Lab3

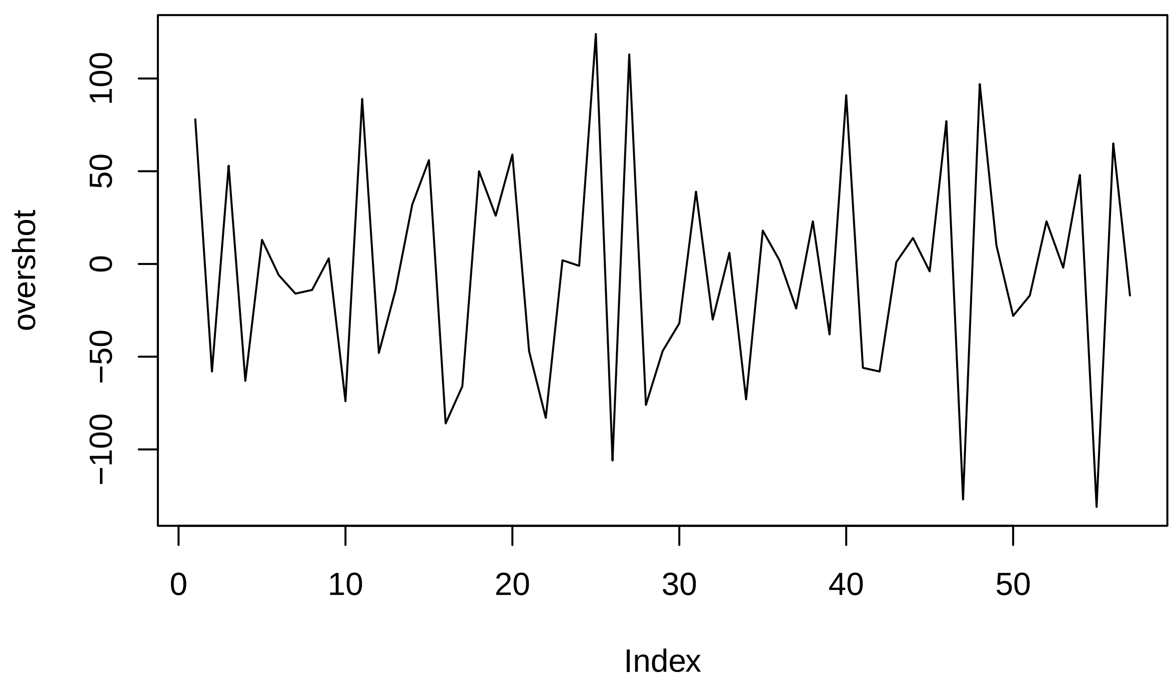
马宇骁

2022 年 5 月 21 日

习题 1

(1) 自行模拟 P69 例 3-10 并得出书本中的结论

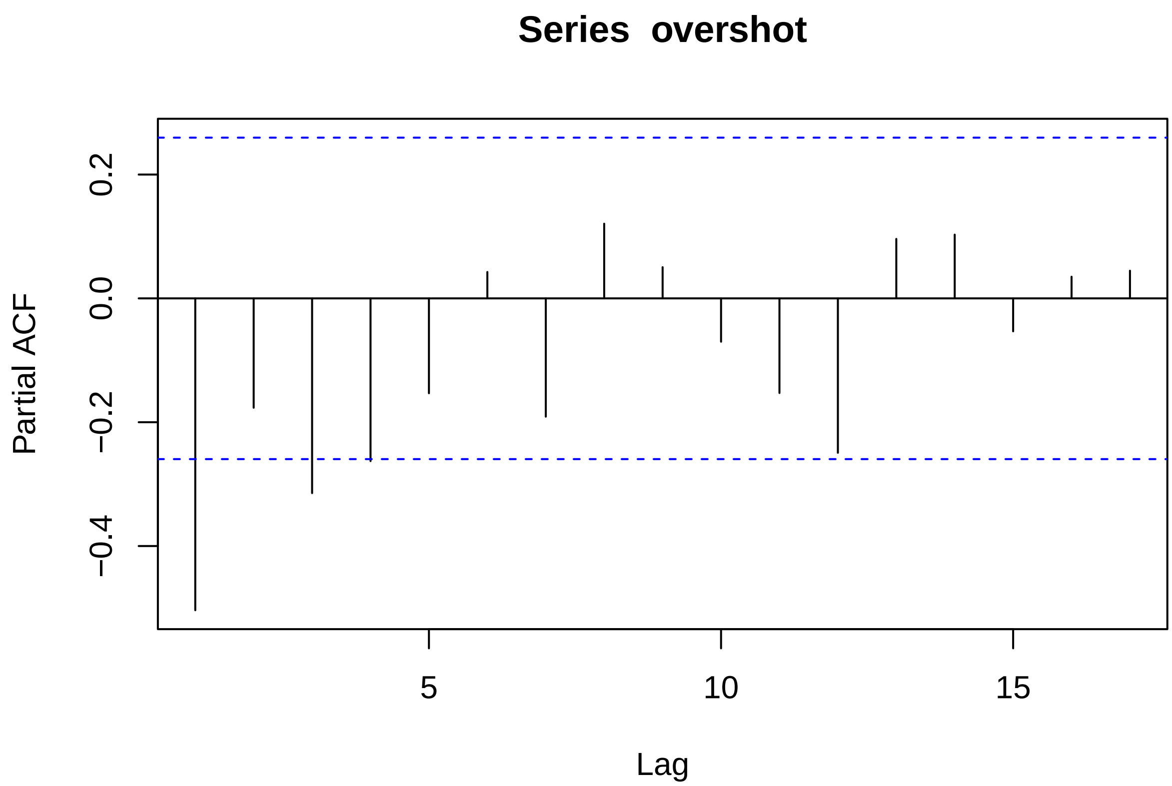
```
overshot<-c(78, -58, 53, -63, 13, -6, -16, -14, 3, -74, 89, -48, -14, 32, 56, -86, -66, 50, 26, 59

