Time Series Project

Chris Corona

2022-12-12

Introduction

The Gallatin River is an important resource for the community of Bozeman, MT. It provides water for irrigation of farming businesses and families. It is a renowned fly fishing destination that brings lots of tourist dollars to the area. Low water and droughts will negatively impact the agriculture and tourism economic sectors. River flows are also an indicator for the filling of local reservoirs with drinking water. It is known that the drinking water supply in Bozeman is quickly shrinking with its growing population. With more growth and more drought, the water supply in this region may be insufficient.

Figure 1: Agriculture and tourism are important industries in Montana.

On the flip side, too much water leads to flooding. We have seen first hand the damage from recent historic flooding. The spring of 2022 saw an above-average snowpack combined with unusually wet months. The community of Red Lodge, MT suffered massive destruction and the road through the north entrance of Yellowstone National Park was totally destroyed from river flooding, shown in Figure 2. Predicting river flow is important to warn our communities of imminent danger.

Figure 2: Photos of Yellowstone River and tributaries during spring 2022 flood

My research question asks: can I predict the hourly water flow of the Gallatin River using previous water flow measurements and data from three weather stations at various locations upstream measuring snow depth, precipitation accumulation, and temperature?

The Data Set

The data set in this project is a compilation of publicly available data from the USGS (https://waterdata.usgs.gov/monitoring-location/06043500/) and SNOTEL websites (https://wcc.sc.egov.usda.gov/reportGenerator/). I downloaded data for water flow of the Gallatin River measured at the mouth of the Gallatin Canyon, shown in Figure 3 as the yellow circle, and weather data collected from three SNOTEL weather stations within the Gallatin River watershed (Carrot Basin, Shower Falls, and Lone Mountain), shown in Figure 3 as red stars.

