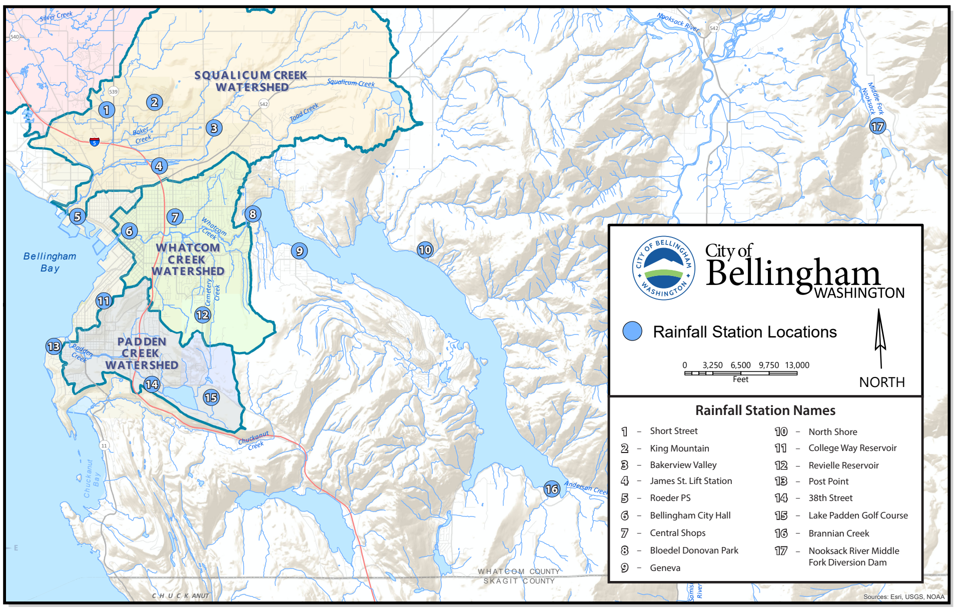
Forecasting Average Rainfall

Using an ARIMA Model to predict weather in Bellingham, WA.

Seren Dances

Time Series Analysis
Math 456



12/6/2021

Abstract

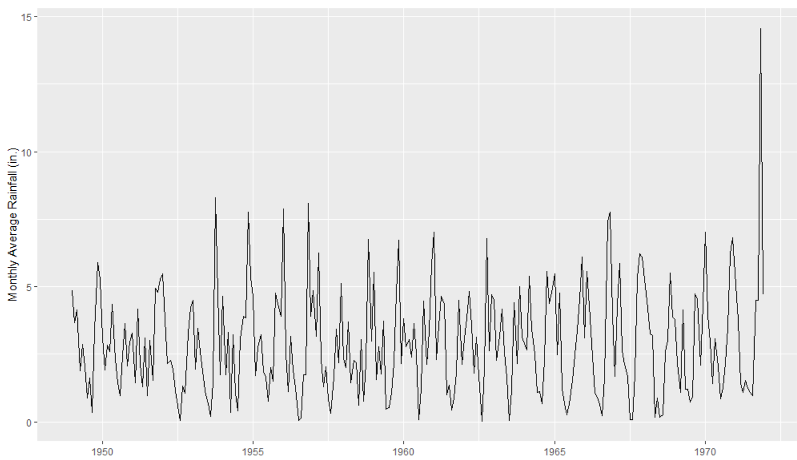
The upper-left corner of the Pacific northwest is known for its rainfall, cloudy skies and lush vegetation. This region, like much of the world, is beginning to feel the impact of global warming. This is particularly notable in the source of the 2021 heat wave, followed by a series of floods in fall 2021. These floods have been caused by record rainfall in the pacific northwest through the month of November. On November 22nd, the recorded rainfall was 2.78 inches, which is a third of what is generally expected in the entire month of November. Are these extreme weather conditions and aberration, or are the part of a larger trend set in play by global warming? In particular, what can we expect from the average monthly rainfall in coming years.

