第12章 波动率和波动率模型

内容

- 传统的风险测度
- 波动率特点
- 波动率模型和模拟

概述

- 金融风险往往来源于未来的不确定性;
- 通常假设股票价格服从对数正态分布，因而股票收益率服从正态分布。基于这个假设，股票收益率的标准方差用来度量金融风险，也称波动率;
- 通常观察到波动率不是恒定的，是随机的；具有聚集效应(clustering)；具有均值回复的特性;杠杆效应等;
- 刻画波动率模型（条件异方差模型：从ARCH模型到GARCH模型）。

传统的风险测度

标准方差

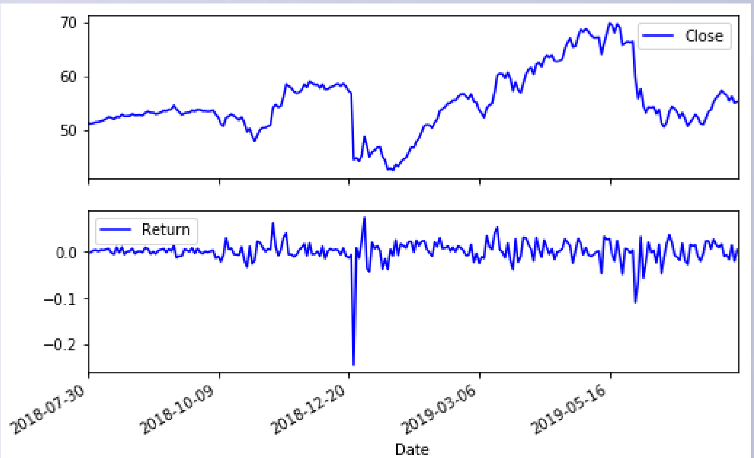
- 假设股票收益率r_t服从正态分布

$$r_t = \log(P_t / P_{t-1})$$

- 用收益率标准差(或方差)来衡量风险
- 通常称收益率标准差是波动率
- 编程计算DELL20180728-20190728日波动率和年波动率

```
import numpy as np
import pandas as pd
import scipy as sp
dell=pd.read_csv("D:/dataset/DELL20180728-20190728.csv",index_col=0)
dell['Return'] = np.log(dell['Close']
                         / dell['Close'].shift(1))
dell[['Close', 'Return']].plot(subplots=True, style='b',
                          figsize=(8, 5))

daily_v = np.std(dell['Return'])
daily_v
annual_v = daily_v*np.sqrt(len(dell['Return']))
annual_v
```



- 用Shapiro-Wilk方法检验正态分布

stats.shapiro()

- 收益率的其他统计量刻画：偏度skew;
  峰度kurtosis
- 下偏标准方差LPSD=$\frac{\sum_{i=1}^{m}(\mathrm{r}_t-\mathrm{r}_f)^2}{n-1}$，其中$\mathrm{r}_t-\mathrm{r}_f>0$
- 用Bartlett方法检验方差是否相等

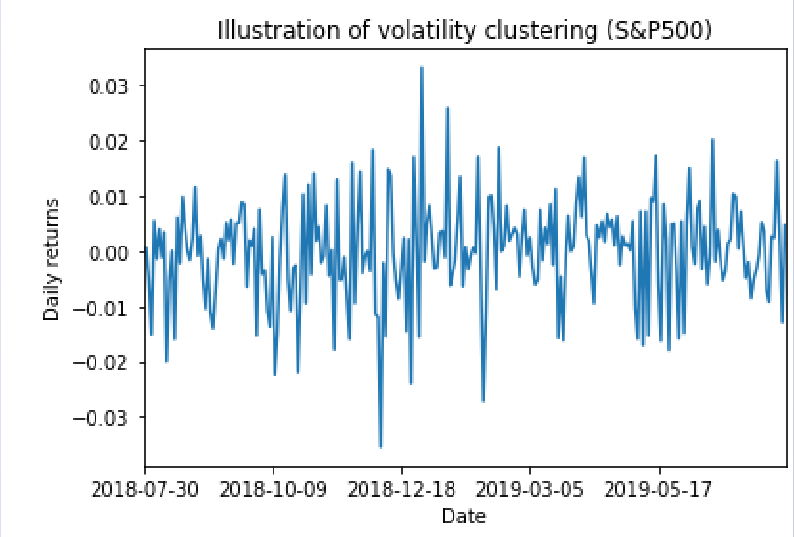
stats.bartlett()

