# Assignment10

Haoran Xu

2024-11-26

```
library(forecast)
```

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

```
library(changepoint)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

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

```
## Successfully loaded changepoint package version 2.3
##  WARNING: From v.2.3 the default method in cpt.* functions has changed from AMOC to PELT.
##  See NEWS for details of all changes.
```

```
l1 <-rnorm(1000)
l2 <- rep(0,length(l1))
l3 <- rep(0,length(l1))
l4 <- rep(0,length(l1))

for(i in 5:length(l1)){
  l2[i] <- (l1[i-4]+rnorm(1))/2
  l3[i] <- (l1[i-1]+l1[i-2]+l1[i])/3
  l4[i] <- l3[i]+0.001*i
}

plot(l4,pch=20,col="red",xlab="time")
```

```
#t-1
outlag1 <- lm(l4[2:1000] ~ l4[1:999])
summary(outlag1)
plot(l4[2:1000] ~ l4[1:999])
```

**Q1. Use this same approach to write code to create a predictive model that models t (l4) as a function of t-2, and then a second model that predicts t as a function of t-3. Call the models outlag2 and outlag3 respectively. Describe the differences in R-squared values between these models.**

```
#t-2
outlag2 <- lm(l4[3:1000] ~ l4[1:998])
summary(outlag2)
plot(l4[3:1000] ~ l4[1:998])
```

```
#t-3
outlag3 <- lm(l4[4:1000] ~ l4[1:997])
summary(outlag3)
plot(l4[4:1000] ~ l4[1:997])
```

R-squared values in the three models are decreasing along with increasing of lags between t and (t-x). This means the predictive power in models diminished each time, also suggesting lowering autocorrelation if lengthening the laps. This is reasonable as `l4` is calculated based on `l3`, which is based on the moving average of the past three (including itself) values, which are `l1[i]`, `l1[i-1]`, and `l1[i-2]`.

---

```
# Test for autocorrelation in data
Box.test(l1, lag=1, type = "Ljung-Box")

# Test for autocorrelation in the model1 residuals
Box.test(residuals(outlag1),
         lag=1, type = "Ljung-Box")

# plotting the residuals
plot(outlag1$fitted.values, outlag1$residuals,
     main = "Residuals vs Fitted Values",
     xlab = "Fitted Values",
     ylab = "Residuals",
     pch = 20, col = "blue")
abline(h = 0, col = "red", lwd = 2)
```

```
# plotting time series plot
plot(outlag1$residuals, type = "l",
     main = "Residuals Time Series",
     xlab = "Time",
     ylab = "Residuals",
     col = "purple")
```

**Q2. Write code that applies the Box-Ljung test to the residuals from outlag2 and outlag3 and briefly interpret the p-value.**

```
# Test for autocorrelation in the model2 residuals
Box.test(residuals(outlag2),
```

```
         lag=1, type = "Ljung-Box")

# Test for autocorrelation in the model3 residuals
Box.test(residuals(outlag3),
         lag=1, type = "Ljung-Box")
```

For the `outlag2` and `outlag3` models, the Box-Ljung tests both produced a p-value of less than 2.2e-16, meaning that the null hypotheses of having no autocorrelation should be rejected. These low p-values imply that both models have residuals that are not independent.

```
plot(seq(1:length(residuals(outlag1))), residuals(outlag1))
```

---

```
# adding a "time" for all the 1000 variables in l4
d_ts2 <- ts(l4, start=c(1900, 1), frequency = 12)
```

### Q3. Using l4, create a data object called d_ts that starts on January 1st 1999 and has a daily frequency.

```
d_ts <- ts(l4, start=c(1999, 1), frequency = 365)
```

---

```
plot(d_ts2)
```

```
plot(d_ts)
```

```
d_ts_diff <- diff(d_ts)
plot(d_ts_diff)
```

### Q4. Use the Box-Ljung test to determine if there is any remaining autocorrelation in the data. Interpret the result.

