sample route

Mingze Li 300137754

2025-02-27

```
source('traveltimeCLTfunctions.R')
```

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

## Warning:   'doParallel' R 4.3.3

##      foreach

##      iterators

##      parallel

## Warning:   'igraph' R 4.3.3

##
##    'igraph'

## The following object is masked from 'package:traveltimeCLT':
##
##     time_bins

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union

## Warning:   'tidygraph' R 4.3.3

##
##    'tidygraph'

## The following object is masked from 'package:igraph':
##
##     groups
```

```
## The following object is masked from 'package:stats':
##
##     filter

## Warning:   'ggraph' R 4.3.3

##      ggplot2

## Warning:   'ggplot2' R 4.3.3
```

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

```
id <- sample(unique(trips$trip),1000)
sampled_1000_trips <- trips[trip %in% id, c("trip", "linkId", "time")]
sampled_1000_trips$timeBin<-time_bins_readable(sampled_1000_trips$time)
sampled_1000_trips$time<-NULL
sampled_1000_trips<-merge(sampled_1000_trips, edge_x_timeBin, by = c("linkId", "timeBin"), all.x = TRUE)
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(
```

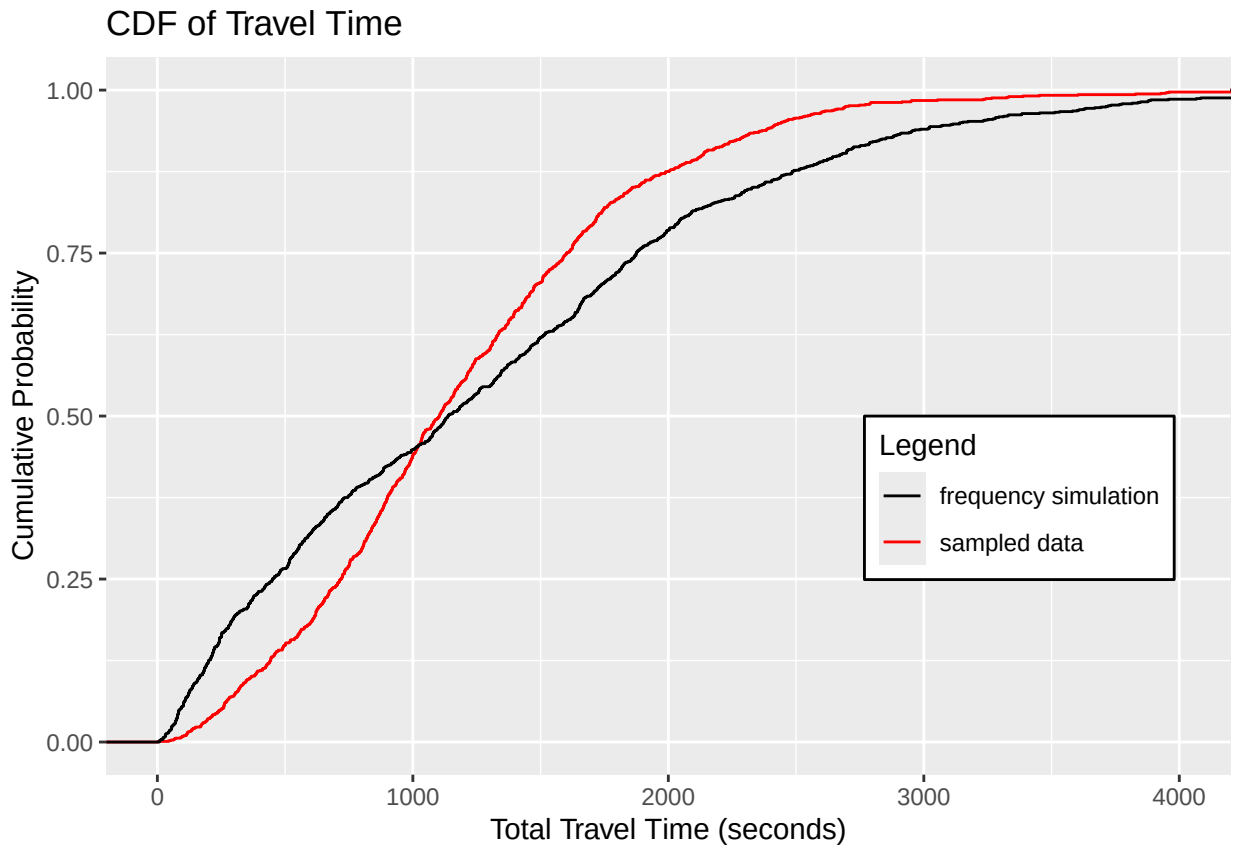
```
sampled_stat <- data.table(tripID=unique(sampled_1000_trips$trip),real_frequency=sampled_1000_trips[, .
```

