AV Profit

Bijesh Mishra, Ph.D.

Table of contents

Analysis in this file start by loading data saved after simulating tomato, strawberry, and squash AV profits. See simulation file for more details. The result tables I have here are quite big. Results are summarized in separate excel file (Results.xlsx).

1 Setting Up

1.1 Housekeeping

```
# #| echo: TRUE
rm(list = ls()) # Clean the environment.
options(
  warn=0, # Warnings. options(warn=-1) / options(warn=0)
  scipen=999 # No scientific notations.
  )
```

1.2 Working directory

Codes and output are suppressed. Errors and warnings are visible. No warning and no error means code is working as it should.

1.3 Load libraries

```
library(tidyverse, warn.conflicts = FALSE, quietly = TRUE)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

```r
library(psych, warn.conflicts = FALSE, quietly = TRUE)
library(likert,  warn.conflicts = FALSE, quietly = TRUE) # Likert Items
library(mice,  warn.conflicts = FALSE, quietly = TRUE)
library(openxlsx2, warn.conflicts = FALSE, quietly = TRUE)
library(ggpubr, warn.conflicts = FALSE, quietly = TRUE) # Scatter plot
library(gmodels,  warn.conflicts = FALSE, quietly = TRUE) # Crosstab
library(reshape2, warn.conflicts = FALSE, quietly = TRUE) # Reshape data
library(pacman,  warn.conflicts = FALSE, quietly = TRUE) # Package Management
library(progress, warn.conflicts = FALSE, quietly = TRUE) #progress bar
library(arrow, warn.conflicts = FALSE, quietly = TRUE) #progress bar
```

```
Some features are not enabled in this build of Arrow. Run `arrow_info()` for more information
The repository you retrieved Arrow from did not include all of Arrow's features.
You can install a fully-featured version by running:
`install.packages('arrow', repos = 'https://apache.r-universe.dev')`.
```

```r
pacman::p_loaded()
```

```
 [1] "arrow"     "progress"  "pacman"    "reshape2"  "gmodels"   "ggpubr"
 [7] "openxlsx2" "mice"      "likert"    "xtable"    "psych"     "lubridate"
[13] "forcats"   "stringr"   "dplyr"     "purrr"     "readr"     "tidyr"
[19] "tibble"    "ggplot2"   "tidyverse"
```

1.4 Progress bar

Tracking data processing progress.

```r
####### Progress Bar #####
pb = progress_bar$new(
  format = "Processing data at :rate. Processed :bytes in :elapsed.",
  clear = TRUE,
  total = NA,
  width = 80)
f = function() {
  for (i in 1:100) {
    pb$tick(sample(1:100 * 1000, 1))
    Sys.sleep(2/100)
  }
  pb$tick(1e7)
  #invisible()
}
```

1.5 Theme for plots

Setting theme for plots:

```
####### Plotting Data: #####
# Map Theme:
plottheme <- ggplot() +
  theme_void() +
  # Mapping theme:
  theme(axis.title = element_blank(),
        axis.ticks = element_blank(),
        axis.text = element_blank(),
        panel.border = element_blank(),
        plot.margin = margin(t = 0,
                             r = 0,
                             b = 0,
                             l = 0,
                             unit = "cm"),
        plot.title = element_text(hjust = 0.5),
        plot.background = element_rect(fill = "white",
                                       color = "black",
                                       linewidth = 0),
        panel.background = element_rect(fill = "white",
                                        color = "black",
                                        linewidth = 0),
        panel.grid.major.x = element_line(color = "lightgrey",
                                          linetype = 2,
                                          linewidth = 0),
        panel.grid.minor.x = element_line(color = "lightgrey",
                                          linetype = 2,
                                          linewidth = 0),
        panel.grid.major.y = element_line(color = "grey",
                                          linetype = 2,
                                          linewidth = 0),
        panel.grid.minor.y = element_line(color = "grey",
                                          linetype = 2,
                                          linewidth = 0),
        axis.line.x.top = element_line(color = "white",
                                       linetype = 2,
                                       linewidth = 0),
        axis.line.y.right = element_line(color = "white",
                                         linetype = 2,
                                         linewidth = 0),
```

```
        axis.line.x.bottom = element_line(color = "black",
                                          linetype = 1,
                                          linewidth = 0),
        axis.line.y.left = element_line(color = "black",
                                        linetype = 1,
                                        linewidth = 0),
        # Text formatting:
        text = element_text(family = "serif", # font
                            size = 12, # font size
                            colour = "black"# font color
        ),
        legend.position = c(0.95, -0.05),
        legend.key = element_rect(color = "black",
                                  fill = NA,
                                  linewidth = 0.05,
                                  linetype = 1),
        legend.justification = "right",
        legend.direction = "horizontal")
```

```
Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2
3.5.0.
i Please use the `legend.position.inside` argument of `theme()` instead.
```

2 Import data

Import necessary data.

2.1 Tomato AV

sprop = proportion of solar in agrivoltaic system (0 to 1 in 0.5 increment.0

al_regs = four regions of Alabama. Northern, Central, Black Belt, Southern.

array = Solar array; Sun tracking (Tracking) and non-tracking (Fixed).

dc_kw = DC system size (kW) See PVWatts® Calculator.

panels = number of solar panels.

energy = total energy generated from solar system. See: PVWatts® Calculator.

elecprc = electricity price (1 cents to 6 cents).

height = clearance height of solar panels. 4.6 ft., 6.4 ft., and 8.2 ft.

capex = AV system capex per kW. See: Capex Cost for AV table 1 and table 3.

ttlcost = total solar system cost in AV. See: Capex Cost for AV table 1 and table 3.

anncost = annualized total cost.

moncost = monthly total cost.

eprofit = profit from electricity.

eannprof = annualized total profit from electricity.

emonprof = monthly total profit from electricity.

yldvar = crop yield variation (10% to 200%)

yield = crop yield variation based on yldvar.

price = crop yield price per bucket.

profit = profit from crops.

tav_profit = total profit from solar and tomato.

```
tav_profit <- as.data.frame(read_feather(file = "tav_profit.feather"))
dim(tav_profit)
```

```
[1] 776160     21
```

```
#str(tav_profit)
head(tav_profit); head(tav_profit)
```

```
  sprop    al_regs    array dc_kw panels energy elcprc elcrev height    capex
1     0 Black Belt    Fixed     0      0      0   0.01      0    4.6 1.593333
2     0 Black Belt    Fixed     0      0      0   0.01      0    6.4 1.850000
3     0 Black Belt    Fixed     0      0      0   0.01      0    8.2 2.330000
4     0 Black Belt Tracking     0      0      0   0.01      0    4.6 1.733333
5     0 Black Belt Tracking     0      0      0   0.01      0    6.4 1.921667
6     0 Black Belt Tracking     0      0      0   0.01      0    8.2 2.110000
  ttlcost anncost moncost eprofit eannprof emonprof yldvar yield price   profit
1       0       0       0       0        0        0      2  2720    17 21679.38
2       0       0       0       0        0        0      2  2720    17 21679.38
3       0       0       0       0        0        0      2  2720    17 21679.38
4       0       0       0       0        0        0      2  2720    17 21679.38
5       0       0       0       0        0        0      2  2720    17 21679.38
```

```
6       0       0       0       0        0        0      2  2720    17 21679.38
  tav_profit
1   21679.38
2   21679.38
3   21679.38
4   21679.38
5   21679.38
6   21679.38
```

