PS3

DanlingMa

2/9/2019

# data preparation  
library(forecast)

## Warning: package 'forecast' was built under R version 3.5.2

library(ggplot2)  
library(tidyverse)

## ── Attaching packages ────────────────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ tibble 1.4.2 ✔ purrr 0.2.5  
## ✔ tidyr 0.8.1 ✔ dplyr 0.7.6  
## ✔ readr 1.1.1 ✔ stringr 1.3.1  
## ✔ tibble 1.4.2 ✔ forcats 0.3.0

## ── Conflicts ───────────────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

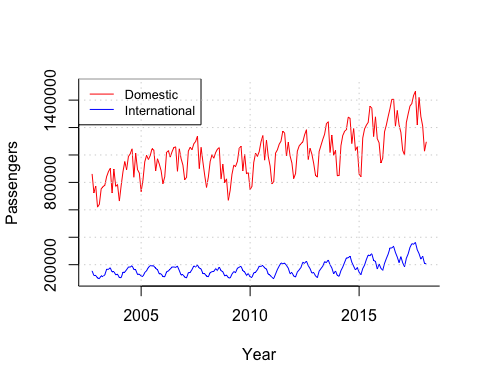
setwd("~/Desktop/Forecasting/ps3-danlingma")  
logan = read.csv('LoganPassengers.csv')  
logan <- logan[logan$total<5000000,]

# 1 For this problem load the Logan airport data set from the lecture csv file.

## 1. **Create time series for the domestic and international passengers.**

## 2. **Plot both of these lines in the same plot.**

logan.ts.tot = ts(logan$total, start = c(2002,10), frequency = 12)  
logan.ts.dom = ts(logan$domestic, start = c(2002,10), frequency = 12)  
logan.ts.int = ts(logan$international, start = c(2002,10), frequency = 12)  
plot(logan.ts.dom,xlab="Year",ylab="Passengers",bty="l",ylim=range(100000,1500000),col="red")  
lines(logan.ts.int,col="blue")  
legend("topleft",legend=c("Domestic", "International"),col=c("red","blue"),lty=1:1, cex=0.8)  
grid()



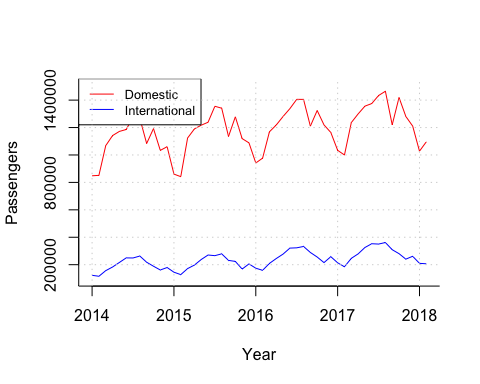
## 3. **Describe any similarities and differences between the two.**

They both have seasonality: During the summer period, passengers traffic increase dramatically. While for the winter season, the traffic decrease. And generally, with the time pass, the traffic increase a bit.

However, there is a huge gap between domestic and international market. Domestic market has a higher traffic than international fights. And the seasonality inflation of domestic is also drastic than international flight.

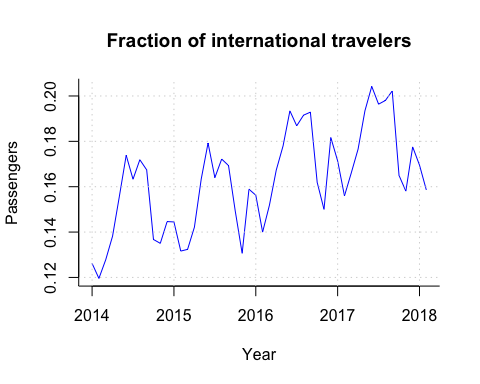
## 4. **Repeat this plot for a short window from 2014 to the end of the data set.**

