STA 368 HW #2

Jack Liddicoat

2024-03-06

Problem 1

(a)

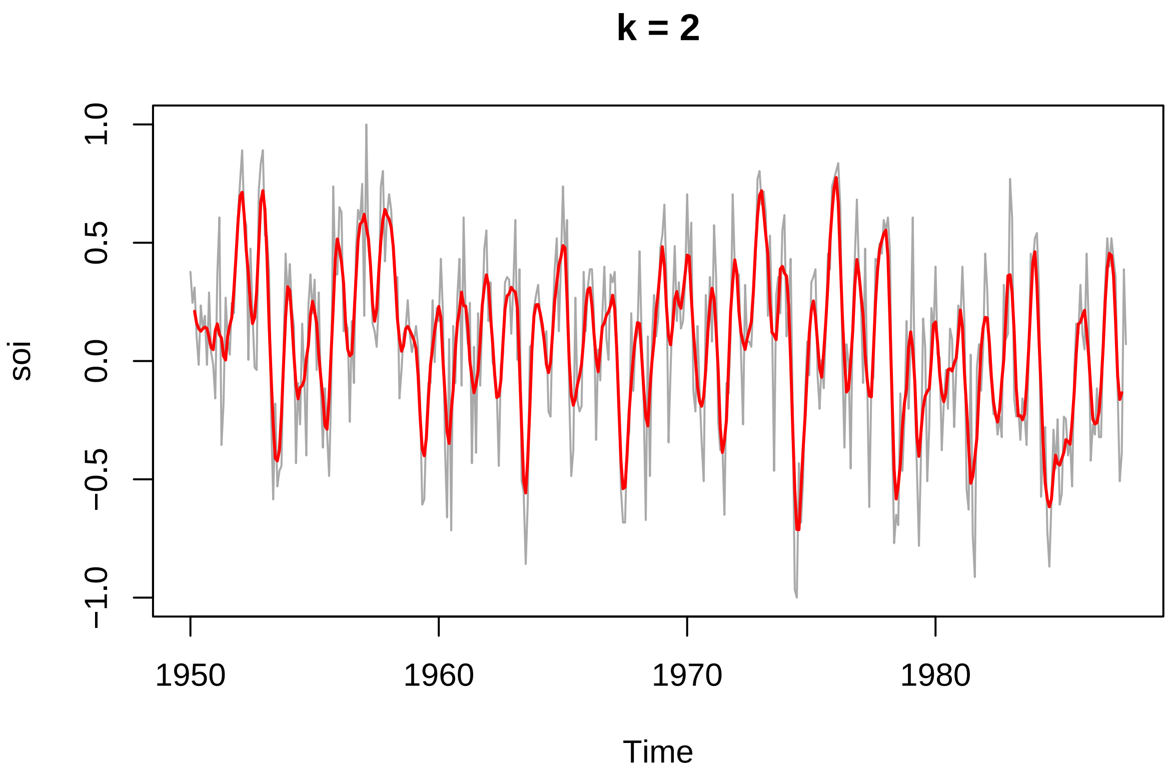
The equation for a moving average smoother is:

$$m_t = \sum_{j=-k}^{k} a_j x_{t-j}$$

a_j represents the constant weight of the smoother. The sum of the weights a_j must be equal to one, otherwise the data is not symmetric. The equation is averaging $-k$ points away from x and $+k$ points away from x in order to depict a smoother representation of variable data, which allows for better visualization and interpretation of a given time-series.