```
  sprop    al_regs    array dc_kw panels energy elcprc elcrev height    capex
1     0 Black Belt    Fixed     0      0      0   0.01      0    4.6 1.593333
2     0 Black Belt    Fixed     0      0      0   0.01      0    6.4 1.850000
3     0 Black Belt    Fixed     0      0      0   0.01      0    8.2 2.330000
4     0 Black Belt Tracking     0      0      0   0.01      0    4.6 1.733333
5     0 Black Belt Tracking     0      0      0   0.01      0    6.4 1.921667
6     0 Black Belt Tracking     0      0      0   0.01      0    8.2 2.110000
  ttlcost anncost moncost eprofit eannprof emonprof yldvar yield price   profit
1       0       0       0       0        0        0      2  2720    17 21679.38
2       0       0       0       0        0        0      2  2720    17 21679.38
3       0       0       0       0        0        0      2  2720    17 21679.38
4       0       0       0       0        0        0      2  2720    17 21679.38
5       0       0       0       0        0        0      2  2720    17 21679.38
6       0       0       0       0        0        0      2  2720    17 21679.38
  tav_profit
1   21679.38
2   21679.38
3   21679.38
4   21679.38
5   21679.38
6   21679.38
```

2.2 Strawberry AV

See tomato for variable descriptions.

sbav_profit = total profit from solar and strawberry.

```
sbav_profit <- as.data.frame(read_feather(file = "sbav_profit.feather"))
dim(sbav_profit)
```

```
[1] 776160     21
```

```
#str(sbav_profit)
head(sbav_profit); tail(sbav_profit)
```

```
  sprop    al_regs    array dc_kw panels energy elcprc elcrev height    capex
1     0 Black Belt    Fixed     0      0      0   0.01      0    4.6 1.593333
2     0 Black Belt    Fixed     0      0      0   0.01      0    6.4 1.850000
3     0 Black Belt    Fixed     0      0      0   0.01      0    8.2 2.330000
4     0 Black Belt Tracking     0      0      0   0.01      0    4.6 1.733333
5     0 Black Belt Tracking     0      0      0   0.01      0    6.4 1.921667
6     0 Black Belt Tracking     0      0      0   0.01      0    8.2 2.110000
  ttlcost anncost moncost eprofit eannprof emonprof yldvar yield price
1       0       0       0       0        0        0      2  6150     3
2       0       0       0       0        0        0      2  6150     3
3       0       0       0       0        0        0      2  6150     3
4       0       0       0       0        0        0      2  6150     3
5       0       0       0       0        0        0      2  6150     3
6       0       0       0       0        0        0      2  6150     3
     profit sbav_profit
1 -1740.495   -1740.495
2 -1740.495   -1740.495
3 -1740.495   -1740.495
4 -1740.495   -1740.495
5 -1740.495   -1740.495
6 -1740.495   -1740.495
```

```
       sprop  al_regs    array  dc_kw panels   energy elcprc   elcrev height
776155     1 Southern    Fixed 423.74    885 598720.5   0.06 35923.23    4.6
776156     1 Southern    Fixed 423.74    885 598720.5   0.06 35923.23    6.4
776157     1 Southern    Fixed 423.74    885 598720.5   0.06 35923.23    8.2
776158     1 Southern Tracking 423.74    885 695415.0   0.06 41724.90    4.6
776159     1 Southern Tracking 423.74    885 695415.0   0.06 41724.90    6.4
776160     1 Southern Tracking 423.74    885 695415.0   0.06 41724.90    8.2
          capex  ttlcost  anncost  moncost  eprofit eannprof emonprof yldvar
776155 1.593333 675.1591 47.90419 3.946913 35248.07 35875.33 2989.656    0.1
776156 1.850000 783.9190 55.62098 4.582712 35139.31 35867.61 2989.020    0.1
776157 2.330000 987.3142 70.05237 5.771740 34935.92 35853.18 2987.831    0.1
776158 1.733333 734.4827 52.11335 4.293713 40990.42 41672.79 3472.781    0.1
776159 1.921667 814.2870 57.77567 4.760241 40910.61 41667.12 3472.315    0.1
776160 2.110000 894.0914 63.43798 5.226769 40830.81 41661.46 3471.848    0.1
       yield price    profit sbav_profit
776155 307.5     9 -10855.74    25019.59
```

```
776156 307.5     9 -10855.74    25011.87
776157 307.5     9 -10855.74    24997.44
776158 307.5     9 -10855.74    30817.05
776159 307.5     9 -10855.74    30811.38
776160 307.5     9 -10855.74    30805.72
```

2.3 Squash AV

See tomato for variable descriptions.

sqav_profit = total profit from solar and squash.

```
sqav_profit <- as.data.frame(read_feather(file = "sqav_profit.feather"))
dim(sqav_profit)
```

```
[1] 776160     21
```

```
#str(sqav_profit)
head(sqav_profit); tail(sqav_profit)
```

```
  sprop    al_regs    array dc_kw panels energy elcprc elcrev height    capex
1     0 Black Belt    Fixed     0      0      0   0.01      0    4.6 1.593333
2     0 Black Belt    Fixed     0      0      0   0.01      0    6.4 1.850000
3     0 Black Belt    Fixed     0      0      0   0.01      0    8.2 2.330000
4     0 Black Belt Tracking     0      0      0   0.01      0    4.6 1.733333
5     0 Black Belt Tracking     0      0      0   0.01      0    6.4 1.921667
6     0 Black Belt Tracking     0      0      0   0.01      0    8.2 2.110000
  ttlcost anncost moncost eprofit eannprof emonprof yldvar yield price   profit
1       0       0       0       0        0        0      2  2180    11 10309.12
2       0       0       0       0        0        0      2  2180    11 10309.12
3       0       0       0       0        0        0      2  2180    11 10309.12
4       0       0       0       0        0        0      2  2180    11 10309.12
5       0       0       0       0        0        0      2  2180    11 10309.12
6       0       0       0       0        0        0      2  2180    11 10309.12
  sqav_profit
1    10309.12
2    10309.12
3    10309.12
4    10309.12
5    10309.12
6    10309.12
```

```
       sprop  al_regs    array  dc_kw panels   energy elcprc   elcrev height
776155     1 Southern    Fixed 423.74    885 598720.5   0.06 35923.23    4.6
776156     1 Southern    Fixed 423.74    885 598720.5   0.06 35923.23    6.4
776157     1 Southern    Fixed 423.74    885 598720.5   0.06 35923.23    8.2
776158     1 Southern Tracking 423.74    885 695415.0   0.06 41724.90    4.6
776159     1 Southern Tracking 423.74    885 695415.0   0.06 41724.90    6.4
776160     1 Southern Tracking 423.74    885 695415.0   0.06 41724.90    8.2
          capex  ttlcost  anncost  moncost  eprofit eannprof emonprof yldvar
776155 1.593333 675.1591 47.90419 3.946913 35248.07 35875.33 2989.656    0.1
776156 1.850000 783.9190 55.62098 4.582712 35139.31 35867.61 2989.020    0.1
776157 2.330000 987.3142 70.05237 5.771740 34935.92 35853.18 2987.831    0.1
776158 1.733333 734.4827 52.11335 4.293713 40990.42 41672.79 3472.781    0.1
776159 1.921667 814.2870 57.77567 4.760241 40910.61 41667.12 3472.315    0.1
776160 2.110000 894.0914 63.43798 5.226769 40830.81 41661.46 3471.848    0.1
       yield price    profit sqav_profit
776155   109    17 -2370.133    33505.19
776156   109    17 -2370.133    33497.48
776157   109    17 -2370.133    33483.04
776158   109    17 -2370.133    39302.65
776159   109    17 -2370.133    39296.99
776160   109    17 -2370.133    39291.33
```

3 Tabulating Results

3.1 Tomato AV

