Marathon Record Progression
A Time Series Analysis

Tim Chen

Spring Quarter 2023

1 Introduction

The marathon is a prestigious long-distance running event that captures the imagination and showcases the resilience and determination of athletes. Originating from the ancient Greek legend of Pheidippides, who ran from the city of Marathon to Athens to deliver a message of victory. Nowadays, the marathon holds great significance in the world of sports as the race has become a platform for individuals to test their physical and mental limits, pushing the boundaries of human capabilities. The event brings together people from diverse backgrounds, fostering a sense of camaraderie and unity as runners strive towards a common goal. The evaluation of marathon records over time is crucial in measuring the progress and evolution of the sport. Monitoring the progression of marathon records can help track advancements in training techniques, equipment, and overall athletic performance. Of course, we have to mention the best when we discuss any sport. As of now (June 9, 2023), the men's world record for the marathon is held by Eliud Kipchoge of Kenya, who completed the 2018 Berlin Marathon by a time of 2 hours, 1 minute, and 39 seconds.

Known factors impacting the result of a race:

1. Course Elevation and Terrain: The elevation profile and terrain of a marathon course play a significant role in performance. Hilly courses with steep inclines can slow down runners, while flat and fast courses may facilitate faster times.

2. Season and Temperature: The time of year and temperature can affect performance. Cool temperatures during autumn or spring are generally more favorable for runners, as opposed to the heat and humidity of summer months.

3. Training and Preparation: The quality and intensity of an athlete's training regimen, including factors like mileage, speed work, and strength, conditioning, can impact their speed and endurance during a marathon. I.e. People are getting faster in each generation.

4. Competition Level: The presence of highly competitive and skilled athletes in a race can create a competitive environment that encourages faster times. A strong field of participants often pushes each other to achieve their best performances.

2 Data

The Data is obtained from World Athletics Organization using python scraping package (beautifulsoup), and the data after preprocessing is converted into csv form to be used later for time serie analysis in R.
The raw data is recorded in a tidy format, with each row being observation, and the columns being name, date of birth, date and venue of the race, and most importantly, the record time. I summarized the data by extracting the best performance from all the runners of each months from January 2001 to May 2023, with 7 missing months (two months without data, and 5 months during COVID quarantine). Since time series data depends heavily on periodicity, the best way of dealing with missing data is through imputing. I decided to use the mean of each months to impute due to its yearly cycle that seems to appear in the data, which will be talked about in the next section in detail. I also reindexed the first month as 1, second month as 2, etc for the purpose of understanding the time unit, which in our case is one month. Resulting time is also converted to seconds in order to perform quantitative analysis. Then the data becomes a standard time series data and is ready to be analyzed.
First we split the data into training and testing, with the first 80% being training (January 2001 to July 2018), and the last part up till today being the testing set. The original time serie plot for the whole dataset is shown in Fig.0. An initial analysis involves plotting the ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) of the original data. The result shown in Fig.1. It seems like both plots are indication that the original sample is not a stationary serie, which means that trend removal and cycle removal process is crucial to transform the data as the first step.

3 Trend and Cycle Removal

For the general trend, I followed the procedure of fitting an exponential curve with negative coefficient on time, proposed by Angus in 2019, I get the result in Fig.2. With the trend function modeled as

$$y = e^{8.962-0.0013t} \tag{1}$$

The difference between our data (both training and testing) and the trend prediction becomes our detrended data.
For the spectral analysis, I use mvspec to find the periodogram of the detrended data, and the result are in Fig.3. It seems like that there are two clear narrow-band peak, at 1/12, and 1/6 respectively. It also seems like the harmonics of 1/12 all seem to be showing in the plot as well, so I propose that the period of 12 months is a good estimate of the cycle in this time-series dataset. To further analyze the cyclical result, I used the danielle kernel with parameter 3 to smooth the plot non-parametrically, and as shown in Fig.4, the peaks around 1/12 is still significant. For a parametric AR estimate of the periodogram, we found out that an AR(12) is a great candidate by using the BIC/AIC curve metric (Fig.5). The result shown in Fig.6 further indicates my hypothesis of the yearly cycle.
To remove the cycle, I looked at the mean of each month for which it will be deleted from both the training and testing data. The plot of the mean is shown in Fig.7, and this plot seems to indicate that race result in summer seems to increase significantly compared to

the race result in winter. A very intuitive hypothesis is that the hot weather and humidity makes it harder for the runner to perform at their very best.

4 Model Fitting

As for the fitting of the time-series data, we have to dig deep at the residuals of the detrended/decycled training data. After plotting the ACF and PACF of the residual (Fig.8), and compare with the original training data (Fig.1), it is not hard to notice that both plots are maintained under the .2 mark, a sign indicating the data is stationary. Also in Fig.8, we can see that both plots don't have a clear cut-off point, and instead, tails off slowly as lag distance increases. Such phenomenon corresponds to an ARMA model.

From now on, it is important to note that all the analysis and metric results are based on the detrended and decycled data. Except for the times when I transformed the fitted data to the original data space (by adding the mean for each month, and apply the inverse function of the trend) in order to better visualize the model result and give a more intuitive sense of how well the model fits and predicts.

