

# 时间序列分析作业与第五次实验报告

姓名：康江睿

学号：2018213779

指导老师： 张晓飞

2020 年 11 月 26 日

## 1 模拟并可视化ARIMA序列与习题5.10

### 1.1 函数simARIMA简介

函数simARIMA实现了对给定参数的ARIMA(p,d,q)模型的仿真模拟。代码如下：

```
1 simARIMA <- function(AR.param = integer(0), MA.param = integer(0), diff, noise.sd = 1, ARMA.mean = 0, seq.  
  length = 10){  
2  
3   l = seq.length+1;  
4   d = diff;  
5   if ((length(AR.param)>0)&(length(MA.param)>0)){  
6     m = length(MA.param);n = length(AR.param);mode = 1  
7   }else if ((length(AR.param)>0)&(length(MA.param)==0)){  
8     n = length(AR.param);mode = 2;  
9   }else if ((length(AR.param)==0)&(length(MA.param)>0)){  
10    m = length(MA.param);mode = 3;  
11  }else if ((length(AR.param)==0)&(length(MA.param)==0)){  
12    mode = 4;  
13  }  
14  
15  if (mode==1){  
16    e = rnorm(n = l+m-1, mean = 0,sd = noise.sd);  
17    X = matrix(0, nrow = l-1, ncol = m);  
18    for (i in 1:m){  
19      X[,i] = e[(m-i+1):(l+m-i-1)]  
20    }  
21    W.MA = e[(m+1):(l+m-1)]-X%%MA.param  
22  }
```

```

23   W.AR = c(rnorm(n = n, mean = 0,sd = noise.sd), vector("numeric",length = l-1))
24   p = 0;
25   while (p<(l-1)){
26     W.AR[n+1+p] = crossprod(rev(W.AR[(1+p):(n+p)]),AR.para)+W.MA[p+1]
27     p = p+1
28   }
29
30   W = W.AR[(n+1):(n+l-1)]
31
32 }else if (mode==2){
33   e = rnorm(n = l-1, mean = 0,sd = noise.sd);
34
35   W.AR = c(rnorm(n = n, mean = 0,sd = noise.sd), vector("numeric",length = l-1))
36   p = 0;
37   while (p<(l-1)){
38     W.AR[n+1+p] = crossprod(rev(W.AR[(1+p):(n+p)]),AR.para)+e[p+1]
39     p = p+1
40   }
41
42   W = W.AR[(n+1):(n+l-1)]
43
44 }else if (mode==3){
45   e = rnorm(n = l+m-1, mean = 0,sd = noise.sd);
46   X = matrix(0, nrow = l-1, ncol = m);
47   for (i in 1:m){
48     X[,i] = e[(m-i+1):(l+m-i-1)]
49   }
50   W = e[(m+1):(l+m-1)]-X%*%MA.para
51 }else if (mode==4){
52   W = rnorm(n = l-1, mean = 0,sd = noise.sd);
53 }
54 W = W+ARMA.mean
55
56 if (d>0){
57   while (d>0){
58     dY = cumsum(W);
59     d = d-1;
60   }
61   Y = c(0,dY)
62 }else{
63   Y = W
64 }
65
66
67 return(Y)
68
69 }

```

## 1.2 函数plot.simARIMA简介

函数plot.simARIMA实现了对给定参数的ARIMA(p,d,q) 模型仿真模拟结果的可视化。代码如下：

```

1 plot.simARIMA <- function(AR.param = integer(0), MA.param=integer(0),
2                             diff, noise.sd = 1, ARMA.mean = 0, seq.length = 10, batch = 9){
3
4     library(ggplot2)
5     library(reshape2)
6
7     data = matrix(NA, nrow = seq.length+1, ncol = batch+1)
8     if ((length(AR.param)>0)&(length(MA.param)>0)){
9         para = AR.param; para.char = "AR "; m = length(para)
10        for (i in 1:m){
11            if (i == m){
12                para.char = paste(para.char, para[i])
13            }else{
14                para.char = paste(para.char, para[i], ",")
15            }
16        }
17        para.char = paste(para.char, "]")
18        para = MA.param; para.char = paste(para.char, "MA "); m = length(para)
19        for (i in 1:m){
20            if (i == m){
21                para.char = paste(para.char, para[i])
22            }else{
23                para.char = paste(para.char, para[i], ",")
24            }
25        }
26        para.char = paste(para.char, "]")
27    }else if ((length(AR.param)>0)&(length(MA.param)==0)){
28        para = AR.param; para.char = "AR "; m = length(para)
29        for (i in 1:m){
30            if (i == m){
31                para.char = paste(para.char, para[i])
32            }else{
33                para.char = paste(para.char, para[i], ",")
34            }
35        }
36        para.char = paste(para.char, "]")
37    }else if ((length(AR.param)==0)&(length(MA.param)>0)){
38        para = MA.param; para.char = "MA "; m = length(para)
39        for (i in 1:m){
40            if (i == m){
41                para.char = paste(para.char, para[i])
42            }else{
43                para.char = paste(para.char, para[i], ",")

