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

Brittany Dougall, Steve Hall, Prabhu Narsina, and Edward Salinas

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'.

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.

```
opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE,
    warning = FALSE, message = FALSE)

str(co2)
```

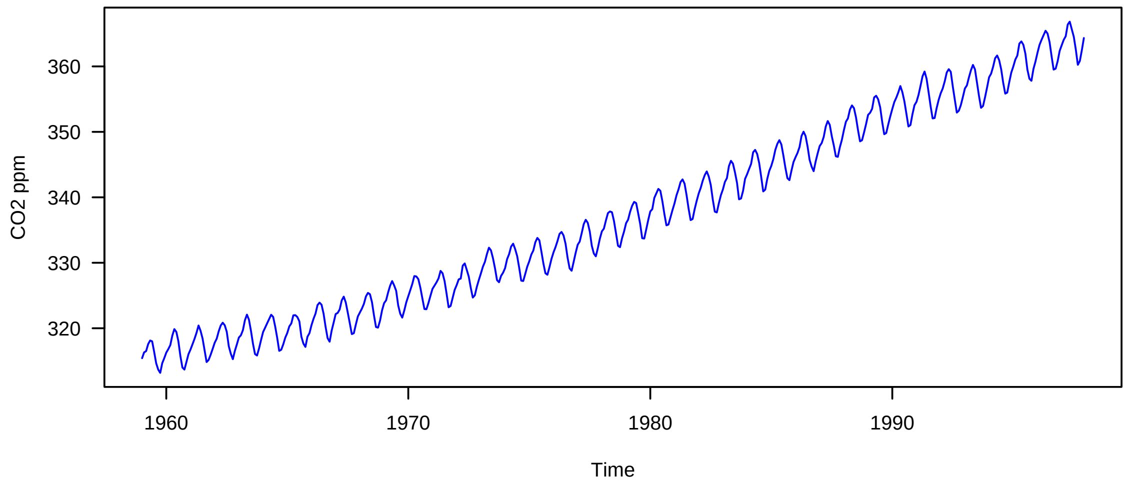
```
##  Time-Series [1:468] from 1959 to 1998: 315 316 316 318 318 ...
```

```
summary(co2)
```

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

```
co2.decompose = decompose(co2)
co2.diff = diff(co2, differences = 1)
co2.seasdiff = diff(co2, lag = 12)
co2.bothdiff = diff(co2.diff, lag = 12)

co2.deseasoned = co2 - co2.decompose$seasonal
co2.detrended = co2 - co2.decompose$trend

par(mfrow = c(3, 1))

plot(co2, ylab = expression("CO2 ppm"), col = "blue", las = 1)
title(main = "Figure1: Monthly Mean CO2 Variation")

boxplot(co2 ~ cycle(co2), main = "Boxplot of CO2 (ppm) by month")

plot(co2.deseasoned, main = expression("Figure2: Presence of CO2 in air  after removing season
    xlab = "year", ylab = expression("CO2 ppm"))
```

Figure1: Monthly Mean CO2 Variation



Boxplot of CO2 (ppm) by month

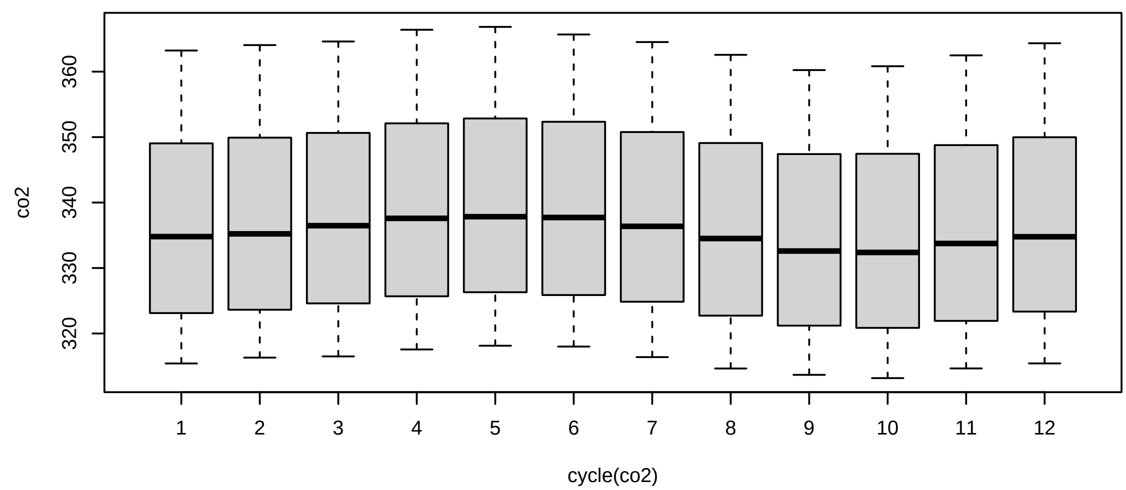
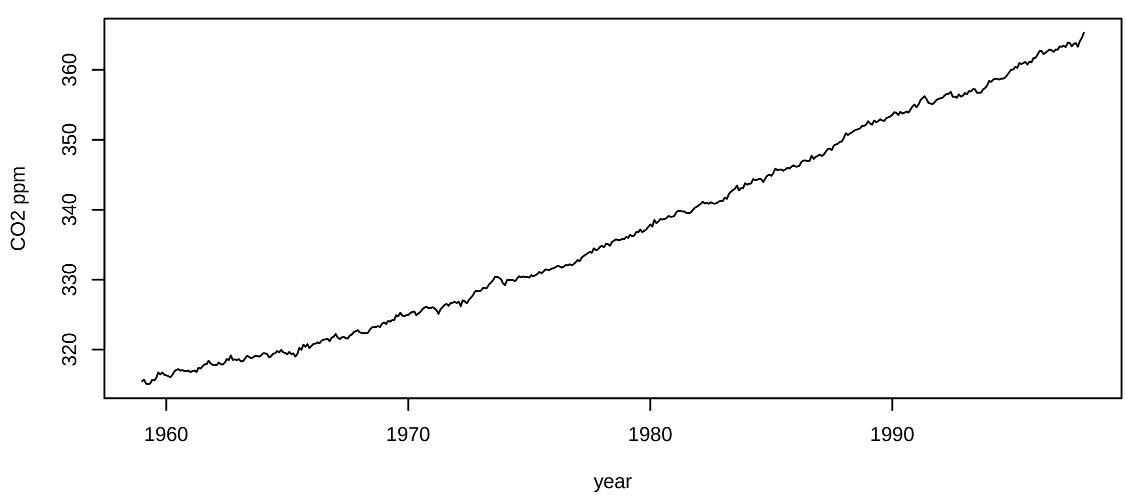


Figure2: Presence of CO2 in air after removing season



```
plot(co2.detrended, main = expression("Figure3: Presence of CO2 in air after removing trend"),
    xlab = "year", ylab = expression("CO2 ppm"), col = "red",
    las = 1)

abline(h = 0)

plot(co2.diff, main = expression("Figure4: Presence of CO2 in air after differencing"),
    xlab = "year", ylab = expression("CO2 ppm"), col = "red",
    las = 1)

abline(h = 0)

plot(co2.seasdiff, main = expression("Figure5: Presence of CO2 in air after seasonal differenc
    xlab = "year", ylab = expression("CO2 ppm"), col = "red",
    las = 1)
abline(h = 0)
```

Figure3: Presence of CO2 in air after removing trend

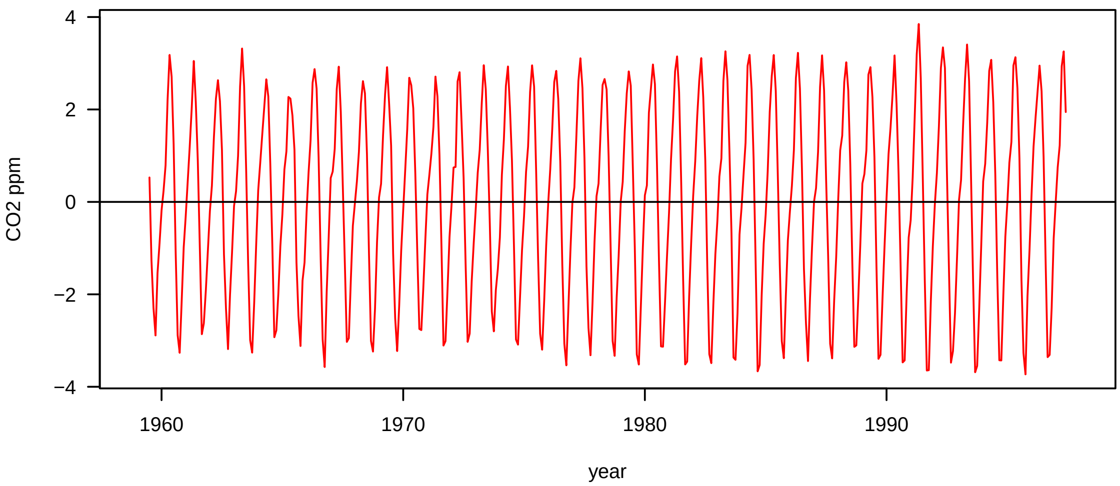


Figure4: Presence of CO2 in air after differencing

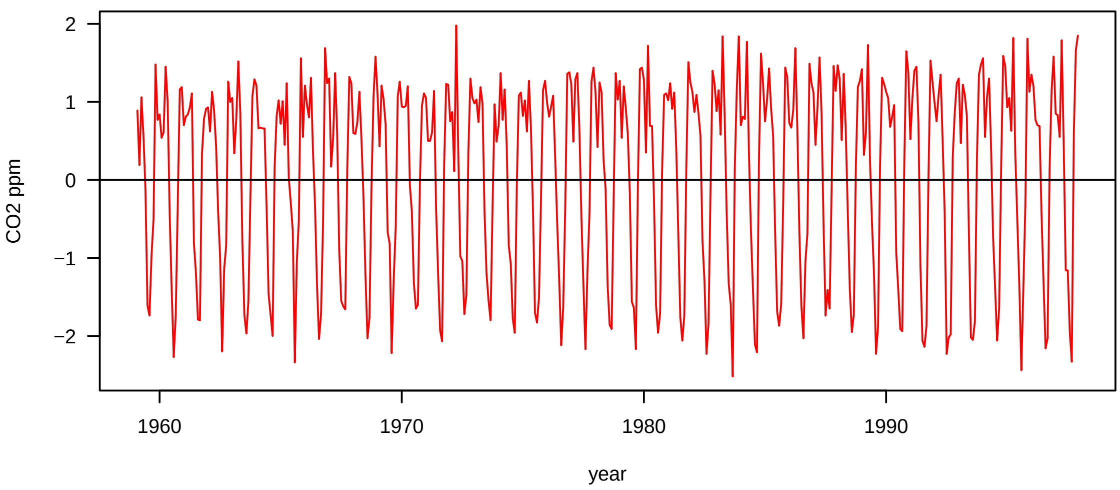
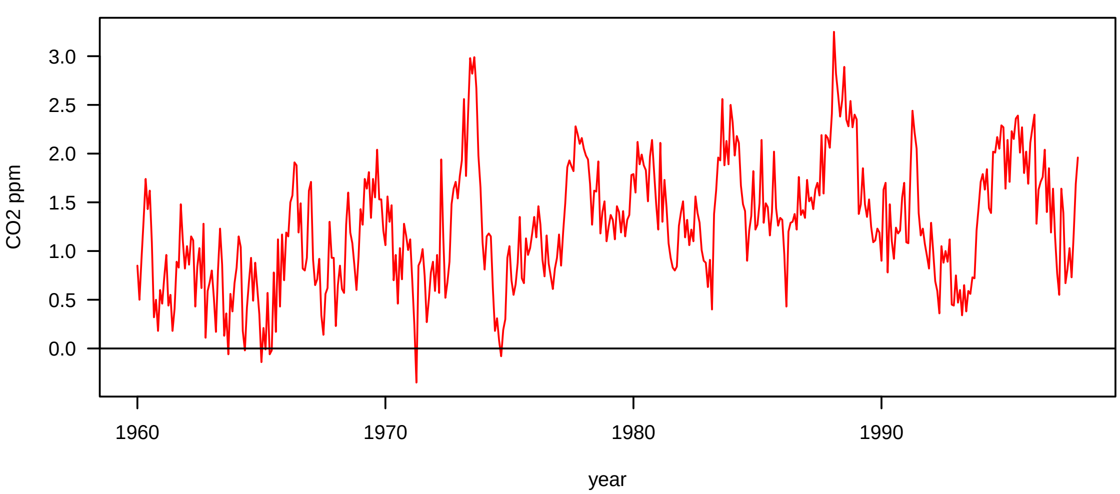
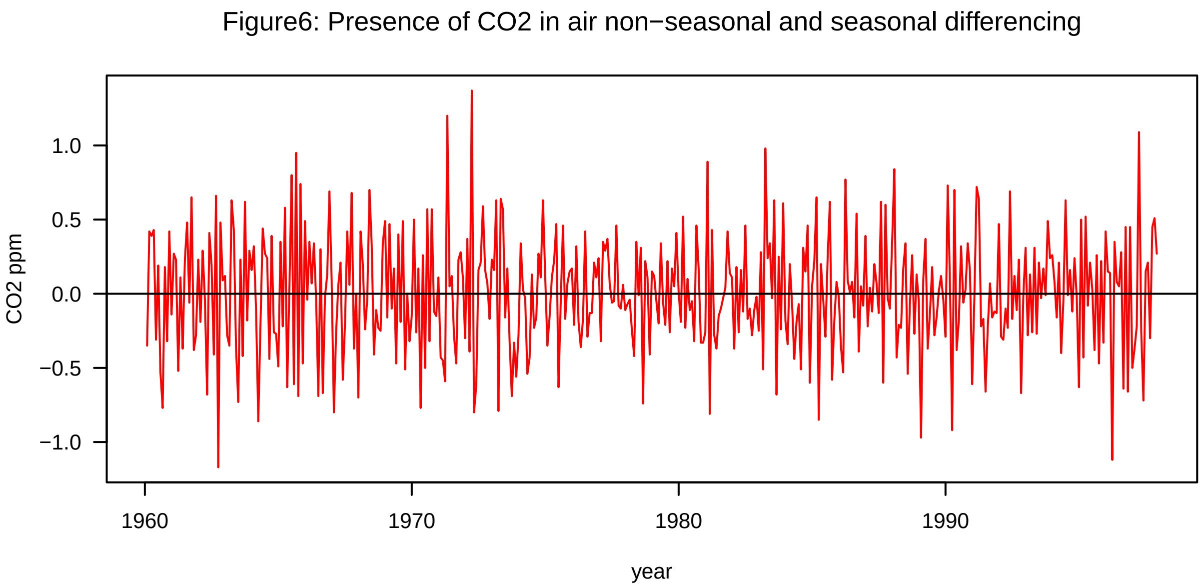


