Bayesian Parameter Estimation for Etoro Copytrader Performance Evaluation

September 17, 2021
JeanLuc Oudshoorn
jeanluc@oudshoorn.pro

1. Project Motivation

Trading assets is hot. The past year has seen a tremendous influx of retail investors entering the market.[1] There have been several factors driving this trend. Among these are excess savings that consumers have not been able to spend on services during coronavirus lockdowns, large-scale economic stimulus programs, and a society-wide move towards increased digitization. These new investors have channeled an incredible amount of funds into a finite number of assets; broadly pushing up most major market indices for what looks to be a record breaking year. Arguably, this is in turn attracting even more potential investors.

Etoro is among the most used trading platforms used by retail investors across Europe and North America. Because of its status as a major recognized platform it has benefited greatly from the inflow of new users. Etoro's claim to fame is its unique copytrader function, which allows users to copy the actions of an active fund manager against zero additional fees. If the fund manager purchases an asset, it will be purchased

---

[1] Joshua Oliver, "US retail investors drive summer surge in stocks," *Financial Times,* 6 September, 2021, https://www.ft.com/content/d87c6631-55f0-4897-9634-bf0ad969e27d.

simultaneously on the account of the copier. The fund managers are commonly referred to as 'copytraders', even though they are not the ones doing the copying.

For transparency, every copytrader has a stats page detailing their monthly performance, portfolio holdings and risk score. This creates the opportunity to **analyze popular copytraders for their outperformance versus the market. This information can be used to decide which traders are worth copying.** For retail investors who may not be as financially literate as they would like, a model estimating the true return and volatility parameters of popular traders based on observed **data could earn them thousands of additional dollars over their lifetime.** On top of that, such a model could help prevent unbearable losses by helping investors make a copying decision tha is more in line with their personal risk tolerance.

The aim of this project is to estimate probability density functions for a grid of returns values for a select group of traders. Besides, bayesian parameter estimation will be used to test for consistent outperformance comapred to major market indices. (S&P 500, Nasdaq Composite)

2. The Dataset

To complete this project data has to be assembled from several different sources. The statistics on the stat pages of each copytrader will have to be copied and placed into a dataframe. Unfortunately, the Etoro server does not allow scraping these elements and returns a 403 'request denied' –error. Etoro does have an API but does not ordinarily allow non-developers to make use of it. Thus the data had to be gathered and assembled manually. An example of this can be seen on screen. The full excel sheet can be found

	jeppe	mariano	victor	jurgen	reinhardt	martina	wesley
aug-21	0.54	2.28	-0.3	1.46	0.4	2.08	2.16
jul-21	-0.94	-0.13	7.67	-3.55	-0.68	-2.45	-5.84
jun-21	1.85	9.43	0.13	3.17	11.9	8.92	3.96
mei-21	2.5	-1.86	2.65	-0.94	-3.86	-1.99	-2.03
apr-21	0.5	6.76	-0.84	3.44	5.66	3.57	-0.12
mrt-21	2.28	-2.85	3.94	-4.65	-8.97	-3.16	0.38
feb-21	3.38	1.11	2.62	-0.07	3.23	-2.88	6.58
jan-21	4.13	2.95	0.72	3.44	1.63	0.56	1.65
dec-20	4.46	6.72	4.24	4.84	5.48	5.84	9.15
nov-20	19.72	15.53	-0.3	0.12	18.03	10.97	16.03

Example of trader returns: the values represent percentual returns per month

here. What is not obvious here is that each copytrader started using the Etoro platform at a different point in time. Because of this, some traders have as much as 99 monthly return data points while others have only 27. This may lead much larger uncertainty bands around the population parameter estimates of some copytraders. Nevertheless, median imputing does not suffice in this case because it greatly reduces variability in the data, suggesting that returns for certain copytraders are very stable. This is considered an exceptionally good trait in investment management.

The indices monthly return data can either be scraped online or downloaded directly using the quantmod package. In this case downloading it through quantmod from yahoo finance turned out to be the better option as not all monthly return data was easily available online for the full period of the data. After indexing properly, the data was converted to an XTS time-series object. The benefit of this is that the dependency of the observations is preserved; for each trading month there is both an observation from the copytraders as well as the market indices used.

```
library(quantmod)
library(xts)
library(readxl)

nasdaq <- getSymbols(Symbols = "^IXIC", src = "yahoo", auto.assign = FALSE)
world <- getSymbols(Symbols = "URTH", src = "yahoo", auto.assign = FALSE)
sp <- getSymbols(Symbols = "^GSPC", src = "yahoo", auto.assign = FALSE)

nasdaq <- to.period(nasdaq, period = "months", k = 1, OHLC = FALSE)
world <- to.period(world, period = "months", k = 1, OHLC = FALSE)
sp <- to.period(sp, period = "months", k = 1, OHLC = FALSE)

nasdaq <- nasdaq["2013-07/2021-08"]
world <- world["2013-07/2021-08"]
sp <- sp["2013-07/2021-08"]

idx <- index(nasdaq)

etorocopy <- read_excel("Etoro.xlsx", col_names = TRUE)
etorocopy <- xts(etorocopy, order.by = idx)

nasdaq$nasdaq <- monthlyReturn(nasdaq)*100
world$world <- monthlyReturn(world)*100
sp$sp <- monthlyReturn(sp)*100

etorocopy <- merge(etorocopy, nasdaq$nasdaq, world$world, sp$sp, join =
"left")
etorocopy <- round(etorocopy[,-1], 2)
```

R-script for reading the Copytrader returns into R and merging them with downloaded market index data using quantmod.

3. Analysis Plan

To determine copytrader performance a number of steps will have to be followed. We already have access to historic data which we assume contains information on how well any given copytrader will perform in the future. To model this, the historic data is treated as a sample which is used to find the unknown population parameters. In this case, the unknown population parameters are the mean monthly return and volatility (standard deviation) given unlimited time. With each additional observation (i.e. new month of data available) the parameters of the historic data will converge closer to the true parameters.

Using an expectation-maximization algorithm, a T-distribution can be fitted to the data available for each copytrader. This fit provides both a point estimate as well as confidence intervals. Using the confidence intervals to model our uncertainty over the parameter estimates for the T-distribution, one can generate a large sample to represent the steady-state returns over many iterations. This sample can then be used to calculate quantiles and plot a probability density function over the return parameter for each copytrader.

While instructive, the problem with this density function is that it assumes market conditions are unchanging. Thus, the starting date of each copytrader should not matter. In reality this is not true because market conditions change constantly and it is easier to perform better in favorable market conditions. To provide a fair estimate, bayesian parameter estimation can be deployed using a markov chain monte carlo sampling technique. When estimating both the copytrader parameters, as well as the market index parameters, the two can be subtracted to create a density function that details the probability of each difference in mean monthly returns.

Based on the original density function of the plain return data, and the bayesian estimation of market outperformance, copytraders can be evaluated. To do so, copytraders will be ranked across several categories. Then, the rank sum average can be taken to determine the overall best trader.

