AV Simulation REAP50

Bijesh Mishra, Ph.D.

2024-12-11

Table of contents

Collocating Specialty Crops and Solar panels in Alabama, Southeastern USA. A paper for Choice Magazine, AAEA.

1 Setting Up

1.1 Housekeeping

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

1.2 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 becom
```

```
library(psych, warn.conflicts = FALSE, quietly = TRUE)
library(likert,  warn.conflicts = FALSE, quietly = TRUE)
library(mice,  warn.conflicts = FALSE, quietly = TRUE)
library(openxlsx2, warn.conflicts = FALSE, quietly = TRUE)
library(ggpubr, warn.conflicts = FALSE, quietly = TRUE)
library(gmodels,  warn.conflicts = FALSE, quietly = TRUE)
library(reshape2, warn.conflicts = FALSE, quietly = TRUE)
library(pacman,  warn.conflicts = FALSE, quietly = TRUE)
library(progress, warn.conflicts = FALSE, quietly = TRUE)
library(arrow, warn.conflicts = FALSE, quietly = TRUE)
```

1.3 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.key = element_rect(color = "black",
                                  fill = NA,
                                  linewidth = 0.05,
                                  linetype = 1),
        legend.justification = "right",
        legend.direction = "horizontal")
```

2 Import data

Import necessary data.

2.1 Tomato

- Yield = Total tomato production (total bucket of 25 lb) from 1 acres of land which varies from 10% to 200% of total production (100%). The range was simulated by multiplying 100% yield by yldvar.
- yldvar = Yield variation parameter ranges from 10% to 200%.
- Rev17 to Rev23 = Revenue for price ranges of $17 to $23 per bucket of tomato.
- Total cost = Total cost of production for the given yield.
- rolac17 to rolac23= Return to operator, labor and capital for price range of $17 to $23.
- operator Cost = Operator labor cost at $15/hour for given yield. For 100% yield, total hours = 90.
- rlc17 to 23 = Return to land and capital after subtracting operator cost from total revenue.

```r
tomato <- read_xlsx("Data/Parameters.xlsx",
                    sheet = "Tomato",
                    start_row = 2,
                    start_col = 9,
                    skip_empty_rows = TRUE,
                    skip_empty_cols = TRUE,
                    col_names = TRUE) %>%
  rename(yield = Yield,
         yldvar = `Yield Variation (%)`)
str(tomato)
```

```
'data.frame':	21 obs. of  25 variables:
 $ yldvar       : num  2 1.9 1.8 1.7 1.6 1.5 1.4 1.3 1.2 1.1 ...
 $ yield        : num  2720 2584 2448 2312 2176 ...
 $ Rev17        : num  46240 43928 41616 39304 36992 ...
 $ Rev18        : num  48960 46512 44064 41616 39168 ...
 $ Rev19        : num  51680 49096 46512 43928 41344 ...
 $ Rev20        : num  54400 51680 48960 46240 43520 ...
 $ Rev21        : num  57120 54264 51408 48552 45696 ...
 $ Rev22        : num  59840 56848 53856 50864 47872 ...
 $ Rev23        : num  62560 59432 56304 53176 50048 ...
 $ Total Cost   : num  24561 23863 23165 22467 21769 ...
 $ rolac17      : num  21679 20065 18451 16837 15223 ...
 $ rolac18      : num  24399 22649 20899 19149 17399 ...
 $ rolac19      : num  27119 25233 23347 21461 19575 ...
 $ rolac20      : num  29839 27817 25795 23773 21751 ...
 $ rolac21      : num  32559 30401 28243 26085 23927 ...
 $ rolac22      : num  35279 32985 30691 28397 26103 ...
 $ rolac23      : num  37999 35569 33139 30709 28279 ...
 $ Operator Cost: num  2700 2565 2430 2295 2160 ...
 $ rlc17        : num  18979 17500 16021 14542 13063 ...
 $ rlc18        : num  21699 20084 18469 16854 15239 ...
 $ rlc19        : num  24419 22668 20917 19166 17415 ...
 $ rlc20        : num  27139 25252 23365 21478 19591 ...
 $ rlc21        : num  29859 27836 25813 23790 21767 ...
 $ rlc22        : num  32579 30420 28261 26102 23943 ...
 $ rlc223       : num  35299 33004 30709 28414 26119 ...
```

2.2 Strawberry

- Everything same as tomato.

- Numbers 3 to 9 in names are price ranges for strawberry.

```r
strawberry <- read_xlsx("Data/Parameters.xlsx",
                        sheet = "Strawberry",
                        start_row = 2,
                        start_col = 7,
                        skip_empty_rows = TRUE,
                        skip_empty_cols = TRUE,
                        col_names = TRUE) %>%
  rename(yield = Yield,
         yldvar = `Yield Variation (%)`)
str(strawberry)
```

```
'data.frame':	21 obs. of  25 variables:
 $ yldvar       : num  2 1.9 1.8 1.7 1.6 1.5 1.4 1.3 1.2 1.1 ...
 $ yield        : num  6150 5843 5535 5228 4920 ...
 $ Rev3         : num  18450 17529 16605 15684 14760 ...
 $ Rev4         : num  24600 23372 22140 20912 19680 ...
 $ Rev5         : num  30750 29215 27675 26140 24600 ...
 $ Rev6         : num  36900 35058 33210 31368 29520 ...
 $ Rev7         : num  43050 40901 38745 36596 34440 ...
 $ Rev8         : num  49200 46744 44280 41824 39360 ...
 $ Rev9         : num  55350 52587 49815 47052 44280 ...
 $ Total Cost   : num  17731 17386 17040 16694 16348 ...
 $ rolac3       : num  719 143 -435 -1010 -1588 ...
 $ rolac4       : num  6869 5986 5100 4218 3332 ...
 $ rolac5       : num  13019 11829 10635 9446 8252 ...
 $ rolac6       : num  19169 17672 16170 14674 13172 ...
 $ rolac7       : num  25319 23515 21705 19902 18092 ...
 $ rolac8       : num  31469 29358 27240 25130 23012 ...
 $ rolac9       : num  37619 35201 32775 30358 27932 ...
 $ Operator Cost: num  2700 2565 2430 2295 2160 ...
 $ rlc3         : num  -1981 -2422 -2865 -3306 -3748 ...
 $ rlc4         : num  4169 3421 2670 1922 1172 ...
 $ rlc5         : num  10319 9264 8205 7150 6092 ...
 $ rlc6         : num  16469 15107 13740 12378 11012 ...
 $ rlc7         : num  22619 20950 19275 17606 15932 ...
 $ rlc8         : num  28769 26793 24810 22834 20852 ...
 $ rlc9         : num  34919 32636 30345 28062 25772 ...
```

