Project 4

Strategy 1

```r
require(quantmod)
```

```
## Loading required package: quantmod
```

```
## Warning: package 'quantmod' was built under R version 3.6.2
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```r
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:xts':
##
##     first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
require(tidyquant)
```

```
## Loading required package: tidyquant
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
## Loading required package: PerformanceAnalytics
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

```
## ── Need to Learn tidyquant? ──────────────────────────────────────────────────────
## Business Science offers a 1-hour course - Learning Lab #9: Performance Analysis & Portfolio Optimization with tidyquant!
## </> Learn more at: https://university.business-science.io/p/learning-labs-pro </>
```

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
stock_prices <- c("SPY") %>% tq_get(get = "stock.prices",
                                                      from = "1990-01-01", to = "2020-01-01")
stock_prices
```

```
## # A tibble: 6,780 x 8
##    symbol date        open  high   low close  volume adjusted
##    <chr>  <date>     <dbl> <dbl> <dbl> <dbl>   <dbl>    <dbl>
##  1 SPY    1993-01-29  44.0  44.0  43.8  43.9 1003200     26.3
##  2 SPY    1993-02-01  44.0  44.2  44.0  44.2  480500     26.5
##  3 SPY    1993-02-02  44.2  44.4  44.1  44.3  201300     26.5
##  4 SPY    1993-02-03  44.4  44.8  44.4  44.8  529400     26.8
##  5 SPY    1993-02-04  45.0  45.1  44.5  45    531500     26.9
##  6 SPY    1993-02-05  45.0  45.1  44.7  45.0  492100     26.9
##  7 SPY    1993-02-08  45.0  45.1  44.9  45.0  596100     26.9
##  8 SPY    1993-02-09  44.8  44.8  44.6  44.7  122100     26.7
##  9 SPY    1993-02-10  44.7  44.8  44.5  44.7  379600     26.8
## 10 SPY    1993-02-11  44.8  45.1  44.8  44.9   19500     26.9
## # … with 6,770 more rows
```

