R Sourse Codes: Analysis of the Stationarity and Correlation of the Global Temperature and Carbon Dioxide Time Series

Upul Rupassara

2023-04-19

NOTE: WHEN RUNING THE CODES, REMOVE THE "#" IN FRONT OF "INSTALL.PACKAGES()".

Install Packages

```
#install.packages("ggplot2")
library(ggplot2)
```

```
#install.packages("astsa")
library(astsa)
```

###Transforming data into time series data

```
#install.packages("xts")
library(xts)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

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

Transforming the data into timeseries format

```
tc = read.csv("tc.csv")
x = tc$temperature
y = tc$co2
z = tc$ca

temp = ts(x, frequency = 1, start = 1960, end = 2022)
co2 = ts(y, frequency = 1, start = 1960, end = 2022)
ca = ts(z, frequency = 1, start = 1960, end = 2022)


par(mfrow = c(3,1))
par(mar=c(2, 2, 2, 2), mfrow=c(2,1),oma = c(1, 1, 1, 1))
par(adj = 0)
```

Model 1:

```r
# Temperature
model1 = lm(temp ~ time(temp), na.action = NULL)
model1
```

```
##
## Call:
## lm(formula = temp ~ time(temp), na.action = NULL)
##
## Coefficients:
## (Intercept)   time(temp)
##   -33.81971      0.01717
```

```r
summary(model1)
```

```
##
## Call:
## lm(formula = temp ~ time(temp), na.action = NULL)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.207692 -0.080426  0.003065  0.076083  0.215518
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.382e+01  1.404e+00  -24.09   <2e-16 ***
## time(temp)   1.717e-02  7.051e-04   24.35   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1018 on 61 degrees of freedom
## Multiple R-squared:  0.9067, Adjusted R-squared:  0.9052
## F-statistic:   593 on 1 and 61 DF,  p-value: < 2.2e-16
```

AIC for Model 1:

```r
# Temperature
AIC(model1)/nrow(tc) - log(2*pi)
```

```
## [1] -3.507239
```

BIC for Model 1

```r
# Temperature
BIC(model1)/nrow(tc) - log(2*pi)
```

```
## [1] -3.405184
```

SSE for Model 1

```r
# Temperature
sum(resid(model1)^2)
```

```
## [1] 0.6316992
```

AIC with formula, model 1

```
k = 2
log(sum(resid(model1)^2)/nrow(tc)) + (nrow(tc)+(2*k))/(nrow(tc))
```

```
## [1] -3.538985
```

```
nrow(tc)
```

```
## [1] 63
```

MSEfor model 1 using summary of the model 1

```
mean(summary(model1)$residuals^2)
```

```
## [1] 0.01002697
```

```
summary(model1)$df
```

```
## [1]  2 61  2
```

```
sum(summary(model1)$residual^2)
```

```
## [1] 0.6316992
```

MSE model 1

```
sum(summary(model1)$residual^2)/61
```

```
## [1] 0.01035573
```

AICc with formula, model 1

```
k = 2
log(sum(resid(model1)^2)/nrow(tc)) + (nrow(tc)+k)/(nrow(tc)-k-2)
```

```
## [1] -3.500782
```

BIC with formula, model 1

```
k = 2
log(sum(resid(model1)^2)/nrow(tc)) + k*log(nrow(tc))/(nrow(tc))
```

```
## [1] -4.470949
```

Model 2

```
# Temp dependent on CO2

cc = mean(co2)
C = co2 - cc
model2 = lm(temp ~ time(temp) + C)
model2
```

```
##
## Call:
```

```
## lm(formula = temp ~ time(temp) + C)
##
## Coefficients:
## (Intercept)   time(temp)            C
##    4.942728    -0.002299     0.011860
```

```r
summary(model2)
```

```
##
## Call:
## lm(formula = temp ~ time(temp) + C)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.180255 -0.076860 -0.004128  0.076063  0.167490
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.942728   9.624375   0.514 0.609443
## time(temp)  -0.002299   0.004834  -0.476 0.636076
## C            0.011860   0.002920   4.062 0.000143 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09087 on 60 degrees of freedom
## Multiple R-squared:  0.9268, Adjusted R-squared:  0.9244
## F-statistic: 380.1 on 2 and 60 DF,  p-value: < 2.2e-16
```

```r
sum(summary(model2)$residual^2)
```

```
## [1] 0.4954444
```

CALCULATION (F-STATISTICS)

```r
(0.632 - 0.495)/(0.495/60)
```

```
## [1] 16.60606
```

```r
qf(0.001, 1, 60, lower.tail = FALSE)
```

```
## [1] 11.97299
```

AIC with formula, model 2

```r
k = 3
log(sum(resid(model2)^2)/nrow(tc)) + (nrow(tc)+2*k)/(nrow(tc))
```

```
## [1] -3.750197
```

AIC (ALTERNATIVE METHOD)

```r
AIC(model2)/nrow(tc) - log(2*pi)
```

```
## [1] -3.718451
```

AICc with formula, model 2

```
k = 3
log(sum(resid(model2)^2)/nrow(tc)) + (nrow(tc)+k)/(nrow(tc)-k-2)
```

```
## [1] -3.707504
```

BIC with formula, model 2

```
k = 3
r=  (resid(model2))^2
log(sum(r)/nrow(tc)) + k*log(nrow(tc))/(nrow(tc))
```

