Forecasting the Score of a Player

Nam Vo

Dr. Jinwon Kim

April 16, 2017

**Abstract & Introduction**

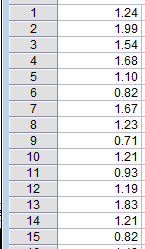
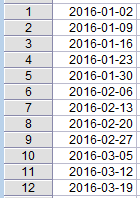
The data I will be analyzing and forecasting will be several different values of a player’s score from an online game called “Counter-Strike”. To preface my research, my data involves an online game called “Counter-Strike” which is a first-person shooter involving players on opposing teams who try to kill each other to obtain the highest score to win different matches. The higher score each player earns, the higher chance their respective team wins each match. These players perform at different tournaments, across the world, and attempt to win prize money for every tournament they win. The specific data I am analyzing involves a player named ColdZera and his score on a weekly basis from January 2016 to April 2017 with a total of 68 different observations. First, I will find out different information regarding my data by viewing its descriptive statistics then test the stationarity of the data by viewing its correlogram and line graph. From this, I will suggest different models which matches my data and test the reliability of the different models by performing an in-sample forecast and evaluating its errors, viewing the white noise characteristic of the residual, the standard error of regression, Akaike Information Criterion, Schwartz Criterion, and the significance of my estimates. My main motivation involves how unconventional my forecast is. Most forecasters work with macroeconomic data involving GDP, inflation, interest rates, and various prices; whereas I am working with the online scores of a professional videogame player. I want to see if random shocks or past values have any forecasting power in Coldzera’s score and if I’m able to accurately forecast how well he can perform during his next tournament play. Although very intriguing to me, this information may not be very important in the real world.

**Methods & Theoretical Analysis**

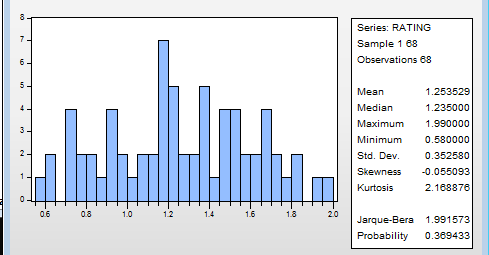
First off, I obtained my values from a reputable website which tracks every professional player’s score.

<https://play.esea.net/users/720230?tab=history>

From here, I manually entered the scores into an excel sheet along with its corresponding dates all the way down to t=68

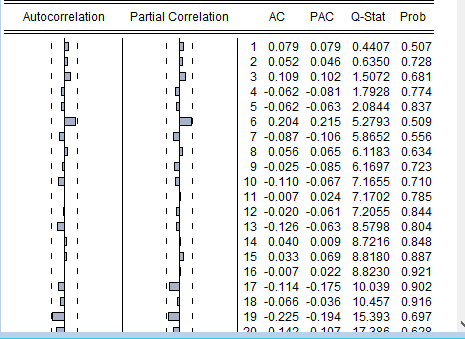


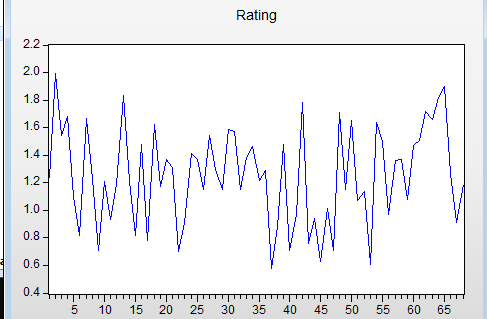
I checked the descriptive statistics and found:



I find that the central values of the data tend between 1.2 and 1.3 and vary with a standard deviation of 0.35. So, on an average day, Coldzera obtains a score of around 1.25

From here, I checked the correlogram and the line graph which looks like

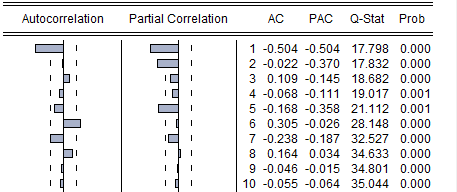
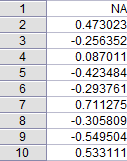