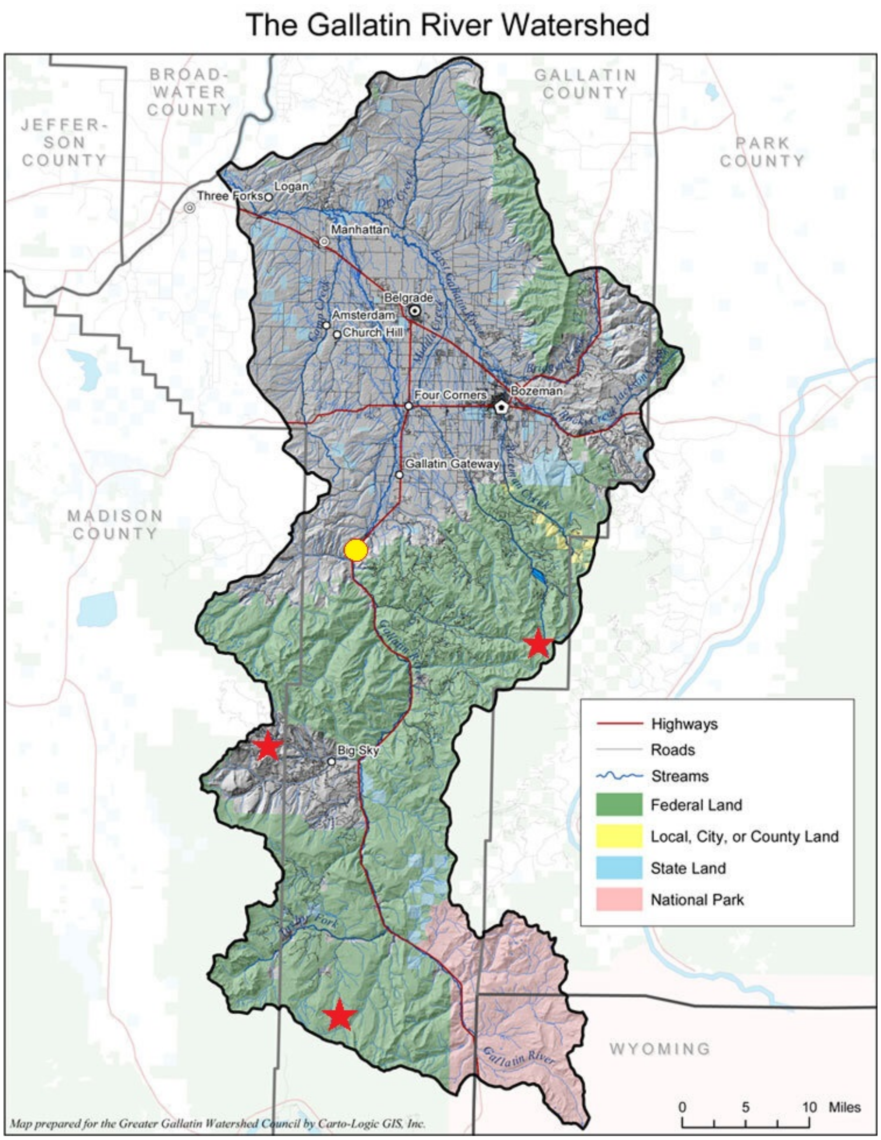


Figure 3: Map of the Gallatin River watershed with measurement locations for the USGS waterflow sensor (yellow circle) and three SNOTEL weather stations in the mountains (red stars).

An initial look at the raw data set shows that there are significant amounts of missing data for all measurements. Starting with the response variable `discharge`, we see that the missing data occurs in large blocks (a few days or weeks at a time). Figure 4 shows a typical set of missing blocks for one particular year. I imputed these missing values, shown in red, with a Holt-Winters model trained on the data leading up to the missing block. Visually, the imputations look remarkably similar to the existing data. This is not ideal because we will eventually validate the forecasting models with these observations (including the imputations). But given the nature of this data set, it might be the best we can do without entirely removing those observations. Patterns of missingness also occur in the predictor variables. In total I imputed 28,648 values or 3.3% of the data set (1,787 or 2% in the response and 26,861 or 3.4% across the nine predictor variables).

