Medaf

Le MEDAF ou CAPM (Capital Asset Pricing Model), suppose qu'il existe un actif sans risque avec un rendement Rf. Dans ce modèle chaque portefeuille efficient est une combinaison de l'actif sans risque et du portefeuille de marché M, qui correspond au point de tangence entre les deux frontières efficientes (avec et sans l'actif sans risque).

```
library("readxl")
library("dplyr")
library("quantmod")
library("purrr")
library("PerformanceAnalytics")
library("timetk")
library("tidyr")
library("tidyverse")
library("tidyquant")
library("broom.mixed")
setwd("D:/Documents/Cours M2 MoSEF/Projets M2 MoSEF/Finance Base/")
```

```
table <-read_excel("Fidelity Funds - European Growth Fund.xlsx", skip = 11)
table <- table[3:227, ]
table$Shift <- lag(table$PX_LAST ,1,na.pad = TRUE)
table$Rendement = (table$PX_LAST - table$Shift) /table$Shift
table$'Shift'=NULL
table$'PX_LAST'=NULL
table$Rf= as.numeric(table$Rf)
```

```
#Extraction de l'ETF sur Yahoo finance
spy_monthly_xts <-
  getSymbols("EXW1.DE",
             src = 'yahoo',
             from = "2000-12-01",
             to = "2019-09-30",
             auto.assign = TRUE,
             warnings = FALSE) %>%
  map(~Ad(get(.))) %>%
  reduce(merge) %>%
  `colnames<-`("EXW1.DE") %>%
  to.monthly(indexAt = "last", OHLC = FALSE)
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
## Warning: EXW1.DE contains missing values. Some functions will not work if
## objects contain missing values in the middle of the series. Consider using
## na.omit(), na.approx(), na.fill(), etc to remove or replace them.
```

```
## Warning in to.period(x, "months", indexAt = indexAt, name = name, ...): missing
## values removed from data
```

```
market_returns_xts <-
  Return.calculate(spy_monthly_xts, method = "log") %>%
  na.omit()

market_returns_tidy <-
  market_returns_xts %>%
  tk_tbl(preserve_index = TRUE, rename_index = "date") %>%
  na.omit() %>%
  select(date, returns = EXW1.DE)
```

```
#table$Dates <- as.Date(table$Dates)
#table <- table[-c(225,219,108),]
table <- table[-c(225,1),]

a <- table %>%
  mutate(market_returns = market_returns_tidy$returns)
a <- a[-1,]
a$Rf <- a$Rf/100

#Calcul du Bêta du CAPM
a$cov <- cov(a$Rendement, a$market_returns)
a$beta <- a$cov/ var(a$market_returns)
mean(a$beta) # Bêta du CAPM calculé à la mains
```

```
## [1] 0.3927188
```

```
# indépendamment, avec une régression classique, on retrouve le Bêta du marché par la modélisation
beta_dplyr_byhand <-  a %>%
  do(model = lm(Rendement ~ market_returns, data = a)) %>%
  tidy(model) %>%
  mutate(term = c("alpha", "beta"))

beta_dplyr_byhand
```

```
## # A tibble: 2 x 5
##   term  estimate std.error statistic  p.value
##   <chr>    <dbl>     <dbl>     <dbl>    <dbl>
## 1 alpha  0.00417   0.00265      1.57 1.17e- 1
## 2 beta   0.393     0.0499       7.86 1.67e-13
```

```
a$pred <- 0.004174 + 0.392719 * a$market_returns #On retrouve bien le beta (=0.39)
```

