MATH 8090: Autoregressive Integrated Moving Average (ARIMA) Models and Seasonal ARIMA Models

Whitney Huang, Clemson University

10/7-10/9/2025

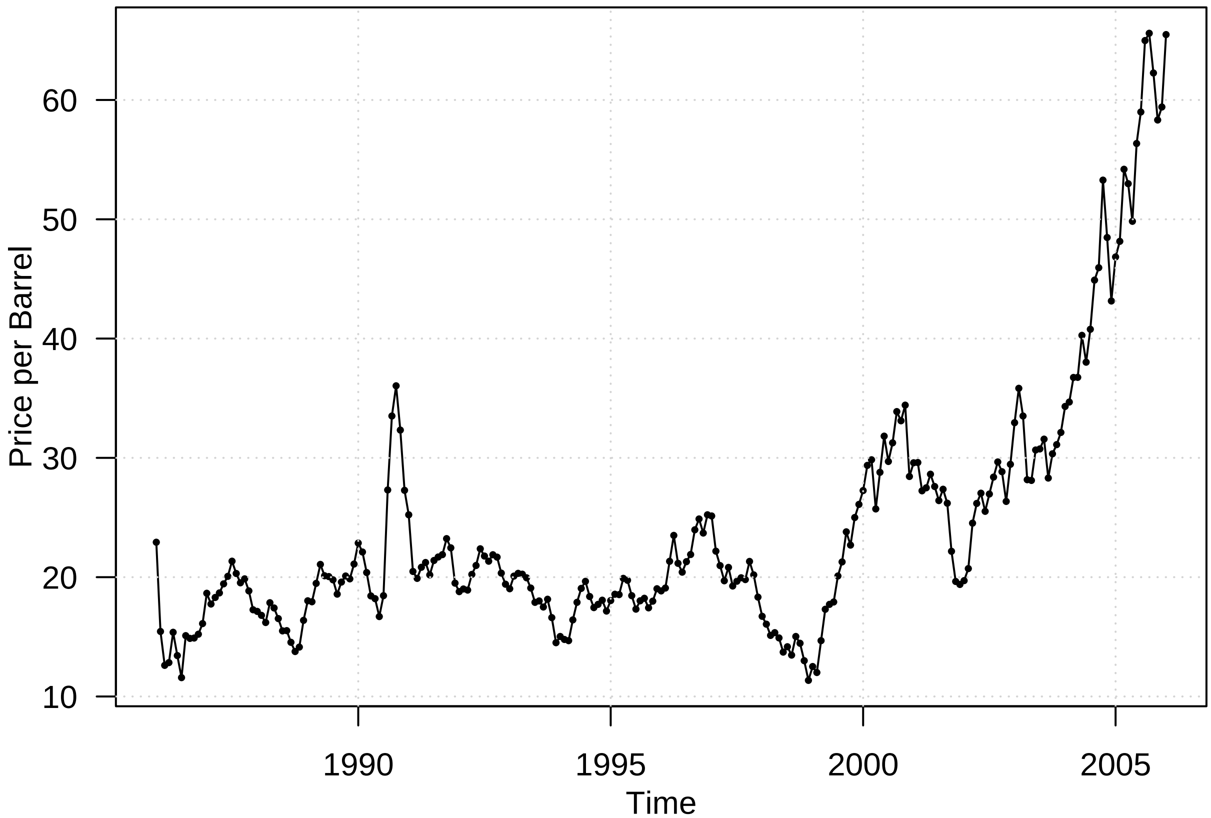
Contents

- ARIMA . . . 1
  - Monthly Price of Oil: January 1986–January 2006 . . . 1
  - An explosive AR model . . . 2
- Seasonal Autoregressive Integrated Moving Average (SARIMA) . . . 4
  - Stochastic and Deterministic Trends . . . 4
  - SARIMA Simulation . . . 5
  - Simulating a Seasonal AR Model . . . 6
  - Simulating a Seasonal MA Model . . . 7
  - Simulating a SARIMA Model . . . 8
- Monthly International Airline Passenger Data . . . 9
  - Read the data . . . 9
  - Plot the time series . . . 10
  - Plot sample ACF/PACF . . . 11
  - Trying Different Orders of Differencing . . . 12
  - Plot ACF and PACF . . . 12
  - Show the ACF and PACF for the $d = 1$, $D = 0$ case. . . . 14
  - A useful function for the model diagnostics (courtesy of Peter Craigmile) . . . 14
  - Fitting the SARIMA$(1, 1, 0) \times (1, 0, 0)$ model . . . 15
  - Fitting the SARIMA$(0, 1, 0) \times (1, 0, 0)$ model . . . 16
  - Forecasting 1971 Data . . . 17
  - Evaluating Forecast Performance . . . 20

ARIMA

Monthly Price of Oil: January 1986–January 2006

```r
library(TSA)
data(oil.price)

par(las = 1, mgp = c(2, 1, 0), mar = c(3.5, 3.5, 0.8, 0.6))
plot(oil.price, ylab = 'Price per Barrel', type = 'l')
points(oil.price, pch = 16, cex = 0.5)
grid()
```

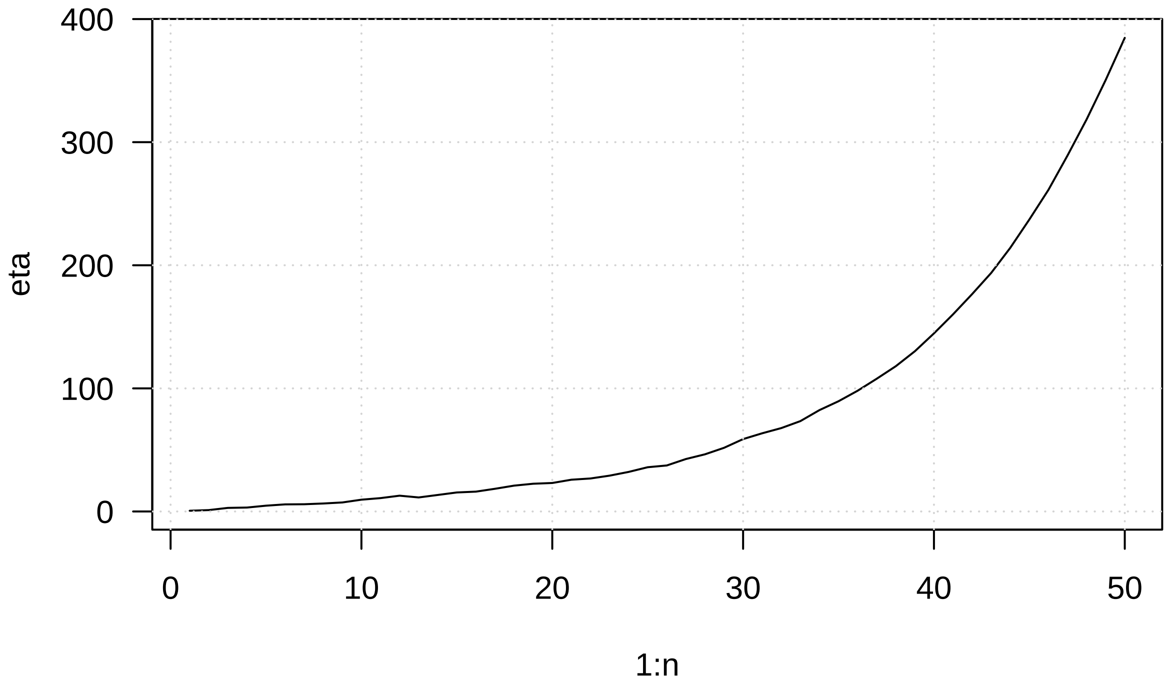


A stationary model does not seem to be reasonable. However, it is also not clear which (deterministic) trend model is appropriate

