Design of software tools for continuous characteristic analysis

Ross Gayler

25/8/2011

The practice & practicalities of credit scoring

Themes of interest

- How do we (as credit scoring analysts) do what we do?
- How do we do it well?

- The role of tools and tool building in credit scoring
- The role of light-weight tools

  (not complete scorecard development systems, not quantum leap conceptual advances)
  - Better understanding of materials and methods

- The role of design
  - Building tools/products that are adapted for purpose
  - Attention to detail
    - Functionality
    - Task ergonomics
  - Integration and synergy (it all works together well)

Tool building and the innovation mindset

- Small technical advances as tools

Innovation mindset:

- Look at every situation as an opportunity to develop tools
- Even if you don't develop a tool in every situation you will probably understand the situation better
- The insights gained are potential components of tools for future situations

Betty (New Caledonian Crow)

Tool user

Tool builder

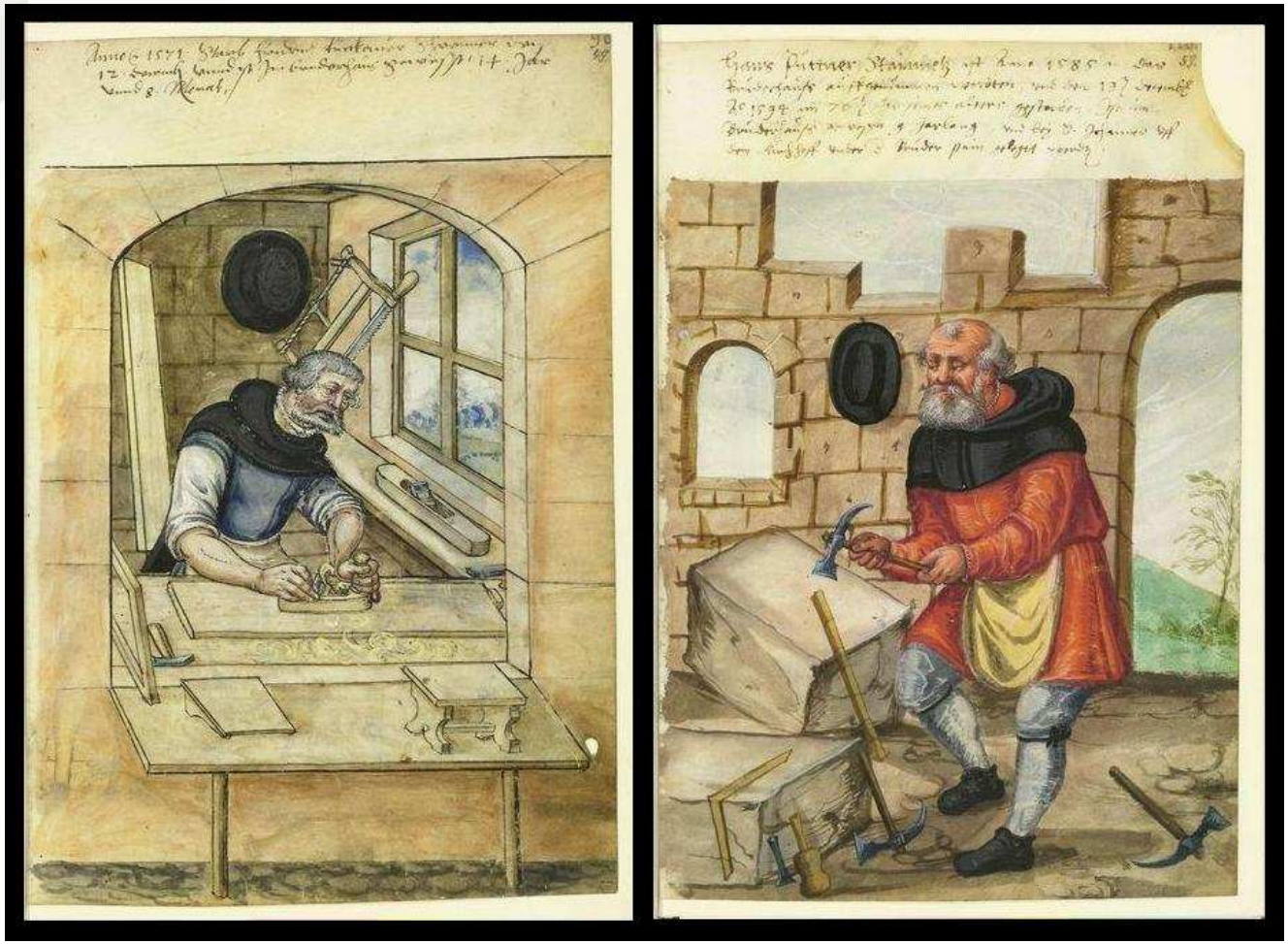
Source: Behavioural Ecology Research Group (http://users.ox.ac.uk/~kgroup/tools/crow_photos.shtml)

The importance of design

- Design is about structural arrangement of components to achieve functional goals
- Novel designs can be made from familiar components (value add is in the novel structure)
- Important to deeply understand the potential and limits of the components
- Design aims for desired composite performance while respecting the component constraints

Credit scoring analyst as artisan

Perfecting the art / Attention to detail



Source: http://www.flickr.com/photos/bibliodyssey/3085763753/sizes/o/in/photostream/ with permission of peacay

Continuous characteristic analysis

Goals for a characteristic analysis tool

- Purpose of characteristic analysis: To understand each potential predictor in isolation
  - Check consistency with expectations
  - Spot data oddities
  - Assess how predictive the characteristic might be
  - Suggest how the characteristic might be encoded to use as a predictor in a model
- Functional goals for a characteristic analysis tool (interrelated)
  - Make data patterns obvious
  - Make data issues obvious
    - Engage visual skills
  - Speed up the characteristic analysis workflow
    - Automate the characteristic analysis generation process
      - Avoid unnecessary actions & unnecessary thinking
    - Can be performed on all characteristics
      - Shows when it is not applicable to a characteristic

Continuous characteristics

- Continuous *x*: numeric, ordered, smooth relation to *y* (neighbouring x → neighbouring *y*)
- Many credit scoring predictors have that form (e.g. age of account)
- But deviations from ideal continuity are quite common, e.g.
  - Sparse values (number of defaults in the last 6 months)
  - Discontinuous at special values (balance = 0)

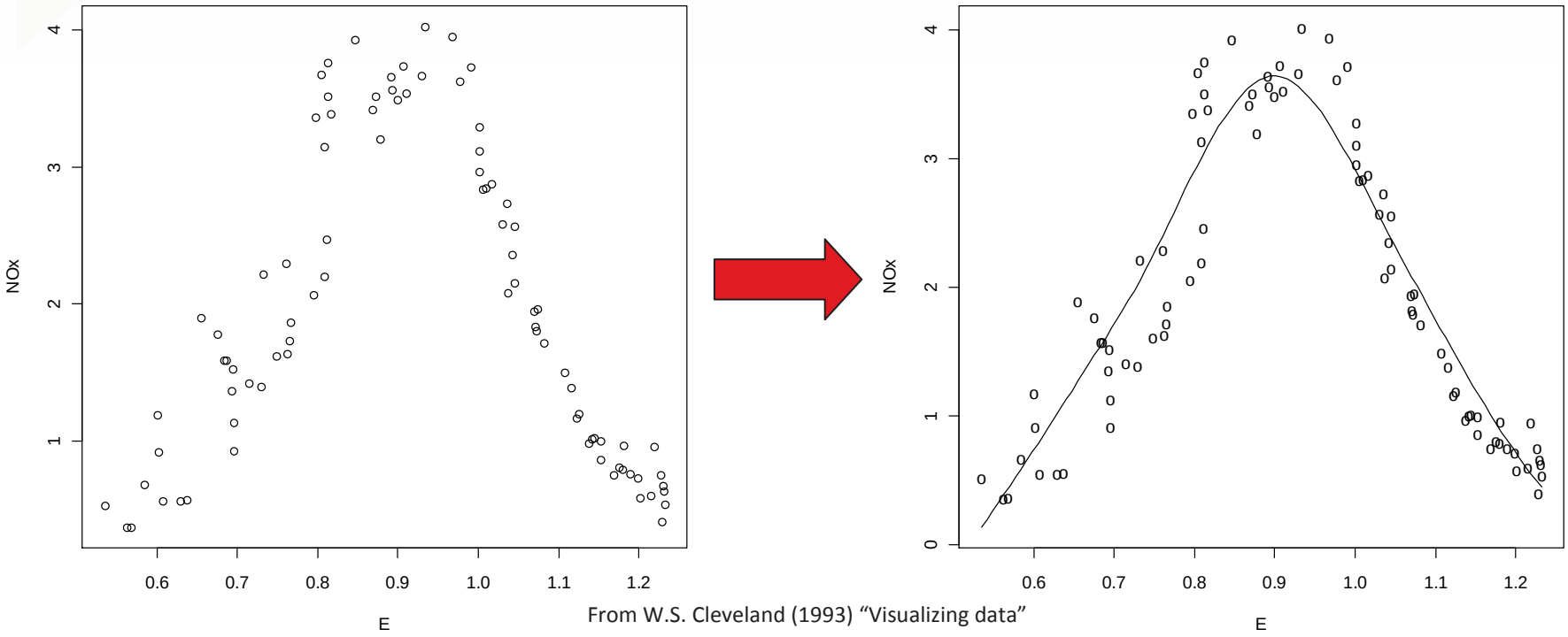
Want a characteristic analysis that:

