The “yield curve” is the name given to the graphic showing the different annual rates paid on investors who in effect lend money to the US Treasury. In normal circumstances, investors who are prepared to make the loan on a long term basis demand higher effective interest rates as compensation for risk associated with inflation and future interest rates. When the situation goes in the other direction, and higher effective returns come from shorter term investments, it’s commonly seen as a sign of a coming recession. This isn’t because the “inverted yield curve” causes the recession, but because (amongst other things) it reflects investors’ fears that interest rates are about to be cut (in response to lower economic growth), hence best to lock in the return you can get now.

**Animation options**

Anyway, here is my version of the animation, now with 29 years of daily data, streaming from You Tube (because it was too large to embed directly in my web page):

Alternatively, here’s a streamlined social-media-friendly version (as a GIF) that has monthly averages and scoots through a bit quicker.

I’m not sure these are great graphics. For one thing, with three frames for every day with data reported, the daily version is over 20,000 frames long and even at 25 frames per second that’s a pretty tedious thing to watch – 14.5 minutes in fact. More generally, it suffers from the common problems of animated statistical graphics:

* you have to be looking at just the right time to see the interesting things – like the yield inversion and other volatility in that 2006 to 2008 period
* some changes over time are actually less obvious to the viewer than if they could be presented on a single page.

**Static alternatives**

Animations aren’t always the best. In fact, it’s surely significant that I loathe video tutorials because I can’t flick my eye directly to the part I want but have to sit through the whole thing – even at high speed this is frustrating. Animations can have a similar problem.

For showing the longer history at a glance, consider this alternative presentation, still related to the animation’s visual structure. This has all 8,000 or so yield curves from those 29 years on a single static chart, each one drawn with 10% opacity to avoid colouring the page solid.

What I like about this version is that because it’s all there at once, you can make the comparisons across a broader period of time than is possible with the animation. You can see:

* The secular decrease in rates over this time period
* The period around 2005 when there were no 30 year rates for some reason
* The different yield inversions in context – showing as upwards swooshes in blue, green and yellow (for different periods) at the left of the graph

Plus it’s very pretty.

Finally, how about a much more traditional time series line chart. I actually find this the easiest to interpret of them all.

* The secular decrease is obvious again.
* The carefully chosen colours let you compare the different terms quite easily.
* This is the most dramatic depictiton of the way the shorter term yields hit the floor after 2008.
* Easiest chart of them all to interpret because of the familiar convention of the horizontal axis meaning time
* Still very pretty.

**Reflections**

So what do I think?

Sometimes static graphs are beter than animations. Sometimes an old fashioned time series plot is better than less standard innovations. To make an animation tell a good story you often need more polishing, annotation and framing, not less.

More information on yield curves is easy to find on the Internet:

[](https://www.ft.com/content/cf9eb29a-5220-11e9-9c76-bf4a0ce37d49)

One of the things that makes a difference to this, and to the animations in the same story, is the careful use of annotations to tell a story. That’s definitely what it takes to get a graphic to the next level, but it’s not something I’ve got time for right now.

Here’s the code for this (apart from that last one from the FT) – all pretty straightforward. Grabbing the data is easy with Wickham’s rvest R package, and animations are so much easier than a few years ago thanks to Thomas Pedersen’s gganimate. And let’s not forget the Viridis colours, originally developed for Python by Nathaniel Smith and Stefan van der Walt, and ported into an R package by Simon Garnier. In these graphics I use three variants of the Viridis colour schemes to represent different variables – the difference between long and short term yields at any one point in time, year, and the term of the yield. In each case, the variable I am trying to show is fundamentally ordered, and Viridis is at its best in showing that ordered nature in its colour.

#------------------------------load up functionality and import data-------------

library(rvest)

library(tidyverse)

library(scales)

library(gganimate)

library(svglite)

library(frs)

library(lubridate)

the\_caption <- "Data from https://www.treasury.gov/, analysis by freerangestats.info"

# we read in the data a year at a time because although there is a page with all 29 years of data,

# it was too difficult to read in all at once ie it crashed my humble laptop. So we define a function

# to get one year's data:

read\_a\_year <- function(y){

url\_stub <- "https://www.treasury.gov/resource-center/data-chart-center/interest-rates/pages/TextView.aspx?data=yieldYear&year=XXXX"

url <- gsub("XXXX", y, url\_stub)

yield\_page <- read\_html(url)

yield\_data <- as\_tibble(html\_table(yield\_page, fill = TRUE)[[2]] )

return(yield\_data)

}

# bring in the data one year at a time.

yields\_l <- lapply(1990:2019, read\_a\_year)

# clean and reshape the data with some handy variables for charting

periods <- tibble(

period = paste(c(1,2,3,6,1,2,3,5,7,10,20,30), rep(c("mo", "yr"), c(4,8))),

period\_n = c(30, 60, 90, 180, 365.25 \* c(1,2,3,5,7,10,20,30))

)

yields <- do.call("rbind", yields\_l) %>%

mutate(Date = as.Date(Date, format = c("%m/%d/%y"))) %>%

gather(period, value, -Date) %>%

mutate(value = suppressWarnings(as.numeric(value))) %>%

left\_join(periods, by = "period") %>%

group\_by(Date) %>%

mutate(ratio\_5\_30 = value[period == "30 yr"] / value[period == "5 yr"],

yield\_av = mean(value, na.rm = TRUE, tr = 0.2),

yield\_30yr = value[period == "30 yr"],

yield\_3mo = value[period == "3 mo"],

diff303 = yield\_30yr - yield\_3mo) %>%

ungroup() %>%

filter(!is.na(value))

