351447  
## DE . . 0.3303604 . . 0.3831370 .   
## HE . . . . . . .   
## IT . . 0.9963218 . . 0.5641204 .   
## JA . . . . . . .   
## ES . . . . . 0.3426568 .   
## PL . . . . . . .   
## PT . . . . . . .   
## RU . . . . . . .   
## RO . . . . . . .   
##   
## AR .   
## ZHS .   
## ZHT .   
## NL .   
## EN 0.7209308  
## FR .   
## DE .   
## HE .   
## IT .   
## JA .   
## ES .   
## PL .   
## PT .   
## RU .   
## RO .

E(g)$width <- runif(ecount(g))  
get.adjacency(g, attr="width")

## 15 x 15 sparse Matrix of class "dgCMatrix"

## [[ suppressing 15 column names 'AR', 'ZHS', 'ZHT' ... ]]

##   
## AR . . . . 0.3173997 0.1990435 .   
## ZHS . . . . 0.6790989 . .   
## ZHT . . . . 0.7577452 . .   
## NL . . . . 0.7507769 0.3803242 0.7267607  
## EN 0.07494712 0.8666187 0.6899915 0.4450436 . 0.8690926 0.2112155  
## FR 0.04280448 0.4330717 . 0.7440613 0.8110644 . 0.4379991  
## DE 0.48045945 . . 0.7993773 0.8520805 . .   
## HE . . . . . 0.7377749 .   
## IT . . . . 0.1289876 . .   
## JA . 0.5265848 . . 0.5922658 . .   
## ES . . . . . . .   
## PL . . . . . . .   
## PT . . . . . . .   
## RU . . . . . . .   
## RO . . . . . . .   
##   
## AR . . . . . .   
## ZHS . . 0.6616973 . . .   
## ZHT . . . . . .   
## NL . . . 0.78897722 . .   
## EN 0.99942653 0.2645482 0.6980489 0.01634439 0.1886732 0.3153534  
## FR 0.06839586 0.6321598 0.8340429 0.49896858 0.8299299 0.4542885  
## DE . . 0.4262329 . . 0.8105446  
## HE . . . . . .   
## IT . . 0.1700874 . . 0.5231601  
## JA . . . . . .   
## ES . . . . . 0.7847865  
## PL . . . . . .   
## PT . . . . . .   
## RU . . . . . .   
## RO . . . . . .   
##   
## AR . .   
## ZHS . .   
## ZHT . .   
## NL . .   
## EN 0.1194842 0.1358895  
## FR 0.7178766 .   
## DE . .   
## HE . .   
## IT . .   
## JA . .   
## ES . .   
## PL . .   
## PT . .   
## RU . .   
## RO . .

E(g)$name <- runif(ecount(g))  
get.adjacency(g, attr="name")

## 15 x 15 sparse Matrix of class "dgCMatrix"

## [[ suppressing 15 column names 'AR', 'ZHS', 'ZHT' ... ]]

##   
## AR . . . . 0.18679832 0.2119506  
## ZHS . . . . 0.64674704 .   
## ZHT . . . . 0.14363565 .   
## NL . . . . 0.48437440 0.9762041  
## EN 0.36843741 0.4204713 0.09737423 0.1485483 . 0.2012075  
## FR 0.04998318 0.8918697 . 0.4908143 0.71371932 .   
## DE 0.16602235 . . 0.7965818 0.06888053 .   
## HE . . . . . 0.1253218  
## IT . . . . 0.62210145 .   
## JA . 0.3339363 . . 0.53422758 .   
## ES . . . . . .   
## PL . . . . . .   
## PT . . . . . .   
## RU . . . . . .   
## RO . . . . . .   
##   
## AR . . . . . .   
## ZHS . . . 0.17947176 . .   
## ZHT . . . . . .   
## NL 0.7933228 . . . 0.25405994 .   
## EN 0.2380741 0.3603057 0.09196550 0.90255752 0.06511877 0.9569323  
## FR 0.5626724 0.6324085 0.03604536 0.01387262 0.17189592 0.8244220  
## DE . . . 0.91637991 . .   
## HE . . . . . .   
## IT . . . 0.78362784 . .   
## JA . . . . . .   
## ES . . . . . .   
## PL . . . . . .   
## PT . . . . . .   
## RU . . . . . .   
## RO . . . . . .   
##   
## AR . . .   
## ZHS . . .   
## ZHT . . .   
## NL . . .   
## EN 0.5460209 0.8993011 0.9403033  
## FR 0.2661126 0.6624591 .   
## DE 0.5253567 . .   
## HE . . .   
## IT 0.3939191 . .   
## JA . . .   
## ES 0.8893748 . .   
## PL . . .   
## PT . . .   
## RU . . .   
## RO . . .

