

Applications of Nonlinear Regression Methods in Insurance

Agenda

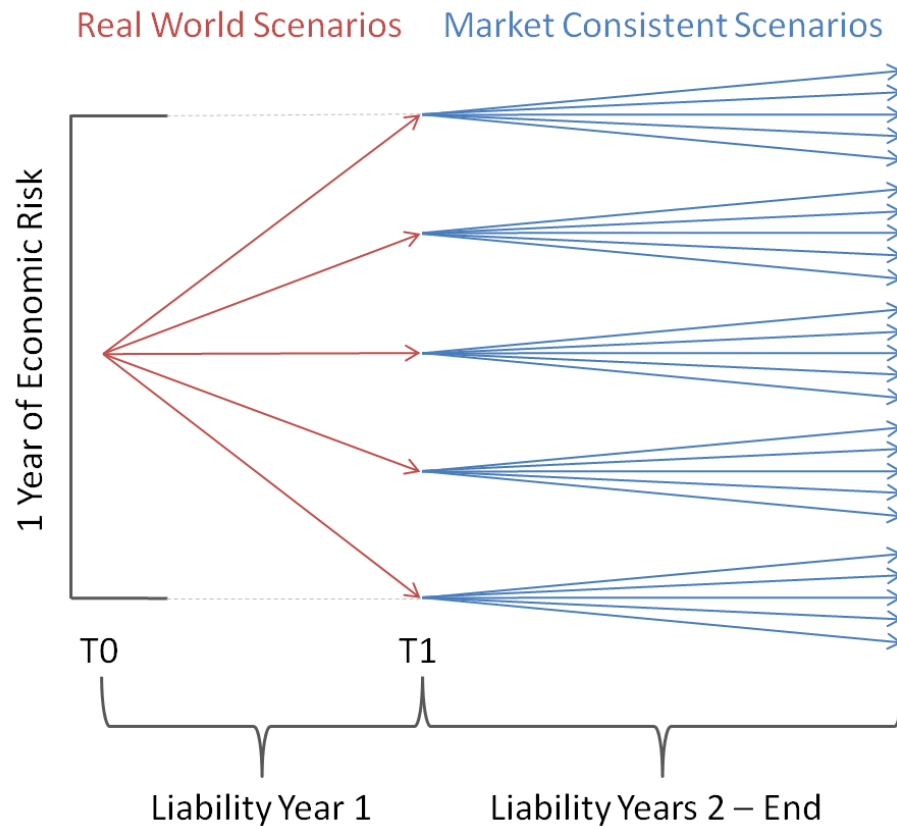
1. **Features of a Good Proxy**
2. **Multiple Polynomial Regression**
3. **Artificial Neural Networks**
4. **Motivating Example**
5. **Neural Network Analysis in more detail**
6. **Conclusion**

Motivation

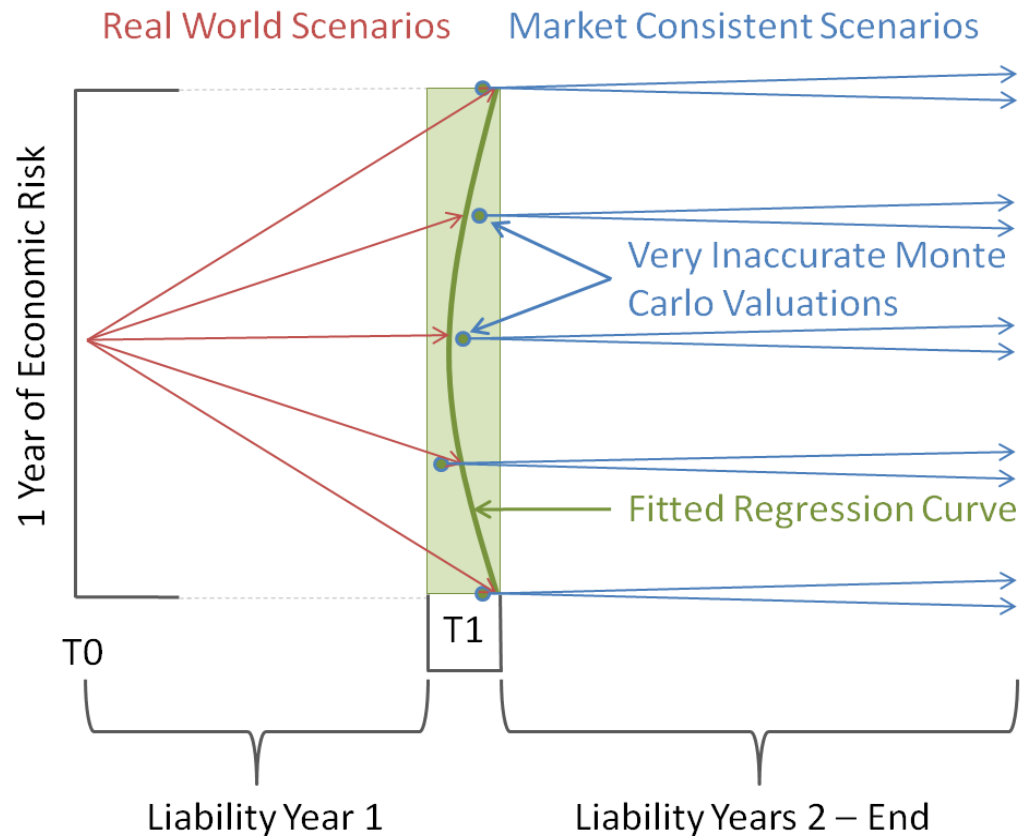
- » Proxy Generator uses multiple polynomial regression in LSMC which
 - is a well known and robust statistical method
 - has great intuitive appeal
 - has straight-forward formulae
 - uses a simple forward stepwise approach to find a “best” model
- » Many proxy generation problems can successfully rely upon polynomials
- » In our experience, we do see a small number of problems which are more challenging
- » To avoid too much analyst intervention for the more challenging fits when hundreds of proxies are needed, is there an alternative regression technique we can rely on?
- » In this presentation we ask “what other techniques are out there?”

Nested-stochastic simulations

Solvency 2 Regulations Require a “downside risk” measurement



Least Squares Monte-Carlo Solution



Features of a Good Proxy

Features of a Good Proxy: I

» *Parsimony*

- It should use a minimally sufficient set of risk drivers (including powers and cross terms)

» *Compatibility with downstream software*

- Ease of communication with downstream software
- It should use a relatively small number of parameters in a succinct representation

» *Good validation on “accurate” Validation Scenarios*

» *High goodness-of-fit measure without over-fit*

- The in-sample R-squared should be as high as possible
- The out-of-sample R-squared should be as close as possible to the in sample R-squared

Features of a Good Proxy: II

- » *Unbiased predictions of minimum variance*
 - Any evidence of systematic over- or under-estimation in the model predictions is evidence of bias
 - This often involves trading bias against variance in finding an optimal estimator

- » *Scalability to high dimensions*
 - For large numbers of risk drivers and fitting scenarios, the memory requirements and the time taken can become considerable
 - When a large number of parameters are being estimated, their standard errors are large and our ability to recover a meaningful model is reduced

- » *Short model fitting time*

- » *Good model specification*
 - Proxy models which are well specified will be able to approximate arbitrarily closely the underlying data generation process, given enough fitting scenarios

Alternative Regression Methods for LSMC

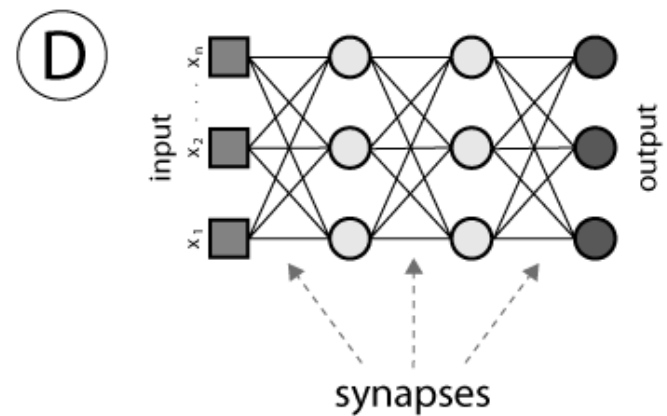
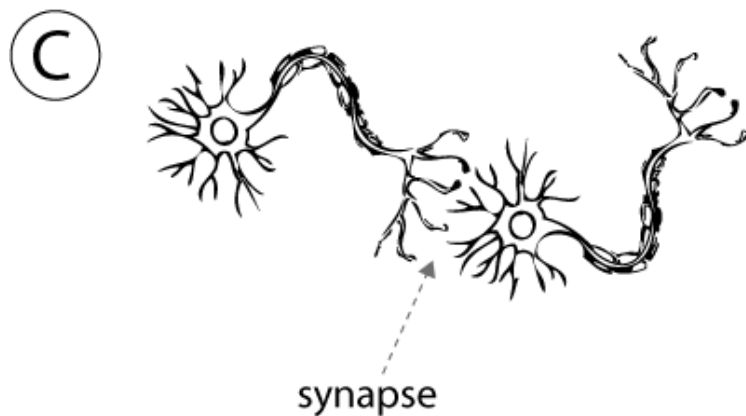
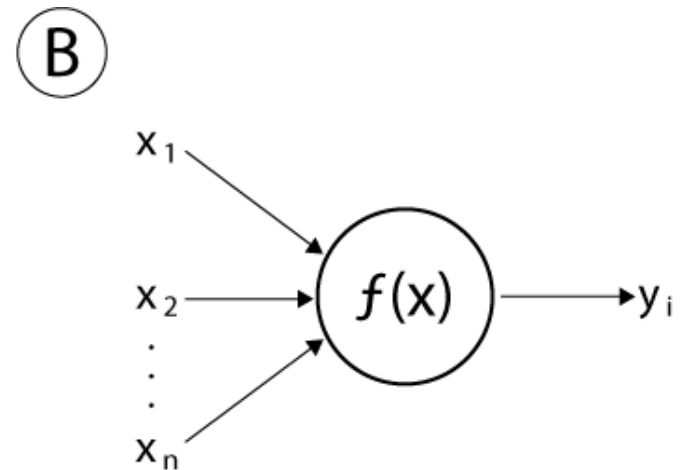
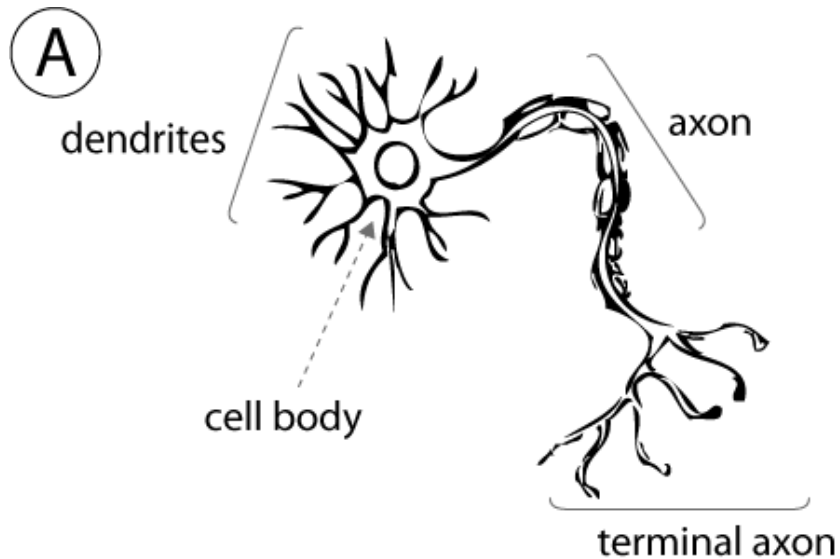
- » Examples of linear and nonlinear regression methods:
 - Mixed Effects Multiple Polynomial Regression
 - Generalized Additive Models
 - Artificial Neural Networks
 - Regression Trees
 - Finite Element Methods
- » In other work we have considered local regression methods such as
 - kernel smoothing and
 - loess / lowess
- » In this presentation we consider the merits of artificial neural networks

