

Risk Return Trade-off

```
ff.data <- data.table::fread(file.path(data.dir, "F-F_Research_Data_Factors.CSV"))
```

```
Warning in data.table::fread(file.path(data.dir, "F-
F_Research_Data_Factors.CSV")): Stopped early on line 1128. Expected 5 fields
but found 0. Consider fill=TRUE and comment.char=. First discarded non-empty
line: <<Annual Factors: January-December >>
```

```
ff.data$Date <- as.yearmon(parse_date(as.character(ff.data$V1), format = "%Y%m"))
```

```
ff.data$V1 <- NULL
```

```
factor.data <- ff.data[, .(Date, RmxRF = `Mkt-RF` / 100, SMB, HML, Rf = RF / 100)]
```

```
FF <- factor.data[Date >= "1963-12-01" &
                  Date <= "2013-12-31"][, .(Date, RmxRF, Rf)]
```

```
FF[, Rm := RmxRF + Rf]
```

```
FF$Gross.Rm <- 1 + FF$Rm
```

```
FF$Gross.Rm[1] <- 1
```

```
FF$Gross.Rf <- 1 + FF$Rf
```

```
FF$Gross.Rf[1] <- 1
```

```
FF
```

	Date	RmxRF	Rf	Rm	Gross.Rm	Gross.Rf
1:	Dec 1963	0.0183	0.0029	0.0212	1.0000	1.0000
2:	Jan 1964	0.0224	0.0030	0.0254	1.0254	1.0030
3:	Feb 1964	0.0154	0.0026	0.0180	1.0180	1.0026
4:	Mar 1964	0.0141	0.0031	0.0172	1.0172	1.0031
5:	Apr 1964	0.0010	0.0029	0.0039	1.0039	1.0029

597:	Aug 2013	-0.0271	0.0000	-0.0271	0.9729	1.0000
598:	Sep 2013	0.0377	0.0000	0.0377	1.0377	1.0000
599:	Oct 2013	0.0418	0.0000	0.0418	1.0418	1.0000
600:	Nov 2013	0.0312	0.0000	0.0312	1.0312	1.0000
601:	Dec 2013	0.0281	0.0000	0.0281	1.0281	1.0000

```
FF[, Cum.Rm := cumprod(FF$Gross.Rm)]
```

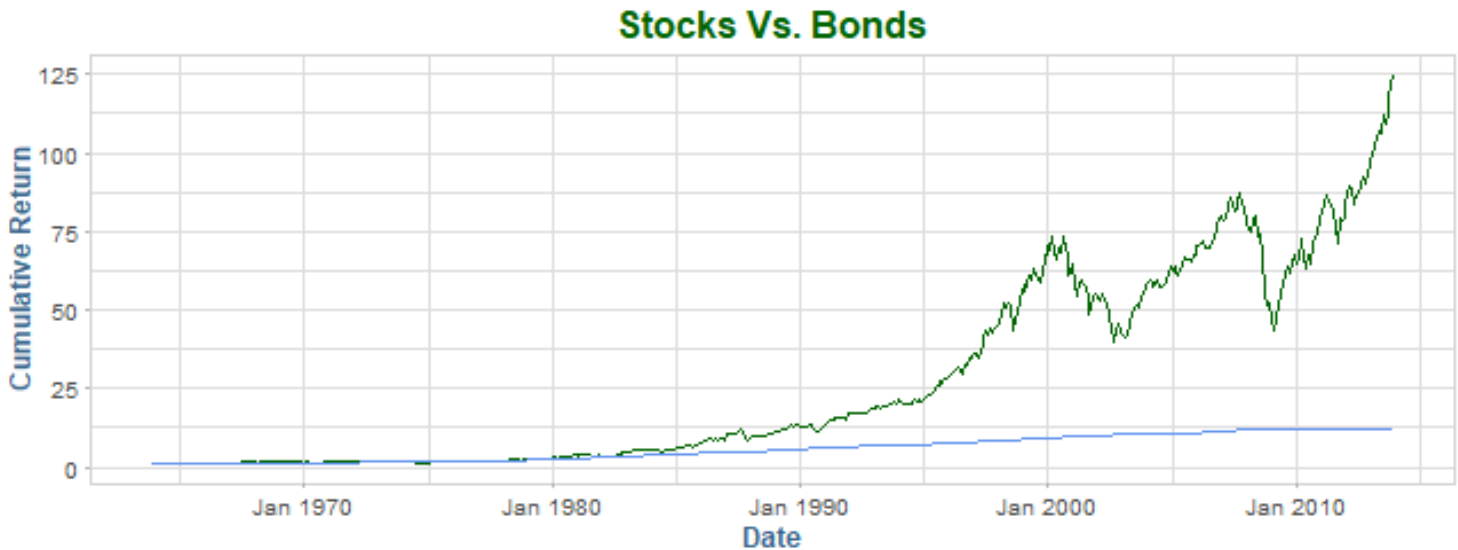
```
FF[, Cum.Rf := cumprod(FF$Gross.Rf)]
```

```
y.range <- range(FF$Cum.Rm, FF$Cum.Rf)
```

```
y.range
```

```
[1] 1.0000 124.7471
```

