Time Series Project

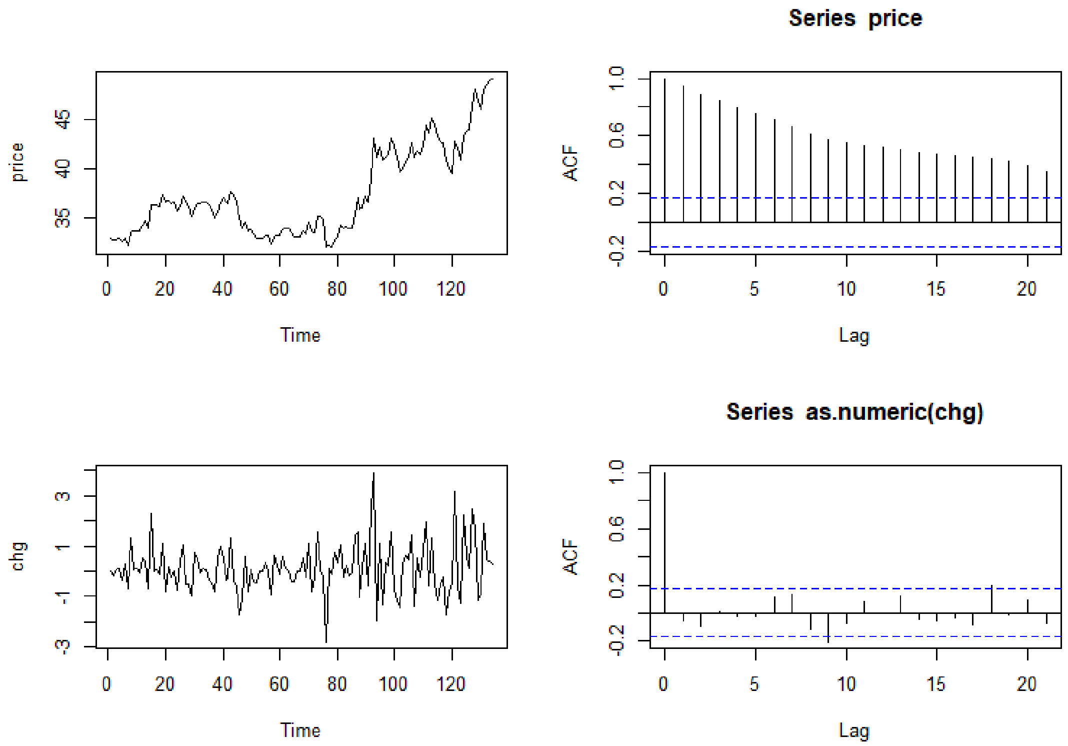
Zecheng Zhang
1155091892

Introduction:

We establish the ARIMA model in the prediction of stock prices of Hualan Bio (002007.SZ) in this report. We will use the data from Sep.16th 2019 to Apr.03rd 2020 and follow the process from exploratory analysis, model selection, diagnostics, to future stock price forecasting. Finally, we will see how a trading strategy based on the prediction of such models can profit during this time period.

Exploratory Data Analysis:

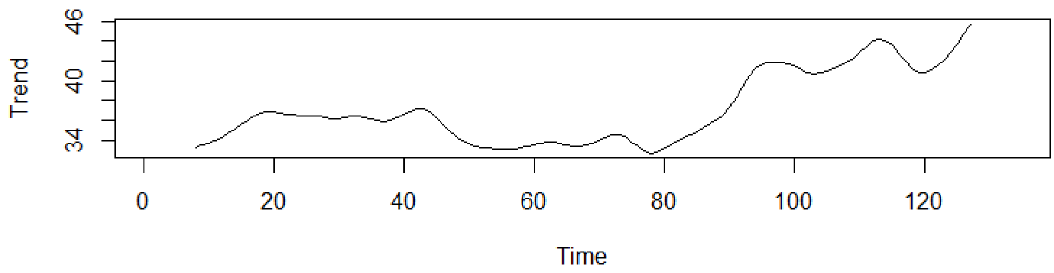
We plot the time series and its auto covariance function of the stock price and its difference.

