Time Series Forecast

YOU CHEN

2019/7/16

```
library(fpp2)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: forecast
```

```
## Loading required package: fma
```

```
## Loading required package: expsmooth
```

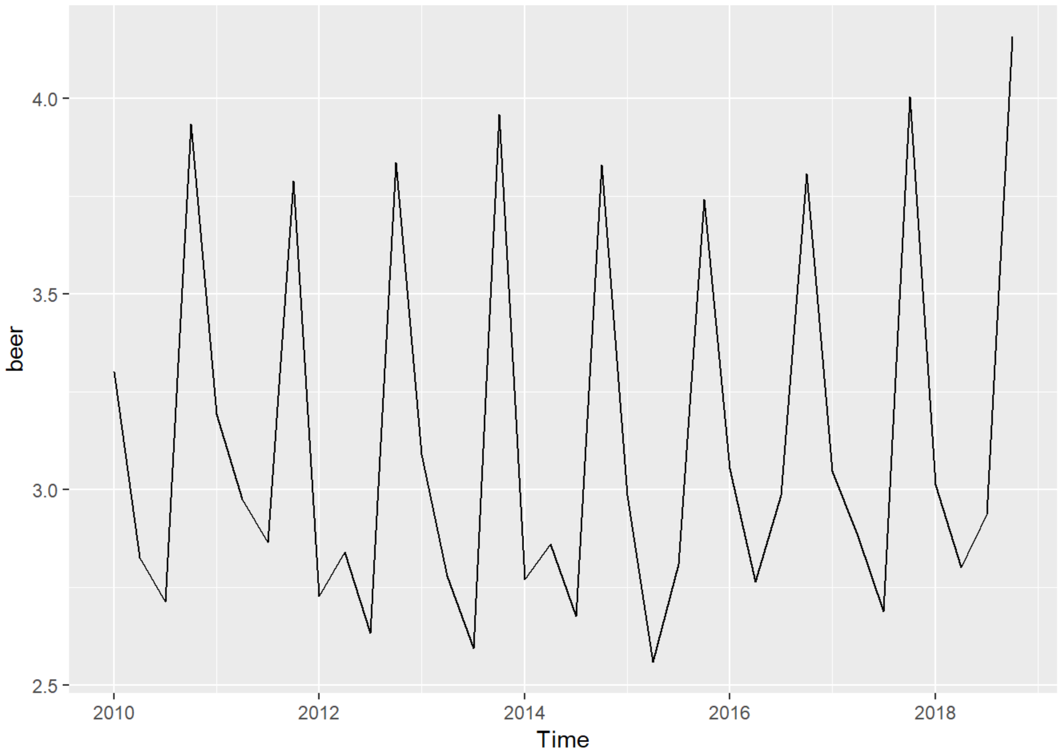
```
library(readxl)
NZ_TotalBeer_Quarterly <- read_excel("NZ_TotalBeer_Quarterly.xlsx",skip=1)
```

```
## New names:
## * `` -> ...1
```

```
View(NZ_TotalBeer_Quarterly)
beer <- ts(NZ_TotalBeer_Quarterly[,c(2)], start=2010,frequency = 4)
beer
```