```

```

44     }
45   }
46   para.char = paste(para.char, "]")
47 }else if ((length(AR.para)==0)&(length(MA.para)==0)){
48   para = "No parameters";
49 }
50
51 for (i in 1:batch){
52   data[,i] = simARIMA(AR.para, MA.para, diff, noise.sd, ARMA.mean, seq.length)
53 }
54 data[,batch+1] = 0:(seq.length);
55 name = c("serie1","serie2","serie3","serie4",
56         "serie5","serie6","serie7","serie8",
57         "serie9","x");
58 data = as.data.frame(data);names(data) = name[c(1:batch,10)];
59 data.melt=melt(data,id.vars = c("x"),variable.name = c("V"),value.name = c("y"));
60
61 max.data = ceiling(max(data.melt[,3]));min.data = floor(min(data.melt[,3]));
62 max.abs.data = max(abs(min.data),max.data)
63 if (max.data-min.data>80){
64   sz = 8;by.x = 5
65 }else if (max.data-min.data>60){
66   sz = 10;by.x = 2
67 }else{
68   sz = 12;by.x = 2
69 }
70 if (seq.length>100){
71   by.y = 10
72 }else{
73   by.y = 5
74 }
75
76 p_line = ggplot(data.melt)+
77   geom_line(aes(x=data.melt[,1],y=data.melt[,3],color=data.melt[,2]), size=1)+
78   scale_color_brewer(palette = "Set1")+
79   scale_x_continuous(breaks = seq(0, seq.length, by.y))+
80   scale_y_continuous(breaks = seq(-max.abs.data, max.abs.data, by.x))+
81   labs(x="Time",y="Value",color="Group")+
82   ggtitle(paste("Parameter(s) : ",para.char))+
83   theme(
84     legend.title = element_text(size = 15, face = "bold"),
85     legend.text = element_text(size = 15, face = "bold"),
86     legend.position = "right",
87     legend.key.size=unit(0.6,'cm'),
88     axis.ticks.x=element_blank(),
89     axis.text.x=element_text(size = 12,face = "bold", vjust = 0.5, hjust = 0.5),
90     axis.text.y=element_text(size = sz,face = "bold", vjust = 0.5, hjust = 0.5),
91     axis.title.x = element_text(size = 15,face = "bold", vjust = -0.5, hjust = 0.5),

```

```

92     axis.title.y = element.text(size = 15, face = "bold", vjust = 1.2, hjust = 0.5),
93     panel.background = element.rect(fill = "transparent", colour = "black"),
94     panel.grid.minor = element.line(color = "lightgrey", size = 0.1),
95     panel.grid.major = element.line(color = "lightgrey", size = 0.1),
96     plot.background = element.rect(fill = "transparent", colour = "white"),
97     plot.title = element.text(size = 18, face = "bold", vjust = 0.5, hjust = 0.5));
98
99     return(p_line);
100
101 }

```

### 1.3 函数使用说明

对于ARMA(p,q)过程，记模型为

$$W_t = \phi_1 W_{t-1} + \phi_2 W_{t-2} + \cdots + \phi_p W_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q}$$

对应的ARIMA(p,d,q)过程记为

$$\Delta^d Y = W_t$$

记  $\phi = [\phi_1, \phi_2, \dots, \phi_p]$ ,  $\theta = [\theta_1, \theta_2, \dots, \theta_q]$

