Week 4: Term Structure and Splines

Imagine this. . .

So. . .

- Your organization is about to raise several hundred million dollars in the European Union.
- Negative interest rates abound in the markets as political events and central bankers vie for control of the euro and its relationship to major currencies.
- Your task is to help the CFO understand how the term structure of interest rates might change and thus impact the amount and pricing of the funds your organization is about to raise.

What do you think?

Just two questions

1. What is the term structure of interest rates?
2. Why would the term structure matter to the CFO?

Some thoughts. . . and answers

1. "Term" refers to the maturity of a debt instrument, the "loan" we get to buy a house. "Term structure" is the schedule of interest rates posted for each maturity. The "term structure" is also known as the "yield curve."
2. "So what?" you ask. If the term on your loan (or your company's bonds) rises, a higher yield might just convince investors (like you or TIAA/CREF or other cash-rich investor) to wait longer for a return of the principal they lend to you.
3. Higher yields mean also that you would have to price the bond at a higher coupon rate. This means more income is paid out to lenders and bondholders than to management, staff, and shareholders.

Purpose

- Conceive a model of bond prices that reflects the underlying dynamics of the term structure of interest rates.
- Use "regression splines" to interpolate and extrapolate values of forward rates from the term structure.
- Implement the model using financial theory and estimation with live data.

Process

- Start with statistical definitions and financial models of bond prices.
- Move into the working example of US Treasury STRIPs (zero-coupon bonds) and explore the possibilities.
- Build a financially informed model of the term structure of forward empirical rates.
- Estimate the model with nonlinear least squares.
- Compare and contrast two competing model specifications.

Product

- Extensible quadratic spline model of the forward curve
- R skills to explore data with plots and translate theoretical forward curve model into an estimable model

Whitman
SCHOOL of MANAGEMENT
SYRACUSE UNIVERSITY

MBA@SYRACUSE

The bond

A typical bond is a financial instrument that pays fixed cash flows for the maturity of the bond with a repayment of the principal (notional or face value) of the bond at maturity. In symbols,

$$V = \sum_{t=0}^{mT} \frac{cP}{(1+y/m)^t} + \frac{P}{(1+y/m)^{mT}},$$

where V is the present value of the bond, T is the number of years to maturity, m is the number of periods cash flows occur per year, y is the bond yield per year, c is the coupon rate per year, and P is the bond's principal (notional or face value).

Using the idea of an annuity that pays $(c/m)P$ per period for mT periods and P at maturity period mT we get a nice formula for the present value sum of coupon payments:

$$V = (c/m)P\left(\frac{1}{y/m} - \frac{1}{(y/m)(1+y/m)^{mT}}\right) + \frac{P}{(1+y/m)^{mT}}.$$

From a financial engineering point of view

This is the same as constructing a position that is long a perpetuity that pays a unit of currency starting the next compounding period and short another perpetuity that starts to pay a unit of currency at the maturity of the bond plus one compounding period.

A typical bond pays coupons twice a year, so $m = 2$. If the bond's maturity in years is 10 years, then $mT = 10x2 = 20$ compounding periods. We will assume that there is no accrual of interest as of the bond's valuation date for the moment.

```
c <- 0.05
P <- 100
y <- c(0.04, 0.05, 0.06)
m <- 2
T <- 10
(V <- (c/m) * P * (1/(y/m) - 1/((y/m) *
    (1 + (y/m))^(m * T))) + P/(1 + (y/m))^(m *
    T))
```

```
## [1] 108.17572 100.00000  92.56126
```

A quick question for you and then, try this

The quick question

1. If the coupon rate is greater than the yield, why is the price greater than par value?

Behold the negative interest rate

2. Negative interest rates abound, so set `y <- c(-0.02, -0.01, 0.00, .01, .02)` and recalculate the potential bond values.

Thinking. . .

Answers

A quick answer

The bond pays out at a rate greater than what is required in the market. Buyers pay more than par to make up the difference.

More Results

```r
c <- 0.05
P <- 100
y <- c(-0.02, -0.01, 0, 0.01, 0.02)
m <- 2
T <- 10
(V <- (c/m) * P * (1/(y/m) - 1/((y/m) *
    (1 + (y/m))^(m * T))) + P/(1 + (y/m))^(m *
    T))
```

```
## [1] 177.9215 163.2689      NaN 137.9748 127.0683
```

Why a `NAN`? Because we are dividing by '0'!

