Advanced Statistical Methods
Subject code: 301115
Assignment -1

Topic: Car crashed data analysis and building model with given data-set: assignment1.csv
Data Source: Given data of .csv format inside data are in quantitative and categorical.
Inside data information: Car crashed data of 1500 information of 5 feature (Color, Hour, Minutes, Longitude, Latitude)

1.a

Create an new variable of day-of-time form given data frame of Hour and Minute new data frame of Timeofday=c(Hour+Minutes) [code: 1a]

	Colour	Hour	Minutes	Longitude	Latitude	TimeofDay
1	dark	16	7	151.0476	-33.80303	1607
2	light	15	36	151.0911	-33.76662	1536
3	light	15	49	150.7100	-33.74679	1549
4	dark	16	32	151.2432	-33.76362	1632
5	light	8	44	150.8819	-33.88775	844

1.b

Kernel density estimation for light and dark color vehicle.
Procedure: Create subset for light colour and dark colour respectively and calculate their density then plot their density respectively [code: 1b].

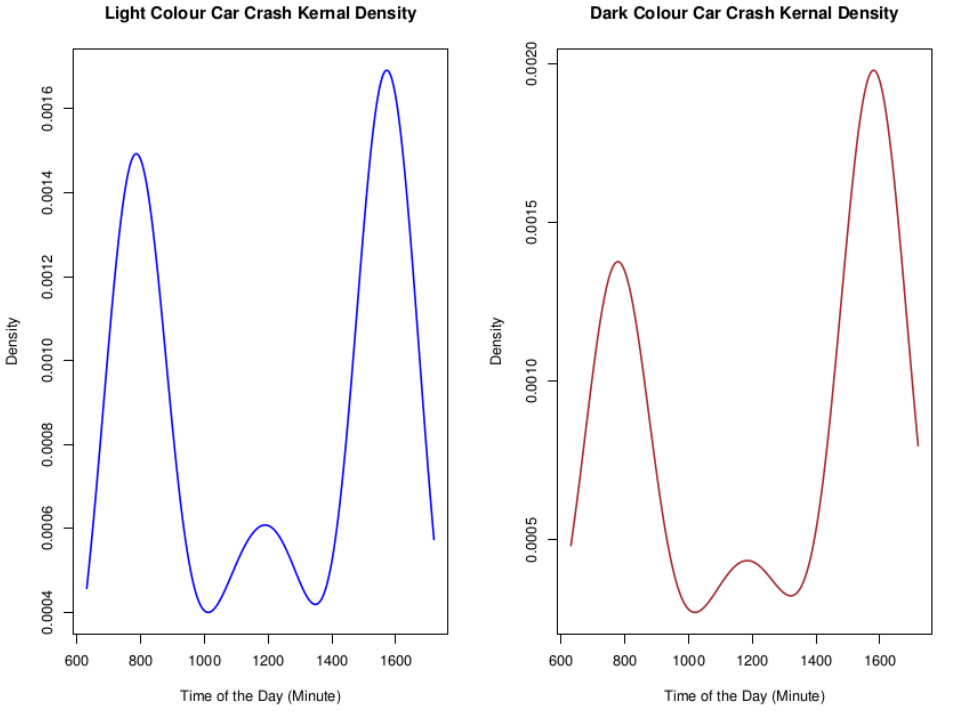

Figure Summary: The left hand side figure is showing the kernel density of light colour where the x-axis is Time of the Day and y-axis is the density estimation. The right hand side of the curve is represent the dark colour car density.

1.c:

To construct a plot using KDEs and Bayes rule need to calculate density of car_data using the data frame of DateofTime and density of light with data frame DateofTime of color='light' and dark colour density with data frame of DateofTime of color='dark' of car data-set respectively then calculate the value of dark colour car mean. Then calculate KDE and Bayes rule to construct a plot. To predict the colour of any certain point of car-crashed data-set using the formula - probability of getting colour (round(prob of dark colour)of value x ==Time)*value of the probability(p) / (probability of dark density of function value y (round(density of dark of factor value of x)==Time) + (1 - probability)* probability of light density*round (light color density of factor x)==Time)) .This function will return a value as string as of 'light' or 'dark' [code: 1c].

