A3: Spatial Network

Αντώνιος Νεκτάριος Ντουντούκαλης

15 January, 2025

Περιεχόμενα

[Δεδομένα δικτύου 2](#_Toc187873309)

[Ενσωμάτωση των σημείων 4](#_Toc187873310)

[Ανάθεση 6](#_Toc187873311)

[Στατιστικά ανάθεσης 8](#_Toc187873312)

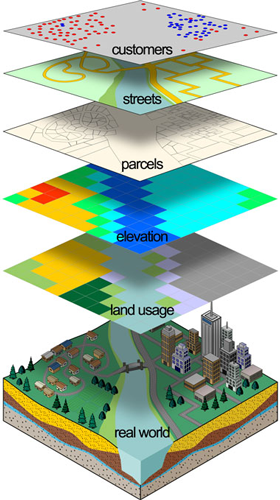
[Περιορισμός απόστασης 9](#_Toc187873313)

[Περιορισμός πλήθους 10](#_Toc187873314)

[Πηγαίος κώδικας 12](#_Toc187873315)

[Βιβλιογραφία 17](#_Toc187873316)

**Μάθημα: Γεωγραφική Ανάλυση**



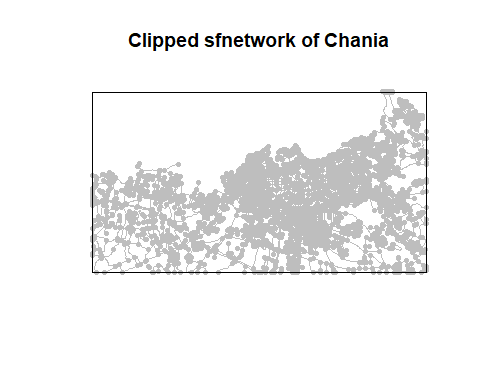
# Δεδομένα δικτύου

Εδώ ορίζουμε το BoundingBox (ΒΒ) των Χανίων. Κατεβάζει οδικό δίκτυο εντός του BB της περιοχής μελέτης.

q = c(23.9639, 35.4894, 24.0611, 35.5322) # Bounding Box πόλης Χανίων  
net2 = goal::osm.getRoads(q, withBB=TRUE, outcrs=4326)

##   
## Using bbox

poly = goal::osm.bb\_2\_pol(q, outcrs = 4326) # bbox σε spatial polygon  
net3 = goal::osm.ClipSFnetwork\_with\_poly(net2, poly) # clip network by spatial polygon  
  
plot(net3,col="grey", main="Clipped sfnetwork of Chania")  
plot(poly,add=T)



# Create a random point  
#gps = sfheaders::sf\_point(data.frame(y = 26.55257, x = 39.10575, ID=-1)) %>% st\_set\_crs(4326)  
  
# nearest edge (road) to the point. The network must have edges activated.  
#near\_edge = st\_nearest\_feature(gps, net3 %>% st\_as\_sf())  
  
#near\_edge  
#st\_as\_sf(net3)[near\_edge,]  
  
#p3 = ggplot() +  
#geom\_sf(data = st\_as\_sf(net3), color = 'black') +  
#geom\_sf(data = gps, color = 'red') +  
#geom\_sf(data = st\_as\_sf(net3)[near\_edge,], color = 'orange')  
#p3  
#plot(net3)  
#net3

Non-Isolated Nodes

net3 %>% sfnetworks::activate("nodes") %>% dplyr::filter(!tidygraph::node\_is\_isolated())