- Takes advantage of continuity when it is present (as expected)
- Works reasonably for the more deviant continuous *x* (so we can throw all the characteristics at it without being forced to think or recode)
- Shows where the assumptions don't hold (so we know when the data requires us to think)

Characteristic analysis

As function estimation

- Estimate a function/model: $\hat{E}(y) = \hat{f}(x; data)$
  - "Expectation" of *y* conditional on *x* (based on the data at hand)
  - Expectation ≈ mean, median, probability, log-odds, ...
  - In credit scoring *y* is usually binary, so probability or log-odds are typical expectations



From W.S. Cleveland (1993) "Visualizing data"

Characteristic analysis

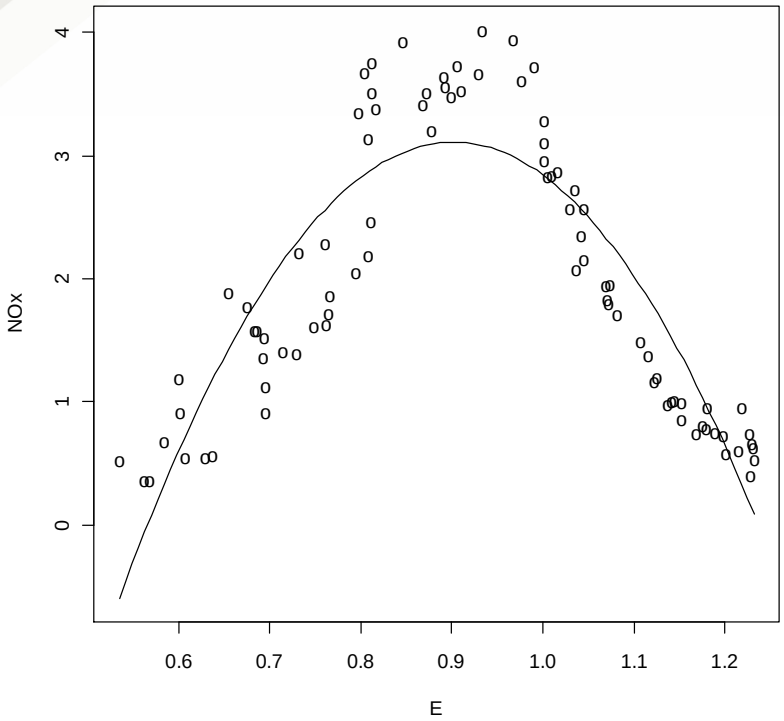
Putting your data in the bin

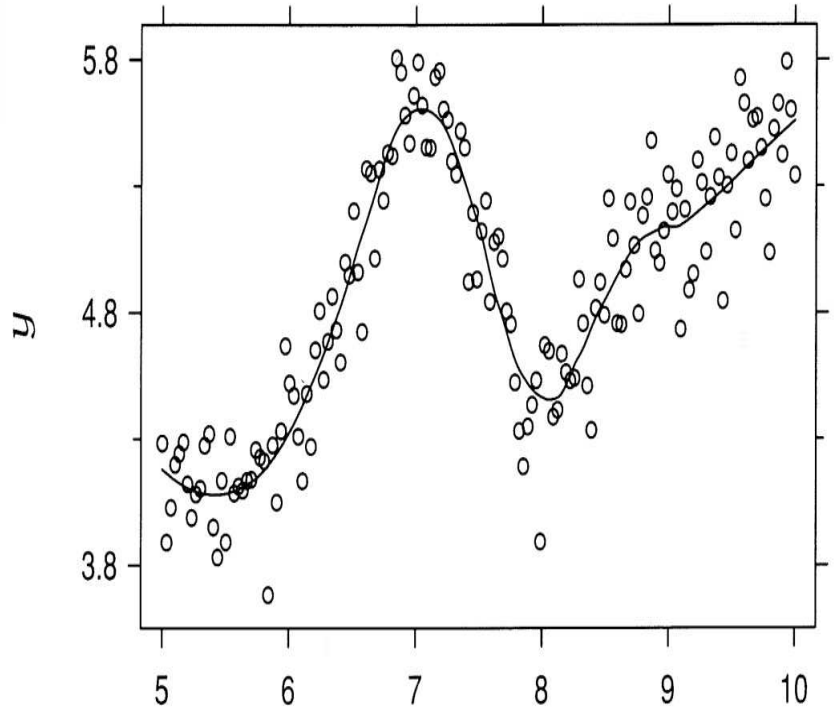
- Estimate a function/model: $\hat{E}(y) = \hat{f}(x; data)$
  - "Expectation" of y conditional on x
  - Expectation ≈ mean, median, probability, log-odds, ...
  - In credit scoring *y* is usually binary, so probability or log-odds are typical expectations
- Conceptually, partition *x* into infinitesimally fine bins and calculate expectation of *y* by bin
  - Imposes no arbitrary bin boundaries
  - Imposes no continuity constraints
  - OK if you have infinite data
- For finite data could use wider bins
  - Contradicts assumption of continuity
    - Arbitrariness of bin boundaries
    - Assumes *y* values are bin-wise constant
  - Inefficient use of data if continuity assumption holds
    - Higher sampling variance

Characteristic analysis

As function estimation by regression

- Could use analytical functional regression (e.g. log, polynomial)
  - No guarantee the best functional form is analytically simple





From W.S. Cleveland (1993) "Visualizing data"

Characteristic analysis

As function estimation by regression

- Could use analytical functional regression (e.g. log, polynomial)
  - No guarantee the best functional form is analytically simple
- Could search for the best functional form
  - Time and effort required to choose the function manually
  - Search can be automated, but no guarantee the best form is analytically simple
- Sensitivity of fitting functions and search process to deviations from continuity
- Indeterminacy of functional approximations
  - Need to trade off goodness of fit and simplicity of function
  - Analytical simplicity not relevant to the goals of characteristic analysis

