Leverage and big data

Aimee Barciauskas, Harihara Subramanyam, Roger Cuscó

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
library(mvtnorm)
library(scales)
library(psych)
library(MASS)
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60))

# Generate 100k replications of 30 inputs
m <- 30
n <- 100000
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)
```

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.

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

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,]
  }
}
```

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 $i-th$ observation is the value $H_{i,i}$ of the hatmatrix 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 χ^2-squared distribution when the covariance equation:

X^TX

is scaled by the number of observations

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

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

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

These scaled datasets follow a χ^2 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 χ^2- distribution.

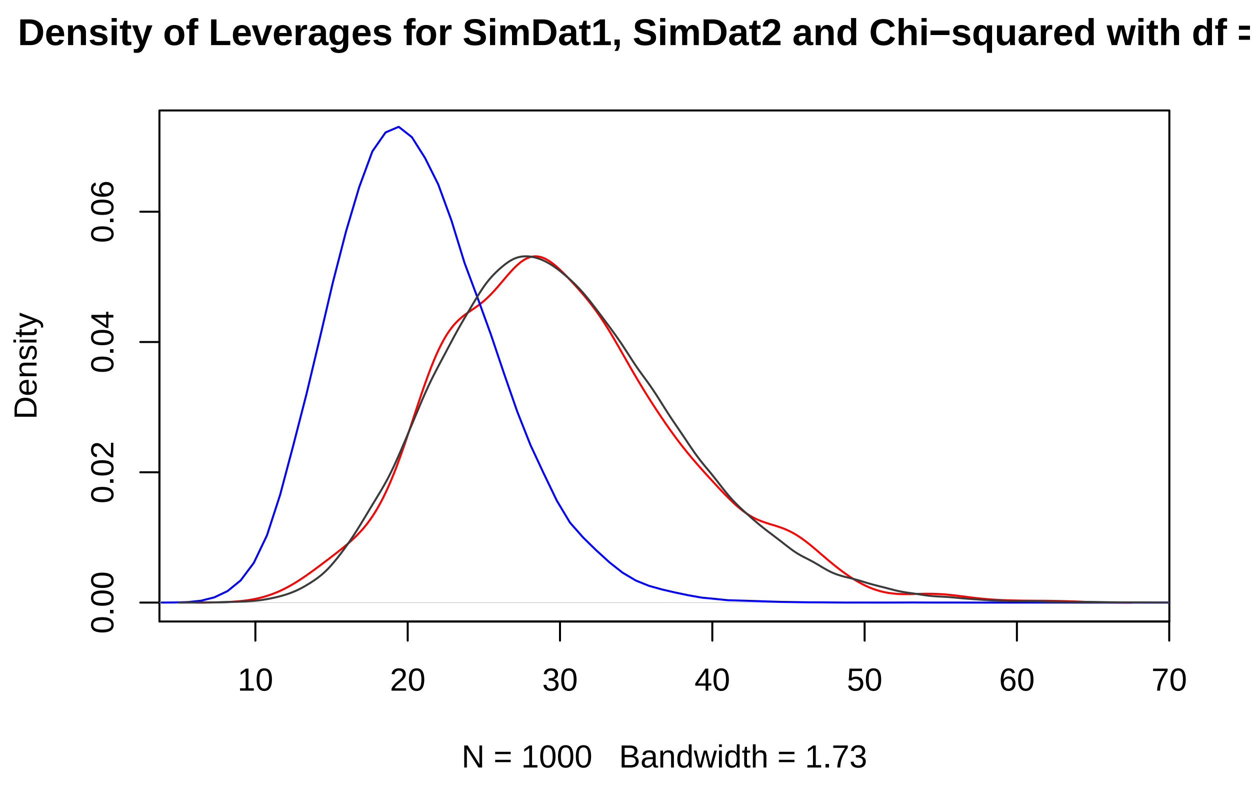
Below we define functions for calculating scaled leverages.

```r
precision.matrix <- function(phi, scaled = TRUE) {
  n <- nrow(phi)
  if (scaled == TRUE) {
    solve((1/n)*(t(phi)%*%phi))
  } else {
    solve((t(phi)%*%phi))
  }
}

hatmatrix <- function(phi, scaled = TRUE) {
  n <- nrow(phi)
  if (scaled == TRUE) {
    phi %*% solve((1/n)*(t(phi)%*%phi)) %*% t(phi)
  } else {
    phi %*% solve((t(phi)%*%phi)) %*% t(phi)
  }
}
leverages <- function(phi, scaled = TRUE) diag(hatmatrix(phi, scaled))

# Unscaled function for calculating precision sequentially
sequential.precision <- function(file, max.size = Inf, only.cols = c()) {
  chunk.index <- 0
  chunk.size <- 1000
  data <- read.csv(file, nrow = chunk.size, skip = chunk.index)
  chunk.index <- chunk.index + chunk.size
  data <- data[,only.cols]
  X.current <- as.matrix(data)

  C.inv.current <- ginv(t(X.current) %*% X.current)
  C.inv.prev <- C.inv.current
  # read file from index to chunk and update vars
  while ((chunk.index+chunk.size) <= max.size) {
    # Read chunk of file as X_{k+1}
    data <- read.csv(file, nrow = chunk.size, skip = chunk.index)
    data <- data[,only.cols]
    X.current <- as.matrix(data)
    print(chunk.index)

    # Update C
    C.inv.current <- C.inv.prev - C.inv.prev %*% t(X.current) %*% ginv(diag(1, nrow(X.current)) + X.cur

    # If size of current chunk is less than chunk size, break
    if (nrow(X.current) < chunk.size) break
    C.inv.prev <- C.inv.current
    chunk.index <- chunk.index + chunk.size
  }
  C.inv.current
}

leverages.sequential <- function(data.mat, prec.mat) {
  leverages <- c()
  for (i in 1:nrow(data.mat)) {
    leverages <- append(leverages, t(data.mat[i,]) %*% prec.mat %*% data.mat[i,])
  }
  leverages
}
```

