Time Series Forecasting
Part 1: Extracting Trend & Seasonal Components

BC2407 Seminar 9

Reference

- Avril Coghlan (2018), A Little Book of R for Time Series
  - https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/
- Chew C.H. (2020) Artificial Intelligence, Analytics and Data Science, Vol. 2.
  - Est. Q4 2020.

What is a Time Series?

- A vector of data measured across time.
- Examples:
  - Share Price at end of each day since IPO till today.
  - Height of a child vs months since birth till last month.
  - Revenue of a company by quarters since incorporation till last quarter.
  - GDP by year since independence till last year.

Difficulty in Forecasting Time Series

- Concerns:
  - To forecast (i.e. to extrapolate) from given data.
  - Circumstances in future might be different compared to existing time series data.
- Industry Practice:
  - Forecast is just a baseline.
  - To be adjusted based on
    - New data arrivals.
    - New events that has a significant impact.

Time Series Forecasting methods

1. Moving Average Method
2. Decomposition Methods
    1. Trend
    2. Seasonal
    3. Random Error
3. Exponential Smoothing Methods
4. Causal Methods
5. ARIMA method

Our course (part 1) will mainly cover the first 3 methods in depth and briefly mention 4 and 5.

Source: Chew C.H. (2020) A.I., Analytics & Data Science, Vol. 2.

Autocorrelation

- Can lagged data points be used to predict the next data points?
- Lag k: k period behind
- Example for monthly data:
  - Today is Mar 2020, Lag 1: Feb 2020, Lag 2: Jan 2020, etc.
- Autocorrelation by lag k
  - Correlation between the vector X_t and vector X_{t-k}

Example of Autocorrelation of Phone Sales by Lags 1, 2 & 3.

	A	B	C	D	E
1	Month	Sales	Lag 1	Lag 2	Lag 3
2	Jan-2013	226			
3	Feb-2013	254	226		
4	Mar-2013	204	254	226	
5	Apr-2013	193	204	254	226
6	May-2013	191	193	204	254
7	Jun-2013	166	191	193	204
8	Jul-2013	175	166	191	193
9	Aug-2013	217	175	166	191
10	Sep-2013	167	217	175	166
11	Oct-2013	192	167	217	175
12	Nov-2013	127	192	167	217
13	Dec-2013	148	127	192	167
14	Jan-2014	184	148	127	192
15	Feb-2014	209	184	148	127
47	Oct-2016	185	175	181	179
48	Nov-2016	245	185	175	181
49	Dec-2016	177	245	185	175

Note that each successive lag just "pushes the variable down" by a row. As an example, the lag 2 residual for July-2013 is just the residual 2 months ago, i.e., for May-2013.

G	H	J
	Autocorrelation	
Lag1	0.357	=CORREL(B3:B49,C3:C49)
Lag2	0.084	=CORREL(B4:B49,D4:D49)
Lag3	0.089	=CORREL(B5:B49,E5:E49)

Source: Chew C.H. (2020) A.I., Analytics & Data Science, Vol. 2.

Create Time Series Object in Base R via ts()

Time-Series Objects

Description

The function `ts` is used to create time-series objects.

`as.ts` and `is.ts` coerce an object to a time-series and test whether an object is a time series.

Usage

```
ts(data = NA, start = 1, end = numeric(), frequency = 1,
   deltat = 1, ts.eps = getOption("ts.eps"), class = , names = )
as.ts(x, ...)
is.ts(x)
```

Arguments

`data`	a vector or matrix of the observed time-series values. A data frame will be coerced to a numeric matrix via `data.matrix`. (See also 'Details'.)
`start`	the time of the first observation. Either a single number or a vector of two integers, which specify a natural time unit and a (1-based) number of samples into the time unit. See the examples for the use of the second form.
`end`	the time of the last observation, specified in the same way as `start`.
`frequency`	the number of observations per unit of time.

Create ts object from HDB Sales Data

```
library("TTR")                # For MA via SMA()
library("forecast")           # To generate h-period ahead forecasts

setwd("C:/NC/Datasets/ML")

flatsales.df <- read.csv("5 room flat resale applications.csv")

# create time series object
flatsales.ts <- ts(flatsales.df$Sales.5rm, frequency = 4, start = c(2007,1))
```

	A	B
1	Quarter	Sales 5rm
2	2007-Q1	1402
3	2007-Q2	2305
4	2007-Q3	1901
5	2007-Q4	1667
6	2008-Q1	1574
7	2008-Q2	1997
8	2008-Q3	2172
9	2008-Q4	1578
10	2009-Q1	1506
11	2009-Q2	2713

Only one column of data values

4 times per year

Data start in 2007, quarter 1

50	2019-Q1	1148
51	2019-Q2	1520
52	2019-Q3	1547
53	2019-Q4	1519

Source: Chew C.H. (2020) A.I., Analytics & Data Science, Vol. 2.

