Statistics for Financial Engineering: Some R Examples

David Ruppert

Cornell University

April 25, 2009

Outline

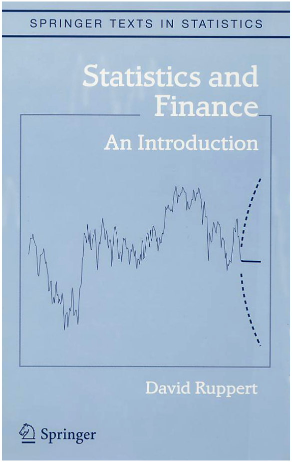
A little about myself

- BA and MA in mathematics
- PhD in statistics in 1977
- taught in the statistics department at North Carolina for 10 years
- have been in Operations Research and Information (formerly Industrial) Engineering at Cornell since 1987

A little about myself

- starting teaching Statistics and Finance to undergraduates in 2001
  - textbook published in 2004
- starting teaching Statistics for Financial Engineering to master's students in 2008
  - working on revised and expanded textbook
- now programming exclusively in R

Undergraduate Textbook



A little about my research

- have done research in
  - asymptotic theory of splines
  - semiparametric modeling
  - measurement error in regression
  - smoothing (nonparametric regression and density estimation)
  - transformation and weighting
  - stochastic approximation
  - biostatistics
  - environmental engineering
  - modeling of term structure
  - executive compensation and accounting fraud

Three types of regression

Linear regression

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p} + \epsilon_i, \;\; i = 1, \ldots, n$$

Nonlinear regression

$$Y_i = m(X_{i,1}, \ldots, X_{i,p}; \beta_1, \ldots, \beta_q) + \epsilon_i, \;\; i = 1, \ldots, n$$

where m is a known function depending on unknown parameters

Nonparametric regression

$$Y_i = m(X_{i,1}, \ldots, X_{i,p}) + \epsilon_i, \;\; i = 1, \ldots, n$$

where m is an unknown "smooth" function

Usual assumptions on the noise

Usually $\epsilon_1, \ldots, \epsilon_n$ are assumed to be:

- mutually independent (or at least uncorrelated)
- homoscedastic (constant variance)
- normally distributed

Much research over the last 50+ years has looked into ways of

1. checking these assumptions
2. statistical methods that require less assumptions

Transform-both-sides model

Ideal model (no errors):

$$Y_i = f(\boldsymbol{X}_i, \boldsymbol{\beta})$$

Statistical model (first attempt):

$$Y_i = f(\boldsymbol{X}_i, \boldsymbol{\beta}) + \epsilon_i$$

where $\epsilon_1, \ldots, \epsilon_n$ are iid Gaussian

TBS model:

$$h\{Y_i\} = h\{f(\boldsymbol{X}_i, \boldsymbol{\beta})\} + \epsilon_i$$

where

- $\epsilon_1, \ldots, \epsilon_n$ are iid Gaussian
- h is an "appropriate" transformation

Estimation of Default Probabilities

Data:

- ratings: 1=Aaa (best),...,16=B3 (worse)
- default frequency (estimate of default probability)

Some statistical models

- **nonlinear model:**

$$\text{Pr}(\text{default}|\text{rating}) = \exp\{\beta_0 + \beta_1 \text{rating}\}$$

- **linear/transformation model (in recent textbook):**

$$\log\{\text{Pr}(\text{default}|\text{rating})\} = \beta_0 + \beta_1 \text{rating}$$

  - **Problem:** cannot take logs of default frequencies that are 0
  - **(Sub-optimal) solution in textbook:** throw out these observations

A better statistical model

- **Transform-both-sides (TBS) model** – see Carroll and Ruppert (1984, 1988):
- using a power transformation:

$$\{\Pr(\text{default}|\text{rating}) + \kappa\}^{\lambda} = \{\exp(\beta_0 + \beta_1 \text{rating}) + \kappa\}^{\lambda}$$

  - λ chosen by residual plots (or maximum likelihood)
  - $\lambda = 1/2$ works well for this example
  - log transformations are also commonly used
  - $\kappa > 0$ will shift data away from 0

The Box-Cox family

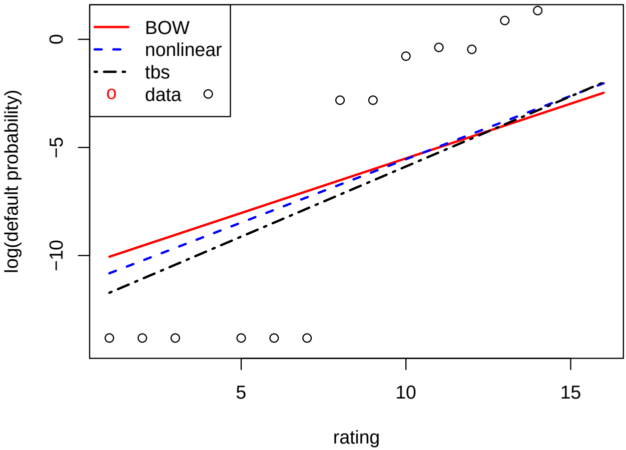
- the most common transformation family is due to Box and Cox (1964):

$$
\begin{aligned}
h(y, \lambda) &= \frac{y^\lambda - 1}{\lambda} \text{ if } \lambda \neq 0 \\
&= \log(y) \text{ if } \lambda = 0
\end{aligned}
$$

