

Exchange Traded Funds

Harold Choi

26 Oct 2020

```
library(QRM)
library(qrmtools)
library(readr)
library(tidyverse)
library(zoo)
library(xts)
library(quantmod)
library(ggplot2)
library(magrittr)
library(broom)
library(lubridate)
library(PerformanceAnalytics)
```

Reading the data

- GSPC: S&P500
- VOO: Vanguard 500 Index Fund ETF Invests in stocks in the S&P 500 Index, representing 500 of the largest U.S. companies. Goal is to closely track the index's return, which is considered a gauge of overall U.S. stock returns. Offers high potential for investment growth; share value rises and falls more sharply than that of funds holding bonds. More appropriate for long-term goals where money growth is essential.
- QQQ: Tracks the Nasdaq 100 Index. Focus is on large international and U.S. companies in the technology, healthcare, industrial, consumer discretionary, and telecommunications sectors.
- XBI: SPDR S&P Biotech ETF. XBI tracks an equal-weighted index of US biotechnology stocks.
- VYM: Vanguard High Dividend Yield ETF. Seeks to track the performance of the FTSE® High Dividend Yield Index, which measures the investment return of common stocks of companies characterized by high dividend yields. Provides a convenient way to track the performance of stocks that are forecasted to have above-average dividend yields. Follows a passively managed, full-replication approach.

```
mystocks <- new.env(hash=TRUE)
getSymbols(c("QQQ", "XBI", "VYM", "VOO", "^GSPC"), env=mystocks, from ="2016-01-04", to ="2020-10-22")
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
## [1] "QQQ" "XBI" "VYM" "VOO" "^GSPC"
```

```
etf <- do.call(cbind,eapply(mystocks, Ad))
```

```
dim(etf)
```

```
## [1] 1210    5
```

```
colnames(etf)
```

```
## [1] "GSPC.Adjusted" "V00.Adjusted" "QQQ.Adjusted" "XBI.Adjusted"
## [5] "VYM.Adjusted"
```

```
head(etf)
```

```
##           GSPC.Adjusted V00.Adjusted QQQ.Adjusted XBI.Adjusted VYM.Adjusted
## 2016-01-04      2012.66    167.8177    105.13933    67.24879    56.71138
## 2016-01-05      2016.71    168.1181    104.95689    66.64403    56.99515
## 2016-01-06      1990.26    165.9875    103.94870    63.79861    56.24703
## 2016-01-07      1943.09    161.9448    100.69370    61.29029    55.03454
## 2016-01-08      1922.03    160.2239     99.86797    59.99151    54.41540
## 2016-01-11      1923.67    160.2422    100.17522    56.65039    54.47560
```

```
str(etf)
```

```
## An 'xts' object on 2016-01-04/2020-10-21 containing:
##   Data: num [1:1210, 1:5] 2013 2017 1990 1943 1922 ...
##   - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:5] "GSPC.Adjusted" "V00.Adjusted" "QQQ.Adjusted" "XBI.Adjusted" ...
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
##   List of 2
##   $ src      : chr "yahoo"
##   $ updated: POSIXct[1:1], format: "2020-10-26 23:27:57"
```

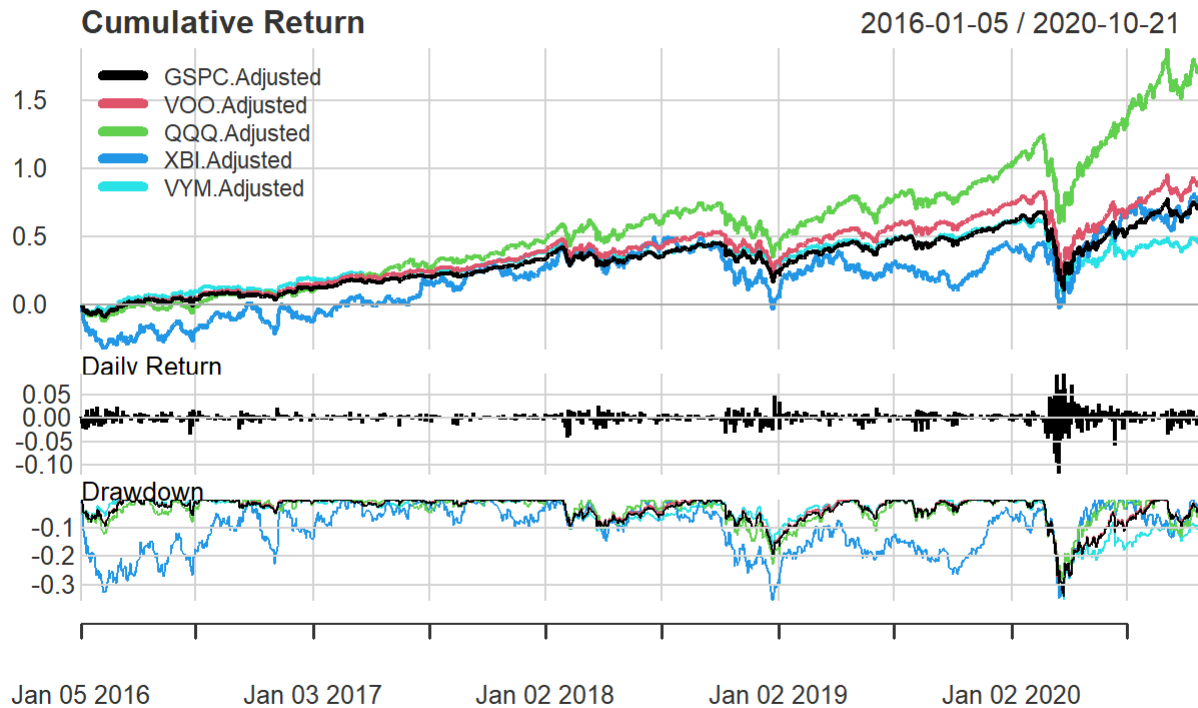
Calculating returns

```
etf_returns_discrete = Return.calculate(etf, method = c("discrete"))
etf_returns_log = Return.calculate(etf, method = c("log"))
```

