Temperature in Mumbai From 2000-2020

Abstract:
This data is from meteorological data from NASA research, the data containing long-term climate averaged estimates and surface solar energy fluxes. The data collected is monthly frequency data for a particular latitude and longitude in Mumbai for the period 2000-2020. The data consists of the following variables: specific humidity, relative humidity, temperature, and precipitation. For this project, we are going to look at the variable temperature, and analyze it using the time series methods we learned in class.

Rohan Majumdar (rmajumdar@ucsb.edu)

Introduction

Purpose: The purpose of this data is to understand the patterns of temperature in Mumbai from 2000-2020. We will use time series methods in order to determine what the causes, trends, or patterns were for temperature in Mumbai during this time span.

Reason: I chose this project because I have familial ties to India, where Mumbai is located, and I am interested in how the temperature has changed over time, considering that I have heard it can get rather hot there. That is what I found interesting about this dataset.

Past: The specific humidity, relative humidity, temperature, and precipitation have been what have been studied in this data, starting in 2000.

Methods: We will be using the sarima(p, d, q) model, as well as the garch() model in order to perform our time series analysis on the temperature in Mumbai from 2000-2020.

Discovery: There is something significant at lag 12 when using the GARCH model. I found that to be interesting.

Data

Time Range: 2000-2020

Frequency: monthly values

Size: 252 observations of 7 variables

Reason: I chose this project because I have familial ties to India, where Mumbai is located, and I am interested in how the temperature has changed over time, considering that I have heard it can get rather hot there. That is what I found interesting about this dataset.

Link: https://www.kaggle.com/datasets/poojag718/rainfall-timeseries-data

Background: The data is collected from the Power Data Access Viewer. The POWER Meteorological data is prediction or observation given by NASA's GMAO MERRA-2 assimilation model.

Significance: The significance of this dataset is that it helps us predict trends or patterns of the climate of a location, in this case, Mumbai.

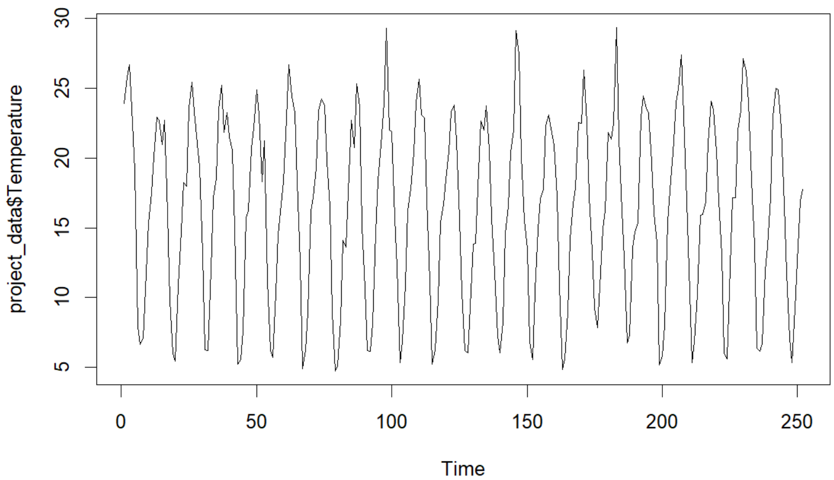
Purpose: The purpose of this data is to understand the patterns of climate statistics in Mumbai from 2000-2020. We will use time series methods in order to determine what the causes, trends, or patterns were for temperature in Mumbai during this time span.