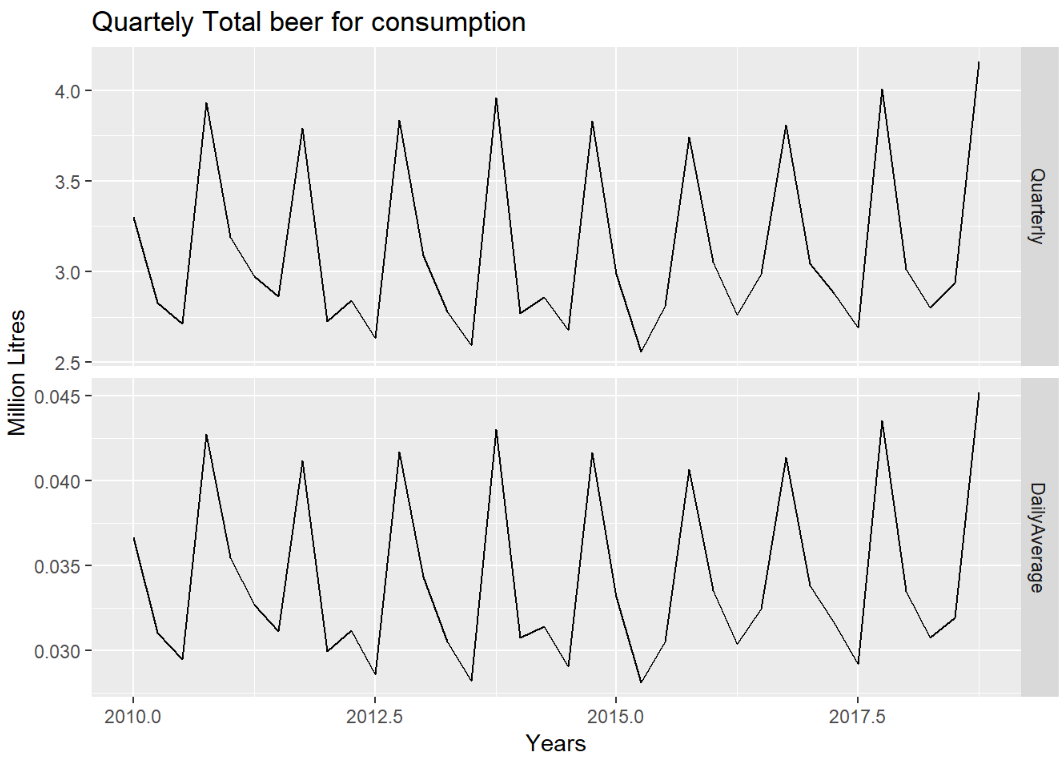
```
##       Qtr1  Qtr2  Qtr3  Qtr4
## 2010 3.301 2.826 2.712 3.934
## 2011 3.192 2.975 2.865 3.789
## 2012 2.728 2.840 2.633 3.837
## 2013 3.090 2.779 2.594 3.960
## 2014 2.771 2.860 2.676 3.831
## 2015 2.986 2.558 2.810 3.743
## 2016 3.054 2.764 2.985 3.807
## 2017 3.046 2.880 2.689 4.005
## 2018 3.013 2.800 2.937 4.158
```

```
autoplot(beer)
```



