Local Regression

Yoshio Kajitani
Department of Statistical and Actuarial Sciences
University of Western Ontario
London, Ontario N6A 5B7
Canada

Jan 8, 1999

1 Introduction

Local regression was first well developed by actuaries in the early twentieth century. Graduation methods derived by actuaries are a form of local fitting. We used to fit a straght line to the data by using least squares regression, but this is not always suitable. We want to be more sophisticated than just fitting a straight line or polynomial to all the data. Rather we use only a point.

The purpose of this project is to provide solutions to the problems in the first two chapter of the forthcoming monograph "Local Regression" by Clive Loader. We also illustrate the use of Splus statistical library LOCFIT.

1.1 Graduation Using Summation Formula

Local regression is one of the way to give the graduation of series of observations. Similarly, as one major class of techniques used, there were summation formulae, which also give a graduated value.

The 15 point formula of Spencer rule is one of the graduation rules using summation formula and this is written as

$$|\hat{y}_j| = \frac{S_{5,4,4}}{5 \cdot 4 \cdot 4}(-3y_{j-2} + 3y_{j-1} + 4y_j + 3y_{j+1} - 3y_{j+2}) \tag{1.1}$$

Graduation rules were commonly expressed using difference operator:

$$\nabla y_i = y_{i+1/2} - y_{i-1/2}$$

By using this the general k-point moving average is given by the following result.

Theorem 1.1 *The k-point moving average has the representation*

$$\frac{S_k}{k} y_i = y_i + \frac{k^2-1}{24}\nabla^2 y_i + \frac{(k^2-1)(k^2-9)}{1920}\nabla^4 y_i + O(\nabla^6 y_i). \tag{1.2}$$

Also Spencer's rule can be written

$$\hat{y}_j = y_j + O(\nabla^4 y_j)$$

This implies that Spencer's rule has cubic reproduction property, since $\nabla^4 = 0$, when y_j is a cubic polynomial. An alternative way to see the cubic reproducing property of Spencer's formula is through the weight diagram.

1.2 Henderson formula

Henderson (1916) is the one who was interested in the local polinomial fittng. He difined non-negative weigths w_k and fitted a local cubic polinomial, solving the systme equations

$$\begin{aligned}
\sum w_k(a + bk + ck^2 + dk^3) &= \sum w_k y_{j+k} \\
\sum k w_k(a + bk + ck^2 + dk^3) &= \sum k w_k y_{j+k} \\
\sum k^2 w_k(a + bk + ck^2 + dk^3) &= \sum k^2 w_k y_{j+k} \\
\sum k^3 w_k(a + bk + ck^2 + dk^3) &= \sum k^3 w_k y_{j+k}
\end{aligned} \tag{1.3}$$

for unkown coefficient a, b, c, d. We note that this is cubic-reproducing. And also this provides graduated values right up to the boundaries. This is more appealing than Spencer's formula in that we don't need to use any extrapolation method. He determined the weight w_i, deriving perhaps the first optimality result in smoothing. Acutally, for fixed m, to produce the smoothest possible graduation based on $2m+1$ data points, the solution, which became known as Henderson's ideal formula, was a local cubic smooth with weights

$$w_k = ((m+1)^2 - k^2)((m+2)^2 - k^2)((m+3)^2) - k^2); k = -m, ..., m$$

Henderson showed the weight diagram l_k for this procedure is simply w_k multiplied by a cubic polynomial.

1.3 Exercises

Question 1.1 Consider the graduation rule

$$\hat{y}_j = \frac{S_{k,k}}{k^2} y_j$$

with k even. That is, $\hat{y}_j$ is formed by taking two successive moving average of length k.

a) Let k=4. Show the graduation rule has the explicit form

$$\hat{y}_j = \frac{1}{16}(y_{j-3} + 2y_{j-2} + 3y_{j-1} + 4y_j + 3y_{j+1} + 2y_{j+2} + y_{j+3})$$

Show this has the difference representation

$$\hat{y}_j = y_j + \frac{5}{4}\nabla^4 y_j + \frac{1}{16}\nabla^6 y_j$$

$$\begin{aligned} S_{4,4}\ y_j &= \frac{S_4}{16}(y_{j-1.5} + y_{j-0.5} + y_{j+0.5} + y_{j+1.5}) \\ &= \frac{1}{16}(y_{j-3} + 2y_{j-2} + 3y_{j-1} + 4y_j + 3y_{j+1} + 2y_{j+2} + y_{j+3}) \end{aligned}$$

Latter part is shown as below.

$$\begin{aligned} \nabla^4 y_j &= \nabla^2(y_{j+1} - 2y_j + y_{j-1}) \\ &= y_{j-2} - 4y_{j-1} + 6y_j - 4y_{j+1} + y_{j+2} \end{aligned}$$

$$\nabla^6 y_j = y_{j-3} - 6y_{j-2} + 15y_{j-1} - 20y_j + 15y_{j+1} - 6y_{j+2} + y_{j+3}$$

$$\begin{aligned} \hat{y}_j &= y_j + \frac{5}{4}\nabla^2 y_j + \frac{1}{2}\nabla^4 y_j + \frac{1}{16}\nabla^6 y_j \\ &= \frac{1}{16}(y_{j-3} + 2y_{j-2} + 3y_{j-1} + 4y_j + 3y_{j+1} + 2y_{j+2} + y_{j+3}) \end{aligned}$$

b) Let $y_j = j^2/2$. For general k. show

$$\hat{y}_0 = \frac{k^2-1}{12}$$

Because of symmetry

$$\begin{aligned}
S_{k,k}\ y_0 &= \frac{1}{k^2}\sum_{j=-k+1}^{0}\frac{j^2}{2} + \frac{1}{k^2}\sum_{j=0}^{k-1}(k-j)\frac{j^2}{2} \\
&= \frac{1}{k}\sum_{j=0}^{k-1} j^2 - \frac{1}{k^2}\sum_{j=1}^{k-1}(k-j)\frac{j^3}{2} \\
&= \frac{1}{k}\frac{k(k-1)(2k-1)}{6} - \frac{1}{k^2}\frac{k^2(k-1)^2}{4} \\
&= \frac{k^2-1}{12}
\end{aligned}$$

c) Let $y_j = (j^4 - j^2)/24$. For general k, show

$$\hat{y}_0 = \frac{(k^2-1)(k^2-4)}{360}$$

As same as b)

$$\begin{aligned}
S_{k,k}\ y_0 &= 2 * \frac{1}{k^2}\sum_{j=-k+1}^{0}(k-j)(j^4-j^2)/24 \\
&= \frac{1}{12k}\sum_{j=0}^{k-1}(j^4-j^2) - \frac{1}{12k^2}\sum_{j=0} k-1(j^5-j^3) \\
&= \frac{(k^2-1)(k^2-4)}{360}
\end{aligned}$$

d) Show

$$\hat{y}_j = (I + \frac{k^2-1}{12}\nabla^2 + \frac{(k^2-1)(k^2-4)}{360}\nabla^4 + O(\nabla^6))y_j$$

If we define y_i like the last two questions, all lower and higher order differences which include k character are 0. Moreover, as $y_0 = \frac{0^2}{2} = 0, y_o = \frac{0^4 - 0^2}{24} = 0$, we can say

$$\hat{y}_j = (I + \frac{k^2 - 1}{12}\nabla^2 + \frac{(k^2 - 1)(k^2 - 4)}{360}\nabla^4 + O(\nabla^6))y_j$$

This is valid for any k.

Here, as $\frac{S_{k,k}}{k^2}y_j = \frac{S_{k,k}}{S_{k,k}}y_j$, by using $\sqrt{(1+x)} = 1 + \frac{x}{2} - \frac{x^2}{8} + O(x^3)$,

$$\begin{aligned}
S_{k,k}y_j &= \sqrt{I + \frac{k^2 - 1}{12}\nabla^2 + \frac{(k^2 - 1)(k^2 - 4)}{360}\nabla^4 + O(\nabla^6)y_j} \\
&= (I + \frac{k^2 - 1}{24}\nabla^2 + \frac{(k^2 - 1)(k^2 - 4)}{720}\nabla^4 - \frac{1}{8}\frac{(k^2 - 1)^2}{144}\nabla^4 + O(\nabla^6))y_j \\
&= (I + \frac{k^2 - 1}{24}\nabla^2 + \frac{(k^2 - 1)(k^2 - 9)}{1920}\nabla^4 + O(\nabla^6))y_j
\end{aligned}$$

Question 1.6 For each year in the age range 20 to 45, 1000 customers each wish to buy a \$10000 life insurance policy. Two competing companies set premiums as follows: first, estimate the mortality rate for each age, then set the premium to cover the expected payout, plus a 10 For example, if the company estimates 40 year olds to have a mortality rate of 0.01, the expected (per customer) payout is $0.01 \times \$10000 = \$100$, so the premium is \$110. Both companies use Spencer's mortality data to estimate mortality rates. The Gauss Life Company uses a least squares fit to the data, while Grad underwriting Co. applies Spencer's 15 point formula.

a) Compute for each age group the premiums charged by each company.

Graduated values by each method and the premiums charged by each company are shown in Table 1.3.

b) Suppose perfect customer behaviour, so, for example, all the 40 years old customers choose the company offering the lowest premium to 40 year olds. Also suppose Spencer's 21 point rule provides the true mortality rates. Under these assumptions, compute the expected profit (or loss) for each of the two companies.

Assuming each age group has the same population, Grad Underwriting Co., which uses Spencer's 15 point rule, gets more profit per person (\$ 6.80) than Gauss Life Company, which uses least square method (\$ 2.92).

age	**LSFIT**	Premium	**Spencer's 15 point rule**	Premium
20	.00334	36.74	.00421	46.31
21	.00363	39.93	.00429	47.19
22	.00393	43.23	.00441	48.51
23	.00422	46.42	.00455	50.05
24	.00452	49.72	.00469	51.59
25	.00481	52.91	.00482	53.02
26	.00511	56.21	.00495	54.45
27	.00541	59.51	.00508	55.88
28	.00570	62.70	.00526	57.86
29	.00600	66.00	.00549	60.39
30	.00629	69.19	.00577	63.47
31	.00659	72.49	.00610	67.10
32	.00688	75.68	.00647	71.17
33	.00718	78.98	.00684	75.24
34	.00748	82.28	.00721	79.31
35	.00777	85.47	.00755	83.05
36	.00807	88.77	.00786	86.46
37	.00836	91.96	.00814	99.54
38	.00866	95.26	.00842	92.62
39	.00896	98.58	.00877	96.47
40	.00925	101.75	.00919	101.09
41	.00955	105.05	.00968	106.48
42	.00984	108.24	.01019	112.09
43	.01014	111.54	.01067	117.37
44	.01043	114.73	.01105	121.55
45	.01073	118.03	.01133	124.63
total		2011.35		2012.89

Table 1.1: Premiums charged by least square fit and Spencer's 15 point rule

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
Ungraduated Rate	**Devide by 8=** u	**Sum in 3's**	$u_{-2}+u_{+2}$	**(3)-(4)**	**(5) multiplied by 3**	**(6)+(2)**	**Sum in 5's**	**Devide by 40**	**Sum in 4's**	**Sum in 4's = Graduated Rate**
	48	144								
	49	147	98	49	147	196				
	50	150	100	50	150	200				
	51	153	102	51	153	204	1032	25.8		
	52	156	104	52	156	208	1046	26.15	103.85	
	53	159	102	57	171	224	1056	26.4	104.1	
0.000539	54	158	106	52	156	210	1020	25.5	105.375	0.00421
0.000511	51	159	106	53	159	210	1042	26.05	108.05	0.00429
0.000536	54	158	120	38	114	168	1097	27.425	111.55	0.00441
0.000528	53	173	114	59	177	230	1163	29.075	115.75	0.00455
0.000663	66	182	111	71	213	279	1160	29	119.575	0.00469
0.000631	63	186	115	71	213	276	1210	30.25	122	0.00482
0.000574	57	182	132	50	150	207	1250	31.25	124.675	0.00495
0.000624	62	185	133	52	156	218	1260	31.5	128.325	0.00508
0.000658	66	198	130	68	204	270	1267	31.675	133.025	0.00526
0.000704	70	209	136	73	219	289	1356	33.9	139.5	0.00549
0.000734	73	217	147	70	210	283	1438	35.95	147.95	0.00577
0.000744	74	228	154	74	222	296	1519	37.975	156.75	0.00610
0.000809	81	239	166	73	219	300	1605	40.125	165.8	0.00646
0.000836	84	258	169	89	267	351	1708	42.7	175.95	0.00684
0.000933	93	272	178	94	282	375	1800	45	185.55	0.00721
0.000960	95	285	188	97	291	386	1925	48.125	193.475	0.00755
0.000973	97	296	199	97	291	388	1989	49.725	200.35	0.00786
0.001035	104	307	200	107	321	425	2025	50.625	206.55	0.00813
0.001058	106	315	212	103	309	415	2075	51.875	213.075	0.00842
0.001045	105	326	224	102	306	411	2173	54.325	222.425	0.00877
0.001145	115	340	233	107	321	436	2250	56.25	234.9	0.00919
0.001195	120	362	240	122	366	486	2399	59.975	248.5	0.00968
0.001268	127	382	257	125	375	502	2574	64.35	262.1	0.01019
0.001345	135	404	261	143	429	564	2717	67.925	273.925	0.01067
0.001418	142	418	270	148	444	586	2794	69.85	281.975	0.01105
0.001405	141	426	280	146	438	579	2872	71.8	286.7	0.01133
	143	429	289	140	420	563	2896	72.4	290.125	
	145	435	290	145	435	580	2906	72.65		
	147	441	294	147	441	588	2931	73.275		
	149	447	298	149	447	596				
	151	453	302	151	453	604				
	153	459								