An explosive AR model

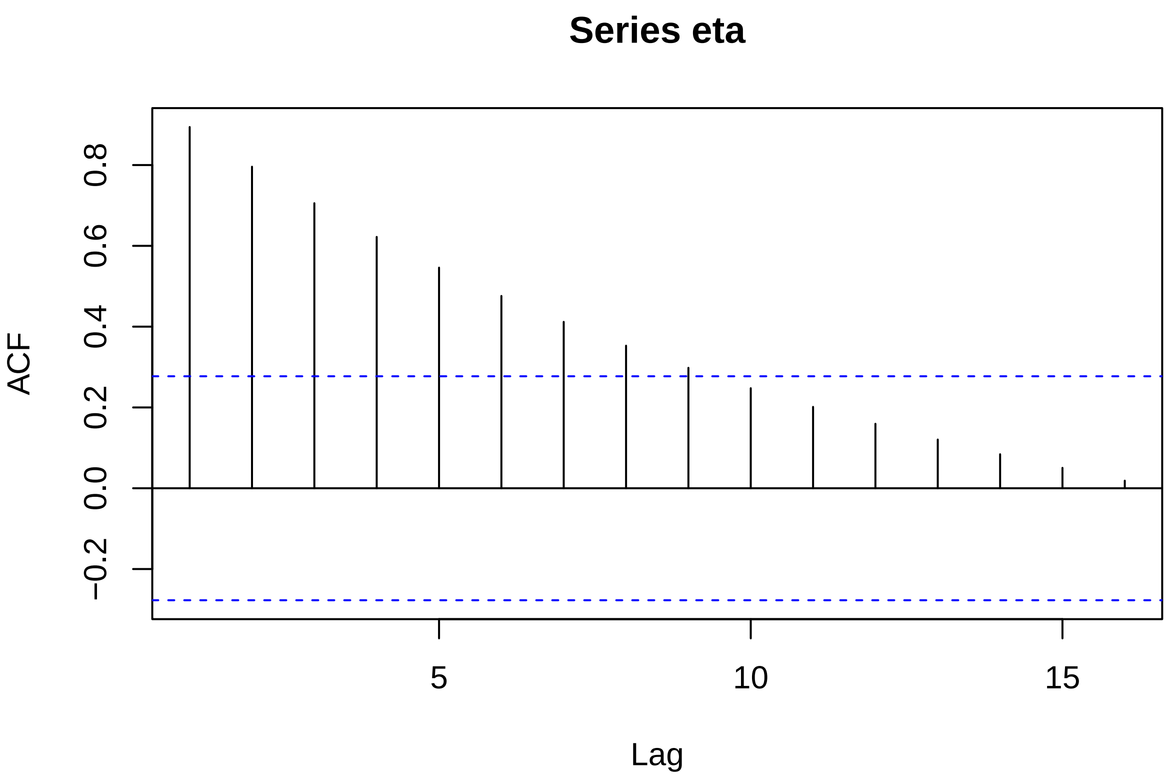
$\eta_t = 1.1\eta_{t-1} + Z_t$

```r
n <- 50; phi <- 1.1
set.seed(128)
z <- rnorm(n)
eta <- c()
eta[1] <- z[1]
for (i in 2:n) eta[i] <- phi * eta[i - 1] + z[i]
plot(1:n, eta, las = 1, type = "l")
grid()
```



400
300
200
100
0
eta
0
10
20
30
40
50
1:n


```r
acf(eta)
```



Series eta
ACF
0.8
0.6
0.4
0.2
0.0
−0.2
5
10
15
Lag

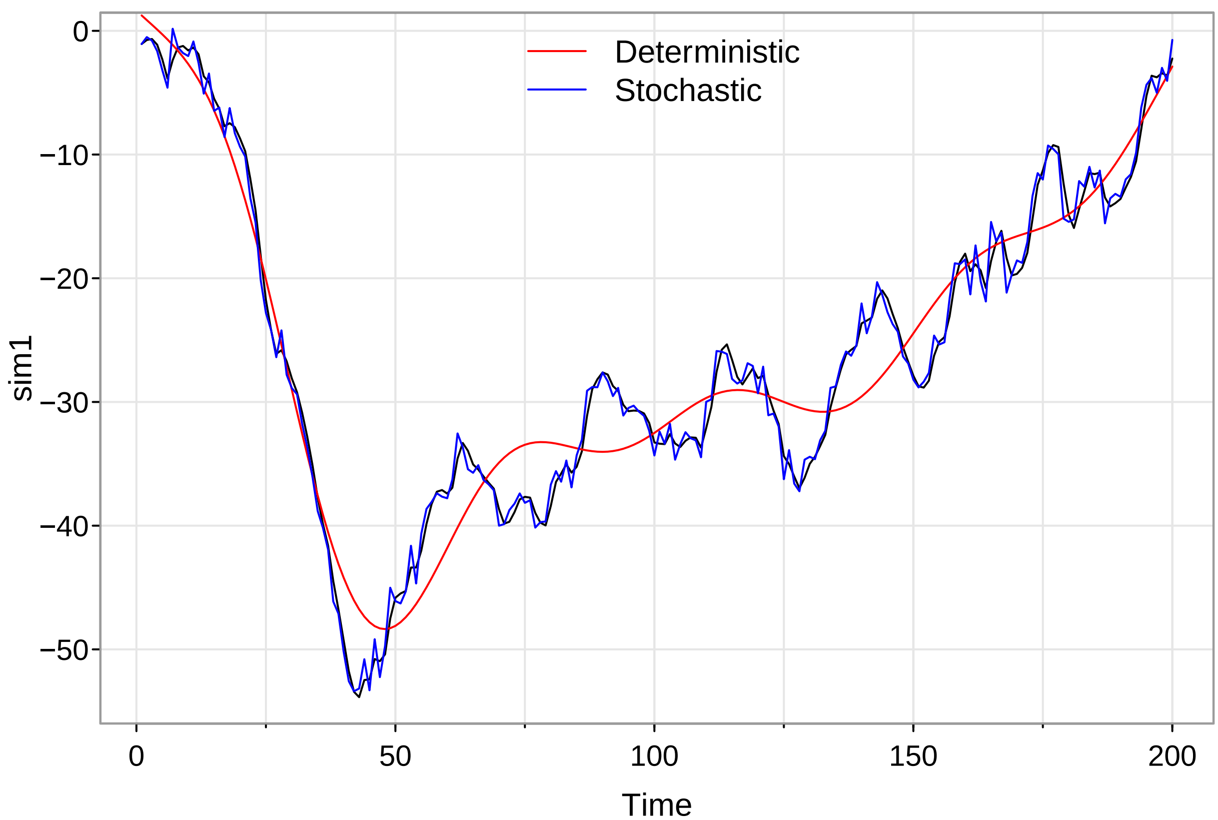
Seasonal Autoregressive Integrated Moving Average (SARIMA)

Stochastic and Deterministic Trends

```r
library(astsa)
set.seed(1234)
n = 200
t <- 1:n
sim1 <- arima.sim(list(order = c(1, 1, 0), ar = 0.6), n = n)[-1]

par(las = 1, mar = c(3.5, 3.5, 1, 0.5))
tsplot(sim1)
# Fit a deterministic trend
library(mgcv)
```

```
## Loading required package: nlme

## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```r
dFit <- gam(sim1 ~ s(t, k = 10))
lines(t, predict(dFit), col = "red")
# Fit a stochastic trend
sFit <- arima(sim1, order = c(1, 1, 0))
lines(t, sFit$residuals + sim1, col = "blue")
legend("top", legend = c("Deterministic", "Stochastic"),
       col = c("red", "blue"), lty = 1, bty = "n")
```

