Time Series Analysis of Chill Data
Everest at 8000m
May 13 to June 1, 2024

PMW

Objectives

- Build a model for the reported “chill” temperature of Mount Everest at 8000m
    - This mountain and elevation was selected arbitrarily from my large dataset of weather forecasts from mountain-forecast.com

- Evaluate the performance of the model compared to the forecasts which appear on mountain-forecast.com
    - The Naive forecasting method will also be used to compare these models

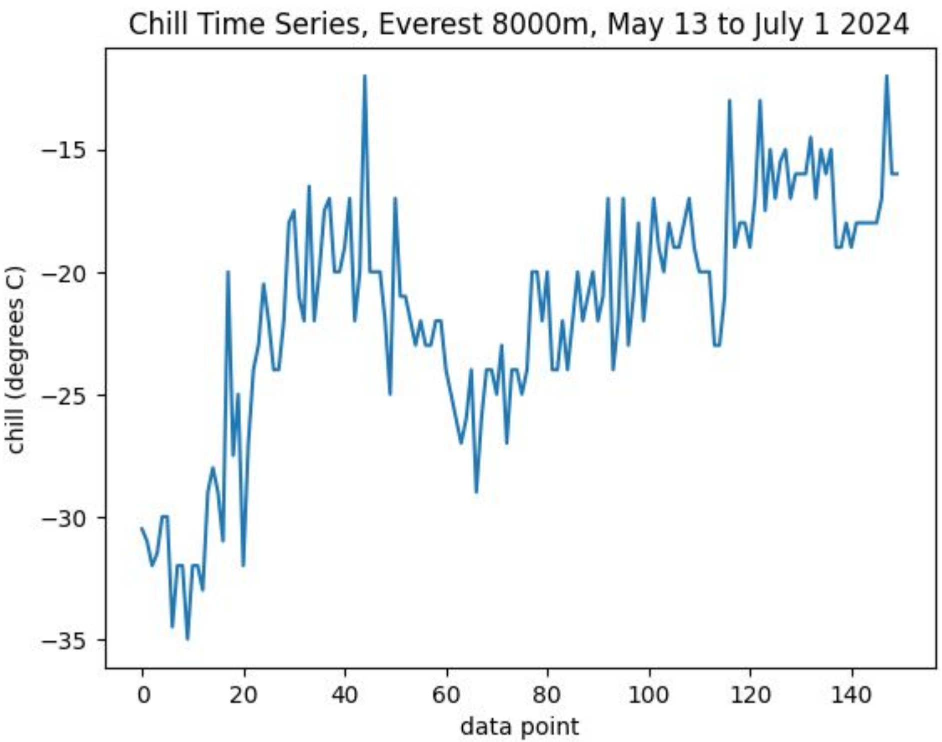
Project overview

- Mage.ai and Airflow were used to orchestrate scraping weather data from mountain-forecast.com
    - The website shows three data time points per day: morning, afternoon and night. So, each data point is roughly six hours apart.

- Google BigQuery was used as the Data Warehouse

- The data were modeled using dbt Cloud

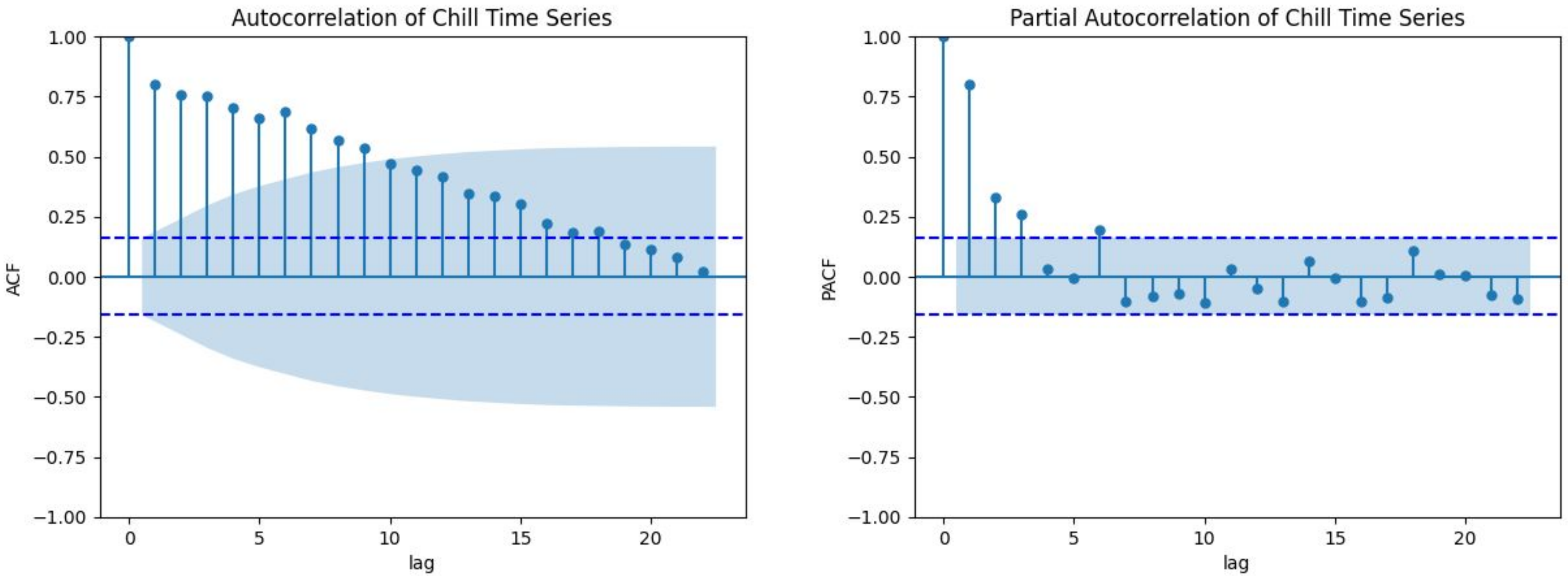
- Details, scripts, notebooks, etc. can be found at the project repo on Github
    - https://github.com/pmwaddell/scrape-mountain-weather-data



An upward trend in the data is clearly evident, and the spikes may indicate some seasonality.

KPSS test (p-value of ~0.09) indicates the time series is stationary around a trend. KPSS and ADF tests on the *differenced* data indicate that it is stationary.

Let's begin by looking at the autocorrelations, which may give us important information about which model to select.



Blue shaded region: 95% CI from Bartlett's formula; Blue dashed lines: 95% CI computed from +/- $1.96/(n^{0.5})$

Clearly, there is a high degree of autocorrelation, which shows "tapering-off" or "dampening" behavior.

Some seasonality is evident at lags of 3 and 6. This makes sense: 3 observations are recorded per day.

