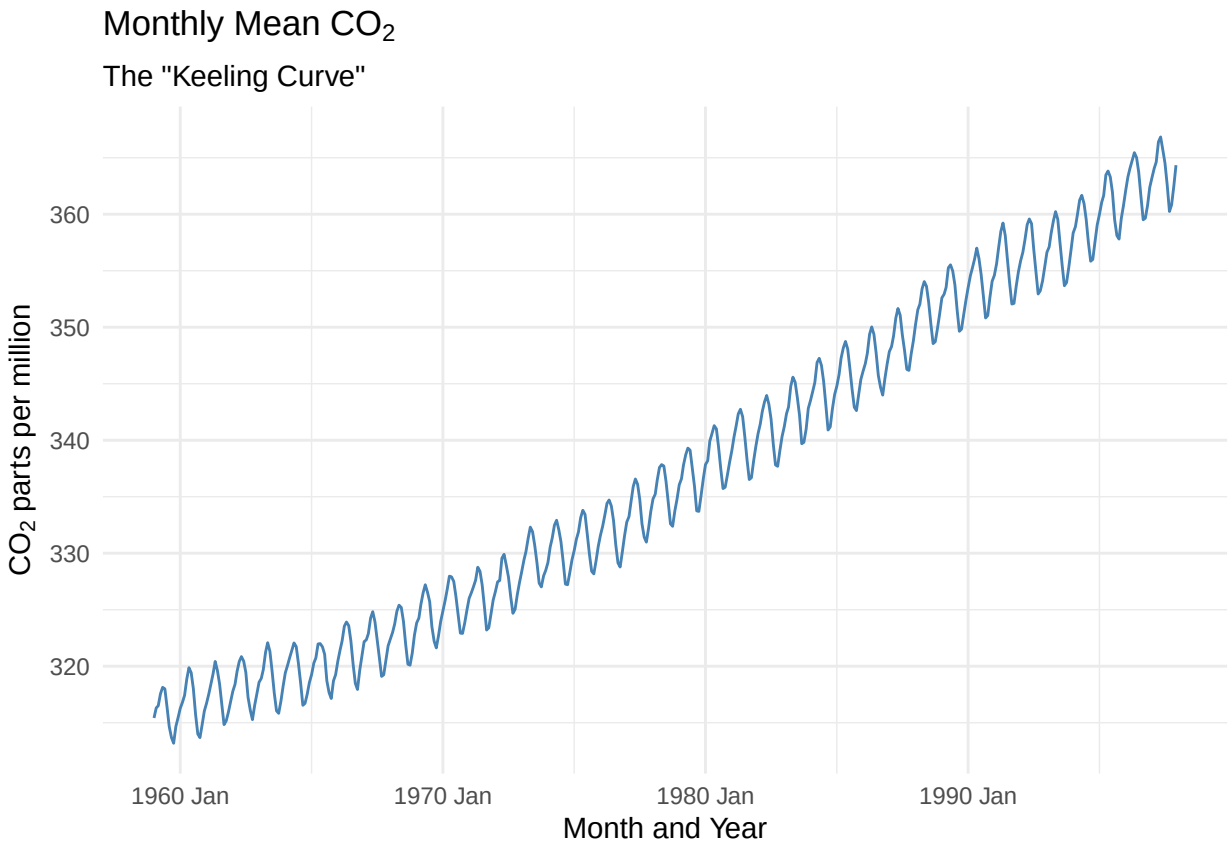
Statistical Methods for Discrete Response, Time Series, and Panel Data (W271): Lab 2

The Keeling Curve

In the 1950s, the geochemist Charles David Keeling observed a seasonal pattern in the amount of carbon dioxide present in air samples collected over the course of several years. He was able to attribute this pattern to the variation in global rates of photosynthesis throughout the year, caused by the difference in land area and vegetation cover between the Earth's northern and southern hemispheres.

In 1958 Keeling began continuous monitoring of atmospheric carbon dioxide concentrations from the Mauna Loa Observatory in Hawaii and soon observed a trend increase carbon dioxide levels in addition to the seasonal cycle. He was able to attribute this trend increase to growth in global rates of fossil fuel combustion. This trend has continued to the present, and is known as the "Keeling Curve."


Monthly Mean CO_2
The "Keeling Curve"
CO_2 parts per million
360
350
340
330
320
1960 Jan
1970 Jan
1980 Jan
1990 Jan
Month and Year

Your Assignment

Your goal in this assignment is to produce a comprehensive analysis of the Mona Loa CO2 data that you will be read by an interested, supervising data scientist. Rather than this being a final report, you might think of this as being a contribution to your laboratory. You and your group have been initially charged with the task of investigating the trends of global CO2, and told that if you find "anything interesting" that the team may invest more resources into assessing the question.

Because this is the scenario that you are responding to:

1. Your writing needs to be clear, well-reasoned, and concise. Your peers will be reading this, and you have a reputation to maintain.
2. Decisions that you make for your analysis need also be clear and well-reasoned. While the main narrative of your deliverable might only present the modeling choices that you determine are the most appropriate, there might exist supporting materials that examine what the consequences of other choices would be. As a concrete example, if you determine that a series is an AR(1) process your main analysis might provide the results of the critical test that led you to that determination and the results of the rest of the analysis under AR(1) modeling choices. However, in an appendix or separate document that is linked in your main report, you might show what a MA model would have meant for your results instead.
3. Your code and repository are a part of the deliverable. If you were to make a clear argument that this is a question worth pursuing, but then when the team turned to continue the work they found a repository that was a jumble of coding idioms, version-ed or outdated files, and skeletons it would be a disappointment.

Report from the Point of View of 1997

For the first part of this task, suspend reality for a short period of time and conduct your analysis from the point of view of a data scientist doing their work in the early months of 1998. Do this by using data that is included in *every* R implementation, the `co2` dataset. This dataset is lazily loaded with every R instance, and is stored in an object called `co2`.

```
co2 <- as_tsibble(co2) %>% filter(year(index)<1998)
```

(3 points) Task 0a: Introduction

Introduce the question to your audience. Suppose that they *could* be interested in the question, but they don't have a deep background in the area. What is the question that you are addressing, why is it worth addressing, and what are you going to find at the completion of your analysis. Here are a few resource that you might use to start this motivation.

- Wikipedia
- First Publication
- Autobiography of Keeling

(3 points) Task 1a: CO2 data

Conduct a comprehensive Exploratory Data Analysis on the `co2` series. This should include (without being limited to) a description of how, where and why the data is generated, a thorough investigation of the trend, seasonal and irregular elements. Trends both in levels and growth rates should be discussed (consider expressing longer-run growth rates as annualized averages).