- Need some other regression based method to estimate the function

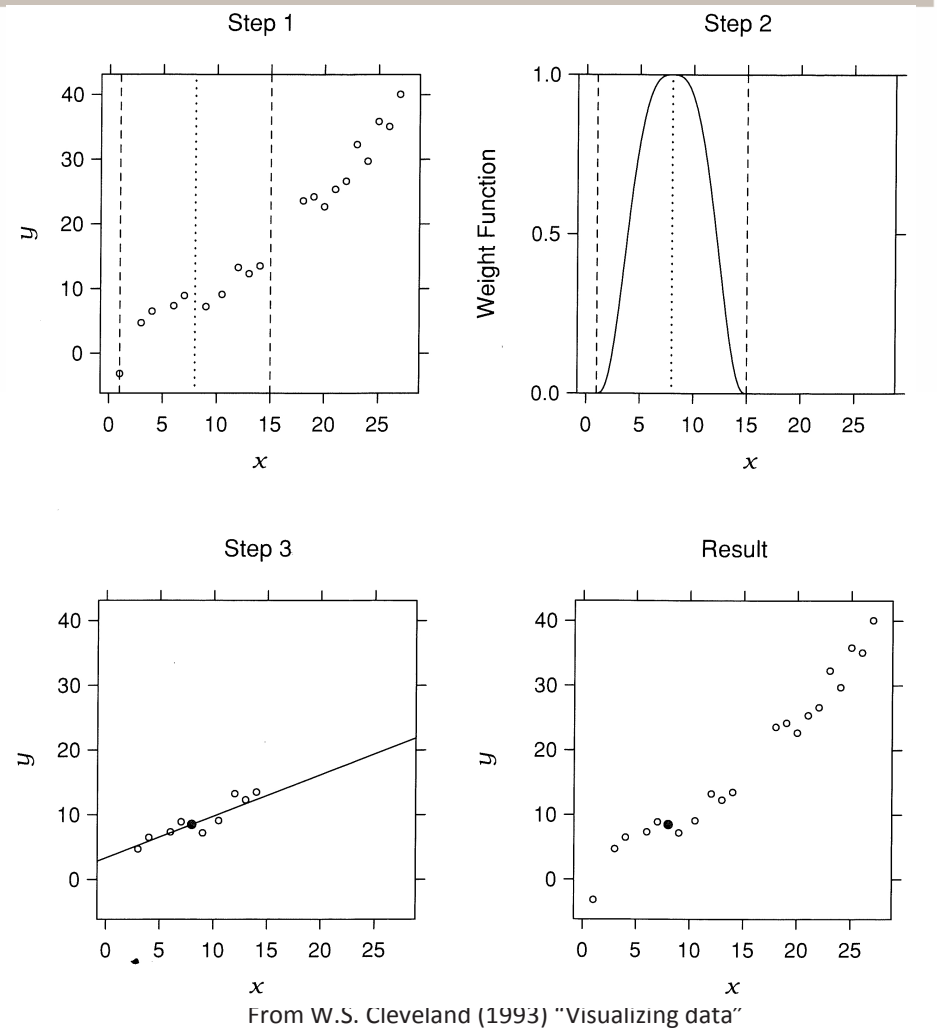
Local regression

Automated estimation of the function

Moving regressions

- Moving averages is moving regression where the regression consists only of a constant term
- Fit a regression within a local window
- Predict *y* at the window midpoint
- Move the window and repeat
- The set of point predictions (*x*,*y*) defines the characteristic analysis (i.e. smooth, continuous function from *x* to *y*)

- Makes efficient use of data if smooth
  - Works well for small *n*
  - Good for tracking patterns in small segments of the population



From W.S. Cleveland (1993) "Visualizing data"

Local regression

- It is regression so you can do all the things you love to do in regression
- GLM - logistic regression for binary outcome data
  - Natural smooth is log-odds(*y*) as a function of *x*
- Specify more complex regression models (smoothing formulae in R)
  - *y* is a smooth function of *x*
    - *y* ~ s(*x*)
  - *y* is a smooth function of *x* with discontinuous special values
    - *y* ~ s(*x*) + I(x == 0) + I(x >= 999)
  - *y* is a smooth function of *x* with *score* partialled out
    - *y* ~ s(*x*) + offset(score)

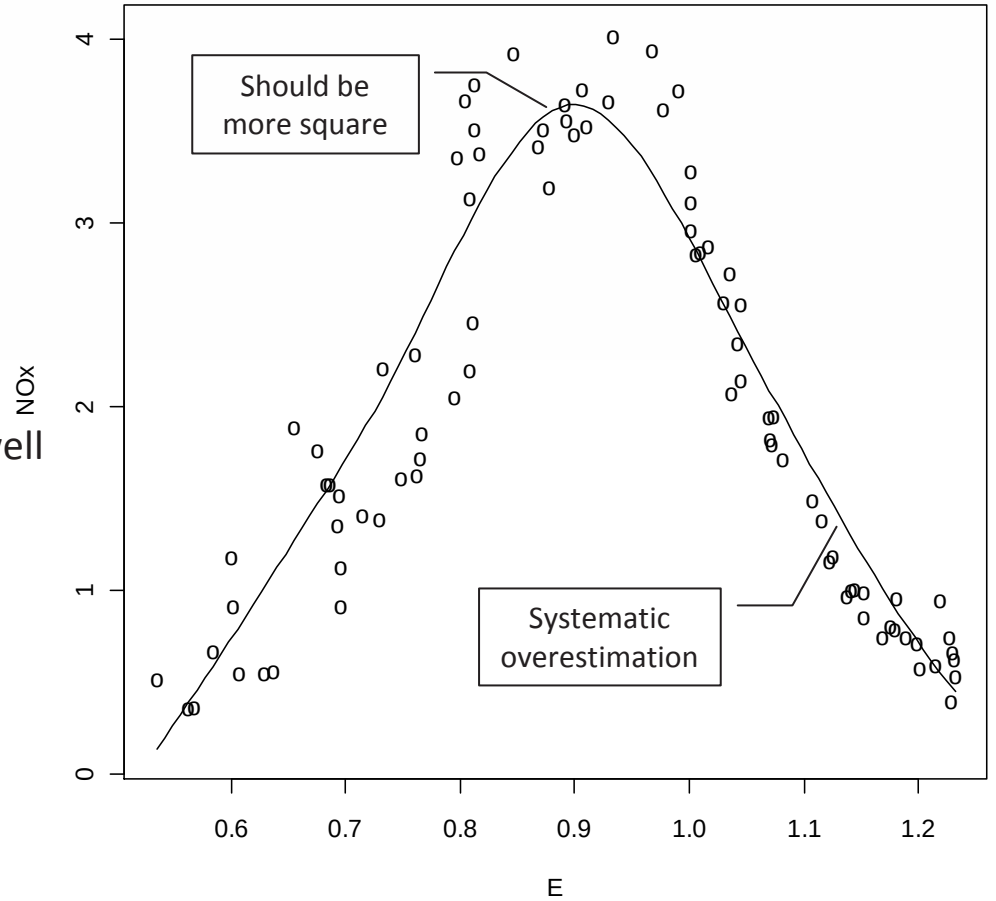
    (score must be calibrated and scaled to match the linear predictor)

Tuneable smoothing parameters

- Want the curve to accurately fit all real features in the data
- Smoothing parameters control the bias-variance trade-off

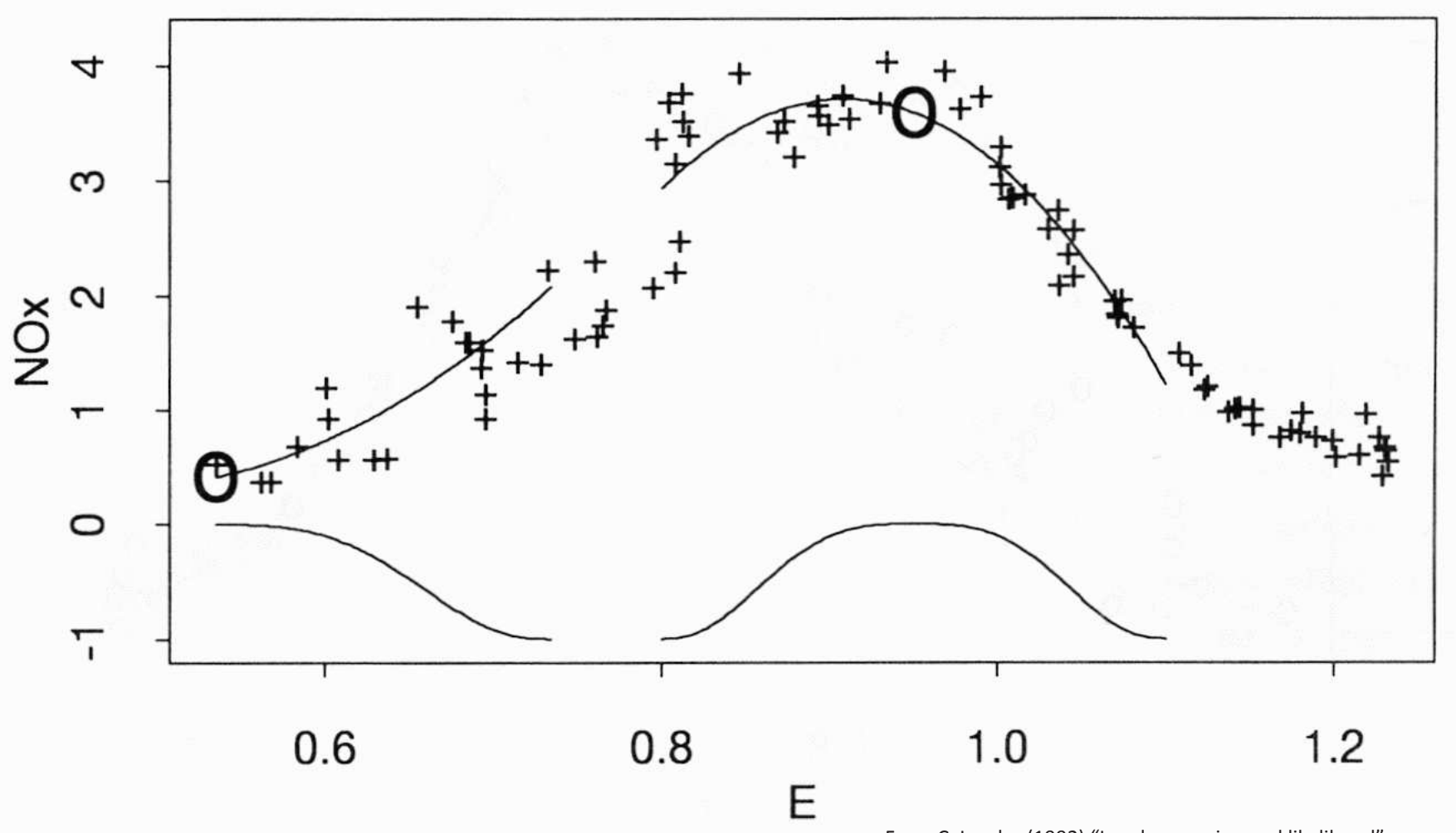
- Fit is locally linear (so far)
- Wide fitting window may not fit curves well
- Narrow fitting window tracks data better but more sensitive to noise

