post02-xinran-liu

Xinran Liu

December 1, 2017

Time Series Analysis in R

1. Introduction and Motivation

- Throughout the course we have dealt with many different sets of data (especially the NBA players' statistics; I actually expanded my knowledge on that because of this course), but in real life, we would be highly likely to encounter data sets that are listed in a time order such as annual birth rate and/or death rate, monthly wages in a country, daily temperature, etc. It would be important for analysts to read them, and analyze them with consideration of time.
- In this post I will try to present some basic ideas of time series analysis, and hopefully it can bring some insights to this topic! Here we go!

2. Time Series Data

Let's first look at how we can take our data set and transform into a time series object (the following data comes from Time series data library-monthly milk production):

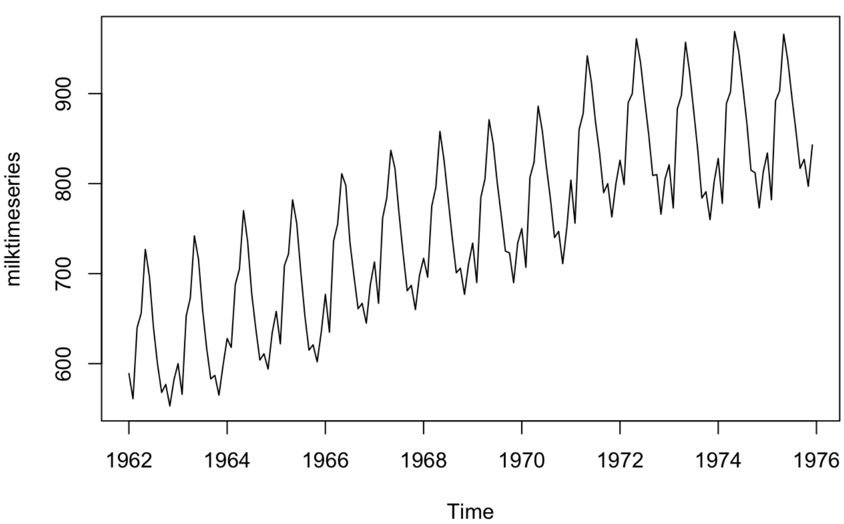
```
# read my target data
milk <- read.csv('monthly-milk-production-pounds-p.csv')
# change the name of the second column to shorter
names(milk)[2] <- "Monthly.milk.production"
# create time series object (notice the start, end, and frequency arg)
milktimeseries <- ts(milk$Monthly.milk.production, start = c(1962, 1),
                     end = c(1975, 12), frequency = 12)
milktimeseries
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1962 589 561 640 656 727 697 640 599 568 577 553 582
## 1963 600 566 653 673 742 716 660 617 583 587 565 598
## 1964 628 618 688 705 770 736 678 639 604 611 594 634
## 1965 658 622 709 722 782 756 702 653 615 621 602 635
## 1966 677 635 736 755 811 798 735 697 661 667 645 688
## 1967 713 667 762 784 837 817 767 722 681 687 660 698
## 1968 717 696 775 796 858 826 783 740 701 706 677 711
## 1969 734 690 785 805 871 845 801 764 725 723 690 734
## 1970 750 707 807 824 886 859 819 783 740 747 711 751
## 1971 804 756 860 878 942 913 869 834 790 800 763 800
## 1972 826 799 890 900 961 935 894 855 809 810 766 805
## 1973 821 773 883 898 957 924 881 837 784 791 760 802
## 1974 828 778 889 902 969 947 908 867 815 812 773 813
## 1975 834 782 892 903 966 937 896 858 817 827 797 843
```

Here we have transformed our data into time series using the function ts(). Notice we have several arguments here in the function: start, end and frequency. start and end will determine which data to collect, and the argument frequency allows us to manipulate data that are collected at intervals less than one year. If data are collected monthly, frequency will be 12; if they are collected quarterly, frequency will be 4.

Next, let's plot time series! We will be using plot.ts() function:

```
# plot milk production time series
plot.ts(milktimeseries)
```



From the plot we could see that for each year there is a peak around April and May, and the production is lowest in winter every year. Moreover, this time series could be described using an additive model since fluctuations seem to be constant over time.

