Time Series Project 3

Congyu Hang

2023-11-28

Practice

For this part, you will work with the *gas furnace data*, a classical time series[1] on industrial/chemical process control. The data consists of two related series:

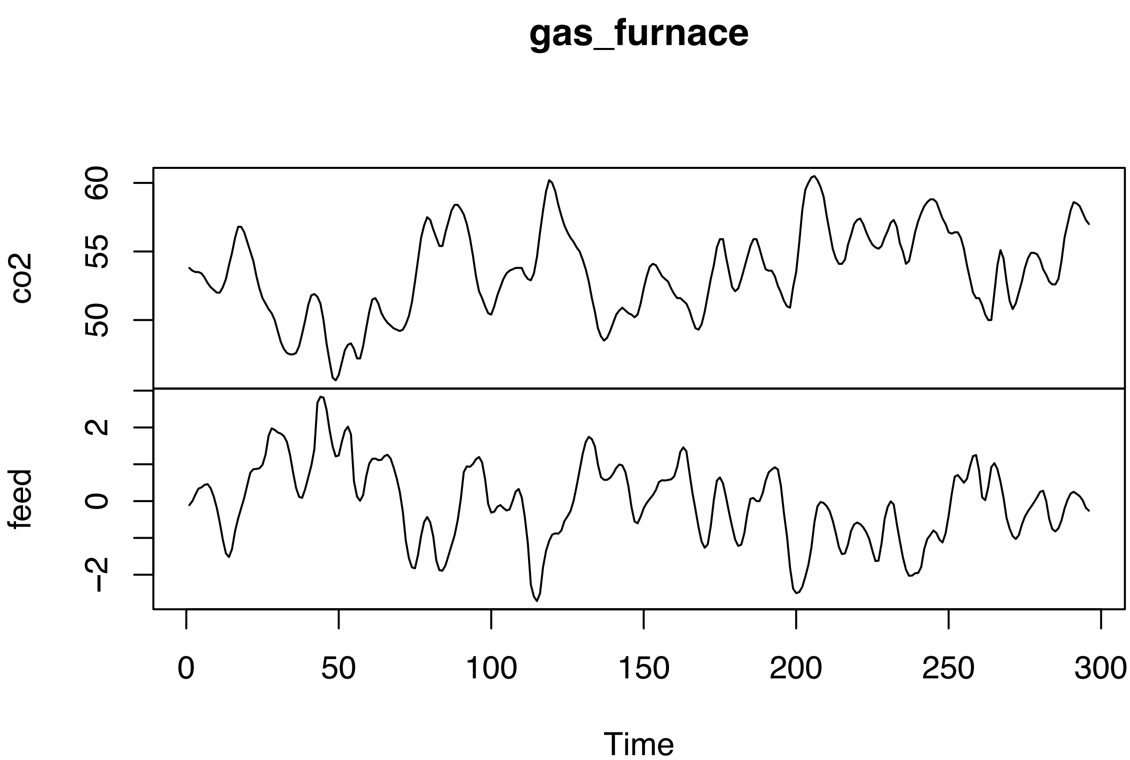
- `feed` = input feed rate of methane in gas furnace
- `co2` = output concentration of CO2 from gas furnace

You can load and view the data with:

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.3.2
```

```
library(astsa)
load( "gas_furnace.RData")
plot(gas_furnace)
```



The purpose of the analysis is to look at the relationship between the feed rate and the output CO2.

[1] see Time Series Analysis: Forecasting and Control, 4Ed, by Box, Jenkins, and Reinsel, chapter 12

1. [6 marks] Fit a *univariate* ARMA(p, q) model to `co2` *alone*. Select the model using AIC with `forecast::auto.arima` (without differencing). Report the fitted model specification, the residual standard deviation, the model diagnostics (residual plot, ACF, and Q-Q plot), and plot the cross-correlation between the model's residuals and `feed`, and comment.

```r
library(forecast)
```

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

##
## Attaching package: 'forecast'

## The following object is masked from 'package:astsa':
##
##     gas
```

```r
co2 = gas_furnace[,1]
co2_fit = forecast::auto.arima(co2,d=0,D=0)
summary(co2_fit)
```

```
## Series: co2
## ARIMA(3,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ma1     ma2     mean
##       2.2096  -1.7648  0.5271  -0.1103  0.2855  53.6484
## s.e.  0.0985   0.1743  0.0843   0.0986  0.0796   0.7775
##
## sigma^2 = 0.111:  log likelihood = -94.84
## AIC=203.69   AICc=204.07   BIC=229.52
##
## Training set error measures:
##                         ME      RMSE       MAE          MPE      MAPE      MASE
## Training set -0.0005713264 0.3296994 0.2412947 -0.005178562 0.4515254 0.4026128
##                    ACF1
## Training set 0.01609169
```

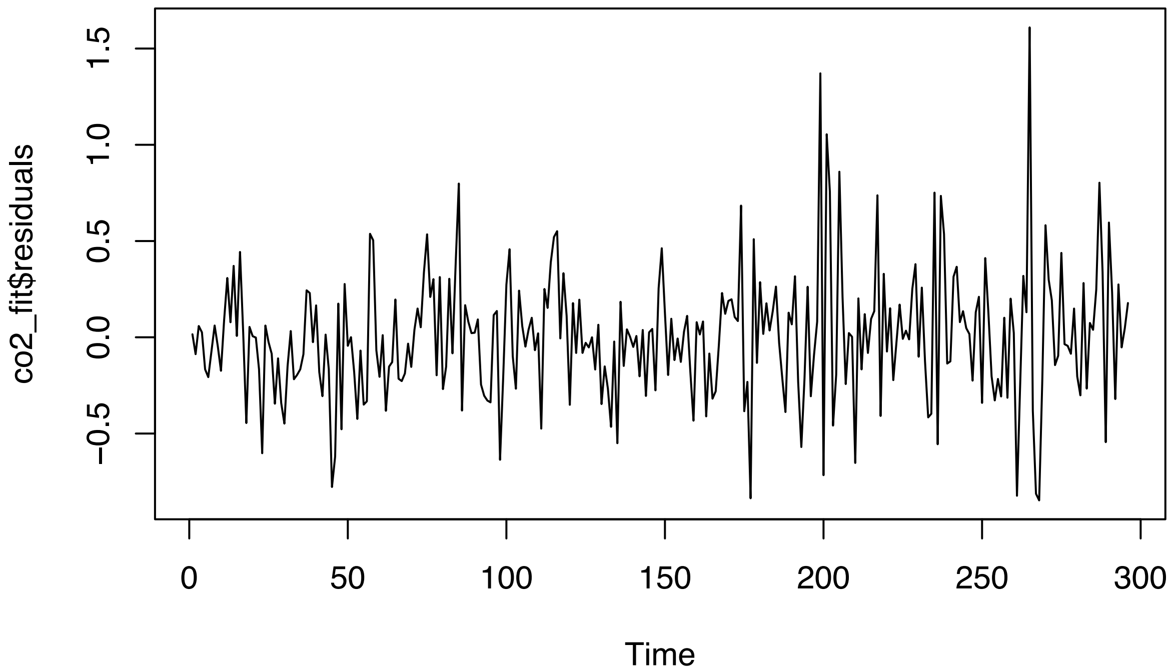
The model is ARIMA(2,1,2), with 1 order differencing, phi1 = 1.3392, phi2 = -0.6443, theta1 = -0.1847, theta2 = 0.2615

