SINGAPORE MANAGEMENT UNIVERSITY

# DSA301 (G1): Time Series Data Analysis

## Group Project

**Prepared for**

**Professor Benjamin EE**

**Prepared by**

**Group 3**


Chan Zhi Cheng Clarice (01398710)

Gao Qianyun (01449320)

Gloria Goh Su Yi (01436260)

Qistina Purnamasari (01443112)

# 1. Introduction

Currently, climate change stands out as one of the most pressing global challenges. If left unaddressed, it poses a significant threat to long-term growth and prosperity, directly impacting economic stability. Therefore, employing time series data analysis, our project aims to investigate the relationship between climate change and three crucial variables: *greenhouse gas emissions*, *mean sea level*, and *anomalies in land-surface air and sea-surface water temperatures*, by examining the patterns and trends within these variables. The dataset used encompasses global, yearly records spanning from 1880 to 2020. Furthermore, we have conducted forecasts for these variables over the next 80 years, providing insights into the persistence or potential mitigation of the climate change issue in the foreseeable future.

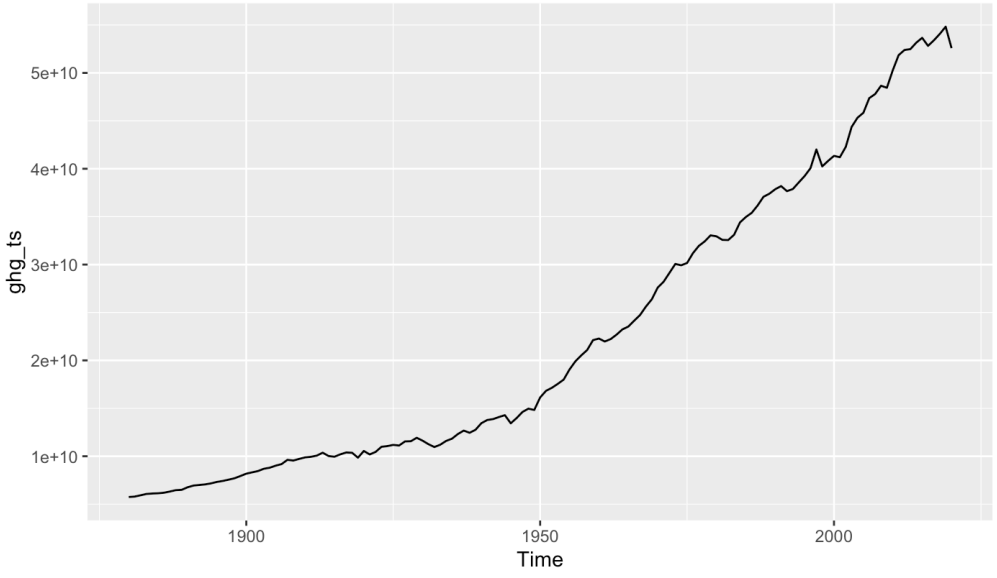
# 2. Read data into R

## 2.1 Convert to Time Series

We first convert our combined data, as well as all three individual variables (greenhouse gas, mean sea level, land-surface air and sea-surface water temperature anomalies) into time series and used autoplot to visualise how each of these individual variables evolve over time, and to identify any patterns/trends if any.

### *2.1.1 Greenhouse gas emissions*

```
ghg_ts = ts(data = data['ghg.emission'], start = 1880, frequency = 1)
autoplot(ghg_ts)
```

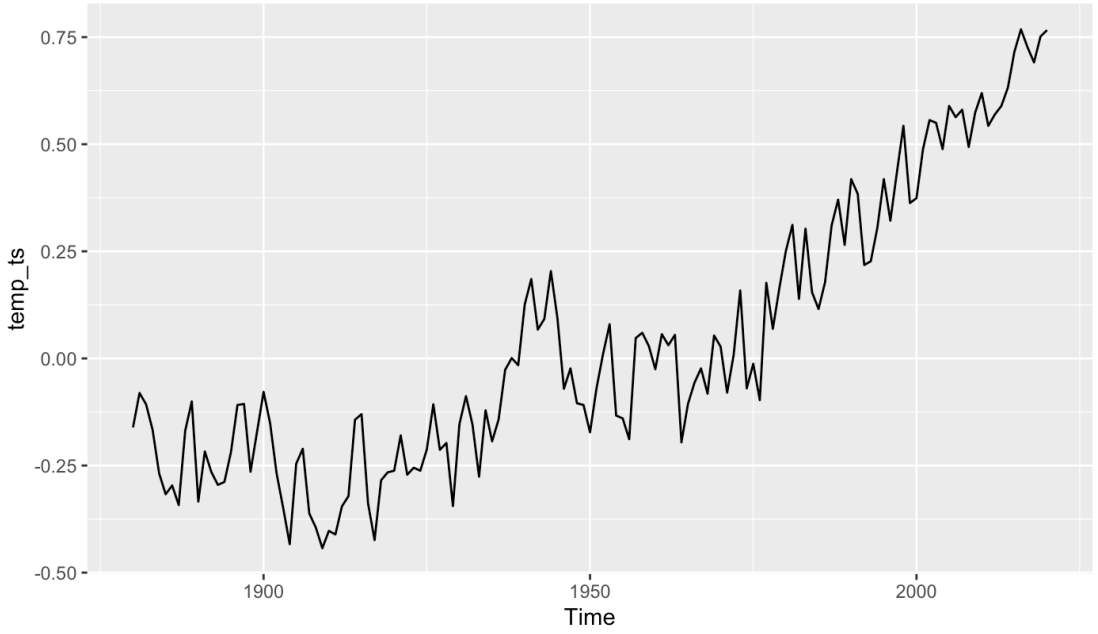


We notice an accelerating upward trend in greenhouse gas emissions. This is probably due to increasing human activities. The primary sources of GHG emissions include burning fossil fuels for electricity, heat, and transportation, as well as deforestation, agriculture, and industrial processes. We also hypothesise that this will contribute to an increase in the surface temperature of the Earth, as this increase in greenhouse gas emissions will trap heat from the sun in the Earth's atmosphere.

*2.1.2 Temperature anomalies*

This data consists of monthly mean temperature anomalies, expressed as deviations from the corresponding 1951-1980 means, for combined land-surface air and sea-surface water temperatures (Land-Ocean Temperature Index). To pre-process the temperature anomalies data, we converted the dataset using the tanh() function to remove extreme outliers, and to ensure that all the values are between -1 and 1.

```
temp_ts = ts(data = data['temp.anomalies'], start = 1880, frequency = 1)
temp_ts = tanh(temp_ts)
autoplot(temp_ts)
```

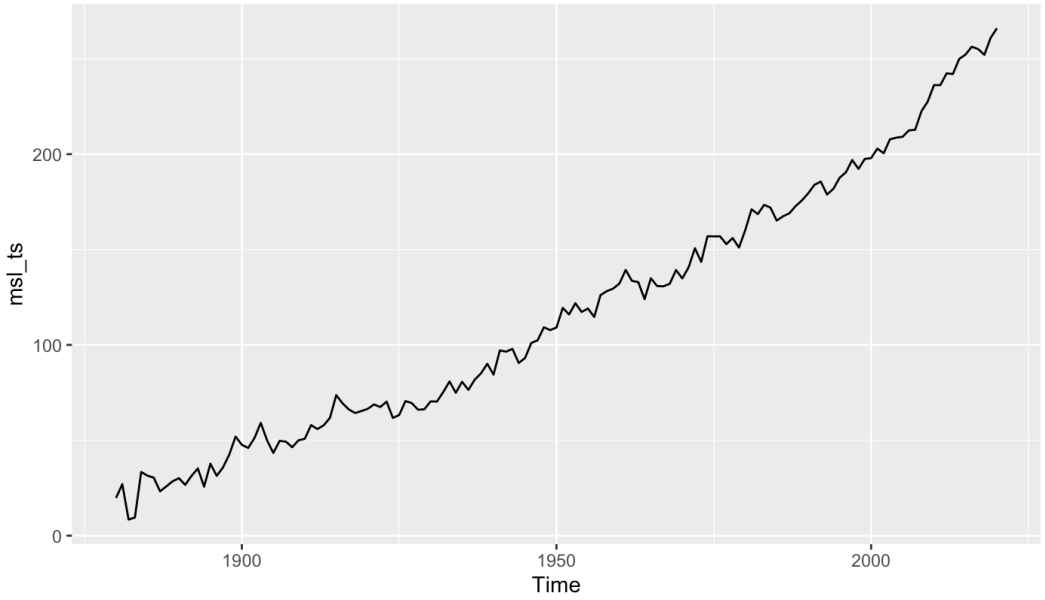



Similarly, we also observe an increasing upward trend. This can be attributed to the accelerating increase in greenhouse gas emissions as mentioned in 2.1.1, increasing the surface temperature of the Earth. This is otherwise known as global warming. This in turn increases the mean sea level over the years as well.

*2.1.3 Mean sea level*

The values given in this dataset are shown as change in sea level compared to the 1993-2008 average, which explains why some values are negative. Hence, we pre-processed the data to ensure the values are not negative. Therefore, we chose to add a constant of 200, as our lowest value in the dataset is -183.

```
msl_ts = ts(data = data['mean.sea.level'], start = 1880, frequency = 1)
msl_ts = msl_ts + 200
autoplot(msl_ts)
```




We also observe an upward trend in mean sea level over the past 140 years. This is probably due to thermal expansion and melting of land-based ice, caused by the increase in Earth's temperature.

*2.1.4 Full dataset*

We will now investigate how all these variables affect one another by looking at all three datasets together.

```
data_ts = ts(data, start = 1880, frequency = 1)
View(data_ts)
```

## 3. Test for stationarity

We used the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test to evaluate if the mean and variance of a time series are constant over time. The null hypothesis suggests that the data is stationary. Hence, if our test statistic is more than 0.05, this indicates that our null hypothesis is rejected and hence differencing is required.

3.1 Unit root test

```
library(urca)
ur.kpss(ghg_ts)
```

The value of the test statistic is: 2.7334.

```
ur.kpss(temp_ts)
```

The value of the test statistic is: 2.4629.

```
ur.kpss(msl_ts)
```

The value of the test statistic is: 2.8499.

Since all of the above three t-statistics exceed the critical value at our chosen significance level of 5%, it therefore suggests that data for the three variables are not stationary. Hence, we would have to do differencing.

## 4. Differencing

We differenced our data to remove trends and/or seasonality, to make it stationary. Given our non-seasonal data (since our dataset primarily focuses on macro-level phenomena, the influence of seasonal fluctuations might be negligible), we only use the ndiffs() function instead of nsdiffs() for this evaluation.

4.1 Find the number of differencing required

```
ndiffs(ghg_ts) # 2
ndiffs(temp_ts) # 1
ndiffs(msl_ts) # 1
```

After determining the necessary number of differences, we adjusted our data accordingly and performed the KPSS test once more to confirm the stationarity of our data.