Figure5: Presence of CO2 in air after seasonal differencing

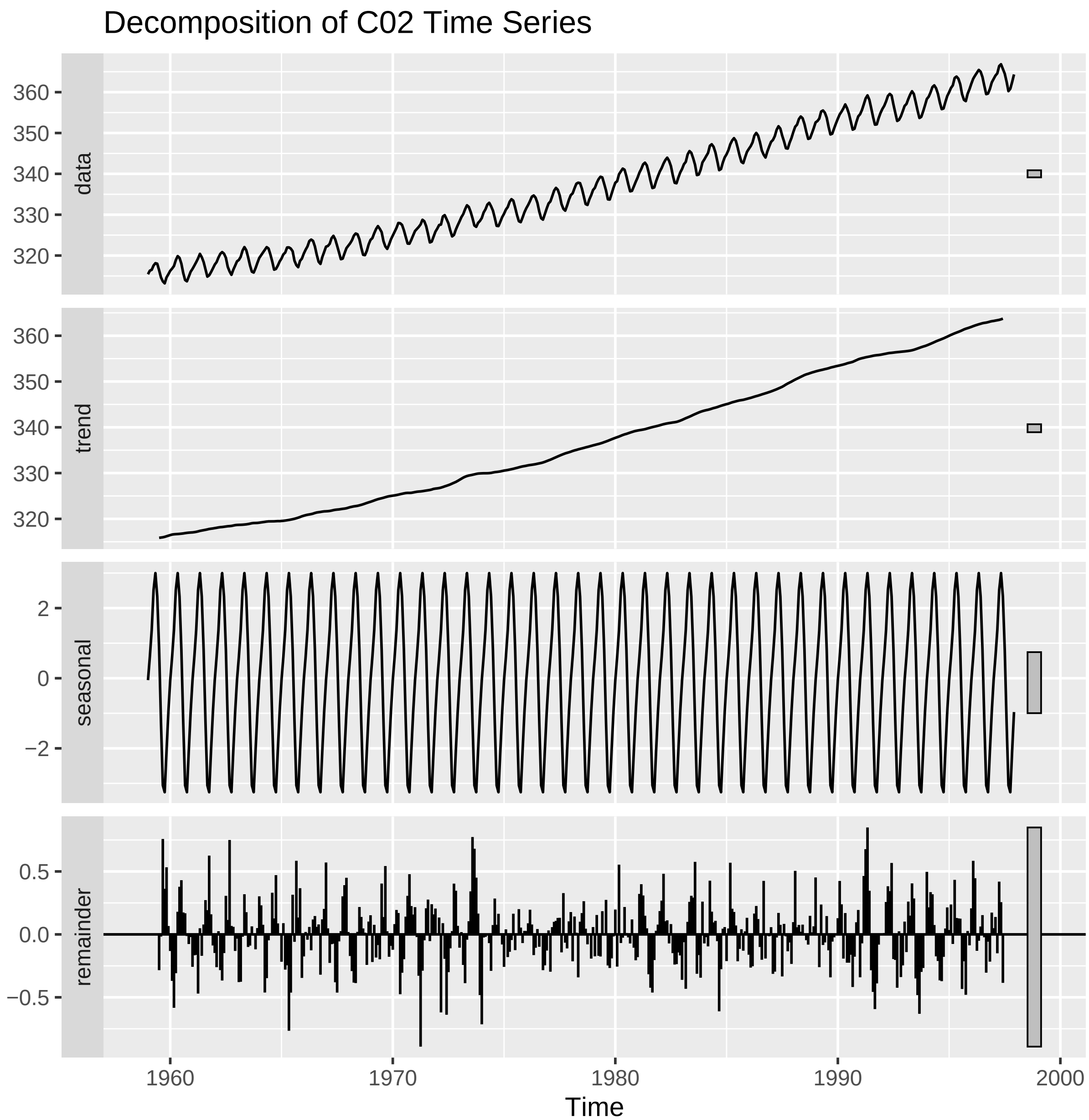


```r
plot(co2.bothdiff, main = expression("Figure6: Presence of CO2 in air non-seasonal and seasonal
    xlab = "year", ylab = expression("CO2 ppm"), col = "red",
    las = 1)
abline(h = 0)
```



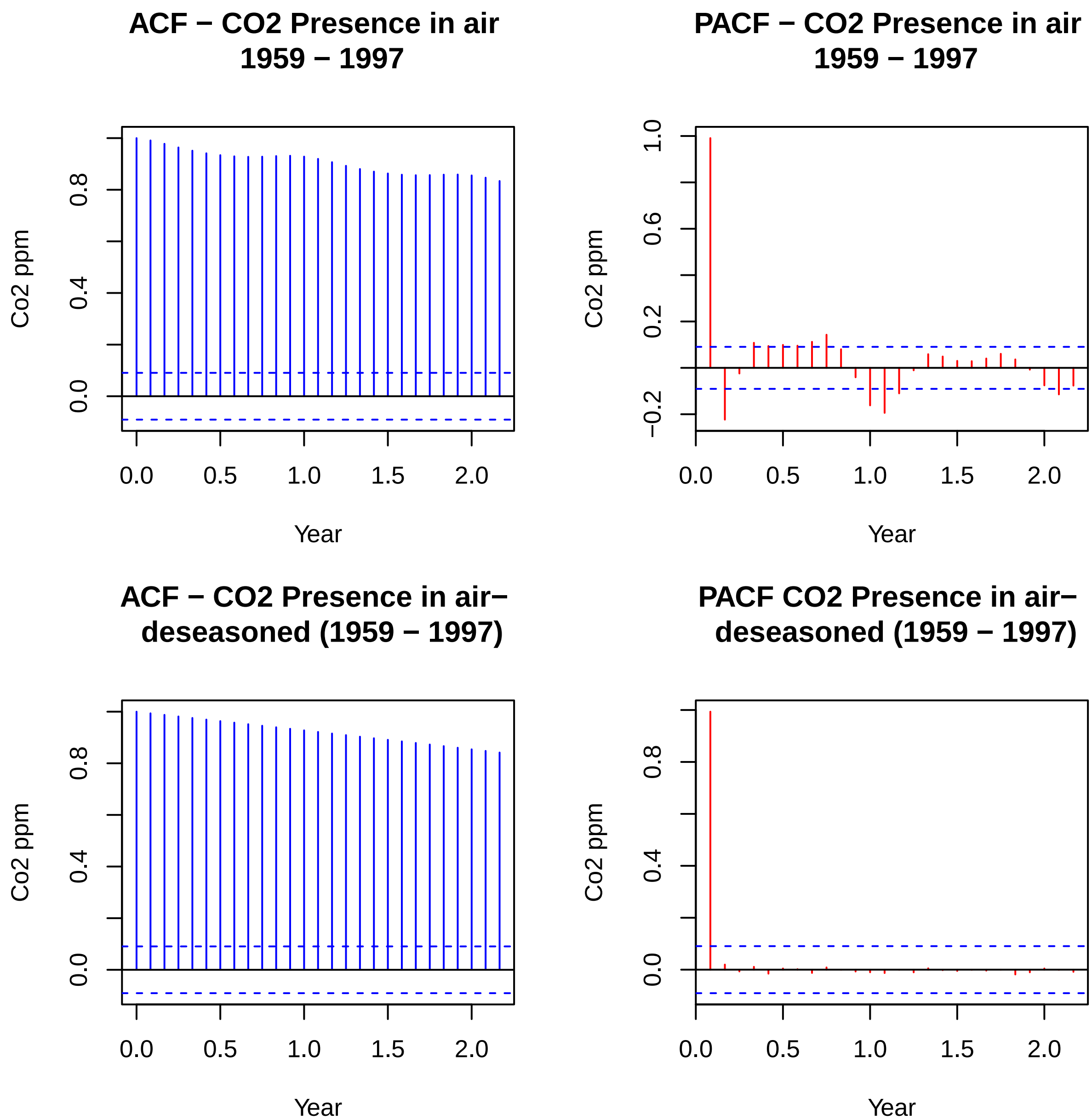
Data provided has CO2 presence in the air (parts per million) in monthly time series format from 1959 to 1998.
From Figure1: The time series plot of the mean of co2 presence in the air indicates a clear trend and seasonal effect. We also observe that the variance is constant over time, which suggests no need for transformation.
From Figure2: We see a clear upward trend in the mean of the presence of Co2 in the air
From Figure3: Co2 presence in the air after removing the trend component from the time series indicates the persistent yearly seasonal effect.
From Figure4: Trend is abstracted after taking the 2-period difference of the time series. It suggests we use ARIMA with integration/difference of 2
From Figure5: Seasonality absent after applying difference of 12 lags for the season. We still see trends present.
From Figure6: Seasonality and trend are absent after difference at two lags and 12 lags for the season. It is much closer to white noise series with non-constant variance. It suggests a possible need of Seasonal adjustment for the ARIMA model

```r
autoplot(co2.decompose, main = "Decomposition of CO2 Time Series")
```

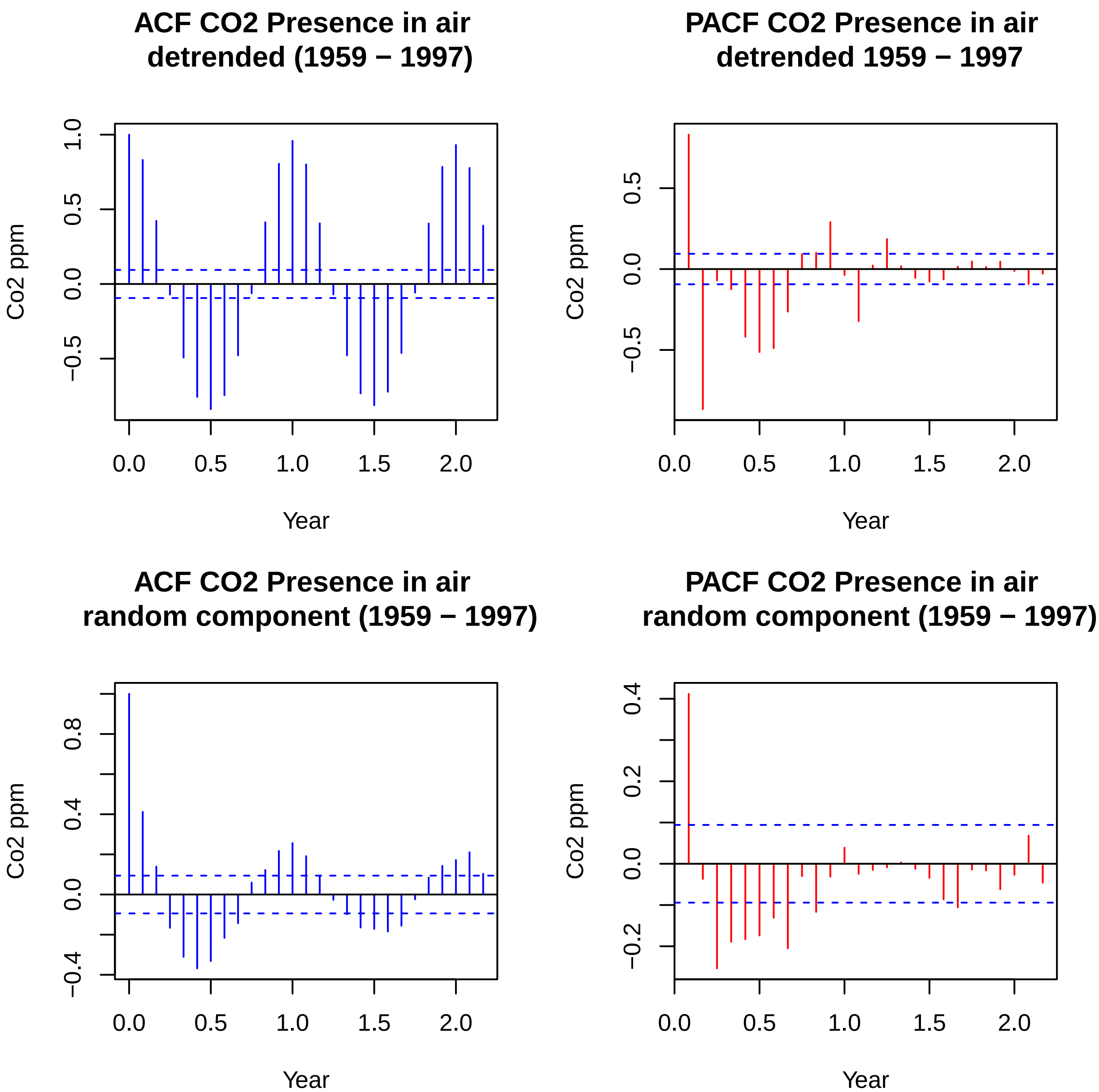



```r
plot.acf.alldata = acf(co2, plot = FALSE)
plot.pacf.alldata = pacf(co2, plot = FALSE)

plot.acf.deseasoned = acf(co2.deseasoned, plot = FALSE)
plot.pacf.deseasoned = pacf(co2.deseasoned, plot = FALSE)

plot.acf.detrended = acf(window(co2.detrended, start = c(1960),
    end = c(1996)), plot = FALSE)
plot.pacf.detrended = pacf(window(co2.detrended, start = c(1960),
    end = c(1996)), plot = FALSE)

plot.acf.residual = acf(window(co2.decompose$random, start = c(1960),
    end = c(1996)), plot = FALSE)
```

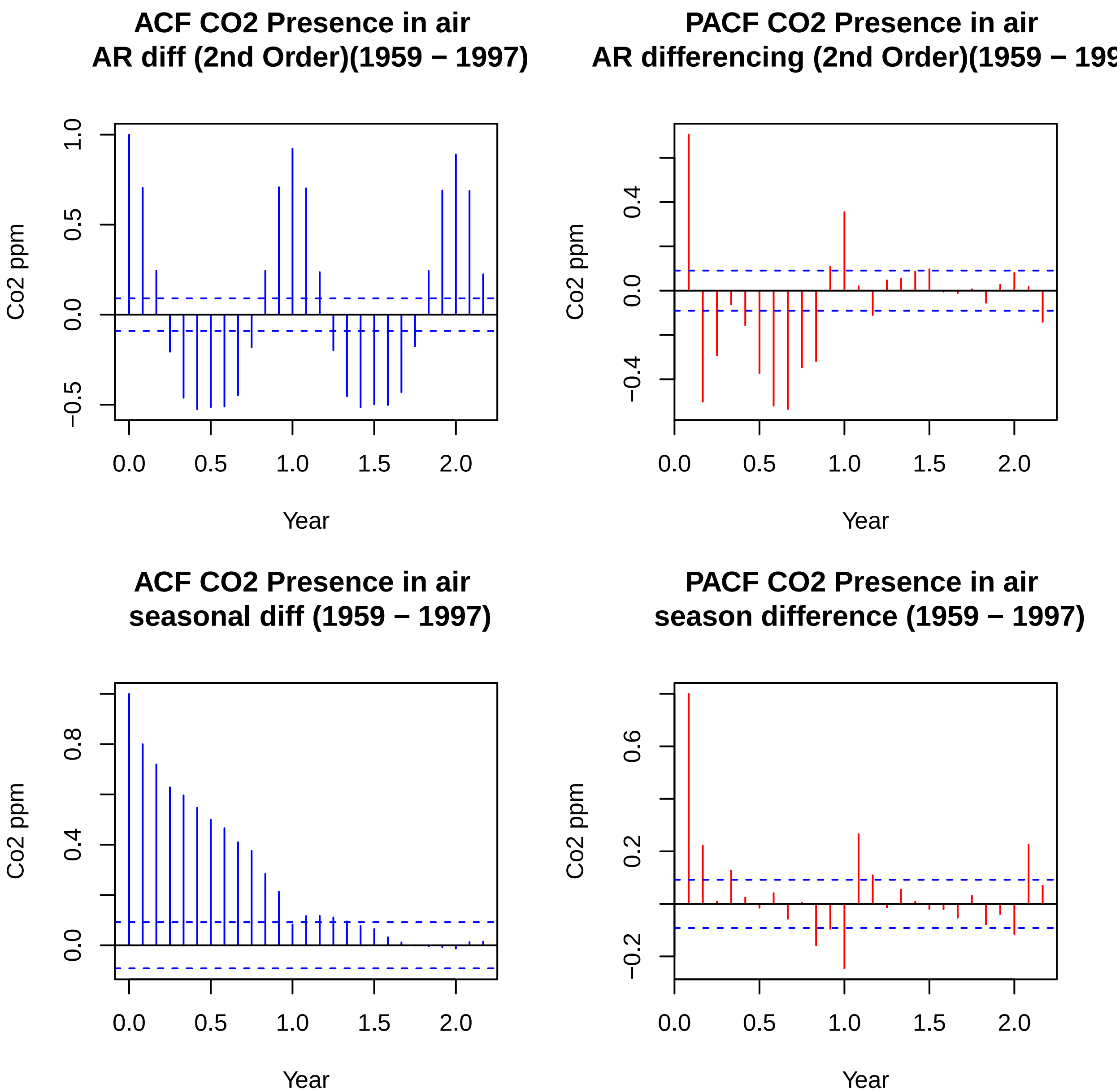
```
plot.pacf.residual = pacf(window(co2.decompose$random, start = c(1960),
    end = c(1996)), plot = FALSE)

plot.acf.diff = acf(co2.diff, plot = FALSE)
plot.pacf.diff = pacf(co2.diff, plot = FALSE)

plot.acf.seasondiff = acf(co2.seasdiff, plot = FALSE)
plot.pacf.seasondiff = pacf(co2.seasdiff, plot = FALSE)

plot.acf.bothdiff = acf(co2.bothdiff, plot = FALSE)
plot.pacf.bothdiff = pacf(co2.bothdiff, plot = FALSE)

par(mfrow = c(2, 2))
plot(plot.acf.alldata, main = "ACF - CO2 Presence in air \n 1959 - 1997",
    xlab = "Year", ylab = "Co2 ppm", col = "blue", cex.main = 0.5)
plot(plot.pacf.alldata, main = "PACF - CO2 Presence in air \n 1959 - 1997",
    xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)


plot(plot.acf.deseasoned, main = "ACF - CO2 Presence in air- \n deseasoned (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "blue")
plot(plot.pacf.deseasoned, main = "PACF CO2 Presence in air- \n deseasoned (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)
```



