Homework9

2022-07-04

6.5

```
library(itsmr)
library(tseries)
```

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

```
##
## Attaching package: 'tseries'
```

```
## The following object is masked from 'package:itsmr':
##
##     arma
```

```
adf.test(lake)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  lake
## Dickey-Fuller = -2.7796, Lag order = 4, p-value = 0.254
## alternative hypothesis: stationary
```

```
set.seed(1)
ar1=arima.sim(list(order=c(0,1,0)),n=200); ar1 = ar1[2:201]
adf.test(ar1)
```

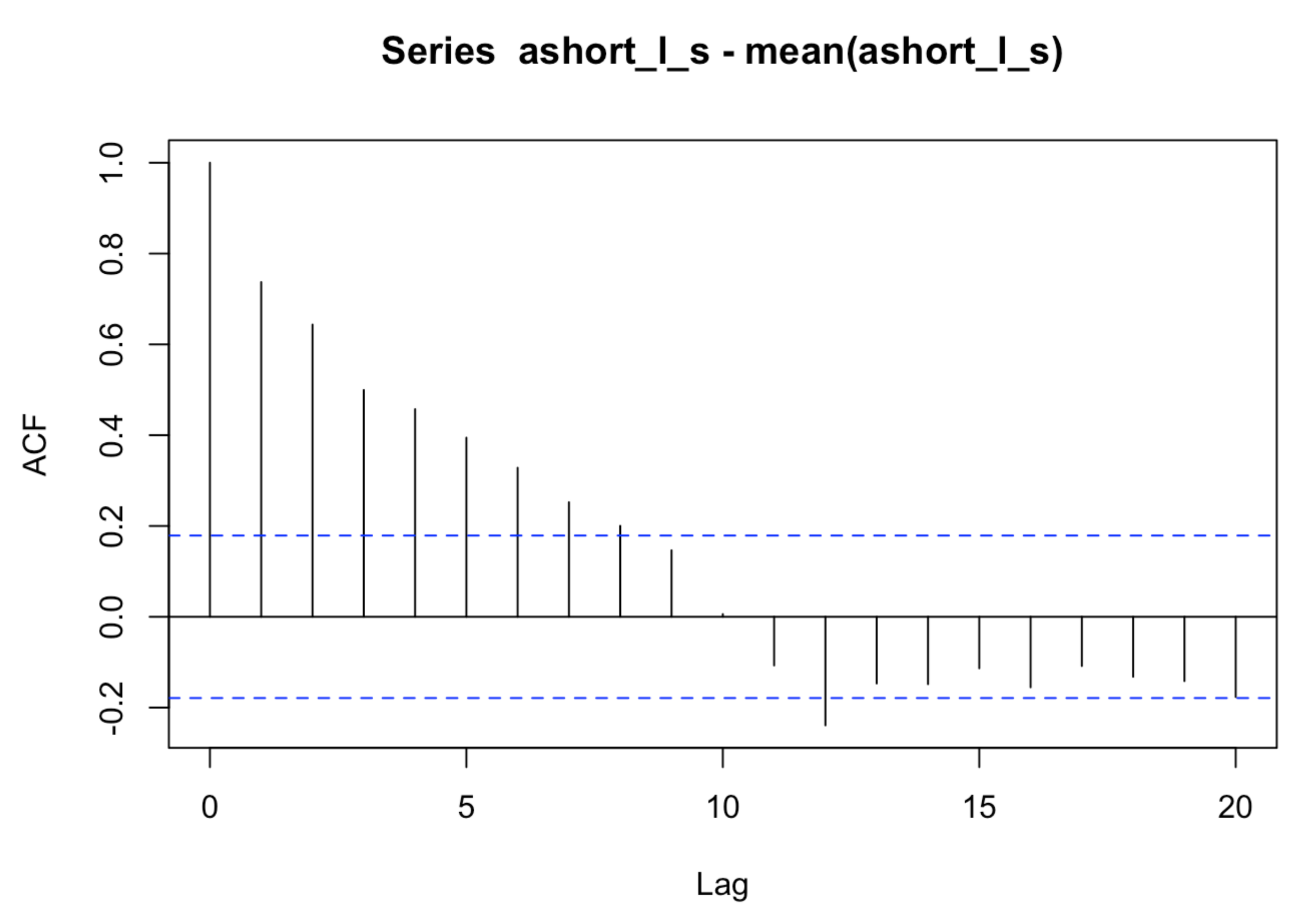
```
##
##  Augmented Dickey-Fuller Test
##
## data:  ar1
## Dickey-Fuller = -1.9512, Lag order = 5, p-value = 0.5965
## alternative hypothesis: stationary
```

```
set.seed(1)
ar2=arima.sim(list(order=c(1,1,0),ar=0.8),n=200); ar2 = ar2[2:201]
adf.test(ar2)
```

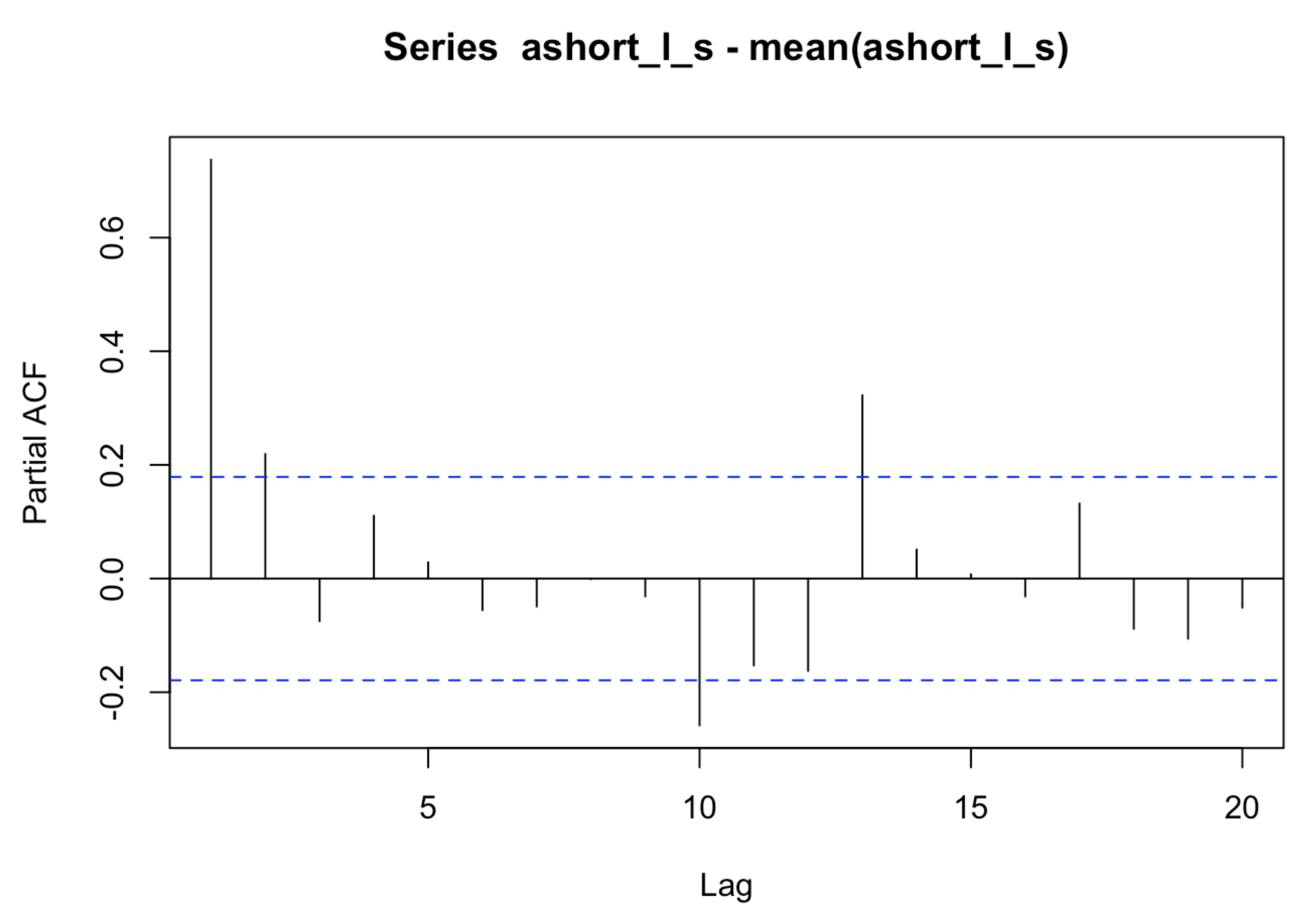
```
##
##  Augmented Dickey-Fuller Test
##
## data:  ar2
## Dickey-Fuller = -2.456, Lag order = 5, p-value = 0.3851
## alternative hypothesis: stationary
```

6.7

```
ashort = airpass[1:(length(airpass)-12)]
ashort_l = log(ashort)
ashort_l_s = rep(NA, length(ashort) - 12)
