Does nonlinearity exist in financial TS?
Yes, especially in volatility & high-freq data

We focus on simple nonlinear models & neural networks

What is a linear time series?

\[ x_t = \mu + \sum_{i=0}^{\infty} \psi_i a_{t-i} \]

where \( \mu \) is a constant, \( \psi_i \) are real numbers with \( \psi_0 = 1 \), and \( \{a_t\} \) is an iid \( (0, \sigma^2_a) \).

**General concept:** Let \( F_{t-1} \) denote the information available at time \( t - 1 \).
Conditional mean:

\[ \mu_t = E(x_t|F_{t-1}) \equiv g(F_{t-1}) , \]

Conditional variance:

\[ \sigma_t^2 = \text{Var}(x_t|F_{t-1}) \equiv h(F_{t-1}) \]

where \( g(.) \) and \( h(.) \) are well-defined functions with \( h(.) > 0 \).
For a linear series, \( g(.) \) is a linear function of \( F_{t-1} \) and \( h(.) = \sigma^2_a \).

Statistics literature: focuses on \( g(.) \)
See the book by Tong (Oxford University Press, 1990)
Financial econometrics literature: focuses on \( h(.) \)

**Some specific models**

**TAR model:** a piecewise linear model in the space of a threshold variable.
Example: 2-regime AR(1) model

\[ x_t = \begin{cases} 
-1.5x_{t-1} + a_t & \text{if } x_{t-1} < 0, \\
0.5x_{t-1} + a_t & \text{if } x_{t-1} \geq 0,
\end{cases} \]

where \( a_t \)'s are iid \( N(0, 1) \).

Here the delay is 1 time period, \( x_{t-1} \) is the threshold variable, and the threshold is 0. The threshold divides the range (or space) of \( x_{t-1} \) into two regimes with Regime 1 denoting \( x_{t-1} < 0 \).

What is so special about this model? See the time plot.

Special features of the model: (a) asymmetry in rising and declining patterns, (more data points are positive than negative) (b) the mean of \( x_t \) is not zero even though there is no constant term in the model, (c) the lag-1 coefficient may be greater than 1 in absolute value.

Financial applications:


\[ r_t = \alpha + \beta r_{m,t} + \epsilon_t. \]
A simple nonlinear model:

\[
    r_t = \begin{cases} 
    \alpha_1 + \beta_1 r_{m,t} + \epsilon_t, & \text{if } r_{m,t} \leq 0 \\
    \alpha_2 + \beta_2 r_{m,t} + \epsilon_t, & \text{if } r_{m,t} > 0.
    \end{cases}
\]

```r
> da=read.table("m-gmsp6708.txt",header=T)
> head(da)
   Date    GM    SP
1 19670331 0.053541 0.039410
6 19670831 -0.004720 -0.011715
> gm=log(da$GM+1)
> sp=log(da$SP+1)
> m1=lm(gm~sp)  # Market model
> summary(m1)

Call: lm(formula = gm ~ sp)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0049  0.003434  -1.415 0.158
sp          1.0725  0.077177  13.897 <2e-16 ***
---
Residual standard error: 0.07652 on 500 degrees of freedom
Multiple R-squared: 0.2786, Adjusted R-squared: 0.2772

> length(gm)
[1] 502
> idx=c(1:502)[sp <= 0]  # Locate all non-positive market returns
> nsp=rep(0,502)  # Create the variable of non-positive market returns
> nsp[idx]=sp[idx]
> c1=rep(0,502)  # Create a variable for intercept of non-positive market returns.
> c1[idx]=1
> xx=cbind(gm,sp,c1,nsp)  # Show the resulting variables
> head(xx)
   gm sp  c1 nsp
[1,] 0.052156871 0.03865324 0 0.0000000
[2,] 0.126126796 0.04137128 0 0.0000000
[3,] -0.083130553 -0.05386607 1 -0.05386607
[4,] -0.024098039 0.01736043 0 0.0000000
[5,] 0.097524998 0.04434602 0 0.0000000
[6,] -0.004731174 -0.01178416 1 -0.01178416

> m2=lm(gm~c1+sp)  # with different intercepts
> summary(m2)

Call: lm(formula = gm ~ c1 + sp)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0049  0.003434  -1.415 0.158
sp          1.0725  0.077177  13.897 <2e-16 ***
---
Residual standard error: 0.07652 on 500 degrees of freedom
Multiple R-squared: 0.2786, Adjusted R-squared: 0.2772
```
Coefficients:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|---------|
| (Intercept) | -0.014971 | 0.005931 | -2.524 | 0.0119 * |
| c1       | 0.021994  | 0.010538 | 2.087  | 0.0374 * |
| sp       | 1.258037  | 0.117556 | 10.702 | <2e-16 *** |

Residual standard error: 0.07626 on 499 degrees of freedom
Multiple R-squared: 0.2849, Adjusted R-squared: 0.282

> m3=lm(gm~sp+nsp)
> summary(m3)

Call: lm(formula = gm ~ sp + nsp)

Coefficients:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|---------|
| (Intercept) | 0.002329 | 0.005288 | 0.440  | 0.6598 |
| sp       | 0.848133  | 0.147421 | 5.753  | 1.53e-08 *** |
| nsp      | 0.421989  | 0.236424 | 1.785  | 0.0749 . |

