Leverage and big data

Aimee Barciauskas, Roger Cusco, Hari

November 21, 2015

1. SimDat1

Simulate a dataset from a linear regression model as follows. Generate 100K replications of 30 inputs using a 30-dim Gaussian with mean 0 and covariance the identity matrix. Generate the output as a linear regression on the inputs (without intercept) with coefficients randomly drawn from a multivariate Gaussian with mean 0 and variance 3. This will be SimDat1.

```r
# install.packages('mvtnorm')
# install.packages('scales')
library(mvtnorm)
library(scales)

# Generate 100k replications of 30 inputs
m <- 30
# n <- 100000
n <- 10000
sigma <- diag(m)
simdat1.xs <- rmvnorm(n, mean = rep(0, m), sigma = sigma)
# How can this be a multivariate gaussian?
simdat1.betas <- rnorm(m, mean = 0, sd = 1.5)

# SimDat1
simdat1.ys <- simdat1.xs %*% simdat1.betas
```

2. Revise and briefly describe what is a mixture of Gaussian

distributions (see e.g. Bishop) and describe a procedure for simulating from one.

A mixture of Gaussian distributions is a linear combination of multiple Gaussian Normal distributions, each having a mixing coefficient where the sum of mixture coefficients is equal to 1. Intuitively, the mixture coefficient k is the probability of an observation having been drawn from distribution k.

To simulate from a mixture of k Gaussians, one could randomly sample from a multinomial distribution where the kth Gaussian has the kth mixing coefficient (probability). This observation would be drawn from distribution k.

3. SimDat2

Simulate another linear regression dataset as follows. Generate 100K replications of 30 inputs from a mixture of 2 30-dim Gaussians; the first with mean 0 and covariance the identity matrix; the second with mean 0 and covariance 10 times the identity. The first component has weight 0.95 and the second 0.05. Generate the output as a linear regression on the inputs (without intercept) with coefficients randomly drawn from a multivariate Gaussian with mean 0 and variance 3. This will be SimDat2.

Question: Why do we generate the output?

```r
# Assign observations to mixtures
mixture_assignment <- rmultinom(n = n, size = 1, prob = c(0.95,0.05))

# Check we didn't do anything silly
# sum(mixture_assignment[1,])/n
# sum(mixture_assignment[2,])/n
assignments <- apply(mixture_assignment, 2, function(col) { match(1,col)})

sigma.gaussian1 <- diag(m)
sigma.gaussian2 <- diag(m)*10

rdraw.gaussian1 <- function() {
  rmvnorm(1, mean = rep(0, m), sigma = sigma.gaussian1)
}

rdraw.gaussian2 <- function() {
  rmvnorm(1, mean = rep(0, m), sigma = sigma.gaussian2)
}

simdat2.xs <- matrix(0, nrow = n, ncol = m)
# draw from 2 different gaussians
# waiting ...
for (idx in 1:length(assignments)) {
  assignment <- assignments[idx]
  if (assignment == 1) {
    simdat2.xs[idx,] <- rdraw.gaussian1()[1,]
  } else {
    simdat2.xs[idx,] <- rdraw.gaussian2()[1,]
  }
}

# SimDat2
simdat2.betas <- rnorm(m, mean = 0, sd = 1.5)
simdat2.ys <- simdat2.xs %*% simdat2.betas
```

4. Describe briefly what is leverage for linear regression, how it is computed and propose a linear transformation of the leverage that makes it numerically comparable across datasets with different number of inputs and different number of replications.

Leverage of the ith observation is the value $H_{i,i}$ of the hat matrix H:

$H = \phi(\phi^T\phi)^{-1}\phi^T$

So all the values of leverage would be the diagonal entries of the hatmatrix.

The value of leverage of an individual datapoint:

$h_{i,i} = H[i,i]$

can be interpreted as the "number of paramters" being used to fit the corresponding observation. In the case of extreme overfitting, where $m = n$, $h_{i,i}$ for each datapoint will be equal to 1. As n grows large, the influence each observation has on the parameters decreases. However, observations with high leverage will have a disproportionately large contribution to the values of the parameters.

