Self Organising Maps

1829344

05/08/2021

1. Loading packages required for Data Pre-processing

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(fpc)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(kohonen)
```

```
##
## Attaching package: 'kohonen'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

2. Loading all datasets for the project & Removing Unnecessary Columns

```r
# Loading dataset

cali = read.csv("fire.csv")

# Drop unncessary columns

cali$X = NULL
cali$FireSizekm2 = NULL
cali$Fire.size = NULL
```

3. Correlation Heatmap

```r
# Create Correlation map of variables

library(corrplot)
```

```
## corrplot 0.84 loaded
```

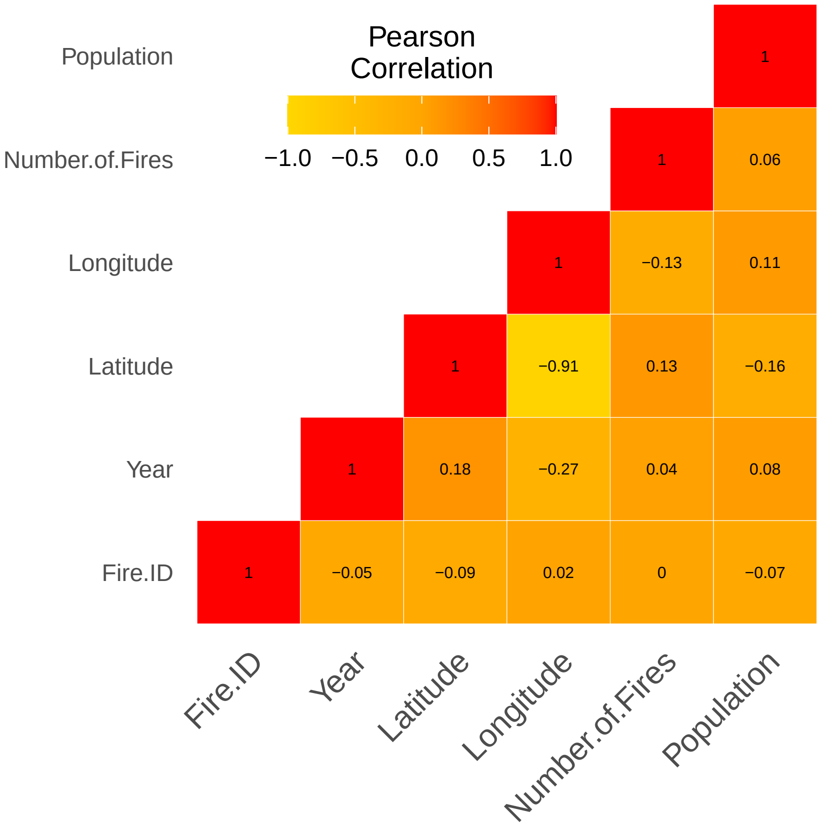
```r
corr.set <- select_if(cali,is.numeric)
corr.matrix = round(cor(corr.set),2)
corr.matrix
```

```
##                 Fire.ID  Year Latitude Longitude Number.of.Fires Population
## Fire.ID            1.00 -0.05    -0.09      0.02            0.00      -0.07
## Year              -0.05  1.00     0.18     -0.27            0.04       0.08
## Latitude          -0.09  0.18     1.00     -0.91            0.13      -0.16
## Longitude          0.02 -0.27    -0.91      1.00           -0.13       0.11
## Number.of.Fires    0.00  0.04     0.13     -0.13            1.00       0.06
## Population        -0.07  0.08    -0.16      0.11            0.06       1.00
```

```r
# Correlation Heat Map - Upper Triangle

get_upper_tri <- function(corr.matrix){
    corr.matrix[lower.tri(corr.matrix)]= NA
    return(corr.matrix)
}

# using upper triangle function to return the upper half of the correlation map

upper_tri <- get_upper_tri(corr.matrix)
upper_tri
```