- The local fit can be a polynomial of *x* (usually ≤ 3rd order)
- Higher order tracks data better but more sensitive to noise



From W.S. Cleveland (1993) "Visualizing data"

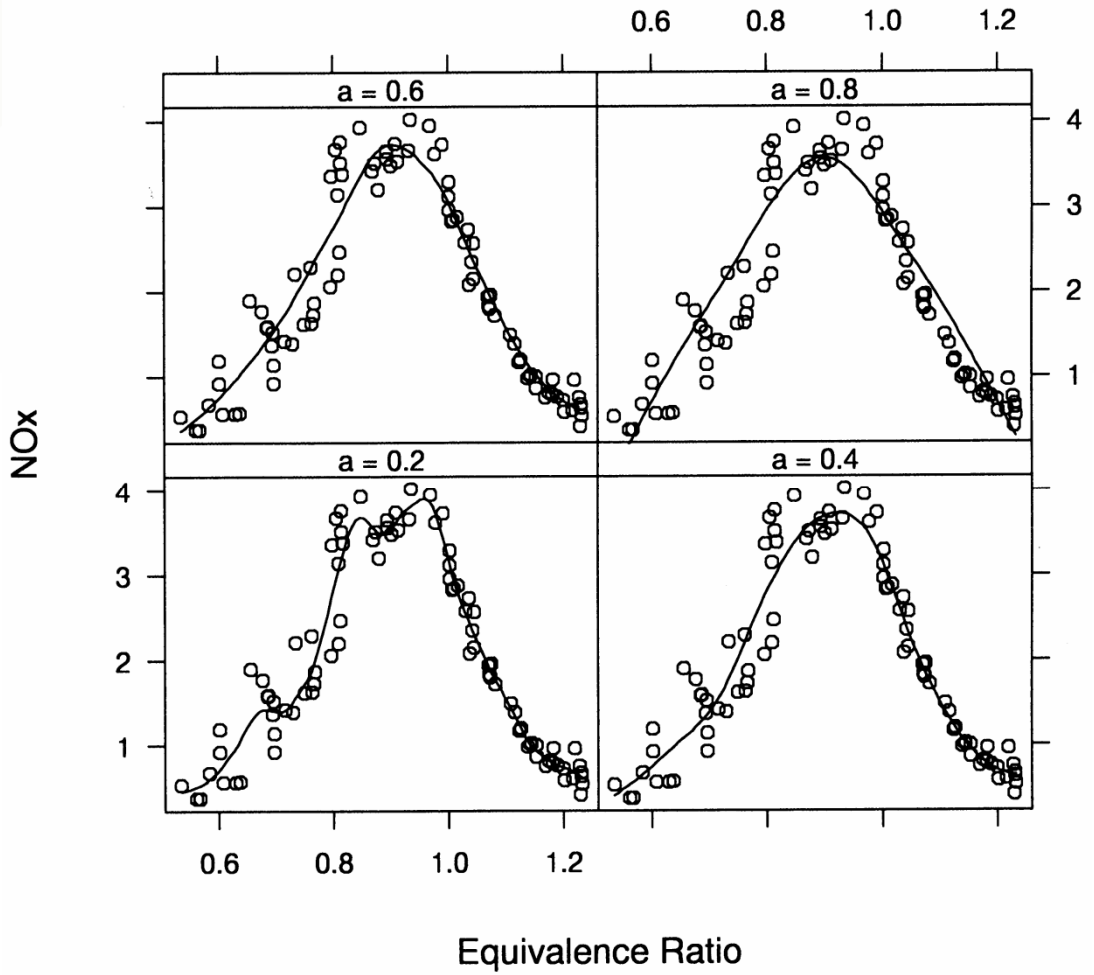
Local polynomial fit



From C. Loader (1992) “Local regression and likelihood”

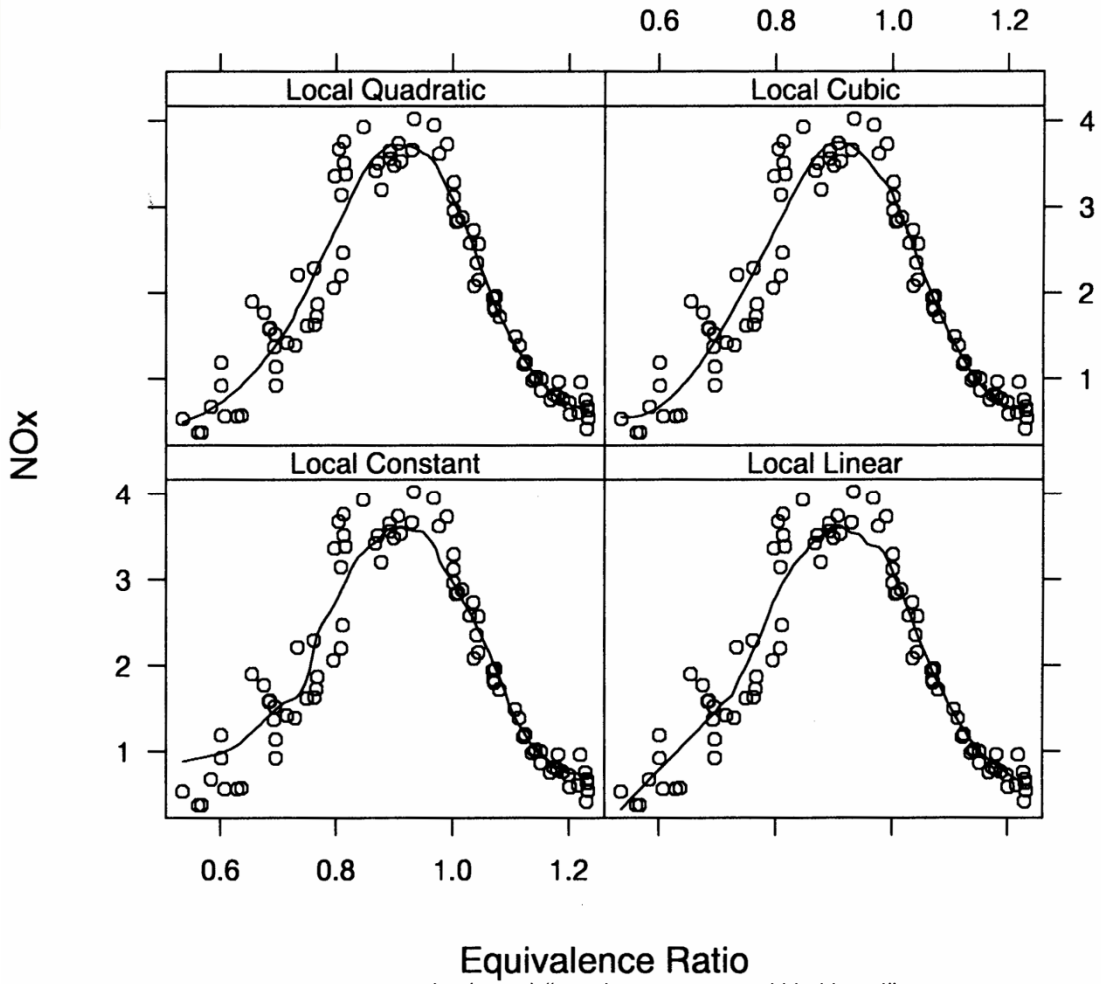
Varying the width of the fitting window

Typically specified as a fixed fraction of the observations



From C. Loader (1992) "Local regression and likelihood"

