**14.13 R语言应用**

**# 陈文贤 着 《大话统计学》 清华大学出版社 2022年**

if(!require(aTSA)){install.packages("aTSA")} ; library(aTSA)

if(!require(fma)){install.packages("fma")} ; library(fma)

if(!require(astsa)){install.packages("astsa")} ; library(astsa)

if(!require(tseries)){install.packages("tseries")} ; library(tseries)

if(!require(ggplot2)){install.packages("ggplot2")} ; library(ggplot2)

if(!require(TSA)){install.packages("TSA")} ; library(TSA)

if(!require(forecast)){install.packages("forecast")} ; library(forecast)

# R例14.2

x <- c(800,1400,1000,1500,1500,1300,1800,1700,1300,1700,1700,1500,2300,2300,2000)

y <- ts(x) # ts 时间序列格式 tsdata <- ts(data, start=, end=, frequency=)

ma <- MA(y, nlag =1, ) # aTSA::MA 移动平均 n = 2\*nlag+1 # 只能作奇数期移动平均

ma$estimate

dma <- MA(ma$estimate, nlag =1) ; dma # 双重移动平均

# R例14.3

x <- c(200,135,195,197.5,310,175,155,130,220,277.5,235)

y <- ts(x) # ts 时间序列格式

es <- expsmooth(y, alpha=0.5) # trend = 1: 简单指数平滑

plot(y,type = "b") ; lines(es$estimate,col = 2)

expsmooth(y, trend = 2, alpha=0.5) # trend = 2: 双重指数平滑

expsmooth(y,trend = 3, alpha=0.5) # trend = 3: 三重指数平滑

# R例14.6

x <- c(34,40,35,39.41,36,33,38,43,40) ; y <- ts(x) # ts 时间序列格式

es <- expsmooth(y, trend = 1, alpha=0.2, beta=0.3) ; es # trend = 1: 简单指数平滑

es <- expsmooth(y, trend = 2, alpha=0.2, beta=0.3) ; es # trend = 2: 双重指数平滑

es <- expsmooth(y, trend = 3, alpha=0.2, beta=0.3) ; es # trend = 3: 三重指数平滑

# R例14.6

x <- c(34,40,35,39.41,36,33,38,43,40) ; y <- ts(x) # ts 时间序列格式

ses(y, h=1, alpha=0.2, beta=0.3, lambda="auto")

fc <- ses(y, h=1, alpha=0.2) ; round(accuracy(fc),2)

autoplot(fc) + autolayer(fitted(fc), series="Fitted")

# R例14.5

x = read.csv("C:/大话统计学 网络资源/StatData/Chap14\_5.csv",header=TRUE)

# 读入 Chap14\_5.csv

fit <- lm(Y~t, data=x) ; summary(fit)

# R例14.10

x = read.csv("C:/大话统计学 网络资源/StatData/Chap14\_10.csv",header=TRUE)

# 读入 Chap14\_10.csv

fit <- lm(Y~t+Q1+Q2+Q3, data=x) ; summary(fit) ; fit

y <- data.frame(t=20, Q1=1, Q2=0, Q3=0)

predict(fit, y) ; x <- ts(x)

ndiffs(x, alpha = 0.05, test = c("kpss", "adf", "pp"), type = c("level", "trend"), max.d=2)

# 单位根检验

x <- rnorm(1000) ; pp.test(x) # 没有单位根

y <- cumsum(x) ; pp.test(y) # 有单位根

tseries::kpss.test(y, null = "Trend")

ndiffs(x, alpha = 0.05, test = c("kpss", "adf", "pp"), type = c("level", "trend"), max.d=2)

ndiffs(x, alpha = 0.05, test = c("kpss", "adf", "pp"),type = c("level", "trend") )

tseries :: adf.test(x)

# 尼罗河流量数据1871-1970

par(mfrow = c(1, 1)) ; data(Nile) ; str(Nile) # 尼罗河流量1871-1970

plot(Nile, col="blue") ; abline(reg = lm(Nile ~ time(Nile)), col="red")

ylim <- c(min(Nile), max(Nile)) ; plot(ma(Nile, 3), main="SMA (k=3)", ylim=ylim)

ndiffs(Nile) # 计算差分的数目

Nile\_diff <- diff(Nile, lag = 1) ; par(mfrow = c(2,2)) ; plot(Nile) ; acf(Nile) ; pacf(Nile)

ar(Nile) # AR(2)模型

cpgram(ar(Nile)$resid)

par(mfrow = c(1,1)) ; arima(Nile, c(2, 0, 0))