By the way, these are some hefty premia...

Check out this article on negative yields...

http://www.ft.com/cms/s/0/312f0a8c-0094-11e6-ac98-3c15a1aa2e62.html

Here's the kicker

- The yield in this model is an average of the forward rates the markets use to price future cash flows *during future periods of time*.
- Question: How can we incorporate a changing forward rate into a model of bond prices?
- For starters: use the past as a guide for the future and calibrate rates to a curve of rates versus maturities.
- Use the regression spline model to find rates at any maturity.
- Then: use these interpolated (even extrapolated) rates as building blocks in the valuation of bonds.

Whitman
SCHOOL of MANAGEMENT
SYRACUSE UNIVERSITY

MBA@SYRACUSE

What's a spline?

A *spline* is a function that is constructed piece-wise from polynomial functions. Very mathematical sounding. But imagine the polynomials are pieces of the term structure of interest rates...

- Each section is marked off by a *knot* at a location in the data.
- Knots are most commonly placed at quantiles to put more knots where data is clustered close together.
- A different polynomial function is estimated for each range and domain of data between each knot: this is the spline.
- The spline function `spline` in `R` will allow us to interpolate between knots and even extrapolate beyond a knot.

Now we will use some finance to build a polynomial regression spline out of US Treasury zero-coupon data.

What's a polynomial?

An expression that looks like this:

$$f(x) = a_0 x^0 + a_1 x^1 + a_2 x^2 + ... + a_p x^p,$$

where the a's are constant coefficients, and $x^0 = 1$ and $x^1 = x$.

Polynomials and you

- If $p = 0$, then we have a *constant* function.
- If $p = 1$, we have a *linear* function.
- If $p = 2$, we have a *quadratic* function.
- If $p = 3$, we have a *cubic* function, and so on...

We will definitely be using a *cubic* function

Motive and Opportunity

Investors give an issuer today the present value, the bond "price" P_T of receiving back the 100% of the face value of a zero-coupon bond at maturity year (or fraction thereof), T.

The price of a zero coupon bond, quoted in terms of the percentage of face value, is this expression for discrete compounding at rate y_T^d (say, monthly you get a percentage of last month's balance):

$$P_T = \frac{100}{(1 + y^d(T))^T}.$$

This translates into continuous compounding (you get a percentage of the last nanosecond's balance at the end of this nanosecond... as the nanoseconds get ever smaller...) as

$$P_T(\theta) = 100 exp(-y_T T).$$

This is the present value of receiving a cash flow of 100% of face value at maturity. If the bond has coupons, we can consider each of the coupon payments as a mini-zero bond.

Suppose we have forward rates $r(t, \theta)$, where t moves from the valuation date 0 to the maturity date T of a zero coupon bond, and θ contains all of the information we need about the shape of r across maturities. We can estimate the forward curve from bond prices $P(T)$ of the Tth maturity with

$$-\frac{\Delta log(P(T_i))}{\Delta T_i} = -\frac{log(P(T_i)) - log(P(T_{i-1}))}{T_i - T_{i-1}}.$$

The Δ stands for the difference in one price or maturity i and the previous price and maturity $i - 1$. The $log()$ function is the natural logarithm. An example will follow.

The *yield* is then the *average of the forward rates* from date 0 to date T of a zero bond. We use the integral, which amounts to a cumulative sum, to compute this average:

$$y_T(\theta) = T^{-1} \int_0^T r(t, \theta) dt.$$

The numerator is the cumulative sum of the forward rates for each maturity up to the last maturity T. In this expression, *rdt* is the forward rate across a small movement in maturity. The denominator is the number of maturity years T.

Try this out . . .

Load these lines of R into the RStudio console:

```
maturity <- c(1, 5, 10, 20, 30)  # in years
price <- c(99, 98, 96, 93, 89)  # in percentage of face value
```

A. Now let's experiment on these zero-coupon prices with their respective maturities:

1. Calculate the `log(price)/100`. Then find the forward rates using using

```
(forward <- -diff(log(price))/diff(maturity))
```

2. Compare `log(price)` with `price`.
3. What does the forward rate formula indicate? What would we use it for?

B. Find the yield-to-maturity curve and recover the bond prices using

```
(forward.initial <- -log(price[1]/100))
(forward.integrate <- c(forward.initial,
    forward.initial + cumsum(forward *
        diff(maturity))))
# a rolling integration of rates
# across maturities
(price <- 100 * exp(-forward.integrate))
# present value of receiving 100% of
# face value
```

1. What is the interpretation of the `forward.integrate` vector?
2. What happened to the first bond price?
3. Did we recover the original prices?