Varying the local polynomial degree



From C. Loader (1992) "Local regression and likelihood"

Setting the smoothing parameters

- Constant default parameters unlikely to be optimal
- Can use diagnostics to manually choose parameters
  - Time consuming manual effort
- Choose global smoothing parameters by search over cross-validation performance
  - No guarantee that constant smoothing parameters are optimal across the range of *x*
  - Special values create local discontinuities
    - Implies smaller fitting window and/or higher order polynomial at that point

- Possible to use visual diagnostics to identify local lack of fit
- Partition the fit into smooth and discontinuous terms by adding indicators to the formula (smooth fit no longer responsible for fitting the discontinuities)
  - Time consuming manual effort

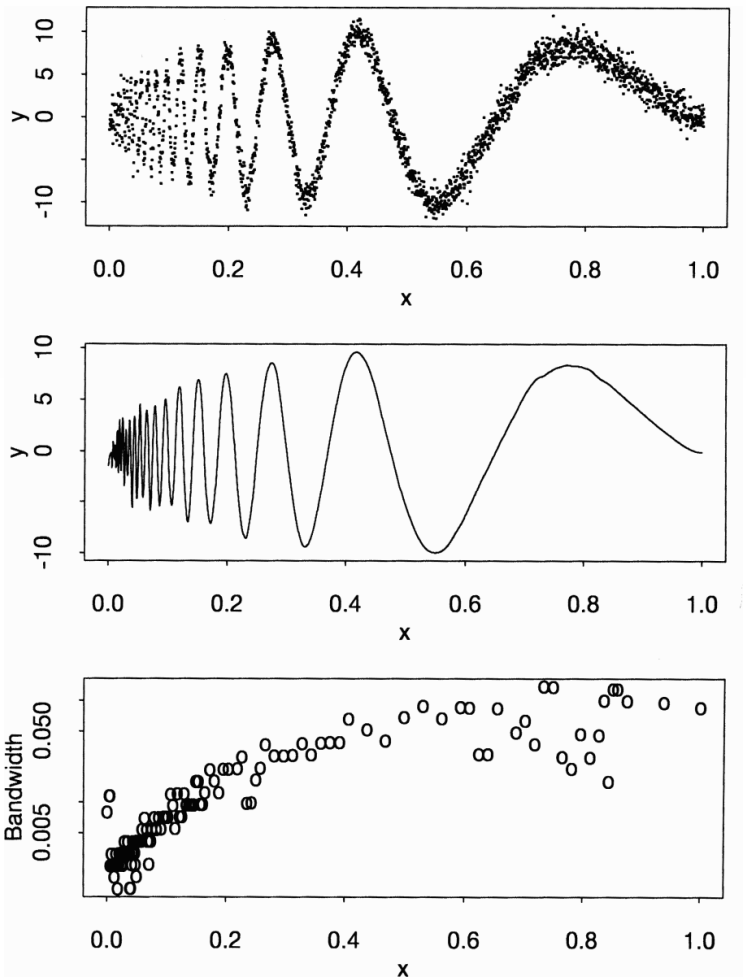
Adaptive smoothing parameters

Automated setting of local parameters

Local smoothing parameters

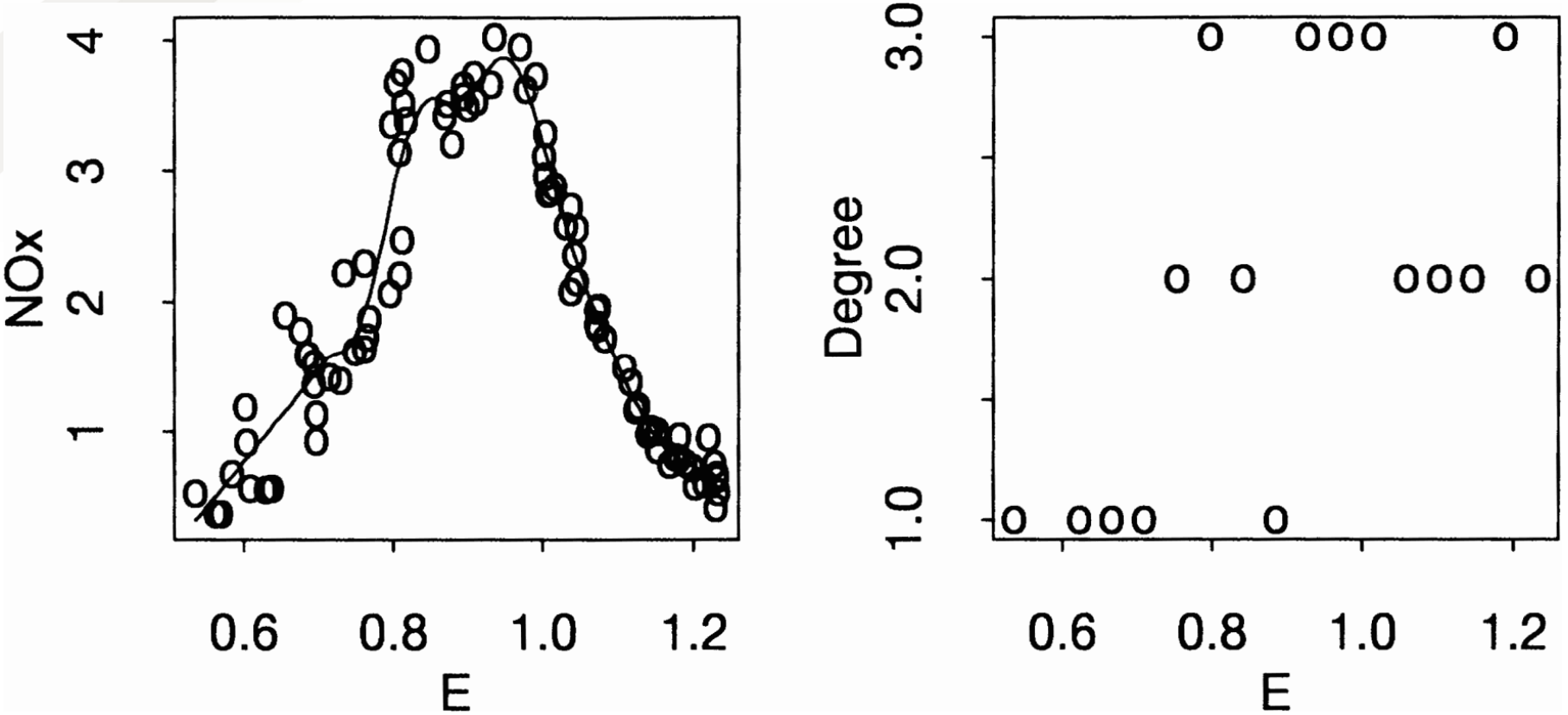
- The smoothing parameters can vary between windows (i.e. locations on *x*)
- Choose local smoothing parameters by automated search over cross-validation performance

Locally adaptive fitting window width



From C. Loader (1992) "Local regression and likelihood"

