

TP3 - Analyse financière

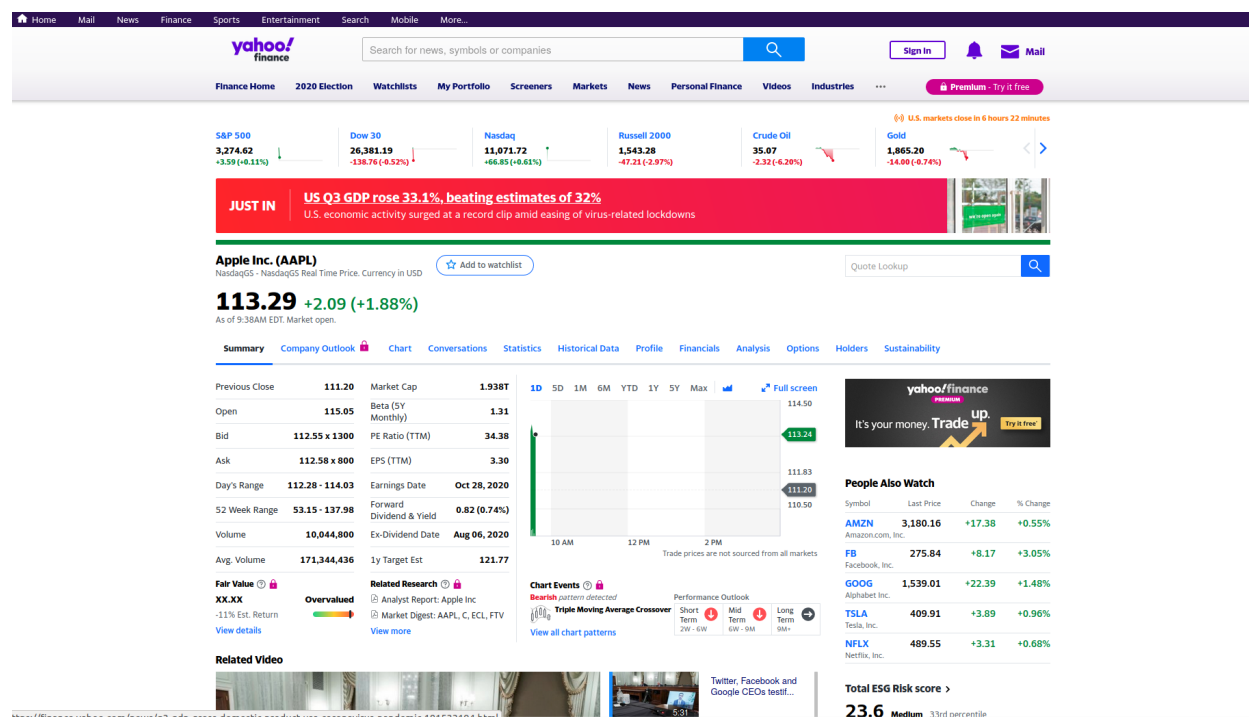
Sonia Tiew

29/10/2020

Installation de la librairie quantmode

```
library("quantmod")
```

Téléchargement des données financières à partir de YAHOO



Q: Inspecter la fonction `getSymbols` avec `help()` et inspecter le dataframe crée `df`, commentez les colonnes:

```
df <- data.frame(getSymbols("AAPL", auto.assign = F))  
head(df)  
tail(df)
```

Q. Rappel: pour s'échauffer avec les dataframes:

1. Sélectionner les 3ères lignes.
2. Sélectionner toutes les lignes et les colonnes 1 et 4.
3. Sélectionner toutes les colonnes pour la ligne "2015-12-31"
4. Quels sont les noms des colonnes ?
5. Renommer les colonnes comme ceci: Open, High, Low, Close, Volume, Adjusted
6. Renommer df par AAPL

Q. A l'aide de la fonction `Cl()` fournie par `quantmod`, extraire le prix de fermeture de toute la période. Puis représenter le avec la fonction `plot()` en ajoutant l'argument `type='l'`

Q. Faire la même chose avec les prix ajustés (Adjusted prices) avec 2 méthodes différentes. 1. Trouver la fonction qui permet de récupérer les prix ajustés.