for (i in 13:length(ashort)){
  ashort_l_s[i-12] = ashort_l[i] - ashort_l[i-12]
}
acf(ashort_l_s - mean(ashort_l_s))
```




```
acf(ashort_l_s - mean(ashort_l_s),type = 'partial')
```




```
ar.13 = arima(ashort_l_s - mean(ashort_l_s),order=c(13,0,0),include.mean=FALSE,transform.pars = FALSE)
ma.12 = arima(ashort_l_s - mean(ashort_l_s),order=c(0,0,12),include.mean=FALSE,transform.pars= FALSE)
ar.13$aic; ma.12$aic
```

```
## [1] -428.9966
```

```
## [1] -438.7975
```

```
ar.13_ = arima(ashort_l_s - mean(ashort_l_s),order=c(13,0,0),include.mean=FALSE,transform.pars = FALSE)
ar.13_$aic
```

```
## [1] -428.9966
```

```
arma.1.12 = arima(ashort_l_s - mean(ashort_l_s),order=c(1,0,12),include.mean=FALSE,transform.pars = FALSE)
arma.1.12$aic
```

```
## [1] -437.8818
```

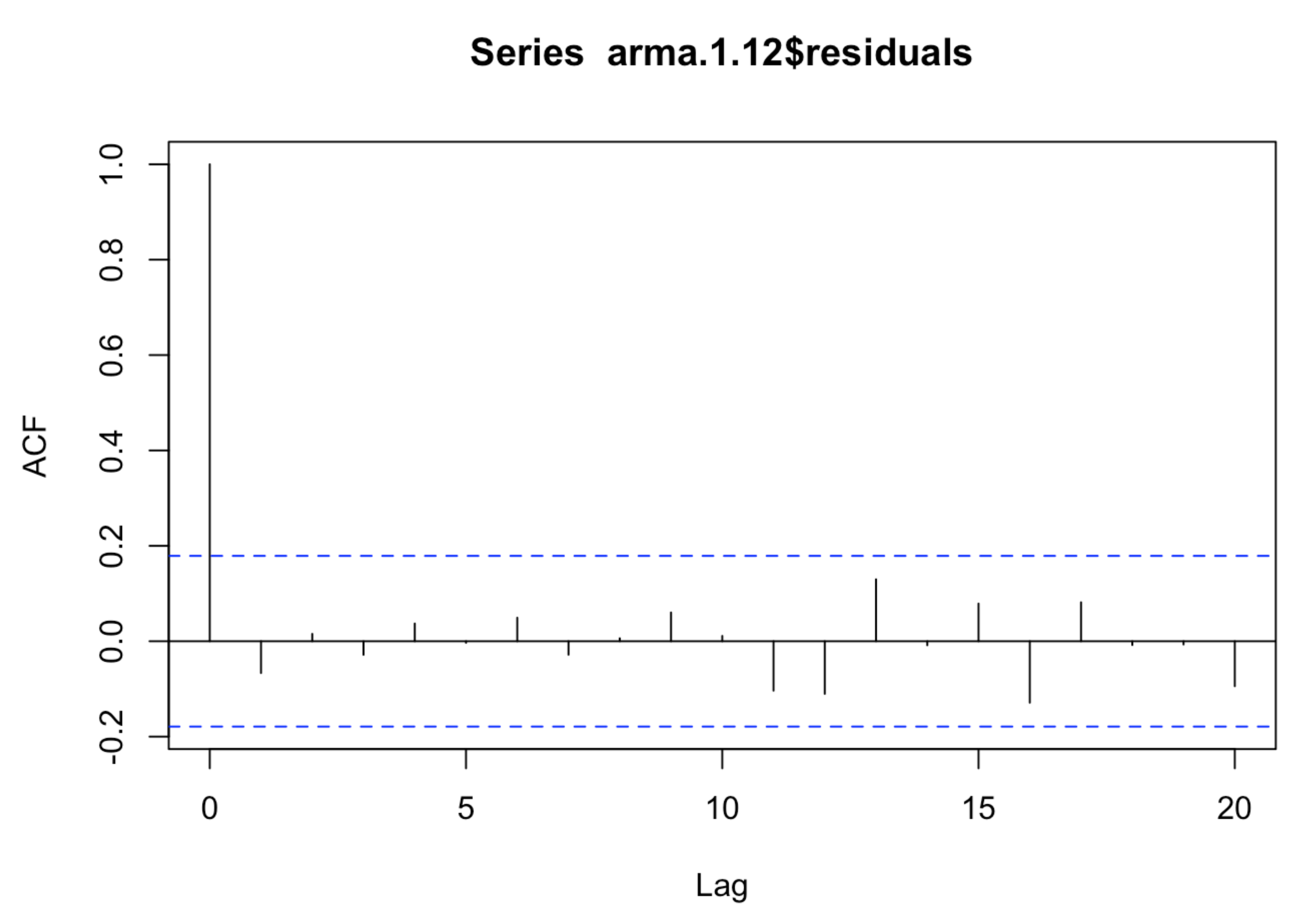
```
arma.1.12
```

```
##
## Call:
## arima(x = ashort_l_s - mean(ashort_l_s), order = c(1, 0, 12), include.mean = FALSE,
##     transform.pars = FALSE)
##
## Coefficients:
##          ar1      ma1     ma2      ma3      ma4     ma5     ma6     ma7
##       0.9253  -0.2897  0.0857  -0.1878  -0.1935  0.0957  0.0016  0.0383
## s.e.  0.0429   