What you report in the deliverable should not be your own process of discovery, but rather a guided discussion that you have constructed so that your audience can come to an understanding as succinctly and successfully as possible. This means that figures should be thoughtfully constructed and what you learn from them should be discussed in text; to the extent that there is *any* raw output from your analysis, you should intend for people to read and interpret it, and you should write your own interpretation as well.

```
p1 <- autoplot(co2) +geom_smooth(color="lightgrey")+
  ggtitle("Fig.1 Atmospheric CO2 concentration\n monthly average, parts per million (ppm) ") +
  xlab(NULL) + ylab(NULL)
```

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

```
p2 <- co2 %>% index_by(year = year(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\n concentration, %")
p3 <- gg_season(co2) +
  xlab(NULL) + ylab(NULL) +
  ggtitle("Fig.3 Seasonal plot of CO2 concentration")
```

```
## Plot variable not specified, automatically selected `y = value`
```

```
p4 <- co2 %>% model(STL(value ~ trend(window = 120) + season(window = "periodic"),
                        robust = TRUE)) %>%
  components() %>% pull(remainder) %>% gghistogram() +
  ggtitle("Fig.4 Histogram of irregular\n component by STL")
# p3 <- ggAcf(co2$value)
# p4 <- ggPacf(co2$value)
(p1 | p2) / (p3 | p4)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```

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

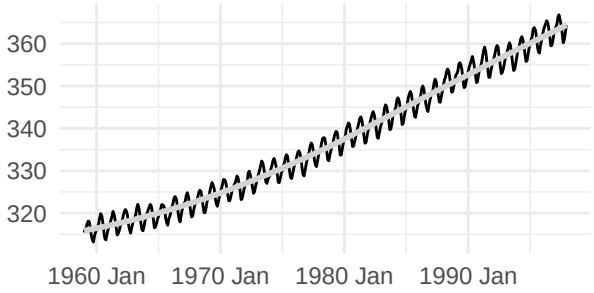


Fig.2 Annual growth rate of concentration, %

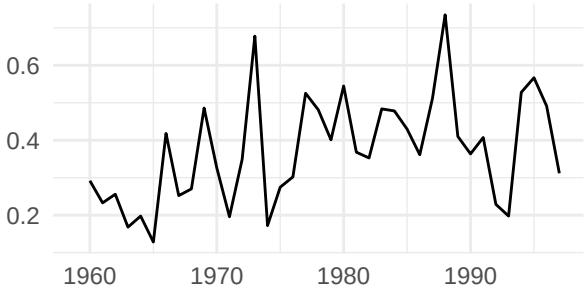


Fig.3 Seasonal plot of CO2 concentration

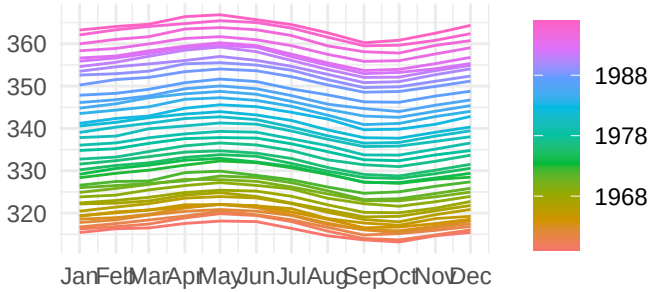
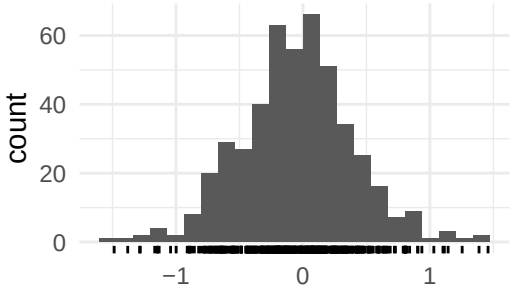


Fig.4 Histogram of irregular component by STL



.

The data measures the monthly average atmospheric CO2 concentration from 1959 to 1997, expressed in parts per million (ppm). It was initially collected by a 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.

Another feature of the data is its robust seasonal pattern, with peak in May and bottom in October almost every year (see Fig.3). Keeling believes it was the result of the activity of land plants.

Fig.4 is the histogram of the remaining or irregular components after removing the trend and the seasonal components from the data with STL[1]. It

[1]Cleveland, 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.

looks like a normal distribution without obvious outliers.

```
co2 %>%
  features(value, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1      7.82        0.01
```

```
co2 %>%
  mutate(d_value=difference(value)) %>%
  features(d_value, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1    0.0124         0.1
```

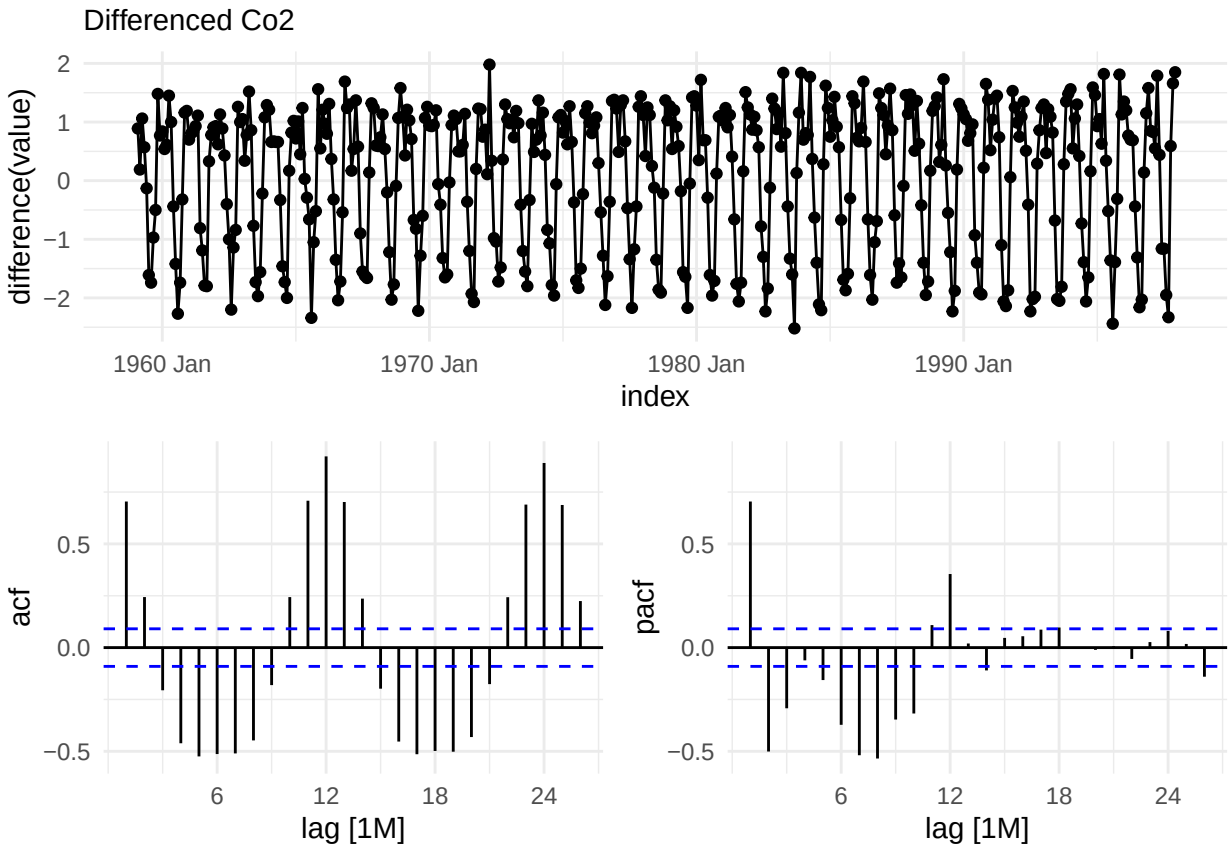
```
co2 %>%
  features(value, unitroot_ndiffs)
```