(1)两个函数的输入参数几乎一样（只有batch是plot.simARIMA一个函数的输入参数），其名称与含义如下表：

输入参数名称	含义
AR.para	以上定义的 $1 \times p$ 向量 $\phi$
MA.para	以上定义的 $1 \times q$ 向量 $\theta$
diff	ARIMA(p,d,q)过程中的差分数d
noise.sd	噪声的标准差，默认为1
ARMA.mean	对应的ARMA模型的均值，默认为0
seq.length	模拟序列的长度
batch	plot.simARIMA函数中模拟序列的条数

表 1: 输入参数的名称与含义

(2)simARIMA函数的输出参数是一条长度为  $(1+seq.length)$  的向量  $Y$ ，其中  $Y[1] = 0$ ；plot.simARIMA函数的输出参数是一张图片。

(3)batch最低为1，最高为9；如果需要模拟更多条序列，请在绘图命令中将scale\_color\_brewer 定义的颜色集进行更改。

(4)注意：使用plot.simARIMA函数前请下载ggplot2包和reshape2包。

## 1.4 应用函数解决习题5.10

对于IMA(1,1)序列，设置模型参数集为 $\{-1, -0.3, 0.3, 1\}$ ；设置模拟条数为9，序列长度为100，模拟结果如下：

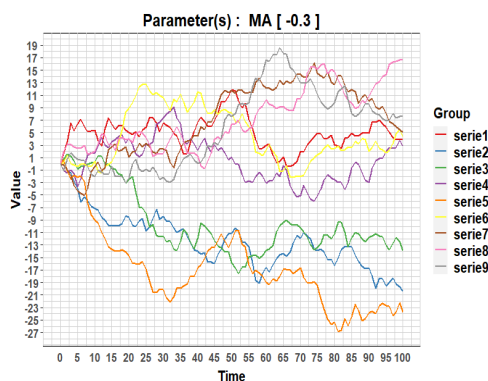
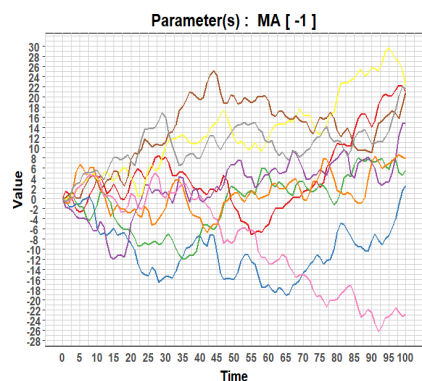


图 1: 参数为 $[-1]$ 的IMA(1,1)序列模拟结果图 图 2: 参数为 $[-0.3]$ 的IMA(1,1)序列模拟结果图

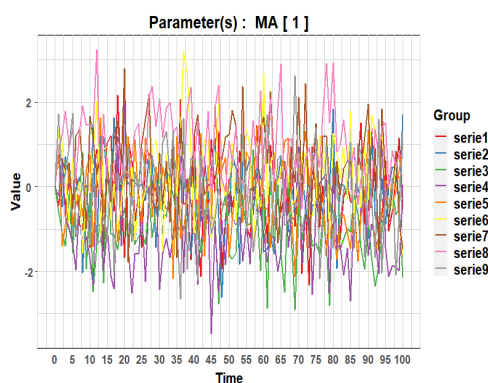
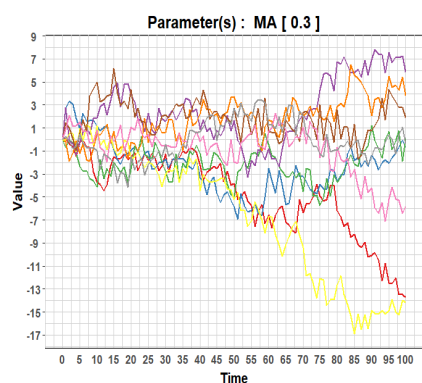


