Rule of Thumb and Cross Validation Methods for Kernel Density Estimation

Huong Li Nguyen and Khanh Truong

November 18, 2018

1 Introduction

1.1 Motivation

Density estimation is a popular statistical technique for estimating an underlying probability density function $f(x)$ from observed data. Non-parametric approaches are useful, as they make less rigid assumptions about the underlying distribution.

The histogram estimator is the simplest non-parametric density estimator. However, several drawbacks such as "being unstable" and "non-smooth" motivated the use of kernel density estimators, which are smoother and converge to the true density at a faster rate. It is widely known that the performance of these estimators depend crucially on the chosen smoothing parameter, referred to as the bandwidth. In case of undersmoothing, the resulting estimated density is too rough and contains spurious features that are artifacts of the sampling process. In case of oversmoothing, important features of the underlying structure might be smoothed away (Jones et al., 1999) [6].

This project focuses on reviewing the rule of thumb and some of the most common cross-validation methods (least squares cross-validation and likelihood cross-validation) in the context of univariate kernel density estimation using fixed and adaptive bandwidths. In general, cross-validation provides a method for estimating generalization errors. The idea is to split the observed data available into a cross-validation training set and a cross-validation test set. The first is used to train the model, while the accuracy on the test set provides a generalization error estimate.

A short theoretical review of kernel density estimators is given in Section 2, which motivates the use of the bandwidth selection methods presented in Section 3 in the context of fixed bandwidth and Section 4 in the context of adaptive bandwidth. A comparison between the different methods will be illustrated by applying them on a real data set using R packages *KernSmooth*, *kedd* and *np*.

1.2 Data

The chosen data set is provided by the World Bank [13] and contains the time required to get a permanent electricity connection in days for the years of 2009-2017 of 185 countries. While electricity is taken for granted in developed countries, around 1.2 billion people worldwide did not have access to electricity in 2016. That's nearly 17% of the world's population. By estimating the density for the year 2017, we want to get a clearer picture of the current existing disparity.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
10.00	55.00	78.00	92.12	110	482.00

Table 1: Summary Statistics

Looking at the summary statistics in Table 1, we can already see that there are significant discrepancies. The United Arab Emirates is the country where to obtain electricity with the minimum value of 10 days, while in Liberia it takes 482 days. Interestingly, it takes 71 days in France and 28 days in Germany.

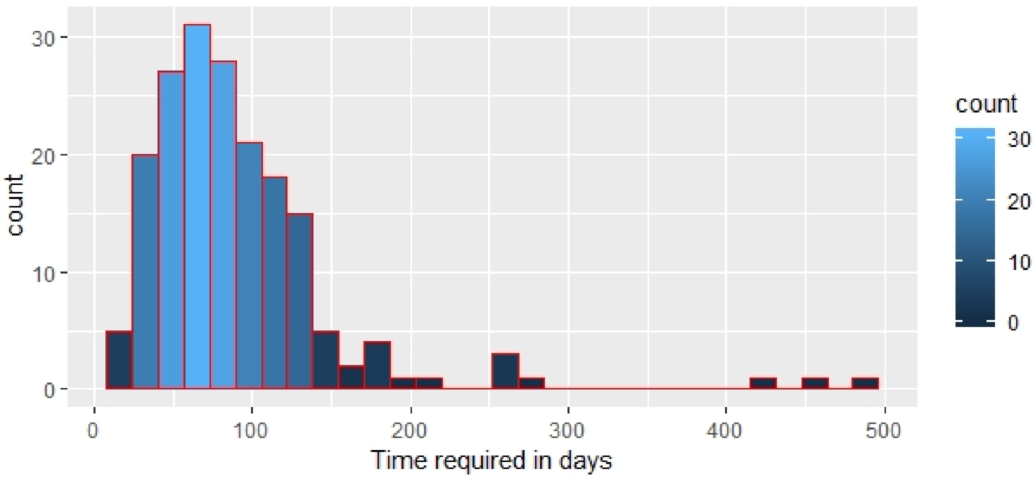

Figure 1: Histogram of Days required to obtain Electricity

The histogram in Figure 1 shows that most observation points center around the median. The distribution looks quite symmetric in the area of days below 110. After 110 days, there are only few observations points indicating that a greater bandwidth might be more suitable in this area, while a smaller bandwidth might be more adequate in the area of days below 110.

