Appendix

Trevor Rizzi

2023-06-09

```
# Reading in data
co2data <- read.csv("/Users/trevorrizz/Downloads/co2_mm_gl.csv") # global

co2data <- co2data[c("decimal", "average")]
sum(is.na(co2data)) # checking for any NA values

co2.train = head(co2data, 770) #training
co2.test = tail(co2data, 12) #testing - just the last 12 months

# Time series data for training, and for entire data set
ts <- ts(co2.train[,2], start = c(1979),end = c(2022), frequency = 12)
ts_ALL<-ts(co2data[,2], start = c(1979),end = c(2023), frequency = 12)

#Plotting time series
plot(ts, ylab = "CO_2 Mole Fraction (ppm)", xlab = "Year")
fit <- lm(ts ~ as.numeric(time(ts)));abline(fit, col="blue")

length(ts)
length(co2data[,2])
```

```
# Plot acf and histogram of original data
acf(ts, lag.max = 100, main = "ACF")
hist(ts)
```

```
# Code for Box-cox transform
library(MASS)
t = 1:length(ts)
fit = lm(ts ~ t)
bcTransform = boxcox(ts ~ t,plotit = TRUE)
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
ts.bc = (1/lambda)*(ts^lambda-1)
```

```
# Plotting the time series with transforms for comparison
op <- par(mfrow = c(2,2))
# Original
ts.plot(ts,main = "Original data")
fit <- lm(ts ~ as.numeric(time(ts)));abline(fit, col="blue")
# Box cox
ts.plot(ts.bc,main = "Box-Cox tranformed data")
fit <- lm(ts.bc ~ as.numeric(time(ts.bc)));abline(fit, col="blue")
# Log
ts.plot(log(ts), main = "Log transform")
fit <- lm(ts ~ as.numeric(time(ts)));abline(fit, col="blue")
# Square Root
ts.plot(sqrt(ts), main = "Square root transform")
fit <- lm(ts ~ as.numeric(time(ts)));abline(fit, col="blue")
```

```
# Plotting histograms of transformed data
op <- par(mfrow = c(2,2))

hist(ts)
hist(ts.bc)
hist(log(ts))
hist(sqrt(ts))

# Shapiro test for normality
shapiro.test(ts)
library(e1071)
skewness(log(ts)) # moderately skewed
```

```
# Code for decomposition
library(ggplot2)
library(ggfortify)

y <- ts(ts, frequency = 12)
decomp <- decompose(y)
plot(decomp)
```

```
# Differenced at lag 12 to remove seasonality
var(ts)
ts12 <- diff(ts, lag=12)
plot.ts(ts12, main="U_t differenced at lag 12")
var(ts12)
fit <- tslm(ts12 ~ as.numeric(time(ts12))); abline(fit, col="red")

mean(ts12)
abline(h=mean(ts12), col="blue")
```

```
# Differenced at lag 1 to remove trebd
ts1_12 <- diff(ts12, lag=1)
plot.ts(ts1_12, main="U_t differenced at lag 12 and then lag 1")
var(ts1_12)
fit <- tslm(ts1_12 ~ as.numeric(time(ts1_12))); abline(fit, col="red")

mean(ts1_12)
abline(h=mean(ts1_12), col="blue")
```

```
# Code for acf and pacf of original data, first differencing, second differencing
par(mar=c(5,6,4,1)+.1)
acf(ts, lag.max = 40,main = "Original ACF")
pacf(ts, lag.max = 40, ,main = "Original PACF")

acf(ts12, lag.max = 40,main = "ACF differenced at lag 12")
pacf(ts12, lag.max = 40,main = "PACF differenced at lag 12")

acf(ts1_12, lag.max = 40,main = "ACF differenced at lag 12 and lag 1")
pacf(ts1_12, lag.max = 40,main = "PACF differenced at lag 12 and lag 1")
