DS 6030 HW05 Density

Ben Wilson

'10/10/2022

Contents

Bat Data

Load data for R

```
#load data
bat_data <- read.csv("C:\\Users\\brwil\\Desktop\\SY MSDS\\DS 6030 Stat Learning\\Week 6\\geo_profile.cs

names(bat_data)[names(bat_data) == 'X2.5817631825286242'] <- 'bat_count'
```

Problem 1 Geographic Profiling

Geographic profiling, a method developed in criminology, can be used to estimate the home location (roost) of animals based on a collection of sightings. The approach requires an estimate of the distribution the animal will travel from their roost to forage for food.

A sample of 283 distances that pipistrelle bats traveled (in meters).

1a. Derive the MLE for theta (i.e., show the math).

```
#insert picture of math derivation
knitr::include_graphics('C:/Users/brwil/Desktop/SY MSDS/DS 6030 Stat Learning/Week 6/Problem1AMath.png')
```

$$L(\theta) = \prod_{i=1}^{n} f(x_i|\theta) = \prod_{i=1}^{n} \frac{x_i}{\theta} \exp\left(\frac{-x_i^2}{2\theta}\right)$$

$$\text{Log } L(\theta) = \sum_{i=1}^{n} \log\left(\frac{x_i}{\theta} \exp\left(\frac{-x_i^2}{2\theta}\right)\right)$$

$$= \sum_{i=1}^{n} \left(\log\left(\frac{x_i}{\theta}\right) + \log \exp\left(\frac{-x_i^2}{2\theta}\right)\right)$$

$$= \sum_{i=1}^{n} \log x_i - \log(\theta) \frac{x_i^2}{2\theta}$$

$$= -n \log(\theta) + \sum_{i=1}^{n} \left(\log(x_i) \frac{-x_i^2}{2\theta}\right.$$

$$\frac{\partial \log L(\theta)}{\partial \theta} = \frac{-n}{\theta} + \sum_{i=1}^{n} \frac{+x_i^2}{2\theta^2} = \frac{-n}{\theta} + \frac{1}{2\theta^2} \sum_{i=1}^{n} x_i^2$$

