GRS261

FINAL PROJ

Time series sequence $\Rightarrow$ $\begin{cases} \text{Stationary.} \\ \gamma(h),\ \rho(h). \\ \text{Simultaneous test: Ljung-Box, in R: Box.test( ).} \end{cases}$

Ljung-Box test: $\underbrace{H_0: \rho_1 = \rho_2 = \cdots = \rho_K = 0}_{\text{stationary}}$ H_1: at least one not equal to 0

sample ACF plot

AR(1): stationary when $|\phi| < 1$

random walk when $|\phi| = 1$. (var increase when $t \uparrow$)

in R: arima( )

```
        Box-Ljung test

data:  bmw
X-squared = 44.987, df = 5, p-value = 1.460e-08
```

The p-value is very small, indicating that at least one of the first five autocorrelations is nonzero. Whether the amount of dependence is of any practical importance is debatable, but an AR(1) model to account for the small amount of autocorrelation might be appropriate.

Next, an AR(1) model was fit using the `arima()` command in R.

```
fitAR1 = arima(bmw, order = c(1,0,0))
print(fitAR1)
```

The `order` parameter will be explained later, but for an AR(1) process it should be `c(1,0,0)`. A summary of the output is below.

```
Call:
arima(x = bmw, order = c(1, 0, 0))

Coefficients:
           ar1  intercept
      0.081116   0.000340
s.e.  0.012722   0.000205

sigma^2 estimated as 0.000216260:  log-likelihood = 17212.34,
aic = -34418.68
```

We see that $\widehat{\phi} = 0.081$ and $\widehat{\sigma}^2 = 0.00022$. Although $\widehat{\phi}$ is small, it is statistically significant since it is 6.4 times its standard error 0.013, so its p-value is near zero. As just mentioned, whether this small, but nonzero, value of $\widehat{\phi}$ is of practical significance is another matter. A non-zero value of ϕ means that there is some information in today's return that could be used for prediction of tomorrow's return, but a small value of ϕ means that the prediction will not be very accurate. The potential for profit might be negated by trading costs.

Compute ACF for ARMA: in R, ARMAacf( )

Many statistical software packages have functions to automate the search for the AR model that optimizes AIC or other criteria. The `auto.arima` function in R's `forecast` package found that $p = 8$ is the first local minimum of AIC.

```
library(forecast)
auto.arima(diff(y), max.p = 20, max.q = 0, d = 0, ic = "aic")

  Series: diff(y)
  ARIMA(8,0,0) with zero mean
```

MA(q) model

ARMA(p, q) model: ARMA(1,1)

ARIMA(p, d, q): if $\Delta^d Y_t$ is ARMA(p, q)

if log Return ~ ARMA(p, q), then log Price ~ ARIMA(p, 1, q)

ARIMA(p, d, q) is stationary iff $d = 0$

Test stationary: unit root test: if all roots of $1-\phi_1 x-\phi_2 x^2-\cdots-\phi_p x^p=0$ has absolute value greater than 1. in R: polyroot( )

augmented Dickey-Fuller Test: H_0: There is a unit root. H_1: stationary (default) (<1)

in R: adf.test( )

explosive series (>1)

Phillips-Perron Test: similar. in R: pp.test( )

KPSS Test: H_0: stationary H_1: unitroot in R: kpss.test( )

```
library(tseries)
adf.test(y)

  Augmented Dickey-Fuller Test
  data:  y
  Dickey-Fuller = -3.8651, Lag order = 7, p-value = 0.01576
  alternative hypothesis: stationary

pp.test(y)

  Phillips-Perron Unit Root Test
  data:  y
  Dickey-Fuller Z(alpha) = -248.75, Truncation lag parameter = 5,
    p-value = 0.01
  alternative hypothesis: stationary
```

Select ARIMA model: in R: auto.arima( ) to select p, d, q

AIC/BIC KPSS

In this example, auto.arima() is applied to the inflation rates. The ARIMA (1,1,1) model is selected by auto.arima() using either AIC or BIC to select p and q after $d=1$ is selected by the KPSS test.

```
auto.arima(y, max.p = 5, max.q = 5, ic = "bic", trace = FALSE)

Series: y
ARIMA(1,1,1)

Coefficients:
         ar1      ma1
       0.238   -0.877
s.e.   0.055    0.027

sigma^2 estimated as 8.55:  log likelihood=-1221.6
AIC=2449.2   AICc=2449.3   BIC=2461.8
```

forcasting time series: in R: predict( )

prediction interval for ARIMA Process:

$$\left\{\phi^{2(k-1)} + \phi^{2(k-2)} + \cdots + \phi^2 + 1\right\}\sigma_\epsilon^2 = \left(\frac{1-\phi^{2k}}{1-\phi^2}\right)\sigma_\epsilon^2$$
$$\to \frac{\sigma_\epsilon^2}{1-\phi^2} \text{ as } k \to \infty. \quad (12.49)$$

An important point here is that the variance of the forecast error does not diverge as $k \to \infty$, but rather the variance converges to $\gamma(0)$, the marginal covariance of the AR(1) process given by (12.7). This is an example of the general principle that for any stationary ARMA process, the variance of the forecast error converges to the marginal variance.

For non stationary process, prediction error diverges as time lag $k \to \infty$.