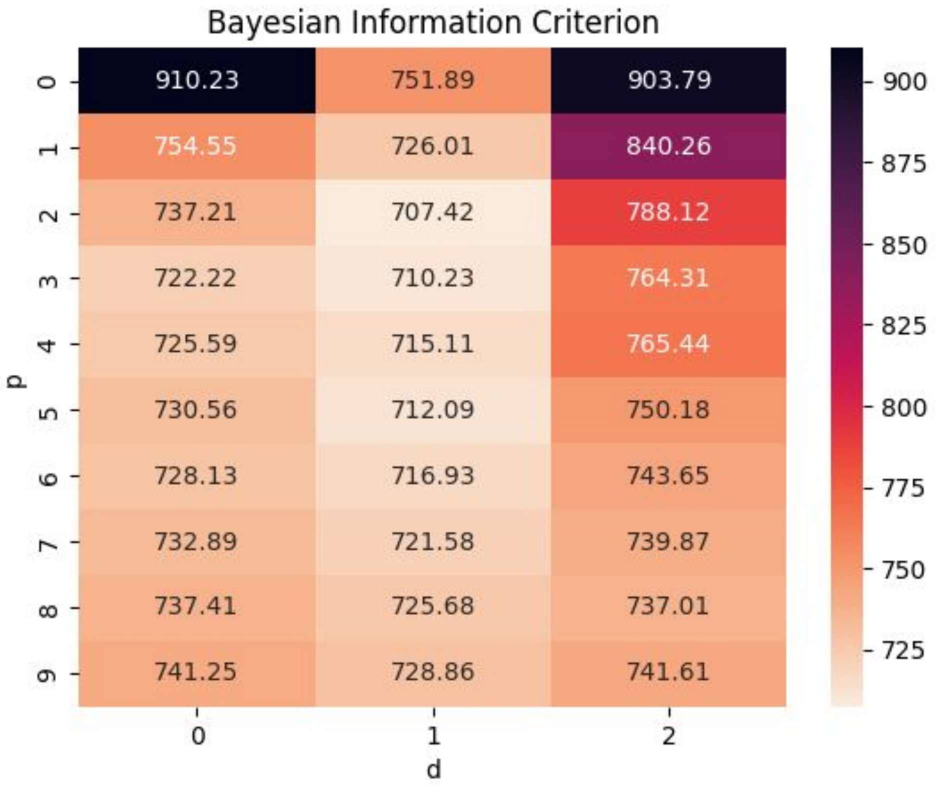
Partial autocorrelations decline quickly. Again seasonality at lag 6 is evident. Taken together, these plots suggest an autoregressive model, potentially AR(3) or AR(6).

Model Selection

- ARIMA(p,d,0) models were trained on the time series, and their Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC) were determined.

- A matrix with p (autoregressive terms) ranging from 0 to 9 and d (degree of differencing) ranging from 0 to 2 was used.

- From these data, ARIMA(2,1,0) and ARIMA(3,1,0) stand out as the best models, consistent with our observations from the autoregression plots.

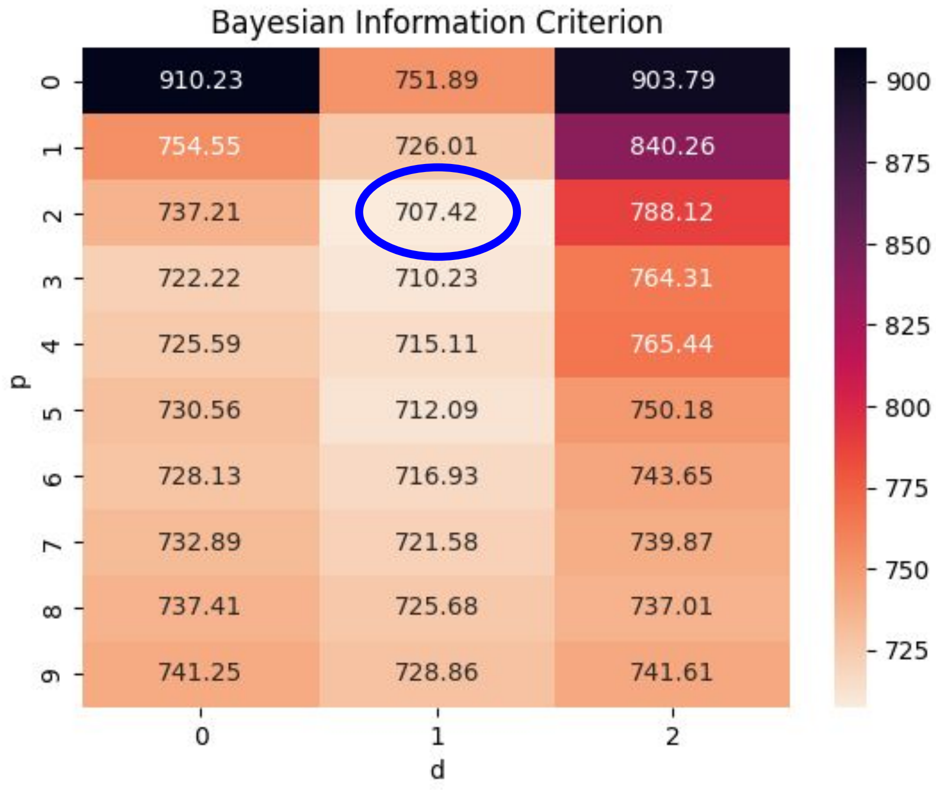


Model Selection

- ARIMA(p,d,0) models were trained on the time series, and their Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC) were determined.

- A matrix with p (autoregressive terms) ranging from 0 to 9 and d (degree of differencing) ranging from 0 to 2 was used.

- From these data, ARIMA(2,1,0) and ARIMA(3,1,0) stand out as the best models, consistent with our observations from the autoregression plots.