Performance charts

```
charts.PerformanceSummary(etf_returns_discrete[,c(1:5)]
,main='Stock Absolute Performance',
legend.loc="topleft",
lwd=5)
```

Stock Absolute Performance



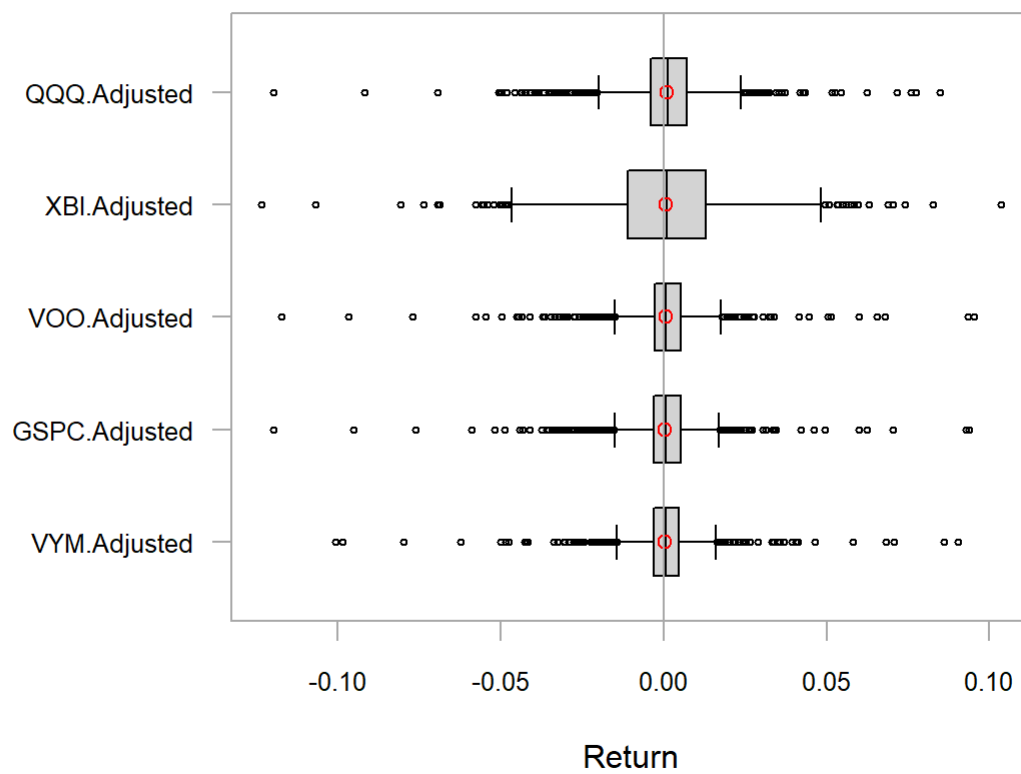
```
table.Stats(etf_returns_discrete[,c(1:5)])
```

```
##          GSPC.Adjusted V00.Adjusted QQQ.Adjusted XBI.Adjusted
## Observations      1209.0000    1209.0000    1209.0000    1209.0000
## NAs                1.0000      1.0000      1.0000      1.0000
## Minimum           -0.1198     -0.1174     -0.1198     -0.1235
## Quartile 1        -0.0029     -0.0028     -0.0038     -0.0110
## Median             0.0007      0.0008      0.0014      0.0009
## Arithmetic Mean    0.0005      0.0006      0.0009      0.0007
## Geometric Mean     0.0004      0.0005      0.0008      0.0004
## Quartile 3         0.0052      0.0053      0.0073      0.0129
## Maximum            0.0938      0.0954      0.0847      0.1037
## SE Mean            0.0004      0.0004      0.0004      0.0006
## LCL Mean (0.95)    -0.0002     -0.0001      0.0001     -0.0005
## UCL Mean (0.95)    0.0012      0.0013      0.0017      0.0018
## Variance            0.0001      0.0001      0.0002      0.0004
## Stdev              0.0122      0.0122      0.0139      0.0207
## Skewness           -0.7229     -0.6912     -0.6186     -0.1675
## Kurtosis           20.7238     20.7670     10.9160      2.7835
##          VYM.Adjusted
## Observations      1209.0000
## NAs                1.0000
## Minimum           -0.1006
## Quartile 1        -0.0029
## Median             0.0006
## Arithmetic Mean    0.0004
## Geometric Mean     0.0003
## Quartile 3         0.0048
## Maximum            0.0905
## SE Mean            0.0003
## LCL Mean (0.95)    -0.0003
## UCL Mean (0.95)    0.0010
## Variance            0.0001
## Stdev              0.0116
## Skewness           -0.5832
## Kurtosis           20.2682
```

