Week 4: Kernel Density Estimation

MATH-517 Statistical Computation and Visualization

Tomas Masak

October 14th 2022

The Problem

Setup: $X_1, \dots, X_n$ is a random sample from a density $f(x)$

Goal: Estimate f nonparametrically.

We already know of **histogram**, which requires a specification of

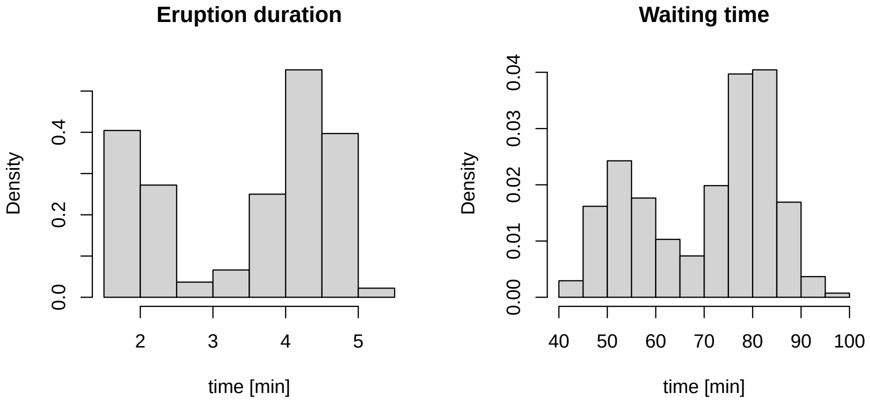
- *origin* and *binwidth*, or
- *breaks* - more general, but non-equidistant binning is bad anyway, so think only about origin and bindwidth

Runnning Ex.: Yellowstone's Old Faithful geyser - `faithful` data:

- `waiting` - time between eruptions
- `eruptions` - duration of the eruptions

Basic Histogram

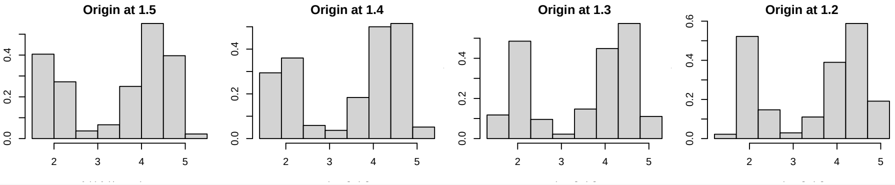
```r
data(faithful)
par(mfrow=c(1,2))
hist(faithful$eruptions, probability=T, main = "Eruption duration", xlab="time [min]")
hist(faithful$waiting, probability=T, main = "Waiting time", xlab="time [min]")
```



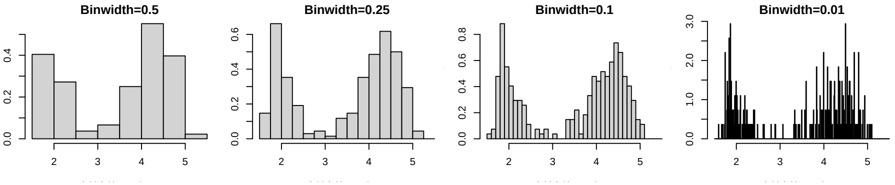
- breaks specified, so a rule of thumb used to choose origin and binwidth

Changin Origin and Binwidth

```r
par(mfrow=c(1,4), mar = c(3.2, 3, 1.6, 0.2)) # reduce the white space around individual plots
hist(faithful$eruptions, probability=T, main="Origin at 1.5");
hist(faithful$eruptions, breaks=seq(1.4,5.4,by=0.5), probability=T, main="Origin at 1.4", xlab="time [min]")
hist(faithful$eruptions, breaks=seq(1.3,5.3,by=0.5), probability=T, main="Origin at 1.3", xlab="time [min]")
hist(faithful$eruptions, breaks=seq(1.2,5.2,by=0.5), probability=T, main="Origin at 1.2", xlab="time [min]")
```




```r
par(mfrow=c(1,4), mar = c(3.2, 3, 1.6, 0.2))
hist(faithful$eruptions, probability=T, main="Binwidth=0.5")
hist(faithful$eruptions, breaks=seq(1.5,5.5,by=0.25), probability=T, main="Binwidth=0.25")
hist(faithful$eruptions, breaks=seq(1.5,5.5,by=0.1), probability=T, main="Binwidth=0.1")
hist(faithful$eruptions, breaks=seq(1.5,5.5,by=0.01), probability=T, main="Binwidth=0.01")
```



Issues with Histogram

Histogram is great for quick visualization, but does not pass as a density estimator.