2 Kernel Density Estimation

A short review of the most relevant concepts of kernel density estimation is presented in this section. Let $X_1, X_2, ...X_n$ be independent realizations of size n from a random variable with density f. By Rosenblatt (1956) [9] and Parzen (1962) [8], the kernel density estimator of f at point x is given by:

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - X_i}{h}) \tag{1}$$

where K is the kernel and h the bandwidth.The kernel refers to any smooth function K satisfying the following conditions:

- $\int K(x)dx = 1$
- $\int xK(x)dx = 0$
- $\int x^2 K(x)dx = \mu_2(K) > 0$

The two generally accepted methods to assess the performance of $\hat{f}_h$ are the integrated squared error (ISE):

$$ISE(h) = \int \hat{f}_h(x) - f(x))^2 dx = \int \hat{f}_h^{\,2}(x)dx \tag{2}$$

or alternatively, the mean integrated squared error (MISE):

$$MISE(\hat{f}_h(x)) = \int Bias(\hat{f}_h(x))^2 dx + \int Var\hat{f}_h(x))dx \tag{3}$$

where B$ias(\hat{f}_h(x)) = \frac{h^2}{2}\mu_2(K)f\,''(x) + o(h^2)$, V$ar\hat{f}_h(x) = \frac{h^2}{2}R(K)f(x) + o(\frac{1}{nh})$ and R$(K) = \int K^2(y)dy$. Under standard technical assumptions (Silverman, 1986 [12]), MISE can be asymptotically approximated by the asymptotic mean integrated squared error (AMISE).

$$AMISE(\hat{f}_h(x)) = \frac{1}{nh}R(K) + \frac{h^4}{4}\mu_2(K)^2 R(f\,'') \tag{4}$$

where R$(f\,'') = \int (f\,'')^2 dy$ and $\mu_2(K) = \int x^2 K(x)dx$. We already know that the choice of the kernel K is not crucial, but the choice of the bandwidth h is. The effect of the smoothing parameter h becomes evident in this equation. The integrated variance is large if h is chosen too small, while the integrated squared bias is large when h is chosen too large. The $AMISE(h)$ is minimized for:

$$h_{AMISE} = \frac{1}{n^{\frac{1}{5}}}\left(\frac{R(K)}{\mu_2(K)^2 R(f\,'')}\right)^{\frac{1}{5}} \tag{5}$$

However, in practice we do not know the true density f, disabling us to use the formula to obtain the global optimal bandwidth. Bandwidth selection methods are therefore used to solve this issue. One distinguishes mainly between two classes of bandwidth selection methods: the cross-validation methods (CV) and the plug-in methods.

Plug-in methods use the MISE as an objective function, while cross-validation methods usually use the ISE. Asymptotically, plug-in estimators always have a smaller variance compared to cross-validation methods (Hall and Marron, 1987 [5]). However, an advantage of cross-validation methods is that they allow us to choose the bandwidth without making assumptions about the smoothness to which the unknown density belongs to. Plug-in methods, on the other hand, are usually less volatile and have a faster convergence rate compared to cross-validation methods, but they heavily depend on prior information.

Silverman's (1986) [12] rule of thumb is the most popular method among the plug-in methods and will be presented in the next section, along with some of the most common cross-validation methods (likelihood cross-validation and least squares cross-validation).

3 Fixed Bandwidth Methods

3.1 The Rules of Thumb (RoT)

This method was popularized for kernel density estimators by Silverman (1986) [12] and involves the replacement of the unknown part $R(f\,'')$ of h_{AMISE} by an estimated value based on a parametric family. Assuming that the underlying distribution of the bandwidth is normal and using a Gaussian kernel, Silverman showed that:

$$\hat{h_{rot}} \approx 1.06\hat{\sigma}n^{\frac{-1}{5}} \tag{6}$$

where $\hat{\sigma}$ denotes the standard deviation of X. A practical problem here is that the bandwidth is sensitive to outliers. A single outlier may cause the estimator of σ to be large, resulting in a bandwidth chosen too large. A more robust rule of thumb is given by:

$$\hat{h_{rot}} \approx 0.79 I\hat{Q}R n^{\frac{-1}{5}} \tag{7}$$

where $I\hat{Q}R$ denotes the interquartile range of X. Combining the first and the robust rule of thumb, a "Better-rule-of-thumb" is given by:

$$\hat{h_{rot}} = 1.06 min(\hat{\sigma}, \frac{I\hat{Q}R}{1.34}) n^{\frac{-1}{5}} \tag{8}$$

However, Silverman's rule of thumb performs well only if the true underlying density is unimodal, fairly symmetric and does not have fat tails. In other words, if the assumptions are faily met and the underlying distribution is not far from a Gaussian distribution. Otherwise, this method can yield widely inaccurate estimates. It may oversmooth for bimodal or multimodal data and is therefore not applicable within these frameworks.

