MATH 4070 R Session 4: Time Series Regression

Whitney

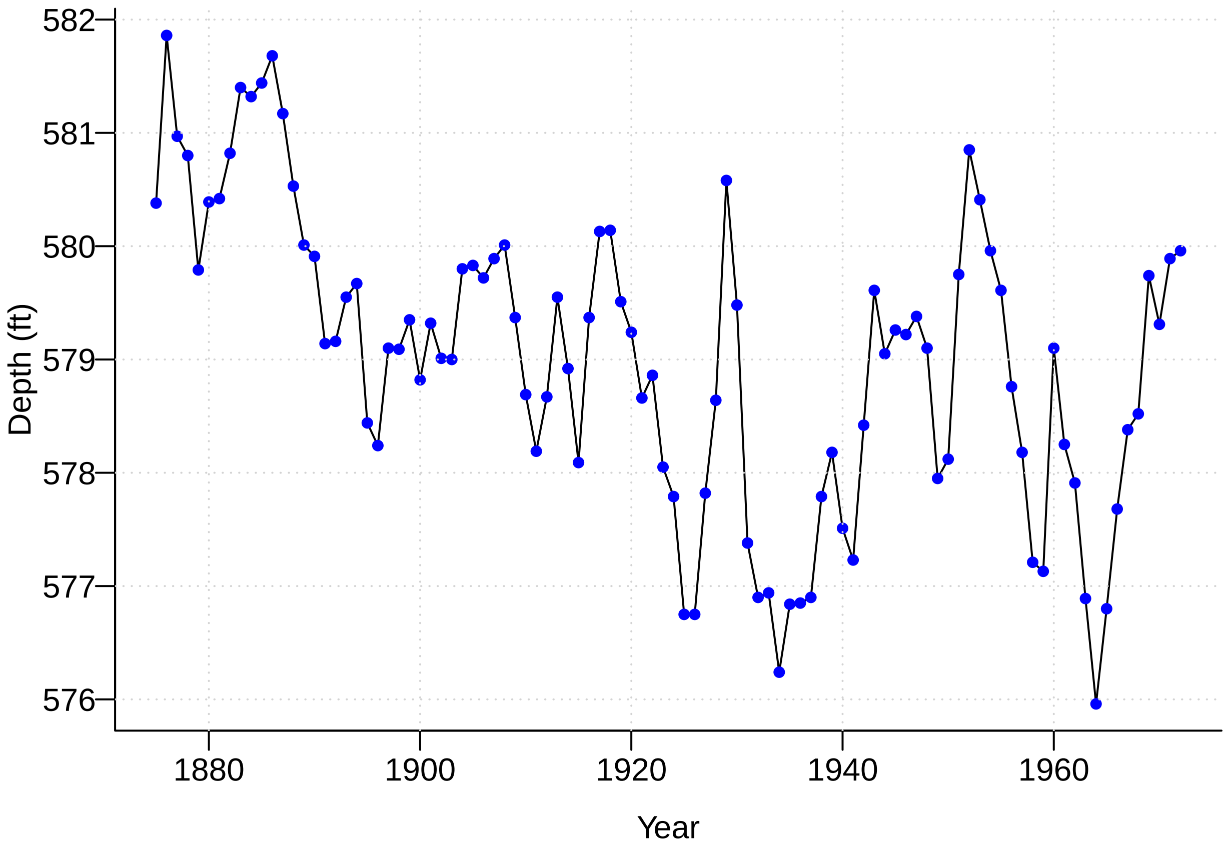
Contents

Time Series Data

Lake Huron Time Series

Annual measurements of the level of Lake Huron in feet

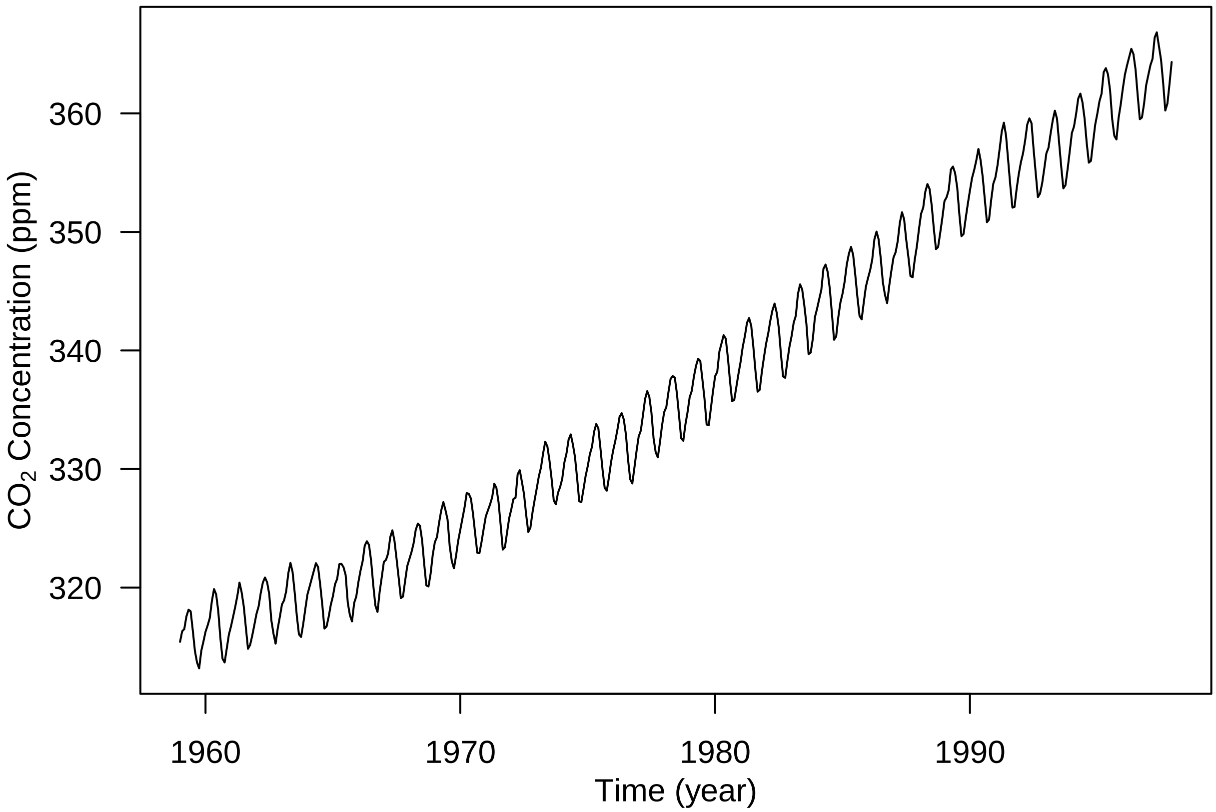
```
par(mar = c(3.2, 3.2, 0.5, 0.5), mgp = c(2, 0.5, 0), bty = "L")
data(LakeHuron)
plot(LakeHuron, ylab = "Depth (ft)", xlab = "Year", las = 1)
points(LakeHuron, cex = 0.8, col = "blue", pch = 16)
grid()
```



CO_2 **Concentration**

The Mauna Loa monthly atmospheric concentrations of CO_2 are expressed in parts per million (ppm) and reported on the preliminary 1997 SIO manometric mole fraction scale.

```
data(co2)
par(mar = c(3.8, 4, 0.8, 0.6))
plot(co2, las = 1, xlab = "", ylab = "")
mtext("Time (year)", side = 1, line = 2)
mtext(expression(paste("CO"[2], " Concentration (ppm)")), side = 2, line = 2.5)
```



Trend, Seasonality, and Noise

A time series can often be decomposed into three components: *trend*, *seasonal effect*, and *noise* (i.e., the remaining variation once trend and seasonal effects have been removed).