Methodology

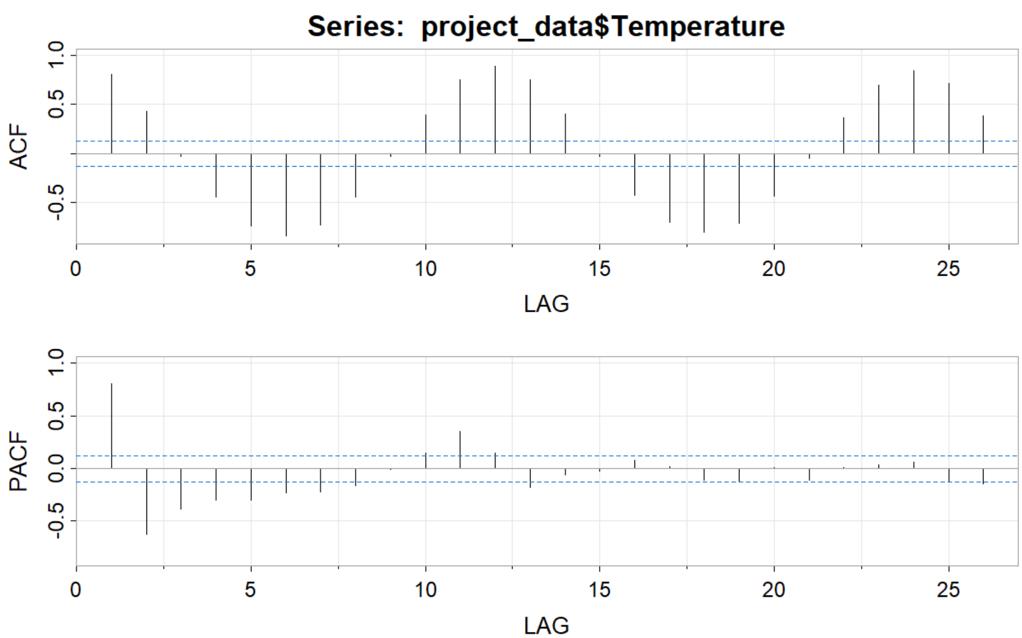
The SARIMA (p, d, q) x (p, d, q) model , also known as the multiplicative seasonal autoregressive integrated moving average model, is used to perform regression with autocorrelated errors. Similar to ARIMA models, the SARIMA model takes into account past values to predict future values, but takes into account any seasonal patterns when predicting future values.

The GARCH() model, also known as the generalized autoregressive conditional heteroskedasticity model. We use this model when an ARMA model is assumed for error variance, which is what we have based our data collection from the data in its transformed and untransformed state.

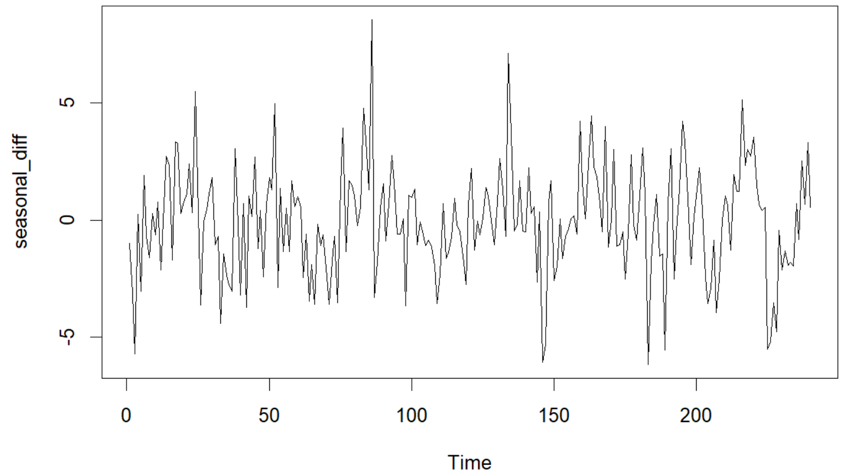
Results



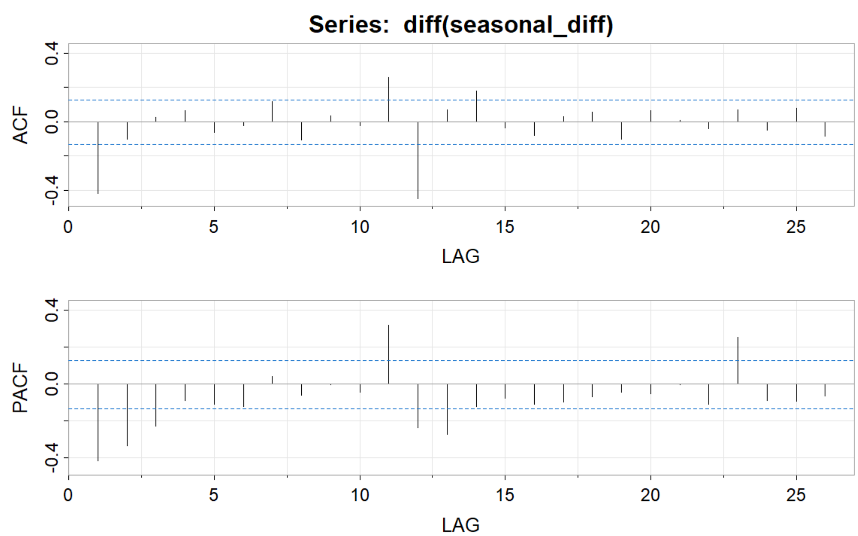
This is the plot of time series data, variable being temperature. The data appears stationary in the figure above.



