Some Recent Developments in Statistics

Alcuni sviluppi recenti della metodologia statistica

Elvezio Ronchetti

Department of Econometrics
University of Geneva
Switzerland

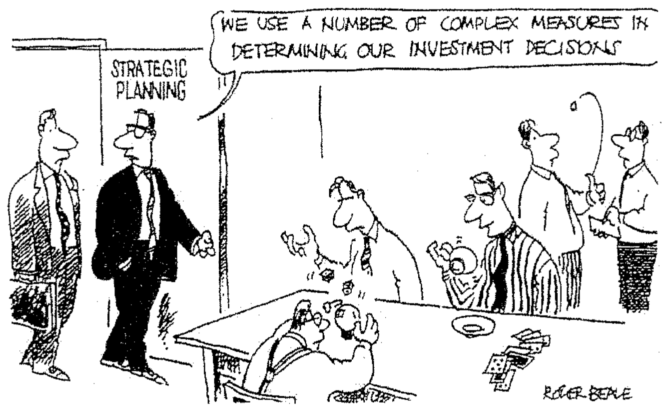
http://www.unige.ch/ses/metri/ronchetti/

Outline

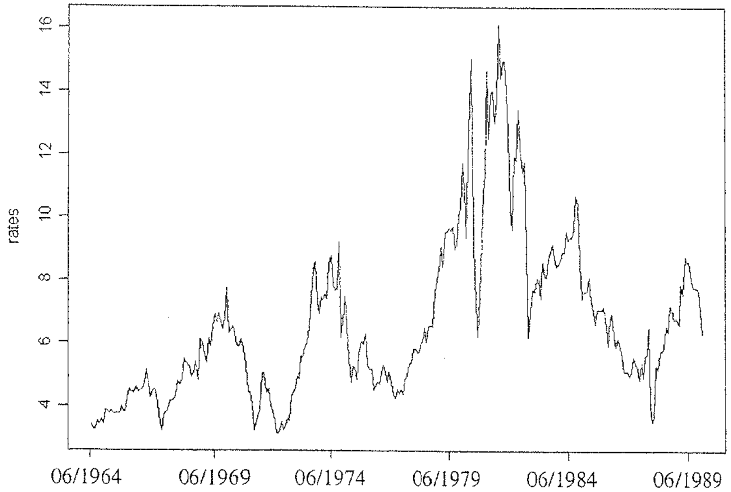
- Some statistical models for time series
- Bootstrap methods
- Conclusions

Statistical models for time series

- Stylized facts
- Diffusion models
- Estimation
- Prediction



US monthly risk-free rates



US monthly risk-free rates

- One-month yields based on the average of bid and ask prices for Treasury bills.
- Normalized to reflect a standard month of 30.4 days.
- They are monthly observations covering the period from June 1964 to December 1989, a total of 307 observations.

data = model + error

"All models are wrong, but some are useful."

G.E.P. Box

Models

Description of the data — Forecasting

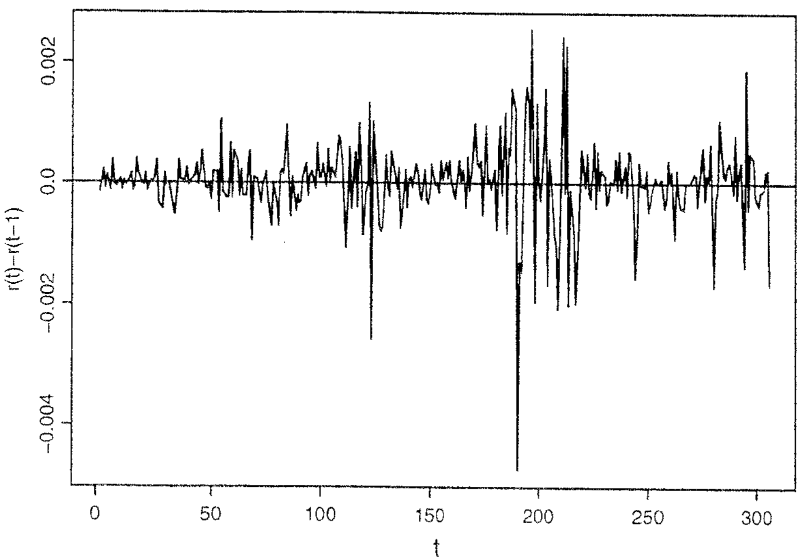
Which models?