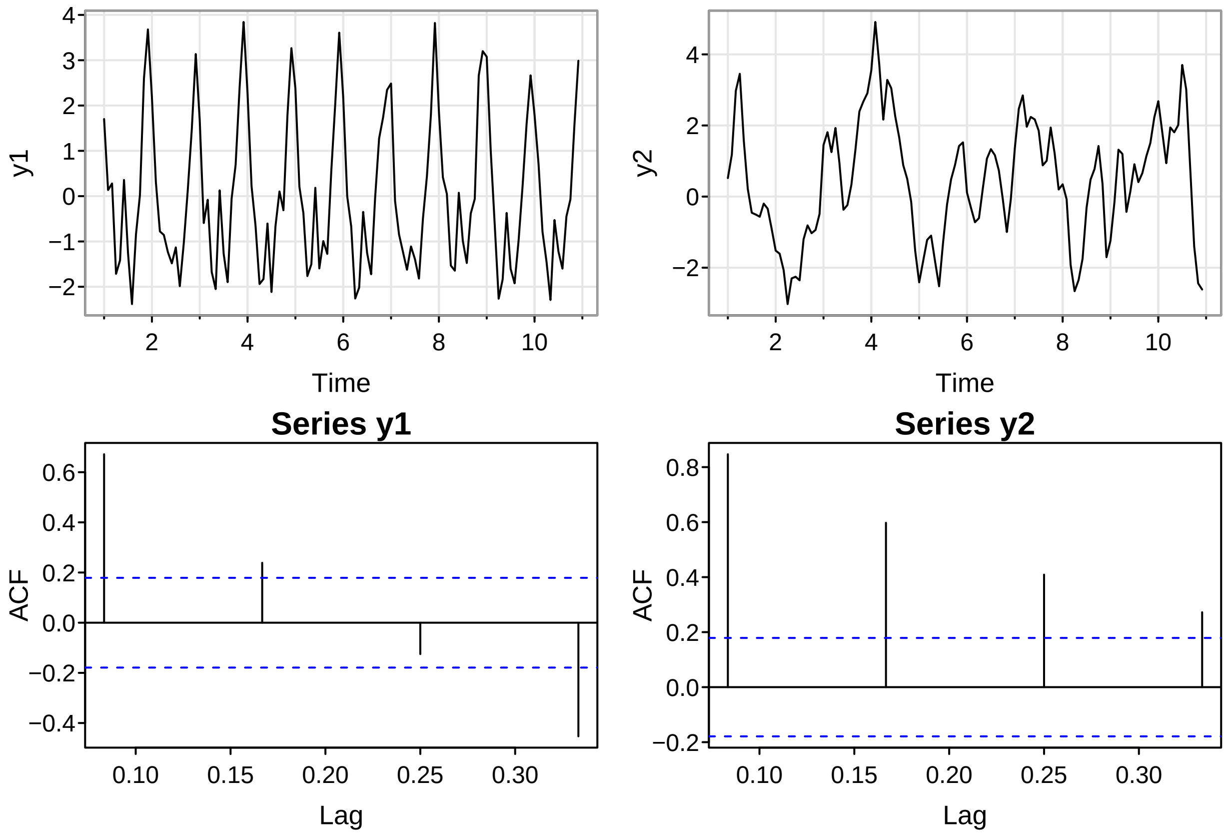


SARIMA Simulation

```r
n = 120
t <- 1:n
# Deterministic seasonality
season_d <- 2 * cos(2 * pi * (t / 12)) + 1 * cos(2 * pi * (t / 6)) + 0.5 * cos(2 * pi * (t / 3))
set.seed(123)
y1 = season_d + rnorm(n, sd = 0.5)
# Convert to a time series with monthly frequency
y1 <- ts(y1, frequency = 12, start = 1)

par(las = 1, mfrow = c(2, 2))
tsplot(y1)
library(forecast)
(sarma_model <- Arima(y1, order = c(1, 0, 1), seasonal = c(1, 0, 0)))
```

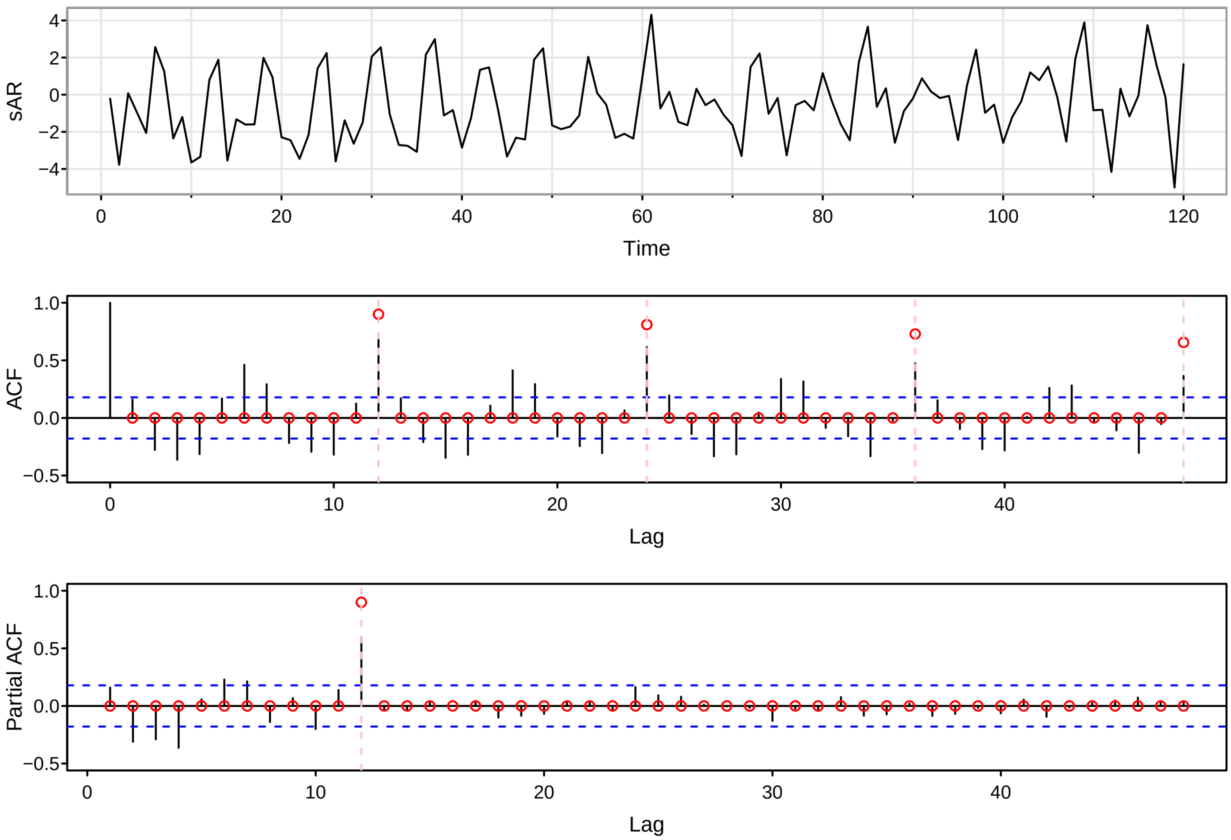
```
## Series: y1
## ARIMA(1,0,1)(1,0,0)[12] with non-zero mean
##
## Coefficients:
##           ar1     ma1    sar1    mean
##       -0.8183  1.0000  0.9135  0.0027
## s.e.   0.0554  0.0124  0.0289  0.4161
##
## sigma^2 = 0.4135:  log likelihood = -125.5
## AIC=261   AICc=261.53   BIC=274.94
```

```r
set.seed(12)
# Stochastic seasonality
m <- list(order = c(1, 0, 1),
          seasonal = list(order = c(1, 0, 0), period = 12),
          ar = c(0.8), ma = c(0.95), sar = c(0.9))
# Simulate the SARIMA model
y2 <- arima.sim(model = m, n = n, sd = 0.75)
# Convert to a time series with monthly frequency
y2 <- ts(y2, frequency = 12, start = 1)
tsplot(y2)

acf(y1, lag.max = 4)
acf(y2, lag.max = 4)
```



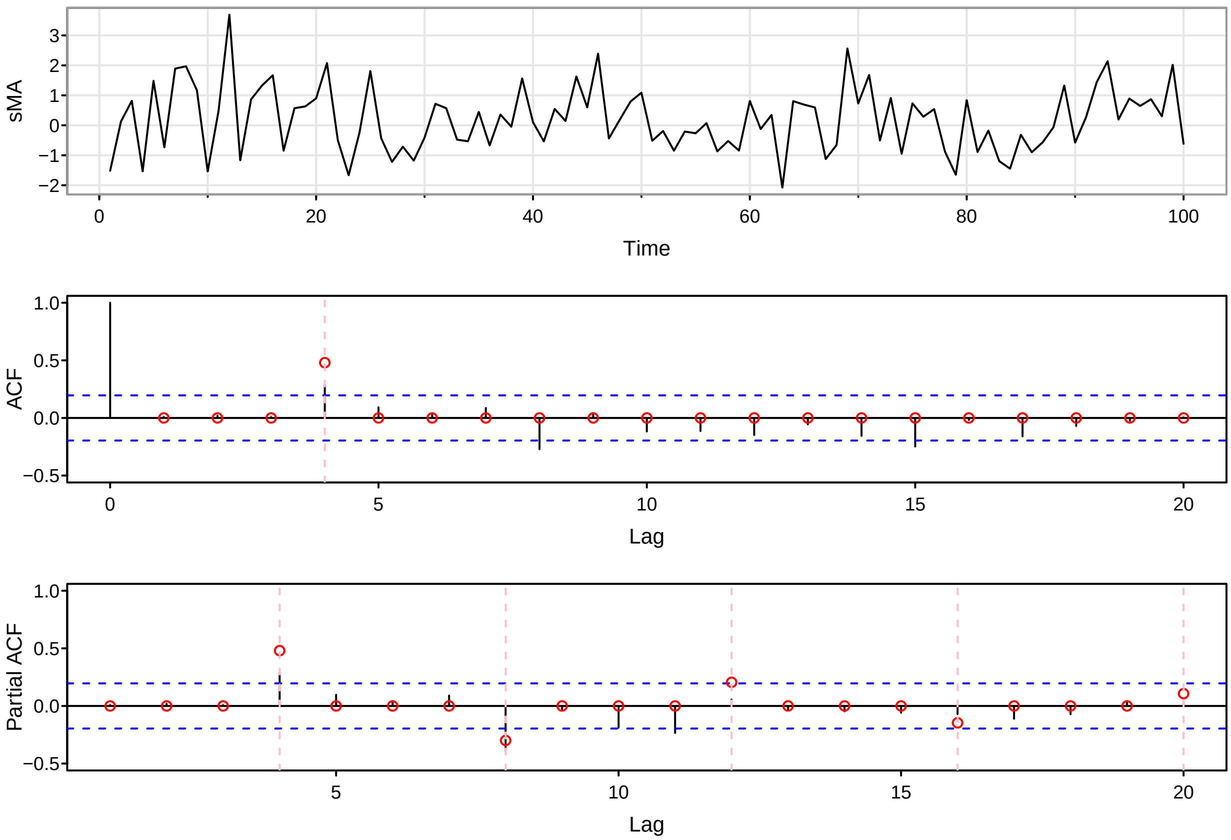
Simulating a Seasonal AR Model

```r
n = 120; Phi = 0.9
set.seed(1234)
sAR = sarima.sim(sar = Phi, S = 12, n = n)
sAR <- ts(sAR, frequency = 1, start = 1)
par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2.5, 1, 0), mfrow = c(3, 1))
tsplot(sAR, xlab = "Time")
stats::acf(sAR, lag.max = 48, ylim = c(-0.5, 1))
trueACF <- ARMAacf(ar = c(rep(0, 11), Phi), lag.max = 48)
points(1:48, trueACF[2:49], col = "red")
abline(v = 12 * (1:4), col = "pink", lty = 2)
stats::pacf(sAR, lag.max = 48, ylim = c(-0.5, 1))
truePACF <- ARMAacf(ar = c(rep(0, 11), Phi), lag.max = 48, pacf = T)
points(1:48, truePACF, col = "red")
abline(v = 12, col = "pink", lty = 2)
```