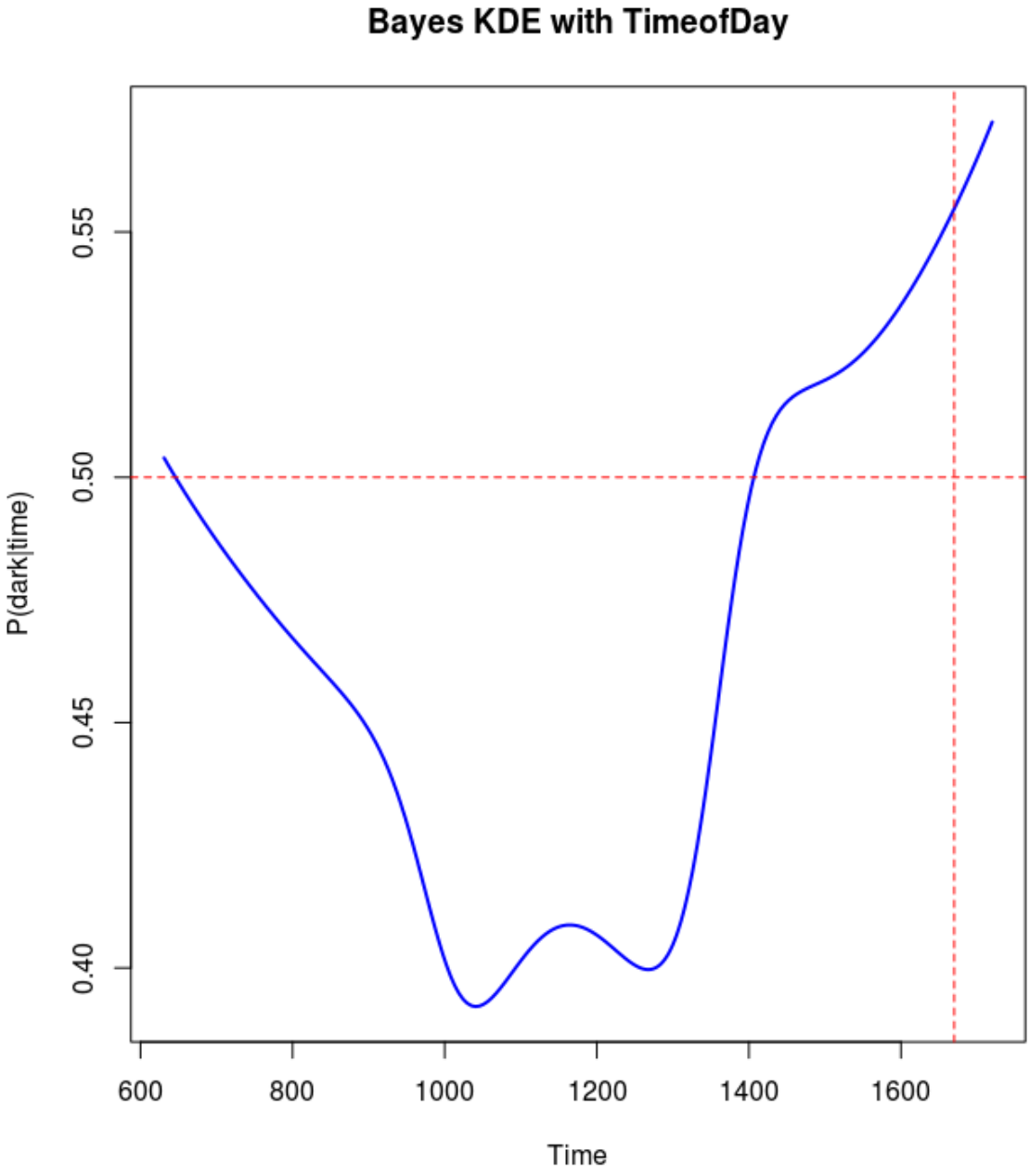

Figure Summary: Bayes Kernel Density Estimation for car crash data for colour dark and light w.r.t Time in x-Axis. In the above graph there is an horizontal and vertical line is there which is cross each others in a certain points. The vertical red line will vary in every iteration randomly w.r.t time and the vertical line is the density estimate line which is fixed at points 0.5. The cross section of both line estimate the colour at that point.

2.

To find the relationship among car accident and TimeofDay when the accident occurs. You need to install package ‘mixtool’. Now to follow the procedure with parametric bootstrap of 2, 3 and 4 component. With the mixture model we should have to find the density of accident occurrence. Run the model for 1000 iteration for 4 component using function “normalmixEM” and find the relationship with Akaike’s Informatino Criteria (AIC) and Bayesian Information Criteria (BIC) . Before compare the result with AIC and BIC need to calculate the bootstrap comparison using function (use function: **“boot.comp” which will take more than 2 hours for 4 component**).
To do those things need to estimate normal maximum estimation for 2, 3 and 4 component.
Bellow picture represent the 2, 3 and 4 component estimation for density.

