ORIE 4630 Final Project: Communication Services

Benjamin Tang, Ashwin Tayur, Erin Xu

2024-12-14

```r
# 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"
```

```r
# 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"
```

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

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

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

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

```r
# 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.

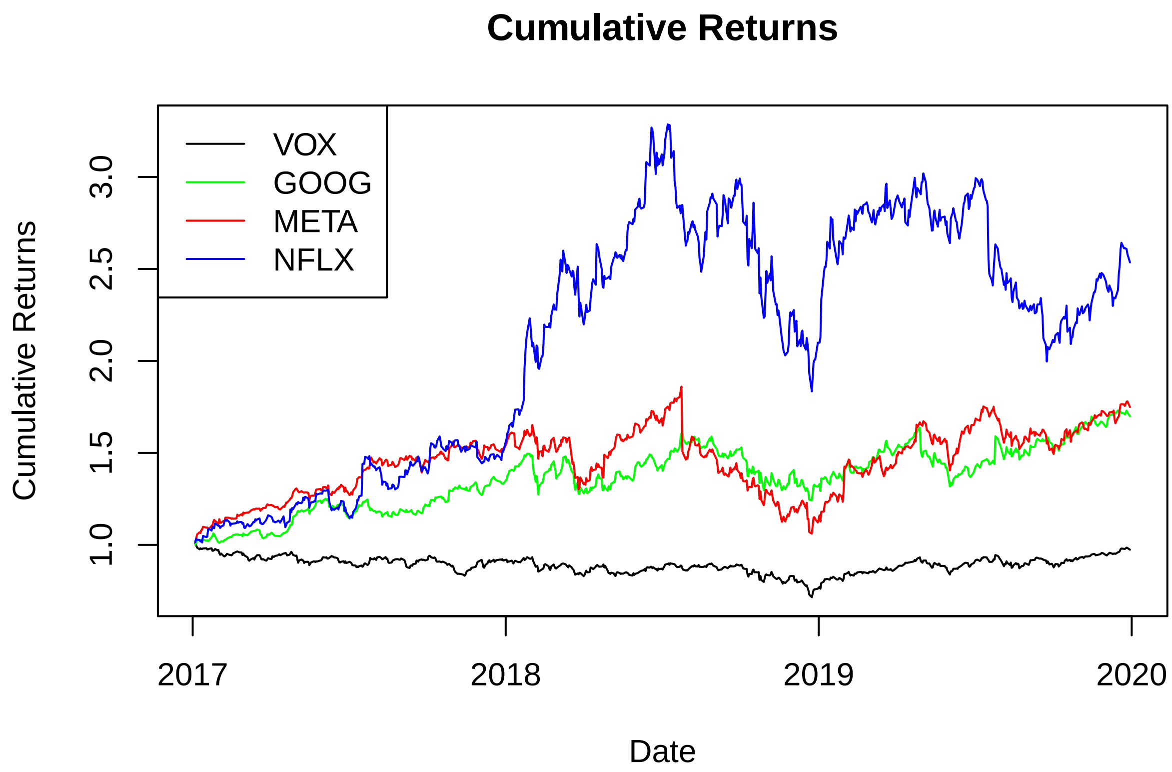
```r
VOX_returns <- dailyReturn(VOX$VOX.Adjusted, type = "arithmetic")[-1]
GOOG_returns <- dailyReturn(GOOG$GOOG.Adjusted, type = "arithmetic")[-1]
META_returns <- dailyReturn(META$META.Adjusted, type = "arithmetic")[-1]
NFLX_returns <- dailyReturn(NFLX$NFLX.Adjusted, type = "arithmetic")[-1]

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  0.006865129  0.0009668652  0.015659715  0.0150600504
## 2017-01-05 -0.007298387  0.0090481495  0.016682145  0.0185456598
## 2017-01-06 -0.013833684  0.0152766825  0.022706605 -0.0056140676
## 2017-01-09 -0.009711881  0.0006203140  0.012073573 -0.0009156204
## 2017-01-10  0.005943535 -0.0023060122 -0.004403633 -0.0080946742
## 2017-01-11 -0.003840510  0.0038768784  0.013992722  0.0046962862
```

```r
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.

