Week 5 Assignment - Group 2

Solutions:

In this file we implement the solutions of the questions that involve programming (Questions A, B). For more detailed explanations please see the attached word file.

First we import the relevant libraries:

```
library(readxl)
library(forecast)
```

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

```
library(dplyr)
```

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

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

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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v stringr 1.4.0
## v tidyr   1.0.0     v forcats 0.4.0
## v readr   1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

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

```
library(padr)
library(tm)
```

```
## Loading required package: NLP

##
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(rlist)
library(ggplot2)
library(SciViews)
```

Then we read the hourly prices for energy in excel file:

```
data_orig <- read_excel("Week 5 Assignment data.xlsx")
```

```
## New names:
## * `` -> ...2
## * `` -> ...3
```

```
data_orig
```

```
## # A tibble: 1,349 x 3
##    `Week 5 Assignment Data`  ...2 ...3
##    <chr>                    <dbl> <chr>
##  1 <NA>                        NA <NA>
##  2 <NA>                        NA price
##  3 <NA>                        NA DK_1
##  4 Time stamp                  NA Energinet.dk
##  5 <NA>                        NA <NA>
##  6 43491                        1 51.52
##  7 43491.041666666657           2 50.98
##  8 43491.083333333343           3 50.85
##  9 43491.125                    4 51.52
## 10 43491.166666666657           5 52.01
## # ... with 1,339 more rows
```

We make our dataset easier to use by deleting the first 5 rows and changing the column names:

```r
data_orig <- data_orig[-(1:5), ]
colnames(data_orig)<- c("timestamp","hour","price")
data_orig
```

```
## # A tibble: 1,344 x 3
##    timestamp           hour price
##    <chr>              <dbl> <chr>
##  1 43491                  1 51.52
##  2 43491.041666666657     2 50.98
##  3 43491.083333333343     3 50.85
##  4 43491.125              4 51.52
##  5 43491.166666666657     5 52.01
##  6 43491.208333333343     6 52.44
##  7 43491.25               7 53.35
##  8 43491.291666666657     8 53.99
##  9 43491.333333333343     9 56.28
## 10 43491.375             10 58.62
## # ... with 1,334 more rows
```

Question A

Then we need to find the average price of energy for every hour of the day, given the 56 days we have available.

```r
data_new <- mutate(data_orig, hour_day = rep(1:24, times=56))
data_new$price <-as.double(data_new$price)
by_hour <- group_by(data_new,hour_day)
mean_price <- summarise(by_hour, price_hour = mean(price,na.rm = TRUE))
data_new
```

```
## # A tibble: 1,344 x 4
##    timestamp           hour price hour_day
##    <chr>              <dbl> <dbl>    <int>
##  1 43491                  1  51.5        1
##  2 43491.041666666657     2  51.0        2
##  3 43491.083333333343     3  50.8        3
##  4 43491.125              4  51.5        4
##  5 43491.166666666657     5  52.0        5
##  6 43491.208333333343     6  52.4        6
##  7 43491.25               7  53.4        7
##  8 43491.291666666657     8  54.0        8
##  9 43491.333333333343     9  56.3        9
## 10 43491.375             10  58.6       10
## # ... with 1,334 more rows
```

The average price per hour is the following:

```r
mean_price
```

```
## # A tibble: 24 x 2
##    hour_day price_hour
```

```
##        <int>      <dbl>
##   1        1       32.1
##   2        2       31.1
##   3        3       31.3
##   4        4       32.3
##   5        5       34.8
##   6        6       39.8
##   7        7       45.1
##   8        8       46.1
##   9        9       44.6
##  10       10       42.9
## # ... with 14 more rows
```

Question B

We first define the decision variables:

```
decision_variables <- mean_price
decision_variables$p <- 1
decision_variables$g <- 1
decision_variables$s <- 7*10^6

price <- decision_variables$price_hour
pump <- decision_variables$p
gen <- decision_variables$g
res <- decision_variables$s
```

We now implement the optimisation function:

```
profit.optim <- function(x, price){
  #assigning parameters
  bm <- x[1]
  ba <- x[2]
  Cm <- x[3]
  Ca <- x[4]
  Dm <- x[5]
  Da <- x[6]
  p <- x[7:30]
  g <- x[31:54]
  s <- x[55:78]
  price <- price
  #Calculations
  s[1]<- 5000000+14000*1*60
  for (t in 2:12){
    if (s[t-1]+bm*price[t-1]<=Cm){
      p[t]<-1
    }else{
      p[t]<-0
    }
    if(s[t-1]+bm*price[t-1]>=Dm){
      g[t]<-1
    }else{
      g[t]<-0
```

```r
    }
    s[t]<- 14000*p[t]*60-16000*g[t]*60+s[t-1]
  }
  for (t in 13:24){
    if (s[t-1]+ba*price[t-1]<=Ca){
      p[t]<-1
    }else{
      p[t]<-0
    }
    if(s[t-1]+ba*price[t-1]>=Da){
      g[t]<-1
    }else{
      g[t]<-0
    }
    s[t]<- 14000*p[t]*60-16000*g[t]*60+s[t-1]
  }
  profit <- sum(g*270*price-p*300*price)
  return(profit)
}

dec_var <- c(500000,500000,6*10^6,8*10^6,6*10^6,8*10^6, pump, gen, res)
lower_bound <- c(-Inf, -Inf, 0, 0, 0, 0, rep(0,24), rep(0,24), rep(0,24))
upper_bound <- c(+Inf, +Inf, 10*10^6, 10*10^6, 10*10^6, 10*10^6, rep(1,24),
                 rep(1,24), rep(10*10^6,23), 5000000)

optim(x <- dec_var,fn = profit.optim, price = price, method="L-BFGS-B",
      lower = lower_bound, upper = upper_bound, control = list(fnscale = -1))[1:2]
```

```
## $par
##  [1] 5e+05 5e+05 6e+06 8e+06 6e+06 8e+06 0e+00 1e+00 1e+00 1e+00 1e+00 1e+00
## [13] 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00
## [25] 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00
## [37] 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00
## [49] 1e+00 1e+00 1e+00 1e+00 1e+00 1e+00 7e+06 7e+06 7e+06 7e+06 7e+06 7e+06
## [61] 7e+06 7e+06 7e+06 7e+06 7e+06 7e+06 7e+06 7e+06 7e+06 7e+06 7e+06 7e+06
## [73] 7e+06 7e+06 7e+06 7e+06 7e+06 5e+06
##
## $value
## [1] 197901.1
```

Hence, the maximum profit is $197,901.1$.