COMPUTATIONAL FINANCE & RISK MANAGEMENT

UNIVERSITY *of* WASHINGTON

Department of Applied Mathematics

# Calculating Portfolio Returns

CFRM 425 (009)

R Programming for Quantitative Finance

# References/Reading/Topics

- These slides
- Ang Ch 3, § §3.2-3.3.2 (loosely)

- Topics:
  - Preliminaries
    - The apply(.) function
    - The components of an xts object
  - Calculating portfolio returns

# Preliminaries

# The `apply(.)` function

- The **`apply(.)`** function is a more general method for applying vectorized operations
- Unlike sapply(.), it is not limited to a dataframe argument
  - matrix
  - n-dimensional array
  - xts object
- **`apply(X = data, MARGIN = {1, 2}, FCN = function)`**
  - MARGIN = 1 => apply by row
  - MARGIN = 2 => apply by column
  - MARGIN > 2 => greater dimensions of an n-dimensional array

# The apply(.) function

```
# Simple example on daily prices:
dailyOpen <- merge(Op(AMZN), Op(SPY), Op(IBM))
res <- apply(dailyOpen, MARGIN = 1, FUN = sum)

# Can convert to xts (see sample code)
res.xts <- as.xts(res)
```

# Components of an xts object

- Index:
  - the date/time column
  - **`index(xts_object)`** **`# Returns a date type`**
- Core Data:
  - data in the columns to the left of the index
  - **`coredata(xts_object)`** **`# Returns a matrix type`**

- See sample code for more details

# Calculating Portfolio Returns

# Calculating portfolio returns

- A portfolio return is calculated by taking the dot product of the portfolio weights and the returns for each asset
- The mean return is estimated by taking the dot product of the weights (assuming they are held constant) and the mean returns for each asset
- The portfolio return at each time step is often used in fund tracking
- The mean portfolio return is often used for computing the weights that will minimize the portfolio risk (we will cover this soon)

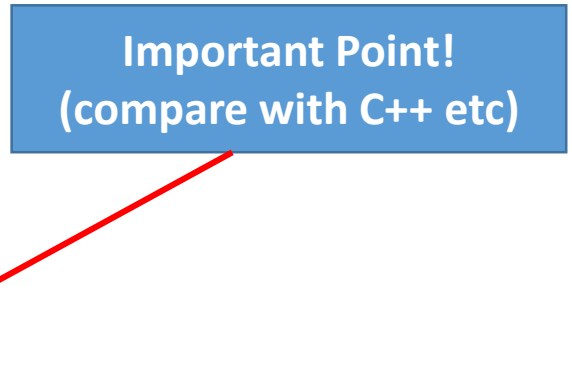
# Calculating portfolio returns

- Plotting of prices and returns, with some additional features to improve the presentations of the results:

```
wts <- c(0.2, 0.6, 0.2)

# Then,
dotWts <- function(v)
{
  v %*% wts      # wts set outside function (captured by R)
}

# Again, use the apply function; MARGIN = 1 => by row.
# Not limited to dataframe; works for matrix here:
portRtn <- apply(rtns, MARGIN = 1, FUN = dotWts)
class(portRtn)  # numeric, not xts (in fact, it's a vector)
```



# Calculating portfolio returns

- We can then coerce the result back to an xts object, and then merge with the market returns; however, this comes with a caveat:

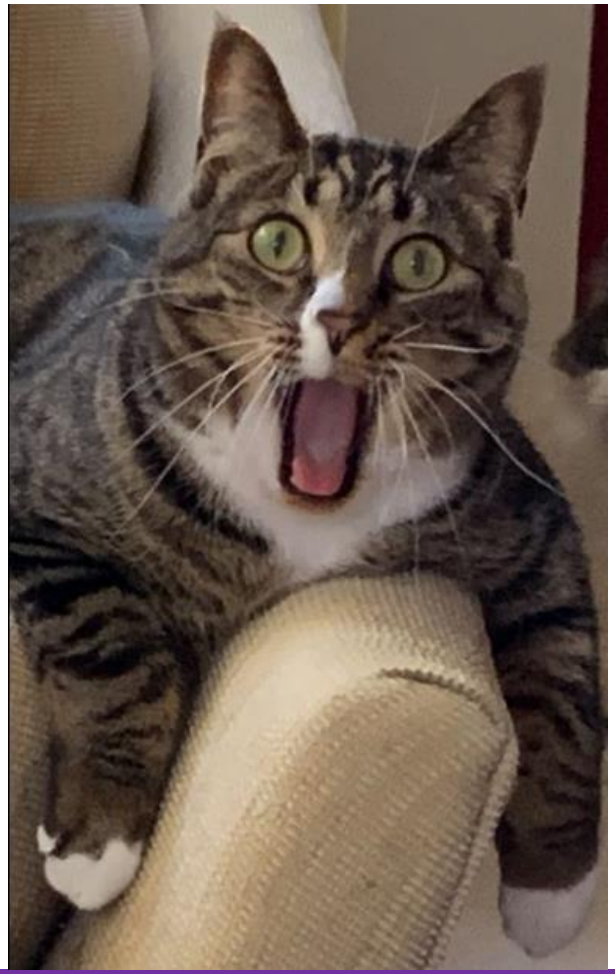
```
portRtn.xts <- as.xts(portRtn)
index(portRtn.xts[1,]) == index(rtns[1,]) # not comparable - why?
# Reason is that the date formats are different:
class(index(portRtn.xts[1,]))   # "POSIXct" "POSIXt" (carries TZ)
class(index(rtns[1,]))          # "Date"

# Try again, but change the date format:
portRtn.xts <- as.xts(portRtn, dateFormat = "Date")
class(index(portRtn.xts[1,]))   # "Date"
index(portRtn.xts[1,]) == index(rtns[1,])   # TRUE!

# We can now merge the two:
allRtns <- merge(rtns, portRtn.xts)
```

# Calculating portfolio returns

```
# Calculate Mean Monthly Portfolo Return
meanRtns <- apply(rtns, MARGIN = 2, mean)
(meanPortRtn <- meanRtns %*% wts)
```




[END]