Time Series Analysis

KM

2018.07.06･

# Section 1

## LEcture 3: Main Functions

1. Basic functions in R Base are:

* decompose()
* stl()
* arima()
* HolWinters()
* acf(), pacf()
* plot()
* ts()

1. Add-on package - forecast package

# package load  
library(forecast)

## Warning: package 'forecast' was built under R version 3.5.1

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.5.1

### Forecast structure

Function forecast() + standard model = Forecasting

Standar model is (1) ARIMA, (2) Sesonal decomposition, (3) Exponential smoothing, and (4) simple models (e.g., naive mean).

Forecasting Results are always in the same structure regardless of the model(which are different from function predict()).

In the Forecast library, automatic parameter selection is available both for the ARIMA model and the exponential smoothing.

### ARIMA models in Library Forecast

auto.arima(time\_series) -> most suitable ARIMA model auto.arima sets the complexity with the arguments stepwise and approximation. WE can get a list of possible models, as well as the information criteria.

Arima(time\_series) we could set the parameters manually (manual parameter selecion). In the manual selection, we look fro autocorrelation based on the acf() and pacf() plots. we adjust the model by time\_series - lags until no more autocorrelation.

### Exponential smooothing in Library Forecast

Automatic function - ets() manual functions: ses(), hw(), holt()

### Plotting with Library Forrecast

R Base Plots: - plot() - monthplot() - seasonplot()

ggplot2 plots: - autoplot() - ggmothplot() - ggseasonplot()

### Model comparison with Library Forecast

accruacy() getting the model accuracy with a training and a test set.

tsCV() time series cross validation for small datasets.

### Other packages

we would use getSymbols(), quantmod() and xts() packages.

# install.packages("quantmod")  
# install.packages("xts")

## Lecture 4: Supporting Resources

R Time series Task View curated by Rob Hyndman (Blog on Otexts.org) <https://robjhyndman.com/>

Vignetts some packages come with vignettes (description, PDF)

Free e-Books “Forecasting: Principles and Practice” <https://otexts.org/fpp2/index.html> <https://robjhyndman.com/seminars/uwa2017/>

R Community on Stackoverflow.com <https://stackoverflow.com/questions/tagged/r>

## Lecture 5: Course Link List

Time Series Task View: <https://cran.r-project.org/web/views/TimeSeries.html>

Blog, Ebook and Forecast Documentation by Rob Hyndman: <https://otexts.org/fpp2/intro.html>

Stackoverflow Community: <https://stackoverflow.com/questions>

Singapur Data of Project I: <https://docs.google.com/spreadsheets/d/1frieoKODnD9sX3VCZy5c3QAjBXMY-vN7k_I9gR-gcU8/pub> <http://www.gapminder.org/data/>

German Inflation Data of Project II: <https://www.statbureau.org/en/germany/inflation-tables>

# Section 2: Project I Trending <Data:Singapur> Labor Force Participation Rate

## LEcture 6: Importing the data

Uneployment rate - Used for propaganda purposes - Easy to manipulate - Who is unemployed? - Who doesn’t show up in the metric?

Labor Force Participatin Rate - Harder to manipulate - Ratio of people in the work force vs available people of a particular age range - Factors for manipulation /bias <https://www.gapminder.org/data/>

We need to compara things that are the same or similar level. Unbalanced comparisons require well thought out methodology to adjust for a mismatch.

# package load  
library(forecast)  
library(ggplot2)  
library(quantmod)

## Warning: package 'quantmod' was built under R version 3.5.1

## Loading required package: xts

## Warning: package 'xts' was built under R version 3.5.1

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.5.1

##   
## Attaching package: 'zoo'

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

## Loading required package: TTR

## Version 0.4-0 included new data defaults. See ?getSymbols.

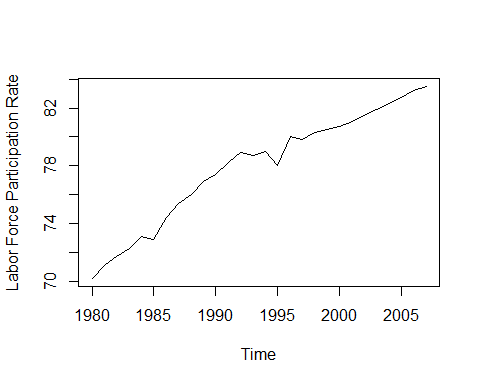
library(xts)  
  
# Import with scan  
# "70.19999695 71.09999847 71.69999695 72.30000305 73.09999847 72.90000153 74.40000153 75.40000153 76 76.90000153 77.40000153 78.19999695 78.90000153 78.69999695 79 78 80 79.80000305 80.30000305 80.5 80.69999695 81.09999847 81.5 81.90000153 82.30000305 82.69999695 83.19999695 83.5"  
# singapur=scan()  
# singapur  
singapur <-c(70.19999695, 71.09999847, 71.69999695, 72.30000305, 73.09999847, 72.90000153, 74.40000153, 75.40000153, 76, 76.90000153, 77.40000153, 78.19999695, 78.90000153, 78.69999695, 79, 78, 80, 79.80000305, 80.30000305, 80.5, 80.69999695, 81.09999847, 81.5, 81.90000153, 82.30000305, 82.69999695, 83.19999695, 83.5)  
# item 28 are shown by this code   
singapur

