

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

Due Monday July 12 2021 11:59pm

Instructions (Please Read Carefully):

- Submit by the due date. **Late submissions will not be accepted**
- No page limit, but be reasonable
- Do not modify fontsize, margin or line-spacing settings
- One student from each group should submit the lab to their student github repo by the deadline
- Submit two files:

1. A pdf file that details your answers. Include all R code used to produce the answers
2. The R markdown (Rmd) file used to produce the pdf file

The assignment will not be graded unless **both** files are submitted

- Name your files to include all group members names. For example, if the students' names are Stan Cartman and Kenny Kyle, name your files as follows:

- StanCartman_KennyKyle_Lab2.Rmd
- StanCartman_KennyKyle_Lab2.pdf

- Although it sounds obvious, please write your name on page 1 of your pdf and Rmd files
- All answers should include a detailed narrative; make sure that your audience can easily follow the logic of your analysis. All steps used in modelling must be clearly shown and explained; do not simply 'output dump' the results of code without explanation
- If you use libraries and functions for statistical modeling that we have not covered in this course, you must provide an explanation of why such libraries and functions are used and reference the library documentation
- For mathematical formulae, type them in your R markdown file. Do not e.g. write them on a piece of paper, snap a photo, and use the image file
- Incorrectly following submission instructions results in deduction of grades
- Students are expected to act with regard to UC Berkeley Academic Integrity.

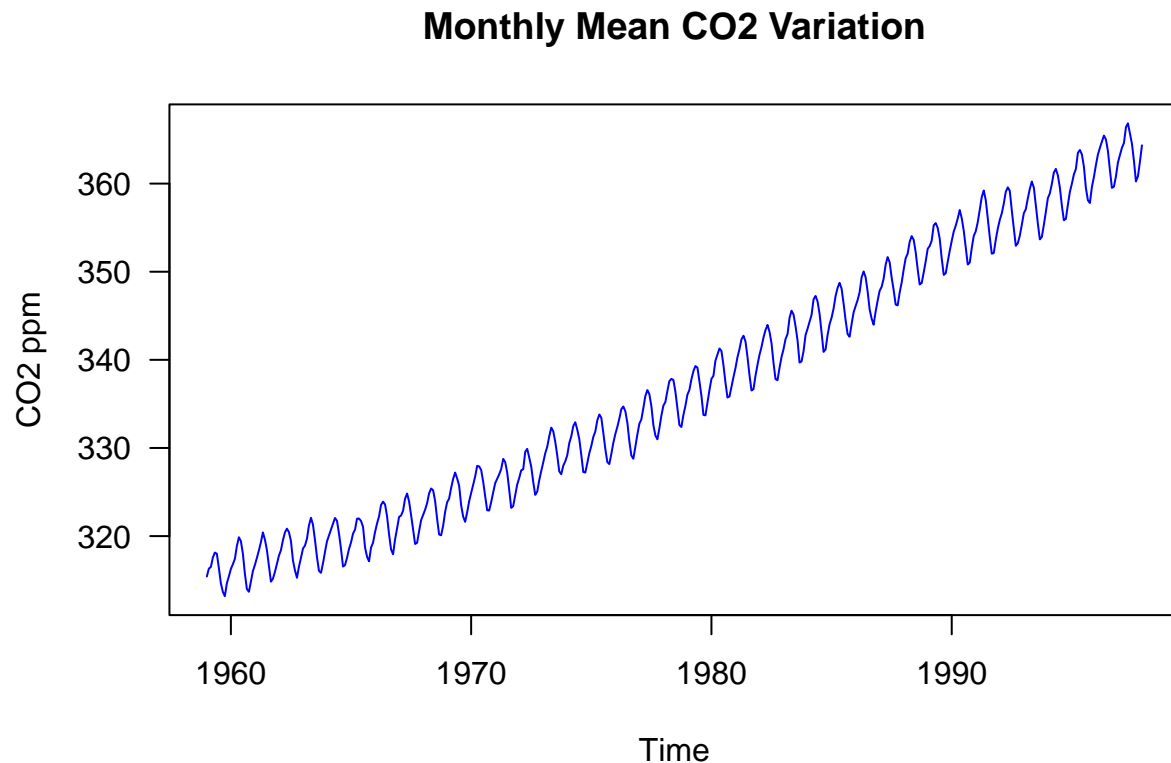
The Keeling Curve

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 attributed this pattern to varying rates of photosynthesis throughout the year, caused by differences 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. He soon observed a trend increase carbon dioxide levels in addition to the seasonal cycle, attributable to growth in global rates of fossil fuel combustion. Measurement of this trend at Mauna Loa has continued to the present.

The `co2` data set in R's `datasets` package (automatically loaded with base R) is a monthly time series of atmospheric carbon dioxide concentrations measured in ppm (parts per million) at the Mauna Loa Observatory from 1959 to 1997. The curve graphed by this data is known as the 'Keeling Curve'.

```
plot(co2, ylab = expression("CO2 ppm"), col = 'blue', las = 1)
title(main = "Monthly Mean CO2 Variation")
```



Part 1 (3 points)

Conduct a comprehensive Exploratory Data Analysis on the `co2` series. This should include (without being limited to) a thorough investigation of the trend, seasonal and irregular elements.

Response: We started off the EDA with looking at the general statistics of the data. The looks like there is a value every month start in 1959, there are zero missing values, and the frequency of the data is 12, which makes sense since there are 12 months in a year. The mean of the data set is 337.1 and the range is from 313.2 to 366.8. The histogram of the data shows that the data is generally equal throughout the range, the data is not skewed in any direction. From the boxplot graph, we can see that mean of the data seems to decrease in the second half of the year, showing there might be a bit of a seasonal trend in the data.

The decomposition plot shows a strong trend in the data as well as a strong seasonality trend, which showed to us that we might think about using a seasonal model later on and that the data might need to be difference in order to be made stationary. Lastly, from the ACF, the data does not quickly go to zero and therefore this may mean that the data is not stationary and we may need to difference the data, and the PACF graph shows two spikes in the beginning and another two spikes 12 months in, which makes us think that a model will need not only a moving average term of 1 or 2 but also a seasonal moving average term of 1 or 2.

Overall, the data looks very clean with very strong trend and seasonality impacts which will need to be correctly modeled to obtain good time series predictions.

```
head(co2)
```

```
##           Jan      Feb      Mar      Apr      May      Jun
## 1959 315.42 316.31 316.50 317.56 318.13 318.00
```

```
sum(is.na(co2))
```

```
## [1] 0
```

