Forecast Methods

Pawan Kumar

02/06/2021

Average method

Forecast at time step “h” is average of the previous historic data

$y_{t+h|t} = (y_1 + y_2 + \ldots\ldots.. y_t) / \mathrm{t}$

```
# mean(y,h) -> y is the time series and h is the forecast horizon

library(fpp2)
beer_data <- window(ausbeer, start = 1992, end = c(2007,3))
beer_data[1:20]
```

```
##  [1] 443 410 420 532 433 421 410 512 449 381 423 531 426 408 416 520 409 398 398
## [20] 507
```

```
frequency(beer_data)
```

```
## [1] 4
```

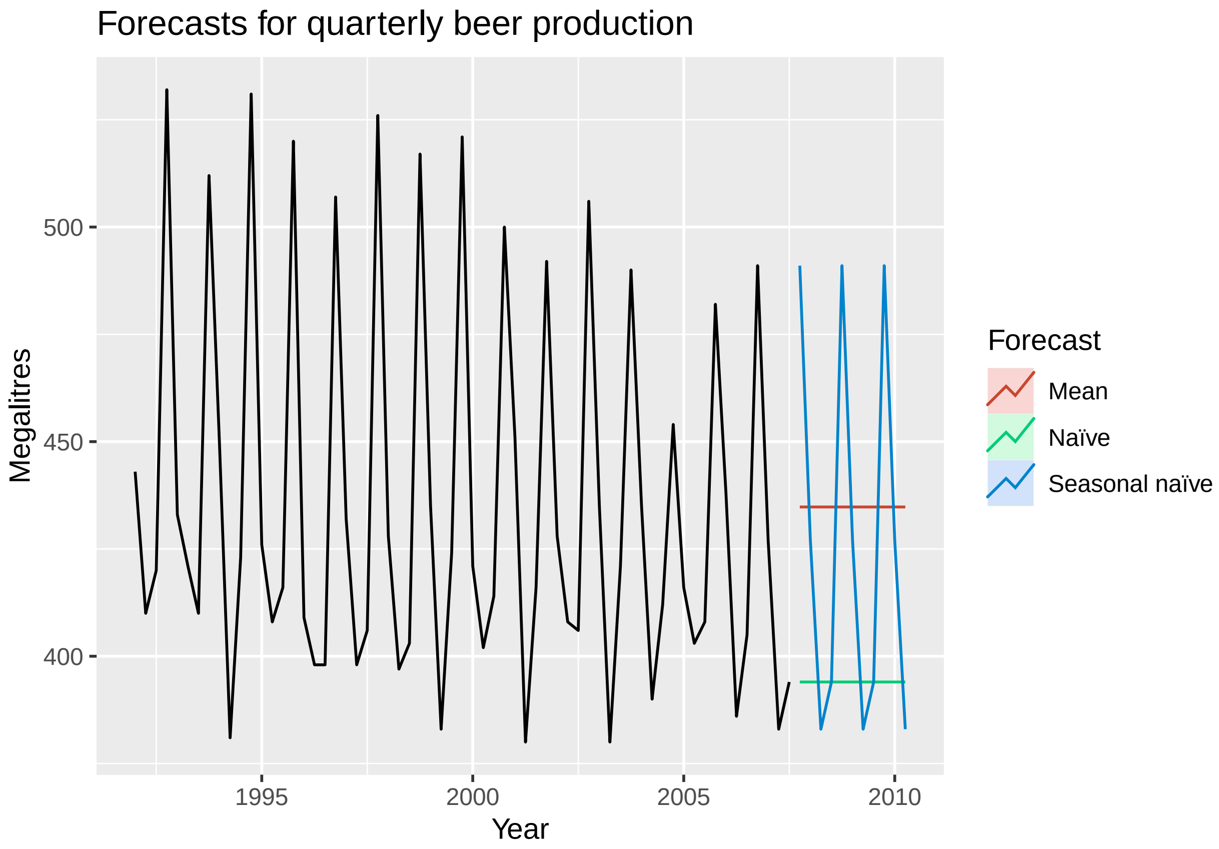
Table 1: Frequency table

Frequency	Meaning
12	Monthly
4	Quarterly
2	Semi annually
1	Annual

```
summary(beer_data)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   380.0   405.5   421.0   434.8   450.0   532.0
```

