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), il dataset è stato costruito utilizzando l'API 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")
```

```r
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 %>%
```

```r
  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()))
```

```r
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
  )
}
```

```r
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)))
  # }
```

```r
  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)))
  )
}
```

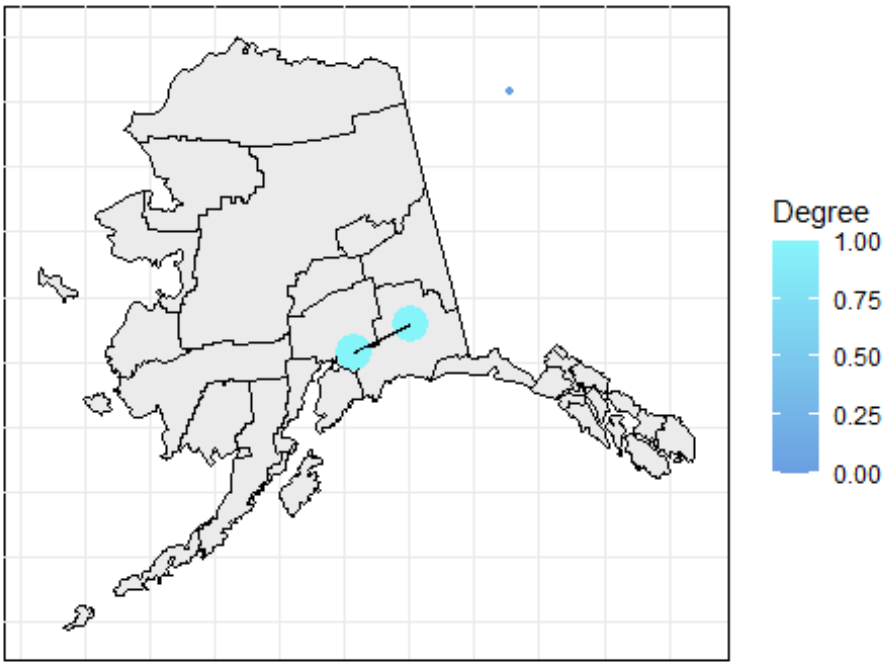
```
## Warning in min(cc[[1]], na.rm = TRUE): nessun argomento non-mancante al minimo;
## si restituisce Inf
```

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

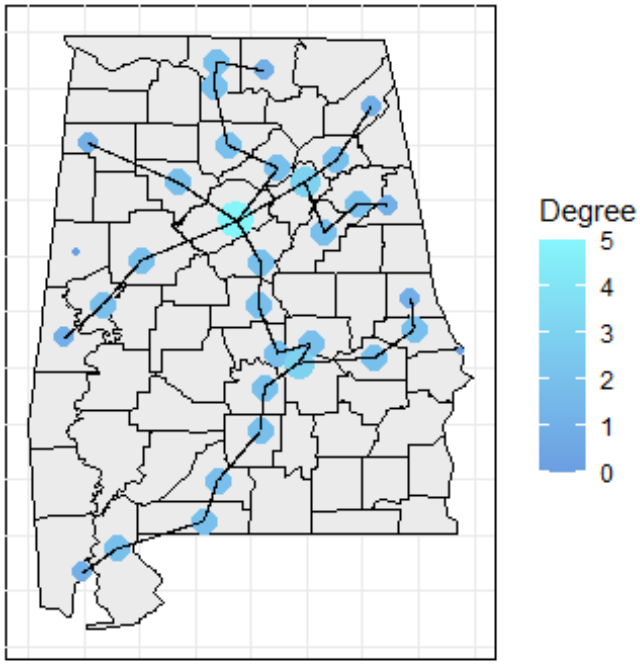
```
## Warning in max(cc[[1]], na.rm = TRUE): nessun argomento non-mancante al
## massimo; si restituisce -Inf
```

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

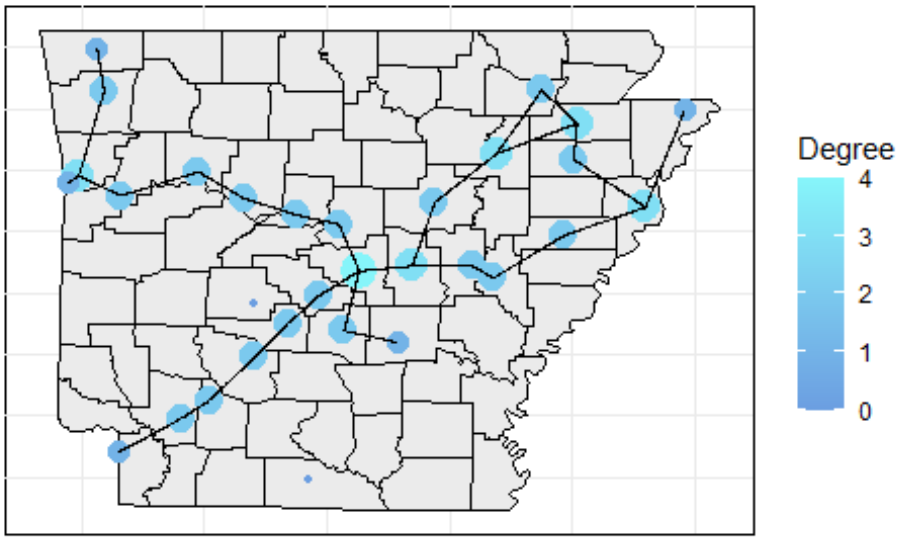
Alaska



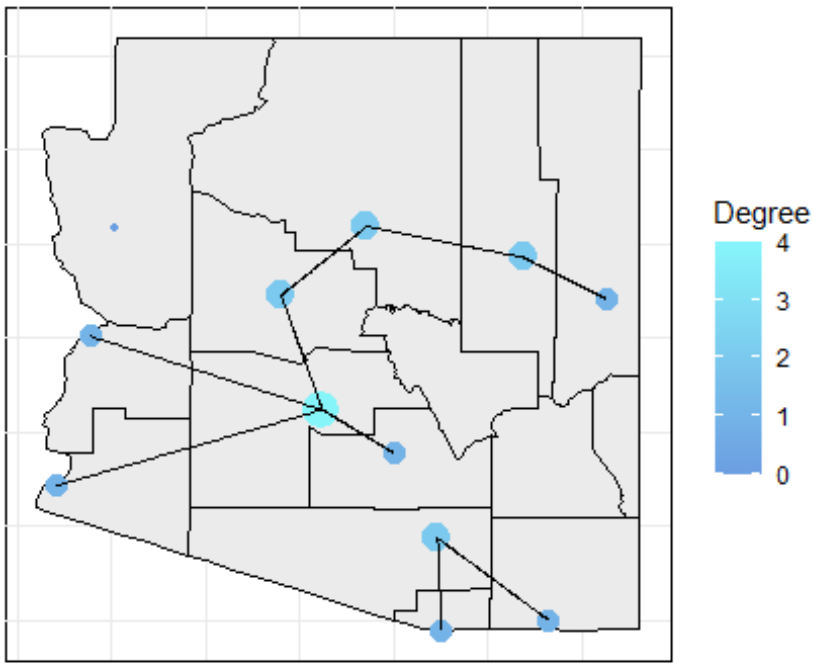
Alabama



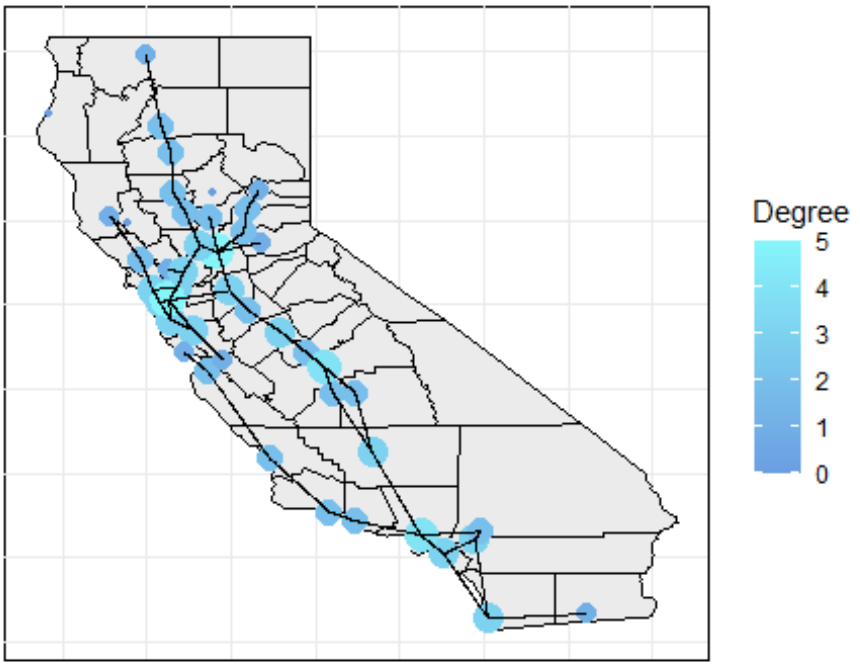
Arkansas



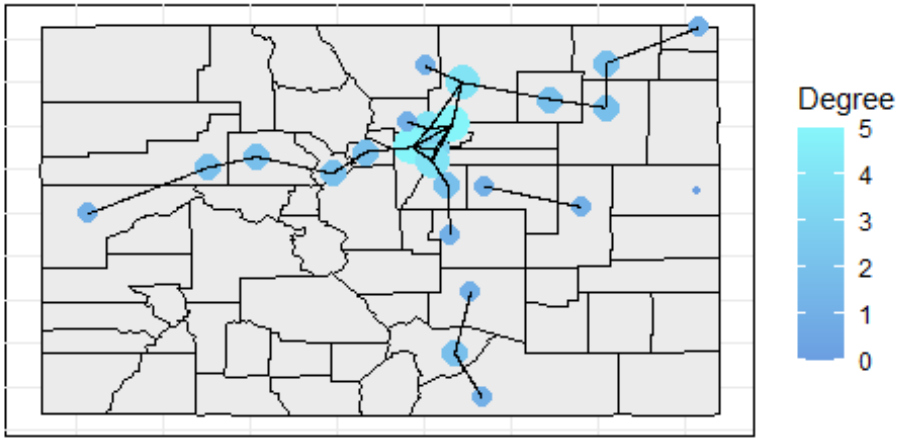
Arizona



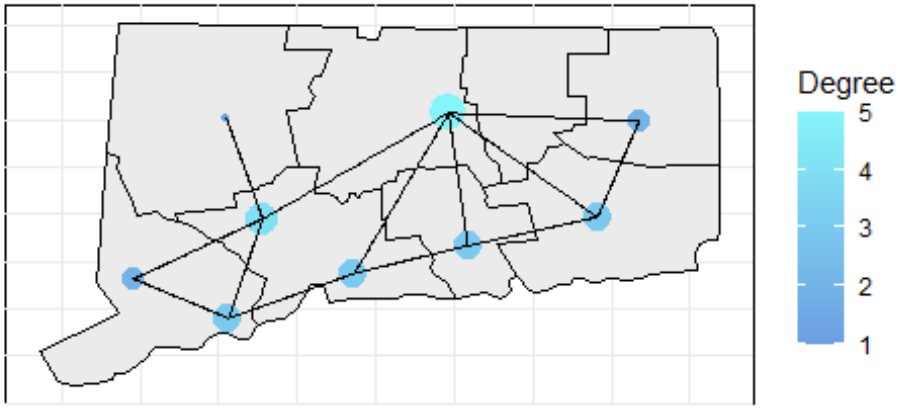
California



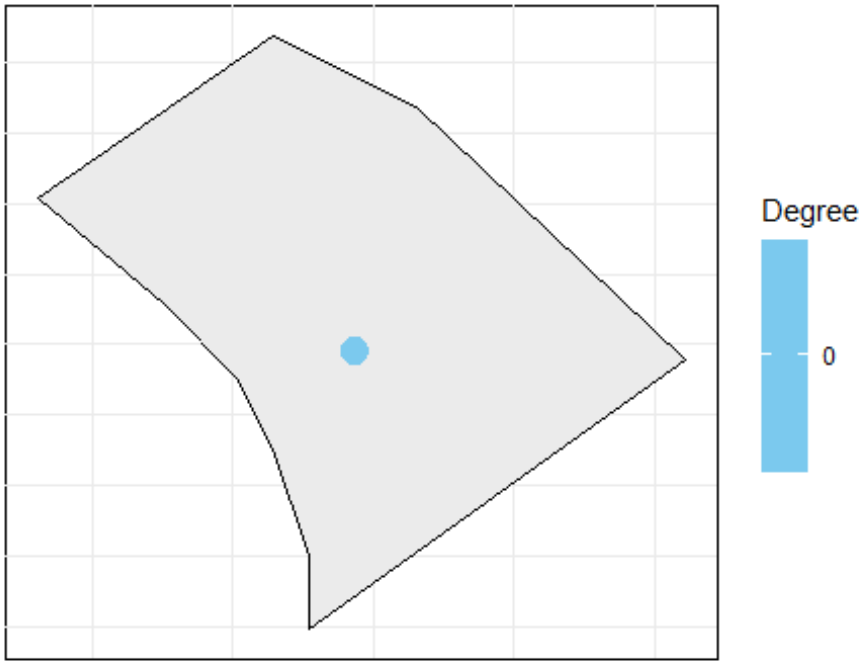
Colorado



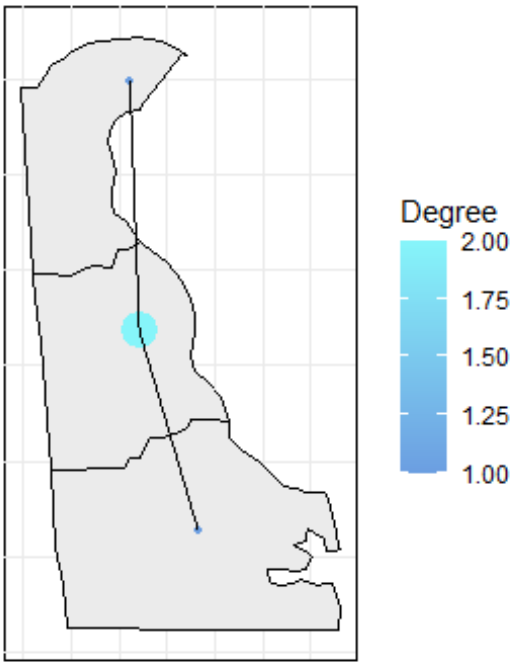
Connecticut



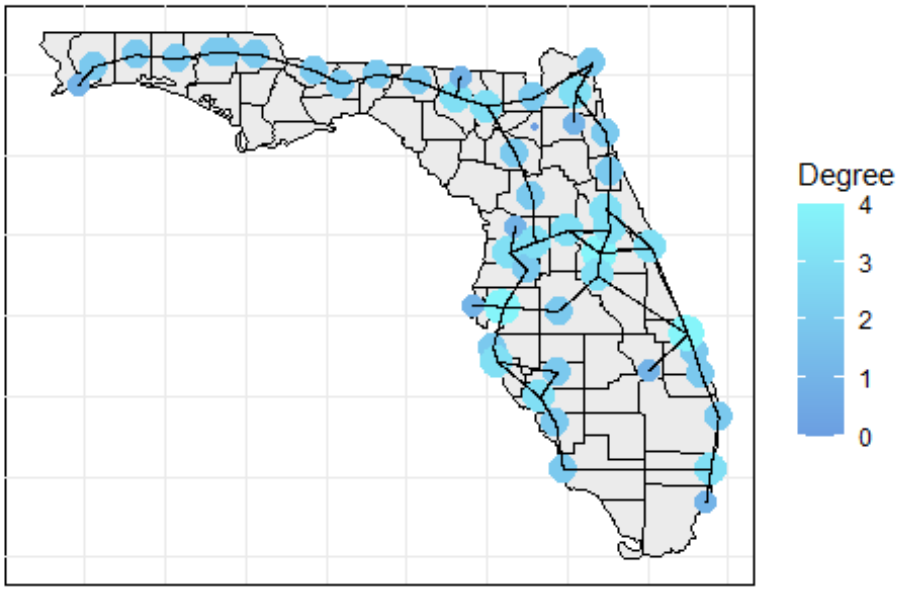
Washington District of Columbia

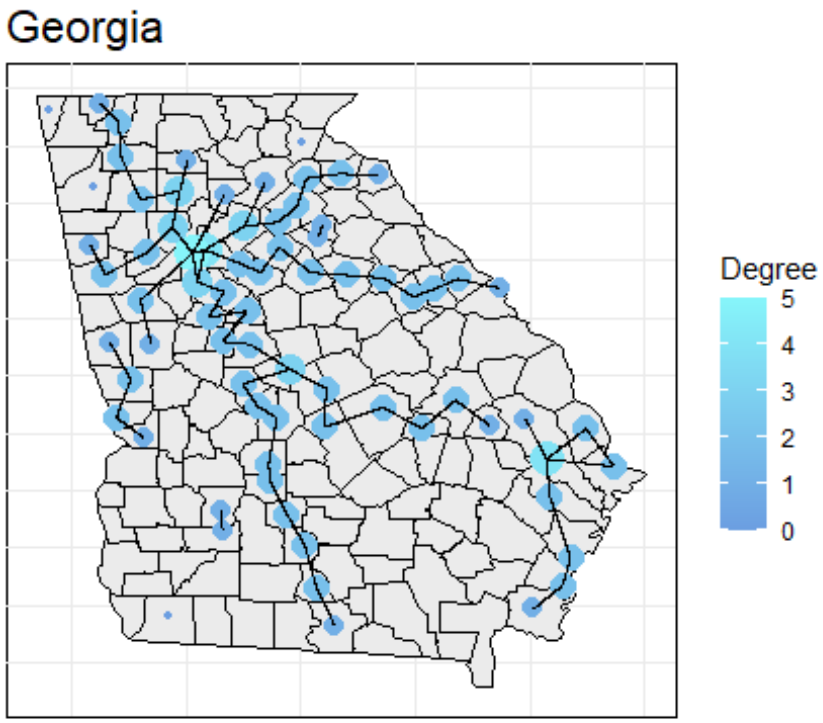


Delaware



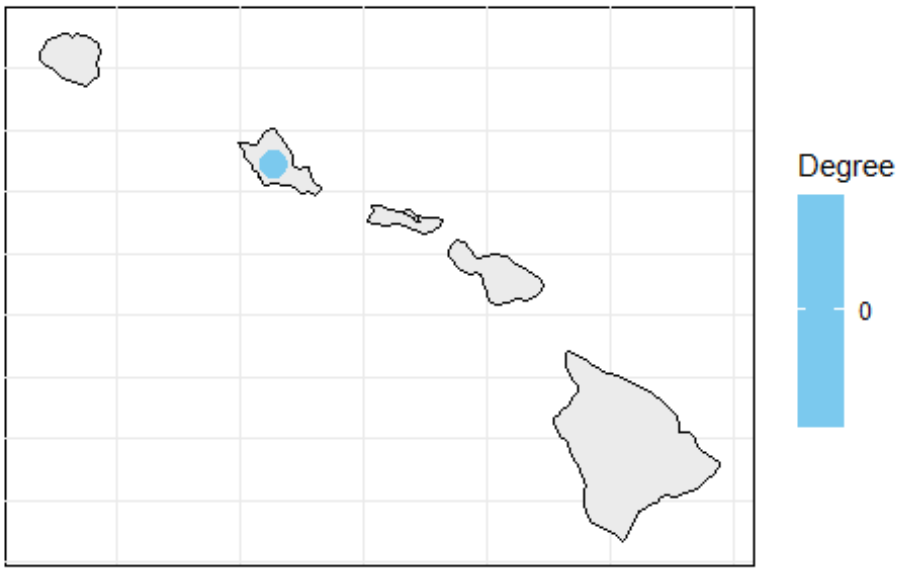
Florida



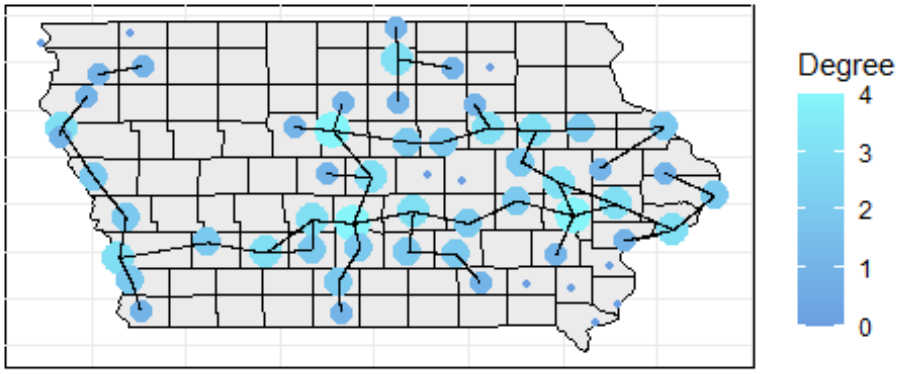