By year from 2017-2019

```
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(round(mean_table,3))
```

```
##       2017   2018  2019
## VOX  0.000 -0.001 0.001
## GOOG 0.001  0.000 0.001
```

```
## META 0.002 -0.001 0.002
## NFLX 0.002  0.002 0.001
```

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

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

```r
print(round(sd_table,3))
```

```
##       2017  2018  2019
## VOX  0.008 0.012 0.009
## GOOG 0.010 0.018 0.015
## META 0.011 0.024 0.018
## NFLX 0.018 0.029 0.022
```

By time period

```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(VOX_returns_2020, GOOG_returns_2020, META_returns_2020, NFLX_returns_2020)
colnames(returns_2020) <- c("VOX", "GOOG", "META", "NFLX")

# calculate mean, sd by period
mu.vec <- colMeans(returns, na.rm = TRUE)
sd.vec <- apply(returns, 2, sd, na.rm = TRUE)
mu2020.vec <- colMeans(returns_2020, na.rm = TRUE)
sd2020.vec <- apply(returns_2020, 2, sd, na.rm = TRUE)

mu_table = cbind(round(mu.vec,3), round(mu2020.vec,3))
colnames(mu_table) <- c("2017-2019", "2020-2022")
cat("Mean of Daily Returns, by Time Period (2017-2019, 2020-2022):\n")
```

```
## Mean of Daily Returns, by Time Period (2017-2019, 2020-2022):
```

```r
mu_table
```

```
##      2017-2019 2020-2022
## VOX      0.000     0.000
## GOOG     0.001     0.001
## META     0.001     0.000
## NFLX     0.002     0.000
```

```r
sd_table <- cbind(round(sd.vec,3), round(sd2020.vec,3))
colnames(sd_table) <- c("2017-2019", "2020-2022")
cat("Standard Deviation of Daily Returns, by Time Period (2017-2019, 2020-2022):\n")
```

```
## Standard Deviation of Daily Returns, by Time Period (2017-2019, 2020-2022):
```

```r
sd_table
```

```
##      2017-2019 2020-2022
## VOX      0.010     0.017
## GOOG     0.015     0.022
## META     0.018     0.031
## NFLX     0.023     0.033
```

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).

We are using the 1-month treasury bills as the risk-free rate.

```r
rf_annual <- mean(tbill1mo)
rf <- rf_annual / 360 / 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       0.303
## META       0.211
## NFLX       0.486
```

```r
voxpf = rbind(mu.vec[1], sd.vec[1])
rownames(voxpf) = c("mean", "sd")
print(round(voxpf,3))
```

```
##       VOX
## mean 0.00
## sd   0.01
```