波动率特点

典型特征

- 随机波动率：波动率既不是恒定，也没有确定的变化规律;
- 波动率聚集：高波动率时间经常聚集在一段时间，波动率测度通常会存在正的自相关性;
- 波动率均值回复，但是均值也会随时间变化;
- 杠杆效应：波动率和资产的收益率之间是负相关的。收益率增加，波动率减小。
- 厚尾：与正态分布相比，过高的收益率和过低的收益率都更常见

- 波动率微笑曲线：期权定价模型中，隐含波动率对行使价格的函数关系呈微笑状态，先下降，后上升。
- 波动率聚集图像：




```
import numpy as np
import pandas as pd
import scipy as sp
Import matplotlib.pyplot as plt
SP500=pd.read_csv("D:/dataset/^GDAXI.csv",index_col=0)
ret = np.log(SP500['Close']/ SP500['Close'].shift(1))
plt.title('Illustration of volatility clustering (S&P500)')
plt.ylabel('Daily returns')
plt.xlabel('Date')
ret.plot()
```

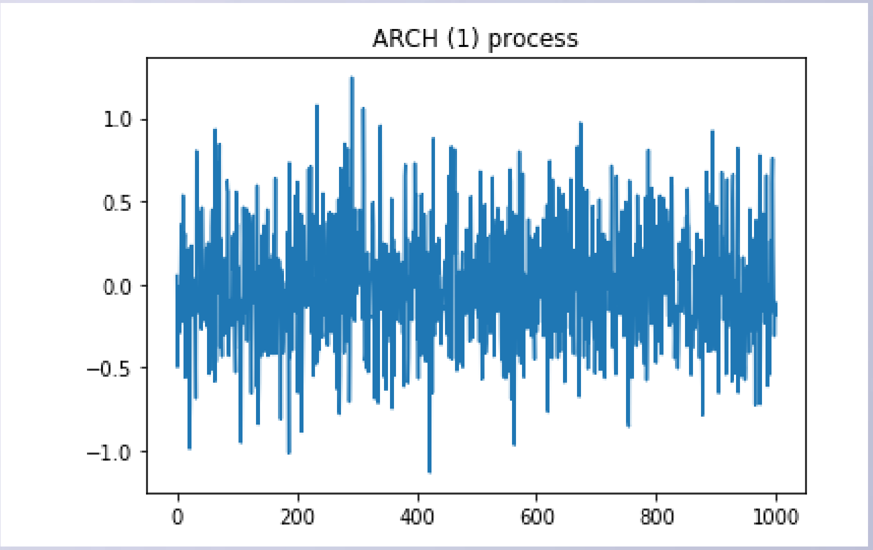
波动率模型

条件异方差模型：ARCH(q)

$$\sigma_t^2=\alpha_0+\sum_{i=1}^{q}\alpha_i\, e_{t-\mathrm{i}}^2$$

- 其中σ_t^2是在t时刻的方差，α_i是在t-i时刻的误差项的系数，$e_{t-\mathrm{i}}^2$是t-i时刻误差项的平方，q是模型阶数。
- 最简单情形：ARCH(1)：$\sigma_t^2=\alpha_0+\alpha_1 e_{t-1}^2$
- 假设e_{t-1}是iid标准正态，α_0=0.1，α_1=0.3，模拟ARCH(1)

```
#ARCH(1)
import scipy as sp
from matplotlib.pyplot import*
sp.random.seed(12345)
n=1000  # n is the number of observations
n1=100  # we need to drop the first several observations
n2=n+n1  # sum of two numbers
a=(0.1,0.3) # ARCH (1) coefficients alpha0 and alpha1, see Equation
errors=sp.random.normal(0,1,n2)   #assume i.i.d standard normal distributed
t=sp.zeros(n2)
t[0]=sp.random.normal(0,sp.sqrt(a[0]/(1-a[1])),1)
for i in range(1,n2-1):
    t[i]=errors[i]*sp.sqrt(a[0]+a[1]*t[i-1]**2)
y=t[n1-1:-1] # drop the first n1 observations
title('ARCH (1) process')
```

