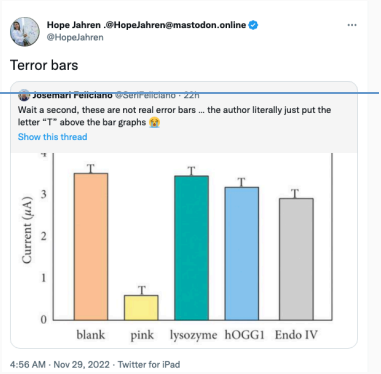
Methods of Applied Statistics I

STA2101H F LEC9101

Week 11

November 30 2022



link

Written by Witten: Impostors Anonymous


Jenny Jimenez

Contributing Editor Daniela Witten writes:

I was in my second year of grad school when I first heard of "impostor syndrome", the well-studied psychological phenomenon by which highly talented and accomplished people doubt their talent and accomplishments, and live in constant fear that the outside world will discover them as frauds. I remember marveling at the possibility that some of the breathtakingly brilliant statisticians in my department might question their own abilities—had they no self-awareness?!? The absurdity of their impostor syndrome stood in stark contrast to what I knew to be true: that the people around me had their figurative ducks all in a row, whereas *I was the real fraud*.

Friends, it did not occur to me that I might, in fact, suffer from impostor syndrome, until several years into my extremely hard. If all of my accomplishments to date were due to luck and/or trickery, then I had better hurry to accomplish as much as possible before my luck changed and/or my trickery was discovered! I am certain that I would not have achieved the same level of success as early in my career without my impostor syndrome. But, I might have been a lot happier and 90% as successful, and I firmly believe that this would have been enough, for any reasonable definition of "enough".

(I also acknowledge that impostor syndrome can manifest in different ways. For instance, some people might find themselves unable to complete a research paper due to a fear that others will discover them to be a fraud.)

Over the years, I've learned that my impostor syndrome places a burden on those around me. If I believe that everyone else is smarter than me, then I will have unrealistically high expectations for others. This manifests not only in thinking that all of my grad students are brilliant (and in fact, they are!) but also in expecting them to constantly

Today

1. Upcoming events
2. Project
3. Recap
4. Nonparametric regression

Project due December 19 (11.59),
no extensions
So think of it as due on December 16 :)

Preliminary versions accepted
for feedback up to Dec 11

Project Guidelines STA 2101F: Methods of Applied Statistics I 2022

Outline

- Part I 3–5 pages, non-technical — 12 point type, 1.5 vertical spacing, thank you
  1. a description of the scientific problem of interest
  2. how (and why) the data being analyzed was collected
  3. preliminary description of the data (plots and tables)
  4. non-technical summary for a non-statistician of the analysis and conclusions
- Part II 3–5 pages, technical — LaTeX or R markdown; submit .Rmd and .pdf files
  1. models and analysis
  2. summary for a statistician of the analysis and conclusions
- Part III Appendix — submit .Rmd and .pdf or .html files
  R script or .Rmd file; additional plots; additional analysis; References

Project Marking

- 40 points total
- Part I:
  description of data and scientific problem 5
  suitability of plots and tables 5 — clear, non-technical, concise but thorough
  quality of the presentation 5
- Part II:
  summary of the modelling and methods 5 — justification for choices
  suitability and thoroughness of the analysis 10 — model checks, data checks
- Part III:
  relevance of additional material 5
  complete and reproducible submission 5

Upcoming

- December 1 11.00-4.30 PostDoc Day
  Room 9014, Hydro Building
  Register here

- December 5 3.30 Data Science ARES Amy Kuceyeski, Cornell
  "Quantitative modeling of brain-behavior relationships"
  online

- December 8 3.30-4.30 Statistics Seminar
  Room 9014, Hydro Building
  Emma Hubert, Princeton
  "Continuous-Time Incentives in Hierarchies"

Recap

- proportional hazards regression; estimation of survivor function; interpretation of coefficients
- factorial experiments, fixed and random effects, nested and crossed factors
- expected mean squares and components of variance
- fitting with `lme4::lmer` gives estimates of fixed and random effects
- general formulation of mixed models

- random effects are useful when the factor levels are not themselves meaningful, but rather a sample of possible levels
- random effects can be used to induce dependence between measurements on the same unit

Recap: Components of variance

Some factorial models:

$$y_{ij} = \mu + \alpha_i + \epsilon_{ij}$$

$$y_{ijk} = \mu + \alpha_i + \beta_{ij} + \epsilon_{ijk}$$

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

$$y_{ij} = \beta_0 + \beta_1 x_i + b_0 + b_1 x_i + \epsilon_{ij}$$

$$y = X\beta + Zb + \epsilon$$

	school	ses	mathach	sector	cses	meanses
9	1224	-0.888	1.527	Public	-0.45362	-0.43438
248	1433	1.332	18.496	Catholic	0.62000	0.71200
1094	2467	0.062	6.415	Public	0.39173	-0.32973
1195	2629	0.942	11.437	Catholic	1.07965	-0.13765
1283	2639	-1.088	-0.763	Public	-0.12357	-0.96443
2334	3657	-0.288	13.156	Public	0.36118	-0.64918
2783	4042	0.792	14.500	Catholic	0.39000	0.40200
2806	4042	0.482	3.687	Catholic	0.08000	0.40200
2886	4223	1.242	20.375	Catholic	1.33600	-0.09400
3278	4511	-0.178	15.550	Catholic	-0.07086	-0.10714
3317	4511	0.342	7.447	Catholic	0.44914	-0.10714
3656	5404	0.902	18.802	Catholic	0.07702	0.82498
5180	7232	0.442	23.591	Public	0.53212	-0.09012
5223	7276	-1.098	-1.525	Public	-1.17623	0.07823
5278	7332	-0.508	16.114	Catholic	-0.80500	0.29700
5467	7364	-0.178	20.325	Catholic	-0.08864	-0.08936
6292	8707	-0.228	18.463	Public	-0.38313	0.15513