4.2 Check for stationarity again

```
diff_ghg = diff(diff(ghg_ts))
```

```
diff_temp = diff(temp_ts)
diff_msl = diff(msl_ts)
```

```
ur.kpss(diff_ghg)
```
The p-value is 0.1383.

```
ur.kpss(diff_temp)
```
The p-value is 0.1664.

```
ur.kpss(diff_msl)
```
The p-value is 0.4139.

As all three of our p-values are more than 0.05, this indicates that we fail to reject the null hypothesis, implying that our data exhibits stationarity.

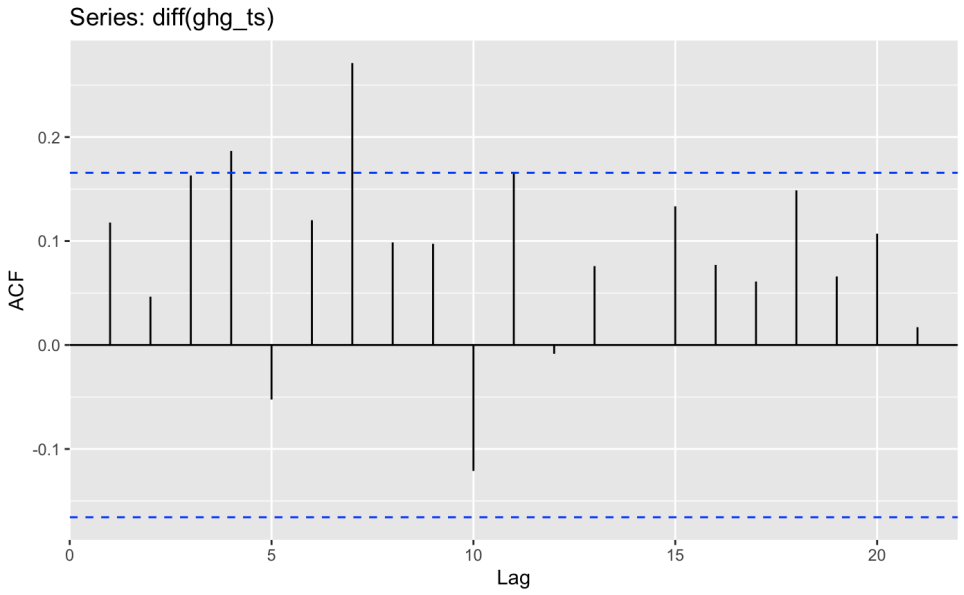
# 5. Univariate time analysis

## 5.1 Greenhouse Gas Emissions (GHG)

### *5.1.1 ACF/PACF*

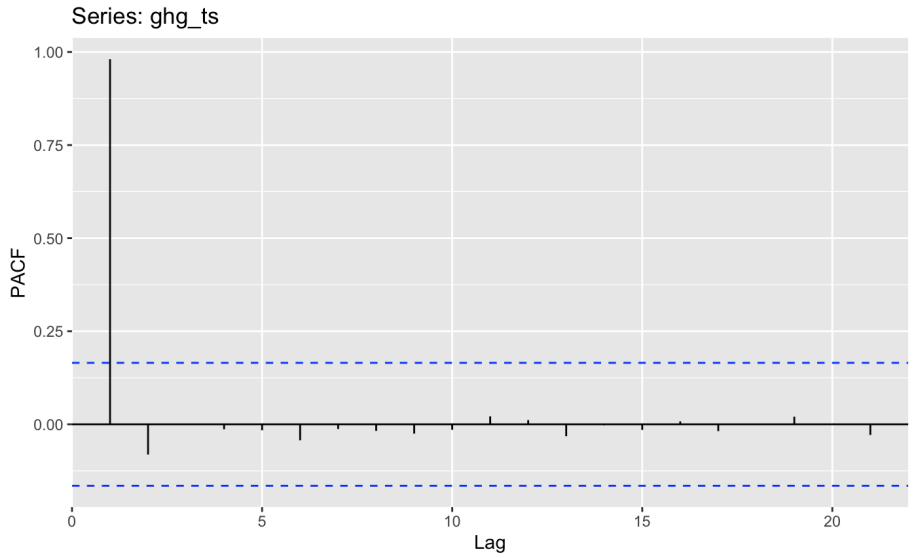
We use ACF and PACF to help identify patterns and relationships between observations at different time lags. By examining the decay or cut-off patterns in the ACF and PACF plots, we can determine the order of autoregressive (AR) and moving average (MA) components in models like ARIMA (Autoregressive Integrated Moving Average).

```
ggAcf(diff(ghg_ts))
```



As the 4th lag corresponds to the last significant spike before the ACF values drop within the confidence interval (blue dotted line), it indicates the number of terms in the MA model, in this case, MA(4).

```
ggPacf(ghg_ts)
```




From this PACF graph, we see there is 1 significant spike at lag 1 and the PACF values fall sharply within the confidence interval afterwards. This suggests an AR(1) model.

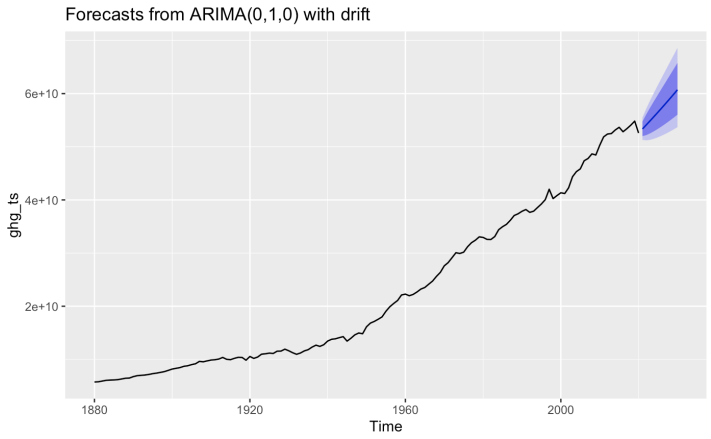
*5.1.2 Auto Arima*

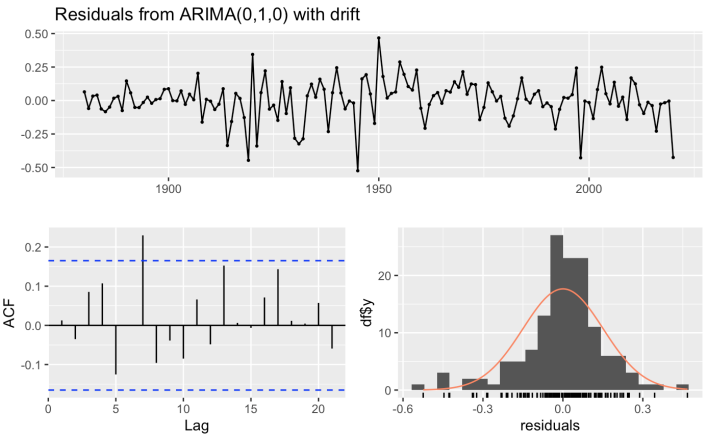
We ran auto Arima as it is convenient for automatic model selection since it automatically searches through a range of possible ARIMA model specifications to identify the best-fitting model based on selected criteria (e.g. AIC, BIC).

```
# auto.arima (0, 1, 0) MAPE: 2.874838 / AICc = -129.11
model_ghg2 = auto.arima(ghg_ts, lambda = "auto")
summary(model_ghg2)
autoplot(forecast(model_ghg2)) # (0, 1, 0)
checkresiduals(model_ghg2)
ghg_split = ts_split(ghg_ts, sample.out = 80)
accuracy(x = ghg_split$test, forecast(auto.arima(ghg_split$train)))
```

```
Series: ghg_ts1
ARIMA(0,1,0) with drift
Box Cox transformation: lambda= 0.0819848

Coefficients:
      drift
     0.1094
s.e.  0.0127

sigma^2 = 0.0228:  log likelihood = 66.6
AIC=-129.2   AICc=-129.11   BIC=-123.31

Training set error measures:
                    ME      RMSE       MAE         MPE     MAPE      MASE       ACF1
Training set -10078753 541373695 330494607 -0.0168317 1.534341 0.6861922 0.03589983
```

```
        Ljung-Box test

data:  Residuals from ARIMA(0,2,2)
Q* = 11.383, df = 8, p-value = 0.181

Model df: 2.   Total lags used: 10
```




| | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 | Theil's U |
|---|---|---|---|---|---|---|---|---|
| Training set | -0.05403026 | 5.656736 | 4.245517 | -0.07441686 | 2.872108 | 0.8648468 | 0.1674349 | NA |
| Test set | 8.06832577 | 9.097516 | 8.104091 | -8.37420205 | 8.406872 | 1.6508701 | 0.4945517 | 2.136591 |

### *5.1.3 Manual Arima*

However, running auto Arima may not capture all relevant aspects of the data, hence we decided to run manual Arima as well (for all three of our models) as it allowed for manual specification of our model parameters, which gives us more flexibility and control over the modelling process.

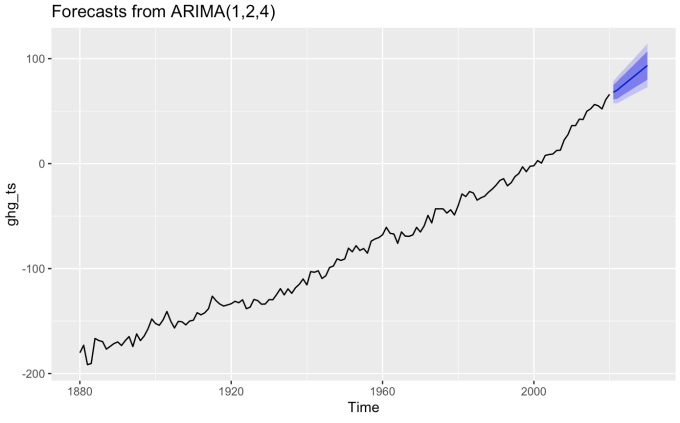
Hence, based on the results from our ACF and PACF graph, we have decided to go with ARIMA (1, 2, 4) and try ARIMA (2, 2, 4) and ARIMA (0, 2, 4) for our Manual Arima model as well.

