# R Notebook

## Reading the data

```
mcmc <- setNames(data.frame(matrix(ncol = 6, nrow = 1)), c("n", "k", "D", "kD", "Stop_Time", "nu2"))
mcmc
```

```
##    n  k  D kD Stop_Time nu2
## 1 NA NA NA NA        NA  NA
```

```
read.mcmc <- function(D, K, n, df) {
    tv.fileName = paste("data/TV-V", toString(n), "K", toString(K), "D", toString(D), ".csv", sep="")
    nu2.fileName = paste("data/NU2-V", toString(n), "K", toString(K), "D", toString(D), ".csv", sep="")
    KD <- paste(toString(K), toString(D), sep="/")
    stop <- nrow(read.csv(tv.fileName, skip = 2))
    nu2 <- tail(read.csv(nu2.fileName, skip=2), n=1)$NU2.est
    row <- c(n, K, D, KD, stop, nu2)
    df <- rbind(df, row)
    return(df)
}

# degree 3
D = 3
for (K in 5:5){
  for (n in seq(4, 10, 2)){
    mcmc <- read.mcmc(D, K, n, mcmc)
  }
}

for (K in 6:7){
  for (n in seq(4, 8, 2)){
    mcmc <- read.mcmc(D, K, n, mcmc)
  }
}

# degree 4
D = 4
for (K in 6:7){
  for (n in seq(5, 9, 1)){
    mcmc <- read.mcmc(D, K, n, mcmc)
  }
}

# degree 4
D = 4
for (K in 6:6){
  for (n in seq(9, 11, 1)){
    mcmc <- read.mcmc(D, K, n, mcmc)
  }
}

# degree 4
D = 4
for (K in 7:7){
  for (n in seq(9, 9, 1)){ # change to 10
    mcmc <- read.mcmc(D, K, n, mcmc)
  }
}
```

## Read V12 K5 D3

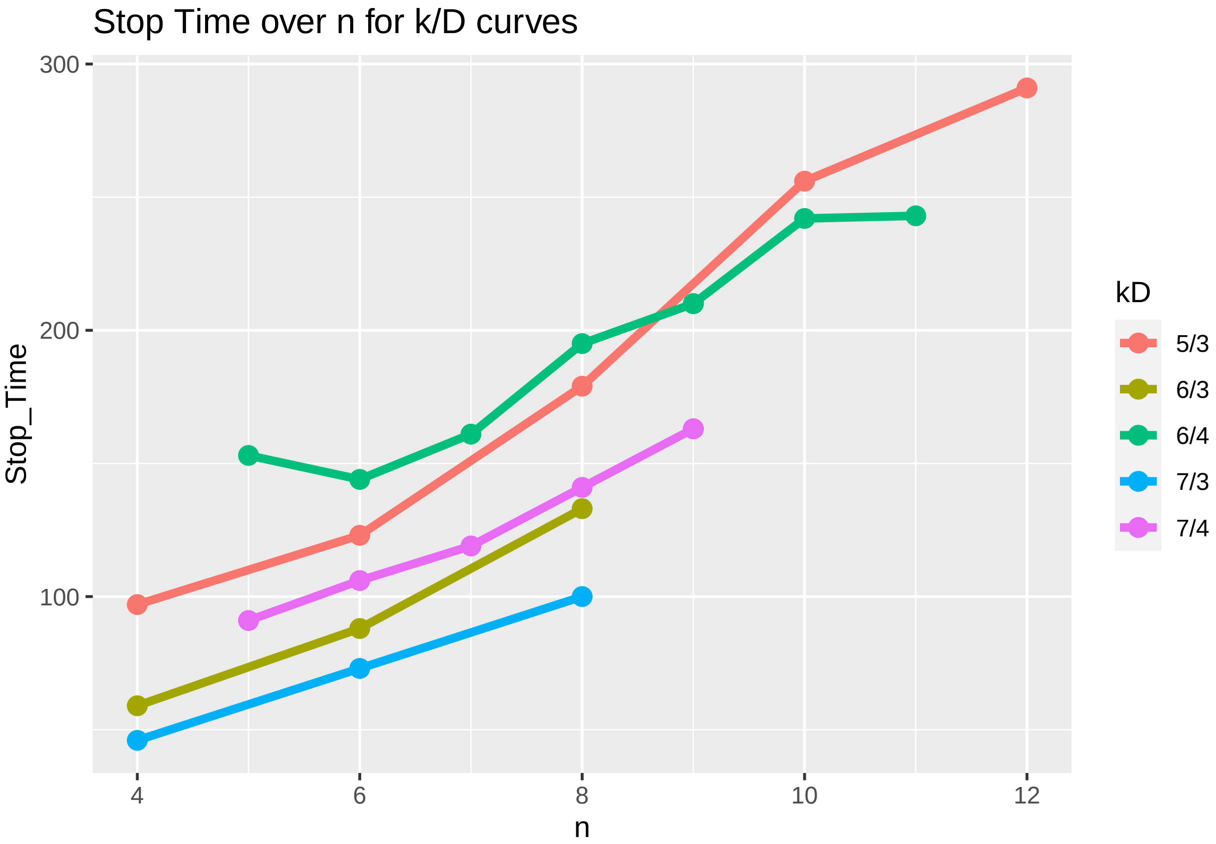
```
D = 3
n = 12
K = 5

    tv.fileName = paste("data/TV-V", toString(n), "K", toString(K), "D", toString(D), ".csv", sep="")
    nu2.fileName = paste("data/NU2-V", toString(n), "K", toString(K), "D", toString(D), ".csv", sep="")
    KD <- paste(toString(K), toString(D), sep="/")
    stop <- nrow(read.csv(tv.fileName, skip = 2))
    row <- c(n, K, D, KD, stop, NA)
    mcmc <- rbind(mcmc, row)
```