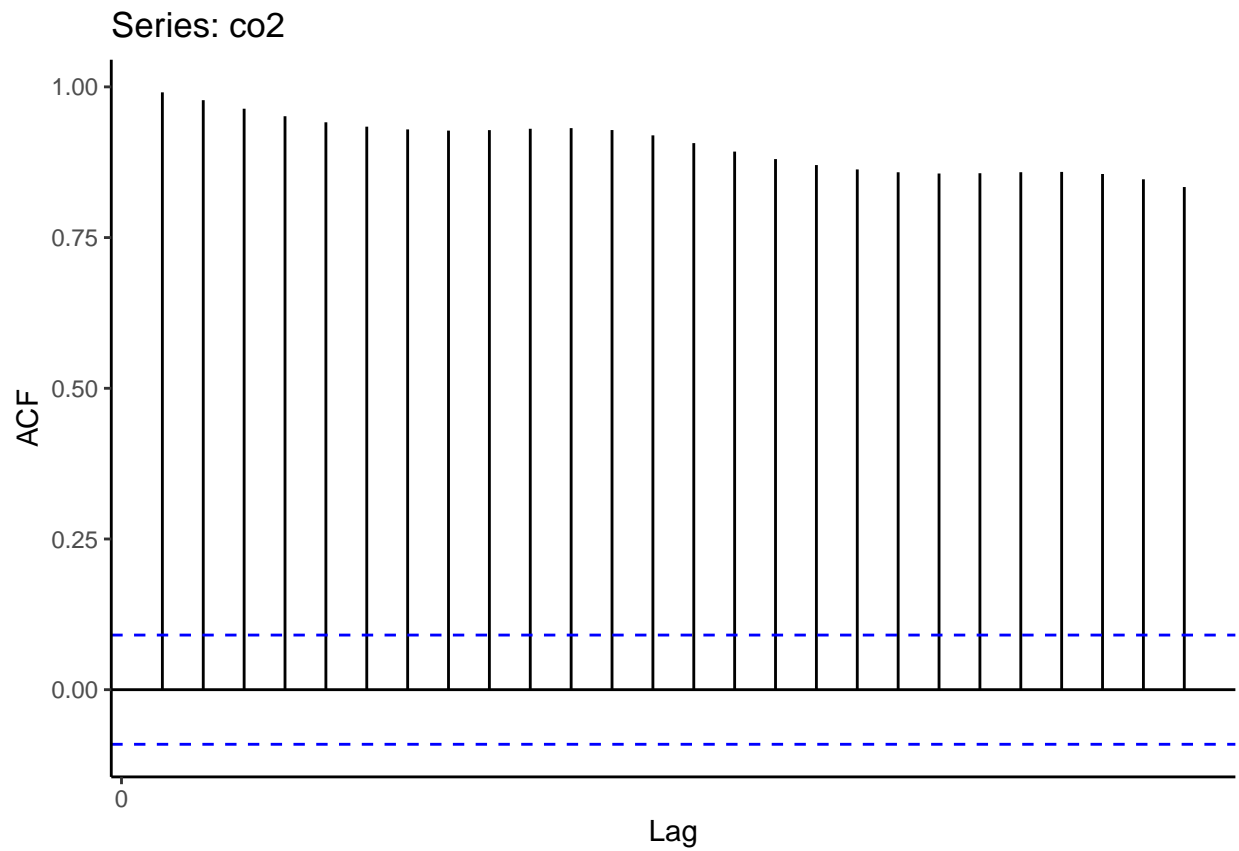
```
frequency(co2)
```

```
## [1] 12
```

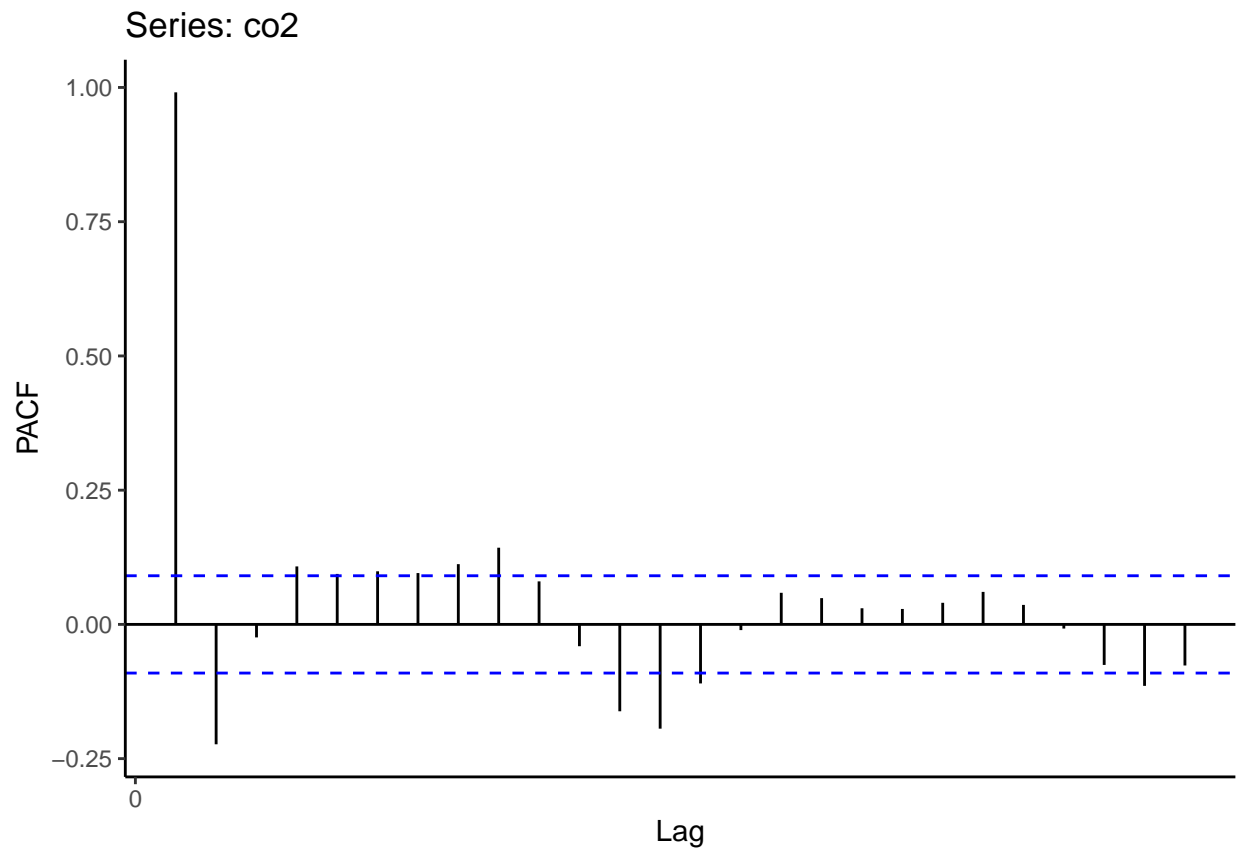
```
summary(co2)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    313.2   323.5   335.2   337.1   350.3   366.8
```