4. Performance Metrics

If the return data for the copytraders is close to normality, both the EM algorithm for finding a T-distribution fit and the MCMC algorithm for bayesian parameter estimation should converge. If this does not happen, it becomes impossible to generate a sample and make inferences about the data. If convergence takes place a number of other factors have to be taken into account:

Residual Standard Error:

The EM algorithm finds the parameters for the T-fit with the highest likelihood. This is however only a point estimate. In reality there is a range of uncertainty around this value which is denoted in the standard error for each parameter. The smaller the standard errors, the more sure one can be that the point estimate is close to being correct.

While the residual standard error tells us how certain the model is of the point estimate, there are no actual metrics to evaluate model performance. The reason for this is that this project conerns neither a regression nor classification problem. Instead, this is simply a matter of statistical inference using both frequentist and bayesian methods. As long as the appropriate assumptions are met, methods for frequentist and bayesian inference can safely be applied.

The performance of the copytraders can be evaluated through several metrics:

Quantiles:

Quantiles provide clarity on the probability density function of each copytrader. One can for example estimate what the average expected loss would be for the 1% worst trading months.

AUC for X > 0:

Using the 'area under the curve' of the probability density function for each trader we can calculate the percentage of months that we expect returns to be positive.

Mean market outperformance:

This metric measures the expected outperformance of an individual copytrader to the market index over the long run. It comes with a band of uncertainty which can be translated to a credible interval.

Market correlation:

If a copytraders returns are not highly correlated with market indices this means that returns should be more stable when market conditions are volatile. In investment management lack of correlation between assets in a portfolio is considered good as it improves diversification. The same principle applies to copying multiple copytraders with low correlation to each other.

5. Exploratory Data Analysis

The purpose of the exploratory data analysis is to check if there are any problems with with the dataset that need to be solved before the next steps can proceed. As mentioned before, there are a large number of NA's because of the different time periods in which copytraders were active.

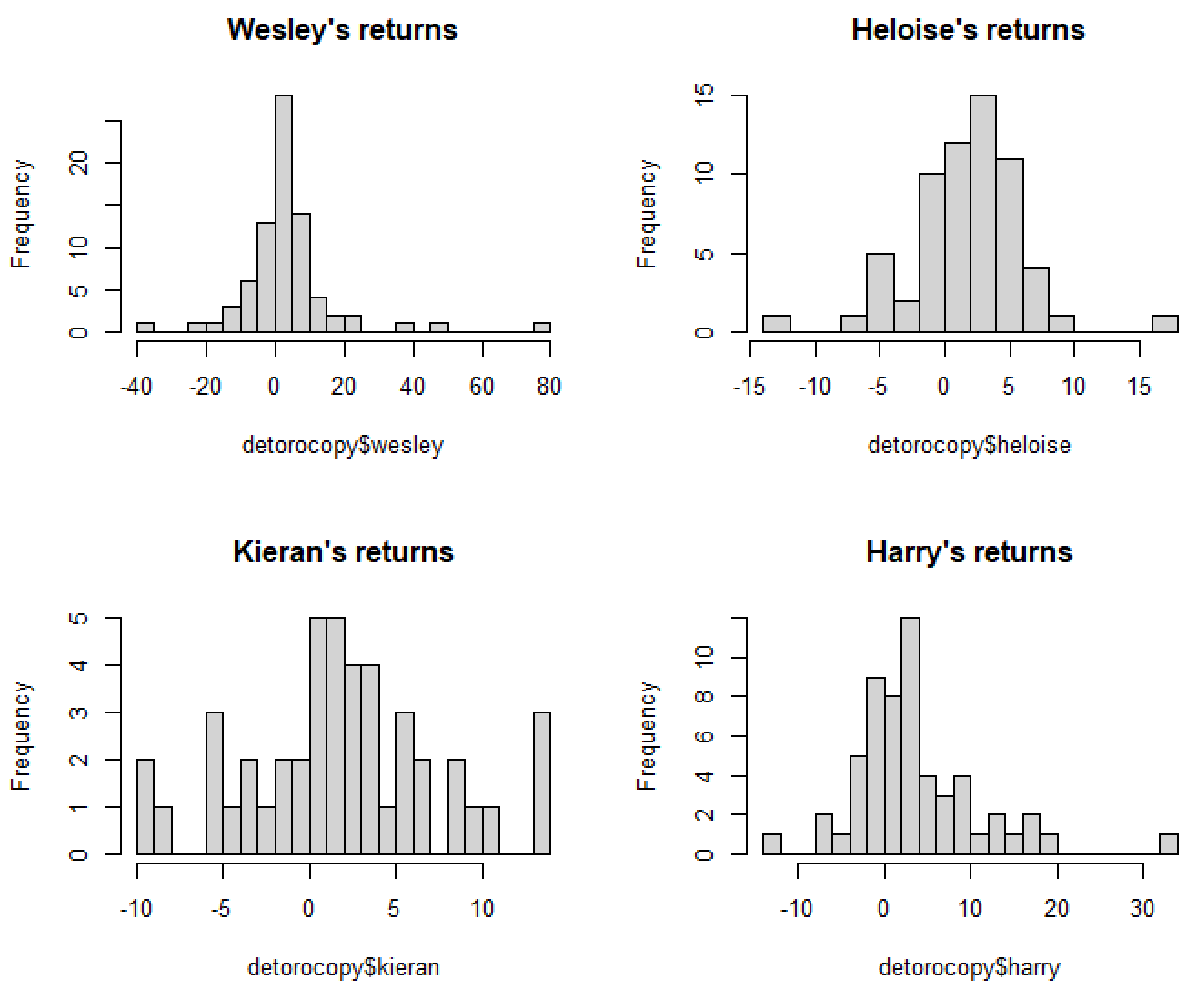
```
> sum(is.na(detorocopy))
[1] 555
```