2.3 Squash

- Everything same as tomato and strawberry.
- Numbers 11 to 17 in names are price ranges for squash.

```
squash <- read_xlsx("Data/Parameters.xlsx",
                    sheet = "Squash",
                    start_row = 2,
                    start_col = 9,
                    skip_empty_rows = TRUE,
                    skip_empty_cols = TRUE,
                    col_names = TRUE) %>%
  rename(yield = Yield,
         yldvar = `Yield Variation (%)`)
str(squash)
```

```
'data.frame':   21 obs. of  25 variables:
 $ yldvar       : num  2 1.9 1.8 1.7 1.6 1.5 1.4 1.3 1.2 1.1 ...
 $ yield        : num  2180 2071 1962 1853 1744 ...
 $ Rev11        : num  23980 22781 21582 20383 19184 ...
 $ Rev12        : num  26160 24852 23544 22236 20928 ...
 $ Rev13        : num  28340 26923 25506 24089 22672 ...
 $ Rev14        : num  30520 28994 27468 25942 24416 ...
 $ Rev15        : num  32700 31065 29430 27795 26160 ...
 $ Rev16        : num  34880 33136 31392 29648 27904 ...
 $ Rev17        : num  37060 35207 33354 31501 29648 ...
 $ Total Cost   : num  13671 13174 12676 12179 11682 ...
 $ rolac11      : num  10309 9607 8906 8204 7502 ...
 $ rolac12      : num  12489 11678 10868 10057 9246 ...
 $ rolac13      : num  14669 13749 12830 11910 10990 ...
 $ rolac14      : num  16849 15820 14792 13763 12734 ...
 $ rolac15      : num  19029 17891 16754 15616 14478 ...
 $ rolac16      : num  21209 19962 18716 17469 16222 ...
 $ rolac17      : num  23389 22033 20678 19322 17966 ...
 $ Operator Cost: num  2700 2565 2430 2295 2160 ...
 $ rlc11        : num  7609 7042 6476 5909 5342 ...
 $ rlc12        : num  9789 9113 8438 7762 7086 ...
 $ rlc13        : num  11969 11184 10400 9615 8830 ...
 $ rlc14        : num  14149 13255 12362 11468 10574 ...
 $ rlc15        : num  16329 15326 14324 13321 12318 ...
 $ rlc16        : num  18509 17397 16286 15174 14062 ...
 $ rlc17        : num  20689 19468 18248 17027 15806 ...
```

2.4 Electricity price

Electricity price ranges from 1 cents to 6 cents in 0.5 cent increment. Previously, I used AL retail electricity price as described below. It's no longer in use but I put description below for the record.

Electricity price ($/kWh) was retail electricity price range for Alabama based on retail electricity price in April 2023 and April 2024 taken from DOE Database. Retail electricity price range in Alabama was from 6.44 to 15.85 cents/kWh in April 2023 and April 2024 which represents industry, commercial, and residential prices.

```
elec_price <- read_xlsx("Data/Parameters.xlsx",
                              sheet = "Electricity Price") %>%
  rename(epr_kwh = `Electricity Price ($/kWh)`)
as.list(elec_price)
```

```
$epr_kwh
 [1] 0.010 0.015 0.020 0.025 0.030 0.035 0.040 0.045 0.050 0.055 0.060
```

2.5 PV system cost

- Data taken from "Capital Costs for Dual-Use Photovoltaic Installations: 2020 Benchmark" Table 1 and Figure 3.
- This data was used to estimate CAPEX.
- avtyps = agrivoltaic types.
- item = itemized component of system.
- cost = cost of each item.
- height = ground to panel clearance height (ft.)
- tcost = Total cost is the sum of all itemized cost for AV system. See figure 3 and table 1 in above document for more detail.

```
pvsc <- wb_read(file = "Data/Parameters.xlsx",
                sheet = "PV system Cost (NREL)",
                rows = c(1:109),
                cols = c(1:5),
                col_names = TRUE) %>%
  rename(avtyps = `AV Types`,
         item = Item,
         cost = `Cost ($/W)`,
         height = `Panel Height (ft.)`,
         tcost = `Total Cost ($/W)`
```

```
        )
str(pvsc)
```

```
'data.frame':   108 obs. of  5 variables:
 $ avtyps: chr  "Typical Fixed PV" "Typical Fixed PV" "Typical Fixed PV" "Typical Fixed PV"
 $ item  : chr  "EPC/Developer Net Profit" "Developer Overhead" "Contingency(3%)" "Interconne
 $ cost  : num  0.11 0.15 0.05 0.03 0.02 0.05 0.12 0.18 0.24 0.11 ...
 $ height: num  4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 ...
 $ tcost : num  1.53 1.53 1.53 1.53 1.53 1.53 1.53 1.53 1.53 1.53 ...
```

2.6 Capex and Plot

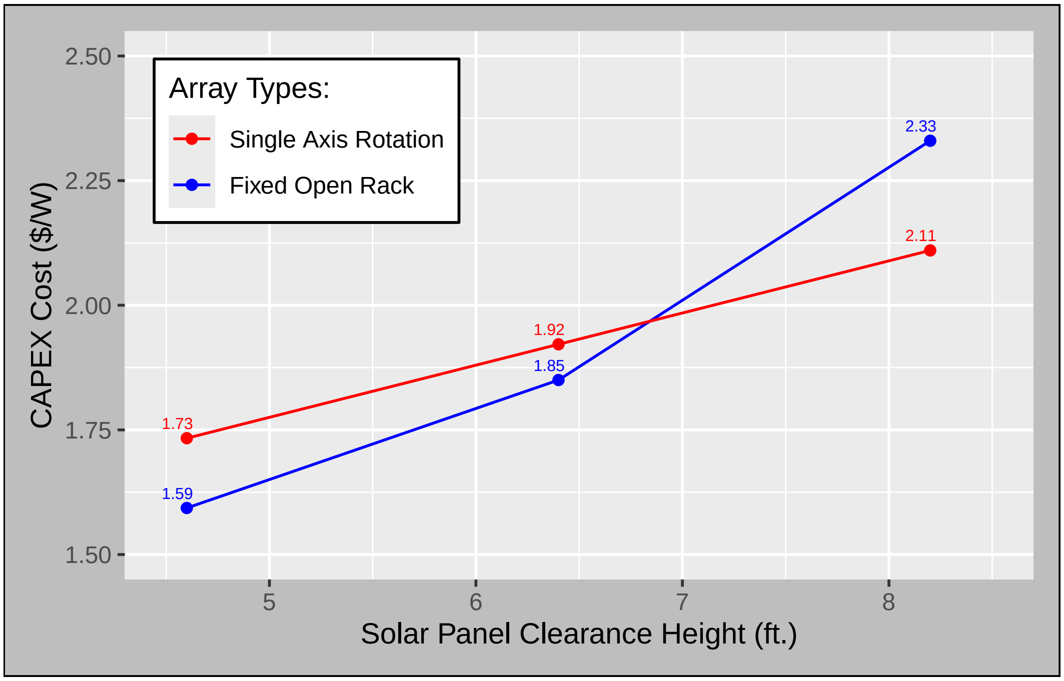
Variable Descriptions:

- Capex: Capital investment cost (\$/W) to develop solar energy system. Capex includes cost of physical structure, developer's overhead and EPC/Developer's net profit.
- capex estimated as f(height, tracker) using OLS for 6.4 ft Tracking system.
- Height = ground to panel clearance in ft.
- array: Solar array. Tracker = Single axis sun tracking panels; Fixed = Non-tracking panels.
- Source: Horowitz, 2020. CAPEX AV.

```
capex <- read.table(file = "Data/CAPEX.txt",
                    header = TRUE,
                  sep = "\t") %>%
  rename(capex = cost,
         height = pheight,
         array = tracker)
```

```
capex %>%
  ggplot(aes(
    x = height,
    y = capex,
    color = array,
    group = array
  )) +
  geom_point() +
  geom_line() +
  # Display the rounded capex values
```

```
geom_text(aes(label = sprintf("%.2f", capex)),
          vjust = -0.8,
          hjust = 0.8,
          size = 2,
          check_overlap = TRUE,
          show.legend = FALSE
          ) +
labs(
  #title = "CAPEX Cost by Solar Panel Height",
  x = "Solar Panel Clearance Height (ft.)",
  y = "CAPEX Cost ($/W)",
  color = "Array Types:"
  ) +
scale_x_continuous(limits = c(4.5, 8.5)) +
scale_y_continuous(limits = c(1.5, 2.5)) +
guides(color = guide_legend(reverse = TRUE)) +
theme(
  plot.background = element_rect(
    fill = "grey",
    color = "black"
    ),
  legend.position = "inside",
  legend.position.inside = c(0.2, 0.8),
  legend.background = element_rect(
    fill = "white",
    color = "black"
    ),
  plot.margin = margin(10, 10, 10, 10)
) +
scale_color_manual(
  values = c("Fixed" = "blue",
             "Tracking" = "red"),
  labels = c("Fixed Open Rack",
             "Single Axis Rotation")
)
```




```r
# Save the plot
ggsave(
  filename = "Plots/CAPEX Solar Panels.png",
  width = 8,
  height = 6,
  units = "in"
  )
```

2.7 Panel Configuration

- Panel configuration and DV system output (W).

```r
panconf <- wb_read(file = "Data/Parameters.xlsx",
                sheet = "PV Spacing",
                start_row = 2,
                start_col = 1,
                skip_empty_rows = TRUE,
                skip_empty_cols = TRUE,
                col_names = TRUE)
str(panconf)
```

```
'data.frame':   21 obs. of  21 variables:
 $ Total Area (Acre)                 : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Total Area (Sq. Ft.)              : num  43560 43560 43560 43560 43560 ...
 $ Solar Proportion                  : num  1 0.95 0.9 0.85 0.8 0.75 0.7 0.65 0.6 0.55 ...
 $ Solar Proportion Area (Sq. Ft.): num  43560 41382 39204 37026 34848 ...
 $ Solar Proportion Area (Sq.M.)  : num  4047 3845 3642 3440 3237 ...
 $ Side Length (ft.)                 : num  209 209 209 209 209 ...
 $ YSide Length (ft.)                : num  209 209 209 209 209 ...
 $ XSide length (ft.)                : num  209 198 188 177 167 ...
 $ Panel Length (ft.)                : num  7.75 7.75 7.75 7.75 7.75 7.75 7.75 7.75 7.75 7.75 .
 $ Row Seperator (ft.)               : num  6 6 6 6 6 6 6 6 6 6 ...
 $ Panel Width(ft.)                  : num  3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 3.5 ...
 $ Panel Area (Sq. ft.)              : num  27.1 27.1 27.1 27.1 27.1 ...
 $ Panels/Row                        : num  59 59 59 59 59 59 59 59 59 59 ...
 $ Total Rows                        : num  15 14 13 12 12 11 10 9 9 8 ...
 $ Total Panels                      : num  885 826 767 708 708 649 590 531 531 472 ...
 $ Array Area (Sq. Ft.)              : num  24006 22405 20805 19205 19205 ...
 $ Array Area (Sq. M.)               : num  2230 2082 1933 1784 1784 ...
 $ XSide Open Length (ft)            : num  92 100 107 115 115 123 131 138 138 146 ...
 $ Inter Panel Spacing (ft)          : num  6 7 8 10 10 12 14 17 17 20 ...
 $ Panel Efficienfy                  : num  0.19 0.19 0.19 0.19 0.19 0.19 0.19 0.19 0.19 0.19 .
 $ DC System Size (kW)               : num  424 395 367 339 339 ...
```

2.8 Energy output

Energy output was simulated using NREL PV Watts Calculator.

- sprop = land proportion covered by solar in 1 acres. Value ranges from 0 to 1.
- Panels = Total number of panels in 1 acres of land.
- datalot: 1 = first simulation done for four regions of AL; 2 = second simulation done for four regions of AL. Two simulations have two unique zipcodes for each simulated region.
- al_regs = regions of Alabama
- zips = zipcodes selected from each region of AL for simulation.
- array = Fixed (open rack); 1AxisRot = 1 Axis Tracking. See above NREL tool for more detail.
- dc_kw = DC system size, calculated for each solar panel heights considering solar panels efficiency and area covered by solar panels.

- energy = total energy output (kWh/Year) considering system parameters. Total hours considered by the model is 8,760 (See PV Watts Calculator Results > help (below the result) > results > download monthly or hourly results).

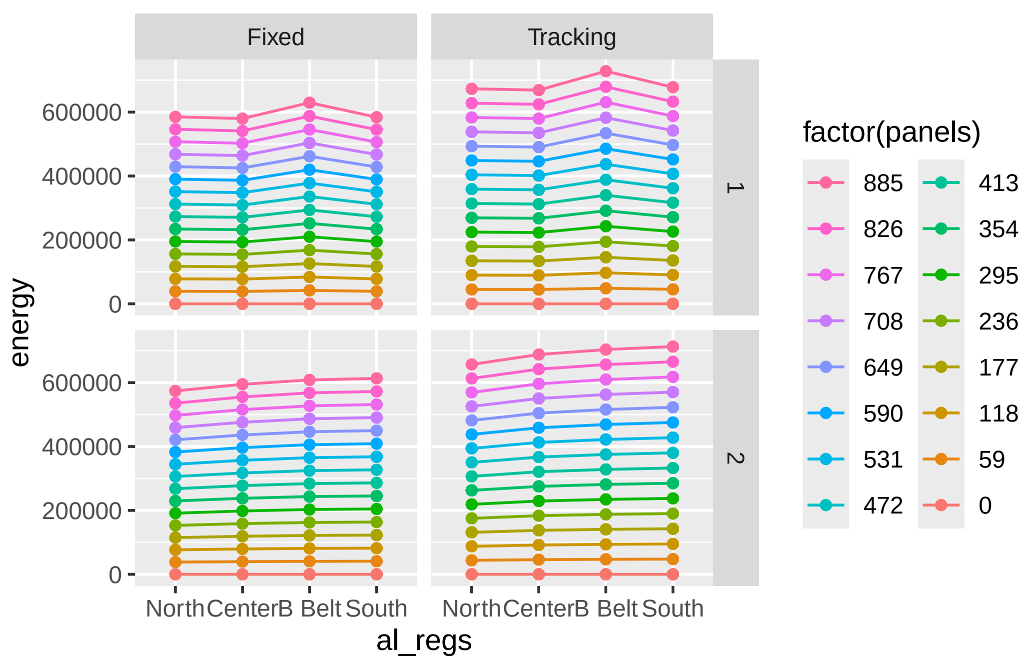
```
energy_output <- read_xlsx("Data/Parameters.xlsx",
                           sheet = "Energy Output",
                           start_row = 1,
                           start_col = 1,
                           skip_empty_rows = TRUE,
                           skip_empty_cols = TRUE,
                           col_names = TRUE) %>%
  rename(sprop = `Solar Proportion`,
         panels = `Total Panels`,
         datalot = DataLot,
         al_regs = `Region of AL`,
         zips = ZIPCODE,
         array = `Array Type`,
         dc_kw = `DC System Size (kW)`,
         energy = `Energy (kWh/Year)`) %>%
  mutate(
    dc_kw = round(dc_kw,2),
    array = case_when(
      array == "1AxisRot" ~ "Tracking",
      array == "FixedOpen" ~ "Fixed",
      TRUE ~ array)
    )
str(energy_output)
```

```
'data.frame':   336 obs. of  8 variables:
 $ sprop  : num  1 1 1 1 1 1 1 1 0.95 0.95 ...
 $ panels : num  885 885 885 885 885 885 885 885 826 826 ...
 $ datalot: num  1 1 1 1 1 1 1 1 1 1 ...
 $ al_regs: chr  "Northern" "Northern" "Central" "Central" ...
 $ zips   : num  35801 35801 35223 35223 36117 ...
 $ array  : chr  "Tracking" "Fixed" "Tracking" "Fixed" ...
 $ dc_kw  : num  424 424 424 424 424 ...
 $ energy : num  672887 585225 668895 579758 728181 ...
```