```
## [1] -4.648143
```

```
(BIC(model2))/nrow(tc) - log(2*pi)
```

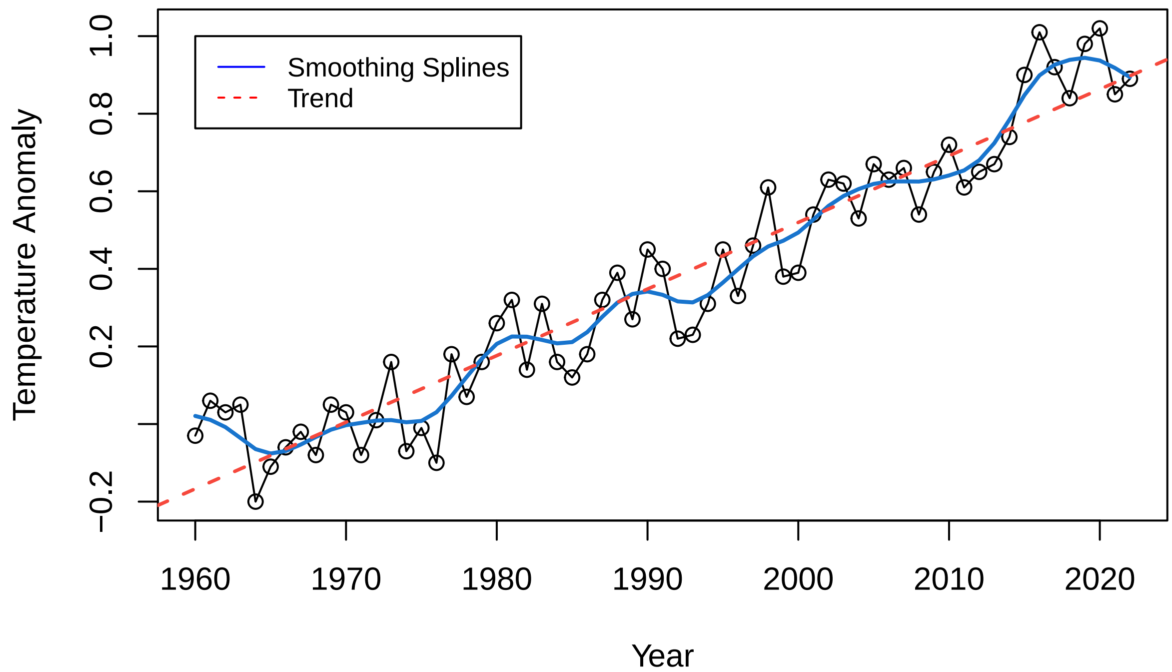
```
## [1] -3.582379
```

Smoothing splines for temprature

SOURCE CODES FOR FIGURE 2

```
#png("fig02.png", width = 12, height = 10, units = 'cm', res = 300)
plot(x = tc$Year, y = tc$temperature, type = "o", xlab = "Year", ylab = "Temperature Anomaly")

lines(smooth.spline(time(temp), temp, spar = 0.5), lwd = 2, col = 4)
abline(model1, col = 2, lty = "dashed", lwd = 1.75)
legend(1960, 1, legend=c("Smoothing Splines", "Trend"),
       col=c("blue", "red"), lty=1:2, cex=0.8)
```



Durbin-Watson test (Optional)

```
library(lmtest)
dwtest(formula = model1,  alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  model1
## DW = 1.3527, p-value = 0.005308
## alternative hypothesis: true autocorrelation is not 0
```

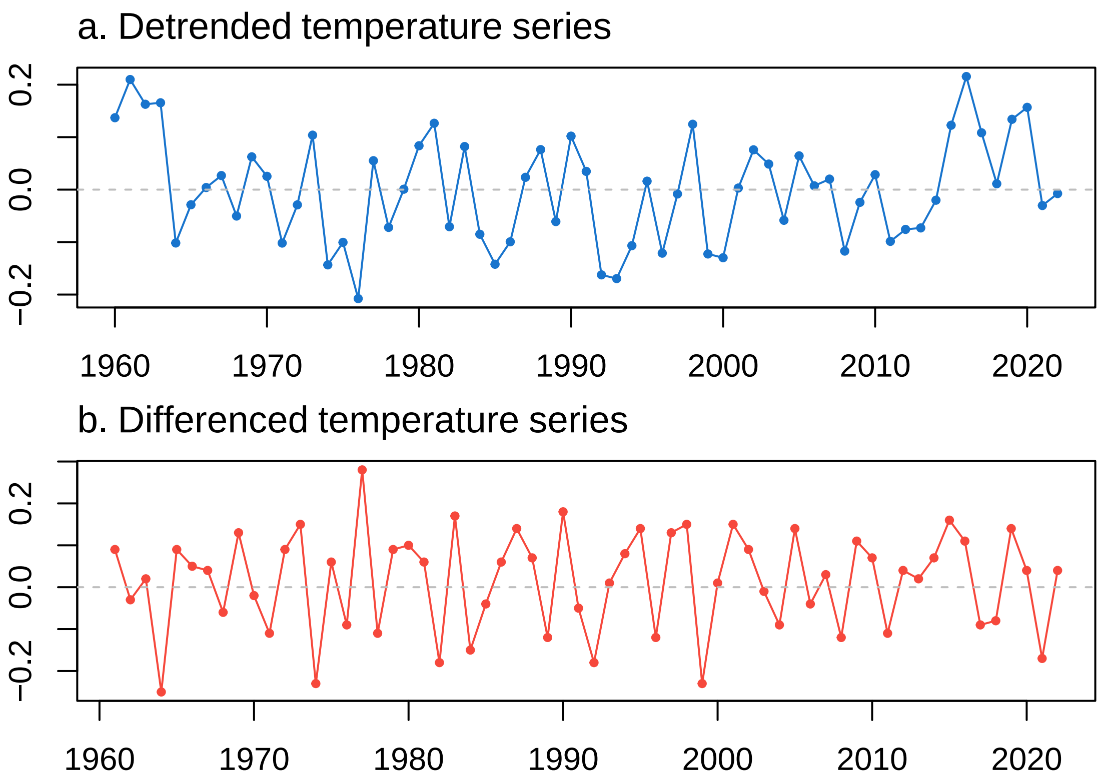
Temperature Detrended and differenced

SOURCE CODES FOR FIGURE 3