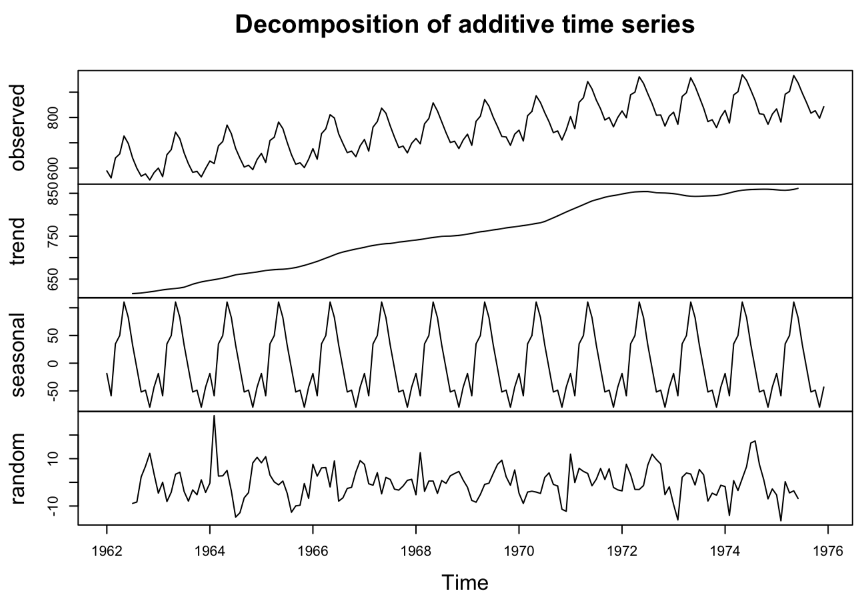
3. Analysis

The above time series appear to have a seasonal variation, which makes this data set to be a seasonal time series. In order to better understand the pattern of the data, we could use function decompose(). It will separate the time series into its constituent components, which are a trend component, an irregular component, and a seasonal component. If the time series is not seasonal, it will not have the seasonal component. (Cited from Time Series 0.2 documentation.) Now try to analyze our ata!

```
# decompose
decomposemilk <- decompose(milktimeseries)
# take a look at seasonal component
decomposemilk$seasonal
```

```
##            Jan        Feb        Mar        Apr        May        Jun
## 1962 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
## 1963 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
## 1964 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
## 1965 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
## 1966 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
## 1967 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
## 1968 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
## 1969 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
## 1970 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
## 1971 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
## 1972 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
## 1973 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
## 1974 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
## 1975 -18.669605 -58.858707  34.897703  49.913729 110.407318  82.404113
##            Jul        Aug        Sep        Oct        Nov        Dec
## 1962  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
## 1963  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
## 1964  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
## 1965  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
## 1966  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
## 1967  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
## 1968  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
## 1969  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
## 1970  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
## 1971  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
## 1972  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
## 1973  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
## 1974  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
## 1975  32.698985  -9.563835 -51.926015 -48.679220 -79.467682 -43.156784
```

```
# plot decomposed components
plot(decomposemilk)
```



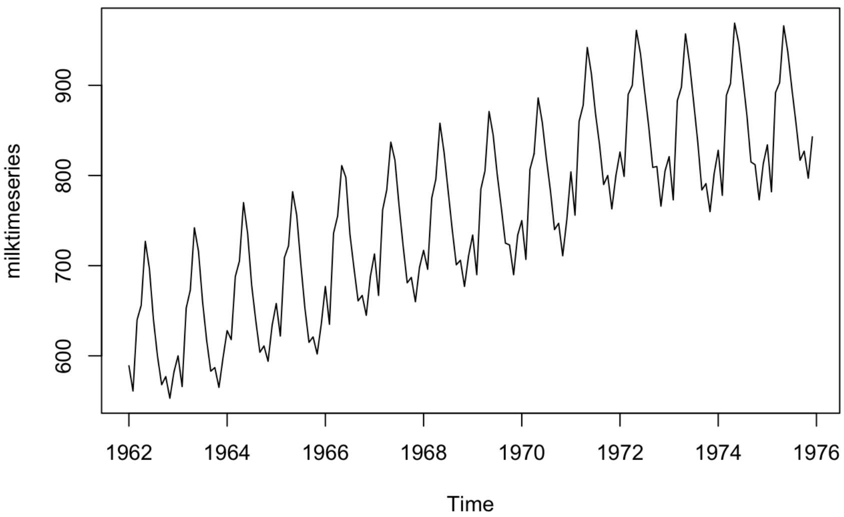
We can see in the plot the second row actually shows the general trend of the data set; the third row presents the seasonal changes.

For seasonal time series, we could even use functions in package forecast to deseasonalize it to model the data without the seasonal effects. Let's take a look!

