

Lecture 5 Homework and Project: More Multivariate and NonNormal
Exploratory Data Analysis {# Corrected 2/10,apm}

(Homework Due by Lecture 6 in Chalk FINM331 Digital Dropbox) and Project
Professional Report Due by Lecture 7 in Chalk FINM331 Digital Dropbox

You must show your work, code and/or worksheet for full credit.

1. (20 points) In order to convert an invertible, fully infinite domain distribution $F_X(x; \vec{\theta})$, with parameter vector $\vec{\theta}$, to a realistic, renormalized finite domain (FD) distribution, on $R_1 < \mu < R_2$,

$$u = F_X^{(\text{fd})}(x; \vec{\theta}, R_1, R_2) \equiv \left(F_X(x; \vec{\theta}) - F_X(R_1; \vec{\theta}) \right) / \left(F_X(R_2; \vec{\theta}) - F_X(R_1; \vec{\theta}) \right),$$

where u is a standard uniform sample variable, i.e., on $(0, 1)$.

- (a) Show that the finite domain sample variable x on (R_1, R_2) is related to the standard uniform sample variable by

$$x = (F_X)^{-1} \left(\left(F_X(R_2; \vec{\theta}) - F_X(R_1; \vec{\theta}) \right) \cdot u + F_X(R_1; \vec{\theta}) \right).$$

- (b) In general, the properties of the parameter vector will not be preserved except partially in the some cases with symmetric ranges, so show this for the case of the finite-domain normal (FDN) with statistics $(\mu^{(\text{fdn})}, (\sigma^{(\text{fdn})})^2)$ by showing

$$\mu^{(\text{fdn})} = \mu - \sigma^2(f_2 - f_1)/F_{12}$$

where $F_1 \equiv F_X^{(n)}(R_1; \mu, \sigma^2)$, $F_2 \equiv F_X^{(n)}(R_2; \mu, \sigma^2)$, $F_{12} \equiv F_2 - F_1$, $f_1 \equiv f_X^{(n)}(R_1; \mu, \sigma^2)$, $f_2 \equiv f_X^{(n)}(R_2; \mu, \sigma^2)$, and

$$\begin{aligned} (\sigma^{(\text{fdn})})^2 &= (\mu - \mu^{(\text{fdn})})^2 + \sigma^2(1 - ((R_2 - \mu)f_2 + (\mu - R_1)f_1)/F_{12} \\ &\quad - 2(\mu - \mu^{(\text{fdn})})(f_2 - f_1)/F_{12}) \end{aligned}$$

{Hint: Try integration by parts only with powers of $(x - \mu)$ multiplying the density.}

- (c) Show, if R_1 and R_2 are symmetrically located about the mean μ , then $\mu^{(\text{fdn})} = \mu$, but that $(\sigma^{(\text{fdn})})^2 = \sigma^2(1 - 2(R_2 - \mu)f_2/F_{12}) \leq \sigma^2$.

2. (20 points) Apply the theoretical results of Problem 1:

- (a) To construct a MATLAB type finite-domain normal RNG subfunction, `fdnormrnd(mu, sigma, R1, R2, M, N)`, that uses `rand(M, N)` basic standard uniform RNG, `norminv` and `normcdf`. Next test the simulations with this RNG subfunction by
- (b) Producing a Q-Q plot of your HW2.4 S&P data against the FD-Normal simulations with parameters `mu`, `sigma` and domain range common with the data;
- (c) Compute the absolute difference between $\mu^{(\text{fdn})}$ and the common μ in (a), as well as the relative difference of $\sigma^{(\text{fdn})}$ relative to the common σ in (a).
- (d) A `hist3` histogram plot of the same with 50 bins for each;

- (e) Superimpose a plot of the corresponding finite-domain normal (FDN) density $f_X^{(\text{fdn})}(x; \vec{\theta}, R_1, R_2)$ with the common parameters as in (b) onto the 50-bin histogram of the S&P data;
- (f) Investigate how expanding the range beyond the S&P data range would help in fitting the tails, say by adding 15% on each tail, under the idea that the sample represented by the data is too small to represent the most extreme jumps.

Discuss the results, with particular emphasis on the handling of fat tails.

Duplicate Problem 3 Deleted.