```
cl <- makeCluster(8)
registerDoParallel(cl)
clusterExport(cl, c("edge_x_timeBin", "sampled_stat", "dependent_uniform"))
simulated_time <- foreach(
  i = 1:1000,
  .combine = c,
  .packages = c("data.table"  ,"mvtnorm")
) %dopar% {
  target_length <- sampled_stat[i, 2]
  target_timeBin <- sampled_stat[i, 3]
  fiction_set <- data.table()
  remaining <- as.numeric(target_length)
  target_timeBin <- as.character(target_timeBin)
  while (remaining > 0) {
    sampleset <- edge_x_timeBin[frequency <= remaining & timeBin == target_timeBin]
    p <- sampleset$frequency
    index<-which(rmultinom(1,1,p)==1)
    fiction_set <- rbind(fiction_set, sampleset[index])
    remaining <- remaining - sampleset[index, frequency]
}
  Mu <- fiction_set$mean
  Sd <- fiction_set$sd
  U <- dependent_uniform(length(Mu))
  sum(exp(Mu + Sd * qnorm(U)))
}
stopCluster(cl)
sampled_time$frequency_simulation <- simulated_time
```

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

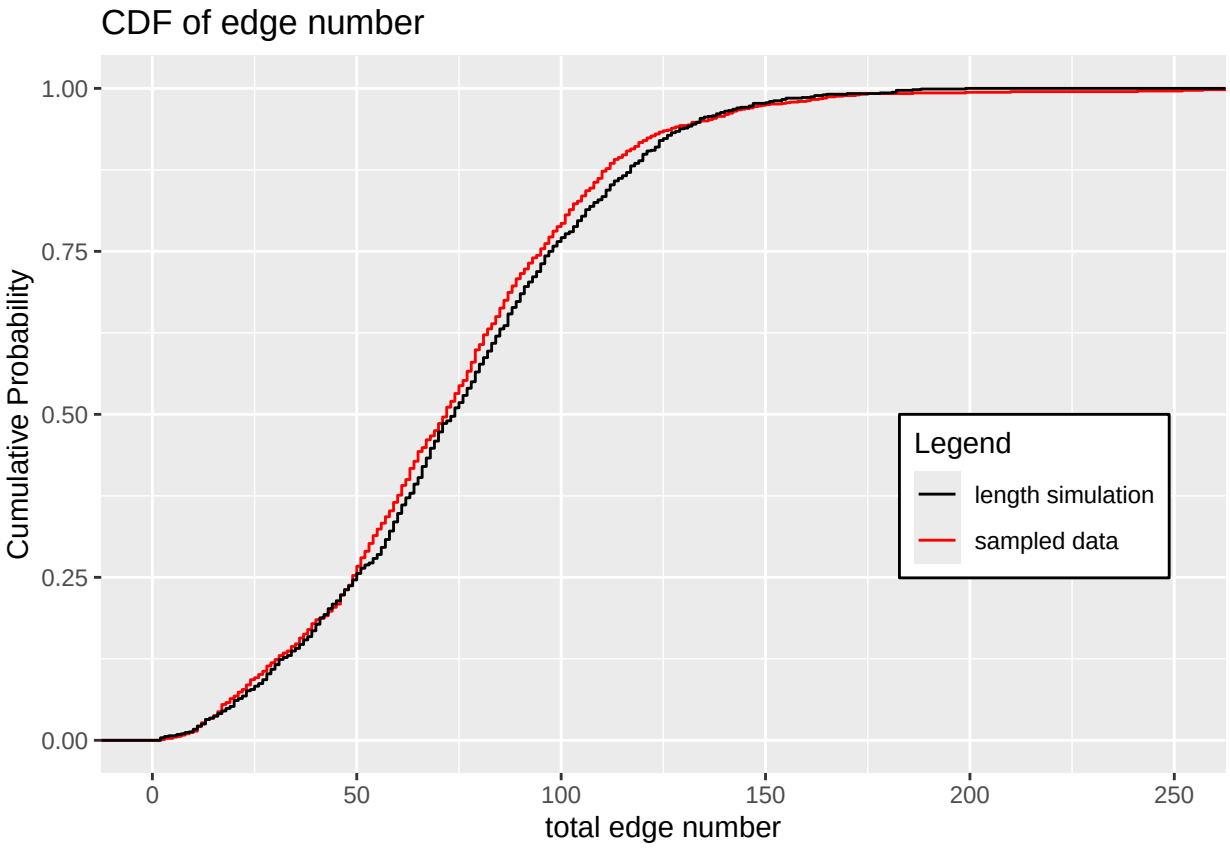
```
## Warning: The `legend.text.align` argument of `theme()` is deprecated as of ggplot2
## 3.5.0.
## i Please use theme(legend.text = element_text(hjust)) instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2
## 3.5.0.
## i Please use the `legend.position.inside` argument of `theme()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```r
edge_num_mean<-mean((sampled_stat$real_edge_num))
edge_num_sd<-sd((sampled_stat$real_edge_num))
min<-min(sampled_stat$real_edge_num)