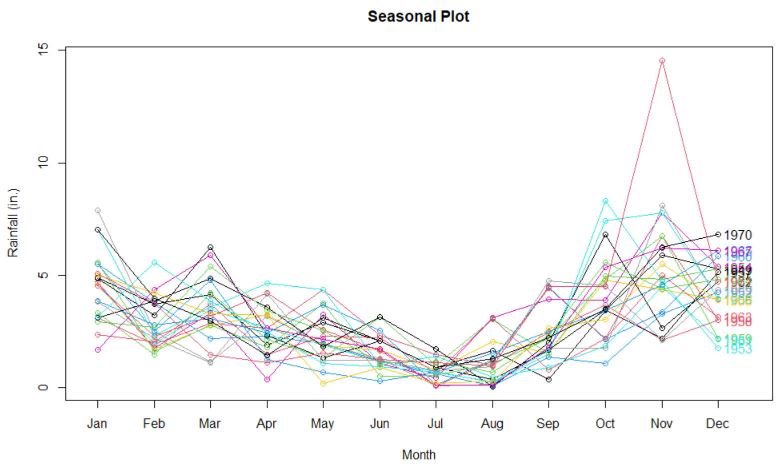
Methodology

I sourced public data from the Bellingham Airport records. This data spans from 1949 to the present day (December 2021.) There were some missing values present so some alterations were necessary. Firstly, I replaced the missing values with the averages for the respective months. Additionally, the years of 1997 and 1998 had half of their data missing. Because of this, and with the desire to prevent over-fitting, I decided to use the data from 1999 to 2019 for my training data, and 2020 and 2021 for my testing/ validation data.

Data Exploration

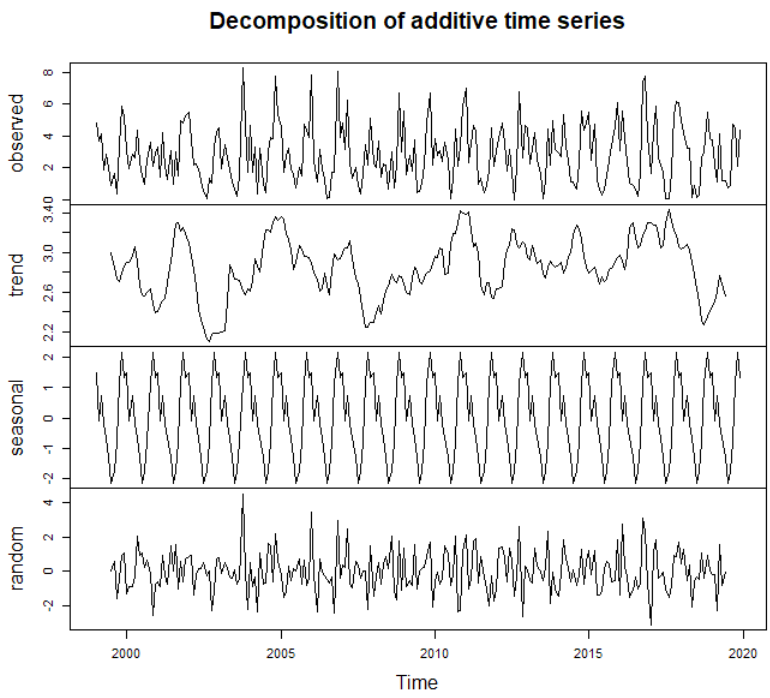
To begin exploring the data, I graphed the time series as well as an overlay of all years on record by average rainfall in a particular month.





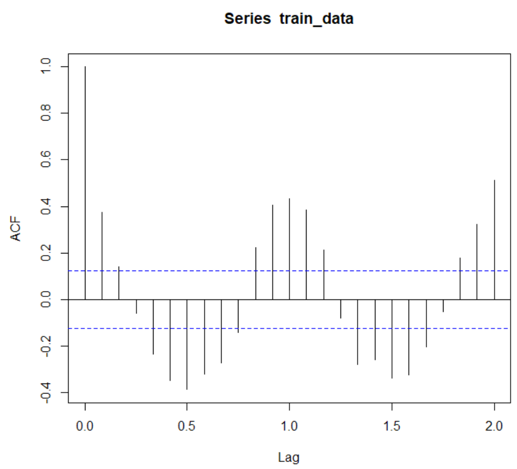
```
rainfall = read.csv("rainfall_1949.csv")
rainfallts = ts(rainfall[600:875,], frequency = 12, start = 1949)
#ploting the raw data
autoplot(rainfallts, ylab = "Monthly Average Rainfall (in.)", xlab = "")
#ploting all years by month (the 14.57 point was from this november)
seasonplot(rainfallts, year.labels = TRUE, col = 1:13,
main =  "Seasonal Plot", ylab= "Rainfall (in.)")
```

