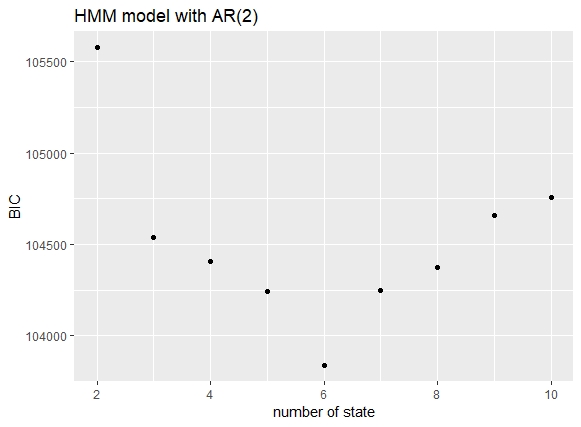
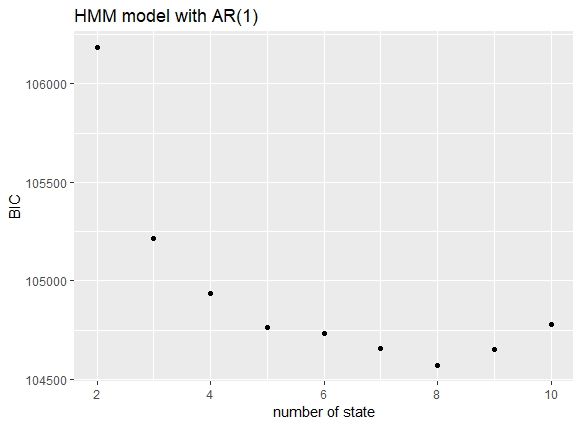
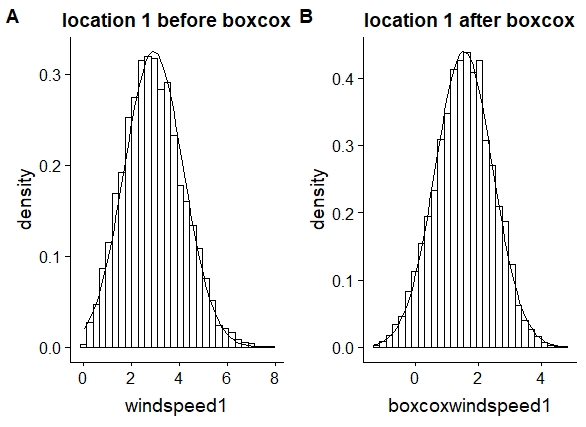
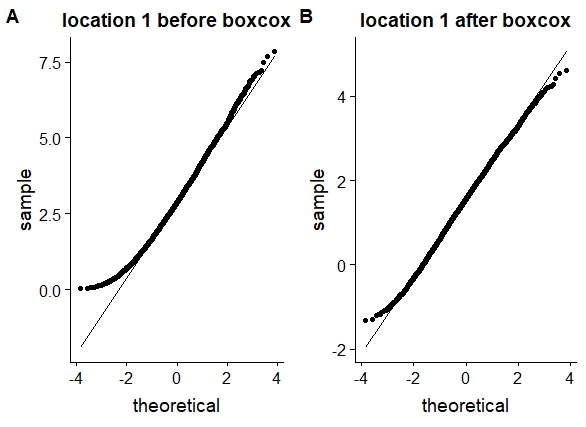
It is a bit long, so I use a word to make it clear.

This is the three HMM-AR model without boxcox the value.

It seems reasonable and making sense, the BIC decrease and increase, and we can get an optimal BIC.