2.8.1 By # of panels

Plotting Energy output by number of solar panels in one acres of AV system from fixed and single axis rotation system for two zipcodes (1, 2) within each of the four regions of AL.

```r
lox <- c("Northern", "Central", "Black Belt", "Southern")
array_levs = c("Single Axis Rotation", "Fixed Open Rack")
datalot_levs = c("Location 1", "Location 2")
ggplot(data = energy_output,
         mapping = aes(x = al_regs,
                       y = energy,
                       #fill = energy,
                       color = factor(panels),
                       group = factor(panels))) +
  geom_line()+
  geom_point() +
  facet_grid(datalot~array) +
  scale_x_discrete(limits = lox,
                   labels = c("North", "Center",
                              "B Belt", "South")) +
  guides(color = guide_legend(ncol = 2,
                              reverse = TRUE))
```

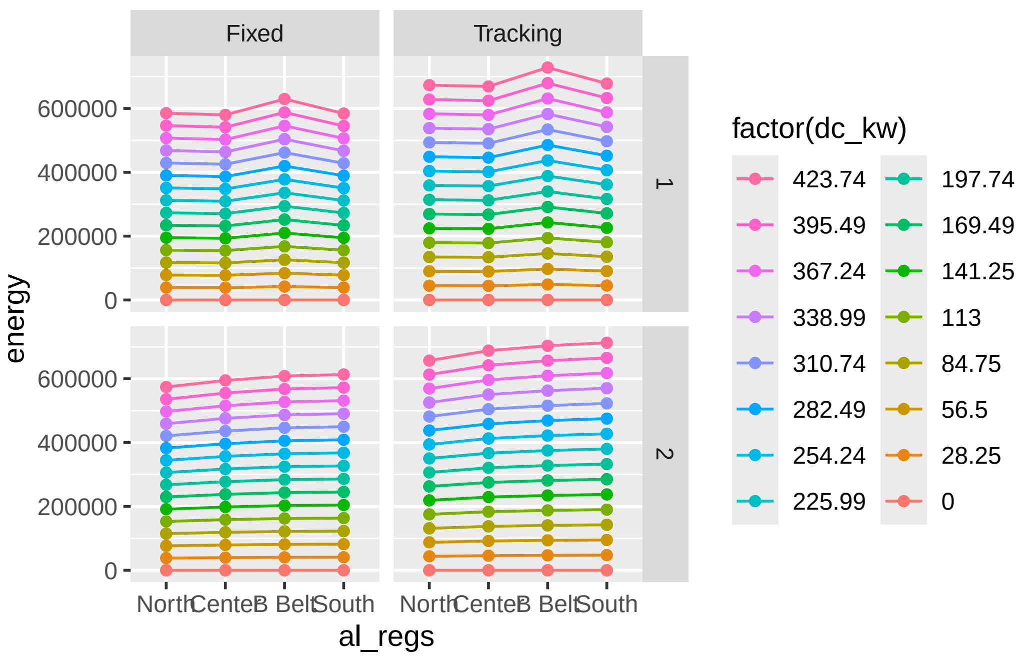



```r
rm(lox); rm(array_levs); rm(datalot_levs)
```

2.8.2 By DC System Size

Plotting Energy output by DC System Size from fixed and single axis rotation system for two zipcodes (1, 2) within each of the four regions of AL.

```
lox <- c("Northern", "Central", "Black Belt", "Southern")
ggplot(data = energy_output,
         mapping = aes(x = al_regs,
                       y = energy,
                       #fill = energy,
                       color = factor(dc_kw),
                       group = factor(dc_kw))) +
  geom_line()+
  geom_point() +
  facet_grid(datalot~array) +
  scale_x_discrete(limits = lox,
                   labels = c("North", "Center",
                              "B Belt", "South")) +
  guides(color = guide_legend(ncol = 2,
                              reverse = TRUE))
```



```r
rm(lox)
```

3 Solar Energy Calculation

3.1 Simulation: Energy Revenue

- elcprc = electricity price. See Electricity price data for more detail.
- elcrev = Revenue from electricity for given electricity prices. See "energy output" and "electricity price" dataset for more details.
- I filtered datalot 2–I did not take average of "energy" from datalot 1 and datalot 2–to minimize computation time.

```r
# Convert to data frames if they are not already
matrix1 <- energy_output  %>%
  group_by(sprop, al_regs, array, dc_kw, panels) %>%
  dplyr::filter(datalot == 2) %>%
  # Compute mean of datalot 1 and datalot 2:
  summarise(
    energy = mean(energy),
    .groups = 'drop'
    ) # dimension of matrix is 168*6
matrix2 <- elec_price # dimension of matrix is 11*1

# Initialize the result data frame
# energy_revenue <- data.frame(matrix(nrow = 1848, ncol = 9))
energy_revenue <- data.frame(
  matrix(nrow = nrow(matrix2)*nrow(matrix1),
         ncol = ncol(matrix2)+ncol(matrix1)+1))

# Variable to keep track of the row index in the result matrix
row_index <- 1

# Loop through each value of the second matrix
for (i in 1:nrow(matrix2)) {
  # Loop through each value of the second matrix
  for (j in 1:nrow(matrix1)) {
    # First matrix, second matrix, combined two matrices.
    new_row <- c(matrix1[j, ],
                 matrix2[i, ],
```

```
                 matrix1$energy[j] * matrix2$epr_kwh[i])
    # Assign the new row to the result matrix
    energy_revenue[row_index, ] <- new_row
    # Increment the row index
    row_index <- row_index + 1
  }
}
# Name the columns
colnames(energy_revenue) <- c(colnames(matrix1), "elcprc", "elcrev")
# Check for any NAs in the result
if(any(is.na(energy_revenue))) {
  na_indices <- which(is.na(energy_revenue), arr.ind = TRUE)
  print(paste("NAs found at rows:", unique(na_indices[, 1])))
} else {
  print("No NAs found in the result data frame.")
}
```

```
[1] "No NAs found in the result data frame."
```

```
str(energy_revenue)
```

```
'data.frame':   1848 obs. of  8 variables:
 $ sprop  : num  0 0 0 0 0 0 0 0 0.05 0.05 ...
 $ al_regs: chr  "Black Belt" "Black Belt" "Central" "Central" ...
 $ array  : chr  "Fixed" "Tracking" "Fixed" "Tracking" ...
 $ dc_kw  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ panels : num  0 0 0 0 0 0 0 0 0 0 ...
 $ energy : num  0 0 0 0 0 0 0 0 0 0 ...
 $ elcprc : num  0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
 $ elcrev : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
rm(matrix1, matrix2, new_row, i, j, row_index)
```