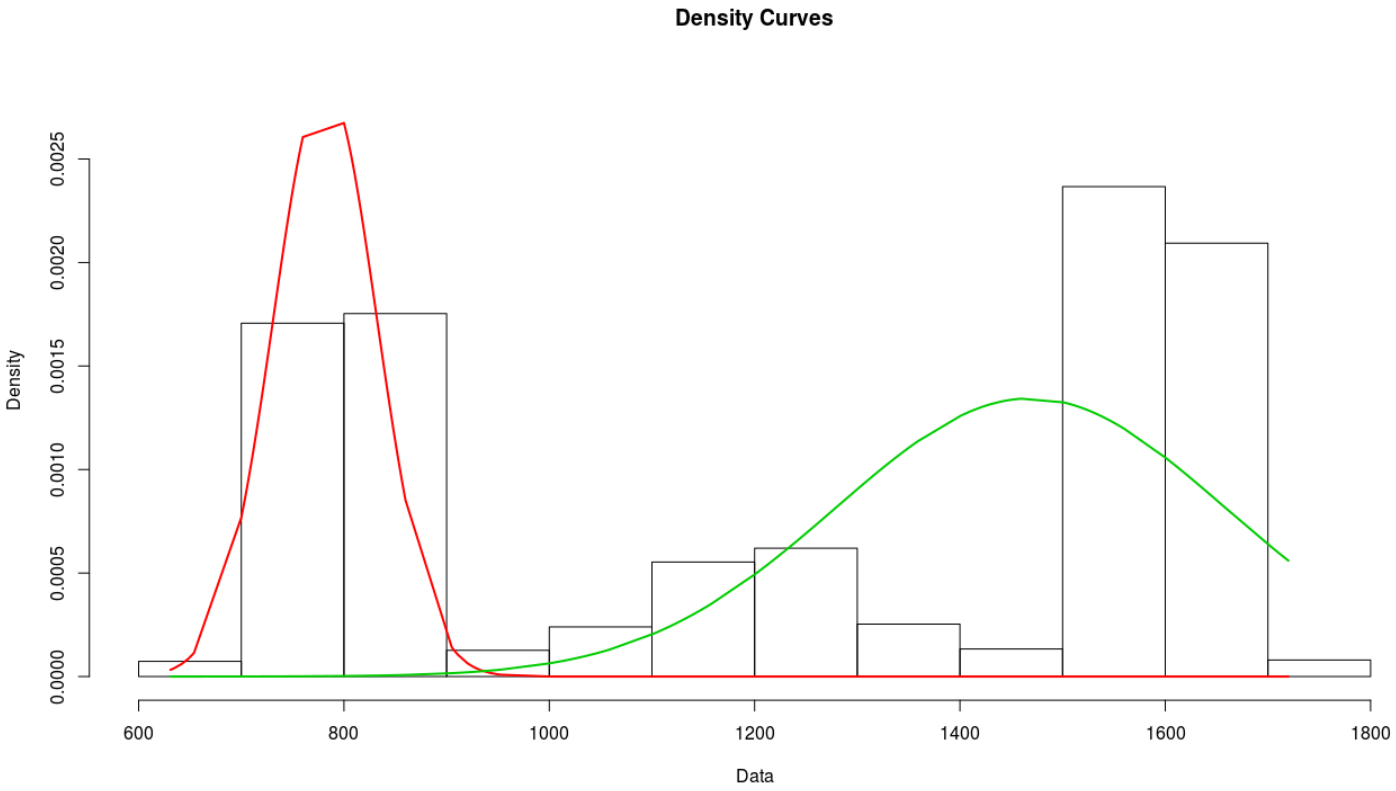

Figure Summary: The above picture is the normal distribution with **2 component** for data frame ‘TimeofDay’ of car crash data-set. This two component value are the `posterior values of the given component of n*k matrix. Where n is the number of observation and k is the number of component. Here is the green curve for component1 and red curve for companent2 and for n number of observation it will plot a graph with their respective values. The vertical bars are the final ‘loglik’ values for log-likehood method. Here the number of evaluated log-likehood final values is 12.`

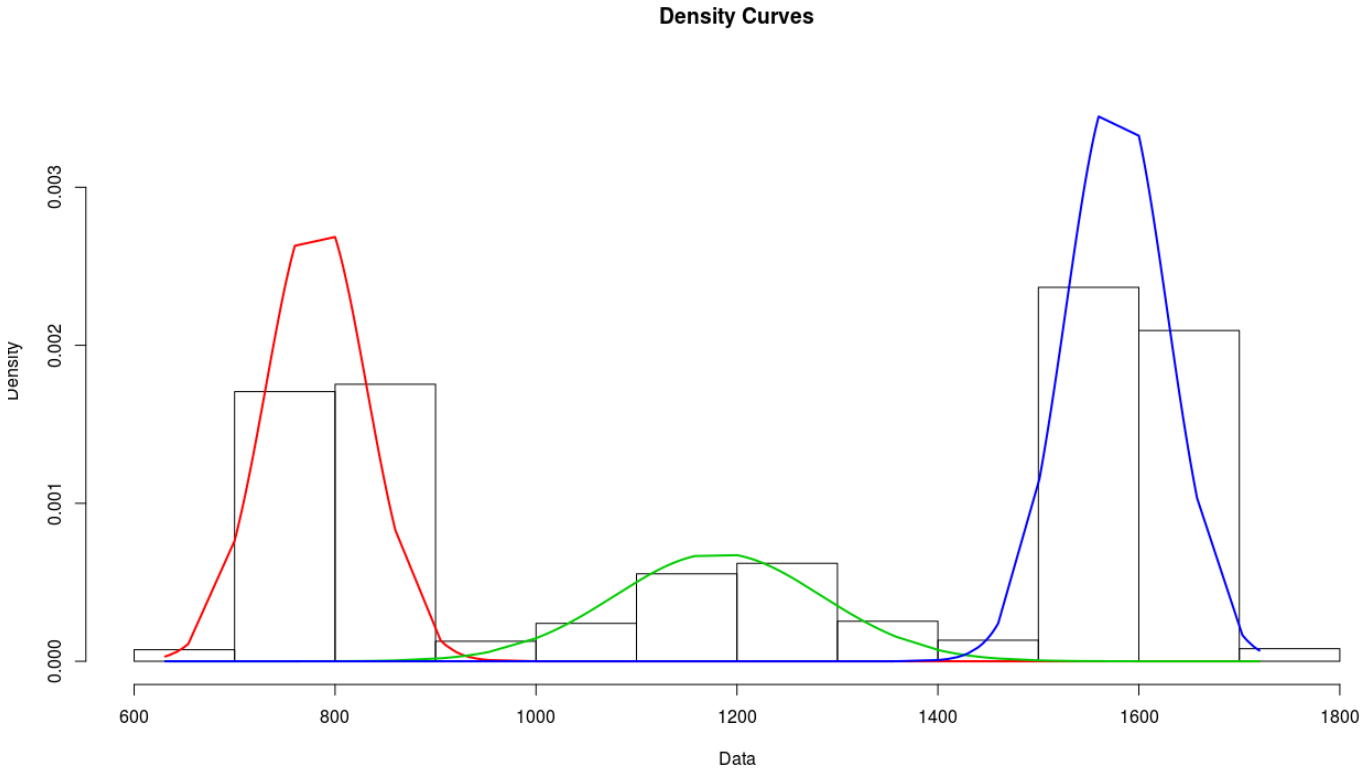

Figure Summary: The above picture is the normal distribution with **3 component** for data frame 'TimeofDay' of car crash data-set. This three component value are the posterior values of the given component of n*k matrix. Where n(=333) is the number of observation and k (=3) is the number of component. Here is the green curve for component1 and red curve for companent2 and blue curve component3 for n number of observation it will plot a graph with their respective values. The vertical bars are the final 'loglik' values for log-likehood method. Here the number of evaluated log-likehood final values is 33.

