```
# singleIndex.r                 script file for single Index model regression analysis
# author: Eric Zivot
# created: November 12, 2003
# updated: December 1, 2009
#
# comments
# data are imported from singleIndexPrices.csv
# which contains monthly closing prices on the following assets
#
# sp500, sbux, msft, nord, boeing
#
# over the period 10/98 - 10/03
#

#
# 1. Load data
#
options(digits=4)
library(zoo)
library(car)
loadPath = "C:\\Users\\ezivot.SOCIOLOGY\\Dropbox\\FinBook\\EXCEL\\"
singleIndexPrices.df = read.csv(file=paste(loadPath, "singleIndexPrices.csv", sep=""),
                                stringsAsFactors=F)
colnames(singleIndexPrices.df)
#
# 2. Create zooreg object from data and dates in singleIndexPrices.df
#
td = seq(as.Date("1998-01-01"), as.Date("2003-01-01"), by="months")
singleIndexPrices.z = zoo(singleIndexPrices.df[,-1], td)

my.panel <- function(...) {
  lines(...)
  abline(h=0)
}
plot(singleIndexPrices.z, lwd=2, col="blue")
#
# 3. create returns
#
si.z = diff(log(singleIndexPrices.z))
si.df = as.data.frame(si.z)
head(si.df)

# returns excluding market
ret.mat = as.matrix(si.df[,-1])

# plot returns over full sample
plot(si.z, panel=my.panel, lwd=2, col="blue")

#
# compute alphas and betas using descriptive statistics
#

ret.mat = as.matrix(si.df)
muhat.vals = colMeans(si.df)
cov.mat = var(ret.mat)
beta.vals = cov.mat[,1]/cov.mat[1,1]
alpha.vals = muhat.vals[-1] - beta.vals[-1]*muhat.vals["sp500"]
beta.vals
alpha.vals
```

```
#
# plot asset returns and market returns
#

# distribution of market return
par(mfrow=c(2,2))
        hist(si.df$sp500,main="S&P 500 monthly cc returns",
        probability=T, ylab="cc return",col="slateblue1")
        boxplot(si.df$sp500,outchar=T, ylab="cc return",col="slateblue1")
        plot(density(si.df$sp500),type="l",xlab="cc return",
        ylab="density estimate", main="smoothed density",col="slateblue1")
        qqnorm(si.df$sp500,col="slateblue1")
        qqline(si.df$sp500)
par(mfrow=c(1,1))


# sp500 vs sbux
plot(si.z[,c("sp500","sbux")], plot.type="single",
main="Monthly cc returns on S&P 500 and Starbucks",
ylab="returns", col=c("blue","orange"), lwd=c(2,2))
abline(h=0)
legend(x="bottomright",legend=c("S&P 500","SBUX"),
       lwd=c(2,2),col=c("blue","orange"))

plot(si.df$sp500,si.df$sbux, col="slateblue1",lwd=2, pch=16, cex=2,
main="Monthly cc returns on S&P 500 and Starbucks",
xlab="cc return on S&P 500", ylab="cc return on SBUX")
abline(h=0,v=0)
abline(a=alpha.vals["sbux"],b=beta.vals["sbux"],
       col="orange", lwd=3)


# sp500 vs msft
plot(si.z[,c("sp500","msft")], plot.type="single",
main="Monthly cc returns on S&P 500 and Microsoft",
ylab="returns", col=c("blue","orange"), lwd=c(2,2))
abline(h=0)
legend(x="bottomright",legend=c("S&P 500","MSFT"),
       lwd=c(2,2),col=c("blue","orange"))

plot(si.df$sp500,si.df$msft, col="slateblue1",lwd=2, pch=16, cex=2,
     main="Monthly cc returns on S&P 500 and Microsoft",
     xlab="cc return on S&P 500", ylab="cc return on msft")
abline(h=0,v=0)
abline(a=alpha.vals["msft"],b=beta.vals["msft"],
       col="orange", lwd=3)

# sp500 vs nord
plot(si.z[,c("sp500","nord")], plot.type="single",
