

Sea Level Time Series

Michael Kwok

12/11/2020

Abstract

This report aims to model historical sea level time series data in order to forecast future sea levels. Time series techniques are split into the following sections of initial plot and analysis, transformation and differencing, ACF and PACF (Autocovariance Function and Partial Autocovariance Function) analysis, model fitting and diagnostic checking, and forecasting. Of the 11 models proposed, the final model is (Autoregressive Integrated Moving Average) ARIMA(2,1,2), which passes all residual diagnostic tests and has the lowest AIC (Akaike Information Criterion) compared to the other models. The majority of this model's forecast is accurate. Note that towards the end of the dataset, sea levels may be increasing at a faster rate than before, in which case a new model can be developed in the future as more data is collected.

Introduction

The dataset, from Church and White (2011; modified 2013), contains 134 observations of time series data that tracks yearly global mean sea levels from 1880 to 2013. The goal is to use this time series data to forecast future sea level activity as the Earth continues to warm. This is interesting and important because rising sea levels pose a threat to coastal populations and natural habitats due to the permanent loss of land. Thus, people must understand this data and its implications, and work to prevent and prepare for future scenarios.

The technique to be used is as follows. First, examine the data for any trend, seasonality, or increasing variance. Apply Box-Cox and/or differencing accordingly to make the data stationary. Next, examine the ACF/PACF graphs to identify potential models for MA and AR parameters. Then, compare the AIC of each model and decide which ones to test; keep in mind the principle of parsimony, i.e. less parameters is better if possible. After that, conduct the unit circle test to ensure that the data is stationary/invertible, and then do diagnostic checking on the residuals to ensure they are Gaussian white noise. If these tests pass, then the model is a good fit and can be used for forecasting; otherwise, consider additional models.

In this project, six original models were suggested based on the ACF/PACF graphs. Two of these models were tested: ARIMA(1,1,0) and ARIMA(0,1,1). These models failed the residual diagnostic tests, and the residual ACF/PACF graphs gave an additional five models to consider. Of these five extra models, two of them displayed significantly lower AICs than the remaining nine models under consideration. ARIMA(2,1,2) had the lowest AIC, so it was tested. It passed all residual diagnostic tests, so it was declared as the final model. Note that more models were tested (primarily, ones with fewer parameters), but they failed residual diagnostic tests and/or had higher AICs. The results were omitted to prevent this report from becoming excessively long and confusing, but they can be supplied separately upon request.

The conclusion is that ARIMA(2,1,2) is decent for modeling sea levels. Its forecast does not perfectly capture all of the actual test data within its confidence intervals, but it is fairly close.

Software that was used includes Excel and R. The dataset was obtained from the Commonwealth Scientific and Industrial Research Organisation (CSIRO), an Australian Government agency responsible for scientific research. The URL is provided under References.

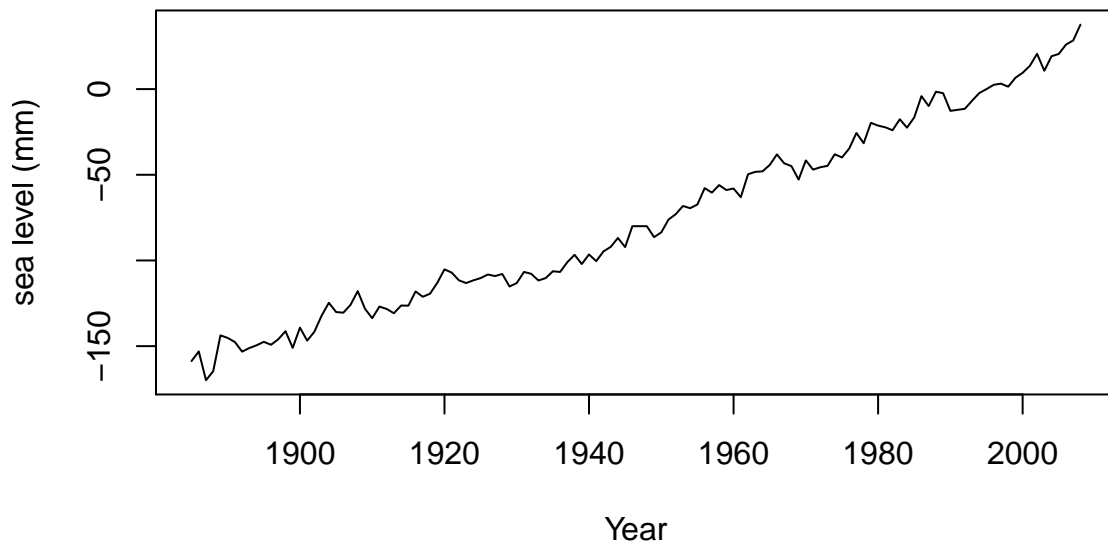
Section I - Plot & Analysis

The column of interest in the dataset is column 2: the global mean sea level (GMSL) measured yearly in millimeters. It contains values from 1880 up to 2013.

Let us start by splitting our dataset into a training set and a test set: 124 and 10 observations, respectively. We will use the training data to build the model and the test data for forecasting.

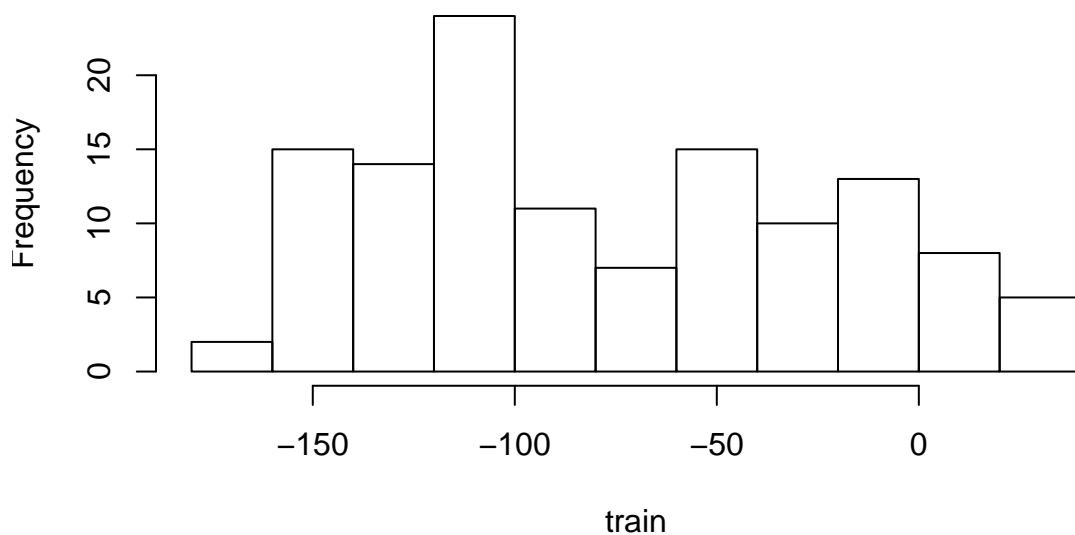
The plot of the data is shown below. We observe that i) there is a trend that looks linear, ii) there is no noticeable seasonal component, and iii) there are no noticeable sharp changes in behavior.

Sea levels



Plotting the histogram, we see that the data is not normally distributed. Furthermore, note the variance to be 2986.576.

