"cross-validation"

Data driven approach:

$$ISE = \int [f(u) - \hat{f}(u)]^2 du$$

$$= \underbrace{R(f)}_{\text{irrelevant for choosing } h} + \underbrace{R(\hat{f})}_{\text{can be computed in closed form}} - 2\int \hat{f}(u) f(u) du$$

this is not wrt the data sample we have.

$$-2\int \hat{f}(u) f(u) du = -2E[\hat{f}(u)], \; U \sim f$$

$\Rightarrow$ one idea is to estimate something proportional to ISE so that we could find h that would minimize.

Dropping the i^{th} case from $\{Y_1, \ldots, Y_n\}$ leads to an estimate $\hat{f}_{-i}(Y_i)$

histogram of $n-1$ points dropping Y_i

$\Rightarrow$ We can obtain

$$CV = R(\hat{f}) - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{-i}(Y_i)$$

evaluate this for many values of h, choose h that minimizes CV.

Why use ISE, not MISE? We still don't know f and trying to come up w/ something we can do. (practical).

Does this work?

Unfortunately often leads to undersmoothing (high variance).

2 Frequency Polygon

connect midpoint of histogram bins w/ straight lines!

The histogram is simple, useful and piecewise constant.

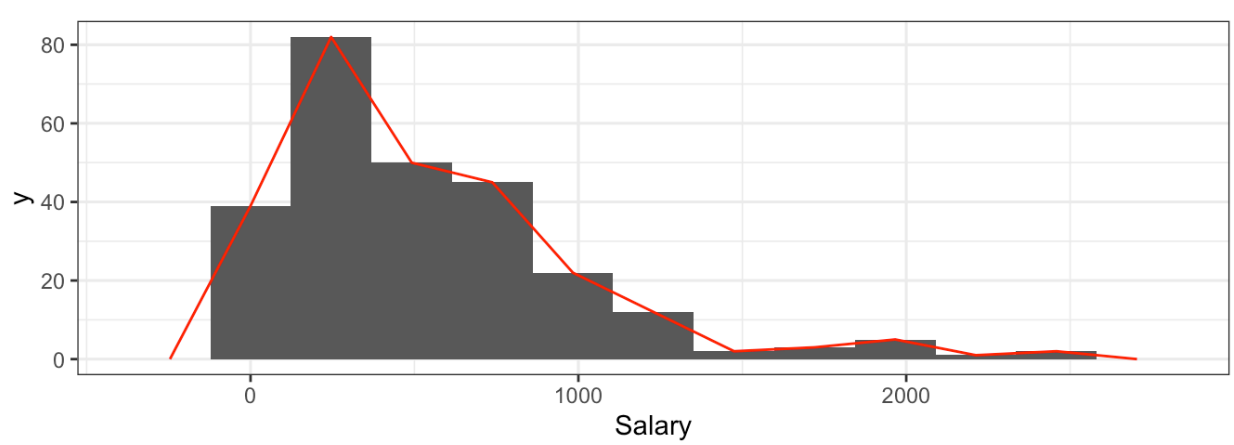
simple.

For continuous RVs, need something smoother.

```
library(ISLR)

# optimal h based on normal method
h_0 <- 3.491 * sd(Hitters$Salary, na.rm = TRUE) *
         sum(!is.na(Hitters$Salary))^(-1/3)

## original histogram with optimal h
ggplot(Hitters) +
  geom_histogram(aes(Salary), binwidth = h_0) -> p

## get values to build freq polygon
vals <- ggplot_build(p)$data[[1]]
poly_dat <- data.frame(x = c(vals$x[1] - h_0,
                             vals$x, vals$x[nrow(vals)] + h_0),
                       y = c(0, vals$y, 0))

## plot freq polygon
p + geom_line(aes(x, y), data = poly_dat, colour = "red")
```

optimal binwidth w/ plugin method.

pull out bin information.