main="Monthly cc returns on S&P 500 and Nordstrom",
ylab="returns", col=c("blue","orange"), lwd=c(2,2))
abline(h=0)
legend(x="bottomright",legend=c("S&P 500","NORD"),
       lwd=c(2,2),col=c("blue","orange"))

plot(si.df$sp500,si.df$nord, col="slateblue1",lwd=2, pch=16, cex=2,
     main="Monthly cc returns on S&P 500 and Nordstrom",
     xlab="cc return on S&P 500", ylab="cc return on nord")
abline(h=0,v=0)
abline(a=alpha.vals["nord"],b=beta.vals["nord"],
       col="orange", lwd=3)
```

```
# sp500 vs. boeing
plot(si.z[,c("sp500","boeing")], plot.type="single",
main="Monthly cc returns on S&P 500 and Boeing",
ylab="returns", col=c("blue","orange"), lwd=c(2,2))
abline(h=0)
legend(x="bottomright",legend=c("S&P 500","BOEING"),
       lwd=c(2,2),col=c("blue","orange"))

plot(si.df$sp500,si.df$boeing, col="slateblue1",lwd=2,pch=16,cex=2,
     main="Monthly cc returns on S&P 500 and Boeing",
     xlab="cc return on S&P 500", ylab="cc return on boeing")
abline(h=0,v=0)
abline(a=alpha.vals["boeing"],b=beta.vals["boeing"],
       col="orange", lwd=3)

#
# 6. create scatterplot with arbitrary regression line
#
plot(si.df$sp500,si.df$msft,col="slateblue1",pch=16, cex=2,
     ylab="msft",xlab="sp500")
# plot line with intercept=0, slope=1
abline(a=0,b=1, lwd=3)
abline(h=0,v=0)

# illustrate least squares as best fitting line
plot(si.df$sp500,si.df$msft,col="slateblue1",lwd=2, pch=16, cex=2,
     main="",
     ylab="msft",xlab="sp500")
# plot line with intercept=0, slope=1
abline(a=0,b=1, col="black", lwd=3)
abline(h=0,v=0)
# plot best fitting regression line
abline(a=alpha.vals["msft"], b=beta.vals["msft"],
       col="orange", lwd=3)

#
# 7. estimate single Index model for Microsoft using lm function
#
colnames(si.df)

?lm
args(lm)
msft.fit = lm(msft~sp500,data=si.df)
class(msft.fit)
names(msft.fit)
msft.fit$coef

# print method
msft.fit
# summary method - gives SE values, t-stats etc
summary(msft.fit)
# extractor functions
coef(msft.fit)
residuals(msft.fit)[1:5]
fitted(msft.fit)[1:5]

# store information from summary
msft.summary = summary(msft.fit)
names(msft.summary)
msft.summary$coef
```

```
msft.summary$r.squared

# plot scatterplot with regression line
plot(si.df$sp500,si.df$msft,col="slateblue1", pch=16, cex=2,
     main="Estimated SI model for Microsoft",
     ylab="msft",xlab="sp500")
# plot line with intercept=least squares intercept,
# slope=least squares slope
abline(h=0,v=0)
abline(msft.fit, col="orange", lwd=3)

# compute 95% confidence intervals
# use R function confint


# extractor functions

# plot residuals and fitted values
        layout(matrix(c(1,1,2,2), 2, 2, byrow=T))
      layout.show(2)

        ts.plot(si.df$msft, main="actual and fitted",
               col="blue", lwd=2, ylab="returns")
      abline(h=0)
      lines(fitted(msft.fit), col="orange", lwd=3)
        legend(x="bottomleft",legend=c("fitted","actual"),
               lwd=c(2,2),col=c("orange","blue"))
        ts.plot(residuals(msft.fit),main="residuals",
              ylab="returns", lwd=3, col="green")
        abline(h=0)

#
# diagnostic plots for residuals
#
ehat = residuals(msft.fit)
par(mfrow=c(2,2))