Plot



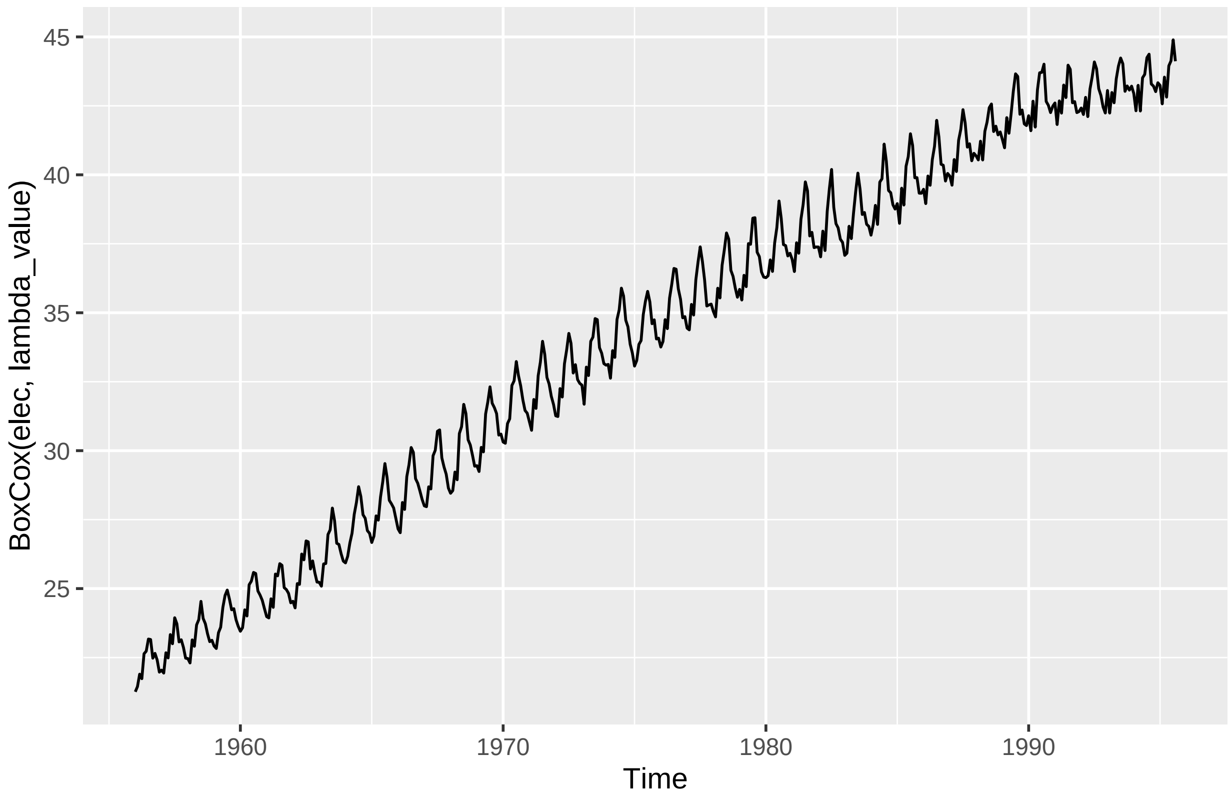
Box-Cox Transformation

Usage of BoxCox.lambda() function to choose adequate lambda value:

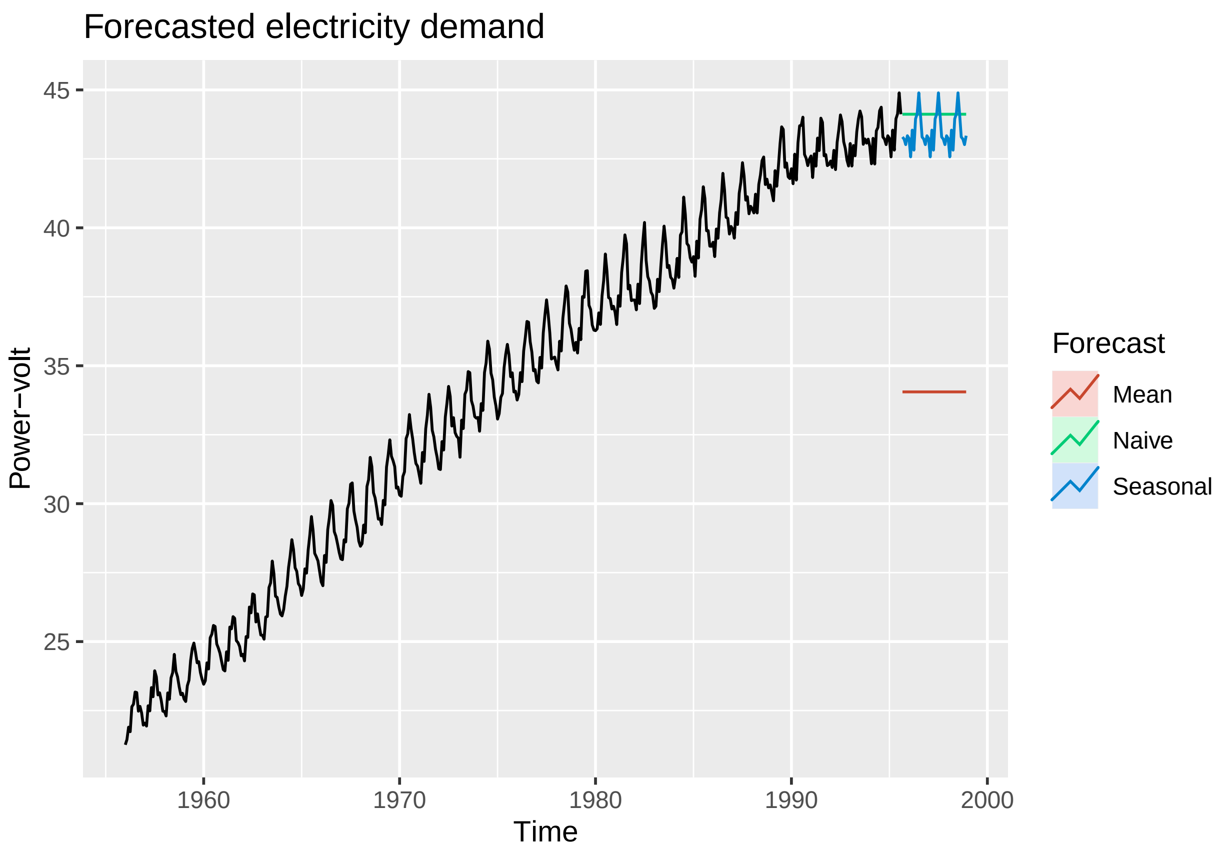
```
lambda_value <- BoxCox.lambda(elec)
lambda_value
```

```
## [1] 0.2654076
```

```
autoplot(BoxCox(elec,lambda_value))
```



```
transformed_series <- BoxCox(elec,lambda_value)
# Plotting
autoplot(transformed_series)+
autolayer(meanf(transformed_series,h=40), series = "Mean", PI = FALSE) +
autolayer(naive(transformed_series, h = 40), series = "Naive", PI = FALSE) +
autolayer(snaive(transformed_series,  h = 40), series = "Seasonal", PI = FALSE)+
ggtitle("Forecasted electricity demand") +
xlab("Time") +
ylab("Power-volt") +
guides(color = guide_legend(title = "Forecast"))
```



Note: Mean and naive forecast make no sense, to an extent the seasonal forecast aligns, still it lacks to capture the trend. As the transformed series has trend and seasonality.