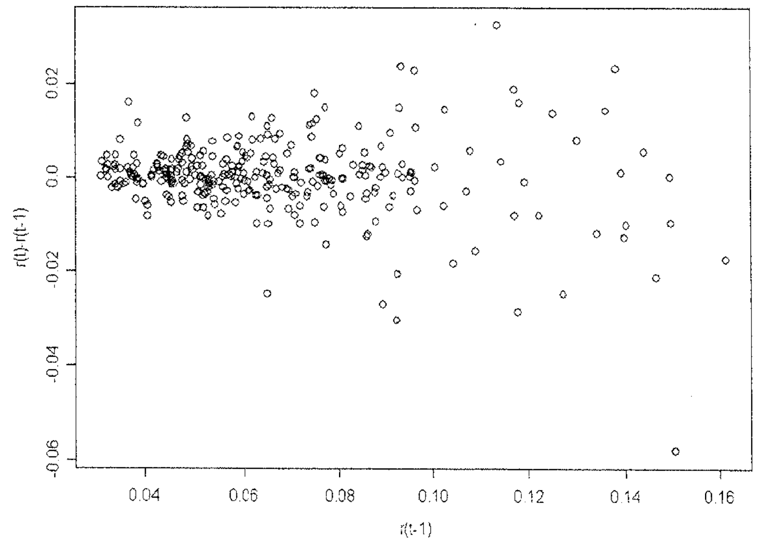
They should reflect observed features of the data, *e.g.* here

- Variance of monthly rate changes varies over time.

  => Model conditional distribution at time t given the information up to time $(t-1)$, *e.g.* ARCH, GARCH (Engle)

US rates: first differences





Stochastic model (*single-factor short-rate model*):

$$r_t - r_{t-1} = (\alpha + \beta\, r_{t-1}) + \sigma\, r_{t-1}^{\gamma} \epsilon_t$$

with ϵ_t identically distributed standard normal variables.

Drift: $\alpha + \beta r_{t-1}$

Volatility: $\mathrm{Var}(\sigma r_{t-1}^{\gamma} \epsilon_t | r_{t-1}) = \sigma^2 r_{t-1}^{2\gamma}$

Volatility at time t depends on the level of the interest rates up to time $t-1$.

Alternative Models of the Short Rate

$$r_t - r_{t-1} = (\alpha + \beta\, r_{t-1}) + \sigma\, r_{t-1}^{\gamma} \epsilon_t$$

Model	α	β	σ	γ
Merton		0		0
Vasicek				0
Cox Ing. Ross				$\frac{1}{2}$
Dothan	0	0		1
Geom. B. Mot.	0			1
Brenn. Schw.				1
Variab. Rate	0	0		$\frac{3}{2}$
Const. El. Var.	0			

Continuous time (*single-factor short-rate model*):

$$dr_t = \mu(r_t)dt + \sigma(r_t)dW(t)$$

where $W(t)$ is a standard Brownian motion.

Estimate
the drift $\mu(r_t)$ and the volatility $\sigma(r_t)$
parametrically or nonparametrically.

$$r_t - r_{t-1} = (\alpha + \beta\, r_{t-1}) + \sigma\, r_{t-1}^{\gamma} \epsilon_t$$

Model	α	β	σ	γ
Unres.	0.0034	−0.0493	0.3656	1.4999
Merton	0.0005	0	0.0062	0
Vasicek	0.0005	−0.0013	0.0062	0
CIR	0.0011	−0.0102	0.0254	$\frac{1}{2}$
Dothan	0	0	0.0320	1
GBM	0	0.0084	0.0993	1
BS	0.0020	−0.0262	0.0994	1
VR	0	0	0.05	$\frac{3}{2}$
CEV	0	0.0086	0.1554	1.1711