## Ensure proper types

```
mcmc <- mcmc[-1,]
mcmc$k <- as.integer(mcmc$k)
mcmc$D <- as.integer(mcmc$D)
mcmc$n <- as.integer(mcmc$n)
mcmc$Stop_Time <- as.integer(mcmc$Stop_Time)
mcmc$nu2 <- as.numeric(mcmc$nu2)
mcmc
```

```
##      n k D  kD Stop_Time      nu2
## 2    4 5 3 5/3        97 0.008345
## 3    6 5 3 5/3       123 0.000534
## 4    8 5 3 5/3       179 0.000059
## 5   10 5 3 5/3       256 0.000030
## 6    4 6 3 6/3        59 0.002784
## 7    6 6 3 6/3        88 0.000127
## 8    8 6 3 6/3       133 0.000017
## 9    4 7 3 7/3        46 0.001195
## 10   6 7 3 7/3        73 0.000043
## 11   8 7 3 7/3       100 0.000011
## 12   5 6 4 6/4       153 0.001404
## 13   6 6 4 6/4       144 0.000260
## 14   7 6 4 6/4       161 0.000074
## 15   8 6 4 6/4       195 0.000034
## 16   9 6 4 6/4       210 0.000024
## 17   5 7 4 7/4        91 0.000406
## 18   6 7 4 7/4       106 0.000076
## 19  7 7 4 7/4       119 0.000024
## 20  8 7 4 7/4       141 0.000016
## 21  9 7 4 7/4       163 0.000017
## 22  9 6 4 6/4       210 0.000024
## 23 10 6 4 6/4       242 0.000026
## 24 11 6 4 6/4       243 0.000028
## 25  9 7 4 7/4       163 0.000017
## 26 12 5 3 5/3       291       NA
```