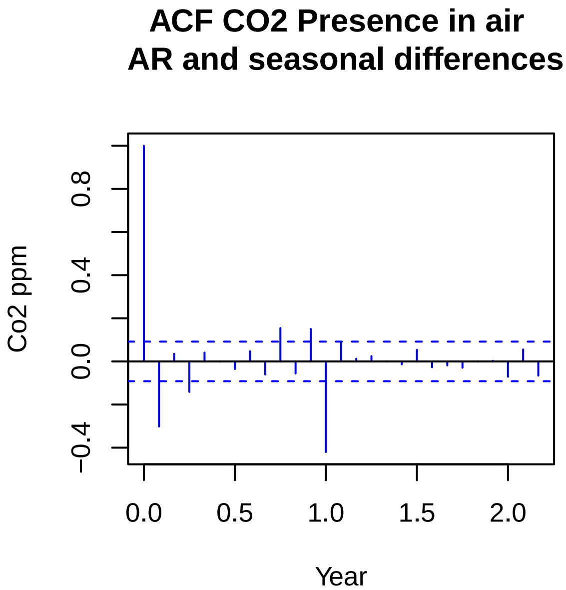

```r
plot(plot.acf.detrended, main = "ACF CO2 Presence in air \n detrended (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "blue")
plot(plot.pacf.detrended, main = "PACF CO2 Presence in air \n detrended 1959 - 1997",
    xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)

plot(plot.acf.residual, main = "ACF CO2 Presence in air \n random component (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "blue")
plot(plot.pacf.residual, main = "PACF CO2 Presence in air \n random component (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)
```

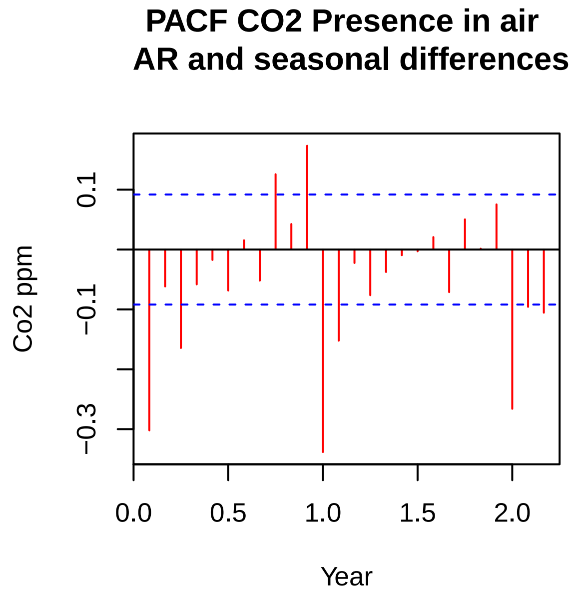



```
plot(plot.acf.diff, main = "ACF CO2 Presence in air \n AR diff (2nd Order)(1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "blue")
plot(plot.pacf.diff, main = "PACF CO2 Presence in air \n AR differencing (2nd Order)(1959 - 19
    xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)

plot(plot.acf.seasondiff, main = "ACF CO2 Presence in air \n seasonal diff (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "blue")
plot(plot.pacf.seasondiff, main = "PACF CO2 Presence in air \n season difference (1959 - 1997)"
    xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)
```



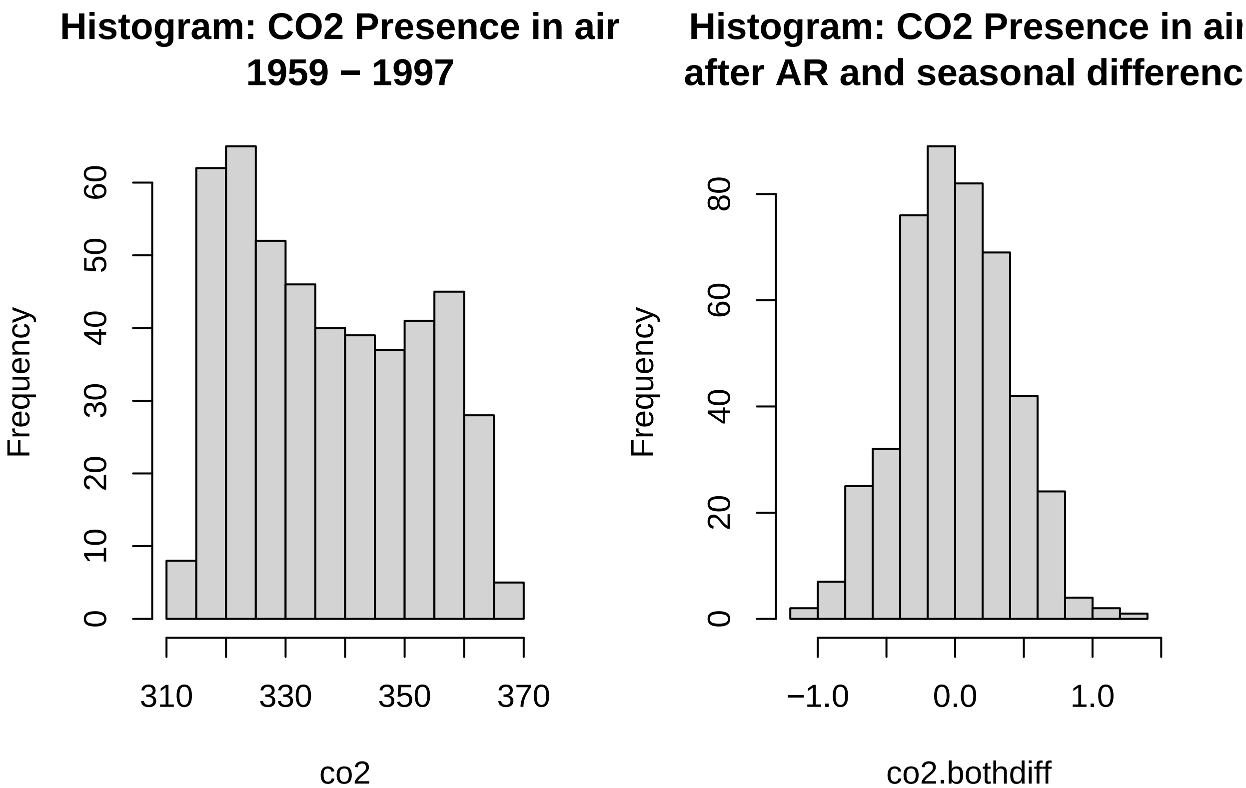

```
plot(plot.acf.bothdiff, main = "ACF CO2 Presence in air \n AR and seasonal differences",
    xlab = "Year", ylab = "Co2 ppm", col = "blue")
plot(plot.pacf.bothdiff, main = "PACF CO2 Presence in air \n AR and seasonal differences",
    xlab = "Year", ylab = "Co2 ppm", col = "red", cex.main = 0.5)
```





Decomposition graph confirms the findings from EDA, trend and seasonality are present int he time series.
Above ACF and PACF graph shows for different adjustments of time series: 1) original series 2) deseasoned 3)detrended 4) random component of time series 5) Two period differenced for trend 5) Two period difference and seasonal differenced time series. Few observations from above graphs
* PACF graph shows autocorrelation dying off at second log after deseasoned. This suggests to use only 1st order Auto regressive model. This also suggests removing seasonality is important
* ACF graph shows clear seasonal effect after removing trend
* ACF graph after performing auto regressive (AR) and seasonal differences looks closer to white noise ACF graph. This confirms the need for seasonal and Integrated treatment for our model

```
par(mfrow = c(1, 2))
hist(co2, main = "Histogram: CO2 Presence in air \n 1959 - 1997")
hist(co2.bothdiff, main = "Histogram: CO2 Presence in air\n after AR and seasonal difference")
```

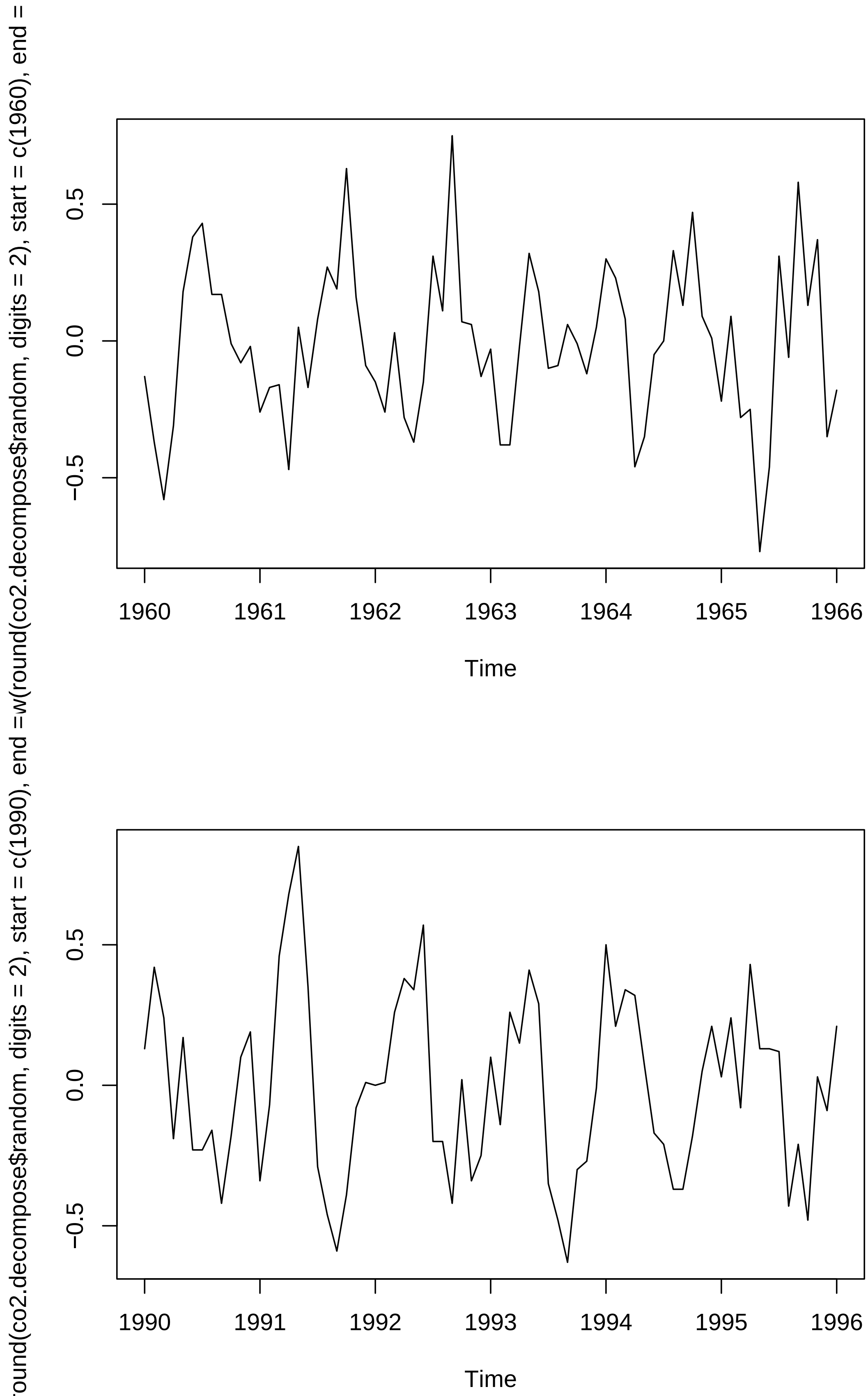


Histogram after applying seasonal and regressive difference looks close to guassian distribution.

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.

```
par(mfrow = c(2, 1))
plot(window(round(co2.decompose$random, digits = 2), start = c(1960),
    end = c(1966)))
plot(window(round(co2.decompose$random, digits = 2), start = c(1990),
    end = c(1996)))
```


w(round(co2.decompose$random, digits = 2), start = c(1990), end =w(round(co2.decompose$random, digits = 2), start = c(1960), end =
0.5
0.0
−0.5
1960
1961
1962
1963
1964
1965
1966
Time
0.5
0.0
−0.5
1990
1991
1992
1993
1994
1995
1996
Time

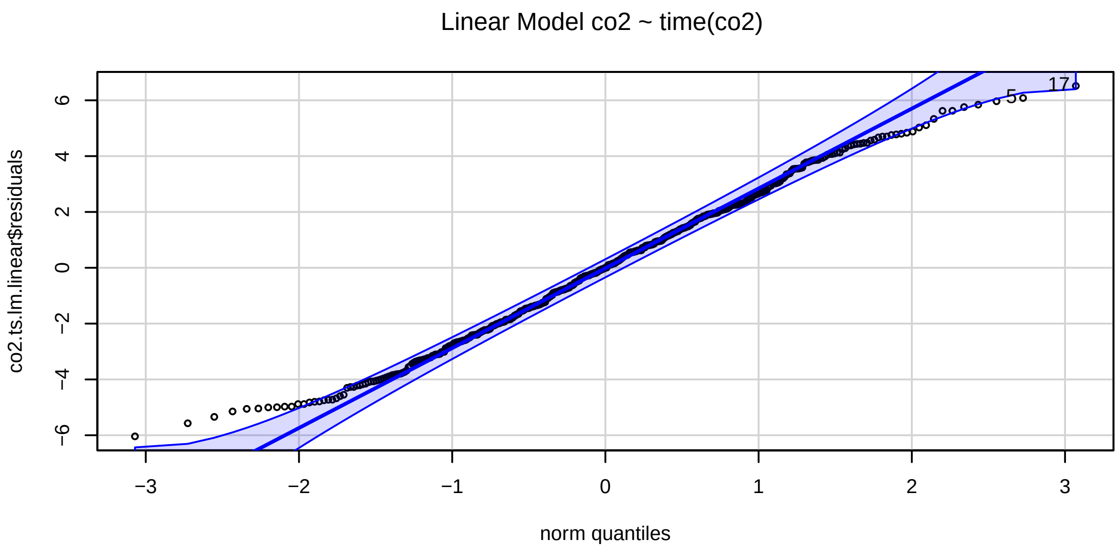
```r
par(mfrow = c(3, 1))
co2.ts.lm.linear = lm(co2 ~ time(co2))
summary(co2.ts.lm.linear)
```

