U.S. Stocks vs. Cryptocurrencies

RiskAndReturn–Finance

Cheryl Isabella

Table of Contents

# **Overview**

This project aims to evaluate the Risk and Return for U.S. stocks and cryptocurrencies.

# **Data**

The excel file “datacase1.xls” contains daily prices (end of the day) for a variety of US stocks and stock indices. The stock price data are end of the day closing prices. Daily prices of some cryptocurrencies are included in a second data file (Datacase2.xls) and started to trade much later (source: finance:yahoo:com). Assume a risk-free rate (annualized) equal to 3%. The data file contains the following series:

* US stock indices: Dow, S&P500, NASDAQ, Russel 2000.
* Individual US stocks: General Electric, Bank of America, Coca Cola, Intel, Apple inc, Mc Donalds Corp, Procter and Gamble, American Airlines, Caterpillar, Wallmart.4
* Cryptocurrencies: Bitcoin, Ethereum, Ripple.

## Importing and Cleaning the Data

## # A tibble: 84,304 × 3  
## date name price  
## <dttm> <chr> <dbl>  
## 1 1995-01-03 00:00:00 SP\_500 459.   
## 2 1995-01-03 00:00:00 Russel\_2000 247.   
## 3 1995-01-03 00:00:00 nasdaq 744.   
## 4 1995-01-03 00:00:00 Dow 3838.   
## 5 1995-01-03 00:00:00 General\_electric 8.5   
## 6 1995-01-03 00:00:00 Coke 26   
## 7 1995-01-03 00:00:00 Intel 3.98  
## 8 1995-01-03 00:00:00 Apple 1.37  
## 9 1995-01-03 00:00:00 Mc\_Donalds 14.6   
## 10 1995-01-03 00:00:00 ProcterGamble 15.6   
## # … with 84,294 more rows

## a) Creating new Columns

#creating the daily returns  
df\_sc\_return <- df\_sc %>%  
 group\_by(name) %>%  
 mutate(daily\_return = log(lead(price)/price)) %>%  
 na.omit   
  
#creating other variables per stock  
measures <- df\_sc\_return %>%  
 group\_by(name) %>%  
 summarise(avg\_annual\_return = mean(daily\_return) \* 252,  
 annual\_risk = sd(daily\_return) \* sqrt(252),  
 min\_return = min(daily\_return),  
 max\_return = max(daily\_return),  
 sharpe\_ratio = (avg\_annual\_return-0.03)/annual\_risk) %>%  
 mutate(type = ifelse(name %in% c("bitcoin", "etherium", "ripple"),"crypto", ifelse(name %in% c("Russel\_2000", "SP\_500", "nasdaq", "Dow"), "index" , "stock")))  
  
measures

## # A tibble: 17 × 7  
## name avg\_annual\_retu… annual\_risk min\_return max\_return sharpe\_ratio type   
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>  
## 1 American\_Airlines 0.0422 0.443 -0.342 0.302 0.0275 stock  
## 2 Apple 0.208 0.460 -0.731 0.287 0.388 stock  
## 3 Bank\_of\_America 0.0719 0.684 -0.362 0.462 0.0612 stock  
## 4 bitcoin 1.06 1.12 -0.849 1.47 0.916 cryp…  
## 5 Caterpillar 0.102 0.327 -0.157 0.137 0.221 stock  
## 6 Coke 0.0820 0.294 -0.162 0.158 0.177 stock  
## 7 Dow 0.0796 0.176 -0.0820 0.105 0.281 index  
## 8 etherium 1.28 1.31 -0.916 0.383 0.956 cryp…  
## 9 General\_electric 0.0184 0.293 -0.137 0.180 -0.0394 stock  
## 10 Intel 0.110 0.382 -0.249 0.183 0.209 stock  
## 11 Mc\_Donalds 0.104 0.240 -0.137 0.103 0.308 stock  
## 12 nasdaq 0.0970 0.245 -0.102 0.133 0.274 index  
## 13 ProcterGamble 0.0695 0.227 -0.360 0.0973 0.174 stock  
## 14 ripple 0.765 1.91 -0.997 1.03 0.385 cryp…  
## 15 Russel\_2000 0.0788 0.223 -0.126 0.0886 0.219 index  
## 16 SP\_500 0.0755 0.186 -0.0947 0.110 0.244 index  
## 17 Walmart 0.0909 0.260 -0.107 0.105 0.235 stock

# **Risk and Return**

# b) Riskiness

measures %>%  
 group\_by(type) %>%  
 summarise(Avg\_Annual\_Risk = mean(annual\_risk)) %>%  
 arrange(desc(Avg\_Annual\_Risk))