## [1] 70.2 71.1 71.7 72.3 73.1 72.9 74.4 75.4 76.0 76.9 77.4 78.2 78.9 78.7  
## [15] 79.0 78.0 80.0 79.8 80.3 80.5 80.7 81.1 81.5 81.9 82.3 82.7 83.2 83.5

singapur <- ts(singapur,start=1980)  
singapur

## Time Series:  
## Start = 1980   
## End = 2007   
## Frequency = 1   
## [1] 70.2 71.1 71.7 72.3 73.1 72.9 74.4 75.4 76.0 76.9 77.4 78.2 78.9 78.7  
## [15] 79.0 78.0 80.0 79.8 80.3 80.5 80.7 81.1 81.5 81.9 82.3 82.7 83.2 83.5

plot(singapur,ylab="Labor Force Participation Rate")



From the above chart, the possible models are: - ARIMA - Holt linear trend method

However, one feature of this data is that the values cannnot exceed 100% in the model.There needs to be some press hold. Holt method has a nice damping parameter (holt()).

## Lecture 7: Mission Statement

From this lecture, we would focous on the theory and practical implementation through the four questions. - How to handle time series with trend?  
- Which methods are available?  
- What are the pitfalls?  
- How to visualize time series data?

The process is 1. Get the dataset  
2. Get the mission statement  
3. Work on the project  
4. Proceed with lectures and check the work

The potential models are:  
- Linear trend model with holt()  
- With damping paramter  
- Without damping parameter  
- ARIMA model

These models can be applied for forecasts, but with pitfalls, visualizations, alternatives.

For applying the above models, we just need:  
(1) Libraries “forecast” and “ggplo2” + R base functions  
(2) Resources from the introductory section

## Lecture 9: Exponential smoothing

Forecast package - Simple exponential smooothing: ses() - Holt’s linear trend model]: holt() +damped - Holt-Winters seasonal method: hw() - Automated exponential smooothing: ets()

Holt linear trend model is:

- : estimated forecast value at time point  
- : level value at time point  
- : trend value at time point multiplied by

Should the model react only to recent data or should it consider older data as well? -> Smoothing parameters

Smoothing parameters of a Holt Linear Trend Model are:  
1. :smoothing parameter for the level 2. :smoothing parameter for the trend

The closer the smoothing parameter is to zero, the model becomes smoooth model (older date is consdered too), otherwise (close to 1) the model is the reactive model, heavily relying on recent data.

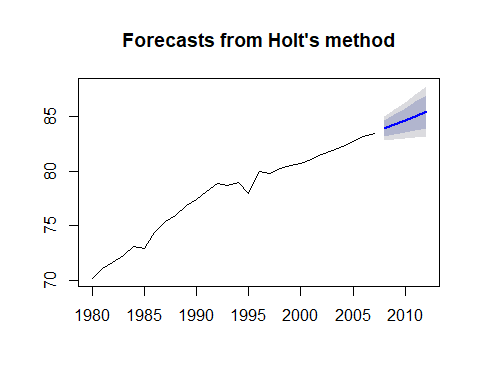
## Lecture 10: The Holt Linear Trend Model

In the Project I, we use holt linear trend model. In practice, we firstly activate the library forecast, and use the holt() function to create the model.  
- data=the time series to model  
- h=forecast length

# holt(data, h=x, damped=FALSE, level=c(80,95),fan=FALSE,initial=c("optimal","simple"),exponential=FALSE,alpha=NULL,beta=NULL,phi=NULL,lambda=NULL)  
  
library(forecast)  
holt\_trend <- holt(singapur,h=5)  
summary(holt\_trend)

##   
## Forecast method: Holt's method  
##   
## Model Information:  
## Holt's method   
##   
## Call:  
## holt(y = singapur, h = 5)   
##   
## Smoothing parameters:  
## alpha = 0.6378   
## beta = 0.1212   
##   
## Initial states:  
## l = 69.619   
## b = 0.6666   
##   
## sigma: 0.5529  
##   
## AIC AICc BIC   
## 65.79969 68.52697 72.46072   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE  
## Training set -0.08247984 0.5118728 0.3084414 -0.1056525 0.3986079  
## MASE ACF1  
## Training set 0.5047223 -0.09190023  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2008 83.90071 83.19216 84.60926 82.81708 84.98435  
## 2009 84.28751 83.39799 85.17703 82.92711 85.64791  
## 2010 84.67431 83.58796 85.76065 83.01289 86.33573  
## 2011 85.06111 83.76361 86.35860 83.07676 87.04545  
## 2012 85.44790 83.92606 86.96975 83.12045 87.77536

plot(holt\_trend)



Most of the above arguments are the same for ses() and hw(). By summary() function, we can obtain smoothing parameters, and initial state values. In the Singapur case, the participation cannot exceed

If we encounter simiular situation, exact parameters and thresholds. They know the literature, and experience, background information.