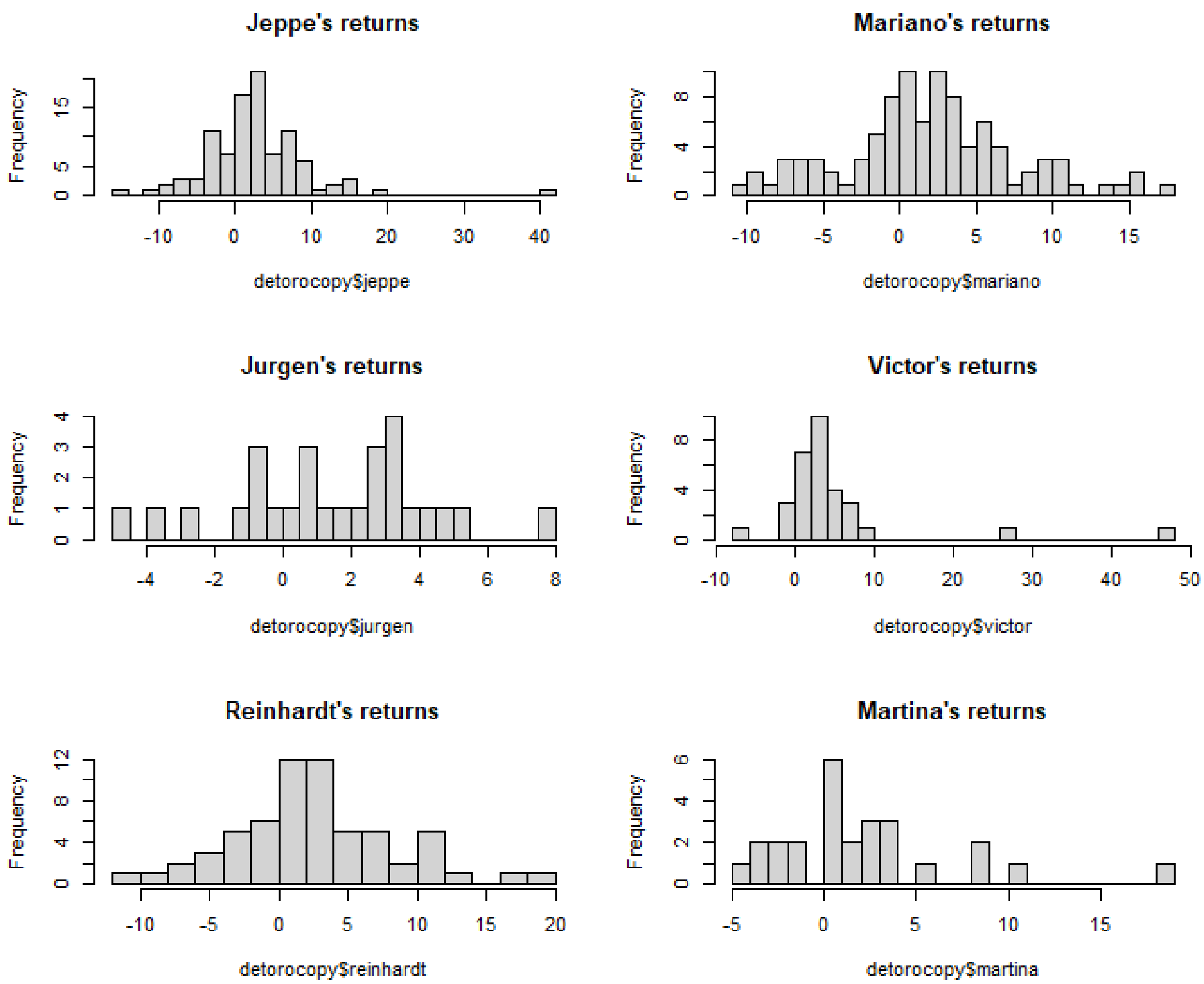
This means that when writing for loops to apply T-distribution fits for the copytraders, a function must be inserted that removes all the NA values before the fitting process.

Secondly the structure and class of the loaded object can be checked to ensure that the conversion to a time-series object was successful and we are dealing with numeric data. This is indeed the case. As certain functions cannot work with XTS-objects, we will convert and save the object as a dataframe too.

Next we can create histograms to test our assumptions of normality and homoskedacity. The assumption of independence is fulfilled either way because the returns of one trader cannot affect those of another.

```
par(mfrow=c(2,2))
hist(detorocopy$wesley, breaks = 20, main = "Wesley's returns")
hist(detorocopy$heloise, breaks = 20, main = "Heloise's returns")
hist(detorocopy$kieran, breaks = 20, main = "Kieran's returns")
hist(detorocopy$harry, breaks = 20, main = "Harry's returns")
```



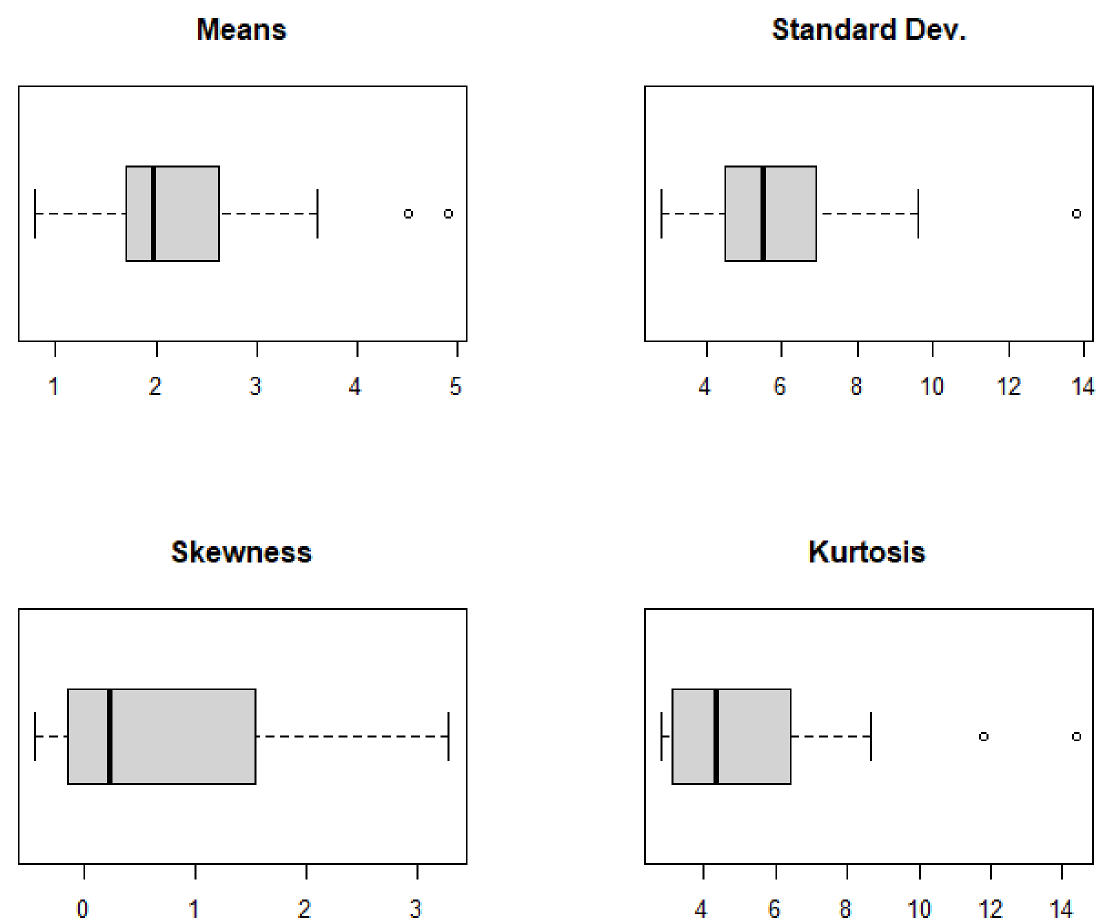


Looking at the histograms of return values of these eight traders one can spot rather large differences. It looks like the returns of copytraders that have been active for longer than about four to five years are relatively normally distributed. For individuals where this is not the case the some of the distributions look rather odd. For example, it is very likely that the EM algorithm cannot find a good fit for Jurgen’s returns, considering how uniform they are ditributed. In general, most histograms look relatively normal apart from a large frequency of outliers. This suggest that fitting a T-distribution is a good choice, as it captures both of these elements.

