Exploring our data

CASE STUDIES: NETWORK ANALYSIS IN R

Edmund Hart
Instructor

```r
library(igraph)
library(dplyr)
library(lubridate)
bike_dat <- read.csv("datasets/bike2_test3.csv", stringsAsFactors = FALSE)
str(bike_dat)
```

```
'data.frame':    52800 obs. of  13 variables:
 $ tripduration     : int  295 533 1570 2064 2257 296 412...
 $ from_station_id  : int  49 165 25 300 85 174 75 45 85 99 ...
 $ from_station_name: chr  "Dearborn St & Monroe St" ...
 $ to_station_id    : int  174 308 287 296 313 198 56 147 174 99 ...
 $ to_station_name  : chr  "Canal St & Madison St" ...
 $ usertype         : chr  "Subscriber" "Subscriber" "Customer"...
 $ gender           : chr  "Male" "Male" "" "" ...
 $ birthyear        : int  1964 1972 NA NA 1963 1973 1989 1965 1983 1983 ...
 $ from_latitude    : num  41.9 42 41.9 41.9 41.9 ...
 $ from_longitude   : num  -87.6 -87.7 -87.6 -87.6 -87.6 ...
 $ to_latitude      : num  41.9 41.9 41.9 41.9 41.9 ...
 $ to_longitude     : num  -87.6 -87.7 -87.6 -87.6 -87.6 ...
 $ geo_distance     : num  859 1882 2159 288 3044 ...
```

Creating the bike sharing graph

```
trip_df <- bike_dat %>%
    group_by(from_station_id, to_station_id) %>%
    summarize(weights = n())
head(trip_df)
```

```
# A tibble: 6 x 3
# Groups: from_station_id [1]
  from_station_id to_station_id weights
            <int>         <int>   <int>
1               5             5       2
2               5            14       1
3               5            16       1
4               5            25       3
5               5            29       3
6               5            33       1
```

Creating the bike sharing graph

```
trip_g <- graph_from_data_frame(trip_df[, 1:2])
# add edge weights
E(trip_g)$weight <- trip_df$weights
# Quick exploration of our graph
gsize(trip_g)
```

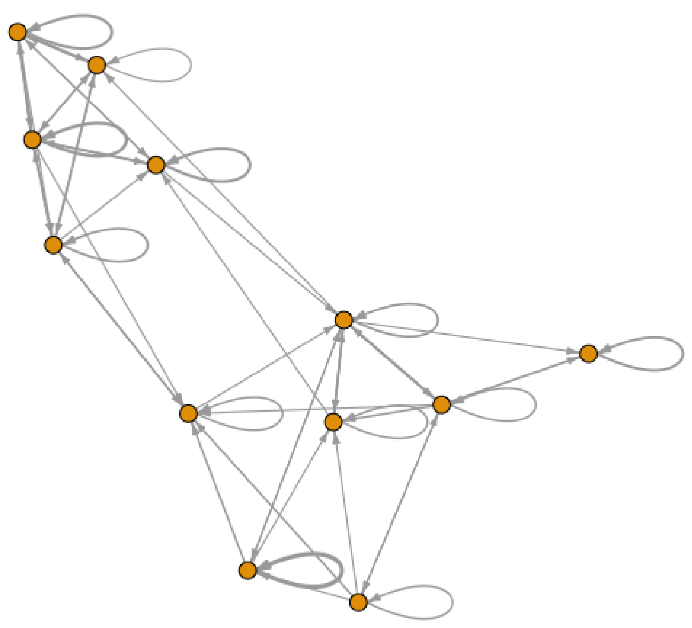
```
19052
```

```
gorder(trip_g)
```

```
300
```

Explore the graph

```
sg <- induced_subgraph(trip_g, 1:12)
plot(sg, vertex.label = NA, edge.arrow.width = 0.8,
     edge.arrow.size = 0.6,
      margin = 0,
     vertex.size = 6,
     edge.width = log(E(sg)$weight + 2))
```



