```r
source("/Users/aaronkruchten/Desktop/Data Science Internship/PSC interface github copy.R")


#given a data frame that contains DataOctetsOut, DataSegsOut, OctetsRetrans,SegsRetrans or any subset o
#computes the difference between these if applicable and returns a new frame
#also OctetsRetrans and SegsRetrans are both cumulative. We transform them so that they are not.
no_retransmitted_data <- function(df){
  new_df = data.frame(df)
  if(length(new_df$HCDataOctetsOut) > 0 & length(new_df$OctetsRetrans) > 0){
    new_df$CleanOctetsOut = new_df$HCDataOctetsOut - new_df$OctetsRetrans
  }
  if(length(new_df$DataSegsOut) > 0 & length(new_df$SegsRetrans) > 0){
    new_df$CleanSegsOut = new_df$DataSegsOut - new_df$SegsRetrans
  }
  remove_rows_vector = c()
  remove_rows_index = 1
  #for(i in 1:nrow(new_df)){
   # if((!is.na(new_df$CleanSegsOut[i]) & new_df$CleanSegsOut[i] < 0) | (!is.na(new_df$CleanOctetsOut[i
    #  remove_rows_vector[remove_rows_index] = i
     # remove_rows_index = remove_rows_index + 1
  #  }
  #  if(length(remove_rows_vector) >= 1 ){
   #   new_df = new_df[-remove_rows_vector,]
    #}
  #}
  return(new_df)
}

query_vector <- c("AbruptTimeouts","ActiveOpen","CERcvd","CongAvoid","CongSignals","CountRTT","CurAppRQ
                  "CurRTO","CurReasmQueue","CurRwinRcvd","CurRwinSent","CurSsthresh","CurTimeoutCount
                  "DataSegsIn","DataSegsOut","DupAckEpisodes","DupAcksIn","DupAcksOut","ECESent","ECN
                  "EarlyRetransDelay","ElapsedMicroSecs","ElapsedSecs","EndTime","FastRetran","HCData
                  "HCThruOctetsAcked","HCThruOctetsReceived","InRecovery","IpTosIn","IpTosOut","IpTtl
                  "MaxAppRQueue","MaxAppWQueue","MaxCaCwnd","MaxMSS","MaxPipeSize","MaxRTO","MaxRTT",
                  "MaxSsCwnd","MaxSsthresh","MinMSS","MinRTO","MinRTT","MinSsthresh","Nagle","NonReco
                  "OtherReductions","OtherReductionsCM","PipeSize","PostCongCountRTT","PostCongSumRTT
                  "RTTVar","RcvNxt","RcvRTT","RecInitial","RetranThresh","SACKBlocksRcvd","SACKsRcvd"
                  "SendStall","SlowStart","SmoothedRTT","SndInitial","SndLimTimeCwnd","SndLimTimePace
                  "SndLimTimeTSODefer","SndLimTransCwnd","SndLimTransPace","SndLimTransRwin","SndLimT
                  "SndMax","SndNxt","SndUna","SoftErrorReason","SoftErrors","SpuriousFrDetected","Spu
                  "State","SubsequentTimeouts","SumOctetsReordered","SumRTT","ThruOctetsAcked","ThruO
                  "WillSendSACK","WillUseSACK","WinScaleRcvd","WinScaleRcvd","WinScaleSent","ZeroRwin

all_measurements = ""
for(i in 1:length(query_vector)){
  if(i == 1){
    all_measurements = paste(all_measurements,query_vector[i] ,sep = "")
  } else {
  all_measurements = paste(all_measurements,",",query_vector[i] ,sep = "")
  }

}
```

```
bryan_measurements = "HCDataOctetsOut,OctetsRetrans,HCDataOctetsIn,ElapsedSecs,CurMSS,PipeSize,MaxPipeS
most_data_flow <- form_dataframe("/Users/aaronkruchten/Desktop/large data flow",bryan_measurements)
imputed_most_data_flow <- impute_frame(most_data_flow,10)
imputed_most_data_flow <- no_retransmitted_data(imputed_most_data_flow)
library(mgcv)
```

```
## Warning: package 'mgcv' was built under R version 3.5.2
```

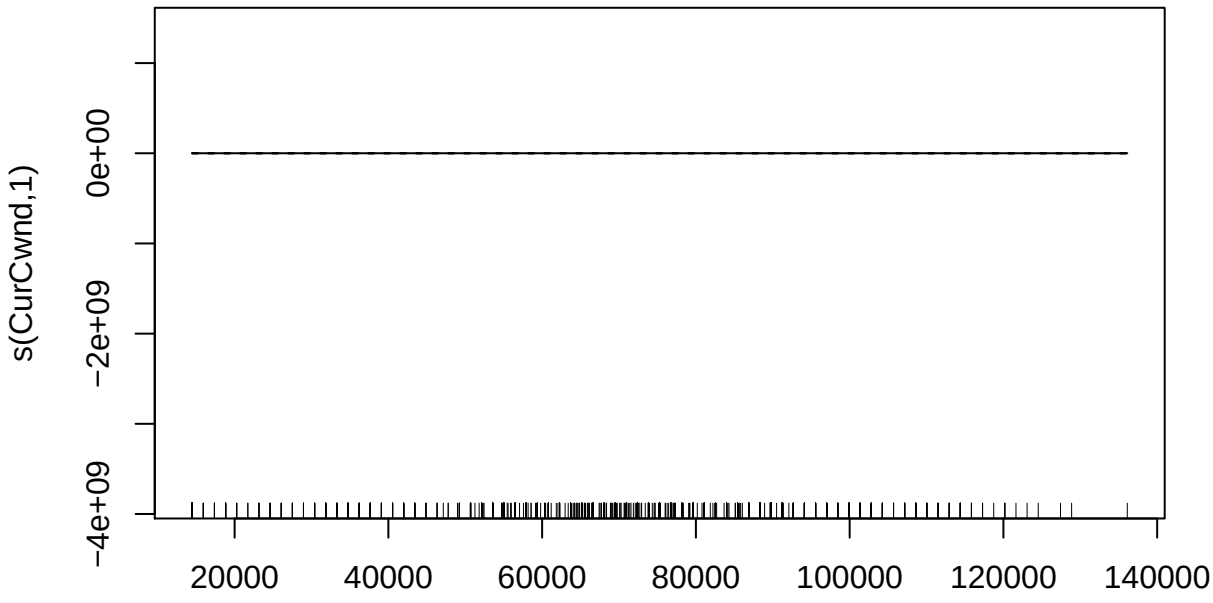
```
## Loading required package: nlme
```