- *origin* is completely arbitrary
- *binwidth* relates to smoothness of f, but histogram cannot be smooth anyway

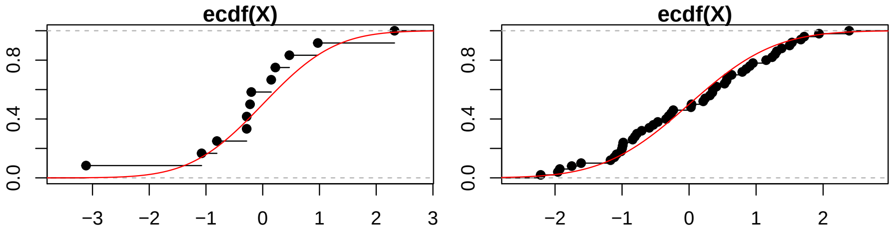
Let us now address these two issues by a naive version of kernel density estimation (KDE).

Prerequisite: empirical (cumulative) distribution function (ECDF):

$$\widehat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{[X_i \leq x]}$$

ECDF

```r
edf_plot <- function(N){
  X <- rnorm(N)
  EDF <- ecdf(X)
  plot(EDF)
  x <- seq(-4,4,by=0.01)
  points(x,pnorm(x),type="l",col="red")
}
set.seed(517)
par(mfrow=c(1,2), mar=c(2,2,1,1))
edf_plot(12)
edf_plot(50)
```



Naive KDE

- The ECDF $\widehat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{[X_i \leq x]}$ is an estimator of F
  - by Glivenko-Cantelli theorem uniformly almost surely consistent:

$$\sup_x |\widehat{F}(x) - F(x)| \overset{a.s.}{\to} 0$$

- f is the derivative F: $f(x) = \lim_{h \to 0_+} \frac{F(x+h) - F(x-h)}{2h}$
- Fix $h = h_n$ as something small depending on n and plug it in:

$$\widehat{f}(x) = \frac{\widehat{F}_n(x + h_n) - \widehat{F}_n(x - h_n)}{2h_n}$$

- we need $h_n \to 0_+$ for $n \to 0$ in order to have any hope in consistency

Consistency

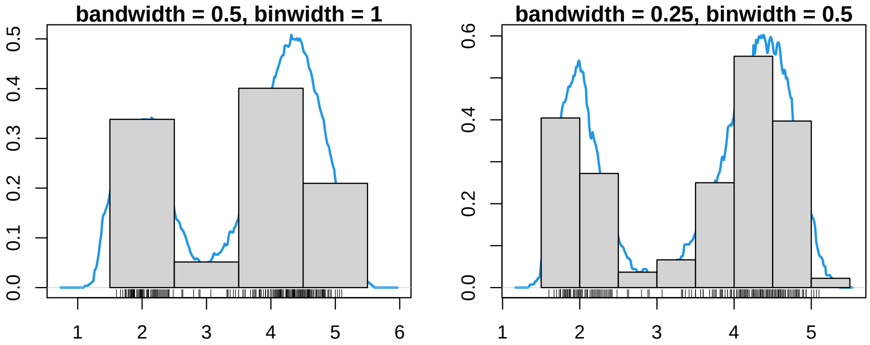
- $\mathbb{E}\widehat{f}(x) = \frac{F(x+h_n)-F(x-h_n)}{2h_n} \to f(x) \quad \text{if} \quad h_n \to 0_+$
- since $\widehat{f}(x) = \frac{1}{2nh_n}\sum_{i=1}^n \quad \underbrace{\mathbb{I}_{\left[X_i \in (x-h_n, x+h_n]\right]}}_{Ber\left(F(x+h_n)-F(x-h_n)\right)} \quad :$

