

# Gestion de Portefeuille

## TP-8: Gestion Obligataire

Patrick Hénaff

Février-Mars 2020

L'objet de ce TP est de construire un modèle de gestion obligataire qui combine les méthodes d'adossement flux à flux et d'immunisation.

La méthode d'adossement flux à flux est adaptée aux échéances court-terme, car elle évite de faire des transactions trop nombreuses. Par contre, elle manque de flexibilité, et se révèle être une solution chère. Pour un horizon plus lointain, on lui préfère donc l'immunisation.

## Données

Les données sont simulées pour les besoins du TP.

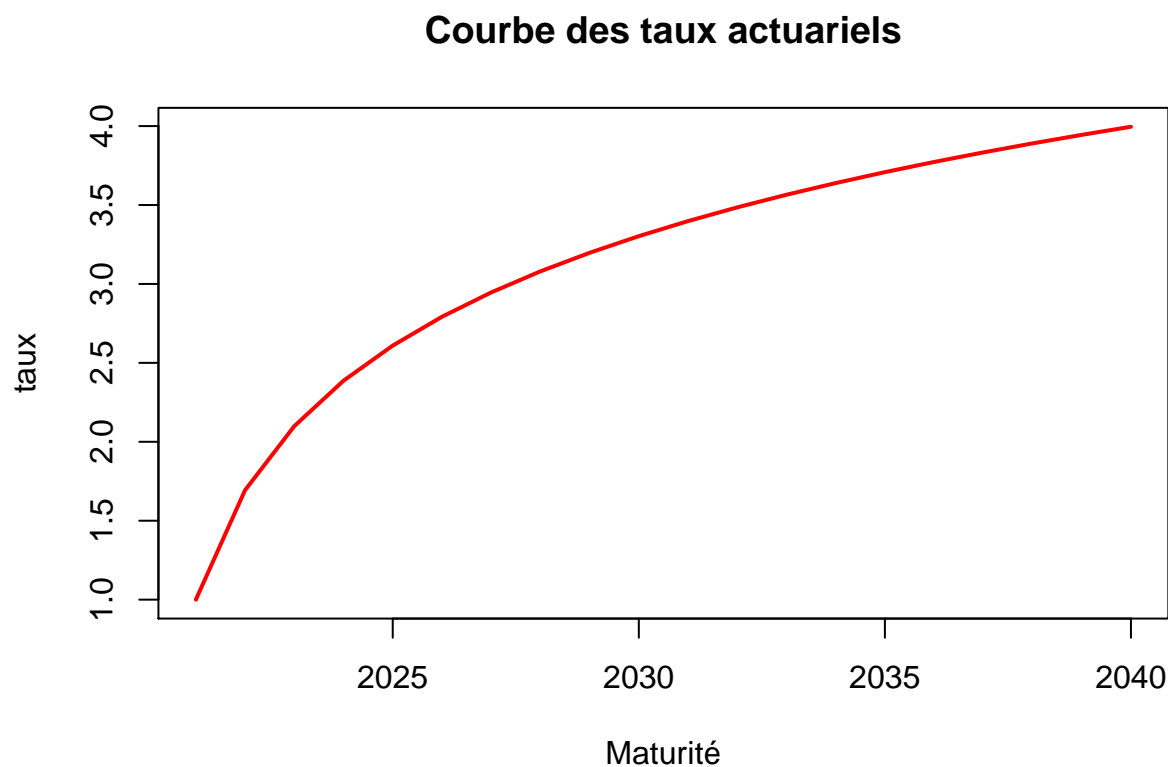
## Obligations

```
df.o <- read.csv("obligations.csv",
                 colClasses=c("character", "character", NA))
df.o$dtE <- as.Date(df.o$dtE, format("%m/%d/%Y"))
df.o$dtM <- as.Date(df.o$dtM, format("%m/%d/%Y"))
df.o$Nom <- sapply(seq_along(df.o$dtE), function(x) paste('Bond-',x, sep=''))
nb.bonds <- nrow(df.o)
kable(df.o[1:10,],
      col.names=c("Dt Emission", "Dt Maturité", "Coupon (%)", "Nom"),
      format="latex", booktabs=T)%>%
  kable_styling(position="center")
```

Dt Emission	Dt Maturité	Coupon (%)	Nom
2018-06-01	2021-06-01	1.7	Bond-1
2016-12-01	2021-12-01	3.5	Bond-2
2019-04-01	2022-04-01	1.7	Bond-3
2017-10-01	2022-10-01	2.6	Bond-4
2020-02-01	2023-02-01	2.7	Bond-5
2016-04-01	2023-04-01	1.2	Bond-6
2018-08-01	2023-08-01	5.0	Bond-7
2020-12-01	2023-12-01	2.2	Bond-8
2017-02-01	2024-02-01	3.5	Bond-9
2019-06-01	2024-06-01	2.7	Bond-10