Table 1.2: The way to apply Spencer's 15 point rule

age	**LSFIT**	Premium − Payment	**Spencer's 15 point rule**	Premium − Payment	**Spencer's 21** point rule	Payment
20	.00334	-3.96	.00421		.00407	40.7
21	.00363	-2.27	.00429		.00422	42.2
22	.00393	-0.47	.00441		.00437	43.7
23	.00422	1.22	.00455		.00452	45.2
24	.00452	3.02	.00469		.00467	46.7
25	.00481	4.81	.00482		.00481	48.1
26	.00511		.00495	4.85	.00496	49.6
27	.00541		.00508	4.58	.00513	51.3
28	.00570		.00526	4.66	.00532	53.2
29	.00600		.00549	4.89	.00555	55.5
30	.00629		.00577	5.27	.00582	58.2
31	.00659		.00610	5.70	.00614	61.4
32	.00688		.00647	6.37	.00648	64.8
33	.00718		.00684	7.04	.00682	68.2
34	.00748		.00721	7.71	.00716	71.6
35	.00777		.00755	8.15	.00749	74.9
36	.00807		.00786	8.46	.00780	78.0
37	.00836		.00814	8.34	.00812	81.2
38	.00866		.00842	8.22	.00844	84.4
39	.00896		.00877	8.47	.00880	88.0
40	.00925		.00919	9.29	.00918	91.8
41	.00955	9.05	.00968		.00960	96.0
42	.00984	7.74	.01019		.01005	100.5
43	.01014	6.24	.01067		.01053	105.3
44	.01043	4.43	.01105		.01103	110.3
45	.01073	2.33	.01133		.01157	115.7
Average		2.92		6.80		

Table 1.3: Profit to undergrad for each company

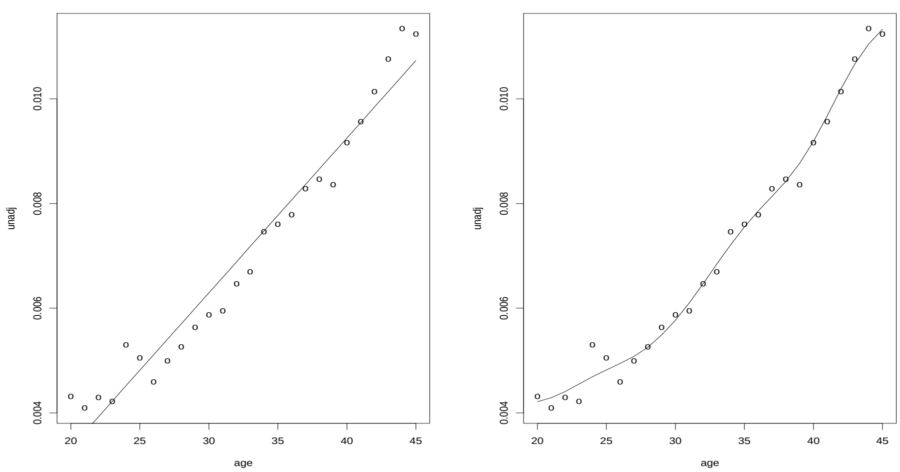

Figure 1: Mortality Rates and a Least Squares Fit(left) and Spencer's 15 point rule(right)

2 Univariate Local Regression

2.1 What is local Regression

Basic definition of local regression

Local regression is used to model a relation between a predictor variable X and response variable Y. We assume the model of the form

$$Y_i = \mu(x_i) + \epsilon_i$$

The way to determine the weights

For a point x, define a bandwidth $h(x)$ and a smoothing window$(x - h(x), x + h(x))$. To estimate $\mu(x)$, we shall use only observations within this window. Moreover, we assign weights according to a formula

$$w_i(x) = W(\frac{x_i - x}{h(x)}) \tag{2.4}$$

where $W(x)$ is a symmetric function satisfying $W(0) = 1, W(1) = 0$ and W is decreasing on the interval $[0, 1]$.

polynomial approximation to the function μ

For exmaple, a local quadratic approximation

$$\mu(u) \approx a_0 + a_1(u - x) + \frac{1}{2}a_2(u - x)^2 \tag{2.5}$$

whenever $|u - x| < h(x)$. A compact notation that we use throught for polynomials is

$$a_0 + a_1(u - x) + \tfrac{1}{2}a_2(u - x)^2 =< \mathbf{a}, \mathbf{A}(u - x) >$$

where a is a vector of coefficients and $A(\dot{})$ is a vector of the fitting functions; for local polynomials of degree p,

$$a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix}$$

$$A(v) = \begin{pmatrix} 1 \\ v \\ \vdots \\ \frac{v^p}{p!} \end{pmatrix}$$

The way to estimate cofficients

The coefficients a can be estimated by locally weighted least squares. That is, choosing $\hat{a}$ to be the minimizer of

$$\sum_{i=1}^{n} w_i(x)(Y_i - < a, A(x_i, x) >)^2 \tag{2.6}$$

Definition2.1 (Local Regression Estimate) The local regression estimate is

$$\hat{\mu}(x) =< \hat{\mathbf{a}}, \mathbf{A}(0) >= \hat{a}_0 \tag{2.7}$$

obtained by setting $u = x$ in (2.2).

Matrix form

Let $\mathbf{X}$ be the $n \times (p+1)$ design matrix with rows $\mathbf{A}(x_i - x)^T$: $\mathbf{W}$ be the diagonal matrix with entries $w_i(x)$ and $Y = (Y_1, Y_2, ...Y_n)^T$ The weight ed sum of squares (2.3) can be written in matrix form:

$$(Y - \mathbf{X}a)^T \mathbf{W}(Y - \mathbf{X}a)$$

If $\mathbf{WX}$ has full column rank. least squares theory gives the explicit expression

$$\hat{a} = (\mathbf{X}^T\mathbf{WX})^{-1}\mathbf{X}^T\mathbf{WY} \tag{2.8}$$

2.1.1 Interpreting the local regression estimate

In local regression, instead of concentrating on the coefficients estimate, we focus on the fitted curve.

2.2 Local Regression with LOCFIT

The main component of the S version is the function **locfit** , which calls the C code to do the fitting and returns an object of class "locfit". The following is an example of locfit.

```
> fit <- locfit($No_x$, data=ethanol, alpha=0.5)
```

alpha controls the bandwidth. To look at this plot, we use the command

```
> plot(fit, get.data=T)
```

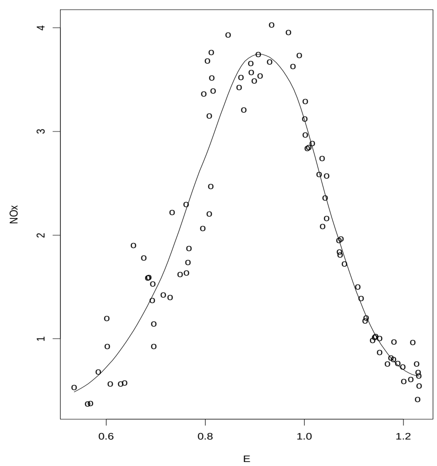

Figure 2: Local regression fit of the ethanol data

2.3 Model Choice: Bias and Variance

The weight function, the band width, the degree of local ploynomial control a bias-variance trade off. if the band width is too small, insufficeint data will fall in the smoothing window, and a noisy estimate with large variance will result. If the band width is too large, real features of the dataset may be lost, and the smoothed estimate will have substantial bias. We would like more general procedures that can be used when these assumptions are violated.

2.3.1 Bandwidth

Example of a diabetes datasets. We displayed the results of quadratic fitting with four different bandwidths:$h = 2, 4, 5$ and 16. At the smallest bandwidth, $h = 2$, the fitted curve seems far too noisy, given the sample size. At $h = 4$, the fit is still slightly too noisy; $h = 8$ produces a better result. The largest bandwidth $h = 16$, oversmooths, with the transition from the initial increase to the flat being rounded off too much.

k-nearest neighbor bandwidths and fixed bandwidths

In LOCFIT, a nearest neighbor bandwidth is specified as a single component smoothing parameter **alpha**. Also specifing a nearest neithbor component $\alpha_0 = 0.4$ and fixed component $\alpha_1 = 3$ is achieved by giving the **alpha** $=$ $\mathbf{c(0.4, 3)}$ argument to the **locfit** function.

2.3.2 Local Polynomial Degree

Like the bandwidth, the degree of local polynomial used governs a trade-off between bias and variance. The degree of local polynomial can be determined by using **deg** argument to the **locfit()** function.

2.3.3 The Weight Function

The weight function $W(u)$ in (2.1) is usually chosen to be symmetric, peaked at 0 and decay to 0 for large values of u. The choice of weight function influences the bias-variance tradeoff, but much less so than bandwidth and degree of fit. This is achieved by using **kern** $=$ argument. The LOCFIT weight functions are as follows

rect	Rectangular	$W(x) = 1, \lvert x \rvert < 1$
tria	Triangular	$W(x) = 1 - \lvert x \rvert, \lvert x \rvert < 1$
epan	Epanechnikov	$W(x) = 1 - x^2, \lvert x \rvert < 1$
bisq	Bisqurare	$W(x) = 1 - x^2, \lvert x \rvert < 1$
tcub	Tricube	$W(x) = (1 - \lvert x \rvert^3)^3, \lvert x \rvert < 1$
trwt	Triweight	$W(x) = (1 - \lvert x \rvert^2)^3, \lvert x \rvert < 1$
gauss	Gaussian	$W(x) = exp(-(2.5x)^2/2)$

Table 2.4: The LOCFIT weight functions

2.4 Linear Estimation

The local regression estimate defined by (2.4) is linear in the responses Y_i; that is, it can be written in the form

$$\hat{\mu}(x) = \sum_{i=1}^{n} l_i(x) Y_i =< l(x), Y > \tag{2.9}$$

for an appropriate vector function $l(x)^T = (l_1(x) \ldots l_n(x))$ From (2.5) we have explicitly,

$$l(x)^T = {e_1}^T (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \tag{2.10}$$

assuming $\mathbf{X}^T \mathbf{W} \mathbf{X}$ is non-singular. The hat matrix is the $n \times n$ matrix $\mathbf{L}$ with rows $l(x_i)^T$, which maps the data to the fitted values;

$$\begin{pmatrix} \hat{\mu}(x_1) \\ \vdots \\ \hat{\mu}(x_n) \end{pmatrix} = \mathbf{L} S \tag{2.11}$$

Definition 2.2 (Weight Diagram) For fixed x the weight vector

$$l(x)^T = (l_1(x) \ldots l_n(x))$$

is called the weight diagram.

hat matrix() returns a matrix with n rows and columns corresponding to the fitting points; if we set **ev** = "**data**", this matrix is the transpose of the hat matrix defined by (2.11)

Theorem 2.2 *(Henderson's Theorem) The weight diagram for a local polynomial fit of degree p has the form*

$$l_i(x) = W(\frac{\mathbf{X_i} - x}{h}) < \alpha, A(\mathbf{X_i} - x) > \tag{2.12}$$

2.4.1 Influence, Variance and Degrees of Freedom

The variance of the linear estimate (2.6) is

$$\text{Var}\,(\hat{\mu}(x)) = \sigma^2 \sum_{i=1}^{n} l_i(x)^2 = \sigma^2 \| \, l(x) \, \|^2$$

In matrix notation, the variance is

$$var(\hat{\mu}(x)) = \sigma^2 {e_1}^T (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{W}^2 \mathbf{X}) (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} e_1 \tag{2.13}$$

A quantity closely related to the variance is the influence function.

Definition 2.2 (Influence Function)

$$\text{infl}\,(x) = {e_1}^T (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} e_1 W(0) \tag{2.14}$$

Note the diagonal elements of the hat matrix $\mathbf{L}$ are given by $l_i(x_i) = \text{infl}\,(x_i)$

Theorem 2.3 *Suppose the weight function W(u) is symmetric, non-negative and decreasing on [0, ∞). Then*

- *the influence function dominates the variance:*

$$\frac{1}{\sigma^2} var(\hat{\mu}(x)) \leq \text{infl}\,(x) \tag{2.15}$$

  For the rectangular weight function, this is an equality

- *at the observation points x_i,*

$$\text{infl}\,(x_i) \leq 1 \tag{2.16}$$

  and hence local regression is variance-reducing.

- *the influence function is a decreasing function of the bandwidths: i.e. if $h_1 < h_2$ and* $\mathrm{infl}_1(x)$ *are the corresponding influence functions, then*

$$\mathrm{infl}_1(x) \geq \mathrm{infl}_2(x) \tag{2.17}$$

Definition 2.4 (Degrees of Freedom)

Two defintions for the degrees of freedom of a smoother are

$$\begin{aligned}
\nu_1 &= \sum_{i=1}^{n} \mathrm{infl}\,(x_i) = \mathrm{tr}\,(\mathbf{L}) \\
\nu_2 &= \sum i = 1^n \| \, l(x_i) \, \|^2 = \mathrm{tr}\,(\mathbf{L}^T\mathbf{L})
\end{aligned}$$

In the case of a parametric fit, both these definitions are easily shown to reduce to the number of parameters. For local fits, they are in general not equal, and we will shortly see that both definitions are useful in certain circumstances.

2.4.2 Bias

The bias of the linear estimate (2.9) is

$$\begin{aligned}
b(x) &= E\hat{\mu}(x) - \mu(x) \\
&= \sum_{i=1}^{n} l_i(x)\mu(x) - \mu(x)
\end{aligned}$$

Using Taylor series expansion

$$\begin{aligned}
b(x) &= \sum_{i=1}^{n} l_i(x)(\mu(x_i) - \mu(x)) \\
&= \frac{\mu^{(p+1)}(x)}{(p+1)!} \sum_{i=1}^{n} l_i(x)(x_i - x)^{p+1} + \frac{\mu^{(p+2)}(x)}{(p+2)!} + \cdots
\end{aligned} \tag{2.18}$$

2.4.3 Asymptotic Approximations

Explicit closed form expressions such as (2.10) are suitable for numerical computation. However, they only give limited insight into the behavior of the variance function as, for example, the design, sample size or bandwidth change. Thus, various asymptotic approximations can be useful. Suppose the x_i are sampled from a continuous density $f(x)$. Then $n^{-1}\mathbf{X}^T\mathbf{W}\mathbf{X}$ is a sample average, and for large n,

$$\begin{aligned} \frac{1}{n}\mathbf{X}^T\mathbf{W}\mathbf{X} &= \frac{1}{n}\sum w_i(x)A(x_i - x)A(x_i - x)^T \\ &\approx \int W(\frac{u}{h})A(u)A(u^T)f(x+u)du \\ &\approx hf(x)\mathbf{H}\int W(v)A(v)A(v)^T dv\mathbf{H} \end{aligned} \tag{2.19}$$

As well as a sample average, we can define asymptotic influence, variance, kernels, degrees of freedom and bias.

