Datos de Cañadas

Emanuel Becerra Soto

Noviembre 19 2018

Cañadas Basura

```
set.seed(389)

file <- 'encuestas_v4.csv'
canadas <- read_csv(file)
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   genero = col_character(),
##   rango_de_edad = col_character(),
##   colonia = col_character(),
##   lugar_donde_vive_es = col_character(),
##   frecuencia_de_visita = col_character(),
##   como_se_siente = col_character(),
##   actividad_otro = col_character(),
##   estado_de_la_banquetas = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
can_binary <- canadas [ sapply(canadas,class) == 'integer' ]
can_mult <- canadas [ sapply(canadas,class) == 'character' ]

canadas_char <- canadas %>%
  mutate_all(as.character)
```

Ánalisis de los datos de la colonia Cañadas, para evaluar la hipótesis de que hay un problema de basura por la zona.

Los datos fueron obtenidos de los vecinos del lugar los cuales emitieron su opinión en Octubre 2018.

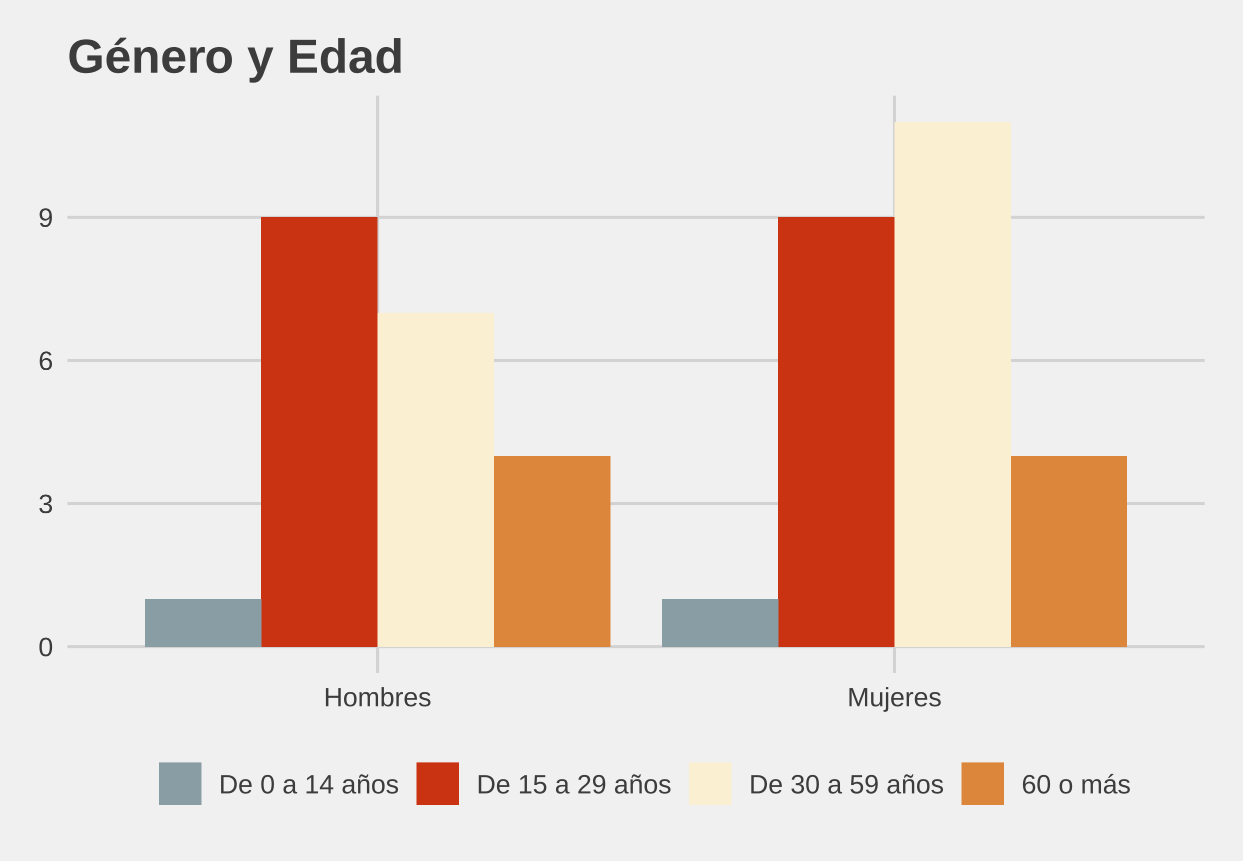
Tamaños de los datos: 46 renglones, columnas 46

