USA OSM - Esame di Advanced data science

Gianluca Fabris

# Intoduzione

Il segeuente documento presenta uno studio sulla rete stradale degli USA.

Ispirato dal dataset [US OSM roads (2018)](https://osf.io/preprints/socarxiv/7fxjz), il dataset è stato costruito utilizzando l’API [https://overpass-turbo.eu](https://overpass-turbo.eu/), per ottenere i dati aggiornati da OSM.

Ai fini dell’analisi vengono fatte alcune assunzioni, allo scopo di semplificare il modello: viene presa la rete stradale relativamente alle autostrade, viene assunto che se c’è un’autostrada che attraversa il confine tra due contee allora collega le due città.

Il grafo è stato creato nel seguente modo: - per ogni stato (compreso lo stato federale Washington DC) sono stati scaricati i dati delle contee (nome contea, nome città “capoluogo”, posizione città), le contee saranno i nodi del grafo - per ogni coppia di contee (intra-stato e inter-stato), viene verificata la presenza di un’autostrada che attraversa il confine tra le contee

Viene proposta l’analisi sulle reti dei singoli stati e sulla rete degli USA nel loro complesso.

library(tidyr)  
library(dplyr)

##   
## Caricamento pacchetto: 'dplyr'

## I seguenti oggetti sono mascherati da 'package:stats':  
##   
## filter, lag

## I seguenti oggetti sono mascherati da 'package:base':  
##   
## intersect, setdiff, setequal, union

library(purrr)  
library(readr)  
library(stringr)  
library(moments)  
library(igraph)

## Warning: il pacchetto 'igraph' è stato creato con R versione 4.4.2

##   
## Caricamento pacchetto: 'igraph'

## I seguenti oggetti sono mascherati da 'package:purrr':  
##   
## compose, simplify

## I seguenti oggetti sono mascherati da 'package:dplyr':  
##   
## as\_data\_frame, groups, union

## Il seguente oggetto è mascherato da 'package:tidyr':  
##   
## crossing

## I seguenti oggetti sono mascherati da 'package:stats':  
##   
## decompose, spectrum

## Il seguente oggetto è mascherato da 'package:base':  
##   
## union

library(tidygraph)

##   
## Caricamento pacchetto: 'tidygraph'

## Il seguente oggetto è mascherato da 'package:igraph':  
##   
## groups

## Il seguente oggetto è mascherato da 'package:stats':  
##   
## filter

library(ggraph)

## Caricamento del pacchetto richiesto: ggplot2

library(sf)

## Warning: il pacchetto 'sf' è stato creato con R versione 4.4.2

## Linking to GEOS 3.12.2, GDAL 3.9.3, PROJ 9.4.1; sf\_use\_s2() is TRUE

library(ggplot2)  
  
#nomi stati  
states = data.frame(  
 name = c("Alaska", "Alabama", "Arkansas", "Arizona", "California", "Colorado", "Connecticut", "Washington District of Columbia", "Delaware", "Florida", "Georgia", "Hawaii", "Iowa", "Idaho", "Illinois", "Indiana", "Kansas", "Kentucky", "Louisiana", "Massachusetts", "Maryland", "Maine", "Michigan", "Minnesota", "Missouri", "Mississippi", "Montana", "North Carolina", "North Dakota", "Nebraska", "New Hampshire", "New Jersey", "New Mexico", "Nevada", "New York", "Ohio", "Oklahoma", "Oregon", "Pennsylvania", "Rhode Island", "South Carolina", "South Dakota", "Tennessee", "Texas", "Utah", "Virginia", "Vermont", "Washington", "Wisconsin", "West Virginia", "Wyoming"),  
 abbr = c("AK", "AL", "AR", "AZ", "CA", "CO", "CT", "DC", "DE", "FL", "GA", "HI", "IA", "ID", "IL", "IN", "KS", "KY", "LA", "MA", "MD", "ME", "MI", "MN", "MO", "MS", "MT", "NC", "ND", "NE", "NH", "NJ", "NM", "NV", "NY", "OH", "OK", "OR", "PA", "RI", "SC", "SD", "TN", "TX", "UT", "VA", "VT", "WA", "WI", "WV", "WY")  
)

create\_state\_graph = function(state\_nodes, state\_edges){  
 if(nrow(state\_edges) == 0){  
 graph = graph\_from\_data\_frame(d = data.frame(u = character(0), v = character(0)), vertices = state\_nodes %>% select(county, city\_name, lat, lon) %>% rename(y = lat, x = lon), directed = FALSE)  
 return(graph)  
 }else{  
 nodes = state\_nodes %>%  
 select(county, city\_name, lat, lon, state) %>%  
 rename(y = lat, x = lon) %>%  
 distinct()  
 edges = state\_edges %>%  
 select(county1, county2, dist) %>%  
 distinct()  
 graph = graph\_from\_data\_frame(d = edges, vertices = nodes, directed = FALSE)  
 return(graph)  
 }  
}  
  
#leggi nodi  
state\_dirs = list.dirs("data/roads", recursive = FALSE)  
nodes = map(state\_dirs, ~ {  
 csv\_path = file.path(.x, "nodes.csv")  
 if(file.exists(csv\_path)){  
 read\_csv(csv\_path, show\_col\_types = FALSE)  
 }else{  
 NULL  
 }  
})  
names(nodes) = basename(state\_dirs)  
  
#leggi archi  
state\_dirs = list.dirs("data/roads", recursive = FALSE)  
edges = map(state\_dirs, ~ {  
 csv\_path = file.path(.x, "edges.csv")  
 if(file.exists(csv\_path)){  
 read\_csv(csv\_path, show\_col\_types = FALSE)  
 }else{  
 NULL  
 }  
})  
names(edges) = basename(state\_dirs)  
global\_csv\_path = file.path("data/roads", "edges.csv")  
edges[["global"]] = read\_csv(global\_csv\_path, show\_col\_types = FALSE)  
  
#pulizia dati  
words\_to\_remove = c("County", "Borough", "Parish", "Planning Region")  
  
nodes$MD = nodes$MD %>%  
 mutate(county = ifelse(county == "Baltimore", paste0(county, " City"), county))  
edges$MD = edges$MD %>%  
 mutate(county1 = ifelse(county1 == "Baltimore", paste0(county1, " City"), county1), county2 = ifelse(county2 == "Baltimore", paste0(county2, " City"), county2))  
  
nodes$MO = nodes$MO %>%  
 mutate(county = ifelse(county == "Saint Louis", paste0(county, " City"), county))  
edges$MO = edges$MO %>%  
 mutate(county1 = ifelse(county1 == "Saint Louis", paste0(county1, " City"), county1), county2 = ifelse(county2 == "Saint Louis", paste0(county2, " City"), county2))  
  
nodes$VA = nodes$VA %>%  
 mutate(county = ifelse(county == "Roanoke", paste0(county, " City"), county))  
edges$VA = edges$VA %>%  
 mutate(county1 = ifelse(county1 == "Roanoke", paste0(county1, " City"), county1), county2 = ifelse(county2 == "Roanoke", paste0(county2, " City"), county2))  
  
nodes$VA = nodes$VA %>%  
 mutate(county = ifelse(county == "Franklin", paste0(county, " City"), county))  
edges$VA = edges$VA %>%  
 mutate(county1 = ifelse(county1 == "Franklin", paste0(county1, " City"), county1), county2 = ifelse(county2 == "Franklin", paste0(county2, " City"), county2))  
  
nodes = map(nodes, ~ .x %>%  
 mutate(across(c(county, city\_name), ~ str\_remove\_all(.x, str\_c(words\_to\_remove, collapse = "|")) %>% str\_trim()))  
)  
  
edges = map(edges, ~ if(nrow(.x) > 0 && ncol(.x) > 0){  
 .x %>%  
 mutate(across(c(county1, county2), ~ str\_remove\_all(.x, str\_c(words\_to\_remove, collapse = "|")) %>% str\_trim()))  
 }else{  
 .x  
 }  
)  
  
#equirectangular projection  
nodes = map(nodes, ~ .x %>%  
 st\_as\_sf(coords = c("lon", "lat"), crs = 4326) %>%  
 st\_transform(crs = "+proj=eqc") %>%  
 mutate(lon = st\_coordinates(.)[, 1], lat = st\_coordinates(.)[, 2]) %>%  
 as.data.frame()  
)  
  
#distanze  
edges[1:(length(edges) - 1)] = map2(edges[1:(length(edges) - 1)], nodes, ~ {  
 if(nrow(.x) > 0){  
 .x %>%  
 left\_join(.y, by = c("county1" = "county")) %>%  
 rename(y\_u = lat, x\_u = lon) %>%  
 left\_join(.y, by = c("county2" = "county")) %>%  
 rename(y\_v = lat, x\_v = lon) %>%  
 mutate(dist = sqrt((x\_v - x\_u)^2 + (y\_v - y\_u)^2)) %>%  
 select(county1, county2, dist)  
 }else{  
 .x  
 }  
})  
  
edges$global = edges$global %>%  
 left\_join(states, by = c("state1" = "name")) %>%  
 rename(state1\_abbr = abbr) %>%  
 left\_join(states, by = c("state2" = "name")) %>%  
 rename(state2\_abbr = abbr) %>%  
 select(-state1, -state2) %>%  
 rename(state1 = state1\_abbr, state2 = state2\_abbr)  
  
edges$global = edges$global %>%  
 left\_join(bind\_rows(nodes, .id = "state"), by = c("state1" = "state", "county1" = "county")) %>%  
 rename(y\_u = lat, x\_u = lon) %>%  
 left\_join(bind\_rows(nodes, .id = "state"), by = c("state2" = "state", "county2" = "county")) %>%  
 rename(y\_v = lat, x\_v = lon) %>%  
 mutate(dist = sqrt((x\_v - x\_u)^2 + (y\_v - y\_u)^2)) %>%  
 select(state1, county1, state2, county2, dist)  
  
nodes = map(nodes, ~ .x %>%  
 mutate(state = "")  
)  
  
#creazione grafi  
roads\_by\_state = map2(nodes, edges[1:(length(edges) - 1)], create\_state\_graph)  
  
nodes\_global = bind\_rows(imap\_dfr(nodes, ~ .x %>%  
 mutate(county = paste(.y, .x$county, sep = " - "), state = .y) %>%  
 select(county, city\_name, lat, lon, state) %>%  
 distinct()))  
  
edges\_global = bind\_rows(imap\_dfr(edges[1:(length(edges) - 1)], ~ {  
 if(nrow(.x) > 0){  
 .x %>%  
 mutate(county1 = paste(.y, .x$county1, sep = " - "),  
 county2 = paste(.y, .x$county2, sep = " - ")) %>%  
 select(county1, county2, dist) %>%  
 distinct()  
 }else{  
 data.frame(county1 = character(0), county2 = character(0), dist = numeric(0))  
 }  
 }),  
 bind\_rows(  
 edges$global %>%  
 mutate(county1 = paste(.data$state1, .data$county1, sep = " - "),  
 county2 = paste(.data$state2, .data$county2, sep = " - ")) %>%  
 select(county1, county2, dist) %>%  
 distinct()  
 ))  
  
roads = create\_state\_graph(nodes\_global, edges\_global)  
  
roads\_by\_state\_nodes = nodes  
roads\_by\_state\_edges = edges  
roads\_nodes = nodes\_global  
roads\_edges = edges\_global  
rm(nodes)  
rm(edges)  
rm(nodes\_global)  
rm(edges\_global)

