CO2 Keeling Curve and Historical Ice Core Data

David Elser

05-03-2021

Please note that I am not a Computer Scientist and my programming skills are very limited.

Data preprocessing fetch the Data file and check if file already exists

The data is available from the Web site of the Scripps Institute

https://scrippsco2.ucsd.edu/data/atmospheric_co2/primary_mlo_co2_record.html https://scrippsco2.ucsd.edu/data/atmospheric_co2/icecore_merged_products.html

Soure script for the Keeling Curve available at : https://github.com/DemirhanOzelTrojan/Keeling_Predict/blob/master/main.R

`install.packages('forecast', dependencies = TRUE) install.packages("imputeTS") install.packages("tidyve`

Load librarys

```
library(ggplot2)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(imputeTS)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Fetch CO2 measurements from Mauna Loa observatory, Hawaii, USA

```
data_url = "https://scrippsco2.ucsd.edu/assets/data/atmospheric/stations/in_situ_co2/monthly/monthly_in
data_file = "CO2.csv"

if (!file.exists(data_file)) {
    download.file(data_url, data_file, method="auto")
}
```

Fetch Ice Core CO2 Data

```
data_url = "https://scrippsco2.ucsd.edu/assets/data/atmospheric/merged_ice_core_mlo_spo/merged_ice_core
data_file1 = "Ice-CO2.csv"

if (!file.exists(data_file1)) {
    download.file(data_url, data_file1, method="auto")
}
```

Skip the frist 36 lines

```
data_ice = read.csv(data_file1, header=FALSE, skip=36)
```

Skip the frist 5 lines of the file

```
data = read.csv(data_file, header=TRUE, skip=5)
```

Let's have a look at what we got and check NA

```
head(data_ice)
```

```
##        V1     V2
## 1   13.30 276.75
## 2   29.52 277.88
## 3   55.96 277.38
## 4  104.48 277.51
## 5  135.98 278.13
## 6  168.21 280.05
```

```
tail(data_ice)
```

```
##          V1     V2
## 179 2015.5 399.26
## 180 2016.5 402.43
## 181 2017.5 404.54
## 182 2018.5 406.89
## 183 2019.5 409.53
## 184 2020.5 411.99
```

```r
na_records = apply(data_ice, 1, function (x) any(is.na(x)))
data[na_records,]
```

```
##  [1] X        X.1      Excel    X.2      X.ppm.   X.ppm..1 X.ppm..2 X.ppm..3
##  [9] X.ppm..4 X.ppm..5
## <0 rows> (or 0-length row.names)
```

Conversion of decimal-date into Years, rename V2 into ppm

```r
data_ice$date<- format(date_decimal(data_ice$V1), "%Y")
head(data_ice)
```

```
##       V1     V2 date
## 1  13.30 276.75 0013
## 2  29.52 277.88 0029
## 3  55.96 277.38 0055
## 4 104.48 277.51 0104
## 5 135.98 278.13 0135
## 6 168.21 280.05 0168
```

```r
tail(data_ice)
```

```
##         V1     V2 date
## 179 2015.5 399.26 2015
## 180 2016.5 402.43 2016
## 181 2017.5 404.54 2017
## 182 2018.5 406.89 2018
## 183 2019.5 409.53 2019
## 184 2020.5 411.99 2020
```

```r
names(data_ice)[names(data_ice) == "V2"] <- "ppm"
data_ice$Year=as.numeric(data_ice$date)
head(data_ice)
```

```
##       V1    ppm date Year
## 1  13.30 276.75 0013   13
## 2  29.52 277.88 0029   29
## 3  55.96 277.38 0055   55
## 4 104.48 277.51 0104  104
## 5 135.98 278.13 0135  135
## 6 168.21 280.05 0168  168
```

```r
tail(data_ice)
```

```
##         V1    ppm date Year
## 179 2015.5 399.26 2015 2015
## 180 2016.5 402.43 2016 2016
## 181 2017.5 404.54 2017 2017
## 182 2018.5 406.89 2018 2018
## 183 2019.5 409.53 2019 2019
## 184 2020.5 411.99 2020 2020
```

Let's have a look at what we got from Mauna Loa observatory and check NA