Quality of the models

↓

Based on predictive power

Simulation of the model

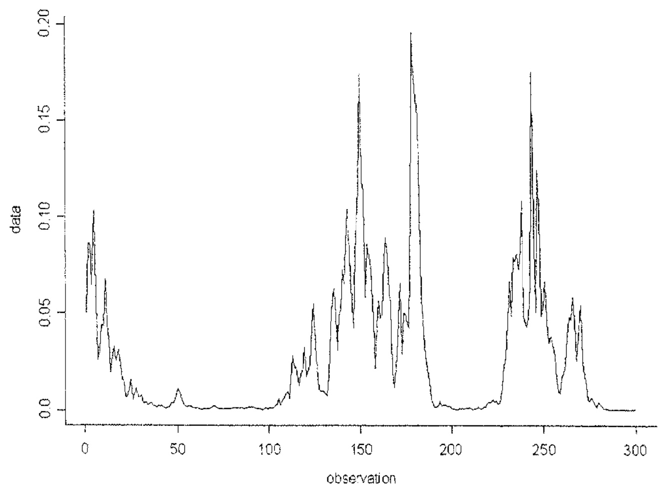
- What does it mean?
- How do we do that?
  - Generation of random normal variables $\epsilon_t \sim \mathcal{N}(0,1)$

Alternative Models of the Short Rate

$$r_t - r_{t-1} = (\alpha + \beta r_{t-1}) + \sigma r_{t-1}^{\gamma} \epsilon_t$$

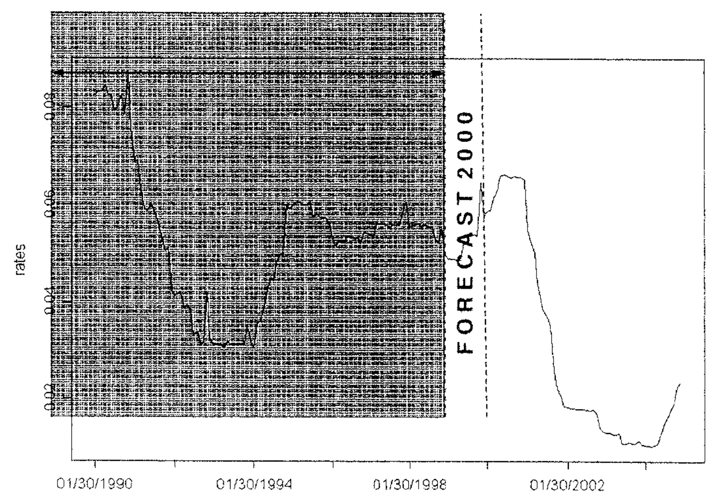
Model	α	β	σ	γ
Merton		0		0
Vasicek				0
Cox Ing. Ross				$\frac{1}{2}$
Dothan	0	0		1
Geom. B. Mot.	0			1
Brenn. Schw.				1
Variab. Rate	0	0		$\frac{3}{2}$
Const. El. Var.	0			

Simulated data from a GBM with $\beta = 0.1, \sigma = 0.5$:



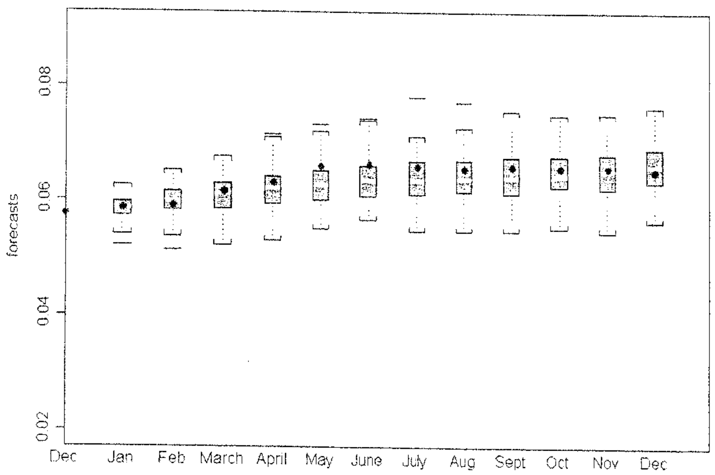
Cross-validation (out-of-sample performance)

