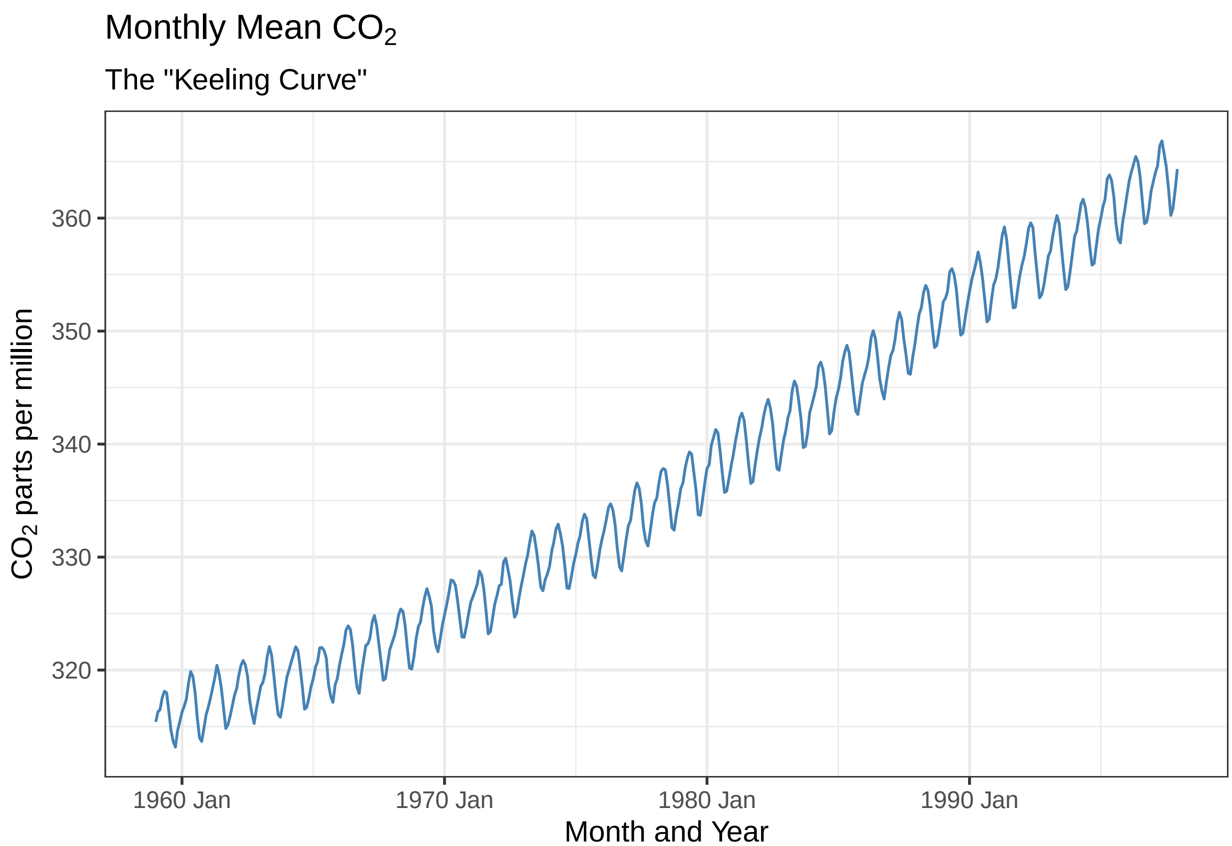
Statistical Methods for Discrete Response, Time Series, and Panel Data (W271): Lab 2

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."




```
## # A tsibble: 468 x 2 [1M]
##       index value
##       <mth> <dbl>
##  1 1959 Jan  315.
```

```
##   2 1959 Feb  316.
##   3 1959 Mar  316.
##   4 1959 Apr  318.
##   5 1959 May  318.
##   6 1959 Jun  318
##   7 1959 Jul  316.
##   8 1959 Aug  315.
##   9 1959 Sep  314.
##  10 1959 Oct  313.
## # i 458 more rows
```

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`.

(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.

Introduction

Co2 is classified as a "greenhouse gas," meaning that it traps heat in the atmosphere and lead to rising global temperatures when in high concentrations. It can be important to track Co2 levels as rising global temperatures can lead to imbalances in ecosystems and rising water levels that impact both animal and human life.

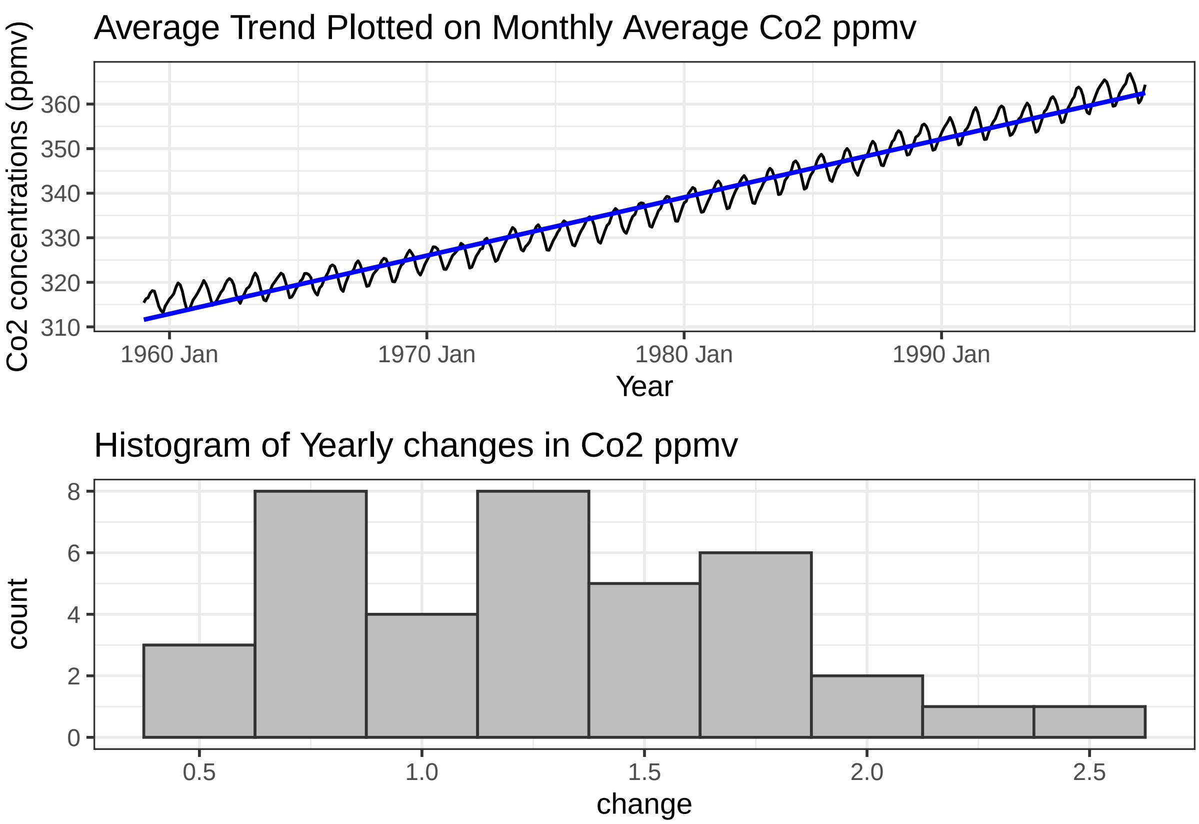
Description of data The current data is gathered from measurements made at the Mauna Loa Observatory in Hawaii (Cleaveland, 1993). Measurements were taken by a chemical gas analyzer sensor, with detections based on infrared absorption. This data measures monthly Co2 concentration levels from January 1959 to December 1997. Units are in parts per million of CO2 (abbreviated as ppmv) using the SIO manometric mole fraction scale. The principal investigator responsible for the initial findings of trending Co2 concentrations, Dr. Charles Keeling, initially designed a device to detect Co2 concentrations to detect Co2 emitted from limestone near bodies of water. But his measurements revealed a pattern of increasing Co2 concentrations at the global scale, urging further need to continue tracking the gas (Keeling, 1998).

