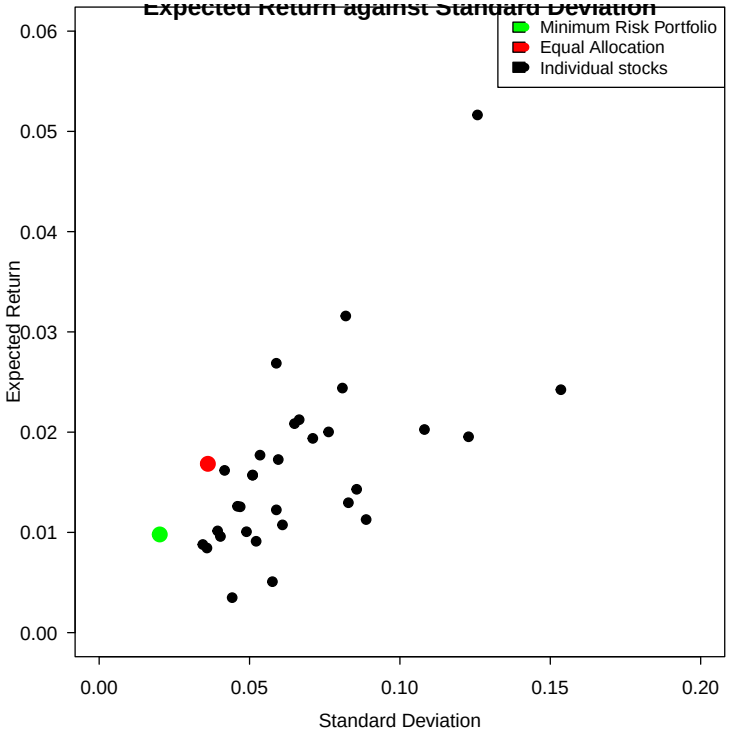
The results below are generated from an R script.

```
#^GSPC,
#AAPL,MSFT,NVDA,TSM,ASML,AVGO,
#GOOGL,META,DIS,TMUS,VZ,CMCSA,
#AMZN,TSLA,HD,BABA,MCD,TM,
#WMT,PG,KO,PEP,COST,FMX,
#BHP,LIN,RIO,VALE,APD,SCCO

a_all <- read.csv("stockData.csv", sep=",", header=TRUE)
#Convert adjusted close prices into returns:
a <- a_all[1:60,] # Use 5 year data to train
r <- (a[-1,3:ncol(a)]-a[-nrow(a),3:ncol(a)])/a[-nrow(a),3:ncol(a)]

#Compute mean vector:
means <- colMeans(r)

#Compute variance covariance matrix
covmat <- cov(r)

#Compute correlation matrix:
cormat <- cor(r)

#Compute the vector of variances:
variances <- diag(covmat)

#Compute the vector of standard deviations:
stdev <- diag(covmat)^.5

#Plot the 31 assets on the space expected return against standard deviation

plot(stdev, means,
     main="Expected Return against Standard Deviation",
     xlab="Standard Deviation",
     ylab="Expected Return",
     xlim = c(0, 0.2),
     ylim = c(0, 0.06),
     pch=19)

#Assume equal allocation portfolio using the 30 stocks.
new_means <- colMeans(r[,-1])
new_covmat <- cov(r[,-1])
new_cormat <- cor(r[,-1])
new_variances <- diag(new_covmat)
new_stdev <- diag(new_covmat)^.5

number_of_stocks = 30
ones_vector <- rep(1, number_of_stocks)
equal_weight_vector <- ones_vector/number_of_stocks

equal_varp <- t(equal_weight_vector) %*% new_covmat %*% equal_weight_vector
equal_sdp <- sqrt(equal_varp)

equal_Rp <- t(equal_weight_vector) %*% new_means
```

```r
points(equal_sdp, equal_Rp, pch = 19, lwd=5, col="red")

#Assume minimum risk portfolio

ones_vector <- rep(1, number_of_stocks)
inverse_new_covmat <- solve(new_covmat)
min_risk_weight_vector <- inverse_new_covmat  %*% ones_vector / as.numeric(t(ones_vector)  %*% inverse_n

min_risk_varp <- t(min_risk_weight_vector) %*% new_covmat %*% min_risk_weight_vector
min_risk_sdp <- sqrt(min_risk_varp)
min_risk_Rp <- t(min_risk_weight_vector) %*% new_means

points(min_risk_sdp, min_risk_Rp, pch=19, lwd=5, col="green")

legend("topright", legend=c("Minimum Risk Portfolio", "Equal Allocation", "Individual stocks"),
       col=c("green", "red","black"),pch = 19,fill =c("green", "red","black"),cex=0.8)
```



The R session information (including the OS info, R version and all packages used):

```r
sessionInfo()

## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8  LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8 LC_NUMERIC=C
```

```
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] quantmod_0.4.21 TTR_0.24.3      xts_0.13.0      zoo_1.8-11      pdfetch_0.2.8
## [6] shiny_1.7.4
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.10       highr_0.10        jquerylib_0.1.4   bslib_0.4.2
##  [5] pillar_1.9.0      compiler_4.2.2    later_1.3.0       tools_4.2.2
##  [9] digest_0.6.31     evaluate_0.20     memoise_2.0.1     jsonlite_1.8.4
## [13] lubridate_1.9.2   lifecycle_1.0.3   tibble_3.2.1      timechange_0.2.0
## [17] lattice_0.20-45   pkgconfig_2.0.3   rlang_1.1.0       cli_3.6.0
## [21] yaml_2.3.7        curl_5.0.0        xfun_0.37         fastmap_1.1.1
## [25] knitr_1.42        dplyr_1.1.1       httr_1.4.5        sass_0.4.5
## [29] generics_0.1.3    vctrs_0.6.1       grid_4.2.2        tidyselect_1.2.0
## [33] fontawesome_0.5.0 glue_1.6.2        R6_2.5.1          fansi_1.0.4
## [37] XML_3.99-0.14     rmarkdown_2.20    tidyr_1.3.0       purrr_1.0.1
## [41] magrittr_2.0.3    promises_1.2.0.1  ellipsis_0.3.2    htmltools_0.5.4
## [45] mime_0.12         xtable_1.8-4      httpuv_1.6.9      utf8_1.2.3
## [49] cachem_1.0.7

Sys.time()

## [1] "2023-04-08 13:53:09 PDT"
```