$$0 = \frac{-n}{\theta} + \frac{1}{2\theta^2} \sum_{i=1}^{n} x_i^2$$

$$\frac{n}{\theta} = \frac{1}{2\theta^2} \sum_{i=1}^{n} x_i^2$$

$$2n\theta^2 - \theta \sum x_i^2 = 0$$

$$\theta(2n\theta - \sum x_i^2) = 0$$

$$\frac{2n\theta}{2n} = \frac{\sum x_i^2}{2n}$$

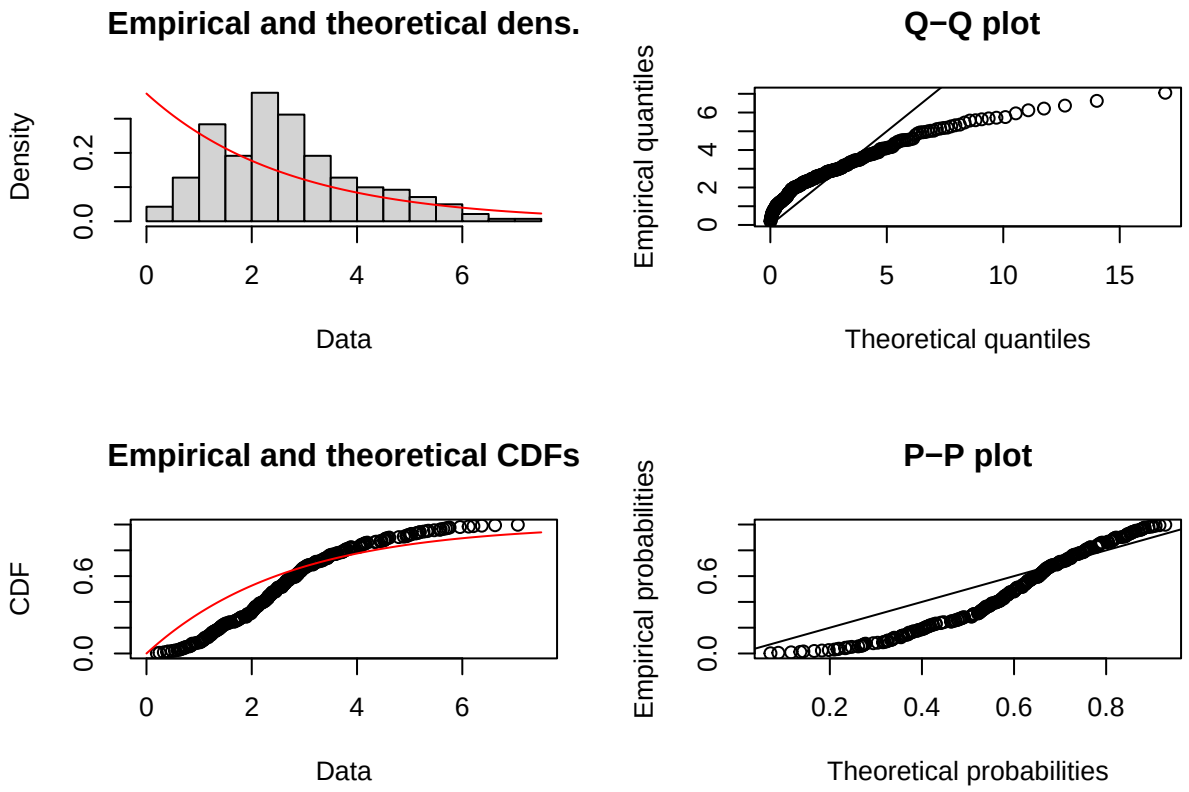
$$\theta = 0, \frac{\sum x_i^2}{2n}$$

↳ due to 0 not being an allowed value for θ, use second option of $\frac{\sum x_i^2}{2n}$

