sample route3

Mingze Li 300137754

2025-03-15

```
#source('traveltimeCLTfunctions.R')
library(traveltimeCLT)
library(data.table)
```

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

```
trips <- fread('data/trips.csv')
```

```
id <- sample(unique(trips$trip),1000)
#id <- seq(1:1000)
sampled_1000_trips <- trips[trip %in% id, c("trip", "linkId", "time")]
sampled_1000_trips$timeBin<-time_bins_readable(sampled_1000_trips$time)
sampled_1000_trips<-na.omit(sampled_1000_trips)
sampled_time<-data.table(tripID=unique(sampled_1000_trips$trip),real_time=trips[trip %in%id, .(time[.N]-
sampled_time$real_time<-as.numeric(sampled_time$real_time)
sampled_length<-data.table(tripID=unique(sampled_1000_trips$trip),real_length=trips[trip %in%id, .(sum(l
train = trips[!trips$trip %in% id,]
edge_x_timeBin = get_timeBin_x_edges(train)
```

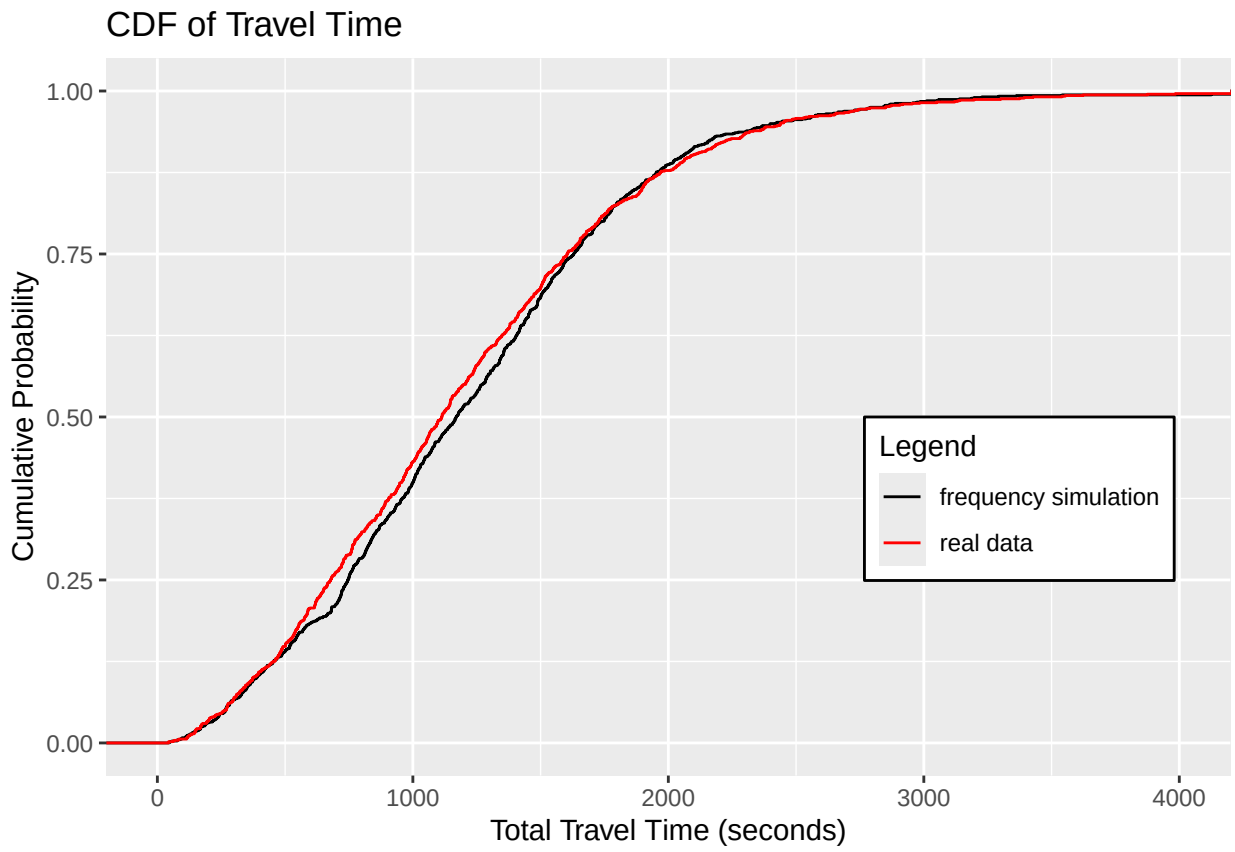
```
observed_edge_num <- sampled_1000_trips[, .(len = .N), by = trip]
simulated_edge_num <- sampled_1000_trips[, .(len = sample(observed_edge_num$len,1,T)), by = trip]
simulated_data <- simulated_edge_num[, .(trip = rep(trip,len)), by = trip]
simulated_data <- simulated_edge_num[, .(trip = rep(trip,len)), by = trip]
