Module 7.4: Static Risk Management Common Stocks
R Commentary

See module *Ch 7.4 SRM Stocks*.

We comment here on snippets of R code.
SRM Two Underlying Instruments Test.R (Selected excerpts and output)
Within this code, we provide computations for two financial instruments independently and then appraise them together. We contrast Wendy's and McDonalds as illustration.

```
#
# Import large data files so we trim with start and end date
#
StartDate = 20150101
EndDate = 20200630
RollingWindow = 130 # 1/2 year (260 trading days per year)
# Instrument #1
FileName <- "WEN.PRN"
mTitle1 <- paste0("Wendy's")
source('Instrument 1.R')
dfResults1 = dfStockRisk1
# Instrument #2 -- can reference different files, but same format
FileName <- "MCD.PRN"
mTitle2 <- paste0("McDonalds")
source('Instrument 2.R') # Use with same File, but different contracts
dfResults2 = dfStockRisk2
mTitle = paste0(mTitle1, " and ", mTitle2)
```

In the file Instrument 1.R, we use the zoo package extensively as it handles time series data better.

```
# Instrument 1.R
FP <- read.csv(FileName)
FP <- FP[FP$Date >= StartDate,]
FP <- FP[FP$Date <= EndDate,]
FP$Year <- floor(FP$Date/10000)
FP$Month <- floor(FP$Date/100) - FP$Year*100
FP$Day <- FP$Date - FP$Year*10000 - FP$Month*100
Month <- zoo(FP$Month)
Day <- zoo(FP$Day)
Year <- zoo(FP$Year)
Price <- FP$Close
rm(FP, FileName)
# Month, Day, Year, Price, Return
CDate = mdy.date(Month, Day, Year)
ZDate <- zoo(CDate)
ZPrice <- zoo(Price)
LagPrice <- lag(ZPrice, -1, na.pad = TRUE)
CCReturn <- log(ZPrice) - log(LagPrice) # Continuously compounded rate of Return
DCReturn <- (ZPrice - LagPrice)/LagPrice # Discretely compounded rate of Return
ACPrice <- ZPrice - LagPrice # Actual change in Price (ABM measure)
```

The method rollapply is a useful function for computing rolling statistics. But first we create a dataframe to hold the entire set of information.

```
# Instrument 1.R
FP <- read.csv(FileName)
FP <- FP[FP$Date >= StartDate,]
FP <- FP[FP$Date <= EndDate,]
FP$Year <- floor(FP$Date/10000)
FP$Month <- floor(FP$Date/100) - FP$Year*100
FP$Day <- FP$Date - FP$Year*10000 - FP$Month*100
Month <- zoo(FP$Month)
Day <- zoo(FP$Day)
Year <- zoo(FP$Year)
Price <- FP$Close
rm(FP, FileName)
# Month, Day, Year, Price, Return
```

```
CDate = mdy.date(Month, Day, Year)
ZDate <- zoo(CDate)
ZPrice <- zoo(Price)
LagPrice <- lag(ZPrice, -1, na.pad = TRUE)
CCReturn <- log(ZPrice) - log(LagPrice) # Continuously compounded rate of Return
DCReturn <- (ZPrice - LagPrice)/LagPrice # Discretely compounded rate of Return
ACPrice <- ZPrice - LagPrice # Actual change in Price (ABM measure)
LengthPrice = length(ZPrice)
dfStockRisk1 = data.frame(matrix(vector(), LengthPrice, 22, dimnames=list(c(),
  c("Date", "Month", "Day", "Year", "Price", "CCReturn", "DCReturn",
    "ACPrice", "CCStDev", "DCStDev", "ACStDev", "ACCCStDev", "ACDCStDev",
    "CCMean", "DCMean", "ACMean","CCSkewness", "DCSkewness", "ACSkewness",
    "CCKurtosis", "DCKurtosis", "ACKurtosis"
    ))), stringsAsFactors=F)
dfStockRisk1$Date <- ZDate
dfStockRisk1$Month <- Month
dfStockRisk1$Day <- Day
dfStockRisk1$Year <- Year
dfStockRisk1$Price <- ZPrice
dfStockRisk1$CCReturn <- CCReturn
dfStockRisk1$DCReturn <- DCReturn
dfStockRisk1$ACPrice <- ACPrice
dfStockRisk1 <- dfStockRisk1[-1,]      # Remove first line (NAs)
CCMean <- rollapply(dfStockRisk1$CCReturn, RollingWindow,
  function(x)(mean(x)), align = "right")
CCStDev <- rollapply(dfStockRisk1$CCReturn, RollingWindow,
  function(x)(sqrt(var(x))), align = "right")
CCSkewness <- rollapply(dfStockRisk1$CCReturn, RollingWindow,
  function(x)(skewness(x)), align = "right")
CCKurtosis <- rollapply(dfStockRisk1$CCReturn, RollingWindow,
  function(x)(kurtosis(x)), align = "right")
DCMean <- rollapply(dfStockRisk1$DCReturn, RollingWindow,
  function(x)(mean(x)), align = "right")
DCStDev <- rollapply(dfStockRisk1$DCReturn, RollingWindow,
  function(x)(sqrt(var(x))), align = "right")
DCSkewness <- rollapply(dfStockRisk1$DCReturn, RollingWindow,
  function(x)(skewness(x)), align = "right")
DCKurtosis <- rollapply(dfStockRisk1$DCReturn, RollingWindow,
  function(x)(kurtosis(x)), align = "right")
ACMean <- rollapply(dfStockRisk1$ACPrice, RollingWindow,
  function(x)(mean(x)), align = "right")
ACStDev <- rollapply(dfStockRisk1$ACPrice, RollingWindow,
  function(x)(sqrt(var(x))), align = "right")
ACSkewness <- rollapply(dfStockRisk1$ACPrice, RollingWindow,
  function(x)(skewness(x)), align = "right")
ACKurtosis <- rollapply(dfStockRisk1$ACPrice, RollingWindow,
  function(x)(kurtosis(x)), align = "right")
```

The final part of this program explores statistical differences between Wendy's and McDonalds.