The average leverage will be $(m+1)/n$, where m is the dimension of the inputs and n is the number of observations.

Leverages follow a chi-squared distribution when the covariance equation:

$X^T X$

is scaled by the number of observations

$\frac{X^T X}{N}$

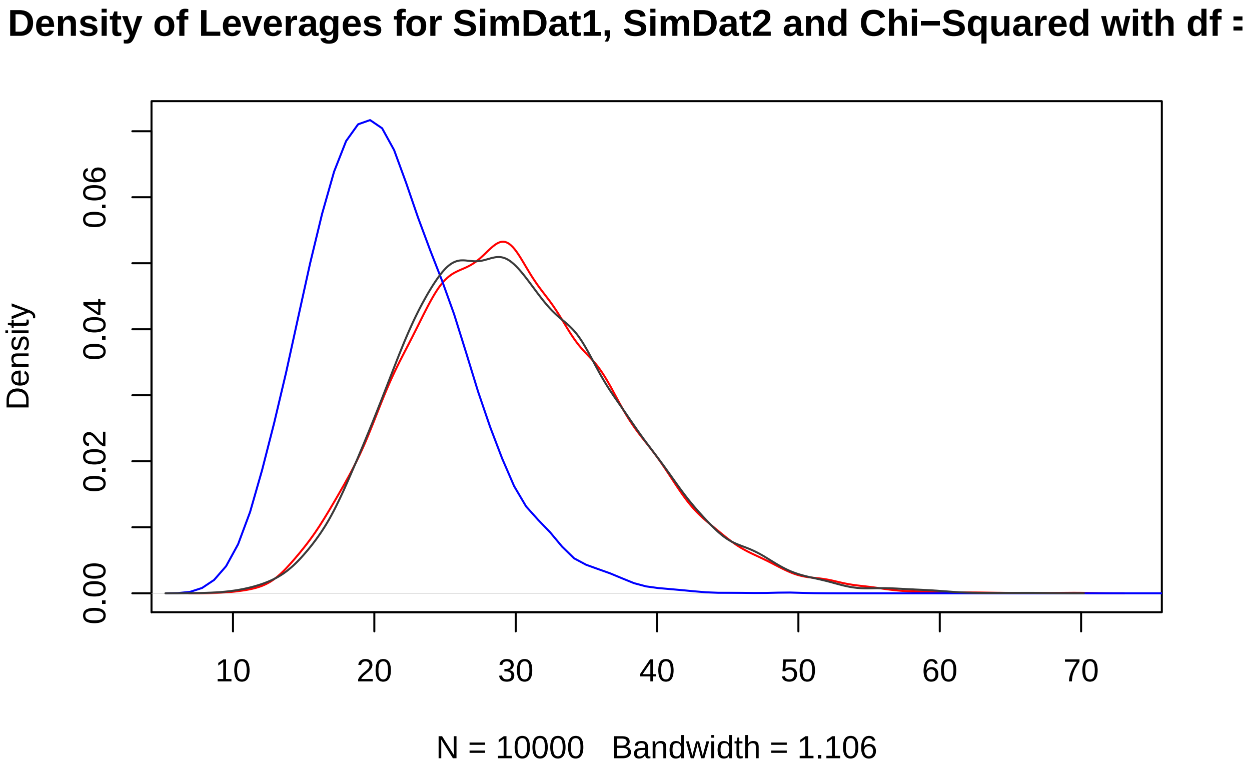
So the calculation for the "scaled hat matrix" is:

$X^T(\frac{X^T X}{N})^{-1}X)$

These scaled datasets follow a chi-squared distribution, with degrees of freedom equal to m.

To compare leverages across different data sets, we can evaluate the scaled leverages against the standard deviation of the corresponding chi-squared distribution.

The following is an example of this comparison for the simulated data sets.