Residual Standard Deviation

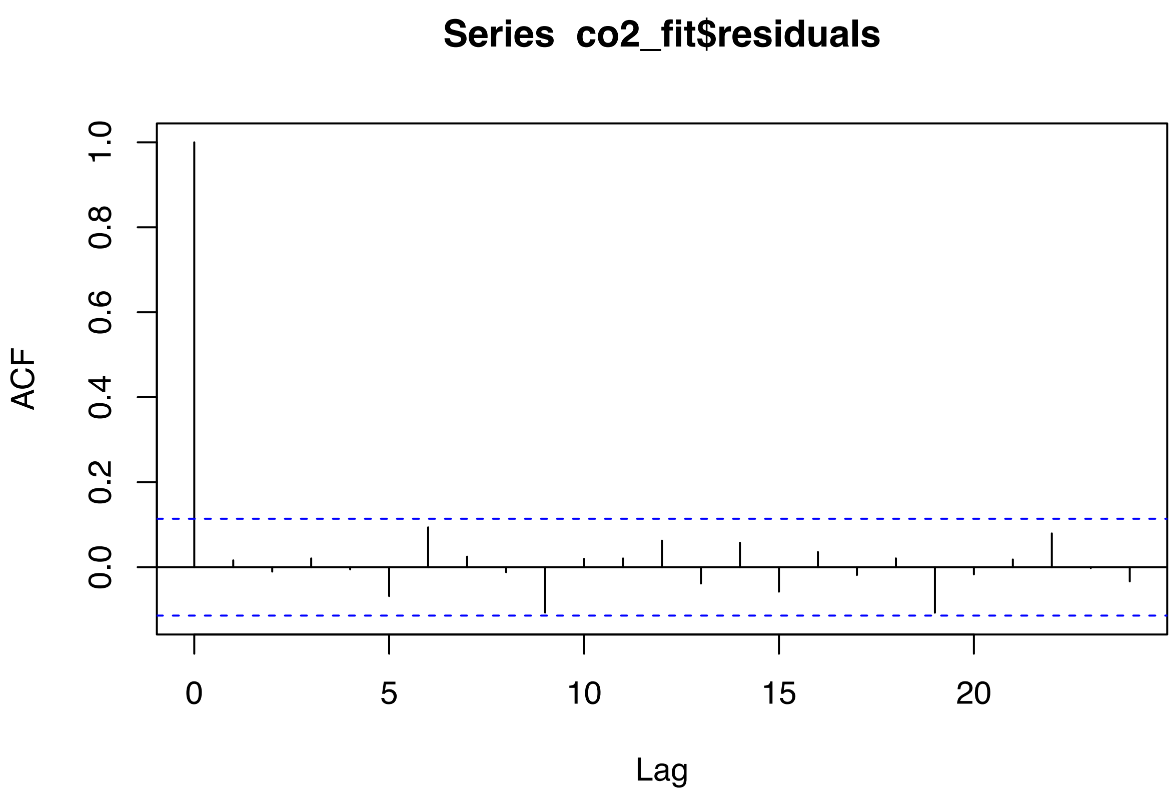
```r
sqrt(co2_fit$sigma2)
```

```
## [1] 0.3330926
```

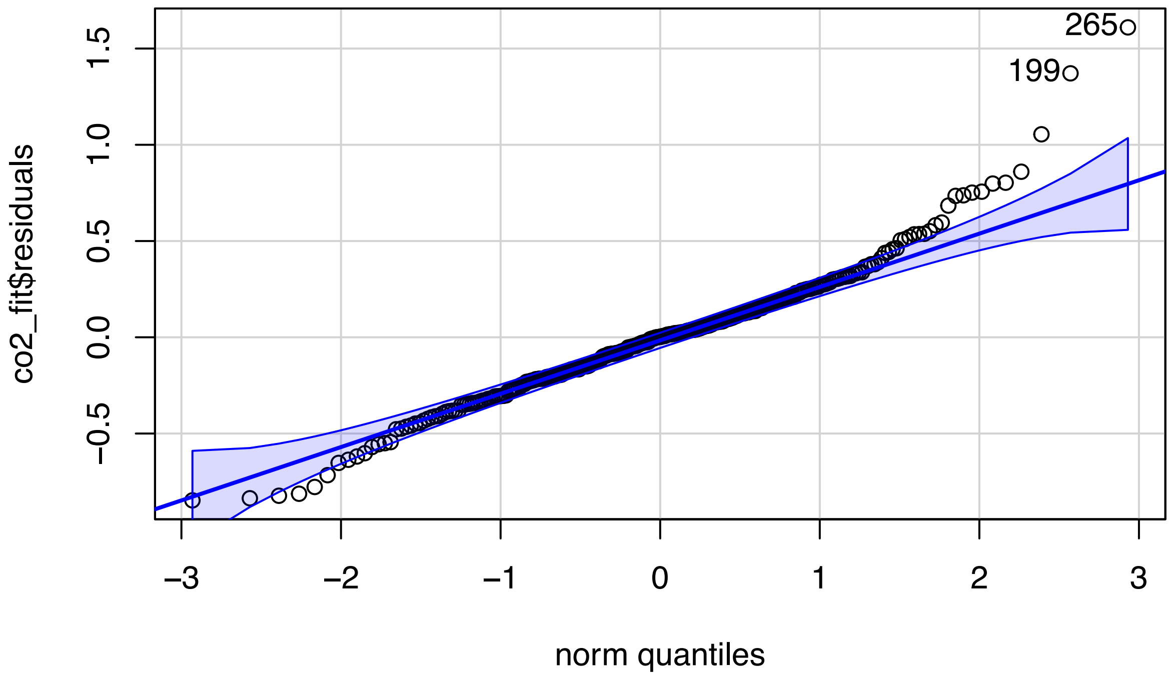
```r
plot(co2_fit$residuals)
```




```r
acf(co2_fit$residuals)
```




```r
library(car)
qqPlot(co2_fit$residuals)
```

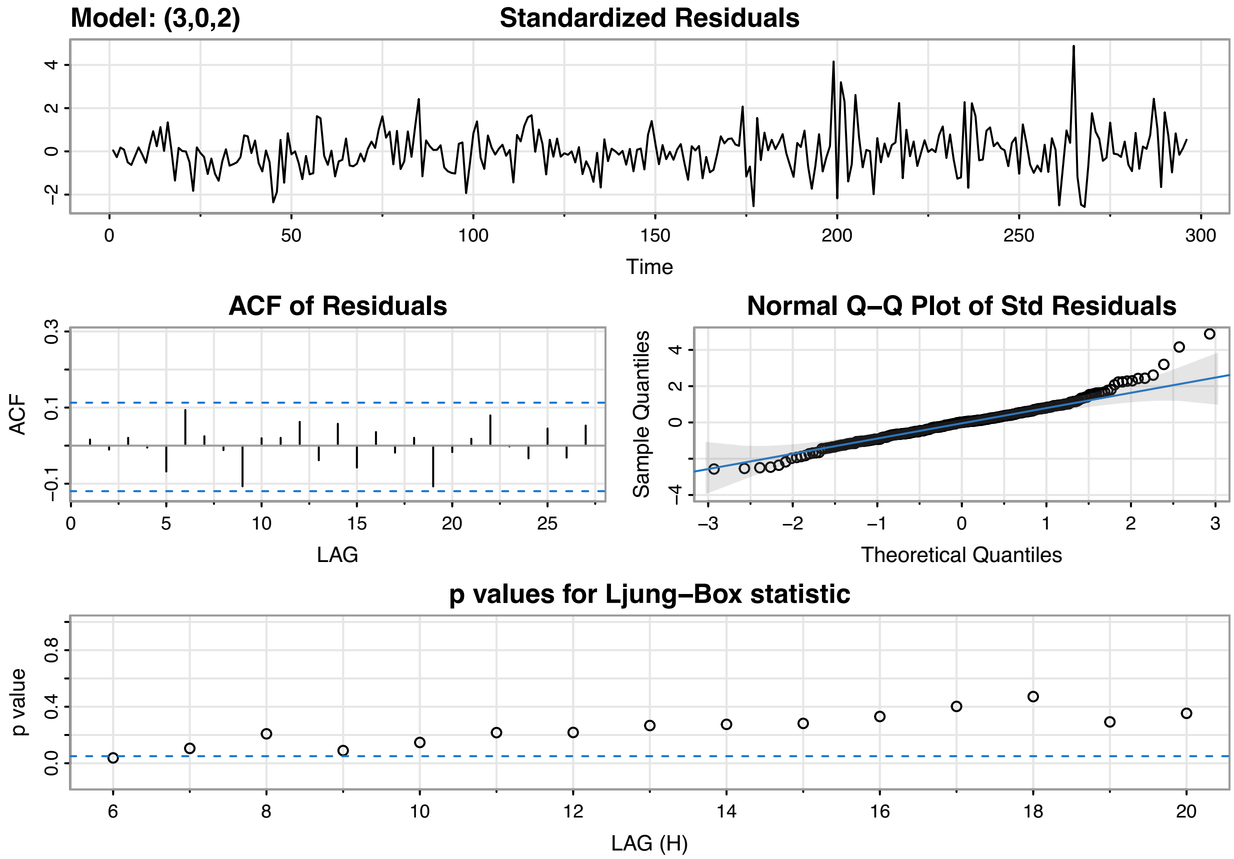



```
## [1] 265 199
```

```
sarima(co2,3,0,2)
```

```
## initial  value 1.167199
## iter   2 value 0.708622
## iter   3 value 0.568430
## iter   4 value 0.429354
## iter   5 value 0.285406
## iter   6 value -0.034695
## iter   7 value -0.549118
## iter   8 value -0.593887
## iter   9 value -0.638304
## iter  10 value -0.660178
## iter  11 value -0.693721
## iter  12 value -0.716601
## iter  13 value -0.793709
## iter  14 value -0.830912
## iter  15 value -0.881641
## iter  16 value -0.915006
## iter  17 value -1.007848
## iter  18 value -1.028141
## iter  19 value -1.038382
## iter  20 value -1.041285
## iter  21 value -1.042544
## iter  22 value -1.048434
## iter  23 value -1.062420
## iter  24 value -1.070861
## iter  25 value -1.088927
## iter  26 value -1.093388
## iter  27 value -1.094719
## iter  28 value -1.097361
## iter  29 value -1.099220
```

```
## iter  30 value -1.099544
## iter  31 value -1.100743
## iter  32 value -1.102198
## iter  33 value -1.103349
## iter  34 value -1.104407
## iter  35 value -1.104636
## iter  36 value -1.104649
## iter  37 value -1.104657
## iter  38 value -1.104677
## iter  39 value -1.104707
## iter  40 value -1.104748
## iter  41 value -1.104762
## iter  42 value -1.104768
## iter  43 value -1.104776
## iter  44 value -1.104776
## iter  45 value -1.104778
## iter  46 value -1.104781
## iter  47 value -1.104785
## iter  48 value -1.104789
## iter  49 value -1.104790
## iter  50 value -1.104791
## iter  51 value -1.104791
## iter  52 value -1.104791
## iter  53 value -1.104792
## iter  54 value -1.104792
## iter  55 value -1.104792
## iter  56 value -1.104792
## iter  57 value -1.104792
## iter  57 value -1.104792
## iter  57 value -1.104792
## final  value -1.104792
## converged
## initial  value -1.098468
## iter   2 value -1.098474
## iter   3 value -1.098517
## iter   4 value -1.098519
## iter   5 value -1.098523
## iter   6 value -1.098524
## iter   7 value -1.098524
## iter   8 value -1.098524
## iter   9 value -1.098524
## iter  10 value -1.098524
## iter  11 value -1.098524
## iter  12 value -1.098524
## iter  13 value -1.098525
## iter  14 value -1.098525
## iter  15 value -1.098525
## iter  15 value -1.098525
## iter  15 value -1.098525
## final  value -1.098525
## converged
```




```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2     ar3      ma1     ma2    xmean
##       2.2096  -1.7648  0.5271  -0.1103  0.2855  53.6484
## s.e.  0.0985   0.1743  0.0843   0.0986  0.0796   0.7775
##
## sigma^2 estimated as 0.1087:  log likelihood = -94.84,  aic = 203.69
##
## $degrees_of_freedom
## [1] 290
##
## $ttable
##       Estimate     SE  t.value p.value
## ar1     2.2096 0.0985  22.4214  0.0000
## ar2    -1.7648 0.1743 -10.1271  0.0000
## ar3     0.5271 0.0843   6.2537  0.0000
## ma1    -0.1103 0.0986  -1.1190  0.2641
## ma2     0.2855 0.0796   3.5879  0.0004
## xmean  53.6484 0.7775  69.0015  0.0000
##
```

```
## $AIC
## [1] 0.688125
##
## $AICc
## [1] 0.689107
##
## $BIC
## [1] 0.775397
```