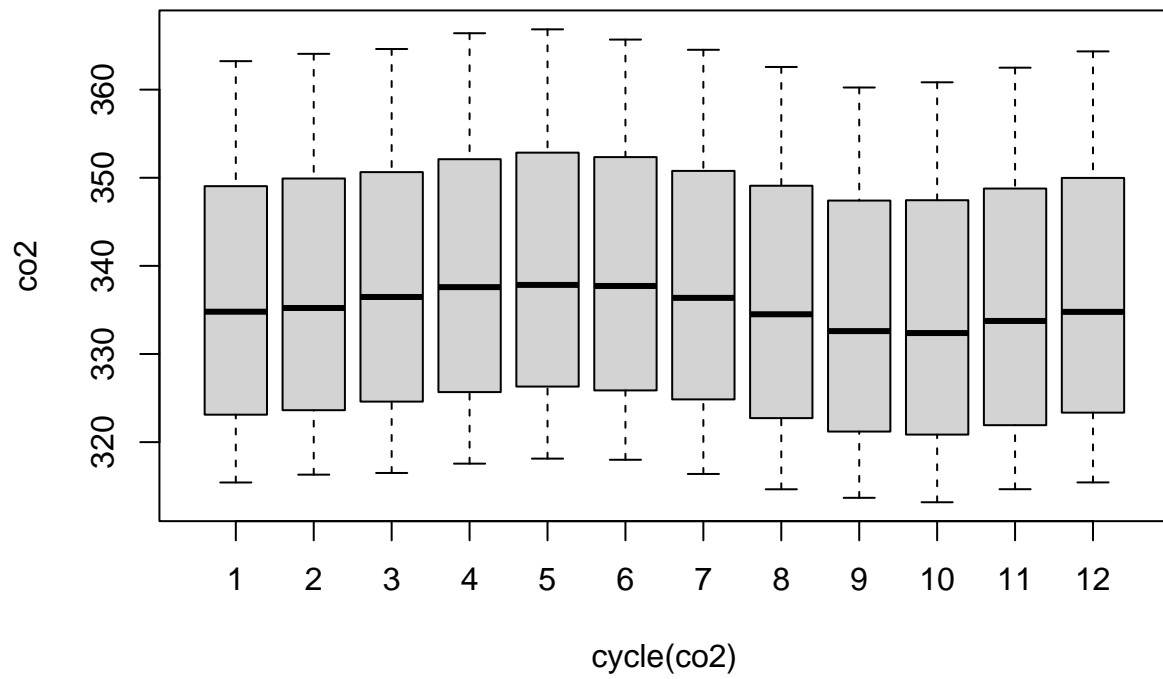
```
autoplot(acf(co2, plot = FALSE)) + theme_classic()
```



```
autoplot(pacf(co2, plot = FALSE)) + theme_classic()
```

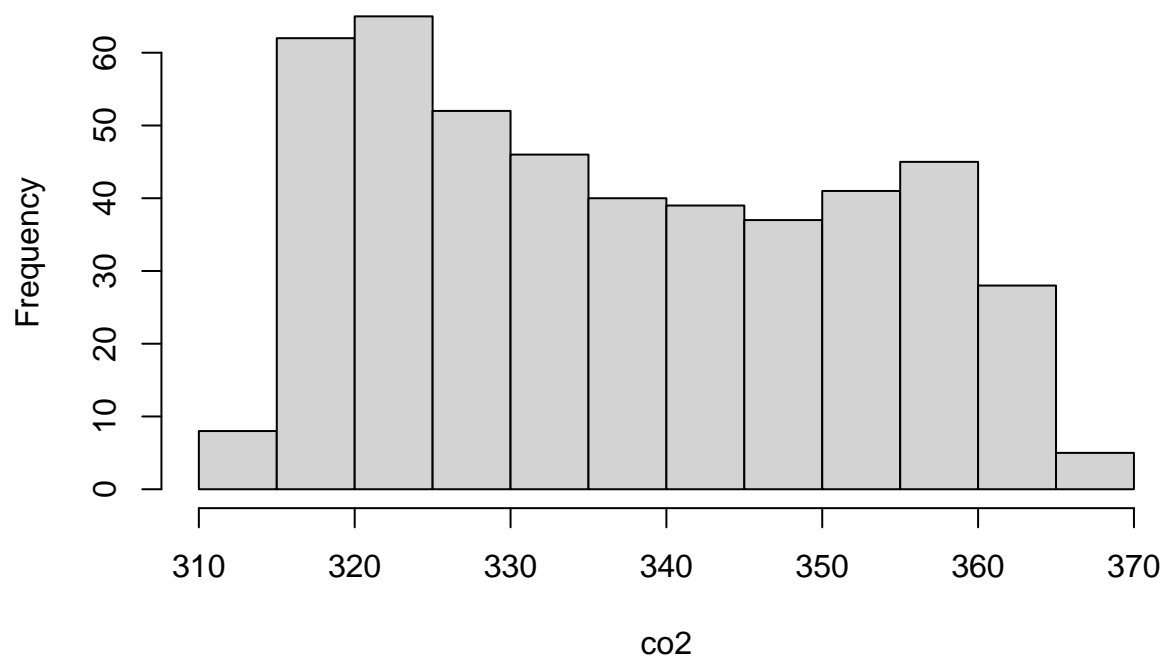


```
boxplot(co2 ~ cycle(co2))
```



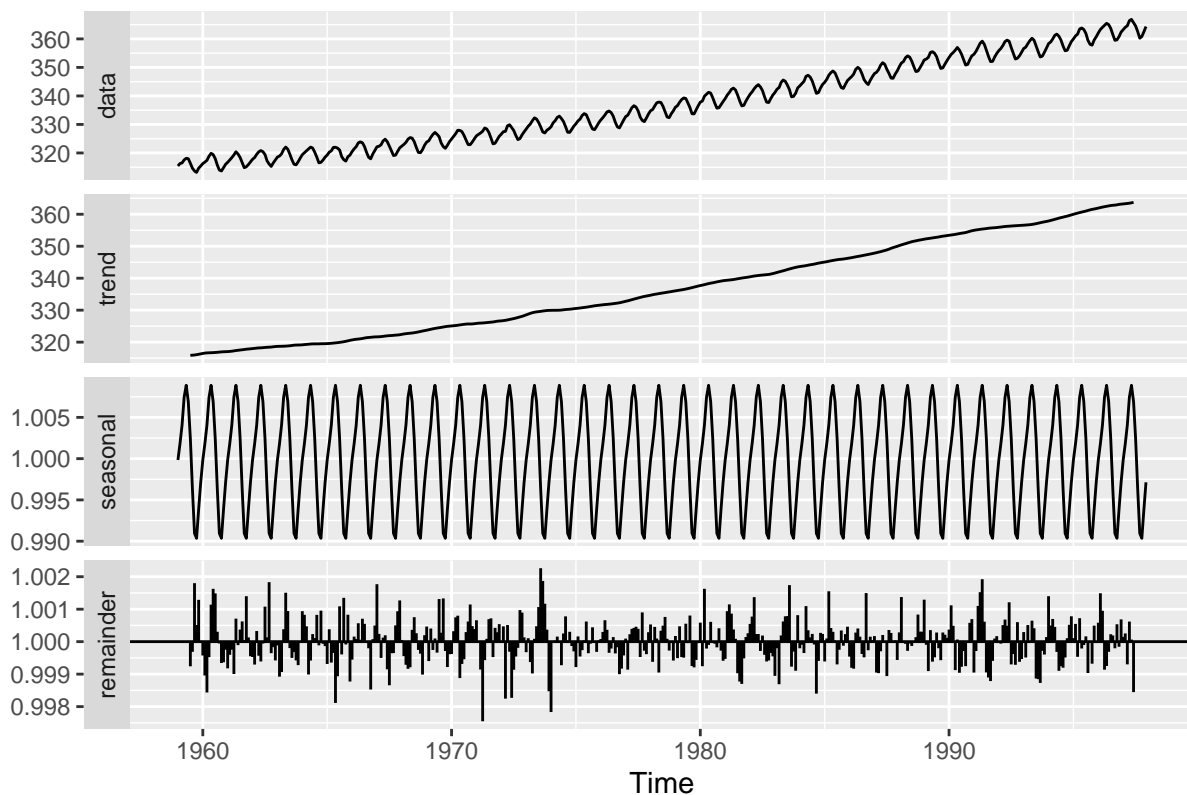
```
hist(co2)
```

Histogram of co2



```
decompose_data1 <- decompose(co2, "multiplicative")  
autoplot(decompose_data1)
```

Decomposition of multiplicative time series



Part 2 (3 points)

Fit a linear time trend model to the `co2` series, and examine the characteristics of the residuals. Compare this to a higher-order polynomial 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 up to the present.

