时间序列分析 第四次作业

辛柏嬴 2020111753

2023-04-29

### 第一题

**模拟一个MA(2)模型**

#### (a) 计算极大似然估计

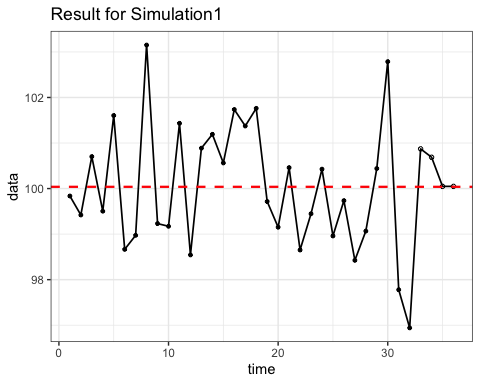
rm(list=ls())  
# generate random MA(2) data  
set.seed(10)  
ma\_order <- 2  
theta\_1 <- 1  
theta\_2 <- -0.6  
mu <- 100  
offset <- 100  
len <- 36  
dat\_raw <- arima.sim(n = len+offset,   
 list(ma = c(theta\_1,theta\_2)),  
 innov = rnorm(len+offset))+mu  
ar2dat <- ts(dat\_raw[(offset+1):(offset+len)])  
indat <- ts(ar2dat[1:32])  
outdat <- ts(ar2dat[33:36],start=33)  
# MLE  
ar.mle <- TSA::arima(indat, order=c(0,0,2),method='ML',include.mean = TRUE)  
ar.mle

##   
## Call:  
## TSA::arima(x = indat, order = c(0, 0, 2), include.mean = TRUE, method = "ML")  
##   
## Coefficients:  
## ma1 ma2 intercept  
## 0.1741 -0.3925 100.0482  
## s.e. 0.1834 0.1996 0.1917  
##   
## sigma^2 estimated as 1.783: log likelihood = -54.87, aic = 115.74

通过上述输出可以看到，极大似然估计的结果分别为0.1741,-0.3925,100.0482

#### (b) 预测并绘图

ar.mle.pred <- forecast(indat,model=ar.mle,h=4)  
pred <- data.frame(dat=as.matrix(ar.mle.pred$mean))  
indat.df <- data.frame(dat=as.matrix(indat))  
mix <- rbind(indat.df,pred)  
ggplot()+  
 geom\_point(data=indat.df,mapping = aes(y=dat,x=c(1:32)),  
 size=1)+  
 theme\_bw()+  
 geom\_point(data=pred,mapping=aes(y=dat,x=c(33:36)),  
 shape=21,size=1.2)+  
 geom\_line(data=mix, mapping=aes(y=dat,x=c(1:36)),  
 linewidth=0.6)+  
 labs(x = "time", y = "data")+  
 geom\_hline(yintercept=mean(mix$dat),  
 color='red',linetype="dashed",linewidth=0.8)+  
 ggtitle("Result for Simulation1")



print(ar.mle.pred$mean)

## Time Series:  
## Start = 33   
## End = 36   
## Frequency = 1   
## [1] 100.8712 100.6876 100.0482 100.0482

预测值以及绘制的时间序列图如上所示。

#### (c) 第三、四个预测值有什么特别之处？

第三四个预测值都是相同的，并且都基本落在了数据的均值水平线上。

#### (d) 将最后四个预测值与真实值进行比较，计算MSE与MAE

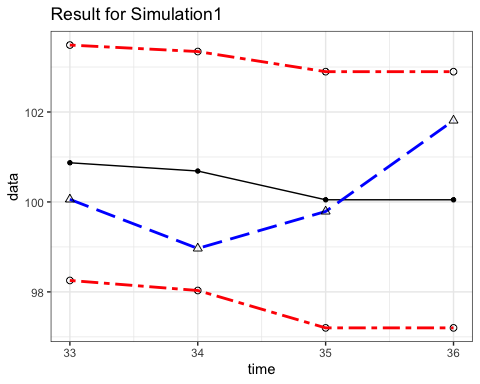
accuracy(ar.mle.pred$mean,outdat)

## ME RMSE MAE MPE MAPE ACF1 Theil's U  
## Test set -0.2584472 1.304165 1.139847 -0.2708068 1.136528 0.1244757 1.013609

MSE与MAE的输出如上所示，分别为1.304165，1.139847

#### (e) 画出预测区间，真实值是否在该区间内？

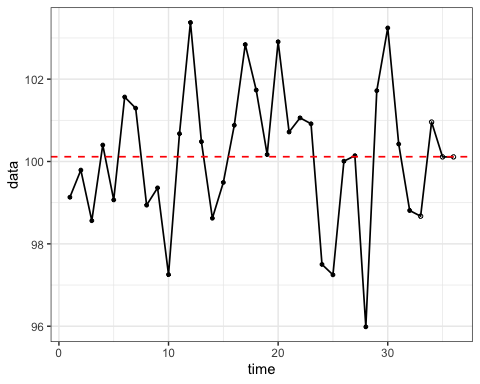
low <- data.frame(dat=ar.mle.pred$lower[,2])  
high <- data.frame(dat=ar.mle.pred$upper[,2])  
out <- data.frame(dat=outdat)  
ggplot()+  
 geom\_point(data=pred,mapping=aes(y=dat,x=c(33:36)),  
 shape=20,size=2)+  
 geom\_line(data=pred,mapping=aes(y=dat,x=c(33:36)),  
 linetype="solid")+  
 geom\_point(data=low,mapping=aes(y=dat,x=c(33:36)),  
 shape=1,size=2)+  
 geom\_line(data=low,mapping=aes(y=dat,x=c(33:36)),  
 linetype="twodash",linewidth=1,color='red')+  
 geom\_point(data=high,mapping=aes(y=dat,x=c(33:36)),  
 shape=1,size=2)+  
 geom\_line(data=high,mapping=aes(y=dat,x=c(33:36)),  
 linetype="twodash",linewidth=1,color='red')+  
 geom\_line(data=out,mapping=aes(y=dat,x=c(33:36)),  
 linetype="longdash",linewidth=1,color='blue')+  
 geom\_point(data=out,mapping=aes(y=dat,x=c(33:36)),  
 shape=2,size=2)+  
 labs(x = "time", y = "data")+  
 theme\_bw()+  
 ggtitle("Result for Simulation1")