Use simulation to compute forecast limits. ?

pacf: identify AR(p) order p. in R: pacf()

Seasonality removal: seasonal differencing $\Delta_s = 1 - B^s$

$$(1-\phi B)(\Delta Y_t - \mu) = \epsilon_t \quad (13.1)$$

and a purely seasonal ARIMA$(1,1,0)_s$ model

$$(1-\phi^* B^s)(\Delta_s Y_t - \mu) = \epsilon_t \quad (13.2)$$

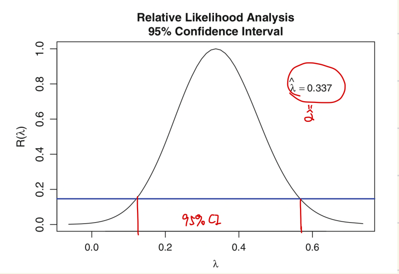
to obtain the multiplicative model

$$(1-\phi B)(1-\phi^* B^s)\{\Delta(\Delta_s Y_t) - \mu\} = \epsilon_t. \quad (13.3)$$

Model (13.2) is called "purely seasonal" and has a subscript "s" since it uses only B^s and Δ_s; it is obtained from the ARIMA(1,1,0) by replacing B and Δ by B^s and Δ_s. For a monthly time series ($s = 12$), model (13.2) gives 12 independent processes, one for Januaries, a second for Februaries, and so forth. Model (13.3) uses the components from (13.1) to tie these 12 series together.

※ Box-Cox transformation for $Y_t^{(\alpha)}$, "α"

in R, FitAR package, BoxCox, Arima()




```
data(Hstarts, package="Ecdat")
x = ts(Hstarts[,1], start=1960, frequency=4)
fit2 = arima(x, c(1,1,1), seasonal = list(order = c(0,1,1),
  period = 4))
fit2
```

→ create time series object; non seasonal model; seasonal model; S

```
Call:
arima(x = hst, order = c(1, 1, 1), seasonal
= list(order = c(0, 1, 1), period = 4))

Coefficients:
         ar1     ma1    sma1
       0.675  -0.890  -0.822
s.e.   0.142   0.105   0.051

sigma^2 estimated as 0.0261: log-likelihood = 62.9,
   aic = -118
```

⇒ The fitted model is:

$(1-0.675B)Y_t^* = (1-0.890B)(1-0.822B^4)\varepsilon_t$

where $Y_t^* = \Delta(\Delta^4 Y_t)$, $\varepsilon_t \sim wn(0, 0.0261)$

After fitting Linear Regression

☆ residual correlation: { adjust the correlation matrix
use ARMA model on noise instead of white noise

spurious regression: Because of the test is based on incorrect assumptions of independent error
: provides misleading statistical evidence of a linear relationship between independent non-stationary variables

Test correlation: Durbin-Watson Test, in R: durbinWatsonTest() in car package
or lmtest package, dwtest()

we make this comparison? Suppose that X_t and Y_t are time series following the regression model

$$Y_t = \alpha + \beta_0 t + \beta_1 X_i + \epsilon_t. \tag{13.6}$$

Note the linear time trend $\beta_0 t$. Then, upon differencing, we have

$$\Delta Y_t = \beta_0 + \beta_1 \Delta X_i + \Delta\epsilon_t, \tag{13.7}$$

so the original intercept α is removed, and the time trend's slope β_0 in (13.6) becomes an intercept in (13.7). The time trend could be omitted in (13.6) if the intercept in (13.7) is not significant, as happens in this example. The slope β_1 in (13.6) remains unchanged in (13.7). However, if ϵ_t is $I(1)$, then the regression of Y_t on X_t will not provide a consistent estimate of β_1, but the regression of ΔY_t on ΔX_i will consistently estimate β_1, so the estimates

When dependence between ϵ_t is not strong, we still have methods to estimate consistent standard errors for coefficient estimates: HC & HAC

in R: NeweyWest( ) in package sandwich

The HC and HAC covariance matrix estimates can be computed using the `NeweyWest()` function from the `R` package `sandwich`. The first argument is a fitted model object, in this case `fit`. In both cases we set `prewhite = F`. For the HAC estimate, the argument `lag` corresponds to the maximal lag L used in the Bartlett weight function above. If no value is specified, one is selected automatically via the `bwNeweyWest()` function (see the help file for more information). For the HC estimate we specify `lag = 0`. The HC estimate and HAC estimate with $L = 3$ are shown below.

```
library(sandwich)
options(digits=2)
NeweyWest(fit, lag = 0, prewhite = F)
```

lm object

```
            (Intercept) cm10_dif cm30_dif
(Intercept)     4.7e-06  7.3e-06 -1.1e-05
cm10_dif        7.3e-06  6.3e-03 -6.2e-03
cm30_dif       -1.1e-05 -6.2e-03  6.7e-03
```

```
NeweyWest(fit, lag = 3, prewhite = F)
```

```
            (Intercept) cm10_dif cm30_dif
(Intercept)     4.6e-06 -0.00003  2.6e-05
cm10_dif       -3.0e-05  0.00666 -6.6e-03
cm30_dif        2.6e-05 -0.00662  7.0e-03
```

☆ Linear Regression + ARMA model

The linear regression model with ARMA errors combines the linear regression model (9.1) and the ARMA model (12.26) for the noise, so that

$$Y_t = \beta_0 + \beta_1 X_{t,1} + \cdots + \beta_p X_{t,p} + \epsilon_t, \tag{13.13}$$

where

$$(1 - \phi_1 B - \cdots - \phi_p B^p)\,\epsilon_t = (1 + \theta_1 B + \cdots + \theta_q B^q)\,u_t, \tag{13.14}$$

and $u_1, \ldots, u_n$ is white noise.

multivariate time series: $(Y_{1,t}, \ldots, Y_{d,t})$

CCF $= \rho(Y_j, Y_i)(h) = \mathrm{corr}(Y_j(t), Y_i(t-h)) \Rightarrow \rho_{Y_j,Y_i}(h) = \rho_{Y_i,Y_j}(-h)$, in R. ccf()

weakly stationary: μ, Σ finite and not depend on t.

cross correlation only show statistical association, not causation. 因为两个variable之间还未建立联系(ex. by regression)

in R: acf(n×d matrix) can give you d sample ACF plot & $\frac{d(d-1)}{2}$ sample ccf plot

The following commands will conduct the multivariate Ljung–Box test in R for the bivariate series $(\Delta cpi, \Delta ip)'$.

```
source("SDAFE2.R")
mLjungBox(CPI_IP, lag = 10)
```

```
     K   Q(K) d.f. p-value
  1 10 532.48   40       0
```

The multivariate Ljung–Box test statistic was 532.48, and the approximate p-value was 0, confirming that there is significant serial correlation in the first $K = 10$ lags.

in R: ar( ) can fit bivariate AR model

acf( ) create matrix of plot

Bootstrap resampling.

