

Statistical Methods for Discrete Response, Time Series, and Panel Data (W271): Lab 2

Investigating the Keeling Curve and forecasting CO2 levels in Earth's atmosphere

Denny Lehman, Mingxi Liu, Aruna Bisht, Deepika Maddali

Abstract

The Keeling curve is a [INTRODUCE THE CONCEPT IN ONE SENTENCE HERE and SHOW IMPORTANCE]. In this paper, the Keeling curve was analyzed from two perspectives, one from a researcher in 1998 and one from today (2023). From the perspective of 1998, EDA was performed from 1959 to 1997 and a linear time trend model was fit. After analysis of the assumptions, a cubic polynomial model was selected. Using the Box-Jenkins method, a SARIMA model was constructed and forecast into 2100. To contrast those models, we present the same analysis from the perspective of 2023. A modern data pipeline was constructed for CO2 data and the linear model and SARIMA model were compared to the actual CO2 levels. Both models under predicted CO2 with [ADD SOMETHING ON RMSE or OTHER HERE]. Finally, new models were fit on the 2023 a forecasted into the future with the goal of reviewing model predictions in a future analysis.

Contents

1	Introduction	2
2	Report from the Point of View of 1997	2
2.1	Data	2
2.2	Linear model	3
2.3	ARIMA times series model	5
2.4	Atmospheric CO2 growth Forecast	7
3	Report from the Point of View of the Present	8
3.1	0b.Introduction	8
3.2	1b.Data	8
3.3	MOVE CODE TO THE APPENDIX	8
3.4	2b.Compare linear model forecasts against realized CO2	13
3.5	I'd recommend starting the forecast in 1998	13
3.6	3b.Compare ARIMA models forecasts against realized CO2	14
3.7	4b.Evaluate the performance of 1997 linear and ARIMA models	14
3.8	5b. Train best models on present data	16

1 Introduction

We all know the debate about global warming and its connection to human activities. But to study this topic in a scientific way, we need reliable data. The Keeling Curve is a milestone in this aspect. It shows the ongoing increase in atmospheric carbon dioxide (CO₂) concentrations over time. It is named after Charles David Keeling, the scientist who initiated and maintained the measurements. Keeling began monitoring atmospheric CO₂ levels in 1958 at the Mauna Loa Observatory in Hawaii. He chose this location because it is remote and far from major sources of pollution, providing an ideal site to measure baseline CO₂ concentrations. The Keeling Curve graphically represents the seasonal variations in atmospheric CO₂ concentrations, as well as the long-term increasing trend. Keeling believes the seasonal pattern is a result of the Earth's vegetation absorbing CO₂ during the growing season and releasing it during the dormant period, while the trend is primarily driven by human activities, particularly the burning of fossil fuels such as coal, oil, and natural gas, which release large amounts of CO₂ into the atmosphere. The Keeling Curve is an important tool for scientists, policymakers, and the general public to understand the impact of human activities on the Earth's climate. It serves as a stark reminder of the need to reduce greenhouse gas emissions and address the causes and consequences of climate change.

Our research is based on the data from the Keeling Curve above. We first build a model based on data from 1959 to 1997 and make long-term predictions to the present. Then we combine the actual data with our prediction and discuss the implication of this comparison.

2 Report from the Point of View of 1997

2.1 Data

The data measures the monthly average atmospheric CO₂ concentration from 1959 to 1997, expressed in parts per million (ppm). It was initially collected by an infrared gas analyzer installed at Mauna Loa in Hawaii, which was one of the four analyzers installed by Keeling to evaluate whether there was a persistent increase in CO₂ concentration.

Fig.1 shows a clear long-term upward trend, which is confirmed by Fig.2 where the growth rate for each year is above zero. Fig.2 also suggests the average growth rate after 1970 is higher than that before 1970, although there's no evidence of accelerating growth. The ACF plots in Fig.3 and Fig.4 suggest the original data is non-stationary but its first difference is stationary. More formally, the KPSS tests below confirm the observations above.

Table 1: KPSS test of original and 1st difference

	kpss_stat	kpss_pvalue
original	7.8173	0.01
1st_difference	0.0124	0.10

Another feature of the data is its robust seasonal pattern, with the peak in May and the bottom in October almost every year (see Fig.5). This seasonality can also be seen in Fig.4. Keeling believes it was the result of plant photosynthesis absorbing CO₂ from the atmosphere.

Fig.4 is the histogram of the remaining or irregular components after removing the trend and the

seasonal components from the data with STL¹. It looks like a normal distribution without obvious outliers.

Fig.1 Atmospheric CO2 concentration monthly average, parts per million (ppm)

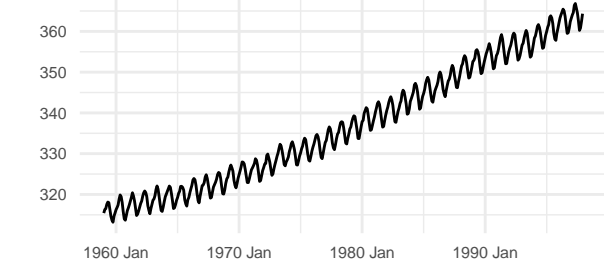


Fig.2 Annual growth rate of concentration, %

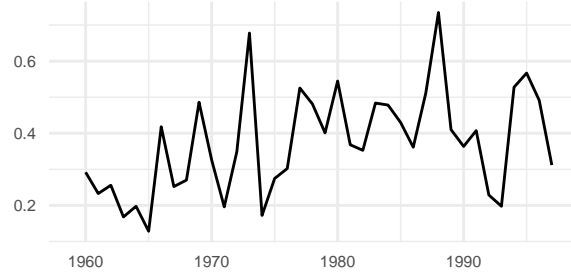


Fig.3 ACF of CO2 concentration

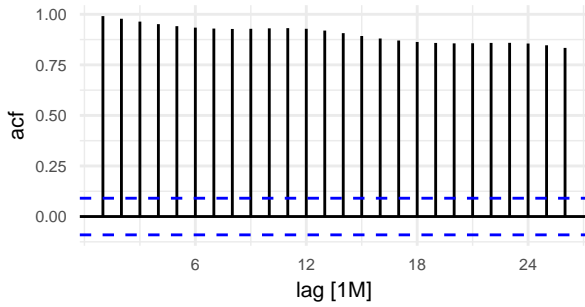


Fig.4 ACF of differenced CO2 concentration

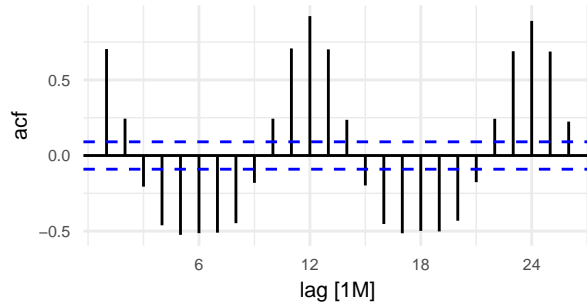


Fig.5 Seasonal plot of CO2 concentration

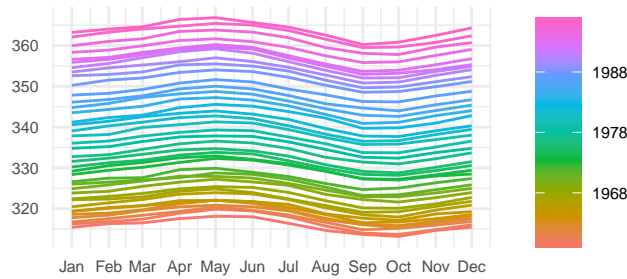
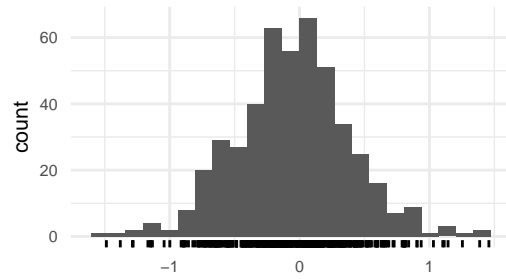


Fig.6 Histogram of irregular component by STL



2.2 Linear model

Before building the model, we need to consider whether the data need a log transformation. Normally, a log transformation is required when the data shows exponential growth or the variance expands or shrinks over time. From Fig.1 and Fig.2 we can see the slope or the growth rate of the data is stable, which suggests the growth is more close to linear instead of exponential. Also, Fig.5 shows the difference between the annual high and the annual low almost remained the same over the years, suggesting the variance is nearly constant. Therefore, the log transformation is not necessary. We can first fit the original data with a linear time trend model as:

¹Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. J. (1990). STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1), 3–33.

