Time Series and Moving Average Methods

Time Series Forecasting (Part 1)

Lecture Video Slides

1. What is a Time Series?
2. What is Moving Average?
3. How to describe a Time Series using Moving Averages?

Main 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/

What is a Time Series?

- A vector of data values measured across time.
- Minimal Dataset: Two columns
  - Time
  - Data Value
- 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 and Decomposition
    a. Trend
    b. Seasonal
    c. Random Error
2. Exponential Smoothing Methods
3. ARIMA method
4. Causal Methods
5. Machine Learning Methods

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

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

Only one column of data values

4 times per year

Data start in 2007, quarter 1

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
```

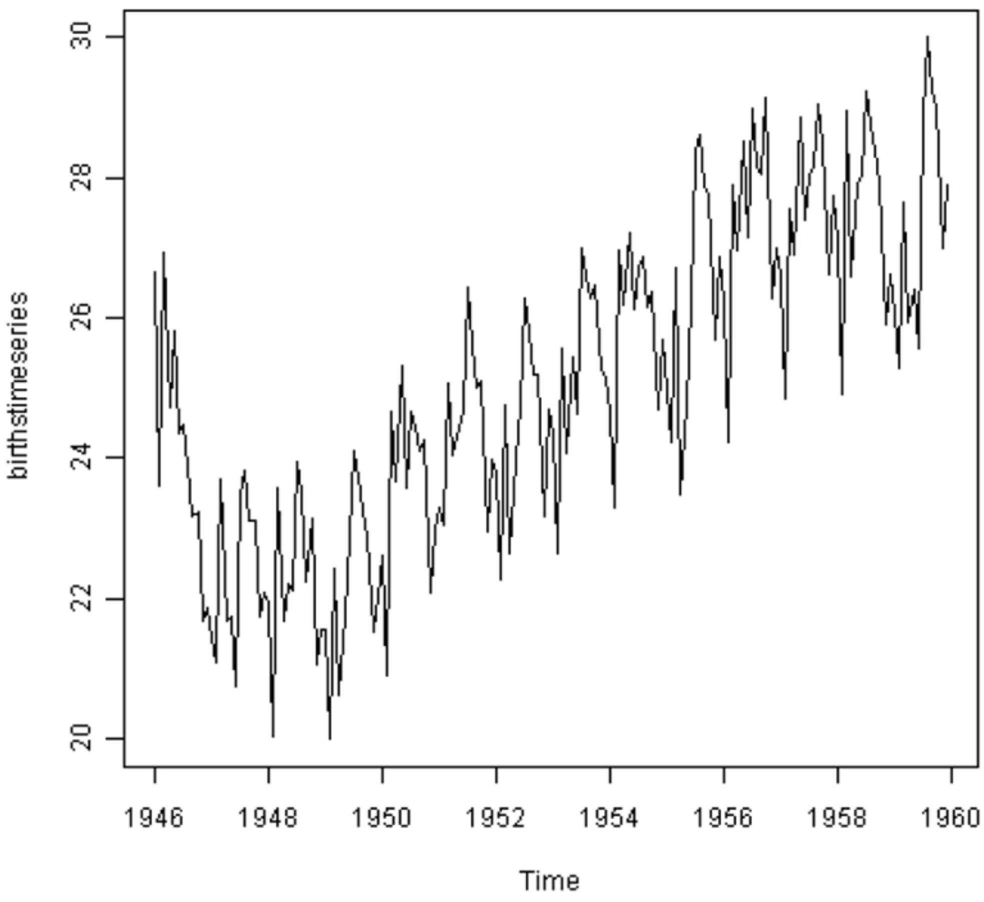
Seasonal Effects vs Cycles

- Seasonal effects occur year after year i.e. pattern repeats within a year, every year.
  - Examples:
    - Rainfall
    - Vacation hotel bookings
- Cycle repeats after many years.
  - Examples:
    - Flu epidemic
    - Bull/Bear market
- This course will only consider Trend and Seasonal Effects Estimation, not cycles.

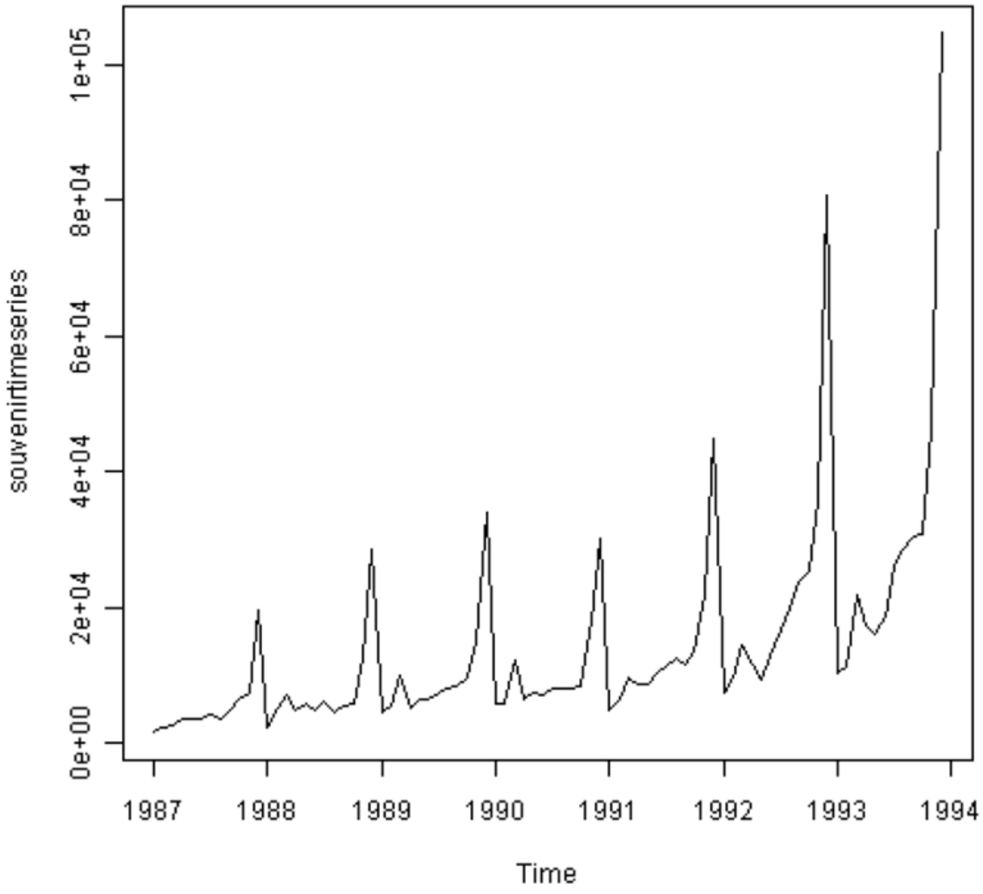
Assumes Additive or Multiplicative Time Series?

- Affects how to seasonally adjust (i.e. deseasonalize) the time series.
  - i.e. used to remove the effects of seasonality