## # A tibble: 3 × 2  
## type Avg\_Annual\_Risk  
## <chr> <dbl>  
## 1 crypto 1.45   
## 2 stock 0.361  
## 3 index 0.207

**The relative riskiness of the stocks, stock indices and the cryptos based on the annualized standard deviation:**

* We can evaluate the riskiness of an investment based on its standard deviation, also known as volatility. A higher volatility means that the return deviates more from the average return (distribution is more spread out). Hence, the investment is riskier.
* The average annual volatility of the cryptocurrencies in our data is about 145%. Since this is the highest value, investments in cryptocurrencies are a lot riskier than investments in stocks or indices. Individual stocks have an average annual volatility of 36% and indices a volatility of 21%. Since indices have the lowest average volatility, they bear the least risk. Very interesting to observe, is the high difference (of more than 100%) between the riskiness of stocks/indices and cryptocurrencies.

# c) Sharpe Ratios

measures %>%  
 group\_by(type) %>%  
 summarise(Avg\_Sharpe\_Ratio = mean(sharpe\_ratio)) %>%  
 arrange(desc(Avg\_Sharpe\_Ratio))

## # A tibble: 3 × 2  
## type Avg\_Sharpe\_Ratio  
## <chr> <dbl>  
## 1 crypto 0.752  
## 2 index 0.255  
## 3 stock 0.176

**Comparison of the Sharpe ratios of the portfolios vs. the Sharpe ratios of the individual US stocks, and the Sharpe ratios of the cryptos relative to the other investments:**

* The Sharpe ratio measures the ratio of reward to volatility and is used to evaluate the return of an investment compared to its risk. The higher the ratio, the higher the reward per unit of volatility.
* The portfolios/indices have, in comparison, a higher Sharpe ratio than the individual US stocks. Hence, the portfolios would be the optimal investment to combine with the risk-free investment.
* The average Sharpe ratio of the cryptocurrencies is a lot higher than the average ratio of the indices and stocks. As we have seen before, investments in cryptocurrencies are riskier. However, Investors are compensated for taking on this additional risk. Still, the Sharpe ratio of cryptocurrencies indicates that the risk premium per unit of volatility is higher than for stocks and indices.

# d) Investment Decision

measures %>%  
 select(type, name, sharpe\_ratio) %>%  
 arrange(desc(sharpe\_ratio))

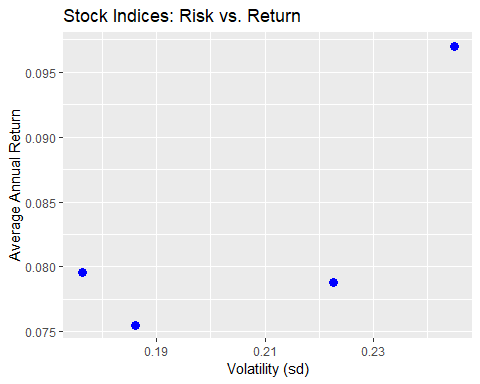
## # A tibble: 17 × 3  
## type name sharpe\_ratio  
## <chr> <chr> <dbl>  
## 1 crypto etherium 0.956   
## 2 crypto bitcoin 0.916   
## 3 stock Apple 0.388   
## 4 crypto ripple 0.385   
## 5 stock Mc\_Donalds 0.308   
## 6 index Dow 0.281   
## 7 index nasdaq 0.274   
## 8 index SP\_500 0.244   
## 9 stock Walmart 0.235   
## 10 stock Caterpillar 0.221   
## 11 index Russel\_2000 0.219   
## 12 stock Intel 0.209   
## 13 stock Coke 0.177   
## 14 stock ProcterGamble 0.174   
## 15 stock Bank\_of\_America 0.0612  
## 16 stock American\_Airlines 0.0275  
## 17 stock General\_electric -0.0394

**Result:**

* We would invest in the cryptocurrency Ethereum since it has the highest Sharpe ratio. This means this investment would give the highest return per unit of volatility.