Box.test(Nile\_diff, type = "Ljung-Box")

par(mfrow = c(2, 2))

plot(Nile, main= '差分前', col="blue") ; plot(Nile\_diff, main='差分后', col="blue")

abline(reg = lm(Nile\_diff ~ time(Nile\_diff)))

acf(Nile\_diff, main = '差分后acf' , col="red")

pacf(Nile\_diff, main = '差分后pacf', col="red")

par(mfrow = c(1, 1))

Nile.arima <- arima(Nile\_diff, order = c(0,1,1),method = "ML")

Nile.pre <- forecast(Nile.arima, h = 10) # 预测

plot(Nile.pre) ; Nile.arima

qqnorm(Nile.arima$residuals)

qqline(Nile.arima$residuals) # 残差正态性检验法

shapiro.test(Nile.arima$residuals) # 残差相关性

Box.test(Nile.arima$residuals, type = "Ljung-Box")

par(mfrow = c(1, 2)) ; plot.ts(Nile.arima$residuals)

abline(reg = lm(Nile.arima$residuals ~ time(Nile.arima$residuals)))

acf(Nile.arima$residuals)

# 航空旅客数据 AirPassengers

lx <- log(AirPassengers)

par(mai = c(0.9, 0.9, 0.1, 0.1), omi = c(0, 0, 0, 0))

plot.ts(lx, las = 1, ylab = "")

ma(lx, 5) # 移动平均

ggAcf(lx) + ggtitle('')

air\_decompose <- decompose(AirPassengers, type = "multiplicative")

plot(air\_decompose) # 时间数据分解成分

seas\_2 <- decompose(lx)$seasonal # 季节成分

par(mai = c(0.9, 0.9, 0.1, 0.1), omi = c(0, 0, 0, 0))

plot.ts(seas\_2, las = 1, ylab = "")

ee <- decompose(lx)$random

decomp <- decompose(lx) ; decomp ; plot(decomp)

# Holt-winters 指数平滑模型

airpass = AirPassengers # 航空旅客数据 airpass = AirPassengers

autoplot(airpass)

ses5 <- ses(airpass, h=5)

accuracy(ses5)

autoplot(ses5)+ autolayer(fitted(ses5), series = "Fitted")

holt5 <- holt(airpass, h=5)

autoplot(holt5)+ autolayer(fitted(holt5), series = "Fitted")

holt5damped <- holt(airpass, damped = TRUE, phi = 0.9, h = 15)

autoplot(holt5damped) + autolayer(fitted(holt5damped), series = "Fitted")

hw1 <- hw(airpass, seasonal = "additive")

hw2 <- hw(airpass, seasonal = "multiplicative")

auto.arima(airpass)

ndiffs(airpass, alpha=0.05,test=c("kpss", "adf", "pp"), type = c("level", "trend"), max.d=2)

air\_decompose <- decompose(AirPassengers, type = "multiplicative")

plot(air\_decompose)

airpass.hw <- HoltWinters(airpass)

plot(airpass.hw, col = "blue", col.predicted = "red")

airpass.hw$SSE ; airpass.hw$alpha ; airpass.hw$beta ; airpass.hw$gamma

airpass.forecast <- forecast(airpass.hw, h=20)

plot(airpass.forecast)

airpass.arima <- auto.arima(airpass) # 自动化 ARIMA

summary(airpass.arima)

airpass.forecast <- forecast(airpass.arima, h=10) # 预测10期后

plot(airpass.forecast) # 预测图

sarima(log(AirPassengers),p=0,d=1,q=1,P=0,D=1,Q=1,S=12)

(model <- sarima(log(AirPassengers),0,1,1,0,1,1,12))

summary(model$fit) # 模型综述

plot(resid(model$fit)) # 残差图

sarima(log(AirPassengers),0,1,1,0,1,1,12,details=TRUE)$BIC # sarima模型的BIC

model1 <- ets(airpass, model="AAA") ; model2 <- ets(airpass, model="MMM")

autoplot(model1) ; summary(model1)

summary(model2) ; checkresiduals(model2)

autoplot(forecast(model2))

airpass = AirPassengers

for (p in c(0,1)) {

for (q in c(0,1)) {

for (P in c(0,1)) {

for (Q in c(0,1)) {

mod = arima(airpass, order=c(p,1,q), seasonal=list(order=c(P,1,Q) , period=12))

aic = mod$aic

cat("p =", p, "q =", q , "P =", P , "Q =", Q, "AIC =", aic, "\n" )

}}}}