Linear Statistical Models

Making a vector map stitching satellite screenshots

INDIAN STATISTICAL INSTITUTE

B.STAT 3'rd Year

AVINANDAN ROY

Roll-BS2023

Year-2022

Contents

1 Introduction

In this project we will make a vector map of an area and point out different localities which are important for that locality. Eventually it happen that in google map or in similar devices we have to zoom in and out to get different zones and in a very small part we can not contain all the important locations of a locality together. This project will help us to make such a Vector map so that every locations of interest will be available and we will also get the road map. Also this project will help us to discover many new ways in the locality that the locals generally omit.

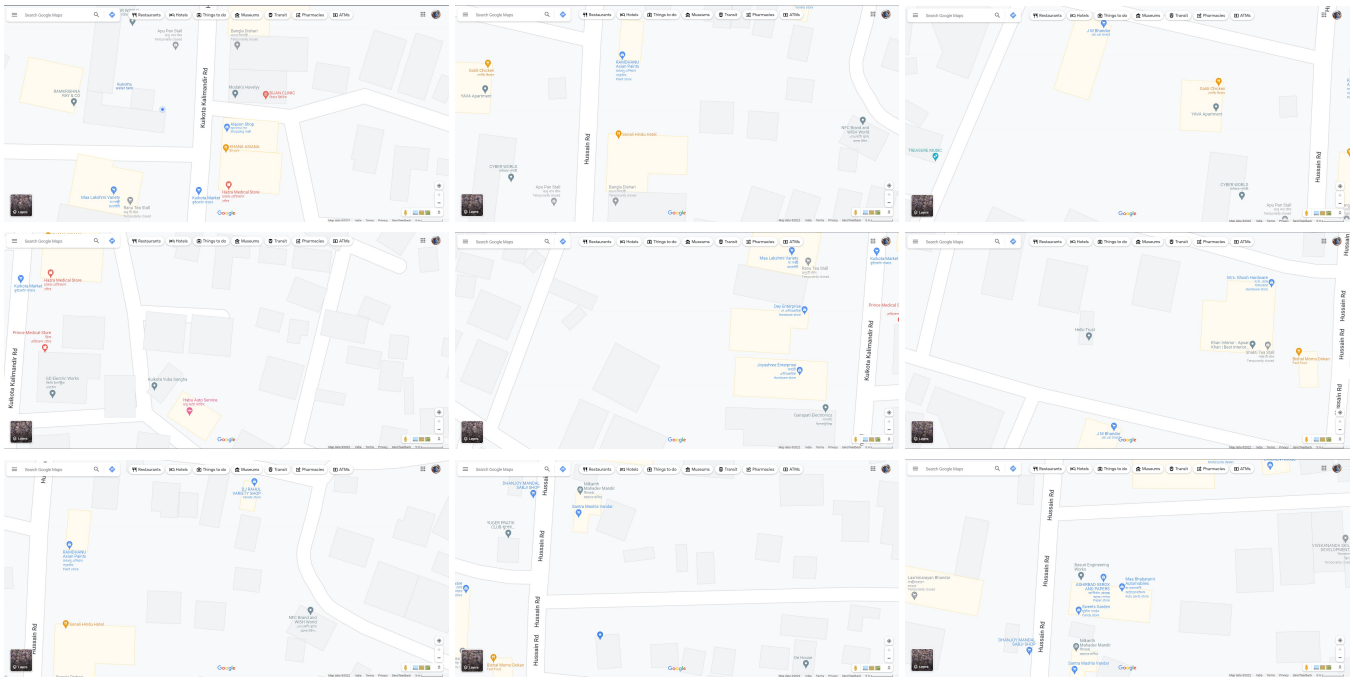
2 Brief Overview of the Project

In this project I had made a vector map of my locality in Midnapore where I live and the important places nearby me. The steps, I followed in making the Vector map are stated briefly below-

- First I took 9 screenshots from google map of my area such that every pair of screenshots share at least one place in common
- First I note down the important places in each of the screenshots and then I saved them in a .csv file
- With the help of R , I identified the places according to the .csv file in the screenshots and stored the locations in the screenshots in a local co-ordinate system
- Then, I have fitted a linear model to have the locations all-together in a single map such that I get a full map
- After getting the map, I added the roads connecting the places accordingly

3 Data Collection

I have collected my data using the software Google Map. I searched my location of hometown then took screenshots of various places in my locality. The screenshots are-



I have also shared my data set compiled with these screenshot in a folder.

4 Theory

In this section we are going to explain the Linear Model , used for estimating process.