## # A sfnetwork with 3851 nodes and 2128 edges  
## #  
## # CRS: EPSG:4326   
## #  
## # A directed acyclic simple graph with 1726 components with spatially explicit edges  
## #  
## # Node data: 3,851 × 1 (active)  
## geometry  
## <POINT [°]>  
## 1 (24.02499 35.51064)  
## 2 (24.02294 35.51349)  
## 3 (24.02119 35.51146)  
## 4 (24.02025 35.51258)  
## 5 (24.02968 35.51043)  
## 6 (24.05179 35.52471)  
## # ℹ 3,845 more rows  
## #  
## # Edge data: 2,128 × 6  
## from to osm\_id name highway geometry  
## <int> <int> <chr> <chr> <chr> <LINESTRING [°]>  
## 1 1938 1 5738798 Νεάρχου secondary (24.02504 35.50692, 24.0250…  
## 2 2 3 5738801 Μπονιάλη Κτιστάκη residential (24.02294 35.51349, 24.0224…  
## 3 1939 4 5738803 Αποκορώνου secondary (24.01965 35.51346, 24.0196…  
## # ℹ 2,125 more rows

Isolated Nodes

net3 %>% sfnetworks::activate("nodes") %>% dplyr::filter(tidygraph::node\_is\_isolated()) %>% st\_as\_sf() %>% nrow()

## [1] 0

Loop Edges

net3 %>% sfnetworks::activate("edges") %>% dplyr::filter(tidygraph::edge\_is\_loop()) %>% st\_as\_sf() %>% nrow()

## [1] 0

Multiple Edges (has any parallel siblings)

net3 %>% sfnetworks::activate("edges") %>% dplyr::filter(tidygraph::edge\_is\_multiple()) %>% st\_as\_sf() %>% nrow()

## [1] 0

Simplify network.

net3b = net3 %>% activate("edges") %>%  
 filter(!edge\_is\_multiple()) %>%  
 filter(!edge\_is\_loop())

# Ενσωμάτωση των σημείων

Κατασκευή τοπολογικά ορθού δικτύου (net4) από τα κατεβασμένα δεδομένα της περιοχής μελέτης.  
 Subdivide network Smooth pseudo nodes

net4 = tidygraph::convert(net3b, to\_spatial\_subdivision)  
net4 = tidygraph::convert(net4, to\_spatial\_smooth)  
net4 = tidygraph::convert(net4, to\_spatial\_simple)

Πόσοι κόμβοι στο δίκτυο μας (nodes)?

n\_nodes = net4 %>% activate(nodes) %>% st\_as\_sf()%>% nrow()

Σε κάθε δρόμο (edges) πρόσθεσε μια νέα στήλη “length” με το μήκος του κάθε δρόμου.

net4 = net4 %>% activate(edges) %>%  
 mutate(length = edge\_length())%>% st\_set\_crs(4326)

Σε κάθε κόμβο (nodes), πρόσθεσε μια νέα στήλη “ID” αλλά και μια νέα στήλη “ishouse”=1.

net4 = net4 %>% activate(nodes) %>%  
 mutate(ID = 1:n\_nodes)%>%  
 mutate(ishouse = 1)%>% st\_set\_crs(4326)

Αυτά είναι τα επιλεγμένα σημεία καταφυγής μας. Τα βρήκαμε από το GoogleΜaps και μπορεί να είναι πάρκα, εκκλησίες, σχολεία, γήπεδα.

gps1 = sfheaders::sf\_point(data.frame(y =23.976000, x =35.506798)) %>% st\_set\_crs(4326)  
gps2 = sfheaders::sf\_point(data.frame(y =23.995920, x =35.499531 )) %>% st\_set\_crs(4326)  
gps3 = sfheaders::sf\_point(data.frame(y =24.016876, x =35.509592 )) %>% st\_set\_crs(4326)  
gps4 = sfheaders::sf\_point(data.frame(y =24.046765, x =35.511828)) %>% st\_set\_crs(4326)  
  
  
mycol = c("blue","green",'red',"purple")  
evac\_points = rbind(gps1, gps2, gps3,gps4) #%>% rowid\_to\_column() # Νέα στήλη 'rowid'

Ενσωμάτωση των χώρων καταφυγής, στο δίκτυο μας ως 4 νέους κόμβους (nodes):

blended = st\_network\_blend(net4, evac\_points )