Monthly Eurodollar rates



Millennium forecast of the Eurodollar rates

Unrestricted model, IRGMM estimation



More details in:

- **Czellar, V., Karolyi, G.A. and Ronchetti, E.,(2007) 'Indirect Robust Estimation of the Short-term Interest Rate Process'**, *Journal of Empirical Finance*

Webpages:

http://www.unige.ch/ses/metri/ronchetti

https://studies2.hec.fr/jahia/Jahia/czellarv

Bootstrap methods

Conclusions

- Statistical methods are powerful tools for analysis and prediction.
- Interaction between theoretical (methodological) developments and computer simulation

Bootstrapping LS and Robust Estimators
======================================

Elvezio Ronchetti
Dept. of Econometrics
University of Geneva
Switzerland

Data
====

Job eval. points X1	Sex X2	Years at company X3	Years at job X4	Performance rating X5	Monthly salary $ Y
350	1	2	2	5	1000
350	1	5	5	5	1400
350	0	4	4	4	1200
350	1	20	20	1	1800
425	0	10	2	3	2800
425	1	15	10	3	4000
425	0	1	1	4	2500
425	1	5	5	4	3000
600	1	10	5	2	3500
600	0	4	3	4	2900
600	1	20	10	2	3800
600	1	7	7	5	4200
700	1	8	8	1	4600
700	0	25	15	5	5000
700	1	19	16	4	4600
700	0	20	14	5	4700
400	0	6	4	3	1800
400	1	20	8	3	3400
400	0	5	3	5	2000
500	1	22	12	3	3200
500	1	25	10	3	3200
500	0	8	3	4	2800
500	0	2	1	5	2400
800	1	10	10	3	5200
475	1	10	4	3	2400
475	0	3	3	4	2400
475	1	8	8	2	3000
475	1	6	6	4	2800
475	0	12	4	3	2500
475	0	4	2	5	2100

LS-Regression
=============

Residual Standard Error = 487.1608, Multiple R-Square = 0.8445
N = 30, F-statistic = 26.0594 on 5 and 24 df, p-value = 0

	coef	std.err	t.stat	p.value	
Intercept	-1218.8057	539.7278	-2.2582	0.0333	*
X1	7.2617	0.8108	8.9562	0.0000	*
X2	265.4467	213.1381	1.2454	0.2250	
X3	40.8374	21.7706	1.8758	0.0729	
X4	-8.5019	35.6586	-0.2384	0.8136	
X5	10.0480	86.5031	0.1162	0.9085	

Residual Standard Error = 552.472, Multiple R-Square = 0.7666
N = 30, F-statistic = 91.9726 on 1 and 28 df, p-value = 0

```
                coef  std.err  t.stat p.value
Intercept -1047.8282 434.6394 -2.4108  0.0227 *
       X1     8.0287   0.8372  9.5902  0.0000 *
```

Bootstrap LS
=============

```
f.BOOTREG
function(X, y, nrep)
{
        boot <- NULL
        p <- ncol(X) + 1
        ssize <- length(y)
        val <- matrix(0, ncol = p, nrow = nrep)
        u <- runif(nrep * ssize)
        ut <- ceiling(u * ssize)
        index <- matrix(ut, ncol = ssize, nrow = nrep, byrow =
                T)
        for(i in 1:nrep) {
                val[i,  ] <- lsfit(X[index[i,  ],  ], y[index[
                        i,  ]])$coef
                abline(val[i,  ])
        }
        boot$val <- val
        boot$mean <- rep(0, p)
        boot$sd <- boot$mean
        for(j in 1:p) {
                boot$mean[j] <- mean(val[, j])
                boot$sd[j] <- sqrt(var(val[, j]))
        }
        boot
}
```