```
# Define the values for each variable
sprop <- c(0, 0.25, 0.50, 0.75, 1.00)
array <- c("Fixed", "Tracking")
height <- c(4.6, 6.4, 8.2)
yldvar <- c(0.10, 0.30, 0.50, 0.70, 1.00, 1.20, 1.50, 1.80, 2.00)
al_regs <- c("Northern", "Central", "Black Belt", "Southern")
price <- c(17, 20, 23)
elcprc <- c(0.01, 0.03, 0.06)

# Define the required columns
required_columns <- c("sprop", "array", "height",
                      "al_regs", "yldvar", "price", "elcprc")

# Check if the columns exist in tav_profit
```

```r
missing_columns <- setdiff(required_columns,
                           names(tav_profit))
if (length(missing_columns) > 0) {
  stop("Missing columns in tav_profit: ",
       paste(missing_columns, collapse = ", "))
}

# Generate column names using reversed order of expand.grid
col_names <- apply(expand.grid(height, array, sprop), 1,
                   function(x) paste0(x[3], "%_", x[2], "_", x[1]))

# Generate row names using reversed order of expand.grid
row_names <- apply(expand.grid(elcprc,
                               price,
                               yldvar,
                               al_regs), 1,
                   function(x) paste0(x, collapse = "_"))

# Create an empty matrix to store the results
result_matrix <- matrix(NA, nrow = length(row_names),
                        ncol = length(col_names))
colnames(result_matrix) <- col_names
rownames(result_matrix) <- row_names

# Create a data frame with
# all combinations of parameters in reversed order
param_combinations <- expand.grid(elcprc = elcprc,
                                  price = price,
                                  yldvar = yldvar,
                                  al_regs = al_regs,
                                  height = height,
                                  array = array,
                                  sprop = sprop)

# Merge with tav_profit to get av_profit values for each combination
merged_data <- merge(param_combinations,
                     tav_profit,
                     by = required_columns,
                     all.x = TRUE)

# Reshape merged_data to fill result_matrix with
# reversed column and row names
```

```
merged_data$col_name <- apply(
  merged_data[, c("sprop", "array", "height")], 1,
  function(x) paste0(x[1], "%_", x[2], "_", x[3]))

merged_data$row_name <- apply(
  merged_data[, c("al_regs", "yldvar", "price", "elcprc")], 1,
  function(x) paste0(x[4], "_",
                     x[3], "_",
                     x[2], "_", x[1]))

# Fill the matrix with av_profit values
for (i in seq_len(nrow(result_matrix))) {
  row_condition <- rownames(result_matrix)[i]
  row_data <- merged_data[
    merged_data$row_name == row_condition, ]
  if (nrow(row_data) > 0) {
    result_matrix[i,
                  match(row_data$col_name,
                        colnames(result_matrix))] <- round(
                          row_data$tav_profit, 2)
  }
}
```

```
write.csv(as.data.frame(result_matrix),
          row.names = TRUE,
          col.names = TRUE,
          file = "tav_chtbl.csv")
```

```
Warning in write.csv(as.data.frame(result_matrix), row.names = TRUE, col.names
= TRUE, : attempt to set 'col.names' ignored
```

```
# Display the result matrix
dim(as.data.frame(result_matrix))
```

```
[1] 324  30
```

- Row naming: Electricity Price_Crop Price_Solar Proportion_Alabama Regions
- Column naming: Solar Proportion_Array Types_Solar Panel Height.
- Solar Proportion can be converted to total number of panels.

- Only selected values from each variables are extracted for tabulation purpose.
- Values displayed in the table are profit from Tomato AV system.

```
head(result_matrix)
```

```
                    0.00%_Fixed_4.6 0.00%_Fixed_6.4 0.00%_Fixed_8.2
0.01_17_0.1_Northern         -8986.62        -8986.62        -8986.62
0.03_17_0.1_Northern         -8986.62        -8986.62        -8986.62
0.06_17_0.1_Northern         -8986.62        -8986.62        -8986.62
0.01_20_0.1_Northern         -8578.62        -8578.62        -8578.62
0.03_20_0.1_Northern         -8578.62        -8578.62        -8578.62
0.06_20_0.1_Northern         -8578.62        -8578.62        -8578.62
                    0.00%_Tracking_4.6 0.00%_Tracking_6.4 0.00%_Tracking_8.2
0.01_17_0.1_Northern            -8986.62           -8986.62           -8986.62
0.03_17_0.1_Northern            -8986.62           -8986.62           -8986.62
0.06_17_0.1_Northern            -8986.62           -8986.62           -8986.62
0.01_20_0.1_Northern            -8578.62           -8578.62           -8578.62
0.03_20_0.1_Northern            -8578.62           -8578.62           -8578.62
0.06_20_0.1_Northern            -8578.62           -8578.62           -8578.62
                    0.25%_Fixed_4.6 0.25%_Fixed_6.4 0.25%_Fixed_8.2
0.01_17_0.1_Northern         -7836.72        -7838.27        -7841.15
0.03_17_0.1_Northern         -5517.77        -5519.32        -5522.20
0.06_17_0.1_Northern         -2039.35        -2040.89        -2043.78
0.01_20_0.1_Northern         -7428.72        -7430.27        -7433.15
0.03_20_0.1_Northern         -5109.77        -5111.32        -5114.20
0.06_20_0.1_Northern         -1631.35        -1632.89        -1635.78
                    0.25%_Tracking_4.6 0.25%_Tracking_6.4 0.25%_Tracking_8.2
0.01_17_0.1_Northern            -7667.10           -7668.23           -7669.36
0.03_17_0.1_Northern            -5007.21           -5008.34           -5009.47
0.06_17_0.1_Northern            -1017.37           -1018.50           -1019.64
0.01_20_0.1_Northern            -7259.10           -7260.23           -7261.36
0.03_20_0.1_Northern            -4599.21           -4600.34           -4601.47
0.06_20_0.1_Northern             -609.37            -610.50            -611.64
                    0.50%_Fixed_4.6 0.50%_Fixed_6.4 0.50%_Fixed_8.2
0.01_17_0.1_Northern         -6304.10        -6307.70        -6314.44
0.03_17_0.1_Northern          -894.36         -897.96         -904.70
0.06_17_0.1_Northern          7220.25         7216.65         7209.91
0.01_20_0.1_Northern         -5896.10        -5899.70        -5906.44
0.03_20_0.1_Northern          -486.36         -489.96         -496.70
0.06_20_0.1_Northern          7628.25         7624.65         7617.91
                    0.50%_Tracking_4.6 0.50%_Tracking_6.4 0.50%_Tracking_8.2
0.01_17_0.1_Northern            -5908.09           -5910.73           -5913.37
```