simulated_edge_num <- sapply(1:1000, function(i) {
  repeat {
    sim_val <- ceiling(((1)*edge_num_mean + edge_num_sd * qnorm(runif(1))))
    if (sim_val >= min) {
```

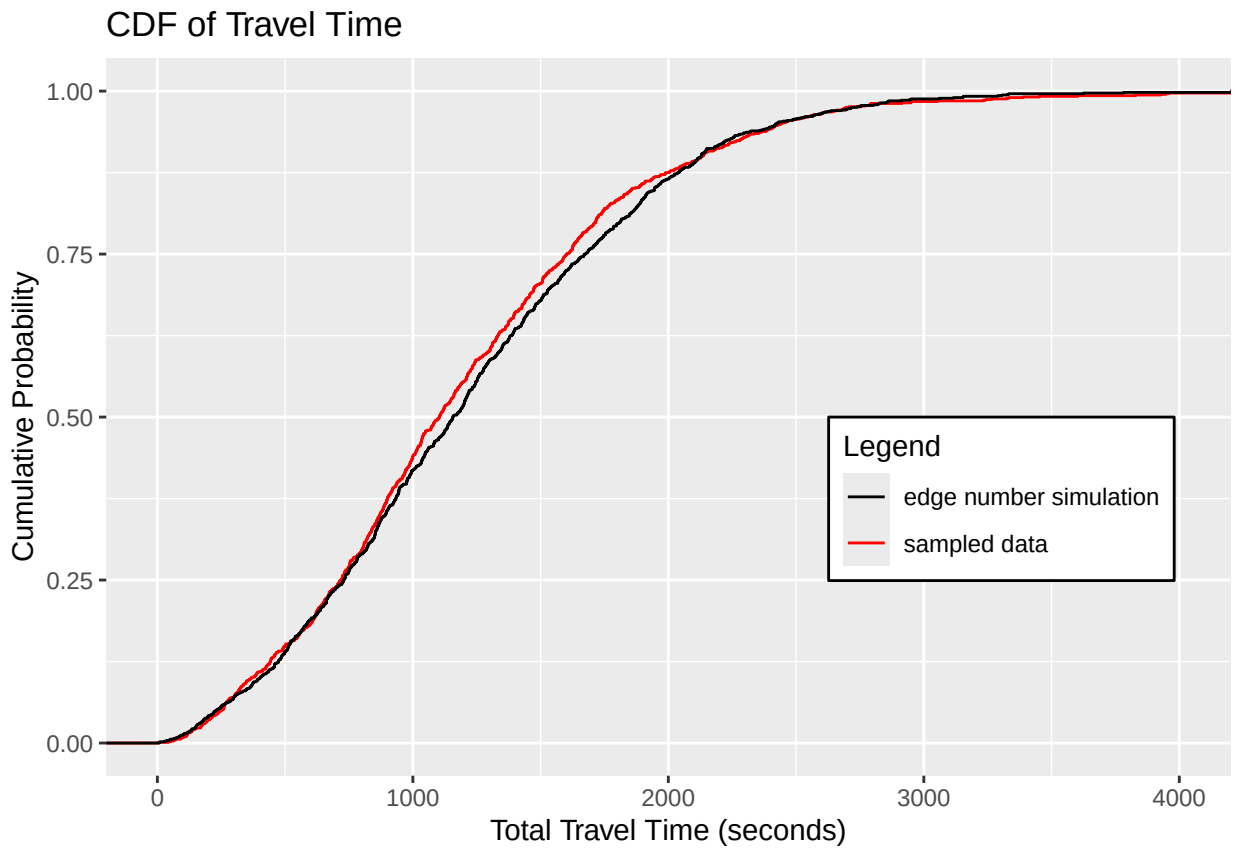
```
      return(sim_val)
    }
  }
})