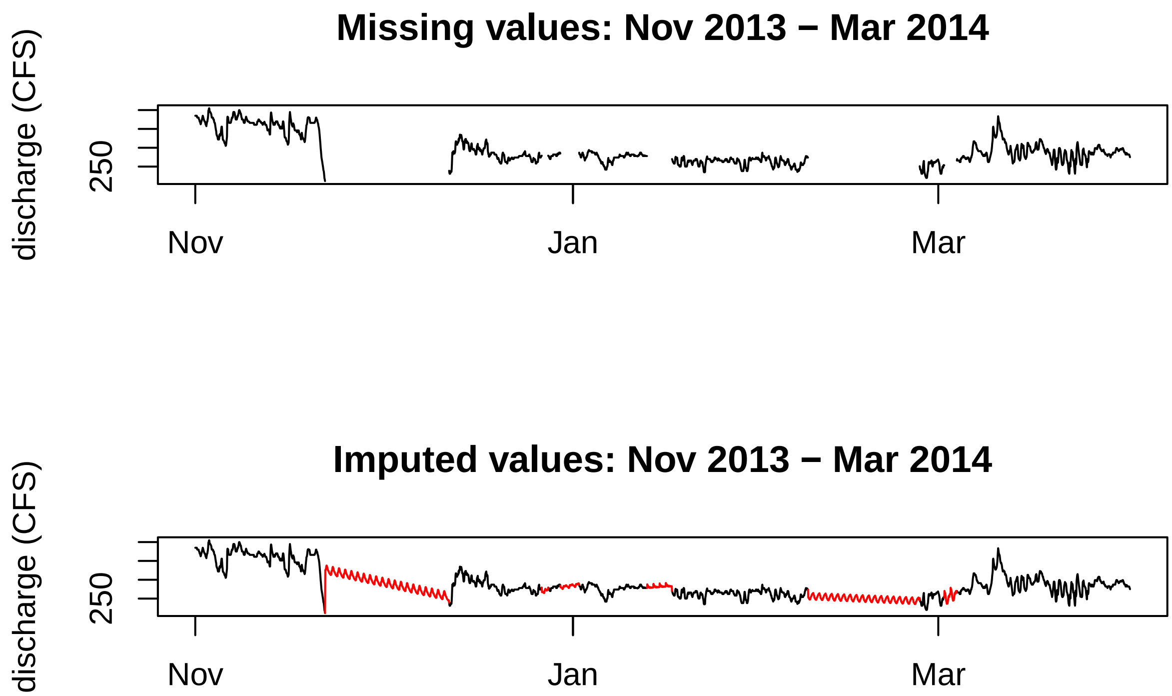


Figure 4: Large blocks of missing data in the response variable and imputed values in red.

In Figure 5, we can see that there is a lot of redundancy in the three weather measurements from each of the three SNOTEL sites. Having all three weather stations is unnecessary and introduces complications from multicollinearity, so I will choose just the Carrot Basin measurements.

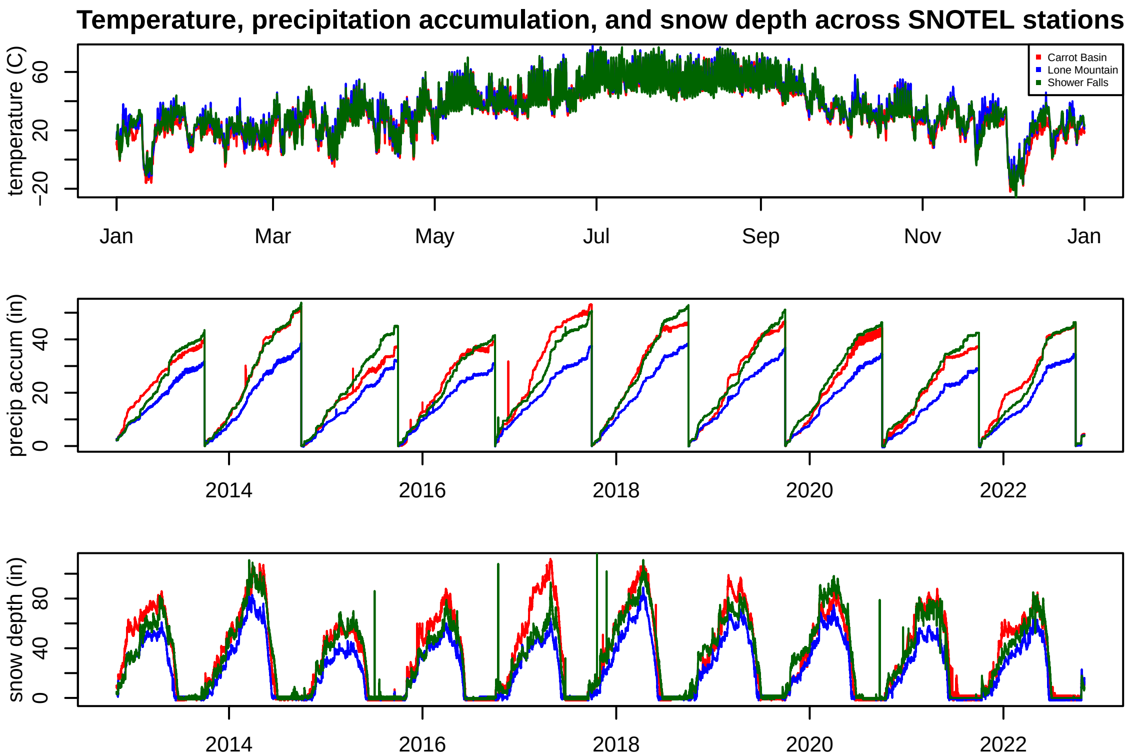


Figure 5: Comparing predictors across the different SNOTEL stations.