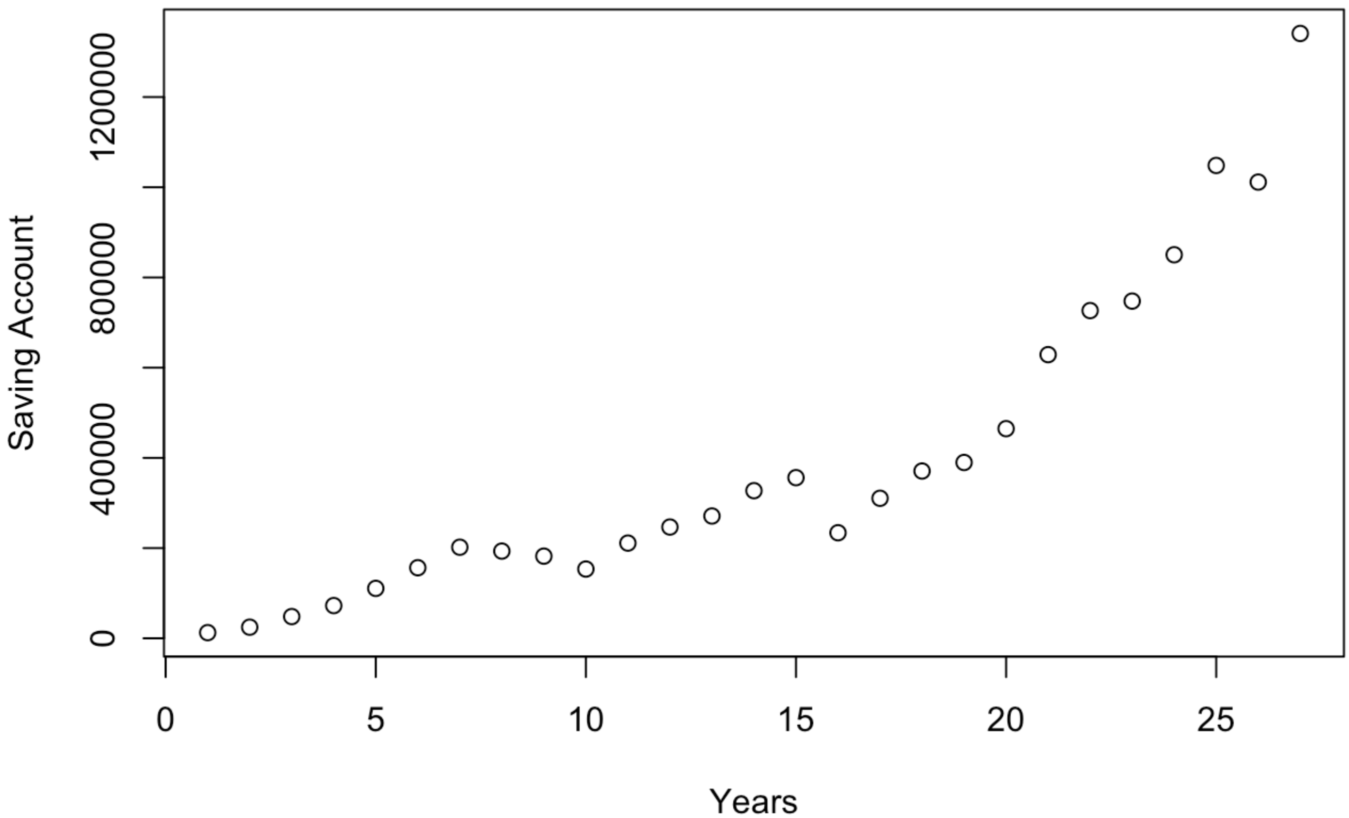
```
stock_returns_monthly <- stock_prices %>% group_by(symbol) %>% tq_transmute(select = adjusted, mutate_fun = periodReturn, period = "monthly", col_rename = "Ra")
stock_returns_monthly
```

```
## # A tibble: 324 x 3
## # Groups:   symbol [1]
##    symbol date             Ra
##    <chr>  <date>        <dbl>
##  1 SPY    1993-01-29  0
##  2 SPY    1993-02-26  0.0107
##  3 SPY    1993-03-31  0.0224
##  4 SPY    1993-04-30 -0.0256
##  5 SPY    1993-05-28  0.0270
##  6 SPY    1993-06-30  0.00361
##  7 SPY    1993-07-30 -0.00485
##  8 SPY    1993-08-31  0.0383
##  9 SPY    1993-09-30 -0.00728
## 10 SPY    1993-10-29  0.0197
## # … with 314 more rows
```

```
saving<-0
a = c()
for (i in 1:length(stock_returns_monthly$Ra)){
  saving <- saving + 1000
  saving <- saving + saving * stock_returns_monthly$Ra[[i]]
  if (i%%12 == 0){
    a <- c(a,saving)
  }
}
df <- data.frame(Saving=unlist(a))
df
```

```
##          Saving
## 1    12581.82
## 2    24679.09
## 3    48183.59
## 4    72559.72
## 5   110693.90
## 6   156500.21
## 7   201963.72
## 8   193351.05
## 9   182178.61
## 10  153557.38
## 11  211092.96
## 12  246686.47
## 13  271138.21
## 14  327291.79
## 15  356189.44
## 16  234022.45
## 17  310440.28
## 18  370848.51
## 19  389811.37
## 20  464789.82
## 21  628894.27
## 22  726551.34
## 23  747603.74
## 24  850344.25
## 25 1048305.86
## 26 1011466.75
## 27 1340940.13
```

```
plot(a, xlab = "Years", ylab = "Saving Account")
```




```
print("The total money accumulated at the end of 30 years ")
```

```
## [1] "The total money accumulated at the end of 30 years "
```

```
tail(a, n=1)
```