Artificial Neural Networks

Artificial Neural Networks

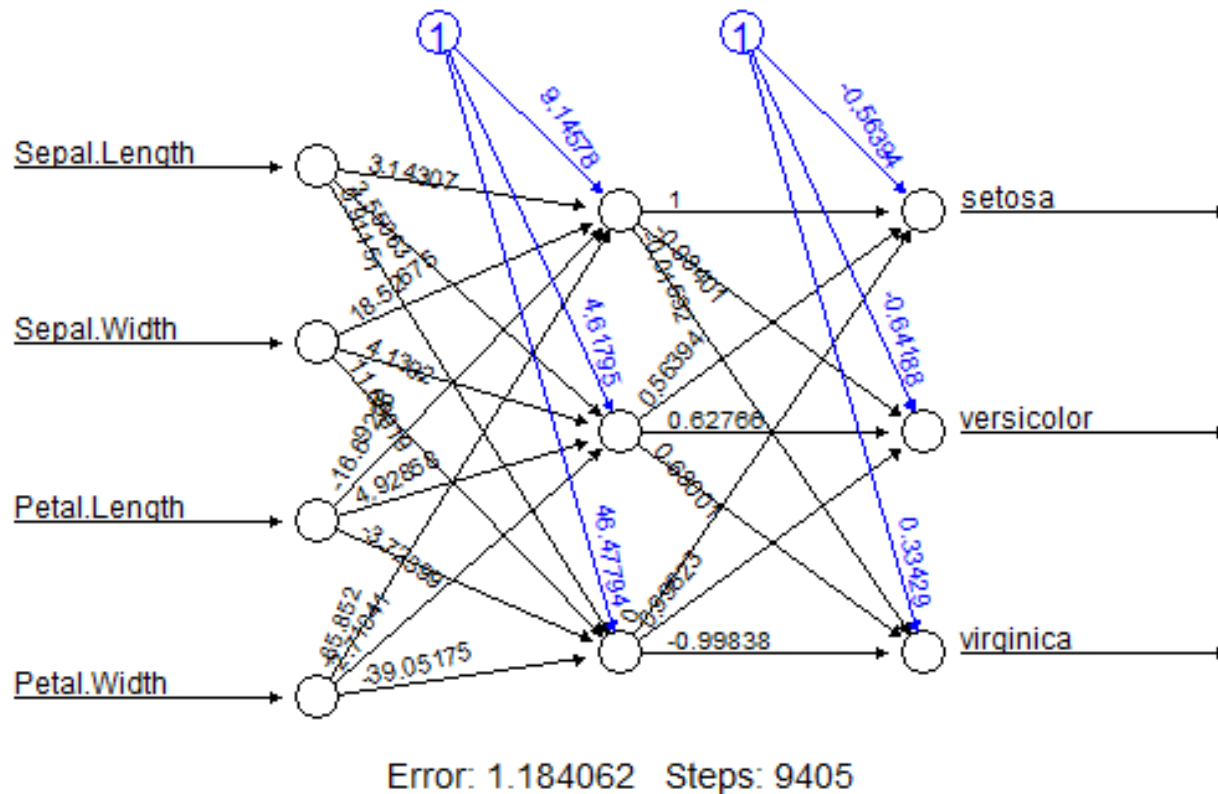
- » These were simultaneously invented by the computer science and statistics communities
- » They have a heritage of being used in
 - Classification problems such as in spam filters or shopping preferences, learning as they “see” more and more data
 - They are a natural alternative to logistic regression problems
 - They can also be used as nonlinear regression tools
- » They also have the unfortunate heritage of being known as “black-box” techniques with little intuitive appeal – they *just work*
- » They are often quoted as being *accurate* but subject to *over-fitting* at the same time
- » However, if we think of them as nonlinear regression tools then they are simple statistical constructs with parameters to be found by minimizing the mean squared prediction error

But what *is* a neural network?



Neural Network Structure

Input layer / hidden layer / output layer



Formulae

Both multiple polynomials and neural networks have similar functional forms

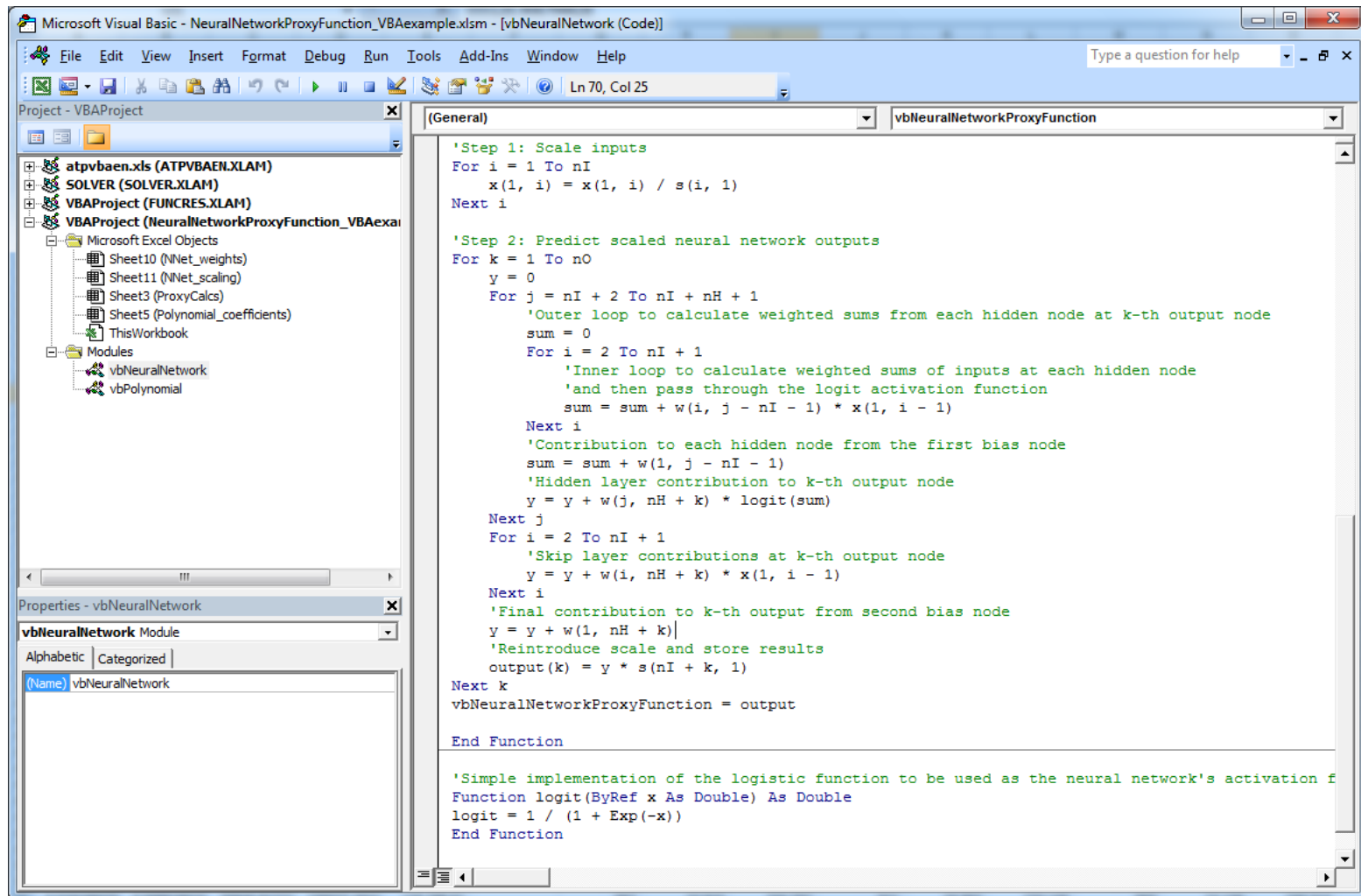
Polynomial Regression

$$y_k = \beta_0 + \sum_{i=1}^p \beta_i^{(1)} x_i + \sum_{i=1}^p \sum_{j=1}^i \beta_{ij}^{(2)} x_i x_j + \text{higher order terms}$$

Neural Network and Activation Function

$$y_k = \alpha_k + \sum_{i \rightarrow k} w_{ik} x_i + \sum_{j \rightarrow k} w_{jk} \varphi \left(\alpha_j + \sum_{i \rightarrow j} w_{ij} x_i \right), \quad \varphi(x) = \frac{1}{1 + \exp(-x)}$$

VBA Implementation



Fitting a Neural Network

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains the R script for fitting a neural network.
- Console:** Shows the execution output of the script.
- Workspace/History:** Empty.
- Files/Plots/Packages/Help:** The Help pane is active, showing the documentation for the `nnet` package.