```

```
start_times <- sampled_1000_trips[, .(start_time = time[1]), by = trip]
simulated_start_times <- start_times[
  sample(.N, 1000, replace = TRUE), start_time]
simulated_start_times <- time_bins_readable(simulated_start_times)
simulated_data <- simulated_data[,timeBin:=simulated_start_times[which(trip[1]==unique(sampled_1000_trip
simulated_data[,1]=NULL
```

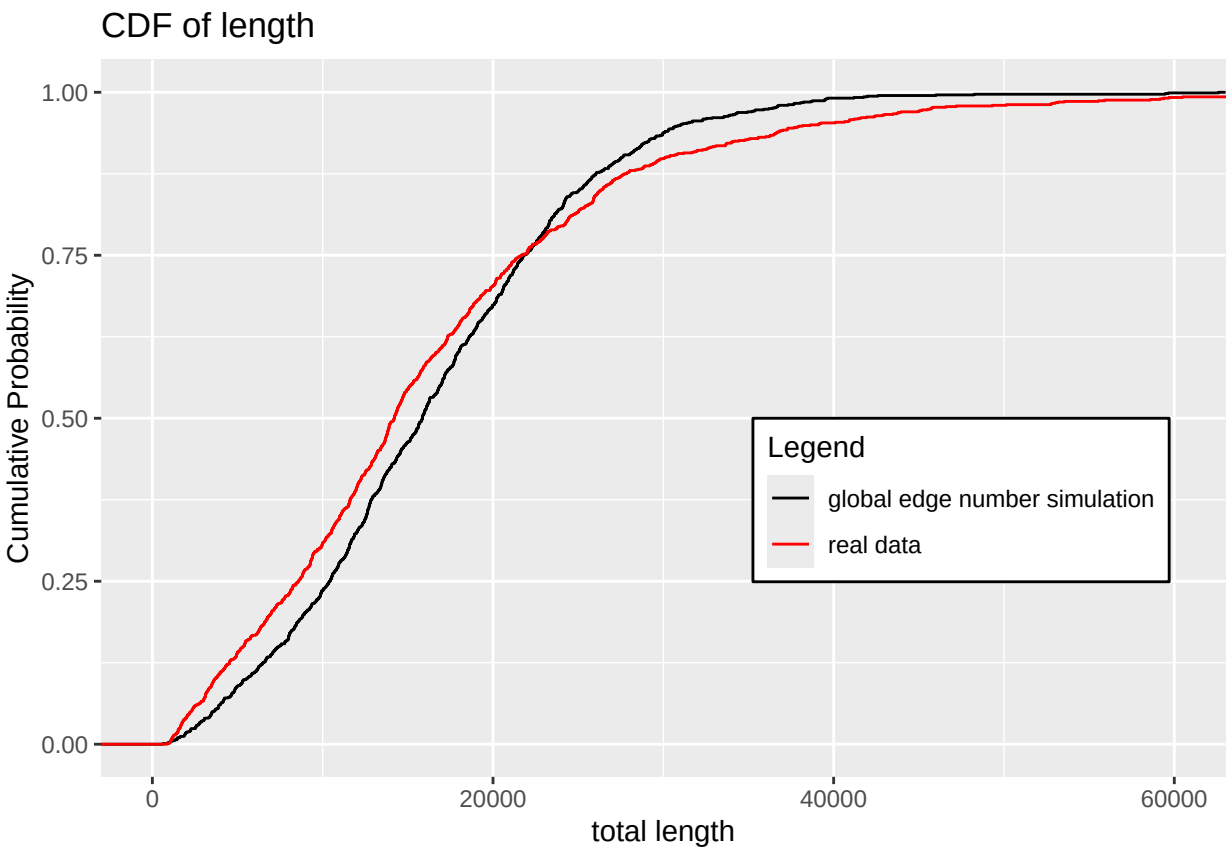
```
simulated_data<-simulated_data[,  sampled_linkId := {
  current_edges <- edge_x_timeBin[timeBin == .BY$timeBin]