Exploratory Data Analysis

The time series shows a clear upward trend of global Co2 concentrations from 1959 to 1998, with an average increase in 1.26 Co2 ppmv and a standard deviation of .51 Co2 ppmv. Upon inspection of the yearly increases, the bulk of changing Co2 levels are between 0.5 and 2.0 Co2 ppmv.

```r
# monthly time series with the line of best fit

co2_trend_plot <- co2_tsib %>%
  ggplot(aes(x = index, y = value)) +
  geom_line(color = 'black', size = .5) +
  geom_smooth(method = "lm", formula = "y ~ x",se = F, color = 'blue', size = .8) +
  labs(title = 'Average Trend Plotted on Monthly Average Co2 ppmv') +
  xlab('Year') +
  ylab('Co2 concentrations (ppmv)')



# average_yearly_increase
co2_tsib_yearly_change <- co2_tsib %>% as_tibble() %>%
  mutate(year = year(index)) %>%
  group_by(year) %>%
  summarise(`yearly_co2` = mean(value)) %>%
    ungroup() %>%
    mutate(lag_co2 = lag(yearly_co2),
           change = yearly_co2 - lag_co2,
           percent_change = ((yearly_co2 - lag_co2)/yearly_co2)*100)


# getting average increase (i.e. size of the trend)
```

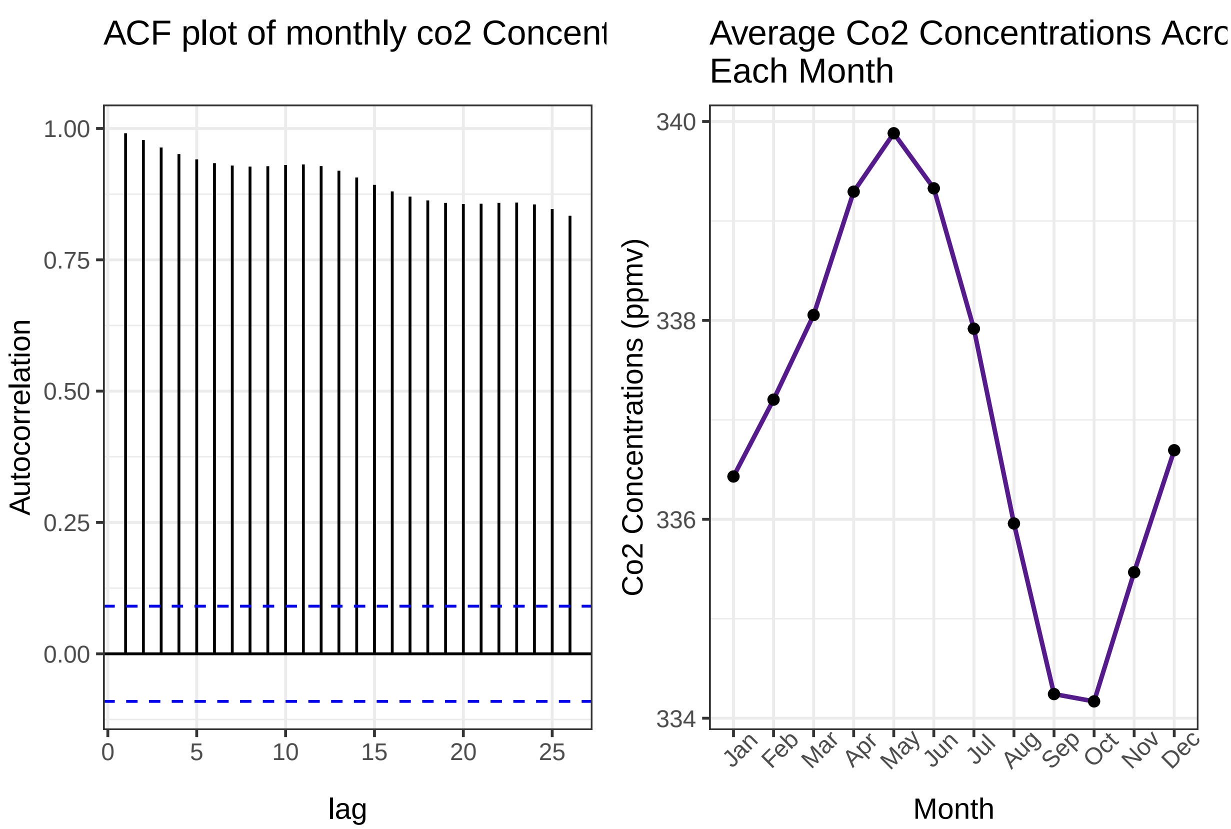
```r
yearly_mean <- mean(co2_tsib_yearly_change$change, na.rm = T) # average 1.26 units of co2 change each y
yearly_sd <- sd(co2_tsib_yearly_change$change, na.rm = T) # with a sd of .51 units of co2

change_hist <- ggplot(co2_tsib_yearly_change, aes(x = change)) +
  geom_histogram(color = 'gray20', fill = 'gray', binwidth = .25) +
  scale_x_continuous(breaks = seq(floor(min(co2_tsib_yearly_change$change, na.rm =T)),
                                  ceiling(max(co2_tsib_yearly_change$change, na.rm =T)), by = 0.5)) +
  ggtitle('Histogram of Yearly changes in Co2 ppmv')
# bulk of increases yearly seem to be between .5 and 2 units on the


co2_trend_plot / change_hist
```



The time series also shows strong evidence of seasonality corresponding closely with the meteorological seasons of Autumn, Winter, Spring, and Summer. Autumn and Winter are seen to have higher Co2 concentrations than in the Spring and Summer. This is likely due to the organic decomposition of plant life in these seasons (Keeling, 1960). The seasonality is evident in the consistent "peaks and valleys" in the above monthly time series plot. Evidence of seasonality is also found in the Autocorrelation Function Plot below, where a scallop/wave shaped pattern emerges among correlations between the current value with growing lags. Clearer evidence of seasonality is shown when inspecting the monthly average the Co2 ppmv, when averaged across all years in the available data.

```r
# inspecting acf and graph of co2 concentrations over time

co2_acf <- acf(co2_tsib$value, plot = F)
```

```r
co2_acf_plot <-  autoplot(co2_acf) +
  labs(title = "ACF plot of monthly co2 Concentrations", x = 'lag', y = 'Autocorrelation')


monthly_co2_ave_plot <- co2_tsib %>% as_tibble() %>%
  mutate(month = month(index)) %>%
  group_by(month) %>%
  summarise(co2_monthly_ave = mean(value, na.rm = T)) %>%
  mutate(month_str = factor(month.abb[month], levels = month.abb)) %>%
  ungroup() %>%
  ggplot(aes(x = month_str, y = co2_monthly_ave, group = 1)) +
  geom_line(size = .8, color = 'purple4') +
  geom_point(size = 1.5) +
  ggtitle("Average Co2 Concentrations Across\nEach Month") +
  xlab('Month') +
  ylab('Co2 Concentrations (ppmv)') +
  theme(axis.text.x = element_text(angle = 45))


co2_acf_plot | monthly_co2_ave_plot
```