Source Editor Code:

```
> net2<-nnet(x,y,size=2,decay=1e-3,
skip=TRUE,linout=T,maxit=1000,trace=TRUE,rang=0.1)
# weights: 8
initial value 44.040561
iter 10 value 13.952072
iter 20 value 10.725342
iter 30 value 10.449023
iter 40 value 10.435875
iter 50 value 10.431929
iter 60 value 10.431642
final value 10.431362
converged
> summary(net2)
a 1-2-1 network with 8 weights
options were - skip-layer connections linear output units decay=0.001
b->h1 i1->h1
0.84 -0.93
b->h2 i1->h2
-0.36 -1.82
b->o h1->o h2->o i1->o
0.02 1.13 -1.89 0.00
> Rsq(y,fitted.values(net2),p=8)
R2 R2a
0.5856764 0.5827557
> |
```

Console Output:

```
> net2<-nnet(x,y,size=2,decay=1e-3,
skip=TRUE,linout=T,maxit=1000,trace=TRUE,rang=0.1)
# weights: 8
initial value 44.040561
iter 10 value 13.952072
iter 20 value 10.725342
iter 30 value 10.449023
iter 40 value 10.435875
iter 50 value 10.431929
iter 60 value 10.431642
final value 10.431362
converged
> summary(net2)
a 1-2-1 network with 8 weights
options were - skip-layer connections linear output units decay=0.001
b->h1 i1->h1
0.84 -0.93
b->h2 i1->h2
-0.36 -1.82
b->o h1->o h2->o i1->o
0.02 1.13 -1.89 0.00
> Rsq(y,fitted.values(net2),p=8)
R2 R2a
0.5856764 0.5827557
> |
```

Help Pane: R: Fit Neural Networks

nnet {nnet} R Documentation

Fit Neural Networks

Description

Fit single-hidden-layer neural network, possibly with skip-layer connections.

Usage

```
nnet(x, ...)
```

S3 method for class 'formula'

```
nnet(formula, data, weights, ...,
      subset, na.action, contrasts = NULL)
```

Default S3 method:

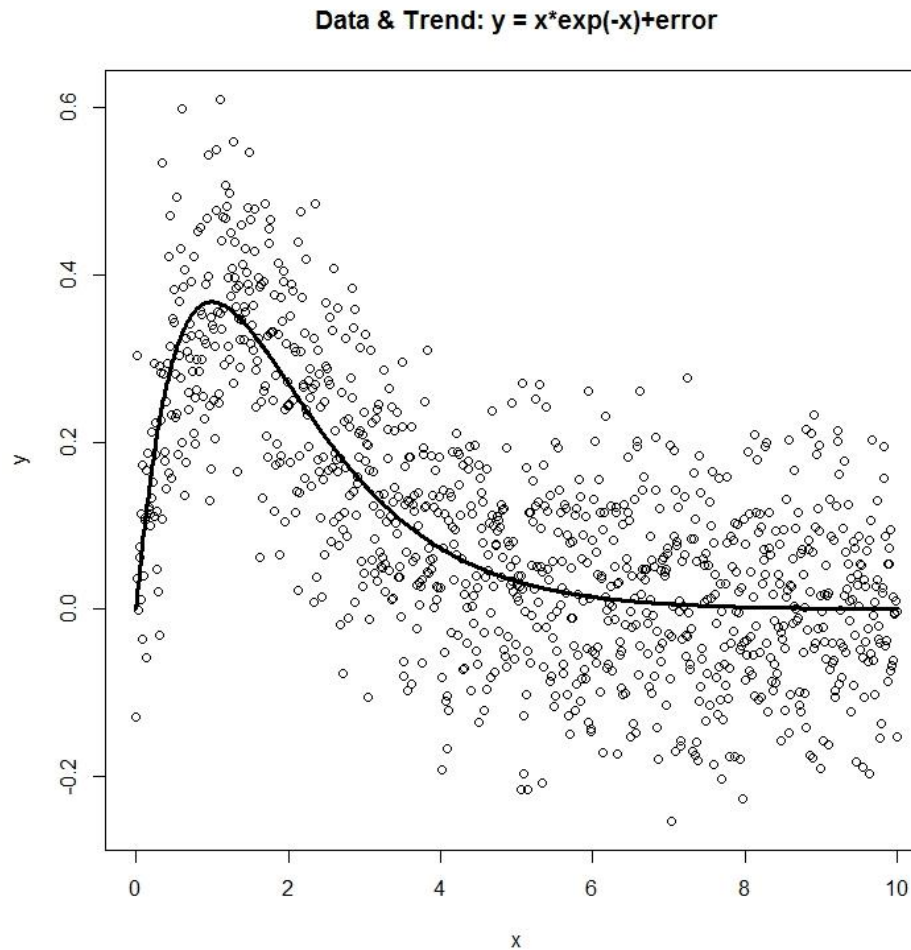
```
nnet(x, y, weights, size, Wts, mask,
      linout = FALSE, entropy = FALSE, softmax = FALSE,
      censored = FALSE, skip = FALSE, rang = 0.7, decay = 0,
      maxit = 100, Hess = FALSE, trace = TRUE, MaxNWts = 1000,
      abstol = 1.0e-4, reltol = 1.0e-8, ...)
```

Arguments

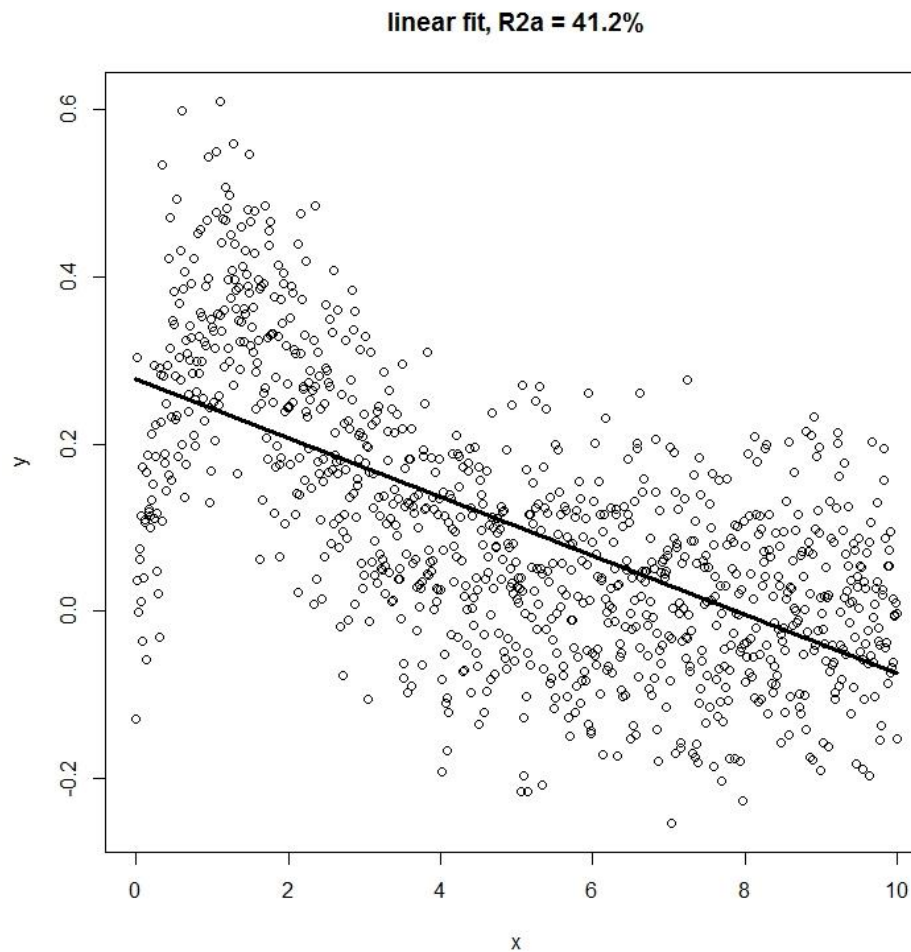
formula	A formula of the form <code>class ~ x1 + x2 + ...</code>
x	matrix or data frame of x values for examples.
y	matrix or data frame of target values for examples.