Άλλαξε τις τιμές των στηλών: ‘isevac’, ‘ishouse’ ανάλογα:

blended = blended %>% activate(nodes) %>%  
 mutate(isevac = ifelse(is.na(ishouse), 1, 0) ) %>%  
 mutate(ishouse = ifelse(is.na(ishouse), 0, 1) )

Οι 10 τελευταίοι κόμβοι (nodes) του δικτύου μας μέχρι στιγμής. Φαίνονται οι 4 νέοι κόμβοι (nodes) isevac = 1

blended %>% activate("nodes") %>% st\_as\_sf() %>% as.data.frame() %>% tail(10)

## geometry .tidygraph\_node\_index ID ishouse isevac  
## 3131 POINT (24.00446 35.49387) 3131 3131 1 0  
## 3132 POINT (24.05312 35.4931) 3132 3132 1 0  
## 3133 POINT (24.00385 35.51132) 3133 3133 1 0  
## 3134 POINT (24.05049 35.49514) 3134 3134 1 0  
## 3135 POINT (23.9639 35.49405) 3135 3135 1 0  
## 3136 POINT (23.9639 35.49361) 3136 3136 1 0  
## 3137 POINT (23.9761 35.50642) NA NA 0 1  
## 3138 POINT (24.01688 35.50938) NA NA 0 1  
## 3139 POINT (24.04682 35.512) NA NA 0 1  
## 3140 POINT (23.99585 35.49946) NA NA 0 1

Αφαίρεση των disconnected

table(components(blended)$membership)

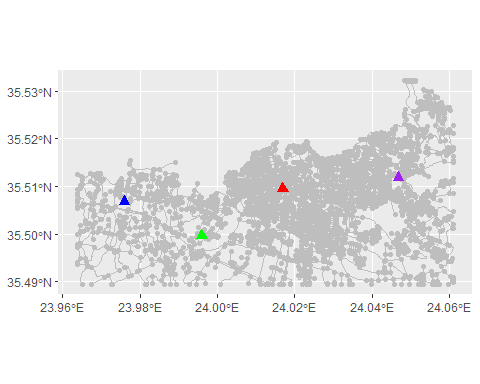
##   
## 1 2 3 4 5 6 7 8 9 10 11 12   
## 3082 10 6 24 4 2 2 2 2 2 2 2

blended = blended %>% activate('nodes')%>%  
 filter(components(blended)$membership == 1)

# Ανάθεση

Το δίκτυο μέχρι στιγμής:

ggplot() +  
 geom\_sf(data = st\_as\_sf(net4%>% activate("edges")), color = 'grey') +  
 geom\_sf(data = st\_as\_sf(net4%>% activate("nodes")), color = 'grey')+  
 geom\_sf(data = evac\_points, color = mycol, cex=3, pch=17)



Τα ‘rowids’ των χώρων καταφυγής isevac=1 :

rowids\_evac = blended %>% activate("nodes") %>% as.data.frame() %>% rowid\_to\_column() %>% filter(isevac == 1) %>% pull(rowid)  
tail(rowids\_evac)

## [1] 3079 3080 3081 3082

Τα ‘rowids’ των σπιτιών ishouse=1 :

rowids\_houses = blended %>% activate("nodes") %>% as.data.frame() %>% rowid\_to\_column() %>% filter(ishouse == 1) %>% pull(rowid)  
tail(rowids\_houses)

## [1] 3073 3074 3075 3076 3077 3078

Δημιουργία ‘Spatial object’ των σπιτιών, χώρων καταφυγής. Θα μας χρειαστούν πιο μετά για εξαγωγή σε Shapefiles.

evac\_sf = blended %>% activate("nodes") %>% st\_as\_sf() %>% filter(isevac==1)  
houses\_sf = blended %>% activate("nodes") %>% st\_as\_sf() %>% filter(ishouse==1) #%>% rowid\_to\_column()

Πόσοι δρόμοι (edges)?

n\_edges = blended %>% activate(edges) %>% st\_as\_sf()%>% nrow()