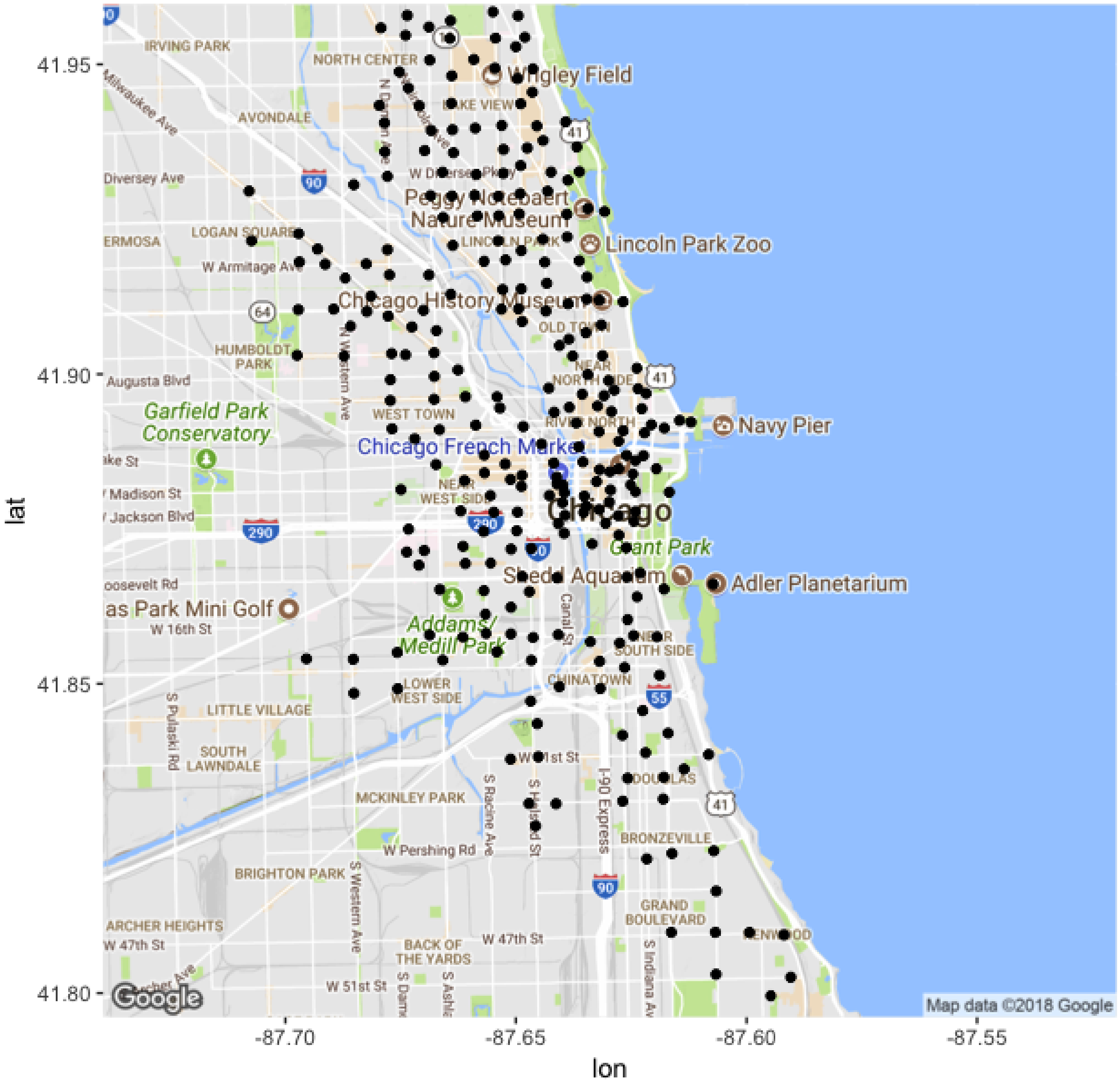
Let's practice!

CASE STUDIES: NETWORK ANALYSIS IN R

Compare graph distance vs. geographic distance

CASE STUDIES: NETWORK ANALYSIS IN R

Edmund Hart
Instructor


41.95
41.90
41.85
41.80
lat
-87.70
-87.65
-87.60
-87.55
lon
IRVING PARK
NORTH CENTER
Wrigley Field
LAKE VIEW
AVONDALE
Diversey Ave
LOGAN SQUARE
W Armitage Ave
Peggy Notebaert
Nature Museum
LINCOLN PARK
Lincoln Park Zoo
Chicago History Museum
OLD TOWN
HUMBOLDT PARK
Augusta Blvd
Garfield Park
Conservatory
WEST TOWN
RIVER NORTH
Navy Pier
Chicago French Market
NEAR WEST SIDE
Chicago
Grant Park
Shedd Aquarium
Adler Planetarium
Roosevelt Rd
W 16th St
Addams/
Medill Park
NEAR SOUTH SIDE
CHINATOWN
LOWER WEST SIDE
LITTLE VILLAGE
SOUTH LAWNDALE
W 31st St
DOUGLAS
MCKINLEY PARK
BRONZEVILLE
W Pershing Rd
BRIGHTON PARK
GRAND BOULEVARD
ARCHER HEIGHTS
W 47th St
BACK OF THE YARDS
W 51st St
Map data ©2018 Google

