Interest Rate Modeling

John Williams

November 19, 2015

Contents

Introduction

The instantaneous forward interest rate of a zero-coupon bond is defined as *f(t,T)* and modeled below by nine methods: Nelson-Siegel, polynomial, Svensson, Vasicek, Cox-Ingersoll-Ross, local polynomial, cubic B-spline, and smoothing spline.

The price *P(t,T)* of a zero-coupon bond is then:

$$P(t,T) = \exp\left[-\int_t^T f(t,u)du\right]$$

The spot rate or yield *R(t,T)* of a zero-coupon bond is:

$$R(t,T) = \text{-}\frac{log\,P(t,T)}{T\text{-}t}$$

Diablod Li data.txt contains yields of different maturities at each month from 1970 to 2000.

```
library(knitr)
library(ggplot2)
library(minpack.lm)
library(KernSmooth)
library(splines)
library(servr)
```

```
knitr::opts_chunk$set(fig.width = 12, fig.height = 7, fig.path = 'Figs/',
                      warning = FALSE, message = FALSE,
                      scipen = 1, digits = 4)
```

```
maturity <- as.numeric(matrix(scan("./data/Diabold_Li_data.txt",nlines = 1),ncol = 19,byrow = T))
yields <- as.data.frame(matrix(scan("./data/Diabold_Li_data.txt",skip = 1) ,ncol = 19,byrow = T))
yield <- as.numeric(yields[which(yields[,1] == 19900531),])
df <- as.data.frame(cbind(maturity[2:19], yield[2:19]))
names(df) <- c("maturity", "yield")
```

1. Nelson and Siegel's model

Nelson and Siegel's model for instantaneous forward interest rate *f(0,s)*:

$$f(0,s) = \beta_1 + \beta_2 \exp(-\lambda s) + \beta_3 \lambda s \exp(-\lambda s)$$

The spot rate or yield for Nelson and Siegel's model is then:

$$R(0,s) = \beta_1 + \beta_2\left(\frac{1-\exp(-\lambda s)}{\lambda s}\right) + \beta_3\left(\frac{1-\exp(-\lambda s)}{\lambda s} - \exp(-\lambda s)\right)$$

Nelson-Siegel parameter interpretation:

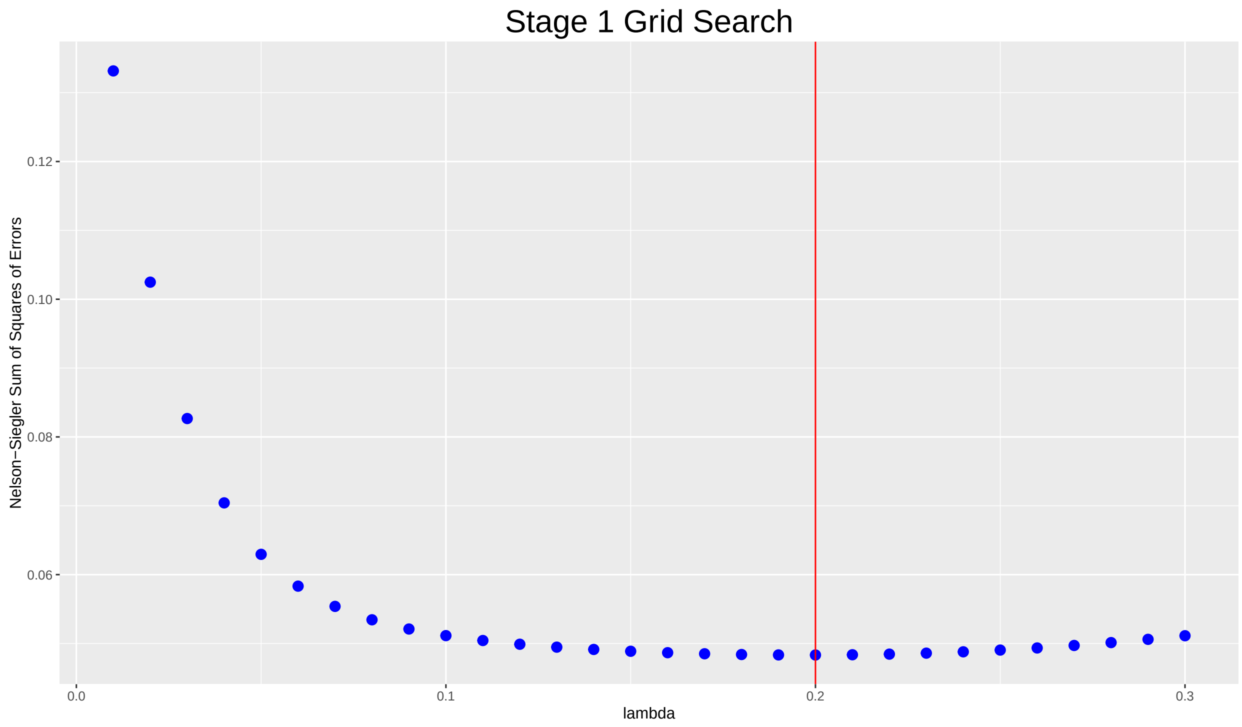
β_1: long-maturity limiting forward rate
β_2: used to accommodate different shapes of zero price curve
β_3: used to accommodate different shapes of zero price curve
$\frac{1}{\lambda}$: time constant measuring how fast the forward rate tends to change with maturity

Two Stage grid search to minimize the sum of squared errors of the Nelson-Siegel model.

Stage 1: Grid of 30 points: 0.01, 0.02, . . . , 0.3 for λ.

```
NS.SSE <- function(lambda, df) {
  z1 <- exp(-lambda*df$maturity)
  z2 <- (1-z1)/(lambda*df$maturity)
  z3 <- z2-z1
  NSdata <- as.data.frame(cbind(df$yield, z2, z3))
  names(NSdata)[1] <- "yield"
  NSmod <- lm(yield ~., data = NSdata )
  return(sum(NSmod$residuals^2))
}
SSEs <-sapply(1:30/100, function(i) NS.SSE(i, df))
```

```
df1 <- as.data.frame(cbind(SSEs, (1:30)/100))
names(df1) <- c("SSE", "lambda")
minSSE <- min(df1$SSE)
lambda.minSSE1 <- df1[which(df1$SSE == minSSE),]$lambda
names(df1) <- c("SSE", "lambda")
ggplot(df1, aes(lambda, SSE)) +
  geom_point(size=3.00, col="blue") +
  theme(plot.title = element_text( size=22)) +
  ggtitle("Stage 1 Grid Search") +
  xlab("lambda") +
  ylab("Nelson-Siegler Sum of Squares of Errors") +
  geom_vline(xintercept = lambda.minSSE1, col = "red")
```

