ORIE 4630 Final Project: Communication Services

Benjamin Tang, Ashwin Tayur, Erin Xu

2024-12-14

```
# between the most popular ETFs tracking the communications sector,
# VOX was the only one that started before 2017
getSymbols("VOX", from = "2017-01-01", to = "2019-12-31")
```

```
## [1] "VOX"
```

```
# these are the top 3 largest holdings in VOX, and also the largest
# companies in the sector
getSymbols("GOOG", from = "2017-01-01", to = "2019-12-31")
```

```
## [1] "GOOG"
```

```
getSymbols("META", from = "2017-01-01", to = "2019-12-31")
```

```
## [1] "META"
```

```
getSymbols("NFLX", from = "2017-01-01", to = "2019-12-31")
```

```
## [1] "NFLX"
```

```
# using the 3-month treasury bill as the risk free rate
 DTB3 <- read.csv(paste0("./DTB3.csv"), header = TRUE)
 DTB3 <- DTB3[!is.na(DTB3$DTB3),]

# risk free rate (1-month treasury bill)
tbill1mo= getSymbols("DGS1MO", from = "2017-01-01", to = "2019-12-31", src="FRED", auto.assign = FALSE)
names(tbill1mo) = tolower(names(tbill1mo))
tbill1mo= na.omit(tbill1mo[,'dgs1mo'])
```

Question 1

Calculate daily returns for your industry for 2017-2019. Download two types of data for this - the industry level series on Yahoo Finance, and the top 3 or 4 companies in the sector. Justify the ticker choice you made to represent the industries (e.g., the most famous ones, largest ones, ones you are particularly curious about, etc.). Plot the cumulative performance of the industry and the individual assets.

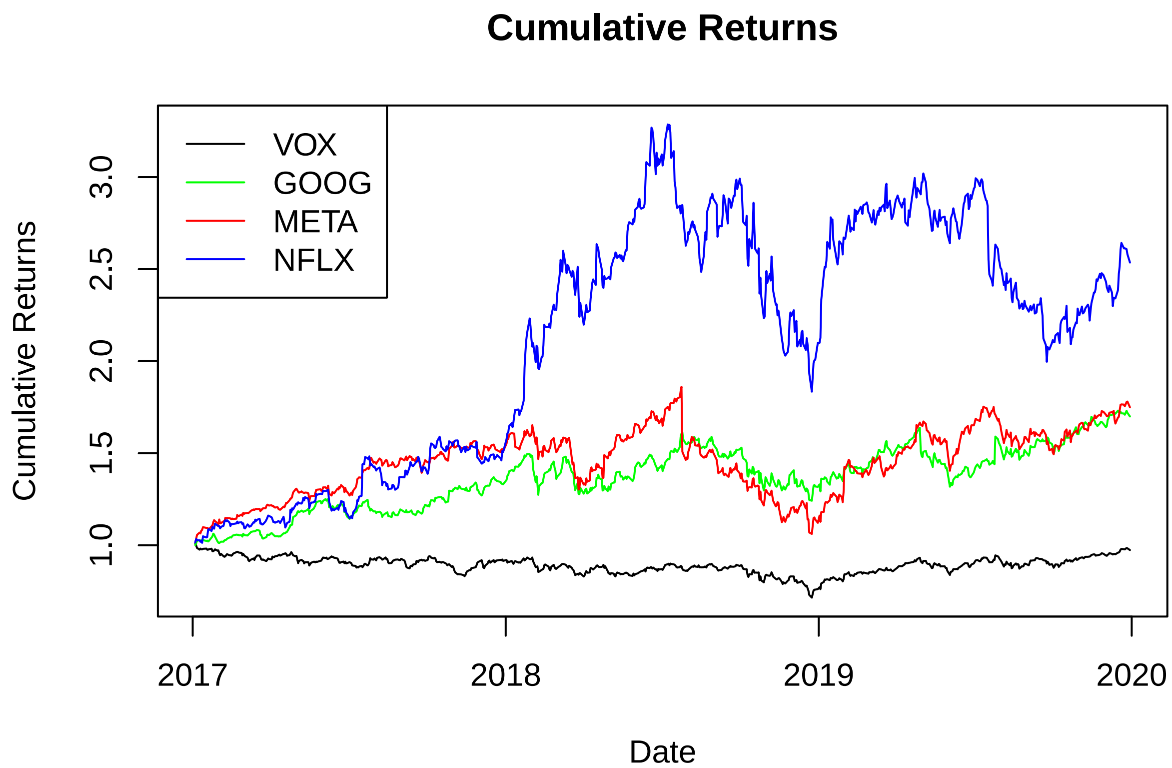
For the industry, we chose to use Vanguard Communication Services Index Fund ETF (VOX). VOX is one of the largest ETFs in the sector, and has been traded since October 2004. For comparison,the SPDR Communications Sector ETF (XLC), only began trading in July, 2018.