Nombre de las columnas

```
names(canadas)
```

```
##  [1] "genero"
##  [2] "rango_de_edad"
##  [3] "colonia"
##  [4] "lugar_donde_vive_es"
##  [5] "frecuencia_de_visita"
##  [6] "no_frecuenta_por_basura"
##  [7] "no_frecuenta_por_robos"
##  [8] "no_frecuenta_por_lugares_solos"
##  [9] "no_frecuenta_por_drogas"
## [10] "no_frecuenta_por_violencia"
## [11] "no_frecuenta_por_vandalismo"
## [12] "como_se_siente"
## [13] "actividad_deporte"
## [14] "actividad_leer"
## [15] "actividad_juegos_de_mesa"
## [16] "actividad_ejercicio"
## [17] "actividad_jugar"
## [18] "actividad_otro"
## [19] "horario_de_visita_mananas"
## [20] "horario_de_visita_medio_dia"
## [21] "horario_de_visita_tarde"
## [22] "horario_de_visita_noche"
## [23] "estado_de_la_banquetas"
## [24] "elemento_deseado_iluminacion"
## [25] "elemento_deseado_wi_fi"
## [26] "elemento_desesado_banos"
## [27] "elemento_deseado_juegos_infantiles"
## [28] "elemento_deseado_bancas"
## [29] "elemento_deseado_aparatos_ejercitadores"
## [30] "elemento_deseado_arboles"
## [31] "elemento_deseado_mesas"
## [32] "elemento_deseado_canchas"
## [33] "elemento_desdeado_cestos_de_basura"
## [34] "elemento_deseado_area_para_mascotas"
## [35] "elemento_deseado_otro"
## [36] "actividad_deseada_cine_al_aire_libre_"
## [37] "actividad_deseada_danza"
## [38] "actividad_desedada_teatro"
## [39] "actividad_deseada_circulos_de_lectura"
## [40] "actividad_deseada_jugar"
## [41] "actividad_deseada_deporte"
## [42] "actividad_deseada_conciertos_de_musica"
## [43] "actividad_deseda_ejercicio"
## [44] "actividad_deseada_juegos_de_mesa"
## [45] "actividad_deseada_comer"
## [46] "actividad_deseada_artes_plasicas"
## [47] "actividad_deseada_otro"
```

```r
ggplot(canadas, aes(x = genero, fill = rango_de_edad))+
  geom_bar(position='dodge')+
  ggtitle('Género y Edad')+
  scale_x_discrete(labels=c('Hombres','Mujeres'))+
  scale_fill_manual(values=wes_palette("Royal1"),
                    name=NULL,
                    labels=c("De 0 a 14 años", "De 15 a 29 años",
                             "De 30 a 59 años", "60 o más"))+
  theme_fivethirtyeight()
```



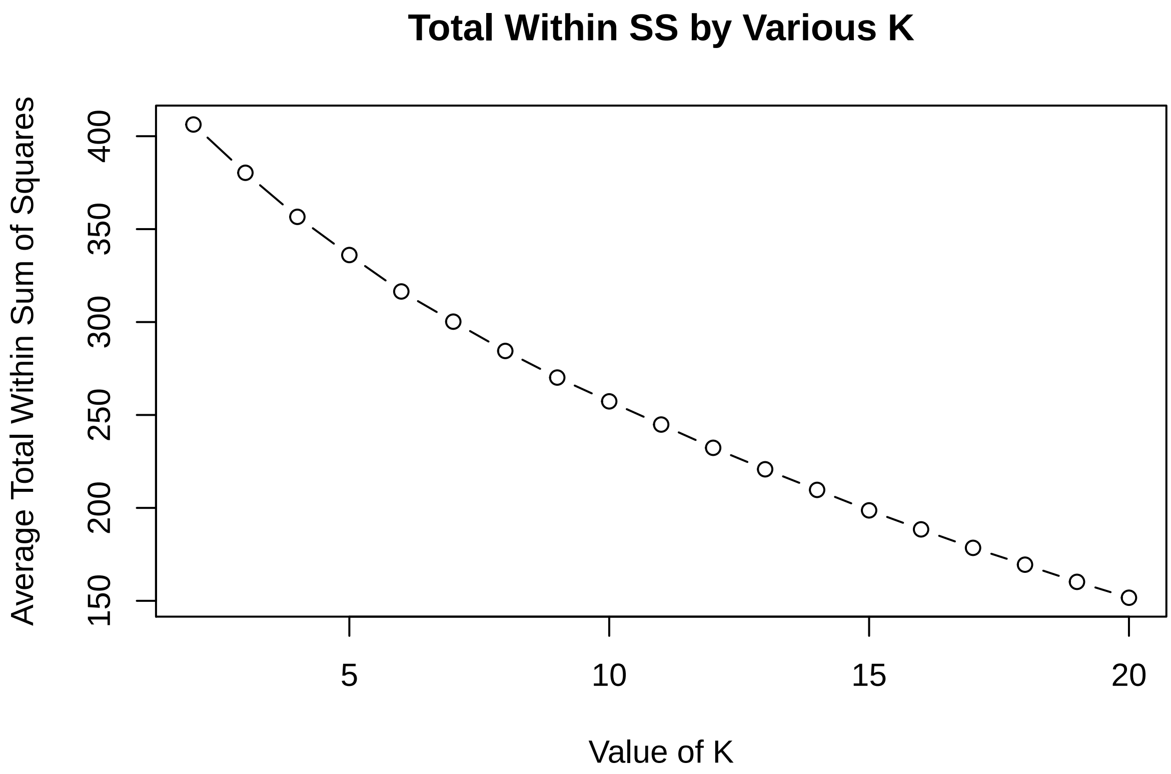
Gráfica de barras con la edad y el género de las personas entrevistadas.

```
###### One-Hot Encoding ######

numeric_feats <- names(can_binary)
categorical_feats <- names(can_mult)

dummies <- dummyVars( ~. , canadas[categorical_feats] )
cat_1_hot <- predict( dummies, canadas[ categorical_feats ] )

canadas_one <- as.tibble( cbind( canadas[ numeric_feats ], cat_1_hot ) )
```

```
# K-means

rng<-2:20 #K from 2 to 20
tries <-100 #Run the K Means algorithm 100 times
avg.totw.ss <-integer(length(rng)) #Set up an empty vector to hold all of points
for(v in rng){ # For each value of the range variable
 v.totw.ss <-integer(tries) #Set up an empty vector to hold the 100 tries
 for(i in 1:tries){
 k.temp <-kmeans(canadas_one,centers=v) #Run kmeans
 v.totw.ss[i] <-k.temp$tot.withinss#Store the total withinss
 }
 avg.totw.ss[v-1] <-mean(v.totw.ss) #Average the 100 total withinss
}
```

```
## Warning: did not converge in 10 iterations
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
```

