CCAR_Yield_Curve_Comparison

Jingze Sun

12/12/2020

1. R Studio Environment Setup:

```
# Clear work space:
rm(list = ls())

# Import required packages:
library(data.table)
library(readxl)
library(pracma)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x purrr::cross()     masks pracma::cross()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```
library(data.table)
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
```

```
##     yday, year
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(ggplot2)
```

2. Regression Analysis: Mortgage Rate Spreads:

```r
# Import the historic 30-year treasury rates data:
UST_30 <- read_xlsx("WRDS_UST_30.xlsx")
UST_30 <- as.data.table(UST_30)
UST_30 <- UST_30[,c(2,3)]
colnames(UST_30) <- c("Date","UST_30_r")
UST_30[,Date:=as.Date(Date)]

# Correct the dates:
month_end_list <- seq(as.Date("2008-02-01"),length=nrow(UST_30),by="months")-1
UST_30$Date <- month_end_list

# Import the domestic historic data:
his_dom <- fread("Historic_Domestic.csv")
his_dom <- his_dom[,c(2,13)]
colnames(his_dom) <- c("Quart","Mort_r")
his_dom[,Date:=as.Date(as.yearqtr(Quart), frac=1)] # Note: "frac=1" converts
                                                    # date to end-of-month
his_dom <- his_dom[,c(3,2)]

# Combine mortgage rates data and 30-year treasury rates data:
Mort_UST <- UST_30 %>% inner_join(his_dom, by="Date") %>% as.data.table()

# Compute the mortgage rate spread:
Mort_UST[,Mort_Spread:=Mort_r-UST_30_r]
# Get the lag mortgage rate spread column:
Mort_UST[,lag_Mort_Spread:=shift(Mort_Spread)]

# Prepare data for regression analysis:
Mort_UST <- drop_na(Mort_UST[,c(4,5)])

# Run the regression, then get the intercept and correlation coefficient:
reg <- lm(Mort_Spread ~ lag_Mort_Spread, data=Mort_UST)
reg_int <- reg$coefficients[[1]]
reg_coef <- reg$coefficients[[2]]
```

3. Scenario Data:

1) Set up base domestic scenario data table:

```r
# Import the baseline domestic scenario data:
scen_base <- fread("Table_2A_Supervisory_Baseline_Domestic.csv")

# Set the maximum number of predicted quarters:
```

```r
max_Q_len <- 4

# Tidy the scenario data:
scen_base <- scen_base[1:max_Q_len,c(2,9,10,11,13)]
colnames(scen_base) <- c("Quart","UST_3m","UST_5yr","UST_10yr","Mort_r")
scen_base[,Date:=as.Date(as.yearqtr(Quart), frac=1)]
scen_base <- scen_base[,c(6,2:5)]
```

2) Calculate predicated base domestic scenario data:

```r
# Create a copy of the cleaned scenario data:
scen_base_up <- scen_base

# Compute the predicted mortgage rate spreads:
scen_base_up[,Mort_Spread:=NA]
ini_base_spread <- Mort_UST[nrow(Mort_UST),lag_Mort_Spread]
scen_base_up$Mort_Spread[1] <- reg_int+reg_coef*ini_base_spread

for (i in 2:nrow(scen_base_up)) {
  scen_base_up$Mort_Spread[i] <-
    scen_base_up$Mort_Spread[i-1]*reg_coef + reg_int
}

# Compute predicated 30-year Treasury rates:
scen_base_up[,Pred_UST_30yr:=Mort_r-Mort_Spread]
# Condense the updated scenario data table:
scen_base_up <- scen_base_up[,c(1:4,7)]
# Change the data table to a dataframe so that we can run our function later:
scen_base_up <- as.data.frame(scen_base_up)
```

3) Set up severely adverse domestic scenario data table:

```r
# Import the baseline domestic scenario data:
scen_sev <- fread("Table_3A_Supervisory_Severely_Adverse_Domestic.csv")

# Set the maximum number of predicted quarters:
max_Q_len <- 4

# Tidy the scenario data:
scen_sev <- scen_sev[1:max_Q_len,c(2,9,10,11,13)]
colnames(scen_sev) <- c("Quart","UST_3m","UST_5yr","UST_10yr","Mort_r")
scen_sev[,Date:=as.Date(as.yearqtr(Quart), frac=1)]
scen_sev <- scen_sev[,c(6,2:5)]
```