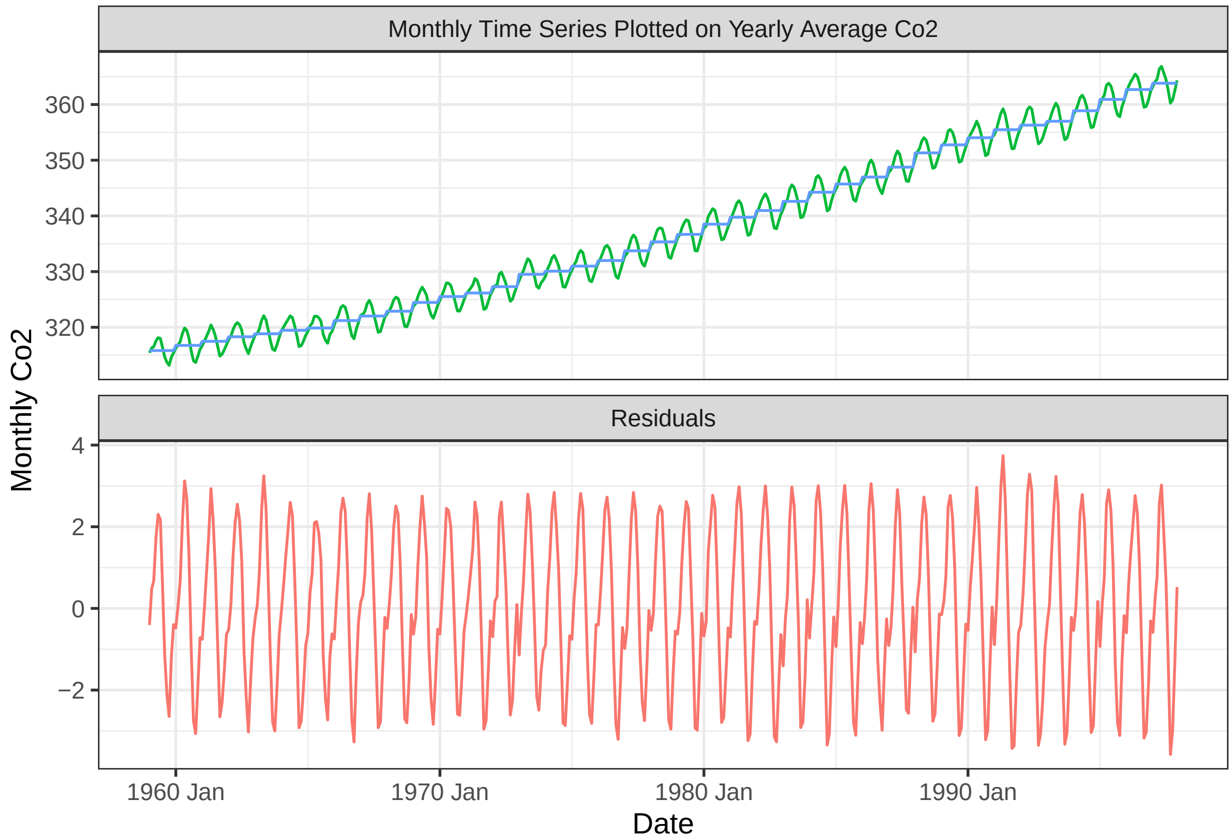


While the average Co2 ppmv is consistently higher each year, the variation across seasons are not exactly the same each year. Also the magnitude of how much the co2 concentrations increase each year is somewhat variable. This is evident when fitting a yearly Co2 average on the monthly time series, and inspecting the residuals from year to year. While slight, there seems to be shifting variance of residuals. These irregularities however, regress to a constant variance over time, according to a significant Augmented Dickey–Fuller Test, which suggests stationarity. In particular, this suggests to constant variance over time, as well as a non-moving average once accounting for the yearly increases in co2 ppmv.

```
# making a plot to show how the relationship looks like with yearly averages over the seasons
yearly_ave_w_residuals <- co2_tsib %>% as_tibble() %>%
  mutate(year = year(index)) %>%
  group_by(year) %>%
  mutate(`Yearly Co2` = mean(value)) %>%
  mutate(residual = value - `Yearly Co2`) %>%
  pivot_longer(cols = c(value, `Yearly Co2`, residual), names_to = "type", values_to = "Monthly Co2") %
  mutate(residual_bool = if_else(type == "residual", "Residuals", "Monthly Time Series Plotted on Yearl


yearly_ave_w_residuals_plot <- yearly_ave_w_residuals %>%
  ggplot(aes(x = index, y = `Monthly Co2`, color = type)) +
  geom_line() +
  facet_wrap(~residual_bool, scales = "free_y", ncol = 1) +
  xlab('Date') +
  theme(legend.position = "none")

# residuals of a simple yearly average look fairly stationary
  # with some years having larger and smaller co2 variances



adf_result <- yearly_ave_w_residuals %>%
  filter(residual_bool == 'Residuals') %>%
  ungroup() %>%
  pull(`Monthly Co2`) %>%
  adf.test()
```

```
## Warning in adf.test(.): p-value smaller than printed p-value
```

```
# Extract the relevant results into a data frame
adf_summary <- data.frame(
  Statistic = adf_result$statistic,
  P_Value = adf_result$p.value,
  Method = adf_result$method,
  Alternative = adf_result$alternative
)


yearly_ave_w_residuals_plot
```




```
kable(adf_summary, caption = "ADF Test Results")
```

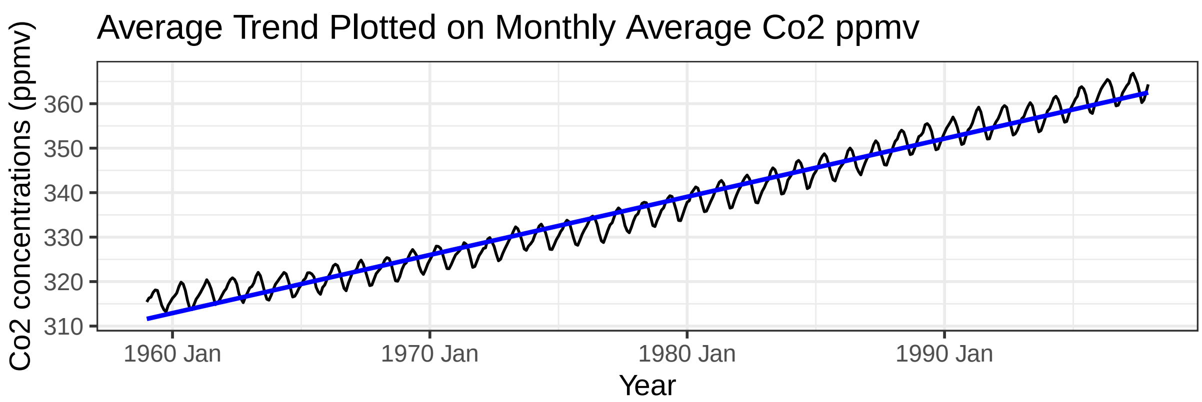
Table 1: ADF Test Results

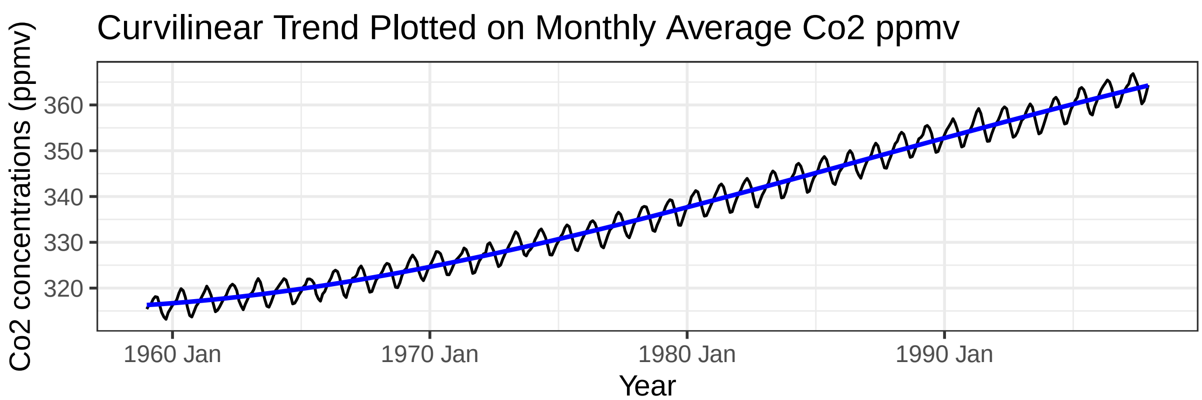
	Statistic	P_Value	Method	Alternative
Dickey-Fuller	-29.06892	0.01	Augmented Dickey-Fuller Test	stationary

Variability in the yearly trend is also observed through plotting. While the average change in Co2 ppmv is 1.26, this increase varies per year, with some years experiencing heavier spikes in increasing Co2 concentrations than others. Additionally, upon further inspection of the monthly average trend plot above, the fitted line appears to be systematically overestimating values at certain points and underestimating values at other points. When fitting a curvilinear trend to the time series, we see a closer fit the central Co2 ppmv for each year

```
# monthly time series with the curvilinear line of best fit

co2_curv_trend_plot <-
  co2_tsib %>%
  ggplot(aes(x = index, y = value)) +
  geom_line(color = 'black', size = .5) +
  geom_smooth(method = "lm", formula = "y ~ poly(x,3)",se = F, color = 'blue', size = .8) +
  labs(title = 'Curvilinear Trend Plotted on Monthly Average Co2 ppmv') +
  xlab('Year') +
  ylab('Co2 concentrations (ppmv)')
```

```
co2_trend_plot / co2_curv_trend_plot
```