#local  
  
centrality = function(g, state){  
 data.frame(  
 state = state,  
 node = V(g)$name,  
 degree = degree(g),  
 degreeW = strength(g, weights = E(g)$dist),  
 closeness = closeness(g, weights = (E(g)$dist+0.000001), normalized = TRUE),  
 betweenness = betweenness(g, weights = (E(g)$dist+0.000001), normalized = TRUE),  
 eigen = eigen\_centrality(g, weights = E(g)$dist)$vector,  
 katz = alpha\_centrality(g, weights = E(g)$dist),  
 pagerank = page\_rank(g, weights = E(g)$dist)$vector #funziona su diretti  
 #hits\_a = authority\_score(g, weights = E(g)$dist)$vector, #funziona solo su diretti  
 #hits\_h = hub\_score(g, weights = E(g)$dist)$vector #funziona solo su diretti  
 )  
}  
  
similarity = function(g, mode = "col"){  
 if(ecount(g) == 0){  
 return(list(cosine = NA, pearson = NA, global = NA))  
 }  
 A = as\_adjacency\_matrix(g, attr = "dist", sparse = FALSE)  
 if(mode == "row"){  
 A = t(A)  
 }  
 cosine = function(A){  
 euclidean = function(x){  
 sqrt(x %\*% x)  
 }  
 D = diag(1/apply(A, 2, euclidean))  
 S = D %\*% t(A) %\*% A %\*% D  
 return(S)  
 }  
 global = function(A){  
 S = solve(diag(1, vcount(g)) - 0.85 / max(abs(eigen(A)$values)) \* A)  
 S = S - diag(diag(S))  
 return(S)  
 }  
 return(list(cosine = cosine(A), pearson = cor(A), global = global(A)))  
}  
  
heterogeneity = function(g, mode = "col"){  
 if(ecount(g) == 0){  
 return(list(shannon = NA, simpson = NA))  
 }  
 A = as\_adjacency\_matrix(g, attr = "dist", sparse = FALSE)  
 D = A  
 if(mode == "col"){  
 A = A %\*% diag(1/colSums(A))  
 dim = 2   
 }else{  
 A = diag(1/rowSums(A)) %\*% A  
 dim = 1   
 }  
 shannon = function(p){  
 x = p \* log2(p)  
 x = replace(x, is.nan(x), 0)  
 return(-sum(x))  
 }  
 simpson = function(p){  
 x = 1 - sum(p \* p)  
 return(x)  
 }  
 # rao = function(p, D){ #lento su grafi grandi  
 # x = diag(p) %\*% D %\*% diag(p)  
 # return(sum(c(x)))  
 # }  
 return(list(shannon = apply(A, dim, shannon), simpson = apply(A, dim, simpson)))  
}  
  
#group  
  
communities = function(g){  
 methods = list(  
 "edge\_betweenness" = cluster\_edge\_betweenness,  
 "fast\_greedy" = cluster\_fast\_greedy,  
 "label\_prop" = cluster\_label\_prop,  
 "leading\_eigen" = cluster\_leading\_eigen,  
 "louvain" = cluster\_louvain,  
 "walktrap" = cluster\_walktrap,  
 #"spinglass" = function(graph) cluster\_spinglass(graph, spins = 10), #funziona solo su connesso  
 "infomap" = cluster\_infomap  
 #"optimal" = cluster\_optimal #lento su grafi grandi  
 )  
 do.call(rbind, lapply(names(methods), function(method) {  
 tryCatch(  
 data.frame(method = method, result = I(list(methods[[method]](g)))),  
 error = function(e) {  
 message("Error with method: ", method)  
 data.frame(method = method, result = I(list(NA)))  
 }  
 )  
 }))  
}  
  
clustering = function(g){  
 methods = c("average", "centroid", "single", "complete")  
 d = as.dist(as\_adjacency\_matrix(g, attr = "dist", sparse = FALSE))  
 do.call(rbind, lapply(methods, function(method){  
 data.frame(method = method, result = I(list(hclust(d, method = method))))  
 }))  
}  
  
#global  
  
connettivity = function(g){  
 list(  
 components = components(g),  
 biconnected\_components = biconnected\_components(g),  
 cohesive\_blocks = cohesive\_blocks(g)  
 )  
}  
  
resilience = function(g){  
 percolate = function(g, size, d){  
 giant = vector()  
 c = components(g)  
 giant[1] = max(c$csize)  
 names(d) = 1:length(d)  
 d = sort(d, decreasing=TRUE)  
 vital = as.integer(names(d[1:size]))  
 for (i in 1:size) {  
 c = components(delete\_vertices(g, vital[1:i]))  
 giant[i+1] = max(c$csize)  
 }  
 giant  
 }  
 size = floor(vcount(g)/2)  
 c = centrality(g, "")  
 data.frame(  
 rand = percolate(g, size, d = sample(V(g), size)),  
 degree = percolate(g, size, d = c$degree),  
 degreeW = percolate(g, size, d = c$degreeW),  
 closeness = percolate(g, size, d = c$closeness),  
 betweenness = percolate(g, size, d = c$betweenness),  
 eigen = percolate(g, size, d = c$eigen),  
 katz = percolate(g, size, d = c$katz),  
 pagerank = percolate(g, size, d = c$pagerank)  
 )  
}  
  
geodesic = function(g, state){  
 list(  
 geodesic\_mean = mean\_distance(g),  
 geodesic\_diameter = diameter(g),  
 geodesic\_diameter\_nodes = get\_diameter(g)  
 )  
}  
  
power\_law = function(g){  
 ccdf = function(d){  
 p = rep(0, max(d))  
 for (i in 1:length(p)) {  
 p[i] = length(d[d >= i]) / length(d)  
 }   
 p  
 }  
 d = degree(g)  
 list(  
 degreeD = d,  
 distD = distances(g)[is.finite(distances(g))],  
 summary = summary(d),  
 skewness = skewness(d),  
 ccdf = ccdf(d)  
 )  
}  
  
assortativita = function(g, m, v){  
 data.frame(  
 assortativityE = modularity(g, membership = m),  
 assortativityS = assortativity(roads, v),  
 assortativityD = assortativity\_degree(g)  
 )  
}  
  
motif = function(g){  
 data.frame(  
 transitivity = transitivity(g, type = "global")  
 )  
}

# Analisi singoli stati

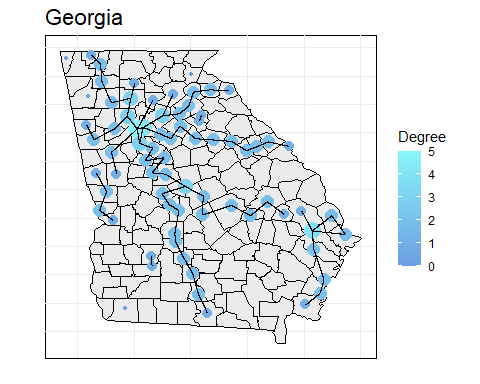
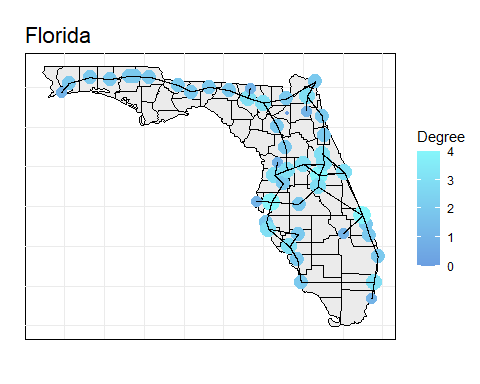
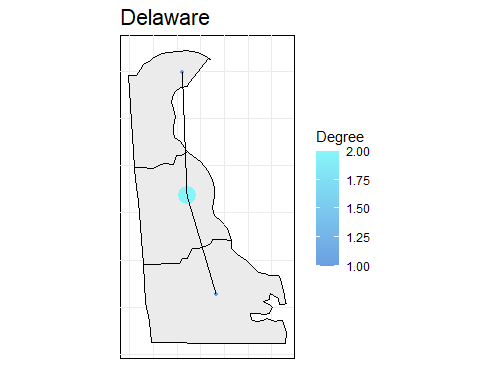
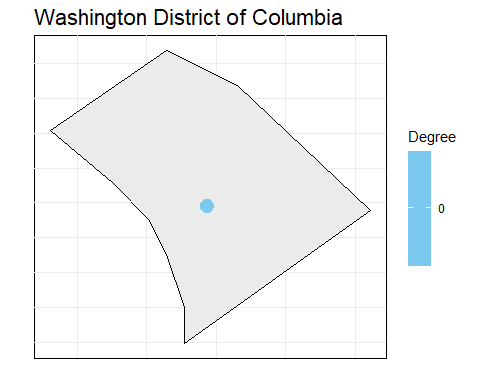
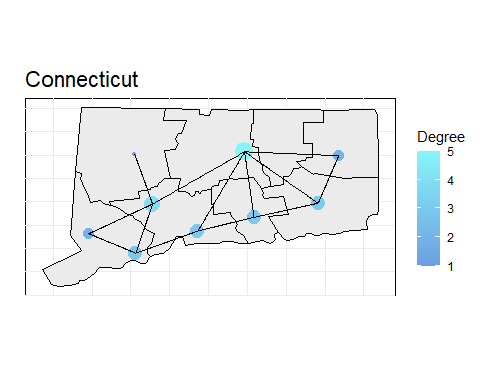
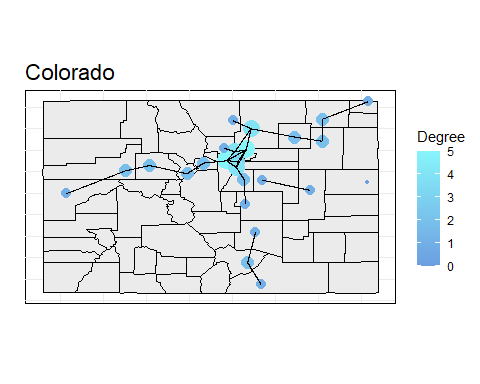
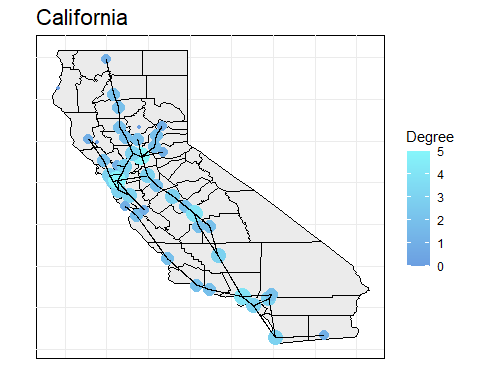
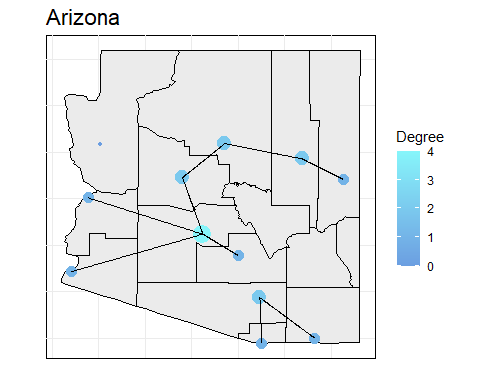
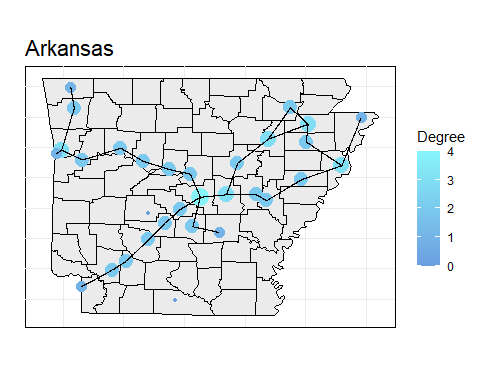
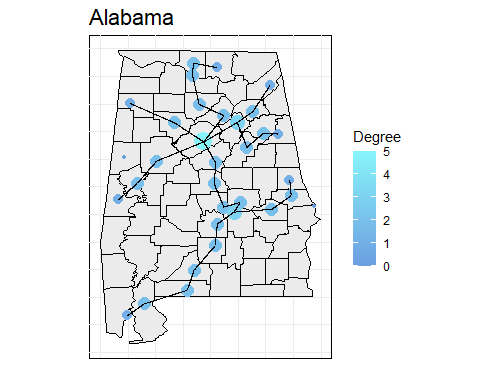
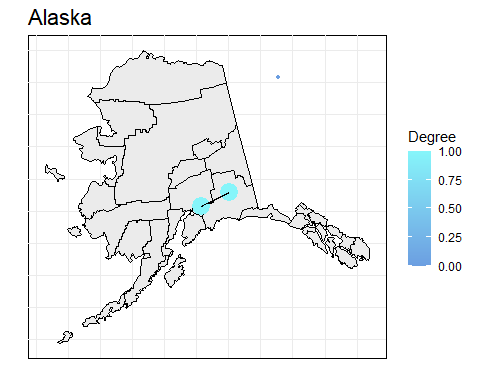
for(state in names(roads\_by\_state)){  
 state\_map = map\_data("county") %>%  
 filter(region == ifelse(state == "DC", tolower("District of Columbia"), states %>% filter(abbr %in% state) %>% pull(name) %>% tolower())) %>%  
 #equirectangular projection  
 st\_as\_sf(coords = c("long", "lat"), crs = 4326) %>%  
 st\_transform(crs = "+proj=eqc") %>%  
 mutate(x = st\_coordinates(.)[, 1], y = st\_coordinates(.)[, 2]) %>%  
 as.data.frame()  
 if(state == "AK"){  
 state\_map = read.csv("data/outline\_AK\_HI\_county.csv") %>%  
 filter(abbr == "AK") %>%  
 mutate(x = x-15550000, y = y+8850000)  
 }  
 if(state == "HI"){  
 state\_map = read.csv("data/outline\_AK\_HI\_county.csv") %>%  
 filter(abbr == "HI") %>%  
 mutate(x = x-17100000, y = y+4400000)  
 }  
 print(  
 roads\_by\_state[[state]] %>%  
 as\_tbl\_graph() %>%  
 activate(nodes) %>%  
 mutate(degree = centrality\_degree()) %>%  
 ggraph(layout = 'manual', x = x, y = y) +   
 geom\_polygon(data = state\_map, aes(x = x, y = y, group = group), fill = "#ebebeb", color="black") +  
 geom\_node\_point(aes(size = degree/2, colour = degree)) +  
 geom\_edge\_link() +  
 #geom\_node\_text(aes(label = ifelse(degree > 3, name, ""))) +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_gradient(low="#6B9EE1", high="#86F5FA") +  
 scale\_size\_continuous(guide = "none") +  
 labs(title = states %>% filter(abbr %in% state) %>% pull(name), x = "", y = "", size = "", color = "Degree") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))  
 )  
}