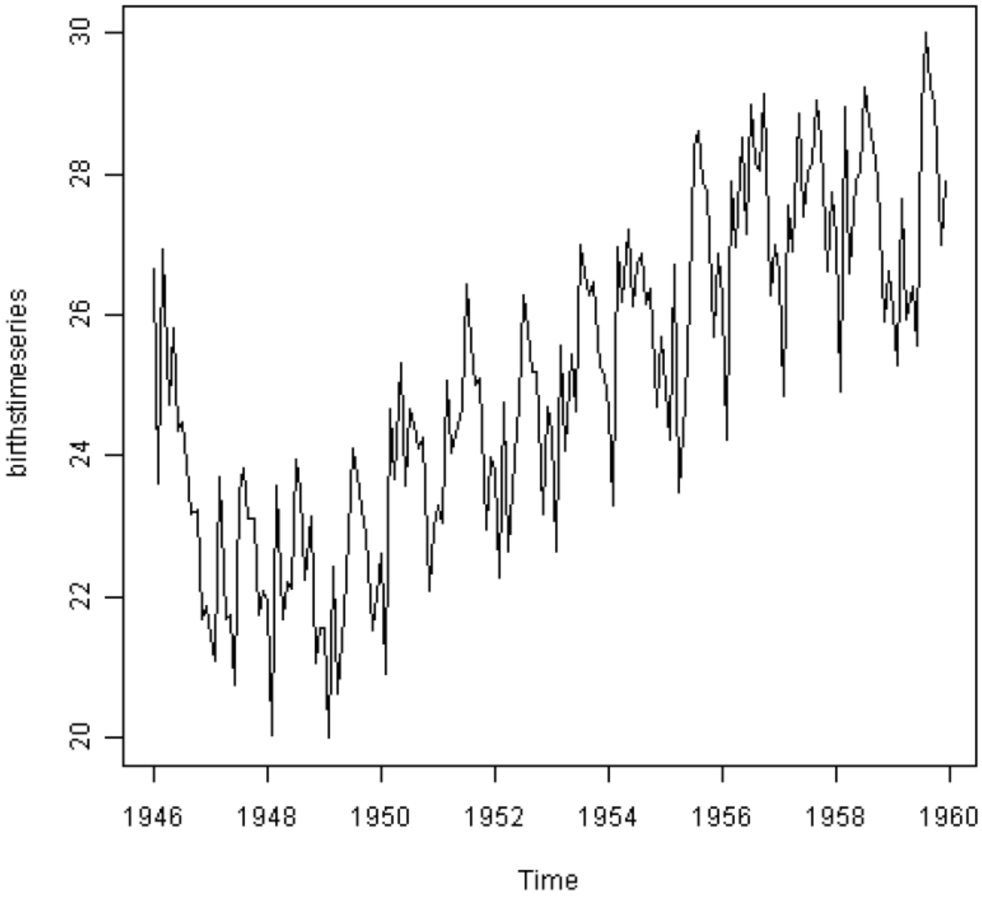
Time Series Object from ts()

```
> flatsales.ts
     Qtr1 Qtr2 Qtr3 Qtr4
2007 1402 2305 1901 1667
2008 1574 1997 2172 1578
2009 1506 2713 3422 2187
2010 2047 2240 1975 1477
2011 1582 1635 1415 1412
2012 1370 1594 1558 1288
2013  962 1070  969  784
2014  726  913  960 1024
2015  925 1210 1140 1113
2016 1023 1369 1283 1186
2017 1055 1407 1428 1403
2018 1111 1559 1797 1350
2019 1148 1520 1547 1519
```

Additive or Multiplicative Time Series Model

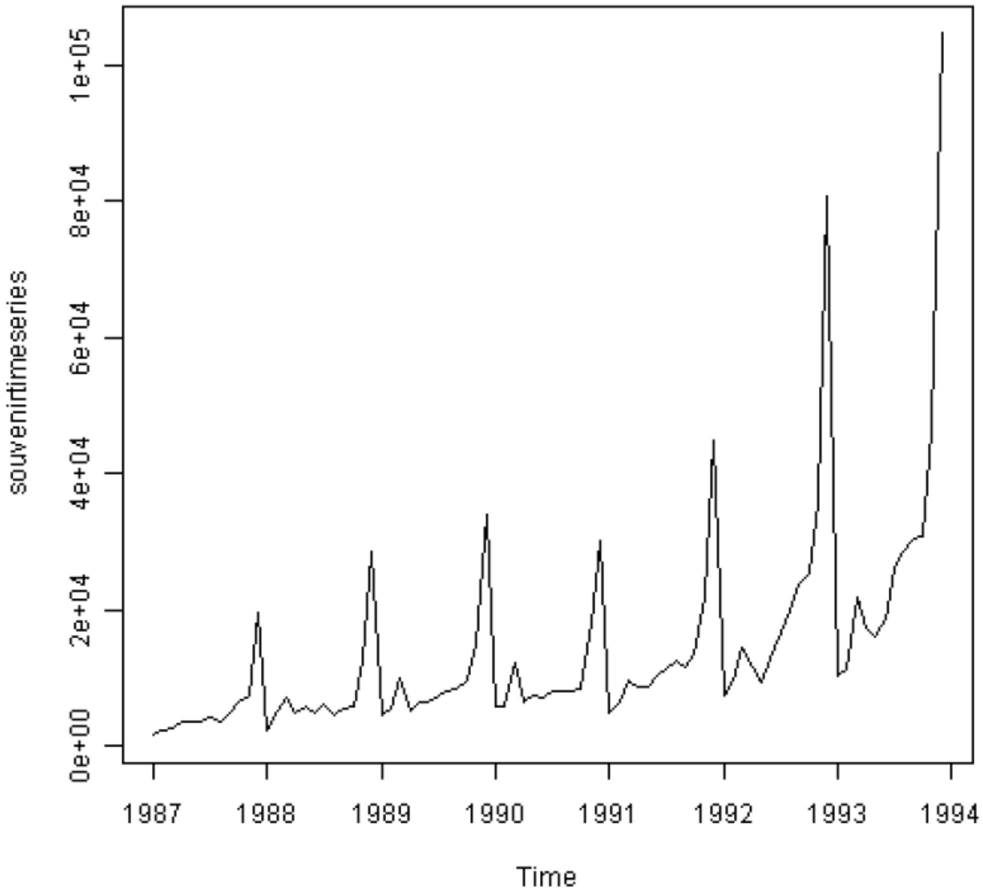
- Affects how to seasonally adjust (i.e. deseasonalize) the time series.
  - i.e. remove the effects of seasonality
  - Additive implies subtract seasonality effects
  - Multiplicative implies divide seasonality effects
- So as to more cleanly estimate the trend component.
- Finally, the forecast will include both trend and seasonality i.e. need to re-seasonalize
  - Additive implies add seasonality effects
  - Multiplicative implies multiply seasonality effects

