5장 Decomposition 모델2

Data structure & basic concept

```
# 시계열 자료 예1 (사분기 자료(2001년 2분기부터 10개 관찰 값)
d1=ts(c(1,3,2,4,3,2,4,1,5,4), frequency=4, start=c(2001, 2));
x11(); plot(d1, main='time series');
class(d1); start(d1); end(d1); cycle(d1); frequency(d1);
window(d1, c(2001,3), c(2002,4)); tsp(d1); # 시작 종료 주기

# 시계열 자료 예2 (월 단위 자료(2011년 3월부터 16개 관찰 값)
(d2=ts(c(10, 9, 11, 9, 11, 12, 14, 13, 14, 12, 11, 12, 14, 11, 12, 15), frequency=12, start=c(2011, 3)));
x11(); plot(d2, main='time series');

# time lag (시차)
lag1=lag(d2, k=1);   # lag=1

# difference (차분)
diff(d2);          # 1차 차분
diff(d2,2);        # 2차 차분

#실제 데이터 예
x11(); plot(AirPassengers, main='Air Passengers');
x11(); plot(diff(AirPassengers), main='Difference – Air Passengers');
```

Moving average smoothing

```
# Moving average
data(AirPassengers);
m3=filter(AirPassengers, filter=rep(1/3, 3));
m6=filter(AirPassengers, filter=rep(1/6, 6));
m12=filter(AirPassengers, filter=rep(1/12, 12));

x11(); par(mfrow=c(4,1));
plot(AirPassengers, main='AirPassengers');
plot(m3, main='3-points moving average');
plot(m6, main='6-points moving average');
plot(m12, main='12-points moving average');

# Example
da=c(1342, 1442, 1252, 1343, 1425, 1362, 1456, 1272, 1243, 1359, 1412, 1253, 1201, 1478, 1322, 1406,
1254, 1289, 1497, 1208);
da1=ts(da, frequency=4, start=c(2006, 1));

# 단순 이동평균 평활법
(f=filter(da1, filter=rep(1, 4)/4, method='convolution', sides=1));
x11(); plot(da1, main='simple moving average smoothing');
lines(f, col='red', lty=2, lwd=2); abline(h=mean(da1), col='red');

res=f[-c(1:3)]-da1[-c(1:3)];
x11(); tsdisplay(res, main='residuals by moving average'); Box.test(res);
```

Moving average forecasting

```
################  ma 함수 이용
# 이동평균 평활법
library(forecast); w1=rep(1, 4)/4;
(f=filter(da1, filter=w1, method='convolution', sides=1));
(f2=filter(da1, filter=w1, method='convolution', sides=2));
(mm1=ma(da1, order=4, centre=T)); #중심 있는 이동평균(5개 자료 평균)
(mm2=ma(da1, order=4, centre=F)); #중심 없는 이동평균(4개 자료 평균)
res2=mm1-da1; Box.test(res2);

library(forecast);
# 예측 ses(simple exponential smoothing) 함수 이용
sa=ses(da1); summary(sa);
forecast(sa, h=4); accuracy(sa);
```

Trend & Seasonality

```
# Example 1
nd1=c(1142, 1242, 1452, 1543, 1125, 1262, 1456, 1572, 1143, 1259, 1462, 1553, 1121, 1258, 1472, 1546,
1154, 1249, 1477, 1548);  # 계절 변동
nd2=c(1142, 1242, 1452, 1543, 1225, 1362, 1556, 1672, 1343, 1459, 1662, 1753, 1421, 1558, 1772, 1846,
1554, 1649, 1877, 1948);  # 계절, 추세 변동
ds=ts(nd1, start=c(2016,1), frequency=4); x11(); plot(ds);
dst=ts(nd2, start=c(2016,1), frequency=4); x11();plot(dst);

# additive model
x11(); plot(decompose(ds, type='additive'));
x11(); plot(decompose(dst, type='additive'));
ad=decompose(dst, type='additive'); ad$seasonal; ad$trend; ad$random;
del_s=dst-ad$seasonal; # 계절성 제거
del_t=dst-ad$trend      # 추세선 제거
x11(); plot(del_s, main='after deleting seasonality');
x11(); plot(del_t, main= 'after deleting trend');

# multiplicative model
x11(); plot(decompose(ds, type='multiplicative'));
x11(); plot(decompose(dst, type='multiplicative'));
mu=decompose(dst, type='multiplicative'); mu$seasonal; mu$trend; mu$random;
del_s1=dst/mu$seasonal;    # 계절성 제거
del_t1=dst/mu$trend        # 추세선 제거
x11(); plot(del_s1, main='after deleting seasonality');
x11(); plot(del_t1, main= 'after deleting trend');
```

Prediction (additive: trend+season)

```
# Example 2: Tourist
tourist=read.table("Tourists.txt", header=T) ;
head(tourist);

library(ggplot2); x11();
ggplot(data=tourist, aes(x=time, y=tourists))+geom_point()+geom_smooth(method="lm");
n=dim(tourist)[1];
in_sample=tourist[1:(n-12),] ;                    # in-sample: 1992~1997
out_sample=tourist[(n-11):n,]  ;                  # out-of-sample: 1998
out=lm(tourists~time+factor(month), data=in_sample);
summary(out)  ;

pred=predict(out, newdata=out_sample) ;           # prediction using new data
actual=out_sample$tourists;
ms=mean((actual-pred)^2)  ;                       #mse
sqrt(ms) ;                                        #rmse
```

Prediction (multiplicative: season+trend)

```
# Example 3: Tie shop
tie_shop=read.table("tie_shop.txt", header=T); head(tie_shop);

tie_shop$time=seq(1:12) ;              # create a new variable(time)
n=dim(tie_shop)[1];

library(ggplot2); ggplot(data=tie_shop, aes(x=time, y=sales))+geom_point();
x=tie_shop$sales; x1=ts(x, start=c(1,1), end=c(4,3), frequency=3);
m=decompose(x1, type="multiplicative");          # seasonal decomposition
season=m$figure;                    # seasonal index
x11(); plot(m)

# deseaonalization
for(i in c(1,4,7,10)){
   tie_shop$deseason[i]=tie_shop$sales[i]/season[1]
   }
for(i in c(2,5,8,11)){
   tie_shop$deseason[i]=tie_shop$sales[i]/season[2]
   }
for(i in c(3,6,9,12)){
   tie_shop$deseason[i]=tie_shop$sales[i]/season[3]
   }
out1=lm(deseason~time, data=tie_shop)
summary(out1)
```

Prediction (multiplicative: season+trend)

```
# trend prediction w/ deseasonal data
pred1=predict(out1, data=tie_shop);

# seasonality back
tie_shop$trend=pred1;
for(i in c(1,4,7,10)){
 tie_shop$forecast[i]=tie_shop$trend[i]*season[1]
  }
for(i in c(2,5,8,11)){
 tie_shop$forecast[i]=tie_shop$trend[i]*season[2]
  }
for(i in c(3,6,9,12)){
  tie_shop$forecast[i]=tie_shop$trend[i]*season[3]
  }
############## forecasting # trend prediction w/ de-seasonal data
new1=data.frame(time=c(13,14,15))        # new data set
pred2=predict(out1, newdata=new1)      # trend prediction using de-seasonal data

# seasonality back
forecast2=rep(0,3);
for(i in 1:3){
 forecast2[i]=pred2[i]*season[i]
  }
```