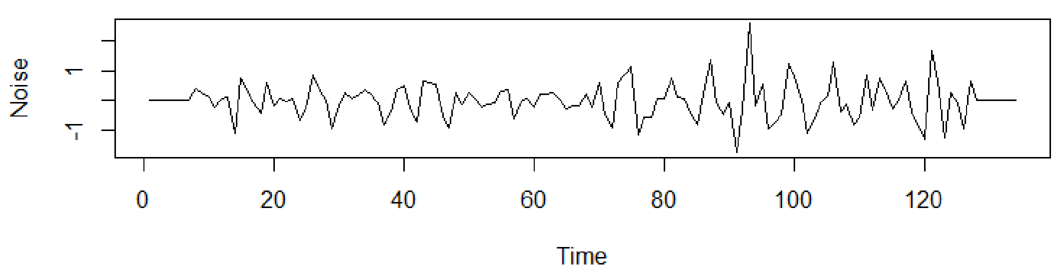


To filter out the trend, we use the Spencer 15-points filter:

$$\frac{1}{320}(-3,-6,-5,3,21,46,67,74,67,46,21,3,-5,-6,-3)$$

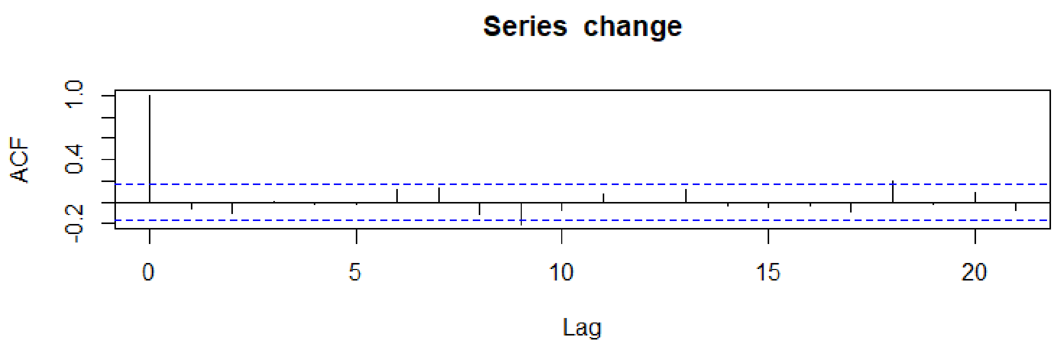
Below is the plot of the trend and its noise:

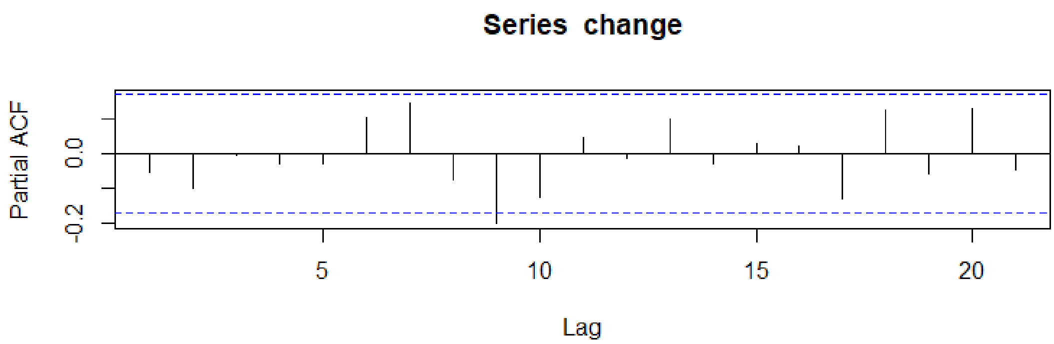




Model Selection and Fitting:

We notice that the price is nonstationary in both mean and variance, so we consider the difference in price.