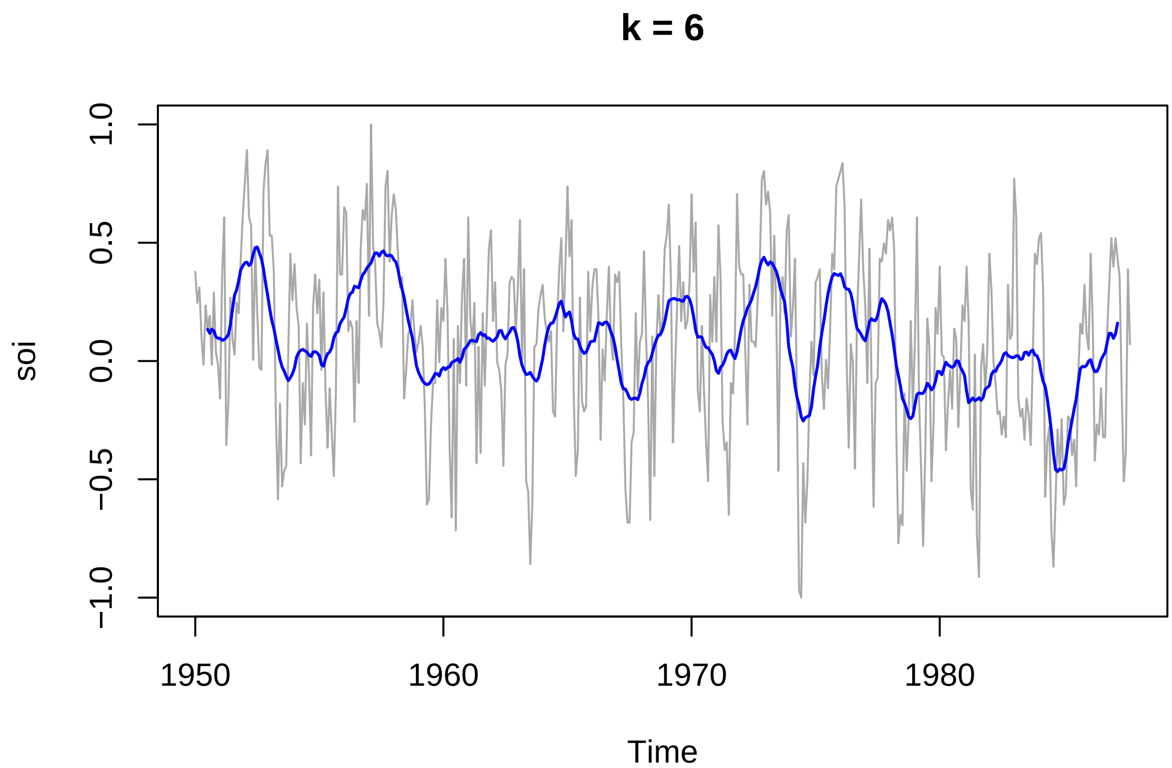
(b)

```
a = c(.5, rep(1, 3), .5)/4 # sums to 1
soif = filter(soi, sides = 2, filter = a)
plot(soi, col = 'darkgrey', main = 'k = 2')
lines(soif, lwd = 1.5, col = 'red')
```



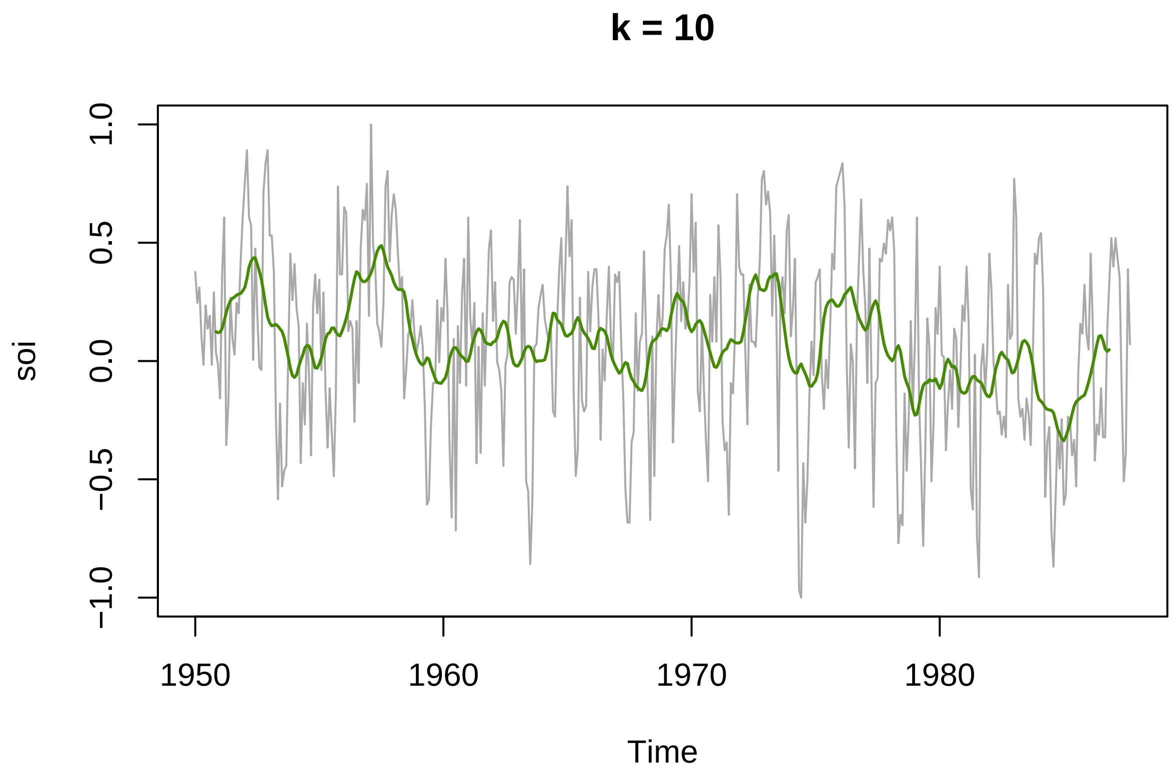
(c)

```
a = c(.5, rep(1, 11), .5)/12 # sums to 1
soif = filter(soi, sides = 2, filter = a)
plot(soi, col = 'darkgrey', main = 'k = 6')
lines(soif, lwd = 1.5, col = 'blue')
```



