HANDOUT #4: SAMPLE ESTIMATORS OF THE CDF, PDF, AND QUANTILE FUNCTION

1. Sample Estimator of Cumulative Distribution Function
   (a) Raw Estimator
   (b) Two versions of Smoothed Estimator
2. Sample Estimator of Quantile Function
   (a) Raw Estimator
   (b) Many versions of a Smoothed Estimator
3. Sample Estimator of pdf(pmf)
   (a) Discrete pmf:
      i. bar graph
      ii. line plot
   (b) Continuous pdf:
      i. Histogram: frequency, relative frequency, density estimator
      ii. Kernel Density Estimator

Supplemental Reading:

- Chapter 2 and Sections 4.1, 4.2, 4.3.4 in Tamhane/Dunlop book

Sample Estimators of the cdf F

Let $Y_1, Y_2, \cdots, Y_n$ independent, identically distributed r.v.'s, that is, a random sample from a population or n independent observations from a random process. Suppose that the $Y_i's$ have cdf F, quantile function Q, and pdf(pmf) f. We will now define sample estimators of F, Q, and f.

Sample Estimator of the cdf $F(\cdot)$:

By definition,

$$F(y) = P[Y \leq y]$$

that is, $F(y)$ is the probability that the r.v. Y is less than or equal to y

or $F(y)$ is the proportion of population values less than or equal to y

or $F(y)$ is the proportion of times that process produces values of Y that are less than or equal to y.

From the above definitions of $F(y)$, a reasonable estimator of $F(y)$ based on the data is given by

Data: n iid realizations, $Y_1, Y_2, \cdots, Y_n$ Our Data

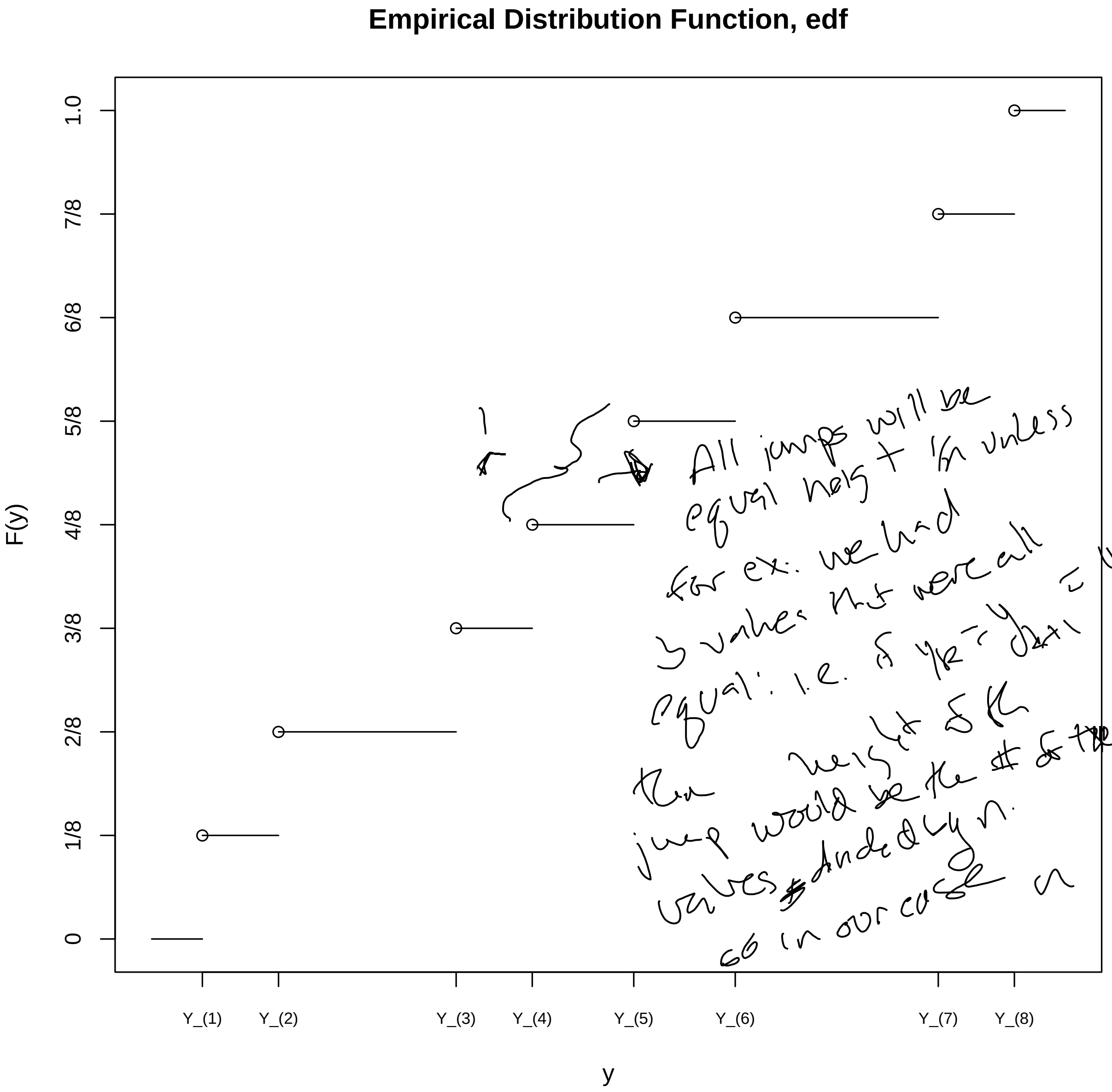
$$\begin{aligned}
\widehat{F}(y) &= \text{ proportion of } \ (Y_1, Y_2, \cdots, Y_n) \leq y \\
&= (\# \text{ of } \ Y_i's \leq y)/n \\
&= \sum_{i=1}^{n} I(Y_i \leq y)/n \\
&= \frac{1}{n} \sum_{i=1}^{n} I(Y_i \leq y), \\
&= \frac{1}{n}[\# Y_i \leq y], \\
&= \hat{p}
\end{aligned}$$

where $I(Y_i \leq y) = 1$ if $Y_i \leq y$ and $I(Y_i \leq y) = 0$ if $Y_i > y$ and

$p = P[Y \leq y]$

$\widehat{F}()$ is called the empirical distribution function (edf).

A graph of the edf is given on the next page.



All jumps will be equal height 1/n unless for ex: we had 3 values that were all equal: i.e. if $Y_k = Y_{k+1} = Y_{k+2}$ then weight of the jump would be the # of tied values ~~#~~ divided by n.

66 in our case n

Let $Y_{(1)} \leq Y_{(2)} \leq \cdots \leq Y_{(n)}$ be the n data values ordered from smallest to largest, called the **order statistics**.

Note, $\widehat{F}()$ is a piecewise constant function with jumps of height $\frac{1}{n}$ at each of the ordered values $Y_{(1)} \leq Y_{(2)} \leq \cdots \leq Y_{(n)}$; unless there are ties (i.e., several Y_i's having the same value). In this case, the height of the jump at the tied value would be $\frac{k}{n}$ where k is the number of Y_i's having the tied value.

The "raw" edf is a step function for all data sets. If the data is from a discrete distribution, then the plot would be an appropriate plot. However, if we have observations from a continuous cdf, then the raw edf, a step function, would not be an accurate portrayal of the population cdf, a continuous function.

A very simple improvement to the raw edf is to simply connect the midpoints of each of the flat regions in the edf, called a **smoothed edf**. If there are no ties in the data, this definition yields a strictly increasing, continuous function which is piecewise linear.

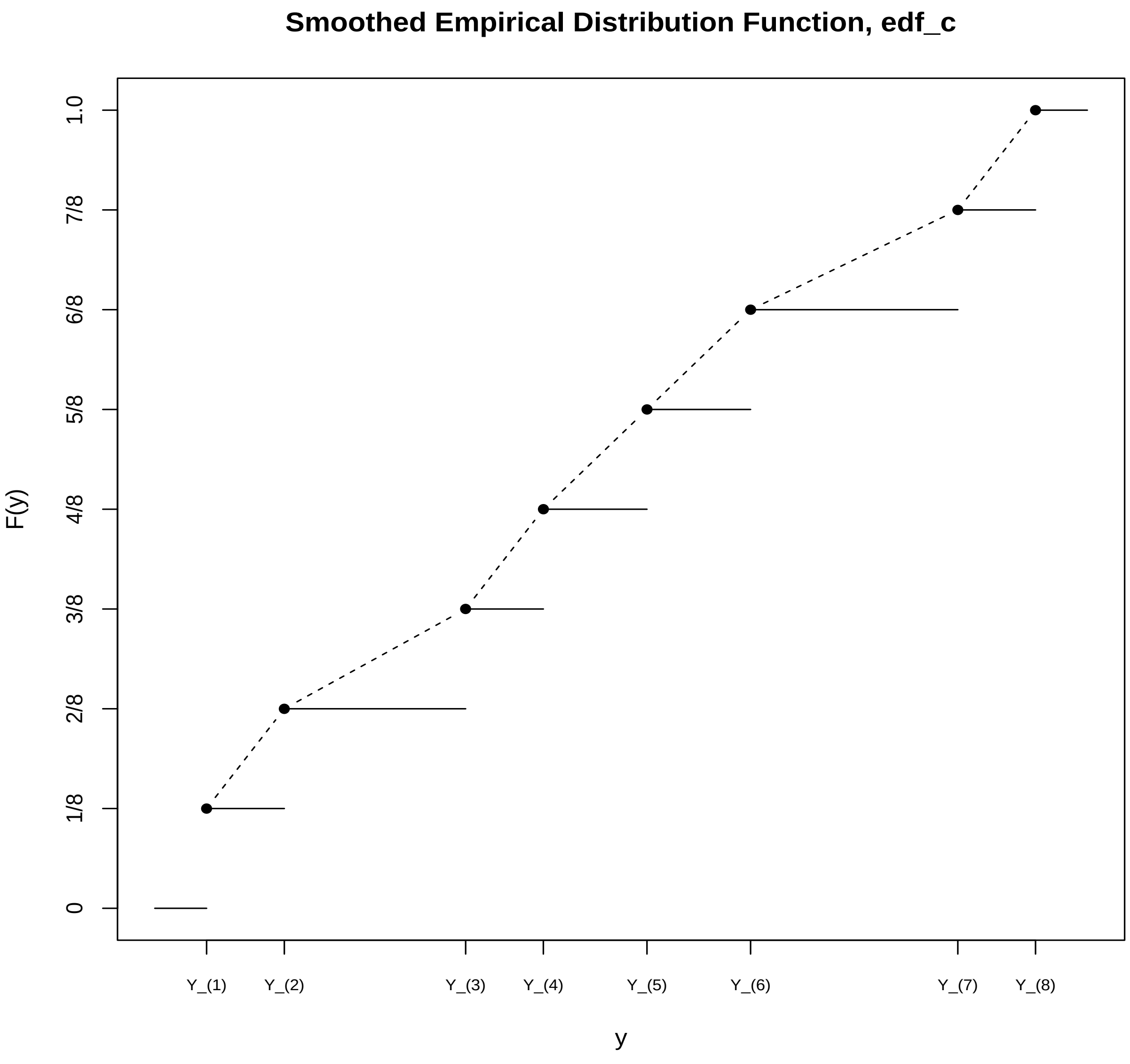
An alternative version of the smoothed edf connects the endpoints of the empirical cdf instead of the midpoints yielding, $\widehat{F}^C(Y_{(i)})$:

$$\widehat{F}^C(y) = \begin{cases} 0 & \text{if } \; y < Y_{(1)} \\ \left[\frac{i}{n}\right] \frac{Y_{(i+1)}-y}{Y_{(i+1)}-Y_{(i)}} + \left[\frac{i+1}{n}\right] \frac{y-Y_{(i)}}{Y_{(i+1)}-Y_{(i)}} & \text{if } \; Y_{(i)} \leq y < Y_{(i+1)} \;\; \text{for } i = 1, \ldots, n-1 \\ 1 & \text{if } \; y \geq Y_{(n)} \end{cases}$$

It can be easily seen that for $Y_{(i)} \leq y < Y_{(i+1)}$ and $i = 1, \ldots, n-1$

$$\widehat{F^C}(y) = \left[\widehat{F}(Y_{(i)})\right] \frac{Y_{(i+1)} - y}{Y_{(i+1)} - Y_{(i)}} + \left[\widehat{F}(Y_{(i+1)})\right] \frac{y - Y_{(i)}}{Y_{(i+1)} - Y_{(i)}}$$

with $\widehat{F^C}(Y_{(k)}) = \widehat{F}(Y_{(k)}) = \frac{k}{n}$ for $k = 1, \ldots, n$


Smoothed Empirical Distribution Function, edf_c
F(y)
0
1/8
2/8
3/8
4/8
5/8
6/8
7/8
1.0
Y_(1)
Y_(2)
Y_(3)
Y_(4)
Y_(5)
Y_(6)
Y_(7)
Y_(8)
y

4(a)

START CLASS NOTES 9/15/ 1

Sample Estimator of the Quantile Function $Q(\cdot)$:

Recall the definition of the quantile function, for $0 \leq u \leq 1$,

IF F is cont. & strictly increasing

$$Q(u) = F^{-1}(u) = \underbrace{inf(y : F(y) \geq u)}.$$

IF F has flat parts or is discontinuous.

The quantile function evaluated at u is thus the value of the r.v. Y, $Q(u)$ for which $F(Q(u)) = u$ for strictly increasing, continuous cdf's F or the smallest value of Y for which $F(Q(u)) \geq u$. Thus, $Q(u)$ is the value of Y for which at least 100u% of the distribution of Y is less than or equal to $Q(u)$ and at least 100(1-u)% of the distribution of Y is greater than or equal to $Q(u)$.

Based on $Y_1, Y_2, \cdots, Y_n$, iid r.v.'s from a distribution with cdf F and quantile function Q, a sample estimator of Q is given by the following expression. Let $\frac{j-1}{n} < u \leq \frac{j}{n}$, then

$$\widehat{Q}^R(u) = \widehat{F}^{-1}(u).$$

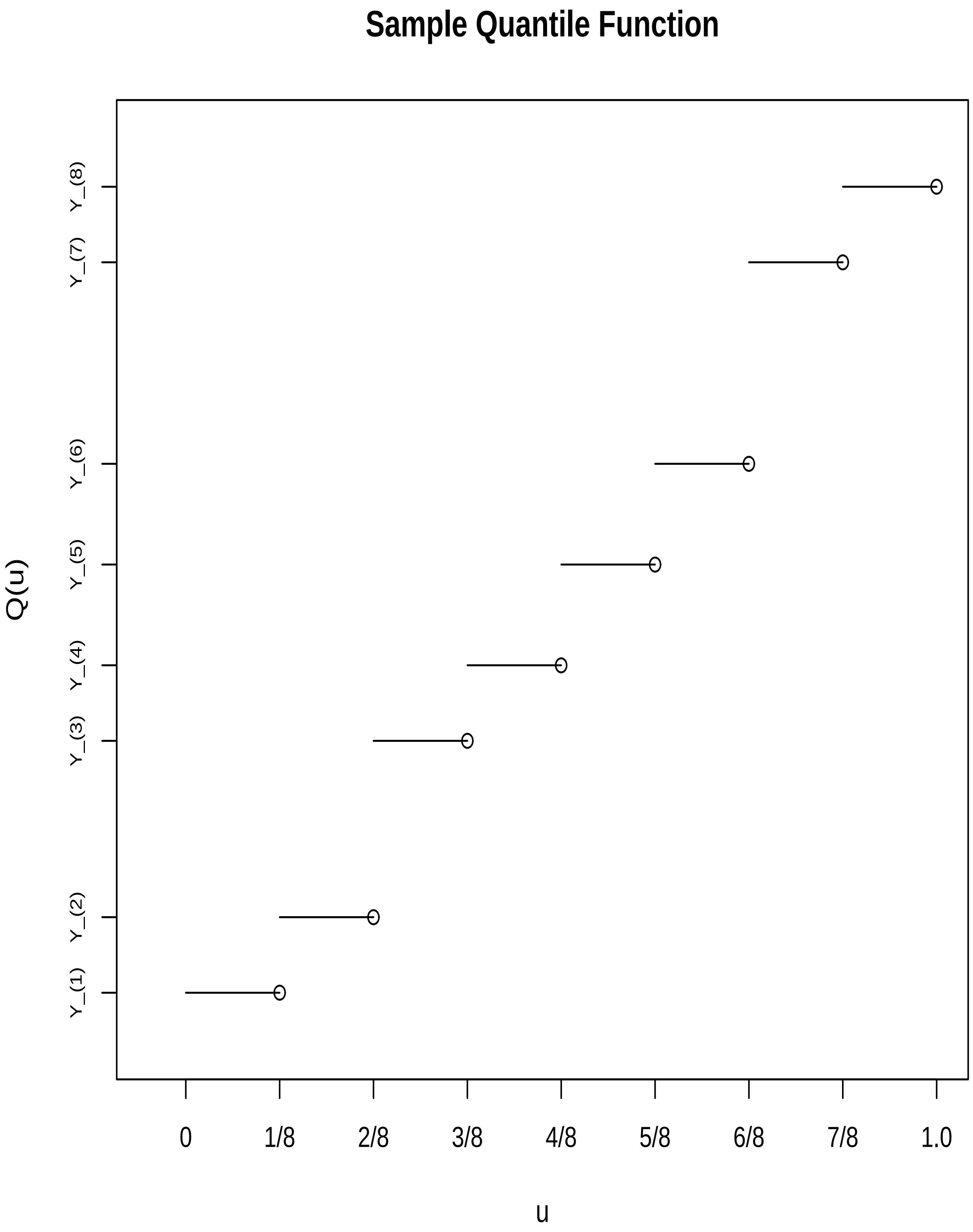
This is a piecewise constant function and is expressed in terms of the order statistics, $Y_{(1)} \leq Y_{(2)} \leq \cdots \leq Y_{(n)}$ of the sample by the following expression.

$$\begin{aligned}
\widehat{Q}^R(u) &= inf(y : \widehat{F}(y) \geq u) \\
&= Y_{(j)} \;\text{ for }\; \frac{j-1}{n} < u \leq \frac{j}{n} \;\text{ for } j = 1, 2, \ldots, n-1 \\
&= Y_{(n)} \;\text{ for }\; 1 - \frac{1}{n} < u < 1
\end{aligned}$$

That is,

$$\begin{aligned}
\widehat{Q}^R(u) &= Y_{(1)} \;\text{ for }\; 0 < u \leq \frac{1}{n} \\
&= Y_{(2)} \;\text{ for }\; \frac{1}{n} < u \leq \frac{2}{n} \\
&\vdots \\
&= Y_{(n-1)} \;\text{ for }\; 1 - \frac{2}{n} < u \leq 1 - \frac{1}{n} \\
&= Y_{(n)} \;\text{ for }\; 1 - \frac{1}{n} < u < 1
\end{aligned}$$

A plot of the raw sample quantile function is given on the next page.


Sample Quantile Function
Q(u)
Y_(1)
Y_(2)
Y_(3)
Y_(4)
Y_(5)
Y_(6)
Y_(7)
Y_(8)
0
1/8
2/8
3/8
4/8
5/8
6/8
7/8
1.0
u

Note that $\widehat{Q}^R()$ is just a rotation of the mirror image of $\widehat{F}()$:

- flat regions in $\widehat{F}()$ become jumps in $\widehat{Q}^R()$
- jumps in $\widehat{F}()$ become flat regions in $\widehat{Q}^R()$.
- $\widehat{F}()$ is right continuous, $\widehat{Q}^R()$ is left continuous, just as was true for the population cdf and quantile functions.