This is the ACF and PACF of the original data's temperature. There are significant lags for the PACF at 1 and 2, meaning that this follows an AR(2) model.

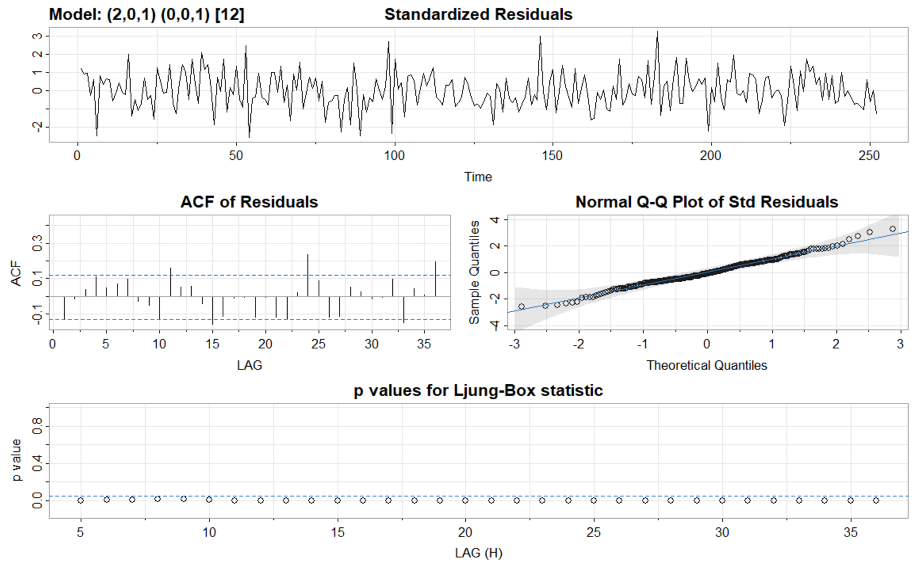


This is the time series plot of the seasonal difference of the temperature in Mumbai. The plot does not appear as stationary as the plot without the seasonal difference did.



This is the ACF and PACF for the seasonal difference of the temperature in Mumbai. Here, there is significant lag at 1 for the ACF of the seasonal difference, meaning it follows a MA(1) pattern.

SARIMA Model:



Here are the diagnostics of the data after doing the sarima() method on the data, using the seasonal difference. These are the results of our sarima() model, where we predict the 12 future values for the temperature based on our previous values. The residuals appear somewhat stationary, though not as much as the non-transformed data time series plot. The ACF for the residuals does not experience significant lag until it is too late to matter. The normal qq-plot of standard residuals seems to fit as well. So, with the seasonal difference, it appears that the temperature follows a similar pattern here as the earlier time series plots.

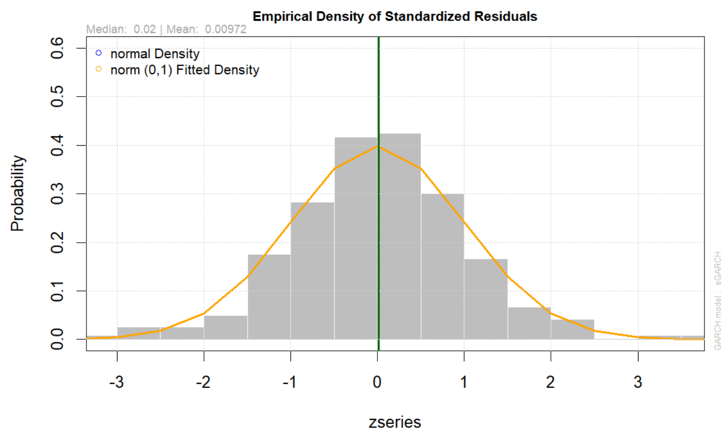
```
Estimate      SE  t.value p.value
ar1      1.6490 0.0281  58.6804       0
ar2     -0.9004 0.0275 -32.7237       0
ma1     -0.8357 0.0363 -23.0410       0
sma1     0.2676 0.0558   4.7985       0
xmean   16.3043 0.1220 133.6332       0
$AIC
[1] 4.575702
$AICc
[1] 4.57667
$BIC
[1] 4.659737
```

Above are numerical values that showcase the quality of our model.