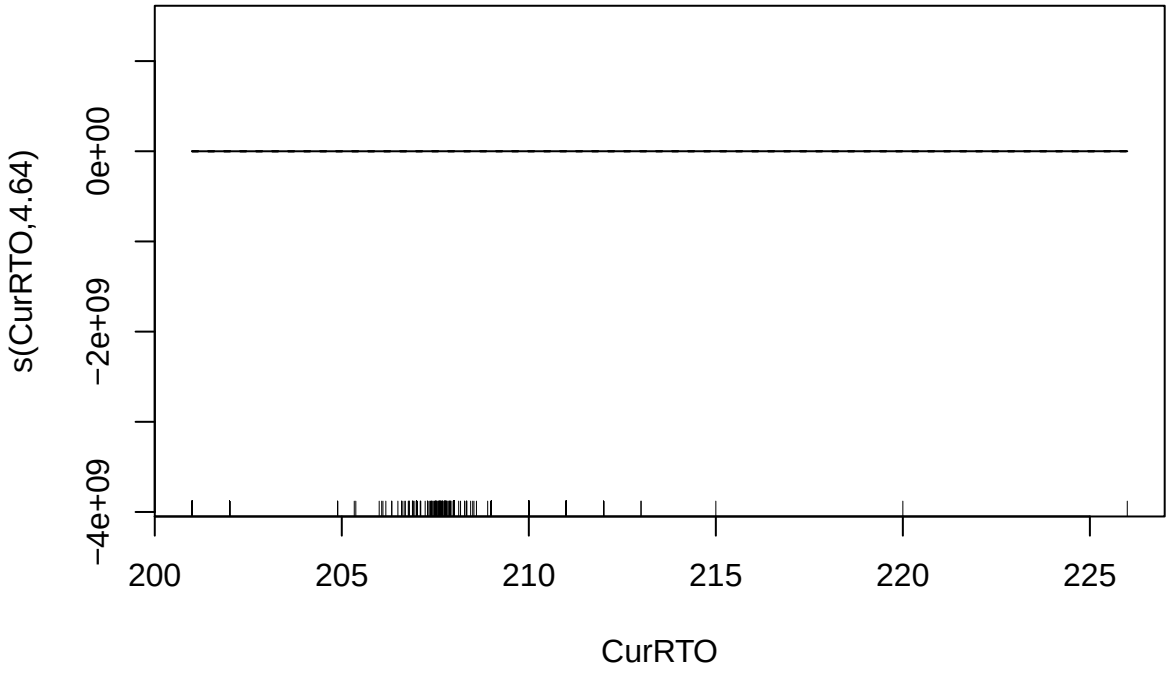
```
## Warning: package 'nlme' was built under R version 3.5.2
```

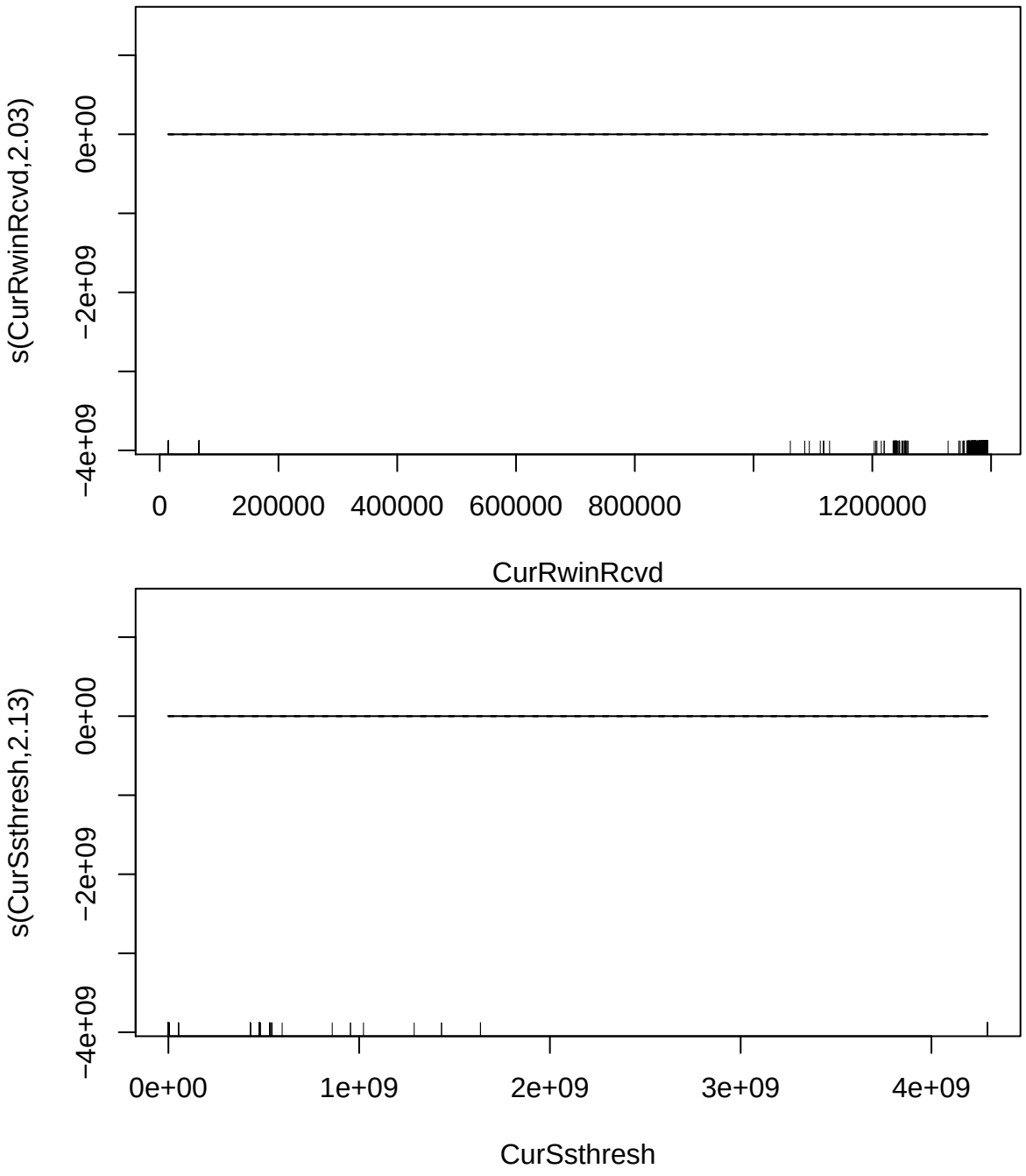
```
## This is mgcv 1.8-28. For overview type 'help("mgcv-package")'.
```