F & W show separate regressions for each school, and then a more parsimonious model with fixed effects for SES and sector and random slope for student-level SES

- design: one factor with I levels; J responses at each level
- model

$$y_{ij} = \mu + \alpha_i + \epsilon_{ij}, \quad j = 1, \ldots J; i = 1, \ldots I; \quad \epsilon_{ij} \sim (0, \sigma_\epsilon^2)$$

Analysis of variance table

Term	degrees of freedom	sum of squares	mean square	F-statistic
treatment	$(I-1)$	$\sum_{ij}(\bar{y}_{i.} - \bar{y}_{..})^2$	$\sum_{ij}(\bar{y}_{i.} - \bar{y}_{..})^2/(I-1)$	$MS_{treatment}/MS_{error}$
error	$I(J-1)$	$\sum_{ij}(y_{ij} - \bar{y}_{i.})^2$	$\sum_{ij}(y_{ij} - \bar{y}_{i.})^2/\{I(J-1)\}$	
total(corrected)	$IJ-1$	$\sum_{ij}(y_{ij} - \bar{y}_{..})^2$		

$\mathrm{E}(SS_{error}) = I(J-1)\sigma_\epsilon^2$,
$\mathrm{E}(SS_{treatment}) = (I-1)(J\sigma_\alpha^2 + \sigma_\epsilon^2)$

Expected Mean Squares

$$\mathrm{E}(SS_{error}) = I(J-1)\sigma_\epsilon^2,$$
$$\mathrm{E}(SS_{treatment}) = (I-1)(J\sigma_\alpha^2 + \sigma_\epsilon^2)$$

Mixed effects models

- more usual to have a model with some fixed effects: treatments, explanatory variables (age, income, ...)
- and some random effects: cluster, family, school, hospital, ...

- the general form of a linear mixed effect model is

$$y = X\beta + Z\gamma + \epsilon; \quad \epsilon \sim N(0, \sigma^2 I_n), \gamma \sim N(0, \sigma^2 D) \implies y \sim N(X\beta, \sigma^2(I_n + ZDZ^T))$$

- model matrix $X_{n\times p}$, fixed effects β; model matrix $Z_{n\times q}$, random effects γ

- fit with maximum likelihood for β, "REML" for variance components
- rely on `lme4` for inference about fixed and random effects

... Example 9.18

```
> summary(rat.mixed}
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ week + (week | rat)
   Data: rat.growth
...
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 rat      (Intercept) 119.54   10.933
          week         12.49    3.535   0.18
 Residual              33.84    5.817
Number of obs: 150, groups:  rat, 30

Fixed effects:
            Estimate Std. Error t value
(Intercept) 156.0533     2.1590   72.28
week         43.2667     0.7275   59.47
```

"the estimated mean weight in week 1 is 156, but the variability from rat to rat has standard deviation of about 11 about this.

The slopes show similarly large variation.

The measurement error variance $\hat{\sigma}^2 = 5.82^2$ is smaller than the inter-rat variation in intercepts but exceed that for slopes"

- model $y_i = f(x_i) + \epsilon_i, \quad i = 1, \ldots, n \quad x_i$ scalar
- mean function $f(\cdot)$ assumed to be "smooth"
- introduce a kernel function $K(u)$ and define a set of weights

$$w_i = \frac{1}{\lambda} K\left(\frac{x_i - x_0}{\lambda}\right)$$

- estimate of $f(x)$, at $x = x_0$:

$$\hat{f}_\lambda(x_0) = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i}$$