Residual standard error: 0.07635 on 499 degrees of freedom
Multiple R-squared: 0.2832, Adjusted R-squared: 0.2803

> m4=lm(gm~sp+c1+nsp)
> summary(m4)

Call: lm(formula = gm ~ sp + c1 + nsp)

Coefficients:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|---------|
| (Intercept) | -0.007778 | 0.007369 | -1.055 | 0.2917 |
| sp       | 1.041129  | 0.176838 | 5.887  | 7.21e-09 *** |
| c1       | 0.020713  | 0.010550 | 1.963  | 0.0502 . |
| nsp      | 0.387630  | 0.236399 | 1.640  | 0.1017 |

Residual standard error: 0.07613 on 498 degrees of freedom
Multiple R-squared: 0.2887, Adjusted R-squared: 0.2844

(B) Modeling the leverage effect in volatility: Recall EGARCH, GJR, TGARCH, and APARCH models.

Markov switching models
Two-state MS model:

\[ x_t = \begin{cases} 
  c_1 + \sum_{i=1}^{p} \phi_{1,i} x_{t-i} + a_{1t} & \text{if } s_t = 1, \\
  c_2 + \sum_{i=1}^{p} \phi_{2,i} x_{t-i} + a_{2t} & \text{if } s_t = 2, 
\end{cases} \]

where \( s_t \) assumes values in \( \{1,2\} \) and is a first-order Markov chain with trans. prob.

\[ P(s_t = 2|s_{t-1} = 1) = w_1, \quad P(s_t = 1|s_{t-1} = 2) = w_2, \]

where \( 0 \leq w_1 \leq 1 \) is the probability of switching out State 1 from time \( t-1 \) to time \( t \). A large \( w_1 \) means that it is easy to switch out State 1, i.e. cannot stay in State 1 for long. The inverse, \( 1/w_1 \), is the expected duration (number of time periods) to stay in State 1. Similar idea applies to \( w_2 \).

**Example:** Growth rate of US quarterly real GNP 47-91. See Figure 4.4 of the textbook (p.188).

<table>
<thead>
<tr>
<th>State 1</th>
<th>Par</th>
<th>( c_i )</th>
<th>( \phi_1 )</th>
<th>( \phi_2 )</th>
<th>( \phi_3 )</th>
<th>( \phi_4 )</th>
<th>( \sigma_i )</th>
<th>( w_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Est</td>
<td>0.909</td>
<td>0.265</td>
<td>0.029</td>
<td>-0.126</td>
<td>-0.110</td>
<td></td>
<td>0.816</td>
<td>0.118</td>
</tr>
<tr>
<td>S.E</td>
<td>0.202</td>
<td>0.113</td>
<td>0.126</td>
<td>0.103</td>
<td>0.109</td>
<td></td>
<td>0.125</td>
<td>0.053</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State 2</th>
<th>Par</th>
<th>( c_i )</th>
<th>( \phi_1 )</th>
<th>( \phi_2 )</th>
<th>( \phi_3 )</th>
<th>( \phi_4 )</th>
<th>( \sigma_i )</th>
<th>( w_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Est</td>
<td>-0.420</td>
<td>0.216</td>
<td>0.628</td>
<td>-0.073</td>
<td>-0.097</td>
<td></td>
<td>1.017</td>
<td>0.286</td>
</tr>
<tr>
<td>S.E</td>
<td>0.324</td>
<td>0.347</td>
<td>0.377</td>
<td>0.364</td>
<td>0.404</td>
<td></td>
<td>0.293</td>
<td>0.064</td>
</tr>
</tbody>
</table>

**Discussion**

- Regime 2, which has a negative expectation (or growth), denotes “recession” periods. The S.E. of the estimates are large due to the small number of data in the regime.
• The expected durations for Regime 1 and 2 are 8.5 and 3.5 quarters, respectively. \((1/w_i)\)

**Discussion**: Threshold model vs Markov switching model. Deterministic switching vs stochastic switching. They are basically trying to handle similar nonlinearity in a time series.

**Neural networks**

• a semi-parametric approach to data analysis

• Structure of a network
  – Output layer
  – Input layer
  – Hidden layer
  – Nodes

• Activation function:
  – Logistic function:

\[
\ell(z) = \frac{\exp(z)}{1 + \exp(z)}
\]
– Heaviside (or threshold) function:

\[ H(z) = \begin{cases} 
1 & \text{if } z > 0 \\
0 & \text{if } z \leq 0 
\end{cases} \]