We observe a stronger correlation in lag 9 in both acf and pacf of the difference in price, so we can consider AR(9), MA(9) and ARMA(9) as our choices. But to select the order of ARIMA model for the change in price, we compare the Final prediction error, AIC and BIC.

	FPE	AIC	BIC
AR(9)	1.017347	384.484840	255.000653
MA(9)	0.8999492	386.0346928	246.8928899
ARMA(9)	0.9057786	394.2037333	245.3453308

We observe that MA(9) has the least FPE and relatively small AIC and BIC. So we choose MA(9) as our model. This selection also conforms with the graphical method for order selection, since there is a cut-off at lag 9 for acf plot.

We use the Maximum likelihood estimator to fit the model:

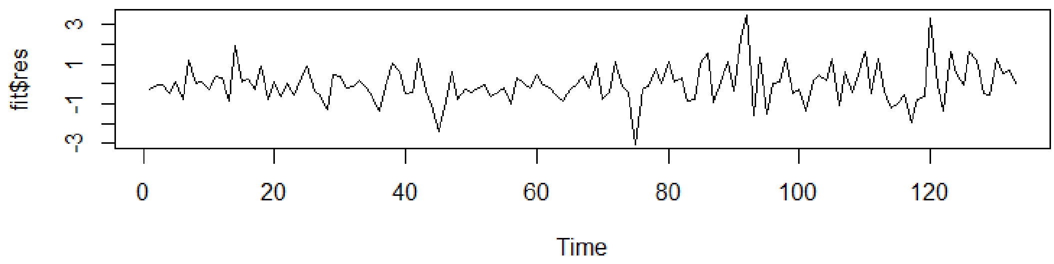
```
Call:
arima(x = change, order = c(0, 0, 9))

Coefficients:
          ma1      ma2     ma3      ma4      ma5     ma6     ma7      ma8      ma9  intercept
      -0.0949  -0.0383  0.0555  -0.0030  -0.0071  0.0650  0.0772  -0.1664  -0.1646     0.1119
s.e.   0.0840   0.0847  0.0907   0.0895   0.0884  0.0899  0.0982   0.0960   0.0800     0.0610

sigma^2 estimated as 0.8999:  log likelihood = -182.02,  aic = 386.03
```

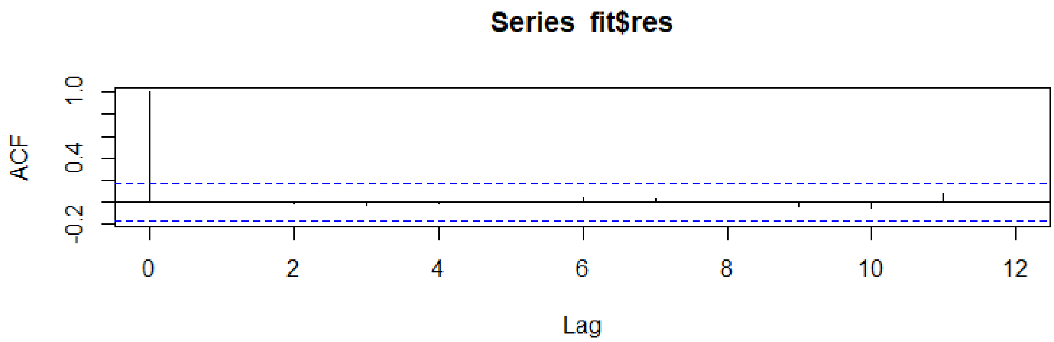
If we denote Y_t as the change in price from t to $t+1$, $Z_t \sim WN(0,0.9)$ be the white noise, then

$$Y_t = Z - 0.0949Z_{t-1} - 0.0382Z_{t-2} + 0.0555Z_{t-3} - 0.003Z_{t-4} - 0.0071Z_{t-5} + 0.065Z_{t-6} + 0.0772Z_{t-7} - 0.1664Z_{t-8} - 0.1646Z_{t-9}$$

Model Diagnostics:

We plot the residue between the fitting and the real data, and it look like white noise. Also, the acf plot of the residue are within the confidence interval. This means the MA(9) fit the data well.