```
## Warning in min(cc[[1]], na.rm = TRUE): nessun argomento non-mancante al minimo;
## si restituisce Inf

## 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
```

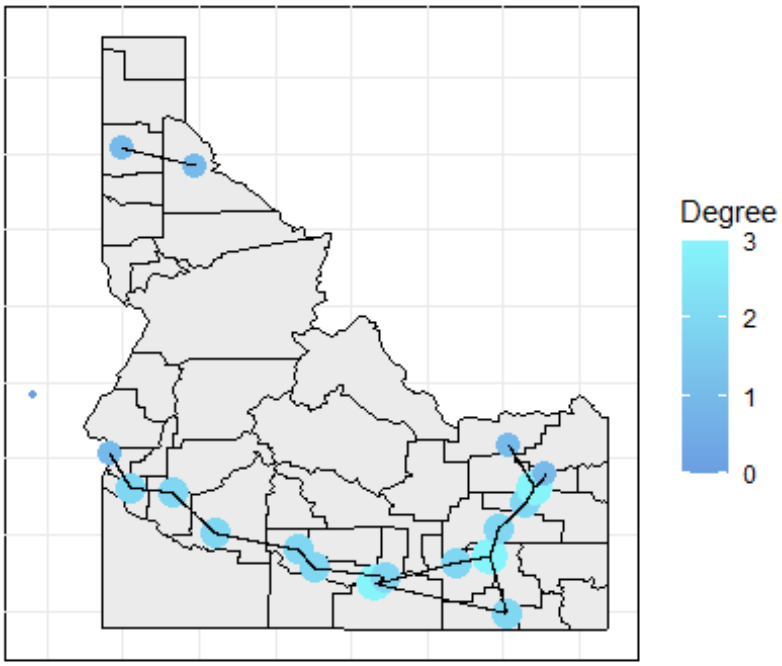
Hawaii



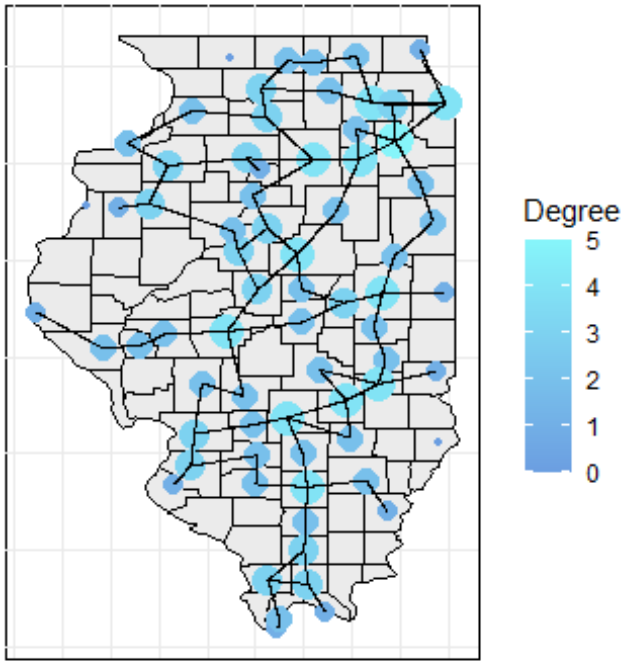
Iowa



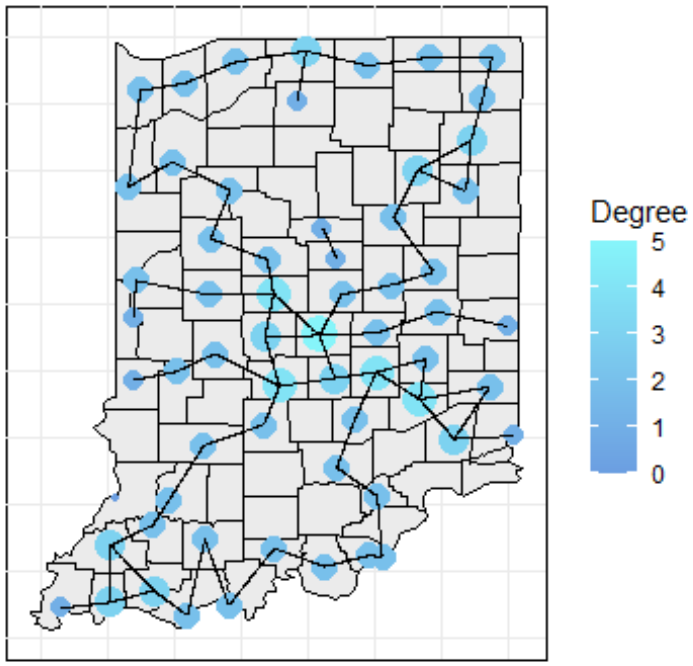
Idaho



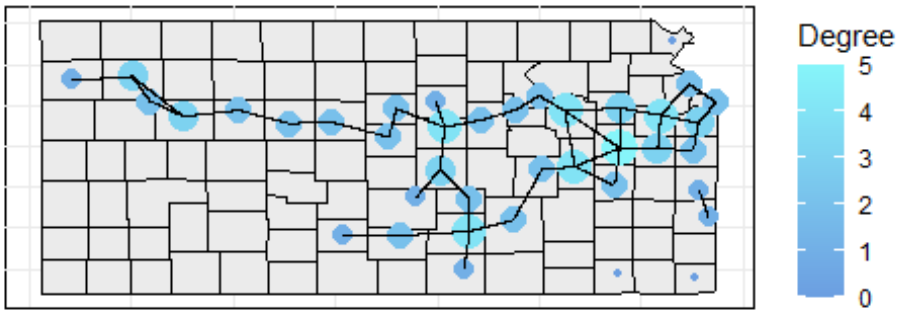
Illinois



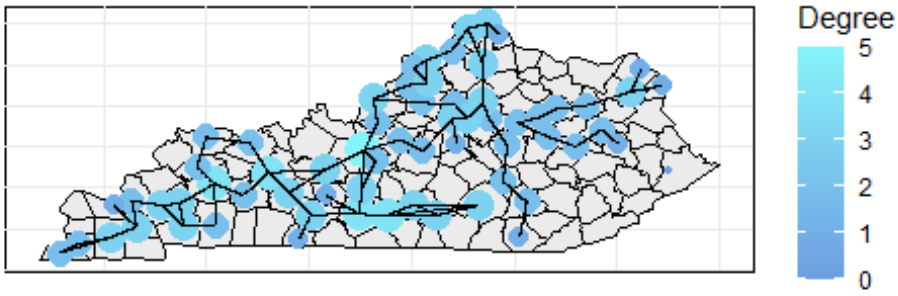
Indiana



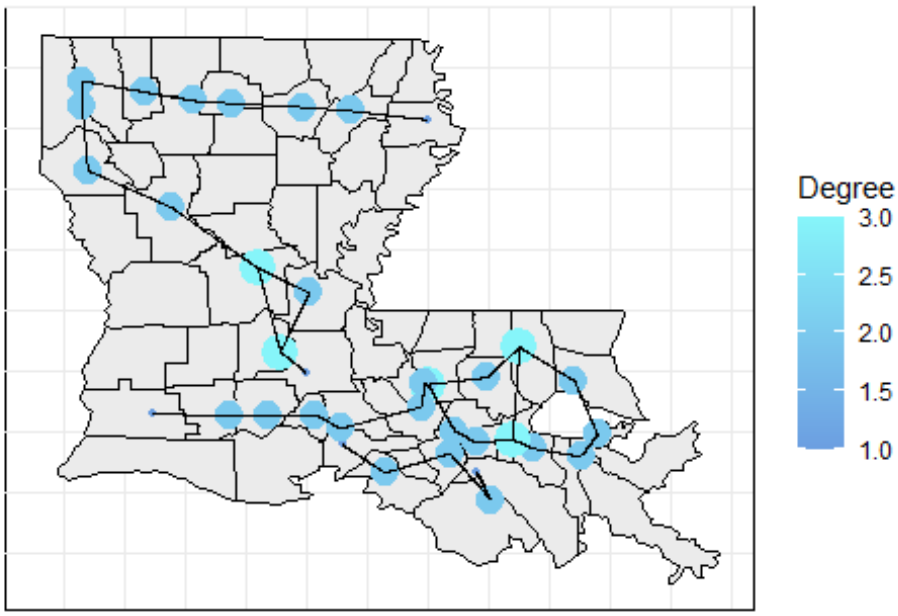
Kansas



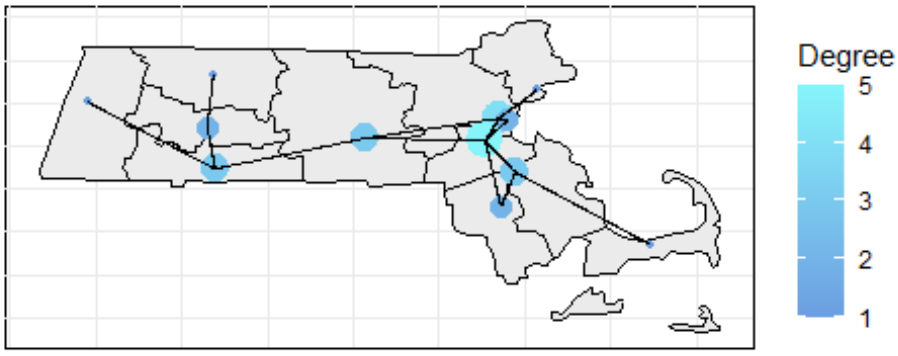
Kentucky



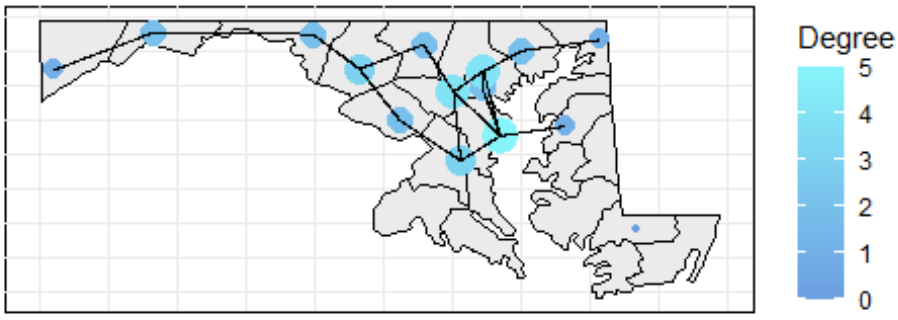
Louisiana



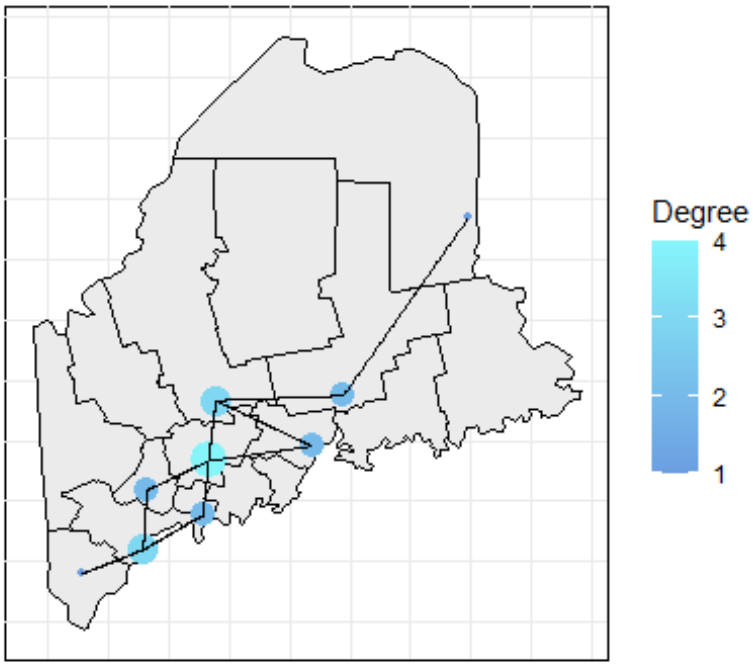
Massachusetts



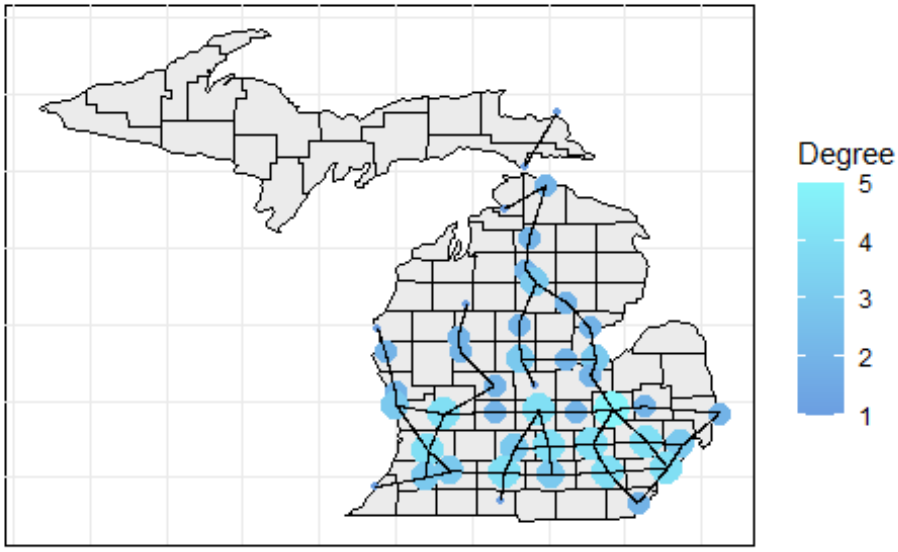
Maryland



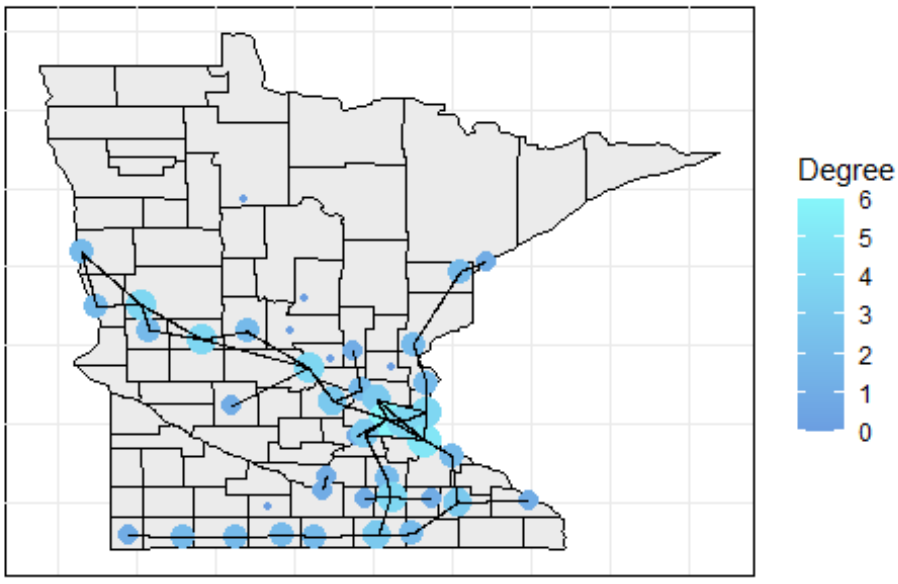
Maine



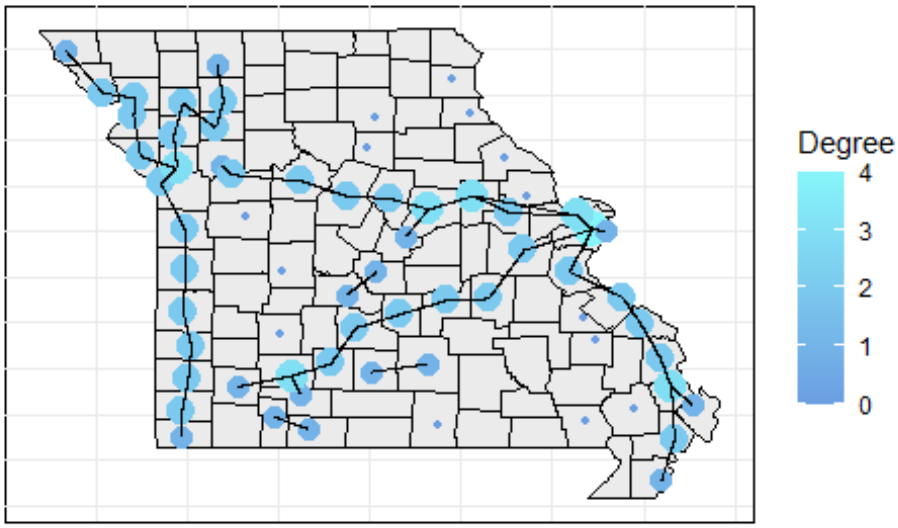
Michigan



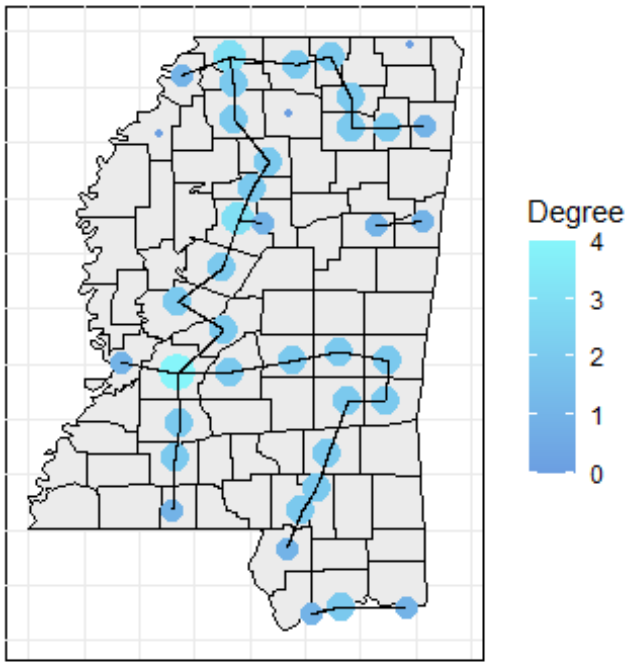
Minnesota



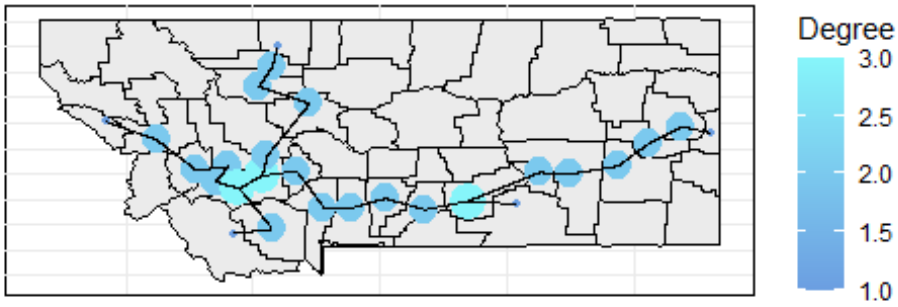
Missouri



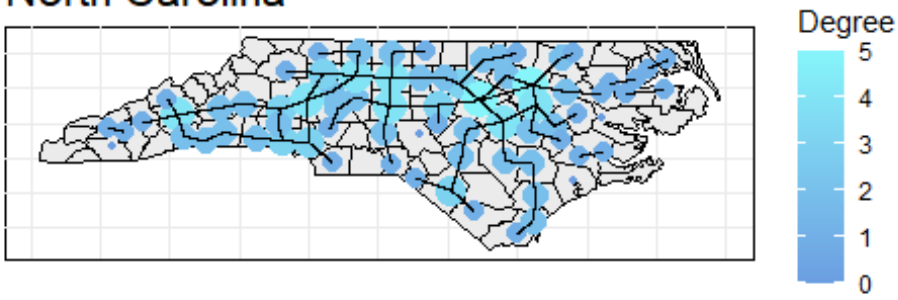
Mississippi



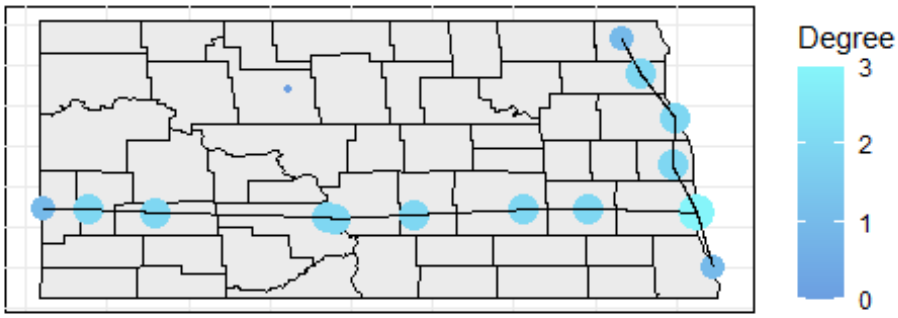
Montana



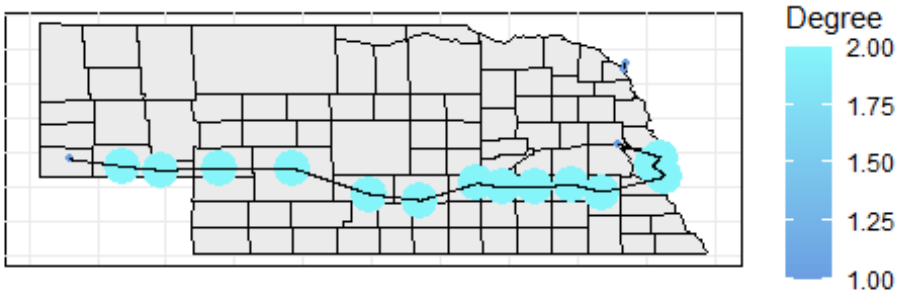
North Carolina



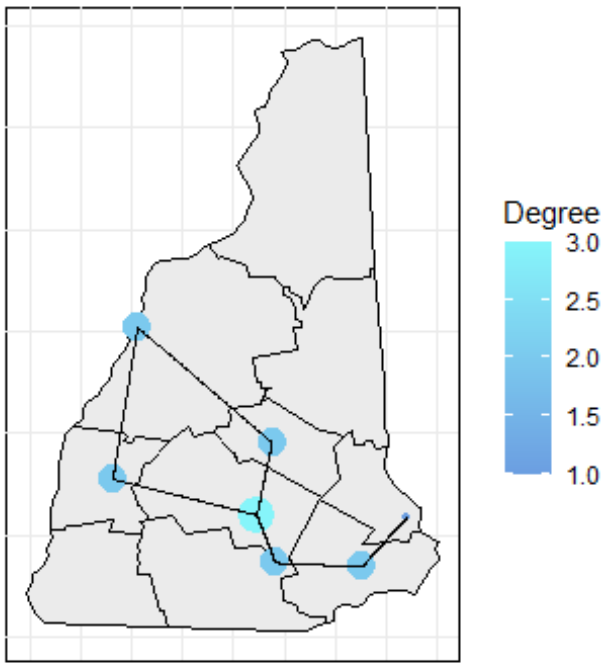
North Dakota



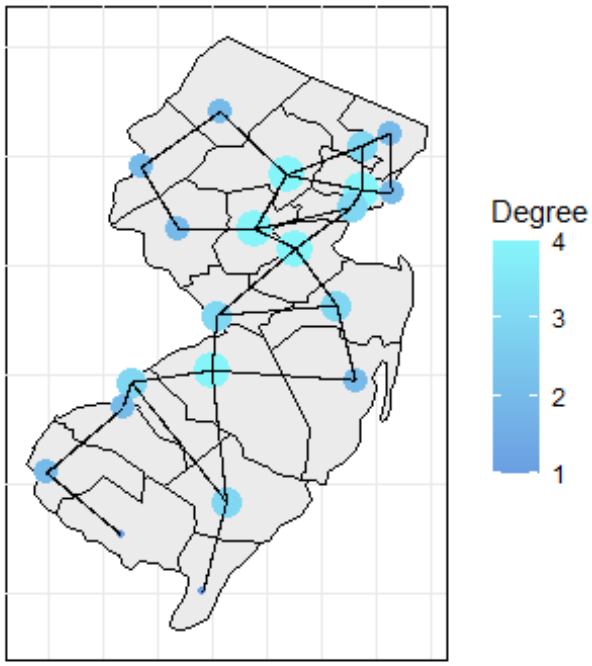
Nebraska



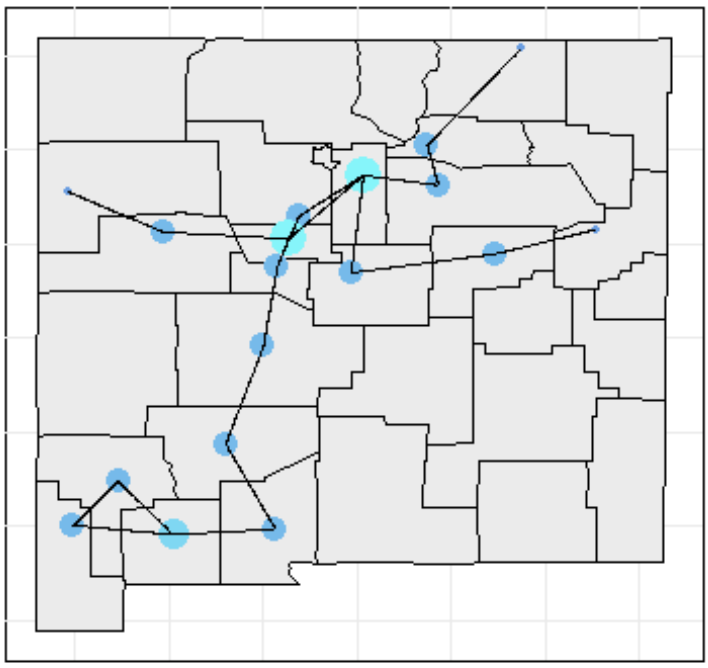
New Hampshire



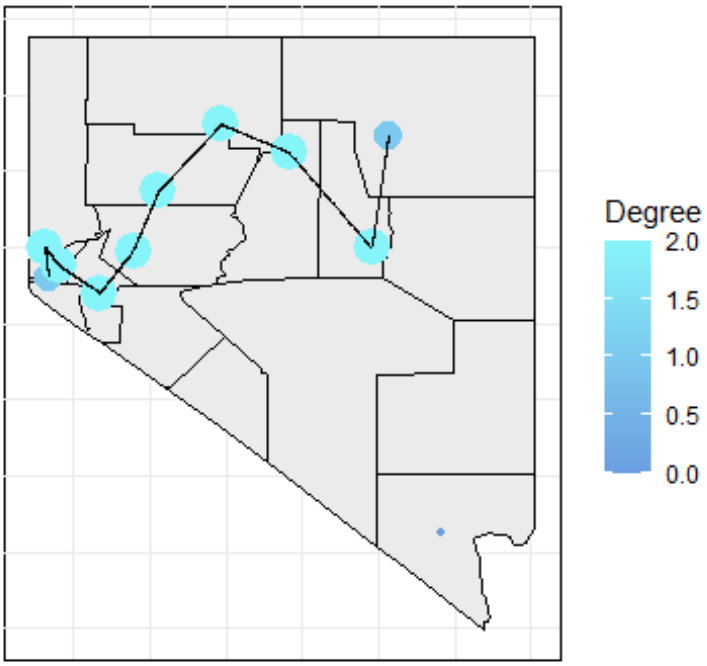
New Jersey



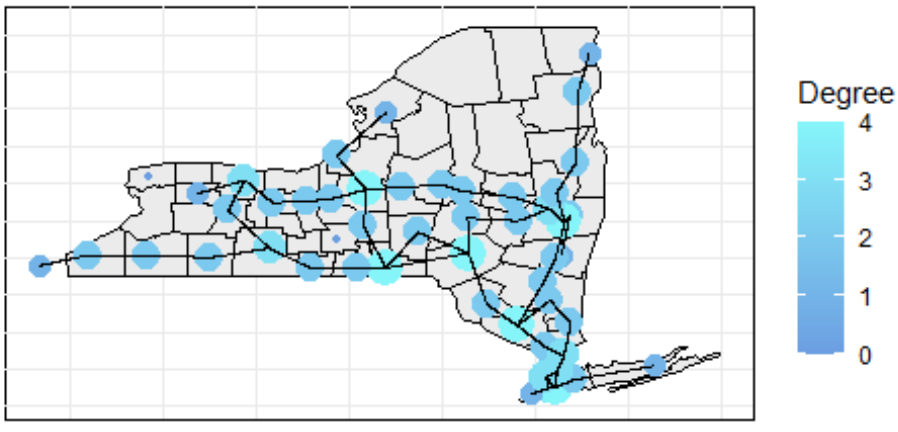
New Mexico



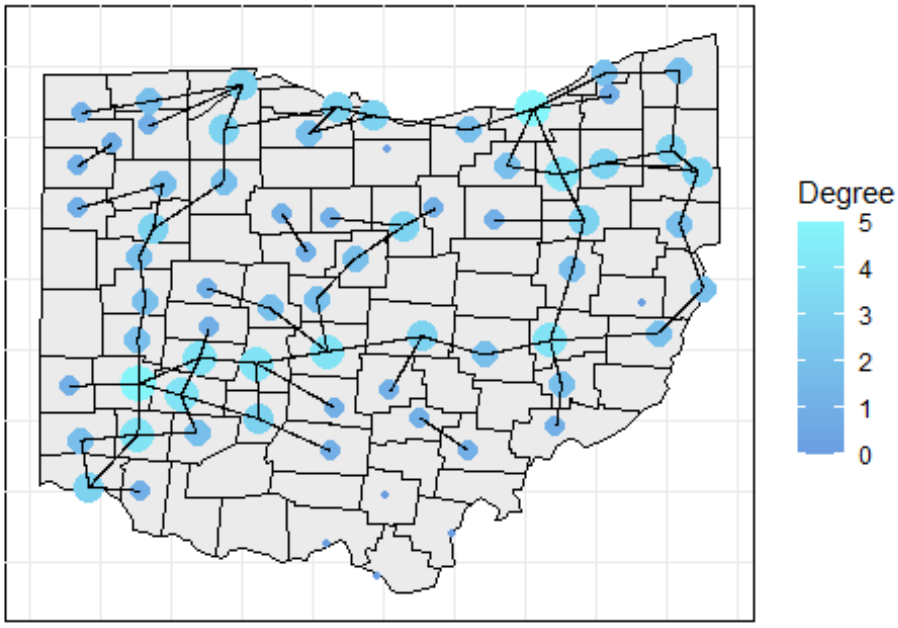
Nevada



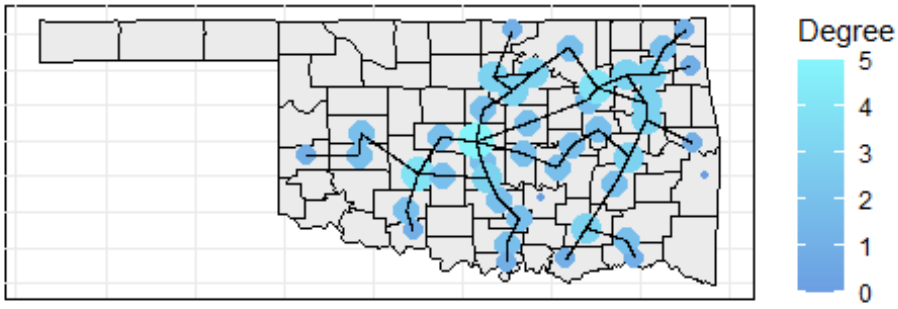
New York



