DATA SMOOTHING

GENERAL IDEAS:

- The goals of many of these methods is to reconstruct the p.d.f from a set of data points
- Can help to visualize the data (scatter plot → curve)
- Helps to remove outliers from the data

DEFINITIONS:

Bandwidth: a value that determines the width of the "neighborhood" used to estimate the local behavior for things like KDEs

Oversmooth: the method fails to capture the variability in the data

Undersmooth: the method contains too much "noise" and is bad at making a generalization

Kernel Function: the choice of function that determines the shape of the smoothing estimations

METHODS:

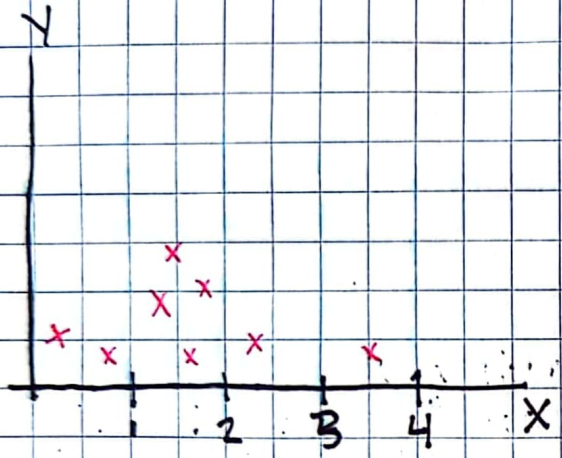
Histograms

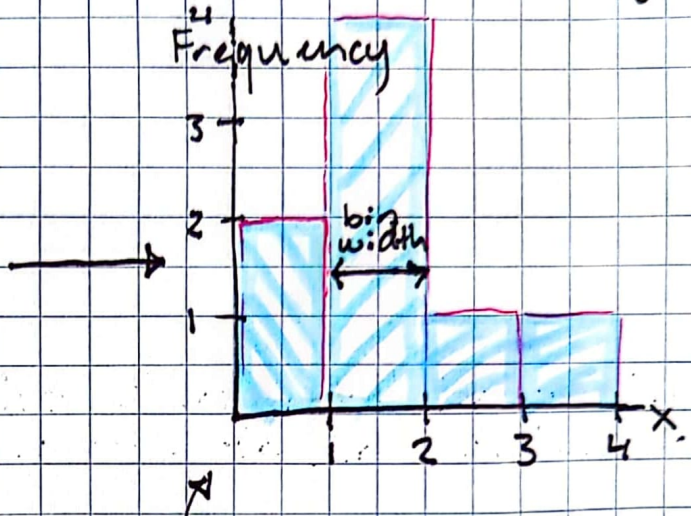
$$\hat{f}_{hist}(x) = \frac{N_m(x)}{n\,h(x)}$$

$N_m(x)$: # of points within the m-th bin

$h(x)$: bin width

- Histograms are good for visualizing the data
- You can't really do quantitative analysis w/ a histogram





One challenge is finding a good bin width. The bin width can change how the data appears. One option is the "Scott" bin width:

$$h_{scott} = \frac{3.5\ s.d.}{n^{1/3}} \quad \text{or} \quad \frac{2\,IQR}{n^{1/3}}$$

This is just the x component of the data, but it can be done for both dimensions

s.d = standard deviation; IQR = interquartile range

Kernel Density Estimators (KDE)

- KDEs are used to visualize the data, it estimates the pdf
- Makes discrete data into a continuous function
- To use this you need to specify the KDE function & bandwidth

Formulas:

$$\hat{f}_{Kern}(x, h) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)$$

h - bandwidth
X_i - variables / vectors
K - Kernel function
$\int K(x)\,dx = 1$

Usually fit $\hat{f}_{Kern}$ by minimizing the mean integrated square error (MISE):

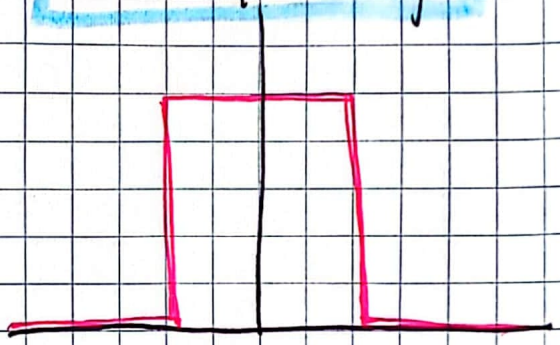
$$MISE(\hat{f}_{Kern}) = E\int [\hat{f}_{Kern}(x) - f(x)]^2\,dx$$
$$= \int Bias^2[\hat{f}_{Kern}(x)]\,dx + \int Var[\hat{f}_{Kern}(x)]\,dx$$