The functions above are used to calculate the leverages for the simulated data and the leverages are compared with a χ^2-distribution having 30 degrees of freedom.

First, a density plot assures us that the SimDat1 reflects a χ^2-distribution.

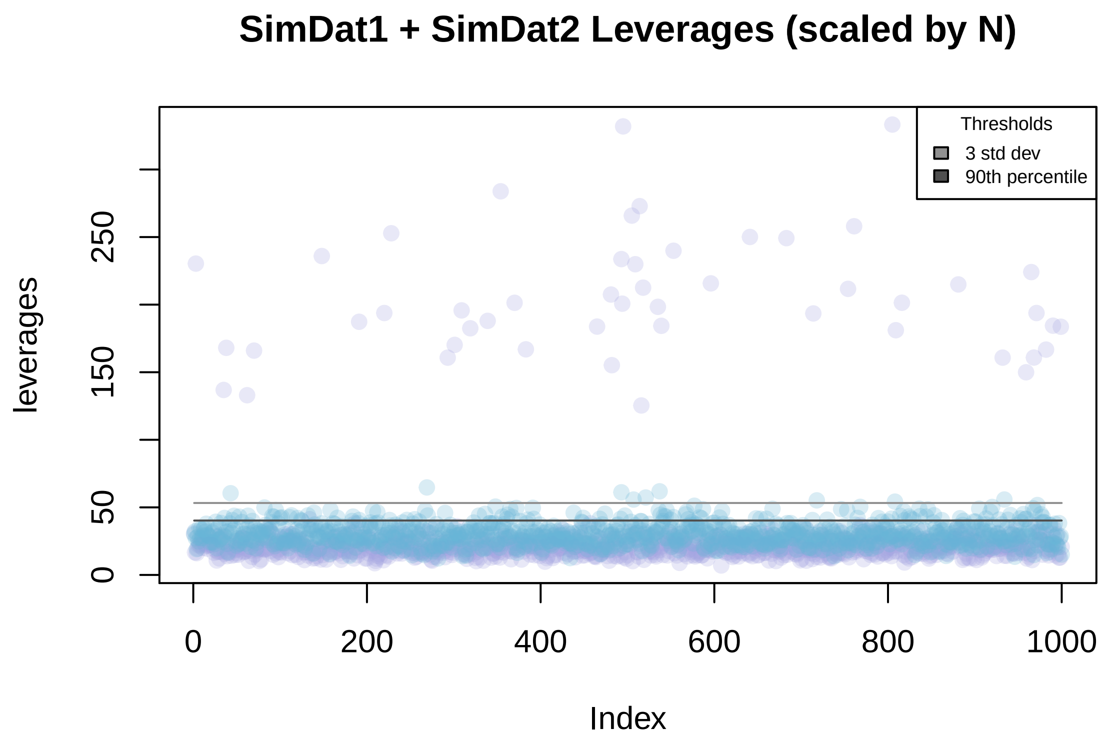
```
simdat1.prec.mat <- precision.matrix(simdat1.xs)
simdat2.prec.mat <- precision.matrix(simdat2.xs)
simdat1.leverages <- leverages.sequential(simdat1.xs, simdat1.prec.mat)
simdat2.leverages <- leverages.sequential(simdat2.xs, simdat2.prec.mat)

plot(density(sample(simdat1.leverages, 1000)), col = 'red', ylim = c(0,max(density(sample(simdat2.levera
lines(density(simdat2.leverages), col = 'blue')

rchis <- rchisq(n, df = 30)

lines(density(rchis), col = 'grey23')
```