Importantly, after chosing a transformation, we need to revert it back to reflect upon the original data and generate and thus gets its forecast value.

```
seasonal_forecast <- snaive(transformed_series, h = 11)

View(seasonal_forecast)
length(seasonal_forecast[["lower"]])
```

```
## [1] 22
```

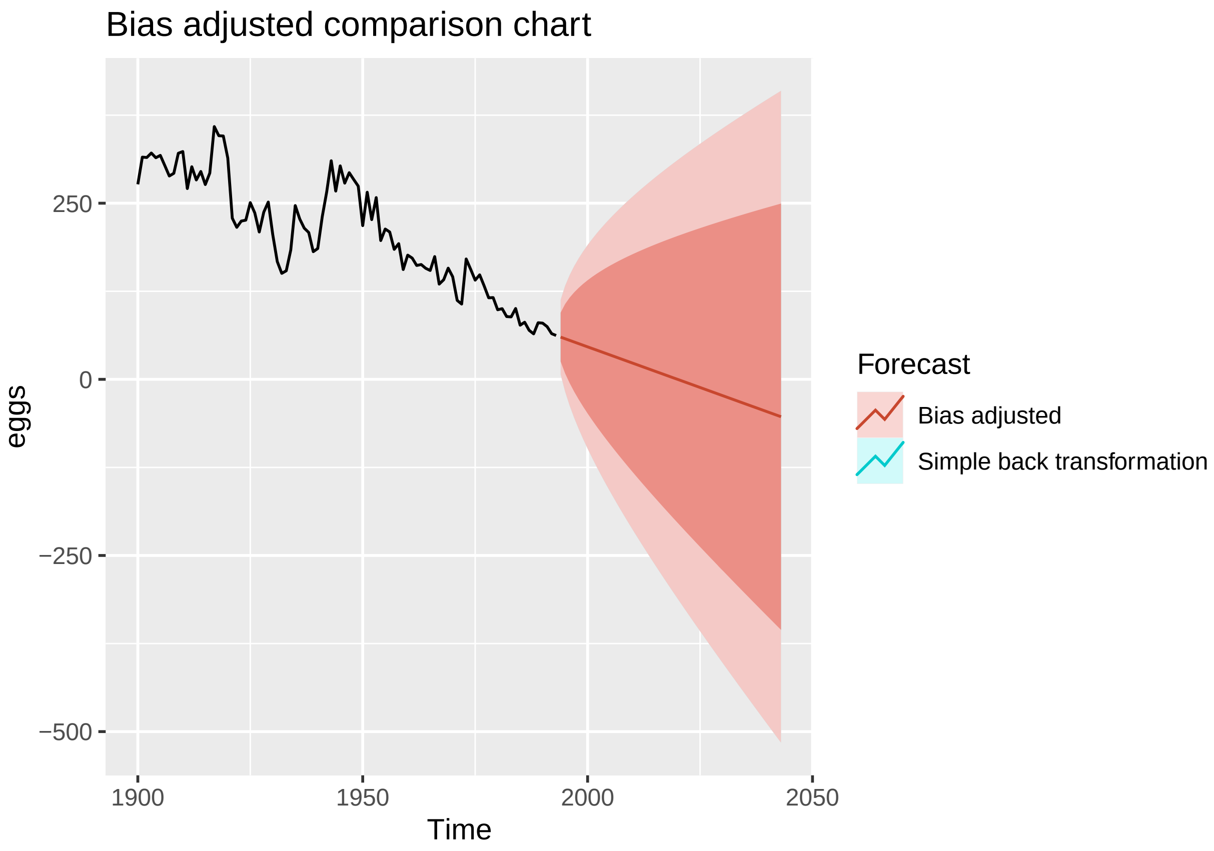
```
# lower values
reverse_transformation <- InvBoxCox(seasonal_forecast[["lower"]][1:22], lambda_value)
```

Bias Adjusted Means

```
View(eggs)

fc1 <- rwf(eggs, h = 50, drift = TRUE)
fc2 <- rwf(eggs, h = 50, drift = TRUE, biasadj = TRUE)

autoplot(eggs) +
autolayer(fc1, series = "Simple back transformation") +
autolayer(fc2, series = "Bias adjusted") +
ggtitle("Bias adjusted comparison chart") +
guides(color = guide_legend(title = "Forecast"))
```


Bias adjusted comparison chart
eggs
Time
Forecast
Bias adjusted
Simple back transformation