        hist(ehat, xlab="residuals",ylab="frequency",
        main="Residuals from SI model for MSFT",
      col="slateblue1")
        boxplot(ehat,outchar=T, col="slateblue1")
        plot(density(ehat),type="l",xlab="residual",
      ylab="density", col="slateblue1", main="smoothed density")
        qqnorm(ehat, col="slateblue1")
        qqline(ehat)
par(mfrow=c(1,1))

# autocorrelation plots

par(mfrow=c(2,1))
  plot(zoo(ehat, as.Date(names(ehat))),main="MSFT residuals", ylab="ehat")
  abline(h=0)
  acf(ehat, main="ACF of MSFT residuals", lwd=3)
par(mfrow=c(1,1))

# fitted models for remaining assets
sbux.fit = lm(sbux~sp500,data=si.df)
nord.fit = lm(nord~sp500,data=si.df)
boeing.fit = lm(boeing~sp500,data=si.df)

#
# estimate SI model for a portfolio
```

```
#

# create equally weighted portfolio
# and add to data frame si.df
port = (si.df$msft + si.df$sbux + si.df$nord + si.df$boeing)/4
si.df$port = port
colnames(si.df)

# estimate SI model for equally weighted portfolio
port.fit = lm(port~sp500,data=si.df)

# show regression output
summary(port.fit)

# create scatterplot with regression line
plot(si.df$sp500,si.df$port,main="SI model for 4 Asset Portfolio",
     col="slateblue", pch=16, cex=2)    # create scatterplot
abline(port.fit, col="orange", lwd=3)
# add regression line
abline(h=0,v=0)


#
# 7. compute covariance matrix implied by single index model
#

# Ex: use coef extractor function to get regression coefficients
coef(sbux.fit)

sbux.beta = coef(sbux.fit)[2]
msft.beta = coef(msft.fit)[2]
nord.beta = coef(nord.fit)[2]
boeing.beta = coef(boeing.fit)[2]

beta.vec = c(sbux.beta,msft.beta,nord.beta,boeing.beta)
names(beta.vec) = c("SBUX","MSFT","NORD","BOEING")
beta.vec

# compute variance of market

sig2.sp500 = var(si.df$sp500)
sig2.sp500

# compute var(Rm)*beta*beta'

cov.market = sig2.sp500*(beta.vec%*%t(beta.vec))
cov.market

# compute residual variances
# note: use residuals extractor function to get OLS residuals
residuals(sbux.fit)

sig2e.sbux = var(residuals(sbux.fit))
sig2e.msft = var(residuals(msft.fit))
sig2e.nord = var(residuals(nord.fit))
sig2e.boeing = var(residuals(boeing.fit))

D.mat = diag(c(sig2e.sbux,sig2e.msft,sig2e.nord,sig2e.boeing))
D.mat

# compute si covariance matrix
```

```
cov.si = cov.market + D.mat
cov.si

# compare with sample covariance matrix
cov.hat = var(si.df[,2:5])
cov.si
cov.hat
cov.si - cov.hat

# compute si correlation matrix

cor.si = cov2cor(cov.si)

# compare with sample correlation matrix
cor.hat = cor(si.df[,2:5])
dimnames(cor.si) = dimnames(cor.hat)

cor.si
cor.hat
cor.si - cor.hat


# compute rolling betas - user PerformanceAnalytics functions
chart.RollingRegression(si.z[,"msft",drop=F], si.z[,"sp500",drop=F], width=24)
abline(h=beta.vals["msft"], lwd=2, col="red")
chart.RollingRegression(si.z[,"sbux",drop=F], si.z[,"sp500",drop=F], width=24)
abline(h=beta.vals["sbux"], lwd=2, col="red")
chart.RollingRegression(si.z[,"nord",drop=F], si.z[,"sp500",drop=F], width=24)
abline(h=beta.vals["nord"], lwd=2, col="red")
chart.RollingRegression(si.z[,"boeing",drop=F], si.z[,"sp500",drop=F], width=24)
abline(h=beta.vals["boeing"], lwd=2, col="red")
```