For the top companies, we chose the three largest in the sector, namely, GOOG, META, and NFLX. These three companies are the largest holdings in VOX, and the largest companies by market cap in the sector.

```
VOX_returns <- dailyReturn(VOX$VOX.Adjusted, type = "arithmetic")[-1]
GOOG_returns <- dailyReturn(GOOG$GOOG.Adjusted, type = "arithmetic")[-1]
META_returns <- dailyReturn(META$META.Adjusted, type = "arithmetic")[-1]
NFLX_returns <- dailyReturn(NFLX$NFLX.Adjusted, type = "arithmetic")[-1]

returns <- merge(VOX_returns, GOOG_returns, META_returns, NFLX_returns)
colnames(returns) <- c("VOX", "GOOG", "META", "NFLX")
head(returns)
```

```
##                     VOX          GOOG         META          NFLX
## 2017-01-04  0.006865628  0.0009668652  0.015659715  0.0150600504
## 2017-01-05 -0.007298879  0.0090481495  0.016682145  0.0185456598
## 2017-01-06 -0.013833852  0.0152766825  0.022706605 -0.0056140676
## 2017-01-09 -0.009711716  0.0006203140  0.012073573 -0.0009156204
## 2017-01-10  0.005943621 -0.0023060122 -0.004403633 -0.0080946742
## 2017-01-11 -0.003840341  0.0038768784  0.013992722  0.0046962862
```

```
VOX_cum <- cumprod(1 + VOX_returns)
GOOG_cum <- cumprod(1 + GOOG_returns)
META_cum <- cumprod(1 + META_returns)
NFLX_cum <- cumprod(1 + NFLX_returns)

plot(index(VOX_cum), VOX_cum, type = "l", col = "black", lwd = 1,
     xlab = "Date", ylab = "Cumulative Returns", main = "Cumulative Returns",
     ylim = range(c(VOX_cum, GOOG_cum, META_cum, NFLX_cum), na.rm = TRUE))
lines(index(GOOG_cum), GOOG_cum, col = "green", lwd = 1)
lines(index(META_cum), META_cum, col = "red", lwd = 1)
lines(index(NFLX_cum), NFLX_cum, col = "blue", lwd = 1)
legend("topleft", legend = c("VOX", "GOOG", "META", "NFLX"),
       col = c("black", "green", "red", "blue"), lty = 1, lwd = 1)
```



Question 2

Report summary statistics like mean and sd for daily returns by year and industry i.e., in two Nx2 table with periods along the columns and assets along the rows. N denotes the number of assets (i.e., number of individual companies and the overall series). One table is for reporting mean, the other for sd.

```
returns_by_year <- split(returns, format(index(returns), "%Y"))

mean_table <- sapply(returns_by_year, function(year_data) colMeans(year_data, na.rm = TRUE))
sd_table <- sapply(returns_by_year, function(year_data) apply(year_data, 2, sd, na.rm = TRUE))

cat("Mean of Daily Returns, by Year (2017-2019):\n")
```

```
## Mean of Daily Returns, by Year (2017-2019):
```

```
print(mean_table)
```

```
##               2017          2018         2019
## VOX  -0.0003190945 -0.0006583959 0.0010236314
## GOOG  0.0011913282  0.0001158744 0.0011306652
## META  0.0017069666 -0.0008865413 0.0019244564
## NFLX  0.0017891463  0.0017513091 0.0009908407
```

```r
cat("\nStandard Deviation of Daily Returns, by Year (2017-2019):\n")
```

```
##
## Standard Deviation of Daily Returns, by Year (2017-2019):
```

```r
print(sd_table)
```

```
##             2017       2018       2019
## VOX  0.008447891 0.01197733 0.00948216
## GOOG 0.009695776 0.01772353 0.01523799
## META 0.010700617 0.02397114 0.01759557
## NFLX 0.017563650 0.02924640 0.02185288
```

Question 3

Use techniques we learnt in portfolio optimization to create an optimal portfolio of the individual companies in your industry. Comment on the riskiness of the assets, the overall series and your constructed portfolio in the Markowitz portfolio optimization world (meaning using standard deviations and Sharpe ratios).

update this: I am using the 3-month treasury bills as the risk-free rate. https://fred.stlouisfed.org/series/DTB3