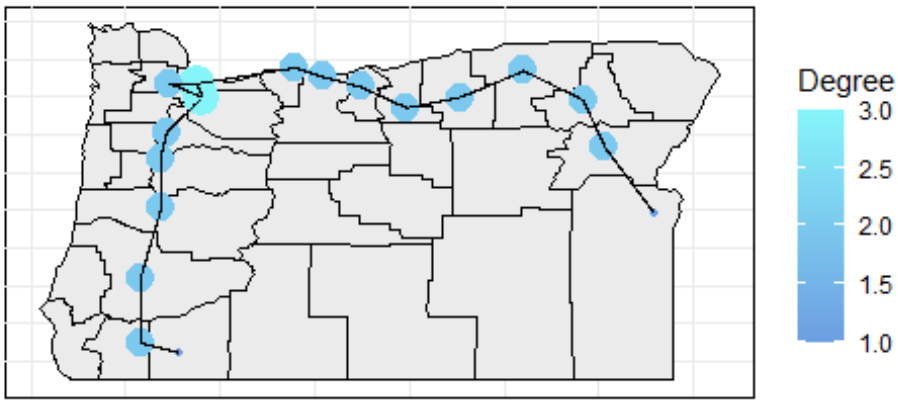
Ohio



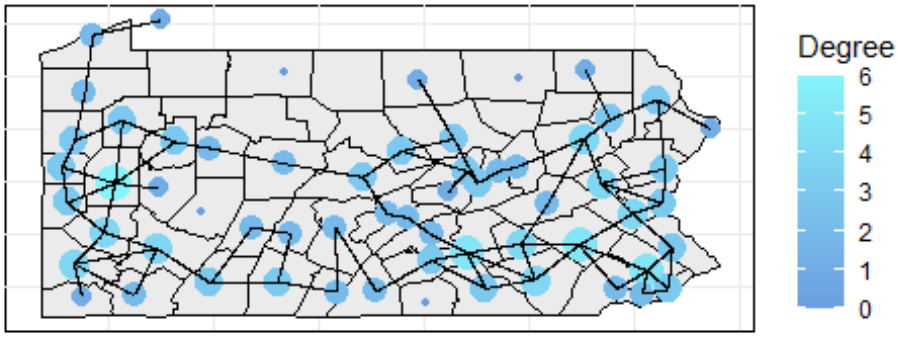
Oklahoma



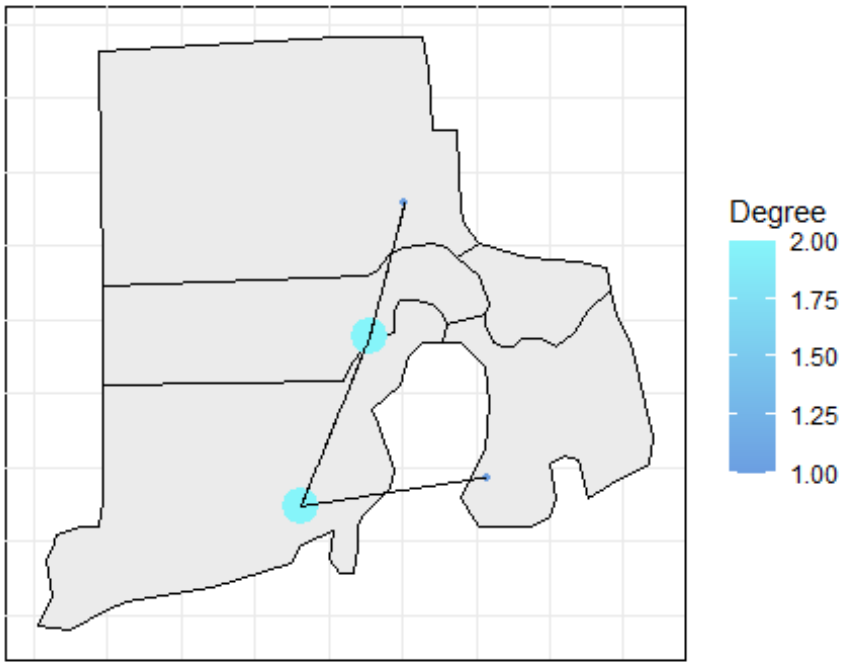
Oregon



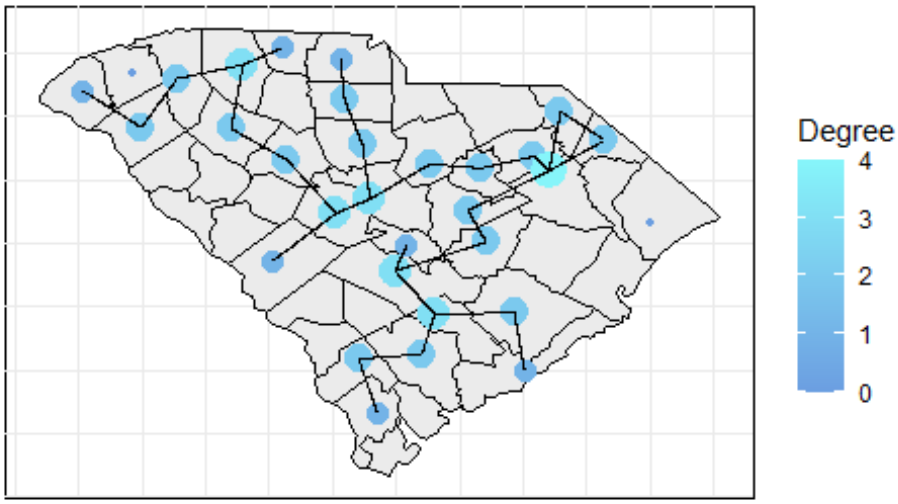
Pennsylvania



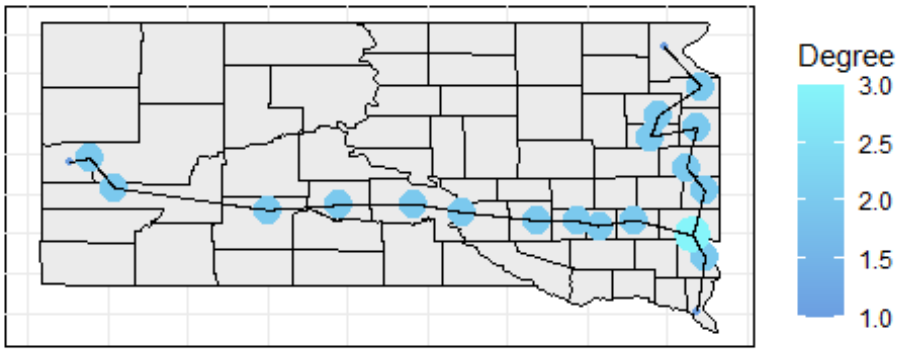
Rhode Island



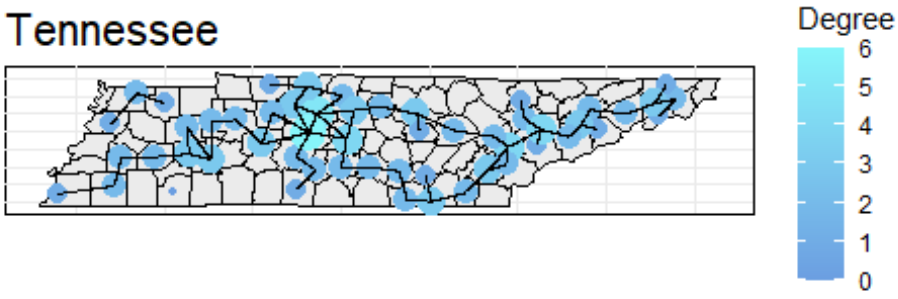
South Carolina



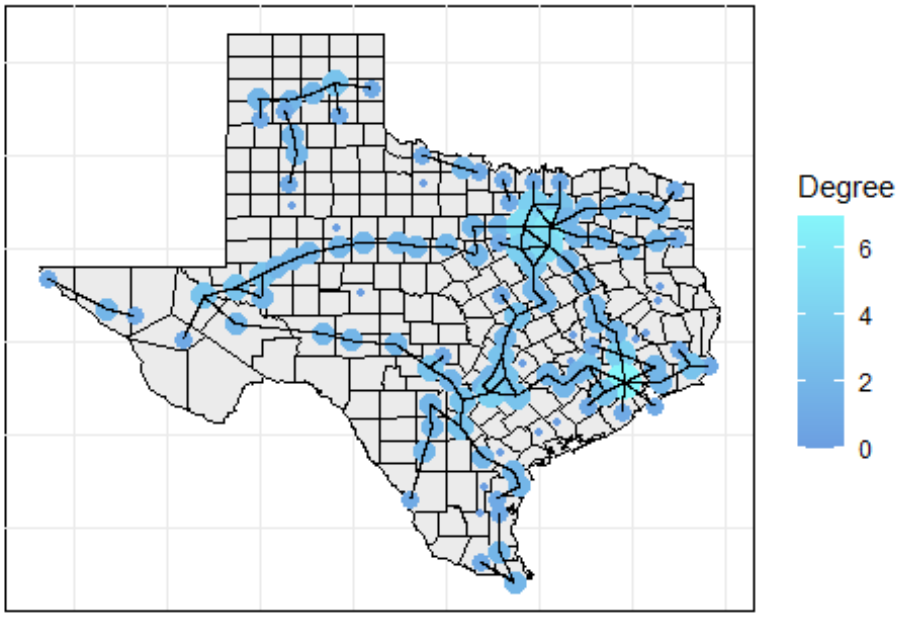
South Dakota



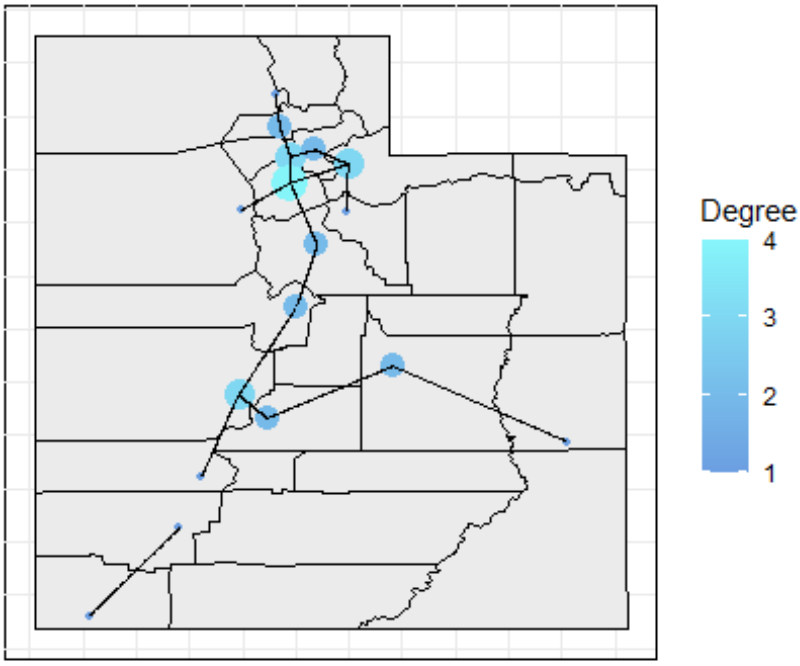
Tennessee



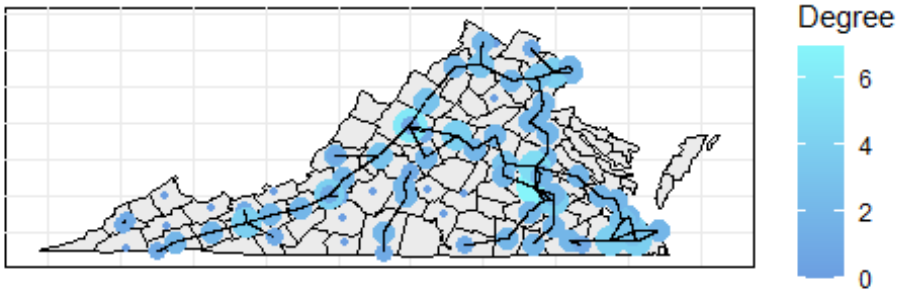
Texas



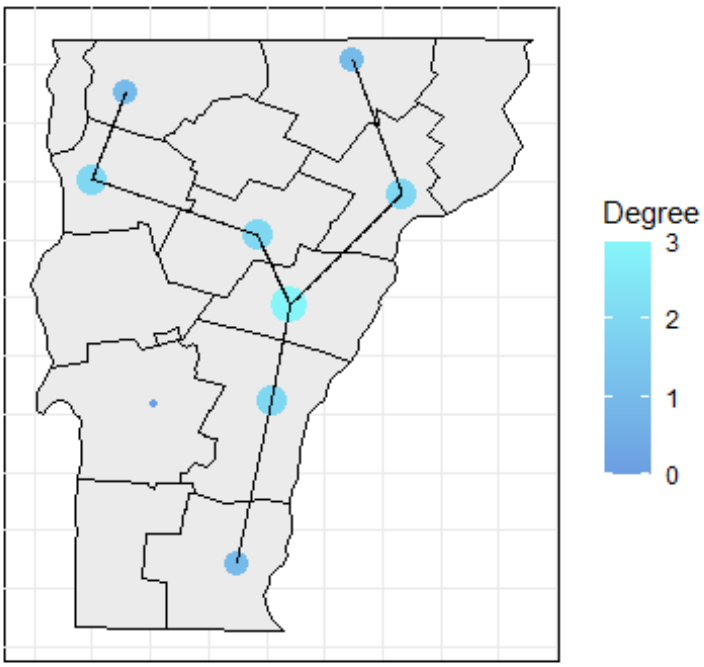
Utah



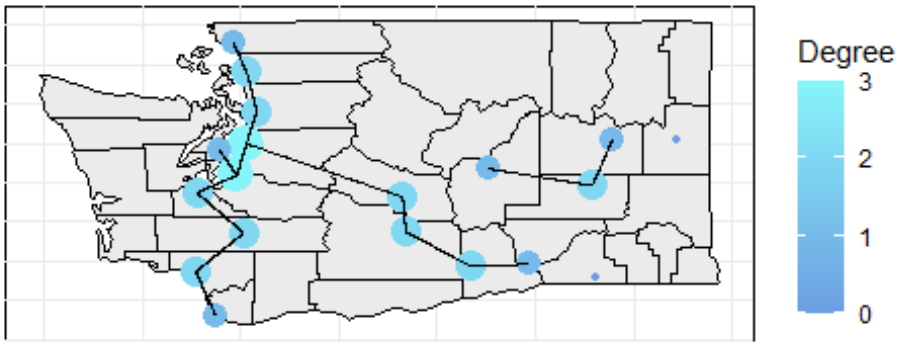
Virginia



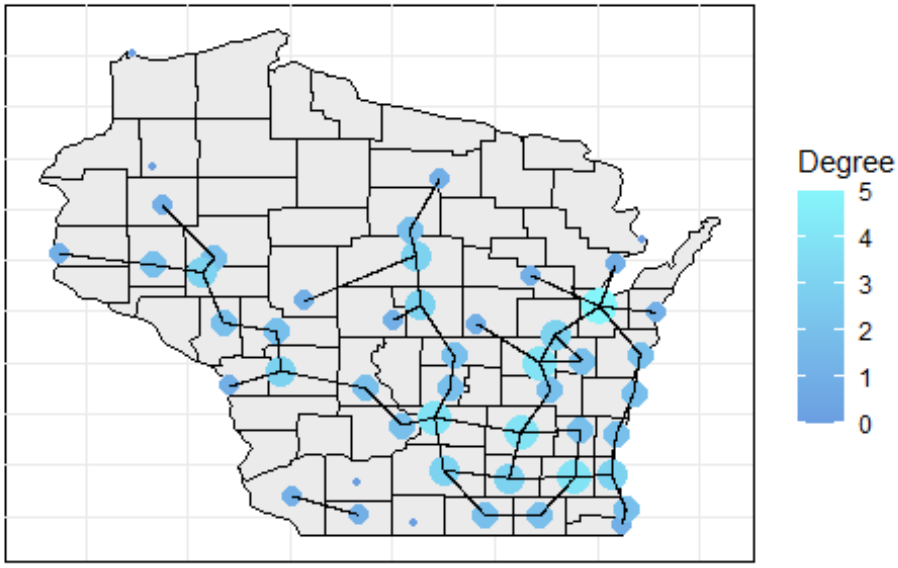
Vermont



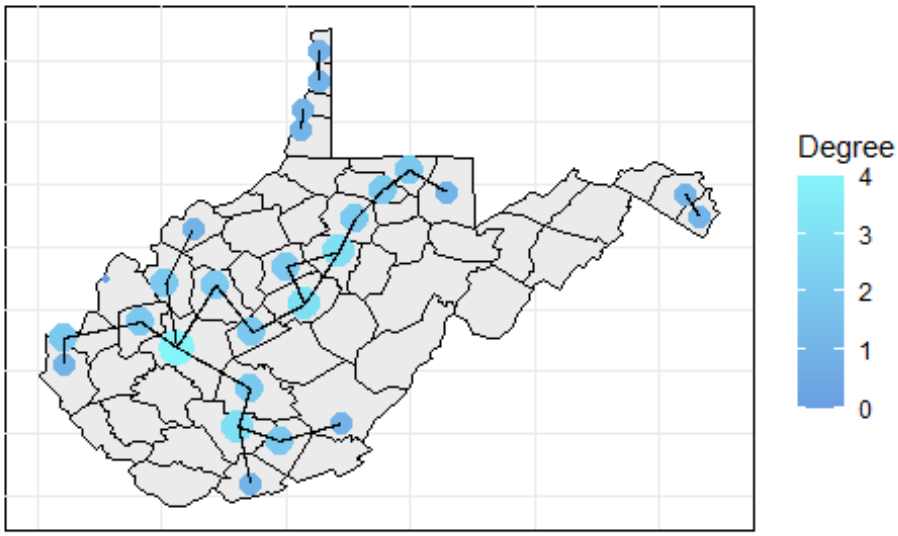
Washington



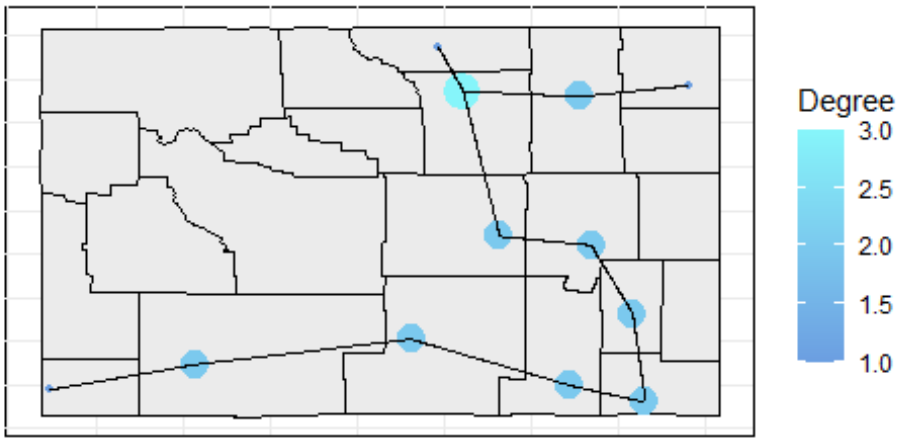
Wisconsin



West Virginia



Wyoming



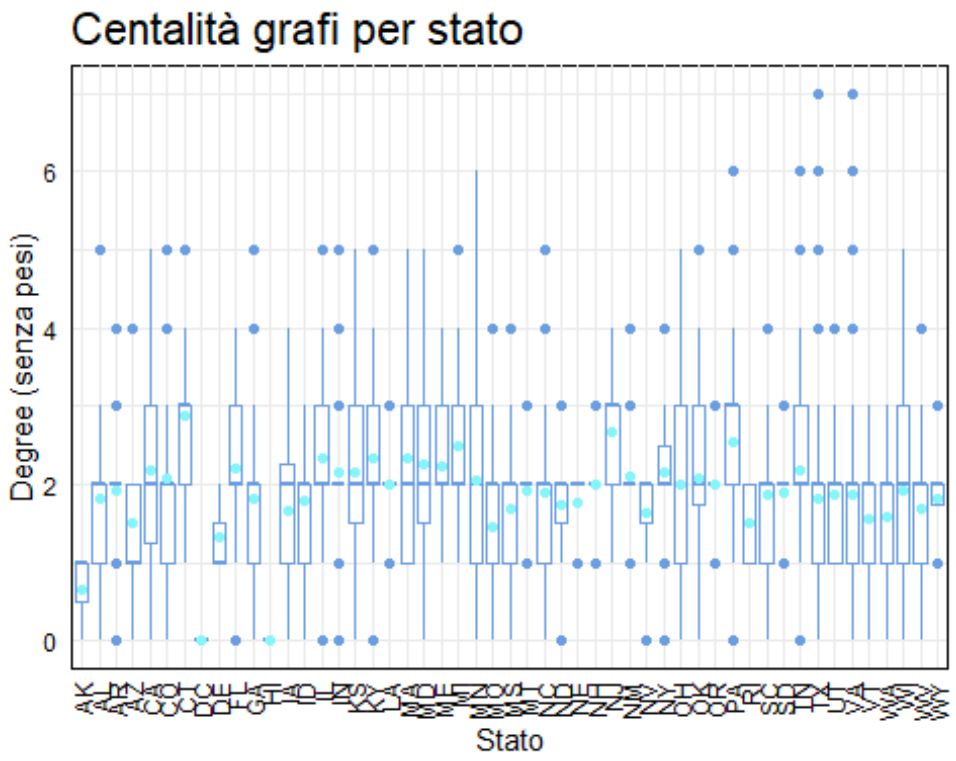
Si può notare come molti stati abbiano una topologia circolare (California, Connecticut, Florida, Massachusetts, Missouri, Mississippi, Montana, North Carolina, New Mexico, South Carolina, Utah, Virginia) o a stella (Alabama, Arkansas, Arizona, Colorado, Georgia, Iowa, Illinois, Indiana, Kansas, Kentucky,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

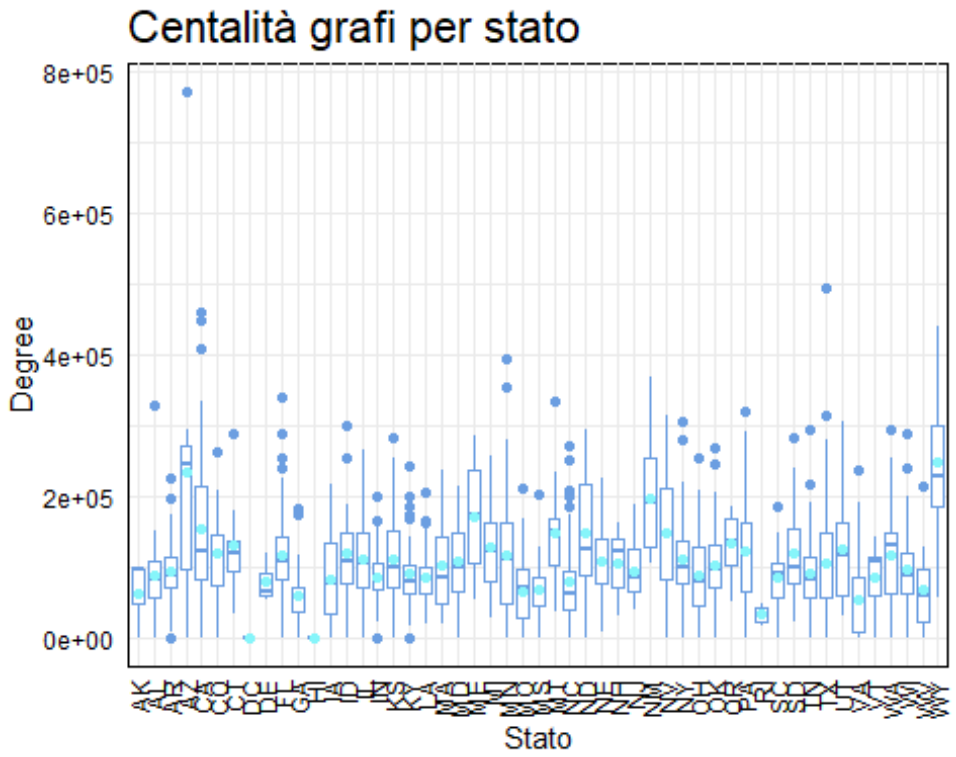
Di seguito l'analisi locale sulle reti di ogni stato.

(Distribuzione delle) Centralità

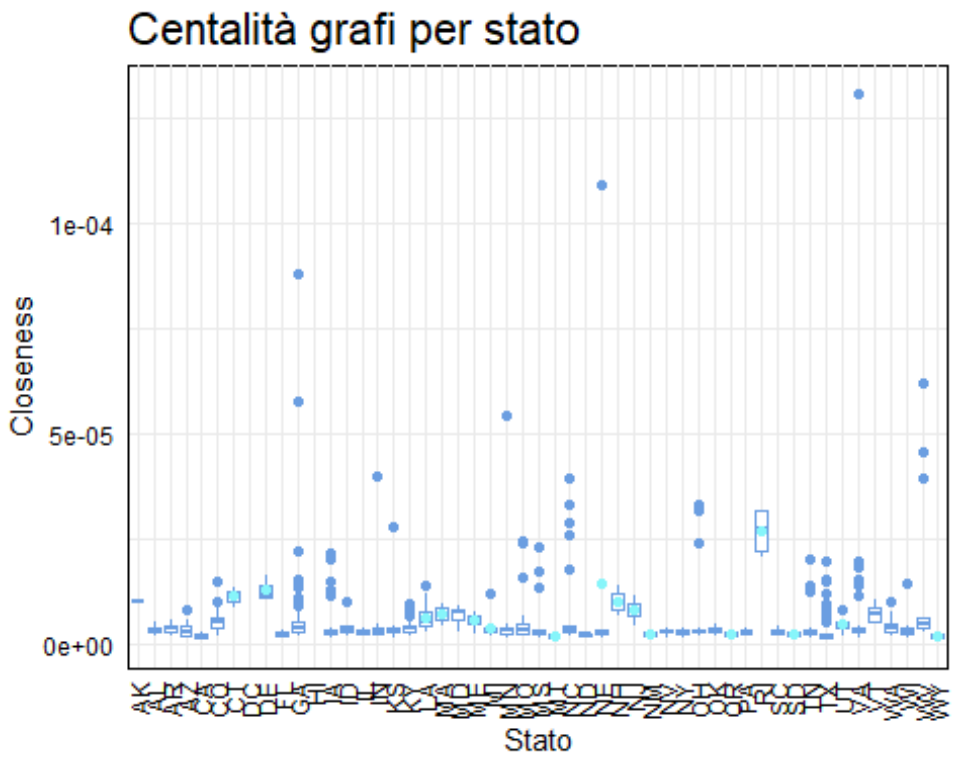
```
rbs_c = do.call(rbind, lapply(names(roads_by_state), function(state){
  centrality(roads_by_state[[state]], state)
}))

