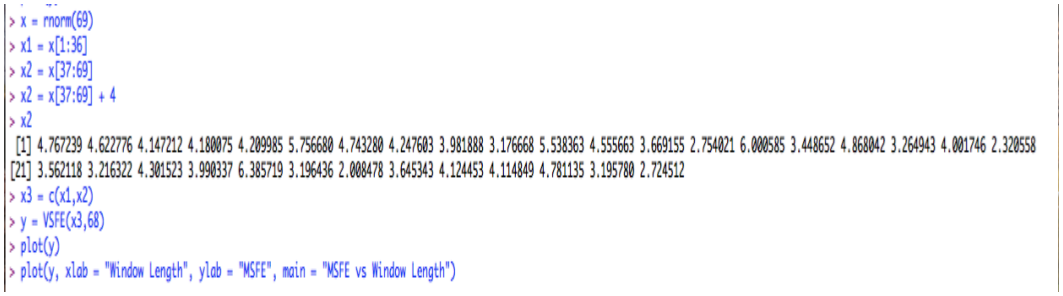
Marcus Crowder
STA-3920
Learning Exercise 11

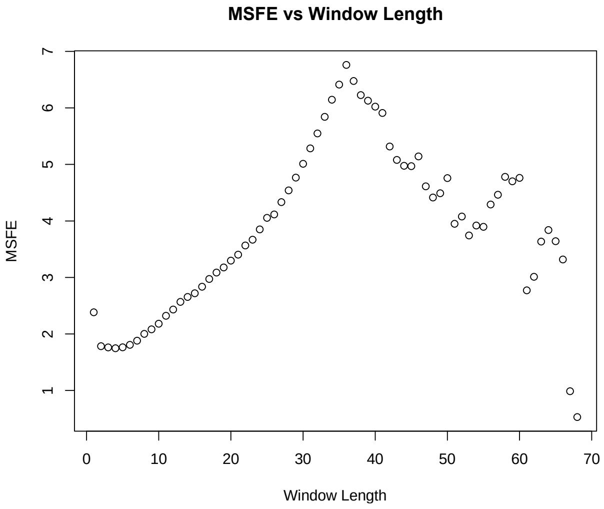
11.2 In R, create a vector with 69 simulated values from the standard Normal distribution. Add a value of 4.0 to the points starting at observation number 37. This will give you the same kind of data as I created in Excel. Using R, apply our VSFE function to that series. Show the graph of the msfe versus window length and report the value which minimized the msfe. Do this again 3 times on fresh random data, showing the output. Were the results stable?

To Accomplish this I first created the the simulated values from the standard normal distribution with x = rnorm(60) Then I sliced the data set and added + 4 to the values from 37:69 and concatenated both x1 and x2 for the final data set.

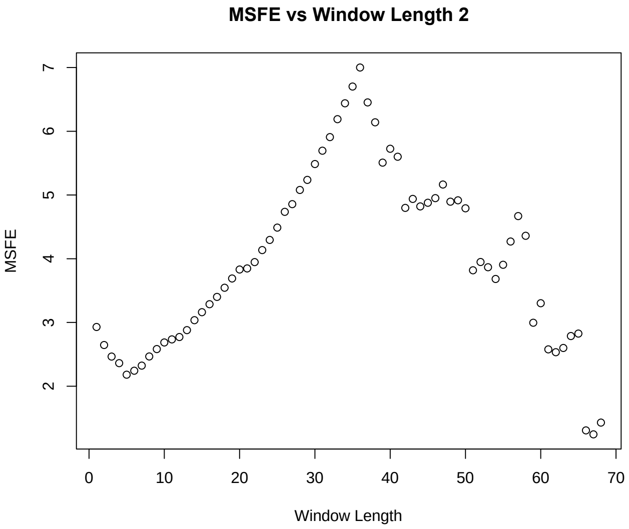


```
> x = rnorm(69)
> x1 = x[1:36]
> x2 = x[37:69]
> x2 = x[37:69] + 4
> x2
 [1] 4.767239 4.622776 4.147212 4.180075 4.209985 5.756680 4.743280 4.247603 3.981888 3.176668 5.538363 4.555663 3.669155 2.754021 6.000585 3.448652 4.868042 3.264943 4.001746 2.320558
[21] 3.562118 3.216322 4.301523 3.990337 6.385719 3.196436 2.008478 3.645343 4.124453 4.114849 4.781135 3.195780 2.724512
> x3 = c(x1,x2)
> y = VSFE(x3,68)
> plot(y)
> plot(y, xlab = "Window Length", ylab = "MSFE", main = "MSFE vs Window Length")
```