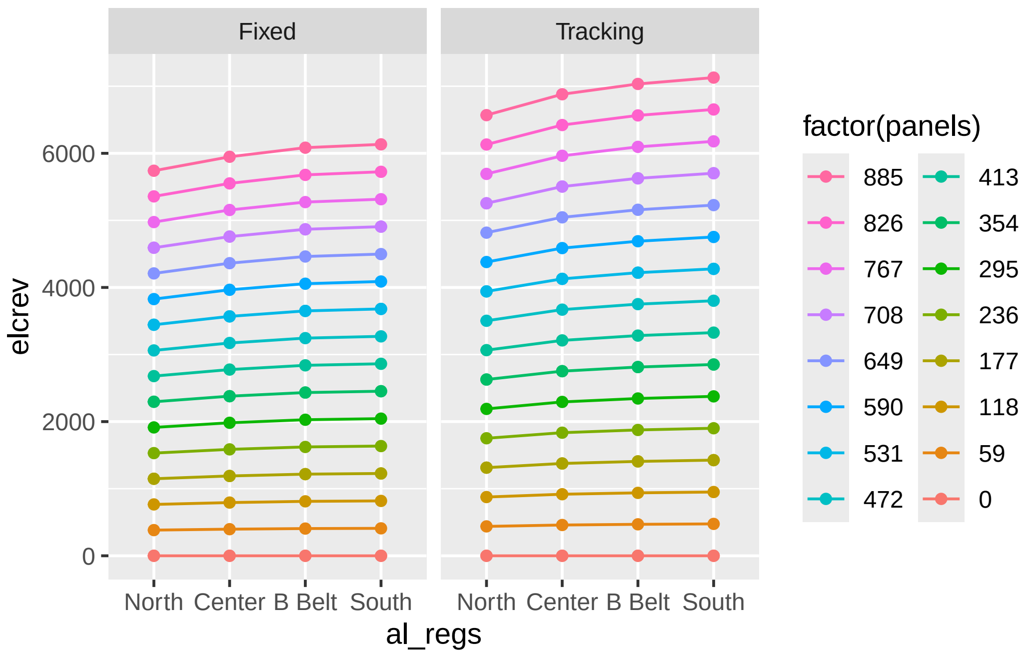
3.2 Plot Revenue from Energy

3.2.1 By # of panels

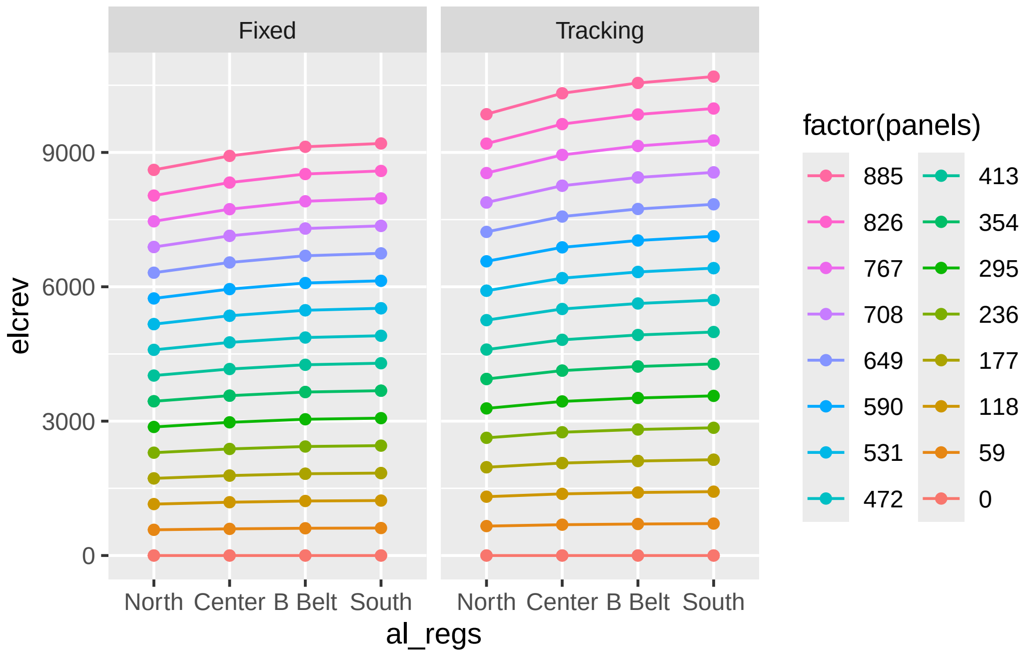
I am using data from simulation 1 for this visualization. This code plots one chart per electricity cost. There are 11 electricity cost resulting into 11 charts. Electricity revenue is average revenue of first and second lots of simulation.

```r
lox <- c("Northern", "Central", "Black Belt", "Southern")
array_levs = c("Single Axis Rotation", "Fixed Open Rack")
datalot_levs = c("Location 1", "Location 2")
for (i in unique(energy_revenue$elcprc)) {
 a = ggplot(data = (energy_revenue %>%
  dplyr::filter(elcprc == i)),
         mapping = aes(x =al_regs,
                       y = elcrev,
                       #fill = energy,
                       color = factor(panels),
                       group = factor(panels)))+
  geom_line()+
  geom_point()+
  facet_grid(.~array) +
  scale_x_discrete(limits = lox,
                   labels = c("North", "Center", "B Belt", "South")) +
   guides(color = guide_legend(ncol = 2, reverse = TRUE))
 cat("Electricity Price = ", i)
 print(a)
}
```