```
#png("fig03.png", width = 12, height = 12, units = 'cm', res = 300)
par(mfrow = c(2,1))
par(mar=c(2, 2, 2, 2), mfrow=c(2,1),oma = c(1, 1, 1, 1)) #margin a nd outer margin
par(adj = 0) #shift the title to left

plot(resid(model1), type = "o", main = expression(a.~Detrended~temperature~series), pch = 19, cex = 0.5

abline(h = 0, lty = 2, col = "gray")
plot(diff(temp), type = "o", main = expression(b.~Differenced~temperature~series), pch = 19, cex =0.5,
abline(h = 0, lty = 2, col = "gray")
```

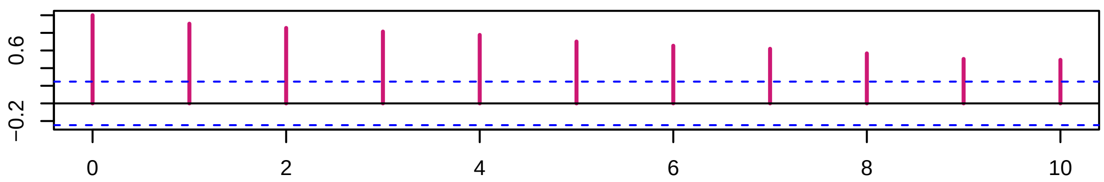


ACF plots

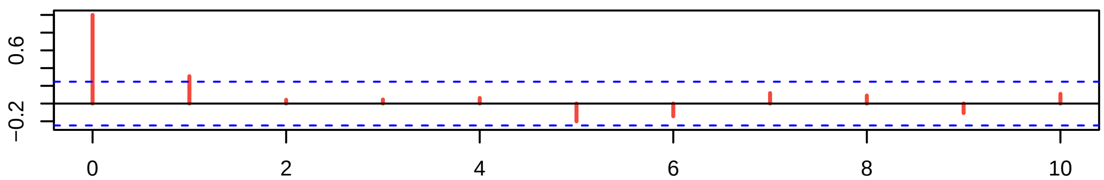
SOURCE CODES FOR FIGURE 4