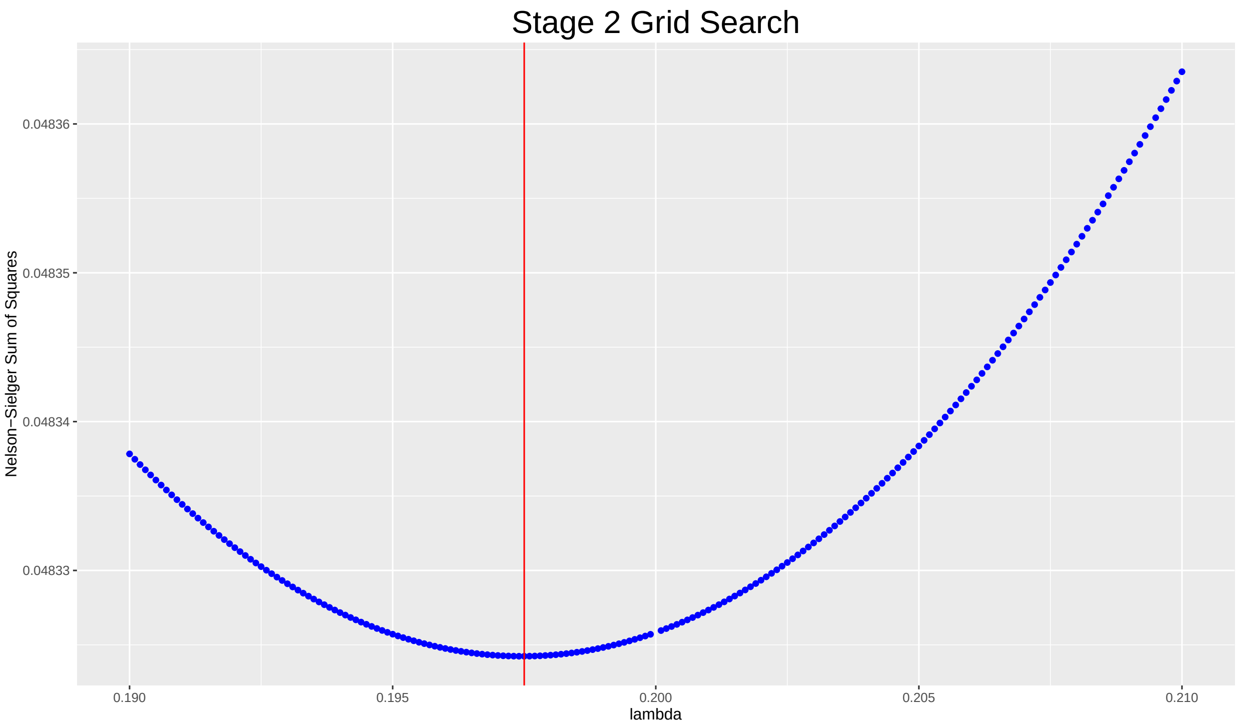


Minimum SSE of stage 1 search is 0.0483258 at λ *= 0.2.*

Stage 2: grid of 199 points around the minimum λ found in stage 1

```
gridVals <- 1:100/10000
lamplus <- lambda.minSSE1 + gridVals
lamneg  <- lambda.minSSE1 - gridVals
lambdas <- c(lamneg, lamplus)
SSEs <-sapply(lambdas, function(i) NS.SSE(i, df))
```

```
df1 <- as.data.frame(cbind(SSEs, lambdas))
names(df1) <- c("SSE", "lambda"); minSSE <- min(df1$SSE)
lambda.minSSE2 <- df1[which(df1$SSE == minSSE),]$lambda
accx <- 0.0001/lambda.minSSE2
names(df1) <- c("SSE", "lambda")
ggplot(df1, aes(lambda, SSE)) +
  geom_point(size=1.50, col="blue") +
  theme(plot.title = element_text( size=22)) +
  ggtitle("Stage 2 Grid Search") +
  xlab("lambda") +
  ylab("Nelson-Sielger Sum of Squares") +
  geom_vline(xintercept = lambda.minSSE2, col = "red")
```



The minimum SSE of the Nelson-Siegler stage 2 search is 0.0483242 at λ *= 0.1975.*

The margin of error of the lambda estimate is +/- 0.05%‘.

2. Nelson-Siegel plot

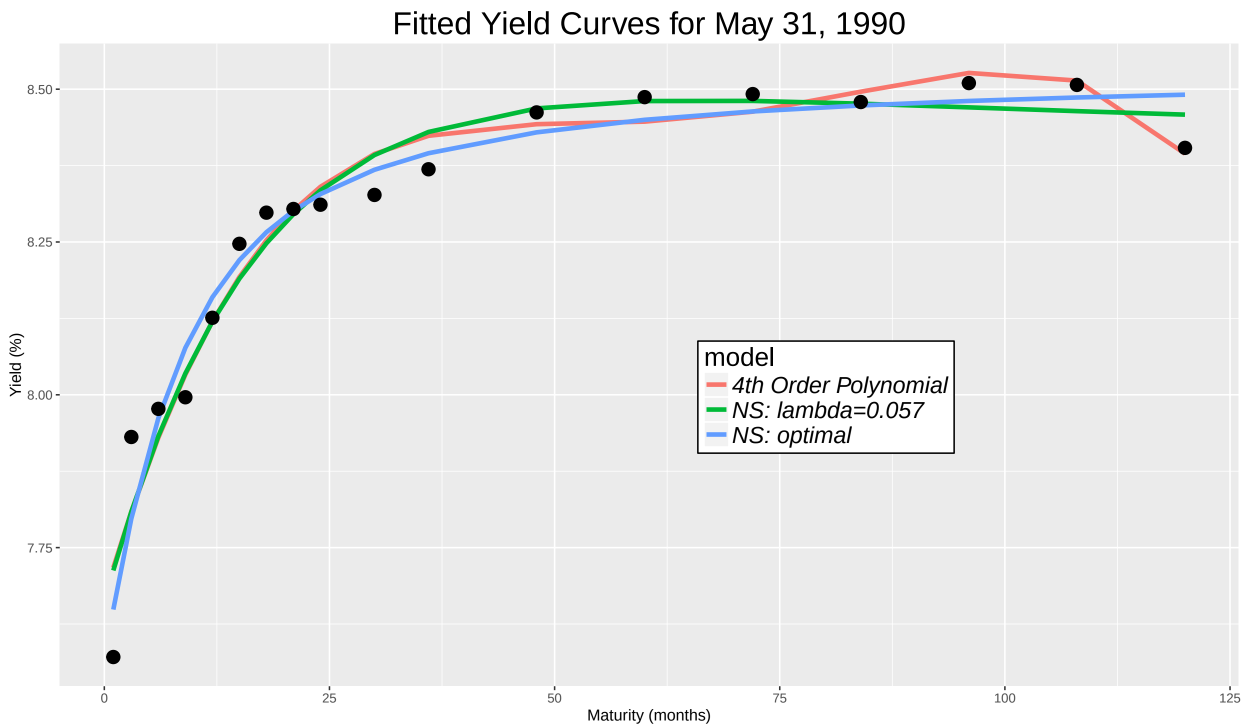
Comparison Nelson-Siegel (NS) curve found above, with an NS curve using $\lambda = 0.057$, and a 4th order polynomial curve.

```
lambda <- lambda.minSSE2
z1 <- exp(-lambda*df$maturity)
z2 <- (1 - z1)/(lambda*df$maturity)
z3 <- z2 - z1
NSdata <- as.data.frame(cbind(df$yield, z2, z3))
names(NSdata)[1] <- "yield"
NSmod <- lm(yield ~., data = NSdata )
df2.1 <- as.data.frame(cbind(df$maturity, NSmod$fitted))
names(df2.1) <- c("maturity", "yield")
```

```
s.mat <- (df$maturity - mean(df$maturity))/sd(df$maturity)
df2 <- as.data.frame(cbind(df$yield, s.mat))
names(df2) <- c("yield", "maturity")
fitpoly <- lm(yield ~ poly(maturity, 4, raw=TRUE), df2)
```

```
lambda <- 0.057
z1 <- exp(-lambda*df$maturity)
z2 <- (1-z1)/(lambda*df$maturity)
z3 <- z2-z1
NSdata <- as.data.frame(cbind(df$yield, z2, z3))
names(NSdata)[1] <- "yield"
NSmod <- lm(yield ~., data = NSdata )
df2.2 <- as.data.frame(cbind(df$maturity, NSmod$fitted))
names(df2.2) <- c("maturity", "yield")
```

```
df2.1$model <- "NS: optimal"
df2.2$model <- "NS: lambda=0.057"
df2.3 <- as.data.frame(cbind(maturity=df$maturity, yield=fitpoly$fitted.values))
df2.3$model <- "4th Order Polynomial"
df2.all <- rbind(df2.1, df2.2, df2.3)
ggplot(data=df2.all, aes(x=maturity, y=yield, colour=model)) +
  geom_line(size=1.5) +
  geom_point(data=df, aes(x=maturity, y=yield), size=4.00, col="black") +
  theme(plot.title = element_text( size=22)) +
  ggtitle("Fitted Yield Curves for May 31, 1990") +
  ylab("Yield (%)") +
  xlab("Maturity (months)") +
  theme(legend.position = c(.65, 0.45)) +
  theme(legend.background = element_rect(fill = "white", colour = "black")) +
  theme(legend.title = element_text(colour = 'black', angle = 0, size = 18, hjust = 3, vjust = 7)) +
  theme(legend.text = element_text(angle=0, size=16, hjust=3, vjust=3, color="black", face="italic"))
```