The *additive model*:

$Y_t = \mu_t + s_t + \eta_t$

The Mauna Loa monthly atmospheric CO_2 concentration time series is an example where the additive model may be appropriate for describing the time series.

The most commonly used modeling approach is to first focus on the trend and seasonal variation, and then model the remaining 'noise' term, which may exhibit temporal correlation, as a stationary time series. This approach suggests a two-stage procedure:

1. To estimate the tend μ_t and seasonal variation s_t.
2. To calculate the residual time series $\hat{\eta}_t$

$$\hat{\eta}_t = y_t - \hat{\mu}_t - \hat{s}_t, \quad t = 1, \cdots, T$$

   Methods for accomplishing the first stage will be described in this week's materials.

Let's start with the trend estimation. First, for ease of presentation, we assume there is no seasonal variation, that is, $Y_t = \mu_t + \eta_t$ with $\mathbb{E}[\eta_t] = 0, \forall t$.

Trend Estimation

Usually, the form of the trend is unknown and needs to specified and estimated. The first method is Linear Regression.

Method I: Regression

To represent the trend component as

$\mu_t = \beta_0 + \beta_1 x_{1t} + \cdots + \beta_p x_{pt}$

Here, a couple of assumptions have been made: first, there is no seasonal component, and therefore, the model for describing what the observed time series might have been generated can be written as

$$Y_t = \mu_t + \eta_t$$

Here we assume that η_t is a zero-mean process (i.e., $\mathbb{E}[\eta_t] = 0, \forall t$), we have $\mu_t = \mathbb{E}[Y_t], \forall t$. Without additional time series $\{x_t\}$ to be served as a covariate, we can use $\{t\} = \{1875, 1876, \cdots, 1972\}$ as the covariate to perform a linear regression by assuming *there is a linear trend in time*, that is

$$Y_t = \beta_0 + \beta_1 t + \eta_t$$

Next, we need to *estimate* the parameters β_0 and β_1. Like in regression analysis, we can use the method of least squares to obtain the estimated parameters $\hat{\beta}$ and $\hat{\beta}_1$. Specifically, we estimate these parameters by *ordinary least squares*, which finds the minimizer of the following objective function

$$\ell_{ols} = \sum_{t=1}^{T} (y_t - \beta_0 - \beta_1 t)^2,$$

the estimated parameters $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)^T = (X^T X)^{-1} X^T \boldsymbol{y}$, where $\boldsymbol{X} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & \vdots \\ 1 & T \end{bmatrix}$ and $\boldsymbol{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_T \end{bmatrix}$.

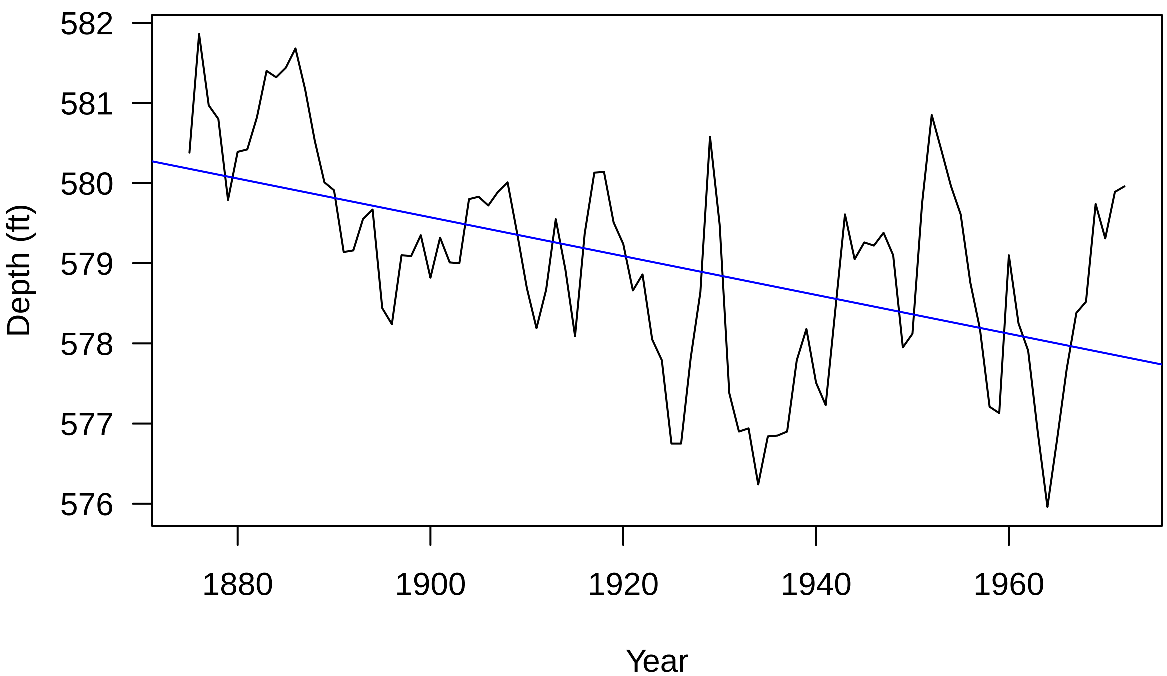
Below we plot the estimated trend $\hat{\mu}_t = \hat{\beta}_0 + \hat{\beta}_1 t$

```r
library(astsa)
yr <- 1875:1972
lm <- lm(LakeHuron ~ yr)
summary(lm)
```

```
##
## Call:
## lm(formula = LakeHuron ~ yr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.50997 -0.72726  0.00083  0.74402  2.53565
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 625.554918   7.764293  80.568  < 2e-16 ***
## yr           -0.024201   0.004036  -5.996 3.55e-08 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.13 on 96 degrees of freedom
## Multiple R-squared:  0.2725, Adjusted R-squared:  0.2649
## F-statistic: 35.95 on 1 and 96 DF,  p-value: 3.545e-08
```