To find the parameters that best fit our training data, I used BIC and RMSE to see the result of 100 different ARMA(p,q) models using grid-search, with p and q ranging from 0 to 9. The result of using RMSE as metric is shown in Fig.9, and we pick the parameters that has the lowest RMSE value, which is an ARMA(7,6) model, with an RMSE of 83.08. However, it is worth noting that values of several models are close to the 83.08 value, meaning there are many possible candidates that works almost as well as the best model by such metric. It also means that a slight shift in the training data (maybe using only the last 80%) can lead to entirely different model parameters. So I did the same choosing process using BIC(Bayesian information criterion), and this time we get ARMA(1,1) as the best model (Fig.10). So I proceed with both models and plot how well it does on the training data. (Fig.11, Fig.12) The instability in ARMA(1,1) in the beginning is because that I keep only the starting prediction at time 1 as training and go on as such, but keep 7 predictions from time 1-7 as training for ARMA(7,6), which understandably makes the first few prediction more accurate. Overall, I decided to proceed with the ARMA(1,1) since it renders more meaning and interpretability to the residual analysis.

5 Forecasting

I tried two ways of forecasting. The first being 1-step look ahead, and it is implemented manually. The second way is through the forecast function from imported r package forecast, and the look ahead period is set automatically to 1, meaning 12 months in our case. Using both method, I fit the model on the test data using ARMA(1,1). The resulting plot for testing with 1-step look ahead is shown in Fig.13, with an RMSE of 154.9, and the plot for testing with 12-step look ahead is shown in Fig.14, with an RMSE of 89.26825. This indicates that the 1-period look ahead forecast actually performs better than the 1-step look ahead in our case, and the reason for that is probably due to the less variance in the 1-period look ahead.

6 Conclusion

In general, I fit only the ARMA model because doing differencing doesn't make intuitive sense even though it may help make the serie more stationary. The test results indicate that the ARIMA(1,0,1) model seems to be a good fit for our specific training data. However, it is important to note that this model's performance may vary when applied to different datasets, even when using a subset of the training data. An interesting fact about our model in the long run is that human being may break the 2 hour mark in an official Marathon race by the average time of May 2031.
To further improve the model, future work should involve a deep dive into covariates, such as weather data, steepness of venue routes and other relevant factors that may affect the outcome. Additionally, dealing with missing data seems to be especially crucial in this model to ensure accurate predictions based on our close result from training metrics. Furthermore, it is recommended to apply the model to other race events or similar scenarios to assess its generalizability and performance in different contexts. This would provide a broader perspective on the model's effectiveness and applicability.

References

[1] Worldathletics.Org *"World Athletics: Marathon - Men - Senior - Outdoor - 2023."*, www.worldathletics.org/records/toplists/road-running/marathon/outdoor/men/senior/2023?page=1 .Accessed 2 June 2023.

[2] Angus, Simons. *A Statistical Timetable for the Sub–2-Hour Marathon*, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6613719/pdf/mss-51-1460.pdf

[3] Shumway, Robert H., and David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples. Springer*, 2017. Chapter 3-4.