To visualize which data points are qualified as *high-leverages*, we plotof SimDat1 and SimDat 2 with grey lines marking for the comparison χ^2 distribution, 3 standard deviations above the mean and the 90th precentile. Three standard deviations and the 90th precentile provide 2 options for qualifying data points as high leverage.

In this plot you can clearly see that nearly all the values above the 3 standard deviations threshold are from the distribution with a variance of 10. It also appears that there is a higher mean value for SimDat1. This makes mathematical sense: given all the leverages should add up to the number of features, if a data set has a few very high leverage observations, the others will be necessarily lower.

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

par(mar=c(6, 4.1, 4.1, 4), xpd=TRUE)

plot.leverages.density <- function(m, leverages, dataset.title) {
  n <- length(leverages)
  rchisqs <- rchisq(1000, df = m)

  plot(density(sample(leverages, 1000)), col = 'red', main = paste0('Density of Leverages for ', datase

  lines(density(rchisqs), col = 'grey23')
}

plot.leverages <- function(m, leverages, dataset.title, leverages.2 = c()) {
  n <- length(leverages)
  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)
  }
  # Thresholds
  lines(y = rep(threshold, n), x = seq(1,n,1), col = 'gray31', lwd = 1)
  lines(y = rep((3*lev.sd + m), n), x = seq(1,n,1), col = 'gray57', lwd = 1)
  legend('topright',
         title='Thresholds',
         c('3 std dev','90th percentile'),
         fill=c('gray57','gray31'), cex = 0.6)
}