① model based ✓

② model free: block resampling / block bootstrap, in R: tsboot()

function in R's boot package. The idea is to break the time series into roughly equal-length blocks of consecutive observations, to resample the blocks with replacement, and then to paste the blocks together. For example, if the time series is of length 200 and one uses 10 blocks of length 20, then the blocks are the first 20 observations, the next 20, and so forth. A possible resample is the fourth block (observations 61 to 80), then the last block (observations 181 to 200), then the second block (observations 21 to 40), then the fourth block again, and so on until there are 10 blocks in the resample. 可以让block重复

GARCH MODEL: allow heteroskedasticity, non constant variance

$$Y_t = f(X_{1,t}, \ldots, X_{p,t}) + \varepsilon_t\, \sigma(X_{1,t}, \ldots, X_{p,t})$$

ARCH(1): $a_t = \varepsilon_t \sqrt{w + \alpha a_{t-1}^2}$, $w > 0$, $0 \le \alpha < 1$, for stationary and finite variance

GARCH(p,q), similar with ARMA(p,q)

GARCH(p, q) model is

$$a_t = \sigma_t \epsilon_t,$$

in which

$$\sigma_t = \sqrt{\omega + \sum_{i=1}^{p} \alpha_i a_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2}. \qquad (14.8)$$

AR(1) + GARCH(1,1) ☆ 常用

calculations, we obtain a necessary condition for a_t to be stationary:

$$\sum_{i=1}^{\max(p,q)} (\alpha_i + \beta_i) < 1. \qquad (14.15)$$

14.10 GARCH(1,1) Processes

The GARCH(1,1) is the most widely used GARCH process, so it is worthwhile to study it in some detail. If a_t is GARCH(1,1), then as we have just seen, a_t^2 is ARMA(1,1). Therefore, the ACF of a_t^2 can be obtained from formulas (12.31) and (12.32). After some algebra, one finds that

$$\rho_{a^2}(1) = \frac{\alpha(1 - \alpha\beta - \beta^2)}{1 - 2\alpha\beta - \beta^2} \qquad (14.16)$$

and

$$\rho_{a^2}(h) = (\alpha + \beta)^{h-1} \rho_{a^2}(1), \quad h \ge 2. \qquad (14.17)$$

This example uses the daily BMW stock log returns. The `ugarchfit()` function from R's `rugarch` package is used to fit an AR(1)+GARCH(1,1) model to this series. Although `ugarchfit()` allows the white noise to have a nonGaussian distribution, we begin this example using Gaussian white noise (the default). First the model is specified using the `ugarchspec()` function; for an AR(1)+GARCH(1,1) model we specify `armaOrder=c(1,0)` and `garchOrder=c(1,1)`. The commands and abbreviated output are below.

```
library(rugarch)
data(bmw, package="evir")
arma.garch.norm = ugarchspec(mean.model=list(armaOrder=c(1,0)),
                             variance.model=list(garchOrder=c(1,1)))
bmw.garch.norm = ugarchfit(data=bmw, spec=arma.garch.norm)
show(bmw.garch.norm)
```

```
GARCH Model : sGARCH(1,1)
Mean Model : ARFIMA(1,0,0)
Distribution : norm

Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000453    0.000175   2.5938 0.009493
ar1     0.098135    0.014261   6.8813 0.000000
omega   0.000009    0.000000  23.0613 0.000000
alpha1  0.099399    0.005593  17.7730 0.000000
beta1   0.863672    0.006283 137.4591 0.000000
```

$\hat{\mu}$ $\hat{\phi}_1$ $\hat{\omega}$

normal ✗

fit t-distribution to noise:

in R: fitdistr(res, "t")

The MLE of the degrees-of-freedom parameter was 4.1. This confirms the good fit by this distribution seen in Fig. 14.5. The AR(1)+GARCH(1,1) model was refit assuming t-distributed errors, so `distribution.model = "std"` in `ugarchspec()`. The commands and abbreviated results are below.

```
arma.garch.t = ugarchspec(mean.model=list(armaOrder=c(1,0)),
                          variance.model=list(garchOrder=c(1,1)),
                          distribution.model = "std")
bmw.garch.t = ugarchfit(data=bmw,spec=arma.garch.t)
show(bmw.garch.t)
```

APARCH Model: negative return → higher volatility (than that of positive return with same magnitude)

