Time series forcasting methods: Moving average

Dr Nikhil Chandra Sarkar, Data Scientist & Founder of Data Simulation Research Lab

2022-01-12

Let's get started with installation of package manager called pacman

```
# install.packages('packman')
```

Load the pacman package using the library() function:

```
library(pacman)
```

```
## Warning: package 'pacman' was built under R version 4.0.5
```

We can use p_load() to install the remaining packages we will need for the rest of this time series data analytics

```
p_load('tidyverse',      # data manipulation and visualization
       'ggplot2',        # data visualization
       'skimr',          # data description and summary
       'lubridate',      # working with date and time data
       'fpp2',           # working with time series data
       'zoo')            # working with time series data
```

Here I use US economic time series data for time series analytics

economics is a data frame with 574 rows and 6 variables.

Variables name and description.

date: Month of data collection; pce: personal consumption expenditures, in billions of dollars; pop: total population, in thousands; psavert: personal saving rate; uempmed: median duration of unemployment, in weeks; unemploy: number of unemployed in thousands.

```
data <- economics # import data from ggplot2
data %>%
  as_tibble() %>%
  print()
```

```
## # A tibble: 574 x 6
##    date         pce    pop psavert uempmed unemploy
##    <date>     <dbl>  <dbl>   <dbl>   <dbl>    <dbl>
##  1 1967-07-01  507. 198712    12.6     4.5     2944
##  2 1967-08-01  510. 198911    12.6     4.7     2945
##  3 1967-09-01  516. 199113    11.9     4.6     2958
##  4 1967-10-01  512. 199311    12.9     4.9     3143
##  5 1967-11-01  517. 199498    12.8     4.7     3066
##  6 1967-12-01  525. 199657    11.8     4.8     3018
##  7 1968-01-01  531. 199808    11.7     5.1     2878
##  8 1968-02-01  534. 199920    12.3     4.5     3001
##  9 1968-03-01  544. 200056    11.7     4.1     2877
## 10 1968-04-01  544  200208    12.3     4.6     2709
## # ... with 564 more rows
```

Get a summary of the data to help locate any potential data quality issues

```
skim(data)
```

Table 1: Data summary

Name	data
Number of rows	574
Number of columns	6
Column type frequency:	
Date	1
numeric	5
Group variables	None

Variable type: Date

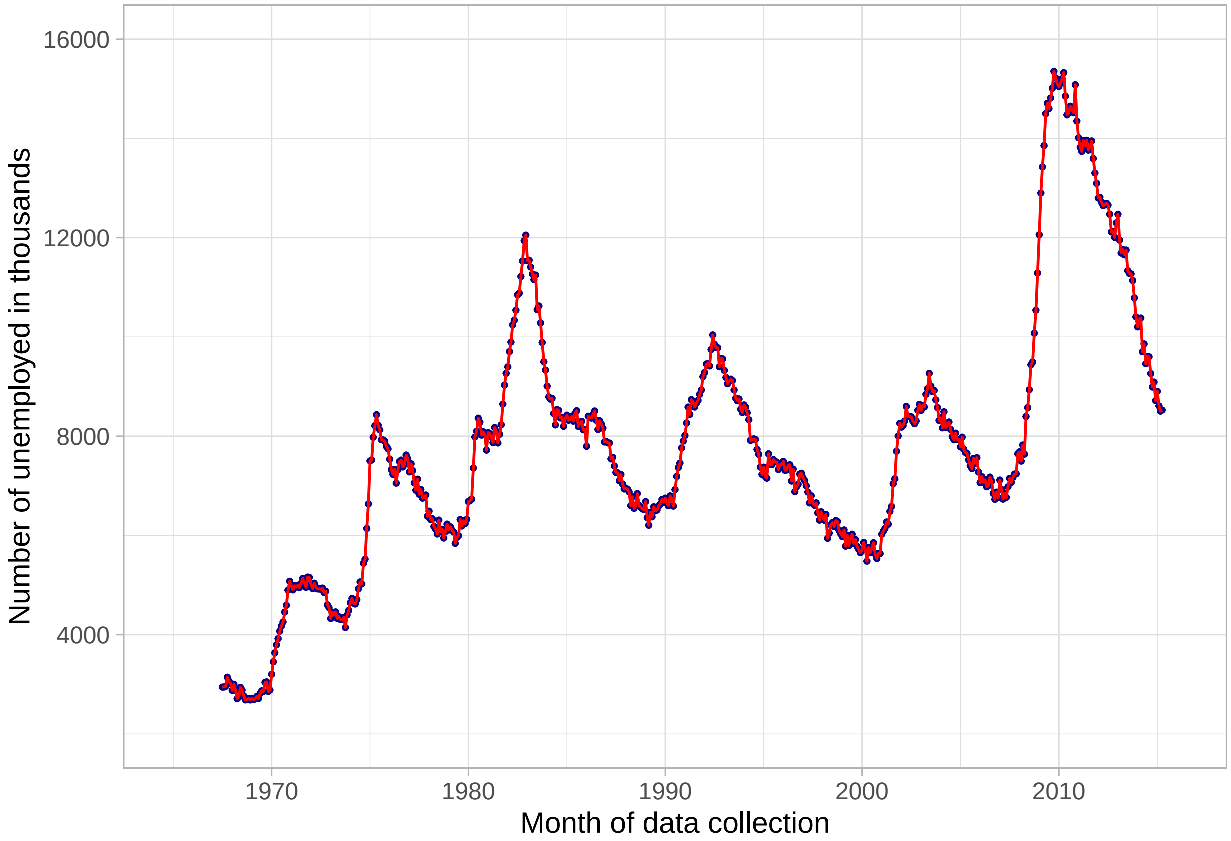
skim_variable	n_missing	complete_rate	min	max	median	n_unique
date	0	1	1967-07-01	2015-04-01	1991-05-16	574