```

```
# Histogram analysis of seasonal data
hist(ts1_12, density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m<-mean(ts1_12)
std<- sqrt(var(ts1_12))
curve( dnorm(x,m,std), add=TRUE )

# test if data is stationary
library(tseries)
adf.test(ts1_12)

hist(ts, density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m<-mean(ts)
std<- sqrt(var(ts))
curve( dnorm(x,m,std), add=TRUE )
```

```
#install.packages('MuMIn')
library(MuMIn)
# Trying SMA models:
#
####
arima(ts, order=c(0,1,1), seasonal = list(order = c(0,1,1), period = 12), method="ML")
AICc(arima(ts, order=c(0,1,1), seasonal = list(order = c(0,1,1), period = 12), method="ML"))

# Call:
# arima(x = ts, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1), period = 12),
#     method = "ML")
#
# Coefficients:
#          ma1     sma1
#       0.8219  -0.8665
# s.e.  0.0226   0.0254
#
# sigma^2 estimated as 0.0135:  log likelihood = 360.68,  aic = -715.37
# [1] -715.3212

####
arima(ts, order=c(0,1,2), seasonal = list(order = c(0,1,1), period = 12), method="ML")
AICc(arima(ts, order=c(0,1,2), seasonal = list(order = c(0,1,1), period = 12), method="ML"))

# Call:
# arima(x = ts, order = c(0, 1, 2), seasonal = list(order = c(0, 1, 1), period = 12),
#     method = "ML")
#
# Coefficients:
#          ma1     ma2     sma1
#       0.8277  0.0065  -0.8668
# s.e.  0.0532  0.0531   0.0258
#
# sigma^2 estimated as 0.0135:  log likelihood = 360.69,  aic = -713.38
# [1] -713.3038

####
arima(ts, order=c(0,1,7), seasonal = list(order = c(0,1,1), period = 12), method="ML")
AICc(arima(ts, order=c(0,1,7), seasonal = list(order = c(0,1,1), period = 12), method="ML"))

# Call:
# arima(x = ts, order = c(0, 1, 7), seasonal = list(order = c(0, 1, 1), period = 12),
#     method = "ML")
#
# Coefficients:
#          ma1      ma2      ma3      ma4      ma5     ma6     ma7     sma1
#       0.8066  -0.1465  -0.1686  -0.0478  -0.0685  0.0301  0.0701  -0.8492
# s.e.  0.0462   0.0599   0.0592   0.0537   0.0581  0.0615  0.0525   0.0274
#
# sigma^2 estimated as 0.01309:  log likelihood = 369.09,  aic = -720.18
# [1] -719.8183

####
arima(ts, order=c(0,1,11), seasonal = list(order = c(0,1,1), period = 12), method="ML")
AICc(arima(ts, order=c(0,1,11), seasonal = list(order = c(0,1,1), period = 12), method="ML"))


# Call:
# arima(x = ts, order = c(0, 1, 11), seasonal = list(order = c(0, 1, 1), period = 12),
#     method = "ML")
#
# Coefficients:
#          ma1      ma2      ma3      ma4      ma5     ma6     ma7     ma8     ma9    ma10    ma11     sma1
#       0.8098  -0.1477  -0.1731  -0.0569  -0.0730  0.0277  0.1436  0.1099  0.0422  0.0680  0.0305  -0.8569
# s.e.  0.0450   0.0577   0.0581   0.0583   0.0588  0.0581  0.0604  0.0641  0.0620  0.0656  0.0483   0.0265
#
# sigma^2 estimated as 0.01293:  log likelihood = 372.13,  aic = -718.27
# [1] -717.5244

# Trying SAR models:
#

####
arima(ts, order=c(4,1,0), seasonal = list(order = c(1,1,0), period = 12), method="ML")