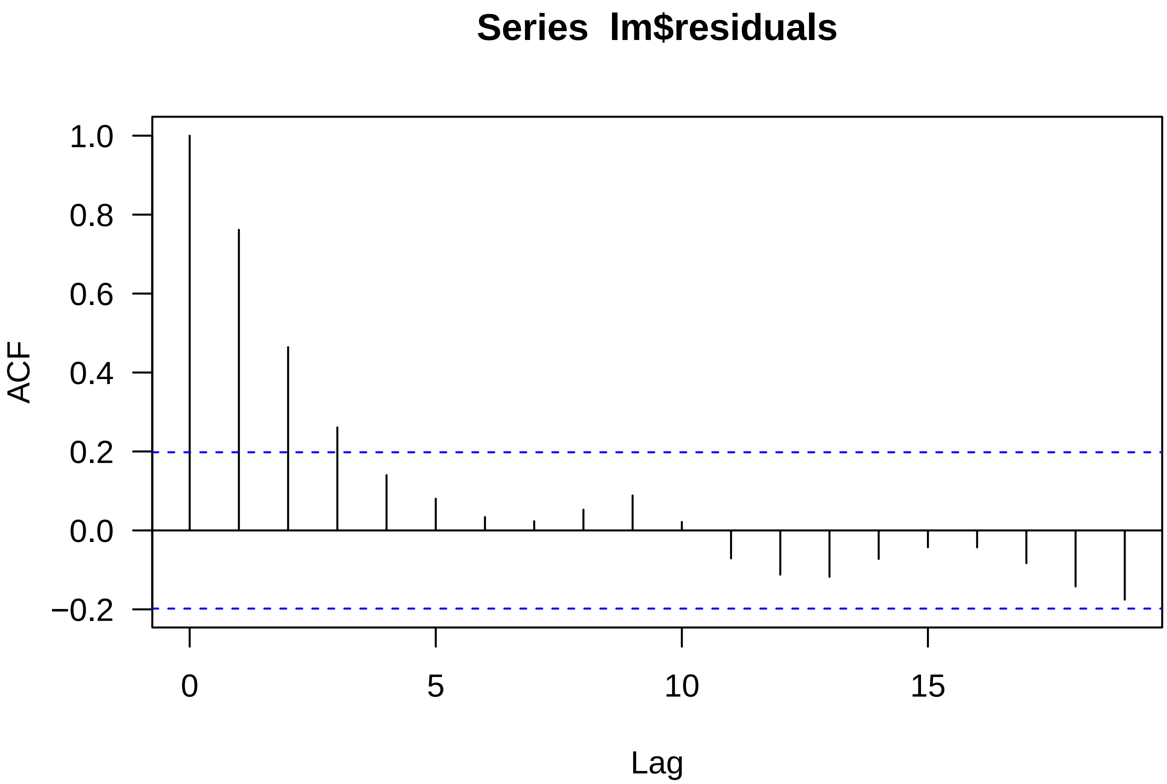
```r
plot(LakeHuron, ylab = "Depth (ft)", xlab = "Year", las = 1)
abline(lm, col = "blue")
```



Note

1. Ordinary least squares (OLS) estimation assumes the observations are independent, which may not be appropriate in the time series context. However, this assumption is typically made to remove the trend, before the correlation in $\{\eta_t\}$ is explicitly modeled by a stationary time series process.
2. Since $\{\eta_t\}$ is typically not an i.i.d. (independent and identically distributed) process (see the ACF plot below), statistical inferences regarding parameters will be invalid (because OLS estimates were calculated assuming i.i.d. errors)

```r
acf(lm$residuals, las = 1)
```



Method II: Smoothing or Local Averaging

The second approach to modeling the trend is using some *smoothing* techniques, which can be thought as performing non-parametric regression.

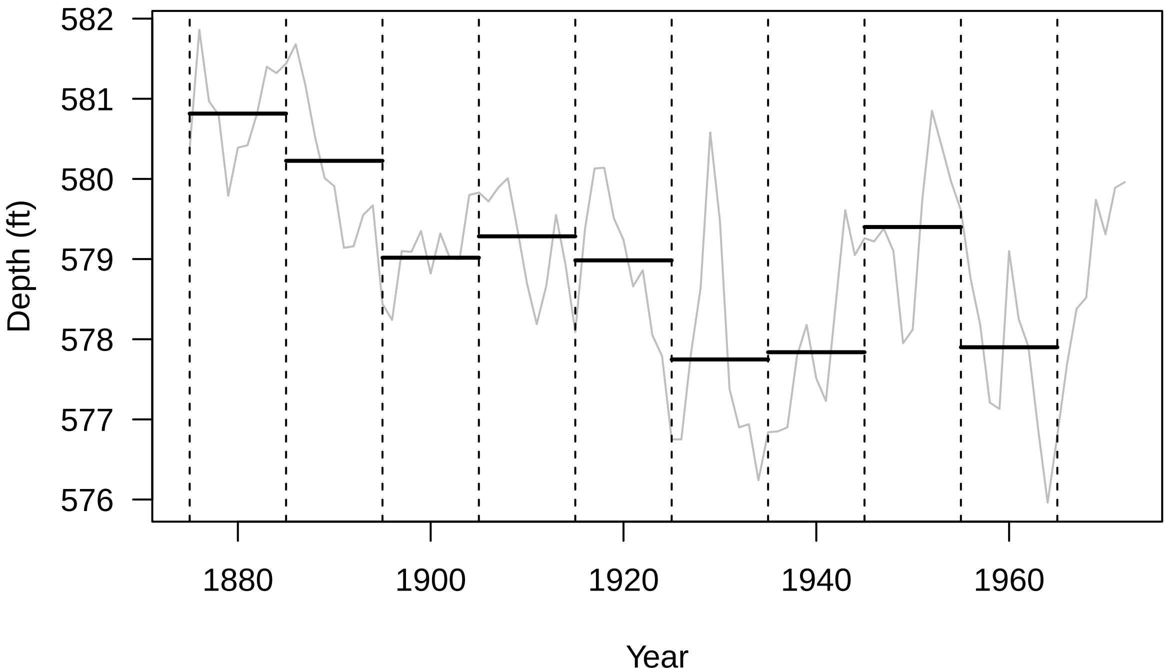
First, we break the time series up into 'small' blocks (each with 10 years of data) and average each block.

```
yr_group <- c(rep(1:9, each = 10), rep(10, 8))

lakeHuron_10yr <- data.frame(cbind(depth = LakeHuron, yrGroup = as.factor(yr_group)))

mean_10yr <- tapply(lakeHuron_10yr$depth, lakeHuron_10yr$yrGroup, mean)

plot(LakeHuron, las = 1, col = "gray", xlab = "Year", ylab = "Depth (ft)")
brk <- seq(1875, 1974, 10)
abline(v = brk, lty = 2)
for (i in 1:9) segments(brk[i], mean_10yr[i], brk[i + 1], lwd = 2)
```