## Courbe des taux actuariels

```
dt.mat <- seq(ymd('2021-01-01'), ymd('2040-01-01'), by="year")
tx <- 1 + log(seq_along(dt.mat))
df.cdt <- data.frame(mat=dt.mat, tx=tx)
plot(dt.mat, tx, type='l', lwd=2, col='red',
     main='Courbe des taux actuariels',
     xlab="Maturité", ylab="taux")
```



## Calculs préliminaires

- Ecrire une fonction qui permet d'interpoler la courbe de taux pour une date de maturité donnée.
- Choisir une obligation de la liste, interpoler le rendement actuariel et calculer le prix "pied de coupon," le coupon couru, le prix "avec coupon couru," et les indicateurs de risque. Utiliser le paquet "Bond-Valuation" et la convention AFB ACT/ACT pour les décomptes de jours.

```
get.yield <- function(dt) {
  res <- approx(df.cdt$mat, df.cdt$tx, xout=dt)
  res$y
}

bond <- df.o[6,]
dt.calc <- as.Date('2021-03-17')
```

```
res <- BondVal.Price(
  YtM=get.yield(bond$dtM),
  SETT=dt.calc,
  Em=bond$dtE,
  Mat=bond$dtM,
  CpY=1,
  Coup=bond$Coupon,
  DCC=3)
```

Ce qui donne les résultats suivants:

Table 1: Prix Pied de coupon (CP) et coupon couru (CC) de l'obligation Bond-6.

	dtE	dtM	Coupon	Nom	CP	CC
6	2016-04-01	2023-04-01	1.2	Bond-6	98.08398	1.150685

## Partie 1: Immunisation

Soit un passif de 10,000,000€ payable le 2/1/2025. Construisez un portefeuille de deux obligations ayant, au 17/3/2021, la même valeur et la même PV01 que le passif. Optimisez le rendement moyen du portefeuille ainsi construit.

## Calcul de prix et indicateurs de risque des obligations

La PV01 est calculée par différence:

```
bond.calc <- function(df.row, dt.calc) {
  if(dt.calc>=df.row$dtM) {
    c(0, 0, 0)
  } else {
    y <- get.yield(df.row$dtM)
    res <- BondVal.Price(YtM=y, SETT=dt.calc, Em=df.row$dtE,
                        Mat=df.row$dtM,
                        CpY=1, Coup=df.row$Coupon, DCC=3)
    res.plus <- BondVal.Price(YtM=y+0.01, SETT=dt.calc, Em=df.row$dtE,
                             Mat=df.row$dtM,
                             CpY=1, Coup=df.row$Coupon, DCC=3)
    PV01 <- res$DP - res.plus$DP
    c(y, res$DP, PV01)
  }
}

df.o1 <- df.o
df.o1$y = NA
df.o1$DP = NA
df.o1$PV01 = NA

for(i in seq(nrow(df.o1))) {
  res <- bond.calc(df.o1[i,], dt.calc)
```

```
df.o1$y[i] <- res[1]
df.o1$DP[i] <- res[2]
df.o1$PV01[i] <- res[3]
}
```

Les prix “coupon couru inclus” (DP) et les PV01 des obligations à la date de calcul sont résumées dans le tableau ci-dessous.

dtE	dtM	Coupon	DP	PV01
2018-06-01	2021-06-01	1.7	101.4282	0.0021062
2016-12-01	2021-12-01	3.5	102.3135	0.0071763
2019-04-01	2022-04-01	1.7	101.5343	0.0102166

## Prix et PV01 du passif

On calcule de même la valeur présente et la PV01 du passif:

```
zc.calc <- function(dt.calc, dt.mat) {
  # Prix et PV01 en dt.calc de 1 euro payé à la date dt.mat
  mat <- as.numeric(dt.mat-dt.calc)/365
  y <- get.yield(dt.mat)
  P <- (1+y/100)^(-mat)
  PV01 <- mat*(1+y/100)^(-mat-1) * 0.0001
  c(P, PV01, y)
}

nominal <- 10000000
dt.mat.L <- as.Date("2025-01-02")
res <- zc.calc(dt.calc, dt.mat.L)
P.L <- nominal * res[1]
PV01.L <- nominal * res[2]
```