Locally adaptive polynomial degree



From C. Loader (1992) "Local regression and likelihood"

veda
applied intelligence

y scaling

Facilitating comparison

Scaling the *y* axis for comparability

- Local regression is logistic, so prediction is scaled as natural log of absolute odds
  - Compatibility with predictions from other logistic regressions (e.g. scorecard model)
  - Absolute (rather than relative to portfolio) makes comparison to known log-odds easier
- Add a secondary axis for relabelling the scale
  - Supply an arbitrary function to map natural log-odds onto secondary scale, e.g.:
    - Subtract portfolio log-odds gives WOE
    - Linearly rescale to points (compatible with score scale)
- Add a secondary curve, e.g. score (able to be plotted because of known scale)
  - Supply arbitrary function to map another variable (e.g. score) to the log-odds scale
    - Identity function if secondary data is natural log-odds score
    - Inverse of secondary axis function if score is scaled in points
  - Secondary data is transformed by the supplied function and smoothed w.r.t. *x*
  - Smoothed curve is added to characteristic analysis graph

veda
applied intelligence

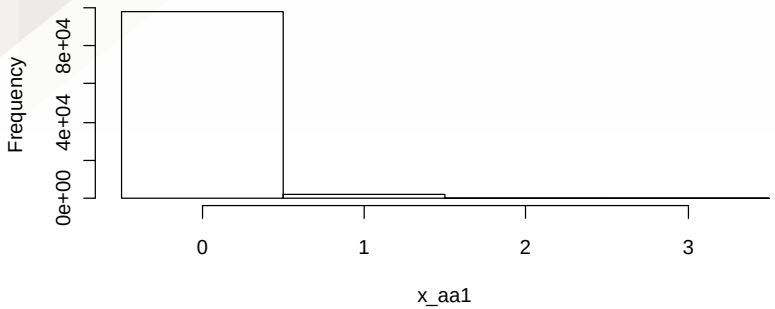
x jitter

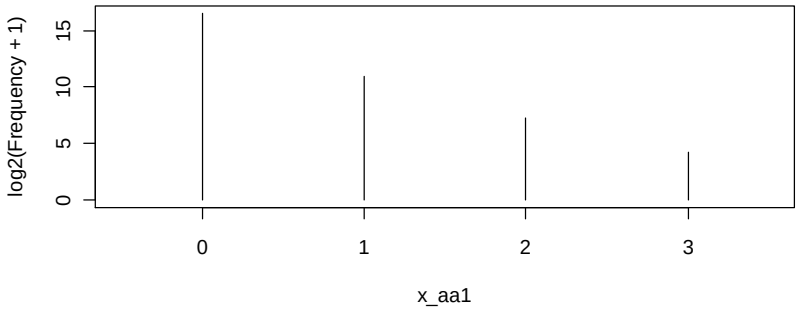
Small windows and gaps

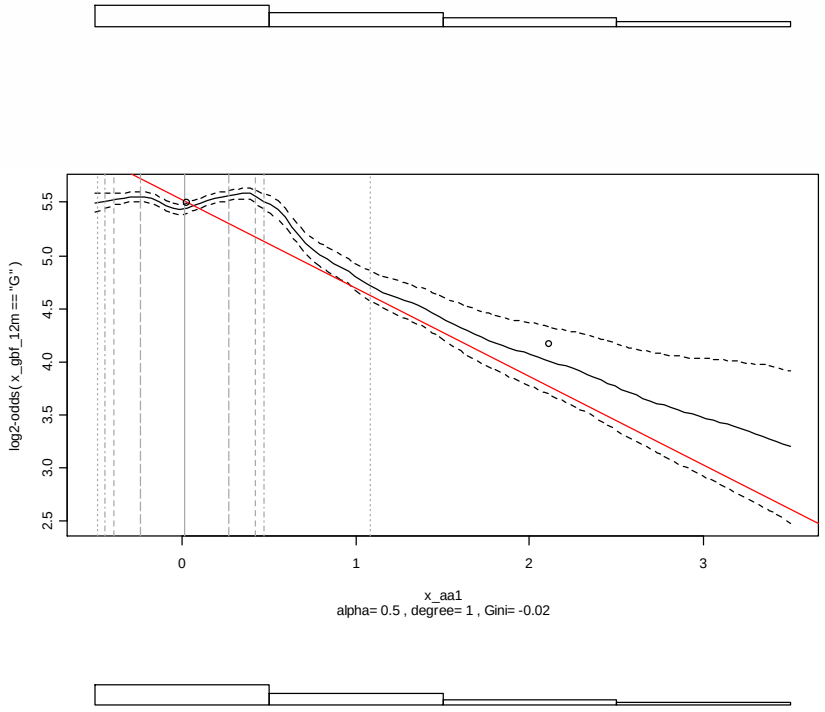
- "continuous" values are typically recorded to a fixed resolution (e.g. age in whole years)
  - Sometimes a high proportion of observations have exactly the same *x* value
  - There are gaps between the observed values (an encoding artefact, gaps don't exist))
- Fitting window width usually specified as a fraction of the observations
  - Popular *x* value may have more observations than the fitting window width
  - No *x* variance in the fitting window (local regression fails)
- Local regression carried out on the scale of *x* (so far)
  - Confidence intervals become wider in the gaps (interferes with visual interpretation)
- Solution - Add jitter (uniformly distributed random noise) to the *x* values before smoothing
  - Identical *x* values are no longer identical in the local regression window
  - Jitter magnitude chosen to be ± 0.5 minimum difference between adjacent *x* values
    - Gaps between equally spaced observations are completely filled

A not very continuous predictor

- A continuous variable with 4 values
- 95% of observations on one *x* value







veda
applied intelligence

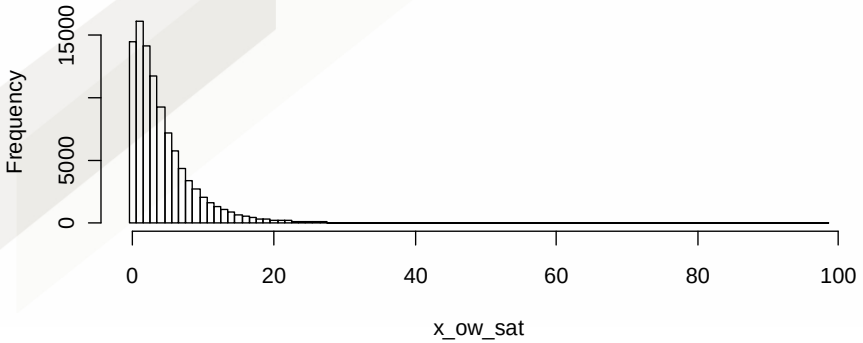
x transformation

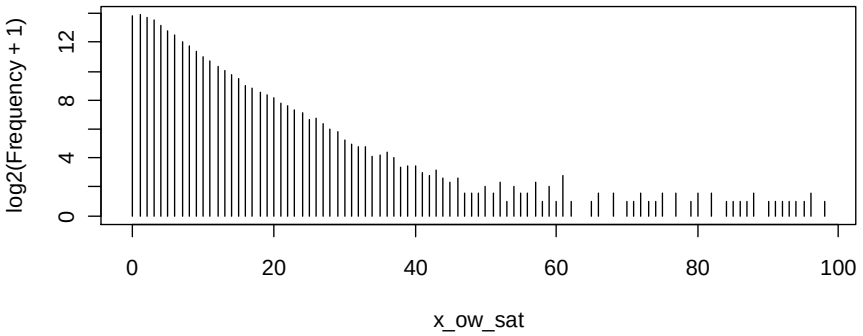
Nasty distributions of *x*

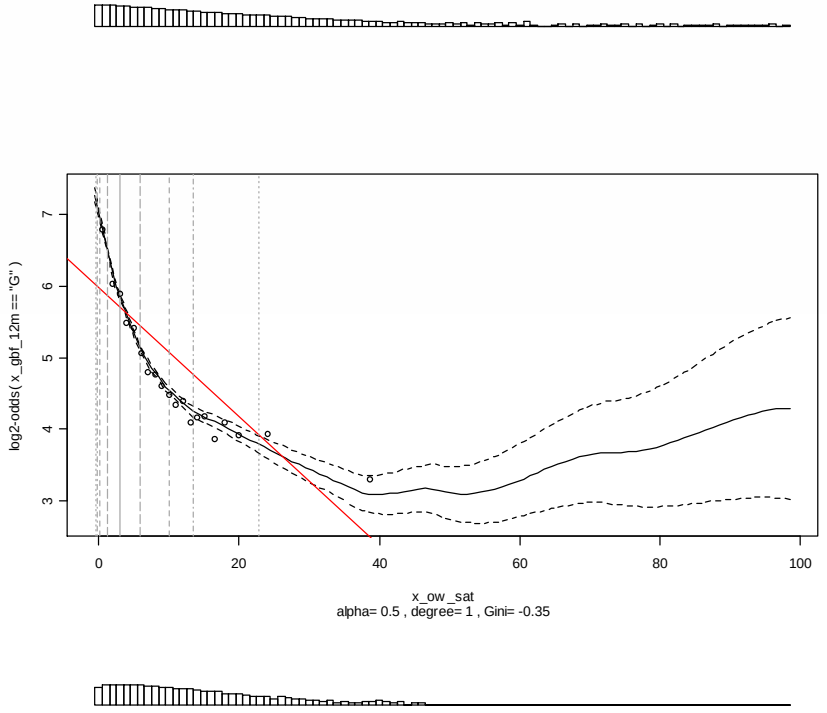
- Some characteristics are very skewed or unpleasantly distributed in other ways
  - Extreme leverage of extreme *x* values
  - Smoothing window becomes 1-sided as it approaches the extrema of *x*
  - Can get qualitatively different results depending for different transformations of *x*
    - Unwanted arbitrariness in the results