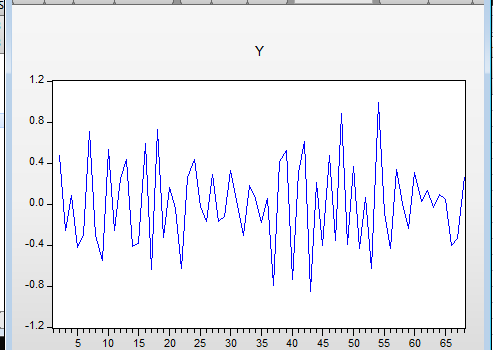


So the correlogram exhibits a white noise process which means it has no forecasting power and acts very erratically. I will transform my independent variable into a growth function to further transform the data into a stationary process and view the forecasting power of the growth of the data. I input this line of code into eviews:



To generate a new, more stationary variable, Y, and got this new correlogram and line graph as well different data values:





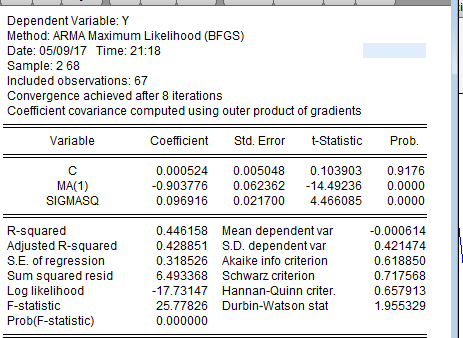
This correlogram, from our growth function, looks much more favorable to forecast with as we have significant values under the autocorrelation and partial autocorrelation functions. From here, I can choose either an AR(p), MA(q), or ARMA(p,q) model and test their reliability according to the correlogram. **AR processes** see a smooth decay towards zero in autocorrelation as well as a limited number of spikes, different from zero in the partial correlation; while **MA processes** see a smooth decay towards zero in the partial correlation function and a limited number of spikes, different from zero in the autocorrelation function. For my correlogram, I don’t see a smooth decay towards zero for either ACF or PACF (perhaps in the PACF but my data isn’t too clear) so I will estimate an AR(p), MA(q), and ARMA(p,q) and compare the three. The first order of autocorrelation looks to be significant while the subsequent orders are insignificant, which is a characteristic of an MA(1) process so I’ll estimate an MA(1). The first two orders of the PACF are significant until the 5th order so I can estimate an AR(2) or even an AR(5) but I’ll stick to AR(2). Furthermore, I see some cycles where the Y values have periods where it stays about its mean or below its mean around this area.



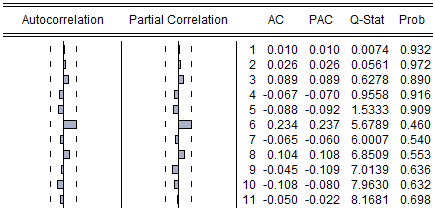
Because I can see a decay towards zero in both ACF and PACF, I’ll combine the two models to estimate an ARMA(2,1). To estimate MA(1) in eviews, I input this line of code



Got this output window,



Got this correlogram for its residuals



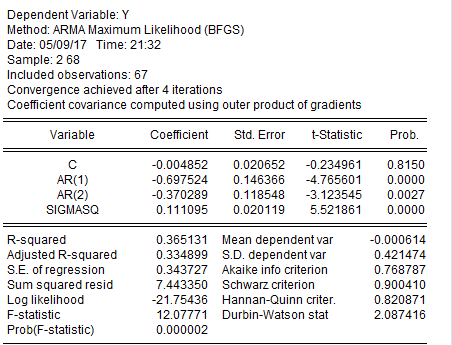
And performed an in-sample forecast using the data from January 1st, 2016 to March 25th, 2017, and forecasted April 1st, 8th, and 13th, then evaluated the error in my data

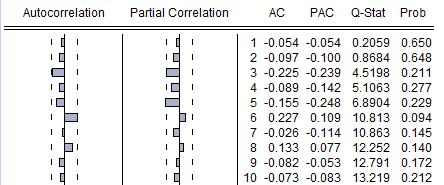


I noticed how the values after the first forecast horizon revert back to its unconditional mean which is characteristic of an MA(1) process.

These are the values of my in-sample forecast. I will continue to estimate equations, the correlogram for the residuals, and perform an in-sample forecast for my next two models, then interpret and evaluate them thereafter. Here is the resulting process when estimating an AR(2) model.