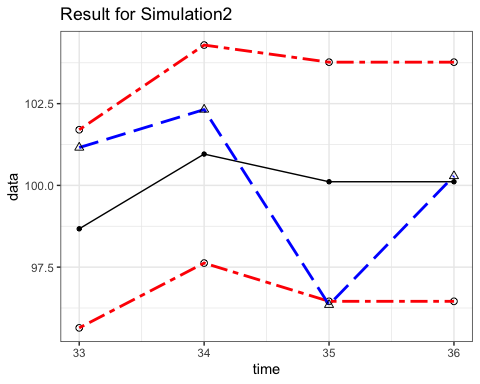
结果如上图所示，红色点线区间为95%的预测区间，黑色直线为预测值，蓝色长虚线为真实值。由此可见，真实值处在95%的预测区间内。

#### (f) 更改随机数种子，重复该过程

rm(list=ls())  
# generate random MA(2) data  
set.seed(12345)  
ma\_order <- 2  
theta\_1 <- 1  
theta\_2 <- -0.6  
mu <- 100  
offset <- 100  
len <- 36  
dat\_raw <- arima.sim(n = len+offset,   
 list(ma = c(theta\_1,theta\_2)),  
 innov = rnorm(len+offset))+mu  
ar2dat <- ts(dat\_raw[(offset+1):(offset+len)])  
indat <- ts(ar2dat[1:32])  
outdat <- ts(ar2dat[33:36],start=33)  
# MLE  
ar.mle <- TSA::arima(indat, order=c(0,0,2),method='ML',include.mean = TRUE)  
print(ar.mle)  
  
#predict  
ar.mle.pred <- forecast(indat,model=ar.mle,h=4)  
pred <- data.frame(dat=as.matrix(ar.mle.pred$mean))  
indat.df <- data.frame(dat=as.matrix(indat))  
mix <- rbind(indat.df,pred)  
ggplot()+  
 geom\_point(data=indat.df,mapping = aes(y=dat,x=c(1:32)),  
 size=1)+  
 theme\_bw()+  
 geom\_point(data=pred,mapping=aes(y=dat,x=c(33:36)),  
 shape=21,size=1.2)+  
 geom\_line(data=mix, mapping=aes(y=dat,x=c(1:36)),  
 linewidth=0.6)+  
 labs(x = "time", y = "data")+  
 geom\_hline(yintercept=mean(mix$dat),  
 color='red',linetype="dashed",linewidth=0.6)



print(ar.mle.pred$mean)  
  
print(accuracy(ar.mle.pred$mean,outdat))  
  
low <- data.frame(dat=ar.mle.pred$lower[,2])  
high <- data.frame(dat=ar.mle.pred$upper[,2])  
out <- data.frame(dat=outdat)  
ggplot()+  
 geom\_point(data=pred,mapping=aes(y=dat,x=c(33:36)),  
 shape=20,size=2)+  
 geom\_line(data=pred,mapping=aes(y=dat,x=c(33:36)),  
 linetype="solid")+  
 geom\_point(data=low,mapping=aes(y=dat,x=c(33:36)),  
 shape=1,size=2)+  
 geom\_line(data=low,mapping=aes(y=dat,x=c(33:36)),  
 linetype="twodash",linewidth=1,color='red')+  
 geom\_point(data=high,mapping=aes(y=dat,x=c(33:36)),  
 shape=1,size=2)+  
 geom\_line(data=high,mapping=aes(y=dat,x=c(33:36)),  
 linetype="twodash",linewidth=1,color='red')+  
 geom\_line(data=out,mapping=aes(y=dat,x=c(33:36)),  
 linetype="longdash",linewidth=1,color='blue')+  
 geom\_point(data=out,mapping=aes(y=dat,x=c(33:36)),  
 shape=2,size=2)+  
 labs(x = "time", y = "data")+  
 theme\_bw()+  
 ggtitle("Result for Simulation2")



##   
## Call:  
## TSA::arima(x = indat, order = c(0, 0, 2), include.mean = TRUE, method = "ML")  
##   
## Coefficients:  
## ma1 ma2 intercept  
## 0.4888 -0.5112 100.1114  
## s.e. 0.1971 0.1673 0.2697  
##   
## sigma^2 estimated as 2.321: log likelihood = -60.37, aic = 126.74  
## Time Series:  
## Start = 33   
## End = 36   
## Frequency = 1   
## [1] 98.67206 100.95828 100.11142 100.11142  
## ME RMSE MAE MPE MAPE ACF1 Theil's U  
## Test set 0.06393199 2.356877 1.945897 0.01332909 1.966639 -0.1013636 0.5432556

本次模拟的具体预测数值及MSE、MAE等见上述输出。从图中可以看出真实值稍有偏差95%的估计区间之外。但第三四次的预测值仍基本落在水平均值上，该规律没有改变。

### 第二题

**模拟一个ARIMA(1,1)模型**

#### (a) 计算极大似然估计

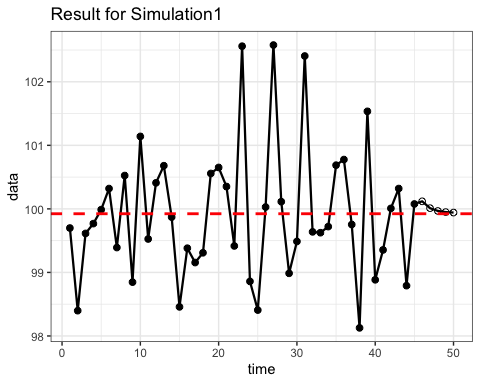
rm(list=ls())  
# generate random MA(2) data  
set.seed(100)  
offset <- 200  
len <- 50  
mu <- 100  
dat\_raw <- arima.sim(n = len+offset,   
 list(ma = c(-0.5),ar=c(0.7)),  
 innov = rnorm(len+offset))+mu  
arimadat <- ts(dat\_raw[(offset+1):(offset+len)])  
indat <- ts(arimadat[1:45])  
outdat <- ts(arimadat[46:50],start=46)  
# MLE  
fit <- TSA::arima(indat, order=c(1,0,1),method='ML',include.mean = TRUE)  
fit