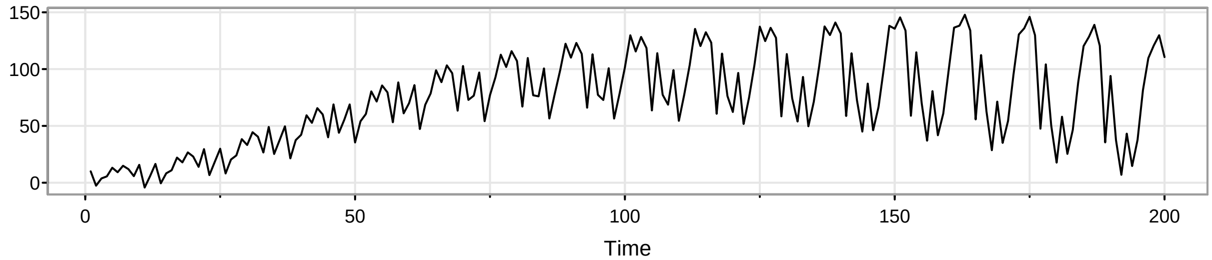
Simulating a Seasonal MA Model

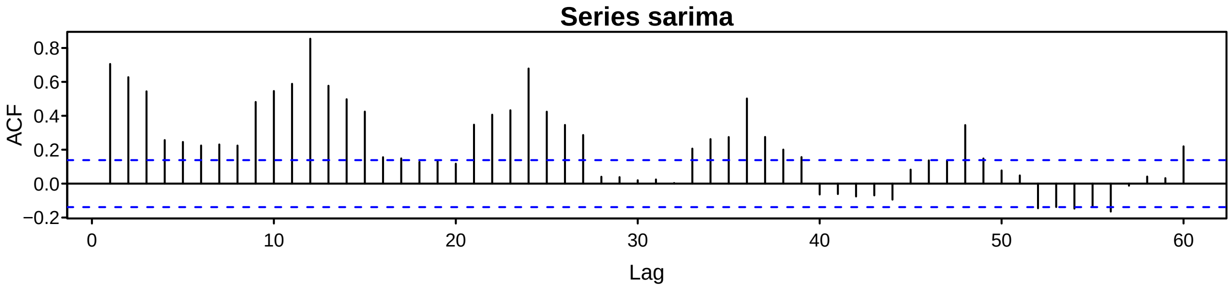
```
n = 100; Theta = 0.75
set.seed(1234)
sMA = sarima.sim(sma = Theta, S = 4, n = n)
sMA <- ts(sMA, frequency = 1, start = 1)
par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2.5, 1, 0), mfrow = c(3, 1))
tsplot(sMA, xlab = "Time")
stats::acf(sMA, lag.max = 20, ylim = c(-0.5, 1))
trueACF <- ARMAacf(ma = c(rep(0, 3), Theta), lag.max = 20)
points(1:20, trueACF[2:21], col = "red")
abline(v = 4, col = "pink", lty = 2)
stats::pacf(sMA, lag.max = 20, ylim = c(-0.5, 1))
truePACF <- ARMAacf(ma = c(rep(0, 3), Theta), lag.max = 20, pacf = T)
points(1:20, truePACF, col = "red")
abline(v = 4 * (1:5), col = "pink", lty = 2)
```

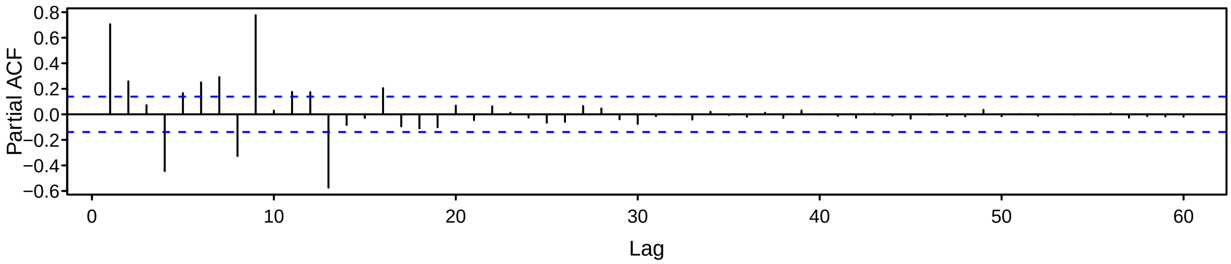


Simulating a SARIMA Model

```
par(las = 1, mar = c(3.5, 3.5, 1, 0.5), mgp = c(2.5, 1, 0), mfrow = c(3, 1))
set.seed(123)
sarima <- sarima.sim(d = 1, ar = -.25, sar = .9, D = 1, sma = 0.75, S = 12, n = 200)
sarima <- ts(sarima, frequency = 1, start = 1)
tsplot(sarima, ylab = "")
acf(sarima, lag.max = 60)
pacf(sarima, lag.max = 60)
```