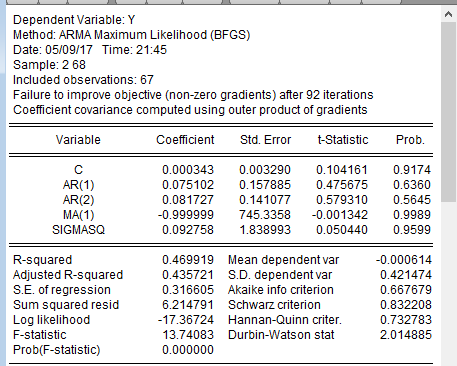
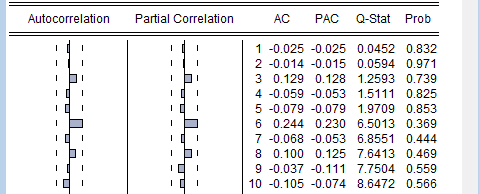






And my last ARMA(2,1) results and estimations are as follows





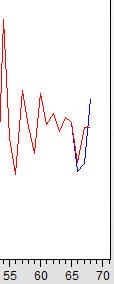
Now, I can assess each model using my previously stated criteria. Each model contains a correlogram, of the residuals, which resembles a white noise process, which means all the error terms cannot forecast other errors, which is good. Next, I can view each model’s standard error of regression which is the variability of the errors across all models; so, the smaller the value, the better. The smallest standard error is our ARMA model with a value of ~0.32 which might be attributed to the fact that this model has more variables; so we look to our AIC and SC which considers trade-offs when adding more variables. According to the AIC and SC criterion, our MA model is the most reliable with the lowest values of 0.61 and 0.71 respectively. When considering the significance of each estimated coefficient, our MA and AR models seem to have significant values, of phi and theta, with the lowest P-values near 0 with high P-values nearing 1 for our constant, across all models, so I may consider dropping it for a better forecast. Lastly, I will estimate the forecasting error associated with my in-sample forecast, sum them up, and evaluate the total sum of errors across the in-sample forecast of t=66, 67, and 68. My realized values are



So I will take these values, subtract it from my in-sample forecast across my 3 different models, and find the model with the absolute value of the lowest summed forecasting error to be the most reliable, assuming a symmetric loss function.

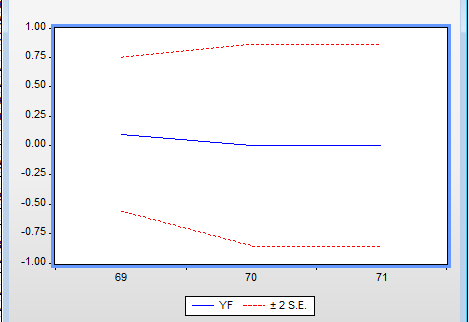
For my MA(1) model, my summed error is [(-0.40+0.32)+(-0.33-0.0005)+(0.27-0.0005)]=-0.141  
For my AR(2) model, my summed error is [(-0.40+0.07)+(-0.33-0.03)+(0.27+0.0005)]=-0.420  
For my ARMA(2,1) model, my summed error is [(-0.40+0.38)+(0.33+0.02)+(0.27+0.03)]=0.63

So my MA model has the lowest summed error and will be my model of choice to point and interval forecast, as illustrated by my combined graph of the in-sample forecast and realized values.

 The blue line represents my in-sample forecast while the red line represents the actual realized values. The distance between the two lines represents the forecasting error which is minimized under the MA(1) model from the previous calculation.

First, I restructured my work file to include t+1, t+2, and t+3 (periods 69, 70, and 71) to perform a 1 to 3 step forecast then performed the forecast.





The first window is my 3-step forecast and the second window is my interval forecast up until t=71. To interpret this data, Coldzera will most likely earn a growth of score rating by 0.91 in his next game of Counterstrike, based off the information set, which is a major improvement. Since writing starting this paper, Coldzera has played his next game and improved his rating by about 0.41 so our forecasting error would be 0.41 – 0.91 which is -0.5. In the future, I can improve this forecast, and minimize my forecasting error, by including more observations, estimating more ARMA models, dropping insignificant coefficients, or finding more variables to perform a VAR forecast. A VAR forecast would require me to find out variables that might affect Coldzera’s rating like the amount of sleep he gets on tournament days or the numbers of hours he’s practiced.

**Conclusion**

Overall, I’m very satisfied with my estimation results as I found the most accurate process to model my data by suggesting different models based off the different correlograms, evaluated each model on six different criteria, and performed a multi-step point and interval forecast with said model. If I could give Coldzera a tip to maximize his performance and score, I would tell him to continue doing what he’s doing as he’ll most likely see a growth of 0.91 in his score.