After all the cleaning described so far, the data set in its final form includes the following variables: (1) the date and time of the measurements, (2) air temperature in degrees Celsius at Carrot Basin, (3) precipitation accumulation from the past 48 hours in inches at Carrot Basin, (4) snow depth in inches at Carrot Basin, and (5) water discharge in cubic feet per second of the Gallatin River. I collected 10 years of hourly data for all measurements from 2012-11-03 00:00:00 PDT to 2022-11-02 00:00:00 PDT. In total, there are 87,648 observations and 5 variables.

Exploratory Data Analysis

Figure 6 shows a time series plot of all five variables through a typical year (note: the values are scaled to show relationships among variables). This is a visual check for any obvious correlation between the predictors and the response. We can see that precipitation accumulation is positively correlated and snow depth is negatively correlated with discharge. If temperature has any correlation with discharge, it is weak. There also appears to be some correlation among the predictors: temperature and snow depth are negatively correlated, while precipitation accumulation and snow depth are positively correlated. As a final exploratory check, we look at any potential autocorrelation. Figure 7 shows the ACF and PACF plots of the residuals from a mean-only model. The ACF plot shows definite autocorrelation, while the PACF indicates that maybe an AR(2) model is appropriate with some seasonal components.

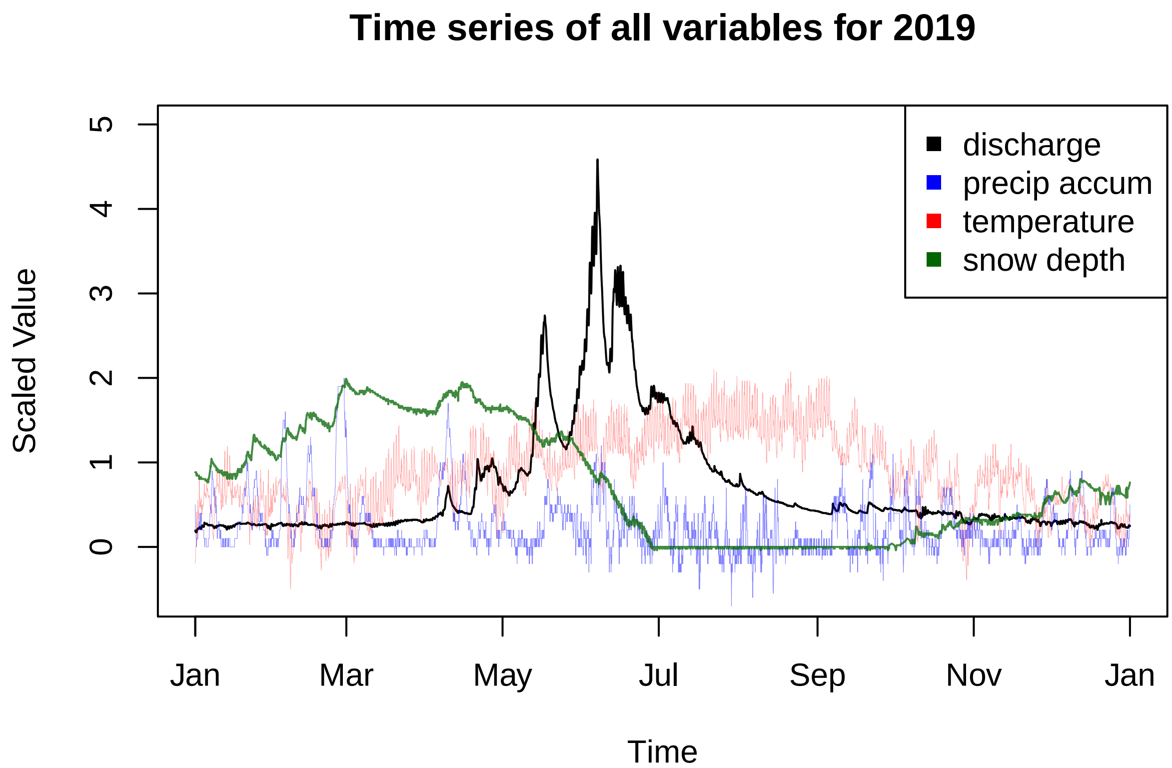
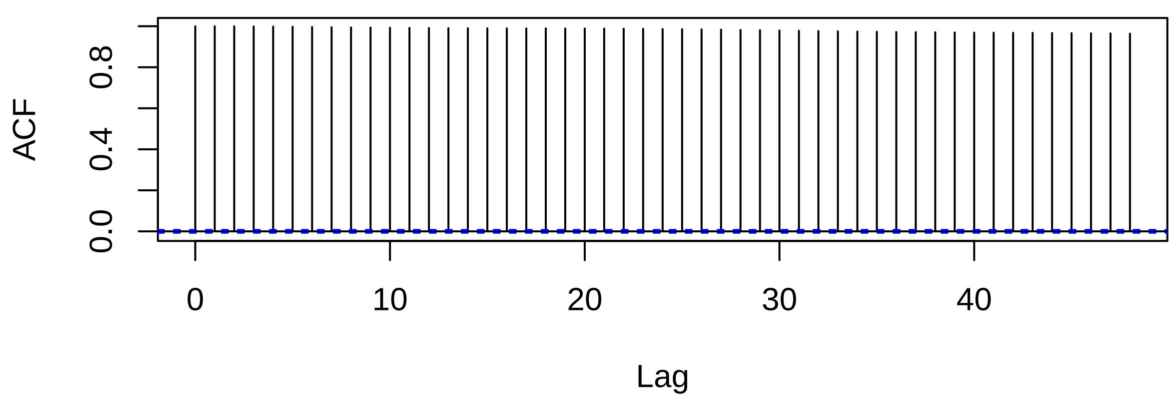