  sample(current_edges$linkId, size = .N, prob = current_edges$frequency, replace = TRUE)
}, by = timeBin]


simulated_data <- merge(
  simulated_data[, .(trip, timeBin, sampled_linkId)],
```

```
  edge_x_timeBin[, .(linkId, timeBin, mean, sd, length)],
  by.x = c("sampled_linkId", "timeBin"),
  by.y = c("linkId", "timeBin"),
  all.x = TRUE
)

simulated_result <- simulated_data[, {
  n <- .N
  U <- dependent_uniform(n)
  time_sim <- sum(exp(mean + sd * qnorm(U)))
  length_sim <- sum(length)
  .(sim_time = time_sim, sim_length = length_sim)
}, by = trip]


sampled_time$simulation <- simulated_result[,2]
sampled_length$simulation <- simulated_result[,3]
```

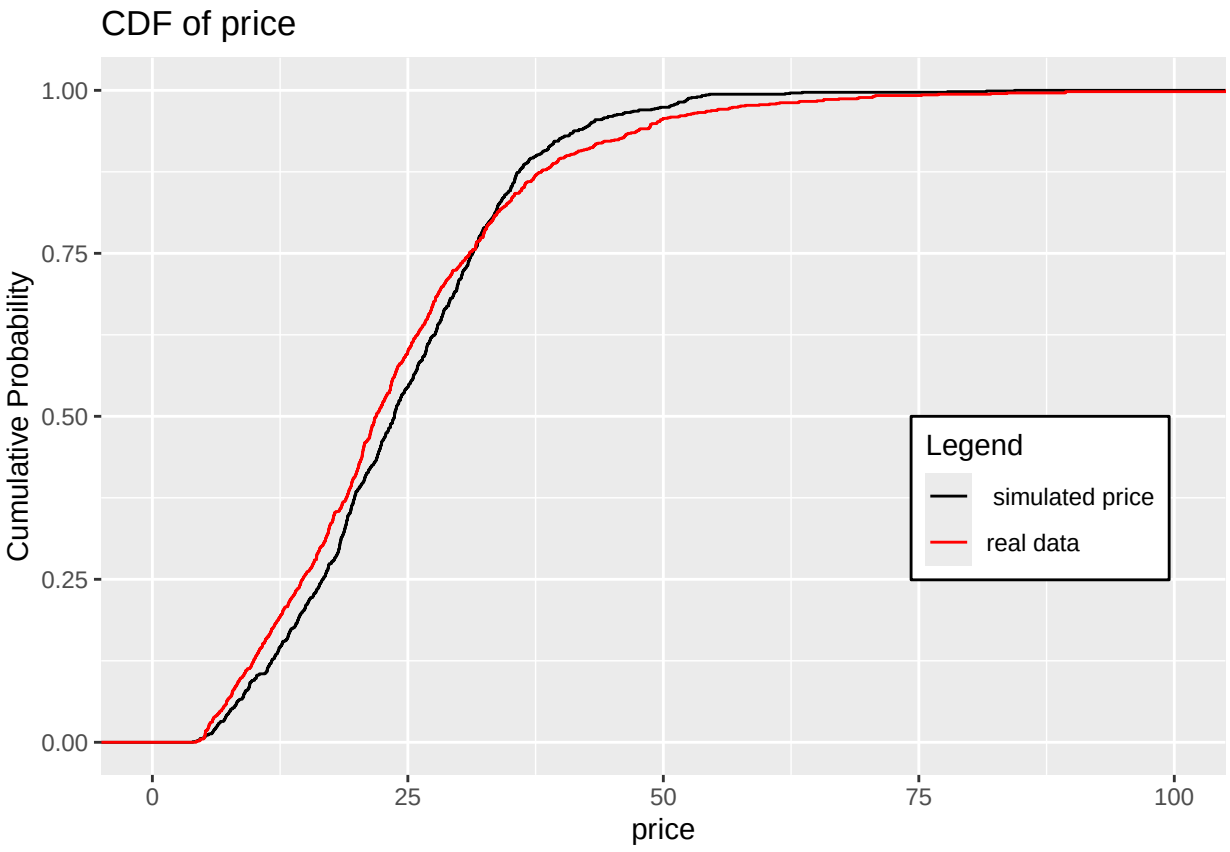
```
plot_CDF_compare(sampled_time$real_time,sampled_time$simulation,"frequency simulation")
```



```
plot_CDF_compare(sampled_length$real_length,sampled_length$simulation,"global edge number simulation","
```




```
sampled_price<-data.table(trip=simulated_result$trip,real_price=price(sampled_time$real_time,sampled_le
plot_CDF_compare(sampled_price$real_price.arrive_price,sampled_price$simulated_price.arrive_price," simu
```




```