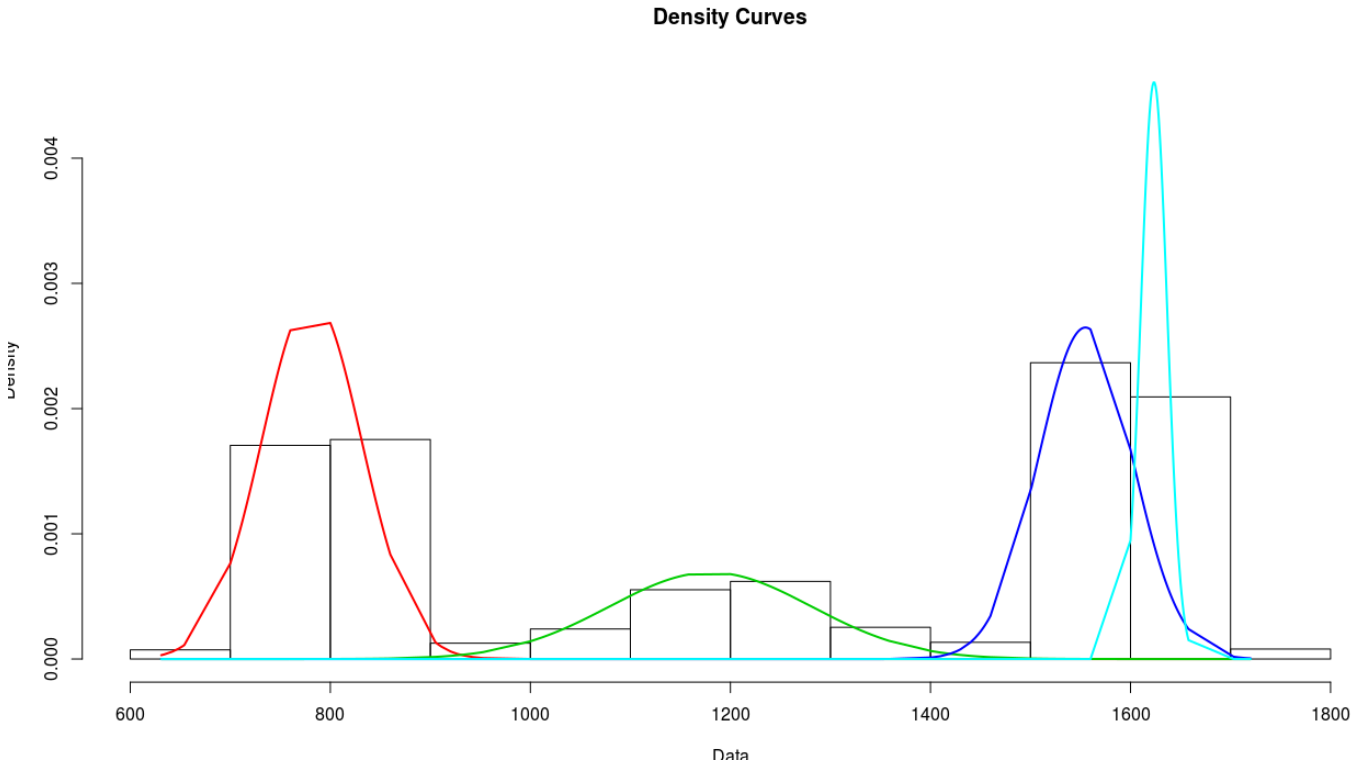

Figure Summary: The above picture is the normal distribution with **4 component** for data frame 'TimeofDay' of car crash data-set. This three component value are the posterior values of the given component of n*k matrix. Where n(=250) is the number of observation and k (=4) is the number of component. Here is the green curve for component1 and red curve for companent2 and blue curve component3 and cyan curve for component4 for n number of observation it will plot a graph with their respective values. The vertical bars are the final 'loglik' values for log-likehood method. Here the number of evaluated log-likehood final values is 41.