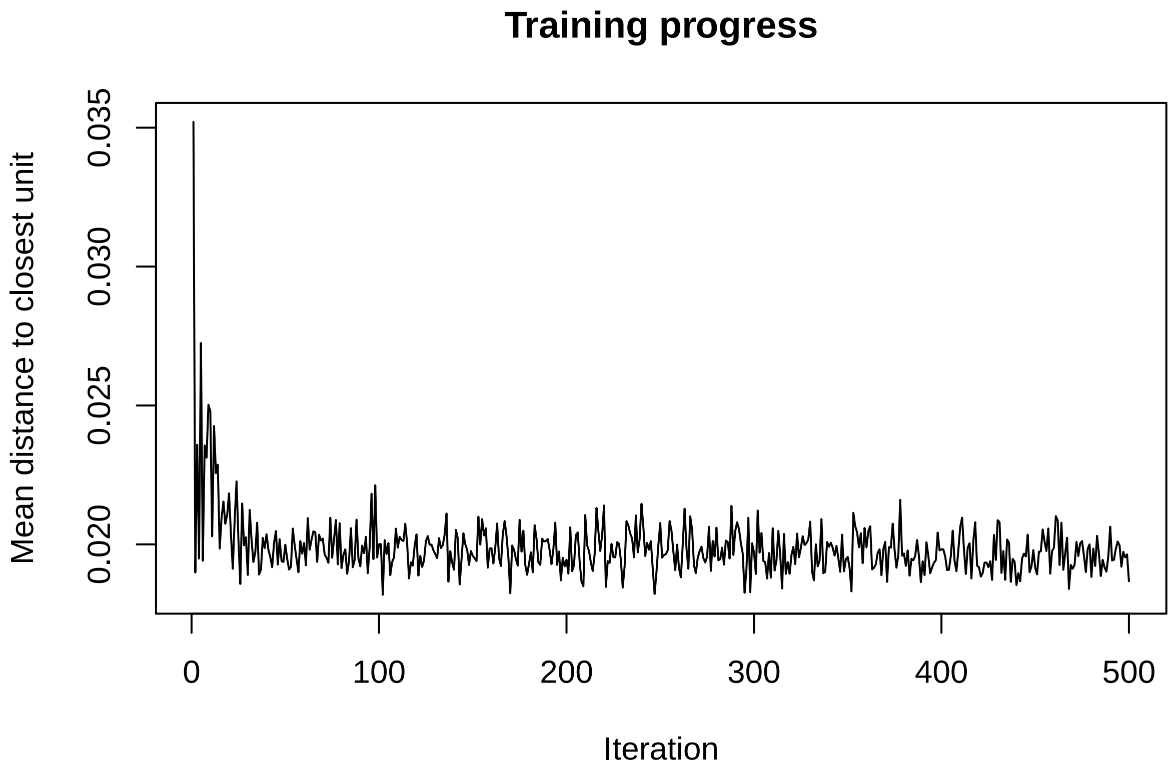
```
##                 Fire.ID  Year Latitude Longitude Number.of.Fires Population
## Fire.ID               1 -0.05    -0.09      0.02            0.00      -0.07
## Year                 NA  1.00     0.18     -0.27            0.04       0.08
## Latitude             NA    NA     1.00     -0.91            0.13      -0.16
## Longitude            NA    NA       NA      1.00           -0.13       0.11
## Number.of.Fires      NA    NA       NA        NA            1.00       0.06
## Population           NA    NA       NA        NA              NA       1.00
```

```r
# Correlation Heat Map - Upper Triangle

melt_cormat <- melt(upper_tri, na.rm = TRUE)

# Heatmap

ggheatmap = ggplot(data = melt_cormat, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "gold", high = "red", mid = "orange",
   midpoint = 0, limit = c(-1,1), space = "Lab",
   name="Pearson\nCorrelation") +
  theme_minimal()+
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 12, hjust = 1))+
 coord_fixed()

# Adding correlation coefficients on the heatmap

corelcoef = ggheatmap + geom_text(aes(Var2, Var1, label = value), color = "black", size = 2) +
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.position = c(0.6, 0.7),
  legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                title.position = "top", title.hjust = 0.5))

# Print the correlation coefficient

corelcoef
```