Besides looking at the histograms we can also find summary statistics of the distributions. The means and standard deviations can be calculated quite easily, but to find the skewness and kurtosis of the distributions the moments packages is needed.

```
for (i in 1:ncol(detorocopy)){
  means[i] <- mean(na.omit(detorocopy[,i]))
  sdev[i] <- sd(na.omit(detorocopy[,i]))
  skew[i] <- skewness(na.omit(detorocopy[,i]))
  kurt[i] <- kurtosis(na.omit(detorocopy[,i]))
}
```

To visualize these summary statistics we can generate a boxplot for each moment. The median monthly mean return for the copytraders looks to be around 2% per month, which is very high compared to the most used S&P 500 index. The standard deviations on the other hand are quite large, with some extreme values. Skewness for a normal distribution is around zero but most copytraders in our sample seem to have positive scores which indicates a longer right-tail of the distribution. Lastly, the kurtosis of a normal distribution is around three. While some copytraders appear to have a kurtosis value very close to three, some others have a very high value which indicates a very high number of outliers.



6. Model Development

While the workings of the underlying EM algorithm are rather complex, (it seeks to optimize a likelihood function through two iterative steps; expectation and maximization) actually fitting the T-distributions to the observed data is rather straightforward. One first creates empty dataframes to store the parameter estimates and standard erros for these parameters found by the algorithm. Because the observations of each individual copytrader are stored in the columns, the next step will be to construct a for loop that applies the algorithm to each column of observations and stores the result in the previously created dataframes. After doing so, the dataframes can be named using the column names of the original loaded excel sheet and binded to each other. Unfortunately the algorithm can't find adequate parameter values for two traders which results in an error that breaks the for loop. In order to fix this, the loop is re-run while excluding these two traders.

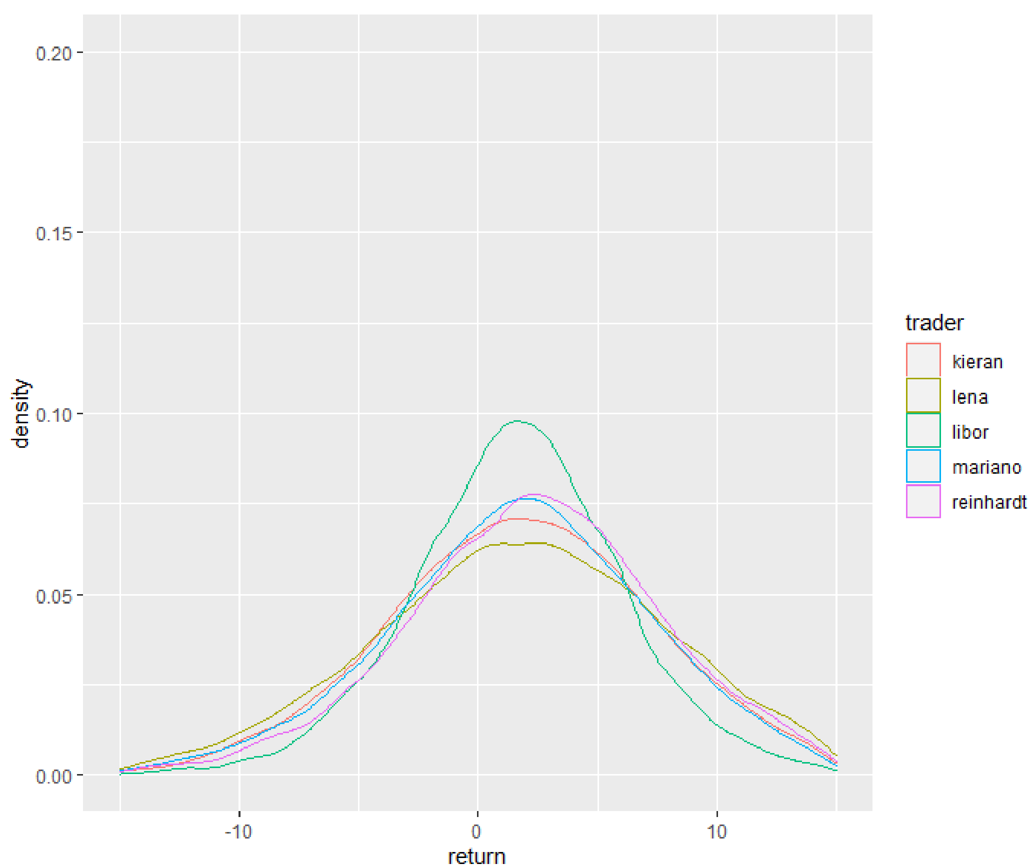
```
fit <- data.frame(nu = rep(0, 18), mu = rep(0, 18), sigma = rep(0,18))
ses <- data.frame(nus = rep(0, 18), mus = rep(0, 18), sigmas = rep(0,18))

output <- c()

for (i in 1:ncol(etorocopy)){
    if(i == 4 | i == 14){
      next
    }
    output <- fit.st(na.omit(etorocopy[,i]))
    print(output$par.ests)
    print(output$par.ses)
    fit[i,] <- output$par.ests
    ses[i,] <- output$par.ses
}

names <- colnames(etorocopy)
rownames(fit) <- names
rownames(ses) <- names

fit <- cbind(fit, ses)
```

The resulting dataframe contains all the information necessary to generate a very large random sample that approximates the population parameters. One can generate the random values from a distribution by passing the estimates of the fit to the rt-function. In order to propogate the uncertainty around the point estimates the input to the rt-function is randomly generated from a normal distribution with the mean being the parameter estimate and standard deviation being the standard error.