$$\operatorname{var}\left(\widehat{f}(x)\right) = \frac{1}{4nh_n^2}\left[F(x+h_n) - F(x-h_n)\right]\left[1 - F(x+h_n) + F(x-h_n)\right]$$
$$= \frac{F(x+h_n) - F(x-h_n)}{2h_n}\frac{1 - F(x+h_n) + F(x-h_n)}{2nh_n} \to 0$$

$\Rightarrow$ consistency if $h_n \to 0$ and $nh_n \to \infty$

Naive KDE ≡ Moving Histogram

- when *binwidth* for the histogram is taken as $2h$, the naive KDE gives exactly the histogram value in the middle of every bin
  - *origin* does not matter anymore



Naive KDE Rewritten

The naive KDE can be written as

$$\begin{aligned}\widehat{f}(x) &= \frac{1}{2nh_n} \sum_{i=1}^{n} \mathbb{I}_{\left[X_i \in (x-h_n, x+h_n]\right]} \\ &= \frac{1}{2nh_n} \sum_{i=1}^{n} \mathbb{I}_{\left[1 \leq \frac{X_i - x}{h_n} \leq 1]\right]} \\ &= \frac{1}{nh_n} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h_n}\right)\end{aligned}$$

where $K(\cdot)$ is the density of $U[-1,1]$.

Next step: replace $K(\cdot)$ for something else.

KDE - Definition and Properties

Definition. KDE of f based on $X_1, \ldots, X_N$ is

$$\widehat{f}(x) = \frac{1}{nh_n} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h_n}\right),$$

where the **kernel** $K(\cdot)$ satisfies:

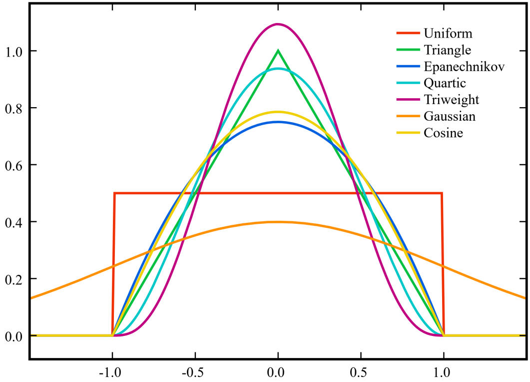
1. $K(x) \geq 0$ for all $x \in \mathbb{R}$
2. $K(-x) = K(x)$ for all $x \in \mathbb{R}$
3. $\int_{\mathbb{R}} K(x) dx = 1$

1. $\lim_{|x| \to \infty} |x| K(x) = 0$
2. $\sup_x |K(x)| < \infty$

- $K(\cdot)$ is usually taken to be a density, and the assumptions
  - 1-3 hold if it is symmetric
  - 4 holds if it has a finite absolute moment
  - 5 holds if it is uniformly bounded
- if $h_n \to 0$ and $nh_n \to \infty$ we have pointwise consistency
  - we will show this in a bit
  - also uniform consistency, but tricky to show

Common Kernels

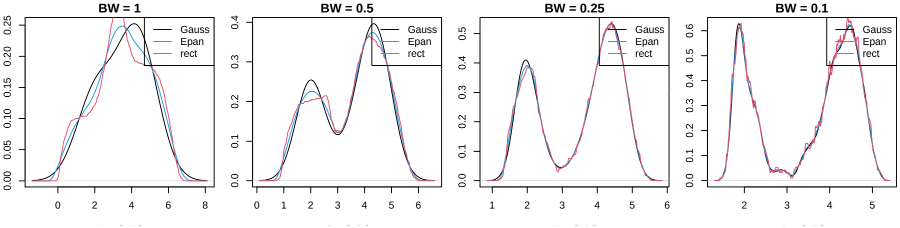
Kernel Name	Formula
Epanechnikov	$K(x) \propto (1 - x^2)\mathbb{I}_{[\lvert x \rvert \leq 1]}$
Tricube (a.k.a. Triweight)	$K(x) \propto (1 - \lvert x \rvert^3)^3\mathbb{I}_{[\lvert x \rvert \leq 1]}$
Gaussian	$K(x) \propto \exp(-x^2/2)$
...	...