Table 2: Valeur présente et PV01 du passif

Valeur	PV01
9067346	3357.951

## Programme d’immunisation: maximiser le rendement du portefeuille.

Le rendement actuariel moyen d’un portefeuille est

$$r^* = \frac{\sum_i q_i r_i V_i}{\sum_i q_i V_i}$$

avec  $q_i$  la quantité de titre  $i$  dans le portefeuille,  $r_i$  son rendement actuariel et  $V_i$  sa PV01. Comme le dénominateur est fixé, le problème se réduit à maximiser  $\sum_i q_i r_i V_i$ .

$$\max \sum_i q_i r_i \mathcal{PV}01_i \quad (1)$$

s.t.

$$\sum_i q_i \mathcal{PV}01_i(T) = \mathcal{PV}01_L(T) \quad (2)$$

$$\sum_i q_i P_i(T) = P_L(T) \quad (3)$$

$$q_i \geq 0, i = 1, \dots, n \quad (4)$$

avec  $T$ : date d'immunisation. Le programme linéaire est construit ainsi:

```
obj <- df.o1$y * df.o1$PV01
A.VL <- df.o1$DP
A.PV01 <- df.o1$PV01
A.pos <- diag(nrow(df.o1))
N <- nrow(df.o1)

res <- lp(direction="max", objective.in=obj,
          const.mat=rbind(A.VL, A.PV01, A.pos),
          const.rhs=c(P.L, PV01.L, rep(0, N)),
          const.dir=c("=", "=", rep(">=", N)))
```

La solution ne comprend que deux titres:

Table 3: Immunisation d'un zero-coupon de maturité 2025

	Dt Emission	Dt Maturité	Coupon	Rendement	DP	PV01	Nb Titres
1	2018-06-01	2021-06-01	1.7	1.286754	101.4282	0.0021062	48950.21
28	2020-10-01	2031-10-01	5.9	3.462975	123.8963	0.0982992	33111.65

## Vérification

On peut vérifier que les valeurs et PV01 du portefeuille et du passif sont bien égales:

	Valeur	PV01
Actif	9,067,346	3,357.951
Passif	9,067,346	3,357.951

## Partie 2: Adossement flux à flux et immunisation

On considère maintenant un passif composé de plusieurs flux, comme indiqué dans le tableau ci-dessous:

On veut construire un portefeuille de rendement maximum tel que:

- les 4 premiers flux de passif sont adossés

Table 4: Echancier du passif à financer

Date	Montant
2021-10-01	1,000,000
2022-04-01	1,000,000
2022-10-01	1,000,000
2023-04-01	1,000,000
2023-10-01	1,000,000
2024-04-01	1,000,000
2024-10-01	10,000,000

- au 01/04/2023 (date d'immunisation), la PV et PV01 de l'actif et du passif sont égales.

On suppose que la courbe des taux au 01/04/2023 sera la même qu'au 17/03/2021.

### Prix et PV01 des obligations à la date d'immunisation

```
dt.immu <- as.Date("2023-04-01")
df.o2 <- df.o
df.o2$y = NA
df.o2$DP = NA
df.o2$PV01 = NA

for(i in seq(nrow(df.o2))) {
  res <- bond.calc(df.o2[i,], dt.immu)
  df.o2$y[i] <- res[1]
  df.o2$DP[i] <- res[2]
  df.o2$PV01[i] <- res[3]
}
```

Les premiers résultats sont résumés ci-dessous:

	Nom	dtE	dtM	Coupon	DP	PV01
7	Bond-7	2018-08-01	2023-08-01	5.0	104.2108	0.0034569
8	Bond-8	2020-12-01	2023-12-01	2.2	100.6115	0.0066208
9	Bond-9	2017-02-01	2024-02-01	3.5	101.4543	0.0083367
10	Bond-10	2019-06-01	2024-06-01	2.7	102.4954	0.0114095
11	Bond-11	2017-12-01	2024-12-01	4.8	105.1406	0.0166376
12	Bond-12	2020-04-01	2025-04-01	4.7	103.9339	0.0198004

### Prix et PV01 du passif à la date d'immunisation

```
idx <- df.flow$dt > dt.immu
df.immu <- df.flow[idx,]
```

```

P.L <- 0
PV01.L <- 0
for(i in seq(nrow(df.immu))) {
  res <- zc.calc(dt.immu, df.immu[i, "dt"])
  P.L <- P.L + df.immu[i, "vx"]*res[1]
  PV01.L <- PV01.L + df.immu[i, "vx"]*res[2]
}

```

Les valeurs à reproduire avec le portefeuille de titres sont:

Table 5: Valeur présente et PV01 du passif à la date d'immunisation

Valeur	PV01
11592567	1556.071

## Matrice des cash-flows

Les flux sont affectés aux dates de paiements immédiatement postérieures.

```

# cash flow matrix
cf.mat <- matrix(0, nrow=4, ncol=nrow(df.o1))
for(j in seq(nrow(df.o1))) {
  cf <- suppressWarnings(AnnivDates(Em=df.o1[j, "dtE"], Mat=df.o1[j, "dtM"],
                                   CpY=1, Coup=df.o1[j, "Coupon"], DCC=3,
                                   RV=100)$PaySched)

  N <- length(cf$CoupPayments)
  cf$CoupPayments[N] <- cf$CoupPayments[N] + 100
  dt.first = dt.calc
  for(i in seq(4)) {
    dt.last <- df.flow$dt[i]
    idx <- cf$CoupDates>dt.first & cf$CoupDates <= dt.last
    cf.mat[i,j] = sum(cf$CoupPayments[idx])
    dt.first=dt.last
  }
}

```

## Programme linéaire

On résoud le programme linéaire, en supposant un taux annuel de réinvestissement de 1% des excédents de liquidité

$$\max \sum_i q_i r_i \mathcal{PV}01_i \quad (5)$$

s.t.

$$\sum_i q_i F_i(1) - C(1) = L(1) \quad (6)$$

$$(1+r)C(t-1) + \sum_i q_i F_i(t) - C(t) = L(t) \quad t = 2, \dots, 4 \quad (7)$$

$$\sum_i q_i \mathcal{PV}01_i(T) = \mathcal{PV}01_L(T) \quad (8)$$

$$\sum_i q_i P_i(T) = P_L(T) \quad (9)$$

$$q_i \geq 0, i = 1, \dots, n$$

$$C(t) \geq 0, t = 1, \dots, 4$$

(10)

```
money.market.rate = 0.01
slack.mat = diag(rep(-1,4))
for(i in seq(3)) {
  slack.mat[i+1,i] = 1 + money.market.rate / 2
}

A1.mat <- cbind(cf.mat, slack.mat)
A2.mat <- rbind(df.o2$DP,
               df.o2$PV01)
A2.mat <- cbind(A2.mat, matrix(0, nrow=2, ncol=4))
A.mat <- rbind(A1.mat, A2.mat)
rhs <- c(df.flow$vx[1:4], P.L, PV01.L, rep(0, ncol(A.mat)))

A.mat <- rbind(A.mat,
              diag(1, nrow=ncol(A.mat)))

obj <- c(df.o2$y*df.o2$PV01, rep(0, 4))

res <- lp(direction="max", objective.in=obj,
          const.mat=A.mat,
          const.rhs=rhs,
          const.dir=c(rep("=", 6), rep(">=", 32)))
```

## Solution

Le coût de la solution est 14,933,305 euros.

```
idx <- res$solution > 0
df.sol <- df.o2[idx,c("Nom", "dtE", "dtM", "Coupon", "y")]
df.sol$Solution <- res$solution[idx]
kable(df.sol, format="latex", booktabs=T,
      format.args = list(big.mark=",", booktabs=T),
```



```
caption="Adossement flux à flux et immunisation.")%>%
kable_styling(position="center", latex_options = "HOLD_position")
```

Table 6: Adossement flux à flux et immunisation.

	Nom	dtE	dtM	Coupon	y	Solution
1	Bond-1	2018-06-01	2021-06-01	1.7	0.000000	3,957.455
3	Bond-3	2019-04-01	2022-04-01	1.7	0.000000	9,716.247
4	Bond-4	2017-10-01	2022-10-01	2.6	0.000000	4,308.485
6	Bond-6	2016-04-01	2023-04-01	1.2	0.000000	9,881.423
7	Bond-7	2018-08-01	2023-08-01	5.0	2.265704	93,490.830
28	Bond-28	2020-10-01	2031-10-01	5.9	3.462975	15,338.187

## Vérification

On vérifie que les flux générés par le portefeuille correspondent bien au passif, et que les contraintes sont vérifiées à l'horizon d'immunisation.

```
df.cashflow.match <- df.flow[1:4,]
df.cashflow.match$bond.cf <- cf.mat %% as.matrix(res$solution[1:28], ncol=1)
kable(df.cashflow.match, format="latex", booktabs=T,
      format.args = list(big.mark=",", scientific=F, booktabs=T),
      col.names = c("Date de flux", "Passif", "Portefeuille"),
      caption="Cash flow généré par le portefeuille durant les 4 premières périodes") %>%
kable_styling(position="center", latex_options="HOLD_position")
```