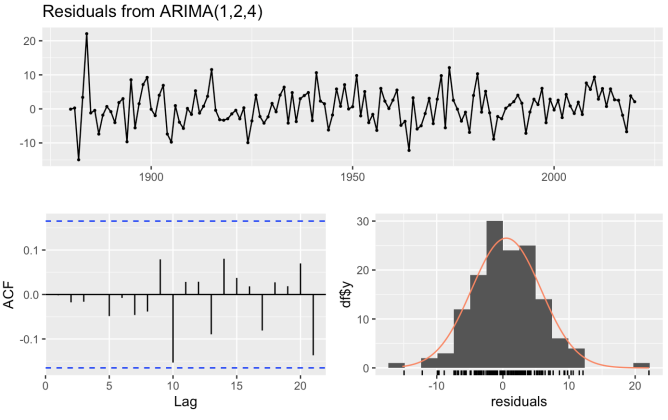
# Manual Arima: GHG (1, 2, 4) MAPE: 3.222849 / AICc = 5998.29

```
model_ghg = Arima(ghg_ts, order = c(1, 2, 4), include.constant = TRUE, include.drift = TRUE)

summary(model_ghg)

autoplot(forecast(model_ghg))

checkresiduals(model_ghg)

ghg_split = ts_split(ghg_ts, sample.out = 80)

accuracy(x = ghg_split$test[,1:1], forecast(Arima(ghg_split$train[,1:1], order = c(1, 2, 4), include.constant = TRUE)))
```

```
Series: ghg_ts
ARIMA(1,2,4)

Coefficients:
          ar1      ma1      ma2     ma3     ma4
      -0.6643  -0.6939  -0.6596  0.1892  0.2042
s.e.   0.2485   0.2524   0.3449  0.1357  0.0989

sigma^2 = 28.94:  log likelihood = -430.71
AIC=873.42   AICc=874.06   BIC=891.03

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE         ACF1
Training set 0.5166682 5.244451 3.99874 -4.560687 13.81976 0.8500234 -0.001772992
```

```
        Ljung-Box test

data:  Residuals from ARIMA(1,2,4)
Q* = 5.5515, df = 5, p-value = 0.3523

Model df: 5.   Total lags used: 10
```



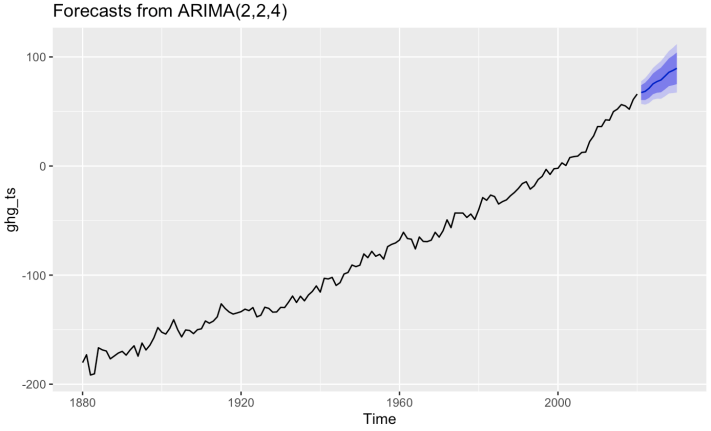
```
                    ME      RMSE       MAE         MPE     MAPE      MASE        ACF1 Theil's U
Training set -0.4309499  5.353468 3.961277  0.2339902 2.702733 0.8069449 -0.00410995        NA
Test set      9.4007415 10.319261 9.400741 -9.7168666 9.716867 1.9150086  0.50217777  2.399182
```

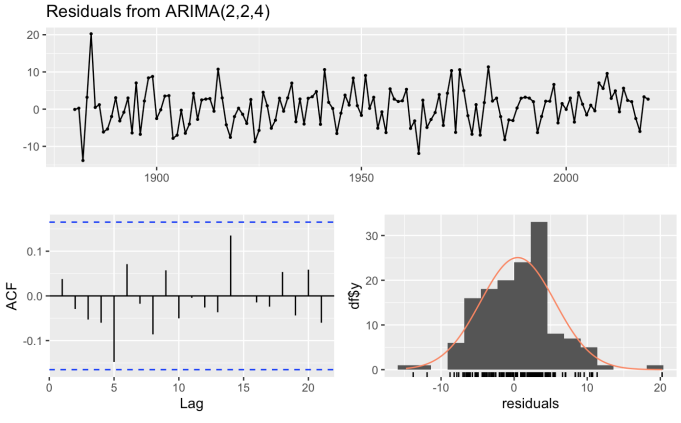
# BEST Manual Arima: GHG (2, 2, 4) MAPE: 3.395982 / AICc = 5998.53

```
model2_ghg = Arima(ghg_ts, order = c(2, 2, 4), include.constant = TRUE)

summary(model2_ghg)

autoplot(forecast(model2_ghg))

checkresiduals(model2_ghg)

ghg_split = ts_split(ghg_ts, sample.out = 80)

accuracy(x = ghg_split$test[,1:1], forecast(Arima(ghg_split$train[,1:1], order = c(2, 2, 4), include.constant = TRUE)))
```

```
Series: ghg_ts1
ARIMA(0,1,0) with drift
Box Cox transformation: lambda= 0.0819848

Coefficients:
      drift
     0.1094
s.e.  0.0127

sigma^2 = 0.0228:  log likelihood = 66.6
AIC=-129.2   AICc=-129.11   BIC=-123.31

Training set error measures:
                    ME      RMSE       MAE        MPE     MAPE      MASE       ACF1
Training set -10078753 541373695 330494607 -0.0168317 1.534341 0.6861922 0.03589983
```

```
        Ljung-Box test

data:  Residuals from ARIMA(2,2,4)
Q* = 7.3196, df = 4, p-value = 0.1199

Model df: 6.   Total lags used: 10
```



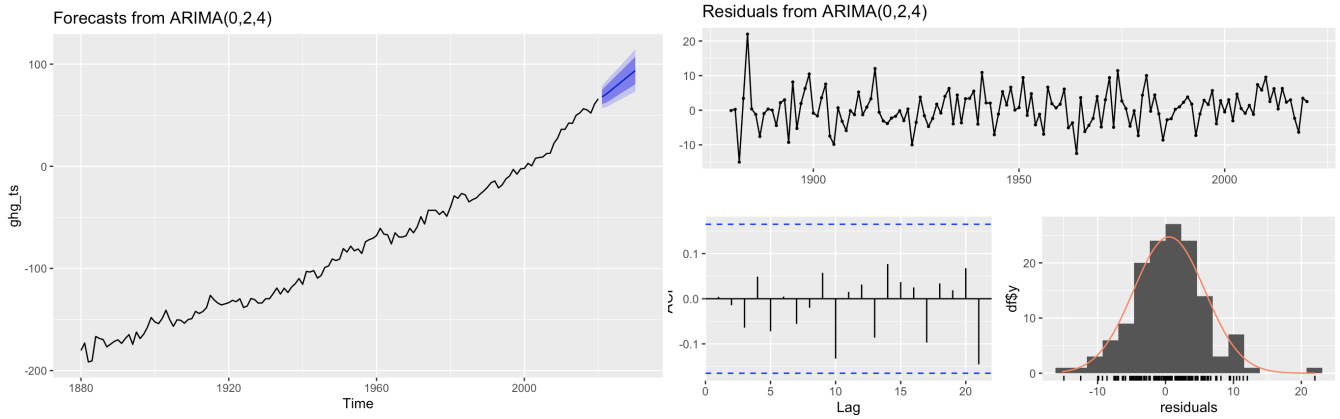
| | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 | Theil's U |
|---|---|---|---|---|---|---|---|---|
| Training set | -0.05403026 | 5.656736 | 4.245517 | -0.07441686 | 2.872108 | 0.8648468 | 0.1674349 | NA |
| Test set | 8.06832577 | 9.097516 | 8.104091 | -8.37420205 | 8.406872 | 1.6508701 | 0.4945517 | 2.136591 |

# Manual Arima: GHG (0, 2, 4) MAPE: 3.220900 / AICc = 5997.29 (LOWEST AIC & MAPE)

```
model3_ghg = Arima(ghg_ts, order = c(0, 2, 4), include.constant = TRUE, include.drift = TRUE)

summary(model3_ghg)

autoplot(forecast(model3_ghg))

checkresiduals(model3_ghg)

ghg_split = ts_split(ghg_ts, sample.out = 80)

accuracy(x = ghg_split$test[,1:1], forecast(Arima(ghg_split$train[,1:1], order = c(0, 2, 4), include.constant = TRUE)))
```

```
Series: ghg_ts
ARIMA(0,2,4)

Coefficients:
          ma1     ma2     ma3     ma4
      -1.3711  0.2607  0.0708  0.0638
s.e.   0.0868  0.1398  0.1342  0.0866

sigma^2 = 28.99:  log likelihood = -431.32
AIC=872.63   AICc=873.08   BIC=887.3

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE        ACF1
Training set 0.5400928 5.268011 4.003012 -5.068068 14.61649 0.8509314 0.004331155
```

```
        Ljung-Box test

data:  Residuals from ARIMA(0,2,4)
Q* = 5.5006, df = 6, p-value = 0.4814

Model df: 4.   Total lags used: 10
```




| | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 | Theil's U |
|---|---|---|---|---|---|---|---|---|
| Training set | -0.3649068 | 5.371124 | 3.945442 | 0.1843644 | 2.697747 | 0.8037192 | 0.0180249 | NA |
| Test set | 9.3803316 | 10.310132 | 9.380332 | -9.6962496 | 9.696250 | 1.9108509 | 0.4877137 | 2.389482 |

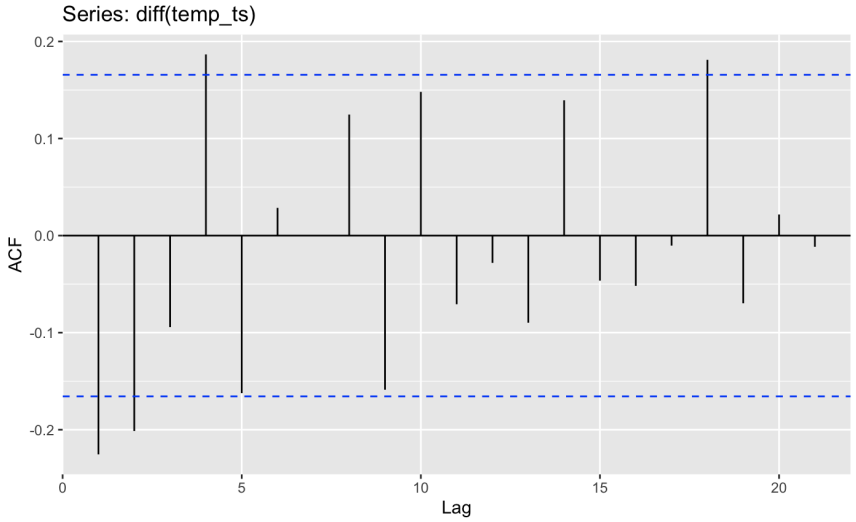
*5.1.4 Best model*

We conclude that the best ARIMA model for GHG is (0, 1, 0) as it has a lower MAPE and AICc than our best manual Arima.