# e) Stock Indices

#filtering for stock indices  
indices <- filter(measures, type == "index")  
  
#plot  
ggplot(indices, aes(annual\_risk, avg\_annual\_return)) +  
 geom\_point(size = 3, color = "blue") +  
 labs(title = "Stock Indices: Risk vs. Return", x = "Volatility (sd)", y = "Average Annual Return")

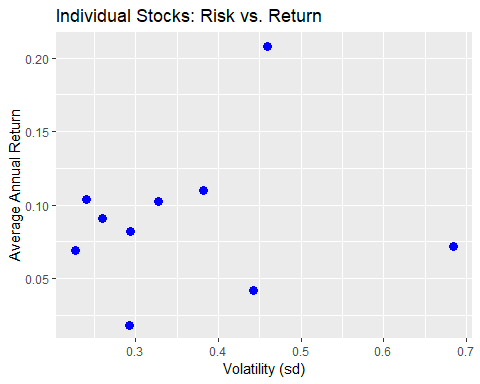


**Results:**

* An investment with higher volatility must offer a higher return to compensate investors for the additional risk. The excess return gives the risk premium investors get.
* On average, this corresponds with our data. The higher volatility/risk, the higher the average annual return. The only exception is the index SP 500. Its volatility is slightly higher than the volatility of the Dow index; still, the average return is lower.

# f) Individual Stocks

#filtering for individual stocks  
stocks <- filter(measures, type == "stock")  
  
#plot  
ggplot(stocks, aes(annual\_risk, avg\_annual\_return)) +  
 geom\_point(size = 3, color = "blue") +  
 labs(title = "Individual Stocks: Risk vs. Return", x = "Volatility (sd)", y = "Average Annual Return")



**Results:**

* The assumption that a more volatile investment should have a higher return does not hold for the individual stocks. The plot shows no clear relationship between volatility and return.
* We can also observe that the volatility for individual stocks is higher than for the indices. In general, individual stocks are typically more volatile than indices/portfolios. Furthermore, larger stocks have lower volatility overall.

# g) 2-Stock Portfolio

* Caterpillar (A)
* Walmart (B)

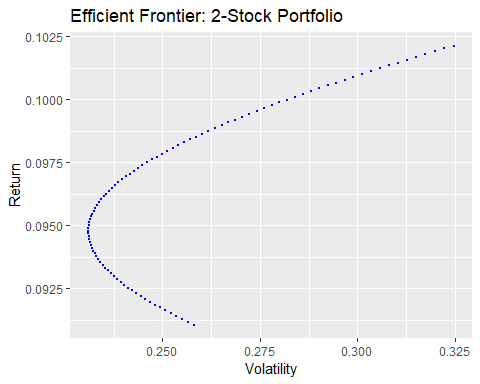
#weights  
A\_w <- c(rep(1:99)/100)  
B\_w <- c(rep(99:1)/100)  
weights <- as.data.frame(cbind(A\_w, B\_w))  
  
#Return and Risk of both stocks  
A\_return <- as.numeric(measures %>% filter(name == "Caterpillar") %>% select(avg\_annual\_return))  
B\_return <- as.numeric(measures %>% filter(name == "Walmart") %>% select(avg\_annual\_return))  
A\_risk <- as.numeric(measures %>% filter(name == "Caterpillar") %>% select(annual\_risk))  
B\_risk <- as.numeric(measures %>% filter(name == "Walmart") %>% select(annual\_risk))  
  
#correlation between returns  
A\_r <- df\_sc\_return %>% filter(name == "Caterpillar")  
B\_r <- df\_sc\_return %>% filter(name == "Walmart")   
A\_B\_r <- cbind(A\_r[,4], B\_r[,4])  
  
cor(A\_B\_r)

## daily\_return daily\_return  
## daily\_return 1.0000000 0.3052991  
## daily\_return 0.3052991 1.0000000

#dataframe  
portfolio <- mutate(weights, Return = A\_w\*A\_return + B\_w\*B\_return,  
 Volatility = sqrt(A\_w^2\*A\_risk^2 + B\_w^2\*B\_risk^2 + 2\*A\_w\*B\_w\*0.3052991\*A\_risk\*B\_risk))  
portfolio