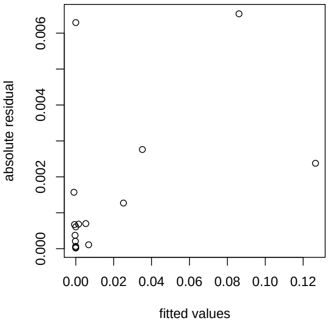
- derivative has simple form:

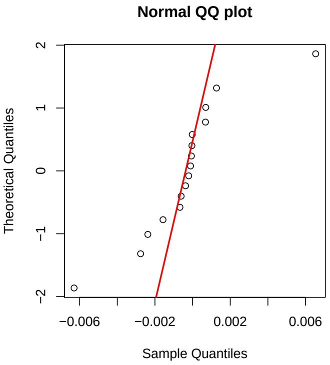
$$
h_y(y, \lambda) = \frac{d}{dy} h(y, \lambda) = y^{\lambda - 1} \text{ for all } \lambda
$$

TBS fit compared to others

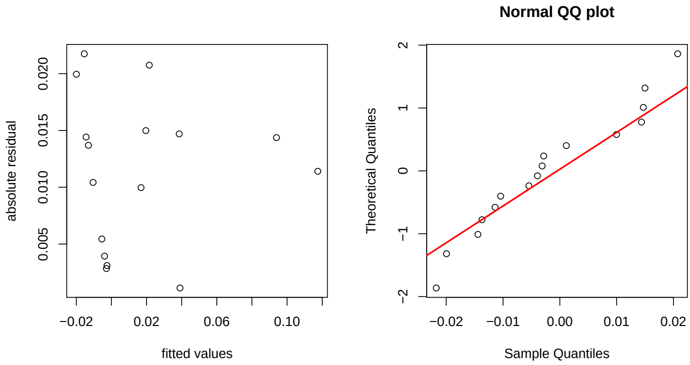


Nonlinear regression residuals





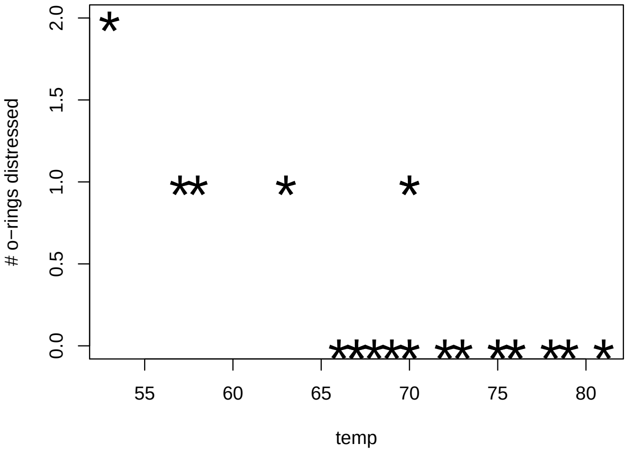
TBS residuals



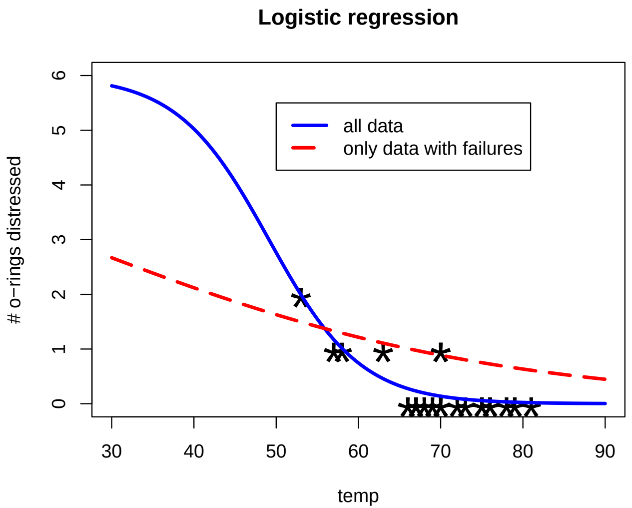
Estimated default probabilities

method	$\widehat{Pr}\{\text{default}|\text{Aaa}\}$	as % of TEXTBOOK est
TEXTBOOK	0.005%	100%
nonlinear	0.002%	40%
TBS	0.0008%	16%