simulated_edge_num <- data.table(timeBin=sampled_stat$timeBin,simulated_edge_num=simulated_edge_num)
plot_CDF_compare(sampled_stat$real_edge_num,simulated_edge_num$simulated_edge_num,"length simulation","
```



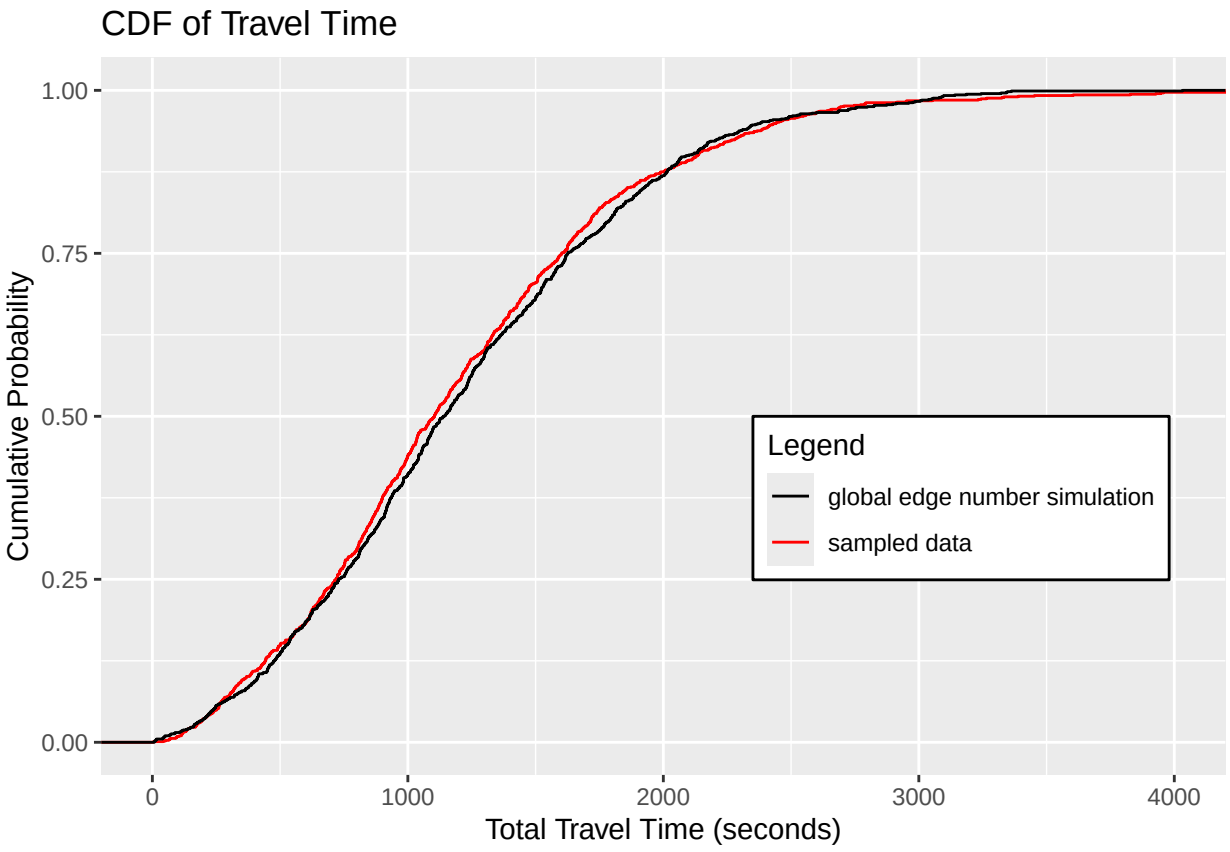