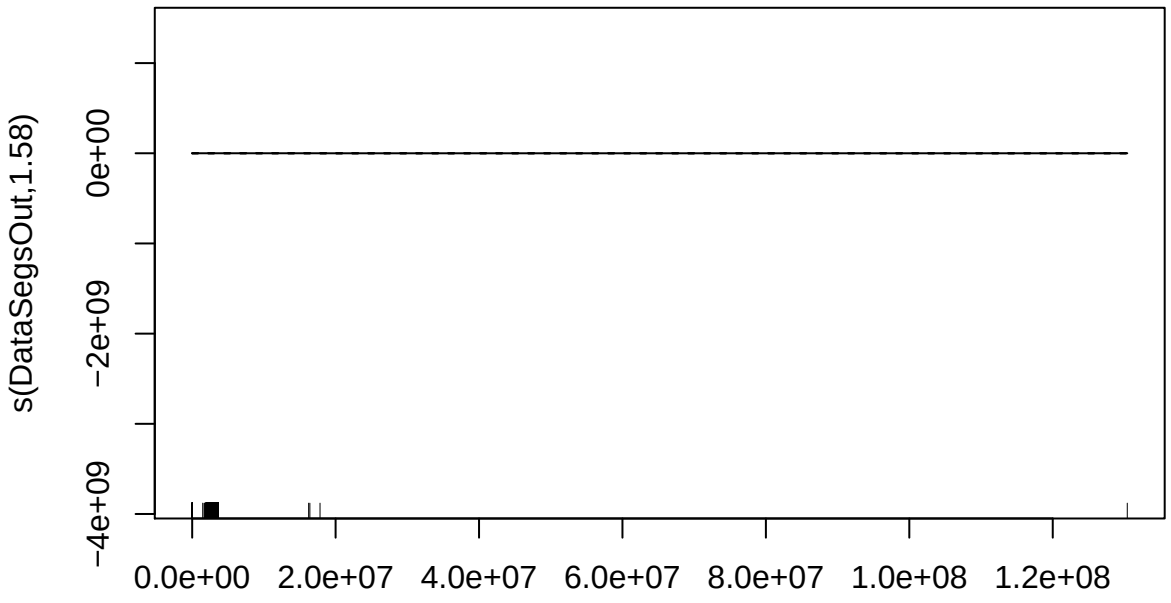
```
model <- gam(CleanOctetsOut ~ s(HCDataOctetsIn) + s(ElapsedSecs) + CurMSS + PipeSize + MaxPipeSize + Cu

model_mse = mean((predict(model) - imputed_most_data_flow$CleanOctetsOut)^2)
clean_octets_out_mean = mean(imputed_most_data_flow$CleanOctetsOut)
dumb_mse = mean((clean_octets_out_mean - imputed_most_data_flow$CleanOctetsOut)^2)
#model slightly better than predicting the average
#
#plot(model)
```

