Project Deliverable 2

Rees Klintworth

October 30, 2014

All components of the Class Project, Deliverable 2, are present in this document. Little emphasis was placed on formatting, since this will all be changed once the project is converted to Shiny.

1. ERP

# Load necessary packages  
suppressMessages(library(Quandl))  
suppressMessages(library(lubridate))  
suppressMessages(library(lattice))  
suppressMessages(library(reshape2))  
Quandl.auth("Xwpyys22sxHPzyXBrGdH")  
## Part 6 - ERP  
# Date variables to make modification easier when moved to Shiny  
years\_erp <- 10  
start\_date\_erp <- as.Date(ymd(Sys.Date()) - years(years\_erp))  
start\_date\_beta <- as.Date(ymd(Sys.Date()) - years(3))  
end\_date <- Sys.Date()  
# US stock market (S&P 500) values  
gspc\_erp <- Quandl("YAHOO/INDEX\_GSPC", start\_date=start\_date\_erp, end\_date=end\_date)  
# Risk-free rate of return values  
tnx <- Quandl("YAHOO/INDEX\_TNX", start\_date=start\_date\_erp, end\_date=end\_date)  
# Reverse the data to get the normal form  
gspc\_erp <- gspc\_erp[nrow(gspc\_erp):1,]  
tnx <- tnx[nrow(tnx):1,]  
# Calculate the rates of return for gspc  
gspc\_rates <- diff(gspc\_erp$'Adjusted Close')/gspc\_erp$'Adjusted Close'[-length(gspc\_erp$'Adjusted Close')]  
# Find the Average rate for each, and then annualize it  
gspc\_rate <- mean(gspc\_rates)  
gspc\_rate <- (1 + gspc\_rate)^(length(gspc\_rates)/years\_erp) - 1  
# TNX data is already annualized, just need to find the average  
tnx\_rate <- mean(tnx$'Adjusted Close'/100)  
erp <- gspc\_rate - tnx\_rate  
cat("The calculated ERP is:", sprintf("%1.2f%%", 100\*erp))

The calculated ERP is: 4.65%

7/8. Beta + Expected Annualized Rate of Return for each stock

## Part 7/8 - BETA and Expected Rate of Return  
# Three stocks I've selected for this part  
my\_stocks <- c("BA", "CAG", "SCHW")  
gspc\_beta <- Quandl("YAHOO/INDEX\_GSPC", start\_date=start\_date\_beta, end\_date=end\_date)  
gspc\_beta <- gspc\_beta[nrow(gspc\_beta):1,]  
# Returns a matrix of the rates of returns for all of the stocks  
stock\_rates <- sapply(my\_stocks, function(symbol)  
 {  
 cat("Analyzing", symbol, "\n")  
 quandl\_code = paste0("YAHOO/", symbol)  
 stock <- Quandl(quandl\_code, start\_date=start\_date\_beta, end\_date=end\_date)  
 # Reverse the data so it isn't backwards  
 stock <- stock[nrow(stock):1,]  
 # Calculate the rates of return for the symbol  
 stock\_rates <- diff(stock$'Adjusted Close')/stock$'Adjusted Close'[-length(stock$'Adjusted Close')]  
 # Calculate the rates of return for the S&P 500  
 gspc\_beta\_rates <- diff(gspc\_beta$'Adjusted Close')/gspc\_beta$'Adjusted Close'[-length(gspc\_beta$'Adjusted Close')]  
 # Create a linear model to find the line of best fit  
 res=lm(stock\_rates~gspc\_beta\_rates)  
 # Beta is the slope of the Linear model  
 beta = res$coefficients[[2]]  
 cat("Beta:", beta, "\n")  
 # Find the expected return using the formula from the CAPM model  
 expected\_return = tnx\_rate + beta\*erp  
 cat("Expected Annualized Rate of Return:",sprintf("%1.2f%%", 100\*expected\_return), "\n")  
 stock\_rates  
 })

## Analyzing BA   
## Beta: 1.031   
## Expected Annualized Rate of Return: 8.15%   
## Analyzing CAG   
## Beta: 0.5632   
## Expected Annualized Rate of Return: 5.97%   
## Analyzing SCHW   
## Beta: 1.592   
## Expected Annualized Rate of Return: 10.75%

# Add column names to the matrix  
colnames(stock\_rates) = my\_stocks

9/10: Covariance and Correlation

## Part 9/10 - Covariance and Correlation  
cov\_matrix <- cov(stock\_rates)  
cor\_matrix <- cor(stock\_rates)  
cat("Covariance Matrix:")

Covariance Matrix:

print(cov\_matrix)

BA CAG SCHW  
BA 0.0001700 4.100e-05 1.088e-04  
CAG 0.0000410 1.125e-04 5.649e-05  
SCHW 0.0001088 5.649e-05 3.109e-04

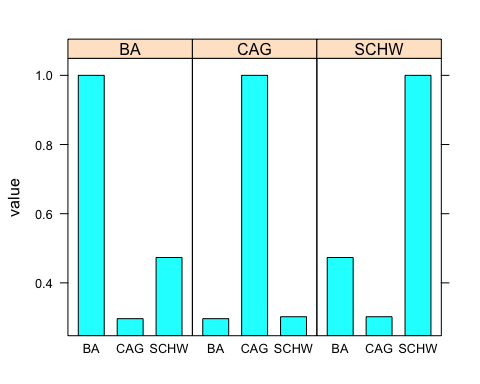
cat("Correlation Matrix")

Correlation Matrix

print(cor\_matrix)

BA CAG SCHW  
BA 1.0000 0.2964 0.4733  
CAG 0.2964 1.0000 0.3020  
SCHW 0.4733 0.3020 1.0000

# Need to adjust the data format to be able to plot it  
print\_cor <- melt(cor\_matrix)  
  
# Plot the correlation on a barchart using lattice  
barchart(value ~ Var1 | Var2, data = print\_cor, layout=c(3,1))

 11: Input weights

## Part 11 - Input Weights  
# Naively weight stocks now, but users will be able to select weights in shiny app  
percent\_per\_stock <- 100/length(my\_stocks)  
invisible(sapply(my\_stocks, function(symbol)  
 {  
 cat(symbol,sprintf("%1.2f%%", percent\_per\_stock), "\n")  
 }))

BA 33.33%   
CAG 33.33%   
SCHW 33.33%