2.5 Residuals

After performing a local fit, we need to perform diagnostic model checks. Plots that can be useful include

- Residuals vs predictor variables, for detecting lack of fit, such as a trimmed peak.
- Residuals(or absolute residuals) vs the fitted values, for detecting lack of homogeneity; in particular, to find dependence of the residual variance on response.
- QQ plots, to detect departure from nomality, such as skewness or heavy tails, in the residual distribution.
- Serial plots of $\hat{\epsilon}$ vs $\hat{\epsilon}_{i-1}$ to detect correlation between residuals.
- Sequential plot of residuals, in the order the data were collected. In an industrial experiment, this may detect gradual shift in experimental conditions over time.

Studentization

By defitition. the residual variance is $\sigma^2 = E({\epsilon_i}^2)$ But particularly when small bandwidths are used, the estimated residuals $\hat{\epsilon}_i$ tend to be slightly smaller. Formally, we have

$$\begin{aligned}
\text{Var}\,(\hat{\epsilon}_i) &= \text{Var}\,(\mathbf{Y}_i - \hat{\mu}(x_i)) \\
&= \text{Var}\,(\mathbf{Y}_i) - 2\ \text{Cov}\,(\mathbf{Y}_i, \hat{\mu}(x_i)) + \ \text{Var}\,(\hat{\mu}(x_i)) \\
&= \sigma^2(1 - 2\ \text{infl}\,(x_i) + \parallel l(x_i) \parallel^2) \qquad (2.20)
\end{aligned}$$

2.5.1 Residuals in LOCFIT

S-plus commands

```
> res <- residuals(fit)
> residuals(fit, studentize=T)
```

Example 2.1. We construct smoothed residual plots for the four smooths.

```
> h <- c(2,4,8,16)
> for(i in 1:length(h)){
+    fit <- locfit(lcp~age, data=diab, alpha=c(0, h[i]))
+    newd <- cbind(diab, resid=residuals(fit))
+    fitr <- locfit(resid~age, data=newd, alpha=c(0,2))
+    plot(fitr, main=paste("h=", h[i]), ylim=c(-2,2), get.data=T)
+    abline(h=0, lty=2)
}
```

2.5.2 Variance estimation

Suppose the fit is unbiased, so $E(\hat{\epsilon}_i) = 0$. Then by (2.20), we have

$$\begin{aligned}
E\sum_{i=1}^{n} \hat{\epsilon}_i^2 &= \sigma^2 \sum_{i=1}^{n} (1 - 2\ \text{infl}\,(x) + \parallel l(x_i) \parallel^2) \\
&= \sigma^2(n - 2\ \text{tr}\,(\mathbf{L}) + \ \text{tr}\,(\mathbf{L}^T\mathbf{L})
\end{aligned}$$

Thus, an estimate of the variance is

$$\hat{\sigma}^2 = \frac{1}{n - 2\ \text{tr}\,(\mathbf{L}) + \ \text{tr}\,(\mathbf{L}^T\mathbf{L})} \sum_{i=1}^{n} \hat{\epsilon}_i^2 \qquad (2.21)$$

2.6 Cross Validation and Cp

2.6.1 Theory and Definitions

In this section we show the criterion to estimate how good a local regression estimate is. One possible criterion is the predicted mean squared error for future observations:

$$\text{PMSE}(\hat{\mu}) = E(Y_{new} - \hat{\mu}(x_{new}))^2 \tag{2.22}$$

Definition 2.5 (Cross Validation) *The cross validation estimate of the* PMSE *of an estimate* $\hat{\mu}$ *is*

$$CV(\hat{\mu}) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{\mu}_{-i}(x_i))^2 \tag{2.23}$$

The cross validation score $\text{CV}(h)$ looks like a lot of work to compute. Since $\hat{\mu}_{-1}(x_i)$ must be computed for each observation in turn. Fortunately, there is a short cut.

Theorem 2.3 *If* $\text{infl}(x_i) < 1$, *the leave-one-out cross validation estimate* $\hat{\mu}_{-1}(x)$ *is given by*

$$\hat{\mu}_{-i}(x_i) = \frac{\hat{\mu}(x_i) - \text{infl}(x_i)Y_i}{1 - \text{infl}(x_i)}$$

and

$$\text{CV}(\hat{\mu}) = \frac{1}{n}\sum_{i=1}^{n}\frac{(Y_i - \hat{\mu}(x_i))^2}{(1 - \text{infl}(x_i))^2} \tag{2.24}$$

Definition 2.6 (Generalized cross validation) The generalized cross validation of a local estimate $\hat{\mu}$, with hat matrix $\mathbf{L}$, is

$$\text{GCV}(\hat{\mu}) = n\frac{\sum_{i=1}^{n}(Y_i - \hat{\mu}(x_i))^2}{(n - tr(\mathbf{L}))^2}$$

Risk Function

The prediction error (2.22) is the sum of components; the random error of Y_{new} and the estimation error in $\hat{\mu}(x_{new})$. Thus,

$$\text{PMSE}(\hat{\mu}) = \sigma^2 + E(\hat{\mu}(x_{new}) - \mu(x_{new}))^2$$

Thus, minimizeing the prediction error is equivalent to minimizing the estimation error $E(\hat{\mu}(x_{new}) - \mu(x_{new}))^2$. This motivates consideration of the risk function

$$R(\hat{\mu}, \mu) = \frac{1}{\sigma^2} \sum_{i=1}^{n} E(\hat{\mu}(x_i) - \mu(x_i))^2 \tag{2.25}$$

A bias-variance decomposition gives

$$\begin{aligned} R(\hat{\mu}, \mu) &= \frac{1}{\sigma^2} \sum_{i=1}^{n} b(x_i)^2 + \frac{1}{\sigma^2} \sum_{i=1}^{n} v(x_i) \\ &= \frac{1}{\sigma^2} \sum_{i=1}^{n} b(x_i)^2 + tr(\mathbf{L}^T \mathbf{L}) \end{aligned} \tag{2.26}$$

where $b(x)$ and $v(x)$ are the bias and variance of $\hat{\mu}(x)$ respectively. This depends on the unknown mean function μ, so cannot be computed directly. However, consider the expected residual sum of squares, which can again be decomposed into bias and variance components.

$$\begin{aligned} E \sum_{i=1}^{n} (Y_i - \hat{\mu}(x_i))^2 &= \sum_{i=1}^{n} b(x_i)^2 + \sum_{i=1}^{n} \text{ Var } (Y_I - \hat{\mu}(x_i)) \\ &= \sum_{i=1}^{n} b(x_i)^2 + \sigma^2 tr((\mathbf{I} - \mathbf{L})^T (\mathbf{I} - \mathbf{L})) \end{aligned} \tag{2.27}$$

By eliminating the squared bias terms from (2.26) and (2.27), we arrive at an unbiased estiamte of the risk of the local estimate.

Definition 2.7 (C_P estimate of risk)

$$\begin{aligned} \text{CP } (\hat{\mu}) &= \frac{1}{\sigma^2} \sum_{i=1}^{n} (Y_i - \hat{\mu}(x_i))^2 - tr((\mathbf{I} - \mathbf{L})^T (\mathbf{I} - \mathbf{L})) + tr(\mathbf{L}^T \mathbf{L}) \\ &= \frac{1}{\sigma^2} \sum_{i=1}^{n} (Y_i - \hat{\mu}(x_i))^2 - n + 2tr(\mathbf{L}) \end{aligned}$$

2.6.2 Implementation and Computation

Splus Command to get cross validation and C_p criteria

GCV

```
> fit <- locfit(NOx~E, data=ethanol, alpha=0.5)
> fit$dp
 nnalph fixh adpen cut  lk  t0  t1   rv
```

This has extracted a vector dp from the fitted object. Those of interest to us are lk (-0.5 times the residual sum of squares); t0 (tr($\mathbf{L}$)) and t1(tr($\mathbf{L}$'$\mathbf{L}$)). Since this dataset contains 88 points, the GCV score is computed as

```
> gcv <- 88*(-2*fit$dp["lk"])/(88-fit$dp["t0"])^2
```

Of course, we can write a function to do this for us:

```
> gcv <- function(...)
{
 m <- match.call()
m[[1]] <- as.name("locfit")
fit <- eval(m, sys.parent())
z <- fit$dp[c("lk","t0","t1")]
n <- fit$mi["n"]
z <- c(z, (-2 * n * z[1])/(n-z[2])^2)
names(z) <- c("lik", "infl", "vari", "gcv")
z
}
```

Example 2.2

We apply the **gcv()** function to the ethanol dataset for a range of smoothing parameters:

```
> alpha <- seq(0.2, 0.8, by= 0.05)
> g <- matrix(nrow=length(alpha), ncol=4)
> for(i in 1:length(alpha))
+   g[i,] <- gcv(NOx~E, data=ethanol, alpha=alpha[i])
```

We plot the results against the fitted degrees of freedom:

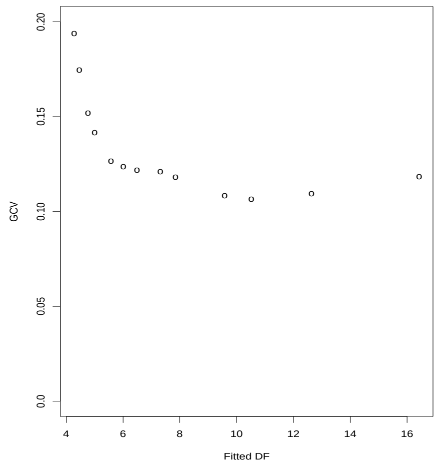

Figure 3: Generalized cross validation scores for the ethanol dataset

```
> plot(g[,3], g[,4], ylim=c(0,0.2), xlab="Fitted DF", ylab="GCV")
```

One may be tempted to go to extensive lengths to find a smoothing parameter that minimizes the GCV score. But a consequence of Figure (3) is that going to extensive lengths to minimize GCV is

- Very time-consuming.
- Very data sensitive.
- Likely to produce an unsatisfactory fit.
- No grounds for claiming a smoothing parameter is 'the best' or 'optimal'.

At best criterion such as GCV only provide a guide as to a reasonable range of smoothing parameters.

Cross validation

There are three ways to carriculate. For small datasets,

```
> fit <- locfit(NOx~E, data=ethanol, alpha=0.5, ev="cross")
```

```
> -2*fit$dp["lk"]/88
```

This is a very precise way. But, it takes a lot of time. So, For large datasets, we can use

```
> fit <- locfit(NOx~E, data=ethanol, alpha=0.5)
> r <- residuals(fit)
> infl <- fitted(fit, what="infl")
> mean((r/(1-infl))^2)
```

A slightly simpler alternative is

```
> mean((ethanol$NOx-fitted(fit,cv=T))^2)
```

When provided with the $cv = T$ argument, the **fitted.locfit()** function computes thes values

$$\hat{\mu}(x_i) - \operatorname{infl}(x_i)(Y_i - \hat{\mu}(x_i)) \tag{2.28}$$

Thus the sum-of-squares in this case is

$$\sum_{i=1}^{n} ((1 + \operatorname{infl}(x_i))(Y_i - \hat{\mu}(x_i)))^2 \tag{2.29}$$

Clearly the two approaches will be aymptotically equivalent in large samples, when $\operatorname{infl}(x_i)$ is small.

2.7 Bandwidth and Order of Fit

To further understand the issues involved in choosing bandwidth and order of local polynomial fit, it is instructive perform computations on simulated examples. We consider the risk fucntion $R(\hat{\mu}, \mu)$.

Example 2.3

We choose 100 equally spaced design points on $[0, 1]$, and use the mean function

$$\mu(x) = 1 - 48x + 218x^2 - 315x^3 + 145x^4$$

The risk is evaluated for a variety of bandwidths;

```
> x <- seq(0, 1, length.out = 100)
> y <- 1 - x *(48 - x * (218 - x * (315 - x * 145)))
> s <- diff(range(y))/4
> h <- seq(0.1, 0.5, by = 0.05)
> for(i in 1:length(h))
+    print(risk(y~x, ev="data", alpha=c(0,h[i]), sigma=s, deg=2)
```

The **risk()** function should be given a formula of the form y x with x the predictor variable(s), and y a vector of the true mean values $f(x)$. The residual standard deviation is given as the σ argument.

h	Bias	DF	Risk
0.10	0.00043	15.939	15.940
0.15	0.00648	11.170	11.177
0.20	0.04222	8.779	8.828
0.25	0.17346	7.343	7.542
0.30	0.52984	6.386	6.993
0.35	1.31276	5.701	7.205
0.40	2.77708	5.188	8.369
0.45	5.18106	4.789	10.722
0.50	8.71095	4.469	14.446

Table 2.5: Risks of a local quadratic smooth

Table 2.2 shows the result, For small bandwidths, such as $h = 0.1$, the bias is negligible. But the large variance (measured by the degrees of freedom. DF) results in a large risk. As the bandwidth is increased, the variance drops, and the risk attains it's minimum at $h = 0.3$. As the bandwidth increases further, the bias term begins to dominate, and the risk increases.