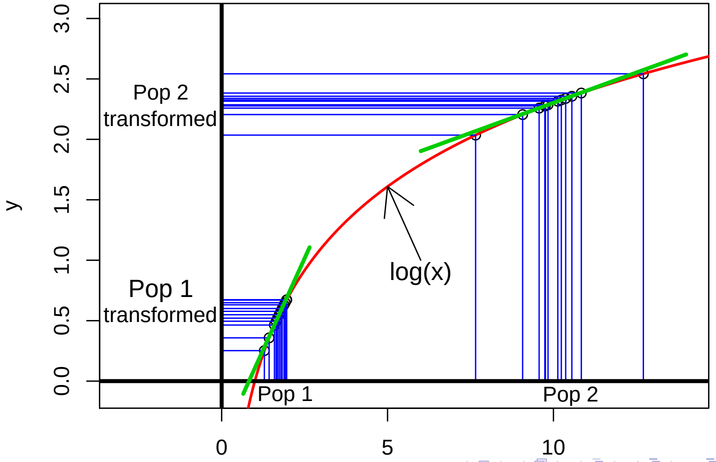
A Similar Problem: Challenger Data



Challenger Data: Extrapolation to 31°



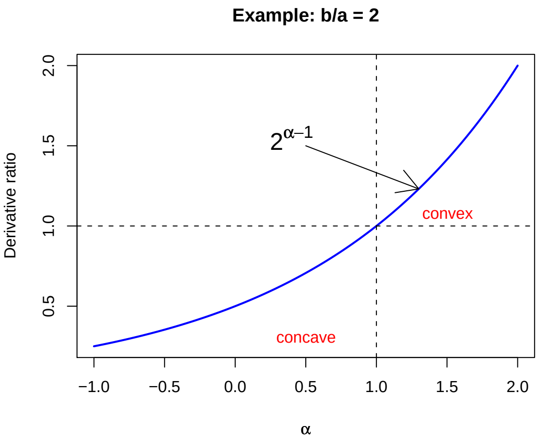
Variance stabilizing transformation: how it works



Strength of Box-Cox family

- Take $a < b$
- Then
$$\frac{h_y(b,\lambda)}{h_y(a,\lambda)} = \left(\frac{b}{a}\right)^{\lambda-1}$$
which is increasing in λ and equals 1 when $\lambda = 1$
- $\lambda = 1$ is the dividing point between concave and convex transformations
- $h(y,\lambda)$ becomes a stronger concave transformation as λ decreases from 1
- also, $h(y,\lambda)$ becomes a stronger convex transformation as λ increases from 1

Strength of Box-Cox family, cont.

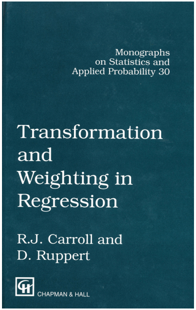


Maximum likelihood

$$\mathcal{L}(\boldsymbol{\beta}, \lambda, \sigma) = n\log(\sigma) - \sum_{i=1}^{n} \frac{\left[h(Y_i+\kappa,\lambda) - h\{f(\boldsymbol{X}_i,\boldsymbol{\beta})+\kappa,\lambda\}\right]^2}{2\sigma^2} + \underbrace{\sum_{i=1}^{n}(\lambda-1)\log(Y_i)}_{\text{from Jacobian}}$$

- can maximize over σ analytically:
  - $\widehat{\sigma}^2 = n^{-1}\sum_{i=1}^{n}\left[h(Y_i+\kappa,\lambda) - h\{f(\boldsymbol{X}_i,\boldsymbol{\beta})+\kappa,\lambda\}\right]^2$
- they maximize over $(\boldsymbol{\beta}, \lambda)$ with `optim`, for example
- κ is fixed in advance

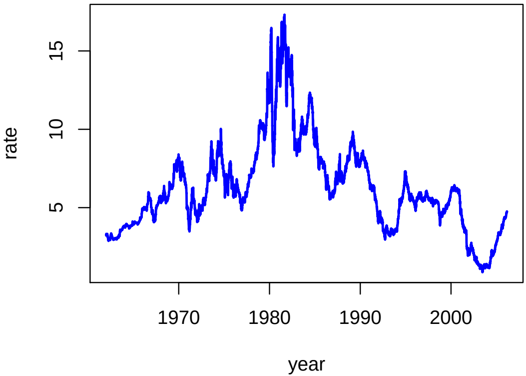
Reference for TBS



Transformation and Weighting in Regression by Carroll and Ruppert (1988)