Constant Fluctuations Over Time -> Additive Time Series



Source: Avril Coghlan (2018), A Little Book of R for Time Series-

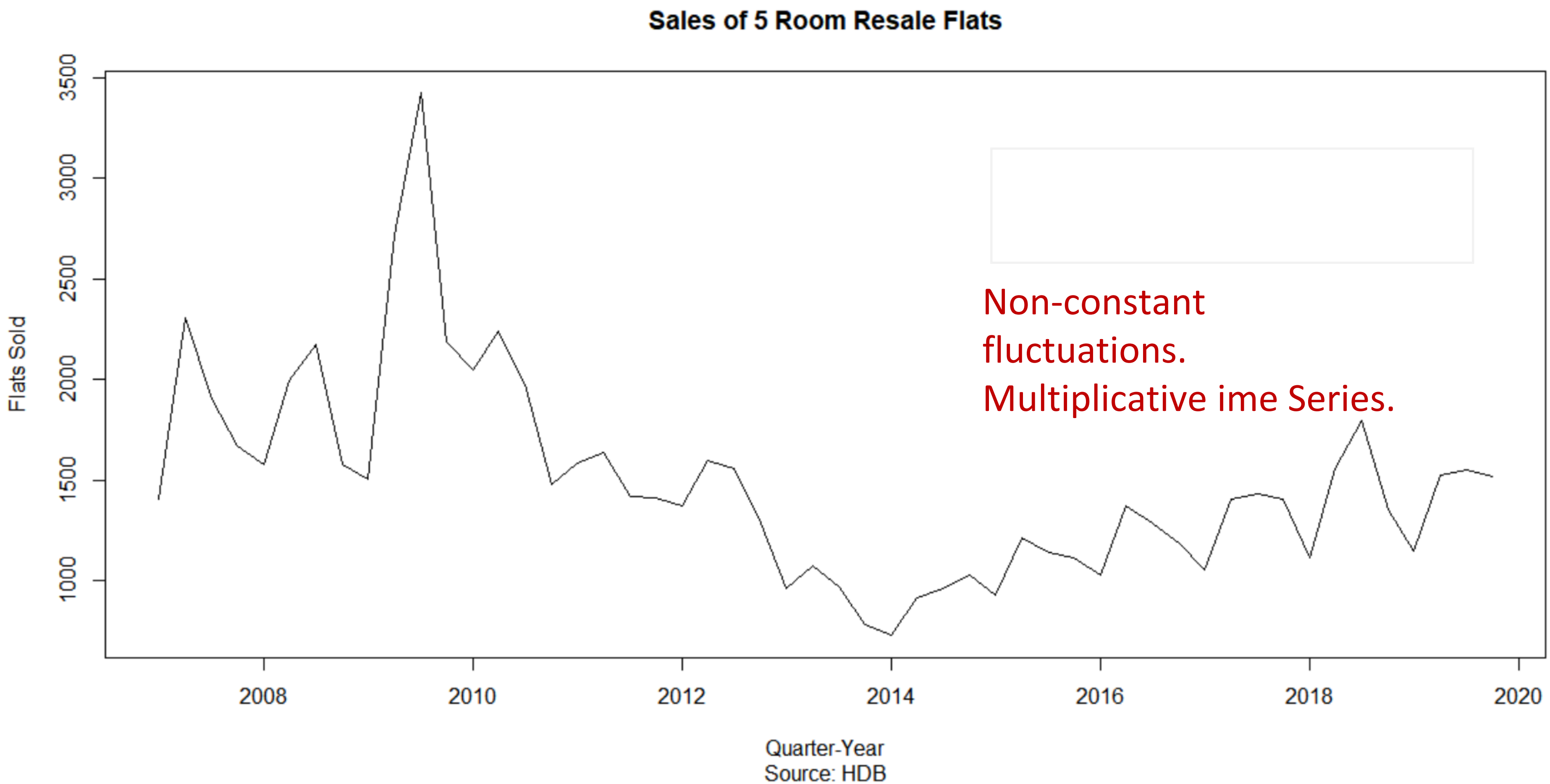
Changing Fluctuations Over Time -> Multiplicative Time Series



Source: Avril Coghlan (2018), A Little Book of R for Time Series-

Flat sales time series is additive or multiplicative?

```
plot.ts(flatsales.ts, ylab = "Flats Sold", xlab = "Quarter-Year",
        main = "Sales of 5 Room Resale Flats",
        sub = "Source: HDB")
```



Non-constant fluctuations.
Multiplicative ime Series.

Source: Chew C.H. (2020) A.I., Analytics & Data Science, Vol. 2.

Moving Average (MA) Forecast

- A moving average is the mean of the observations in the past few periods, where the number of terms considered in the mean is called the span.
- The larger, the span, the more items are averaged, and thus looks more smoothed.
- Thus, Moving Avg is also considered as a (simple) Smoothing Method.