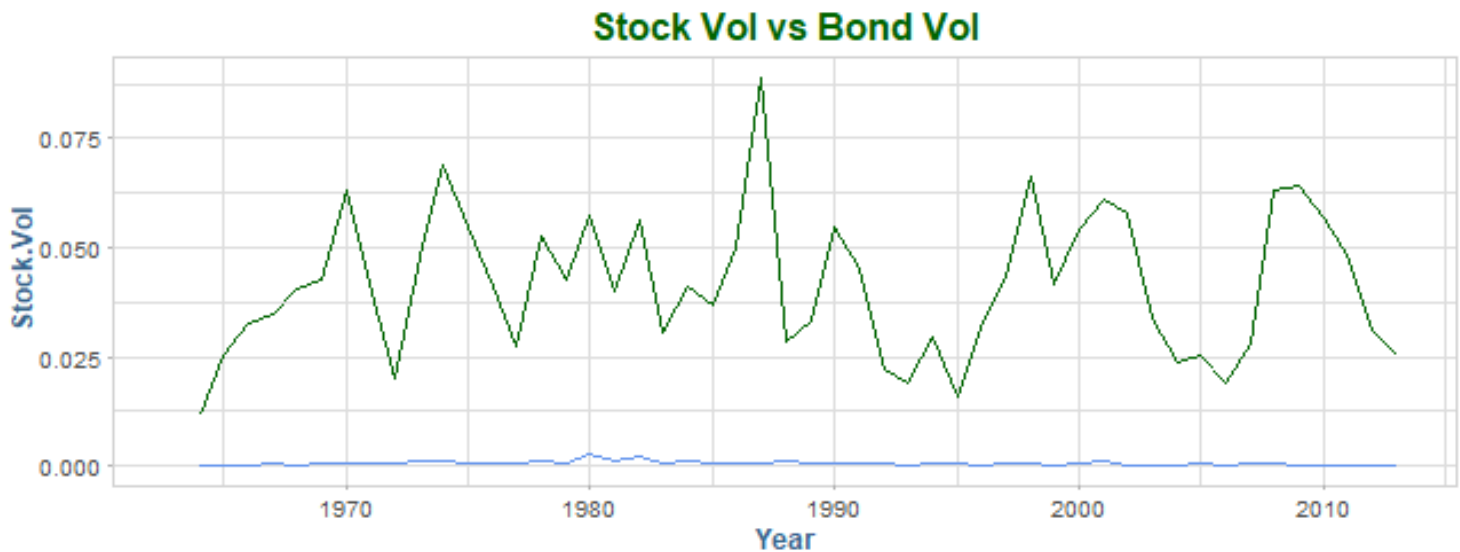
```
ggplot(FF) +  
  geom_line(aes(Date, Cum.Rm), col = "darkgreen") +  
  geom_line(aes(Date, Cum.Rf), col = "cornflowerblue") +  
  labs(y = "Cumulative Return", title = "Stocks Vs. Bonds")
```



```
FF[, Year := year(Date)]  
  
vol <- FF[, .(Stock.Vol = sd(Rm), Bond.Vol = sd(Rf)), by = Year]  
  
ggplot(vol) +  
  geom_line(aes(Year, Stock.Vol), col = "darkgreen") +  
  geom_line(aes(Year, Bond.Vol), col = "cornflowerblue") +  
  labs(title = "Stock Vol vs Bond Vol")
```

Warning: Removed 1 row(s) containing missing values (geom_path).

Warning: Removed 1 row(s) containing missing values (geom_path).



Security Risk

```
date.from <- "2010-12-31"; date.to <- "2013-12-31"
```

```
AMZN <- getSymbols("AMZN", from = date.from, to = date.to, auto.assign = F)
```

'getSymbols' currently uses auto.assign=TRUE by default, but will use auto.assign=FALSE in 0.5-0. You will still be able to use 'loadSymbols' to automatically load data. getOption("getSymbols.env") and getOption("getSymbols.auto.assign") will still be checked for alternate defaults.