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)))
```



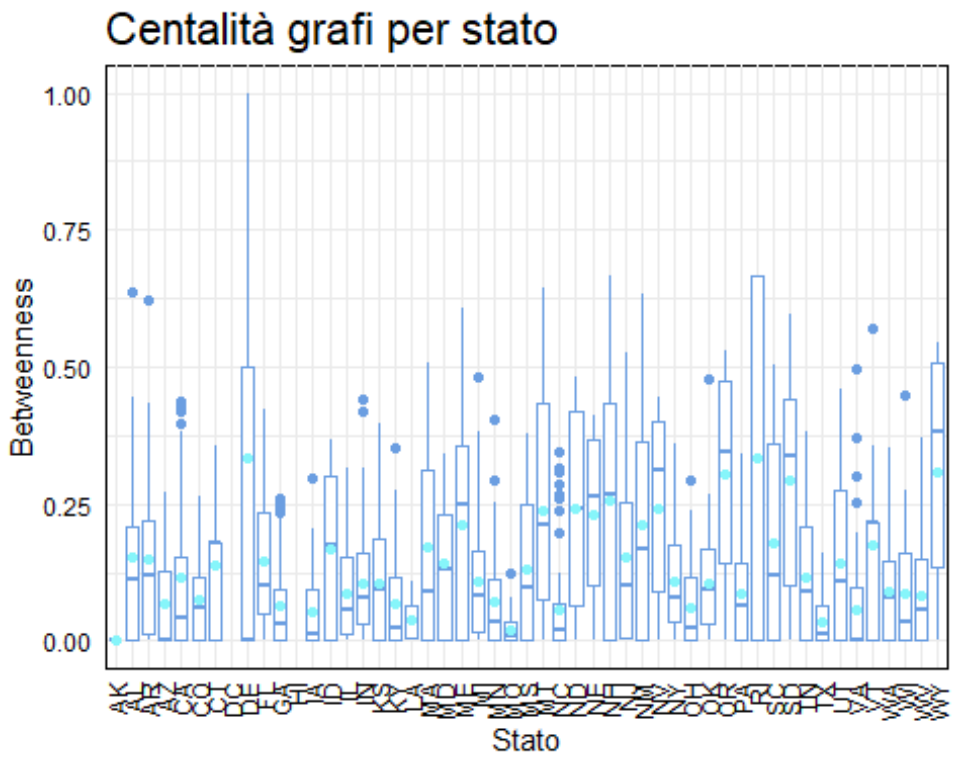