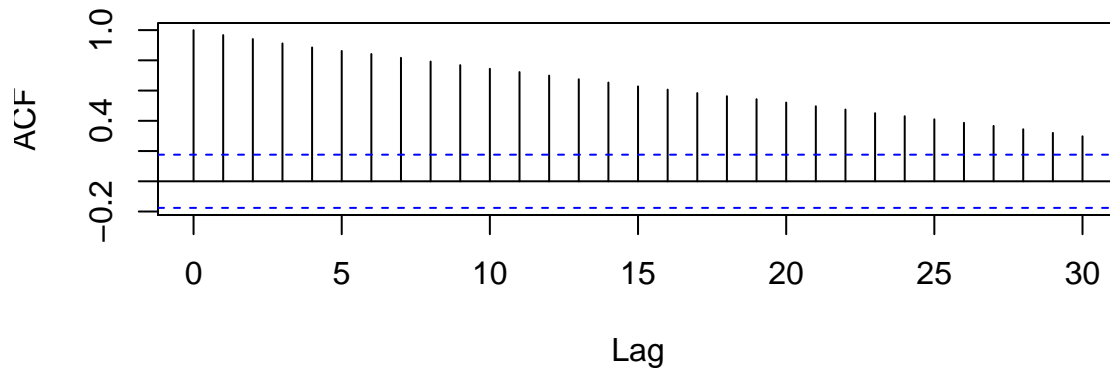
Histogram of data



Variance of data: 2986.576

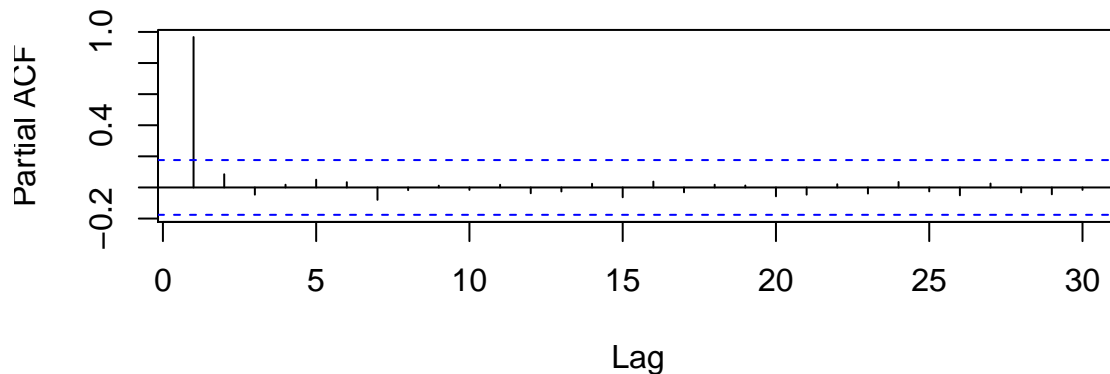
Meanwhile, the ACF plot shows nonstationarity since it is a gradual decay.

ACF of data



On the other hand, the PACF plot only has one value that exceeds the 95% confidence intervals at lag = 1.

PACF of data



Section II - Transformation and Differencing

Since the variance does not appear to increase over time, there is no need to do a Box-Cox variance. It also does not appear seasonal, so there is no need to difference at a seasonal lag.

Since there is an apparent linear trend, we suspect that differencing at lag 1 will remove the trend and make the data stationary. Before doing so, let's also try the `auto.arima()` function to see what parameters are recommended.

As shown below, the result is ARIMA(1,1,1), i.e. differencing at lag 1 and having $p=1$ for the AR component and $q=1$ for the MA component.

```
Series: train
ARIMA(1,1,1) with drift
```

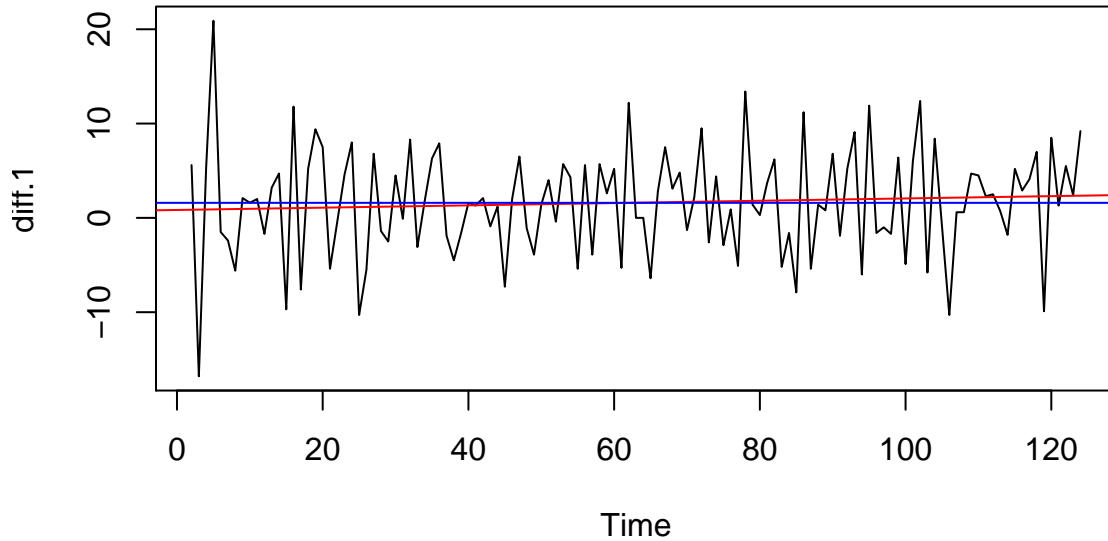
```
Coefficients:
```

	ar1	ma1	drift
	0.3288	-0.7556	1.5500
s.e.	0.1572	0.1139	0.1776

```
sigma^2 estimated as 28.52: log likelihood=-379.26
AIC=766.52 AICc=766.86 BIC=777.77
```

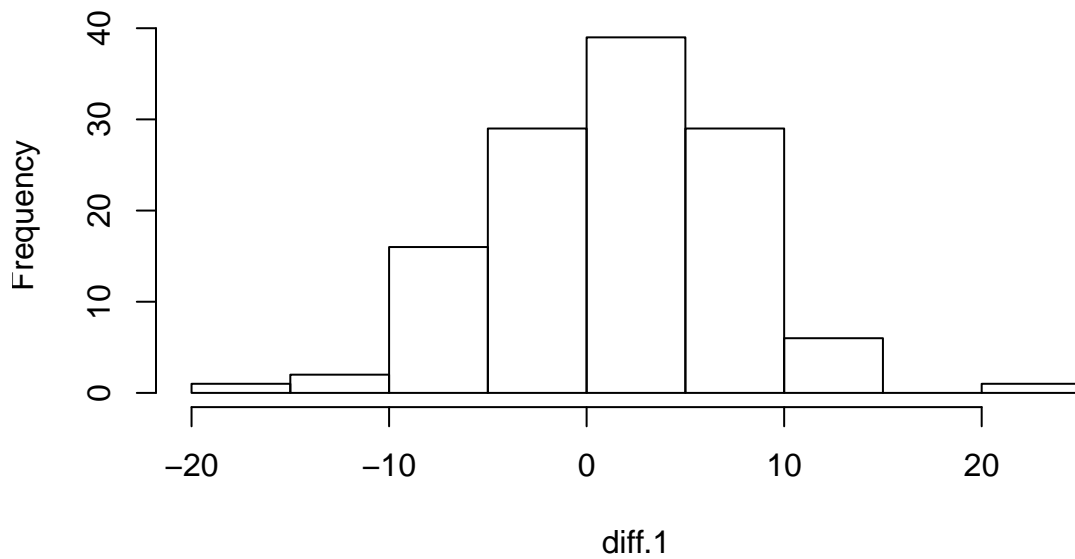
Let us proceed to difference the data at lag 1. Examining the new data plot, we see that the differenced data looks stationary and essentially has no trend.