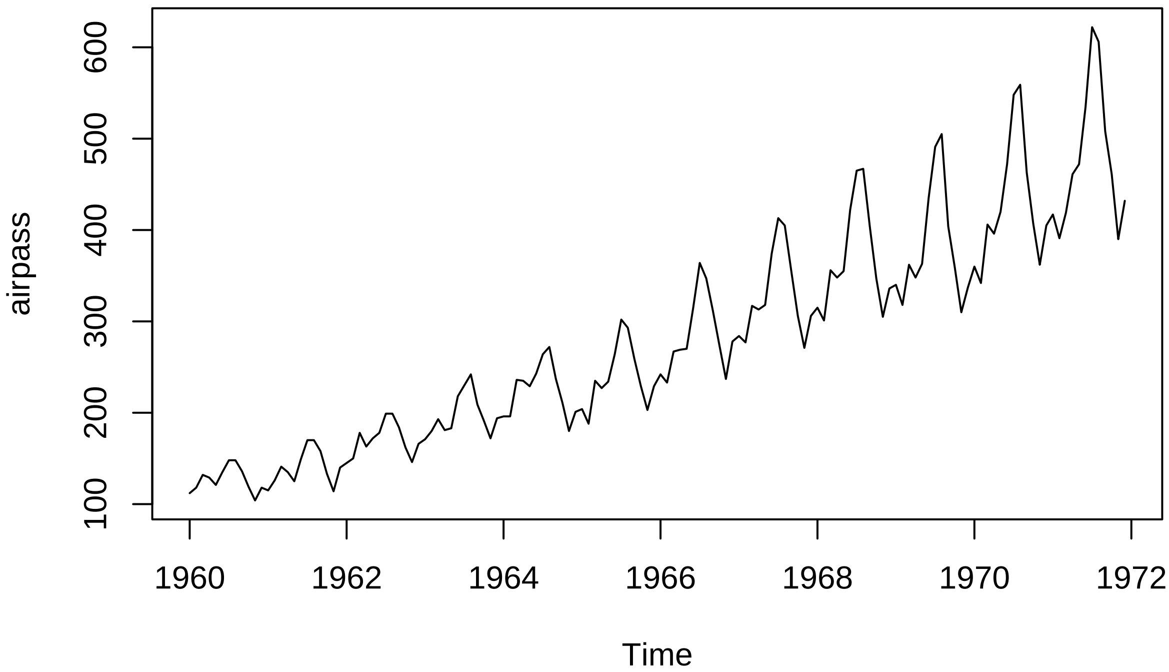




Monthly International Airline Passenger Data

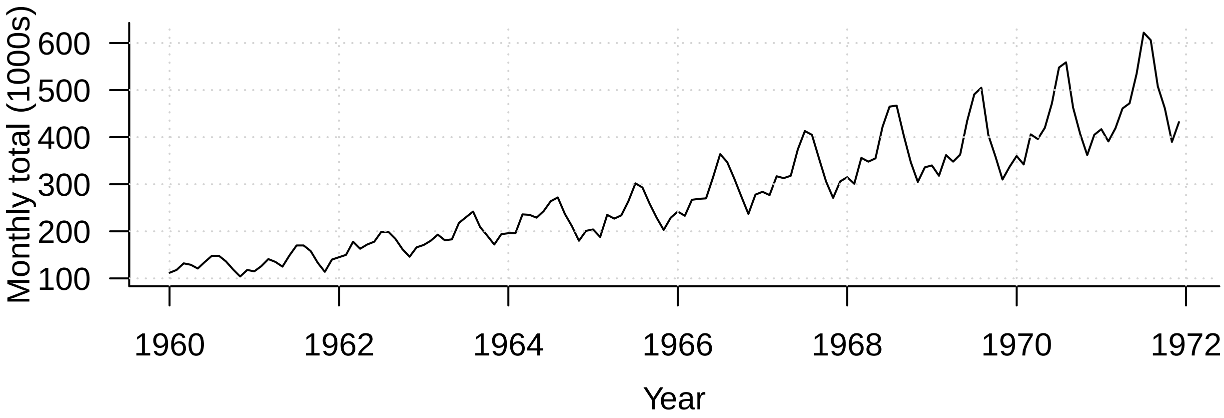
Read the data

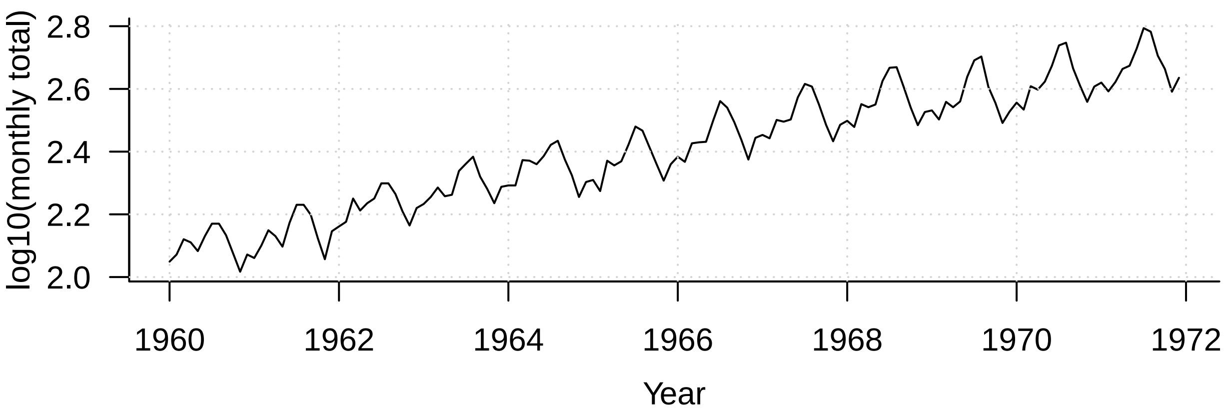
```
library(TSA)
data(airpass)
plot(airpass)
```



Plot the time series

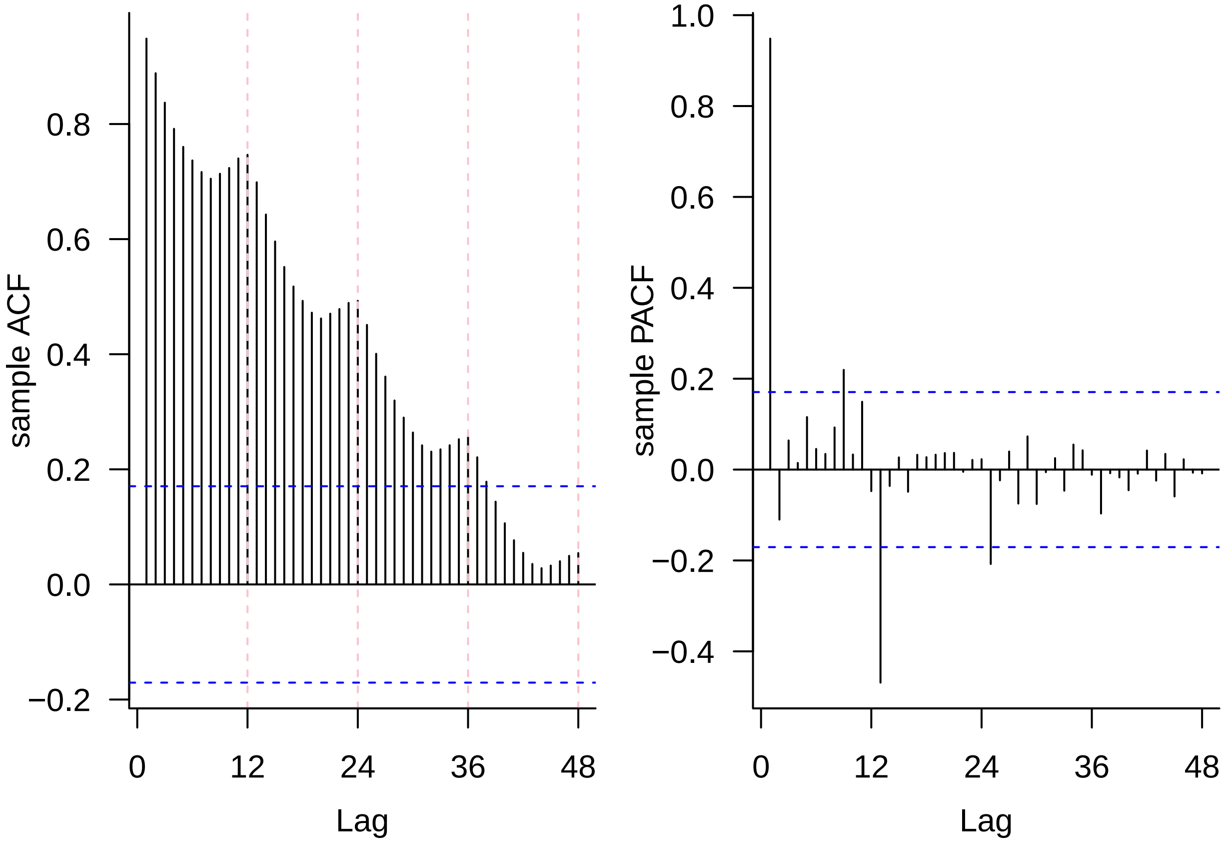
```
par(bty = "L", mar = c(3.6, 3.5, 0.8, 0.6), mgp = c(2.4, 1, 0), las = 1, mfrow = c(2, 1))
## plot the time series.
plot(airpass, xlab = "Year", ylab = "Monthly total (1000s)")
grid()
## take a log (to the base 10) of the air passenger data.
log.airpass <- log10(airpass)
plot(log.airpass, type = "l", xlab = "Year", ylab = "log10(monthly total)")
grid()
```





Plot sample ACF/PACF

```
yr <- time(airpass)
log.shortair <- log.airpass[1:132]
shortyears <- yr[1:132]
par(bty = "L", mar = c(3.6, 3.5, 0.8, 0.6), mgp = c(2.4, 1, 0), las = 1, mfrow = c(1, 2))
acf(log.shortair, ylab = "sample ACF", main = "", lag.max = 48, xaxt = "n")
abline(v = 12 * (1:4), col = "pink", lty = 2)
axis(side = 1, at = seq(0, 48, 12))
pacf(log.shortair, ylab = "sample PACF", main = "", lag.max = 48, xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
```



