Adil-Gokturk_HW6.R

HAG

2020-03-29

```
# Adil Gokturk
# FIN 659

# HW6: SECURITIZATION

## Textbook Reference:      Section 8.1, pp. 184-188


## Problem
## The key principle to take away from this problem is that
# there is an effect on securitized financial instruments
# when borrowers default on their payments on
#the underlying assets making up the securitized product.

# A mortgage originator has recently decided to pool 2,000 of its mortgages into a portfolio.
# The average value of each mortgage in the pool is $250,000, and
# the average interest rate is 4.80%.
# The mortgage originator plans to sell the mortgages to an investment bank,
#but it will continue to service the mortgages for a small fee (0.50% per year).

# The cash flows from the assets will be allocated to tranches.
# The investment bank plans to create a mortgage-backed security with only three tranches:
# a senior tranche,
# a mezzanine tranche, and
# an equity tranche.
# The principal from the portfolio of mortgages will be allocated as follows:
# 70% will go to the senior tranche (rated 'AAA' and earning 2.75%),
# 20% will go to the mezzanine tranche (rated 'A' and earning 3.75%), and
# the remainder will go to the equity tranche (rated 'CCC').
# The investment bank plans to keep 0.80% per year as
# its share for the administration of the securitization process.

# set working directory
setwd("~/Desktop/Spring2020/FIN659/Assignments/hw6")
getwd()
```

```
## [1] "/Users/HAG/Desktop/Spring2020/FIN659/Assignments/hw6"
```

```
# Load the libraries
library(tidyverse)
```

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

```
## v ggplot2 3.3.0     v purrr   0.3.3
```

```
## v tibble  2.1.3     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ---------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

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

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

```
##
## Attaching package: 'xts'
```

```
## The following objects are masked from 'package:dplyr':
##
##     first, last
```

```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
library(optiRum)
library(jrvFinance)
library(knitr)

# This information is summarized in the tables below.

number.of.mortgages <- 2000
average.mortgage.value <- 250000 # US $
average.interest.rate <- 0.048 # %
mortgage.originator.fee <- 0.005 # % per year
investment.bank.fee <- 0.008 # % per year

senior.tranche <- c(0.7, 0.0275)
mezzanine.tranche <- c(0.2, 0.0375)
equity.tranche <- c(0.1, "NA")

(tranches <- rbind(senior.tranche, mezzanine.tranche, equity.tranche))
```

```
##                   [,1]  [,2]
## senior.tranche    "0.7" "0.0275"
## mezzanine.tranche "0.2" "0.0375"
## equity.tranche    "0.1" "NA"
```

```r
c("MBS Principal Allocation", "Rate") -> colnames(tranches)

# Let's take a look at the tranche data
kable(tranches)
```

	MBS Principal Allocation	Rate
senior.tranche	0.7	0.0275
mezzanine.tranche	0.2	0.0375
equity.tranche	0.1	NA

```r
## Question: What is the total value of the mortgage pool being sold to the investment bank?
options(scipen = 20) # convert scientific format penalize if is more than 20
(total.value.of.the.mortgage.pool <- number.of.mortgages * average.mortgage.value)
```

```
## [1] 500000000
```

```r
print("Answer: the total value of the mortgage pool = $500,000,000")
```

```
## [1] "Answer: the total value of the mortgage pool = $500,000,000"
```

```r
## What is the total annual interest expense paid by
## the borrowers (those who took out mortgages to buy their homes)?

(annual.interest.expense.paid.by.borrowers <- average.interest.rate * total.value.of.the.mortgage.pool)
```

```
## [1] 24000000
```

```r
print("Answer: the total annual interest expense paid by the borrowers = $24,000,000")
```

```
## [1] "Answer: the total annual interest expense paid by the borrowers = $24,000,000"
```

```r
## From the payments obtained from the borrowers,
# fees are paid to the mortgage originator and the investment bank,
# and returns are provided to the MBS investors.
# Complete the table below,
# showing what the annual cash flow should be to each of the parties.