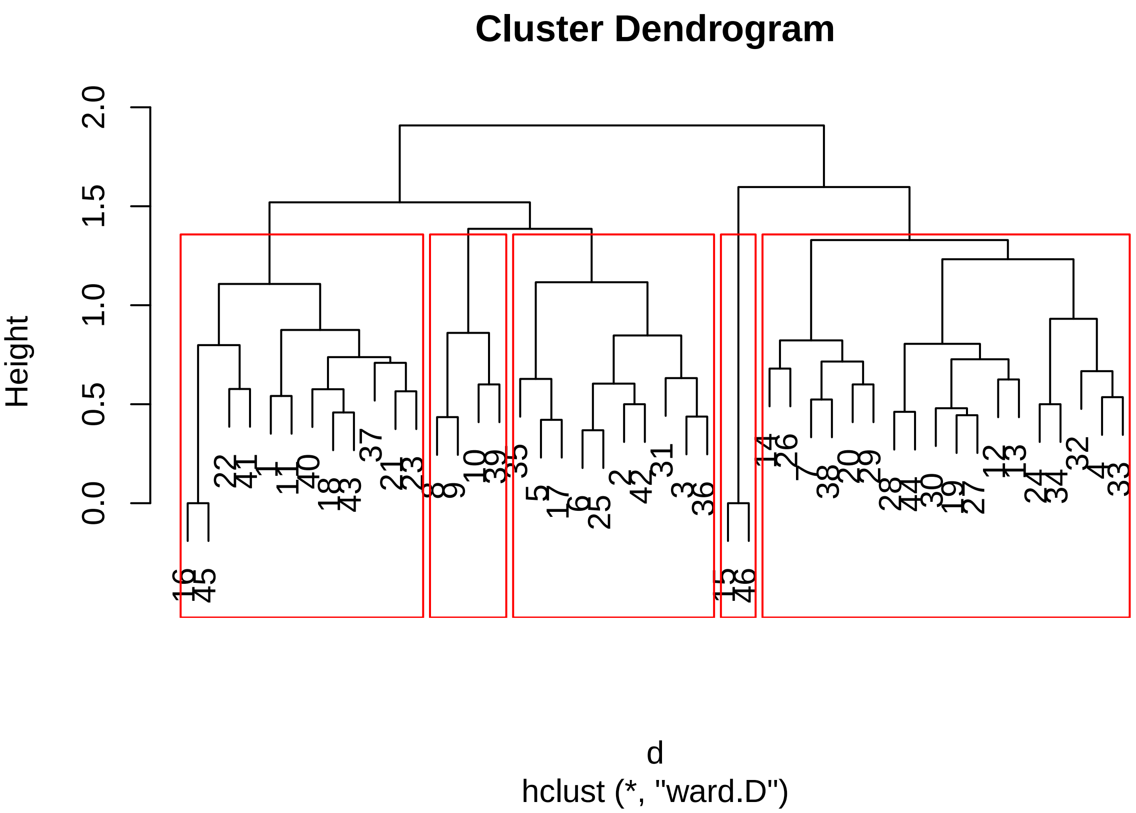
```r
plot(rng,avg.totw.ss,type="b", main="Total Within SS by Various K",
 ylab="Average Total Within Sum of Squares",
 xlab="Value of K")
```