0.0954  0.1050   0.0941   0.0929  0.0926  0.0973  0.1071
##           ma8     ma9    ma10    ma11     ma12
##       -0.1228  0.0657  0.0035  0.0996  -0.5962
## s.e.   0.1158  0.1000  0.0824  0.0898   0.0860
##
## sigma^2 estimated as 0.001105:  log likelihood = 232.94,  aic = -437.88
```

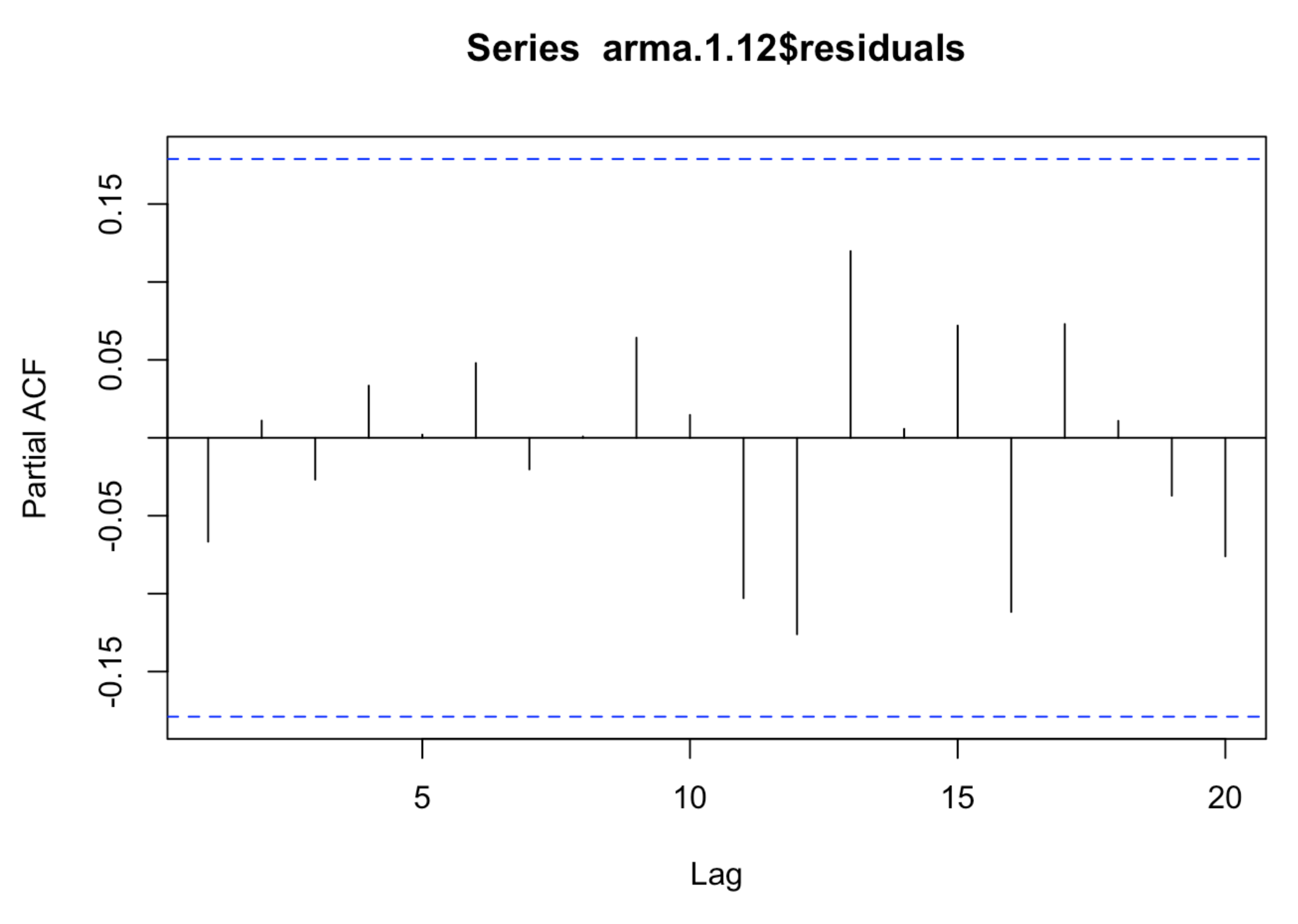
```
coef = arma.1.12$coef[arma.1.12$coef != 0]
sd = sqrt(diag(arma.1.12$var.coef))
paste0('Confidence interval for coefficient ', names(coef), 'is [', coef-1.96*sd, ',', coef+1.96*sd, ']')
```

```
##  [1] "Confidence interval for coefficient ar1is [0.841153195481726,1.00948776954607]"
##  [2] "Confidence interval for coefficient ma1is [-0.476711978746828,-0.102689083095956]"
##  [3] "Confidence interval for coefficient ma2is [-0.120187721832528,0.291566559518715]"
##  [4] "Confidence interval for coefficient ma3is [-0.372147443278466,-0.00337718404092166]"
##  [5] "Confidence interval for coefficient ma4is [-0.375721061605265,-0.011369637154756]"
##  [6] "Confidence interval for coefficient ma5is [-0.0858811255642349,0.27725520698876]"
##  [7] "Confidence interval for coefficient ma6is [-0.189232254126071,0.192365293168107]"