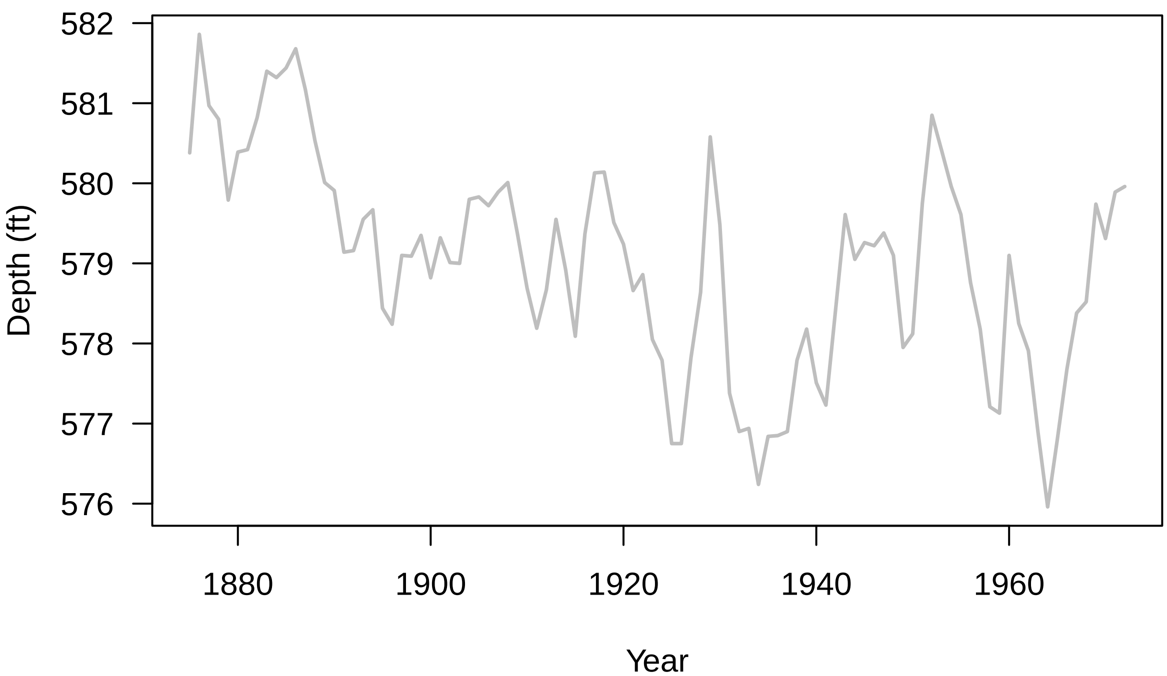


Obviously, this is a very rough estimate of the trend. Next we apply a moving average filter to estimate the trend.

A **Moving average smoother** estimates the trend at time t by averaging the current observation and the q either side. That is

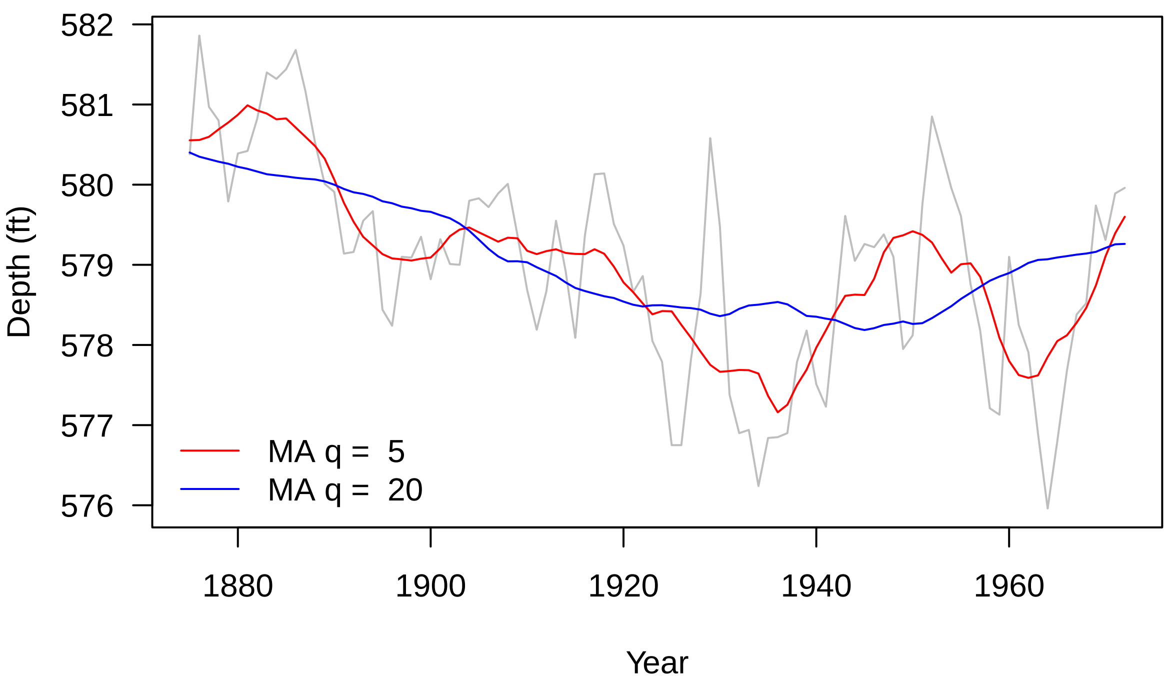
$$\hat{\mu}_t = \frac{1}{2q+1} \sum_{j=-q}^{q} y_{t-j}$$

```
plot(LakeHuron, las = 1, col = "gray", xlab = "Year",
     ylab = "Depth (ft)", lwd = 1.8)
```



```r
### This R function is taken from Donald Percival's Time Series Analysis course (UW Stat 519)
filter.with.padding <- function(x, the.filter, iter = 1){
  q <- (length(the.filter) - 1) / 2
  n <- length(x)
  w <- filter(c(rep(x[1], q), x, rep(x[n], q)),
              the.filter)[(q + 1):(q + n)]
  if(iter > 1) for(i in 2:iter)
    w <- filter(c(rep(w[1], q), w, rep(w[n], q)),
                the.filter)[(q + 1):(q + n)]
  return(w)
}

plot(yr, LakeHuron, col = "gray", xlab = "Year", ylab = "Depth (ft)",
     type = "l", main = "", las = 1)
MA.5 <- filter.with.padding(LakeHuron, rep(1 / 11, 11))
lines(yr, MA.5, col = "red")
MA.20 <- filter.with.padding(LakeHuron, rep(1 / 41, 41))
lines(yr, MA.20, col = "blue")
legend("bottomleft", legend = paste("MA q = ", c(5, 20)),
       col = c("red", "blue"), bty = "n", lty = 1)
```



Exponential smoothing

Let $\alpha \in [0, 1]$ be some fixed constant, defined

$$\hat{\mu}_t = \begin{cases} Y_1 & \text{if } t = 1; \\ \alpha Y_t + (1-\alpha)\hat{\mu}_{t-1} & t = 2, \cdots T. \end{cases}$$