Δημιουργία στήλης σε κάθε δρόμο (edge) με τίτλο στήλης IDedge

blended = blended %>% activate(edges) %>% mutate(IDedge = 1:n\_edges)

Υπολογισμός του Distance Matrix :

dm = st\_network\_cost(blended, from =rowids\_houses , to =rowids\_evac , direction="all")  
head(dm)

## Units: [m]  
## [,1] [,2] [,3] [,4]  
## [1,] 4853.665 980.0194 2640.836 3342.103  
## [2,] 4748.423 874.7771 2540.399 3236.861  
## [3,] 4708.623 810.6987 2437.149 3197.062  
## [4,] 4754.572 858.6964 2422.334 3243.010  
## [5,] 4716.059 820.1837 2341.662 3204.497  
## [6,] 4686.025 825.8693 2548.788 3174.463

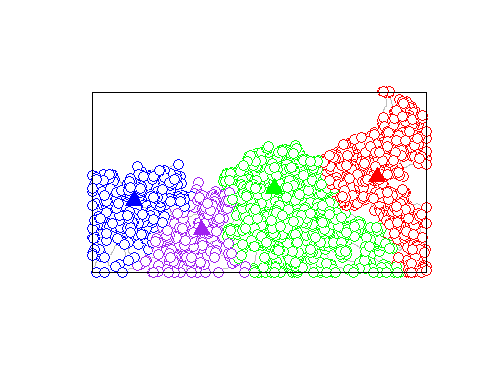
Για κάθε σπίτι, εύρεση ποιου από τους 4 χώρους καταφυγής είναι πιο κοντά. Για κάθε σπίτι, κατασκευή δυο νέων στηλών:

* closest\_index: Ποιος χώρος καταφυγής είναι πιο κοντά?
* closest\_index\_dist: Σε τι απόσταση είναι ο πιο κοντινός χώρος καταφυγής?

houses\_sf$closest\_index = apply(dm, 1, function(x) which(x == min(x))[1])  
houses\_sf$closest\_index\_dist = apply(dm, 1, function(x) min(x)[1])

Οπτικοποίηση τελικής ανάθεσης στους 4 χώρους καταφυγής:

plot(blended, col="grey")  
plot(st\_geometry(houses\_sf), cex=1.5, col=mycol[houses\_sf$closest\_index], pch=21, add=T)  
plot(st\_geometry(evac\_sf) , cex=2, pch=17, add=T, col=mycol)  
plot(poly, add=T)



## Στατιστικά ανάθεσης

Πόσοι κόμβοι ανατέθηκαν σε κάθε χώρο καταφυγής?

table(houses\_sf$closest\_index)

##   
## 1 2 3 4   
## 348 1518 923 289

Απόσταση μεταξύ κόμβων και χώρων καταφυγής (Ελάχιστη, Μέγιστη, Μέσος Όρος):

houses\_sf %>% as.data.frame()%>% group\_by(closest\_index) %>%  
 summarise( min\_dist=min(closest\_index\_dist),  
 max\_dist=max(closest\_index\_dist),  
 mean\_dist=mean(closest\_index\_dist) )

## # A tibble: 4 × 4  
## closest\_index min\_dist max\_dist mean\_dist  
## <int> <dbl> <dbl> <dbl>  
## 1 1 151. 2644. 1161.  
## 2 2 43.4 4323. 1482.  
## 3 3 42.7 4260. 1615.  
## 4 4 92.9 2784. 1273.

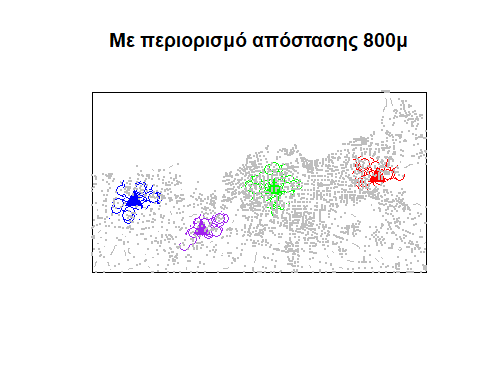
# Περιορισμός απόστασης

Ορισμός μιας μεγίστης απόστασης περπατήματος.