1b. What is the MLE of theta for the bat data? (Use results from a, or use computational methods.)

Distribution fit and plot of output

```
expon_dist <- fitdist(bat_data$bat_count, "exp", method="mle")
plot(expon_dist)
```



Theta & Standard Error

```
summary(expon_dist)
```

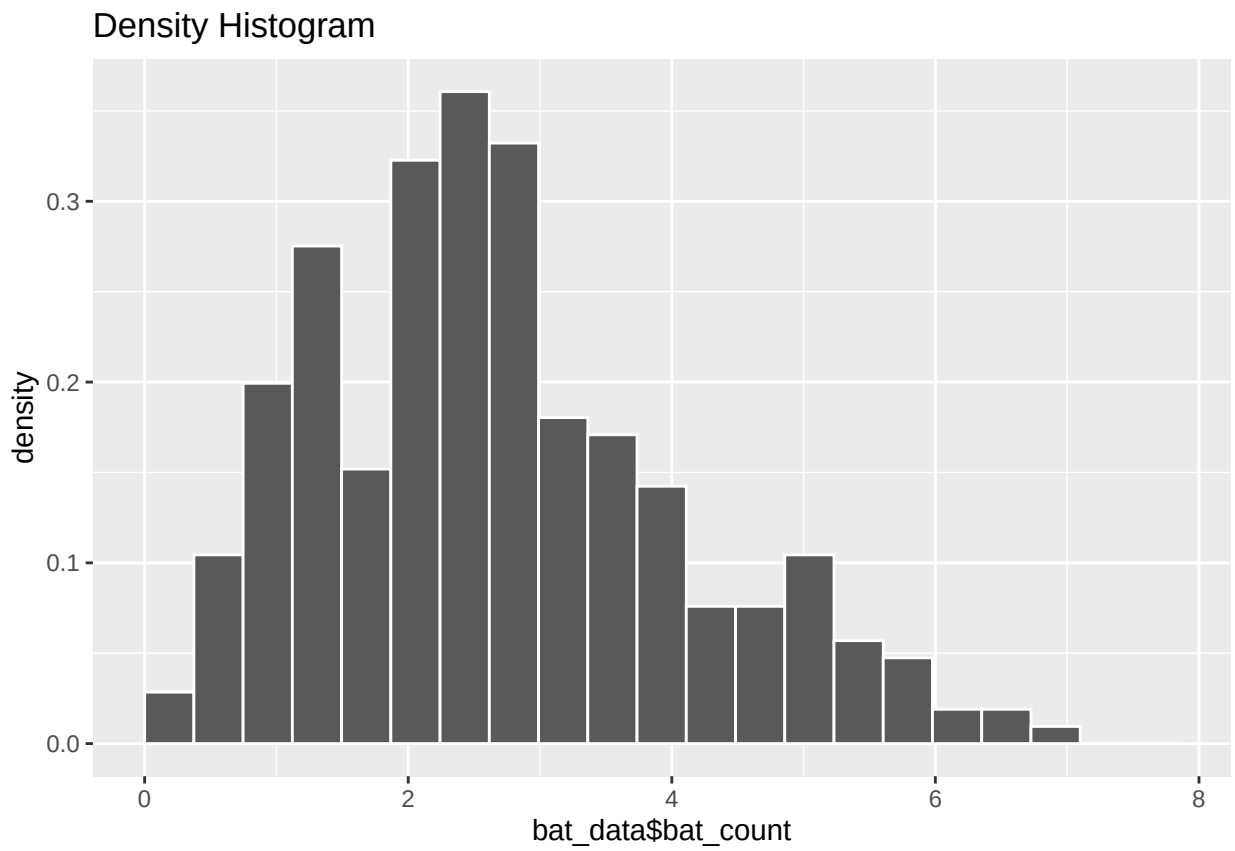
```
## Fitting of the distribution ' exp ' by maximum likelihood
## Parameters :
##       estimate Std. Error
## rate 0.3736245 0.02224885
## Loglikelihood:  -559.6302   AIC:  1121.26   BIC:  1124.902
```