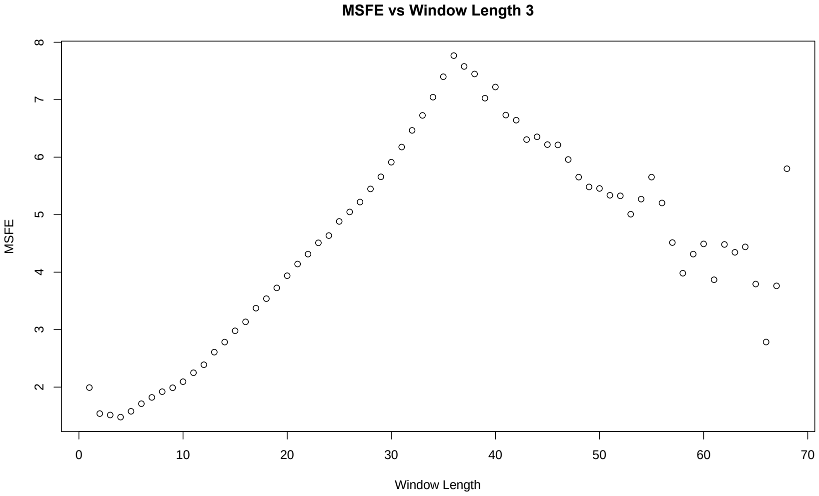
The value that minimized msfe was surprisingly on the ends the msfe of 67 and 69 seemed to be the winners of this festival. It is also worth noting how the MSFE values start off small then gradually rise until about window length 35 where it then goes back down. Which is a very interesting pattern.



The previous steps to create a vector were used again for the VSFE function and surprisingly it is a bit different than the first one the winner appears to be value 67 for the lowest MSFE with value 67 being 2nd and 68 being 3rd the values do however display a similar pattern compared to the first MSFE vs Window Length with a decrease in MSFE around window length 35.



The 3rd test trial produced a much higher result for window length 68 which isn't close to the winner which appear to be window length 4. In this trial the beginning window lengths won this party while ending window lengths did not do as well as before.



After 2 tests I believed the results were stable but the 3rd test showed me the light. The results aren't stable the windows lengths with the lowest MSFE values have seem to trade places. However there does seem to be a trend in all of them at around window length 35 where the MSFE appears to be at it's peak then in each example it gradually decreases for the majority of points.

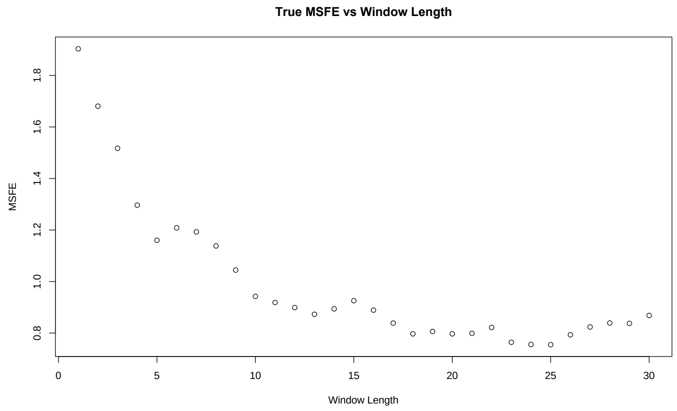
Code:

```
> x1 = x[1:36]
> x2 = x[37:69] + 4
> x3 = c(x1,x2)
> y = VSFE(x3,68)
> plot(y, xlab = "Window Length", ylab = "MSFE", main = "MSFE vs Window Length")
> plot(y, xlab = "Window Length", ylab = "MSFE", main = "MSFE vs Window Length 2")
> x = rnorm(69)
> x1 = x[1:36]
> x2 = x[37:69] + 4
> x3 = c(x1,x2)
> y = VSFE(x3,68)
> plot(y, xlab = "Window Length", ylab = "MSFE", main = "MSFE vs Window Length 3")
>
```