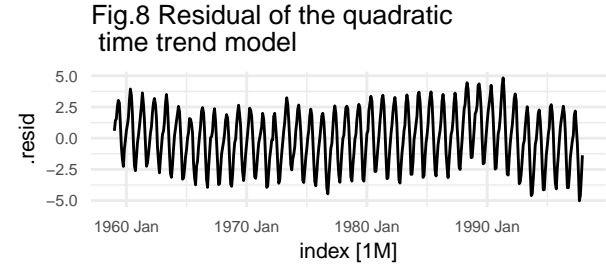
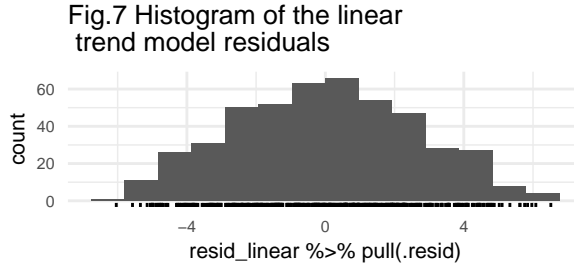
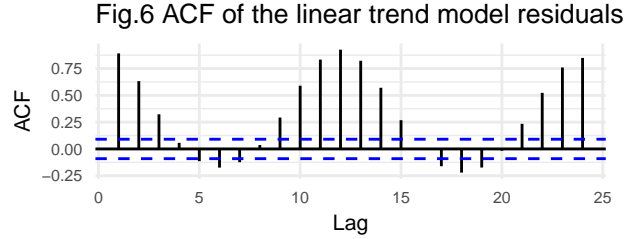
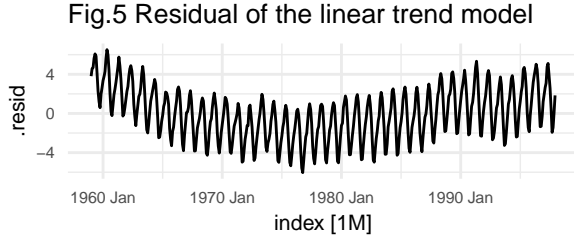
$$\text{CO}_2 = \beta_0 + \beta_1 t + \epsilon_t, \quad (1)$$

which gives the parameters as:

$$\text{CO}_2 = 311.5 + 0.11t + \epsilon_t \quad (2)$$

This linear trend model implies that the CO_2 concentration increased by 0.11 ppm/month on average from 1959 to 1997. However, the residual plots in Fig.5 to Fig.7 suggest this simple linear trend model is not adequate in the following two aspects.

First, the mean of the residual forms a “U” shape over time, suggesting a quadratic or higher-order polynomial time trend model may be more appropriate. For instance, the residual from a quadratic time trend model shows a more constant mean over time, as shown in Fig.8.



In addition, the ACF plot in Fig.6 indicates strong seasonal patterns exist in the residuals, suggesting we should consider seasonal factors in the model. One solution is to include 11 dummy variables in the model to indicate the 12 months.

Based on the two points above, we compare the 2 candidates: a quadratic time trend model and a cubic one, as below.

$$\text{Quadratic time trend: } \text{CO}_2 = \alpha + \beta_0 t + \beta_1 t^2 + \sum_{i=1}^{11} \gamma_i \text{Month}_{it} + \epsilon_t \quad (3)$$

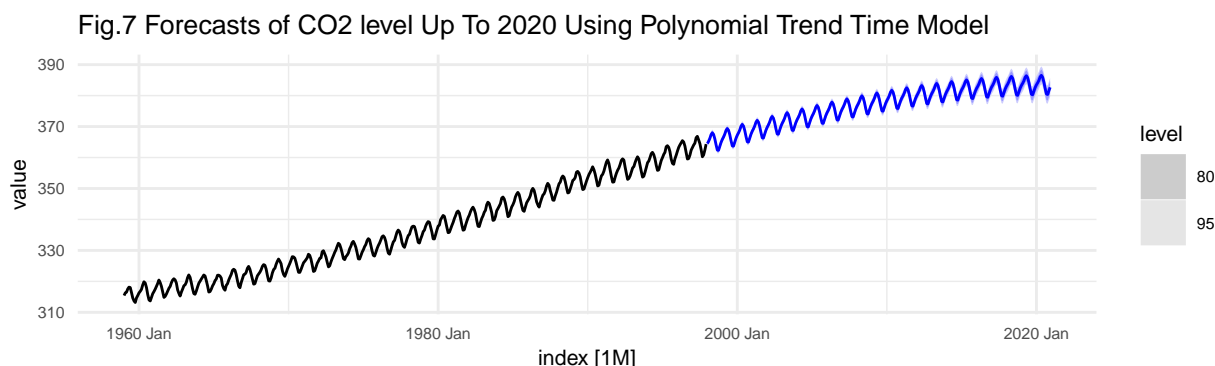
$$\text{Cubic time trend: } \text{CO}_2 = \alpha + \beta_0 t + \beta_1 t^2 + \beta_2 t^3 + \sum_{i=1}^{11} \gamma_i \text{Month}_{it} + \epsilon_t \quad (4)$$

We use the data before 1991 as the training set and the rest as the validation set (similar to an 80-20 split). Our final choice of the model depends on the combination of 2 guidelines: 1) the information criterion (AICc, BIC) from the model fitting process and 2) the root mean square error (RMSE) of predictions on the validation set, which are listed in Table.1. Both information criterion (AICc,

BIC) and RMSE favor the cubic model. Therefore, the cubic time trend model becomes our final choice. Its details are in the Appendix. We plot the forecast of this model until 2020 in Fig.7. One thing to note is that because the coefficient of the cubic term is negative, the predicted values will eventually begin to decrease when predicting the far future. In fact, we can see from Fig.7 that the predicted values have almost topped. This may be inappropriate extrapolation behavior. In that case, we should confine our predicting interval to the near term.

Table 2: Information Criterion of model fitting and RMSE of validation

.model	AIC	AICc	BIC	RMSE
cubic	-659.6147	-658.1324	-596.4044	2.112194
quadratic	-639.1525	-637.8481	-579.8928	2.796572



2.3 ARIMA times series model

We will use the Box Jenkins process to find the best ARIMA model via the following steps:

- Determine the appropriate model from EDA
- Find the best parameters
- Examine the residuals using diagnostic plots and statistical tests

The EDA revealed that the time series of CO2 had both autoregressive and seasonal components. Considering the ACF plot's low slow decay of autocorrelation, we expect differencing to be a key part of any time series model. In addition, we predict that the model will require seasonal components to model the 12 month cycle of seasonal variations. Therefore, we expect a seasonal arima model (SARIMA) with differencing and seasonality terms to be best.

In this section, we fit the best SARIMA model and analyze the results. We choose BIC as our information criteria for model selection. Simplicity is a desirable property in data science models to help explain the relationship between variables. We choose BIC as our information criteria because it penalizes complex models more than AIC or AICc and therefore selects more simple models with fewer parameters as the best ones. Lower BIC scores are better.

```
## Series: value
## Model: ARIMA(0,1,1)(1,1,2)[12]
##
```

```
## Coefficients:
##          ma1      sar1      sma1      sma2
##      -0.3482 -0.4986 -0.3155 -0.4641
## s.e.   0.0499  0.5282  0.5165  0.4367
##
## sigma^2 estimated as 0.08603:  log likelihood=-85.59
## AIC=181.18   AICc=181.32   BIC=201.78
```

After searching over seasonal and non-seasonal P, D, and Q variables, the best model was an ARIMA(0,1,1)(1,1,2)[12] model with BIC score of 201.78. Next, we evaluate the model via diagnostic plots and statistical tests, concluding the Box Jenkins process.