#Graph attributes  
edge\_attr(g, name = "weight", index = E(g))

## [1] 0.28159542 0.57606225 0.95585474 0.89041879 0.36736013 0.78957896  
## [7] 0.79381346 0.43833263 0.89100394 0.14339791 0.69468753 0.93715563  
## [13] 0.12976837 0.49149446 0.27621878 0.96257841 0.51575863 0.41947998  
## [19] 0.13387998 0.74458928 0.27120974 0.56916777 0.72093076 0.44477054  
## [25] 0.35497803 0.35790891 0.95681239 0.08370281 0.68970213 0.31827433  
## [31] 0.62606525 0.08101332 0.32491255 0.43514465 0.99421912 0.61407443  
## [37] 0.19100725 0.30873878 0.33036043 0.38313698 0.64631917 0.55464603  
## [43] 0.99632180 0.56412036 0.30809183 0.16377217 0.34265682

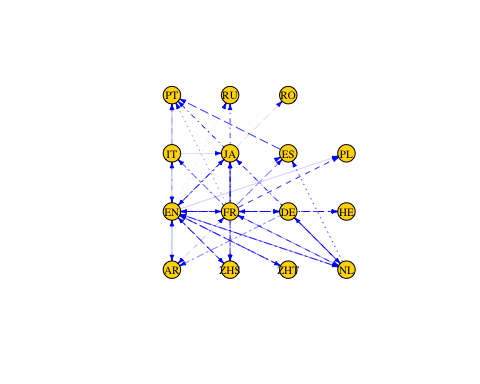
edge\_attr(g, name = "width", index = E(g))

## [1] 0.31739965 0.19904351 0.67909891 0.66169735 0.75774521 0.75077694  
## [7] 0.38032421 0.72676070 0.78897722 0.07494712 0.86661866 0.68999153  
## [13] 0.44504357 0.86909265 0.21121545 0.99942653 0.26454817 0.69804895  
## [19] 0.18867316 0.31535337 0.11948422 0.01634439 0.13588953 0.04280448  
## [25] 0.43307173 0.74406128 0.81106439 0.43799914 0.06839586 0.63215982  
## [31] 0.83404289 0.82992990 0.45428850 0.71787661 0.49896858 0.48045945  
## [37] 0.79937728 0.85208051 0.42623285 0.81054464 0.73777493 0.12898759  
## [43] 0.17008744 0.52316008 0.52658479 0.59226576 0.78478648

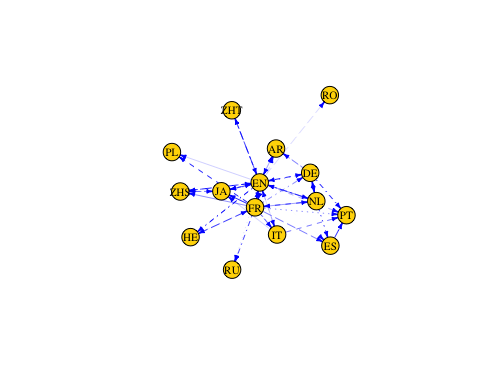
edge\_attr(g, name = "name", index = E(g))

## [1] 0.18679832 0.21195062 0.64674704 0.17947176 0.14363565 0.48437440  
## [7] 0.97620412 0.79332277 0.25405994 0.36843741 0.42047132 0.09737423  
## [13] 0.14854835 0.20120753 0.23807415 0.36030575 0.09196550 0.90255752  
## [19] 0.95693229 0.54602091 0.89930109 0.06511877 0.94030330 0.04998318  
## [25] 0.89186966 0.49081431 0.71371932 0.56267237 0.63240847 0.03604536  
## [31] 0.01387262 0.82442201 0.26611260 0.66245913 0.17189592 0.16602235  
## [37] 0.79658178 0.06888053 0.91637991 0.52535670 0.12532176 0.62210145  
## [43] 0.78362784 0.39391912 0.33393635 0.53422758 0.88937481

ceb <- cluster\_edge\_betweenness(g)   
  
#dendPlot(ceb, mode="hclust")  
#class(diam)  
  
  
#Final Plot  
E(g)$routes <- runif(ecount(g))  
E(g)$weight <- runif(length(E(g)),.1,20)  
E(g)$edge <- runif(length(E(g)),.1,20)  
E(g)$lty <- 1:47  
  
#grid style  
plot(g, layout=layout.grid, edge.width=E(g)$width,  
 edge.arrow.size=.3, vertex.color="gold", vertex.size=20, vertex.label.color="black",   
 vertex.label.cex=.7, vertex.label.dist=0, edge.width=E(g)$weigth, edge.color="blue", edge.width=.01)



#reingold style  
plot(g, layout=layout.fruchterman.reingold, edge.width=E(g)$width,  
 edge.arrow.size=.3, vertex.color="gold", vertex.size=20, vertex.label.color="black",   
 vertex.label.cex=.7, vertex.label.dist=0, edge.width=E(g)$weigth, edge.color="blue", edge.width=.01)



#automatic style  
plot(g, layout=layout\_nicely, edge.width=E(g)$width,  
 edge.arrow.size=.3, vertex.color="gold", vertex.size=20, vertex.label.color="black",   
 vertex.label.cex=.7, vertex.label.dist=0, edge.width=E(g)$weigth, edge.color="blue", edge.width=.01)