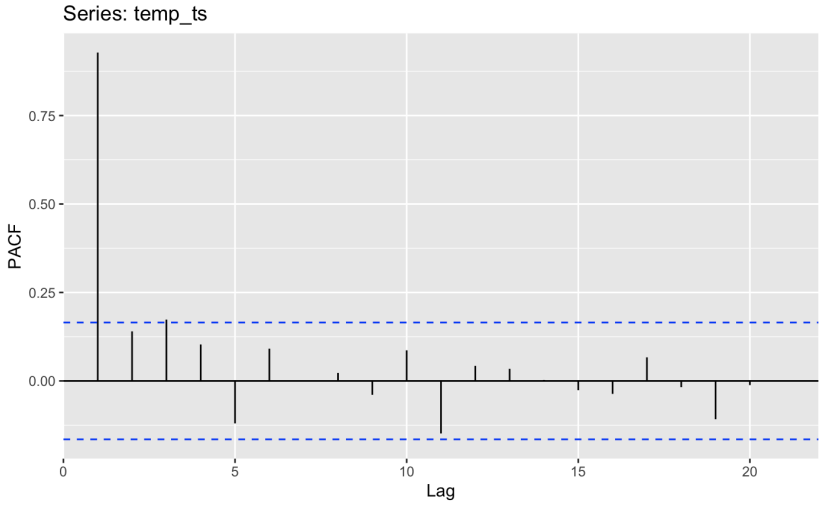
5.2 Temperature Anomalies

*5.2.1 ACF/PACF*

```
ggAcf(diff(temp_ts))
```




```
ggPacf(temp_ts)
```



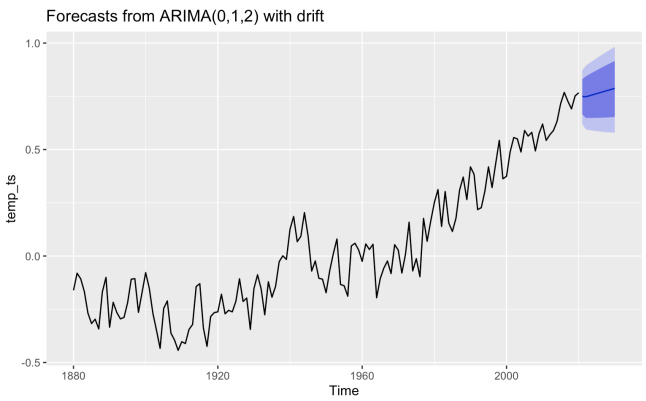

*5.2.2 Auto Arima*

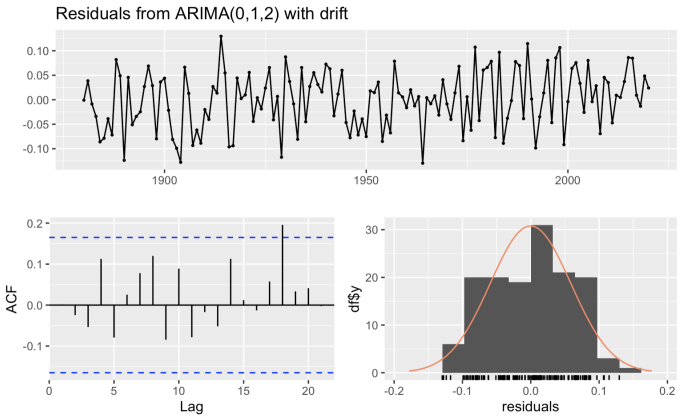
# auto.arima AICc = -385.48 / MAPE: 96.09611

```
model_temp2 = auto.arima(temp_ts, lambda = "auto")
summary(model_temp2)
autoplot(forecast(model_temp2)) # (0, 1, 2)
checkresiduals(model_temp2)
temp_split = ts_split(temp_ts, sample.out = 80)
accuracy(x = temp_split$test[,1:1], forecast(auto.arima(temp_split$train[,1:1])))
```

```
Series: temp_ts
ARIMA(0,1,2) with drift
Box Cox transformation: lambda= 1.253547

Coefficients:
          ma1      ma2   drift
      -0.3710  -0.2426  0.0046
s.e.   0.0787   0.0737  0.0020

sigma^2 = 0.003582:  log likelihood = 196.89
AIC=-385.78   AICc=-385.48   BIC=-374.01

Training set error measures:
                      ME       RMSE        MAE      MPE     MAPE      MASE        ACF1
Training set -0.001869223 0.09302429 0.07689913 83.65737 126.8209 0.9288366 -0.01905851
```

```
        Ljung-Box test

data:  Residuals from ARIMA(0,1,2) with drift
Q* = 8.993, df = 8, p-value = 0.3429

Model df: 2.   Total lags used: 10
```




| | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 | Theil's U |
|---|---|---|---|---|---|---|---|---|
| Training set | 0.006552768 | 0.08995245 | 0.07695409 | 143.31483 | 215.40489 | 0.9565581 | 0.005167274 | NA |
| Test set | 0.011865395 | 0.11716223 | 0.10616997 | 96.09611 | 96.09611 | 1.3197186 | 0.577995268 | 1.299916 |

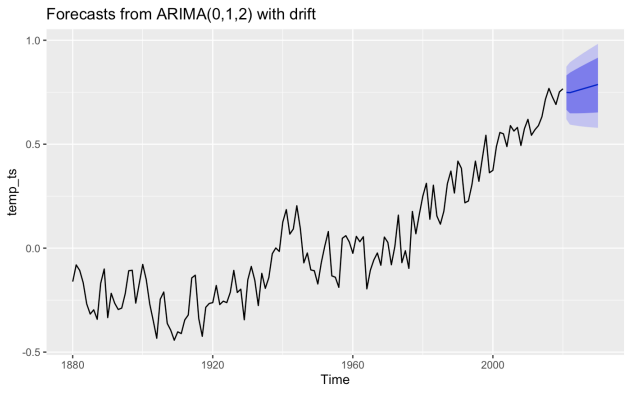
*5.2.3 Manual Arima*

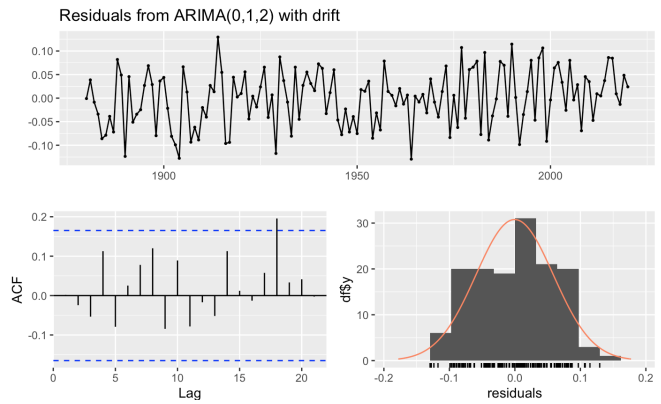
# Manual Arima: TEMP (3, 1, 4) MAPE: 179.9973 / AICc = -257.59

```
model_temp = Arima(temp_ts, order = c(3, 1, 4), include.constant = TRUE)

summary(model_temp)

autoplot(forecast(model_temp))

checkresiduals(model_temp)

temp_split = ts_split(temp_ts, sample.out = 80)

accuracy(x = temp_split$test[,1:1], forecast(Arima(temp_split$train[,1:1], order = c(3, 1, 4), include.constant = TRUE)))
```

```
Series: temp_ts
ARIMA(0,1,2) with drift
Box Cox transformation: lambda= 1.253547

Coefficients:
          ma1      ma2   drift
      -0.3710  -0.2426  0.0046
s.e.   0.0787   0.0737  0.0020

sigma^2 = 0.003582:  log likelihood = 196.89
AIC=-385.78   AICc=-385.48   BIC=-374.01

Training set error measures:
                      ME       RMSE        MAE      MPE     MAPE      MASE        ACF1
Training set -0.001869223 0.09302429 0.07689913 83.65737 126.8209 0.9288366 -0.01905851
```

```
        Ljung-Box test

data:  Residuals from ARIMA(3,1,4) with drift
Q* = 2.5449, df = 3, p-value = 0.4672

Model df: 7.   Total lags used: 10
```



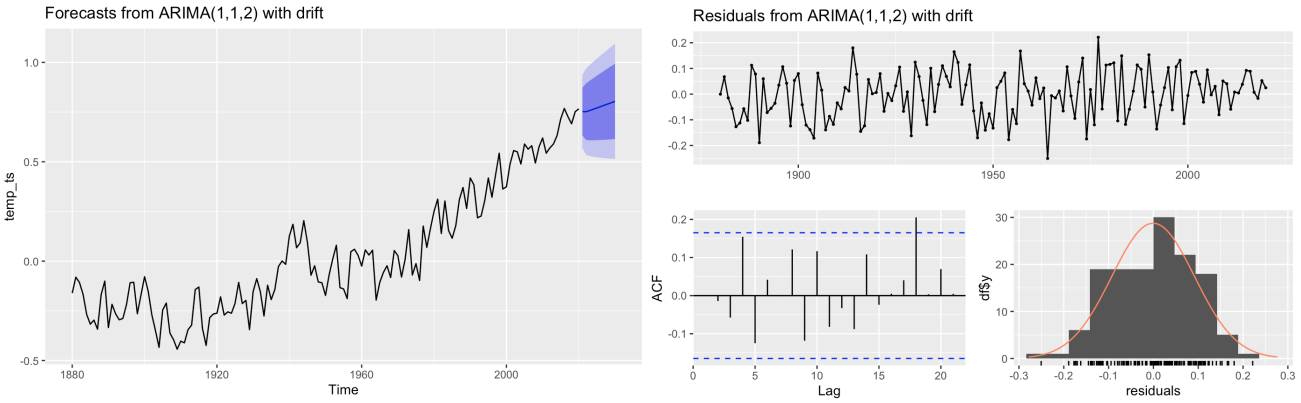

```
                     ME       RMSE        MAE       MPE      MAPE      MASE        ACF1 Theil's U
Training set 0.006552768 0.08995245 0.07695409 143.31483 215.40489 0.9565581 0.005167274        NA
Test set     0.011865395 0.11716223 0.10616997  96.09611  96.09611 1.3197186 0.577995268  1.299916
```

```
model2_temp = Arima(temp_ts, order = c(1, 1, 2), include.constant = TRUE)

summary(model2_temp)

autoplot(forecast(model2_temp))
```

```
checkresiduals(model2_temp)

temp_split = ts_split(temp_ts, sample.out = 80)

accuracy(x = temp_split$test[,1:1], forecast(Arima(temp_split$train[,1:1], order = c(1, 1, 2), include.constant = TRUE)))
```

```
Series: temp_ts
ARIMA(1,1,2) with drift

Coefficients:
         ar1      ma1      ma2   drift
      0.1348  -0.5103  -0.1742  0.0066
s.e.  0.2510   0.2403   0.1362  0.0029

sigma^2 = 0.008782:  log likelihood = 134.62
AIC=-259.24   AICc=-258.8   BIC=-244.54

Training set error measures:
                      ME       RMSE        MAE      MPE     MAPE      MASE          ACF1
Training set -0.000621884 0.09203792 0.07589915 79.50868 118.4592 0.9167582 -9.817821e-05
```

```
        Ljung-Box test

data:  Residuals from ARIMA(1,1,2) with drift
Q* = 14.602, df = 7, p-value = 0.04146

Model df: 3.   Total lags used: 10
```




```
                     ME       RMSE        MAE      MPE     MAPE      MASE        ACF1 Theil's U
Training set -0.0013654685 0.08951252 0.07626554 127.2526 192.5931 0.9479993 -0.004493271        NA
Test set     -0.0009821481 0.12299503 0.11232021 106.8269 106.8269 1.3961676  0.591934368  1.475971
```