The are several problems with this definition of the sample quantile. STOP. Class Note 9/17/19

Problem 1: $\widehat{Q}^R(.5)$ does not agree with the common definition of the sample median:

$$\widehat{Q}(.5) = \begin{cases} Y_{((n+1)/2)} & \text{if } n \text{ is odd} \\ (Y_{(n/2)} + Y_{(n+2)/2)})/2 & \text{if } n \text{ is even} \end{cases}$$

How we usually define the median

By definition, $F(Y_{(i)}) = \frac{i}{n}$ and $\frac{1}{2} = \frac{n/2}{n}$.

For $\widehat{Q}^R$ agrees if n odd

For n odd, $\frac{n}{2}$ is not an integer, but $i = \frac{n+1}{2}$ is the smallest integer such that $F(Y_{(i)}) \geq \frac{1}{2}$

$$\text{therefore,} \qquad inf\left\{y: \ \widehat{F}(y) \geq \frac{1}{2}\right\} = Y_{(\frac{n+1}{2})}, \ \Rightarrow \ \widehat{Q}^R(.5) = Y_{(\frac{n+1}{2})} = \widehat{Q}(.5)$$

Problem if n even.

For n even, $\frac{n}{2}$ is an integer,

$$\text{therefore,} \qquad inf\left\{y: \ \widehat{F}(y) \geq \frac{1}{2}\right\} = Y_{(\frac{n}{2})}, \ \Rightarrow \ \widehat{Q}^R(.5) = Y_{(\frac{n}{2})} \neq \widehat{Q}(.5)$$

$$Y_{(\frac{n}{2})} \neq \frac{(Y_{(\frac{n}{2})} + Y_{(\frac{n+2}{2})})}{2}$$

Problem 2:

- $Q(0)$ is the smallest possible value of Y in the population,
- $Q(1)$ is the largest possible value of Y in the population.

However, the above definition of $\widehat{Q}^R(u)$ yields

$\widehat{Q}^R(0) = Y_{(1)}$ and $\widehat{Q}^R(1) = Y_{(n)}$

These are not realistic estimators because we would not expect the smallest and largest values in the sample to always be very close to the smallest and largest values in the population, especially when n is relatively small. Furthermore, the raw sample quantile function is a piecewise constant function, whereas the population quantile function is continuous. Therefore, we will define several alternatives to the *raw* estimator $\widehat{Q}^R(u)$ of the quantile function.

We can define a continuous sample quantile $Q^C(u)$ in terms of the **fractional order statistics**, $Y_{(k+r)}$ as defined by the following equation:

For $k = 1, \ldots, n-1$ and $0 < r < 1$, define the $k + r$ fractional order statistic, $Y_{(k+r)}$, by

$$Y_{(k+r)} = Y_{(k)} + r[Y_{(k+1)} - Y_{(k)}] = (1-r)Y_{(k)} + rY_{(k+1)}$$

For example, $Y_{(2.4)} = Y_{(2)} + .4\left(Y_{(3)} - Y_{(2)}\right) = .6Y_{(2)} + .4Y_{(3)}$

```
         |             |                  |
         |<-- 40%  --->|<------ 60% ----->|
---------|-------------|------------------|-------
       Y_(2)        Y_(2.4)             Y_(3)
```

- $\widehat{Q}^C(u)$, the continuous sample quantile function is then defined by

$$\widehat{Q}^C(u) = Y_{(nu+.5)} \quad \text{for} \quad \frac{1}{2n} \leq u \leq 1 - \frac{1}{2n}$$

- $\widehat{Q}^C(u)$ is undefined for $0 \leq u < \frac{1}{2n}$ and $1 - \frac{1}{2n} < u \leq 1$.

By having $\widehat{Q}^C(u)$ undefined for $0 \leq u < \frac{1}{2n}$ and $1 - \frac{1}{2n} < u \leq 1$, the problem of having an inappropriate estimate of $Q(u)$ for very small values of u is eliminated.

If the sample of n observations consists of n distinct values (no ties), then $\widehat{Q}^C(u)$ is a piecewise linear function connecting the values

$$Y_{(j)} = \widehat{Q}^C\left(\frac{j-.5}{n}\right) = \widehat{Q}^C\left(\frac{2j-1}{2n}\right)$$

That is

$$Y_{(1)} = \widehat{Q}^C\left(\frac{1}{2n}\right), \quad Y_{(2)} = \widehat{Q}^C\left(\frac{3}{2n}\right), \quad \ldots, \quad Y_{(n-1)} = \widehat{Q}^C\left(\frac{2n-3}{2n}\right) \quad Y_{(n)} = \widehat{Q}^C\left(1 - \frac{1}{2n}\right)$$

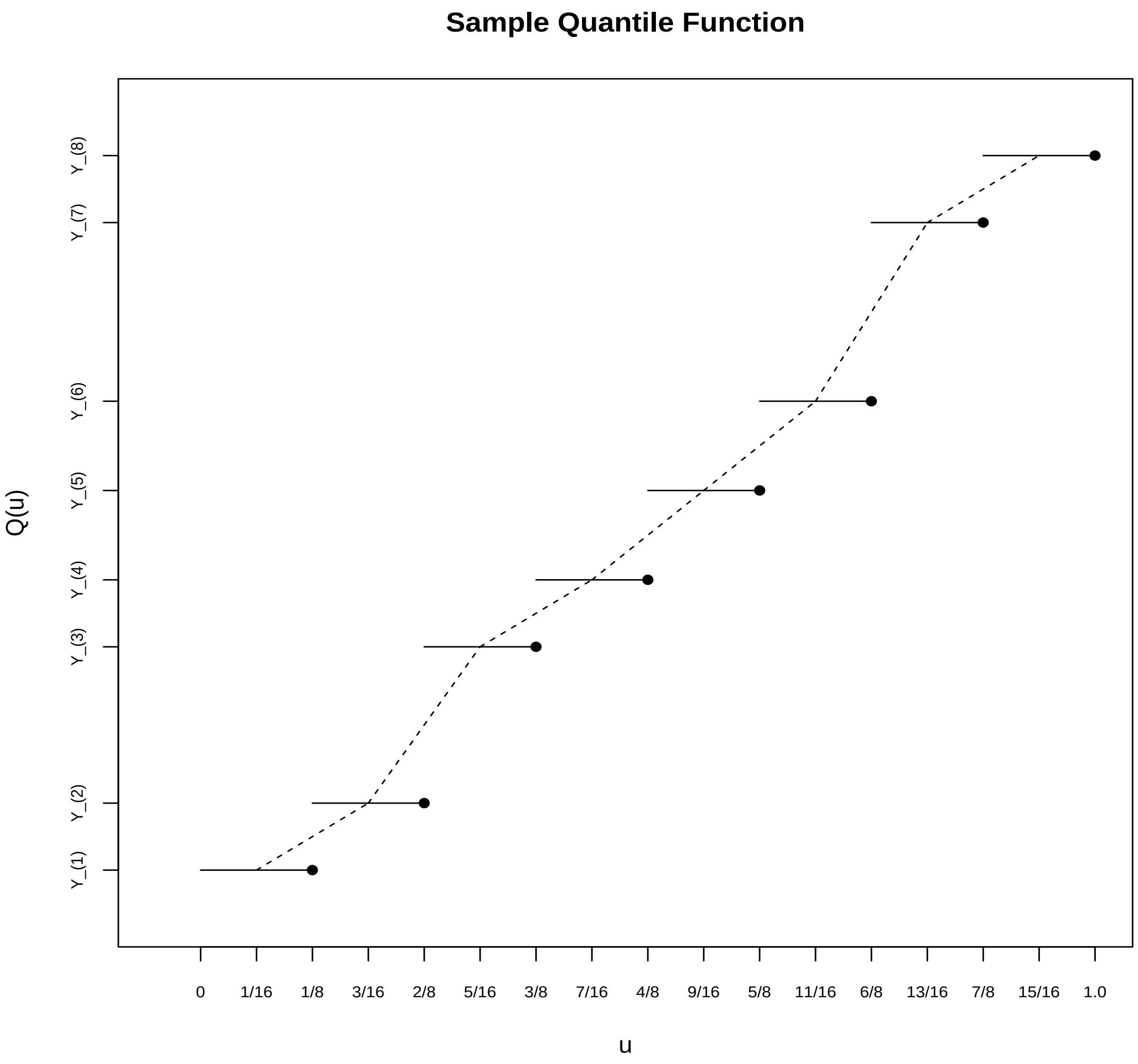
$\widehat{Q}^C(u)$ just connects these points with a straight line.

- For n odd, $\widehat{Q}^C(.5) = Y_{(.5n+.5)} = Y_{(\frac{n+1}{2})}$ because $\frac{n+1}{2}$ is an integer

- For n even $\frac{n+1}{2}$ is noninteger and $\frac{n}{2} < \frac{n+1}{2} < \frac{n+2}{2} \Rightarrow$

  $\widehat{Q}^C(.5) = Y_{(\frac{n+1}{2})} = .5Y_{(n/2)} + .5Y_{((n+2)/2)}$

Thus, $\widehat{Q}^C(.5)$ is consistent with the standard definition of the sample median.


Sample Quantile Function
Q(u)
Y_(1)
Y_(2)
Y_(3)
Y_(4)
Y_(5)
Y_(6)
Y_(7)
Y_(8)
0
1/16
1/8
3/16
2/8
5/16
3/8
7/16
4/8
9/16
5/8
11/16
6/8
13/16
7/8
15/16
1.0
u

8(a)

A number of the software packages use variations on the definition of $\widehat{Q}^C(u)$. They are all piecewise linear functions but differ on the value of u for which $\widehat{Q}(u)$ is assigned to the n order statistics.

1. The Excel definition has $\widehat{Q}((j-1)/(n-1))$ assigned to $Y_{(j)}$. Then connect these values with a piecewise linear function: $\widehat{Q}(u) = Y_{((n-1)u+1)} \;\Rightarrow\; j = (n-1)u+1 \;\Rightarrow\; u = \frac{j-1}{n-1}$

- $\widehat{Q}(0) = Y_{(1)}$ and $\widehat{Q}(1) = Y_{(n)}$, same problem as with $\widehat{Q}^R(0)$ and $\widehat{Q}^R(1)$

2. Minitab and the Tamhane/Dunlop book have $\widehat{Q}(j/(n+1))$ assigned to $Y_{(j)}$. Then connect these values with a piecewise linear function: $\widehat{Q}(u) = Y_{((n+1)u)}$ with $\widehat{Q}(u)$ undefined for $0 \le u < \frac{1}{n+1}$ and $\frac{n}{n+1} < u \le 1$

- $\widehat{Q}(\frac{1}{n+1}) = Y_{(1)}$ and $\widehat{Q}(\frac{n}{n+1}) = Y_{(n)}$, just slightly different from $\widehat{Q}^C(u)$

3. R, SAS, and SPSS have 5-10 different definitions. The default definition in SAS and SPSS is $\widehat{Q}^C(u) = Y_{(nu+.5)}$, the same definition as Parzen quantile, $\widehat{Q}^C(u)$.

   The default in R is $\widehat{Q}(u) = Y_{((n-1)u+1)}$

   To almost obtain the Parzen definition in R, you need to specify type=5 in the quantile function: **quantile(x,type=5)**

   However, $\widehat{Q}(u) = Y(1)$ for $0 \le u \le \frac{1}{2n}$ and $\widehat{Q}(u) = Y(n)$ for $1 - \frac{1}{2n} \le u \le 1$

4. A unified definition in terms of parameters a and b is given by

$$\widehat{Q}(u; a, b) = Y_{((n-b)u+a)}$$

   John Tukey recommends using $a = 3/8, b = 1/4$ which yields assigning the order statistics

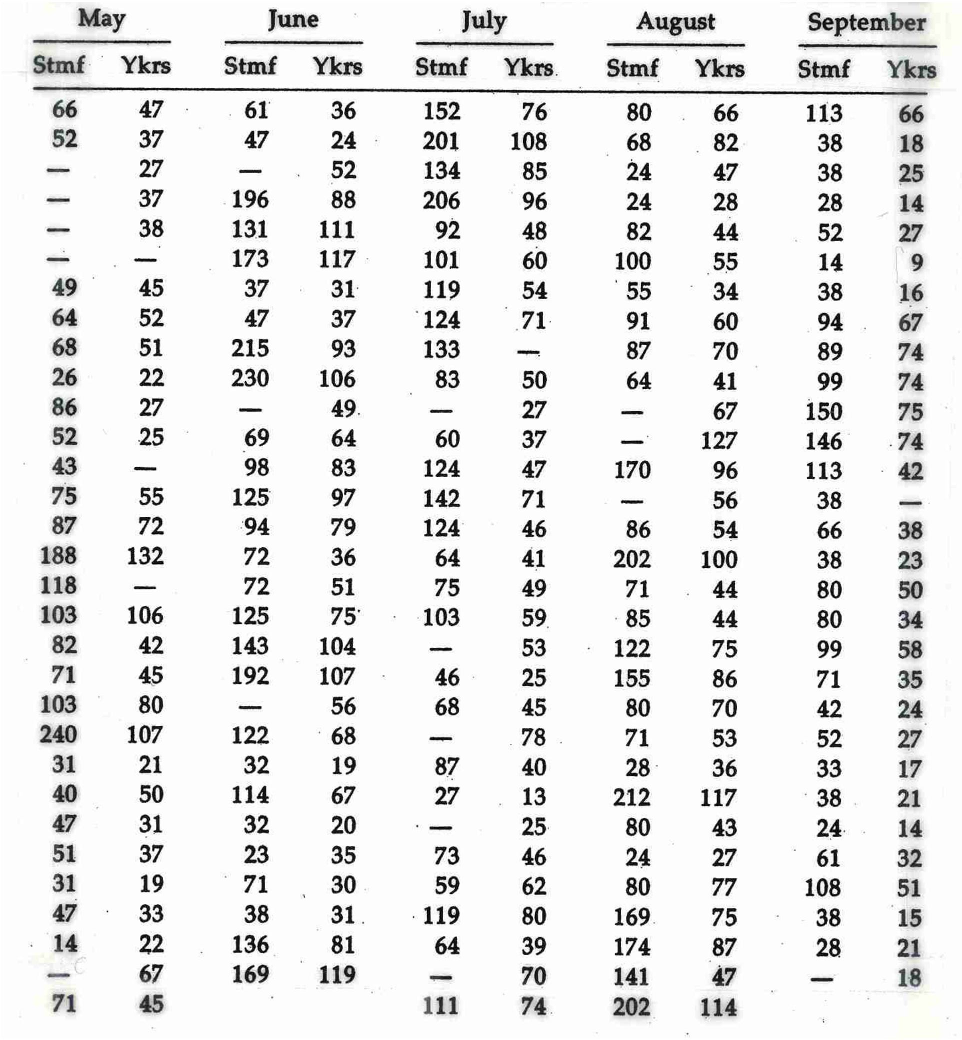
$$Y_{(j)} \quad \text{to} \quad \widehat{Q}\left(\frac{j-\frac{3}{8}}{n-\frac{1}{4}}\right) \text{ whereas } Q^C(u) \text{ assigns } Y_{(j)} \text{ to } \widehat{Q}^C\left(\frac{j-.5}{n}\right)$$

   1. $a = b = 1 \;\Rightarrow\; \widehat{Q}(u) = Y_{((n-1)u+1)}$
   2. $a = 0,\; b = -1 \;\Rightarrow\; \widehat{Q}(u) = Y_{((n+1)u)}$
   3. Many variations of the definition
   4. $a = .5,\; b = 0 \;\Rightarrow\; \widehat{Q^c}(u) = Y_{(nu+.5)}$

We will illustrate the computation of the above plots using SAS and R on an ozone concentration data set.

Maximum Daily Ozone Concentrations

Daily maximum ozone concentrations in parts per billion (ppb) at ground level recorded between May 1 and September 30 at sites in Stamford,Connecticut and Yonkers, New York are given below. (There are 17 missing days of data at Stamford and 5 at Yonkers due to equipment malfunction.) The current federal standard for ozone states that the concentration should not exceed 120 ppb more than one day per year at any particular location. A day with ozone concentration above 220 ppb is regarded as heavily polluted. Initially, we will disregard the time aspect of the data and just consider the data set as random samples of ozone readings from the two cities. There are 5 missing values for Yonkers and 17 for Stamford yielding $n_Y = 148$ and $n_S = 136$



May		June		July		August		September	
Stmf	Ykrs	Stmf	Ykrs	Stmf	Ykrs	Stmf	Ykrs	Stmf	Ykrs
66	47	61	36	152	76	80	66	113	66
52	37	47	24	201	108	68	82	38	18
—	27	—	52	134	85	24	47	38	25
—	37	196	88	206	96	24	28	28	14
—	38	131	111	92	48	82	44	52	27
—	—	173	117	101	60	100	55	14	9
49	45	37	31	119	54	55	34	38	16
64	52	47	37	124	71	91	60	94	67
68	51	215	93	133	—	87	70	89	74
26	22	230	106	83	50	64	41	99	74
86	27	—	49	—	27	—	67	150	75
52	25	69	64	60	37	—	127	146	74
43	—	98	83	124	47	170	96	113	42
75	55	125	97	142	71	—	56	38	—
87	72	94	79	124	46	86	54	66	38
188	132	72	36	64	41	202	100	38	23
118	—	72	51	75	49	71	44	80	50
103	106	125	75	103	59	85	44	80	34
82	42	143	104	—	53	122	75	99	58
71	45	192	107	46	25	155	86	71	35
103	80	—	56	68	45	80	70	42	24
240	107	122	68	—	78	71	53	52	27
31	21	32	19	87	40	28	36	33	17
40	50	114	67	27	13	212	117	38	21
47	31	32	20	—	25	80	43	24	14
51	37	23	35	73	46	24	27	61	32
31	19	71	30	59	62	80	77	108	51
47	33	38	31	119	80	169	75	38	15
14	22	136	81	64	39	174	87	28	21
—	67	169	119	—	70	141	47	—	18
71	45			111	74	202	114		

The following R Code generates various graphical representations of the
Ozone data. The ozone data is in the files ozone1.DAT and ozone2.DAT