##  [8] "Confidence interval for coefficient ma7is [-0.171557361846856,0.248176567808049]"
##  [9] "Confidence interval for coefficient ma8is [-0.349701241755938,0.104046204801276]"
## [10] "Confidence interval for coefficient ma9is [-0.130215669466621,0.261679359276934]"
## [11] "Confidence interval for coefficient ma10is [-0.157923501808407,0.165013232250334]"
## [12] "Confidence interval for coefficient ma11is [-0.0764699938946508,0.275570350859528]"
## [13] "Confidence interval for coefficient ma12is [-0.764898625041529,-0.427590604703648]"
```

```
acf(arma.1.12$residuals)
```




```
acf(arma.1.12$residuals, type = 'partial')
```




```
Box.test(arma.1.12$residuals, lag = 12, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  arma.1.12$residuals
## X-squared = 4.8601, df = 12, p-value = 0.9625
```

```
pred_l = predict(arma.1.12,12)$pred + mean(ashort_l_s) + ashort_l[(length(ashort_l)-11)]
```

```
## Warning in predict.Arima(arma.1.12, 12): MA part of model is not invertible
```

```
L_l = pred_l - 1.96*sqrt(arma.1.12$sigma2)
R_l = pred_l + 1.96*sqrt(arma.1.12$sigma2)

Results <- matrix(NA, 12, 5)
for (i in 1:12){
