Happy 2018 and welcome to our first reproducible finance post of the year! What better way to ring in a new beginning than pondering/calculating/visualizing returns distributions.

We ended 2017 by tackling [skewness](https://rviews.rstudio.com/2017/12/13/introduction-to-skewness/), and we will begin 2018 by tackling kurtosis.

Kurtosis is a measure of the degree to which portfolio returns appear in the tails of our distribution. A normal distribution has a kurtosis of 3, which follows from the fact that a normal distribution does have some of its mass in its tails. A distribution with a kurtosis greater than 3 has more returns out in its tails than the normal, and one with kurtosis less than 3 has fewer returns in its tails than the normal. That matters to investors because more bad returns out in tails means that our portfolio might be at risk of a rare but huge downside. The terminology is a bit confusing. Negative kurtosis is considered less risky because it has fewer returns out in the tails. Negative == less risky? We’re not used to that in finance.

Kurtosis is often has the word ‘excess’ appended to its description, as in ‘negative excess kurtosis’ or ‘positive excess kurtosis’. That ‘excess’ is in comparison to a normal distribution kurtosis of 3. A distribution with negative excess kurtosis equal to -1 has an actual kurtosis of 2.

Enough with the faux investopedia entry, let’s get to the calculations, R code and visualizations.

Here’s the equation for excess kurtosis. Note that we subtract 3 at the end:

\[Kurtosis=\sum\_{t=1}^n (x\_i-\overline{x})^4/n \bigg/ (\sum\_{t=1}^n (x\_i-\overline{x})^2/n)^{2}-3 \]

By way of reminder, we will be working with our usual portfolio consisting of:

+ SPY (S&P500 fund) weighted 25%

+ EFA (a non-US equities fund) weighted 25%

+ IJS (a small-cap value fund) weighted 20%

+ EEM (an emerging-mkts fund) weighted 20%

+ AGG (a bond fund) weighted 10%

Before we can calculate kurtosis, we need to find portfolio monthly returns, which was covered in [this post](https://rviews.rstudio.com/2017/10/11/from-asset-to-portfolio-returns/).

Building off that previous work, we will be working with two objects of portfolio returns:

+ portfolio\_returns\_xts\_rebalanced\_monthly (an xts of monthly returns)

+ portfolio\_returns\_tq\_rebalanced\_monthly (a tibble of monthly returns)

Now we are going to test our past self’s [work on skewness](https://rviews.rstudio.com/2017/12/13/introduction-to-skewness/), and reuse that code flow to expedite the kurtosis work. The logic will remain the same, but we will call different built-in functions and different by-hand calculations.

For the xts world, we use the kurtosis() function instead of the skewness() function.

kurt\_xts <- kurtosis(portfolio\_returns\_xts\_rebalanced\_monthly$returns)

kurt\_xts

## [1] 0.5267736

For tidy, we have the same piped flow and use the formula for kurtosis for our by-hand calculations. Our by-hand result is labeled with kurt\_byhand, and involves quite a few parentheticals to map it back to the kurtosis equation above.

kurt\_tidy <-

portfolio\_returns\_tq\_rebalanced\_monthly %>%

summarise(

kurt\_builtin = kurtosis(returns),

kurt\_byhand =

((sum((returns - mean(returns))^4)/length(returns))/

((sum((returns - mean(returns))^2)/length(returns))^2)) - 3) %>%

select(kurt\_builtin, kurt\_byhand)

Let’s confirm that we have consistent calculations.

kurt\_xts

## [1] 0.5267736

kurt\_tidy$kurt\_builtin

## [1] 0.5267736

kurt\_tidy$kurt\_byhand

## [1] 0.5267736

We have consistent results from xts and the tidy built-in/by-hand worlds, and we were able to reuse our code from above to shorten the development time here. dd

Let’s do the same with the visualizations and head straight for a density plot, starting with the same portfolio\_density\_plot.

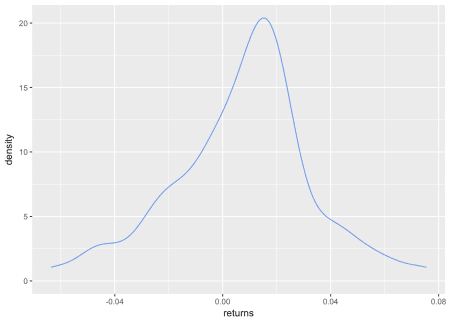
portfolio\_density\_plot <-

portfolio\_returns\_tq\_rebalanced\_monthly %>%

ggplot(aes(x = returns)) +

stat\_density(geom = "line", alpha = 1, colour = "cornflowerblue")

portfolio\_density\_plot



We are interested in *both* tails for kurtosis, so let’s shade at 2 standard deviations above and below the mean return (for our skewness work, we only shaded the negative tail).