# create subwindow (3 years)  
logan.ts.zoom.dom <- window(logan.ts.dom,start=c(2014,01))  
logan.ts.zoom.inter <- window(logan.ts.int,start=c(2014,01))  
  
plot(logan.ts.zoom.dom,xlab="Year",ylab="Passengers",bty="l",ylim=range(100000,1500000),col="red")  
lines(logan.ts.zoom.inter,col="blue")  
legend("topleft",legend=c("Domestic", "International"),col=c("red","blue"),lty=1:1, cex=0.8)  
grid()



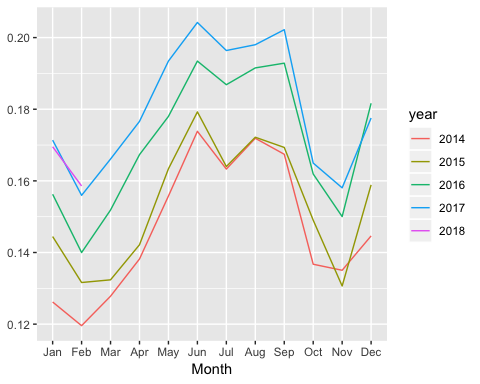
## 5. **Now generate a time series over the full sample of the fraction of international travelers (international/total).**

logan.frac = logan %>% mutate(`fraction` = international/total)  
logan.ts.frac = ts(logan.frac$fraction, start = c(2002,10), frequency = 12)  
logan.ts.frac.zoom = window(logan.ts.frac, start=c(2014,01))  
plot(logan.ts.frac.zoom,xlab="Year",ylab="Passengers",bty="l",col="blue",main="Fraction of international travelers")  
grid()



## 6. **Plot this series in a short window from 2014 to the end of the data set. Does this series appear to have any seasonality. Try a plot to look for seasonals with ggseasonplot().**

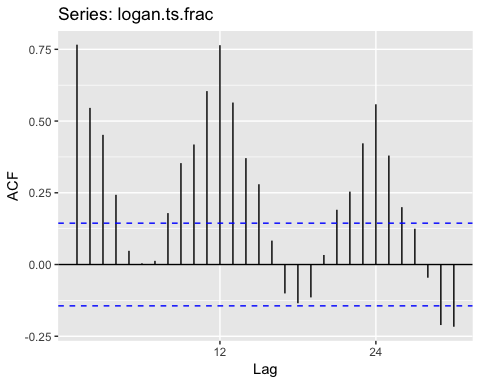
ggseasonplot(logan.ts.frac.zoom, main=NULL)



#ggseasonplot(logan.ts.frac.zoom, polar=T, main=NULL)

## 7. Report the autocorrelations for the series with 30 lags. Do you see any interesting patterns? Do this for the full sample only.

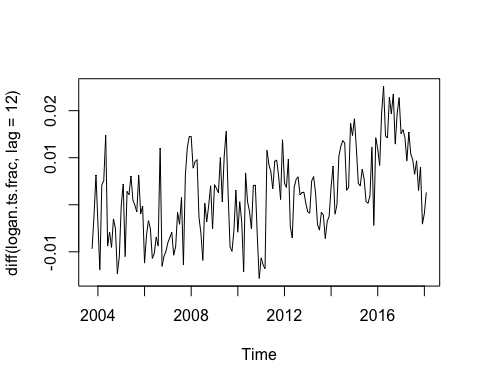
ggAcf(logan.ts.frac, lag.max = 30)



* There is a seasonality in this dataset. The traffic growth rate first increase then decrease. And with the time pass, the maximum of the growth rate decrease.

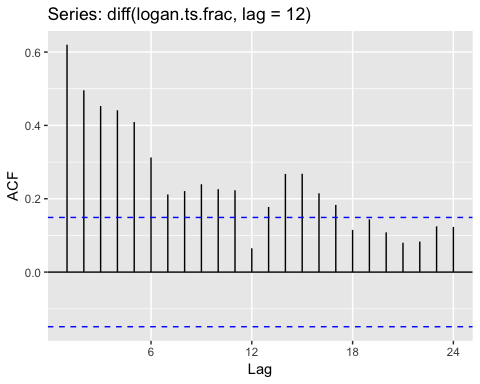
## 8. Now take a twelve lag difference of your time series. Do this with the diff function and use the argument lag=12. **This now converts your series into month to same month changes.** Plot this new series.

plot(diff(logan.ts.frac, lag=12))



## 9. Plot the autocorrelations of this series. Are they similar or different to your autocorrelations on the raw series?