- Lots of examples
- But none in finance ☹

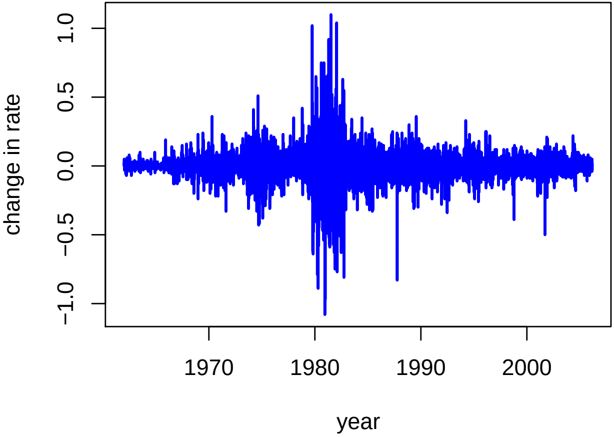
1-Year Treasury Constant Maturity Rate, daily data



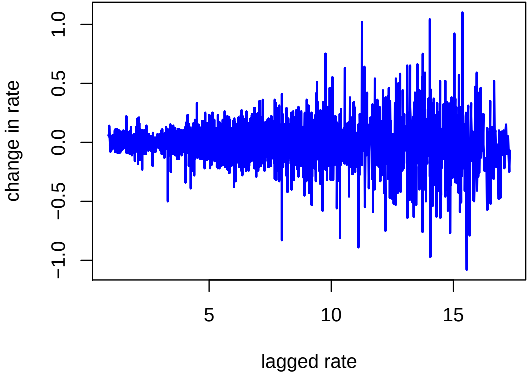
Source: Board of Governors of the Federal Reserve System

http://research.stlouisfed.org/fred2/

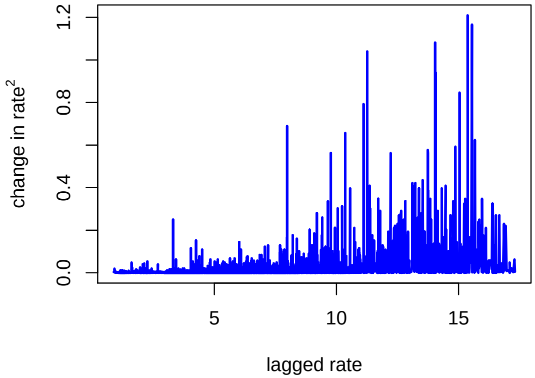
ΔR_t versus year



ΔR_t versus R_{t-1}



ΔR_t^2 versus R_{t-1}



Drift function

Discretized diffusion model:

$$\Delta R_t = \mu(R_{t-1}) + \sigma(R_{t-1})\epsilon_t$$

- $\mu(x)$ is the drift function
- $\sigma(x)$ is the volatility function (as before)

Estimating Volatility

Parametric model:

$$\mathrm{Var}\{(\Delta R_t)\} = \beta_0 R_{t-1}^{\beta_1}$$

(Common in practice)

Nonparametric model:

$$\mathrm{Var}\{(\Delta R_t)\} = \sigma^2(R_{t-1})$$

where $\sigma(\cdot)$ is a smooth function

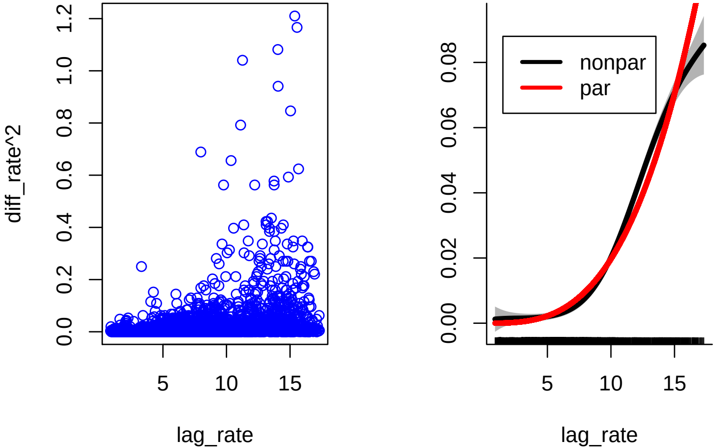
- will be modeled as a spline
- In these models: no dependence on t

Spline Software

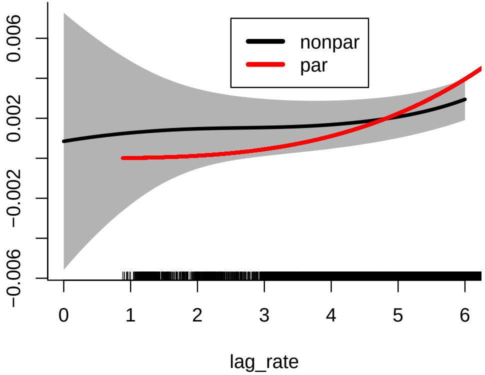
The penalized spline fits shown here were obtained using the function `spm`