In this example, an AR(1)+APARCH(1,1) model with t-distributed errors is fit to the BMW log returns. The commands and abbreviated output from `ugarchfit()` is below. The estimate of δ is 1.48 with a standard error of 0.14, so there is strong evidence that δ is not 2, the value under a standard GARCH model. Also, $\widehat{\gamma}_1$ is 0.12 with a standard error of 0.045, so there is a statistically significant leverage effect, since we reject the null hypothesis that $\gamma_1 = 0$. However, the leverage effect is small, as can be seen in the plot in Fig. 14.8 with $\gamma = 0.12$. The leverage might not be of practical importance.

```
arma.aparch.t = ugarchspec(mean.model=list(armaOrder=c(1,0)),
                variance.model=list(model="apARCH",
                                    garchOrder=c(1,1)),
                                    distribution.model = "std")
bmw.aparch.t = ugarchfit(data=bmw, spec=arma.aparch.t)
show(bmw.aparch.t)
```

```
GARCH Model : apARCH(1,1)
Mean Model : ARFIMA(1,0,0)
Distribution : std
Optimal Parameters
------------------------------------
        Estimate  Std. Error  t value Pr(>|t|)
mu      0.000048    0.000147   0.3255 0.744801
ar1     0.063666    0.012352   5.1543 0.000000
omega   0.000050    0.000032   1.5541 0.120158
alpha1  0.098839    0.012741   7.7574 0.000000
beta1   0.899506    0.013565  66.3105 0.000000
gamma1  0.121947    0.044664   2.7303 0.006327
delta   1.476643    0.142442  10.3666 0.000000
shape   4.073809    0.234417  17.3784 0.000000

LogLikelihood : 18161

Information Criteria
------------------------------------
Akaike       -5.9073
Bayes        -5.8985
Shibata      -5.9073
Hannan-Quinn -5.9042
```

(normalized)

☆ factors to choose

interest rate. CPI. GDP. unemployment.

$$\boxed{Y_t = \beta_0 + \beta_1 X_{1t} + \cdots + \beta_p X_{pt} + \boxed{\varepsilon_t}}$$ $\varepsilon_t \sim N(0, \sigma^2)$

~~lm (Y — + ε_t)~~

time series lm (Y — $\beta_p X_{pt}$) ε_t

问题: fit + predict stock price & return 时, 如果用 time series model, 什么时候用纯 time series? 什么时候用 linear regression + time series?
(只有时间 t) (有别的 factor?)

只有 t 也可以用 linear regression ($x = t$). lm($Y_t \sim t$)
$= (1, 2, 3, \cdots)$

如果手上有一段 period 的 data, 先 fit linear regression, check residual? 还是先设 time series object, 再 auto fit timeseries formula?

PCA + copula : risk management.
time series : price / return prediction.
} How to combine?

ARMA : constant variance, stationary, mean not constant.
ARIMA: it's difference is ARMA, non stationary
⇒ residual correlation : HC & HAC (neweywest())
⇒ not normal : t dist.
ARCH: variance not constant, mean constant = 0 (like AR)
GARCH: except for autocorrelation, also include σ term (like ARMA)
APARCH: capture negative and positive a_t's difference in volatility.
O-GARCH: orthogonal GARCH model, reduce number of parameters
DOC: Dynamic Orthogonal Component
DCC: Dynamic Conditional Correlation
EWMA: Exponentially Weighted Moving Average

factanal function P545.

prediction interval
AR(1) : predictors → ~~confidence~~ prediction interval → constant variance
GARCH(1,1): update variance
VaR: for β大, β小, compare.
different strategies

Chapter 4: 洗数据: Kernel, 画图, diagnostic, data description 学妹

Chapter 6: bootstrap. 4个人

Chapter 8: 如何用copula做risk management. 悦芝 Yvonne

可以考虑用Gaussian copula

Chapter 12-14. Time series model Cheng

Chapter 18: factor model, (1-7)

cross-sectional factor models: 要用times + regression

具体什么factor影响

regress R_{jt} on $\beta_0 + \beta_1 X_{1t} + \cdots + \beta_p X_{pt} + \beta_{p+1} t + \varepsilon_t$

j-th asset at time t.

in R: factanal( ) 马雨诺 最后三页

Chapter 19: risk management. Cheng 悦芝 Yvonne 马雨诺

non-parametric: $\mathrm{VaR}(\alpha) = -S \times \hat{q}(\alpha)$, S is the size of current position, $\hat{q}(\alpha)$ is the sample quantile of return

To estimate ES, let $R_1, \ldots, R_n$ be the historic returns and define $\mathcal{L}_i = -S \times R_i$. Then

expect会lose多少钱, 在1-α confidence内

$$\widehat{\mathrm{ES}}^{\mathrm{np}}(\alpha) = \frac{\sum_{i=1}^n \mathcal{L}_i I\{\mathcal{L}_i > \widehat{\mathrm{VaR}}(\alpha)\}}{\sum_{i=1}^n I\{\mathcal{L}_i > \widehat{\mathrm{VaR}}(\alpha)\}} = -S \times \frac{\sum_{i=1}^n R_i I\{R_i < \widehat{q}(\alpha)\}}{\sum_{i=1}^n I\{R_i < \widehat{q}(\alpha)\}}, \quad (19.5)$$

which is the average of all $\mathcal{L}_i$ exceeding $\widehat{\mathrm{VaR}}^{\mathrm{np}}(\alpha)$. Here $I\{\mathcal{L}_i > \widehat{\mathrm{VaR}}^{\mathrm{np}}(\alpha)\}$ is the indicator that $\mathcal{L}_i$ exceeds $\widehat{\mathrm{VaR}}^{\mathrm{np}}(\alpha)$, and similarly for $I\{R_i < \widehat{q}(\alpha)\}$.

```
data(SP500, package="Ecdat")
n = 2783
SPreturn = SP500$r500[(n - 999):n]
year = 1981 + (1:n) * (1991.25 - 1981) / n
year = year[(n - 999):n]
alpha = 0.05
q = as.numeric(quantile(SPreturn, alpha))
VaR_nonp = -20000 * q
IEVaR = (SPreturn < q)
sum(IEVaR)
ES_nonp = -20000 * sum(SPreturn * IEVaR) / sum(IEVaR)
options(digits = 5)
VaR_nonp
ES_nonp
```

unit \$

parametric: $\mathrm{VaR}(\alpha) = -S \times F^{-1}(\alpha \mid \hat{\theta})$

$\widehat{\mathrm{ES}}(\alpha) = -\frac{S}{\alpha} \times \int_{-\infty}^{F^{-1}(\alpha)} x f(x \mid \hat{\theta})\, dx$