Bandwidth > Kernel

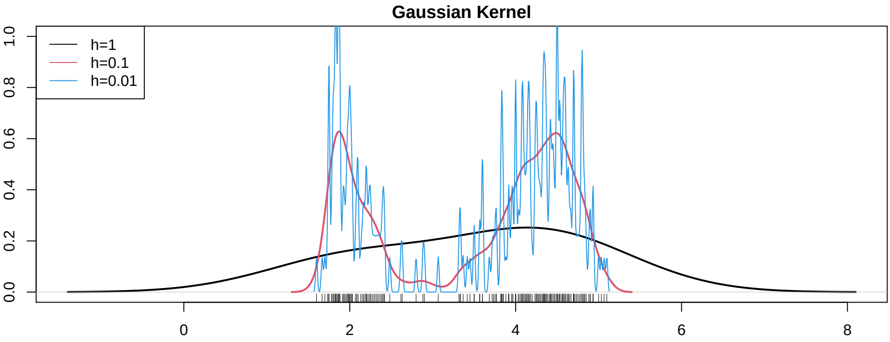
While there is some improvement when not using the naive rectangular kernel, choosing good bandwidth h is much more important.

```r
# This is only a handle, not a proper function! Why?
plot_kdes <- function(bw){
  plot(density(faithful$eruptions, kernel="gaussian", bw=bw),
       main=paste("BW = ",bw,sep=""), xlab="time [min]")
  lines(density(faithful$eruptions, kernel="epanechnikov", bw=bw), col=4)
  lines(density(faithful$eruptions, kernel="rectangular", bw=bw), col=2)
  legend("topright", col=c(1, 4, 2), lty=1, legend=c("Gauss", "Epan", "rect"))
}
par(mfrow=c(1,4), mar = c(3.2, 3, 1.6, 0.2))
plot_kdes(1); plot_kdes(0.5); plot_kdes(0.25); plot_kdes(0.1)
```



Bandwidth

- the bandwidth h is a *tuning parameter* and needs to be chosen somehow in practice
  - h small $\rightarrow$ wiggly estimator
  - h large $\rightarrow$ slowly-varying estimator



Bias-variance Trade-off

Goal: choose the tuning parameter h so that the mean squared error of the estimator is minimized:

$$\underbrace{\mathbb{E}\big[\widehat{f}(x) - f(x)\big]^2}_{MSE} = \mathbb{E}\big[\widehat{f}(x) \pm \mathbb{E}\widehat{f}(x) - f(x)\big]^2 = \underbrace{\big[\mathbb{E}\widehat{f}(x) - f(x)\big]^2}_{bias^2} + \underbrace{\mathrm{var}\big(\widehat{f}(x)\big)}_{variance}$$

Blackboard calculations (available in the lecture notes) give

$$\mathrm{bias}(\widehat{f}(x)) = \frac{1}{2}h^2 f''(x) \int z^2 K(z) dz + o(h^2)$$
$$\mathrm{var}\big(\widehat{f}(x)\big) = \frac{1}{nh} f(x) \int \big[K(z)\big]^2 dz + o\left(\frac{1}{nh}\right)$$

This shows consistency for $h = h_n \to 0$ and $nh_n \to \infty$ and encapsulates the trade-off:

- small $h \Rightarrow$ small bias but large variance
- large $h \Rightarrow$ large bias but small variance

Optimal Bandwidth

Plugging this back in the MSE formula ignoring the *little-o* terms, deriving the MSE by h and setting it to zero leads to asymptotically optimal bandwidth choice:

$$h_{opt}(x) = n^{-1/5} \left(\frac{f(x) \int K(z)^2 dz}{\big[f''(x)\big]^2 \big[\int z^2 K(z) dz\big]^2} \right)^{-1/5}$$

