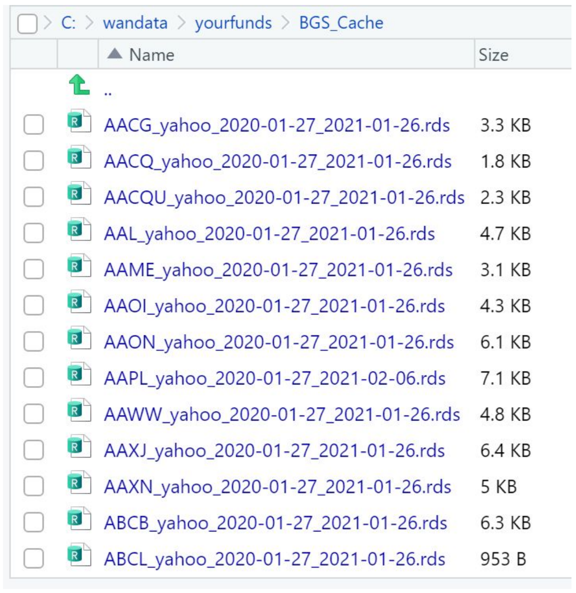
**yourfunds**

I organized the codes into scripts, and I'm ready to build the Shiny App. I started to build the Home page and I realized that I didn't know how to get hourly data Robinhood. I found a way, but seems like the function for that is broken, because the hours are messed up and incomplete. I will need to get some support from the creator of the package.
I planned on having the Home page built this week, but I didn't spend enough time on the project. I'm planning on compensating the next week and catch up with the schedule and deliverables.
My next step will be to actually start building the app.

Below are parts of the code that I created or fixed this week.

**Code**

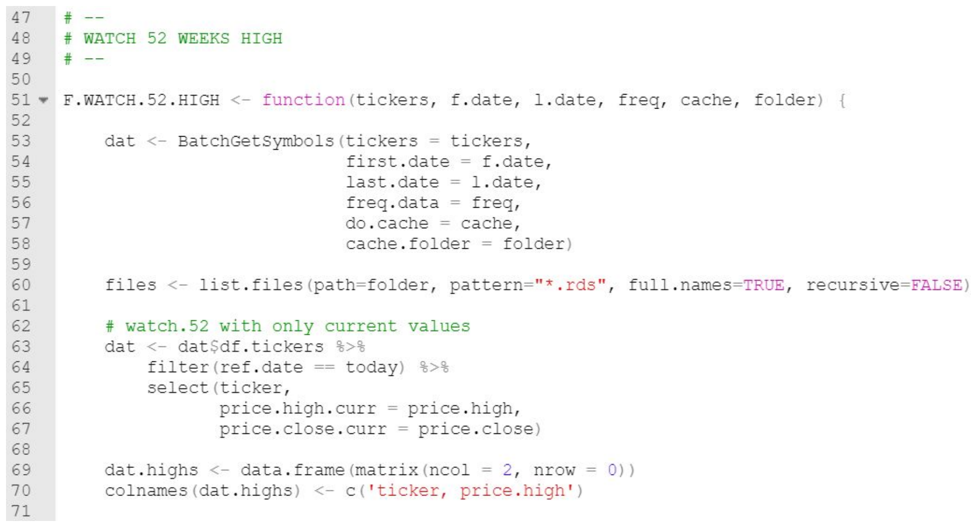
Folder with 52 weeks of daily data for ALL stocks in RobinHood:



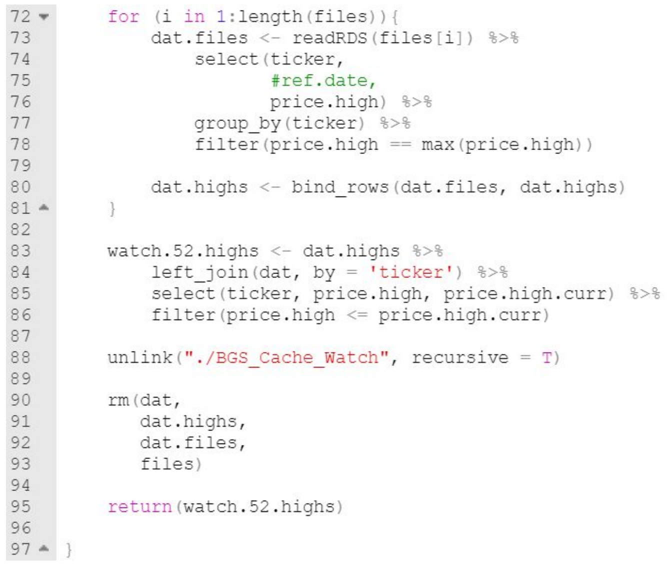
C: > wandata > yourfunds > BGS_Cache

| Name | Size |
|---|---|
| .. | |
| AACG_yahoo_2020-01-27_2021-01-26.rds | 3.3 KB |
| AACQ_yahoo_2020-01-27_2021-01-26.rds | 1.8 KB |
| AACQU_yahoo_2020-01-27_2021-01-26.rds | 2.3 KB |
| AAL_yahoo_2020-01-27_2021-01-26.rds | 4.7 KB |
| AAME_yahoo_2020-01-27_2021-01-26.rds | 3.1 KB |
| AAOI_yahoo_2020-01-27_2021-01-26.rds | 4.3 KB |
| AAON_yahoo_2020-01-27_2021-01-26.rds | 6.1 KB |
| AAPL_yahoo_2020-01-27_2021-02-06.rds | 7.1 KB |
| AAWW_yahoo_2020-01-27_2021-01-26.rds | 4.8 KB |
| AAXJ_yahoo_2020-01-27_2021-01-26.rds | 6.4 KB |
| AAXN_yahoo_2020-01-27_2021-01-26.rds | 5 KB |
| ABCB_yahoo_2020-01-27_2021-01-26.rds | 6.3 KB |
| ABCL_yahoo_2020-01-27_2021-01-26.rds | 953 B |

Function to run every time the user wants to see if any stock in one of his watchlists hit 52 weeks high.
I get 52 weeks of data for all stocks in my watchlists, then I loop through each one of them comparing if the highest price of the day is the highest price within the 52 weeks.
The function takes only a few seconds to run and return a vector with all the stocks that the current/last price is the highest in 52 weeks.

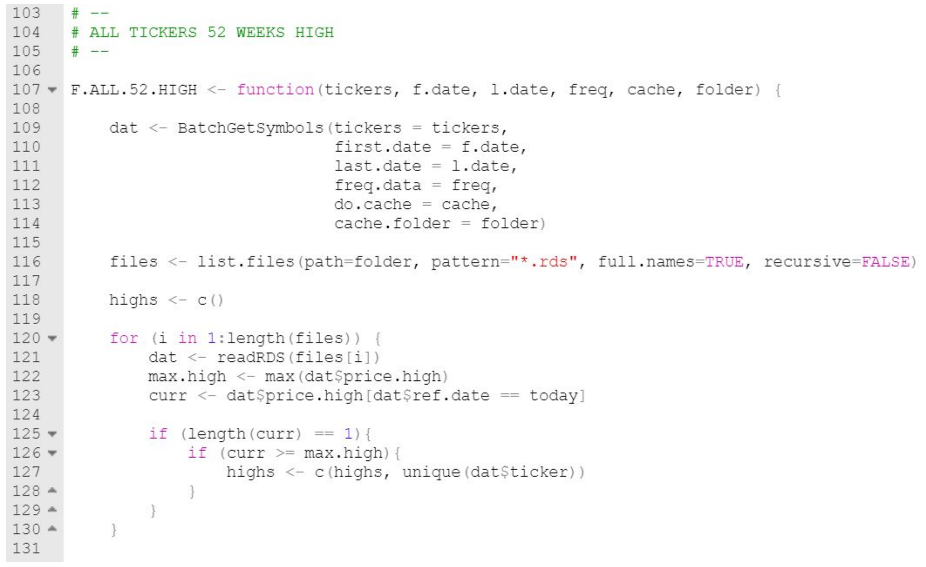


```r
# --
# WATCH 52 WEEKS HIGH
# --

F.WATCH.52.HIGH <- function(tickers, f.date, l.date, freq, cache, folder) {

    dat <- BatchGetSymbols(tickers = tickers,
                           first.date = f.date,
                           last.date = l.date,
                           freq.data = freq,
                           do.cache = cache,
                           cache.folder = folder)

    files <- list.files(path=folder, pattern="*.rds", full.names=TRUE, recursive=FALSE)

    # watch.52 with only current values
    dat <- dat$df.tickers %>%
        filter(ref.date == today) %>%
        select(ticker,
               price.high.curr = price.high,
               price.close.curr = price.close)

    dat.highs <- data.frame(matrix(ncol = 2, nrow = 0))
    colnames(dat.highs) <- c('ticker, price.high')