SOFR was only caluclated since 04-2018.

```r
mu.vec <- colMeans(returns, na.rm = TRUE)
sd.vec <- apply(returns, 2, sd, na.rm = TRUE)
trading_days <- nrow(returns)

#rf_annual <- mean(DTB3$DTB3[DTB3$observation >= as.Date("2017-01-01") & DTB3$observation <= as.Date("2
rf_annual <- mean(tbill1mo)
rf <- rf_annual / trading_days / 100
Sigma = cov(returns[,-1])
```

Calculate the tangency portfolio

```r
num = solve(Sigma)%*%(mu.vec[-1]-rf)
den = as.numeric(t(rep(1,3))%*%solve(Sigma)%*%(mu.vec[-1]-rf))
tan.vec = num/den
mu_tan = as.numeric(crossprod(tan.vec, mu.vec[-1]))
sd_tan = sqrt(as.numeric(t(tan.vec)%*%Sigma%*%tan.vec))

tanpf = rbind(mu_tan, sd_tan, tan.vec)
rownames(tanpf) = c("mean", "sd", "GOOG", "META", "NFLX")
colnames(tanpf) = c("tangency pf")

print(round(tanpf,3))
```

```
##      tangency pf
## mean       0.001
## sd         0.017
```

```
## GOOG       0.315
## META       0.211
## NFLX       0.474
```

```r
voxpf = rbind(mu.vec[1], sd.vec[1])
rownames(voxpf) = c("mean", "sd")
print(voxpf)
```

```
##               VOX
## mean 1.582511e-05
## sd   1.009340e-02
```

Calculate the Sharpe Ratios

```r
top.mat = cbind(2*Sigma, rep(1, 3))
bot.vec = c(rep(1, 3), 0)
Am.mat = rbind(top.mat, bot.vec)
b.vec = c(rep(0, 3), 1)
zm.mat = solve(Am.mat)%*%b.vec
wmin.vec = zm.mat[1:3,1]
mu_min = as.numeric(crossprod(wmin.vec, mu.vec[-1]))
sd_min = sqrt(as.numeric(t(wmin.vec)%*%Sigma%*%wmin.vec))

SR_VOX = (mu.vec[1]-rf)/sd.vec[1]
SR_GOOG = (mu.vec[2]-rf)/sd.vec[2]
SR_META = (mu.vec[3]-rf)/sd.vec[3]
SR_NFLX = (mu.vec[4]-rf)/sd.vec[4]
SR_min = (mu_min-rf)/sd_min
SR_tan = (tanpf[1] - rf)/tanpf[2]
sr = rbind(SR_VOX, SR_GOOG,SR_META, SR_NFLX, SR_min, SR_tan)
rownames(sr) = c("SR_VOX", "SR_GOOG","SR_META", "SR_NFLX", "SR min var", "SR tangency")
colnames(sr) = c("Sharpe Ratios")
round(sr*100,3)
```

```
##             Sharpe Ratios
## SR_VOX             -0.054
## SR_GOOG             5.415
## SR_META             4.884
## SR_NFLX             6.370
## SR min var          5.871
## SR tangency         6.873
```

Analysis

Looking at the individuals stocks, we see that Netflix has the highest Sharpe Ratio of 6.368, and thus has the most attractive risk-adjusted return. However, Google and Meta are not far behind at 5.411 and 4.881. Thus, the tangential portfolio puts the highest weight on NFLX, but still a significant proportion in the others. All of them are above 3 and thus very good stocks to invest in. The Sharpe Ratio of the tangency portfolio is higher than each individual stock, at 6.870. This is much higher than the Sharpe Ratio of VOX, which is at -0.060.

The tangency porfolio performs quite well, averaging 0.1% growth per trading day but with a high standard deviation of 1.7%. Compared to the VOX, which averages a much lower 0.002% growth per trading day and with a smaller standard deviation of 1.0%. VOX is a safer investment for sure, but gives much worse expected returns, even when adjusted for risk.

TODO: Make graph (Markowitz bullet), check to make sure my calculations (0.1%, 0.002%, etc. are correct and per day)

Question 4

Compare the performance of the pf you constructed to the overall series dur- ing your sample period, and also over the next 3 years, namely, 2020-2022. Comment on your findings.