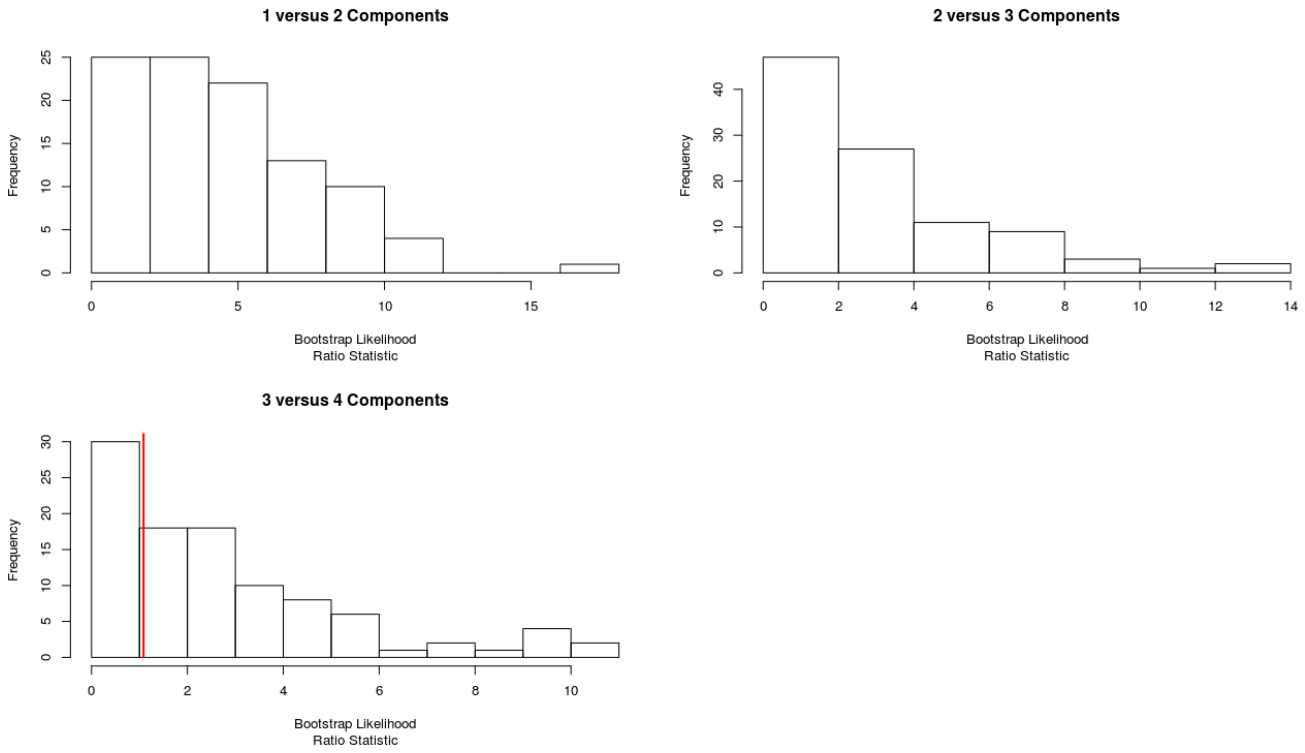

Figure Summery: This picture is shows the 4 component analysis using bootstrap likelihood estimation for sequential testing for n component, where frequency in X-axis and bootstrap estimation in Y-axis. Bootstrap will return with value of p-values for each test of k-components versus (k+1)-components. The B bootstrap realizations of the likelihood ratio statistic of 3 (4-1) component values. Where each 3 companent value will give differnt values for 100 iteration.

Now move to parametric density estimation with bootstrap component analysis, where I using "normalmix" to mixture the component and iterate for 1000 iteration for best performance. After that calculate the mix model summation value which will be use AIC and BIC model. Form an AIC (AIC method, $AIC = -2\log L + 2$) and BIC model (BIC method,BIC=-2logL + Klogn) where by running the code getting value for AIC: and BIC: and plot them in a single plot to get which model is best and giving promising result [code:2].

Now need to estimate The B bootstrap realizations of the likelihood ratio statistic using the following formula – *sum(normal distribution(TimeofDay data frame) mean(TimeofDay data frame) , standard deviation (TimeofDay of car-crashed data))* it returns the value: -10967.03.
After successful running of the code got the value for AIC and BIC as-
AIC: `21944.06 20271.46 19398.21 19233.97`
BIC: `21956.06 20301.46 19446.21 19299.97`

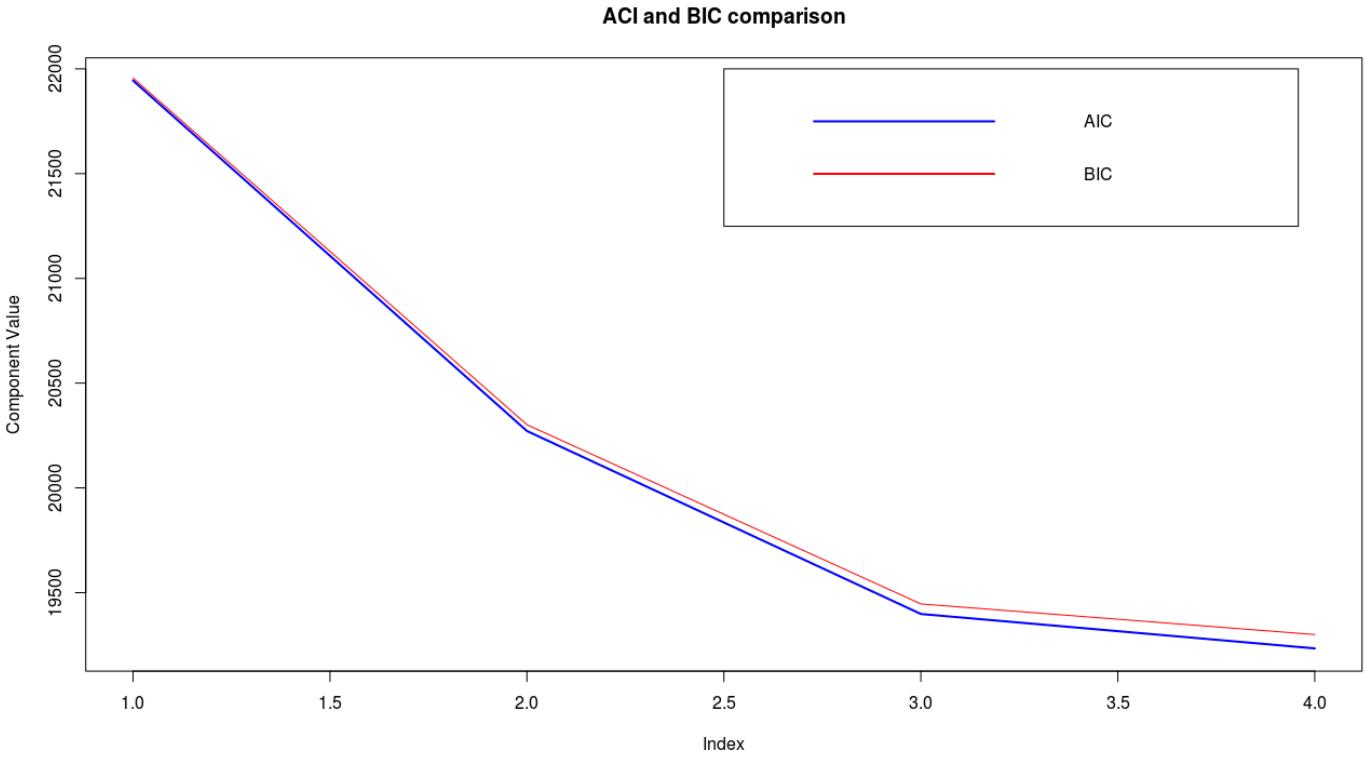

Figure Summary: From the above figure notice that AIC given comparative good result rather than BIC model. The graph is steps of 4 for the selected 4 component. Where the index value is start at position 1 and second at 2 and third at 3 and forth at 4 index. If you notice carefully in the graph then we can say for step 1 results is near to same for both but it's slightly differ in second step of index 2 then it's become more differ in step 3 rather than 2 and finally at step 4 of index 4 it's far away difference rather than previous by near to 50. Where BIC has 50 more values than AIC in component values.