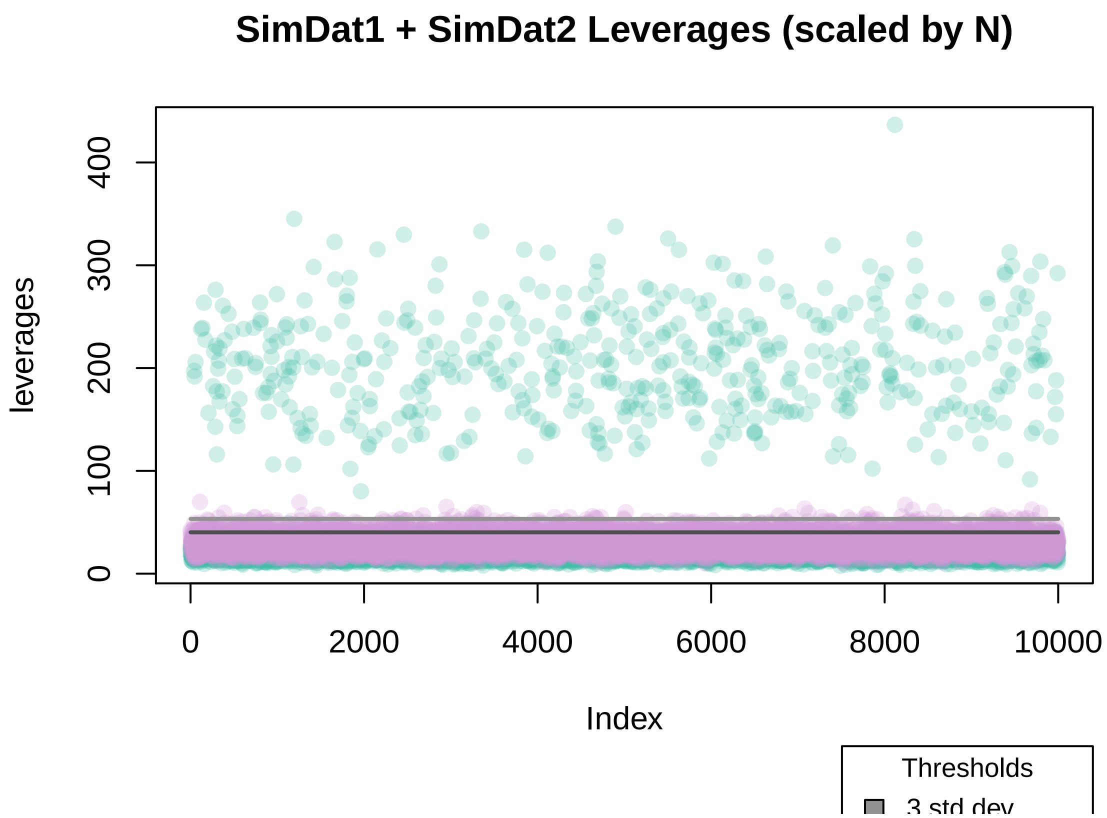


Below is a plot of SimDat1 and SimDat 2. The grey line is the mean plus 3 standard chi-squared deviations. This is the threshold for high leverage. This representation makes it clear that all the values from the distribution with a variance of 10 are classified as *high-leverage*.

```
library(colorspace)
rainbow.cols <- rainbow_hcl(19, alpha = 0.25)

par(mar=c(6, 4.1, 4.1, 4), xpd=TRUE)
plot.leverages <- function(m, leverages, dataset.title, leverages.2 = c()) {
  lev.sd <- sqrt(2*m)
  threshold <- qchisq(0.90, df = m)
  plot(leverages,
       col = rainbow.cols[round(runif(1)*19)],
       pch = 19,
       main = paste(dataset.title, 'Leverages (scaled by N)'))
  if (length(leverages.2 > 0)) {
    points(x = leverages.2,
         col = rainbow.cols[round(runif(1)*19)],
         pch = 19)
  }
  # Draw a line where our threshold should be
  lines(y = rep(threshold, n), x = seq(1,n,1), col = 'gray31', lwd = 2)
  lines(y = rep((3*lev.sd + m), n), x = seq(1,n,1), col = 'gray57', lwd = 2)
  legend('bottomright', inset=c(0,-0.6),
         title='Thresholds',
         c('3 std dev','90th percentile'),
         fill=c('gray57','gray31'), cex = 0.8)
}

plot.leverages(m, simdat2.leverages, 'SimDat1 + SimDat2', simdat1.leverages)
```