Suppose the return has a t-distribution with mean equal to μ, scale parameter equal to λ, and tail index[1] ν. Let f_ν and F_ν be, respectively, the t-density and t-distribution function with ν degrees of freedom. The expected shortfall is

$$\widehat{\mathrm{ES}}^{\mathrm{t}}(\alpha) = S \times \left\{ -\mu + \lambda \left(\frac{f_\nu\{F_\nu^{-1}(\alpha)\}}{\alpha} \left[\frac{\nu + \{F_\nu^{-1}(\alpha)\}^2}{\nu - 1} \right] \right) \right\}. \quad (19.8)$$

The formula for normal loss distributions is obtained by a direct calculation or letting $\nu \to \infty$ in (19.8). The result is

$$\mathrm{ES}^{\mathrm{norm}}(\alpha) = S \times \left\{ -\mu + \sigma \left(\frac{\phi\{\Phi^{-1}(\alpha)\}}{\alpha} \right) \right\}, \quad (19.9)$$

where μ and σ are the mean and standard deviation of the returns and ϕ and Φ are the standard normal density and CDF. The superscripts "t" and

$\widehat{\mathrm{VaR}}^{t}(\alpha) = -S \times \{\hat{\mu} + \hat{\lambda} \times F_\nu^{-1}(\alpha)\}$

$\widehat{\mathrm{VaR}}^{\mathrm{norm}}(\alpha) = -S \times \{\hat{\mu} + \hat{\sigma} \times \Phi^{-1}(\alpha)\}$

$\hat{\lambda} = \frac{\nu - 2}{\nu}$

```r
data(SP500, package="Ecdat")
n = 2783
SPreturn = SP500$r500[(n - 999):n]
year = 1981 + (1:n) * (1991.25 - 1981) / n
year = year[(n - 999):n]
alpha = 0.05
library(MASS)
fitt = fitdistr(SPreturn, "t")
param = as.numeric(fitt$estimate)
mean = param[1]
df = param[3]
sd = param[2] * sqrt((df) / (df - 2))
lambda = param[2]
qalpha = qt(alpha, df = df)
VaR_par = -20000 * (mean + lambda * qalpha)
es1 = dt(qalpha, df = df) / (alpha)
es2 = (df + qalpha^2) / (df - 1)
es3 = -mean + lambda * es1 * es2
ES_par = 20000*es3
VaR_par
ES_par
```

Use ARMA + GARCH Model for VaR and ES

Assume we have $R_1, \cdots, R_n$ historical returns, we need to estimate $\boxed{\text{VaR \& ES}}$ for the next return R_{n+1}. $\boxed{19.2.2}$

```r
library(rugarch)
garch.t = ugarchspec(mean.model=list(armaOrder=c(0,0)),
                     variance.model=list(garchOrder=c(1,1)),
                     distribution.model="std")
sp.garch.t = ugarchfit(data=SPreturn, spec=garch.t)
show(sp.garch.t)

 Optimal Parameters
 ------------------------------------
         Estimate  Std. Error   t value Pr(>|t|)
 mu      0.000714    0.000264   2.70872 0.006754
 omega   0.000003    0.000004   0.79083 0.429046
 alpha1  0.032459    0.019439   1.66979 0.094961
 beta1   0.939176    0.009296 101.02598 0.000000
 shape   4.417464    0.560553   7.88054 0.000000

pred = ugarchforecast(sp.garch.t, data=SPreturn, n.ahead=1) ; pred

        Series    Sigma
 T+1 0.0007144 0.009478

alpha = 0.05
nu = as.numeric(coef(sp.garch.t)[5])
q = qstd(alpha, mean=fitted(pred), sd=sigma(pred), nu=nu)
VaR = -20000*q ; VaR

 T+1              276.7298

lambda = sigma(pred)/sqrt( (nu)/(nu-2) )
qalpha = qt(alpha, df=nu)
es1 = dt(qalpha, df=nu)/(alpha)
es2 = (nu + qalpha^2) / (nu - 1)
es3 = -mean + lambda*es1*es2
ES_par = 20000*es3 ; ES_par

 T+1              413.6518
```

prediction

shape parameter

→ how to specify this, whether std or other?

data → fit different distributions, look at the performance

how do we know where should we specify t or norm?

when to use t / norm → distribution of ε_t, not a_t itself! so generally specify norm.

likelihood. VaR.

P562

first normal, then t

※ Prefer ES better than VaR, since VaR is not subadditive.

问题: fit ARMA+GARCH model, in R, how do we know which ARMA model we use? in ugarchspec function, should we first fit auto.arima function? to find what are p, d, q?

用 AIC/BIC, for loop.

Chapter 4.

histogram plot ✗

kernel density 比较好，原因：large sample size 不容易看出来 outliers.

KDE: b = bandwidth, $\sigma = b$.

sample size = 6，有6条 dashed curve，一条 kernel density (由其他合

b↓, kernel estimator var↑, bias↓.

in R: density (adjust =) ,default is the best.
(b)

用MAD: 排除 outlier 影响，用 sd(), mad().

normal probability plot. : data 在 x 轴
theoretical quantile: 在 y 轴

halfnorm ( ) : to see whether absolute deviation of $|Y_i - \bar{Y}|$ are unusual.
看是不是 outliers..

normal test: Shapiro-Wilk test, most powerful for asymmetric

boxplot:

~~transformation~~: 让 variance 变得 stable. constant. (log)
要用 transformation 可以变得 constant，就可以直接用SLR，...
还是用 Garch model 不管 variance 是否 constant.
concave transformation: remove right skewness

QQ - plot. normality → GARCH model.