```
#png("fig04.png", width = 16, height = 12, units = 'cm', res = 300)
par(mar=c(3, 3, 3, 3), mfrow=c(3,1),oma = c(1, 1, 1, 1)) #margin a nd outer margin
par(adj = 0) #shift the title to left
acf(temp, 10, main = "a. ACF vs Lag: temperature anomaly", col = 6, lwd = 2)
acf(resid(model1),10, main = "b. ACF vs Lag: detrended temperature series", col = 2, lwd = 2)
acf(diff(temp), 10,main = "c. ACF vs Lag: differenced temperature series", col = 3, lwd = 2)
```

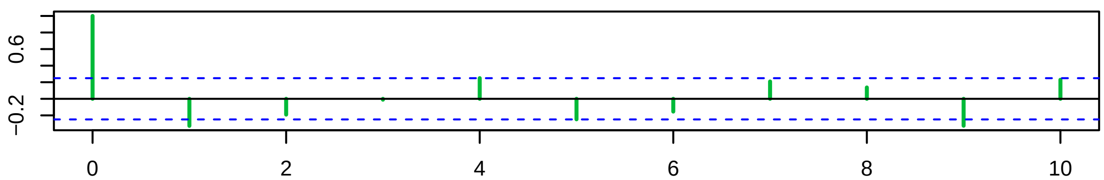
a. ACF vs Lag: temperature anomaly



b. ACF vs Lag: detrended temperature series



c. ACF vs Lag: differenced temperature series



CO2 time series ACF plots

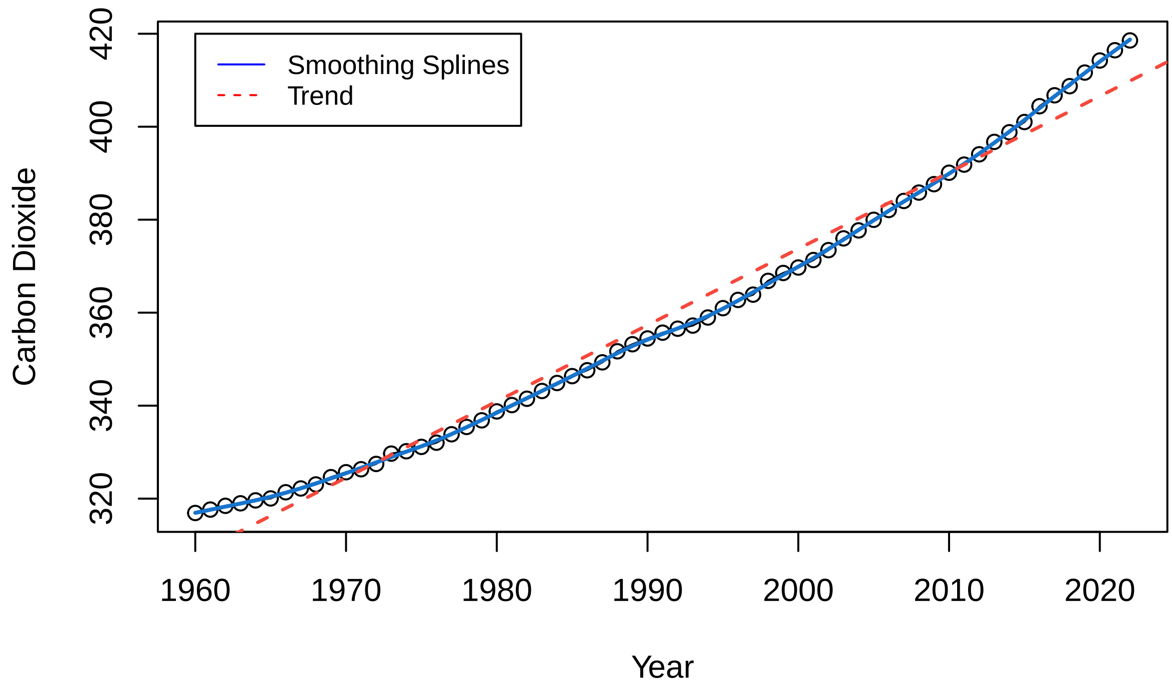
```r
modelc = lm(co2~time(co2), na.action = NULL)
summary(modelc)
```

```
##
## Call:
## lm(formula = co2 ~ time(co2), na.action = NULL)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.031 -2.740 -1.501  2.348  8.832
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.909e+03  5.497e+01  -52.92   <2e-16 ***
## time(co2)    1.642e+00  2.761e-02   59.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.985 on 61 degrees of freedom
## Multiple R-squared:  0.983,  Adjusted R-squared:  0.9828
## F-statistic:  3535 on 1 and 61 DF,  p-value: < 2.2e-16
```

Smoothing Splines for Carbon dioxide

SOURCE CODES FOR FIGURE 5

```
#png("fig05.png", width = 12, height = 10, units = 'cm', res = 300)
plot(x = tc$Year, y = tc$co2, type = "o", xlab = "Year", ylab = "Carbon Dioxide")
lines(smooth.spline(time(co2), co2, spar = 0.5), lwd = 2, col = 4)
abline(modelc, col = 2, lty = "dashed", lwd = 1.75)
legend(1960, 420, legend=c("Smoothing Splines", "Trend"), col=c("blue", "red"), lty=1:2, cex=0.8)
```



CO2

SOURCE CODES FOR FIGURE 6

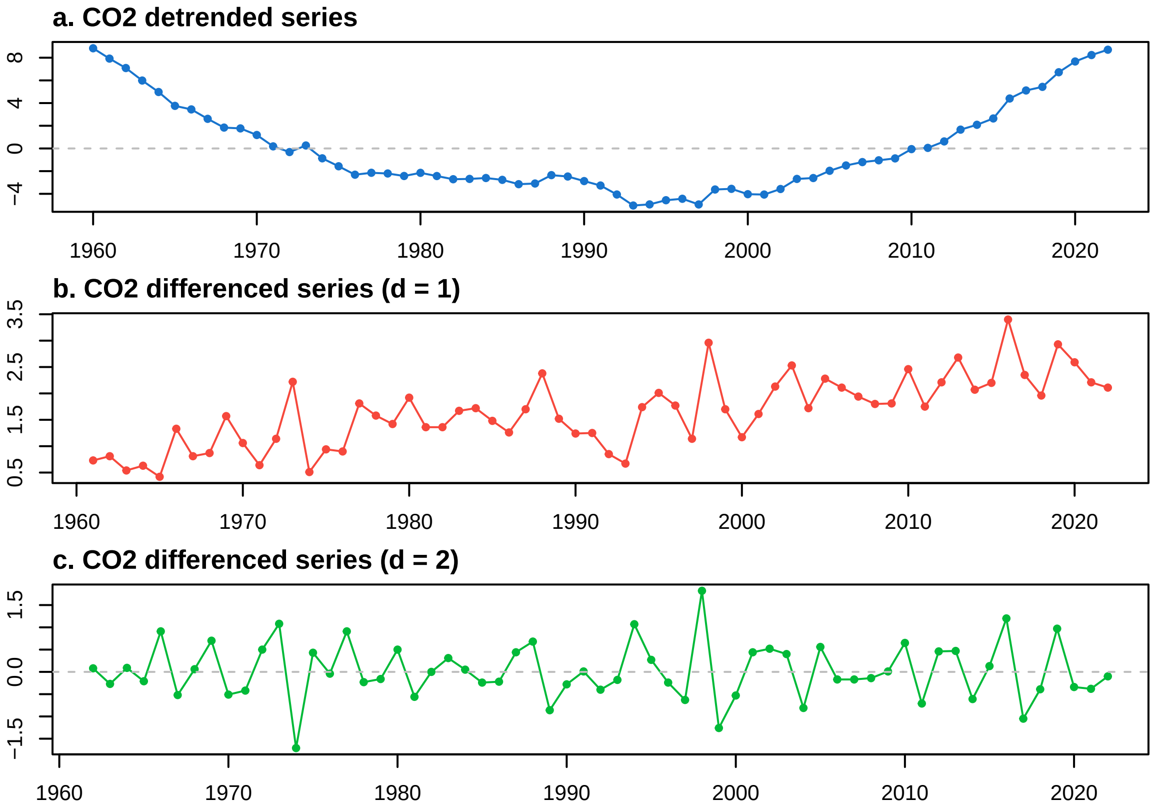
```
#png("fig06.png", width = 12, height = 12,  units = 'cm', res = 300)
par(mar=c(2, 2, 2, 2), mfrow=c(3,1),oma = c(1, 1, 1, 1))
par(adj = 0)