Thinking…

We want some answers!

For question A we ran the forward rates. Let's run the natural logarithm of `price`:

```r
(forward <- -diff(log(price/100))/diff(maturity))
```

```
## [1] 0.002538093 0.004123857 0.003174870 0.004396312
```

```r
(-log(price/100))
```

```
## [1] 0.01005034 0.02020271 0.04082199 0.07257069 0.11653382
```

A.1

`-log(price/100)` seems to give us the yield-to-maturity directly. But do look at `-diff(log(price))`:

```
(-diff(log(price/100)))
```

```
## [1] 0.01015237 0.02061929 0.03174870 0.04396312
```

and these look like rates, because they are. They are the continuous time version of a percentage change, or growth, rate from one maturity to another maturity.

A.2

This is the interpretation: interest rates are simply rates of change of present values relative to how much time they cover in maturity.

```
(-diff(price/100)/(price[-length(price)]/100))
```

```
## [1] 0.01010101 0.02040816 0.03125000 0.04301075
```

… similar, but not quite the same. Note the use of the indexing of price to eliminate the last price, since we want to compute:

$$\frac{P(T_i) - P(T_{i-1})}{P(T_{i-1})}$$

Running the code for question B we get:

```
forward.initial <- -log(price[1]/100)
(forward.integrate <- c(forward.initial,
    forward.initial + cumsum(forward *
        diff(maturity))))
```

```
## [1] 0.01005034 0.02020271 0.04082199 0.07257069 0.11653382
```

```
# a rolling integration of rates
# across maturities
(price <- 100 * exp(-forward.integrate))
```

```
## [1] 99 98 96 93 89
```

```
# present value of receiving 100% of
# face value
```

B answers

1. Yields are the accumulation of forward rates. Thus the use of the cumulative sum as a discrete version of the integral. Rates add up (instead of multiply: nice feature) when we use `log` and `exp` to do our pricing.
2. The first forward rate is just the discount rate on the 1-year maturity bond stored in `forward.initial`.
3. All bond prices are recovered.

And now some more about that bond yield

We restate the definition of the price of a zero-coupon bond. The price of a zero-coupon bond quoted in percentage of face value is this expression for discrete compounding (say, monthly you get a percentage of last month's balance):

$$P_T = \frac{100}{(1 + y^d(T))^T}$$

This translates into continuous compounding (you get a percentage of the last nanosecond's balance at the end of this nanosecond. . .) as

$$P_T(\theta) = 100 exp(-y_T T) = 100 exp(\int_0^T r(t, \theta) dt)$$

This is the present value of receiving a cash flow of 100% of face value at maturity. If the bond has coupons we can consider each of the coupon payments as a mini-zero bond.

The term with the fancy $\int$ symbol is a nanosecond by nanosecond way of summing the forward rates across the maturity of the bond. Equating the two statements and solving for y_T, the continuously compounded yield we get:

$$P_T = \frac{100}{(1 + y(T)_d)^T} = 100 exp(-y_T T).$$

Rearranging with some creative algebra: 100 drops out, and remembering that $exp(-y_T t) = 1/exp(y_T T)$, we have:

$$exp(y_T T) = (1 + y(T)_d)^T.$$

Then, taking logarithms of both sizes

$$exp(y_T T)^{-T} = (1 + y(T)_d)^{T-T} = (1 + y(T)_d).$$

Using the facts that $log(exp(x)) = x$ and $log(x^T) = Tlog(x)$ (very convenient):

$$log(exp(y_T T)^{-T}) = y_T = log(1 + y(T)_d).$$

We now wipe our brows...

```r
y.T.d <- 0.2499  # Usury rate in NYS; d = discrete, T = maturity in years
(y.T <- log(1 + y.T.d))
```

```
## [1] 0.2230635
```

```r
(y.T.d <- exp(y.T) - 1)
```

```
## [1] 0.2499
```

. . . always less than the discrete rate because there are so many more continuous compounding periods.