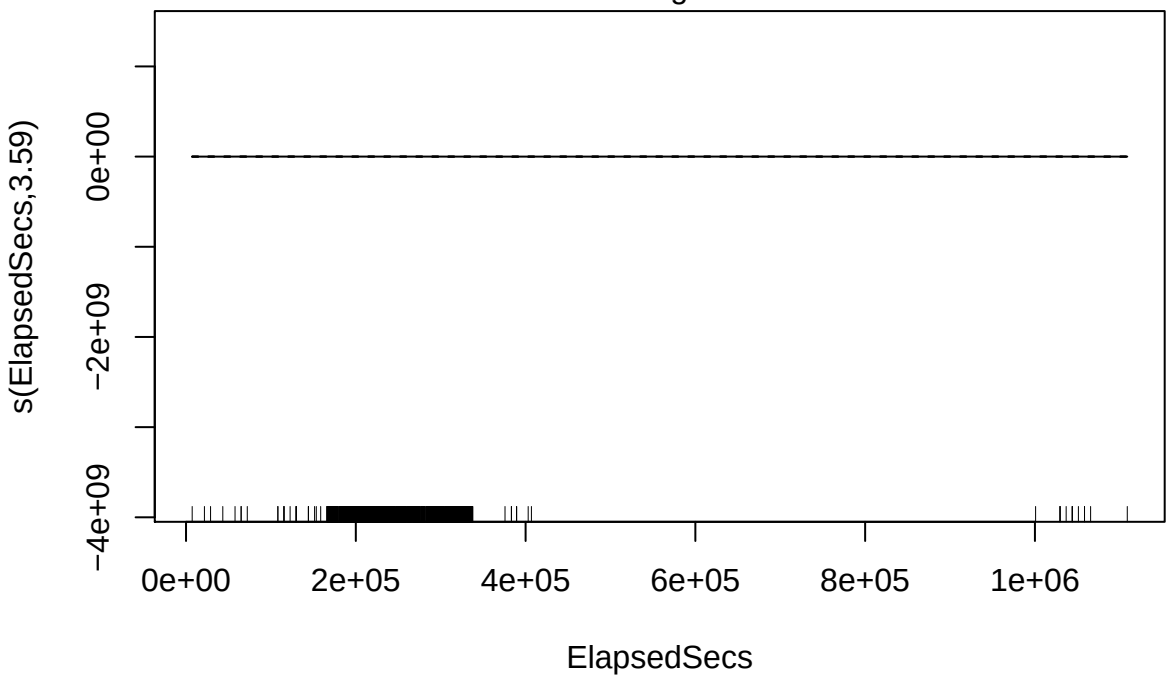
```
lots_of_measurements = "SegsOut,DataSegsOut,HCDataOctetsOut,SegsRetrans,OctetsRetrans,SegsIn,HCDataOcte

most_data_flow_more <- form_dataframe("/Users/aaronkruchten/Desktop/large data flow",lots_of_measuremen
imputed_most_data_flow_more <- impute_frame(most_data_flow_more,10)
imputed_most_data_flow_more <- no_retransmitted_data(imputed_most_data_flow_more)

library(mgcv)

more_model <- gam(CleanOctetsOut ~ CongSignals + s(CurCwnd) + CurMSS + s(CurRTO) + s(CurRwinRcvd) + Curl
plot(more_model)
```

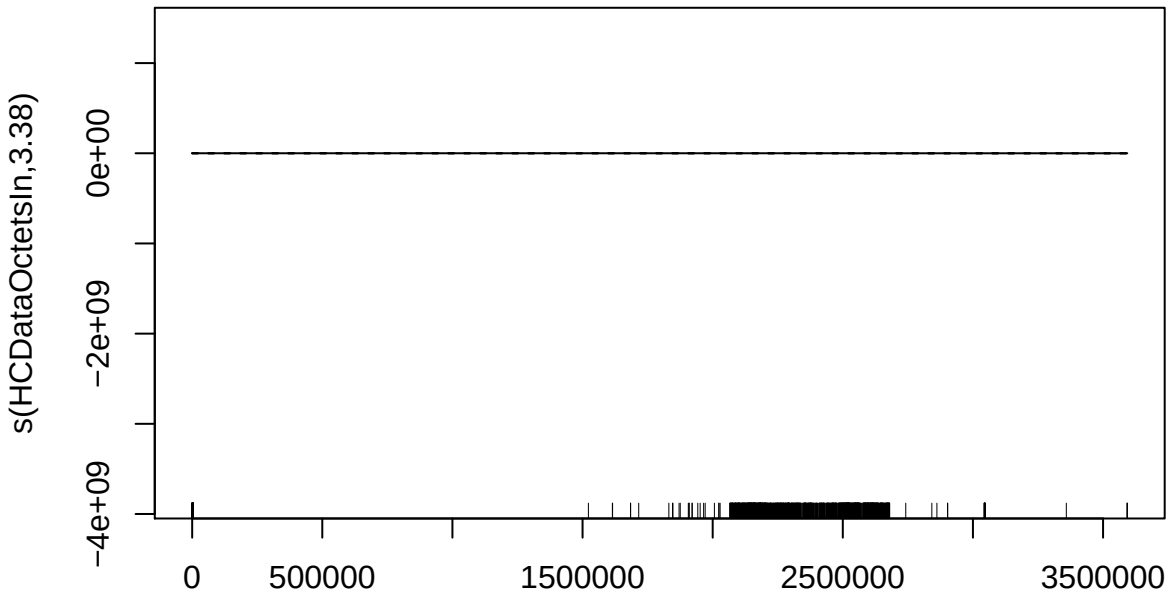

s(CurCwnd,1)
0e+00
−2e+09
−4e+09
20000
40000
60000
80000
100000
120000
140000
CurCwnd