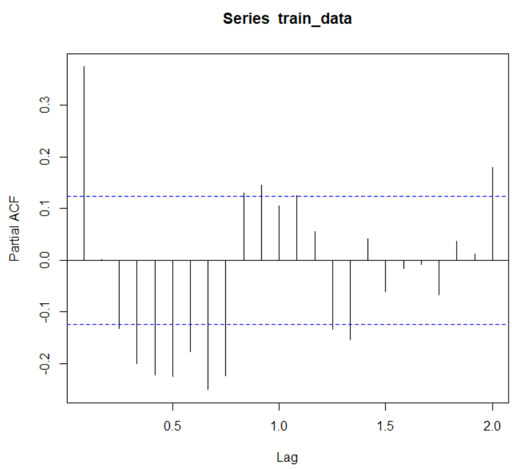
Decomposition




```
decomp = decompose(train_data)
```

Model Selection By looking at the ACF and PACF some idea of what a good arima model might be.






```
acf(train_data) #spike at 1 (with seasonality showing from sinusoidal form)
pacf(train_data) #spike at 2
```

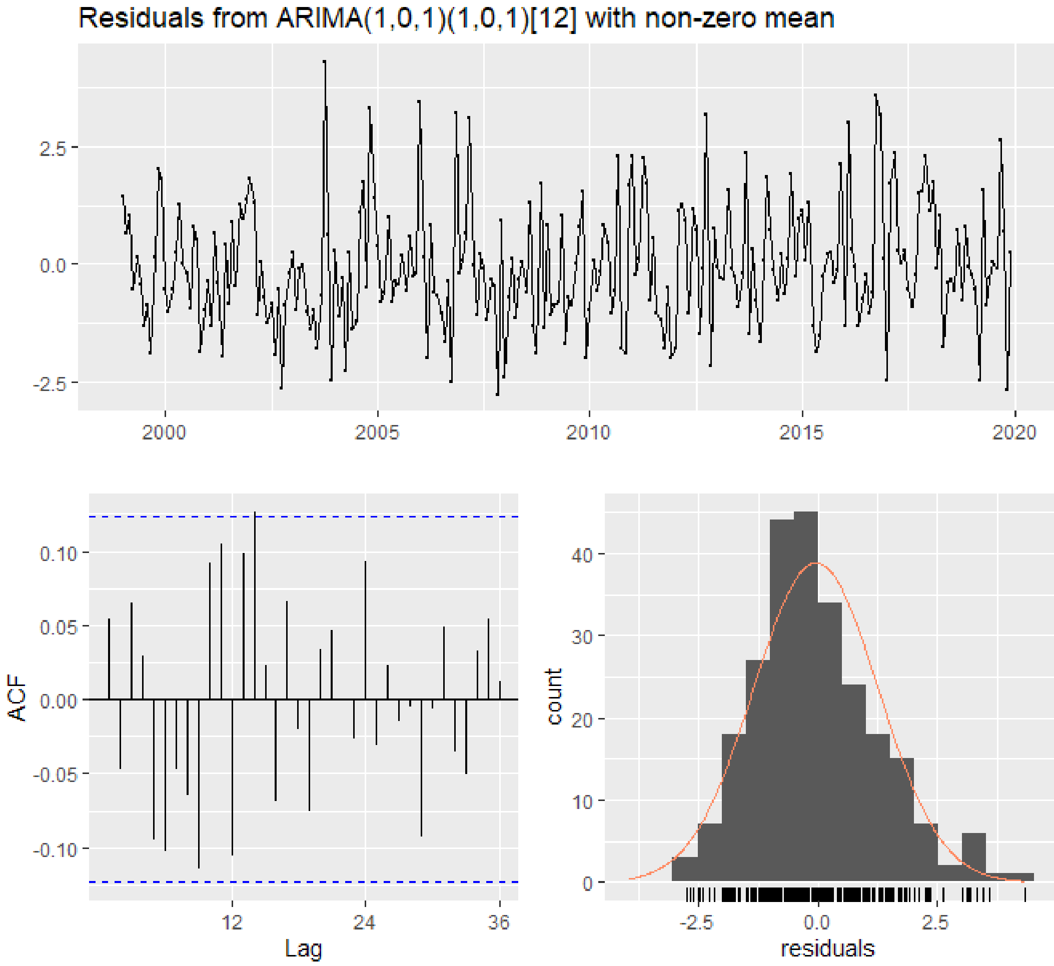
```
#possible models based on acf and pacf
#arima(1,0,2)(1,0,2)[12]
#arima(1,0,1)(1,0,1)[12]
#arima(0,0,2)(0,0,2)[12]
#auto.arima()
```