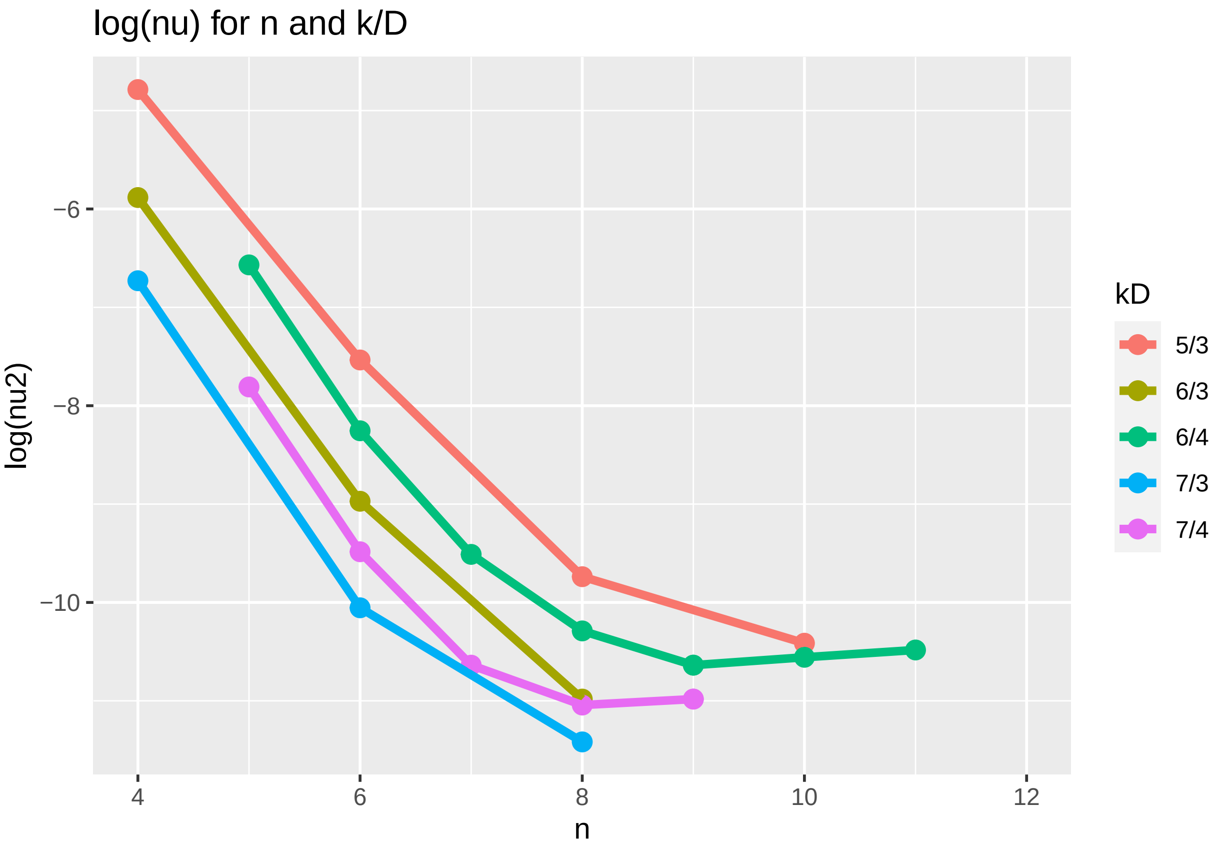
```
library(ggplot2)
ggplot(mcmc, aes(n, Stop_Time)) + geom_point(aes(colour = kD), size = 3) + geom_line(aes(colour = kD), s
```



```
ggplot(mcmc, aes(n, log(nu2))) + geom_point(aes(colour = kD), size = 3) + geom_line(aes(colour = kD), si
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

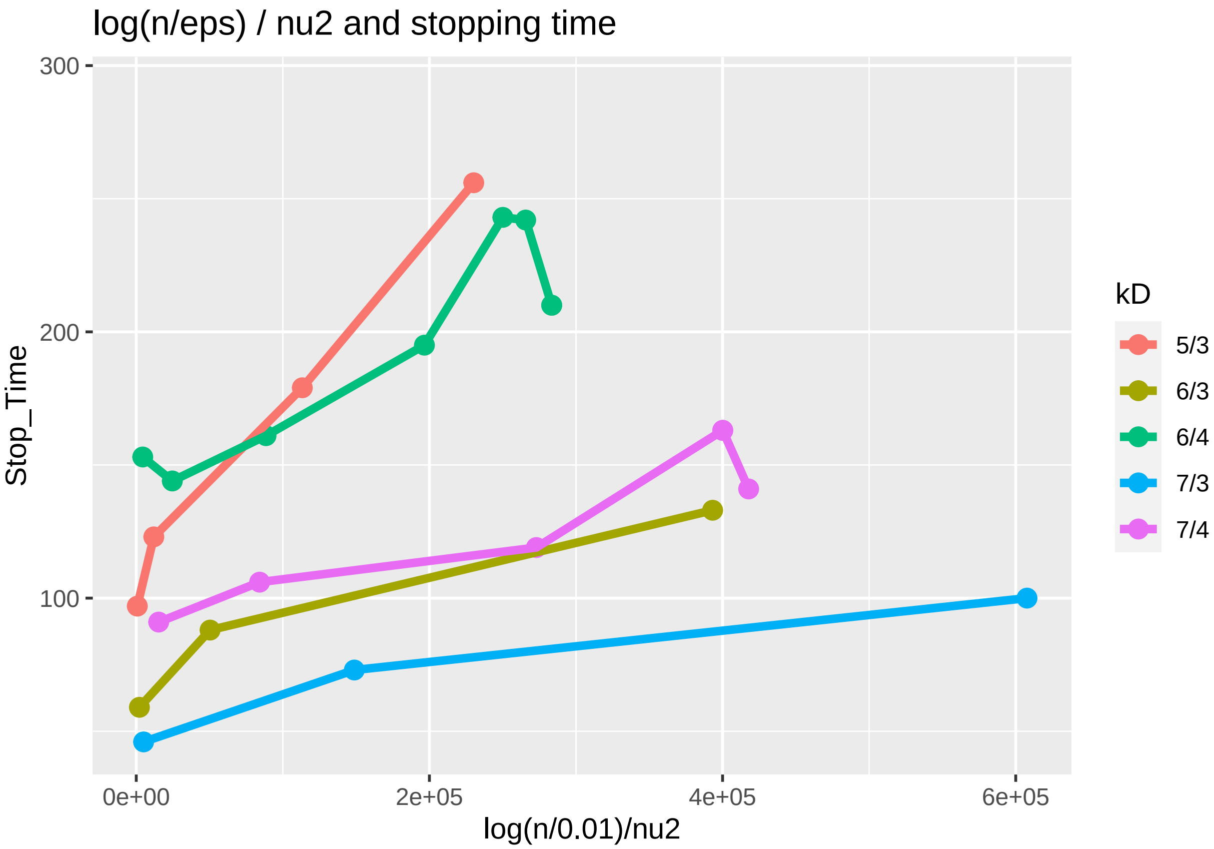