##   
## Call:  
## TSA::arima(x = indat, order = c(1, 0, 1), include.mean = TRUE, method = "ML")  
##   
## Coefficients:  
## ar1 ma1 intercept  
## 0.4065 -0.6848 99.9372  
## s.e. 0.3701 0.3094 0.0835  
##   
## sigma^2 estimated as 0.9496: log likelihood = -62.77, aic = 131.54

#### (b) 预测并绘图

mle.pred <- forecast(indat,model=fit,h=5)  
pred <- data.frame(dat=as.matrix(mle.pred$mean))  
indat.df <- data.frame(dat=as.matrix(indat))  
mix <- rbind(indat.df,pred)  
ggplot()+  
 geom\_point(data=indat.df,mapping = aes(y=dat,x=c(1:45)),  
 size=2)+  
 theme\_bw()+  
 geom\_point(data=pred,mapping=aes(y=dat,x=c(46:50)),  
 shape=21,size=2)+  
 geom\_line(data=mix, mapping=aes(y=dat,x=c(1:50)),  
 size=0.8)+  
 labs(x = "time", y = "data")+  
 geom\_hline(yintercept=mean(mix$dat),  
 color='red',linetype="dashed",size=1)+  
 ggtitle("Result for Simulation1")

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.



print(mle.pred$mean)

## Time Series:  
## Start = 46   
## End = 50   
## Frequency = 1   
## [1] 100.11913 100.01114 99.96725 99.94941 99.94216

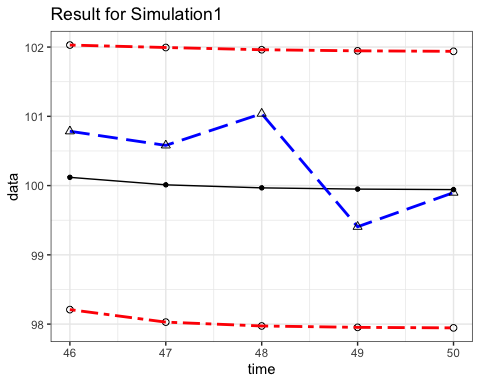
#### (c) 将最后四个预测值与真实值进行比较，计算MSE与MAE

accuracy(mle.pred$mean,outdat)

## ME RMSE MAE MPE MAPE ACF1  
## Test set 0.3438047 0.6652058 0.5786785 0.3392496 0.5754412 -0.04034475  
## Theil's U  
## Test set 0.7496215

#### (d) 画出预测区间，真实值是否在该区间内？

low <- data.frame(dat=mle.pred$lower[,2])  
high <- data.frame(dat=mle.pred$upper[,2])  
out <- data.frame(dat=outdat)  
ggplot()+  
 geom\_point(data=pred,mapping=aes(y=dat,x=c(46:50)),  
 shape=20,size=2)+  
 geom\_line(data=pred,mapping=aes(y=dat,x=c(46:50)),  
 linetype="solid")+  
 geom\_point(data=low,mapping=aes(y=dat,x=c(46:50)),  
 shape=1,size=2)+  
 geom\_line(data=low,mapping=aes(y=dat,x=c(46:50)),  
 linetype="twodash",size=1,color='red')+  
 geom\_point(data=high,mapping=aes(y=dat,x=c(46:50)),  
 shape=1,size=2)+  
 geom\_line(data=high,mapping=aes(y=dat,x=c(46:50)),  
 linetype="twodash",size=1,color='red')+  
 geom\_line(data=out,mapping=aes(y=dat,x=c(46:50)),  
 linetype="longdash",size=1,color='blue')+  
 geom\_point(data=out,mapping=aes(y=dat,x=c(46:50)),  
 shape=2,size=2)+  
 labs(x = "time", y = "data")+  
 theme\_bw()+  
 ggtitle("Result for Simulation1")

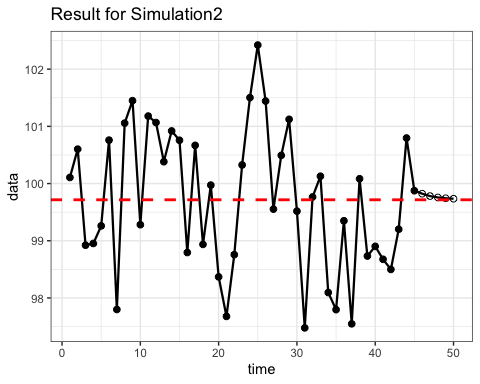


#### (e) 更改随机数种子，重复该过程

rm(list=ls())  
# generate random MA(2) data  
set.seed(999)  
offset <- 200  
len <- 50  
mu <- 100  
dat\_raw <- arima.sim(n = len+offset,   
 list(ma = c(-0.5),ar=c(0.7)),  
 innov = rnorm(len+offset))+mu  
arimadat <- ts(dat\_raw[(offset+1):(offset+len)])  
indat <- ts(arimadat[1:45])  
outdat <- ts(arimadat[46:50],start=46)  
# MLE  
fit <- TSA::arima(indat, order=c(1,0,1),method='ML',include.mean = TRUE)  
print(fit)

##   
## Call:  
## TSA::arima(x = indat, order = c(1, 0, 1), include.mean = TRUE, method = "ML")  
##   
## Coefficients:  
## ar1 ma1 intercept  
## 0.5824 -0.3256 99.7234  
## s.e. 0.3155 0.3621 0.2741  
##   
## sigma^2 estimated as 1.347: log likelihood = -70.6, aic = 147.2

mle.pred <- forecast(indat,model=fit,h=5)  
pred <- data.frame(dat=as.matrix(mle.pred$mean))  
indat.df <- data.frame(dat=as.matrix(indat))  
mix <- rbind(indat.df,pred)  
ggplot()+  
 geom\_point(data=indat.df,mapping = aes(y=dat,x=c(1:45)),size=2)+  
 theme\_bw()+  
 geom\_point(data=pred,mapping=aes(y=dat,x=c(46:50)),shape=21,size=2)+  
 geom\_line(data=mix, mapping=aes(y=dat,x=c(1:50)),size=0.8)+  
 labs(x = "time", y = "data")+  
 geom\_hline(yintercept=mean(mix$dat),color='red',linetype="dashed",size=1)+  
 ggtitle("Result for Simulation2")