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)
```

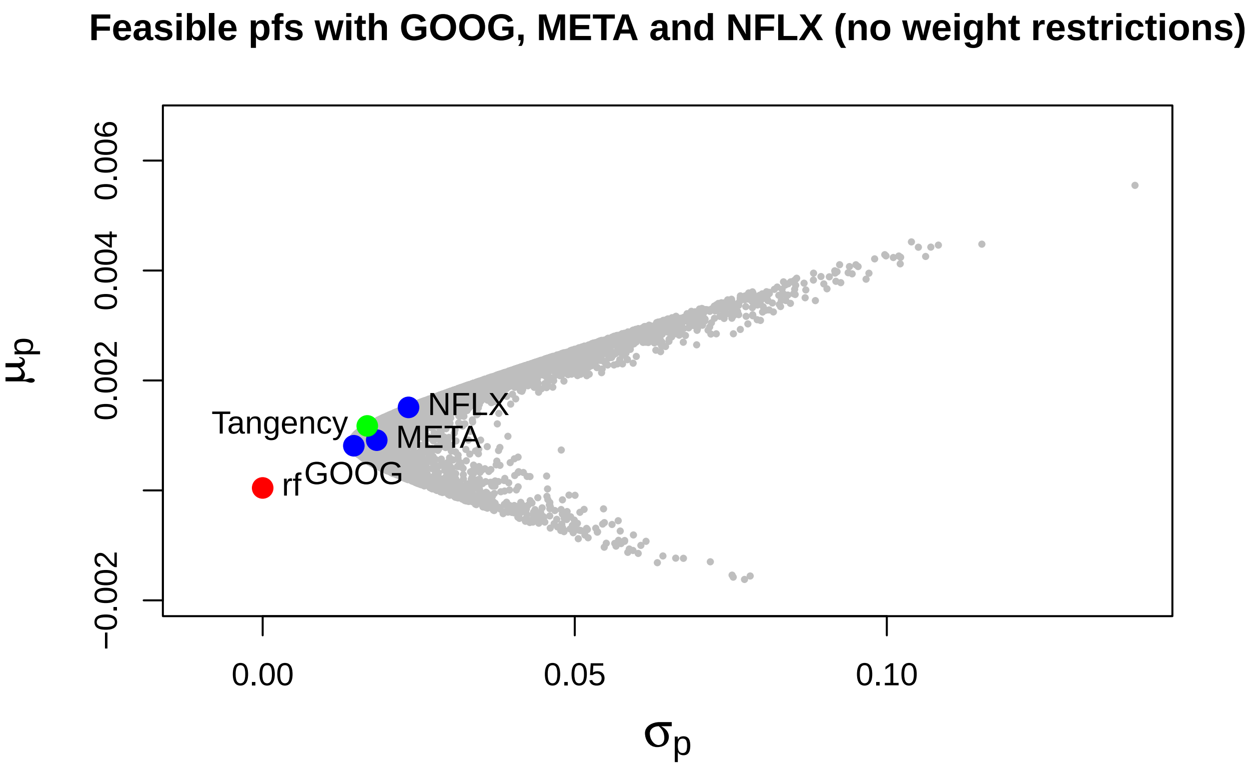
```
##             Sharpe Ratios
## SR_VOX             -0.285
## SR_GOOG             5.256
## SR_META             4.757
## SR_NFLX             6.271
## SR min var          5.707
## SR tangency         6.734
```

Analysis

Looking at the individuals stocks, we see that Netflix has the highest Sharpe Ratio of 6.271, and thus has the most attractive risk-adjusted return. However, Google and Meta are not far behind at 5.256 and 4.757. 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 6.734. This is much higher than the Sharpe Ratio of VOX, which is at -0.284. We also see that the returns are quite high, at 0.1%, 0.1%, and 0.2% per trading day for GOOG, META, NFLX respectively. The standard deviations however, are also higher, at 1.5%, 1.8%, 2.3%, respectively, indicating a rather high risk as well.

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.

```
#markowitz bullet with goog, meta, nflx
n_pf = 5000
set.seed(12345)
w_goog = rnorm(n_pf)
w_meta = rnorm(n_pf)
w_nflx = 1 - w_goog - w_meta
w = rbind(w_goog, w_meta, w_nflx)
rownames(w) = asset.names = c("GOOG", "META", "NFLX")
mu_p=matrix(NA,n_pf)
sd_p=matrix(NA,n_pf)
for (i in 1:n_pf) {
  mu_p[i] = t(w[,i])%*%mu.vec[2:4]
  sd_p[i] = sqrt(t(w[,i])%*%Sigma%*%w[,i])
  i=i+1
}
ymax =1.2*max(mu_p)
ymin =1.2*min(mu_p)
xmax= max(sd_p)
cex.val = 0.5
plot(sd_p, mu_p,ylim=c(ymin, ymax), xlim=c(-0.01, xmax),
ylab=expression(mu[p]), xlab=expression(sigma[p]),
pch=16, col="grey", cex=cex.val, cex.lab=1.5)
points(sd.vec[2:4], mu.vec[2:4], pch=16, cex=1.5, lwd=2, col="blue")
points(c(0), c(rf), pch=16, cex=1.5, lwd=2, col="red")
points(sd_tan, mu_tan, pch=16, cex=1.5, lwd=2, col="green")
text(c(sd.vec[2:4], 0, sd_tan), c(mu.vec[2:4], rf, mu_tan), labels=c(asset.names, "rf", "Tangency"), pos
title("Feasible pfs with GOOG, META and NFLX (no weight restrictions)")
```



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