Comparing distribution using boxplots

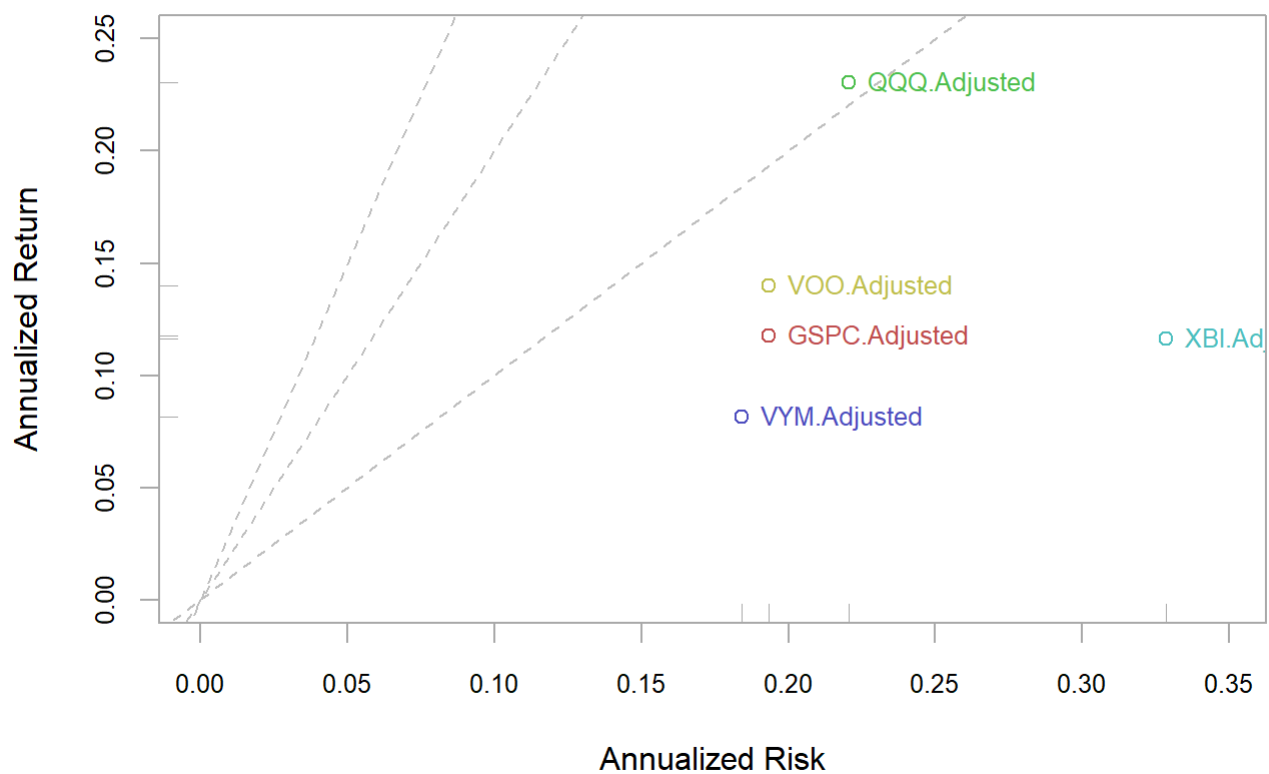
```
chart.Boxplot(etf_returns_discrete[,c(1:5)], main = "Trailing 36-Month Returns")
```