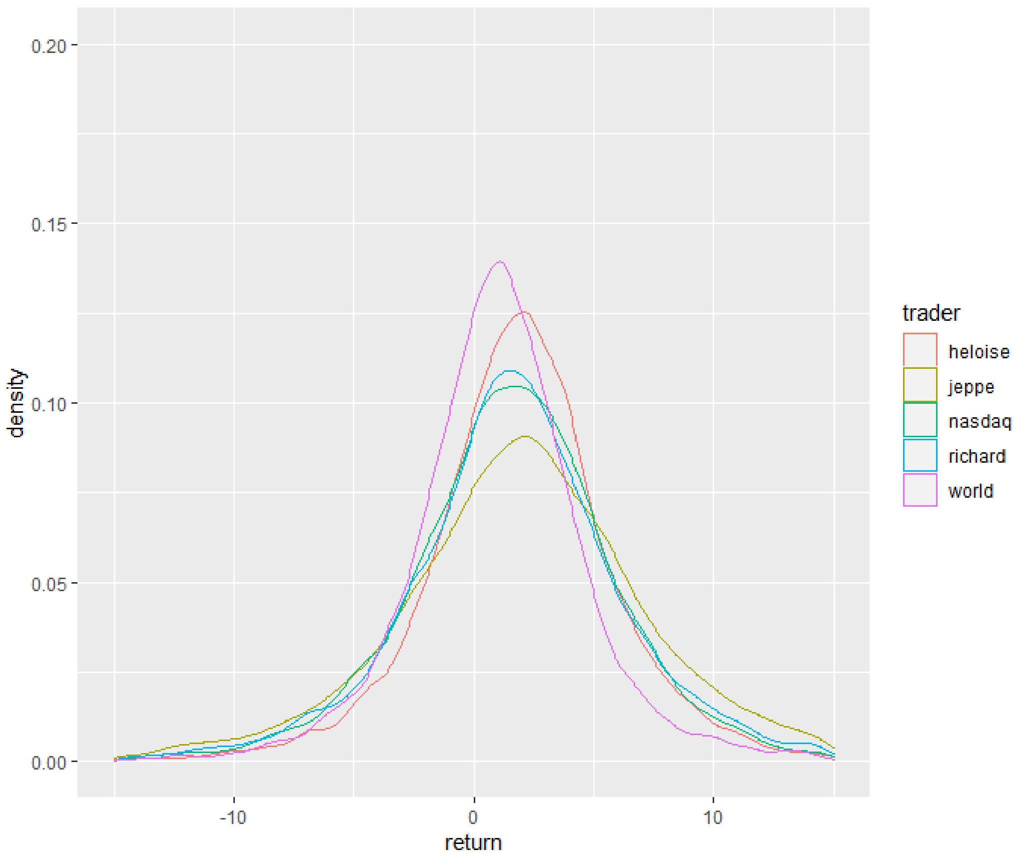
To visualize the resulting approximation of the population a density function can be plotted using the ggplot2 package. However the raw sample input is not suited for

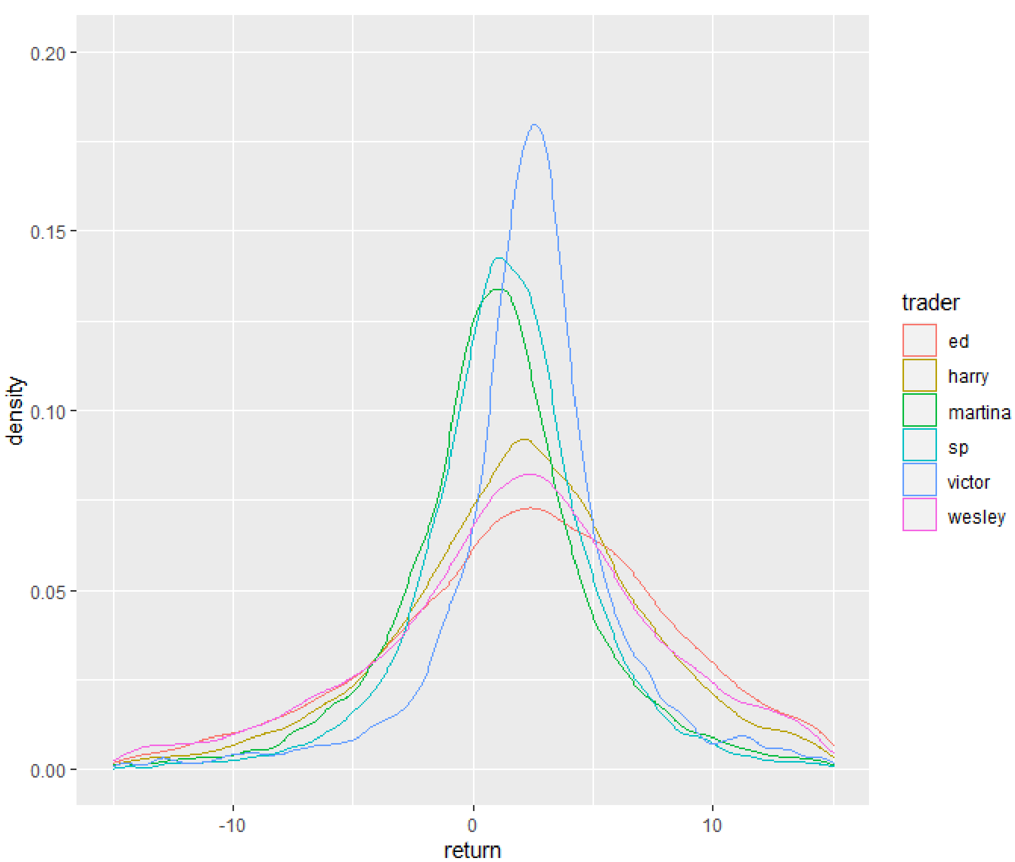
```
attach(fit)
ranking <- fit[order(-nu),]
print(ranking)

samples <- matrix(nrow = 10000, ncol = 18)

for(i in 1:nrow(ranking)){
 samples[,i] <- (rt(10000, df = ranking$nu[i]) * rnorm(10000, ranking$sigma[i],
ranking$sigmas[i]) +
                   rnorm(10000, ranking$mu[i], ranking$mus[i]))
}
colnames(samples) <- rownames(ranking)

metrics <- as.data.frame(matrix(nrow = 18, ncol = 8))
rownames(metrics) <- colnames(samples)
colnames(metrics) <- c("99%", "VaR", "70%", "50%", "20%", "5%", "1%", "ExpPos")

for (i in 1:16){
  metrics[i,] <- quantile(samples[,i], c(0.01, 0.05, 0.3, 0.5, 0.8, 0.95, 0.99, 1))
  metrics$ExpPos[i] <- sum(samples[,i] > 0)/length(samples[,i])*100
}

metrics <- round(metrics[1:16,], 1)

samples <- as.data.frame(samples)

long_samples <- pivot_longer(samples, cols = 1:16, names_to = "trader", values_to =
"return")
long_samples <- long_samples[,3:4]

sub_1 <- long_samples$trader %in% unique(long_samples$trader)[1:5]
long_samples[sub_1,]

  ggplot(long_samples[sub_1,], aes(x = return, color = trader)) +
  geom_density() +
  xlim(-15, 15) +
  ylim(0, 0.2)

sub_2 <- long_samples$trader %in% unique(long_samples$trader)[6:10]
long_samples[sub_2,]

  ggplot(long_samples[sub_2,], aes(x = return, color = trader)) +
    geom_density() +
    xlim(-15, 15) +
    ylim(0, 0.2)

sub_3 <- long_samples$trader %in% unique(long_samples$trader)[11:16]
long_samples[sub_3,]

  ggplot(long_samples[sub_3,], aes(x = return, color = trader)) +
    geom_density() +
    xlim(-15, 15) +
    ylim(0, 0.2)
```

generated values for all traders into a single column. An extra column has to be added that contains the name of the trader that the value belongs to, so that the data can still be grouped. Lastly the values for all traders and market indices are divided into three subgroups so that three separate plots can be made. If this isn't done, a single plot will contain all 16 density functions which is extremely difficult to read. While the plots display the information in the most intuitive way, a table of quantiles can also be calculated that will be useful for the ranking model to evaluate the copytraders over all metrics. In the script above the quantiles are saved in table named metrics, together with the expected percentage of months that the returns will be positive. The density plots are displayed below and can also be viewed on [GitHub.]()