```
## # A tibble: 1 x 1
##   ndiffs
##    <int>
## 1      1
```

```
co2 %>% gg_tsdisplay(difference(value), plot_type="partial") +labs(subtitle = "Differenced Co2")
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```



(3 points) Task 2a: Linear time trend model

Fit a linear time trend model to the `co2` series, and examine the characteristics of the residuals. Compare this to a quadratic time trend model. Discuss whether a logarithmic transformation of the data would be appropriate. Fit a polynomial time trend model that incorporates seasonal dummy variables, and use this model to generate forecasts to the year 2020.

```
linear_trend_model<-co2 %>% model(TSLM(value~trend()))
```

Since the long term trend of the CO_2 data looks linear and the variation around the trend seems stable, a log transformation of the data is not necessary (which is also supported by the stable and symmetric residuals in Fig.5) and we can fit the original data with a linear time trend model as:

$$\mathrm{CO}_2 = \beta_0 + \beta_1 t + \epsilon_t \tag{1}$$

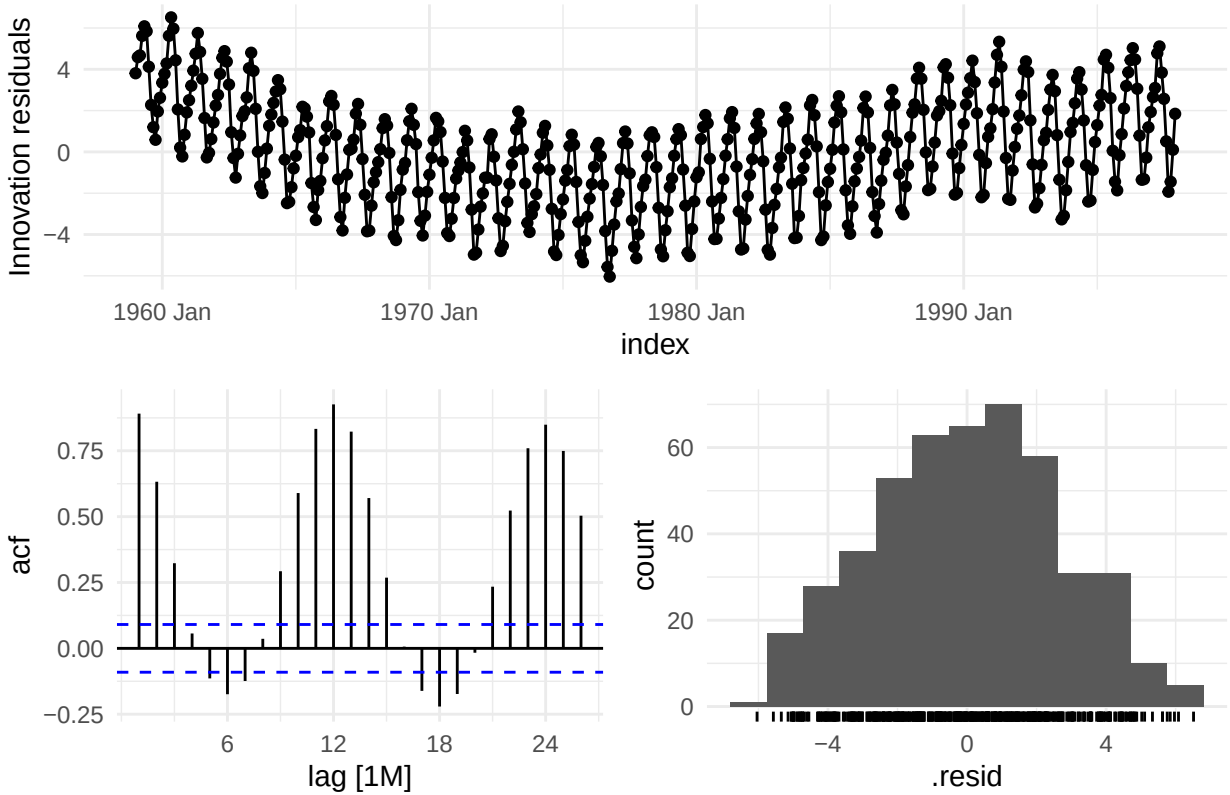
, which gives the parameters as:

$$\mathrm{CO}_2 = 311.5 + 0.11 * t + \epsilon_t \tag{2}$$

This linear trend model implies that the CO_2 concentration increased 0.11/month on average during 1959 to 1997. However, the residuals plots in Fig.5 suggest this simple linear trend model is not adequate since: 1) the mean of the residual forms a "U" shape along time, suggesting a quadratic or higher order polynomial trend model may be better; 2) the ACF plots indicates strong seasonal patterns exists in the residuals, suggesting seasonal dummy variables should be included in the model.

```
gg_tsresiduals(linear_trend_model) + ggtitle("Fig.5 Residual plot of the linear trend model")
```

Fig.5 Residual plot of the linear trend model




```
co2_copy <- co2 %>% append_row(600) %>%
  mutate(
    num_index = time(index),
    num_index_qudratic = num_index ^ 2,
    num_index_cubic = num_index ^ 3,
  )
for (i in 1:11) {
  name =
    co2_copy <-
    co2_copy %>% mutate("month_{i}" := ifelse(month(index) == i, 1, 0))
}
```

```
co2_training = co2_copy %>% filter(year(index) < 1991)
co2_valid = co2_copy %>% filter(year(index) < 1998,year(index) >= 1991)
co2_forecast = co2_copy %>% filter(year(index) >= 1998)

# stargazer(model_linear,model_quadratic,model_cubic,type="text",
#           add.lines=list(c("AIC", round(AIC(model_linear),1), round(AIC(model_quadratic),1), round(AIC(model_cubic),1)),
#                          c("BIC", round(BIC(model_linear),1), round(BIC(model_quadratic),1), round(BIC(model_cubic),1))))
dummy_name=paste0("month_",1:11,collapse = "+")
fit <- co2_training |>
  model(
    model_linear = TSLM(as.formula(paste0("value ~ num_index   +",dummy_name))),
    model_quadratic = TSLM(as.formula(paste0("value ~ num_index  + num_index_qudratic +",dummy_name))),
    model_cubic = TSLM(as.formula(paste0("value ~ num_index + num_index_qudratic + num_index_cubic +",dummy_name)))
  )
report(fit)
```