2017-2019

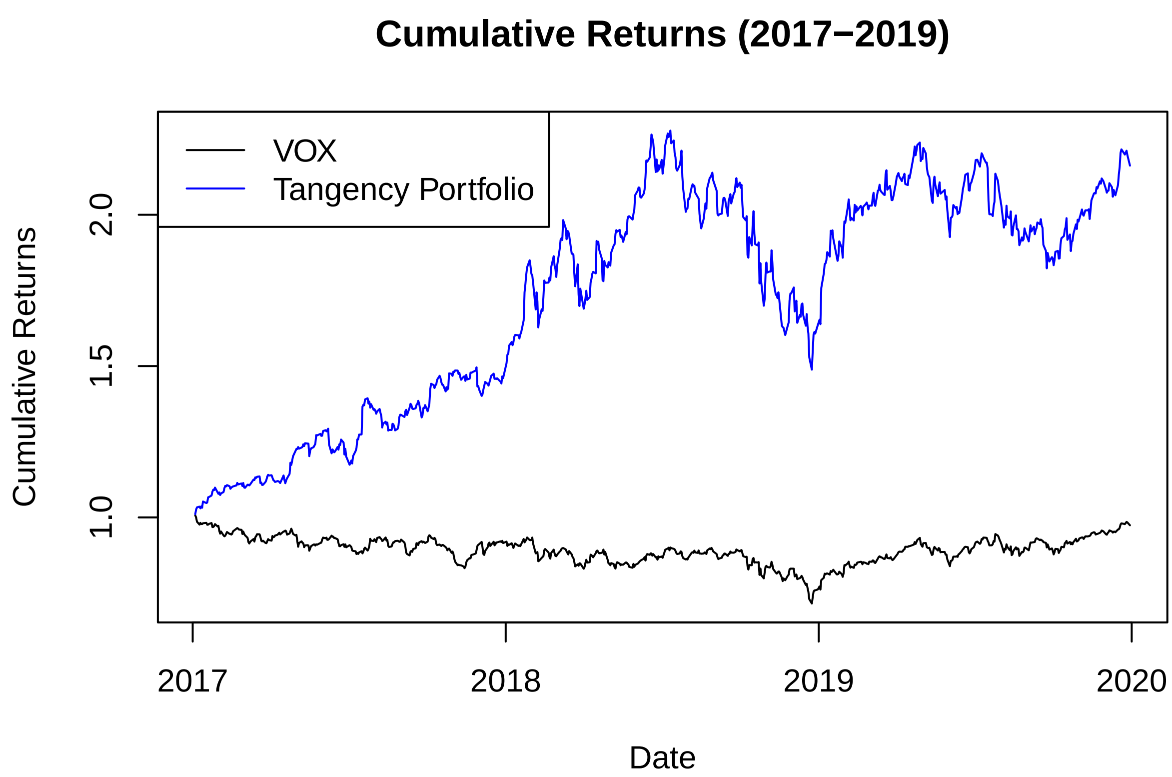
```r
tan_returns <- rowSums(returns[,-1] %*% tan.vec, na.rm = TRUE)
tan_cum <- cumprod(1 + tan_returns)

# Convert to time series
VOX_cum <- xts(VOX_cum, order.by = index(VOX_returns))
tan_cum <- xts(tan_cum, order.by = index(returns[,-1]))
```