  - Additive assumes seasonal component is added to trend.
  - Multiplicative assumes seasonal component is multiplied to trend.
- So as to more cleanly estimate the trend component.
- Most real world time series assume multiplicative.

Constant Fluctuations Over Time implies Additive Time Series



Non-constant Fluctuations Over Time implies Multiplicative Time Series



Simple Multiplicative Time Series Model

$$Y_t = T_t \times S_t \times \varepsilon_t$$

Two ways to use this equation:

1. Given historical time series data Y_t, estimate T_t and S_t via division.

2. Given estimated T_{t+h} and S_{t+h}, produce future forecast of Y_{t+h} by combining T_{t+h} and S_{t+h} via multiplication.

Simple Additive Time Series Model

$$Y_t = T_t + S_t + \varepsilon_t$$

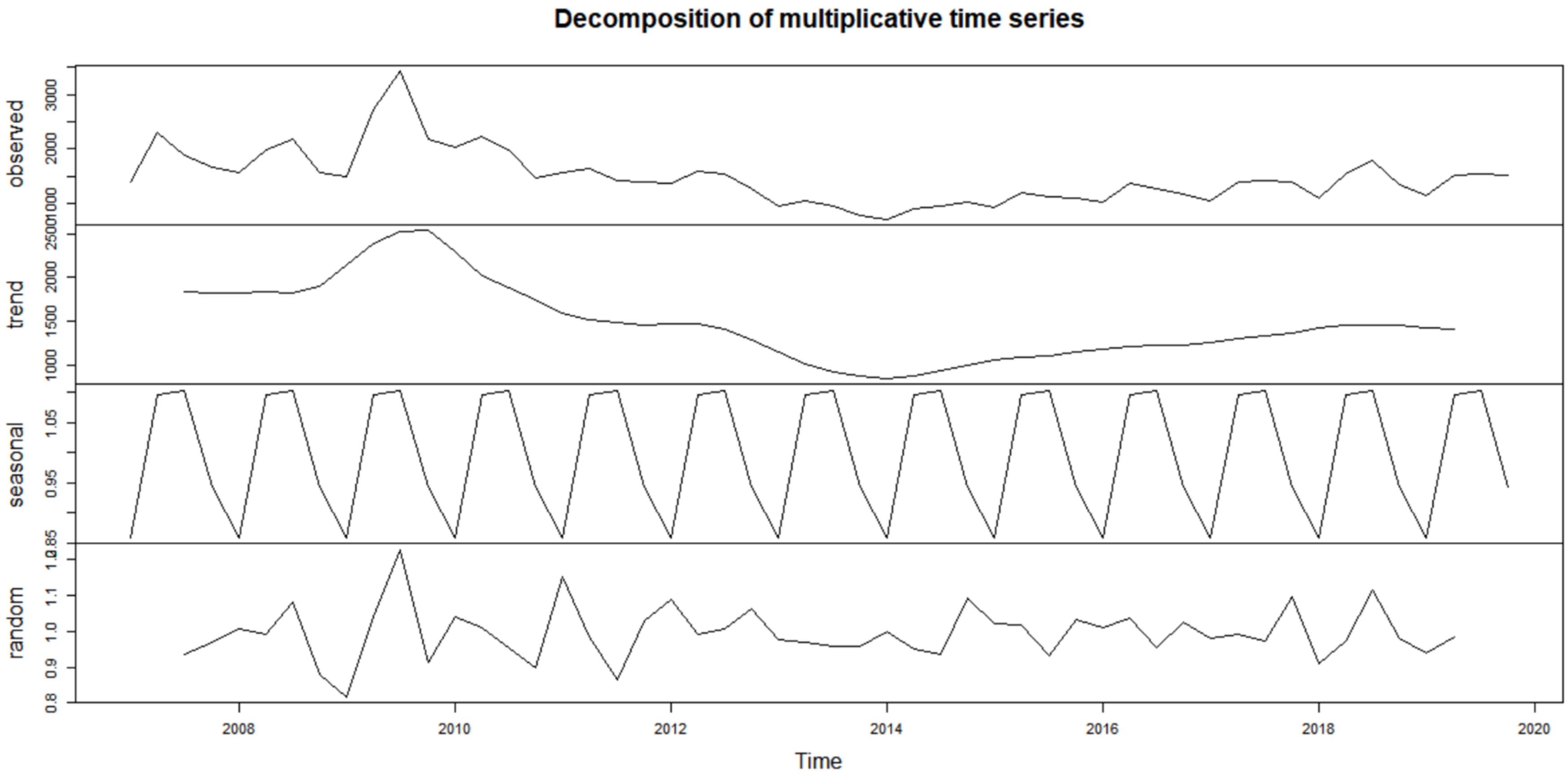
Two ways to use this equation:

1. Given historical time series data Y_t, estimate T_t and S_t via subtraction.

2. Given estimated T_{t+h} and S_{t+h}, produce future forecast of Y_{t+h} by combining T_{t+h} and S_{t+h} via addition.

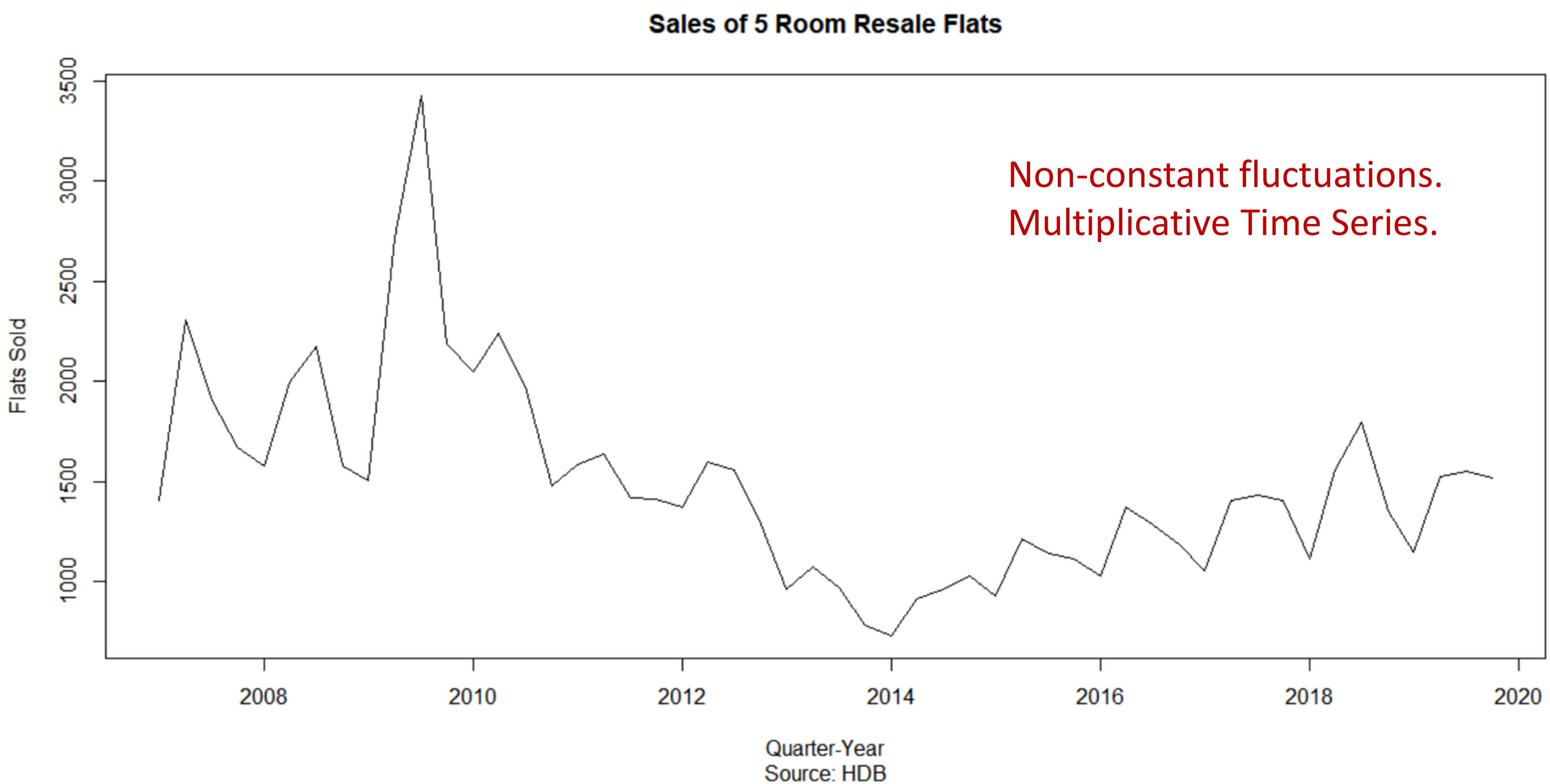
MA Based Decomposition of Time Series into Trend and Seasonal Components

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



Flat sales time series is additive or multiplicative?

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



Moving Average (MA)

- A moving average is the mean of the observations in the recent 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 Average is also a form of (simple) Smoothing Method.