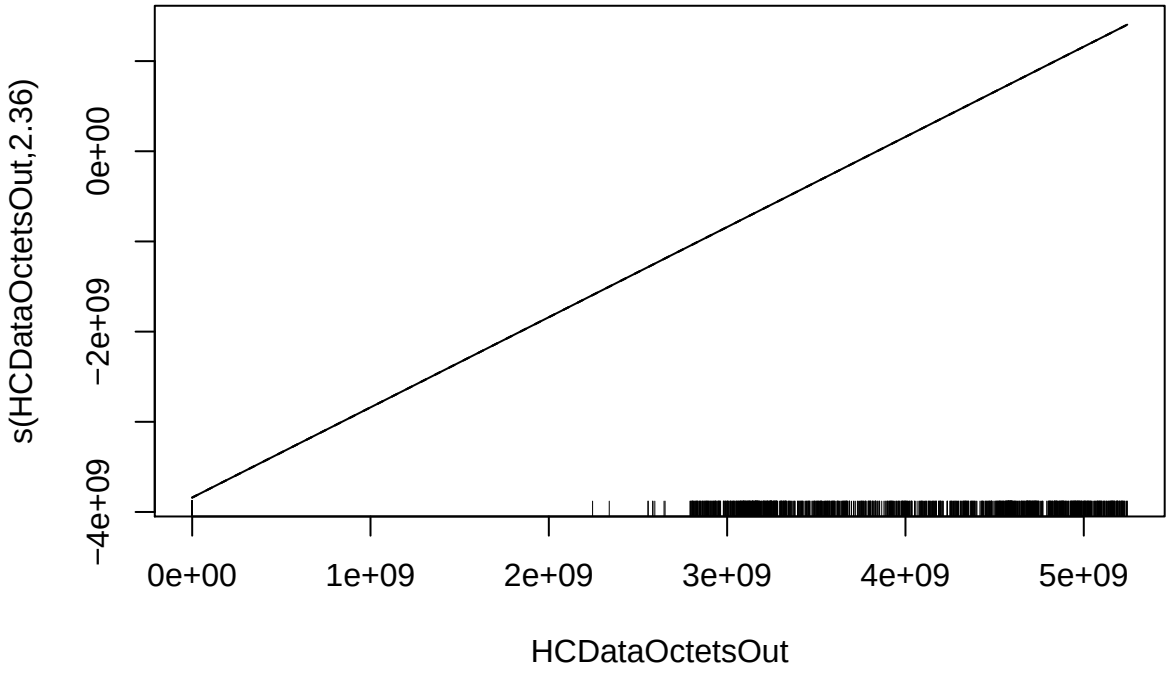

s(CurRTO,4.64)
0e+00
−2e+09
−4e+09
200
205
210
215
220
225
CurRTO

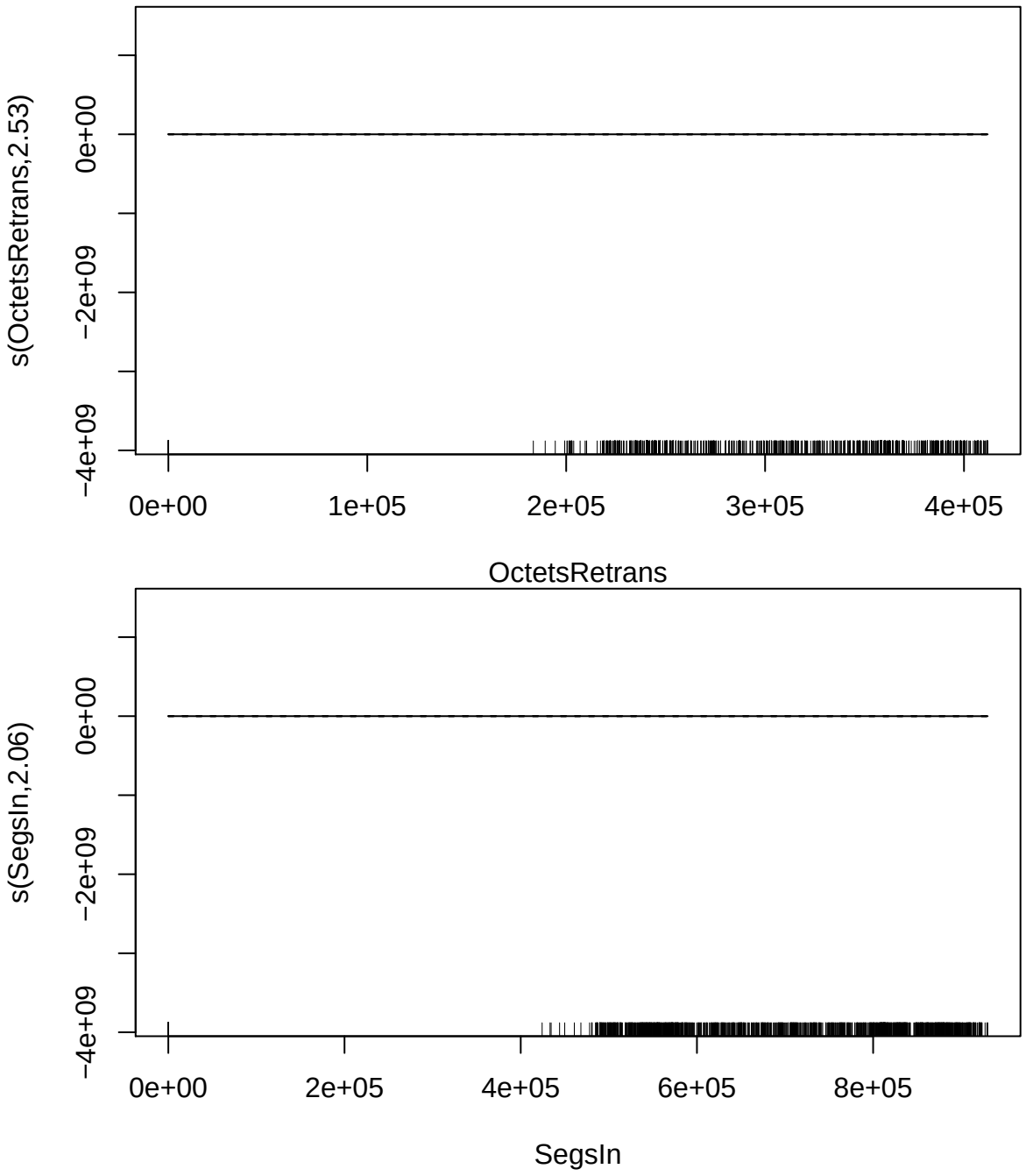

s(CurRwinRcvd,2.03)
0e+00
−2e+09
−4e+09
0
200000
400000
600000
800000
1200000
CurRwinRcvd
s(CurSsthresh,2.13)
0e+00
−2e+09
−4e+09
0e+00
1e+09
2e+09
3e+09
4e+09
CurSsthresh