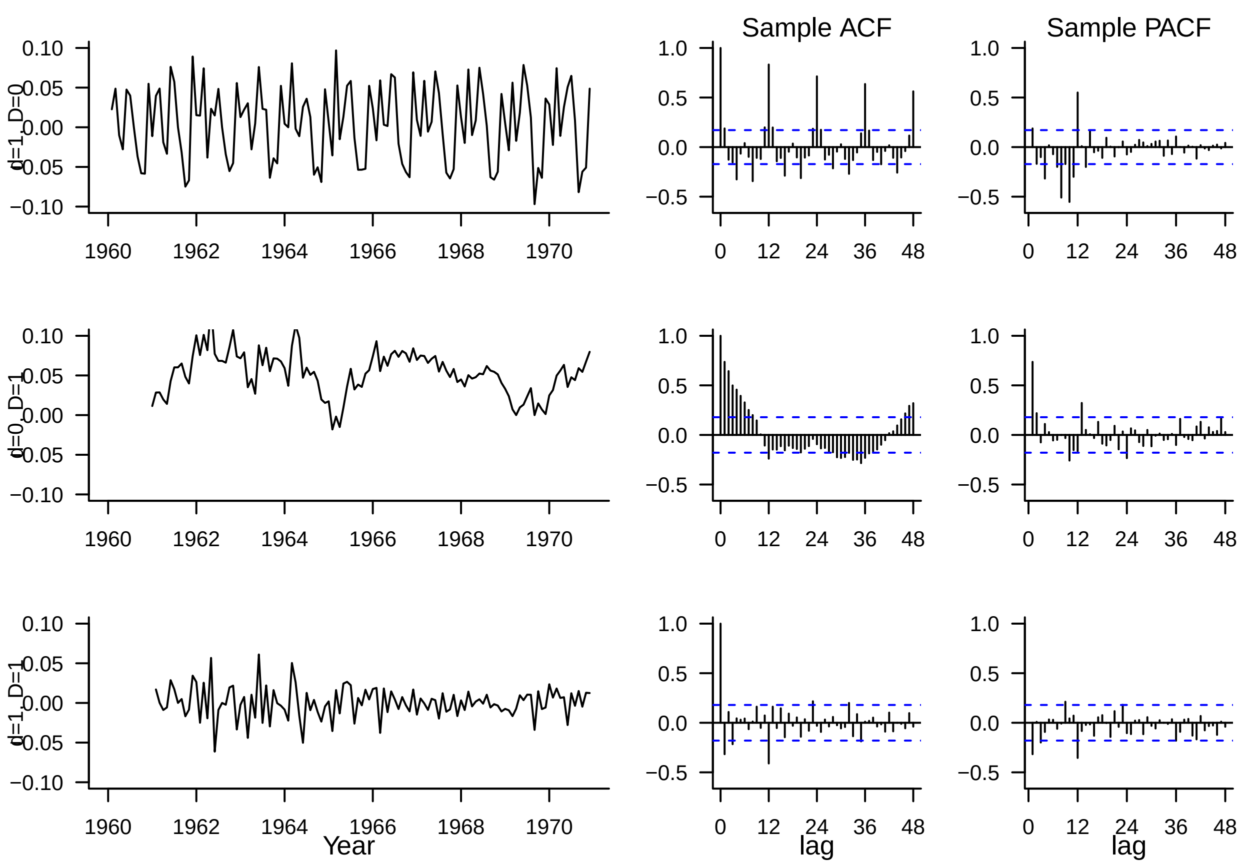
Trying Different Orders of Differencing

```
## take the differences Y_t = (1-B) X_t
diff.1.0 <- diff(log.shortair)
## take the seasonal differences Y_t =(1-B^(12)) X_t
diff.0.1 <- diff(log.shortair, lag = 12, diff = 1)
## take the differences Y_t = (1-B^(12)) (1-B) X_t
diff.1.1 <- diff(diff(log.shortair, lag = 12, diff = 1))
```

Plot ACF and PACF

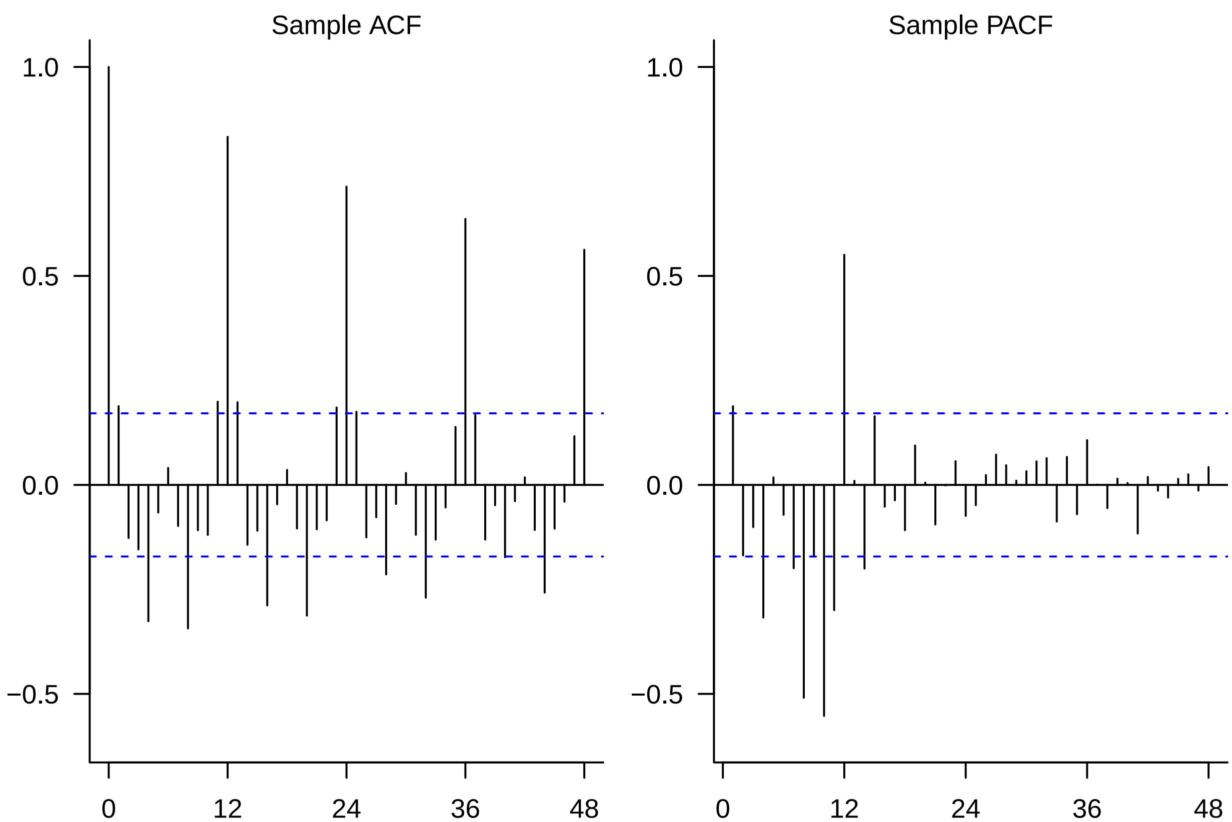
```
par(bty = "L", mar = c(3.6, 3.5, 1, 0.6), mgp = c(2.4, 1, 0), las = 1)
layout.matrix <- matrix(c(1, 1, 2, 3, 4, 4, 5, 6, 7, 7, 8, 9), nrow = 3, ncol = 4, byrow = T)
layout(mat = layout.matrix)
plot(shortyears[-1], diff.1.0, xlab = "", ylab = "d=1, D=0",
     type = "l", ylim = c(-0.1, 0.1), xlim = range(shortyears))

stats::acf(diff.1.0, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
mtext("Sample ACF", side = 3, line = 0, cex = 0.8)

stats::pacf(diff.1.0, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
```

```r
mtext("Sample PACF", side = 3, line = 0, cex = 0.8)

plot(shortyears[-c(1:12)], diff.0.1, xlab = "", ylab = "d=0, D=1",
     type = "l", ylim = c(-0.1, 0.1), xlim = range(shortyears))

stats::acf(diff.0.1, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))


stats::pacf(diff.0.1, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))

plot(shortyears[-c(1:13)], diff.1.1, xlab = "", ylab = "d=1, D=1",
     type = "l", ylim = c(-0.1, 0.1), xlim = range(shortyears))
mtext("Year", side = 1, line = 1.8, cex = 0.8)

stats::acf(diff.1.1, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
mtext("lag", side = 1, line = 1.8, cex = 0.8)

stats::pacf(diff.1.1, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
mtext("lag", side = 1, line = 1.8, cex = 0.8)
```



Show the ACF and PACF for the $d = 1$, $D = 0$ case.

```r
par(mfrow = c(1, 2), cex = 0.8, bty = "L", mar = c(3.6, 3, 1, 0.6), mgp = c(2.4, 1, 0), las = 1)
stats::acf(diff.1.0, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
mtext("Sample ACF", side = 3, cex = 0.8)

stats::pacf(diff.1.0, lag.max = 48, ylab = "", xlab = "", main = "", ylim = c(-0.6, 1), xaxt = "n")
axis(side = 1, at = seq(0, 48, 12))
mtext("Sample PACF", side = 3, cex = 0.8)
```



A useful function for the model diagnostics (courtesy of Peter Craigmile)

```r
plot.residuals <- function (x, y = NULL, lag.max = NULL, mean.line = TRUE,
                acf.ylim = c(-0.25, 1), mfrow = c(2, 2),
                            lags = NULL, ...){
  if (!is.null(mfrow))
    par(mfrow = mfrow)
  if (is.null(y)){
    y <- x
    x <- seq(length(y))
  } else {
    x <- as.numeric(x)
    y <- as.numeric(y)
```

```
  }

  if (is.null(lag.max)) {
    lag.max <-  floor(10 * log10(length(x)))
  }
  plot(x, y, type = "l", ...)
  if (mean.line) abline(h = 0, lty = 2)
  qqnorm(y, main = "", las = 1); qqline(y)
  if (is.null(lags)) {
    stats::acf(y, main = "", lag.max = lag.max, xlim = c(0, lag.max), ylim = acf.ylim,
        ylab = "sample ACF", las = 1)

    stats::pacf(y, main = "", lag.max = lag.max, xlim = c(0, lag.max), ylim = acf.ylim,
         ylab = "sample PACF", las = 1)
  }
  else {
    stats::acf(y, main = "", lag.max = lag.max, xlim = c(0, lag.max), ylim = acf.ylim,
        ylab = "sample ACF", xaxt = "n", las = 1)
    axis(side = 1, at = lags)

    stats::pacf(y, main = "", lag.max = lag.max, xlim = c(0, lag.max), ylim = acf.ylim,
         ylab = "sample PACF", xaxt = "n", las = 1)
    axis(side = 1, at = lags)
  }
  Box.test(y, lag.max, type = "Ljung-Box")
}
```