Trailing 36-Month Returns

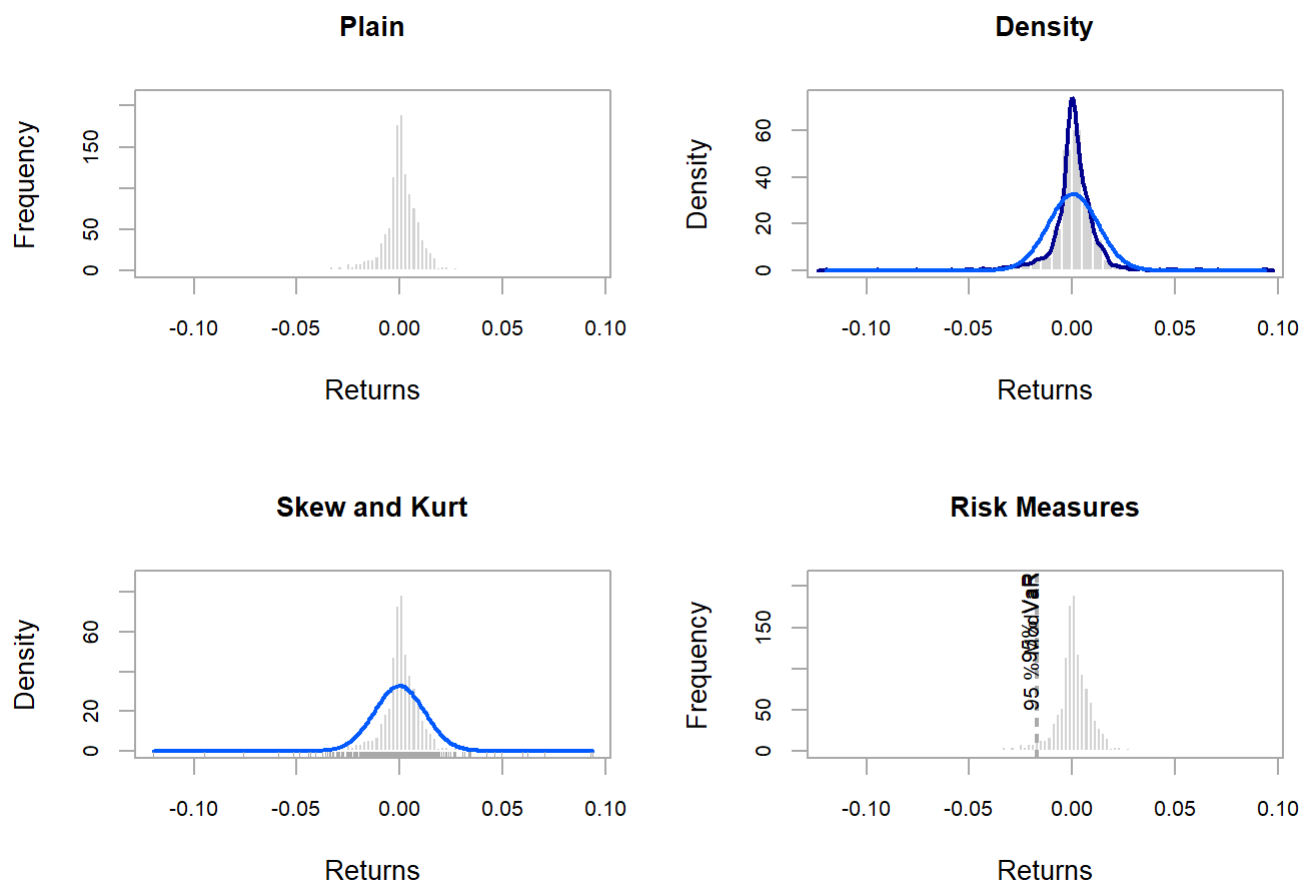


```
chart.RiskReturnScatter(etf_returns_discrete[,c(1:5)],
  main = "Trailing 36-Month Performance",
  colorset=rainbow6equal)
```

Trailing 36-Month Performance



```
layout(rbind(c(1,2),c(3,4)))
chart.Histogram(etf_returns_discrete[,1,drop=FALSE], main = "Plain", methods = NULL)
chart.Histogram(etf_returns_discrete[,1,drop=FALSE], main = "Density", breaks=40, methods = c("add.density", "add.normal"))
chart.Histogram(etf_returns_discrete[,1,drop=FALSE], main = "Skew and Kurt", methods = c("add.centered", "add.rug"))
chart.Histogram(etf_returns_discrete[,1,drop=FALSE], main = "Risk Measures", methods = c("add.risk"))
```