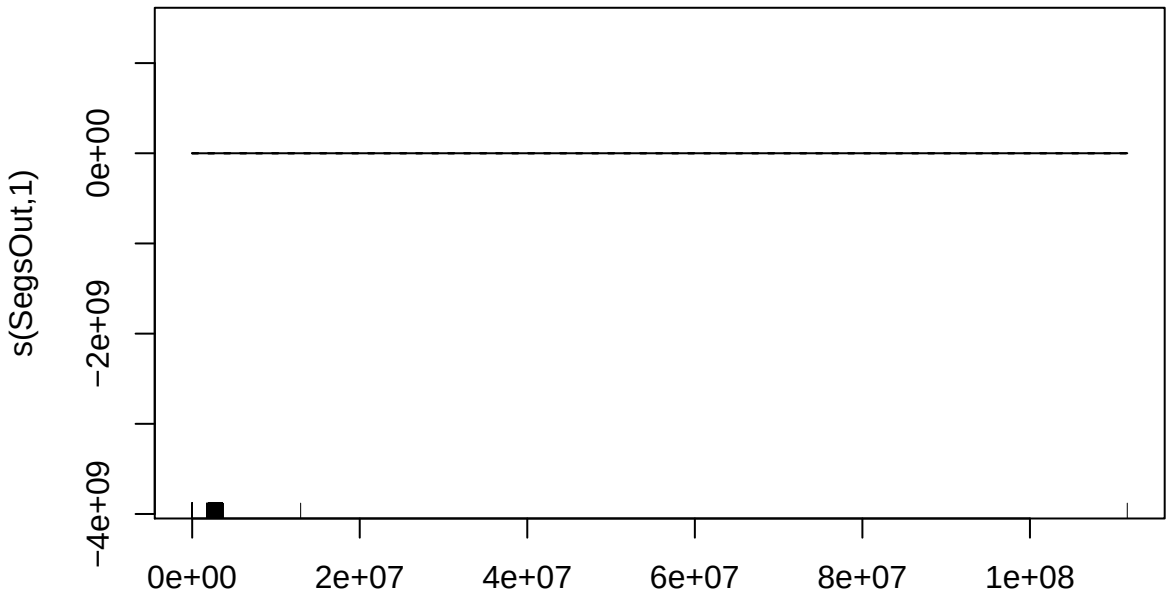

s(DataSegsOut,1.58)
0e+00
−2e+09
−4e+09
0.0e+00
2.0e+07
4.0e+07
6.0e+07
8.0e+07
1.0e+08
1.2e+08
DataSegsOut

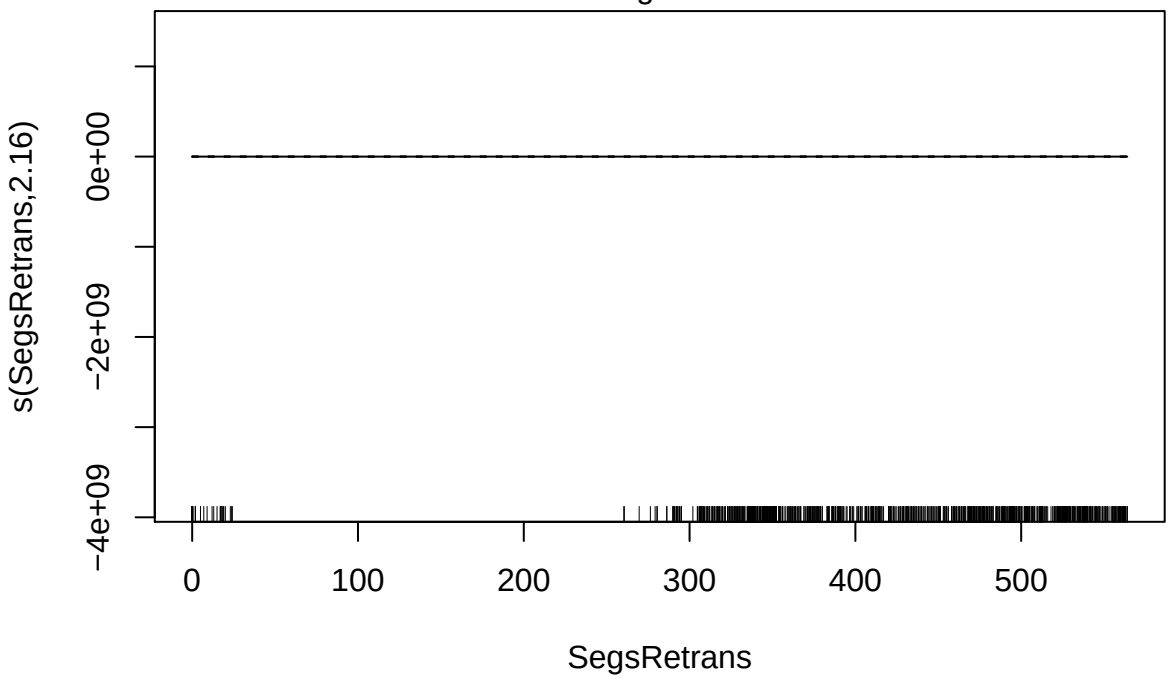

s(ElapsedSecs,3.59)
0e+00
−2e+09
−4e+09
0e+00
2e+05
4e+05
6e+05
8e+05
1e+06
ElapsedSecs