Table 7: Cash flow généré par le portefeuille durant les 4 premières périodes

Date de flux	Passif	Portefeuille
2021-10-01	1,000,000	1,000,000
2022-04-01	1,000,000	1,000,000
2022-10-01	1,000,000	1,000,000
2023-04-01	1,000,000	1,000,000

```
P.A <- sum(res$solution[1:28] * df.o2$DP)
PV01.A <- sum(res$solution[1:28] * df.o2$PV01)
df.at.immu <- data.frame(P=c(P.A, P.L), PV01=c(PV01.A, PV01.L))
rownames(df.at.immu)=c("Actif", "Passif")
kable(df.at.immu, format="latex", col.names = c("Valeur", "PV01"), booktabs=T, format.args = list(big.mark=",", scientific=F, booktabs=T),
      caption="Valeur et PV01 de l'actif et du passif à la date d'immunisation")%>%
kable_styling(position="center", latex_options="HOLD_position")
```

Table 8: Valeur et PV01 de l'actif et du passif à la date d'immunisation

	Valeur	PV01
Actif	11,592,567	1,556.071
Passif	11,592,567	1,556.071

## Autre solution

On peut également minimiser le cout du portefeuille à la date de calcul, sujet aux contraintes précédentes. Le reste du programme est inchangé.

```
obj <- c(df.o1$DP, rep(0, 4))

res.2 <- lp(direction="min", objective.in=obj,
            const.mat=A.mat,
            const.rhs=rhs,
            const.dir=c(rep("=", 6), rep(">=", 32)))
```

## Solution

Le coût de la solution est 14,918,746 euros.

```
idx <- res.2$solution > 0
df.sol <- df.o1[idx,c("Nom", "dtE", "dtM", "Coupon", "y")]
df.sol$solution <- res.2$solution[idx]
kable(df.sol, format="latex", booktabs=T,
      format.args = list(big.mark=",", booktabs=T),
      caption="Adossement flux à flux et immunisation pour un cout minimum à la date de calcul.") %>%
  kable_styling(position="center", latex_options = "HOLD_position")
```

Table 9: Adossement flux à flux et immunisation pour un cout minimum à la date de calcul.

	Nom	dtE	dtM	Coupon	y	Solution
1	Bond-1	2018-06-01	2021-06-01	1.7	1.286754	6,956.003
3	Bond-3	2019-04-01	2022-04-01	1.7	1.793125	5,250.133
4	Bond-4	2017-10-01	2022-10-01	2.6	1.996413	9,746.589
6	Bond-6	2016-04-01	2023-04-01	1.2	2.169548	5,339.385
9	Bond-9	2017-02-01	2024-02-01	3.5	2.405194	59,358.107
12	Bond-12	2020-04-01	2025-04-01	4.7	2.654394	53,595.933

## Vérification

On vérifie que les flux générés par le portefeuille correspondent bien au passif, et que les contraintes sont vérifiées à l'horizon d'immunisation.

```
df.cashflow.match <- df.flow[1:4,]
df.cashflow.match$bond.cf <- cf.mat %*% as.matrix(res.2$solution[1:28], ncol=1)
kable(df.cashflow.match, format="latex", booktabs=T,
      format.args = list(big.mark=",", scientific=F, booktabs=T),
      col.names = c("Date de flux", "Passif", "Portefeuille"),
      caption="Cash flow généré par le portefeuille durant les 4 premières périodes") %>%
  kable_styling(position="center", latex_options="HOLD_position")
```

Table 10: Cash flow généré par le portefeuille durant les 4 premières périodes

Date de flux	Passif	Portefeuille
2021-10-01	1,000,000	1,000,000
2022-04-01	1,000,000	1,000,000
2022-10-01	1,000,000	1,000,000
2023-04-01	1,000,000	1,000,000

```
P.A <- sum(res.2$solution[1:28] * df.o2$DP)
PV01.A <- sum(res.2$solution[1:28] * df.o2$PV01)
df.at.immu <- data.frame(P=c(P.A, P.L), PV01=c(PV01.A, PV01.L))
rownames(df.at.immu)=c("Actif", "Passif")
kable(df.at.immu, format="latex", col.names = c("Valeur", "PV01"), booktabs=T, format.args = list(big.mark=",",
caption="Valeur et PV01 de l'actif et du passif à la date d'immunisation"))>%
  kable_styling(position="center", latex_options="HOLD_position")
```

Table 11: Valeur et PV01 de l'actif et du passif à la date d'immunisation

	Valeur	PV01
Actif	11,592,567	1,556.071
Passif	11,592,567	1,556.071