Whitman
SCHOOL of MANAGEMENT
SYRACUSE UNIVERSITY

MBA@SYRACUSE

Parameter revelation

Now for the heretofore mysterious θ. We suppose the forward rate is composed of a long-term constant, θ_0; a sloping term with parameter θ_1; a more shapely, even "humped" term with parameter θ_2; and so on for p polynomial terms:

$$r(t, \theta) = \theta_0 + \theta_1 t + \theta_2 t^2 + ... + \theta_p t^p$$

Thus the antiderivative

We recall (perhaps not too painfully!) from calculus (not to worry!) that when we integrate a variable to a power (the antiderivative), we raise the variable to the next power (and divide the term by that next power):

$$\int_0^T r(t,\theta)dt = \theta_0 T + \theta_1 \frac{T^2}{2} + \theta_2 \frac{T^3}{3} + ... + \theta_p \frac{T^{p+1}}{p+1}.$$

We then can estimate the yield curve (and then the zero-coupon bond price) using:

$$y_T(\theta) = \theta_0 + \theta_1 \frac{T}{2} + \theta_2 \frac{T^2}{3} + ... + \theta_p \frac{T^p}{p+1}.$$

Before we go any further...

Let's get some data and try this out...

Here is some (very old) data from US Treasury STRIPS (a.k.a. for "Separate Trading of Registered Interest and Principal of Securities").

```
# Be sure to set your working
# directory.
dat = read.table("data/strips_dec95.txt",
    header = TRUE)
```

A question or two

1. What does the data look like?
2. Anything else?

Thinking. . .

An answer or two

Run this code chunk:

```r
head(dat, n = 3)
```

```
##        T  price
## 1 0.1260 99.393
## 2 0.6219 96.924
## 3 1.1260 94.511
```

```r
names(dat)
```

```
## [1] "T"     "price"
```

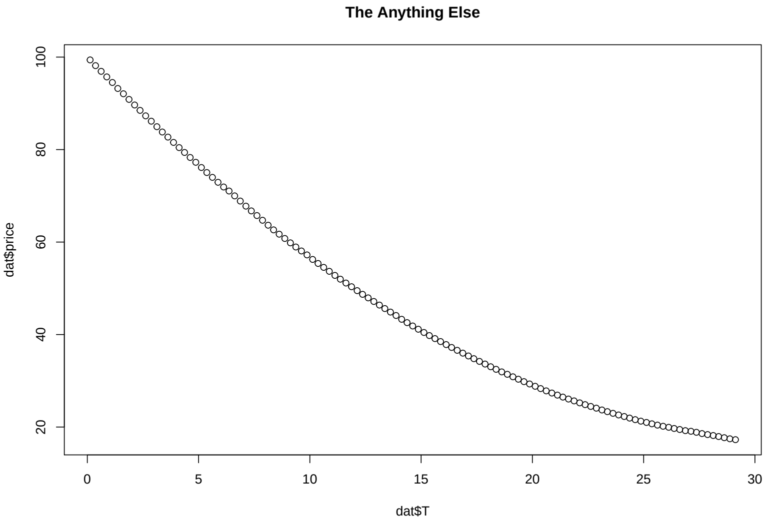
```r
dat = dat[order(dat$T), ]
```

We read in a table of text with a header, look at the first few observations, order the data by maturity T_i.

For more information about STRIPS go to

`https://www.newyorkfed.org/aboutthefed/fedpoint/fed42.html`

```r
plot(dat$T, dat$price, main = "The Anything Else")
```



The empirical forward curve

We estimate the empirical forward curve using:

$$-\frac{\Delta log(P(T_i))}{\Delta T_i} = -\frac{log(P(T_i)) - log(P(T_{i-1}))}{T_i - T_{i-1}},$$

where P is the bond price and T_i is ith maturity,

```
t = seq(0, 30, length = 100)
emp = -diff(log(dat$price))/diff(dat$T)
```

The equation is translated into `R` with the `diff` function. We will use vector `t` later when we plot our models of the forward curve. Definitely view it from the console and check out `??seq` for more information about this function.