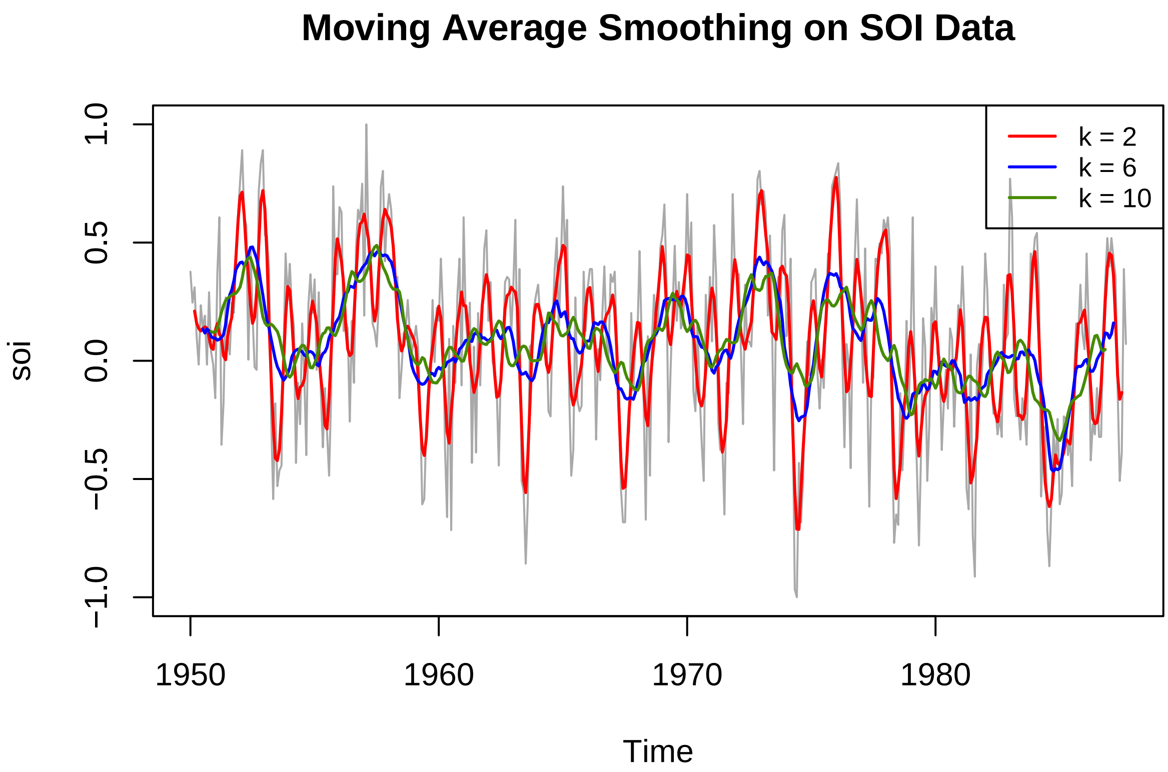
(d)

```
a = c(.5, rep(1, 19), .5)/20 # sums to 1
soif = filter(soi, sides = 2, filter = a)
plot(soi, col = 'darkgrey', main = 'k = 10')
lines(soif, lwd = 1.5, col = 'chartreuse4')
```