AIC and BIC model comparison:
After comparing the graph above we can conclude that for this data-set of 4 component analysis with AIC and BIC, AIC gives good results than BIC where the maximum likelihoods estimation is calculated by using bootstrap method.

3.a: To plot a contour need to calculate the 2-dimensional Kernel Density Estimation for the data frame of Latitude and Longitude of car crash data then make a contour plot with the function contour [code: 3a]. To use the 2-dimensional Kernel Density Estimation need to install 'MASS' packages. Calculate the 2-dimensional Kernel Density Estimation use this formula- *kde2d (Latitude data frame, Longitude data frame).*
Kde2d will return with value x,y the coordinate value of the grid points, vector of length n, here the n value is 25.
A z value will return also of n1 and n2 matrix of the estimated density: rows correspond to the value of x, columns to the value of y.

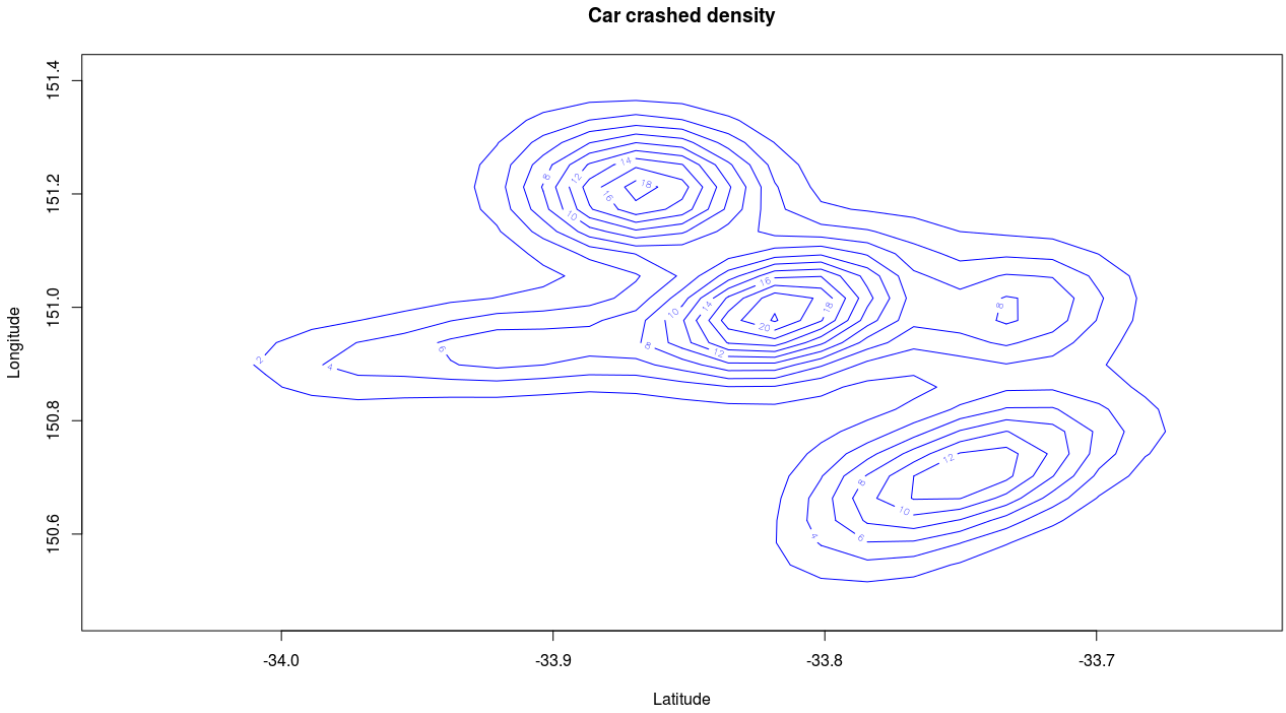

Figure Summary: 2-D Kernel density distribution for car crash data-set of data frame Longitude and Latitude. Here the density estimation for this grid is calculate from the z values of the kde2d value which will return with density estimation with 5 component of 25 rows for each data component.

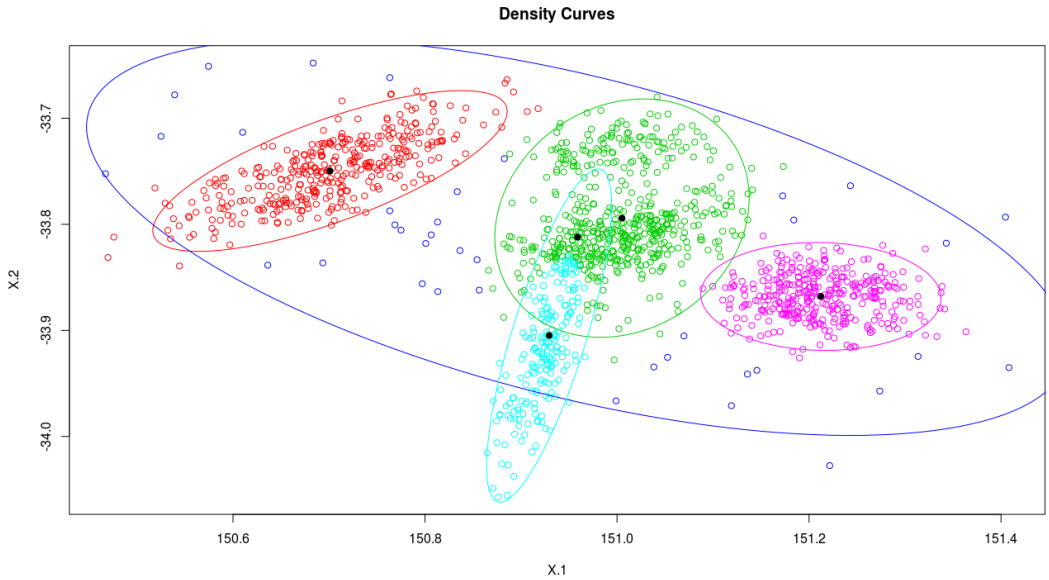

Figure Summary: 2-D normal mixture model density curve with 5 cluster. Where the cluster is distinguished with different color and surrounded with different color. The above picture for blue color cluster take a complete cluster with all the possible data-set.