• Use \( \ell(z) \) for the hidden layer

Feed-forward neural network:
Hidden node:
\[ x_j = f_j(\alpha_j + \sum_{i \rightarrow j} w_{ij}x_i) \]
where \( f_j(.) \) is an activation function which is typically taken to be the logistic function
\[ f_j(z) = \frac{\exp(z)}{1 + \exp(z)}, \]
\( \alpha_j \) is called the bias, the summation \( i \rightarrow j \) means summing over all input nodes feeding to \( j \), and \( w_{ij} \) are the weights.
Output node:
\[ y = f_o(\alpha_o + \sum_{j \rightarrow o} w_{jo}x_j), \]
where the activation function \( f_o(.) \) is either linear or a Heaviside function. By a Heaviside function, we mean \( f_o(z) = 1 \) if \( z > 0 \) and \( f_o(z) = 0 \), otherwise.
General form:
\[ y = f_o \left[ \alpha_o + \sum_{j \rightarrow o} w_{jo}f_j \left( \alpha_j + \sum_{i \rightarrow j} w_{ij}x_i \right) \right]. \]
With direct connections from the input layer to the output layer:
\[ y = f_o \left[ \alpha_o + \sum_{i \rightarrow o} w_{io}x_i + \sum_{j \rightarrow o} w_{jo}f_j \left( \alpha_j + \sum_{i \rightarrow j} w_{ij}x_i \right) \right]. \]
Training and forecasting
Divide the data into training and forecasting subsamples.

**Training**: build a few network systems

**Forecasting**: based on the accuracy of out-of-sample forecasts to select the “best” network.

**Example**: Monthly log returns of IBM stock 26-99. See text for details.

**Some R commands**: with `nnet` package

```r
library(nnet)
x=scan(‘m-ibmln2699.txt’)
y=x[4:864] # select the output: r(t)
# obtain the input variables: r(t-1), r(t-2), and r(t-3)
ibm.x=cbind(x[3:863],x[2:862],x[1:861])
# build a 3-2-1 network with skip layer connections
# and linear output.
ibm.nn=nnet(ibm.x,y,size=2,linout=T,skip=T,maxit=10000,
decay=1e-2,reltol=1e-7,abstol=1e-7,range=1.0)
# print the summary results of the network
summary(ibm.nn)
# compute & print the residual sum of squares.
sse=sum((y-predict(ibm.nn,ibm.x))^2)
print(sse)

# setup the input variables in the forecasting subsample
ibm.p=cbind(x[864:887],x[863:886],x[862:885])
# compute the forecasts
yh=predict(ibm.nn,ibm.p)
# The observed returns in the forecasting subsample
yo=x[865:888]
# compute & print the sum of squares of forecast errors
ssfe=sum((yo-yh)^2)
print(ssfe)

**Remark**: One-step ahead Out-of-sample-forecasts using `nnet` command. A R script, `backnnet.R`, is developed to carry out the evaluation of 1-step ahead out-of-sample forecasts. For illustration,

`> source(‘backnnet.R’)`
> m3=backnnet(x,y,nsize,orig,nl,nsk,miter)


**Analysis of High-Frequency Financial Data & Market Microstructure**

**Market microstructure**: Why is it important?

1. Important in market design & operation, e.g. to compare different markets (NYSE vs NASDAQ)
2. To study price discovery, liquidity, volatility, etc.
3. To understand costs of trading
4. Important in learning the consequences of institutional arrangements on observed processes, e.g.
   - Nonsynchronous trading
   - Bid-ask bounce
   - Impact of changes in tick size, after-hour trading, etc.
   - Impact of daily price limits (many foreign markets)

**Nonsynchronous trading**: Key implication: may induce serial correlations even when the underlying returns are iid.

Setup: log returns \( \{r_t\} \) are iid \((\mu, \sigma^2)\)

For each time index \( t \), \( P(\text{no trade}) = \pi \).

Cannot observe \( r_t \) if there is no trade.
What is the observed log return series $r_t^o$?

It turns out $r_t^o$ is given in Eq. (5.1),

$$ r_t^o = \begin{cases} 
0 & \text{with prob. } \pi \\
 r_t & \text{with prob. } (1 - \pi)^2 \\
 r_t + r_{t-1} & \text{with prob. } (1 - \pi)^2 \pi \\
 \vdots & \vdots \\
 \Sigma_{i=0}^k r_{t-i} & \text{with prob. } (1 - \pi)^2 \pi^k \\
 \vdots & \vdots 
\end{cases} $$

One can use this relation to show that

$$ \text{Var}(r_t^o) = \sigma^2 + \frac{2\pi \mu^2}{1 - \pi} $$

$$ \text{Cov}(r_t^o, r_{t-j}^o) = -\mu^2 \pi^j, \quad j \geq 1. $$

**Bid-ask bounce**

Bid and ask quotes introduce **negative** lag-1 serial correlation.

Setup: simplest case of Roll(1984)

True price $P_t^* = \frac{P_a + P_b}{2}$ is unchanged over time, i.e. $P_t^* = P_{t-1}^*$

$S = P_a - P_b$ is the bid-ask spread

$$ P_t = P_t^* + \begin{cases} 
S/2 & \text{with prob. 0.5} \\
-S/2 & \text{with prob. 0.5} 
\end{cases} $$

Then, $P_t = P_t^* + \frac{S}{2} I_t$ and

$$ \Delta P_t \equiv P_t - P_{t-1} = (I_t - I_{t-1}) \frac{S}{2} $$

where $I_t$ and $I_{t-1}$ are independent binary variables with $P(I_i = 1) = 0.5$ and $P(I_i = -1) = 0.5$.

Note: $E(I_t) = 0$ and $\text{Var}(I_t) = 1$ for all $t$. 

One can show that
\[
\begin{align*}
\text{Var}(\Delta P_t) &= \frac{S^2}{2} \\
\text{Cov}(\Delta P_t, \Delta P_{t-1}) &= -\frac{S^2}{4} \\
\text{Cov}(\Delta P_t, \Delta P_{t-j}) &= 0, \quad j > 1.
\end{align*}
\]

The result continues to hold if \( P_t^* \) follows a random walk model. That is, \( P_t^* = P_{t-1}^* + e_t \) with \( e_t \sim iid(0, \sigma^2_2) \).

**High-Frequency Financial Data**
Observations taken with time intervals 24 hours or less

Some examples:
1. Transaction (or tick-by-tick) data
2. 5-minute returns in FX
3. 1-minute returns on index futures and cash market

Some Basic Features of the Data:
1. Irregular time intervals
2. Leptokurtic or Heavy tails
3. Discrete values, e.g. price in multiples of tick size
4. Large sample size
5. Multi-dimensional variables, e.g. price, volume, quotes, etc.
6. Diurnal Pattern

**An illustration:** Consider the transaction-by-transaction data of Johnson and Johnson from October 4 to October 15, 2010. There
are 418,855 intraday price changes. Original data are from NYSE TAQ.

Time plot and histogram of intraday price changes in consecutive trades: See Figure 2. The histogram indicates most transactions are without price change.

The number of transactions in 5-min time intervals: (a) Time plot and (b) ACF: See Figure 3. The ACF shows a clear diurnal pattern in trading intensity.

**R demonstration**