Figure 6: Plot of time series for all variables through a typical year (note: the values are scaled to show relationships among variables).



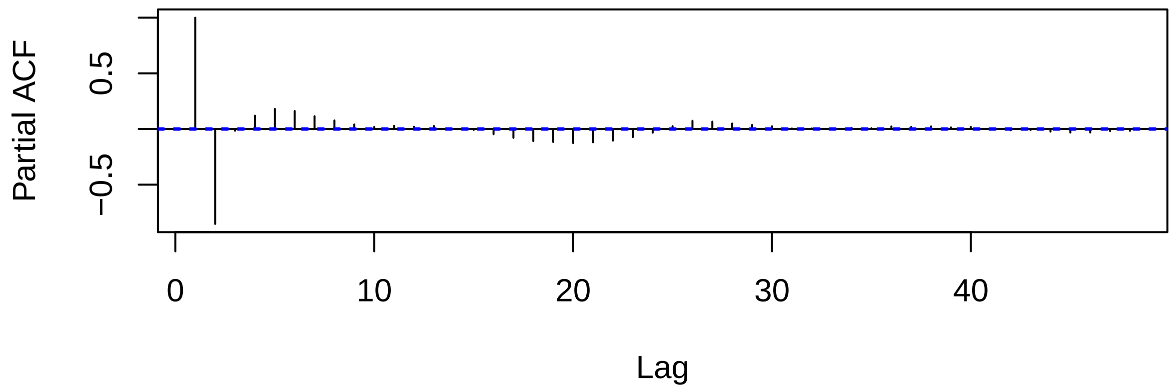


Figure 7: ACF and PACF plots of the residuals from a mean-only model.

