

```
# returnCalculations.r
#
# author: E. Zivot
# created: September 29, 2009
# revision history:
# June 24, 2011
#   Updated code to work with R 2.12.2
# October 10, 2009
#   Added examples from Performance Analytics and quantmod
# October 5, 2009
#   Added code for return calculations
#
# Data used (available at http://faculty.washington.edu/ezivot/econ424/R\_hints.htm)
# sbuxPrices.csv
# msftPrices.csv
#
# Core R functions used
#
# abline
# class
# colnames
# exp
# legend
# lines
# log
# max
# min
# plot
# read.csv
# rownames
# seq
# str
# tail
#
# R packages/functions used
#
# PerformanceAnalytics
#   chart.TimeSeries
#   chart.Cumulative
#   ReturnCalculate
#
# quantmod
#   getSymbols
#
# zoo
#   zoo
#   as.yearmon
#   diff
#   lag
#
# set options
#
options(digits = 4)
options(width = 75)
options(chmhelp=TRUE)
loadPath = "C:\\\\Users\\ezivot\\Documents\\FinBook\\Data\\"
#
# load 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
sbux.df = read.csv(file=paste(loadPath, "sbuxPrices.csv", sep=""),
                    header=TRUE, stringsAsFactors=FALSE)
msft.df = read.csv(file=paste(loadPath, "msftPrices.csv", sep=""),
                    header=TRUE, stringsAsFactors=FALSE)
# sbux.df and msft are a data.frame objects. Data.frames are rectangular data
# objects 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)

# plot prices
plot(msft.df$Adj.Close, type = "l", lty = "solid", lwd = 2,
      col = "blue", ylab = "return")
lines(sbux.df$Adj.Close, lty = "dotted", lwd = 2, col = "black")
legend(x = "topleft", legend=c("MSFT", "SBUX"), lwd = 2,
       lty = c("solid", "dotted"), col = c("blue", "black"))

#
# compute monthly simple returns
#
n = nrow(sbux.df)
sbux.ret = sbux.df$Adj.Close[2:n]/sbux.df$Adj.Close[1:n-1] - 1
msft.ret = msft.df$Adj.Close[2:n]/msft.df$Adj.Close[1:n-1] - 1
head(cbind(sbux.ret, msft.ret))

# compute 2-period returns
sbux.ret.2 = sbux.df$Adj.Close[3:n]/sbux.df$Adj.Close[1:n-2] - 1

#
# compute monthly cc returns
#
sbux.ccret = log(1 + sbux.ret)
msft.ccret = log(1 + msft.ret)

# equivalently can compute
sbux.ccret = log(sbux.df$Adj.Close[2:n]/sbux.df$Adj.Close[1:n-1])
msft.ccret = log(msft.df$Adj.Close[2:n]/msft.df$Adj.Close[1:n-1])
head(cbind(sbux.ccret, msft.ccret))

# rearrange data frame so that the rownames are the character dates and the
# data.frame only contains the price data. Note, using drop=FALSE preserves
# the data.frame object structure

rownames(sbux.df) = sbux.df$Date
rownames(msft.df) = msft.df$Date
sbux.df = sbux.df[, "Adj.Close", drop=FALSE]
msft.df = msft.df[, "Adj.Close", drop=FALSE]

#
# Return calculations using zoo objects and PerformanceAnalytics functions
# note: PerformanceAnalytics automatically loads zoo

```

```

#
library(PerformanceAnalytics)

# create zoo objects from data.frame objects
dates.sbx = as.yearmon(sbx.df$Date, format="%m/%d/%Y")
dates.msft = as.yearmon(msft.df$Date, format="%m/%d/%Y")
sbx.z = zoo(x=sbx.df$Adj.Close, order.by=dates.sbx)
msft.z = zoo(x=msft.df$Adj.Close, order.by=dates.msft)
class(sbx.z)
head(sbx.z)

# subset using dates
# subsetting can be done with a date index and the window function
sbx.z[as.yearmon(c("Mar 1993", "Mar 1994"))]
window(sbx.z, start=as.yearmon("Mar 1993"), end=as.yearmon("Mar 1994"))

# create merged time series
sbxMsft.z = merge(sbx.z, msft.z)
head(sbxMsft.z)

# two series on same graph
plot(msft.z, lwd=2, col="blue", ylab="Prices", xlab="Months")
lines(sbx.z, col="black", lwd=2, lty="dotted")
legend(x="topleft", legend=c("MSFT", "SBUX"), col=c("blue", "black"),
      lwd=2, lty=c("solid", "dotted"))
# two series in two separate panels
plot(sbxMsft.z, lwd=c(2,2), plot.type="multiple",
     col=c("black", "blue"), lty=c("solid", "dotted"),
     ylab=c("SBUX", "MSFT"), main="")

# create merged time series
sbxMsft.z = merge(sbx.z, msft.z)

# compute returns using diff() and lag()

sbxRet.z = diff(sbx.z)/lag(sbx.z, k=-1)
sbxRetcc.z = diff(log(sbx.z))
head(merge(sbxRet.z, sbxRetcc.z))

sbxMsftRet.z = CalculateReturns(sbxMsft.z, method="simple")
head(sbxMsftRet.z)
# continuously compounded returns
sbxMsftRetcc.z = CalculateReturns(sbxMsft.z, method="compound")
head(sbxMsftRetcc.z)

# show prices with dot-com boom shading and labels
# dates are formatted the same way they appear on the x-axis
#shading.dates = list(c("01/98", "10/00"))
#label.dates = c("01/98", "10/00")
shading.dates = list(c("Jan 98", "Oct 00"))
label.dates = c("Jan 98", "Oct 00")

label.values = c("Start of Boom", "End of Boom")
chart.TimeSeries(msft.z, lwd=2, col="blue", ylab="Price",
                main="The rise and fall of Microsoft stock",
                period.areas=shading.dates, period.color="yellow",
                event.lines=label.dates, event.labels=label.values,
                event.color="black")

# plot growth of $1 invested
chart.CumReturns(sbxMsftRet.z, lwd=2, main="Growth of $1", legend.loc="topleft")

```

```
#
# Return calculations and charting using quantmod and xts
#
library(quantmod)
getSymbols("YH00")
class(YH00)
colnames(YH00)
start(YH00)
end(YH00)
head(YH00)
chartSeries(YH00,theme=chartTheme('white'))

#
# Return calculations and plotting using Rmetrics package fPortfolio
#

# to be completed

#
# properties of exponentials and logarithms
#

x = seq(-1, 2, length.out = 100)
e.x = exp(x)
ln.x = log(x[x > 0])
lb = min(e.x, ln.x)
ub = max(e.x, ln.x)
plot(x, e.x, ylim = c(lb, ub), type = "l", lty = 2, lwd = 2, col = "blue",
      xlab = "x", ylab = "y")
lines(x[x > 0], ln.x, lwd = 2, col = "red")
abline(h=0, v=0)
legend(x = "topleft", legend = c("exp(x)", "ln(x)"), lty=c(2, 1), lwd = 2,
      col = c("blue", "red"))
```