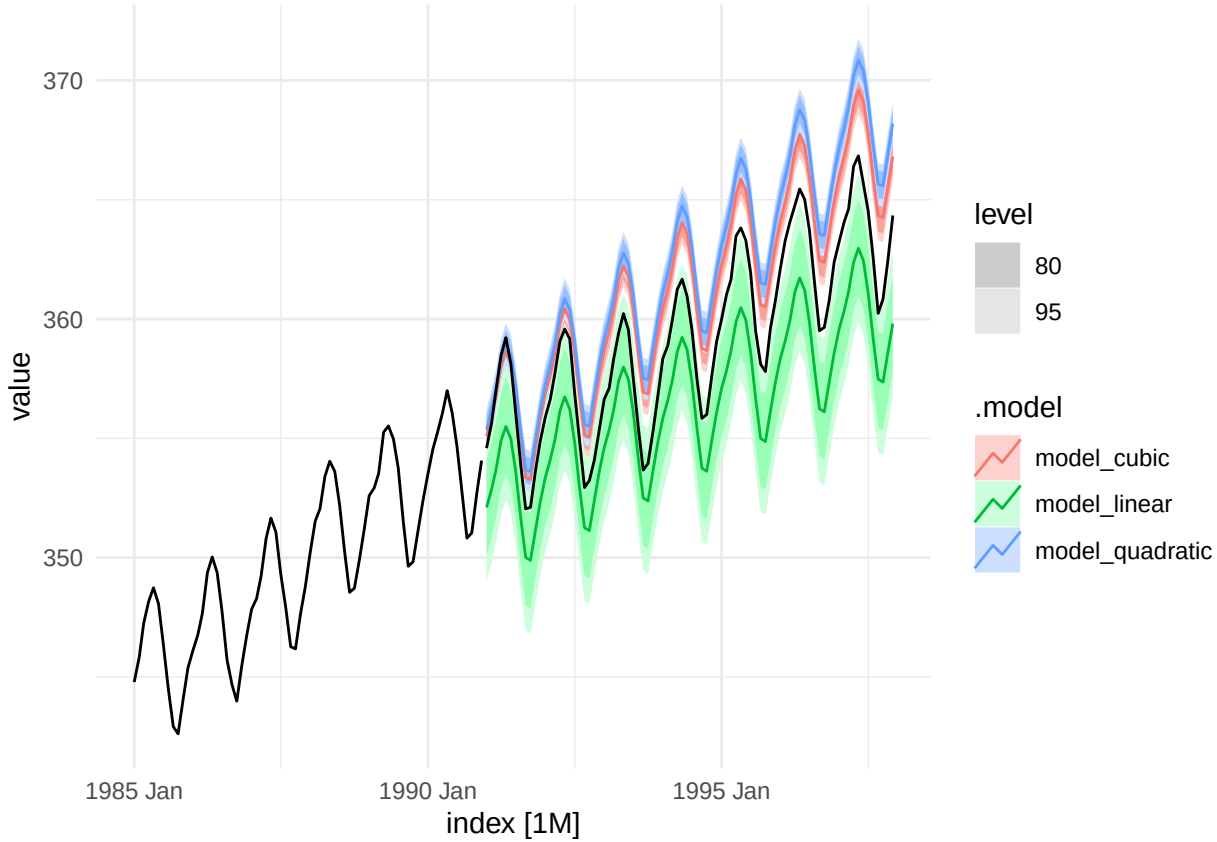
```
## Warning in report.mdl_df(fit): Model reporting is only supported for individual
## models, so a glance will be shown. To see the report for a specific model, use
## `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 3 x 15
##   .model  r_squared adj_r_squared sigma2 statistic   p_value    df log_lik   AIC
##   <chr>       <dbl>         <dbl>  <dbl>     <dbl>     <dbl> <int>   <dbl> <dbl>
## 1 model_~     0.983         0.983  2.35      1842. 4.35e-322    13   -702.  343.
## 2 model_~     0.999         0.999  0.182    22346. 0            14   -210. -639.
## 3 model_~     0.999         0.999  0.172    21942. 0            15   -199. -660.
## # i 6 more variables: AICc <dbl>, BIC <dbl>, CV <dbl>, deviance <dbl>,
## #   df.residual <int>, rank <int>
```

```
vd <- forecast(fit,new_data = co2_valid)
co2_training %>% filter(index>=yearmonth("1985M01")) %>% autoplot(value,PI = FALSE)+autolayer(vd)+autolayer(co2_valid)
```

```
## Warning in geom_line(...): Ignoring unknown parameters: `PI`
```

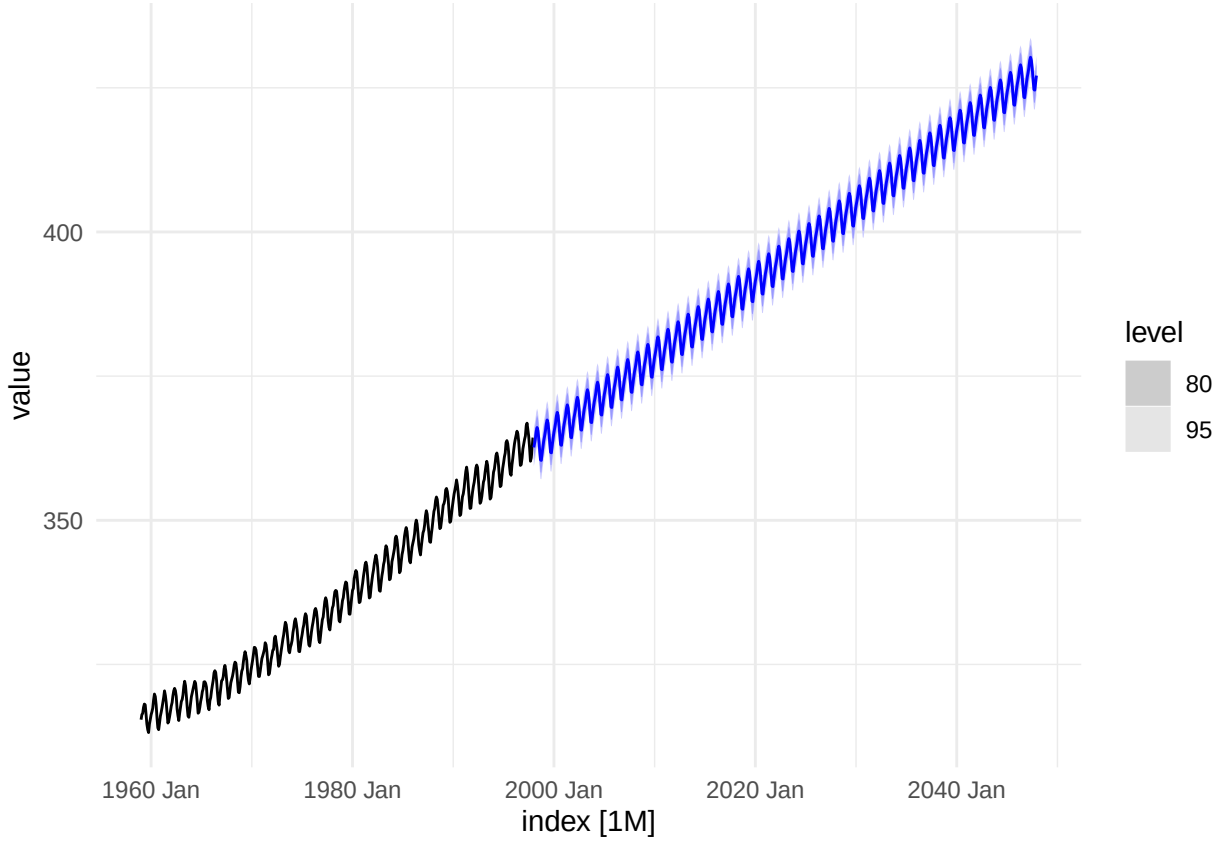
```
## Plot variable not specified, automatically selected `.vars = value`
```




```
fabletools::accuracy(vd,co2_valid)
```

```
## # A tibble: 3 x 10
##   .model          .type    ME  RMSE   MAE    MPE  MAPE  MASE RMSSE  ACF1
##   <chr>           <chr> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 model_cubic     Test  -2.09  2.28  2.12 -0.581 0.590   NaN   NaN 0.915
## 2 model_linear    Test   2.83  2.93  2.83  0.786 0.786   NaN   NaN 0.874
## 3 model_quadratic Test  -2.84  3.08  2.85 -0.790 0.792   NaN   NaN 0.937
```

```r
# although the result above prefer the cubic model, I would suggest a linear or quadratic one.
final_model_poly <- co2_copy %>% filter(year(index)<1998) %>%
  model(TSLM(as.formula(paste0("value ~ num_index + ",dummy_name))))
fc_linear <- final_model_poly %>% forecast(co2_forecast)
co2_copy %>% filter(year(index)<1998) %>% autoplot(value)+autolayer(fc_linear)
```