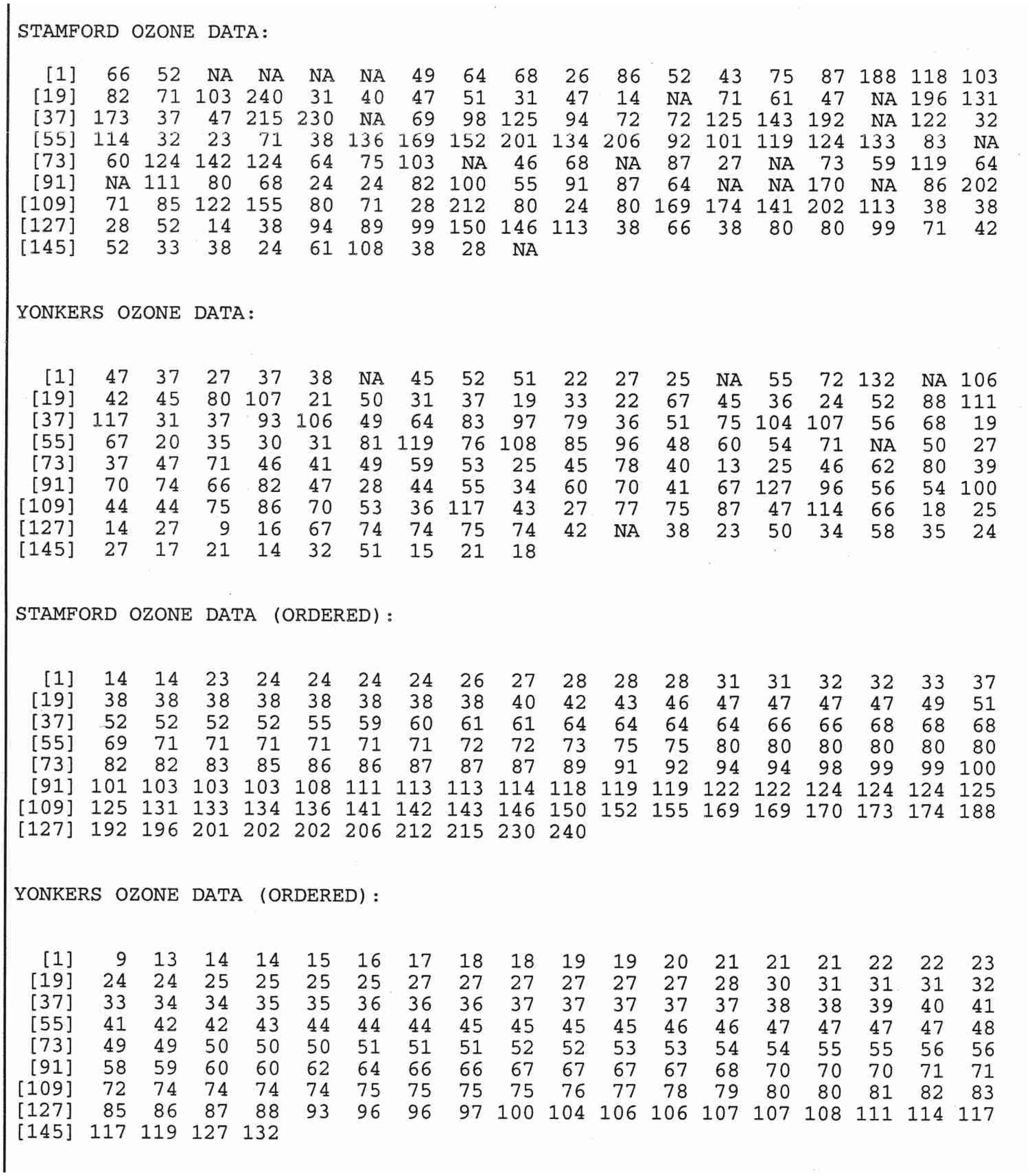
input the data from data files:

```
          y1  =  scan("u:/meth1/Rfiles/StanfordOzoneData.DAT")
          y2  =  scan("u:/meth1/Rfiles/YonkersOzoneData.DAT")
          y1na  =  scan("u:/meth1/Rfiles/ozone1,na.DAT")
          y2na  =  scan("u:/meth1/Rfiles/ozone2,na.DAT")
```

output data to file named ozonedata:

```
          sink("u:/meth1/output/ozonedata")
          y1
          y2
          y1s = sort(y1)
          y2s = sort(y2)
          y1s
          y2s
          sink()
```

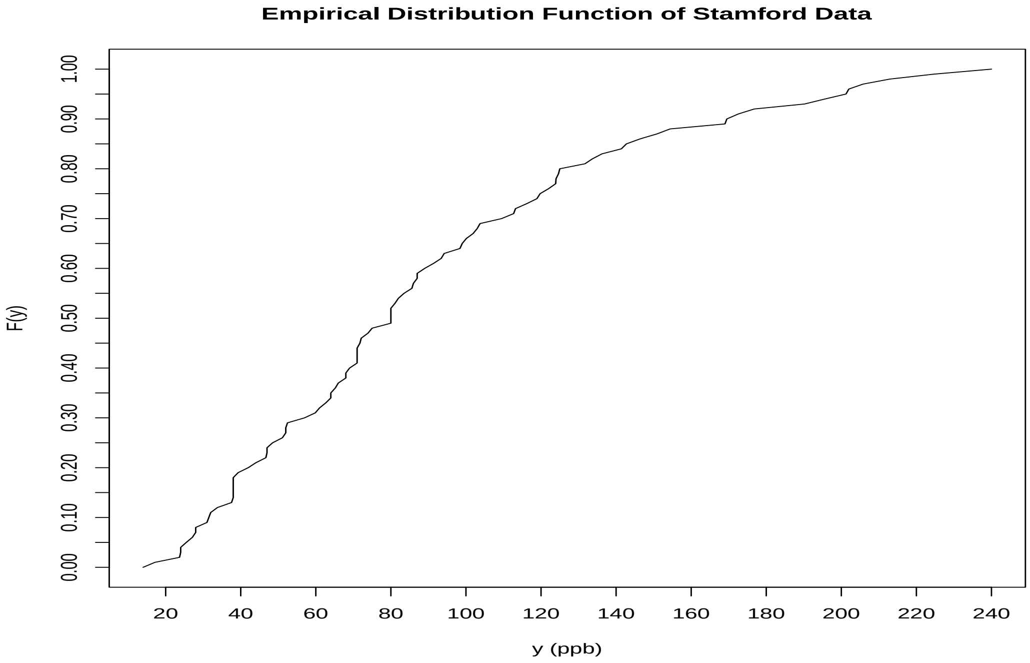
The next page is the output from the **sink** command which creates a file containing the original and sorted ozone data.

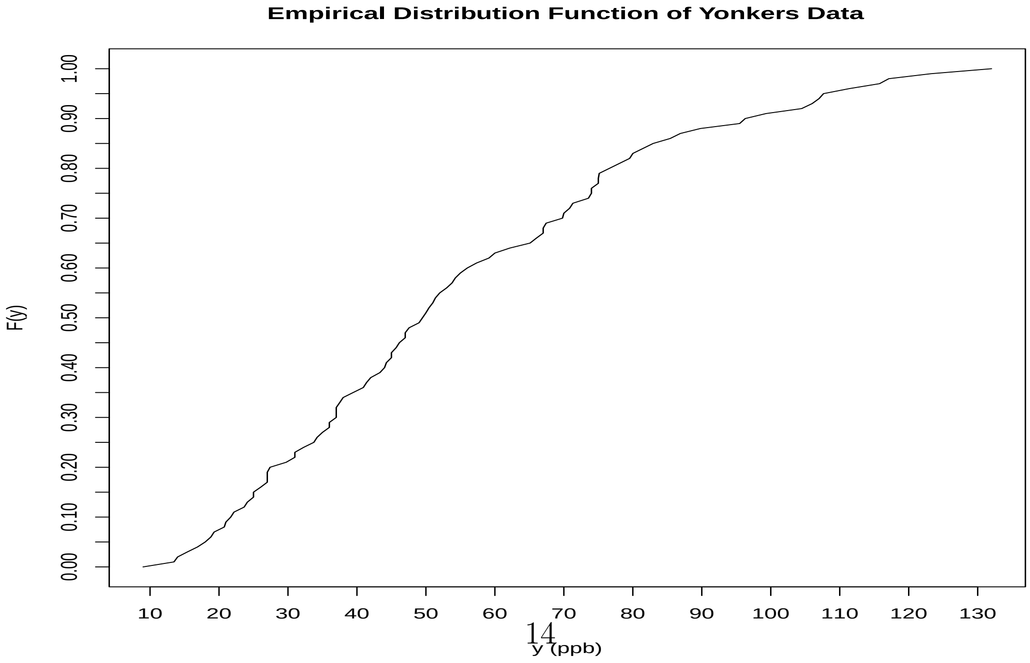


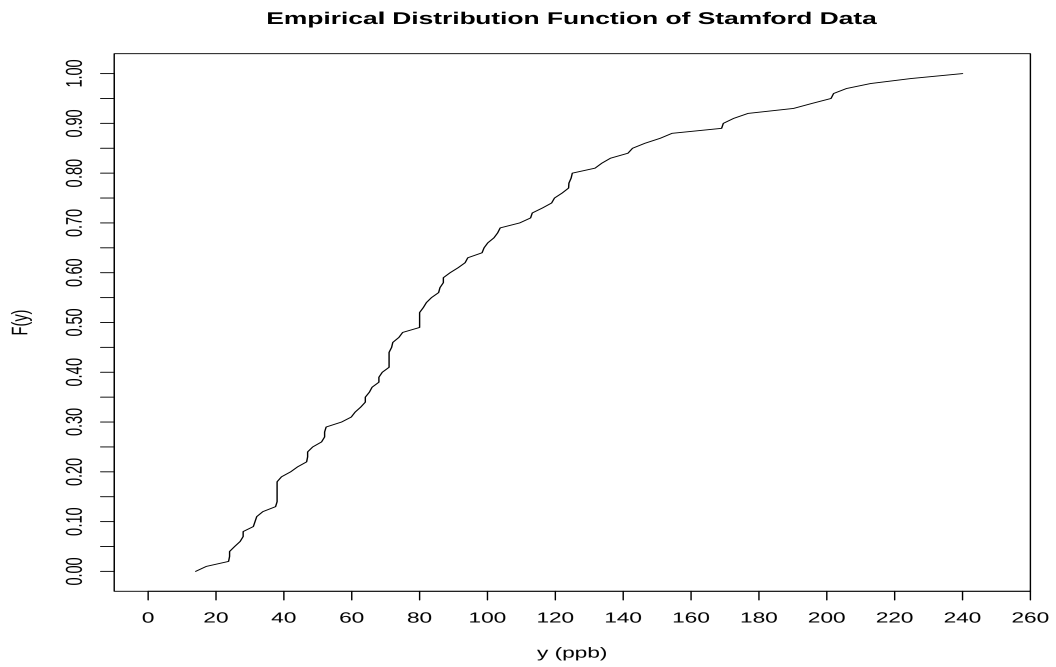
```
STAMFORD OZONE DATA:

  [1]  66  52  NA  NA  NA  NA  49  64  68  26  86  52  43  75  87 188 118 103
 [19]  82  71 103 240  31  40  47  51  31  47  14  NA  71  61  47  NA 196 131
 [37] 173  37  47 215 230  NA  69  98 125  94  72  72 125 143 192  NA 122  32
 [55] 114  32  23  71  38 136 169 152 201 134 206  92 101 119 124 133  83  NA
 [73]  60 124 142 124  64  75 103  NA  46  68  NA  87  27  NA  73  59 119  64
 [91]  NA 111  80  68  24  24  82 100  55  91  87  64  NA  NA 170  NA  86 202
[109]  71  85 122 155  80  71  28 212  80  24  80 169 174 141 202 113  38  38
[127]  28  52  14  38  94  89  99 150 146 113  38  66  38  80  80  99  71  42
[145]  52  33  38  24  61 108  38  28  NA

YONKERS OZONE DATA:

  [1]  47  37  27  37  38  NA  45  52  51  22  27  25  NA  55  72 132  NA 106
 [19]  42  45  80 107  21  50  31  37  19  33  22  67  45  36  24  52  88 111
 [37] 117  31  37  93 106  49  64  83  97  79  36  51  75 104 107  56  68  19
 [55]  67  20  35  30  31  81 119  76 108  85  96  48  60  54  71  NA  50  27
 [73]  37  47  71  46  41  49  59  53  25  45  78  40  13  25  46  62  80  39
 [91]  70  74  66  82  47  28  44  55  34  60  70  41  67 127  96  56  54 100
[109]  44  44  75  86  70  53  36 117  43  27  77  75  87  47 114  66  18  25
[127]  14  27   9  16  67  74  74  75  74  42  NA  38  23  50  34  58  35  24
[145]  27  17  21  14  32  51  15  21  18

STAMFORD OZONE DATA (ORDERED):

  [1]  14  14  23  24  24  24  24  26  27  28  28  28  31  31  32  32  33  37
 [19]  38  38  38  38  38  38  38  38  40  42  43  46  47  47  47  47  49  51
 [37]  52  52  52  52  55  59  60  61  61  64  64  64  64  66  66  68  68  68
 [55]  69  71  71  71  71  71  71  72  72  73  75  75  80  80  80  80  80  80
 [73]  82  82  83  85  86  86  87  87  87  89  91  92  94  94  98  99  99 100
 [91] 101 103 103 103 108 111 113 113 114 118 119 119 122 122 124 124 124 125
[109] 125 131 133 134 136 141 142 143 146 150 152 155 169 169 170 173 174 188
[127] 192 196 201 202 202 206 212 215 230 240

YONKERS OZONE DATA (ORDERED):

  [1]   9  13  14  14  15  16  17  18  18  19  19  20  21  21  21  22  22  23
 [19]  24  24  25  25  25  25  27  27  27  27  27  27  28  30  31  31  31  32
 [37]  33  34  34  35  35  36  36  36  37  37  37  37  37  38  38  39  40  41
 [55]  41  42  42  43  44  44  44  45  45  45  45  46  46  47  47  47  47  48
 [73]  49  49  50  50  50  51  51  51  52  52  53  53  54  54  55  55  56  56
 [91]  58  59  60  60  62  64  66  66  67  67  67  67  68  70  70  70  71  71
[109]  72  74  74  74  74  75  75  75  75  76  77  78  79  80  80  81  82  83
[127]  85  86  87  88  93  96  96  97 100 104 106 106 107 107 108 111 114 117
[145] 117 119 127 132
```

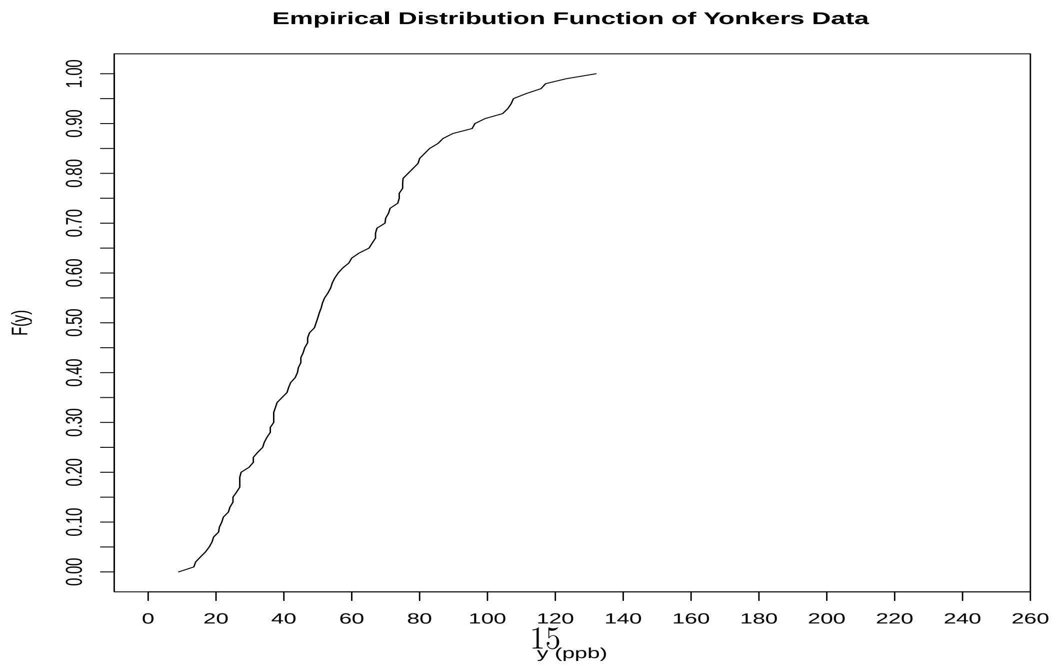
```
File Name:   ozone_CdfQ.R

#input the data from data files:
          y1  =  scan("u:/meth1/Rfiles/StandfordOzoneData.DAT")
          y2  =  scan("u:/meth1/Rfiles/YonkersOzoneData.DAT")
          y1na  =  scan("u:/meth1/Rfiles/ozone1,na.DAT")
          y2na  =  scan("u:/meth1/Rfiles/ozone2,na.DAT")

#creates an empirical cdf (edf) plot for the two locations:


          Q1 = quantile(y1,probs = seq(0,1,.01))
          plot(Q1,probs,type="l",xlab="y (ppb)",ylab="F(y)",
               ylim=c(0,1),lab=c(12,20,7),cex=.75)
          title("Empirical Distribution Function of Stamford Data",cex=.75)

          Q2 = quantile(y2,probs = seq(0,1,.01))
          plot(Q2,probs,type="l",xlab="y (ppb)",ylab="F(y)",
               ylim=c(0,1),lab=c(12,20,7),cex=.75)
          title("Empirical Distribution Function of Yonkers Data",cex=.75)

          Q1 = quantile(y1,probs = seq(0,1,.01))
          plot(Q1,probs,type="l",xlab="y (ppb)",ylab="F(y)",xlim=c(0,250),
               ylim=c(0,1),lab=c(12,20,7),cex=.75)
          title("Empirical Distribution Function of Stamford Data",cex=.75)


          Q2 = quantile(y2,probs = seq(0,1,.01))
          plot(Q2,probs,type="l",xlab="y (ppb)",ylab="F(y)",xlim=c(0,250),
               ylim=c(0,1),lab=c(12,20,7),cex=.75)
          title("Empirical Distribution Function of Yonkers Data",cex=.75)

#creates an empirical quantile plot for the two locations:

          Q1 = quantile(y1,probs = seq(0,1,.01))
          plot(probs,Q1,type="l",ylab="Q(u) (ppb)",xlab="u",ylim=c(0,250),
               xlim=c(0,1),lab=c(10,11,7))
          title("Empirical Quantile of Stamford Data",cex=.75)

          Q2 = quantile(y2,probs = seq(0,1,.01))
          plot(probs,Q2,type="l",ylab="Q(u) (ppb)",xlab="u",ylim=c(0,250),
               xlim=c(0,1),lab=c(10,11,7))
          title("Empirical Quantile of Yonkers Data",cex=.75)
```

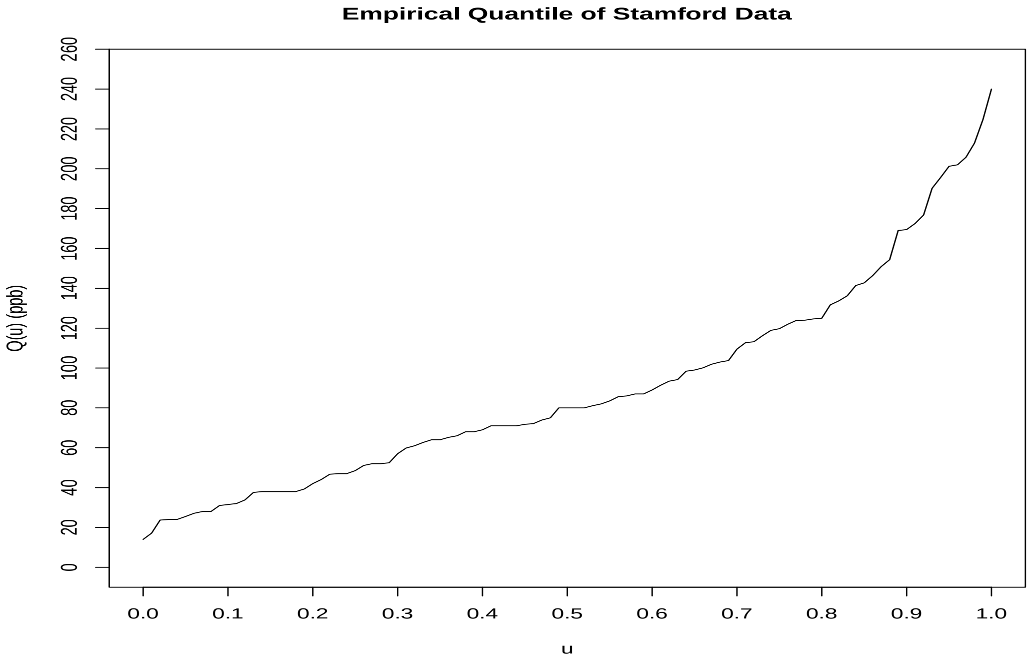
The empirical cdf and sample quantile functions are given on the next pages.