```
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

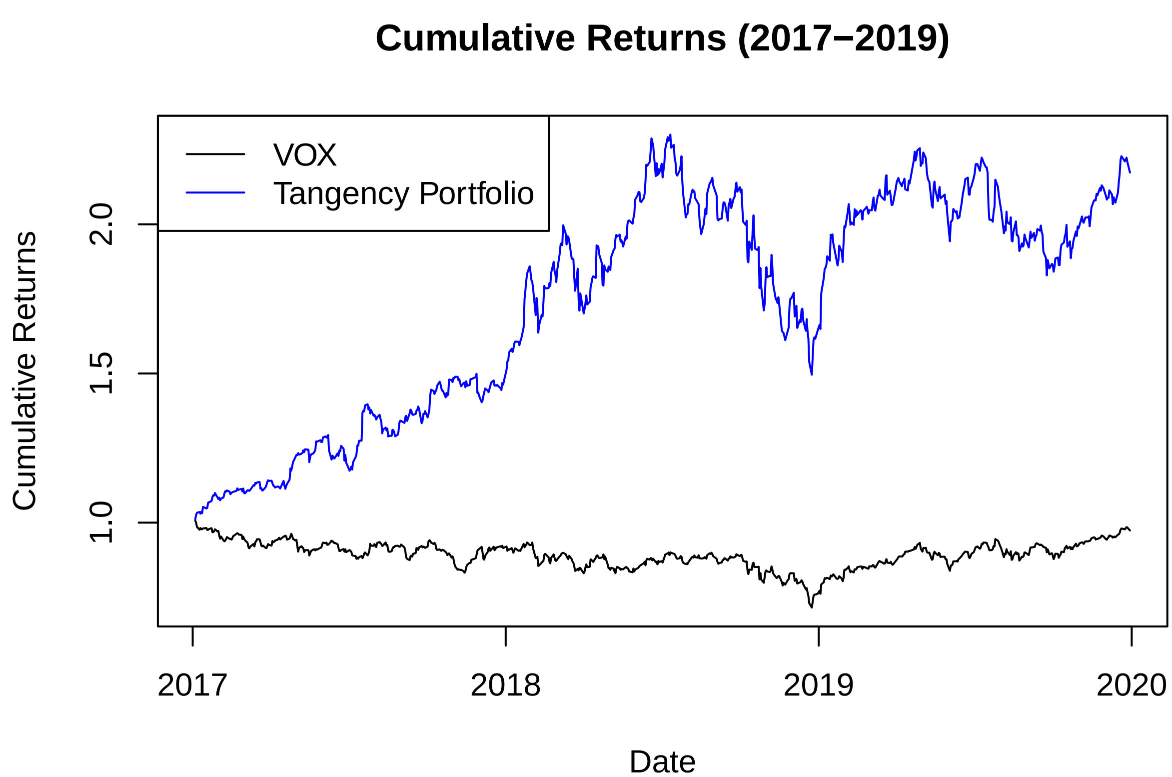
```
tan_returns_2020 <- rowSums(returns_2020[,-1] %*% tan.vec, na.rm = TRUE)

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

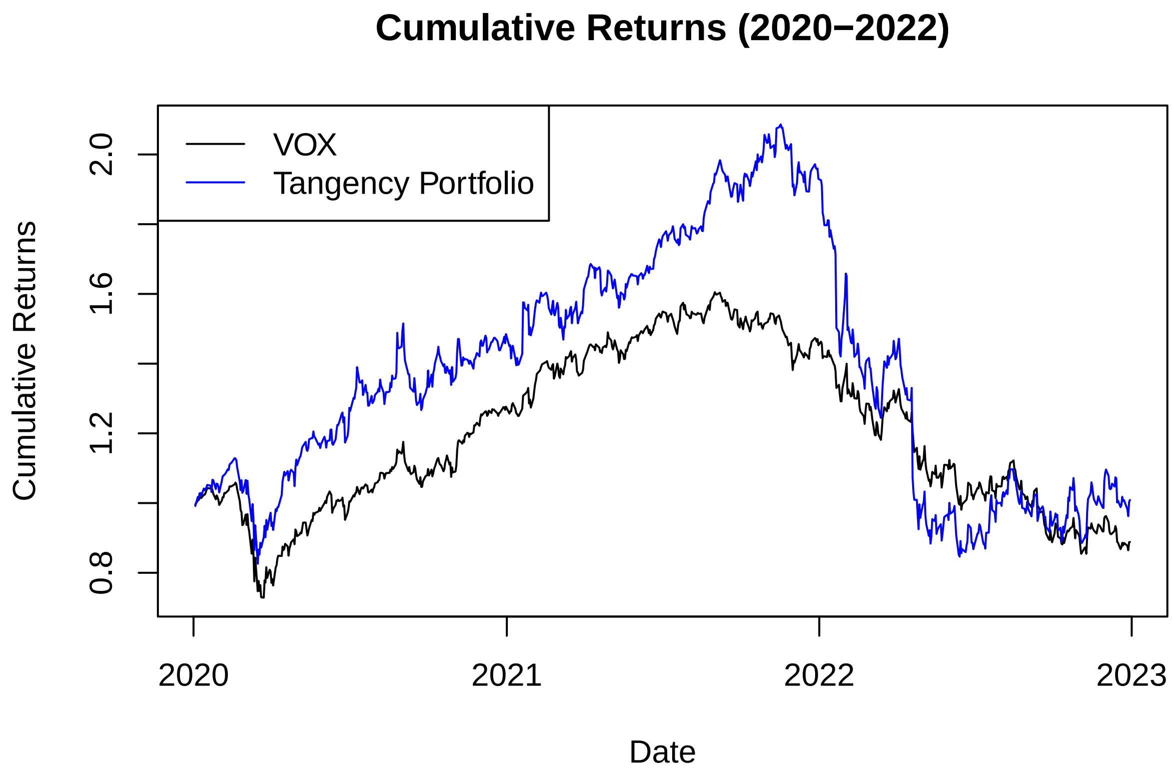
```r
# Convert to time series
VOX_cum_2020 <- xts(VOX_cum_2020, order.by = index(VOX_returns_2020))
tan_cum_2020 <- xts(tan_cum_2020, order.by = index(returns_2020[,-1]))
```

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 (48.6% vs. 4.55%), the black swan affects the tangency portfolio far more.

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

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

ri_goog = dailyReturn(goog$goog.adjusted, type = "arithmetic")
ri_meta = dailyReturn(meta$meta.adjusted, type = "arithmetic")
ri_nflx = dailyReturn(nflx$nflx.adjusted, type = "arithmetic")
ri_vox =  dailyReturn(vox$vox.adjusted, type = "arithmetic")
ri_tang =  tan.vec[1] * ri_goog + tan.vec[2] * ri_meta + tan.vec[3] * ri_nflx # dailyReturn(tang, type
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']
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    0.000 0.000 0.001 0.000    0.000
## Beta     1.333 1.277 1.699 0.946    1.499
## Pr(>|t|) 0.646 0.608 0.341 0.056    0.311
```