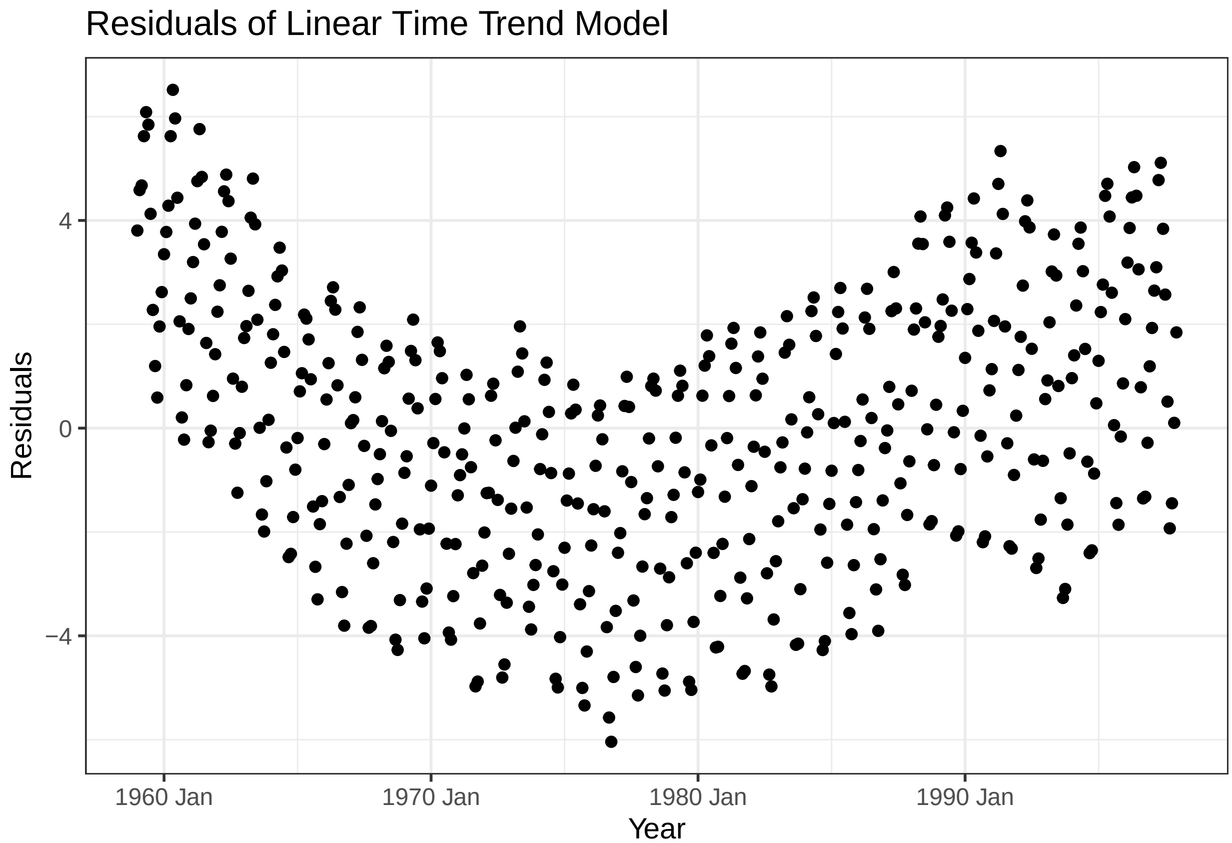




(3 points) Task 2a: Linear time trend model

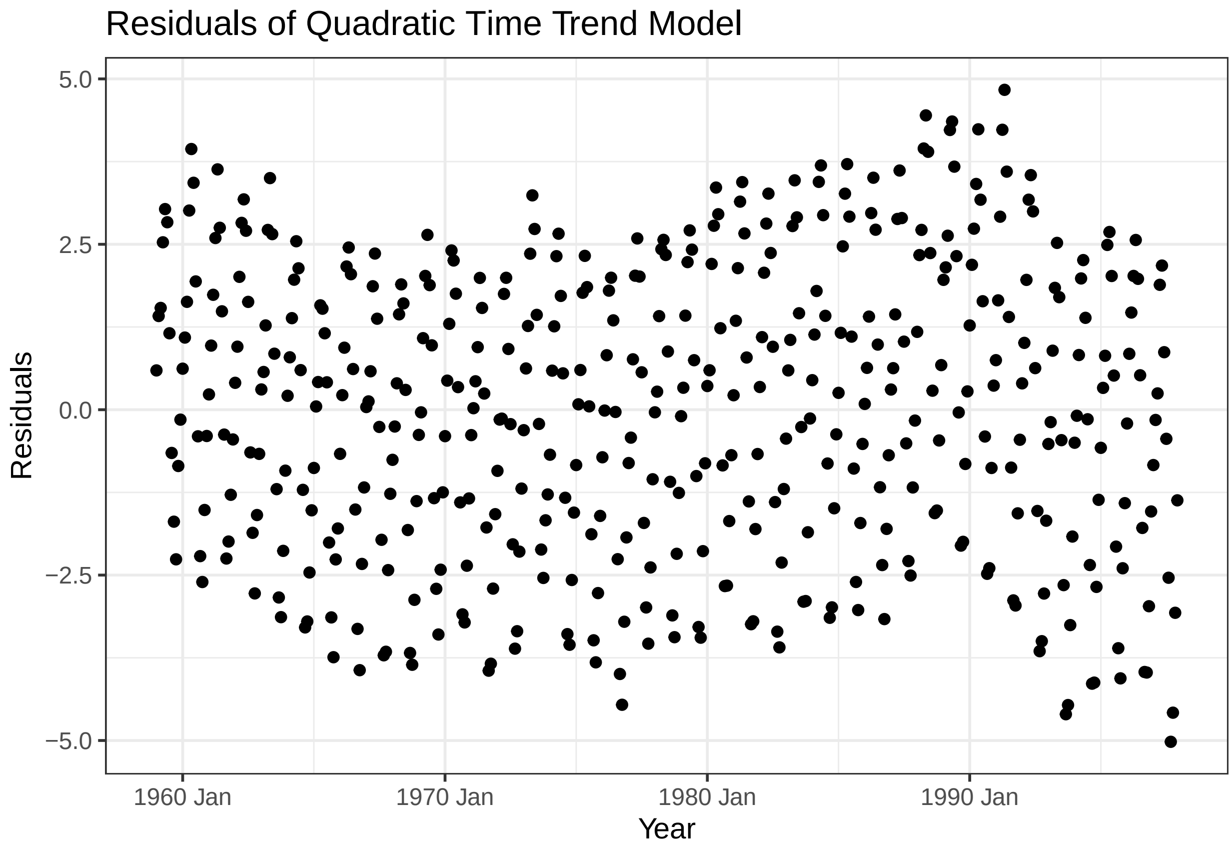
Fit a linear time trend model to the `co2` series, and examine the characteristics of the residuals. Compare this to a quadratic time trend model. Discuss whether a logarithmic transformation of the data would be appropriate. Fit a polynomial time trend model that incorporates seasonal dummy variables,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

```
co2_linear_model <- lm(value ~ index, data = co2_tsib)

# Plot residuals
co2_tsib$residuals_linear <- residuals(co2_linear_model)
lm_residuals_plot <- ggplot(co2_tsib, aes(x = index, y = residuals_linear)) +
  geom_point() +
  labs(title = "Residuals of Linear Time Trend Model", x = "Year", y = "Residuals")

lm_residuals_plot
```



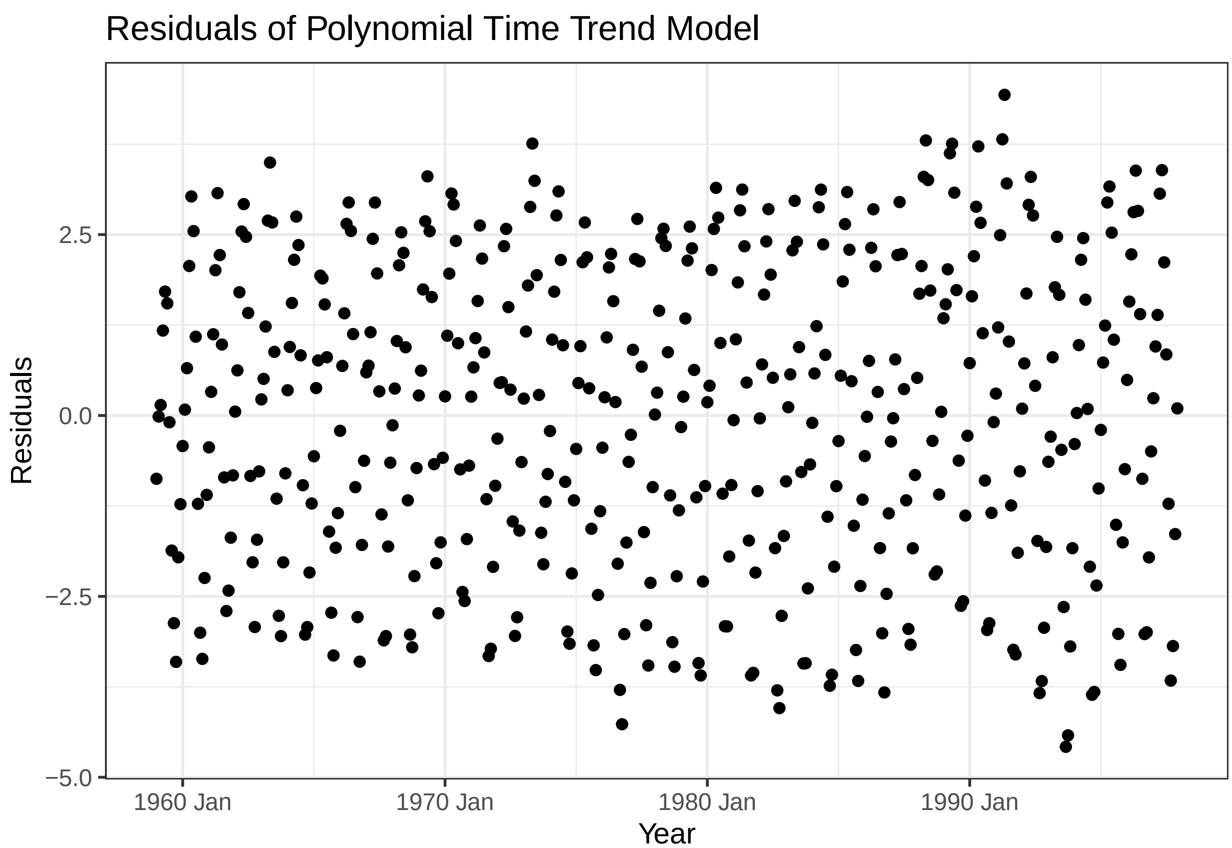
The residuals of the linear model exhibit a cyclical, non-linear pattern, indicating potential seasonality in the data. The visible fluctuations and lack of random distribution around zero suggest that the linear model is insufficient. A quadratic or polynomial model may provide a better fit for capturing the underlying structure.

```
co2_quad_model <- lm(value ~ poly(index, 2), data = co2_tsib)

# Plot residuals
co2_tsib$residuals_quad <- residuals(co2_quad_model)
quad_residuals_plot <- ggplot(co2_tsib, aes(x = index, y = residuals_quad)) +
  geom_point() +
  labs(title = "Residuals of Quadratic Time Trend Model", x = "Year", y = "Residuals")

quad_residuals_plot
```



The quadratic model's residuals indicate a significant reduction in variance, demonstrating an improved fit. While some cyclical behavior remains, it is less prominent, though seasonality persists in the data. Despite the better fit of the quadratic model, further insights may be gained by examining the residuals after fitting a polynomial model to the data.