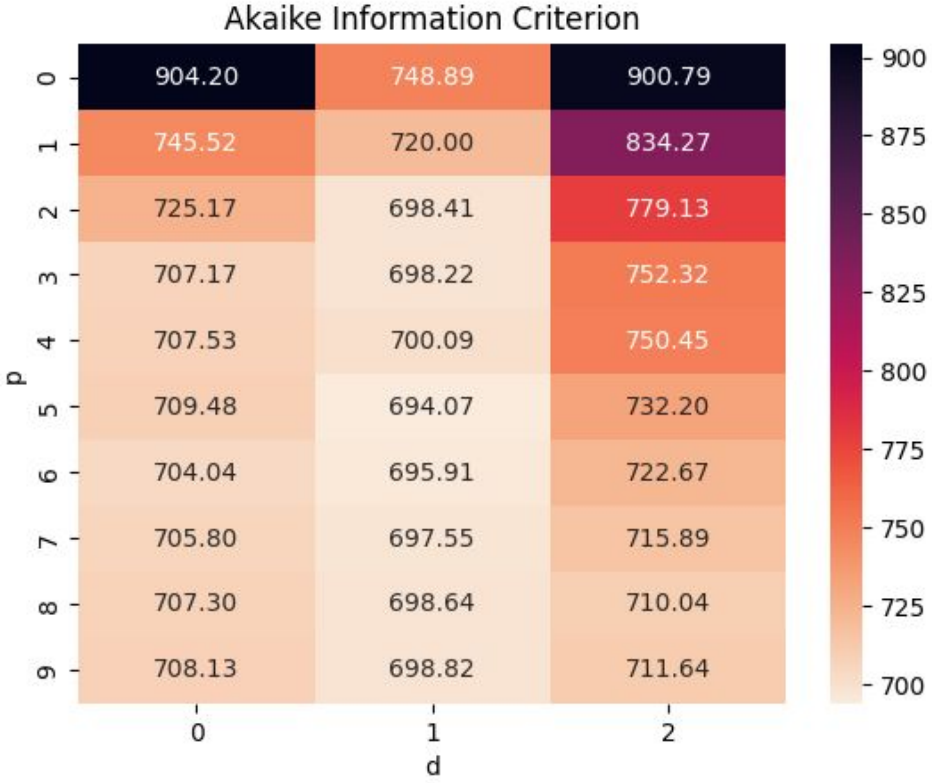


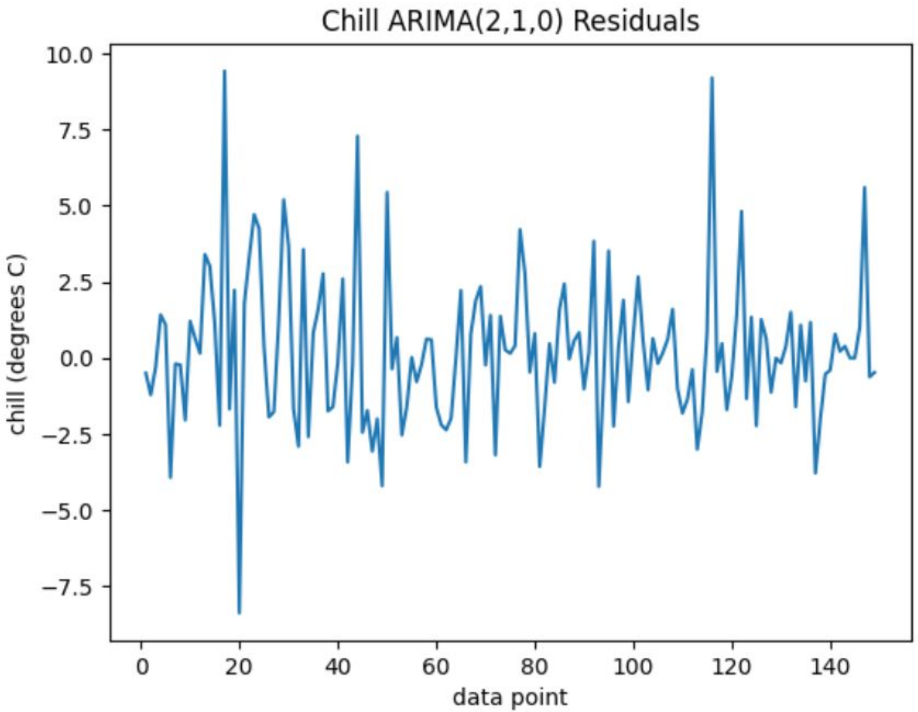
Model Selection

- ARIMA(p,d,0) models were trained on the time series, and their Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC) were determined.

- A matrix with p (autoregressive terms) ranging from 0 to 9 and d (degree of differencing) ranging from 0 to 2 was used.

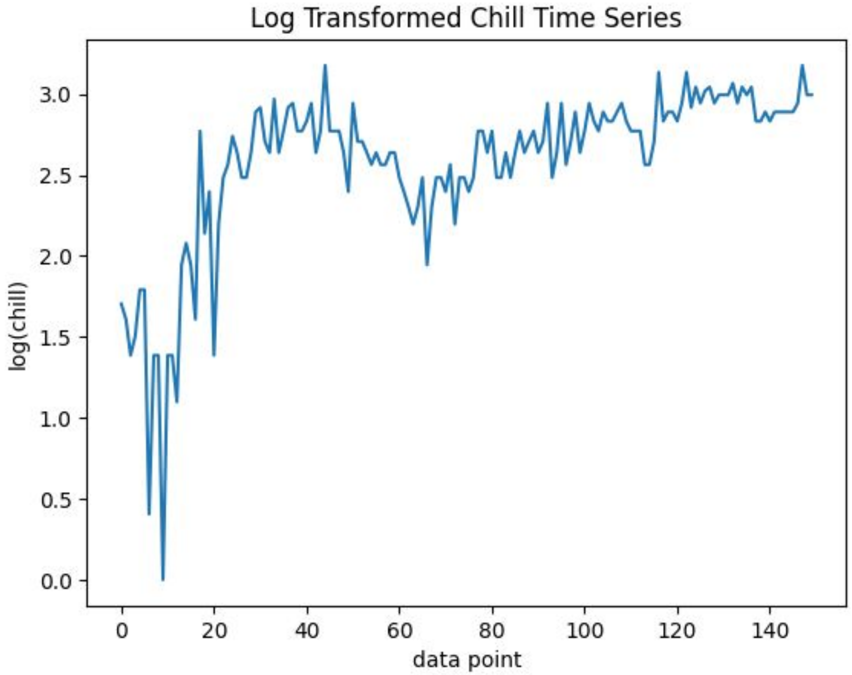
- From these data, ARIMA(2,1,0) and ARIMA(3,1,0) stand out as the best models, consistent with our observations from the autoregression plots.



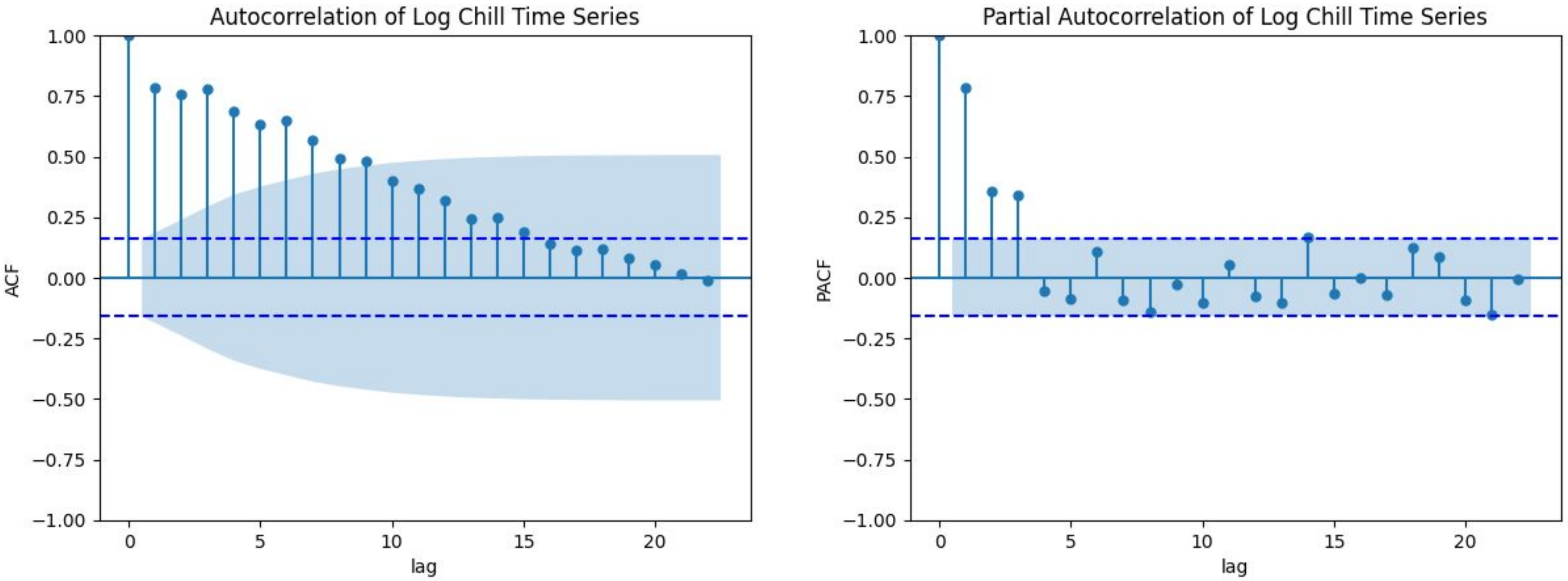


The residuals of the ARIMA(2,1,0) model look mostly like white noise, which is good, but their variance may not be stable; in other words, heteroscedasticity may be evident.

To evaluate this, a Breusch-Pagan test gives a p-value of 0.058. Given this is greater than the significance level of 0.05, a variance-stabilizing transformation is necessary before applying the ARIMA model.



First, the data were translated so that the lowest value was equal to 1. Then, the data were log-transformed to give the time series above. Now, let's examine its autocorrelations and redo our model selection process.