# Mortgage originator
(mortgage.originator <- total.value.of.the.mortgage.pool * mortgage.originator.fee)
```

```
## [1] 2500000
```

```r
print("Answer: Mortgage originator  = $2,500,000")
```

```
## [1] "Answer: Mortgage originator  = $2,500,000"
```

```r
# Investment bank
(investment.bank <- total.value.of.the.mortgage.pool * investment.bank.fee)
```

```
## [1] 4000000
```

```r
print("Answer: Investment bank = $4,000,000")
```

```
## [1] "Answer: Investment bank = $4,000,000"
```

```r
# Senior tranche investors
(senior.tranche.investors <- total.value.of.the.mortgage.pool %*% senior.tranche[1]* senior.tranche[2])
```

```
##          [,1]
## [1,] 9625000
```

```
print("Answer: Senior tranche investors = $9,625,000")
```

```
## [1] "Answer: Senior tranche investors = $9,625,000"
```

```
# Mezzanine tranche investors
(mezzanine.tranche.investors <- total.value.of.the.mortgage.pool %*% mezzanine.tranche[1]  * mezzanine.
```

```
##          [,1]
## [1,] 3750000
```

```
print("Answer: Mezzanine tranche investors = $3,750,000")
```

```
## [1] "Answer: Mezzanine tranche investors = $3,750,000"
```

```
# Equity tranche investors
(equity.tranche.investors <- annual.interest.expense.paid.by.borrowers - mortgage.originator - investmer
```

```
##          [,1]
## [1,] 4125000
```

```
print("Answer: Equity tranche investors = $4,125,000")
```

```
## [1] "Answer: Equity tranche investors = $4,125,000"
```

```
# let's put it in a table
table2 <- rbind(mortgage.originator,
                investment.bank,
                senior.tranche.investors,
                mezzanine.tranche.investors,
                equity.tranche.investors)
# Add row names
c("senior.tranche.investors",
  "mezzanine.tranche.investors",
  "equity.tranche.investors") -> row.names(table2)[3:5] # rename row names

# rename column names
c("US$") -> colnames(table2)
# better one
kable(table2, align = "c")
```

	US$
mortgage.originator	2500000
investment.bank	4000000
senior.tranche.investors	9625000
mezzanine.tranche.investors	3750000
equity.tranche.investors	4125000

```
## Question: What is the MBS principal allocation to the equity tranche in dollar terms?

# convert to numeric value
as.numeric(equity.tranche)-> equity.tranche
```

```
## Warning: NAs introduced by coercion
```

```
(MBS.principal.allocation.to.the.equity.tranche <- total.value.of.the.mortgage.pool * equity.tranche[1])
```

```
## [1] 50000000
```

```
print("Answer: MBS principal allocation to the equity tranche in dollar = $50,000,000")
```

```
## [1] "Answer: MBS principal allocation to the equity tranche in dollar = $50,000,000"
```

```
## Question: If the investment bank plans to keep the equity tranche for themselves,
## what annual return will be obtained on this investment?
(annual.return.obtained.on.investment <- equity.tranche.investors/MBS.principal.allocation.to.the.equity)
```

```
##        [,1]
## [1,] 0.0825
```

```
print("Answer: Annual return obtained on this investment = 8.25%")
```

```
## [1] "Answer: Annual return obtained on this investment = 8.25%"
```

```
## PART II:
# The investment bank plans to create a collateralized mortgage obligation (CMO)
# from the mezzanine tranche of the MBS.
# The principal assigned to the senior, mezzanine, and equity tranches of the CMO
# differ from that assigned to the MBS;
# the credit ratings and coupon rates for the CMO are the same as for the MBS.
# The principal from the mezzanine tranche of the MBS will be allocated as follows:
# 65% will go to the senior tranche (rated 'AAA' and earning 2.75%),
# 25% will go to the mezzanine tranche (rated 'A' and earning 3.75%), and
# the remainder will go to the equity tranche (rated 'CCC').
# All of this information is represented in the diagram and table below.


senior.tranche <- c(0.65, 0.0275)
mezzanine.tranche <- c(0.25, 0.0375)
equity.tranche <- c(0.1, "NA")

#equity.tranche <- as.numeric(equity.tranche)
# let's put it in a table
(tranches2 <- rbind(senior.tranche, mezzanine.tranche, equity.tranche))
```