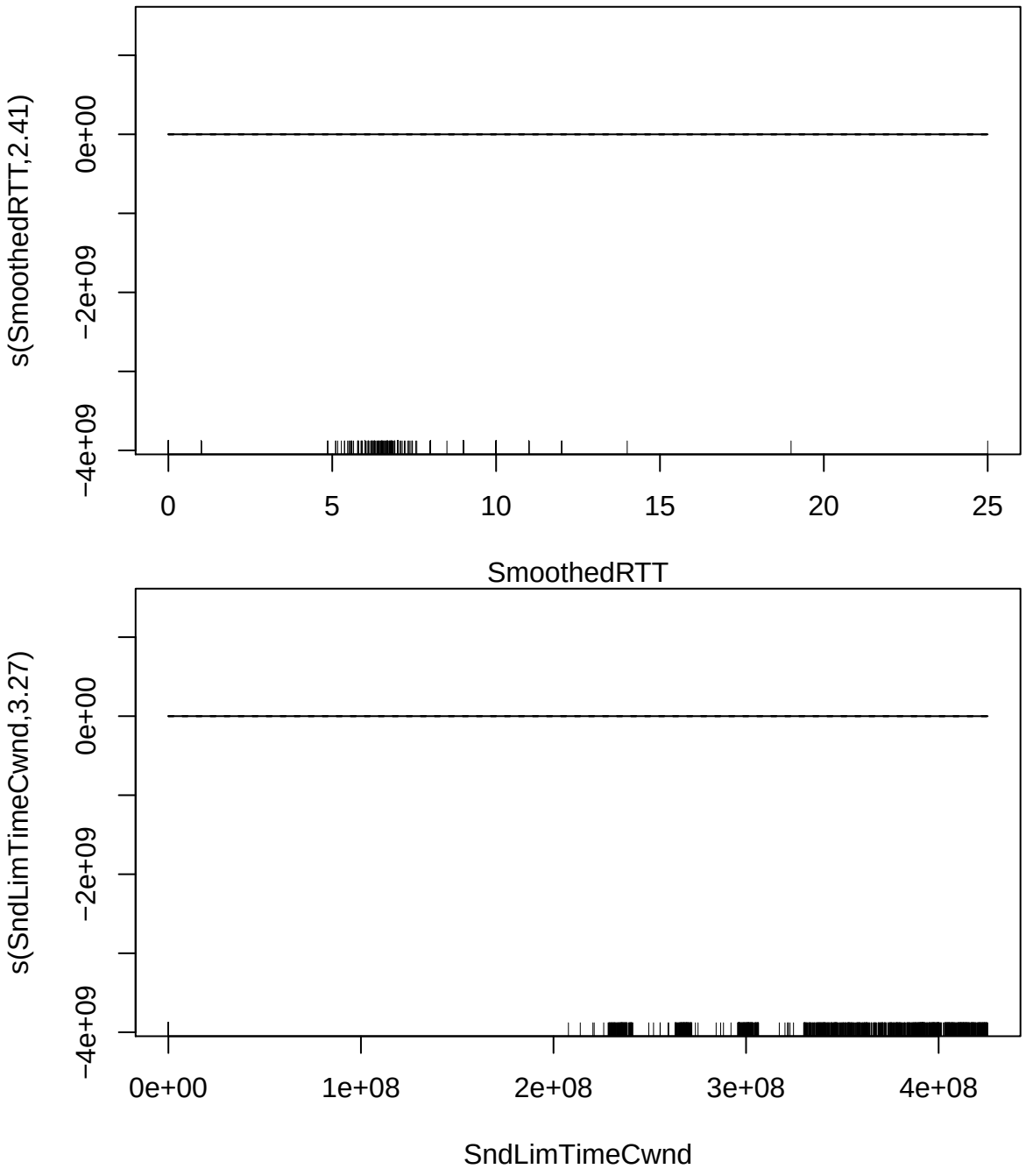

s(HCDataOctetsIn,3.38)
0e+00
−2e+09
−4e+09
0
500000
1500000
2500000
3500000
HCDataOctetsIn