Also we need to do the Portmanteau test:

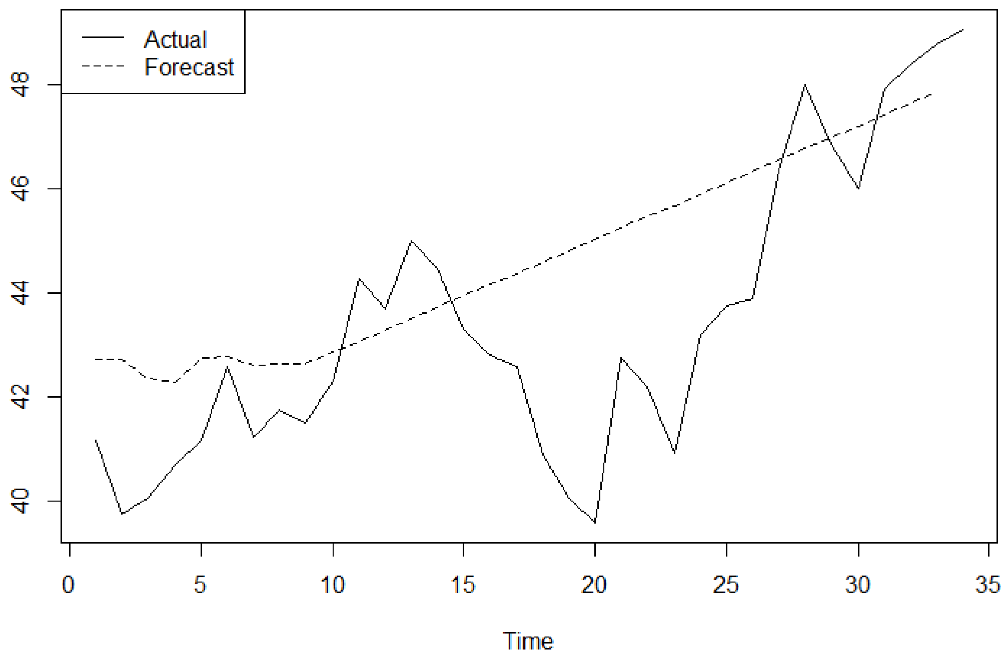
$$Q(h) = n(n+2)\sum_{j=1}^{h} \frac{\hat{r}_z(j)}{n-j} = 2.1554 < 7.815 = \chi^2(h-p-q)$$

So H_0 is not rejected. And therefore MA(9) is a good fitting.

Forecasting:

In this section, we do three different forecasting. In the first two examples, we split the data into two parts, historical data and the “future” data to see how the prediction behave and compare it with the “future” data. And in the last example, we use all our data to predict the real future data.