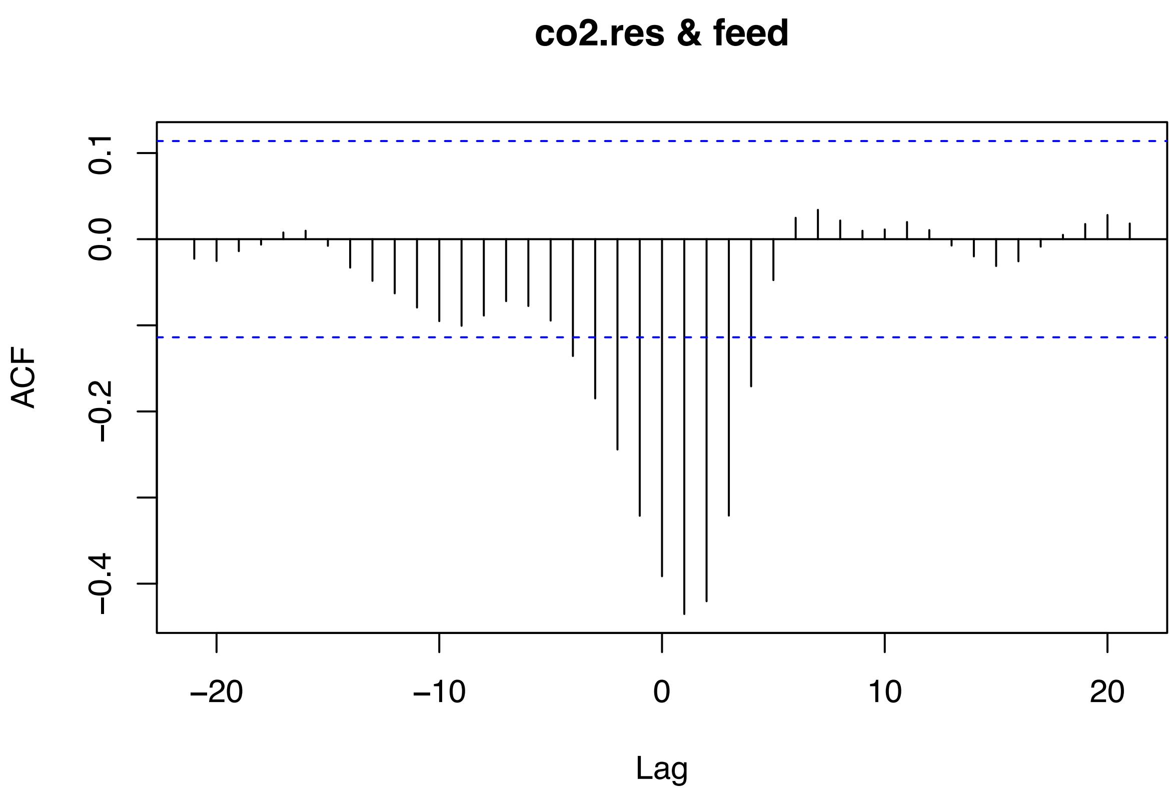
Comment on diagnostics:

The residuals are distributed around 0 with unequal variances, acf cuts off so residuals are uncorrelated and the p-values are above 0.05 so the null hypothesis is not rejected, however, the qqplot of residuals seems to be a bit off from normal.

cross-correlation

```r
feed = gas_furnace[,2]
co2.res = co2_fit$residuals
```

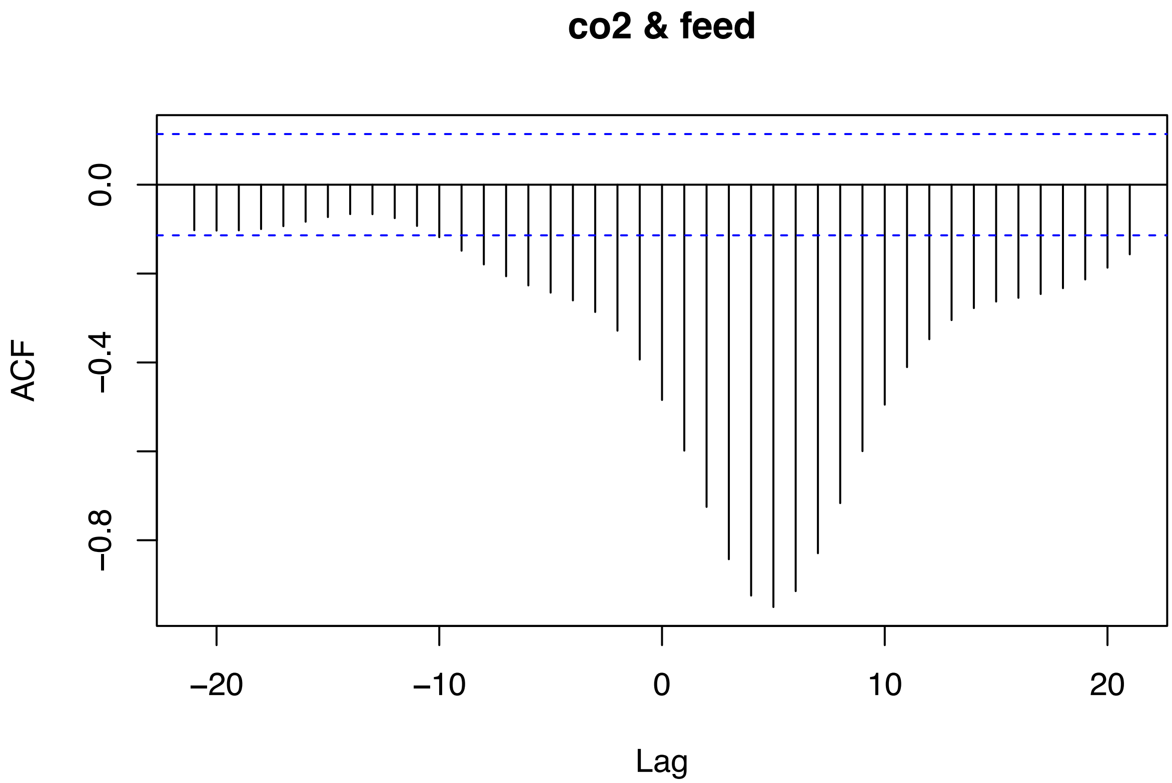
```r
ccf(co2.res,feed)
```



The acf of residuals of co2 model and feed seem to be highly correlated at some of the lags, this indicate that some information is not captured by this model and this model is not good enough.