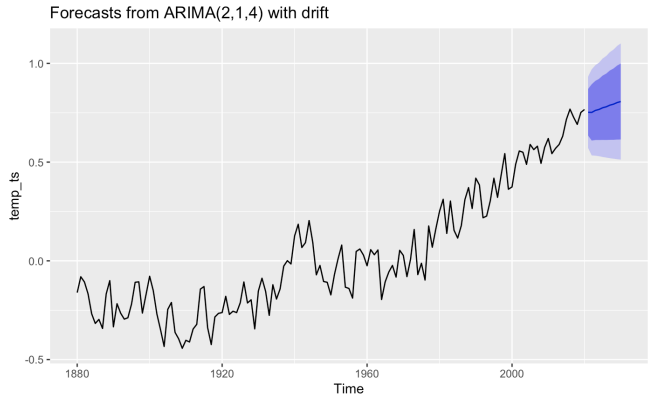
# Manual Arima: TEMP (2, 1, 4) MAPE: 155.6419 / AICc = -237.31

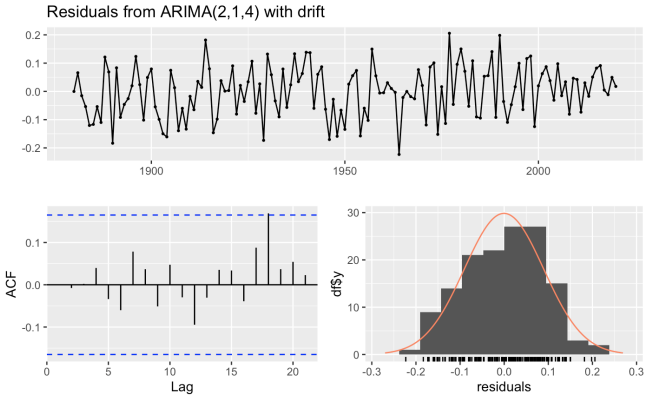
```
model3_temp = Arima(temp_ts, order = c(2, 1, 4), include.constant = TRUE)

summary(model3_temp)

autoplot(forecast(model3_temp))
```

```
checkresiduals(model3_temp)

temp_split = ts_split(temp_ts, sample.out = 80)

accuracy(x = temp_split$test[,1:1], forecast(Arima(temp_split$train[,1:1], order = c(2, 1, 4), include.constant = TRUE)))
```

```
Series: temp_ts
ARIMA(2,1,4) with drift

Coefficients:
          ar1     ar2     ma1      ma2      ma3     ma4   drift
      -0.6375  0.2746  0.3126  -0.7689  -0.1515  0.1425  0.0066
s.e.   1.4052  1.3221  1.3980   0.8784   0.7875  0.4732  0.0030

sigma^2 = 0.008489:  log likelihood = 138.42
AIC=-260.84   AICc=-259.74   BIC=-237.31

Training set error measures:
                      ME       RMSE        MAE      MPE     MAPE      MASE          ACF1
Training set -0.0004673384 0.08948254 0.07416173 50.02817 91.44183 0.8957726 -0.0009954205
```

```
        Ljung-Box test

data:  Residuals from ARIMA(2,1,4) with drift
Q* = 2.4926, df = 4, p-value = 0.646

Model df: 6.   Total lags used: 10
```




```
                    ME      RMSE       MAE       MPE     MAPE     MASE         ACF1 Theil's U
Training set  0.0000499559 0.0886163 0.07560511  74.81137 147.3810 0.939790 -0.001760868        NA
Test set     -0.0175005713 0.1374065 0.12588689 125.41711 125.4171 1.564805  0.587180890  1.821694
```

*5.2.4 Best Model*

We conclude that the best model for temperature anomalies is auto Arima (0, 1, 2) as it has a lower RMSE/MAPE than our best manual Arima.

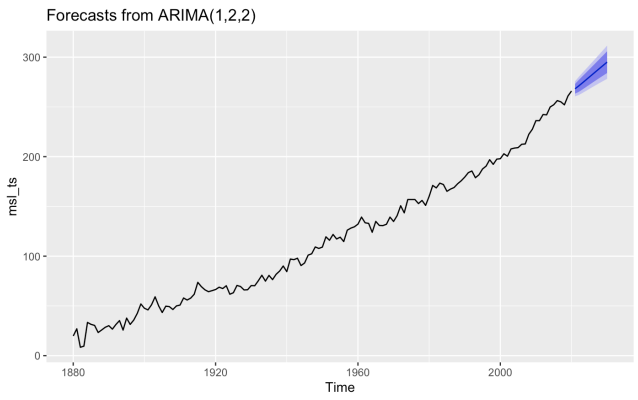
## 5.3 Mean Sea Level

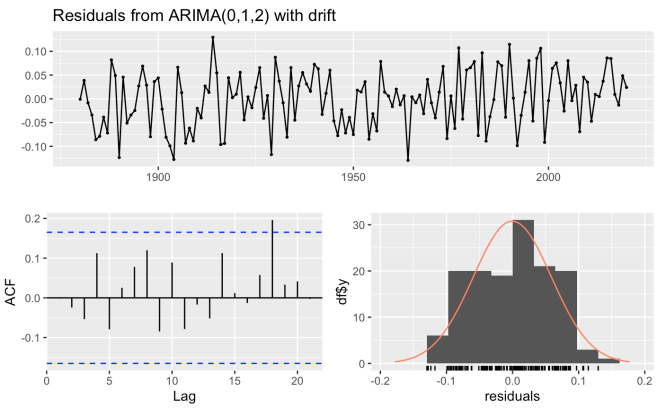
### *5.3.1 Auto Arima*

```
model_msl = auto.arima(msl_ts, lambda = "auto")
summary(model_msl)
autoplot(forecast(model_msl)) # (0,1,2)
checkresiduals(model_temp2)
msl_split = ts_split(msl_ts, sample.out = 80)
accuracy(x = msl_split$test[,1:1], forecast(auto.arima(msl_split$train[,1:1])))
```

```
Series: msl_ts
ARIMA(1,2,2)
Box Cox transformation: lambda= 1.316535

Coefficients:
         ar1      ma1     ma2
      0.3567  -1.6722  0.6989
s.e.  0.1957   0.1550  0.1519

sigma^2 = 498.5:  log likelihood = -629.18
AIC=1266.37   AICc=1266.67   BIC=1278.11

Training set error measures:
                    ME     RMSE      MAE        MPE     MAPE      MASE        ACF1
Training set 0.5148823 5.309408 4.030768 -0.7652999 6.692394 0.8568315 -0.03735401
```

```
        Ljung-Box test

data:  Residuals from ARIMA(0,1,2) with drift
Q* = 8.993, df = 8, p-value = 0.3429

Model df: 2.   Total lags used: 10
```




| | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 | Theil's U |
|---|---|---|---|---|---|---|---|---|
| Training set | -0.05074935 | 5.656688 | 4.242850 | -4.298562 | 13.01068 | 0.8643036 | 0.1679530 | NA |
| Test set | 8.06832080 | 9.097511 | 8.104087 | 7.826382 | 7.86589 | 1.6508693 | 0.4945515 | 1.923896 |

# 6. Multivariate time analysis

## 6.1 ARIMA-X

### *6.1.1 Between two variables*

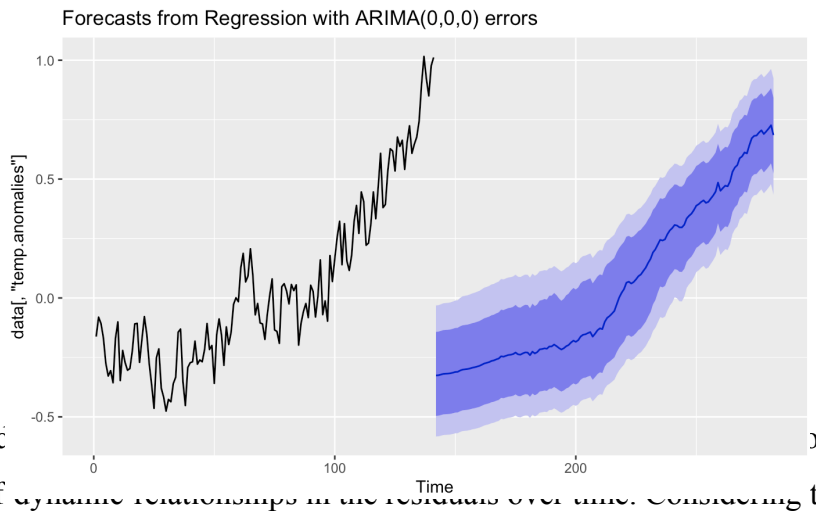
We also did ARIMA-X as we wanted to leverage on the information provided by both the time series data as well as additional exogenous variables to improve forecasting accuracy and to understand the relationship between our target variable and each of the predictors.

a. *Temperature anomalies VS Greenhouse gas emissions*

We first set temp anomalies as our target variable against ghg emissions and used auto Arima to find the model.

```
temp_ghg = auto.arima(data['temp.anomalies' ], xreg=data["ghg.emission"], lambda = "auto")
summary(temp_ghg)
autoplot(forecast(temp_ghg, xreg = data["ghg.emission"]))
```

```
Series: data[, "temp.anomalies"]
Regression with ARIMA(0,0,0) errors
Box Cox transformation: lambda= 1.131028

Coefficients:
      intercept   xreg
        -1.2342  0e+00
s.e.     0.0178  1e-04

sigma^2 = 0.01388:  log likelihood = 102.48
AIC=-198.96   AICc=-198.78   BIC=-190.11

Training set error measures:
                      ME      RMSE       MAE      MPE     MAPE     MASE      ACF1
Training set -0.001715019 0.1464811 0.1161591 151.7282 290.5705 1.267437 0.6909345
```

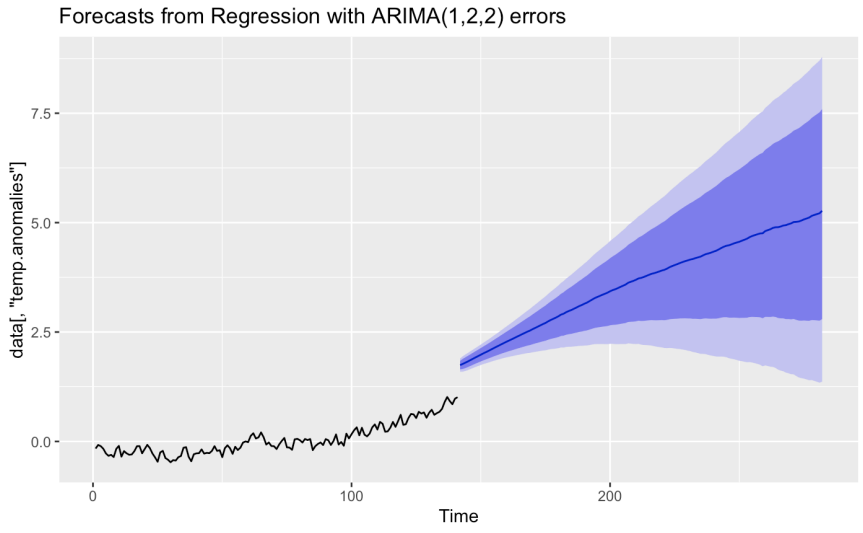



However, we enc optimal as it suggests a lack of dynamic relationships in the residuals over time. Considering the scientific

consensus[1] attributing global warming predominantly to greenhouse gas emissions and their impact on temperature, this finding seemed inadequate. Thus, we proceeded to explore ARIMA models manually with varying orders.

```
tempp_ghg = Arima(data[,'temp.anomalies'], order = c(1, 2, 2), xreg=data[,"ghg.emission"], lambda = "auto")

summary(tempp_ghg)

autoplot(forecast(tempp_ghg, xreg = data[,"ghg.emission"]))
```

```
Series: data[, "temp.anomalies"]
Regression with ARIMA(1,2,2) errors
Box Cox transformation: lambda= 1.131028

Coefficients:
Warning: NaNs produced          ar1      ma1     ma2  xreg
      0.4628  -1.8547  0.8691     0
s.e.  0.0766   0.0083  0.0026   NaN

sigma^2 = 0.006754:  log likelihood = 149.63
AIC=-289.26   AICc=-288.81   BIC=-274.58

Training set error measures:
                     ME      RMSE        MAE      MPE     MAPE      MASE      ACF1
Training set 0.01289628 0.1011968 0.08427159 76.74575 128.5643 0.9195055 0.0308833
```




Eventually, we generated the ARIMA(1,2,2) model, which gave us the lowest AICc value and an improved forecast plot.