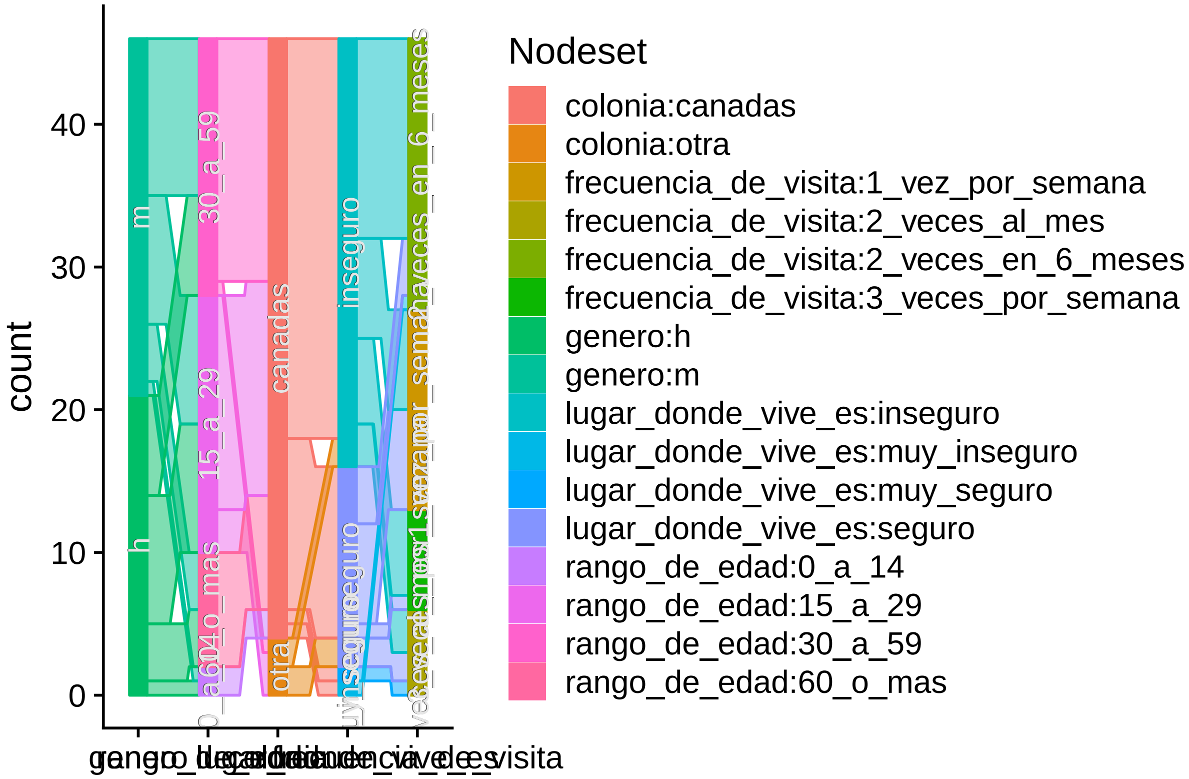
```r
# Ward Hierarchical Clustering

d <- dist(canadas_one, method = "binary") # distance matrix
fit <- hclust(d, method="ward")
```