## Warning in min(cc[[1]], na.rm = TRUE): nessun argomento non-mancante al minimo;  
## si restituisce Inf

## Warning in min(cc[[2]], na.rm = TRUE): nessun argomento non-mancante al minimo;  
## si restituisce Inf

## Warning in max(cc[[1]], na.rm = TRUE): nessun argomento non-mancante al  
## massimo; si restituisce -Inf

## Warning in max(cc[[2]], na.rm = TRUE): nessun argomento non-mancante al  
## massimo; si restituisce -Inf

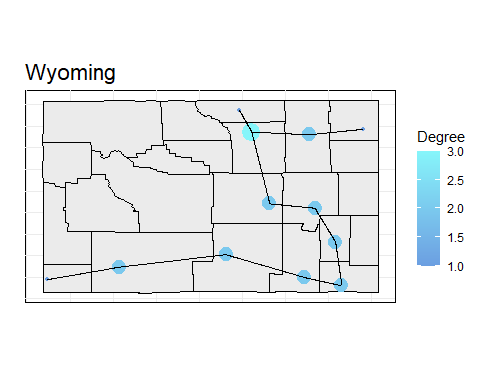
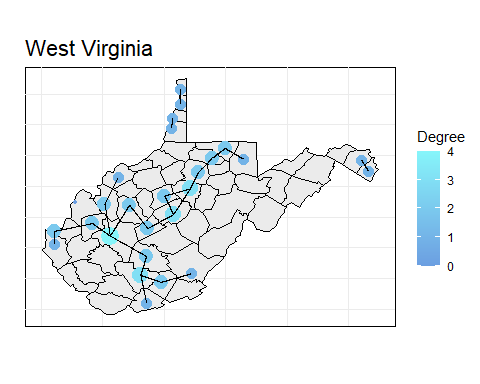
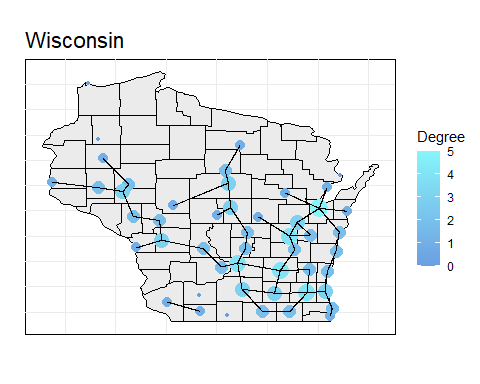
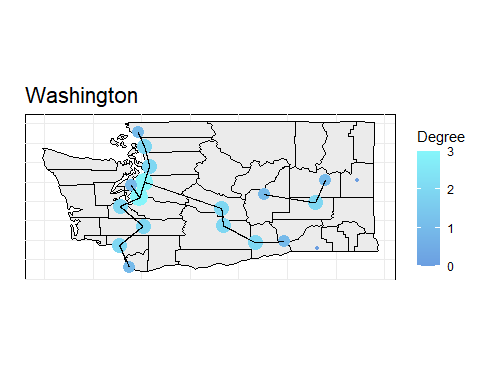
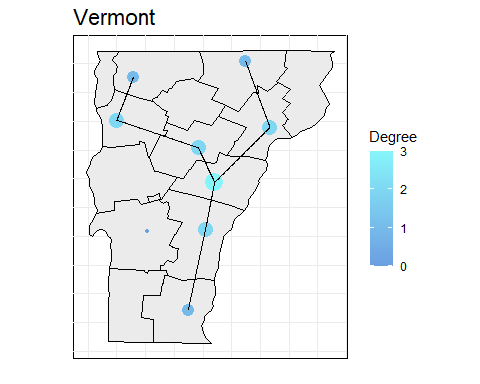
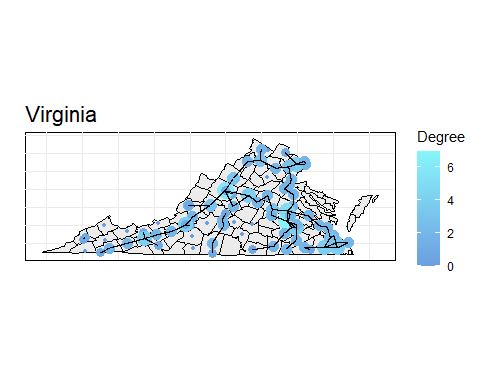
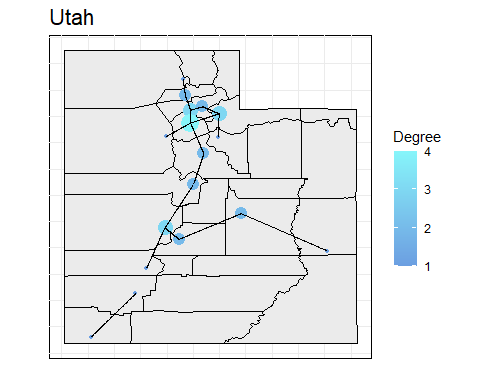
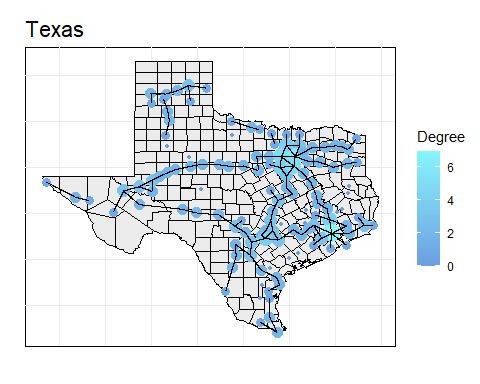
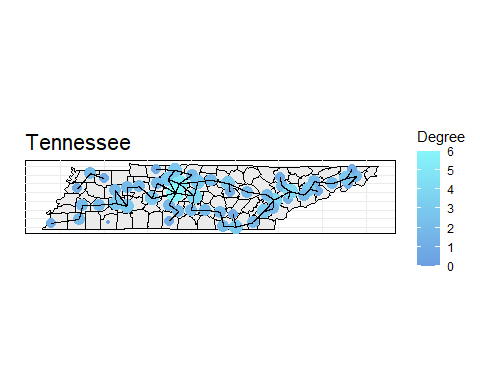
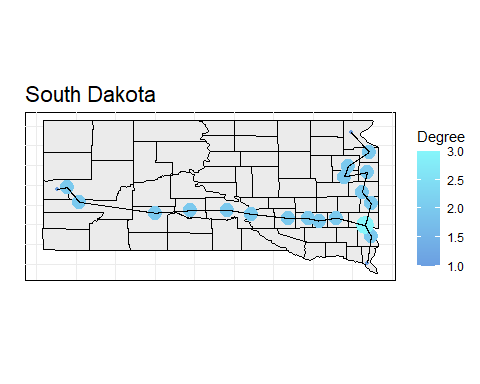
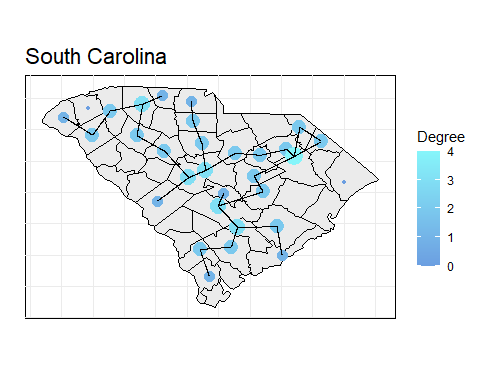
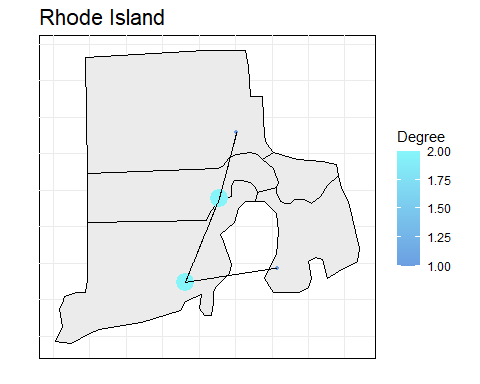
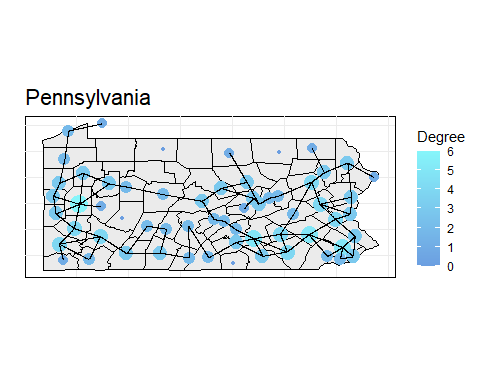
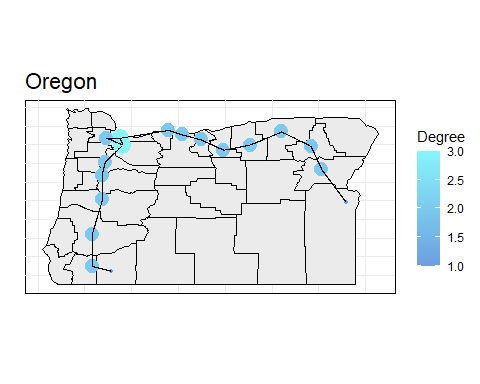
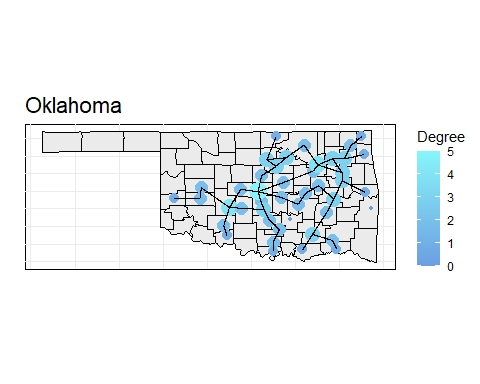