plot.leverages(m, sample(simdat2.leverages, 1000), 'SimDat1 + SimDat2', sample(simdat1.leverages, 1000)
```



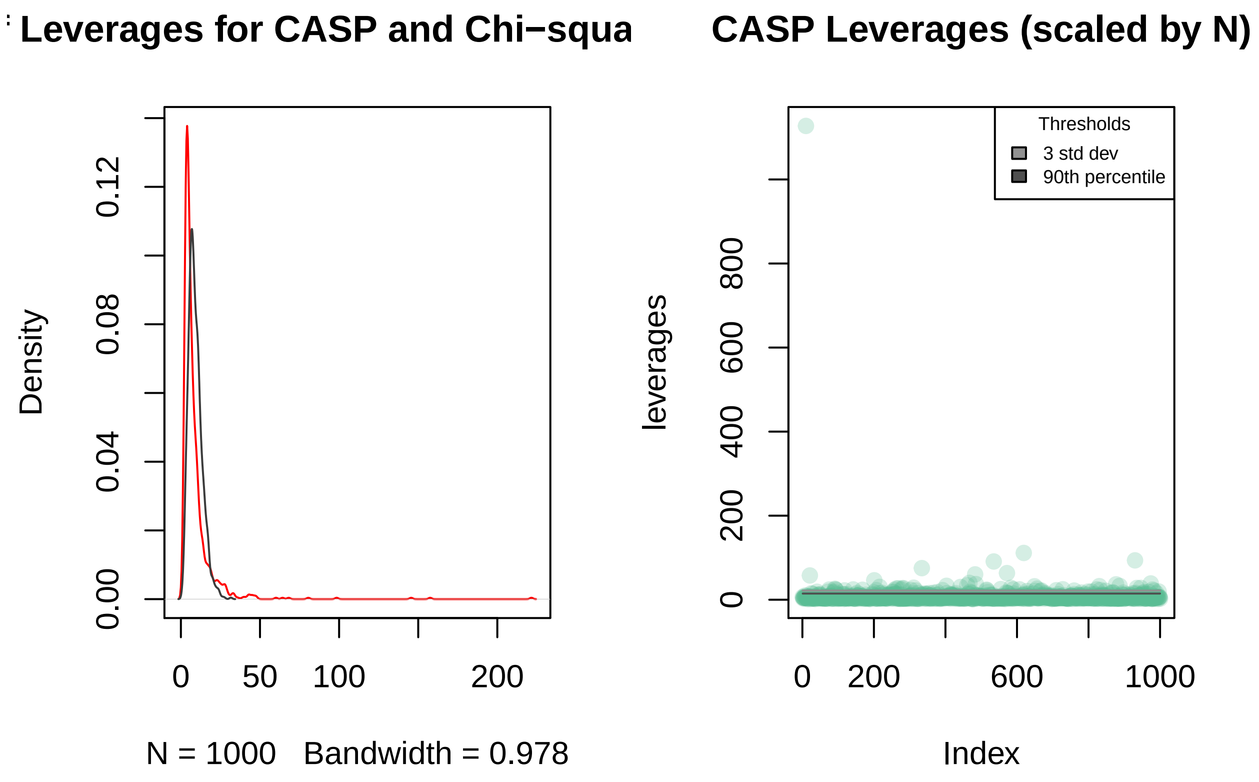
5. Find from ML (or otherwise) data repositories (e.g. UCI) a number of big datasets (maximum 10) with continuous output and continuous inputs (maximum work with 50 inputs). For each of these datasets compute the leverages.

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.

CASP Data: Physicochemical Properties of Protein Tertiary Structure

Dataset Homepage

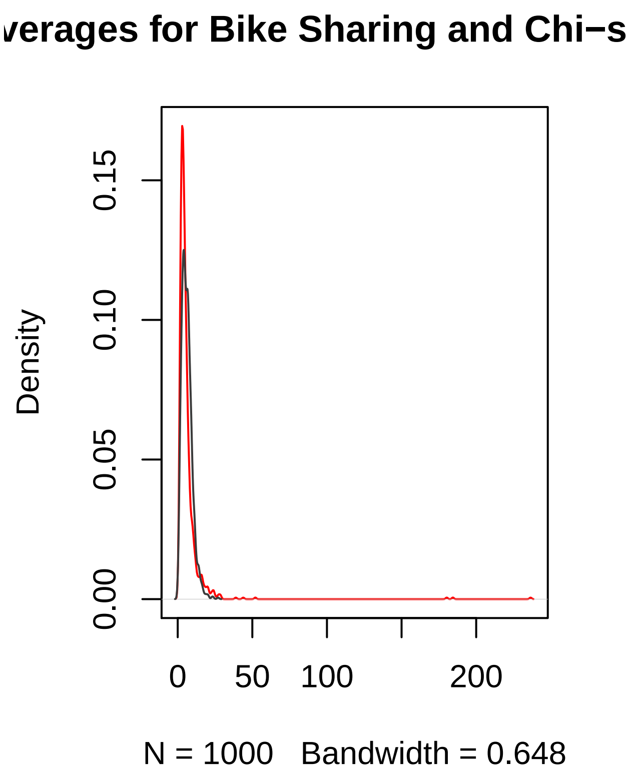
```
casp.data <- read.csv('~/Projects/data_files/CASP.csv')
casp.features <- as.matrix(casp.data[,2:ncol(casp.data)])
casp.prec.mat <- precision.matrix(casp.features)
casp.leverages <- leverages.sequential(casp.features, casp.prec.mat)