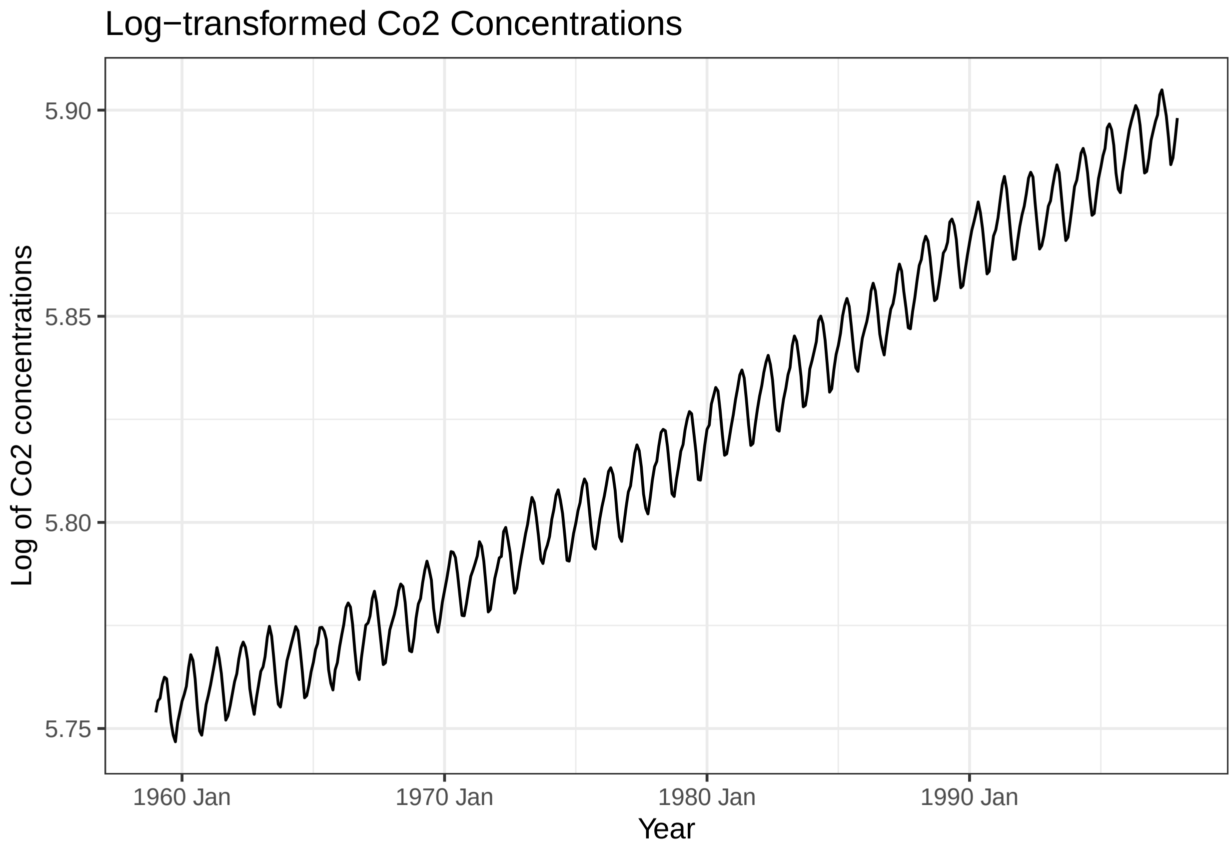
```
co2_poly_model <- lm(value ~ poly(index,3), data = co2_tsib)

# Plot residuals
co2_tsib$residuals_poly <- residuals(co2_poly_model)
poly_residuals_plot <- ggplot(co2_tsib, aes(x = index, y = residuals_poly)) +
  geom_point() +
  labs(title = "Residuals of Polynomial Time Trend Model", x = "Year", y = "Residuals")

poly_residuals_plot
```



The third-order polynomial model demonstrates improved residual behavior compared to quadratic and linear models. We chose to stop at this order to prevent excessive overfitting, as higher-order polynomials showed diminishing returns in model performance.

Apart from transforming the orders of the model, we were interested in data transformations - specifically logarithmic. As such we expermiented with a logarithmic dataset to observe the pattern of the data values.

```
# Log transformation
co2_tsib$log_value <- log(co2_tsib$value)

# Plot log-transformed data
log_data_plot <- ggplot(co2_tsib, aes(x = index, y = log_value)) +
  geom_line() +
  labs(title = "Log-transformed Co2 Concentrations", x = "Year", y = "Log of Co2 concentrations")

log_data_plot
```

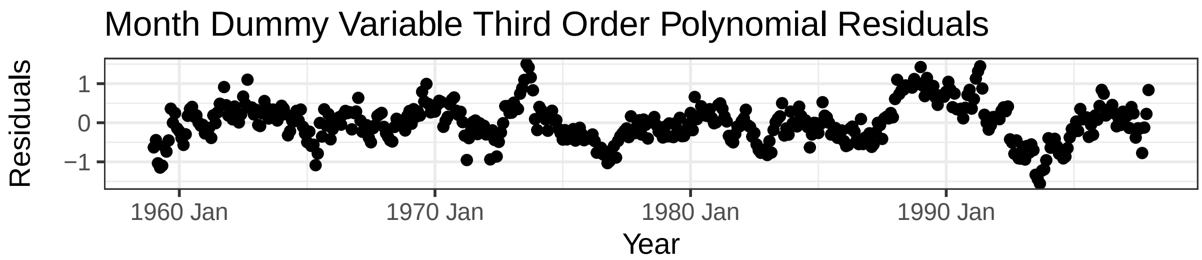


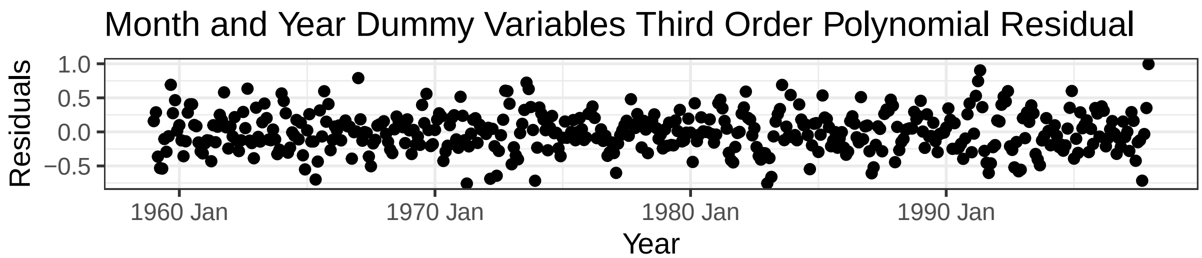
The logarithmic transformation reduces variance but offers minimal improvement compared to traditional plotting. This limited impact is likely due to the cyclical nature of the time series, which the transformation does not adequately address.

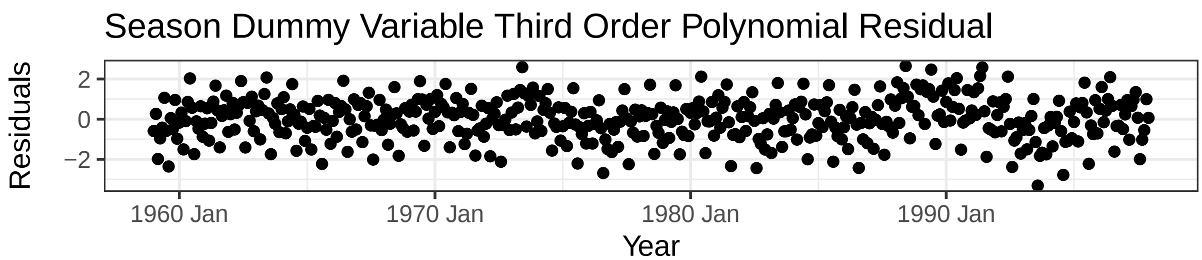
To address the cyclical behavior, we developed another polynomial model that includes a `month` variable. The average monthly CO2 emissions indicate significant cyclic patterns at the monthly level. By incorporating this variable, we anticipate an improvement in the fit of our time series model.

```
# Ensure index is in Date format
co2_tsib$index_date <- as.Date(co2_tsib$index)

# Create seasonal dummy variables
co2_tsib$month <- factor(month(co2_tsib$index))
co2_tsib$year <- factor(year(co2_tsib$index))

# Define a function to convert months into seasons
co2_tsib$season <- case_when(
  month(co2_tsib$index) %in% c(12, 1, 2) ~ "Winter",
  month(co2_tsib$index) %in% c(3, 4, 5) ~ "Spring",
  month(co2_tsib$index) %in% c(6, 7, 8) ~ "Summer",
  month(co2_tsib$index) %in% c(9, 10, 11) ~ "Autumn"
)

# Convert season into a factor
co2_tsib$season <- factor(co2_tsib$season, levels = c("Winter", "Spring", "Summer", "Autumn"))

# Fit polynomial model with seasonal dummies
```

```r
poly_month <- lm(value ~ poly(index, 3) + month, data = co2_tsib)
poly_month_year <- lm(value ~ poly(index, 3) + month + year, data = co2_tsib)
poly_season <- lm(value ~ poly(index, 3) + season, data = co2_tsib)

# Get residuals
co2_tsib$residuals_poly_month <- residuals(poly_month)
co2_tsib$residuals_poly_month_year <- residuals(poly_month_year)
co2_tsib$residuals_poly_season <- residuals(poly_season)

# residual plots
poly_month_plot <- ggplot(co2_tsib, aes(x = index, y = residuals_poly_month)) +
  geom_point() +
  labs(title = "Month Dummy Variable Third Order Polynomial Residuals",
       # subtitle = "Model includes month as a categorical variable",
       x = "Year",
       y = "Residuals")

poly_month_year_plot <- ggplot(co2_tsib, aes(x = index, y = residuals_poly_month_year)) +
  geom_point() +
  labs(title = "Month and Year Dummy Variables Third Order Polynomial Residual",
       # subtitle = "Model includes both month and year as categorical variables",
       x = "Year",
       y = "Residuals")

poly_season_plot <- ggplot(co2_tsib, aes(x = index, y = residuals_poly_season)) +
  geom_point() +
  labs(title = "Season Dummy Variable Third Order Polynomial Residual",
       # subtitle = "Model includes season as a categorical variable (Winter, Spring, Summer, Autumn)",
       x = "Year",
       y = "Residuals")

# Combine the three plots side by side
residual_combined_plot <- poly_month_plot + poly_month_year_plot + poly_season_plot +
  plot_layout(nrow = 3) # Set layout with 3 columns

# Display the combined plot
print(residual_combined_plot)
```