Fitting the SARIMA$(1,1,0) \times (1,0,0)$ model

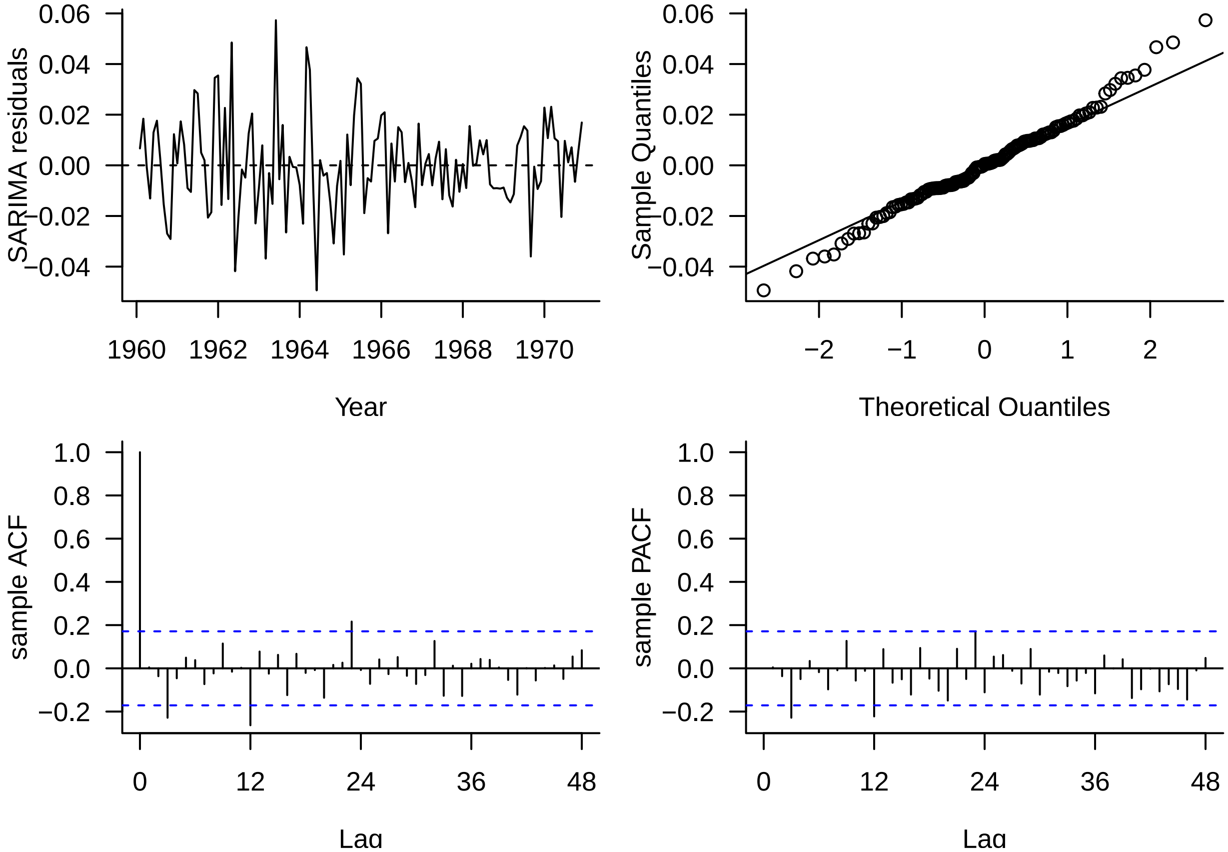
```
(fit1 <- arima(diff.1.0, order = c(1, 0, 0), seasonal = list(order = c(1, 0, 0), period = 12)))
```

```
##
## Call:
## arima(x = diff.1.0, order = c(1, 0, 0), seasonal = list(order = c(1, 0, 0),
##     period = 12))
##
## Coefficients:
##           ar1    sar1  intercept
##       -0.2667  0.9291     0.0039
## s.e.   0.0865  0.0235     0.0096
##
## sigma^2 estimated as 0.0003298:  log likelihood = 327.27,  aic = -648.54
```

```
Box.test(fit1$residuals, lag = 48, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  fit1$residuals
## X-squared = 55.372, df = 48, p-value = 0.2164
```

```r
par(mfrow = c(2, 2), cex = 0.8, bty = "L", mar = c(3.6, 4, 0.8, 0.6),
    mgp = c(2.8, 1, 0), las = 1)
plot.residuals(shortyears[-1], resid(fit1), lag.max = 48,
               ylab = "SARIMA residuals", xlab = "Year", lags = seq(0, 48, 12))
```




```
##
##  Box-Ljung test
##
## data:  y
## X-squared = 55.372, df = 48, p-value = 0.2164
```

Fitting the SARIMA$(0,1,0) \times (1,0,0)$ model

```r
(fit2 <- arima(diff.1.0, seasonal = list(order = c(1, 0, 0), period = 12)))
```

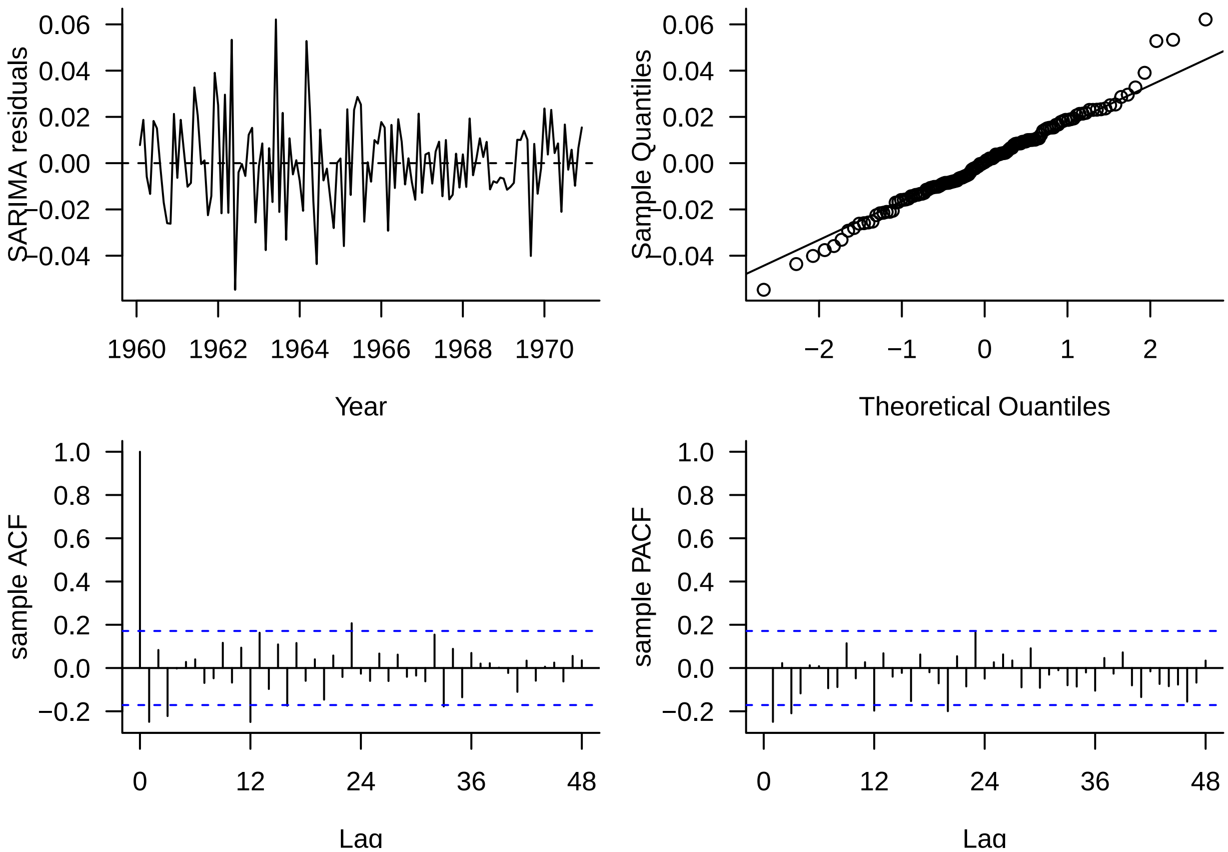
```
##
## Call:
## arima(x = diff.1.0, seasonal = list(order = c(1, 0, 0), period = 12))
##
## Coefficients:
##         sar1  intercept
##       0.9081     0.0040
## s.e.  0.0278     0.0108
```

```
##
## sigma^2 estimated as 0.0003616:  log likelihood = 322.75,  aic = -641.51
```

```
Box.test(fit2$residuals, lag = 48, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  fit2$residuals
## X-squared = 80.641, df = 48, p-value = 0.002209
```

```
par(mfrow = c(2, 2), cex = 0.8, bty = "L", mar = c(3.6, 4, 0.8, 0.6),
    mgp = c(2.8, 1, 0), las = 1)
plot.residuals(shortyears[-1], resid(fit2), lag.max = 48,
               ylab = "SARIMA residuals", xlab = "Year", lags = seq(0, 48, 12))
```




```
##
##  Box-Ljung test
##
## data:  y
## X-squared = 80.641, df = 48, p-value = 0.002209
```