plot(overshot,type = 'l')
```



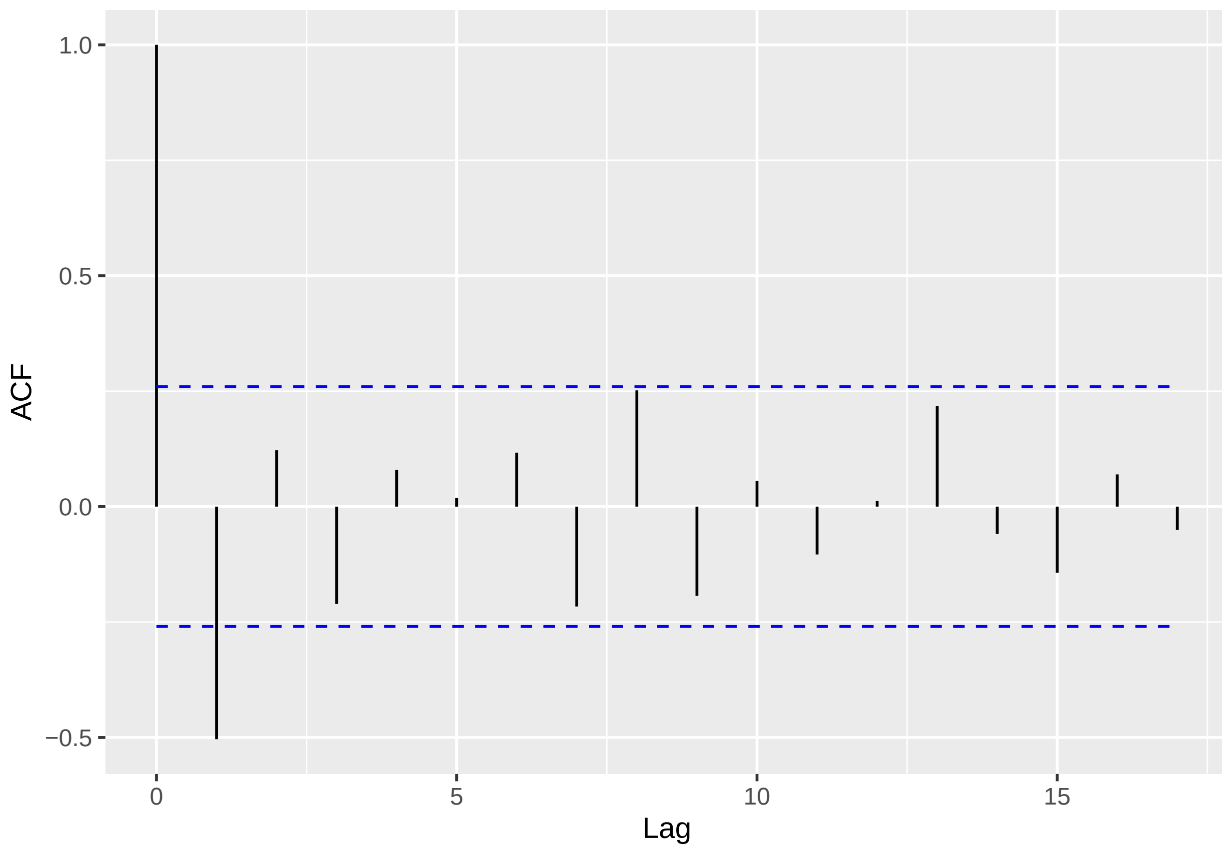

```
## 大致没有看出有非平稳的特征
```

```
acf(overshot, type = "partial")

library(ggfortify)
```

```
## 载入需要的程辑包：ggplot2
```