2. [6 marks] Plot the cross-correlation between `co2` and `feed`, and identify a lead-lag relationship between the variables. Fit a time series regression model with ARMA errors, with some lagged value of `feed` of your choice as external regressor, and select the order of the model using `auto.arima` (without differencing). Report the fitted model specification, the residual standard deviation, the model diagnostics (residual plot, ACF, and Q-Q plot), and plot the cross-correlation between the model's residuals and `feed`, and comment.

```r
ccf(co2,feed)
```




```r
co2_feed = ts.intersect(co2 = co2, feed.l = stats::lag(feed,-5))
(xreg.fit = auto.arima(co2_feed[,1],xreg = co2_feed[,2],d=0,D=0))
```

```
## Series: co2_feed[, 1]
## Regression with ARIMA(1,0,4) errors
##
## Coefficients:
##          ar1     ma1     ma2     ma3     ma4  intercept     xreg
##       0.8926  0.9428  0.8213  0.6054  0.3080    53.5626  -0.9085
## s.e.  0.0287  0.0727  0.1024  0.0924  0.0614     0.5967   0.1349
##
## sigma^2 = 0.09646:  log likelihood = -71.74
## AIC=159.48   AICc=159.99   BIC=188.87
```

```r
sqrt(xreg.fit$sigma2)
```

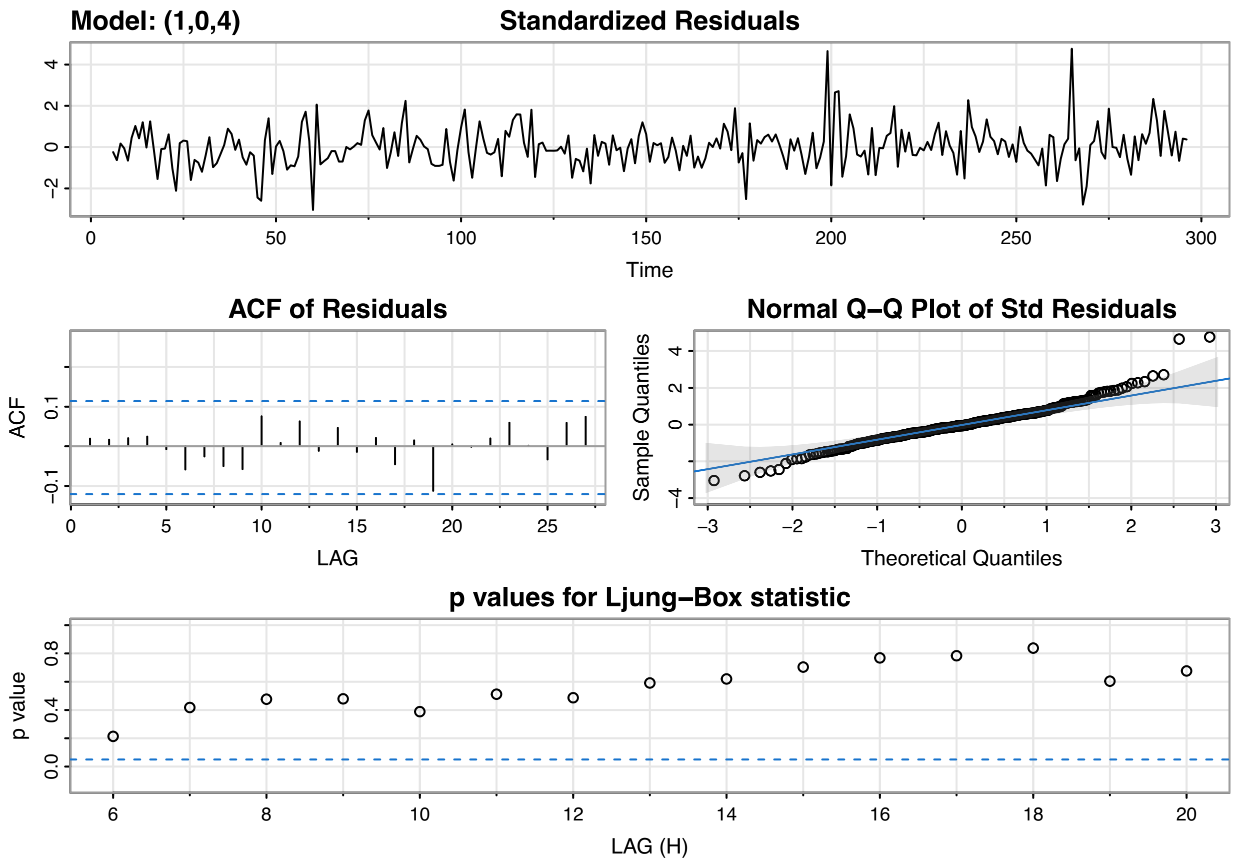
```
## [1] 0.3105876
```

```r
sarima(co2_feed[,1],p=1,d=0,q=4,xreg = co2_feed[,2])
```

```
## initial  value 0.002346
## iter   2 value -0.795477
## iter   3 value -0.859803
## iter   4 value -0.884601
## iter   5 value -0.919222
## iter   6 value -0.957512
## iter   7 value -1.075789
## iter   8 value -1.115811
## iter   9 value -1.122203
```

```
## iter  10 value -1.160874
## iter  11 value -1.169718
## iter  12 value -1.173832
## iter  13 value -1.177881
## iter  14 value -1.178246
## iter  15 value -1.178508
## iter  16 value -1.178656
## iter  17 value -1.178701
## iter  18 value -1.178707
## iter  19 value -1.178718
## iter  20 value -1.178724
## iter  21 value -1.178727
## iter  22 value -1.178729
## iter  23 value -1.178733
## iter  24 value -1.178739
## iter  25 value -1.178739
## iter  25 value -1.178739
## iter  25 value -1.178739
## final  value -1.178739
## converged
## initial  value -1.172279
## iter   2 value -1.172314
## iter   3 value -1.172338
## iter   4 value -1.172353
## iter   5 value -1.172357
## iter   6 value -1.172368
## iter   7 value -1.172369
## iter   8 value -1.172371
## iter   9 value -1.172373
## iter  10 value -1.172376
## iter  11 value -1.172378
## iter  12 value -1.172379
## iter  13 value -1.172380
## iter  14 value -1.172381
## iter  15 value -1.172383
## iter  16 value -1.172387
## iter  17 value -1.172391
## iter  18 value -1.172392
## iter  19 value -1.172392
## iter  20 value -1.172392
## iter  21 value -1.172393
## iter  22 value -1.172394
## iter  23 value -1.172394
## iter  24 value -1.172395
## iter  25 value -1.172397
## iter  26 value -1.172399
## iter  27 value -1.172400
## iter  28 value -1.172400
## iter  29 value -1.172401
## iter  30 value -1.172402
## iter  31 value -1.172404
## iter  32 value -1.172404
## iter  33 value -1.172404
## iter  34 value -1.172404
```

```
## iter  35 value -1.172404
## iter  36 value -1.172405
## iter  37 value -1.172405
## iter  37 value -1.172405
## iter  37 value -1.172405
## final  value -1.172405
## converged
```