Data after differencing at lag 1



The histogram also looks more normal, and the variance has desirably decreased from 2986.576 to 33.6458:

Histogram after differencing at lag 1

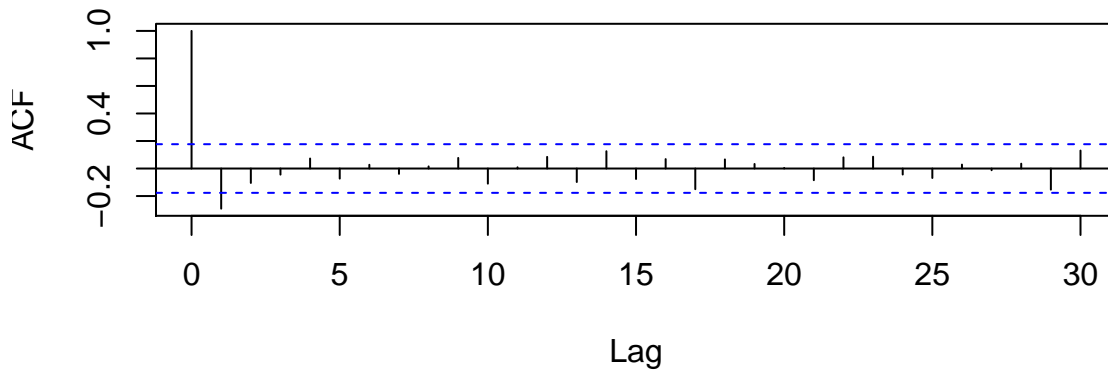


Variance of differenced data: 33.6458

Section III - ACF and PACF analysis

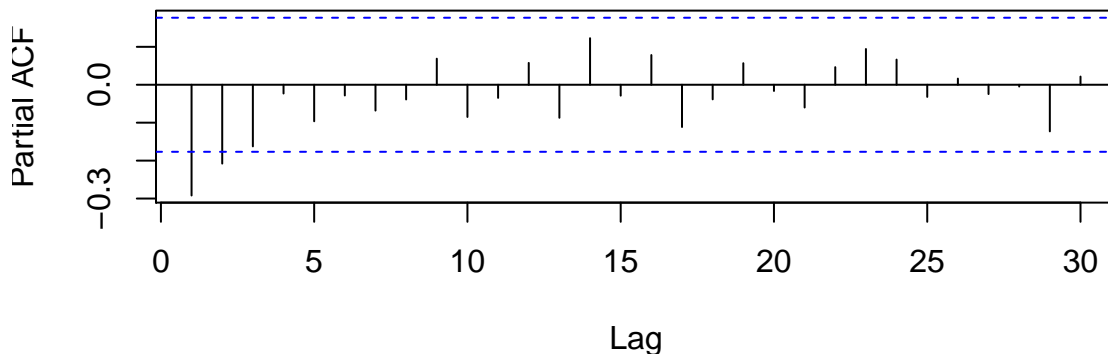
Analyzing the new ACF plot, we see that it no longer has a gradual decay. We suspect the data is stationary now and can proceed to identify a potential MA component. Since the only ACF that extends beyond the confidence interval is at lag 1, then we suspect $q=0$ or $q=1$.

ACF after differencing at lag 1



Let us now analyze the PACF plot to identify a potential AR component. We see that the PACF extends beyond the confidence interval at lag 1 and 2, so, we suspect $p=0$, $p=1$, or $p=2$.

PACF after differencing at lag 1



Thus, potential models are $ARIMA(0,1,0)$, $ARIMA(0,1,1)$, $ARIMA(1,1,0)$, $ARIMA(1,1,1)$, $ARIMA(2,1,0)$, and $ARIMA(2,1,1)$.

Let us fit each model and compare their AICs.

```
AIC for ARIMA(0,1,0): 791.5556
AIC for ARIMA(0,1,1): 787.7406
AIC for ARIMA(1,1,0): 788.1656
AIC for ARIMA(1,1,1): 789.7389
AIC for ARIMA(2,1,0): 789.5456
AIC for ARIMA(2,1,1): 791.545
```

The model with the lowest AIC is $ARIMA(0,1,1)$ followed by $ARIMA(1,1,0)$. Notice that these are also pretty low in their number of parameters, which is favorable by the principle of parsimony.

Thus, we proceed with these two models. If they fail, we can revisit this section to try other models.

Section IV - Model Fitting & Diagnostic Checking

Let us now check the model fit and estimate the coefficients, starting with ARIMA(0,1,1).

```
Call:
arima(x = train, order = c(0, 1, 1), method = "ML")

Coefficients:
      ma1
    -0.2209
s.e.    0.0862

sigma^2 estimated as 34.25:  log likelihood = -391.87,  aic = 787.74
```

The confidence interval of the coefficient `ma1` does not include 0, so we can consider this model as is. Since there is only an MA component, then we know it is stationary. Since this model only goes up to order 1 and the absolute value of the MA coefficient is also less than 1, then it does not have unit roots, so it is also invertible.

Next, we check ARIMA(1,1,0):

```
Call:
arima(x = train, order = c(1, 1, 0), method = "ML")

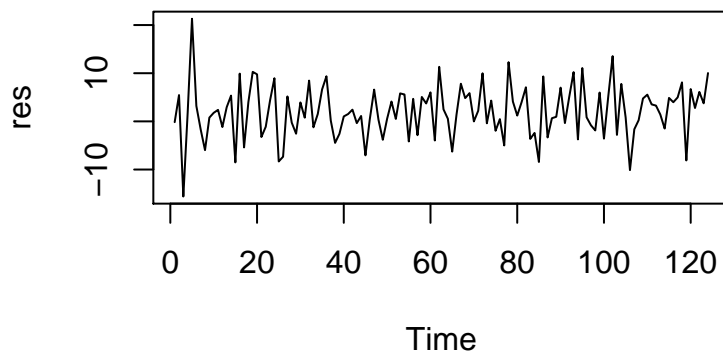
Coefficients:
      ar1
    -0.2089
s.e.    0.0890

sigma^2 estimated as 34.37:  log likelihood = -392.08,  aic = 788.17
```

The confidence interval of the coefficient `ar1` does not include 0 either, so we can consider this model as is. Since there is only an AR component, then we know it is invertible. Since this model only goes up to order 1 and the absolute value of the AR coefficient is also less than 1, then it does not have unit roots, so it is also stationary.

We now check the residual diagnostics, starting with ARIMA(0,1,1). If the residuals resemble Gaussian white noise, then the model fit is good.

ARIMA(0,1,1) Residuals



As shown above, the residuals resemble white noise, which is what we want.

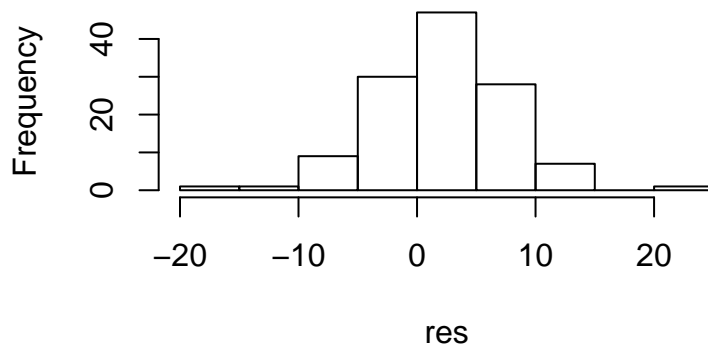
```
Call:
ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
Coefficients:
      1      2
-0.1548 -0.1649
```

```
Order selected 2  sigma^2 estimated as 29.33
```

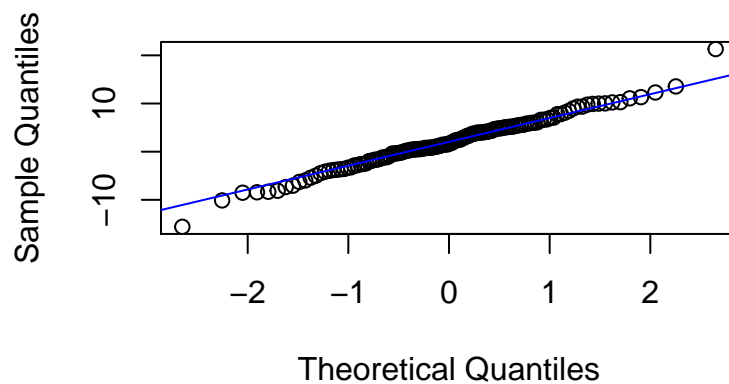
However, the residuals are automatically fitted to an AR(2) model, i.e. not white noise (AR(0)), so this test fails.