1. $h_{opt}(x)$ is a local choice - depends on x
2. global choice can be obtained by integrating out x
3. $h_{opt}(x)$ cannot be directly used, since depends on the unknown f
   - reference method: assume a known f for this formula
   - two-step method: f in the formula estimated by a pilot fit (e.g. visually appealing one)
4. $h_{opt}(x) \asymp n^{-1/5}$ and with this choice

$$MSE \asymp bias^2 \asymp variance = \mathcal{O}(n^{-4/5})$$

   - optimal non-parametric rate

Section 1

Computational Considerations

Computational Considerations

Evaluating

$$\widehat{f}(x_j) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{X_i - x_j}{h}\right)$$

on a grid of points $x_1, \ldots, x_m$ takes naively $\mathcal{O}(mn)$.

- for $n \asymp m$, this means quadratic complexity $\mathcal{O}(n^2)$
- we will show how to reduce this to log-linear complexity $\mathcal{O}(n \log n)$

Circulants and DFT

Definition: A matrix $\mathbf{C} = (c_{jk})_{j,k=0}^{p-1} \in \mathbb{R}^{p \times p}$ is called *circulant* if $c_{jk} = c_{|j-k|}$, where $\mathbf{c} = (c_j) \in \mathbb{R}^p$ is the *symbol* of $\mathbf{C}$ (the first column of $\mathbf{C}$).

$$C = \begin{bmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{bmatrix}$$

Definition: The *discrete Fourier basis* in $\mathbf{R}^p$ is (in the columns of) the matrix $\mathbf{E} = (e_{jk})_{j,k=0}^{p-1} \in \mathbb{R}^{p \times p}$ with entries given by

$$e_{jk} = \frac{1}{\sqrt{p}} e^{2\pi ijk/p}, \qquad j,k = 0,\ldots,p-1$$

- straightforward to check that $\mathbf{E}$ is unitary $\Rightarrow$ really a basis.

Circulants and DFT

Definition: The *discrete Fourier transform* (DFT) of $\mathbf{x} \in \mathbb{R}^p$ is $\mathbf{E}^*\mathbf{x}$ with $\mathbf{E}$ being the discrete Fourier basis from the previous definition. The inverse DFT is the same without the complex conjugate $\mathbf{E}\mathbf{x}$.

FFT: Any algorithm allowing to perform DFT of $\mathbf{x} \in \mathbb{R}^p$ with the log-linear complexity $\mathcal{O}(p \log p)$ is referred to as the fast Fourier transform (FFT).

- think of a function `fft(x)` that returns $\mathbf{E}^*\mathbf{x}$
- the original algorithm due to John W. Tukey

Circulants and DFT

Claim: Circulant matrices are diagonalizable by the DFT. Specifically, the eigendecomposition of a circulant matrix $\mathbf{C}$ (with a symbol $\mathbf{c}$) is given by $\mathbf{C} = \mathbf{E}\mathrm{diag}(\mathbf{q})\mathbf{E}^*$, where $\mathbf{q} = \mathbf{E}^*\mathbf{c}$.

Proof: In the lecture notes, if interested.

Now we know that every circulant matrix can be applied efficiently thanks to the FFT:

$$\mathbf{Cx} = \underbrace{\mathbf{E} \quad \underbrace{\mathrm{diag}\Big(\underbrace{\mathbf{q}}_{=\mathrm{FFT}(\mathbf{c})} \Big) \underbrace{\mathbf{E}^*\mathbf{x}}_{=\mathrm{FFT}(\mathbf{x})}}_{\text{entry-wise prod of those 2 vectors}}}_{\text{inverse FFT of that product}}$$

KDE after Initial Histogram

$$\widehat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{X_i - x_j}{h}\right)$$

Round data $X_1, \ldots, X_n \in (a, b]$ to a common equidistant grid $a = t_0 < t_1 < \ldots < t_{p-1} < t_P = b$