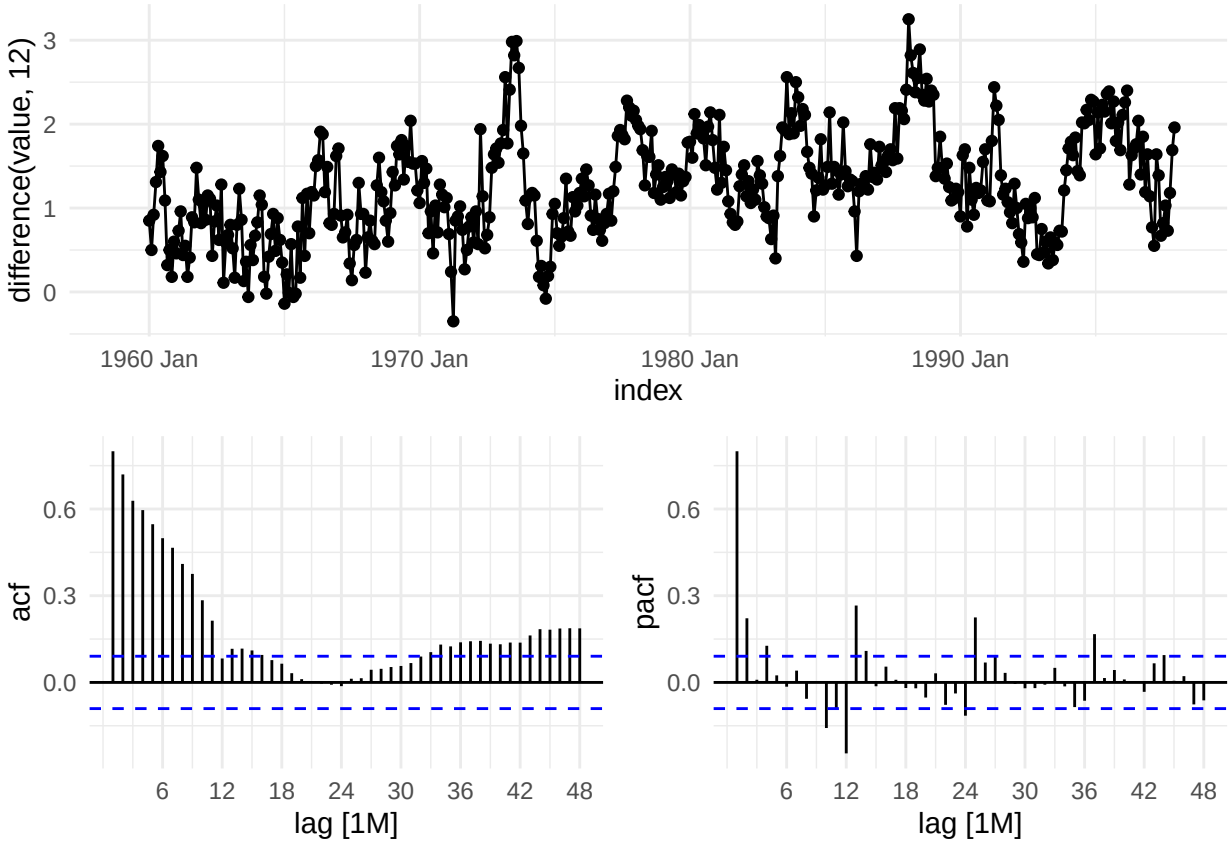
(3 points) Task 3a: ARIMA times series model

Following all appropriate steps, choose an ARIMA model to fit to the series. Discuss the characteristics of your model and how you selected between alternative ARIMA specifications. Use your model (or models) to generate forecasts to the year 2022.

```
co2 |>
  gg_tsdisplay(difference(value,12),lag_max=48, plot_type='partial')
```

```
## Warning: Removed 12 rows containing missing values (`geom_line()`).
```

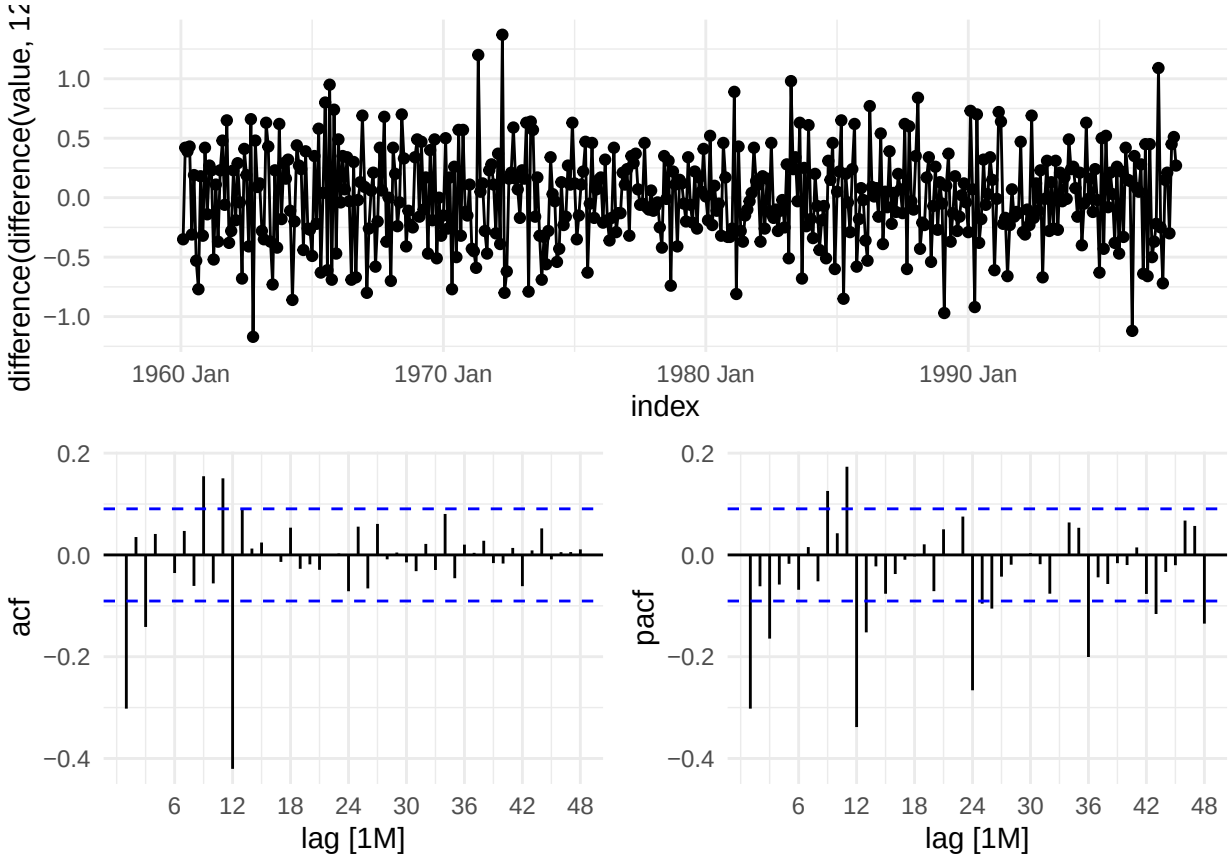
```
## Warning: Removed 12 rows containing missing values (`geom_point()`).
```




```
# still non-stationary after seasonal difference, so one more difference
co2 |>
  gg_tsdisplay(difference(difference(value,12)),lag_max=48, plot_type='partial')
```

```
## Warning: Removed 13 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```