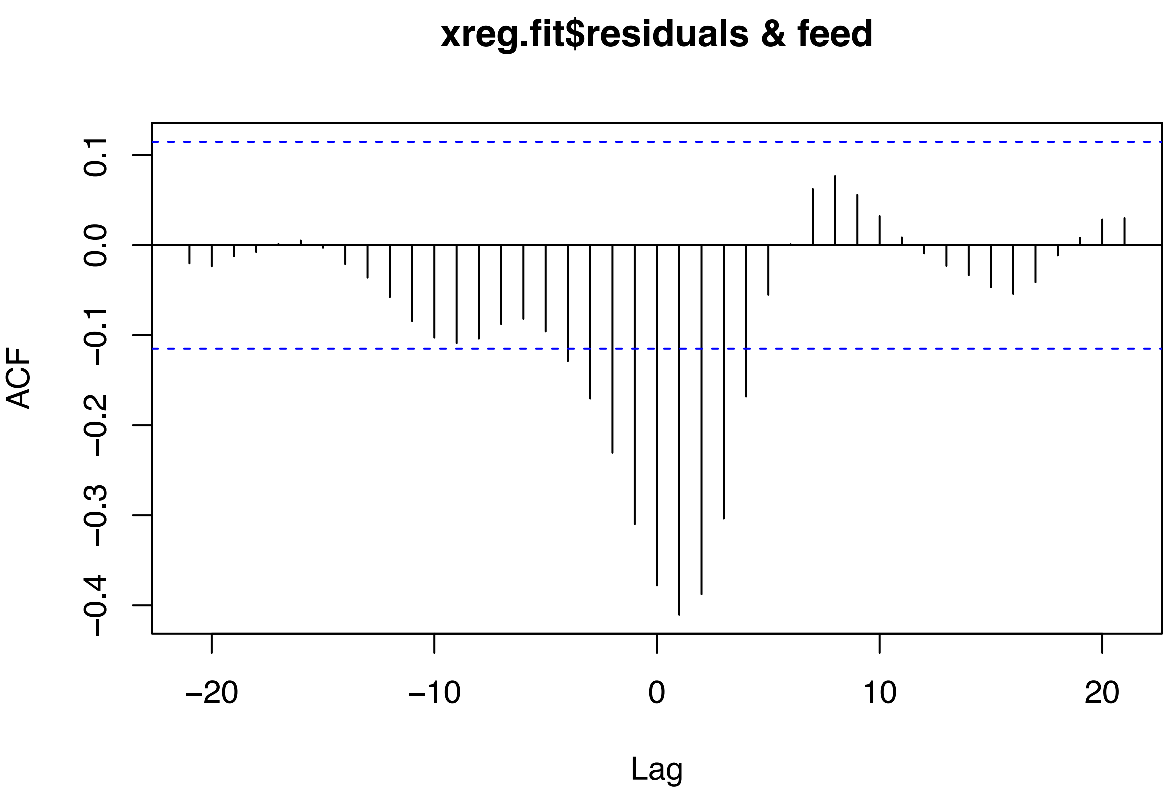

```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xreg, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1     ma1     ma2     ma3     ma4  intercept     xreg
##       0.8926  0.9428  0.8213  0.6054  0.3080    53.5626  -0.9085
## s.e.  0.0287  0.0727  0.1024  0.0924  0.0614     0.5967   0.1349
##
## sigma^2 estimated as 0.09414:  log likelihood = -71.74,  aic = 159.48
##
## $degrees_of_freedom
## [1] 284
##
## $ttable
##           Estimate     SE t.value p.value
```

```
## ar1          0.8926 0.0287 31.0587       0
## ma1          0.9428 0.0727 12.9679       0
## ma2          0.8213 0.1024  8.0182       0
## ma3          0.6054 0.0924  6.5532       0
## ma4          0.3080 0.0614  5.0120       0
## intercept   53.5626 0.5967 89.7599       0
## xreg        -0.9085 0.1349 -6.7364       0
##
## $AIC
## [1] 0.5480507
##
## $AICc
## [1] 0.5494107
##
## $BIC
## [1] 0.6490356
```

Comment on diagnostics:

The residuals are distributed around 0 with unequal variances, acf cuts off so residuals are uncorrelated and the p-values are above 0.05 so the null hypothesis is not rejected, however, the qqplot of residuals seems to be a bit off from normal because of some outliers.

```
ccf(xreg.fit$residuals,feed)
```



The ccf of residuals of co2 model and feed seem to be highly correlated at some of the lags, this indicate that some information is not captured by this model and this model is not good enough.

3. [6 marks] A possible candidate for the transfer function $\beta(B) = \frac{\delta(B)B^d}{\omega(B)}$ is given by the following model

$$Y_t = \frac{(\omega_0 + \omega_1 B + \omega_2 B^2)B^3}{1 - \delta_1 B - \delta_2 B^2} X_t + E_t$$

where E_t follows an ARMA process. Fit this model using time series regression with ARMA errors, where the exogenous variables are lagged values of both the input and output series. You have to identify the appropriate lags of `co2` and `feed` to include in the model, and choose the error ARMA order using `auto.arima` (without differencing). Report the fitted model specification, the residual standard deviation, the model diagnostics (residual plot, ACF, and Q-Q plot), and plot the cross-correlation between the model's residuals and `feed`, and comment.

```
cf.intersect =ts.intersect(co2,co2_l1 = stats::lag(co2,-1),co2_l2 = stats::lag(co2,-2),feed_l3 = stats:
```

```
fit.order = auto.arima(cf.intersect[,1],xreg = cf.intersect[,2:6],d=0,D=0)
summary(fit.order)
```

```
## Series: cf.intersect[, 1]
## Regression with ARIMA(1,0,2) errors
##
## Coefficients:
##          ar1     ma1     ma2  intercept  co2_l1   co2_l2  feed_l3  feed_l4
##       0.7400  0.0691  0.2036    21.3088  0.7311  -0.1303  -0.5392  -0.2758
## s.e.  0.0708  0.1604  0.0713     3.1347  0.1572   0.1085   0.0740   0.1284
##       feed_l5
##       -0.4134
## s.e.   0.1415
##
## sigma^2 = 0.05787:  log likelihood = 5.67
## AIC=8.66   AICc=9.44   BIC=45.39
##
## Training set error measures:
##                        ME      RMSE       MAE          MPE      MAPE      MASE
## Training set 0.0007991579 0.2368184 0.1662971 -0.001029288 0.3094241 0.2738566
##                     ACF1
## Training set 0.009290362
```