Blue shaded region: 95% CI from Bartlett’s formula; Blue lines: 95% CI computed from +/- $1.96/(n^{0.5})$

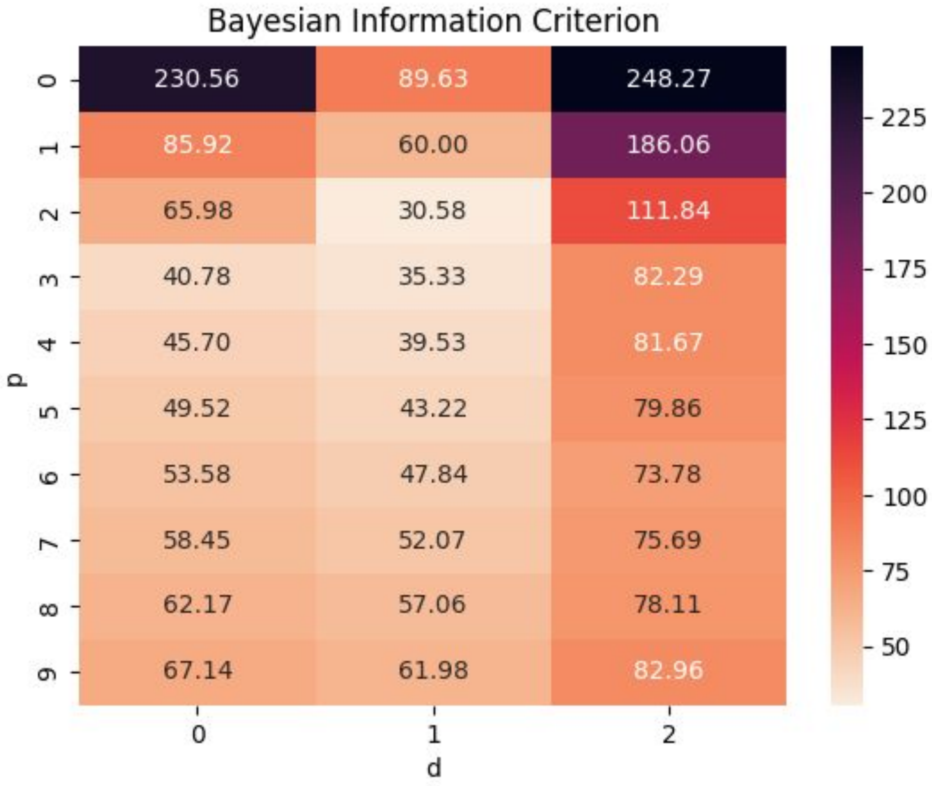
The autocorrelations look very similar to the original time series. However, the partial autocorrelation at lag 6 is not as prominent here.

Model Selection: transformed data

- ARIMA(p,d,0) models were again trained on the time series, and their Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC) were determined.

- A matrix with p (autoregressive terms) ranging from 0 to 9 and d (degree of differencing) ranging from 0 to 2 was used.

- From these data, ARIMA(2,1,0) stands out as the best model, similar to the original time series.

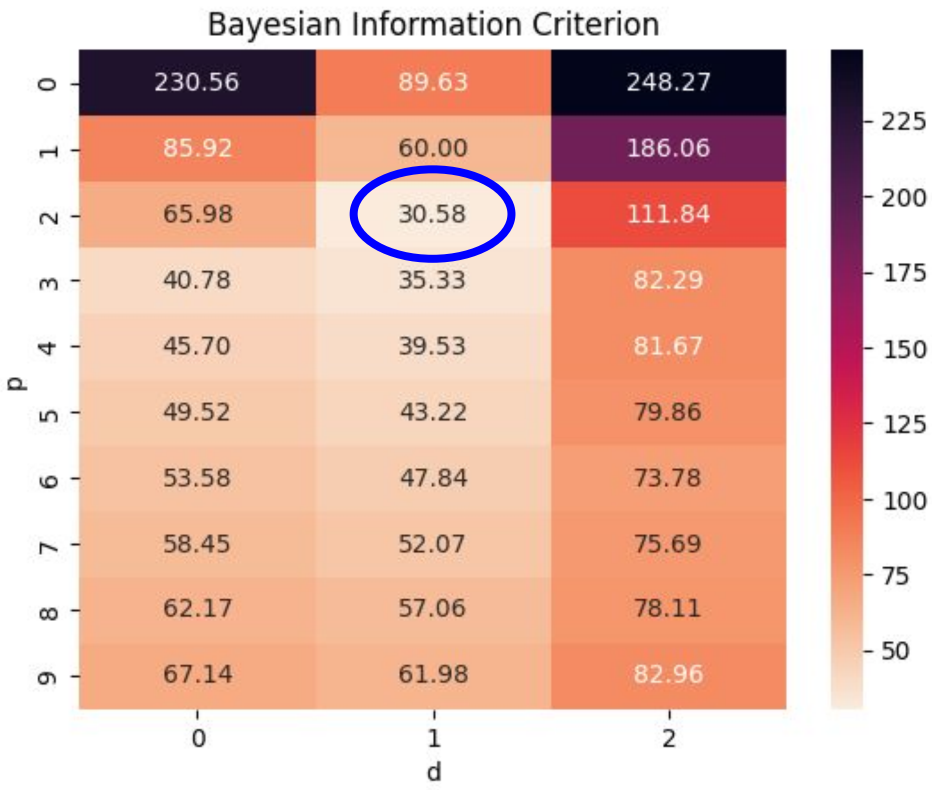


Model Selection: transformed data

- ARIMA(p,d,0) models were again trained on the time series, and their Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC) were determined.

- A matrix with p (autoregressive terms) ranging from 0 to 9 and d (degree of differencing) ranging from 0 to 2 was used.

- From these data, ARIMA(2,1,0) stands out as the best model, similar to the original time series.

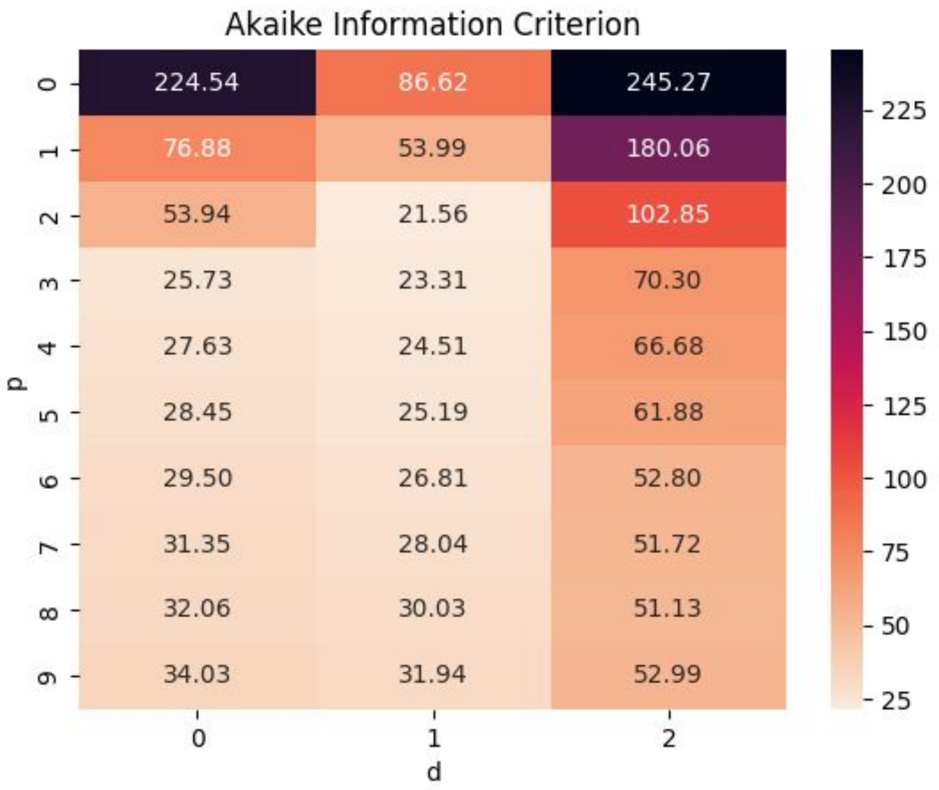


Model Selection: transformed data

- ARIMA(p,d,0) models were again trained on the time series, and their Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC) were determined.