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```r
plot(fit) # display dendogram
groups <- cutree(fit, k=5) # cut tree into 5 clusters
# draw dendogram with red borders around the 5 clusters
rect.hclust(fit, k=5, border="red")
```



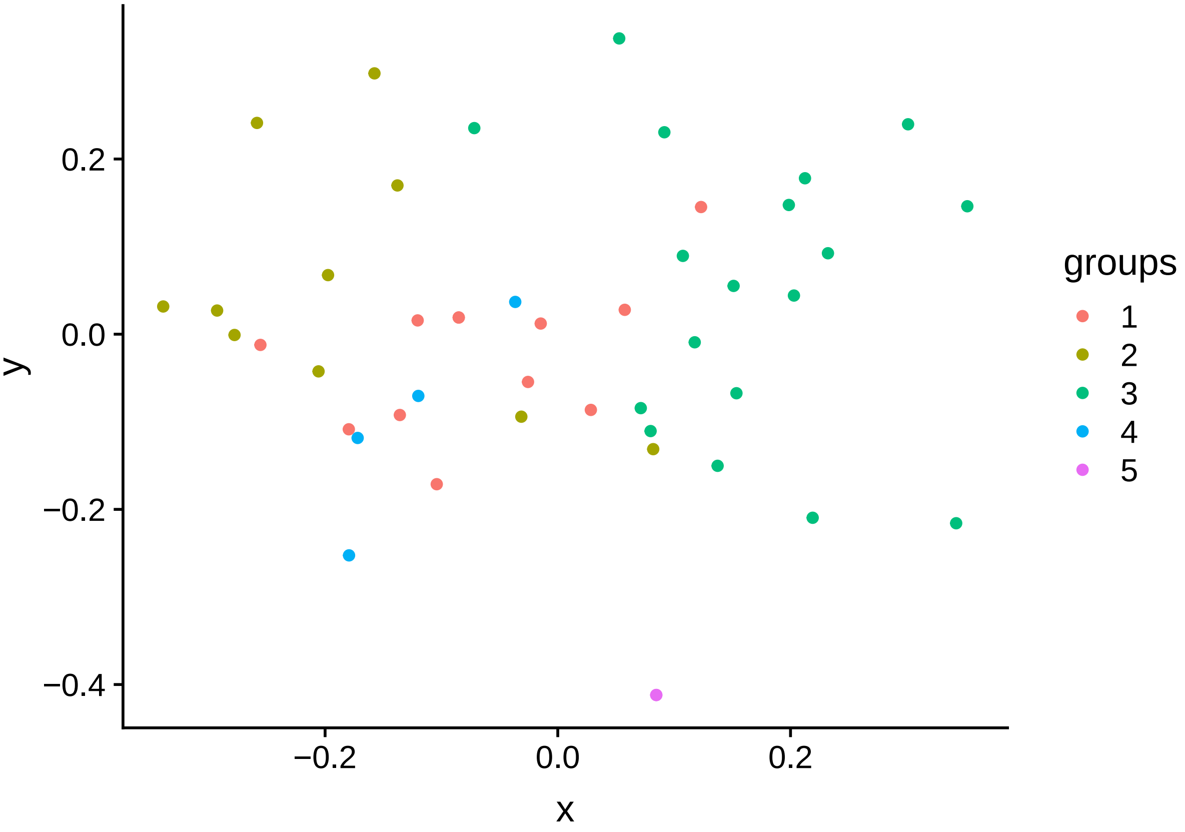
```r
ggparallel( vars = as.list(names(can_mult)[1:5]) , data=as.data.frame(can_mult) )
```



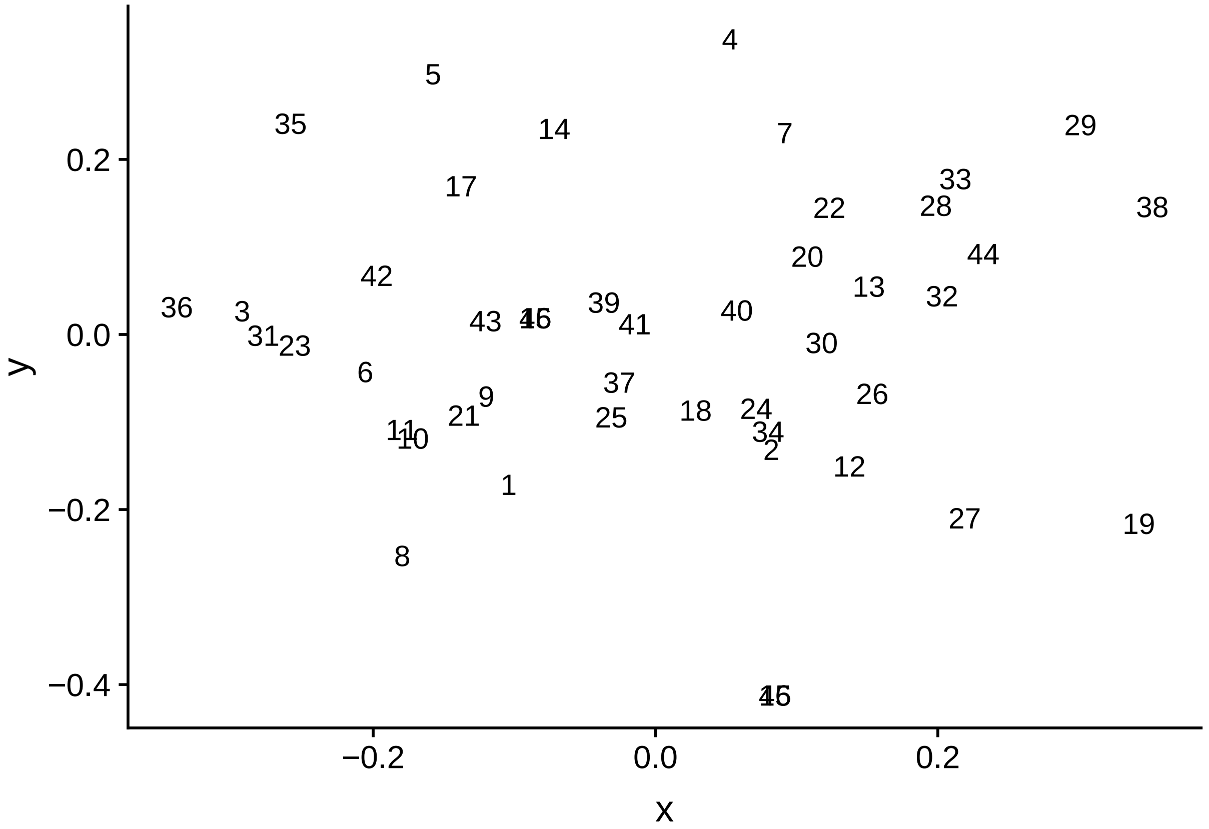
```r
#ggsave('par_01.svg', units = 'cm', width = 20, height = 14)
```