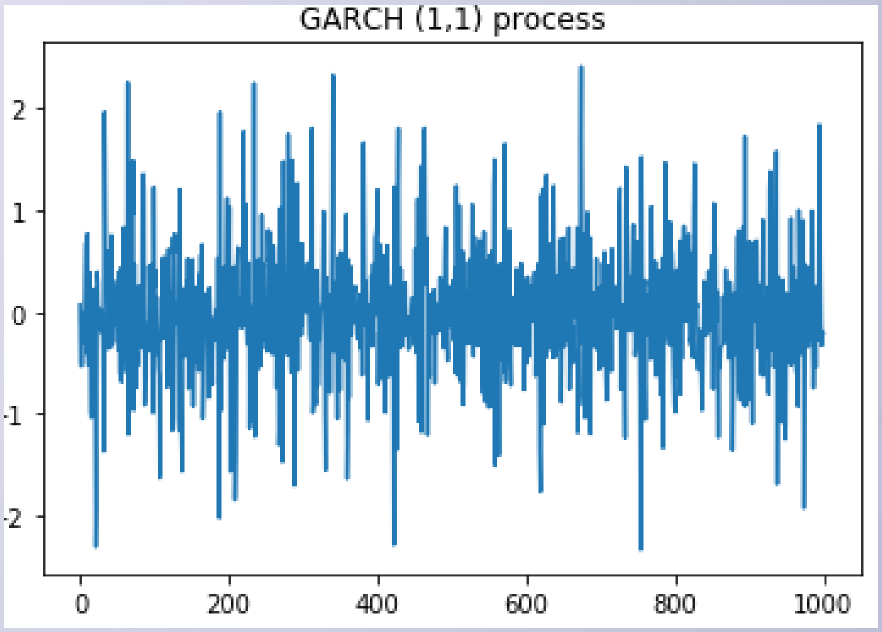


广义条件异方差模型：GARCH(p,q)

$$\sigma_t^2=\alpha_0+\sum_{i=1}^{q}\alpha_i\, e_{t-\mathrm{i}}^2+\sum_{i=1}^{p}\beta_i\,\sigma_{t-\mathrm{i}}^2$$

- 是ARCH模型的扩展
- 最简单情形GARCH(1,1)：$\sigma_t^2=\alpha_0+\alpha_1 e_{t-1}^2+\beta_1\sigma_{t-1}^2$
- 假设e_{t-1}是iid标准正态，α_0=0.1，α_1=0.3，β_1=0.2，模拟GARCH(1,1)

```
from matplotlib.pyplot import*
import scipy as sp
sp.random.seed(12345)
n=1000 # n is the number of observations
n1=100 # we need to drop the first several observations
n2=n+n1 # sum of two numbers
alpha=(0.1,0.3) # GARCH (1,1) coefficients alpha0 and alpha1
beta=0.2
errors=sp.random.normal(0,1,n2)
t=sp.zeros(n2)
t[0]=sp.random.normal(0,sp.sqrt(a[0]/(1-a[1])),1)
for i in range(1,n2-1):
    t[i]=errors[i]*sp.sqrt(alpha[0]+alpha[1]*errors[i-1]**2+beta*t[i-1]**2)
y=t[n1-1:-1] # drop the first n1 observations
title('GARCH (1,1) process')
x=range(n)
plot(x,y)
```



小 结

- 传统的风险测度
- 波动率特点
- 波动率模型和模拟