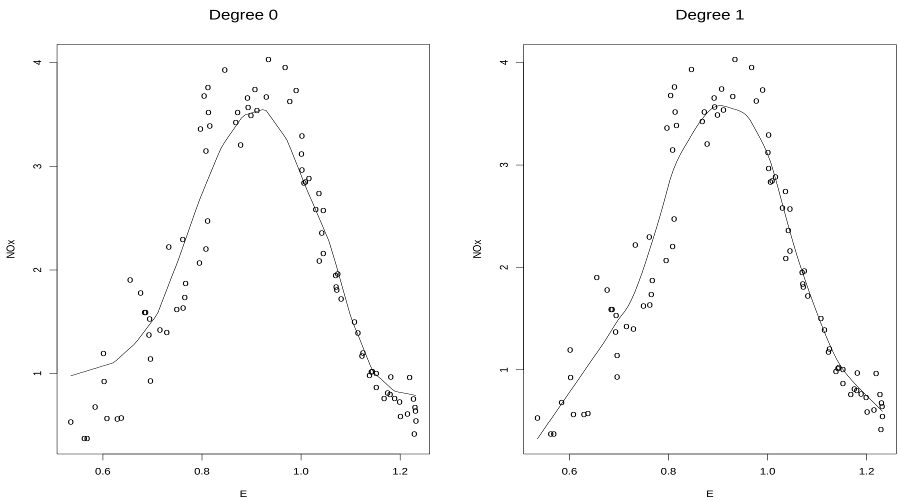

Figure 4: Local constant and local linear fit to the ethanol data

2.8 Exercises

Question 2.1 a) Try to fit the ethanol dataset using local constant and local linear fitting. By varying the bandwidth (using both the fixed and nearest neighbor components, if necessary), can a fit comparable to the local quadratic fit in Figure 2.2 be obtained? Pay attention to both proper modeling of the peak and the leveling off at the boundaries, and to the roughness of the estimates.

```
> for(deg in 0:1){
+       fit <- locfit(NOx~E, data=ethanol, deg=deg, alpha=c(0.30, 0.08))
+       plot(fit,main=paste("Degree", deg), get.data=T)
+       }
```

Local linear and local constant fit can not comparable to local quadratic fit. They can't cover the data of the peak, and also local constant fit doesn't look like that it covers data at the boudary.

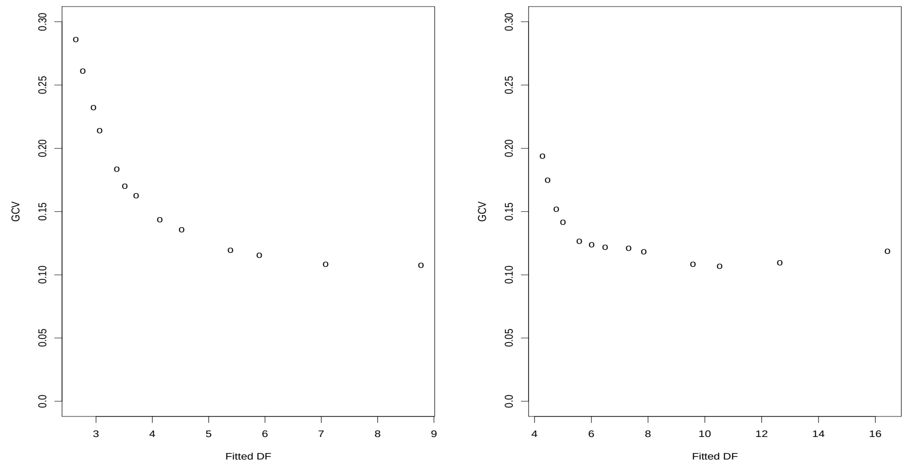

Figure 5: Generalized cross validation scores for the ethanol dataset: local linear fit(left), local quadratic fit(right)

b) Compute the GCV scores for local linear fitting, and compare with the results of local quadratic fitting in Figure 3. Comment on the usefulness of GCV for selecting polynomial order.

```
> alpha <- seq(0.2, 0.8, by=0.05)
> g <- matrix(nrow=length(alpha), ncol=4)
> for(i in 1:length(alpha))
+        g[i,] <- gcv(NOx~E, data=ethanol, deg=1, alpha=alpha[i])
> cbind(alpha, g)
> plot(g[,3],g[,4],ylim=c(0,0.2),xlab="Fitted DF",ylab="GCV")
+}
```

Compared to Quadratic fitting in figure 2.7, GCV from linear fit is totally bigger. This means the quadratic fit is better than the linear fit in the ethanol data set.

Quedtion 2.2 a)Let $\mathbf{V}$ be an invertible $p \times p$ matrix; a a p-vector and λ a constant with $1 + \lambda a^T \mathbf{V}^{-1} a \neq 0$.

Show

$$(\mathbf{V} + \lambda a a^T)^{-1} = \mathbf{V}^{-1} - \frac{\lambda \mathbf{V}^{-1} a a^T \mathbf{V}^{-1}}{1 + \lambda a^T \mathbf{V}^{-1} a} \tag{2.30}$$

Use this result to provide a direct matrix proof of Theorem 2.3.

$$\begin{aligned}
(\mathbf{V} + \lambda a a^T)(\mathbf{V}^{-1} - \frac{\lambda \mathbf{V}^{-1} a a^T \mathbf{V}^{-1}}{1 + \lambda a^T \mathbf{V}^{-1} a}) &= \\
\mathbf{I} + \lambda a a^T \mathbf{V}^{-1} - \frac{1}{1 + \lambda a^T \mathbf{V}^{-1} a} (\lambda (\mathbf{V}\mathbf{V}^{-1}) a a^T \mathbf{V}^{-1} + \lambda^2 a a^T V^{-1} a a^T \mathbf{V}^{-1}) &= \\
\mathbf{I} + \lambda a a^T \mathbf{V}^{-1} - \frac{1}{1 + \lambda a^T \mathbf{V}^{-1} a} \lambda a a^T \mathbf{V}^{-1} 1 + \lambda a a^T \mathbf{V}^{-1} &= \\
\mathbf{I} + \lambda a a^T \mathbf{V}^{-1} - \lambda a a^T \mathbf{V}^{-1} &= \mathbf{I}
\end{aligned}$$

Therefore,

$$(\mathbf{V} + \lambda a a^T)^{-1} = \mathbf{V}^{-1} - \frac{\lambda \mathbf{V}^{-1} a a^T \mathbf{V}^{-1}}{1 + \lambda a^T \mathbf{V}^{-1} a}$$

Proof of Theorem 2.3

$$\hat{\mu}_{-i}(x_i) = {x_i}^T [\mathbf{X}_{(i)}^T W \mathbf{X}_{(\mathbf{i})}] \mathbf{X}_{(i)}^T W_{(i)} \mathbf{Y}_{(\mathbf{i})}$$

Here,

$$\begin{aligned}
(\mathbf{X}_{(i)}^T W_{(i)} \mathbf{X}_{(\mathbf{i})})^{-1} &= (\mathbf{X}^T W \mathbf{X} - w_i x_i {x_i}^T)^{-1} \\
&= (\mathbf{X}^T W \mathbf{X})^{-1} + \frac{w_i (\mathbf{X}^T W \mathbf{X})^{-1} x_i {x_i}^T (\mathbf{X}^T W \mathbf{X})^{-1}}{1 - {x_i}^T (\mathbf{X}^T W \mathbf{X})^{-1} x_i w_i} \\
&= (\mathbf{X}^T W \mathbf{X})^{-1} + \frac{w_i (\mathbf{X}^T W \mathbf{X})^{-1} x_i {x_i}^T (\mathbf{X}^T W \mathbf{X})^{-1}}{1 - \operatorname{infl}(x_i)}
\end{aligned}$$

So,

$$\begin{aligned}
\hat{\mu}_{-i}(x_i) &= {x_i}^T \left((\mathbf{X}^T W \mathbf{X})^{-1} + \tfrac{w_i(\mathbf{X}^T W \mathbf{X})^{-1} x_i {\mathbf{x_i}}^T (\mathbf{X}^T W \mathbf{X})^{-1}}{1- \text{ infl}\,(x_i)} \right) \mathbf{X}_{(i)}^T W_{(i)} \mathbf{Y_{(i)}} \\
&= {x_i}^T (\mathbf{X}^T W \mathbf{X})^{-1} \mathbf{X}_{(i)}^T W_{(i)} \mathbf{Y_{(i)}} + {x_i}^T \frac{w_i(\mathbf{X}^T W \mathbf{X})^{-1} x_i {x_i}^T (\mathbf{X}^T W \mathbf{X})^{-1}}{1- \text{ infl}\,(x_i)} \mathbf{X}_{(i)}^T W_{(i)} \mathbf{Y_{(i)}} \\
&= {x_i}^T (\mathbf{X}^T W \mathbf{X})^{-1} \mathbf{X}_{(i)}^T W_{(i)} \mathbf{Y_{(i)}} + \frac{w_i {x_i}^T (\mathbf{X}^T W \mathbf{X})^{-1} x_i {x_i}^T (\mathbf{X}^T W \mathbf{X})^{-1} \mathbf{X}_{(i)}^T W_{(i)} \mathbf{Y_{(i)}}}{1- \text{ infl}\,(x_i)} \\
&= \frac{{x_i}^T (\mathbf{X}^T W \mathbf{X})^{-1} \mathbf{X}_{(i)}^T W_{(i)} \mathbf{Y_{(i)}}}{1- \text{ infl}\,(x_i)}
\end{aligned}$$

Since $\mathbf{X}^T \mathbf{Y} = \mathbf{X}_{(i)}^T \mathbf{Y_{(i)}} + w_i x_i Y_i$,

$$\begin{aligned}
\hat{\mu}_{-i}(x_i) &= \frac{{x_i}^T (\mathbf{X}^T W \mathbf{X})^{-1} (\mathbf{X}^T \mathbf{Y} - x_i Y_i)}{1- \text{ infl}\,(x_i)} \\
&= \frac{{x_i}^T (\mathbf{X}^T W \mathbf{X})^{-1} \mathbf{X}^T W \mathbf{Y} + {x_i}^T (\mathbf{X}^T W \mathbf{X})^{-1} w_i x_i Y_i}{i- \text{ infl}\,(x_i)} \\
&= \frac{\hat{\mu}(x_i) - \text{ infl}\,(x_i) Y_i}{1- \text{ infl}\,(x_i)}
\end{aligned}$$

b) Show

$$a^T (\mathbf{V} + \lambda a a^T)^{-1} a = \frac{a^T \mathbf{V}^{-1} a}{1 + \lambda a^T \mathbf{V}^{-1} a}$$

Hence prove $\text{infl}\,(x_i) \leq 1$

Using the result of a)

$$\begin{aligned}
a^T (\mathbf{V} + \lambda a a^T)^{-1} a &= a^T \mathbf{V}^{-1} a - \frac{\lambda a^T \mathbf{V}^{-1} a a^T \mathbf{V}^{-1} a}{1 + \lambda a^T \mathbf{V}^{-1} a} \\
&= \frac{a^T \mathbf{V}^{-1} a}{1 + \lambda a^T \mathbf{V}^{-1} a}
\end{aligned}$$

Proof of $\text{infl}\,(x_i) \leq 1$

$$\text{infl}\,(x_i) = \mathbf{x_i}^{\mathbf{T}} ((\mathbf{X}_{(i)}^T W_{(i)} \mathbf{X_{(i)}}) - w_i \mathbf{x_i} {x_i}^T)^{-1} \mathbf{x_i}^{\mathbf{T}}$$

Since $w_i = 1$ at the observation point,

$$\begin{aligned} \text{infl}\,(x_i) &= \frac{x_i{}^T(\mathbf{X}_{(i)}^T W_{(i)} \mathbf{X}_{(\mathbf{i})})^{-1} x_i}{1 + x_i{}^T(\mathbf{X}_{(i)}^T W_{(i)} x_i \mathbf{X}_{(\mathbf{i})})^{-1} x_i} \\ &= \frac{\text{infl}_{(i)}(x_i)}{1 + \ \text{infl}_{(i)}(x_i)} \end{aligned}$$

From (2.12), $\text{infl}\,(x) \geq 0$, so (2.13) is proved.

c) Suppose $\mathbf{V}$ and $\sum$ are non-negative definite. Show

$$a^T(V + \sum)^{-1} \geq a^T V^{-1} a$$

As $\sum$ is nonegative definite, $\sum$ can be written as

$$\sum = \lambda a a^T \geq 0$$

Then,

$$\begin{aligned} a^T(V + \lambda a a^T)^{-1} a &= \frac{a^T V^{-1} a}{1 + \lambda a^T V^{-1} a} \\ &\geq a^T V^{-1} a \end{aligned}$$

Question 2.3 a) Let $\mathbf{V}$, a and λ as in the previous exercise, with $d = \lambda - a^T\mathbf{V}^{-1}a \neq 0$.

Show

$$\begin{pmatrix} \mathbf{V} & a \\ a^T & \lambda \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{V}^{-1} + \frac{1}{d}\mathbf{V}^{-1}aa^T\mathbf{V}^{-1} & -\frac{1}{d}\mathbf{V}^{-1}a \\ -\frac{1}{d}a^T\mathbf{V}^{-1} & \frac{1}{d} \end{pmatrix}$$

Using $d = \lambda - a^T\mathbf{V}a$

$$\begin{aligned}
\begin{pmatrix} \mathbf{V} & a \\ a^T & \lambda \end{pmatrix} \begin{pmatrix} \mathbf{V}^{-1} + \frac{1}{d}\mathbf{V}^{-1}aa^T\mathbf{V}^{-1} & -\frac{1}{d}\mathbf{V}^{-1}a \\ -\frac{1}{d}a^T\mathbf{V}^{-1} & \frac{1}{d} \end{pmatrix} &= \\
\begin{pmatrix} \mathbf{V}\mathbf{V}^{-1} + \frac{1}{d}(\mathbf{V}\mathbf{V}^{-1})aa^T\mathbf{V}^{-1} - \frac{1}{d}aa^T\mathbf{V}^{-1} & -\frac{1}{d}\mathbf{V}\mathbf{V}^{-1}a + \frac{a}{d} \\ a^T\mathbf{V}^{-1} + \frac{1}{d}a^T\mathbf{V}^{-1}aa^T\mathbf{V}^{-1} - \frac{1}{d}\lambda a^T\mathbf{V}^{-1} & -\frac{a^T}{d}\mathbf{V}^{-1}a + \frac{\lambda}{d} \end{pmatrix} &= \\
\begin{pmatrix} \mathbf{I} & 0 \\ \frac{1}{d}(d + a^T\mathbf{V}^{-1} - \lambda)a^T\mathbf{V}^{-1} & \frac{1}{d}(\lambda - a^T\mathbf{V}^{-1}a) \end{pmatrix} &= \\
\begin{pmatrix} \mathbf{I} & 0 \\ 0 & 1 \end{pmatrix} &= \mathbf{I}
\end{aligned}$$

So,

$$\begin{pmatrix} \mathbf{V} & a \\ a^T & \lambda \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{V}^{-1} + \frac{1}{d}\mathbf{V}^{-1}aa^T\mathbf{V}^{-1} & -\frac{1}{d}\mathbf{V}^{-1}a \\ -\frac{1}{d}a^T\mathbf{V}^{-1} & \frac{1}{d} \end{pmatrix}$$

b) Show the influence function innfl(x) increases as the degree of the local polynomial increases.