```
trips <- fread('data/trips.csv')
trips$time <- as.POSIXct( trips$time)
trips$timeBin<-time_bins_readable(trips$time)
trips[, duration := as.numeric(difftime(shift(time, type = "lead"), time, units = "secs")), by = trip]
trips[, log_duration := log(duration)]
trips[, length := log(length)]
trips<-na.omit(trips)
timeBin_stat<-trips[, .(mean = mean(log_duration, na.rm = TRUE),
                        sd = sd_one_input_is_0(log_duration),
                        length = mean(length, na.rm = TRUE),
                        length_sd = sd_one_input_is_0(length)),
                        by = .(timeBin)]
global_stat<-trips[, .( timeBin = "Global",
                        mean = mean(log_duration, na.rm = TRUE),
                        sd = sd_one_input_is_0(log_duration),
                        length = mean(length),
                        length_sd = sd_one_input_is_0(length))]
timeBin_stat<-rbind(timeBin_stat,global_stat)
```

```
edge_num_simulation<-c()
for (i in 1:1000) {
  l=simulated_edge_num$simulated_edge_num[i]
  t=simulated_edge_num$timeBin[i]
  stat<-timeBin_stat[timeBin==t,]
  edge_num_simulation[i]<-sum(exp(stat$mean+stat$sd*qnorm(dependent_uniform(l))))
}
sampled_time$edge_num_simulation<-edge_num_simulation
```

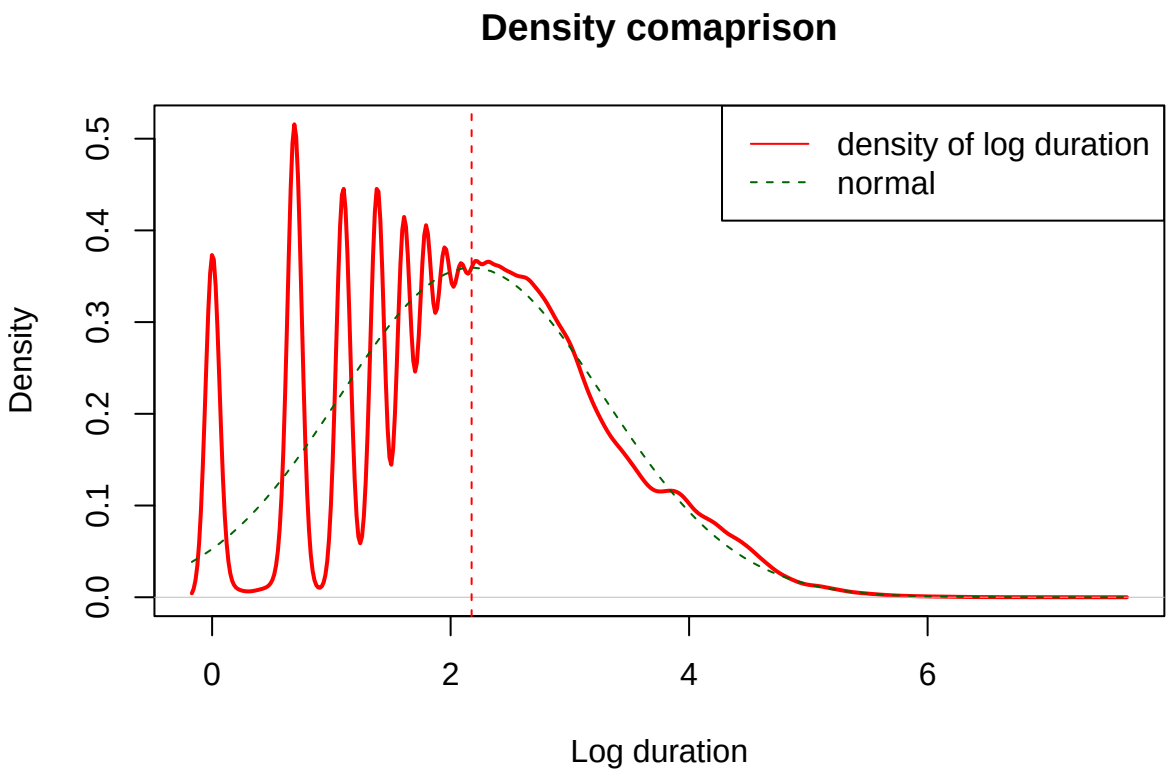
```
plot_CDF_compare(sampled_time$real_time,sampled_time$edge_num_simulation,"edge number simulation")
```



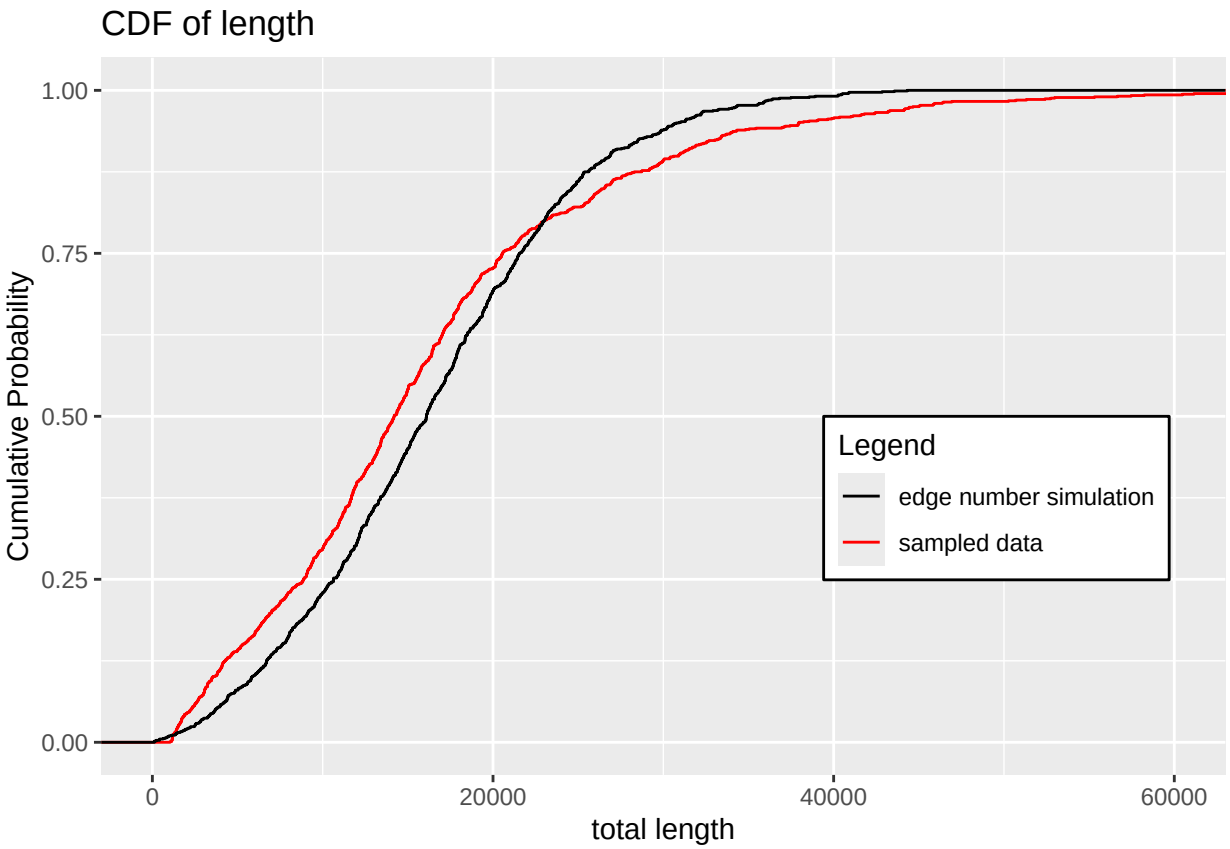