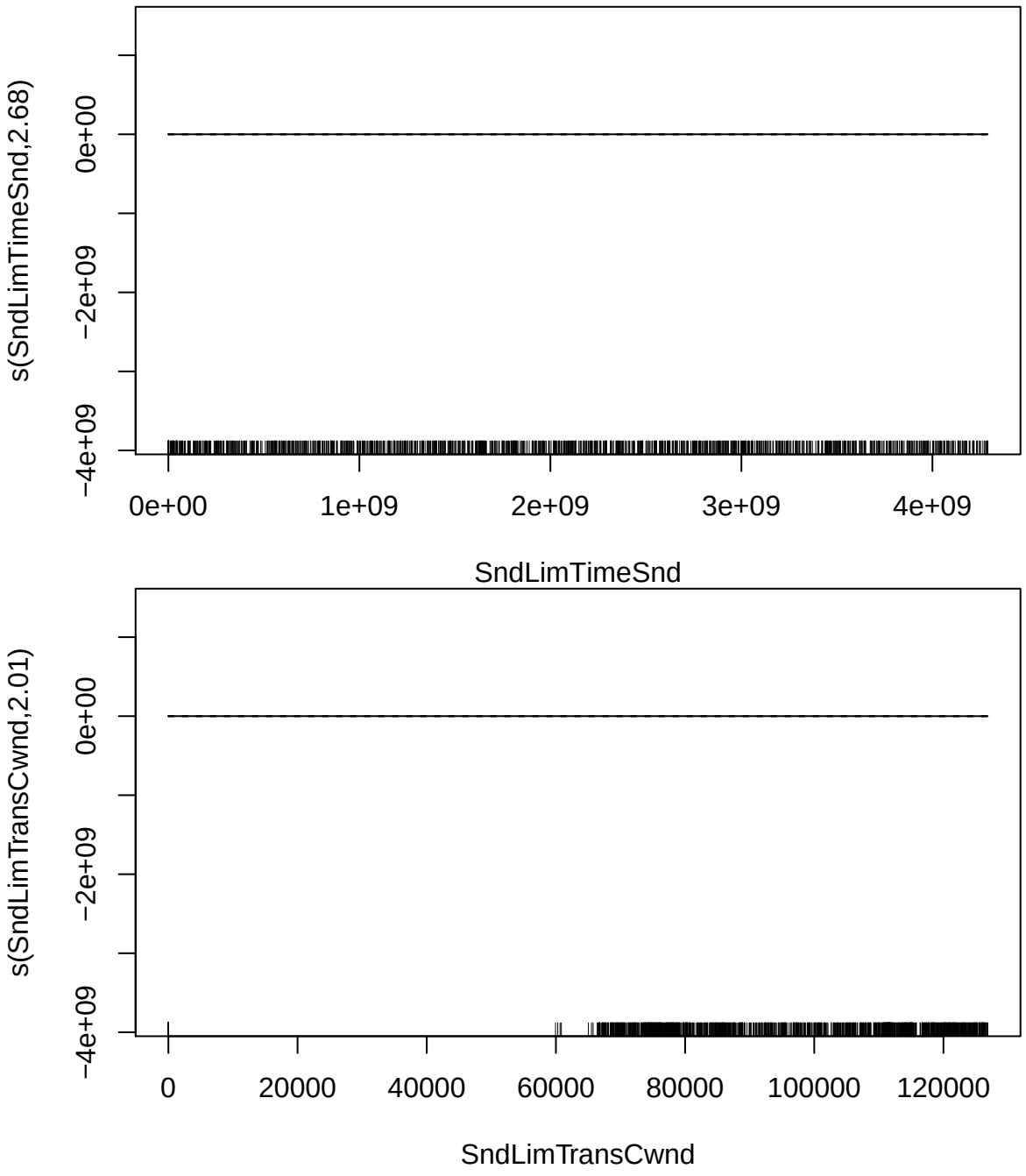

s(HCDataOctetsOut,2.36)
0e+00
−2e+09
−4e+09
0e+00
1e+09
2e+09
3e+09
4e+09
5e+09
HCDataOctetsOut

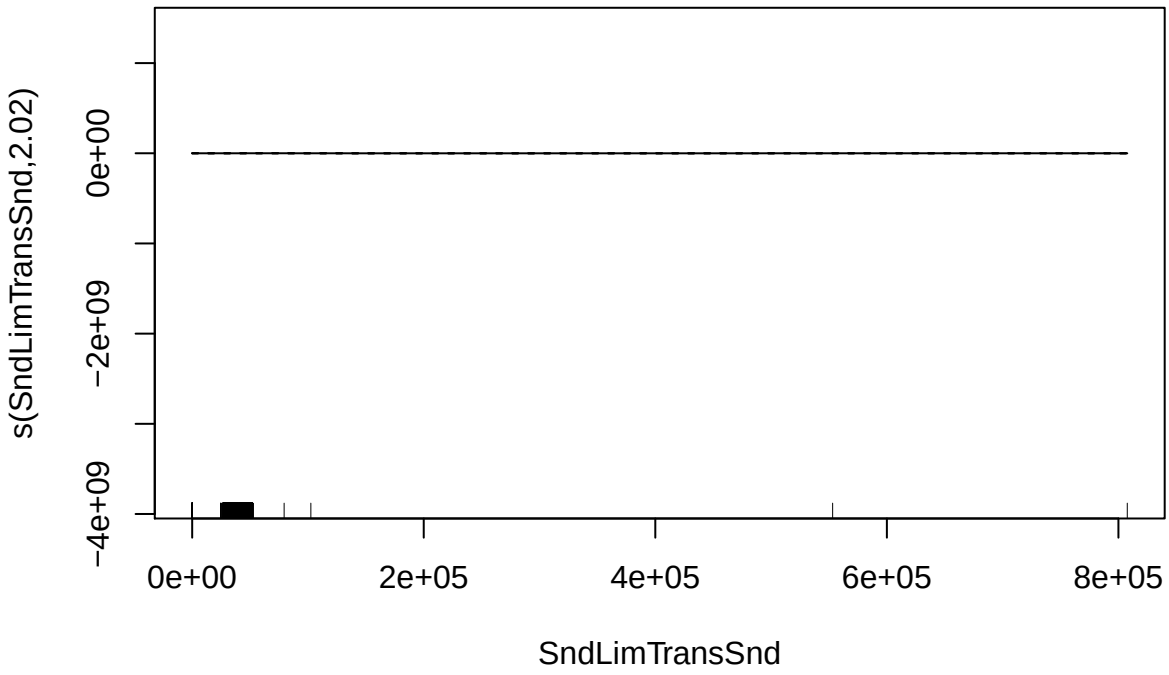

s(OctetsRetrans,2.53)
0e+00
−2e+09
−4e+09
0e+00
1e+05
2e+05
3e+05
4e+05
OctetsRetrans
s(SegsIn,2.06)
0e+00
−2e+09
−4e+09
0e+00
2e+05
4e+05
6e+05
8e+05
SegsIn


s(SegsOut,1)
0e+00
−2e+09
−4e+09
0e+00
2e+07
4e+07
6e+07
8e+07
1e+08
SegsOut


s(SegsRetrans,2.16)
0e+00
−2e+09
−4e+09
0
100
200
300
400
500
SegsRetrans


s(SmoothedRTT,2.41)
0e+00
−2e+09
−4e+09
0
5
10
15
20
25
SmoothedRTT
s(SndLimTimeCwnd,3.27)
0e+00
−2e+09
−4e+09
0e+00
1e+08
2e+08
3e+08
4e+08
SndLimTimeCwnd


s(SndLimTimeSnd,2.68)
0e+00
−2e+09
−4e+09
0e+00
1e+09
2e+09
3e+09
4e+09
SndLimTimeSnd
s(SndLimTransCwnd,2.01)
0e+00
−2e+09
−4e+09
0
20000
40000
60000
80000
100000
120000
SndLimTransCwnd


s(SndLimTransSnd,2.02)
−4e+09
−2e+09
0e+00
0e+00
2e+05
4e+05
6e+05
8e+05
SndLimTransSnd