4) Calculate predicated severely adverse domestic scenario data:

```r
# Create a copy of the cleaned scenario data:
scen_sev_up <- scen_sev

# Compute the predicted mortgage rate spreads:
scen_sev_up[,Mort_Spread:=NA]
ini_base_spread <- Mort_UST[nrow(Mort_UST),lag_Mort_Spread]
```

```r
scen_sev_up$Mort_Spread[1] <- reg_int+reg_coef*ini_base_spread

for (i in 2:nrow(scen_sev_up)) {
  scen_sev_up$Mort_Spread[i] <- scen_sev_up$Mort_Spread[i-1]*reg_coef + reg_int
}

# Compute predicated 30-year Treasury rates:
scen_sev_up[,Pred_UST_30yr:=Mort_r-Mort_Spread]
# Condense the updated scenario data table:
scen_sev_up <- scen_sev_up[,c(1:4,7)]
# Change the data table to a dataframe so that we can run our function later:
scen_sev_up <- as.data.frame(scen_sev_up)
```

4. Yield Curve Calculation:

1) Matching Yield Table Function:

```r
# Write a function that, with a given pricing date and a scenario data set,
# returns a table of matched UST rates:
mat_yield_tab <- function(pricing_date, scenario) {
  yield_table <- data.frame(matrix(ncol=2, nrow=4))
  colnames(yield_table) <- c("t","Mat_Yield_r")
  t_list_eg <- c(0.25,5,10,30)
  yield_table$t <- t_list_eg

  for (i in 1:nrow(yield_table)) {
    yield_table$Mat_Yield_r[i] <- scenario[scenario$Date==pricing_date,i+1]
  }
  return(yield_table)
}
```

2) Cubic-Spline Yield Curve Function:

```r
# Write a function that, with a given list of points in time, and a list of
# matched UST rates, returns a table of cubic-spline fitted yield rates for
# all 60 points in time on the yield curve:
CS_yield_SR <- function(t_list, matched_YR, Min_Yr=0.5, Max_Yr=30, Comp=0.5) {
  xi_list <- seq(Min_Yr,Max_Yr,by=Comp)
  years_pt <- t_list
  rates_pt <- matched_YR

  # Get the cubic-spline fitted yield curve using the imported function:
  CS_output <- cubicspline(x=years_pt, y=rates_pt, xi=xi_list)

  # Initialize the table of cubic-spline fitted yields:
  CS_spot_table <- data.frame(matrix(ncol=2, nrow=length(xi_list)))
  colnames(CS_spot_table) <- c("t","YR")
  CS_spot_table$t <- xi_list
  CS_spot_table$YR <- CS_output
  return(CS_spot_table)
}
```

5. Yield Curve Comparison - End of 2020 Q1:

```
# Specify the quarter dates:
Q1_2020_Date <- as.Date("2020-03-31")
```

1) Yield curve data under the based domestic scenario:

```
MY_Tab_base <- mat_yield_tab(Q1_2020_Date,scen_base_up)
CS_YR_Tab_base <- CS_yield_SR(MY_Tab_base$t,MY_Tab_base$Mat_Yield_r)
```

2) Yield curve data under the severely adverse domestic scenario:

```
MY_Tab_sev <- mat_yield_tab(Q1_2020_Date,scen_sev_up)
CS_YR_Tab_sev <- CS_yield_SR(MY_Tab_sev$t,MY_Tab_sev$Mat_Yield_r)
```