3.2 Least-Squares Cross-Validation (LSCV)

Rudemo (1982) [10] and Bowman (1984) [1] proposed the least-squares (unbiased) cross-validation method (LSCV) in the context of kernel density estimation. Due to its efficiency and simplicity it is widely used in practice. The idea is to minimize the integrated squared error ($ISE(h)$) between the estimated distribution function $\hat{f}_h$ and the true distribution f:

$$ISE(h) = \int (\hat{f}_h(x) - f(x))^2 dx = \int \hat{f_h}^2(x)dx - 2\int \hat{f}_h(x) f(x) dx + \int f^2(x) dx \tag{9}$$

Since the last term is independent from h, the minimization of the $ISE(h)$ is equivalent to the minimization of $L(h)$:

$$L(h) = ISE(h) - \int f^2(x) dx = \int \hat{f_h}^2(x)dx - 2\int \hat{f_h(x)} f(x) dx \tag{10}$$

Here $f(x)$ is unknown, so $L(h)$ is estimated by $CV_{LS}(h)$:

$$CV_{LS}(h) = \int \hat{f_h}^2(x)dx - 2\frac{1}{n}\sum_{i=1}^{n} \hat{f}_{h,-i}(X_i) \tag{11}$$

where n is the number of observations and $\hat{f}_{h,-i}$ is the density estimate without the data point X_i. Hence, minimizing $L(h)$ or $ISE(h)$ will be (asymptotically) equivalent to minimizing $CV_{LS}(h)$. The solution for the minimization problem is:

$$h_{LSCV} = arg \min_h CV_{LS}(h) \tag{12}$$

A drawback of the LSCV includes high variability e.g. estimated bandwidths may have large variance when using different samples from the same distribution. Furthermore, it might capture the noise of the data sample, instead of the general desired trend, resulting in multiple local minimas (Sain et al., 1994 [11]). Also its tendency to undersmooth data is criticized (Sain et al., 1994 [11]), emphasizing the importance of investigating alternative bandwidth selection methods.

3.3 Likelihood Cross-Validation (MLCV)

One alternative to the LSCV is the likelihood cross-validation (MLCV), which was proposed by Habbema et al. (1974) [4] and Duin (1976) [3]. The concept is similar to the LSCV, the only difference is the measurement of the score function. While LSCV is based on minimizing the integrated sequared error, the MLCV minimizes the Kullback-Leibler distance between the estimated and the true underlying distribution.

$$CV_{ML}(h) = \frac{1}{n}\sum_{i=1}^{n} log\hat{f}_{h,-i}(X_i) \tag{13}$$

The optimal h is then obtained by minimizing the score function:

$$h_{MLCV} = arg\min_{h} CV_{ML}(h) \tag{14}$$

Drawbacks of the likelihood cross-validation for kernel density estimation include sensitivity to extreme observations and heavy-tailed distributions (Hall, 1987a).

3.4 Comparison

In this section the three previously presented methods will be compared through an illustration on real data. We recall that the data represents the required days to obtain electricity for 185 countries worldwide in 2017. Using the *kedd* and the *KernSmooth* function in R, we obtain the following optimal bandwidths:

Method	Rule of Thumb	Least-Squares CV	Likelihood CV
h	13.754	15.380	4.995

Table 2: Optimal Bandwidth chosen with Gaussian kernel on 2017 data

For all three methods, we used the Gaussian kernel. While the optimal bandwidth selected by MLCV did not change significantly in magnitude when choosing the Epanechnikov kernel, it did change significantly in magnitude in case of the LSCV. In Table 2 we can see that the estimates of the RoT and the LSCV are very similar in magnitude, while the estimate using the MLCV differs widely from the other estimates.

In Figure 2 below the estimated densities are plotted. Given that the optimal bandwidth by LSCV and RoT are very close in magnitude, the estimated densities by these bandwidth selection methods are very close to each other as well. However, the estimated density by the MLCV differs widely from the others. We can see that the bandwidth chosen by this method is too small leading to undersmoothing.

This might be explained by the fact, that the MLCV is very sensible to extreme values, which are contained in our data set in the area of more than 110 days. The drawback of potential undersmoothing of the LSCV as described in Section 3.2 cannot be confirmed in this data set. However, we can confirm the fact that the LSCV can be unstable for different samples by performing the same analysis on the same data, but for 2009, assuming that the underlying distribution is the same. Results are summarized below.

While the optimal h chosen by RoT and MLCV do not change significantly in magnitude from Table 3 to Table 2, the chosen bandwidth by the LSCV is less than half the size of the chosen bandwidth for the 2017 data set.