For $t = 2, \cdots, T$, we can rewrite $\hat{\mu}_t$ as

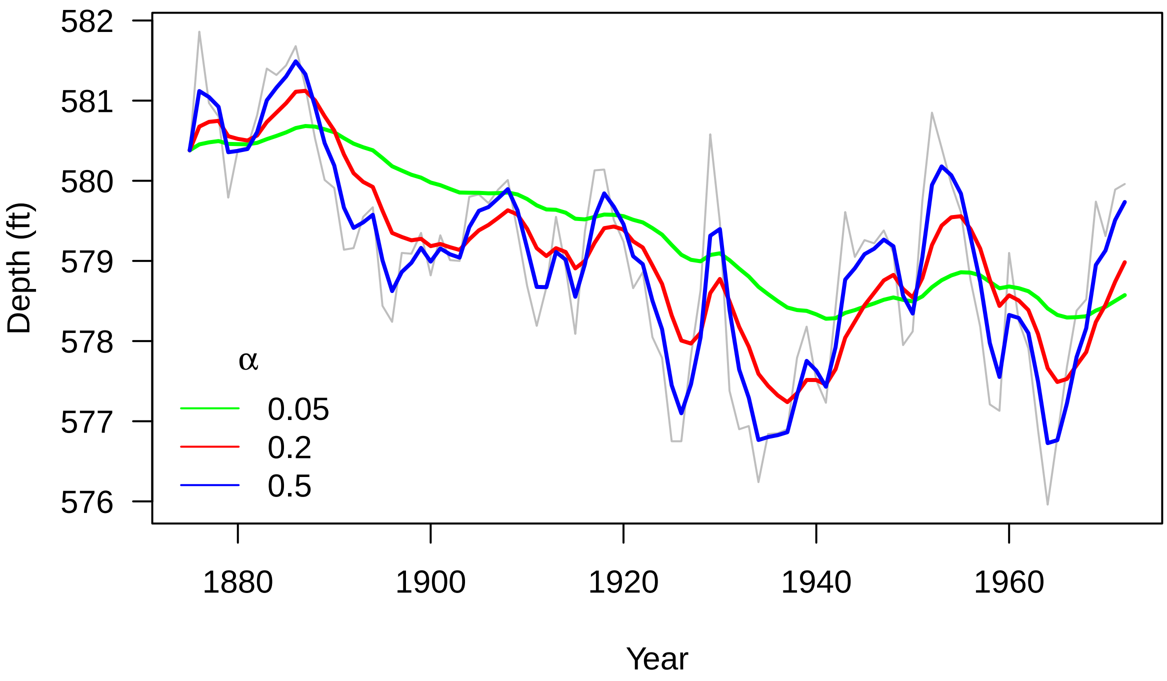
$$\sum_{j=0}^{t-2} \alpha(1-\alpha)^j Y_{t-j} + (1-\alpha)^{t-1} Y_1.$$

⇒ it is a one-sided MA filter with exponentially decreasing weights. One can alter α to control the amounts of smoothing.

```
### This R function is taken from Donald Percival's Time Series Analysis course (UW Stat 519)
exp.smoothing <- function(y, alpha = 0.2){
  n <- length(y)
  mu.hat <- rep(y[1], n)
  if(n > 1) for(i in 2:n) mu.hat[i] <- alpha * y[i] + (1 - alpha) * mu.hat[i - 1]
  return(mu.hat)
}

alpha <- c(0.05, 0.2, 0.5)

plot(yr, LakeHuron, col = "gray", xlab = "Year", type = "l", ylab = "Depth (ft)",
     main = "", las = 1)
ys_0.05 <- exp.smoothing(LakeHuron, alpha = 0.05)
ys_0.2 <- exp.smoothing(LakeHuron, alpha = 0.2)
ys_0.5 <- exp.smoothing(LakeHuron, alpha = 0.5)
lines(yr, ys_0.05, col = "green", lwd = 2)
lines(yr, ys_0.2, col = "red", lwd = 2)
lines(yr, ys_0.5, col = "blue", lwd = 2)
legend("bottomleft", legend = alpha, title = expression(alpha),
       col = c("green", "red", "blue"), bty = "n", lty = 1)
```



Seasonal Component Estimation

Now let's consider the situation where a time series consists of a seasonal component only (assuming the trend has been estimated/removed). That is

$$Y_t = s_t + \eta_t.$$

with $\{s_t\}$ having period d (i.e., $s_t = s_{t+jd}$ for all integers j and t). $\sum_{t=1}^{d} s_t = 0$ and $\mathbb{E}[\eta_t] = 0$. We can use harmonic regression or a seasonal factor model to estimate the seasonal components.

Harmonic Regression

A harmonic regression model has the form

$$s_t = \sum_{j=1}^{k} A_k \cos(2\pi f_j + \phi_j).$$

For each $j = 1, \cdots, k$:

- $A_j > 0$ is the *amplitude* of the jth cosine wave.
- f_j controls the the *frequency* of the j-th cosine wave (how often waves repeats).
- $\phi_j \in [-\pi, \pi]$ is the *phase* of the j-th wave (where it starts)

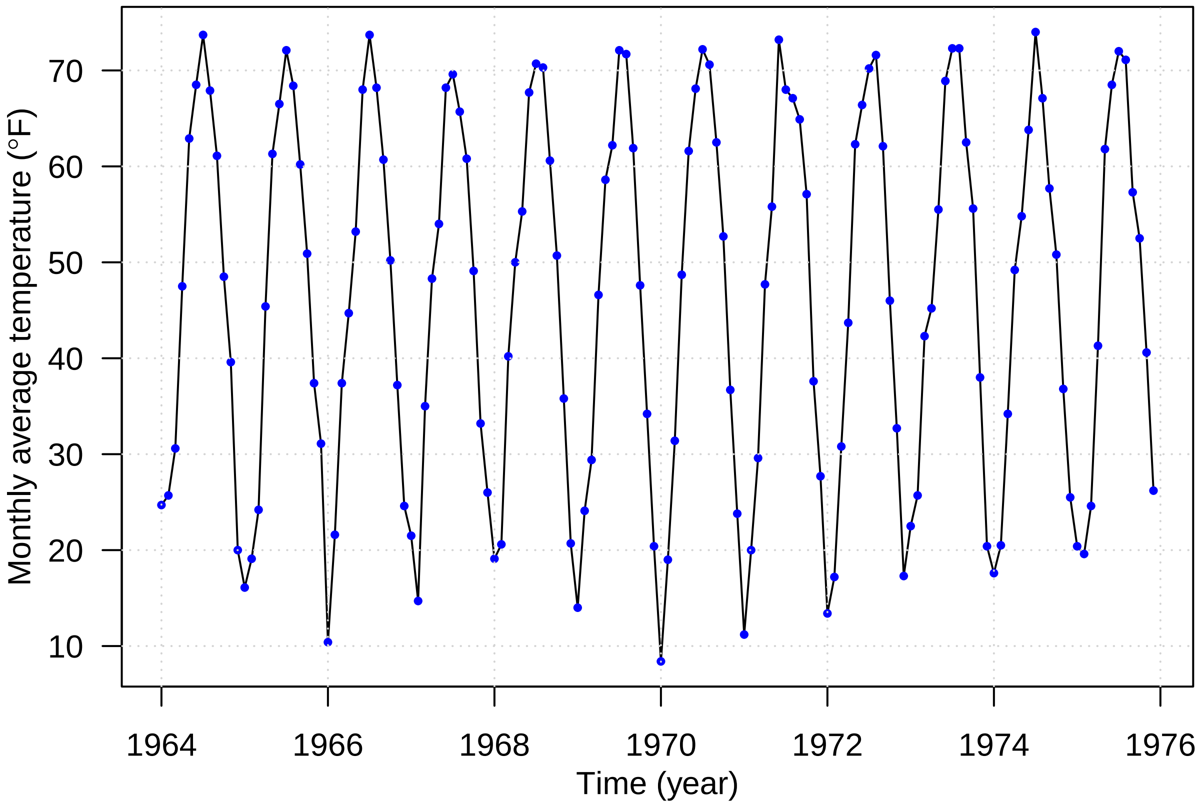
The above can be expressed as

$$\sum_{j=1}^{k} \left(\beta_{1j} \cos(2\pi f_j) + \beta_{2j} \sin(2\pi f_j)\right),$$

where $\beta_{1j} = A_j \cos(\phi_j)$ and $\beta_{2j} = A_j \sin(\phi_j)$. Therefore, if the frequencies $\{f_j\}_{j=1}^k$ are known, we can use regression techniques to estimate the parameters $\{\beta_{1j}, \beta_{2j}\}_{j=1}^k$ by treating $\{\cos(2\pi f_j)\}_{j=1}^k$ and $\{\sin(2\pi f_j)\}_{j=1}^k$ as predictor variables.