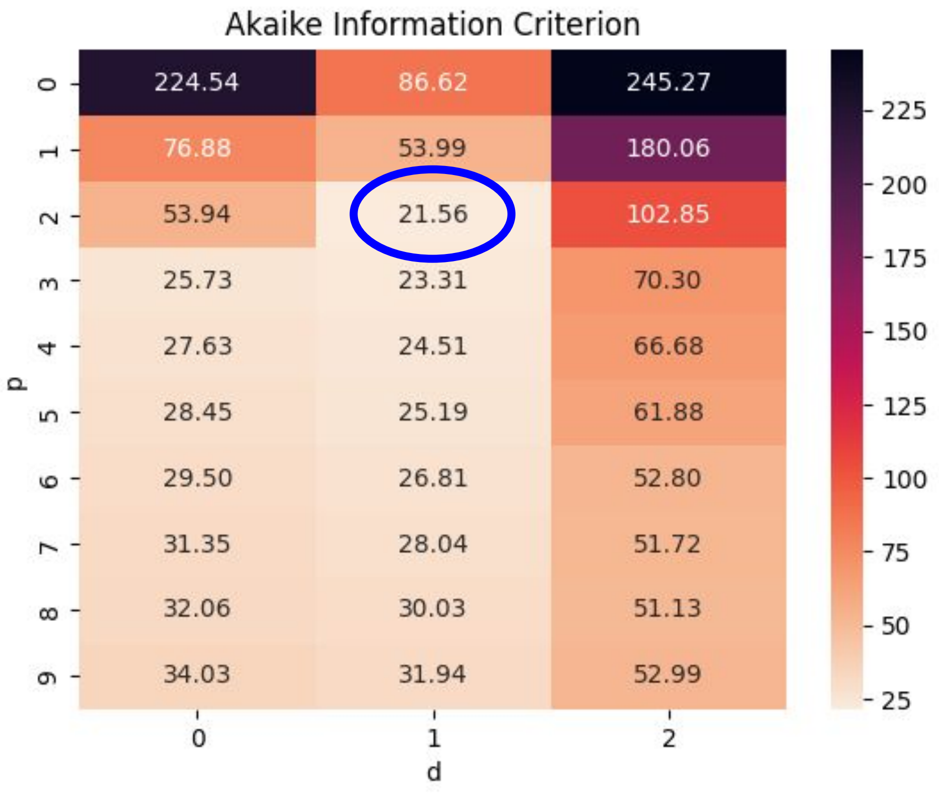
- A matrix with p (autoregressive terms) ranging from 0 to 9 and d (degree of differencing) ranging from 0 to 2 was used.

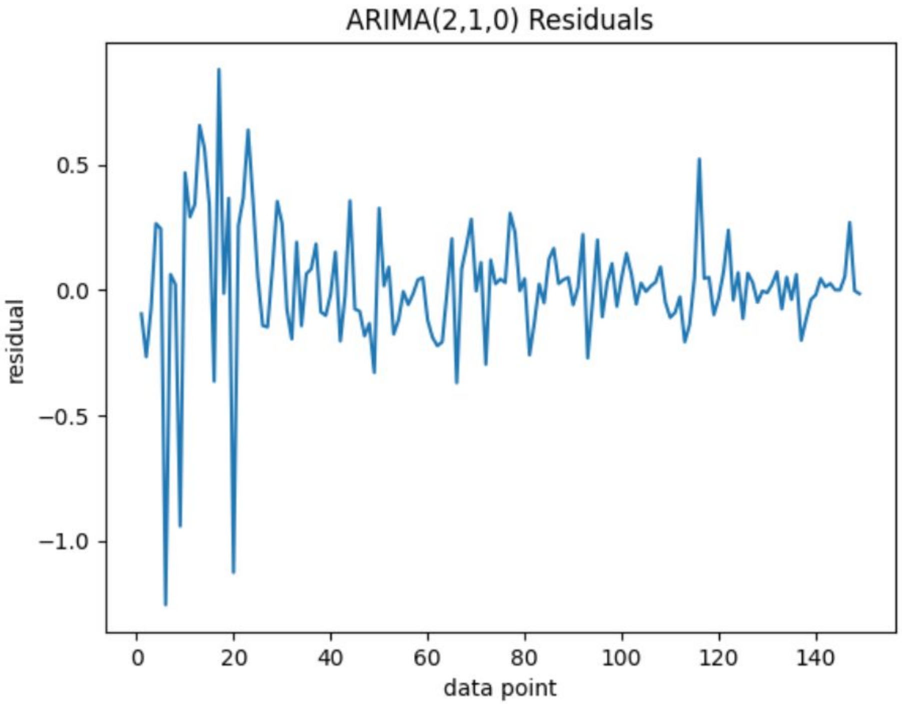
- From these data, ARIMA(2,1,0) stands out as the best model, similar to the original time series.



Model Selection: transformed data

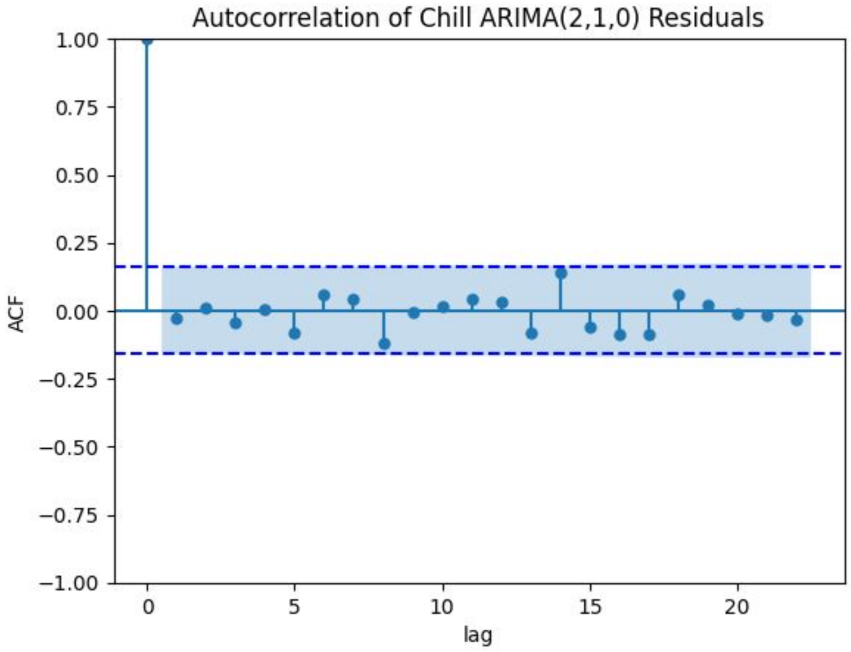
- ARIMA(p,d,0) models were again trained on the time series, and their Akaike Information Criteria (AIC) and Bayesian Information Criteria (BIC) were determined.
- A matrix with p (autoregressive terms) ranging from 0 to 9 and d (degree of differencing) ranging from 0 to 2 was used.
- From these data, ARIMA(2,1,0) stands out as the best model, similar to the original time series.