```r
> da=read.table("taq-jnj-t-oct4t152010.txt",header=T)
> head(da)
   date hour minute second  price  volume
1 20101004     6     25    15  61.75     100
2 20101004     8     33    19  61.56     100
3 20101004     8     41     9  61.56     100
4 20101004     8     48    50  61.60     100
5 20101004     8     48    55  61.60     100
6 20101004     8     49     4  61.60     100

> source("hfchg.R") ### R script to compute price change
> m1=hfchg(da)

number of trading days: 10
> names(m1)
[1] "pchange" "duration" "size"
> par(mfcol=c(2,1)); idx=c(410000:418854)
> plot(m1$pchange,type='l',ylab='change') #plot(idx,m1$pchange[idx],type='l',ylab='pch')
> hist(m1$pchange, nclass=400, xlim=c(-0.04,0.04)) ### May use xlim=c(-0.06,0.06)

> source("hfntra.R") # R script to tabulate number of transactions in a given time interval (measured in minutes).

> m1=hfntra(da,5)
> names(m1)
[1] "ntrad"

**Frequencies of price change**

<table>
<thead>
<tr>
<th>Number(tick)</th>
<th>$\leq -2$</th>
<th>$[-2, -1)$</th>
<th>$[-1, 0)$</th>
<th>0</th>
<th>$(0, 1]$</th>
<th>$(1, 2]$</th>
<th>$\geq 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counts</td>
<td>540</td>
<td>1794</td>
<td>55325</td>
<td>304067</td>
<td>54860</td>
<td>1711</td>
<td>558</td>
</tr>
<tr>
<td>Percentage</td>
<td>0.128</td>
<td>0.428</td>
<td>13.21</td>
<td>72.60</td>
<td>13.10</td>
<td>0.408</td>
<td>0.132</td>
</tr>
</tbody>
</table>
Figure 2: Time plot and histogram of intraday price changes in consecutive trades for JNJ stock from October 4 to October 15, 2010. Only a small portion of the price changes (418854 data points) is shown in the upper plot.

**Econometric models used in the literature**

1. Duration models, e.g. autoregressive conditional duration (ACD) models.

2. Models for price changes

3. Models for bid and ask quotes

We focus on simple models for price change.

**Price Change**: Discrete values


1 **ADS Decomposition Models**

A simple ADS decomposition:
Figure 3: Time plot of the number of transactions in 5-min time intervals and its sample ACF for JNJ stock from October 4 to October 15, 2010.

- Price $P_t = P_0 + \sum_i^{N(t)} C_i$
- Number of transactions in $[0,t]: N(t)$
- $C_i = A_i D_i S_i$
  - Action:
    $$A_i = \begin{cases} 
    1 & \text{if } C_i \neq 0 \\
    0 & \text{otherwise}
    \end{cases}$$
  - Direction, given $A_i = 1$:
    $$D_i = \begin{cases} 
    1 & \text{if } C_i > 0 \\
    -1 & \text{if } C_i < 0
    \end{cases}$$
  - Size, given $A_i = 1$ and $D_i$: multiple of tick size
- Can be estimated by the logistic regression method
  $\logit: \ln[p/(1 - p)] = \text{linear function of explanatory variables}$
A brief introduction of logistic regression: A case of two explanatory variables \( X \) and \( Z \). The probability \( p_i \) is related to the observed values \( X = x_i \) and \( Z = z_i \) via the equation

\[
\ln[p_i/(1 - p_i)] = \beta_0 + \beta_1 x_i + \beta_2 z_i.
\]

This is equivalent to

\[
p_i = \frac{\exp(\beta_0 + \beta_1 x_i + \beta_2 z_i)}{1 + \exp(\beta_0 + \beta_1 x_i + \beta_2 z_i)}.
\]

It has many applications, e.g. probability of approving a loan based on the social and economic variables of an applicant.