```
# small nu => mixes more slowly
# as size of graph gets bigger, then spectral gap gets smaller
```

```
# stopping time <= O(log(n / eps)/nu2)
# this bound is very loose - much bigger than stopping time
# using bounds w nu2 not great for proving conjecture
# ordering is similar to stopping vs n graph
# nu2 picks one random starting vector, will benefit from more trials
ggplot(mcmc, aes(log(n/0.01)/nu2, Stop_Time)) + geom_point(aes(colour = kD), size = 3) + geom_line(aes(
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

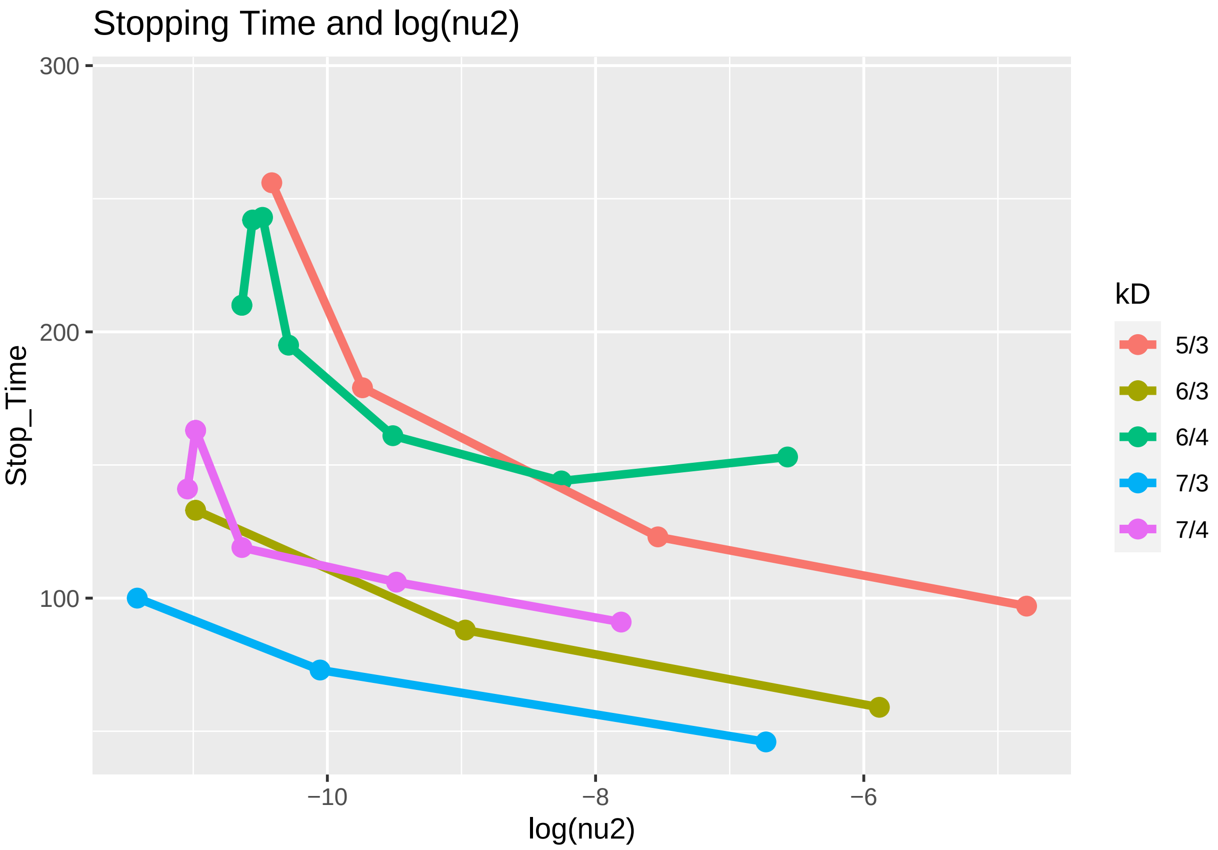
```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