modelc = lm(co2~time(co2), na.action = NULL)


#mat_layout<- matrix(c(1,0,2,0), ncol = 2)
#layout(mat_layout)

plot(resid(modelc), main = "a. CO2 detrended series", type = "o", pch = 19, col = 4)
```

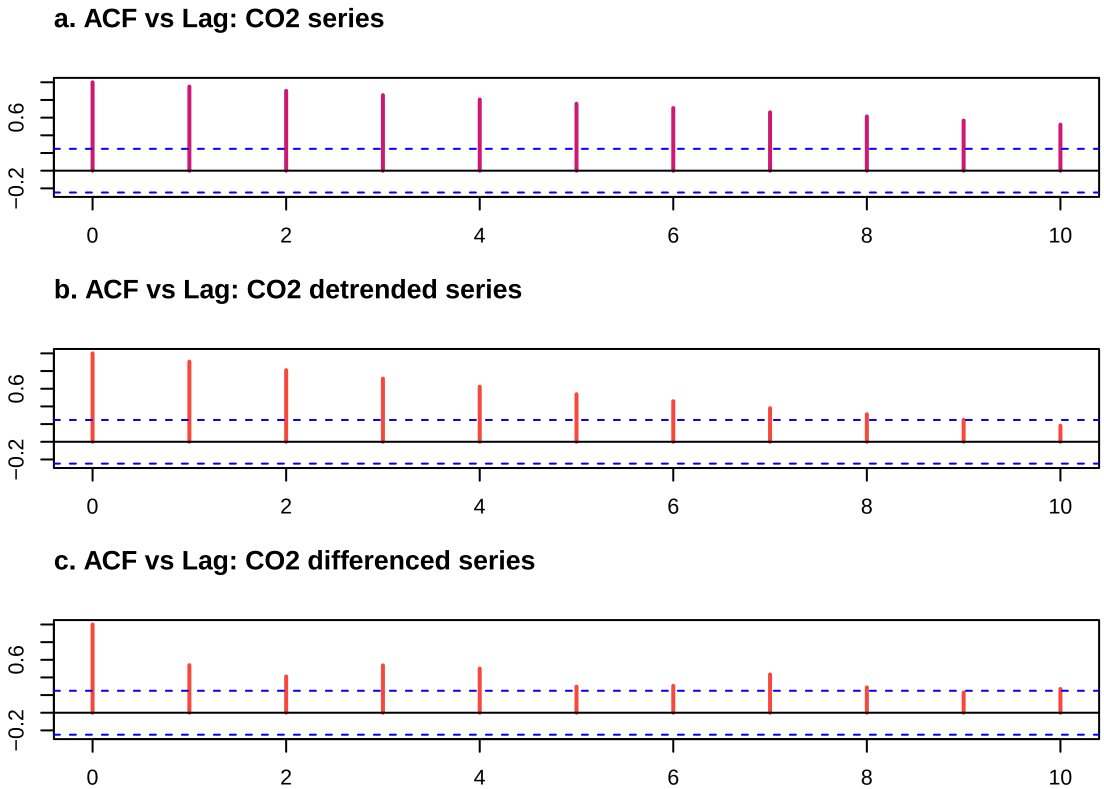
```r
#grid(nx = NULL, ny = NULL,lty = 2, col = "gray", lwd = 2)
abline(h = 0, lty = 2, col = "gray")
plot(diff(co2), main = "b. CO2 differenced series (d = 1)", type = "o",  pch = 19, col = 2)
abline(h = 0, lty = 2, col = "gray")
plot(diff(diff(co2)), type = "o", main = "c. CO2 differenced series (d = 2)", pch = 19, col = 3)
abline(h = 0, lty = 2, col = "gray")
```



SOURCE CODES FOR FIGURE 7