4.1 The Model

For sake of Generality suppose we have total J number of screenshots. And let we have total n clicks of which I number of places are unique.For each click I stored the frame number(as ss variable), the name of the place(as place variable) and the co-ordinates of my click. I fitted linear model for the x-ordinate and the y-ordinate separately and for the error part we are taking Normality assumptions
For the $y-$ordinate the model is-

$$y_{ij} = place_i + ss_j + \epsilon_{ij} \quad \text{where } i = 1 \dots I; j = 1 \dots J$$

$$\epsilon_{ij} \sim N(0, \sigma^2)$$

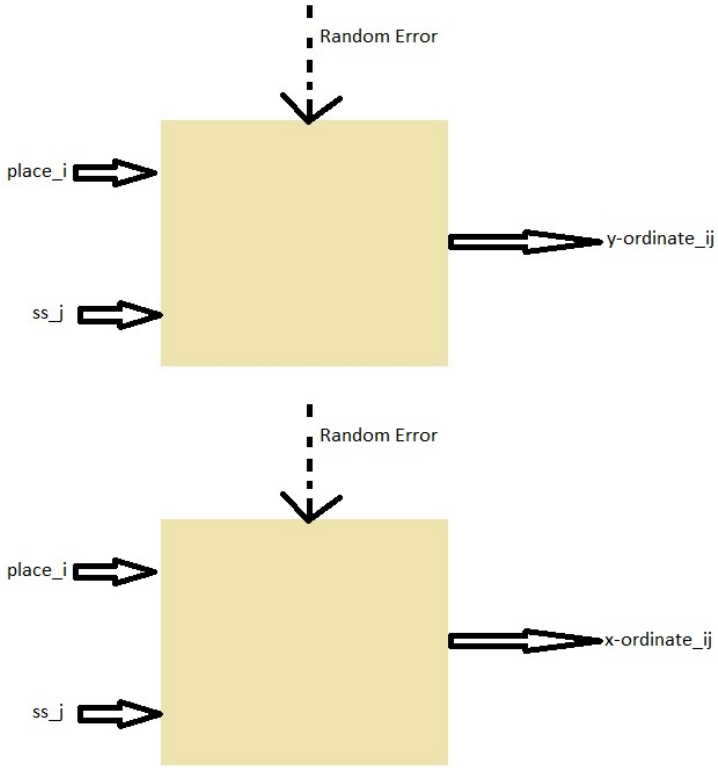
Note that both the inputs are factor. This is a 2-way ANOVA model. We have only two indices (i,j) as for each type(i,j) we have only one observation.
Similarly for the $x-$ordinate the model is-

$$x_{ij} = place_i + ss_j + \epsilon'_{ij} \quad \text{where } i = 1 \dots I; j = 1 \dots J$$

$$\epsilon'_{ij} \sim N(0, \sigma'^2)$$

4.2 Black Box diagram of the problem

The Black boxex we will use for the two models are-



4.3 Design Matrix

For each of the black boxes-

- Number of columns in design matrix
  =Number of arrows(except random error) going in
  $=I + J$
- Now,The Number of rows in the design matrix
  =Total number of clicks
  $=n$

Hence, the design matrices of these models will have dimension $= n \times (I + J)$
The entry corresponding to $place_i$ column

$$= \begin{cases} 1 & \text{if the current observation contains } place_i \\ 0 & \text{o.w} \end{cases}$$

The entry corresponding to ss_j column

$$= \begin{cases} 1 & \text{if the current observation contains } ss_j \\ 0 & \text{o.w} \end{cases}$$

4.4 Rank of the design Matrix

The map formed by the screenshots is connected iff rank of the design matrix is $= I + J - 1$.Where J =Number of screenshots and I =Number of distinct places. The idea behind this claim is the following-
We have Number of Clicks=Number of rows=n
Now,consider the screenshots as vertices of a graph and draw an edge between them of they have at least one important place in common. We know if a graph is connected there will exist at least $(J - 1)$ many edges.Now, if $place_i$ is common in k many of the screenshots, there exits at least $(k - 1)$ edges in the graph.
Hence,total number of edges in the graph is-

$$\sum_{i=1}^{I} \{k_i - 1\} \quad \text{Where } k_i = \text{Number of times } place_i \text{ has occurred}$$

Now,
$J - 1 \leq \sum_{i=1}^{I} \{k_i - 1\} \Rightarrow J - 1 \leq n - J \Rightarrow I + J - 1 \leq n$
If all the rows are independent $n = I + J - 1$
Now, for the columns, if we add all the columns w.r.t. place variable we get a vector of 1's. Similarly adding up all the columns w.r.t. ss variable we get a vector of 1's. All the ss variables are independent of each other and all the place variables are independent of each other. Hence Column rank of the design matrix$= I + J - 1$
Combining two cases we get,the map is connected iff rank of the design matrix is $= I + J - 1$.Using this I checked for faulty clicks from the users in my software later.