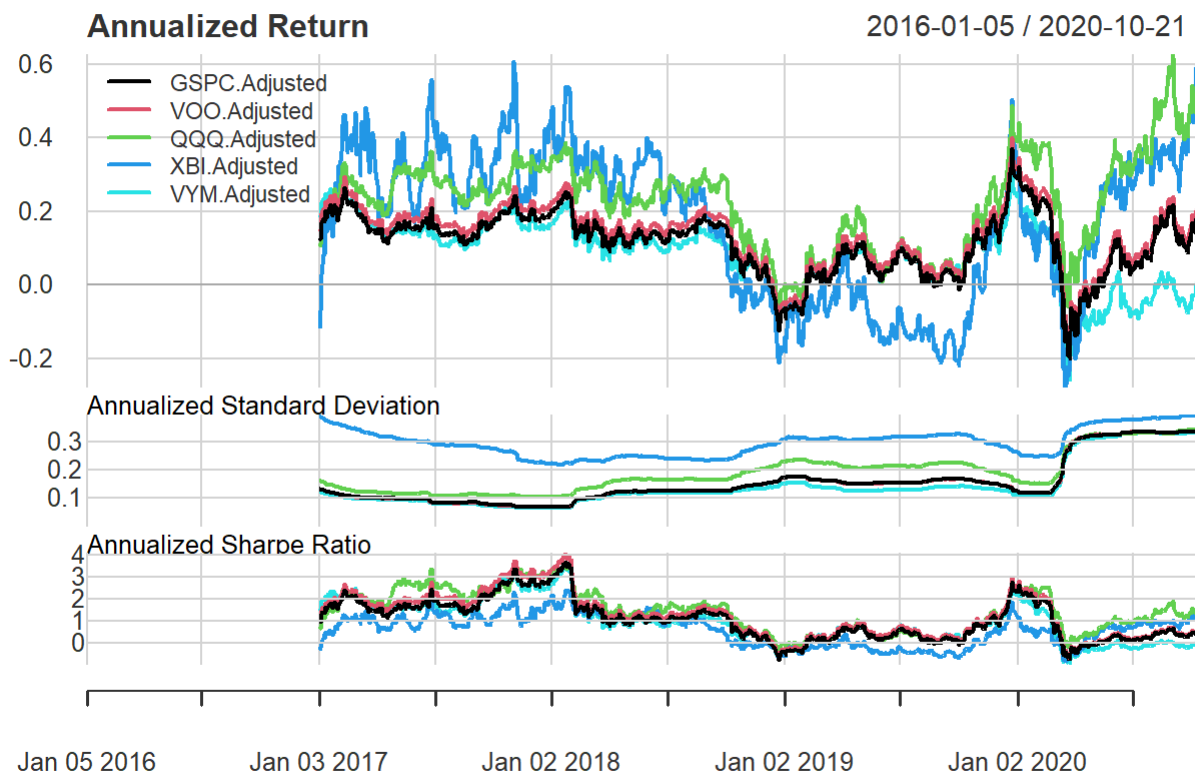


Measuring performance consistency

- Rolling performance is typically used as a way to assess stability of a return stream. Although perhaps it doesn't get much credence in the financial literature because of its roots in digital signal processing, many practitioners find rolling performance to be a useful way to examine and segment performance and risk periods.

```
charts.RollingPerformance(etf_returns_discrete,
                          width = 252,
                          main="Rolling 12-Month Performance",
                          legend.loc="topleft",
                          lwd=2)
```

