

Retornos e Portfólio

Flávio Vieira

01/03/2021

Neste Exercício vamos criar um portfólio de ações brasileiras e compararemos os retornos desse portfólio com os retornos individuais de cada ação.

Importando os pacotes

```
library(tidyverse)
library(lubridate)
library(highcharter)
library(timetk)
library(tibble)
library(quantmod)
library(PerformanceAnalytics)
library(scales)
```

Retorno de Ativos

Nessa seção pegaremos os dados brutos de 7 ativos individuais do índice bovespa e transformá-los em retornos mensais para uma única carteira.

Nossa portfólio consiste nas seguintes ações:

* MGLU3 * TAEE11 * BBDC4 * B3SA3 * BRDT3 * BBSE3 * VALE3

Cada um desses objetos conterá retornos mensais do portfólio de 31 de janeiro de 2018 a 31 de outubro de 2020

Além disso, seguiremos estas etapas:

- 1) Importar preços diários da internet
- 2) Transforme os preços diários em preços mensais
- 3) Transforme os preços mensais em retornos mensais
- 4) Visualize retornos mensais
- 5) Calcular os retornos mensais do portfólio com base nos retornos mensais dos ativos e pesos
- 6) Visualize os retornos do portfólio

Importando os preços diários

```
symbols = c("MGLU3.SA", "TAEE11.SA", "BBDC4.SA", "B3SA3.SA", "BRDT3.SA", "BBSE3.SA", "VALE3.SA")
```

```
prices =getSymbols(symbols,
                    src = 'yahoo',
                    from = "2019-01-31",
                    auto.assign = TRUE,
                    warnings = FALSE) %>%
  map(~Ad(get(.))) %>%
  reduce(merge) %>%
  `colnames<-`(symbols)
```

```
head(prices, 3)
```

```
##           MGLU3.SA TAE11.SA BBDC4.SA B3SA3.SA BRDT3.SA BBSE3.SA VALE3.SA
## 2019-01-31 5.435615 21.58720 30.92408 29.56385 22.32258 25.78769 42.54688
## 2019-02-01 5.427420 21.58720 30.54147 30.00538 22.28907 26.06997 43.24820
## 2019-02-04 5.374906 21.66273 31.21747 30.31539 22.49010 26.25263 41.78011
```

Agora converteremos os preços diários em retornos mensais logarizados.

```
# Transforme os preços diários em preços mensais

Preco_Mensal = to.monthly(prices,
                           indexAt = "lastof",
                           OHLC = FALSE)

# Transforme os preços mensais em retornos mensais

retorno_mensal =
  Return.calculate(Preco_Mensal,
                   method = "log") %>%
  na.omit()

tail(retorno_mensal, 3)
```

```
##           MGLU3.SA TAE11.SA BBDC4.SA B3SA3.SA BRDT3.SA
## 2020-12-31 0.06603387 0.01522656 0.12621971 0.09914126 0.09426917
## 2021-01-31 0.01274406 -0.02577862 -0.09186911 -0.02677220 0.04980104
## 2021-02-28 -0.03626501 -0.05656219 -0.06783692 -0.07805190 -0.15752404
##           BBSE3.SA VALE3.SA
## 2020-12-31 0.03678023 0.11435834
## 2021-01-31 -0.06880003 0.00570127
## 2021-02-28 -0.06258598 0.07352313
```

Usando Tidyverse

```
retorno_mensal_tidyverse =
  prices %>%
  to.monthly(indexAt = "lastof", OHLC = FALSE) %>%
  data.frame(date = index(.)) %>%
  remove_rownames() %>%
  gather(asset, prices, -date) %>%
  group_by(asset) %>%
  mutate(returns = (log(prices) - log(lag(prices)))) %>%
  select(-prices) %>%
  spread(asset, returns) %>%
```

```
select(date, symbols)
retorno_mensal_tydiverse =
  retorno_mensal_tydiverse %>%
  na.omit()
```

```
tail(retorno_mensal_tydiverse, 3)
```

```
## # A tibble: 3 x 8
##   date      MGLU3.SA TAE11.SA BBDC4.SA B3SA3.SA BRDT3.SA BBSE3.SA VALE3.SA
##   <date>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 2020-12-31  0.0660    0.0152    0.126    0.0991    0.0943    0.0368    0.114
## 2 2021-01-31  0.0127   -0.0258   -0.0919  -0.0268    0.0498   -0.0688    0.00570
## 3 2021-02-28 -0.0363   -0.0566   -0.0678  -0.0781   -0.158   -0.0626    0.0735
```