#----------------All dates in one chart---------------------

col\_br <- tibble(

lab = c(1990, 2000, 2010),

date = as.Date(paste0(c(1990, 2000, 2010), "-03-01"))

) %>%

mutate(date\_n = as.numeric(date))

p2 <- yields %>%

# filter(Date < as.Date("1990-01-31")) %>%

ggplot(aes(x = period\_n, y = value, group = Date, colour = as.numeric(Date))) +

geom\_path(alpha = 0.1) +

scale\_colour\_viridis\_c("",

breaks = pull(col\_br, date\_n),

labels = pull(col\_br, lab)) +

scale\_y\_continuous("Treasury yield curve rate") +

scale\_x\_continuous("", breaks = periods[c(10:12), ]$period\_n,

labels = periods[c(10:12), ]$period) +

labs(caption = the\_caption)

print(p2)

#--------------animated MP4 version----------

d <- yields #%>%

filter(Date < as.Date("1991-02-28"))

a <- d %>%

ggplot(aes(x = period\_n, y = value)) +

geom\_segment(data = distinct(d, Date, yield\_3mo, yield\_30yr),

x = 90, xend = 10958, aes(y = yield\_3mo, yend = yield\_30yr),

colour = "grey50") +

geom\_line(size = 1.5, aes(colour = diff303)) +

scale\_y\_continuous("Treasury yield curve rate") +

scale\_x\_continuous("", breaks = periods[c(10:12), ]$period\_n,

labels = periods[c(10:12), ]$period) +

scale\_colour\_viridis\_c("Premium for long term lending:\n30 year yield minus 3 month yield",

option= "A", direction = -1) +

labs(title = "US Treasury Yield Curve Rates, 1990 to 2019",

subtitle = 'Date: {frame\_time}',

caption = the\_caption) +

transition\_time(Date)

# Save the frames in the file system and then manually knit into an animation, because

# there are so many and so large that I like to keep control of the two steps:

dir.create("tmp")

res <- 150

animate(a, nframes = length(unique(d$Date)) \* 3, dev = "png", fps = 30,

type = "cairo-png", antialias = "subpixel",

width = 6 \* res, height = 4.3 \* res, res = res,

renderer = file\_renderer(dir = "tmp", overwrite = TRUE))

od <- setwd("tmp")

system("ffmpeg -i gganim\_plot%04d.png -pix\_fmt yuv420p -s 900x646 -c:v libx264 -r 30 movie.mp4")

setwd(od)

#----------------traditional time series version-----------

p3 <- yields %>%

mutate(period = fct\_reorder(period, period\_n)) %>%

ggplot(aes(x = Date, y = value, colour = period)) +

geom\_line() +

scale\_colour\_viridis\_d(option = "C") +

labs(x = "", colour = "Term", y = "Treasury yield curve rate",

caption = the\_caption) +

ggtitle("US Treasury Yield Curve Rates, 1990 to 2019")

print(p3)

#-----------------Gif version for Twitter------------

# shorter version with just one main observation per month

d2 <- yields %>%

mutate(mon = month(Date),

yr = year(Date)) %>%

group\_by(mon, yr, period\_n) %>%

summarise(value = mean(value),

yield\_3mo = mean(yield\_3mo),

yield\_30yr = mean(yield\_30yr),

diff303 = mean(diff303)) %>%

ungroup() %>%

mutate(Date = as.Date(paste(yr, mon, 15, sep = "-"), format = "%Y-%m-%d"))

a2 <- d2 %>%

ggplot(aes(x = period\_n, y = value)) +

geom\_segment(data = distinct(d, Date, yield\_3mo, yield\_30yr),

x = 90, xend = 10958, aes(y = yield\_3mo, yend = yield\_30yr),

colour = "grey50") +

geom\_line(size = 1.5, aes(colour = diff303)) +

scale\_y\_continuous("Treasury yield curve rate") +

scale\_x\_continuous("", breaks = periods[c(10:12), ]$period\_n,

labels = periods[c(10:12), ]$period) +

scale\_colour\_viridis\_c("Premium for long term lending:\n30 year yield minus 3 month yield",

option= "A", direction = -1) +

labs(title = "US Treasury Yield Curve Rates, 1990 to 2019",

subtitle = 'Date: {frame\_time}',

caption = the\_caption) +

transition\_time(Date)

res <- 150

animate(a, nframes = length(unique(d2$Date)) \* 6, dev = "png", fps = 15,

type = "cairo-png", antialias = "subpixel",

width = 6 \* res, height = 4.3 \* res, res = res)

anim\_save("0151-yield-anim150.gif")

thankr::shoulders() %>%

mutate(maintainer = str\_squish(gsub("<.+>", "", maintainer)),

maintainer = ifelse(maintainer == "R-core", "R Core Team", maintainer)) %>%

group\_by(maintainer) %>%

summarise(`Number packages` = sum(no\_packages),

packages = paste(packages, collapse = ", ")) %>%

arrange(desc(`Number packages`)) %>%

knitr::kable() %>%

clipr::write\_clip()