图 3: 参数为 $[0.3]$ 的IMA(1,1)序列模拟结果图 图 4: 参数为 $[1]$ 的IMA(1,1)序列模拟结果图

对于IMA(1,1)序列，设置模型参数集为P45的特例与习题4.2中的参数，共四组；设置模拟条数为9，序列长度为100，模拟结果如下：

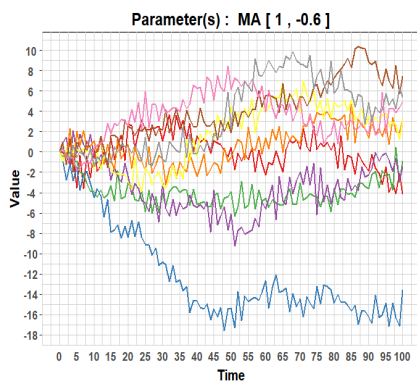
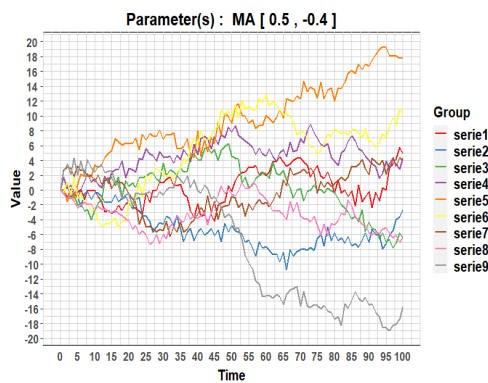


图 5: 参数为[1,-0.6]的IMA(2,2)序列模拟结果图



结果图

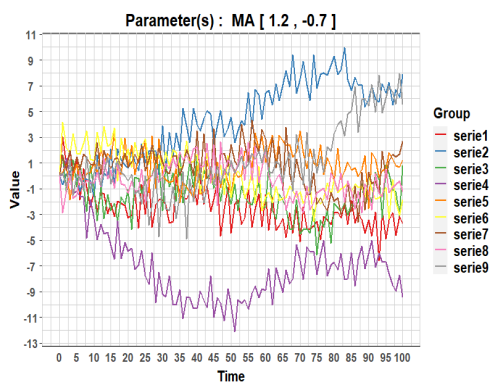
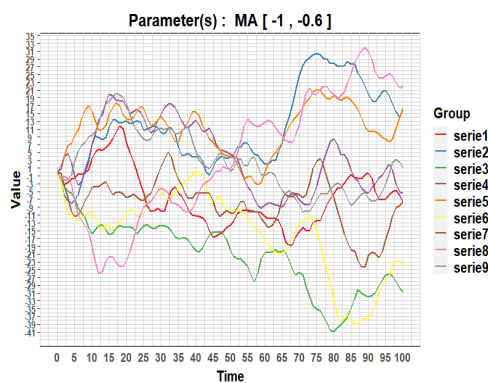


图 7: 参数为[1.2,-0.7]的IMA(2,2)序列模拟结果图



结果图

## 2 其他习题(5.11, 5.12, 5.13)

### 2.1 习题5.11

实现的代码如下:

```
1 library("TSA")
2 data("winnebago")
3 plot.ts(winnebago,type = "o")
4 log.winnebago = log(winnebago);
5 plot.ts(log.winnebago,type = "o")
6 pct.change = (tail(winnebago,-1)-head(winnebago,-1))/head(winnebago,-1)
7 diff.log = log(tail(winnebago,-1))-log(head(winnebago,-1))
8
9 plot(pct.change,type = "o",pch = 15,col = "blue",
```

```

10     xlab = "Time",ylab = "Value")
11 points(1:63, diff.log,pch = 16,col = "red")
12 lines(diff.log,col="red",lty = 2)
13 legend(45,1.8,c("pct.change","diff.log"),col=c("blue","red"),
14       text.col=c("blue","red"),pch=c(15,16),lty=c(1,2))

```

(a) 绘制的时间序列图如下：

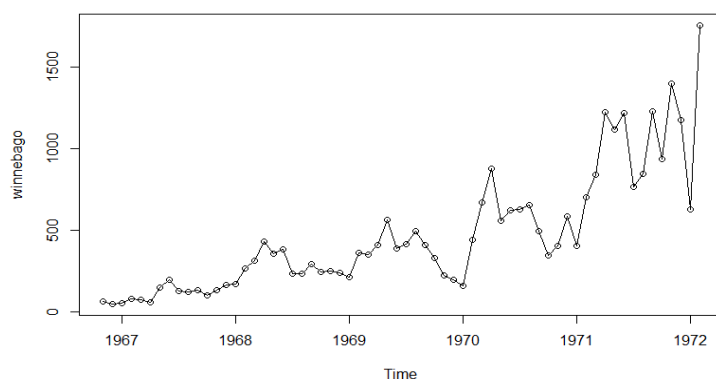


图 9: winnebago数据的时间序列图

可以看出，Winnebago休闲车的月度销量整体上随时间推移逐渐增长。从序列取值有（整体上）逐渐增长的趋势，与因变量值的极差（ $\hat{c}_{1500}$ ）可以看出，这一时间序列不是平稳的。因此我们需要使用某种变换来得到平稳的序列，以方便进一步的研究。