```
rbs_c %>%
  ggplot(aes(x=state, y=degreeW)) +
    geom_boxplot(color="#6B9EE1") +
    geom_point(data = rbs_c %>% group_by(state) %>% summarize(mean_degreeW =
mean(degreeW)), aes(x = state, y = mean_degreeW, group = 1), color = "#86F5FA") +
    scale_y_continuous() +
    labs(title = "Centalità grafi 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)))
```



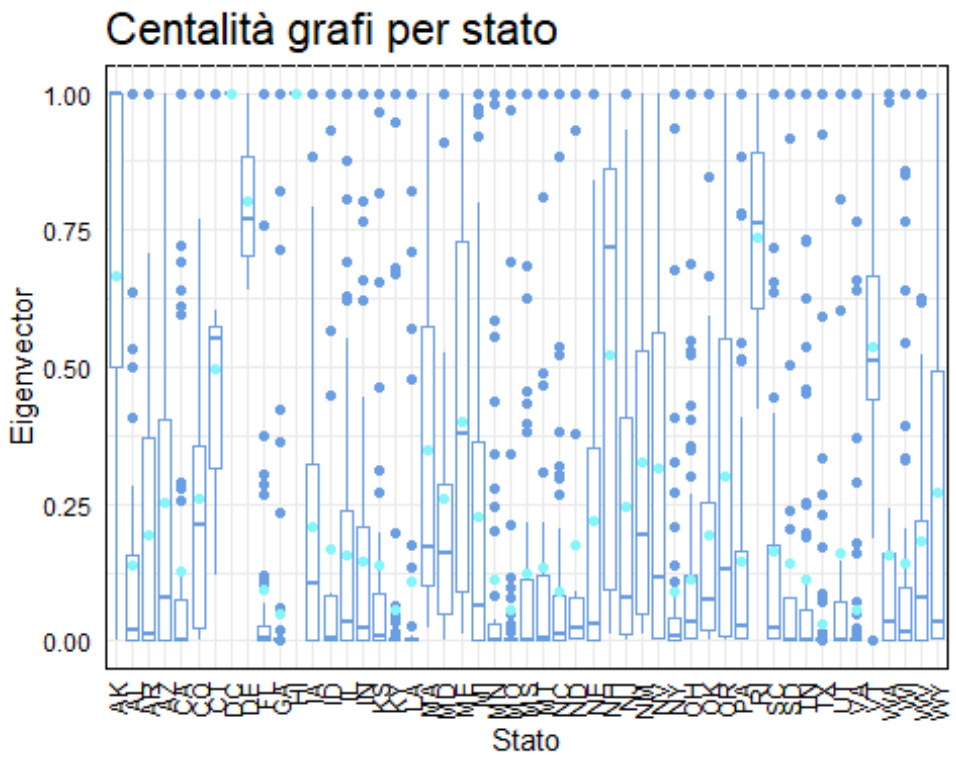

```
rbs_c %>%
  ggplot(aes(x=state, y=closeness)) +
    geom_boxplot(color="#6B9EE1") +
    geom_point(data = rbs_c %>% group_by(state) %>% summarize(mean_closeness =
mean(closeness)), aes(x = state, y = mean_closeness, group = 1), color =
"#86F5FA") +
    scale_y_continuous() +
    labs(title = "Centalità 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()`).
```



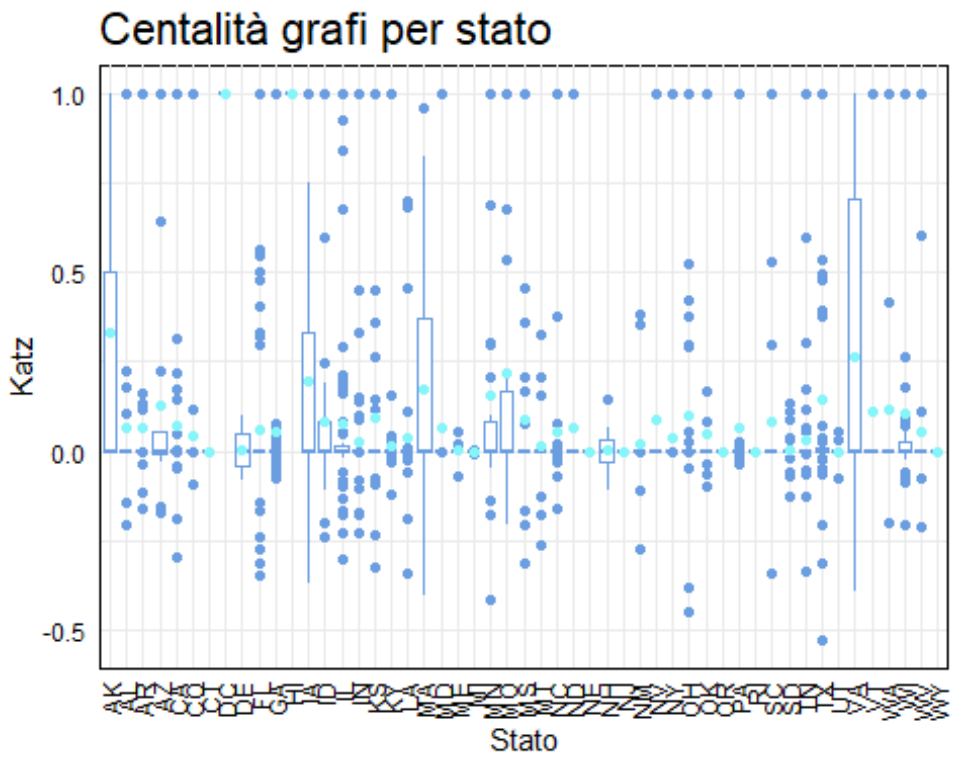

```
rbs_c %>%
  ggplot(aes(x=state, y=betweenness)) +
    geom_boxplot(color="#6B9EE1") +
    geom_point(data = rbs_c %>% group_by(state) %>% summarize(mean_betweenness =
mean(betweenness)), aes(x = state, y = mean_betweenness, group = 1), color =
"#86F5FA") +
    scale_y_continuous() +
    labs(title = "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()`).
```



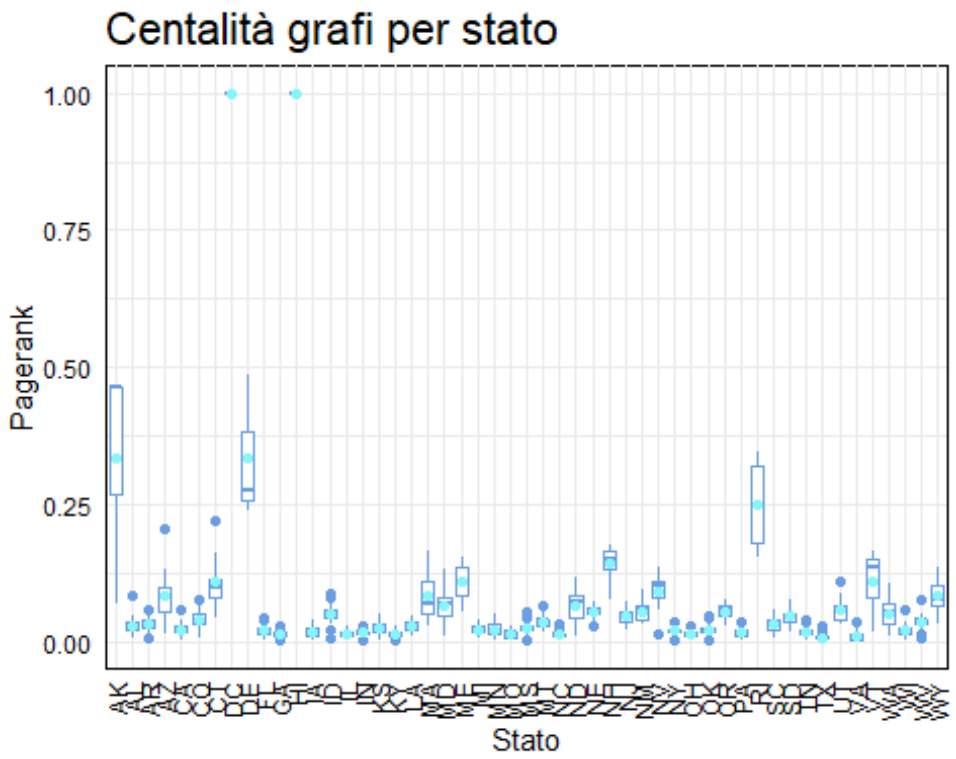

```
rbs_c %>%
  ggplot(aes(x=state, y=eigen)) +
    geom_boxplot(color="#6B9EE1") +
    geom_point(data = rbs_c %>% group_by(state) %>% summarize(mean_eigen =
mean(eigen)), aes(x = state, y = mean_eigen, group = 1), color = "#86F5FA") +
    scale_y_continuous() +
    labs(title = "Centalità grafi per 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)))
```

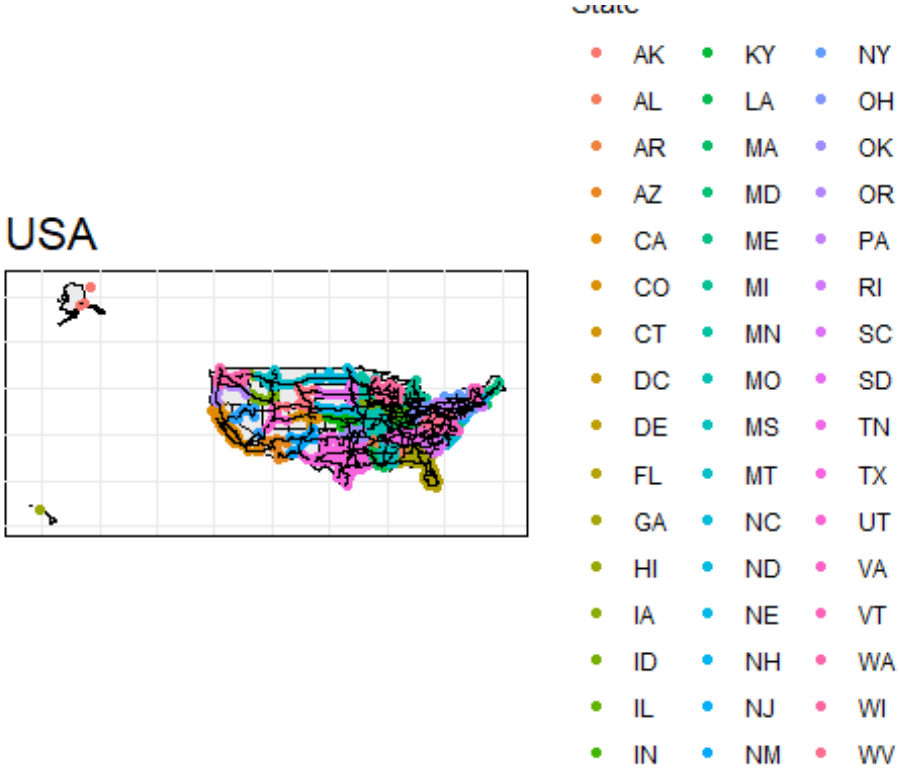


Si può notare come per quasi tutte le metriche di centralità i grafi sui singoli stati abbiano mediamente una bassa centralità (grado intorno al 2, closeness quasi zero, katz e pagerank molti sono vicino a zero).

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)))
```



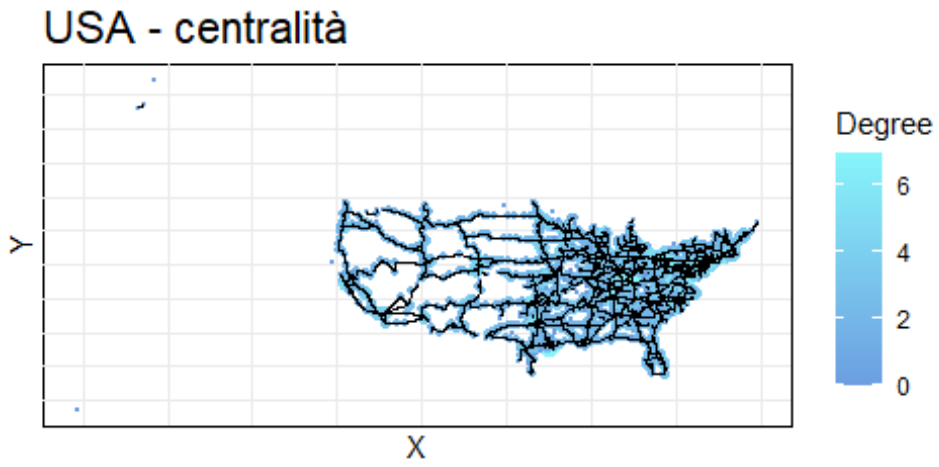
Si può osservare come il grafo sia più connesso rispetto ai singoli stati (anche alcune contee prima disconnesse ora sono connesse), si può notare come la parte ovest sia più rada mentre la parte est sia più densa.