We can use the command \texttt{glm} in R to perform logistic regression analysis.

Model specification of ADS models:

- **Action** \( A_i \): Governed by a logistic regression
  \[
P(A_i = 1|F_{i-1}) = \logit(F_{i-1})
  \]

- **Direction** given \( A_i = 1 \):
  \[
P(D_i = 1|F_{i-1}, A_i = 1) = \logit(A_i, F_{i-1})
  \]

- **Size** given \( A_i = 1 \) and \( D_i \):
  \[
P(S_i = s|A_i = 1, D_i = 1, F_{i-1}) \sim 1 + g(\lambda_{u,i})
  \]
  \[
P(S_i = s|A_i = 1, D_i = -1, F_{i-1}) \sim 1 + g(\lambda_{d,i})
  \]

where \( g(.) \) denotes a Geometric distribution and \( \lambda_{j,i} \) is governed by a logistic equation:

\[
\ln\left(\frac{\lambda_{j,i}}{1 - \lambda_{j,i}}\right) = \text{linear function of } F_{i-1}, A_i = 1, D_i.
\]
Likelihood function:

\[ P(C_i = s | F_{i-1}) = \]

\[ P(S_i = s | A_i = 1, D_i, F_{i-1})P(D_i | A_i = 1, F_{i-1})P(A_i = 1 | F_{i-1}). \]

An example: IBM data 59,775 observations. (Example 5.2 of the textbook.)

- Predictors: \( \{A_{i-1}, D_{i-1}, S_{i-1}, V_{i-1}, x_{i-1}, BA_i\} \)
  
  1. \( V_{i-1} \): volume of the previous trade (divided by 1000)
  2. \( x_{i-1} \): previous duration
  3. \( BA_i \): the prevailing bid-ask spread

- Model:
  
  1. Action: \( P(A_i | F_{i-1}) = p_i, \logit(p_i) = \beta_0 + \beta_1 A_{i-1} \)
  2. Direction: \( P(D_i = 1 | A_i = 1, F_{i-1}) = \gamma_i, \)
     \( \logit(\gamma_i) = \delta_0 + \delta_1 D_{i-1} \)
  3. Size: \( \logit(\lambda_{j,i}) = \theta_{j,0} + \theta_{j,1} S_{i-1} \) with \( j = d \) or \( u \).

- Results:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>( \beta_0 )</th>
<th>( \beta_1 )</th>
<th>( \delta_0 )</th>
<th>( \delta_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimate</td>
<td>-1.057</td>
<td>0.962</td>
<td>-0.067</td>
<td>-2.307</td>
</tr>
<tr>
<td>Std.Err.</td>
<td>0.104</td>
<td>0.044</td>
<td>0.023</td>
<td>0.056</td>
</tr>
<tr>
<td>Parameter</td>
<td>( \theta_{u,0} )</td>
<td>( \theta_{u,1} )</td>
<td>( \theta_{d,0} )</td>
<td>( \theta_{d,1} )</td>
</tr>
<tr>
<td>Estimate</td>
<td>2.235</td>
<td>-0.670</td>
<td>2.085</td>
<td>-0.509</td>
</tr>
<tr>
<td>Std.Err.</td>
<td>0.029</td>
<td>0.050</td>
<td>0.187</td>
<td>0.139</td>
</tr>
</tbody>
</table>

Implication
1. Prob of price change:

\[ P(A_i = 1 | A_{i-1} = 0) = 0.258 \]
\[ P(A_i = 1 | A_{i-1} = 1) = 0.476. \]

2. Interpretation: **Odds ratio**

Because \( A_{i-1} \) is also a binary variable, we have a 2 \( \times \) 2 table:

<table>
<thead>
<tr>
<th>Outcome ( A_i )</th>
<th>Independent variable ( A_{i-1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_{i-1} = 1 )</td>
<td>( P(A_i = 1) = \frac{\exp(\beta_0 + \beta_1)}{1 + \exp(\beta_0 + \beta_1)} )</td>
</tr>
<tr>
<td>( A_{i-1} = 0 )</td>
<td>( P(A_i = 0) = \frac{1}{1 + \exp(\beta_0 + \beta_1)} )</td>
</tr>
</tbody>
</table>

**Odds Ratio**: Row one divided by Row 2, then Column 1 divided by Column 2.

\[ OR = e^{\beta_1}, \quad \text{or} \quad \beta_1 = \ln(OR). \]

3. Direction of price change:

\[ P(D_i = 1 | F_{i-1}, A_i) = \begin{cases} 0.483 & \text{if } D_{i-1} = 0, \text{ i.e. } A_{i-1} = 0 \\ 0.085 & \text{if } D_{i-1} = 1, A_i = 1 \\ 0.904 & \text{if } D_{i-1} = -1, A_i = 1 \end{cases} \]

Bid-ask bounce

4. Weak evidence of price change cluster: price increases

\[ S_i | (D_i = 1) \sim 1 + g(\lambda_{u,i}), \quad \lambda_{u,i} = 2.235 - 0.670S_{i-1}. \]