The estimate of 0.3736245 is the estimated bandwidth.

1c. Using the MLE value of theta from part b, compute the estimated density at a set of evaluation points between 0 and 8 meters. Plot the estimated density.

```r
#histogram settings
bw = 0.3736245    #binwidth parameter
bks = seq(0, 8, by = bw)

#frequency histogram
ggplot() +
  geom_histogram(aes(x=bat_data$bat_count, y=after_stat(density)), breaks = bks, color="white") +
  labs(title="Density Histogram")
```




```r
#calculate density
density(bat_data$bat_count)
```

```
##
## Call:
##  density.default(x = bat_data$bat_count)
##
## Data: bat_data$bat_count (282 obs.); Bandwidth 'bw' = 0.3857
##
##        x                y
##  Min.   :-0.9596   Min.   :4.222e-05
##  1st Qu.: 1.3333   1st Qu.:1.281e-02
##  Median : 3.6262   Median :7.807e-02
##  Mean   : 3.6262   Mean   :1.089e-01
##  3rd Qu.: 5.9191   3rd Qu.:1.986e-01
##  Max.   : 8.2119   Max.   :3.147e-01
```

1d. Estimate the density using KDE. Report the bandwidth you chose and produce a plot of the estimated density.

```
#kde function to estimate bandwidth selection
f.kde = kde(bat_data$bat_count)

#default bandwidth
bw.nrd0(bat_data$bat_count)
```

```
## [1] 0.3856844
```