5. Find from ML (or otherwise) data repositories (e.g. UCI) a number of big datasets (maximum 10) with continuous output and continuous inputs (maximum work with 50 inputs). For each of these datasets compute the leverages.

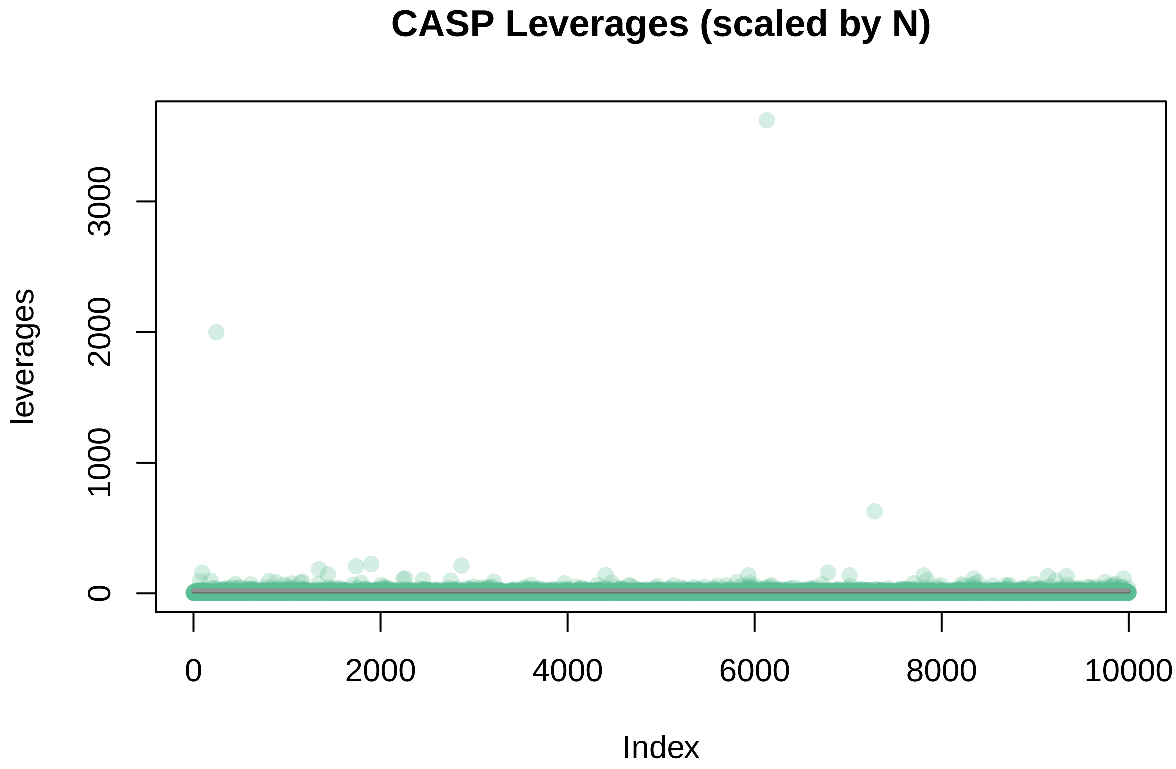
6. The main output of this project is a visualization of the leverage distributions, one for each of the ML datasets as well as SimDat1 and SimDat2, that allows direct comparison between them. The aim is to to uncover structure that might be common in big data sets amenable to regression analysis.