Response:

```
linear_model <- tslm(co2 ~ trend)
summary(linear_model)
```

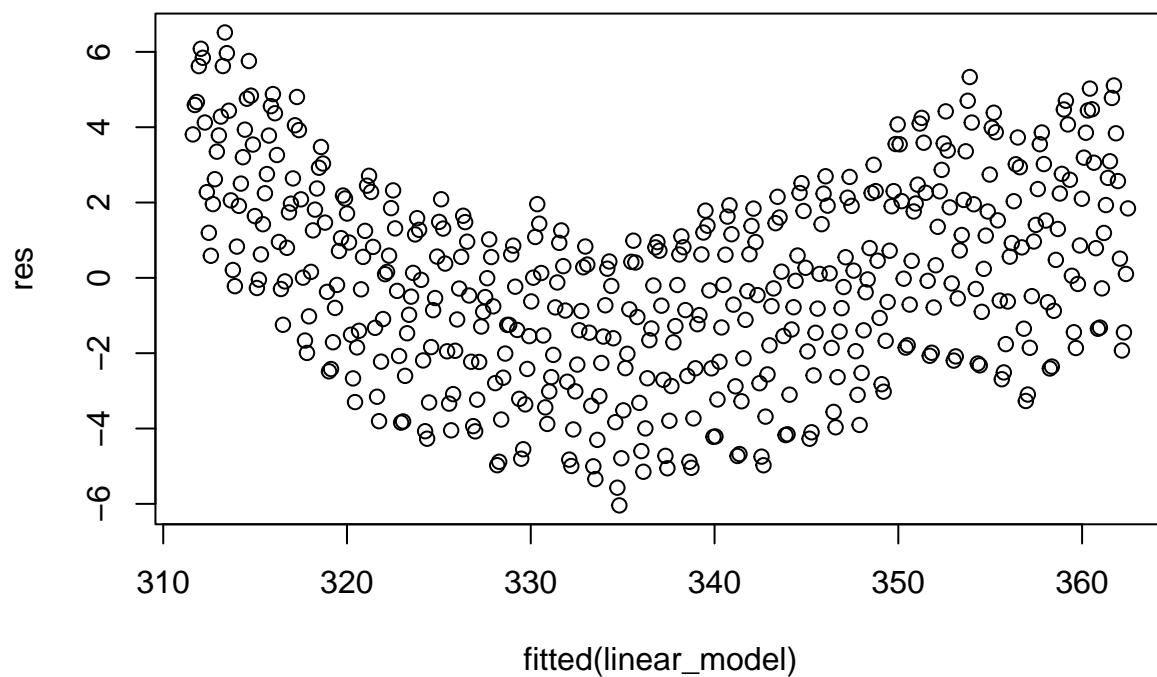
```
##
## Call:
## tslm(formula = co2 ~ trend)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.0399 -1.9476 -0.0017  1.9113  6.5149
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.115e+02  2.424e-01  1284.9  <2e-16 ***
## trend        1.090e-01  8.958e-04   121.6  <2e-16 ***
```



```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.618 on 466 degrees of freedom
## Multiple R-squared:  0.9695, Adjusted R-squared:  0.9694
## F-statistic: 1.479e+04 on 1 and 466 DF,  p-value: < 2.2e-16
```

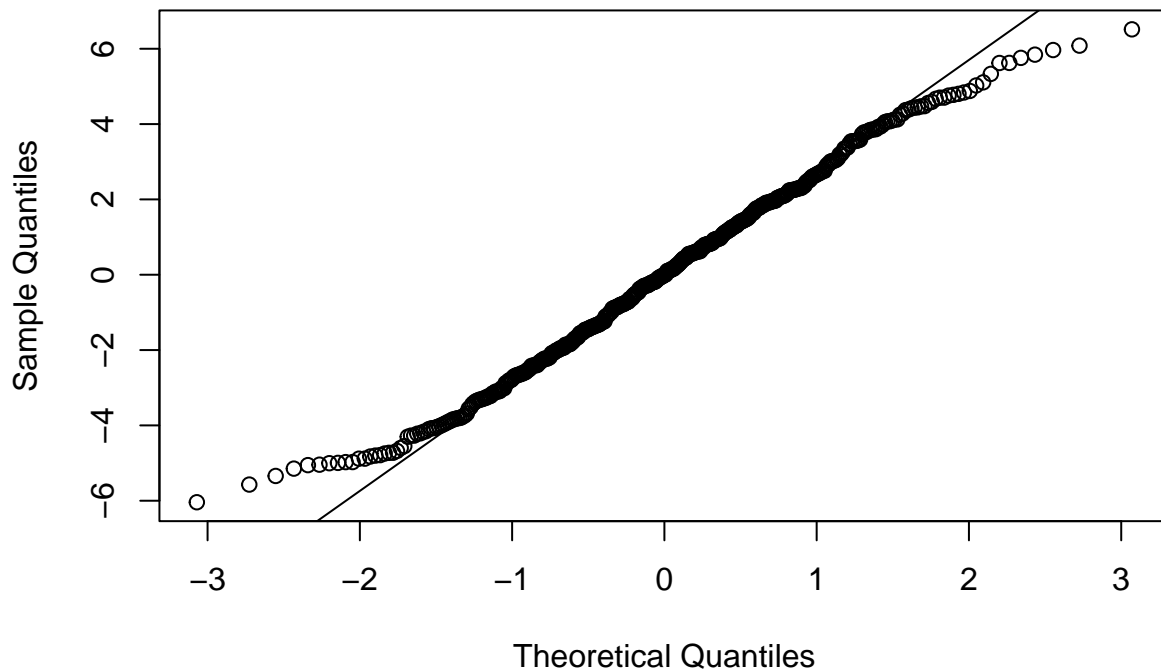
```
res <- resid(linear_model)

plot(fitted(linear_model), res)
```



```
qqnorm(res)
qqline(res)
```

