

Chapter. 05

R시계열 라이브러리 활용

| 앙상블 기반 예측 함수 R function 만들기

FASTCAMPUS
ONLINE

금융공학/퀀트 I

강사. 김경륜

4가지 방법론 고려: stl/tbats/arima/neuralnet
multi seasonality 적용

```
ensemble_v1 <- function(x, frq1, frq2, fh){
```

```
  r1 <- ensemble_test(x, frq1, frq2, fh)
```

```
  p1 <- which(r1==max(r1))
```

```
  qt = x
```

```
  y1 <- ts(qt, frequency = frq1)
```

```
  y2 <- msts(qt, seasonal.periods = c(frq1, frq2))
```

```
  if(p1 == 1){r2 <- forecast(stl(y1, s.window = "per"), h=fh)$mean}
```

```
  else if(p1 == 2){r2 <- forecast(tbats(y2), h=fh)$mean}
```

```
  else if(p1 == 3){r2 <- forecast(auto.arima(y1), h=fh)$mean}
```

```
  else{r2 <- forecast(nnetar(y2), h=fh)$mean}
```

```
  return(list(r1, r2))
```

```
}
```

입력데이터

1. x = 시계열데이터
2. frq1/frq2 = 첫번째/두번째 Frequency
3. fh = 예측할 길이

```
ensemble_test <- function(x, frq1, frq2, fcst_h){
```

```
  qt = x
  len1 = length(qt)
  qt_ms = qt[1:(len1-fcst_h)] # 모델링셋
  qt_ts = qt[(len1-fcst_h+1):len1] # 테스트셋
  y1 <- ts(qt_ms, frequency = frq1) # single seasonality
  y2 <- msts(qt_ms, seasonal.periods = c(frq1, frq2)) # multi-seasonality
```

```
  fc1 <- tryCatch(forecast(stl(y1, s.window = "per"), h=fcst_h), error=function(e) 0) # stl 모델링
  fc2 <- tryCatch(forecast(tbats(y2), h=fcst_h), error=function(e) 0) # tbats 모델링
  fc3 <- tryCatch(forecast(auto.arima(y1), h=fcst_h), error=function(e) 0) # arima 모델링
  fc4 <- tryCatch(forecast(nnetar(y1), h=fcst_h), error=function(e) 0) # nnetar 모델링
```

```
  MAPEr <- c(0, 0, 0, 0)
```

```
  if(sum(is.list(fc1))==1){
    fc1$mean[fc1$mean<0]<-0
    MAPEr[1] <- 100-100*(abs(sum(fc1$mean)-(sum(qt_ts)))/(sum(qt_ts))) } else{MAPEr[1] <- 0} # stl 1-MAPE
  if(sum(is.list(fc2))==1){
    fc2$mean[fc2$mean<0]<-0
    MAPEr[2] <- 100-100*(abs(sum(fc2$mean)-(sum(qt_ts)))/(sum(qt_ts))) } else{MAPEr[2] <- 0} # tbats 1-MAPE
  if(sum(is.list(fc3))==1){
    fc3$mean[fc3$mean<0]<-0
    MAPEr[3] <- 100-100*(abs(sum(fc3$mean)-(sum(qt_ts)))/(sum(qt_ts))) } else{MAPEr[3] <- 0} # arima 1-MAPE
  if(sum(is.list(fc4))==1){
    fc4$mean[fc4$mean<0]<-0
    MAPEr[4] <- 100-100*(abs(sum(fc4$mean)-(sum(qt_ts)))/(sum(qt_ts))) } else{MAPEr[4] <- 0} # neuralnet 1-MAPE
  return(MAPEr)
}
```

입력데이터

1. x = 시계열데이터
2. frq1/frq2 = 첫번째/두번째 Frequency
3. fh = 예측할 길이