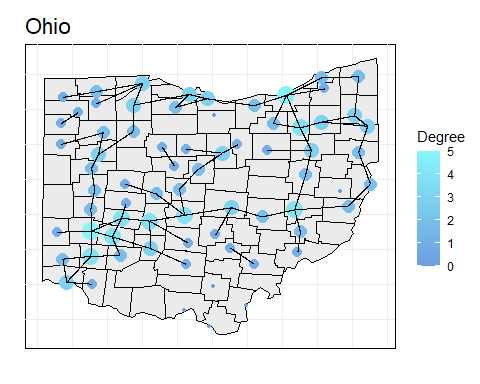
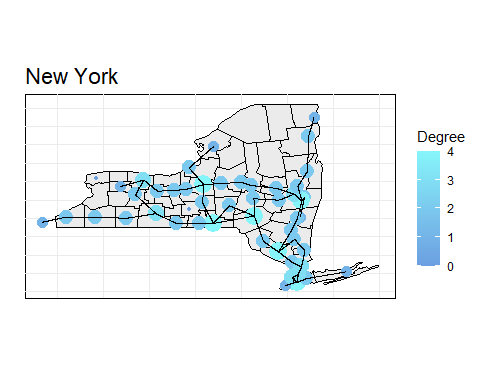
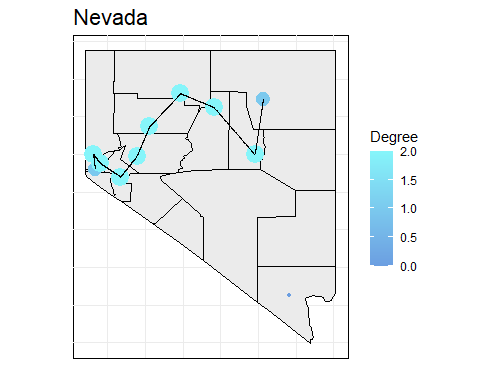
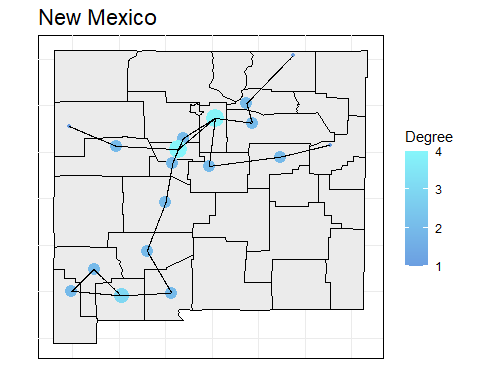
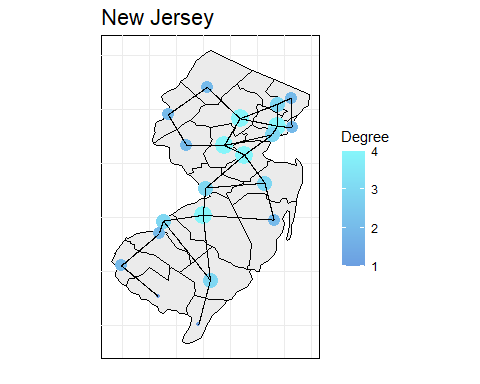
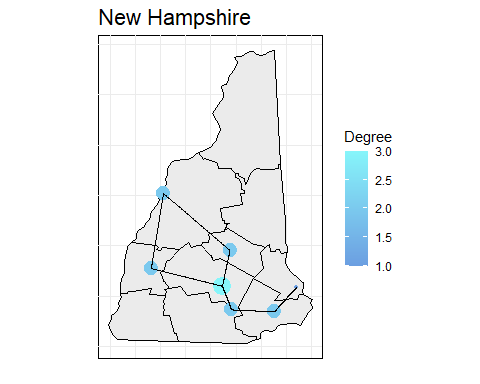
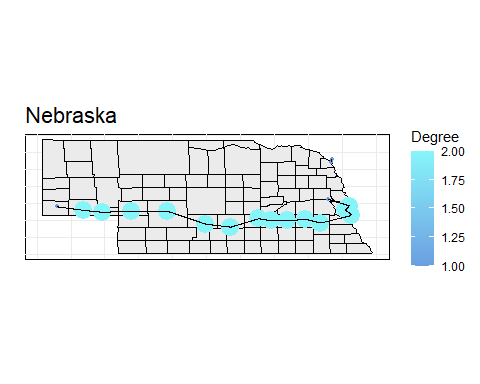
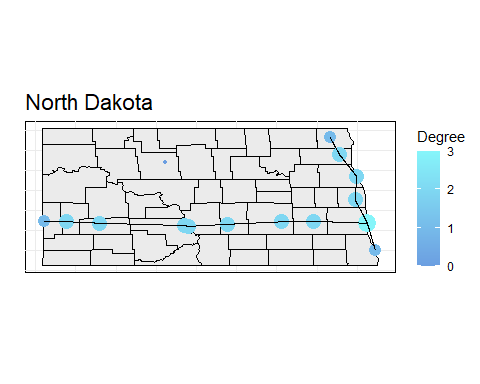
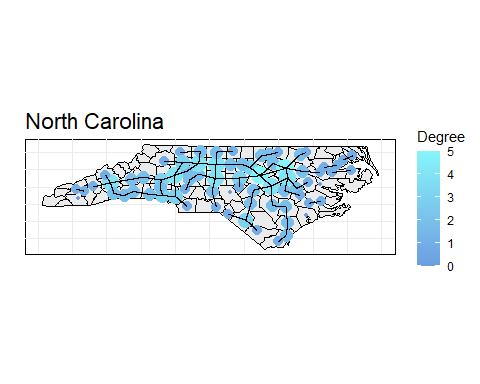
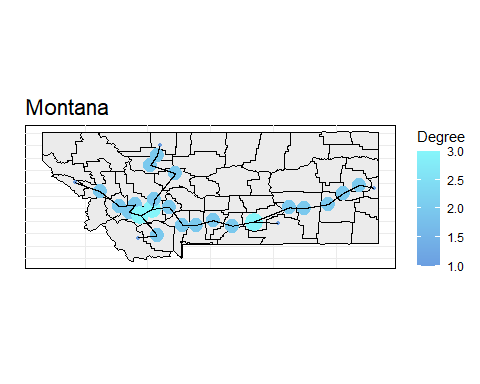
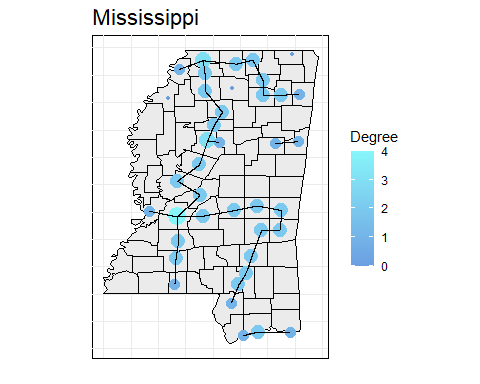
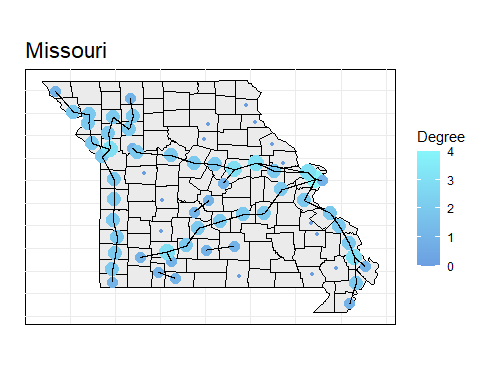
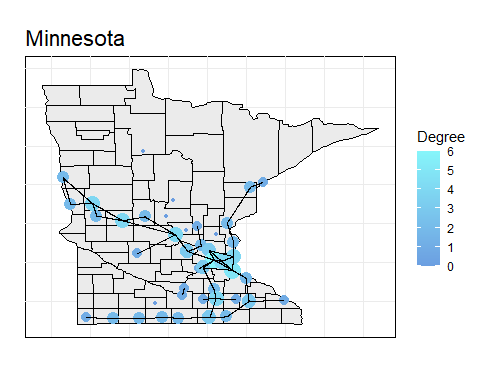
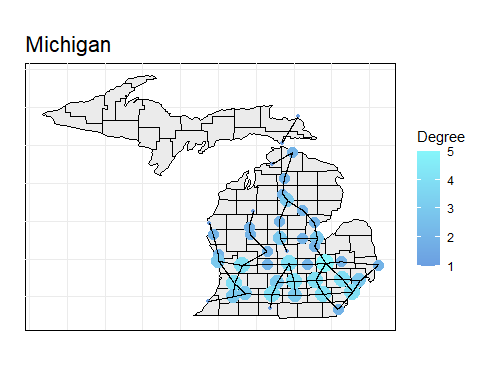
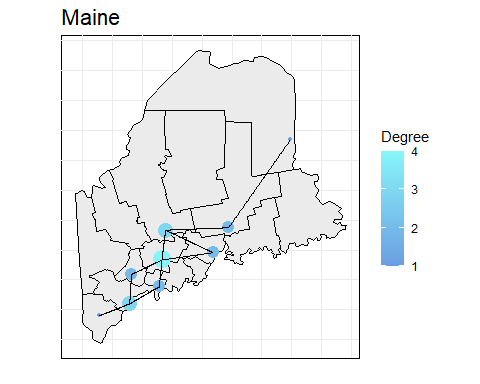
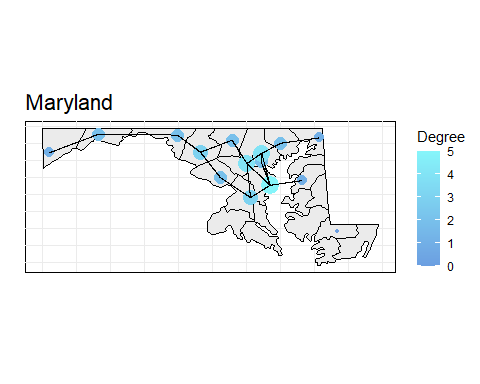
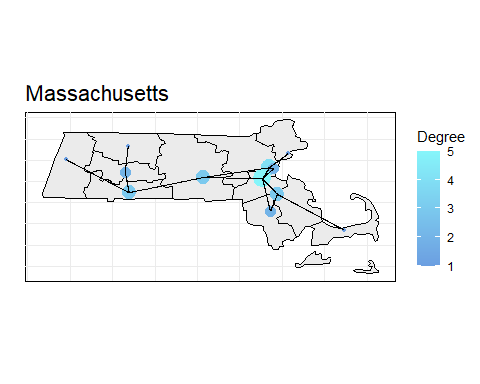
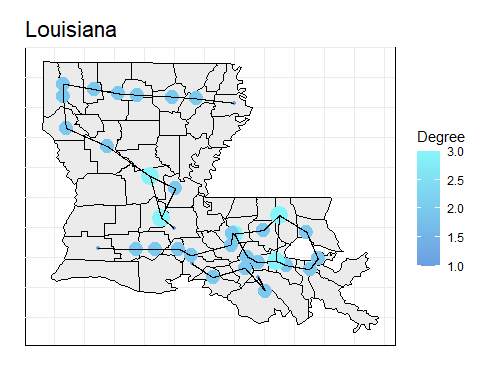
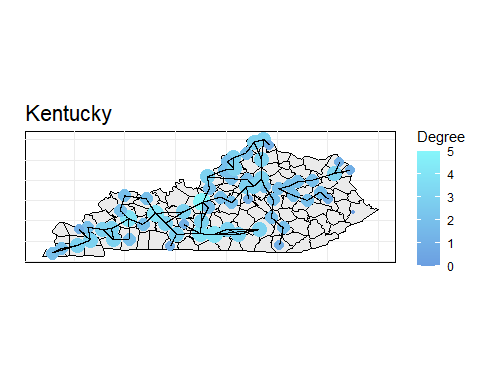
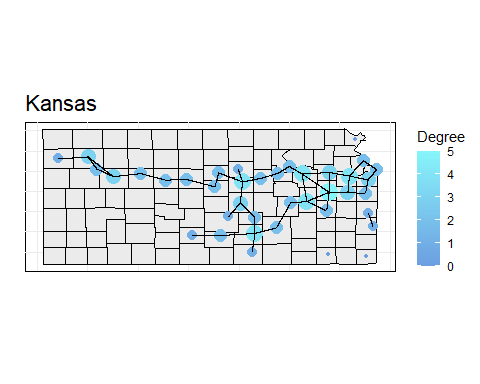
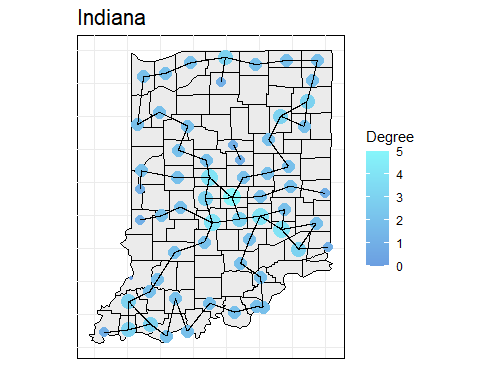
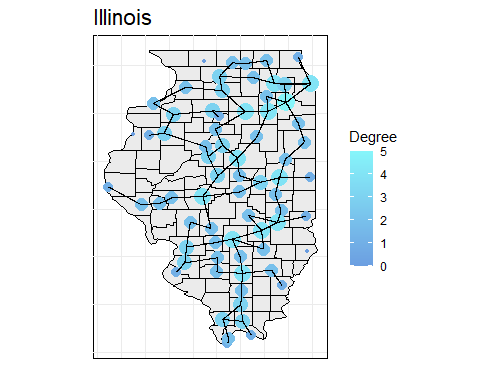
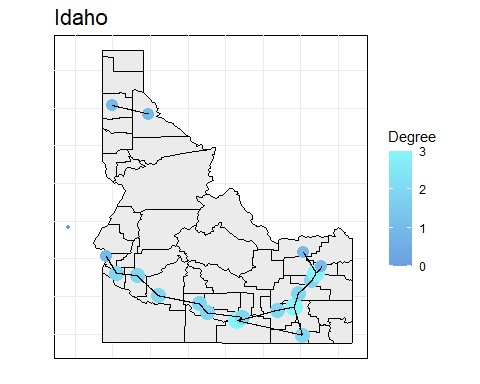
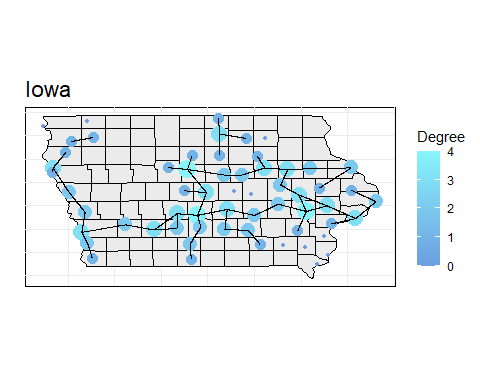
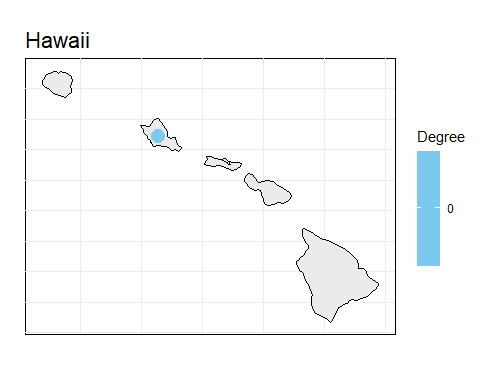