```
0.03_17_0.1_Northern             297.61             294.97             292.33
0.06_17_0.1_Northern            9606.16            9603.52            9600.88
0.01_20_0.1_Northern           -5500.09           -5502.73           -5505.37
0.03_20_0.1_Northern             705.61             702.97             700.33
0.06_20_0.1_Northern           10014.16           10011.52           10008.88
                     0.75%_Fixed_4.6 0.75%_Fixed_6.4 0.75%_Fixed_8.2
0.01_17_0.1_Northern        -4771.19        -4776.85        -4787.43
0.03_17_0.1_Northern         3729.92         3724.26         3713.68
0.06_17_0.1_Northern        16481.58        16475.92        16465.34
0.01_20_0.1_Northern        -4363.19        -4368.85        -4379.43
0.03_20_0.1_Northern         4137.92         4132.26         4121.68
0.06_20_0.1_Northern        16889.58        16883.92        16873.34
                     0.75%_Tracking_4.6 0.75%_Tracking_6.4 0.75%_Tracking_8.2
0.01_17_0.1_Northern           -4148.97           -4153.12           -4157.27
0.03_17_0.1_Northern            5602.76            5598.61            5594.46
0.06_17_0.1_Northern           20230.36           20226.20           20222.05
0.01_20_0.1_Northern           -3740.97           -3745.12           -3749.27
0.03_20_0.1_Northern            6010.76            6006.61            6002.46
0.06_20_0.1_Northern           20638.36           20634.20           20630.05
                     1.00%_Fixed_4.6 1.00%_Fixed_6.4 1.00%_Fixed_8.2
0.01_17_0.1_Northern        -3238.30        -3246.01        -3260.44
0.03_17_0.1_Northern         8354.15         8346.44         8332.01
0.06_17_0.1_Northern        25742.83        25735.11        25720.68
0.01_20_0.1_Northern        -2830.30        -2838.01        -2852.44
0.03_20_0.1_Northern         8762.15         8754.44         8740.01
0.06_20_0.1_Northern        26150.83        26143.11        26128.68
                     1.00%_Tracking_4.6 1.00%_Tracking_6.4 1.00%_Tracking_8.2
0.01_17_0.1_Northern           -2389.85           -2395.51           -2401.18
0.03_17_0.1_Northern           10907.91           10902.25           10896.58
0.06_17_0.1_Northern           30854.55           30848.89           30843.22
0.01_20_0.1_Northern           -1981.85           -1987.51           -1993.18
0.03_20_0.1_Northern           11315.91           11310.25           11304.58
0.06_20_0.1_Northern           31262.55           31256.89           31251.22
```

```
tail(result_matrix)
```

```
                     0.00%_Fixed_4.6 0.00%_Fixed_6.4 0.00%_Fixed_8.2
0.01_20_2.0_Southern        29839.38        29839.38        29839.38
0.03_20_2.0_Southern        29839.38        29839.38        29839.38
0.06_20_2.0_Southern        29839.38        29839.38        29839.38
0.01_23_2.0_Southern        37999.38        37999.38        37999.38
0.03_23_2.0_Southern        37999.38        37999.38        37999.38
```

0.06_23_2.0_Southern	37999.38	37999.38	37999.38
	0.00%_Tracking_4.6	0.00%_Tracking_6.4	0.00%_Tracking_8.2
0.01_20_2.0_Southern	29839.38	29839.38	29839.38
0.03_20_2.0_Southern	29839.38	29839.38	29839.38
0.06_20_2.0_Southern	29839.38	29839.38	29839.38
0.01_23_2.0_Southern	37999.38	37999.38	37999.38
0.03_23_2.0_Southern	37999.38	37999.38	37999.38
0.06_23_2.0_Southern	37999.38	37999.38	37999.38
	0.25%_Fixed_4.6	0.25%_Fixed_6.4	0.25%_Fixed_8.2
0.01_20_2.0_Southern	31027.51	31025.96	31023.08
0.03_20_2.0_Southern	33422.92	33421.37	33418.49
0.06_20_2.0_Southern	37016.03	37014.49	37011.60
0.01_23_2.0_Southern	39187.51	39185.96	39183.08
0.03_23_2.0_Southern	41582.92	41581.37	41578.49
0.06_23_2.0_Southern	45176.03	45174.49	45171.60
	0.25%_Tracking_4.6	0.25%_Tracking_6.4	0.25%_Tracking_8.2
0.01_20_2.0_Southern	31219.96	31218.83	31217.70
0.03_20_2.0_Southern	34001.97	34000.84	33999.71
0.06_20_2.0_Southern	38174.99	38173.86	38172.72
0.01_23_2.0_Southern	39379.96	39378.83	39377.70
0.03_23_2.0_Southern	42161.97	42160.84	42159.71
0.06_23_2.0_Southern	46334.99	46333.86	46332.72
	0.50%_Fixed_4.6	0.50%_Fixed_6.4	0.50%_Fixed_8.2
0.01_20_2.0_Southern	32611.02	32607.42	32600.69
0.03_20_2.0_Southern	38199.01	38195.41	38188.68
0.06_20_2.0_Southern	46581.00	46577.40	46570.66
0.01_23_2.0_Southern	40771.02	40767.42	40760.69
0.03_23_2.0_Southern	46359.01	46355.41	46348.68
0.06_23_2.0_Southern	54741.00	54737.40	54730.66
	0.50%_Tracking_4.6	0.50%_Tracking_6.4	0.50%_Tracking_8.2
0.01_20_2.0_Southern	33060.37	33057.73	33055.08
0.03_20_2.0_Southern	39550.98	39548.34	39545.69
0.06_20_2.0_Southern	49286.89	49284.25	49281.61
0.01_23_2.0_Southern	41220.37	41217.73	41215.08
0.03_23_2.0_Southern	47710.98	47708.34	47705.69
0.06_23_2.0_Southern	57446.89	57444.25	57441.61
	0.75%_Fixed_4.6	0.75%_Fixed_6.4	0.75%_Fixed_8.2
0.01_20_2.0_Southern	34194.86	34189.20	34178.62
0.03_20_2.0_Southern	42976.07	42970.41	42959.83
0.06_20_2.0_Southern	56147.88	56142.22	56131.64
0.01_23_2.0_Southern	42354.86	42349.20	42338.62
0.03_23_2.0_Southern	51136.07	51130.41	51119.83
0.06_23_2.0_Southern	64307.88	64302.22	64291.64

	0.75%_Tracking_4.6	0.75%_Tracking_6.4	0.75%_Tracking_8.2
0.01_20_2.0_Southern	34900.90	34896.74	34892.59
0.03_20_2.0_Southern	45100.36	45096.20	45092.05
0.06_20_2.0_Southern	60399.55	60395.39	60391.24
0.01_23_2.0_Southern	43060.90	43056.74	43052.59
0.03_23_2.0_Southern	53260.36	53256.20	53252.05
0.06_23_2.0_Southern	68559.55	68555.39	68551.24
	1.00%_Fixed_4.6	1.00%_Fixed_6.4	1.00%_Fixed_8.2
0.01_20_2.0_Southern	35778.68	35770.97	35756.54
0.03_20_2.0_Southern	47753.09	47745.38	47730.95
0.06_20_2.0_Southern	65714.71	65706.99	65692.56
0.01_23_2.0_Southern	43938.68	43930.97	43916.54
0.03_23_2.0_Southern	55913.09	55905.38	55890.95
0.06_23_2.0_Southern	73874.71	73866.99	73852.56
	1.00%_Tracking_4.6	1.00%_Tracking_6.4	1.00%_Tracking_8.2
0.01_20_2.0_Southern	36741.42	36735.76	36730.09
0.03_20_2.0_Southern	50649.72	50644.06	50638.39
0.06_20_2.0_Southern	71512.17	71506.51	71500.84
0.01_23_2.0_Southern	44901.42	44895.76	44890.09
0.03_23_2.0_Southern	58809.72	58804.06	58798.39
0.06_23_2.0_Southern	79672.17	79666.51	79660.84

3.1.1 Heatmap

- Result suppressed.

3.2 Strawberry AV

