Predicting Lake Shasta's Storage Levels
STA 137 Final Project

Henry Shipper
996962278

March 8, 2015

Part I

Abstract

In this project I examine historical levels of water storage in Lake Shasta. This is done by decomposing the levels into seasonal, drift, and residual components by means of differencing. Through the insight gained from these analyses I then attempt to forecast future reservoir levels and demonstrate that Calfornia's lower water reservoir levels are primarily explained by the storage in previous years.

Part II

Introduction

Lake Shasta is the largest reservoir in California, with a total capacity of 4,552,000 acre-feet. However, the most recent measurement (2,612,715 as of February 2015) still falls well below the historical average, at just 58%.[1]

The amount of water available to the state of California has tremendous impacts. As water becomes more scarce, it becomes more expensive for households, restaurants, and most importantly for the Californian economy, it becomes more expensive for agriculture. Being able to predict how much water will be available would provide a number of useful applications, including helping farmers plan for upcoming seasons and helping households predict how much water they will be able to use.

Throughout this report, I will be examining how the volume of water stored has varied over time and how we can use it to help predict how much water will be in the reservoir in future years. Should we continue using the water at the pace we are (and enduring similar dry years), this could even provide us with a finite time at which the reservoir will be depleted.

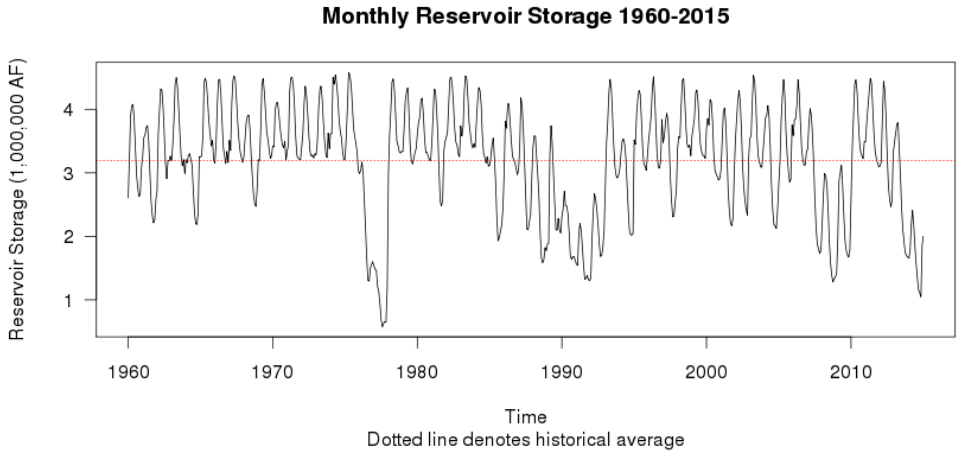
In the following sections I will explain how I have decomposed the data into drift, seasonal, and residual components, forecast the next few years, and then discussing my findings as well as providing analysis on what could have improved my modeling.

[1] `http://cdec.water.ca.gov/cgi-progs/reservoirs/RES`

Part III

Introducing the Data

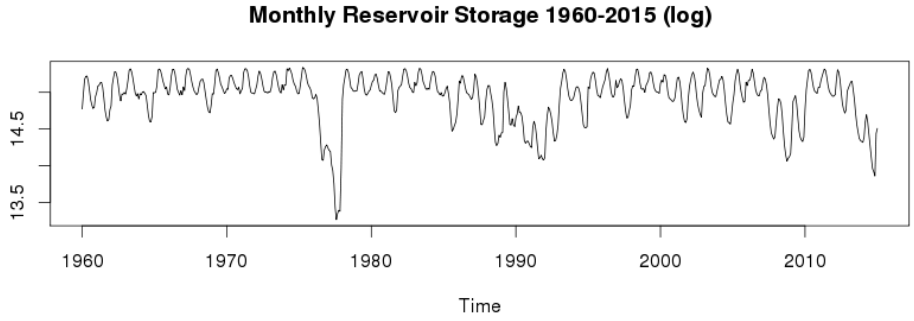
This data was retrieved from the California Data Exchange Center, a part of the California Department of Water Resources [2]. The specific dataset used for this project was pulled using the XML package for R. This tab le contains monthly recordings of the storage level of the Shasta Dam Reservoir, refered to in short as SHA by the Department of Water Resources. The table provides the monthly recording of volume of water in the reservoir in acre-feet (AF) from January, 1960 through January, 2015. For ease of matrix computations, I chose to ignore the data from January 2015, meaning that there are 660 monthly observations in total in the dataset. The time series plot of the data can be seen below. Before we get to the graph, however, one note must be made about the data: there was no recorded measurement for January, 1970. To remove the issue of having missing data, I did fit an autoregressive model on the previous year's worth of data as well as on the following year's worth of data and averaged the two predictions given from each to find an appropriate estimate for it. Onto the time series plot.