The three models are similarly close to the data for maturities of less than 25 months. Between 25 and 75 months, the polynomial model the most divergent from the data, and between 75 & 125 months, the polynomial model is most conformant to the data. Optimal Nelson-Siegler fits the data better between 50 and 85 months, while non-optimal Nelson-Siegler fits the data better between 25 and 40 months, and 85 and 125 months.

3. Least Squares Estimate

Least squares estimate of Nelson-Siegel curve, with unknown β_1, β_2, β_3, λ

```
require(minpack.lm)
resfun <- function(par, y, x) {
  lam <- par[4]
  z0  <- exp(-lam*x)
  z1  <- (1-z0)/(lam*x)
  z2  <- z1-z0
  return(y - par[1] - par[2]*z1 - par[3]*z2)
}
start <- c(1, 1, 1, lambda.minSSE2)
set.seed(11201994)
NLSmod <- nls.lm(par=start, fn=resfun, y=df$yield, x=df$maturity)
NLSdf <- as.data.frame(summary(NLSmod)$coefficients)
NLSdf[,3] <- NULL
coeffs <- c("beta1", "beta2", "beta3", "lambda")
NLSdf <- as.data.frame(cbind(coeffs, NLSdf))
names(NLSdf)[1] <- "Variable"
names(NLSdf)[4] <- "P-value"
NLSdf[,2] <- signif(NLSdf[,2], 4)
NLSdf[,3] <- signif(NLSdf[,3], 4)
kable(NLSdf, format="pandoc", caption = "Nelson-Siegler estimates")
```

Table 1: Nelson-Siegler estimates

Variable	Estimate	Std. Error	P-value
beta1	8.5320000	3.366e-02	0.0000000
beta2	-0.9735000	9.583e-02	0.0000001
beta3	-0.0000217	1.871e+03	1.0000000
lambda	0.1975000	3.795e+02	0.9995922

4. Svensson's model

Svensson's model for instantaneous forward interest rate:

$$\mathit{f(0,s)} = \beta_1 + \beta_2 \exp(-\lambda_1 s) + \beta_3 \lambda_1 s \exp(-\lambda_1 s) + \beta_4 \lambda_2 s \exp(-\lambda_2 s)$$

The spot rate or yield for Svensson's model is then:

$$\mathit{R(0,s)} = \beta_1 + \beta_2 \left(\frac{1-\exp(-\lambda_1 s)}{\lambda_1 s} \right) + \beta_3 \left(\frac{1-\exp(-\lambda_1 s)}{\lambda_1 s} - \exp(-\lambda_1 s) \right) + \beta_4 \left(\frac{1-\exp(-\lambda_2 s)}{\lambda_2 s} - \exp(-\lambda_2 s) \right)$$

Least squares estimate of Svensson's model, with unknown β_1, β_2, β_3, β_4, λ_1, λ_2.

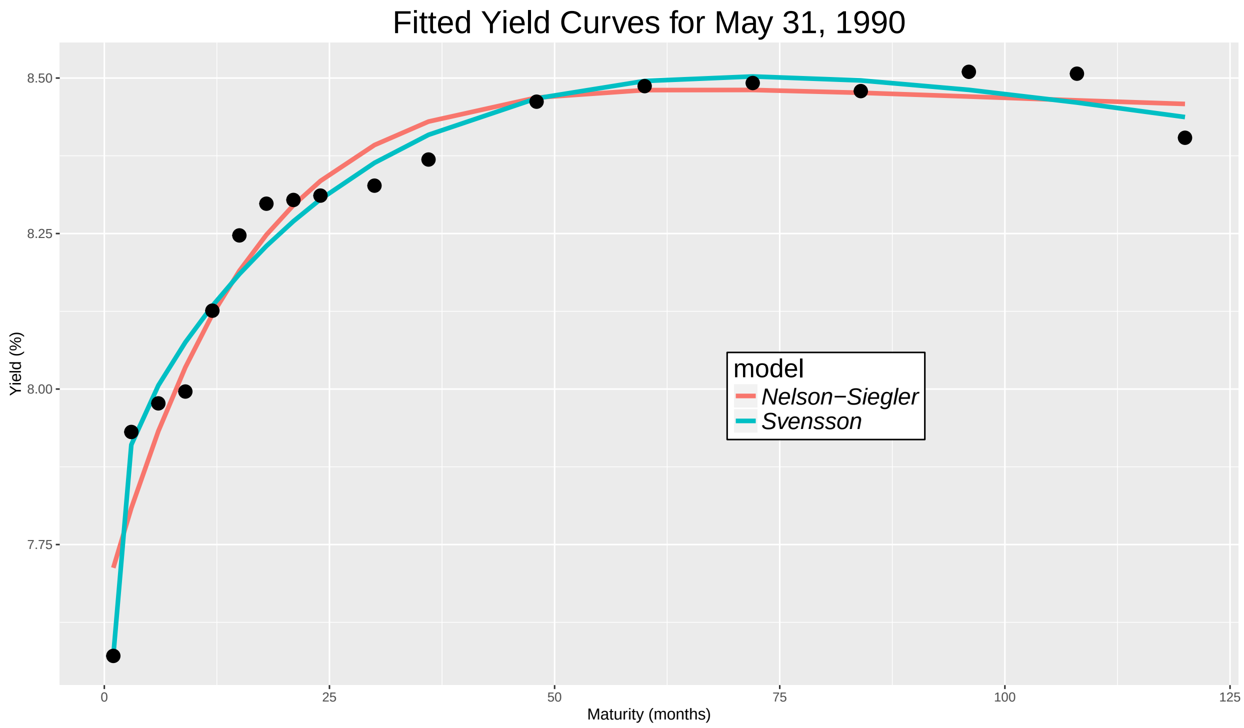
```r
require(minpack.lm)
resfun <- function(par, y, x) {
  lam1 <- par[5]
  lam2 <- par[6]
  z0  <- exp(-lam1*x)
  z1  <- (1-z0)/(lam1*x)
  z2  <- z1-z0
  z3  <- (1-exp(-lam2*x))/(lam2*x)-exp(-lam2*x)
  return(y - par[1] - par[2]*z1 - par[3]*z2 - par[4]* z3)
}
start <- c(8, 1, 1, 2, 0.057, 0.001)
set.seed(11201994)
SVNmod <- nls.lm(par=start, fn=resfun, y=df$yield, x=df$maturity, control=list(maxiter=1000))
SVNdf <- as.data.frame(summary(SVNmod)$coefficients)
SVNdf[,3] <- NULL
coeffs <- c("beta1", "beta2", "beta3", "beta4", "lambda1", "lambda2")
SVNdf <- as.data.frame(cbind(coeffs, SVNdf))
names(SVNdf)[1] <- "Variable"
names(SVNdf)[4] <- "P-value"
SVNdf[,2] <- signif(SVNdf[,2], 4)
SVNdf[,3] <- signif(SVNdf[,3], 4)
kable(SVNdf, format="pandoc", caption = "Svensson estimates")
```