Normal Q-Q Plot

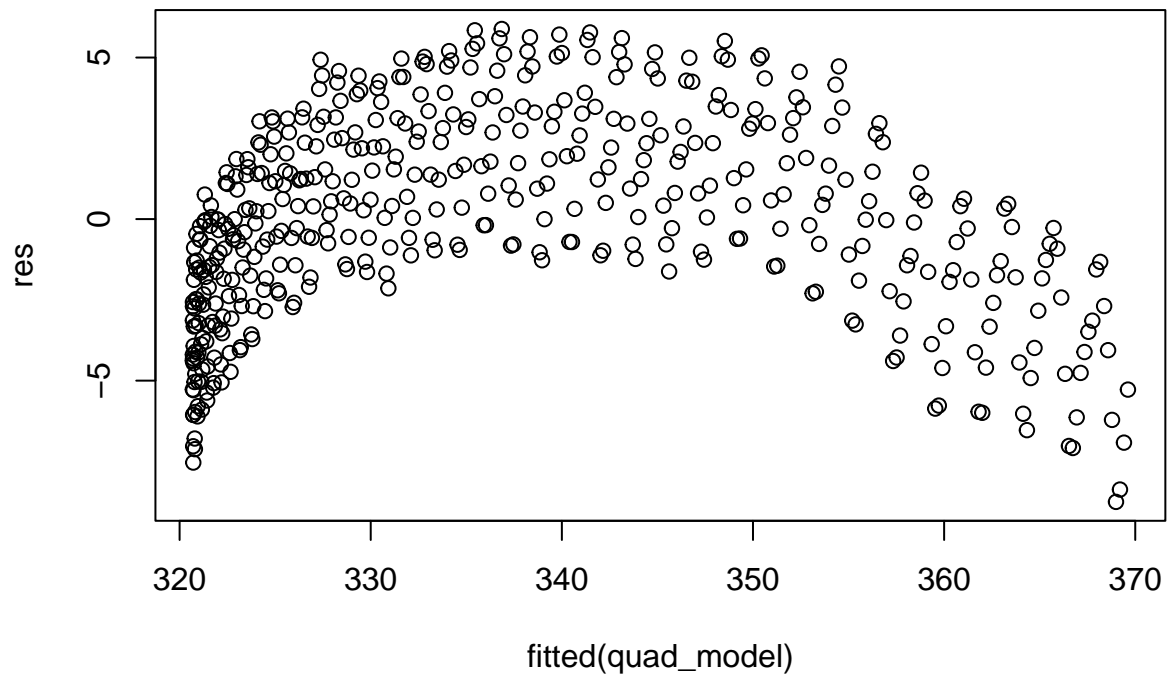


```
quad_model <- tslm(co2 ~ I(trend^2))
summary(quad_model)
```

```
##
## Call:
## tslm(formula = co2 ~ I(trend^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.7523 -1.9818 -0.0023  2.4092  5.8833
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.207e+02  2.187e-01  1466.4  <2e-16 ***
## I(trend^2)    2.234e-04  2.227e-06   100.3  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.152 on 466 degrees of freedom
## Multiple R-squared:  0.9557, Adjusted R-squared:  0.9556
## F-statistic: 1.006e+04 on 1 and 466 DF,  p-value: < 2.2e-16
```

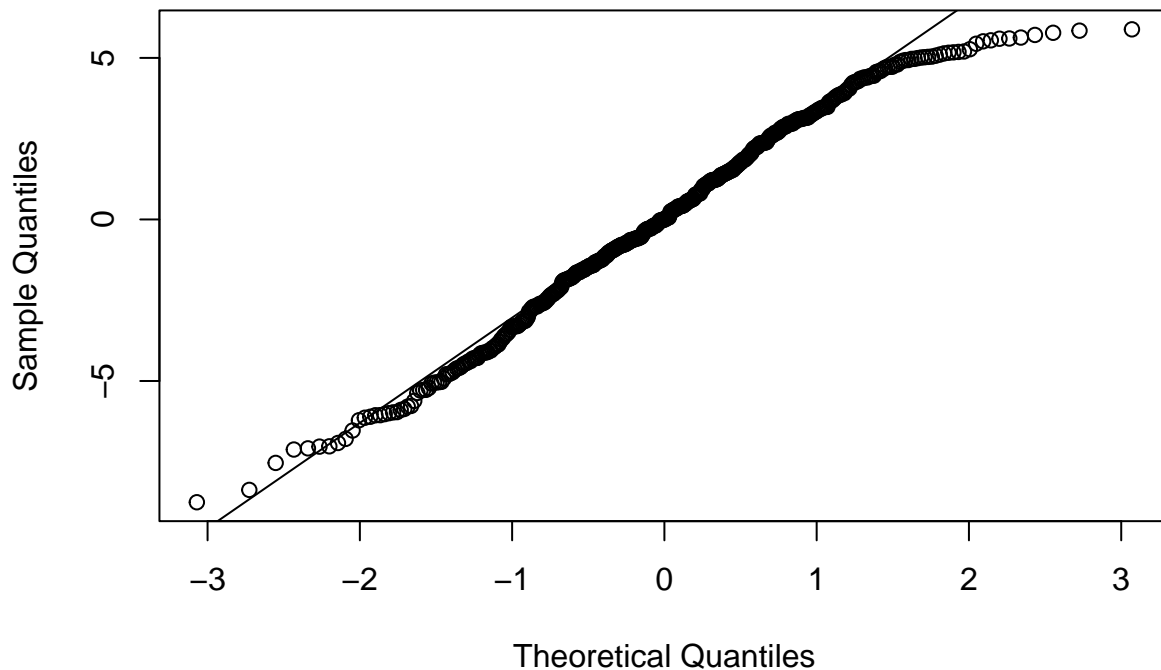
```
res <- resid(quad_model)

plot(fitted(quad_model), res)
```



```
qqnorm(res)
qqline(res)
```

Normal Q-Q Plot



```
seasonal_model <- tslm(co2 ~ I(trend^2) + seasonaldummy(co2))
```

```
summary(seasonal_model)
```

```
##
## Call:
## tslm(formula = co2 ~ I(trend^2) + seasonaldummy(co2))
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-5.7514	-2.2333	0.6857	2.0903	3.3276

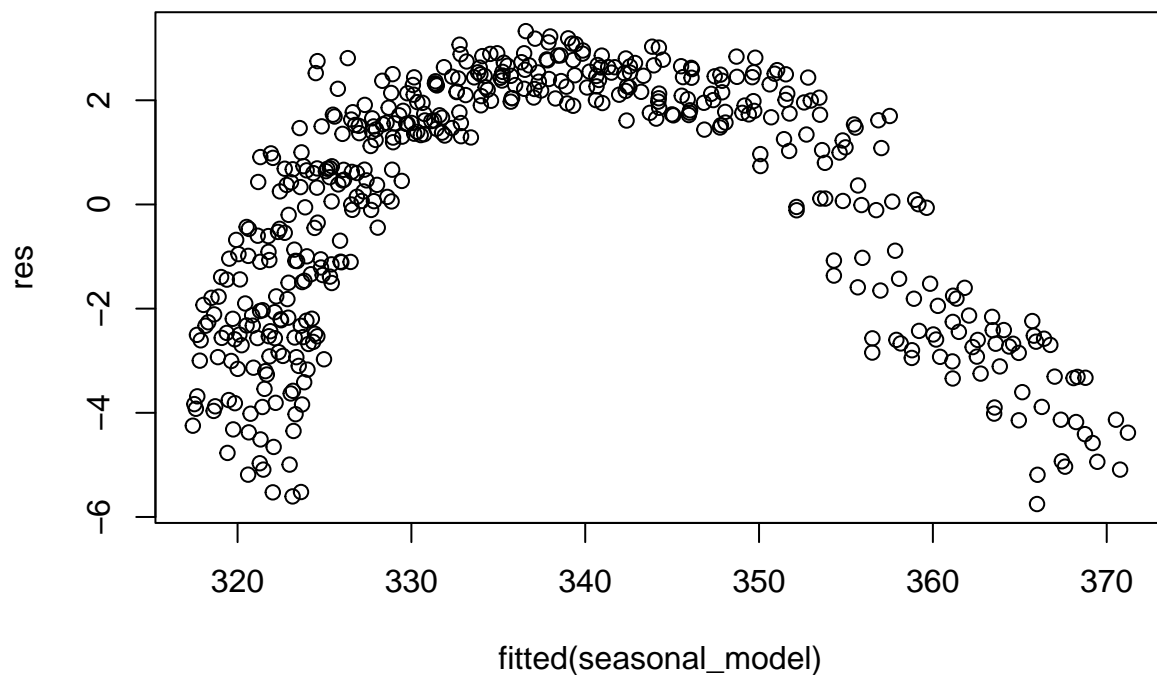
```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.197e+02	4.090e-01	781.610	< 2e-16 ***
I(trend^2)	2.239e-04	1.712e-06	130.774	< 2e-16 ***
seasonaldummy(co2)Jan	8.911e-01	5.486e-01	1.624	0.105024
seasonaldummy(co2)Feb	1.561e+00	5.486e-01	2.845	0.004640 **
seasonaldummy(co2)Mar	2.309e+00	5.486e-01	4.209	3.10e-05 ***
seasonaldummy(co2)Apr	3.445e+00	5.486e-01	6.280	7.93e-10 ***
seasonaldummy(co2)May	3.929e+00	5.486e-01	7.161	3.23e-12 ***
seasonaldummy(co2)Jun	3.270e+00	5.486e-01	5.961	5.03e-09 ***