5 Working in R-The Vector Map

In this section we will explain how we will execute our work through the R-studio.The steps are described below-

5.1 Data Collection and Read Data in R

First we specify the directory where we have saved our screenshots and Also we read a csv file which we have prepared manually by knowing the actual names of the important places and we have listed below the places of our interest in the csv file accordingly our screenshots

```
setwd("D:/D/Avinandan Roy/Semister-5/Endsem/Projects/Linear Model/Final/Frames")
my_data=data.frame()
```

5.2 Necessary Libraries

The Libraries we will need are-

```
install.packages("jpeg")
install.packages("ggplot2")
install.packages("ggrepel")
install.packages("plotly")
library(jpeg)
library(ggplot2)
library(plotly)
library(ggrepel)
```

5.3 Making data frame for Our Model

By this code we will make our data for the places-

```
my_data=data.frame()
construct=function(data,name,k){
    n=length(name)
    map = readJPEG(paste("ss",k,".JPG", sep = "\ n"))
    d = dim(map)[1:2]
    cat("Choose these places in order from the map as shown below-")
    for(i in 1:n)
    {
      cat(i,'.',name[i],"\ n")
    }
    plot(NULL, xlim=c(0,d[2]), ylim=c(0,d[1]), ty='n', xlab="x", ylab="y", asp=1.4)
    rasterImage(as.raster(map),0,0,d[2],d[1])
    pointer = locator(n)
    temp=data.frame(k,name,pointer$x,pointer$y,d[2],d[1])
    names(temp) = c("ss", "place", "x", "y","xlim","ylim")
    temp2=rbind(data,temp)
    my_data=temp2
    return(my_data)
}
for(k in 1:9)
{
  my_data=construct(my_data,na.omit(Name_data[,k]),k)
}
write.csv(my_data,file.choose())
```

5.4 Making of the Plot

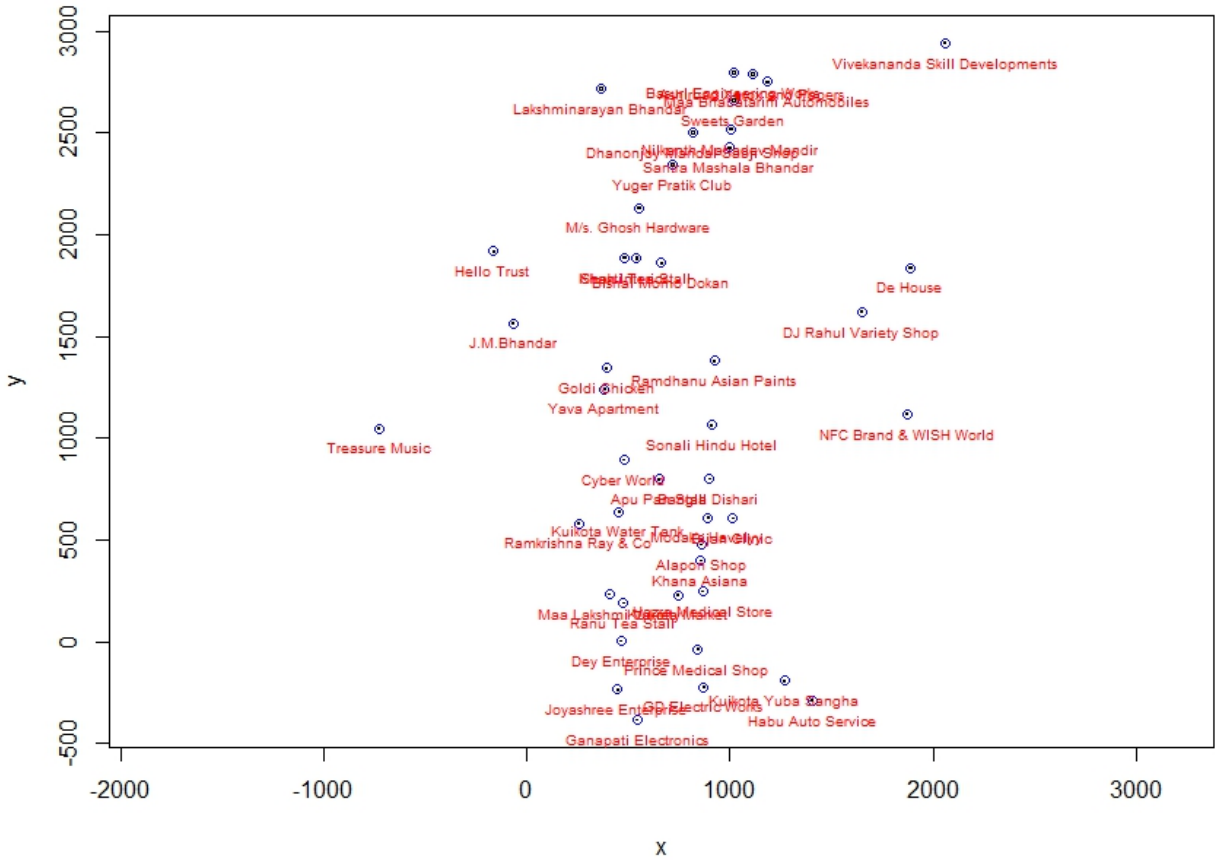
Now we will construct a function in R to make our fitted possible global plot-

```
clicks=read.csv(file.choose())

my_plot<-function(clicks)
```

```
{
  fitx = lm(x~place+ss-1, clicks)
  fity = lm(y~place+ss-1, clicks)
  m = length(unique(clicks$place))
  n = length(unique(clicks$ss))
  x_ordinate = fitx$coef[1:m]
  xerr = summary(fitx)$coef[1:m,2]
  y_ordinate = fity$coef[1:m]
  yerr = summary(fity)$coef[1:m,2]
  df<-data.frame(x_ordinate,y_ordinate)
  plot(x_ordinate, y_ordinate, ylab="y", xlab="x", col="blue", asp=1)
  text(x_ordinate, y_ordinate, sort(unique(clicks$place)), cex=0.65, pos=1, col="red")
  rect(x_ordinate-xerr,y_ordinate-yerr,x_ordinate+xerr,y_ordinate+yerr)
  u = abs((fitx$res)/as.numeric(clicks$xlim))
  v = abs((fity$res)/as.numeric(clicks$ylim))
  if(fitx$rank < (m+n-1)) print("Error:the sreecshots don't make a connected map.")
  if(max(u)>0.05 | max(v)>0.05) print(paste("Warning:the fiited map is no reliable"))
}
my_plot(clicks)

clicks$ss=factor(clicks$ss)
my_plot(clicks)
```