Fig.8 Residuals of the SARIMA model

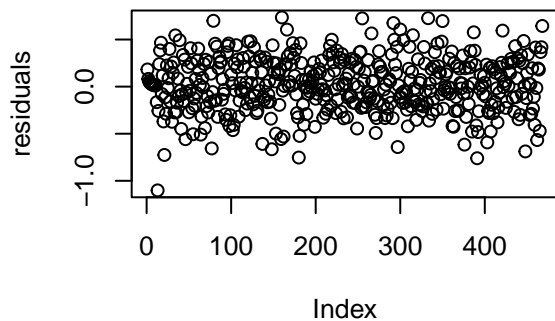


Fig.9 ACF plot of residuals

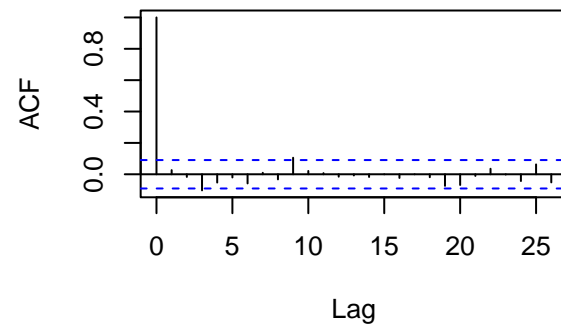


Fig.10 PACF plot of residuals

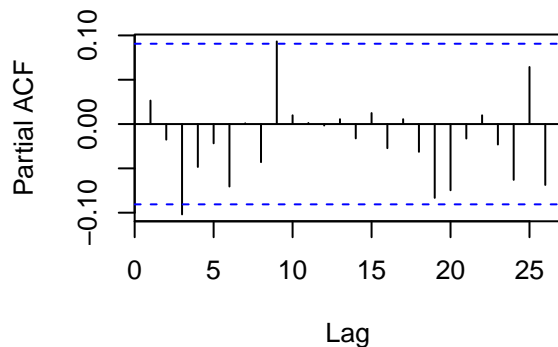
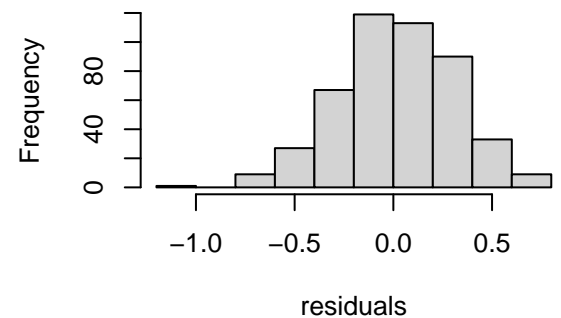


Fig.11 histogram of residuals



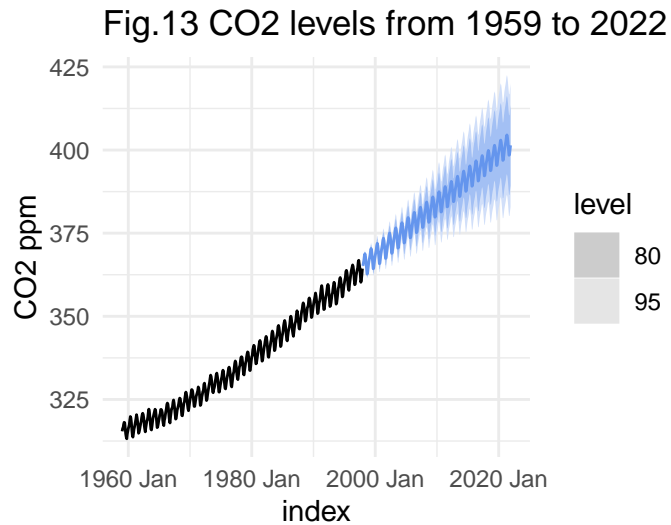
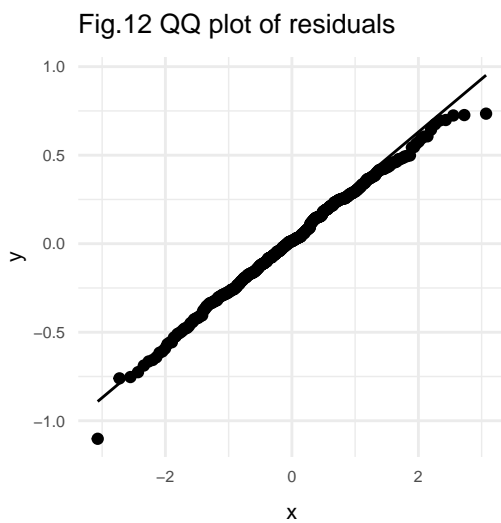
The residual plots (Fig 8-11) show that the SARIMA model was effective, with the residuals looking like stationary white noise (Fig 8). The time series has a mean of 0 with about constant variance, the ACF plot (Fig 9) shows no autocorrelation beyond the initial lag value. The PACF plot (Fig 10) appears to have a significant peak around the 3rd lag term, but this may be due to randomness, as it is barely passing the dashed blue line. The histogram (Fig 11) looks normally distributed at 0 with outliers creating a left tail.

We test the residuals for stationarity with the Augmented Dickey Fuller test (ADF). The ADF test has the null hypothesis that the data is non stationary. With a p-value of 0.01, we reject the null hypothesis because there is enough evidence to say that the residuals are stationary.

The Box-Ljung test has the null hypothesis that the data presented is independently distributed. When presented with the residuals of the ARIMA model, the test had p-values of 0.566 and 0.144 for lag =1 and lag = 10 respectively. For both of those lags, we fail to reject the null hypothesis and conclude that the data is independently distributed.

Finally, we visually inspect the histogram of the residuals (Fig.11) and the QQ plot (Fig.12) to see if the residuals appear normally distributed. The histogram has the Gaussian bell shaped curve with a few outliers. The QQ plot shows that the data matches up with the normal distribution's quantiles. With these plots, we can confidently say that the residuals are visually normally distributed.

To conclude, both diagnostic plots and statistical tests show that the residuals are stationary with mean 0, constant variance, and no autoregression or seasonality. We forecast our model to the year 2022 (Fig 13).



2.4 Atmospheric CO2 growth Forecast

We use our model to make predictions on future levels of CO2, specifically 420 and 500 ppm. We will investigate the earliest, best guess, and latest occurrence of these values. The earliest guess will be based on the first time the upper 95% confidence interval (CI) reaches the specified level and the latest guess will be the last time the value is within the lower 95% CI. The best guess will be the point estimate (mean) of the forecast.