Using CAPM, we find that all three stocks do not have a significant alpha, which implies that CAPM holds for them. The tangency portfolio and VOX also have insignificant alphas.

The Betas for GOOG, META and NFLX all 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.

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 = dailyReturn(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
```

```r
names(tbill1mo_new) = tolower(names(tbill1mo_new))
tbill1mo_new = na.omit(tbill1mo_new[,'dgs1mo'])
rf_new = tbill1mo_new/(360*100)
names(rf_new) = c("rf")

ri_goog_new = dailyReturn(goog_new$goog.adjusted, type = "arithmetic")
ri_meta_new = dailyReturn(meta_new$meta.adjusted, type = "arithmetic")
ri_nflx_new = dailyReturn(nflx_new$nflx.adjusted, type = "arithmetic")
ri_vox_new =  dailyReturn(vox_new$vox.adjusted, type = "arithmetic")
ri_tang_new =  tan.vec[1] * ri_goog_new + tan.vec[2] * ri_meta_new + tan.vec[3] * ri_nflx_new #dailyRet
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)

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)
```

```r
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    0.000 -0.001 0.000 0.000    0.000
## Beta     1.090  1.244 0.987 0.984    1.073
## Pr(>|t|) 0.728  0.397 0.963 0.161    0.845
```

Compared to 2017-2019, there are some similarities and some differences. While the Betas for GOOG and META both remain above 1, they decreased, from 1.333 and 1.277 to 1.090 and 1.244 respectively. The Beta for NFLX also changed, falling from 1.699, above 1, to 0.987, less than 1. The Beta for VOX interestingly saw almost no change in beta, moving from 0.946 to 0.948. Like in 2017-2019, the Beta for the tangency portfolio is between the three sub-components. The betas also varied a lot more in 2017-2019. Once again, there was no significant alpha.

Question 7

Estimate and report the daily VaR and ES for the individual companies and the overall industry at 5% levels of significance using the non-parametric method for 2017-2029. Report the estimates in a compact table form. Comment on your findings.

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)
}

goog_var_es = calc_var_es(0.05, dailyReturn(goog))
meta_var_es = calc_var_es(0.05, dailyReturn(meta))
nflx_var_es = calc_var_es(0.05, dailyReturn(nflx))
vox_var_es = calc_var_es(0.05, dailyReturn(vox))
all_var_es = matrix(c(goog_var_es["var"], meta_var_es["var"], nflx_var_es["var"], vox_var_es["var"], go
rownames(all_var_es) = c("VaR", "ES")
colnames(all_var_es) = c("GOOG", "META", "NFLX", "VOX")
all_var_es
```

```
##         GOOG     META     NFLX      VOX
## VaR 24.53818 25.96728 39.34450 16.53932
## ES  35.51662 43.46968 53.03346 23.80752
```