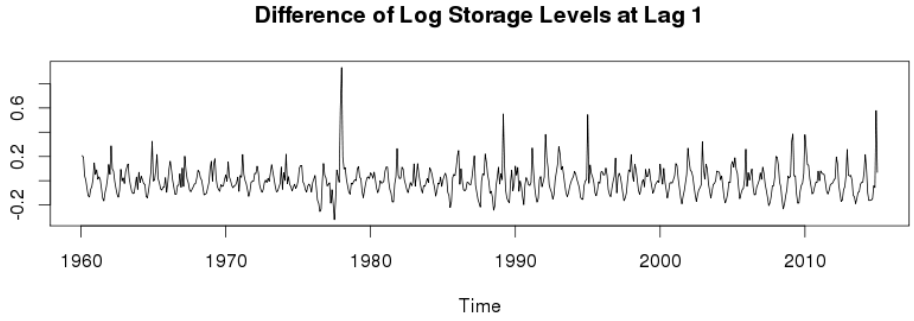


Upon first inspection, a few things stand out about the data. Firstly, around 1976 we start to see significantly more instances where the recorded storage is below the historical average (which itself is brought down significantly by the year 1977, when storage hit an all-time low of 578,000 AF). Despite the increasing magnitude and number of instances of differences below the historical average, the maximum measured storage level does not seem to vary significantly, which makes sense — after all, the reservoir has a limited capacity. As a result of the increasing number of instances that dip below the average, the variance of this dataset does not remain constant, so it must be transformed to reduce the difference in variance. I chose the natural log of the data, as neither the square nor cube root of the data seemed to

[2]URL in Appendix

even the variance quite as well.





As we can see from the difference graph, the change from month-to-month seems to be reasonably constant after the log transformation. Given that we are now happy with the look of the data, we can move onto decomposition.

Part IV
Data Analysis

Time series observations are classically broken into three distinct parts: Seasonality, which accounts for cyclical effects; Drift, which accounts for long term trends; and residual effects, which encompass everything else. The model can be written mathematically as:

$$Y_t = m_t + s_t + X_t$$

Where Y_t represents our observations, m_t represents our drift component, s_t represents our seasonal component, and X_t represents the residual effects.
I chose to employ a weighted average method to eliminate the seasonality component from the model. I created an initial 2-sided moving average

$$\hat{m}_t = \sum_{-6}^{6} \frac{1}{12} * (.5t_{-6} + t_{-5} + ... + t_5 + .5t_6)$$

and then constructed seasonality estimates by first averaging distance from the moving average for each month:

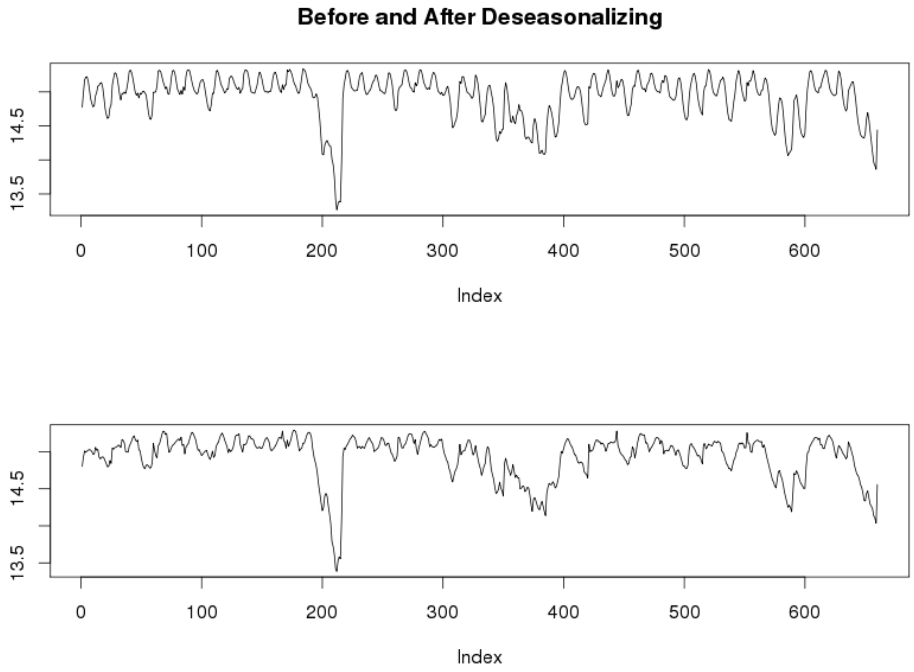
$$\begin{aligned} \mu_k &= \tfrac{1}{54}\textstyle\sum_{j=2}^{55}(Y_{k+d(j-1)} - \hat{m}_{k+d(j-1)}) && k = 1, \ldots, 6 \\ \mu_k &= \tfrac{1}{54}\textstyle\sum_{j=1}^{54}(Y_{k+d(j-1)} - \hat{m}_{k+d(j-1)}) && k = 7, \ldots, 12 \end{aligned}$$

We then subtract the average of the generated μ's from each μ_k to center the seasonality effect around 0.

$$\hat{s}_k = \mu_k - \tfrac{1}{12}\textstyle\sum_{l=1}^{12}\mu_l.$$

Here are the generated seasonality effects:

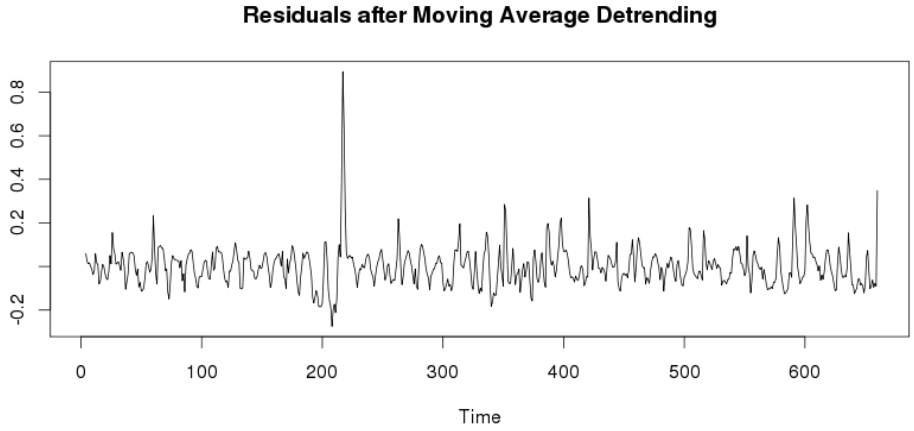
Jan	Feb	Mar	Apr	May	Jun
-0.026834	0.054091	0.166567	0.224858	0.212726	0.137789
Jul	Aug	Sep	Oct	Nov	Dec
0.006302	-0.119357	-0.173831	-0.186021	-0.174243	-0.116776