Some choices for the Kernel function:

Gaussian

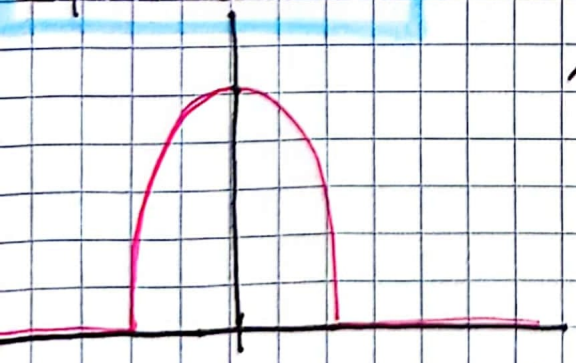
$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

Uniform/Rectangle



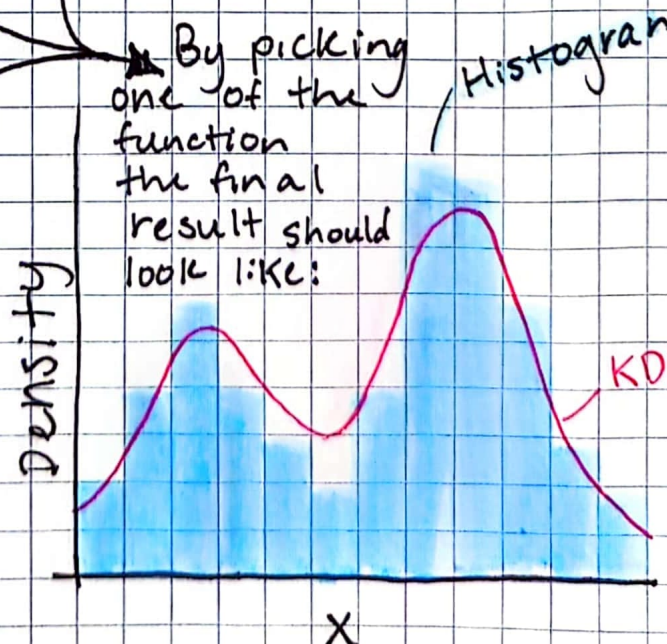
$$K(x) = \frac{1}{2} \mathbb{I}(-1 \le x \le 1)$$

