# Library loading

```
options(warn=-1)
library(tidyverse)
library(forecast)
library(tseries)
```

# Loading data from the Mauna Loa observatory

Let's first load the data. From the comments included in this dataset, the original columns are: > Year, month, decimal date, monthly CO2 average, deseasonalized monthly CO2 average, number of days in month, std. dev of days, unc. of monthly mean

For this analysis, we are only interested in the year, month and CO2 concentration.

```
co2_monthly_full <- read.delim(file = 'co2_mm_mlo.txt', comment.char = '#', header = F, sep = '')
co2_monthly_full <- co2_monthly_full[, c(1, 2, 4)]
names(co2_monthly_full) <- c('year', 'month', 'co2_concentration')
head(co2_monthly_full)
```

```
##   year month co2_concentration
## 1 1958     3            315.70
## 2 1958     4            317.45
## 3 1958     5            317.51
## 4 1958     6            317.24
## 5 1958     7            315.86
## 6 1958     8            314.93
```

# Data cleanup and transformations

Let's first check if the dataset contains null values

```
which(is.na(co2_monthly_full))
```

```
## integer(0)
```

Perfect, we don't have any null data. This is because the months in the downloaded dataset had already been interpolated.

Readings for years 1958 and 2022 will be removed since they do not have the 12 months included.

```
co2_monthly_full <- co2_monthly_full %>% filter(year != 2022 & year != 1958)
```

We will also create a single date field based on month and year, which should improve the readability of our plots in the future.

```
co2_monthly_full$date <- as.Date(paste(co2_monthly_full$year, co2_monthly_full$month, 1, sep = '-'), fo
co2_monthly <- co2_monthly_full[, c('date', 'co2_concentration')]
head(co2_monthly)
```

```
##         date co2_concentration
## 1 1959-01-01            315.58
## 2 1959-02-01            316.48
## 3 1959-03-01            316.65
## 4 1959-04-01            317.72
## 5 1959-05-01            318.29
## 6 1959-06-01            318.15
```

Finally, we'll transform the dataframe into a time series object, which enables us to use some helpful R functions further in the analysis.