```

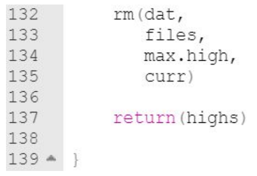


```r
    for (i in 1:length(files)){
        dat.files <- readRDS(files[i]) %>%
            select(ticker,
                   #ref.date,
                   price.high) %>%
            group_by(ticker) %>%
            filter(price.high == max(price.high))

        dat.highs <- bind_rows(dat.files, dat.highs)
    }

    watch.52.highs <- dat.highs %>%
        left_join(dat, by = 'ticker') %>%
        select(ticker, price.high, price.high.curr) %>%
        filter(price.high <= price.high.curr)

    unlink("./BGS_Cache_Watch", recursive = T)

    rm(dat,
       dat.highs,
       dat.files,
       files)

    return(watch.52.highs)

}
```

Function to run every night and get all stocks that exist in Robinhood that hit 52 weeks high that day.
Every time this function runs, it will go over the files for all the stocks and append new days of data. It will return a vector with all stocks that reached 52 weeks high during that day.

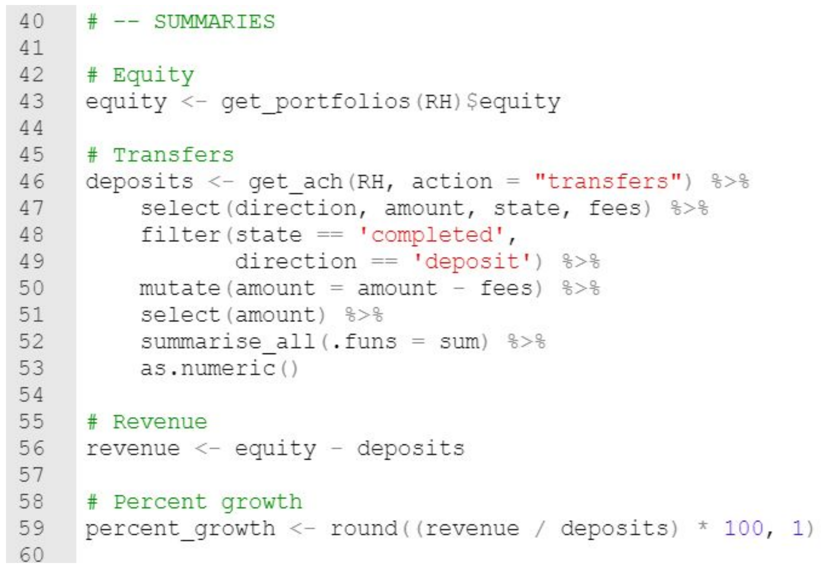


```r
# --
# ALL TICKERS 52 WEEKS HIGH
# --

F.ALL.52.HIGH <- function(tickers, f.date, l.date, freq, cache, folder) {

    dat <- BatchGetSymbols(tickers = tickers,
                           first.date = f.date,
                           last.date = l.date,
                           freq.data = freq,
                           do.cache = cache,
                           cache.folder = folder)

    files <- list.files(path=folder, pattern="*.rds", full.names=TRUE, recursive=FALSE)

    highs <- c()

    for (i in 1:length(files)) {
        dat <- readRDS(files[i])
        max.high <- max(dat$price.high)
        curr <- dat$price.high[dat$ref.date == today]

        if (length(curr) == 1){
            if (curr >= max.high){
                highs <- c(highs, unique(dat$ticker))
            }
        }
    }

```



```r
    rm(dat,
       files,
       max.high,
       curr)

    return(highs)

}
```

Summaries (current equity, all money invested out of pocket, return, percent growth)



```r
# -- SUMMARIES

# Equity
equity <- get_portfolios(RH)$equity

# Transfers
deposits <- get_ach(RH, action = "transfers") %>%
    select(direction, amount, state, fees) %>%
    filter(state == 'completed',
           direction == 'deposit') %>%
   mutate(amount = amount - fees) %>%
    select(amount) %>%
    summarise_all(.funs = sum) %>%
    as.numeric()

# Revenue
revenue <- equity - deposits

# Percent growth
percent_growth <- round((revenue / deposits) * 100, 1)

```