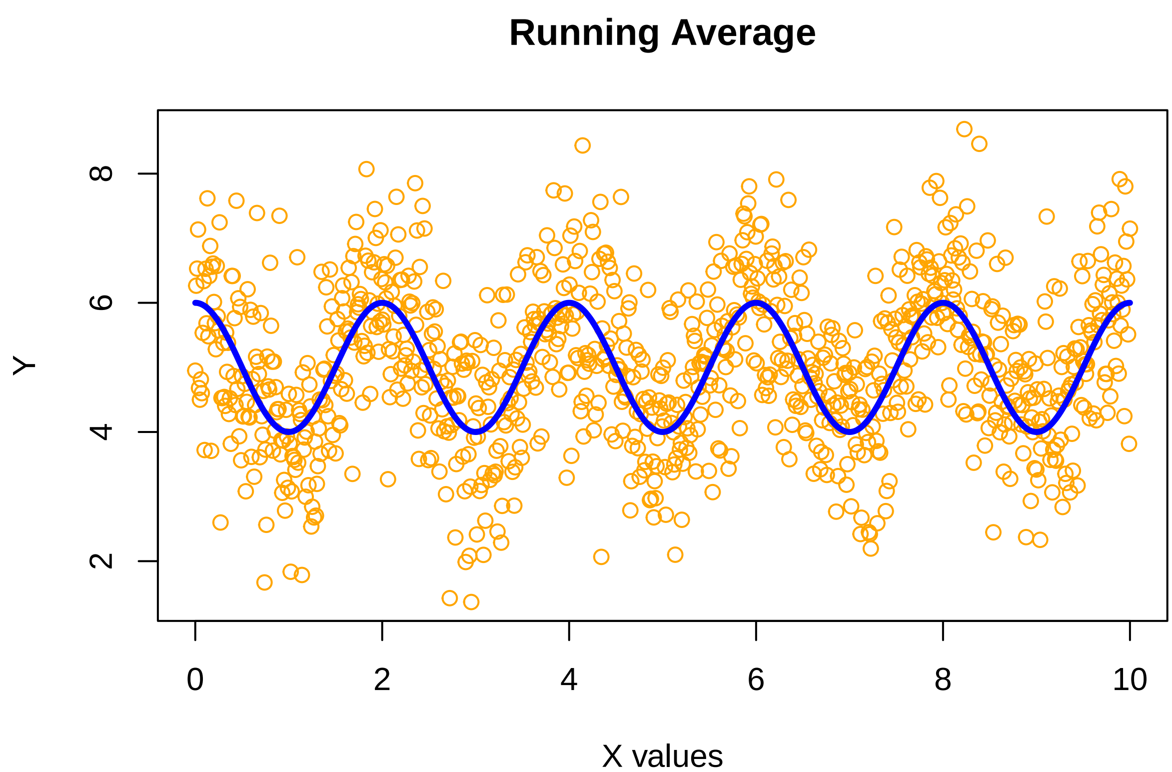
Running Average Implementation & Analysis

Stephen J George

2024-03-13

Let us assume the function (on which we want to perform running average regression) as 5+cos(pi*x)+noise and the domain to be [0,10]

```
n = 1000
x = seq(0,10,length.out=n)
y = 5 + cos(pi*x) + rnorm(n)
#y = 5 + cos(pi*x)
plot(x,y,col="orange",main="Running Average",xlab="X values",ylab="Y")
curve(5+cos(pi*x),min(x),max(x),n,add=TRUE,col="blue",lwd=3)
```