- Do the results reflect the true pattern (i.e. what would be seen in the infinite data case) or reflect an artefact of the fitting procedure?

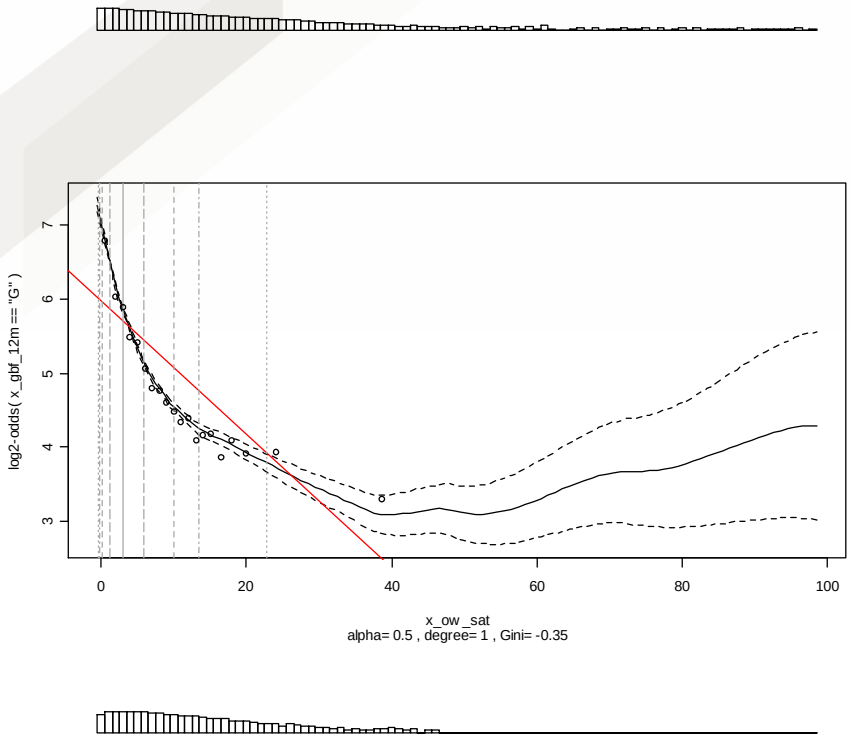
A very long-tailed distribution

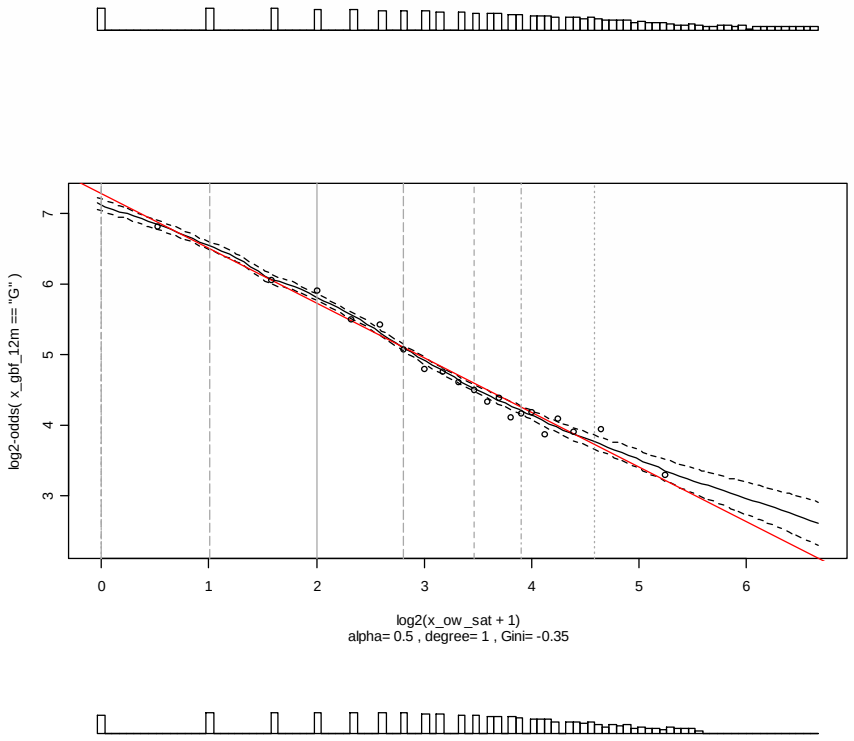






Differing results by transformations of *x*



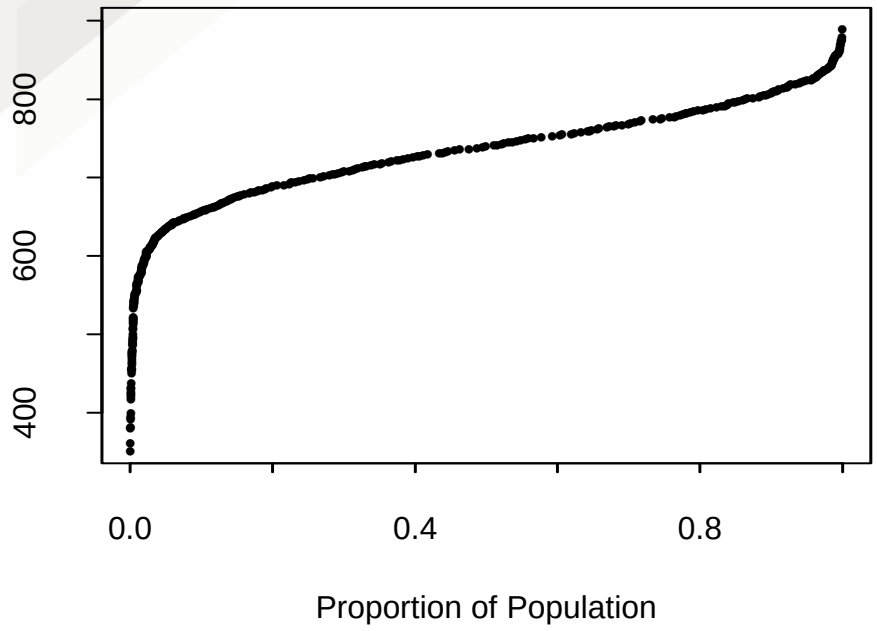


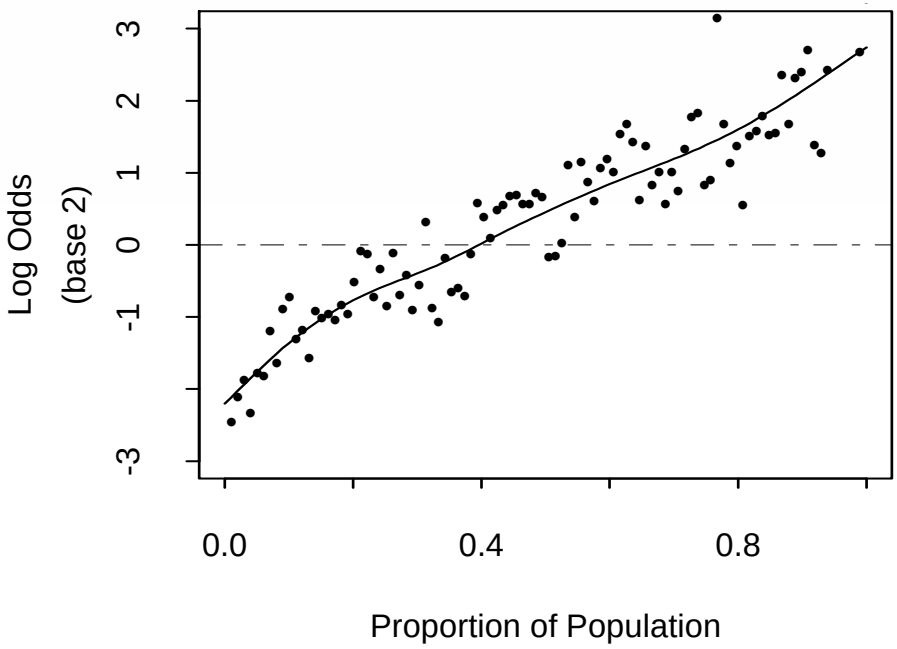
Apply a canonical transformation of *x*

- Transform to uniformly distributed *x* before smoothing
  - Estimate the cumulative distribution function (CDF) of *x*
    - Empirical CDF is discontinuous and has gaps
      - Problem if treating CA as a model (potentially applying it to new data)
      - Estimate CDF with a smoother; add a small uniform PDF to fill gaps; renormalise
  - *x*' = CDF(*x*)
- Removes impact of skewness on fitting
- Emphasises patterns over larger proportions (we don't care about very small proportions)
- Acts as a low pass filter (a very steep curve is better modelled as a discontinuity)

- Display CA graph on transformed *x* axis
  - Equal proportions of population occupy equal space on the graph
    - Better aligned to the analyst's view of practical relevance

Transform *x* for smoothing and display





veda
applied intelligence

y jitter

Small windows with respect to *y*

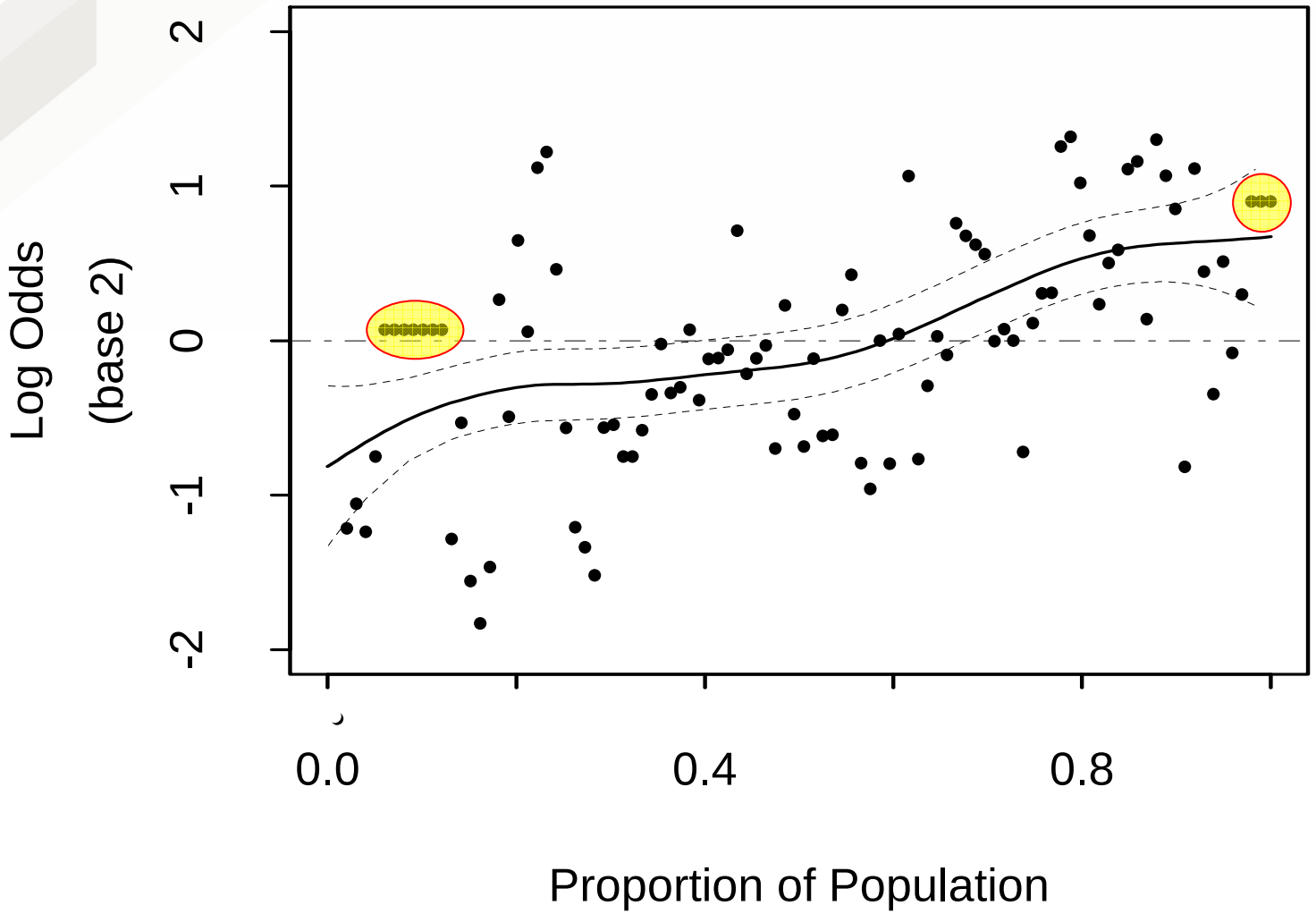
- Sometimes there is only one outcome type observed in a smoothing window
  - For example: mortgages, accept/reject modelling
  - No variance with respect to the outcome (local regression fails)
- One method to address this – Randomly flip a fixed proportion of the outcomes
  - Equivalent to linear shrinkage on probability scale towards 0.5
    - Zero impact at p = 0.5, Maximal impact at p = 0 or 1
  - Nonlinear effect on log-odds scale (but qualitatively same pattern)
  - Smooth using *y*-jittered data
  - Transform the smoothed function back to the unjittered scale
    - Fine for probability scale (p = 1 ends up at p = 1)
    - Still end up with infinities on log-odds scale (display y-jittered log-odds?)
- Issues
  - Throwing away some data, so confidence intervals are wider