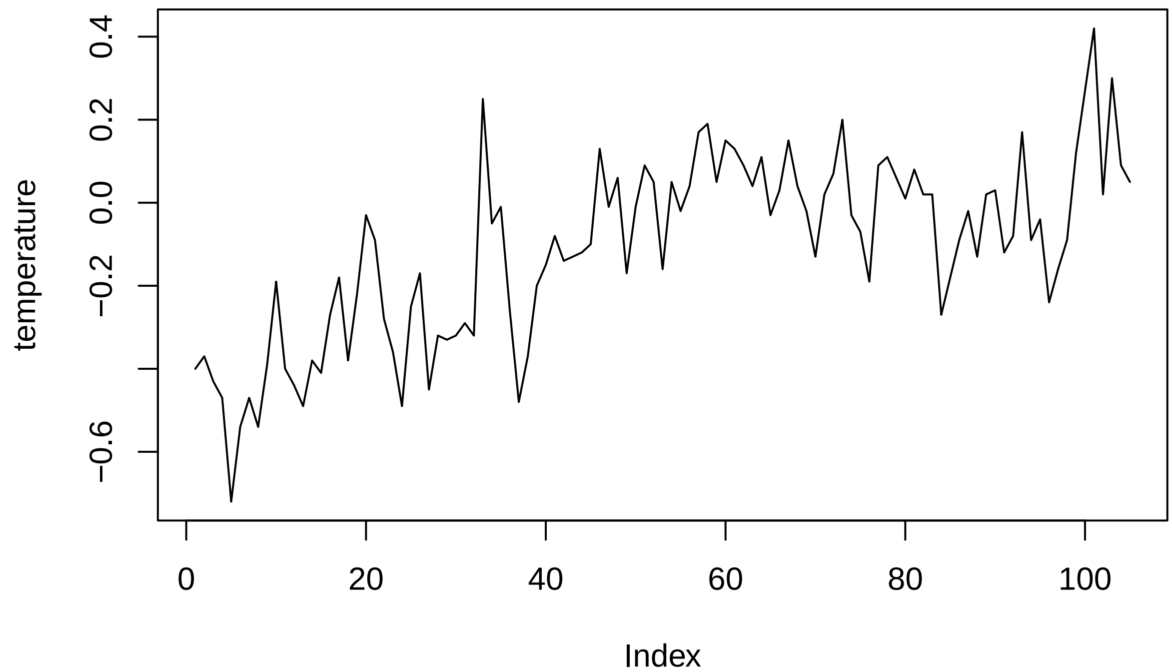

```
autoplot(acf(overshot,type = "correlation",plot = FALSE))
```