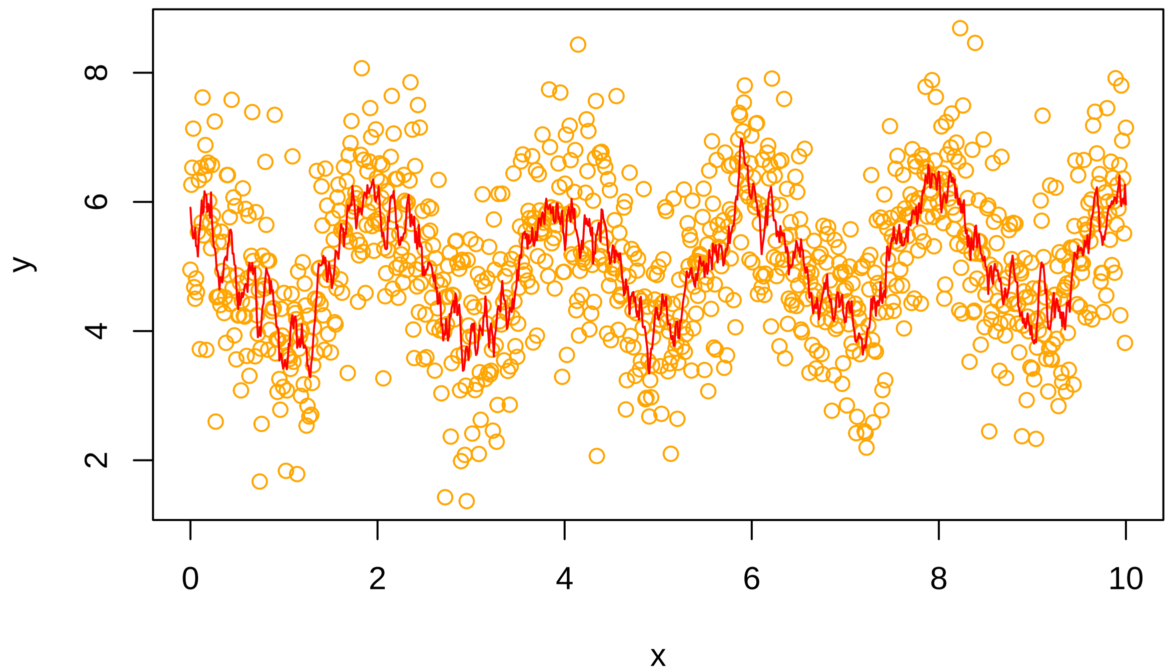


Specifying a predefined width for the (sliding) window (as 0.05).We initialize all the estimates for the n different Y's to 0.Since each point has a window around it, there are n y's to be calculated.

```
w = 0.05
y.est=rep(0,n)
#y estimate for each x[i]
```

```
#store endpoints of window in vector b
for (i in 1:n){
    b=x[i]+w*c(-1,1)
    y.est[i] = mean(y[which((b[1]<=x)&(x<=b[2]))])
}

#plotting horizontal lines for each window
plot(x,y,col="orange")
for (i in 1:n-1){
    # b=x[i]+w*c(-1,1)
    # points(x[i],y.est[i],col="green",pch=20,cex=0.3)
    lines(c(x[i],x[i+1]),c(y.est[i],y.est[i+1]),col="red")
}
```



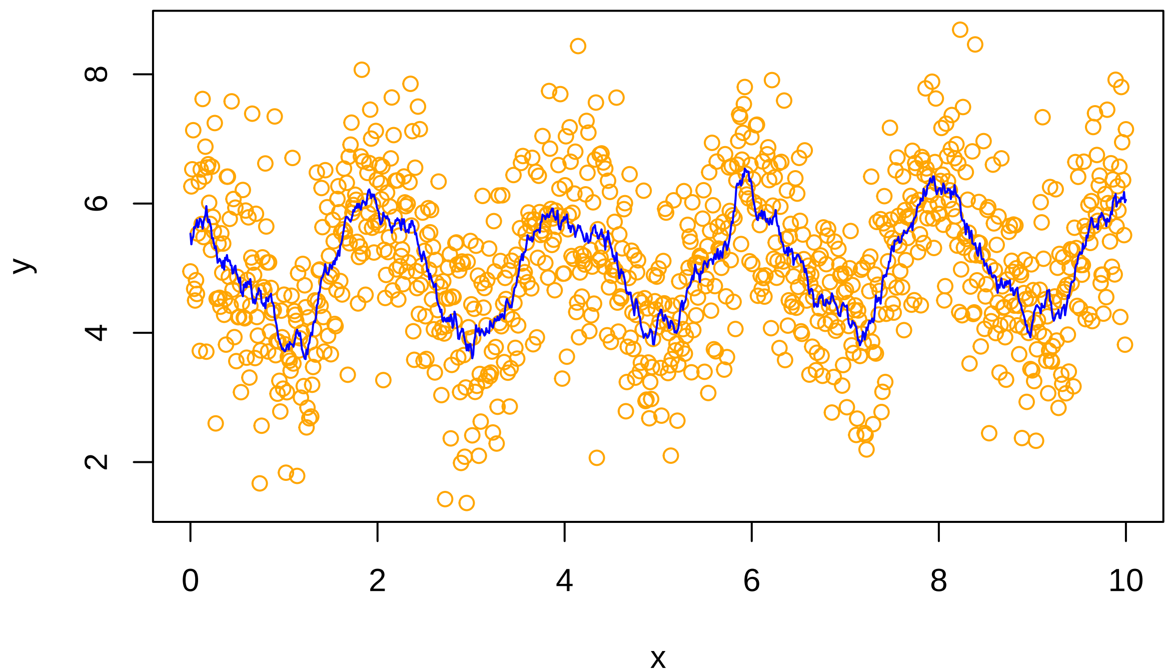
Suppose we now increase the size of the window width to 0.1 and repeat the same procedure as above.

```
w = 0.1
y.est=rep(0,n)
#y estimate for each x[i]

#store endpoints of window in vector b
for (i in 1:n){
    b=x[i]+w*c(-1,1)
    y.est[i] = mean(y[which((b[1]<=x)&(x<=b[2]))])
}

#plotting horizontal lines for each window
```

```
plot(x,y,col="orange")
for (i in 1:n){
    # b=x[i]+w*c(-1,1)
    lines(c(x[i],x[i+1]),c(y.est[i],y.est[i+1]),col="blue")
}
```



We can thus conclude that increasing the width of the window gives us a better overall fit that matches the true shape of the function (in question) in a much improved manner.