2020-2022

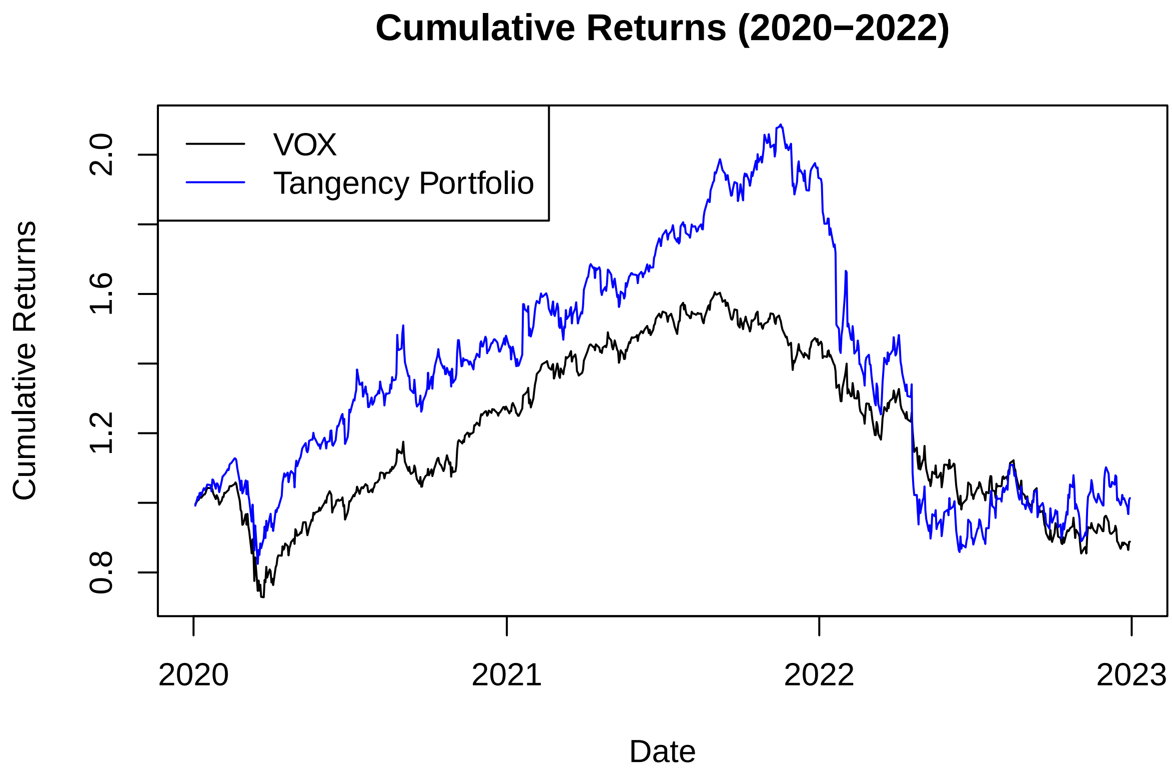
```r
# Fetch next 3 years (2020-2022)
VOX_new <- getSymbols("VOX", from = "2020-01-01", to = "2022-12-31", auto.assign = FALSE)
GOOG_new <- getSymbols("GOOG", from = "2020-01-01", to = "2022-12-31", auto.assign = FALSE)
META_new <- getSymbols("META", from = "2020-01-01", to = "2022-12-31", auto.assign = FALSE)
NFLX_new <- getSymbols("NFLX", from = "2020-01-01", to = "2022-12-31", auto.assign = FALSE)

VOX_returns_2020 <- dailyReturn(VOX_new$VOX.Adjusted, type = "arithmetic")[-1]
GOOG_returns_2020 <- dailyReturn(GOOG_new$GOOG.Adjusted, type = "arithmetic")[-1]
META_returns_2020 <- dailyReturn(META_new$META.Adjusted, type = "arithmetic")[-1]
NFLX_returns_2020 <- dailyReturn(NFLX_new$NFLX.Adjusted, type = "arithmetic")[-1]

returns_2020 <- merge(GOOG_returns_2020, META_returns_2020, NFLX_returns_2020)
colnames(returns_2020) <- c("GOOG", "META", "NFLX")
tan_returns_2020 <- rowSums(returns_2020 %*% tan.vec, na.rm = TRUE)

# Calculate cumulative returns
VOX_cum_2020 <- cumprod(1 + VOX_returns_2020)
tan_cum_2020 <- cumprod(1 + tan_returns_2020)

# Convert to time series
VOX_cum_2020 <- xts(VOX_cum_2020, order.by = index(VOX_returns_2020))
tan_cum_2020 <- xts(tan_cum_2020, order.by = index(returns_2020))
```

Plots

```r
# Plot for 2017-2019
plot(index(VOX_cum), VOX_cum, type = "l", col = "black", lwd = 1,
     xlab = "Date", ylab = "Cumulative Returns", main = "Cumulative Returns (2017-2019)",
     ylim = range(c(VOX_cum, tan_cum), na.rm = TRUE))
lines(index(tan_cum), tan_cum, col = "blue", lwd = 1)
legend("topleft", legend = c("VOX", "Tangency Portfolio"),
       col = c("black", "blue"), lty = 1, lwd = 1)
```




```r
# Plot for 2020-2022
plot(index(VOX_cum_2020), VOX_cum_2020, type = "l", col = "black", lwd = 1,
     xlab = "Date", ylab = "Cumulative Returns", main = "Cumulative Returns (2020-2022)",
     ylim = range(c(VOX_cum_2020, tan_cum_2020), na.rm = TRUE))
lines(index(tan_cum_2020), tan_cum_2020, col = "blue", lwd = 1)
legend("topleft", legend = c("VOX", "Tangency Portfolio"),
       col = c("black", "blue"), lty = 1, lwd = 1)
```



Analysis

Looking at the plots above, we found that the cumulative returns for our portfolio performs much better than VOX in 2017-2019, confirming our expectations from the previous question. However, in 2020-2022, the tangency portfolio seems to outperform VOX somewhat until the end of 2021, from which it performs worse. By the end of 2022, the cumulative returns for our portfolio and VOX end up to be approximately the same. A major factor in the sharp decline of the tangency portfolio is the struggles of Netflix following the pandemic. The pandemic caused a significant growth in streaming, as movie theaters closed. This led to many companies creating their own streaming platforms, which eroded the industry dominance of Netflix. The pandemic can be seen as a black swan for Netflix, as it led to a far more competitive industry, causing Netflix to lose 51% over the year 2022. As the tangency portfolio weights Netflix more than VOX (47.4% vs. 4.55%), the black swan affects the tangency portfolio far more.