Stock returns uncorrelated, so go with GARCH & ARCH.
volatility.

grounded in the fact that in practice, an investment manager will often have a predefined set of characteristics that he uses to compare stocks, for instance company size, balance sheet variables, credit ratings, sector, momentum, market vs book value, return on assets, management team, online sentiment, technical indicators, 'beta', etc. The investment manager will typically choose trading characteristics so that they are informative enough to unveil market ineffi-

compare different strategies. (portfolios).

① $\frac{1}{n}$

② Bonds + stocks $\Rightarrow$ do VaR.

different combinations here.

assumption: normal / t in ARCH model.

~~copula~~:

PE ratio : 一年 data

copula, estimate joint distribution, 然后 simulate returns

estimate $r_p = \sum w_i x_i$

用 copula : $\hat{F}(x_1, \dots, x_p)$. $\longrightarrow$ estimate portfolio returns.

By simulation,

Markowitz modern portfolio theory. (MVP, CAPM).

diversity weighted portfolios based on stochastic portfolio theory (Functionally Generated portfolios).

constant correlation model

multi-factor model

equal allocation model : $\frac{1}{n}$, 比较好可以分散 risk.

sharpe portfolio selection model. (single index model).

Two similar models were used for the empirical tests performed in the study. For identifying real estate or common stock efficient frontiers, the Sharpe [11] diagonal model was used. To identify portfolios consisting of both assets, the Cohen-Pogue [1] multi-index model was used.

When Markowitz [7] introduced his work, a primary hypothesis was that an investor wanted to maximize return for a given level of risk or minimize risk for a given level of return. This means that at each level of return or risk only one set of stocks would satisfy the constraints. These portfolios are denoted to be "efficient." At any level of return there is only one efficient portfolio and at any level of risk there is only one efficient portfolio. It should be noted that one efficient portfolio is not clearly better than any other efficient portfolio. The investor's risk-return preference function determines which portfolio is selected.

other things also show this : bagging method decrease variance ...

(1) $R_i = A_i + B_i I + C_i \quad i=1,\ldots,N$

where

(2) R_i = return on asset i

(2') I = random variable denoting the level of an index

$= A_{N+1} + C_{N+1}$ and

(2") $E(I) = A_{N+1}$

(2"') $E(C_{N+1}) = 0$

(2"") $\operatorname{Var}(I) = E(C^2_{N+1}) = Q_{N+1}$

With assumptions concerning the covariance of the returns, the calculations are simplified with the following results for portfolio return (P).

(3) $E(P) = \sum_{i=1}^{N+1} X_i A_i = A'X$

(4) $\operatorname{Var}(P) = \sum_{i=1}^{N+1} X_i^2 Q_i = X' O_{N+1} X$

where

(4') X, A, O_{N+1} have N+1 elements and

(4") O_{N+1} is a diagonal covariance matrix

where

X_i = proportion of wealth invested in security i

The full model is:

(5) Maximize $\lambda A'X - X' O_{N+1} X$ for all λ.

Subject to:

(6) $X_{N+1} = \sum_{i=1}^{N} X_i B_i$

(6") $\sum_{i=1}^{N} X_i = 1$

(6"') $X_i \geq 0$, for all i, i=1,....,N

where

λ is the "shadow price of risk.

ARCH:

① predict future volatility.

$\hat{X}_t^2 = \hat{\alpha}_0 + \hat{\alpha}_1 \hat{X}_{t-1}^2$

X_t^2 : the realization

compare them

⇒ risk management

calculate value at risk

use smoothing, ... to model volatility

Title:

portfolio management on ___ , ___ , ___ (theory)

Gaussian Graphic Model: precision matrix.

$$\begin{pmatrix}x_1\\ \vdots\\ x_p\end{pmatrix} \sim N(0,\Sigma) \Rightarrow \frac{1}{(\sqrt{2\pi})^p}\,\frac{1}{|\Sigma|^{\frac{1}{2}}}\, e^{-\frac{x^T\Sigma^{-1}x}{2}} = f_1(x_1)\, f_2(x_2)$$

$$X_1 = \begin{pmatrix}x_1\\ \vdots\\ x_{p_1}\end{pmatrix} \quad X_2 = \begin{pmatrix}x_{p_1+1}\\ \vdots\\ x_p\end{pmatrix}, \quad X_1 \perp\!\!\!\perp X_2, \quad \Sigma^{-1} = \begin{pmatrix}A_1 & 0\\ 0 & A_2\end{pmatrix}$$

Now, there is no direct link between X_i & X_j,
$$\begin{tikzcd} X_i \arrow[rrr, dash, "\times"] \arrow[dr] & & & X_j \\ & X_k \arrow[r, leftrightarrow] & X_r \arrow[ur, leftrightarrow] & \end{tikzcd}$$
, but there is undirect correlation through X_k, X_r, etc.

How to use it? $\begin{pmatrix}x_1\\ \vdots\\ x_p\end{pmatrix}$ — returns of p stocks. Estimate Σ^{-1}.

Add penalty to generate a sparse one. $\Rightarrow$ risk management,
calculate VaR
stock selection for portfolios.

It is related to AR(1) model: X_{t-2} affect X_t through X_{t-1}.

$$\begin{tikzcd} X_t & X_{t-1} \arrow[l] & X_{t-2} \arrow[l] \arrow[ll, dashed, bend left] \end{tikzcd}$$

AR(2) model:
$$\begin{tikzcd} & X_{t-1} \arrow[dl] & X_{t-3} \arrow[l] \arrow[dl] \\ X_t & & \\ & X_{t-2} \arrow[ul] \arrow[uu] & X_{t-4} \arrow[l] \end{tikzcd}$$

①Efficient frontier: μ_{ft}, R_{it}, ---,

Σ — true, known.

simulation: $\underbrace{\hat{\Sigma}_1, \hat{\Sigma}_2, \cdots}_{\text{very different.}}$ $\hat{\Sigma}, \Sigma$ close, but $\hat{\Sigma}^{-1}$ & Σ^{-1} not.

use strategies to compare with index stock (S&P 500).