```
ggplot(mcmc, aes(log(nu2), Stop_Time)) + geom_point(aes(colour = kD), size = 3) + geom_line(aes(colour =
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

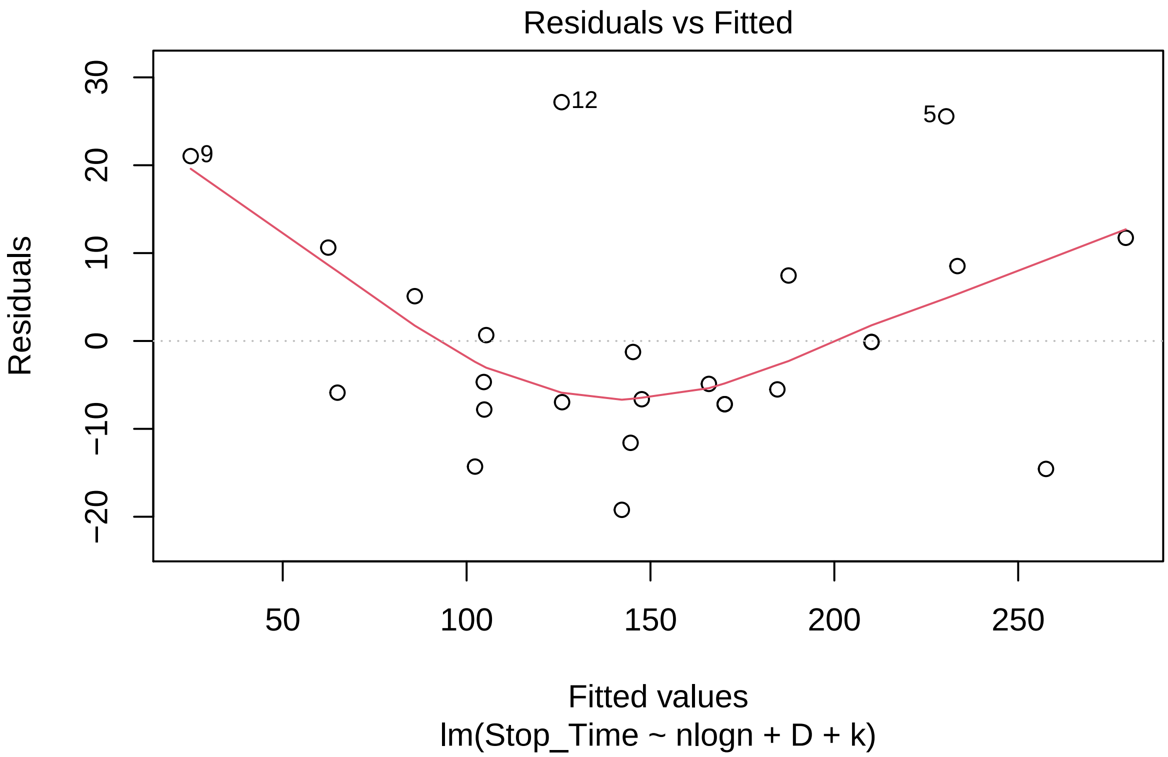