```
global_edge_num_simulation<-c()
for (i in 1:1000) {
  l=simulated_edge_num$simulated_edge_num[i]
  t="Global"
  stat<-timeBin_stat[timeBin==t,]
  global_edge_num_simulation[i]<-sum(exp(stat$mean+stat$sd*qnorm(dependent_uniform(l))))
}
sampled_time$global_edge_num_simulation<-global_edge_num_simulation

plot_CDF_compare(sampled_time$real_time,sampled_time$global_edge_num_simulation,"global edge number simu
```



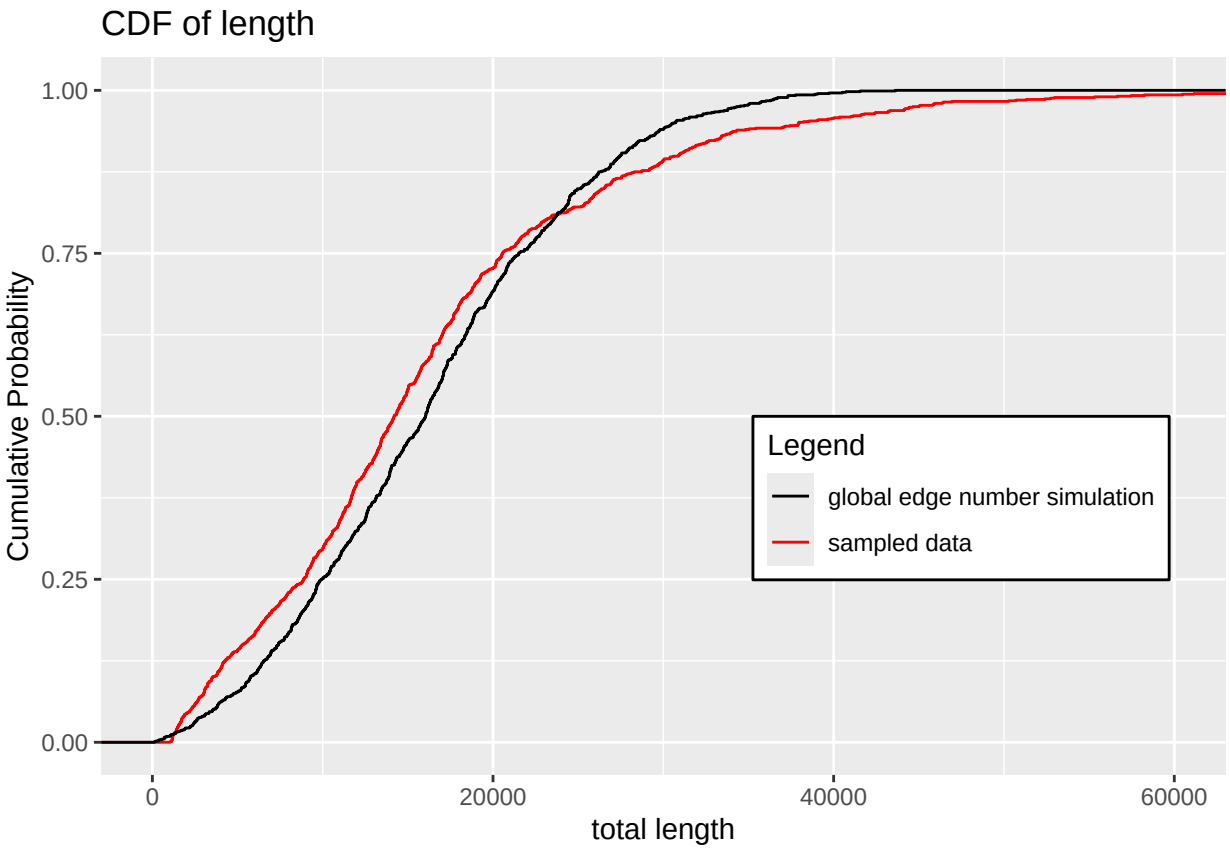