- Nadaraya-Watson estimator – local averaging

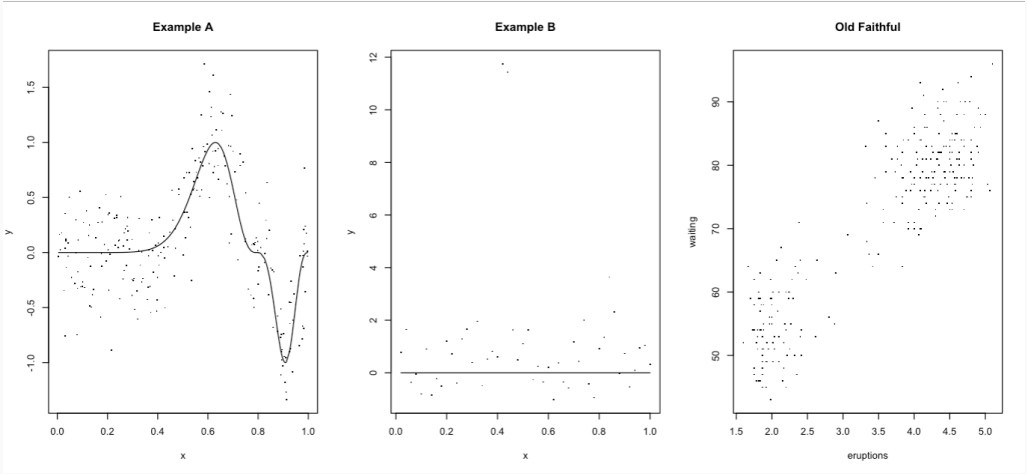
local polynomial of degree 0

Kernel smoothers

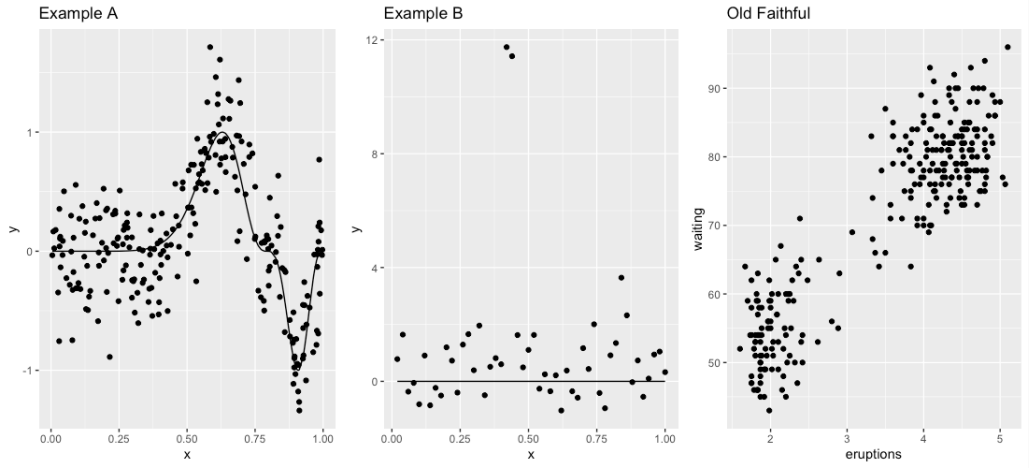
- choice of bandwidth, λ controls smoothness of function
- larger bandwidth $=$ more smoothing
- kernel estimators are biased
- making the estimate smoother increases bias, decreases variance

- choice of kernel function, $K(\cdot)$, controls smoothness and "local-ness"
- Faraway recommends Epanechnikov kernel $K(x) = \frac{3}{4}(1 - x^2), |x| \leq 1$
- `ksmooth(base)` offers only uniform (box) or normal
- `bkde(KernSmooth)` offers `normal, box, epanech, biweight, triweight`
- `biweight`: $K(x) \propto (1 - |x|^2)^2, |x| \leq 1$ `triweight`: $K(x) \propto (1 - |x|^2)^3, |x| \leq 1$

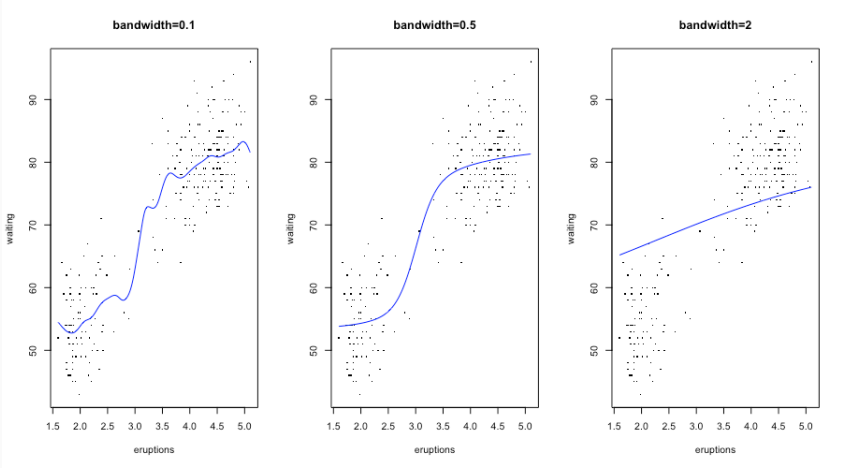
Examples



```
exb <- data.frame(exb)

plota <- ggplot(exa) + geom_point(aes(x,y)) +
geom_line(aes(x,m))+ ggtitle("Example A")

plotb <- ggplot(exb) + geom_point(aes(x,y)) +
geom_line(aes(x,m))+ ggtitle("Example B")

plotc <- ggplot(faithful) + geom_point(aes(eruptions,waiting)) +
 ggtitle("Old Faithful")

grid.arrange(plota, plotb, plotc, nrow=1) #in gridExtra library
```


Example A
y
x
1
0
-1
0.00
0.25
0.50
0.75
1.00
Example B
y
x
12
8
4
0
0.00
0.25
0.50
0.75
1.00
Old Faithful
waiting
eruptions
90
80
70
60
50
2
3
4
5

```
with(faithful, plot(eruptions, waiting, cex=2, main = "bandwidth=0.1", pch="."))
lines(locpoly(faithful$eruptions,faithful$waiting,drv=0L,
  degree=0,bandwidth=.1), col = "blue")

with(faithful, plot(eruptions, waiting, cex=2, main = "bandwidth=0.5", pch="."))
lines(locpoly(faithful$eruptions,faithful$waiting,drv=0L,
  degree=0, bandwidth=.5), col = "blue")

with(faithful, plot(eruptions, waiting, cex=2, main = "bandwidth=2", pch="."))
lines(locpoly(faithful$eruptions,faithful$waiting,drv=0L,
  degree=0, bandwidth=2), col = "blue")
```



These are smoother than the plots in ELM using `base::ksmooth`

- Nadaraya-Watson: $\hat{f}_\lambda(x) = \Sigma w_i y_i / \Sigma w_i$; $\quad w_i = \frac{1}{\lambda} K(\frac{x_i - x_0}{\lambda})$
- $\hat{f}_\lambda(x)$ is biased

$$\mathrm{E}\{\hat{f}_\lambda(x)\} \doteq \frac{1}{2}\lambda^2 f''(x)$$

$$\mathrm{var}\{\hat{f}_\lambda(x)\} \doteq \frac{\sigma^2}{n\lambda g(x)} \int_{-1}^{1} K^2(u)du$$