Methods

In this analysis I compare various Holt-Winters and ARIMA models, where a mean-only model is the benchmark. The Holt-Winters models are exponentially weighted moving averages with a tuning parameter α that controls the amount of smoothing. I implement a basic grid search for $\alpha \in \{0.1, 0.25, 0.5, 0.75, 0.9\}$ with smaller values corresponding to more smoothing, less weight on the last observed value, and larger values corresponding to less smoothing, more weight on the last observed value. The ARIMA models can have auto-regressive (AR), differencing (I), and/or moving average (MA) components. I'm fitting eight ARIMA models, one for every possible combination of my three covariates shown in Table 1. I let the `auto.arima()` function fit the best model each time. RMSE is used as the loss function for all models.

name	covariates
arima1	none
arima2	temp
arima3	precip accum
arima4	snow depth
arima23	temp + precip accum
arima24	temp + snow depth
arima34	precip accum + snow depth
arima234	temp + precip accum + snow depth

Table 1: ARIMA model name key

After a few initial designs for cross-validation, I decided on the following. I will train each model on 10 consecutive days worth of data (240 observations) and test on the 11th day's data (24 observations). The cross-validation loop will step through all 10 years worth of data one day at a time such that I have test predictions for every day (excluding the very first 10 days in the data set). I chose 10 days of training data because more is too slow and less pushes the boundary of not enough data points for the 2 and 3 covariate ARIMA models. I chose 1 day ahead of prediction because it seems to be a fairly practical amount for this analysis. Other cross-validation setups could be the focus of future work (eg. less training data using only Holt-Winters models or expanding to further ahead forecasting).

Results

After training all the models, I computed the RMSE over all 10 years worth of data, shown in Table 2. It is clear that all the models significantly out-perform the benchmark mean-only model. The RMSE for these models is roughly between 100 and 150 cubic feet per second. Considering that the median water flow is around 400 cfs, the average is around 800 cfs, and the max is 8,600 cfs, the performance of all the models is fairly good.

	model	score
mean	mean	360.86499
hw10	hw10	118.95024
hw25	hw25	106.54739
hw50	hw50	99.26605
hw75	hw75	98.18728
hw90	hw90	103.39135
arima1	arima1	162.20975
arima2	arima2	140.35201
arima3	arima3	150.39021
arima4	arima4	140.75456
arima23	arima23	132.41761
arima24	arima24	126.33372
arima34	arima34	132.35751
arima234	arima234	121.82025

Table 2: RMSE scores for all models

Choosing the best overall model is less straight-forward. Within the Holt-Winters class of models, setting $\alpha = 0.75$ achieves the lowest RMSE (even compared to all ARIMA models). Within the ARIMA class, the model with temperature and snow depth as covariates achieved the lowest RMSE (negligibly better than the model with all three covariates). The no-covariate ARIMA model performed the worst. The best single-covariate ARIMA model uses temperature. The best two-covariate ARIMA model uses temperature and snow depth. Viewing this from a step-wise selection perspective, we can see that temperature is the most informative covariate, followed by snow depth and precipitation accumulation. This makes sense because hot temps would melt more snow and be associated with higher water flow, until the snow depth reaches zero of course. Cold temps would be associate with lower water flow since less (or if below freezing: zero) snow is being melted into runoff. Since precipitation accumulation includes rain and snow, high precipitation does not always translate to higher water flow if the precipitation is snow (temps are below freezing). This points to including interaction terms, which could be the focus of future work. But with these models, the Holt-Winters with $\alpha = 0.75$ is the best. Figure 10 shows boxplots for all models errors.