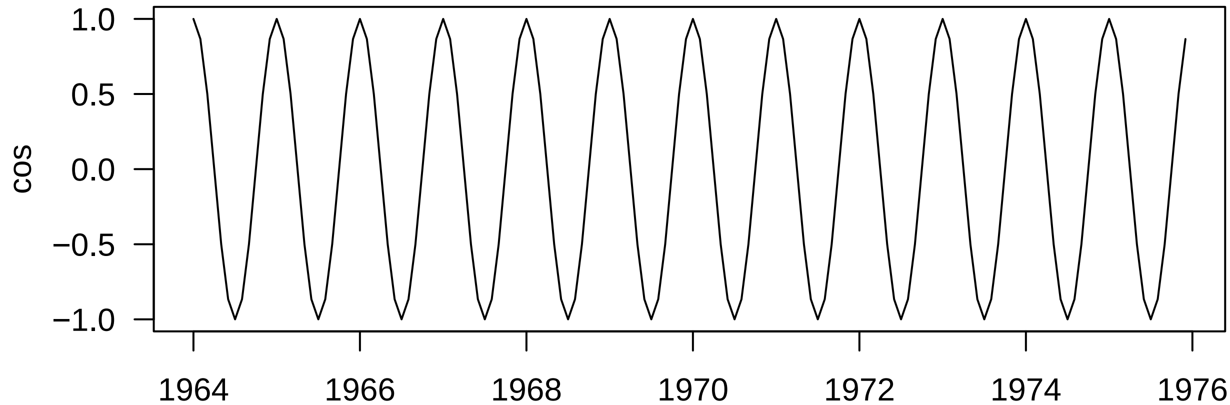
Let's use the monthly average temperature (in degrees Fahrenheit) recorded in Dubuque, IA from Jan. 1964 - Dec. 1975.

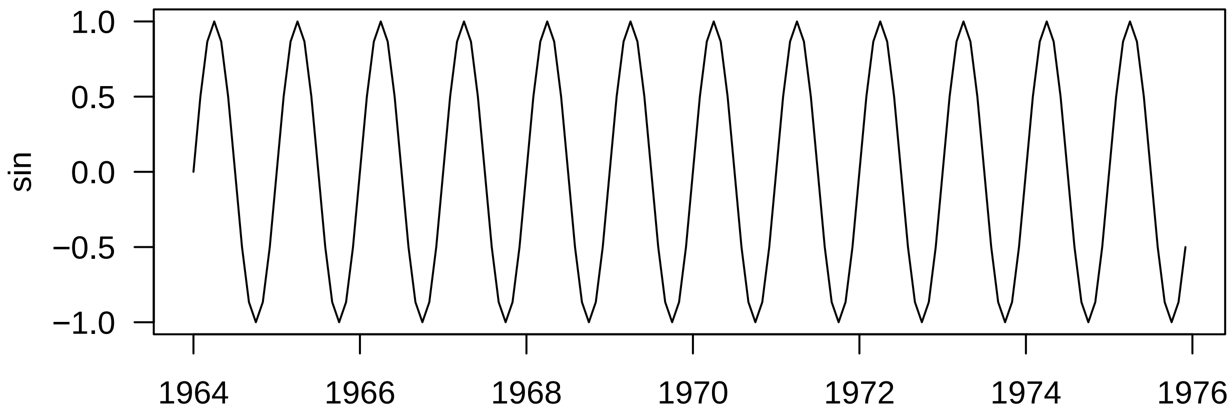
```r
library(TSA)
data(tempdub)
time <- as.numeric(time(tempdub))
par(mar = c(4, 4, 0.8, 0.6))
plot(time, tempdub, type = "l", las = 1, xlab = "", ylab = "")
points(time, tempdub, pch = 16, col = "blue", cex = 0.6)
grid()
mtext("Time (year)", side = 1, line = 2)
mtext(expression(paste("Monthly average temperature (", degree, "F)")), side = 2, line = 2)
```



First, we need to set up the harmonics (assuming yearly cycle)

```
harmonics <- harmonic(tempdub, 1)
time <- as.numeric(time(tempdub))
par(mfrow = c(2, 1), las = 1, mar = c(2, 4, 0.8, 0.6))
plot(time, harmonics[, 1], type = "l", ylab = "cos")
plot(time, harmonics[, 2], type = "l", ylab = "sin")
```



