PRACTICAL-1

library(igraph)

g<-graph.formula(1-2,1-3,2-3,2-4,3-5,4-5,4-6,4-7,5-6,6-7)

plot(g)

ecount(g)

vcount(g)

degree(g)

dg<-graph.formula(1-+2,1-+3,2++3)

plot(dg)

degree(dg,mode = "in")

degree(dg,mode = "out")

V(dg)$name[degree(dg)==min(degree(dg))]

V(dg)$name[degree(dg)==max(degree(dg))]

neighbors(g,5)

neighbors(g,2)

get.adjlist(dg)

get.adjacency(g)

PRACTICAL-2

getwd()

setwd("C:/Free time")

nodes<-read.csv("nodes.csv",header=T,as.is = T)

head(nodes)

links<-read.csv("edges1.csv",header=T,as.is = T)

head(links)

net <- graph.data.frame(d=links, vertices=nodes, directed=T)

m=as.matrix(net)

get.adjacency(m)

plot(net)

PRACTICAL-3

library(igraph)

vcount(g)

ecount(g)

ecount(g)/(vcount(g)\*(vcount(g)-1)/2)

degree(net)

dg<-graph.formula(1-+2,1-+3,2++3)

plot(dg)

reciprocity(dg)

dyad.census(dg)

2\*dyad.census(dg)$mut/ecount(dg)

kite<-graph.famous("Krackhardt\_Kite")

atri<-adjacent.triangles(kite)

plot(kite,vertex.label=atri)

transitivity(kite,type = "local")

adjacent.triangles(kite)/(degree(kite)\*(degree(kite)-1)/2)

g2<-barabasi.game(50,p=2,directed = F)

g1<-watts.strogatz.game(1,size = 100,nei = 5,p=0.05)

g<-graph.union(g1,g2)

g<-simplify(g)

plot(g)

PRACTICAL-4

library(igraph)

matt<-as.matrix(read.table(text=

"node R S T U

R 7 5 0 0

S 7 0 0 2

T 0 6 0 0

U 4 0 1 0",header=T))

nms<-matt[,1]

matt<-matt[,-1]

colnames(matt)<-rownames(matt)<-nms

matt[is.na(matt)]<-0

g<-graph.adjacency(matt,weighted = TRUE)

plot(g)

s.paths<-shortest.paths(g,algorithm = "dijkstra")

print(s.paths)

shortest.paths(g,v="R",to="S")

plot(g,edge.label=E(g)$weight)

dg<-graph.formula(1-+2,1-+3,2++3)

plot(dg)

graph.density(dg,loops = TRUE)

graph.density(simplify(dg),loops = FALSE)

PRACTICAL-5

library(igraph)

ng<-graph.formula(Andy++Garth,Garth-+Bill,Bill-+Elena,Elena++Frank,Carol-+Andy,Carol-+Elena,Carol++Dan,Carol++Bill,Dan++Andy,Dan++Bill)

plot(ng)

get.adjacency(ng)

E(ng)

get.adjedgelist(ng,mode = "in")

PRACTICAL-6

library(sna)

library(igraph)

setwd("C:/Free time")

links2<-read.csv("edges1.csv",header=T,row.names = 1)

eq<- equiv.clust(links2)

plot(eq)

g.se<-sedist(links2)

plot(cmdscale(as.dist(g.se)))

b<-blockmodel(links2,eq,h=10)

plot(b)

PRACTICAL-8

library(igraph)

a<-matrix(c(1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,1,1,1),9,4)

print(a)

svd(a)