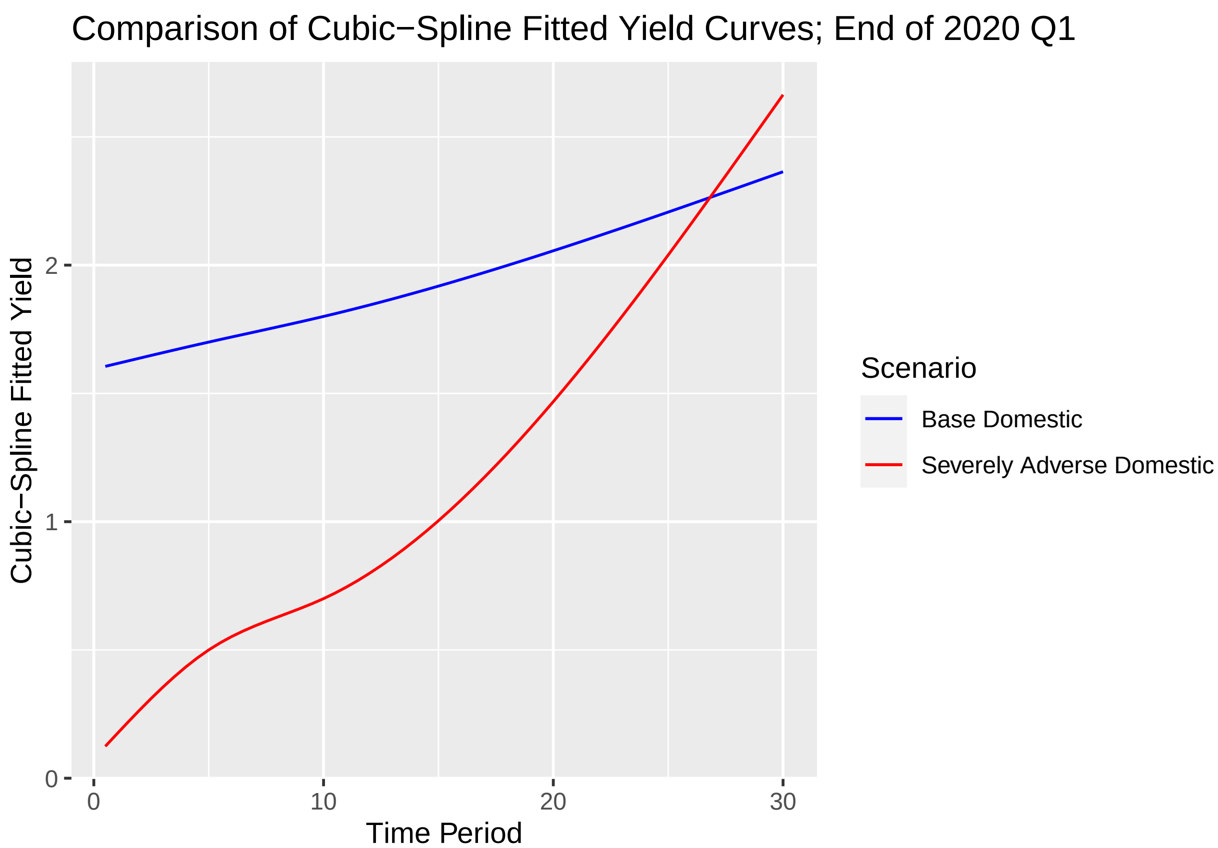
3) Aggregate the yield curve data:

```
colnames(CS_YR_Tab_base) <- c("t", "Yield_Base")
colnames(CS_YR_Tab_sev) <- c("t", "Yield_Severe")
Yield_Data <- data.frame(matrix(ncol = 3, nrow = nrow(CS_YR_Tab_base)))
colnames(Yield_Data) <- c("t", "Yield_Base", "Yield_Severe")
Yield_Data$t <- CS_YR_Tab_base$t
Yield_Data$Yield_Base <- CS_YR_Tab_base$Yield_Base
Yield_Data$Yield_Severe <- CS_YR_Tab_sev$Yield_Severe
```

4) Plot the yield curves under the two different scenarios:

```
colors <- c("Base Domestic"="blue", "Severely Adverse Domestic"="red")

ggplot(data=Yield_Data, aes(x=t)) +
  geom_line(mapping = aes(y=Yield_Base, color="Base Domestic")) +
  geom_line(mapping = aes(y=Yield_Severe, color="Severely Adverse Domestic")) +
  ggtitle("Comparison of Cubic-Spline Fitted Yield Curves; End of 2020 Q1") +
  labs(x = "Time Period",
       y = "Cubic-Spline Fitted Yield",
       color = "Scenario") +
  scale_color_manual(values = colors)
```



6. Yield Curve Comparison - End of 2020 Q2:

```
# Specify the quarter dates:
Q2_2020_Date <- as.Date("2020-06-30")
```

1) Yield curve data under the based domestic scenario:

```
MY_Tab_base <- mat_yield_tab(Q2_2020_Date,scen_base_up)
CS_YR_Tab_base <- CS_yield_SR(MY_Tab_base$t,MY_Tab_base$Mat_Yield_r)
```