Let $b_1, \ldots, b_{K+1}$ represent bin edges of bins with width h and $n_1, \ldots, n_K$ be the number of observations falling into the bins. Let $c_0, \ldots, c_{k+1}$ be the midpoints of the bin interval.

$$c_j = \frac{(b_j + b_{j+1})}{2} \quad j = 1, \ldots, K$$

$$c_0 = b_1 - \frac{h}{2}, \quad c_{K+1} = b_{K+1} + \frac{h}{2}$$

The frequency polygon is defined as

$$\hat{f}(x) = \frac{1}{nh^2}\left[n_j c_{j+1} - n_{j+1} c_j + (n_{j+1} - n_j)x\right] \text{ for } x \in [c_j, c_{j+1}]$$

If f'' is absolutely continuous and $R(f), R(f'), R(f''), R(f''')$ all finite,

$$\text{MISE} = \underbrace{\frac{2}{3nh}}_{\text{var}} + \underbrace{\frac{49 h^4 R(f'')}{2880}}_{\text{bias}^2} + O(n^{-1}) + O(h^6)$$

as $h \downarrow$, this increases

as $h \downarrow$, this decreases.

additional smoothness assumed (compared to Lipschitz for histogram).

compare the MISE for histogram: $\frac{1}{nh} + \frac{h^2 R(f')}{12} + O(n^{-1}) + O(h^3)$.

AMISE $\frac{2}{3nh} + \frac{49h^4 R(f'')}{2880}$

minimization of AMISE results $h_0 = 2\left[\frac{15}{49 R(f'')}\right]^{1/5} n^{-1/5}$

results in minimal AMISE $\text{AMISE}_0 = \frac{5}{12}\left[\frac{49 R(f'')}{15}\right]^{1/5} n^{-4/5}$

improved convergence rate (vs. $n^{-2/3}$ of histogram).

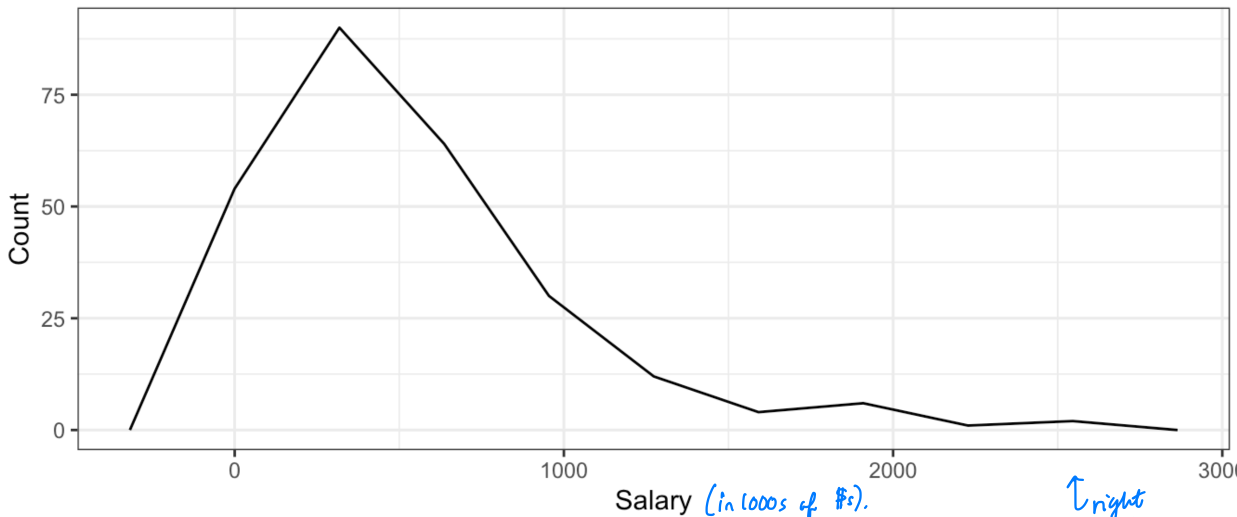
Optimal binwidth for freq. polygon will be asymptotically larger than histogram.

Gaussian rule for binwidth

Again, we do not know $R(f^{(4)}) \Rightarrow$ assume f = Gaussian and plug-in:

$$h_0 = 2.15\, \sigma\, n^{-1/5}$$

mode of salary ≈ 400k



(in 1000s of $s).

↑ right skew.

Major league baseball salaries from 1986 and 1987.

Let's go back to local error for a histogram for a moment:

$$MSE(\hat{f}(x)) = Var(\hat{f}(x)) + Bias(\hat{f}(x))^2$$

$$= \underbrace{\frac{f(x)}{nh}} + \underbrace{\frac{f'(x)^2}{4}\left[h - 2(x - b_j)\right]^2} + O(n^{-1}) + o(h^3).$$

If we want to minimize the 1st term (variance), binwidth should be larger in regions w/ high density.

