Time series data analysis using package ‘forecast’ & ’ggplot

Eriko Funasato

December 3, 2017

Introduction

In this course, we have learnt about how to analyze the data given, of which is about the event happened in the past. Therefore I become curious about predicting on the things will happen in the future based on data using r. The package “forecast” in r can provide methods and tools to analyze the forecast of the time series dataset based on some kinds of modeling. Time series is series of data listed in time order with some period. For example, the weather in a year, the stock flow in a period, and the earning of a shop are all time series. To know more about the time series, see wikipedia.

In this post, I would like to show how to use the function in the package ‘forecast’to organize, forecast, and display based on a time series dataset.

First, install and load the package ’forecast’:

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.4.3
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 3.4.3
```

Then, we will use the datasets “nottem” already built in r. The data set is about ‘Average Monthly Temperatures at Nottingham, 1920-1939’, which is a typical time series data. And I would like to rename the ‘nottem’ data as “dat”.

```
dat <- data.frame(time=c(time(nottem)),tem=c(nottem))
```

Organize and display the dataset

- To make the data easy to analyze by ‘forecast’, we have to store the data as ‘ts’ object in r. About ts, see time series.

```
head(nottem)
```

```
##       Jan  Feb  Mar  Apr  May  Jun
## 1920 40.6 40.8 44.4 46.7 54.1 58.5
```

```
head(dat)
```

```
##       time  tem
## 1 1920.000 40.6
## 2 1920.083 40.8
## 3 1920.167 44.4
## 4 1920.250 46.7
## 5 1920.333 54.1
## 6 1920.417 58.5
```

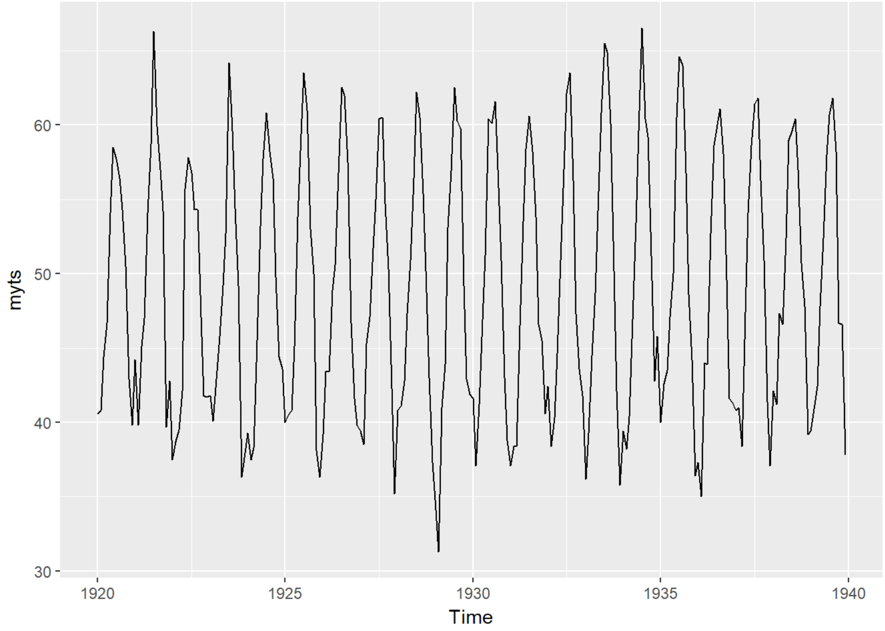
```
myts <- ts(dat[,2],start = c(1920.000,1),frequency = 12)
```

By checking the class of the data, we know that it is already in ts form.

Autoplot

- Then, we will like to display or graph the dataset, using the function ‘autoplot’. With the optional argument ‘facets’, we can produce time plot with or without facetting.

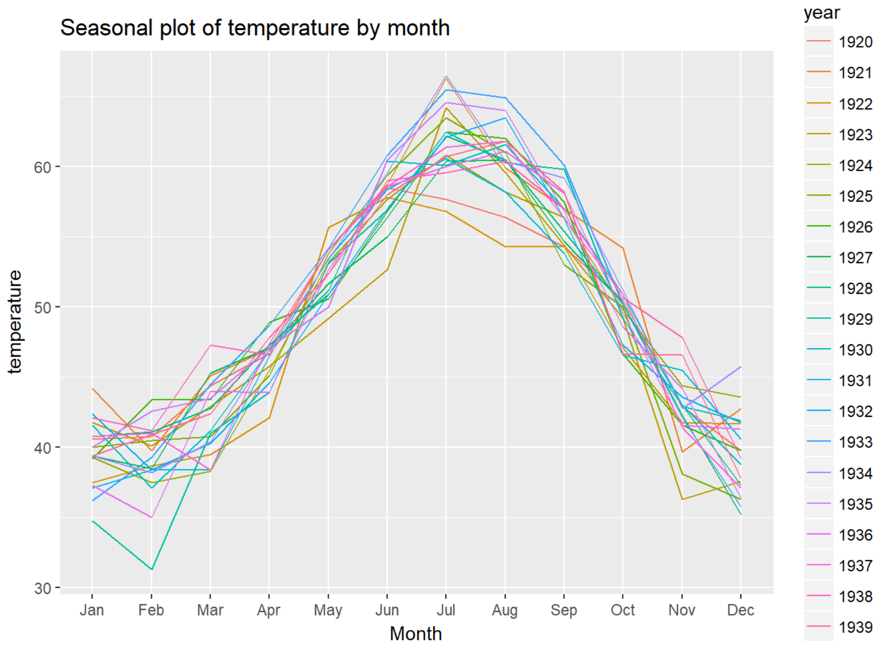
```
# with no facetting
autoplot(myts,facets=FALSE)
```



Seasonal plot: ggseasonplot

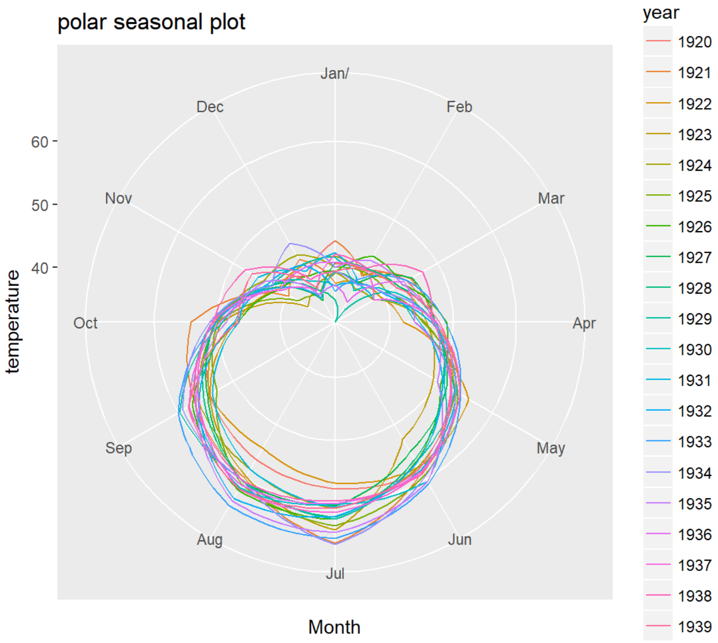
- Use ggseasonplot to plot the graph again, with the optional argument ‘polar’, we can create a circular axis graph rather than horizontal axis