(b) 对月度销售量求自然对数后，绘制对数销售量的时间序列图如下：

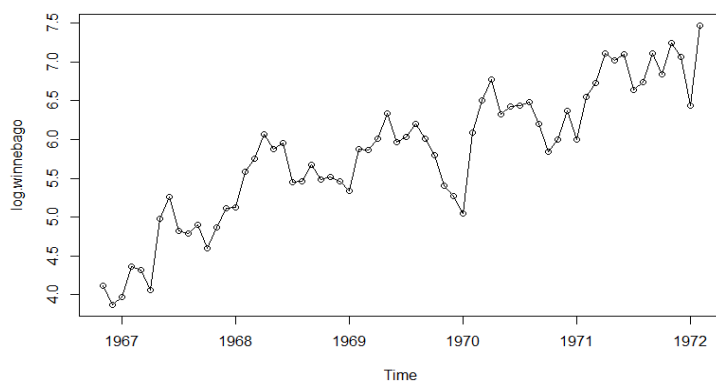


图 10: 对数变换下winnebago数据的时间序列图

此时的时间序列图仍然显现出了整体增长的趋势，但是明显增长的速率放



缓了许多：此时因变量值的极差只有3.5。事实上，在这个问题中，对数变换可以使时间序列变得相对稳定，并且可以让时间序列值处于同一个量级，方便之后对序列做进一步的研究。

(c) 对两种方法的比较可视化如下：

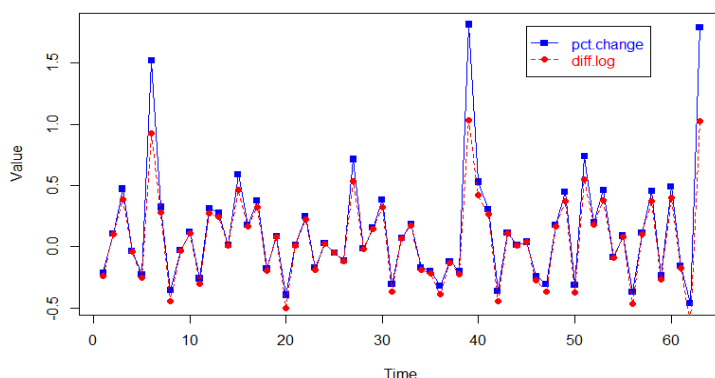


图 11: winnebago数据的相对变化率与对数差分的比较

可以发现，通过这两种方法得到的序列的增减趋势完全一致。对于原数据中变化为较小数值的情形，相对变化率和对数差分的取值非常接近；对于原数据中变化为较大数值的情形，对数差分要比相对变化率明显更接近0；在整体上，对数差分序列比相对变化率明显要平稳。

## 2.2 习题5.12

实现的代码如下：

```
1 library("TSA")
2 data("SP")
3 plot.ts(SP,type = "o")
4 log.SP = log(SP);
5 plot.ts(log.SP,type = "o")
6 pct.change = (tail(SP,-1)-head(SP,-1))/head(SP,-1)
7 diff.log = log(tail(SP,-1))-log(head(SP,-1))
8
9 plot(pct.change,type = "o",pch = 15,col = "blue",
10      xlab = "Time",ylab = "Value")
11 points(1:167, diff.log,pch = 16,col = "red")
12 lines(diff.log,col="red",lty = 2)
13 legend(100,0.38,c("pct.change","diff.log"),col=c("blue","red"),
14        text.col=c("blue","red"),pch=c(15,16),lty=c(1,2))
```