Try this next

Plot the empirical forward curves using this line of code:

```
plot(dat$T[2:length(dat$T)], emp, ylim = c(0.025,
    0.075), xlab = "maturity", ylab = "empirical forward rate",
    type = "b", cex = 0.75, lwd = 2,
    main = "US Treasury STRIPs - 1995")
```

Try to answer these questions before moving on:

1. What exactly will `dat$T[2:length(dat$T)]` do when executed?
2. What effect will `ylim`, `lwd`, `xlab`, `ylab`, `type`, `cex`, `main` have on the plot?
3. Is there an break in the curve? Write the plot command to zoom in on maturities from 10 to 20 years.

Thinking. . .

You get these results:

1. What exactly will `dat$T[2:length(dat$T)]` do when executed?

```r
length(dat$T)
```

```
## [1] 117
```

```r
head(dat$T[2:length(dat$T)])
```

```
## [1] 0.3699 0.6219 0.8740 1.1260 1.3699 1.6219
```

We retrieve the `T` vector from the `dat` data frame using `dat$T`. Then `length` returns the number of maturities in the `dat` data frame. `dat$T[2:length(dat$T)]` truncates the first observation. Why? Because we differenced the prices to get forward rates and we need to align the forward rates with their respective maturities T_i.

Result 2: plot parameters

Parameters

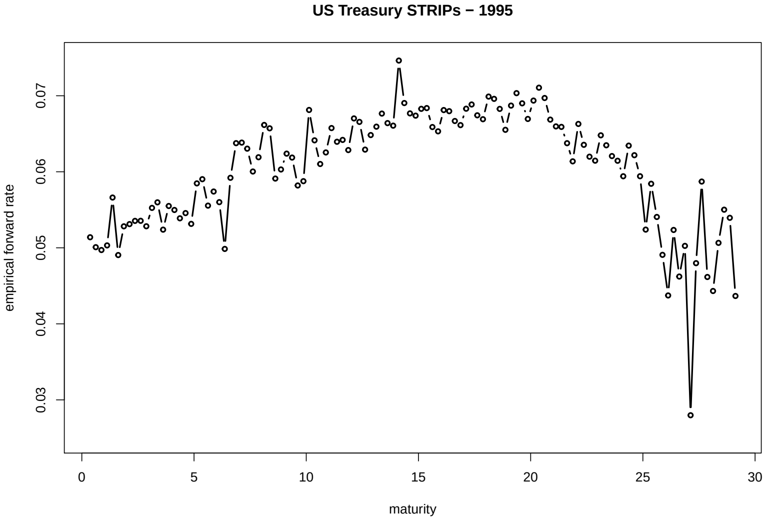
- `ylim` zooms in the y-axis data range
- `lwd` changes the line width
- `xlab` specifies the x-axis label
- `ylab` specifies the y-axis label
- `type` specifies the line
- `cex` changes the scaling of text and symbols
- `main` specifies the plot title.

Go to

`http://www.statmethods.net/advgraphs/parameters.html`
to find more information and the answer to the zoom-in question.

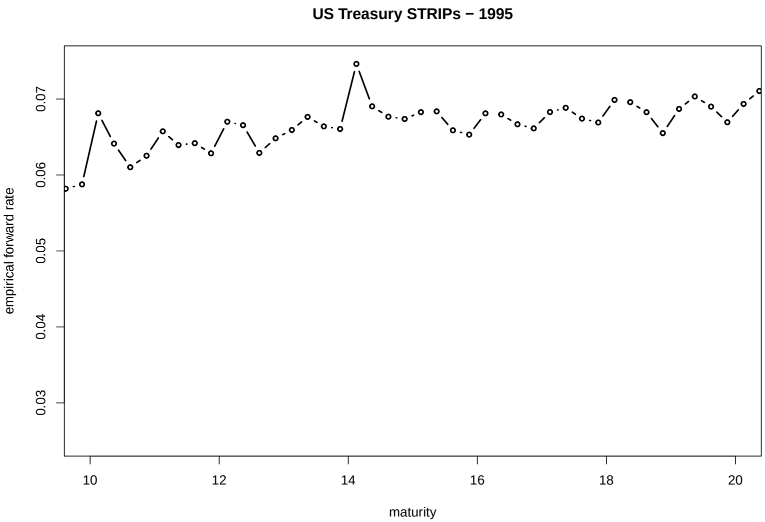
The plot

```r
plot(dat$T[2:length(dat$T)], emp, ylim = c(0.025,
    0.075), xlab = "maturity", ylab = "empirical forward rate",
    type = "b", cex = 0.75, lwd = 2,
    main = "US Treasury STRIPS - 1995")
```