2) Yield curve data under the severely adverse domestic scenario:

```
MY_Tab_sev <- mat_yield_tab(Q2_2020_Date,scen_sev_up)
CS_YR_Tab_sev <- CS_yield_SR(MY_Tab_sev$t,MY_Tab_sev$Mat_Yield_r)
```

3) Aggregate the yield curve data:

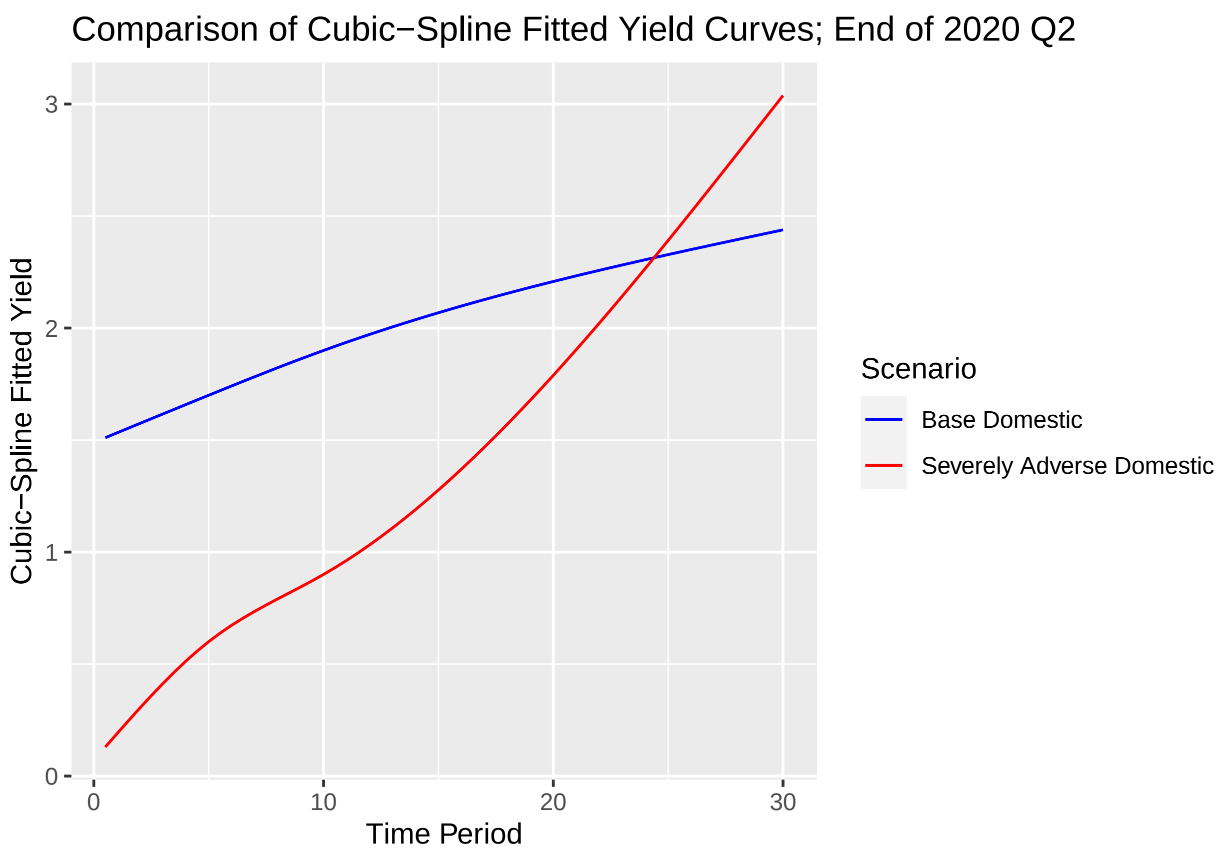
```
colnames(CS_YR_Tab_base) <- c("t", "Yield_Base")
colnames(CS_YR_Tab_sev) <- c("t", "Yield_Severe")
Yield_Data <- data.frame(matrix(ncol = 3, nrow = nrow(CS_YR_Tab_base)))
colnames(Yield_Data) <- c("t", "Yield_Base", "Yield_Severe")
Yield_Data$t <- CS_YR_Tab_base$t
```

```
Yield_Data$Yield_Base <- CS_YR_Tab_base$Yield_Base
Yield_Data$Yield_Severe <- CS_YR_Tab_sev$Yield_Severe
```