```
#ggparallel( vars = as.list(names(can_mult)[1:5]) , data=as.data.frame(can_mult), method = 'hammock' )
#ggsave('par_02.svg', units = 'cm', width = 20, height = 14)

#ggparallel( vars = as.list(names(can_mult)[1:5]) , data=as.data.frame(can_mult), method = 'parset' )
```

```
##PCO using cmdscale function
#K equal to number of dimensions to return

bin_distance <- dist(canadas_one, method = 'binary')
k <- 3
pcoa <- cmdscale(bin_distance, k=k, eig=T)

points <- pcoa$points
eig <- pcoa$eig
points <- as_tibble(points)
colnames(points) <- c("x", "y", "z")

points$groups <- as.character(cutree(fit, k=5))

ggplot(points, aes(x = x, y =y))+
  geom_point( aes(color=groups) )
```



```
ggplot(points, aes(x = x, y =y))+
  geom_text( aes(label=rownames(points) ))
```

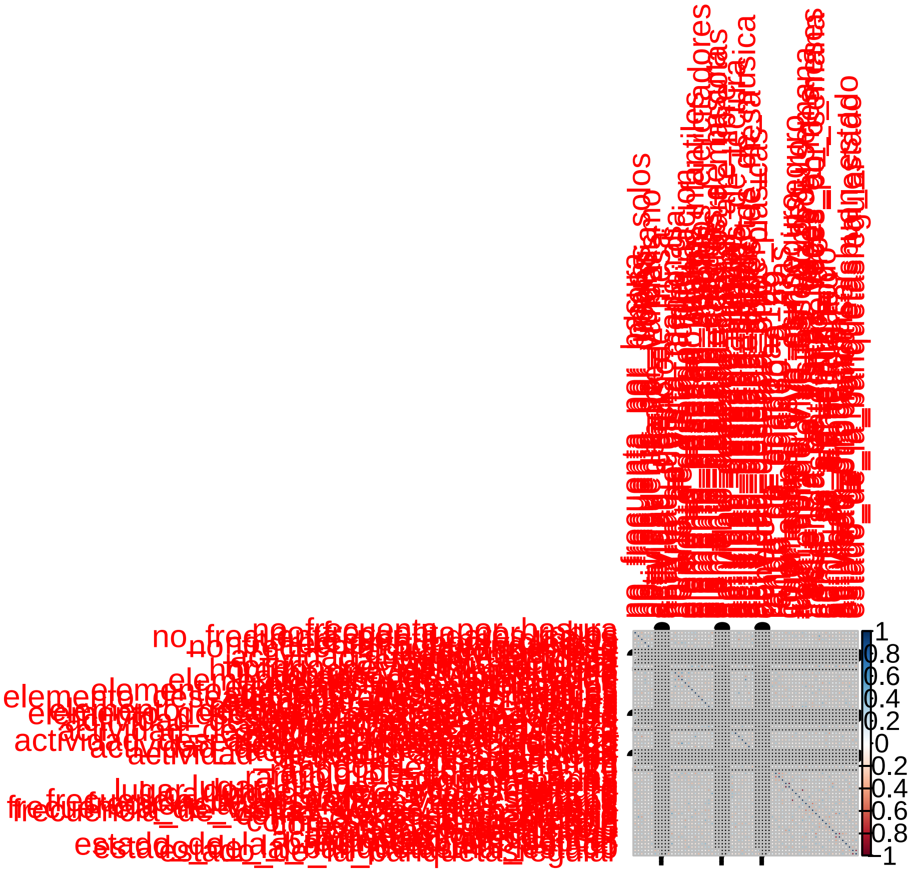