(e)

```
a = c(.5, rep(1, 3), .5)/4 # k = 2
b = c(.5, rep(1,11), .5)/12 # k = 6
c = c(.5, rep(1, 19), .5)/20 # k = 10
soif_a = filter(soi, sides = 2, filter = a)
soif_b = filter(soi, sides = 2, filter = b)
soif_c = filter(soi, sides = 2, filter = c)
plot(soi, col = "darkgrey", main = "Moving Average Smoothing on SOI Data")
lines(soif_a, col = 'red', lwd = 1.5)
lines(soif_b, col = 'blue', lwd = 1.5)
lines(soif_c, col = 'chartreuse4', lwd = 1.5)
legend("topright", legend = c("k = 2", "k = 6", "k = 10"), col = c('red', 'blue', 'chart
       lwd = 1.5, cex = .8)
```



Problem 2

(a)

The equation for a kernel smoother is:

$$m_t = \sum_{i=1}^{n} w_i(t) x_i$$

where

$$w_i(t) = K(\frac{t-i}{b}) / \sum_{j=1}^{n} K(\frac{t-j}{n})$$

Unlike the moving average where we pick the weights for the data points that we want to average, the kernel smoother is the shape of the function that is used to average the neighboring points. So, the weight of the points is defined by the kernel. We can pick the shape of the function that we use to average the points. For example, we can pick a Gaussian kernel, a uniform kernel, etc. Similar to the moving average, it smooths data with lots of variance for better visualization and interpretation.

(b)

The linear kernal smoother is the equation:

$$k(x, y) = x^T \cdot y$$

The polynomial kernal smoother is the equation:

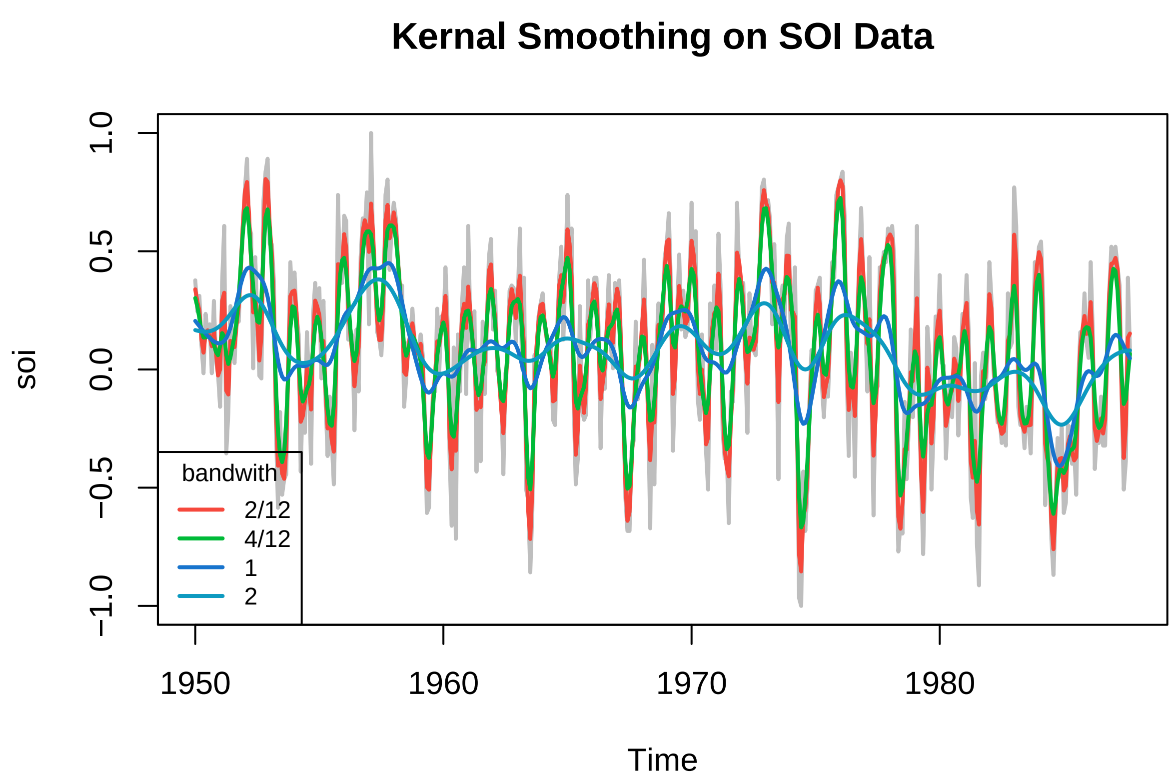
$$k(x, y) = (x^T \cdot y + 1)^p$$

where p is the degree of the polynomial.

https://www.researchgate.net/figure/Formulation-for-kernel-function_tbl1_337183572

(c)