Para que os retornos dos ativos sejam organizados, precisamos de uma coluna chamada “data”, uma coluna chamada “ativo” e uma coluna chamada “retornos”

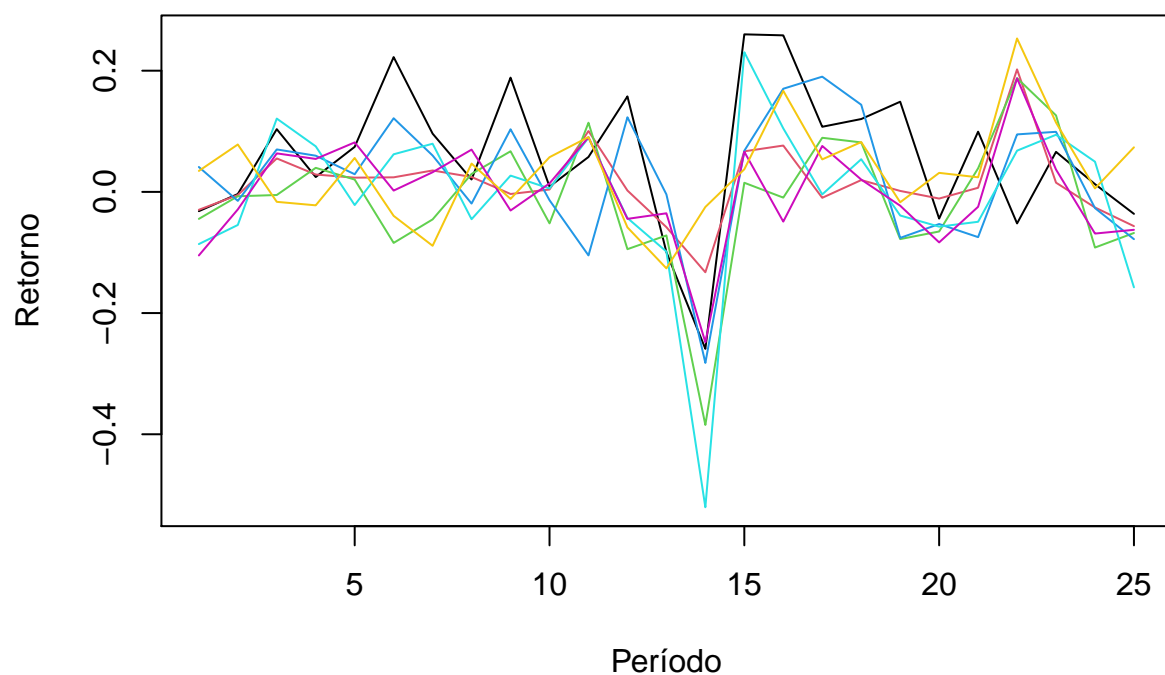
```
asset_returns_long =
  retorno_mensal_tydiverse %>%
  gather(asset, returns, -date) %>%
  group_by(asset)
tail(asset_returns_long, 3)
```

```
## # A tibble: 3 x 3
## # Groups:   asset [1]
##   date      asset  returns
##   <date>    <chr>    <dbl>
## 1 2020-12-31 VALE3.SA  0.114
## 2 2021-01-31 VALE3.SA  0.00570
## 3 2021-02-28 VALE3.SA  0.0735
```

Vizualizando os retornos.

```
ts.plot(retorno_mensal, col = 1:7, main = "Retorno Mensal Logarizado",
  xlab = "Período", ylab = "Retorno")
```