Graph distance

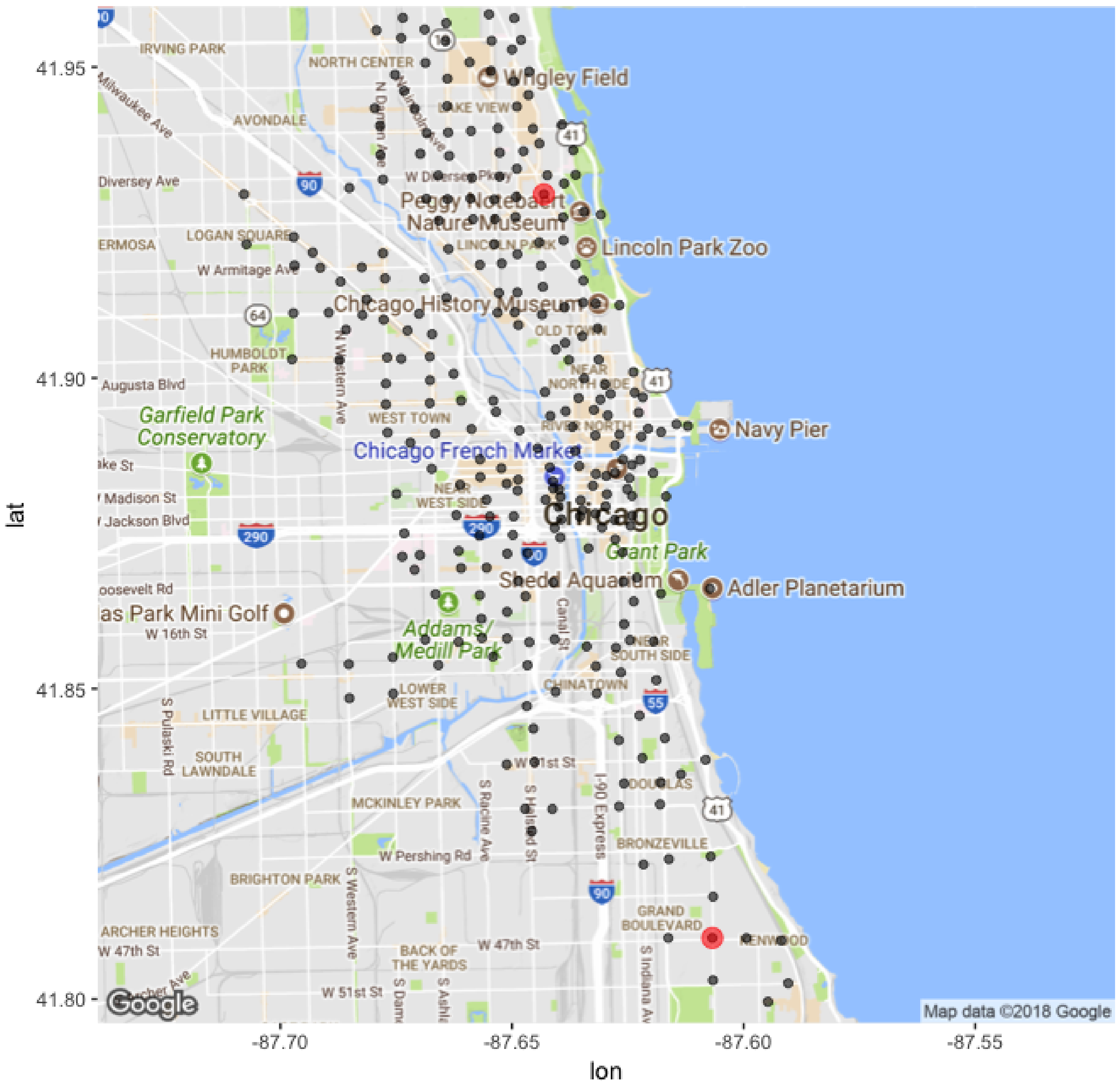
```
farthest_vertices(trip_g_simp)
```

```
$vertices
+ 2/300 vertices, named, from 20dcfff:
[1] 336 340

$distance
[1] 5
```

```
get_diameter(trip_g_simp)
```

```
+ 4/300 vertices, named, from 20dcfff:
[1] 336 267 76  340
```