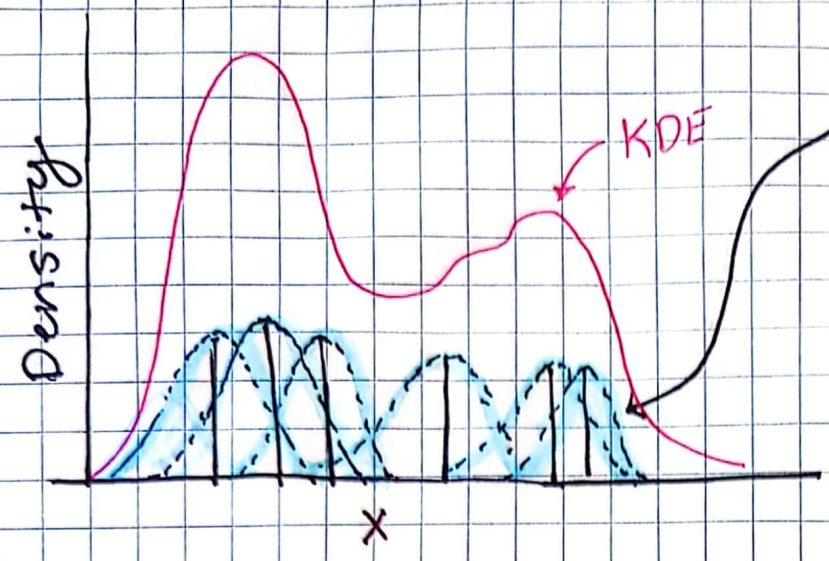
Epanechnikov



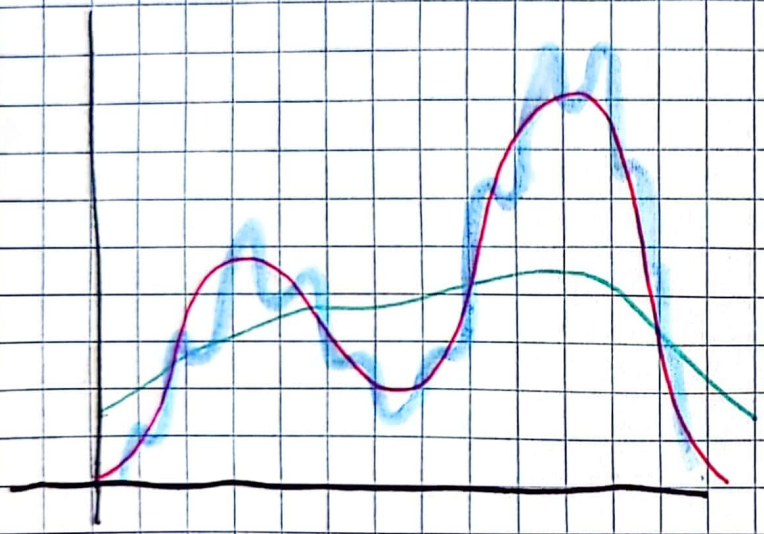
$$K(x) = \frac{3}{4} \max\{1 - x^2, 0\}$$

By picking one of the function the final result should look like:





→ A bunch of the kernel functions add together to create the KDE



- ideal KDE, good bin choice

- too wiggly, bin size too small, undersmoothed

- no bumps, bin size too large, oversmoothed

Picking The Bandwidth:

the goal is to minimized MISE, but this can be hard to do, so it often has to be done numerically.

Cross-validation: this is one option for minimizing the error.

↳ the data set is split and the data used to make the KDE is not used in the evaluation.

n data sets w/ $n-1$ points, KDE is calculated for each data set.

$$CV(h) = \frac{1}{n}\sum_{i=1} \ln\left[\hat{f}_{-i,kern}(x_i)\right]$$

→ optimal bandwidth h maximizes this.

What if you can't find a bin width that shows features well?

Adaptive Kernel Estimator may work.

↳ this changes the bin width at different locations denser regions of data will have a smaller bin width.

a couple options for this:

→ $h(x) = \frac{h}{\sqrt{\hat{f}(x)}}$ where $\hat{f}(x)$ is the estimated p.d.f

→ $\lambda_i = \left[\hat{f}(x_i)/g\right]^{-\alpha}$; $\ln(g) = n^{-1}\sum_{i=1}^{n} \ln \hat{f}(x_i)$

λ_i : bandwidth factor, α : sensitivity parameter $0 \leq \alpha \leq 1$

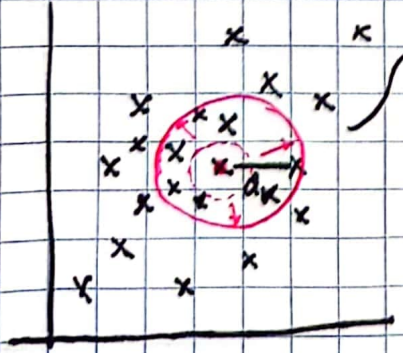
g : geometric mean of pilot estimator

$$\hat{f}_{adp,kern}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h_{opt}\lambda_i} K\left[\frac{x - X_i}{h_{opt}\lambda_i}\right]$$

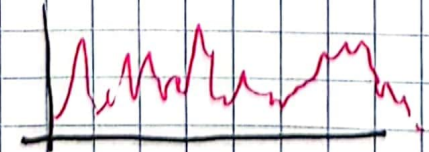
K-th nearest neighbors

Used to estimate the density at each data point.