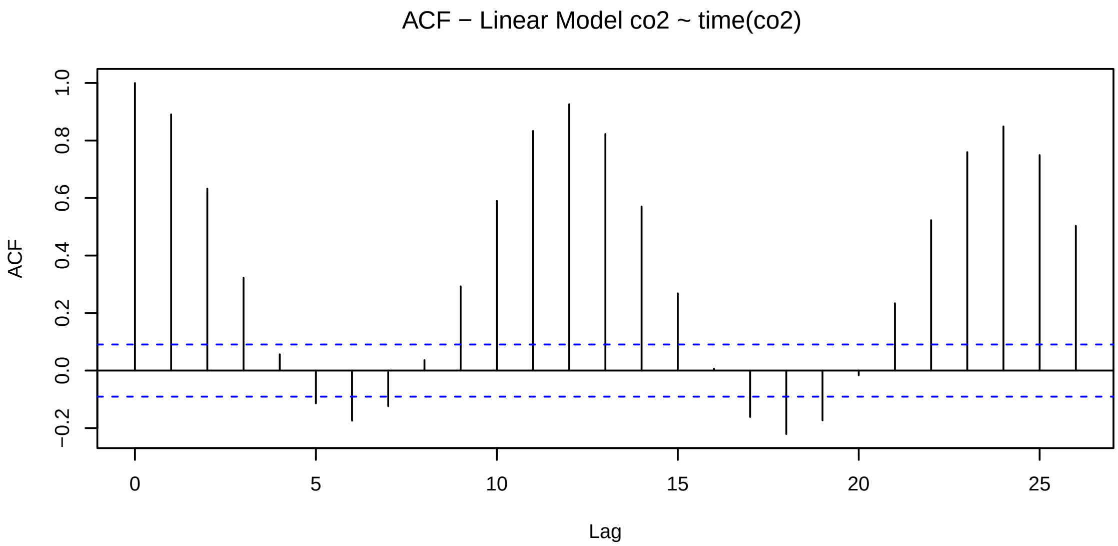
```
##
## Call:
## lm(formula = co2 ~ time(co2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.0399 -1.9476 -0.0017  1.9113  6.5149
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.250e+03  2.127e+01  -105.8   <2e-16 ***
## time(co2)    1.308e+00  1.075e-02   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
```

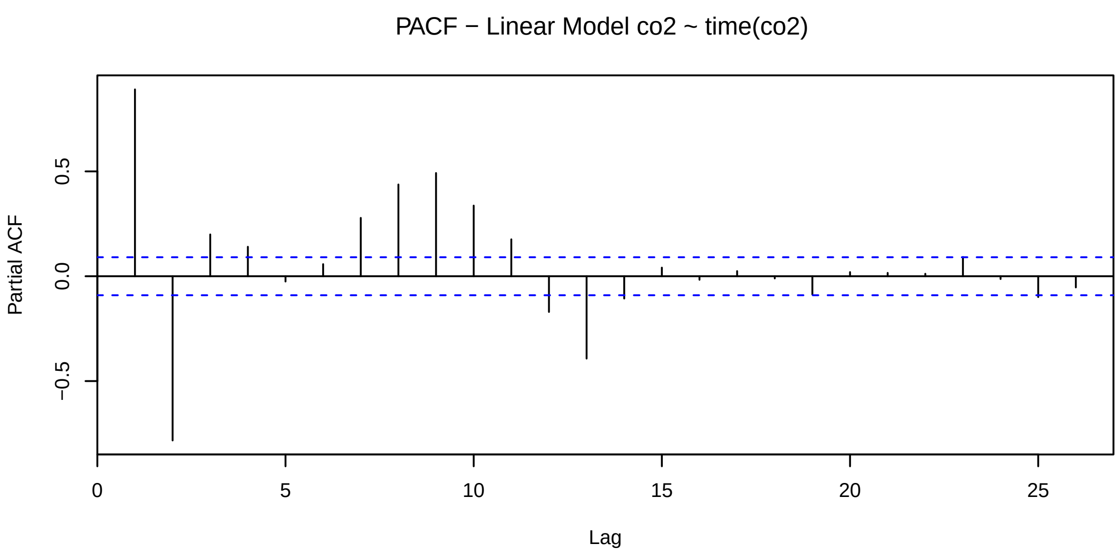
```r
qqPlot(co2.ts.lm.linear$residuals, main = expression("Linear Model co2 ~ time(co2) "))
```

```
## [1] 17  5
```

```r
plt.acf = acf(co2.ts.lm.linear$residuals, plot = FALSE)
plt.pacf = pacf(co2.ts.lm.linear$residuals, plot = FALSE)
plot(plt.acf, main = expression("ACF - Linear Model co2 ~ time(co2) "))
plot(plt.pacf, main = expression("PACF - Linear Model co2 ~ time(co2) "))
```


Linear Model co2 ~ time(co2)
co2.ts.lm.linear$residuals
norm quantiles
17
5


ACF − Linear Model co2 ~ time(co2)
ACF
Lag


PACF − Linear Model co2 ~ time(co2)
Partial ACF
Lag

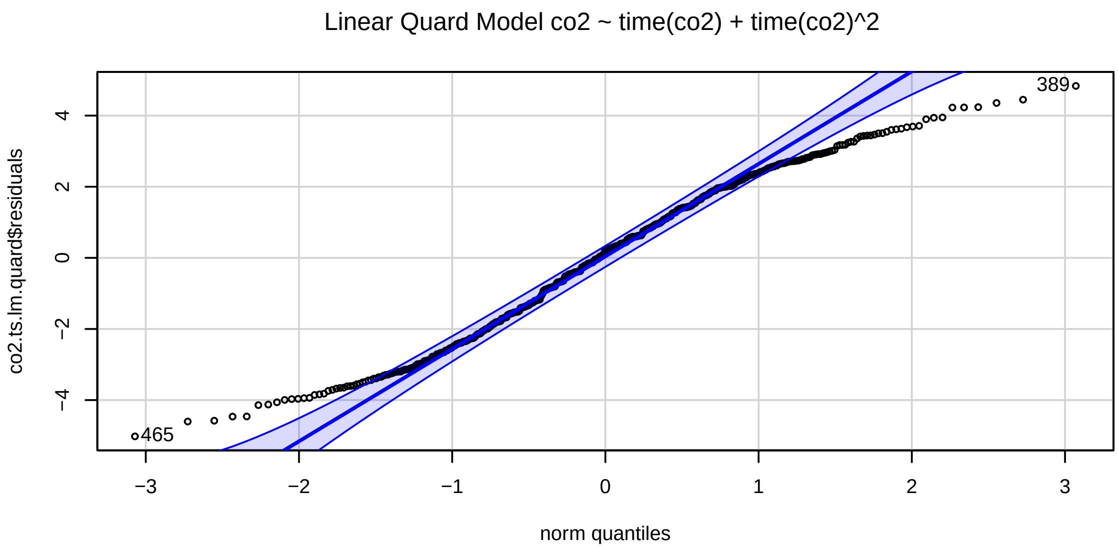
```r
co2.ts.lm.quard = lm(co2 ~ time(co2) + I(time(co2)^2))
summary(co2.ts.lm.quard)
```

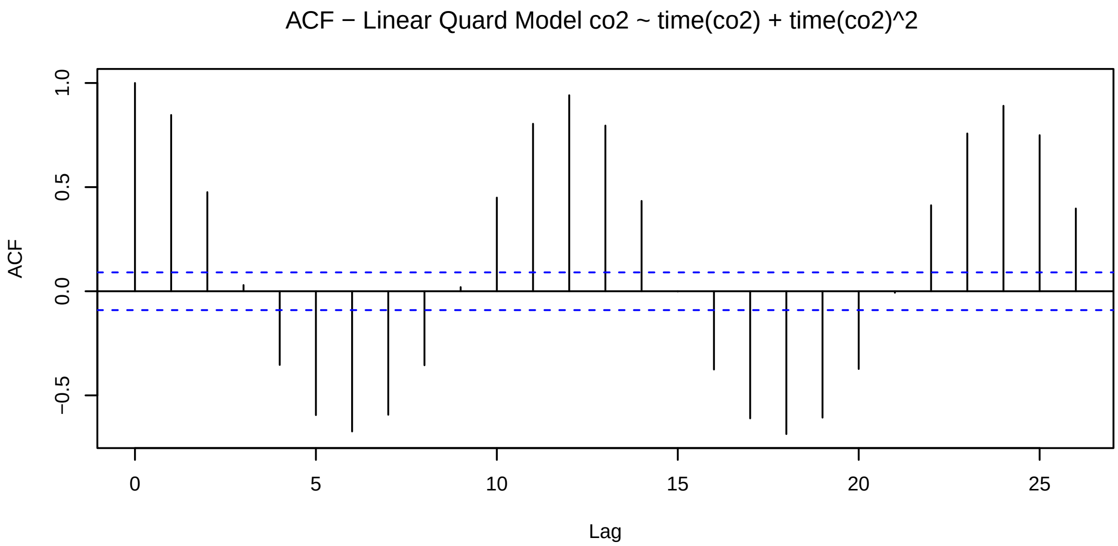
```
##
## Call:
## lm(formula = co2 ~ time(co2) + I(time(co2)^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0195 -1.7120  0.2144  1.7957  4.8345
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.770e+04  3.483e+03   13.70   <2e-16 ***
## time(co2)      -4.919e+01  3.521e+00  -13.97   <2e-16 ***
## I(time(co2)^2)  1.276e-02  8.898e-04   14.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.182 on 465 degrees of freedom
## Multiple R-squared:  0.9788, Adjusted R-squared:  0.9787
## F-statistic: 1.075e+04 on 2 and 465 DF,  p-value: < 2.2e-16
```

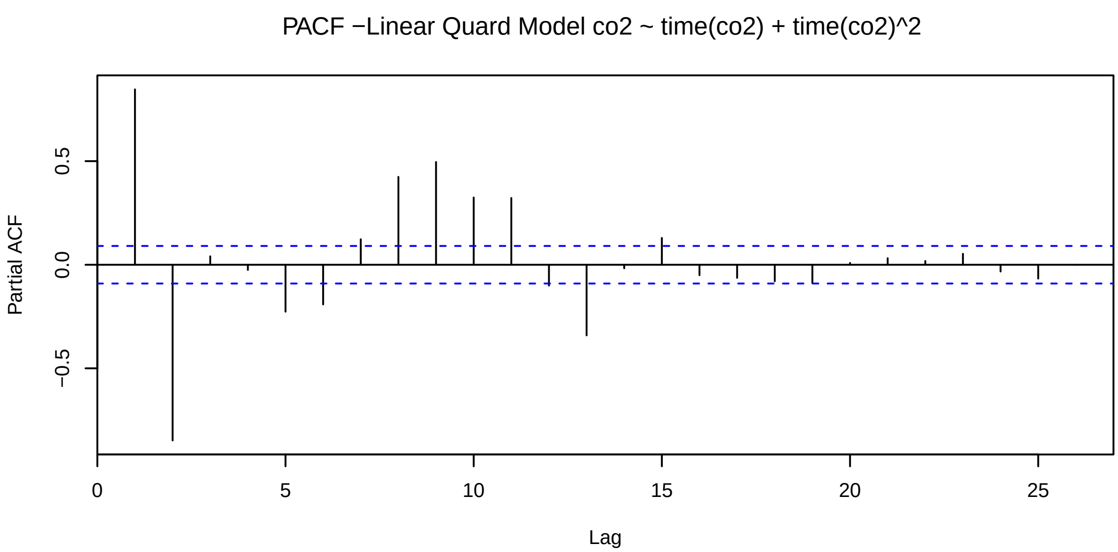
```r
qqPlot(co2.ts.lm.quard$residuals, main = expression("Linear Quard Model co2 ~ time(co2) + time
```

```
## [1] 465 389
```

```r
plt.acf = acf(co2.ts.lm.quard$residuals, plot = FALSE)
plt.pacf = pacf(co2.ts.lm.quard$residuals, plot = FALSE)
plot(plt.acf, main = expression("ACF - Linear Quard Model co2 ~ time(co2) + time(co2)^2 "))
plot(plt.pacf, main = expression("PACF -Linear Quard Model co2 ~ time(co2) + time(co2)^2 "))
```


Linear Quard Model co2 ~ time(co2) + time(co2)^2
co2.ts.lm.quard$residuals
norm quantiles
465
389


ACF − Linear Quard Model co2 ~ time(co2) + time(co2)^2
ACF
Lag


PACF −Linear Quard Model co2 ~ time(co2) + time(co2)^2
Partial ACF
Lag

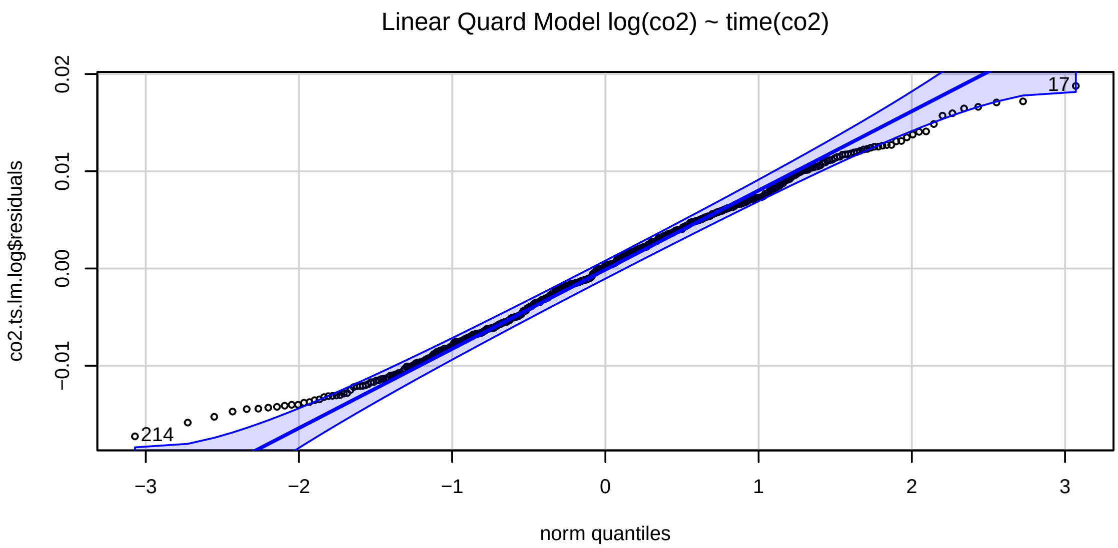
```r
co2.ts.lm.log = lm(log(co2) ~ time(co2))
summary(co2.ts.lm.log)
```

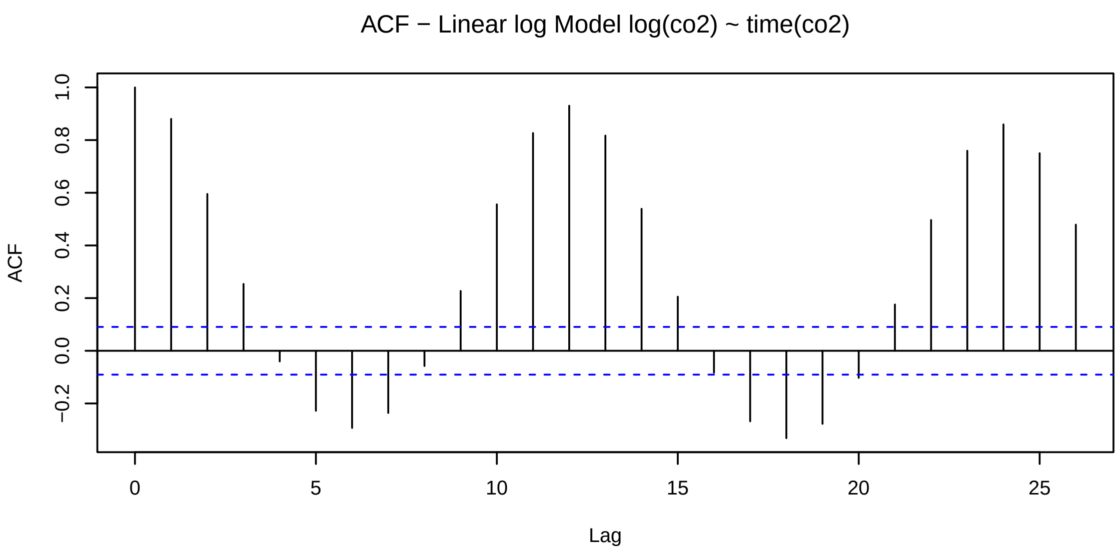
```
##
## Call:
## lm(formula = log(co2) ~ time(co2))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0172650 -0.0056145  0.0002764  0.0053760  0.0187770
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.835e+00  5.991e-02  -30.64   <2e-16 ***
## time(co2)    3.869e-03  3.028e-05  127.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.007375 on 466 degrees of freedom
## Multiple R-squared:  0.9722, Adjusted R-squared:  0.9722
## F-statistic: 1.633e+04 on 1 and 466 DF,  p-value: < 2.2e-16
```