$g(\cdot)$ limiting density of x's

- could choose λ to minimize MSE $=$ bias^2 $+$ var, at x
- could choose λ to minimize integrated MSE

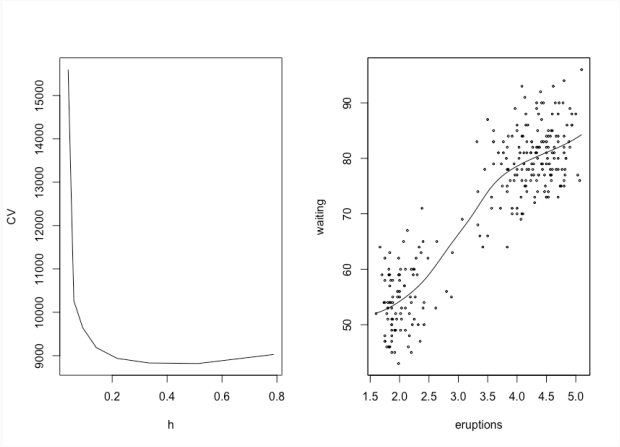
- more usual to use cross-validation

SM 10.7.1 (no n); ELM 11.1

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\{y_i - \hat{f}_{-i}(x_i)\}^2$$

Cross-validation

```
library(sm)
hm <- hcv(faithful$eruptions,
   faithful$waiting, display = "lines")
sm.regression(faithful$eruptions,
   faithful$waiting, h = hm,
   xlab = "eruptions",
   ylab = "waiting")
```



- above uses local averaging based on kernel function
- better estimates can be obtained using local regression at point x

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & (x_1 - x_0) & \cdots & (x_1 - x_0)^k \\ \vdots & \vdots & & \vdots \\ 1 & (x_n - x_0) & \cdots & (x_n - x_0)^k \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix},$$

- attach a weight to each observation (y_i, x_i) according to the distance from x_i to x_0

$$\hat{\beta} = (X^T W X)^{-1} X^T W y, \quad W = \mathrm{diag}(w), \quad w_i = \frac{1}{\lambda} K(\frac{x_i - x_0}{\lambda})$$

$$\hat{f}_\lambda(x_0) = \hat{\beta}_0$$

- usually evaluate the function at sample points: $\hat{f}_\lambda(x_i), i = 1, \ldots, n$

- odd-order polynomials work better than even; usually local linear fits are used

- kernel function is often a Gaussian density, or the tricube kernel

$$K(u) = (1 - |u|^3)^3, \quad |u| \leq 1$$

- as with N-W (local averaging) estimators, choice of bandwidth controls smoothness
- **loess** is the most widely used, and is the default in `ggplot2`

- fits a local linear regression, but not by least squares

- uses a robust version of least squares that downweights outliers
- the result is that the bandwidth can change with x

- $\hat{\beta} = (X^T W X)^{-1} X^T W y \qquad W = \text{diag}(w_1, \ldots, w_n)$
- $\hat{f}_\lambda(x_0) = \hat{\beta}_0 = \sum_{i=1}^n S(x_0; x_i, \lambda) y_i$
- $S(x_0; x_1, \lambda), \ldots, S(x_0; x_n, \lambda)$ first row of "hat" matrix
- this makes it relatively easy to analyse the behaviour of local polynomial smoothers