**R demonstration**: **glm** stands for generalized linear model.
> da=read.table("ibm91-ads.txt",header=T)
> da1=read.table("ibm91-adsx.txt",header=T)
> head(da)
   Ai  Di  Si
 1  0  0  0
 2  0  0  0
 3  0  0  0
 4  0  0  0
 5  1  1  1
 6  1 -1  1
> head(da1)
   Vim1 Durm1 BAi Aim1 Dim1 Sim1
 1     8  0.4  0.125  0  0  0
 2     0  0.1  0.370  0  0  0
 3     1  1.0  0.125  0  0  0
 4     5  0.1  0.125  0  0  0
 5     4  0.1  0.625  0  0  0
 6  62  1.0  0.625  1  1  1
> Ai=da$Ai
> Di=da$Di
> Aim1=da1$Aim1
> Dim1=da1$Dim1
> m1=glm(Ai~Aim1,family=binomial)  % fit a linear logistic regression
> summary(m1)
Call:
glm(formula = Ai ~ Aim1, family = binomial)
Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.05667  0.01142 -92.55 <2e-16 *** % See Table 5.6 of the text
Aim1        0.96164  0.01827  52.62 <2e-16 *** 
---

> di=Di[Ai==1]
> dim1=Dim1[Ai==1]
> di=(di+abs(di))/2  % Transform di into a binary variable
> m2=glm(di~dim1,family=binomial)
> summary(m2)
Call:
glm(formula = di ~ dim1, family = binomial)
Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.06663  0.01728  -3.855  0.000116 *** % See Table 5.6 of the text
dim1       -2.30693  0.03595  -64.171 < 2e-16 *** 
---
2 Ordered Probit Model

Let $y_i^*$ be the unobservable price change of the asset under study (i.e., $y_i^* = P_t^* - P_{t-1}^*$), where $P_t^*$ is the virtual price of the asset at time $t$. The ordered probit model assumes that $y_i^*$ is a continuous random variable and follows the model

$$y_i^* = x_i \beta + \epsilon_i,$$  \hspace{1cm} (1)

where $x_i$ is a $p$-dimensional row vector of explanatory variables available at time $t_{i-1}$, $\beta$ is a $p \times 1$ parameter vector, $E(\epsilon_i|x_i) = 0$, $\text{Var}(\epsilon_i|x_i) = \sigma_i^2$, and $\text{Cov}(\epsilon_i, \epsilon_j) = 0$ for $i \neq j$. The conditional variance $\sigma_i^2$ is assumed to be a positive function of the explanatory variable $w_i$ — that is,

$$\sigma_i^2 = g(w_i),$$  \hspace{1cm} (2)

where $g(.)$ is a positive function. For financial transactions data, $w_i$ may contain the time interval $t_i - t_{i-1}$ and some conditional heteroscedastic variables. Typically, one also assumes that the conditional distribution of $\epsilon_i$ given $x_i$ and $w_i$ is Gaussian.

Suppose that the observed price change $y_i$ may assume $k$ possible values. In theory, $k$ can be infinity, but countable. In practice, $k$ is finite and may involve combining several categories into a single value. For example, we have $k = 7$ in Table 1, where the first value “$< -2$ cents” means that the price drops more than 2 cents. We denote the $k$ possible values as $\{s_1, \ldots, s_k\}$. The ordered probit model postulates the relationship between $y_i$ and $y_i^*$ as

$$y_i = s_j \quad \text{if} \quad \alpha_{j-1} < y_i^* \leq \alpha_j, \quad j = 1, \cdots, k,$$  \hspace{1cm} (3)
where $\alpha_j$ are real numbers satisfying $-\infty = \alpha_0 < \alpha_1 < \cdots < \alpha_{k-1} < \alpha_k = \infty$. Under the assumption of conditional Gaussian distribution, we have

$$P(y_i = s_j|x_i, w_i) = P(\alpha_{j-1} < x_i \beta + \epsilon_i \leq \alpha_j | x_i, w_i)$$

\begin{align*}
&= \begin{cases} 
P(x_i \beta + \epsilon_i \leq \alpha_1 | x_i, w_i) & \text{if } j = 1, \\
P(\alpha_{j-1} < x_i \beta + \epsilon_i \leq \alpha_j | x_i, w_i) & \text{if } j = 2, \ldots, k - 1, \\
P(\alpha_{k-1} < x_i \beta + \epsilon_i | x_i, w_i) & \text{if } j = k,
\end{cases}
\end{align*}

\begin{align*}
&= \begin{cases} 
\Phi \left[ \frac{\alpha_1 - x_i \beta}{\sigma_i(w_i)} \right] & \text{if } j = 1, \\
\Phi \left[ \frac{\alpha_j - x_i \beta}{\sigma_i(w_i)} \right] - \Phi \left[ \frac{\alpha_{j-1} - x_i \beta}{\sigma_i(w_i)} \right] & \text{if } j = 2, \ldots, k - 1, \\
1 - \Phi \left[ \frac{\alpha_{k-1} - x_i \beta}{\sigma_i(w_i)} \right] & \text{if } j = k,
\end{cases}
\end{align*}

where $\Phi(x)$ is the cumulative distribution function of the standard normal random variable evaluated at $x$, and we write $\sigma_i(w_i)$ to denote that $\sigma_i^2$ is a positive function of $w_i$. From the definition, an ordered probit model is driven by an unobservable continuous random variable. The observed values, which have a natural ordering, can be regarded as categories representing the underlying process. See Figure 4 for a case of $k = 5$.