4) Plot the yield curves under the two different scenarios:

```
colors <- c("Base Domestic"="blue", "Severely Adverse Domestic"="red")

ggplot(data=Yield_Data, aes(x=t)) +
  geom_line(mapping = aes(y=Yield_Base, color="Base Domestic")) +
  geom_line(mapping = aes(y=Yield_Severe, color="Severely Adverse Domestic")) +
  ggtitle("Comparison of Cubic-Spline Fitted Yield Curves; End of 2020 Q2") +
  labs(x = "Time Period",
       y = "Cubic-Spline Fitted Yield",
       color = "Scenario") +
  scale_color_manual(values = colors)
```



7. Yield Curve Comparison - End of 2020 Q3:

```
# Specify the quarter dates:
Q3_2020_Date <- as.Date("2020-09-30")
```

1) Yield curve data under the based domestic scenario:

```
MY_Tab_base <- mat_yield_tab(Q3_2020_Date,scen_base_up)
CS_YR_Tab_base <- CS_yield_SR(MY_Tab_base$t,MY_Tab_base$Mat_Yield_r)
```

2) Yield curve data under the severely adverse domestic scenario:

```
MY_Tab_sev <- mat_yield_tab(Q3_2020_Date,scen_sev_up)
CS_YR_Tab_sev <- CS_yield_SR(MY_Tab_sev$t,MY_Tab_sev$Mat_Yield_r)
```

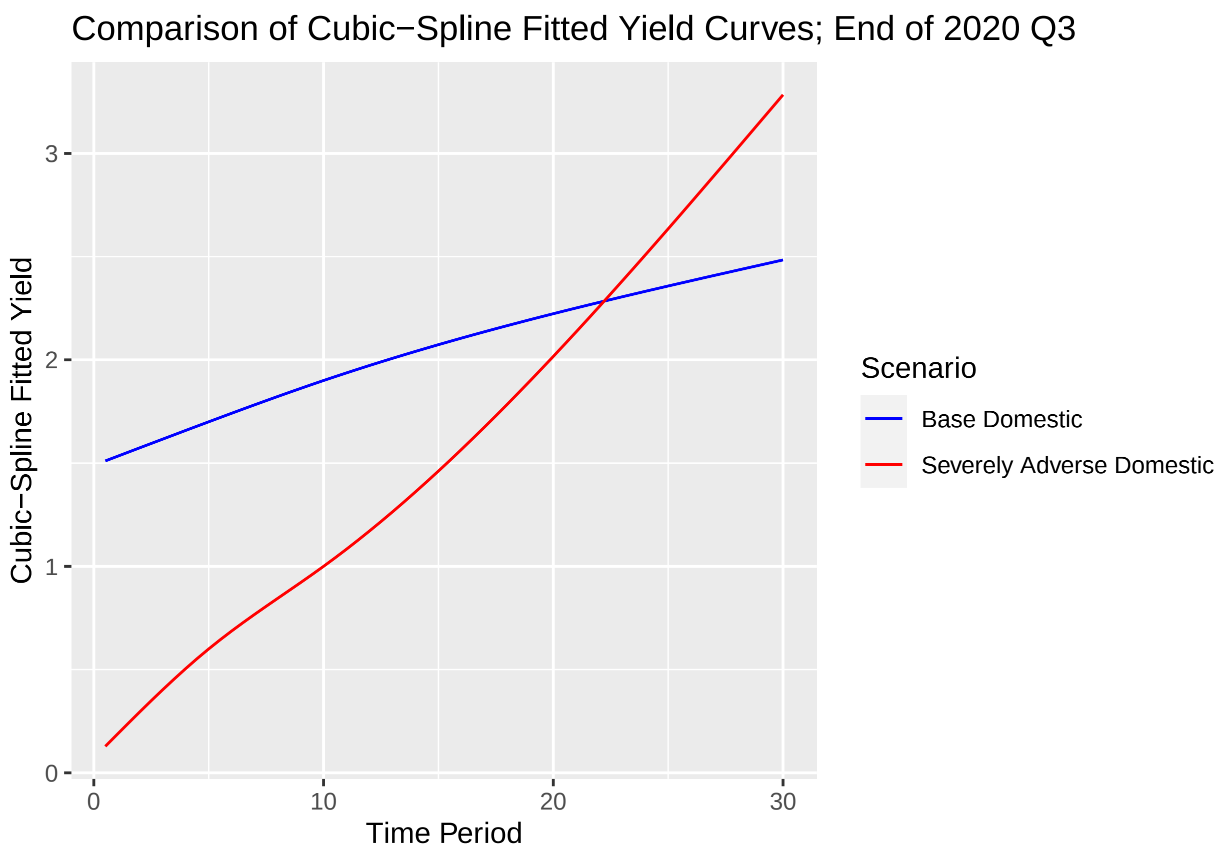
3) Aggregate the yield curve data:

```
colnames(CS_YR_Tab_base) <- c("t", "Yield_Base")
colnames(CS_YR_Tab_sev) <- c("t", "Yield_Severe")
Yield_Data <- data.frame(matrix(ncol = 3, nrow = nrow(CS_YR_Tab_base)))
colnames(Yield_Data) <- c("t", "Yield_Base", "Yield_Severe")
Yield_Data$t <- CS_YR_Tab_base$t
Yield_Data$Yield_Base <- CS_YR_Tab_base$Yield_Base
Yield_Data$Yield_Severe <- CS_YR_Tab_sev$Yield_Severe
```