```
plot(density(trips$log_duration), col="red", main="Density comaprison",xlab="Log duration", lwd=2)
curve(dnorm(x, mean=mean(trips$log_duration), sd=sd(trips$log_duration)), add=TRUE, col="darkgreen", lty
abline(v=mean((trips$log_duration)), col="red", lty=2)
legend("topright",legend=c("density of log duration", "normal"), col=c("red", "darkgreen"), lty=1:2)
```



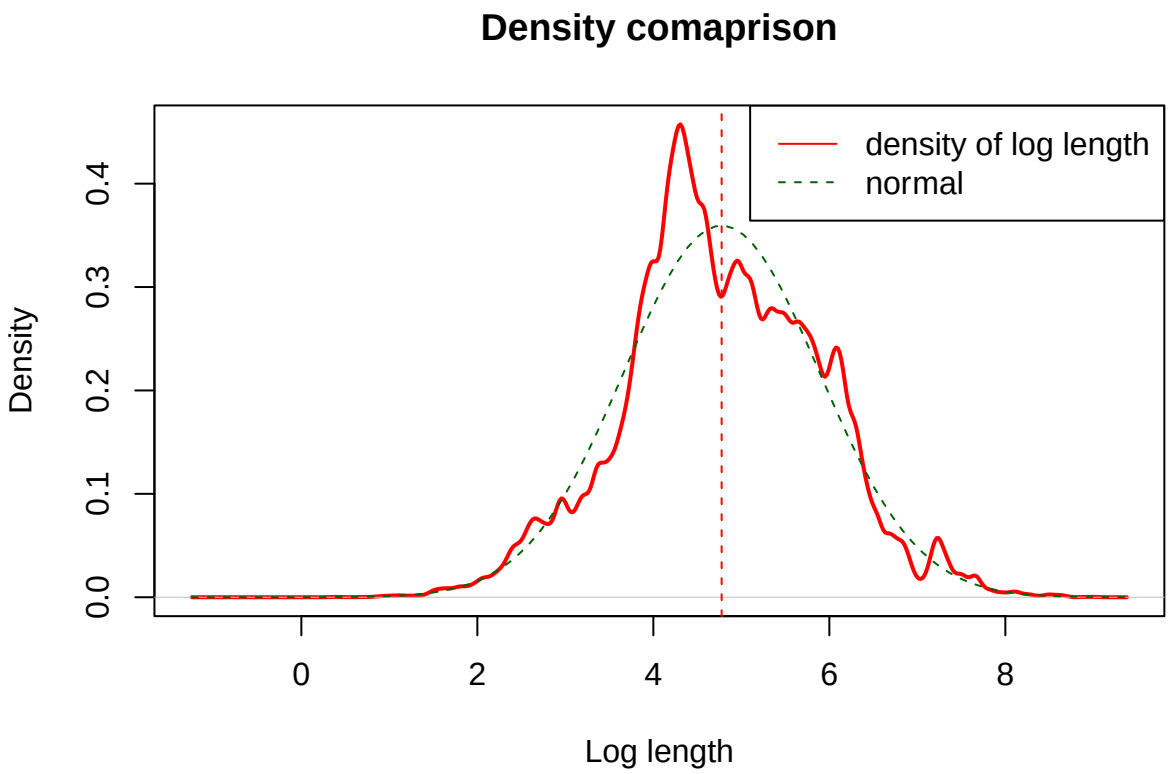

```
edge_num_simulation<-c()
for (i in 1:1000) {
  l=simulated_edge_num$simulated_edge_num[i]
  t=simulated_edge_num$timeBin[i]
  stat<-timeBin_stat[timeBin==t,]
  edge_num_simulation[i]<-sum(exp(stat$length+stat$length_sd*qnorm(runif(l))))
}
sampled_length$edge_num_simulation<-edge_num_simulation

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




```
global_edge_num_simulation<-c()

for (i in 1:1000) {
  l=simulated_edge_num$simulated_edge_num[i]
  t="Global"
  stat<-timeBin_stat[timeBin==t,]
  global_edge_num_simulation[i]<-sum(exp(stat$length+stat$length_sd*qnorm(runif(l))))
}
sampled_length$global_edge_num_simulation<-global_edge_num_simulation

plot_CDF_compare(sampled_length$real_length,sampled_length$global_edge_num_simulation,"global edge numb
```




```
plot(density(trips$length), col="red", main="Density comaprison",xlab="Log length", lwd=2)
curve(dnorm(x, mean=mean(trips$length), sd=sd(trips$length)), add=TRUE, col="darkgreen", lty=2)
abline(v=mean((trips$length)), col="red", lty=2)
legend("topright",legend=c("density of log length", "normal"), col=c("red", "darkgreen"), lty=1:2)
```


Density comaprison
density of log length
normal
Density
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8
Log length