```
# ggplot(points, aes(x = x, y =z))+
#   geom_point()
#
# ggplot(points, aes(x = y, y =z))+
#   geom_point()
```

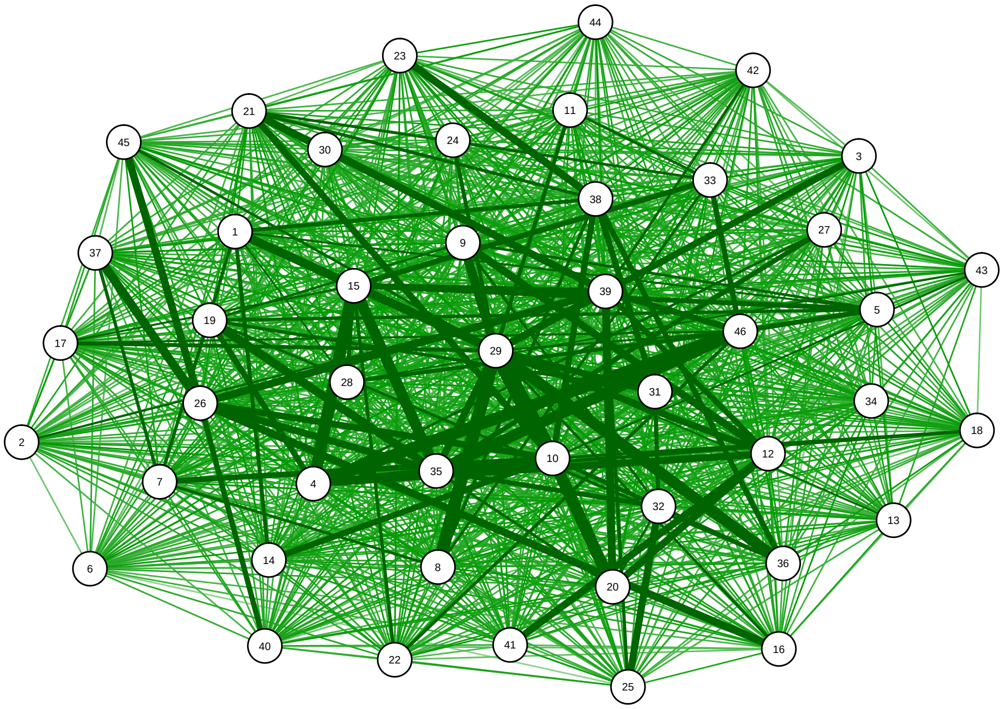
```
#jpeg('cor_can.jpg', width=3000, height=3000, unit='px')
M <- cor(as.matrix(canadas_one))
```

```
## Warning in cor(as.matrix(canadas_one)): the standard deviation is zero
```

```
corrplot(M, method = "circle")
```



```
#dev.off()

dist_mi <- dist(canadas_one, method = 'binary')
#jpeg('distance_people.jpg', width=3000, height=3000, unit='px')
qgraph(dist_mi, layout='spring', vsize=3)
```