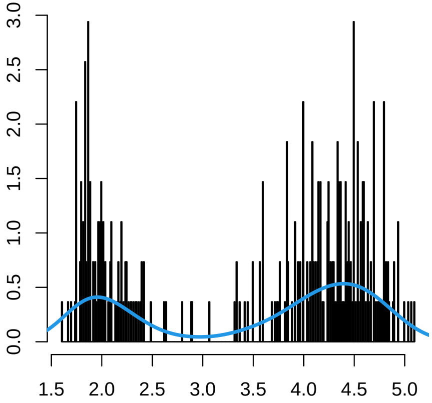
- let $\widetilde{X}_1, \ldots, \widetilde{X}_n$ denote rounded data and $\mathbf{y} \in \mathbb{R}^p$ the counts, i.e. $y_j = \sum_{i=1}^{n} \mathbb{I}_{\left[X_i \in (t_{i-1}, t_i]\right]}$
- this is nothing but a histogram (only an initial one, with a small binwidth)

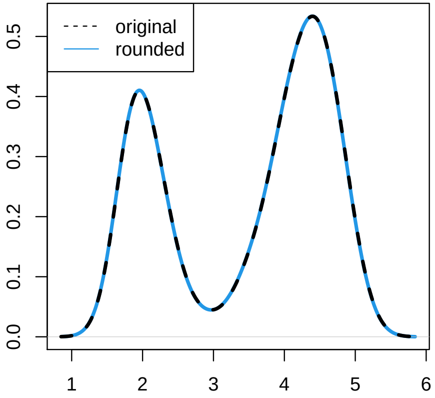
KDE of the rounded data $\equiv$ linear smoother of the initial histogram:

$$\widehat{f}(x) = \frac{1}{nh_n} \sum_{i=1}^{n} K\left(\frac{\widetilde{X}_i - x}{h_n}\right) = \frac{1}{nh_n} \sum_{j=1}^{p} K\left(\frac{t_j - x}{h_n}\right) y_j$$

Effect of Rounding is Small

By rounding, we only commit negligible (numerical) error:





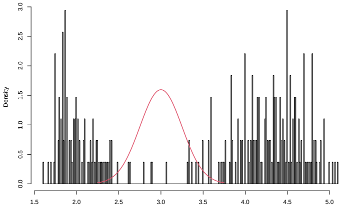
KDE as a Linear Smoother

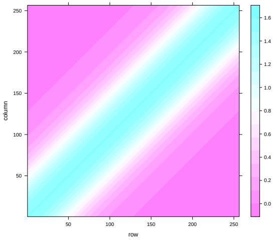
$$\widehat{f}(x) = \frac{1}{nh_n} \sum_{j=1}^{p} K\left(\frac{t_j - x}{h_n}\right) y_j$$

Say we want to evaluate $\widehat{f}$ on the grid $t_1, \ldots, t_p$. Then

$$\left(\widehat{f}(t_1), \ldots, \widehat{f}(t_p)\right)^\top = \mathbf{S} y, \quad \text{with} \quad s_{ij} = \frac{1}{nh_n} K\left(\frac{t_j - t_i}{h_n}\right)$$

- matrix transformation of the input (here $\mathbf{y}$) $\equiv$ linear smoother



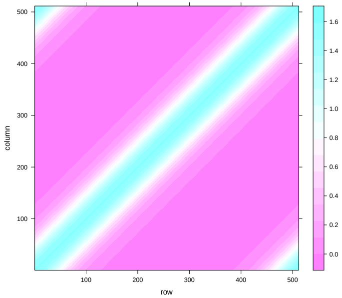


Toeplitz into Circulant

The hat matrix $\mathbf{S}$ is a Toeplitz (stationary) matrix.