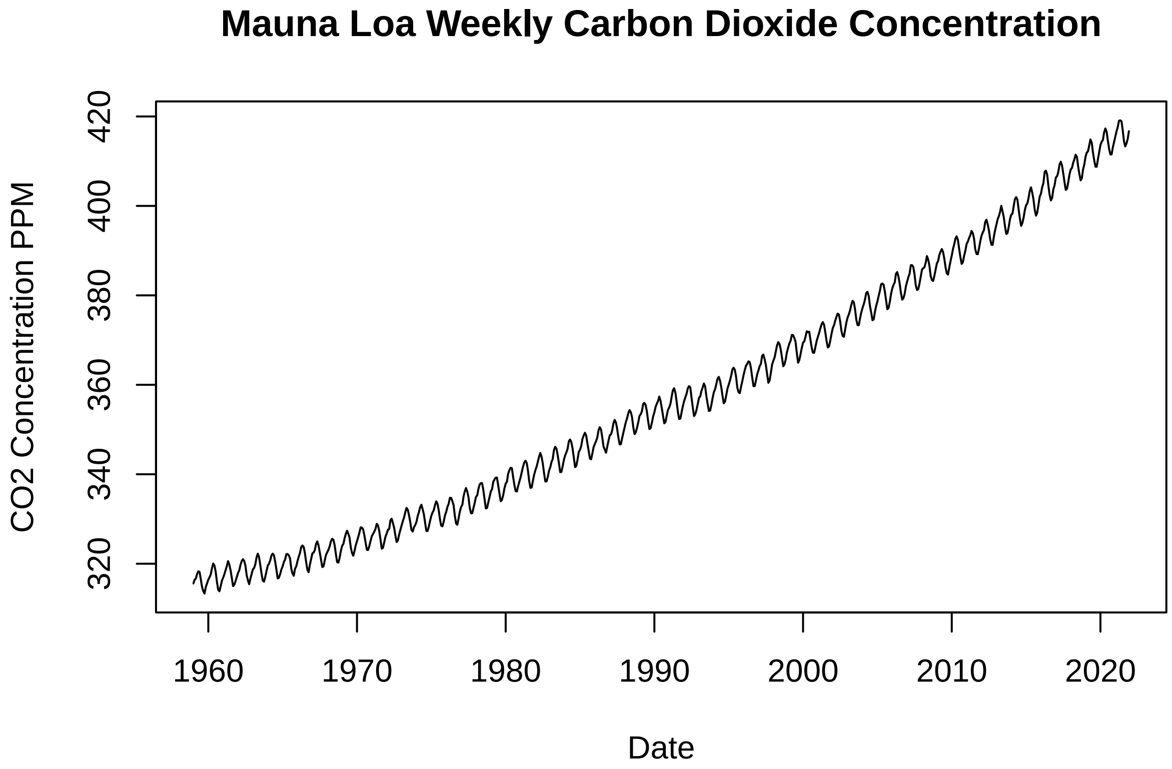
```
co2_ts <- ts(co2_monthly$co2_concentration, start = c(1959, 1), frequency = 12)
co2_ts
```

```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 1959 315.58 316.48 316.65 317.72 318.29 318.15 316.54 314.80 313.84 313.33
## 1960 316.43 316.98 317.58 319.03 320.04 319.59 318.18 315.90 314.17 313.83
## 1961 316.89 317.70 318.54 319.48 320.58 319.77 318.57 316.79 314.99 315.31
## 1962 317.94 318.55 319.68 320.57 321.02 320.62 319.61 317.40 316.25 315.42
## 1963 318.74 319.07 319.86 321.38 322.25 321.48 319.74 317.77 316.21 315.99
## 1964 319.57 320.01 320.74 321.84 322.26 321.89 320.44 318.69 316.70 316.87
## 1965 319.44 320.44 320.89 322.14 322.17 321.87 321.21 318.87 317.81 317.30
## 1966 320.62 321.60 322.39 323.70 324.08 323.75 322.38 320.36 318.64 318.10
## 1967 322.33 322.50 323.04 324.42 325.00 324.09 322.54 320.92 319.25 319.39
## 1968 322.57 323.15 323.89 325.02 325.57 325.36 324.14 322.11 320.33 320.25
## 1969 324.00 324.42 325.63 326.66 327.38 326.71 325.88 323.66 322.38 321.78
## 1970 325.06 325.98 326.93 328.13 328.08 327.67 326.34 324.69 323.10 323.06
## 1971 326.17 326.68 327.17 327.79 328.93 328.57 327.36 325.43 323.36 323.56
## 1972 326.77 327.63 327.75 329.72 330.07 329.09 328.04 326.32 324.84 325.20
## 1973 328.55 329.56 330.30 331.50 332.48 332.07 330.87 329.31 327.51 327.18
## 1974 329.35 330.71 331.48 332.65 333.19 332.20 331.07 329.15 327.33 327.28
## 1975 330.73 331.46 331.94 333.11 333.95 333.42 331.97 329.95 328.50 328.36
## 1976 331.56 332.74 333.36 334.74 334.72 333.98 333.08 330.68 328.96 328.72
## 1977 332.68 333.17 334.96 336.14 336.93 336.17 334.89 332.56 331.29 331.28
## 1978 334.94 335.26 336.66 337.69 338.02 338.01 336.50 334.42 332.36 332.45
## 1979 336.14 336.69 338.27 338.82 339.24 339.26 337.54 335.72 333.97 334.24
## 1980 337.90 338.34 340.07 340.93 341.45 341.36 339.45 337.67 336.25 336.14
## 1981 339.29 340.55 341.63 342.60 343.04 342.54 340.82 338.48 336.95 337.05
## 1982 340.93 341.76 342.77 343.96 344.77 343.88 342.42 340.24 338.38 338.41
## 1983 341.57 342.79 343.37 345.40 346.14 345.76 344.32 342.51 340.46 340.53
## 1984 344.21 344.92 345.68 347.37 347.78 347.16 345.79 343.74 341.59 341.86
## 1985 345.48 346.41 347.91 348.66 349.28 348.65 346.90 345.26 343.47 343.35
## 1986 346.78 347.48 348.25 349.86 350.52 349.98 348.25 346.17 345.48 344.82
## 1987 348.73 348.92 349.81 351.40 352.15 351.58 350.21 348.20 346.66 346.72
## 1988 350.51 351.70 352.50 353.67 354.35 353.88 352.80 350.49 348.97 349.37
## 1989 353.07 353.43 354.08 355.72 355.95 355.44 354.05 351.84 350.09 350.33
## 1990 353.86 355.10 355.75 356.38 357.38 356.39 354.89 353.06 351.38 351.69
## 1991 354.93 355.82 357.33 358.77 359.23 358.23 356.30 353.97 352.34 352.43
## 1992 356.34 357.21 357.97 359.22 359.71 359.44 357.15 354.99 353.01 353.41
## 1993 357.10 357.42 358.59 359.39 360.30 359.64 357.45 355.76 354.14 354.23
## 1994 358.36 359.04 360.11 361.36 361.78 360.94 359.51 357.59 355.86 356.21
## 1995 360.04 361.00 361.98 363.44 363.83 363.33 361.78 359.33 358.32 358.14
## 1996 362.20 363.36 364.28 364.69 365.25 365.06 363.69 361.55 359.69 359.72
## 1997 363.24 364.21 364.65 366.49 366.77 365.73 364.46 362.40 360.44 360.98
## 1998 365.39 366.10 367.36 