AICc(arima(ts, order=c(4,1,0), seasonal = list(order = c(1,1,0), period = 12), method="ML"))

# Call:
# arima(x = ts, order = c(4, 1, 0), seasonal = list(order = c(1, 1, 0), period = 12),
#     method = "ML")
```

```
#
# Coefficients:
#          ar1      ar2     ar3      ar4     sar1
#       0.7335  -0.6861  0.4254  -0.2808  -0.5074
# s.e.  0.0428   0.0504  0.0504   0.0429   0.0394
#
# sigma^2 estimated as 0.01773:  log likelihood = 298.71,  aic = -585.43
# [1] -585.2564

####
arima(ts, order=c(9,1,0), seasonal = list(order = c(1,1,0), period = 12), method="ML")
AICc(arima(ts, order=c(9,1,0), seasonal = list(order = c(1,1,0), period = 12), method="ML"))

# Call:
# arima(x = ts, order = c(9, 1, 0), seasonal = list(order = c(1, 1, 0), period = 12),
#     method = "ML")
#
# Coefficients:
#          ar1      ar2     ar3      ar4     ar5      ar6     ar7      ar8     ar9     sar1
#       0.7922  -0.8072  0.6251  -0.5325  0.3504  -0.1942  0.2301  -0.1155  0.1195  -0.4932
# s.e.  0.0443   0.0565  0.0665   0.0719  0.0740   0.0720  0.0672   0.0582  0.0458   0.0400
#
# sigma^2 estimated as 0.01667:  log likelihood = 313.98,  aic = -605.95
# [1] -605.4159

####
arima(ts, order=c(11,1,0), seasonal = list(order = c(1,1,0), period = 12), method="ML")
AICc(arima(ts, order=c(11,1,0), seasonal = list(order = c(1,1,0), period = 12), method="ML"))
# Call:
# arima(x = ts, order = c(11, 1, 0), seasonal = list(order = c(1, 1, 0), period = 12),
#     method = "ML")
#
# Coefficients:
#          ar1      ar2     ar3      ar4     ar5      ar6     ar7      ar8     ar9    ar10     ar11     sar1
#       0.7938  -0.7919  0.6126  -0.5053  0.3270  -0.1520  0.1625  -0.0282  0.0070  0.1181  -0.1672  -0.4170
# s.e.  0.0439   0.0562  0.0667   0.0722  0.0753   0.0767  0.0764   0.0752  0.0705  0.0628   0.0533   0.0494
#
# sigma^2 estimated as 0.01633:  log likelihood = 318.99,  aic = -611.97
# [1] -611.2286

####
arima(ts, order=c(4,1,0), seasonal = list(order = c(2,1,0), period = 12), method="ML")
AICc(arima(ts, order=c(4,1,0), seasonal = list(order = c(2,1,0), period = 12), method="ML"))

# Call:
# arima(x = ts, order = c(4, 1, 0), seasonal = list(order = c(2, 1, 0), period = 12),
#     method = "ML")
#
# Coefficients:
#          ar1      ar2     ar3      ar4     sar1     sar2
#       0.7250  -0.6975  0.4061  -0.2824  -0.6893  -0.3590
# s.e.  0.0428   0.0505  0.0505   0.0430   0.0432   0.0434
#
# sigma^2 estimated as 0.01555:  log likelihood = 330.14,  aic = -646.27
# [1] -646.0446

########## Pure AR component raised the AICc too much,

# Trying SARIMA models:
#

####
arima(ts, order=c(4,1,1), seasonal = list(order = c(1,1,1), period = 12), method="ML")
AICc(arima(ts, order=c(4,1,1), seasonal = list(order = c(1,1,1), period = 12), method="ML"))