$\hat{f}_{Knn}(x) = \frac{K/2n}{d_K(x)}$; K - which neighbor # you're going to
$d_K(x)$ - distance to the Kth neighbor



can think of it like a circle that goes outward until it contains K neighbors

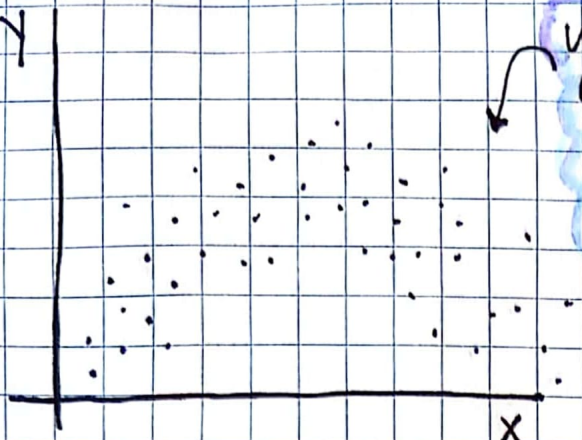


Low K value

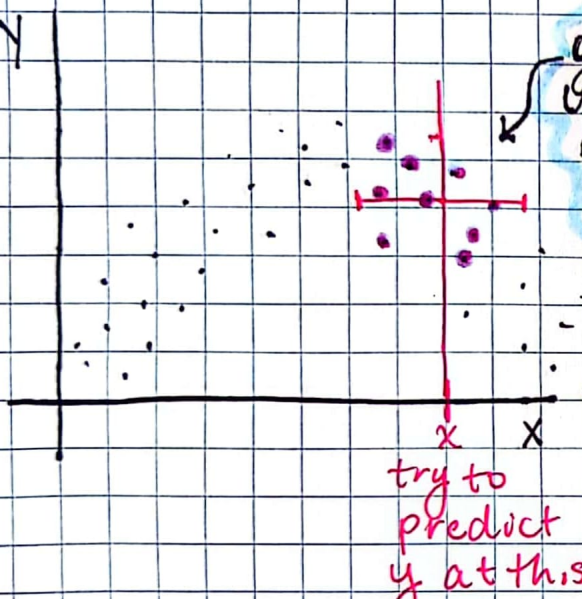
high K value

↳ These are not p.d.fs

Nadaraya-Watson estimator:



we have some sort of data set where we want to predict the "y" from the "x"



goal is to use nearby points to predict the value

try to predict y at this x

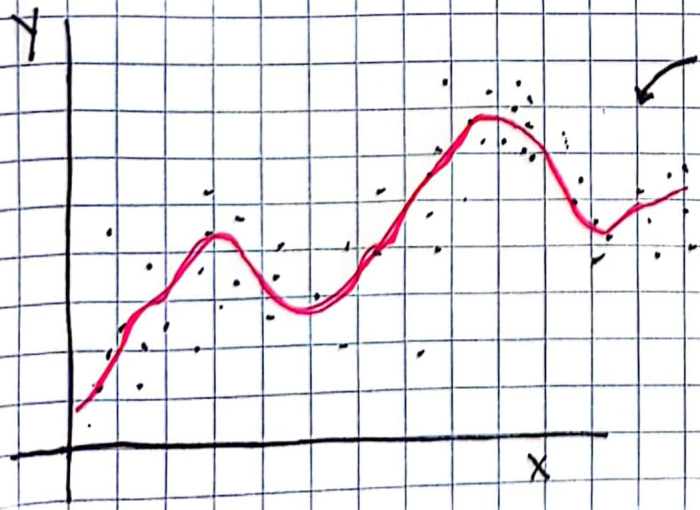
NW estimator:

$$\hat{r}_{NW}(x) = \frac{\sum_{i=1}^{n} Y_i K\left(\frac{x - X_i}{h_i}\right)}{\sum_{i=1}^{n} K\left(\frac{x - X_i}{h_i}\right)}$$

it uses the kernal function to weight the points. Nearby points have a higher weight

Spline smoothing:

goal is to use a predictor variable to estimate the response variable.



A spline may end up looking something like this, but there are many methods & parameters that can alter the fit

A spline is a piecewise function that pieces polynomials together → cubic splines use degree 3 polynomials. the splines will be continuous

↳ with enough polynomials, you could interpolate between data points, but this is not the goal
→ So there's a "cost" to overfitting

polynomials functions $\hat{f}(x)$ are picked to minimize:

$$\sum_{i=1}^{n} [Y_i - \hat{f}(x_i)]^2 + \lambda \int f''(x)^2 \, dx$$

λ is smoothing paramater, at $\lambda = 0$ the spline interpolates
at $\lambda \to \infty$ the spline look like least squares.