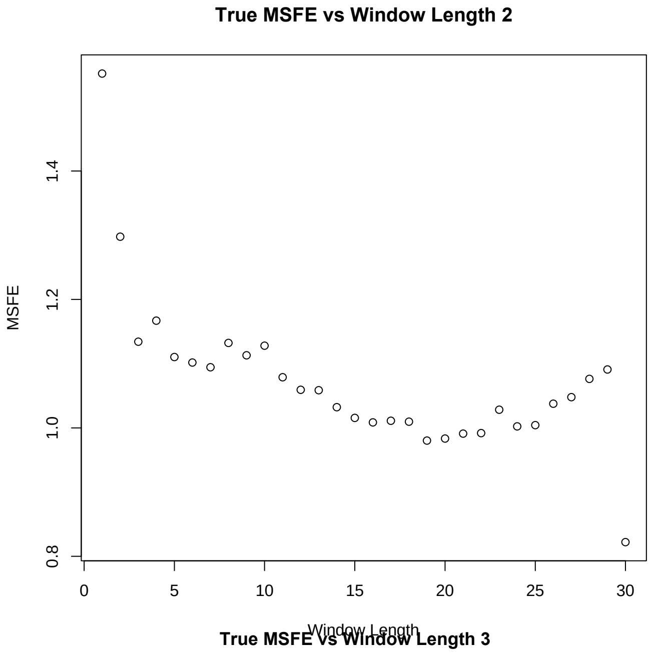
11.3 In Excel or in R, find the true MSFE for all window lengths 1 to 30. Graph that and compare it against the results from the previous question.

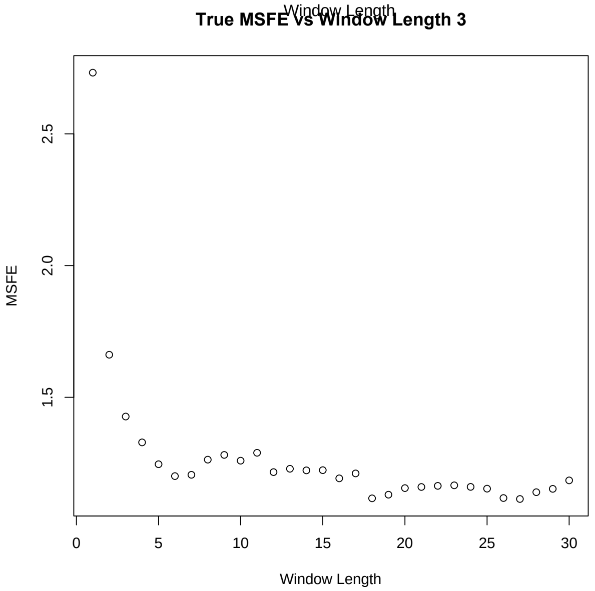
Code:

```
> x4 = rnorm(69)
> y1 = VSFE(x4,30)
> plot(y1, xlab = "Window Length", ylab = "MSFE", main = "True MSFE vs Window Length")
> x4 = rnorm(69)
> y1 = VSFE(x4,30)
> plot(y1, xlab = "Window Length", ylab = "MSFE", main = "True MSFE vs Window Length")
> plot(y1, xlab = "Window Length", ylab = "MSFE", main = "True MSFE vs Window Length
    2")
> x4 = rnorm(69)
> y1 = VSFE(x4,30)
> plot(y1, xlab = "Window Length", ylab
    = "MSFE", main = "True MSFE vs
    Window Length 3")
```


True MSFE vs Window Length

For the True MSFE I did not include the jump from the previous example where +4 was added to values above 37. This graph is noticeably different from the other one first thing being less window lengths (this one only stopping at 30 while previous 68) The other extremely noticeably thing is how the MSFE for the window lengths become smaller over time with some bumps in the road then finally going back up. In the previous example this was not the case as the MSFE started off smaller then continued to rise to the sky until the 30ish mark this can be attributed to the Bias that was created with the jump in the previous question of +4. Also these results seem more stable in 2 of the cases the lowest msfe appeared around window 18 except for the 2nd example where it was 2nd and window length of 30 was the lowest.