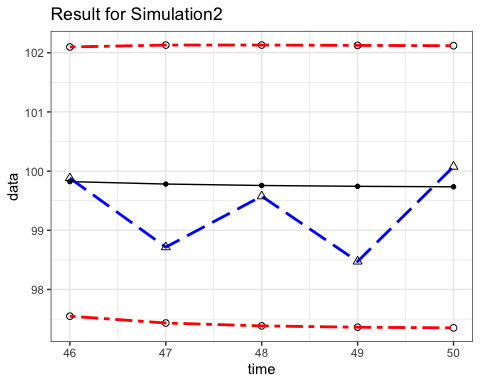
print(mle.pred$mean)

## Time Series:  
## Start = 46   
## End = 50   
## Frequency = 1   
## [1] 99.82281 99.78129 99.75711 99.74303 99.73482

print(accuracy(mle.pred$mean,outdat))

## ME RMSE MAE MPE MAPE ACF1 Theil's U  
## Test set -0.423807 0.7618719 0.5833369 -0.430714 0.5901691 -0.6572111 0.6977817

low <- data.frame(dat=mle.pred$lower[,2])  
high <- data.frame(dat=mle.pred$upper[,2])  
out <- data.frame(dat=outdat)  
ggplot()+  
 geom\_point(data=pred,mapping=aes(y=dat,x=c(46:50)),shape=20,size=2)+  
 geom\_line(data=pred,mapping=aes(y=dat,x=c(46:50)),linetype="solid")+  
 geom\_point(data=low,mapping=aes(y=dat,x=c(46:50)),shape=1,size=2)+  
 geom\_line(data=low,mapping=aes(y=dat,x=c(46:50)),linetype="twodash",  
 size=1,color='red')+  
 geom\_point(data=high,mapping=aes(y=dat,x=c(46:50)),shape=1,size=2)+  
 geom\_line(data=high,mapping=aes(y=dat,x=c(46:50)),linetype="twodash",  
 size=1,color='red')+  
 geom\_line(data=out,mapping=aes(y=dat,x=c(46:50)),linetype="longdash",  
 size=1,color='blue')+  
 geom\_point(data=out,mapping=aes(y=dat,x=c(46:50)),shape=2,size=2)+  
 labs(x = "time", y = "data")+  
 theme\_bw()+  
 ggtitle("Result for Simulation2")



### 第三题

#### (a) 用IMA(1,1)预测未来3个时刻的值，计算95%区间

rm(list=ls())  
data("robot")  
robot.inner <- ts(robot[1:321],start=1)  
robot.outer <- ts(robot[322:324],start=322)  
fit.ima <- TSA::arima(robot.inner,order = c(0,1,1),  
 include.mean = TRUE, method = "ML")  
robot.pred <- forecast(model = fit.ima,robot.inner,h=3)  
robot.pred$mean  
robot.pred$lower[,2]  
robot.pred$upper[,2]

## Time Series:  
## Start = 322   
## End = 324   
## Frequency = 1   
## [1] 0.001227982 0.001227982 0.001227982  
## Time Series:  
## Start = 322   
## End = 324   
## Frequency = 1   
## [1] -0.003602297 -0.003639542 -0.003676504  
## Time Series:  
## Start = 322   
## End = 324   
## Frequency = 1   
## [1] 0.006058261 0.006095506 0.006132468

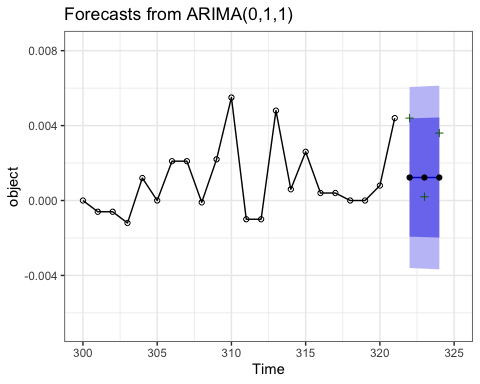
#### (b) 绘图展示预测值、95%预测区间、真实值，解释结果

autoplot(robot.pred)+  
 theme\_bw()+  
 xlim(c(300,325))+  
 geom\_point(aes(x=time(robot.pred$mean),y=robot.pred$mean))+  
 geom\_point(aes(x=time(robot.outer),y=robot.outer),  
 shape=3,color='darkgreen')+  
 geom\_point(aes(x=time(robot.inner),y=robot.inner),  
 shape=21)

## Scale for x is already present.  
## Adding another scale for x, which will replace the existing scale.

## Warning: Removed 299 rows containing missing values (`geom\_line()`).

## Warning: Removed 299 rows containing missing values (`geom\_point()`).



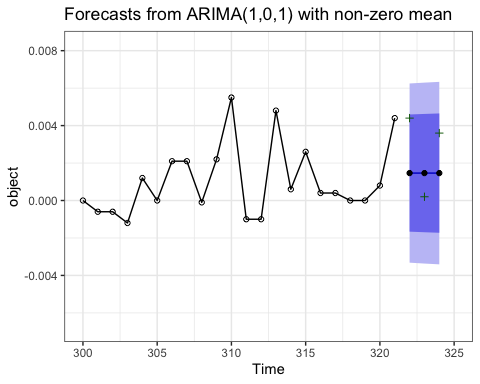
#### (c) 用ARMA(1,1)再次进行预测并比较两个模型的结果

fit.arima <- TSA::arima(robot.inner, order = c(1,0,1),   
 include.mean = TRUE, method="ML")  
robot.pred.arima <- forecast(model = fit.arima, robot.inner, h=3)  
  
robot.pred.arima$mean  
robot.pred.arima$lower[1]  
robot.pred.arima$upper[1]  
  
autoplot(robot.pred.arima)+  
 theme\_bw()+  
 xlim(c(300,325))+  
 geom\_point(aes(x=time(robot.pred.arima$mean),y=robot.pred.arima$mean))+  
 geom\_point(aes(x=time(robot.outer),y=robot.outer),  
 shape=3,color='darkgreen')+  
 geom\_point(aes(x=time(robot.inner),y=robot.inner),  
 shape=21)