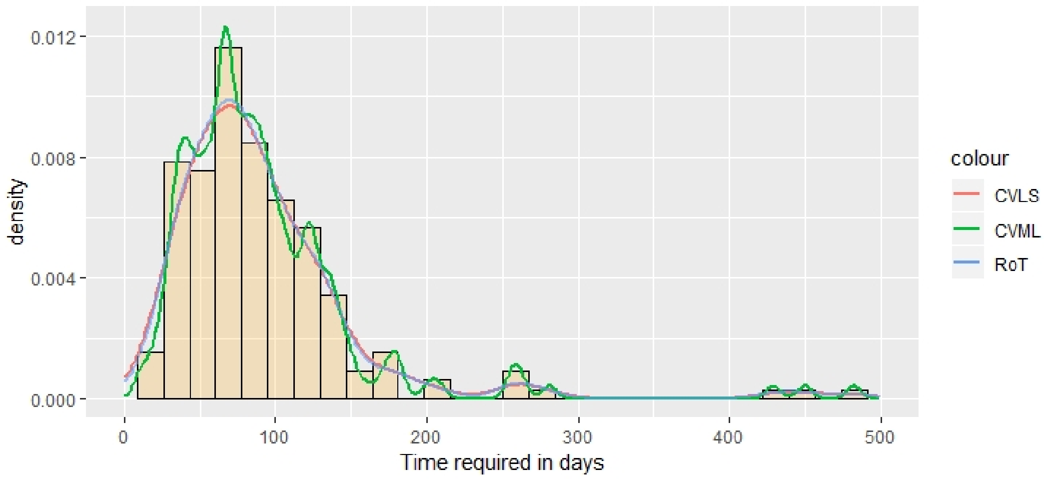

Figure 2: Comparison of the estimated densities for 2017 data

Method	Rule of Thumb	Least-Squares CV	Likelihood CV
h	15.745	7.487	4.993

Table 3: Optimal Bandwidth chosen with Gaussian kernel on 2009 data

In Figure 3 we can see, that the estimated density by the MLCV and the LSCV now both undersmooth the data. The chosen bandwidth by these two methods are closer in magnitude now. The RoT and the MLCV seem to be more robust methods considering this data set as the optimal bandwidth does not change much in magnitude from the 2017 to the 2009 data set.

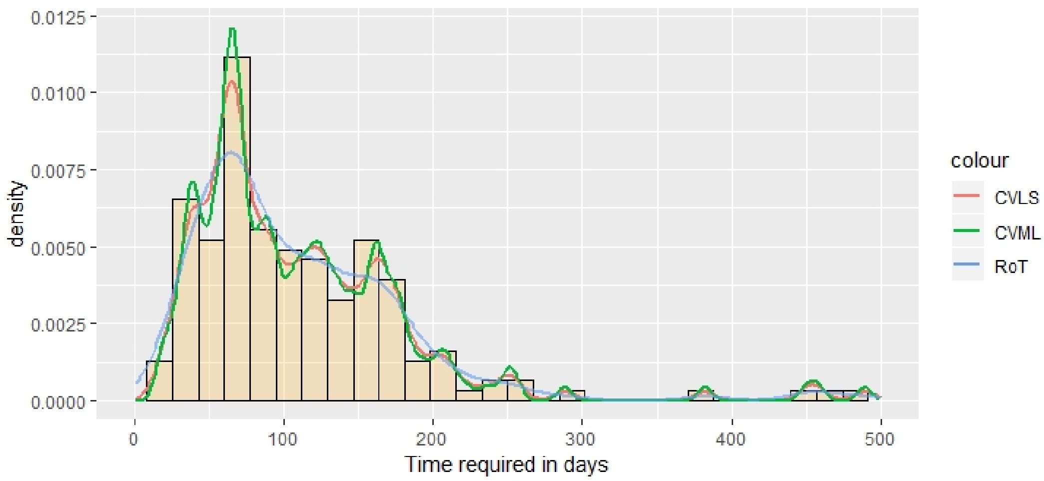

Figure 3: Comparison of the estimated densities for the 2009 data set

Overall, one can see that the performance of the different methods depend significantly on the data set. The MLCV performed similar on the 2009 and 2017 data set, but tends to undersmooth the data in both cases. The LSCV tends to undersmooth the 2009 data, which was not the case for the 2017 data. Overall, the chosen bandwidth by LSCV differed widely for the two data sets, confirming the instability of this method depending on the data set. The Rule of Thumb seems to perform best in stability and smoothness. This might be explained by the fact that the data fits the prior assumption of the Rule of Thumb quite well (unimodal and fairly symmetric data).

4 Adaptive Bandwidth

4.1 Adaptive Nearest-Neighbour Bandwidth

So far, the methods have been compared in the context of a fixed bandwidth h. However, one h which is suitable in some areas of high density may not necessarily be appropriate in low-density regions. In Section 1.2, we have already seen that this might be the case with our data set as well. While the density is high in the area of days below 110, it is quite low in the area after 110 days. Therefore, instead of using fixed bandwidths, Breiman et al. (1977) [2] introduced the adaptive bandwidth h_i, which varies along the range of data points (called adaptive nearest-neighbour bandwidths or point-wise estimator). The idea is that we enlarge the bandwidth at X_i if there are only few observations around and contrariwise. By doing so, one can reduce the variance and increase the bias in areas with only few observations.