Let $\mathbf{X}$ be the $n \times (P+1)$ design matrix, and Let $\mathbf{X}'$ be the $n \times (P+2)$ design matrix. So, $\mathbf{X}'$ can be written as $\mathbf{X}' = (\mathbf{X}\mathbf{x})$ Then,

$$\begin{aligned}
\mathbf{X}'^T W \mathbf{X}' &= \begin{pmatrix} \mathbf{X}^T \\ \mathbf{x}^T \end{pmatrix} W(\mathbf{X} \ \mathbf{x}) \\
&= \begin{pmatrix} \mathbf{X}^T W \mathbf{X} & \mathbf{X}^T W \mathbf{x} \\ \mathbf{x}^T W \mathbf{X} & \mathbf{x}^T W \mathbf{x} \end{pmatrix}
\end{aligned}$$

So,

$$\begin{aligned}
\text{infl}_{P+2}(x) &= {e_1}^T(\mathbf{X}'^T W\ \mathbf{X}')^{-1} e_1 \\
&= {e_1}^T \begin{pmatrix} \mathbf{X}^T W\ \mathbf{X} & \mathbf{X}^T W\ \mathbf{x} \\ \mathbf{x}^T W\ \mathbf{X} & w\ \mathbf{x}^T W\ \mathbf{x} \end{pmatrix}^{-1} e_1
\end{aligned}$$

Here, we can apply $V = \mathbf{X}^T W\ \mathbf{X}$, $a = \mathbf{X}^T W\ \mathbf{x}$, $\lambda = \mathbf{x}^T W\ \mathbf{x}$ to the result of a). Then,

$$\begin{aligned}
\text{infl}_{P+2}(x) &= {e_1}^T \mathbf{V}_1 e_1 + e_1 \left(\tfrac{1}{d}\mathbf{V}^{-1} a a^T \mathbf{V}^{-1} - \tfrac{1}{d}\mathbf{V}^{-1} a - \tfrac{1}{d} a^T \mathbf{V}^{-1} + \tfrac{1}{d}\right) \\
&= \text{infl}_{p+1}(x) + {e_1}^T \left(\tfrac{1}{d}\mathbf{V}^{-1} a (a^T \mathbf{V}^{-1} - 1) + \tfrac{1}{d}(1 - a^T \mathbf{V}^{-1})\right) e_1 \\
&= \text{infl}_{p+1} + \frac{1}{d} e_1 (\mathbf{V}^{-1} a - 1)(a^T \mathbf{V}^{-1} - 1) e
\end{aligned}$$

We can prove this problem only if we can prove $d = \lambda - a^T \mathbf{V} a \geq 0$.

$$\begin{aligned}
d = \lambda - a^T \mathbf{V}^{-1} a &= \mathbf{x}^T W\ \mathbf{x} - \mathbf{x}^T W X (\mathbf{X}^T W\ \mathbf{X})^{-1}\ \mathbf{X}^T W\ \mathbf{x} \\
&= \mathbf{x}^T W (\mathbf{I} W^{-1} - \mathbf{X} (\mathbf{X}^T W\ \mathbf{X})^{-1}\ \mathbf{X}^T W) W\ \mathbf{x}
\end{aligned}$$

As $(\mathbf{I} - \mathbf{X}(\mathbf{X}^T W\ \mathbf{X})^{-1}\ \mathbf{X})$ is positive semi definite and $W^{-1}\mathbf{I} > \mathbf{I}$, we can say $d > 0$.

c) Provide an example to show the local linear fitting can be less variable than local constant fitting.

We use a rubber data set.

```
attach(rubber)
Degree=0


fit <- locfit(abrasion.loss~hardness, data=rubber,deg=0,alpha=c(0.3,5))
   newd <- cbind(rubber, resid=residuals(fit))
   fitr <- locfit(resid~hardness, data=newd,alpha=c(0.3,5))
   plot(fitr, main=paste("Deg=", 0), get.data=T )
```

```
   abline(h=0, lty=2)
   res <- residuals(fit)
   > lapply(list(res^2), FUN=sum)
  [[1]]:
  [1] 99399.29
```

```
Degree=1
fit <- locfit(abrasion.loss~hardness, data=rubber, deg=1,alpha=c(0.3,5))
   newd <- cbind(rubber, resid=residuals(fit))
   fitr <- locfit(resid~hardness, data=newd, alpha=c(0.3,5))
   plot(fitr, main=paste("Deg=",1), get.data=T )
   abline(h=0, lty=2)
   res <- residuals(fit)
   > lapply(list(res^2), FUN=sum)
[[1]]:
[1] 90892.96
```

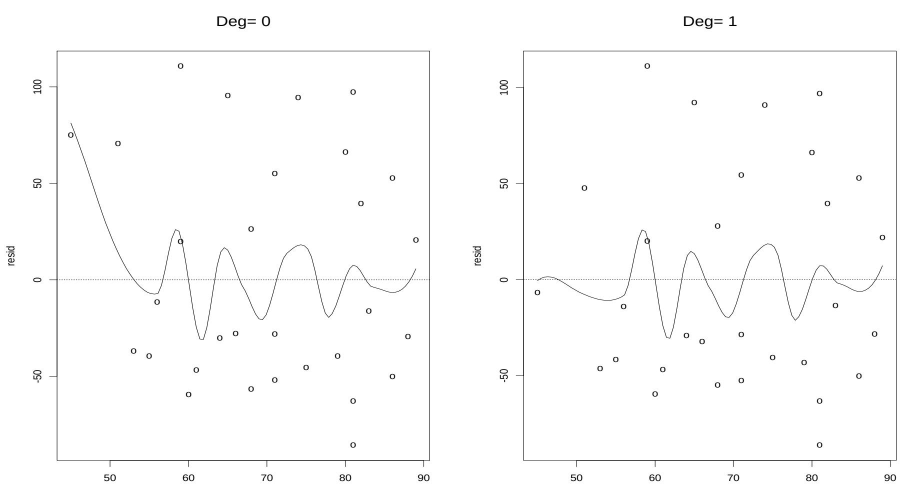

Figure 6: Residuals for local constant fit(left) and residuals for local linear fit(right)

Conclusion

The fact that $E({\epsilon_i}^2) = 9939.92$ for local constant fit and $E({\epsilon_i}^2) = 90892.96$ tells us that local linear fitting can be less variable than local constant fitting.

Question 2.4 Fitting a local constantt polynomial produces the Nadaraya-Watson kernel estimate.

$$\hat{\mu}(x) = \frac{\sum_{i=1}^{n} W_i(x)}{\sum_{i=1}^{n} W_i(x)} Y_i$$

a) Write down the variance of this estimate, assuming the Y_i are independent and have constant variance.

$$\begin{aligned}
\text{Var}\,(\hat{\mu}(x)) &= \text{Var}\,(\frac{\sum_{i=1}^{n} W_i(x) Y_i}{\sum_{i=1}^{n} W_i(x)}) \\
&= \frac{\sum_{i=1}^{n}\{{W_i}^2(x)\}}{\{\sum_{i=1}^{n} W_i(x)\}^2}\ \text{Var}\,(Y_i)
\end{aligned}$$

b) Suppose the weight function $W(u)$ is symmetric., non-negative and $\frac{uW'(u)}{W(u)}$ is monotone decreasing on $[0,\infty)$. Show for fixed x the variance is decreasing function of the bandwidth h. Show all six weight functions supported in LOCFIT satisfy this condition.

Let's show $\frac{\partial}{\partial h} V(\hat{\mu}(x)) < 0$. First of all

$$u_i = \frac{x_i - x}{h}$$

$$\begin{aligned}
u_i' &= -\frac{x_i - x}{h^2} \\
&= -\frac{u_i}{h}
\end{aligned}$$

$$\begin{aligned}
V'(\hat{\mu}(x)) &= \frac{\left(\frac{2}{h}\sum_{i=1}^{n} W(u_i) \sum_{i=1}^{n} W'(u_i)u_i \sum_{i=1}^{n} W^2(u_i) - \frac{2}{h}\sum_{i=1}^{n} W(u_i)W'(u_i)u_i (\sum_{i=1}^{n} W(u_i))^2\right)}{(\sum_{i=1}^{n} W_i(x))^2} \\
&= \frac{\left(\frac{2}{h}\sum_{i=1}^{n} W(u_i) \sum_{i=1}^{n} W(u_i) \frac{\sum_{i=1}^{n} W'(u_i)u_i}{W(u_i)} \sum_{i=1}^{n} W^2(u_i) - \frac{2}{h}\sum_{i=1}^{n} W^2(u_i)\frac{W'(u_i)u_i}{W(u_i)} (\sum_{i=1}^{n} W(u_i))^2\right)}{(\sum_{i=1}^{n} W_i(x))^2} \\
&= \frac{\sum_{i=1}^{n} W(u_i)\frac{2}{h}\left(\sum_{i=1}^{n} W(u_i)\frac{W'(u_i)u_i}{W(u_i)} \sum_{i=1}^{n} W^2(u_i) - \sum_{i=1}^{n} W^2(u_i)\frac{W'(u_i)u_i}{W(u_i)} (\sum_{i=1}^{n} W(u_i))\right)}{(\sum_{i=1}^{n} W_i(x))^2}
\end{aligned}$$

Let $F(u)$ be the function as follows. $F(u) = \frac{W'(u)u}{W(u)}$ As $F(0) = 0$ and also $F(u)$ is a decreasing function, F(u) is always below 0. So,

$$\begin{aligned}
\sum_{i=1}^{n} W(u_i)\frac{W'(u_i)u_i}{W(u_i)}\sum_{i=1}^{n} W^2(u_i) - \sum_{i=1}^{n} W^2(u_i)\frac{W'(u_i)u_i}{W(u_i)}\sum_{i=1}^{n} W(u_i) &= \\
\sum W(u_i)F(u_i)\sum W^2(u_i) - \sum W^2(u_i)F(u_i)\sum W(u_i) &\leq \sum W(u_i)F(u_i) - \sum W^2(u_i)F(u_i) \\
&\leq 0
\end{aligned}$$

Because $\sum W^2(u_i) \leq \sum W(u_i)$

Rectangular weight function $W(x) = 1, |x| < 1$

This is apparently symmetric and $\frac{xW'(x)}{W(x)} = 0$ satisfies the decreasing condition.

Triangular weight function $W(x) = 1 - |x|, |x| < 1$

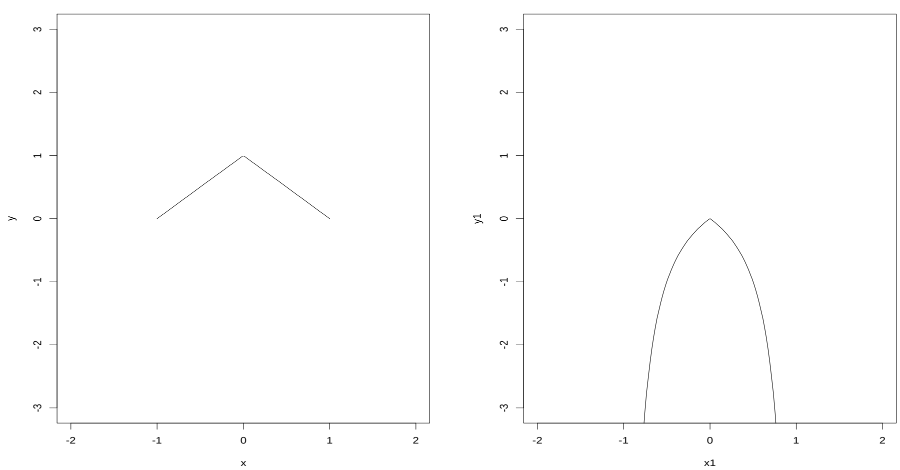

Figure 7: Triangular weight function(left) and its $\frac{xW'(x)}{W(x)}$ function

Epanechnikov weight function $W(x) = 1 - x^2, |x| < 1$

Bisquare weight function $W(x) = (1 - x^2)^2, |x| < 1$

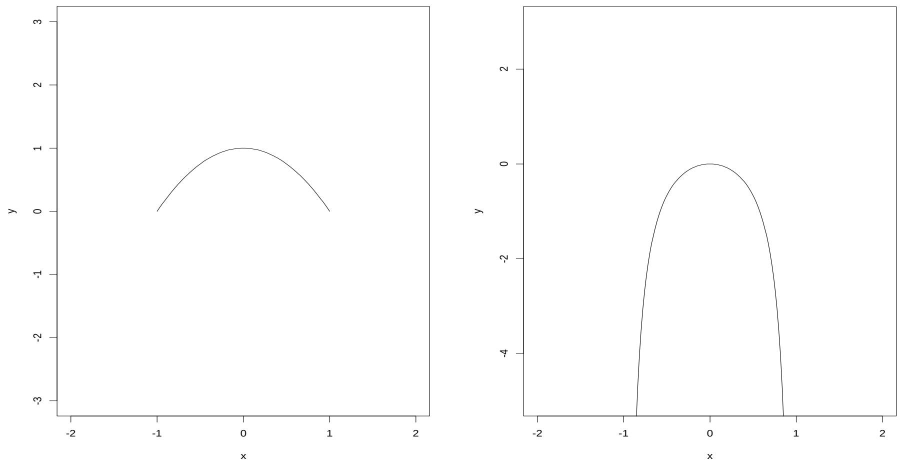

Figure 8: Epanechnikov weight function(left) and its $\frac{xW'(x)}{W(x)}$ function