- in R's `SemiPar` package
- author is Matt Wand

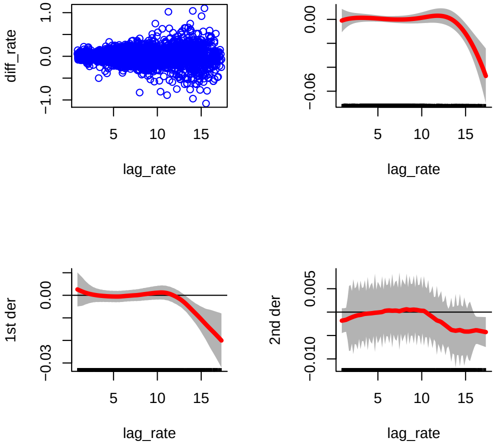
Comparing parametric and nonparametric volatility fits



Comparing parametric and nonparametric volatility fits: zooming in near 0



Spline fitting – Estimation of drift function



Residuals for diffusion model

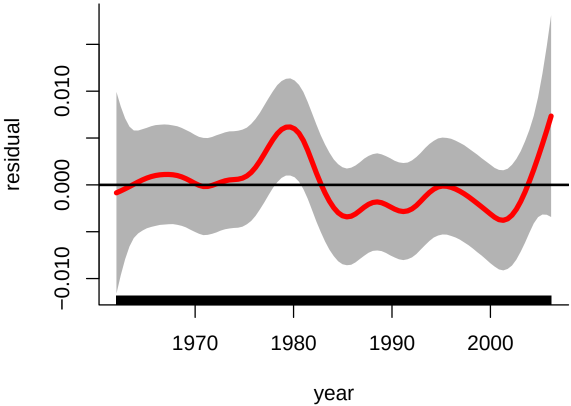
$$\begin{aligned} \text{residual}_t &:= \Delta R_t - \widehat{\mu}(R_{t-1}) \\ E(\text{residual}_t) &= 0 \end{aligned}$$

$$\begin{aligned} \text{std residual}_t &:= \frac{\text{residual}_t}{\widehat{\sigma}(R_{t-1})} \\ E(\text{std residual}_t^2) &= 1 \end{aligned}$$

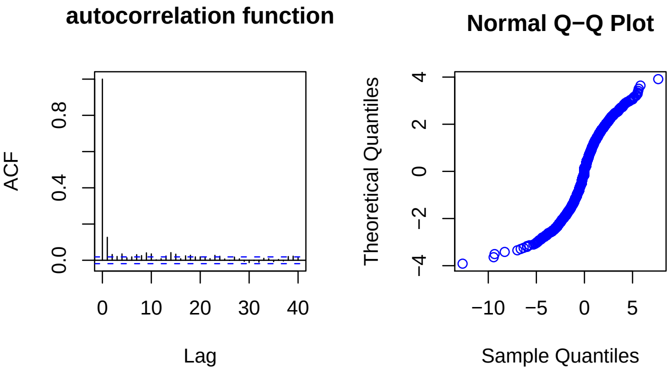
Question

Are the drift and volatility functions constant in time?

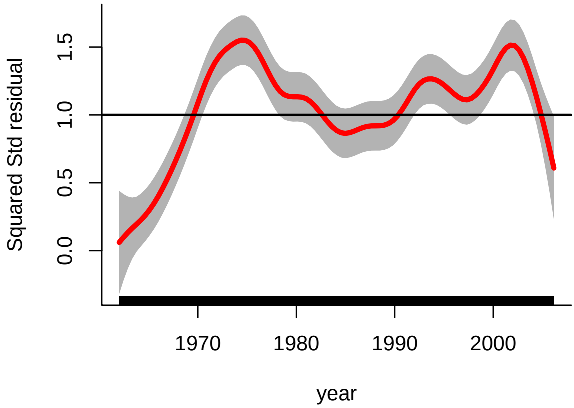
Residual plots: ordinary residuals



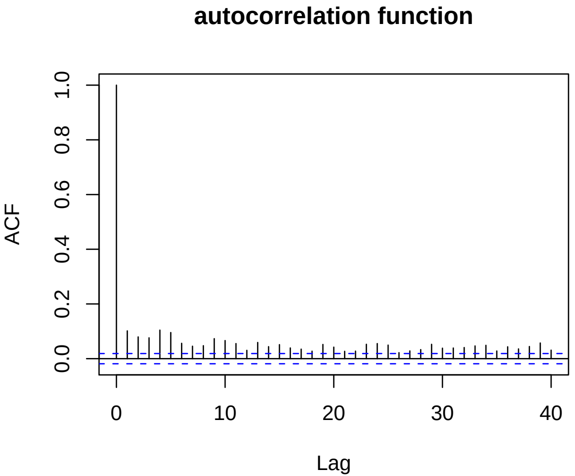
Residual plots: standardized residuals



Residual plots: Squared standardized residuals



Residual plots: Squared standardized residuals



GARCH(p, q) model

The GARCH(p, q) model is

$$a_t = \epsilon_t \sigma_t,$$

where

$$\sigma_t = \sqrt{\alpha_0 + \sum_{i=1}^{q} \alpha_i a_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2}.$$