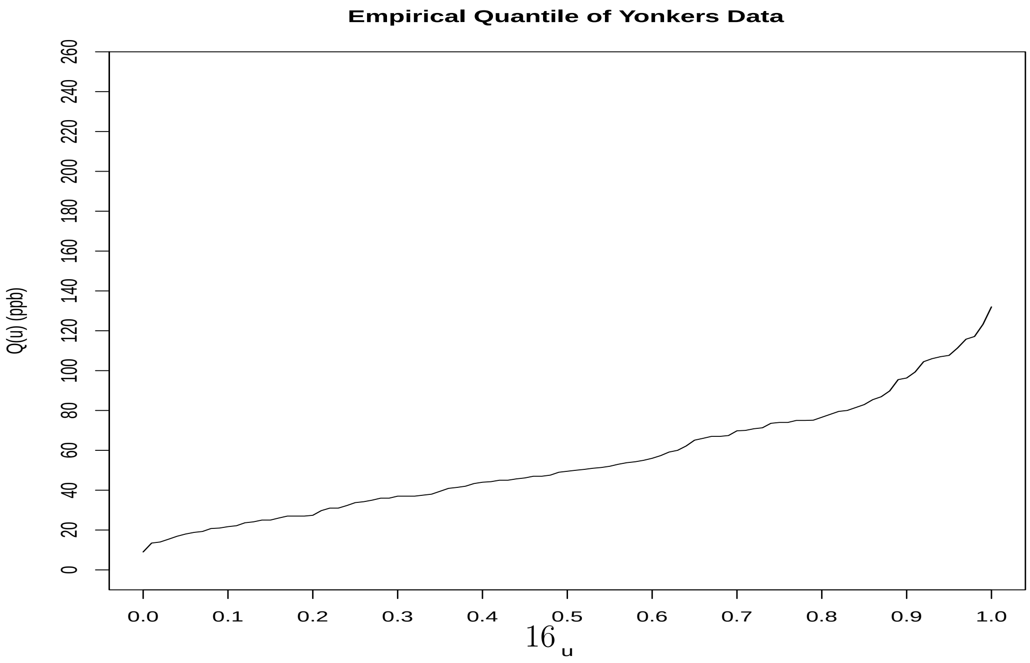

Empirical Distribution Function of Stamford Data
F(y)
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
20
40
60
80
100
120
140
160
180
200
220
240
y (ppb)


Empirical Distribution Function of Yonkers Data
F(y)
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
10
20
30
40
50
60
70
80
90
100
110
120
130
y (ppb)


Empirical Distribution Function of Stamford Data
F(y)
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
0
20
40
60
80
100
120
140
160
180
200
220
240
260
y (ppb)


Empirical Distribution Function of Yonkers Data
F(y)
0.00
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00
0
20
40
60
80
100
120
140
160
180
200
220
240
260
y (ppb)


Empirical Quantile of Stamford Data


Empirical Quantile of Yonkers Data

Sample Estimator of the Probability Density (Mass) Function $f(\cdot)$:

In this section we will discuss methods for estimating the pdf (pmf) for a population/process based on a random sample (iid r.v.'s) $Y_1, Y_2, \cdots, Y_n$ from a population/process having pdf (pmf) $f(\cdot)$.

Case 1 Discrete Distributions:

Let $f(\cdot)$ be the probability mass function for a discrete r.v. Y having k possible values $y_1, y_2, \cdots, y_k$ with probabilities $f(y_j) = Pr[Y = y_j]$ for $j = 1, 2, \cdots, k$.

Suppose we have n iid observations $Y_1, Y_2, \cdots, Y_n$ on Y with observed frequencies

$$\widehat{f}_j = \{\#Y_i = y_j\} = \sum_{i=1}^{n} I(Y_i = y_j)$$

# of tm $Y_i = y_j$

Estimate $f(y) = Pr[Y = y]$ with

$$\widehat{f}(y) = \begin{cases} \widehat{f}_j/n & \text{if } \; y = y_j \\ 0 & \text{otherwise} \end{cases}$$

Suppose we have a population/process consisting of units which contain 10 individual parts packaged in sealed containers. A container is randomly selected and the 10 parts are inspected. Let Y be the r.v. which represents the number of defectives in each of the sealed containers in a large warehouse. Thus, Y has values $0, 1, 2, \cdots, 10$. Suppose we observe the inspection results from 1000 randomly selected containers over a long period of time and obtain the following frequency table with

y_j the number of defects per containers,

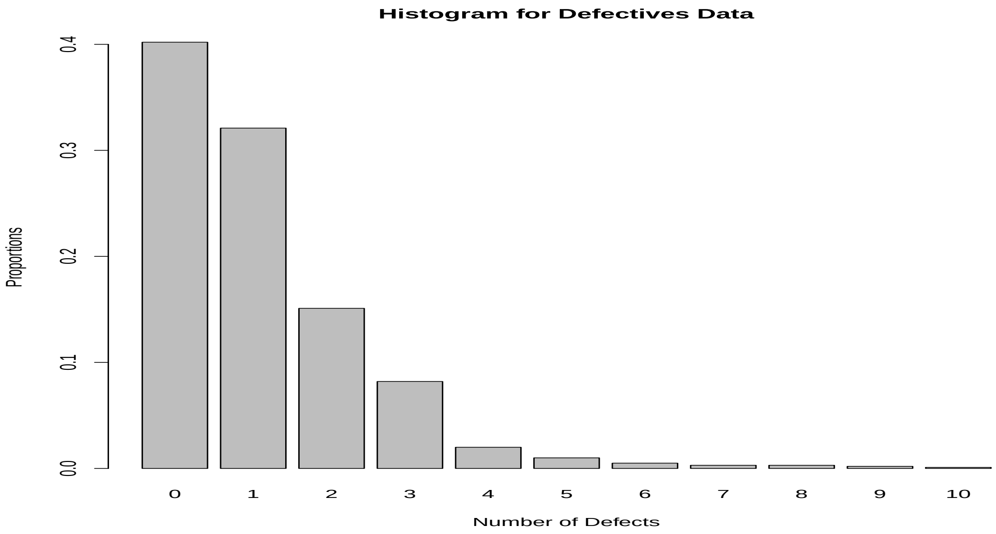
f_j the number of containers with y_j defects, and

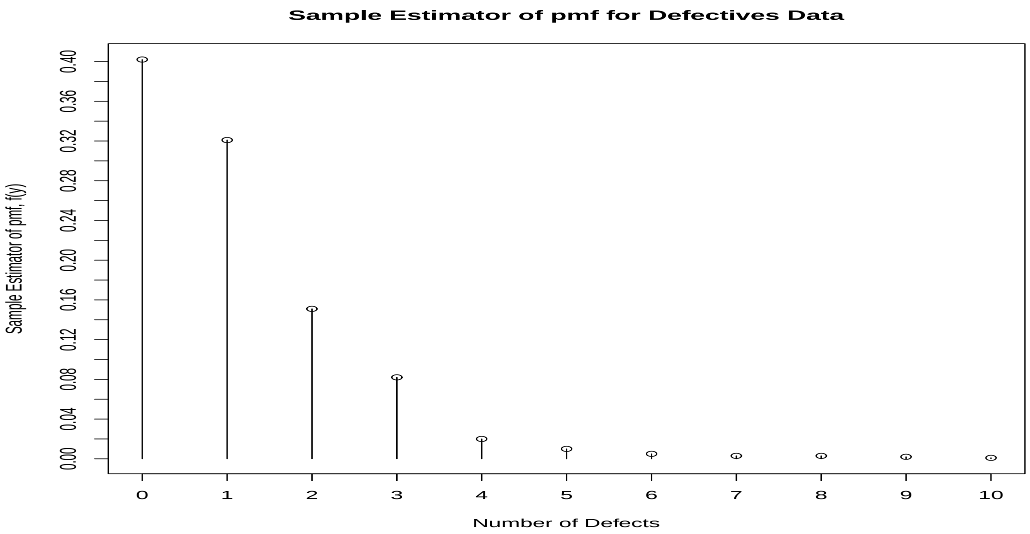
$\hat{f}(y_j) = f_j/1000$, the estimated pmf for Y at y_j

which is the proportion of the containers with y_j defects.

y_j	0	1	2	3	4	5	6	7	8	9	10	Total
f_j	402	321	151	82	20	10	5	3	3	2	1	1000
$\hat{f}(y_j)$	.402	.321	.151	.082	.020	.010	.005	.003	.003	.002	.001	1.0

```
#File name: pmf,defects.R

#R code for plotting a bar chart for discrete data and
# a line chart for the estimated pmf


defects.f = c(402,321,151,82,20,10,5,3,3,2,1)

defects.n = c("0","1","2","3","4","5","6","7","8","9","10")

barplot(prop.table(defects.f),names=defects.n,ylab="Proportions",
xlab="Number of Defects",beside=F)

title("Histogram for Defectives Data")

postscript("u:/meth1/Rfiles/defectsf2.ps",height=8,horizontal=F)

defects = c(0,1,2,3,4,5,6,7,8,9,10)

zero = c(0,0,0,0,0,0,0,0,0,0,0)

plot(defects,defects.f/1000,ylab="Sample Estimator of pmf, f(y)",
xlab="Number of Defects",lab=c(11,17,4))

segments(defects,zero,defects,defects.f/1000)

title("Sample Estimator of pmf for Defectives Data")
```


Histogram for Defectives Data
Proportions
0.0
0.1
0.2
0.3
0.4
0
1
2
3
4
5
6
7
8
9
10
Number of Defects


Sample Estimator of pmf for Defectives Data
Sample Estimator of pmf, f(y)
0.00
0.04
0.08
0.12
0.16
0.20
0.24
0.28
0.32
0.36
0.40
0
1
2
3
4
5
6
7
8
9
10
Number of Defects

Case 2 Continuous Distributions:

Let $f(\cdot)$ be the probability density function for a r.v. Y having a strictly increasing continuous distribution function.

Suppose we have Data: n iid observations $Y_1, Y_2, \cdots, Y_n$ on Y.

That is, we have observed n iid realizations from the random variable Y

Following the methodology for finding estimators of the pmf, cdf and quantile functions, we need to first examine the basic definition of the pdf in order to develop its estimator, $\hat{f}(x)$:

$$f(y) = \frac{d}{dy}\overbrace{F(y)}^{\text{cdf}} \Rightarrow f(y) = \lim_{\Delta \to 0} \frac{F(y + \frac{1}{2}\Delta) - F(y - \frac{1}{2}\Delta)}{\Delta} \Rightarrow$$

$$\text{for small } \Delta, \qquad \Delta f(y) \approx F(y + \frac{1}{2}\Delta) - F(y - \frac{1}{2}\Delta) \Rightarrow$$

$$\Delta f(y) \approx Pr\left(y - \frac{1}{2}\Delta \leq Y \leq y + \frac{1}{2}\Delta\right)$$

We will use this interpretation to obtain estimators for $f(y)$ based on n iid realizations on Y.

Definition The Local Density of the distribution of Y at a value $Y = y$ is the relative concentration of the distribution of Y in an interval of width h centered at $y : (y - \frac{1}{2}h, y + \frac{1}{2}h)$.

Let $\widehat{f}(y)$ be the estimated local density at y, then

$$\widehat{f}(y) = \frac{\text{fraction of n data values in } (y - \frac{1}{2}h, y + \frac{1}{2}h)}{\text{length of interval}}$$

$$\widehat{f}(y) = \frac{[\text{\#of n data values in } (y - \frac{1}{2}h, y + \frac{1}{2}h)]/n}{h}$$

$h\widehat{f}(y) \approx$ estimated chance of Y realizing a value in $(y - \frac{1}{2}h, y + \frac{1}{2}h)$ for small values of h.

Estimator # 1: Relative Frequency Histogram with Equal Class Widths

Let $Y_{(1)} \leq Y_{(2)} \leq \cdots \leq Y_{(n)}$ be the ordered values of the n data values.

Let $t_0 < t_1 < \cdots < t_k$ (how k is selected will be discussed later) be a mesh with equal bin width

$$t_0 = Y_{(1)}, \quad t_k = Y_{(n)}, \quad t_j = t_{j-1} + h, \quad \text{for} \quad j = 1, \ldots, k$$

$$\Rightarrow \quad t_j = t_0 + jh \quad \text{with} \quad h = \frac{Y_{(n)} - Y_{(1)}}{k} = \frac{\text{range}}{k}$$

```
  t_0        t_1         t_2                                   t_k-1      t_k
--|----------|----------|----------------------------------|----------|---------
 Y(1)     Y(1)+h      Y(1)+2h                         Y(1)+(k-1)h    Y(n)
```

$$\text{Let} \quad n_j = \#Y_j's \quad \text{in the interval} \quad [t_{j-1}, t_j) = [t_0 + (j-1)h, t_0 + jh)$$

$$n_j's \quad \text{are the Frequencies}$$

$$\text{Let} \quad R_j = \frac{n_j}{n} = \quad \text{Relative Frequencies}$$

$$\text{Let} \quad \widehat{f}_j = \frac{R_j}{h} = \quad \text{Concentration in} \quad jth \quad \text{interval}$$

This leads to our first estimator of the pdf:

$$\widehat{f}(y) = \begin{cases} \hat{f}_j & \text{if} \;\; t_{j-1} \leq y < t_j \\ 0 & \text{otherwise} \end{cases}$$

$$\text{For} \quad y \epsilon [t_{j-1}, t_j) \quad \widehat{f}(y) = \frac{1}{nh} \sum_{i=1}^{n} I\left(Y_i \epsilon [t_{j-1}, t_j)\right)$$

A plot of n_j vs y or R_j vs y or f_j vs y all yield essentially the same graph (except for the scale on the vertical axis) when the mesh size h is the same for all intervals, $[t_{j-1}, t_j)$. However, only f_j vs y is a “true” density estimator, that is a graph with only non-negative areas and total area under the curve equal to 1.

END LASDNOTES 9/15/21

STAT Class Notes 9/17/21

Estimator # 1A: Relative Frequency Histogram with UnEqual Class Widths

A more general definition allows for unequal mesh sizes:

Let $t_0 < t_1 < \cdots < t_k$ be a mesh with bin widths

$$t_0 = Y_{(1)} - c, t_k = Y_{(n)} + c, \quad \text{with mesh sizes} \quad h_j = t_j - t_{j-1} \quad \text{for} \quad j = 1, \ldots, k$$

```
  t_0        t_1              t_2                 t_k-1                      t_n
 --|---------|---------------|------------------|--------------------------|---------
Y(1)-c   Y(1)-c+h_1      Y(1)-c+h_2          Y(1)-c+h_k-1                Y(n)+c
```

$$\text{Let} \quad n_j = \#Y_i's \quad \text{in the interval} \quad [t_{j-1}, t_j)$$

$$\text{Let} \quad \widehat{f}_j = \frac{n_j}{nh_j} = \quad \text{Concentration in} \quad jth \quad \text{interval}$$

Our estimator of the pdf is:

For $\quad y\epsilon[t_{j-1}, t_j)$,

$$\widehat{f}(y) = \begin{cases} \widehat{f}_j & \text{if} \;\; t_{j-i} \leq y < t_j \\ 0 & \text{otherwise} \end{cases}$$

With unequal interval widths, h_j, the plots of n_j vs y, R_j vs y, and $\widehat{f}_j$ vs y do not yield equivalent graphs.

The plots of n_j vs y and R_j vs y would be a graphical distortion because they would have too much area in those intervals having greater width relative to the concentration of data in those intervals.

The next two pages of plots will illustrate these ideas using the ozone data.

In some of the plots, I have added 6 outliers to the Yonkers Ozone data: 243, 357, 425, 567, 780, 870. The plots were obtained using R.

The R function

hist(y,nclass,breaks,plot=T,prob=T)

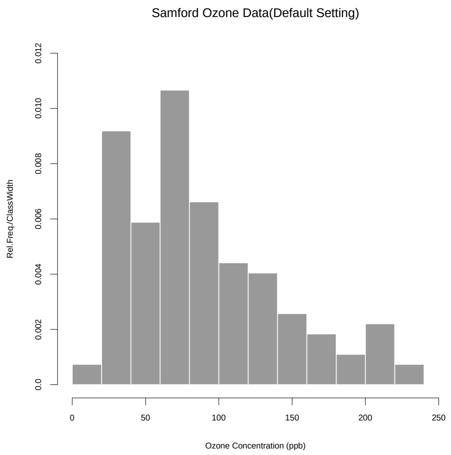
has the following interpretation:

1. y = data vector
2. nclass=number of intervals (default=$1 + log_2(n)$)
3. breaks=$t_j's$ = vector of values for intervals (default=equally spaced)
4. plot=F just produces heights of rectangles, no plot
5. prob=T produces a histogram having vertical axis being the relative frequency/class width. That is, a plot of $\widehat{f_j}$ vs classes
6. prob=F has histogram with vertical axis counts, plot of n_j vs classes

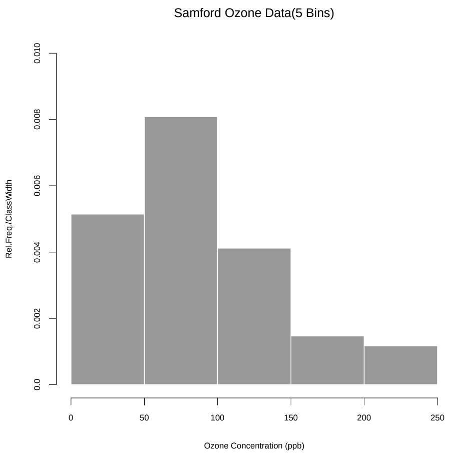
There are a number of suggested ways of selecting the value for h, interval (bin) width and number of bins in the equally spaced case:

1. Most basic statistics books: Let k be 5 to 15 with $h = (Y_{(n)} - Y_{(1)})/k = range/k$
2. Default in R: $h = range/(1 + log_2(n))$
3. Scott's Rule: $h = 3.5\hat{\sigma}n^{-1/3}$, where $\hat{\sigma}$ is the sample standard deviation (specify nclass="scott")
4. Freedman-Diaconis's Rule: $h = 2(IQR)n^{-1/3}$, where $IRQ = \widehat{Q}(.75) - \widehat{Q}(.25)$ (specify nclass="fd")