TODO: Is this sufficient? Might be useful to analyze excess returns or sharpe ratios (?)

TODO: Tried to add analysis for the 2021 drop, its mostly Netflix losing 60% that quarter

Question 5

Use CAPM to estimate the alphas and betas for the individual companies, your constructed pf, and the overall series with respect to SP500 during 2017-2019. Comment on your findings

```
# sp500 index. GSPC = Global Standard & Poor's Composite
sp500 = getSymbols("^GSPC", from = "2017-01-01", to = "2019-12-31", auto.assign = FALSE)
names(sp500) = tolower(names(sp500))
sp500 = sp500[,'gspc.adjusted']

names(GOOG) = tolower(names(GOOG))
goog = GOOG[,'goog.adjusted']
names(META) = tolower(names(META))
meta = META[,'meta.adjusted']
names(NFLX) = tolower(names(NFLX))
nflx = NFLX[,'nflx.adjusted']
names(VOX) = tolower(names(VOX))
vox = VOX[, 'vox.adjusted']

rm = monthlyReturn(sp500$gspc.adjusted, type = "arithmetic")
names(rm) = c("mkt")
rf = tbill1mo*30/(360*100)
names(rf) = c("rf")

tang = tan.vec[1] * goog + tan.vec[2] * meta + tan.vec[3] * nflx

ri_goog = monthlyReturn(goog$goog.adjusted, type = "arithmetic")
ri_meta = monthlyReturn(meta$meta.adjusted, type = "arithmetic")
ri_nflx = monthlyReturn(nflx$nflx.adjusted, type = "arithmetic")
ri_vox =  monthlyReturn(vox$vox.adjusted, type = "arithmetic")
ri_tang =  monthlyReturn(tang, type = "arithmetic")
names(ri_goog) = c("goog")
names(ri_meta) = c("meta")
names(ri_nflx) = c("nflx")
names(ri_vox) = c("vox")
names(ri_tang) = c("tang")

tmp1 = merge.xts(rm,rf,join="inner")
tmp2 = merge.xts(tmp1,ri_goog,join="inner")
tmp3 = merge.xts(tmp2,ri_meta,join="inner")
tmp4 = merge.xts(tmp3,ri_vox,join="inner")
tmp5 = merge.xts(tmp4,ri_tang,join="inner")
all_ret = merge.xts(tmp5,ri_nflx,join="inner")

ex_goog = all_ret[,'goog']-all_ret[,'rf']
ex_meta = all_ret[,'meta']-all_ret[,'rf']
```

```r
ex_nflx = all_ret[,'nflx']-all_ret[,'rf']
ex_vox = all_ret[,'vox']-all_ret[,'rf']
ex_tang = all_ret[,'tang']-all_ret[,'rf']
ex_mkt = all_ret[,'mkt']-all_ret[,'rf']
fit_goog = lm(ex_goog~ex_mkt)
fit_meta = lm(ex_meta~ex_mkt)
fit_nflx = lm(ex_nflx~ex_mkt)
fit_vox = lm(ex_vox~ex_mkt)
fit_tang = lm(ex_tang~ex_mkt)

beta_goog = round(fit_goog$coefficients[2],3)
beta_meta = round(fit_meta$coefficients[2],3)
beta_nflx = round(fit_nflx$coefficients[2],3)
beta_vox = round(fit_vox$coefficients[2], 3)
beta_tang = round(fit_tang$coefficients[2], 3)

alpha_goog = round(fit_goog$coefficients[1],3)
alpha_meta = round(fit_meta$coefficients[1],3)
alpha_nflx = round(fit_nflx$coefficients[1],3)
alpha_vox = round(fit_vox$coefficients[1], 3)
alpha_tang = round(fit_tang$coefficients[1], 3)
prob_goog = round(summary(fit_goog)$coefficients[1, "Pr(>|t|)"], 3)
prob_meta = round(summary(fit_meta)$coefficients[1, "Pr(>|t|)"], 3)
prob_nflx = round(summary(fit_nflx)$coefficients[1, "Pr(>|t|)"], 3)
prob_tang = round(summary(fit_tang)$coefficients[1, "Pr(>|t|)"], 3)
prob_vox = round(summary(fit_vox)$coefficients[1, "Pr(>|t|)"], 3)

alpha_beta = matrix(c(alpha_goog, alpha_meta, alpha_nflx, alpha_vox, alpha_tang, beta_goog, beta_meta, b
colnames(alpha_beta) = c("GOOG", "META", "NFLX", "VOX", "Tangency")
rownames(alpha_beta) = c("Alpha", "Beta", "Pr(>|t|)")

alpha_beta
```

```
##           GOOG  META  NFLX    VOX Tangency
## Alpha    0.006 0.005 0.015 -0.009    0.011
## Beta     0.998 1.383 1.709  0.826    1.556
## Pr(>|t|) 0.414 0.685 0.384  0.033    0.411
```