## Warning in min(cc[[1]], na.rm = TRUE): nessun argomento non-mancante al minimo;  
## si restituisce Inf

## Warning in min(cc[[2]], na.rm = TRUE): nessun argomento non-mancante al minimo;  
## si restituisce Inf

## Warning in max(cc[[1]], na.rm = TRUE): nessun argomento non-mancante al  
## massimo; si restituisce -Inf

## Warning in max(cc[[2]], na.rm = TRUE): nessun argomento non-mancante al  
## massimo; si restituisce -Inf



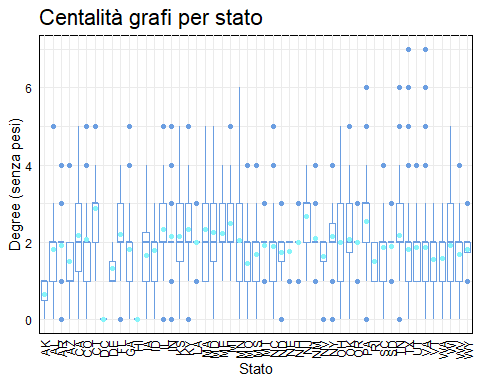
Si può notare come molti stati abbiano una topologia circolare (California, Connecticut, Florida, Massachusetts, Missouri, Mississippi, Montana, North Carolina, New Mexico, South Carolina, Utah, Virginia) o a stella (Alabama, Arkansas, Arizona, Colorado, Georgia, Iowa, Illinois, Indiana, Kansas, Kentucky, Michigan, Minnesota, New Jersey, New York, Ohio, Pennsylvania, Texas).

È da notare che ci sono alcuni stati con pochi nodi (Alaska, Washington District of Columbia, Delaware, Hawaii, Rhode Island), questo è dovuto al fatto che sono stati con poche contee o remoti, e con alcuni paesi con nodi non connessi, questo probabilmente è dovuto alle assunzioni fatte, sicuramente sono collegati per esempio tramite una strada statale.

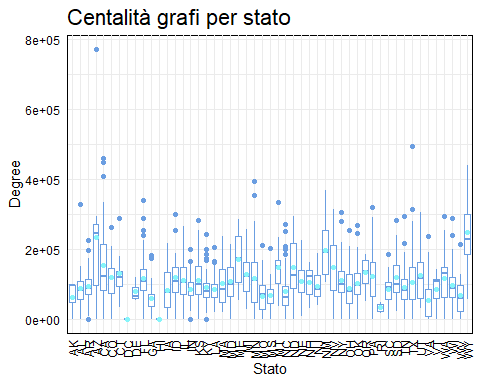
Di seguito l’analisi locale sulle reti di ogni stato.

## (Distribuzione delle) Centralità

rbs\_c = do.call(rbind, lapply(names(roads\_by\_state), function(state){  
 centrality(roads\_by\_state[[state]], state)  
}))  
  
rbs\_c %>%  
 ggplot(aes(x=state, y=degree)) +  
 geom\_boxplot(color="#6B9EE1") +  
 geom\_point(data = rbs\_c %>% group\_by(state) %>% summarize(mean\_degree = mean(degree)), aes(x = state, y = mean\_degree, group = 1), color = "#86F5FA") +   
 scale\_y\_continuous() +  
 labs(title = "Centalità grafi per stato", x = "Stato", y = "Degree (senza pesi)") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000", angle = 90, vjust = 0.5, hjust = 1), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



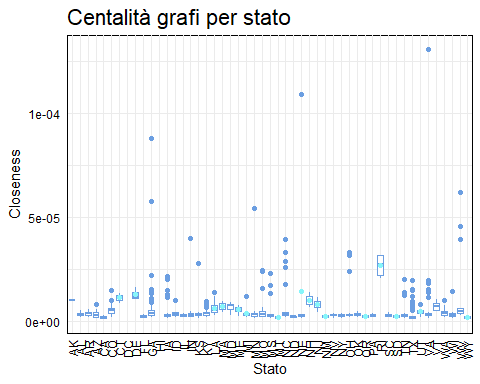
rbs\_c %>%  
 ggplot(aes(x=state, y=degreeW)) +  
 geom\_boxplot(color="#6B9EE1") +  
 geom\_point(data = rbs\_c %>% group\_by(state) %>% summarize(mean\_degreeW = mean(degreeW)), aes(x = state, y = mean\_degreeW, group = 1), color = "#86F5FA") +   
 scale\_y\_continuous() +  
 labs(title = "Centalità grafi per stato", x = "Stato", y = "Degree") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000", angle = 90, vjust = 0.5, hjust = 1), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



rbs\_c %>%  
 ggplot(aes(x=state, y=closeness)) +  
 geom\_boxplot(color="#6B9EE1") +  
 geom\_point(data = rbs\_c %>% group\_by(state) %>% summarize(mean\_closeness = mean(closeness)), aes(x = state, y = mean\_closeness, group = 1), color = "#86F5FA") +   
 scale\_y\_continuous() +  
 labs(title = "Centalità grafi per stato", x = "Stato", y = "Closeness") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000", angle = 90, vjust = 0.5, hjust = 1), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))

## Warning: Removed 121 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).

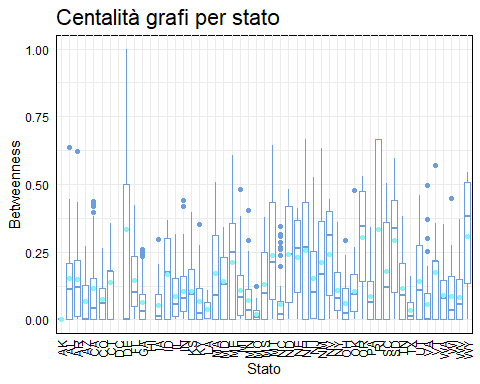
## Warning: Removed 35 rows containing missing values or values outside the scale range  
## (`geom\_point()`).



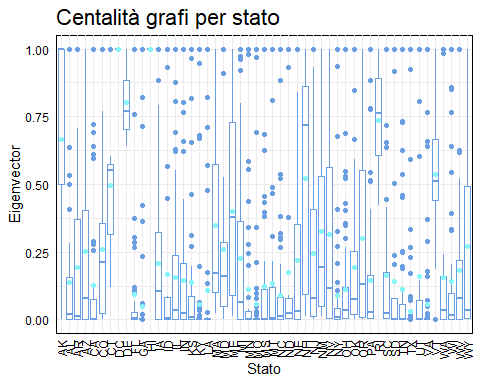
rbs\_c %>%  
 ggplot(aes(x=state, y=betweenness)) +  
 geom\_boxplot(color="#6B9EE1") +  
 geom\_point(data = rbs\_c %>% group\_by(state) %>% summarize(mean\_betweenness = mean(betweenness)), aes(x = state, y = mean\_betweenness, group = 1), color = "#86F5FA") +   
 scale\_y\_continuous() +  
 labs(title = "Centalità grafi per stato", x = "Stato", y = "Betweenness") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000", angle = 90, vjust = 0.5, hjust = 1), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))

## Warning: Removed 2 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).

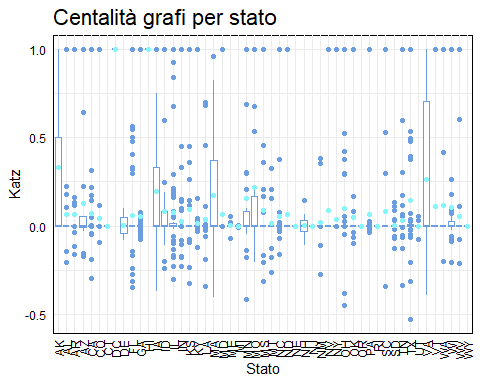
## Warning: Removed 2 rows containing missing values or values outside the scale range  
## (`geom\_point()`).



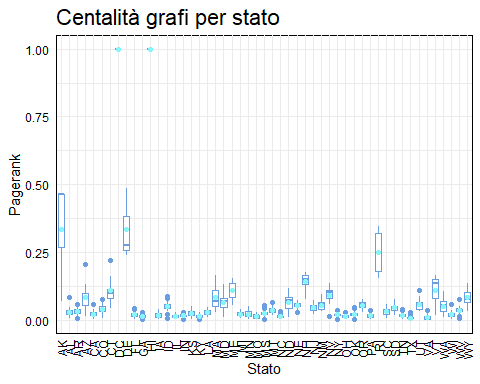
rbs\_c %>%  
 ggplot(aes(x=state, y=eigen)) +  
 geom\_boxplot(color="#6B9EE1") +  
 geom\_point(data = rbs\_c %>% group\_by(state) %>% summarize(mean\_eigen = mean(eigen)), aes(x = state, y = mean\_eigen, group = 1), color = "#86F5FA") +   
 scale\_y\_continuous() +  
 labs(title = "Centalità grafi per stato", x = "Stato", y = "Eigenvector") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000", angle = 90, vjust = 0.5, hjust = 1), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



rbs\_c %>%  
 ggplot(aes(x=state, y=katz)) +  
 geom\_boxplot(color="#6B9EE1") +  
 geom\_point(data = rbs\_c %>% group\_by(state) %>% summarize(mean\_katz = mean(katz)), aes(x = state, y = mean\_katz, group = 1), color = "#86F5FA") +   
 scale\_y\_continuous() +  
 labs(title = "Centalità grafi per stato", x = "Stato", y = "Katz") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000", angle = 90, vjust = 0.5, hjust = 1), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



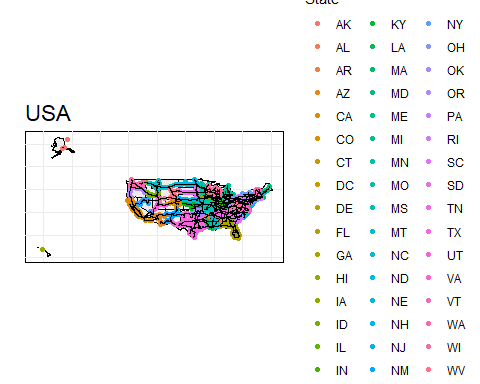
rbs\_c %>%  
 ggplot(aes(x=state, y=pagerank)) +  
 geom\_boxplot(color="#6B9EE1") +  
 geom\_point(data = rbs\_c %>% group\_by(state) %>% summarize(mean\_pagerank = mean(pagerank)), aes(x = state, y = mean\_pagerank, group = 1), color = "#86F5FA") +   
 scale\_y\_continuous() +  
 labs(title = "Centalità grafi per stato", x = "Stato", y = "Pagerank") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000", angle = 90, vjust = 0.5, hjust = 1), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



Si può notare come per quasi tutte le metriche di centralità i grafi sui singoli stati abbiano mediamente una bassa centralità (grado intorno al 2, closeness quasi zero, katz e pagerank molti sono vicino a zero).