368.79 369.56 369.13 367.98 366.10 364.16 364.54
## 1999 368.35 369.28 369.84 371.15 371.12 370.46 369.61 367.06 364.95 365.52
## 2000 369.45 369.71 370.75 371.98 371.75 371.87 370.02 368.27 367.15 367.18
## 2001 370.76 371.69 372.63 373.55 374.03 373.40 371.68 369.78 368.34 368.61
## 2002 372.70 373.37 374.30 375.19 375.93 375.69 374.16 372.03 370.92 370.73
## 2003 375.07 375.82 376.64 377.92 378.78 378.46 376.88 374.57 373.34 373.31
## 2004 377.17 378.05 379.06 380.54 380.80 379.87 377.65 376.17 374.43 374.63
## 2005 378.63 379.91 380.95 382.48 382.64 382.40 380.93 378.93 376.89 377.18
## 2006 381.58 382.40 382.86 384.80 385.22 384.24 382.65 380.60 379.04 379.33
## 2007 383.10 384.12 384.81 386.73 386.78 386.33 384.73 382.24 381.20 381.37
## 2008 385.78 386.06 386.28 387.33 388.78 387.99 386.61 384.32 383.41 383.21
## 2009 387.17 387.70 389.04 389.76 390.36 389.70 388.24 386.29 384.95 384.64
## 2010 388.91 390.41 391.37 392.67 393.21 392.38 390.41 388.54 387.03 387.43
## 2011 391.50 392.05 392.80 393.44 394.41 393.95 392.72 390.33 389.28 389.19
## 2012 393.31 394.04 394.59 396.38 396.93 395.91 394.56 392.59 391.32 391.27
## 2013 395.78 397.03 397.66 398.64 400.02 398.81 397.51 395.39 393.72 393.90
## 2014 398.04 398.27 399.91 401.51 401.96 401.43 399.27 397.18 395.54 396.16
## 2015 400.18 400.55 401.74 403.35 404.15 402.97 401.46 399.11 397.82 398.49
## 2016 402.73 404.25 405.06 407.60 407.90 406.99 404.59 402.45 401.23 401.79
## 2017 406.36 406.66 407.54 409.22 409.89 409.08 407.33 405.32 403.57 403.82
## 2018 408.15 408.52 409.59 410.45 411.44 410.99 408.90 407.16 405.71 406.19
## 2019 411.03 411.96 412.18 413.54 414.86 414.16 411.97 410.18 408.76 408.75
## 2020 413.61 414.34 414.74 416.45 417.31 416.60 414.62 412.78 411.52 411.51
## 2021 415.52 416.75 417.64 419.05 419.13 418.94 416.96 414.47 413.30 413.93
##         Nov    Dec
## 1959 314.81 315.58
## 1960 315.00 316.19
## 1961 316.10 317.01
## 1962 316.69 317.70
## 1963 317.07 318.35
## 1964 317.68 318.71
## 1965 318.87 319.42
## 1966 319.78 321.03
## 1967 320.73 321.96
## 1968 321.32 322.89
## 1969 322.86 324.12
## 1970 324.01 325.13
## 1971 324.80 326.01
## 1972 326.50 327.55
## 1973 328.16 328.64
## 1974 328.31 329.58
## 1975 329.38 330.78
## 1976 330.16 331.62
## 1977 332.46 333.60
## 1978 333.76 334.91
## 1979 335.32 336.82
## 1980 337.30 338.29
## 1981 338.57 339.91
## 1982 339.44 340.78
## 1983 341.79 343.20
## 1984 343.31 345.00
## 1985 344.73 346.12
## 1986 346.22 347.49
## 1987 348.08 349.28
## 1988 350.42 351.62
## 1989 351.55 352.91
## 1990 353.14 354.41
## 1991 353.89 355.21
## 1992 354.42 355.68
## 1993 355.53 357.03
## 1994 357.65 359.10
## 1995 359.61 360.82
## 1996 361.04 362.39
## 1997 362.65 364.51
## 1998 365.67 367.30
## 1999 366.88 368.26
## 2000 368.53 369.83
## 2001 369.94 371.42
## 2002 372.43 373.98
## 2003 374.84 376.17
## 2004 376.33 377.68
## 2005 378.54 380.31
## 2006 380.35 382.02
## 2007 382.70 384.19
## 2008 384.41 385.79
## 2009 386.23 387.63
## 2010 388.87 389.99
## 2011 390.48 392.06
## 2012 393.20 394.57
## 2013 395.36 397.03
## 2014 397.40 399.08
## 2015 400.27 402.06
## 2016 403.72 404.64
## 2017 405.31 407.00
## 2018 408.21 409.27
## 2019 410.48 411.98
## 2020 413.12 414.26
## 2021 415.01 416.71
```