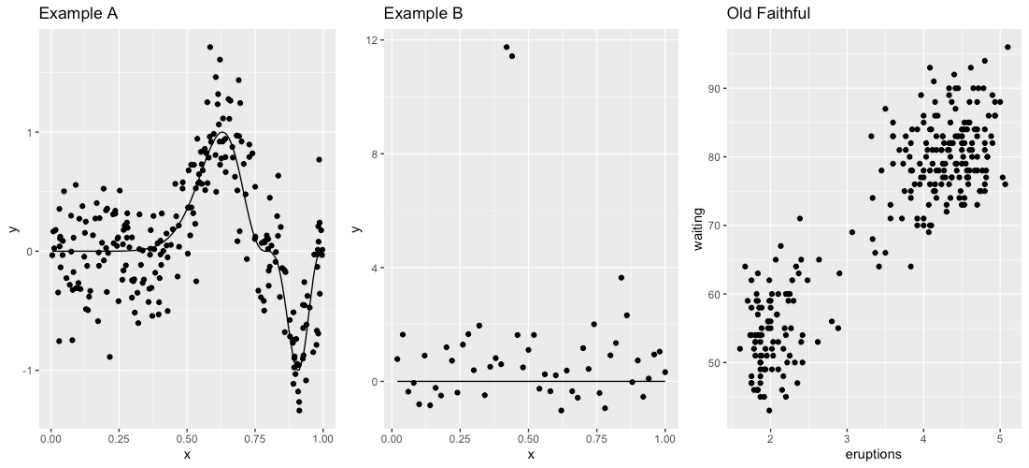
SM 10.7

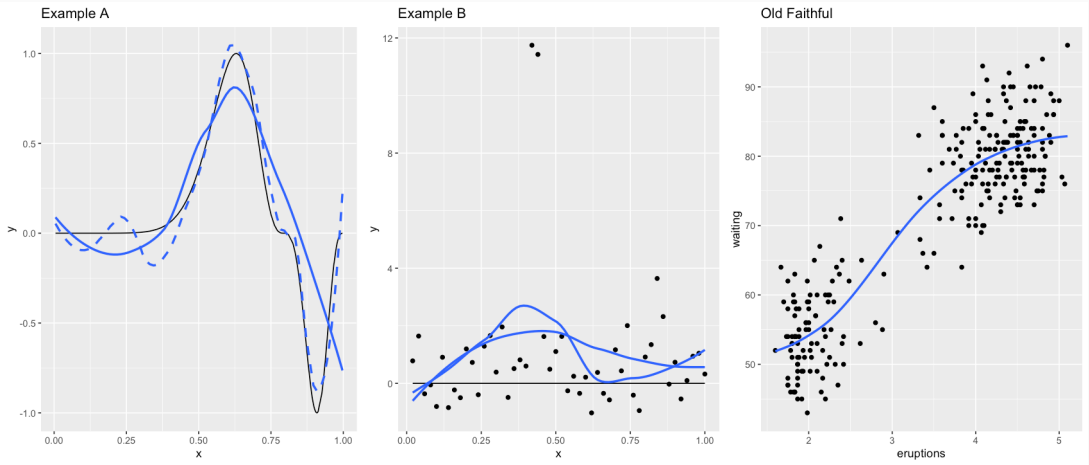
- and to simplify the expression for the cross-validation criterion $CV(\lambda)$
- fitting at each sample value gives

$$\hat{f}_\lambda(x_i) = \sum_{j=1}^{n} S(x_i; x_j, \lambda) y_j$$

- $$CV(\lambda) = \sum_{i=1}^{n} \{y_i - \hat{f}_{-i}(x_i)\}^2$$
- for local polynomials
$$CV(\lambda) = \sum_{i=1}^{n} \left\{ \frac{y_i - \hat{f}_\lambda(x_i)}{1 - S_{ii}(\lambda)} \right\}^2$$
- even simpler
$$GCV(\lambda) = \sum_{i=1}^{n} \left\{ \frac{y_i - \hat{f}_\lambda(x_i)}{1 - \mathrm{tr}(S_\lambda)/n} \right\}^2$$
- $$\hat{f}_\lambda(x_i) = \sum_{j=1}^{n} S(x_i; x_j, \lambda) y_j$$

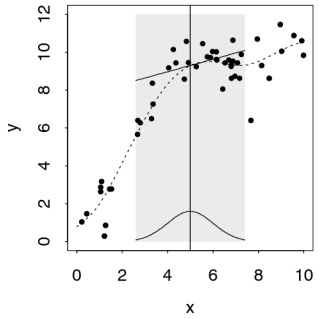
Examples


Example A
y
x
1
0
-1
0.00
0.25
0.50
0.75
1.00
Example B
y
x
12
8
4
0
0.00
0.25
0.50
0.75
1.00
Old Faithful
waiting
eruptions
90
80
70
60
50
2
3
4
5



`geom_smooth` in `ggplot` uses local polynomial fitting

robustified



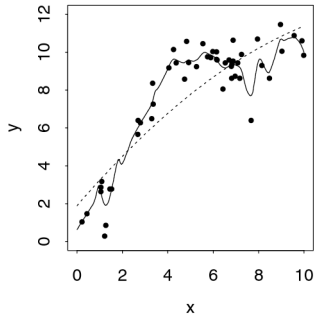
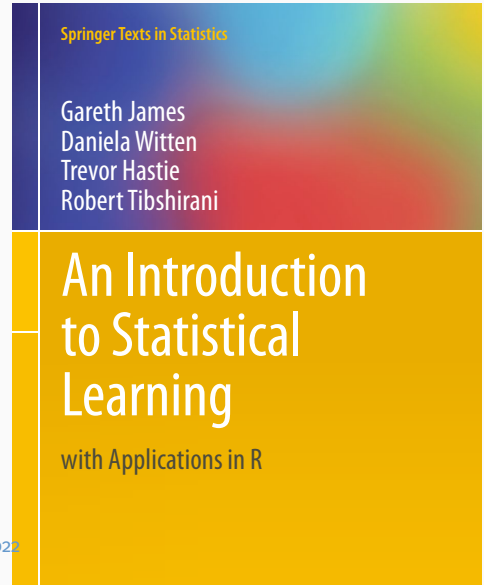


Figure 10.14 Construction of a local linear smoother. Left panel: observations in the shaded part of the panel are weighted using the kernel shown at the foot, with $h = 0.8$, and the solid straight line is fitted by weighted least squares. The local estimate is the fitted value when $x = x_0$, shown by the vertical line. Two hundred local estimates formed using equi-spaced x_0 were interpolated to give the dotted line, which is the estimate of $g(x)$. Right panel: local linear smoothers with $h = 0.2$ (solid) and $h = 5$ (dots).

Recall that a kernel function $w(u)$ is a unimodal density function symmetric about $u = 0$ and with unit variance. One choice of w is the standard normal density. Another is a rescaled form of the *tricube* function

$$w(u) = \begin{cases} (1 - |u|^3)^3, & |u| \le 1, \\ 0, & \text{otherwise}, \end{cases} \qquad (10.37)$$

Great reference



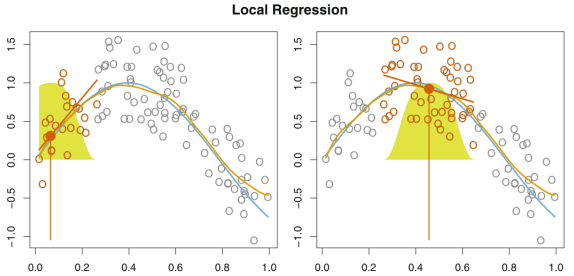


FIGURE 7.9. *Local regression illustrated on some simulated data, where the blue curve represents $f(x)$ from which the data were generated, and the light orange curve corresponds to the local regression estimate $\hat{f}(x)$. The orange colored points are local to the target point x_0, represented by the orange vertical line. The yellow bell-shape superimposed on the plot indicates weights assigned to each point, decreasing to zero with distance from the target point. The fit $\hat{f}(x_0)$ at x_0 is obtained by fitting a weighted linear regression (orange line segment), and using the fitted value at x_0 (orange solid dot) as the estimate $\hat{f}(x_0)$.*