```
# Define the values for each variable
sprop <- c(0, 0.25, 0.50, 0.75, 1.00)
array <- c("Fixed", "Tracking")
height <- c(4.6, 6.4, 8.2)
yldvar <- c(0.10, 0.30, 0.50, 0.70, 1.00, 1.20, 1.50, 1.80, 2.00)
al_regs <- c("Northern", "Central", "Black Belt", "Southern")
price <- c(3, 6, 9)
elcprc <- c(0.01, 0.03, 0.06)

# Define the required columns
required_columns <- c("sprop", "array", "height",
                      "al_regs", "yldvar", "price", "elcprc")
```

```r
# Check if the columns exist in tav_profit
missing_columns <- setdiff(required_columns,
                           names(sbav_profit))
if (length(missing_columns) > 0) {
  stop("Missing columns in sbav_profit: ",
       paste(missing_columns, collapse = ", "))
}

# Generate column names using reversed order of expand.grid
col_names <- apply(expand.grid(height, array, sprop), 1,
                   function(x) paste0(x[3], "%_", x[2], "_", x[1]))

# Generate row names using reversed order of expand.grid
row_names <- apply(expand.grid(elcprc,
                               price,
                               yldvar,
                               al_regs), 1,
                   function(x) paste0(x, collapse = "_"))

# Create an empty matrix to store the results
result_matrix <- matrix(NA, nrow = length(row_names),
                        ncol = length(col_names))
colnames(result_matrix) <- col_names
rownames(result_matrix) <- row_names

# Create a data frame with
# all combinations of parameters in reversed order
param_combinations <- expand.grid(elcprc = elcprc,
                                  price = price,
                                  yldvar = yldvar,
                                  al_regs = al_regs,
                                  height = height,
                                  array = array,
                                  sprop = sprop)

# Merge with tav_profit to get av_profit values for each combination
merged_data <- merge(param_combinations,
                     sbav_profit,
                     by = required_columns,
                     all.x = TRUE)

# Reshape merged_data to fill result_matrix with
```

```
# reversed column and row names
merged_data$col_name <- apply(
  merged_data[, c("sprop", "array", "height")], 1,
  function(x) paste0(x[1], "%_", x[2], "_", x[3]))

merged_data$row_name <- apply(
  merged_data[, c("al_regs", "yldvar", "price", "elcprc")], 1,
  function(x) paste0(x[4], "_",
                     x[3], "_",
                     x[2], "_", x[1]))

# Fill the matrix with av_profit values
for (i in seq_len(nrow(result_matrix))) {
  row_condition <- rownames(result_matrix)[i]
  row_data <- merged_data[
    merged_data$row_name == row_condition, ]
  if (nrow(row_data) > 0) {
    result_matrix[i,
                  match(row_data$col_name,
                        colnames(result_matrix))] <- round(
                          row_data$sbav_profit, 2)
  }
}
```

```
write.csv(as.data.frame(result_matrix),
          row.names = TRUE,
          col.names = TRUE,
          file = "sbav_chtbl.csv")
```

```
Warning in write.csv(as.data.frame(result_matrix), row.names = TRUE, col.names
= TRUE, : attempt to set 'col.names' ignored
```

```
# Display the result matrix
dim(as.data.frame(result_matrix))
```

```
[1] 324  30
```

- Row naming: Electricity Price_Crop Price_Solar Proportion_Alabama Regions
- Column naming: Solar Proportion_Array Types_Solar Panel Height.
- Solar Proportion can be converted to total number of panels.

- Only selected values from each variables are extracted for tabulation purpose.
- Values displayed in the table are profit from Strawberry AV system.

```
head(result_matrix)
```

```
                   0.00%_Fixed_4.6 0.00%_Fixed_6.4 0.00%_Fixed_8.2
0.01_3_0.1_Northern       -12700.74       -12700.74       -12700.74
0.03_3_0.1_Northern       -12700.74       -12700.74       -12700.74
0.06_3_0.1_Northern       -12700.74       -12700.74       -12700.74
0.01_6_0.1_Northern       -11778.24       -11778.24       -11778.24
0.03_6_0.1_Northern       -11778.24       -11778.24       -11778.24
0.06_6_0.1_Northern       -11778.24       -11778.24       -11778.24
                   0.00%_Tracking_4.6 0.00%_Tracking_6.4 0.00%_Tracking_8.2
0.01_3_0.1_Northern          -12700.74          -12700.74          -12700.74
0.03_3_0.1_Northern          -12700.74          -12700.74          -12700.74
0.06_3_0.1_Northern          -12700.74          -12700.74          -12700.74
0.01_6_0.1_Northern          -11778.24          -11778.24          -11778.24
0.03_6_0.1_Northern          -11778.24          -11778.24          -11778.24
0.06_6_0.1_Northern          -11778.24          -11778.24          -11778.24
                   0.25%_Fixed_4.6 0.25%_Fixed_6.4 0.25%_Fixed_8.2
0.01_3_0.1_Northern       -11550.85       -11552.39       -11555.28
0.03_3_0.1_Northern        -9231.90        -9233.44        -9236.33
0.06_3_0.1_Northern        -5753.47        -5755.01        -5757.90
0.01_6_0.1_Northern       -10628.35       -10629.89       -10632.78
0.03_6_0.1_Northern        -8309.40        -8310.94        -8313.83
0.06_6_0.1_Northern        -4830.97        -4832.51        -4835.40
                   0.25%_Tracking_4.6 0.25%_Tracking_6.4 0.25%_Tracking_8.2
0.01_3_0.1_Northern          -11381.22          -11382.35          -11383.48
0.03_3_0.1_Northern           -8721.33           -8722.46           -8723.59
0.06_3_0.1_Northern           -4731.49           -4732.63           -4733.76
0.01_6_0.1_Northern          -10458.72          -10459.85          -10460.98
0.03_6_0.1_Northern           -7798.83           -7799.96           -7801.09
0.06_6_0.1_Northern           -3808.99           -3810.13           -3811.26
                   0.50%_Fixed_4.6 0.50%_Fixed_6.4 0.50%_Fixed_8.2
0.01_3_0.1_Northern       -10018.22       -10021.83       -10028.56
0.03_3_0.1_Northern        -4608.48        -4612.09        -4618.82
0.06_3_0.1_Northern         3506.13         3502.52         3495.79
0.01_6_0.1_Northern        -9095.72        -9099.33        -9106.06
0.03_6_0.1_Northern        -3685.98        -3689.59        -3696.32
0.06_6_0.1_Northern         4428.63         4425.02         4418.29
                   0.50%_Tracking_4.6 0.50%_Tracking_6.4 0.50%_Tracking_8.2
0.01_3_0.1_Northern           -9622.21           -9624.85           -9627.49
```