5. Continued

CASP Data: Physicochemical Properties of Protein Tertiary Structure

Dataset Homepage

```
casp.data <- read.csv('~/Projects/data_files/CASP.csv')
casp.features <- casp.data[1:10000,2:ncol(casp.data)]
casp.leverages <- leverages(as.matrix(casp.features))
plot.leverages(ncol(casp.features), casp.leverages, 'CASP')
```

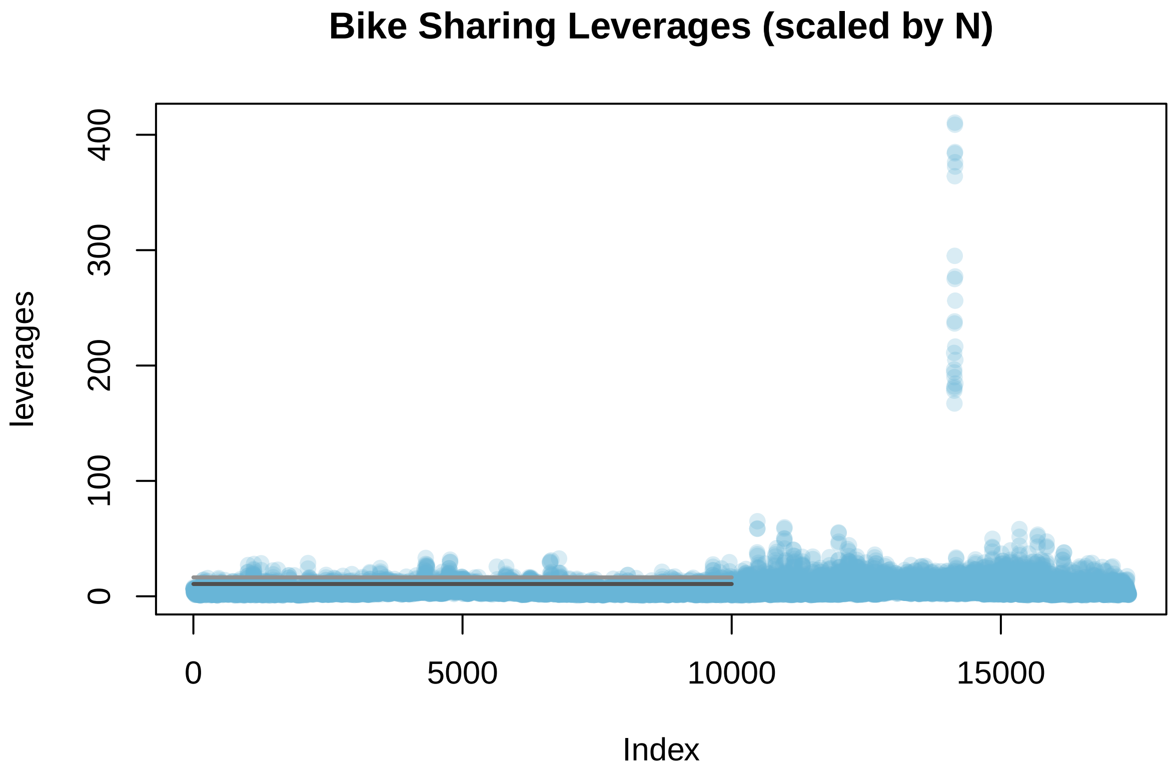


Bike Sharing Dataset

Dataset Homepage

```
bikes.data <- read.csv('~/Projects/data_files/Bike-Sharing-Dataset/hour.csv')
bikes.features <- bikes.data[,c('temp','atemp','hum','windspeed','casual', 'registered')]

bikes.leverages <- leverages(as.matrix(bikes.features))
plot.leverages(ncol(bikes.features), bikes.leverages, 'Bike Sharing')
```