Inference after fitting local polynomials

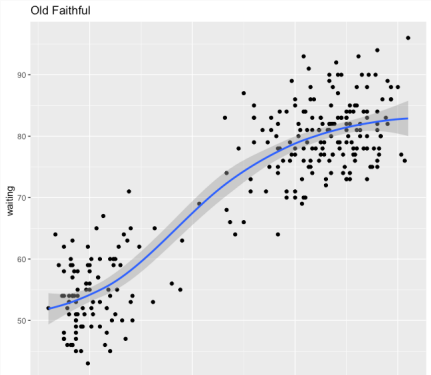
- model: $y_i = f(x_i) + \epsilon_i, \quad i = 1, \ldots, n; \mathrm{E}(\epsilon_i) = 0; \mathrm{var}(\epsilon_i) = \sigma^2$
- $\hat{f}_\lambda(x_0) = \hat{\beta}_0 = \sum_{i=1}^n S(x_0; x_i, \lambda) y_i$
- $\mathrm{E}\{\hat{f}_\lambda(x_0)\} =$
- $\mathrm{var}\{\hat{f}_\lambda(x_0)\} =$
- how many parameters did we fit?
- by analogy with least squares, estimates of 'degrees of freedom' are $\nu_1 = \mathrm{tr}(S_\lambda)$, or $\nu_2 = \mathrm{tr}(S_\lambda^T S_\lambda)$

$$\tilde{\sigma}^2 = \frac{1}{n - 2\nu_1 + \nu_2} \sum \{y_i - \hat{f}_\lambda(x_i)\}^2$$

... inference after fitting local polynomials

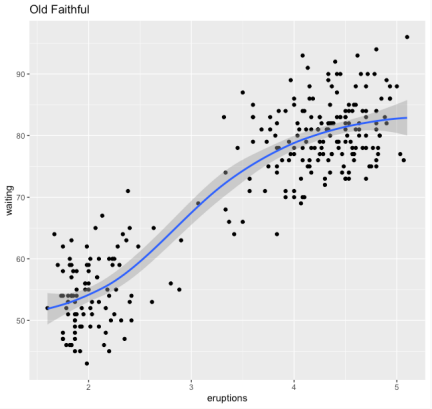
- $\mathrm{E}\{\hat{f}_\lambda(x_0)\} = \sum_{i=1}^n S(x_0; x_i, \lambda) f(x_i), \qquad \mathrm{var}\{\hat{f}_\lambda(x_0)\} = \sigma^2 \sum_{i=1}^n S^2(x_0; x_i, \lambda)$

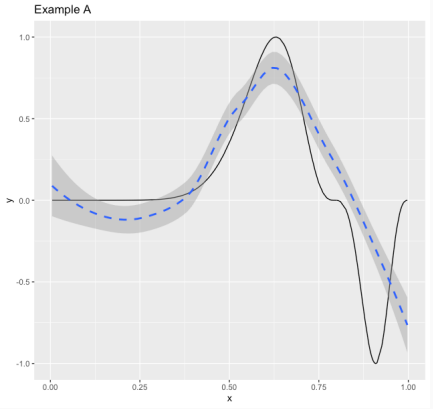
$$\frac{\hat{f}_\lambda(x_0) - \mathrm{E}\{\hat{f}_\lambda(x_0)\}}{\widehat{\mathrm{var}}\{\hat{f}_\lambda(x_0)\}^{1/2}} \dot{\sim} N(0, 1)$$



… inference after fitting local polynomials

SM 10.7





Regression splines

- model $y_i = f(x_i) + \epsilon_i$ $\quad$ $f(\cdot)$ "flexible"
- above $f(\cdot)$ is estimated at several points using local constants or local linear regression

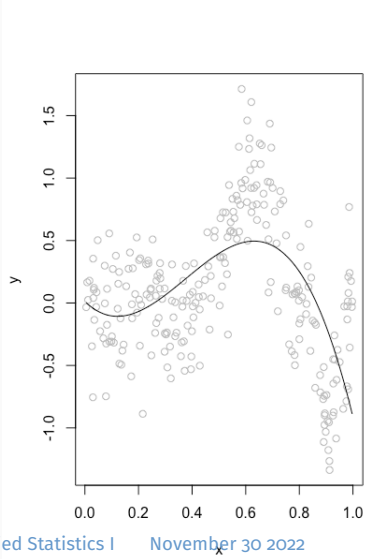
KernSmooth::locpoly

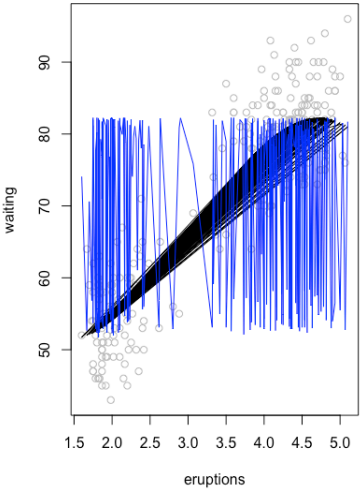
- another popular approach is to use some very flexible, but parametric form, for f
- for example, $f(x) = \sum_{m=1}^{M} \beta_m \phi_m(x)$
- examples of ϕ_m: $1, x, x^2, x^3$; $\quad$ $1, \sin(x), \cos(x), \sin(2x), \cos(2x)$