```
head(data)
```

```
##      X X.1 Excel      X.2 X.ppm. X.ppm..1 X.ppm..2 X.ppm..3 X.ppm..4 X.ppm..5
## 1 1958   1 21200 1958.041 -99.99   -99.99   -99.99   -99.99   -99.99   -99.99
## 2 1958   2 21231 1958.126 -99.99   -99.99   -99.99   -99.99   -99.99   -99.99
## 3 1958   3 21259 1958.203 315.70   314.43   316.19   314.90   315.70   314.43
## 4 1958   4 21290 1958.288 317.45   315.16   317.30   314.98   317.45   315.16
## 5 1958   5 21320 1958.370 317.51   314.71   317.86   315.06   317.51   314.71
## 6 1958   6 21351 1958.455 -99.99   -99.99   317.24   315.14   317.24   315.14
```

```
tail(data)
```

```
##        X X.1 Excel      X.2 X.ppm. X.ppm..1 X.ppm..2 X.ppm..3 X.ppm..4 X.ppm..5
## 763 2021   7 44392 2021.537 -99.99   -99.99   -99.99   -99.99   -99.99   -99.99
## 764 2021   8 44423 2021.622 -99.99   -99.99   -99.99   -99.99   -99.99   -99.99
## 765 2021   9 44454 2021.707 -99.99   -99.99   -99.99   -99.99   -99.99   -99.99
## 766 2021  10 44484 2021.789 -99.99   -99.99   -99.99   -99.99   -99.99   -99.99
## 767 2021  11 44515 2021.874 -99.99   -99.99   -99.99   -99.99   -99.99   -99.99
## 768 2021  12 44545 2021.956 -99.99   -99.99   -99.99   -99.99   -99.99   -99.99
```

```
na_records = apply(data, 1, function (x) any(is.na(x)))
data[na_records,]
```

```
##  [1] X        X.1      Excel    X.2      X.ppm.   X.ppm..1 X.ppm..2 X.ppm..3
##  [9] X.ppm..4 X.ppm..5
## <0 rows> (or 0-length row.names)
```

impute data of column X.ppm..4(CO2, seasonally filled,adjusted filled)

```
data$X.ppm..4[data$X.ppm..4 == -99.99] <- NA
data$X.ppm..4 <- na.interpolation(data$X.ppm..4)
```

```
## Warning: na.interpolation will be replaced by na_interpolation.
##            Functionality stays the same.
##            The new function name better fits modern R code style guidelines.
##            Please adjust your code accordingly.
```

```
head(data)
```

```
##      X X.1 Excel      X.2 X.ppm. X.ppm..1 X.ppm..2 X.ppm..3 X.ppm..4 X.ppm..5
## 1 1958   1 21200 1958.041 -99.99   -99.99   -99.99   -99.99   315.70   -99.99
## 2 1958   2 21231 1958.126 -99.99   -99.99   -99.99   -99.99   315.70   -99.99
## 3 1958   3 21259 1958.203 315.70   314.43   316.19   314.90   315.70   314.43
## 4 1958   4 21290 1958.288 317.45   315.16   317.30   314.98   317.45   315.16
## 5 1958   5 21320 1958.370 317.51   314.71   317.86   315.06   317.51   314.71
## 6 1958   6 21351 1958.455 -99.99   -99.99   317.24   315.14   317.24   315.14
```

```
tail(data)
```

```
##        X X.1 Excel      X.2 X.ppm. X.ppm..1 X.ppm..2 X.ppm..3 X.ppm..4 X.ppm..5
## 763 2021   7 44392 2021.537 -99.99   -99.99   -99.99   -99.99   415.16   -99.99
## 764 2021   8 44423 2021.622 -99.99   -99.99   -99.99   -99.99   415.16   -99.99
## 765 2021   9 44454 2021.707 -99.99   -99.99   -99.99   -99.99   415.16   -99.99
## 766 2021  10 44484 2021.789 -99.99   -99.99   -99.99   -99.99   415.16   -99.99
## 767 2021  11 44515 2021.874 -99.99   -99.99   -99.99   -99.99   415.16   -99.99
## 768 2021  12 44545 2021.956 -99.99   -99.99   -99.99   -99.99   415.16   -99.99
```

data preparation

```
data.ts <- ts(data$X.ppm..4, start = c(1958, 1), end = c(2021, 1), freq = 12)
```

split training and testing data, use 90% of available data in the co2.csv file

```
train.test.split <- 0.9
split.index <- round(length(data.ts) * train.test.split)
```

train time series

```
data.train.ts <- window(data.ts, start = c(1958, 1), end = c(1958, split.index))
```

test time series

```
data.test.ts <- window(data.ts, start = c(1958, split.index + 1), end = c(1958, length(data.ts)))
```

create a fit using the time series