ARIMA(0,1,1) Histogram of residuals



The residuals appear normally distributed, which is what we want.

ARIMA(0,1,1) Q-Q plot of residuals



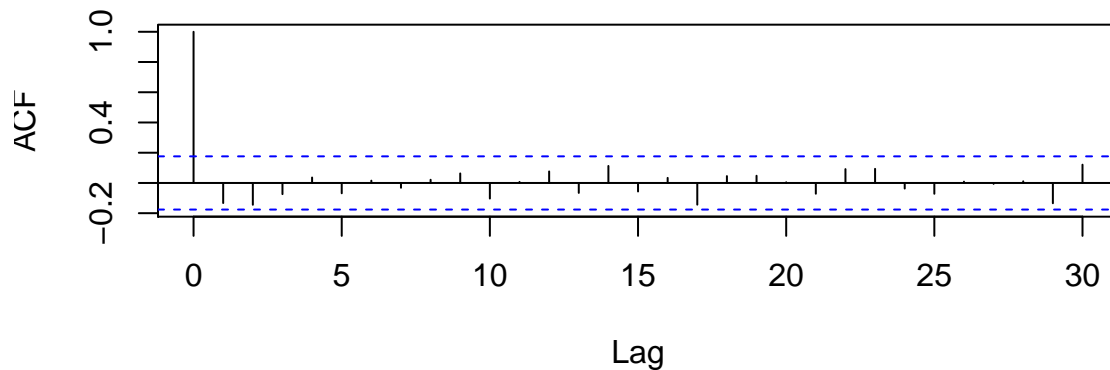
The Q-Q plot also appears normal.

Shapiro-Wilk normality test

```
data: res
W = 0.98958, p-value = 0.4727
```

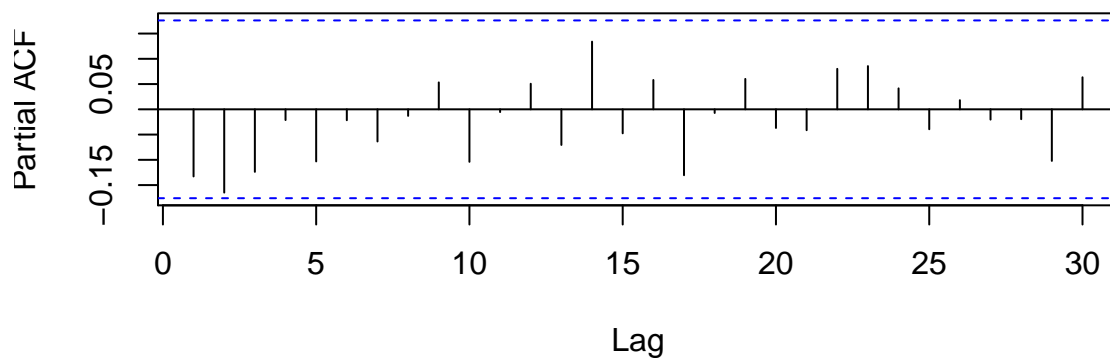
The p-value of the Shapiro-Wilk test exceeds 0.05. Thus, we conclude that the residuals are normally distributed.

ARIMA(0,1,1) ACF of residuals



As shown above, the residuals do not have an MA component since none of the lags past 0 exceed the 95% confidence interval.

ARIMA(0,1,1) PACF of residuals



As shown above, the residuals do not have an AR component since none of the lags past 0 exceed the 95% confidence interval.

Box-Pierce test

```
data: res
X-squared = 8.9486, df = 11, p-value = 0.6266
```

Box-Ljung test

```
data: res
X-squared = 9.4507, df = 11, p-value = 0.5804
```

Box-Ljung test

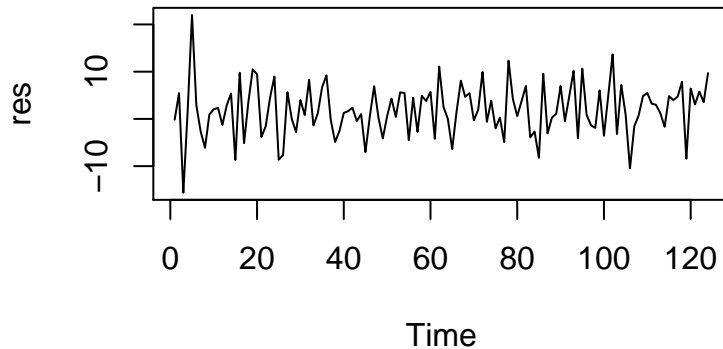
```
data: (res)^2
X-squared = 7.3011, df = 12, p-value = 0.8371
```

As shown above, the residuals pass all three Portmanteau tests (Box-Pierce, Ljung-Box, McLeod-Li) since the p-values all exceed 0.05.

To summarize, the residuals pass the normality tests but not the AR(0) test, indicating that the residuals may not be white noise and therefore the model may not be a good fit.

We now check the residual diagnostics for ARIMA(1,1,0).

ARIMA(1,1,0) Residuals



As shown above, the residuals resemble white noise, which is what we want.

Call:

```
ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

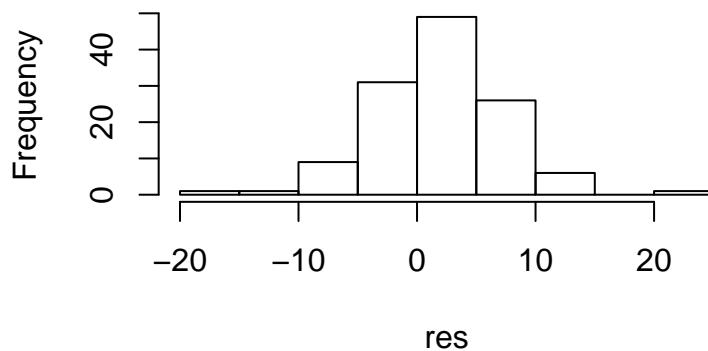
Coefficients:

```
      1      2  
-0.1581 -0.2070
```

Order selected 2 sigma^2 estimated as 29.4

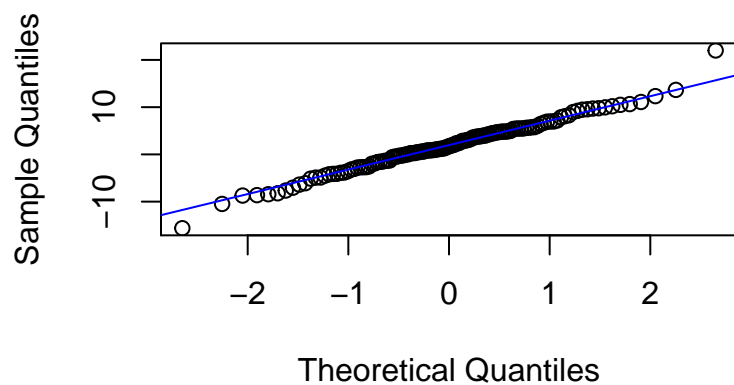
However, the residuals are once again automatically fitted to an AR(2) model, i.e. not white noise (AR(0)), so this test fails.

ARIMA(1,1,0) Histogram of residuals



The residuals appear normally distributed, which is what we want.

ARIMA(1,1,0) Q-Q plot of residuals



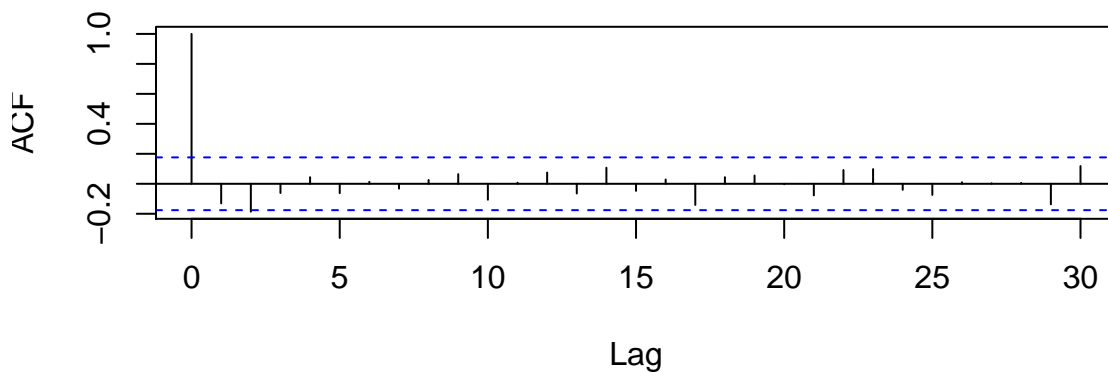
The Q-Q plot also appears normal.

Shapiro-Wilk normality test

```
data: res
W = 0.98846, p-value = 0.3829
```

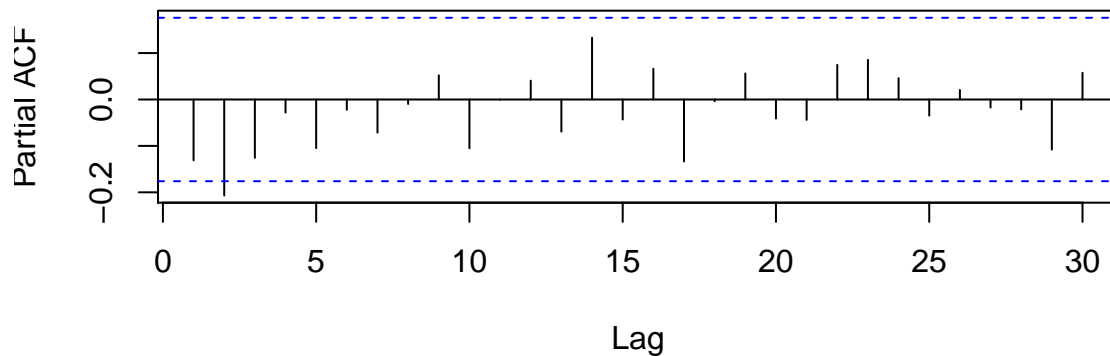
The p-value of the Shapiro-Wilk test exceeds 0.05. Thus, we conclude that the residuals are normally distributed.

ARIMA(1,1,0) ACF of residuals



As shown above, the residuals might have an MA component as demonstrated by the ACF at lag 2.

ARIMA(1,1,0) PACF of residuals



As shown above, the residuals might have an AR component as demonstrated by the PACF at lag 2.

Box-Pierce test

```
data: res
X-squared = 10.49, df = 11, p-value = 0.4869
```

Box-Ljung test

```
data: res
X-squared = 11.047, df = 11, p-value = 0.4393
```

Box-Ljung test

```
data: (res)^2
X-squared = 7.1273, df = 12, p-value = 0.8491
```

As shown above, the residuals pass all three Portmanteau tests (Box-Pierce, Ljung-Box, McLeod-Li) since the p-values all exceed 0.05.

To summarize, the residuals pass the normality tests but not all of the white noise tests, indicating that the model may not be a good fit.

Since the residuals of ARIMA(1,1,0) may have a AR(2) and/or an MA(2) component, we add a list of additional models to consider on top of the ones already stated in Section 3. New models include: ARIMA(3,1,0), ARIMA(3,1,1), ARIMA(1,1,2), ARIMA(2,1,2), ARIMA(3,1,2). Let's check their AICs and also compare them to the ones in Section 3.

Original Section 3 models:

```
AIC for ARIMA(0,1,0): 791.5556
AIC for ARIMA(0,1,1): 787.7406
AIC for ARIMA(1,1,0): 788.1656
AIC for ARIMA(1,1,1): 789.7389
AIC for ARIMA(2,1,0): 789.5456
AIC for ARIMA(2,1,1): 791.545
```

New models to consider:

AIC for ARIMA(3,1,0): 791.5426

AIC for ARIMA(3,1,1): 777.9474

AIC for ARIMA(1,1,2): 791.7398

AIC for ARIMA(2,1,2): 773.0074

AIC for ARIMA(3,1,2): 791.9663

We notice that ARIMA(3,1,1) and ARIMA(2,1,2) drastically stand out as having the lowest AICs of all 11 models under consideration. Since ARIMA(2,1,2) has the lower AIC and still has the same number of coefficients as ARIMA(3,1,1), let us proceed with ARIMA(2,1,2) for now. We can revisit other models later if necessary.

Let us now check the model fit and estimate the coefficients for ARIMA(2,1,2).

Call:

```
arima(x = train, order = c(2, 1, 2), method = "ML")
```

Coefficients:

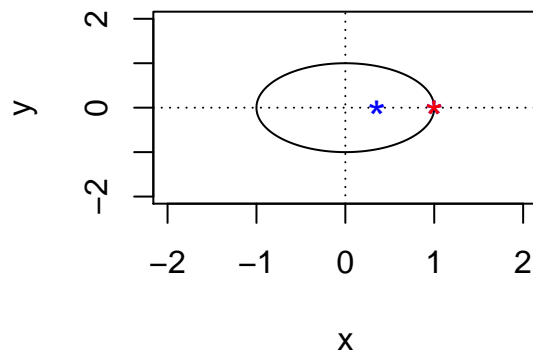
	ar1	ar2	ma1	ma2
	1.3544	-0.3549	-1.7602	0.7677
s.e.	0.1628	0.1628	0.1192	0.1181

sigma² estimated as 28.28: log likelihood = -381.5, aic = 773.01

None of the coefficient confidence intervals contain 0, so we can use the model as it is.

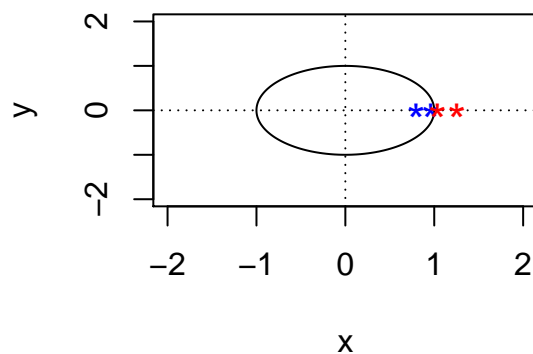
We now check ARIMA(2,1,2) for invertibility and stationarity via the unit circle test (all roots must be outside the unit circle) using the estimated coefficients found above.

ARIMA(2,1,2): Roots of AR part



AR roots: 1.000776-0i 2.81551+0i

ARIMA(2,1,2): Roots of MA part

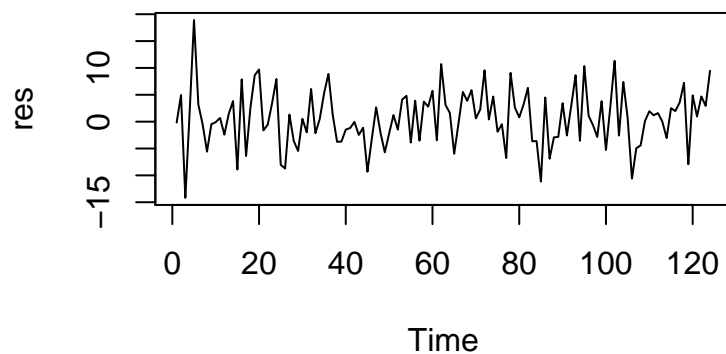


MA roots: 1.038398+0i 1.254425-0i

As shown above, all roots (both AR and MA) are outside the unit circle. Thus, we conclude that the test passes and ARIMA(2,1,2) is both stationary and invertible.

We now check the residual diagnostics for ARIMA(2,1,2).

ARIMA(2,1,2) Residuals



As shown above, the residuals resemble white noise, which is what we want.

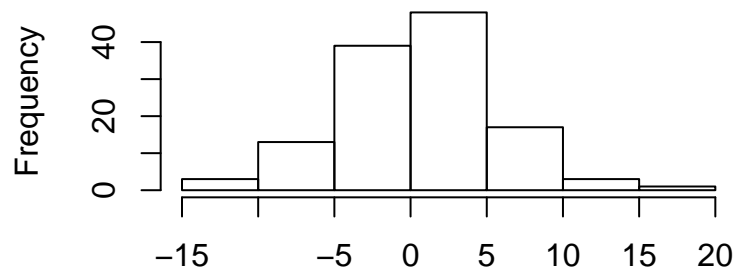
Call:

```
ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

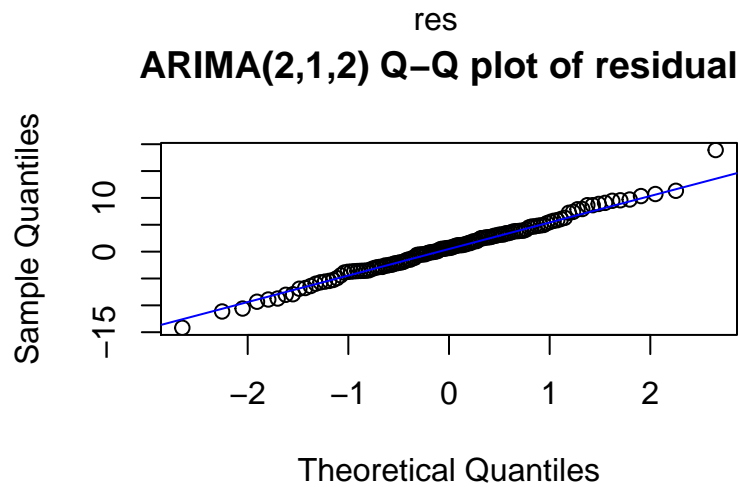
Order selected 0 sigma^2 estimated as 27.83

The residuals are automatically fitted to AR(0), i.e. white noise, so this test passes.

ARIMA(2,1,2) Histogram of residuals



ARIMA(2,1,2) Q-Q plot of residuals

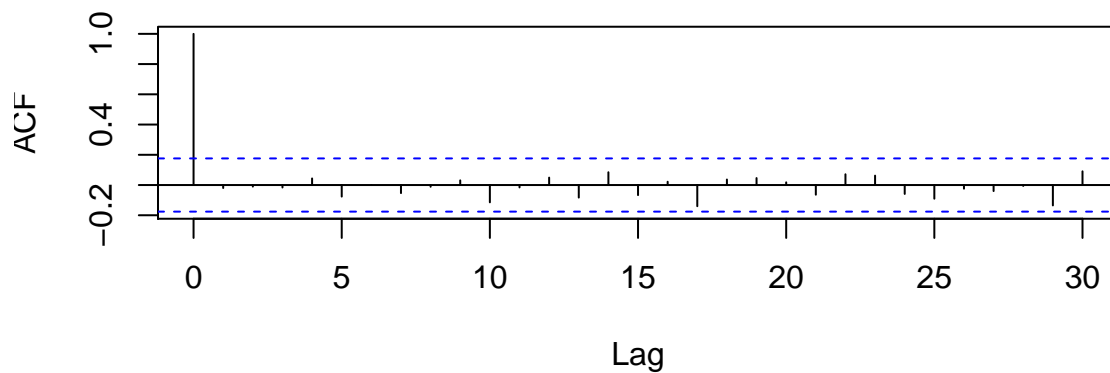


Shapiro-Wilk normality test

```
data: res
W = 0.99239, p-value = 0.7384
```

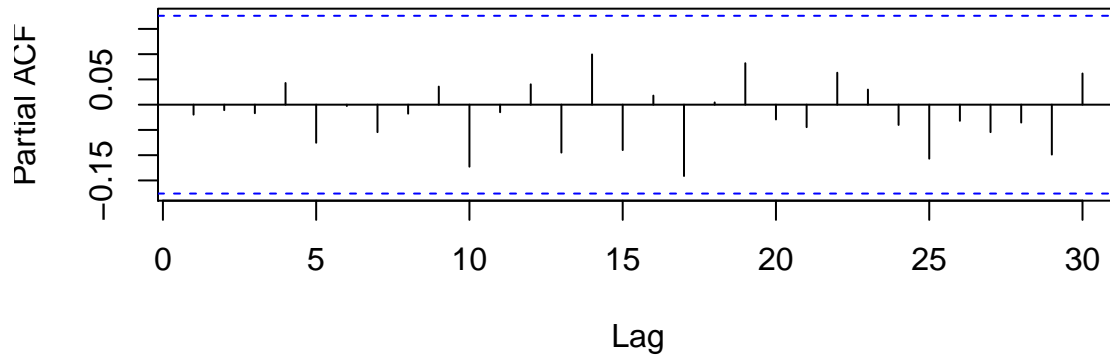
The histogram and Q-Q plot show the residuals to be normally distributed, and the p-value of the Shapiro-Wilk test also exceeds 0.05, so the residuals are normally distributed.

ARIMA(2,1,2) ACF of residuals



As shown above, the residuals themselves do not have an MA component.

ARIMA(2,1,2) PACF of residuals



Nor do they have an AR component.

Box-Pierce test

```
data: res
X-squared = 3.5071, df = 8, p-value = 0.8986
```

Box-Ljung test

```
data: res
X-squared = 3.8171, df = 8, p-value = 0.8732
```

Box-Ljung test

```
data: (res)^2
X-squared = 9.2155, df = 12, p-value = 0.6844
```

As shown above, the residuals pass all three Portmanteau tests (Box-Pierce, Ljung-Box, McLeod-Li) since the p-values all exceed 0.05.