Table 3: Predicted occurrences of key CO2 levels

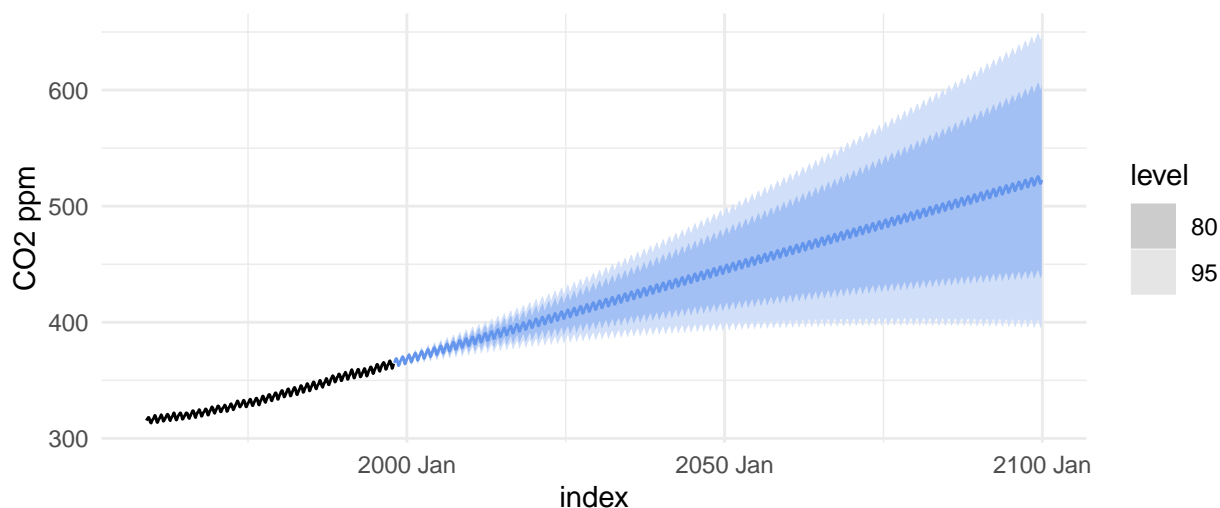
CO2.ppm.level	earliest.occurance	point_estimate	final.occurance
420 ppm	2022 Apr	2031 May	never
500 ppm	2054 Apr	2083 Apr	never

Based on our model, the first time we could potentially see CO2 at 420 ppm is 2022-04-01 because that is when the upper 95% confidence interval (CI) of our model first reaches 420 ppm. The model's lower 95% CI hovers around 420, so there is no predicted final time. Knowing what we know today

in 2023, 419 ppm was reached on May 2021, which was *before* our model's earliest guess. CO2 levels have risen faster than our model anticipated. This is a precursor to the analysis provided later in this paper. The first time our model predicts the earth to reach 500 ppm CO2 on 2054-04-01, which is when the 95% CI reaches 500 ppm. The model's lower 95% CI never reaches 500, so there is no predicted final time.

Below is the prediction of our model to the year 2100. Confidence intervals are shown fanning outward. The error of the predictions compounds overtime which expands the confidence intervals into a funnel shape. The farther out in time from the recorded data points, the less accurate the prediction.

Fig.14 CO2 levels from 1959 to 2100



3 Report from the Point of View of the Present

3.1 0b.Introduction

In our original 1997 paper, we made several predictions on the expected level atmospheric CO2. Currently, we will evaluate the accuracy of those predictions using time series analysis and extrapolate from present data to make predictions about the future.

3.2 1b.Data

A modern data pipeline was constructed to load both weekly and monthly CO2 data from January 1959 to June 2023. This will allow us to compare our forecasts in the previous section to the actual CO2 levels. The code for the pipeline can be found in the appendix.

3.3 MOVE CODE TO THE APPENDIX

```
library(zoo)

##
## Attaching package: 'zoo'
```



```

## The following object is masked from 'package:tsibble':
##
##     index

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

if (!require("lubridate")) {
  install.packages("lubridate")
}

## Loading required package: lubridate

##
## Attaching package: 'lubridate'

## The following object is masked from 'package:tsibble':
##
##     interval

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

co2_url <- "https://gml.noaa.gov/webdata/ccgg/trends/co2/co2_weekly_mlo.csv"
co2_monthly_url <- "https://gml.noaa.gov/webdata/ccgg/trends/co2/co2_mm_mlo.csv"

co2_present_raw=read.csv(co2_url,skip=51)
co2_present1 <- co2_present_raw %>%
  mutate(time_index=make_date(year,month,day)) %>%
  dplyr::select(time_index,average) %>%
  as_tsibble(index = time_index) %>%
  mutate(average =replace(average,average<=-999,NA)) %>%
  mutate(average = na.approx(average))

co2_present_raw=read.csv(co2_monthly_url,skip=56)

co2_present_month <- co2_present_raw %>%
  mutate(time_index=make_date(year,month)) %>%
  dplyr::select(time_index,average) %>%
  as_tsibble(index = time_index) %>%
  mutate(average =replace(average,average<=-999,NA)) %>%
  mutate(average = na.approx(average))

#read.csv(
#  url(co2_monthly_url,skip=56),
#  skip = 56,

```

```

# header = TRUE,
# col.names = c("year", "month", "decimal.date", "average", "deseasonalized", "ndays", "sdev",
# mutate(time_index = make_datetime(year, month)) %>%
# mutate(time_index = yearmonth(time_index)) %>%
# mutate(average = ifelse(average == -999.99, NA, average)) %>%
# fill(average) %>%
# filter(year < 2023) %>%
# as_tsibble(index = time_index) -> co2_present_monthly

glimpse(co2_present1)

## Rows: 2,565
## Columns: 2
## $ time_index <date> 1974-05-19, 1974-05-26, 1974-06-02, 1974-06-09, 1974-06-16~
## $ average <dbl> 333.37, 332.95, 332.35, 332.20, 332.37, 331.73, 331.69, 331~

glimpse(co2_present_month)

## Rows: 784
## Columns: 2
## $ time_index <date> 1958-03-01, 1958-04-01, 1958-05-01, 1958-06-01, 1958-07-01~
## $ average <dbl> 315.70, 317.45, 317.51, 317.24, 315.86, 314.93, 313.20, 312~

library(zoo)
co2_present_raw=read.csv("https://gml.noaa.gov/webdata/ccgg/trends/co2/co2_weekly_mlo.csv",skip=1)

co2_present <- co2_present_raw %>%
  mutate(time_index=lubridate::make_date(year,month,day)) %>%
  dplyr::select(time_index,average) %>%
  as_tsibble(index = time_index) %>%
  mutate(average =replace(average,average<=-999,NA)) %>%
  mutate(average = na.approx(average))

```

Fig 20. CO2 time series

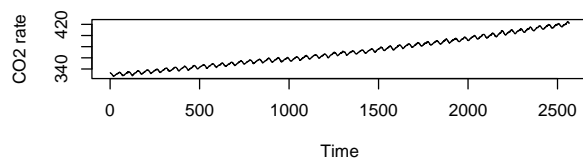


Fig 21. PACF of CO2

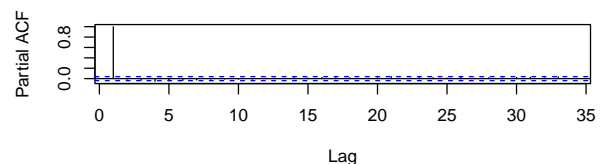


Fig 22. ACF of CO2

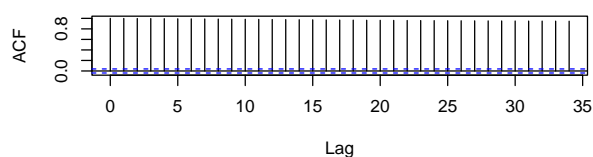
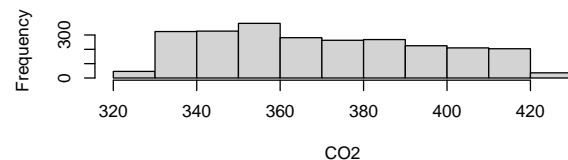


Fig 23. CO2 Distribution



```

library(ggplot2)

p1 <- autoplot(co2_present1) +
  ggtitle("Fig.20 Atmospheric CO2 concentration\n monthly average, parts per million (ppm) ") +
  xlab(NULL) + ylab(NULL)+
  theme(text = element_text(size = 8))

## Plot variable not specified, automatically selected `.vars = average`

p2 <- co2_present1 %>% index_by(year = lubridate::year(time_index)) %>%
  summarise(annual_avg = mean(value)) %>%
  mutate(annual_growth = (annual_avg / lag(annual_avg, 1) - 1) * 100) %>%
  autoplot(.vars = annual_growth) +
  xlab(NULL) + ylab(NULL) +
  ggtitle("Fig.2 Annual growth rate of concentration, %")+
  theme(text = element_text(size = 8))

p2 <- co2_present1 %>% PACF(value) %>% autoplot()+
  ggtitle("Fig.3 ACF of CO2 concentration")+
  theme(text = element_text(size = 8))

p3 <- co2_present1 %>% ACF(value) %>% autoplot()+
  ggtitle("Fig.3 ACF of CO2 concentration")+
  theme(text = element_text(size = 8))

p4 <- co2_present1 %>% ggplot() +
  geom_histogram(aes(average))+
  ggtitle("Fig.4 Histogram")

(p1|p2)/(p3|p4)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