5.5 Final Plot

Now from the code we get our global plot -



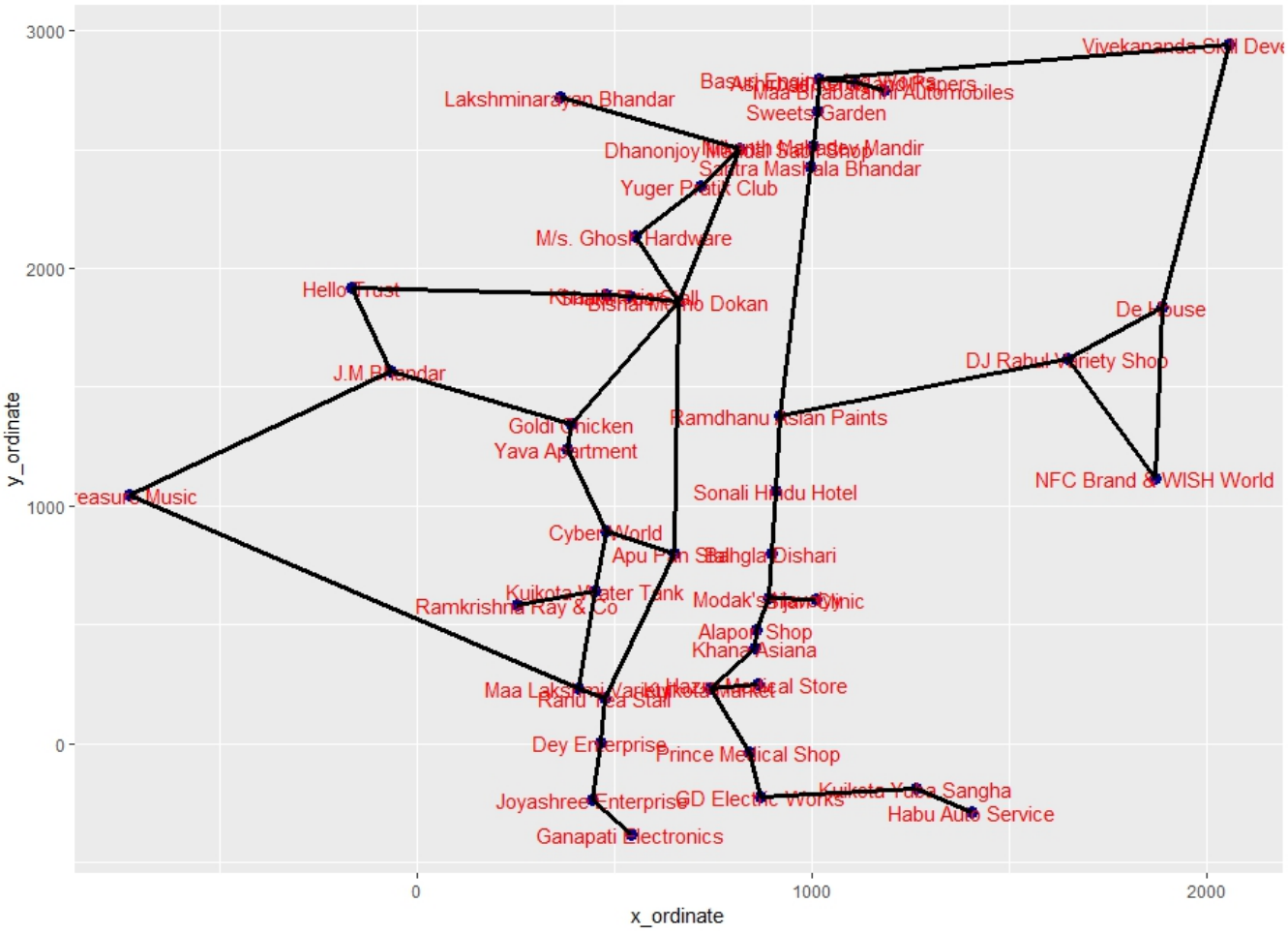
5.6 A complete Road Map

Here we give code for the complete Road Map of the Area-

```
new2=data.frame(x_ordinate,y_ordinate)
write.csv(new2,file.choose())
new=read.csv(file.choose())
```

5.7 The Road Map

Now the Road map is -



The R-code-

```
sp<-ggplot(new,aes(x_ordinate,y_ordinate,label = X))+
  geom_point(colour='darkblue',size=3)
sp
data_text<-data.frame(location=new$X,
                      x=new$x_ordinate,
                      y=new$y_ordinate)
sp=sp+
  geom_text(data=data_text,
            mapping=aes(x=x,
                        y=y,
                        label=location)
            ,colour='red')
name=sort(unique(clicks$place))
k=1
while(k>0)
{
  print(sp)
  cat("Write the two places' name you want to connect by road")
  place1=readline("place1-")
  place2=readline("place2-")
```

```
  index1=which(name==place1)
  index2=which(name==place2)
  if(length(index1)==0||length(index2)==0)
  {
    cat("Location Entered does not exist in the map")