Nonlinear edge case example

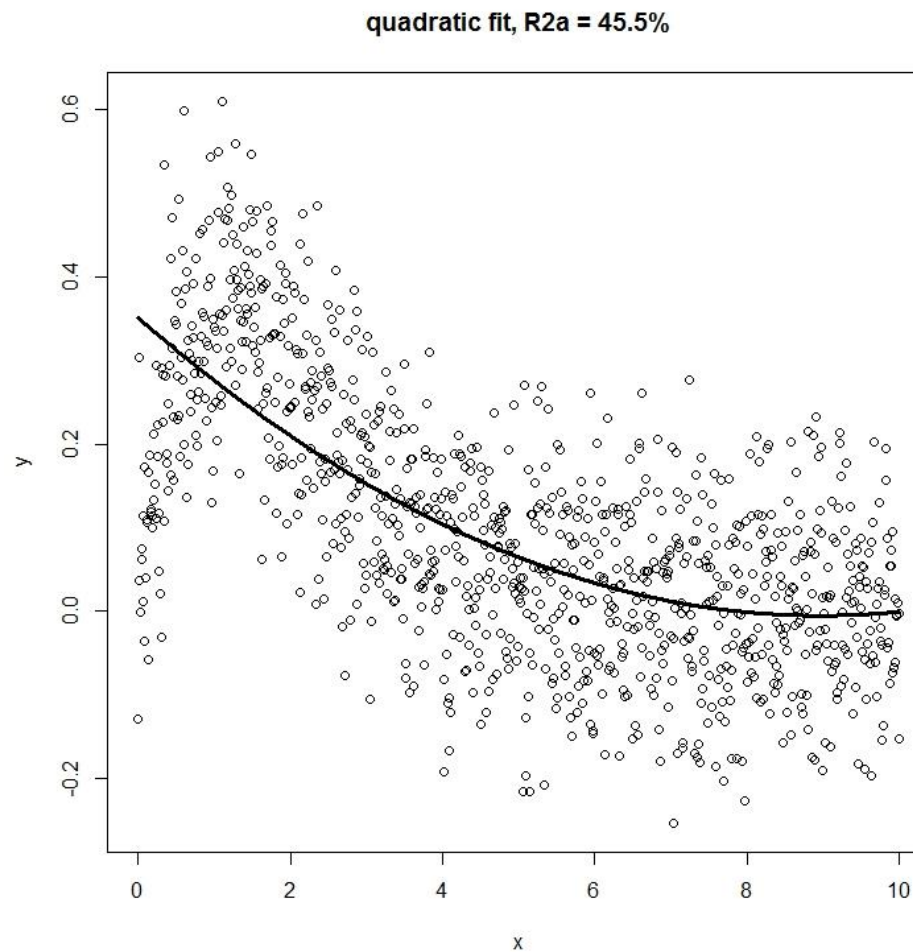
Example 1000 pairs x, y with normal errors (sd 0.1)