```

Fig.20 Atmospheric CO2 concentration monthly average, parts per million (ppm)

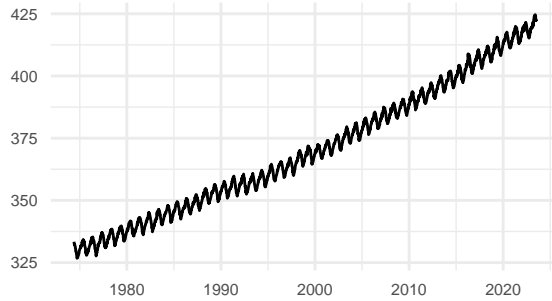


Fig.3 ACF of CO2 concentration

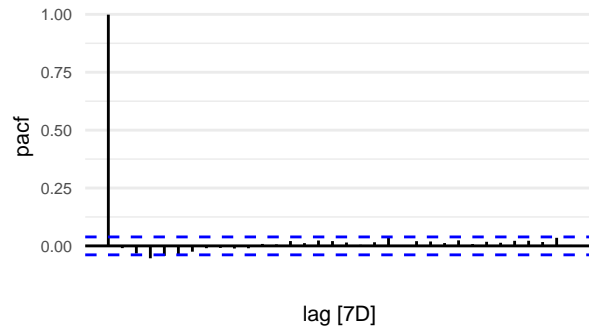


Fig.3 ACF of CO2 concentration

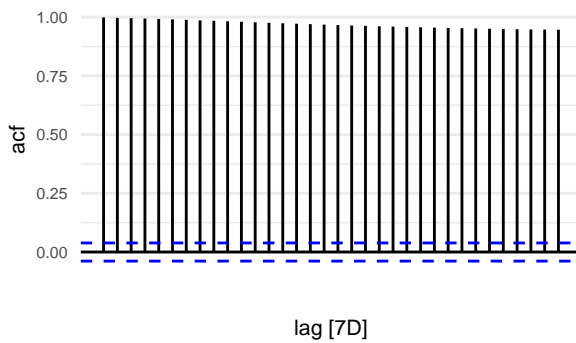
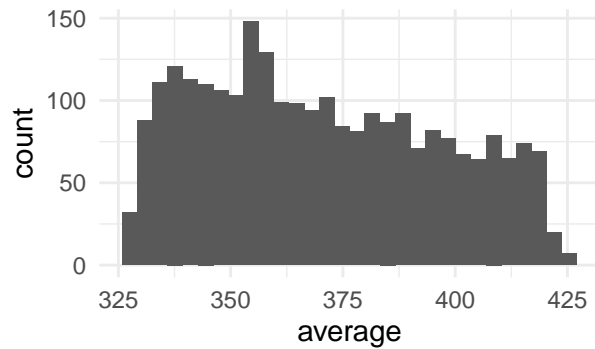


Fig.4 Histogram

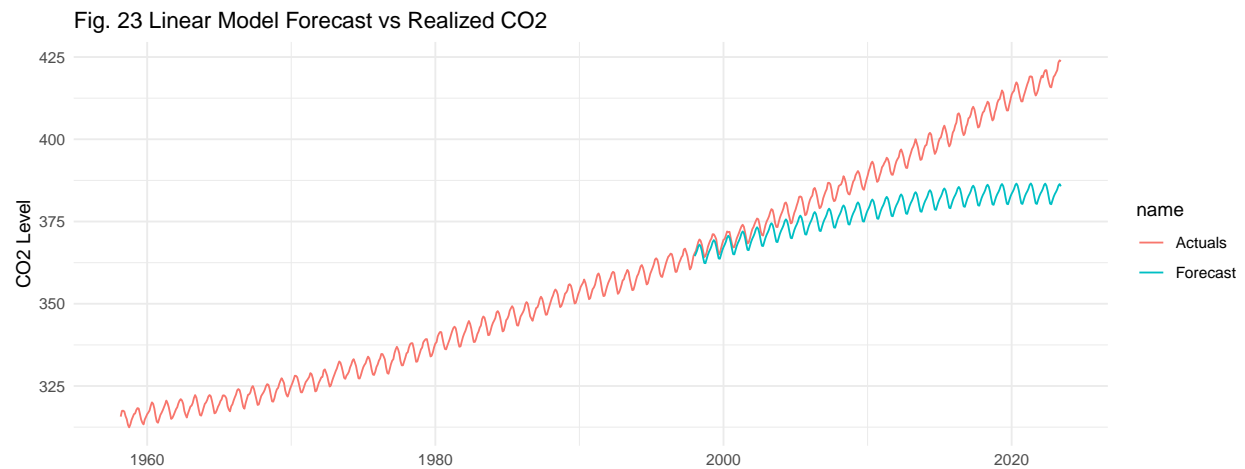
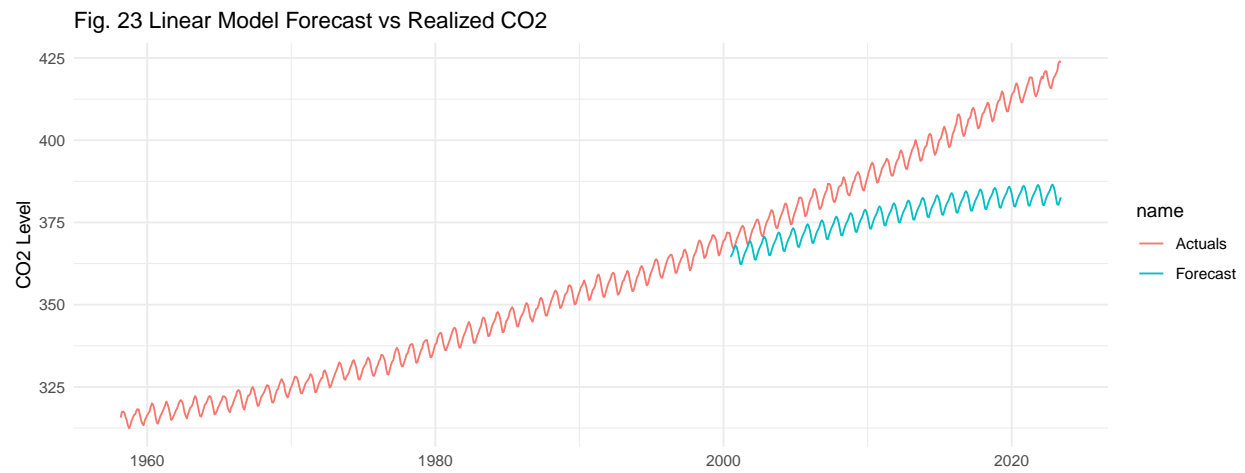


The CO2 levels have continued to grow since 1997, but the growth has not been dramatic. Fig. 20 The time series plot shows that the CO2 levels have increased at a steady rate, with no major spikes or dips.

The most notable difference between the CO2 levels in 1997 and now is the distribution of the data. In 1997, the distribution was almost bimodal, meaning that there were two distinct peaks in the data. The distribution in Fig 22. shows more heavy-tailed pattern, meaning that there are more values at the high end of the distribution. This further suggests that there are more extreme CO2 levels now than there were in 1997.