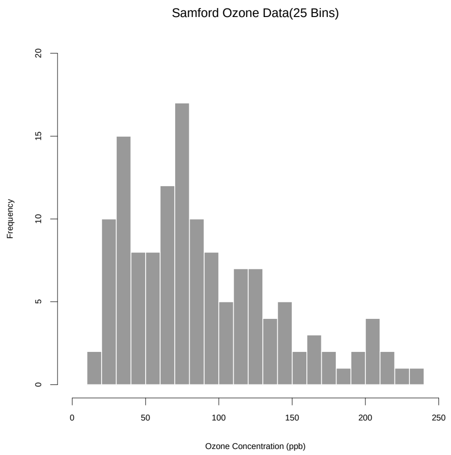
```
#The following R code generates various histograms
#for the Samford Ozone Data, in eCampus,   ozonehistograms.R

          hist(y1,plot=TRUE,prob=T,
          main="Samford Ozone Data(Default Setting)",
          ylab="Rel.Freq./ClassWidth",
          xlab="Ozone Concentration (ppb)",cex=.70,xlim=c(0,250),ylim=c(0,.012))

          hist(y1,nclass=5,plot=TRUE,prob=T,
          main="Samford Ozone Data(5 Bins)",
          ylab="Rel.Freq./ClassWidth",
          xlab="Ozone Concentration (ppb)",cex=.70,xlim=c(0,250),ylim=c(0,.010))

          hist(y1,nclass=25,plot=TRUE,
          main="Samford Ozone Data(25 Bins)",
          ylab="Frequency",
          xlab="Ozone Concentration (ppb)",cex=.70,xlim=c(0,250),ylim=c(0,20))

          hist(y1,plot=TRUE,prob=T,
          main="Samford Ozone Data(Default Setting)",
          ylab="Rel.Freq./ClassWidth",
          xlab="Ozone Concentration (ppb)",cex=.70,xlim=c(0,250),ylim=c(0,.012))

#The following R code generates various histograms
#for the Yonker's Ozone Data (with and without outliers)


          y2p  =  c(y2,243,357,425,567,780,870)

          hist(y2,plot=TRUE,prob=T,
          main="Yonkers Ozone(Default Setting)",
          ylab="Rel.Freq./ClassWidth",
          xlab="Ozone Concentration (ppb)",cex=.75,xlim=c(0,150),ylim=c(0,.02))

          hist(y2p,plot=TRUE,prob=T,
          main="Yonkers Ozone With Outliers(Default Setting)",
          ylab="Rel.Freq./ClassWidth",
          xlab="Ozone Concentration (ppb)",cex=.75,xlim=c(0,1000),ylim=c(0,.010))

          breaks2  =  seq(0,140,20)
          breaks2  =  c(breaks2,500,1000)


          hist(y2p,breaks=breaks2,plot=TRUE,lab=c(6,10,7),
          main="Yonkers Ozone With Outliers(Unequal Class Widths)",
          ylab="Frequency",
          xlab="Ozone Concentration (ppb)",cex=.75,xlim=c(0,1000),ylim=c(0,45))


          hist(y2p,breaks=breaks2,prob=T,plot=TRUE,lab=c(6,8,7),
          main="Yonkers Ozone With Outliers(Unequal Class Widths)",
          ylab="Rel.Freq./ClassWidth",
          xlab="Ozone Concentration (ppb)",cex=.75,xlim=c(0,1000),ylim=c(0,.015))
```



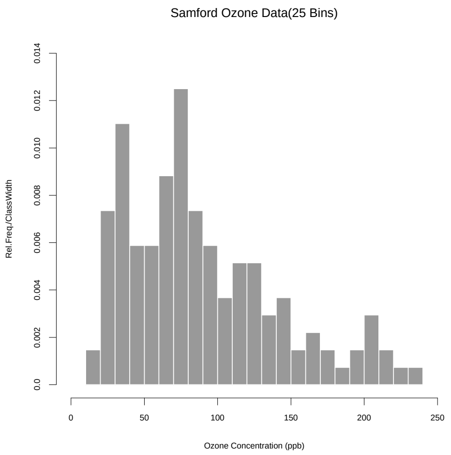
Samford Ozone Data(Default Setting)



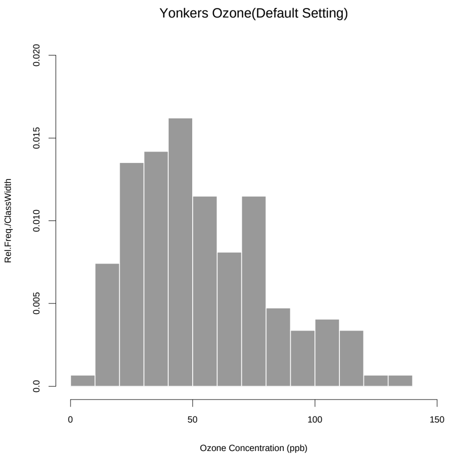
Samford Ozone Data(5 Bins)

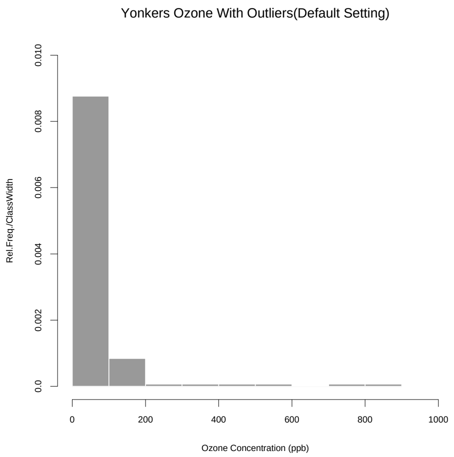


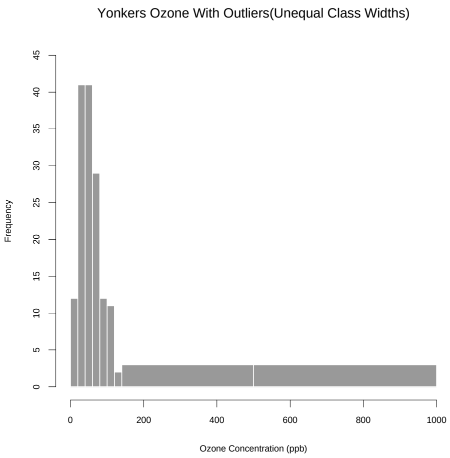
Samford Ozone Data(25 Bins)

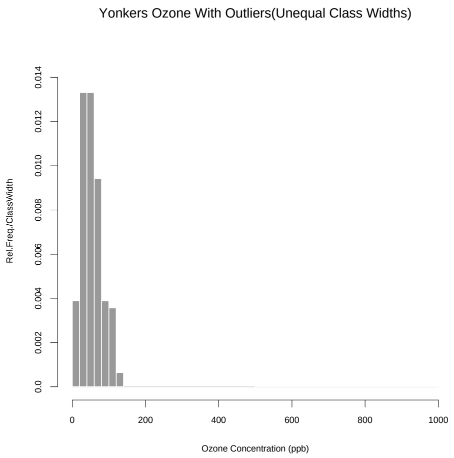


Samford Ozone Data(25 Bins)


Yonkers Ozone(Default Setting)
Rel.Freq./ClassWidth
0.0
0.005
0.010
0.015
0.020
0
50
100
150
Ozone Concentration (ppb)


Yonkers Ozone With Outliers(Default Setting)
Rel.Freq./ClassWidth
0.0
0.002
0.004
0.006
0.008
0.010
0
200
400
600
800
1000
Ozone Concentration (ppb)


Yonkers Ozone With Outliers(Unequal Class Widths)
Frequency
0
5
10
15
20
25
30
35
40
45
0
200
400
600
800
1000
Ozone Concentration (ppb)


Yonkers Ozone With Outliers(Unequal Class Widths)
Rel.Freq./ClassWidth
0.0
0.002
0.004
0.006
0.008
0.010
0.012
0.014
0
200
400
600
800
1000
Ozone Concentration (ppb)

There are several major problems with using the relative frequency histogram as an estimator of the pdf.

- First, the relative frequency histogram only measures the local density at the midpoint of each of the bins: $[t_{j-1}, t_j)$. This local density is then assigned to ALL $y's$ in the interval $[t_{j-1}, t_j)$. Thus, $\widehat{f}(\cdot)$ is a piecewise constant function. This is not a very realistic portrayal of a continuous function.

- Also, the relative frequency histogram uses only the data within the interval $[t_{j-1}, t_j)$ containing y to estimate $f(y)$. All the other data is ignored. To overcome many of these problems, we will now discuss the kernel density estimator.

Estimator # 2: Kernel Density Estimator

The are a number of articles and books written on the estimation of the density function. Two excellent sources are

An article by Simon Sheather in 2004, **Density Estimation**, *Statistical Science*, Vol. 19, No. 4, pp. 588-597.

The book by David Scott published in 1992, *Multivariate Density Estimation: Theory, Practice, and Visualization.* Wiley, New York.

Most of the following has been selected directly from Dr. Sheather's paper. I have posted Dr. Sheather's article in eCampus - Lecture Notes.

Let $Y_1, Y_2, \ldots, Y_n$ denote a random sample of size n from a random variable, Y, with pdf f.

The kernel density estimate of $f(y)$ is given by

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

where the kernel K is a non-negative function which satisfies $\int K(x)dx = 1$ and the smoothing parameter h is known as the bandwidth. In practice, the kernel K is generally chosen to be a unimodal pdf symmetric about zero. In this case, K satisfies the conditions:

$$\int_{-\infty}^{\infty} K(x)dx = 1, \qquad \int_{-\infty}^{\infty} xK(x)dx = 0, \qquad \int_{-\infty}^{\infty} x^2K(x)dx = \mu_2(K) > 0.$$

Note: $\widehat{f}$ is a valid pdf:

$\widehat{f}(y) \geq 0$ for all y because $K(x) \geq 0$ for all x

$$\begin{aligned}\int_{-\infty}^{\infty} \widehat{f}(y)dy &= \int_{-\infty}^{\infty} \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{y - Y_i}{h}\right) dy \\ &= \frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{\infty} K\left(u\right) du \\ &= 1\end{aligned}$$

A popular choice for K is the Gaussian kernel, namely,

$$K(x) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{x^2}{2}\right).$$

There are many other choices for kernels. A number of these kernel functions are given below. These figures are from *Graphical Methods for Data Analysis*, by J. Chambers, W. Cleveland, B. Kleiner, and P. Tukey.

Kernel density estimator

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

Kernel	$K(u)$
Uniform	$\frac{1}{2} I(\lvert u\rvert \le 1)$
Triangle	$(1 - \lvert u\rvert) I(\lvert u\rvert \le 1)$
Epanechnikov	$\frac{3}{4}(1 - u^2) I(\lvert u\rvert \le 1)$
Quartic	$\frac{15}{16}(1 - u^2)^2 I(\lvert u\rvert \le 1)$
Triweight	$\frac{35}{32}(1 - u^2)^3 I(\lvert u\rvert \le 1)$
Gaussian	$\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}u^2)$
Cosinus	$\frac{\pi}{4} \cos(\frac{\pi}{2}u) I(\lvert u\rvert \le 1)$

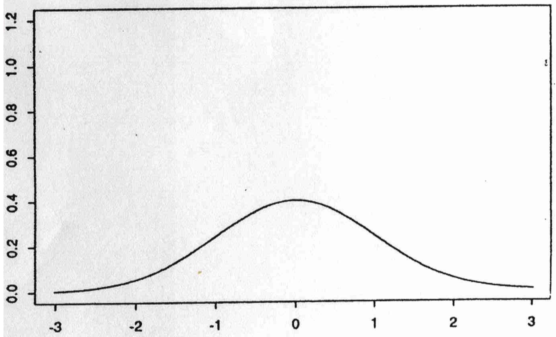


Figure 2.5 : Gaussian kernel.

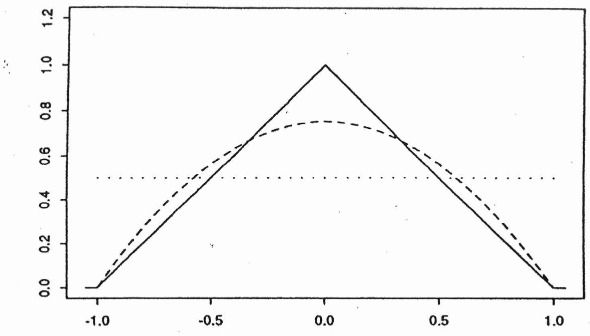


Figure 2.3 : Uniform kernel (points), Triangle kernel (solid line), and Epanechnikov kernel (dashed line).

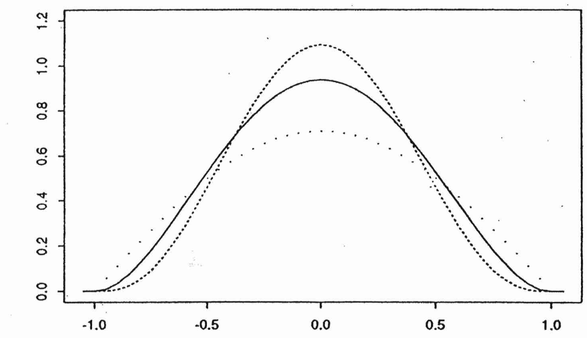


Figure 2.4 : Quartic kernel (solid line), Triweight kernel (dashed line), and Cosinus kernel (points).

In the article by Dr. Sheather, the computation of the kernel density estimator is illustrated using a data set with 10 observations. The data consist of a simulated sample of size $n = 10$ from a normal mixture distribution made up of observations from $N(-1, (1/3)^2)$ and $N(1, (1/3)^2)$, each with a probability of 0.5. Figure 1 on the next page shows a kernel estimate of the density for these data using the Gaussian kernel with a bandwidth of $h = 0.3$ (the dashed curve) along with the true underlying density (the solid curve). The 10 data points are marked by vertical lines on the horizontal axis.

Each data point's contribution to the overall density estimator, $\widehat{f}(y)$, is the sum of the intersection of a vertical line through y with each of the 10 Gaussian curves which are centered at $Y_1,\ Y_1,\ \ldots,\ Y_{10},$.

That is,

$$\frac{1}{nh} K\left(\frac{y - Y_i}{h}\right) \quad \text{for } i = 1, 2, \ldots, 10$$

$$\frac{1}{(10)(.3)} K\left(\frac{y - Y_1}{.3}\right),\ \frac{1}{(10)(.3)} K\left(\frac{y - Y_2}{.3}\right),\ \ldots,\ \frac{1}{(10)(.3)} K\left(\frac{y - Y_{10}}{.3}\right),$$

where $K(u) = \frac{1}{\sqrt{2\pi}} exp\left(-\frac{u^2}{2}\right)$.

The density estimate at the argument y, $\widehat{f}(y)$, (the dashed curve) is the sum of these scaled normal densities:

$$\widehat{f}(y) = \sum_{i=1}^{10} \frac{1}{(10)(.3)} K\left(\frac{y - Y_i}{.3}\right).$$

If the value of h was increased, each of the 10 normal curves would widen and hence smooth out the modes currently apparent in the density estimate.

h ↑ - smoother desity estimate

Important ?

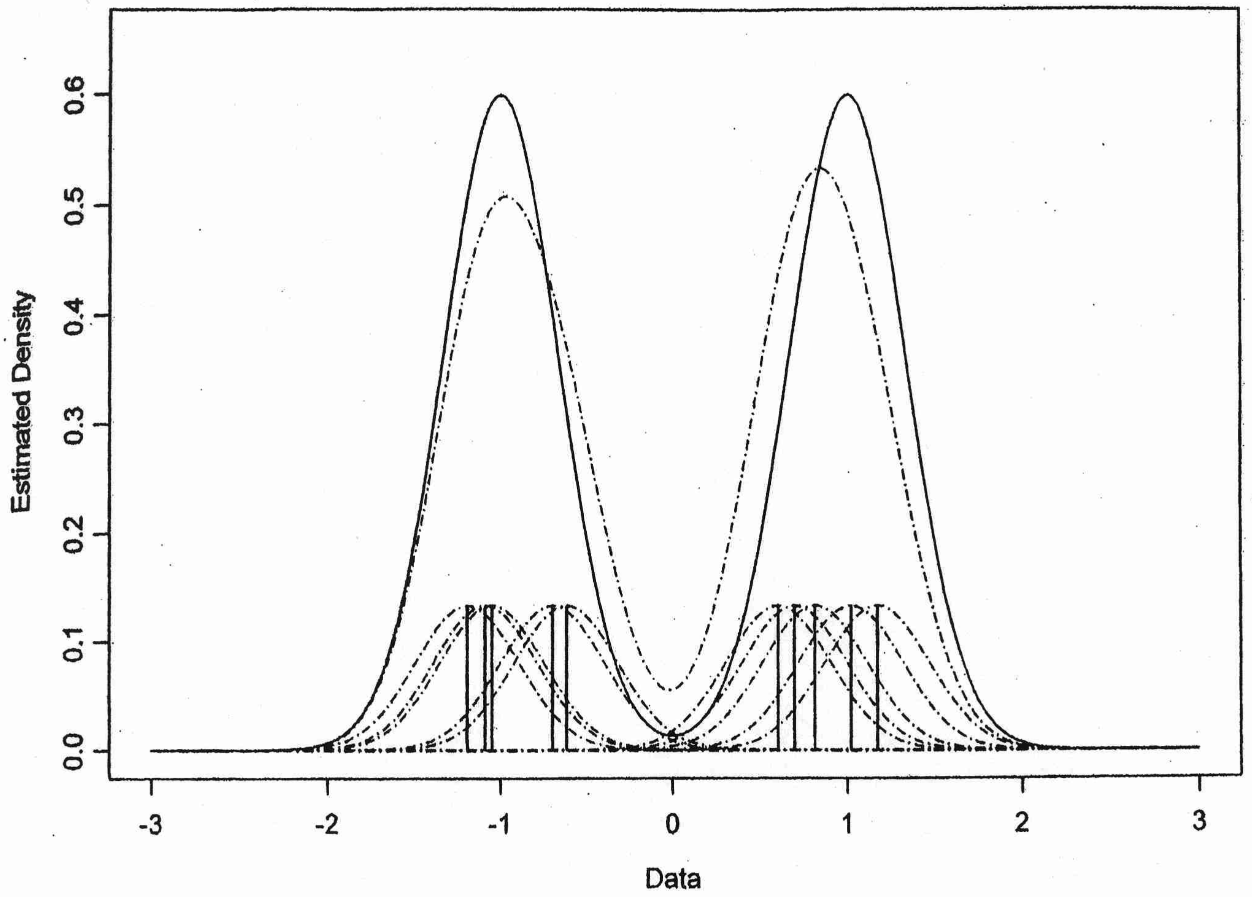
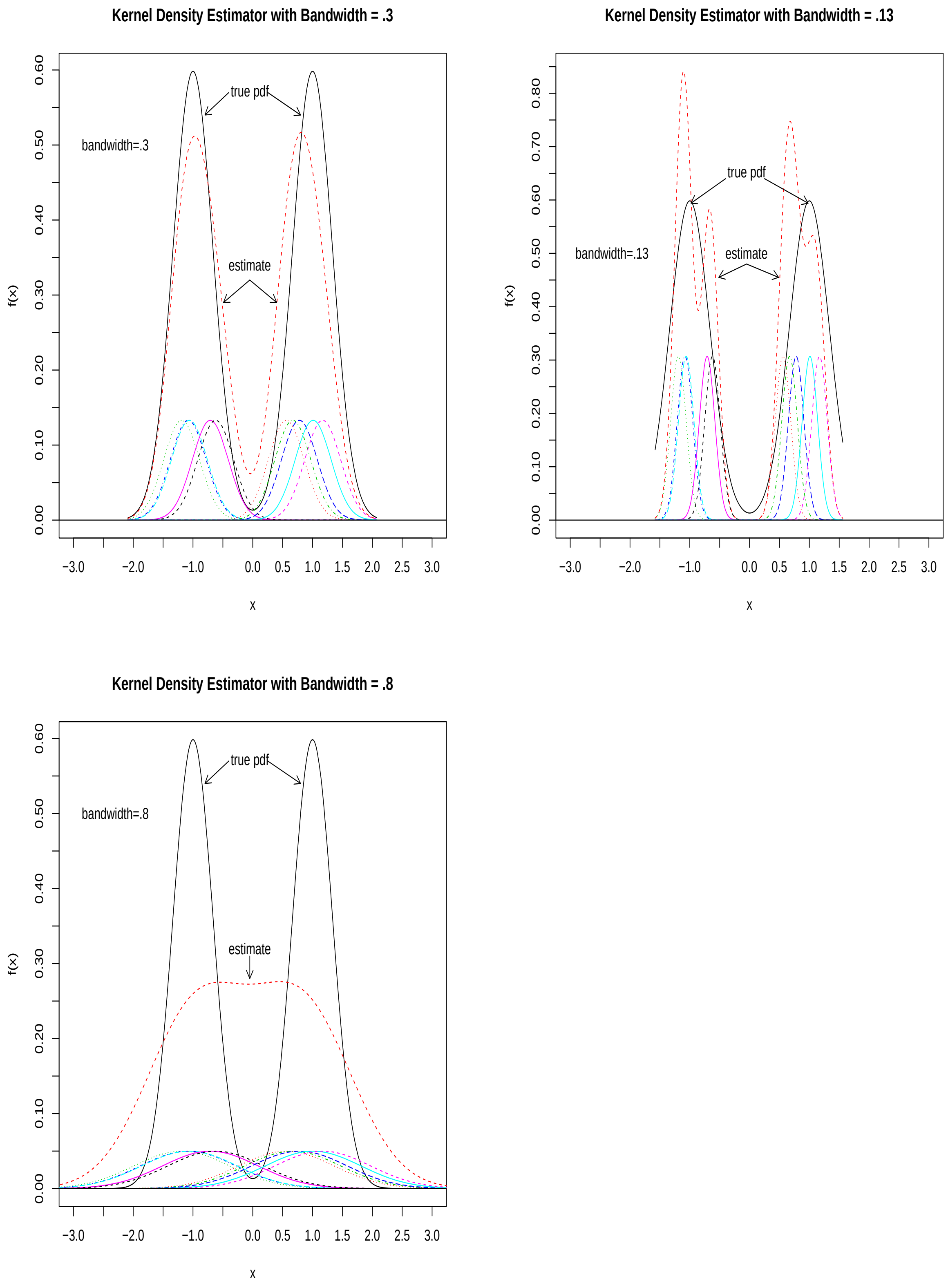


FIG. 1. *Kernel density estimate and contributions from each data point (dashed curve) along with the true underlying density (solid curve).*

Three graphs illustrating the impact of the size of the bandwidth on the estimate of the pdf.