## Scale for x is already present.  
## Adding another scale for x, which will replace the existing scale.

## Warning: Removed 299 rows containing missing values (`geom\_line()`).

## Warning: Removed 299 rows containing missing values (`geom\_point()`).

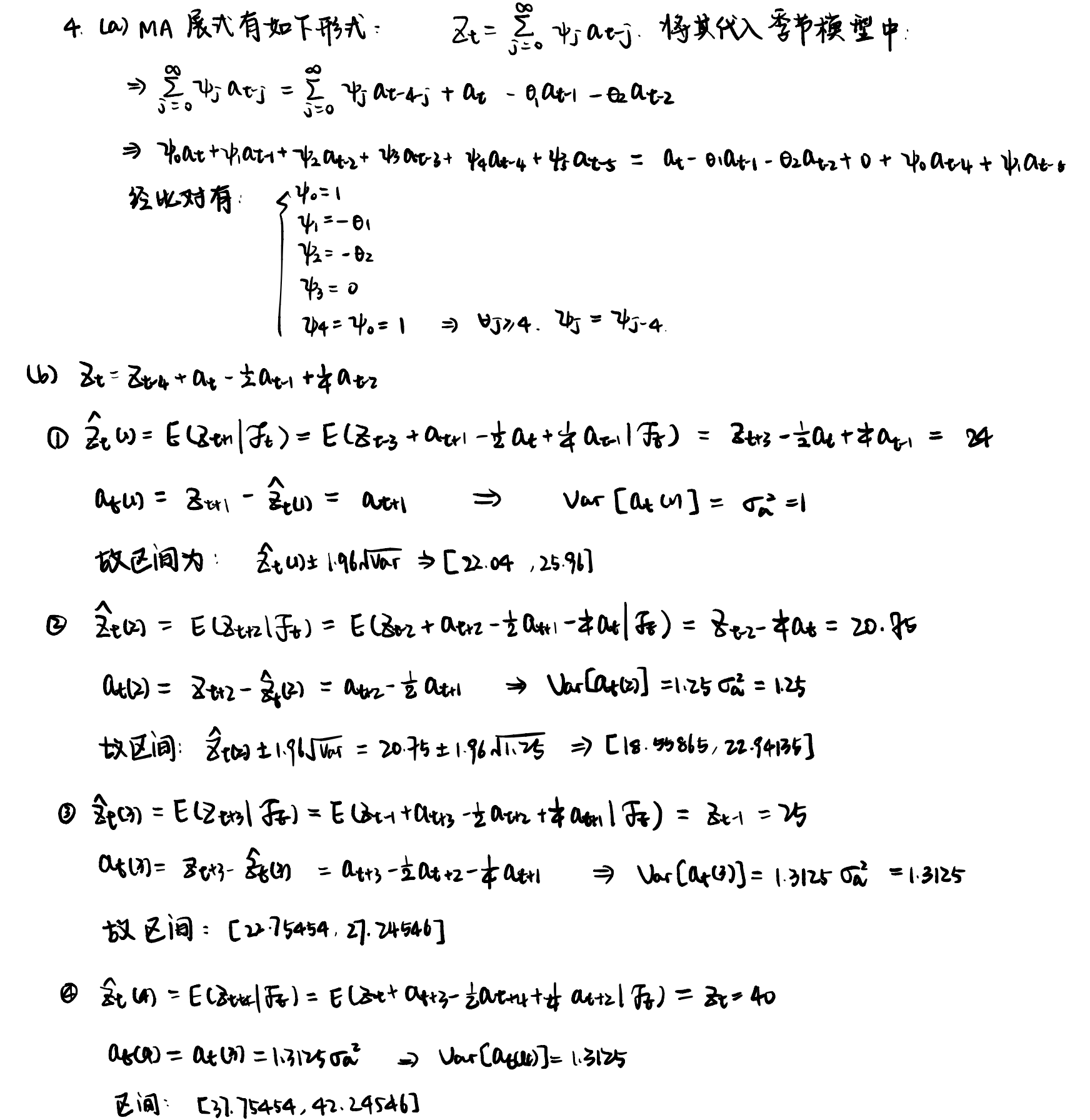


## Time Series:  
## Start = 322   
## End = 324   
## Frequency = 1   
## [1] 0.001465155 0.001464246 0.001463384  
## [1] -0.001662404  
## [1] 0.004592714