```
ggseasonplot(myts)+ylab('temperature')+ggtitle("Seasonal plot of temperature by month")
```



- By making the optional argument “polar” to be TRUE, to get a circular graph

```
# polar = TRUE
ggseasonplot(myts, polar = TRUE)+ylab("temperature")+ggtitle("polar seasonal plot")
```



- From the graphs above, we can conclude that the trend of the temperature data is seasonal.

White noise series

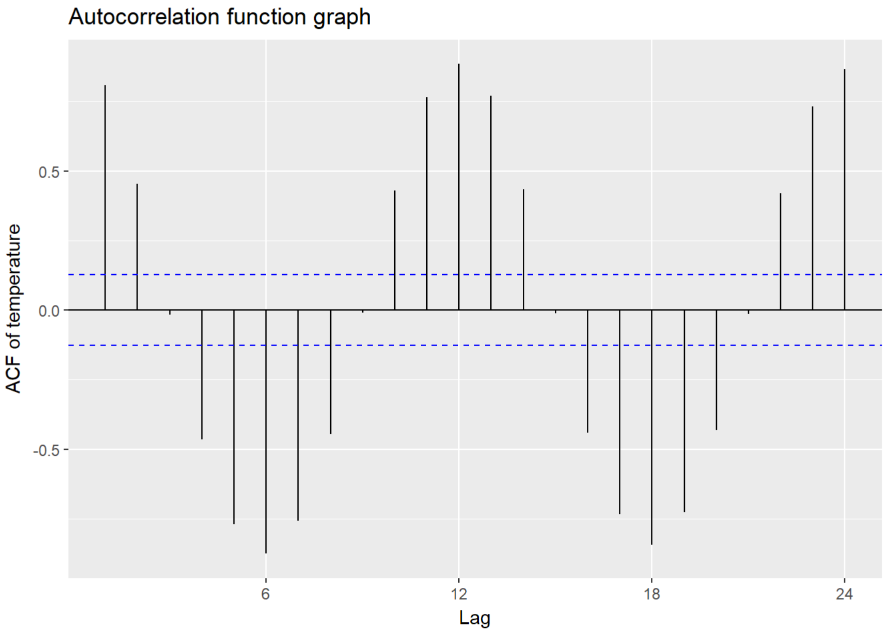
- Here, I would like to introduce that “white noise series” (wikipedia) refers to the purely random series, which means there is no trend, no tendency in the data.It also means that the data is not usable as time series. To check if it is a whitenoise series, we use box test

```
Box.test(myts)
```

```
##
##  Box-Pierce test
##
## data:  myts
## X-squared = 156.57, df = 1, p-value < 2.2e-16
```

- From the result of the p-value, we get to know that we succeed to reject, which means our ts od temperature data is not white noise series.
- Also, we can plot a Acf FUnction graph, autocorrelation function, to see if our correlation is high enough as a time series.