Table 2: Svensson estimates

Variable	Estimate	Std. Error	P-value
beta1	7.8950	1.419e-01	0.0000000
beta2	-56.0500	2.420e+06	0.9999819
beta3	55.0500	2.413e+06	0.9999822
beta4	2.0440	4.811e-01	0.0011284
lambda1	5.7480	4.397e+04	0.9998978
lambda2	0.0252	2.061e-03	0.0000000

```r
SVNreg <- function(y, x, coef){
  lam1 <- coef[5]
  lam2 <- coef[6]
  zz1  <- (1-exp(-lam1*x))/lam1/x
  zz2  <- zz1 - exp(-lam1*x)
  zz3  <- (1-exp(-lam2*x))/lam2/x-exp(-lam2*x)
return (coef[1]+coef[2]*zz1 + coef[3]*zz2 + coef[4]*zz3)
}
SVNyield <- SVNreg(y=df$yield, x=df$maturity, coef=SVNdf[,2])
df4.1 <- as.data.frame(cbind(df$maturity, NSmod$fitted))
names(df4.1) <- c("maturity", "yield")
df4.1$model <- "Nelson-Siegler"
df4.2 <- as.data.frame(cbind(maturity=df$maturity, yield=SVNyield))
df4.2$model <- "Svensson"
df4.all <- rbind(df4.1, df4.2)
```

```
ggplot(data=df4.all, aes(x=maturity, y=yield, colour=model)) +
  geom_line(size=1.5) +
  geom_point(data=df, aes(x=maturity, y=yield), size=4.00, col="black") +
  theme(plot.title = element_text( size=22)) +
  ggtitle("Fitted Yield Curves for May 31, 1990") +
  ylab("Yield (%)") +
  xlab("Maturity (months)") +
  theme(legend.position = c(.65, 0.45)) +
  theme(legend.background = element_rect(fill = "white", colour = "black")) +
  theme(legend.title = element_text(colour = 'black', angle = 0, size = 18, hjust = 3, vjust = 7)) +
  theme(legend.text = element_text(angle=0, size=16, hjust=3, vjust=3, color="black", face="italic"))
```



The NS model least squares estimates of β_1 & β_2 are significantly different from zero, while β_3 & λ are not. In Svensson's model, β_1, β_4 and λ_2 are significantly different from zero, while β_2, β_3 and λ_1 are not. Visually, Svensson seems to provide a closer fit to the data, likely due to the β_4 and λ_2 terms that Svensson added to the NS model.

5. Vasicek's model

Vasicek is a mean-reverting interest rate model with constant volatility.

Least square estimate of the Vasicek's model.

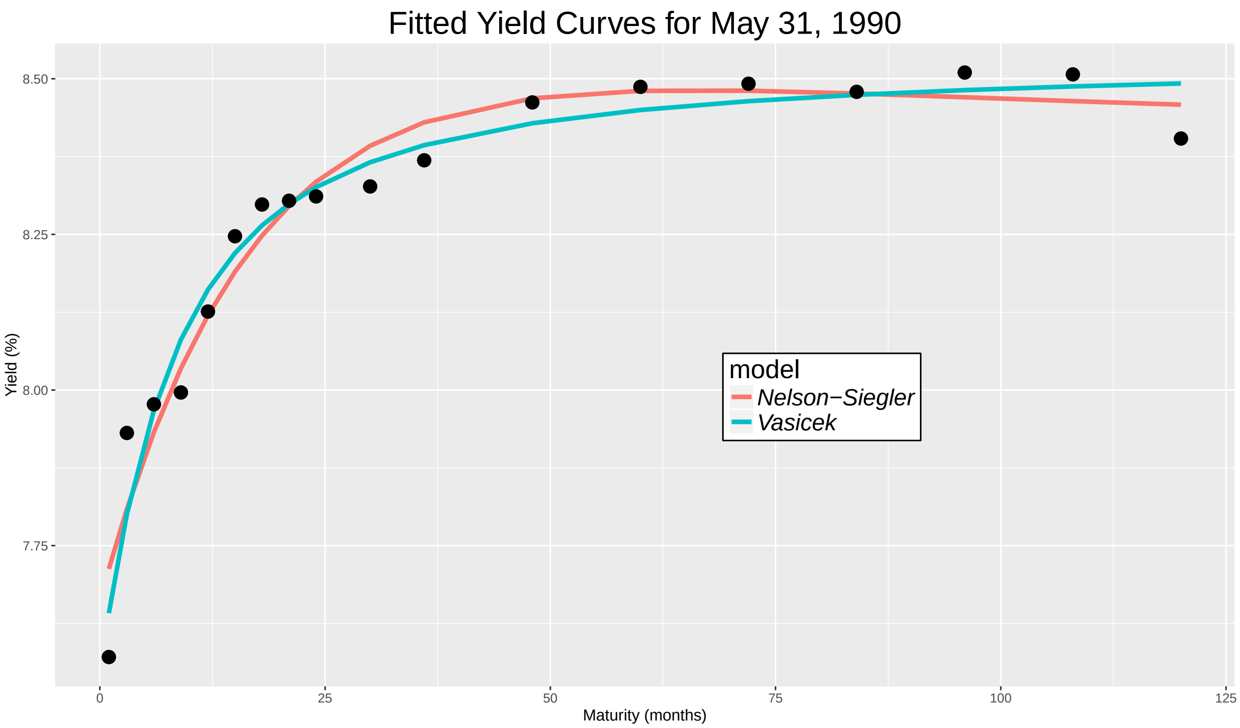
```
VASres <- function(par, y, x) {
  kk <- par[1]
  ss <- par[2]
  tt <- par[3]
  rr <- par[4]
  bb <- (1-exp(-kk*x))/kk
  aa <- (tt-ss/2/kk/kk)*(bb-x)-ss/4/kk*bb*bb
  return( y-(-aa/x+rr*bb/x))
}
par5.init <- c(0.4, 1, 0.05, 7)
VASmod <- nls.lm(par=par5.init, fn=VASres, y=df$yield, x=df$maturity)
VASdf <- as.data.frame(summary(VASmod)$coefficients)
VASdf[,3] <- NULL
coeffs <- c("kappa", "sig2", "theta", "r0")
VASdf <- as.data.frame(cbind(coeffs, VASdf))
names(VASdf)[1] <- "Variable"
names(VASdf)[4] <- "P-value"
VASdf[,2] <- signif(VASdf[,2], 4)
VASdf[,3] <- signif(VASdf[,3], 4)
kable(VASdf, format="pandoc", caption = "Vasicek estimates")
```