Tricube weight function $W(x) = (1 - |x|^3)^3, |x| < 1$

Triweight weight function $(W(x) = 1 - |x|^2)^3, |x| < 1$

Gaussian weight function $(W(x) = exp((-2.5x)^2/2)$

All these weight functions satisfy the condition.

c) Relaxing the conditions on the weight function, provide an example to show the variance can increase as the bandwidth h increaseas.

```
w <- seq(0.01,0.3,by=.01)

h=4
fit <- locfit(abrasion.loss~hardness, data=rubber,weights=w, deg=2,alpha=c(0.05,4))
```

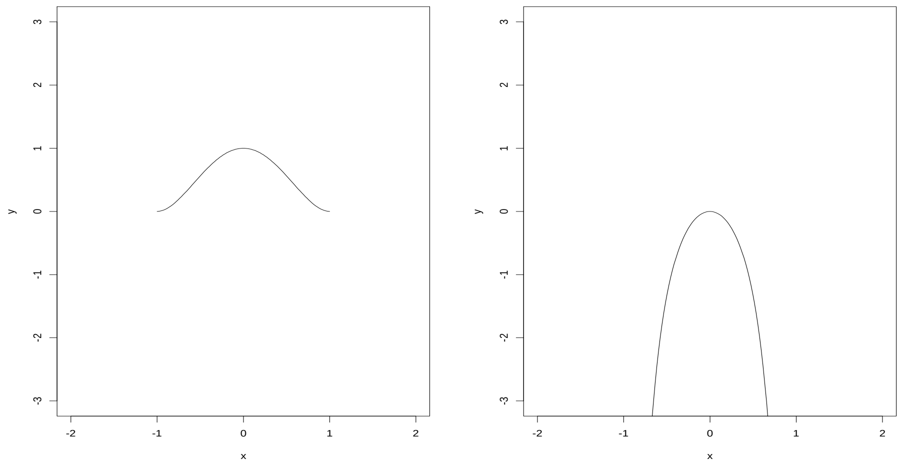

Figure 9: Bisquare weight function(left) and its $\frac{xW'(x)}{W(x)}$ function

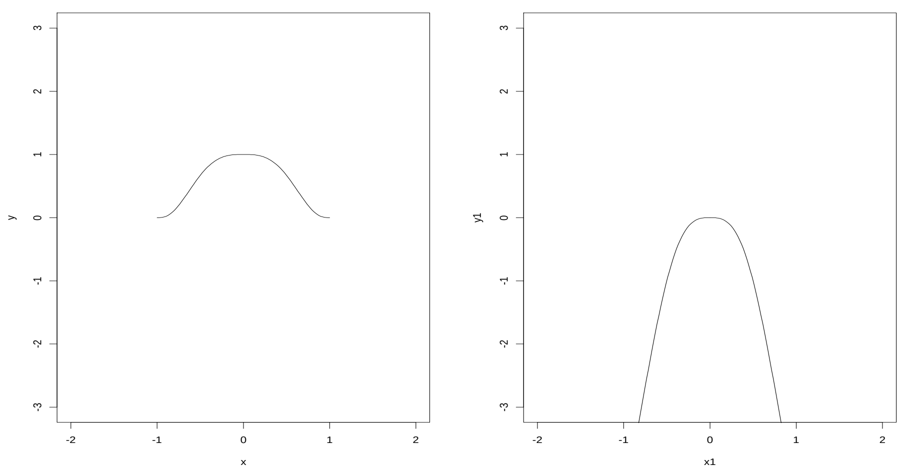

Figure 10: Tricube weight function(left) and its $\frac{xW'(x)}{W(x)}$ function

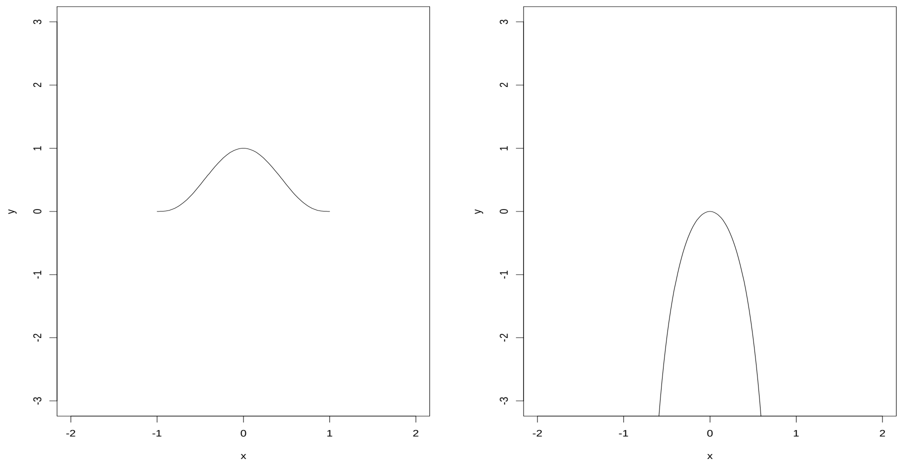

Figure 11: Triweight weight function(left) and its $\frac{xW'(x)}{W(x)}$ function

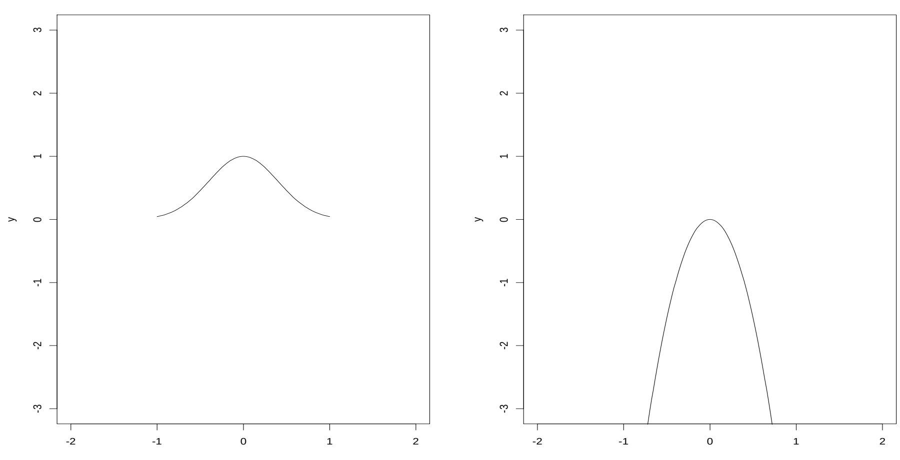

Figure 12: Gaussian weight function(left) and its $\frac{xW'(x)}{W(x)}$ function

```
va <- fitted(fit, what="nlx")
apply(matrix(va^2),2,"sum")
[1] 222.7865
```

```
h=5
fit <- locfit(abrasion.loss~hardness, data=rubber,weights= w, deg=2,alpha=c(0.05,5))
va <- fitted(fit, what="nlx")
apply(matrix(va^2),2,"sum")
[1] 194.3549
```

```
h=6
fit <- locfit(abrasion.loss~hardness, data=rubber, deg=0,alpha=c(0.05,6))
va <- fitted(fit, what="nlx")
apply(matrix(va^2),2,"sum")
[1] 178.1387
```

As the bandwidth becomes bigger, the variance can also become bigger.

Question 2.5 *Double smoothing.* The local regression estimate $\hat{\mu} = \mathbf{L}Y$ has bias $b = E(\hat{\mu}) - \mu = -(\mathbf{I} - \mathbf{L}\mu$. Double smoothing estimates the bias by $\hat{b} = -(\mathbf{I} - \mathbf{L})\hat{\mu}$, producing the bias corrected estimate $\hat{\hat{\mu}} = \hat{\mu} - \hat{b} = (2\mathbf{I} - \mathbf{L})\mathbf{L}Y$. Suppose the design is equally spaced with $x_i = i; i = ..., -2, -1, 0, 1, 2, ...$ the doubly infinite sequence avoids boundary effects.

a) Suppose the smoother is a moving average (local constant) fit with bandwidth h;

$$\hat{\mu}(i) = \frac{1}{2h+1} \sum_{j=-h}^{h} Y_{i+j}$$

Show

$$\hat{\hat{\mu}}(i) = \frac{1}{2h} \sum_{j=-2h}^{2} h\left(2\mathbf{I}(|j| \leq h) - 1 + \tfrac{|j|}{2h+1})\right) Y_{i+j}$$

$$\begin{aligned}
\mathbf{LL}Y &= (\frac{1}{2h+1})^2 \sum_{j=-h}^{h} Y_{i-h+j} + \cdots + \sum_{j=-h}^{h} Y_i + \cdots \sum_{j=-h}^{n} Y_{i+h+j} \\
&= (\frac{1}{2h+1})^2 (\sum_{j=-h}^{h} (Y_{i-h+j} + Y_{i+h+j}) + (\sum_{j=-h}^{h} (Y_{i-h+j+1} + Y_{i+h+j-1}) + \cdots + \sum_{j=-h}^{h} Y_i) \\
&= (\frac{1}{2h+1})^2 ((Y_{i-2h} + \cdots + 2Y_i + \cdots + Y_{i+2h}) \\
&+ (Y_{i-2h+1} + \cdots + 2Y_{i-1} + 2Y_i + 2Y_{i+1} + \cdots + Y_{i+2h-1}) \\
&+ \cdots + (2Y_{i-h+1} + \cdots + 2Y_{i+h-1}) + \cdots + \sum_{j=-h}^{h} Y_i) \\
&= (\frac{1}{2h+1})^2 (Y_{i-2h} + 2Y_{i-2h+1} + \cdots + 2h + 1Y_i + \cdots + Y_{i+2h}) \\
&= \frac{1}{2h+1}(\frac{1}{2h+1}Y_{i-2h} + \frac{2}{2h+1} + \cdots + \frac{1}{2h+1}Y_{i+2h}) \\
&= \frac{1}{2h} \sum_{j=-2h}^{2h} (1 - \frac{|j|}{2h+1})Y_{i+j}
\end{aligned}$$

Therefore,

$$\begin{aligned}
\hat{\hat{\mu}}(i) &= (2\mathbf{I} - \mathbf{L})\mathbf{L}Y \\
&= \frac{1}{2h}\sum_{j=-2h}^{2} h\left(2\mathbf{I}(|j| \le h) - 1 + \tfrac{|j|}{2h+1})\right) Y_{i+j}
\end{aligned}$$

b) Show the weight diagram for $\hat{\hat{\mu}}(i)$ is quadratic reproducing, and hence by henderson's theorem $\hat{\hat{\mu}}(i)$ is a local quadratic smooth.

Determine l_j as follows.

$$l_j = \frac{1}{2h+1}\sum_{j=-2h}^{2h}\left(2\mathbf{I}(|j| \le h) - 1 + \tfrac{|j|}{2h+1}\right)$$

We can show weight diagram for $\hat{\hat{\mu}}$ is quadratic reproducing if we can verify $\sum_{j=-2h}^{2h} l_j = 1$, $\sum_{j=-2h}^{2h} jl_j = 0$, $\sum_{j=-2h}^{2h} j^2 l_j = 0$.

$$\begin{aligned}
\sum_{j=-2h}^{2h} l_j &= \frac{1}{2h+1}\left(\sum_{j=-h}^{h} 2 - \sum_{j=-2h}^{2h} + \tfrac{2}{2h+1}\sum_{j=1}^{2h} j\right) \\
&= \frac{1}{2h+1}\left(2(2h+1) - (2\times 2h+1) + \tfrac{2}{h+1}\tfrac{(1+2h)2h}{2h+1}\right) \\
&= \frac{1}{2h}(4h - 2 - 4h - 1 + 2h) \\
&= 1
\end{aligned}$$

$$\begin{aligned}
\sum_{j=-2h}^{2h} jl_j &= \frac{1}{2h+1}\left(\sum_{j=-h}^{h} 2j - \sum_{j=-2h}^{j=2h} j + \tfrac{2}{2h+1}\sum_{j=-2h}^{2h} j^2\right) \\
&= 0
\end{aligned}$$

Because all the elements of this function are symmetry.

$$\sum_{j=-2h}^{2h} j^2 l_j = \frac{1}{2h+1}\left(2\sum_{j=0}^{h} 2j^2 - 2\sum_{j=0}^{j=2h} j^2 + \tfrac{2}{2h+1}\sum_{j=0}^{2h} j^3\right)$$

$$\begin{aligned}
\sum_{j=0}^{h} 2j^2 &= 2\frac{1}{6}h(h+1)(2h+1) \\
&= \frac{1}{6}(4h^3 + 6h^2 + 2h)
\end{aligned}$$

$$\begin{aligned}\sum_{j=0}^{j=2h} j^2 &= \frac{1}{6}2h(2h+1)(4h+1) \\ &= \frac{1}{6}(16h^3+12h^2+2h)\end{aligned}$$

$$\begin{aligned}\frac{1}{2h+1}\sum_{j=0}^{2h} j^3 &= \frac{(\frac{1}{2}2h(2h+1))^2}{2h+1} \\ &= 2h^3+h^2\end{aligned}$$

So,

$$\sum_{j=-2h}^{2h} j^2 l_j = 0$$

Weight Diagram

$$\frac{1}{(2h+1)^2}(-1,-2,\cdots,-h,3h+1,3h,\cdots,2h+2,2h+1,2h+2,\cdots,3h,3h+1,-h,\cdots,-2,-1)$$

I think it's sensible because there are some big jumps. A little difference cause a big difference.