The ordered probit model contains parameters $\beta$, $\alpha_i$ ($i = 1, \ldots, k - 1$), and those in the conditional variance function $\sigma_i(w_i)$ in Eq. (2). These parameters can be estimated by the maximum likelihood or Markov chain Monte Carlo methods. In this handout, we use the command `polr` of the R package `MASS` to estimate ordered probit.
Example 6.1. To illustrate we consider the intraday price changes of Caterpillar stock on January 4, 2010. There are 37,716 transactions during the normal trading hours so that we have 37,715 price changes. For simplicity, we classify the price change into 7 categories shown in Table 1. Our analysis focuses on the dynamic dependence of intraday price changes. As such, we define indicator (or dummy) variables for lagged price changes:

\[ y_{\ell,j} = \begin{cases} 
1 & \text{if } y_{i-\ell} = s_j \\
0 & \text{otherwise,} 
\end{cases} \]

where \( s_j \) denotes the \( j \)th category of price change and \( y_{i-\ell} \) is the \((i-\ell)\)th price change at time \( t_{i-\ell} \), where \( j = 2, \ldots, 7 \) and \( \ell = 1 \) and 2. In other words, we employ the classifications of price changes for the previous 2 consecutive trades. As usual, with 7 categories, only six indicator variables are needed in modeling.

We also employ the observed price changes \( y_{i-\ell} \) for \( \ell = 1, 2, 3 \) and the lag-2 transaction volume defined as \( v_{i-2} = V_{i-2}/100 \), where \( V_{i-2} \)
Table 1: Frequencies of Price Change for Caterpillar Stock on January 4, 2010.

<table>
<thead>
<tr>
<th>Category</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cents</td>
<td>&lt; -2</td>
<td>[-2, -1)</td>
<td>[-1, 0)</td>
<td>0</td>
<td>(0, 1]</td>
<td>(1, 2]</td>
<td>&gt; 2</td>
</tr>
<tr>
<td>Percentage</td>
<td>0.605</td>
<td>1.692</td>
<td>15.20</td>
<td>64.98</td>
<td>15.04</td>
<td>1.832</td>
<td>0.655</td>
</tr>
</tbody>
</table>

is the actual volume. We do not use price volume because price is relatively stable in a trading day. Consequently, the model entertained is

\[ \mathbf{x}_i \beta = \beta_1 v_{i-2} + \sum_{\ell=1}^{3} \beta_{1+\ell} y_{i-\ell} + \sum_{j=2}^{7} \gamma_{1,j} y_{1,j} + \sum_{j=2}^{7} \gamma_{2,j} y_{2,j}. \]  (5)

For simplicity, we start with \( \sigma_i^2(\mathbf{w}_i) = \sigma^2 \), a constant. Parameter estimates of the model are given in Table 2, where all estimates but one are statistically significant at the usual 5% level. The parameter estimates of Eq. (5) are negative, because a negative sign is used in Equation (6). As a matter of fact, the model shown is a simplified one after removing some explanatory variables that were not statistically significant. For instance, we also included the time duration \( \Delta t_i = t_i - t_{i-1} \) in the preliminary analysis and decided to drop the variable because its estimate is not statistical significant at the 5% level. The significance of the indicator variables shows that there exists dynamic dependence in intraday price change. The fitted model thus can be used to provide probability forecasts for the next transaction price change. Indeed, the model provides probability for each category of price change at each transaction.

It is interesting to study the fitted boundary partitions of the ordered probit model in Table 2. First, because the explanatory variables may have nonzero means, the estimates of boundary parameters \( \alpha_i \) are not symmetric with respect to zero. Second, \( \hat{\alpha}_2 - \hat{\alpha}_1 = 0.577 \) and \( \hat{\alpha}_6 - \hat{\alpha}_5 = 0.601 \). The two intervals roughly have the same length. Similarly,
Table 2: Estimation Results of an Ordered Probit Model for the Intraday Price Changes of Caterpillar Stock on January 4, 2010 with 37,716 transactions. The Model is in Equation (5) and $t$ Denotes $t$-ratio.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$\alpha_1$</th>
<th>$\alpha_2$</th>
<th>$\alpha_3$</th>
<th>$\alpha_4$</th>
<th>$\alpha_5$</th>
<th>$\alpha_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimate</td>
<td>-4.594</td>
<td>-4.017</td>
<td>-2.860</td>
<td>-0.853</td>
<td>0.287</td>
<td>0.888</td>
</tr>
<tr>
<td>$t$</td>
<td>-31.48</td>
<td>-27.80</td>
<td>-19.89</td>
<td>-5.944</td>
<td>2.000</td>
<td>6.188</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Par.</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\beta_3$</th>
<th>$\beta_4$</th>
<th>$\gamma_{1,2}$</th>
<th>$\gamma_{1,3}$</th>
<th>$\gamma_{1,4}$</th>
<th>$\gamma_{1,5}$</th>
<th>$\gamma_{1,6}$</th>
<th>$\gamma_{1,7}$</th>
<th>$\gamma_{1,8}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Est.</td>
<td>0.004</td>
<td>7.837</td>
<td>10.86</td>
<td>12.28</td>
<td>0.274</td>
<td>0.743</td>
<td>1.331</td>
<td>1.858</td>
<td>2.262</td>
<td>2.493</td>
<td>0.099</td>
</tr>
<tr>
<td>$t$</td>
<td>3.983</td>
<td>5.363</td>
<td>7.098</td>
<td>15.93</td>
<td>2.971</td>
<td>8.173</td>
<td>13.81</td>
<td>17.83</td>
<td>18.57</td>
<td>15.95</td>
<td>1.053</td>
</tr>
</tbody>
</table>