# Analisi USA

state\_map = bind\_rows(  
 map\_data("state") %>%  
 #equirectangular projection  
 st\_as\_sf(coords = c("long", "lat"), crs = 4326) %>%  
 st\_transform(crs = "+proj=eqc") %>%  
 mutate(x = st\_coordinates(.)[, 1], y = st\_coordinates(.)[, 2]) %>%  
 as.data.frame() %>%  
 select(group, region, x, y),  
 read.csv("data/outline\_AK\_HI\_state.csv") %>%  
 mutate(x = ifelse(abbr == "AK", x - 15550000, x), y = ifelse(abbr == "AK", y + 8850000, y)) %>%  
 mutate(x = ifelse(abbr == "HI", x - 17100000, x), y = ifelse(abbr == "HI", y + 4400000, y)) %>%  
 rename(region = full) %>%  
 select(group, region, x, y)  
)  
  
roads %>%  
 as\_tbl\_graph() %>%  
 activate(nodes) %>%  
 mutate(degree = centrality\_degree()) %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_polygon(data = state\_map, aes(x = x, y = y, group = group), fill = "#ebebeb", color="black") +  
 geom\_node\_point(aes(colour = state)) +  
 geom\_edge\_link() +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 labs(title = "USA", x = "", y = "", color = "State") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))

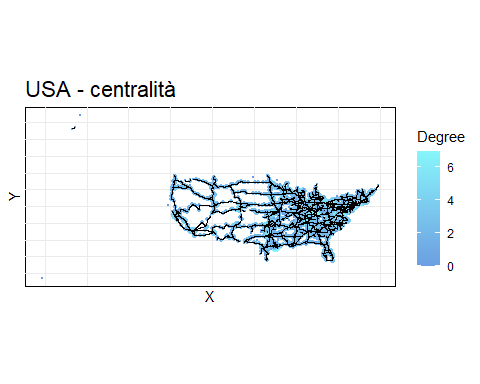


Si può osservare come il grafo sia più connesso rispetto ai singoli stati (anche alcune contee prima disconnesse ora sono connesse), si può notare come la parte ovest sia più rada mentre la parte est sia più densa.

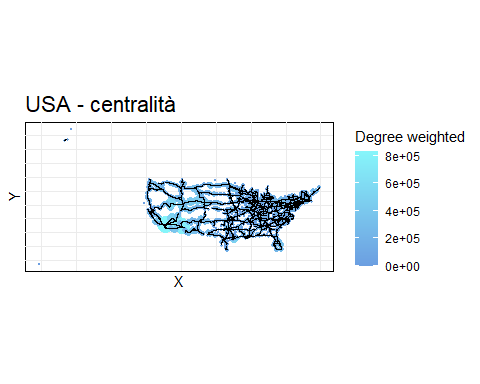
Di seguito l’analisi locale sulle reti di ogni stato.

## Locale - centralità

r\_c = centrality(roads, "USA")  
  
roads %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(size = I(r\_c$degree/2), colour = r\_c$degree)) +  
 geom\_edge\_link() +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_gradient(low="#6B9EE1", high="#86F5FA") +  
 scale\_size\_continuous(guide = "none") +  
 labs(title = "USA - centralità", x = "X", y = "Y", size = "", colour = "Degree") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))

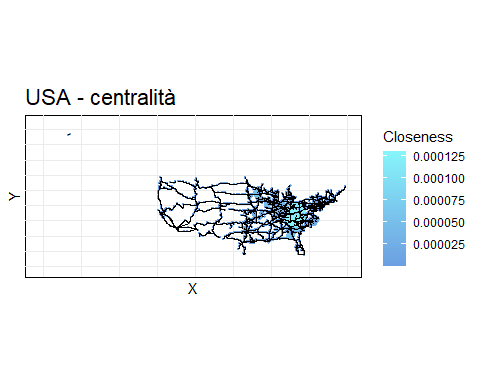


roads %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(size = I(r\_c$degreeW/150000), colour = r\_c$degreeW)) +  
 geom\_edge\_link() +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_gradient(low="#6B9EE1", high="#86F5FA") +  
 scale\_size\_continuous(guide = "none") +  
 labs(title = "USA - centralità", x = "X", y = "Y", size = "", colour = "Degree weighted") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))

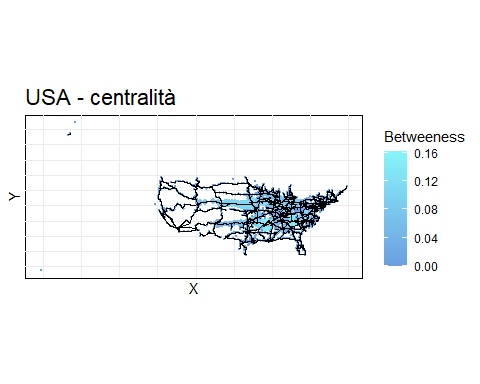


roads %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(size = I(r\_c$closeness\*50000), colour = r\_c$closeness)) +  
 geom\_edge\_link() +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_gradient(low="#6B9EE1", high="#86F5FA") +  
 scale\_size\_continuous(guide = "none") +  
 labs(title = "USA - centralità", x = "X", y = "Y", size = "", colour = "Closeness") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))

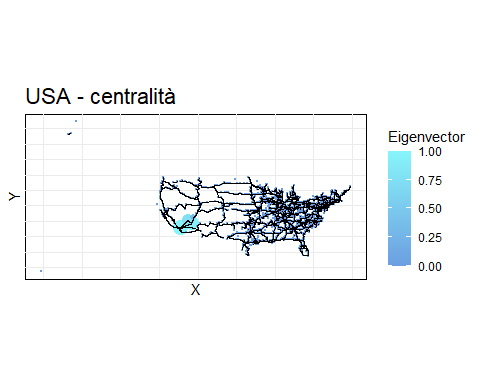
## Warning: Removed 97 rows containing missing values or values outside the scale range  
## (`geom\_point()`).



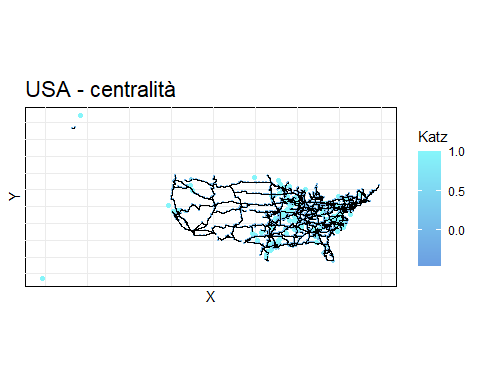
roads %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(size = I(r\_c$betweenness\*25), colour = r\_c$betweenness)) +  
 geom\_edge\_link() +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_gradient(low="#6B9EE1", high="#86F5FA") +  
 scale\_size\_continuous(guide = "none") +  
 labs(title = "USA - centralità", x = "X", y = "Y", size = "", colour = "Betweeness") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



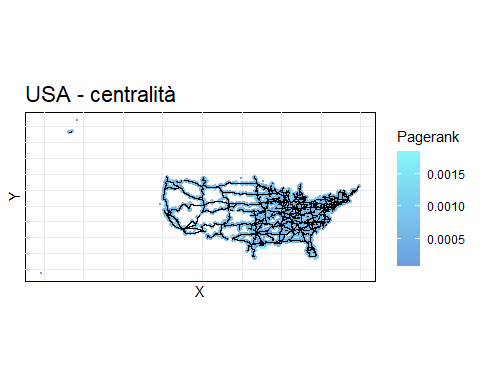
roads %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(size = I(r\_c$eigen\*5), colour = r\_c$eigen)) +  
 geom\_edge\_link() +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_gradient(low="#6B9EE1", high="#86F5FA") +  
 scale\_size\_continuous(guide = "none") +  
 labs(title = "USA - centralità", x = "X", y = "Y", size = "", colour = "Eigenvector") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



roads %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(size = I(r\_c$katz+0.5), colour = r\_c$katz)) +  
 geom\_edge\_link() +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_gradient(low="#6B9EE1", high="#86F5FA") +  
 scale\_size\_continuous(guide = "none") +  
 labs(title = "USA - centralità", x = "X", y = "Y", size = "", colour = "Katz") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



roads %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(size = I(r\_c$pagerank\*2000), colour = r\_c$pagerank)) +  
 geom\_edge\_link() +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_gradient(low="#6B9EE1", high="#86F5FA") +  
 scale\_size\_continuous(guide = "none") +  
 labs(title = "USA - centralità", x = "X", y = "Y", size = "", colour = "Pagerank") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



Si può notare (sia da Degree weighted che da Closeness - uno è l’opposto dell’altro dato che come peso è stata presa la distanza) che la parte ovest è più lontana (e rada) e la parte est è più vicina (e densa). Dalla Betweeness si può notare come la parte centrale sia in mezzo ai cammini, ciò è anche intuitivo perché per passare da una costa all’altra devo passare per gli stati centrali. Pagerank e Degree sono abbastanza omogenei (questo è osservabile anche nell’analisi dei singoli stati).

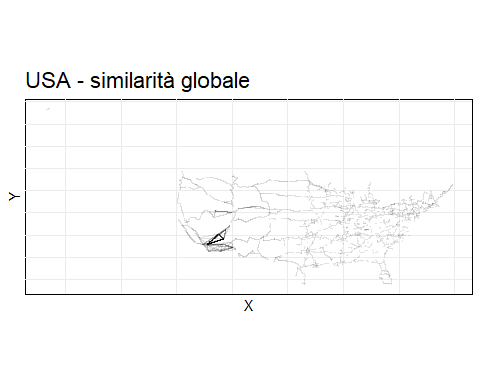
## Locale - similarità e eterogeneità

r\_s = similarity(roads)

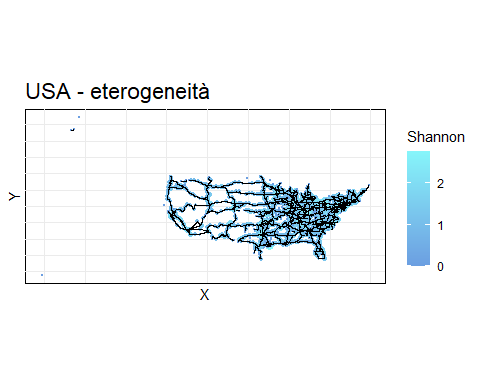
## Warning in cor(A): la deviazione standard è zero

r\_h = heterogeneity(roads)  
  
# graph\_from\_adjacency\_matrix(r\_s$cosine, mode = "undirected", weighted = TRUE) %>%  
# as\_tbl\_graph() %>%  
# mutate(x = vertex.attributes(roads)$x, y = vertex.attributes(roads)$y) %>%  
# ggraph(layout = 'manual', x = x, y = y) +  
# #geom\_node\_point()+  
# geom\_edge\_link(aes(alpha = weight/2, filter = (weight > quantile(weight, 0.999, na.rm = TRUE)))) +   
# scale\_x\_continuous(guide = "none") +  
# scale\_y\_continuous(guide = "none") +  
# coord\_fixed(ratio = 1) +  
# scale\_edge\_alpha\_continuous(guide = "none") +  
# labs(title = "USA - similarità coseno", x = "X", y = "Y", edge\_alpha = "") +  
# theme\_minimal() +  
# theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))  
#   
# graph\_from\_adjacency\_matrix(r\_s$pearson, mode = "undirected", weighted = TRUE) %>%  
# as\_tbl\_graph() %>%  
# mutate(x = vertex.attributes(roads)$x, y = vertex.attributes(roads)$y) %>%  
# ggraph(layout = 'manual', x = x, y = y) +  
# #geom\_node\_point()+  
# geom\_edge\_link(aes(alpha = weight/2, filter = (weight > quantile(weight, 0.999, na.rm = TRUE)))) +   
# scale\_x\_continuous(guide = "none") +  
# scale\_y\_continuous(guide = "none") +  
# coord\_fixed(ratio = 1) +  
# scale\_edge\_alpha\_continuous(guide = "none") +  
# labs(title = "USA - similarità pearson", x = "X", y = "Y", edge\_alpha = "") +  
# theme\_minimal() +  
# theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))  
  
graph\_from\_adjacency\_matrix(r\_s$global, mode = "undirected", weighted = TRUE) %>%  
 as\_tbl\_graph() %>%  
 mutate(x = vertex.attributes(roads)$x, y = vertex.attributes(roads)$y) %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 #geom\_node\_point()+  
 geom\_edge\_link(aes(alpha = weight/2, filter = (weight > quantile(weight, 0.999, na.rm = TRUE)))) +   
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_edge\_alpha\_continuous(guide = "none") +  
 labs(title = "USA - similarità globale", x = "X", y = "Y", edge\_alpha = "") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))

## Warning: The `adjmatrix` argument of `graph\_from\_adjacency\_matrix()` must be symmetric  
## with mode = "undirected" as of igraph 1.6.0.  
## ℹ Use mode = "max" to achieve the original behavior.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