Testing out a few different variations based on the observations from the ACF and PACF.

```
x1 <- Arima(train_data, order = c(1,0,2), seasonal = c(1,0,2))
x2 <- Arima(train_data, order = c(1,0,1), seasonal = c(1,0,1))
x3 <- Arima(train_data, order = c(0,0,2), seasonal = c(0,0,2))
x_auto <- auto.arima(train_data)
data.frame('x1' = x1$aicc, 'x2' = x2$aicc, 'x3' = x3$aicc, 'auto.arima' = x_

                 x1       x2       x3 auto.arima
AICc Value 887.266 886.1808 949.9325   920.1606
```

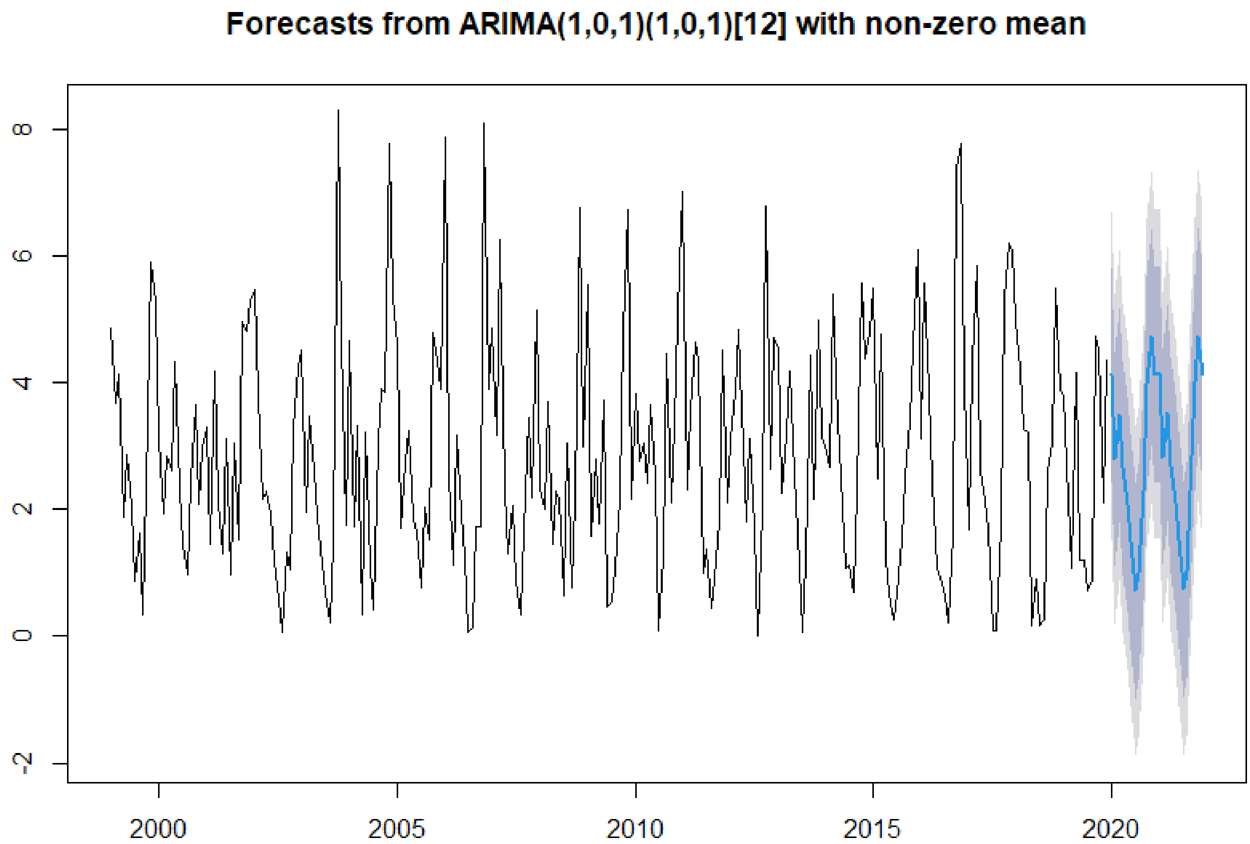
Based on the AIC values obtained, X2 seems to be the best model to explore further. Based on the ACF plot of the residuals, we seem to have accounted for the trend and seasonality. Now it is time to forecast.