```
0.03_3_0.1_Northern           -3416.51           -3419.15           -3421.79
0.06_3_0.1_Northern            5892.04            5889.40            5886.76
0.01_6_0.1_Northern           -8699.71           -8702.35           -8704.99
0.03_6_0.1_Northern           -2494.01           -2496.65           -2499.29
0.06_6_0.1_Northern            6814.54            6811.90            6809.26
                    0.75%_Fixed_4.6 0.75%_Fixed_6.4 0.75%_Fixed_8.2
0.01_3_0.1_Northern        -8485.31        -8490.97        -8501.56
0.03_3_0.1_Northern           15.80           10.14           -0.45
0.06_3_0.1_Northern        12767.46        12761.80        12751.22
0.01_6_0.1_Northern        -7562.81        -7568.47        -7579.06
0.03_6_0.1_Northern          938.30          932.64          922.05
0.06_6_0.1_Northern        13689.96        13684.30        13673.72
                    0.75%_Tracking_4.6 0.75%_Tracking_6.4 0.75%_Tracking_8.2
0.01_3_0.1_Northern           -7863.09           -7867.24           -7871.40
0.03_3_0.1_Northern            1888.64            1884.49            1880.33
0.06_3_0.1_Northern           16516.23           16512.08           16507.93
0.01_6_0.1_Northern           -6940.59           -6944.74           -6948.90
0.03_6_0.1_Northern            2811.14            2806.99            2802.83
0.06_6_0.1_Northern           17438.73           17434.58           17430.43
                    1.00%_Fixed_4.6 1.00%_Fixed_6.4 1.00%_Fixed_8.2
0.01_3_0.1_Northern        -6952.42        -6960.14        -6974.57
0.03_3_0.1_Northern         4640.03         4632.31         4617.88
0.06_3_0.1_Northern        22028.71        22020.99        22006.56
0.01_6_0.1_Northern        -6029.92        -6037.64        -6052.07
0.03_6_0.1_Northern         5562.53         5554.81         5540.38
0.06_6_0.1_Northern        22951.21        22943.49        22929.06
                    1.00%_Tracking_4.6 1.00%_Tracking_6.4 1.00%_Tracking_8.2
0.01_3_0.1_Northern           -6103.97           -6109.64           -6115.30
0.03_3_0.1_Northern            7193.79            7188.12            7182.46
0.06_3_0.1_Northern           27140.43           27134.76           27129.10
0.01_6_0.1_Northern           -5181.47           -5187.14           -5192.80
0.03_6_0.1_Northern            8116.29            8110.62            8104.96
0.06_6_0.1_Northern           28062.93           28057.26           28051.60
```

```r
tail(result_matrix)
```

```
                    0.00%_Fixed_4.6 0.00%_Fixed_6.4 0.00%_Fixed_8.2
0.01_6_2.0_Southern        16709.51        16709.51        16709.51
0.03_6_2.0_Southern        16709.51        16709.51        16709.51
0.06_6_2.0_Southern        16709.51        16709.51        16709.51
0.01_9_2.0_Southern        35159.51        35159.51        35159.51
0.03_9_2.0_Southern        35159.51        35159.51        35159.51
```

0.06_9_2.0_Southern	35159.51	35159.51	35159.51
	0.00%_Tracking_4.6	0.00%_Tracking_6.4	0.00%_Tracking_8.2
0.01_6_2.0_Southern	16709.51	16709.51	16709.51
0.03_6_2.0_Southern	16709.51	16709.51	16709.51
0.06_6_2.0_Southern	16709.51	16709.51	16709.51
0.01_9_2.0_Southern	35159.51	35159.51	35159.51
0.03_9_2.0_Southern	35159.51	35159.51	35159.51
0.06_9_2.0_Southern	35159.51	35159.51	35159.51
	0.25%_Fixed_4.6	0.25%_Fixed_6.4	0.25%_Fixed_8.2
0.01_6_2.0_Southern	17897.63	17896.09	17893.20
0.03_6_2.0_Southern	20293.04	20291.50	20288.61
0.06_6_2.0_Southern	23886.15	23884.61	23881.72
0.01_9_2.0_Southern	36347.63	36346.09	36343.20
0.03_9_2.0_Southern	38743.04	38741.50	38738.61
0.06_9_2.0_Southern	42336.15	42334.61	42331.72
	0.25%_Tracking_4.6	0.25%_Tracking_6.4	0.25%_Tracking_8.2
0.01_6_2.0_Southern	18090.09	18088.95	18087.82
0.03_6_2.0_Southern	20872.10	20870.96	20869.83
0.06_6_2.0_Southern	25045.11	25043.98	25042.85
0.01_9_2.0_Southern	36540.09	36538.95	36537.82
0.03_9_2.0_Southern	39322.10	39320.96	39319.83
0.06_9_2.0_Southern	43495.11	43493.98	43492.85
	0.50%_Fixed_4.6	0.50%_Fixed_6.4	0.50%_Fixed_8.2
0.01_6_2.0_Southern	19481.15	19477.54	19470.81
0.03_6_2.0_Southern	25069.14	25065.53	25058.80
0.06_6_2.0_Southern	33451.12	33447.52	33440.78
0.01_9_2.0_Southern	37931.15	37927.54	37920.81
0.03_9_2.0_Southern	43519.14	43515.53	43508.80
0.06_9_2.0_Southern	51901.12	51897.52	51890.78
	0.50%_Tracking_4.6	0.50%_Tracking_6.4	0.50%_Tracking_8.2
0.01_6_2.0_Southern	19930.49	19927.85	19925.21
0.03_6_2.0_Southern	26421.10	26418.46	26415.82
0.06_6_2.0_Southern	36157.02	36154.37	36151.73
0.01_9_2.0_Southern	38380.49	38377.85	38375.21
0.03_9_2.0_Southern	44871.10	44868.46	44865.82
0.06_9_2.0_Southern	54607.02	54604.37	54601.73
	0.75%_Fixed_4.6	0.75%_Fixed_6.4	0.75%_Fixed_8.2
0.01_6_2.0_Southern	21064.98	21059.32	21048.74
0.03_6_2.0_Southern	29846.19	29840.53	29829.95
0.06_6_2.0_Southern	43018.01	43012.35	43001.76
0.01_9_2.0_Southern	39514.98	39509.32	39498.74
0.03_9_2.0_Southern	48296.19	48290.53	48279.95
0.06_9_2.0_Southern	61468.01	61462.35	61451.76

```
                     0.75%_Tracking_4.6 0.75%_Tracking_6.4 0.75%_Tracking_8.2
0.01_6_2.0_Southern           21771.02           21766.87           21762.71
0.03_6_2.0_Southern           31970.48           31966.33           31962.17
0.06_6_2.0_Southern           47269.67           47265.52           47261.36
0.01_9_2.0_Southern           40221.02           40216.87           40212.71
0.03_9_2.0_Southern           50420.48           50416.33           50412.17
0.06_9_2.0_Southern           65719.67           65715.52           65711.36
                     1.00%_Fixed_4.6 1.00%_Fixed_6.4 1.00%_Fixed_8.2
0.01_6_2.0_Southern        22648.81        22641.09        22626.66
0.03_6_2.0_Southern        34623.22        34615.50        34601.07
0.06_6_2.0_Southern        52584.83        52577.11        52562.68
0.01_9_2.0_Southern        41098.81        41091.09        41076.66
0.03_9_2.0_Southern        53073.22        53065.50        53051.07
0.06_9_2.0_Southern        71034.83        71027.11        71012.68
                     1.00%_Tracking_4.6 1.00%_Tracking_6.4 1.00%_Tracking_8.2
0.01_6_2.0_Southern           23611.54           23605.88           23600.22
0.03_6_2.0_Southern           37519.84           37514.18           37508.52
0.06_6_2.0_Southern           58382.29           58376.63           58370.97
0.01_9_2.0_Southern           42061.54           42055.88           42050.22
0.03_9_2.0_Southern           55969.84           55964.18           55958.52
0.06_9_2.0_Southern           76832.29           76826.63           76820.97
```

3.2.1 Heatmap

- Result suppressed.

3.3 Squash AV