2 Le faire en sélectionnant la colonne

Q. Combien y a t'il de valeurs de "prix ajustés" ?

Q. Calculer le rendement journalier (en pourcentage) entre :

- "Jour 1" et le "Jour 2"

- "Jour 2" et le "Jour 3"

Q. Faire une boucle, pour récupérer les rendements journaliers (en pourcentage) des 10^{ers} jours:

Aide Pour faire une boucle :

```
for (i in 1:10){  
  print(i)  
}
```

```
## [1] 1  
## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5  
## [1] 6  
## [1] 7  
## [1] 8  
## [1] 9  
## [1] 10
```

Q: Heureusement que dans la librarire *quantmod* on peut obtenir un rendement sur la période choisie (journalier, hebdomadaire, mensuel, annuel...). Commentez:

```
dailyReturn(Ad(as.xts(AAPL))) #daily return ne s'applique pas sur un dataframe mais uniquement sur les xts  
head(dailyReturn(Ad(as.xts(AAPL))))
```

Q: Faites un plot des rendements

Q: Exécutez et Commentez:

```
head(cumprod(1+dailyReturn(Ad(as.xts(AAPL)))))  
plot(cumprod(1+dailyReturn(Ad(as.xts(AAPL))))) , type = "l")
```

Q: Faire une boucle, pour récupérer les log rendements journaliers des 10^{ers} jours:

Q: Trouver une alternative avec la libraririe *quantmode* (indice: *diff*).

Représentez le graphiquement

Q: Autre alternative, découvrons la fonction *apply*. Au lieu de faire une bouclé, on peut "vectoriser" le travail ainsi:

```
#Exemple, on applique sur les colonnes "close" et "adjusted"  
#la fonction logarithme sur les lignes ( margin=2 pour appliuer la fonction sur les lignes)  
apply(AAPL[,c("Close","Adjusted")] , MARGIN = 2 , log)  
  
#ON applique par dessus en couche, la fonction diff:  
apply(apply(AAPL[,c("Close","Adjusted")] , MARGIN = 2 , log),2, diff)  
  
#Stockons le dans une variable pour pouvoir l'utiliser  
compareAdtoCl <-apply(apply(AAPL[,c("Close","Adjusted")] , MARGIN = 2 , log),2, diff)  
  
#Commentez
```

```
plot(compareAdtoCl[,1] , type = "l")
lines(compareAdtoCl[,2] , type = "l" , col = "red")
```

On va plutôt regarder les sommes cumulées:

```
compareAdtoClCumSum <- data.frame(apply(compareAdtoCl,2,cumsum))
compareAdtoClCumSum
plot(compareAdtoClCumSum[,1] , type = 'l')
lines(compareAdtoClCumSum[,2] , type = "l" , col = "red")
```

Q: Tester la fonction *barChart* suivante:

```
barChart(getSymbols("AAPL", auto.assign = F) ,theme=chartTheme('black'))
```

Q: Ajouter le MACD de la manière suivante:

```
barChart(getSymbols("AAPL", auto.assign = F) ,theme=chartTheme('white'))
addMACD()
addBBands()
```

Librarie: Performances Analytiques

```
#install.packages("PerformanceAnalytics")
library(PerformanceAnalytics)
```

Pour utiliser les fonctions de la librarie *PerformanceAnalytics* il faut avoir des objets type *xts*, on va donc convertir comme ci dessous. Commentez la figure:

```
data <- as.xts(compareAdtoCl)
charts.PerformanceSummary(data, main = "Comparaison")
```

Comparons les données de 2 firmes

ON choisit de calculer le rendement journalier des prix ajustés avec la fonction *dailyReturn()* pour deux compagnies: - Tesla : TSLA et stocker les informations dans une variable appelée *TSLA* et appliquer *charts.PerformanceSummary* - Microsoft: MSFT et stocker les informations dans une variable appelée *TSLA* et appliquer *charts.PerformanceSummary*

Q: Comparer les dates de début.

Par la suite on cherche à concaténer les deux tables, il faut donc renommer les colonnes pour éviter toute confusion

```
colnames(MSFT) <- "MSFT"
colnames(TSLA) <- "TSLA"
```

Q: Concatenez maintenant avec la fonction *merge()* , que remarquez vous ? Retirez les lignes contenant des *NA*. (indice: soit à l'aide d'un argument de la fonction *merge* ou sot avec la fonction *na.omit*). Stockez ce tableau concaténé dans une variable appelée *MSFTvsTSLA*:

```
#solution1
MSFTvsTSLA <- merge(TSLA,MSFT,all=F)
#solution2
MSFTvsTSLA <- merge(TSLA,MSFT)
MSFTvsTSLA <- na.omit(MSFTvsTSLA)
```

Q: Appliquer *charts.PerformanceSummary* sur ce nouveau tableau *MSFTvsTSLA*

```
charts.PerformanceSummary(MSFTvsTSLA , main = "Comparaison entre TSLA et MSFT")
```

Balance entre : return/risk

On va calculer le ratio de Sharp (mesure de la rentabilité) du portefeuille risqué. Rappel: $\text{Ratio de Sharp} = (\text{Rendement} - \text{taux sans risque}) / \text{volatilité}$

-Si le ratio est négatif, on en conclut que le portefeuille sous performe un placement sans risque et donc il n'est pas logique d'investir dans un tel portefeuille.