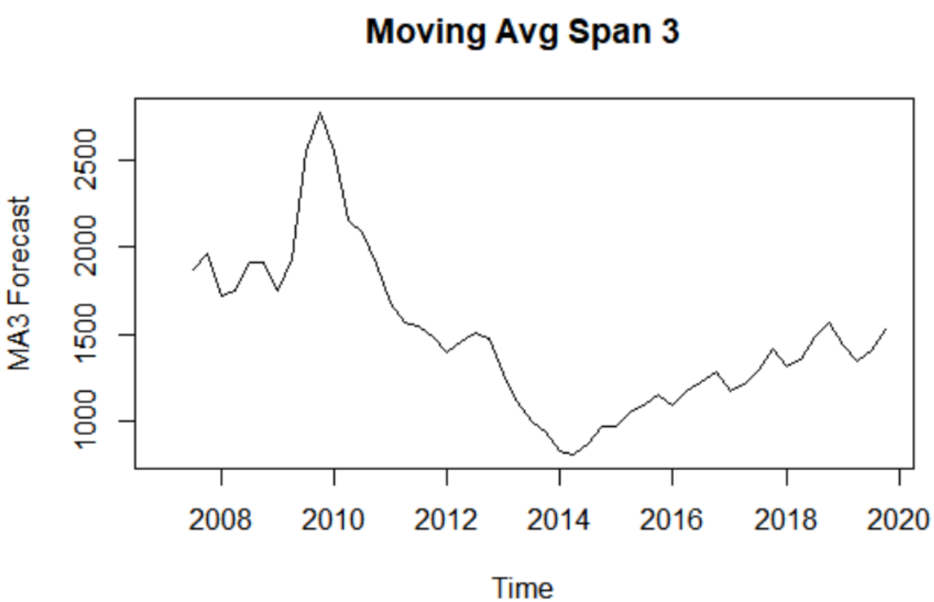
Setting the Span in Moving Avg

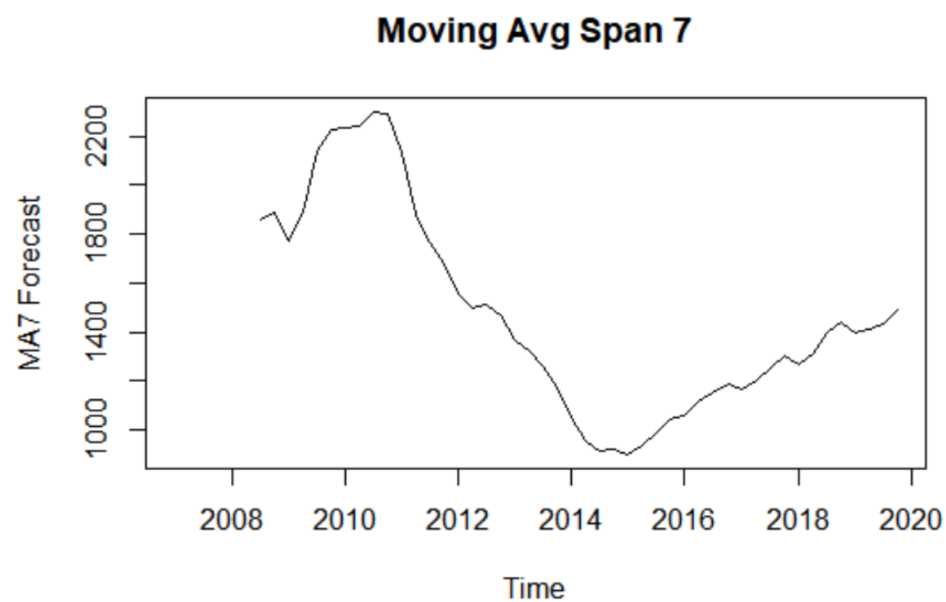
- Setting a span requires some judgment:
  - If you think fluctuations in the series are mainly due to random noise, use a relatively large span.
  - Otherwise, use a smaller span.

Moving Averages with Span = 3 vs 7

What's the differences?

```
m.ma3 <- SMA(flatsales.ts, n = 3)
plot(m.ma3, main = "Moving Avg Span 3", ylab = "MA3 Forecast")

m.ma7 <- SMA(flatsales.ts, n = 7)
plot(m.ma7, main = "Moving Avg Span 7", ylab = "MA7 Forecast")
```

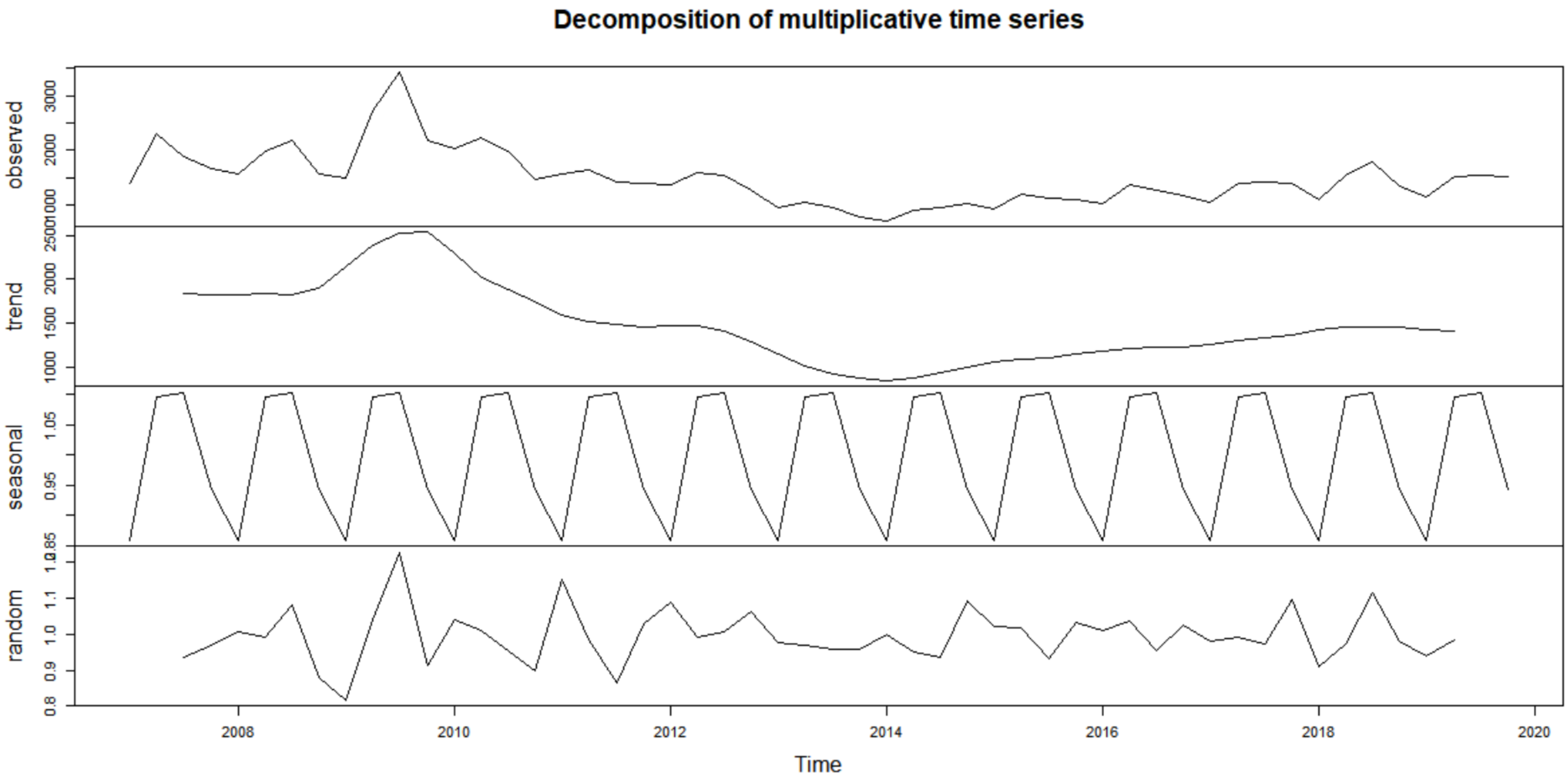




MA7 is smoother (less fluctuations) and start later than MA3.