Forecasting

Here I am forecasting the next two years, and then checking the accuracy of the model based on the testing data. The chosen forecast incorporates confidence intervals of 95% and 80%.

```
> f1 = forecast(x2, h = 24)
> plot(f1)
> accuracy(forecast(x2, h=12), test_data)
                      ME     RMSE      MAE         MPE      MAPE      MASE (
Training set -0.04506082 1.304313 1.015620 -145.379281 165.91517 0.6893152 (
Test set      0.65845421 1.375700 1.046903    9.257295  29.57413 0.7105477 (
```



Conclusion

By using the training data to check the accuracy of the forecast we are able to understand how well calibrated the chosen model is. In this case, the model did a good job of predicting the future patterns. However, there was a slight decrease in accuracy based on the recent uptick in flooding. This model accounts for the seasonality and variance of rainfall well. In the future, it would be interesting to test the same model with future values and see how it predicts changes under new circumstances.

RCODE

```
library(forecast)

rainfall = read.csv("rainfall_1949.csv")
rainfallts = ts(rainfall[600:875,], frequency = 12, start = 1949)
rainfallts
#Training Data
train_data = ts(rainfall[600:851,], frequency = 12, start = 1999)
#Testing Data
test_data = ts(rainfall[852:875,], frequency = 12, start = 2020)

#ploting the raw data
autoplot(rainfallts, ylab = "Monthly Average Rainfall (in.)", xlab = "")
#ploting all years by month (the 14.57 point was from this november)
seasonplot(rainfallts, year.labels = TRUE, col = 1:13,
           main =  "Seasonal Plot", ylab= "Rainfall (in.)")

#decomposition
decomp = decompose(train_data)
plot(decomp)

#model selection
acf(train_data) #spike at 1 (with seasonality showing from sinusoidal form)
pacf(train_data) #spike at 2

#possible models based on acf and pacf
#arima(1,0,2)(1,0,2)[12]
#arima(1,0,1)(1,0,1)[12]
#arima(0,0,2)(0,0,2)[12]
#auto.arima()

x1 <- Arima(train_data, order = c(1,0,2), seasonal = c(1,0,2))
x2 <- Arima(train_data, order = c(1,0,1), seasonal = c(1,0,1))
x3 <- Arima(train_data, order = c(0,0,2), seasonal = c(0,0,2))
x_auto <- auto.arima(train_data)

data.frame('x1' = x1$aicc, 'x2' = x2$aicc, 'x3' = x3$aicc, 'auto.arima' = x_
#based on the aic, we should use model 2

checkresiduals(x2) # based on the acf plot, this seems like an adaquite mode

#forecasting
f1 = forecast(x2, h = 24)
plot(f1)
accuracy(forecast(x2, h=12), test_data)
```

Sources

Title page image of weather stations in Bellingham was from the website: https://cob.org/services/environment/restoration/rainfall-data

Data-set was from: https://nowdata.rcc-acis.org/sew/