-Si le ratio est compris entre 0 et 1, cela signifie que l'excédent de rendement par rapport au taux sans risque est plus faible que le risque pris.

-Si le ratio est supérieur à 1, alors le portefeuille surperforme un placement sans risque et donc il génère une plus forte rentabilité.

Plus le ratio est élevé et plus le portefeuille est performant.

Q: Tester et commentez, qui de Microsoft ou Tesla possède un plus gros ratio

```
table.AnnualizedReturns(MSFTvsTSLA , scale=252 , Rf = 0.005/252) #252 le nombre de jours ouvrés dans l'
```

Calculer le VaR pour les portefeuilles des stocks

ON va reprendre les données de Microsoft, mais cette fois ci entre le 1er janvier 2013 et le 1er janvier 2017. On peut préciser la source (yahoo) car il en existe d'autres (Google Finance, FRED, Oanda, local databases, CSVs).

Attention pour la date, il faut respecter le format suivant: "yyyy-mm-dd".

```
#On fixe les dates qui définissent la période
```

```
debut="2013-01-01"
```

```
fin="2017-01-01"
```

```
#on récupère les prix
```

```
MSFT <- getSymbols("MSFT", src = "yahoo", auto.assign = F, from = debut, to = fin)
```

```
#On récupère les rendements journaliers
```

```
MSFT.daily <- dailyReturn(MSFT)
```

On va calculer le Value-at-Risk (VaR) qui représente "la perte potentielle maximale d'un investisseur sur la valeur d'un actif ou d'un portefeuille d'actifs financiers compte tenu d'un horizon de détention et d'un intervalle de confiance."

Q: Inspecter la fonction VaR avec un *help* et calculer le VAR avec un niveau de confiance de 0.95 et avec la méthode *historical*. Faire la même chose avec un niveau de confiance de 0.99.

Quelles sont les autres méthodes pour calculer le VAR ?

Q: Comparaison du VAR avec les 3 méthodes différentes sur les 3 portefeuilles **Etapas:**

1. Récupérer les prix ajustés pour: MSFT, AAPL et TSLA. 2. Fusionner les prix en un seul tableau appelé *all.adprices* (n'oubliez pas de retirer les valeurs manquantes NA)

3. Cette fois ci, au lieu d'utiliser le *DailyReturn* pour le rendement journalier, on va utiliser la fonction *ROC(all.adprices , type = "discrete")* pour calculer le changement sur une série de période. Appelez l'objet *all.returns*. et renommez les colonnes avec le noms des 3 entreprises.

4. Calculez sur *all.returns* le VAR avec un intervalle de confiance de 0.95 et le *portfolio_method = "component"* et la *method = "historical"*

5. Commentez le résultat; on s'aperçoit, d'après les contributions, que TSLA est plus à risque que AAPL et

MSFT.

5bonus. Explorer les fonctions ETL, ES.

6. Comparez les 3 méthodes pour calculer le VAR (avec $p=0.95$ et `portfolio_method = "single"`) : Hist, Gaus, Mod. (Conseil, stockez les séparément dans les variables: *Var.Hist*, *VAR.Gaus* et *VAR.mod* puis créer un *data.frame* nommé *All.VAR* pour les rassembler)

7. A la fin vous devez obtenir le tableau suivant:

—	MSFT	AAPL	TSLA
Hist	-0.02290069	-0.02596607	-0.04905951
Gaus	-0.02547391	-0.02817714	-0.05629856
Mod	-0.02309386	-0.02749275	-0.05226586

8. Utilisez la fonction *abs* sur le tableau pour avoir des valeur positives.

Q: On va représenter ces informations graphique à l'aide de la librarire *ggplot*.

Avant cela, exécutez:

```
ALL.VAR$Type <- c("Hist","Gaus","Mod")
library("reshape2")
library("ggplot2")

#si elles ne sont pas installées, faire:
#install.packages("reshape2")
#install.packages("ggplot2")
plotVar <- melt(ALL.VAR, variable.name = "Entreprises", value.name = "VaR")
ggplot(plotVar , aes(x=Type,y=VaR , fill = Entreprises)) +
  geom_bar(stat = "identity" , position = "dodge" )
```

Découvrir la librairie highcharter pour une visalisation interactive

```
install.packages("highcharter")

library("highcharter")

lot1 <- highchart(type = "stock") %>%
  hc_title(text = "Stocks Evolution of ") %>%
  hc_add_series(Ad(AAPL), name="Apple") %>%
  hc_add_series(Ad(MSFT), name="Microsoft")
```