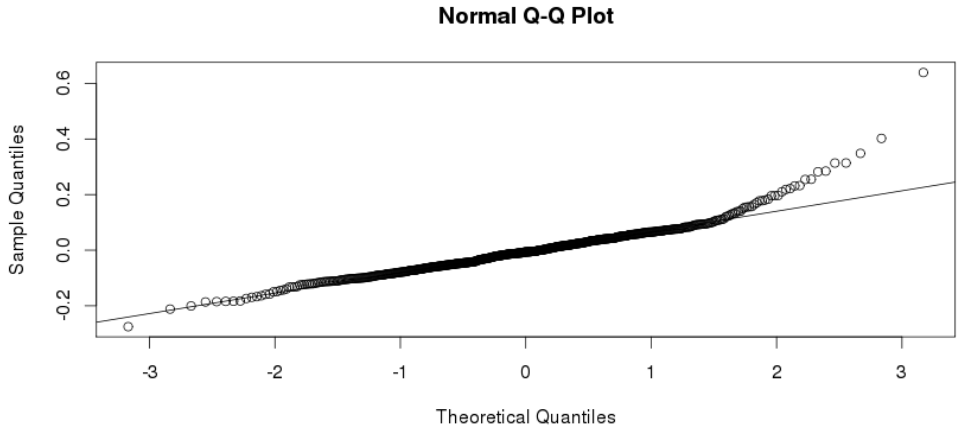
As we look at the graph of the time series before and after deseasonalizing, it appears that there is still some periodic trending occuring, though not to the same degree as before. Since the data does not appear to be a linear trend, a moving average will again be our best option. However, since we ultimately wish to forecast, we will need to use a one-sided moving average instead of two. I chose to apply the following formula to obtain this moving average:

$$\hat{x}_t = \frac{1}{4}x_t + \frac{1}{4}x_{t-1} + \frac{1}{4}x_{t-2} + \frac{1}{4}x_{t-3}$$

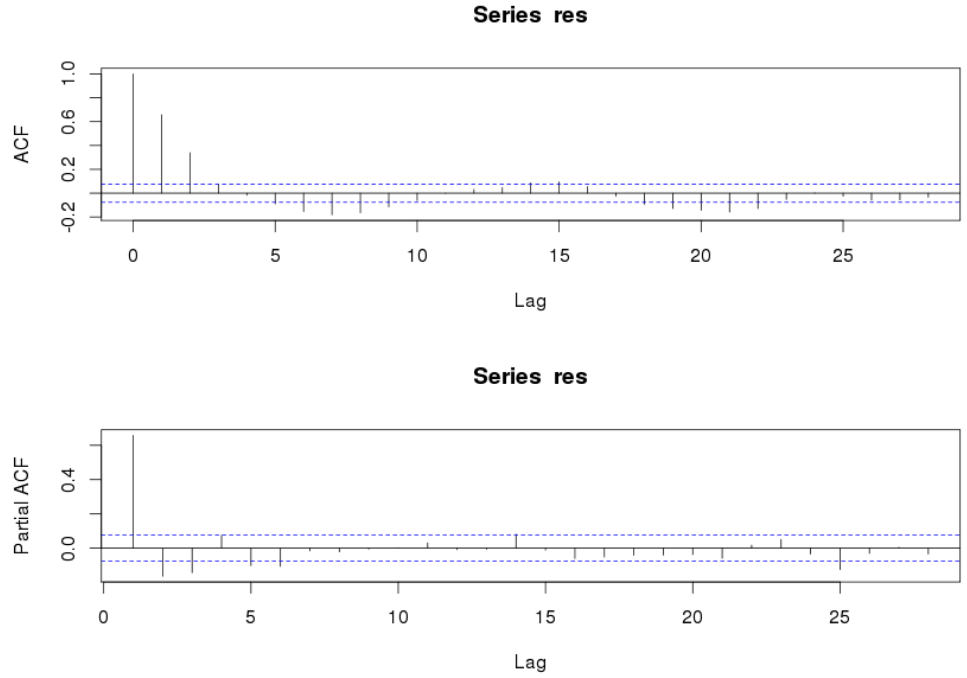
Subtracting this from the deseasonalized data provides a relatively-stationary looking residual plot, though there is one outlier that we may wish to replace for the sake of improving our predictive power for the residual component.