3.c:

Here to show the difference of K-means and normal distribution of 5 component first we plot the data then comparing the model to find which one is given the best results. Along with that construct a cross-cluster membership for competitive comparison among the above two model and find which one is given the best results.

Procedure:

1. Find “kmeans” for 5 cluster with the function “kmeans”.
2. make graph segment for two parallel graph of k-means and normal distribution.
3. plot the k-means distribution.
4. plot the 2-D normal distribution.
5. calculate normal mixture membership with function “apply”
6. calculate k-means cluster membership with function “kmeans”
7. Make cross_cluster_membership using function “addmargins” of argument normal mixture membership and k-means cluster membership.
8. calculate cross membership for comparison with the normal distribution and k-means distribution results.

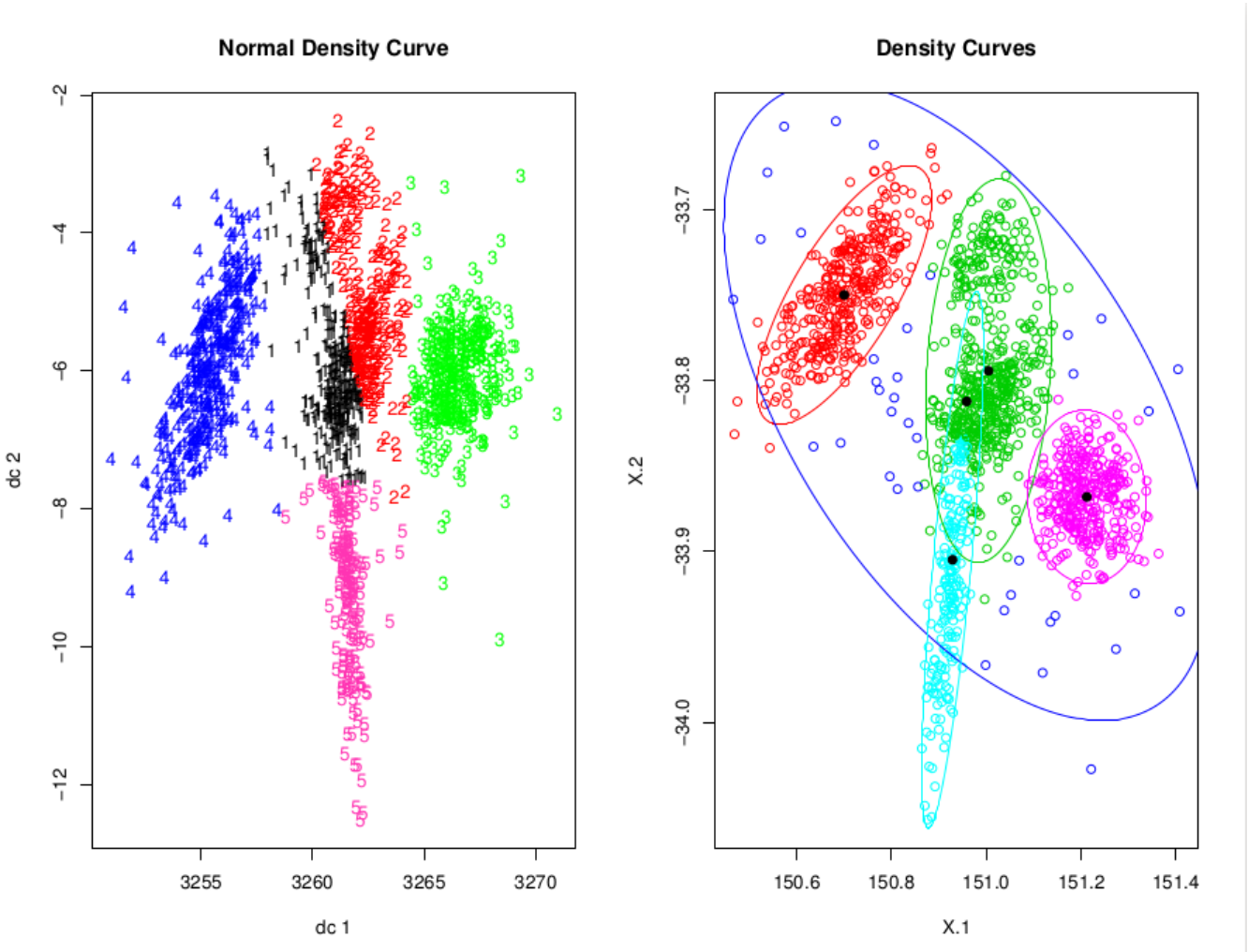

Figure Summery: Left hand picture shows the normal density distribution where the cluster is segregate as number of 1-5 which is appear in different color .Right hand plot is the K-means where the cluster plot is bounded by a circle with different color for different cluster.

