

Avaliação Parcial 2

Alysson da Silva Moura

24/06/2022

Questões

Baixar dados de 01/01/2021 a 15/06/2022 da ação selecionada. Vamos analisar 6 trimestres.

1. Introdução (2 PONTOS) a) Análise estatística e gráfica da série histórica em nível e de retornos. b) Baixe os dados de uma empresa concorrente e observe se o comportamento gráfico é semelhante (colocar as duas séries no mesmo gráfico).
2. Value at Risk: Considere um investimento de R\$100mil e o $Z=-1,645$ (nível de confiança de 95%). (4 PONTOS) a) Calcular para cada um dos 6 trimestres o Value at Risk Relativo e Absoluto b) Usar gráficos de barras para analisar o comportamento do risco ao longo dos 6 trimestres.
3. CAPM: Considere o ativo livre de risco 6%a.a. e base 252. (4 PONTOS)
 - a) Calcular para cada um dos 6 trimestres o BETA do CAPM.
 - b) Usar gráficos de barra para analisar o comportamento do BETA ao longo dos 6 trimestres.
 - c) Analisar dentro de cada trimestre se o retorno médio do ativo é justo pela equação do CAPM, e se o ativo está sobreprecificado ou subprecificado?

Pacotes

```
library(quantmod)
library(tidyverse)
library(patchwork)
library(stargazer)
```

Os *ticks* escolhidos foram *BTC* e *ETH*, vamos utilizar o pacote `quantmod` podemos usar a função `quantmod::getSymbols()` para pegar os dados do yahoo:

```
btc <- quantmod::getSymbols("BTC-USD",
                             src = "yahoo",
                             from = as.Date("2021-01-01"),
                             to = as.Date("2021-06-15"), auto.assign = F)
eth <- quantmod::getSymbols("ETH-USD", src = "yahoo",
                             from = as.Date("2021-01-01"),
                             to = as.Date("2021-06-15"), auto.assign = F)
```

Primeiro pegaremos só o preço de fechamento dessas ações:

```
btc_p <- btc$`BTC-USD.Close`  
eth_p <- eth$`ETH-USD.Close`
```

Para juntar conseguir juntar os dois stocks precisaremos sair do formato `xts` para um `DataFrame`, para isso foi criada uma função chamada `into_df` que vai converter esse formato. A partir disso, pode-se *plotar* os gráficos do BTC-USD e do ETH-USD dos retornos diários dos ativos, e outra para retornar os ativos em nível.

```
into_df <- function (data, ticker) {  
  data.frame(date = zoo::index(data),  
             price = zoo::coredata(data),  
             ticker = rep(ticker, length(zoo::coredata(data)))) %>%  
  setNames(c("date", "price", "ticker")) %>%  
  dplyr::arrange(dplyr::desc(date))  
}
```

Para conseguir os retornos diários em porcentagem foi utilizado a função `quantmod::dailyReturn()` que dá os retornos diários do ativo, e passaremos os resultados para a função `into_df` que retorna um `DataFrame` com a data, valor e o *ticker* informado

```
btc_r_df <- btc_p %>% dailyReturn() %>% into_df("btc")  
eth_r_df <- eth_p %>% dailyReturn() %>% into_df("eth")
```

```
head(btc_r_df)
```

```
##      date      price ticker  
## 1 2021-06-15 0.004669321   btc  
## 2 2021-06-14 0.028661855   btc  
## 3 2021-06-13 0.099721319   btc  
## 4 2021-06-12 -0.047727642   btc  
## 5 2021-06-11 0.017214062   btc  
## 6 2021-06-10 -0.017205017   btc
```

Foi combinado os dois `DataFrame`'s em um só, na variável `combined_df_r`, e, a partir dela, foi criado o plot dos retornos diários

```
combined_df_r <- rbind(btc_r_df, eth_r_df)
```

```
dailyPlot <- combined_df_r %>%  
  ggplot(aes(x = date, y = price, group = ticker, colour = ticker)) +  
  geom_line(show.legend = F) +  
  labs(y = "Retorno (%)", x = "", title = "Retorno (em %)") +  
  facet_wrap(~ticker, scales = "free_y", ncol = 1) +  
  scale_y_continuous(labels = scales::percent) +  
  scale_color_discrete(limits = c("btc", "eth"),  
                      labels = c("BTC-US$", "ETH-US$")) +  
  theme_minimal(base_size = 9) +  
  theme(plot.title = element_text(hjust = 0.5))
```

e para o gráfico em nível:

```

btc_df <- btc_p %>% into_df("btc")
eth_df <- eth_p %>% into_df("eth")

combined_df <- rbind(btc_df, eth_df)

