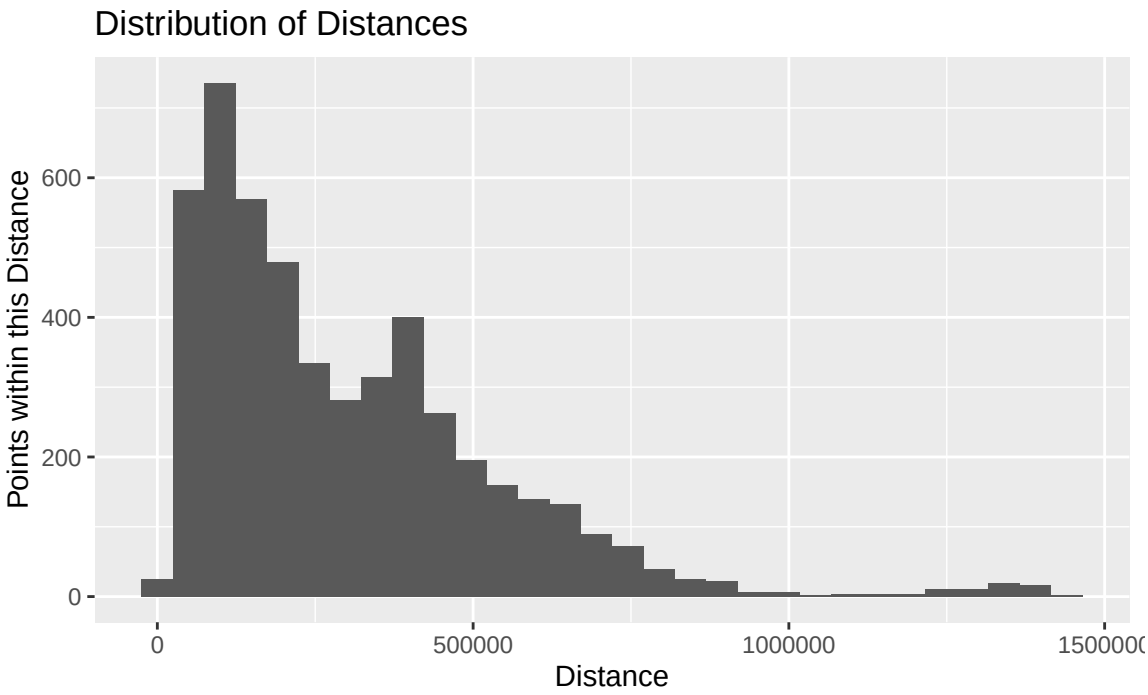
Lending Club Mapper

```
##### Packages ####
library(readr)
library(tidyverse)
library(igraph)
library(RColorBrewer)
library(FredsVietorisRips)
```

```
##### Import Data ####
df_0 <- read_csv("D:/lendingclubQ12018.csv")

##### Numeric Columns Only ####
df <- df_0 %>%
  lapply(is.numeric) %>%
  unlist()

##### Sample and Merge ####
sample_size <- 100
df <- df_0[df== TRUE] %>%
  sample_n(sample_size)
```

```
##### Distance Matrix ####
# Euclidean distance, first 11 columns
# Not including the "Quality" indicator
d <- df %>%
  dist() %>%
  as.matrix()
```

```
##### Tidy Data ####
d_tidy <- TidyDistanceFrame(d)

##### Distance Distribution ####
d_tidy %>%
  ggplot(aes(Distance)) +
  geom_histogram(bins = 30) +
  ylab("Points within this Distance") +
  ggtitle("Distribution of Distances")
```