```
# Test the differencing results
Box.test(d_ts_diff,
         lag=1, type = "Ljung-Box")
```

The results showed a p-value of $0.66 > 0.05$, indicating that there is no autocorrelation in the data at lag 1 after differencing.

---

```r
set.seed(2112)
t <- seq(1:500)
x1 <- rnorm(500)
y <- 10 + t + 3.5 * x1 + arima.sim(list(order=c(1,0,0), ar=0.8),500)+rnorm(500)*10

out <- auto.arima(y, xreg = x1, seasonal = FALSE)
summary(out)

out_manual <- arima(y, order= c(1,1,0), xreg = x1)

auto <- fitted(out)
manual <- fitted(out_manual)
plot(auto, manual)
```

```r
checkresiduals(out$residuals)
```

```r
forecast_out <- forecast(out, xreg = x1[401:500])

testout_mean <- forecast_out$mean
testout_low <- forecast_out$lower
testout_up <- forecast_out$upper
observed <- y[401:500] # observed testing data

plot(1:100, observed, pch=20, col="red", ylim=c(300, 600))
points(1:100, testout_mean, pch=21, col="black")
points(1:100, testout_low[,2], pch=20, col="black", cex=.25)
points(1:100, testout_up[,2], pch=20, col="black", cex=.25)
```

**Q5. Calculate the root mean squared error (RMSE) between the observed data and the testout_mean data.**

```r
rmse <- sqrt(mean((observed - testout_mean)^2))
```

---

```r
dates <- seq(as.Date("2010/1/1"), as.Date("2010/12/31"), by = "day")
t <- seq(1:365)

l1a <- rnorm(180)
l1b <- rnorm(185) + 4
l1 <- c(l1a, l1b)

l2 <- sin(t / 10)
l3 <- l1 + l2
plot(l3)
```

**Q6. Based on the code above, how many important change points might you expect in l3 and why?**

I would expect one important change point and it might be around `t = 180`. Because starting from `t = 181`, `l1b` was added an extra `4` than `l1a`, this makes the sequencing numbers all higher than `l1a`. I do not know if or how will the sine wave of `l2` would impact the `l3`. It currently only adds to a periodic variations to `l3`, but without creating new "change points".

---

```r
cpt <- cpt.mean(l3, method = "PELT")
summary(cpt)

test_statistics <- cpt@param.est$mean
index <- which.max(test_statistics)
test_statistic <- test_statistics[index]
test_statistic

perm <- 0
for (i in 1:1000) {
  p_l3 <- sample(l3)
  p_cpt <- cpt.mean(p_l3, method = "PELT")
  if (!is.null(p_cpt@param.est$mean)) {
    temp <- p_cpt@param.est$mean
    p_index <- which.max(temp)
    perm[i] <- temp[p_index]
  } else {
    perm[i] <- mean(p_l3)
  }
}

p_value <- mean(perm >= test_statistic)
p_value
```

---

**Q8. Consider this time series. Model this time series for the purpose of univariate time-series forecasting. The following is the test data set for evaluation of your forecast model. Use whatever method you want on the training data set and only evaluate the quality of your forecasts against the test data set after you are finished training. Ensure that you've calculated the RMSE against the training data and the test data.**

```r
ts <- c(-2.6,-2.6,0.1,0.5,1.4,4.4,3.5,4.1,0.9,2.8,0.5,-4.2,-3.8,-1.8,-3.0,-0.5,-1.5,-0.3,1.2,3.1,3.2,5.

ts_data <- ts(ts, frequency = 12)
ts_data <- diff(ts_data, differences = 1)

fit <- auto.arima(ts_data, seasonal = TRUE, stepwise = FALSE, approximation = FALSE)
forecast_result <- forecast(fit, h = 6)
```

```r
predicted_values <- forecast_result$mean
print(predicted_values)

# test data
test <- c(0.4, 2.4, -1.0, -2.3, 1.9, 2.2)

rmse <- sqrt(mean((predicted_values - test)^2))
r_squared <- 1 - (sum((test - predicted_values)^2) / sum((test - mean(test))^2))
aic_value <- AIC(fit)
bic_value <- BIC(fit)
print(rmse)
print(r_squared)
print(aic_value)
print(bic_value)
```

I used an ARIMA model with automatic parameter selection to forecast future values of a seasonal time series. The model's goodness-of-fit indexes suggest the forecasting model could be improved. The predicted values are close to zero, meaning that it deviates from the real data. `RMSE` of 1.86 suggests the model has a moderate prediction error, and the $R^2$ value (-0.12) suggests that the model is not performing good. Additionally, `AIC` and `BIC` are high, meaning the model might be over-fitted or under-fitted. The results suggest there could be better models or parameter refinement.