(a) 绘制的时间序列图如下：

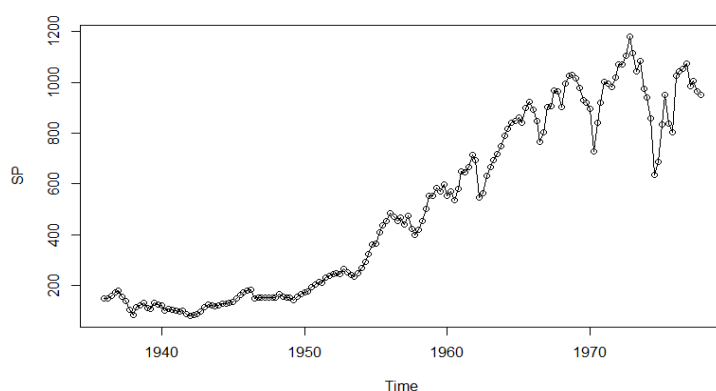


图 12: SP数据的时间序列图

可以看出，标普指数在约1968年前有明显的增长趋势，在1968年以后均值较稳定但是波动程度随时间推移增大。显然这一时间序列不是平稳的，因此我们需要使用某种变换来得到平稳的序列，以进行更深入的研究。

(b) 对标普指数求自然对数后，绘制对数标普指数的时间序列图如下：

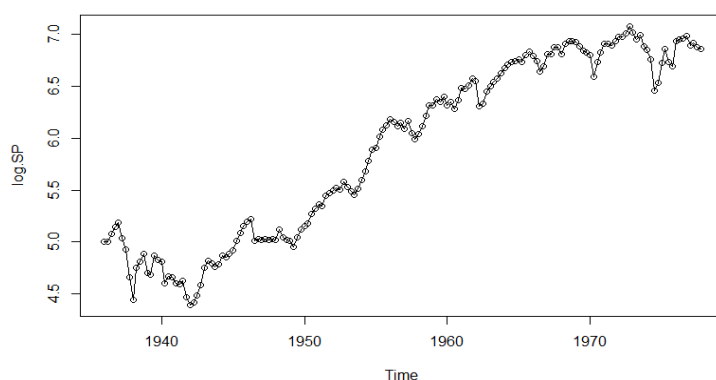


图 13: 对数变换下SP数据的时间序列图

此时的时间序列图显现出了整体增长的趋势，其中前期呈现不规律波动，约1950年开始有明显的增长趋势。在这个问题中，对数变换没有改变或者减缓数据的增长趋势，但是变换使得时间序列值处于同一个量级，方便之后对序列做进一步的研究。

(c) 对两种变换的比较可视化如下：

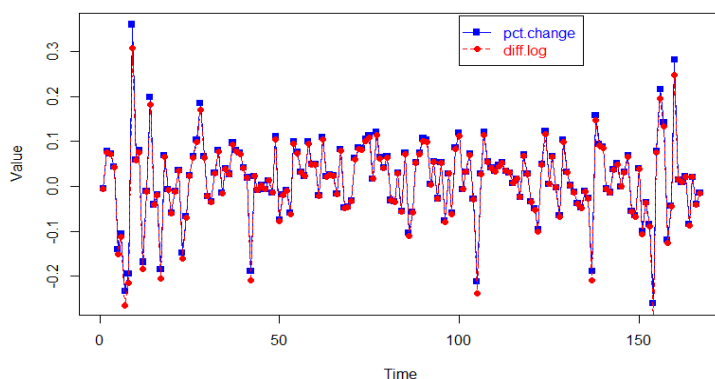


图 14: SP数据的相对变化率与对数差分的比较

可以发现，两种变换得到的序列的增减趋势完全一致。不管对于多大的变化，两个序列取值总是非常接近，不过对数差分的数据在尺度上更接近0。虽然如此，对于数值较大的情形，对数差分也没有比相对变化率小多少，因此可以近似地认为两种变换的效果相同。

## 2.3 习题5.13

实现的代码如下：

```

1 library("TSA")
2 data("airpass")
3 plot.ts(airpass,type="o")
4 log.airpass = log(airpass);
5 plot.ts(log.airpass,type="o")
6 pct.change = (tail(airpass,-1)-head(airpass,-1))/head(airpass,-1)
7 diff.log = log(tail(airpass,-1))-log(head(airpass,-1))
8
9 plot(pct.change,type="o",pch=15,col="blue",
10      xlab="Time",ylab="Value")
11 points(1:143,diff.log,pch=16,col="red")
12 lines(diff.log,col="red",lty=2)
13 legend(110,0.26,c("pct.change","diff.log"),col=c("blue","red"),
14        text.col=c("blue","red"),pch=c(15,16),lty=c(1,2))