Rolling 12-Month Performance



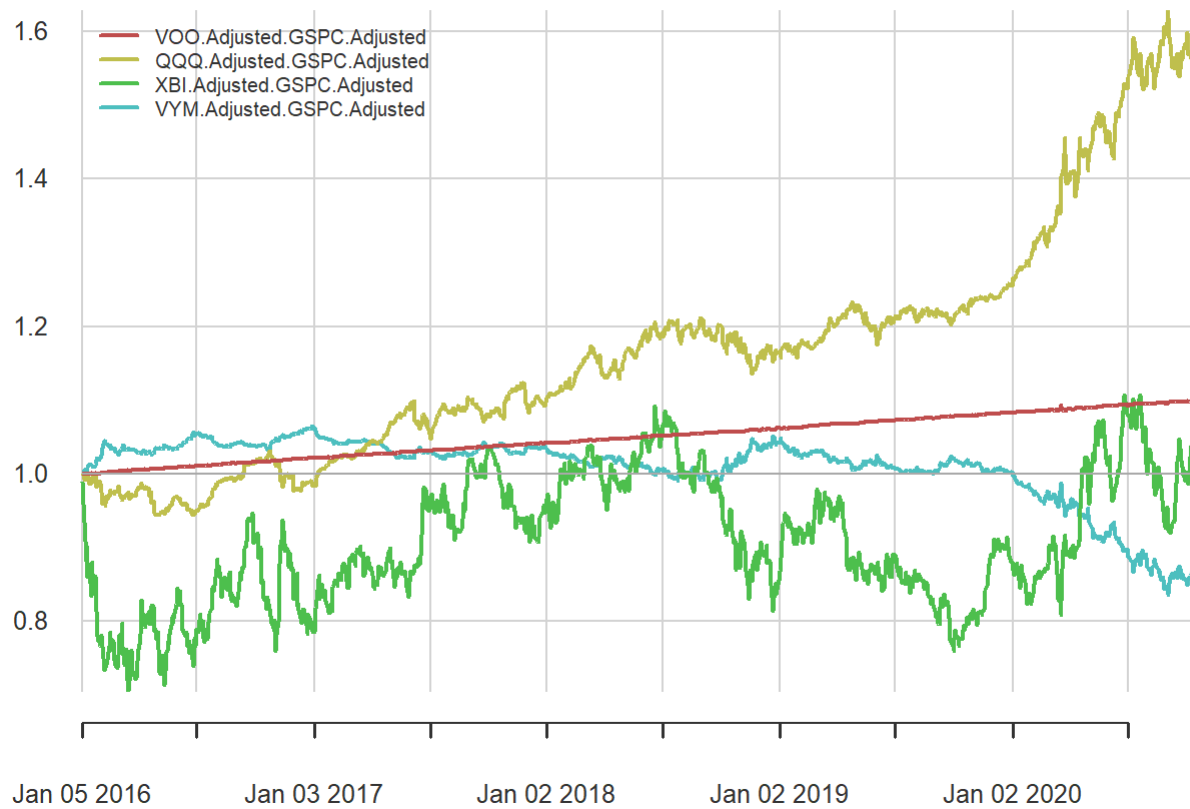
Measuring relative performance

- Identifying and using a benchmark can help us assess and explain how well we are meeting our investment objectives, in terms of a widely held substitute. A benchmark can help us explain how the portfolios are managed, assess the risk taken and the return desired, and check that the objectives were respected. Benchmarks are used to get better control of the investment management process and to suggest ways to improve selection.

```
chart.RelativePerformance(etf_returns_discrete[,c(2:5), drop =FALSE],
  etf_returns_discrete[,c(1),drop =FALSE],
  colorset = rainbow6equal,
  lwd = 4,
  legend.loc = "topleft",
  main="Relative performance to S&P500")
```


Relative performance to S&P500

2016-01-05 / 2020-10-21

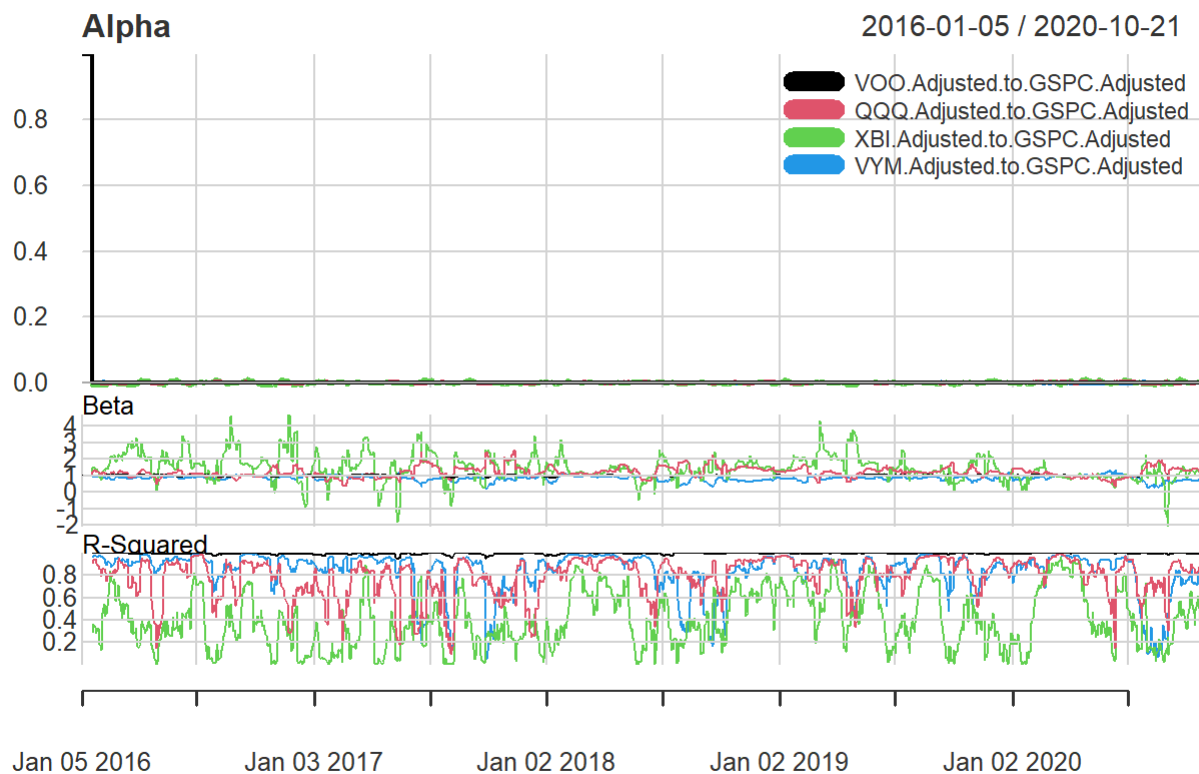


```
table.CAPM(etf_returns_discrete[,c(2:5), drop =FALSE],
  etf_returns_discrete[,c(1),drop =FALSE],)
```