Table 3: Vasicek estimates

Variable	Estimate	Std. Error	P-value
kappa	0.15360	0.50250	0.7643111
sig2	0.01962	0.09134	0.8329914
theta	8.95100	4.67500	0.0762036
r0	7.54100	0.07198	0.0000000

```
out <- matrix(VASmod$par, nr = 4, nc=1)
colnames(out) <- c("Vasicek estimates")
rownames(out) <- c("kappa", "sig2", "theta", "r0")
VASreg <- function(x, coef){
  kk <- coef[1]
  ss <- coef[2]
  tt <- coef[3]
  rr <- coef[4]
  bb <- (1-exp(-kk*x))/kk
  aa <- (tt-ss/2/kk/kk)*(bb-x)-ss/4/kk*bb*bb
  return (-aa/x+rr*bb/x)
}
VASyield <- VASreg(df$maturity, out)
```

```r
df5.1 <- as.data.frame(cbind(df$maturity, NSmod$fitted))
names(df5.1) <- c("maturity", "yield")
df5.1$model <- "Nelson-Siegler"
df5.3 <- as.data.frame(cbind(df$maturity, VASyield))
names(df5.3) <- c("maturity", "yield")
df5.3$model <- "Vasicek"
df5.all <- rbind(df5.1, df5.3)
ggplot(data=df5.all, aes(x=maturity, y=yield, colour=model)) +
  geom_line(size=1.5) +
  geom_point(data=df, aes(x=maturity, y=yield), size=4.00, col="black") +
  theme(plot.title = element_text( size=22)) +
  ggtitle("Fitted Yield Curves for May 31, 1990") +
  ylab("Yield (%)") +
  xlab("Maturity (months)") +
  theme(legend.position = c(.65, 0.45)) +
  theme(legend.background = element_rect(fill = "white", colour = "black")) +
  theme(legend.title = element_text(colour = 'black', angle = 0, size = 18, hjust = 3, vjust = 7)) +
  theme(legend.text = element_text(angle=0, size=16, hjust=3, vjust=3, color="black", face="italic"))
```



The estimates for kappa, sig2 and theta in Vasicek's model are not significantly different from zero. Visually, Nelson-Siegler appears to be a better fit to the data.

6. Cox-Ingersoll-Ross model

Cox-Ingersoll-Ross (CIR) is a mean-reverting interest rate model with varying volatility.

Least squares estimate of the Cox-Ingersoll-Ross model , plotted with NS model (optimal λ)

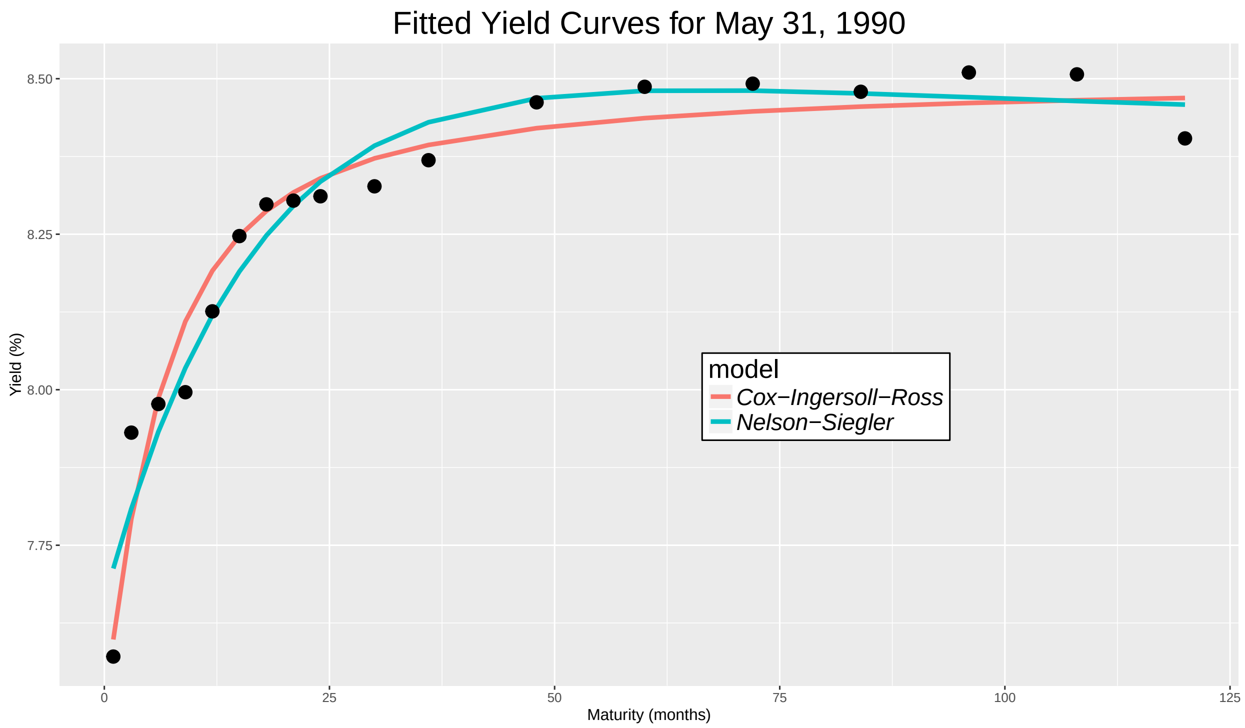
```
CIRres = function(par, y, x1){
  kk <- par[1]
  ss <- par[2]
  tt <- par[3]
  rr <- par[4]
  hh <- sqrt(kk^2 + 2*ss^2)
  GG <- 2*hh + (kk+hh)*(exp(x1*hh)-1)
  bb <- 2*(exp(x1*hh)-1)/GG
  aa <- 2*kk*tt/(ss^2)*(log(2*hh) + (kk+hh)*x1/2 -log(GG))
  return(y-(-aa/x1 + rr*bb/x1))
}
par6.init <- c(0.4, 1, 0.05, 7)
CIRmod <- nls.lm(par=par6.init, fn=CIRres, y=df$yield, x=df$maturity, control=list(maxiter = 1000))
CIRdf <- as.data.frame(summary(VASmod)$coefficients)
CIRdf[,3] <- NULL
coeffs <- c("kappa", "sig2", "theta", "r0")
CIRdf <- as.data.frame(cbind(coeffs, CIRdf))
names(CIRdf)[1] <- "Variable"
names(CIRdf)[4] <- "P-value"
CIRdf[,2] <- signif(CIRdf[,2], 4)
CIRdf[,3] <- signif(CIRdf[,3], 4)
kable(CIRdf, format="pandoc", caption = "Cox-Ingersoll-Ross estimates")
```

