```
# econ424lab1.r
#
# author: E. Zivot
# created: September 24, 2009
# revision history:
# June 27, 2011
#   updated code for Summer 2011
#
#
# R functions used
#
# as.Date()                     coerce to Date object
# as.numeric()          coerce to numeric object
# class()                       return or set class of object
# colnames()            extract column names
# format()                      format output
# head()                        show fist few rows of data object
# read.csv()            read comma separated file into R
# rownames()
# seq()                 create sequence
# tail()                        show last few rows of data object
#

# set output options to show only 4 significant digits
options(digits = 4)

#
# read .csv files containing Yahoo! monthly adjusted closing price data on sbux
# from March, 1993 through March 2008. The files sbuxPrices.csv is
# assumed to be in the directory C:\Users\ezivot\Documents\classes\econ424\fall2009.
# Change to the appropriate directory where you have saved the data.
#

# read the sbux prices into a data.frame object. First look at the online help file
# for read.csv
?read.csv
# now read in the data - make sure to change the path to where the data is on your
# system
setwd("C:/Users/ezivot/Documents/classes/econ424/summer2011/")
sbux.df = read.csv(file="sbuxPrices.csv",
                   header=TRUE, stringsAsFactors=FALSE)
# sbux.df is a data.frame object. Data.frames are rectangular data objects typically with
# observations in rows and variables in columns
class(sbux.df)
str(sbux.df)
head(sbux.df)
tail(sbux.df)
colnames(sbux.df)
class(sbux.df$Date)
class(sbux.df$Adj.Close)

# notice how dates are not the end of month dates. This is Yahoo!'s fault when
# you download monthly data. Yahoo! doesn't get the dates right for the adjusted
# close data.

#
# subsetting operations
#

# extract the first 5 rows of the price data.
```

```
sbux.df[1:5, "Adj.Close"]
sbux.df[1:5, 2]
sbux.df$Adj.Close[1:5]

# in the above operations, the dimension information was lost. To preserve
# the dimension information use drop=FALSE
sbux.df[1:5, "Adj.Close", drop=FALSE]
sbux.df[1:5, 2, drop=FALSE]
sbux.df$Adj.Close[1:5, drop=FALSE]
# drop=FALSE had no effect on the last command, why?

# find indices associated with the dates 3/1/1994 and 3/1/1995
which(sbux.df$Date == "3/1/1994")
which(sbux.df == "3/1/1995")
# extract prices between 3/1/1994 and 3/1/1995
sbux.df[13:25,]

# create a new data.frame containing the price data with the dates as the row names
sbuxPrices.df = sbux.df[, "Adj.Close", drop=FALSE]
rownames(sbuxPrices.df) = sbux.df$Date
head(sbuxPrices.df)

# with Dates as rownames, you can subset directly on the dates
# find indices associated with the dates 3/1/1994 and 3/1/1995
sbuxPrices.df["3/1/1994", 1]
sbuxPrices.df["3/1/1995", 1]
# to show the rownames use drop=FALSE
sbuxPrices.df["3/1/1994", 1, drop=FALSE]

#
# plot the data
#

# note: the default plot is a "points" plot
plot(sbux.df$Adj.Close)

# let's make a better plot
# type="l" specifies a line plot
# col="blue" specifies blue line color
# lwd=2 doubles the line thickness
# ylab="Adjusted close" adds a y axis label
# main="Monthly closing price of SBUX" adds a title
plot(sbux.df$Adj.Close, type="l", col="blue",
     lwd=2, ylab="Adjusted close",
     main="Monthly closing price of SBUX")
# now add a legend
legend(x="topleft", legend="SBUX",
       lty=1, lwd=2, col="blue")

#
# compute returns
#

# simple 1-month returns
n = nrow(sbuxPrices.df)
sbux.ret = (sbuxPrices.df[2:n,1] - sbuxPrices.df[1:(n-1),1])/sbuxPrices.df[1:(n-1),1]
# notice that sbux.ret is not a data.frame object
class(sbux.ret)
# now add dates as names to the vector.
names(sbux.ret) = rownames(sbuxPrices.df)[2:n]
head(sbux.ret)
```

```
# Note: to ensure that sbux.ret is a data.frame use drop=FALSE when computing returns
sbux.ret.df = (sbuxPrices.df[2:n,1,drop=FALSE] - sbuxPrices.df[1:(n-
1),1,drop=FALSE])/sbuxPrices.df[1:(n-1),1,drop=FALSE]

# continuously compounded 1-month returns
sbux.ccret = log(1 + sbux.ret)
# alternatively
sbux.ccret = log(sbuxPrices.df[2:n,1]) - log(sbuxPrices.df[1:(n-1),1])
names(sbux.ccret) = rownames(sbuxPrices.df)[2:n]
head(sbux.ccret)

# compare the simple and cc returns
head(cbind(sbux.ret, sbux.ccret))

# plot the simple and cc returns in separate graphs
# split screen into 2 rows and 1 column
par(mfrow=c(2,1))
# plot simple returns first
plot(sbux.ret, type="l", col="blue", lwd=2, ylab="Return",
     main="Monthly Simple Returns on SBUX")
abline(h=0)
# next plot the cc returns
plot(sbux.ccret, type="l", col="blue", lwd=2, ylab="Return",
     main="Monthly Continuously Compounded Returns on SBUX")
abline(h=0)
# reset the screen to 1 row and 1 column
par(mfrow=c(1,1))

# plot the returns on the same graph
plot(sbux.ret, type="l", col="blue", lwd=2, ylab="Return",
     main="Monthly Returns on SBUX")
# add horizontal line at zero
abline(h=0)
# add the cc returns
lines(sbux.ccret, col="red", lwd=2)
# add a legend
legend(x="bottomright", legend=c("Simple", "CC"),
       lty=1, lwd=2, col=c("blue","red"))

#
# calculate growth of $1 invested in SBUX
#

# compute gross returns
sbux.gret = 1 + sbux.ret
# compute future values
sbux.fv = cumprod(sbux.gret)
plot(sbux.fv, type="l", col="blue", lwd=2, ylab="Dollars",
     main="FV of $1 invested in SBUX")
```