True MSFE vs Window Length 2

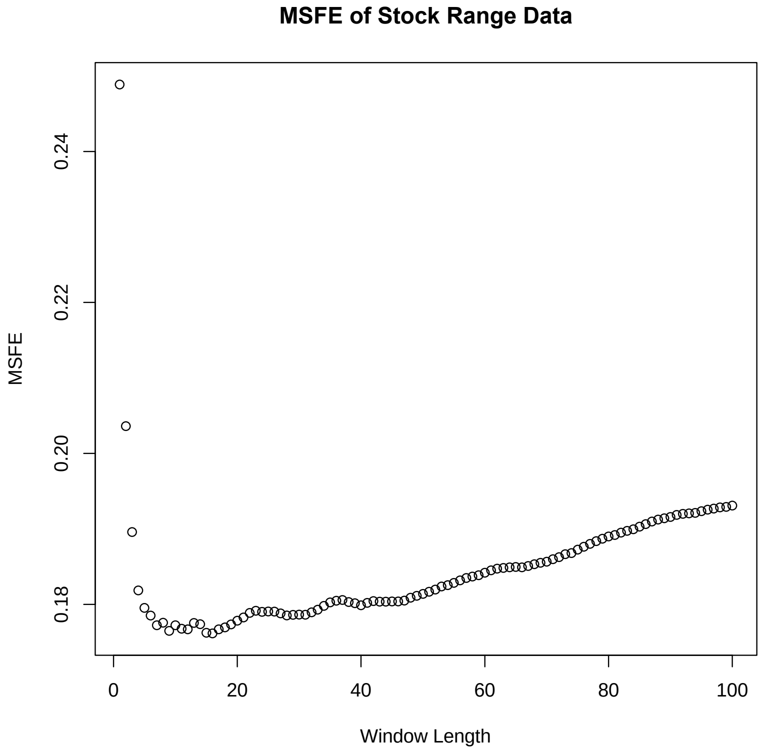

True MSFE vs Window Length 3

11.4 Create a series of simulated standard Normal random variables of length 2500. Experiment with adding different kinds of patterns of mean shifts with the intention of getting a VSFE generated plot that resembles that for the daily range of your data. An example of a mean shift was when 4 was added at time t=37 and beyond, above.

Here is the original VSFE of my Stock Data for MKC

Code:

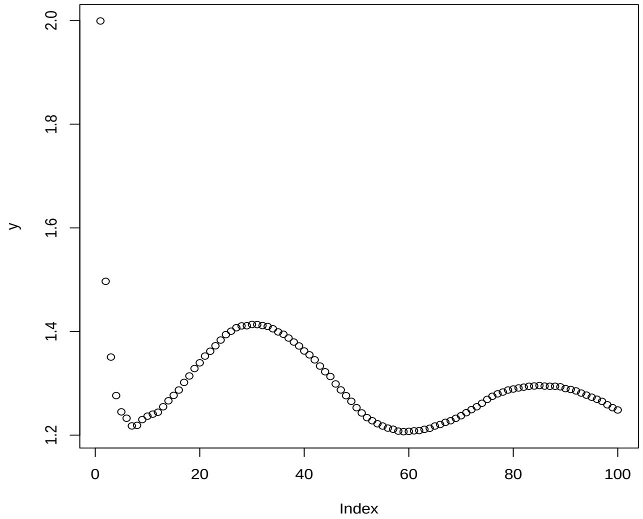
```
> MKC <- read.csv('~/Desktop/MKC.csv')
> b = VSFE(MKC$LGF_range, 100)
> plot(b, xlab="Window Length", ylab= "MSFE", main = "MSFE of Stock Range Data")
```



First Attempt: Following Professor Tatum strategy of repeating 0 and 1 every 25 spaces.

```
> x1 = rep(c(rep(0,25),rep(1,25)),50)
> plot(x1)
> x2 = x1+rnorm(2500)
> plot(x2)
> y = VSFE(x2,100)
> plot(y)
```

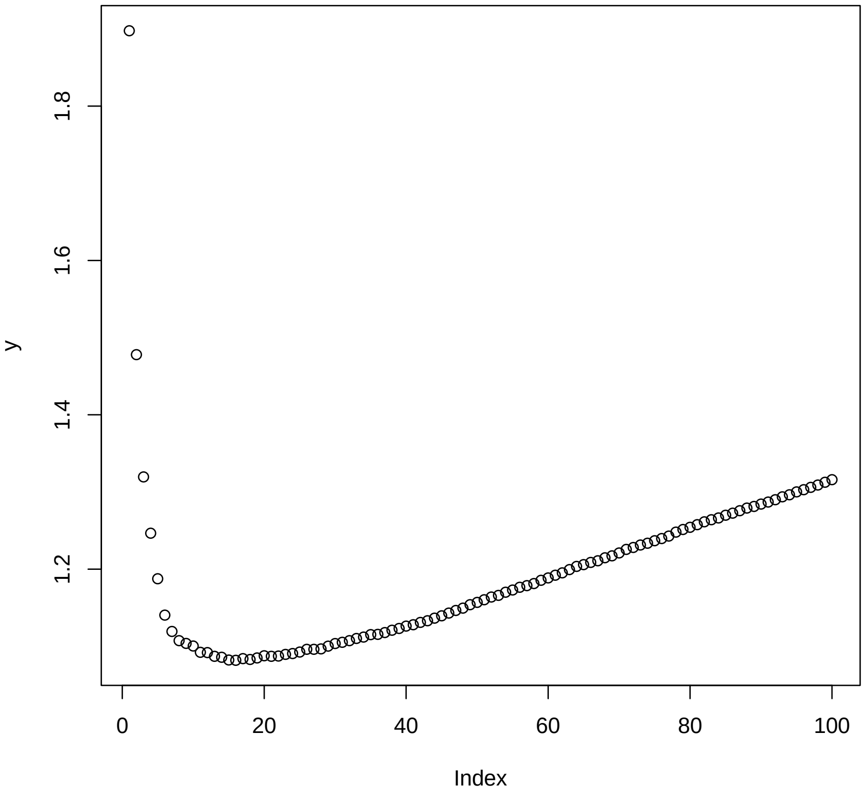
Started off with so much potential but no cigar.