# Call:
# arima(x = ts, order = c(4, 1, 1), seasonal = list(order = c(1, 1, 1), period = 12),
#     method = "ML")
#
# Coefficients:
#          ar1      ar2     ar3      ar4     ma1     sar1     sma1
#       0.1897  -0.2963  0.0890  -0.1330  0.6277  -0.0418  -0.8413
# s.e.  0.1067   0.0893  0.0804   0.0621  0.1026   0.0543   0.0320
#
# sigma^2 estimated as 0.01308:  log likelihood = 369.35,  aic = -722.69
# [1] -722.4023

####
arima(ts, order=c(4,1,2), seasonal = list(order = c(1,1,1), period = 12), method="ML")
AICc(arima(ts, order=c(4,1,2), seasonal = list(order = c(1,1,1), period = 12), method="ML"))
```

```
# Call:
# arima(x = ts, order = c(4, 1, 2), seasonal = list(order = c(1, 1, 1), period = 12),
#     method = "ML")
#
# Coefficients:
#          ar1      ar2     ar3      ar4     ma1      ma2     sar1     sma1
#       0.3164  -0.2983  0.1078  -0.1293  0.5003  -0.1029  -0.0403  -0.8425
# s.e.  0.2659   0.0881  0.0853   0.0611  0.2651   0.2080   0.0544   0.0320
#
# sigma^2 estimated as 0.01307:  log likelihood = 369.47,  aic = -720.94
# [1] -720.5743

####
arima(ts, order=c(4,1,7), seasonal = list(order = c(1,1,1), period = 12), method="ML")
AICc(arima(ts, order=c(4,1,7), seasonal = list(order = c(1,1,1), period = 12), method="ML"))

# Call:
# arima(x = ts, order = c(4, 1, 7), seasonal = list(order = c(1, 1, 1), period = 12),
#     method = "ML")
#
# Coefficients:
# Warning: NaNs produced         ar1      ar2    ar3      ar4      ma1      ma2      ma3     ma4     ma5     ma6
ma7     sar1     sma1
#       1.1192  -0.8434  0.986  -0.7759  -0.3083  -0.2095  -0.3161  0.0104  0.5947  0.0955  -0.0464  -0.0129  -0.
8654
# s.e.     NaN      NaN    NaN   0.0885      NaN      NaN   0.1276     NaN  0.0705  0.0333   0.0428   0.0563   0.
0359
#
# sigma^2 estimated as 0.01281:  log likelihood = 373.99,  aic = -719.98
# [1] -719.1221

####
arima(ts, order=c(9,1,7), seasonal = list(order = c(1,1,1), period = 12), method="ML")
AICc(arima(ts, order=c(9,1,7), seasonal = list(order = c(1,1,1), period = 12), method="ML"))
# Warning: possible convergence problem: optim gave code = 1
# Call:
# arima(x = ts, order = c(9, 1, 7), seasonal = list(order = c(1, 1, 1), period = 12),
#     method = "ML")
#
# Coefficients:
#          ar1     ar2     ar3      ar4      ar5     ar6     ar7      ar8     ar9     ma1      ma2      ma3     m
a4     ma5     ma6      ma7
#       0.4660  0.3075  0.1504  -0.1695  -0.7167  0.2974  0.0327  -0.0584  0.0283  0.3497  -0.8242  -0.5333  0.12
40  0.9116  0.2972  -0.3040
# s.e.  0.3968  0.1344  0.1093   0.1452   0.1232  0.3453  0.0989   0.0882  0.1174  0.4037   0.3535   0.1666  0.16
89  0.1661  0.3531   0.3224
#          sar1     sma1
#       -0.0439  -0.8669
# s.e.   0.0739   0.0354
#
# sigma^2 estimated as 0.01272:  log likelihood = 375.61,  aic = -713.21
# Warning: possible convergence problem: optim gave code = 1[1] -711.6419

####
arima(ts, order=c(9,1,2), seasonal = list(order = c(1,1,1), period = 12), method="ML")
AICc(arima(ts, order=c(9,1,2), seasonal = list(order = c(1,1,1), period = 12), method="ML"))