### is this a difference between this year and last year?  
ggAcf(diff(logan.ts.frac,lag = 12))



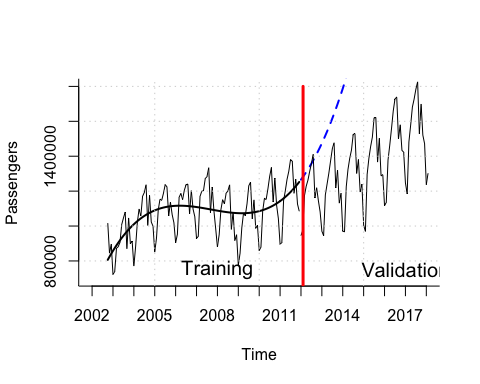
* Both series have seasonality. But for the difference series, the trend always increase ,rather than decrease, eventhough the increase amount turn to be small with the time pass.

# 2 For this question use the total number of passengers.

Divide the data into testing and validation periods. Start the validation data in January, 2012. All earlier data is training.

## 1. **Estimate a cubic trend on the training data. Plot the fitted model, and your forecast as we have done in class.**

train.ts <- window(logan.ts.tot, end = c(2011, 12))  
valid.ts <- window(logan.ts.tot, start = c(2012, 1))  
  
logan.lm <- tslm(train.ts ~ trend + I(trend^2) + I(trend^3))  
logan.lm.pred <- forecast(logan.lm, h = length(valid.ts), level = 0)  
  
  
plot(logan.lm.pred, ylab = "Passengers", bty='l',xlab = "Time",xaxt="n",   
 ylim=c(700000,1800000), xlim = c(2002,2018), main = "", flty = 2)  
axis(1, at = seq(2002, 2018, 1))   
lines(logan.lm$fitted, lwd = 2)  
lines(valid.ts)  
grid()  
lines(c(2012.1,2012.1), c(0, 1800000),lwd=3,col="red")   
text(2008, 750000, "Training",cex=1.25)  
text(2017, 750000, "Validation",cex=1.25)



## 2. **Report the MSE, and MAE for this model in both the training and validation periods.**

valid.err <- valid.ts - logan.lm.pred$mean  
train.err <- train.ts - logan.lm$fitted.values  
  
mseTrain <- mean( train.err^2 )  
mseValid <- mean( valid.err^2 )  
maeTrain = mean(abs(train.err))  
maeValid = mean(abs(valid.err))  
tbl = as.table(c(mseTrain, mseValid, maeTrain, maeValid))  
names(tbl) = c("MSE.train","MSE.valid","MAE.train","MAE.valid")  
knitr::kable(tbl, caption = "Prediction Error")

Prediction Error

|  |  |
| --- | --- |
| Var1 | Freq |
| MSE.train | 1.525667e+10 |
| MSE.valid | 2.017267e+12 |
| MAE.train | 1.023181e+05 |
| MAE.valid | 1.129895e+06 |

## 3. **Now fit a naive (random walk) model and report its MSE and MAE in the validation period.**

logan.lm.naive <- naive(train.ts, h = length(valid.ts))  
  
valid.err.naive <- valid.ts - logan.lm.naive$mean  
  
maeValid = mean(abs(valid.err.naive))  
mseValid <- mean(valid.err.naive^2)  
tbl = as.table(c(mseValid, maeValid))  
names(tbl) = c("MSE.valid","MAE.valid")  
knitr::kable(tbl, caption = "Prediction Error")

Prediction Error

|  |  |
| --- | --- |
| Var1 | Freq |
| MSE.valid | 1.175954e+11 |
| MAE.valid | 2.897971e+05 |