and

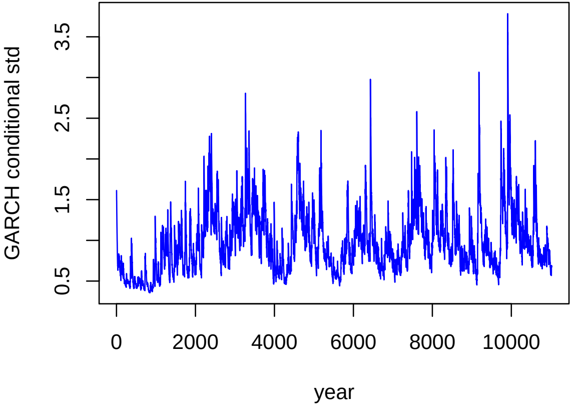
$$\epsilon_t \text{ is an iid (strong) white noise process}$$

- a_t is weak white noise
- uncorrelated but with volatility clustering

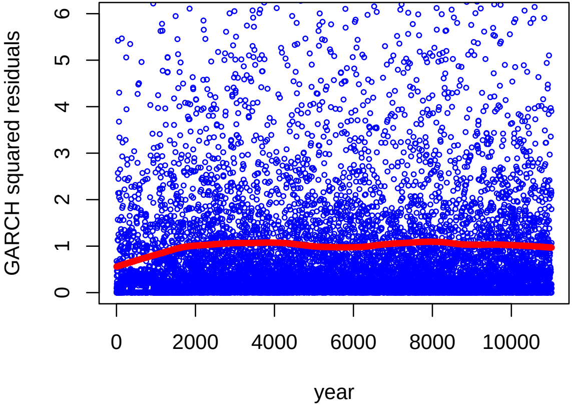
GARCH(1,1) fit using `garch` in `tseries`

```
Call:
garch(x = std_drift_resid^2, order = c(1, 1))

Model:
GARCH(1,1)

Coefficient(s):
    Estimate  Std. Error  t value Pr(>|t|)
a0   0.27291     0.00148      184   <2e-16 ***
a1   0.44690     0.00252      177   <2e-16 ***
b1   0.80490     0.00075     1073   <2e-16 ***

        Box-Ljung test

data:  Squared.Residuals
X-squared = 0.13, df = 1, p-value = 0.7186
```

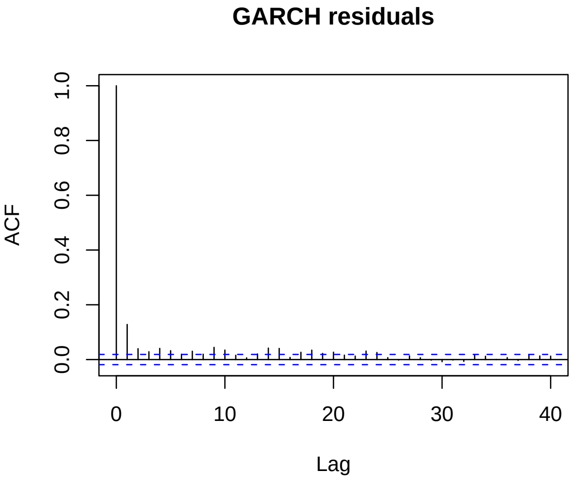
GARCH: estimated conditional standard deviations



