

# Package: mylib2 (via r-universe)

October 22, 2026

**Title** Time Series Analysis Practicals  
**Version** 1.0.0  
**Description** Reference material for six sets of time series analysis  
practicals.  
**License** MIT  
**Encoding** UTF-8  
**Roxygen** list(markdown = TRUE)  
**Repository** <https://anuragk007.r-universe.dev>  
**Date/Publication** 2026-10-22 19:47:51 UTC  
**RemoteUrl** <https://github.com/AnuragK007/ts>  
**RemoteRef** HEAD  
**RemoteSha** 976b87ec07ca913d5b5470b2581d015b0e872164  
**RemoteSubdir** mylib2

## Contents

|              |           |
|--------------|-----------|
| set1         | 2         |
| set2         | 3         |
| set3         | 4         |
| set4         | 6         |
| set5         | 7         |
| set6         | 9         |
| <b>Index</b> | <b>11</b> |

**Usage**

```
set1()
```

**Details**

The following listing contains the practical exercises for Set 1.

**Examples**

```
## Not run:
QUESTION 1: Exponential, Moving Average & Holt-Winters Smoothing
```

```
library(forecast)
data(AirPassengers)
x <- AirPassengers
```

```
plot(x, main="Monthly Air Passengers")
ma <- stats::filter(x, rep(1/12, 12), sides=2)
ma <- stats::filter(ma, c(0.5, 0.5), sides=2)
lines(ma, col="red", lwd=2)
```

```
ses <- ses(x, h=12)
holt_model <- holt(x, h=12)
add <- hw(x, seasonal="additive", h=12)
mul <- hw(x, seasonal="multiplicative", h=12)
```

```
parameters <- data.frame(
  Method=c("SES", "Holt", "Additive", "Multiplicative"),
  Alpha=c(ses$model$par["alpha"], holt_model$model$par["alpha"],
    add$model$par["alpha"], mul$model$par["alpha"]),
  Beta=c(NA, holt_model$model$par["beta"],
    add$model$par["beta"], mul$model$par["beta"]),
  Gamma=c(NA, NA, add$model$par["gamma"], mul$model$par["gamma"])
)
print(parameters)
```

```
plot(x, main="Smoothing Methods Comparison", col="black")
lines(fitted(ses), col="red")
lines(fitted(holt_model), col="blue")
lines(fitted(add), col="green")
lines(fitted(mul), col="purple")
legend("topleft",
  legend=c("Original", "SES", "Holt", "Additive", "Multiplicative"),
  col=c("black", "red", "blue", "green", "purple"), lty=1)
```

Interpretation: Interpretation: Alpha controls level smoothing, beta controls trend smoothing, and gamma controls s

## QUESTION 2: AR &amp; MA Models

```

data(AirPassengers)
x <- AirPassengers

acf(x, main="ACF")
pacf(x, main="PACF")

trend <- lm(as.numeric(x) ~ time(x))
d <- ts(residuals(trend), start=start(x), frequency=12)

ar1 <- arima(d, order=c(1,0,0))
ma1 <- arima(d, order=c(0,0,1))

print(ar1)
print(ma1)
print(sqrt(diag(ar1$var.coef)))
print(sqrt(diag(ma1$var.coef)))

phi <- ar1$coef["ar1"]
cat("AR(1) Stationary:", abs(phi) < 1)

```

Interpretation: Interpretation: For AR(1), stationarity requires  $|\phi| < 1$ . The ACF and PACF help identify dependencies.

## End(Not run)

---

set2

---

*ARMA Models and ARIMA Forecasting*


---

**Usage**

```
set2()
```

**Details**

The following listing contains the practical exercises for Set 2.

**Examples**

```

## Not run:
QUESTION 1: ARMA(1,1): Estimation, Roots, Stationarity & Invertibility

data(AirPassengers)
x <- AirPassengers
d <- diff(x)

model <- arima(d, order=c(1,0,1))
print(model)

```

```

phi <- model$coef["ar1"]
theta <- model$coef["ma1"]
mu <- model$coef["intercept"]

cat("ARMA Equation: Yt =", mu, "=", phi,
    "(Yt-1 =", mu, ") + et +", theta, "et-1\n")

ar_root <- polyroot(c(1, -phi))
ma_root <- polyroot(c(1, theta))
print(ar_root)
print(ma_root)

cat("Stationary:", all(Mod(ar_root) > 1), "\n")
cat("Invertible:", all(Mod(ma_root) > 1))

```

Interpretation: Interpretation: AR roots must lie outside the unit circle for stationarity; MA roots must lie outside the unit circle for invertibility.

#### QUESTION 2: ADF Test, Differencing & ARIMA Forecasting

```

library(tseries)
library(forecast)
data(AirPassengers)
x <- AirPassengers

adf.test(x)
d <- ndiffs(x)
print(d)

dx <- diff(x, differences=d)
adf.test(dx)

model <- auto.arima(x, d=d, seasonal=FALSE)
summary(model)

fc <- forecast(model, h=12, level=95)
print(fc)
plot(fc, main="ARIMA 12-Month Forecast")

```

Interpretation: Interpretation: ADF null hypothesis is a unit root. A p-value below 0.05 is evidence against the unit root hypothesis.

```
## End(Not run)
```

#### Usage

```
set3()
```

## Details

The following listing contains the practical exercises for Set 3.