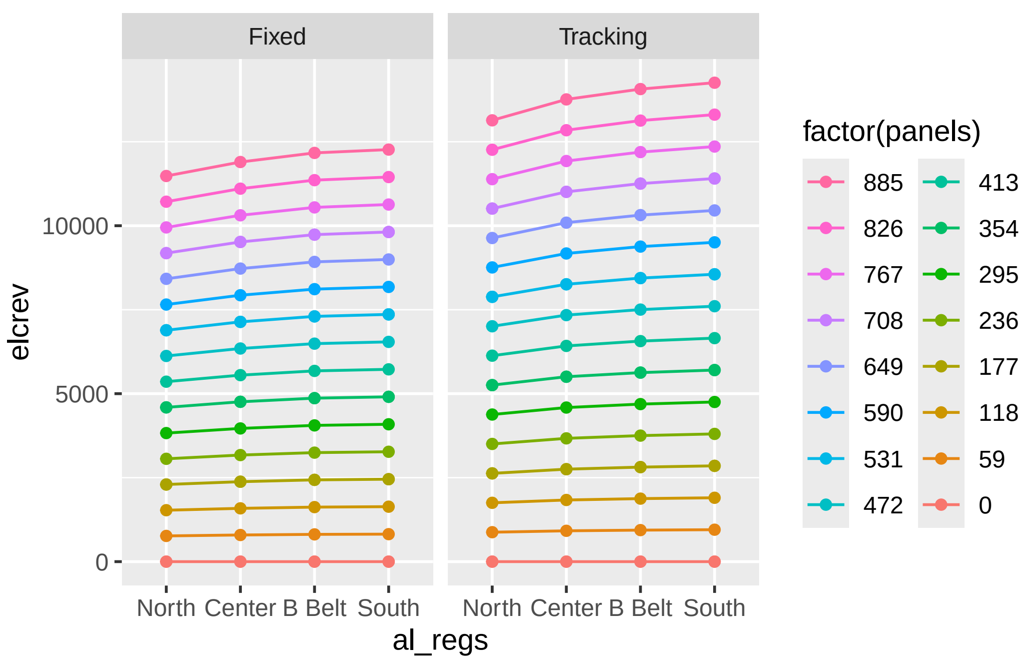
```
Electricity Price =  0.01
```



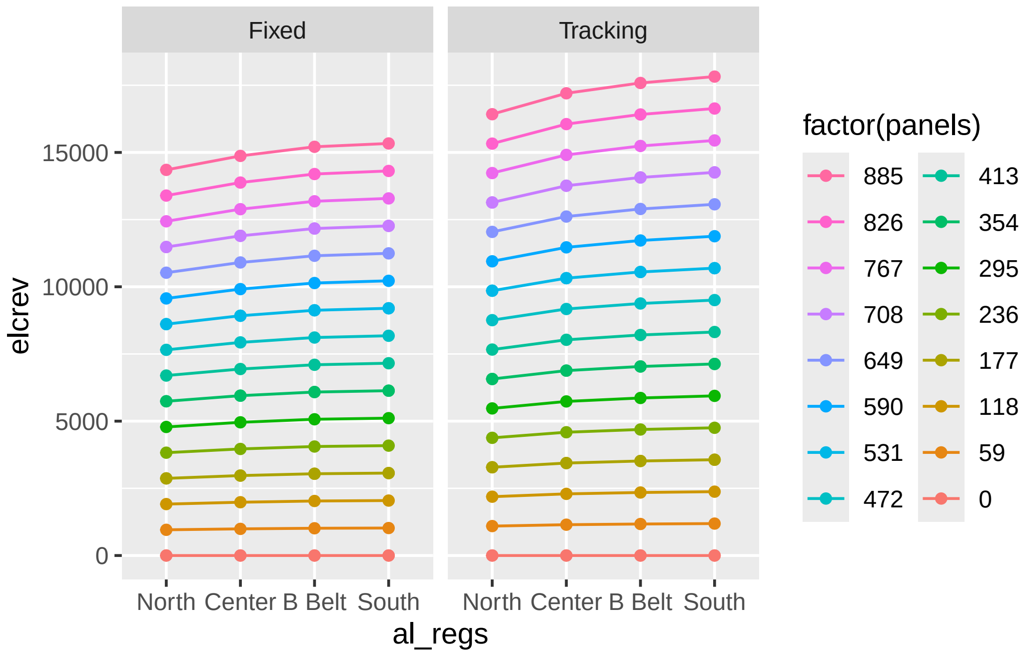
```
Electricity Price =  0.015
```



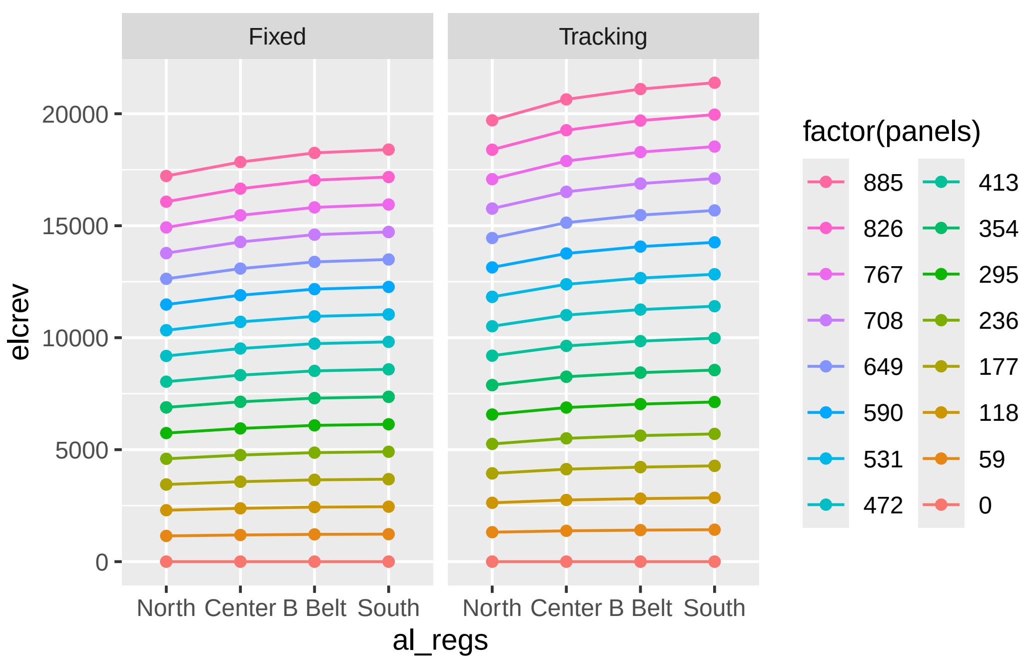

```
Electricity Price =  0.02
```



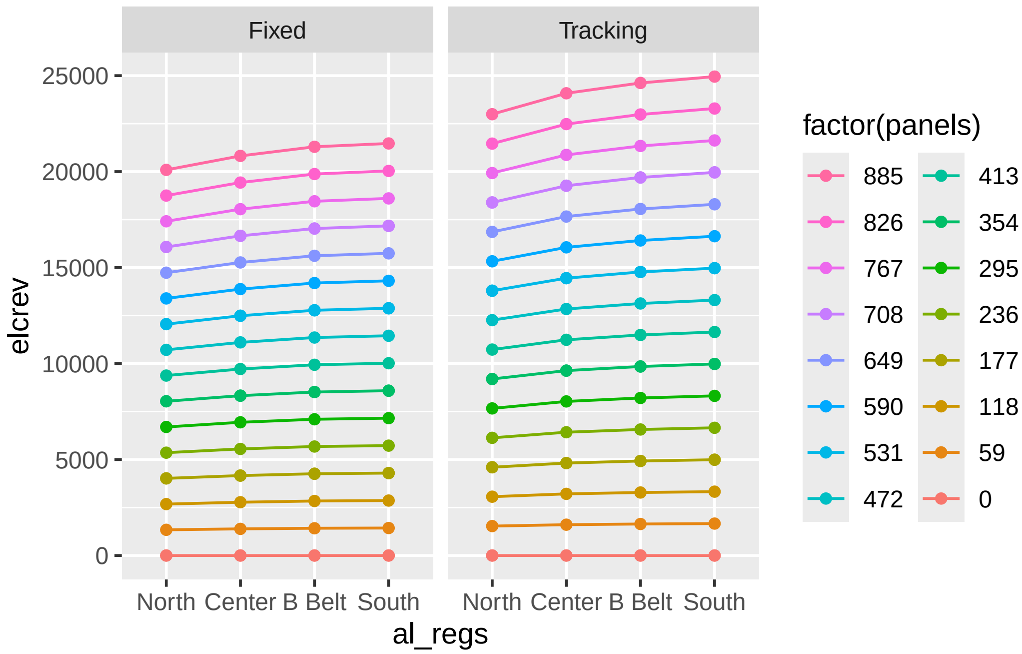
```