Appendix

Figure 0: Time Series Plot of the Entire Dataset

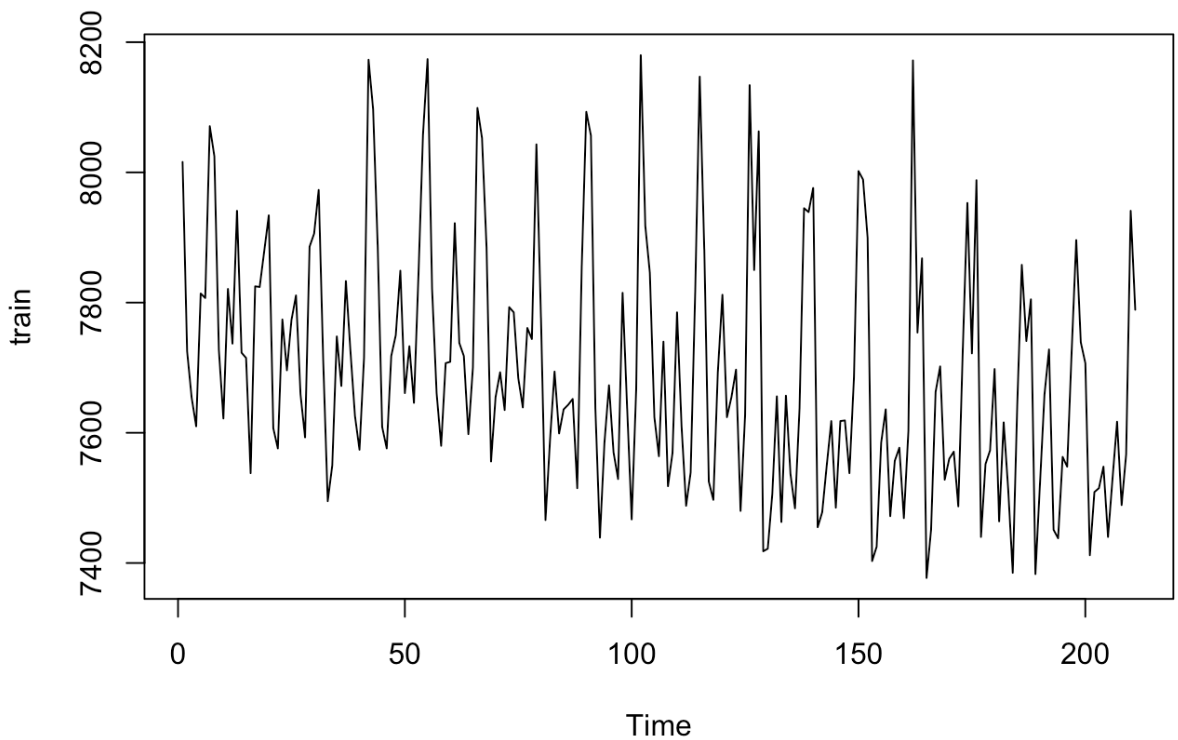

Figure 1: ACF & PACF of the Original Dataset

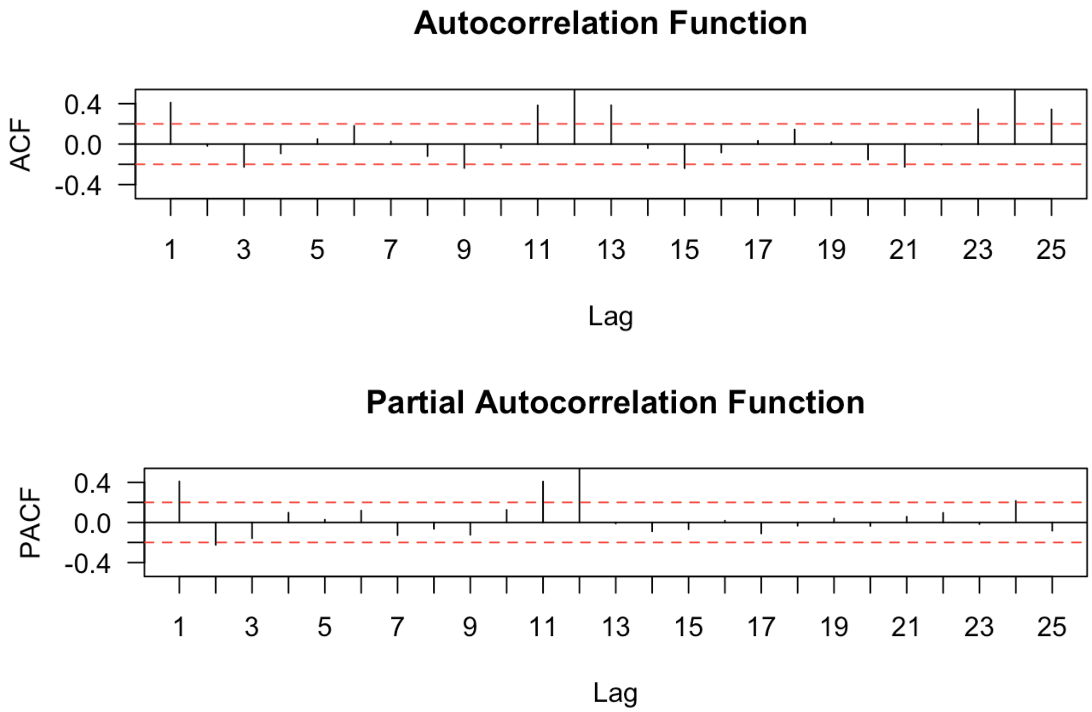

Figure 2: Fitted Trend $y = e^{8.962-0.0013t}$

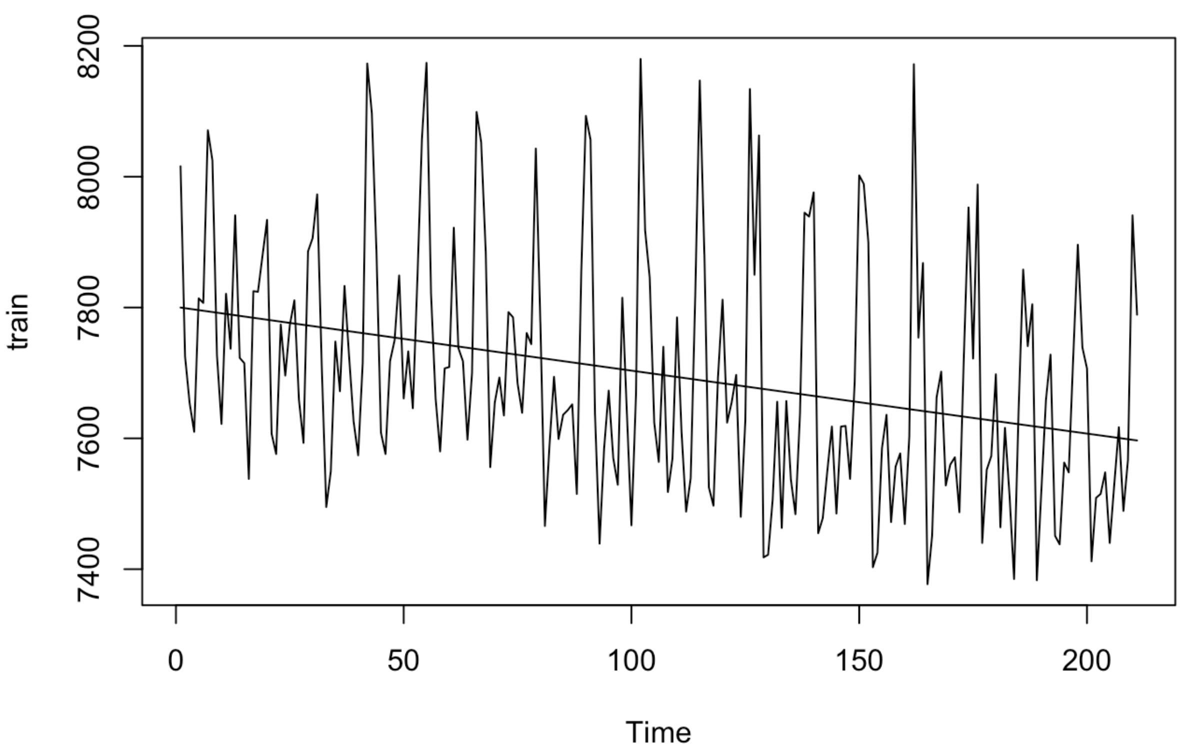

Figure 3: Raw Sample Periodogram

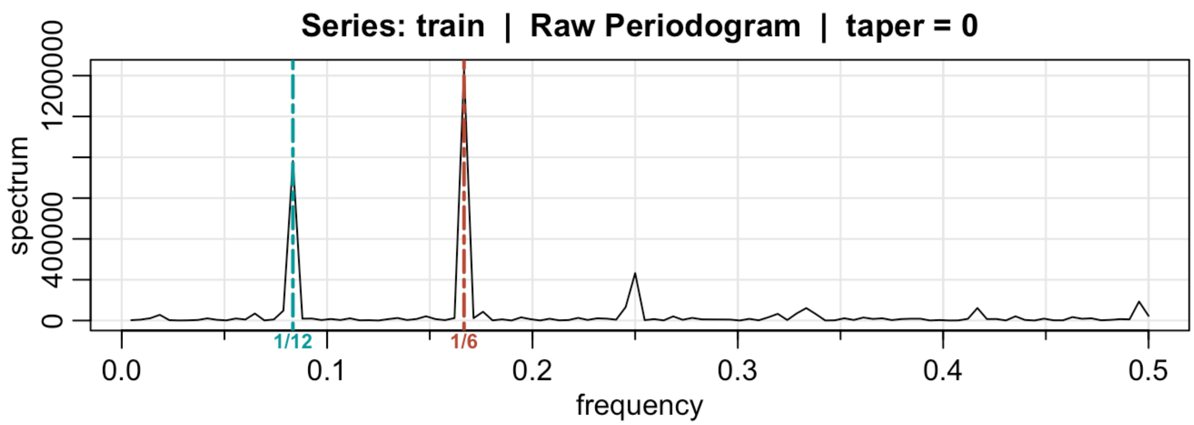

Figure 4: Non-Parametric Smoothed Periodogram

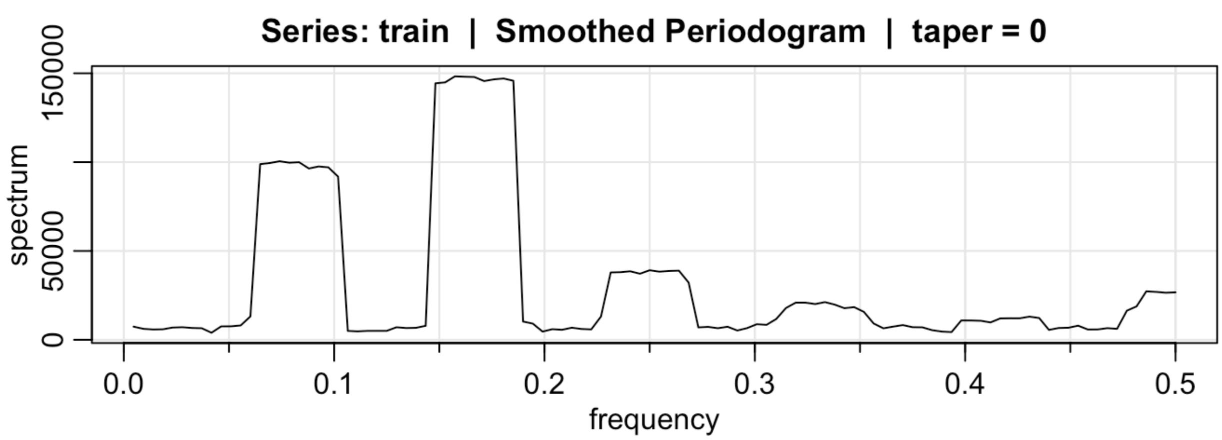