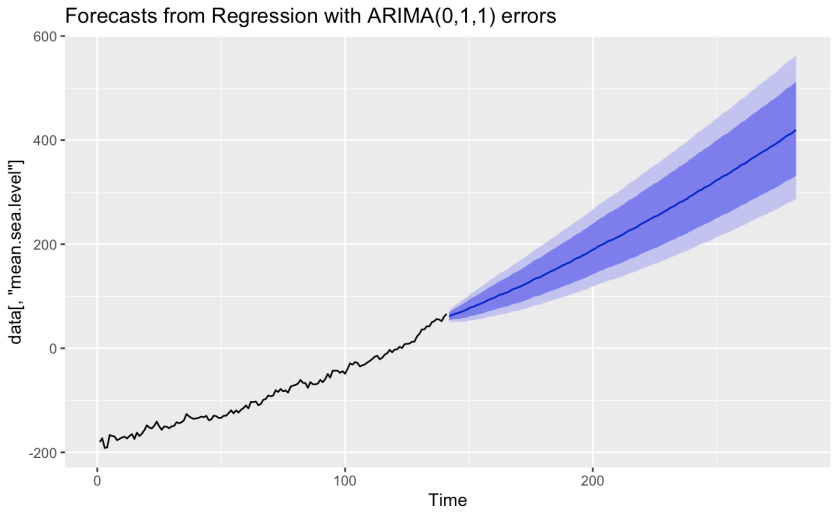
*b. Mean sea level VS Temperature anomalies*

Next, we set mean sea level as our target variable against temp anomalies.

[1] Chen, L., Gao, J., & Vahid, F. (2022). Global temperatures and greenhouse gases: A common features approach. *Journal of Econometrics*, *230*(2), 240–254. https://doi.org/10.1016/j.jeconom.2021.04.003

```
msl_temp = auto.arima(data[,"mean.sea.level"], xreg=data[,"temp.anomalies"], lambda = "auto")

summary(msl_temp)

autoplot(forecast(msl_temp, xreg = data[,"temp.anomalies"]))
```

```
Series: data[, "mean.sea.level"]
Regression with ARIMA(0,1,1) errors
Box Cox transformation: lambda= 0.7605642

Coefficients:
          ma1   drift    xreg
      -0.3113  0.6992  1.6075
s.e.   0.0798  0.1177  1.6278

sigma^2 = 4.098:  log likelihood = -295.92
AIC=599.83   AICc=600.13   BIC=611.6

Training set error measures:
                     ME     RMSE      MAE       MPE     MAPE      MASE         ACF1
Training set -0.3419077 5.478202 4.227579 -2.295289 13.57332 0.8986682 -0.008332118
```




c. *Mean sea level VS Greenhouse gas emissions*

Lastly, we set mean sea level as our target against ghg emissions.

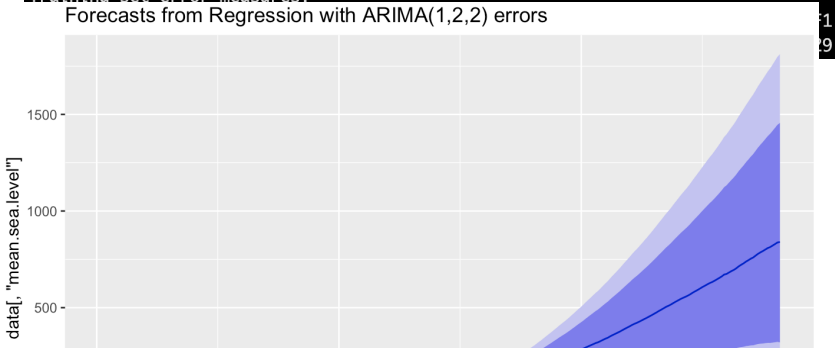
```
msl_ghg = auto.arima(data[,"mean.sea.level"], xreg=data[,"ghg.emission"], lambda = "auto")

summary(msl_ghg)

autoplot(forecast(msl_ghg, xreg = data[,"ghg.emission"]))
```

```
Series: data[, "mean.sea.level"]
Regression with ARIMA(0,0,0) errors
Box Cox transformation: lambda= 0.7605642

Coefficients:
      intercept   xreg
       -76.1545  0e+00
s.e.     0.6316  1e-04

sigma^2 = 17.52:  log likelihood = -400.92
AIC=807.84   AICc=808.01   BIC=816.68
```


However, we ended up with the ARIMA(0,0,0) model again, which may not be the best model. Hence, we went on to try manual Arima with different orders and resulted in ARIMA(1, 2, 2) being the best model, with the lowest AICc value.

```
msll_ghg = Arima(data[,"mean.sea.level"], order = c(1,2,2), xreg=data[,"ghg.emission"], lambda = "auto")

summary(msll_ghg)

autoplot(forecast(msll_ghg, xreg = data[,"ghg.emission"]))
```

```
Series: data[, "mean.sea.level"]
Regression with ARIMA(1,2,2) errors
Box Cox transformation: lambda= 0.7605642

Coefficients:
Warning: NaNs produced          ar1      ma1     ma2  xreg
      0.3391  -1.6974  0.7256     0
s.e.  0.1241   0.0430  0.0448   NaN

sigma^2 = 3.897:  log likelihood = -291.58
AIC=593.17   AICc=593.62   BIC=607.84

Training set error measures:
```



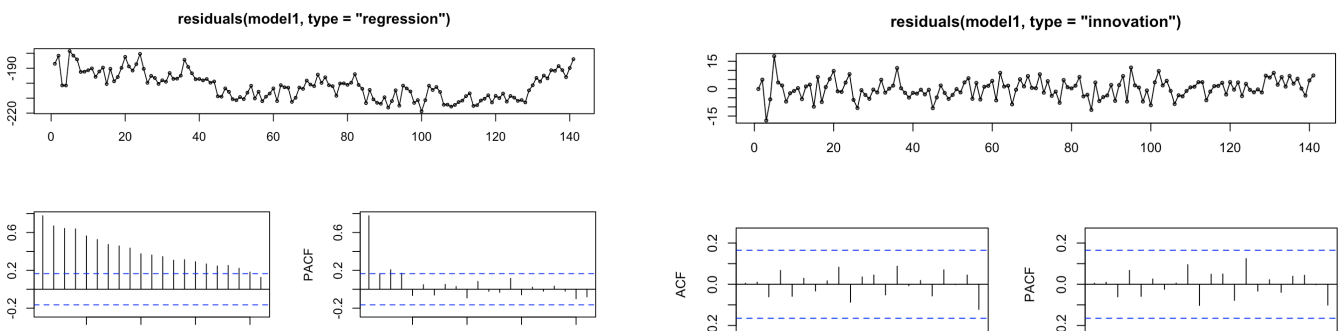
*6.1.2 Among the three variables*

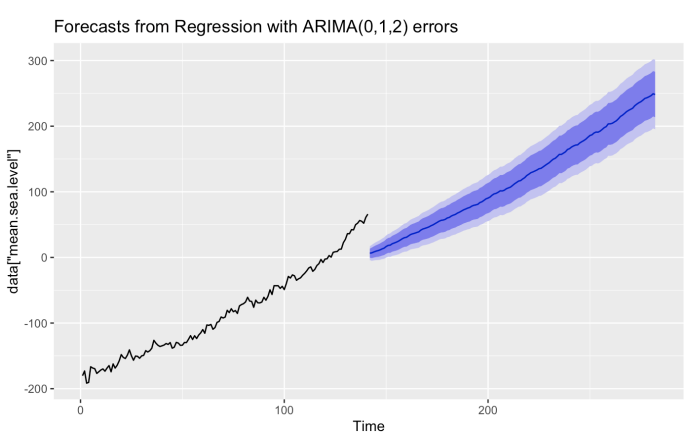
a. *Mean sea level as Y variable*

```
model1 = auto.arima(data['mean.sea.level'], xreg = as.matrix(cbind(data['ghg.emission'], data['temp.anomalies'])))
summary(model1)
tsdisplay(residuals(model1, type="regression"))
tsdisplay(residuals(model1, type="innovation"))
checkresiduals(model1)
autoplot(forecast(model1, xreg = as.matrix(cbind(data['ghg.emission'], data['temp.anomalies'])), h=80))
```

```
Series: data["mean.sea.level"]
Regression with ARIMA(0,1,2) errors

Coefficients:
          ma1      ma2   drift  ghg.emission  temp.anomalies
      -0.4201  -0.1637  1.3287             0          5.0938
s.e.   0.0836   0.0838  0.1010             0          4.3947

sigma^2 = 28.68:  log likelihood = -431.2
AIC=874.4   AICc=875.04   BIC=892.05

Training set error measures:
                     ME     RMSE      MAE       MPE     MAPE      MASE        ACF1
Training set -0.02364897 5.239836 4.072119 -5.896119 15.85623 0.8656217 0.005115978
```

```
        Ljung-Box test

data:  Residuals from Regression with ARIMA(0,1,2) errors
Q* = 4.2628, df = 8, p-value = 0.8327

Model df: 2.   Total lags used: 10
```