Question 2.6 Some visual experiments:

a) Construct a dataset with mean function having flat and steep regions. For example, let x_i be uniform on $[-5, 5]$ and $Y_i = \Phi(x_i) + \epsilon_i$, where N$(x)$ is the standard normal distribution function and ϵ_i is normally distributed with $\sigma = 0.1$. Plot the dataset. Does the residual variance look constant?

Splus command

```
X <- runif(100, -5, 5)
Y <- dnorm(X, mean=0, sd=1)+rnorm(100, 0, 0.1)
fit <- locfit(Y~X, deg=2, alpha=c(0.2, 0.5))
plot(fit, get.data=T)
```

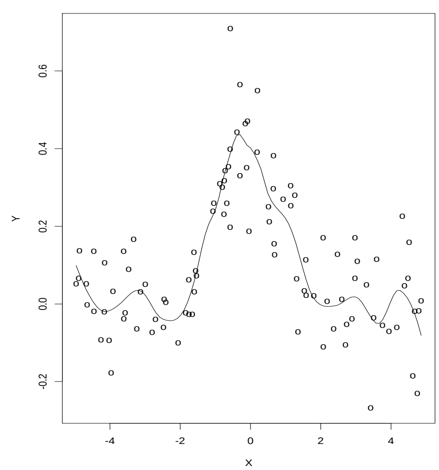

Figure 13: Locfit for Q2-6 a)

```
res <- residuals(fit)
fitr <- locfit(res~X, deg=1, alpha=c(0.2, 0.5))
plot(fitr, get.data=T)
abline(h=0, lty=2)
```

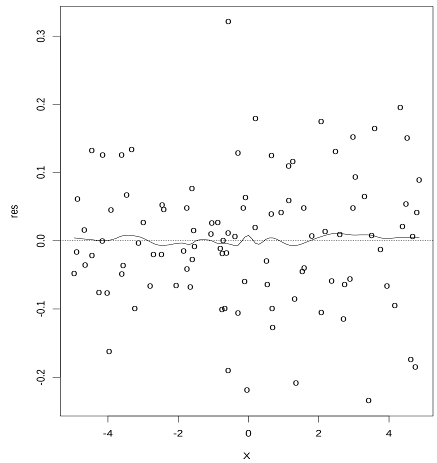

Figure 14: Residual of the fit

Conclusion

The residual variance look constant.

b) Construct a non-uniform predictor variable. For example, `x <- sqrt(unif(100)`. Generate an independent N(0,1) response variable, `y <- rnorm(100)`. Plot the data. Does the residual variance look constant? This experiment may take two or three attempts; eventually, large residuals in the high density region should be distracting.

```
x <-sqrt(runif(100))
y <-rnorm(100)
fit <-locfit(y~x, deg=1, alpha=c(0.2, 0.5))
plot(fit, get.data=T)
res <- residuals(fit)
fitr <- locfit(res~x, deg=1, alpha=c(0.2, 0.5))
plot(fitr, get.data=T)
abline(h=0, lty=2)
```

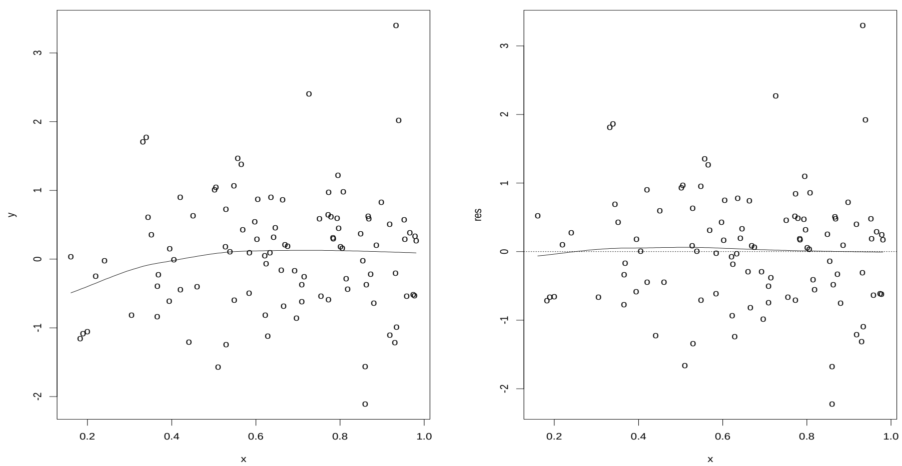

Figure 15: Experiment 1

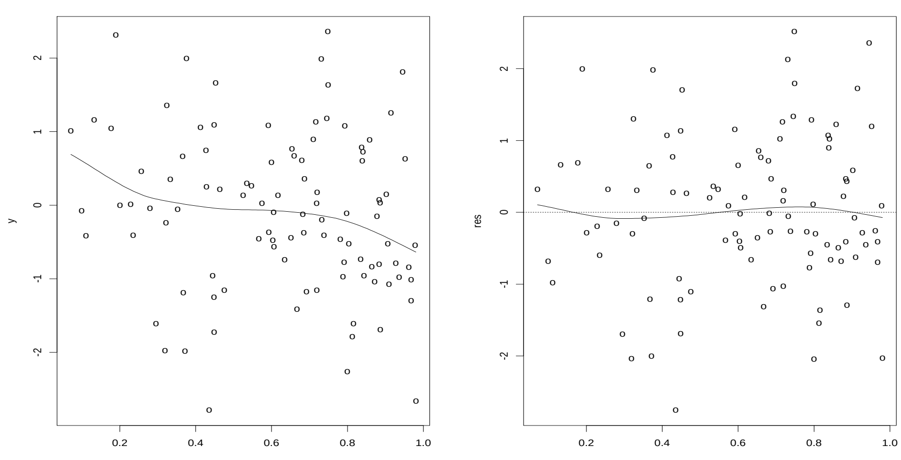

Figure 16: Experiment 2

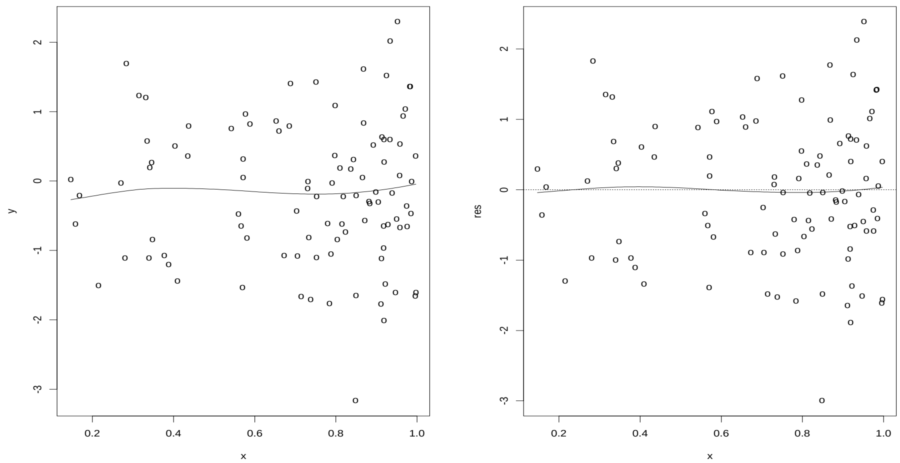

Figure 17: Experiment 3

Conclusion

Three experiments show the residual variance doesn't look constant (fanning out) , this is because of the high density region. So, eventually, large residuals in the high density region should be distracting.

c)The dataset is subject to both the previous problems of this exercise. Consturuct appropriate residual plots to further investigate the residual variance of this dataset.

```
h <- c(1, 2, 5, 10)
for(i in 1:length(h)){
 fit <- locfit(Y~X, alpha=c(0, h[i]))
 res <- residuals(fit)
 fitr <- locfit(res~X, alpha=c(0.2, 1))
 plot(fitr, main=paste("h=", h[i]), get.data=T)
 abline(h=0, lty=2)
 }
```

Conclusion

Figure 2.8 and 2.8 show that the bandwidth becomes bigger, the variance of figure 2.8 shows more nonhomogeneous trend than the varianceof figure 2.8. This is because in the nonhomogeneous data, amount of data in one bandwidth doesn't change much even if the bandwidth becomes bigger. On the contrary, in the homogeneous data, amount of data in one bandwidth changes much and this causes the big difference of variances among each bandwidth.

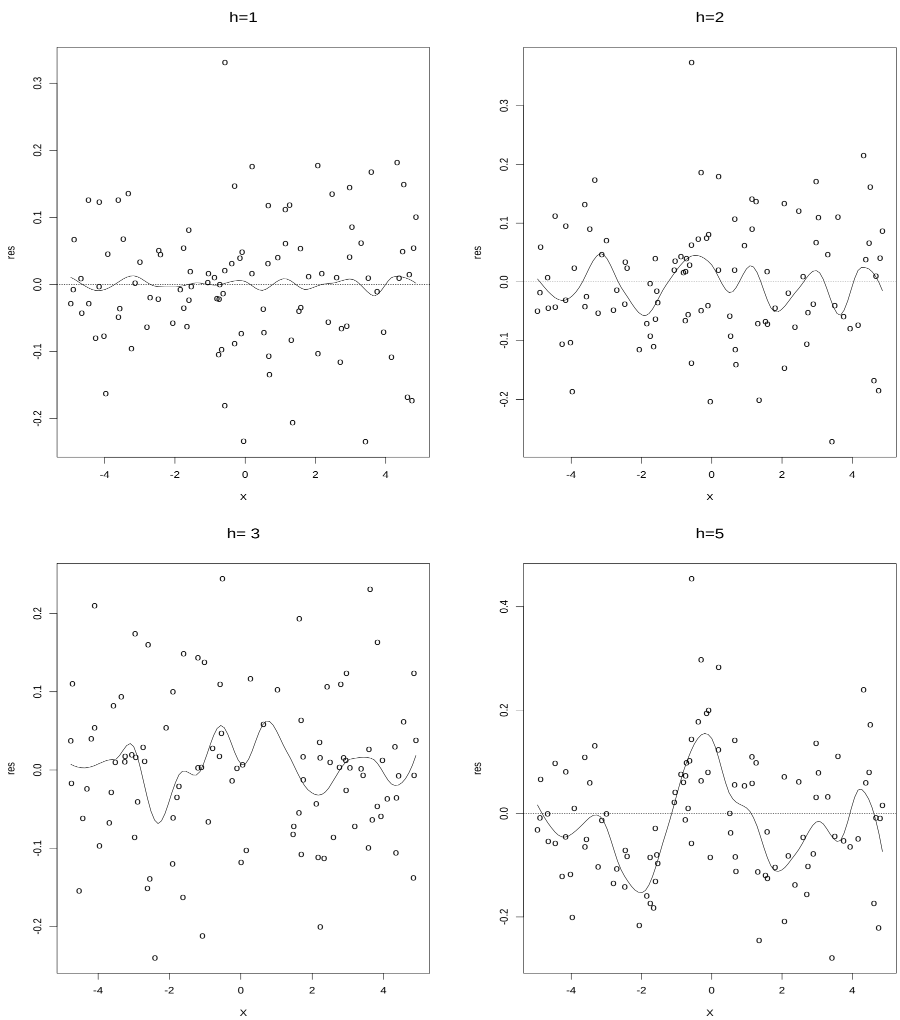

Figure 18: Question 2-6 c) for Q2-6 a))

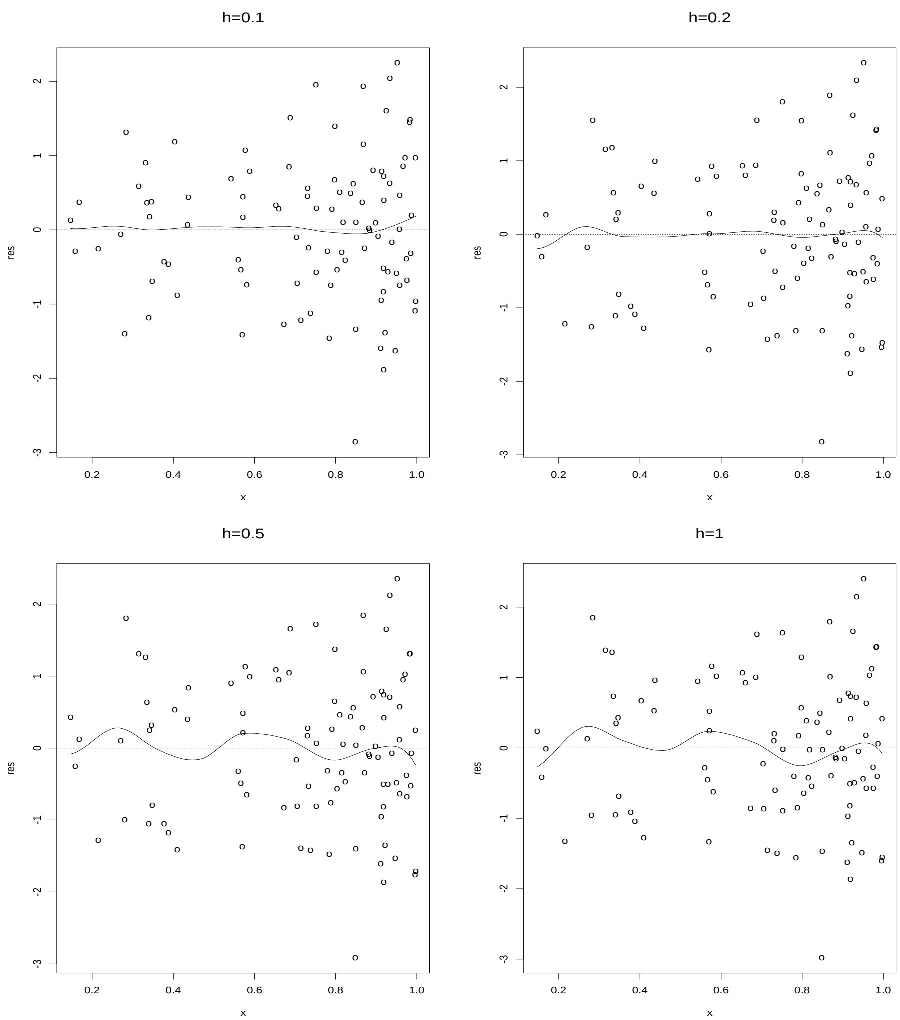

Figure 19: Question 2-6 c) for Q2-6 b)