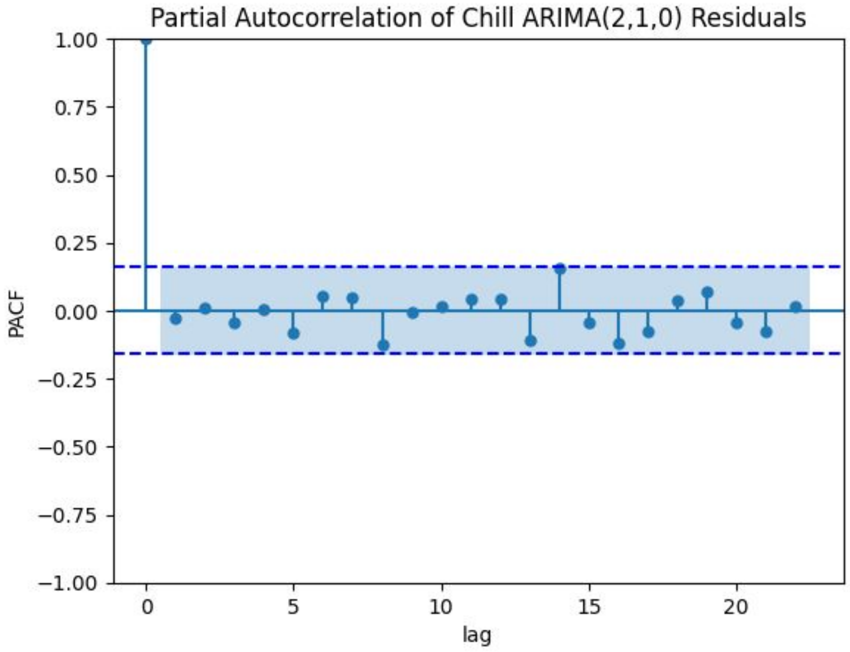




Here are the ARIMA(2,1,0) residuals for the transformed time series. Note how much smaller the scale on the y-axis is compared to the original time series’ residuals, which ran from about -8 to +9.

Going forward, ARIMA(2,1,0) will refer to the log-transformed ARIMA(2,1,0) model.



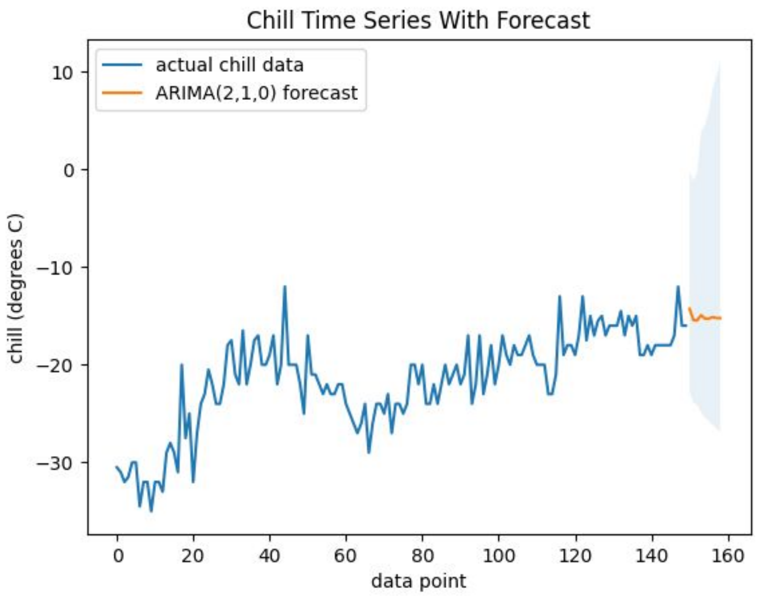


Blue shaded region: 95% CI from Bartlett's formula; Blue dashed lines: 95% CI computed from +/- $1.96/(n^{0.5})$

Happily, the residuals for the ARIMA(2,1,0) model on the transformed data show no significant autocorrelation! Results of a Ljung-Box test agree with this assessment (all p-values > 0.7).

Breusch-Pagan test gives a p-value of less than $0.1*10^{-4}$, strongly indicating that the residuals have stable variance, or are homoscedastic.

Forecasting with the Log-Transformed ARIMA(2,1,0) Model



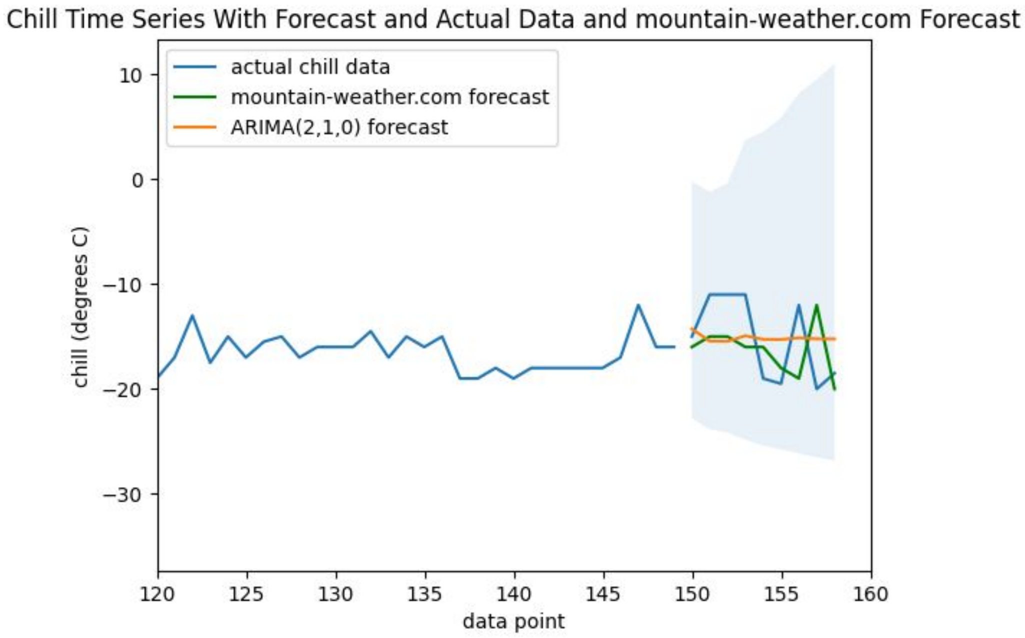
The orange line shows the forecast from the ARIMA(2,1,0) model for 9 data points after the end of the time series, or for the following three days.

The blue shaded region represents a 95% confidence interval for the forecast.

- Mean ARIMA RMSE: 3.81

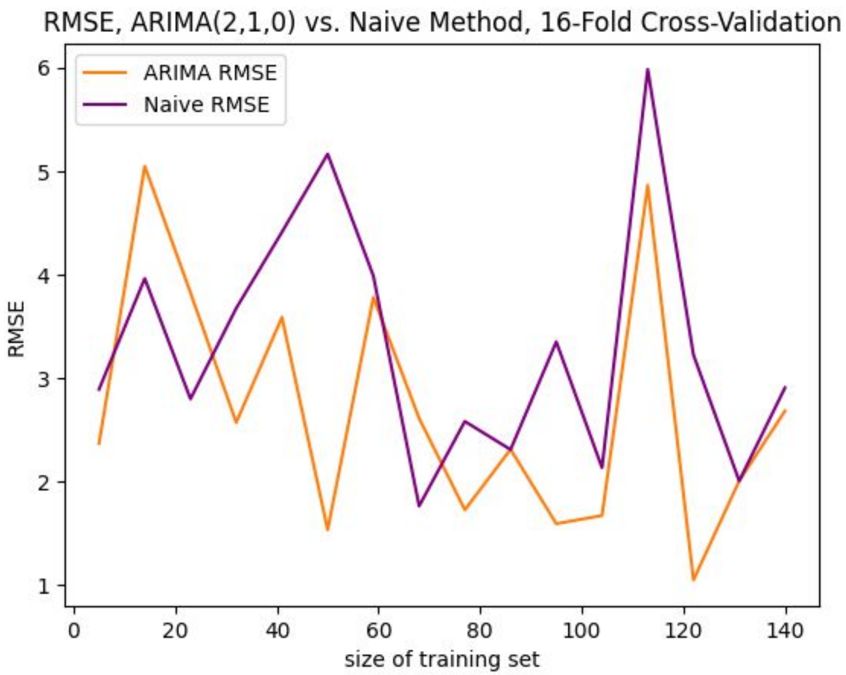
- Mean mountain-weather.com forecast RMSE: 4.53

- Naive model RMSE: 3.88



This view shows the very end of the time series, followed by the forecast region for the next nine points. The orange line is for the ARIMA forecast, the green line shows the forecast from mountain-weather.com posted at the time of the end of the original time series, and the blue line shows the actual chills measured.