Wine Quality Dataset

Dataset Homepage

```
redwine.data <- read.delim('~/Projects/data_files/wine/winequality-red.csv', sep=';')
redwine.features <- redwine.data[,setdiff(colnames(redwine.data), 'quality')]

redwine.leverages <- leverages(as.matrix(redwine.features))

whitewine.data <- read.delim('~/Projects/data_files/wine/winequality-white.csv', sep=';')
whitewine.features <- whitewine.data[,setdiff(colnames(whitewine.data), 'quality')]

whitewine.leverages <- leverages(as.matrix(whitewine.features))
dev.off()
```

```
## null device
##           1
```

```
par(mfrow = c(2,1))

plot.leverages(ncol(redwine.features), redwine.leverages, 'Red Wine')

plot.leverages(ncol(whitewine.features), whitewine.leverages, 'White Wine')

dev.off()
```

```
## null device
##           1
```

```
par(mfrow = c(1,1))
```

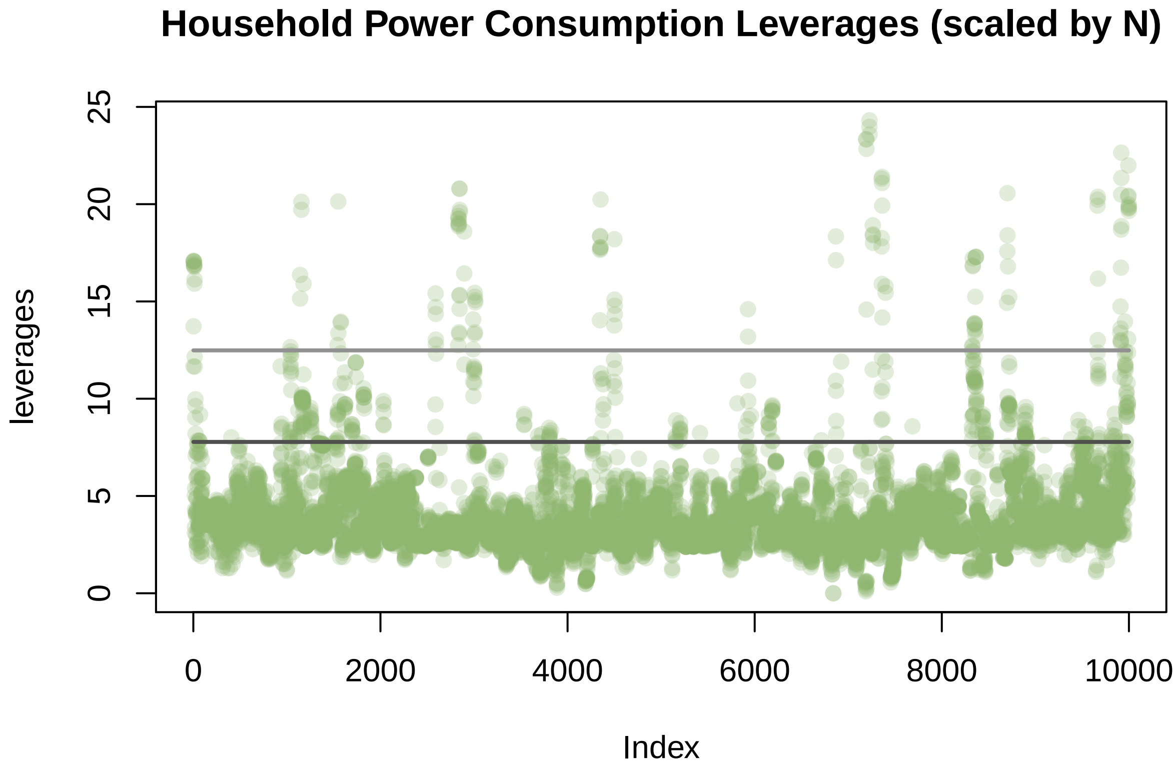
Individual household electric power consumption Data Set