```

```

levelPlot <- combined_df %>%
  ggplot(aes(x = date, y = log(price), colour = ticker)) +
  geom_line() +
  labs(y = "log(Preço (US$))", x = "",
       title = paste0("log(Preço (US$)) dos Stocks"),
       colour = "Stock") +
  scale_color_discrete(limits = c("btc", "eth"),
                      labels = c("BTC-US$", "ETH-US$")) +
  theme_minimal(base_size = 9) +
  theme(plot.title = element_text(hjust = 0.5))

```

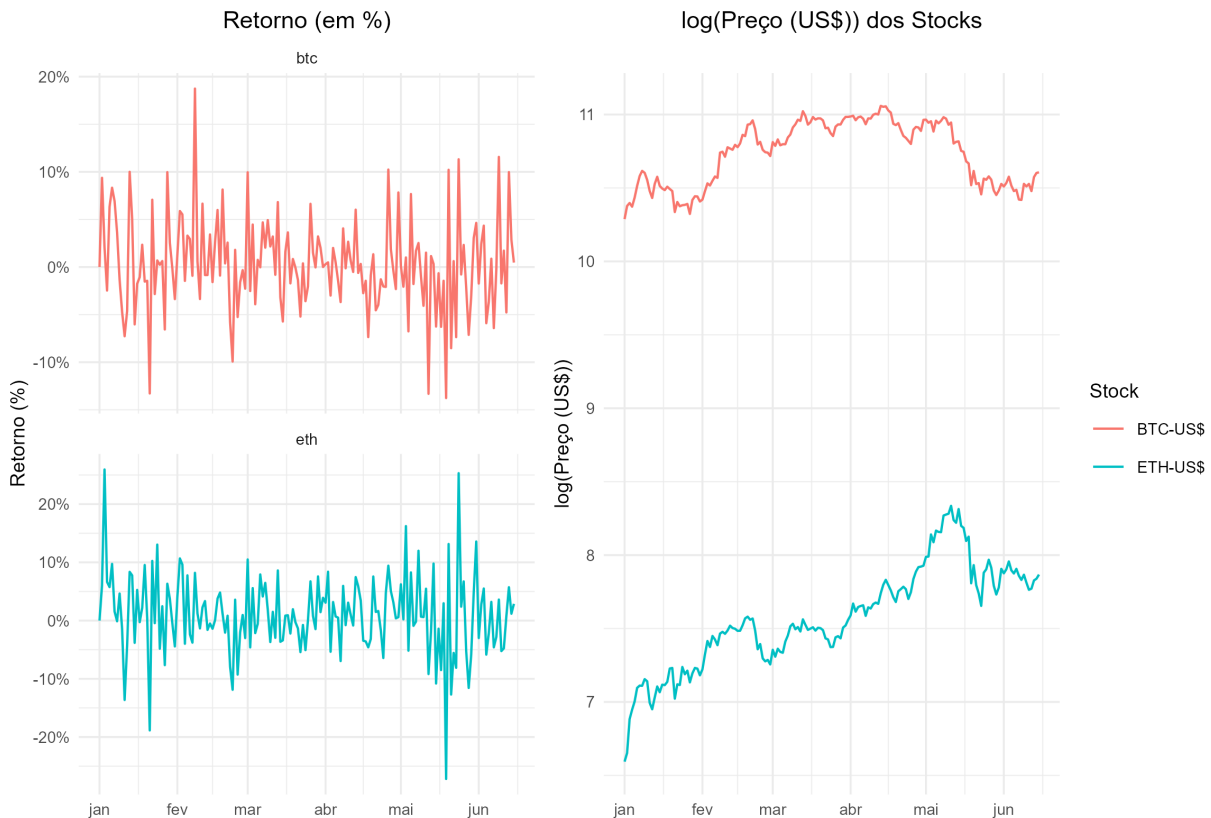
Então chamamos a função `plotData` para os valores do `btc_p` e `eth_p` e utilizamos o pacote `patchwork` para dividir os *plots*:

```

plots <- dailyPlot | levelPlot

plots

```



E as estatísticas descritivas:

Table 1: Estadísticas descriptivas

Statistic	N	Mean	St. Dev.	Min	Max
BTC-USD	166	46,931.030	9,817.876	29,374.150	63,503.460
ETH-USD	166	2,071.129	724.679	730.368	4,168.701

```
bhp2 = BHP$BHP.AX.Close
# covert to returns

bhp_ret = dailyReturn(bhp2, type = "log")
den1_r = coredata(bhp_ret)
den1_bhp = dnorm(x = den1_r, mean = mean(den1_r), sd = sd(den1_r))
data_rd = data.frame(den1_r, den1_bhp)
```