MA Based Decomposition of Time Series into Trend and Seasonal Components

```r
# Classical Seasonal Decomposition by Moving Averages
m.ma.mul <- decompose(flatsales.ts, type = "multiplicative")
plot(m.ma.mul)
```



MA vs Exponential Smoothing

- MA essentially ignores data beyond the span and only considers the most recent “window” of data.
- In contrast, exponential smoothing considers all the data values since the start date in the dataset, but weighs each data value.
  - The more recent data has higher weights than older data.

Source: Chew C.H. (2020) A.I., Analytics & Data Science, Vol. 2.

Exponential Smoothing Methods

1. Simple Exponential Smoothing
    - for a series without trend and seasonality.
2. Holt’s method
    - for a series with trend but no seasonality.
3. Winters’ method (or Holt-Winters’ method)
    - for a series with seasonality and possibly trend.

Source: Chew C.H. (2020) A.I., Analytics & Data Science, Vol. 2.

Simple Exponential Smoothing

- Every exponential model has at least one **smoothing parameter**, between 0 and 1.
- Simple exponential smoothing has a single smoothing constant denoted by α.
- The *level* of the series at time *t* (L_t) is an estimate of where the series would be at time *t* if there were no random noise.
- The simple exponential method is defined by the following two equations:

$$L_t = \alpha Y_t + (1 - \alpha) L_{t-1}$$

$$F_{t+k} = L_t$$

1. *What is the role played by α?*
2. *How many data points are used in calculating L_t?*
3. *Is L_t used to forecast Y_t?*

  - The *k*-period-ahead forecast, F_{t+k}, of Y_{t+k} made at period *t* is essentially the most recently estimated level, L_t.

Simple Exponential Smoothing on 5 Room Flat Sales

Est. 15 mins

- Using Excel, execute simple exponential smoothing on the 5 room resale flat data with:
  - *α = 0.2*
  - *α = 0.8*
- What value did you set for the first value of L_t to kickstart the model? *[Note: more than 1 way]*
- What did you observe when you use the different alpha values above? Which value is more suitable in your opinion?
- What is your one year ahead forecast of sales (i.e. 2020 Q1 to 2020 Q4) with each of the two alphas?

Holt's method

- When there is a trend in the series, Holt's method deals with it explicitly by including a trend term, T_t, and it's corresponding smoothing constant β.
  - The interpretation of L_t is exactly as before.
  - The interpretation of T_t is that it represents an estimate of the *change* in the series from one period to the next.
- The equations for Holt's model are:

$$L_t = \alpha Y_t + (1 - \alpha)(L_{t-1} + T_{t-1})$$

$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$$

$$F_{t+k} = L_t + kT_t$$

What value can be set for the first value of T_t to kickstart the model? [More than 1 way]

Seasonality

- Seasonality is the consistent pattern that repeats each year. Eg: month-to-month, quarter-to-quarter, events.
  - The easiest way to check for seasonality is graphically: Look for a *regular* pattern of ups and/or downs.
- There are three methods for dealing with seasonality:
  - Winters' exponential smoothing model
  - Deseasonalize the data (then use any forecasting method to model the deseasonalized data and finally, "reseasonalize" these forecasts)
  - Multiple regression with dummy variables for the seasons

Winter's method (aka Holt-Winters' method

- Winters' model is very similar to Holt's model, but it adds seasonal indexes and a corresponding smoothing constant γ.
  - The new smoothing constant controls how quickly the method reacts to observed changes in the seasonality.
    - If the constant is small, the method reacts slowly.
    - If it is large, the method reacts more quickly.
  - The equations for this method are shown below:

$$L_t = \alpha \frac{Y_t}{S_{t-M}} + (1 - \alpha)(L_{t-1} + T_{t-1})$$

$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$$

$$S_t = \gamma \frac{Y_t}{L_t} + (1 - \gamma)S_{t-M}$$

$$F_{t+k} = (L_t + kT_t)S_{t+k-M}$$

1. Is this a additive model or multiplicative model?
2. Why is the subscript for S: t – M instead of t – 1?
3. What value can be set for the first values of S_t to kickstart the model? [More than 1 way]

Simple Exponential Smoothing

```
# Simple Exponential Smoothing ---------------------------------------------
m.ses <- HoltWinters(flatsales.ts, seasonal = "multiplicative", beta=F, gamma=F)

m.ses
## Optimal value of alpha = 0.5288797

plot(m.ses)

m.ses$fitted

m.ses$alpha*1519 + (1-m.ses$alpha)*1485.8267
## verifying the meaning of coef = 1503.371 is L_{last t}

RMSE.ses <- round(sqrt(m.ses$SSE/nrow(m.ses$fitted)))
## 359
```