41.95
41.90
41.85
41.80
lat
-87.70
-87.65
-87.60
-87.55
lon
IRVING PARK
NORTH CENTER
Wrigley Field
LAKE VIEW
AVONDALE
Diversey Ave
LOGAN SQUARE
Peggy Notebaert
Nature Museum
Lincoln Park Zoo
W Armitage Ave
Chicago History Museum
OLD TOWN
HUMBOLDT PARK
Augusta Blvd
NEAR NORTH SIDE
Garfield Park
Conservatory
WEST TOWN
RIVER NORTH
Navy Pier
Chicago French Market
Madison St
Jackson Blvd
NEAR WEST SIDE
Chicago
Grant Park
Shedd Aquarium
Adler Planetarium
Roosevelt Rd
W 16th St
Addams/
Medill Park
NEAR SOUTH SIDE
CHINATOWN
LOWER WEST SIDE
LITTLE VILLAGE
SOUTH LAWNDALE
W 31st St
DOUGLAS
MCKINLEY PARK
W Pershing Rd
BRONZEVILLE
BRIGHTON PARK
GRAND BOULEVARD
ARCHER HEIGHTS
W 47th St
BACK OF THE YARDS
KENWOOD
W 51st St
Google
Map data ©2018 Google

Geographic distance

```
library(geosphere)
# Get the to stations coordinates
st_to <- bike_dat %>%
            filter(from_station_id == 336) %>%
            sample_n(1) %>%
            select(from_longitude, from_latitude)
# Get the from stations coordinates
st_from <- bike_dat %>%
            filter(from_station_id == 340) %>%
            sample_n(1) %>%
            select(from_longitude, from_latitude)
# find the geographic distance
farthest_dist <- distm(st_from, st_to, fun = distHaversine)
farthest_dist
```

```
[1, ] 13660.66
```

Geographic distance

```
bike_dist <- function(station_1, station_2, divy_bike_df){
    st1 <- divy_bike_df %>%
                filter(from_station_id == station_1) %>%
                sample_n(1) %>%
                select(from_longitude, from_latitude)
    st2 <- divy_bike_df %>%
                filter(from_station_id == station_2) %>%
                sample_n(1) %>%
                select(from_longitude, from_latitude)

    farthest_dist <- distm(st1, st2, fun = distHaversine)
    return(farthest_dist)
}
```

Let's practice!

CASE STUDIES: NETWORK ANALYSIS IN R

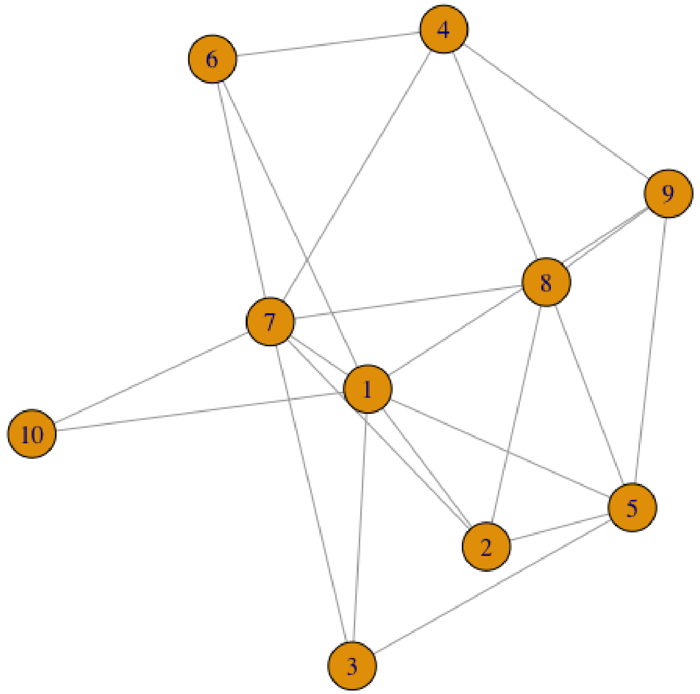
Connectivity

CASE STUDIES: NETWORK ANALYSIS IN R

Edmund Hart
Instructor

Measuring connectivity

```
rand_g <- erdos.renyi.game(10, 0.4, "gnp", directed = FALSE)
plot(rand_g)
```