    flag=readline("If you want to add more road press y or If you want to stop here press n")
    if(flag=='y')k=k+1
    if(flag=='n')k=0
  }
  else {sp=sp+geom_segment(x=x_ordinate[index1],
  y=y_ordinate[index1],xend=x_ordinate[index2],yend=y_ordinate[index2],size=1.3)
  flag=readline("If you want to add more road press y or If you want to stop here press n")
  if(flag=='y')k=k+1
  if(flag=='n')k=0}
}
```

5.8 Making a Zoomable Road-map

Here I am giving the code to make the plot zoomable and nevigatable Here is the Code-

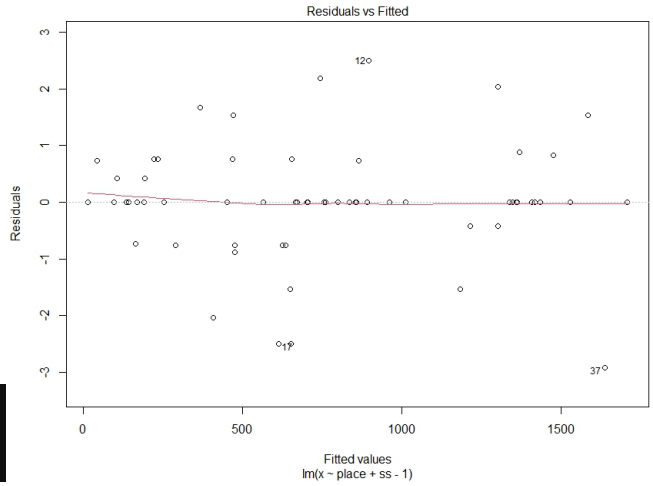
```
q <- ggplotly(sp, dynamicTicks = TRUE)
config(q, scrollZoom = TRUE)%>%layout(plot_bgcolor='#e5ecf6',
                                      xaxis = list(
                                        zerolinecolor = '#ffff',
                                        zerolinewidth = 2,
                                        gridcolor = 'ffff'),
                                      yaxis = list(
                                        zerolinecolor = '#ffff',
                                        zerolinewidth = 2,
                                        gridcolor = 'ffff')
)
```