Question 2.7 (Estimation under the loss function)

a) Suppose $X \sim \mathrm{N}(\mu, \sigma^2)$. Show

$$\mathbf{E}|X| = \mu\left(\Phi(\tfrac{\mu}{\sigma}) - \Phi(-\tfrac{\mu}{\sigma})\right) + 2\sigma\phi(\frac{\mu}{\sigma})$$

Here, $\phi()$ and $\Phi()$ denote the standard normal density and distribution function.

$$\begin{aligned}
E|X| &= \int_{-\infty}^{\infty} \frac{|x|}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \\
&= \int_{-\infty}^{\infty} (\frac{(|x|-\mu)\sigma}{\sigma} + \mu) \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{-(x-\mu)^2}{2\sigma^2}} dx \\
&= \int_{-\infty}^{\infty} \frac{|x|-\mu}{2\sigma^2} \frac{1}{\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} dx + \mu \int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} dx \\
&= 2\int_{0}^{\infty} \frac{x-\mu}{\sigma} \frac{1}{\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} dx + \mu \int_{-\infty}^{0} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} dx + \mu \int_{0}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} dx \\
&= 2\sigma \int_{-\frac{\mu}{\sigma}}^{\infty} \frac{z}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz + \mu \int_{-\infty}^{-\frac{\mu}{\sigma}} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz + \mu \int_{-\frac{\mu}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{-z^2}{2}} dz \\
&= 2\sigma \int_{-\infty}^{\frac{\mu}{\sigma}} -\frac{z}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz + \mu \int_{-\infty}^{-\frac{\mu}{\sigma}} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz - \mu \int_{-\infty}^{\frac{\mu}{\sigma}} \frac{1}{\sqrt{2\pi}} e^{\frac{-z^2}{2}} dz \\
&= \mu\left(\Phi(\tfrac{\mu}{\sigma}) - \Phi(-\tfrac{\mu}{\sigma})\right) + 2\sigma\phi(\frac{\mu}{\sigma})
\end{aligned}$$

b) Consider the risk function $R_1(\hat{\mu}, \mu) = n^{-1}\sum_{i=1}^{n} |\hat{\mu}(x_i) - \mu(x_i)|$. Derive an explicit expression for $R_1(\hat{\mu}, \mu)$ in terms of bias and variance.

As $\hat{\mu}(x_i) - \mu(x_i) \sim N(\beta, \hat{\sigma}^2)$, apply the result of a) to this question.

$$\begin{aligned}
R_1(\hat{\mu}, \mu) &= n^{-1}\sum_{i=1}^{n} |\hat{\mu}(x_i) - \mu(x_i)| \\
&= E|N(\beta, \hat{\sigma}^2)| \\
&= \beta\left(\Phi(\tfrac{\beta}{\hat{\sigma}}) - \Phi(-\tfrac{\beta}{\hat{\sigma}})\right) + 2\sigma\phi(\frac{\mu}{\hat{\sigma}})
\end{aligned}$$

c) Write a `risk1()` function similar to `risk()` in section 2.7, to compute $R_1(\hat{\mu}, \mu)$. Compute risks for several bandwidths.

```
risk1 <- function(..., sigma=1){
m <- match.call()
m[[1]] <- as.name("locfit")
m$sigma <- NULL
fit <- eval(m,sys.parent())
res <- residuals(fit)
res1 <-list(abs(res))
res2 <-lapply(res1,FUN=sum)
res <- c(unlist(res2)/length(res), fit$dp["t1"])
va <- fitted(fit, what="nlx")
sd <- sigma * sqrt(unlist(lapply(list((va)^2),FUN=sum)))
integrand <- function(x) {(1/sqrt(2*pi))*exp(-x^2)}
int <- integrate(integrand, lower=-Inf, upper=res[1]/sd)
Risk <- res[1]*(pnorm(res[1]/sd)-pnorm(-res[1]/sd))+2*sd*(int$integral)
ans <-c(res[1], res[2], Risk)
names(ans) <-c("Bias", "DF", "Risk")
ans
}

x <- seq(0, 1, length.out=100)
y <- 1-x *(48-x*(218-x*(315-x*145)))
s <- diff(range(y))/4
h <- seq(0.1, 0.5, by=0.05)
for(i in 1:length(h))
 print(risk1(y~x, ev="data",alpha=c(0, h[i]), sigma=s, deg=2))
```

Table 2.8 shows the result. For small bandwidths, such as h=0.1, the bias is negligible. But the large variance(measured by the degrees of freedom) results in a large risk. As the bandwidth is increased, the variance drops. As the bandwidth increases further, the bias

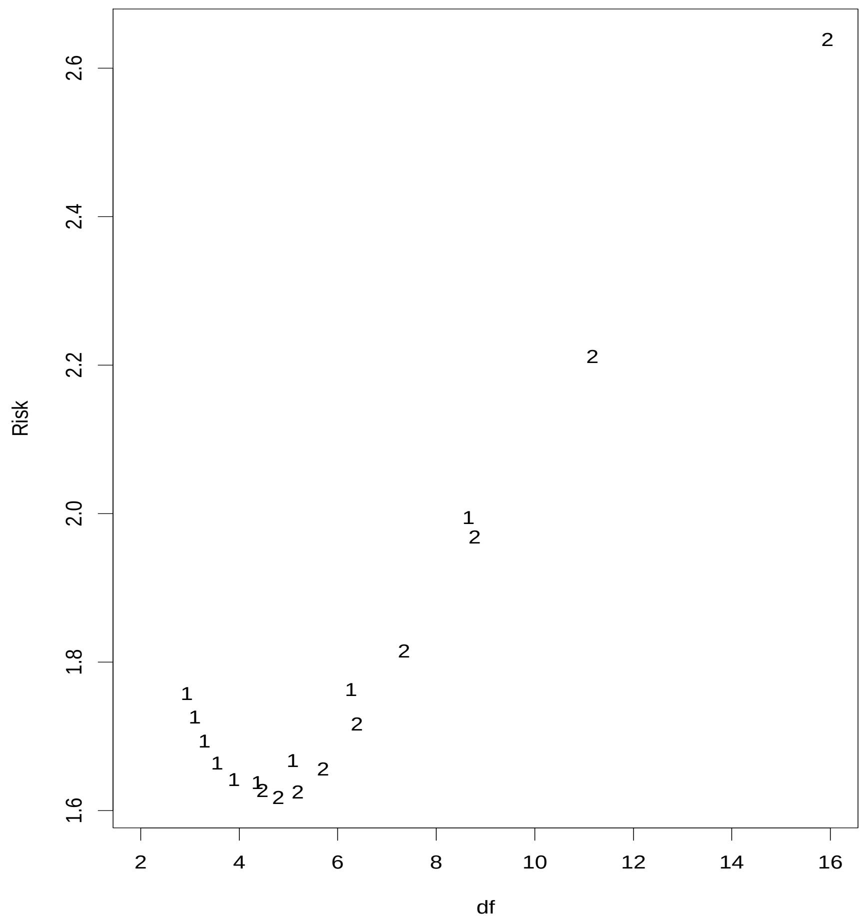

Figure 20: Risk Computation for local linear fitting('1') and local quadratic fitting ('2').

h	Bias	DF	Risk
0.10	0.0010899	15.93925	2.63880
0.15	0.0049488	11.16981	2.21222
0.20	0.0143128	8.779164	1.96922
0.25	0.0313240	7.343388	1.81581
0.30	0.0580691	6.385693	1.71714
0.35	0.0950323	5.701389	1.65669
0.40	0.1419296	5.188035	1.62560
0.45	0.1969542	4.788686	1.61767
0.50	0.2570195	4.469161	1.62735

Table 2.6: Risks of a local quadratic smooth by using **risk1()** function

term begins to dominate, and the risk increases.

Question 2.8 (Cross Validation)

Suppose the risk function for predicting a future observation x_{new}, Y_{new} is $\mathbf{E}|Y_{new} - \hat{\mu}(x_{new})|$.

We estimate this by the L_1 cross validation criterion

$$\mathrm{CV}_1(\hat{\mu}) = \frac{1}{n}\sum_{i=1}^{n} |Y_i - \hat{\mu}_{-i}(x_i)|$$

a) Show

$$\mathrm{CV}_1(\hat{\mu}) = \frac{1}{n}\sum_{i=1}^{n} |\frac{Y_i - \hat{\mu}(x_i)}{1 - \operatorname{infl}(x_i)}|$$

Propose an L_1 version of generalized cross validation

$$\begin{aligned} CV_1(\hat{\mu}) &= \frac{1}{n}\sum_{i=1}^{n} |Y_i - \hat{\mu}_{-i}(x_i)| \\ &= \frac{1}{n}\sum_{i=1}^{n} |Y_i - \frac{\hat{\mu}(x_i) - \operatorname{infl}(x_i) Y_i}{1 - \operatorname{infl}(x_i)}| \\ &= \frac{1}{n}\sum_{i=1}^{n} |\frac{Y_i - \hat{\mu}(x_i)}{1 - \operatorname{infl}(x_i)}| \end{aligned}$$

$$GCV(\hat{\mu}) = \frac{\sum_{i=1}^{n}(Y_i - \hat{\mu}(x_i))}{n - tr(\mathbf{L})}$$

b) Write an modified version of `gcv()` to implement the L_1 generalized cross validation. Apply this to the ethanol dataset and compare with the old version of `gcv()`.

```
gcv1 <- function(...){
m <- match.call()
m[[1]] <- as.name("locfit")
fit <- eval(m, sys.parent())
```

```
fit
z <- fit$dp[c("t0", "t1")]
res <- residuals(fit)
res1 <- list(abs(res))
res2 <- lapply(res1, FUN=sum)
n <- fit$mi["n"]
z <- c(z, unlist(res2)/(n-z[1]))
names(z) <- c("infl", "vari", "gcv1")
z
}

alpha <- seq(0.2, 0.8, by= 0.05)
g <- matrix(nrow =length(alpha), ncol=3)
for(i in 1:length(alpha))
 g[i, ] <- gcv1(NOx~E, data=ethanol, deg=2, alpha= alpha[i])
cbind(alpha, g)
> cbind(alpha, g)
      alpha
 [1,]  0.20 18.812689 16.426519 0.2438392
 [2,]  0.25 13.967833 12.634874 0.2378341
 [3,]  0.30 11.469545 10.516466 0.2360207
 [4,]  0.35 10.463944  9.577061 0.2394911
 [5,]  0.40  8.435035  7.850665 0.2520384
 [6,]  0.45  7.859468  7.316354 0.2542524
 [7,]  0.50  7.013307  6.487448 0.2568973
 [8,]  0.55  6.525372  6.011243 0.2594486
 [9,]  0.60  6.070958  5.573666 0.2654261
[10,]  0.65  5.338625  5.000415 0.2992648
[11,]  0.70  5.091619  4.764938 0.3070468
[12,]  0.75  4.770184  4.462727 0.3346825
[13,]  0.80  4.569641  4.278240 0.3574357
```

```
> plot(g[,2], g[,3], xlab="Fitted DF", ylab="GCV")
```

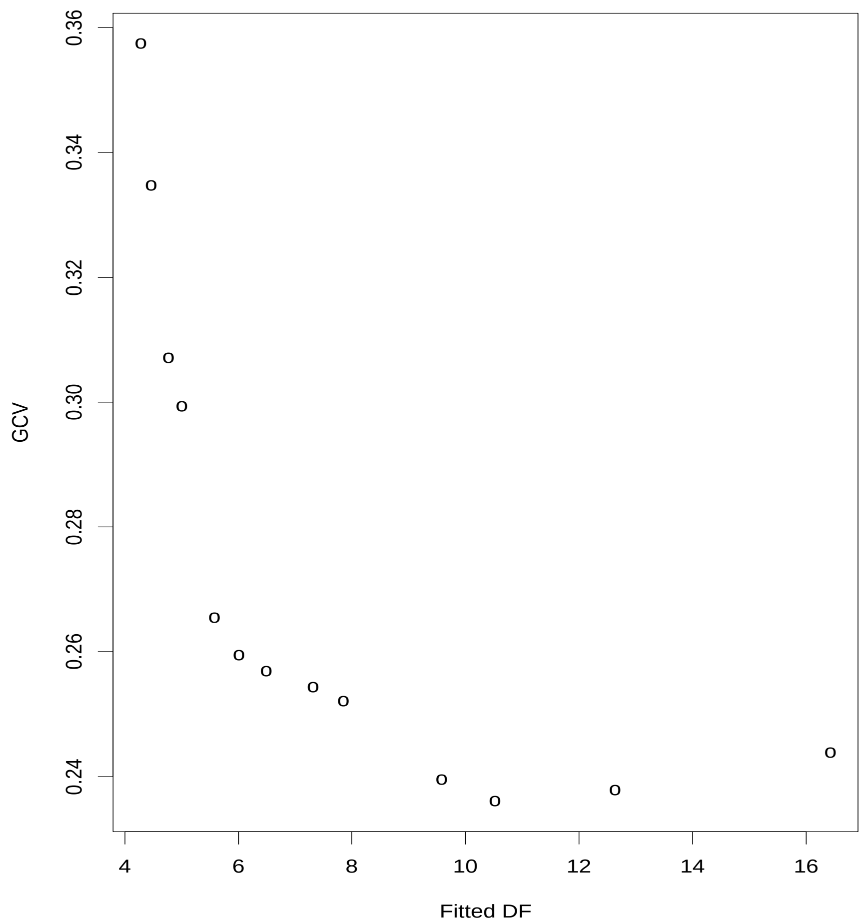

Figure 21: Generalized cross validation scores for the ethanol dataset by using **gcv1() function**

As same as figure 3 when fitted DF=10, GCV is smallest.