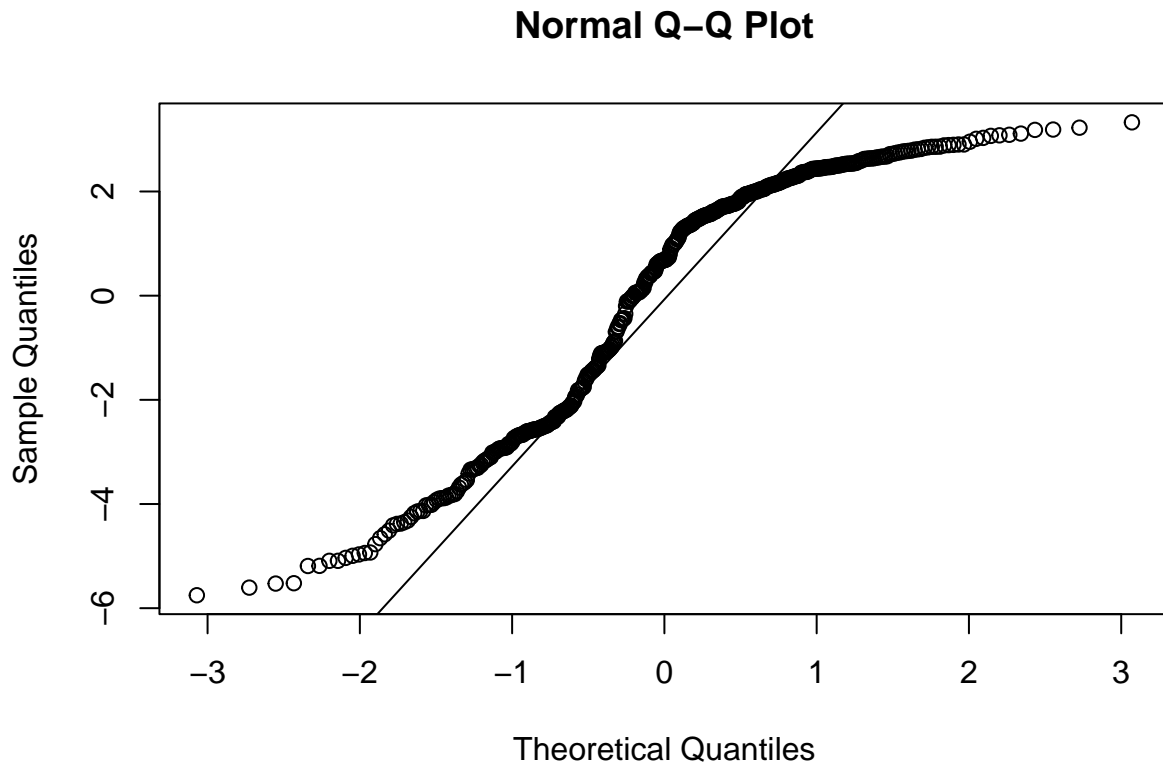
```
## seasonaldummy(co2)Jul  1.753e+00  5.486e-01   3.196 0.001489 **
## seasonaldummy(co2)Aug -3.104e-01  5.486e-01  -0.566 0.571773
## seasonaldummy(co2)Sep -2.131e+00  5.486e-01  -3.885 0.000117 ***
## seasonaldummy(co2)Oct -2.311e+00  5.486e-01  -4.213 3.03e-05 ***
## seasonaldummy(co2)Nov -1.119e+00  5.486e-01  -2.041 0.041864 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.422 on 455 degrees of freedom
## Multiple R-squared:  0.9745, Adjusted R-squared:  0.9738
## F-statistic: 1448 on 12 and 455 DF, p-value: < 2.2e-16
```

```
res <- resid(seasonal_model)

plot(fitted(seasonal_model), res)
```



```
qqnorm(res)
qqline(res)
```



```
# forecast <- forecast(seasonal_model, h=300, level=95)
# plot(forecast)
```

Part 3 (4 points)

Following all appropriate steps, choose an ARIMA model to fit to this `co2` series. Discuss the characteristics of your model and how you selected between alternative ARIMA specifications. Use your model to generate forecasts to the present.

Response: Below I have fitted the best ARIMA model from multiple iterations, the starting point coming from looking at some of the EDA graphs completed above. Since there is a strong seasonality cycle in the decomposition plot, I will be using a seasonal ARIMA model. In a seasonal ARIMA model the format is shown below:

ARIMA(p,d,q)(P,D,Q)m

From this model, the lowercase letters are for the non-seasonal terms and the uppercase letters are the seasonal terms with m being the seasonal frequency. The p is the auto-regressive value, the d is the differencing term, and the q is the moving average term.

From the EDA completed above, there were a key points to take into consideration when deciding what kind of ARIMA model to use. First, from the decomposition plot, there is an obvious trend, which will need to be taken out by differencing once. However, I was not sure if it should be a seasonal or non-seasonal differencing value, therefore I tested both to see which had better results. From the ACF plot in the EDA section, the values do not decline towards zero quickly, therefore

the auto-regressive value will definitely not be zero. Lastly, for the PACF graph shows that the model may have a moving average value of 1 or 2 and a seasonal moving average value of 1 or 2.

With all of these values as starting points, I iteratively tried the different combination to look for the model with the lowest AIC value (172.71). I also looked through the residual values and the ACF and PACF graphs of the residuals to see how the model performed. The model I chose as the best has non-seasonal values of $p = 1$, $d = 1$, $q = 1$, and seasonal values of $p = 2$, $d = 1$, $q = 1$, with the seasonal frequency being 12 for the months. The residuals of this model are all fairly low, and the ACF and PACF graphs only have one spike that goes past the blue dotted line.