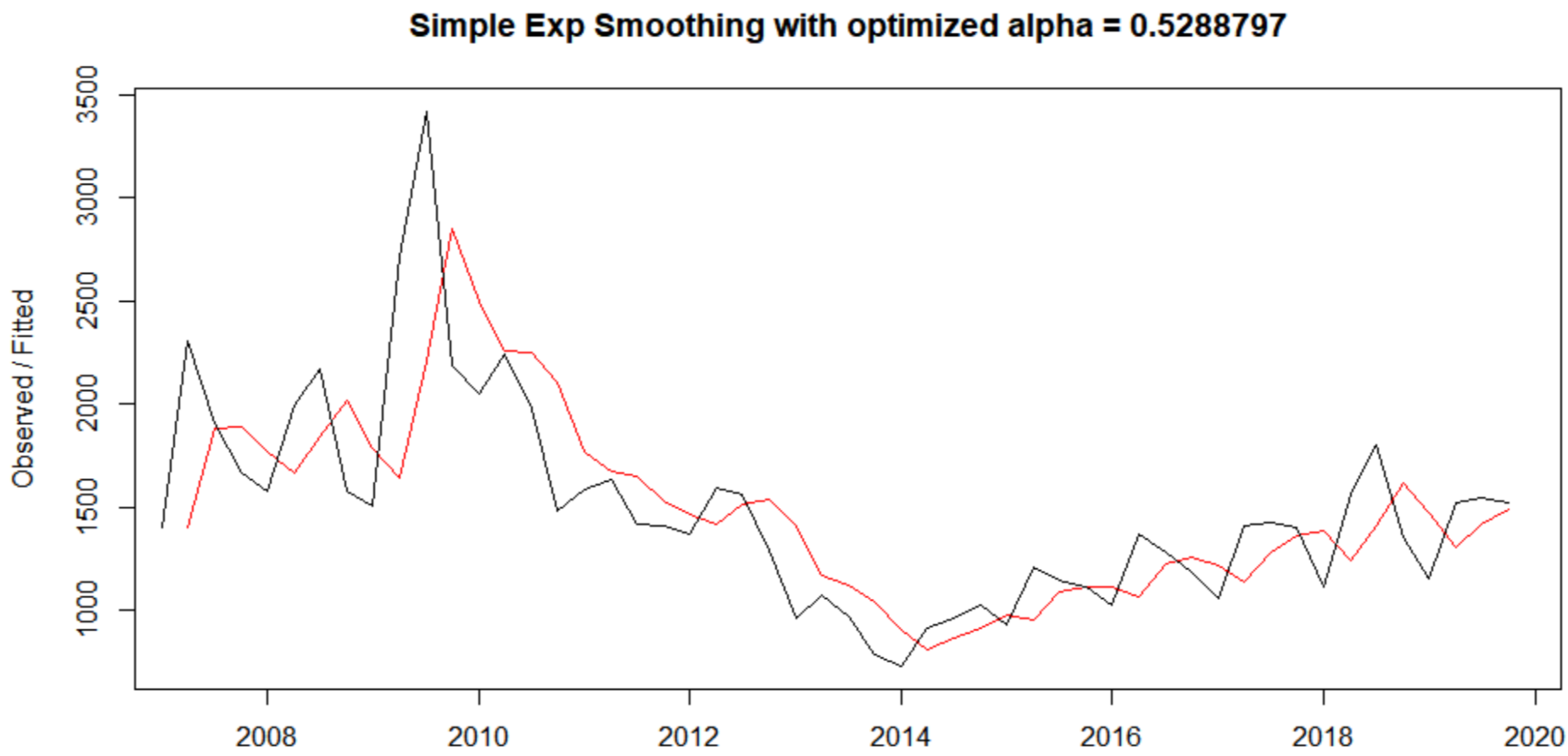
```
> m.ses
Holt-Winters exponential smoothing without trend and with
out seasonal component.

Call:
HoltWinters(x = flatsales.ts, beta = F, gamma = F, season
al = "multiplicative")

Smoothing parameters:
 alpha: 0.5288797
 beta : FALSE
 gamma: FALSE

Coefficients:
      [,1]
a 1503.371
```

How to get this?

What is this?

1. Min SSE of one period ahead forecast.
2. The last value of L_t

```
41  plot(m.ses, main = "Simple Exp Smoothing with optimized alpha = 0.5288797")
```



Which coloured line rep observed vs one period ahead SES forecast?

Black line rep observed; Red line rep one period ahead SES forecast.

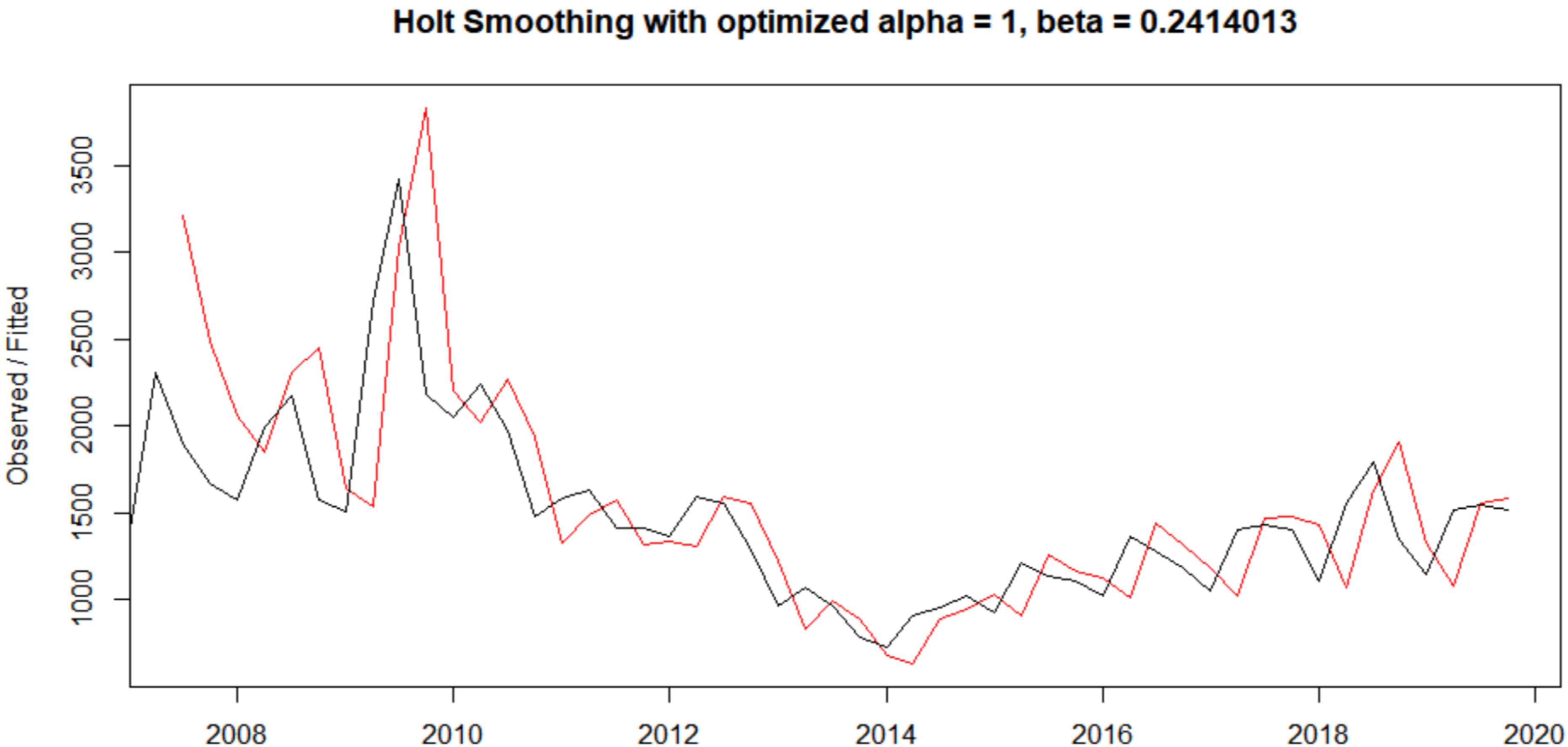
Holt Smoothing Method

```
# Holt's method -----------------------------------------------
m.holt <- HoltWinters(flatsales.ts, seasonal = "multiplicative", gamma=F)

m.holt
## Optimal value of alpha = 1, beta = 0.2414013

plot(m.holt, main = "Holt Smoothing with optimized alpha = 1, Beta = 0.2414013")

RMSE.holt <- round(sqrt(m.holt$SSE/nrow(m.holt$fitted)))
```

```
> m.holt
Holt-Winters exponential smoothing with trend and without seasonal component.

Call:
HoltWinters(x = flatsales.ts, gamma = F, seasonal = "multiplicative")

Smoothing parameters:
 alpha: 1
 beta : 0.2414013
 gamma: FALSE

Coefficients:
        [,1]
a 1519.00000
b   21.43562
```