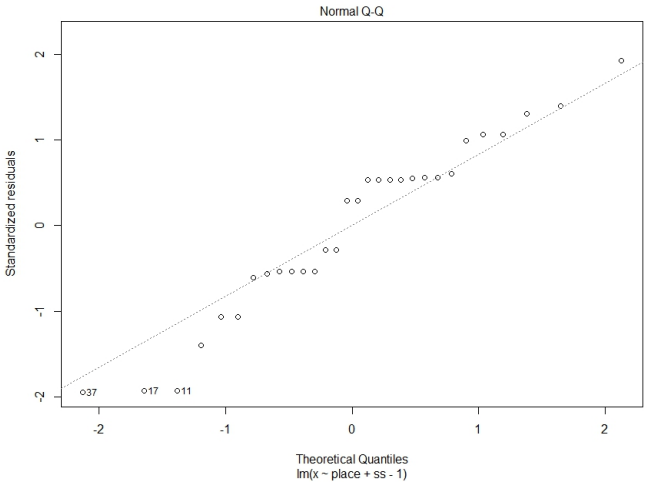
6 Residual Analysis

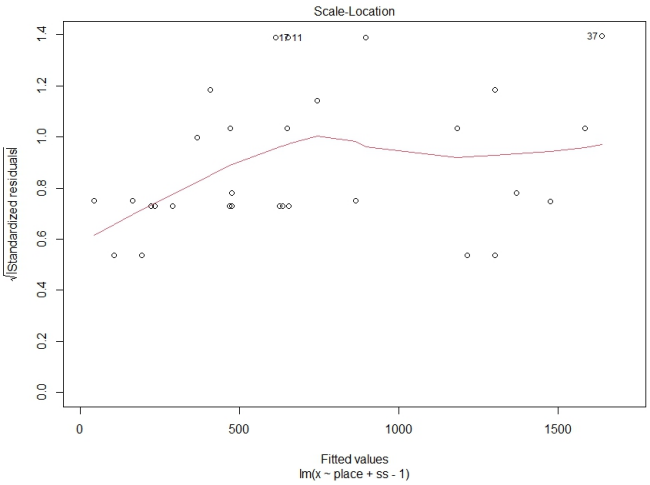
6.1 For $x-$ordinates

We have got three plots for the residual analysis.They are R is suggesting that observation 11,12,17,37 are outliers.

```
fitx = lm(x~place+ss-1, clicks)
plot(fitx)
```