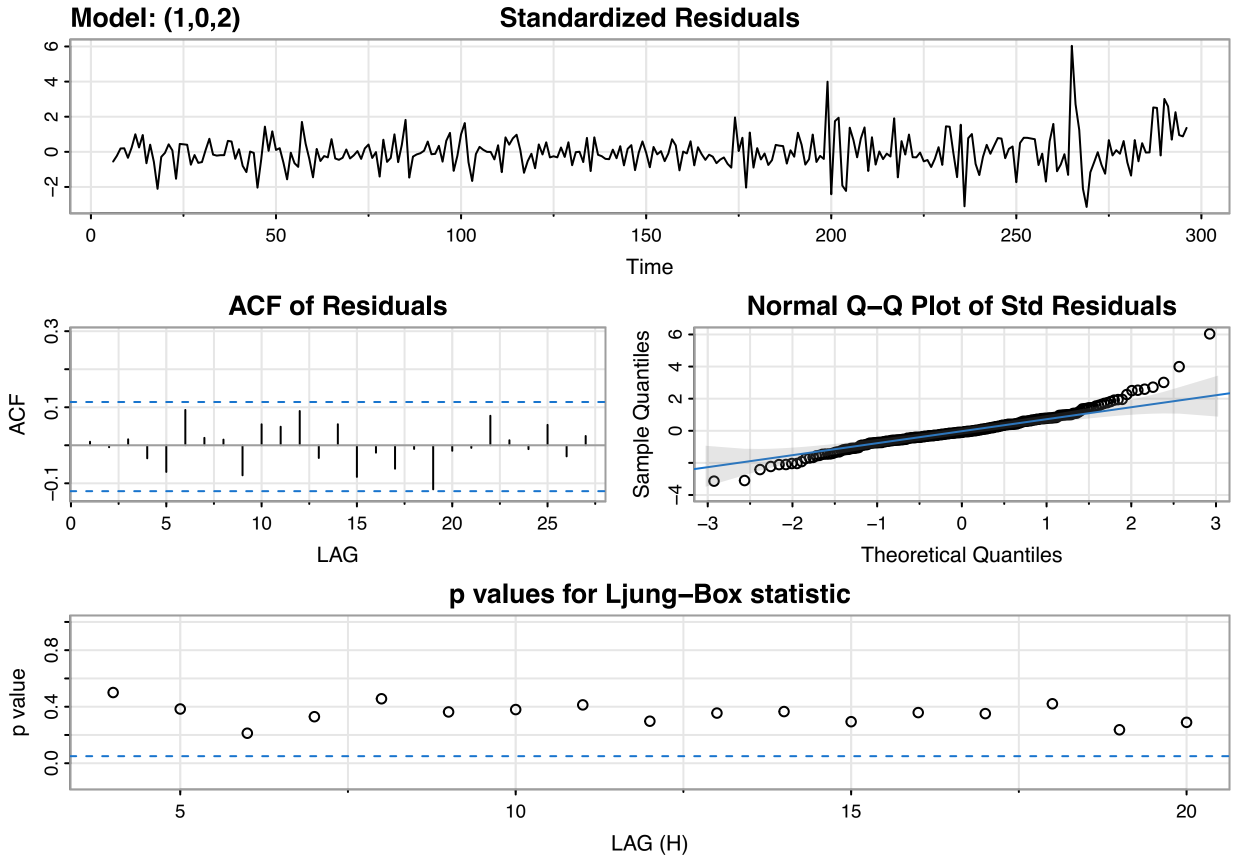
```
sqrt(fit.order$sigma2)
```

```
## [1] 0.2405678
```

```
sarima(cf.intersect[,1],xreg = cf.intersect[,2:6],1,0,2)
```

```
## initial  value -1.394333
## iter   2 value -1.397507
## iter   3 value -1.403599
## iter   4 value -1.406324
## iter   5 value -1.411289
## iter   6 value -1.413313
## iter   7 value -1.414337
## iter   8 value -1.415110
## iter   9 value -1.421112
## iter  10 value -1.421235
## iter  11 value -1.424630
## iter  12 value -1.426385
## iter  13 value -1.428776
## iter  14 value -1.429879
## iter  15 value -1.431490
## iter  16 value -1.432265
## iter  17 value -1.434849
## iter  18 value -1.436330
```

```
## iter  19 value -1.437017
## iter  20 value -1.437547
## iter  21 value -1.437967
## iter  22 value -1.438553
## iter  23 value -1.438832
## iter  24 value -1.438957
## iter  25 value -1.439062
## iter  26 value -1.439195
## iter  27 value -1.439252
## iter  28 value -1.439262
## iter  29 value -1.439262
## iter  30 value -1.439262
## iter  31 value -1.439263
## iter  32 value -1.439264
## iter  33 value -1.439265
## iter  34 value -1.439265
## iter  35 value -1.439266
## iter  36 value -1.439266
## iter  37 value -1.439267
## iter  38 value -1.439267
## iter  39 value -1.439267
## iter  39 value -1.439267
## iter  39 value -1.439267
## final  value -1.439267
## converged
## initial  value -1.438392
## iter   2 value -1.438398
## iter   3 value -1.438400
## iter   4 value -1.438405
## iter   5 value -1.438409
## iter   6 value -1.438412
## iter   7 value -1.438415
## iter   8 value -1.438419
## iter   9 value -1.438424
## iter  10 value -1.438427
## iter  11 value -1.438428
## iter  12 value -1.438429
## iter  13 value -1.438429
## iter  14 value -1.438429
## iter  15 value -1.438429
## iter  16 value -1.438429
## iter  17 value -1.438429
## iter  18 value -1.438430
## iter  19 value -1.438430
## iter  19 value -1.438430
## iter  19 value -1.438430
## final  value -1.438430
## converged
```



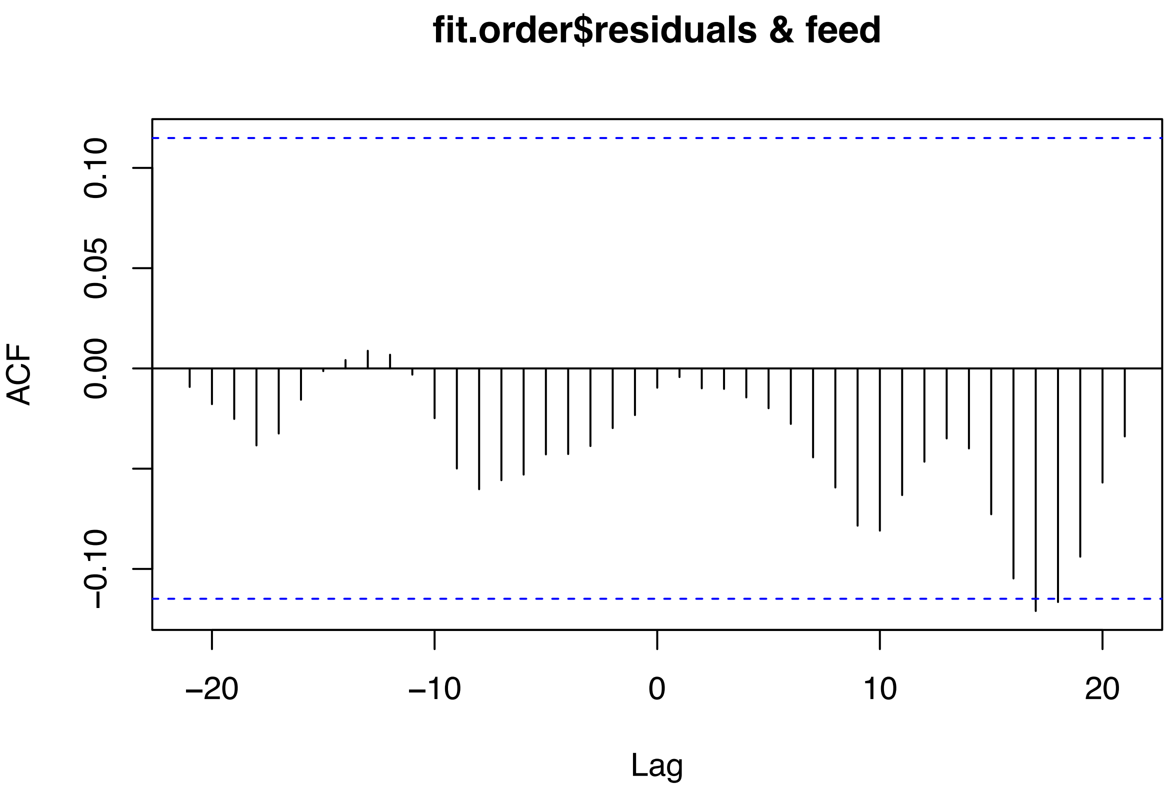

```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xreg, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1     ma1     ma2  intercept  co2_l1   co2_l2  feed_l3  feed_l4
##       0.7400  0.0691  0.2036    21.3088  0.7311  -0.1303  -0.5392  -0.2758
## s.e.  0.0708  0.1604  0.0713     3.1347  0.1572   0.1085   0.0740   0.1284
##       feed_l5
##       -0.4134
## s.e.   0.1415
##
## sigma^2 estimated as 0.05608:  log likelihood = 5.67,  aic = 8.66
##
## $degrees_of_freedom
## [1] 282
##
## $ttable
##           Estimate     SE t.value p.value
## ar1         0.7400 0.0708 10.4467  0.0000
## ma1         0.0691 0.1604  0.4306  0.6671
## ma2         0.2036 0.0713  2.8551  0.0046
## intercept  21.3088 3.1347  6.7977  0.0000
```

```
## co2_l1       0.7311 0.1572  4.6522  0.0000
## co2_l2      -0.1303 0.1085 -1.2005  0.2310
## feed_l3     -0.5392 0.0740 -7.2841  0.0000
## feed_l4     -0.2758 0.1284 -2.1478  0.0326
## feed_l5     -0.4134 0.1415 -2.9226  0.0038
##
## $AIC
## [1] 0.02974651
##
## $AICc
## [1] 0.03194778
##
## $BIC
## [1] 0.1559775
```

Comment on diagnostics:

The residuals are distributed around 0 with unequal variances, acf cuts off so residuals are uncorrelated and the p-values are above 0.05 so the null hypothesis is not rejected, however, the qqplot of residuals seems to be a bit off from normal because of some outliers.

```r
ccf(fit.order$residuals,feed)
```



The ccf shows that the residuals uncorrelated, the model has well captured information and is good.