##	V00.Adjusted to GSPC.Adjusted	QQQ.Adjusted to GSPC.Adjusted
## Alpha	0.0001	0.0004
## Beta	0.9986	1.0596
## Beta+	1.0014	1.0215
## Beta-	0.9996	1.0185
## R-squared	0.9983	0.8640
## Annualized Alpha	0.0201	0.0982
## Correlation	0.9992	0.9295
## Correlation p-value	0.0000	0.0000
## Tracking Error	0.0079	0.0821
## Active Premium	0.0222	0.1124
## Information Ratio	2.8024	1.3687
## Treynor Ratio	0.1403	0.2174
##	XBI.Adjusted to GSPC.Adjusted	VYM.Adjusted to GSPC.Adjusted
## Alpha	0.0001	-0.0001
## Beta	1.1356	0.9164
## Beta+	0.9825	0.9466
## Beta-	1.0906	0.9291
## R-squared	0.4471	0.9256
## Annualized Alpha	0.0162	-0.0236
## Correlation	0.6687	0.9621
## Correlation p-value	0.0000	0.0000
## Tracking Error	0.2457	0.0528
## Active Premium	-0.0017	-0.0363
## Information Ratio	-0.0070	-0.6876
## Treynor Ratio	0.1023	0.0891

```
charts.RollingRegression(etf_returns_discrete[,c(2:5), drop =FALSE],
                        etf_returns_discrete[,c(1),drop =FALSE],
                        lwd = 10,
                        legend.loc = "topright")
```

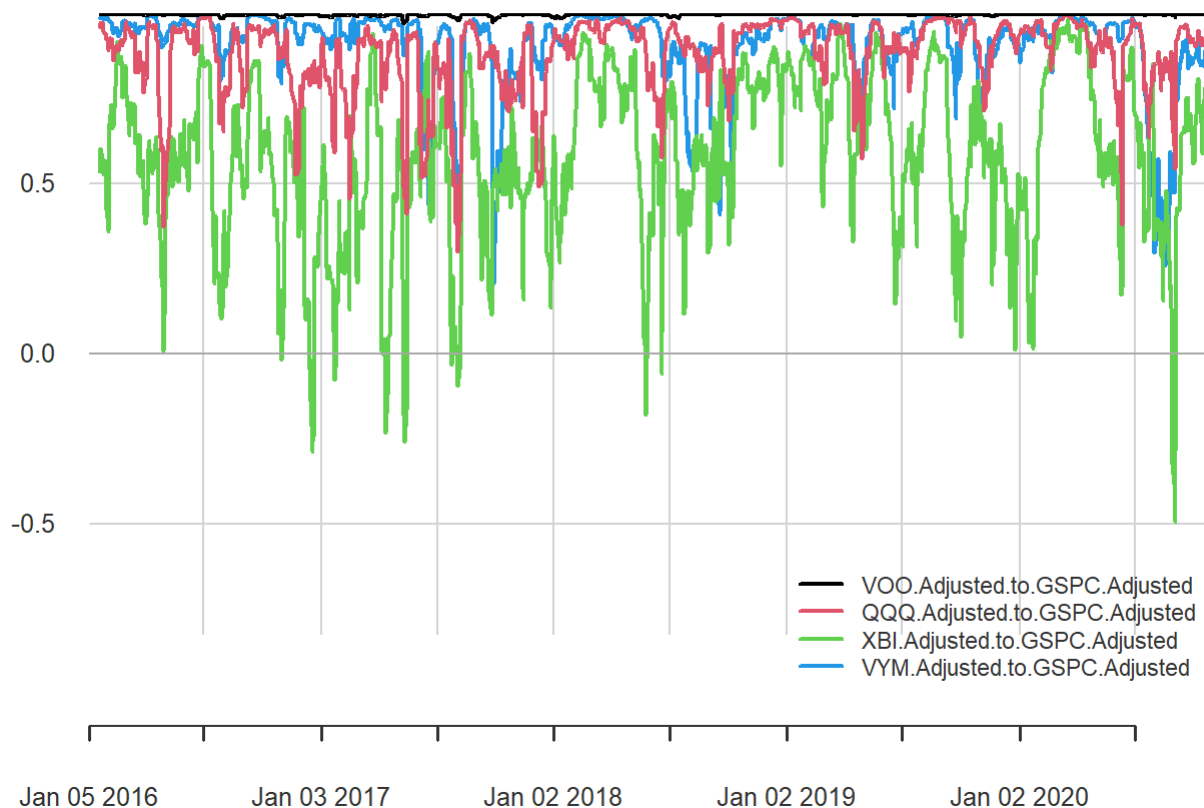
Rolling 12-day Regressions



```
chart.RollingCorrelation(etf_returns_discrete[,c(2:5), drop =FALSE],
  etf_returns_discrete[,c(1),drop =FALSE],
  lwd = 2,
  main = "12-Month Rolling Correlation",
  legend.loc = "bottomright")
```