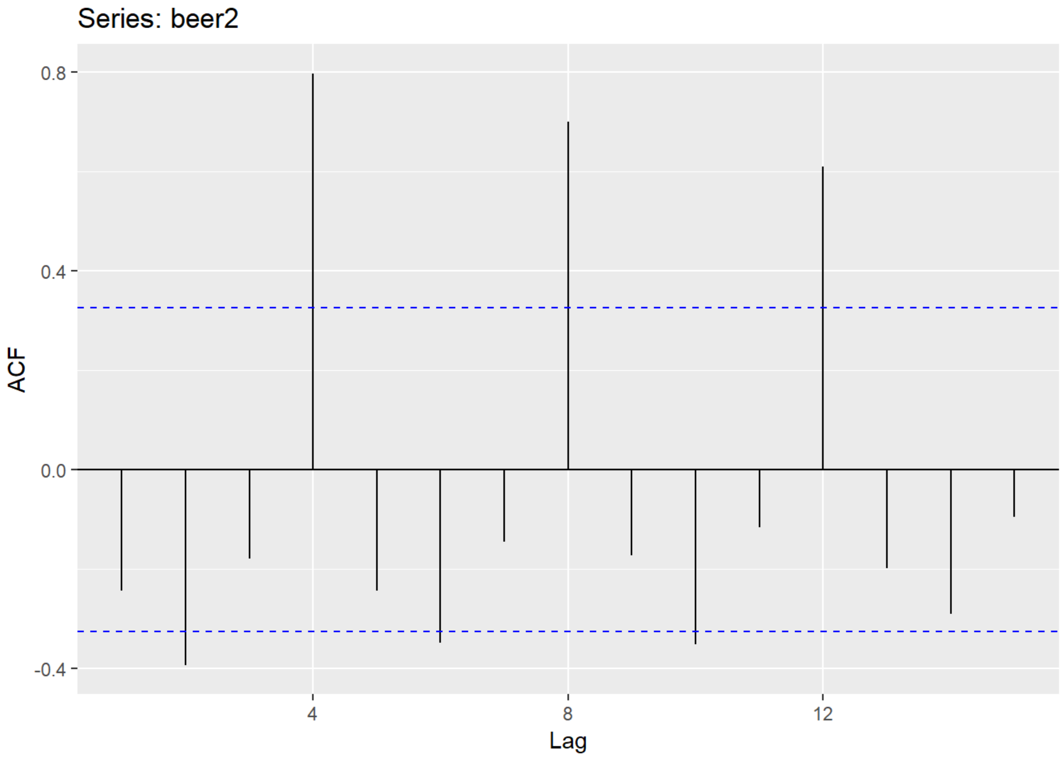
The data shows the total beer consumption for each quarter from 2010 to 2018. From the graph, we can find the consumption is always low between quarter 1 and quarter 2, and it tends to drop from quarter 1 to quarter to 2 every year, while it has a great increase from quarter 2 to quarter 3 and then decrease in quarter four yearly. That means it has strong seasonality, but it is hard to find the cycle and trend from this graph.

```
dframe <- cbind(Quarterly = beer,DailyAverage = beer/monthdays(beer))
autoplot(dframe, facet=TRUE) +
xlab("Years") + ylab("Million Litres") +
ggtitle("Quartely Total beer for consumption")
```



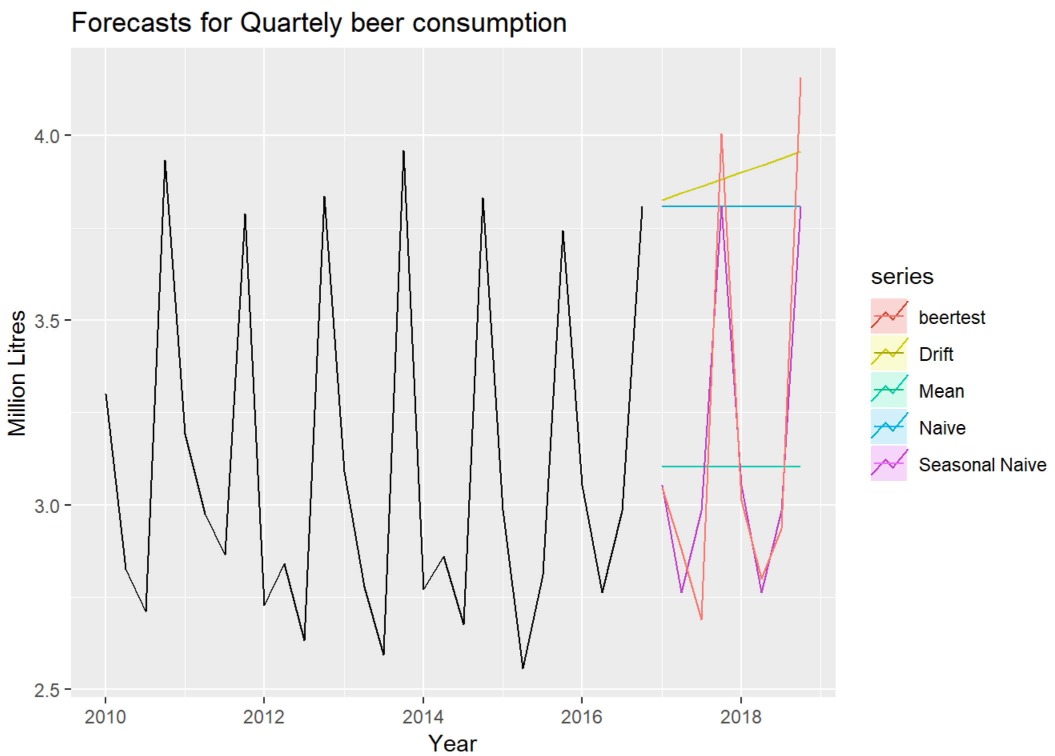
We can find that there is no variation due to the different month lengths, so there is no transformation needed.

```
beer2 <- window(beer,start=2010)
ggAcf(beer2)
```