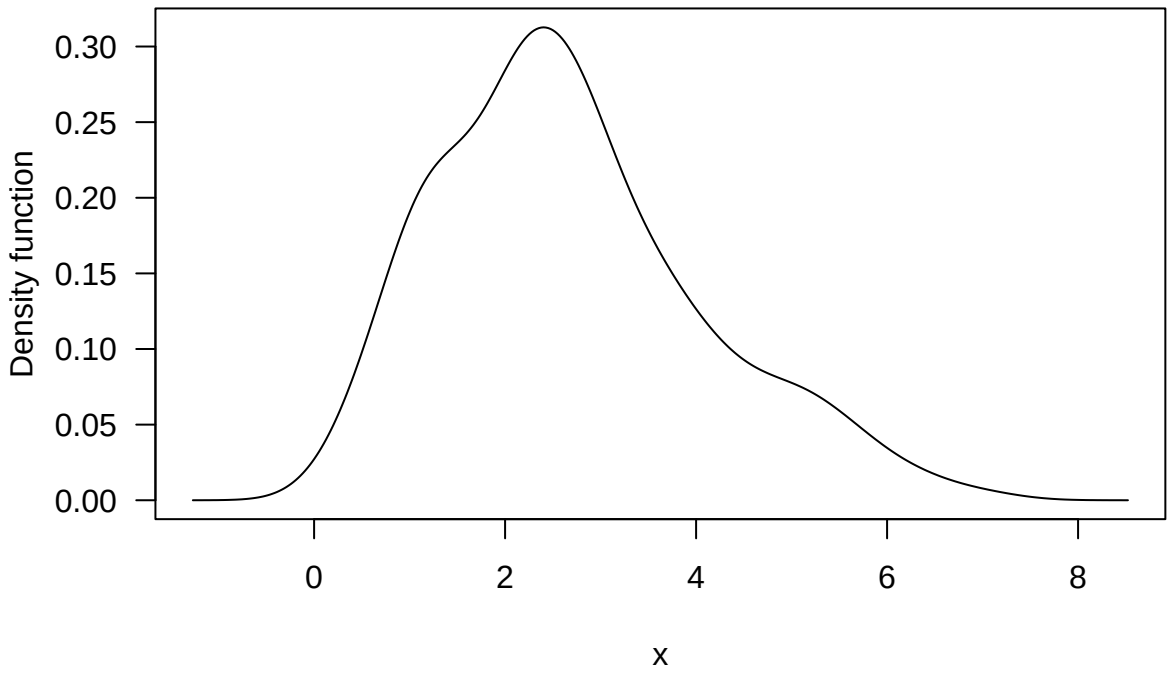
```
#multiple bandwidth selection options
c(bw.nrd0 = bw.nrd0(bat_data$bat_count), bw.nrd=bw.nrd(bat_data$bat_count), bw.bcv=bw.bcv(bat_data$bat_
bw.SJ = bw.SJ(bat_data$bat_count), bw.ucv=bw.ucv(bat_data$bat_count))
```

```
##   bw.nrd0    bw.nrd    bw.bcv     bw.SJ    bw.ucv
## 0.3856844 0.4542505 0.4930675 0.3832301 0.3572091
```

```
#kde function for bandwidth selection
f.kde$h
```

```
## [1] 0.3965503
```

```
#mile marker density plot
plot(f.kde, las = 1)
```



```
#bandwidth selected
f.kde$h
```

```
## [1] 0.3965503
```

Final density of 0.3966.

1e. Which model do you prefer, the parametric or KDE?

The KDE model provides a smoother view of the data which, upon eye balling quickly, provides and easier estimate to gauge density for an observation of X. If I am to be more specific in my result though, parametric would likely be the plot of choice.

Problem 2: Interstate Crash Density

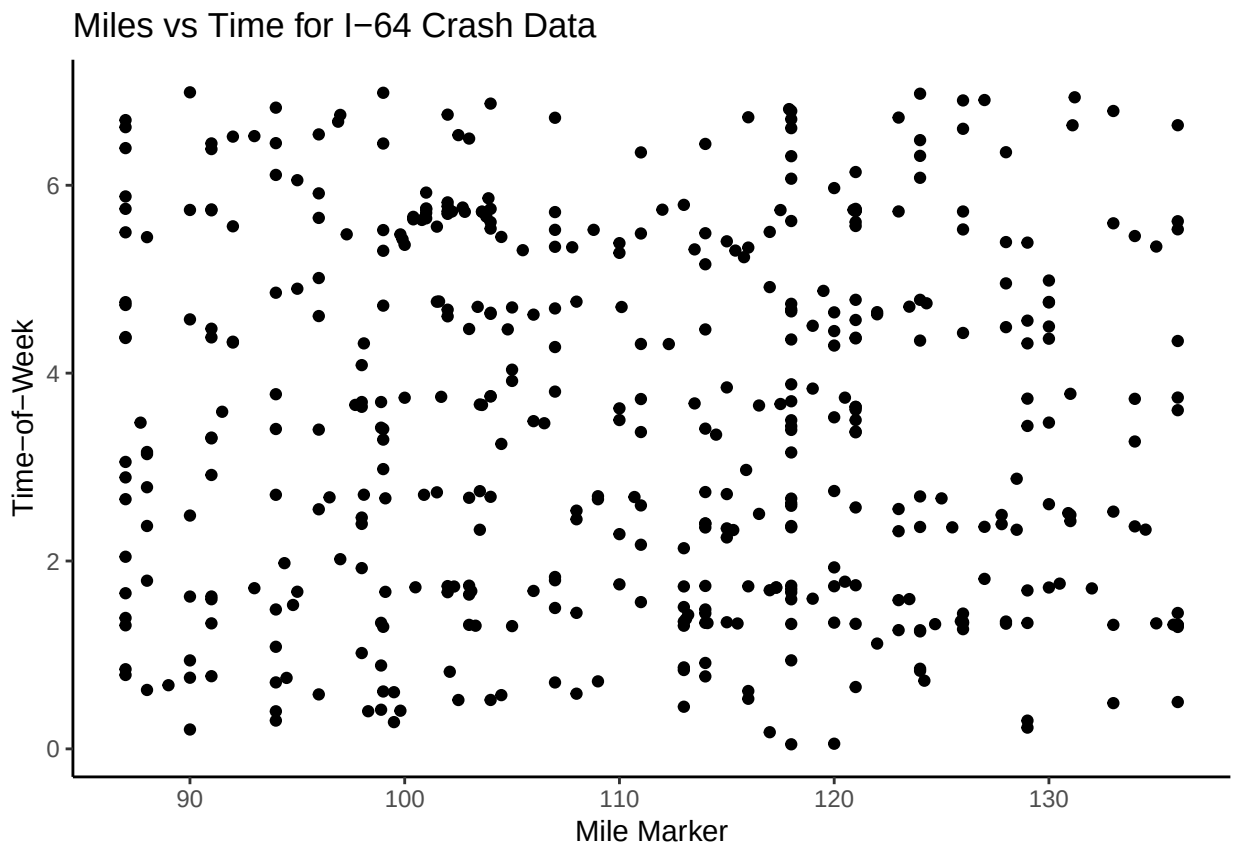
Interstate 64 (I-64) is a major east-west road that passes just south of Charlottesville. Where and when are the most dangerous places/times to be on I-64? The crash data (link below) gives the mile marker and fractional time-of-week for crashes that occurred on I-64 between mile marker 87 and 136 in 2016. The time-of-week data takes a numeric value of .<hour/24>, where the dow starts at 0 for Sunday (6 for Sat) and the decimal gives the time of day information. Thus time=0.0417 corresponds to Sun at 1am and time=6.5 corresponds to Sat at noon.