## Examples

```
## Not run:
QUESTION 1: SARIMA & Time Series Regression

data(AirPassengers)
x <- AirPassengers
plot(x, main="Monthly Air Passengers")
monthplot(x)

y <- log(x)
sarima <- arima(y, order=c(0,1,1),
                seasonal=list(order=c(0,1,1), period=12))
print(sarima)

trend <- 1:length(x)
month <- factor(cycle(x))
reg <- lm(as.numeric(y) ~ trend + month)
summary(reg)

actual <- as.numeric(y)
sarima_fit <- actual - as.numeric(residuals(sarima))
reg_fit <- fitted(reg)

r2 <- function(a, f) 1 - sum((a-f)^2)/sum((a-mean(a))^2)
rmse <- function(a, f) sqrt(mean((a-f)^2))

comparison <- data.frame(
  Model=c("SARIMA", "Regression"),
  R2=c(r2(actual, sarima_fit), r2(actual, reg_fit)),
  RMSE=c(rmse(actual, sarima_fit), rmse(actual, reg_fit))
)
print(comparison)
```

Interpretation: Interpretation: Higher  $R^2$  and lower RMSE indicate better in-sample fit. The SARIMA residual-based f

QUESTION 2: MLE, Conditional Sum of Squares & AR/MA Grid Search

```
data(AirPassengers)
x <- diff(log(AirPassengers))

mle <- arima(x, order=c(1,0,1), method="ML")
cls <- arima(x, order=c(1,0,1), method="CSS")
print(mle)
print(cls)

results <- data.frame()
for(p in 0:3){
  for(q in 0:3){
```

```

    model <- try(arima(x, order=c(p,0,q), method="ML"), silent=TRUE)
    if(!inherits(model, "try-error")){
      results <- rbind(results,
        data.frame(p=p, q=q, AIC=AIC(model)))
    }
  }
}
results <- results[order(results$AIC), ]
print(results)
cat("Best model on differenced log data: ARMA(",
  results$p[1], ", ", results$q[1], ")\n", sep="")

```

Interpretation: Interpretation: ML and CSS estimates can differ. The model with the lowest AIC is preferred among the

## End(Not run)

---

set4

*Model Selection and Volatility Diagnostics*

---

## Usage

```
set4()
```

## Details

The following listing contains the practical exercises for Set 4.

## Examples

```

## Not run:
QUESTION 1: AIC, BIC, FPE & Residual Plots

data(AirPassengers)
x <- AirPassengers

m1 <- arima(x, order=c(1,1,1))
m2 <- arima(x, order=c(2,1,1))
m3 <- arima(x, order=c(0,1,2))
models <- list(m1, m2, m3)
n <- length(x)

results <- data.frame(
  Model=c("ARIMA(1,1,1)", "ARIMA(2,1,1)", "ARIMA(0,1,2)"),
  LogLik=sapply(models, logLik),
  AIC=sapply(models, AIC),
  BIC=sapply(models, BIC),
  FPE=sapply(models, function(m)
    m$sigma2*(n+length(m$coef))/(n-length(m$coef)))
)
print(results)

```

```

best <- models[[which.min(results$AIC)]
res <- residuals(best)

par(mfrow=c(1,3))
acf(res, main="Residual ACF")
hist(res, probability=TRUE, main="Histogram")
lines(density(res), col="red")
qqnorm(res, main="Normal Q-Q Plot")
qqline(res, col="red")
par(mfrow=c(1,1))

```

Interpretation: Interpretation: Prefer lower AIC/BIC/FPE and higher log-likelihood. Residual ACF checks serial dep

QUESTION 2: Ljung-Box, ARCH LM & GARCH(1,1)

```

library(FinTS)
library(rugarch)
data(AirPassengers)
x <- AirPassengers

m1 <- arima(x, order=c(1,1,1))
m2 <- arima(x, order=c(2,1,1))
m3 <- arima(x, order=c(0,1,2))
models <- list(m1, m2, m3)
best <- models[[which.min(sapply(models, AIC))]]
res <- na.omit(as.numeric(residuals(best)))

Box.test(res, lag=12, type="Ljung-Box")
Box.test(res^2, lag=12, type="Ljung-Box")
arch <- ArchTest(res, lags=12)
print(arch)

if(arch$p.value < 0.05){
  spec <- ugarchspec(
    variance.model=list(model="sGARCH", garchOrder=c(1,1)),
    mean.model=list(armaOrder=c(0,0), include.mean=FALSE)
  )
  fit <- ugarchfit(spec, data=res)
  show(fit)
  plot(as.numeric(sigma(fit))^2, type="l",
    main="GARCH(1,1) Conditional Variance",
    xlab="Time", ylab="Variance")
}

```

Interpretation: Interpretation: For Ljung-Box and ARCH LM tests,  $p < 0.05$  indicates evidence against the null of no

## End(Not run)

**Usage**

```
set5()
```

**Details**

The following listing contains the practical exercises for Set 5.