Source: Chew C.H. (2020) A.I., Analytics & Data Science, Vol. 2.

```
57 plot(m.holt, main = "Holt Smoothing with optimized alpha = 1, beta = 0.2414013")
```

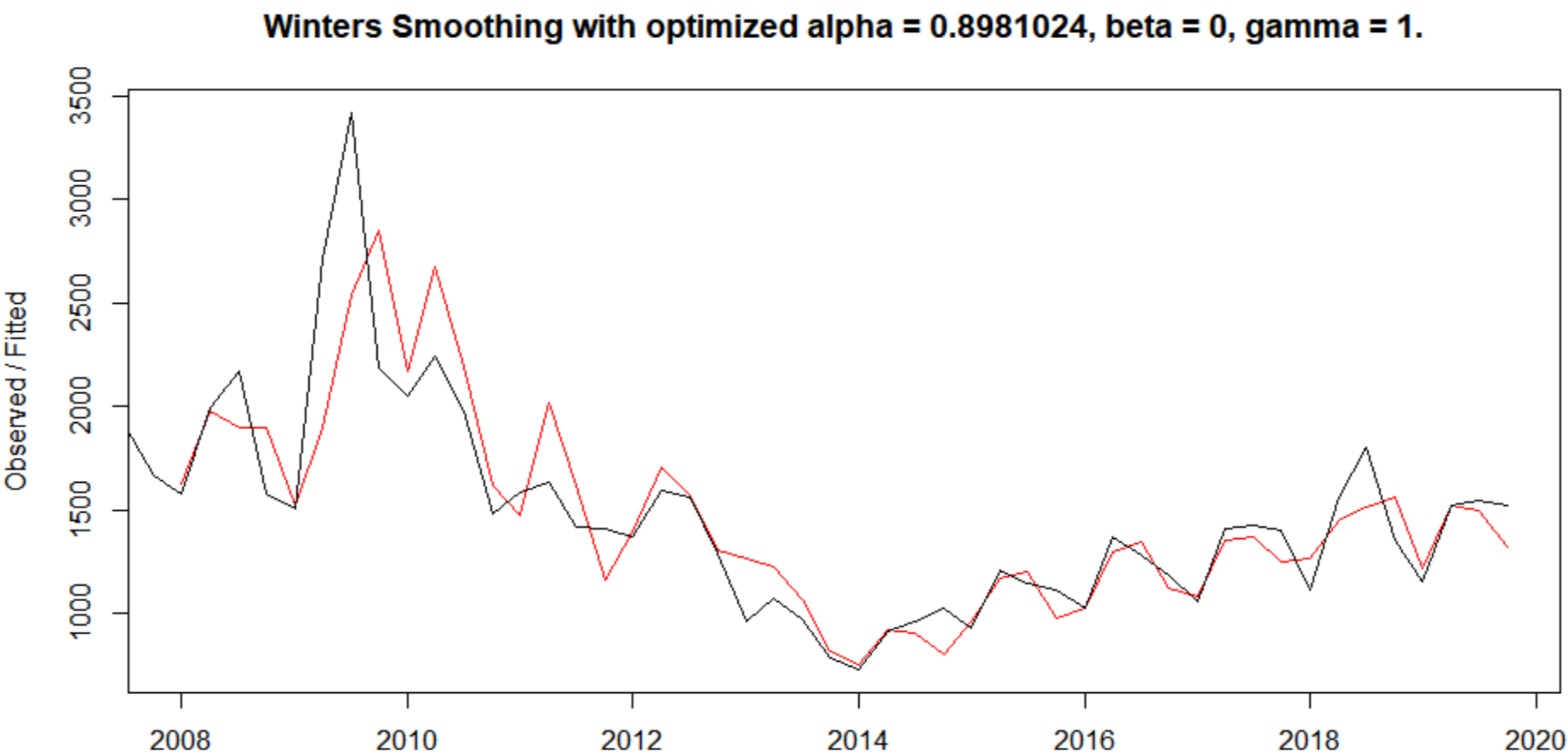


Winters Smoothing Method

```r
# Winter's method ------------------------------------------------------------
m.winters <- HoltWinters(flatsales.ts, seasonal = "multiplicative")

m.winters
## Optimal value of alpha = 0.8981024, beta = 0, gamma = 1.

plot(m.winters, main = "Winters Smoothing with optimized alpha = 0.8981024, beta = 0, gamma = 1.")

RMSE.winters <- round(sqrt(m.winters$SSE/nrow(m.winters$fitted)))
## Winters method has the lowest RMSE
```

```
> m.winters
Holt-Winters exponential smoothing with trend and multiplicative seasonal
 component.

Call:
HoltWinters(x = flatsales.ts, seasonal = "multiplicative")

Smoothing parameters:
 alpha: 0.8981024
 beta : 0
 gamma: 1

Coefficients:
           [,1]
a  1617.4851043
b    -0.1250000
s1    0.8314238
s2    1.0986646
s3    1.0853405
s4    0.9391122
```

```
68 plot(m.winters, main = "Winters Smoothing with optimized alpha = 0.8981024, beta = 0, gamma = 1.")
```