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p
pce	0	1	4820.09	3556.80	506.7	1578.3	3936.85	7626.33	1219
pop	0	1	257159.65	36682.40	198712.0	224896.0	253060.00	290290.75	32040
psavert	0	1	8.57	2.96	2.2	6.4	8.40	11.10	1
uempmed	0	1	8.61	4.11	4.0	6.0	7.50	9.10	2
unemploy	0	1	7771.31	2641.96	2685.0	6284.0	7494.00	8685.50	1535

Time series graph for personal saving rate

```
ggplot(data = data, aes(x = date, y = unemploy)) +
      geom_point(color='dark blue', size=0.6) +
      geom_line(color = 'red') +
      #geom_line(color = "goldenrod") +
      theme_light() +
     coord_cartesian(xlim = c(date("1965-01-01"), date("2016-01-01")), ylim = c(2000, 16000)) +
      xlab('Month of data collection') +
      ylab('Number of unemployed in thousands')
```



Centered Moving Averages

The most straightforward time series data analytic method is a simple moving average. For this method, we choose a number of neighborhood points and average them to estimate the trend. When calculating a simple moving average, it is beneficial to use an odd number of points so that the calculation is symmetric.

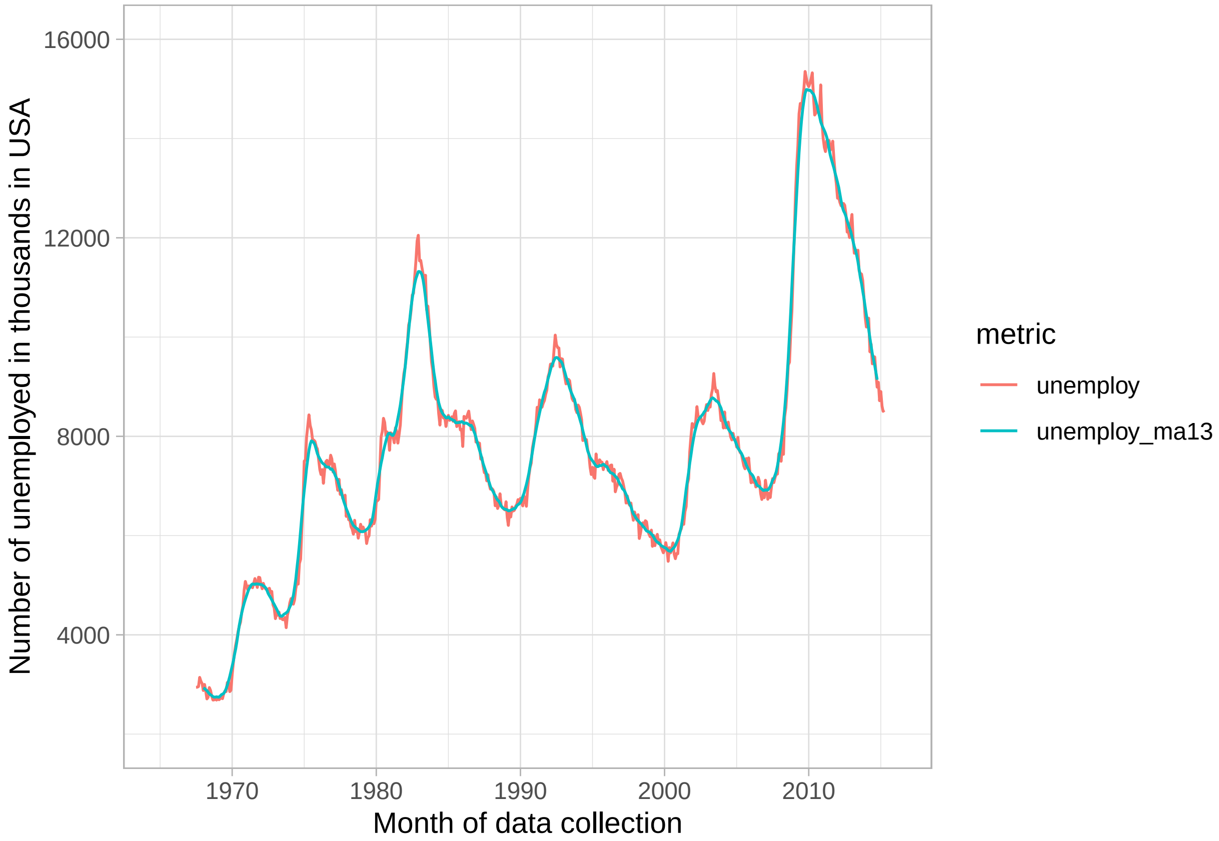
Here first, I compute the 13 month moving average values and add this data back to the data frame.

```
unemployed <- data %>%
  select(date, unemploy) %>%
  mutate(unemploy_ma13 = rollmean(unemploy, k = 13, fill = NA))
```

Now we can go ahead and plot these values and compare the actual data to the different moving average smoothers.

```
unemployed %>%
  gather(metric, value, unemploy:unemploy_ma13) %>%
  ggplot(aes(date, value, color = metric)) +
  geom_line() +
  coord_cartesian(xlim = c(date("1965-01-01"), date("2016-01-01")), ylim = c(2000, 16000)) +
      theme_light() +
      xlab('Month of data collection') +
      ylab('Number of unemployed in thousands in USA')
```