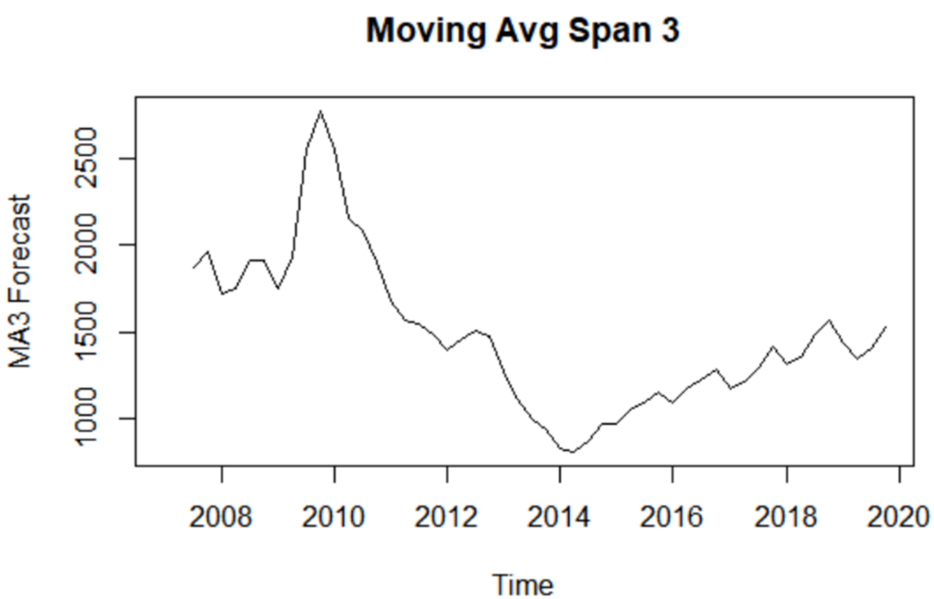
Setting the Span in Moving Average

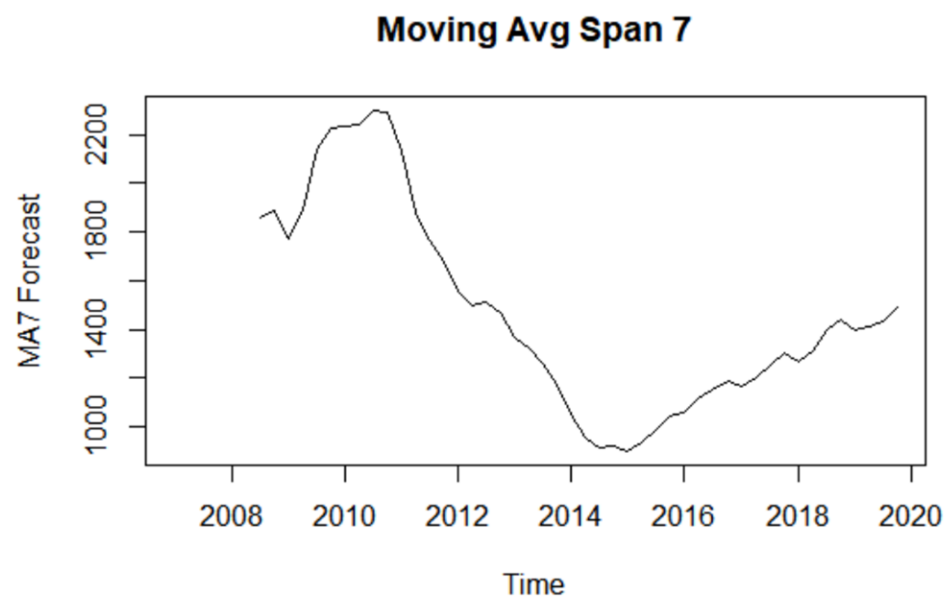
- Setting a span requires human opinion:
  - If you think fluctuations in the series are mainly due to random noise, use a relatively large span.
  - Otherwise, use a smaller span.

Moving Averages with Span = 3 vs 7
What's the differences?

```r
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 (a) smoother (less fluctuations) and (b) start later than MA3.

MA Smoothing 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.
- MA Based Decomposition assumes seasonal effects are constant year after year.
- In contrast, seasonal effects in exponential smoothing could change every year. i.e. non-constant.