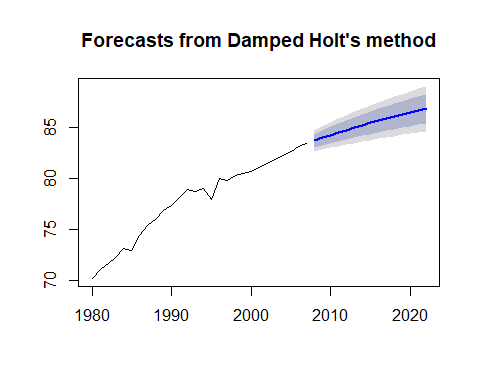
For the above case, we use the damped argument. In the Holt linear trend model, the smoothing parameters are:  
- : smoothing parameter for the level alpha= - : smoothing parameter for the trend beta= - : damping parameter 0<<1 phi=

We can easily adjust the model by the damping parameter.

If is one, its close to the original slope of the Holt trend model. If the is between 0.85 to 0.95, it is the generally recommended range of phi. If the is close to 0, its a flattened curve.

**When the damped parameter is used, the slope of the trend cannnot be constant: It changes over time**

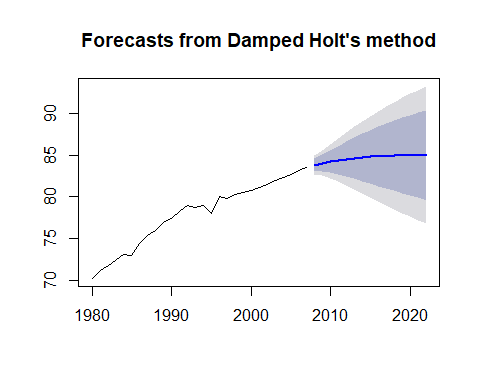
#phi auto generated  
plot(holt(singapur,h=15,damped=T))



# To see the generated value for phi: phi=0.96  
summary(holt(singapur,h=15,damped=T))

##   
## Forecast method: Damped Holt's method  
##   
## Model Information:  
## Damped Holt's method   
##   
## Call:  
## holt(y = singapur, h = 15, damped = T)   
##   
## Smoothing parameters:  
## alpha = 0.5149   
## beta = 1e-04   
## phi = 0.9666   
##   
## Initial states:  
## l = 69.5404   
## b = 0.771   
##   
## sigma: 0.5149  
##   
## AIC AICc BIC   
## 62.62649 66.62649 70.61972   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.01419094 0.4667051 0.3250851 0.01746896 0.419571 0.5319574  
## ACF1  
## Training set -0.001595197  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2008 83.72693 83.06701 84.38685 82.71766 84.73620  
## 2009 84.00520 83.26291 84.74750 82.86996 85.14044  
## 2010 84.27418 83.45776 85.09060 83.02557 85.52279  
## 2011 84.53417 83.64979 85.41856 83.18163 85.88672  
## 2012 84.78548 83.83798 85.73298 83.33641 86.23456  
## 2013 85.02840 84.02171 86.03509 83.48880 86.56800  
## 2014 85.26320 84.20060 86.32580 83.63809 86.88831  
## 2015 85.49016 84.37443 86.60589 83.78380 87.19653  
## 2016 85.70954 84.54308 86.87600 83.92560 87.49348  
## 2017 85.92159 84.70651 87.13668 84.06328 87.77990  
## 2018 86.12656 84.86471 87.38841 84.19673 88.05639  
## 2019 86.32469 85.01773 87.63164 84.32587 88.32350  
## 2020 86.51619 85.16563 87.86675 84.45068 88.58170  
## 2021 86.70130 85.30848 88.09412 84.57117 88.83143  
## 2022 86.88023 85.44639 88.31407 84.68736 89.07310

# manual setting of phi  
plot(holt(singapur,h=15,damped=T,phi=0.8))



# ARIMA auto generated  
singapurarima=auto.arima(singapur)  
summary(singapurarima)

## Series: singapur   
## ARIMA(1,1,0) with drift   
##   
## Coefficients:  
## ar1 drift  
## -0.3690 0.4904  
## s.e. 0.1763 0.0720  
##   
## sigma^2 estimated as 0.2779: log likelihood=-20.05  
## AIC=46.1 AICc=47.14 BIC=49.99  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE  
## Training set 0.006855948 0.4981113 0.3755194 0.01821962 0.4863707  
## MASE ACF1  
## Training set 0.6144862 0.05505323

## Lecture 1:: The ARIMA model - Project I

The Box Jenkins models are standard modeling system for time series model. There are three parameters

* AR: Autoregressive such as seasonality, trend P
* |: Integreation - defferencing of the dataset D
* MA: Moving average - movement around a constant mean Q

The ARIMA model is very flexible for explaining: - Random Walk - Exponential Smooothing - Autoregressive models such as AR(1), ARIMA(1,0,0) - Moving average (MA(1), ARIMA(0,0,1))

The labour participation rate can be modeled by ARIMA. For that, we need to take into consideration, trending, autocorrelation (AR). Please note that, it seems there is no no moving average, seasonality, thus no differencing is needed.