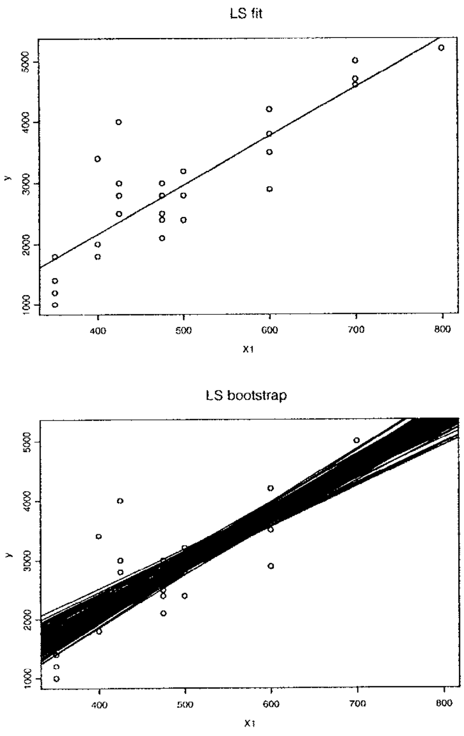
Bootstrap results (LS; 100 replicates)

```
       Intercept      X1
 [1,] -1518.88870 8.952058
 [2,] -1714.76868 8.919573
 [3,]  -646.59344 7.272577
 [4,]  -689.47536 7.344992
 [5,] -1451.59709 8.503735
 [6,]  -960.56338 7.941901
 [7,] -1141.10665 8.225868
 [8,]  -687.92021 7.647950
 [9,] -1624.50099 8.912042
[10,] -1305.06903 8.656805
[11,]  -751.07238 7.296650
[12,]  -867.87540 7.718666
[13,]  -949.53956 7.634279
[14,]   -86.79141 6.432581
[15,] -1707.75956 9.180328
[16,] -1133.94860 8.221963
[17,]  -609.01769 7.301208
[18,] -1332.00135 8.545208
[19,] -1001.40865 7.943847
[20,] -1312.22961 8.329319
[21,]  -906.06424 7.841177
[22,] -1042.50711 7.951600
```