# 1. Exploratory analysis

Let’s plot the data for the first time:

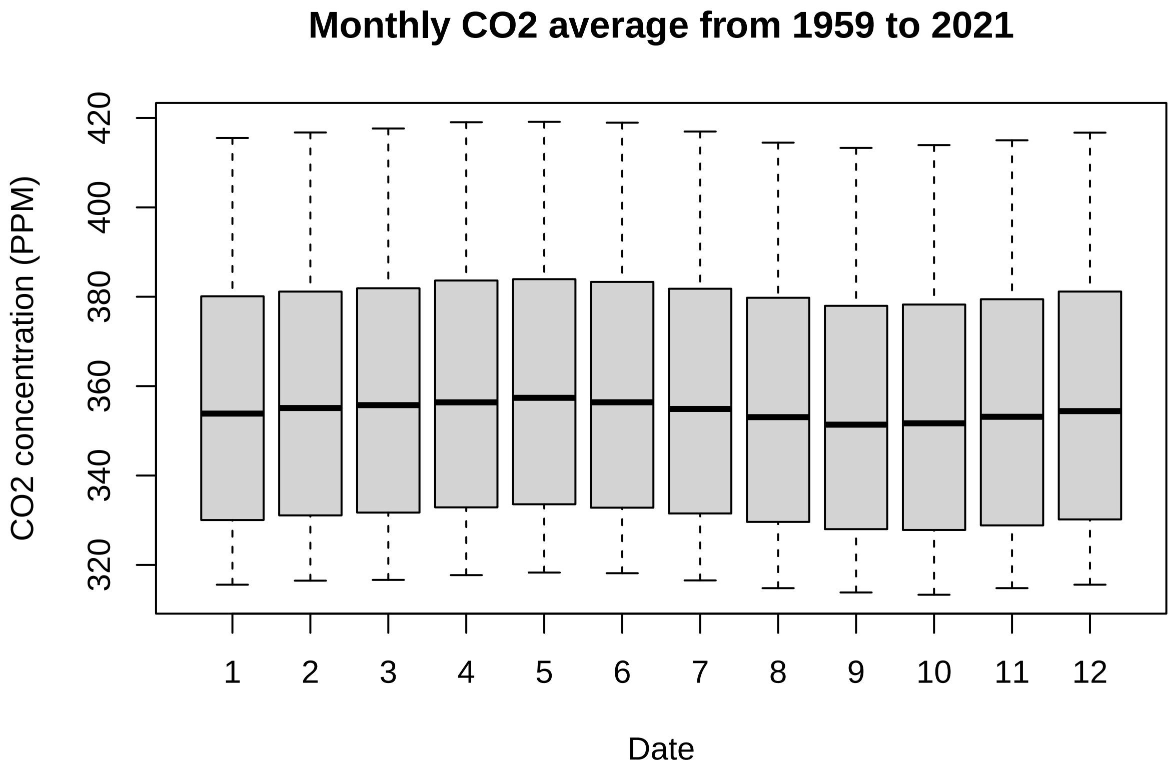
```
plot(
  co2_monthly$date,
  co2_monthly$co2_concentration,
  type = 'l',
  xlab = 'Date',
  ylab = 'CO2 Concentration PPM',
  main = 'Mauna Loa Weekly Carbon Dioxide Concentration'
)
```



From a purely visual analysis, it seems like the data presents both seasonality and an upward trend. Let's first take a look at the seasonality:

### Examining seasonality

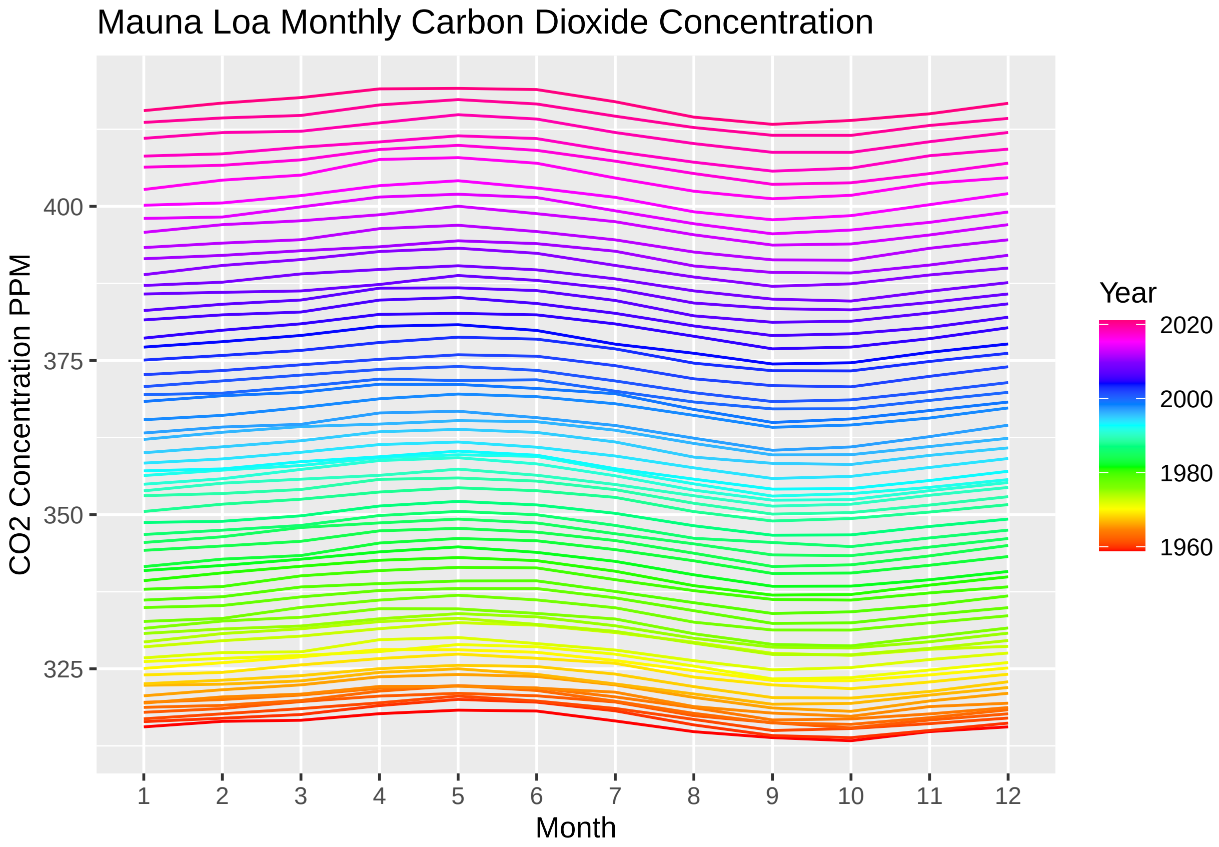
```
boxplot(co2_ts~cycle(co2_ts),xlab="Date", ylab = "CO2 concentration (PPM)",main ="Monthly CO2 average f
```



Here we can see that the monthly concentrations show seasonality, with a high point around the months 4-5 and a low point around the months 9-10.