## A\_w B\_w Return Volatility  
## 1 0.01 0.99 0.09106016 0.2580354  
## 2 0.02 0.98 0.09117313 0.2564951  
## 3 0.03 0.97 0.09128611 0.2549936  
## 4 0.04 0.96 0.09139908 0.2535316  
## 5 0.05 0.95 0.09151205 0.2521097  
## 6 0.06 0.94 0.09162503 0.2507286  
## 7 0.07 0.93 0.09173800 0.2493891  
## 8 0.08 0.92 0.09185098 0.2480917  
## 9 0.09 0.91 0.09196395 0.2468372  
## 10 0.10 0.90 0.09207692 0.2456262  
## 11 0.11 0.89 0.09218990 0.2444593  
## 12 0.12 0.88 0.09230287 0.2433373  
## 13 0.13 0.87 0.09241585 0.2422606  
## 14 0.14 0.86 0.09252882 0.2412300  
## 15 0.15 0.85 0.09264180 0.2402460  
## 16 0.16 0.84 0.09275477 0.2393091  
## 17 0.17 0.83 0.09286774 0.2384200  
## 18 0.18 0.82 0.09298072 0.2375792  
## 19 0.19 0.81 0.09309369 0.2367872  
## 20 0.20 0.80 0.09320667 0.2360445  
## 21 0.21 0.79 0.09331964 0.2353515  
## 22 0.22 0.78 0.09343262 0.2347087  
## 23 0.23 0.77 0.09354559 0.2341165  
## 24 0.24 0.76 0.09365856 0.2335753  
## 25 0.25 0.75 0.09377154 0.2330855  
## 26 0.26 0.74 0.09388451 0.2326473  
## 27 0.27 0.73 0.09399749 0.2322611  
## 28 0.28 0.72 0.09411046 0.2319271  
## 29 0.29 0.71 0.09422344 0.2316455  
## 30 0.30 0.70 0.09433641 0.2314166  
## 31 0.31 0.69 0.09444938 0.2312405  
## 32 0.32 0.68 0.09456236 0.2311173  
## 33 0.33 0.67 0.09467533 0.2310471  
## 34 0.34 0.66 0.09478831 0.2310299  
## 35 0.35 0.65 0.09490128 0.2310658  
## 36 0.36 0.64 0.09501425 0.2311548  
## 37 0.37 0.63 0.09512723 0.2312967  
## 38 0.38 0.62 0.09524020 0.2314915  
## 39 0.39 0.61 0.09535318 0.2317390  
## 40 0.40 0.60 0.09546615 0.2320391  
## 41 0.41 0.59 0.09557913 0.2323916  
## 42 0.42 0.58 0.09569210 0.2327963  
## 43 0.43 0.57 0.09580507 0.2332527  
## 44 0.44 0.56 0.09591805 0.2337608  
## 45 0.45 0.55 0.09603102 0.2343200  
## 46 0.46 0.54 0.09614400 0.2349302  
## 47 0.47 0.53 0.09625697 0.2355907  
## 48 0.48 0.52 0.09636995 0.2363014  
## 49 0.49 0.51 0.09648292 0.2370616  
## 50 0.50 0.50 0.09659589 0.2378709  
## 51 0.51 0.49 0.09670887 0.2387288  
## 52 0.52 0.48 0.09682184 0.2396348  
## 53 0.53 0.47 0.09693482 0.2405884  
## 54 0.54 0.46 0.09704779 0.2415890  
## 55 0.55 0.45 0.09716077 0.2426360  
## 56 0.56 0.44 0.09727374 0.2437287  
## 57 0.57 0.43 0.09738671 0.2448667  
## 58 0.58 0.42 0.09749969 0.2460492  
## 59 0.59 0.41 0.09761266 0.2472757  
## 60 0.60 0.40 0.09772564 0.2485454  
## 61 0.61 0.39 0.09783861 0.2498578  
## 62 0.62 0.38 0.09795158 0.2512121  
## 63 0.63 0.37 0.09806456 0.2526076  
## 64 0.64 0.36 0.09817753 0.2540438  
## 65 0.65 0.35 0.09829051 0.2555199  
## 66 0.66 0.34 0.09840348 0.2570352  
## 67 0.67 0.33 0.09851646 0.2585890  
## 68 0.68 0.32 0.09862943 0.2601807  
## 69 0.69 0.31 0.09874240 0.2618095  
## 70 0.70 0.30 0.09885538 0.2634748  
## 71 0.71 0.29 0.09896835 0.2651758  
## 72 0.72 0.28 0.09908133 0.2669120  
## 73 0.73 0.27 0.09919430 0.2686826  
## 74 0.74 0.26 0.09930728 0.2704869  
## 75 0.75 0.25 0.09942025 0.2723242  
## 76 0.76 0.24 0.09953322 0.2741940  
## 77 0.77 0.23 0.09964620 0.2760955  
## 78 0.78 0.22 0.09975917 0.2780281  
## 79 0.79 0.21 0.09987215 0.2799911  
## 80 0.80 0.20 0.09998512 0.2819840  
## 81 0.81 0.19 0.10009810 0.2840060  
## 82 0.82 0.18 0.10021107 0.2860566  
## 83 0.83 0.17 0.10032404 0.2881352  
## 84 0.84 0.16 0.10043702 0.2902411  
## 85 0.85 0.15 0.10054999 0.2923738  
## 86 0.86 0.14 0.10066297 0.2945326  
## 87 0.87 0.13 0.10077594 0.2967171  
## 88 0.88 0.12 0.10088891 0.2989266  
## 89 0.89 0.11 0.10100189 0.3011606  
## 90 0.90 0.10 0.10111486 0.3034186  
## 91 0.91 0.09 0.10122784 0.3057000  
## 92 0.92 0.08 0.10134081 0.3080043  
## 93 0.93 0.07 0.10145379 0.3103310  
## 94 0.94 0.06 0.10156676 0.3126796  
## 95 0.95 0.05 0.10167973 0.3150496  
## 96 0.96 0.04 0.10179271 0.3174405  
## 97 0.97 0.03 0.10190568 0.3198519  
## 98 0.98 0.02 0.10201866 0.3222833  
## 99 0.99 0.01 0.10213163 0.3247342