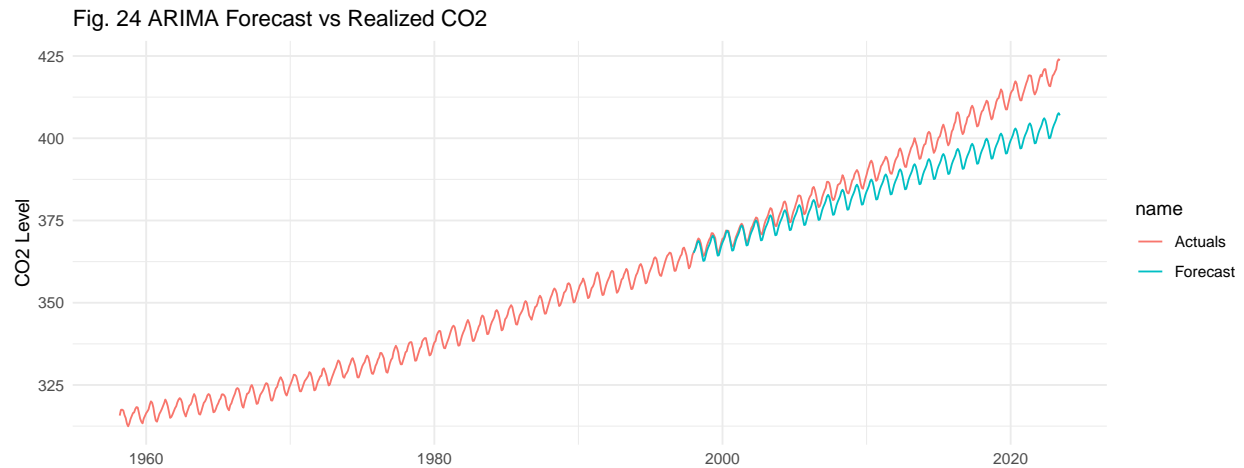
3.4 2b.Compare linear model forecasts against realized CO2

3.5 I'd recommend starting the forecast in 1998



The linear model in (Fig.23) forecast may not have capture the trend of the realized CO2 levels. The forecast appears to predict a stabilization in the CO2 levels, whereas the actual CO2 level trend increased.

3.6 3b.Compare ARIMA models forecasts against realized CO2



```
## # A tibble: 784 x 3
##   time_index name      value
##   <date>      <chr>    <dbl>
## 1 1958-03-01 Forecast     NA
## 2 1958-04-01 Forecast     NA
## 3 1958-05-01 Forecast     NA
## 4 1958-06-01 Forecast     NA
## 5 1958-07-01 Forecast     NA
## 6 1958-08-01 Forecast     NA
## 7 1958-09-01 Forecast     NA
## 8 1958-10-01 Forecast     NA
## 9 1958-11-01 Forecast     NA
## 10 1958-12-01 Forecast     NA
## # i 774 more rows
```

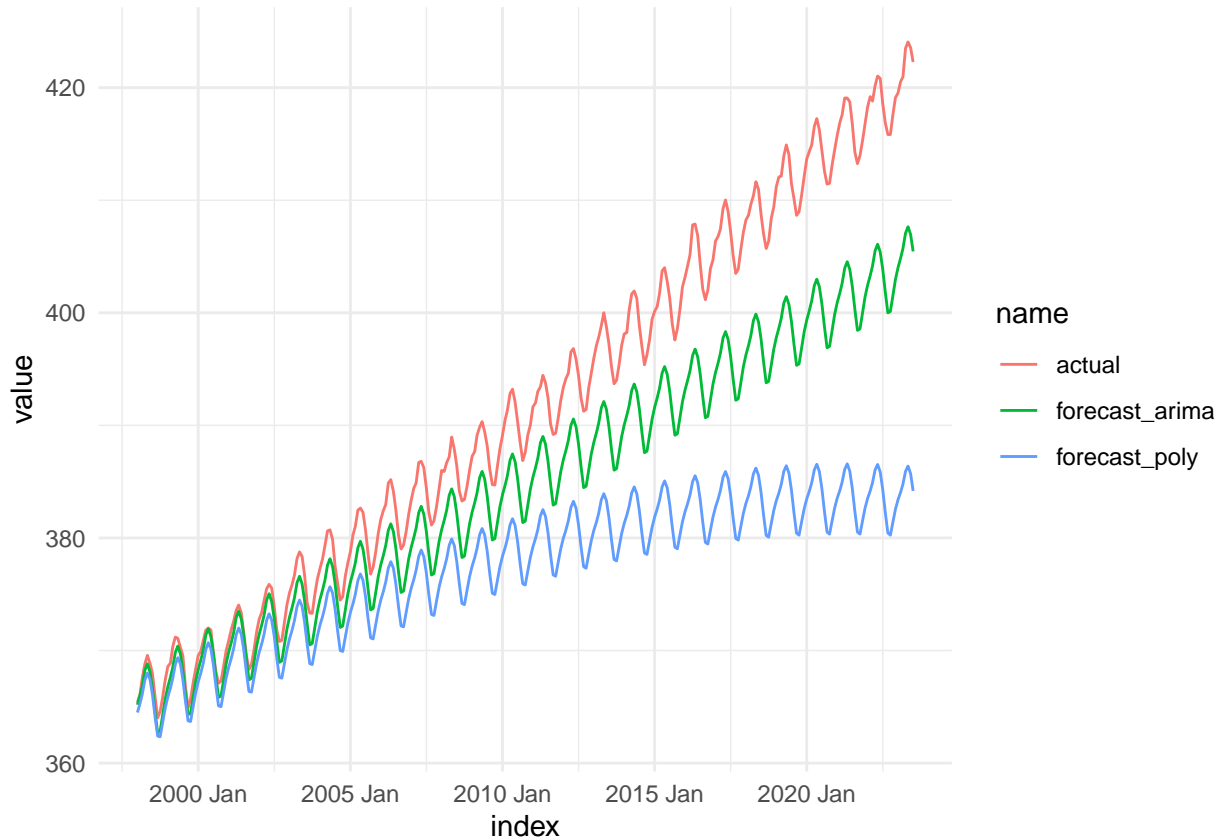
The ARIMA forecast(Fig.24) is much closer to the realized CO2 levels than the Linear Model forecast. The only difference observed, is that the ARIMA model appears to have forecasted a linear trend, while the realized CO2 levels followed an almost exponential growth.

3.7 4b.Evaluate the performance of 1997 linear and ARIMA models

```
co2_present_monthly<-co2_present %>% index_by(index=yearmonth(time_index))%>%
  summarise(value=mean(average))
co2_present_monthly_since1998 <-co2_present_monthly%>%filter(year(index)>1997)
fc_poly_new <- co2_ts %>%
  model(TSLM(value ~ trend() + I(trend() ^ 2) + I(trend() ^ 3) +
    season())) %>%forecast(h=(2022-1997)*12+7)
fc_arima_new <- model.bic %>% forecast(h=(2022-1997)*12+7)

compared_data=data.frame(index=co2_present_monthly_since1998$index,actual=co2_present_monthly_
```

```
compared_data%>%pivot_longer(cols=c(actual,forecast_poly,forecast_arima)) %>% ggplot(aes(x=index
```



```
compare_test=rbind(
  fabletools::accuracy(fc_poly_new,co2_present_monthly_since1998),
  fabletools::accuracy(fc_arima_new,co2_present_monthly_since1998)
)
compare_test$.model=c("Best Polynomial","Best ARIMA")
kable(compare_test %>% dplyr::select(-.type,-MASE,-RMSSE))
```

.model	ME	RMSE	MAE	MPE	MAPE	ACF1
Best Polynomial	14.395713	18.016760	14.395713	3.568692	3.568692	0.9887861
Best ARIMA	6.808948	8.363163	6.808948	1.690605	1.690605	0.9862961

Now we evaluate the accuracy for the best polynomial and ARIMA models built on the data till 1997. The forecast and actual values are plotted in Fig.X, and a quick glance would tell the both forecast are systematically lower than the actual data. More formally, the RMSE of prediction from the best polynomial model reaches 18.02, and that of the best ARIMA model is 8.36.