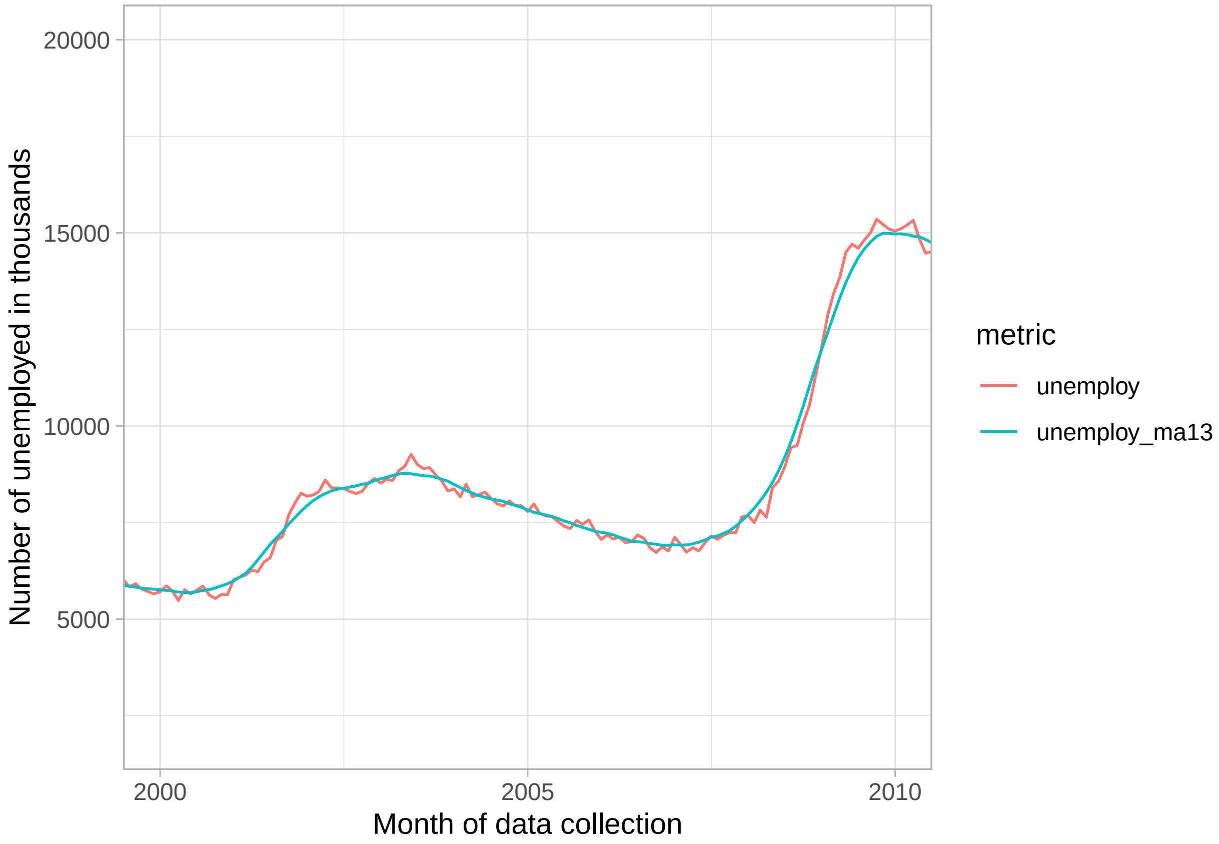
```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```



We can see this by zooming into the 2000-2015 time range:

```
unemployed %>%
  gather(metric, value, unemploy:unemploy_ma13) %>%
  ggplot(aes(date, value, color = metric)) +
  geom_line() +
  coord_cartesian(xlim = c(date("2000-01-01"), date("2010-01-01")), ylim = c(2000, 20000)) +