Measuring connectivity

```
rand_g <- erdos.renyi.game(10, 0.4, "gnp", directed = FALSE)
vertex_connectivity(rand_g)
```

```
2
```

```
edge_connectivity(rand_g)
```

```
2
```

Minimum number of cuts

```
min_cut(rand_g, value.only = FALSE)
```

```
$value
[1] 2

$cut
+ 2/18 edges from 17a8fad:
[1] 10--7 10--1

$partition1
+ 1/10 vertex, from 17a8fad:
[1] 10

$partition2
+ 9/10 vertices, from 17a8fad:
[1]  1 2 3  4  5  6  7  8  9
```

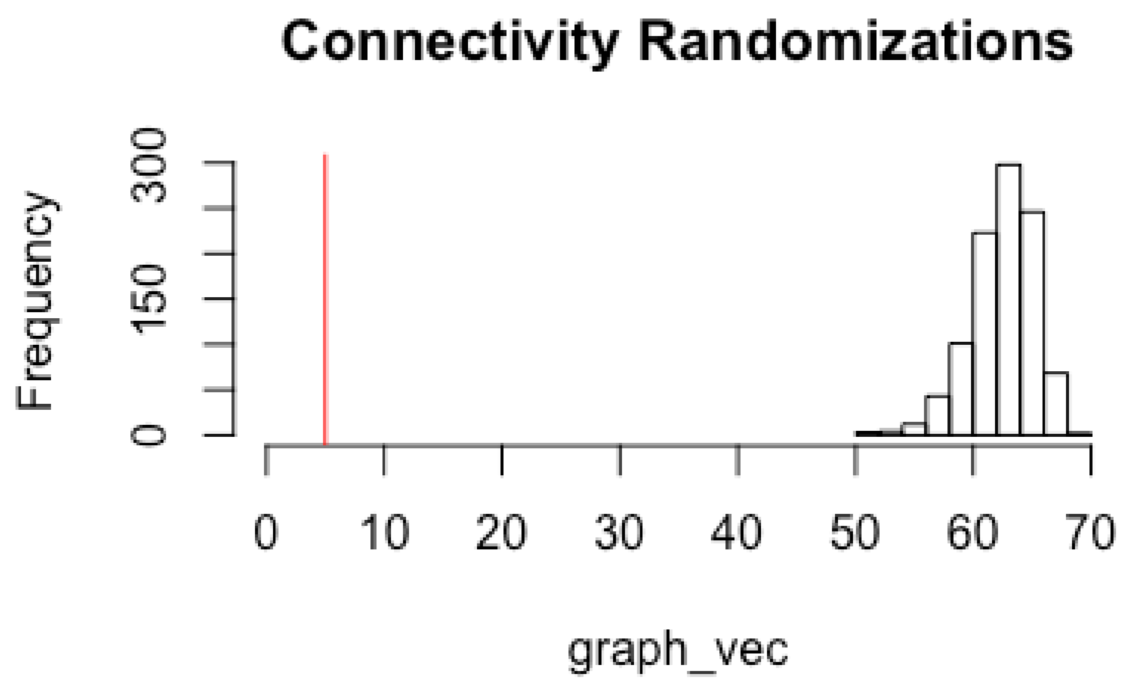
Connectivity randomizations

```r
# Get parameters to simulate graph
nv <- gorder(trip_g_ud)
ed <- edge_density(trip_g_ud)

# Empty vector to store output
graph_vec <- rep(NA, 1000)
# Generate 1000 random graphs and find the edge connectivity
for(i in 1:1000) {
  w1 <- erdos.renyi.game(nv, ed, "gnp", directed = TRUE)
  graph_vec[i] <- edge_connectivity(w1)
}
```

Connectivity randomizations

```
# Find actual connectivity
econn <- edge_connectivity(trip_g_ud)
hist(graph_vec, xlim = c(0, 140))
abline(v = edge_connectivity(trip_g_ud))
```



Let's practice!

CASE STUDIES: NETWORK ANALYSIS IN R