Figure 5: AR(p) parameter search using AIC/BIC

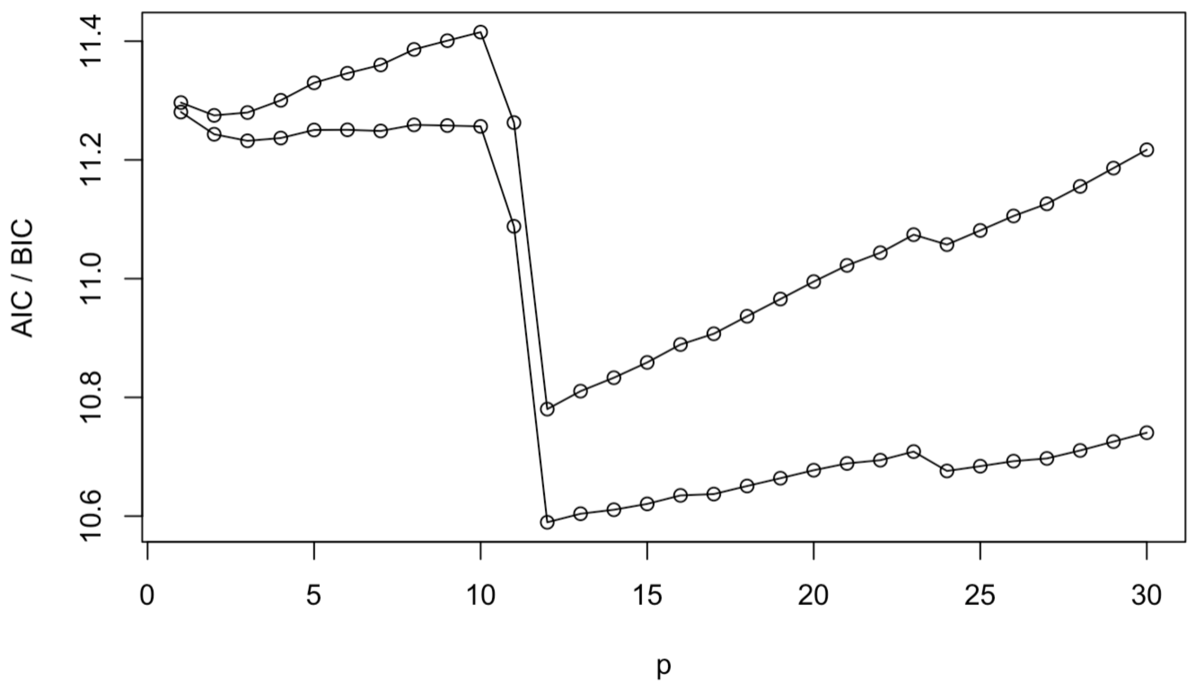

Figure 6: Parametric Smoothed Periodogram

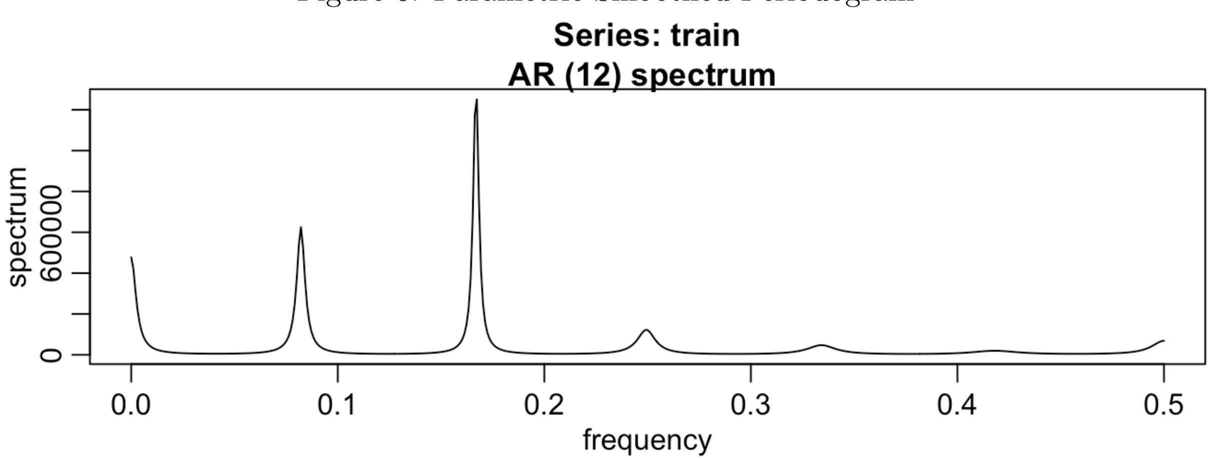

Figure 7: Monthly Mean Plot

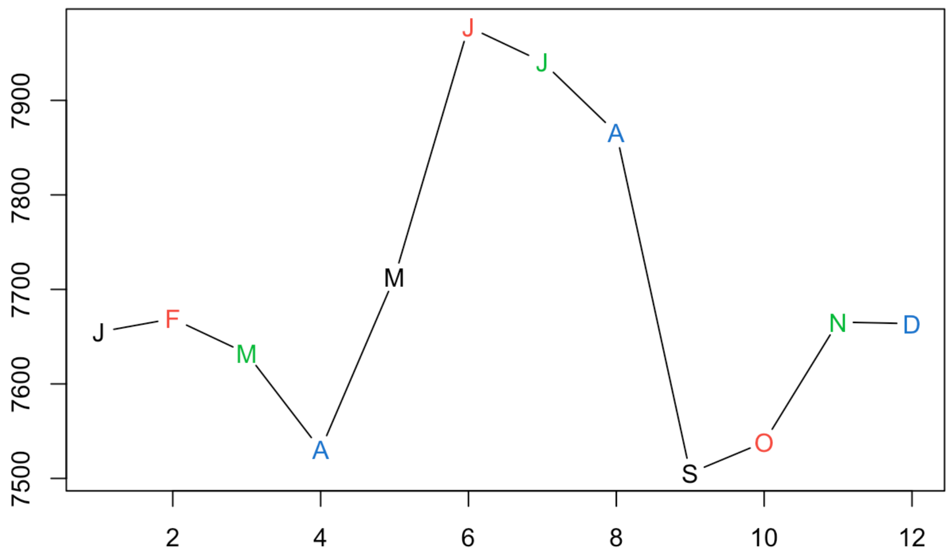

Figure 8: ACF & PACF of Residuals

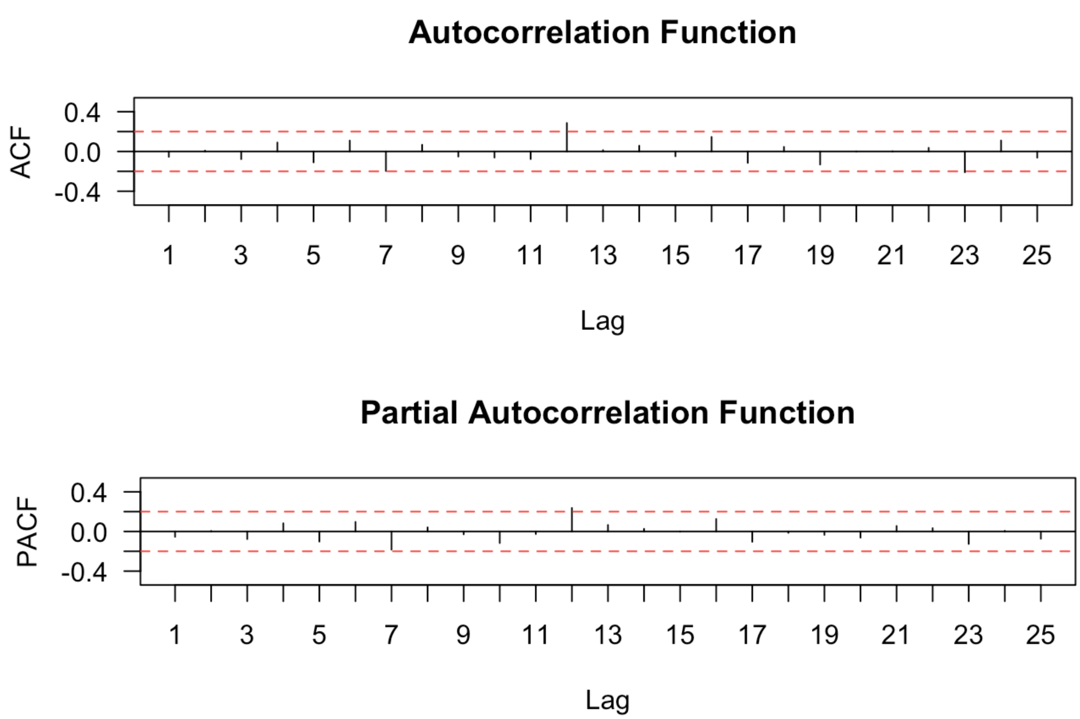

Figure 9: ARMA(p,q) Parameter Grid Search (RMSE)

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	92.39901	95.68758	95.65219	94.36388	93.31652	91.05821	88.34669	87.61045	87.25847	86.04724
[2,]	90.64180	100.88067	87.42562	92.22620	86.91695	87.41688	101.47109	85.78565	85.88436	85.44019
[3,]	90.58195	115.88636	179.37185	107.17336	102.26921	87.35414	98.31313	89.43111	87.02495	91.38327
[4,]	90.37507	89.36075	93.04617	135.90803	106.04821	115.99287	89.21444	99.51944	96.33291	99.15321
[5,]	89.74203	86.79325	86.48071	86.79697	87.68128	87.71589	129.02656	125.04273	96.53153	98.33598
[6,]	89.93517	87.34736	86.92376	90.50506	87.44833	131.39315	104.47757	126.31991	93.88085	84.90459
[7,]	88.77410	94.66051	88.10991	147.18374	132.40931	146.17898	139.42975	89.71582	93.76902	137.54698
[8,]	85.53173	85.43159	86.67199	137.22468	162.73005	83.48060	83.08321	96.98145	100.19755	109.47116
[9,]	85.14055	85.23019	84.84785	86.45289	103.20927	85.04579	105.24499	91.52096	135.45307	87.24750
[10,]	84.78767	84.63329	120.04583	84.23023	134.66566	83.38553	85.75732	105.06556	105.19466	173.28177