      theme_light() +
      xlab('Month of data collection') +
      ylab('Number of unemployed in thousands')
```

```
## Warning: Removed 12 row(s) containing missing values (geom_path).
```



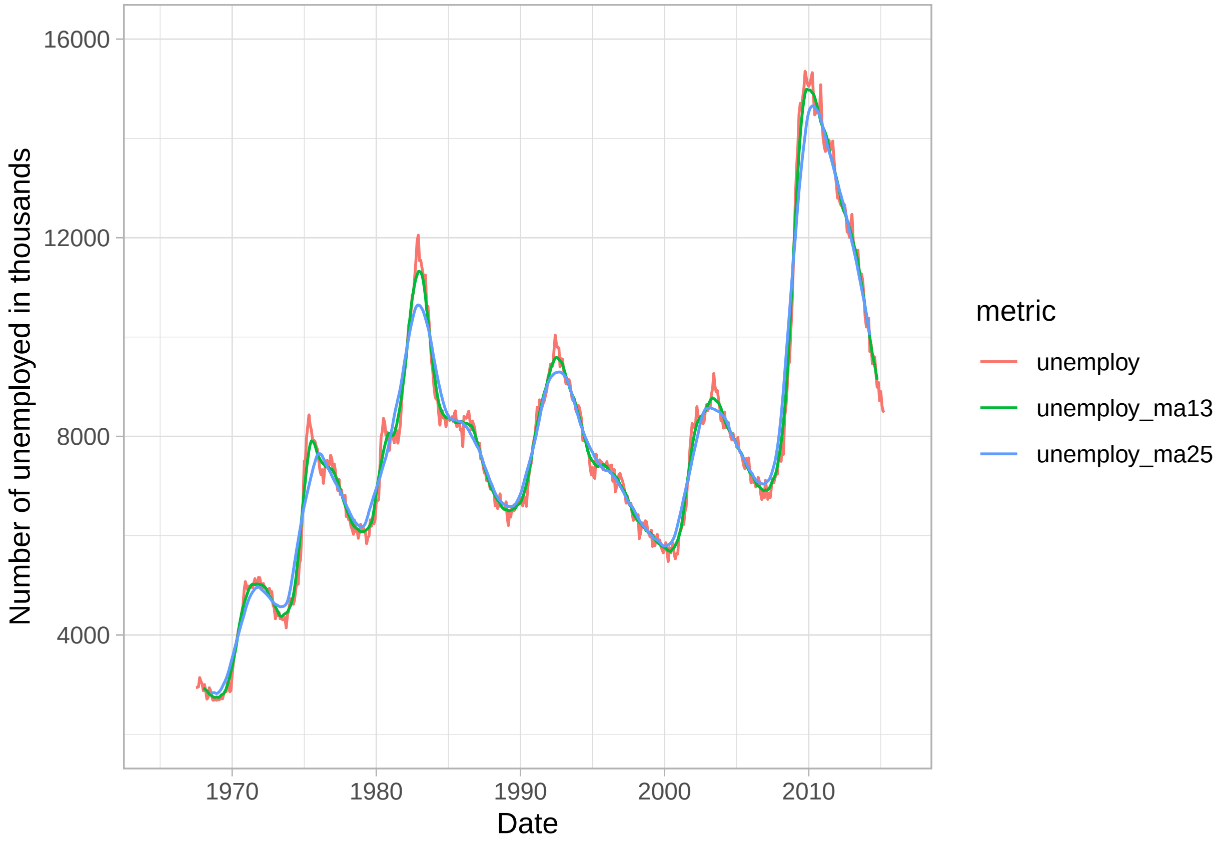
Here, I compute the 13 and 25 month moving average values and add this data back to the data frame.