```r
#png("fig07.png", width = 16, height = 12, units = 'cm', res = 300)

par(mar=c(3, 3, 3, 3), mfrow=c(3,1),oma = c(1, 1, 1, 1)) #margin a nd outer margin
par(adj = 0) #shift the title to left
acf(tc$co2, 10, main = "a. ACF vs Lag: CO2 series", col = "6", lwd = 2)
acf(resid(modelc),10, main = "b. ACF vs Lag: CO2 detrended series", col = 2, lwd = 2 )
acf(diff(tc$co2), 10, main = "c. ACF vs Lag: CO2 differenced series", col = 2, lwd = 2)
```



REFERENCE FOR ADJUSTING GAP

https://stackoverflow.com/questions/15848942/how-to-reduce-space-gap-between-multiple-graphs-in-r

DW test for CO2 (OPTIONAL)

```
dwtest(formula = modelc,  alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  modelc
## DW = 0.028277, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is not 0
```

DW test for Temperature (OPTIONAL)

```
dwtest(formula = model1,  alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  model1
## DW = 1.3527, p-value = 0.005308
## alternative hypothesis: true autocorrelation is not 0
```

According to the ACF plots detrended series represents the stationary behavior than the differenced series.

Lag plot

'DW test

H0: residuals are independent H1: Residuals are not independent

```
dwtest(formula = model1, alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  model1
## DW = 1.3527, p-value = 0.005308
## alternative hypothesis: true autocorrelation is not 0
```

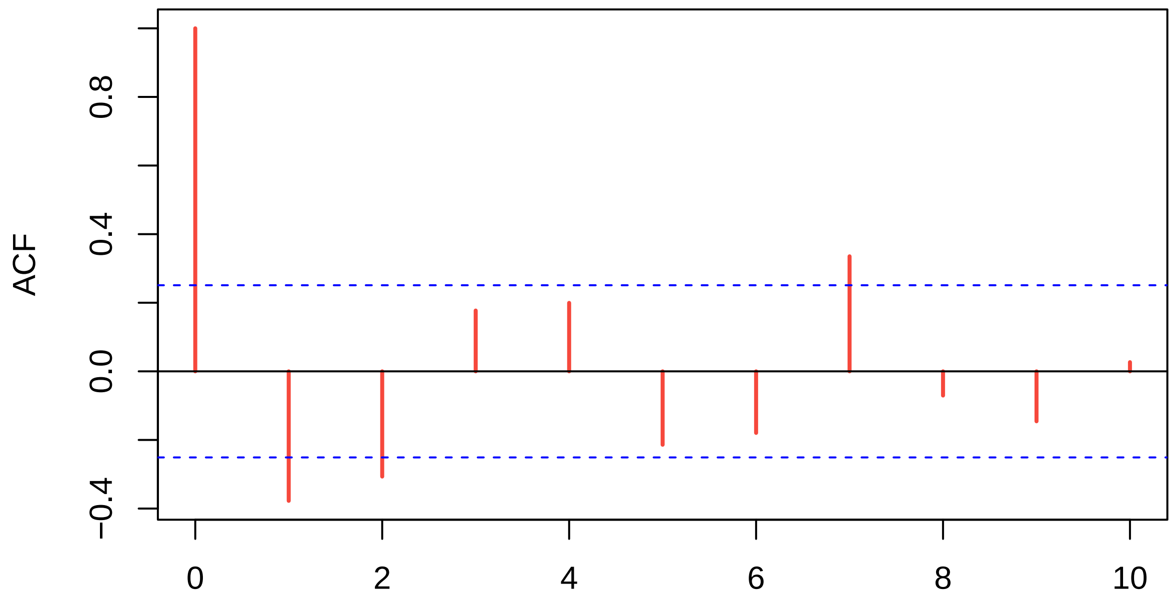
```
dwtest(model2, alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  model2
## DW = 1.6634, p-value = 0.1049
## alternative hypothesis: true autocorrelation is not 0
```

According to the Durbin-Watson test the second model follows the assumptions than th first model. The assumption of residuals are independent. ### ACF for CO2 second differenced series

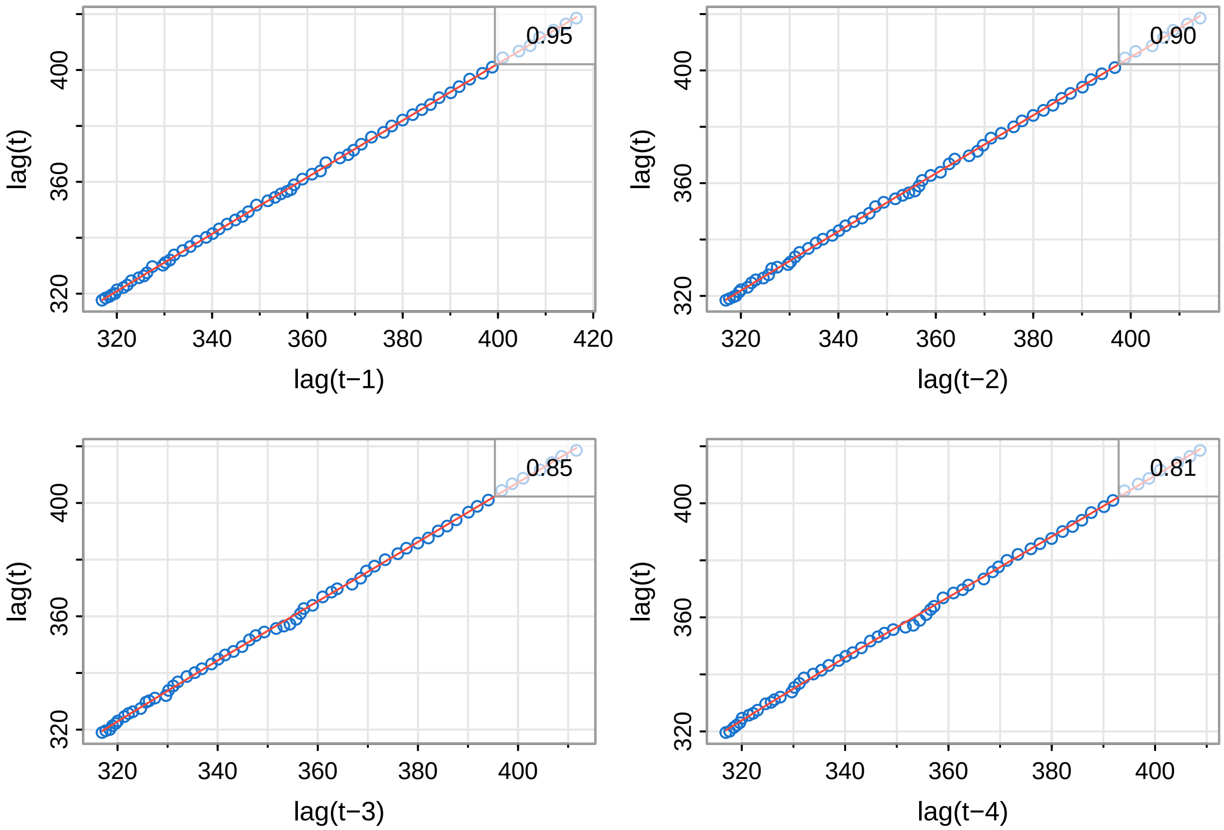
SOURCE CODES FOR FIGURE 8