② volatility. option pricing

- Markowitz. CAPM
- ARCH / GARCH model: models $\hat{\sigma}_t^2$, do prediction returns.

$X_t = \sigma_t \varepsilon_t$, get correct prediction interval.

to say the realization fall inside / outside.

different models. compare different strategies.

③ VaR: risk management.

level α. ? #{loss > VaR} / # days.

④ macro data: inflation. CPI.

unemployment

international market

加 8 个 covariant

问题: PE ratio multifactor model to predict return,

How to combine them into one single factor for 8 stocks? Similarly, how about β?

Bootstrap conf. interval for VaR & ES

Copula for VaR & ES

ts 结果超差, 为什么? 不能用吗??

historical data compare VaR & ES,

predict VaR at each day.

下周：写 report & presentation

① 储存每天的 weight
② 研究好 bcrp ...
③ VaR, ES: copula, timeseries "backtesting"
(difficult)

14.12 Linear Regression with ARMA+GARCH Errors

When using time series regression, one often observes autocorrelated residuals. For this reason, linear regression with ARMA disturbances was introduced in Sect. 13.3.3. The model considered was

$$Y_t = \beta_0 + \beta_1 X_{t,1} + \cdots + \beta_p X_{t,p} + e_t, \tag{14.19}$$

where

$$(1 - \phi_1 B - \cdots - \phi_p B^p)(e_t - \mu) = (1 + \theta_1 B + \ldots + \theta_q B^q) a_t, \tag{14.20}$$

and $\{a_t\}$ is i.i.d. white noise. This model is sufficient for serially correlated errors, but it does not accommodate volatility clustering, which is often found in the residuals.

One solution is to model the noise as an ARMA+GARCH process. Therefore, we will now assume that, instead of being i.i.d. white noise, $\{a_t\}$ is a GARCH process so that

$$a_t = \sigma_t \epsilon_t, \tag{14.21}$$

where

$$\sigma_t = \sqrt{\omega + \sum_{i=1}^{p} \alpha_i a_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2}, \tag{14.22}$$

and $\{\epsilon_t\}$ is i.i.d. white noise. The model given by (14.19)–(14.22) is a *linear regression model with ARMA+GARCH disturbances.*

Some software, including the `ugarchfit()` function from `R`'s `rugarch` package, can fit the linear regression model with ARMA+GARCH disturbances in one step. Another solution is to adjust or correct the estimated covariance matrix of the regression coefficients, via the HAC estimator from Sect. 13.3.2, by using the `NeweyWest()` function from the `R` package `sandwich`. However, if such software is not available, then a three-step estimation method is the following:

1. estimate the parameters in (14.19) by ordinary least-squares;
2. fit model (14.20)–(14.22) to the ordinary least-squares residuals;
3. reestimate the parameters in (14.19) by weighted least-squares with weights equal to the reciprocals of the conditional variances from step 2.

Example 14.4. Regression analysis with ARMA+GARCH errors of the Nelson–Plosser data

In Example 9.9, we saw that a parsimonious model for the yearly log returns on the stock index `diff(log(sp))` used `diff(log(ip))` and `diff(bnd)` as predictors. Figure 14.9 contains ACF plots of the residuals [panel (a)] and squared residuals [panel (b)]. Externally studentized residuals were used, but the plots for the raw residuals are similar. There is some autocorrelation in both the residuals and squared residuals.

```
nelsonplosser = read.csv("nelsonplosser.csv", header = TRUE)
new_np = na.omit(nelsonplosser)
attach(new_np)
fit.lm1 = lm(diff(log(sp)) ~ diff(log(ip)) + diff(bnd))
summary(fit.lm1)
```

先fit linear regression

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.01657    0.02100   0.789 0.433316
diff(log(ip))   0.69748    0.16834   4.143 0.000113 ***
diff(bnd)      -0.13224    0.06225  -2.124 0.037920 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1509 on 58 degrees of freedom
Multiple R-squared:  0.3087,Adjusted R-squared:  0.2848
F-statistic: 12.95 on 2 and 58 DF,  p-value: 2.244e-05
```

The `auto.arima()` function from R's `forecast` package selected an MA(1) model [i.e., ARIMA(0,0,1)] for the residuals. Next an MA(1)+ARCH(1) model was fit to the regression model's raw residuals. Sample ACF plots of the standardized residuals from the MA(1)+ARCH(1) model are in Fig. 14.9c and d. One sees essentially no short-term autocorrelation in the ARMA+GARCH standardized or squared standardized residuals, which indicates that the ARMA+GARCH model accounts for the observed dependence in the regression residuals satisfactorily. A normal plot showed that the standardized residuals are close to normally distributed, which is not unexpected for yearly log returns.

Finally, the linear model was refit with the reciprocals of the conditional variances as weights. The estimated regression coefficients are given below along with their standard errors and p-values.