In this graph: 1. r4 is higher than for the other lags. This is due to the seasonal pattern in the data: the peaks tend to be four quarters apart and the troughs tend to be four quarters apart, so there is a seasonality in it. 2. r2 is more negative than for the other lags because troughs tend to be two quarters behind peaks. 3. The dashed blue lines indicate whether the correlations are significantly different from zero, we can find lag 4,8,12 are significant different from zero, and as the lines are not all within the blue lines, there is no autocorrelation for these lags.

```
beer3 <- window(beer, start = 2010,end =c(2016,4))
beertest <- window(beer,start=2017)
autoplot(beer3)+
autolayer(meanf(beer3,h=8),series="Mean",PI=FALSE)+
autolayer(naive(beer3,h=8),series="Naive",PI=FALSE)+
autolayer(snaive(beer3,h=8),series = "Seasonal Naive",PI=FALSE)+
autolayer(rwf(beer3,drift=TRUE,h=8),series = "Drift",PI=FALSE)+
ggtitle("Forecasts for Quartely beer consumption")+
xlab("Year")+ylab("Million Litres")+
guides(colours =guide_legend(title ="Forecast"))+
autolayer(beertest)
```



From the graph, we can find seasonal naive is the best forecast method to predict the last two years' test data as the fluctuation is quite similar compared to the original data, the only thing that it doesn't quite catch is the increased trend, while drift method can show the trend is increasing.

```
beer4 <- window(beer, start=2017)
beerfit1 <- meanf(beer3,h=8)
beerfit2 <- naive(beer3,h=8)
beerfit3 <- snaive(beer3,h=8)
beerfit4 <- rwf(beer3,h=8)
accuracy(beerfit1,beer4)
```

```
##                         ME      RMSE       MAE       MPE     MAPE     MASE
## Training set -6.344325e-17 0.4587890 0.3901327 -2.015788 12.24724 2.411327
## Test set      8.742857e-02 0.5335806 0.4015357  0.409625 11.55703 2.481807
##                    ACF1 Theil's U
## Training set -0.2571958        NA
## Test set     -0.1935511 0.7592566
```

```
accuracy(beerfit2,beer4)
```

```
##                       ME      RMSE       MAE        MPE     MAPE     MASE
## Training set  0.01874074 0.7273785 0.5960741  -1.892379 18.29363 3.684208
## Test set     -0.61600000 0.8102595 0.7532500 -22.162665 25.50901 4.655679
##                    ACF1 Theil's U
## Training set -0.4144973        NA
## Test set     -0.1935511  1.022384
```

```
accuracy(beerfit3,beer4)
```

```
##                        ME      RMSE       MAE        MPE     MAPE
## Training set -0.006791667 0.1932374 0.1617917 -0.3766898 5.502496
## Test set      0.038500000 0.1833105 0.1367500  0.5541664 4.120551
##                   MASE        ACF1 Theil's U
## Training set 1.0000000 -0.03592964        NA
## Test set     0.8452228 -0.47546252 0.2569898
```

```
accuracy(beerfit4,beer4)
```

```
##                       ME      RMSE       MAE        MPE     MAPE     MASE
## Training set  0.01874074 0.7273785 0.5960741  -1.892379 18.29363 3.684208
## Test set     -0.61600000 0.8102595 0.7532500 -22.162665 25.50901 4.655679
##                    ACF1 Theil's U
## Training set -0.4144973        NA
## Test set     -0.1935511  1.022384
```

From the result, we can find the smallest RMSE of test set which is 0.0099269 form Seasonal Naive Method. Therefore, Seasonal Naive is the best method for this time series.