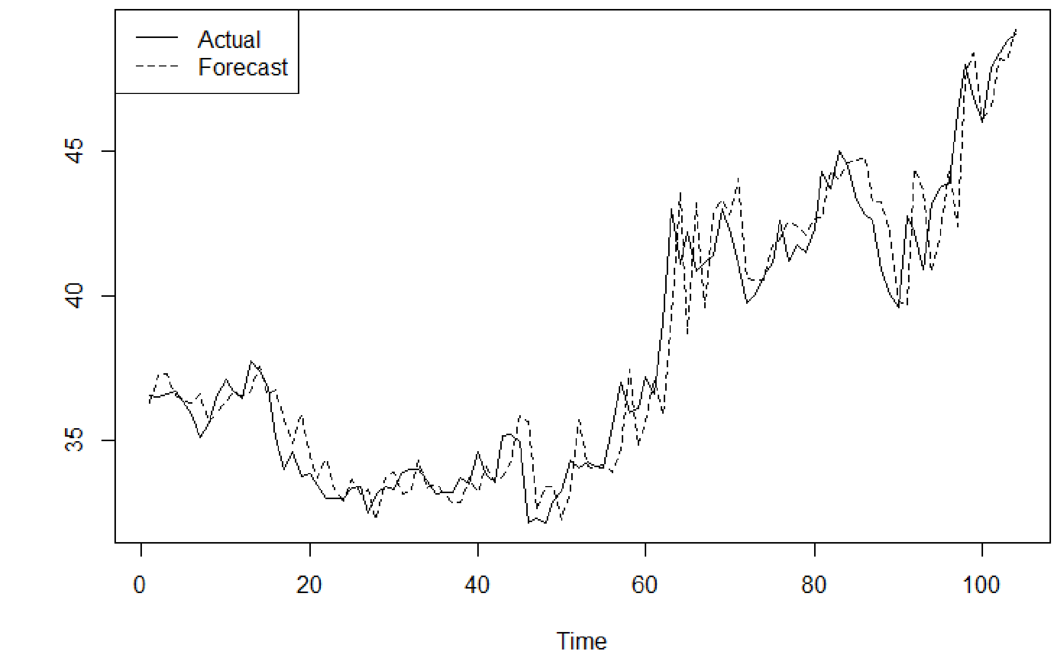
We use the previous 100 days price price (so the difference in price has 99 data) to build an MA(9) model to predict the next 34 days stock price by Box and Jenkins approach:



We observe that the prediction is quite close to the actual data, especially for the first 15 days, but it failed to predict the slump after day 15. But on such observation we could assume that the model can predict accurately in a short period.

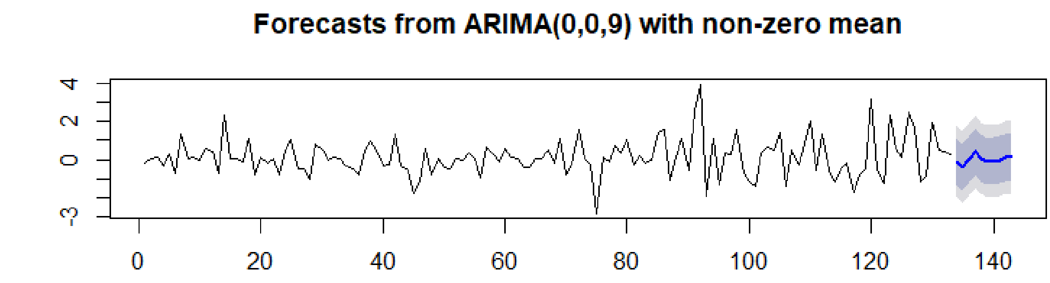
Then we also forecast the stock price of the next day by recursively adding the current stock price to the predicition of the change in price in the next day based on the MA(9) model built by previous 30 days stock price. If we denote S_t be the stock price on day t, $\widehat{S_t}$ be the predicted stock price on day t, $\widehat{Y_t}$ be the predicted change from day t to $t+1$. Then our forecast for $t+1$ is

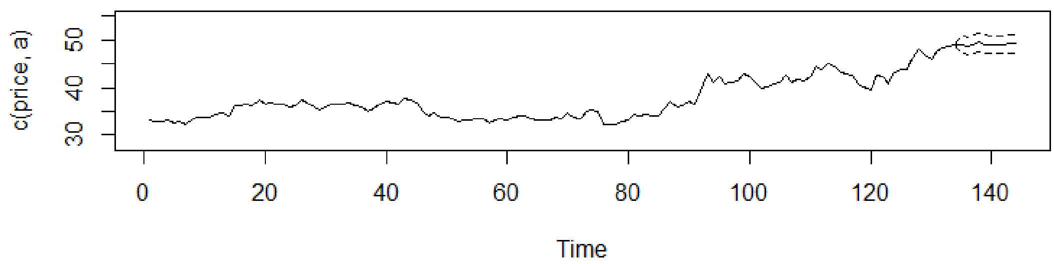
$$\hat{S}_{t+1} = S_t + \hat{Y}_t$$



We still see that the forecast is well coincided with the “future” data, this means that the hsitorical data used to built MA(9) doesn’t have to be very large.