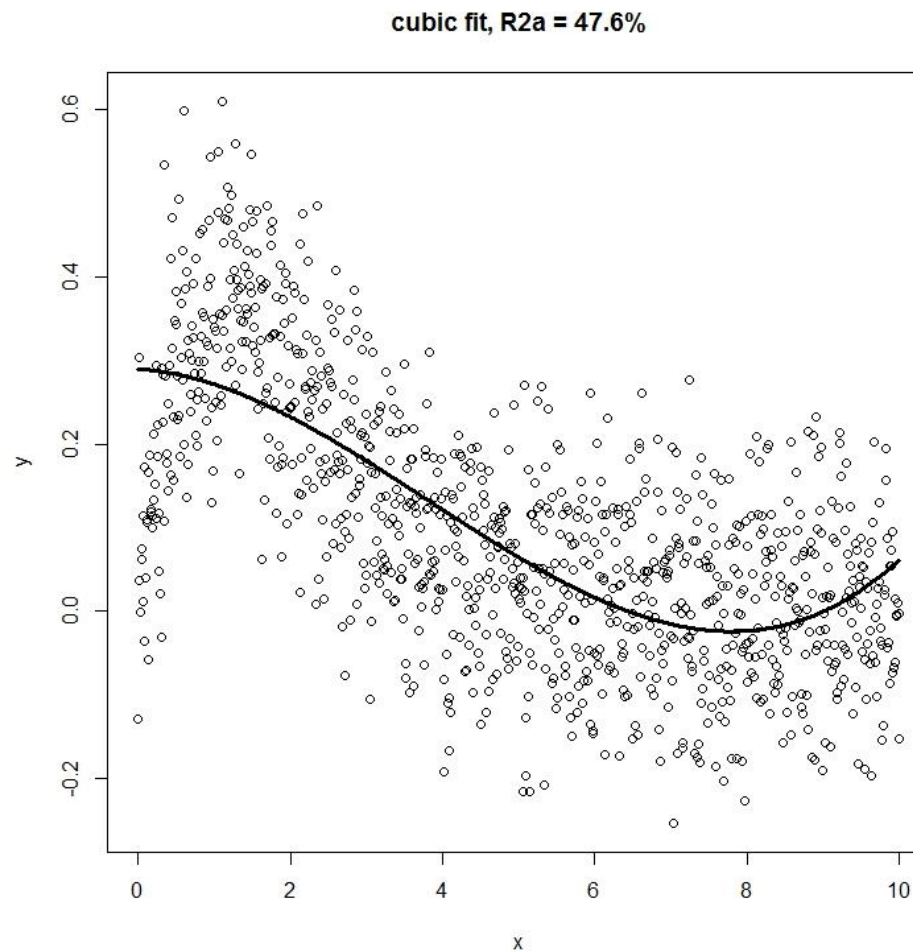
Degree 1 polynomial fit



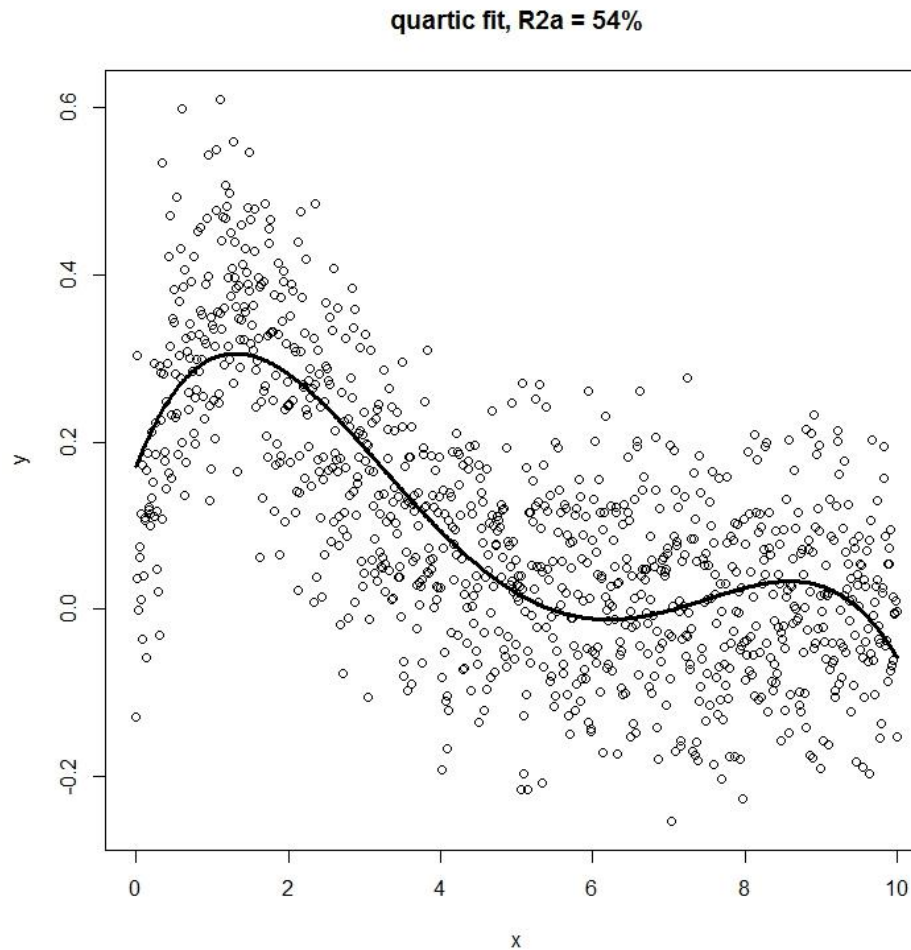
Degree 2 polynomial fit



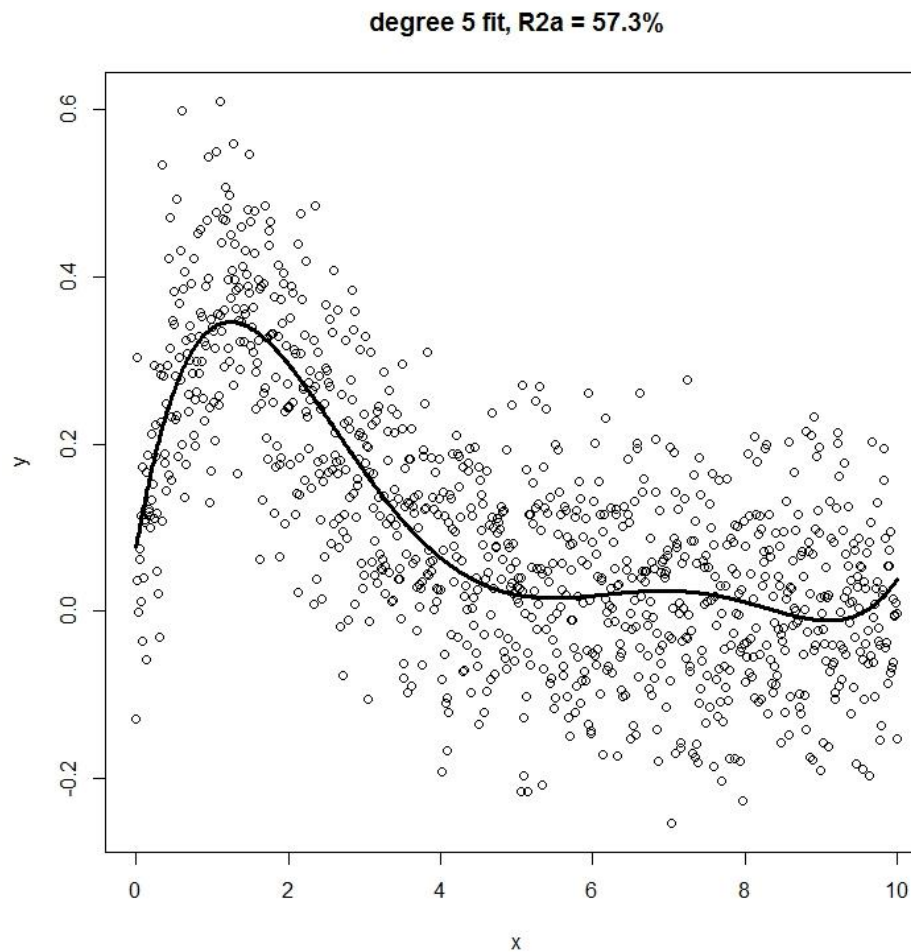
Degree 3 polynomial fit



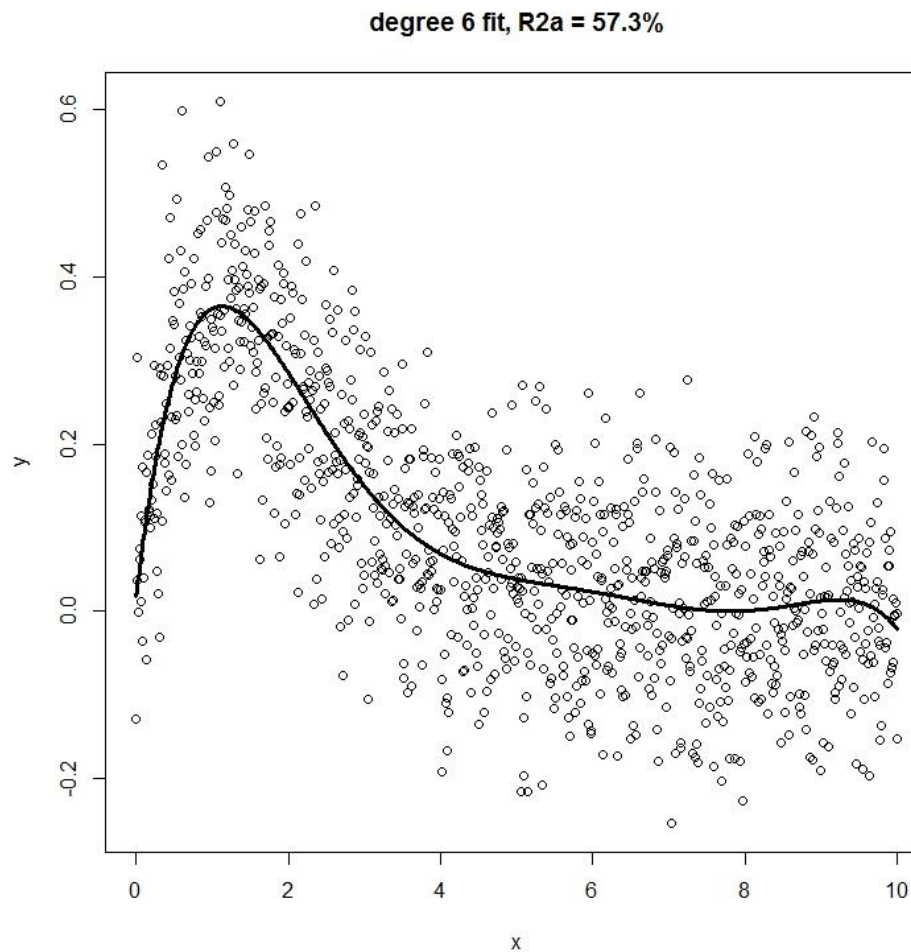
Degree 4 polynomial fit



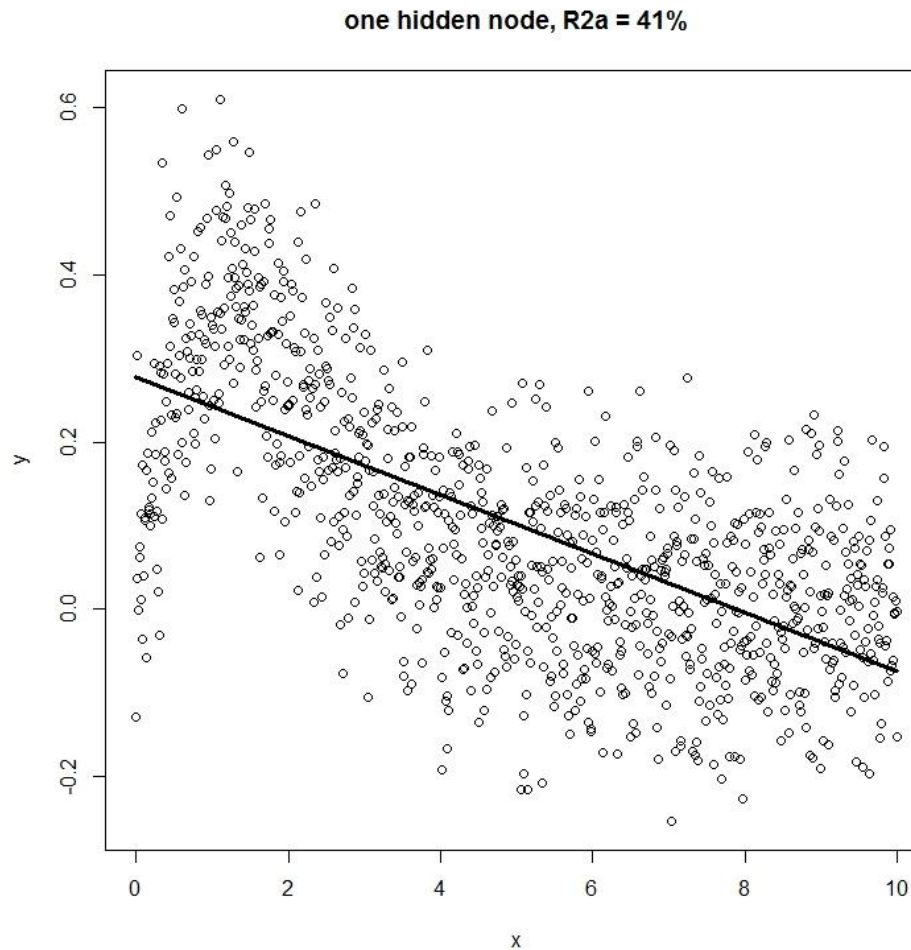
Degree 5 polynomial fit



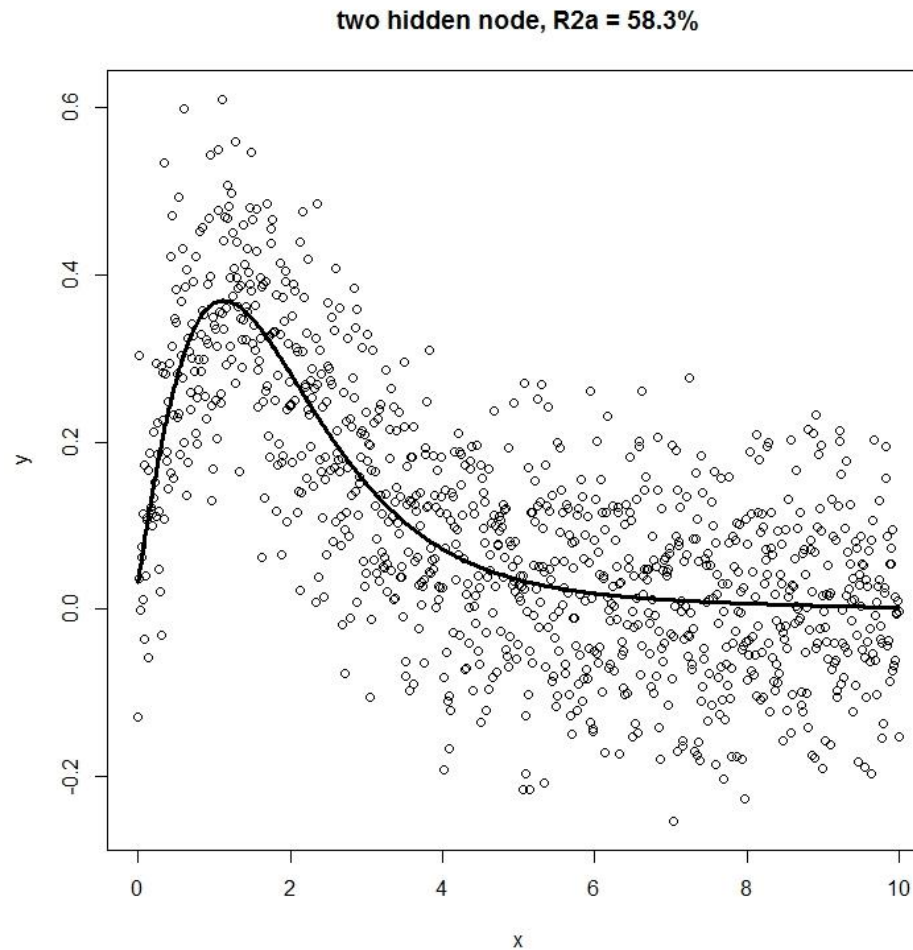
Degree 6 polynomial fit



Neural network one hidden node fit

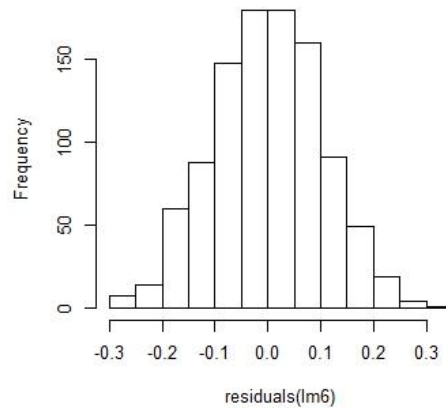


Neural network two hidden node fit

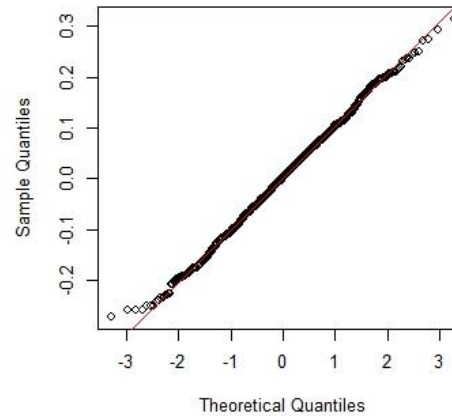


Residuals Analysis

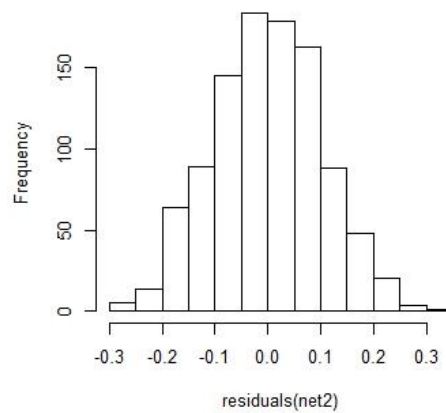
degree 6 polynomial



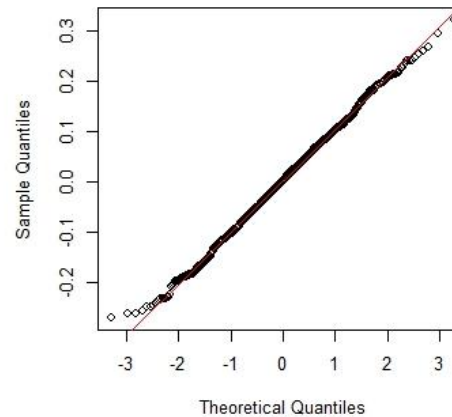
QQ plot



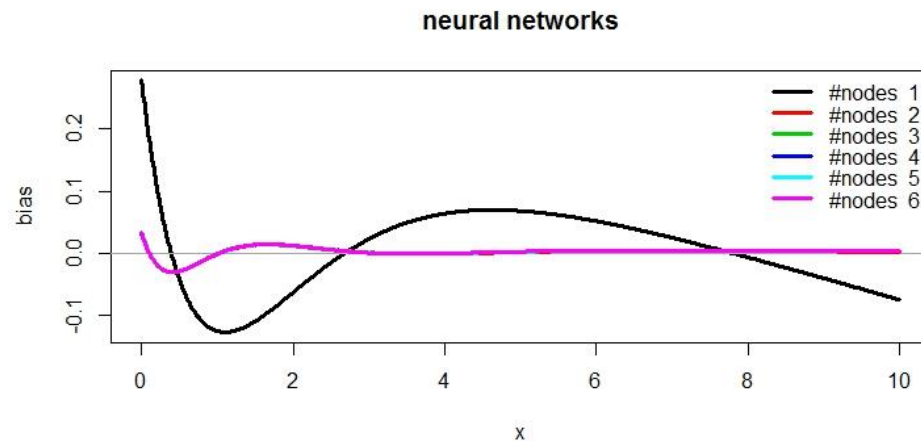
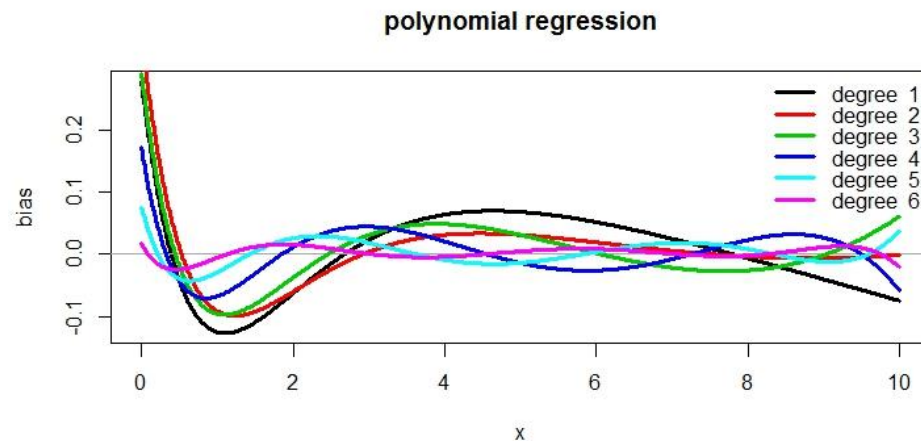
2 node neural network



QQ plot



Actual trend subtract the fit



Motivating example

Variable Liability Value