#plot  
ggplot(portfolio, aes(Volatility, Return)) +  
 geom\_point(size = 0.5, color = "blue") +  
 ggtitle("Efficient Frontier: 2-Stock Portfolio")



**Results:**

* The efficiency frontier is the set of optimal portfolios with the highest possible return for a given level of volatility. Portfolios below the efficiency frontier are called inefficient portfolios because there is another portfolio that is better in terms of return and volatility. The efficient portfolio is also the portfolio with the highest Sharpe ratio.
* To answer the question, the portfolio with the lowest risk, which is still an efficient frontier, is the one furthest on the left on the graph: 35% Caterpillar and 65% Walmart

# h) Portfolios

Correlation matrix between returns:

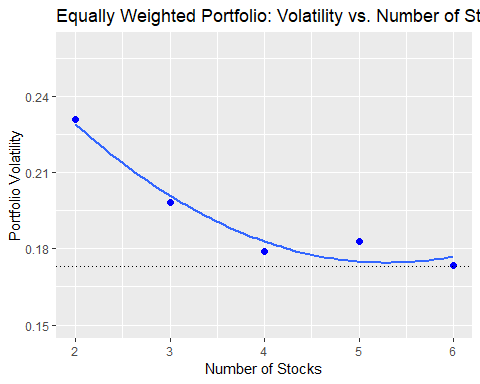
## General\_electric Walmart Coke ProcterGamble Caterpillar  
## General\_electric 1.0000000 0.3992157 0.1996721 0.3632182 0.4859691  
## Walmart 0.3992157 1.0000000 0.1267063 0.3292443 0.3052991  
## Coke 0.1996721 0.1267063 1.0000000 0.1357331 0.1931332  
## ProcterGamble 0.3632182 0.3292443 0.1357331 1.0000000 0.2851554  
## Caterpillar 0.4859691 0.3052991 0.1931332 0.2851554 1.0000000  
## Mc\_Donalds 0.3506315 0.3194159 0.1416126 0.3346362 0.2981023  
## Mc\_Donalds  
## General\_electric 0.3506315  
## Walmart 0.3194159  
## Coke 0.1416126  
## ProcterGamble 0.3346362  
## Caterpillar 0.2981023  
## Mc\_Donalds 1.0000000

Dataframe of portfolio metrics:

## n volatility correlation portfolio\_volatility  
## 1 2 0.2763231 0.3992157 0.2311237  
## 2 3 0.2821342 0.2418647 0.1984141  
## 3 4 0.2684255 0.2589649 0.1789059  
## 4 5 0.2801813 0.2823347 0.1828422  
## 5 6 0.2735053 0.2835459 0.1736177

Plot of Equally Weighted Portfolio: Volatility vs. Number of Stocks

#plot  
ggplot(portfolios, aes(n, portfolio\_volatility)) +  
 geom\_point(size = 2, color = "blue") +  
 stat\_smooth(method = "lm", formula = y ~ x + I(x^2), se = F) +  
 geom\_hline(yintercept = 0.173, linetype = "dotted") +  
 labs(title = "Equally Weighted Portfolio: Volatility vs. Number of Stocks", x = "Number of Stocks") +  
 scale\_y\_continuous("Portfolio Volatility", limits = c(0.15,0.26))



**Results:**

* The graph shows a relationship between the volatility of the portfolios and the number of stocks. The more stocks are added to the portfolio, the lower the volatility gets. Furthermore, the relationship flattens out. The effect of adding one stock when there are two stocks in the portfolio is higher than when there are already five stocks in the portfolio. By adding more stocks to the portfolio, we eliminate the diversifiable risk. At one point, the volatility cannot get lower since the correlated market risk/common risk cannot be diversified.
* The curvature in the graph is due to the correlation between the stocks. The first portfolio contains the stocks of General Electric and Walmart. The correlation matrix shows that they have a relatively high correlation coefficient of 0.4, which gives the correlated market risk. The correlated market risk is displayed by the area underneath the curve, which cannot be eliminated.