apostasi = 800  
  
dm2 = dm  
dm2 = units::drop\_units(dm2) # Αφαίρεση μονάδων μέτρησης από το Distance Matrix  
dm2[dm2>=apostasi] = NA  
  
  
houses\_sf$closest\_index2 = apply(dm2, 1, function(x) which(x == min(x, na.rm=T))[1])  
houses\_sf$closest\_index\_dist2 = apply(dm2, 1, function(x) min(x, na.rm=T)[1])

Οπτικοποίηση τελικής ανάθεσης στους 4 χώρους καταφυγής:

plot(blended, col="grey", main = sprintf("Με περιορισμό απόστασης %sμ", apostasi))  
plot(st\_geometry(houses\_sf), cex=1.5, col=mycol[houses\_sf$closest\_index2], pch=21, add=T)  
plot(st\_geometry(evac\_sf) , cex=2, pch=17, add=T, col=mycol)  
plot(poly, add=T)  
plot(st\_geometry(houses\_sf),cex=0.5, add=T, col="grey", pch=20)



Πόσοι κόμβοι δεν εξυπηρετούνται?

table(houses\_sf$closest\_index2, useNA="ifany")

##   
## 1 2 3 4 <NA>   
## 74 266 132 43 2563

# Περιορισμός πλήθους

Κάθε χώρος καταφυγής μπορεί να εξυπηρετήσει μόνο μέχρι 55 άτομα.

dm3 = dm  
  
df = as.data.frame(dm3)  
df$whichMin = apply(dm3, 1, which.min)  
df$minDistance = apply(dm3, 1, FUN=min, na.rm=T)

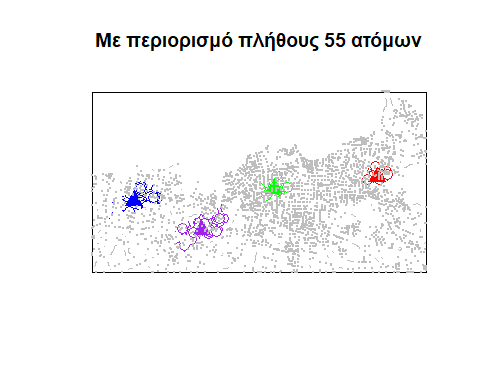
Ιεράρχηση ανά γκρούπ

library(dplyr)  
df3 = df %>%  
 group\_by(whichMin) %>%  
 mutate(my\_ranks = order(order(minDistance, decreasing=F)))  
  
  
df3

## # A tibble: 3,078 × 7  
## # Groups: whichMin [4]  
## V1 V2 V3 V4 whichMin minDistance my\_ranks  
## [m] [m] [m] [m] <int> <dbl> <int>  
## 1 4854. 980. 2641. 3342. 2 980. 417  
## 2 4748. 875. 2540. 3237. 2 875. 324  
## 3 4709. 811. 2437. 3197. 2 811. 276  
## 4 4755. 859. 2422. 3243. 2 859. 306  
## 5 4716. 820. 2342. 3204. 2 820. 282  
## 6 4686. 826. 2549. 3174. 2 826. 285  
## 7 4632. 736. 2638. 3121. 2 736. 224  
## 8 4562. 666. 2675. 3051. 2 666. 168  
## 9 4439. 543. 2742. 2927. 2 543. 107  
## 10 4371. 622. 2807. 2860. 2 622. 148  
## # ℹ 3,068 more rows

df3$whichMin2 = NA  
df3[df3$whichMin==1 & df3$my\_ranks %in% c(1:55),]$whichMin2 = 1  
df3[df3$whichMin==2 & df3$my\_ranks %in% c(1:55),]$whichMin2 = 2  
df3[df3$whichMin==3 & df3$my\_ranks %in% c(1:55),]$whichMin2 = 3  
df3[df3$whichMin==4 & df3$my\_ranks %in% c(1:55),]$whichMin2 = 4  
  