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

Locale - centralità

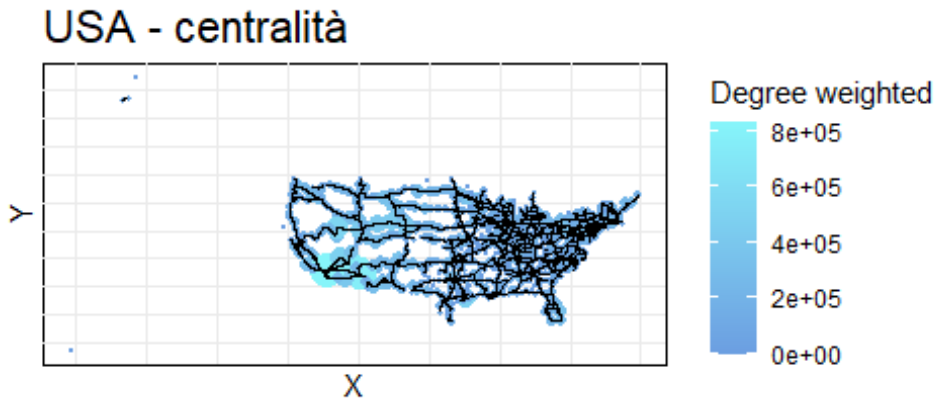
```
r_c = centrality(roads, "USA")

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)))
```



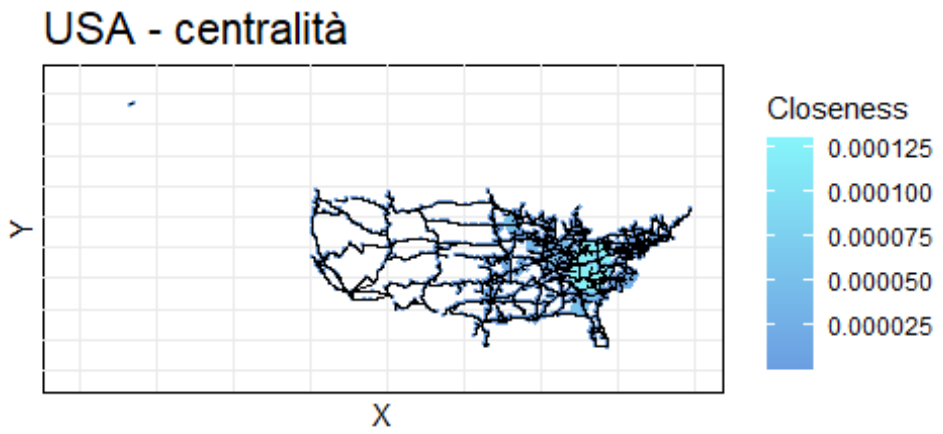

```
roads %>%
  ggraph(layout = 'manual', x = x, y = y) +
    geom_node_point(aes(size = I(r_c$degreeW/150000), colour = r_c$degreeW)) +
    geom_edge_link() +
    scale_x_continuous(guide = "none") +
    scale_y_continuous(guide = "none") +
    coord_fixed(ratio = 1) +
    scale_color_gradient(low="#6B9EE1", 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)))
```

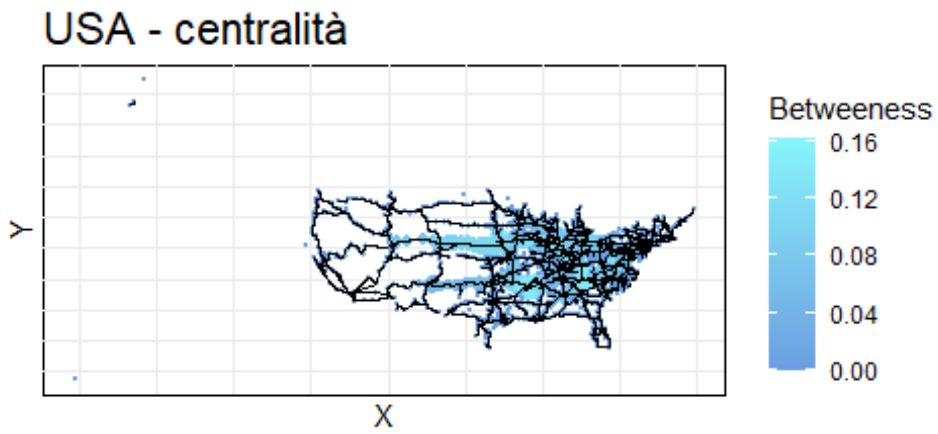



```
roads %>%
  ggraph(layout = 'manual', x = x, y = y) +
    geom_node_point(aes(size = I(r_c$closeness*50000), colour = r_c$closeness)) +
    geom_edge_link() +
    scale_x_continuous(guide = "none") +
    scale_y_continuous(guide = "none") +
    coord_fixed(ratio = 1) +
    scale_color_gradient(low="#6B9EE1", 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"),
```

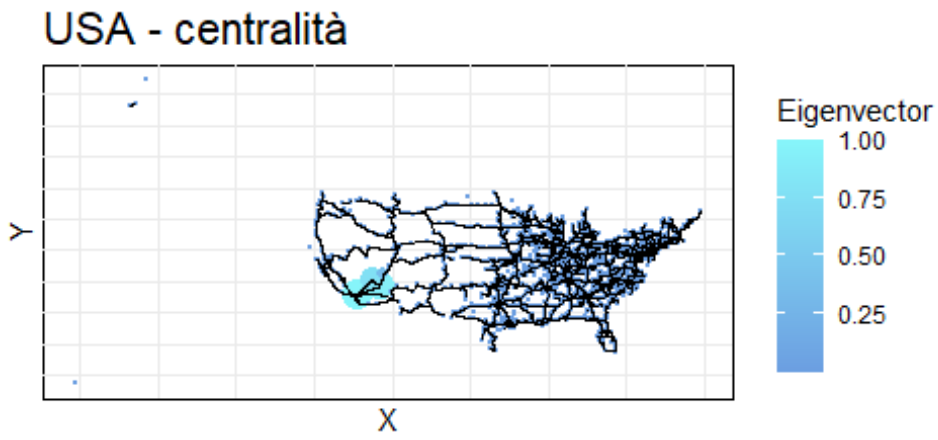
```
panel.grid.major.y=element_line(colour="#ebebeb"),
plot.title=element_text(size=rel(1.5)))

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



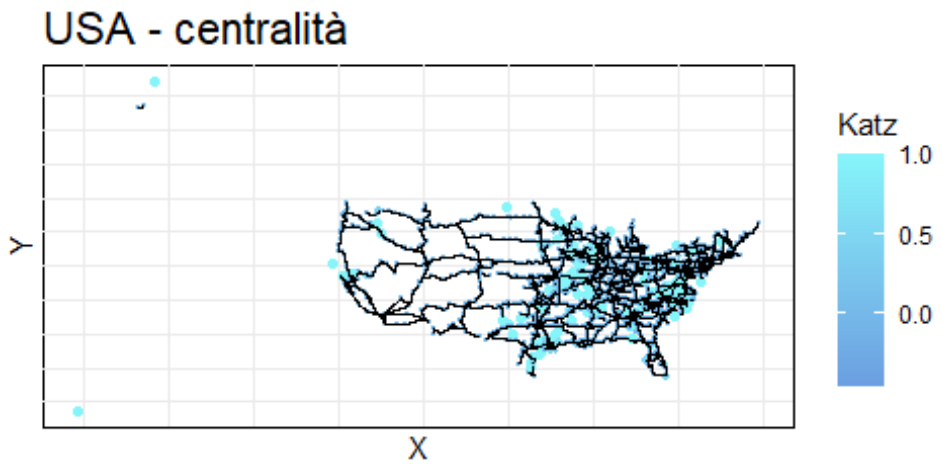

```
roads %>%
  ggraph(layout = 'manual', x = x, y = y) +
    geom_node_point(aes(size = I(r_c$betweenness*25), colour = r_c$betweenness)) +
    geom_edge_link() +
    scale_x_continuous(guide = "none") +
    scale_y_continuous(guide = "none") +
    coord_fixed(ratio = 1) +
    scale_color_gradient(low="#6B9EE1", 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)))
```



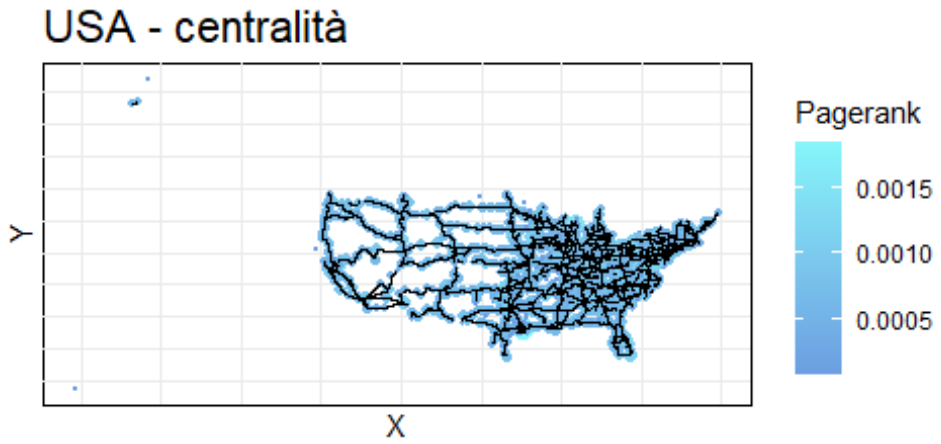

```
roads %>%
  ggraph(layout = 'manual', x = x, y = y) +
    geom_node_point(aes(size = I(r_c$eigen*5), colour = r_c$eigen)) +
    geom_edge_link() +
    scale_x_continuous(guide = "none") +
    scale_y_continuous(guide = "none") +
    coord_fixed(ratio = 1) +
    scale_color_gradient(low="#6B9EE1", 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).

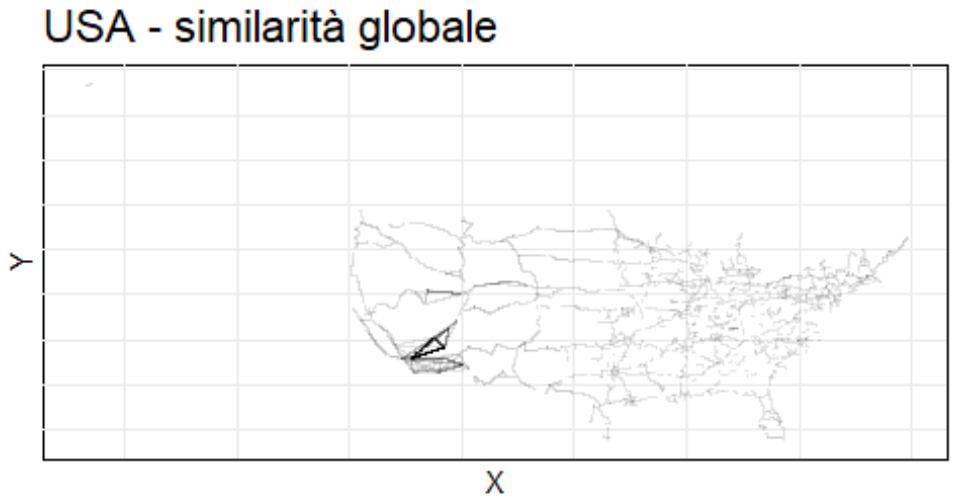
Locale - similarità e eterogeneità

```
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"),
```

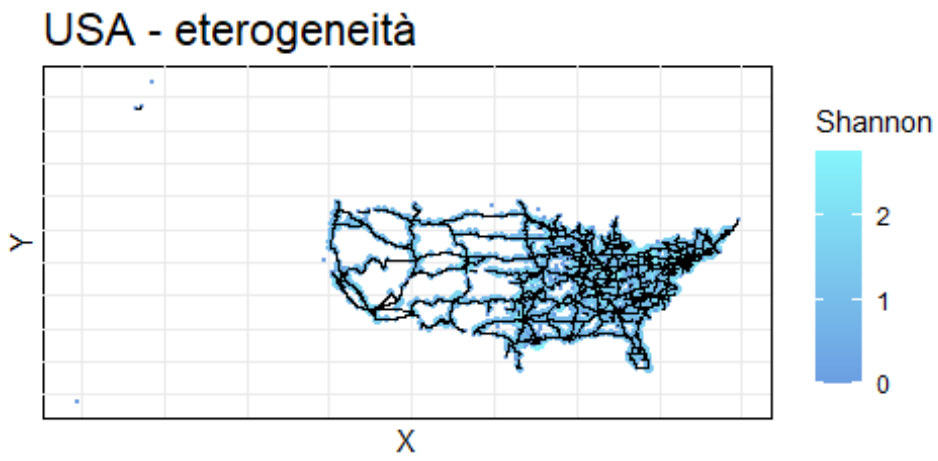
```
panel.grid.major.y=element_line(colour="#ebebeb"),
plot.title=element_text(size=rel(1.5)))
```

```
## 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.
```

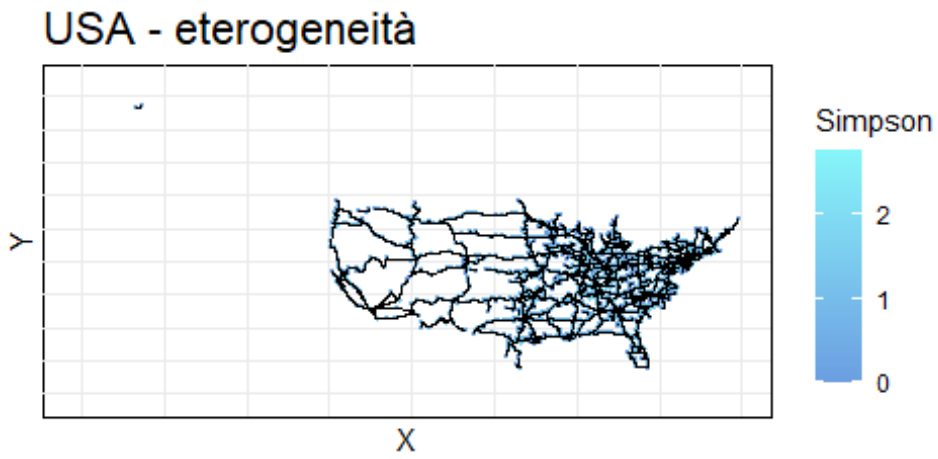



```
roads %>%
  ggraph(layout = 'manual', x = x, y = y) +
    geom_node_point(aes(size = I(r_h$shannon), colour = r_h$shannon))+
    geom_edge_link() +
    scale_x_continuous(guide = "none") +
    scale_y_continuous(guide = "none") +
    coord_fixed(ratio = 1) +
    scale_color_gradient(low="#6B9EE1", 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).

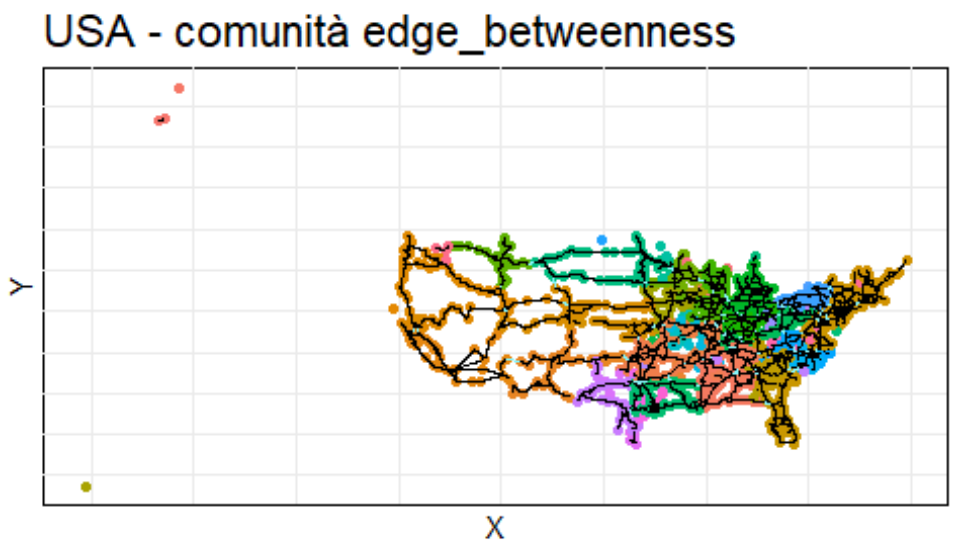
Per l'eterogeneità (misura opposta alla similarità), si vede come la parte centro-est sia un po' più eterogenea.

Gruppi - comunità e clustering

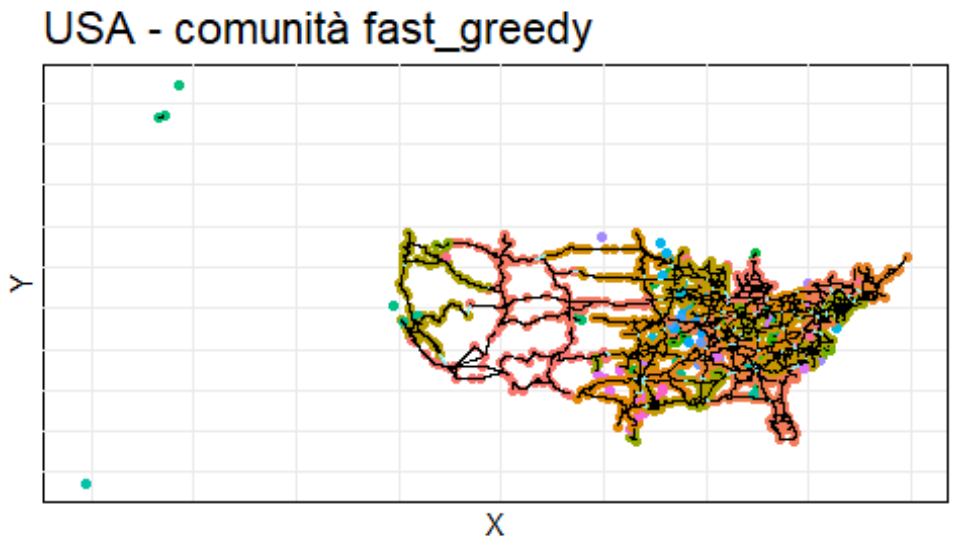
```
#creazione grafo fully connected per clustering
edges = expand.grid(county1 = as.character(roads_nodes$county), county2 =
as.character(roads_nodes$county), stringsAsFactors = FALSE)
edges = edges[edges$county1 < edges$county2, ]
edges = edges %>%
  left_join(roads_nodes, by = c("county1" = "county")) %>%
  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)))
  )
}
```

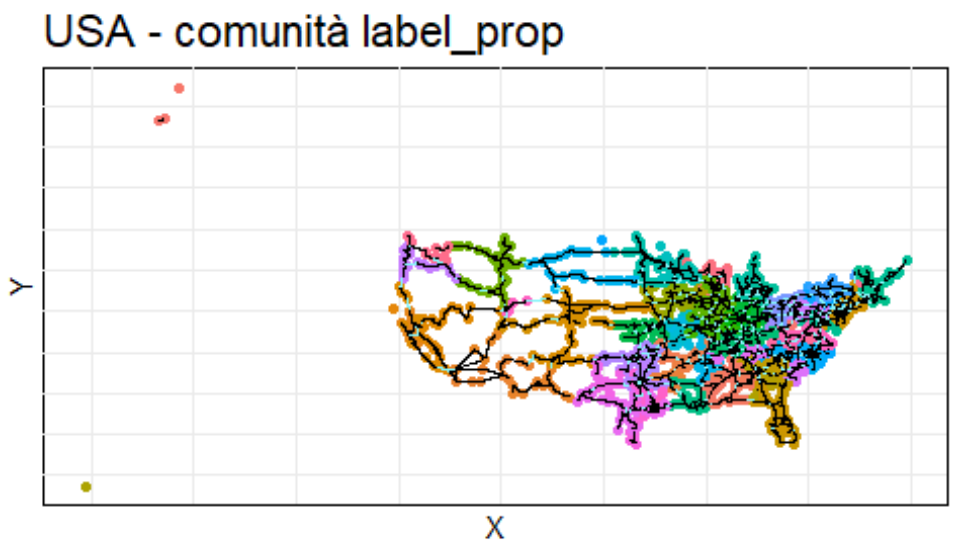
```
## [1] "Comunità edge_betweenness : modularità 0.920843633522139 numero comunità
158"
```



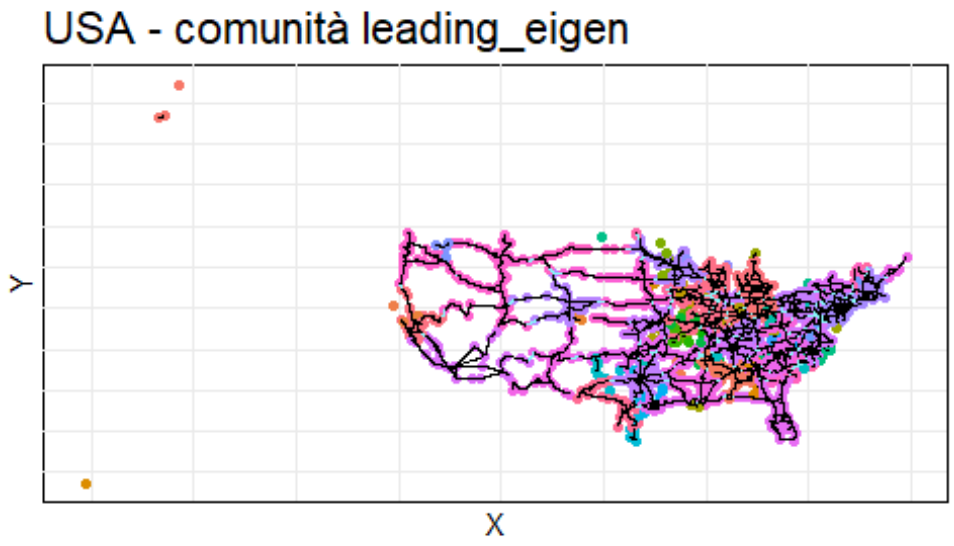