  - How to choose smallest flip probability so that no smoothing window is p = 0 or 1

Unsmoothed diagnostics

Unsmoothed visual diagnostics

- Need an unsmoothed estimate of the characteristic analysis to compare with smoothed CA
  - Looking for lack of fit (e.g. at discontinuities)
- Divide *x* into bins and calculate empirical mean *y* (We use two alternative binning strategies)
  - Fixed, non-adaptive binning
  - Adaptive binning
- Fixed, non-adaptive binning
  - Use quantile bins (to be consistent with transformed *x* display of characteristic analysis)
  - Want relatively many bins (so we can track details of the curve)
    - Usually centiles (credit scorers tend not to care about segments smaller than 1%)
  - How to handle x values with more than 1% of observations?
    - Jittered *x* to the rescue
    - Linear interpolation on empirical CDFs of each outcome type with knots at all unique *x* values (virtual *x*-jittering)
      - Centiles from the same *x* value have the same *y* value (more obvious on the display)

Fixed, non-adaptive binning



Adaptive binning

- Interested in identifying where the smoothed CA doesn't fit
- Focus on areas of lack of fit
- Base bins on residuals from smoothed fit
  - Calculate residuals
  - Build a recursive partitioning tree with *x* as predictor and residuals as outcome
  - Use leaf definitions as bins
  - Calculate empirical mean *y* per bin
- Obviously biased to make the smooth fit look as bad as possible

Future work

Future work

- Lots of loose ends around good choice of tuning parameters
- Integration of all techniques into good display(s)
  - Convey all relevant information
  - Avoid overcrowding the display
- User interface for specifying modelling of special values
- Integration into bulk scanning interface
  - Never crash regardless of data
  - One at a time display production
  - Storage of all results into a single data structure for calculating statistics over the set of characteristic analyses

Thank you

Source: http://www.flickr.com/photos/bibliodyssey/3085763437/sizes/o/in/photostream/ with permission of peacay