Forecasting 1971 Data

Fit the SARIMA$(1, 1, 0) \times (1, 0, 0)$ Model

```r
(fit1 <- arima(log.shortair, order = c(1, 1, 0),
                     seasonal = list(order = c(1, 0, 0), period = 12)))
```

```
##
## Call:
## arima(x = log.shortair, order = c(1, 1, 0), seasonal = list(order = c(1, 0,
##     0), period = 12))
##
## Coefficients:
##           ar1    sar1
##       -0.2665  0.9298
## s.e.   0.0866  0.0233
##
## sigma^2 estimated as 0.0003299:  log likelihood = 327.19,  aic = -650.38
```

Fit the SARIMA$(0,1,0) \times (1,0,0)$ Model

```r
(fit2 <- arima(log.shortair, order = c(0, 1, 0),
                     seasonal = list(order = c(1, 0, 0), period = 12)))
```

```
##
## Call:
## arima(x = log.shortair, order = c(0, 1, 0), seasonal = list(order = c(1, 0,
##     0), period = 12))
##
## Coefficients:
##         sar1
##       0.9088
## s.e.  0.0276
##
## sigma^2 estimated as 0.0003617:  log likelihood = 322.69,  aic = -643.38
```

```r
AIC.to.AICC <- function(aic, n, npars){ aic - 2 * npars * (1 - n / (n - 1 - npars)) }
```

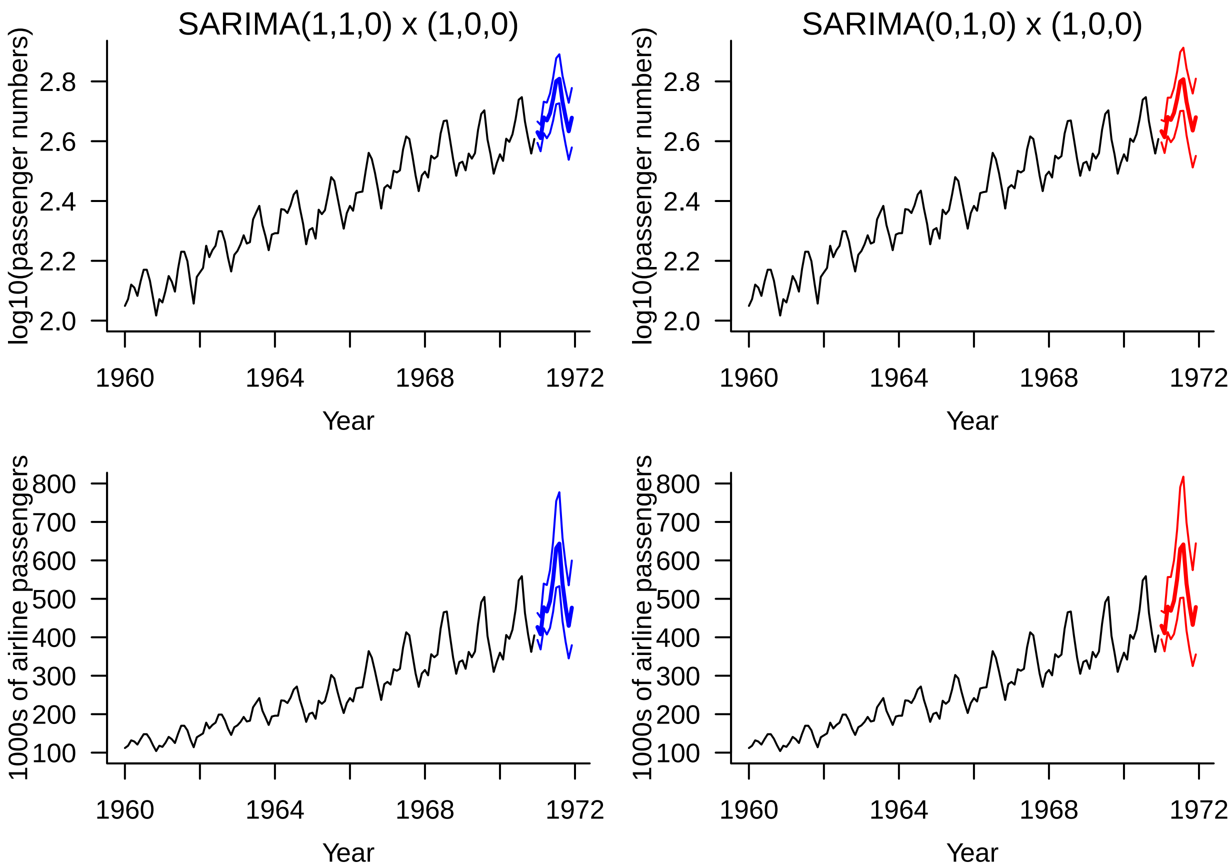
Define the forecasting time points

```r
fyears <- yr[133:144]
```

Calculate the predictions and prediction intervals for both models

```r
preds1 <- predict(fit1, 12)
forecast1 <- preds1$pred
flimits1  <- qnorm(0.975) * preds1$se

preds2 <- predict(fit2, 12)
forecast2 <- preds2$pred
flimits2  <- qnorm(0.975) * preds2$se
```

```r
par(mfrow = c(2, 2), cex = 0.8, bty = "L", mar = c(3.6, 4, 1, 0.6),
    mgp = c(2.4, 1, 0), las = 1)
plot(shortyears, log.shortair, type = "l", xlab = "Year",
     ylab = "log10(passenger numbers)", xlim = range(yr), ylim = c(2, 2.9))
mtext("SARIMA(1,1,0) x (1,0,0)")
## plots the forecasts
lines(fyears, forecast1, lwd = 2, col = "blue")
## plot the 95% prediction intervals.
lines(fyears, forecast1 + flimits1, col = "blue")
lines(fyears, forecast1 - flimits1, col = "blue")

plot(shortyears, log.shortair, type = "l", xlab = "Year",
     ylab = "log10(passenger numbers)", xlim = range(yr), ylim = c(2, 2.9))
mtext("SARIMA(0,1,0) x (1,0,0)")
## plots the forecasts
lines(fyears, forecast2, lwd = 2, col = "red")
## plot the 95% prediction intervals.
lines(fyears, forecast2 + flimits2, col = "red")
lines(fyears, forecast2 - flimits2, col = "red")

plot(shortyears, 10^log.shortair, type = "l", xlab = "Year",
     ylab="1000s of airline passengers", xlim = range(yr), ylim = c(100, 800))
lines(fyears, 10^forecast1, lwd = 2, col = "blue")
lines(fyears, 10^(forecast1 + flimits1), col = "blue")
lines(fyears, 10^(forecast1 - flimits1), col = "blue")

plot(shortyears, 10^log.shortair, type = "l", xlab = "Year",
     ylab="1000s of airline passengers", xlim = range(yr), ylim = c(100, 800))
lines(fyears, 10^forecast2, lwd = 2, col = "red")
lines(fyears, 10^(forecast2 + flimits2), col = "red")
lines(fyears, 10^(forecast2 - flimits2), col = "red")
```



Evaluating Forecast Performance

Root mean square error (RMSE)

```
sqrt(mean((10^forecast1 - 10^log.airpass[133:144])^2))
```

```
## [1] 30.36384
```

```
sqrt(mean((10^forecast2 - 10^log.airpass[133:144])^2))
```

```
## [1] 31.32376
```

Mean relative prediction error

```
mean((10^forecast1 - 10^log.airpass[133:144]) / 10^log.airpass[133:144])
```

```
## [1] 0.05671086
```

```
mean((10^forecast2 - 10^log.airpass[133:144]) / 10^log.airpass[133:144])
```

```
## [1] 0.05951677
```

Empirical coverage rate

```r
CI_fit1 <- cbind(as.numeric(10^(forecast1 + flimits1)),
                 as.numeric(10^(forecast1 - flimits1)))
out <- CI_fit1 - 10^log.airpass[133:144]
hits <- apply(out, 1, function(x) prod(x) < 0)

sum(hits) / length(10^log.airpass[133:144])
```

```
## [1] 0.9166667
```

```r
CI_fit2 <- cbind(as.numeric(10^(forecast2 + flimits2)),
                 as.numeric(10^(forecast2 - flimits2)))
out <- CI_fit2 - 10^log.airpass[133:144]
hits <- apply(out, 1, function(x) prod(x) < 0)

sum(hits) / length(10^log.airpass[133:144])
```

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