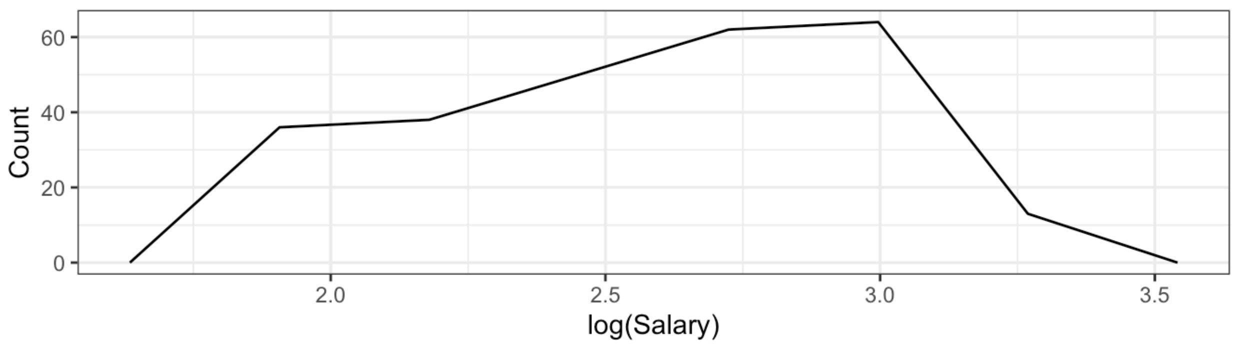
binwidth should be inversely related to $|f'(x)|$ to minimize 2nd term (bias).

$\Rightarrow$ a histogram w/ locally varying bin width could be more accurate to fixed width.

In practice, a simple way to construct locally varying binwidth histograms is by transforming the data to a different scale and then smoothing the transformed data. The final estimate is formed by simply transforming the constructed bin edges $\{b_j\}$ back to the original scale. how about $\hat{f}(x)$? No, just bin locations.

Again salary information for MLB 1986 + 1987 (1000's of dollars).

Fixed bin



Fixed bin on log 10 scale.

more detailed information here where we have more data

Varying bin width after back-transforming the bins.

3 Kernel Density Estimation

Recall the definition of a density function

$$f(y) \equiv \frac{d}{dy}F(y) \equiv \lim_{h\to 0}\frac{F(y+h)-F(y-h)}{2h} = \underbrace{\lim_{h\to 0}\frac{F(y+h)-F(y)}{h}}_{\text{took this and approximate w/ small fixed } h \text{ using the ecdf.}},$$

where $F(x)$ is the cdf of the random variable Y.

$$\hat{f}(x) = \frac{\hat{F}_n(y+h) - \hat{F}_n(y)}{h} \quad \text{histogram.}$$

What if instead, we replace $\underbrace{F(x+h) - F(x-h)}_{\text{with ecdf}}$?

$$\Rightarrow \hat{f}(x) = \frac{\hat{F}_n(x+h) - \hat{F}_n(x-h)}{2h} = \frac{\#\{x_i \in (x-h, x+h]\}}{2nh}$$

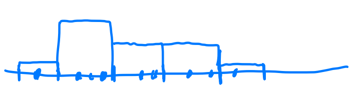
$$= \frac{\sum_{i=1}^{n} \mathbb{I}(x_i \in (x-h, x+h])}{2nh}$$

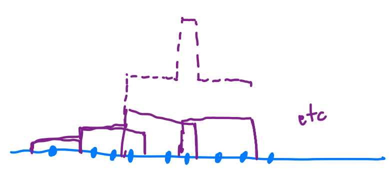
$$= \frac{1}{nh}\sum_{i=1}^{n} \frac{1}{2}\mathbb{I}(x-h < x_i \le x+h)$$

$$= \frac{1}{nh}\sum_{i=1}^{n} \frac{1}{2}\mathbb{I}\left(-1 < \frac{x_i - x}{h} \le 1\right)$$

$$= \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x_i - x}{h}\right) \text{ where } K \text{ is a Uniform density on } (-1,1].$$

"kernel" function





This looks almost like a histogram but has overlapping bins.

still not continuous! (because Uniform density is not cts!)

$\Rightarrow$ another Kernel may (will) lead to an even smoother estimate!

A Kernel function assign weights to the contribution given by each x_i to $\hat{f}(x)$ depending on proximity to x.

This will weight all points within h of x equally. A univariate *kernel density estimator* will allow a more flexible weighting scheme.

$\Rightarrow \frac{x_i - x}{h} = \frac{x - x_i}{h}$ in the kernel.

Typically, kernel functions are positive everywhere and symmetric about zero.