# Call:
# arima(x = ts, order = c(9, 1, 2), seasonal = list(order = c(1, 1, 1), period = 12),
#     method = "ML")
#
# Coefficients:
#          ar1      ar2     ar3      ar4     ar5      ar6     ar7      ar8     ar9      ma1     ma2     sar1
sma1
#       0.9032  -0.9741  0.7414  -0.6327  0.4185  -0.2652  0.2446  -0.1523  0.1156  -0.0923  0.1000  -0.0134  -0.
8392
# s.e.  0.7274   0.5189  0.4284   0.4235  0.3043   0.2634  0.1623   0.0812  0.0731   0.7303  0.8086   0.0996   0.
0329
#
# sigma^2 estimated as 0.01293:  log likelihood = 372.27,  aic = -716.53
# [1] -715.6741
```

```
# Taking the models with lowest AICc and setting possible coefficients to 0

## Model A, further exploration
arima(ts, order=c(4,1,1), seasonal = list(order = c(1,1,1), period = 12), method="ML")
AICc(arima(ts, order=c(4,1,1), seasonal = list(order = c(1,1,1), period = 12), method="ML"))
# can take ar1, ar3 as 0

arima(ts, order=c(4,1,1), seasonal = list(order = c(1,1,1), period = 12), fixed = c(NA, NA, 0, NA , NA , NA ,NA),
method="ML")
AICc(arima(ts, order=c(4,1,1), seasonal = list(order = c(1,1,1), period = 12), fixed = c(NA, NA, 0, NA , NA , NA
,NA), method="ML"))
# taking ar3 as 0 lowered the AICc so we will move forward with that.
# -723.1316

## Model B, further exploration
arima(ts, order=c(4,1,2), seasonal = list(order = c(1,1,1), period = 12), method="ML")
AICc(arima(ts, order=c(4,1,2), seasonal = list(order = c(1,1,1), period = 12), method="ML"))
# can take ar1, ar3, ma1, as 0


arima(ts, order=c(4,1,2), seasonal = list(order = c(1,1,1), period = 12),fixed = c(0,NA,0,NA,NA,NA,NA,NA) ,method
="ML")
AICc(arima(ts, order=c(4,1,2), seasonal = list(order = c(1,1,1), period = 12), fixed = c(0,NA,0,NA,NA,NA,NA,NA),
method="ML"))

#taking them as 0 lowers the AICc so we will proceed with this.
#-723.2003

## Model C
arima(ts, order=c(0,1,7), seasonal = list(order = c(0,1,1), period = 12), method="ML")
AICc(arima(ts, order=c(0,1,7), seasonal = list(order = c(0,1,1), period = 12), method="ML"))
# we can take ma4, ma5, ma6 as 0

arima(ts, order=c(0,1,7), seasonal = list(order = c(0,1,1), period = 12), fixed = c(NA,NA,NA,0,0,0,NA,NA),method
="ML")
AICc(arima(ts, order=c(0,1,7), seasonal = list(order = c(0,1,1), period = 12), fixed = c(NA,NA,NA,0,0,0,NA,NA), m
ethod="ML"))

# Taking the three of them as 0 gives the lowest AICc, so we move forward with this.
#-723.7441
```

```
#Model A
# WARNING Warning: seasonal MA part of model is not invertible!!!!!!!!!
source("plot.roots.r")
arima(ts, order=c(4,1,1), seasonal = list(order = c(1,1,1), period = 12), fixed = c(0, NA, NA, NA , NA , NA ,NA),
method="ML")

#roots of ar part:
plot.roots(NULL, polyroot(c(1, 0 , -0.1785 , -0.0221 , -0.0610)), main="(A) roots of AR part")
# roots of AR appear to be outside unit circle, indicating stationarity

#ma part need not show roots, the ma part is invertible and always stationary