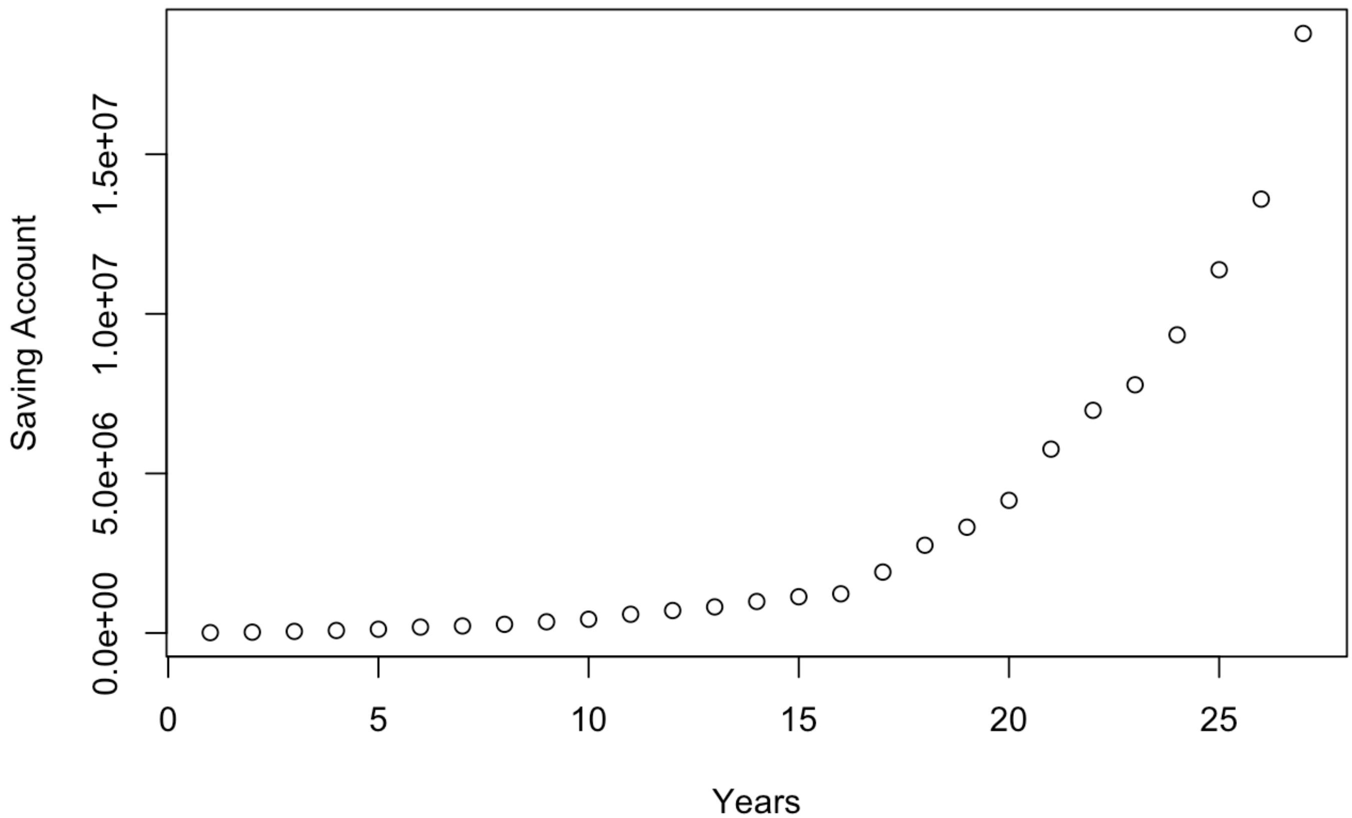
```
## [1] 1340940
```

Strategy 2

```r
require(quantmod)
require(dplyr)
require(tidyquant)
require(ggplot2)
stock_prices <- c("SPY") %>% tq_get(get = "stock.prices",
                                    from = "1993-01-01", to = "2020-01-01")
sma <- SMA(stock_prices$close, n=50)
stock_returns_monthly <- stock_prices %>% group_by(symbol) %>% tq_transmute(select =
adjusted, mutate_fun = periodReturn, period = "monthly", col_rename = "Ra")
saving<-0
b<-c(1,1,1)
j<-0
for (j in seq(from=63, to=length(stock_prices$close), by=21)){
  if(stock_prices$close[[j]] > sma[[j]])
    b<- c(b,1)
  else
    b<- c(b,0)
}
b<-c(b,1)
a = c()
for (i in 1:length(stock_returns_monthly$Ra)){
  saving <- saving + 1000
  if (i < 3 ){
    saving <- saving + saving * stock_returns_monthly$Ra[[i]]
  }
  else if (b[i] == 1){
    saving <- saving + saving * stock_returns_monthly$Ra[[i]]
  }
  if (i%%12 == 0){
    a <- c(a,saving)
  }
}
df2 <- data.frame(Saving=unlist(a))
df2
```

```
##         Saving
## 1     12694.79
## 2     27163.27
## 3     51612.98
## 4     80117.66
## 5    118990.83
## 6    186185.31
## 7    221793.32
## 8    272976.66
## 9    351416.29
## 10   430892.73
## 11   587083.32
## 12   705353.63
## 13   818308.72
## 14   988216.98
## 15  1136343.79
## 16  1228570.15
## 17  1909976.54
## 18  2750546.37
## 19  3315658.10
## 20  4157076.24
## 21  5761229.98
## 22  6977383.16
## 23  7777067.30
## 24  9341001.43
## 25 11381894.82
## 26 13594541.32
## 27 18784534.54
```

```
plot(a, xlab = "Years", ylab = "Saving Account")
```




```
print("The total money accumulated at the end of 30 years ")
```

```
## [1] "The total money accumulated at the end of 30 years "
```

```
tail(a, n=1)
```

```
## [1] 18784535
```

The 2nd strategy is better as the amount of money accumlated over 30 years is higher in the 2nd case. This is because in the 2nd strategy we only invest in stock when we see a upward trend thereby making it more profitable. We check for the upward trend by comparing the close day price to the 50 day SMA.