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h(X_i)}K(\frac{x-X_i}{h(X_i)}) \tag{15}$$

The bandwidth $h(X_i)$ is calculated by the distance between X_i and the k-th nearest data point to X_i.

Procedure:

1. Given sample with N observations. For each observation X_i, rank every other points in the sample by Euclidian (or Minkowski or Manhattan) distance to X_i. We get $X_{(1)}, X_{(2)}, ..., X_{(N-1)}$.
2. Fix an integer k, $d_k(X_i)$ denotes the distance between X_i and $X_{(k)}$.
3. Set the bandwidth $h(X_i)$ at X_i equal to $d_k(X_i)$: $h(X_i) = d_k(X_i)$
4. Calculate the kernel density estimator $\hat{f}(x)$ as given by the above formula 15).

The smaller the k, the smaller the neighbourhood around X_i, indicating an increase in the variance, while simultaneously reducing the bias. On the other hand, selecting a large k will result in a smoother estimator (lower variance) but potentially higher bias. This phenomenon is exactly similar to the choice of h. Thus, choosing k depends on the sample. An illustration how a different choice of k affects the density is illustrated in Figure 4.

4.2 Generalized Nearest-Neighbour Bandwidths

Another adaptive bandwidth approach is that h varies along the support of X, instead of the range of sample realization (Loftsgaarden and Quesenberry, 1965 [7]). This method is sometimes called generalized nearest-neighbour bandwidth or balloon estimator. The kernel density estimator is given here by:

$$\hat{f}(x) = \frac{1}{nh(x)}\sum_{i=1}^{n}K(\frac{x-X_i}{h(x)}) \tag{16}$$

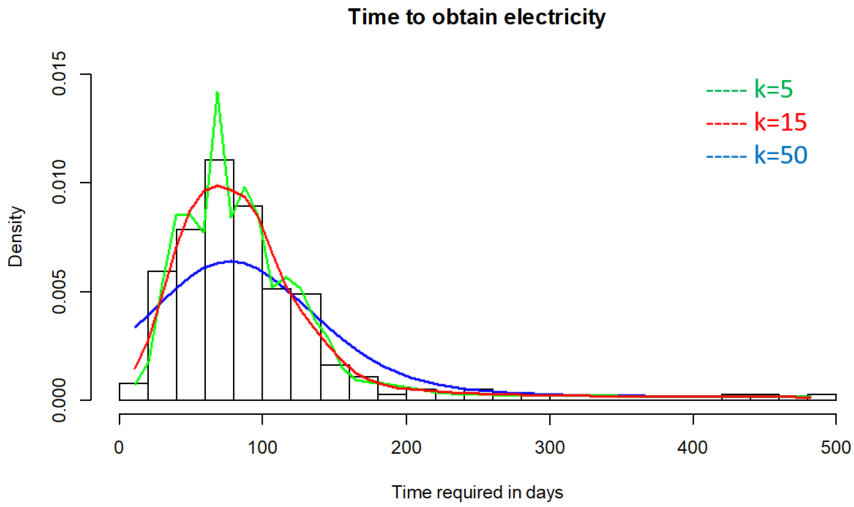

Figure 4: Adaptive kernel estimators with different values of k

Generalized nearest-neighbour bandwidths share similar features as adaptive nearest-neighbour bandwidths, which will smooth out local maximas in areas where the density is sparse. However, this method is less advantageous because it does not guarantee the integration of the estimator to equal 1 (Zambom and Dias, 2013 [14]).

4.3 Combination with Cross-Validation

We can apply least-squares cross-validation and likelihood cross-validation again to find the optimal k in the context of adaptive bandwidth. The k is then chosen from $1, ..., N-1$ such that it minimizes the $ISE(k)$. Depending on the kernel functions, the optimal k may vary from 11 to 18 in our sample data set (Table 4).

Method/Kernel	Epanechnikov	Gaussian	Uniform
Least Square CV	18	18	17
Maximum Likelihood CV	16	11	17

Table 4: Optimal k chosen in adaptive bandwidth (Data 2017)

We will again compare the results for the 2017 data set with the 2009 data set to verify whether the cross-validation methods used are sensitive to samples. It turns out that the optimal k of data in the 2009 data set is notably different compared to the one in 2017 (Table 5). Apart from uniform kernel, the optimal k using Epanechnikov and Gaussian kernels remarkably drop down which means the bandwidths are generally smaller than previously. This confirms our early statement that LSCV is sensitive to our data sample, regardless of employing fixed or adaptive bandwidth methods; and the same applies for MLCV but only in the context of adaptive bandwidth.