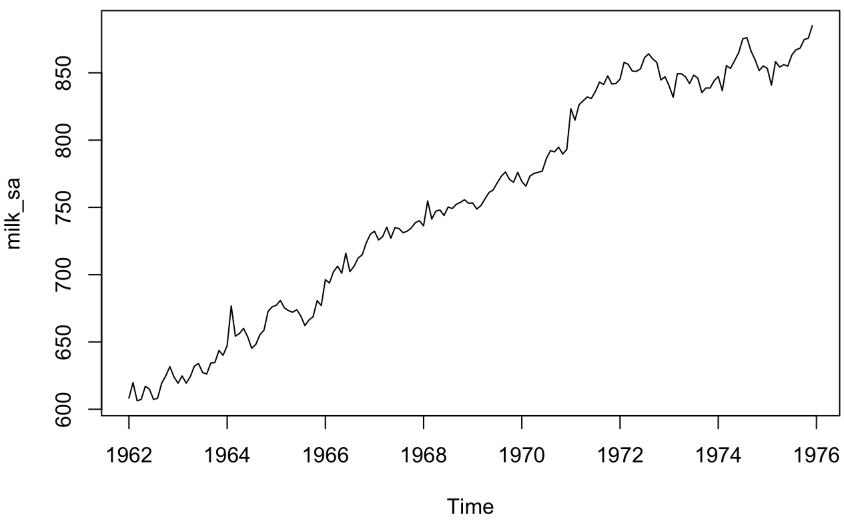
```
# load package
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.4.2
```

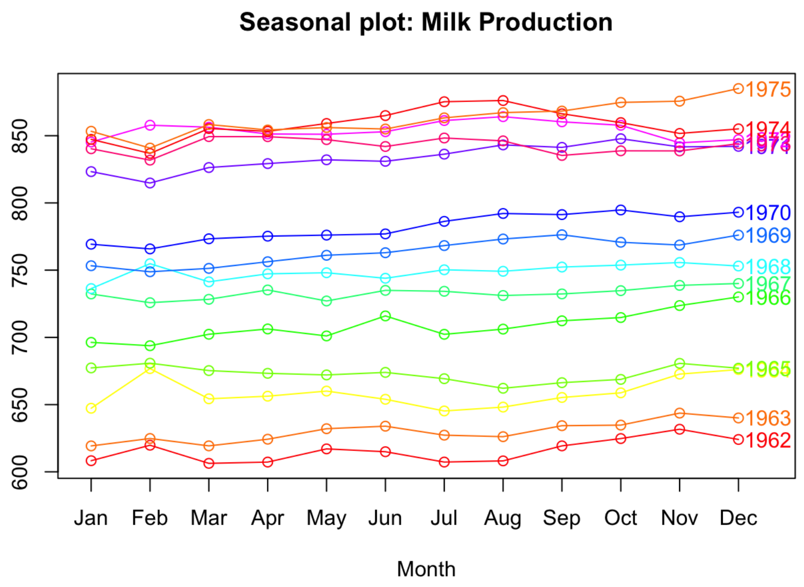
```r
# use stl to decompose
milk_stl <- stl(milktimeseries, "periodic")
# de-seasonalize
milk_sa <- seasadj(milk_stl)
# original series
plot(milktimeseries, type="l")
```




```r
# seasonal adjusted
plot(milk_sa, type="l")
```




```r
# de-seasonalize plot
seasonplot(milk_sa, 12, col = rainbow(12), year.labels=TRUE, main="Seasonal plot: Milk Production")
```



We could see that in seasonal plot the data are arranged more clearly, and we could see the pattern of data from more perspectives.

4. End for now

When I was researching for this post, I could see that there are so many data sets collected and arranged with respect to time. It is so important in real life to be capable of analyzing time series, so some basic functions and analysis presented in this post would be crucial for all future data analysts~ Hope you have learned something from it and enjoyed reading it!

tbc……

References

Coghlan, Avril. *Using R for Time Series Analysis*. Time Series 0.2 documentation, 2010, https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html#time-series-analysis. Accessed on Dec. 1st.

Hyndman, R.J. *"Time Series Data Library"*, http://data.is/TSDLdemo . Accessed on Dec. 1st, 2017.

Kabacoff, Robert I. *Time Series and Forecasting*. Quick-R, https://www.statmethods.net/advstats/timeseries.html. Accessed on Dec. 1st.

Prabhakaran, Selva. *Time Series Analysis*. r-statistics.co, http://r-statistics.co/Time-Series-Analysis-With-R.html. Accessed on Dec. 1st, 2017.

Srivastava, Tavish. *A Complete Tutorial on Time Series Modeling in R*. Analytics Vidhya, Dec. 16, 2015, https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/. Accessed on Dec. 1st, 2017.

Walters, Troy. *Time Series Analysis in R Part 1: The Time Series Object*. R-bloggers, Sep. 19, 2017, https://www.r-bloggers.com/time-series-analysis-in-r-part-1-the-time-series-object/. Accessed on Dec. 1st, 2017.

Zucchini, Walter & Nenadic Oleg. *Time Series Analysis with R - Part I*. http://www.statoek.wiso.uni-goettingen.de/veranstaltungen/zeitreihen/sommer03/ts_r_intro.pdf. Accessed on Dec. 1st, 2017.