# Sanity check - should equal number of features
# nfeatures <- sum(casp.leverages/nrow(casp.features))

par(mfrow=c(1,2))
plot.leverages.density(ncol(casp.features), casp.leverages, 'CASP')
plot.leverages(ncol(casp.features), sample(casp.leverages, 1000), 'CASP')
```

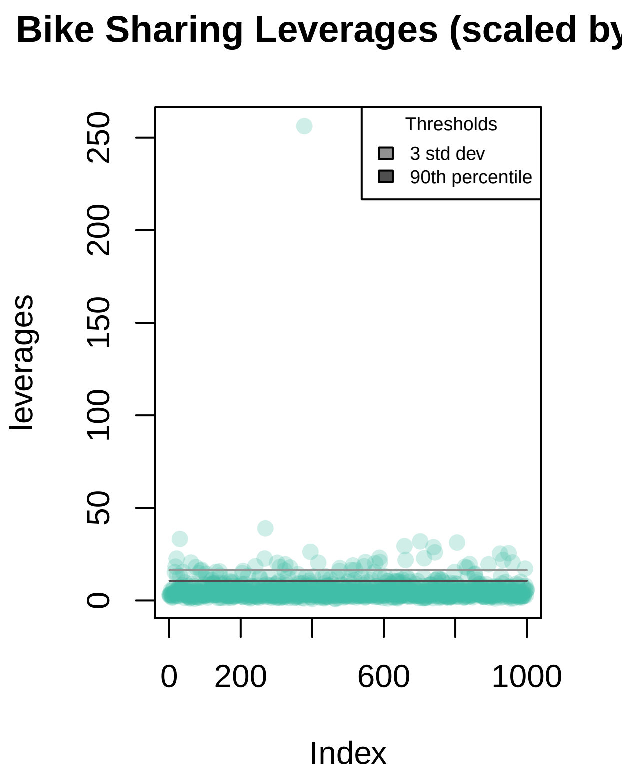


Bike Sharing Dataset

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))

par(mfrow=c(1,2))
plot.leverages.density(ncol(bikes.features), sample(bikes.leverages, 1000), 'Bike Sharing')
plot.leverages(ncol(bikes.features), sample(bikes.leverages, 1000), 'Bike Sharing')
```