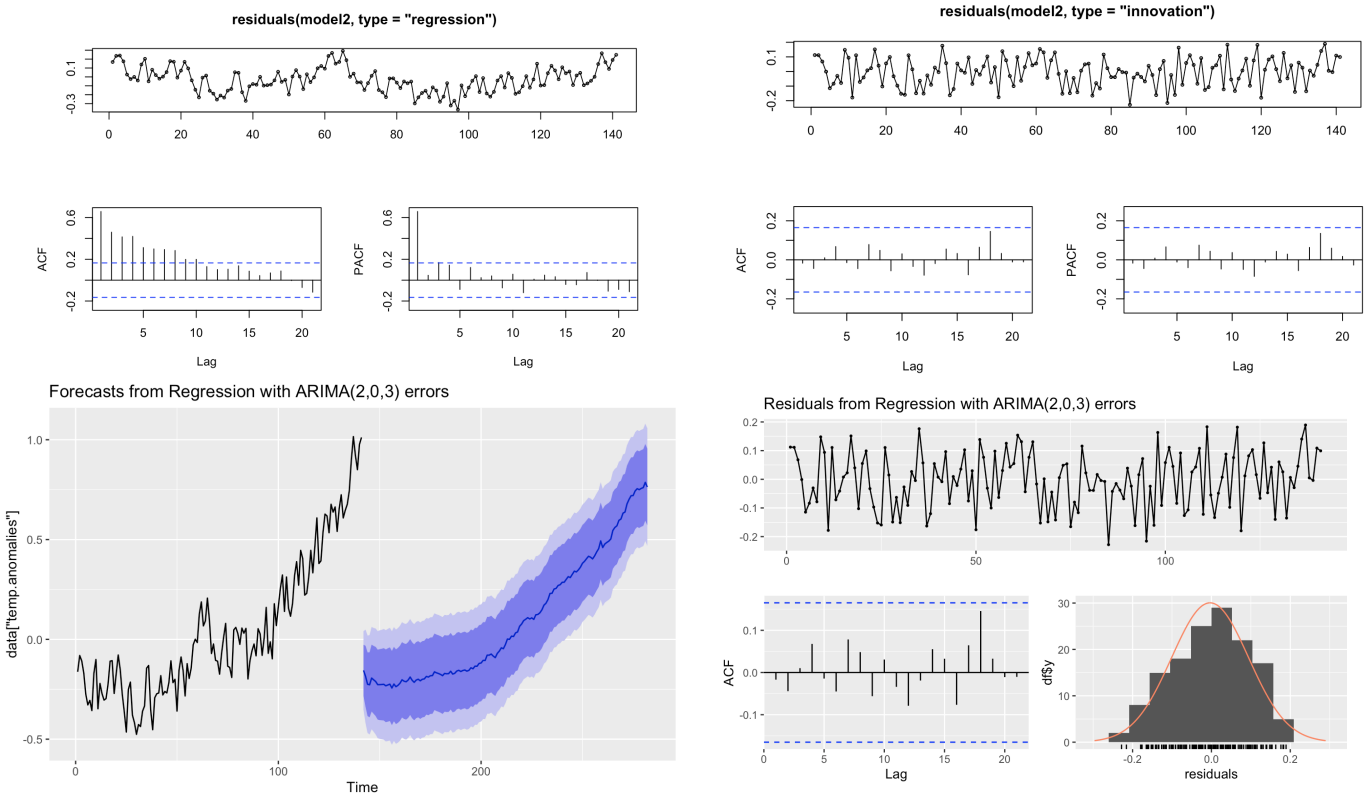


*b. Temperature anomalies as Y variable*

```
model2 = auto.arima(data['temp.anomalies'], xreg = as.matrix(cbind(data['ghg.emission'], data['mean.sea.level'])))
summary(model2)
tsdisplay(residuals(model2, type="regression")))
tsdisplay(residuals(model2, type="innovation"))
checkresiduals(model2)
autoplot(forecast(model2,  xreg = as.matrix(cbind(data['ghg.emission'], data['mean.sea.level'])), h=80))
```

```
Series: data["temp.anomalies"]
Regression with ARIMA(2,0,3) errors

Coefficients:
         ar1     ar2     ma1      ma2      ma3  intercept  ghg.emission  mean.sea.level
      0.0058  0.8720  0.6365  -0.5240  -0.3082    -0.0994             0          0.0017
s.e.  0.0732  0.0672  0.1047   0.0992   0.0839     0.2462             0          0.0014

sigma^2 = 0.01005:  log likelihood = 127.77
AIC=-237.55   AICc=-236.17   BIC=-211.01

Training set error measures:
                      ME       RMSE       MAE      MPE     MAPE      MASE         ACF1
Training set -0.003360672 0.09738181 0.0804904 47.35752 95.14296 0.8782482 -0.01695221
```

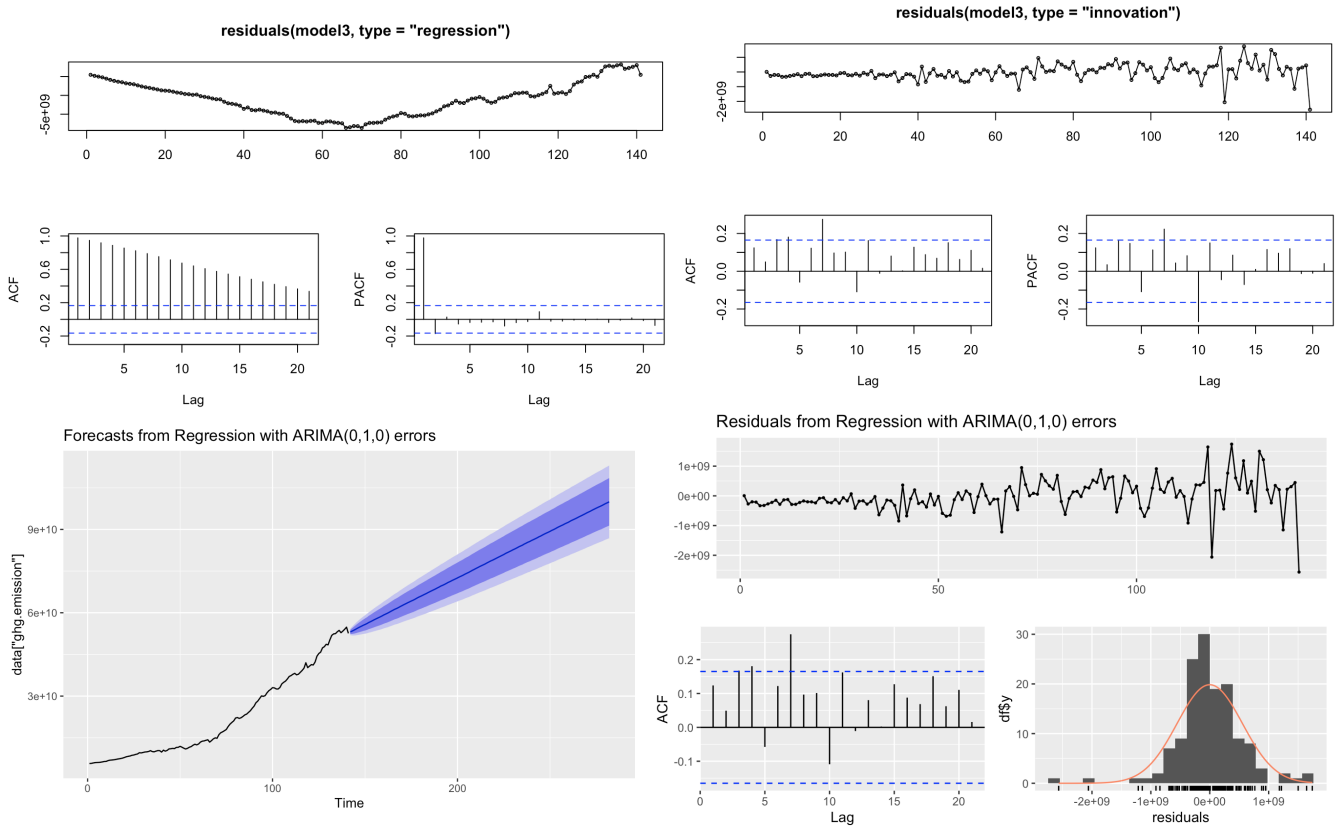
```
        Ljung-Box test

data:  Residuals from Regression with ARIMA(2,0,3) errors
Q* = 3.2539, df = 5, p-value = 0.6609

Model df: 5.   Total lags used: 10
```




*c. Greenhouse gas emissions as Y variable*

```
model3 = auto.arima(data['ghg.emission'], xreg = as.matrix(cbind(data['temp.anomalies'], data['mean.sea.level'])))
summary(model3)
tsdisplay(residuals(model3, type="regression"))
tsdisplay(residuals(model3, type="innovation"))
checkresiduals(model3)
autoplot(forecast(model3, xreg = as.matrix(cbind(data['temp.anomalies'], data['mean.sea.level'])), h = 80))
```

```
Series: data["ghg.emission"]
Regression with ARIMA(0,1,0) errors

Coefficients:
         drift  temp.anomalies  mean.sea.level
     335764015      -259772359        562071.7
s.e.  55027437       188254706       8497514.4

sigma^2 = 3.148e+17:  log likelihood = -3017.48
AIC=6042.95   AICc=6043.25   BIC=6054.72

Training set error measures:
                    ME       RMSE        MAE        MPE     MAPE      MASE      ACF1
Tra          Ljung-Box test                                        .1238872

data:  Residuals from Regression with ARIMA(0,1,0) errors
Q* = 30.38, df = 10, p-value = 0.0007423

Model df: 0.   Total lags used: 10
```




## 6.2 Vector Autoregression (VAR)

We used VAR analysis because we want to conduct a simultaneous time series forecast of 2 interdependent variables. We first assessed the feasibility of estimating a VAR model by considering factors such as the number of observations we have, endogenous variables, and the proposed order of the VAR model. Only when we have confirmed that we have at least 10 times as many observations as coefficients that we are estimating, we decided to carry on with VAR.

### *6.2.1 Chosen variables*

Due to model complexity, we chose to only study the relationship between temperature anomalies and changes in sea level, to see if indeed one variable can be used to predict the other.

```
#TEMP vs MSL
tempmsl <- data[, c("mean.sea.level", "temp.anomalies")]
tempmsl_ts = ts(data = tempmsl)
```

*6.2.2 Determine the appropriate lag length for VAR model*

```
VARselect(tempmsl)
```

```
$selection
AIC(n)  HQ(n)  SC(n) FPE(n)
     4      2      1      4

$criteria
                1          2          3          4          5          6          7          8          9
AIC(n) -1.2194397 -1.3016037 -1.3123297 -1.3293649 -1.2882188 -1.2705232 -1.2262336 -1.1822807 -1.1627808
HQ(n)  -1.1659288 -1.2124189 -1.1874709 -1.1688322 -1.0920122 -1.0386427 -0.9586791 -0.8790523 -0.8238785
SC(n)  -1.0877513 -1.0821230 -1.0050567 -0.9342996 -0.8053613 -0.6998734 -0.5677914 -0.4360463 -0.3287541
FPE(n)  0.2954004  0.2721153  0.2692471  0.2647601  0.2759802  0.2810529  0.2939881  0.3074825  0.3139063
               10
AIC(n) -1.1335708
HQ(n)  -0.7589945
SC(n)  -0.2117518
FPE(n)  0.3236826
```

Since SC shows 1, we will start with VAR(1), and iterate on the selection of VAR(1) until there is no longer any residual serial correlation.

```
var_model1 <- VAR(tempmsl_ts, p = 1, type = "const")
serial.test(var_model1, lags.pt = 10, type = "PT.asymptotic")
```

```
        Portmanteau Test (asymptotic)

data:  Residuals of VAR object var_model1
Chi-squared = 49.823, df = 36, p-value = 0.06251
```