Table 4: Cox-Ingersoll-Ross estimates

Variable	Estimate	Std. Error	P-value
kappa	0.15360	0.50250	0.7643111
sig2	0.01962	0.09134	0.8329914
theta	8.95100	4.67500	0.0762036
r0	7.54100	0.07198	0.0000000

```
out <- matrix(CIRmod$par, nr = 4, nc = 1)
colnames(out) <- c("CIR estimates")
rownames(out) <- c("kappa", "sig2", "theta", "r0")
CIRreg <- function(x, coef){
  kk <- coef[1]
  ss <- coef[2]
  tt <- coef[3]
  rr <- coef[4]
  hh <- sqrt(kk^2 + 2*ss^2)
  GG <- 2*hh + (kk+hh)*(exp(x*hh)-1)
  bb <- 2*(exp(x*hh)-1)/GG
  aa <- 2*kk*tt/ss^2*(log(2*hh) + (kk+hh)*x/2 -log(GG))
  return(-(aa/x) + rr*bb/x)  }
CIRyield <- CIRreg(df$maturity, out)
```

```
df6.1 <- as.data.frame(cbind(df$maturity, NSmod$fitted))
names(df6.1) <- c("maturity", "yield")
df6.1$model <- "Nelson-Siegler"
df6.4 <- as.data.frame(cbind(df$maturity, CIRyield))
names(df6.4) <- c("maturity", "yield")
df6.4$model <- "Cox-Ingersoll-Ross"
df6.all <- rbind(df6.1, df6.4)
ggplot(data=df6.all, aes(x=maturity, y=yield, colour=model)) +
  geom_line(size=1.5) +
  geom_point(data=df, aes(x=maturity, y=yield), size=4.00, col="black") +
  theme(plot.title = element_text( size=22)) +
  ggtitle("Fitted Yield Curves for May 31, 1990") +
  ylab("Yield (%)") +
  xlab("Maturity (months)") +
  theme(legend.position = c(.65, 0.45)) +
  theme(legend.background = element_rect(fill = "white", colour = "black")) +
  theme(legend.title = element_text(colour = 'black', angle = 0, size = 18, hjust = 3, vjust = 7)) +
  theme(legend.text = element_text(angle=0, size=16, hjust=3, vjust=3, color="black", face="italic"))
```



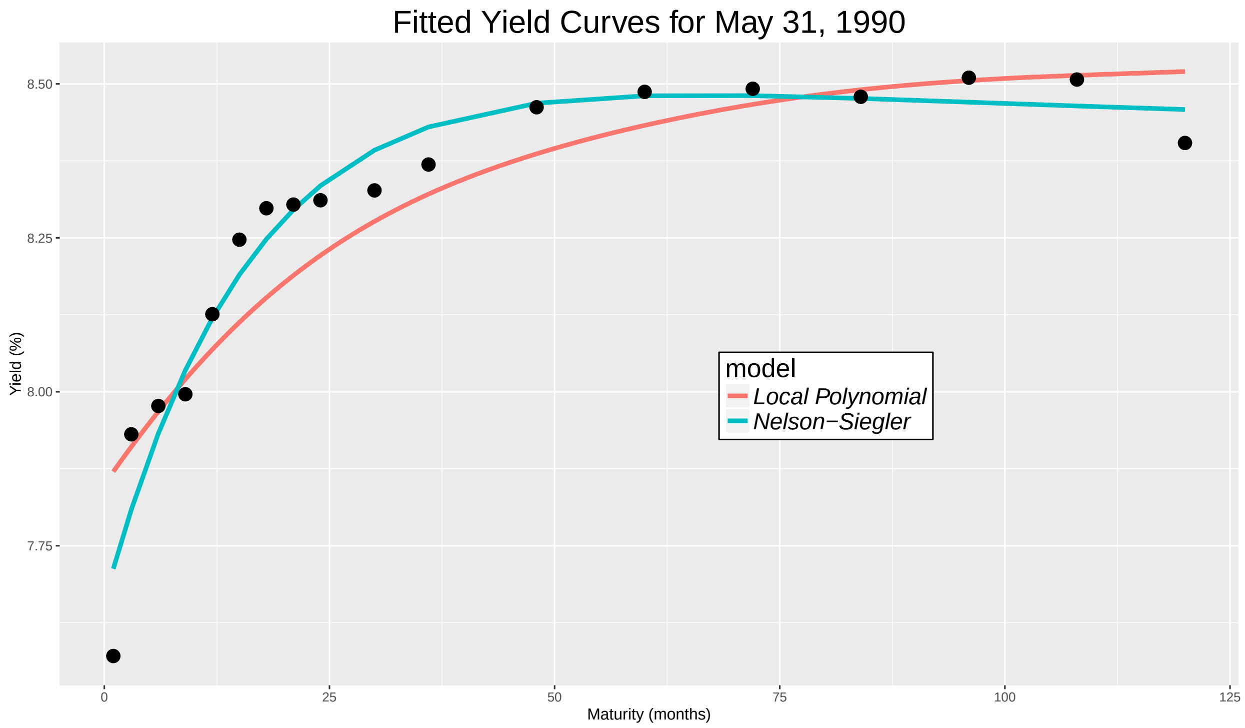
The estimates for kappa, sig2 and theta of the CIR model are not significantly different from zero. Visually, Nelson-Siegler appears to be a marginally better fit to the data.

7. Local Polynomial model

Yield curve using local polynomial model with bandwidth from dpill() function.

```r
require(KernSmooth)
bw <- dpill(df$maturity, df$yield)
LPmod <- locpoly(df$maturity, df$yield, bandwidth=bw)
```

```r
df7.1 <- as.data.frame(cbind(df$maturity, NSmod$fitted))
names(df7.1) <- c("maturity", "yield")
df7.1$model <- "Nelson-Siegler"
df7.5 <- as.data.frame(cbind(LPmod$x, LPmod$y))
names(df7.5) <- c("maturity", "yield")
df7.5$model <- "Local Polynomial"
df7.all <- rbind(df7.1,  df7.5)
ggplot(data=df7.all, aes(x=maturity, y=yield, colour=model)) +
  geom_line(size=1.5) +
  geom_point(data=df, aes(x=maturity, y=yield), size=4.00, col="black") +
  theme(plot.title = element_text( size=22)) +
  ggtitle("Fitted Yield Curves for May 31, 1990") +
  ylab("Yield (%)") +
  xlab("Maturity (months)") +
  theme(legend.position = c(.65, 0.45)) +
  theme(legend.background = element_rect(fill = "white", colour = "black")) +
  theme(legend.title = element_text(colour = 'black', angle = 0, size = 18, hjust = 3, vjust = 7)) +
  theme(legend.text = element_text(angle=0, size=16, hjust=3, vjust=3, color="black", face="italic"))
```