12-Month Rolling Correlation

2016-01-05 / 2020-10-21



```
table.Correlation(etf_returns_discrete[,c(2:5), drop =FALSE],
  etf_returns_discrete[,c(1),drop =FALSE],
  legend.loc = "lowerleft")
```

##		Correlation	p-value	Lower CI	Upper CI
##	VOO.Adjusted to GSPC.Adjusted	0.9991589	0.00000e+00	0.9990585	0.9992487
##	QQQ.Adjusted to GSPC.Adjusted	0.9295250	0.00000e+00	0.9214345	0.9368098
##	XBI.Adjusted to GSPC.Adjusted	0.6686687	1.63713e-157	0.6362769	0.6987070
##	VYM.Adjusted to GSPC.Adjusted	0.9621029	0.00000e+00	0.9576703	0.9660794

Downside Risk

- Semi deviation looks only at negative price fluctuation and is an alternative measurement to standard deviation or variance. It is used to evaluate the downside risk of an investment. A higher number represents greater fluctuations in the negative price of the ETF.
- Gain deviation is similar calculation to standard deviation and the opposite of loss deviation. It calculates the deviation using only up period returns, variances and the number of up periods, a higher number is means there is more deviation in the gains of the ETF.
- Loss deviation is similar to standard deviation but calculates the deviation using only the down period returns, variances and number of down periods.

```
table.DownsideRisk(etf_returns_discrete[,1:5])
```

##	GSPC.Adjusted	V00.Adjusted	QQQ.Adjusted
## Semi Deviation	0.0091	0.0091	0.0103
## Gain Deviation	0.0089	0.0089	0.0095
## Loss Deviation	0.0111	0.0112	0.0121
## Downside Deviation (MAR=210%)	0.0133	0.0132	0.0143
## Downside Deviation (Rf=0%)	0.0089	0.0088	0.0100
## Downside Deviation (0%)	0.0089	0.0088	0.0100
## Maximum Drawdown	0.3392	0.3399	0.2856
## Historical VaR (95%)	-0.0175	-0.0176	-0.0214
## Historical ES (95%)	-0.0310	-0.0309	-0.0350
## Modified VaR (95%)	-0.0168	-0.0166	-0.0212
## Modified ES (95%)	-0.0168	-0.0166	-0.0458
##	XBI.Adjusted	VYM.Adjusted	
## Semi Deviation	0.0148	0.0086	
## Gain Deviation	0.0136	0.0087	
## Loss Deviation	0.0143	0.0105	
## Downside Deviation (MAR=210%)	0.0192	0.0129	
## Downside Deviation (Rf=0%)	0.0145	0.0084	
## Downside Deviation (0%)	0.0145	0.0084	
## Maximum Drawdown	0.3526	0.3521	
## Historical VaR (95%)	-0.0325	-0.0166	
## Historical ES (95%)	-0.0458	-0.0291	
## Modified VaR (95%)	-0.0332	-0.0158	
## Modified ES (95%)	-0.0517	-0.0158	