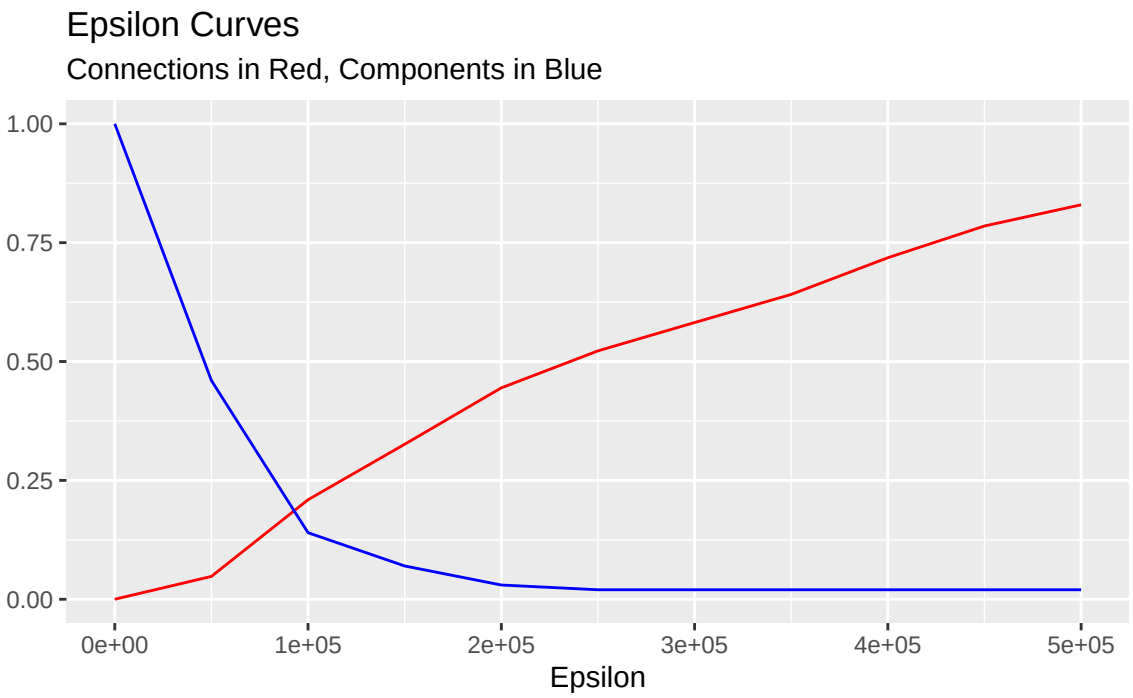

```
##### Create our "Epsilon Frame" ####
# From a vector of epsilon values
# we determine
# 1. The proportion of connections made
# 2. The number of components present
eframe <- CreateEpsilonFrame(d, seq(0, 500000, by = 50000))

##### Print Fancy Table ####
eframe %>%
  kable(caption = "Epsilon Frame")
```

Table 1: Epsilon Frame

Epsilon	Connections	Components
0	0.0000000	100
50000	0.0480808	46
100000	0.2092929	14
150000	0.3260606	7
200000	0.4446465	3
250000	0.5224242	2
300000	0.5820202	2
350000	0.6412121	2
400000	0.7183838	2
450000	0.7850505	2
500000	0.8296970	2

```
##### Visualize Curves ####
eframe %>%
```

```
  mutate(Components = Components / 100) %>%
  ggplot(aes(x = Epsilon)) +
  geom_line(aes(y = Connections), color = "red") +
  geom_line(aes(y = Components), color = "blue") +
  ylab(NULL) +
  ggtitle("Epsilon Curves",
          "Connections in Red, Components in Blue")
```



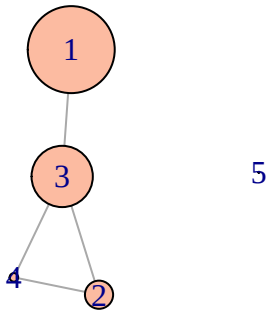

```
##### Open Covers ####
# Distance matrix -> Adjacency Matrix -> Open Covers
epsilon <- 350000
covers <- d %>%
  AdjacencyMatrix(epsilon) %>%
  OpenCoverEballs()

##### Adjacent Covers ####
# Create a graph from an
# Adjacency matrix of our covers
covers.plot <- covers %>%
  CoverAdjacencies() %>%
  graph_from_adjacency_matrix(mode="undirected")
```

```
##### Node Features ####
df <- df %>%
  left_join(df_0)

##### Coloring ####
redscale <- brewer.pal(8, "Reds")
df <- df %>% mutate( grade =
  case_when( grade =
```

```
    grade == "A" ~ 1,
    grade == "B" ~ 2,
    grade == "C" ~ 3,
    grade == "D" ~ 5,
    grade == "E" ~ 6,
    grade == "F" ~ 7,
    grade == "G" ~ 8
  ))

covers.color <- vector()
for (cover in 1:length(covers)) {
  covers.color[cover] <- round( mean( df$grade[ covers[[ cover ]] ] ) )
  cat("Average Grade for Node ", cover, ": ", round( mean( df$grade[ covers[[ cover ]] ] ) ), "\n")
}
```

```
## Average Grade for Node  1 :  3
## Average Grade for Node  2 :  3
## Average Grade for Node  3 :  3
## Average Grade for Node  4 :  3
## Average Grade for Node  5 :  3
```

```
V(covers.plot)$color <- redscale[covers.color]

##### Sizing ####
covers.size <- covers %>%
  lapply(length) %>%
  unlist()
V(covers.plot)$size <- covers.size

##### Plot ####
plot(covers.plot)
```


1
3
5
4
2