4. [7 marks] Finally, fit a VAR(p) model to the bivariate series, using AIC to select the order (with `vars::VARselect()`). Pass the resulting model through the `vars::restrict` (with default options), in order to drop insignificant terms. Report the fitted model specification for `co2`, the residual standard deviation for `co2`, the model diagnostics (residual plot, ACF, and Q-Q plot) for `co2`, and plot the cross-correlation between the model's `co2` residuals and `feed`, and comment.

```r
library(vars)
```

```
## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Loading required package: urca

## Loading required package: lmtest
```

```r
vars::VARselect(gas_furnace)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      6      4      4      6
##
## $criteria
##                  1           2            3            4            5
## AIC(n) -3.55979306 -5.91271216 -5.998940762 -6.052683612 -6.045189391
## HQ(n)  -3.52904975 -5.86147331 -5.927206379 -5.960453692 -5.932463933
## SC(n)  -3.48309393 -5.78488028 -5.819976128 -5.822586226 -5.763959252
## FPE(n)  0.02844475  0.00270486  0.002481428  0.002351641  0.002369412
##                   6            7            8            9           10
## AIC(n) -6.064318582 -6.043106955 -6.045457350 -6.031392444 -6.007456373
## HQ(n)  -5.931097585 -5.889390421 -5.871245278 -5.836684834 -5.792253225
## SC(n)  -5.731955690 -5.659611311 -5.610828953 -5.545631295 -5.470562471
## FPE(n)  0.002324633  0.002374628  0.002369262  0.002403089  0.002461642
```

```r
out = VAR(gas_furnace,6,ic="AIC")
vars::restrict(out)
```

```
##
## VAR Estimation Results:
## =======================
##
## Estimated coefficients for equation co2:
## ========================================
## Call:
## co2 = co2.l1 + co2.l2 + feed.l3 + co2.l4 + feed.l6 + const
##
##      co2.l1      co2.l2     feed.l3      co2.l4     feed.l6       const
##  1.55971795 -0.69829888 -0.46535086  0.06759753  0.23245755  3.79161640
##
##
## Estimated coefficients for equation feed:
## =========================================
## Call:
## feed = feed.l1 + feed.l2 + feed.l5 + feed.l6
##
```

```
##    feed.l1    feed.l2    feed.l5    feed.l6 
##  1.8942519 -1.0753724  0.2430774 -0.1138111
```

co2 = co2.l1 + co2.l2 + feed.l3 + co2.l4 + feed.l6 + const co2.l1 co2.l2 feed.l3 co2.l4 feed.l6 const 1.55971795 -0.69829888 -0.46535086 0.06759753 0.23245755 3.79161640

```r
restrict.fit = vars::restrict(out)
summary(restrict.fit)
```

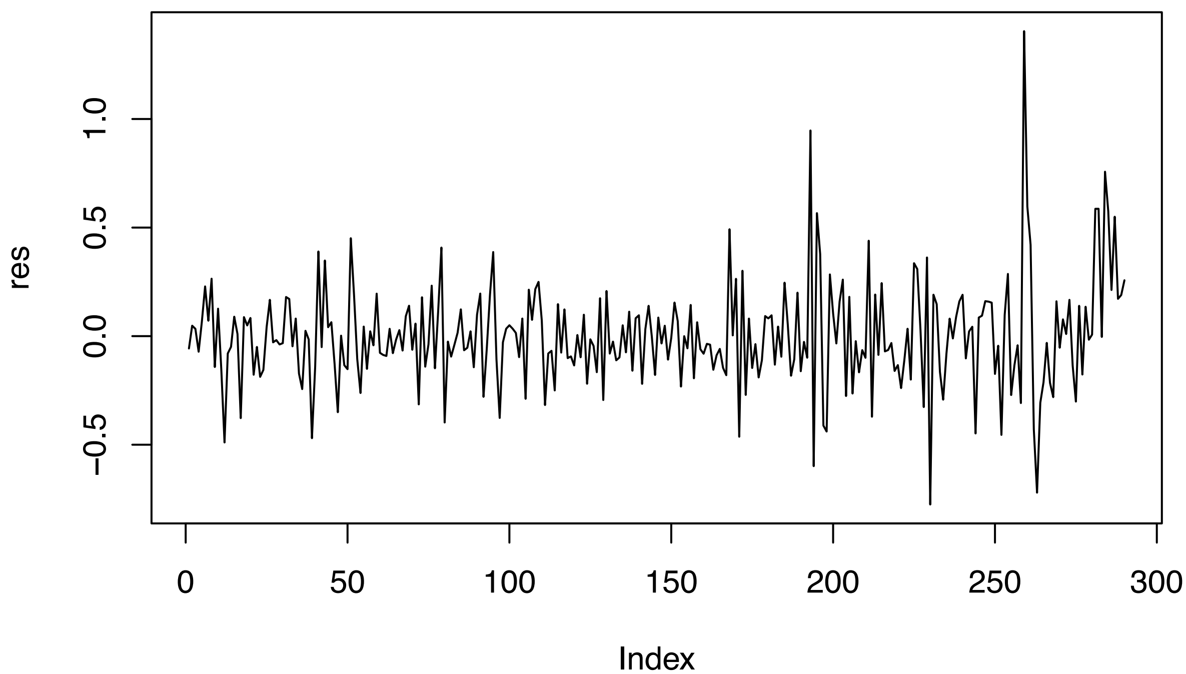
```
## 
## VAR Estimation Results:
## ========================= 
## Endogenous variables: co2, feed 
## Deterministic variables: const 
## Sample size: 290 
## Log Likelihood: 79.922 
## Roots of the characteristic polynomial:
## 0.8459 0.7805 0.7805 0.7528 0.7528 0.5741 0.5741 0.5726 0.5726 0.2437     0     0
## Call:
## VAR(y = gas_furnace, p = 6, ic = "AIC")
## 
## 
## Estimation results for equation co2: 
## ==================================== 
## co2 = co2.l1 + co2.l2 + feed.l3 + co2.l4 + feed.l6 + const 
## 
##         Estimate Std. Error t value Pr(>|t|)    
## co2.l1   1.55972    0.04478  34.832  < 2e-16 ***
## co2.l2  -0.69830    0.05598 -12.474  < 2e-16 ***
## feed.l3 -0.46535    0.02809 -16.566  < 2e-16 ***
## co2.l4   0.06760    0.01835   3.683 0.000276 ***
## feed.l6  0.23246    0.04719   4.926 1.43e-06 ***
## const    3.79162    0.84433   4.491 1.03e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## 
## Residual standard error: 0.2403 on 284 degrees of freedom
## Multiple R-Squared:     1,   Adjusted R-squared:     1 
## F-statistic: 2.405e+06 on 6 and 284 DF,  p-value: < 2.2e-16 
## 
## 
## Estimation results for equation feed: 
## ===================================== 
## feed = feed.l1 + feed.l2 + feed.l5 + feed.l6 
## 
##         Estimate Std. Error t value Pr(>|t|)    
## feed.l1  1.89425    0.04533  41.786   <2e-16 ***
## feed.l2 -1.07537    0.05731 -18.765   <2e-16 ***
## feed.l5  0.24308    0.05730   4.242    3e-05 ***
## feed.l6 -0.11381    0.04533  -2.511   0.0126 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## 
```

```
## Residual standard error: 0.1884 on 286 degrees of freedom
## Multiple R-Squared: 0.9701,  Adjusted R-squared: 0.9697
## F-statistic:  2323 on 4 and 286 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##           co2     feed
## co2   0.05922 -0.00232
## feed -0.00232  0.03664
##
## Correlation matrix of residuals:
##          co2    feed
## co2   1.0000 -0.0498
## feed -0.0498  1.0000
```