Trader subgroup 1: The X-axis indicates the percentual monthly return while the Y-axis indicates the probability density


0.20
0.15
0.10
0.05
0.00
density
-10
0
10
return
trader
heloise
jeppe
nasdaq
richard
world


0.20
0.15
0.10
0.05
0.00
density
-10
0
10
return
trader
ed
harry
martina
sp
victor
wesley

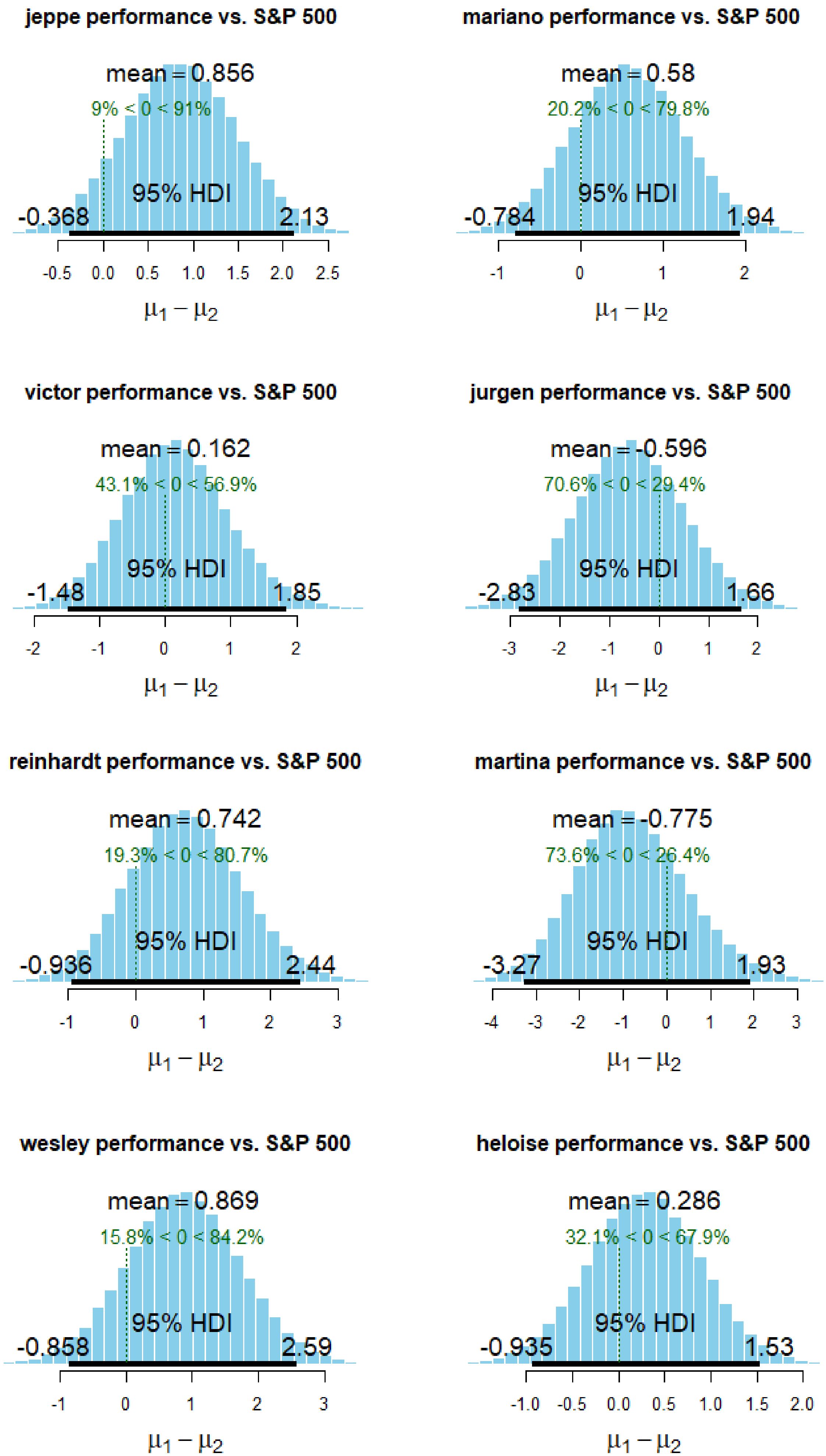
The second step is to measure individual performances versus the total market, so as to see if there is consistent outperformance. Whether or not outperforming the market is possible at all over the long run is a hot topic of debate within the finance and investment management community.[2] To test for outperformance one can use bayesian parameter estimation for both the market and the individual trader and subtract the difference. There are a number of reasons to use bayesian estimation in this case.

Firstly, bayesian inference is more resilient to outliers, which is important considering that we have seen a bountiful amount of outliers in the return data from the copytraders. Secondly, while frequentist statistics assume that there is a single fixed parameter and data is random, bayesian statistical techniques assume the opposite. That is, the underlying assumption is that the data provided is fixed and a distribution of parameters are possible that have generated this data. This means that we do not only get a point estimate but also a density function of parameters that fit the data. Because of this we can easily calculate the percentual chance that a trader will outperform the market in the long run, even despite the fact that both long-run return estimates are subject to a degree of uncertainty. The code looks as follows:

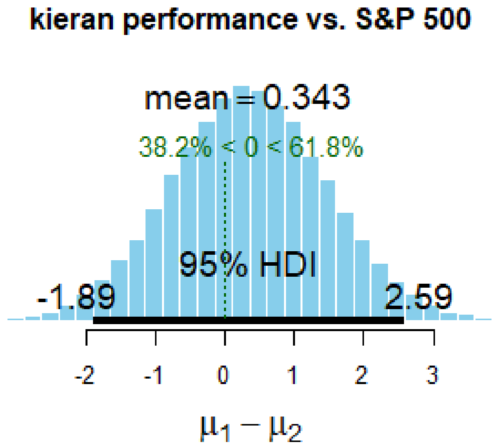
```
library(BEST)
library(dplyr)

bayes_fit <- list()
bayes_fitsp <- list()

for(i in 1:15){
    df <- as.data.frame(etorocopy) %>%
      select(i, nasdaq) %>%
      na.omit()

  BESTout <- BESTmcmc(df[,1], df$nasdaq)
    print(BESTout)
    plot(BESTout)

  bayes_fit[[i]] <- BESTout
}
```

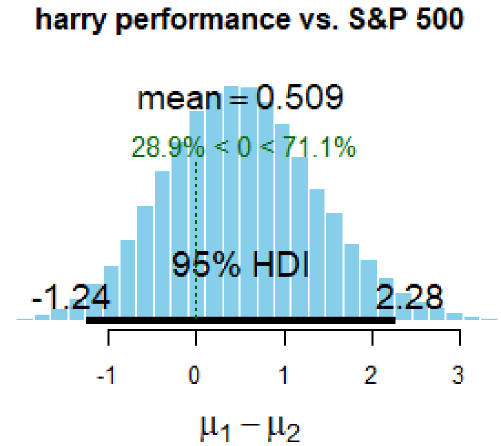
Because the summary output of each model stored in bayes_fit is not very intuitive, the best option is to simply generate plots for each copytrader like with the T-distribution fit. The only major difference is that now each individual requires their own plot. Fortunately the algorithm converged in all cases, unlike previously, which allows us to generate plots for all individuals.