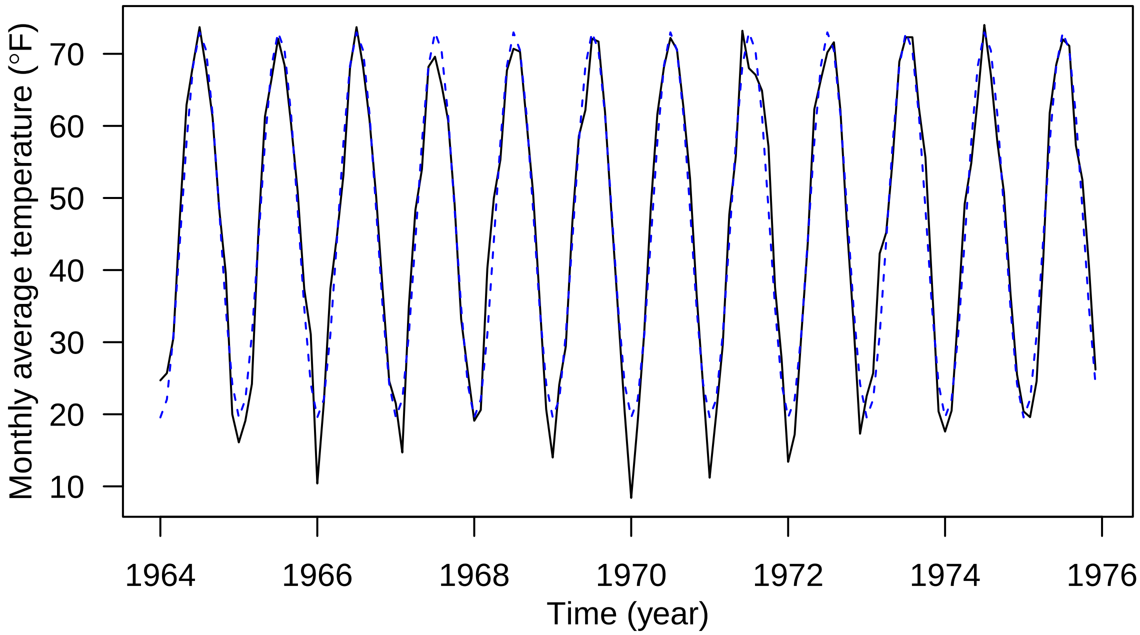


Next, we perform a linear regression using the harmonics we just created as the predictors

```
harReg <- lm(tempdub ~ harmonics)
summary(harReg)
```

```
##
## Call:
## lm(formula = tempdub ~ harmonics)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.1580  -2.2756  -0.1457   2.3754  11.2671
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              46.2660     0.3088 149.816  < 2e-16 ***
## harmonicscos(2*pi*t) -26.7079     0.4367 -61.154  < 2e-16 ***
## harmonicssin(2*pi*t)  -2.1697     0.4367  -4.968 1.93e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.706 on 141 degrees of freedom
## Multiple R-squared:  0.9639, Adjusted R-squared:  0.9634
## F-statistic:  1882 on 2 and 141 DF,  p-value: < 2.2e-16
```

```r
plot(time, tempdub, type = "l", las = 1, xlab = "", ylab = "")
mtext("Time (year)", side = 1, line = 2)
mtext(expression(paste("Monthly average temperature (", degree, "F)")), side = 2, line = 2)
time <- as.numeric(time(tempdub))
lines(time, harReg$fitted.values, col = "blue", lty = 2)
```



Seasonal factors

Harmonic regression assume the seasonal pattern has a regular shape, i.e. the height of the peaks is the same as the depth of the troughs. Assuming the seasonal pattern repeats itself every d time points, a less restrictive approach is to model it as

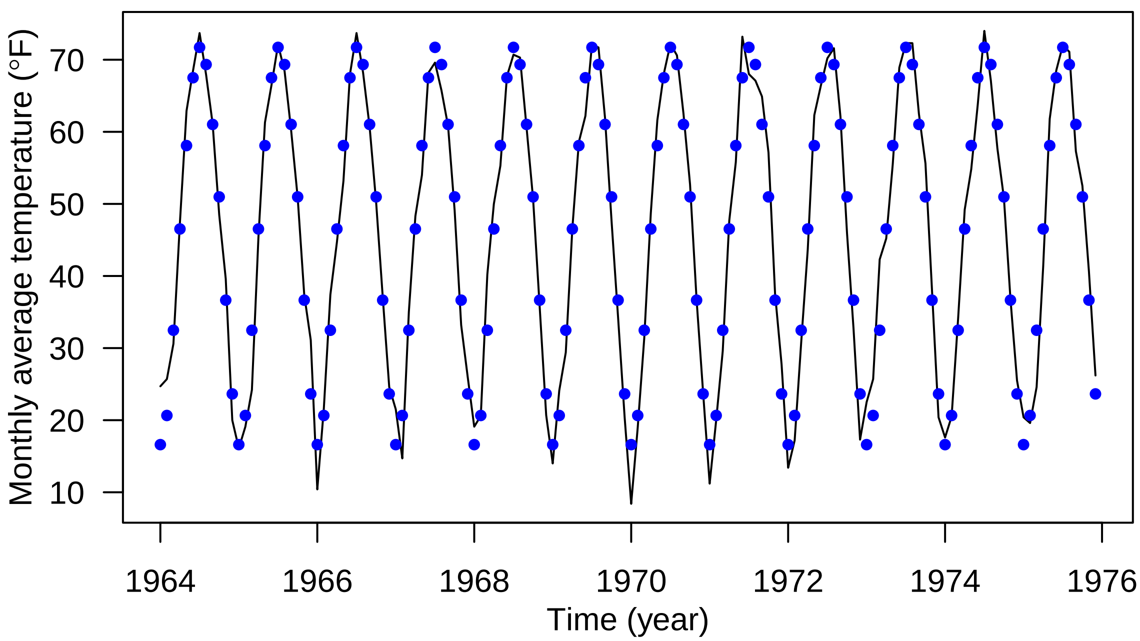
$$s_t = \begin{cases} \beta_1 & \text{for } t = 1, 1+d, 1+2d, \cdots; \\ \beta_2 & \text{for } t = 2, 2+d, 2+2d, \cdots; \\ \vdots & \vdots; \\ \beta_d & \text{for } t = d, 2d, 3d, \cdots. \end{cases}$$

```r
month = season(tempdub)
season_means <- lm(tempdub ~ month - 1)
summary(season_means)
```

```
##
## Call:
## lm(formula = tempdub ~ month - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2750 -2.2479  0.1125  1.8896  9.8250
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## monthJanuary      16.608      0.987   16.83   <2e-16 ***
## monthFebruary     20.650      0.987   20.92   <2e-16 ***
## monthMarch        32.475      0.987   32.90   <2e-16 ***
## monthApril        46.525      0.987   47.14   <2e-16 ***
## monthMay          58.092      0.987   58.86   <2e-16 ***
## monthJune         67.500      0.987   68.39   <2e-16 ***
## monthJuly         71.717      0.987   72.66   <2e-16 ***
## monthAugust       69.333      0.987   70.25   <2e-16 ***
## monthSeptember    61.025      0.987   61.83   <2e-16 ***
## monthOctober      50.975      0.987   51.65   <2e-16 ***
## monthNovember     36.650      0.987   37.13   <2e-16 ***
## monthDecember     23.642      0.987   23.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.419 on 132 degrees of freedom
## Multiple R-squared:  0.9957, Adjusted R-squared:  0.9953
## F-statistic:  2569 on 12 and 132 DF,  p-value: < 2.2e-16
```

```r
plot(time, tempdub, type = "l", las = 1, xlab = "", ylab = "")
mtext("Time (year)", side = 1, line = 2)
mtext(expression(paste("Monthly average temperature (", degree, "F)")), side = 2, line = 2)
points(time, season_means$fitted.values, col = "blue", pch = 16, cex = 0.8)
```