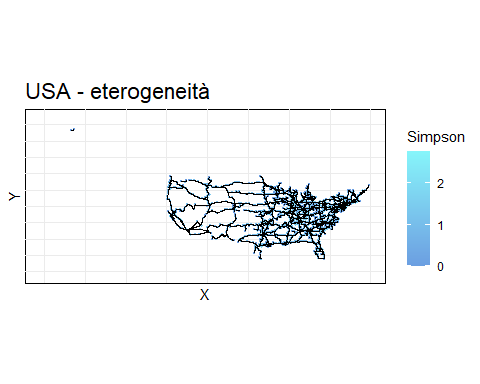


roads %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(size = I(r\_h$shannon), colour = r\_h$shannon))+  
 geom\_edge\_link() +   
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_gradient(low="#6B9EE1", high="#86F5FA") +  
 scale\_size\_continuous(guide = "none") +  
 labs(title = "USA - eterogeneità", x = "X", y = "Y", size = "", colour = "Shannon") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



roads %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(size = I(r\_h$simpson), colour = r\_h$shannon))+  
 geom\_edge\_link() +   
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_gradient(low="#6B9EE1", high="#86F5FA") +  
 scale\_size\_continuous(guide = "none") +  
 labs(title = "USA - eterogeneità", x = "X", y = "Y", size = "", colour = "Simpson") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))

## Warning: Removed 103 rows containing missing values or values outside the scale range  
## (`geom\_point()`).



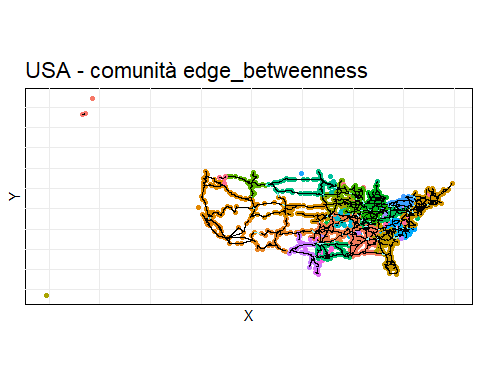
Per quanto riguarda la similarità globale, si nota che i nodi sono per lo più simili ai loro vicini, si nota che nella parte sud-ovest vi è una similarità anche tra nodi non vicini (per creare il grafico è stato preso il 99,9 percentile).

Per l’eterogeneità (misura opposta alla similarità), si vede come la parte centro-est sia un po’ più eterogenea.

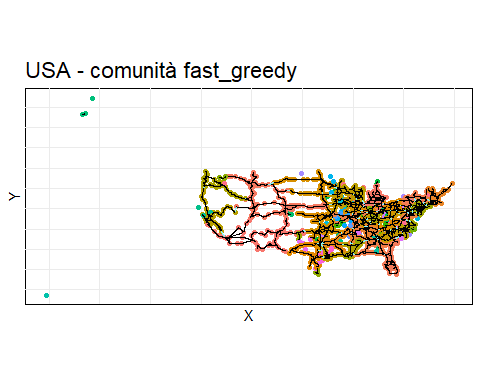
## Gruppi - comunità e clustering

#creazione grafo fully connected per clustering  
edges = expand.grid(county1 = as.character(roads\_nodes$county), county2 = as.character(roads\_nodes$county), stringsAsFactors = FALSE)  
edges = edges[edges$county1 < edges$county2, ]  
edges = edges %>%  
 left\_join(roads\_nodes, by = c("county1" = "county")) %>%  
 rename(y\_u = lat, x\_u = lon) %>%  
 left\_join(roads\_nodes, by = c("county2" = "county")) %>%  
 rename(y\_v = lat, x\_v = lon) %>%  
 mutate(dist = sqrt((x\_v - x\_u)^2 + (y\_v - y\_u)^2)) %>%  
 select(county1, county2, dist)  
  
roads\_FC = create\_state\_graph(roads\_nodes, edges)  
  
roads\_edges\_FC = edges  
rm(edges)  
  
r\_co = communities(roads)  
r\_cl = clustering(roads\_FC)  
  
for(i in 1:nrow(r\_co)){  
 print(paste("Comunità", r\_co$method[[i]], ": modularità", modularity(r\_co$result[[i]]), "numero comunità", length(r\_co$result[[i]])))  
 print(  
 roads %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(colour = factor(membership(r\_co$result[[i]])))) +  
 geom\_edge\_link(aes(colour = crossing(r\_co$result[[i]], roads))) +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_discrete(guide = "none") +  
 scale\_edge\_color\_manual(values = c("TRUE" = "#86F5FA", "FALSE" = "#000000"), guide = "none") +  
 labs(title = paste("USA - comunità", r\_co$method[[i]]), x = "X", y = "Y", color = "", edge\_color = "") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))  
 )  
}

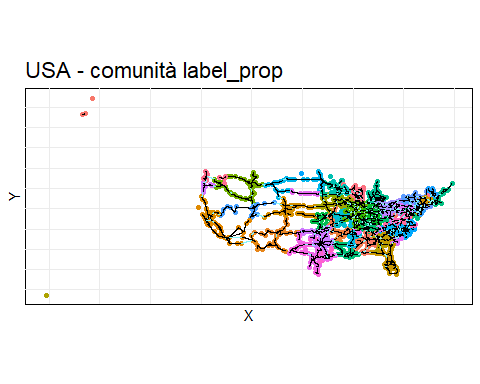
## [1] "Comunità edge\_betweenness : modularità 0.920843633522139 numero comunità 158"



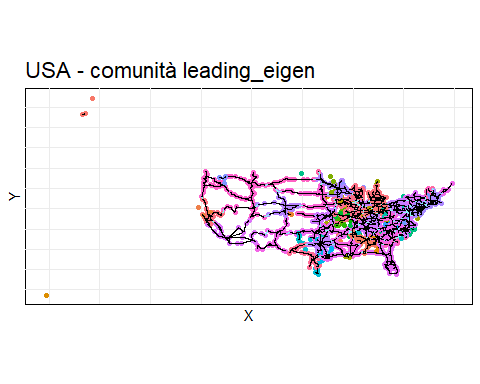
## [1] "Comunità fast\_greedy : modularità 0.922059133734641 numero comunità 161"



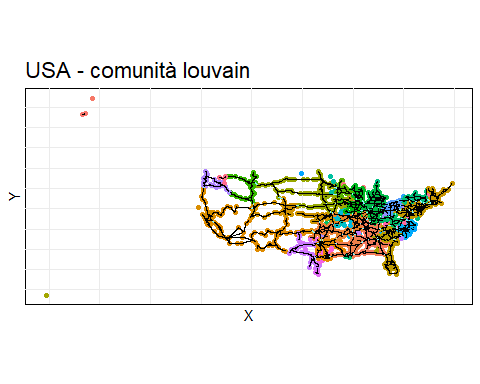
## [1] "Comunità label\_prop : modularità 0.848485911212111 numero comunità 282"



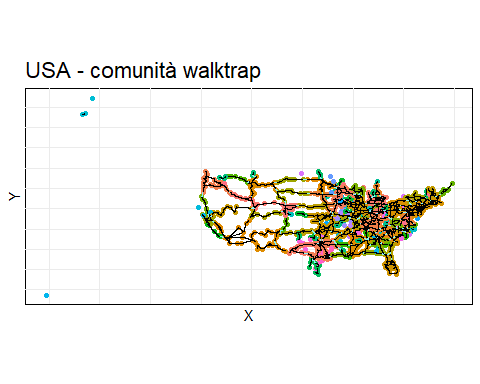
## [1] "Comunità leading\_eigen : modularità 0.899372221027243 numero comunità 179"



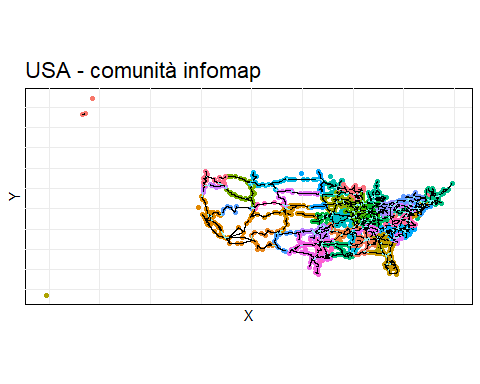
## [1] "Comunità louvain : modularità 0.923610916277779 numero comunità 159"



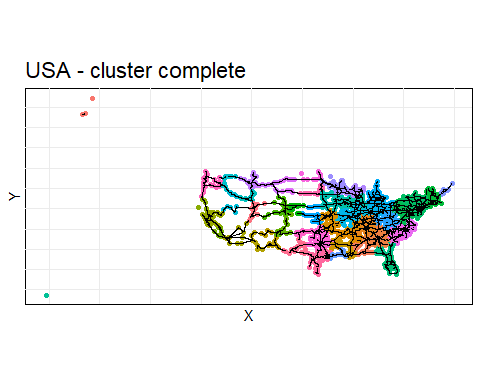
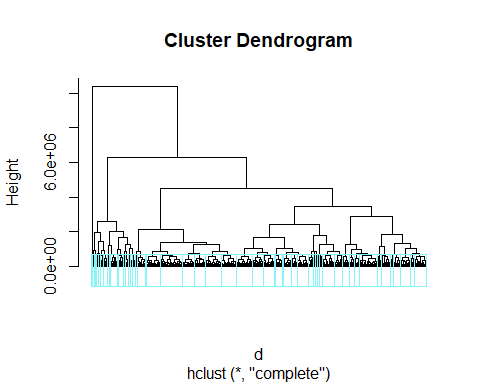
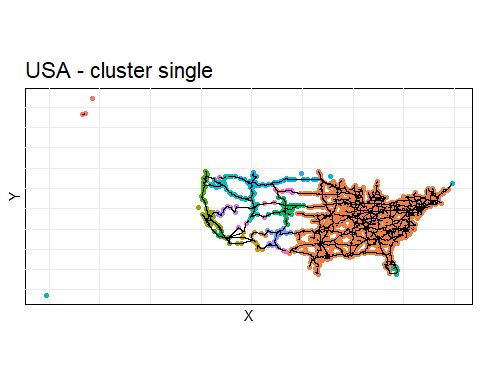
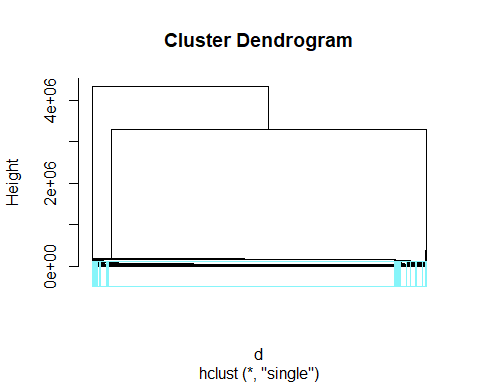
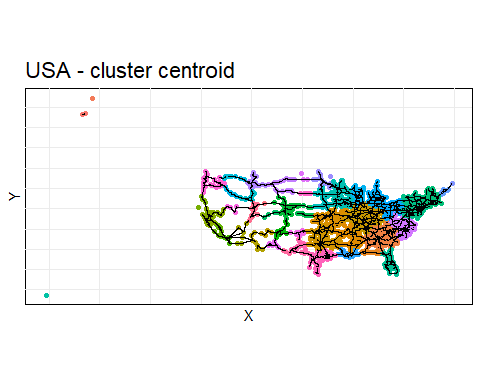
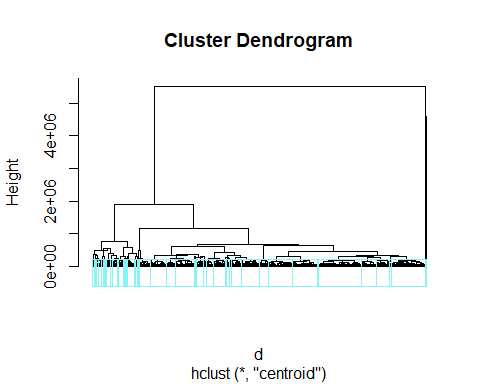
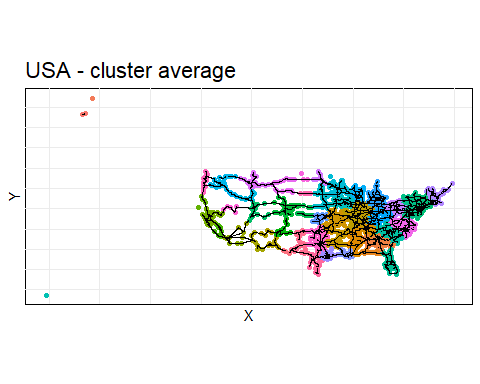
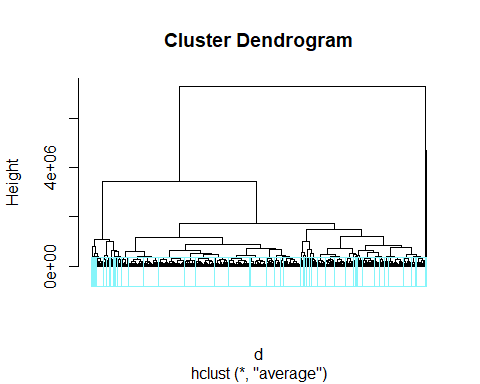
## [1] "Comunità walktrap : modularità 0.902676594334143 numero comunità 205"



## [1] "Comunità infomap : modularità 0.830864145575309 numero comunità 331"



for(i in 1:nrow(r\_cl)){  
 r\_cl$result[[i]]$labels = rep("", length(r\_cl$result[[1]]$labels))  
 plot(r\_cl$result[[i]], hang=-1)  
 rect.hclust(r\_cl$result[[i]], k = 51, border="#86F5FA")  
 print(  
 roads %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(colour = factor(cutree(r\_cl$result[[i]], k = 51)))) +  
 geom\_edge\_link() +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_discrete(guide = "none") +  
 labs(title = paste("USA - cluster", r\_cl$method[[i]]), x = "X", y = "Y", color = "") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))  
 )  
}



Per le comunità, si nota che con tutti i metodi il numero di comunità (160-330) è superiore al numero di stati (51), si può anche notare che in tutti i metodi in generale ci siano più comunità a est ripetto a ovest. Dato che i metodi label\_prop e infomap hanno modulatità minore degli altri metodi, è più probabile che il grafo abbia ~300 comunità, ciò non rappresenta né gli stati (51) né le contee (nodi - 1765).