- » Life policy has an embedded guarantee of 3.25%
- » Involved 9 risk-drivers including equity level and volatility, real and nominal yield curve factors and credit in addition to some non-market risks.
- » The exercise was to model the liability in a single time-step / static regression problem.
- » Firstly, a multiple polynomial regression was performed
 - up to cubic degree in each risk-driver
 - using a layered forward stepwise approach
 - without term removal
- » Secondly, a neural network in 9 input nodes, a bias node, 2 hidden nodes and a skip layer connection was fitted to the same data.

Variable Liability Value (continued)

N=25,000	Regression	Neural Network
Time Taken (seconds)	3797 (1 hr. approx.)	75
Number of terms/weights	52	44
In sample R-squared	72.30%	69.38%
Out of sample R-squared	72.23%	69.28%

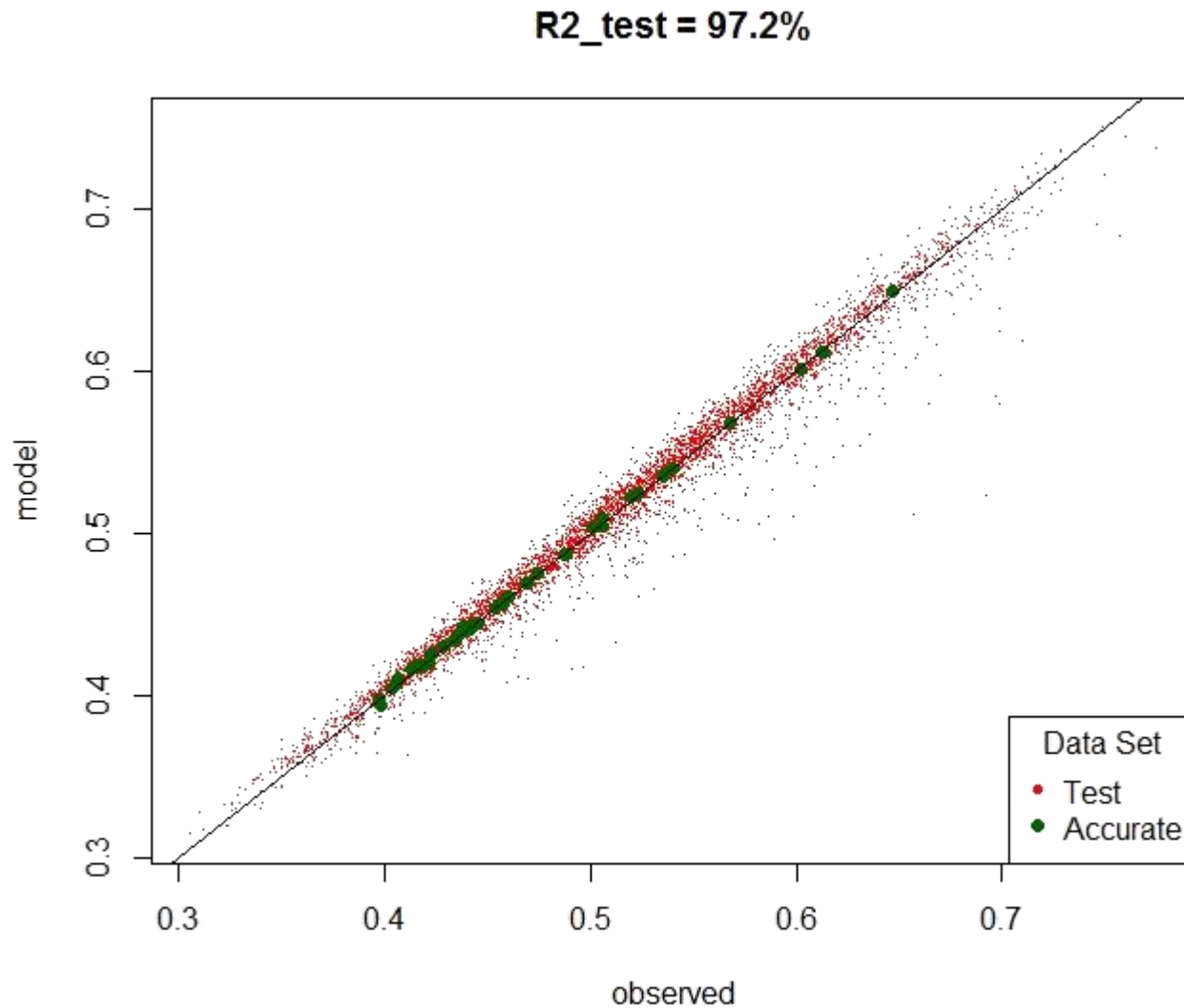
The out-of-sample R-squared is calculated by 10-fold cross validation