US Treasury STRIPs − 1995
empirical forward rate
maturity
0.03
0.04
0.05
0.06
0.07
0
5
10
15
20
25
30

Now the zoom in:

```
plot(dat$T[2:length(dat$T)], emp, xlim = c(10,
    20), ylim = c(0.025, 0.075), xlab = "maturity",
    ylab = "empirical forward rate",
    type = "b", cex = 0.75, lwd = 2,
    pin = c(3, 2), main = "US Treasury STRIPS - 1995")
```



At $T_i = 14$ there appears to be an outlier. More importantly there is a break at $T_i = 15$. This is a natural *knot*.

Whitman

SCHOOL of MANAGEMENT

SYRACUSE UNIVERSITY

MBA@SYRACUSE

Back to our story

Let's add a kink in the yield curve that allows two different quadratic functions: one before the kink, and one after the kink. Let the kink be a knot k at $T_i = 15$. We evaluate a knot as 0 if the maturities $T_i - k < 0$, and equal $T - k$ if $T - k > 0$. We write this as $(T - k)_+$. Let's now drop the knot into our integral:

$$\int_0^T r(t,\theta)dt = \theta_0 T + \theta_1 \frac{T^2}{2} + \theta_2 \frac{T^3}{3} + \theta_2 \frac{(T-15)_+^3}{3}$$

We can divide by T to get the yield. But to calculate the bond price we have to multiply the yield by the bond maturity T, so the bond price is then:

$$P_T(\theta) = 100 exp[-(\theta_0 T + \theta_1 \frac{T^2}{2} + \theta_2 \frac{T^3}{3} + \theta_3 \frac{(T-15)_+^3}{3})]$$

A "phew!" is in order. Now to estimate the thetas.

Gee, that's very nonlinear of you. . .

Yes, the bond price is nonlinear in the θ parameters. Our statistical job now is to find a set of θ such that the difference between the actual bond prices P and our clever model of bond prices (long equation that ended the last slide) is very small in the sense of the sum of squared differences ("errors"). We thus find θ that minimizes

$$\sum_{i=0}^{N}[(P(T_i) - P(T_i, \theta)]^2$$

To find the best set of θ we will resort to a numerical search using the `R` function `nls`, for nonlinear least squares.

Try this out as we get down to business

Back to the data: we now find the θs. The logical `(T>k)` is 1 if `TRUE` and 0 if `FALSE`. We put the `R` version of the bond price into the `nls` function, along with a specification of the data frame `dat` and starting values.

Run these statements to compute the (nonlinear) regression of the term structure:

```
fit.spline = nls(price ~ 100 * exp(-theta_0 *
    T - (theta_1 * T^2)/2 - (theta_2 *
    T^3)/3 - (T > 15) * (theta_3 * (T -
    15)^3)/3), data = dat, start = list(theta_0 = 0.047,
    theta_1 = 0.0024, theta_2 = 0, theta_3 = -7e-05))
```

Just a couple of questions:

1. What are the dependent and independent variables?
2. Which parameter measures the sensitivity of forward rates to the *knot*?

Results

Let's look at our handiwork using `kable` from the `knitr` package:

```
library(knitr)
kable(summary(fit.spline)$coefficients,
    digits = 4)
```

	Estimate	Std. Error	t value	Pr(>|t|)
theta_0	0.0495	1e-04	536.5180	0
theta_1	0.0016	0e+00	51.5117	0
theta_2	0.0000	0e+00	-13.6152	0
theta_3	-0.0002	0e+00	-30.6419	0

```
(sigma <- (summary(fit.spline)$sigma)^0.5)
```

```
## [1] 0.2582649
```

All coefficients are significant and a standard error to compare with other models.