```
fit <- co2_training |>
  model(
    arima111011 = ARIMA(value ~ pdq(1,1,1) + PDQ(0,1,1)),
    arima110011 = ARIMA(value ~ pdq(1,1,0) + PDQ(0,1,1)),
    arima011011 = ARIMA(value ~ pdq(0,1,1) + PDQ(0,1,1)),
    arima013011 = ARIMA(value ~ pdq(0,1,3) + PDQ(0,1,1)),
```

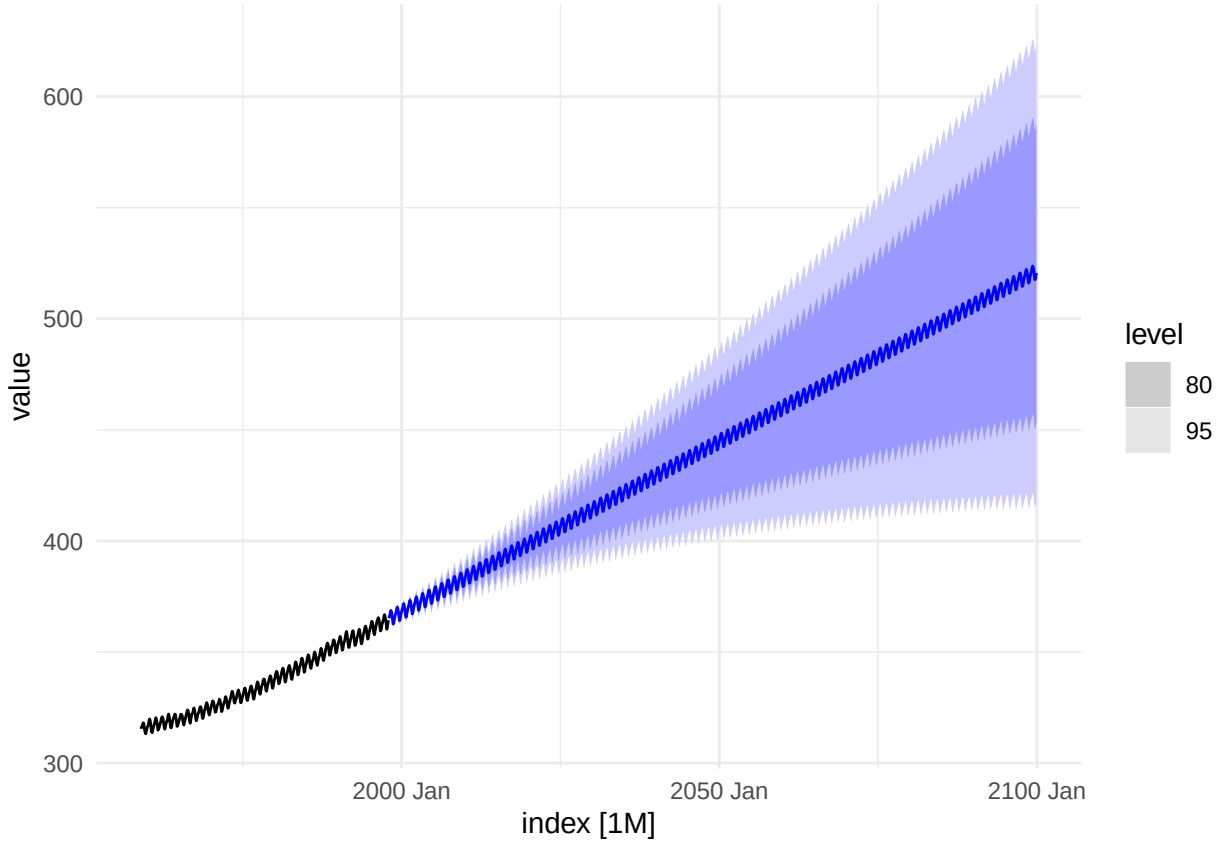
```
    arima310011 = ARIMA(value ~ pdq(3,1,0) + PDQ(0,1,1))
  )
glance(fit) |> arrange(AICc) |> select(.model:BIC)
```

```
## # A tibble: 5 x 6
##   .model      sigma2 log_lik   AIC  AICc   BIC
##   <chr>        <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1 arima013011 0.0838   -64.7  139.  140.  159.
## 2 arima011011 0.0842   -66.8  140.  140.  151.
## 3 arima111011 0.0841   -66.0  140.  140.  156.
## 4 arima310011 0.0844   -66.2  142.  142.  162.
## 5 arima110011 0.0852   -69.5  145.  145.  157.
```

```
vd <- fit %>% forecast(co2_valid)
fabletools::accuracy(vd,co2_valid)
```

```
## # A tibble: 5 x 10
##   .model      .type    ME  RMSE   MAE    MPE  MAPE  MASE RMSSE  ACF1
##   <chr>       <chr> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arima011011 Test  -1.13  1.27  1.16 -0.315 0.324   NaN   NaN 0.850
## 2 arima013011 Test  -1.03  1.18  1.07 -0.287 0.299   NaN   NaN 0.848
## 3 arima110011 Test  -1.16  1.29  1.19 -0.322 0.330   NaN   NaN 0.847
## 4 arima111011 Test  -1.05  1.20  1.09 -0.293 0.304   NaN   NaN 0.848
## 5 arima310011 Test  -1.11  1.25  1.15 -0.309 0.319   NaN   NaN 0.850
```

```
final_model_arima <- co2_copy %>% filter(year(index)<1998) %>%
  model(arima013011 = ARIMA(value ~ pdq(0,1,3) + PDQ(0,1,1)))
fc_arima <- final_model_arima %>% forecast(h=1224)
co2_copy %>% filter(year(index)<1998) %>% autoplot(value)+autolayer(fc_arima)
```



(3 points) Task 4a: Forecast atmospheric CO2 growth

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 2100. How confident are you that these will be accurate predictions?