GARCH: squared residuals with lowess smooth



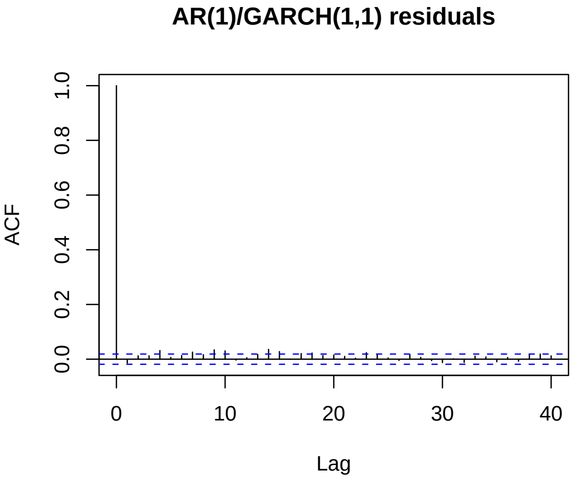
GARCH residuals



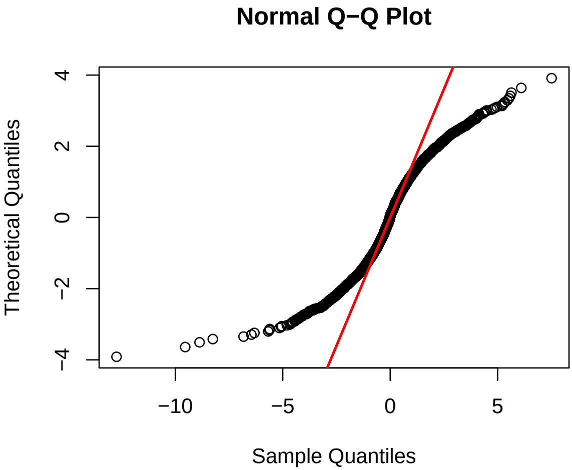
AR(1)/GARCH(1,1)

```
Call:
 garchFit(formula = ~arma(1, 0) + garch(1, 1), data = std_drift_resid)

Mean and Variance Equation:
 data ~ arma(1, 0) + garch(1, 1)
 [data = std_drift_resid]

Conditional Distribution:
 norm

Std. Errors:
 based on Hessian

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.001099    0.007476    0.147    0.883
ar1     0.138691    0.010468   13.248  < 2e-16 ***
omega   0.008443    0.001163    7.257 3.96e-13 ***
alpha1  0.073603    0.005483   13.424  < 2e-16 ***
beta1   0.923098    0.005457  169.158  < 2e-16 ***
```

AR(1)/GARCH(1,1) residuals



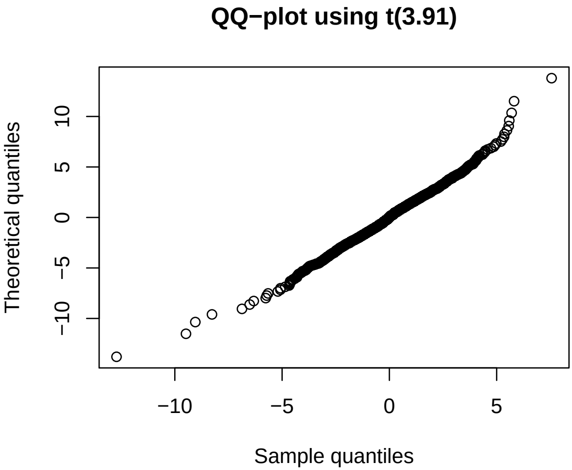
AR(1)/GARCH(1,1) residuals - QQ plot



AR(1)/GARCH(1,1)

```
Call:
 garchFit(formula = ~arma(1, 0) + garch(1, 1), data = std_drift_resid,
    cond.dist = "std")

Mean and Variance Equation:
 data ~ arma(1, 0) + garch(1, 1)
 [data = std_drift_resid]

Conditional Distribution:
 std

Error Analysis:
        Estimate  Std. Error  t value Pr(>|t|)
mu     0.0001087   0.0059401    0.018  0.98540
ar1    0.0969621   0.0090996   10.656  < 2e-16 ***
omega  0.0016722   0.0005955    2.808  0.00498 **
alpha1 0.0664390   0.0065895   10.083  < 2e-16 ***
beta1  0.9413495   0.0052248  180.169  < 2e-16 ***
shape  3.9169920   0.1600835   24.468  < 2e-16 ***
```

AR(1)/GARCH(1,1) residuals - QQ plot

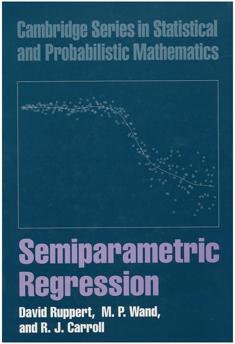


Final model for the interest rate dynamics

$$\Delta R_t = \mu(R_{t-1}) + \sigma(R_{t-1})a_t$$

1. Model was fit in two steps:
   1. estimate $\mu()$ and $\sigma()$
      - spm in SemiPar
   2. model a_t as AR(1)/GARCH(1,1)
      - garchFit in fGarch
2. Could the two step be combined?
3. Would combining them change the results?

Reference for spline modeling



Semiparametric Regression by Ruppert, Wand, and Carroll (2003)

- Lots of examples.
- But most from biostatistics and epidemiology

Bayesian statistics

- Bayesian analysis allows the use of prior information
- hierarchical priors can:
  - specify knowledge that a group of parameters are similar to each other
  - estimate their common distribution
- WinBUGS can be run from inside R using the `R2WinBUGS` package
- there is a similar `BRugs` package that runs `OpenBugs`
  - BRugs is no longer on CRAN

Data

midcapD.ts in fEcofin package

- 500 daily returns on:
  - 20 stocks
  - market

Goal

The goal is to use the first 100 days to estimate the mean returns for the next 400 days

Four possible estimators:

- sample means
- Bayes estimation (shrinkage)
- mean of means (total shrinkage)
- CAPM
  - (expected return) = beta $\times$ (expected market return)

Who won?