Electricity Price =  0.025
```



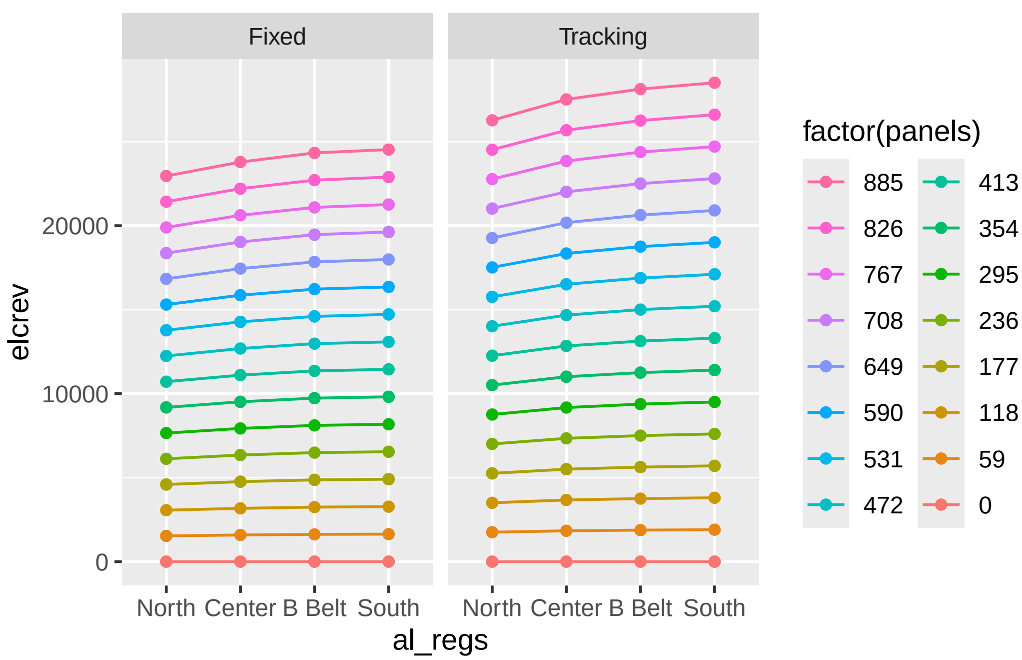

```
Electricity Price =  0.03
```



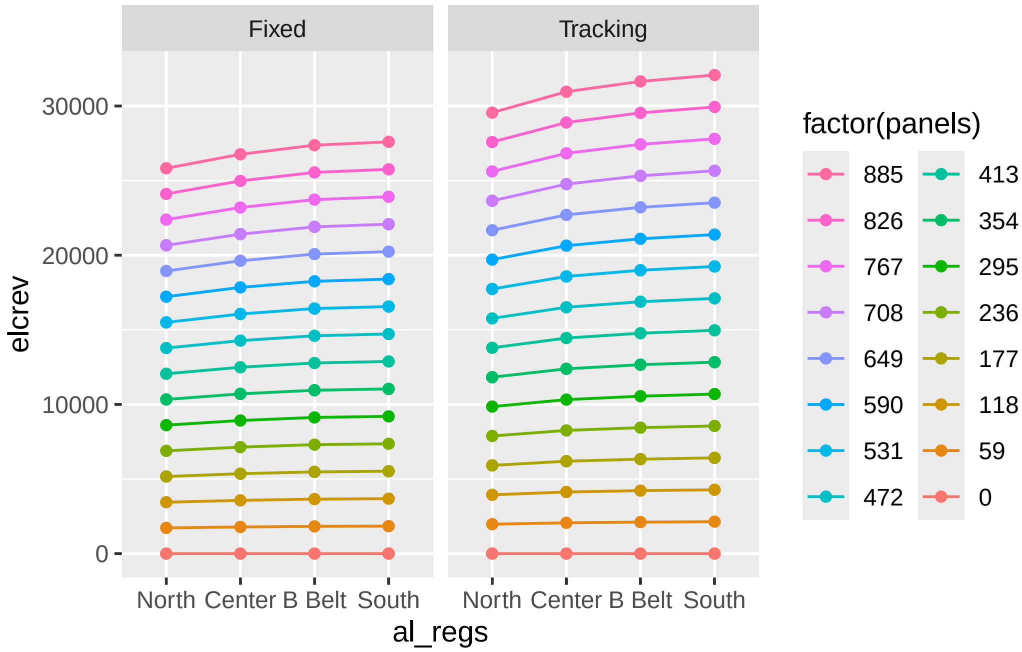
Electricity Price = 0.035



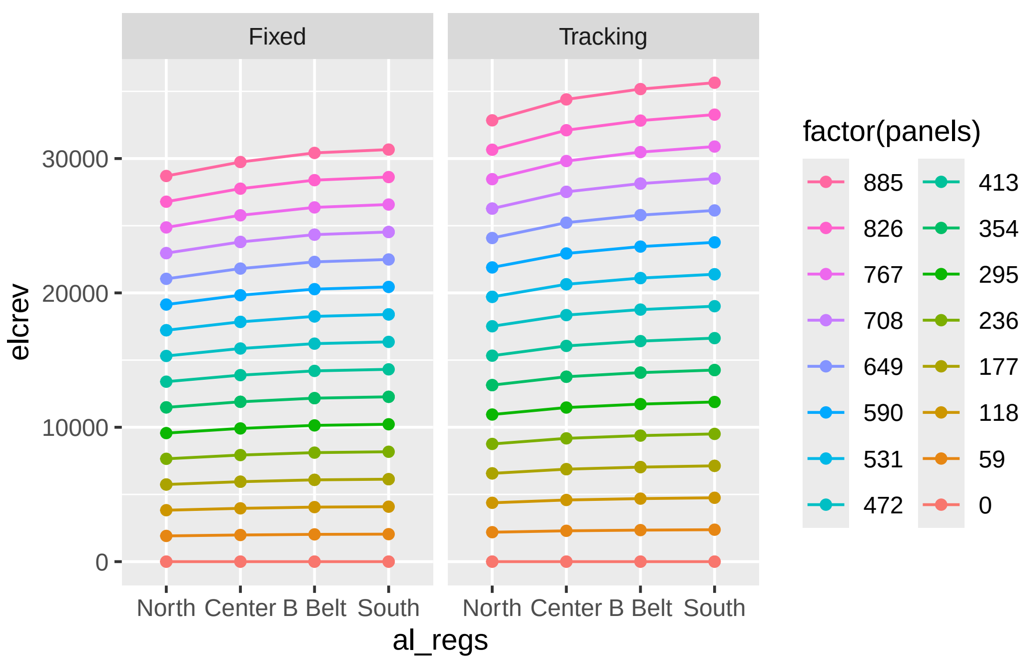
Electricity Price = 0.04



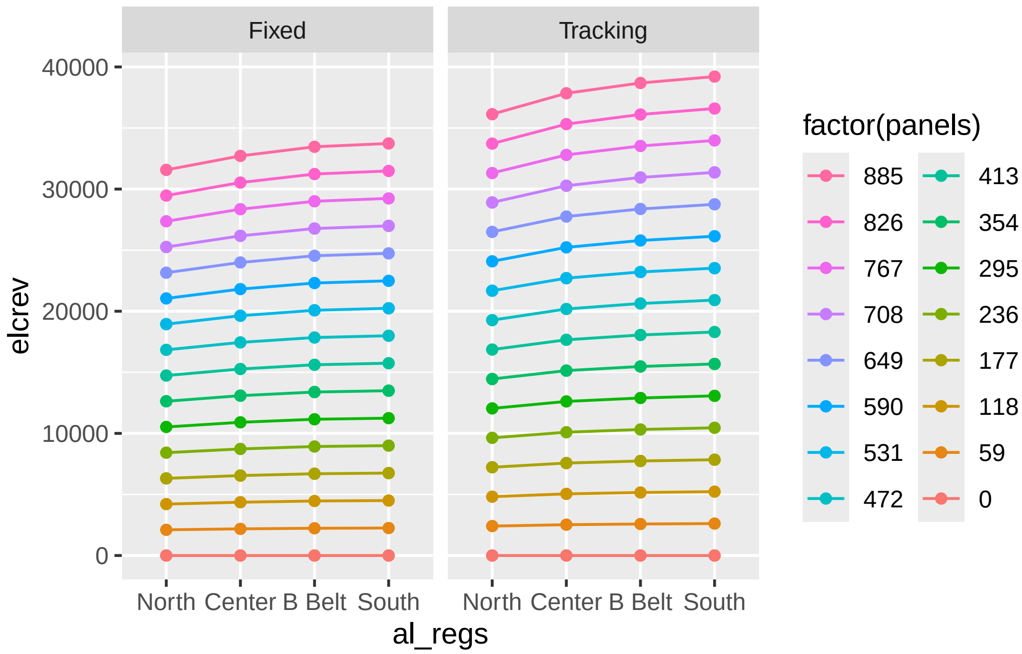
```