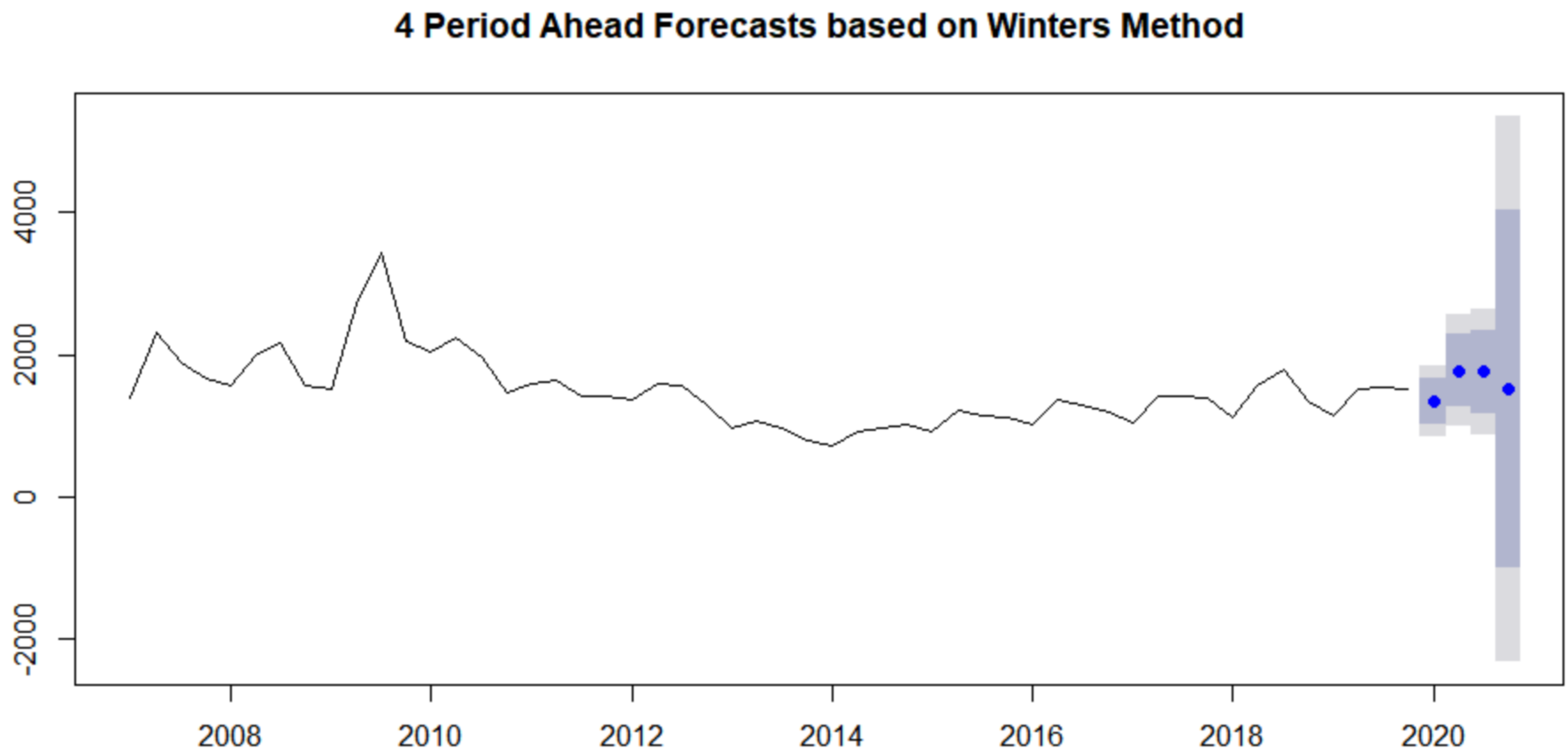


Forecast h periods ahead

```r
# Forecast 4 periods ahead ------------------------------------
m.winters.forecasts <- forecast(m.winters, h = 4)

m.winters.forecasts

plot(m.winters.forecasts, main = "4 Period Ahead Forecasts based on Winters Method")
```

```
> m.winters.forecasts
        Point Forecast     Lo 80    Hi 80      Lo 95    Hi 95
2020 Q1       1344.712 1016.6657 1672.758   843.0086 1846.415
2020 Q2       1776.799 1267.6998 2285.898   998.1990 2555.399
2020 Q3       1755.115 1171.8178 2338.412   863.0389 2647.191
2020 Q4       1518.530 -994.0947 4031.156 -2324.1981 5361.259
```

```r
78 plot(m.winters.forecasts, main = "4 Period Ahead Forecasts based on Winters Method")
```



Blue Points: Point Forecast, Dark Grey Interval: 80% C.I., Light Grey Interval: 95% C.I.

Other Forecasting Methods

- Causal models
  - From theory or domain knowledge
  - Involve Xs and t to forecast Y
- ARIMA models
  - Statistical analysis of autocorrelations
  - Requirement: Transform time series to a Stationary Time Series first before modelling the AR and MA components.
  - Stationary Time Series
    - Trend is constant over time.
    - Variance is constant over time.
    - Autocorrelation is constant over time.

Summary

- Forecasting is inherently extrapolation, unlike other models e.g. Linear Reg, CART, MARS, etc.
- MA models
- Exponential Smoothing models
  - SES
  - Holt
  - Winters
- Industry Practice
  - Model is just a baseline forecast
  - Adjust based on new data and new events.