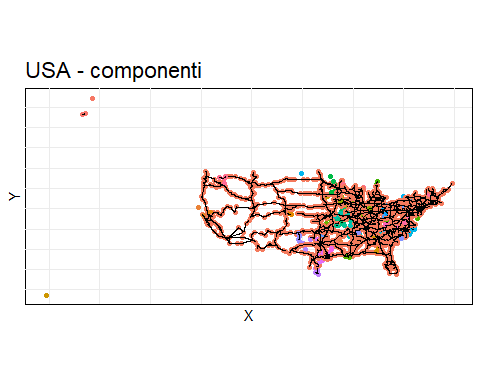
Per il clustering è stato creato un grafo fully connected, in quanto (prossimo paragrafo) il grafo ha 128 componenti, quindi un’analisi sul clustering per dividere il grafo in 51 cluster sarebbe triviale essendoci delle componenti non connesse che hanno distanza infinita. Si nota come solo in alcuni stati (nei metodi complete, average e centroid) i nodi appartenenti a tali stati sono nello stesso cluster (es. Florida e Texas, California), mentre gli stati soprattutto del centro-est i nodi vengono agglomerati in cluster senza seguire gli stati di appartenenza.

## Globale - connettività e resilienza

r\_con = connettivity(roads)  
r\_r = resilience(roads)  
  
print(paste("componenti: numero componenti", r\_con$components$no, "la più grande raggiunge il", round(max(r\_con$components$csize)/vcount(roads)\*100, 2), "% dei nodi"))

## [1] "componenti: numero componenti 128 la più grande raggiunge il 90.59 % dei nodi"

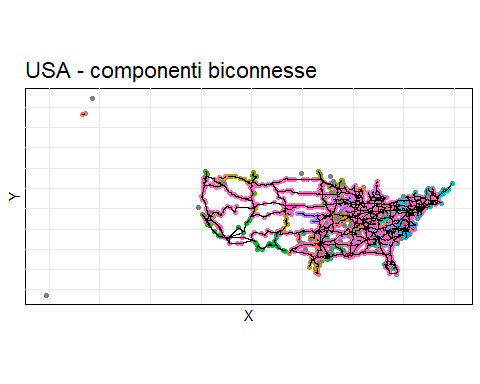
roads %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(colour = factor(r\_con$components$membership))) +  
 geom\_edge\_link() +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_discrete(guide = "none") +  
 labs(title = "USA - componenti", x = "X", y = "Y", color = "") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



print(paste("componenti: numero componenti biconnesse", r\_con$biconnected\_components$no, "la più grande raggiunge il", round(max(sapply(r\_con$biconnected\_components, length))/vcount(roads)\*100, 2), "% dei nodi"))

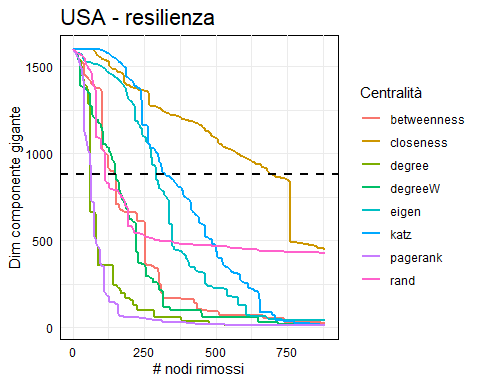
## [1] "componenti: numero componenti biconnesse 399 la più grande raggiunge il 22.61 % dei nodi"

node\_bicomponent = rep(NA, vcount(roads))  
for(i in seq\_along(r\_con$biconnected\_components$components)){  
 node\_bicomponent[r\_con$biconnected\_components$components[[i]]] = i  
}  
  
roads %>%  
 as\_tbl\_graph() %>%  
 activate(nodes) %>%  
 mutate(biconnected\_component = as.factor(node\_bicomponent)) %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(colour = biconnected\_component)) +  
 geom\_edge\_link() +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_discrete(guide = "none") +  
 labs(title = "USA - componenti biconnesse", x = "X", y = "Y", color = "") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



# plot\_hierarchy(r\_con$cohesive\_blocks)  
# blocks(r\_con$cohesive\_blocks)  
# graphs\_from\_cohesive\_blocks(r\_con$cohesive\_blocks, roads)  
# cohesion(r\_con$cohesive\_blocks)  
  
r\_r %>%  
 as\_tibble() %>%  
 mutate(removed\_nodes = 0:floor(vcount(roads)/2)) %>%  
 pivot\_longer(cols = -removed\_nodes, names\_to = "metric", values\_to = "value") %>%  
 ggplot(aes(x = removed\_nodes, y = value, color = metric)) +  
 geom\_line(size = 1) +  
 geom\_hline(yintercept = floor(vcount(roads)/2), linetype = "dashed", color = "black", size = 0.8) +  
 labs(title = "USA - resilienza", x = "# nodi rimossi", y = "Dim componente gigante", color = "Centralità") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



Dalla connettività, si può notare che le componenti sono 128 e la più grande raggiunge il 90.59 % dei nodi, mentre le componenti biconnesse sono 399 e la più grande raggiunge il 22.61 % dei nodi.

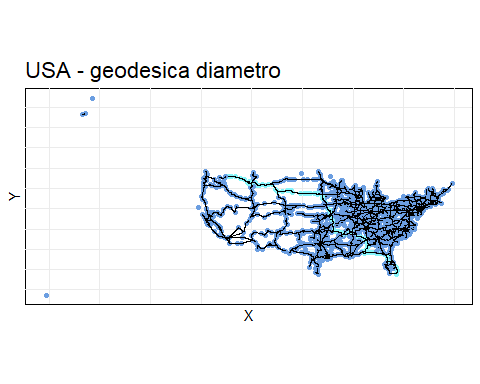
Dalla resilienza, si nota che pagerank e degree sono i più performanti già dopo ~50 nodi rimossi disconnettono la rete oltre al 50%.

## Globale - geodesica

r\_g = geodesic(roads)  
  
print(paste("geodesica: media", r\_g$geodesic\_mean, "diametro", r\_g$geodesic\_diameter))

## [1] "geodesica: media 41.4227357328353 diametro 114"

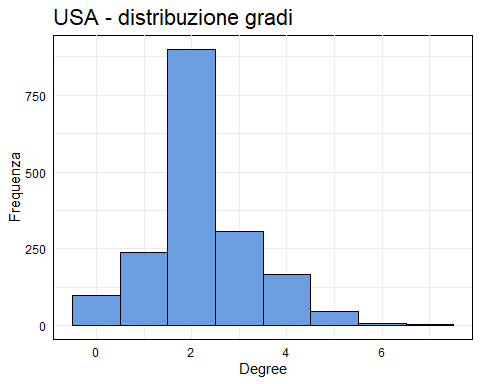
roads %>%  
 ggraph(layout = 'manual', x = x, y = y) +  
 geom\_node\_point(aes(colour = vertex.attributes(roads)$name %in% r\_g$geodesic\_diameter\_nodes$name)) +  
 geom\_edge\_link() +  
 scale\_x\_continuous(guide = "none") +  
 scale\_y\_continuous(guide = "none") +  
 coord\_fixed(ratio = 1) +  
 scale\_color\_manual(values = c("TRUE" = "#86F5FA", "FALSE" = "#6B9EE1"), guide = "none") +  
 labs(title = "USA - geodesica diametro", x = "X", y = "Y", color = "") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



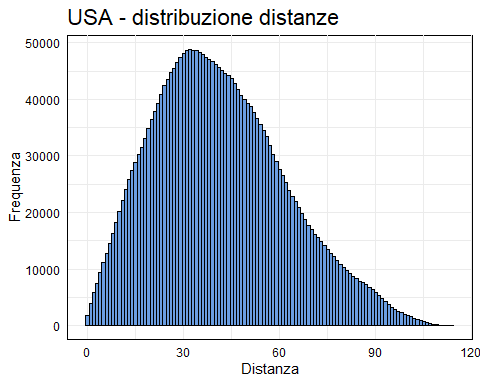
La media della geodesica è 41, è più alta rispetto alle reti piccolo mondo (6).

## Globale - power law

r\_p = power\_law(roads)  
  
data.frame(degreeD = r\_p$degreeD) %>%  
 ggplot(aes(x = degreeD)) +  
 geom\_histogram(binwidth = 1, fill = "#6B9EE1", color = "black") +  
 labs(title = "USA - distribuzione gradi", x = "Degree", y = "Frequenza") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



data.frame(distD = r\_p$distD) %>%  
 ggplot(aes(x = distD)) +  
 geom\_histogram(binwidth = 1, fill = "#6B9EE1", color = "black") +  
 labs(title = "USA - distribuzione distanze", x = "Distanza", y = "Frequenza") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



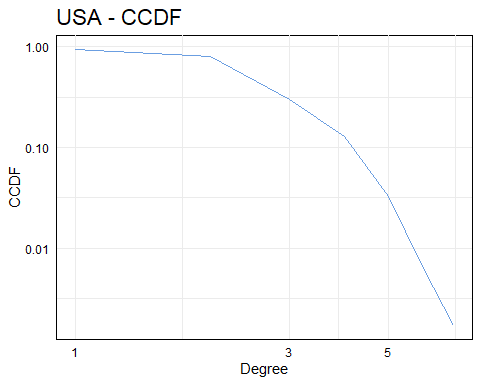
r\_p$summary

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 2.000 2.000 2.223 3.000 7.000

r\_p$skewness

## [1] 0.5654243

data.frame(x = 1:max(r\_c$degree), y = r\_p$ccdf) %>%  
 ggplot(aes(x = 1:max(r\_c$degree), y = r\_p$ccdf)) +  
 geom\_line(color = "#6B9EE1") +  
 scale\_x\_log10() +  
 scale\_y\_log10() +  
 labs(title = "USA - CCDF", x = "Degree", y = "CCDF") +  
 theme\_minimal() +  
 theme(axis.text.x=element\_text(color="#000000"), axis.text.y=element\_text(color="#000000"), axis.line=element\_line(color="#000000"), panel.background=element\_rect(fill="#ffffff"), panel.grid.major.x=element\_line(colour="#ebebeb"), panel.grid.major.y=element\_line(colour="#ebebeb"), plot.title=element\_text(size=rel(1.5)))



Si nota sia per il grado che per le distanze che la distribuzione è poisson (non power law); anche dall’output del summary e skewness si nota un’assimetria nella distribuzione.

## Globale - assortatività

r\_a = assortativita(roads, as.numeric(factor(vertex.attributes(roads)$state)), degree(roads))  
  
print(paste("assortatività: grado", round(r\_a$assortativityD[1],2)))

## [1] "assortatività: grado 0.14"

L’assortatività è bassa (0.14).

## Globale - motivi

r\_m = motif(roads)  
  
print(paste("motivi: transitività", round(r\_m$transitivity[1],2)))

## [1] "motivi: transitività 0.13"

Il motivo transitività è del 13%.

# Conclusioni

Per i singoli stati si è notato che non vi è una sostanziale differenza nella distribuzione dei gradi e delle altre misure di centralità.

Per il grafo completo si è notata una diversità tra gli stati ovest e centro-est, per diverse misure sia a livello locale che di gruppo. Si è notato che la distribuzione non è di tipo power law ma poisson.