Method/Kernel	Epanechnikov	Gaussian	Uniform
Least Square CV	6	6	10
Maximum Likelihood CV	7	7	23

Table 5: Optimal k chosen in adaptive bandwidth (Data 2009)

A natural question is whether we should use fixed h or adaptive h? It depends (as always). As a rule, fixed h will generate a smaller bias but greater variance at the tail, where density is low. Whereas adaptive h tends to be smoother in the tails (Figure 5). On that account, we should consider to use adaptive bandwidth if we want to reduce the importance of extreme values in our sample (e.g. if we assume these extreme values are either outliers or can be discarded as data collecting errors). On the other hand, fixed bandwidth is a good approach if we assume extreme values are part of the true data generating process. Eventually, there is no absolutely best method for all types of problems. The performance of the different methods depend significantly on the data set chosen.

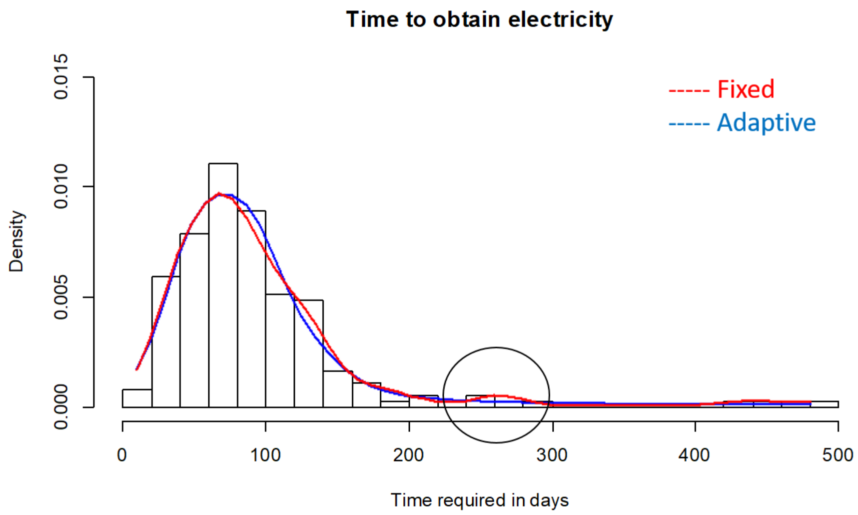

Figure 5: Compare fixed and adaptive methods

5 Coding in R

```
# ======================================================================#
#                       Non Parametric - Project
#                   Huong Li Nguyen | Khanh Truong
# ======================================================================#

# Import libraries -----------------------------------------------------
library(KernSmooth)
library(ggplot2)
library(tidyverse)
library(kedd)
library(np)


# Import and clean data  -----------------------------------------------
temp <- tempfile()
url<-
  'http://api.worldbank.org/v2/en/indicator/IC.ELC.TIME?downloadformat=csv'
download.file(url,temp, mode="wb")
unzip(temp, "API_IC.ELC.TIME_DS2_en_csv_v2_10187384.csv")
data <- read.csv("API_IC.ELC.TIME_DS2_en_csv_v2_10187384.csv", skip=4,
                 check.names = F)
data <- data[c('Country Name','Country Code', 2009:2017)] # Select columns

# Clean to get data only for countries, not continents
country <- data[
  (is.na(data$`2017`)==F) & ((data$`2017` - round(data$`2017`,1)) == 0)
  & (is.na(data$`2016`)==F) & ((data$`2016` - round(data$`2016`,1)) == 0)
  & (is.na(data$`2015`)==F) & ((data$`2015` - round(data$`2015`,1)) == 0),]

# Remove North America as it is a continent
country <- country[-(country$`Country Name`=="North America"),]


# Split data 2017 and 2009 ---------------------------------------------
# Data 2017
time_elec17 <- country[c('Country Name','Country Code',2017)] # 2017 data
time_elec17 <- time_elec17[complete.cases(time_elec17[,
                                        as.character(2017)]),] # Remove NA
names(time_elec17) <- c("COUNTRY_N", "COUNTRY_C", "X2017") # rename

# Data 2009
time_elec09 <- country[c('Country Name','Country Code',2009)] # 2009 data
time_elec09 <- time_elec09[complete.cases(time_elec09[,
                                        as.character(2009)]),] # Remove NA
names(time_elec09) <- c("COUNTRY_N", "COUNTRY_C", "X2009") # rename


# Histogram data 2017 --------------------------------------------------
summary(time_elec17$X2017)
ggplot(time_elec17, aes(x=X2017))+
  geom_histogram(col="red",
```

```r
                  aes(fill=..count..)) +
  xlab("Time required in days")


# Kernel esimators data 2017 --------------------------------------------
# Choose the optimal bandwidth
kernels <- eval(formals(h.mlcv.default)$kernel)

# Rule of thumb
h.rt <- dpik(time_elec17$X2017, scalest = "minim", kernel = "normal")
# Least Squares Cross Validation
h.LS <- h.ucv(time_elec17$X2017, deriv.order = 0, kernel = kernels[1])$h
# Likelihood cross validation
h.ML <- h.mlcv(time_elec17$X2017, kernel = kernels[1])$h

# Print results of the three methods
h.rt
h.ML
h.LS

# Extract (x,y) of binned kernel density estimate of density of the data
estRT <- bkde(time_elec17$X2017, bandwidth = h.rt)
estLS <- bkde(time_elec17$X2017, bandwidth=h.LS)
estML <- bkde(time_elec17$X2017, bandwidth=h.ML)

# Unlist and create a dataframe in order to use ggplot
est.LS <- data.frame(matrix(unlist(estLS), nrow = 401))
est.ML <- data.frame(matrix(unlist(estML), nrow = 401))
est.rt <- data.frame(matrix(unlist(estRT), nrow = 401))