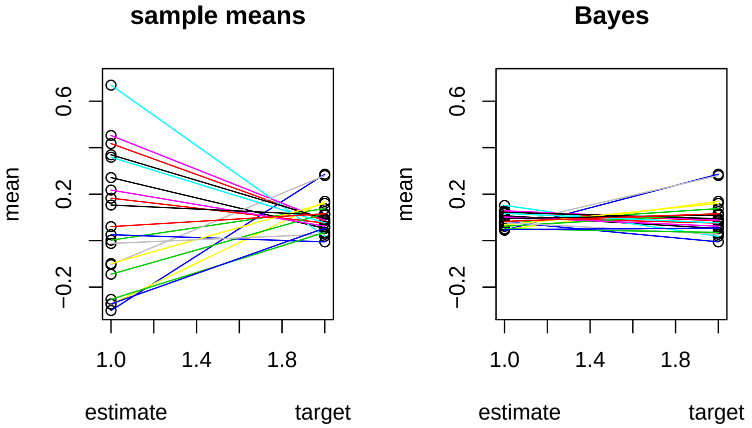
Estimate	Sum of squared errors
sample means	1.9
Bayes	0.17
mean of means	0.12
CAPM 1	0.66
CAPM 2	0.45

Squared estimation errors are summed over the 20 stocks

CAPM 1: use mean of first 100 market returns

CAPM 2: use mean of last 400 market returns

Why does shrinkage help?



Likelihood and prior

$r_{i,t}$ = tth return on i stock

Likelihood:

$$r_{i,t} = \mu_i + \epsilon_{i,t}$$

$$\epsilon_{i,t} \sim IN(0, \sigma_\epsilon^2)$$

IN = "Independent Normal"

Hierarchical Prior:

$$\mu_i \sim IN(\alpha, \sigma_\mu^2)$$

Diffuse (non-informative) priors on α, σ_ϵ^2, σ_μ^2

Auto and cross-sectional correlations are ignored (treated as 0)

Data-driven shrinkage

Hierarchical Prior:

$$\mu_i \sim IN(\alpha, \sigma_\mu^2)$$

- the μ_i are shrunk towards α
- α should be (approximately) the mean of the means
- $\sigma_\mu^2/\sigma_\epsilon^2$ controls the amount of shrinkage
  - large $\sigma_\mu^2/\sigma_\epsilon^2 \Rightarrow$ less shrinkage
- data-driven shrinkage
  - because σ_μ^2 and σ_ϵ^2 are estimated

WinBUGS output

```
> print(means.sim,digits=3)
Inference for Bugs model at "midCap.bug", fit using WinBUGS,
 3 chains, each with 5100 iterations (first 100 discarded)
 n.sims = 15000 iterations saved
            mean    sd     2.5%      25%     50%     75%   97.5% Rhat n.eff
mu[1]     1.1e-01 0.169    -0.22 -1.0e-03 1.1e-01 2.1e-01 4.5e-01    1  4000
mu[2]     1.2e-01 0.170    -0.20  1.5e-02 1.2e-01 2.3e-01 4.7e-01    1  6500
mu[3]     7.7e-02 0.168    -0.27 -2.7e-02 7.9e-02 1.8e-01 4.1e-01    1  3300
mu[4]     4.5e-02 0.176    -0.33 -6.1e-02 5.2e-02 1.6e-01 3.8e-01    1  1300

mu[18]    8.3e-02 0.170    -0.27 -2.4e-02 8.7e-02 1.9e-01 4.1e-01    1  3000
mu[19]    5.1e-02 0.171    -0.32 -5.1e-02 5.7e-02 1.6e-01 3.7e-01    1  1700
mu[20]    4.8e-02 0.175    -0.33 -5.8e-02 5.5e-02 1.6e-01 3.7e-01    1  1800
sigma_mu  1.5e-01 0.065     0.06  9.9e-02 1.3e-01 1.8e-01 3.1e-01    1   520
sigma_eps 4.3e+00 0.068     4.18  4.3e+00 4.3e+00 4.4e+00 4.4e+00    1 15000
alpha     8.8e-02 0.102    -0.11  1.7e-02 8.8e-02 1.6e-01 2.8e-01    1   710
deviance  1.2e+04 3.989 11510.00  1.2e+04 1.2e+04 1.2e+04 1.2e+04    1  5300

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

DIC info (using the rule, pD = Dbar-Dhat)
pD = 4.1 and DIC = 11521.6
DIC is an estimate of expected predictive error (lower deviance is better).
```