houses\_sf$closest\_index3 = df3$whichMin2

plot(blended, col="grey", main = sprintf("Με περιορισμό πλήθους %s ατόμων", 55))  
plot(st\_geometry(houses\_sf), cex=1.5, col=mycol[houses\_sf$closest\_index3], pch=21, add=T)  
plot(st\_geometry(evac\_sf) , cex=2, pch=17, add=T, col=mycol)  
plot(poly, add=T)  
plot(st\_geometry(houses\_sf),cex=0.5, add=T, col="grey", pch=20)



Απόσταση μεταξύ κόμβων και χώρων καταφυγής (Ελάχιστη, Μέγιστη, Μέσος Όρος):

houses\_sf %>% as.data.frame()%>% group\_by(closest\_index2) %>%  
 summarise( min\_dist=min(closest\_index\_dist2),  
 max\_dist=max(closest\_index\_dist2),  
 mean\_dist=mean(closest\_index\_dist2) )

## # A tibble: 5 × 4  
## closest\_index2 min\_dist max\_dist mean\_dist  
## <int> <dbl> <dbl> <dbl>  
## 1 1 151. 797. 603.  
## 2 2 43.4 798. 563.  
## 3 3 42.7 800. 535.  
## 4 4 92.9 796. 484.  
## 5 NA Inf Inf Inf

# Πηγαίος κώδικας

#'  
#library(remotes)  
#library(devtools)  
#remotes::install\_version("rgeos", version = "0.6-4")  
#remotes::install\_version("simplevis", version = "7.1.0")  
#devtools::install\_github("dimitrisk/goal", force=T)  
#'  
#1  
#library(remotes)  
#library(devtools)  
#library(sf)  
#library(tidygraph)  
#library(igraph)  
#library(dplyr)  
#library(tibble)  
#library(ggplot2)  
#library(units)  
#library(tmap)  
#library(osmdata)  
#library(link2GI)  
#library(nabor)  
#library(units)  
#library(sfnetworks)  
#library(dplyr)  
#library(goal)  
  
# Δεδομένα δικτύου   
#q = c(23.9639, 35.4894, 24.0611, 35.5322) # Bounding Box πόλης Χανίων  
#net2 = goal::osm.getRoads(q, withBB=TRUE, outcrs=4326)  
   
#poly = goal::osm.bb\_2\_pol(q, outcrs = 4326) # bbox σε spatial polygon  
#net3 = goal::osm.ClipSFnetwork\_with\_poly(net2, poly) # clip network by spatial polygon  
  
#plot(net3,col="grey", main="Clipped sfnetwork of Chania")  
#plot(poly,add=T)  
  
# Create a random point  
#gps = sfheaders::sf\_point(data.frame(y = 26.55257, x = 39.10575, ID=-1)) %>% st\_set\_crs(4326)  
  
# nearest edge (road) to the point. The network must have edges activated.  
#near\_edge = st\_nearest\_feature(gps, net3 %>% st\_as\_sf())  
  
#near\_edge  
#st\_as\_sf(net3)[near\_edge,]  
  
#p3 = ggplot() +  
#geom\_sf(data = st\_as\_sf(net3), color = 'black') +  
#geom\_sf(data = gps, color = 'red') +  
#geom\_sf(data = st\_as\_sf(net3)[near\_edge,], color = 'orange')  
#p3  
#plot(net3)  
#net3  
   
#net3 %>% sfnetworks::activate("nodes") %>% #dplyr::filter(!tidygraph::node\_is\_isolated())  
   
#net3 %>% sfnetworks::activate("nodes") %>% dplyr::filter(tidygraph::node\_is\_isolated()) %>% st\_as\_sf() %>% nrow()  
   
#net3 %>% sfnetworks::activate("edges") %>% dplyr::filter(tidygraph::edge\_is\_loop()) %>% st\_as\_sf() %>% nrow()  
   