Let's take a look at the seasonality over the years:

```
ggplot(data = co2_monthly_full, aes(factor(month), co2_concentration, colour = year, group = year)) +
  geom_line() +
  xlab('Month') +
  ylab('CO2 Concentration PPM') +
  ggtitle('Mauna Loa Monthly Carbon Dioxide Concentration') +
  scale_color_gradientn('Year', colors = rainbow(length(unique(co2_monthly_full$month))))
```



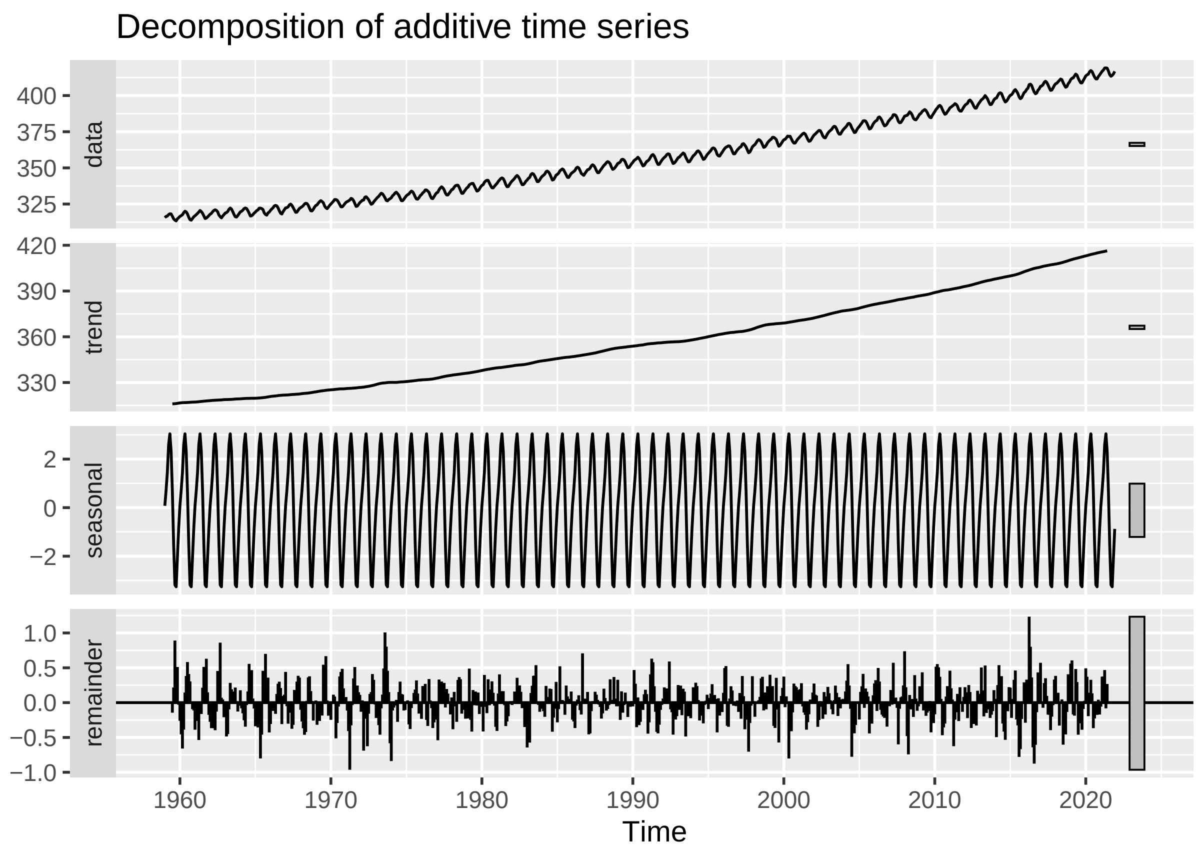
The results seem to be consistent over the years, with the seasonality being maintained through the upward trend in CO2 concentrations.

From these plots we could gather that: - The CO2 concentration shows a clear upward linear trend over the years - The monthly concentrations show seasonality, with a peak around the months 4-5 and a valley around the months 9-10.

## Time Series decomposition

Let's take a better look at the time series decomposition:

```
decomposeCO2 <- decompose(co2_ts,"additive")
autoplot(decomposeCO2)
```



Removing the trend seen above can be useful for further analysis. Let's check if differencing would be enough to make the time series stationary with adf.test():

```
adf.test(diff(co2_ts), alternative="stationary", k=0)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(co2_ts)
## Dickey-Fuller = -11.442, Lag order = 0, p-value = 0.01
## alternative hypothesis: stationary
```

This low P value confirms the alternative hypothesis (that is, the time series has become stationary after one stage of differencing).

**When choosing an approach to deal with this data, this has to be taken into account**. I'll first present a naive approach not dealing with the non-stationarity of the data, and then I'll compare its results to an approach that applies differencing.