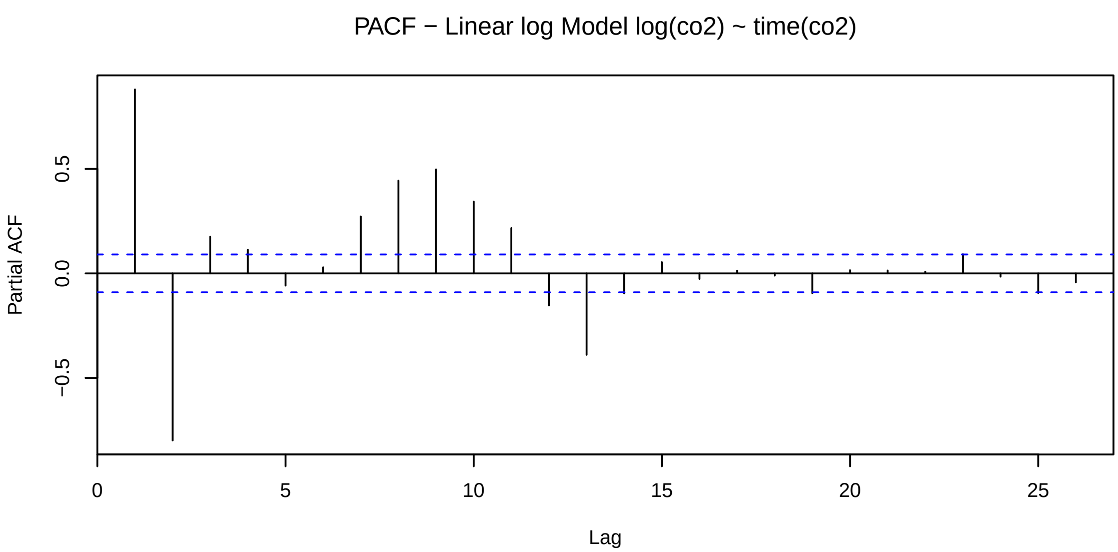
```r
qqPlot(co2.ts.lm.log$residuals, main = expression("Linear Quard Model log(co2) ~ time(co2)"))
```

```
## [1]  17 214
```

```r
plt.acf = acf(co2.ts.lm.log$residuals, plot = FALSE)
plt.pacf = pacf(co2.ts.lm.log$residuals, plot = FALSE)
plot(plt.acf, main = expression("ACF - Linear log Model log(co2) ~ time(co2)"))
plot(plt.pacf, main = expression("PACF - Linear log Model log(co2) ~ time(co2)"))
```


Linear Quard Model log(co2) ~ time(co2)
co2.ts.lm.log$residuals
norm quantiles
214
17
−0.01
0.00
0.01
0.02
−3
−2
−1
0
1
2
3


ACF − Linear log Model log(co2) ~ time(co2)
ACF
Lag
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25


PACF − Linear log Model log(co2) ~ time(co2)
Partial ACF
Lag
−0.5
0.0
0.5
0
5
10
15
20
25

```r
co2.df = data.frame(co2.ppo = c(co2), time = c(time(co2)))
# co2.df$season = as.factor(round(co2.df$time %% 4))
co2.df$season = as.factor(cycle(co2))
head(co2.df, 25)
```

```
##    co2.ppo     time season
## 1   315.42 1959.000      1
## 2   316.31 1959.083      2
## 3   316.50 1959.167      3
## 4   317.56 1959.250      4
## 5   318.13 1959.333      5
## 6   318.00 1959.417      6
## 7   316.39 1959.500      7
## 8   314.65 1959.583      8
## 9   313.68 1959.667      9
## 10  313.18 1959.750     10
## 11  314.66 1959.833     11
## 12  315.43 1959.917     12
## 13  316.27 1960.000      1
## 14  316.81 1960.083      2
## 15  317.42 1960.167      3
## 16  318.87 1960.250      4
## 17  319.87 1960.333      5
## 18  319.43 1960.417      6
## 19  318.01 1960.500      7
## 20  315.74 1960.583      8
## 21  314.00 1960.667      9
## 22  313.68 1960.750     10
## 23  314.84 1960.833     11
## 24  316.03 1960.917     12
## 25  316.73 1961.000      1
```

```r
str(co2.df)
```

```
## 'data.frame':    468 obs. of  3 variables:
##  $ co2.ppo: num  315 316 316 318 318 ...
##  $ time   : num  1959 1959 1959 1959 1959 ...
##  $ season : Factor w/ 12 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
```

```r
co2.ts.lm.quard.season1 = lm(co2 ~ time(co2) + I(time(co2)^2) +
    as.character(cycle(co2)%%4))
summary(co2.ts.lm.quard.season1)
```

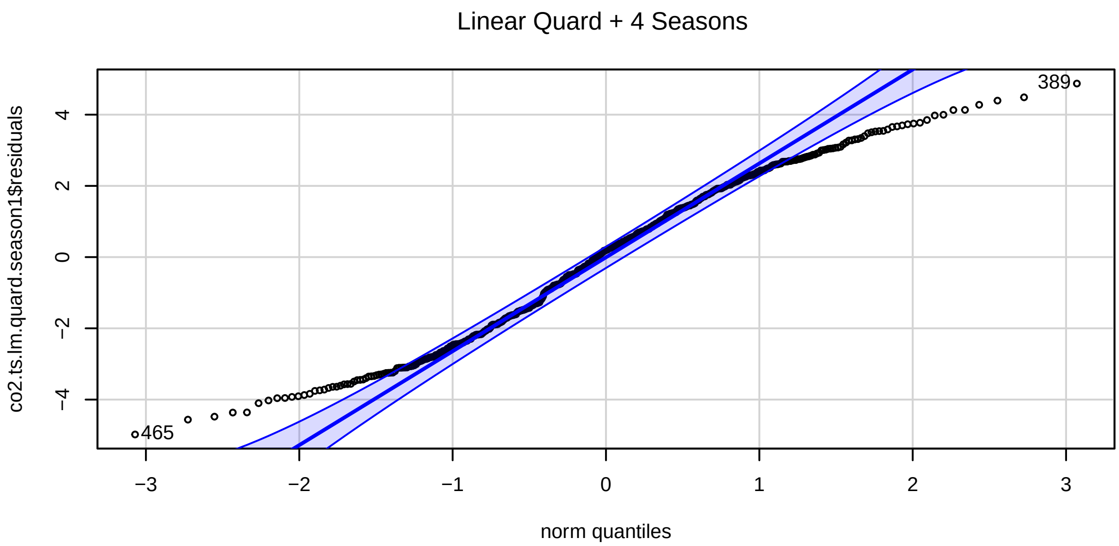
```
##
## Call:
## lm(formula = co2 ~ time(co2) + I(time(co2)^2) + as.character(cycle(co2)%%4))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

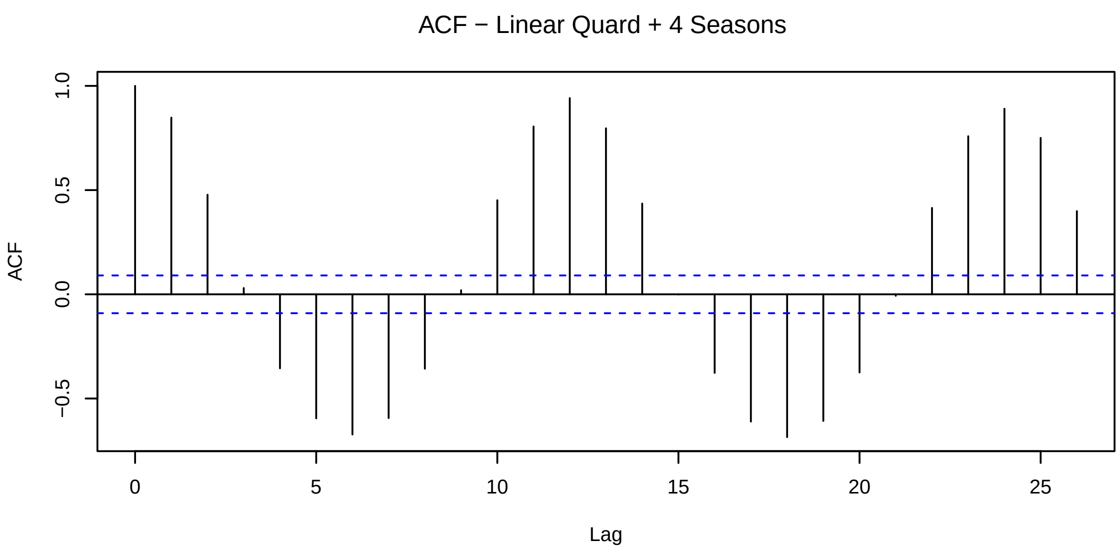
```
## -4.9803 -1.7845  0.1973  1.7723  4.8734
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  4.770e+04  3.492e+03  13.660   <2e-16 ***
## time(co2)                   -4.919e+01  3.530e+00 -13.934   <2e-16 ***
## I(time(co2)^2)               1.276e-02  8.922e-04  14.305   <2e-16 ***
## as.character(cycle(co2)%%4)1 -1.369e-01  2.861e-01  -0.479    0.633
## as.character(cycle(co2)%%4)2 -1.973e-01  2.861e-01  -0.690    0.491
## as.character(cycle(co2)%%4)3 -6.018e-02  2.861e-01  -0.210    0.833
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.188 on 462 degrees of freedom
## Multiple R-squared:  0.9789, Adjusted R-squared:  0.9786
## F-statistic:  4277 on 5 and 462 DF,  p-value: < 2.2e-16
```

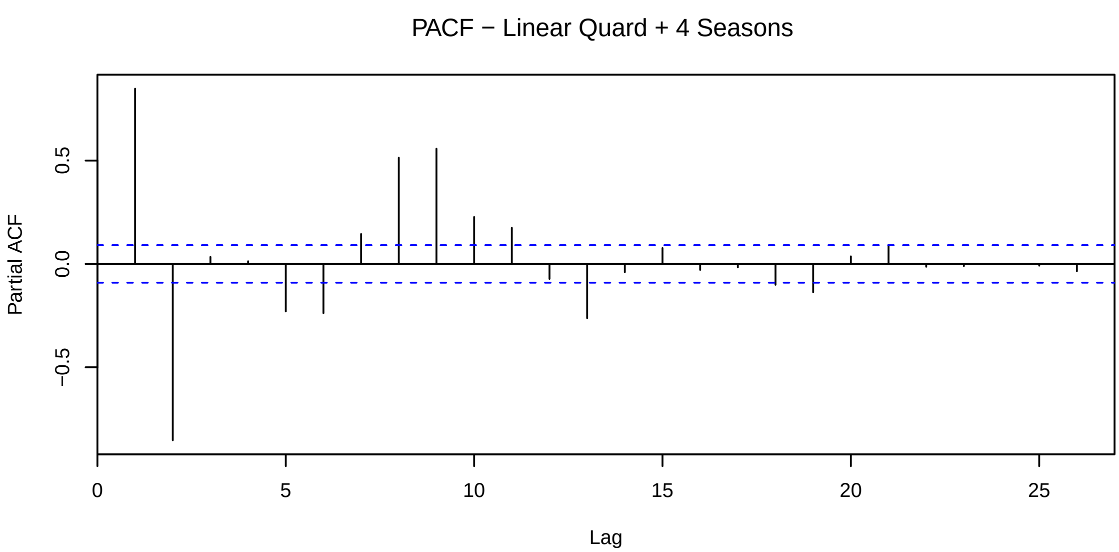
```r
qqPlot(co2.ts.lm.quard.season1$residuals, main = expression("Linear Quard + 4 Seasons "))
```

```
## [1] 465 389
```

```r
plt.acf = acf(co2.ts.lm.quard.season1$residuals, plot = FALSE)
plt.pacf = pacf(co2.ts.lm.quard.season1$residuals, plot = FALSE)
plot(plt.acf, main = expression("ACF - Linear Quard + 4 Seasons "))
plot(plt.pacf, main = expression("PACF - Linear Quard + 4 Seasons "))
```


Linear Quard + 4 Seasons
co2.ts.lm.quard.season1$residuals
norm quantiles
465
389


ACF − Linear Quard + 4 Seasons
ACF
Lag


PACF − Linear Quard + 4 Seasons
Partial ACF
Lag

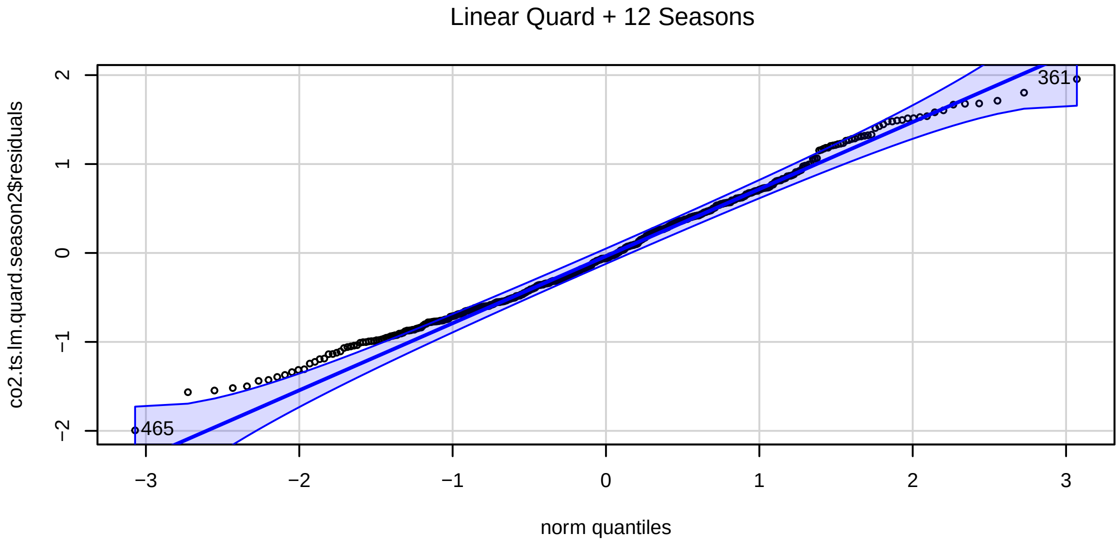
```r
co2.ts.lm.quard.season2 = lm(co2.ppo ~ time + I(time(co2)^2) +
    season, data = co2.df)
summary(co2.ts.lm.quard.season2)
```

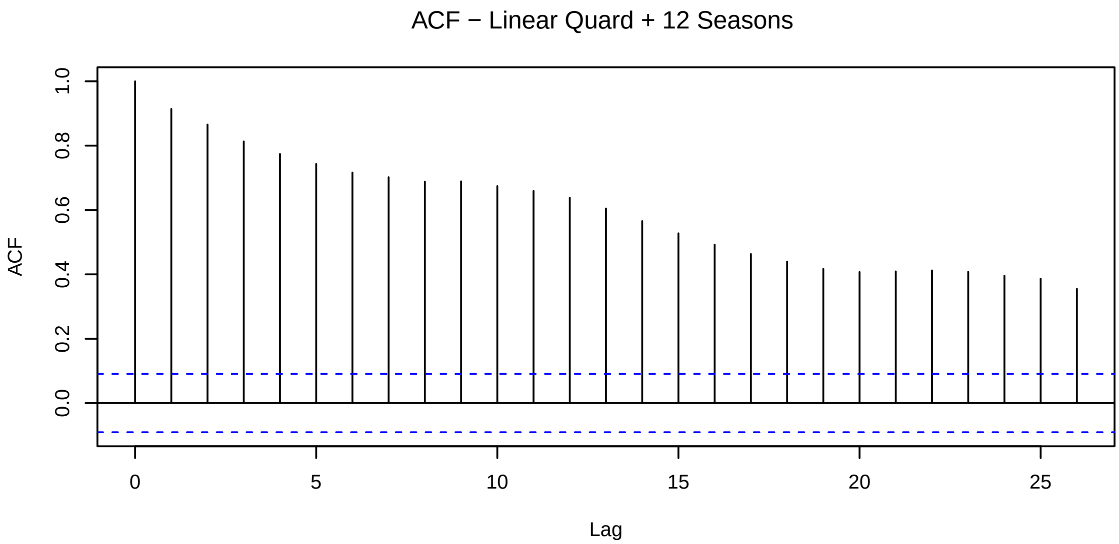
```
##
## Call:
## lm(formula = co2.ppo ~ time + I(time(co2)^2) + season, data = co2.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.99478 -0.54468 -0.06017  0.47265  1.95480
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.771e+04  1.156e+03  41.289  < 2e-16 ***
## time           -4.920e+01  1.168e+00 -42.120  < 2e-16 ***
## I(time(co2)^2)  1.277e-02  2.952e-04  43.242  < 2e-16 ***
## season2         6.642e-01  1.640e-01   4.051 5.99e-05 ***
## season3         1.407e+00  1.640e-01   8.582  < 2e-16 ***
## season4         2.538e+00  1.640e-01  15.480  < 2e-16 ***
## season5         3.017e+00  1.640e-01  18.400  < 2e-16 ***
## season6         2.354e+00  1.640e-01  14.357  < 2e-16 ***
## season7         8.331e-01  1.640e-01   5.081 5.50e-07 ***
## season8        -1.235e+00  1.640e-01  -7.531 2.75e-13 ***
## season9        -3.059e+00  1.640e-01 -18.659  < 2e-16 ***
## season10       -3.243e+00  1.640e-01 -19.777  < 2e-16 ***
## season11       -2.054e+00  1.640e-01 -12.526  < 2e-16 ***
## season12       -9.374e-01  1.640e-01  -5.717 1.97e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.724 on 454 degrees of freedom
## Multiple R-squared:  0.9977, Adjusted R-squared:  0.9977
## F-statistic: 1.531e+04 on 13 and 454 DF,  p-value: < 2.2e-16
```

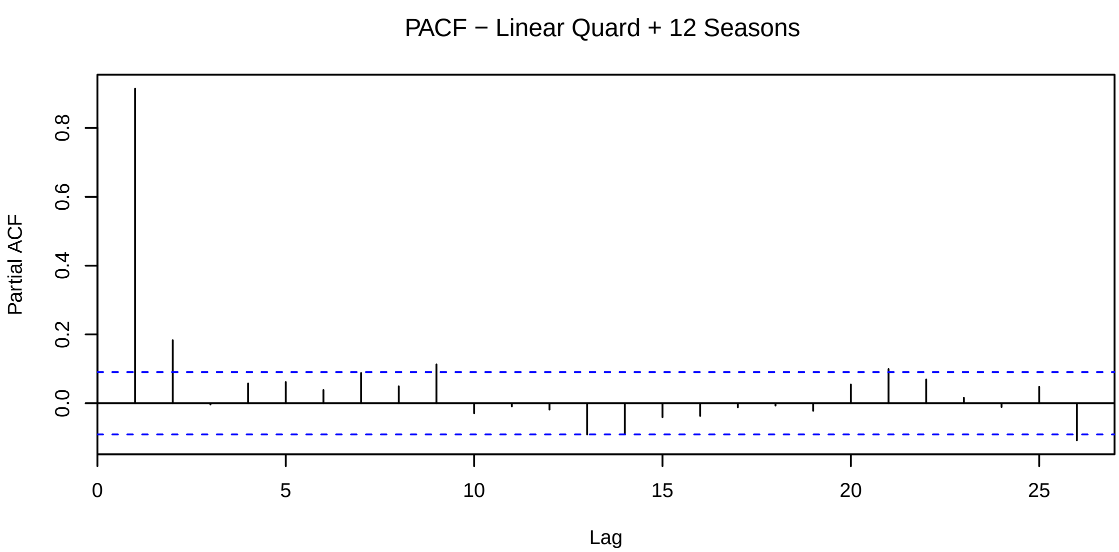
```r
qqPlot(co2.ts.lm.quard.season2$residuals, main = expression("Linear Quard + 12 Seasons "))
```