```
ggAcf(myts)+ylab("ACF of temperature")+ggtitle("Autocorrelation function graph")
```



- From the result, we can see that the blue line is the autocorrelation level, and most of our data exceed the blue line, which means it is not a white noise series.

Forecasting the data, and filling the prediction on the graph

- Now, let's try to do the weatherforecast of the future temperature using the package "forecast".
- To introduce the ARIMA model, Autoregressive integrated moving average, check wikipedia. ARIMA model is the model used to time series so that we can predict future points in the series.
- First, we check if myts is stationary

```
adf.test(myts,alternative="stationary",k= trunc((length(myts)-1)^(1/3)))
```

```
## Warning in adf.test(myts, alternative = "stationary", k =
## trunc((length(myts) - : p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  myts
## Dickey-Fuller = -12.998, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

- By the small p-value we got, we can know that our dataset is stationary, now we can apply the arima model:

```
# arima model
fit <- arima(myts, seasonal = list(order=c(1,1,1),period = 12))
accuracy(fit)
```

```
##                     ME     RMSE      MAE         MPE     MAPE      MASE
## Training set 0.1026845 2.303841 1.768481 -0.09236835 3.800736 0.4094024
##                   ACF1
## Training set 0.2646953
```

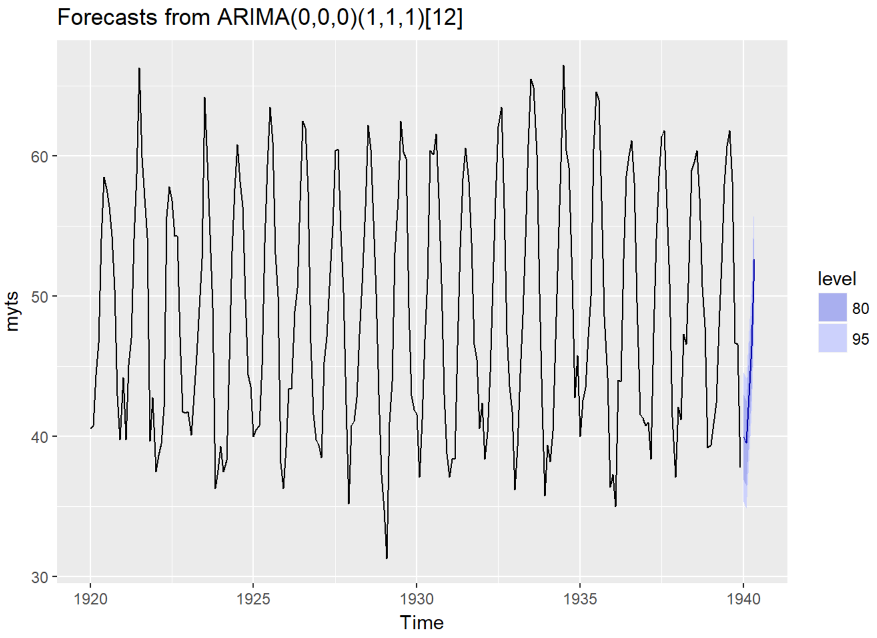
- Now, if we want to predict next five observation of temperature,use forecast(,5)

```
forecast(fit, 5)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1940       39.96709 36.93793 42.99626 35.33438 44.59980
## Feb 1940       39.52311 36.49394 42.55228 34.89040 44.15582
## Mar 1940       42.89968 39.87052 45.92885 38.26697 47.53239
## Apr 1940       46.44401 43.41485 49.47318 41.81130 51.07672
## May 1940       52.61836 49.58920 55.64753 47.98565 57.25107
```

- Then display the prediction on the autoplot

```
autoplot(forecast(fit, 5))
```



- Obviously, the result of the graph make sense because the temperature of the beginning of the year should be increasing by observing the temp tendency before.

Take home questions

- How to make a forecast prediction based on quarter period?(every four months)
- What if I want to make analysis and prediction every ten years?
- ANy other models of forecast on time series?
- Try to do some forecasting on the topic you are interested in, such as stock…

References

1. more about Forecast: https://www.rdocumentation.org/packages/forecast/versions/8.1
2. time series: https://en.wikipedia.org/wiki/Time_series
3. time series in r: https://www.statmethods.net/advstats/timeseries.html
4. white noise series: https://en.wikipedia.org/wiki/White_noise
5. ARIMA model: https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average
6. seasonality: http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc443.htm
7. stationary and differencing: https://www.otexts.org/fpp/8/1