Using CAPM, we find that all three stocks do not have a significant alpha, which implies that CAPM holds for them. The tangency portfolio also has an insignificant alpha. However, it appears VOX has a significant negative alpha.

The Beta coefficient for GOOG shows that it has a similar systematic risk to the market, while the Betas for META and NFLX both show higher systematic risk compared to the market. VOX has a beta less than 1, implying a lower systemic risk in relation to the market. The tangency protfolio has a beta greater than 1 as well, which is expected, as is is comprised of long-only positions of three stocks with beta greater than 1.

TODO: Anything else? Not sure what else to say

Question 6

Use CAPM to estimate the betas for the individual companies, your con- structed pf, and the overall series with respect to SP500 during 2020-2022. Compare with the findings for 2017-2019

```
sp500_new = getSymbols("^GSPC", from = "2020-01-01", to = "2022-12-31", auto.assign = FALSE)
names(sp500_new) = tolower(names(sp500_new))
sp500_new = sp500_new[,'gspc.adjusted']

names(GOOG_new) = tolower(names(GOOG_new))
goog_new = GOOG_new[,'goog.adjusted']
names(META_new) = tolower(names(META_new))
meta_new = META_new[,'meta.adjusted']
names(NFLX_new) = tolower(names(NFLX_new))
nflx_new = NFLX_new[,'nflx.adjusted']
names(VOX_new) = tolower(names(VOX_new))
vox_new = VOX_new[, 'vox.adjusted']

tang_new = tan.vec[1] * goog_new + tan.vec[2] * meta_new + tan.vec[3] * nflx_new


rm_new = monthlyReturn(sp500_new$gspc.adjusted, type = "arithmetic")
names(rm_new) = c("mkt")
tbill1mo_new= getSymbols("DGS1MO", from = "2020-01-01", to = "2022-12-31", src="FRED", auto.assign = FA
names(tbill1mo_new) = tolower(names(tbill1mo_new))
tbill1mo_new = na.omit(tbill1mo_new[,'dgs1mo'])
rf_new = tbill1mo_new*30/(360*100)
names(rf_new) = c("rf")

ri_goog_new = monthlyReturn(goog_new$goog.adjusted, type = "arithmetic")
ri_meta_new = monthlyReturn(meta_new$meta.adjusted, type = "arithmetic")
ri_nflx_new = monthlyReturn(nflx_new$nflx.adjusted, type = "arithmetic")
ri_vox_new =  monthlyReturn(vox_new$vox.adjusted, type = "arithmetic")
ri_tang_new =  monthlyReturn(tang_new, type = "arithmetic")
names(ri_goog_new) = c("goog")
names(ri_meta_new) = c("meta")
names(ri_nflx_new) = c("nflx")
names(ri_vox_new) = c("vox")
names(ri_tang_new) = c("tang")

tmp1 = merge.xts(rm_new,rf_new,join="inner")
tmp2 = merge.xts(tmp1,ri_goog_new,join="inner")
tmp3 = merge.xts(tmp2,ri_meta_new,join="inner")
tmp4 = merge.xts(tmp3,ri_tang_new,join="inner")
tmp5 = merge.xts(tmp4,ri_vox_new,join="inner")
all_ret_new = merge.xts(tmp5,ri_nflx_new,join="inner")

ex_goog_new = all_ret_new[,'goog']-all_ret_new[,'rf']
ex_meta_new = all_ret_new[,'meta']-all_ret_new[,'rf']
ex_nflx_new = all_ret_new[,'nflx']-all_ret_new[,'rf']
ex_vox_new = all_ret_new[,'vox']-all_ret_new[,'rf']
ex_tang_new = all_ret_new[,'tang']-all_ret_new[,'rf']
ex_mkt_new = all_ret_new[,'mkt']-all_ret_new[,'rf']
fit_goog_new = lm(ex_goog_new~ex_mkt_new)
fit_meta_new = lm(ex_meta_new~ex_mkt_new)
fit_nflx_new = lm(ex_nflx_new~ex_mkt_new)
fit_vox_new = lm(ex_vox_new~ex_mkt_new)
fit_tang_new = lm(ex_tang_new~ex_mkt_new)
```

```r
beta_goog_new = round(fit_goog_new$coefficients[2],3)
beta_meta_new = round(fit_meta_new$coefficients[2],3)
beta_nflx_new = round(fit_nflx_new$coefficients[2],3)
beta_vox_new = round(fit_vox_new$coefficients[2], 3)
beta_tang_new = round(fit_tang_new$coefficients[2], 3)

alpha_goog_new = round(fit_goog_new$coefficients[1],3)
alpha_meta_new = round(fit_meta_new$coefficients[1],3)
alpha_nflx_new = round(fit_nflx_new$coefficients[1],3)
alpha_vox_new = round(fit_vox_new$coefficients[1], 3)
alpha_tang_new = round(fit_tang_new$coefficients[1], 3)

prob_goog_new = round(summary(fit_goog_new)$coefficients[1, "Pr(>|t|)"], 3)
prob_meta_new = round(summary(fit_meta_new)$coefficients[1, "Pr(>|t|)"], 3)
prob_nflx_new = round(summary(fit_nflx_new)$coefficients[1, "Pr(>|t|)"], 3)
prob_tang_new = round(summary(fit_tang_new)$coefficients[1, "Pr(>|t|)"], 3)
prob_vox_new = round(summary(fit_vox_new)$coefficients[1, "Pr(>|t|)"], 3)

alpha_beta_new = matrix(c(alpha_goog_new, alpha_meta_new, alpha_nflx_new, alpha_vox_new, alpha_tang_new
colnames(alpha_beta_new) = c("GOOG", "META", "NFLX", "VOX", "Tangency")
rownames(alpha_beta_new) = c("Alpha", "Beta", "Pr(>|t|)")

alpha_beta_new
```

```
##            GOOG   META  NFLX    VOX Tangency
## Alpha    0.004 -0.014 0.000 -0.008   -0.005
## Beta     1.112  1.089 1.107  1.034    1.068
## Pr(>|t|) 0.666  0.434 0.997  0.116    0.683
```