names(train)[c(2,3,5,7,8)]=c("tripID","entry_time","duration_secs","distance_meters","linkID")
train$speed=exp(train$logspeed)
train$timeBin=time_bins_readable(train$entry_time)
fit <- traveltimeCLT(train, lag = 1)
```

```
## Warning in traveltimeCLT(train, lag = 1): 4 trips have less than 1 observation,
## and will not be used to estimate autocorrelations, or residual variance
## parameters
```

```
test = trips[trips$trip %in% id,]
names(test)[c(2,3,5,7,8)]=c("tripID","entry_time","1","distance_meters","linkID")
p=predict(fit, test)
fit2 <- traveltimeCLT(train, model = 'population')
p2=predict(fit, test)
```

```
R1=request_R(p,start_times$start_time-300,start_times$start_time,sampled_length$real_length,risk_free=0)
R2=request_R(p2,start_times$start_time-300,start_times$start_time,sampled_length$real_length,risk_free=0)
all(R1==R2)
```

```
## [1] TRUE
```

```
all(p==p2)
```

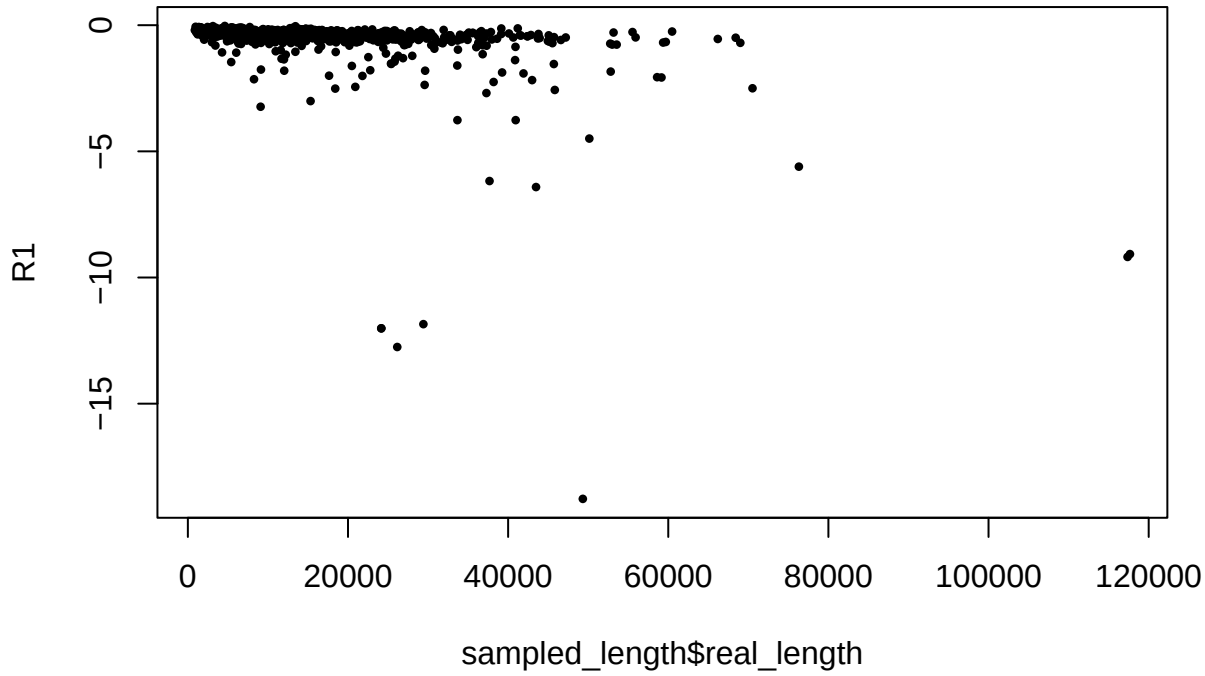
```
## [1] TRUE
```

```r
all(R1<0)
```

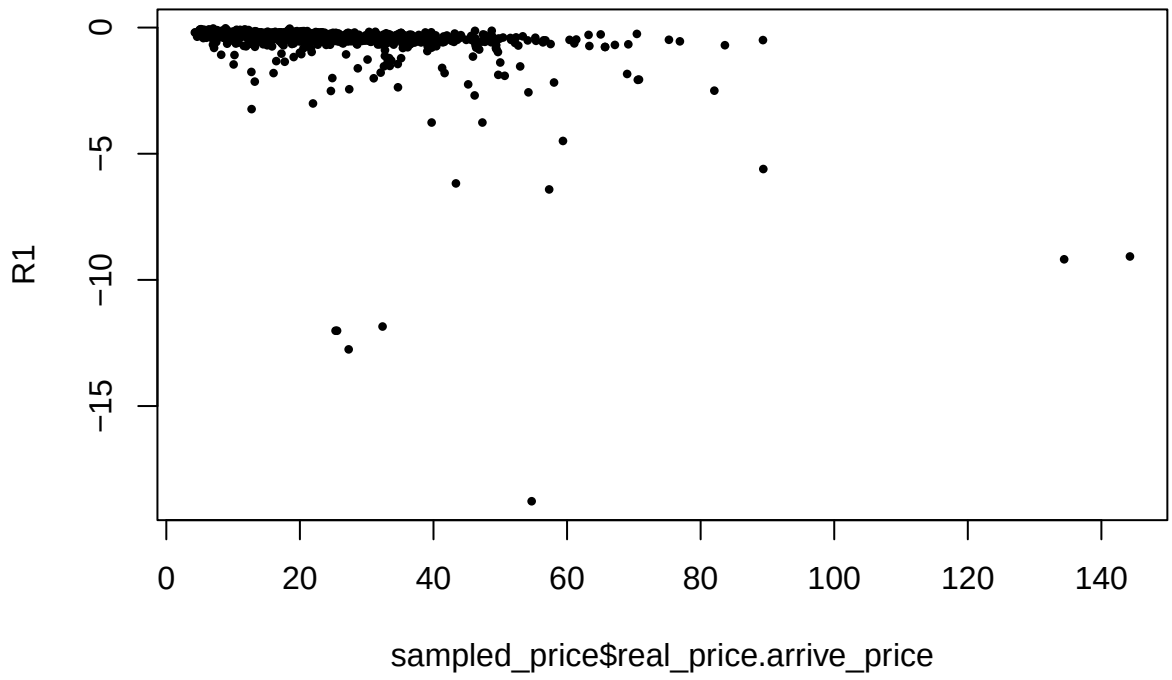
```
## [1] TRUE
```

```r
#all(R1==R2,na.rm = T)
#all(p==p2,na.rm = T)
#which(is.na(p$variance)==T)
```

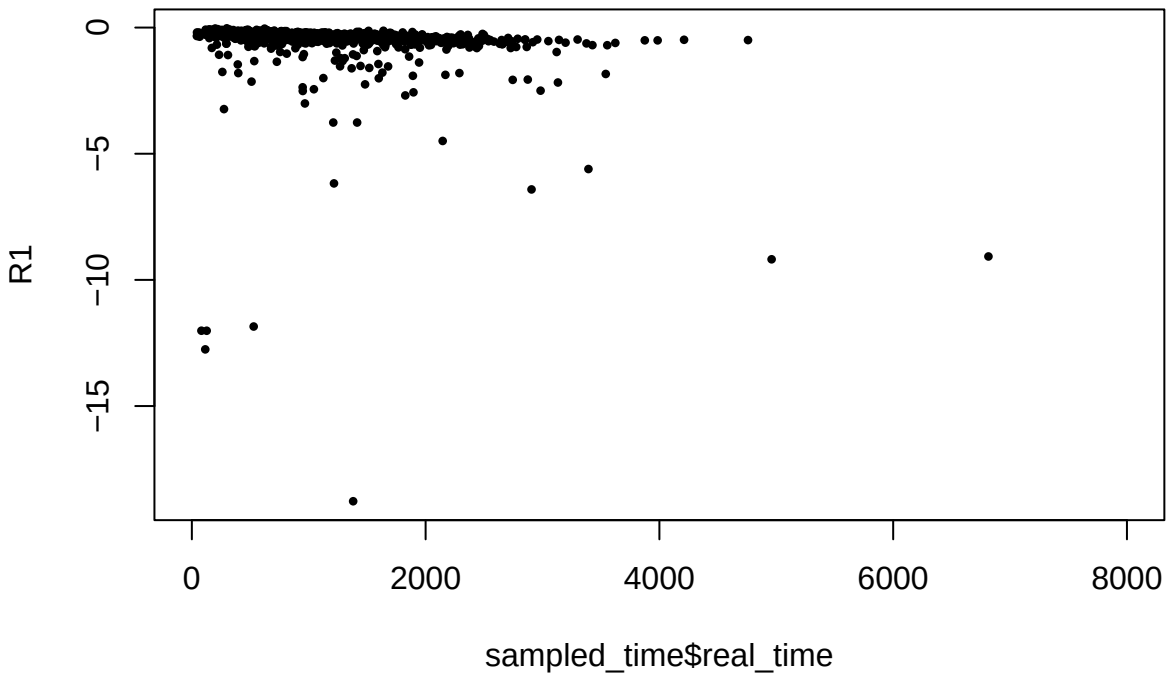
```r
plot(sampled_length$real_length,R1,pch = 16,cex = 0.6)