Example:

The following calculations will yield the values of $\frac{1}{nh}K\left(\frac{y-Y_i}{h}\right)$

the contribution of each of the data values Y_i to the estimate of $\widehat{f}(y)$ at $y=-0.8$ and at $y=0.5$

In these calculations $n=10$, $h=.3$, and the kernel is the Gaussian kernel

$$K(u)=\frac{1}{\sqrt{2\pi}}\;e^{-.5u^2}$$

$$\widehat{f}(-0.8)=\sum_{i=1}^{n}\frac{1}{nh}K\left(\frac{-0.8-Y_i}{h}\right)=\sum_{i=1}^{10}\frac{1}{(10)(.3)}\frac{1}{\sqrt{2\pi}}\;e^{-.5\left(\frac{-0.8-Y_i}{.3}\right)^2}$$

and

$$\widehat{f}(0.5)=\sum_{i=1}^{n}\frac{1}{nh}K\left(\frac{0.5-Y_i}{h}\right)=\sum_{i=1}^{10}\frac{1}{(10)(.3)}\frac{1}{\sqrt{2\pi}}\;e^{-.5\left(\frac{.5-Y_i}{.3}\right)^2}$$

where Y_i are the 10 data values which were used to obtain the estimator $\widehat{f}(y)$ depicted in the graph on the previous page.

i	Y_i	$\frac{1}{(10)(.3)}K\left(\frac{-0.8-Y_i}{.3}\right)$	$\frac{1}{3}K\left(\frac{0.5-Y_i}{.3}\right)$
1	-1.19	0.057123	0.000000
2	-1.08	0.086026	0.000000
3	-1.06	0.091345	0.000000
4	-0.71	0.127129	0.000039
5	-0.62	0.111075	0.000125
6	0.56	0.000005	0.130348
7	0.67	0.000001	0.113256
8	0.78	0.000000	0.086026
9	1.01	0.000000	0.031350
10	1.17	0.000000	0.010983
Sum		0.472704	0.372126

From the above table we have

$$\widehat{f}(-0.8)=\sum_{i=1}^{10}\frac{1}{(10)(.3)}\frac{1}{\sqrt{2\pi}}\;e^{-.5\left(\frac{-0.8-Y_i}{.3}\right)^2}=0.472704$$

$$\widehat{f}(0.5)=\sum_{i=1}^{10}\frac{1}{(10)(.3)}\frac{1}{\sqrt{2\pi}}\;e^{-.5\left(\frac{.5-Y_i}{.3}\right)^2}=.372126$$

The farther the data value Y_i is from y, the smaller its contribution to $\widehat{f}(y)$.

We can not compute the kernel density estimator $\widehat{f}(y)$ for all possible values of the random variable Y (uncountably many). Therefore, we select m plotting points $x_1, x_2, \ldots, x_m$, evaluate the kernel density estimator at each of these points, and obtain

$$\widehat{f}(x_1) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x_1 - Y_i}{h}\right)$$

$$\widehat{f}(x_2) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x_2 - Y_i}{h}\right)$$

$$\vdots$$

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

A smooth curve is then passed through the points, $\widehat{f}(x_1), \widehat{f}(x_2), \ldots, \widehat{f}(x_m)$ to obtain the estimator $\widehat{f}(\cdot)$.

Thus, in order to obtain a kernel density estimator, we must make the following selections:

1. Sample Size n
2. Kernel (Gaussian is popular choice)
3. the number of plotting points m (enough so that curve is relatively smooth, $m \geq 100$ is generally enough)
4. the Bandwidth h (considerable research on how to select h)

The selection of the bandwidth h is a compromise between smoothing enough to remove insignificant bumps and not smoothing too much so as to smear out real peaks in the density. Mathematically, the selection of h is a compromise between

1. The bias in $\widehat{f}(x)$ as an estimator of $f(x)$: Bias$\{\hat{f}(x)\}$ (bias increases with increasing h) and
2. the variance of $\widehat{f}(x) : Var\{\widehat{f}(x)\}$, (variance decreases with increasing h)

Assuming that the underlying pdf $f(\cdot)$ is sufficiently smooth and that the kernel has finite fourth moment, it can be shown using Taylor series that

$$\text{Bias}\{\widehat{f}(x)\} = E[\widehat{f}(x)] - f(x) = \frac{h^2}{2}\mu_2(K)f''(x) + o(h^2),$$

$$\text{Var}\{\widehat{f}(x)\} = \frac{1}{nh}R(K)f(x) + o\left(\frac{1}{nh}\right),$$

where

$$R(K) = \int K^2(y)dy.$$

A widely used choice of an overall measure of the difference between $\widehat{f}$ and f is the mean integrated squared error (MISE), which is given by

$$\begin{aligned} MISE(\widehat{f}) &= E\left\{\int (\widehat{f}(y) - f(y))^2 dy\right\} \\ &= \int Bias(\widehat{f}(y))^2 dy + \int Var(\widehat{f}(y))dy \end{aligned}$$

Under further conditions on f, the asymptotic mean integrated squared error (AIMSE) is given by

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

where

$$R(f'') = \int [f''(y)]^2 dy.$$

In addition to the visual advantage of being a smooth curve, the kernel density estimator has an advantage over the relative frequency histogram in terms of the AMISE of the two estimators. It can be shown that the AMISE of a relatively frequency histogram converges much slower to 0 with increasing sample size n than the AMISE of the kernel density estimator.

The value of the bandwidth h which minimizes the AIMSE is given by

$$h_{AIMSE} = \left[\frac{R(K)}{\mu_2(K)^2 R(f'')}\right]^{1/5} n^{-1/5}$$

When the pdf f is sufficiently smooth, we can show that

$$R(f^{''}) = -\int f^{(4)}(y)f(y)dy,$$

Thus, the functional $R(f^{''})$ is a measure of the roughness or curvature in f. In particular, the larger the value of $R(f^{''})$ is, the larger the value of AIMSE and hence the more difficult it is to estimate f. Therefore, the smaller the value of h_{AIMSE}, that is, the smaller the bandwidth needed to capture the curvature in f.

There is a lengthy discussion on how to select the bandwidth in any given situation in Dr. Sheather's paper. This is still an active research area. A short summary of some Rules of Thumb from Dr. Sheather's paper will now be given.

A computationally simple method for selecting the bandwidth h is based on replacing $R(f^{''})$, the unknown part of h_{AIMSE}, by its value for a parametric family expressed as a multiple of a scale parameter, which is then estimated from the data. This method was developed in Deheuvels(1977), Scott(1979), and Silverman(1986, Section 3.2), who used the normal distribution as the parametric family.

Let σ and IQR denote the standard deviation and interquartile range of X, respectively. Take the Kernel K to be the usual Gaussian kernel. Assuming that the underlying distribution is normal, Silverman(1986) showed that bandwidth h reduces to

$$h_{AMISE_{Normal}} = 1.06\sigma n^{-1/5}$$

$$h_{AMISE_{Normal}} = 0.79(IQR)n^{-1/5}.$$

Jones, Marron, and Sheather(1996) studied the performance of

$$h_{SNR} = 1.06Sn^{-1/5},$$

where S is the sample standard deviation. They found the h_{SNR} had a mean that was unacceptably large and thus often produced oversmoothed density estimates. Furthermore, Silverman(1986) recommended reducing the factor 1.06 to 0.9 in an attempt not to miss bimodality. He further suggested using the smaller of two scale estimates.

$$h_{SROT} = 0.9An^{-1/5} \text{ with A} = \min\{\text{S, (sample IQR)/1.34}\}.$$

In R, Silverman's bandwidth is invoked by width="nrd". A couple of other methods for determining the appropriate bandwidth are Cross-Validation Methods and Plug-in Methods.

The article by Dr. Sheather and the article, "A brief survey of bandwidth selection", *JASA*, 91 (1996), pp. 401-407 with authors Jones, Marron, and Dr. Sheather, provide details on how to select the bandwidth. This article is also in eCampus under LectureNotes.

The R function for obtaining the kernel density estimator based on a random sample of n observations is

density(y,m=?,kernel="?",bw=?,from=?,to=?), where

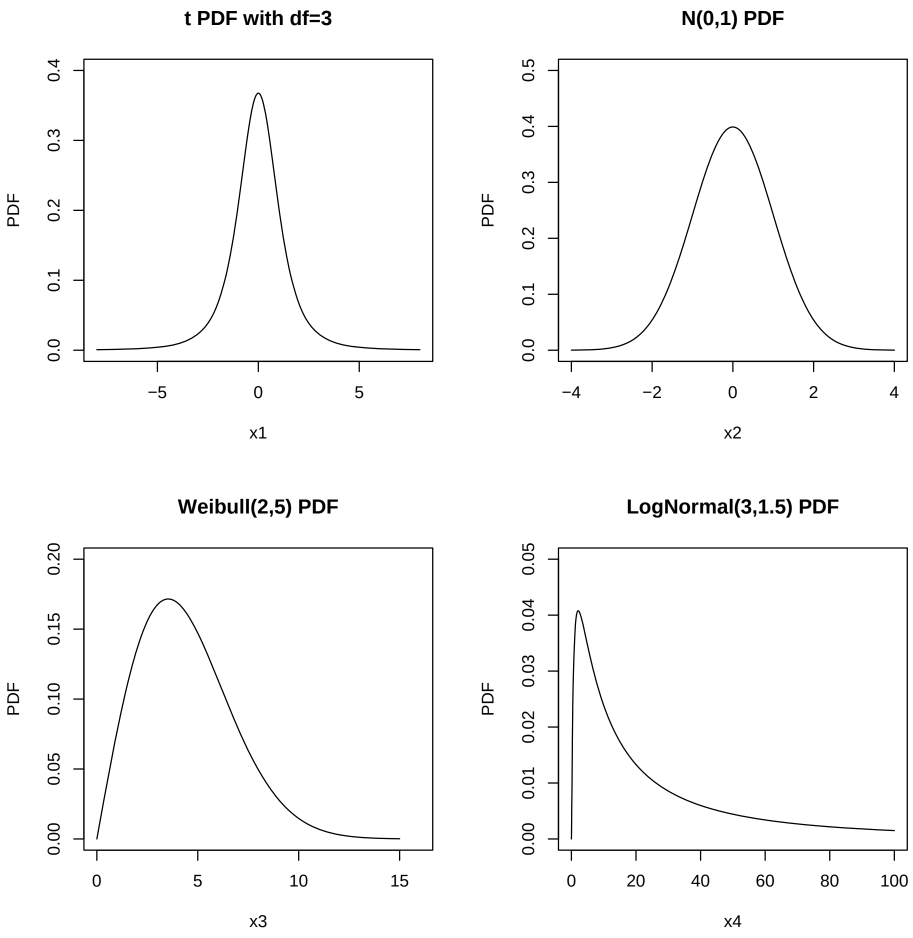
1. **y**: the data vector of n observations from Y
2. **m**: number of equally spaced points at which $\widehat{f}$ will be evaluated, m=50 is the default (generally too few values)
3. **kernel**: kernel function; g=Gaussian, r=rectangular, t=triangular, c=cosine (default is Gaussian kernel)
4. **bw**: bandwidth, the default is Silverman's nrd
5. **from**: minimum value of the equally spaced points (default is $Y_{(1)} - \frac{3}{4}h$)
6. **to**: maximum value of the equally spaced points (default is $Y_{(n)} + \frac{3}{4}h$)

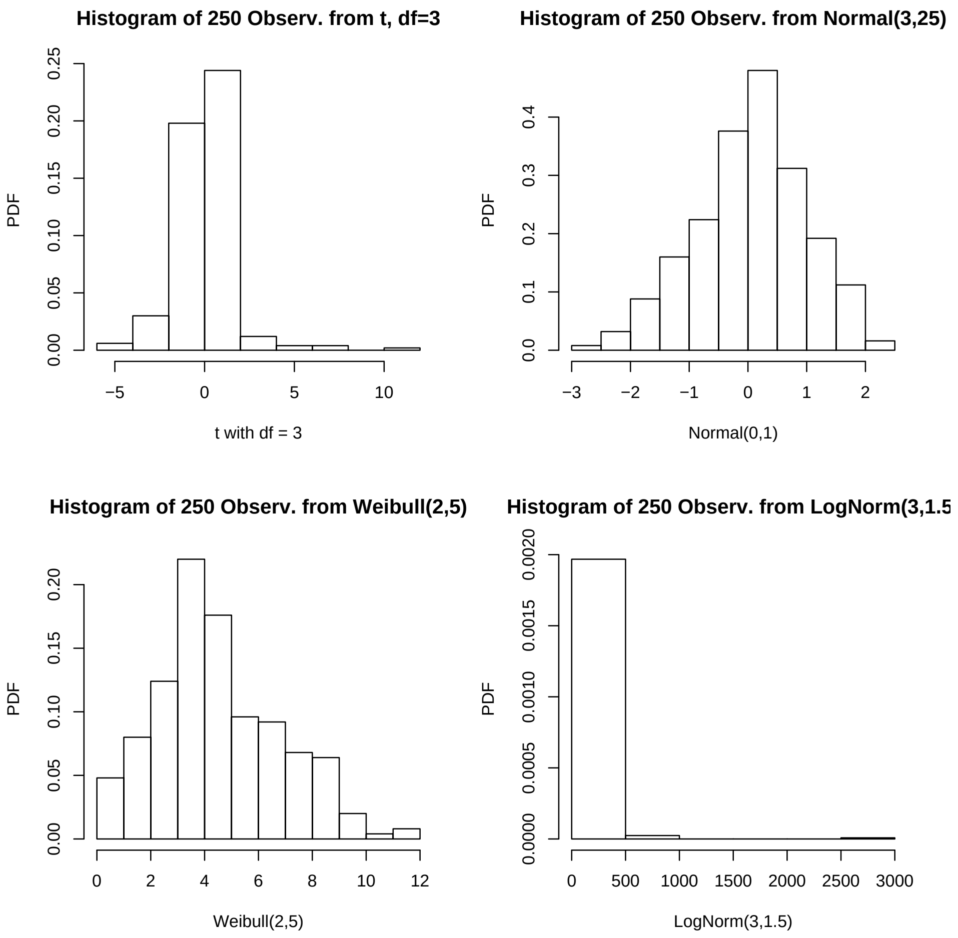
The graphs on the following pages will be used to illustrate the choice of bandwidth and kernel on the kernel density estimator.

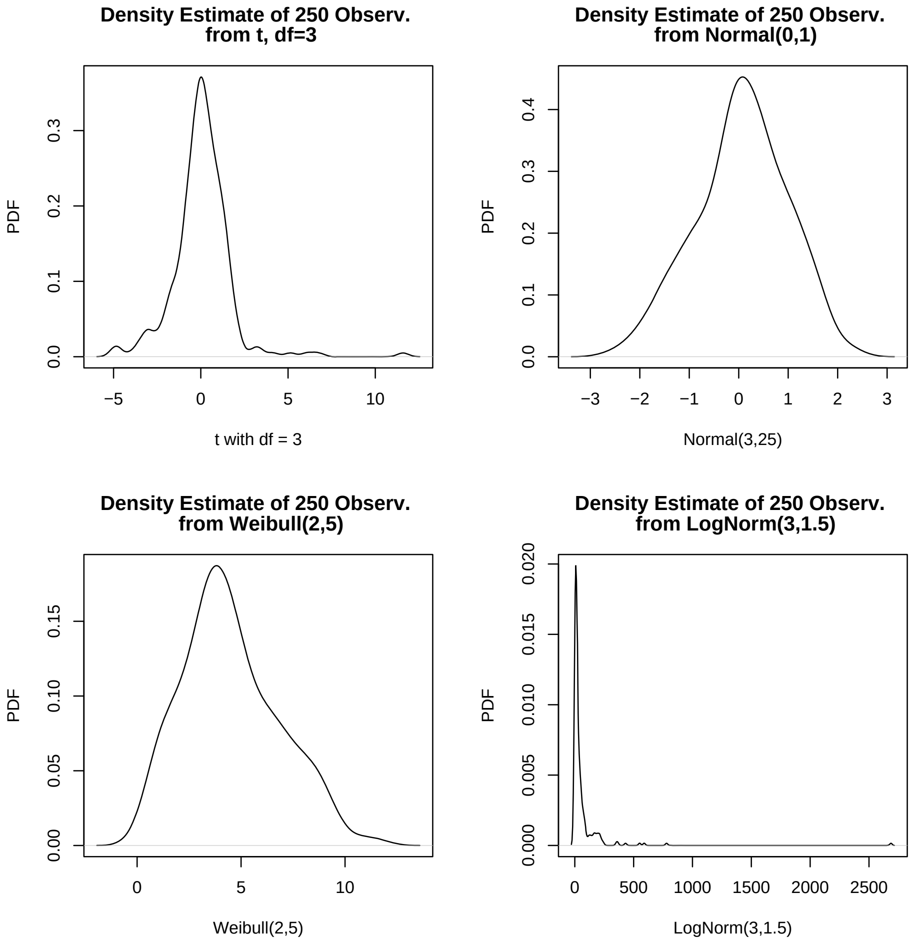
$$plot(density(y, m = 100, kernel = "g", bw = "nrd", from = -6, to = 100)$$

```
#The following R code (distden.R) generates data from various specified distributions
#and the plots the empirical density function of the generated data.
#-------------------------------------------------------------------------------

x1  =  seq(-8,8,length=5000)
x2  =  seq(-4,4,length=5000)
x3  =  seq(0,15,length=5000)
x4  =  seq(0,100,length=5000)

y1  =  dt(x1,3)
y2  =  dnorm(x2,0,1)
y3  =  dweibull(x3,2,5)
y4  =  dlnorm(x4,3,1.5)

#postscript("distpdf.ps",height=8,horizontal=F)

par(mfrow=c(2,2))
plot(x1,y1,main="t PDF with df=3",ylab="PDF",type="l",ylim=c(0,.4),
     xlim=c(-8,8),lab=c(9,8,7),cex=.5)
plot(x2,y2,main="N(0,1) PDF",ylab="PDF",type="l",ylim=c(0,.5),
     xlim=c(-4,4),lab=c(9,10,7),cex=.5)
plot(x3,y3,main="Weibull(2,5) PDF",ylab="PDF",type="l",ylim=c(0,.2),
     xlim=c(0,16),lab=c(8,8,7),cex=.5)
plot(x4,y4,main="LogNormal(3,1.5) PDF",ylab="PDF",type="l",ylim=c(0,.05),
     xlim=c(0,100),lab=c(10,10,7),cex=.5)

#generates 250 observations from t with df=3, normal(0,1)
#Weibull with scale=5 and shape=2,
#LogNormal with logmean=3 and logsd=1.5

t3  =  rt(250,3)
norm  =  rnorm(250,0,1)

wei  =  rweibull(250,2,5)

lnrm  = rlnorm(250,3,1.5)

#The following commands will generate the relative frequency histograms:

#postscript("disthist.ps",height=8,horizontal=F)

par(mfrow=c(2,2))
# Histogram of 250 observations from t with df= 3:

 hist(t3,plot=TRUE,prob=T,
         main="Histogram of 250 Observ. from t, df=3",ylab="PDF",
         xlab="t with df = 3",cex=.50)

# Histogram of 250 observations from Normal:

 hist(norm,plot=TRUE,prob=T,
         main="Histogram of 250 Observ. from Normal(3,25)",ylab="PDF",
         xlab="Normal(0,1)",cex=.50)
```

```
# Histogram of 250 observations from Weibull:

 hist(wei,plot=TRUE,prob=T,
         main="Histogram of 250 Observ. from Weibull(2,5)",ylab="PDF",
         xlab="Weibull(2,5)",cex=.50)

# Histogram of 250 observations from LogNormal:

 hist(lnrm,plot=TRUE,prob=T,
         main="Histogram of 250 Observ. from LogNorm(3,1.5)",ylab="PDF",
         xlab="LogNorm(3,1.5)",cex=.50)

#graphics.off()


#The following commands will generate the nonparametric density estimates:

#postscript("distden.ps",height=8,horizontal=F)
par(mfrow=c(2,2))

# Density of 250 observations from t with df= 3:


 plot(density(t3,bw="nrd"),type="l",
         main="Density Estimate of 250 Observ. \n from t, df=3",ylab="PDF",
         xlab="t with df = 3",cex=.50)

# Density Estimate of 250 observations from Normal:

 plot(density(norm,bw="nrd"),type="l", ylab="PDF",
         main="Density Estimate of 250 Observ. \n from Normal(0,1)",
         xlab="Normal(3,25)",cex=.50)

# Density Estimate of 250 observations from Weibull:

 plot(density(wei,bw="nrd"),type="l", ylab="PDF",
         main="Density Estimate of 250 Observ. \n from Weibull(2,5)",
         xlab="Weibull(2,5)",cex=.50)

# Density Estimate of 250 observations from LogNormal:

 plot(density(lnrm,bw="nrd"),type="l", ylab="PDF",
         main="Density Estimate of 250 Observ. \n from LogNorm(3,1.5)",
         xlab="LogNorm(3,1.5)",cex=.50)

graphics.off()
```