# Plot data 2017 -------------------------------------------------------
ggplot(time_elec17, aes(x=X2017, y=..density..))+
  geom_histogram(col="black",
                  fill = "orange",
                  alpha = 0.2) +
  geom_line(data = est.LS, aes(x = X1, y = X2, colour = "CVLS"),
            size = 0.8) +
  geom_line(data = est.ML, aes(x = X1, y = X2, colour = "CVML"),
            size = 0.8) +
  geom_line(data = est.rt, aes(x = X1, y = X2, colour = "RoT"),
            size = 0.8, alpha = 0.6) +
  xlab("Time required in days")+
  xlim(0, 500)


# Histogram data 2009 ---------------------------------------------------
summary(time_elec09$X2009)
ggplot(time_elec09, aes(x=X2009))+
  geom_histogram(col="red",
                  aes(fill=..count..)) +
  xlab("Time required in days")
```

```r
# Kernel esimators data 2009 -------------------------------------------
# Choose the optimal bandwidth
kernels <- eval(formals(h.mlcv.default)$kernel)

h.rt <- dpik(time_elec09$X2009, scalest = "minim", kernel = "normal")
h.LS <- h.ucv(time_elec09$X2009, deriv.order = 0, kernel = kernels[1])$h
h.ML <- h.mlcv(time_elec09$X2009, kernel = kernels[1])$h

# Results of the three methods
h.rt
h.ML
h.LS

# Extract (x,y) of binned kernel density estimate of density of the data
estRT <- bkde(time_elec09$X2009, bandwidth = h.rt)
estLS <- bkde(time_elec09$X2009, bandwidth=h.LS)
estML <- bkde(time_elec09$X2009, bandwidth=h.ML)

# Unlist and create a dataframe in order to use ggplot
est.LS <- data.frame(matrix(unlist(estLS), nrow = 401))
est.ML <- data.frame(matrix(unlist(estML), nrow = 401))
est.rt <- data.frame(matrix(unlist(estRT), nrow = 401))


# Plot data 2009 -------------------------------------------------------
ggplot(time_elec09, aes(x=X2009, y=..density..))+
  geom_histogram(col="black",
                 fill = "orange",
                 alpha = 0.2) +
  geom_line(data = est.LS, aes(x = X1, y = X2, colour = "CVLS"),
            size = 0.8) +
  geom_line(data = est.ML, aes(x = X1, y = X2, colour = "CVML"),
            size = 0.8) +
  geom_line(data = est.rt, aes(x = X1, y = X2, colour = "RoT"),
            size = 0.8, alpha = 0.6) +
  xlab("Time required in days")+
  xlim(0, 500)

# Compare variance
var(time_elec09$X2009)
var(time_elec17$X2017)


# Adaptive bandwidth ---------------------------------------------------
adap_bw <- npudensbw(time_elec17$`X2017`,bwtype='adaptive_nn',
                     bwmethod='cv.ls',ckertype="gaussian") # bandwidth
adap <- npudens(bws=adap_bw) # estimator
plot(adap)
print(adap$bw) # k in k-th nearest neighbor

# replace with following options:
# bwmethod: cv.ml, cv.ls, normal-reference
# ckertype: gaussian, epanechnikov, uniform
# bwtype: fixed, generalized_nn, adaptive_nn
```

```
hist(time_elec17$`X2017`,probability = T,main="Time to obtain electricity",
     xlab="Time required in days",ylim=c(0,0.015),breaks = 20) # Histogram

# Set different values of k
adap_bw$bw <- 5 # Set k = 5
adap_5 <- npudens(bws=adap_bw)
plot(adap_5,ylim=c(0,0.015),xlab = NULL, ylab=NULL,col="green",lwd=2)

adap_bw$bw <- 15 # Set k = 15
adap_15 <- npudens(bws=adap_bw)
plot(adap_15,ylim=c(0,0.015),xlab = NULL, ylab=NULL,col="red",lwd=2)

adap_bw$bw <- 50 # Set k = 50
adap_50 <- npudens(bws=adap_bw)
plot(adap_50,ylim=c(0,0.015),xlab = NULL, ylab=NULL,col="blue",lwd=2)


# Compare fixed and adaptive -------------------------------------------
fix_bw <- npudensbw(time_elec17$`X2017`,  bwtype='fixed',
                    bwmethod='cv.ls',ckertype="gaussian")
fix <- npudens(bws=fix_bw)
plot(fix,ylim=c(0,0.015),xlab = NULL, ylab=NULL,col="red",lwd=2)
print(fix$bw) # value bandwidth h

adap_bw <- npudensbw(time_elec17$`X2017`, bwtype='adaptive_nn',
                     bwmethod='cv.ls',ckertype="gaussian")
adap <- npudens(bws=adap_bw)
plot(adap,ylim=c(0,0.015),xlab = NULL, ylab=NULL,col="blue",lwd=2)
print(adap$bw) # k in k-th nearest neighbor


# Compare adaptive bandwith between data 2017 and data 2009 -------------
adap_bw <- npudensbw(time_elec09$`X2009`,bwtype='adaptive_nn',
                     bwmethod='cv.ml',ckertype="gaussian")
adap <- npudens(bws=adap_bw)
print(adap$bw) # k in k-th nearest neighbor
# replace different options to get different estimators for data 2009
```