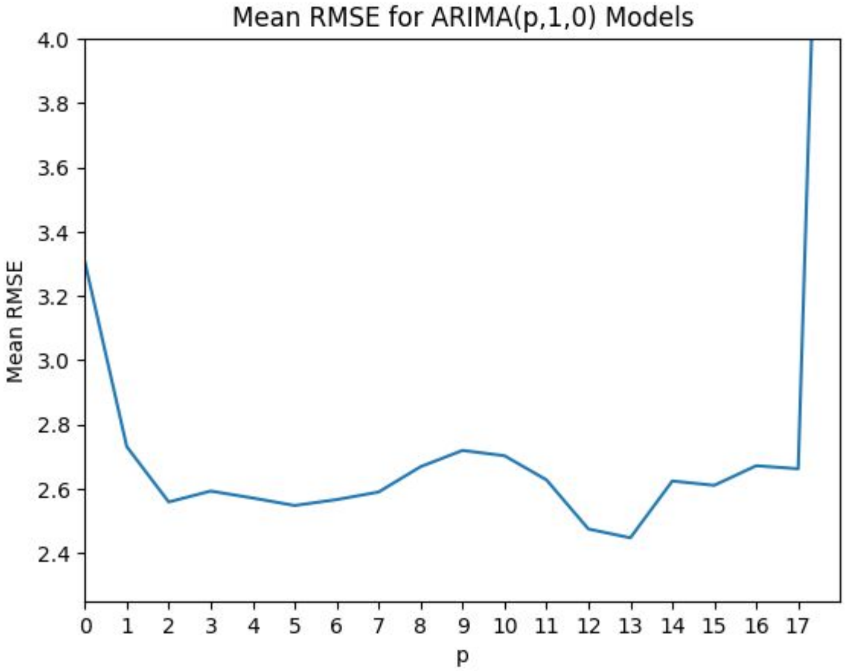
While the mountain-weather.com forecast matches some parts of the shape of the following chill data well, it actually exhibits the highest RMSE over this period. By contrast, the ARIMA model and the Naive method (meaning, predicting the value of the last point in the training set for the duration of the test set) perform a bit better, comparable to one another.



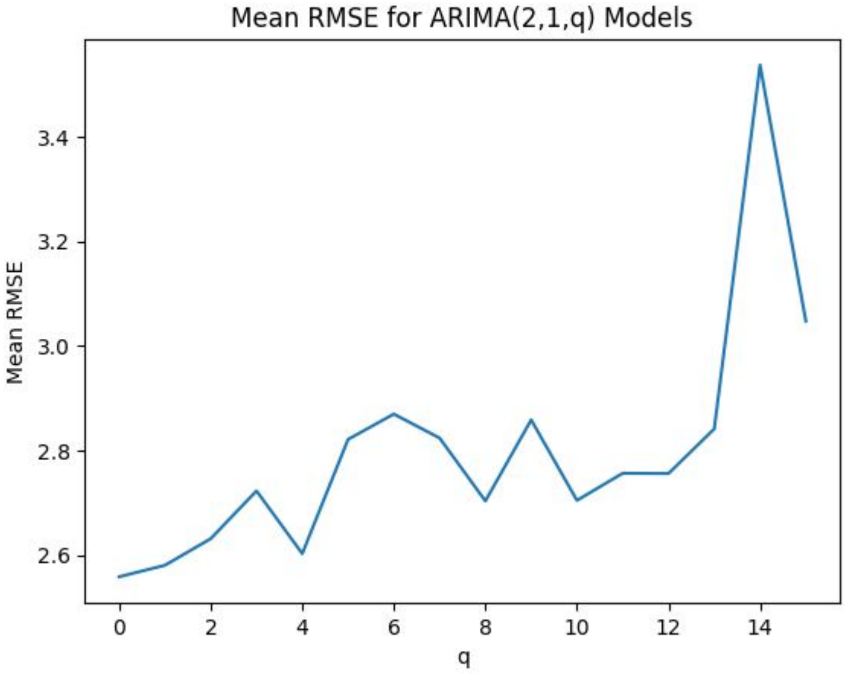
Mean ARIMA RMSE: 2.70; mean Naive model RMSE: 3.21

To get a more proper and thorough picture of the performance of these models, I looked at a 16-fold cross-validation approach using a rolling-origin approach and a training set of size 9. The ARIMA model appears to consistently outperform the Naive model, and its mean RMSE over these trials is lower.

Checking Model Selection by Cross-Validation



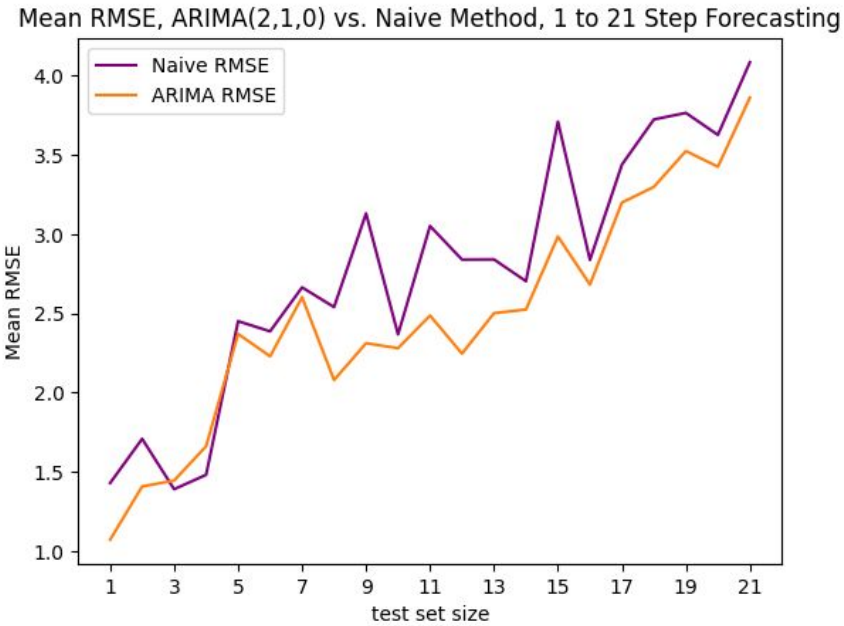
This plot shows RMSE from the same 16-fold cross-validation for ARIMA(p,1,0) models, with p from 0 to 18. The data here validate my choice of ARIMA(2,1,0). Although 11 to 13 seem a little better by this measure, I favor keeping the model much simpler. After 17, dramatic overfitting seems to occur, which increases the mean RMSE a lot.



Similarly, I decided to determine whether adding moving average terms would lead to any improvement. Therefore I evaluated a series of ARIMA(2,1,q) models, with q from 0 to 15. Their mean RMSEs from the 16-fold CV are shown in the plot. Clearly they do not lead to any improvement and actually make the model worse.