3.8 5b. Train best models on present data

In the previous section, we conducted exploratory data analysis to visually assess the forecast of both the linear model and ARIMA model for atmospheric CO2 levels. These models captured the historic trends and patterns effectively up to a certain point 1997. However as we moved from 1997, the forecasted lines started to deviate, making it challenging to determine which model better fits the data. To quantitatively evaluate the accuracy of the models, we conducted a formal evaluation using the Root Mean Squared Error (RMSE) test. Both the ARIMA and linear models were developed using data from 1974 to 1993. We used these models to forecast the atmospheric CO2 levels until 2023. Since models were built on a monthly dataset, the predictions we obtained were also on a monthly basis for the period from 1993 to 2023.

To ensure a fair comparison of these predictions we aggregated the current dataset to a monthly level. The model performance of ARIMA model, after comparing its predictions with the current dataset, resulted in an RMSE of 51.66. On the other hand, the linear model which considered both the seasonal and trend components, had an RMSE of 14.037. It looks like that data has certain trend and pattern which are better captured by linear regression.

Forecasts of Section 2 ARIMA model on Train Dataset [Seasonal adjusted] [Jan 2022 to Dec 2023]

Forecasts of Section 2 LINEAR model on Train Dataset [Seasonal adjusted] [Jan 2022 to Dec 2023]

```
## [1] "Linear model Root Mean Squared Error (RMSE): 6.16520941859495"
```

3.8.1 NON-Season

3.8.2 ARIMA model development on Non-seasonally adjusted training dataset

```
## [1] "Root Mean Squared Error (RMSE): 2.20393454232841"
```

```
test.size = 96 # fix please
```

```
model.forecasts_weekly_non_seasonal <-forecast(fit_linear_NON_season_weekly, h=test.size)
model.forecasts_weekly_non_seasonal
```

```
## # A tibble: 96 x 4 [1W]
## # Key:   .model [1]
##   .model      time_index      value .mean
##   <chr>      <week>      <dbl> <dbl>
## 1 model_trend 2021 W36 N(408, 6.6) 408.
## 2 model_trend 2021 W37 N(408, 6.6) 408.
## 3 model_trend 2021 W38 N(408, 6.6) 408.
## 4 model_trend 2021 W39 N(409, 6.6) 409.
## 5 model_trend 2021 W40 N(409, 6.6) 409.
## 6 model_trend 2021 W41 N(409, 6.6) 409.
## 7 model_trend 2021 W42 N(410, 6.6) 410.
## 8 model_trend 2021 W43 N(410, 6.6) 410.
## 9 model_trend 2021 W44 N(410, 6.6) 410.
## 10 model_trend 2021 W45 N(411, 6.6) 411.
## # i 86 more rows
```



```

# Merge the two data frames on the 'index' column, and handle missing values with 0
merged_df_weekly_non_seasonal <- merge(model.forecasts_weekly_non_seasonal, test_data_non_seasonal, by = "index")
# merged_df_yearly[is.na(merged_df)] <- 0 # Replace missing values with 0

# Select the desired columns and rename them if needed
selected_weekly_non_seasonal <- merged_df_weekly_non_seasonal %>%
  select(time_index, value.y, .mean) %>%
  rename(predicted = .mean, current_value = value.y)

# Calculate the squared differences between 'predicted' and 'current_value'
selected_weekly_non_seasonal$diff_squared <- (selected_weekly_non_seasonal$predicted - selected_weekly_non_seasonal$current_value)^2

# Calculate the RMSE
rmse_weekly_non_seasonal <- sqrt(mean(selected_weekly_non_seasonal$diff_squared))

# Print the RMSE
print(paste("Root Mean Squared Error (RMSE):", rmse_weekly_non_seasonal))

## [1] "Root Mean Squared Error (RMSE): 6.10044846421003"

```

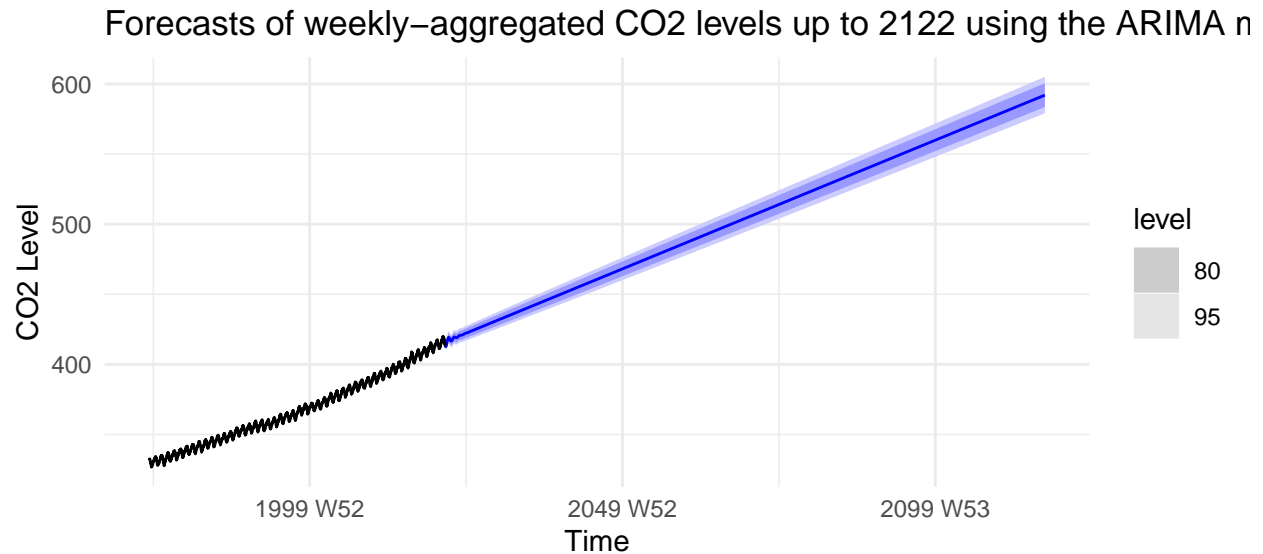
3.8.3 (3 points) Task Part 6b: How bad could it get?

With the non-seasonally adjusted data series, generate predictions for when atmospheric CO₂ is expected to be at 420 ppm and 500 ppm levels for the first and final times (consider prediction intervals as well as point estimates in your answer). Generate a prediction for atmospheric CO₂ levels in the year 2122. How confident are you that these will be accurate predictions? Generate a prediction for atmospheric CO₂ levels in the year 2100. How confident are you that these will be accurate predictions?

Observations The best model we derived from Part 5b was the ARIMA model. Utilizing this model, we projected CO₂ levels on the original weekly dataset which is a non-seasonally adjusted data. Our analysis indicated that CO₂ levels are predicted to reach 420 ppm in 2023, but are not expected to reach 500 ppm. We observed that in the year 3075, the upper boundary touched 500 ppm, but with low confidence. These predictions can be influenced by the data's aggregation level. For instance, weekly aggregated data may yield more vague predictions because the number of datapoints are larger resulting in higher noise and variability in data, whereas monthly aggregated data can provide slightly improved predictions. On examining the weekly level aggregated data, we noticed that the prediction boundary intervals curve downwards, which is not ideal because it captures too much noise and adds variability. However, with monthly level data, predictions are comparatively better, and the boundary intervals are narrower and upward. According to our analysis, the 500 ppm threshold is projected to be reached by the year 2050.

The predictions for 2122 is 425 with boundary limits of 310-555 in the weekly aggregated dataset. However, in the monthly aggregated dataset, the prediction for 2122 is 650 with narrower boundary levels. Another factor that can impact these predictions is the evaluation metric used. In our analysis, we employed the Bayesian Information Criterion (BIC). If an alternative metric had been employed, the predictions might have varied. The use of BIC penalizes the data more, leading to a more conservative model selection that favors simpler models. As a result, the chosen model may be too simplistic to accurately capture the underlying patterns and relationships in the future unseen

data. Hence, considering alternative evaluation metrics could potentially yield different or more accurate predictions.