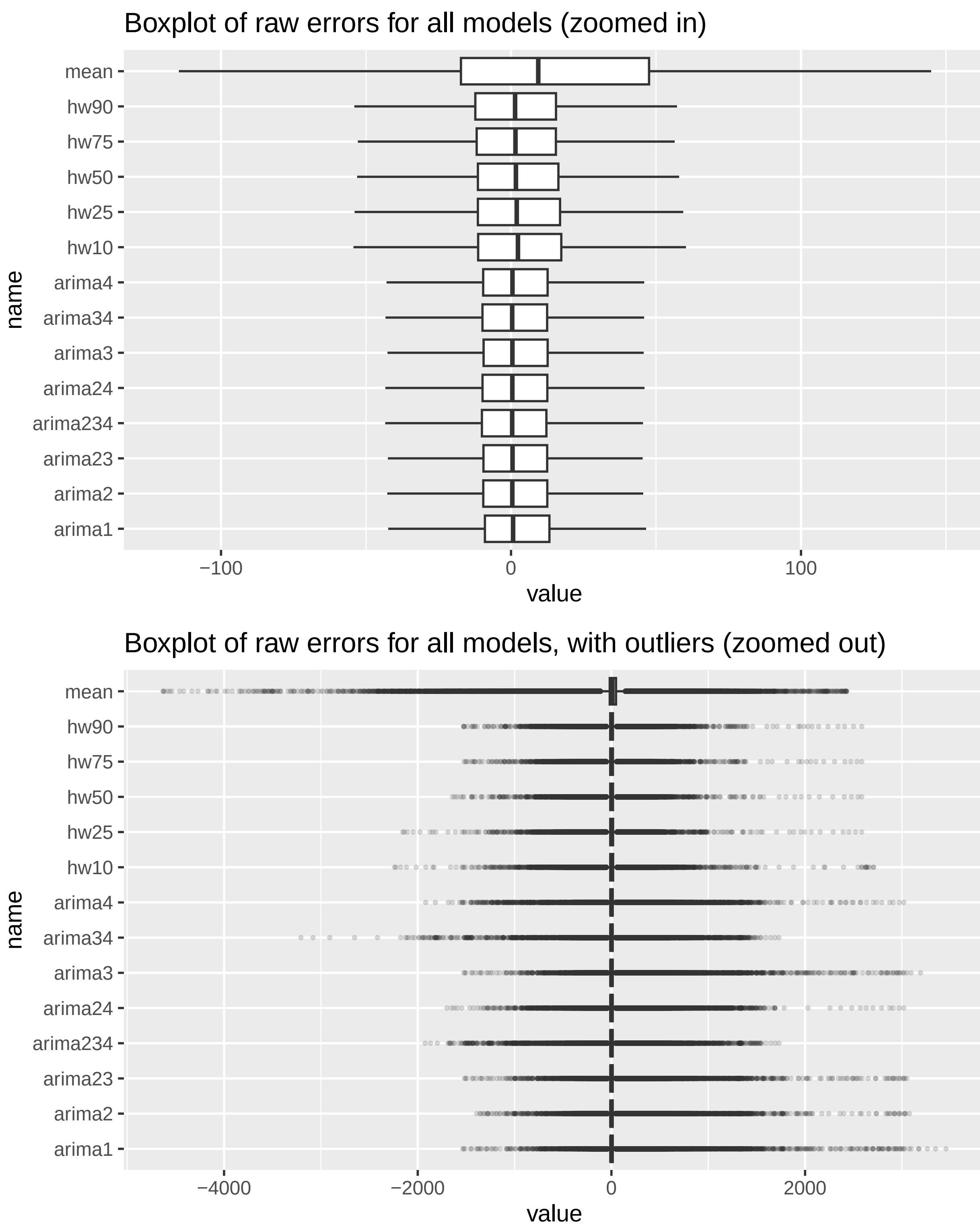


Figure 10: Boxplots of all model errors

Discussion

In the end, the models were quite successful. All of the Holt-Winters models out-performed all ARIMA models. Holt-Winters models use no covariates, which allows for smaller training sets. I want to try smaller-sized training batches (note: this would exclude ARIMA models with covariates), recall I used 10 days of training data for this analysis. It would be interesting to compare the performance of these smaller-sized training sets. Maybe there is an ideal size when predicting with Holt-Winters models on this data. These small batch training models might be quick and effective to deploy as a real-time forecasting tool. The ARIMA models seem to require interaction terms to be most effective. It would be interesting to see if including interaction terms to the ARIMA models would allow them to out-perform the Holt-Winters models. As future work, I would like to implement a script that scrapes recent water flow and weather data and makes a real-time forecast.