## Making predictions

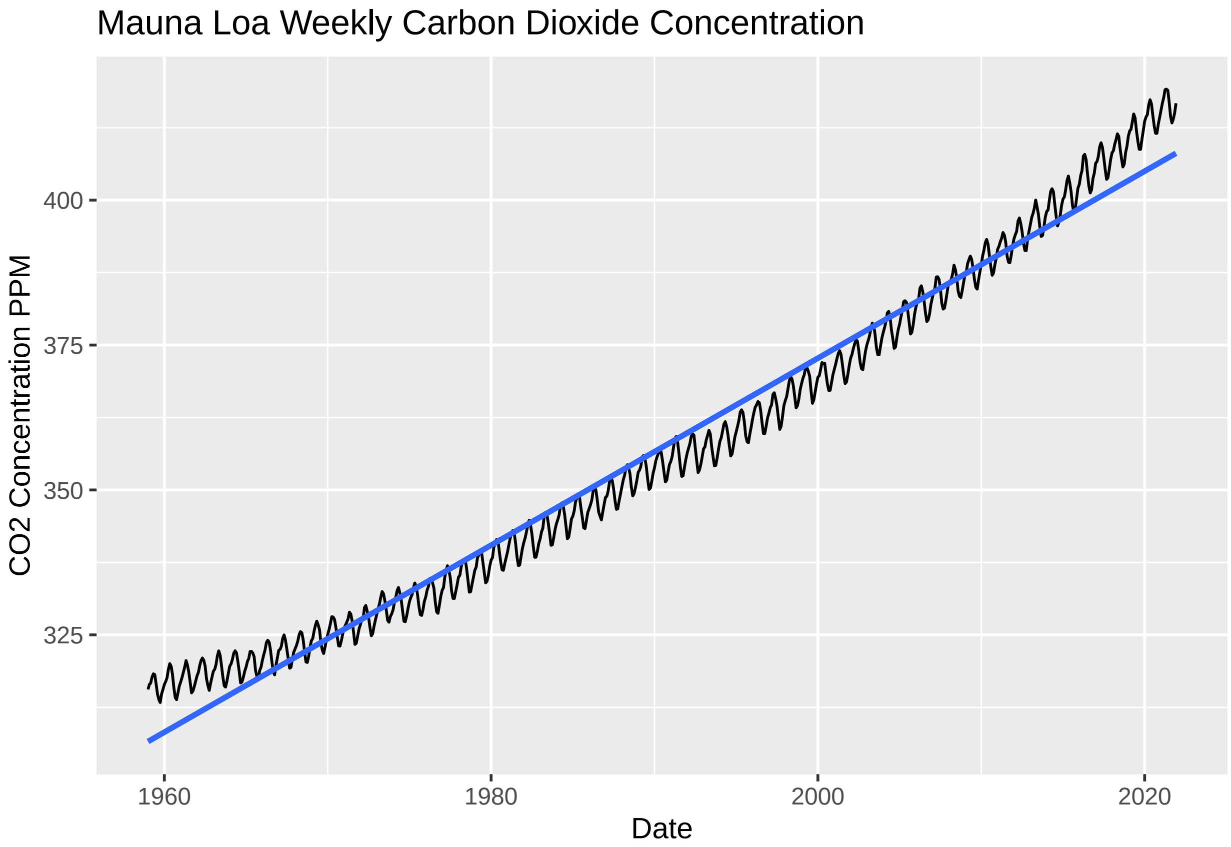
### First attempt:

Let's first take a look at the P value for the trend, using the date as the independent variable.

```
simple_reg_co2 <- lm(co2_concentration ~ date, data = co2_monthly)
```