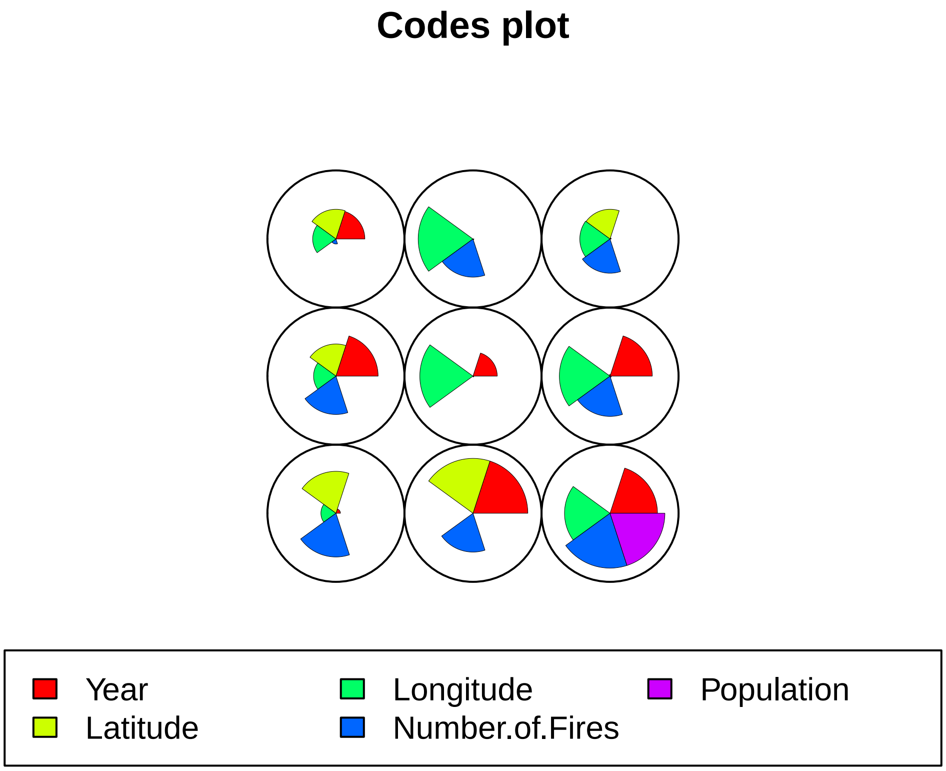


4. Trainning and Visualisation of Self Organising Maps

```
m= scale(corr.set[-1])

set.seed(111)

g = somgrid(xdim = 3,ydim =3, topo = "rectangular")

map = som(m,grid=g,alpha=c(0.05,0.01),
          radius=1, rlen=500)

f =plot(map,type ='changes')
```

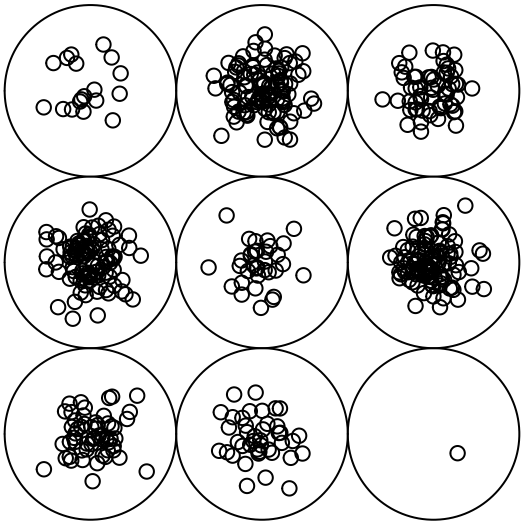


```
plot(map,
     type = "codes",
     palette.name = rainbow,
     )
```

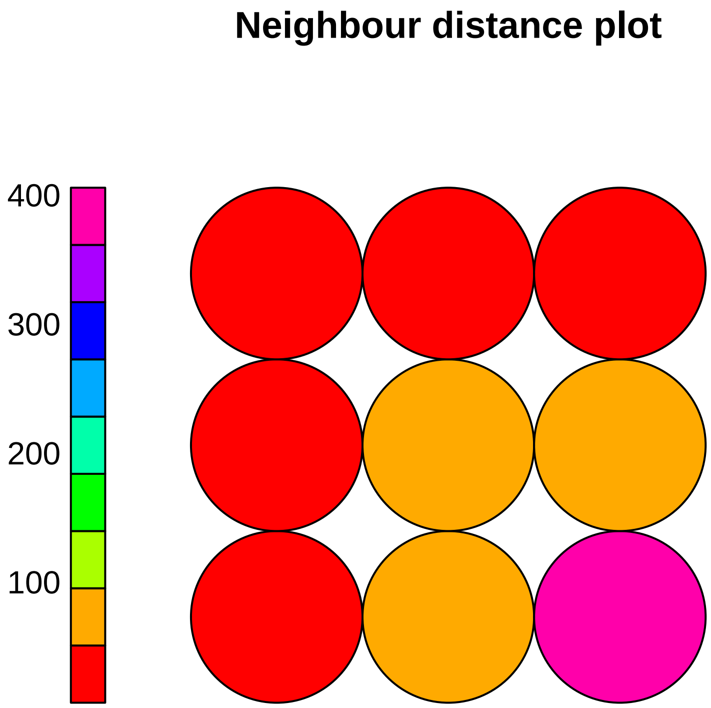


```
d = plot(map,
     type = "mapping",
     palette.name = rainbow,
     )
```

Mapping plot



```
k = plot(map,
     type = "dist.neighbours",
     palette.name = rainbow,
     )
```


Neighbour distance plot
400
300
200
100