```

(a) 绘制的时间序列图如下：

可以看出，月度国际航班乘客数量整体上随时间推移逐渐增长。直接观察数据，很容易发现数据的增减趋势是周期变化的（结合实际问题我们很容易想到这其实是俗称旺季和淡季导致的结果）；且随着时间推移，变化程度会加大。

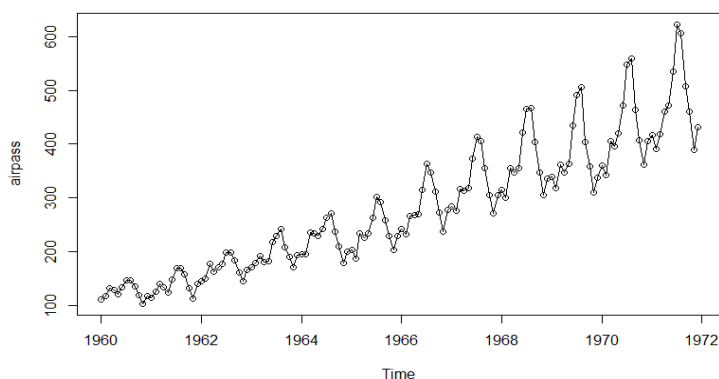


图 15: airpass数据的时间序列图

因此我们可以考虑用差分或者变化率使序列平稳化。

(b) 对月度国际航班乘客数量求自然对数后，绘制对数乘客数量的时间序列图如下：

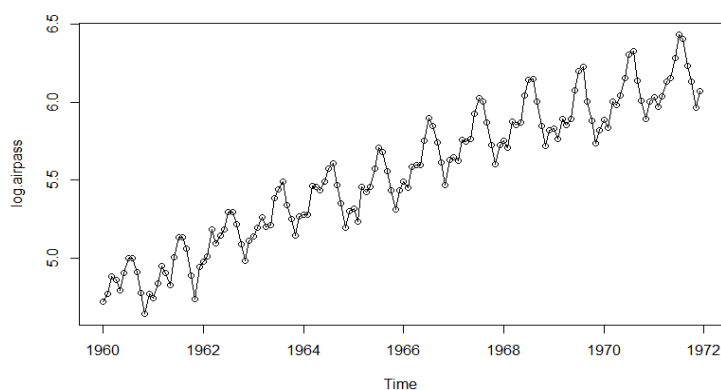


图 16: 对数变换下airpass数据的时间序列图

此时的时间序列图仍然显现出了整体增长的趋势，且数据的增减趋势与对应的周期是不变的。但是在对数序列中，随着时间推移，序列的变化幅度不变。它几乎是基于序列的初始值，每一年都以固定幅度的，先增后减的趋势，增加某个固定的值，而得到的。在这个问题中，对数变换让序列的规律更加突出，这可能是对数变换缩小了数据量级的结果。

(c) 对两种变换的比较可视化如下：

可以发现，两种变换得到的序列的增减趋势完全一致。不管对于多大的变化，两个序列取值总是非常接近，不过相对来说对数差分的数据在尺度上更接

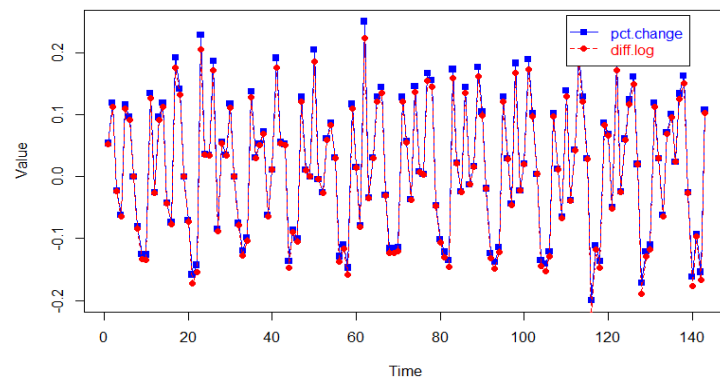


图 17: airpass数据的相对变化率与对数差分的比较

近0。虽然如此，对于数值较大的情形，对数差分也没有比相对变化率小多少，因此可以近似地认为两种变换的效果相同。