**Examples**

```
## Not run:
QUESTION 1: Holt-Winters Forecasting & MAPE

data(AirPassengers)
x <- AirPassengers
train <- window(x, end=c(1958,12))
test <- window(x, start=c(1959,1))

model <- HoltWinters(train, seasonal="multiplicative")
fc <- predict(model, n.ahead=24)

plot(x, main="Holt-Winters Forecast")
lines(fc, col="red", lwd=2)
mape <- mean(abs((as.numeric(test)-as.numeric(fc))/as.numeric(test)))*100
print(mape)

m1 <- HoltWinters(train, alpha=0.2, seasonal="multiplicative")
m2 <- HoltWinters(train, alpha=0.8, seasonal="multiplicative")
plot(train, main="Effect of Alpha")
lines(fitted(m1)[,"xhat"], col="blue")
lines(fitted(m2)[,"xhat"], col="red")
legend("topleft", legend=c("Original", "Alpha=0.2", "Alpha=0.8"),
      col=c("black","blue","red"), lty=1)
```

Interpretation: Interpretation: Lower MAPE indicates better forecast accuracy. A higher alpha generally responds more

QUESTION 2: SARIMA Forecast & Holt-Winters Comparison

```
library(forecast)
data(AirPassengers)
x <- AirPassengers
train <- window(x, end=c(1958,12))
test <- window(x, start=c(1959,1))

sarima <- Arima(train, order=c(1,1,1),
                seasonal=list(order=c(1,1,0), period=12))
print(sarima)
sf <- forecast(sarima, h=24)
plot(sf, main="SARIMA Forecast")
lines(test, col="red")

hw <- HoltWinters(train, seasonal="multiplicative")
hf <- predict(hw, n.ahead=24)
```

```
rmse <- function(a,f) sqrt(mean((as.numeric(a)-as.numeric(f))^2))
mae <- function(a,f) mean(abs(as.numeric(a)-as.numeric(f)))
comparison <- data.frame(
  Model=c("SARIMA", "Holt-Winters"),
  RMSE=c(rmse(test, sf$mean), rmse(test, hf)),
  MAE=c(mae(test, sf$mean), mae(test, hf))
)
print(comparison)
```

Interpretation: Interpretation: The model with lower RMSE/MAE has better test-set accuracy. RMSE penalizes large errors.

```
## End(Not run)
```

---

## set6

## *Parameter Significance and Residual Diagnostics*

---

### Usage

```
set6()
```

### Details

The following listing contains the practical exercises for Set 6.

### Examples

```
## Not run:
QUESTION 1: ARIMA Parameter Significance & Simplification

data(AirPassengers)
x <- log(AirPassengers)
model <- arima(x, order=c(2,1,2))
print(model)

se <- sqrt(diag(model$var.coef))
t_value <- model$coef/se
p_value <- 2*pnorm(-abs(t_value))
print(data.frame(Coefficient=model$coef, SE=se,
  t_value=t_value, p_value=p_value))

candidates <- list(
  c(2,1,2), c(1,1,2), c(2,1,1), c(1,1,1),
  c(2,1,0), c(0,1,2), c(1,1,0), c(0,1,1)
)
fits <- lapply(candidates, function(o)
  try(arima(x, order=o), silent=TRUE))
ok <- !sapply(fits, inherits, "try-error")
fits <- fits[ok]
orders <- candidates[ok]
```

```

aics <- sapply(fits, AIC)
print(data.frame(
  Model=sapply(orders, function(o) paste0("ARIMA(", paste(o, collapse=","), ")")),
  AIC=aics
))
print(fits[[which.min(aics)]])

```

Interpretation: Interpretation: Approximate  $p < 0.05$  indicates a statistically significant coefficient. Compare si

QUESTION 2: AIC/BIC Grid Search, Durbin-Watson & Cumulative Periodogram

```

library(lmtest)
data(AirPassengers)
x <- log(AirPassengers)
results <- data.frame()

for(p in 0:3){
  for(q in 0:3){
    m <- try(arima(x, order=c(p,1,q)), silent=TRUE)
    if(!inherits(m, "try-error")){
      results <- rbind(results, data.frame(p=p, q=q,
                                           AIC=AIC(m), BIC=BIC(m)))
    }
  }
}
print(results)
best <- results[which.min(results$BIC), ]
print(best)

model <- arima(x, order=c(best$p,1,best$q))
res <- na.omit(as.numeric(residuals(model)))
dwtest(res ~ 1)
cpgram(res, main="Cumulative Periodogram")

```

Interpretation: Interpretation: Lower AIC/BIC is preferred. BIC penalizes complexity more strongly for this sample

## End(Not run)

# Index

## \* documentation

set1, [2](#)

set2, [3](#)

set3, [4](#)

set4, [6](#)

set5, [7](#)

set6, [9](#)

set1, [2](#)

set2, [3](#)

set3, [4](#)

set4, [6](#)

set5, [7](#)

set6, [9](#)