```
#png("fig08.png", width = 12, height = 8, units = 'cm', res = 300)
#par(mfrow = c(1,2))
#acf(diff(tc$co2), 10, main = "c. ACF vs Lag: CO2 differenced series", col = 2, lwd = 2)
acf(diff(diff(tc$co2)), 10, main = " ", col = 2, lwd = 2, xlab = " ")
```



SOURCE CODES FOR FIGURE 9

```
#png("fig09.png", width = 12, height = 10, units = 'cm', res = 300)
lag= tc$co2
lag1.plot(lag, 4, col = 4)
```




```r
mean(tc$co2)
```

```
## [1] 358.9675
```

Lag Regression

We need the following package to align the time series before making the model.

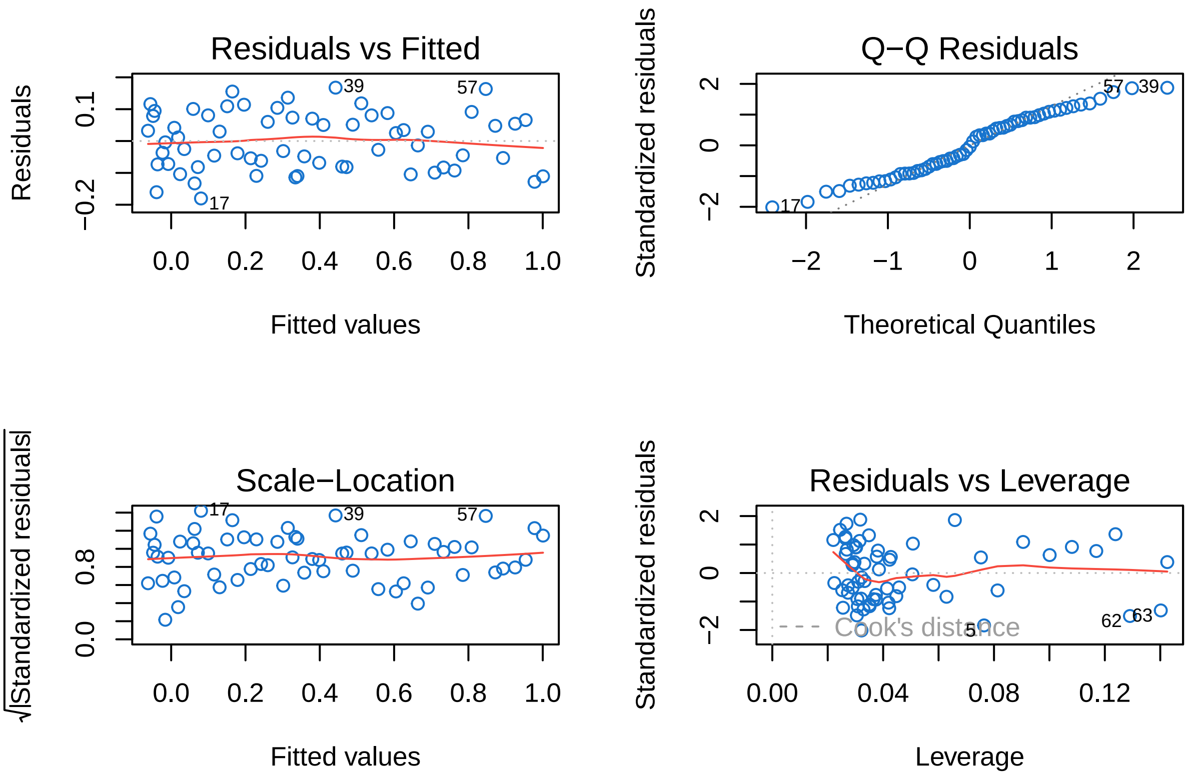
```r
#install.packages("dynlm")
library(dynlm)
```