```



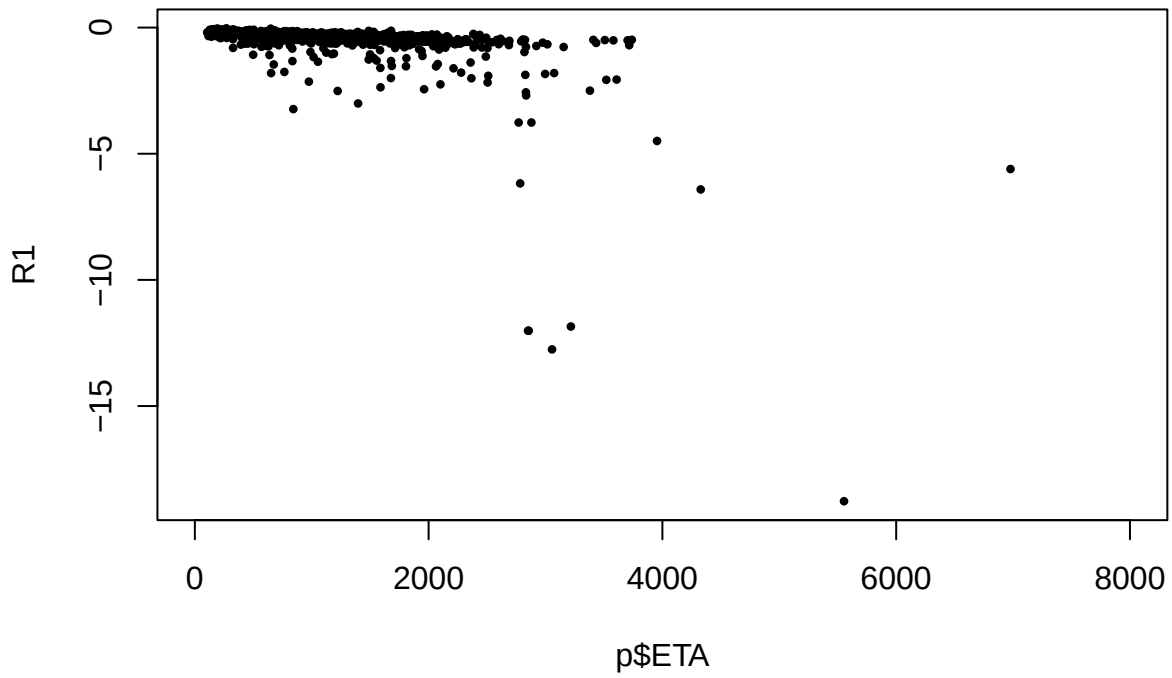

```r
plot(sampled_price$real_price.arrive_price,R1,pch = 16,cex = 0.6)
```



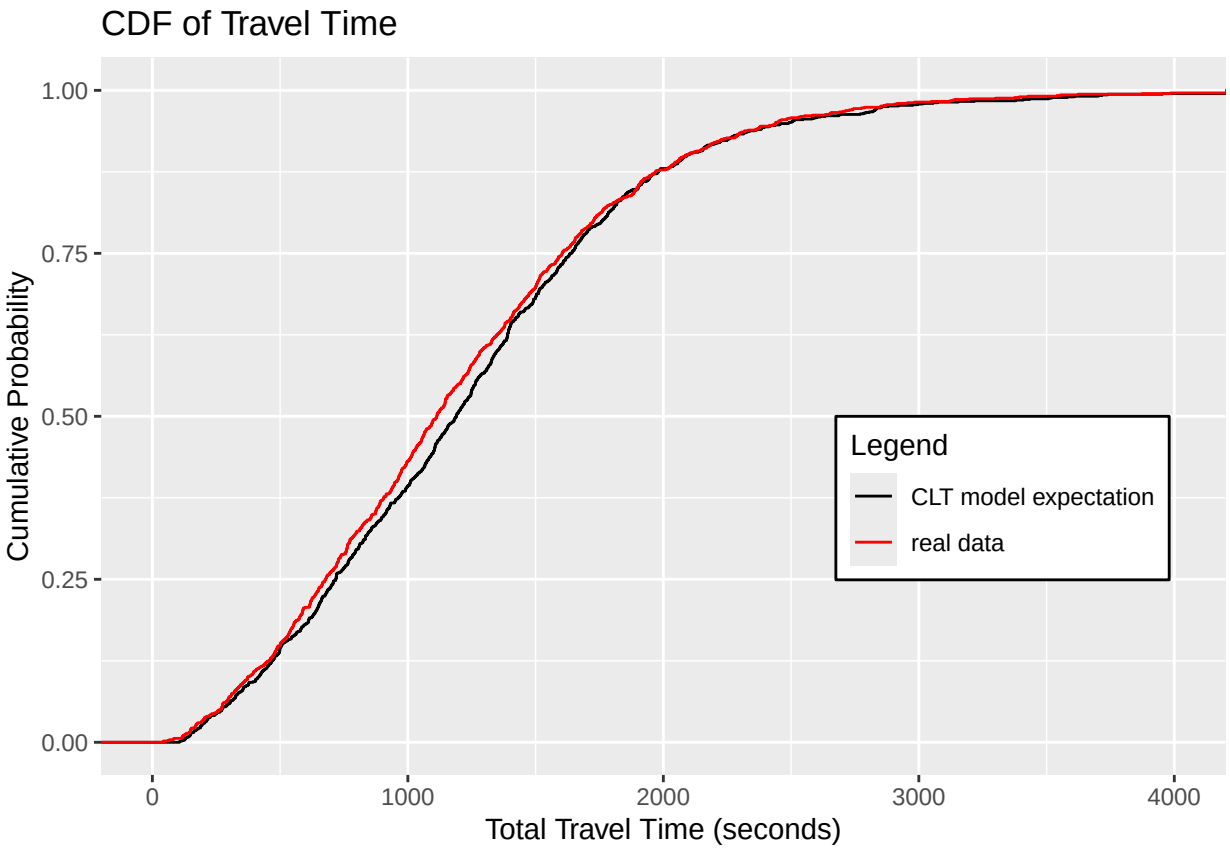

```
plot(sampled_time$real_time,R1,xlim = c(0, 8000),pch = 16,cex = 0.6)
```



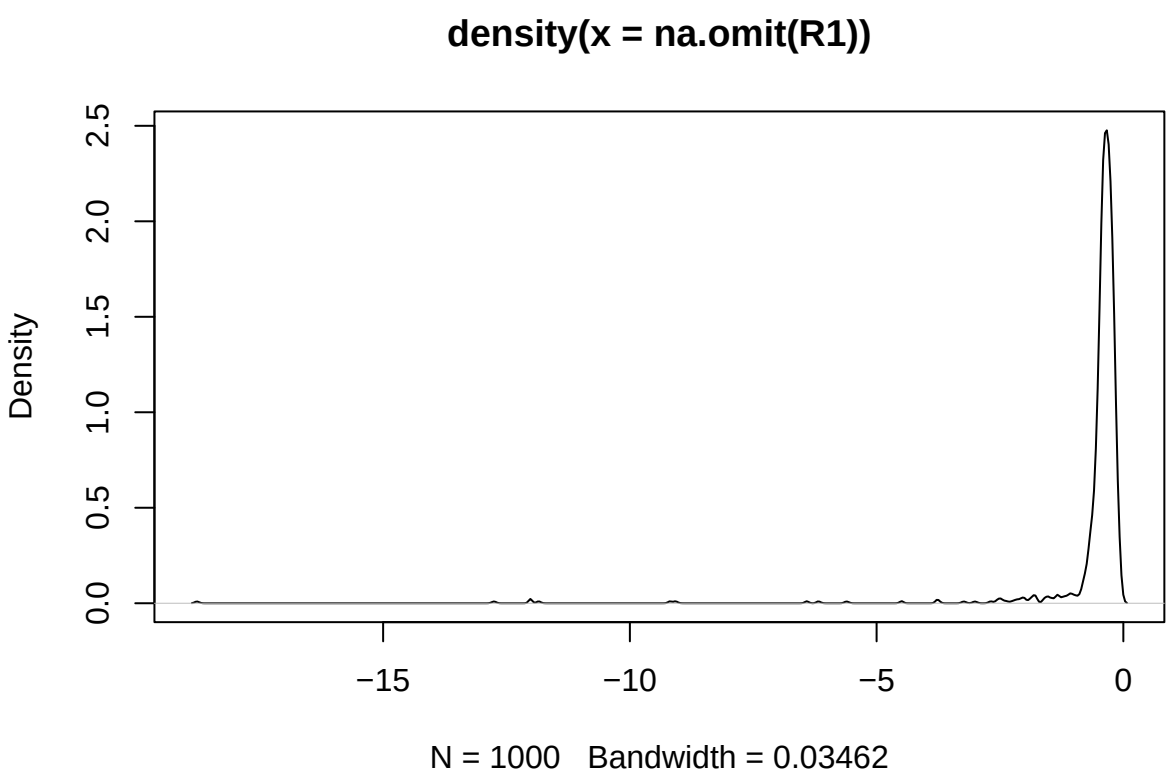

```
plot(p$ETA,R1,xlim = c(0, 8000),pch = 16,cex = 0.6)
```




```
plot_CDF_compare(sampled_time$real_time,p$ETA,"CLT model expectation")
```




```
plot(density(na.omit(R1)))
```


density(x = na.omit(R1))
Density
0.0
0.5
1.0
1.5
2.0
2.5
−15
−10
−5
0
N = 1000 Bandwidth = 0.03462