Any Toeplitz matrix $\mathbf{S}$ of dimensions $n \times n$ can be embedded into a circulant matrix $\mathbf{C}$ of dimensions at most $(2n-1) \times (2n-1)$. The easiest way is to wrap the first row of $\mathbf{S}$, denoted $\mathbf{s}$, to form the first row of $\mathbf{C}$ as

$$\mathbf{c} = \left(s_1, s_2, \ldots, s_{n-1}, s_n, s_{n-1}, \ldots, s_2\right)^\top$$



Toeplitz and FFT

Calculating $\mathbf{Sy}$ with $\mathbf{S}$ Toeplitz can be done fast by

- embedding $\mathbf{S}$ into a circulant $\mathbf{C}$
- noticing that

$$\mathbf{C}\begin{pmatrix}\mathbf{y}\\ \mathbf{0}\end{pmatrix} = \left(\begin{array}{c|c}\mathbf{S} & \cdot \\ \hline \cdot & \cdot\end{array}\right)\left(\begin{array}{c}\mathbf{y} \\ \hline \mathbf{0}\end{array}\right) = \left(\begin{array}{c}\mathbf{Sy} \\ \hline \cdot\end{array}\right)$$

- calculating $\mathbf{Cy}$ using FFT

Summary - Computations

Fast KDE calculation:

1. round up the original data to a common equidistant grid
    - equivalent to calculating initial histogram (with a small binwidth)
    - KDE is now reduced to a linear smoother with a Toeplitz hat matrix
2. Embed the Toeplitz hat matrix into a circulant matrix
3. Use FFT to apply the circulant matrix

From a high-level point of view, disregarding FFT:

- rounding data induces structure (Toeplitz)
- this structure can be used to speed up computations

Summary - Overall

Motivation:

1. On Week 2, we introduced the histogram as a data exploratory tool.
2. On Week 3, we noticed some issues of the histogram.
3. Histogram is a poor estimator of density, because it
   - is never smooth and requires a choice of *origin*
4. Today, we introduced naive KDE by generalizing histogram to its *origin*-free version.
5. Then, we generalized naive KDE by allowing for better kernels.
6. Now we have a decent nonparametric density estimation tool: KDE.
   - in exploratory analysis, histograms often overlaid with KDEs.

Main takeaways:

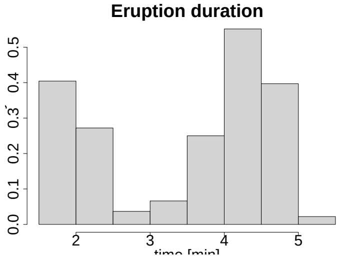
7. Asymptotic properties analyzed using Taylor expansions.
   - suggest a way to choose *bandwidth*
   - the bias-variance trade-off made explicit
8. Working on a grid and using FFT is the key to computational feasibility.

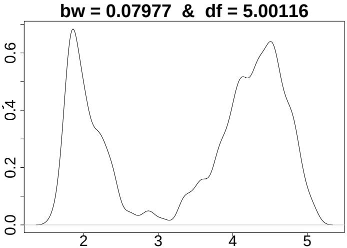
Degrees of Freedom (df)

- in linear models, the model df is the dimension of the space where the model is free to vary
  - equals number of regressors p if no linear dependence between regressors
  - generally $\mathrm{tr}(\mathbf{H})$, i.e. the trace of the hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top \mathbf{X})^\dagger \mathbf{X}^\top$
- more generally for linear smoothers: $\mathrm{df} := \mathrm{tr}(\mathbf{S})$

Example: think of a Gaussian mixture model for the Old Faithful eruption data:

$$f(x) = \tau \varphi_{\mu_1,\sigma_1^2}(x) + (1-\tau)\varphi_{\mu_2,\sigma_2^2}(x)$$





Assignment 2 [5%]

Slide "Bandwidth > Kernel" above gives a vague statement, that choosing bandwidth h is more important than choosing a proper kernel. See this for yourself using a small simulation study. Specifically:

- use Manual 09 to generate data from the Gaussian mixture f
- repeat the following 200 times:
  - generate $N = 100$ samples from the Gaussian mixture
  - perform density estimation, i.e. obtain $\widehat{f}$, for
    - Gaussian, Epanechnikov, and rectangular kernels
    - bandwidth values $h = 0.1, 0.15, 0.2, 0.25, \ldots, 0.9$
  - calculate the error measure $\|f - \widehat{f}\|_2$
- report your findings as a single (well commented) figure

Note: Please check again Section 5 of the Course Organization for submission instructions.