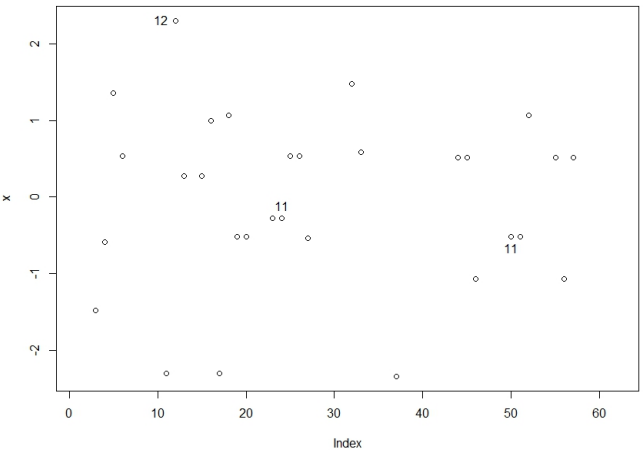




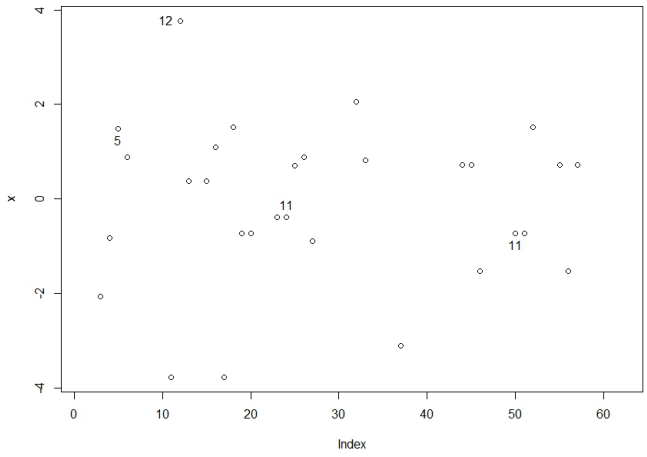


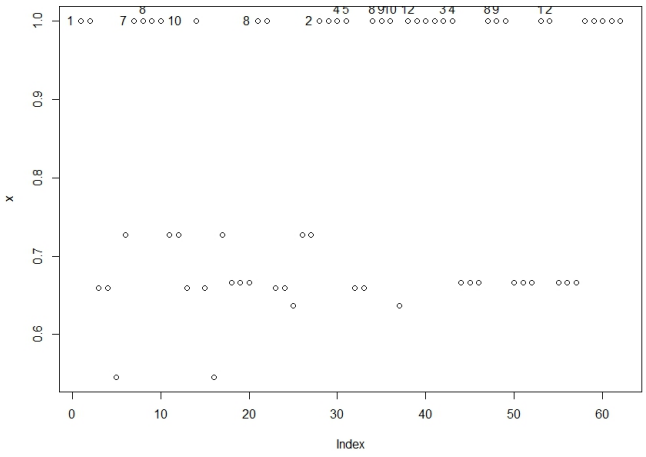
We will try to find this outliers manually by making checkpoint function. Later I will explain why there are points with leverage 1. Manually I find observation 12 and 17 to be outliers. I checked the dataset and found that both

```
checkpoint=function(x,lab)
{
  plot(x)
  identify(x,lab=lab)
}

checkpoint(rstudent(fitx),c(1:length(fitx)))
[1] 12 24 50
checkpoint(dffits(fitx),c(1:length(fitx)))
[1]  5 12 24 50
checkpoint(hatvalues(fitx),c(1:length(fitx)))
[1]  1  7  8 10 21 28 30 31 34 35 36 38 42 43 47 48 53 54
```



of them are the same place in two different screenshots. This is expected. As if I take a wrong measurement of an important point(a point that is overlapping in two consecutive screenshots), the residual of the point in both the

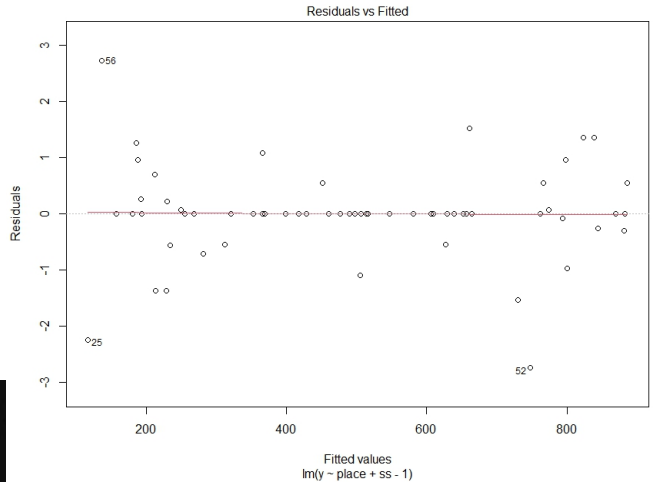




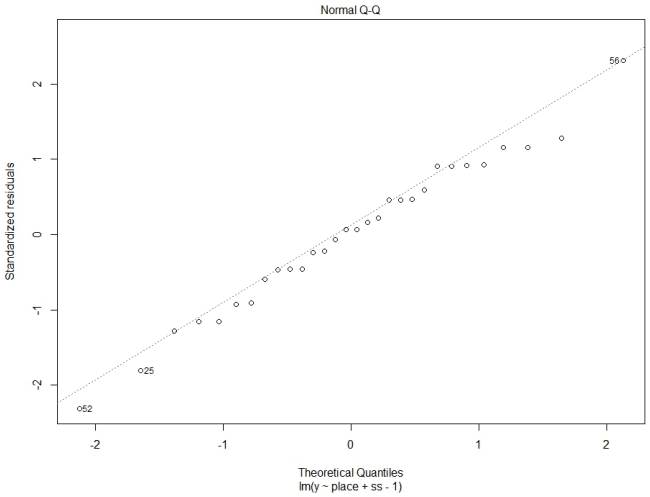
screenshots will be high. Hence outliers will occur in even numbers in mapping problem.
The above argument also gives an idea of why the places which are in only one screenshot have low residual. We observe hatvalues of such points to be very close to 1(exactly 1 is not possible in a statistical model).

