General

Mingze Li 300137754

2025-02-02

```
library(mvtnorm)
library(traveltimeCLT)
library(data.table)
```

```
## Warning:   'data.table' R 4.3.3
```

```
library(dplyr)
```

```
##
##     'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
trips = fread('data/trips.csv')

trips[, whour :=as.POSIXlt(time)$wday*24 + as.POSIXlt(time)$hour]
trips[, time := as.POSIXct(time)]
trips[, duration_secs := as.numeric(difftime(shift(time, type = "lead"), time, units = "secs")), by = ti
trips[, speed := length/duration_secs]

# pick the trip indecs that has at least one edge speed fast enough.
r = trips[3.6*exp(logspeed)>150][, trip[1],trip][, trip]

# remove the trip that has indexs in r. Then order the remaining observations
# base on trip and time.
trips= trips[!trip %in% r][order(trip, time)]
```

```
## removing last observation
trips = trips[,.SD[-.N] , trip]
trips = trips[order(trip, time)]
set.seed(1)
dependent_uniform<-function(n, rho=0.31) {
    S <-diag(n)
    for (i in 1:n) {
        for (j in 2:n) {
            S[i, j] <- rho^(abs(i-j))
    }
}
    S = S +t(S)
    diag(S)<-1
    St = 2 * sin(S * pi/6) # must be positive definite
    U = c(pnorm(rmvnorm(1, sigma = St)))
    U
}
U = dependent_uniform(5, 0.3)
acf(U, lag.max=2, plot=FALSE)
```

```
##
## Autocorrelations of series 'U', by lag
##
##      0      1      2
##  1.000  0.310 -0.116
```

```
trips<-data.frame(trips)
names(trips)[8]="linkID"
unique(trips$timeBin)
```

```
## [1] "Other" "ER"    "MR"
```

```
length(unique(trips$linkID))
```

```
## [1] 41045
```

```
timebins <- unique(trips$timeBin)
num_timebins <- length(timebins)
timebin_x_edge <-trips %>%
  arrange(timeBins,linkID ) %>%
  mutate(timebin_x_edge = (match(timeBins, unique(timeBins)) - 1) * length(linkID) + match(linkID, link
timebin_x_edge <- timebin_x_edge %>%
  mutate(timebin_x_edge_continuous = dense_rank(timebin_x_edge))
```

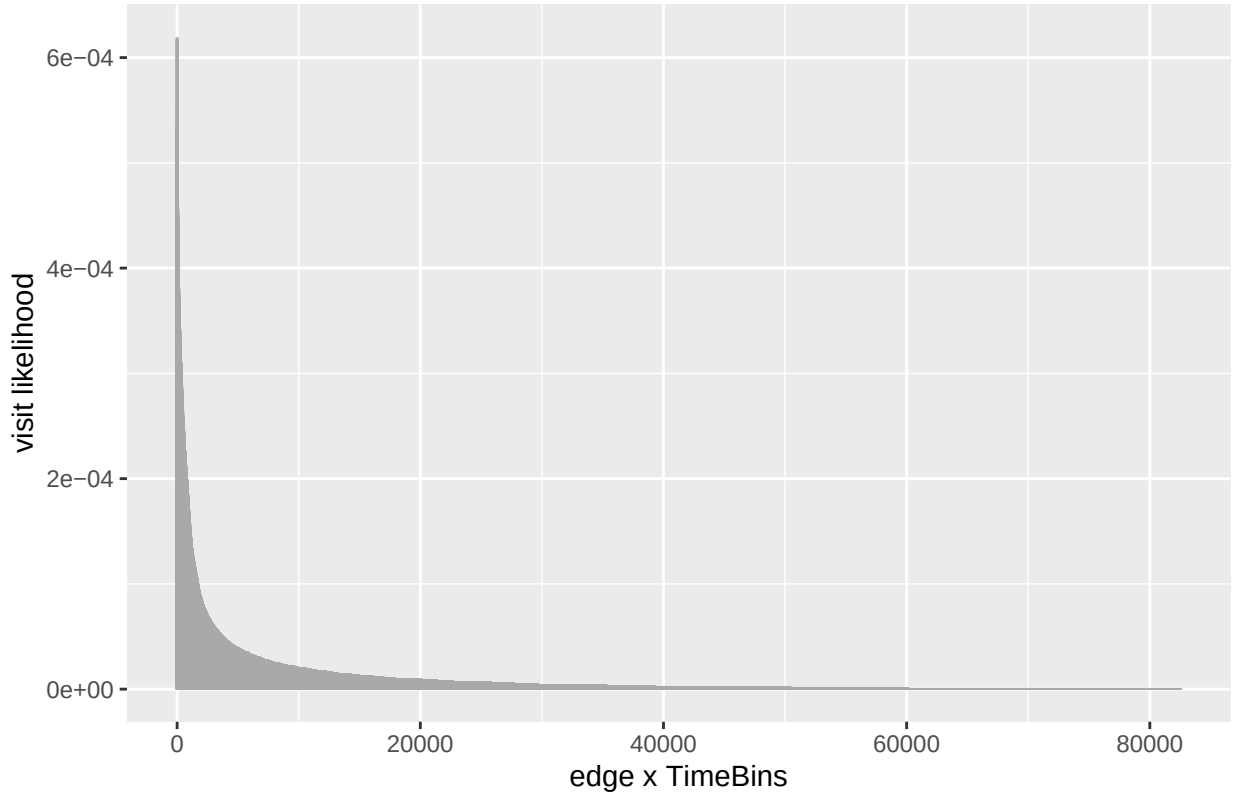
```
timebin_x_edge_sorted <- timebin_x_edge %>%
  count(timebin_x_edge_continuous) %>%
  mutate(density = n / sum(n)) %>%
  arrange(desc(density))
timebin_x_edge_sorted$n <- seq(1,length(timebin_x_edge_sorted$n),1)