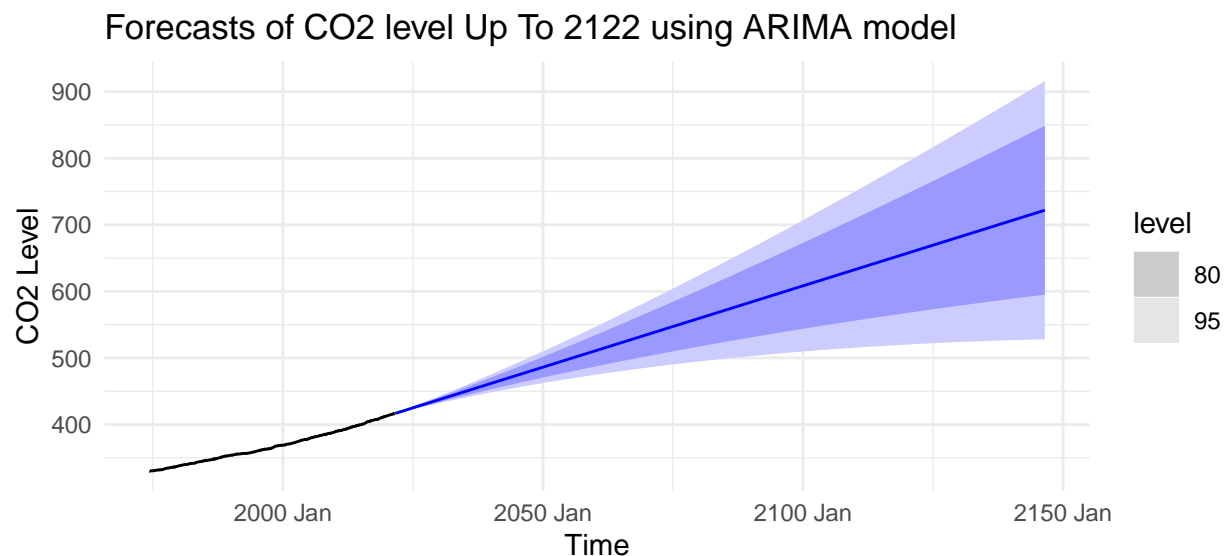


Year 2122 forecast and on Weekly Dataset

```
model_yearly_ARIMA_forecast <- fabletools::forecast(model.bic_monthlyyearly, h=1500)

model_yearly_ARIMA_forecast %>%
  autoplot(train_data_seasonally_adjusted) +
  labs(
    title = "Forecasts of CO2 level Up To 2122 using ARIMA model",
    y = "CO2 Level", x = "Time"

    # 650
  )
```



```
test.size = 12784 # weekly time series unit
```

```
model.forecasts_ARIMA_weekly <-forecast(model.bic_weekly_ARIMA, h=test.size)
model.forecasts_ARIMA_weekly
```

```
## # A fable: 12,784 x 4 [1W]
```

```
## # Key:       .model [1]
```

```
##   .model                                time_index      value .mean
##   <chr>                                <week>          <dist> <dbl>
## 1 "fable::ARIMA(value ~ 0:1 + pdq(0:4, 0:2, 0:4)~ 2021 W36 N(413, 0.23) 413.
## 2 "fable::ARIMA(value ~ 0:1 + pdq(0:4, 0:2, 0:4)~ 2021 W37 N(413, 0.36) 413.
## 3 "fable::ARIMA(value ~ 0:1 + pdq(0:4, 0:2, 0:4)~ 2021 W38 N(413, 0.5) 413.
## 4 "fable::ARIMA(value ~ 0:1 + pdq(0:4, 0:2, 0:4)~ 2021 W39 N(413, 0.68) 413.
## 5 "fable::ARIMA(value ~ 0:1 + pdq(0:4, 0:2, 0:4)~ 2021 W40 N(413, 0.91) 413.
## 6 "fable::ARIMA(value ~ 0:1 + pdq(0:4, 0:2, 0:4)~ 2021 W41 N(413, 1.2) 413.
## 7 "fable::ARIMA(value ~ 0:1 + pdq(0:4, 0:2, 0:4)~ 2021 W42 N(413, 1.5) 413.
## 8 "fable::ARIMA(value ~ 0:1 + pdq(0:4, 0:2, 0:4)~ 2021 W43 N(413, 1.8) 413.
## 9 "fable::ARIMA(value ~ 0:1 + pdq(0:4, 0:2, 0:4)~ 2021 W44 N(414, 2.2) 414.
##10 "fable::ARIMA(value ~ 0:1 + pdq(0:4, 0:2, 0:4)~ 2021 W45 N(414, 2.6) 414.
## # i 12,774 more rows
```

```
# predicted co2 value 420
```

```
# Whether 420ppm CO2 present in data or not?
```

```
# period for [Jan 2022 to Dec 2122]
```

```
library(dplyr)
```

```
# Assuming your model.forecasts_ARIMA_weekly contains the columns 'date' and '.mean'
```

```
# Check if there is any value near 420 (within a range of +/- 10) in the .mean column
```

```
value_420ppm <- any(abs(model.forecasts_ARIMA_weekly$.mean - 420) <= 10)
```

```
# Print the result
```

```
print(value_420ppm)
```

```
## [1] TRUE
```

```
# **420 ppm CO2 level is not present in the forecasted data until the year 2122. Hence upper a
```

```
# predicted co2 value 500
```

```
# Whether 500ppm CO2 present in data or not?
```

```
# period for [Jan 2022 to Dec 2122]
```

```
library(dplyr)
```

```
# Assuming your model.forecasts_ARIMA_weekly contains the columns 'date' and '.mean'
```

```
# Check if there is any value near 420 (within a range of +/- 10) in the .mean column
```

```
value_500ppm <- any(abs(model.forecasts_ARIMA_weekly$.mean - 500) <= 10)
```

```
# Print the result
print(value_500ppm)
```

```
## [1] TRUE
```

```
# **500 ppm CO2 level is not present in the forecasted data until the year 2122.Hence upper and lower bound
```

```
# ## Confidence interval
```

```
model.forecasts_ARIMA_weekly_confidence_interval <-model.forecasts_ARIMA_weekly %>% mutate(upper_bound =
model.forecasts_ARIMA_weekly_confidence_interval
```

```
## # A tibble: 12,784 x 6 [1W]
```

```
## # Key:   .model [1]
```

```
##   .model                                time_index      value .mean upper lower
##   <chr>                                <week>         <dist> <dbl> <dbl> <dbl>
## 1 "fable::ARIMA(value ~ 0:1 + pdq(0:~ 2021 W36 N(413, 0.23) 413. 843. 439.
## 2 "fable::ARIMA(value ~ 0:1 + pdq(0:~ 2021 W37 N(413, 0.36) 413. 843. 439.
## 3 "fable::ARIMA(value ~ 0:1 + pdq(0:~ 2021 W38 N(413, 0.5) 413. 843. 439.
## 4 "fable::ARIMA(value ~ 0:1 + pdq(0:~ 2021 W39 N(413, 0.68) 413. 843. 439.
## 5 "fable::ARIMA(value ~ 0:1 + pdq(0:~ 2021 W40 N(413, 0.91) 413. 843. 439.
## 6 "fable::ARIMA(value ~ 0:1 + pdq(0:~ 2021 W41 N(413, 1.2) 413. 843. 439.
## 7 "fable::ARIMA(value ~ 0:1 + pdq(0:~ 2021 W42 N(413, 1.5) 413. 843. 439.
## 8 "fable::ARIMA(value ~ 0:1 + pdq(0:~ 2021 W43 N(413, 1.8) 413. 843. 439.
## 9 "fable::ARIMA(value ~ 0:1 + pdq(0:~ 2021 W44 N(414, 2.2) 414. 843. 439.
## 10 "fable::ARIMA(value ~ 0:1 + pdq(0:~ 2021 W45 N(414, 2.6) 414. 843. 439.
## # i 12,774 more rows
```