After removing that outlier (via the same method we used to remove the original outlier from the dataset - averaging predictions from autoregressive functions), our residuals look to be distributed normally for the most part, as displayed by the following Q-Q plot.



If our residuals here are stationary and normal, the use of an ARMA process should be able to eliminate all of our remaining dependence, assuming we find one with a good fit. Examining the ACF and PACF graphs for our data will help us determine which process to use.



As we can see, the ACF plot tails off and the PACF plot appears to cut off after lag 6. Therefore we can take a good guess that this is an AR(6) function. Now that we have an AR

ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6
0.800709	-0.089602	-0.251162	0.179913	-0.030520	-0.114155

Table 1: AR(6) Coefficients

process fit, we can combine this with our seasonal trends and our moving average to generate forecasts for the upcoming few months. Our model for generating $E[Y_{t+1}]$ is as follows:

$$E[Y_{t+1}] = E[m_{t+1}] + E[s_{t+1}] + E[X)_{t+1}]$$

$E[s_{t+1}]$ is simple to calculate, as it is just $\hat{s}_{t-(j*d)+1}$ where j represents the number of periods that have been completed since the series started. Therefore, $\hat{s}_{t+1} = \hat{s}_1 = -0.027274$. $E[m_{t+1}]$ can be calculated by find the expected value of the entire formula:

$$E[m_{t+1}] = \tfrac{1}{4}(E[m_{t+1}] + E[m_t] + E[m_{t-1}] + E[m_{t-2}])$$

The expected values of all but $E[m_{t+1}]$ are known from previous calculation. Under the (incorrect, but I'll address this later) assumption that the trend is stationary, we can predict that $E[m_{t+1}]$ is the historical average trend effect. Lastly, we can use OLS to predict our residual value based on our previous computed coefficients.
From these formulas, here are our predicted (log) storage levels for Shasta Lake over the next three months: As we can see, we project the water level to rise, but this can be mostly

January	February	March
14.6361	14.6878	14.7168

explained by seasonal effects, whose values are shown earlier. Unfortunately it is impractical for us to predict beyond three months - as we should not even pay much attention to our third. By the third prediction, our standard error has greater magnitude than our predicted residual, mean that the true value of the residual could be significantly different.

Part V

Discussion and Conclusion

The most interesting information gleaned from performing analysis is that the seasonal and residual effects are not of large magnitude. Rather, it is the drift component (a moving average, in our case) that determines the storage level. Unfortunately, our data does not provide us with a particularly helpful model for forecasting future levels of Lake Shasta's storage. There are a number of reasons that this is possibly the case. To begin with, the reservoir's maximum capacity makes it impossible for there to be any sort of upwards linear trend. The reervoir reached peak capacity in many years, though the amount of water that is depleted does seem to be increasing. Because of this, there is increasing variance from the beginning of the dataset until the end, and performing a log transformation only mitigates that to a certain extent.

Another issue that faces us with this data is that the time series is not stationary. Again, the maximum capacity seems to prevent any possible polynomial fitting, since instead of reaching extrema, the drift component levels off.