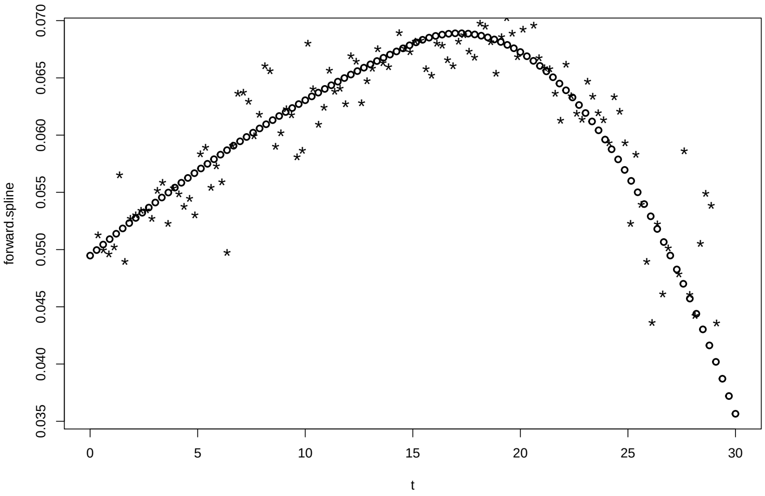
Build a spline

Let's now produce a plot of our results using a sequence of maturities `t`.

First we parse the coefficients from the spline fit and build the spline prediction. Here we construct the forward rate spline across `t` maturities.

```
coef.spline <- summary(fit.spline)$coef[,
    1]
forward.spline <- coef.spline[1] + (coef.spline[2] *
    t)
+(coef.spline[3] * t^2)
+(t > k) * (coef.spline[4] * (t - 15)^2)
```

Now for the plot.



forward.spline
0.070
0.065
0.060
0.055
0.050
0.045
0.040
0.035
0
5
10
15
20
25
30
t

Your turn (again)

Try these as we finish this segment

1. Compare the quadratic spline we just constructed with a pure quadratic polynomial. Simply take the knot out of the `nls` formula and rerun. Remember that "quadratic" refers to the polynomial degree p of the assumed structure of the forward rate $r(t, \theta)$.
2. Plot the data against the quadratic spline and the quadratic polynomial.
3. Interpret financially the terms in θ_0, θ_1, and θ_2.

Thinking. . .

Results from your turn (again)

Run this code: remembering that a "cubic" term represents the integration of the "quadratic" term in

$$r(t, \theta) = \theta_0 + \theta_1 t + \theta_2 t^2$$

```
fit.quad <- nls(price ~ 100 * exp(-theta_0 *
    T - (theta_1 * T^2)/2 - (theta_2 *
    T^3)/3), data = dat, start = list(theta_0 = 0.047,
    theta_1 = 0.0024, theta_2 = 0))
```

This estimate gives us *one* quadratic function through the cloud of zero-coupon price data.

More results...

```
knitr::kable(summary(fit.quad)$coefficients,
    digits = 4)
```

	Estimate	Std. Error	t value	Pr(>|t|)
theta_0	0.0475	2e-04	239.0168	0
theta_1	0.0024	1e-04	46.5464	0
theta_2	-0.0001	0e+00	-33.0016	0

All conveniently significant.

```
(sigma <- (summary(fit.quad)$sigma)^0.5)
```

```
## [1] 0.4498518
```

This model produces a higher standard deviation of error than the quadratic spline.

Run this code for a plot. First some calculations based on the estimations we just performed.

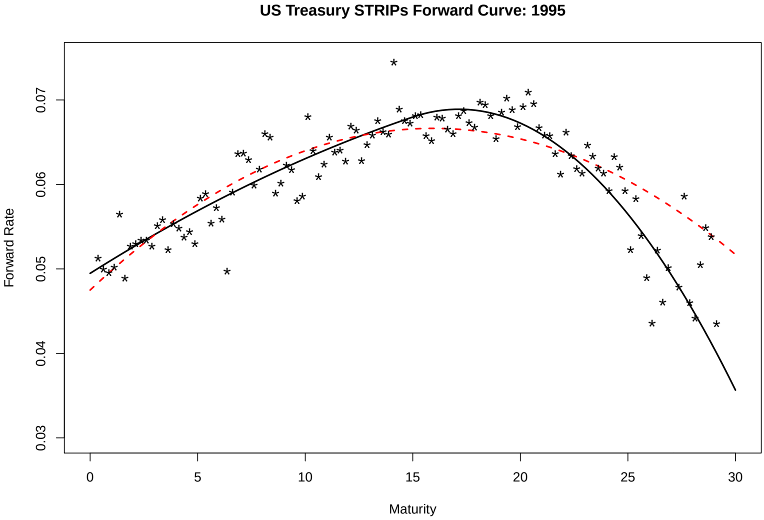
```
coef.spline <- summary(fit.spline)$coef[,
    1]
forward.spline <- coef.spline[1] + (coef.spline[2] *
    t)
+(coef.spline[3] * t^2)
+(t > 15) * (coef.spline[4] * (t - 15)^2)
coef.quad <- summary(fit.quad)$coef[,
    1]
forward.quad <- coef.quad[1] + (coef.quad[2] *
    t) + (coef.quad[3] * t^2)
```