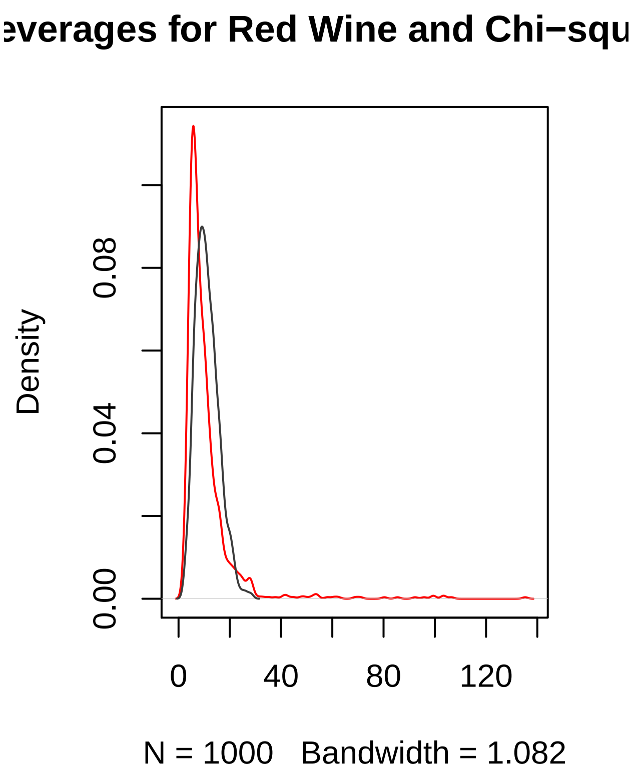


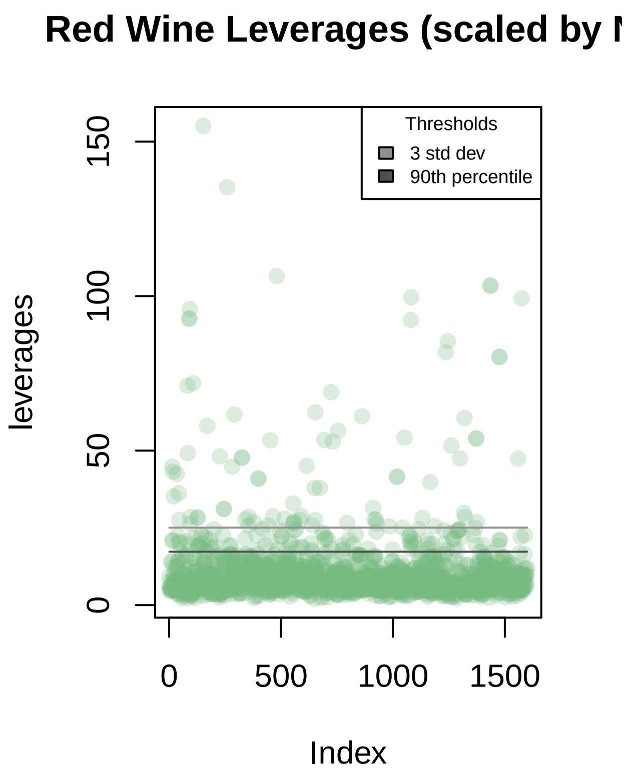


Wine Quality Dataset

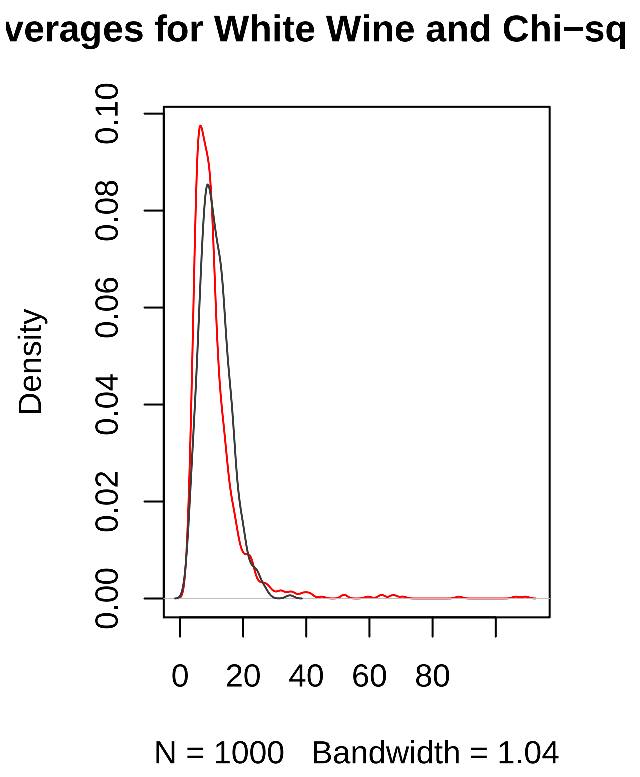
Dataset Homepage

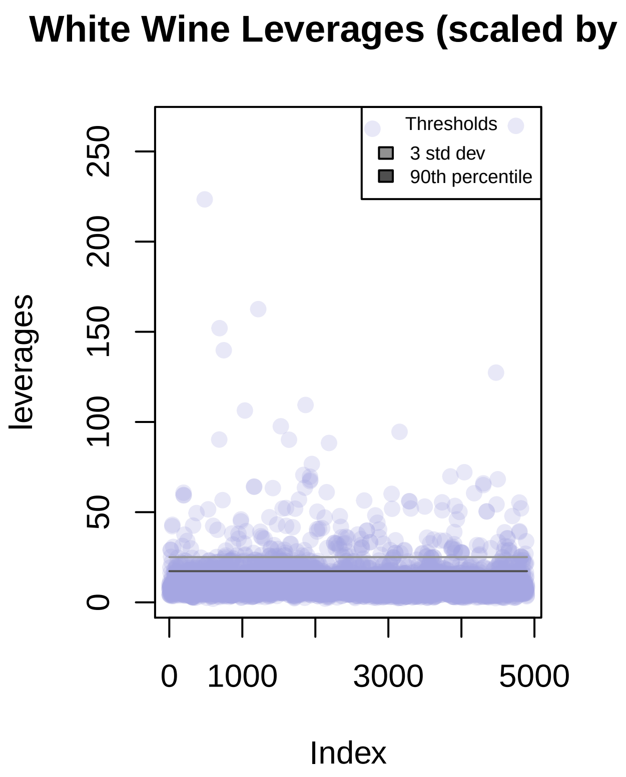
```
redwine.data <- read.delim('~/Projects/data_files/wine/winequality-red.csv', sep=';')
redwine.features <- redwine.data[,setdiff(colnames(redwine.data), 'quality')]
redwine.leverages <- leverages(as.matrix(redwine.features))

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))

par(mfrow=c(1,2))
plot.leverages.density(ncol(redwine.features), redwine.leverages, 'Red Wine')
plot.leverages(ncol(redwine.features), redwine.leverages, 'Red Wine')
```





```r
par(mfrow=c(1,2))
plot.leverages.density(ncol(whitewine.features), whitewine.leverages, 'White Wine')
plot.leverages(ncol(whitewine.features), whitewine.leverages, 'White Wine')
```