### 第四题

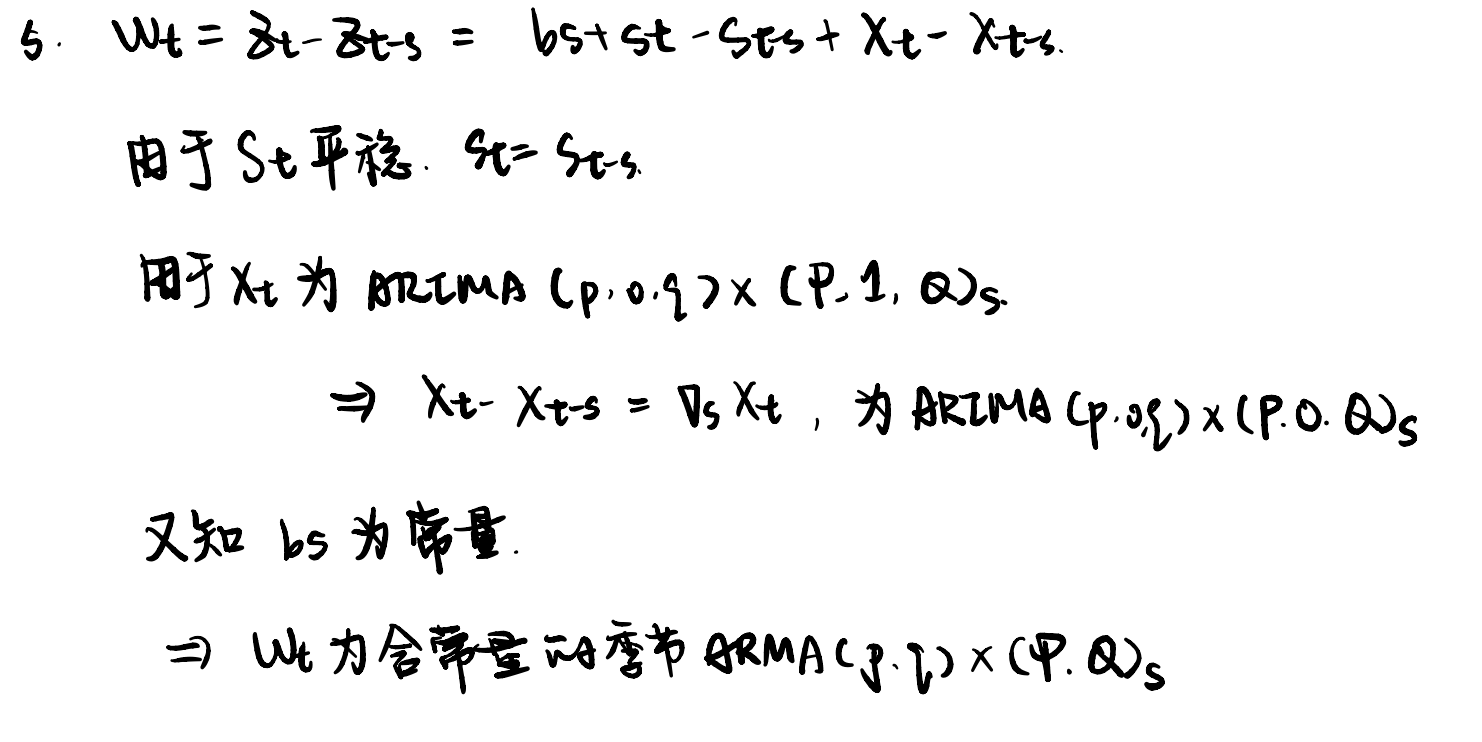
#### 推导MA展示对应的前四个ψ 权重

#### 计算未来四个季度的预测值及95%预测区间



### 第五题

#### 请问 是什么模型



### 第六题

文本, 信件

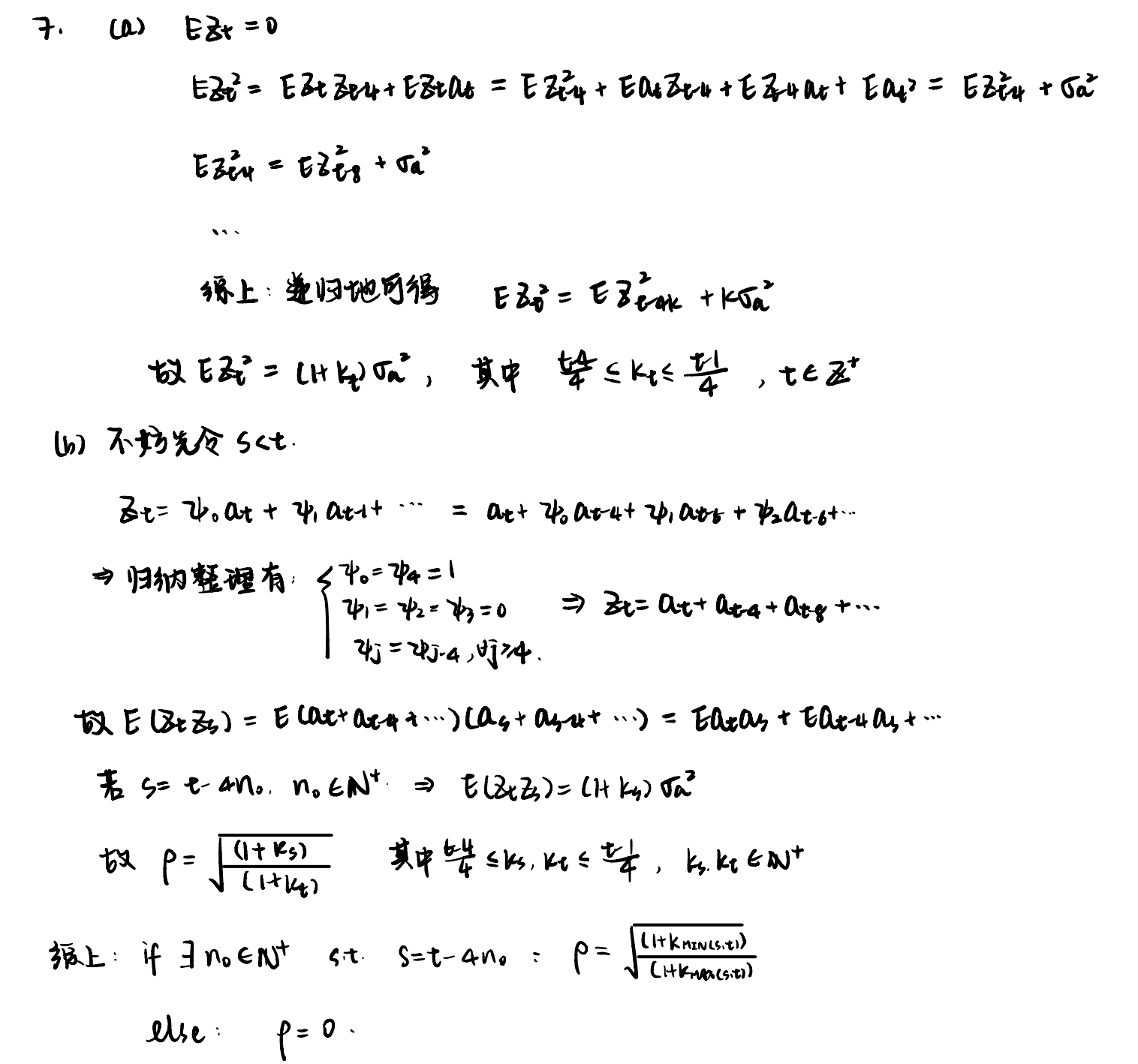
描述已自动生成

### 第七题

#### 求方差函数

#### 求自相关系数函数

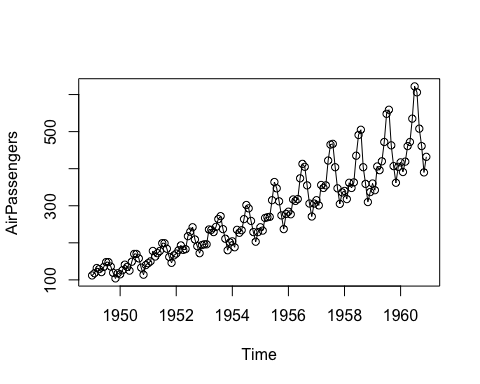
#### 证明是季节ARIMA



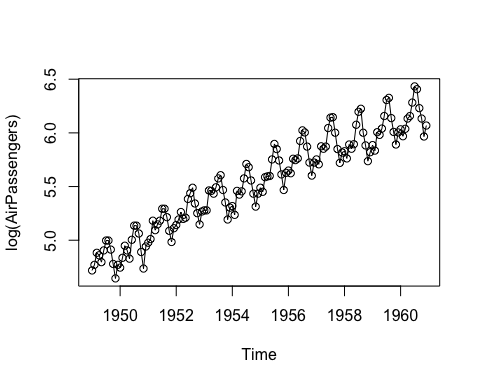
### 第八题

#### (a) 绘制原始时序图和对数时序图，说明对数变换的恰当性

rm(list=ls())  
data("AirPassengers")  
plot(AirPassengers,type='o',main="")

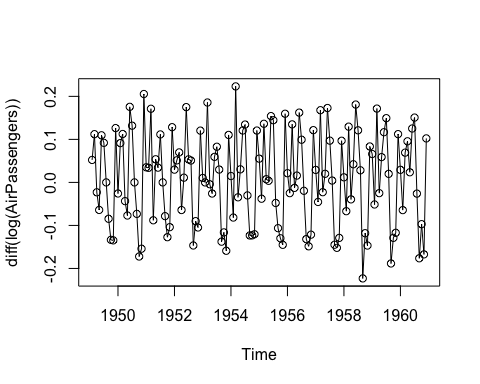


plot(log(AirPassengers),type='o',main="")



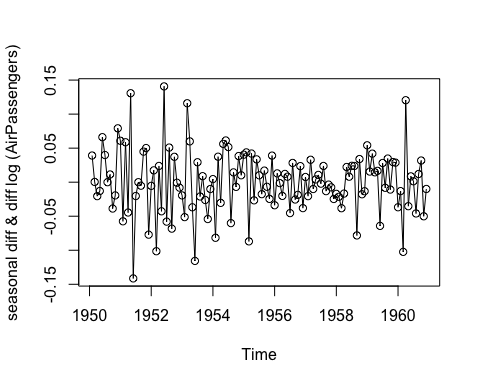
#### (b) 绘制并解释对数差分序列的时序图

plot(diff(log(AirPassengers)),type='o',main='')



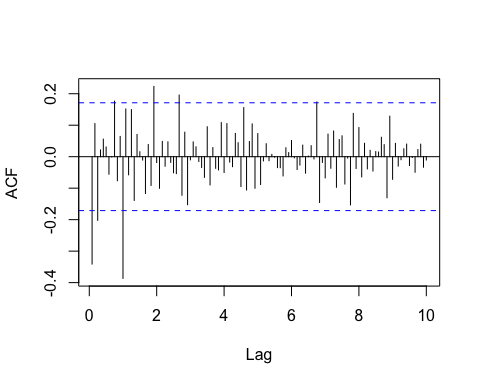
#### (c) 绘制并解释对数序列经一次差分和一次季节差分后的时序图;

plot(diff(diff(log(AirPassengers)),lag=12),type='o',  