t PDF with df=3
PDF
0.0
0.1
0.2
0.3
0.4
−5
0
5
x1
N(0,1) PDF
PDF
0.0
0.1
0.2
0.3
0.4
0.5
−4
−2
0
2
4
x2
Weibull(2,5) PDF
PDF
0.00
0.05
0.10
0.15
0.20
0
5
10
15
x3
LogNormal(3,1.5) PDF
PDF
0.00
0.01
0.02
0.03
0.04
0.05
0
20
40
60
80
100
x4


Histogram of 250 Observ. from t, df=3
PDF
0.00
0.05
0.10
0.15
0.20
0.25
−5
0
5
10
t with df = 3
Histogram of 250 Observ. from Normal(3,25)
PDF
0.0
0.1
0.2
0.3
0.4
−3
−2
−1
0
1
2
Normal(0,1)
Histogram of 250 Observ. from Weibull(2,5)
PDF
0.00
0.05
0.10
0.15
0.20
0
2
4
6
8
10
12
Weibull(2,5)
Histogram of 250 Observ. from LogNorm(3,1.5
PDF
0.0000
0.0005
0.0010
0.0015
0.0020
0
500
1000
1500
2000
2500
3000
LogNorm(3,1.5)



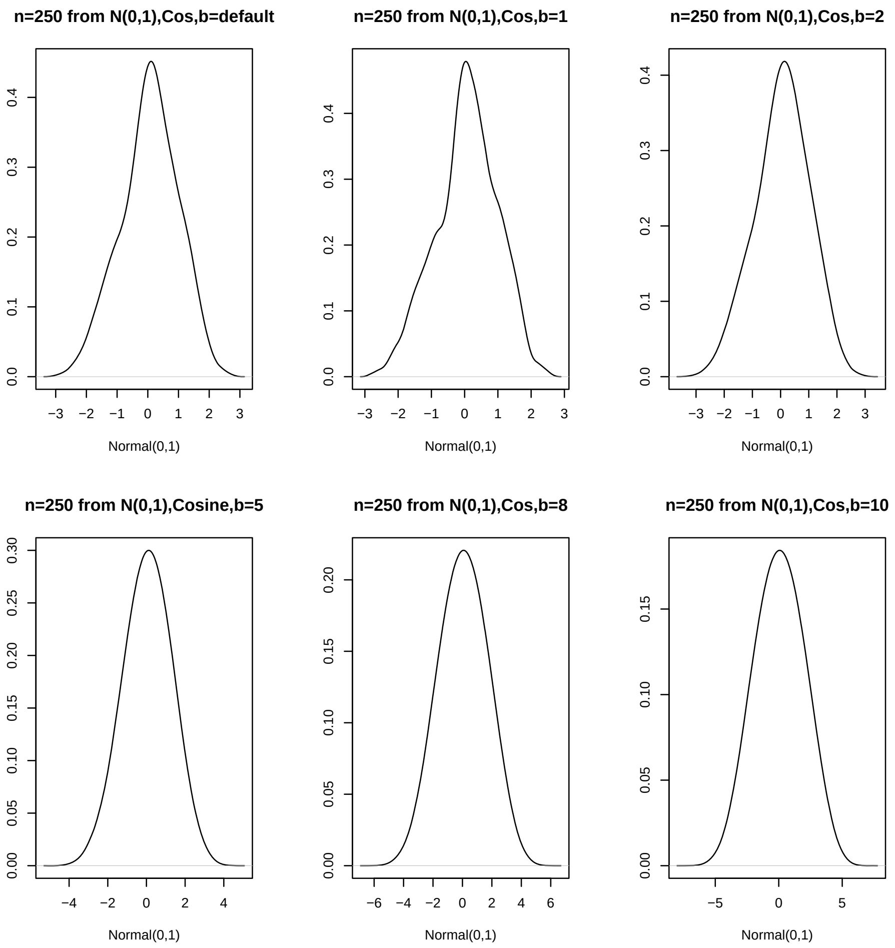
Density Estimate of 250 Observ. from t, df=3

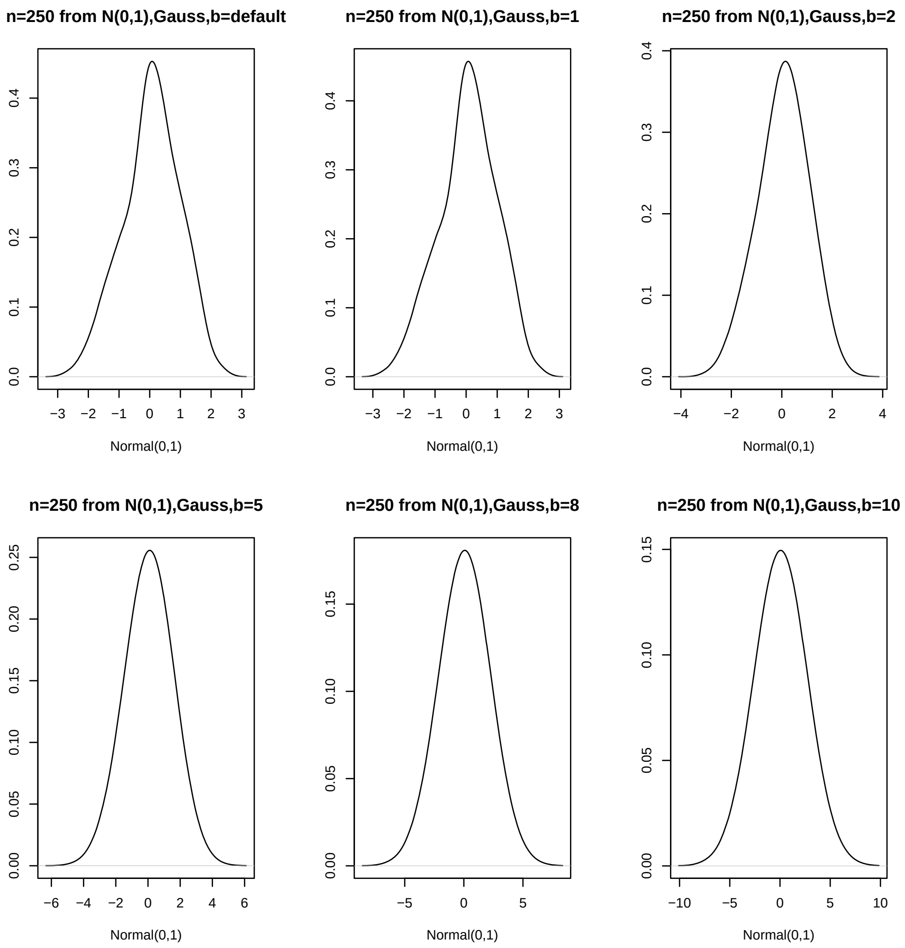
Density Estimate of 250 Observ. from Normal(0,1)

Density Estimate of 250 Observ. from Weibull(2,5)

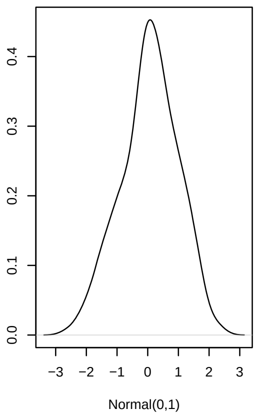
Density Estimate of 250 Observ. from LogNorm(3,1.5)

```
#The following R commands will yield nonparametric density estimates
#for the 250 data values from a N(0,1) distribution:

postscript("normden.ps",height=8,horizontal=F)
par(mfrow=c(2,3))

# Density Estimate of 250 observations from Normal with Cosine Kernel:


 plot(density(norm,window=’cosine’),type="l",
         main="n=250 from N(0,1),Cos,b=default",
         xlab="Normal(0,1)",ylab=" ",cex=.5)

 plot(density(norm,window=’cosine’,width=1),type="l",
         main="n=250 from N(0,1),Cos,b=1",
         xlab="Normal(0,1)",ylab=" ",cex=.5)

 plot(density(norm,window=’cosine’,width=2),type="l",
         main="n=250 from N(0,1),Cos,b=2",
         xlab="Normal(0,1)",ylab=" ",cex=.5)

 plot(density(norm,window=’cosine’,width=5),type="l",
         main="n=250 from N(0,1),Cosine,b=5",
         xlab="Normal(0,1)",ylab=" ",cex=.5)

 plot(density(norm,window=’cosine’,width=8),type="l",
         main="n=250 from N(0,1),Cos,b=8",
         xlab="Normal(0,1)",ylab=" ",cex=.5)

 plot(density(norm,window=’cosine’,width=10),type="l",
         main="n=250 from N(0,1),Cos,b=10",
         xlab="Normal(0,1)",ylab=" ",cex=.5)



# Density Estimate of 250 observations from Normal with Gaussian Kernel:



 plot(density(norm,window=’g’),type="l",
         main="n=250 from N(0,1),Gauss,b=default",
         xlab="Normal(0,1)",ylab=" ",cex=.5)

 plot(density(norm,window=’g’,width=1),type="l",
         main="n=250 from N(0,1),Gauss,b=1",
         xlab="Normal(0,1)",ylab=" ",cex=.5)

 plot(density(norm,window=’g’,width=2),type="l",
         main="n=250 from N(0,1),Gauss,b=2",
         xlab="Normal(0,1)",ylab=" ",cex=.5)

 plot(density(norm,window=’g’,width=5),type="l",
         main="n=250 from N(0,1),Gauss,b=5",
         xlab="Normal(0,1)",ylab=" ",cex=.5)

 plot(density(norm,window=’g’,width=8),type="l",
         main="n=250 from N(0,1),Gauss,b=8",
         xlab="Normal(0,1)",ylab=" ",cex=.5)

 plot(density(norm,window=’g’,width=10),type="l",
         main="n=250 from N(0,1),Gauss,b=10",
         xlab="Normal(0,1)",ylab=" ",cex=.5)



# Density Estimate of 250 observations from Normal with Various Kernel:



 plot(density(norm),type="l",
         main="n=250 from N(0,1),Default Settings",
         xlab="Normal(0,1)",ylab=" ",cex=.5)

 plot(density(norm,window=’cosine’,width=6),type="l",
         main="n=250 from N(0,1),Cosine,b=6",
         xlab="Normal(0,1)",ylab=" ",cex=.5)

 plot(density(norm,window=’g’,width=6),type="l",
         main="n=250 from N(0,1),Gauss,b=6",
         xlab="Normal(0,1)",ylab=" ",cex=.5)

 plot(density(norm,window=’rectangular’,width=6),type="l",
         main="n=250 from N(0,1),Rectangular,b=6",
         xlab="Normal(0,1)",ylab=" ",cex=.5)


 plot(density(norm,window=’triangular’,width=6),type="l",
         main="n=250 from N(0,1),Triangle,b=6",
         xlab="Normal(0,1)",ylab=" ",cex=.5)
graphics.off()
```

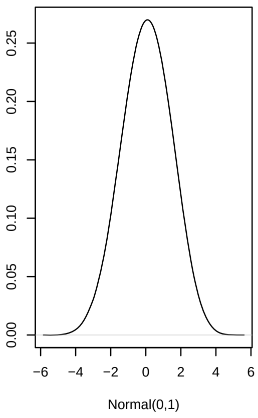

n=250 from N(0,1),Cos,b=default
0.0
0.1
0.2
0.3
0.4
−3 −2 −1 0 1 2 3
Normal(0,1)
n=250 from N(0,1),Cos,b=1
0.0
0.1
0.2
0.3
0.4
−3 −2 −1 0 1 2 3
Normal(0,1)
n=250 from N(0,1),Cos,b=2
0.0
0.1
0.2
0.3
0.4
−3 −2 −1 0 1 2 3
Normal(0,1)
n=250 from N(0,1),Cosine,b=5
0.00
0.05
0.10
0.15
0.20
0.25
0.30
−4 −2 0 2 4
Normal(0,1)
n=250 from N(0,1),Cos,b=8
0.00
0.05
0.10
0.15
0.20
−6 −4 −2 0 2 4 6
Normal(0,1)
n=250 from N(0,1),Cos,b=10
0.00
0.05
0.10
0.15
−5 0 5
Normal(0,1)


n=250 from N(0,1),Gauss,b=default
0.0
0.1
0.2
0.3
0.4
−3 −2 −1 0 1 2 3
Normal(0,1)
n=250 from N(0,1),Gauss,b=1
0.0
0.1
0.2
0.3
0.4
−3 −2 −1 0 1 2 3
Normal(0,1)
n=250 from N(0,1),Gauss,b=2
0.0
0.1
0.2
0.3
0.4
−4 −2 0 2 4
Normal(0,1)
n=250 from N(0,1),Gauss,b=5
0.00
0.05
0.10
0.15
0.20
0.25
−6 −4 −2 0 2 4 6
Normal(0,1)
n=250 from N(0,1),Gauss,b=8
0.00
0.05
0.10
0.15
−5 0 5
Normal(0,1)
n=250 from N(0,1),Gauss,b=10
0.00
0.05
0.10
0.15
−10 −5 0 5 10
Normal(0,1)

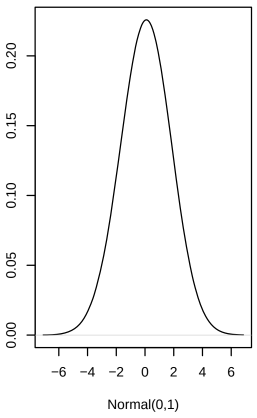
n=250 from N(0,1),Default Settings



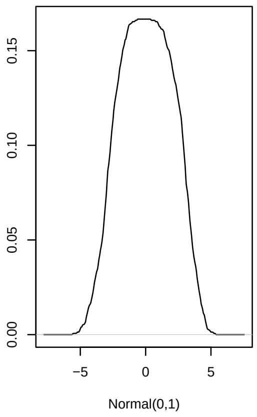
n=250 from N(0,1),Cosine,b=6



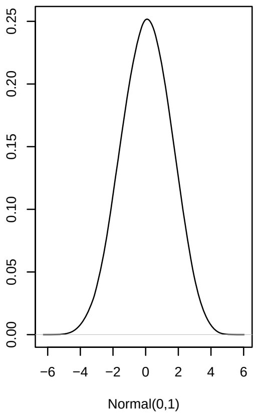
n=250 from N(0,1),Gauss,b=6



n=250 from N(0,1),Rectangular,b=6



n=250 from N(0,1),Triangle,b=6



We can conclude from these graphs that

1. the sample size n is very important

2. the choice of bandwidth h is very important

3. the use of the Gaussian kernel $K(\cdot)$ generally yields acceptable results

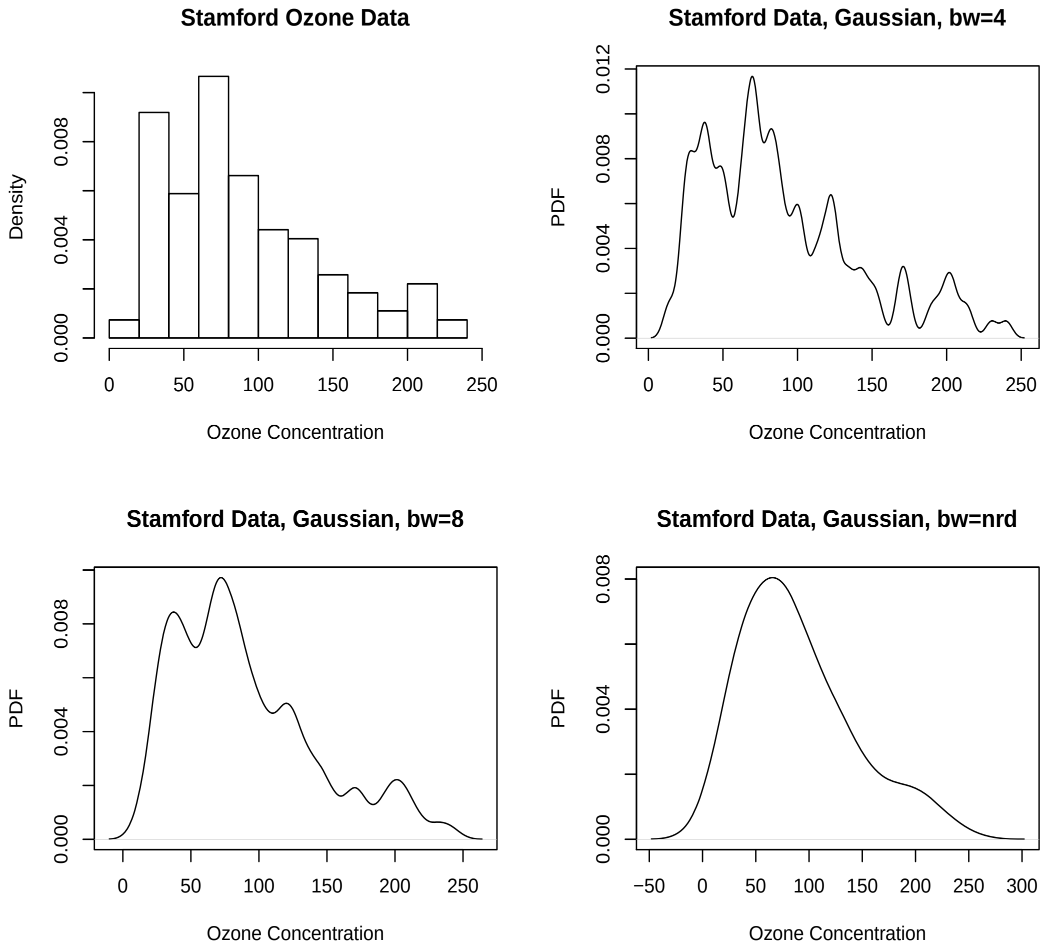
4. the selection of the number of plotting points m is not very important provided we select m large enough so that for the given value of m and plotting points y_i, we have reasonable overlap of the intervals $(y_i - h, y_i + h)$ and $(y_{i+1} - h, y_{i+1} + h)$.