Figure 10: ARMA(p,q) Parameter Grid Search (BIC)

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	2519.517	2524.238	2529.590	2534.047	2538.581	2540.968	2544.472	2543.126	2548.449	2553.570
[2,]	2524.234	2516.400	2521.115	2523.900	2529.100	2534.445	2539.602	2548.456	2553.566	2558.316
[3,]	2529.580	2531.561	2531.782	2536.116	2541.095	2539.787	2544.961	2545.299	2550.430	2555.662
[4,]	2533.663	2524.230	2529.066	2533.153	2533.162	2538.183	2544.886	2550.413	2548.035	2560.268
[5,]	2537.504	2529.193	2534.407	2539.711	2541.870	2547.143	2542.597	2547.253	2558.058	2554.970
[6,]	2540.601	2534.320	2539.406	2541.744	2547.077	2545.167	2542.738	2546.628	2549.045	2557.589
[7,]	2543.827	2539.663	2544.758	2542.359	2541.381	2543.654	2545.392	2549.936	2554.396	2556.844
[8,]	2541.606	2546.578	2542.473	2543.822	2546.521	2547.779	2552.931	2547.675	2553.735	2555.191
[9,]	2546.589	2552.093	2557.281	2550.356	2550.785	2552.761	2557.423	2560.070	2555.718	2561.898
[10,]	2551.764	2556.576	2546.473	2557.151	2552.624	2557.805	2562.953	2556.908	2562.161	2565.597