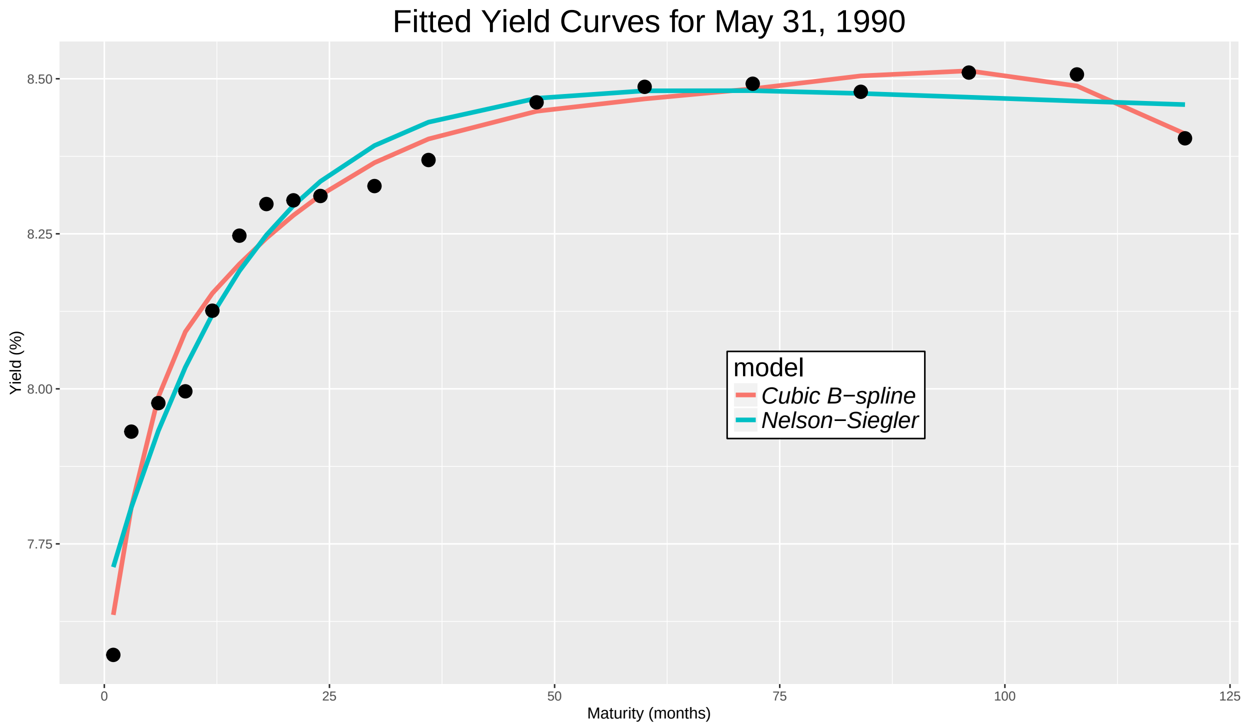
There is significant divergence between the Local polynomial and Nelson-Siegler models, mostly between maturities of 15 and 70 months, and 85 and 125 months. Visually, Nelson-Siegler appears to be a marginally better fit to the data.

8. Cubic B-spline model

Yield curve using cubic B-spline with two internal knots.

```
require(splines)
q <- quantile(df$maturity); knts <- as.numeric(c(q[2], q[4]))
xx <- bs(df$maturity, knots=knts, degree=3); BSyield <- lm(df$yield ~ xx)$fit
```

```
df8.1 <- as.data.frame(cbind(df$maturity, NSmod$fitted))
names(df8.1) <- c("maturity", "yield")
df8.1$model <- "Nelson-Siegler"
df8.6 <- as.data.frame(cbind(df$maturity, BSyield))
names(df8.6) <- c("maturity", "yield")
df8.6$model <- "Cubic B-spline"
df8.all <- rbind(df8.1, df8.6)
ggplot(data=df8.all, aes(x=maturity, y=yield, colour=model)) +
  geom_line(size=1.5) +
  geom_point(data=df, aes(x=maturity, y=yield), size=4.00, col="black") +
  theme(plot.title = element_text( size=22)) +
  ggtitle("Fitted Yield Curves for May 31, 1990") +
  ylab("Yield (%)") +
  xlab("Maturity (months)") +
  theme(legend.position = c(.65, 0.45)) +
  theme(legend.background = element_rect(fill = "white", colour = "black")) +
  theme(legend.title = element_text(colour = 'black', angle = 0, size = 18, hjust = 3, vjust = 7)) +
  theme(legend.text = element_text(angle=0, size=16, hjust=3, vjust=3, color="black", face="italic"))
```



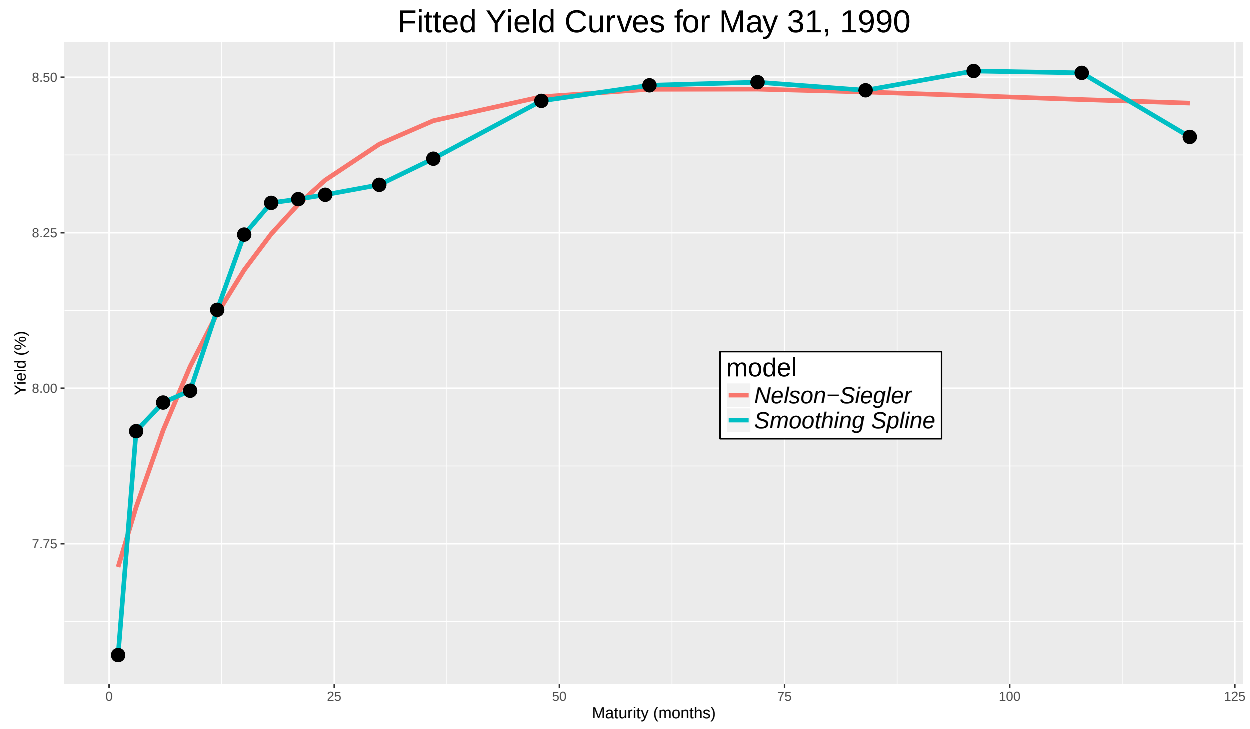
The Cubic B-spline model appears to be a better fit to the data than Nelson-Siegler, possibly introducing some variation error.