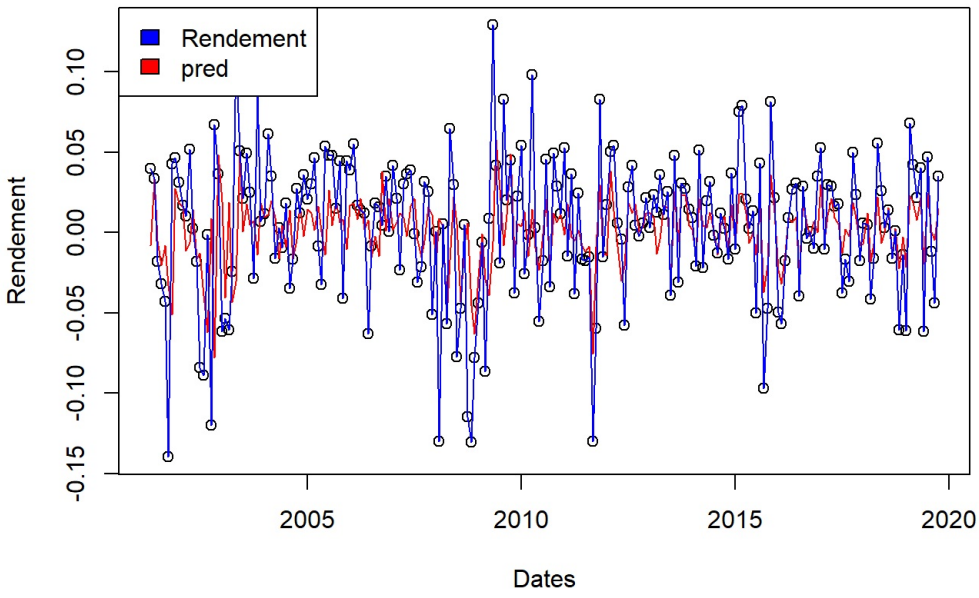
```
########  Modélisation par le CAPM   #########

a$`Rm-Rf` <- a$market_returns - a$Rf


#Estimation des rendements par le MEDAF
# Pour cela, on réutilise le bêta déterminé precédement: calculé à la mains ou par modélisation
a$prévision_MEDAF <- a$Rf + a$beta * a$`Rm-Rf`

#Qualite de l'estimation
a$diff <- a$Rendement - a$pred
mean(a$diff)
```

```
## [1] -4.427041e-07
```

```
StdDev(a$diff)
```

```
##              [,1]
## StdDev 0.03941262
```

```
#Représentation du modèle
plot(a$Dates, a$Rendement, ylab = "Rendement", xlab = "Dates")
lines(a$Dates , a$prévision_MEDAF, col = "red")
lines(a$Dates , a$Rendement, col = "blue")
legend("topleft",
       c("Rendement","pred"),
       fill=c("blue","red"))
```




```
#Calcul R²
sum((a$prévision_MEDAF-mean(a$prévision_MEDAF))^2)/sum((a$Rendement-mean(a$Rendement))^2)
```

```
## [1] 0.2196519
```

```
model=lm(Rendement ~ market_returns, data = a)
summary(model)  # verif on retrouve bien le même R² que celui calculé
```

```
##
## Call:
## lm(formula = Rendement ~ market_returns, data = a)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.141959 -0.017768  0.003054  0.019195  0.141486
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.004174   0.002651   1.574    0.117
## market_returns 0.392719   0.049948   7.863 1.67e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0395 on 220 degrees of freedom
## Multiple R-squared:  0.2194, Adjusted R-squared:  0.2158
## F-statistic: 61.82 on 1 and 220 DF,  p-value: 1.669e-13
```

```
#Calcul ratio de Sharp
Sharp=(mean(a$Rendement)-mean(a$Rf))/StdDev(a$Rendement)
Sharp
```

```
##             [,1]
## StdDev 0.06963434
```

Ce ratio est donc la pente de la CML. Comme proposé par Sharp ce ratio peut être considéré comme une mesure de performance on peut prendre le numérateur comme la prime de risque (positive ou négative) et le dénominateur comme un indicateur de risque. Autrement dit, c'est le rapport entre l'excès de rendement moyen du portefeuille et la mesure totale du risque du portefeuille. Pour exemplifier, le gérant d'un fonds peut regarder si son excès de rendement moyen est suffisant pour compenser un risque plus élevé que celui du portefeuille de marché. Si un portefeuille est bien diversifié, son ratio de Sharp est proche de celui du portefeuille de marché. Plus le ratio de Sharp est élevé mieux c'est.