This message is shown once per session and may be disabled by setting options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

```
IBM <- getSymbols("IBM", from = date.from, to = date.to, auto.assign = F)
```

```
AMZN$return <- Delt(AMZN$AMZN.Adjusted)
```

```
IBM$return <- Delt(IBM$IBM.Adjusted)
```

```
AMZN <- AMZN[-1]
```

```
IBM <- IBM[-1]
```

```
AMZN.var.full <- var(AMZN$return)
```

```
AMZN.sd.full <- sd(AMZN$return)
```

```
AMZN.returns <- data.table(Date = index(AMZN), AMZN$return)
```

```
AMZN.returns$Year <- year(AMZN.returns$Date)
```

```

AMZN.risk <- melt(AMZN.returns[, .(Mean = lapply(.SD, mean),
  `Std Dev` = lapply(.SD, sd),
  Variance = lapply(.SD, var)),
  by = Year,
  .SDcols = c('Return')], id.vars = c("Mean", "Std Dev", "Variance"), v

setcolororder(AMZN.risk, c("Year", "Mean", "Std Dev", "Variance"))

AMZN.risk$Mean <- lapply(AMZN.risk$Mean, function(x) round(x * 252, 4))
AMZN.risk$`Std Dev` <- lapply(AMZN.risk$`Std Dev`, function(x) round(x * sqrt(252), 4))
AMZN.risk$Variance <- lapply(AMZN.risk$Variance, function(x) round(x * 252, 4))

AMZN.risk

```

	Year	Mean	Std Dev	Variance
1:	2011	0.0349	0.384	0.1475
2:	2012	0.4234	0.3181	0.1012
3:	2013	0.4883	0.2705	0.0732

Multi Asset Approach

```

wgt.AMZN <- .25
wgt.IBM <- .75

port.returns <- data.table(Date = index(AMZN), AMZN = AMZN$Return, IBM = IBM$Return)

sd.AMZN <- sd(port.returns$AMZN.Return) * sqrt(252)
sd.IBM <- sd(port.returns$IBM.Return) * sqrt(252)

ret.cov <- cov(port.returns$AMZN.Return, port.returns$IBM.Return) * 252

# verify
ret.correl <- cor(port.returns$AMZN.Return, port.returns$IBM.Return)
ret.cov.long <- ret.correl * sd.AMZN * sd.IBM

round(ret.cov, 6) == round(ret.cov.long, 6)

[1] TRUE

port.var <- wgt.AMZN^2*sd.AMZN^2 + wgt.IBM^2*sd.IBM^2 + 2*ret.cov*wgt.AMZN*wgt.IBM
port.sd <- sqrt(port.var)

port.var; port.sd

```

```
[1] 0.03626207
```

```
[1] 0.190426
```

Multi-Asset Matrix

```
WGT.2asset <- c(0.25, 0.75)
```

```
WGT.2asset
```

```
[1] 0.25 0.75
```

```
WGT.2asset <- matrix(WGT.2asset, 1)
```

```
tWGT.2asset <- t(WGT.2asset)
```

```
mat.Ret <- as.matrix(port.returns[, 2:3])
```

```
VCOV.2asset <- cov(mat.Ret) * 252
```

```
mat.var2asset <- WGT.2asset %*% VCOV.2asset %*% tWGT.2asset
```

```
mat.2sdasset <- sqrt(mat.var2asset)
```

```
mat.var2asset[1]; mat.2sdasset[1]
```

```
[1] 0.03626207
```

```
[1] 0.190426
```

Multiple Assets

```
port.names <- c("AMZN", "MSFT", "IBM", "TSLA")
```

```
getSymbols(port.names, from = date.from, to = date.to, auto.assign = T)
```

```
[1] "AMZN" "MSFT" "IBM"  "TSLA"
```

```
port.returns <- data.table(AMZN$AMZN.Adjusted, IBM$IBM.Adjusted, MSFT$MSFT.Adjusted, TSLA$TSLA.Adjusted)
```

```
port.returns <- data.table(apply(port.returns, 2, Delt))
```

```
port.returns <- port.returns[-1]
```

```
return.mat <- as.matrix(port.returns)
```

```
colnames(return.mat) <- port.names
```

```
VCOV <- cov(return.mat)
```

```
VCOV.annual <- 252 * VCOV
VCOV.annual

          AMZN      MSFT      IBM      TSLA
AMZN 0.10721389 0.02327634 0.02828851 0.04674476
MSFT 0.02327634 0.03703569 0.02120563 0.02430323
IBM 0.02828851 0.02120563 0.05353378 0.03006055
TSLA 0.04674476 0.02430323 0.03006055 0.33646323

wgt <- c(.2, .2, .3, .3)

mat.wgt <- matrix(wgt, 1)

tmat.wgt <- t(mat.wgt)

port.var <- mat.wgt %*% VCOV.annual %*% tmat.wgt
port.sd <- sqrt(port.var)

port.var[1]; port.sd[1]

[1] 0.06260778
[1] 0.2502155
```