ARIMA(2,1,2) passes all residual diagnostic tests, so the fit is good and the model is satisfactory, so this will be our final model. The algebraic equation is:

$$(1 - 1.3544B + 0.3549B^2)(1 - B)X_t = (1 - 1.7602B + 0.7677B^2)Z_t$$

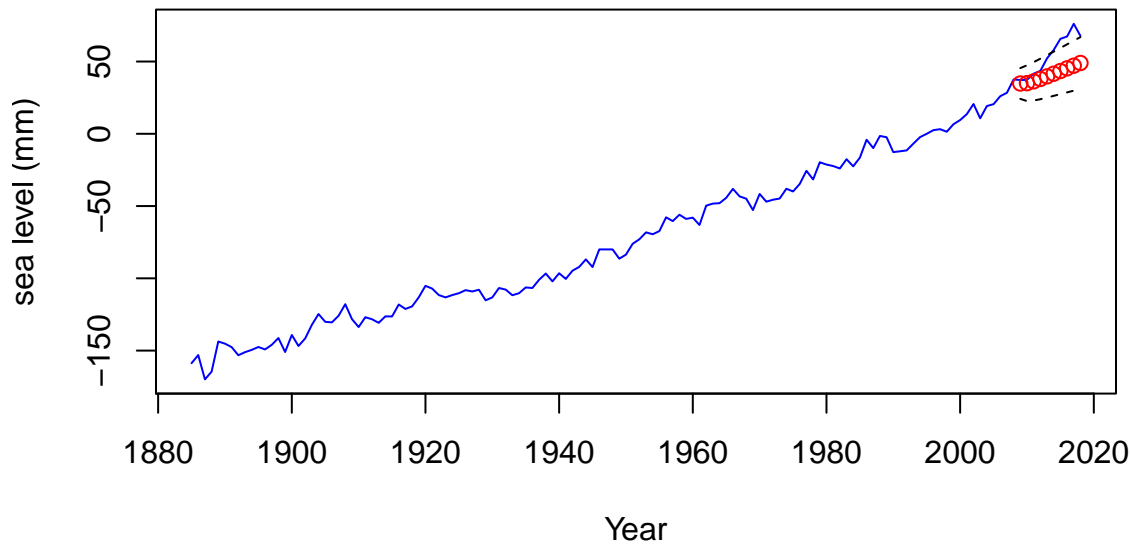
where X_t is the sea level at year t , Z_t is Gaussian white noise, and B is the backshift operator. The left-hand-side polynomial corresponds to the AR portion of the model while the right-hand-side polynomial corresponds to the MA portion, and $(1-B)$ corresponds to differencing once at lag 1. This final model was not one of the initial models suggested by ACF/PACF (in Section 3).

Note that ARIMA(3,1,1) does not need to be tried because even if it passes all residual diagnostics too, it still has a higher AIC, so we would still choose ARIMA(2,1,2) anyways. Further models with fewer parameters were also tried, but they continued to fail the AR(0) white noise test, so the results were omitted to prevent redundancy; these results can be supplied upon request. We now proceed to use ARIMA(2,1,2) for forecasting and compare the results with the hold-out test set's 10 observations (2004-2013).

Section V - Forecasting

Below, the forecasted values are shown in red, the 95% confidence intervals are represented by the black dashed lines, and the actual data is in blue.

ARIMA(2,1,2) Sea levels forecast



Observing the forecast, we see that the 95% confidence interval does not capture all of the true data, but it is fairly close. This is okay because time series forecast isn't meant to predict too far ahead. Furthermore, the true data shows that recent sea levels may be transitioning from a linear trend to more of an exponential trend, and this new information was not incorporated into the model since it was held out in the test set.

Conclusion

In conclusion, ARIMA(2,1,2) is decent for modeling global sea levels. 11 models were available for consideration, and ARIMA(2,1,2) had the lowest AIC and passed all residual diagnostic checks. The forecast for the next 10 observations is fairly accurate, with only a small portion of the true data being outside of the model's confidence intervals. This may be because sea levels have been rising at a faster rate in recent years. If this is true, then a new model can be considered in the future when there is more data to support this claim. But for the scope of this project, the goals were achieved.

The scientific community has long ago concluded that global warming is taking place, and rising sea levels are just one of the many pieces of evidence demonstrating this phenomenon. Rising sea levels can one day become detrimental to human society and the ecosystem, so continuing to combat global warming is crucial.

Special thanks to Professor Raya Feldman and TAs Nicole Yang and Jimin Lin for their informative lectures, section labs, and office hours that helped with this project.

References

Sea level data from Church and White (2011; modified 2013): http://www.cmar.csiro.au/sealevel/sl_data_cmar.html

Raya Feldman, Nicole Yang, Jimin Lin: 2020 UCSB PSTAT 174 Lectures, Labs, and Homework Material, Gauchospace

Appendix (All Code)

Section I - Plot & Analysis

```
# import data, convert into time-series object, split data into training/test
data <- read.csv("sea-level.csv",header=TRUE)
all <- ts(data[, 2], start=c(1880,6))
train <- ts(data[1:124, 2], start=c(1880,6))
test <- ts(data[125:134, 2], start=c(2004,6))
```

```
# plot training data
op <- par(pin=c(5,2))
plot(train, main='Sea levels', xlab='Year', ylab="sea level (mm)")
par(op)
```

```
# plot histogram
op <- par(pin=c(5,2))
hist(train, main="Histogram of data")
par(op)
```

```
cat('Variance of data:', var(train))
```

```
# plot ACF
op <- par(pin=c(5,1))
acf(train, lag.max=30, main="ACF of data")
par(op)
```

```
# plot PACF
op <- par(pin=c(5,1))
pacf(train, lag.max=30, main="PACF of data")
par(op)
```

Section II - Transformation and Differencing

```
# check recommended arima
library(forecast)
auto.arima(train)
```

```
# difference to get rid of trend
train <- ts(data[1:124, 2])
diff.1 <- diff(train, 1)
```

```
# plot differenced data
op <- par(pin=c(5,2))
plot(diff.1, main='Data after differencing at lag 1')
abline(lm(diff.1~as.numeric(1:length(diff.1))),col='red')
abline(h=mean(diff.1),col='blue')
par(op)
```

```
# plot histogram
op <- par(pin=c(5,2))
hist(diff.1, main='Histogram after differencing at lag 1')
par(op)
```

```
cat('Variance of differenced data:', var(diff.1))
```

Section III - ACF and PACF analysis

```
# plot ACF
op <- par(pin=c(5,1))
acf(diff.1, lag.max=30, main='ACF after differencing at lag 1')
par(op)

# plot PACF
op <- par(pin=c(5,1))
pacf(diff.1, lag.max=30, main='PACF after differencing at lag 1')
par(op)

# try out each model and compare their AICs
cat('AIC for ARIMA(0,1,0): ', arima(train, order=c(0,1,0), method="ML")$aic, '\n')
cat('AIC for ARIMA(0,1,1): ', arima(train, order=c(0,1,1), method="ML")$aic, '\n')
cat('AIC for ARIMA(1,1,0): ', arima(train, order=c(1,1,0), method="ML")$aic, '\n')
cat('AIC for ARIMA(1,1,1): ', arima(train, order=c(1,1,1), method="ML")$aic, '\n')
cat('AIC for ARIMA(2,1,0): ', arima(train, order=c(2,1,0), method="ML")$aic, '\n')
cat('AIC for ARIMA(2,1,1): ', arima(train, order=c(2,1,1), method="ML")$aic, '\n')
```