- popular choice – piecewise polynomials: e.g. knots at $\xi_1, \xi_2 \in [0, 1]$
- basis functions $\phi(x) : 1, x, x^2, = x^3, (x - \xi_1)^3_+, (x - \xi_2)^3_+$

- ELM p.219 builds these "by hand"
- `splines::bs()` builds cubic splines automatically

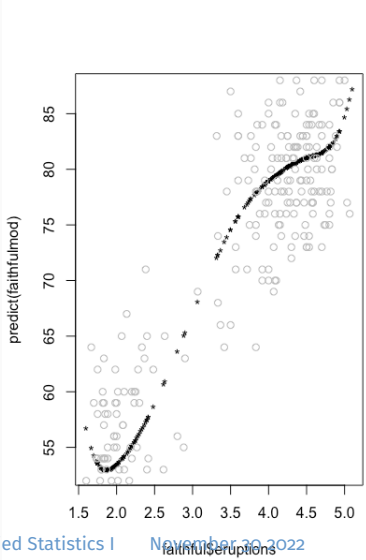
```
> library(splines)
> examod <- lm(y ~ bs(x,3), data = exa)
> summary(examod)
... Coefficients:
             Estimate Std. Error t value  Pr(>|t|)
(Intercept)  0.0025764  0.1134784  0.0227   0.98190
bs(x, 3)1   -0.6606770  0.3249083 -2.0334   0.04306 *
bs(x, 3)2    2.0122935  0.2029124  9.9171 < 2.2e-16 ***
bs(x, 3)3   -0.8910698  0.1668033 -5.3420 2.055e-07 ***
---
Residual standard error: 0.46159 on 252 degrees of freedom
Multiple R-squared:  0.3426, Adjusted R-squared:  0.33478
F-statistic: 43.777 on 3 and 252 DF,  p-value: < 2.22e-16

> model.matrix(examod)
    (Intercept)     bs(x, 3)1     bs(x, 3)2     bs(x, 3)3
1             1 0.0000000e+00 0.00000000000 0.0000000e+00
2             1 1.1399500e-02 0.00004381762 5.6142342e-08
3             1 2.0569481e-02 0.00014401768 3.3611431e-07
4             1 2.5078093e-02 0.00044781161 1.2578287e-06
```

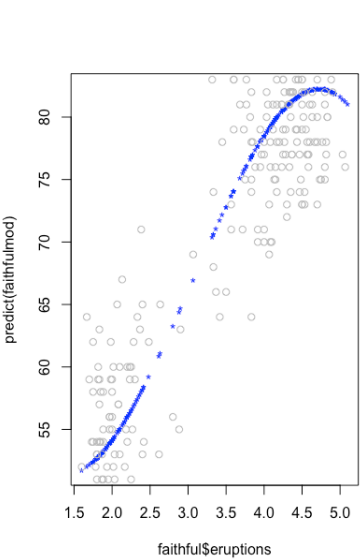

y
x
0.0
0.2
0.4
0.6
0.8
1.0
-1.0
-0.5
0.0
0.5
1.0
1.5


waiting
eruptions
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
50
60
70
80
90

... Regression splines

ELM 14.2 p.303





Smoothing splines

- $y_i = f(x_i) + \epsilon_i, \quad i = 1, \ldots, n$
- choose $f(\cdot)$ to solve

$$\min_f \sum_{i=1}^{n} \{y - f(x_i)\}^2 + \lambda \int_a^b \{f''(t)\}^2 dt, \quad \lambda > 0$$

- solution is a cubic spline, with knots at each observed x_i value

see SM Figure 10.18 for a non-regularized solution

- has an explicit, finite dimensional solution
- $\hat{f} = \{\hat{f}(x_1), \ldots, \hat{f}(x_n)\} = (I + \lambda K)^{-1} y$
- K is a symmetric $n \times n$ matrix of rank $n - 2$

Nonparametric regression

- $y_i = f(x_i) + \epsilon_i$

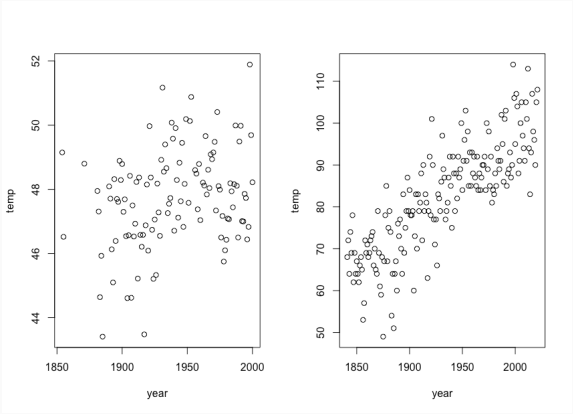
- local polynomial regression – `stats::loess, KernSmooth::locpoly` ELM 11.3; SM 10.7.1
- regression splines – `splines::bs , splines::ns` ELM 11.2b p 218ff
- smoothing splines – `stats:smooth.spline` ELM 11.2a; SM 10.7.2

- penalized splines – `pspline::smooth.Pspline` Peng et al. 2006
- wavelets – `wavethresh::wd` ELM 11.4
- and more... ELM 11.5; ISLR Ch.7