```
## [1] 465 361
```

```r
plt.acf = acf(co2.ts.lm.quard.season2$residuals, plot = FALSE)
plt.pacf = pacf(co2.ts.lm.quard.season2$residuals, plot = FALSE)
plot(plt.acf, main = expression("ACF - Linear Quard + 12 Seasons "))
plot(plt.pacf, main = expression("PACF - Linear Quard + 12 Seasons "))
```


Linear Quard + 12 Seasons
co2.ts.lm.quard.season2$residuals
norm quantiles
465
361
−3
−2
−1
0
1
2
3


ACF − Linear Quard + 12 Seasons
ACF
Lag
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25


PACF − Linear Quard + 12 Seasons
Partial ACF
Lag
0.0
0.2
0.4
0.6
0.8
0
5
10
15
20
25

```r
Box.test(co2.ts.lm.linear$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  co2.ts.lm.linear$residuals
## X-squared = 373.94, df = 1, p-value < 2.2e-16
```

```r
Box.test(co2.ts.lm.quard$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  co2.ts.lm.quard$residuals
## X-squared = 337.42, df = 1, p-value < 2.2e-16
```

```r
Box.test(co2.ts.lm.log$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  co2.ts.lm.log$residuals
## X-squared = 365.1, df = 1, p-value < 2.2e-16
```

```r
Box.test(co2.ts.lm.quard.season1$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  co2.ts.lm.quard.season1$residuals
## X-squared = 338.26, df = 1, p-value < 2.2e-16
```

```r
Box.test(co2.ts.lm.quard.season2$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  co2.ts.lm.quard.season2$residuals
## X-squared = 393.48, df = 1, p-value < 2.2e-16
```

All linear model shows that we cannot reject the null hypothesis of residual are i.i.d. Our model(s) missed important information in the data and, residuals still have significant autocorrelation.

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.

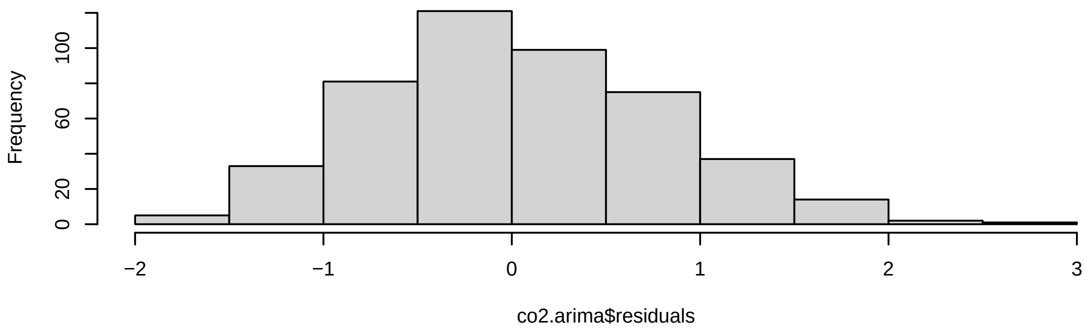
```r
par(mfrow = c(4, 1))

co2.arima = arima(co2, order = c(1, 1, 2))
```

```r
co2.arima
```

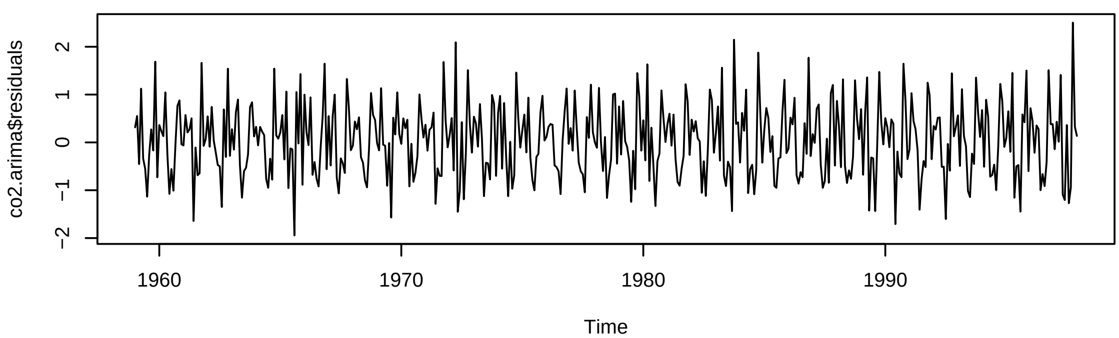
```
##
## Call:
## arima(x = co2, order = c(1, 1, 2))
##
## Coefficients:
##          ar1     ma1     ma2
##       0.4097  0.5626  0.3391
## s.e.  0.0617  0.0620  0.0453
##
## sigma^2 estimated as 0.5771:  log likelihood = -534.84,  aic = 1077.68
```

```r
hist(co2.arima$residuals, main = "ARIMA (1,1,2)")
plot(co2.arima$residuals, main = "ARIMA (1,1,2)")
plt.acf = acf(co2.arima$residuals, plot = FALSE)
plt.pacf = pacf(co2.arima$residuals, plot = FALSE)
plot(plt.acf, main = "ARIMA (1,1,2)")
plot(plt.pacf, main = "ARIMA (1,1,2)")
```

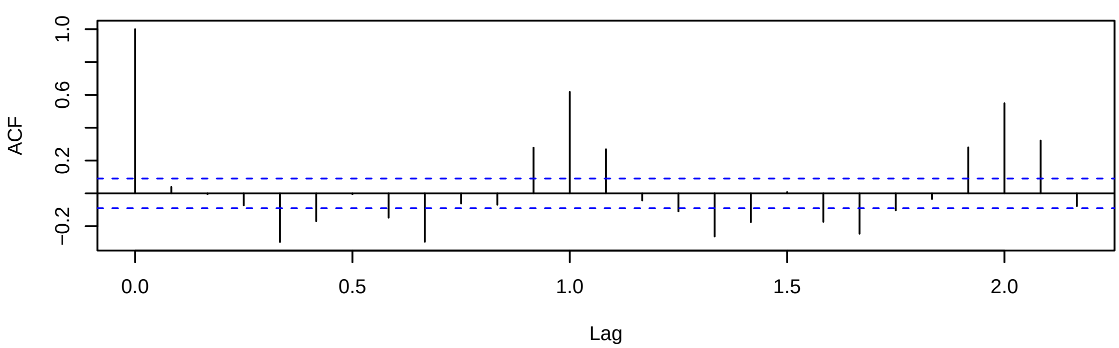
ARIMA (1,1,2)



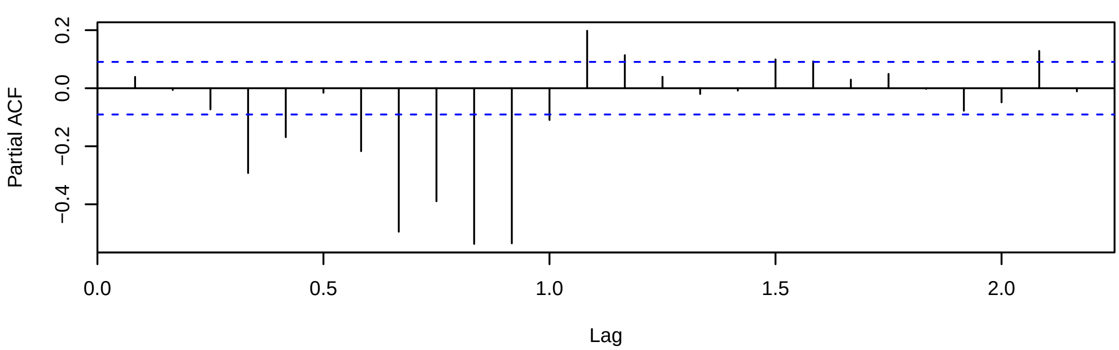
ARIMA (1,1,2)



ARIMA (1,1,2)



ARIMA (1,1,2)



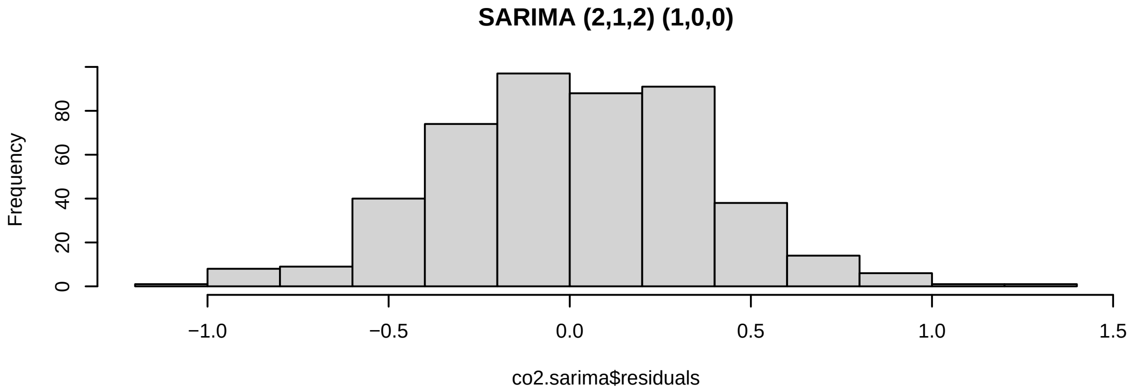
```r
# co2.sarima = arima(co2, order = c(6,1,2), seasonal =
# c(1,0,2))
co2.sarima = arima(co2, order = c(2, 1, 2), seasonal = c(1, 0,
    0), method = "CSS")