$\hat{\alpha}_3 - \hat{\alpha}_2 = 1.157$, which is close to $\hat{\alpha}_5 - \hat{\alpha}_4 = 1.140$. These results are consistent with the empirical observation that price changes appear to be roughly symmetric with respect to zero shown in Table 1.

Finally, the model implies

$$P(y_i^* \leq s_j | x_i, w_i) = \Phi(\alpha_j - x_i \beta),$$

for the Caterpillar transaction data, where $\Phi(.)$ is the cumulative distribution function of $N(0, 1)$.

**Discussion** The command `polr` allows for pre-determined weights to handle heteroscedasticity, but it cannot perform simultaneous estimation of the volatility and probit equations. See Hauseman, Lo, and MacKinlay (1992) and Tsay (2010) for some examples with time-varying $\sigma_i^2(w_i)$ function. Finally, as usual, only 6 indicator variables are needed for each lagged value of $y_i$.

**R Demonstrations for Ordered Probit Models**

Output edited.
> da=read.table("taq-cat-t-jan042010.txt",header=T)
> head(da)
   date hour minute second price size
1 20100104   9     30      0  57.65  3910
6 20100104   9     30      1  57.65   462
> vol=da$size/100
> da1=read.table("taq-cat-cpch-jan042010.txt")
> cpch=da1[,1]  % category of price change
> pch=da1[,2]  % price change
> cf=as.factor(cpch)  % create categories in R
> length(cf)
[1] 37715
> y=cf[4:37715]
> y1=cf[3:37714]  % create indicator variables for lag-1 cpch
> y2=cf[2:37713]  % create indicator variables for lag-2 cpch
> vol=vol[2:37716]
> v2=vol[2:37713]  % create lag-2 volume
> cp1=pch[3:37714]  % select lagged price changes
> cp2=pch[2:37713]; cp3=pch[1:37712]

> library(MASS)  % load package
> m1=polr(y~v2+cp1+cp2+cp3+y1+y2,method="probit")
> summary(m1)
Call: polr(formula = y ~ v2 + cp1 + cp2 + cp3 + y1 + y2, method = "probit")
Coefficients:
   Value  Std. Error   t value
v2  -0.003765  0.00094534  -3.983
cp1  -7.836883  1.46130467  -5.363
cp2  -10.864394  1.53064556  -7.098
cp3 -12.283682  0.77109550 -15.930
y12  -0.274407  0.09235665  -2.971
y13  -0.742792  0.09088543  -8.173
y14  -1.330665  0.09635403 -13.810
y15  -1.858199  0.10422568 -17.829
y16  -1.858199  0.10422568 -17.829
y17  -1.493321  0.15631772 -9.527
y18  -0.098542  0.09359076  -1.053
y19  -0.307034  0.09237250  -3.324
y20  -0.531115  0.09801503  -5.419
y21  -0.744706  0.10624352  -7.009
y22  -0.932655  0.12389182  -7.528
y27 -0.858858 0.1596219 -5.381

Intercepts:

<table>
<thead>
<tr>
<th>Value</th>
<th>Std.Error</th>
<th>t value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2 -4.5941 0.1459 -31.4803</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3 -4.0170 0.1445 -27.7989</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4 -2.8599 0.1438 -19.8926</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5 -0.8528 0.1435 -5.9437</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6 0.2868 0.1434 1.9996</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7 0.8882 0.1435 6.1883</td>
<td></td>
</tr>
</tbody>
</table>

Residual Deviance: 74802.56
AIC: 74846.56

> names(m1)
[1] "coefficients" "zeta" "deviance" "fitted.values"
[5] "lev" "terms" "df.residual" "edf"
[9] "n" "nobs" "call" "method"
[13] "convergence" "niter" "lp" "model"
[17] "contrasts" "xlevels"

> yhat=m1$fitted.values
> print(yhat[1:5,],digits=3)

1 2 3 4 5 6 7
1 1.11e-03 0.005420 0.08605 0.660 0.2134 0.0266 0.007696
2 1.55e-02 0.041461 0.27883 0.608 0.0535 0.0028 0.000444
3 8.99e-06 0.000094 0.00522 0.287 0.4311 0.1605 0.116298
4 1.87e-04 0.001251 0.03267 0.539 0.3343 0.0658 0.027144
5 6.41e-04 0.003470 0.06457 0.630 0.2527 0.0365 0.011836