Dataset Homepage

Note: Sequential nature of data clumps high leverage points

```
power.data <- read.delim('~/Projects/data_files/household_power_consumption.txt', nrow=10000, sep = ';'
power.features <- power.data[,c('Global_active_power','Global_reactive_power','Voltage','Global_intensi
power.features <- data.matrix(power.features)
power.leverages <- leverages(power.features)

plot.leverages(ncol(power.features), power.leverages, 'Household Power Consumption')
```

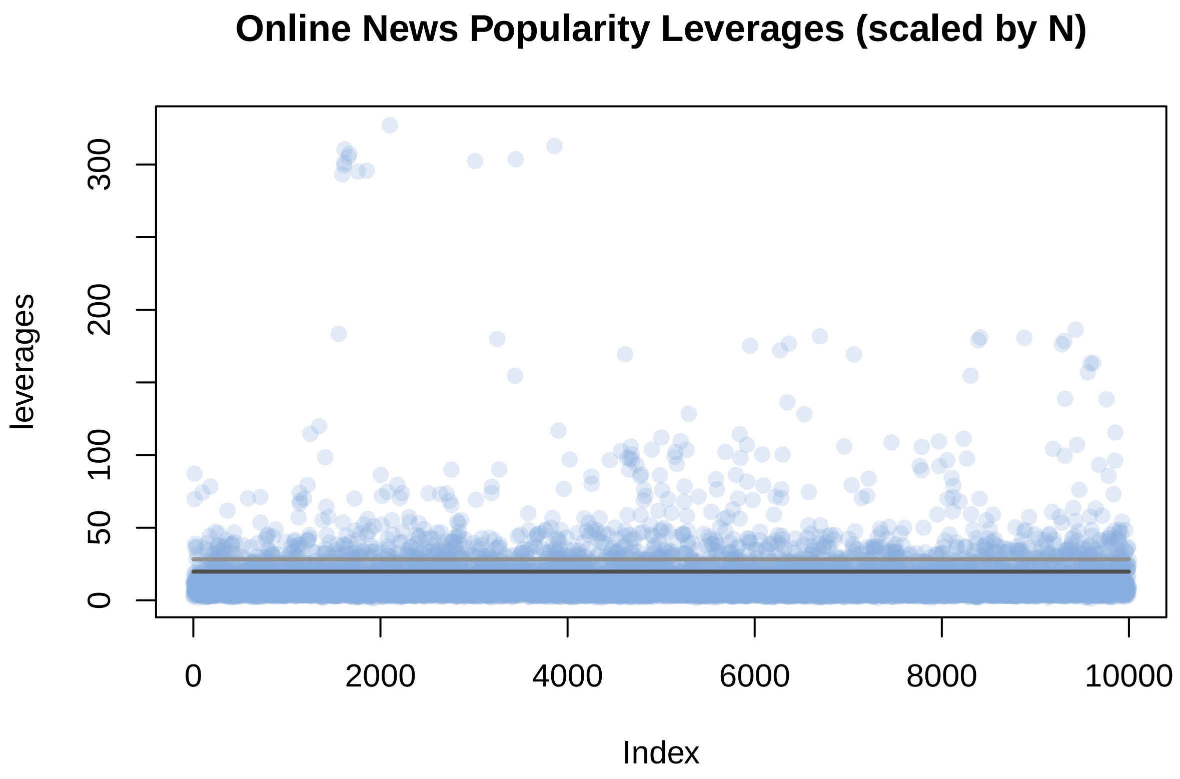


Online News Popularity

```
news.data <- read.csv('~/Projects/data_files/OnlineNewsPopularity/OnlineNewsPopularity.csv', nrow=10000
news.features <- news.data[,c('n_tokens_title', 'n_tokens_content','num_hrefs', 'num_imgs', 'num_videos

news.leverages <- leverages(as.matrix(news.features))

plot.leverages(ncol(news.features), news.leverages, 'Online News Popularity')
```



Final Thoughts

In data which is sequential, often high leverage points are near eachother in the order of rows.