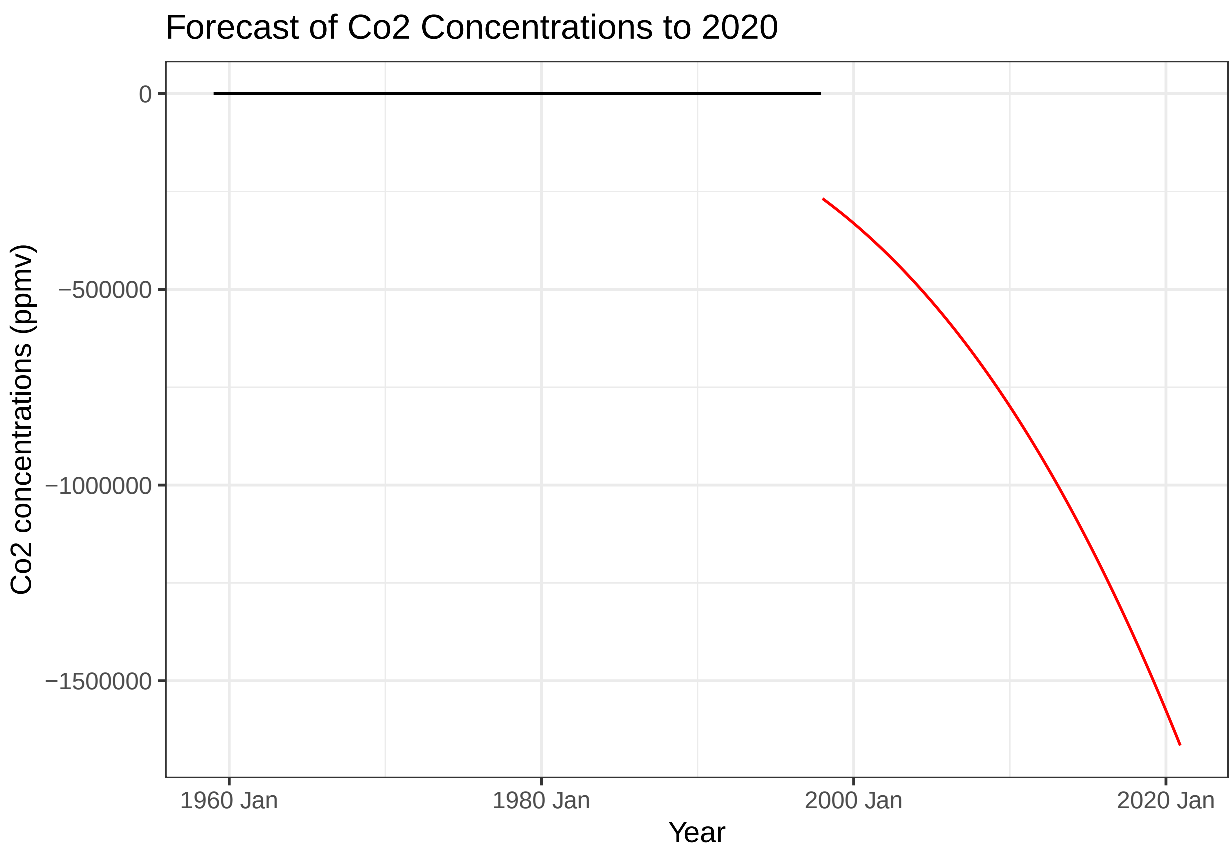


Incorporating the `month` dummy variable brought the residuals closer to zero, ranging between 1 and -1, but they still displayed a non-random pattern. To refine the model, we added a `year` dummy variable, which improved the residual fit, though it was omitted due to multicollinearity concerns. Finally, we introduced a season categorical variable, grouping the months into quarters. This adjustment centered the residuals around zero with a random distribution, though fluctuations remained between 2 and -2. We proceeded with this model, considering it the most robust.

Using the polynomial model with the `season` dummy variable, we then developed a forecast for CO2 emissions through 2020.

```r
# Generate future time points (e.g., monthly until 2020)
future_years <- seq(from = max(co2_tsib$index_date) + months(1), to = as.Date("2020-12-01"), by = "mont
future_data <- data.frame(index = future_years, month = factor(month(future_years)))

# # Create seasonal dummy variables
# future_data$month <- factor(month(future_data$index))
# future_data$year <- factor(year(future_data$index))

# Get the season
future_data$season <- case_when(
  month(future_data$index) %in% c(12, 1, 2) ~ "Winter",
  month(future_data$index) %in% c(3, 4, 5) ~ "Spring",
  month(future_data$index) %in% c(6, 7, 8) ~ "Summer",
  month(future_data$index) %in% c(9, 10, 11) ~ "Autumn"
)

# Predict using the model
```

```r
future_data$forecast <- predict(poly_season, newdata = future_data)

# Plot the forecasts
forecast_plot <- ggplot() +
  geom_line(data = co2_tsib, aes(x = index, y = value), color = "black") +
  geom_line(data = future_data, aes(x = index, y = forecast), color = "red") +
  labs(title = "Forecast of Co2 Concentrations to 2020", x = "Year", y = "Co2 concentrations (ppmv)")

forecast_plot
```



The forecast model using the `season` variable shows very poor performance, with future predictions deviating significantly from expected values, likely due to a misfitting model or improper scaling of the forecasted data. To resolve this, we turn to an ARIMA model, which may better capture the time series' underlying patterns and improve forecast accuracy.

(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.

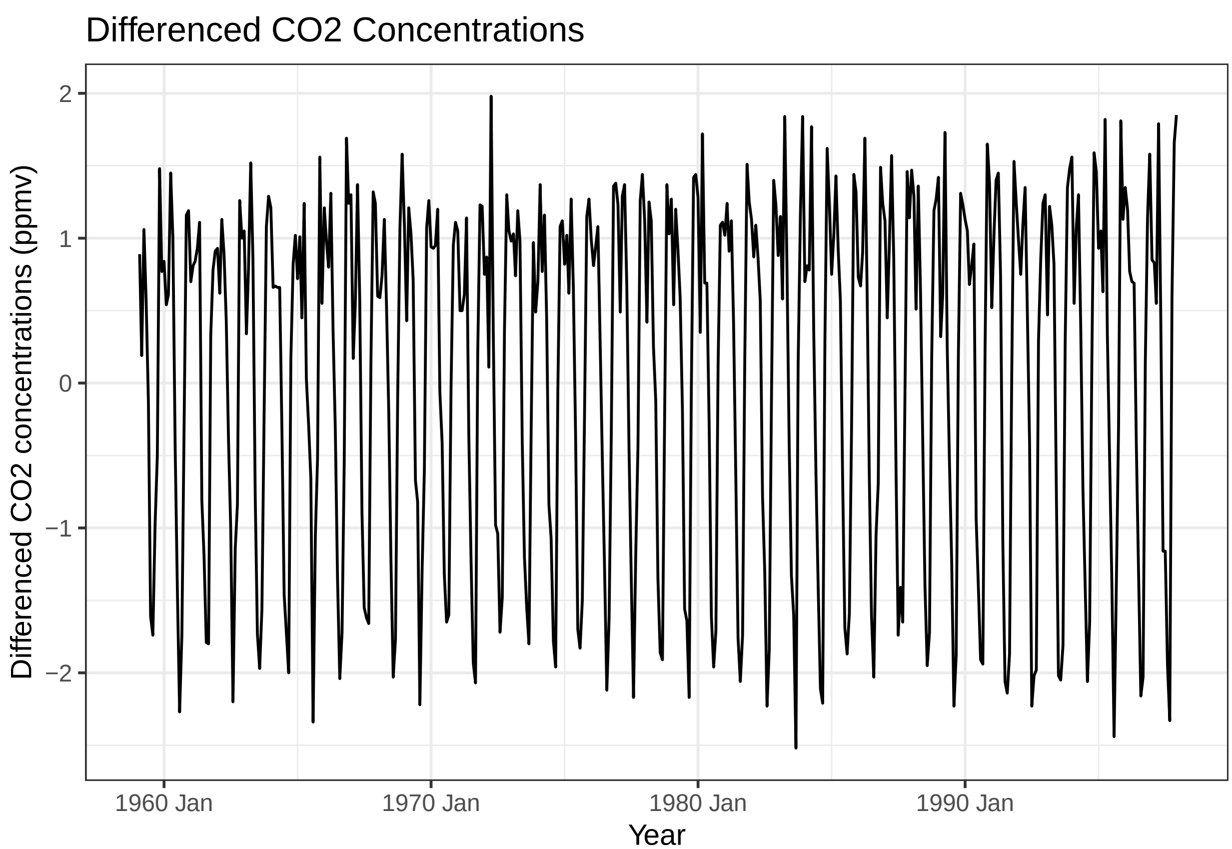
```r
# checking for stationarity
adf.test(co2_tsib$value)
```

```
##
```

```
##  Augmented Dickey-Fuller Test
##
## data:  co2_tsib$value
## Dickey-Fuller = -2.8299, Lag order = 7, p-value = 0.2269
## alternative hypothesis: stationary
```