Section IV - Model Fitting & Diagnostic Checking

```
# ARIMA(0,1,1)
arima(train, order=c(0,1,1), method="ML")

# ARIMA(1,1,0)
arima(train, order=c(1,1,0), method="ML")

# ARIMA(0,1,1) residuals plot
op <- par(pin=c(3,1))
fit <- arima(train, order=c(0,1,1), method="ML")
res = residuals(fit)
ts.plot(res, main='ARIMA(0,1,1) Residuals')
par(op)

# ARIMA(0,1,1) AR model choosing
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

# ARIMA(0,1,1) histogram
op <- par(pin=c(3,1))
hist(res, main='ARIMA(0,1,1) Histogram of residuals')
par(op)

# ARIMA(0,1,1) Q-Q plot
op <- par(pin=c(3,1))
qqnorm(res, main='ARIMA(0,1,1) Q-Q plot of residuals')
qqline(res,col = "blue")
par(op)

# ARIMA(0,1,1) Shapiro-Wilk
shapiro.test(res)

# ARIMA(0,1,1) ACF
op <- par(pin=c(5,1))
acf(res, lag.max=30, main='ARIMA(0,1,1) ACF of residuals')
par(op)
```

```

# ARIMA(0,1,1) PACF
op <- par(pin=c(5,1))
pacf(res, lag.max=30, main='ARIMA(0,1,1) PACF of residuals')
par(op)

# ARIMA(0,1,1) Portmanteau tests
Box.test(res, lag=12, type= "Box-Pierce", fitdf = 1) # Box-Pierce test
Box.test(res, lag=12, type="Ljung-Box", fitdf = 1) # Ljung-Box test
Box.test((res)^2, lag=12, type="Ljung-Box", fitdf = 0) # McLeod-Li test: Ljung-Box for squares

# ARIMA(1,1,0) residuals plot
op <- par(pin=c(3,1))
fit <- arima(train, order=c(1,1,0), method="ML")
res = residuals(fit)
ts.plot(res, main='ARIMA(1,1,0) Residuals')
par(op)

# ARIMA(1,1,0) AR model choosing
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

# ARIMA(1,1,0) histogram
op <- par(pin=c(3,1))
hist(res, main='ARIMA(1,1,0) Histogram of residuals')
par(op)

# ARIMA(1,1,0) Q-Q plot
op <- par(pin=c(3,1))
qqnorm(res, main='ARIMA(1,1,0) Q-Q plot of residuals')
qqline(res,col ="blue")
par(op)

# ARIMA(1,1,0) Shapiro-Wilk
shapiro.test(res)

# ARIMA(1,1,0) ACF
op <- par(pin=c(5,1))
acf(res, lag.max=30, main='ARIMA(1,1,0) ACF of residuals')
par(op)

# ARIMA(1,1,0) PACF
op <- par(pin=c(5,1))
pacf(res, lag.max=30, main='ARIMA(1,1,0) PACF of residuals')
par(op)

# ARIMA(1,1,0) Portmanteau tests
Box.test(res, lag=12, type= "Box-Pierce", fitdf = 1) # Box-Pierce test
Box.test(res, lag=12, type="Ljung-Box", fitdf = 1) # Ljung-Box test
Box.test((res)^2, lag=12, type="Ljung-Box", fitdf = 0) # McLeod-Li test: Ljung-Box for squares

# original Section 3 models
cat('Original Section 3 models:\n')
cat('AIC for ARIMA(0,1,0): ', arima(train, order=c(0,1,0), method="ML")$aic, '\n')
cat('AIC for ARIMA(0,1,1): ', arima(train, order=c(0,1,1), method="ML")$aic, '\n')
cat('AIC for ARIMA(1,1,0): ', arima(train, order=c(1,1,0), method="ML")$aic, '\n')
cat('AIC for ARIMA(1,1,1): ', arima(train, order=c(1,1,1), method="ML")$aic, '\n')
cat('AIC for ARIMA(2,1,0): ', arima(train, order=c(2,1,0), method="ML")$aic, '\n')

```

```
cat('AIC for ARIMA(2,1,1): ', arima(train, order=c(2,1,1), method="ML")$aic, '\n')
```

```
# new models to consider
```

```
cat('\nNew models to consider:\n')
```

```
cat('AIC for ARIMA(3,1,0): ', arima(train, order=c(3,1,0), method="ML")$aic, '\n')
```

```
cat('AIC for ARIMA(3,1,1): ', arima(train, order=c(3,1,1), method="ML")$aic, '\n')
```

```
cat('AIC for ARIMA(1,1,2): ', arima(train, order=c(1,1,2), method="ML")$aic, '\n')
```

```
cat('AIC for ARIMA(2,1,2): ', arima(train, order=c(2,1,2), method="ML")$aic, '\n')
```

```
cat('AIC for ARIMA(3,1,2): ', arima(train, order=c(3,1,2), method="ML")$aic, '\n')
```

```
# ARIMA(2,1,2)
```

```
arima(train, order=c(2,1,2), method="ML")
```

```
# ARIMA(2,1,2)
```

```
op <- par(pin=c(2,1))
```

```
source("plot.roots.R")
```

```
plot.roots(NULL, polyroot(c(1, -1.3544, 0.3549)), main="ARIMA(2,1,2): Roots of AR part") # check roots f
```

```
par(op)
```

```
cat('AR roots:', polyroot(c(1, -1.3544, 0.3549)), '\n')
```

```
# ARIMA(2,1,2)
```

```
op <- par(pin=c(2,1))
```

```
source("plot.roots.R")
```

```
plot.roots(NULL, polyroot(c(1, -1.7602, 0.7677)), main="ARIMA(2,1,2): Roots of MA part") # check roots f
```

```
par(op)
```

```
cat('MA roots:', polyroot(c(1, -1.7602, 0.7677)))
```

```
# ARIMA(2,1,2) residuals plot
```

```
op <- par(pin=c(3,1))
```

```
fit <- arima(train, order=c(2,1,2), method="ML")
```

```
res = residuals(fit)
```

```
ts.plot(res, main='ARIMA(2,1,2) Residuals')
```

```
par(op)
```

```
# ARIMA(2,1,2) AR model choosing
```

```
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
# ARIMA(2,1,2) histogram
```

```
op <- par(pin=c(3,1))
```

```
hist(res, main='ARIMA(2,1,2) Histogram of residuals')
```

```
par(op)
```

```
# ARIMA(2,1,2) Q-Q plot
```

```
op <- par(pin=c(3,1))
```

```
qqnorm(res, main='ARIMA(2,1,2) Q-Q plot of residuals')
```

```
qqline(res, col = "blue")
```

```
par(op)
```

```
# ARIMA(2,1,2) Shapiro-Wilk
```

```
shapiro.test(res)
```

```
# ARIMA(2,1,2) ACF
```

```
op <- par(pin=c(5,1))
```

```

acf(res, lag.max=30, main='ARIMA(2,1,2) ACF of residuals')
par(op)

# ARIMA(2,1,2) PACF
op <- par(pin=c(5,1))
pacf(res, lag.max=30, main='ARIMA(2,1,2) PACF of residuals')
par(op)

# ARIMA(2,1,2) Portmanteau tests
Box.test(res, lag=12, type= "Box-Pierce", fitdf = 4) # Box-Pierce test
Box.test(res, lag=12, type="Ljung-Box", fitdf = 4) # Ljung-Box test
Box.test((res)^2, lag=12, type="Ljung-Box", fitdf = 0) # McLeod-Li test: Ljung-Box for squares

```

Section V - Forecasting

```

# use ARIMA(2,1,2) to forecast the next 10 values
data <- read.csv("sea-level.csv",header=TRUE)
train <- ts(data[1:124, 2], start=c(1880,6))
test <- ts(data[125:134, 2])
all <- ts(data[, 2], start=c(1880,6))
fit <- arima(train, order=c(2,1,2), method="ML")
prediction <- predict(fit, n.ahead=10)

# 95% confidence intervals
UB <- prediction$pred + 2*prediction$se
LB <- prediction$pred - 2*prediction$se

# plot results
# ts.plot(train, xlim=c(1, length(train)+27), ylim=c(min(train), max(UB)))
op <- par(pin=c(5,2))
ts.plot(all, col='blue', main='ARIMA(2,1,2) Sea levels forecast', xlab='Year', ylab="sea level (mm)")
lines(UB, col='black', lty='dashed')
lines(LB, col='black', lty='dashed')
points(prediction$pred, col='red')
par(op)

```