```
fit <- ets(data.ts, model = 'AAA')
```

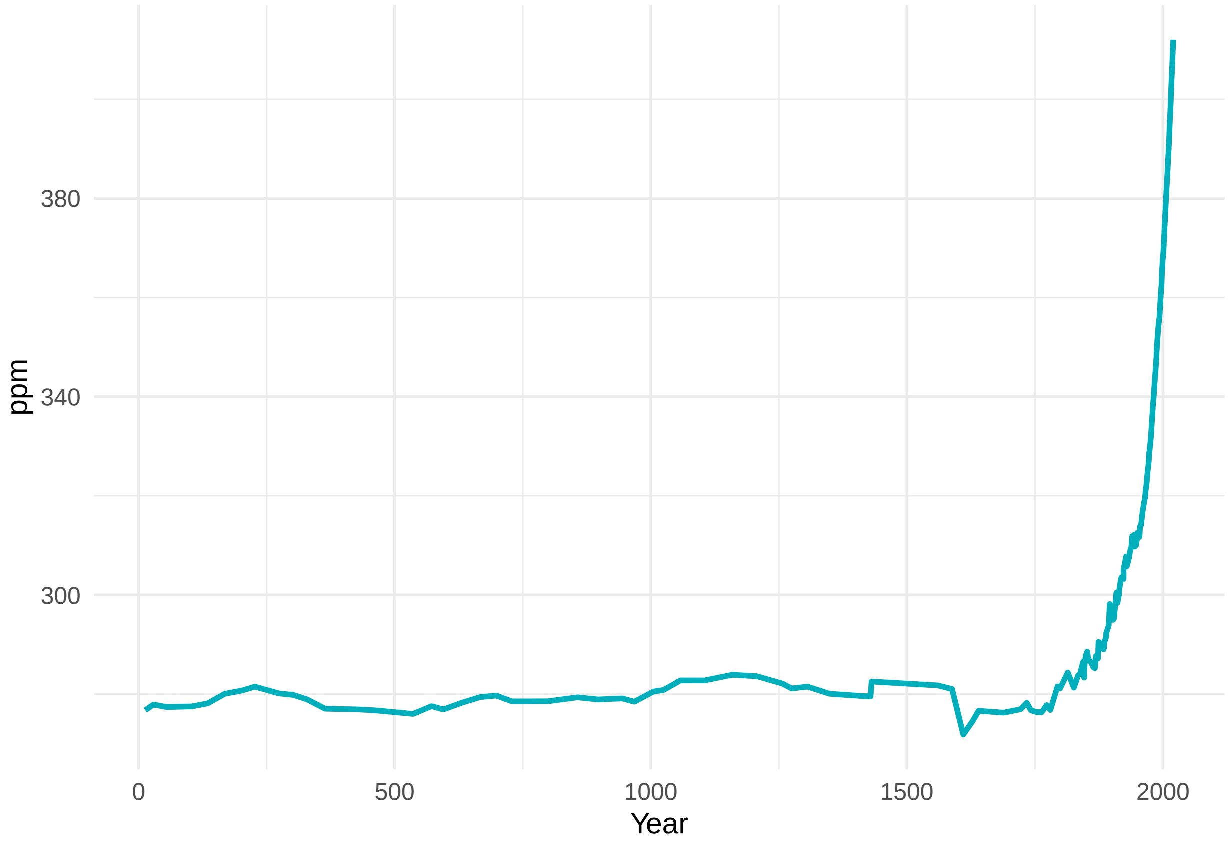
forecast using the fit

```
predictions <- forecast(fit, 12 * 20, level = 0.95)
```

Plot Ice Core Data

```
theme_set(theme_minimal())

ggplot(data = data_ice, aes(x = Year, y = ppm))+
  geom_line(color = "#00AFBB", size = 1)
```



Plot predictions till 2025

```
plot(predictions,
     main = '',
     xlab = 'Time',
     ylab = 'CO2 (ppm)',
     xlim = c(1958, 2025),
     shaded = TRUE,
     shadecols=c('light blue'))
legend('topleft',
       legend=c('Historical Data',
                'Predicted Data',
                '95% CI'),
       fill=c(NA, NA, 'light blue'),
       col=c('black', 'blue', NA),
```

```
       lty=c(1, 1, NA),
       lwd=c(1, 2, NA),
       border=c(NA, NA, 'black'),
       inset=0.01)
```