With a p-value of 0.2269, the time series data fails to reject the null hypothesis of non-stationarity. As a result, we will proceed with differencing the data to make it stationary, which is a crucial step before fitting the ARIMA model effectively.

```
# First differencing the series to remove trend
# Added NA to ensure the data has equivalent numbers of rows.
co2_tsib$diff_value <- c(NA,diff(co2_tsib$value, differences = 1))

# Plot the differenced series to check if it looks stationary
ggplot(co2_tsib, aes(x = index, y = diff_value)) +
  geom_line() +
  labs(title = "Differenced CO2 Concentrations", x = "Year", y = "Differenced CO2 concentrations (ppmv)
```




```
# post-check for stationarity
# remove NA for adf test
adf.test(na.omit(co2_tsib$diff_value))
```

```
##
##  Augmented Dickey-Fuller Test
```

```
##
## data:  na.omit(co2_tsib$diff_value)
## Dickey-Fuller = -30.38, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

With first differencing successfully making the data stationary, we can now proceed with constructing the ARIMA (p,d,q) model using a d value of 1. The next step will involve fine-tuning the p and q parameters to further optimize the model for accurate forecasting.

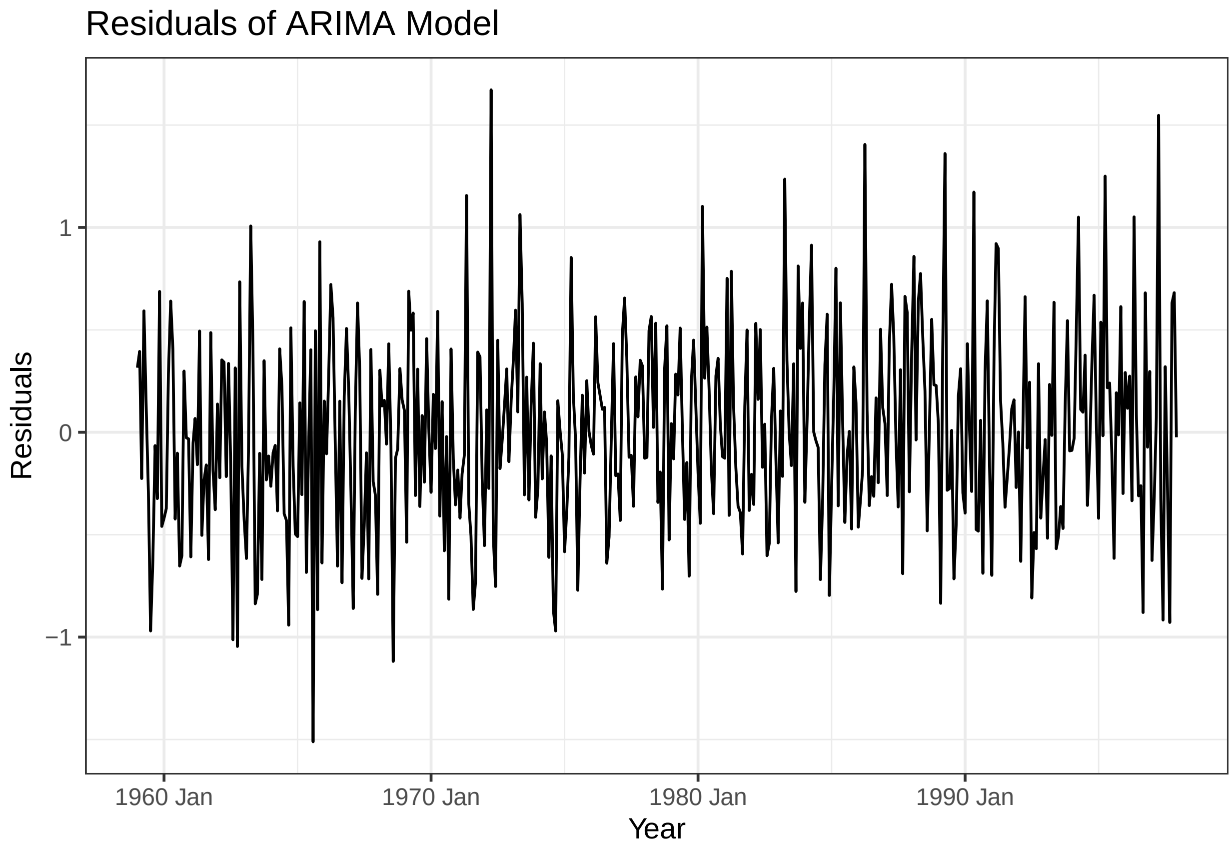
```r
# Fit ARIMA model by testing different lags using the AIC criterion
model.aic <- co2_tsib %>%
  model(ARIMA(value ~ 1 + pdq(0:10, 1, 0:10) + PDQ(0:2, 0, 0:2),
              ic = "aic", stepwise = FALSE))

# Report the best model
model.aic %>%
  report()
```

```
## Series: value
## Model: ARIMA(3,1,1)(0,0,2)[12] w/ drift
##
## Coefficients:
##          ar1      ar2      ar3      ma1    sma1    sma2  constant
##       1.1159  -0.1776  -0.3450  -0.9252  0.6573  0.3792    0.0427
## s.e.  0.0497   0.0738   0.0456   0.0167  0.0506  0.0417    0.0034
##
## sigma^2 estimated as 0.2311:  log likelihood=-321.64
## AIC=659.29   AICc=659.6   BIC=692.46
```

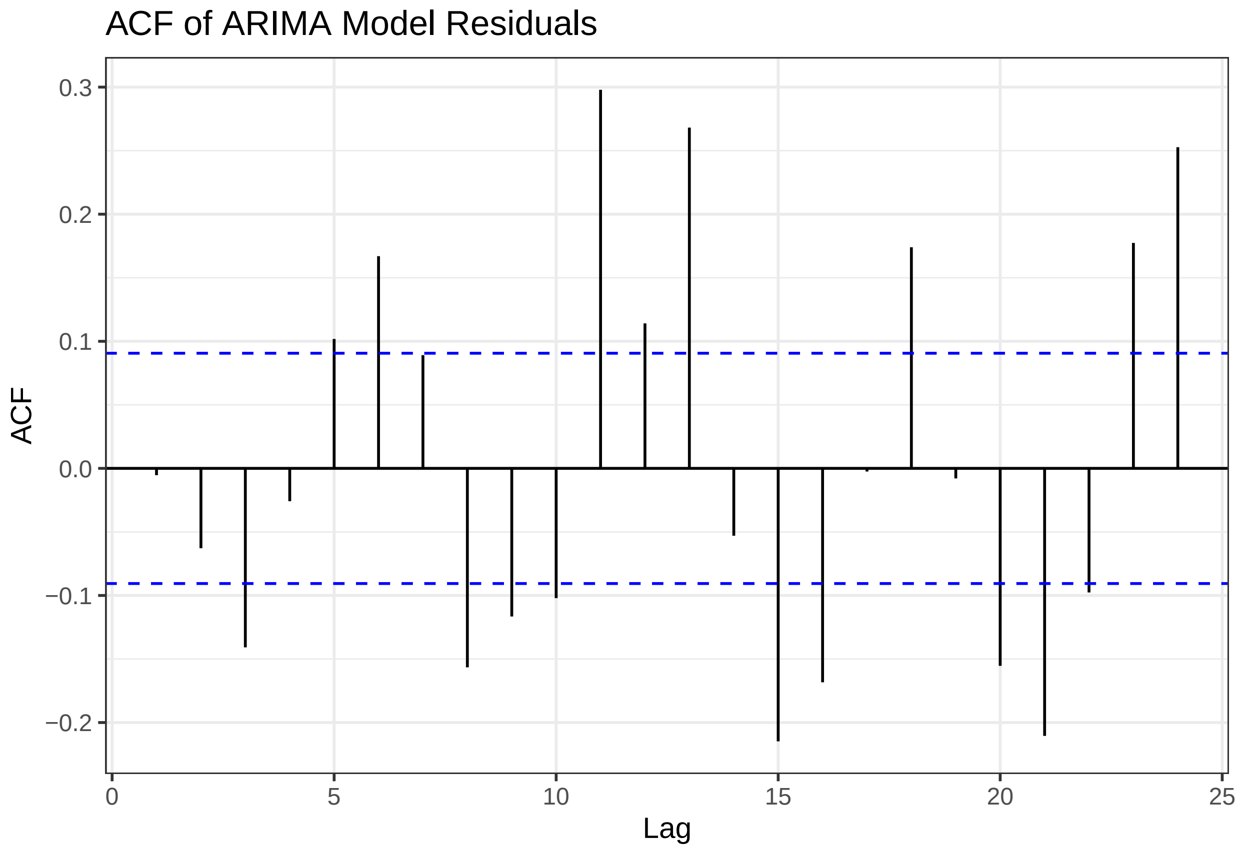
```r
# Extract residuals from the ARIMA model
residuals_arima <- residuals(model.aic)