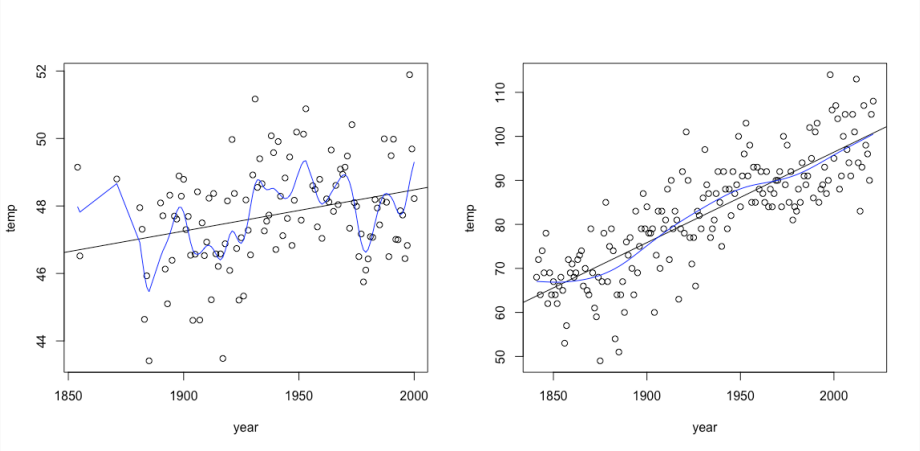
- same ideas can be applied to generalized linear models
- replace linear predictor $\eta_i = x_i^T \beta$ with $f(x_i)$
- use local poly, reg splines, etc. SM Ex. 10.32 logistic regression

Example: ELM Exercise 14.5

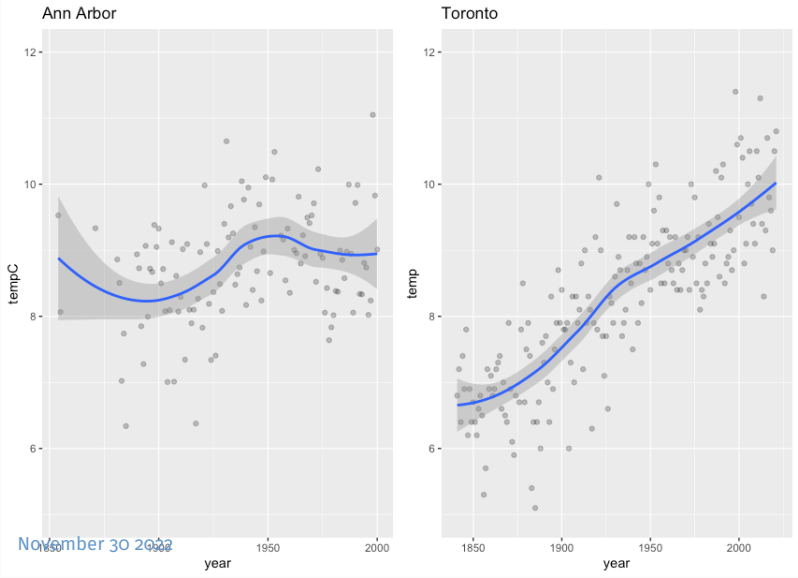
```
> data(aatemp)
> plot(year ~ temp, aatemp)
> View(toronto) # not shown
> plot(year ~ temp, toronto)
```



... Example: ELM Exercise 14.5



... Example: ELM Exercise 14.5



Example: logistic regression

SM Ex.10.29 and 10.32

City	Rain	r/m	City	Rain	r/m	City	Rain	r/m	City	Rain	r/m
1	1735	2/4	11	2050	7/24	21	1756	2/12	31	1780	8/13
2	1936	3/10	12	1830	0/1	22	1650	0/1	32	1900	3/10
3	2000	1/5	13	1650	15/30	23	2250	8/11	33	1976	1/6
4	1973	3/10	14	2200	4/22	24	1796	41/77	34	2292	23/37
5	1750	2/2	15	2000	0/1	25	1890	24/51			
6	1800	3/5	16	1770	6/11	26	1871	7/16			
7	1750	2/8	17	1920	0/1	27	2063	46/82			
8	2077	7/19	18	1770	33/54	28	2100	9/13			
9	1920	3/6	19	2240	4/9	29	1918	23/43			
10	1800	8/10	20	1620	5/18	30	1834	53/75			

Table 10.19 Toxoplamosis data: rainfall (mm) and the numbers of people testing positive for toxoplasmosis, r, our of m people tested, for 34 cities in El Salvador (Efron, 1986).

Terms	df	Deviance
Constant	33	74.21
Linear	32	74.09
Quadratic	31	74.09
Cubic	30	62.63

Table 10.20 Analysis of deviance for polynomial logistic models fitted to the toxoplasmosis data.

Example: logistic regression

SM Ex.10.29 and 10.32

$\longrightarrow$ toxoplasmosis.Rmd

Example: logistic regression

SM Ex.10.29 and 10.32

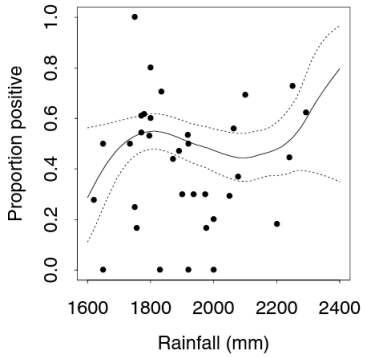


Figure 10.17 Local fits to the toxoplasmosis data. The left panel shows fitted probabilities $\widehat{\pi}(x)$, with the fit of local linear logistic model with $h = 400$ (solid) and 0.95 pointwise confidence bands (dots). Also shown is the local linear fit with $h = 300$ (dashes). The right panel shows the local quadratic fit with $h = 400$ and its 0.95 confidence band. Note the increased variability due to the quadratic fit, and its stronger curvature at the boundaries.