mean <- mean(portfolio\_returns\_tq\_rebalanced\_monthly$returns)

sd\_pos <- mean + (2 \* sd(portfolio\_returns\_tq\_rebalanced\_monthly$returns))

sd\_neg <- mean - (2 \* sd(portfolio\_returns\_tq\_rebalanced\_monthly$returns))

sd\_pos\_shaded\_area <-

ggplot\_build(portfolio\_density\_plot)$data[[1]] %>%

filter(x > sd\_pos )

sd\_neg\_shaded\_area <-

ggplot\_build(portfolio\_density\_plot)$data[[1]] %>%

filter(x < sd\_neg)

portfolio\_density\_plot <-

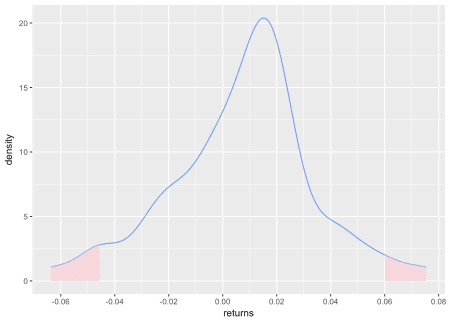
portfolio\_density\_plot +

geom\_area(data = sd\_pos\_shaded\_area, aes(x = x, y = y), fill="pink", alpha = 0.5) +

geom\_area(data = sd\_neg\_shaded\_area, aes(x = x, y = y), fill="pink", alpha = 0.5) +

scale\_x\_continuous(breaks = scales::pretty\_breaks(n = 10))

portfolio\_density\_plot



That density chart is a good look at the mass in both tails, where we have defined ‘tail’ as being two standard deviations away from the mean. We can add a line for the mean, as did in our skewness visualization, with the following.

mean <- mean(portfolio\_returns\_tq\_rebalanced\_monthly$returns)

mean\_line\_data <-

ggplot\_build(portfolio\_density\_plot)$data[[1]] %>%

filter(x <= mean)

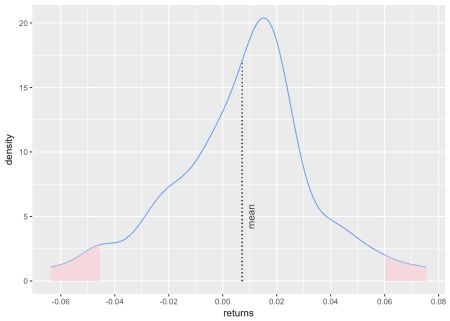
portfolio\_density\_plot +

geom\_segment(data = mean\_line\_data, aes(x = mean, y = 0, xend = mean, yend = density),

color = "black", linetype = "dotted") +

annotate(geom = "text", x = mean, y = 5, label = "mean",

fontface = "plain", angle = 90, alpha = .8, vjust = 1.75)



Finally, we can calculate and chart the rolling kurtosis with the same logic as we did for skewness. The only difference is that here we call fun = kurtosis instead of fun = skewness.

window <- 6

rolling\_kurt\_xts <- na.omit(apply.rolling(portfolio\_returns\_xts\_rebalanced\_monthly, window,

fun = kurtosis))

Now we pop that xts object into highcharter for a visualization.

highchart(type = "stock") %>%

hc\_title(text = "Rolling Kurt") %>%

hc\_add\_series(rolling\_kurt\_xts, name = "Rolling kurtosis", color = "cornflowerblue") %>%

hc\_yAxis(title = list(text = "kurtosis"),

opposite = FALSE,

max = .03,

min = -.03) %>%

hc\_navigator(enabled = FALSE) %>%

hc\_scrollbar(enabled = FALSE)

We didn’t cover this before, but what if we wanted to use ggplot for the rolling kurtosis? We could convert that xts object to a tibble with tk\_tbl() from the timetk package, and then pipe straight to ggplot.

rolling\_kurt\_xts %>%

tk\_tbl(preserve\_index = TRUE, rename\_index = "date") %>%

rename(rolling\_kurtosis = calcs) %>%

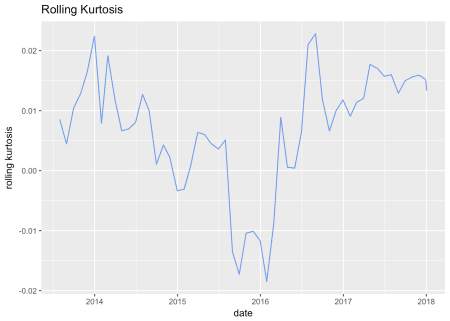
ggplot(aes(x = date, y = rolling\_kurtosis)) +

geom\_line(color = "cornflowerblue") +

xlab("date") +

ylab("rolling kurtosis") +

ggtitle("Rolling Kurtosis")



Interestingly, this portfolio has displayed slight positive rolling excess kurtosis for most of its life, except during the last half of 2015 through early 2016.

That’s all for today. Our work on kurtosis was made a lot more efficient by our work on skewness – so let’s thank our 2017 selves for constructing a reproducible and reusable code flow! See you next time.