粗略确定为 ARMA(2,1)

(2) 自行模拟 P69 例 3-11 并得出书本中的结论

```
temperature<-c(-0.40, -0.37, -0.43, -0.47, -0.72, -0.54, -0.47, -0.54, -0.39, -0.19, -0.40, -0.44,

plot(temperature,type = 'l')
```




```r
## 感觉不太平稳，做 adf 检验具体确定是否平稳

library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

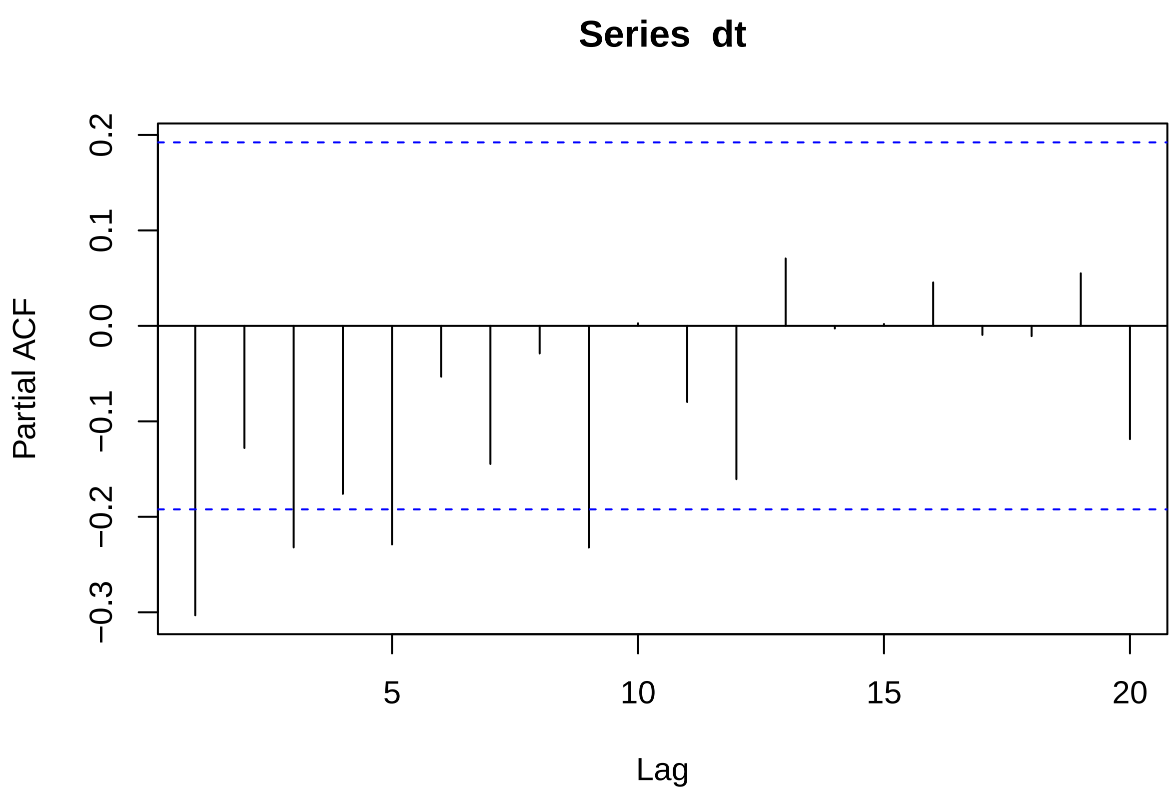