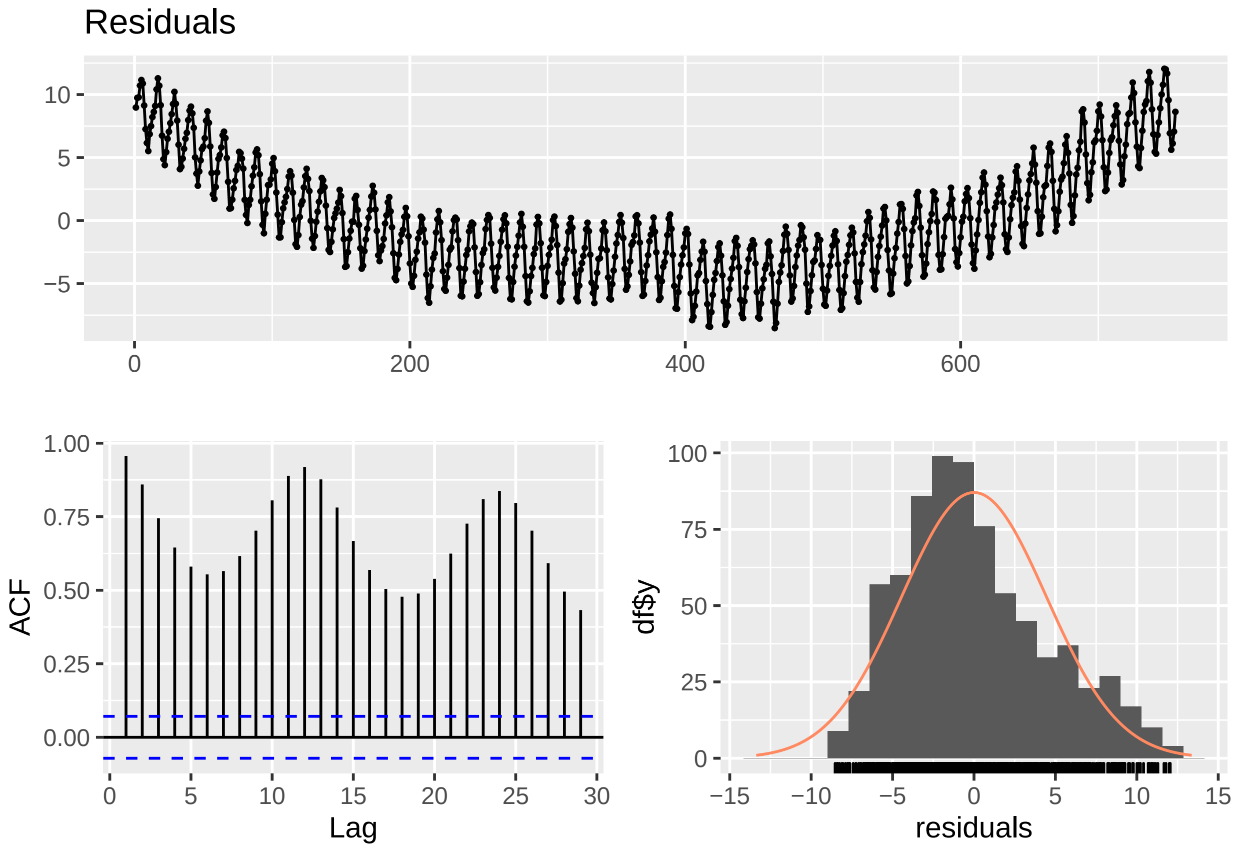
```
ggplot(data = co2_monthly, aes(x = date, y = co2_concentration)) +
  geom_line() +
  xlab('Date') +
  ylab('CO2 Concentration PPM') +
  ggtitle('Mauna Loa Weekly Carbon Dioxide Concentration') +
  stat_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Visually, we can see that this model is far from perfect as it cannot follow the seasonality. I suspect the errors will have a high autocorrelation because of that:

```
checkresiduals(simple_reg_co2)
```



```
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  Residuals
## LM test = 742.63, df = 10, p-value < 2.2e-16
```

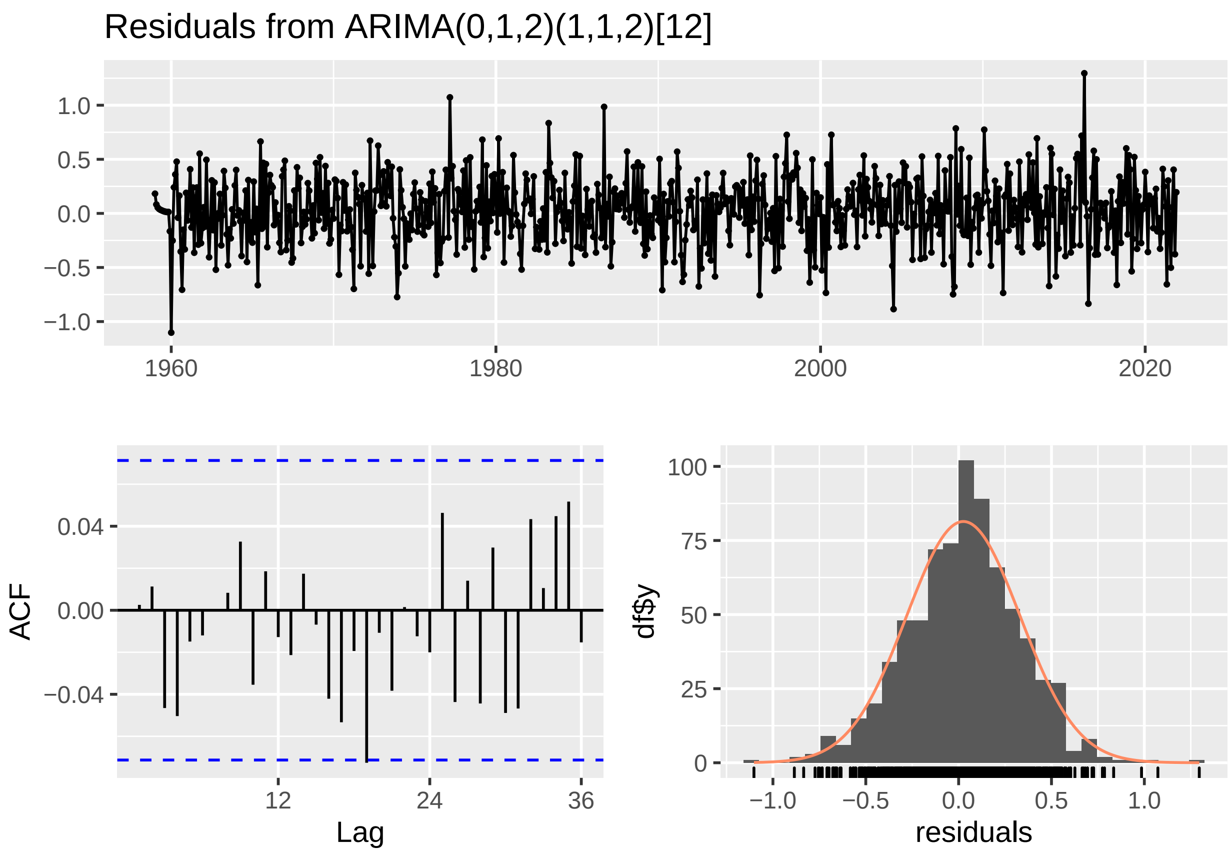
As expected, the autocorrelation of errors is pretty high, which is not what we want from a forecasting model.We can also visually notice that the residuals are skewed by looking at their distribution.

## Second attempt: ARIMA

While ARIMA cannot be used with non-stationary data, auto.arima() will perform the following steps before using ARIMA: > 1. Check for stationarity with ADF > 2. If non-stationary, apply differencing with enough steps to pass ADF > 3. Apply ARIMA