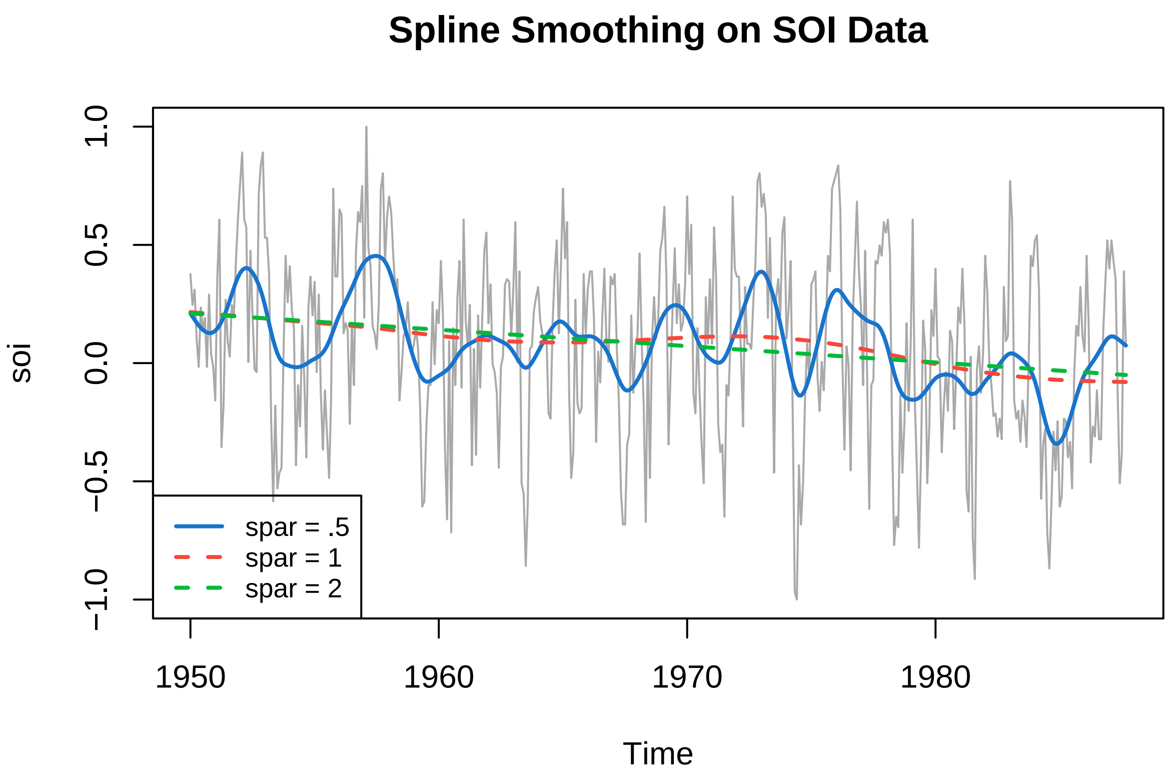
```
plot(soi, col = 'grey', lwd = 2, main = "Kernal Smoothing on SOI Data")
lines(ksmooth(time(soi), soi, "normal", bandwidth=2/12), lwd=2, col=2, type = 'l')
lines(ksmooth(time(soi), soi, "normal", bandwidth=4/12), lwd=2, col=3)
lines(ksmooth(time(soi), soi, "normal", bandwidth=1), lwd=2, col=4)
lines(ksmooth(time(soi), soi, "normal", bandwidth=2), lwd=2, col=5)
legend("bottomleft", title = 'bandwith', legend = c('2/12', '4/12', '1', '2'),
       lwd = 2, col = c(2, 3, 4, 5), cex = .75)
```



Problem 3

(a)

```
plot(soi, col = 'darkgrey', main = "Spline Smoothing on SOI Data")
lines(smooth.spline(time(soi), soi, spar=.5), lwd=2, col=4)
lines(smooth.spline(time(soi), soi, spar= 1), lty=2, lwd=2, col=2)
lines(smooth.spline(time(soi), soi, spar= 2), lty=2, lwd=2, col=3)
legend("bottomleft", legend = c('spar = .5', 'spar = 1', 'spar = 2'), lwd = 2,
       col = c(4, 2, 3), cex = .8, lty = c(1, 2, 2))
```



(b)

The *spar* is monotonically related to the degree of smoothness, which is typically denoted as λ and is the coefficient of the integral of the accelerator/decelerator, m_t'', squared. As spar decreases (it has a lower bound of 0) the smoothing gets more rigid, resembling the original data more closely. As spar increases, the degree of smoothness also increases, and thus the smoothing line is flatter and less rigid.