  Results[i,1] = exp(pred_l[i])
  Results[i,2] = exp(L_l[i])
  Results[i,3] = exp(R_l[i])
  Results[i,4] = airpass[length(airpass)-12+i]
  Results[i,5] = Results[i,4] - Results[i,1]
}
colnames(Results) <- c('Prediction', 'Lower bound', 'Upper bound', 'True Value', 'Error')
round(Results, 5)
```

```
##       Prediction Lower bound Upper bound True Value     Error
##  [1,]   427.6590    400.6849    456.4489        417 -10.65899
##  [2,]   427.2974    400.3462    456.0630        391 -36.29742
##  [3,]   417.5006    391.1673    445.6067        419   1.49938
##  [4,]   419.1472    392.7101    447.3642        461  41.85275
##  [5,]   414.5098    388.3651    442.4146        472  57.49019
##  [6,]   428.6620    401.6246    457.5194        535 106.33803
##  [7,]   423.8042    397.0733    452.3347        622 198.19578
##  [8,]   418.3690    391.9809    446.5336        606 187.63096
##  [9,]   417.4137    391.0859    445.5140        508  90.58625
## [10,]   422.9987    396.3186    451.4749        461  38.00128
## [11,]   414.2840    388.1536    442.1736        390 -24.28404
## [12,]   409.6471    383.8091    437.2245        432  22.35293
```