Load data for R

```
#load data
crash_data <- read.csv("C:\\Users\\brwil\\Desktop\\SY MSDS\\DS 6030 Stat Learning\\Week 6\\crashes16.csv
```

2a. Extract the crashes and make a scatter plot with mile marker on x-axis and time on y-axis.

```
ggplot(crash_data, aes(x=mile, y=time)) +
  geom_point() +
  labs(title="Miles vs Time for I-64 Crash Data",
       x="Mile Marker", y = "Time-of-Week")+
  theme_classic()
```



2b. Use KDE to estimate the mile marker density.

Report the bandwidth. Plot the density estimate.

```
#kde function to estimate bandwidth selection
f_kde_m = kde(crash_data$mile)

#default bandwidth
bw.nrd0(crash_data$mile)
```

```
## [1] 3.589419
```

```
#multiple bandwidth selection options
c(bw.nrd0 = bw.nrd0(crash_data$mile), bw.nrd=bw.nrd(crash_data$mile), bw.bcv=bw.bcv(crash_data$mile),
bw.SJ = bw.SJ(crash_data$mile), bw.ucv=bw.ucv(crash_data$mile))
```

```
## Warning in bw.bcv(crash_data$mile): minimum occurred at one end of the range
```

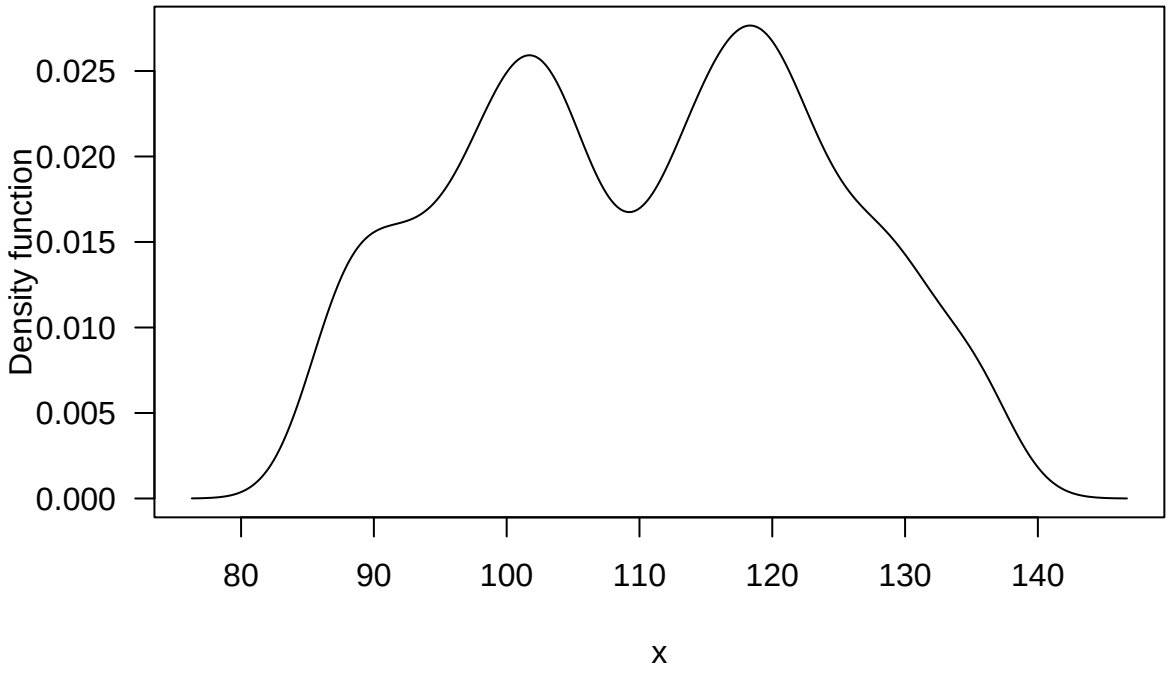
```
## Warning in bw.ucv(crash_data$mile): minimum occurred at one end of the range
```

```
##   bw.nrd0    bw.nrd    bw.bcv     bw.SJ    bw.ucv
## 3.5894186 4.2275374 4.5423349 2.3902272 0.4744332
```

```
#kde function for bandwidth selection
f_kde_m$h
```

```
## [1] 2.894092
```

```
#mile marker density plot
plot(f_kde_m, las = 1)
```




```
#bandwidth selected
f_kde_m$h
```

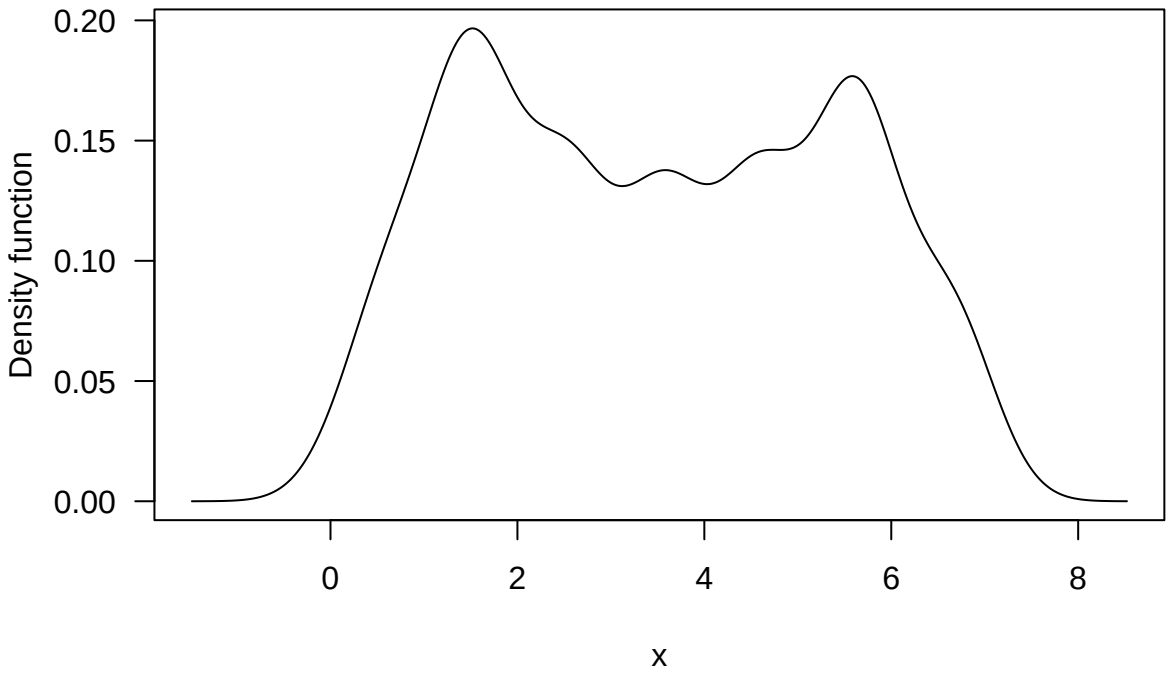
```
## [1] 2.894092
```

2c. Use KDE to estimate the temporal time-of-week density.