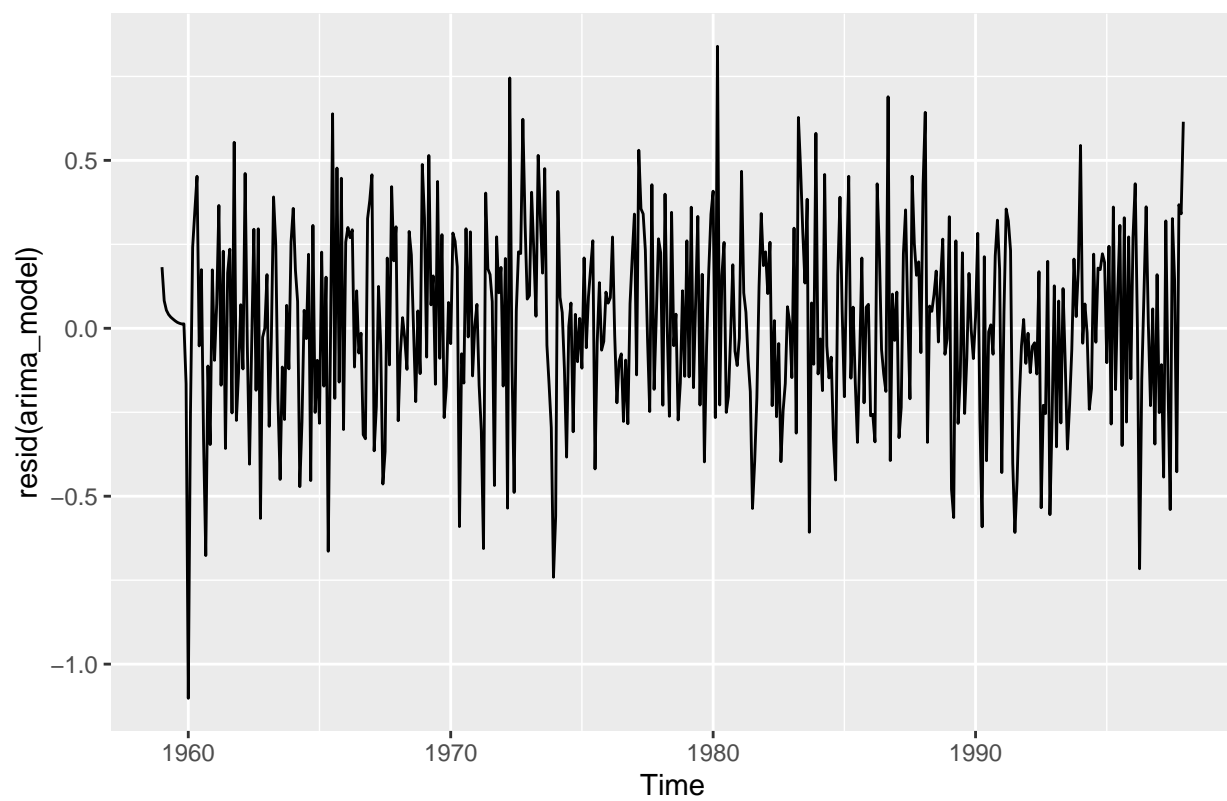
Lastly, there is a forecast plot of the model below that forecasts the next 25 years of data, with a confidence interval on the graph.

```
arima_model <- arima(co2, order = c(1, 1, 1), seas = list(order = c(2,
  1, 2), 12))

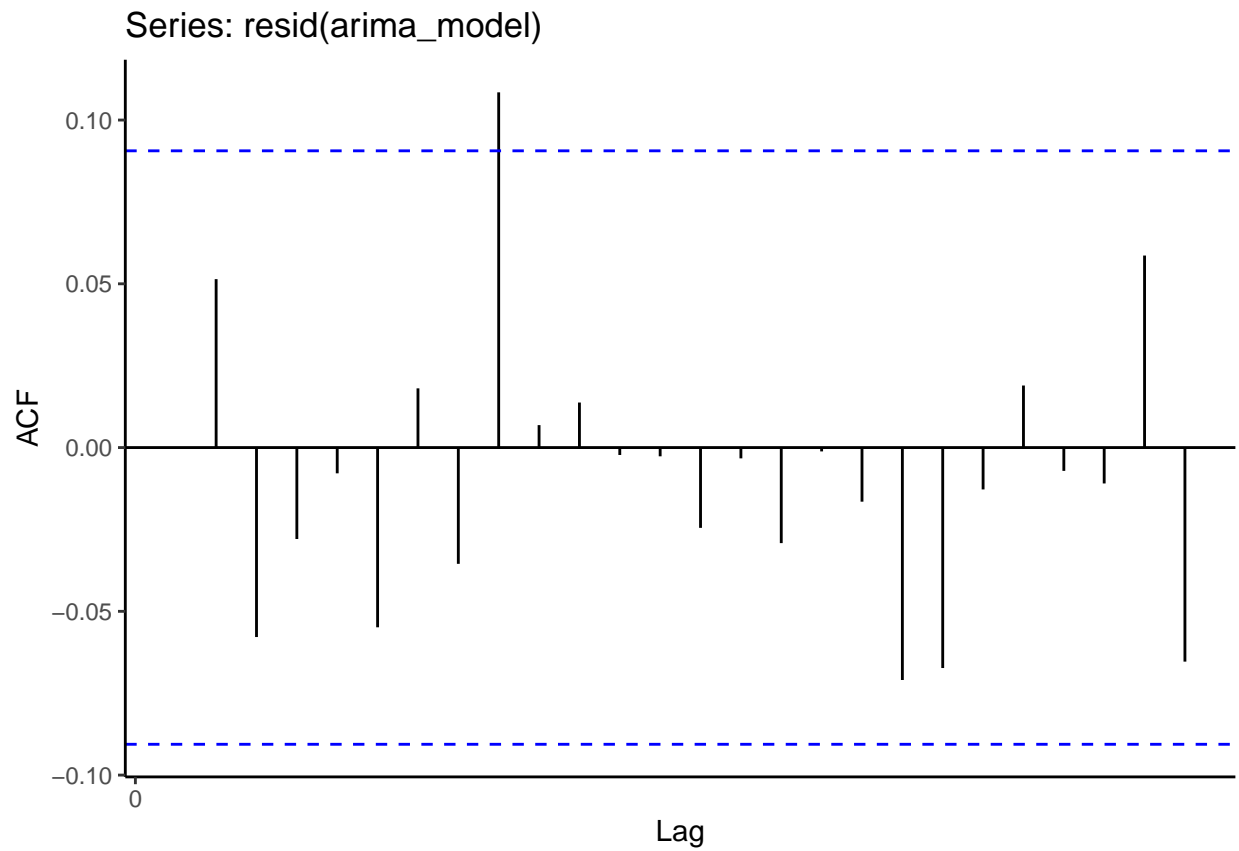
arima_model

##
## Call:
## arima(x = co2, order = c(1, 1, 1), seasonal = list(order = c(2, 1, 2), 12))
##
## Coefficients:
##          ar1          ma1          sar1          sar2          sma1          sma2
##          0.2652   -0.5950   0.9613   -0.1335   -1.8169    0.8564
## s.e.    0.1355    0.1152   0.0789    0.0603    0.0712    0.0624
##
## sigma^2 estimated as 0.0791:  log likelihood = -79.35,  aic = 172.71
```

