Quantitative Assesment for Gelber Group

Riley Robertson

October 17, 2017

1 Scope

This report aims to develop and explain a pricing model for the Global X MSCI Greece ETF (GREK). The model will price the close value of the fund for any business day since its inception. The data used is mostly macroeconomic, like market volumes and exchange rates. I initially tried to search for securities not held by the fund that were highly correlated to the fund or to some of its largest holdings, but I was unable to find any significant relations in that search (exlcuding depository receipts). Unfortunately, many macroeconomic variables like imports/exports, unemployment, GDP, and consumer price infex were only available on a yearly basis, so they were not of much use when trying to price by the day.

2 Findings

After combing through all of the data available, I decided to use solely the USD-Euro exchange rate. I certainly would've liked to build a more sophisticated and robust model, but I was unable to acquire sufficient data (details in "Scope" section above). The relation betwee GREK's closing price and the exchange rate appears to be non-linear, but neither a log nor a square-root transform made a significant difference in the correlation between the two variables.

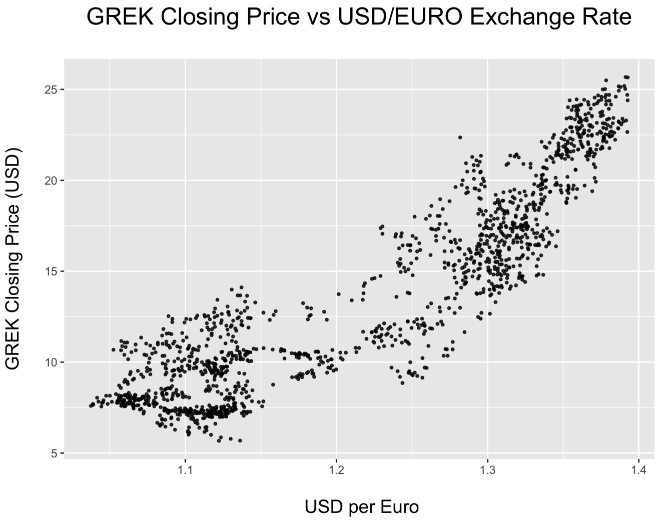

Figure 1: GREK Closing Price vs USD/EURO Exchange Rate

```
cor(data$close, data$ex_rate)

## [1] 0.9100587
```

```
cor(log(data$close), data$ex_rate)

## [1] 0.9126452

cor(sqrt(data$close), data$ex_rate)

## [1] 0.9151675
```

With that observation, I decided on running a LOESS local regression. This model had a surprisingly good fit

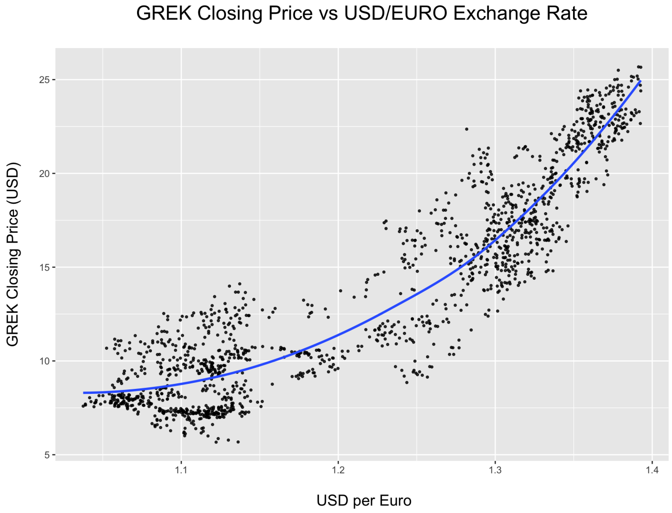

Figure 2: GREK Closing Price vs USD/EURO Exchange Rate

I say it's "surprisingly good" because it's based on just one explanatory variable, and it has the following fit metrics:

```
# The expected difference from the close price and the model is near 0
mean(data$close -  data$loess_prediction)

## [1] -0.01125454

# This model predicts ~88% of variation in close price
cor(data$close, data$loess_prediction)^2

## [1] 0.8786443
```

Because LOESS is non-parametric, we cannot write down what the model is. However, the supplied code allows one to price at any given time t by finding the exchange rate at t and then seeing what the predicted value of GREK's closing price is at that exchange rate.

```
prediction("2012/01/02")

## [1] 16.88624
```

The "YYYY/MM/DD" format must be used, but the given day need not be a business day (the model will predict on the first business day that comes after the inputted day)

3 Conclusion

Admittedly, this model is very simple. As I've said, this is due in largest part to the fact that I had difficulty getting relevant data that was collected daily. Some next steps to improve this model would be to consider more macroeconomic factors as well as use the prices of other securities to predict the price of GREK.

4 Appendix

Here is all of the R code that I've used for this project. The datasets I used can be found on my GitHub.

```r
# adding dependencies
library(Quandl)
library(ggplot2)
Quandl.api_key("sfk1toiCxwvhsvgMwwFe")

# reading the data in
grek <- read.csv("data/GREK_2yrs.csv")
ex_rates <- Quandl("FED/RXI_US_N_B_EU", start_date="2011-12-07")

# cleaning into the formats we'd like
grek$volume <- as.vector(grek$volume)
grek <- grek[2:nrow(grek),]
grek <- grek[order(grek$date),]
grek$Month <- paste(substr(grek$date, 1, 4), substr(grek$date, 6, 7), sep = "-")
ex_rates$date <- format(ex_rates$Date, format = "%Y/%m/%d")
names(ex_rates)[names(ex_rates) == "Value"] <- "ex_rate"

# looking at exchange rate
data <- merge(grek, ex_rates, by.x = "date")
lo <- loess(data$close ~ data$ex_rate, model = TRUE)
data$loess_prediction <- predict(lo, data.frame(close = seq(5, 25, length.out = nrow(data))))

prediction <- function(x) {
  while (nrow(data[data$date == x, ]) == 0) {
    if (substr(x, nchar(x), nchar(x)) != "9") {
      x <- paste(substr(x, 1, nchar(x) - 1),
                 as.numeric(substr(x, nchar(x), nchar(x))) + 1, sep="")
    } else {
      x <- paste(substr(x, 1, nchar(x) - 2),
                 as.numeric(substr(x, nchar(x) - 1, nchar(x) - 1)) + 1, 0, sep= "")
    }
  }
  return(lo$fitted[which(data$date == x)])
}
```