```
fit.lm3 = lm(diff(log(sp)) ~ diff(log(ip)) + diff(bnd),
             weights = 1/sigma.arch^2)
summary(fit.lm3)
```

weighted least square

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.03216    0.02052   1.567  0.12263
diff(log(ip))   0.55464    0.16942   3.274  0.00181 **
diff(bnd)      -0.12215    0.05827  -2.096  0.04051 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1.071 on 57 degrees of freedom
Multiple R-squared:  0.2416,Adjusted R-squared:  0.2149
F-statistic: 9.077 on 2 and 57 DF,  p-value: 0.0003783
```

There are no striking differences between these results and the unweighted fit in Example 9.9. In this situation, the main reason for using the GARCH

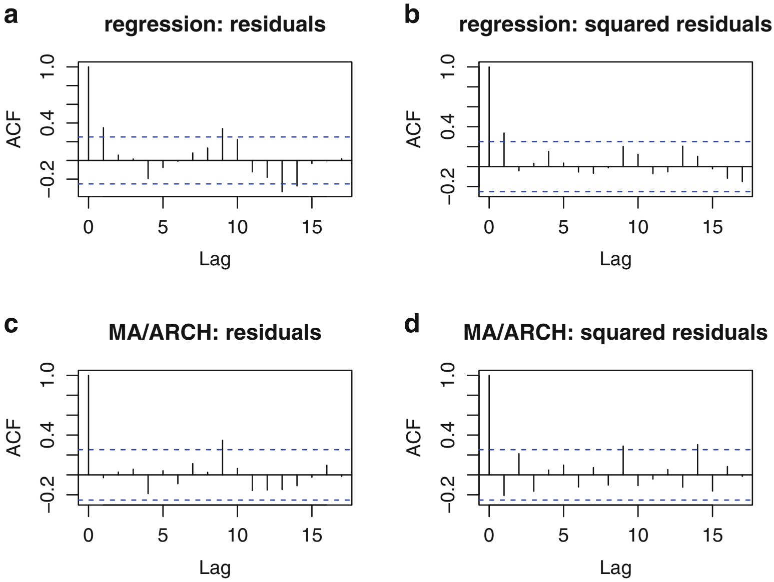


Fig. 14.9. *(a) Sample ACF of the externally studentized residuals and (b) their squared values, from a linear model; (c) Sample ACF of the standardized residuals and (d) their squared values, from an MA(1)+ARCH(1) fit to the regression residuals.*

model for the residuals would be in providing more accurate prediction intervals if the model were to be used for forecasting; see Sect. 14.13. □

14.13 Forecasting ARMA+GARCH Processes

Forecasting ARMA+GARCH processes is in one way similar to forecasting ARMA processes—point estimates, e.g., forecasts of the conditional mean, are the same because a GARCH process is weak white noise. What differs between forecasting ARMA+GARCH and ARMA processes is the behavior of the prediction intervals. In times of high volatility, prediction intervals using an ARMA+GARCH model will widen to take into account the higher amount of uncertainty. Similarly, the prediction intervals will narrow in times of lower volatility. Prediction intervals using an ARMA model without conditional heteroskedasticity cannot adapt in this way.

To illustrate, we will compare the prediction of a Gaussian white noise process and the prediction of a GARCH(1,1) process with Gaussian innovations.

Both have an ARMA(0,0) model for the conditional mean so their forecasts are equal to the marginal mean, which will be called μ. For Gaussian white noise, the prediction limits are $\mu \pm z_{\alpha/2}\sigma$, where σ is the marginal standard deviation. For a GARCH(1,1) process $\{Y_t\}$, the prediction limits at time origin n for h-steps ahead forecasting are $\mu \pm z_{\alpha/2}\sigma_{n+h|n}$ where $\sigma_{n+h|n}$ is the conditional standard deviation of Y_{n+h} given the information available at time n. As h increases, $\sigma_{n+h|n}$ converges to σ, so for long lead times the prediction intervals for the two models are similar. For shorter lead times, however, the prediction limits can be quite different.

Example 14.5. Forecasting BMW log returns

In this example, we will return to the daily BMW stock log returns used in several earlier examples. We have seen in Example 14.2 that an AR(1)+GARCH(1,1) model fits the returns well. Also, the estimated AR(1) coefficient is small, less than 0.1. Therefore, it is reasonable to use a GARCH (1,1) model for forecasting.

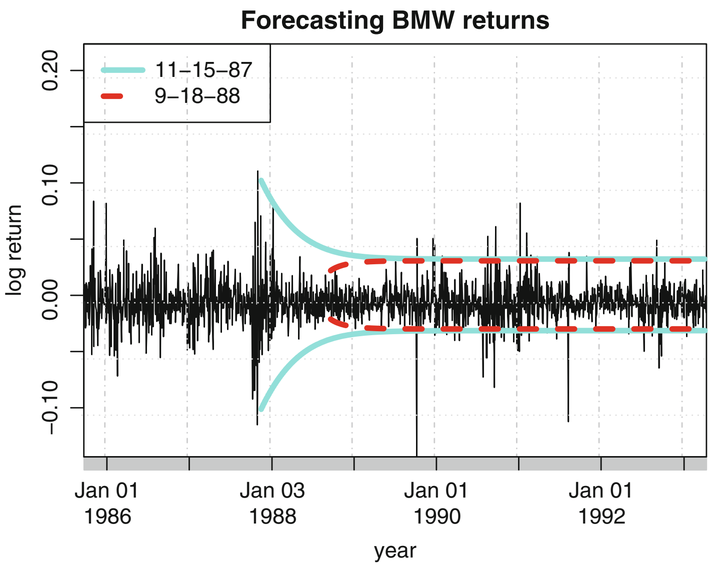


Fig. 14.10. *Prediction limits for forecasting daily BMW stock log returns from two different time origins.*

Figure 14.10 plots the returns from 1986 until 1992. Forecast limits are also shown for two time origins, November 15, 1987 and September 18, 1988. At the first time origin, which is soon after Black Monday, the markets were very volatile. The forecast limits are wide initially but narrow as the conditional standard deviation converges downward to the marginal standard deviation. At the second time origin, the markets were less volatile than usual and the