Cross cluster membership operation:
Cross cluster value of cl1 and cl2 which give an significant cluster location among 5 cluster points.

cl1 \ cl2	1	2	3	4	5	Sum
1	353	0	0	0	1	354
2	12	12	7	6	1	38
3	8	1	0	49	487	545
4	0	179	182	0	2	363
5	0	0	0	189	11	200
Sum	373	192	189	244	502	1500

Cluster and Model comparison:
Here notice that the K-means cluster is very close similar for individuals. Qualitative difference between two cluster is that- in K-means cluster choose the cluster in a certain region which points is similar characteristic only but those points are far away those points are not consider in the cluster. So in a cluster points those values are belong for away k-means clustering are not considering to make a cluster group.

In normal density distribution cluster making on the number not by surrounding in a region so all the data points are similar behavior are consider in a cluster it will not ignore any data points those characteristic is same but position is far away from the cluster center.

Appendix

Code:1a
```
CAR_CRASH<- read.csv("infinizifl.csv")
CAR_CRASH$TimeofDay= CAR_CRASH$Hour*100 + CAR_CRASH$Minutes
```

Code:1b
```
mn=min(CAR_CRASH$TimeofDay)
mx=max(CAR_CRASH$TimeofDay)
lc_car <-subset(CAR_CRASH, CAR_CRASH$Colour=="light")
lc_car_Density <-density(lc_car$TimeofDay,from = mn,to = mx)
plot(lc_car_Density, xlab = 'Time of the Day (Minute)',
    main="Light Colour Car Crash Kernal Density",lwd=2, col='blue' )

dc_car <-subset(CAR_CRASH, CAR_CRASH$Colour=="dark")
dc_car_Density <-density(dc_car$TimeofDay,from = mn,to = mx)
plot(dc_car_Density, xlab = 'Time of the Day (Minute)',
    main="Dark Colour Car Crash Kernal Density",lwd=2, col='brown')
```

code: 1c

```
pred<-function(time){
  f = density(CAR_CRASH$TimeofDay, from=mn, to=mx)
  f_light = density(CAR_CRASH$TimeofDay[CAR_CRASH$Colour=="light"],from=mn, to=mx)
  f_dark = density(CAR_CRASH$TimeofDay[CAR_CRASH$Colour=="dark"],from=mn, to=mx)
  p = mean(CAR_CRASH$Colour=="dark")
  plot(f$x, f_dark$y*p/(p*f_dark$y+(1-p)*f_light$y), main = 'Bayes KDE with TimeofDay',
       type='l', xlab='Time', ylab='P(dark|time)', col='blue',lwd=2)
  abline(v=time, lty='dashed',  col='red')
  abline(h=0.5,lty='dashed', col='red')
  p_dark = f_dark$y[round(f_dark$x)==time]*p/(p*f_dark$y[round(f_dark$x)==time]+(1-
p)*f_light$y[round(f_light$x)==time])
  return (ifelse(p_dark<0.5,'light','dark'))}
r=sample( mn:mx,1)
pred(r)
```

code: 2

```
#2 component Mixtures
library(mixtools) # install mixtools package in system
comp_mix2 = normalmixEM(CAR_CRASH$TimeofDay, k=2) #k=plot
plot(comp_mix2, whichplots=2) # whichplot=2: density plot

#3 component Mixtures
comp_mix3 = normalmixEM(CAR_CRASH$TimeofDay, k=3)
plot(comp_mix3, whichplots=2)

#four component Mixtures
comp_mix4 = normalmixEM(CAR_CRASH$TimeofDay, k=4)
plot(comp_mix4,whichplots=2)
# parametric bootstrap for sequential testing of 4 component
btstrp = boot.comp(CAR_CRASH$TimeofDay, max.comp = 4, mix.type = "normalmix", B=1000)

#Making the mix model
loglik1 = sum(dnorm(CAR_CRASH$TimeofDay, mean(CAR_CRASH$TimeofDay),
sd(CAR_CRASH$TimeofDay), log=TRUE))

#Trying AIC method, AIC=−2logL+2K
aic = -2*c(loglik1+2*2, comp_mix2$loglik+2*(3*2-1), comp_mix3$loglik+2*(3*3-1),
comp_mix4$loglik+2*(3*4-1))
aic

#Trying BIC method,BIC=-2logL + Klogn
n = length(CAR_CRASH$TimeofDay)
aic = c(-2*loglik1+2*2, -2*comp_mix2$loglik+2*(3*2-1), -2*comp_mix3$loglik+2*(3*3-1),
-2*comp_mix4$loglik+2*(3*4-1))
bic
# AIC and BIC Comparison plot
plot(aic, type='l', col='blue',lwd=2, ylab ='Component Value', main = 'ACI and BIC comparison')
lines(bic, type='l', col='red')
legend(3,20500,legend=c('AIC','BIC'), col=c('blue','red'))
```

code:3a

```
library(MASS) #need to install MASS package in system
```

```
CAR_CRASH_2D_KDE = kde2d(CAR_CRASH$Latitude, CAR_CRASH$Longitude)
contour(CAR_CRASH_2D_KDE, main='Car crashed density',xlab='Latitude', ylab='Longitude',
col='blue')
```

code: 3b

```
x = cbind(CAR_CRASH$Longitude, CAR_CRASH$Latitude)
Normal_mixture_2D = mvnormalmixEM(x, k=5, epsilon = 1e-03, maxit = 1000)
plot(Normal_mixture_2D, whichplots=2)
```

code: 3c

```
km <- kmeans(x,5)
par(mfrow=c(1,2))
library(cluster)
library(fpc)
plot(x, km$cluster, main ='K-means density curve')  #k-means cluster
plot(Normal_mixture_2D, whichplots=2) #mixture model cluster

cl1 = apply(Normal_mixture_2D$posterior,1,which.max) # normal mixture cluster membership
cl2 = kmeans(x,5)$cluster #k-means cluster membership
#cross cluster membership of cl1, cl2
cross_cluster_membership<-addmargins(table(cl1,cl2))
cross_cluster_membership
```