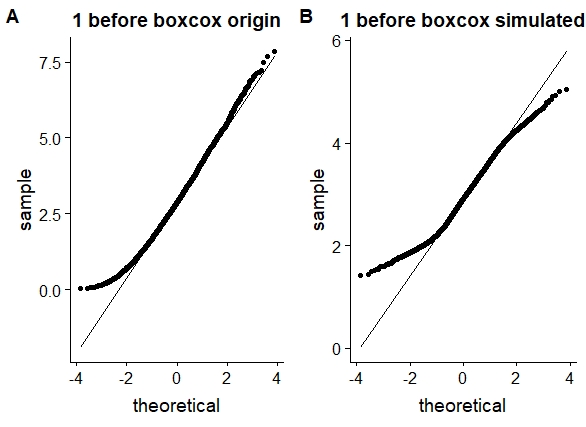


And we apply the boxcox, using ‘MASS’ package,

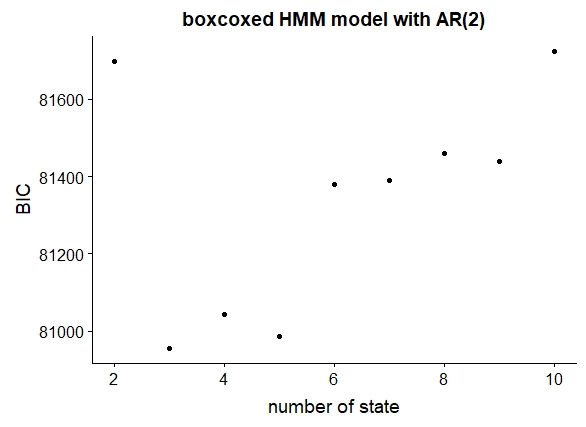
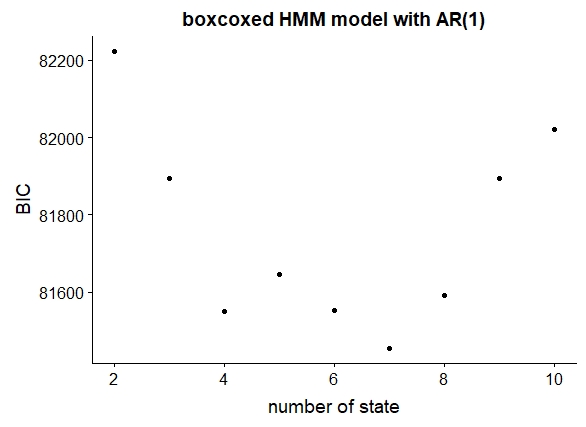
\

I take location1 to check, here is the basic result of the data, before and after boxcox, it seems to be more normal distributed. The tail has been improved a lot.

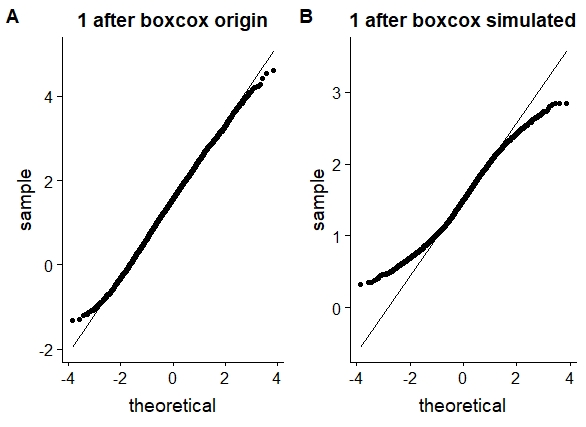
Below is the origin data not boxcoxed, and the origin data with the HMM-AR predict, and we plot the qqplot of the predicted result from the model. (the model using the data not being boxcoxed.



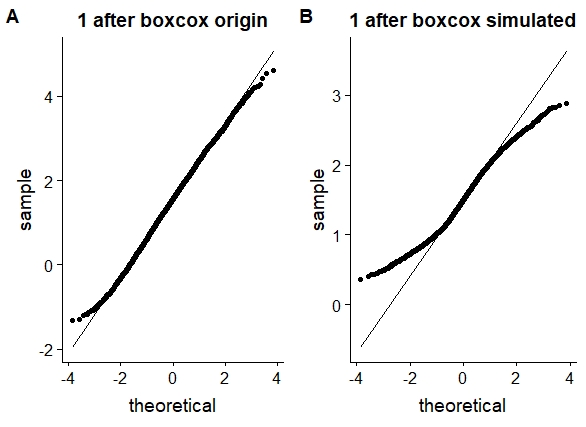
And then, we apply the boxcox-data on the HMM-AR model,

And it become strange, although it still increases after 8. The distribution now is different, and we get different state when optimal.

Here is the boxcox-data based HMM-AR model with first order AR1, and the optimal state 4, we get the below qqplot comparision.



And here is the second order AR2, we can get the below qqplot.

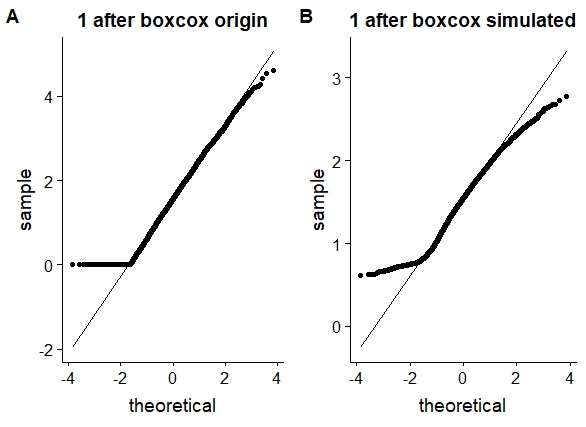


I think is maybe the problem of the negative value in the boxcoxed-value.

Yes, using Boxcox results in some negative value, but not big value, actually they are small value no more than -0.5.

So, I try to set the negative value to 0.

For the AR2, no matter, 0, or 0.1 or 0.3, or random between(0.1,0.5), it always returns the ‘error singular matrix’, so I just do it on AR1.

Here is the AR1, with no negative, the BIC is smaller, but the distribution seems to be worse than the previous one.