co2.sarima
```

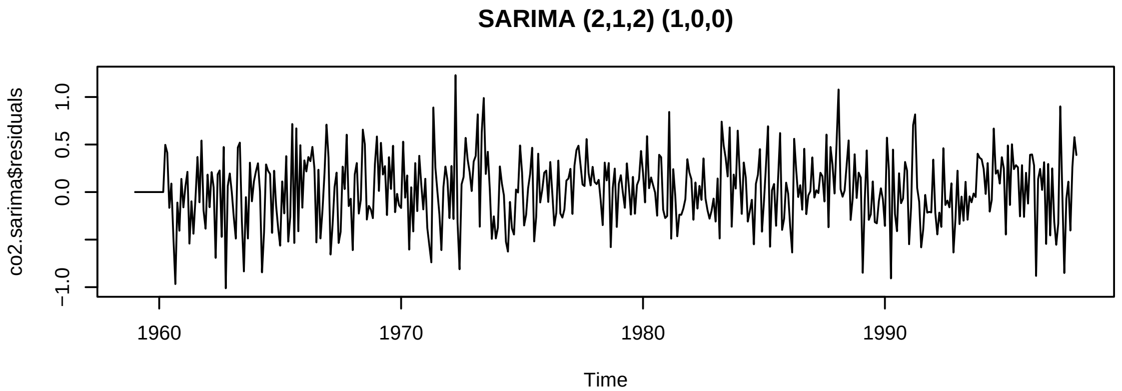
```
##
## Call:
## arima(x = co2, order = c(2, 1, 2), seasonal = c(1, 0, 0), method = "CSS")
##
## Coefficients:
##          ar1     ar2      ma1      ma2    sar1
##       0.1889  0.5038  -0.5322  -0.4318  0.9826
## s.e.  0.1142  0.0878   0.1272   0.1261  0.0141
##
## sigma^2 estimated as 0.132:  part log likelihood = -189.83
```

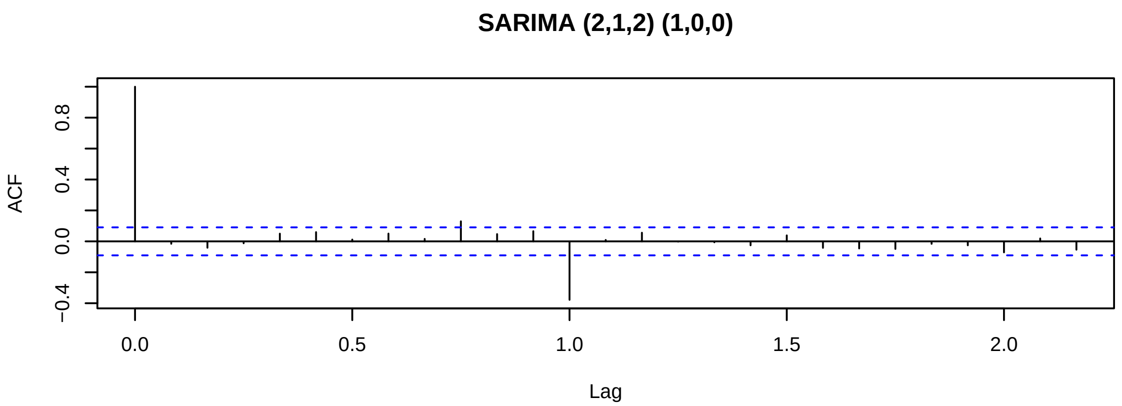
```r
summary(co2.sarima)
```

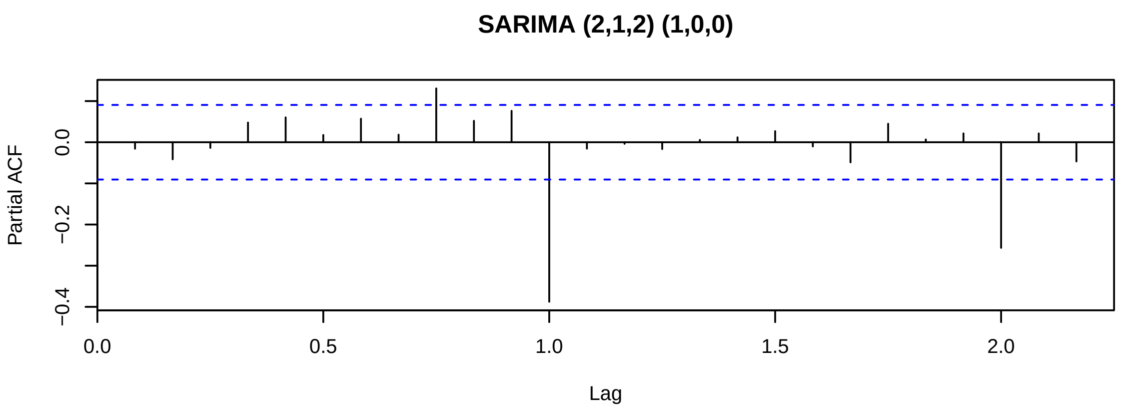
```
##
## Call:
## arima(x = co2, order = c(2, 1, 2), seasonal = c(1, 0, 0), method = "CSS")
##
## Coefficients:
##          ar1     ar2      ma1      ma2    sar1
##       0.1889  0.5038  -0.5322  -0.4318  0.9826
## s.e.  0.1142  0.0878   0.1272   0.1261  0.0141
##
## sigma^2 estimated as 0.132:  part log likelihood = -189.83
##
## Training set error measures:
##                      ME      RMSE       MAE         MPE       MAPE      MASE
## Training set 0.01712798 0.3574549 0.2843781 0.004844505 0.08447418 0.2642246
##                     ACF1
## Training set -0.01570605
```

```r
hist(co2.sarima$residuals, main = "SARIMA (2,1,2) (1,0,0)")
plot(co2.sarima$residuals, main = "SARIMA (2,1,2) (1,0,0)")
plt.acf = acf(co2.sarima$residuals, plot = FALSE)
plt.pacf = pacf(co2.sarima$residuals, plot = FALSE)
plot(plt.acf, main = "SARIMA (2,1,2) (1,0,0)")
plot(plt.pacf, main = "SARIMA (2,1,2) (1,0,0)")
```


SARIMA (2,1,2) (1,0,0)
Frequency
co2.sarima$residuals


SARIMA (2,1,2) (1,0,0)
co2.sarima$residuals
Time


SARIMA (2,1,2) (1,0,0)
ACF
Lag


SARIMA (2,1,2) (1,0,0)
Partial ACF
Lag

```r
co2.auto.arima = auto.arima(co2, trace = TRUE, test = "kpss",
    ic = "bic")
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,1,2)(1,1,1)[12]                    : 240.6967
##  ARIMA(0,1,0)(0,1,0)[12]                    : 470.44
##  ARIMA(1,1,0)(1,1,0)[12]                    : 337.6923
##  ARIMA(0,1,1)(0,1,1)[12]                    : 232.5689
##  ARIMA(0,1,1)(0,1,0)[12]                    : 426.2576
##  ARIMA(0,1,1)(1,1,1)[12]                    : 224.4495
##  ARIMA(0,1,1)(1,1,0)[12]                    : 328.9567
##  ARIMA(0,1,1)(2,1,1)[12]                    : 228.0342
##  ARIMA(0,1,1)(1,1,2)[12]                    : 219.8389
##  ARIMA(0,1,1)(0,1,2)[12]                    : 237.9018
##  ARIMA(0,1,1)(2,1,2)[12]                    : 226.6121
##  ARIMA(0,1,0)(1,1,2)[12]                    : 259.4872
##  ARIMA(1,1,1)(1,1,2)[12]                    : 222.9928
##  ARIMA(0,1,2)(1,1,2)[12]                    : 224.5477
##  ARIMA(1,1,0)(1,1,2)[12]                    : 227.5096
##  ARIMA(1,1,2)(1,1,2)[12]                    : 228.305
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(0,1,1)(1,1,2)[12]                    : 201.7845
##
##  Best model: ARIMA(0,1,1)(1,1,2)[12]
```

```r
co2.auto.arima
```

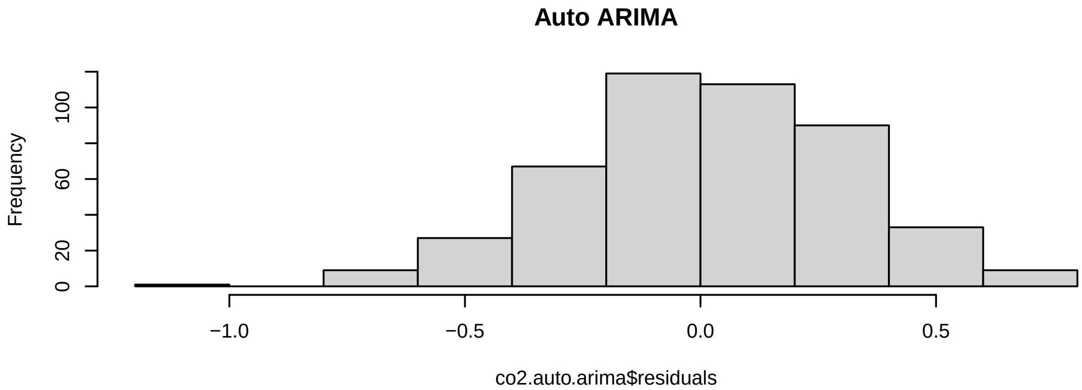
```
## Series: co2
## ARIMA(0,1,1)(1,1,2)[12]
##
## Coefficients:
##           ma1     sar1     sma1     sma2
##       -0.3482  -0.4986  -0.3155  -0.4641
## s.e.   0.0499   0.5284   0.5167   0.4369
##
## sigma^2 estimated as 0.08603:  log likelihood=-85.59
## AIC=181.18   AICc=181.32   BIC=201.78
```

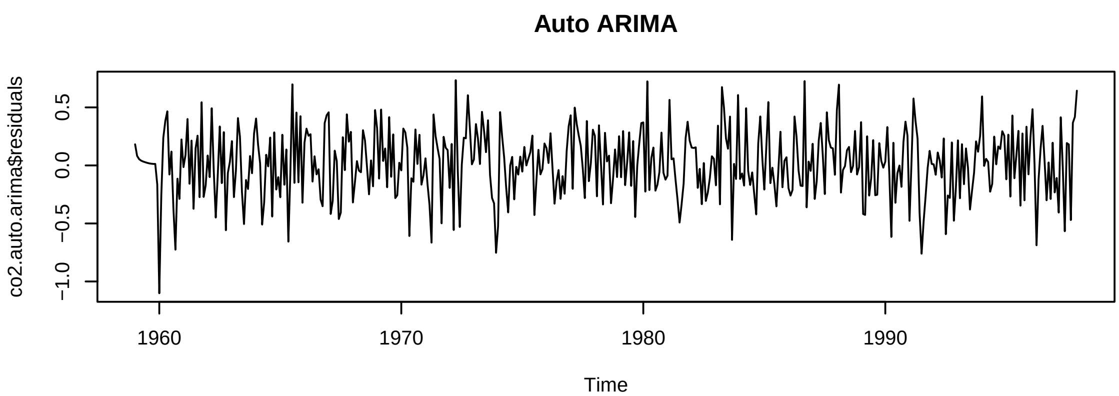
```r
# ARIMA(0,1,1)(1,1,2)
summary(co2.auto.arima)
```

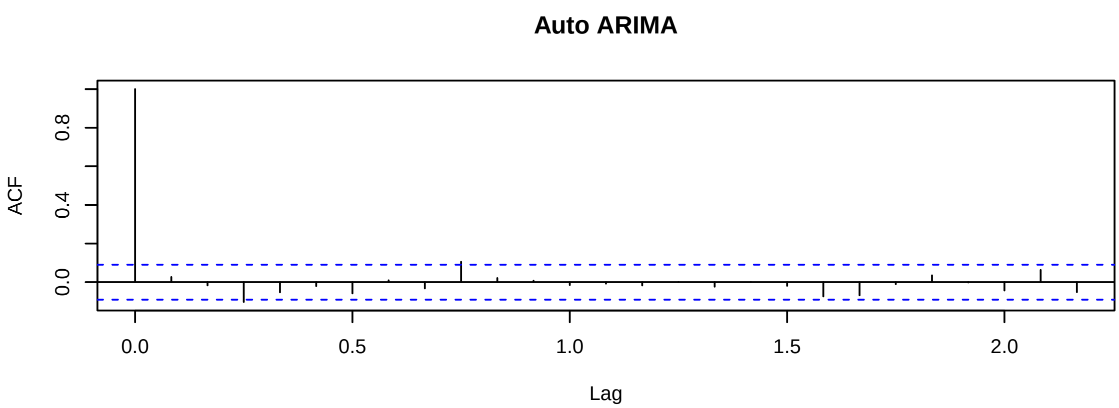
```
## Series: co2
## ARIMA(0,1,1)(1,1,2)[12]
##
## Coefficients:
```

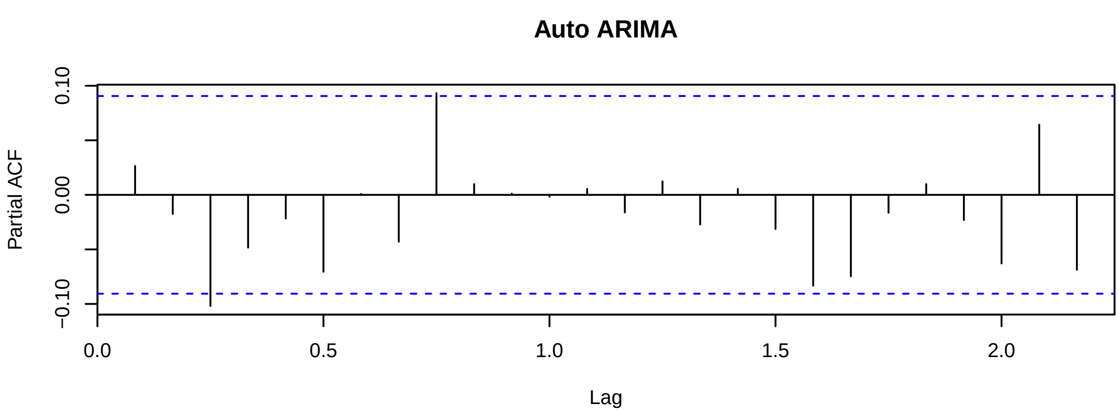
```
##          ma1     sar1     sma1     sma2
##      -0.3482  -0.4986  -0.3155  -0.4641
## s.e.   0.0499   0.5284   0.5167   0.4369
##
## sigma^2 estimated as 0.08603:  log likelihood=-85.59
## AIC=181.18   AICc=181.32   BIC=201.78
##
## Training set error measures:
##                      ME      RMSE       MAE         MPE       MAPE      MASE
## Training set 0.01538153 0.2879337 0.2299909 0.004479982 0.06834581 0.1816409
##                    ACF1
## Training set 0.02645309
```

```
hist(co2.auto.arima$residuals, main = "Auto ARIMA")
plot(co2.auto.arima$residuals, main = "Auto ARIMA")
plt.acf = acf(co2.auto.arima$residuals, plot = FALSE)
plt.pacf = pacf(co2.auto.arima$residuals, plot = FALSE)
plot(plt.acf, main = "Auto ARIMA")
plot(plt.pacf, main = "Auto ARIMA")
```


Auto ARIMA
Frequency
0
20
60
100
−1.0
−0.5
0.0
0.5
co2.auto.arima$residuals


Auto ARIMA
co2.auto.arima$residuals
−1.0
−0.5
0.0
0.5
1960
1970
1980
1990
Time


Auto ARIMA
ACF
0.0
0.4
0.8
0.0
0.5
1.0
1.5
2.0
Lag


Auto ARIMA
Partial ACF
−0.10
0.00
0.10
0.0
0.5
1.0
1.5
2.0
Lag

LJung-BOxtest

```
# test for autocorrelaton of residuals augment(co2.arima)
# %>% features(.resid, ljung_box)

# # inverse roots within unit circle gg_arma(series1_model)
# # modulus of roots exceed unity Mod(polyroot(c(1,
# -coef(series1_model)[['estimate']])))


Box.test(co2.arima$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  co2.arima$residuals
## X-squared = 0.72041, df = 1, p-value = 0.396
```

```
Box.test(co2.sarima$residuals, type = "Ljung-Box")
```

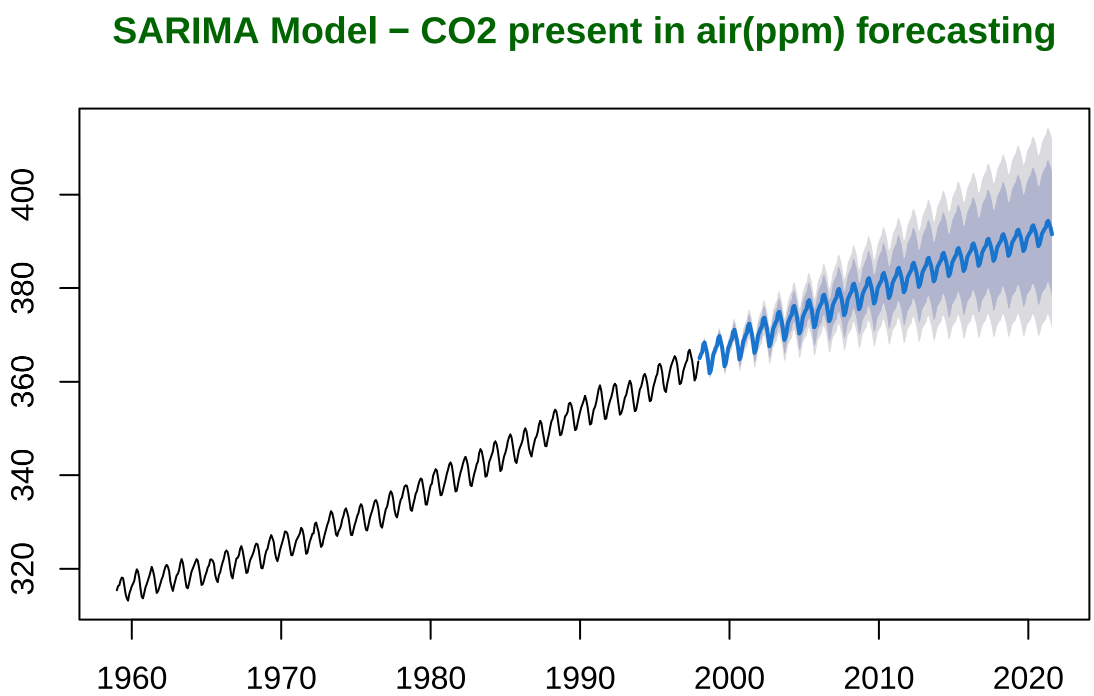
```
##
##  Box-Ljung test
##
## data:  co2.sarima$residuals
## X-squared = 0.11619, df = 1, p-value = 0.7332
```

```
Box.test(co2.auto.arima$residuals, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  co2.auto.arima$residuals
## X-squared = 0.32959, df = 1, p-value = 0.5659
```

###Forecast for next 6 months

```
co2.forecast <- forecast(co2.sarima, 284)
co2.forecast.summary = summary(co2.forecast)
plot(co2.forecast, main = "SARIMA Model - CO2 present in air(ppm) forecasting",
    col.main = "darkgreen")
```



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 predicitons 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.