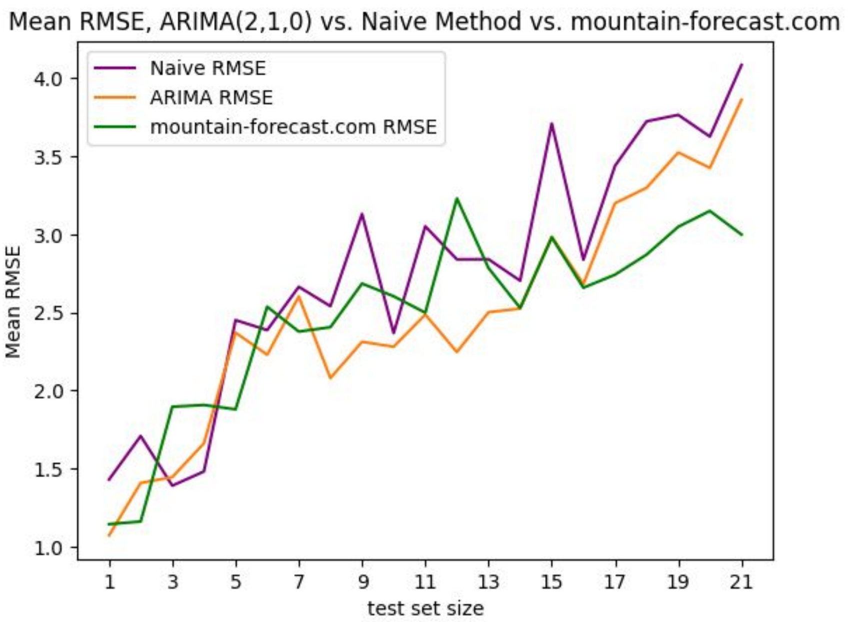
So, let’s stick with the ARIMA(2,1,0) model.

Forecast Accuracy as a Function of Test Set Size



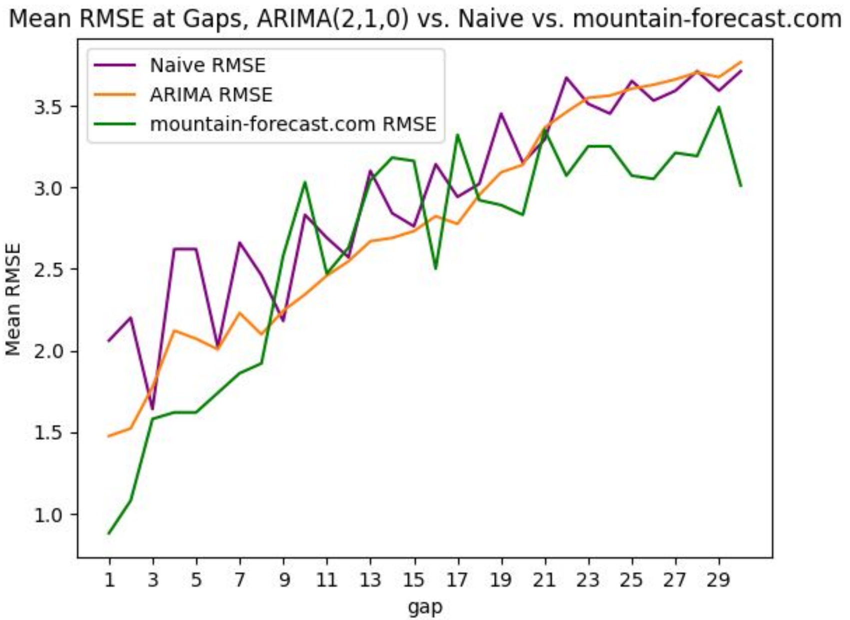
How does the ARIMA model perform on larger test sets, i.e. test sets which extend further into the future? The above plot shows the mean RMSE for this and the Naive model as a function of test set size. The ARIMA model consistently outperforms the Naive model here.

Forecast Accuracy as a Function of Test Set Size



This plot adds the mean MRSEs from the mountain-forecast.com forecasts in. The ARIMA model seems to outperform the mountain-forecast.com forecasts for “medium” test set sizes, i.e. from around 8 to 13, but above 16 mountain-forecast.com appears to be significantly better than both the ARIMA model and the Naive model.

Forecast Accuracy at Different Gaps Between Training and Test Sets

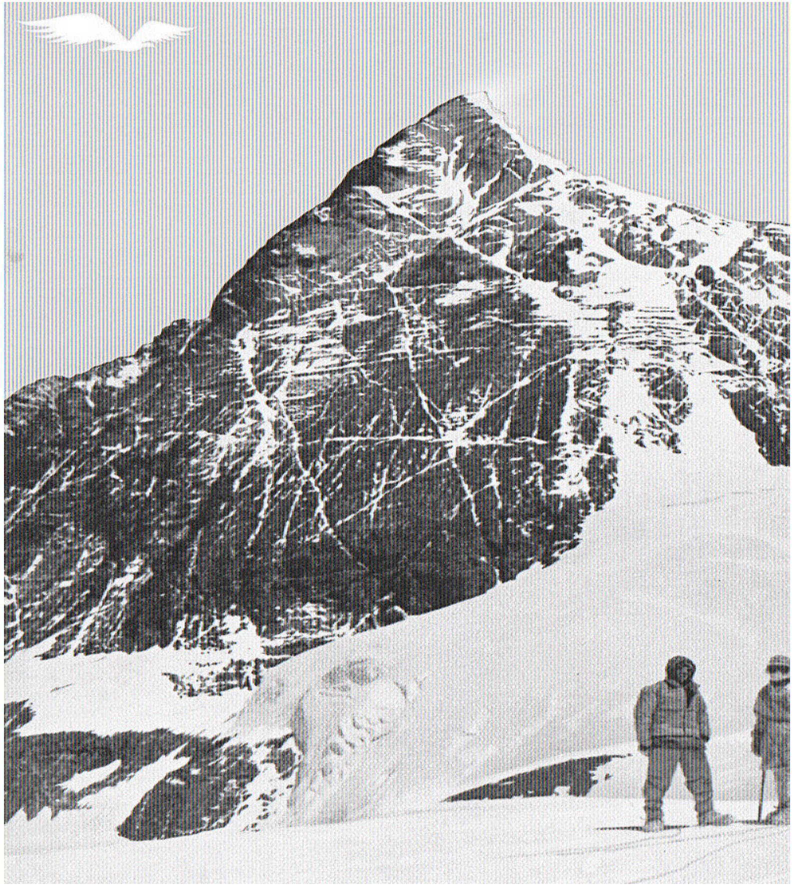


This plot shows mean RMSE for tests of size 1 as a function of the gap (1 to 30) between the test point and the training set. The RMSE values are averaged over fifty training sets, comprising the first 64 points to the first 113 points in the time series.

From these data, we can see that the mountain-forecast.com forecasts perform best at the nearest (1-9) and farthest (20-30) points. However, for intermediate points (~9-15), the ARIMA model appears to perform best.

Conclusions

- For the time series of chill temperatures at 8000m on Mount Everest, a log-transformed ARIMA(2,1,0) was the best performing of my models.
- This model proved to be consistently better than the Naive model, and appeared to be competitive with forecasts from mountain-forecast.com, especially over the range of 3 to 4 days ahead.
- Further studies could look at whether ARIMA(2,1,0) or similar models are effective across different mountains and elevations, or whether other models are better, and whether any patterns in this regard can be found.



Sources

- Weather and forecast data was scraped from mountain-forecast.com

- Images:
  - https://en.wikipedia.org/wiki/Mount_Everest#/media/File:Everest_North_Face_toward_Base_Camp_Tibet_Luca_Galuzzi_2006.jpg
  - https://en.wikipedia.org/wiki/Mount_Everest#/media/File:Condor_Films_1952.jpg