sd_na_is_0<-function(x){
  if(length(x)>=2)return(sd(x))
```

```
  else return(0)
}
speed_statistic <- timebin_x_edge %>%
  group_by(timebin_x_edge_continuous) %>%
  mutate(logspeed = log(duration_secs))%>%
  summarise(    mean_log_duration = mean(logspeed),
    sd_log_duration = sd_na_is_0(logspeed),
    mean_duration = mean(duration_secs),
    sd_duration = sd_na_is_0(duration_secs),
    frequency = length(logspeed),
    ave_speed = mean(speed),
    sd_speed = sd_na_is_0(speed))
timebin_x_edge_sorted <- timebin_x_edge_sorted %>%
  left_join(speed_statistic)
```

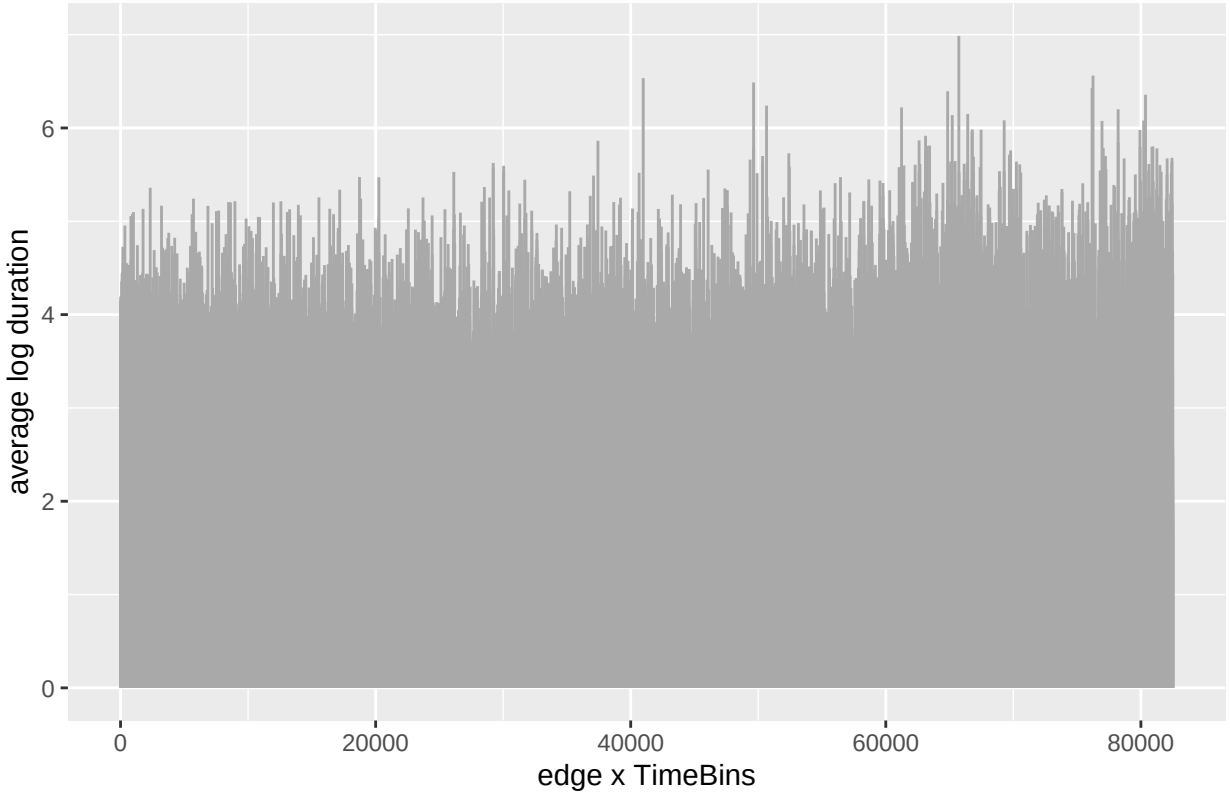
```
## Joining with `by = join_by(timebin_x_edge_continuous)`
```

```
fwrite(timebin_x_edge_sorted,"data/timebin_x_edge_sorted.csv")
fwrite(timebin_x_edge,"data/timebin_x_edge.csv")
```

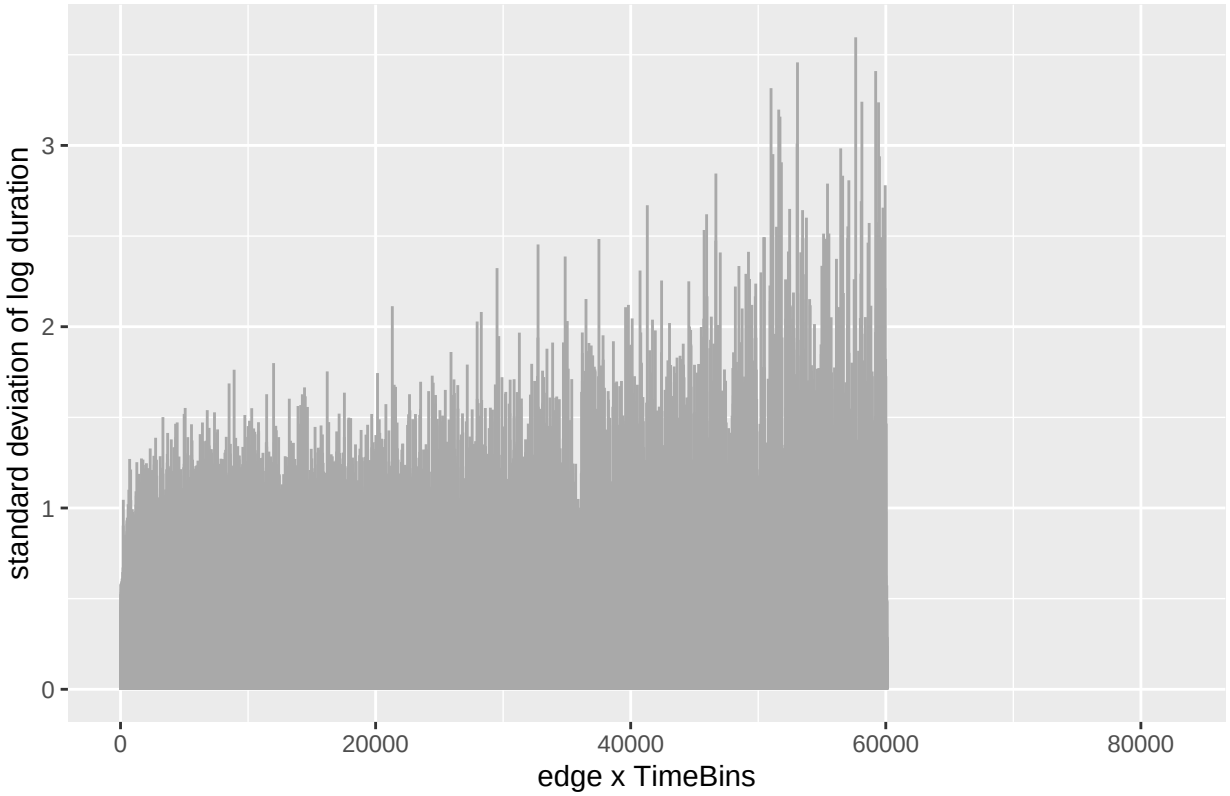
```
ggplot(timebin_x_edge_sorted,aes(x = n, y = density)) +
  geom_col( color = "darkgrey") +
  labs(title = "", x = "edge x TimeBins", y = "visit likelihood")
```



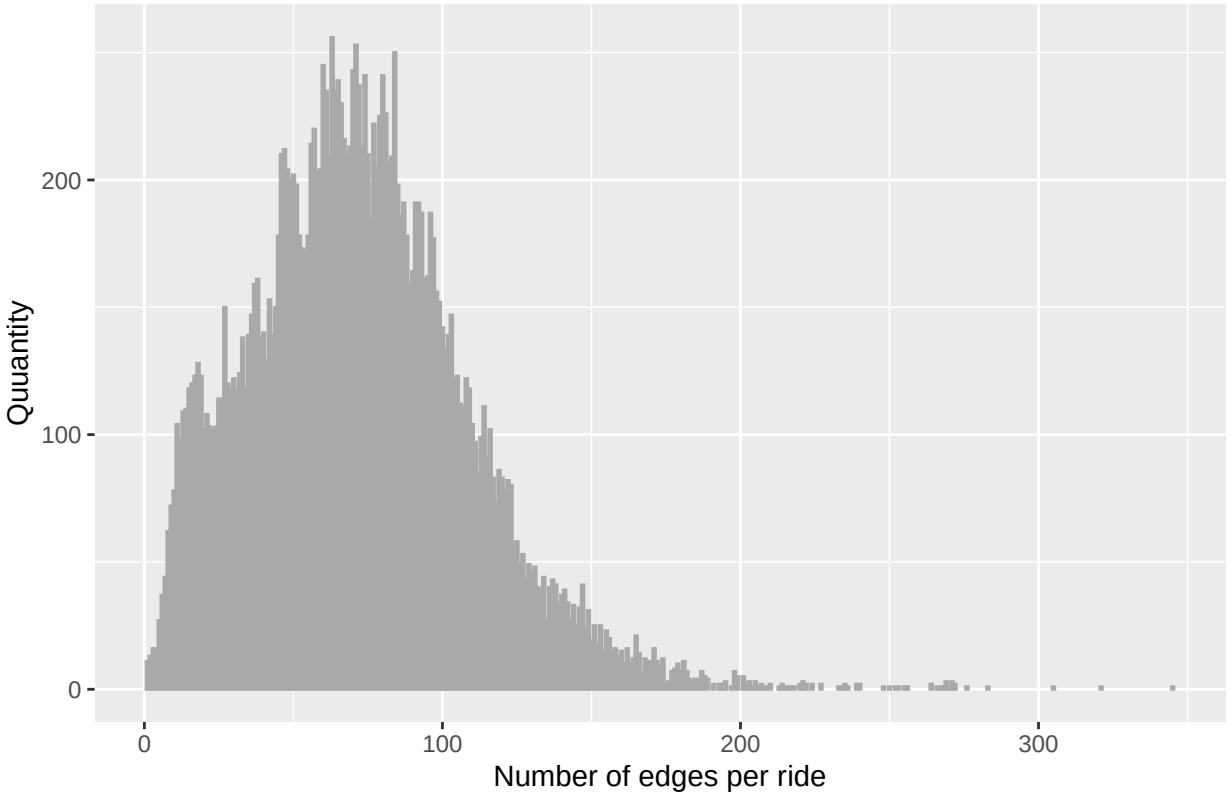
```