Let's put trend and seasonal variation together

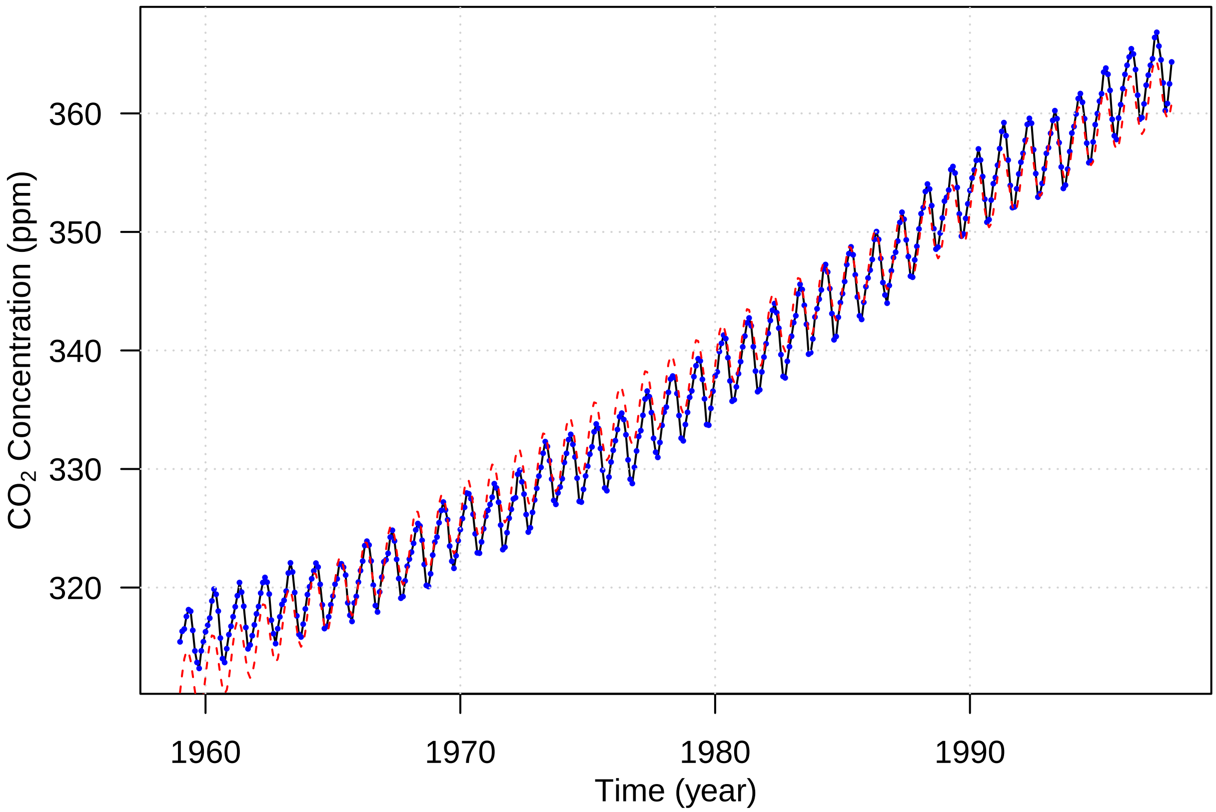
Here we using the CO_2 concentration time series is an example. First, we can perform a linear regression with both time and the harmonics as the covariates.

```r
time <- as.numeric(time(co2))
harmonics <- harmonic(co2, 1)

lm_trendSeason <- lm(co2 ~ time + harmonics)
summary(lm_trendSeason)
```

```
##
## Call:
## lm(formula = co2 ~ time + harmonics)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.433 -1.323 -0.282  1.221  4.615
##
## Coefficients:
##                       Estimate Std. Error  t value Pr(>|t|)
## (Intercept)          -2.256e+03  1.391e+01 -162.155  < 2e-16 ***
## time                  1.311e+00  7.033e-03  186.382  < 2e-16 ***
## harmonicscos(2*pi*t) -3.889e-01  1.120e-01   -3.474  0.00056 ***
## harmonicssin(2*pi*t)  2.772e+00  1.120e-01   24.760  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.712 on 464 degrees of freedom
## Multiple R-squared:  0.987,  Adjusted R-squared:  0.9869
## F-statistic: 1.173e+04 on 3 and 464 DF,  p-value: < 2.2e-16
```

```r
par(mar = c(3.8, 4, 0.8, 0.6))
plot(time, co2, type = "l", las = 1, xlab = "", ylab = "")
points(co2, col = "blue", pch = 16, cex = 0.4)
mtext("Time (year)", side = 1, line = 2)
mtext(expression(paste("CO"[2], " Concentration (ppm)")), side = 2, line = 2.5)
grid()
lines(time, lm_trendSeason$fitted.values, col = "red", lty = 2)
```

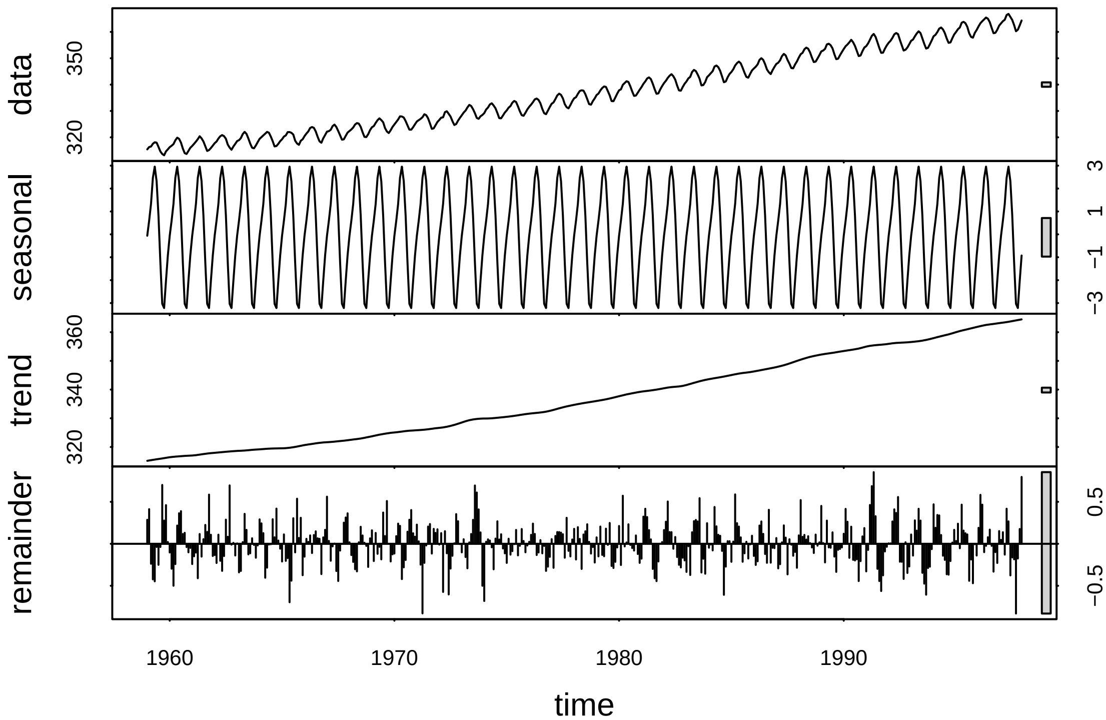


Next we are going to take a quick look of an 'algorithm' to do the decomposition.

STL Decomposition

STL (Seasonal and Trend decomposition using Loess) is a versatile and robust method for decomposing time series. The STL method was developed by Cleveland et al. (1990). Below we show an example of applying STL to Mauna Loa atmospheric CO2 concentration monthly time series data.

```
# Seasonal and Trend decomposition using Loess (STL)
par(mar = c(4, 3.6, 0.8, 0.6))
stl <- stl(co2, s.window = "periodic")
plot(stl, las = 1)
```



References

Cleveland, Robert B, William S Cleveland, Jean E McRae, and Irma Terpenning. 1990. "STL: A Seasonal-Trend Decomposition." *J. Off. Stat* 6 (1): 3–73.