```r
adf.test(temperature)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  temperature
## Dickey-Fuller = -3.3155, Lag order = 4, p-value = 0.07232
## alternative hypothesis: stationary
```

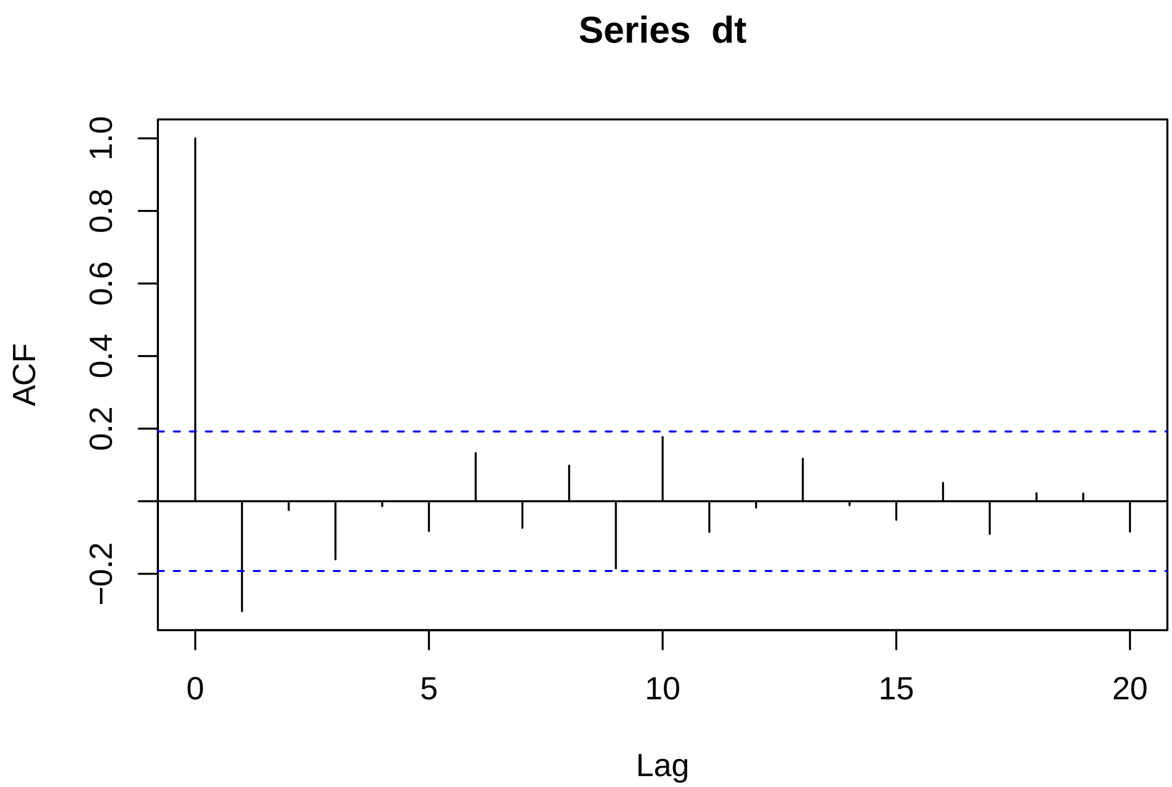
```r
## p 值为 0.07232 大于 0.05，不能拒绝原假设，即存在单位根，序列不平稳。（此时应该考虑做一阶差分）