```
##                   [,1]   [,2]
## senior.tranche    "0.65" "0.0275"
## mezzanine.tranche "0.25" "0.0375"
## equity.tranche    "0.1"  "NA"
```

```
tranches2 <- as.data.frame(tranches2)
# Rename column names
c("CMO Principal Allocation", "Rate") -> colnames(tranches2)

# Let's take a look at the tranche data
kable(tranches2, align = "c")
```

	CMO Principal Allocation	Rate
senior.tranche	0.65	0.0275
mezzanine.tranche	0.25	0.0375
equity.tranche	0.1	NA

```r
# Suppose that the economy experiences a real estate crisis,
# prices drop, and many of the original borrowers decide to default on their mortgages
# and abandon their homes.
# Complete the table below to show the estimated losses to the three tranches of the CMO.

# Losses on underlying assets
(losses.on.underlying.assets <- c(0.1, 0.125, 0.15, 0.175, 0.2))
```

```
## [1] 0.100 0.125 0.150 0.175 0.200
```

```r
as.numeric(equity.tranche)
```

```
## Warning: NAs introduced by coercion
```

```
## [1] 0.1  NA
```

```r
# Losses to mezzanine tranche of MBS
# mezzanine tranche of MBS = 0.2
(equity.tranche <- 0.1)
```

```
## [1] 0.1
```

```r
(losses.to.mezzanine.tranche.of.MBS <- ifelse(test = losses.on.underlying.assets <= equity.tranche,
                                              yes = 0,
                                              no = (losses.on.underlying.assets - equity.tranche) / 0.2
```

```
## [1] 0.000 0.125 0.250 0.375 0.500
```

```r
# Losses to equity tranche of CMO
# mezzanine tranche of MBS = 0.10
(losses.to.equity.tranche.of.CMO <- ifelse(test = losses.to.mezzanine.tranche.of.MBS < 0.10,
                                           yes = equity.tranche / losses.to.mezzanine.tranche.of.MBS,
                                           no = 1 ))
```

```
## [1] Inf   1   1   1   1
```

```r
# Losses to mezzanine tranche of CMO
# Equity tranche = 0.10
# Mezzanine tranche = 0.25
(losses.to.mezzanine.tranche.of.CMO <- ifelse(test = losses.to.mezzanine.tranche.of.MBS< equity.tranche
                                              no = ifelse(test = losses.to.mezzanine.tranche.of.MBS < (
                                                          yes = (losses.to.mezzanine.tranche.of.MBS - e
                                                          no = 1 )))
```

```
## [1] 0.0 0.1 0.6 1.0 1.0
```

```r
# Losses to senior tranche of CMO
# Equity tranche = 0.10
# Mezzanine tranche = 0.25
# Senior tranche = 0.65
(losses.to.senior.tranche.of.CMO <- ifelse(test =  losses.to.mezzanine.tranche.of.MBS < mezzanine.tranch
                                           yes = 0,
                                           no = (losses.to.mezzanine.tranche.of.MBS- mezzanine.tranche[
```

```
## [1] 0.00000000 0.00000000 0.00000000 0.03846154 0.23076923
```

```r
# Let's put all in a table
est.losses.of.all.tranches <- data.frame(losses.on.underlying.assets,
                                         losses.to.mezzanine.tranche.of.MBS,
                                         losses.to.equity.tranche.of.CMO,
```

```
                                        losses.to.mezzanine.tranche.of.CMO,
                                        losses.to.senior.tranche.of.CMO)

# rename column names
c("Underlying Assets",
  "MBS Mezz. Tranche",
  "CMO Equity Tranche",
  "CMO Mezz. Tranche",
  "CMO Senior Tranche") -> colnames(est.losses.of.all.tranches)

# Visualize it
kable(est.losses.of.all.tranches, align = "c")
```

Underlying Assets	MBS Mezz. Tranche	CMO Equity Tranche	CMO Mezz. Tranche	CMO Senior Tranche
0.100	0.000	Inf	0.0	0.0000000
0.125	0.125	1	0.1	0.0000000
0.150	0.250	1	0.6	0.0000000
0.175	0.375	1	1.0	0.0384615
0.200	0.500	1	1.0	0.2307692