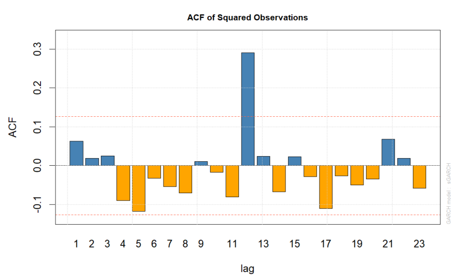
GARCH model:

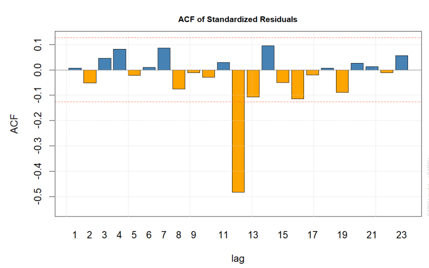
	t	p
Sign Bias	0.01484957	0.9881648
Negative Sign Bias	0.40026807	0.6893227
Positive Sign Bias	0.08525976	0.9321275
Joint Effect	0.22890666	0.9727931

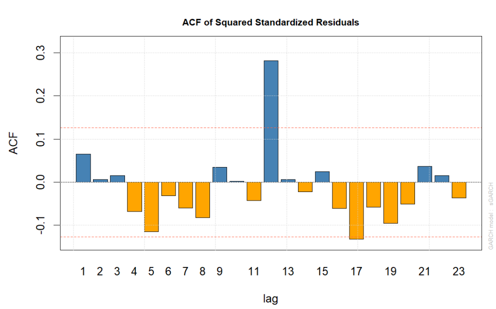
These are the results of our GARCH model. As you can see, the p-values for sign bias, negative sign bias, positive sign bias, and joint effect are all above 0.05, so the sign and joint effects have no significant impact on the volatility of the temperature in our model.



Above is the empirical density of standardized residuals, showing that our residuals follow a normal distribution.







These three plots are the ACF of our squared observations, standardized residuals, and of our squared standardized residuals. We can see that there is no significant lag until around 12 for each plot, so clearly there is something significant happening to the volatility of our data at that point.

Conclusion and Future Study

Our project was meant to create models that help forecast the temperature in Mumbai following a time spanning from 2000 to 2020. The time series plot for the original data appears to be relatively stationary, while the ACF and PACF plots lend themselves to the idea that the original data follows an AR(2) model. We then transformed the data by checking the seasonal difference of the data. The time series plot for the transformed data still appeared somewhat stationary, but not as stationary as the original time series data for temperature in Mumbai from 2000 to 2020. The ACF and PACF plots for the transformed data followed an MA(1) model. These discoveries helped us in creating our sarima() model, which we used to forecast the 12 months following the end of our timespan of temperatures in Mumbai. The residuals from this data seemed similar to the time series plot of our seasonal difference, and there appeared to be no significant lag in the ACF plot. We found the AIC, BIC, and AICc to assess the performance of the model. We then did a GARCH model to assess the volatility of our model. We learned through our diagnostics that sign bias and joint effect had little to no impact on the volatility of our data. We found that our empirical density followed a normal distribution, and that something significant happens at lag 12 with our volatility according to our ACF plots. I found all these things to be very interesting.

References

https://en.wikipedia.org/wiki/Autoregressive_conditional_heteroskedasticity

Time Series Analysis and its Applications with R Examples by R., by H. Shumway and D. S. Stoffer

https://www.kaggle.com/datasets/poojag718/rainfall-timeseries-data

Appendix

Code for initial time series plot:

```
plot.ts(project_data$Temperature)
```

Code for ACF and PACF of original data:

```
acf2(project_data$Temperature
```

Code to remove seasonal difference from data:

```
seasonal_diff <- diff(project_data$Temperature, 12)
```

Code for time series plot of new data:

```
plot.ts(diff(seasonal_diff))
```

Code for ACF and PACF of new data:

```
acf2(diff(seasonal_diff))
```

Code to forecast the next 12 values using sarima():

```
project_sarima <-
sarima(project_data$Temperature,2,0,1,0,0,1,12)
```

Code for our GARCH model:

```
project_garch_spec <-
ugarchspec(variance.model=list(model="sGARCH",
garchOrder=c(1,1)), mean.model=list(armaOrder=c(2,1)))
project_garch <- ugarchfit(spec = project_garch_spec,
data = seasonal_diff)
```

Code for GARCH plots:

```
plot(project_garch)
```