```
unemployed <- data %>%
  select(date, unemploy) %>%
  mutate(unemploy_ma13 = rollmean(unemploy, k = 13, fill = NA),
         unemploy_ma25 = rollmean(unemploy, k = 25, fill = NA)
        )
```

Now we can go ahead and plot these values and compare the actual data to the different moving average smoothers.

```
unemployed %>%
  gather(metric, value, unemploy:unemploy_ma25) %>%
  ggplot(aes(date, value, color = metric)) +
  geom_line() +
  coord_cartesian(xlim = c(date("1965-01-01"), date("2016-01-01")), ylim = c(2000, 16000)) +
      theme_light() +
      xlab('Date') +
      ylab('Number of unemployed in thousands')
```

```
## Warning: Removed 36 row(s) containing missing values (geom_path).
```



## To understand how these different moving averages compare we can compute the mean absolute percentage error (MAPE).This error rate will increase as you choose a larger k to average over.

```
unemployed %>%
  gather(metric, value, unemploy_ma13:unemploy_ma25) %>%
  group_by(metric) %>%
  summarise(MAPE = mean(abs((unemploy - value)/unemploy), na.rm = TRUE))
```

```
## # A tibble: 2 x 2
##   metric          MAPE
##   <chr>          <dbl>
## 1 unemploy_ma13 0.0191
## 2 unemploy_ma25 0.0363
```

Trailing Moving Average for Forecasting