```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

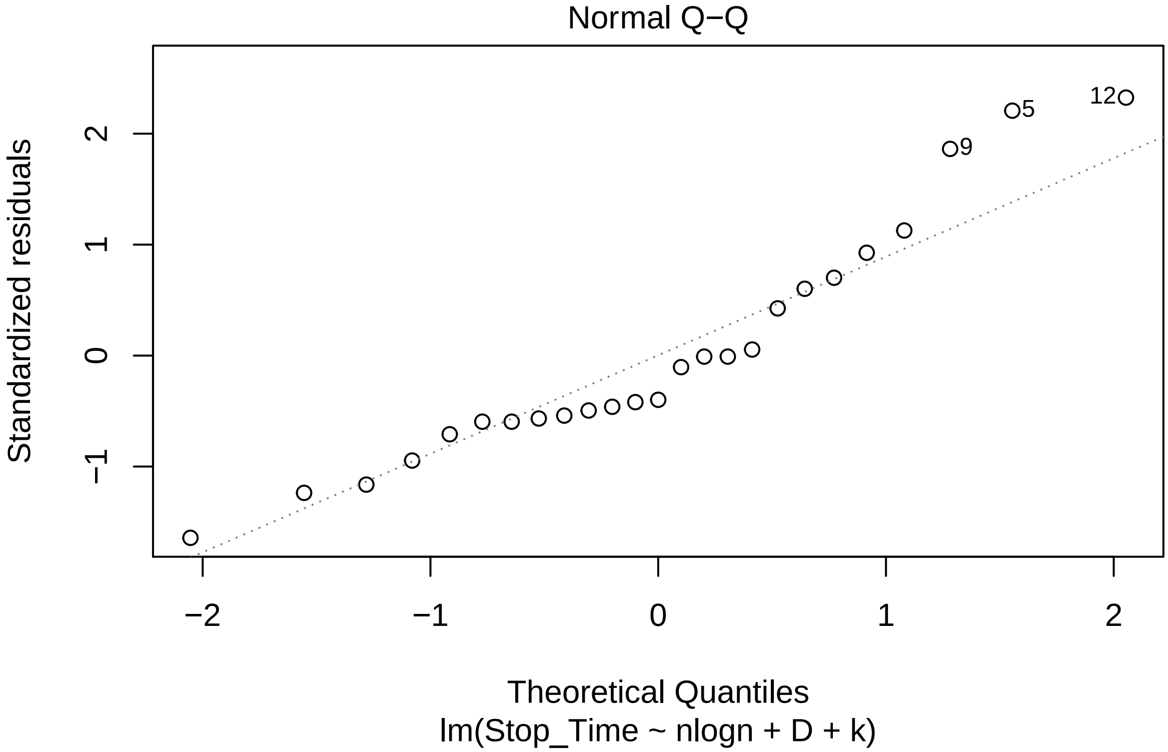


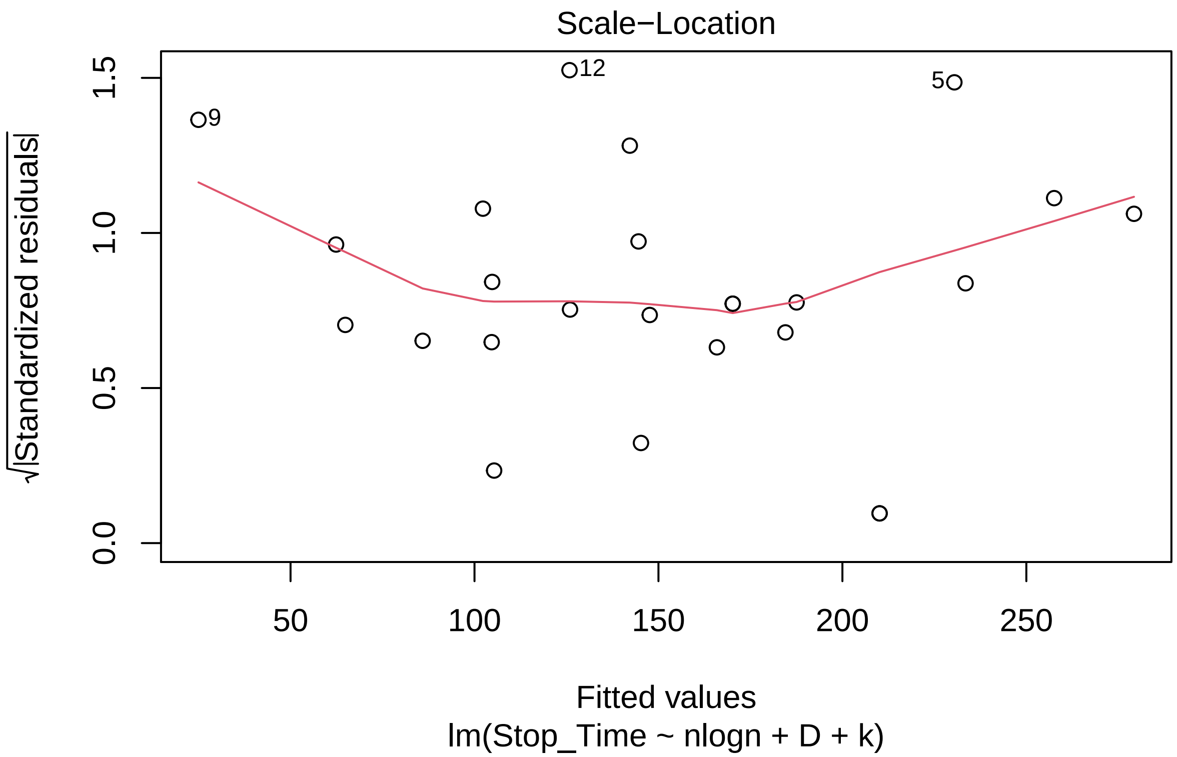
```
# nlogn coefficient