```
fc <-fc_arima %>% mutate(upper=quantile(value,0.95),lower=quantile(value,0.05))
fc %>% filter(.mean>=420) %>% head()
```

```
## # A fable: 6 x 6 [1M]
```

```
## # Key:     .model [1]
##   .model          index       value .mean upper lower
##   <chr>           <mth>      <dist> <dbl> <dbl> <dbl>
## 1 arima013011 2032 Apr N(420, 146)  420.  440.  400.
## 2 arima013011 2032 May N(421, 147)  421.  441.  401.
## 3 arima013011 2032 Jun N(420, 148)  420.  440.  400.
## 4 arima013011 2033 Mar N(421, 157)  421.  441.  400.
## 5 arima013011 2033 Apr N(422, 158)  422.  442.  401.
## 6 arima013011 2033 May N(422, 158)  422.  443.  402.
```

```
fc %>% filter(.mean>=500) %>% head()
```

```
## # A fable: 6 x 6 [1M]
## # Key:     .model [1]
##   .model          index        value .mean upper lower
##   <chr>           <mth>       <dist> <dbl> <dbl> <dbl>
## 1 arima013011 2084 Apr N(500, 1730)  500.  569.  432.
## 2 arima013011 2084 May N(501, 1735)  501.  569.  432.
## 3 arima013011 2084 Jun N(500, 1739)  500.  569.  431.
## 4 arima013011 2085 Mar N(500, 1782)  500.  570.  431.
## 5 arima013011 2085 Apr N(502, 1786)  502.  571.  432.
## 6 arima013011 2085 May N(502, 1791)  502.  572.  433.
```

```
fc %>% filter(upper>=420) %>% head()
```

```
## # A fable: 6 x 6 [1M]
## # Key:     .model [1]
##   .model          index      value .mean upper lower
##   <chr>           <mth>     <dist> <dbl> <dbl> <dbl>
## 1 arima013011 2023 Apr N(407, 69)  407.  420.  393.
## 2 arima013011 2023 May N(407, 70)  407.  421.  393.
## 3 arima013011 2023 Jun N(406, 70)  406.  420.  393.
## 4 arima013011 2024 Feb N(406, 75)  406.  420.  392.
## 5 arima013011 2024 Mar N(407, 75)  407.  421.  392.
## 6 arima013011 2024 Apr N(408, 76)  408.  422.  394.
```

```
fc %>% filter(upper>=500) %>% head()
```

```
## # A fable: 6 x 6 [1M]
## # Key:     .model [1]
##   .model          index       value .mean upper lower
##   <chr>           <mth>      <dist> <dbl> <dbl> <dbl>
## 1 arima013011 2057 May N(459, 619)  459.  500.  418.
## 2 arima013011 2058 Mar N(459, 642)  459.  501.  417.
## 3 arima013011 2058 Apr N(460, 645)  460.  502.  418.
## 4 arima013011 2058 May N(461, 647)  461.  503.  419.
## 5 arima013011 2058 Jun N(460, 650)  460.  502.  418.
## 6 arima013011 2058 Jul N(459, 652)  459.  501.  417.
```

```
fc %>% filter(lower>=420) %>% head()
```

```
## # A fable: 6 x 6 [1M]
## # Key:     .model [1]
##   .model          index       value .mean upper lower
##   <chr>           <mth>      <dist> <dbl> <dbl> <dbl>
## 1 arima013011 2060 May N(464, 707)  464.  508.  420.
## 2 arima013011 2061 Apr N(465, 736)  465.  509.  420.
## 3 arima013011 2061 May N(465, 738)  465.  510.  421.
## 4 arima013011 2062 Apr N(466, 768)  466.  512.  421.
## 5 arima013011 2062 May N(467, 770)  467.  513.  421.
## 6 arima013011 2062 Jun N(466, 773)  466.  512.  421.
```

```
fc %>% filter(lower>=500) %>% head()
```

```
## # A fable: 0 x 6 [?]
## # Key:     .model [0]
## # i 6 variables: .model <chr>, index <mth>, value <dist>, .mean <dbl>,
## #   upper <dbl>, lower <dbl>
```

Report from the Point of View of the Present

One of the very interesting features of Keeling and colleagues' research is that they were able to evaluate, and re-evaluate the data as new series of measurements were released. This permitted the evaluation of previous models' performance and a much more difficult question: If their models'

predictions were "off" was this the result of a failure of the model, or a change in the system?

(1 point) Task 0b: Introduction

In this introduction, you can assume that your reader will have **just** read your 1997 report. In this introduction, **very** briefly pose the question that you are evaluating, and describe what (if anything) has changed in the data generating process between 1997 and the present.

(3 points) Task 1b: Create a modern data pipeline for Mona Loa CO2 data.

The most current data is provided by the United States' National Oceanic and Atmospheric Administration, on a data page [here]. Gather the most recent weekly data from this page. (A group that is interested in even more data management might choose to work with the hourly data.)

Create a data pipeline that starts by reading from the appropriate URL, and ends by saving an object called `co2_present` that is a suitable time series object.

Conduct the same EDA on this data. Describe how the Keeling Curve evolved from 1997 to the present, noting where the series seems to be following similar trends to the series that you "evaluated in 1997" and where the series seems to be following different trends. This EDA can use the same, or very similar tools and views as you provided in your 1997 report.