Network Analysis in more detail

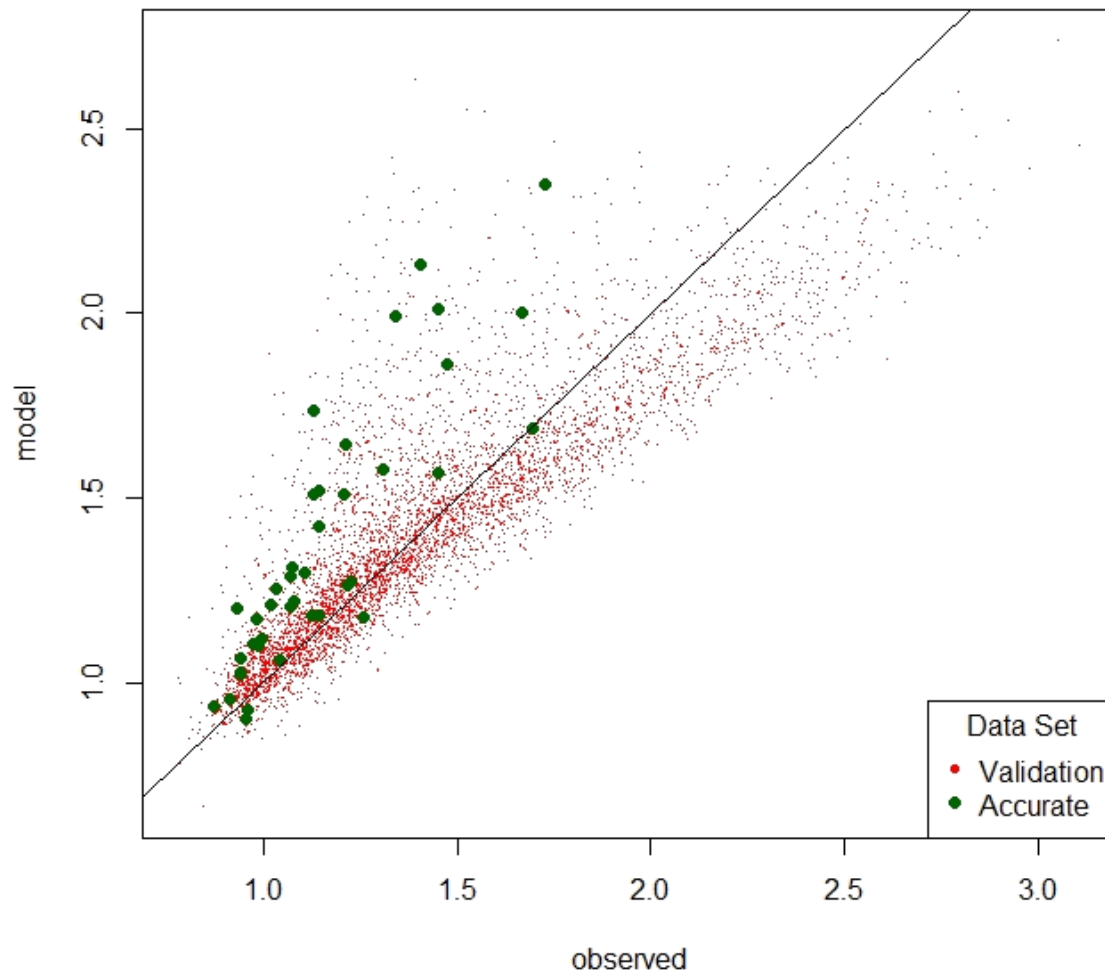
Neural Network Analysis

- » Fitting a network involves determining the network weights over a selection of hidden layer sizes and regularisation parameter values
- » 25,000 fitting scenarios are split into:
 - 15,000 training scenarios to determine the network weights
 - 5,000 validation scenarios to determine the hidden layer size and weight decay
 - 5,000 test scenarios to assess the network on new / unseen scenarios
- » We use the validation set to determine how many scenarios we need
- » Illustrate the bias / variance trade-off with hidden layer size and weight decay
- » Describe how to deal with heteroscedastic effects

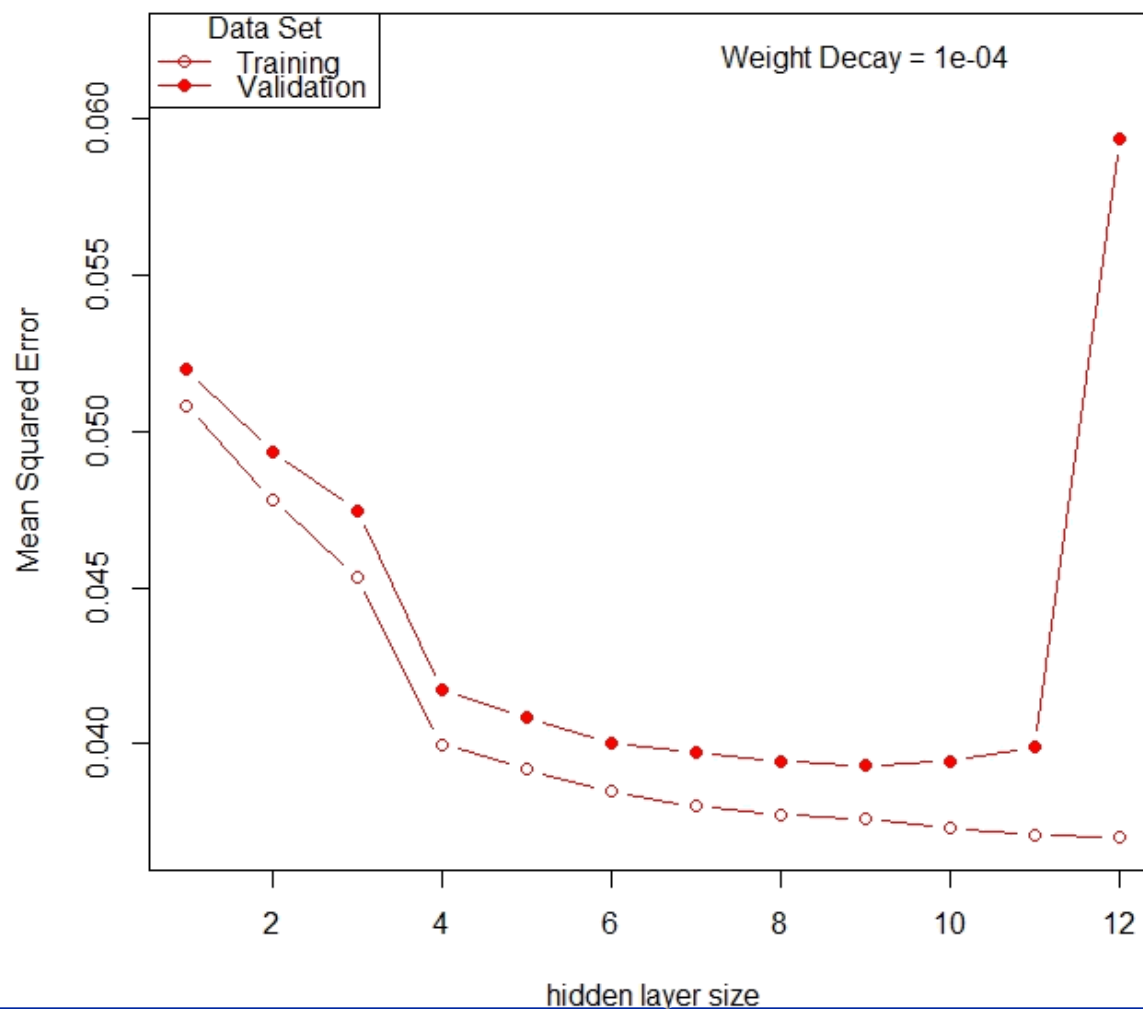
Good model output...



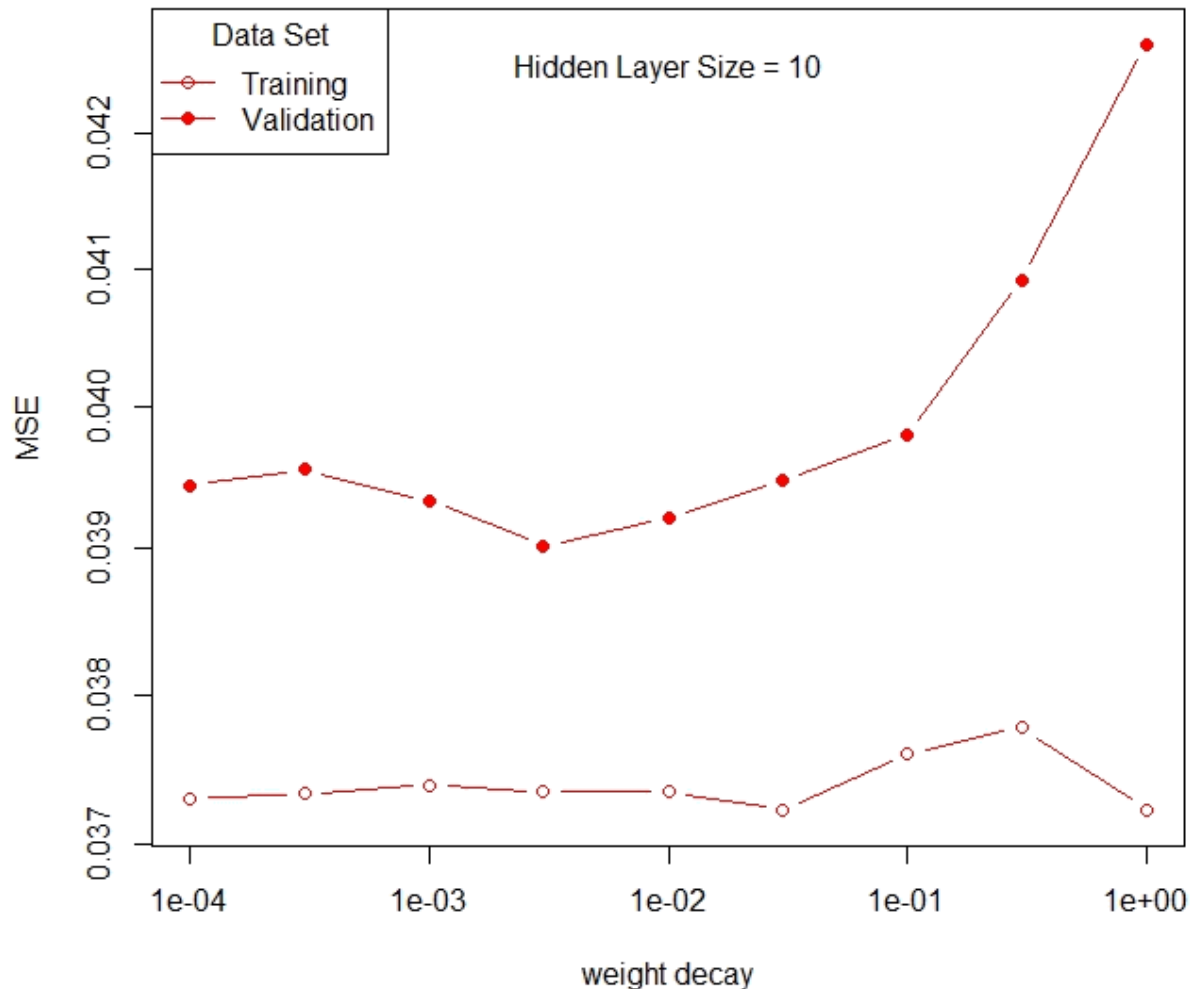
A Challenging Fit!