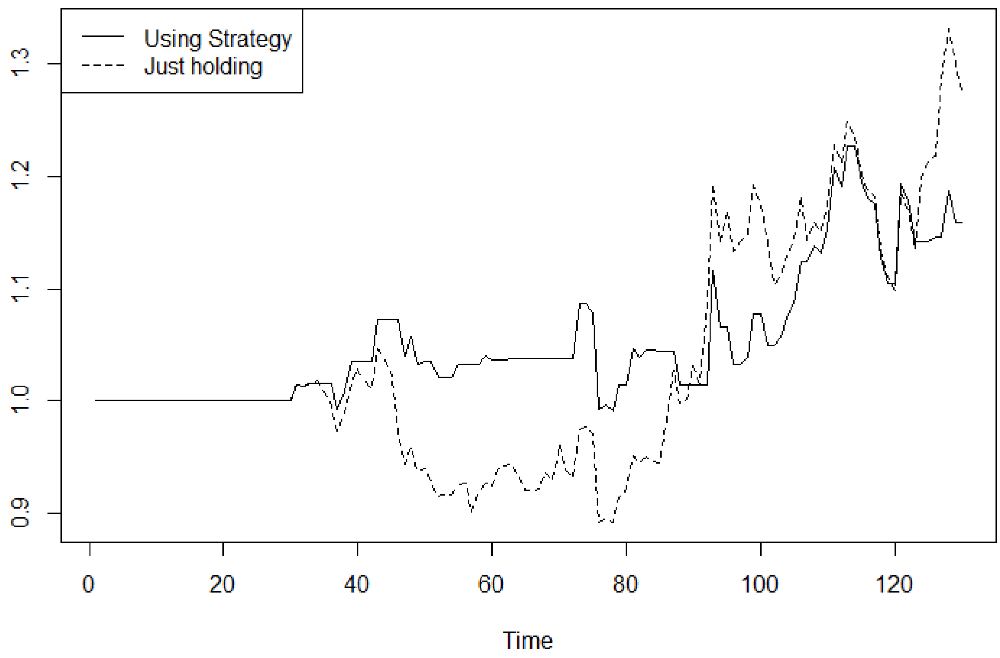
Finally, we do the forecast of the future stock price based on MA(9), and we are convinced that the future change in price will fall in the area in the first graph and the future price will fall in the area in the second graph:





Application:

Based on the forecast and our observation, we can develop a trading strategy so that we can make more profits by repeatedly buying or selling the stock than just holding it.
We use the previous 30 days data to predict the change in the next day. If the change in price is positive, we buy(or hold) the stock, otherwise, we don't buy(or sell) the stock.



Based on the above strategy, we plot a graph comparing the return by just holding the stock from day 0 to day 130 with the return by applying the above strategy, we observe that for the first several days when the stock price falls, this model has a good tenacity, but the model didn't catch the surge in price in around day 90. This occurance is due to the COVID-19 outbreak and Hualan Bio is happen to be an medical company which involves vaccine production as its primary industry. And also in around day 120, the model also missed one increase because the company claimed to have a progress in the vaccine. We could learn from this plot that the ARIMA model is good for stocks in volatility, but there is a delay in prediction if sudden favourable conditions occur.