 ylab='seasonal diff & diff log (AirPassengers)')



#### (d) 计算并解释对数序列经一次差分和一次季节差分后的样本 ACF

acf(diff(diff(log(AirPassengers)),lag=12),main="",lag.max = 120)



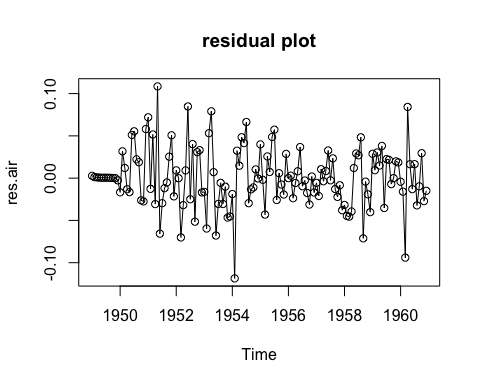
#### (e) 用 “航线模型” ARIMA (0, 1, 1) × (0, 1, 1)12 拟合对数序列

fit.air <- arima(log(AirPassengers),   
 order = c(0,1,1),   
 seasonal = list(order=c(0,1,1), period = 12),  
 method = "ML", include.mean = TRUE)  
fit.air

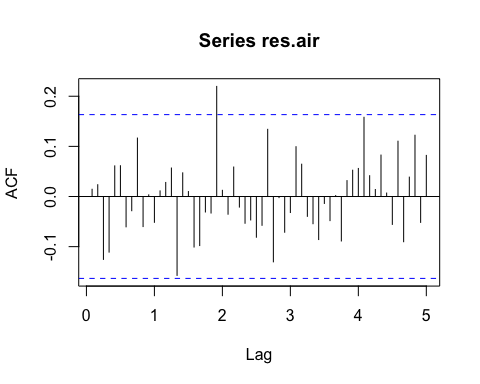
##   
## Call:  
## arima(x = log(AirPassengers), order = c(0, 1, 1), seasonal = list(order = c(0,   
## 1, 1), period = 12), include.mean = TRUE, method = "ML")  
##   
## Coefficients:  
## ma1 sma1  
## -0.4018 -0.5569  
## s.e. 0.0896 0.0731  
##   
## sigma^2 estimated as 0.001348: log likelihood = 244.7, aic = -485.4