Figure 11: ARMA(7,6) fit on the training set

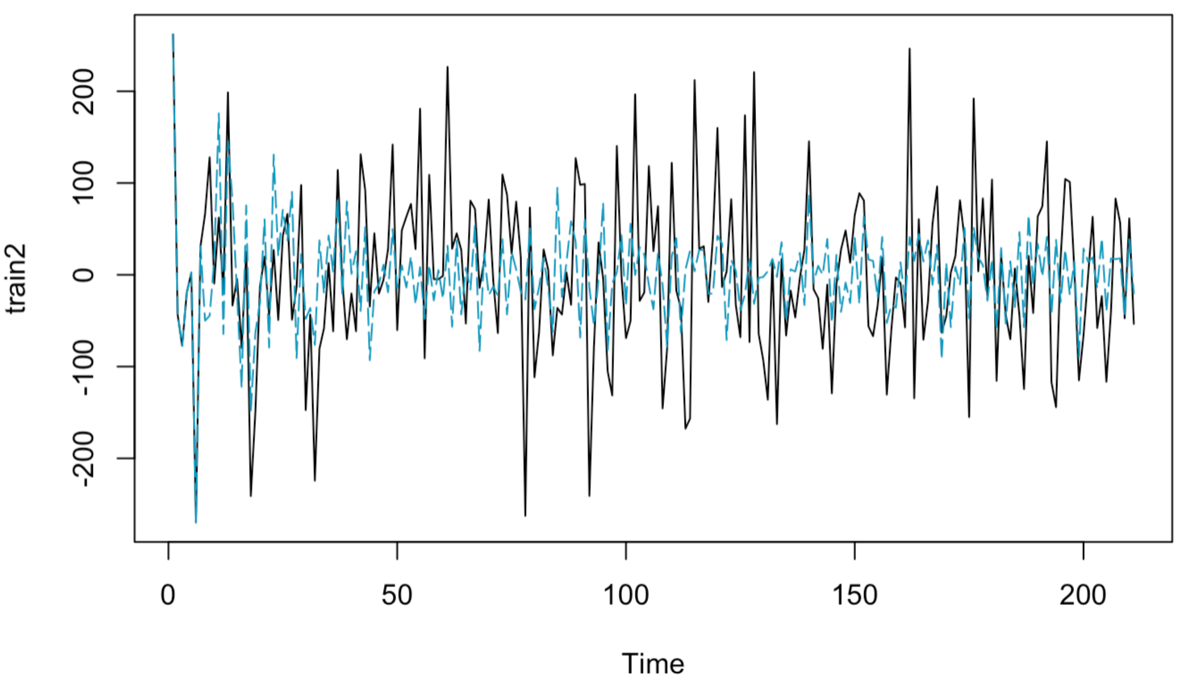

Figure 12: ARMA(1,1) fit on the training set

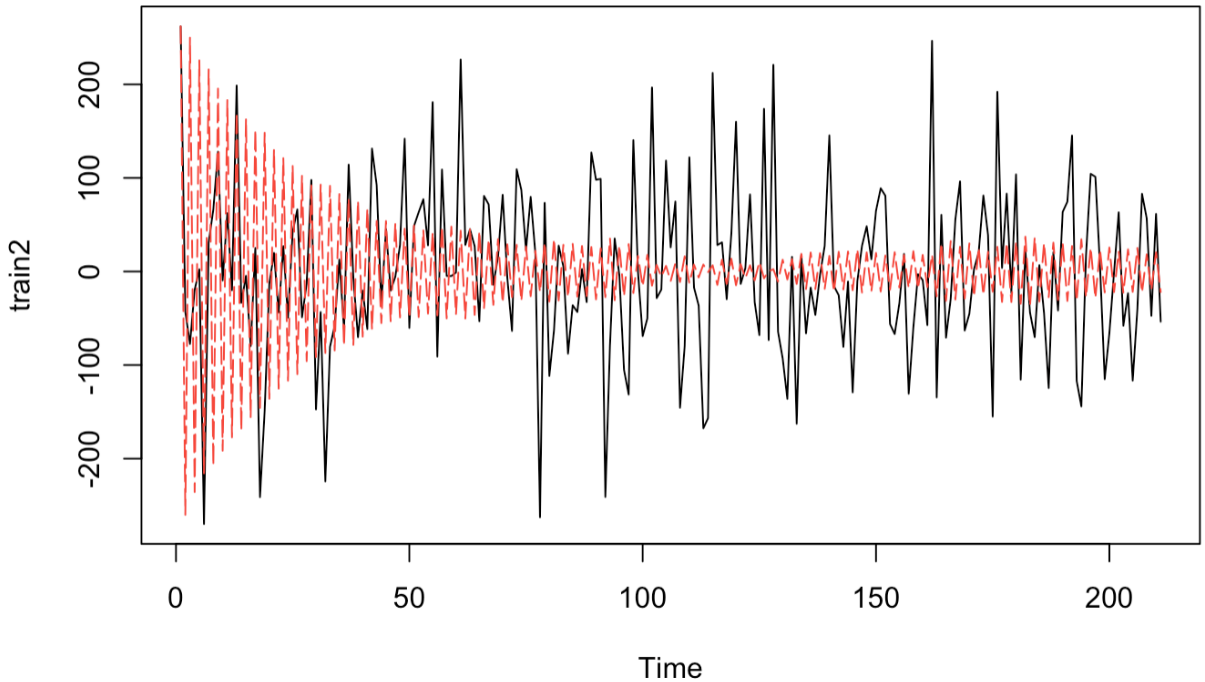

Figure 13: ARMA(1,1) Forecast 1-step look ahead

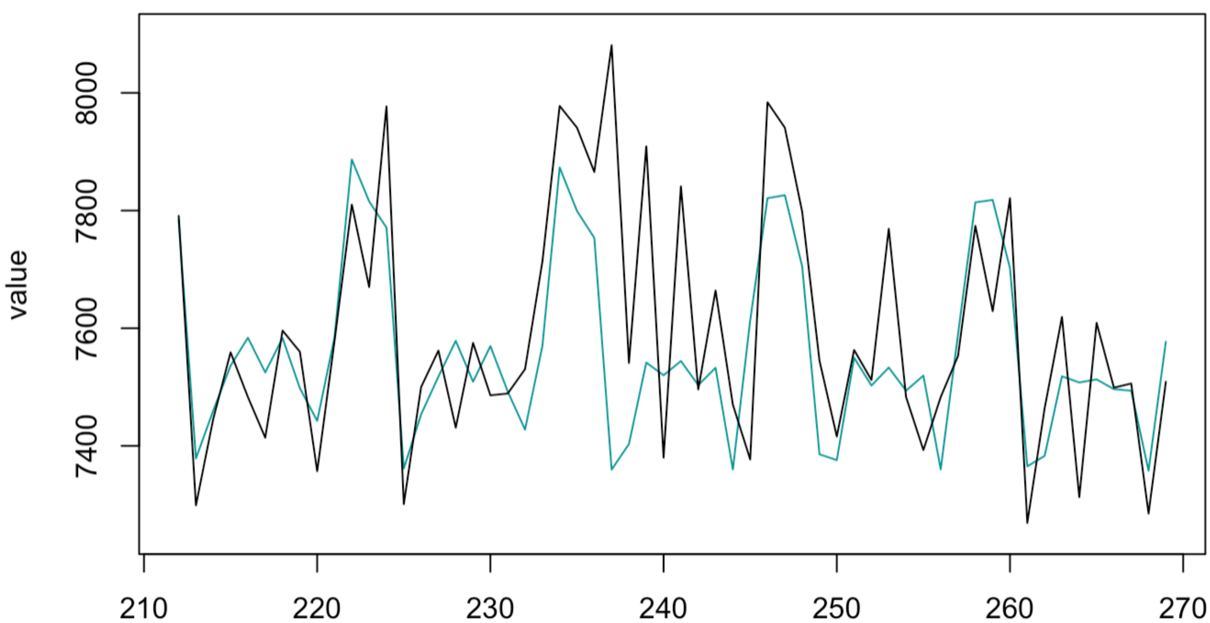

Figure 14: ARMA(1,1) Forecast 12-step look ahead