Report the bandwidth. Plot the density estimate.

```
#kde function to estimate bandwidth selection
f_kde_t = kde(crash_data$time)

#mile marker density plot
plot(f_kde_t, las = 1)
```




```
#bandwidth selected
f_kde_t$h
```

```
## [1] 0.414127
```

Bandwidth selected of 0.414127.

2d. Use KDE to estimate the bivariate mile-time density.

Report the bandwidth parameters. Plot the bivariate density estimate.

```
#save crash data as variable
X = crash_data

#smoothed cross-validation bw estimator
(H1 = Hscv(X))
```

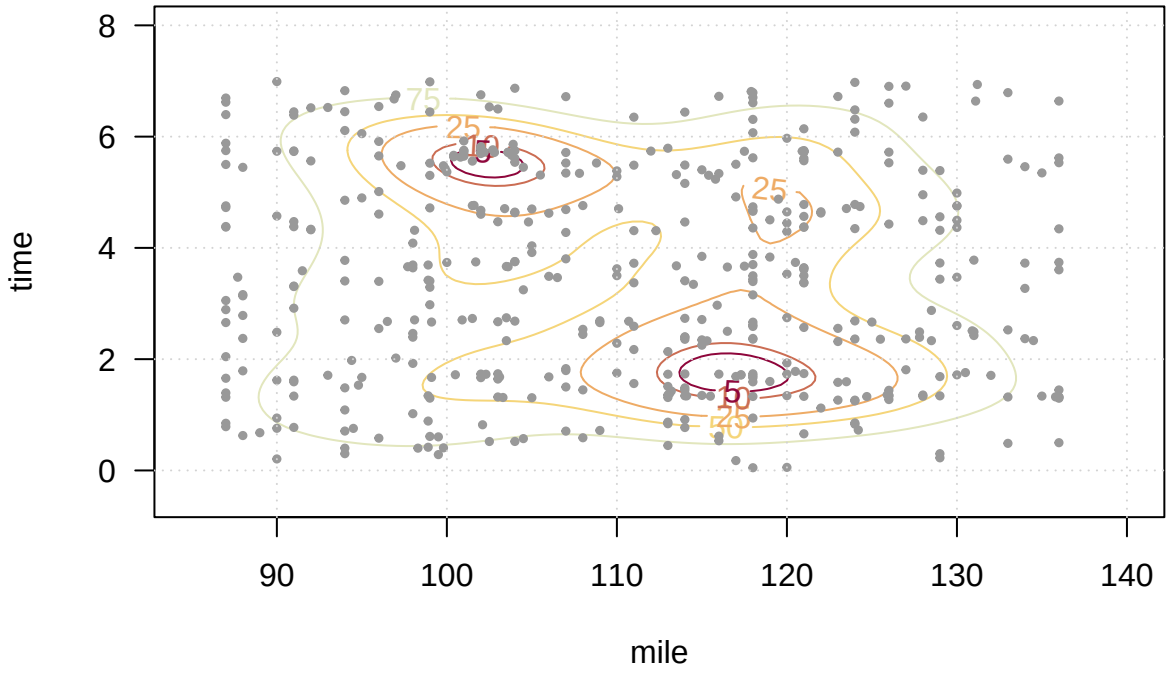
```
##             [,1]       [,2]
## [1,] 25.9921201 -0.1523207
## [2,] -0.1523207  0.3660496
```

```
#use H for multivariate data
f1 = kde(X, H = H1)
```

```
plot(f1,
     cont = c(5, 10, 25, 50, 75), #set contour levels
     las = 1,
     xlim = c(85.0, 140.0), #set x limit based on data
     ylim = c(-0.5, 8)) #set y limit based on data

#add points
points(X, pch = 19, cex = 0.5, col = 'grey60')

#add grid lines
grid()
```



2e. Based on the estimated density, approximate the most dangerous place and time to drive on this strech of road. Identify the mile marker and time-of-week pair.

Based on the bivariate mile-time density above, the two most dangerous places are roughly mile marker 103 at time 5.5 and mile marker 119 at time 1.8. Given that the mile marker 103 at time 5.5 has slightly more observations so likely a greater level of danger.