Then the plot itself.

```
plot(t, forward.spline, type = "l", lwd = 2,
    ylim = c(0.03, 0.075), xlab = "Maturity",
    ylab = "Forward Rate", main = "US Treasury STRIPs Forward Curve: 1995")
lines(t, forward.quad, lty = 2, lwd = 2,
    col = "red")
points(dat$T[2:length(dat$T)], emp, pch = "*",
    cex = 1.5)
```

Let's look at the results.

Plot again



Look at what data the quadratic forward curve misses!

There seem to be three components in the forward curve:

- Constant
- Slope
- Curved (affectionately called “humped”)

Parameter interpretation

- θ_0 is independent of maturity and thus represents the long-run average forward rate.
- θ_1 helps to measure the average sensitivity of forward rates to a change in maturity.
- θ_2 helps to measure the maturity risk of the forward curve for this instrument.

Whitman
SCHOOL *of* MANAGEMENT
SYRACUSE UNIVERSITY

MBA@SYRACUSE

Just one more thing

Here is the infamous **SO WHAT?!**, especially after all of the work we just did.

Here's the "so what"

1. Suppose you just bought a 10 year maturity zero-coupon bond to satisfy collateral requirements for workers' compensation in the (great) State of New York.
2. The forward rate has been estimated as:

$$r(t) = 0.001 + 0.002t - 0.0003(t-7)_+$$

3. In 6 months you then exit business in New York State, have no employees that can claim workers' compensation, sell the 10 year maturity zero-coupon bond, AND the forward curve is now

$$r(t) = 0.001 + 0.0025t - 0.0004(t-7)_+$$

4. How much would you gain or lose on this transaction at your exit?

Some bond maths to (re)consider:

1. The forward rate is the rate of change of the yield-to-maturity
2. This means we integrate (i.e., take the cumulative sum of) forward rates to get the yield
3. The cumulative sum would then be some maturity times the components of the yield curve adjusted for the slope of the forward curve (the terms in t).
4. This adjustment is just one-half $(1/2)$ of the slope term.

Here are the calculations:

```
maturity.now <- 10
maturity.6m <- 9.5
(yield.now <- 0.001 * maturity.now +
    0.002 * maturity.now^2/2 - 3e-04 *
    (maturity.now > 7)^2/2)
```

```
## [1] 0.10985
```

```
(yield.6m <- 0.001 * maturity.6m + 0.0025 *
    maturity.6m^2/2 - 4e-04 * (maturity.6m >
    7)^2/2)
```

```
## [1] 0.1221125
```

```r
(bond.price.now <- exp(-yield.now))
```

```
## [1] 0.8959685
```

```r
(bond.price.6m <- exp(-yield.6m))
```

```
## [1] 0.8850488
```

```r
(return = bond.price.6m/bond.price.now -
    1)
```

```
## [1] -0.01218762
```

```r
2 * return  ### annualized return
```

```
## [1] -0.02437524
```

It appears that we lost something in this exit from NYS.

Whitman

SCHOOL of MANAGEMENT

SYRACUSE UNIVERSITY

MBA@SYRACUSE

Drama? . . . and a light debrief

The quadratic forward curve seems to dramatically underfit the maturities higher than 15 years.

- We learned a lot of bond maths: prices, yields, forward curves.
- We just built two models of the forward curve.
- We then implemented these models in `R` with live data.
- We also learned some math and some more `R`.

To prepare for the live session:

1. What are the top 3 key learnings for you from this week?
2. What pieces are still a mystery?
3. What parts would you like more practice on?
4. Please review the assignment for this week. What questions would you like addressed in the live session?

Thanks!