```r
# Define the values for each variable
sprop <- c(0, 0.25, 0.50, 0.75, 1.00)
array <- c("Fixed", "Tracking")
height <- c(4.6, 6.4, 8.2)
yldvar <- c(0.10, 0.30, 0.50, 0.70, 1.00, 1.20, 1.50, 1.80, 2.00)
al_regs <- c("Northern", "Central", "Black Belt", "Southern")
price <- c(11, 14, 17)
elcprc <- c(0.01, 0.03, 0.06)

# Define the required columns
required_columns <- c("sprop", "array", "height",
                      "al_regs", "yldvar", "price", "elcprc")
```

```r
# Check if the columns exist in tav_profit
missing_columns <- setdiff(required_columns,
                           names(sqav_profit))
if (length(missing_columns) > 0) {
  stop("Missing columns in sqav_profit: ",
       paste(missing_columns, collapse = ", "))
}

# Generate column names using reversed order of expand.grid
col_names <- apply(expand.grid(height, array, sprop), 1,
                   function(x) paste0(x[3], "%_", x[2], "_", x[1]))

# Generate row names using reversed order of expand.grid
row_names <- apply(expand.grid(elcprc,
                               price,
                               yldvar,
                               al_regs), 1,
                   function(x) paste0(x, collapse = "_"))

# Create an empty matrix to store the results
result_matrix <- matrix(NA, nrow = length(row_names),
                        ncol = length(col_names))
colnames(result_matrix) <- col_names
rownames(result_matrix) <- row_names

# Create a data frame with
# all combinations of parameters in reversed order
param_combinations <- expand.grid(elcprc = elcprc,
                                  price = price,
                                  yldvar = yldvar,
                                  al_regs = al_regs,
                                  height = height,
                                  array = array,
                                  sprop = sprop)

# Merge with tav_profit to get av_profit values for each combination
merged_data <- merge(param_combinations,
                     sqav_profit,
                     by = required_columns,
                     all.x = TRUE)

# Reshape merged_data to fill result_matrix with
```

```
# reversed column and row names
merged_data$col_name <- apply(
  merged_data[, c("sprop", "array", "height")], 1,
  function(x) paste0(x[1], "%_", x[2], "_", x[3]))

merged_data$row_name <- apply(
  merged_data[, c("al_regs", "yldvar", "price", "elcprc")], 1,
  function(x) paste0(x[4], "_",
                     x[3], "_",
                     x[2], "_", x[1]))

# Fill the matrix with av_profit values
for (i in seq_len(nrow(result_matrix))) {
  row_condition <- rownames(result_matrix)[i]
  row_data <- merged_data[
    merged_data$row_name == row_condition, ]
  if (nrow(row_data) > 0) {
    result_matrix[i,
                  match(row_data$col_name,
                        colnames(result_matrix))] <- round(
                          row_data$sqav_profit, 2)
  }
}
```

```
write.csv(as.data.frame(result_matrix),
          row.names = TRUE,
          file = "sqav_chtbl.csv")
# Display the result matrix
dim(as.data.frame(result_matrix))
```

```
[1] 324  30
```

- Row naming: Electricity Price_Crop Price_Solar Proportion_Alabama Regions
- Column naming: Solar Proportion_Array Types_Solar Panel Height.
- Solar Proportion can be converted to total number of panels.
- Only selected values from each variables are extracted for tabulation purpose.
- Values displayed in the table are profit from Squash AV system.

```
head(result_matrix)
```

```
                    0.00%_Fixed_4.6 0.00%_Fixed_6.4 0.00%_Fixed_8.2
0.01_11_0.1_Northern         -3024.13        -3024.13        -3024.13
0.03_11_0.1_Northern         -3024.13        -3024.13        -3024.13
0.06_11_0.1_Northern         -3024.13        -3024.13        -3024.13
0.01_14_0.1_Northern         -2697.13        -2697.13        -2697.13
0.03_14_0.1_Northern         -2697.13        -2697.13        -2697.13
0.06_14_0.1_Northern         -2697.13        -2697.13        -2697.13
                    0.00%_Tracking_4.6 0.00%_Tracking_6.4 0.00%_Tracking_8.2
0.01_11_0.1_Northern            -3024.13           -3024.13           -3024.13
0.03_11_0.1_Northern            -3024.13           -3024.13           -3024.13
0.06_11_0.1_Northern            -3024.13           -3024.13           -3024.13
0.01_14_0.1_Northern            -2697.13           -2697.13           -2697.13
0.03_14_0.1_Northern            -2697.13           -2697.13           -2697.13
0.06_14_0.1_Northern            -2697.13           -2697.13           -2697.13
                    0.25%_Fixed_4.6 0.25%_Fixed_6.4 0.25%_Fixed_8.2
0.01_11_0.1_Northern         -1874.24        -1875.78        -1878.67
0.03_11_0.1_Northern           444.71          443.17          440.28
0.06_11_0.1_Northern          3923.14         3921.59         3918.71
0.01_14_0.1_Northern         -1547.24        -1548.78        -1551.67
0.03_14_0.1_Northern           771.71          770.17          767.28
0.06_14_0.1_Northern          4250.14         4248.59         4245.71
                    0.25%_Tracking_4.6 0.25%_Tracking_6.4 0.25%_Tracking_8.2
0.01_11_0.1_Northern            -1704.61           -1705.74           -1706.88
0.03_11_0.1_Northern              955.28             954.15             953.01
0.06_11_0.1_Northern             4945.11            4943.98            4942.85
0.01_14_0.1_Northern            -1377.61           -1378.74           -1379.88
0.03_14_0.1_Northern             1282.28            1281.15            1280.01
0.06_14_0.1_Northern             5272.11            5270.98            5269.85
                    0.50%_Fixed_4.6 0.50%_Fixed_6.4 0.50%_Fixed_8.2
0.01_11_0.1_Northern         -341.62         -345.22         -351.95
0.03_11_0.1_Northern         5068.12         5064.52         5057.79
0.06_11_0.1_Northern        13182.73        13179.13        13172.40
0.01_14_0.1_Northern          -14.62          -18.22          -24.95
0.03_14_0.1_Northern         5395.12         5391.52         5384.79
0.06_14_0.1_Northern        13509.73        13506.13        13499.40
                    0.50%_Tracking_4.6 0.50%_Tracking_6.4 0.50%_Tracking_8.2
0.01_11_0.1_Northern              54.40              51.76              49.11
0.03_11_0.1_Northern            6260.10            6257.46            6254.81
0.06_11_0.1_Northern           15568.65           15566.01           15563.36
0.01_14_0.1_Northern             381.40             378.76             376.11
```