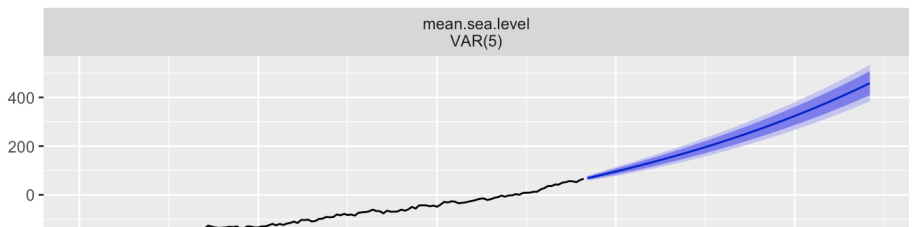
We now gradually increase the order of the model until there is no longer any time series correlation.

```
varmodel5 <- VAR(tempmsl_ts, p = 5, type = "const")
serial.test(varmodel5, lags.pt = 10, type = "PT.asymptotic") # Choose this since it has highest p-value, residuals do not contain serial correlation
```

```
        Portmanteau Test (asymptotic)

data:  Residuals of VAR object varmodel5
Chi-squared = 17.505, df = 20, p-value = 0.62
```

Hence, after trying orders 1-6, we realised that order 5 gives us the highest p-value of 0.62 which means that at the 5-10% level of significance, we fail to reject the null hypothesis that there is no correlation in the residuals. Hence, we plot our forecast model on order 5.

```
autoplot(forecast(varmodel5, h = 80))
AIC(varmodel5) # 594.7769
```



## 6.3 Vector Error Correction Model (VECM)

VECM, being a special case of VAR, includes all the terms that are in the VAR and more. It is able to exploit all the information that the VAR exploits, as well as additional relationships on top of that. Hence, we decided to model VECM.

### *6.3.1 Determining number of cointegrating relationships*

Given that the Johansen procedure requires variables to be integrated at the same order (i.e., cointegrated), we are able to run the analysis only on temperature anomalies and mean sea level as both require differencing of order 1, as highlighted in our results in Section 4.1.

```
summary(ca.jo(tempmsl))
```

```
Test type: maximal eigenvalue statistic (lambda max) , with linear trend

Eigenvalues (lambda):
[1] 0.1222926 0.0258292

Values of teststatistic and critical values of test:

          test 10pct  5pct  1pct
r <= 1 |  3.64  6.50  8.18 11.65
r = 0  | 18.13 12.91 14.90 19.19

Eigenvectors, normalised to first column:
(These are the cointegration relations)

                  mean.sea.level.l2 temp.anomalies.l2
mean.sea.level.l2            1.0000            1.0000
temp.anomalies.l2         -199.2354          142.5095

Weights W:
(This is the loading matrix)

                 mean.sea.level.l2 temp.anomalies.l2
mean.sea.level.d       -0.02337127      0.0074556040
temp.anomalies.d        0.00130072      0.0000577494
```

Based on the results, 3.64 is smaller than all of the critical values cut off, we can conclude that there is 1 cointegrating relationship. Since 0 < r <= N-1, we can proceed to build the VECM model.

*6.3.2 Determining the number of lags*

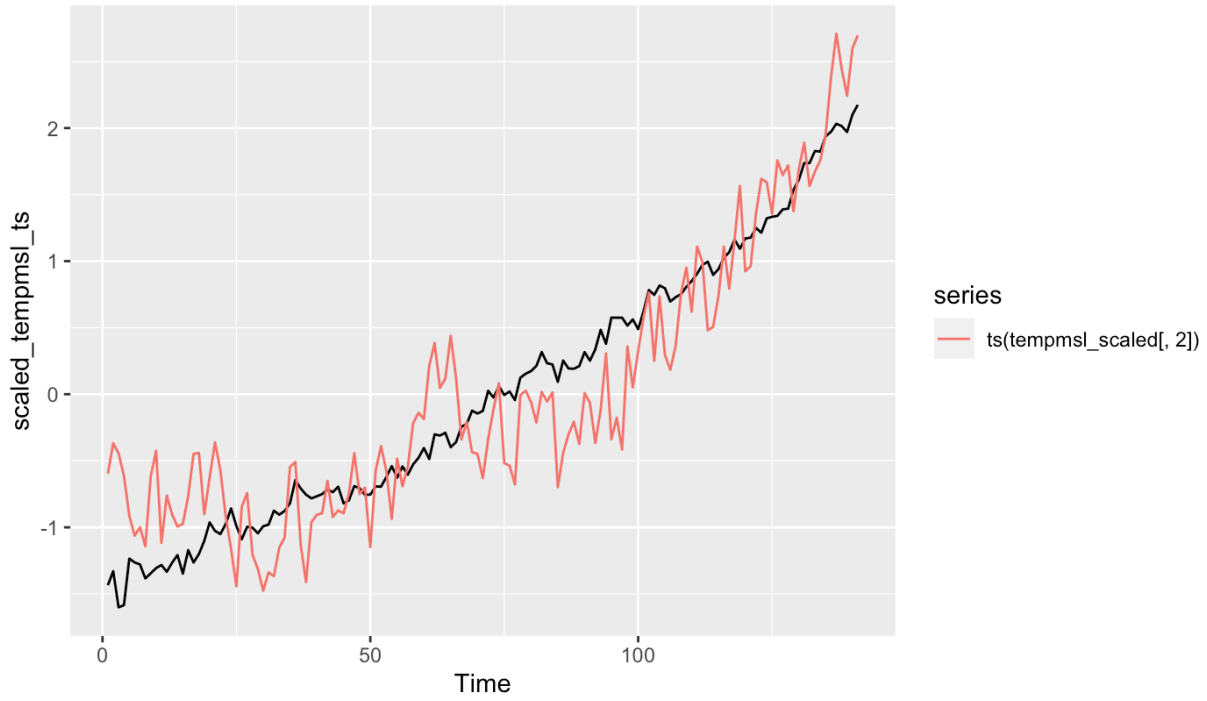
```
VARselect(tempmsl)
```

```
$selection
AIC(n)  HQ(n)  SC(n) FPE(n)
     4      2      1      4

$criteria
               1          2          3          4          5          6          7          8          9
AIC(n) -1.2194397 -1.3016037 -1.3123297 -1.3293649 -1.2882188 -1.2705232 -1.2262336 -1.1822807 -1.1627808
HQ(n)  -1.1659288 -1.2124189 -1.1874709 -1.1688322 -1.0920122 -1.0386427 -0.9586791 -0.8790523 -0.8238785
SC(n)  -1.0877513 -1.0821230 -1.0050567 -0.9342996 -0.8053613 -0.6998734 -0.5677914 -0.4360463 -0.3287541
FPE(n)  0.2954004  0.2721153  0.2692471  0.2647601  0.2759802  0.2810529  0.2939881  0.3074825  0.3139063
              10
AIC(n) -1.1335708
HQ(n)  -0.7589945
SC(n)  -0.2117518
FPE(n)  0.3236826
```

*6.3.3 Building VECM model*

Using the results from the previous 2 functions, we build a VECM model with lag = 1 and 1 cointegrating relationship.

```
tempmsl_scaled <- scale(tempmsl)
scaled_tempmsl_ts <- ts(tempmsl_scaled[, 1], start = 1, frequency = 1)
# Plot scaled temperature anomalies and sea level together
autoplot(scaled_tempmsl_ts) + autolayer(ts(tempmsl_scaled[, 2]))
```




```
vecmmodel = VECM(tempmsl, lag=1, r =1, estim = "ML")
vecmmodel
AIC(vecmmodel) #-147.5885
```

```
                              ECT  Intercept mean.sea.level -1 temp.anomalies -1
Equation mean.sea.level -0.02337127 0.05809115      -0.284712293        1.57031527
Equation temp.anomalies  0.00130072 0.12966909      -0.001091056       -0.08026665
```

*6.3.4 Ljung-Box Test*

```
residuals <- resid(vecmodel)
Box.test(residuals[,1], lag = 1, type = "Ljung") # mean sea level
Box.test(residuals[,2], lag = 1, type = "Ljung") # temp anomalies
```

```
        Box-Ljung test

data:  residuals[, 1]
X-squared = 0.16196, df = 1, p-value = 0.6874


        Box-Ljung test

data:  residuals[, 2]
X-squared = 0.084895, df = 1, p-value = 0.7708
```

Since the p-value of the VECM model for the mean sea level time series and temperature anomalies time series is 0.6874 and 0.7708 respectively, both of which is more than our chosen significance level (0.05), we fail to reject the null hypothesis. This indicates that there is insufficient evidence to conclude that there is autocorrelation. Hence, we have 'passed' the Ljung-Box test as it implies that the residuals are adequately independent or uncorrelated.

*6.3.5 Predicting future values*

Next, we want to predict the future values of the variables within the *tempmsl* dataset for the next 80 time periods.

```
predict(vecmmodel, n.ahead = 80)
```

| | mean.sea.level | temp.anomalies |
|---|---|---|
| 142 | 67.85203 | 0.9567247 |
| 143 | 70.16557 | 0.9291047 |
| 144 | 72.20801 | 0.9089553 |
| 145 | 74.19780 | 0.8963804 |
| 146 | 76.10942 | 0.8891019 |

# 7. Conclusion

7.1 Finding the best model between our ARIMA, VAR and VECM models

We used the AICc selection criteria to find the best model out of all these 3 models. Based on the results shown earlier from our ARIMA-X model (section 6.1.2), we see that when we set temperature anomalies as our Y variable, with ARIMA (2, 0, 3), our AICc gives **-236.17**. Looking at our best VAR model (section 6.2.2), the AICc gives 594.7769. The AICc for our VECM model (section 6.3.3) gives -147.5885. Hence, we conclude that our **ARIMA-X is the best model** as it has the lowest AICc value.

7.2 What can we say based on our forecasts?

Based on our forecasts and predictions, we observe a continued acceleration in all three of our variables. This trend aligns with the understanding that human activities are driving global warming and consequent changes in climate patterns. If our forecasts hold true, this implies a sustained exponential increase in both mean sea level and temperature in the future, bringing about significant adverse effects on our society. Moreover, escalating temperatures

will induce the melting of land-based ice and the expansion of water bodies, further contributing to sea level rise. This interconnected cycle of feedback can exacerbate the detrimental consequences of these shifts. Hence, our findings underscore the pressing need for decisive actions to mitigate climate change and minimize its adverse repercussions.

### 7.3 Future steps

This can be achieved by reducing emissions of greenhouse gases, thereby slowing down the rate of global warming and sea level rise. Governments must also address the escalating temperatures, which are partly due to deforestation and the overexploitation of natural resources. The reduction of forested areas leads to more exposed ground that reflects heat back into the atmosphere, potentially causing a further rise in temperatures. While predicting temperature anomalies resulting from natural disasters is challenging, it is essential for governments to intensify their efforts to lower temperatures. Implementing measures such as promoting the use of renewable energy, supporting reforestation initiatives, and enforcing sustainable planning practices are equally crucial.