9. Smoothing Spline model

Yield curve using smoothing spline, with cross-validated smoothing parameter.

```
SSmod <- smooth.spline(df$maturity, df$yield, cv = T)
```

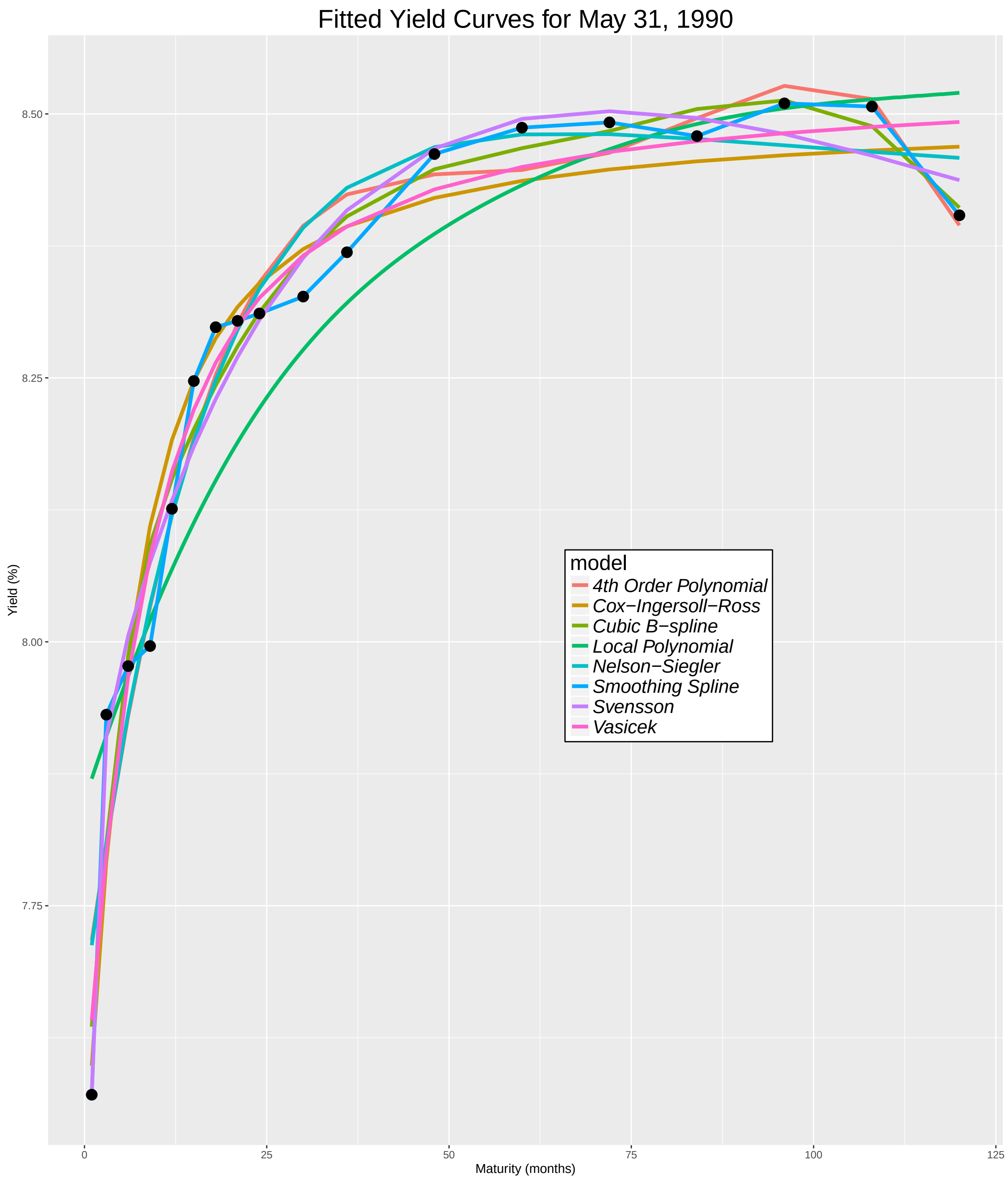
```
df9.1 <- as.data.frame(cbind(df$maturity, NSmod$fitted))
names(df9.1) <- c("maturity", "yield")
df9.1$model <- "Nelson-Siegler"
df9.7 <- as.data.frame(cbind(SSmod$x, SSmod$y))
names(df9.7) <- c("maturity", "yield")
df9.7$model <- "Smoothing Spline"
df9.all <- rbind(df9.1, df9.7)
ggplot(data=df9.all, aes(x=maturity, y=yield, colour=model)) +
  geom_line(size=1.5) +
  geom_point(data=df, aes(x=maturity, y=yield), size=4.00, col="black") +
  theme(plot.title = element_text( size=22)) +
  ggtitle("Fitted Yield Curves for May 31, 1990") +
  ylab("Yield (%)") +
  xlab("Maturity (months)") +
  theme(legend.position = c(.65, 0.45)) +
  theme(legend.background = element_rect(fill = "white", colour = "black")) +
  theme(legend.title = element_text(colour = 'black', angle = 0, size = 18, hjust = 3, vjust = 7)) +
  theme(legend.text = element_text(angle=0, size=16, hjust=3, vjust=3, color="black", face="italic"))
```



The Smoothing Spline model seems to track the data too closely, introducing variation error. The Nelson-Siegler model has a better bias-variance tradeoff.

10. Model Comparison

```r
df10.1 <- as.data.frame(cbind(df$maturity, NSmod$fitted))
names(df10.1) <- c("maturity", "yield")
df10.1$model <- "Nelson-Siegler"
df10.2 <- as.data.frame(cbind(maturity=df$maturity, yield=SVNyield))
df10.2$model <- "Svensson"
df10.3 <- as.data.frame(cbind(df$maturity, VASyield))
names(df10.3) <- c("maturity", "yield")
df10.3$model <- "Vasicek"
df10.4 <- as.data.frame(cbind(df$maturity, CIRyield))
names(df10.4) <- c("maturity", "yield")
df10.4$model <- "Cox-Ingersoll-Ross"
df10.5 <- as.data.frame(cbind(LPmod$x, LPmod$y))
names(df10.5) <- c("maturity", "yield")
df10.5$model <- "Local Polynomial"
df10.6 <- as.data.frame(cbind(df$maturity, BSyield))
names(df10.6) <- c("maturity", "yield")
df10.6$model <- "Cubic B-spline"
df10.7 <- as.data.frame(cbind(SSmod$x, SSmod$y))
names(df10.7) <- c("maturity", "yield")
df10.7$model <- "Smoothing Spline"
df10.8 <- as.data.frame(cbind(maturity=df$maturity, yield=fitpoly$fitted.values))
df10.8$model <- "4th Order Polynomial"
df10.all <- rbind(df10.1, df10.2, df10.3, df10.4, df10.5, df10.6, df10.7, df10.8)
ggplot(data=df10.all, aes(x=maturity, y=yield, colour=model)) +
  geom_line(size=1.5) +
  geom_point(data=df, aes(x=maturity, y=yield), size=4.00, col="black") +
  theme(plot.title = element_text( size=22)) +
  ggtitle("Fitted Yield Curves for May 31, 1990") +
  ylab("Yield (%)") +
  xlab("Maturity (months)") +
  theme(legend.position = c(.65, 0.45)) +
  theme(legend.background = element_rect(fill = "white", colour = "black")) +
  theme(legend.title = element_text(colour = 'black', angle = 0, size = 18, hjust = 3, vjust = 7)) +
  theme(legend.text = element_text(angle=0, size=16, hjust=3, vjust=3, color="black", face="italic"))
```



The Local Polynomial model (green line) has the most bias error, the Smoothing Spline model has the most variance error, and the other models falling in between. The Nelson-Siegler model seems to have the best combination of bias-variance error.

```
sessionInfo()
```

```
## R version 3.2.4 (2016-03-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] splines   stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] servr_0.3          KernSmooth_2.23-15 minpack.lm_1.2-0
## [4] ggplot2_2.1.0      knitr_1.12.3
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.3      digest_0.6.9     plyr_1.8.3       grid_3.2.4
##  [5] gtable_0.2.0     formatR_1.3      magrittr_1.5     evaluate_0.8.3
##  [9] scales_0.4.0     highr_0.5.1      stringi_1.0-1    rmarkdown_0.9.5
## [13] labeling_0.3     tools_3.2.4      stringr_1.0.0    munsell_0.4.3
## [17] httpuv_1.3.3     yaml_2.1.13      colorspace_1.2-6 htmltools_0.3
```