Appendix:

```
library(readxl)
Hualan_info <- read_excel("C:/Users/roderickzzc/Desktop/Hualan_info.xls")
data=na.omit(Hualan_info)
data <- as.data.frame(data)
price=data[,7]
change=diff(price)
ts.plot(price)
```

```
acf(price)
pacf(price);
m=length(change)
T1=rep(NA,m)
filter=c(-3/320,-6/320,-5/320,3/320,21/320,46/320,67/320,74/320,67/320,46/320,21/320,3/320,-5/320,-
6/320,-3/320)
radius=7; start=8; end=m-7
for (k in start:end){
T1[k]=filter%*%price[(k-radius):(k+radius)]
}
ts.plot(T1,ylab="Trend")
IC=function(x,order.input=c(1,1,1)){
fit=arima(x,order=order.input);
n=length(x);p=order.input[1];q=order.input[3];sig=fit$ sigma2;
FPE=sig*(n+p)/(n-p);AIC=fit$ aic
BIC=(n-p-q)*log(n*sig/(n-p-1))+n*(1+log(sqrt(2*pi)))+(p+q)*log((sum(x^2)-n*sig)/(p+q));
return(c(FPE,AIC,BIC)) }
IC(price,order=c(1,1,1))
n=length(change)
fit=arima(change,order=c(0,0,9))
arima(change,order=c(0,0,9),include.mean = F)
par(mfrow=c(2,1))
ts.plot(fit$ res)
r.z=as.numeric(acf(fit$ res,12)$ acf) ## h=12
portmanteau.stat=n*(n+2)*sum((r.z[-1]^2)/(n-(1:12)))
portmanteau.stat>qchisq(0.95,12-9)
tfore=rep(NA,134)
for (i in (1:104)){
fit_i=arima(change[i:(i+28)],order=c(0,0,9))
dfore=predict(fit_i)$pred[1]
tfore[i+30]=price[i+29]+dfore
}
ts.plot(as.ts(price[31:134]),as.ts(tfore[31:134]),lty=c(1:2)) # 1-solid, 2-dashed, 3-dotted
leg.names=c("Actual","Forecast") # Draw legend
legend("topleft",leg.names,lty=c(1:2))
fore1=rep(NA,34)
dfore1=predict(fit_i,n.ahead=33)$pred[1:33]
fore1[1]=price[100]+dfore1[1]
for(i in (1:33)){
fore1[(i+1)]=dfore1[(i+1)]+fore1[(i)]
}
ts.plot(as.ts(price[101:134]),as.ts(fore1),lty=c(1:2)) # 1-solid, 2-dashed, 3-dotted
leg.names=c("Actual","Forecast") # Draw legend
legend("topleft",leg.names,lty=c(1:2))
```

```
library(forecast)
par(mfrow=c(2,1))
plot(forecast(fit))
fore=as.vector(forecast(fit))$mean;
lower=forecast(fit)$lower[,2];
upper=forecast(fit)$upper[,2];
a=as.vector(fore)+price[134]
b=as.vector(lower)+price[134]
c=as.vector(upper)+price[134]
ts.plot(c(price,a),ylim=c(28,55));
lines(c(price,b),lty=2);
lines(c(price,c),lty=2);
money=rep(NA,134)
money[1:30]=1
for (i in (1:100)){
if (tfore[i+30]>price[i+29]){
money[i+30]=money[i+29]/(price[i+29])*(price[i+30])
}else{money[i+30]=money[i+29]}
}
money_hold=rep(NA,134)
money_hold[1:30]=1
for (i in (1:100)){
money_hold[i+30]=money_hold[i+29]/(price[i+29])*(price[i+30])
}
par(mfrow=c(1,1))
ts.plot(as.ts(money[1:130]),as.ts(money_hold[1:130]),lty=c(1:2)) # 1-solid, 2-dashed, 3-dotted
leg.names=c("Using Strategy","Just holding") # Draw legend
legend("topleft",leg.names,lty=c(1:2))
```