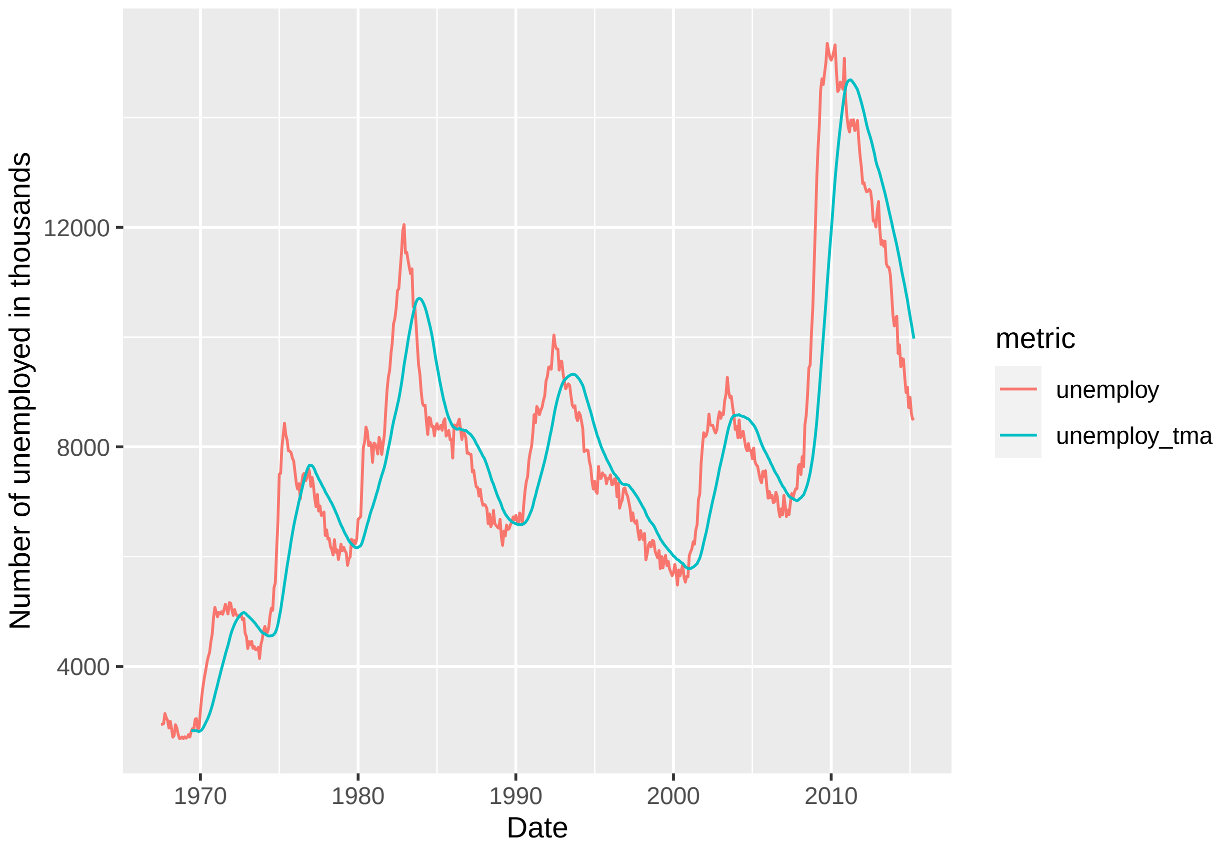
```r
unemployed_tma <- data %>%
  select(date, unemploy) %>%
  mutate(unemploy_tma = rollmean(unemploy, k = 24, fill = NA, align = "right"))

tail(unemployed_tma, 5)
```

```
## # A tibble: 5 x 3
##   date       unemploy unemploy_tma
##   <date>        <dbl>        <dbl>
## 1 2014-12-01     8717       10529.
## 2 2015-01-01     8903       10381.
## 3 2015-02-01     8610       10242.
## 4 2015-03-01     8504       10109.
## 5 2015-04-01     8526        9974.
```

We can visualize how the 24-month trailing moving average predicts future number of unemployed in thousands with the following plot.

```r
unemployed_tma %>%
  gather(metric, value, -date) %>%
  ggplot(aes(date, value, color = metric)) +
  geom_line()+
  xlab('Date') +
  ylab('Number of unemployed in thousands')
```

```
## Warning: Removed 23 row(s) containing missing values (geom_path).
```



# Here, I compute the mean absolute percentage error (MAPE)

```
unemployed_tma %>%
  gather(metric, value, unemploy_tma) %>%
  group_by(metric) %>%
  summarise(MAPE = mean(abs((unemploy - value)/unemploy), na.rm = TRUE))
```

```
## # A tibble: 1 x 2
##   metric        MAPE
##   <chr>        <dbl>
## 1 unemploy_tma 0.103
```

Moving Averages of Moving Averages