# Plot the residuals over time
residual_plot <- autoplot(residuals_arima) +
  labs(title = "Residuals of ARIMA Model", x = "Year", y = "Residuals")
```

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

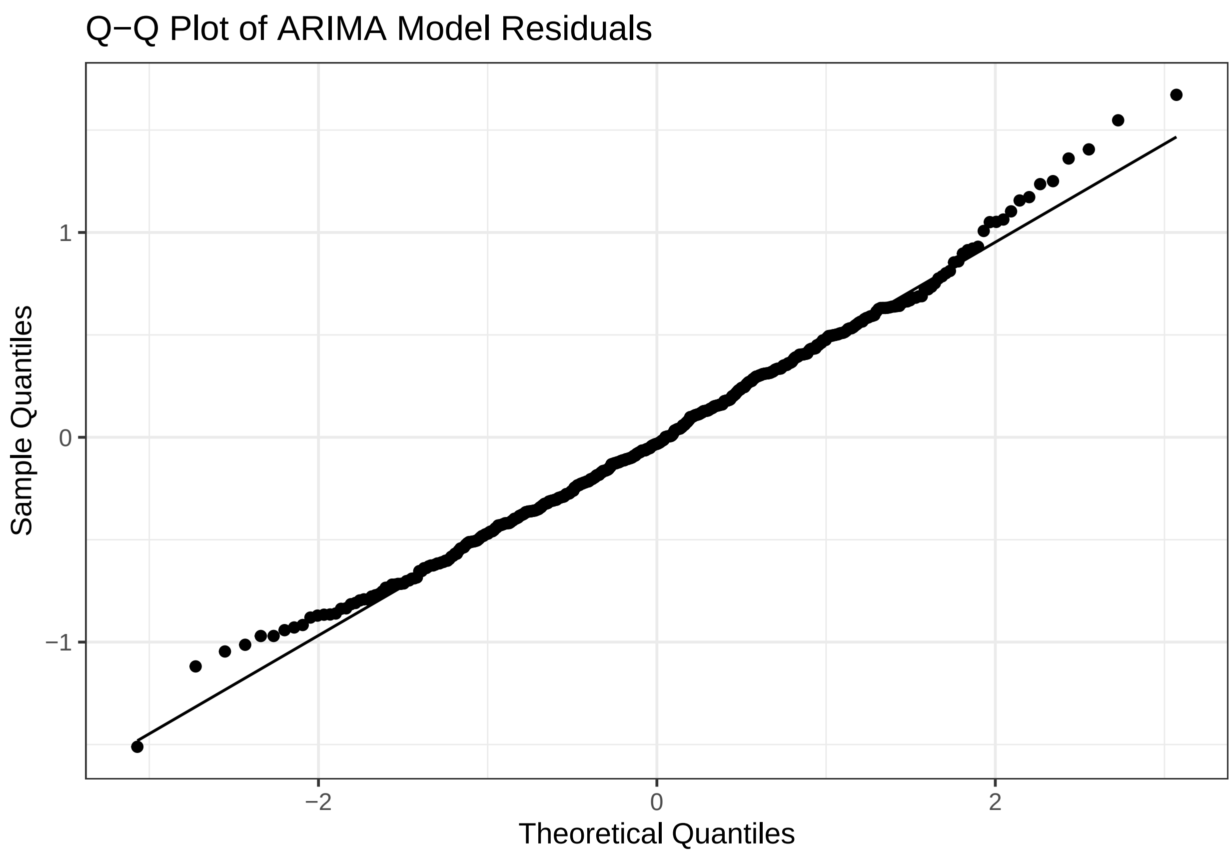
```r
# Display the plot
residual_plot
```




```r
# Plot the ACF of the residuals
acf_plot <- ggAcf(residuals_arima) +
  labs(title = "ACF of ARIMA Model Residuals", x = "Lag", y = "ACF")

# Display the ACF plot
print(acf_plot)
```




```
# Q-Q plot to check normality of residuals
qq_plot <- ggplot(data = as.data.frame(residuals_arima), aes(sample = .resid)) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "Q-Q Plot of ARIMA Model Residuals", x = "Theoretical Quantiles", y = "Sample Quantiles")

# Display the Q-Q plot
print(qq_plot)
```




```r
# Ljung Box Test on residuals
# Calculate the number of observations
N <- nrow(co2_tsib)

# Determine the number of lags (using rule of thumb: sqrt(N))
lags <- floor(sqrt(N))

# get residuals
resid.ts<-model.aic %>%
  augment() %>%
  select(.resid) %>%
  as.ts()

Box.test(resid.ts, lag = lags, type = "Ljung-Box")
```

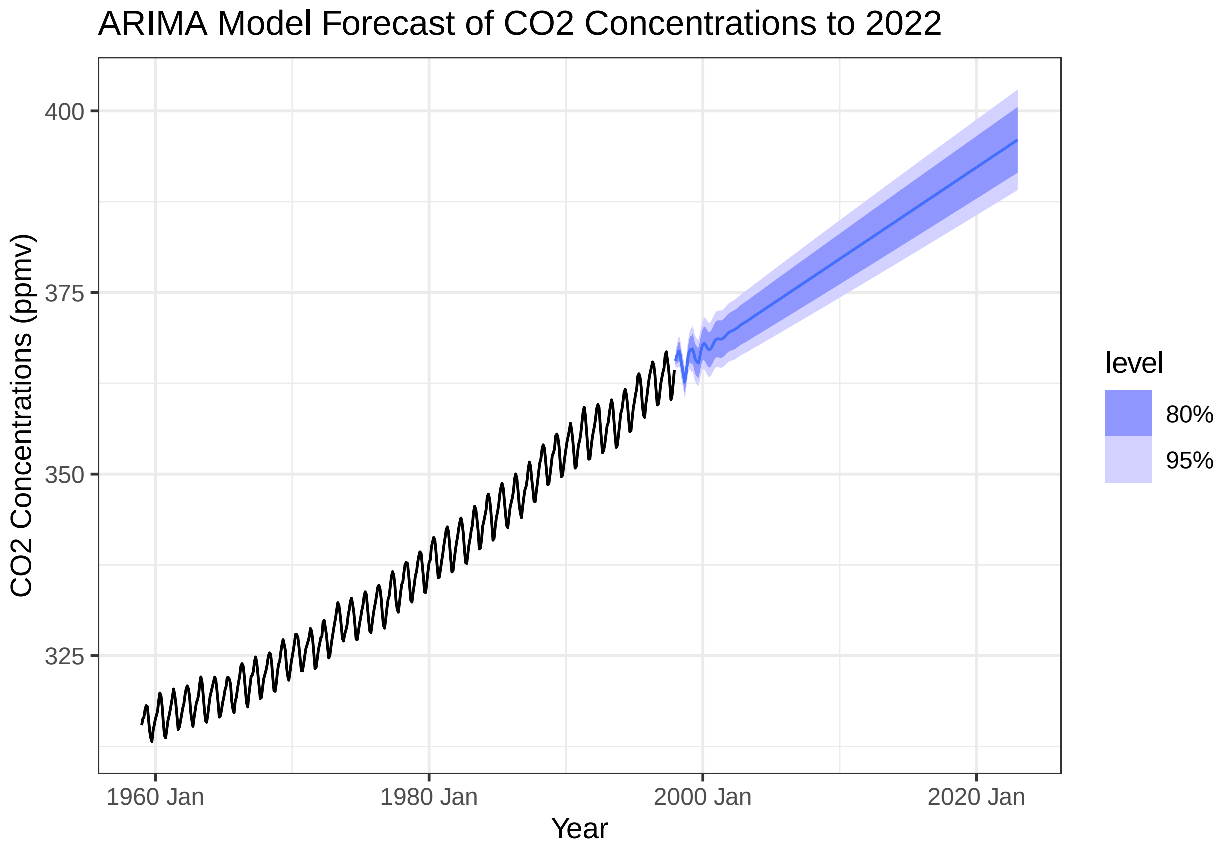
```
##
##  Box-Ljung test
##
## data:  resid.ts
## X-squared = 226.59, df = 21, p-value < 2.2e-16
```

```r
# # Fit ARIMA model by testing different lags using the AICc criterion
# model.aicc <- co2_tsib %>%
#   model(ARIMA(value ~ 1 + pdq(0:10, 1, 0:10) + PDQ(0:2, 0, 0:2),
#               ic = "aicc", stepwise = FALSE))
```

```r
#
# # Report the best model
# model.aicc %>%
#   report()
#
# # Fit ARIMA model by testing different lags using the BIC criterion
# model.bic <- co2_tsib %>%
#   model(ARIMA(value ~ 1 + pdq(0:10, 1, 0:10) + PDQ(0:2, 0, 0:2),
#               ic = "bic", stepwise = FALSE))
#
# # Report the best model
# model.bic %>%
#   report()
```

The AIC analysis identified the optimal ARIMA model parameters as p=3, d=1, and q=1. The residuals are centered around zero with random fluctuations, indicating the model has captured most of the underlying structure. However, the ACF plot shows significant spikes at lags 10 and 24, indicating some remaining autocorrelation, and the Ljung-Box test confirms this with a p-value of $< 2.2e\text{-}16$, suggesting the need for additional terms to improve the model fit. Despite this, the QQ plot indicates that the residuals follow normality well. With these considerations, we can proceed with forecasting the time series data through 2022.

```r
# Determine how many months to forecast (from the last observation to Dec 2022)
last_observation <- max(co2_tsib$index)
end_of_2022 <- as.Date("2022-12-31")

# Calculate the number of months to forecast
horizon <- as.numeric(
  difftime(end_of_2022, last_observation, units = "weeks")) / 4.34524  # Convert weeks to months

# Forecast using the ARIMA model until 2022
forecast_arima <- model.aic %>%
  forecast(h = round(horizon))  # Use calculated horizon

# Plot the forecast
forecast_plot <- forecast_arima %>%
  autoplot(co2_tsib) +
  labs(title = "ARIMA Model Forecast of CO2 Concentrations to 2022",
       x = "Year", y = "CO2 Concentrations (ppmv)")

# Show the plot
print(forecast_plot)
```



(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?

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.

(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.)

(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.

(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.

(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?