References

[1] Bowman, A. W. (1984). An Alternative Method of Cross-Validation for the Smoothing of Density Estimates. *Biometrika*, 71(2): 353-360.

[2] Breiman, L., Meisel, W. and Purcell, E. (1977). Variable kernel estimates of multivariate probability densities. *Technometrics*, 19: 135-144.

[3] Duin, R. P. W. (1976). On the Choice of Smoothing Parameters for Parzen Estimators of Probability Density Functions. *IEEE Transactions on Computers*, 25(11): 1175–1179.

[4] Habbema, J. D. F., Hermans, J. and Van den Broek, K. (1974). A stepwise discriminant analysis program using density estimation. *Compstat 1974: Proceedings in Computational Statistics*, 101–110.

[5] Hall, P. and Marron, J.S. (1987). Extent to which Least-Squares Cross-Validation Minimises Integrated Square Error in Nonparametric Density Estimation. *Probability Theory and Related Fields*, 74(4): 567-581.

[6] Jones, M.C, Marron, J.S. and Sheather, S.J. (1999). A Brief Survey of Bandwidth Selection for Density Estimation. *Journal of the American Statistical Association*, 91(433): 401-407.

[7] Loftsgaarden, D. O. and Quesenberry, C. P. (1965). A Nonparametric Estimate of a Multivariate Density Function. *The Annals of Mathematical Statistics*, 36(3): 1049-1051.

[8] Parzen, E. (1962). On Estimation of a Probability Density Funcion and the Mode. *The Annals of Mathematical Statistics*, 33(3): 1065-1067.

[9] Rosenblatt, M. (1956). Remarks on Some Nonparametric Estimates of a Density Function. *Annals of Mathematical Statistics*, 27(3): 832-837.

[10] Rudemo, M. (1982). Empirical Choice of Histograms and Kernel Density Estimators. *Scandinavian Journal of Statistics*, 9(2): 65-78.

[11] Sain, S. R., Baggerly, K. A. and Scott, D. W. (1994). Cross-Validation of Multivariate Densities. *Journal of the American Statistical Association*, 89(427): 807-817.

[12] Silverman, B.W. (1986). Density Estimation for Statistics and Data Analysis. *Monographs on Statistics and Applied Probability*. Chapman & Hall, London.

[13] World Bank, World Development Indicators. (2017). *Time required to get electricity (days)*. Retrieved from: `https://data.worldbank.org/indicator/IC.ELC.TIME?year_high_desc=true&fbclid=IwAR2Q2kxUsfQ5-0l7ImpKDZxyR6tDrTSVzQFDO397OQkxUSTCYS1YZLmLCHI`

[14] Zambom, A. and Dias, R. (2013). A Review of Kernel Density Estimation with Applications to Econometrics. *International Econometric Review (IER)*, 5(1): 20-42.