```
economics %>%
  mutate(ma4 = ma(unemploy, order = 4, centre = TRUE)) %>%
  head(5)
```

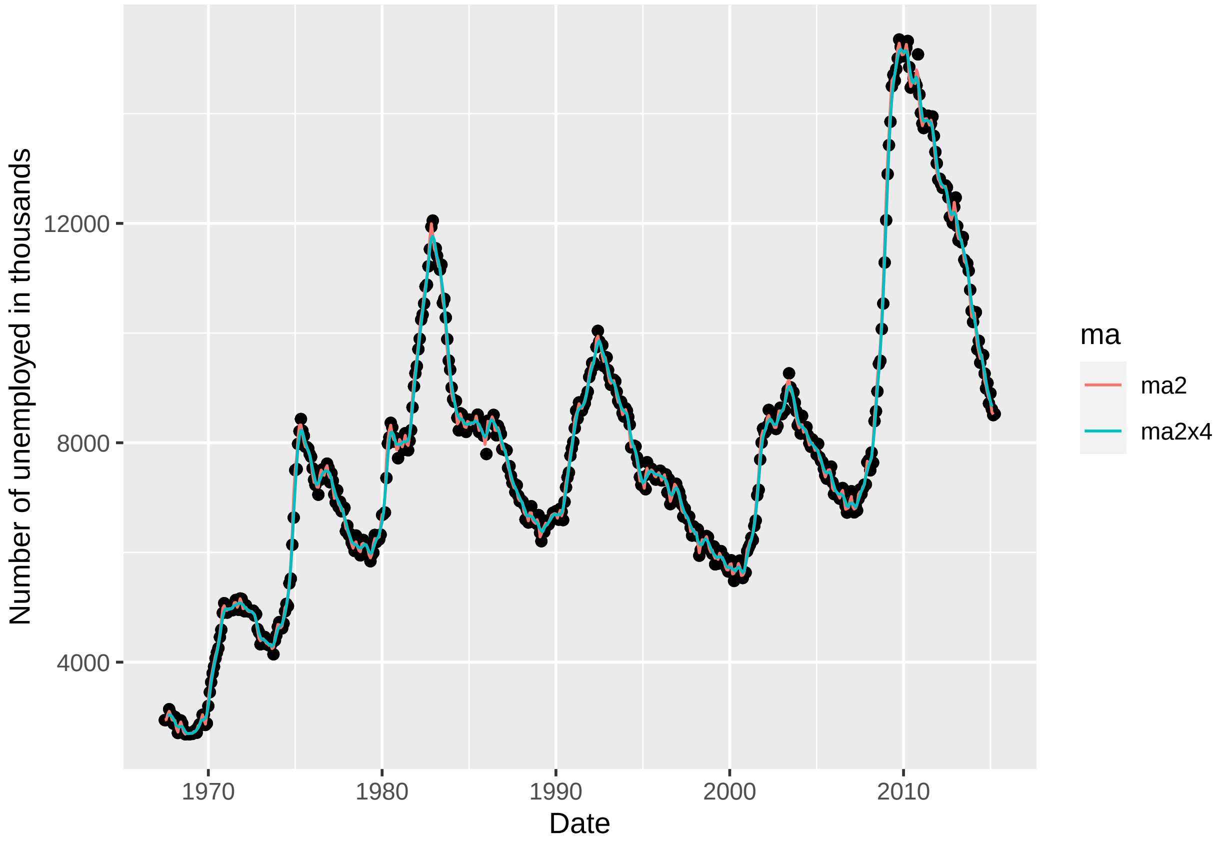
```
## # A tibble: 5 x 7
##   date         pce    pop psavert uempmed unemploy   ma4
##   <date>     <dbl>  <dbl>   <dbl>   <dbl>    <dbl> <dbl>
## 1 1967-07-01  507. 198712    12.6     4.5     2944   NA
## 2 1967-08-01  510. 198911    12.6     4.7     2945   NA
## 3 1967-09-01  516. 199113    11.9     4.6     2958 3013.
## 4 1967-10-01  512. 199311    12.9     4.9     3143 3037.
## 5 1967-11-01  517. 199498    12.8     4.7     3066 3036.
```

To compare this moving average to a regular moving average we can plot the two outputs:

```
# compute 2 and 2x4 moving averages
 economics %>%
  mutate(ma2 = rollmean(unemploy, k = 2, fill = NA),
         ma2x4 = ma(unemploy, order = 4, centre = TRUE)) %>%
  gather(ma, value, ma2:ma2x4) %>%
  ggplot(aes(x = date)) +
  geom_point(aes(y = unemploy)) +
  geom_line(aes(y = value, color = ma))+
  xlab('Date') +
  ylab('Number of unemployed in thousands')
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
## Warning: Removed 5 row(s) containing missing values (geom_path).
```


Number of unemployed in thousands
Date
ma
ma2
ma2x4