mcmc$nlogn <- mcmc$n * log(mcmc$n)

lm2 <- lm(Stop_Time ~ nlogn + D + k, data=mcmc)
summary(lm2)
```

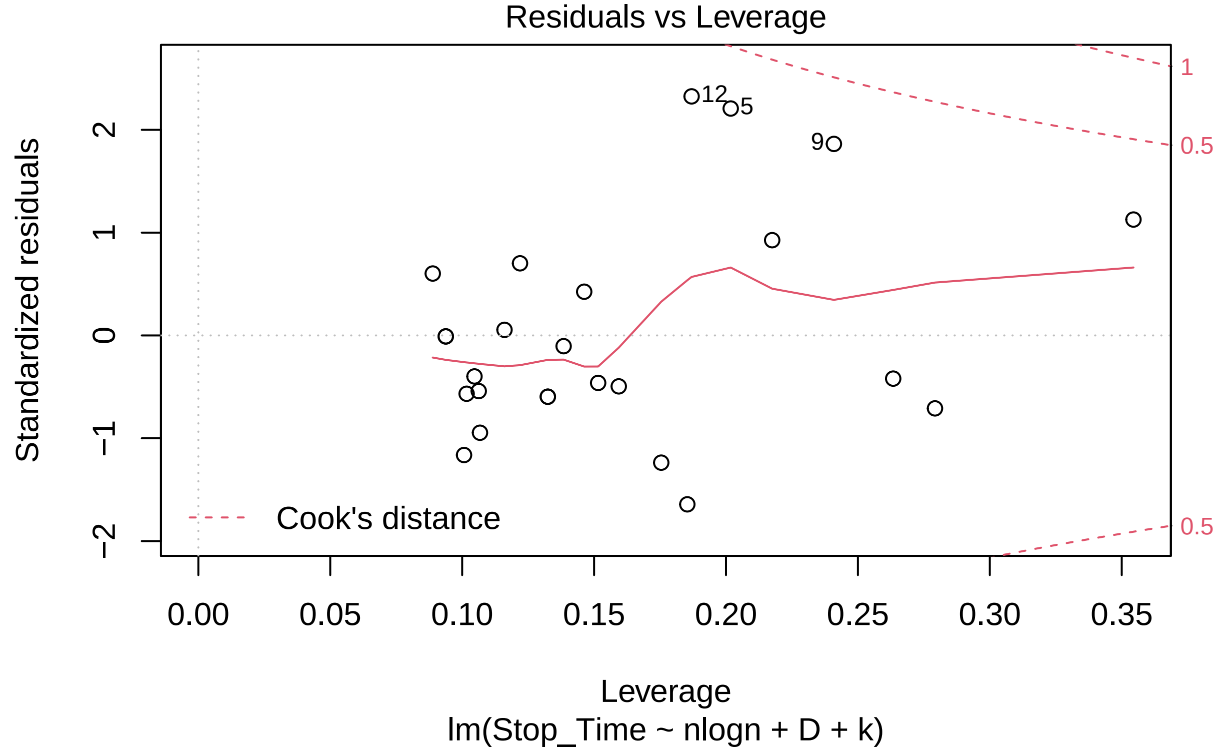
```
##
## Call:
## lm(formula = Stop_Time ~ nlogn + D + k, data = mcmc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.215  -7.193  -4.670   7.450  27.176
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 135.6718    25.6782   5.284 3.08e-05 ***
## nlogn         7.1871     0.4316  16.654 1.41e-13 ***
## D            42.9594     6.0270   7.128 4.98e-07 ***
## k           -39.9202     4.1138  -9.704 3.27e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.96 on 21 degrees of freedom
## Multiple R-squared:  0.9649, Adjusted R-squared:  0.9598
## F-statistic: 192.2 on 3 and 21 DF,  p-value: 2.008e-15
```