4) Plot the yield curves under the two different scenarios:

```
colors <- c("Base Domestic"="blue", "Severely Adverse Domestic"="red")

ggplot(data=Yield_Data, aes(x=t)) +
  geom_line(mapping = aes(y=Yield_Base, color="Base Domestic")) +
  geom_line(mapping = aes(y=Yield_Severe, color="Severely Adverse Domestic")) +
  ggtitle("Comparison of Cubic-Spline Fitted Yield Curves; End of 2020 Q3") +
  labs(x = "Time Period",
       y = "Cubic-Spline Fitted Yield",
       color = "Scenario") +
  scale_color_manual(values = colors)
```



8. Yield Curve Comparison - End of 2020 Q4:

```
# Specify the end of 2020 4th quarter date:
Q4_2020_Date <- as.Date("2020-12-31")
```

1) Yield curve data under the based domestic scenario:

```
MY_Tab_base <- mat_yield_tab(Q4_2020_Date,scen_base_up)
CS_YR_Tab_base <- CS_yield_SR(MY_Tab_base$t,MY_Tab_base$Mat_Yield_r)
```

2) Yield curve data under the severely adverse domestic scenario:

```
MY_Tab_sev <- mat_yield_tab(Q4_2020_Date,scen_sev_up)
CS_YR_Tab_sev <- CS_yield_SR(MY_Tab_sev$t,MY_Tab_sev$Mat_Yield_r)
```

3) Aggregate the yield curve data:

```
colnames(CS_YR_Tab_base) <- c("t", "Yield_Base")
colnames(CS_YR_Tab_sev) <- c("t", "Yield_Severe")
Yield_Data <- data.frame(matrix(ncol = 3, nrow = nrow(CS_YR_Tab_base)))
colnames(Yield_Data) <- c("t", "Yield_Base", "Yield_Severe")
Yield_Data$t <- CS_YR_Tab_base$t
```

```
Yield_Data$Yield_Base <- CS_YR_Tab_base$Yield_Base
Yield_Data$Yield_Severe <- CS_YR_Tab_sev$Yield_Severe
```

4) Plot the yield curves under the two different scenarios:

```
colors <- c("Base Domestic"="blue", "Severely Adverse Domestic"="red")

ggplot(data=Yield_Data, aes(x=t)) +
  geom_line(mapping = aes(y=Yield_Base, color="Base Domestic")) +
  geom_line(mapping = aes(y=Yield_Severe, color="Severely Adverse Domestic")) +
  ggtitle("Comparison of Cubic-Spline Fitted Yield Curves; End of 2020 Q4") +
  labs(x = "Time Period",
       y = "Cubic-Spline Fitted Yield",
       color = "Scenario") +
  scale_color_manual(values = colors)
```