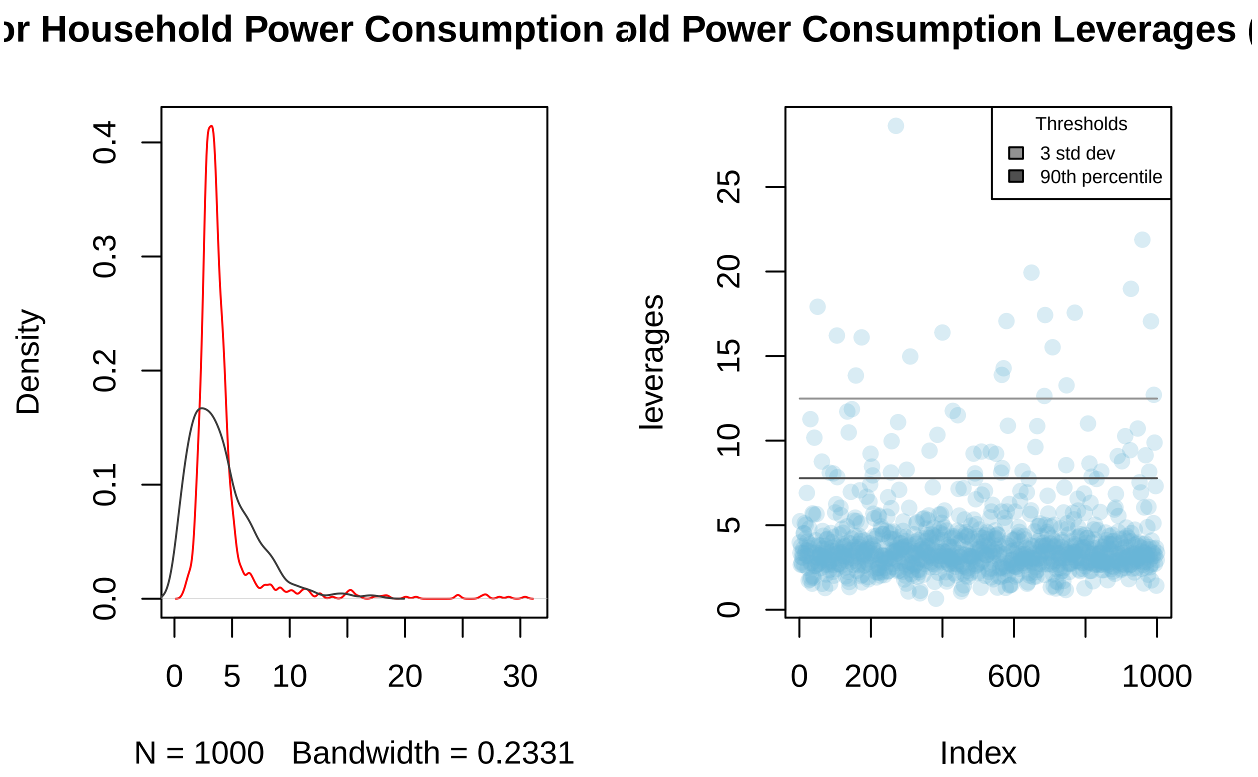




Individual household electric power consumption Data Set

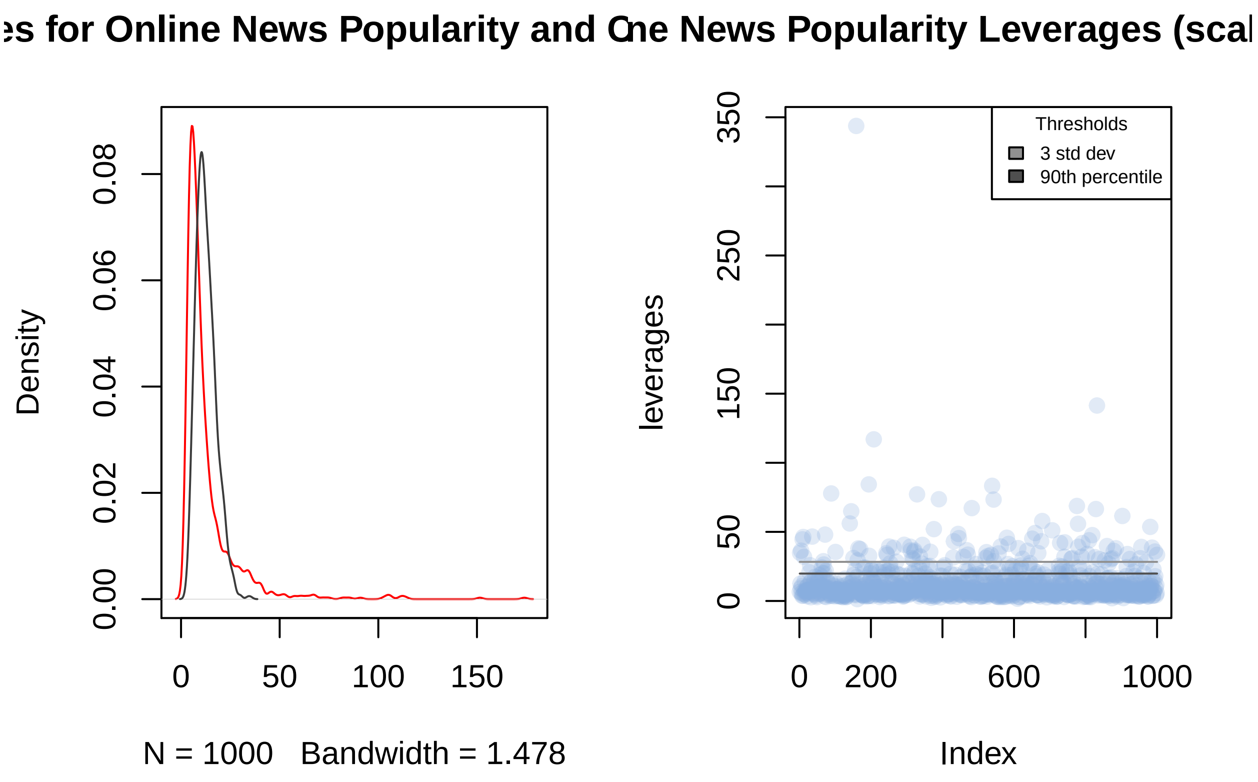
Dataset Homepage

```
power.data <- read.delim('~/Projects/data_files/household_power_consumption.txt', sep = ';')
power.features <- power.data[1:50000,c('Global_active_power','Global_reactive_power','Voltage','Global_
power.features <- data.matrix(power.features)
power.prec.mat <- precision.matrix(power.features)

power.leverages <- leverages.sequential(power.features, power.prec.mat)

par(mfrow=c(1,2))
plot.leverages.density(ncol(power.features), sample(power.leverages, 1000), 'Household Power Consumptio
plot.leverages(ncol(power.features), sample(power.leverages, 1000), 'Household Power Consumption')
```



Online News Popularity

```
news.data <- read.csv('~/Projects/data_files/OnlineNewsPopularity/OnlineNewsPopularity.csv')
news.features <- news.data[,c('n_tokens_title', 'n_tokens_content','num_hrefs', 'num_imgs', 'num_videos
news.prec.mat <- precision.matrix(as.matrix(news.features))
news.leverages <- leverages.sequential(as.matrix(news.features), news.prec.mat)

par(mfrow=c(1,2))
plot.leverages.density(ncol(news.features), news.leverages, 'Online News Popularity')
plot.leverages(ncol(news.features), sample(news.leverages, 1000), 'Online News Popularity')
```



Conclusions

Scaling leverages by the number of observations enables comparison with an expected distribution of the leverages. The χ^2 distribution can be used to inspect if the distribution of leverages is "balanced" or has many extreme values.

The real datasets above demonstrated varying levels of discrepency with the comparison distributions. A commonality was a lower mean and higher variance. This relationship makes sense given leverages must add up to the number of parameters, and thus a few extreme high-leverages must be "balanced out" by a lower mean for the remaining observations.

It is also worth noting that in time-series or other sequential data, it is reasonable to observe