```
plot(lm2)
```


Residuals vs Fitted
Residuals
30
20
10
0
−10
−20
50
100
150
200
250
Fitted values
lm(Stop_Time ~ nlogn + D + k)
9
12
5


Normal Q−Q
Standardized residuals
2
1
0
−1
−2
−1
0
1
2
Theoretical Quantiles
lm(Stop_Time ~ nlogn + D + k)
9
5
12



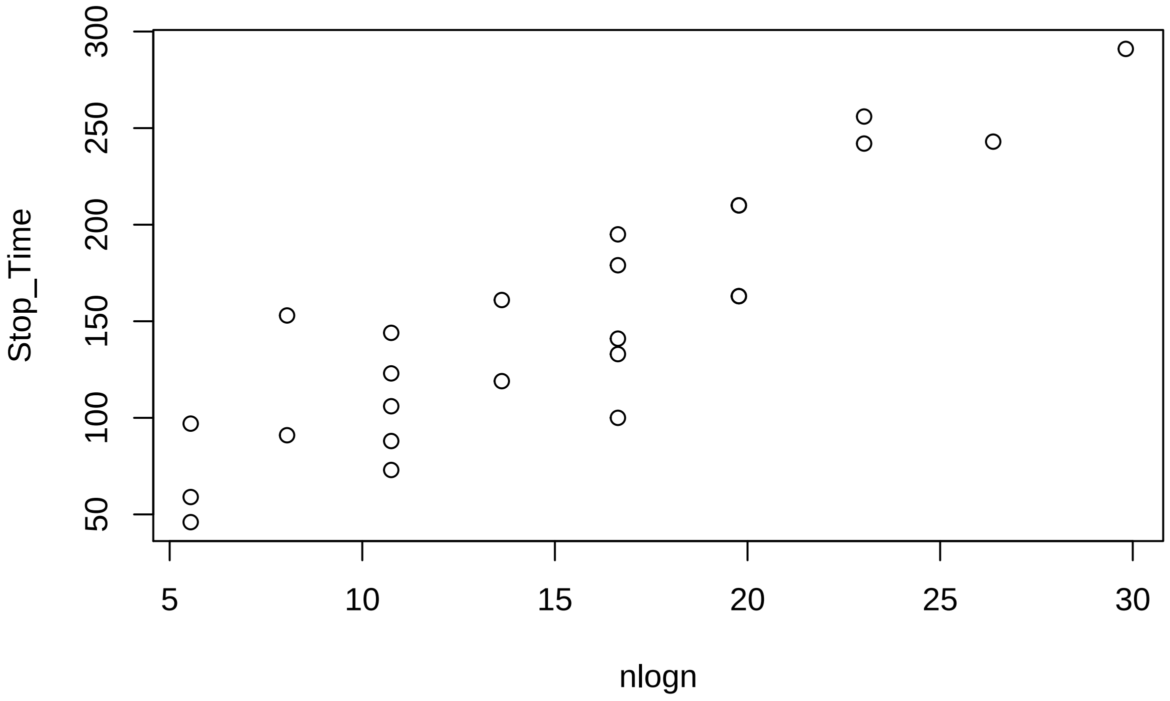


```
lm1 <- lm(Stop_Time ~ nlogn, data=mcmc)
summary(lm1)
```

```
##
## Call:
## lm(formula = Stop_Time ~ nlogn, data = mcmc)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -64.609 -23.609   1.514  21.495  63.514
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  19.0967    15.3159   1.247    0.225
## nlogn         8.7471     0.9311   9.394 2.45e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.04 on 23 degrees of freedom
## Multiple R-squared:  0.7933, Adjusted R-squared:  0.7843
## F-statistic: 88.25 on 1 and 23 DF,  p-value: 2.454e-09
```

```r
lm.n <- lm(Stop_Time ~ n, data=mcmc)
lm.nlogn <- lm(Stop_Time ~ nlogn, data=mcmc)

plot(Stop_Time ~ nlogn, data=mcmc)
```



```r
anova(lm.n, lm.nlogn)
```

```
## Analysis of Variance Table
##
## Model 1: Stop_Time ~ n
## Model 2: Stop_Time ~ nlogn
##   Res.Df   RSS Df Sum of Sq F Pr(>F)
## 1     23 21853
## 2     23 20758  0    1094.6
```

```r
lm4 <- lm(Stop_Time ~ (nlogn + D + k)^2, data=mcmc)
summary(lm4)
```

```
##
## Call:
## lm(formula = Stop_Time ~ (nlogn + D + k)^2, data = mcmc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.796  -5.673  -1.252   7.041  18.944
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -237.9957   155.7889  -1.528   0.1440
## nlogn         17.5774     2.7296   6.439 4.64e-06 ***
## D            120.5573    49.6028   2.430   0.0258 *
## k             13.8573    22.8793   0.606   0.5523
## nlogn:D       -0.6227     0.8547  -0.729   0.4757
## nlogn:k       -1.4290     0.5574  -2.563   0.0195 *
## D:k          -10.2076     7.0439  -1.449   0.1645
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.02 on 18 degrees of freedom
## Multiple R-squared:  0.982,  Adjusted R-squared:  0.976
## F-statistic: 163.8 on 6 and 18 DF,  p-value: 1.047e-14
```