Compared to 2017-2019, there are some similarities and some differences. While the Betas for META, and NFLX all remain above 1, they decreased significantly, from 1.383 and 1.709 to 1.089 and 1.107 respectively. The Beta for GOOG also changed, jumping from 0.998, below 1, to 1.112, greater than 1. The Beta for VOX also changed to be greater than 1. VOX has the biggest difference, as it now no longer has a statistically significant alpha. Interestingly, unlike in 2017-2019, the Beta for the tangency portfolio is lower than any of the sub-components.

#TODO: no idea what else to write.

Question 7

Assuming an initial investment of S = $1000.

```r
s <- 1000
calc_var_es <- function(alpha, ret) {
  res <- numeric(2)
  res[1] <- -s*as.numeric(quantile(ret,alpha))
  L = -s*ret
  res[2] <- mean(L[L>res[1]])
  names(res) <- c("var", "es")
  return(res)
}
```

```r
goog_var_es = calc_var_es(0.05, goog)
meta_var_es = calc_var_es(0.05, meta)
nflx_var_es = calc_var_es(0.05, nflx)
all_var_es = matrix(c(goog_var_es["var"], meta_var_es["var"], nflx_var_es["var"], goog_var_es["es"], me
rownames(all_var_es) = c("VaR", "ES")
colnames(all_var_es) = c("GOOG", "META", "NFLX")
all_var_es
```

```
##           GOOG       META    NFLX
## VaR  -41078.9 -142250.00 -128649
## ES  -133390.4  -40349.26 -137685
```

Question 8

```r
goog_var_es_new = calc_var_es(0.05, goog_new)
meta_var_es_new = calc_var_es(0.05, meta_new)
nflx_var_es_new = calc_var_es(0.05, nflx_new)
all_var_es_new = matrix(c(goog_var_es_new["var"], meta_var_es_new["var"], nflx_var_es_new["var"], goog_v
rownames(all_var_es_new) = c("VaR", "ES")
colnames(all_var_es_new) = c("GOOG", "META", "NFLX")
all_var_es_new
```

```
##            GOOG       META      NFLX
## VaR  -65704.04 -190087.50 -108965.9
## ES  -119692.12  -59330.91 -180185.5
```