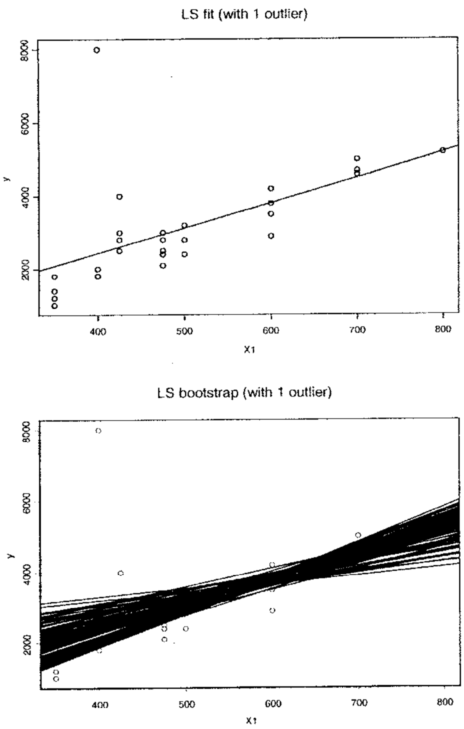
```
[23,]  -492.14135 7.210859
[24,] -1170.90451 8.193850
[25,] -1133.36194 8.013302
[26,] -1301.60690 8.650310
[27,] -1265.44237 8.341067
[28,] -1342.23035 8.518282
[29,] -1658.83709 9.101631
[30,]  -363.17759 6.816965
[31,] -1370.89371 8.417167
[32,]  -738.90475 7.515183
[33,]  -949.53431 7.883887
[34,] -1107.29730 8.182432
[35,]  -846.21212 7.523852
[36,] -1294.95504 8.141359
[37,]    37.99811 5.946354
[38,] -1835.58979 9.574969
[39,] -1606.59067 8.934301
[40,] -1323.56526 8.300660
[41,]  -981.62708 7.819259
[42,]  -732.21965 7.505127
[43,] -1277.40835 8.487504
[44,]  -998.82896 7.758694
[45,] -1074.10107 8.202138
[46,]  -916.24758 7.791103
[47,] -1498.09374 8.740386
[48,] -1461.58301 8.541506
[49,]  -935.62963 7.877584
[50,]  -501.11702 7.202128
[51,]  -214.84896 6.500562
[52,]  -887.53837 7.748829
[53,]  -808.13430 7.607392
[54,] -1457.05302 8.549938
[55,] -1047.37687 7.857602
[56,]  -710.77825 7.359698
[57,] -1219.73748 8.213006
[58,] -1561.40684 8.897338
[59,]    92.85535 6.266734
[60,] -1229.71126 8.282790
[61,] -1162.31337 8.221279
[62,] -1578.94565 9.075311
[63,] -1002.65544 7.971503
[64,] -1546.60032 8.738931
[65,]  -324.96818 6.637047
[66,]  -762.88258 7.685031
[67,] -1230.59909 8.221587
[68,] -1194.70320 8.390823
[69,]  -998.10198 7.977337
[70,] -1211.24762 8.364330
[71,] -1088.56738 8.016993
[72,]  -509.44772 7.169714
[73,] -1197.29730 8.247104
[74,] -1472.68216 8.707276
[75,] -1494.56091 8.883853
[76,] -1530.78567 9.052809
[77,]  -519.84944 7.316331
[78,]  -693.98088 7.569100
[79,] -1911.52566 9.414675
[80,] -1379.57980 8.434666
[81,] -1603.97066 8.894284
[82,] -1154.05455 8.161084
[83,] -1508.13222 8.808761
[84,] -1970.34601 9.666246
[85,]  -838.01821 7.653413
[86,]  -897.65934 7.768174
[87,] -1367.44678 8.591300
[88,]  -601.9835  7.158678
[89,] -1385.8122  8.430488
[90,] -1076.6677  8.079341
```

```
[91,] -1197.7181  8.306116
[92,] -1256.0069  8.514206
[93,]  -127.6532  6.511425
[94,] -1018.8638  7.991046
[95,]  -729.3370  7.708739
[96,] -1087.2939  8.145975
[97,] -1702.3258  9.070088
[98,]  -398.4102  6.856828
[99,]  -357.8578  7.169447
[100,]-1904.1339  9.378481


$mean -1076.19    8.06

$sd     440.74    0.74
```

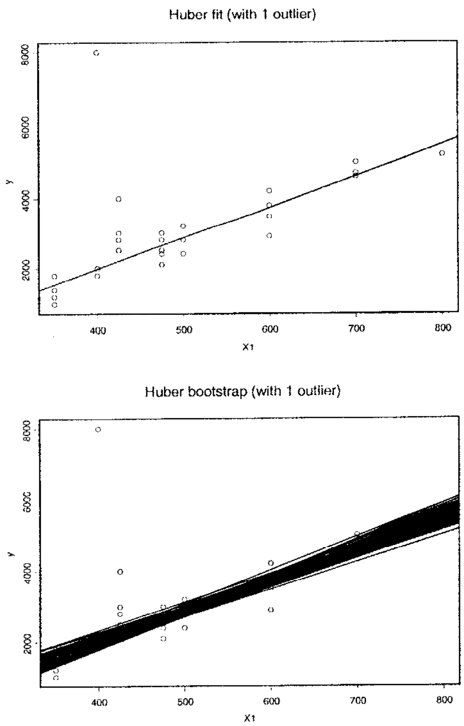
```
Data with 1 outlier
===================

Replace salaryy[18] (=3400) with value 8000

Estimate by LS and robust estimator (Huber estimator)


Bootstrap LS
------------

$mean  -401.23     7.00

$sd     962.74     1.58


Bootstrap Huber
---------------

$mean -1522.25     8.77

$sd     310.44     0.56
```


LS fit
LS bootstrap
y
X1
400
500
600
700
800

1


LS fit (with 1 outlier)
LS bootstrap (with 1 outlier)
y
X1
400
500
600
700
800

2


Huber fit (with 1 outlier)
Huber bootstrap (with 1 outlier)
y
X1
400
500
600
700
800

3