FINM3312/STAT339 Individual Project for Weeks 5-6: Least Squares Regression

Due by Lecture 7 and worth 100 points

Corrected in red: 02/11/2008

- **General Problem Objectives:**

Often in Industry, a worker has to test numerical procedures before selecting which one will be used in production numerical procedures. In MATLAB you can fit dependent ordinate data (e.g. in y) as a function of the independent coordinate x by two (2) of three (3) methods as, long as the function being fitted is a polynomial. The model polynomials will be cubics (degree 3) and will be used to fit separately quarterly Stock Market Index Log-Return Means and quarterly Log-Return Volatilities (Standard Deviations) for Standard and Poor's 500 Data from of Homework 2 – Problem 4. Finally, you will write a professional report on your methods and results.

- **Methodologies:**

Choose at least two of the first three fit methods, in addition to `regress`:

1. `polyfit` and `polyval` polynomial functions (use MATLAB `help polyfit` and `help polyval` commands).
2. `\` back-slash or (pseudo-)inverse operator (See MATLAB `help SLASH`).
3. `svd` or Singular Value Decomposition function (See MATLAB `help svd`).
4. `tbbtregress` or Multiple Linear Regression function (See MATLAB `help regress`).

Remark: If you are familiar with the statistical packages SAS or SPSS, you can substitute for "svd" the GLM or Reg function of SAS or the Regression function of SPSS, similarly with public-domain statistics packages like R or S.)

The objective is to compare two of three methods by fitting to existing data to see which one of the two you should recommend to the boss.

- **Problem Statement:**

The problem is to compare three of four methods on the Standard and Poor's 500 Stock Index Data by fitting both the quarterly stock log-return means and log-return volatilities (standard deviations).

- **Floating Point Time Conversion:** For each of the $4*4 = 16$ quarters for plotting purposes take the midpoint of the quarter

$$TM(iy + 4 * (iy - 1)) = (Y1 - 1) + iy + 0.25/2 + 0.25 * (iqy - 1),$$

where $Y1$ is the first year since year in $iy = 1$ should start TM , i.e. $Y1 = 2005$ here, for $iy = 1 : N$ years and $iqy = 1 : 4$ quarters (Jan-Mar, Apr-Jun, Jul-Sep, Oct-Dec) for each year. A better calculation would just count official trading day and take the fraction representing the midpoint of the quarter's trading days converted into fractions of a year, but the above formula should be used for simplicity.

- **Quarterly Means and Volatilities:** The ultimate objective is to fit the means for each quarter using the MATLAB `mean` function and the volatilities for each quarter using the MATLAB `std` function.
- **Fitting quarterly Means and Volatilities:** Two cubic polynomial fit models are required, one for quarterly log-return means and one for quarterly log-return volatilities. Assign the quarterly means and volatilities to the time at each

$TM(iqy + 4 * (iy - 1)), iy = 1 : N \ \& \ iqy = 1 : 4$

- **Better Conditioned Mid-Year Variable:** Using `polyfit`, MATLAB will rightly complain that the fit is ill-conditioned, so you must center (c) and scale (s) the quarterly time at mid-quarter, such that **for vector TM**

`tpoly = (TM-mean(TM))./std(TM);`

Remark 1: It would be POOR numerical practice for any of the three methods to use the original years of the data because years like 200X are such large numbers, so time does not differ very much in the N years.

Remark 2: Note that ordinary matrix division (/) is NOT used here, but array or element-wise division (./), i.e., there are two kinds of right-sided divisions and you have to use the correct one.

- **Some Hints on Methods:**

1. Use the MATLAB `ones` and `size` functions to construct the matrix of coefficients A of the vector polynomial coefficients $a = [a1 \ a2 \ a3 \ a4]$ in the cubic case:

`help ones`

`help size`

`A=[ones(size(x)) xx.^2x.^3]; % for cubic model fitting`

noting that array or component-wise exponentiation operation (\wedge) is needed rather than regular matrix exponentiation (\wedge) (WATCH those periods!).

2. Use `polyfit` to get the fit coefficient vector

`"a = apoly"`,

where **"apoly"** is the polynomial form outputted by `polyfit`, see item 3 below, from the input

`"x = tpoly"`

vector data and the corresponding output

`"y = y-data"`

vector data, then use `polyval` to compute the the predicted values of

`"ypred = ypoly"`.