```
0.03_14_0.1_Northern            6587.10            6584.46            6581.81
0.06_14_0.1_Northern           15895.65           15893.01           15890.36
                    0.75%_Fixed_4.6 0.75%_Fixed_6.4 0.75%_Fixed_8.2
0.01_11_0.1_Northern         1191.29         1185.63         1175.05
0.03_11_0.1_Northern         9692.40         9686.74         9676.16
0.06_11_0.1_Northern        22444.07        22438.41        22427.83
0.01_14_0.1_Northern         1518.29         1512.63         1502.05
0.03_14_0.1_Northern        10019.40        10013.74        10003.16
0.06_14_0.1_Northern        22771.07        22765.41        22754.83
                    0.75%_Tracking_4.6 0.75%_Tracking_6.4 0.75%_Tracking_8.2
0.01_11_0.1_Northern            1813.52            1809.36            1805.21
0.03_11_0.1_Northern           11565.25           11561.09           11556.94
0.06_11_0.1_Northern           26192.84           26188.69           26184.54
0.01_14_0.1_Northern            2140.52            2136.36            2132.21
0.03_14_0.1_Northern           11892.25           11888.09           11883.94
0.06_14_0.1_Northern           26519.84           26515.69           26511.54
                    1.00%_Fixed_4.6 1.00%_Fixed_6.4 1.00%_Fixed_8.2
0.01_11_0.1_Northern         2724.19         2716.47         2702.04
0.03_11_0.1_Northern        14316.64        14308.92        14294.49
0.06_11_0.1_Northern        31705.31        31697.60        31683.16
0.01_14_0.1_Northern         3051.19         3043.47         3029.04
0.03_14_0.1_Northern        14643.64        14635.92        14621.49
0.06_14_0.1_Northern        32032.31        32024.60        32010.16
                    1.00%_Tracking_4.6 1.00%_Tracking_6.4 1.00%_Tracking_8.2
0.01_11_0.1_Northern            3572.63            3566.97            3561.31
0.03_11_0.1_Northern           16870.39           16864.73           16859.07
0.06_11_0.1_Northern           36817.03           36811.37           36805.71
0.01_14_0.1_Northern            3899.63            3893.97            3888.31
0.03_14_0.1_Northern           17197.39           17191.73           17186.07
0.06_14_0.1_Northern           37144.03           37138.37           37132.71
```

```
tail(result_matrix)
```

```
                    0.00%_Fixed_4.6 0.00%_Fixed_6.4 0.00%_Fixed_8.2
0.01_14_2.0_Southern        16849.12        16849.12        16849.12
0.03_14_2.0_Southern        16849.12        16849.12        16849.12
0.06_14_2.0_Southern        16849.12        16849.12        16849.12
0.01_17_2.0_Southern        23389.12        23389.12        23389.12
0.03_17_2.0_Southern        23389.12        23389.12        23389.12
0.06_17_2.0_Southern        23389.12        23389.12        23389.12
                    0.00%_Tracking_4.6 0.00%_Tracking_6.4 0.00%_Tracking_8.2
0.01_14_2.0_Southern           16849.12           16849.12           16849.12
```

0.03_14_2.0_Southern	16849.12	16849.12	16849.12
0.06_14_2.0_Southern	16849.12	16849.12	16849.12
0.01_17_2.0_Southern	23389.12	23389.12	23389.12
0.03_17_2.0_Southern	23389.12	23389.12	23389.12
0.06_17_2.0_Southern	23389.12	23389.12	23389.12

	0.25%_Fixed_4.6	0.25%_Fixed_6.4	0.25%_Fixed_8.2
0.01_14_2.0_Southern	18037.24	18035.70	18032.81
0.03_14_2.0_Southern	20432.65	20431.11	20428.22
0.06_14_2.0_Southern	24025.77	24024.22	24021.34
0.01_17_2.0_Southern	24577.24	24575.70	24572.81
0.03_17_2.0_Southern	26972.65	26971.11	26968.22
0.06_17_2.0_Southern	30565.77	30564.22	30561.34

	0.25%_Tracking_4.6	0.25%_Tracking_6.4	0.25%_Tracking_8.2
0.01_14_2.0_Southern	18229.70	18228.57	18227.43
0.03_14_2.0_Southern	21011.71	21010.58	21009.44
0.06_14_2.0_Southern	25184.72	25183.59	25182.46
0.01_17_2.0_Southern	24769.70	24768.57	24767.43
0.03_17_2.0_Southern	27551.71	27550.58	27549.44
0.06_17_2.0_Southern	31724.72	31723.59	31722.46

	0.50%_Fixed_4.6	0.50%_Fixed_6.4	0.50%_Fixed_8.2
0.01_14_2.0_Southern	19620.76	19617.16	19610.42
0.03_14_2.0_Southern	25208.75	25205.15	25198.41
0.06_14_2.0_Southern	33590.73	33587.13	33580.40
0.01_17_2.0_Southern	26160.76	26157.16	26150.42
0.03_17_2.0_Southern	31748.75	31745.15	31738.41
0.06_17_2.0_Southern	40130.73	40127.13	40120.40

	0.50%_Tracking_4.6	0.50%_Tracking_6.4	0.50%_Tracking_8.2
0.01_14_2.0_Southern	20070.10	20067.46	20064.82
0.03_14_2.0_Southern	26560.71	26558.07	26555.43
0.06_14_2.0_Southern	36296.63	36293.99	36291.34
0.01_17_2.0_Southern	26610.10	26607.46	26604.82
0.03_17_2.0_Southern	33100.71	33098.07	33095.43
0.06_17_2.0_Southern	42836.63	42833.99	42831.34

	0.75%_Fixed_4.6	0.75%_Fixed_6.4	0.75%_Fixed_8.2
0.01_14_2.0_Southern	21204.59	21198.93	21188.35
0.03_14_2.0_Southern	29985.80	29980.14	29969.56
0.06_14_2.0_Southern	43157.62	43151.96	43141.38
0.01_17_2.0_Southern	27744.59	27738.93	27728.35
0.03_17_2.0_Southern	36525.80	36520.14	36509.56
0.06_17_2.0_Southern	49697.62	49691.96	49681.38

	0.75%_Tracking_4.6	0.75%_Tracking_6.4	0.75%_Tracking_8.2
0.01_14_2.0_Southern	21910.63	21906.48	21902.33
0.03_14_2.0_Southern	32110.09	32105.94	32101.79

0.06_14_2.0_Southern	47409.28	47405.13	47400.98
0.01_17_2.0_Southern	28450.63	28446.48	28442.33
0.03_17_2.0_Southern	38650.09	38645.94	38641.79
0.06_17_2.0_Southern	53949.28	53945.13	53940.98

	1.00%_Fixed_4.6	1.00%_Fixed_6.4	1.00%_Fixed_8.2
0.01_14_2.0_Southern	22788.42	22780.70	22766.27
0.03_14_2.0_Southern	34762.83	34755.11	34740.68
0.06_14_2.0_Southern	52724.44	52716.73	52702.29
0.01_17_2.0_Southern	29328.42	29320.70	29306.27
0.03_17_2.0_Southern	41302.83	41295.11	41280.68
0.06_17_2.0_Southern	59264.44	59256.73	59242.29

	1.00%_Tracking_4.6	1.00%_Tracking_6.4	1.00%_Tracking_8.2
0.01_14_2.0_Southern	23751.15	23745.49	23739.83
0.03_14_2.0_Southern	37659.45	37653.79	37648.13
0.06_14_2.0_Southern	58521.90	58516.24	58510.58
0.01_17_2.0_Southern	30291.15	30285.49	30279.83
0.03_17_2.0_Southern	44199.45	44193.79	44188.13
0.06_17_2.0_Southern	65061.90	65056.24	65050.58

3.3.1 Heatmap

- Result suppressed.