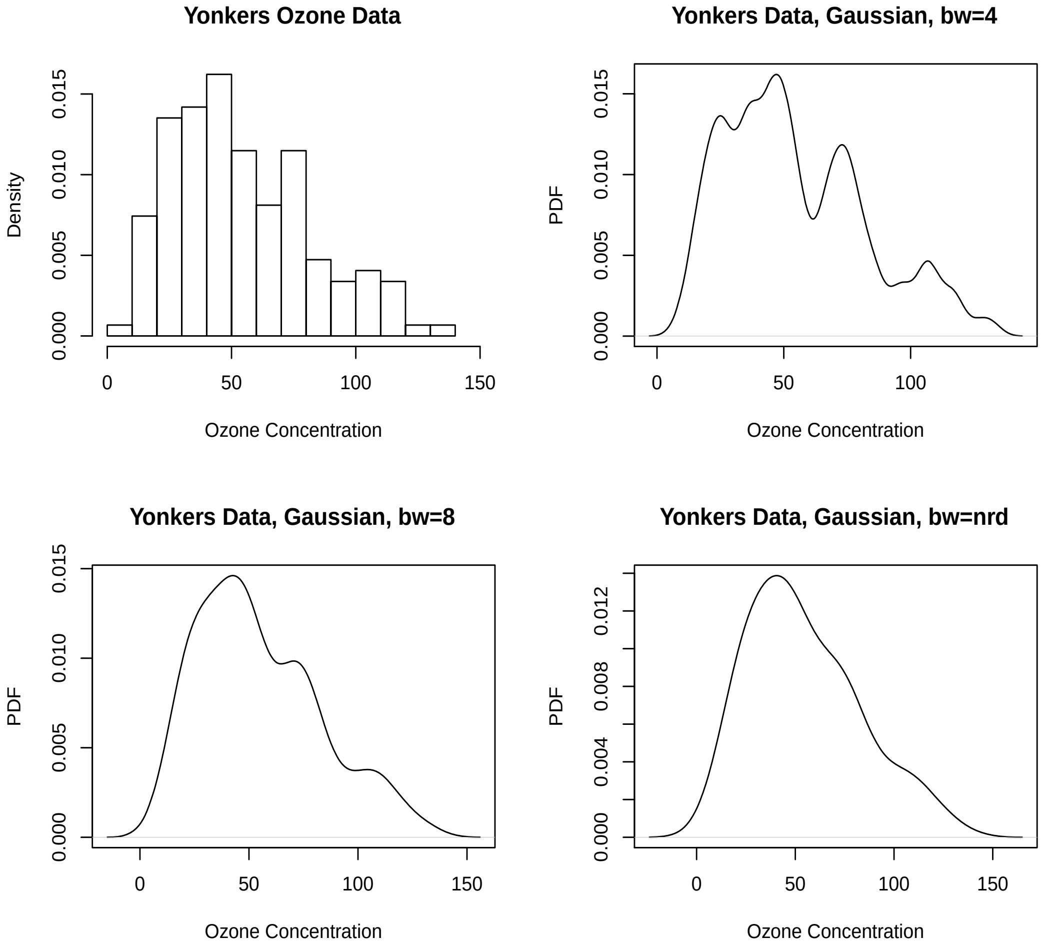
The graphs on the previous pages illustrated some of the above points. However, I would suggest that you also experiment with the kernel density estimator using the R code. Vary the kernel, bandwidth, sample size, and number of plotting points and observe the resulting differences in the plots. This will allow you to gain further understanding of the interrelationships between the various parameters in the kernel density estimator.

The article, “A reliable data-based bandwidth selection method for kernel density estimation”, *JRSS Ser. B 53*(1991), pp. 683-690, by S.J. Sheather and M.C. Jones, provides a method by which the data selects the “best” bandwidth.

The following R code, ozonekern.R, will produce histograms and kernel density estimates for the ozone data:

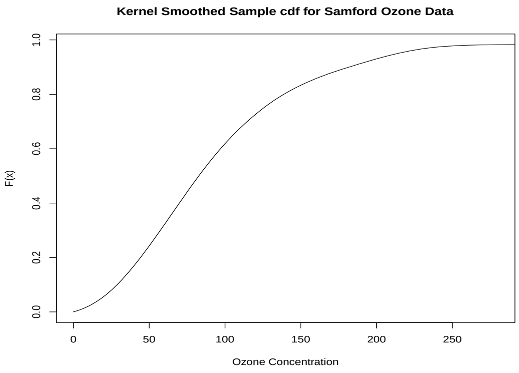
```
y1 = scan("u:/meth1/Rfiles/ozone1.DAT")

y2 = scan("u:/meth1/Rfiles/ozone2.DAT")

#postscript("u:/meth1/Rfiles/ozonekern4p1.ps",height=7,horizontal=F)

par(mfrow=c(2,2))

hist(y1,breaks=10, plot=TRUE, prob=T, xlim=c(0,250),
main="Stamford Ozone Data",
xlab="Ozone Concentration", cex=.75)

plot(density(y1,window=’g’,bw=4),type="l",
xlab="Ozone Concentration",ylab="PDF",
main="Stamford Data, Gaussian, bw=4",cex=.5)

plot(density(y1,window=’g’,bw=8),type="l",
xlab="Ozone Concentration",ylab="PDF",
main="Stamford Data, Gaussian, bw=8",cex=.5)

plot(density(y1,window=’g’,bw="nrd"),type="l",
xlab="Ozone Concentration",ylab="PDF",
main="Stamford Data, Gaussian, bw=nrd",cex=.5)

#postscript("u:/meth1/Rfiles/ozonekern4p2.ps",height=7,horizontal=F)

par(mfrow=c(2,2))

hist(y2,breaks=10, plot=TRUE, prob=T, xlim=c(0,150),
main="Yonkers Ozone Data",
xlab="Ozone Concentration", cex=.75)


plot(density(y2,window=’g’,bw=4),type="l",
xlab="Ozone Concentration",ylab="PDF",
main="Yonkers Data, Gaussian, bw=15",cex=.5)

plot(density(y2,window=’g’,bw=8),type="l",
xlab="Ozone Concentration",ylab="PDF",
main="Yonkers Data, Gaussian, bw=30",cex=.5)

plot(density(y2,window=’g’,bw="nrd"),type="l",
xlab="Ozone Concentration",ylab="PDF",
main="Yonkers Data, Gaussian, bw=nrd",cex=.5)

#graphics.off()
```

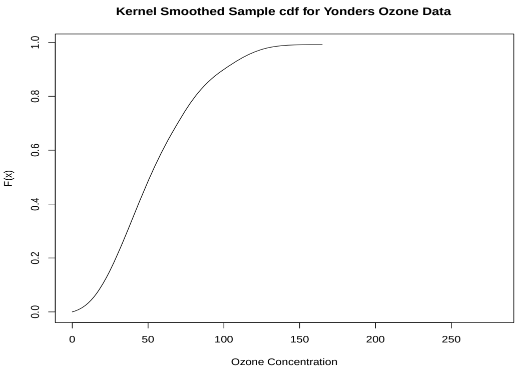

Stamford Ozone Data
Density
0.000
0.004
0.008
0
50
100
150
200
250
Ozone Concentration
Stamford Data, Gaussian, bw=4
PDF
0.000
0.004
0.008
0.012
0
50
100
150
200
250
Ozone Concentration
Stamford Data, Gaussian, bw=8
PDF
0.000
0.004
0.008
0
50
100
150
200
250
Ozone Concentration
Stamford Data, Gaussian, bw=nrd
PDF
0.000
0.004
0.008
−50
0
50
100
150
200
250
300
Ozone Concentration


Yonkers Ozone Data
Density
0.000
0.005
0.010
0.015
0
50
100
150
Ozone Concentration
Yonkers Data, Gaussian, bw=4
PDF
0.000
0.005
0.010
0.015
0
50
100
Ozone Concentration
Yonkers Data, Gaussian, bw=8
PDF
0.000
0.005
0.010
0.015
0
50
100
150
Ozone Concentration
Yonkers Data, Gaussian, bw=nrd
PDF
0.000
0.004
0.008
0.012
0
50
100
150
Ozone Concentration

The cdf is defined as $F(y) = \int_{-\infty}^{y} f(t)dt$ where $f()$ is the pdf associated with $F()$. Thus, $F(y)$ is the area under the pdf $f()$ from $-\infty$ to y. We will use this idea to produce an estimator of the cdf based on the kernel estimator of the pdf. The following R code will produce an estimator of the cdf based on the area under the kernel density estimator of the pdf. This estimator will be considerably smoother in many cases than the smoothed version of the sample cdf defined in the beginning of this handout.

```
y1 = scan("u:/meth1/Rfiles/ozone1.DAT")
deny1 = density(y1,kernel='g',bw="nrd",n=5000,from=0)
xy1 = deny1$x
pdfy1 = deny1$y
cdfy1 = c(rep(0,5000))
areay1 = c(rep(0,5000))
for (i in 1:5000)
{
areay1[i] = abs((pdfy1[i+1]+pdfy1[i])/2)*(xy1[i+1]-xy1[i])
cdfy1[i] = sum(areay1)
}
plot(xy1,cdfy1,type="l",
xlab="Ozone Concentration",ylab="F(x)",
main="Smoothed Sample cdf for Samford Ozone Data",cex=.95)
```


Kernel Smoothed Sample cdf for Samford Ozone Data


Kernel Smoothed Sample cdf for Yonders Ozone Data

STOP 9/17/21

End H:O. 4 - Skip rest of H.O. 4.

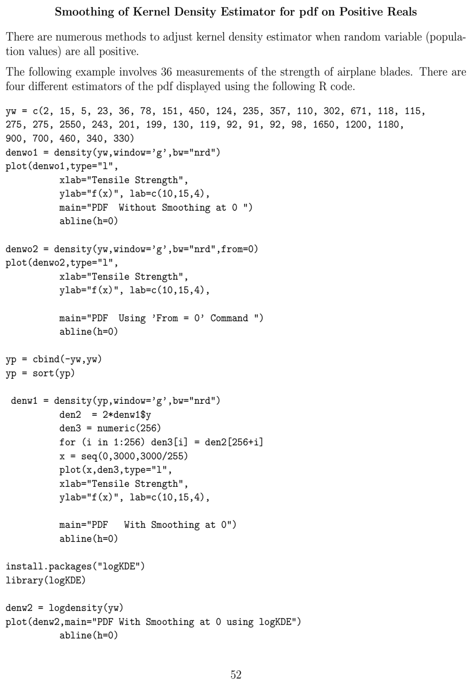
Smoothing of Kernel Density Estimator for pdf on Positive Reals

There are numerous methods to adjust kernel density estimator when random variable (population values) are all positive.

The following example involves 36 measurements of the strength of airplane blades. There are four different estimators of the pdf displayed using the following R code.

```
yw = c(2, 15, 5, 23, 36, 78, 151, 450, 124, 235, 357, 110, 302, 671, 118, 115,
275, 275, 2550, 243, 201, 199, 130, 119, 92, 91, 92, 98, 1650, 1200, 1180,
900, 700, 460, 340, 330)
denwo1 = density(yw,window='g',bw="nrd")
plot(denwo1,type="l",
          xlab="Tensile Strength",
          ylab="f(x)", lab=c(10,15,4),
          main="PDF  Without Smoothing at 0 ")
          abline(h=0)

denwo2 = density(yw,window='g',bw="nrd",from=0)
plot(denwo2,type="l",
          xlab="Tensile Strength",
          ylab="f(x)", lab=c(10,15,4),

          main="PDF  Using 'From = 0' Command ")
          abline(h=0)

yp = cbind(-yw,yw)
yp = sort(yp)

 denw1 = density(yp,window='g',bw="nrd")
          den2  = 2*denw1$y
          den3 = numeric(256)
          for (i in 1:256) den3[i] = den2[256+i]
          x = seq(0,3000,3000/255)
          plot(x,den3,type="l",
          xlab="Tensile Strength",
          ylab="f(x)", lab=c(10,15,4),

          main="PDF   With Smoothing at 0")
          abline(h=0)

install.packages("logKDE")
library(logKDE)

denw2 = logdensity(yw)
plot(denw2,main="PDF With Smoothing at 0 using logKDE")
          abline(h=0)
```

Kernel Density Estimator for pdf on Positive Values

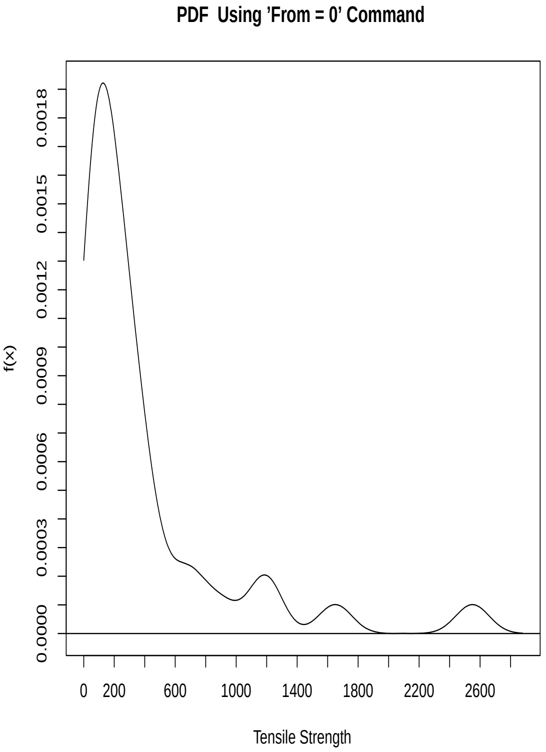


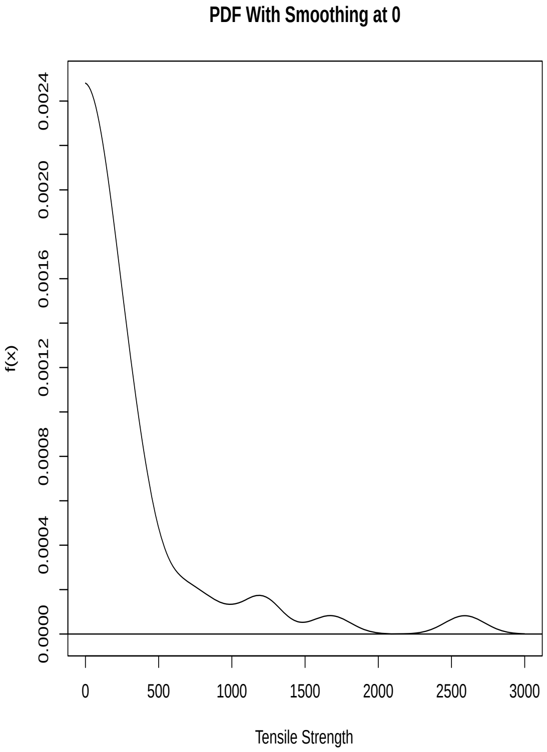
Smoothing of Kernel Density Estimator for pdf on Positive Reals

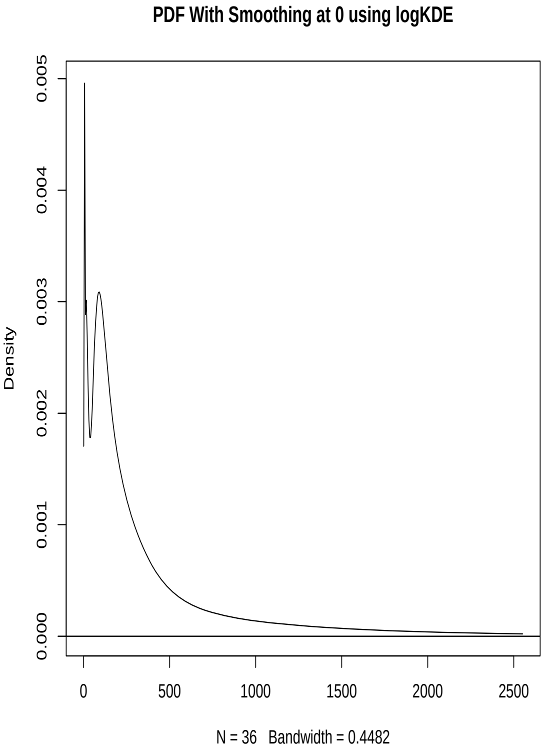
There are numerous methods to adjust kernel density estimator when random variable (population values) are all positive.

The following example involves 36 measurements of the strength of airplane blades. There are four different estimators of the pdf displayed using the following R code.

```
yw = c(2, 15, 5, 23, 36, 78, 151, 450, 124, 235, 357, 110, 302, 671, 118, 115,
275, 275, 2550, 243, 201, 199, 130, 119, 92, 91, 92, 98, 1650, 1200, 1180,
900, 700, 460, 340, 330)
denwo1 = density(yw,window='g',bw="nrd")
plot(denwo1,type="l",
          xlab="Tensile Strength",
          ylab="f(x)", lab=c(10,15,4),
          main="PDF  Without Smoothing at 0 ")
          abline(h=0)

denwo2 = density(yw,window='g',bw="nrd",from=0)
plot(denwo2,type="l",
          xlab="Tensile Strength",
          ylab="f(x)", lab=c(10,15,4),

          main="PDF  Using 'From = 0' Command ")
          abline(h=0)

yp = cbind(-yw,yw)
yp = sort(yp)

 denw1 = density(yp,window='g',bw="nrd")
          den2  = 2*denw1$y
          den3 = numeric(256)
          for (i in 1:256) den3[i] = den2[256+i]
          x = seq(0,3000,3000/255)
          plot(x,den3,type="l",
          xlab="Tensile Strength",
          ylab="f(x)", lab=c(10,15,4),

          main="PDF   With Smoothing at 0")
          abline(h=0)

install.packages("logKDE")
library(logKDE)

denw2 = logdensity(yw)
plot(denw2,main="PDF With Smoothing at 0 using logKDE")
          abline(h=0)
```


PDF Using 'From = 0' Command
f(x)
Tensile Strength


PDF With Smoothing at 0
f(x)
Tensile Strength


PDF With Smoothing at 0 using logKDE
Density
N = 36   Bandwidth = 0.4482