3. Find out how to use the vector output arguments of

`[apoly, struct]=polyfit(tpoly, y, 3)`

and

```
[ypolypred, delpoly]=polyval(apoly, tpoly, struct)
```

to plot the 95% confidence intervals plotting the upper and lower bounding curves
"ypolypred±c*delpoly"

against "*the Time variable*" for appropriate values of the multiplier "c", which is the ratio of "delpoly95" to "delpoly50 = delpoly", where "delpoly95" is increment above and below the fit for a 95% confidence interval. Stuct is a structure form that is used to estimate the errors in the fit and is wise to use polyval to calculate the predicted polynomial values since the MATLAB coefficient vector is not the same used in class. See

`help polyval` and `help polyfit`

for specification on the 50% error bounds. Caution: One MATLAB guide says that "c" should be "2", but "c" is very close to "3" for the usual normal distribution assumption. *This confidence interval problem was an old bug and may have been fixed, but check it out.*

4. Use the back-slash function to solve the $A^*a=y$ problem with

```
aslash = A\y;
```

say, since when A has more rows (m) than columns (n) the back-slash also finds the least squares solution instead of the inverse when $m = n$. Do not forget that you have to find the predicted values $y=yslash$ given tpoly.

5. Use `svd` (see `help svd`) to get the Singular Value Decomposition of

```
A = U*D*V',
```

where V' is MATLAB for transpose of V. Do to the unusual format of the MATLAB `svd`, you will have to do the SVD inversions with extra parenthesis or extra steps to avoid MATLAB matrix algebra confusion, e.g.,

```
asvd = V * (D\ (U' * y));
```

Again you need to find the predicted y-values, say `ysvd` and the corresponding values, say `ypoly`, `yslash`, `ysvd`.

- **General Instructions:** For the two methods fit cubic polynomial models to both quarterly means and quarterly volatilities, you must also present documented output for

1. Plots using the MATLAB `plot` function comparing "y", "ypoly", "yslash" and "ysvd" against the original vector "tpoly", with appropriate labels, where "y" is either the quarterly mean or volatility vector.
2. Standard deviations (see `help std`) of residual or deviation vectors that are the difference between the two of three `ypoly`, `yslash` and `ysvd` predicted vectors from the original vector y, where y is either the the quarterly mean or volatility vector, i.e., find the least squares of the differences between the quarterly data and the cubic model at the same *mid-quarter* times.

3. Standard deviations (std) of the two of three differences (deviations) between each of predicted vectors from each of the two of three methods. Is there much difference between the two of three methods considering the numerical precision in MATLAB (put this answer in your problem comments)? Make of table summarizing the variances of the differences: ypoly versus yslash, yslash versus ysvd and ysvd versus ypoly, where y is either the the quarterly mean or volatility vector.
4. Use the `etime`, `tic` and `toc`,(see `help etime` etc.) function of MATLAB to time each of the three methods, adding the etime to calculate the normal matrix A to only the methods of "slash" and "svd". The purpose here is to measure the efficiency of your MATLAB code. (Caution: New version of MATLAB does not have flops, until a fix is prepared.)

- ***Project Report:***

Your *professional, individual report* needs the following parts:

1. ***Cover Page:*** Put a project title, your name, your affiliation, date and other identifying information on this individual computer project. What you submit must be your own work (this in NOT a group project) and points will be deducted for similar work.
2. ***Executive Summary:*** This is about a page summarizing the project and your results for a busy boss. This should be in the form of a outline or itemized list for easy and fast reading. Also, a summary or/and critical result graph(s) would be important.
3. ***Project Description or Introduction:*** Describe the project in your own words as an introduction to your report, in sufficient depth so that a reader such as yourself would understand it.
4. ***Methods:*** Describe the mathematics and the algoritms behind these methods used to solve the problem, giving both advantages and disadvantages in a fair manner.
5. ***Results:*** Describe the nature of the results and illustrate them with appropriate tables or plots. You can use MATLAB for plotting your results. Clearly label tables and plot figures in a professional manner.
6. ***Discuss:*** Discuss the results, including how they can be used elsewhere for different industrial applications. Explain how and why methods differ or do not differ.
7. ***Acknowledgements:*** Acknowledge what resources you used in this project, including what versions of MATLAB that you used, the operating system, the computer or hardware platform, persons consulted (important: grade is discounted for similar reports and unacknowledged use of other sources), and any other resources (references are listed in the next section) used.
8. ***Conclusions:*** List what you have learned from this project and explain why it is significant.

9. *References*: Cite all books, scientific papers, web-sites and other library or web resources that you used. Give author, title, journal name or book publisher or URL where appropriate, and date of publication or web access.
10. *Appendices*: Include MATLAB documented source code and output.