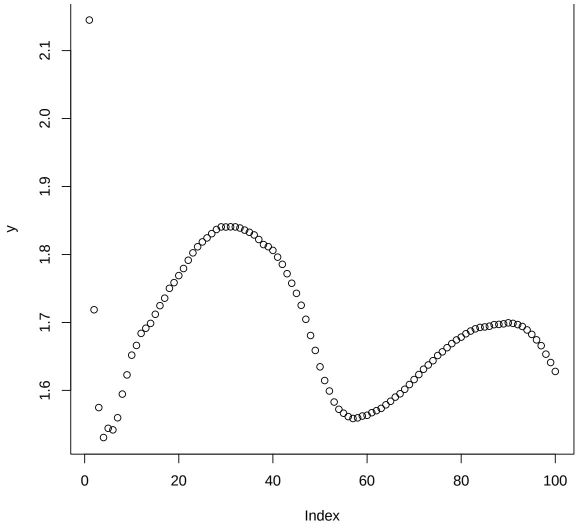
```
> x1 =
rep(c(rep(0,50),rep(1,50)),25)
> x2 = x1+rnorm(2500)
> y = VSFE(x2,100)
> plot(y)
```

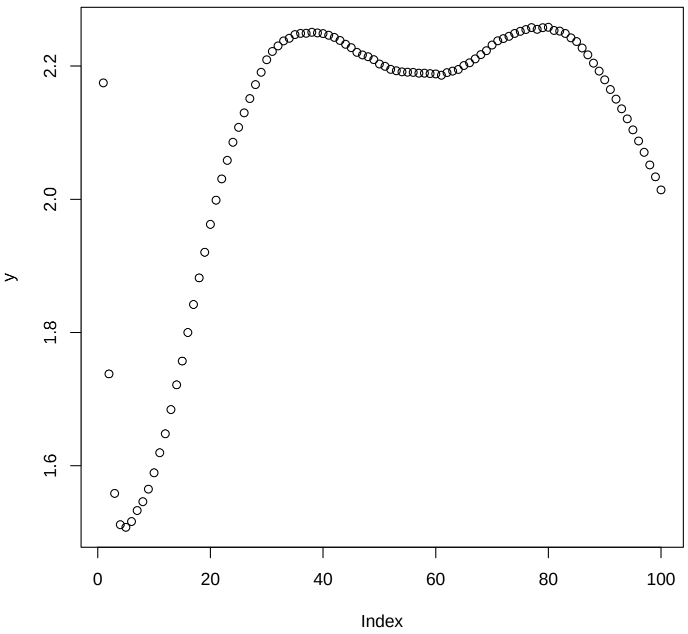
Added 0 to first 50 values then 1 to 2nd 50 for a repition of 25x for this beauty

The plot changed with less hops and seemly the same increasing pattern but there are no curves in the pattern like my original stock data. Also this has a higher msfe but we can work around it



```
> x1 = rep(c(rep(0,20),rep(1,20),rep(2,10)),50)
> x2 = x1+rnorm(2500)
> y = VSFE(x2,100)
> plot(y)
```





```
> x1 = rep(c(rep(0,20),rep(1,20),rep(2,10),rep(1,20),rep(3,20),rep(1,10)),25)
> length(x1)
[1] 2500
> x2 = x1+rnorm(2500)
> y = VSFE(x2,100)
> plot(y)
```

added 0 to the first 20 values then 1 to the other 20 then 2 to 10 then 1 to 20, 3 to 20 and 1 to 10 for a total of 25x or 2500 points which created this monster looks like we are just getting worse so This was fun but the closest was the 2^{nd} example

11.5 Try to come up with your own "story" about a situation involving a bias-variance tradeoff. I am working on the "archer" story. You could try to improve that, or better yet tell a different story. Example: ordering inventory!