Examples:

Standard Normal

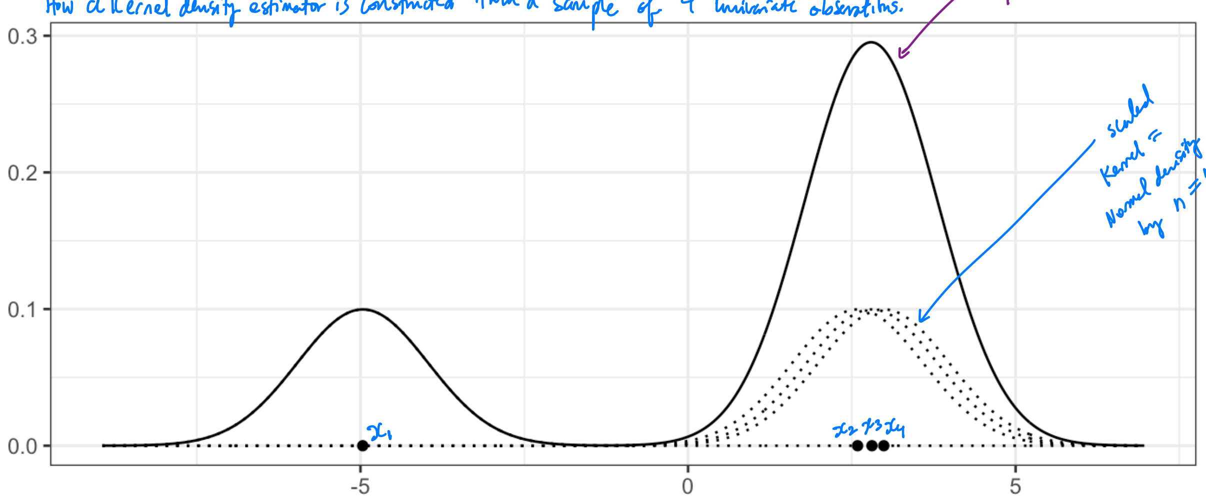
Student's t

(others also exist).

Additionally

Constraining K so that $\int z^2 K(z)\,dz = 1$ allows h to play role of scale parameter (not required).

How a Kernel density estimator is constructed from a sample of 4 univariate observations.



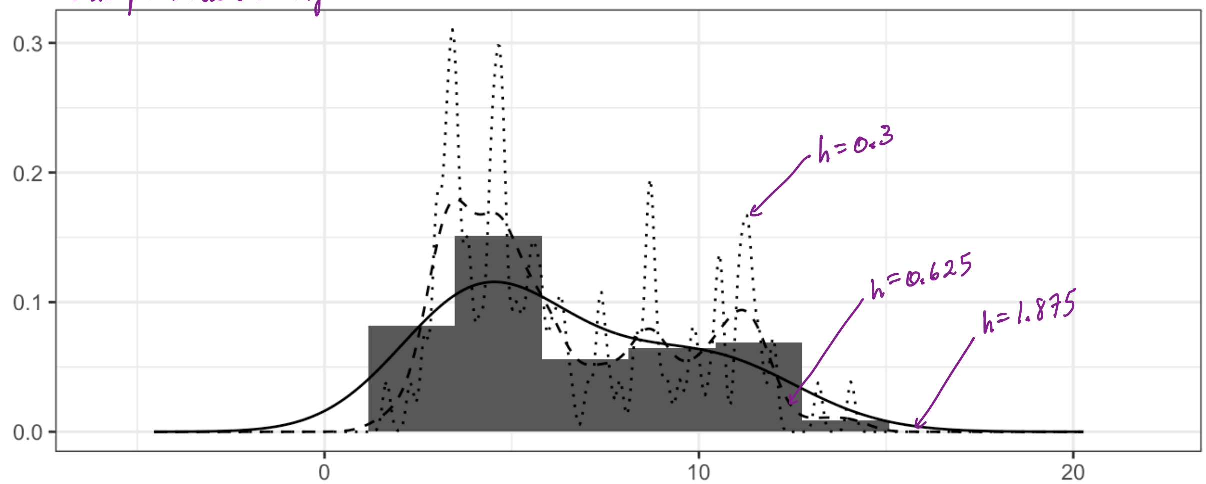
3.1 Choice of Bandwidth

The bandwidth parameter controls the smoothness of the density estimate. for a given Kernel.

↳ bandwidth determines tradeoff btw/ bias and variance.

The tradeoff that results from choosing the bandwidth + kernel can be quantified through a measure of accuracy of $\hat{f}$, such as MISE.

Example: bimodal density



data: equally weighted mixture of $N(4,1)$ and $N(9,4)$.

For h, see oversmoothing (lose 2^{nd} mode), For small h, undersmoothing, many false modes.