```r
dwtest(model1, alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  model1
## DW = 1.3527, p-value = 0.005308
## alternative hypothesis: true autocorrelation is not 0
```

SOURCE CODES FOR FIGURE 10

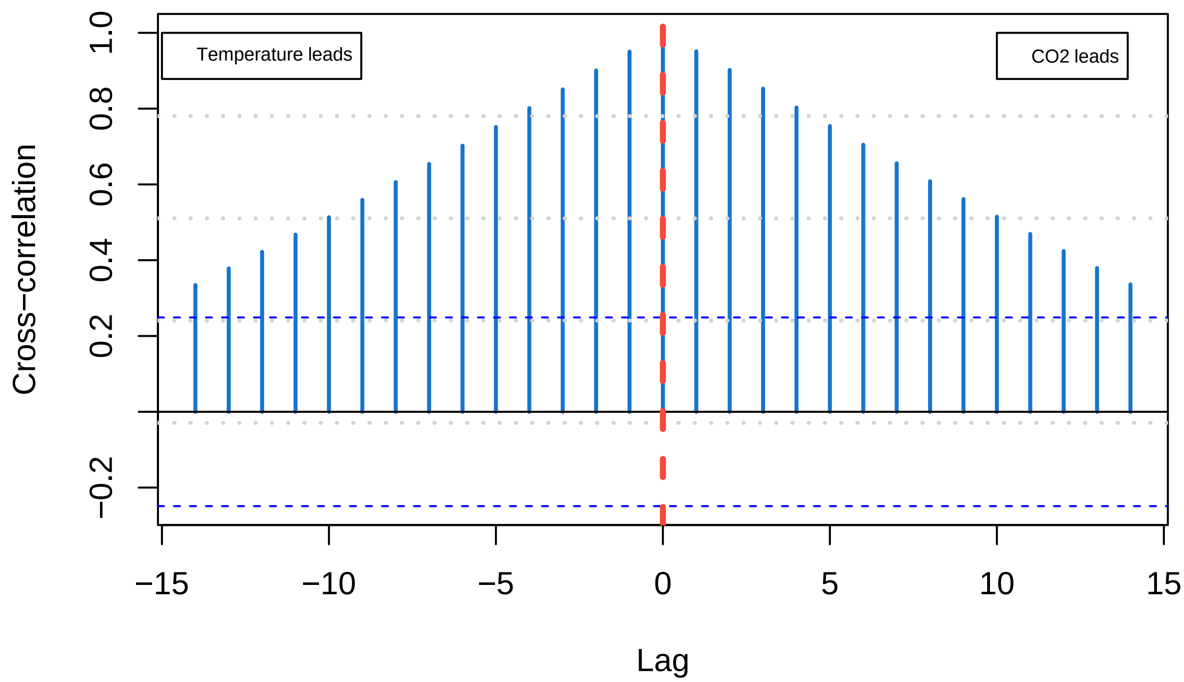
```r
#png("fig10.png", width = 12, height = 10, units = 'cm', res = 300)
par(mfrow = c(2, 2))
plot(model2, col = 4)
```



Cross correlation

SOURCE CODES FOR FIGURE 11

```
#png("fig11.png", width = 14, height = 8, units = 'cm', res = 300)
x = co2
y = lag(x,-1) + temp
ccf(y,x, ylab = "Cross-correlation", type = "correlation", col = 4, lwd = 2, main = " ")
grid(NA, 5, lwd = 2) # grid only in y-direction
abline(v = 0, col = 2, lty = 2, lwd = 3)
legend(-15, 1, expression(Temperature~leads), cex=0.6)
legend(10, 1, expression(CO2~leads), cex=0.6)
```




```
bgtest(model1, order = 1)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  model1
## LM test = 6.0064, df = 1, p-value = 0.01425
```

```
#install.packages("tseries")
library(tseries)
```

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

Dicky Fuller test (OPTIONAL)

REFERENCE

https://atsa-es.github.io/atsa-labs/sec-boxjenkins-aug-dickey-fuller.html

```
#install.packages("urca")
library(urca)
```

```
library(urca)
s= ur.df(temp, type = "trend")
s
```

```
##
```

```
## ###############################################################
## # Augmented Dickey-Fuller Test Unit Root / Cointegration Test #
## ###############################################################
##
## The value of the test statistic is: -5.1294 9.6225 13.3195
```

```r
summary(s)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.17268 -0.07812  0.01604  0.07230  0.17292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.145845   0.038884  -3.751 0.000414 ***
## z.lag.1     -0.764339   0.149010  -5.129 3.63e-06 ***
## tt           0.013636   0.002672   5.103 4.00e-06 ***
## z.diff.lag   0.045225   0.126174   0.358 0.721344
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09515 on 57 degrees of freedom
## Multiple R-squared:  0.3917, Adjusted R-squared:  0.3597
## F-statistic: 12.24 on 3 and 57 DF,  p-value: 2.751e-06
##
##
## Value of test-statistic is: -5.1294 9.6225 13.3195
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```