Retorno Mensal Logarizado



```
asset_returns_long %>%  
  ggplot(aes(x = returns)) +  
  geom_density(aes(color = asset), alpha = 1) +  
  geom_histogram(aes(fill = asset), alpha = 0.45, binwidth = .01) +  
  guides(fill = FALSE) +  
  facet_wrap(~asset) +  
  ggtitle("Monthly Returns Since 2019") +  
  xlab("monthly returns") +  
  ylab("distribution") +  
  theme_update(plot.title = element_text(hjust = 0.5))
```

Monthly Returns Since 2019



Construindo o Portfólio

Primeiramente vamos criar um vetor com o peso de cada ativo no portfólio

```
w = c(0.21, 0.13, 0.12, 0.22, 0.14, 0.095, 0.085)
```

```
# Verificando se os pesos e os ativos estão alinhados
```

```
tibble(w, symbols)
```

```
## # A tibble: 7 x 2
##       w symbols
##   <dbl> <chr>
## 1 0.21  MGLU3.SA
## 2 0.13  TAE11.SA
## 3 0.12  BBDC4.SA
## 4 0.22  B3SA3.SA
## 5 0.14  BRDT3.SA
## 6 0.095 BBSE3.SA
## 7 0.085 VALE3.SA
```

```
# Separando os pesos por ativo
```

```
w1 = w[1]
w2 = w[2]
w3 = w[3]
```

```

w4 = w[4]
w5 = w[5]
w6 = w[6]
w7 = w[7]

# Separando os ativos

asset1 = retorno_mensal[1]
asset2 = retorno_mensal[2]
asset3 = retorno_mensal[3]
asset4 = retorno_mensal[4]
asset5 = retorno_mensal[5]
asset6 = retorno_mensal[6]
asset7 = retorno_mensal[7]

# Portfólio

port =
  (w1 * asset1) +
  (w2 * asset2) +
  (w3 * asset3) +
  (w4 * asset4) +
  (w5 * asset5) +
  (w6 * asset6) +
  (w7 * asset7)

```

Retorno do Portfólio

```

portfolio_return =
  Return.portfolio(retorno_mensal,
    weights = w,
    rebalance_on = "months") %>%
  `colnames<-`("returns")
head(portfolio_return, 3)

```

```

##              returns
## 2019-02-28 -0.025925040
## 2019-03-31 -0.009304482
## 2019-04-30  0.065271675

```

Retorno do Portfólio (Usando Tydiverse)

```

# Nossa primeira tarefa é adicionar uma coluna de pesos à tabela
# Cada ativo deve ser ponderado de acordo com o vetor w.

asset_returns_long %>%
  group_by(asset) %>%
  mutate(weights = case_when(asset == symbols[1] ~ w[1],
                             asset == symbols[2] ~ w[2],
                             asset == symbols[3] ~ w[3],
                             asset == symbols[4] ~ w[4],
                             asset == symbols[5] ~ w[5],