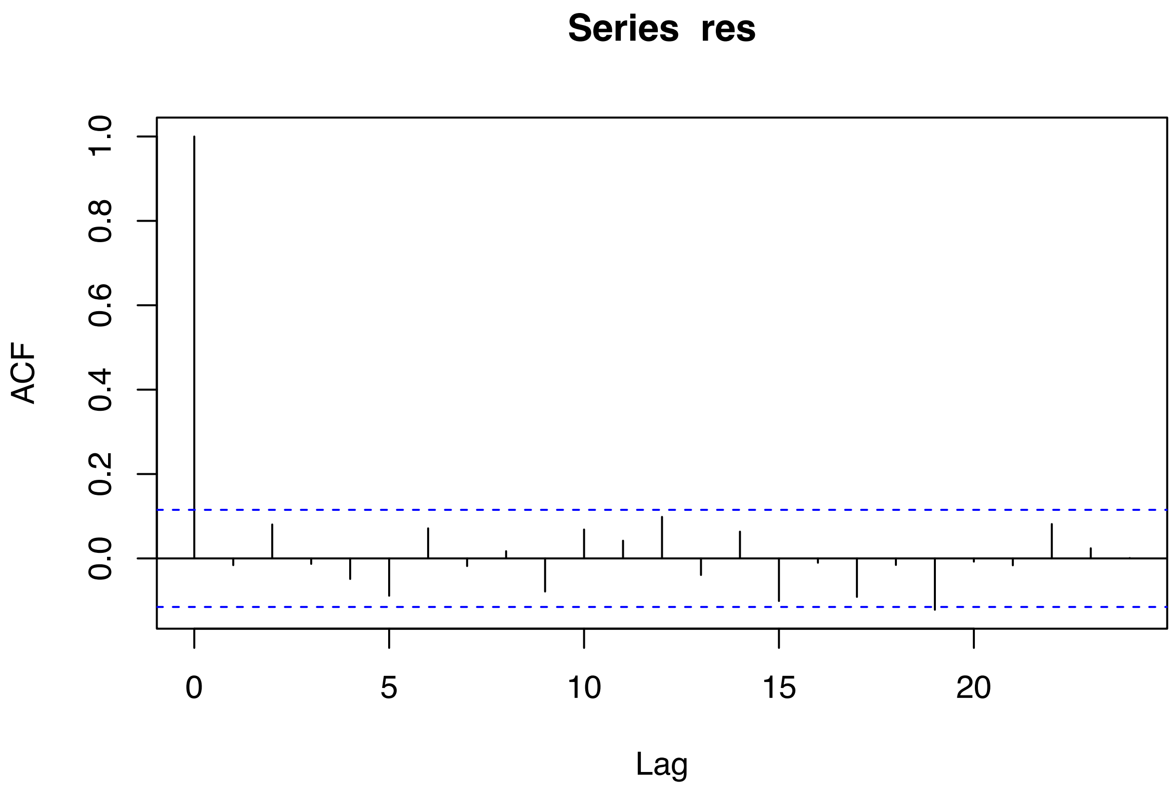
CO2 Residual standard error: 0.2403

```r
co2.restric = restrict.fit$varresult$co2
res = co2.restric$residuals
```

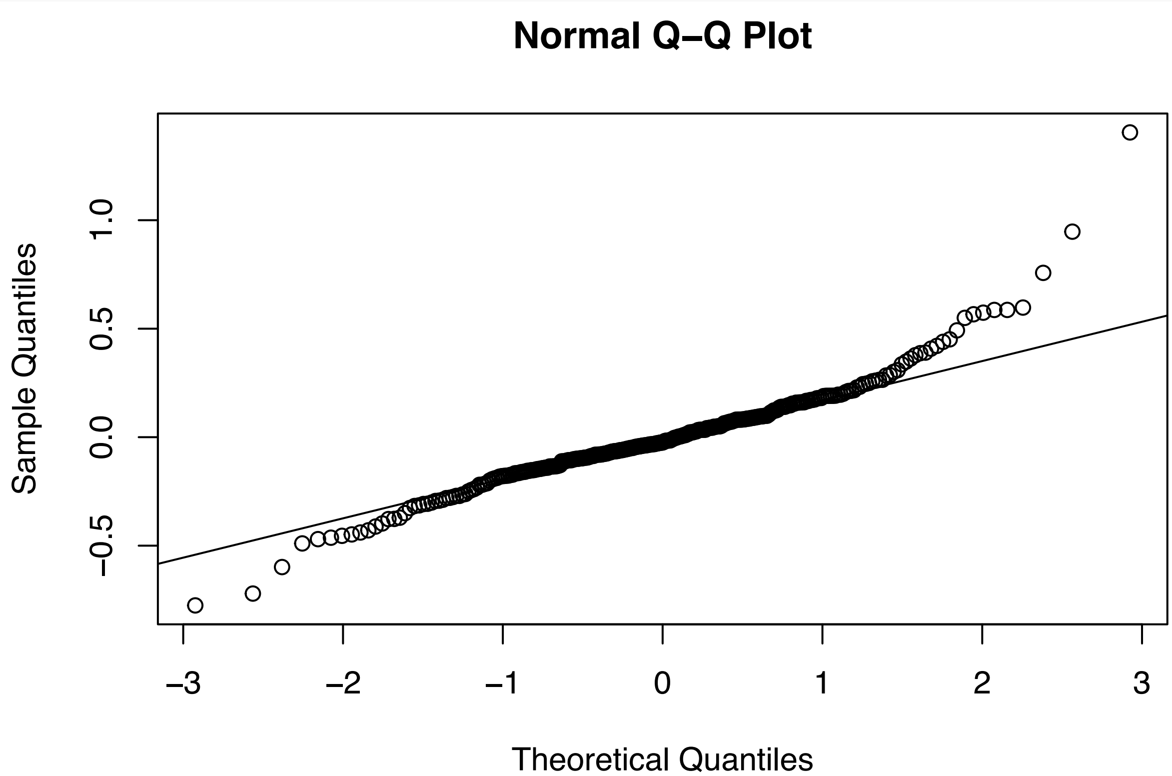
```r
plot(res,type = "l")
```




```r
acf(res)
```



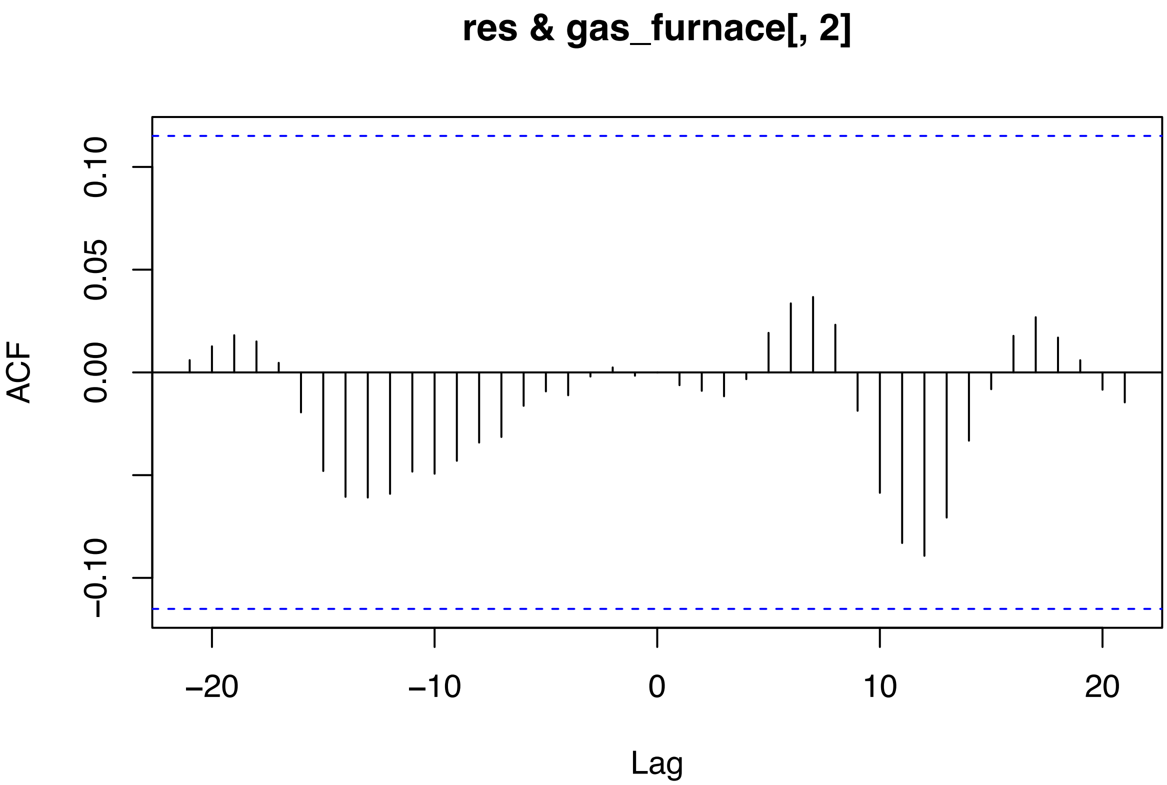

```
qqnorm(res)
qqline(res)
```



Comment on diagnostics:

The residuals are distributed around 0 with unequal variances, acf cuts off so residuals are uncorrelated and the p-values are above 0.05 so the null hypothesis is not rejected, however, the qqplot of residuals seems to be a bit off from normal because of some outliers.

```
ccf(res,gas_furnace[,2])
```



The ccf shows that the residuals have very low correlations with feed series, this model is a good model.