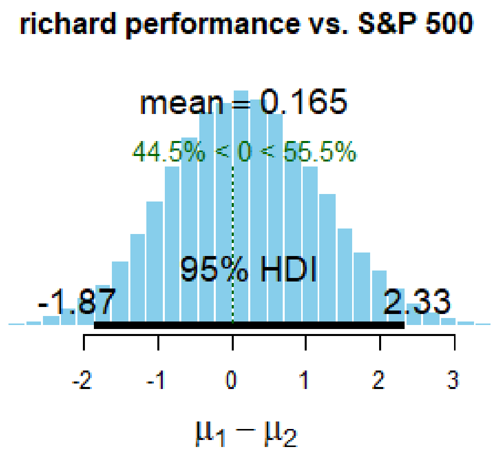
[2] John Jennings, "Why It's So Hard To Beat The Market," *Forbes,* 28 August, 2020, https://www.forbes.com/sites/johnjennings/2020/08/28/why-its-so-hard-to-beat-the-market/?sh=22c28ba2106d.

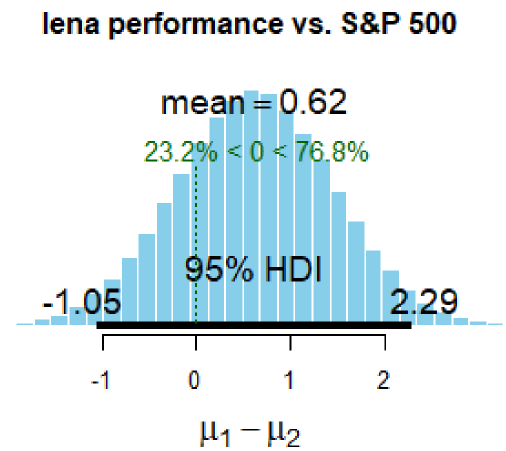


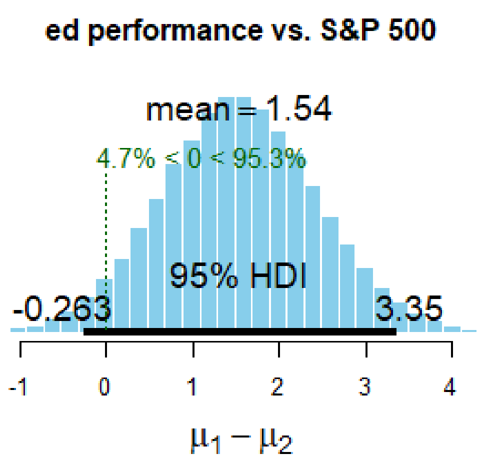
jeppe performance vs. S&P 500
mean = 0.856
9% < 0 < 91%
95% HDI
-0.368
2.13
-0.5 0.0 0.5 1.0 1.5 2.0 2.5
$\mu_1 - \mu_2$
mariano performance vs. S&P 500
mean = 0.58
20.2% < 0 < 79.8%
95% HDI
-0.784
1.94
-1 0 1 2
$\mu_1 - \mu_2$
victor performance vs. S&P 500
mean = 0.162
43.1% < 0 < 56.9%
95% HDI
-1.48
1.85
-2 -1 0 1 2
$\mu_1 - \mu_2$
jurgen performance vs. S&P 500
mean = -0.596
70.6% < 0 < 29.4%
95% HDI
-2.83
1.66
-3 -2 -1 0 1 2
$\mu_1 - \mu_2$
reinhardt performance vs. S&P 500
mean = 0.742
19.3% < 0 < 80.7%
95% HDI
-0.936
2.44
-1 0 1 2 3
$\mu_1 - \mu_2$
martina performance vs. S&P 500
mean = -0.775
73.6% < 0 < 26.4%
95% HDI
-3.27
1.93
-4 -3 -2 -1 0 1 2 3
$\mu_1 - \mu_2$
wesley performance vs. S&P 500
mean = 0.869
15.8% < 0 < 84.2%
95% HDI
-0.858
2.59
-1 0 1 2 3
$\mu_1 - \mu_2$
heloise performance vs. S&P 500
mean = 0.286
32.1% < 0 < 67.9%
95% HDI
-0.935
1.53
-1.0 -0.5 0.0 0.5 1.0 1.5 2.0
$\mu_1 - \mu_2$

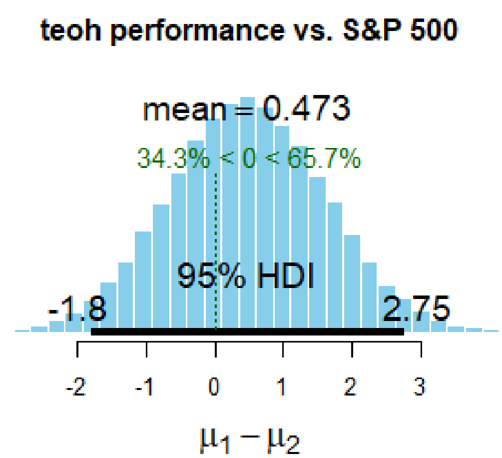

kieran performance vs. S&P 500
mean = 0.343
38.2% < 0 < 61.8%
95% HDI
-1.89
2.59
-2 -1 0 1 2 3
$\mu_1 - \mu_2$

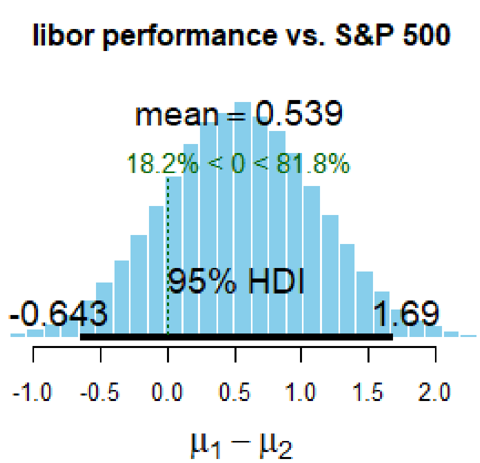

harry performance vs. S&P 500
mean = 0.509
28.9% < 0 < 71.1%
95% HDI
-1.24
2.28
-1 0 1 2 3
$\mu_1 - \mu_2$


richard performance vs. S&P 500
mean = 0.165
44.5% < 0 < 55.5%
95% HDI
-1.87
2.33
-2 -1 0 1 2 3
$\mu_1 - \mu_2$


lena performance vs. S&P 500
mean = 0.62
23.2% < 0 < 76.8%
95% HDI
-1.05
2.29
-1 0 1 2
$\mu_1 - \mu_2$


ed performance vs. S&P 500
mean = 1.54
4.7% < 0 < 95.3%
95% HDI
-0.263
3.35
-1 0 1 2 3 4
$\mu_1 - \mu_2$


teoh performance vs. S&P 500
mean = 0.473
34.3% < 0 < 65.7%
95% HDI
-1.8
2.75
-2 -1 0 1 2 3
$\mu_1 - \mu_2$


libor performance vs. S&P 500
mean = 0.539
18.2% < 0 < 81.8%
95% HDI
-0.643
1.69
-1.0 -0.5 0.0 0.5 1.0 1.5 2.0
$\mu_1 - \mu_2$