Value-At-Risk

```
ewport <- data.table(cbind(Date = index(AMZN)[-1], port.returns))
ewport <- ewport[Date >= '2012-12-31' &
  Date <= '2013-12-31',]
rownames(ewport) <- seq(1:nrow(ewport))

ewport

      Date AMZN.Adjusted IBM.Adjusted MSFT.Adjusted TSLA.Adjusted
1: 2012-12-31 0.023207448 0.009060607 0.006026504 0.019566465
2: 2013-01-02 0.025670679 0.025058809 0.034069827 0.043991793
3: 2013-01-03 0.004547095 -0.005500527 -0.013396361 -0.016685548
4: 2013-01-04 0.002592011 -0.006554782 -0.018715624 -0.010641300
5: 2013-01-07 0.035925129 -0.004381866 -0.001869558 -0.001744244
---
248: 2013-12-23 0.001790157 0.012276045 -0.004891456 0.002164186
249: 2013-12-24 -0.009232604 0.005432995 0.012561526 0.054754447
250: 2013-12-26 0.013001009 0.011625435 0.009708327 0.027012720
251: 2013-12-27 -0.015603818 -0.001456609 -0.004005964 -0.028167235
252: 2013-12-30 -0.011831773 0.007185980 0.000000000 0.008734827
```

```
ewport$AMZN <- 1 + ewport$AMZN
ewport$MSFT <- 1 + ewport$MSFT
ewport$IBM <- 1 + ewport$IBM
ewport$TSLA <- 1 + ewport$TSLA

# Q1

ew.q1 <- subset(ewport,
                ewport$Date >= as.Date("2012-12-31") &
                ewport$Date <= as.Date("2013-03-31"))

ew.q1[1, 2:4] <- 1
ew.cum.q1 <- ew.q1[, lapply(.SD, cumprod), .SDcols = port.names][, Date := ew.q1$Date]

num.sec <- length(colnames(ew.q1)) - 1

ew.idx.q1 <- ew.cum.q1[, lapply(.SD, function(x)(1/num.sec)*x), .SDcols = port.names][, Date :=
ew.idx.q1[, Value := AMZN + MSFT + IBM + TSLA]

# Q2

ew.q2 <- subset(ewport,
                ewport$Date >= as.Date("2013-3-31") &
                ewport$Date <= as.Date("2013-6-30"))

q2.value <- tail(ew.idx.q1, 1)$Value

ew.cum.q2 <- ew.q2[, lapply(.SD, cumprod), .SDcols = port.names][, Date := ew.q2$Date]

ew.idx.q2 <- ew.cum.q2[, lapply(.SD, function(x)(q2.value/num.sec)*x), .SDcols = port.names][,
ew.idx.q2[, Value := AMZN + MSFT + IBM + TSLA]

# Q3

ew.q3 <- subset(ewport,
                ewport$Date >= as.Date("2013-6-30") &
                ewport$Date <= as.Date("2013-9-30"))

q3.value <- tail(ew.idx.q2, 1)$Value

ew.cum.q3 <- ew.q3[, lapply(.SD, cumprod), .SDcols = port.names][, Date := ew.q3$Date]

ew.idx.q3 <- ew.cum.q3[, lapply(.SD, function(x)(q3.value/num.sec)*x), .SDcols = port.names][,
ew.idx.q3[, Value := AMZN + MSFT + IBM + TSLA]
```

```
# Q4

ew.q4 <- subset(ewport,
               ewport$Date >= as.Date("2013-9-30") &
               ewport$Date <= as.Date("2013-12-31"))

q3.value <- tail(ew.idx.q3, 1)$Value

ew.cum.q4 <- ew.q4[, lapply(.SD, cumprod), .SDcols = port.names][, Date := ew.q4$Date]

ew.idx.q4 <- ew.cum.q4[, lapply(.SD, function(x)(q3.value/num.sec)*x), .SDcols = port.names][,
ew.idx.q4[, Value := AMZN + MSFT + IBM + TSLA]

ew.port.value <- rbind(ew.idx.q1[, .(Date, Value)],
                      ew.idx.q2[, .(Date, Value)],
                      ew.idx.q3[, .(Date, Value)],
                      ew.idx.q4[, .(Date, Value)])

port.return <- (1 + port.returns$AMZN) * (1 + port.returns$MSFT) * (1 + port.returns$IBM) * (1
mean(port.return)

[1] 0.005988539

sd(port.return)

[1] 0.05926488
```