ggplot(timebin_x_edge_sorted, aes(n,y = mean_log_duration)) +
  geom_col( color = "darkgrey") +
#  coord_cartesian(ylim = c(0, 40)) +
  labs( title="",x = "edge x TimeBins", y = "average log duration")
```




```
ggplot(timebin_x_edge_sorted, aes(x = n, y = sd_log_duration)) +
   geom_col( color = "darkgrey") +
#    coord_cartesian(ylim = c(0, 25)) +
  labs(title = "", x = "edge x TimeBins", y = "standard deviation of log duration")
```




```
trips_table=data.table(trips)
trip_length = trips_table[,.N,by="trip"]
trip_length_density = trip_length %>%
  count(N) %>%
  mutate(density = n)
ggplot(trip_length_density, aes(x = N, y= density)) +
  geom_bar(stat = "identity" , color = "darkgrey") +
  labs( title="",x = "Number of edges per ride", y = "Quuantity")
```




```
id = sample(unique(timebin_x_edge$trip),1000)
sampled_trips = timebin_x_edge[timebin_x_edge$trip %in% id,]
sampled_trips <- sampled_trips%>% arrange(desc(trip))
sampled_time <- sampled_trips%>%group_by(trip)%>%
  summarise(sampled_time=sum(duration_secs))
log_no_0<-function(x){
  l=length(x)