For the sake of brevity, only the performance versus the S&P 500 index is listed here. As one can see, the distributions vary widely for different individuals. While some traders have indeed consistently outperformed the index over the time they have been active, others have consistently underpeformed, despite their popularity. In particular, it seems that traders that have been active for longer generally perform much better if measured against a market index than on their plain monthly returns.

7. Results and Conclusion

The findings from both models are intriguing in their own right but they ultimately cannot answer the central question of this project: which copytraders should one copy? To do so a ranking system has to be created that takes all the data as imput and generates a final score or rank for each trader. In order to do this, the most important data was gathered and put in a new dataframe. The resulting table contains crucial performance metrics in the columns for each trader. These include seven quantiles from the T-distribution fit, the expect number of 'green' (positive return) –months, the low, mean and high estimate from the credible interval for outperformance versus the Nasdaq Composite, the same for the S&P 500 and lastly the correlation with the aforementioned indices. On top of that, the number of active trading months is also added as 'experience' along with a 'cryptocurrency distortion' column.

This last column is binary and indicates whether or not some of the very large returns copytraders have achieved were due to large holdings in cryptocurrency. Crypto assets had a remarkable bull run in late 2017 and again in late 2020. However the inherent value of a crypto asset is entirely based on speculation, unlike shares from companies which derive their value from the earnings generated by said company and are therefore

Traders\Performance	Alias	Crypto Dist.	Months of Data		Q 99%	Q 95% (VaR)
Jeppe Kirk Bonde	jeppe	NO	98		-17.7	-7.9
Mariano Pardo	mariano	NO	95		-12.7	-8
Victor Pedersen	victor	NO	31		-27.1	-5.9
Jürgen Schmitt	jurgen	NO	27			
Reinhardt Coetzee	reinhardt	NO	62		-12.8	-7.2
Martina Del Giorno	martina	NO	26		-15.8	-6.5
Wesley Warren Nolte	wesley	YES	78		-42.5	-13.9
Heloise Greeff	heloise	YES	63		-10.1	-4.6
Kieran Neil Wilkinson	kieran	NO	45		-11.6	-7.6
Harry Stephen Harrison	harry	NO	57		-21	-8.6
Richard Strout	richard	NO	42		-12.5	-6.4
Lena Birse	lena	NO	83		-14.1	-9

somewhat predictable. Because of this, I opted to exclude traders that benefited from the crypto bull run from the ranking as their results are not so much based in wisdom and skill but rather luck. This is a subjective assesment of course, but so is any ranking model.

In the table above one can find an impression of what the final results table looks like. From this data one can construct a myriad of scoring models. For the purposes of this project the scoring model ranks the traders across eleven categories:

1. Experience (Months of active trading)
2. 95% quantile or Value-at-Risk (Return for the worst month out of 20)
3. 50% quantile return (Median monthly return)
4. 5% quantile return (Return for the best month out of 20)
5. Low 95% credible interval boundary for outperformance vs. Nasdaq Composite
6. Median outperformance vs. Nasdaq Composite
7. High 95% credible interval boundary for outperformance vs. Nasdaq Composite
8. Low 95% credible interval boundary for outperformance vs. S&P 500
9. Median outperformance vs. S&P 500
10. High 95% credible interval boundary for outperformance vs. S&P 500
11. Average correlation to Nasdaq Compiste and S&P 500 combined (lower is better)

Because the EM algorithm failed to find an adequate T-fit for two traders, they are also excluded from the ranking process. This leaves 10 traders in total. Additonally, a second more conservative model is estimated that excludes the 5% quantile return and the upper 95% credible interval boundaries for outperformance. Below is an impression of what the ranking looks like:

Experience	VaR	Median Return	NSDQ Outp. Risk	NSDQ Outp. Mean	SP Outp. Risk
1	7	4	1	1	1
2	8	6.5	3	4	3
9	2	1.5	7	9	7
5	5	1.5	4	3	4
10	4	10	10	10	10
7	6	6.5	9	7	9
6	9	3	6	2	6
8	3	8.5	8	8	8
4	10	5	5	5	5
3	1	8.5	2	6	2

Finally, all the ranks across the categories are summed and divided by the total number of categories to create the rank sum average for both the full model and the risk cautious model. The results for these are as follows:

Total Rank Sum Average (lower is better):	
Jeppe Kirk Bonde	3.64
Mariano Pardo	5.50
Victor Pedersen	6.59
Reinhardt Coetzee	3.59
Martina Del Giorno	8.18
Kieran Neil Wilkinson	5.95
Harry Stephen Harrison	4.55
Richard Strout	6.23
Lena Birse	5.09
Libor Vasa	5.68

Total Rank Sum Average - Risk cautious:	
Jeppe Kirk Bonde	3.13
Mariano Pardo	5.06
Victor Pedersen	5.94
Reinhardt Coetzee	3.94
Martina Del Giorno	8.13
Kieran Neil Wilkinson	6.94
Harry Stephen Harrison	5.38
Richard Strout	6.69
Lena Birse	5.63
Libor Vasa	4.19

Interestingly enough, the results for both the full- and risk cautious model are fairly similar, indicating that traders with high upside potential also tend to have the highest overall performance. Jeppe Kirk Bonde performs extremely well in both categories which explains why he is one of the most copied traders on the platform. However, there are also hidden gems. Reinhardt Coetzee is not all that popular so may be considered a bit overlooked. Libor Vasa performs particularly well in the risk cautious model and has some of the fewest copiers from all traders in this list. Martina has some of the worst scores across all categories and also has the shortest active trading history of everyone in the original list of 15 traders. It is unclear how she made it into the list of most popular traders. One hypothesis is that her posts are in Italian, which might be attractive to other Italians regardless of her performance.

8. References

Joshua Oliver, “US retail investors drive summer surge in stocks,” *Financial Times,* 6 September, 2021, https://www.ft.com/content/d87c6631-55f0-4897-9634-bf0ad969e27d.

John Jennings, “Why It’s So Hard To Beat The Market,” *Forbes,* 28 August, 2020, https://www.forbes.com/sites/johnjennings/2020/08/28/why-its-so-hard-to-beat-the-market/?sh=22c28ba2106d.