#net3 %>% sfnetworks::activate("edges") %>% dplyr::filter(tidygraph::edge\_is\_multiple()) %>% st\_as\_sf() %>% nrow()  
  
#net3b = net3 %>% activate("edges") %>%  
# filter(!edge\_is\_multiple()) %>%  
# filter(!edge\_is\_loop())  
   
# Ενσωμάτωση των σημείων   
   
#net4 = tidygraph::convert(net3b, to\_spatial\_subdivision)  
#net4 = tidygraph::convert(net4, to\_spatial\_smooth)  
#net4 = tidygraph::convert(net4, to\_spatial\_simple)  
   
#n\_nodes = net4 %>% activate(nodes) %>% st\_as\_sf()%>% nrow()  
   
#net4 = net4 %>% activate(edges) %>%  
# mutate(length = edge\_length())%>% st\_set\_crs(4326)  
   
#net4 = net4 %>% activate(nodes) %>%  
# mutate(ID = 1:n\_nodes)%>%  
# mutate(ishouse = 1)%>% st\_set\_crs(4326)  
   
#gps1 = sfheaders::sf\_point(data.frame(y =23.976000, x =35.506798)) %>% st\_set\_crs(4326)  
#gps2 = sfheaders::sf\_point(data.frame(y =23.995920, x =35.499531 )) %>% st\_set\_crs(4326)  
#gps3 = sfheaders::sf\_point(data.frame(y =24.016876, x =35.509592 )) %>% st\_set\_crs(4326)  
#gps4 = sfheaders::sf\_point(data.frame(y =24.046765, x =35.511828)) %>% st\_set\_crs(4326)  
  
#mycol = c("blue","green",'red',"purple")  
#evac\_points = rbind(gps1, gps2, gps3,gps4) #%>% rowid\_to\_column() # Νέα στήλη 'rowid'  
   
#blended = st\_network\_blend(net4, evac\_points )  
   
#blended = blended %>% activate(nodes) %>%  
# mutate(isevac = ifelse(is.na(ishouse), 1, 0) ) %>%  
# mutate(ishouse = ifelse(is.na(ishouse), 0, 1) )  
   
#blended %>% activate("nodes") %>% st\_as\_sf() %>% as.data.frame() %>% tail(10)  
   
#table(components(blended)$membership)  
   
#blended = blended %>% activate('nodes')%>%  
# filter(components(blended)$membership == 1)  
   
# Ανάθεση   
  
#ggplot() +  
# geom\_sf(data = st\_as\_sf(net4%>% activate("edges")), color = 'grey') +  
# geom\_sf(data = st\_as\_sf(net4%>% activate("nodes")), color = 'grey')+  
# geom\_sf(data = evac\_points, color = mycol, cex=3, pch=17)  
  
#rowids\_evac = blended %>% activate("nodes") %>% as.data.frame() %>% rowid\_to\_column() %>% filter(isevac == 1) %>% pull(rowid)  
#tail(rowids\_evac)  
   
#rowids\_houses = blended %>% activate("nodes") %>% as.data.frame() %>% rowid\_to\_column() %>% filter(ishouse == 1) %>% pull(rowid)  
#tail(rowids\_houses)  
   
#evac\_sf = blended %>% activate("nodes") %>% st\_as\_sf() %>% filter(isevac==1)  
#houses\_sf = blended %>% activate("nodes") %>% st\_as\_sf() %>% filter(ishouse==1) #%>% rowid\_to\_column()  
   
#n\_edges = blended %>% activate(edges) %>% st\_as\_sf()%>% nrow()  
   
#blended = blended %>% activate(edges) %>% mutate(IDedge = 1:n\_edges)  
   
#dm = st\_network\_cost(blended, from =rowids\_houses , to =rowids\_evac , direction="all")  
#head(dm)  
   
#houses\_sf$closest\_index = apply(dm, 1, function(x) which(x == min(x))[1])  
#houses\_sf$closest\_index\_dist = apply(dm, 1, function(x) min(x)[1])  
   