```
#dev.off()
```

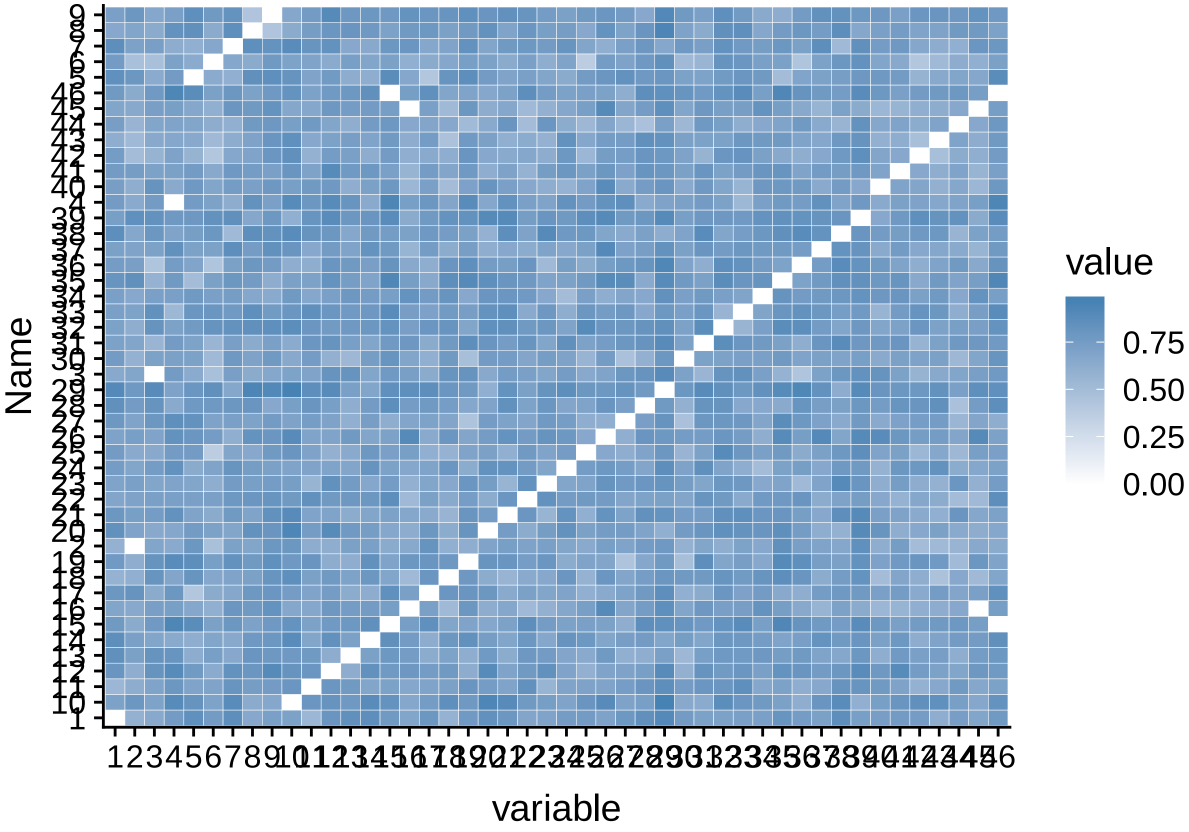
```
tib_dist <- as.tibble(as.matrix( (dist_mi) ) )

n <- length(tib_dist)
tib_dist$Name <- as.character(1:n)
n <- length(tib_dist)
tib_dist[ c( n, 2:n-1) ]
```

```
## # A tibble: 46 x 47
##    Name   `1`   `2`   `3`   `4`   `5`   `6`   `7`   `8`   `9`  `10`  `11`
##    <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 1     0     0.583 0.652 0.759 0.852 0.76  0.862 0.654 0.724 0.708 0.542
##  2 2     0.583 0     0.682 0.64  0.792 0.474 0.708 0.68  0.793 0.792 0.625
##  3 3     0.652 0.682 0     0.76  0.632 0.471 0.727 0.636 0.667 0.7   0.696
##  4 4     0.759 0.64  0.76  0     0.76  0.708 0.625 0.833 0.724 0.893 0.793
##  5 5     0.852 0.792 0.632 0.76  0     0.632 0.6   0.846 0.857 0.818 0.696
##  6 6     0.76  0.474 0.471 0.708 0.632 0     0.667 0.636 0.769 0.7   0.696
##  7 7     0.862 0.708 0.727 0.625 0.6   0.667 0     0.857 0.828 0.88  0.815
##  8 8     0.654 0.68  0.636 0.833 0.846 0.636 0.857 0     0.435 0.636 0.741
##  9 9     0.724 0.793 0.667 0.724 0.857 0.769 0.828 0.435 0     0.667 0.759
## 10 10    0.708 0.792 0.7   0.893 0.818 0.7   0.88  0.636 0.667 0     0.8
## # ... with 36 more rows, and 35 more variables: `12` <dbl>, `13` <dbl>,
## #   `14` <dbl>, `15` <dbl>, `16` <dbl>, `17` <dbl>, `18` <dbl>,
## #   `19` <dbl>, `20` <dbl>, `21` <dbl>, `22` <dbl>, `23` <dbl>,
## #   `24` <dbl>, `25` <dbl>, `26` <dbl>, `27` <dbl>, `28` <dbl>,
## #   `29` <dbl>, `30` <dbl>, `31` <dbl>, `32` <dbl>, `33` <dbl>,
## #   `34` <dbl>, `35` <dbl>, `36` <dbl>, `37` <dbl>, `38` <dbl>,
## #   `39` <dbl>, `40` <dbl>, `41` <dbl>, `42` <dbl>, `43` <dbl>,
## #   `44` <dbl>, `45` <dbl>, `46` <dbl>
```

```
m.tib_dist <- melt(tib_dist)
```

```
## Using Name as id variables
```

```
ggplot(m.tib_dist, aes(variable, Name)) +
  geom_tile(aes(fill = value),colour = "white") +
  scale_fill_gradient(low = "white", high = "steelblue")
```



```
#ggsave('dist_heat.svg',units = 'cm', width = 18, height = 20)
```