```
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
```

```
co2_present_raw=read.csv("https://gml.noaa.gov/webdata/ccgg/trends/co2/co2_weekly_mlo.csv",skip=51)
co2_present <- co2_present_raw %>%
  mutate(time_index=make_date(year,month,day)) %>%
  dplyr::select(time_index,average) %>%
  as_tsibble(index = time_index) %>%
```

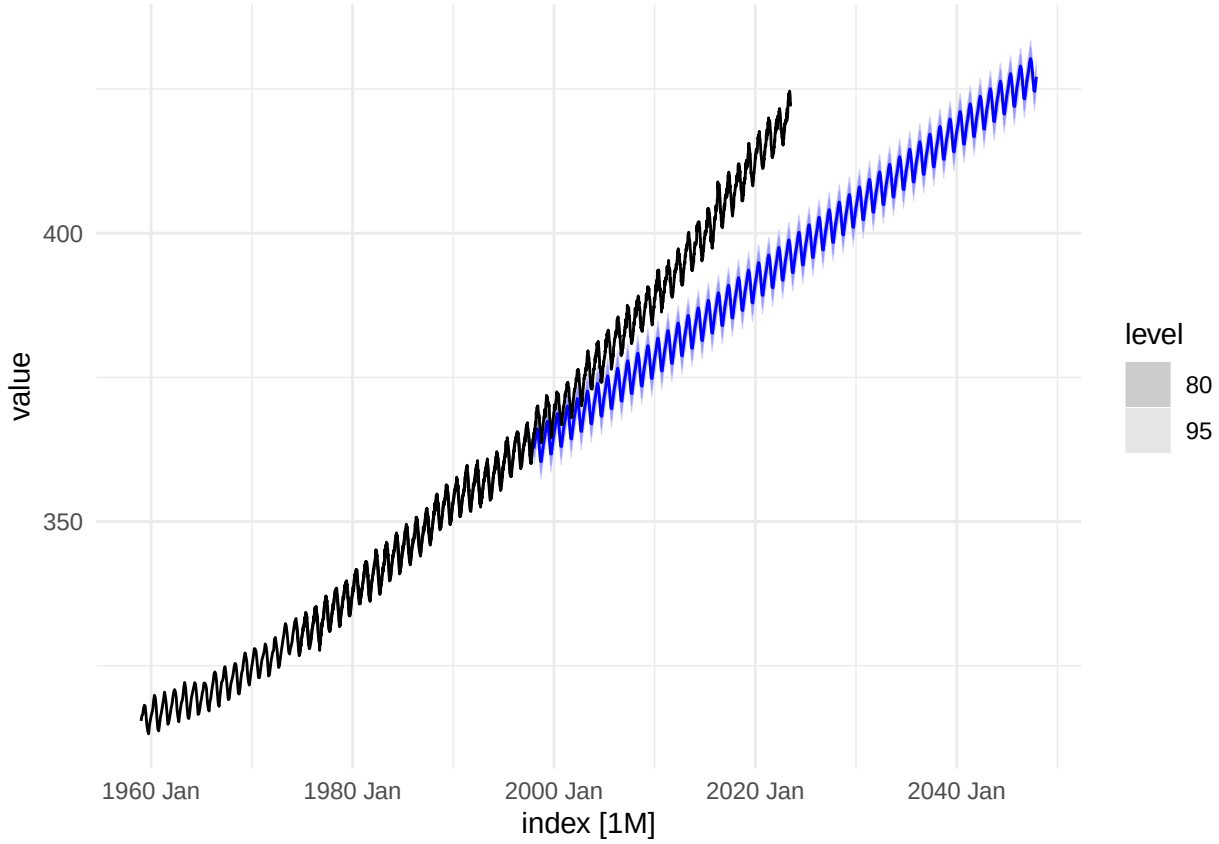
```
  mutate(average =replace(average,average<=-999,NA)) %>%
  mutate(average = na.approx(average))
```

(1 point) Task 2b: Compare linear model forecasts against realized CO2

Descriptively compare realized atmospheric CO2 levels to those predicted by your forecast from a linear time model in 1997 (i.e. "Task 2a"). (You do not need to run any formal tests for this task.)

```
co2 %>% autoplot(value) + autolayer(fc_linear) + autolayer(co2_present)
```

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

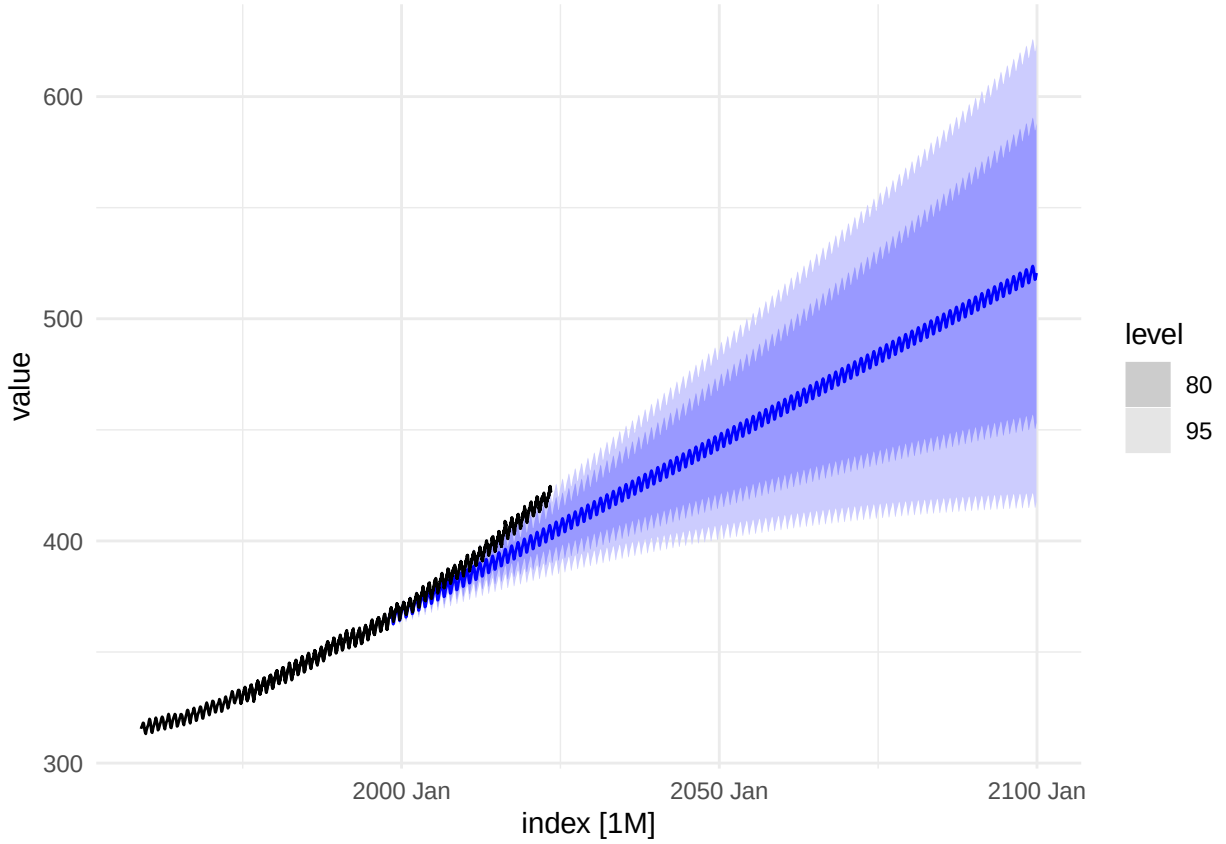


(1 point) Task 3b: Compare ARIMA models forecasts against realized CO2

Descriptively compare realized atmospheric CO2 levels to those predicted by your forecast from the ARIMA model that you fitted in 1997 (i.e. "Task 3a"). Describe how the Keeling Curve evolved from 1997 to the present.

```
co2 %>% autoplot(value) + autolayer(fc_arima) + autolayer(co2_present)
```

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



(3 points) Task 4b: Evaluate the performance of 1997 linear and ARIMA models

In 1997 you made predictions about the first time that CO2 would cross 420 ppm. How close were your models to the truth?

After reflecting on your performance on this threshold-prediction task, continue to use the weekly data to generate a month-average series from 1997 to the present, and compare the overall forecasting performance of your models from Parts 2a and 3b over the entire period. (You should conduct formal tests for this task.)

(4 points) Task 5b: Train best models on present data

Seasonally adjust the weekly NOAA data, and split both seasonally-adjusted (SA) and non-seasonally-adjusted (NSA) series into training and test sets, using the last two years of observations as the test sets. For both SA and NSA series, fit ARIMA models using all appropriate steps. Measure and discuss how your models perform in-sample and (psuedo-) out-of-sample, comparing candidate models and explaining your choice. In addition, fit a polynomial time-trend model to the seasonally-adjusted series and compare its performance to that of your ARIMA model.

```
# Scott mentioned STL for seasonal adjusment
```

(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?