```
## [1] "Comunità fast_greedy : modularità 0.922059133734641 numero comunità 161"
```



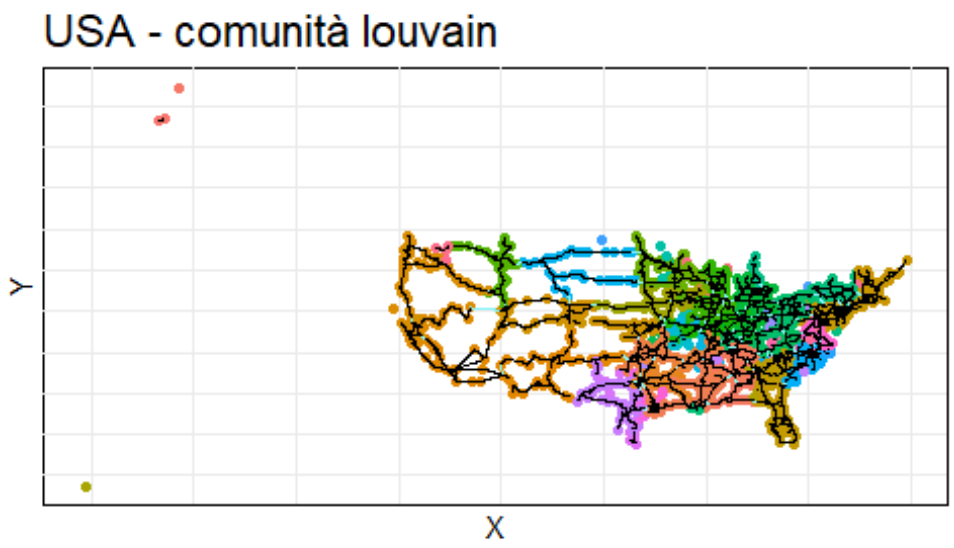

```
## [1] "Comunità label_prop : modularità 0.852055777405775 numero comunità 280"
```



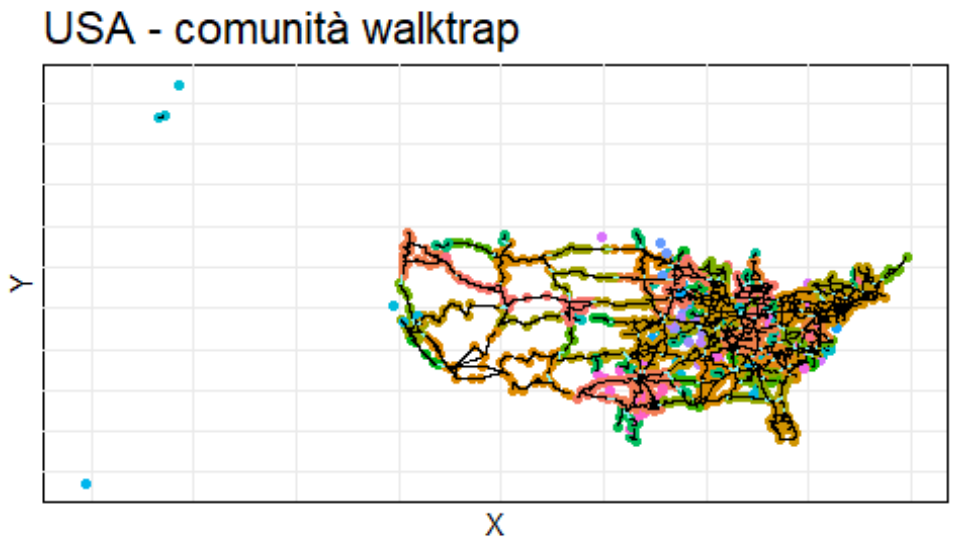

```
## [1] "Comunità leading_eigen : modularità 0.899372221027243 numero comunità 179"
```



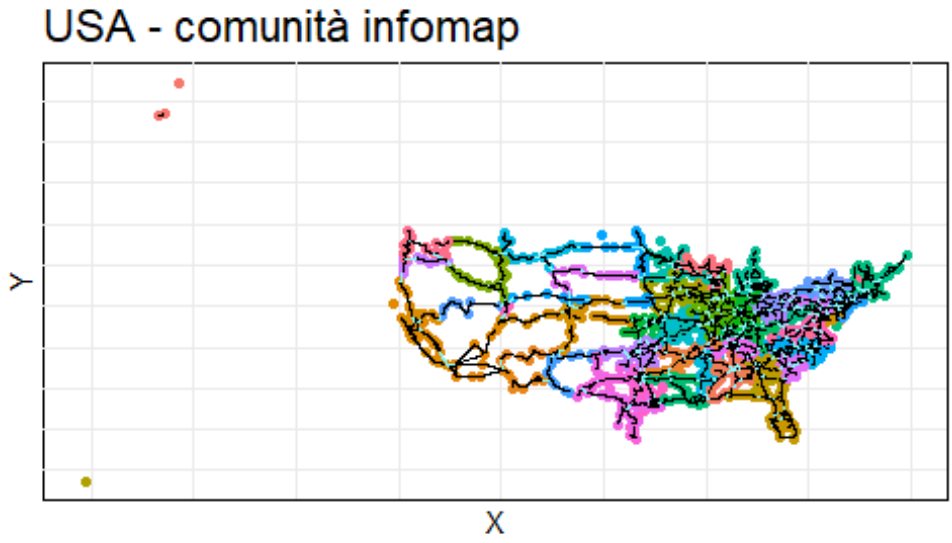

```
## [1] "Comunità louvain : modularità 0.922943547172007 numero comunità 161"
```



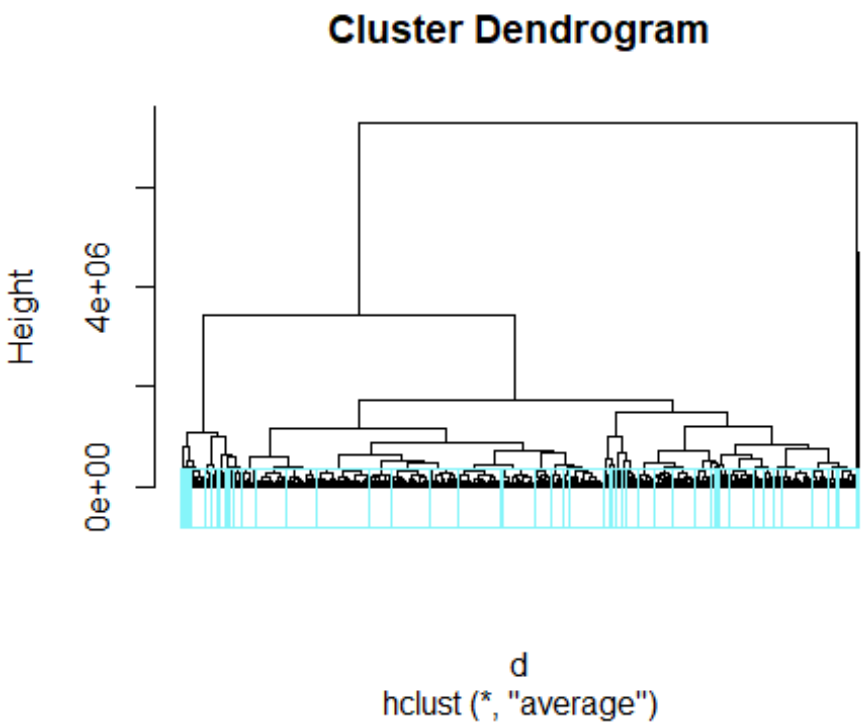

```
## [1] "Comunità walktrap : modularità 0.902676594334143 numero comunità 205"
```

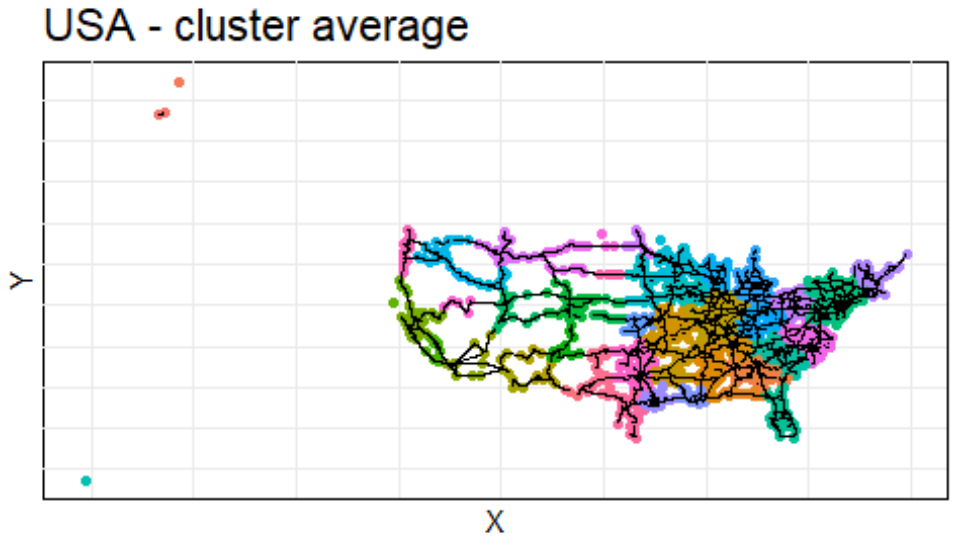


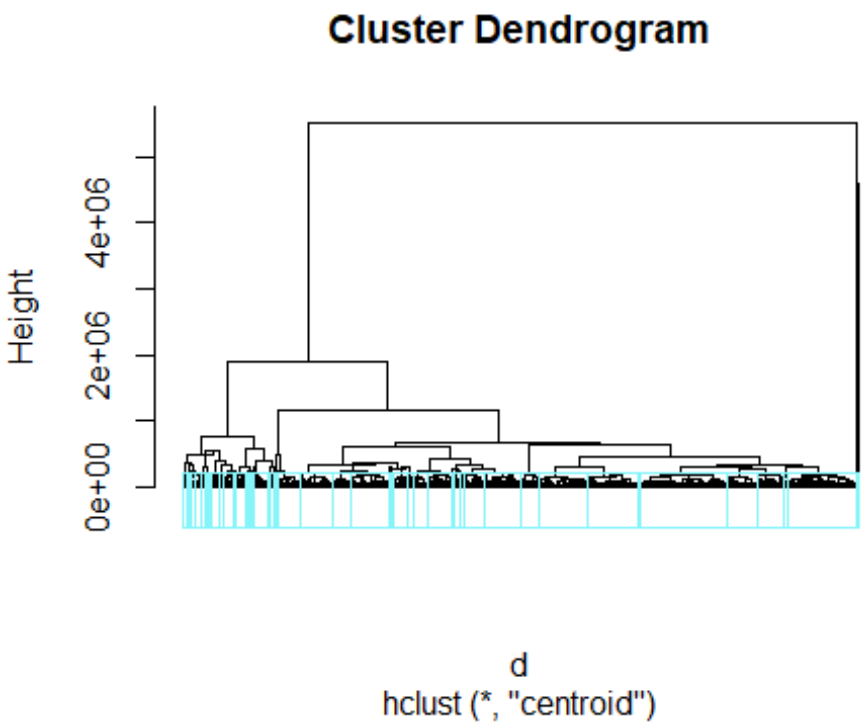

```
## [1] "Comunità infomap : modularità 0.83137694690454 numero comunità 329"
```

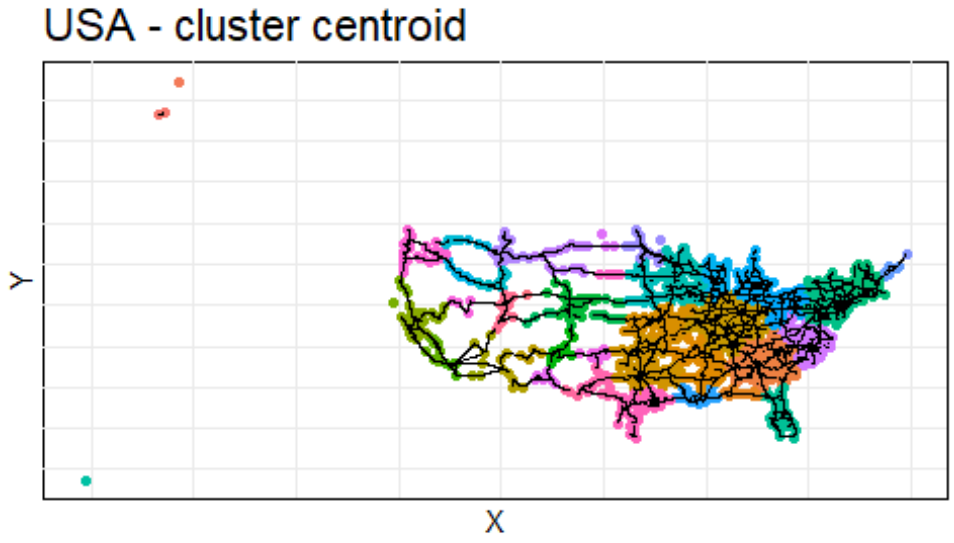


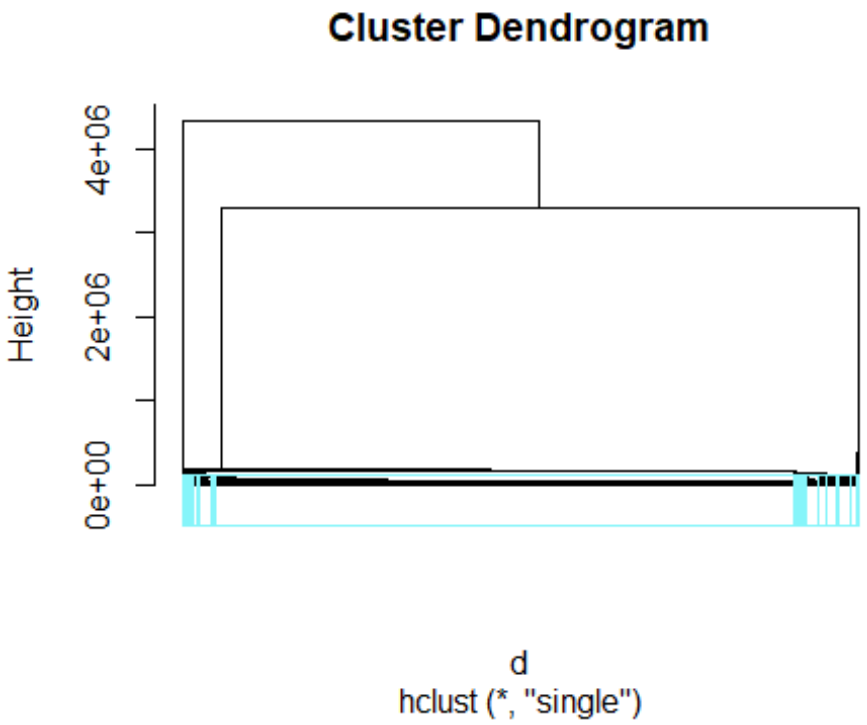