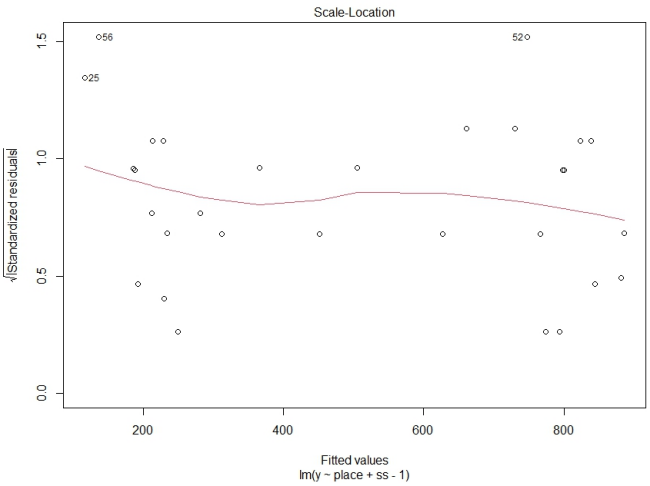
6.2 For $y - ordinate$

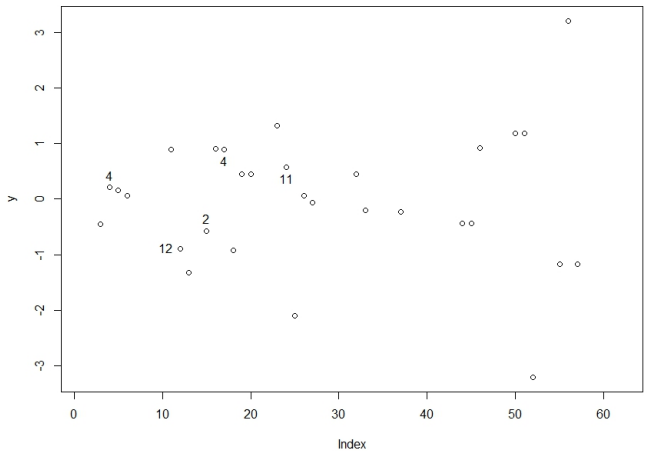
For the $y - ordinate$ also we found similar results in R.
R is suggesting that the observations 25,52 and 56 are the outliers.

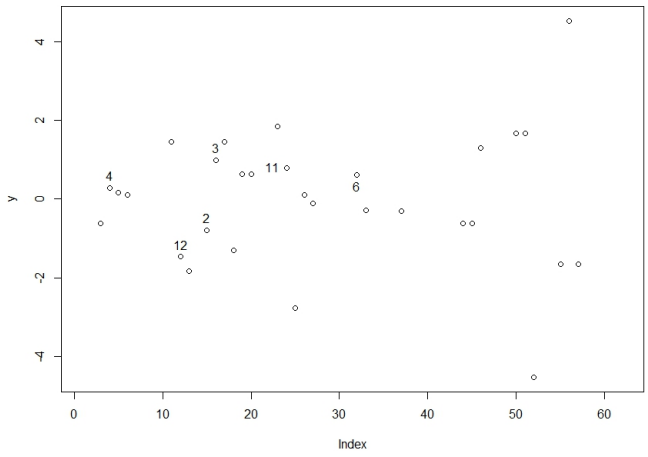


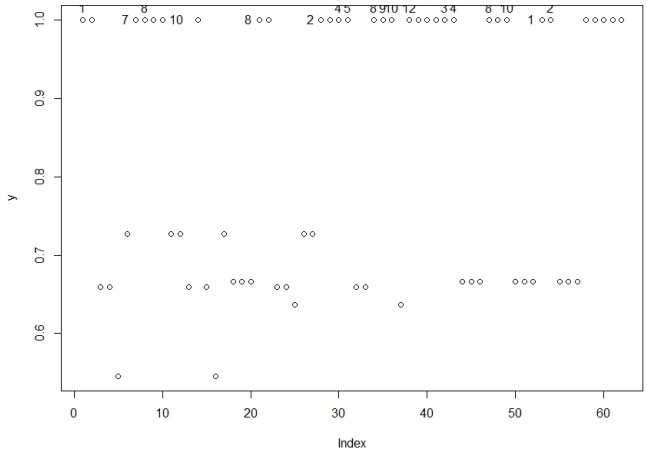

```
fity = lm(y~place+ss-1, clicks)
plot(fity)
```












```
checkpoint=function(y,lab)
{
  plot(y)
  identify(y,lab=lab)
}

checkpoint(rstudent(fity),c(1:length(fity)))
[1]  4 12 15 17 24
checkpoint(dffits(fity),c(1:length(fity)))
[1]  4 12 15 16 24 32
checkpoint(hatvalues(fity),c(1:length(fity)))
[1]  1  7  8 10 21 28 30 31 34 35 36 38 42 43 47 49 53 54
```

Manually I found no outliers here.
Fitting models for the x-ordinates and the y-ordinates are done independently using independent set of values. Hence one observation being an outlier in fitx model does not imply it will be an outlier for the fity model as well.
What I expect to be the same is the observations with high heatvalue(close to 1)