## 4. **Repeat this for the seasonal naive model, snaive().**

logan.lm.snaive <- snaive(train.ts, h = length(valid.ts))  
  
valid.err.snaive <- valid.ts - logan.lm.snaive$mean  
  
maeValid = mean(abs(valid.err.snaive))  
mseValid <- mean( valid.err.snaive^2 )  
tbl = as.table(c(mseValid, maeValid))  
names(tbl) = c("MSE.valid","MAE.valid")  
knitr::kable(tbl, caption = "Prediction Error")

Prediction Error

|  |  |
| --- | --- |
| Var1 | Freq |
| MSE.valid | 51109588390 |
| MAE.valid | 180685 |

## 5. **Set up a time series cross validation experiment with the cubic model. Generate cross validated forecasts in the validation period as we did in class. Report the MSE in the validation period.**

cubtrend <- function(x, h){  
 fmod <- tslm(x ~ trend + I(trend^2)+I(trend^3))  
 forecast(fmod, h=h)  
}  
  
eCV <- tsCV(logan.ts.tot,cubtrend, h = 1)  
eCVValid <- eCV[-(1:length(train.ts)) ]  
mseCVValid <- mean( eCVValid^2,na.rm=TRUE)  
  
sprintf("Cubic Validation, %f",mseCVValid)

## [1] "Cubic Validation, 37192399091.588043"

## 6. **Repeat this cross validation using a naive random walk forecast, but letting the forecast move along with the cross validation, and forecasting one step ahead each time.**

naivetrend <- function(x, h){  
 fmod <- naive(x, h = h)  
}  
  
eCV.naive <- tsCV(logan.ts.tot,naivetrend, h = 1)  
eCVValid.naive <- eCV.naive[-(1:length(train.ts))]  
mseCVValid.naive <- mean(eCVValid.naive^2,na.rm=TRUE)  
  
sprintf("Cubic Validation, %f",mseCVValid.naive)

## [1] "Cubic Validation, 19825099373.739727"

## 7. **Finally, replace the naive forecast with the seasonal naive forecast, snaive(). Again, report the MSE in the validation period.**

snaivetrend <- function(x, h){  
 fmod <- snaive(x, h = h)  
}  
  
eCV.snaive <- tsCV(logan.ts.tot,snaivetrend, h = 1)  
eCVValid.snaive <- eCV.snaive[-(1:length(train.ts)) ]  
mseCVValid.snaive <- mean( eCVValid.snaive^2,na.rm=TRUE)  
  
sprintf("Cubic Validation, %f",mseCVValid.snaive)

## [1] "Cubic Validation, 6263688849.178082"

## 8. **Discuss the relative magnitudes of your various forecasting methods (from parts (2), (3), (4), (5), (6), (7)).**

From part (2), MAE is smaller than MSE. This may because there are some large errors in the data set. Since the errors are squared before they are averaged, the RMSE gives a relatively high weight to large errors. This means the RMSE should be more useful when large errors are particularly undesirable.

For part (3) to (4), we can see that the seasonal naive one is much better than naive model.

And when we conduct cross-validation, the MSE decrease for either the cubic model or naive/snaive models. Therefore, we found that the models actually are not as bad as the result from non-cross-validations. Via cross-validation, we can be more confident to verify whether our model is accurate or not.

# 3 In this problem you are to compare the MSE for several different data pairs.

In each case answer which is smallest, or not enough information to tell. Assume that the function MSE(X,Y) represents the mean squared error for a model estimated on X, and MSE estimated on Y. Estimation uses some form of estimator that minimizes squared errors, like ordinary least squares. For example, a model estimated on the training data, and MSE evaluated on the validation data would be MSE(train,valid). train and valid refer to a 50/50 split of the data set (proportion doesn’t really matter). Let full refer to the entire data set.

## MSE(train,train) versus MSE(train,valid)

* MSE(train,train) is smaller.

## MSE(train,valid) versus MSE(valid,valid)

* MSE(valid,valid) is smaller.

## MSE(full,full) versus MSE(valid,valid)

* MSE(valid,valid) is smaller.

## MSE(train,full) versus MSE(full,full)

* MSE(full,full) is smaller.

## MSE(valid,train) versus MSE(train,train)

* MSE(train,train) is smaller.