```

```

        asset == symbols[6] ~ w[6],
        asset == symbols[7] ~ w[7])) %>%

head(3)

## # A tibble: 3 x 4
## # Groups:   asset [1]
##   date      asset    returns weights
##   <date>    <chr>    <dbl>   <dbl>
## 1 2019-02-28 MGLU3.SA -0.0318    0.21
## 2 2019-03-31 MGLU3.SA -0.00352    0.21
## 3 2019-04-30 MGLU3.SA  0.103      0.21

# Em seguida, precisamos implementar a equação para os retornos da carteira.

port_return_tydiverse =
  asset_returns_long %>%
  group_by(asset) %>%
  mutate(weights = case_when(asset == symbols[1] ~ w[1],
                             asset == symbols[2] ~ w[2],
                             asset == symbols[3] ~ w[3],
                             asset == symbols[4] ~ w[4],
                             asset == symbols[5] ~ w[5],
                             asset == symbols[6] ~ w[6],
                             asset == symbols[7] ~ w[7]),
         weighted_returns = returns * weights) %>%
  group_by(date) %>%
  summarise(returns = sum(weighted_returns))

head(port_return_tydiverse, 3)

## # A tibble: 3 x 2
##   date      returns
##   <date>    <dbl>
## 1 2019-02-28 -0.0259
## 2 2019-03-31 -0.00930
## 3 2019-04-30  0.0653

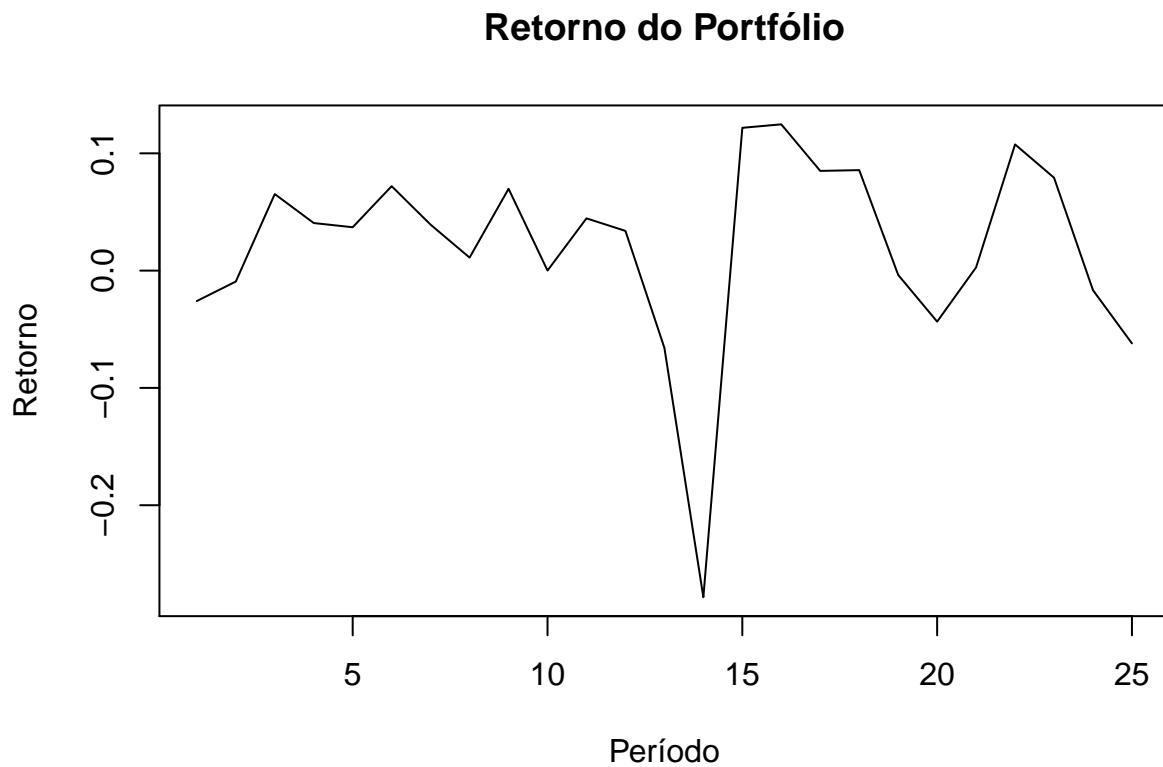
```

Vizualizando o retorno do Portfólio

```

ts.plot(portfolio_return, main = "Retorno do Portfólio",
        xlab = "Período", ylab = "Retorno")

```



Comparando a distribuição do Portfólio com os Ativos

```
asset_returns_long %>%  
  ggplot(aes(x = returns,  
             fill = asset)) +  
  geom_histogram(alpha = 0.15,  
                 binwidth = .01) +  
  geom_histogram(data = port_return_tydiverse,  
                 fill = "cornflowerblue",  
                 binwidth = .01) +  
  ggtitle("Portfolio and Asset Monthly Returns") +  
  theme_update(plot.title = element_text(hjust = 0.5))
```


Portfolio and Asset Monthly Returns