#### (f) 基于残差序列对拟合模型进行诊断，包括诊断残差的自相关性和正态性;

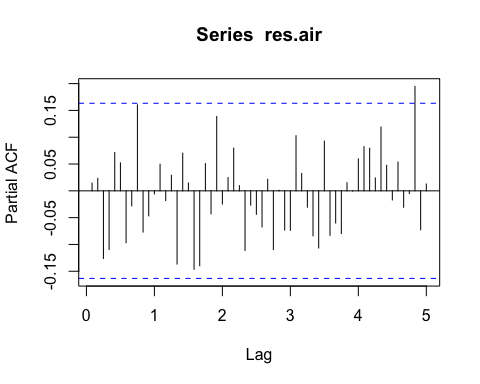
res.air <- fit.air$residuals  
plot(res.air,type='o', main='residual plot')



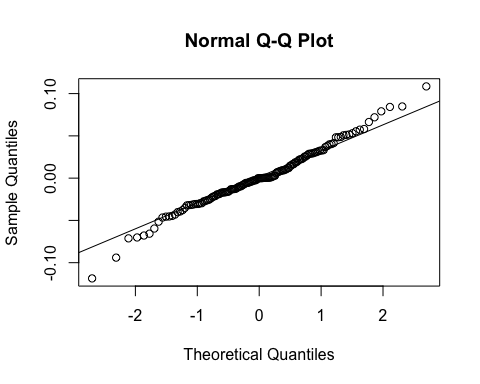
# relationship  
acf(res.air, lag.max = 60)



pacf(res.air, lag.max = 60)



Box.test(res.air, type="Ljung-Box", fitdf = 2, lag = 6)  
Box.test(res.air, type="Ljung-Box", fitdf = 2, lag = 12)  
Box.test(res.air, type="Ljung-Box", fitdf = 2, lag = 18)  
Box.test(res.air, type="Ljung-Box", fitdf = 2, lag = 24)  
Box.test(res.air, type="Ljung-Box", fitdf = 2, lag = 30)  
  
# normality  
qqnorm(res.air);qqline(res.air)

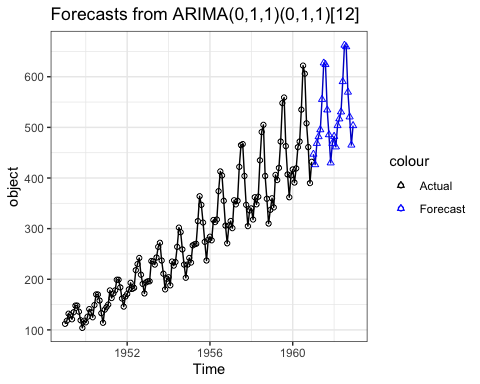


shapiro.test(res.air)

##   
## Box-Ljung test  
##   
## data: res.air  
## X-squared = 5.4434, df = 4, p-value = 0.2447  
##   
##   
## Box-Ljung test  
##   
## data: res.air  
## X-squared = 9.2333, df = 10, p-value = 0.5101  
##   
##   
## Box-Ljung test  
##   
## data: res.air  
## X-squared = 14.358, df = 16, p-value = 0.5721  
##   
##   
## Box-Ljung test  
##   
## data: res.air  
## X-squared = 26.446, df = 22, p-value = 0.233  
##   
##   
## Box-Ljung test  
##   
## data: res.air  
## X-squared = 29.494, df = 28, p-value = 0.3878  
##   
##   
## Shapiro-Wilk normality test  
##   
## data: res.air  
## W = 0.98637, p-value = 0.1674

#### (g) 假设前置时间为两年，对此序列进行预测，给出点预测和 95% 预测区间

air.pred <- forecast(model=fit.air, h=24, AirPassengers, level=c(95))  
autoplot(air.pred, prediction\_intervals = TRUE) +  
 geom\_point(aes(x = time(AirPassengers), y = AirPassengers, color = "Actual"),   
 shape = 21) +  
 geom\_point(aes(x = time(air.pred$mean), y = air.pred$mean, color = "Forecast"),   
 shape = 2) +  
 scale\_color\_manual(values = c("Actual" = "black", "Forecast" = "blue")) +  
 theme\_bw()



air.pred$mean  
air.pred$lower  
air.pred$upper

## Jan Feb Mar Apr May Jun Jul Aug  
## 1961 447.0625 426.3548 468.4009 481.6704 495.3550 555.2917 627.2105 624.1583  
## 1962 482.2070 461.4992 503.5454 516.8149 530.4995 590.4362 662.3550 659.3028  
## Sep Oct Nov Dec  
## 1961 534.4697 485.5144 429.8967 468.3593  
## 1962 569.6141 520.6589 465.0411 503.5037  
## Jan Feb Mar Apr May Jun Jul Aug  
## 1961 446.9906 426.2709 468.3067 481.5668 495.2428 555.1715 627.0829 624.0236  
## 1962 482.0304 461.3121 503.3482 516.6081 530.2836 590.2115 662.1219 659.0616  
## Sep Oct Nov Dec  
## 1961 534.3282 485.3666 429.7427 468.1994  
## 1962 569.3651 520.4022 464.7770 503.2324  
## Jan Feb Mar Apr May Jun Jul Aug  
## 1961 447.1345 426.4386 468.4952 481.7740 495.4672 555.4119 627.3382 624.2930  
## 1962 482.3836 461.6864 503.7426 517.0217 530.7154 590.6608 662.5881 659.5440  
## Sep Oct Nov Dec  
## 1961 534.6111 485.6622 430.0507 468.5191  
## 1962 569.8632 520.9156 465.3053 503.7751