```
R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> #Script Name: dilip.k.lalwani_HW08_Script.R
> #Location: C:\Users\dilip\Google Drive\FALL 2017 CLASSES\STAT 604\HW08
> #Created by Dilip Lalwani
> #Creation Date: 10/07/17
> #Purpose: Practice working with higher level graphics
> #Last executed: 10/10/17
>
> Sys.time()
```

```
[1] "2017-10-10 00:39:56 CDT"

>

> #housekeeping

> objects()

[1] "OKHS"     "Oklahoma" "zipdata"

> ls()

[1] "OKHS"     "Oklahoma" "zipdata"

> rm(list=ls())

>

> #1 Read BBY.csv file into data frame

> bbydf<-read.csv("C:/Users/dilip/Google Drive/FALL 2017 CLASSES/STAT 604/HW08/BBY.csv")

>

> #2 Define the pdf file

> pdf("C:/Users/dilip/Google Drive/FALL 2017 CLASSES/STAT
604/HW08/dilip.k.lalwani_HW08_output.pdf", width=11, height=8.5)###############

>

> #3a Define alpha and N and assign value to alpha

> N <- 30

> alpha<-2/(1+N)

>

> #3b Create vector with zeroes to contain EMA values

> ema<-rep(0, length(bbydf$Adj.Close))

>

> #3c Calculate 30 day average and assign to the 30th member of the EMA vector

> ema[N]<-mean(bbydf$Adj.Close[1:30])

>

> #3d Use a loop to run the EMA formula repetitively from day 31 to the last day

> ca<-c((N+1):length(ema))

> for(i in ca){
```

```
+ ema[i]=(bbydf$Adj.Close[i] *alpha)+(ema[i-1]*(1-alpha))

+ }

>

> #3e Set bgcolor and plot a line showing last 260 EMA values

> par(bg = 'grey90')

> plot(1:260, ema[(length(ema)-259):length(ema)], type="l", col="blue", ylim=c(0,65), main=paste(N,
"Day EMA and Daily Stock Prices"), xlab="Days", ylab="Adjusted Closing Price")

>

> #3f Add the EMA formula

> text(0, 2, bquote(paste('EMA'[i], " = ","(",'P'[i] %*% alpha, ")", " + ", "(", 'EMA'[i-1] %*% "(",1 - alpha, "))
where ", alpha, " = ", frac(2,1+.(N)),  sep="")), col="black", adj=0)

>

> #3g Adding yellow line for actual adjusted closing prices for last 260 days

> lines(1:260,  bbydf$Adj.Close[(length(bbydf$Adj.Close)-259):length(bbydf$Adj.Close)], col="yellow")

>

> #4 Create function

> gplot<-function(vardf, N=30, ylimit=65){

+ alpha<-2/(1+N)

+ ema<-rep(0, length(vardf))

+ ema[N]<-mean(vardf[1:30])

+ ca<-c((N+1):length(ema))

+ for(i in ca){

+ ema[i]=(vardf[i] *alpha)+(ema[i-1]*(1-alpha))

+ }

+ par(bg = 'grey90')

+ plot(1:260, ema[(length(ema)-259):length(ema)], type="l", col="blue", ylim=c(0,65), main=paste(N,
"Day EMA and Daily Stock Prices"), xlab="Days", ylab="Adjusted Closing Price")

+ text(0, 2, bquote(paste('EMA'[i], " = ","(",'P'[i] %*% alpha, ")", " + ", "(", 'EMA'[i-1] %*% "(",1 - alpha, "))
where ", alpha, " = ", frac(2,1+.(N)), sep="")), col="black", adj=0)

+ lines(1:260,  vardf[(length(vardf)-259):length(vardf)], col="yellow")
+ }
>
> #5 Set margins
> par(mfcol=c(1,2))
> par(omi=c(0.5,0.5,1.5,0.5))
> par(mar=c(4,4,2,0))
>
> #6 Creating graphs by calling function
> gplot(bbydf$Adj.Close)
> gplot(bbydf$Adj.Close, 100)
>
> #7 System time at the bottom
> mtext(Sys.time(),adj = 0, side = 1, outer = TRUE)
> dev.off()
null device
          1
>
```