I think that the best way to improve the predictive power of this series would be to instead fit a very weak drift component (possibly negligible, even) and instead attribute much more of the variation in levels to the residuals. Furthermore, storage levels depend on much more than just how much water was previously there. To some degree they can, and will, fluctuate based on State resident and business usage, but they depend very highly on climate and precipitation, two factors which I was unable to incorporate into this model.

```
library(plyr) ###for ldply()
library(XML) ###for readHTMLTable()
library(lattice) ###for plots

url <- "http://cdec.water.ca.gov/cgi-progs/selectQuery?station_id=SHA&dur_code
=M&sensor_num=15&start_date=01/01/1960+00:00&end_date=01/01/2015+00:00"
db <- readHTMLTable(url, stringsAsFactors = FALSE)

reslevel <- db[[1]]
names(reslevel) <- c("Date", "Storage")
reslevel[,1] <- sapply(reslevel[,1], function(x) paste0("01/", x))
reslevel[,1] <- as.Date(reslevel[,1], "%d/%m%Y")
reslevel$Storage <- as.numeric(reslevel$Storage)
reslevel <- ts(reslevel$Storage, frequency = 12, start = c(1960, 1))
reslevel <- as.numeric(reslevel)
#####Fill missing value for January 1970
which( is.na(reslevel == TRUE)) #returns 121 as location of January 1970
subset.previous <- reslevel[108:120] #use previous year to estimate
subset.future <- reslevel[134:122] #use next year to estimate
fit.sub.pre <- ar.yw(subset.previous)
fit.sub.fut <- ar.yw(subset.future)
tmp <- (predict(fit.sub.pre)$pred[1] + predict(fit.sub.fut)$pred[1])/2
plot(reslevel, type = "l")
points(x = 1970, y = tmp, col = "red")
reslevel[121] <- tmp
reslevel <- ts(reslevel, start = c(1960, 1), frequency = 12)
plot(reslevel, type = "l" , main = "Monthly Reservoir Storage 1960-2015",
     ylab = "Reservoir Storage (1,000,000 AF)", yaxt = "n",
       sub = "Dotted line denotes historical average")
axis(2, at = (1:4*(1000000)), labels = 1:4, las = 2)
abline(h = mean(reslevel), lty = 3, col = "red")

par(mfrow = c(2, 1))
plot(log(reslevel), ylab = "", type = "l",
  main = "Monthly Reservoir Storage 1960-2015 (log)")
plot(diff(log(reslevel)), type = "l", ylab = "",
  main = "Difference of Log Storage Levels at Lag 1")

log.res <- log(reslevel[-661])
hatm <- filter(log.res, sides = 2, c(.5, rep(1, 11), .5)/12)

A <- matrix(log.res, ncol = 12, byrow = "TRUE")
```

```
M <- matrix(hatm, ncol = 12, byrow = "TRUE")
mu <- array(0,12)
for(k in 1:6) mu[k] = sum(A[2:55,k]-M[2:55,k])/54
for(k in 7:12) mu[k] = sum(A[1:54,k]-M[1:54,k])/54
hats <- rep(mu - mean(mu), 55)
trends <- array(0, 55)
for(i in 1:55) trends[i] <- sum(A[i, 1:12])/12
fulltrends <- rep(trends, each = 12)

desea <- log.res - hats
plot(log.res, type = "l", main = "Before and After Deseasonalizing", ylab = "")
plot(desea, type = "l", ylab = "")
lines(desea, col = "red")
detrend <- filter(desea, sides = 1, c(rep(1, 4)/4))
res <- desea - detrend
plot(res, main = "Residuals after Moving Average Detrending", ylab = "")

which.max(res) ####returns 217, location of outlier
sub.pre <- res[197:216]
sub.fut <- res[238:218]
fit.pre <- ar.yw(sub.pre)
fit.fut <- ar.yw(sub.fut)
res[217] <- (predict(fit.pre)$pred[1] + predict(fit.fut)$pred[1]) /2
plot(res)

qqnorm(res)
qqline(res)

acf(res, na.action = na.pass)
pacf(res, na.action = na.pass)

fit.ar <- ar.ols(res[-(1:3)], demean = T)
tmp <- detrend
trend <- function(tmp){
  c(tmp, (1/4) * (mean(tmp, na.rm = TRUE) + tmp[length(tmp)] +
  tmp[length(tmp)-1] + tmp[length(tmp)-2]))
}
#####Forecasting for January, February, March 2015
detrend <- trend(detrend)
trend.jan <- detrend[length(detrend)]
jan <- predict(fit.ar)$pred[1] + hats[1] + trend.jan
detrend <- trend(detrend)
```

```
trend.feb <- detrend[length(detrend)]
feb <- predict(fit.ar, n.ahead = 2)$pred[2] + hats[2] + trend.feb
detrend <- trend(detrend)
trend.mar <- detrend[length(detrend)]
mar <- predict(fit.ar, n.ahead = 3)$pred[3] + hats[3] + trend.mar
```