```
# Custom funciton to ignore multiple spaces
txt.custom.read = function(file, skip.rows = NULL, sep) {
    if (!is.null(skip.rows)) {
        tmp = readLines(file)
        tmp = tmp[-(skip.rows)]
    }

    tmpFile = tempfile()
    on.exit(unlink(tmpFile))
    tmp = gsub(" +", x = tmp, replacement = ",", perl = TRUE)
    writeLines(tmp, tmpFile)
    file = tmpFile

    result = read.csv(file, sep = ",", header = FALSE)
    return(result)
}

co2.noaa.df = txt.custom.read("co2_weekly_mlo.txt", skip = (1:49),
    sep = " ")
```

```r
colnames(co2.noaa.df) <- c("blank", "year", "month", "day", "week",
    "ppm", "days", "1yrago", "10yearago", "since1800")

# Remove blank column
co2.noaa.df <- subset(co2.noaa.df, select = -c(blank))

head(co2.noaa.df)
```

```
##   year month day     week    ppm days  1yrago 10yearago since1800
## 1 1974     5  19 1974.380 333.37    5 -999.99   -999.99     50.40
## 2 1974     5  26 1974.399 332.95    6 -999.99   -999.99     50.06
## 3 1974     6   2 1974.418 332.35    5 -999.99   -999.99     49.60
## 4 1974     6   9 1974.437 332.20    7 -999.99   -999.99     49.65
## 5 1974     6  16 1974.456 332.37    7 -999.99   -999.99     50.06
## 6 1974     6  23 1974.475 331.73    5 -999.99   -999.99     49.72
```

```r
summary(co2.noaa.df)
```

```
##       year          month             day             week     
##  Min.   :1974   Min.   : 1.00   Min.   : 1.00   Min.   :1974  
##  1st Qu.:1986   1st Qu.: 4.00   1st Qu.: 8.00   1st Qu.:1986  
##  Median :1997   Median : 7.00   Median :16.00   Median :1998  
##  Mean   :1997   Mean   : 6.52   Mean   :15.72   Mean   :1998  
##  3rd Qu.:2009   3rd Qu.:10.00   3rd Qu.:23.00   3rd Qu.:2010  
##  Max.   :2021   Max.   :12.00   Max.   :31.00   Max.   :2021  
##       ppm                days           1yrago          10yearago      
##  Min.   :-1000.0   Min.   :0.000   Min.   :-1000.0   Min.   : -999.99  
##  1st Qu.:  347.1   1st Qu.:5.000   1st Qu.:  345.6   1st Qu.:  331.48  
##  Median :  365.2   Median :6.000   Median :  363.5   Median :  350.18  
##  Mean   :  358.3   Mean   :5.871   Mean   :  328.4   Mean   :   59.61  
##  3rd Qu.:  388.4   3rd Qu.:7.000   3rd Qu.:  386.2   3rd Qu.:  368.45  
##  Max.   :  420.0   Max.   :7.000   Max.   :  417.8   Max.   :  395.23  
##    since1800     
##  Min.   : -999.99  
##  1st Qu.:   66.95  
##  Median :   84.55  
##  Mean   :   80.38  
##  3rd Qu.:  108.07  
##  Max.   :  136.87
```

```r
str(co2.noaa.df)
```

```
## 'data.frame':    2458 obs. of  9 variables:
##  $ year     : int  1974 1974 1974 1974 1974 1974 1974 1974 1974 1974 ...
##  $ month    : int  5 5 6 6 6 6 6 7 7 7 ...
##  $ day      : int  19 26 2 9 16 23 30 7 14 21 ...
##  $ week     : num  1974 1974 1974 1974 1974 ...
##  $ ppm      : num  333 333 332 332 332 ...
##  $ days     : int  5 6 5 7 7 5 6 6 5 7 ...
```

```
##  $ 1yrago   : num  -1000 -1000 -1000 -1000 -1000 ...
##  $ 10yearago: num  -1000 -1000 -1000 -1000 -1000 ...
##  $ since1800: num  50.4 50.1 49.6 49.6 50.1 ...
```

```
describe(co2.noaa.df)
```

```
## co2.noaa.df
##
##  9  Variables      2458  Observations
## --------------------------------------------------------------------------------
## year
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0       48        1     1997    15.71     1976     1979
##      .25      .50      .75      .90      .95
##     1986     1997     2009     2016     2019
##
## lowest : 1974 1975 1976 1977 1978, highest: 2017 2018 2019 2020 2021
## --------------------------------------------------------------------------------
## month
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0       12    0.993     6.52    3.965        1        2
##      .25      .50      .75      .90      .95
##        4        7       10       11       12
##
## lowest :  1  2  3  4  5, highest:  8  9 10 11 12
##
## Value          1     2     3     4     5     6     7     8     9    10    11
## Frequency    208   190   208   201   211   205   208   208   202   207   202
## Proportion 0.085 0.077 0.085 0.082 0.086 0.083 0.085 0.085 0.082 0.084 0.082
##
## Value         12
## Frequency    208
## Proportion 0.085
## --------------------------------------------------------------------------------
## day
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0       31    0.999    15.72    10.16        2        4
##      .25      .50      .75      .90      .95
##        8       16       23       28       29
##
## lowest :  1  2  3  4  5, highest: 27 28 29 30 31
## --------------------------------------------------------------------------------
## week
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0     2458        1     1998    15.71     1977     1979
##      .25      .50      .75      .90      .95
##     1986     1998     2010     2017     2019
##
```

```
## lowest : 1974.380 1974.399 1974.418 1974.437 1974.456
## highest: 2021.390 2021.410 2021.429 2021.448 2021.467
## ---------------------------------------------------------------------------
## ppm
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0     2148        1    358.3    47.87    332.4    336.1
##      .25      .50      .75      .90      .95
##    347.1    365.2    388.4    404.6    410.6
##
## lowest : -999.99  326.72  326.99  327.07  327.23
## highest:  419.28  419.47  419.53  419.55  420.01
##
## Value      -1000   320   340   360   380   400   420
## Frequency     18    45   638   662   527   435   133
## Proportion 0.007 0.018 0.260 0.269 0.214 0.177 0.054
##
## For the frequency table, variable is rounded to the nearest 20
## ---------------------------------------------------------------------------
## days
##        n  missing distinct     Info     Mean      Gmd
##     2458        0        8    0.896    5.871    1.378
##
## lowest : 0 1 2 3 4, highest: 3 4 5 6 7
##
## Value          0     1     2     3     4     5     6     7
## Frequency     18    14    36   101   176   402   648  1063
## Proportion 0.007 0.006 0.015 0.041 0.072 0.164 0.264 0.432
## ---------------------------------------------------------------------------
## 1yrago
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0     2097        1    328.4    101.7    330.5    334.4
##      .25      .50      .75      .90      .95
##    345.6    363.5    386.2    402.0    408.2
##
## lowest : -999.99  326.73  326.84  326.98  327.21
## highest:  417.09  417.10  417.21  417.46  417.83
##
## Value      -1000   320   340   360   380   400   420
## Frequency     70    45   638   665   523   436    81
## Proportion 0.028 0.018 0.260 0.271 0.213 0.177 0.033
##
## For the frequency table, variable is rounded to the nearest 20
## ---------------------------------------------------------------------------
## 10yearago
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0     1644    0.989    59.61    479.1  -1000.0  -1000.0
##      .25      .50      .75      .90      .95
##    331.5    350.2    368.5    382.4    387.0
```

```
##
## lowest : -999.99  326.66  327.04  327.10  327.26
## highest:  394.08  394.15  394.43  395.13  395.23
##
## Value      -1000   330   340   350   360   370   380   390   400
## Frequency    541   196   328   343   339   286   248   175     2
## Proportion 0.220 0.080 0.133 0.140 0.138 0.116 0.101 0.071 0.001
##
## For the frequency table, variable is rounded to the nearest 10
## --------------------------------------------------------------------------------
## since1800
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     2458        0     2086        1    80.38    43.66    52.11    55.81
##      .25      .50      .75      .90      .95
##    66.95    84.55   108.07   125.10   130.75
##
## lowest : -999.99   49.60   49.65   49.72   49.95
## highest:  136.49  136.61  136.64  136.74  136.87
##
## Value      -1000    50    60    70    80    90   100   110   120   130   140
## Frequency     18   194   326   325   371   270   260   245   200   216    33
## Proportion 0.007 0.079 0.133 0.132 0.151 0.110 0.106 0.100 0.081 0.088 0.013
##
## For the frequency table, variable is rounded to the nearest 10
## --------------------------------------------------------------------------------
```

NOAA data provided in the file has 2458 weekly observations from 1974 to 2021 with 10 variables. Variable `PPM` tracks weekly co2 presence. We will be using `PPM` values for our analysis. Author uses -999 as missing value and we have 18 observation that have `PPM` value as a null, we will fill them in before developing time series model.

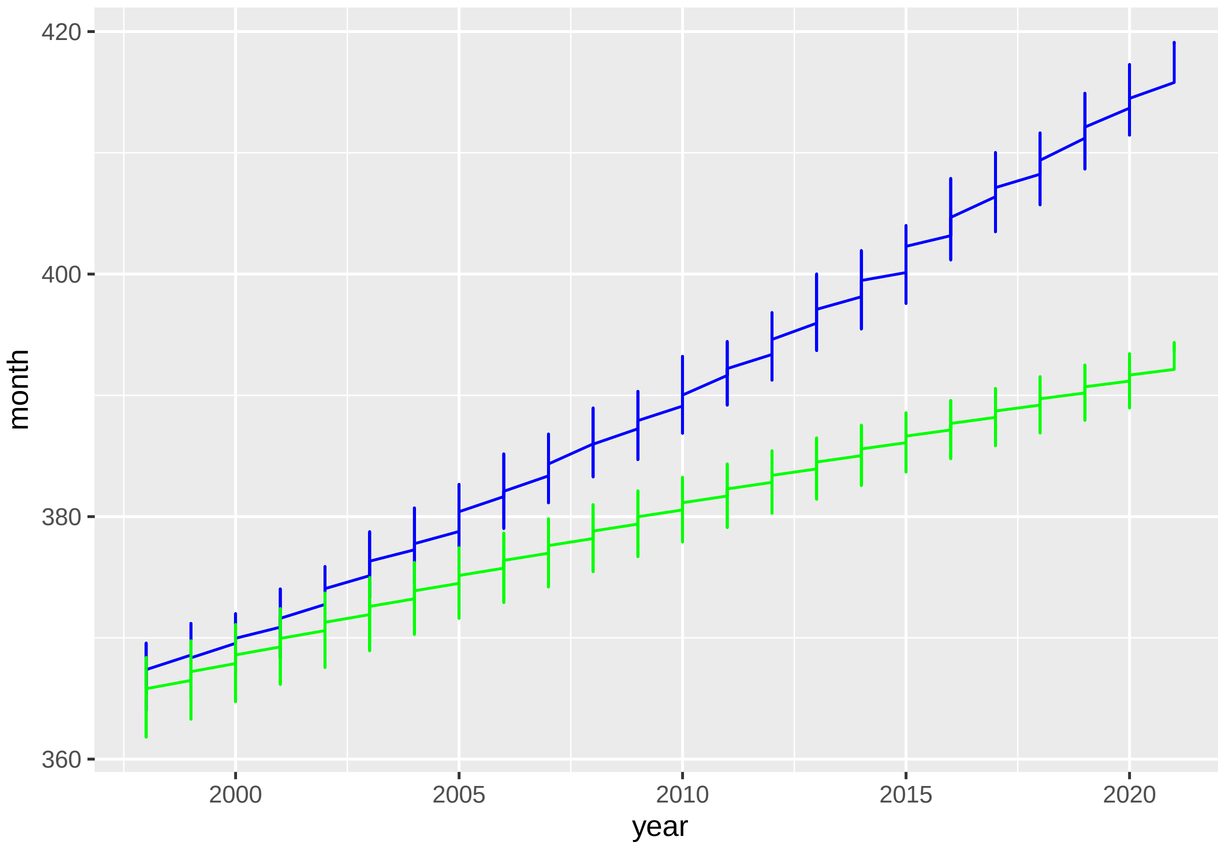
```
# Get monthly averages for replacement after removing NA or
# -999
co2.noaa.month.df <- co2.noaa.df %>%
    filter(ppm > 0) %>%
    group_by(year, month) %>%
    summarise(ppm_month_avg = mean(ppm))

# join to add monthly averages
co2.noaa.imputed.df <- merge(co2.noaa.df, co2.noaa.month.df,
    by = c("year", "month"))

# Calculate imputed value
co2.noaa.imputed.df <- co2.noaa.imputed.df %>%
    mutate(ppm_imputed = ifelse(test = (ppm <= 0), ppm_month_avg,
        no = ppm))


co2.noaa.ts <- ts(co2.noaa.imputed.df$ppm_imputed, start = c(1959),
```

```r
    frequency = 52)

par(mfrow = c(2, 1))

plot(co2.noaa.ts, main = "With imputed values for missing vales Weekly series
           CO2 Presence in air (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "blue", cex.main = 0.5)


# Calculate monthly averages as our forecast is only on
# monthly basis
co2.noaa.month.imputed.df <- co2.noaa.imputed.df %>%
    group_by(year, month) %>%
    summarise(ppm_month_avg = mean(ppm_imputed))

summary(co2.noaa.month.imputed.df)
```

```
##       year          month        ppm_month_avg
##  Min.   :1974   Min.   : 1.000   Min.   :327.3
##  1st Qu.:1986   1st Qu.: 4.000   1st Qu.:347.5
##  Median :1997   Median : 6.000   Median :365.2
##  Mean   :1997   Mean   : 6.487   Mean   :368.3
##  3rd Qu.:2009   3rd Qu.: 9.000   3rd Qu.:388.2
##  Max.   :2021   Max.   :12.000   Max.   :419.1
```

```r
co2.noaa.month.ts <- ts(co2.noaa.imputed.df$ppm_imputed, start = c(1959),
    frequency = 12)

autoplot(co2.noaa.month.ts, main = "NOAA data With imputed values for missing vales
         Monthly series\n CO2 Presence in air (1959 - 1997)",
    xlab = "Year", ylab = "Co2 ppm", col = "blue")


# transforming time series data to dataframe, so that we
# can join
co2.forecast.df <- data.frame(floor(as.numeric(time(co2.forecast.summary[4]$mean))),
    cycle(time(co2.forecast.summary[4]$mean)), co2.forecast.summary[4]$mean)

colnames(co2.forecast.df) <- c("year", "month", "ppm.forecast")

co2.noaa.forecast.merged <- merge(co2.noaa.month.imputed.df,
    co2.forecast.df, all.x = TRUE)

co2.noaa.forecast.merged <- co2.noaa.forecast.merged %>%
    mutate(year.month = paste(year, ".", month))

forecst.df.filtered = co2.noaa.forecast.merged %>%
    filter(year > 1997)
```

```r
ggplot(data = forecst.df.filtered, aes(x = year, month)) + geom_line(aes(y = ppm_month_avg),
    colour = "blue") + geom_line(aes(y = ppm.forecast), colour = "green")


sarima.forecast = predict(object = co2.sarima, new_data = co2.noaa.month.imputed.df)
```



In the above code, we imputed missing values by using monthly average for that period and above graph looks good with imputed values

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 (psuedo-) 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?

```r
# autoplot(flight.prices.training, freq) +
# autolayer(flight.prices.test, freq, colour = 'red')
```