  result=c()
  for (i in 1:l) {
      if(x[i]==0)result=c(result,0)
      else result=c(result,log(x[i]))
  }
  result
}
time_simulator <- function(edges,rho=0.31){
  l <- length(edges)
  if(l>1)U <- dependent_uniform(l, rho)
  else U<-runif(1)
  mu <- (timebin_x_edge_sorted[match(edges,timebin_x_edge_sorted$timebin_x_edge_continuous), 4])
  sigma <- (timebin_x_edge_sorted[match(edges,timebin_x_edge_sorted$timebin_x_edge_continuous), 5])
  sum(exp(mu + sigma * qnorm(U)))
}
simulated_time <- sampled_trips%>%group_by(trip)%>%
  summarise(simulated_time=time_simulator(timebin_x_edge_continuous))
travel_time <- sampled_time
travel_time$simulated_time <- simulated_time$simulated_time
```

```r
ggplot(travel_time) +
  stat_ecdf(aes(x = sampled_time,color="sampled data")) +
  stat_ecdf(aes(x = simulated_time,color="simulated data")) +
  labs(title = "CDF of Travel Time", x = "Total Travel Time (seconds)", y = "Cumulative Probability")+
  coord_cartesian(xlim = c(0, 4000), ylim = c(0, 1))+
  scale_color_manual(name="Legend",values = c("black","red"))
```