#plot(blended, col="grey")  
#plot(st\_geometry(houses\_sf), cex=1.5, col=mycol[houses\_sf$closest\_index], pch=21, add=T)  
#plot(st\_geometry(evac\_sf) , cex=2, pch=17, add=T, col=mycol)  
#plot(poly, add=T)  
   
## Στατιστικά ανάθεσης   
  
#table(houses\_sf$closest\_index)  
   
#houses\_sf %>% as.data.frame()%>% group\_by(closest\_index) %>%  
# summarise( min\_dist=min(closest\_index\_dist),  
# max\_dist=max(closest\_index\_dist),  
# mean\_dist=mean(closest\_index\_dist) )  
   
# Περιορισμός απόστασης   
  
#apostasi = 800  
  
#dm2 = dm  
#dm2 = units::drop\_units(dm2) # Αφαίρεση μονάδων μέτρησης από το Distance Matrix  
#dm2[dm2>=apostasi] = NA  
  
#houses\_sf$closest\_index2 = apply(dm2, 1, function(x) which(x == min(x, na.rm=T))[1])  
#houses\_sf$closest\_index\_dist2 = apply(dm2, 1, function(x) min(x, na.rm=T)[1])  
   
#plot(blended, col="grey", main = sprintf("Με περιορισμό απόστασης %sμ", apostasi))  
#plot(st\_geometry(houses\_sf), cex=1.5, col=mycol[houses\_sf$closest\_index2], pch=21, add=T)  
#plot(st\_geometry(evac\_sf) , cex=2, pch=17, add=T, col=mycol)  
#plot(poly, add=T)  
#plot(st\_geometry(houses\_sf),cex=0.5, add=T, col="grey", pch=20)  
   
#table(houses\_sf$closest\_index2, useNA="ifany")  
   
# Περιορισμός πλήθους   
  
#dm3 = dm  
  
#df = as.data.frame(dm3)  
#df$whichMin = apply(dm3, 1, which.min)  
#df$minDistance = apply(dm3, 1, FUN=min, na.rm=T)  
   
#library(dplyr)  
#df3 = df %>%  
# group\_by(whichMin) %>%  
# mutate(my\_ranks = order(order(minDistance, decreasing=F)))  
  
  
#df3  
   
#df3$whichMin2 = NA  
#df3[df3$whichMin==1 & df3$my\_ranks %in% c(1:55),]$whichMin2 = 1  
#df3[df3$whichMin==2 & df3$my\_ranks %in% c(1:55),]$whichMin2 = 2  
#df3[df3$whichMin==3 & df3$my\_ranks %in% c(1:55),]$whichMin2 = 3  
#df3[df3$whichMin==4 & df3$my\_ranks %in% c(1:55),]$whichMin2 = 4  
  
#houses\_sf$closest\_index3 = df3$whichMin2  
   
#plot(blended, col="grey", main = sprintf("Με περιορισμό πλήθους %s ατόμων", 55))  
#plot(st\_geometry(houses\_sf), cex=1.5, col=mycol[houses\_sf$closest\_index3], pch=21, add=T)  
#plot(st\_geometry(evac\_sf) , cex=2, pch=17, add=T, col=mycol)  
#plot(poly, add=T)  
#plot(st\_geometry(houses\_sf),cex=0.5, add=T, col="grey", pch=20)  
   
#houses\_sf %>% as.data.frame()%>% group\_by(closest\_index2) %>%  
# summarise( min\_dist=min(closest\_index\_dist2),  
# max\_dist=max(closest\_index\_dist2),  
# mean\_dist=mean(closest\_index\_dist2) )

# Βιβλιογραφία

* Καβρουδάκης, Δ. (2022) ΠΠΣ Γεωγραφική Ανάλυση (ΓΕΟ 302).
  + Διαθέσιμο στο: [Open eClass, Πανεπιστήμιο Αιγαίου](https://eclass.aegean.gr/courses/GEO157/)
  + (Πρόσβαση: 14/01/2025).