```
for(i in 1:nrow(r_cl)){
  r_cl$result[[i]]$labels = rep("", length(r_cl$result[[1]]$labels))
  plot(r_cl$result[[i]], hang=-1)
  rect.hclust(r_cl$result[[i]], k = 51, border="#86F5FA")
  print(
    roads %>%
      ggraph(layout = 'manual', x = x, y = y) +
        geom_node_point(aes(colour = factor(cutree(r_cl$result[[i]], k = 51)))) +
        geom_edge_link() +
        scale_x_continuous(guide = "none") +
        scale_y_continuous(guide = "none") +
        coord_fixed(ratio = 1) +
        scale_color_discrete(guide = "none") +
        labs(title = paste("USA - cluster", r_cl$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)))
  )
}
```

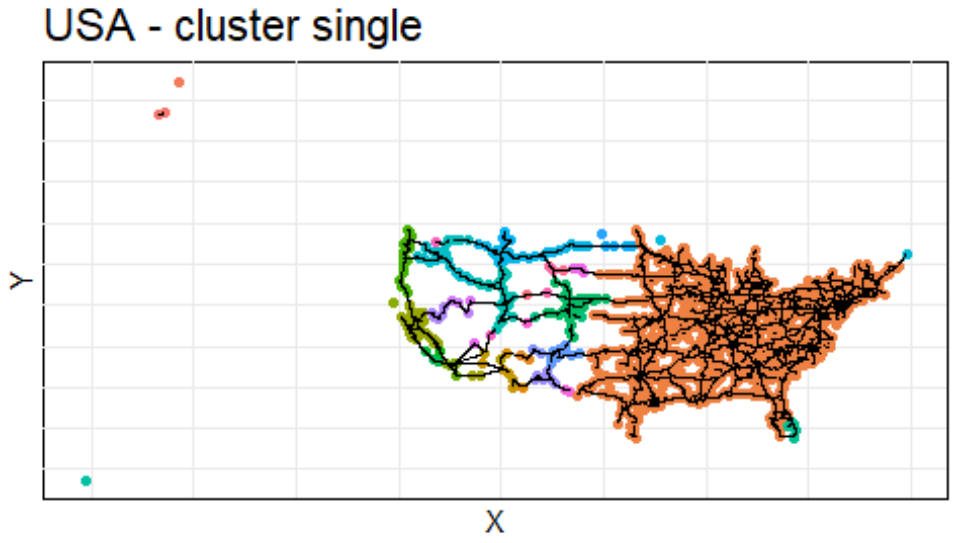

Cluster Dendrogram
Height
4e+06
0e+00
d
hclust (*, "average")

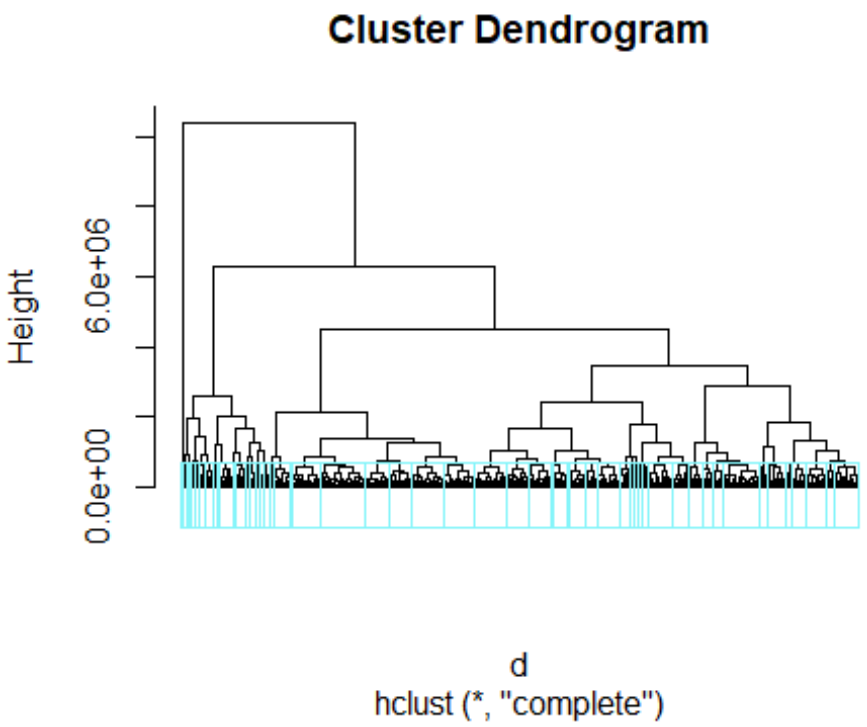

USA - cluster average
Y
X

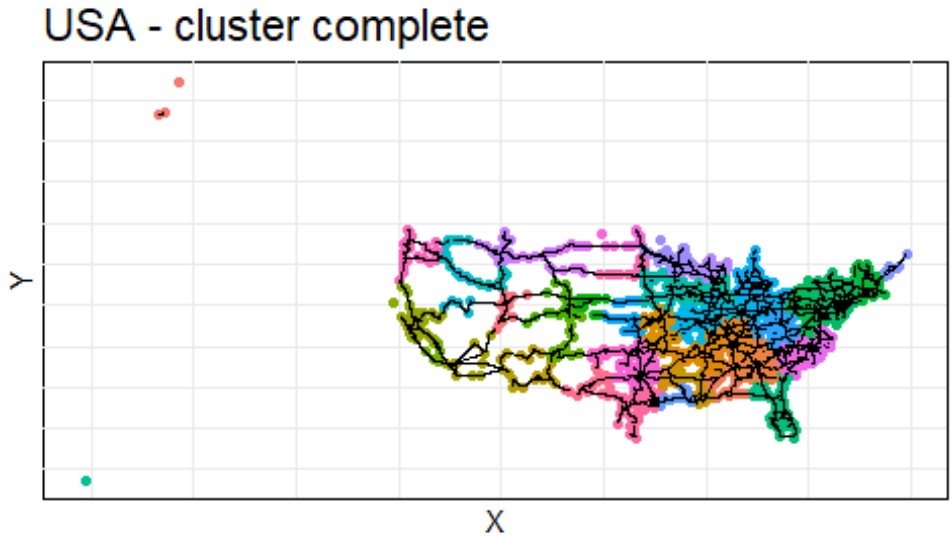

Cluster Dendrogram
Height
0e+00
2e+06
4e+06
d
hclust (*, "centroid")


USA - cluster centroid
Y
X


Cluster Dendrogram
Height
0e+00
2e+06
4e+06
d
hclust (*, "single")


USA - cluster single
Y
X



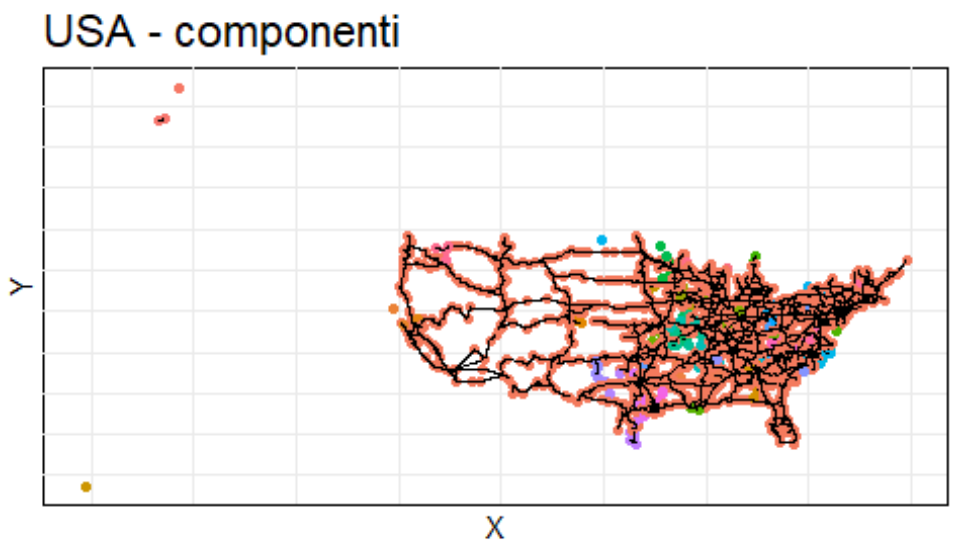


Per le comunità, si nota che con tutti i metodi il numero di comunità (160-330) è superiore al numero di stati (51), si può anche notare che in tutti i metodi in generale ci siano più comunità a est ripetto a ovest. Dato che i metodi label_prop e infomap hanno modulatità minore degli altri metodi, è 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)))
```

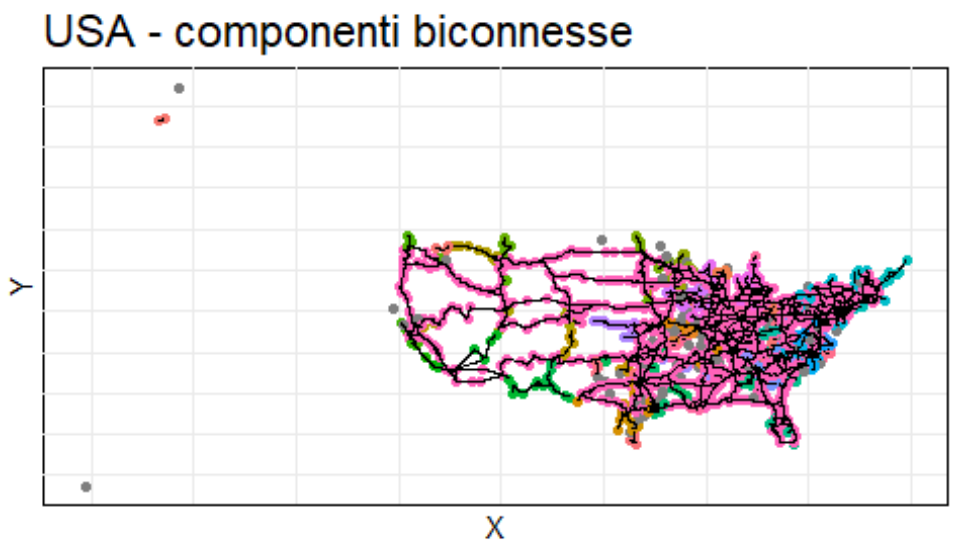



```
print(paste("componenti: numero componenti biconnesse",
r_con$biconnected_components$no, "la più grande raggiunge il",
round(max(sapply(r_con$biconnected_components, length))/vcount(roads)*100, 2), "%
dei nodi"))
```

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

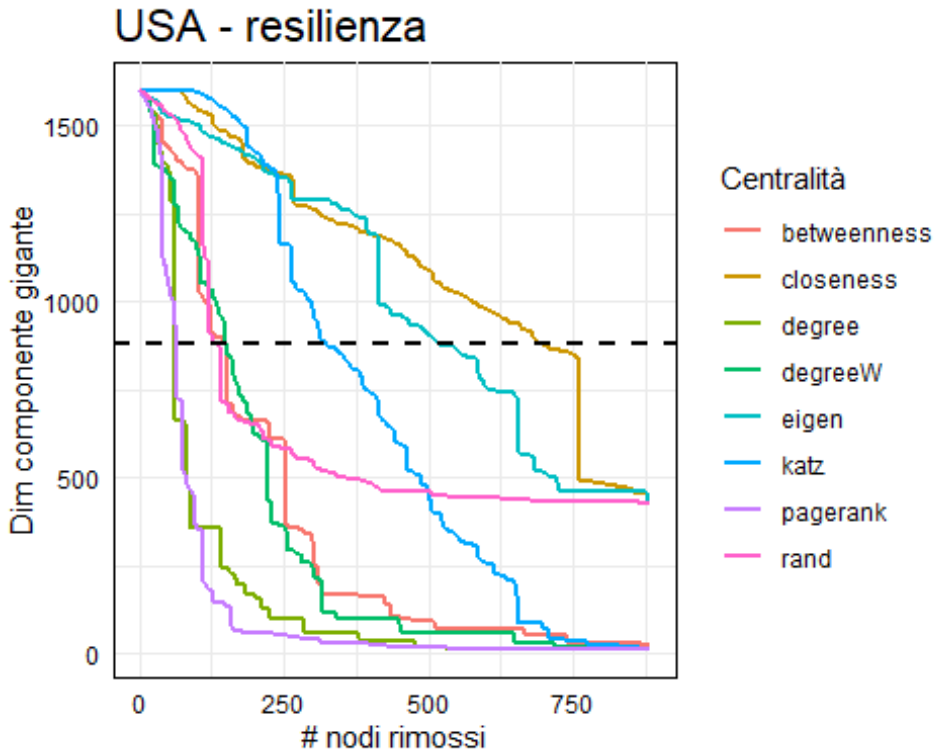
```
node_bicomponent = rep(NA, vcount(roads))
for(i in seq_along(r_con$biconnected_components$components)){
  node_bicomponent[r_con$biconnected_components$components[[i]]] = i
}

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%.

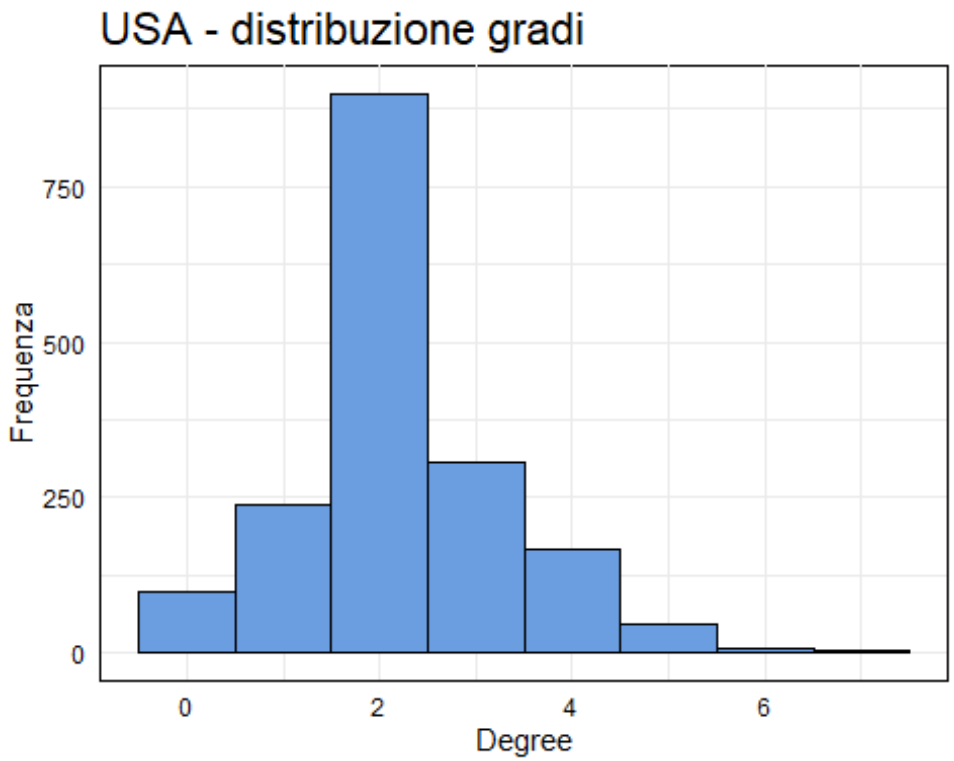
Globale - geodesica

```
r_g = geodesic(roads)

print(paste("geodesica: media", r_g$geodesic_mean, "diametro",
r_g$geodesic_diameter))

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

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

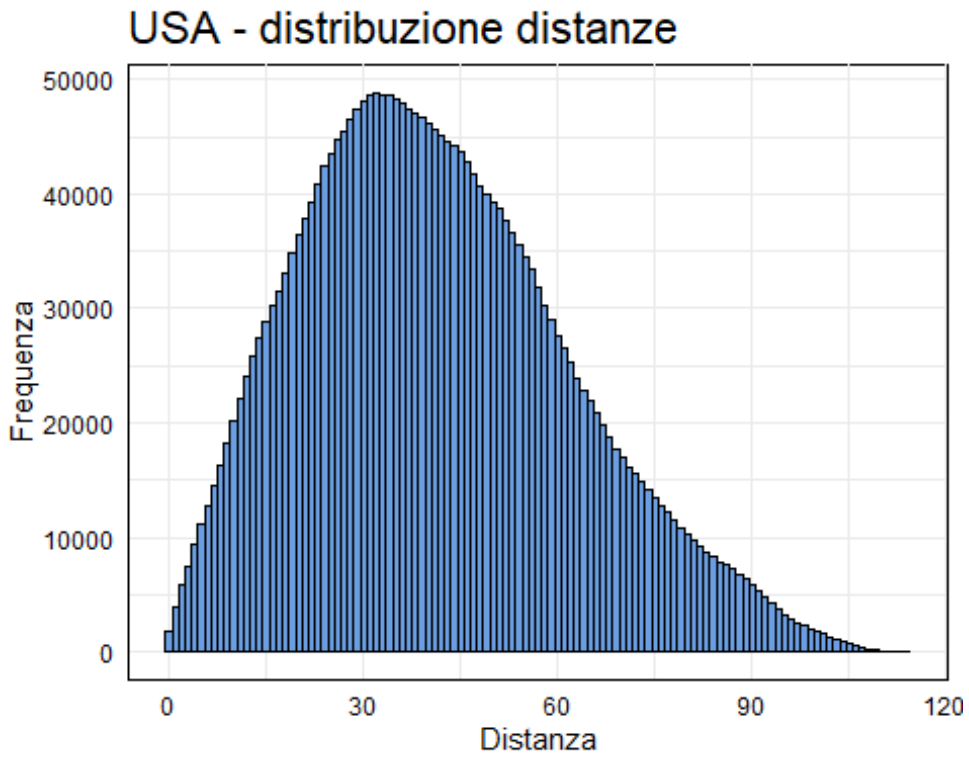
Globale - power law

```
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)))
```



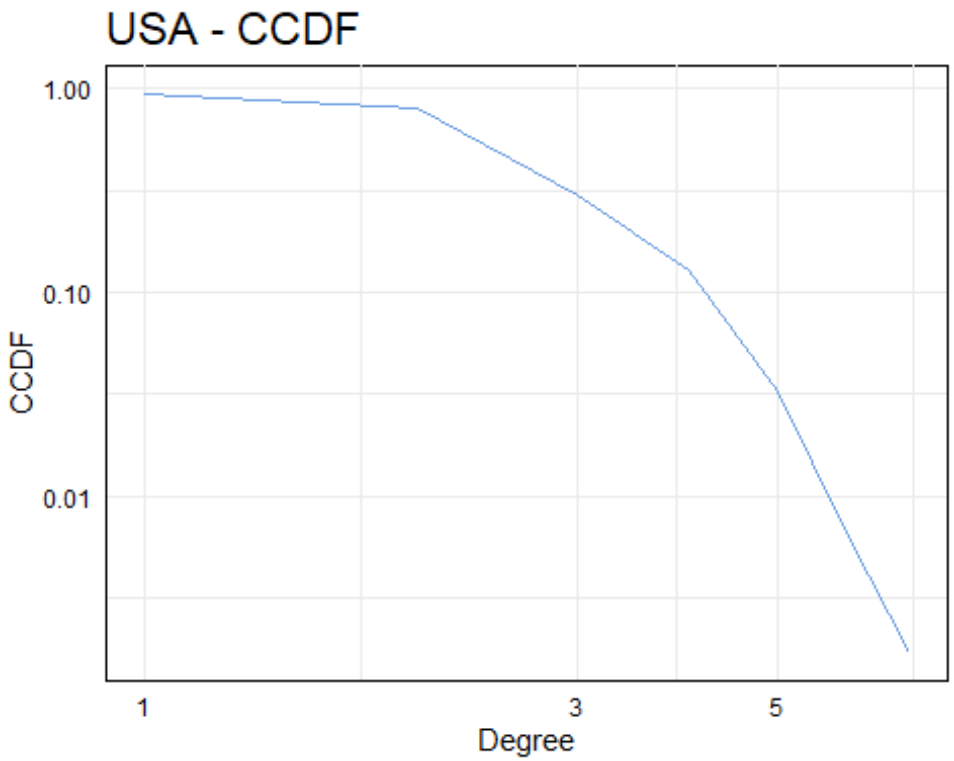

```
r_p$summary
```

```
##    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.