dt = diff(temperature)
adf.test(dt)
```

```
## Warning in adf.test(dt): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  dt
## Dickey-Fuller = -7.6695, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
## 此时平稳

acf(dt, type = "partial")
```




```
acf(dt,type = "correlation")
```



粗略估计 ARIMA（9,1,1）

习题 2

```
# 从习题 1 中任选一个数据完成下面的任务
# 以第二个（温度）为例：
#（1）绘制样本自相关图，偏相关图
## 见习题 1 两个的图像。

#（2）检验平稳性，纯随机性
adf.test(dt)
```

```
## Warning in adf.test(dt): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  dt
## Dickey-Fuller = -7.6695, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```r
## 数据平稳，下对差分后的数据检验随机性
Box.test(dt,type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  dt
## X-squared = 9.836, df = 1, p-value = 0.001711
```

```r
## 数据不是白噪声序列，存在相关性

#（3）模式识别：定阶，参数估计
library(forecast)
```

```
## Registered S3 methods overwritten by 'forecast':
##   method                 from
##   autoplot.Arima         ggfortify
##   autoplot.acf           ggfortify
##   autoplot.ar            ggfortify
##   autoplot.bats          ggfortify
##   autoplot.decomposed.ts ggfortify
##   autoplot.ets           ggfortify
##   autoplot.forecast      ggfortify
##   autoplot.stl           ggfortify
##   autoplot.ts            ggfortify
##   fitted.ar              ggfortify
##   fortify.ts             ggfortify
##   residuals.ar           ggfortify
```

```r
auto.arima(temperature, ic = c("aicc", "aic", "bic"), stepwise = TRUE, trace = TRUE, allowdrift =
```

```
##
##  ARIMA(2,1,2) with drift         : -110.9594
##  ARIMA(0,1,0) with drift         : -89.45279
##  ARIMA(1,1,0) with drift         : -97.27288
##  ARIMA(0,1,1) with drift         : -105.1543
##  ARIMA(0,1,0)                    : -91.45065
##  ARIMA(1,1,2) with drift         : -111.0789
##  ARIMA(0,1,2) with drift         : -111.3938
##  ARIMA(0,1,3) with drift         : -112.3659
##  ARIMA(1,1,3) with drift         : -110.1192
```

```
##  ARIMA(0,1,4) with drift         : -110.1227
##  ARIMA(1,1,4) with drift         : -109.4136
##  ARIMA(0,1,3)                    : -111.0534
##
##  Best model: ARIMA(0,1,3) with drift
```

```
## Series: temperature
## ARIMA(0,1,3) with drift
##
## Coefficients:
##           ma1      ma2      ma3   drift
##       -0.5167  -0.1440  -0.1739  0.0053
## s.e.   0.0966   0.1083   0.0947  0.0024
##
## sigma^2 = 0.01849:  log likelihood = 61.49
## AIC=-112.98   AICc=-112.37   BIC=-99.76
```

```
## ARIMA(0,1,3)
model <- arima(temperature,order=c(0,1,3))
summary(model)
```

```
##
## Call:
## arima(x = temperature, order = c(0, 1, 3))
##
## Coefficients:
##           ma1      ma2      ma3
##       -0.4783  -0.1239  -0.1598
## s.e.   0.0966   0.1057   0.0921
##
## sigma^2 estimated as 0.01844:  log likelihood = 59.73,  aic = -111.46
##
## Training set error measures:
##                      ME      RMSE      MAE      MPE     MAPE      MASE
## Training set 0.02012645 0.1351466 0.101975 2.928319 123.5084 0.8344138
##                     ACF1
## Training set -0.03043571
```

```
#（4）残差的自相关检验
res <- model$residuals
for(i in 1:3) print(Box.test(res,type="Ljung-Box",lag=6*i))
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 2.8424, df = 6, p-value = 0.8284
##
##
##  Box-Ljung test
##
## data:  res
## X-squared = 7.8589, df = 12, p-value = 0.7961
##
##
##  Box-Ljung test
##
## data:  res
## X-squared = 11.991, df = 18, p-value = 0.8477
```

```
## p 值都远大于 0.05，是白噪声序列。
#（5）预测
forecast(model,h=5)
```

```
##     Point Forecast      Lo 80     Hi 80      Lo 95     Hi 95
## 106     0.04946978 -0.1245582 0.2234977 -0.2166830 0.3156226
## 107     0.06814819 -0.1281392 0.2644356 -0.2320475 0.3683439
## 108     0.07925892 -0.1288778 0.2873957 -0.2390588 0.3975766
## 109     0.07925892 -0.1329588 0.2914767 -0.2453001 0.4038180
## 110     0.07925892 -0.1369629 0.2954807 -0.2514238 0.4099416
```