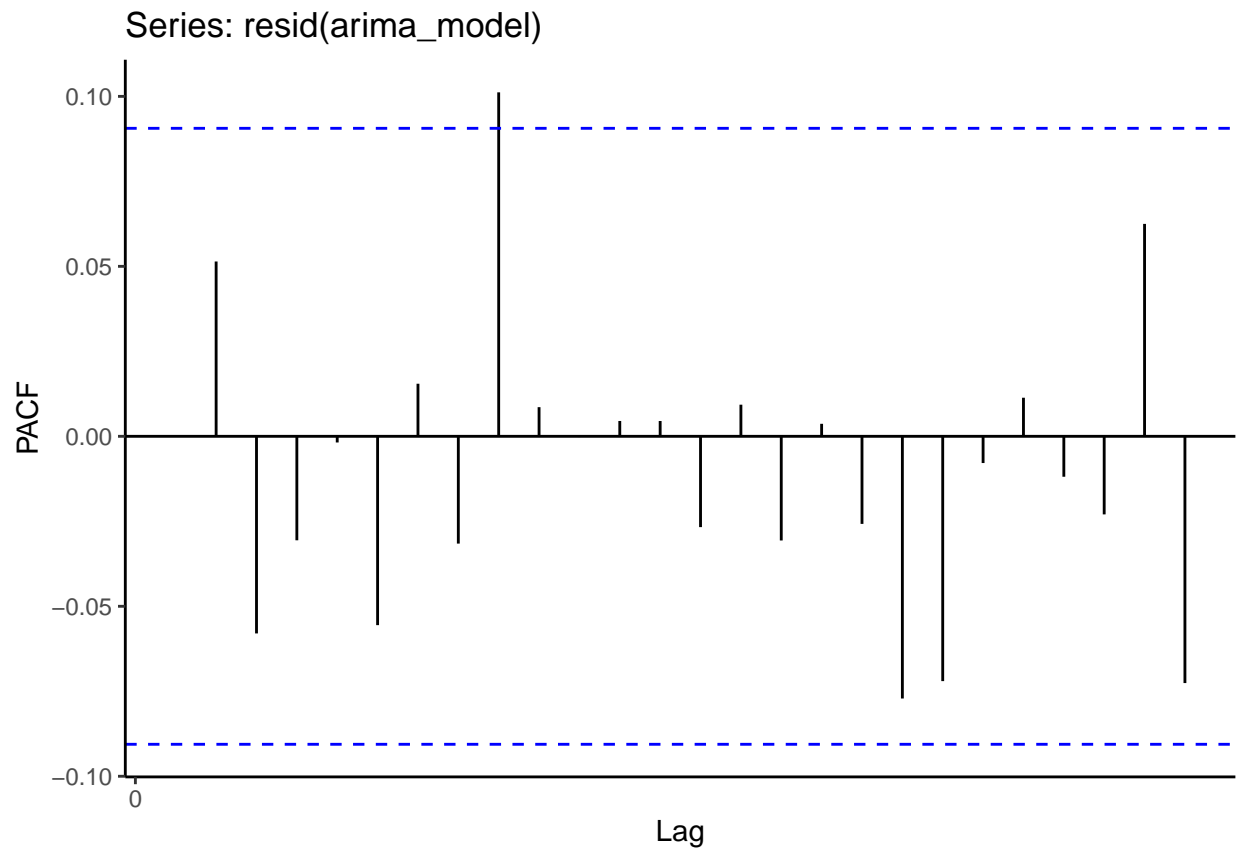
```
autoplot(resid(arima_model))
```



```
autoplot(acf(resid(arima_model), plot = FALSE)) + theme_classic()
```

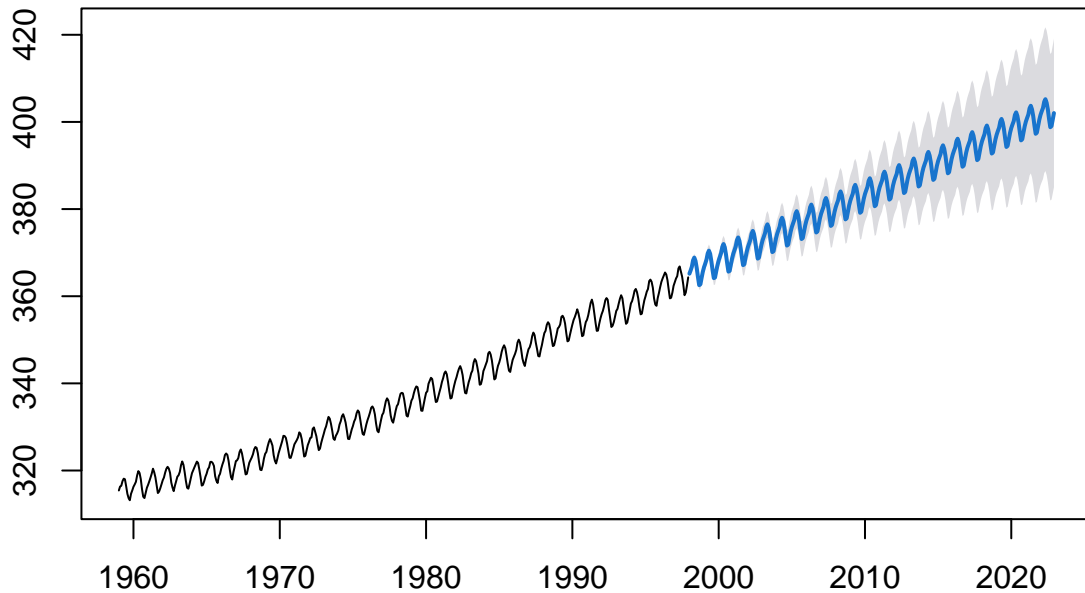



```
autoplot(pacf(resid(arima_model), plot = FALSE)) + theme_classic()
```



```
myforecast <- forecast(arima_model, level = c(95), h = 300)
plot(myforecast)
```

Forecasts from ARIMA(1,1,1)(2,1,2)[12]



Part 4 (5 points)

The file `co2_weekly_mlo.txt` contains weekly observations of atmospheric carbon dioxide concentrations measured at the Mauna Loa Observatory from 1974 to 2020, published by the National Oceanic and Atmospheric Administration (NOAA). Convert these data into a suitable time series object, conduct a thorough EDA on the data, addressing the problem of missing observations and comparing the Keeling Curve's development to your predictions from Parts 2 and 3. Use the weekly data to generate a month-average series from 1997 to the present and use this to generate accuracy metrics for the forecasts generated by your models from Parts 2 and 3.

Part 5 (5 points)

Split the NOAA series into training and test sets, using the final two years of observations as the test set. Fit an ARIMA model to the series following all appropriate steps, including comparison of how candidate models perform both in-sample and (pseudo-) out-of-sample. Generate predictions for when atmospheric CO₂ is expected to reach 450 parts per million, considering the prediction intervals as well as the point estimate. Generate a prediction for atmospheric CO₂ levels in the year 2100. How confident are you that these will be accurate predictions?