# sar 1 part is stationary
# sma1 part is not invertible!!!! so the model is not invertible.
```

```
# Model B
arima(ts, order=c(4,1,2), seasonal = list(order = c(1,1,1), period = 12),fixed = c(0,NA,0,NA,NA,NA,NA,NA) ,method
="ML")

## roots of AR part
plot.roots(NULL, polyroot(c(1, 0 , -0.2364  ,  0 , -0.0965)), main="(B) roots of AR part")
# roots are outside circle, and thus AR(4) is stationary

## rots of MA(2) part
plot.roots(NULL, polyroot(c(1, 0.8150 , 0.0924)), main="(B) roots of MA part")
# roots are outside of circle, indicating MA part is invertible

# sar and sma parts both indicate stationarity and invertibility
```

```
# Model C
arima(ts, order=c(0,1,7), seasonal = list(order = c(0,1,1), period = 12), fixed = c(NA,NA,NA,0,0,0,NA,NA),method
="ML")

# roots of ma part
plot.roots(NULL, polyroot(c(1,  0.8155 , -0.1519 , -0.1852  ,  0 ,   0  ,  0 , 0.0217)), main="(A) roots of MA pa
rt")
# roots are all outside of circle this indicating invertibility - MA models are stationary by default

# sma part is also invertible and stationary
```

```
# Fitting and diagnostic checking of model A
fitA <- arima(ts, order=c(4,1,1), seasonal = list(order = c(1,1,1), period = 12), fixed = c(0, NA, NA, NA , NA ,
NA ,NA), method="ML")
res <- residuals(fitA)

par(mar=c(5,6,4,1)+.1)
hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(res)
fitt <- lm(res ~ as.numeric(time(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
qqnorm(res,main= "Normal Q-Q Plot for Model A")
qqline(res,col="blue")
acf(res, lag.max=40)
pacf(res, lag.max=40)
shapiro.test(res)
Box.test(res, lag = 12, type = c("Box-Pierce"), fitdf = 6)
Box.test(res, lag = 12, type = c("Ljung-Box"), fitdf = 6)
Box.test(res^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
acf(res^2, lag.max=40)
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
# Fitting and diagnostic checking of Model B
par(mar=c(5,6,4,1)+.1)
fitB <- arima(ts, order=c(4,1,2), seasonal = list(order = c(1,1,1), period = 12),fixed = c(0,NA,0,NA,NA,NA,NA,NA)
,method="ML")
res <- residuals(fitB)

hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(res)
fitt <- lm(res ~ as.numeric(time(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
qqnorm(res,main= "Normal Q-Q Plot for Model A")
qqline(res,col="blue")
acf(res, lag.max=40)
pacf(res, lag.max=40)
shapiro.test(res)
Box.test(res, lag = 12, type = c("Box-Pierce"), fitdf = 4)
Box.test(res, lag = 12, type = c("Ljung-Box"), fitdf = 4)
Box.test(res^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
acf(res^2, lag.max=40)
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
# Fitting and diagnostic checking of model C
par(mar=c(5,6,4,1)+.1)
fitC <- arima(ts, order=c(0,1,7), seasonal = list(order = c(0,1,1), period = 12), fixed = c(NA,NA,NA,0,0,0,NA,N
A),method="ML")
res <- residuals(fitC)

hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(res)
fitt <- lm(res ~ as.numeric(time(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
qqnorm(res,main= "Normal Q-Q Plot for Model A")
qqline(res,col="blue")
acf(res, lag.max=40)
pacf(res, lag.max=40)
shapiro.test(res)
Box.test(res, lag = 12, type = c("Box-Pierce"), fitdf = 5)
Box.test(res, lag = 12, type = c("Ljung-Box"), fitdf = 5)
Box.test(res^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
acf(res^2, lag.max=40) # acf of resid squares suggests maybe MA(8) componenent?
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
# Fitting and prediction
# Graph without future data
pred.tr <- predict(fitC, n.ahead = 12)
U.tr = pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
ts.plot(ts, xlim=c(2018,2024), ylim = c(400,max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")

points(seq(from = 2022 + (2-1)/12, to = 2023 + (1-1)/12, by = 1/12), pred.tr$pred, col="red")

# Graph with future data
pred.tr <- predict(fitC, n.ahead = 12)
U.tr = pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
ts.plot(ts_ALL, xlim=c(2018,2024), ylim = c(400,max(U.tr)), col = "green")
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")


points(seq(from = 2022 + (2-1)/12, to = 2023 + (1-1)/12, by = 1/12), pred.tr$pred, col="red")
```