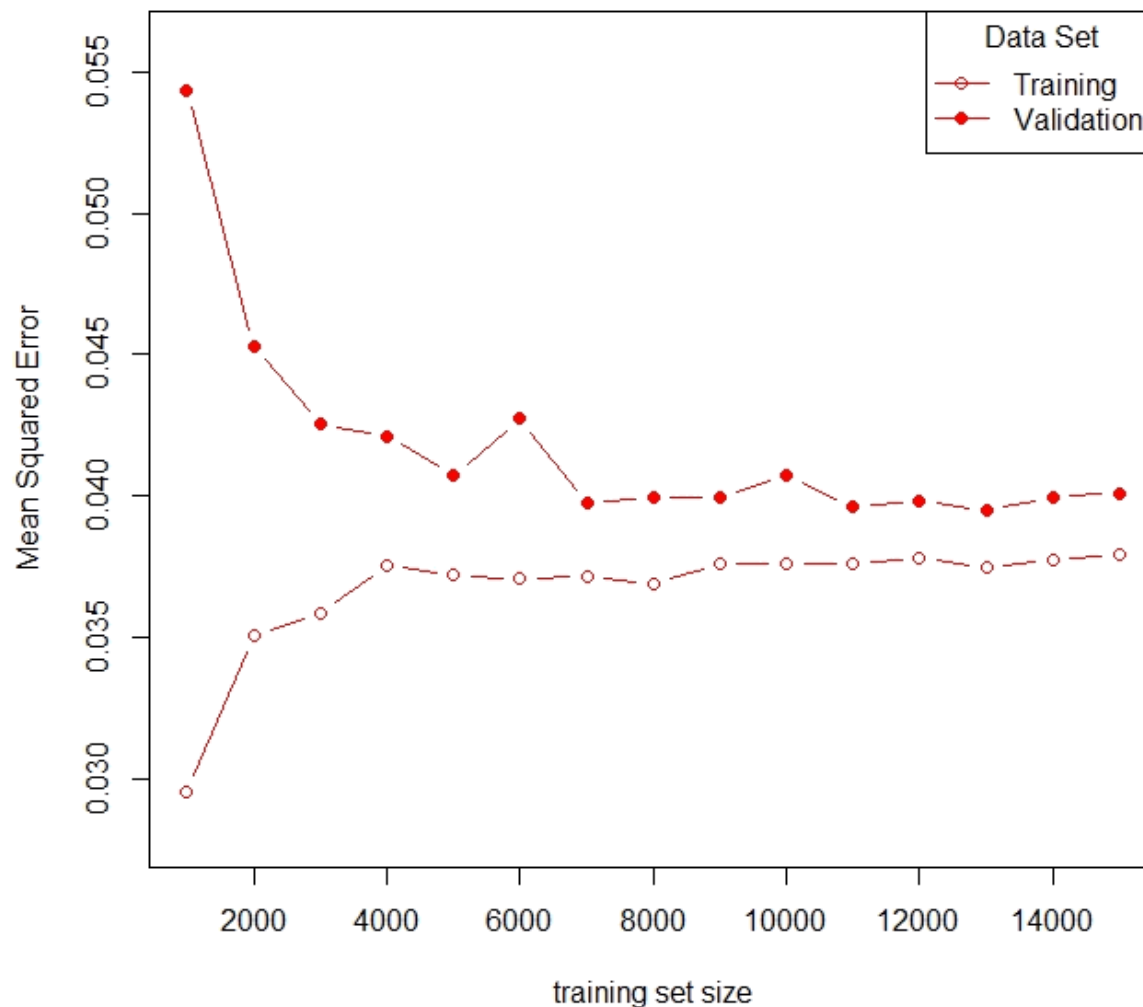
Bias-Variance Trade-off I: for fixed weight decay



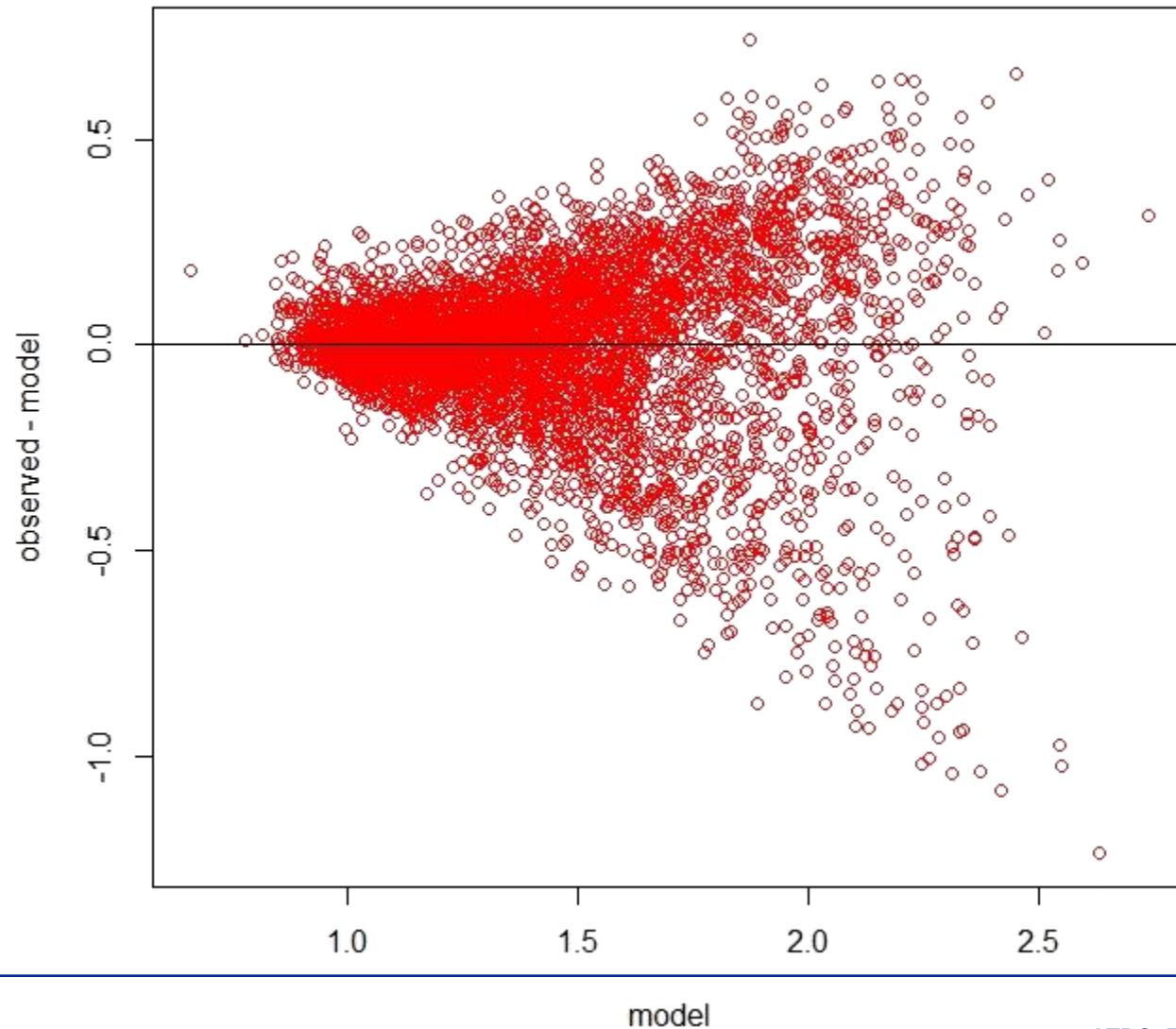
Bias-Variance Trade-off II: for fixed hidden layer size



Variation with Model Size and Fitting Scenario Budget



Heteroscedasticity



Conclusion

- » Multiple polynomial regression is a robust and practical solution to the proxy generation problem working in the majority of cases
- » Some proxy problems can be more challenging
- » Alternative methodologies exist including generalised additive models, local regression methods and artificial neural networks
- » We investigated one of these alternative approaches, neural networks, with a view to perhaps including it as an option within ProxyGenerator in the future
- » Neural networks work at least as well as multiple polynomial regression
- » Bias-variance trade-off and optimal scenario counting was discussed alongwith methods to counteract heteroscedastic effects

© 2012 Moody's Analytics, Inc. and/or its licensors and affiliates (collectively, "MOODY'S"). All rights reserved. ALL INFORMATION CONTAINED HEREIN IS PROTECTED BY COPYRIGHT LAW AND NONE OF SUCH INFORMATION MAY BE COPIED OR OTHERWISE REPRODUCED, REPACKAGED, FURTHER TRANSMITTED, TRANSFERRED, DISSEMINATED, REDISTRIBUTED OR RESOLD, OR STORED FOR SUBSEQUENT USE FOR ANY SUCH PURPOSE, IN WHOLE OR IN PART, IN ANY FORM OR MANNER OR BY ANY MEANS WHATSOEVER, BY ANY PERSON WITHOUT MOODY'S PRIOR WRITTEN CONSENT. All information contained herein is obtained by MOODY'S from sources believed by it to be accurate and reliable. Because of the possibility of human or mechanical error as well as other factors, however, all information contained herein is provided "AS IS" without warranty of any kind. Under no circumstances shall MOODY'S have any liability to any person or entity for (a) any loss or damage in whole or in part caused by, resulting from, or relating to, any error (negligent or otherwise) or other circumstance or contingency within or outside the control of MOODY'S or any of its directors, officers, employees or agents in connection with the procurement, collection, compilation, analysis, interpretation, communication, publication or delivery of any such information, or (b) any direct, indirect, special, consequential, compensatory or incidental damages whatsoever (including without limitation, lost profits), even if MOODY'S is advised in advance of the possibility of such damages, resulting from the use of or inability to use, any such information. The credit ratings, financial reporting analysis, projections, and other observations, if any, constituting part of the information contained herein are, and must be construed solely as, statements of opinion and not statements of fact or recommendations to purchase, sell or hold any securities. NO WARRANTY, EXPRESS OR IMPLIED, AS TO THE ACCURACY, TIMELINESS, COMPLETENESS, MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OF ANY SUCH RATING OR OTHER OPINION OR INFORMATION IS GIVEN OR MADE BY MOODY'S IN ANY FORM OR MANNER WHATSOEVER. Each rating or other opinion must be weighed solely as one factor in any investment decision made by or on behalf of any user of the information contained herein, and each such user must accordingly make its own study and evaluation of each security and of each issuer and guarantor of, and each provider of credit support for, each security that it may consider purchasing, holding, or selling.