```
arima_co2 <- auto.arima(co2_ts)
```

```
checkresiduals(arima_co2)
```



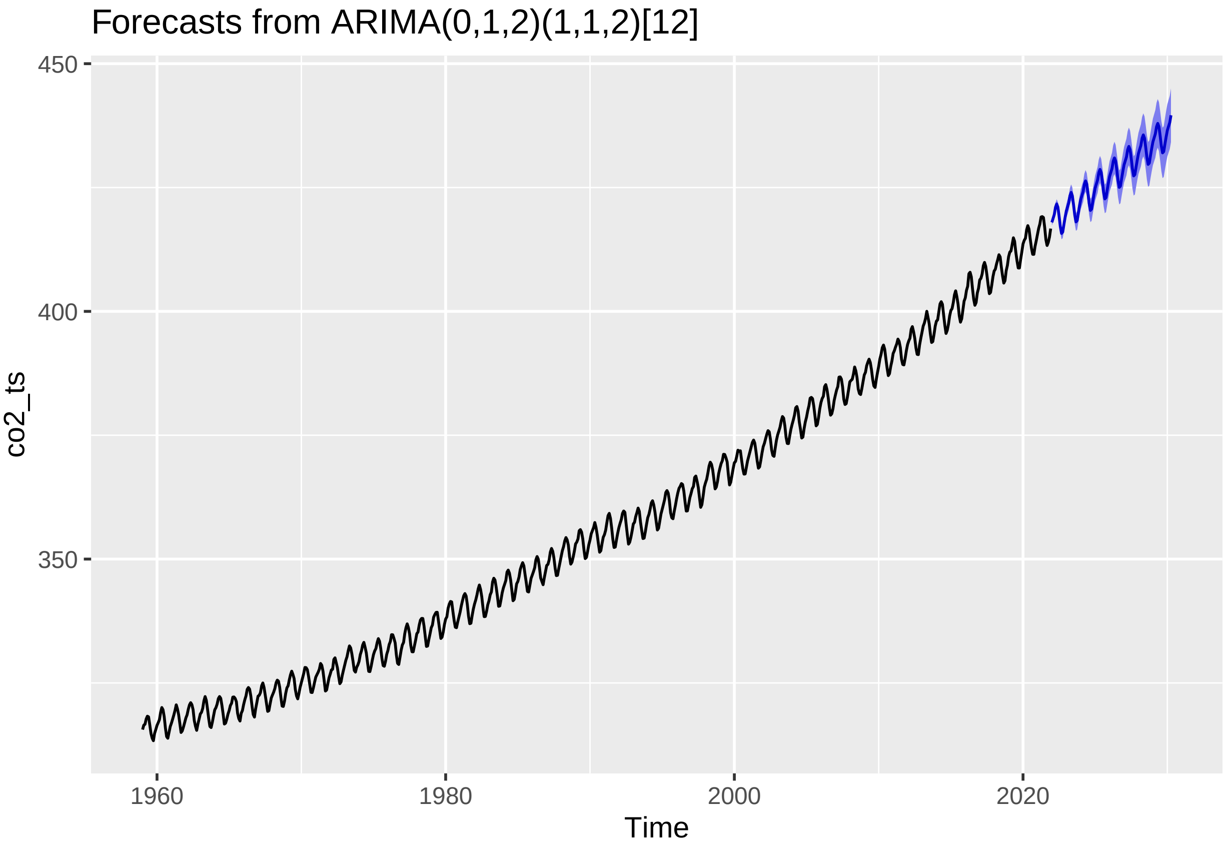
```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,2)(1,1,2)[12]
## Q* = 16.439, df = 19, p-value = 0.6278
##
## Model df: 5.   Total lags used: 24
```

Much better. auto.arima() is able to detect non-stationarity and adapt to it through statistical tests; now we can see that the errors follow a noise (random) pattern, which is preferable for a model.

**Sample forecast:**

Now we can use the forecast function to plot a forecast on the horizon of 100 months, with a confidence interval of 95%.

```
arima_forecast_co2 <- forecast(arima_co2, level = c(95), h = 100)
autoplot(arima_forecast_co2)
```


Forecasts from ARIMA(0,1,2)(1,1,2)[12]
450
400
350
co2_ts
1960
1980
2000
2020
Time