Looking at the VaR and ES table, we see that they are quite high. For example, you can expect to lose at most $39, or 3.9% of your position in NFLX on any day, with a 95% confidence. Also, the VaR and ES are higher for assets we concluded to be riskier (looking at standard deviation). We also see that the VaR and ES are higher for the individual assets than VOX as a whole, showing the effects of lower diversification.

Furthermore, we see that the ES is much higher than the VaR for each asset (and the industry as a whole), indicating that there are rather heavy tails. In particular, the ES for Netflix is significantly higher than the VaR, suggesting that extreme losses are more frequent.

Question 8

Estimate and report the daily VaR and ES for the individual companies and the overall industry at 5% levels of significance using the non-parametric method for 2020-2022. Report the estimates in a compact table form. Compare with the findings for 2017-2019

```
goog_var_es_new = calc_var_es(0.05, dailyReturn(goog_new))
meta_var_es_new = calc_var_es(0.05, dailyReturn(meta_new))
nflx_var_es_new = calc_var_es(0.05, dailyReturn(nflx_new))
vox_var_es_new = calc_var_es(0.05, dailyReturn(vox_new))
all_var_es_new = matrix(c(goog_var_es_new["var"], meta_var_es_new["var"], nflx_var_es_new["var"], vox_va
rownames(all_var_es_new) = c("VaR", "ES")
colnames(all_var_es_new) = c("GOOG", "META", "NFLX", "VOX")
all_var_es_new
```

```
##         GOOG     META     NFLX      VOX
## VaR 35.14762 44.64657 44.71343 28.96217
## ES  49.45134 71.94381 73.68877 41.84081
```

Both the VaR and ES values increased by a notable amount from the 2017-2019 period to the 2020-2022 period for each company. This increase in risk is also reflected in the VOX values, indicating the increase in values is at least partially due to a change affecting the entire industry (not just individual companies). This could be related to the pandemic and a change in the population's media consumption following the pandemic and normal activities resumed. The pandemic triggered massive growth in services like streaming. However, it also caused an increase in competition, leading to a potentially oversaturated market. This also could've contribued to the observed increased in risk.

Facebook's rebranding to Meta in October 2021 and Meta's corresponding direction shift, lead to a highly volatile stock price over the next few years. As a result, of the individual companies, Meta's VaR and ES values increased the most proportionally.

Question 9

Comment on your findings overall, meaning using portfolio optimization, market betas, and the tail risk, and complement with any research/news you found online. Note 2020-2021 were pandemic years.

Overall, each of the three companies we chose outperformed the industry. This was shown through the cumulative returns and the respective betas, as each component stock had a beta near or greater than 1. So, it makes sense that the efficient portfolio combining the three also outperformed the industry, and also had a beta greater than 1. The optimized portfolio also had a higher sharpe ratio than any of its component stocks, and higher than the industry.

The tail risk aspect of the project is illuminating, as it demonstrates the effect of a global market event on risk models. The VaR and ES of all three stocks and VOX all increased in the pandemic, as the risks were far greater than in 2017-2019. The returns and tail risk also accurately show how certain events affected companies. For example, the pandemic was good for Netflix, creating a large growth in subscriber counts. However, the pandemic also brought in more competition for Netflix, and when lockdowns ended, Netflix emerged in a far more competitive landscape. This was reflected in the stock price, and also reflected in the tail risk metrics. The VaR and ES both spiked for Netflix in 2020-22, compared to 2017-19, and the Beta for Netflix also plummeted, as the returns for Netflix in late 2021 and early 2022 were very negative.