Código Bastter

```
library(tidyverse)
library(GetBCBData)
library(BatchGetSymbols)
library(purrr)
library(furrr)
graphics.off()
my_d <- dirname(rstudioapi::getActiveDocumentContext()$path)
setwd(my_d)
plan(multisession(workers = 10))
first_date <- '2000-01-01'
n_sim <- 10000
vec_assets <- c(1, seq(5, 30, by = 5))
vec_years <- seq(1, 15, by = 1)
initial_capital <- 1
options(dplyr.summarise.inform = FALSE)
df_ibov <- BatchGetSymbols(tickers = '^BVSP', first.date = first_date)[[2]] %>%
  mutate(#company = ticker,
    data = ref.date,
    #fechamento = price.adjusted,
    value_ibov = price.adjusted) %>%
  select(data, value_ibov)
my_f <- 'data/dfin_cotacoes.rds'
df <- read_rds(my_f)
df_inflation <- gbcbd_get_series(id = 433, first.date = first_date) %>%
  mutate(inflation_idx = cumprod(1+ value/100),
         ref_month = ref.date) %>%
  select(ref_month, inflation_idx)
df <- df  %>%
  filter(data >= as.Date(first_date),
         str_detect(company, fixed('ON (')) ) %>%
  arrange(company, data) %>%
  left_join(df_ibov) %>%
  mutate(ref_month = as.Date(format(data, '%Y-%m-01')),
```

```
        ref_year = as.numeric(format(data, '%Y'))) %>%
  left_join(df_inflation) %>%
  filter(!is.na(inflation_idx)) %>%
  na.omit()
write_rds(df, 'data/df_stocks_filtered.rds')
sim_fct <- function(i_sim, n_assets, n_years, initial_capital = 1000) {
  options(dplyr.summarise.inform = FALSE)
  # select random year
  first_year <- sample(seq(min(df$ref_year),(max(df$ref_year) - n_years -1)),
                       1)
  last_year <- first_year + n_years

  # message('i_sim = ', i_sim,
  #         '\tfirst_year = ', first_year,
  #         '\tlast_year = ', last_year,
  #         '\tn_assets = ', n_assets)

  #browser()
  temp_df <- df %>%
    filter(ref_year >= first_year,
           ref_year <= last_year)

  rnd_assets <- sample(unique(temp_df$company), n_assets)

  temp_df <- df %>%
    filter(company %in% rnd_assets,
           ref_year >= first_year,
           ref_year <= last_year)

  to_invest <- initial_capital/n_assets

  my_l <- split(temp_df, f = temp_df$company)

  port <- bind_rows(
    map(my_l, .f = fct_invest, to_invest = to_invest)
  ) %>%
    group_by(ref_year) %>%
    summarise(port_value = sum(port_value),
              port_value_ibov = sum(port_value_ibov),
```

```
            port_value_inflation = sum(port_value_inflation)) %>%
    ungroup() %>%
    mutate(n_assets,
           ref_year2 = 1:length(ref_year),
           i_sim,
           n_years,
           beat_ibov = port_value > port_value_ibov,
           beat_inflation = port_value > port_value_inflation)

  return(port)

}
fct_invest <- function(df_in, to_invest) {

  port_temp <- df_in %>%
    mutate(ret = c(0 , na.omit(fechamento/lag(fechamento) - 1)),
           ret_ibov = c(0 , na.omit(value_ibov/lag(value_ibov) - 1)),
           ret_inflation  =  c(0 , na.omit(inflation_idx/lag(inflation_idx) - 1))  ) %>%
    #na.omit() %>%
    mutate(cum_ret = cumprod(1 + ret),
           cum_ret_ibov = cumprod(1+ ret_ibov),
           cum_ret_inflation = cumprod(1 + ret_inflation),
           port_value = to_invest*cum_ret,
           port_value_ibov = to_invest*cum_ret_ibov,
           port_value_inflation = to_invest*cum_ret_inflation) %>%
    group_by(company, ref_year) %>%
    summarise(port_value = last(port_value),
              port_value_ibov = last(port_value_ibov),
              port_value_inflation = last(port_value_inflation) ) %>%
    ungroup()

  return(port_temp)

}

for (i_assets in vec_assets) {
  for (i_year in vec_years) {

    message(str_glue('i_assets = {i_assets} | i_year = {i_year}'))
```

```
    f_out <- file.path('simdata',
                       str_glue('simdata_nsim_{n_sim}_nassets_{i_assets}_nyears_{i_year}.rds'))

    if (file.exists(f_out)) {
      message('Found File!')
      next()
    }

    l_args <- list(i_sim = 1:n_sim,
                   n_assets = i_assets,
                   n_years = i_year,
                   initial_capital = initial_capital)

    l_sim <- future_pmap(.l = l_args, .f = sim_fct, .progress = TRUE)
    #l_sim <- pmap(.l = l_args, .f = sim_fct)

    df_res <- bind_rows(l_sim)

    write_rds(df_res, f_out)

  }

}

write_rds(df_res, 'data/sim_data.rds')
```