Electricity Price =  0.045
```



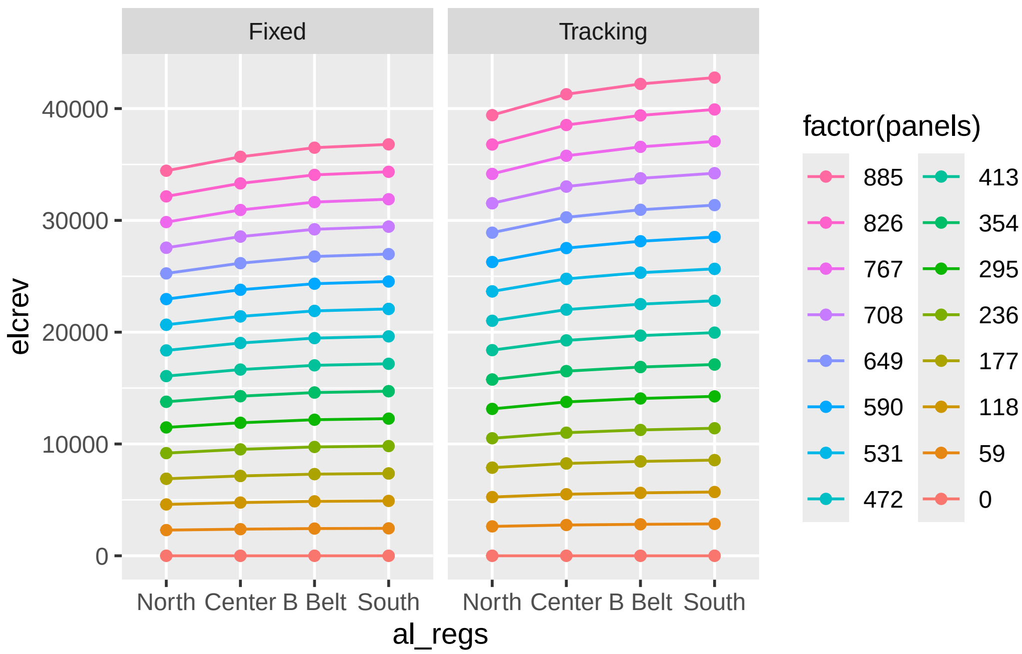

```
Electricity Price =  0.05
```



```
Electricity Price =  0.055
```




```
Electricity Price =  0.06
```



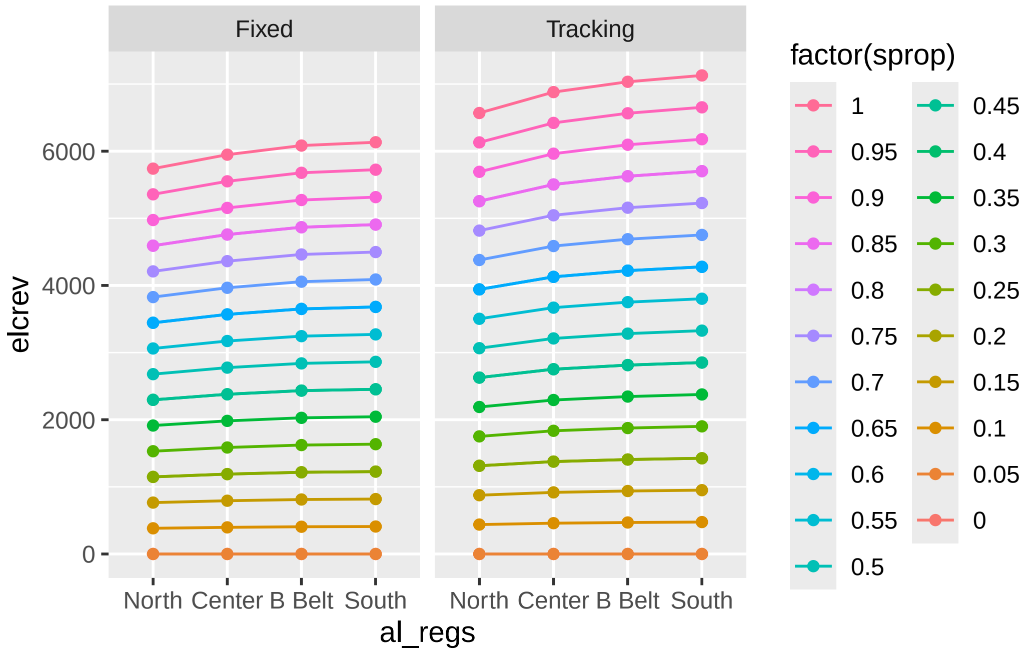
```r
rm(array_levs, datalot_levs, i, lox, a)
```

3.2.2 By Land in Solar

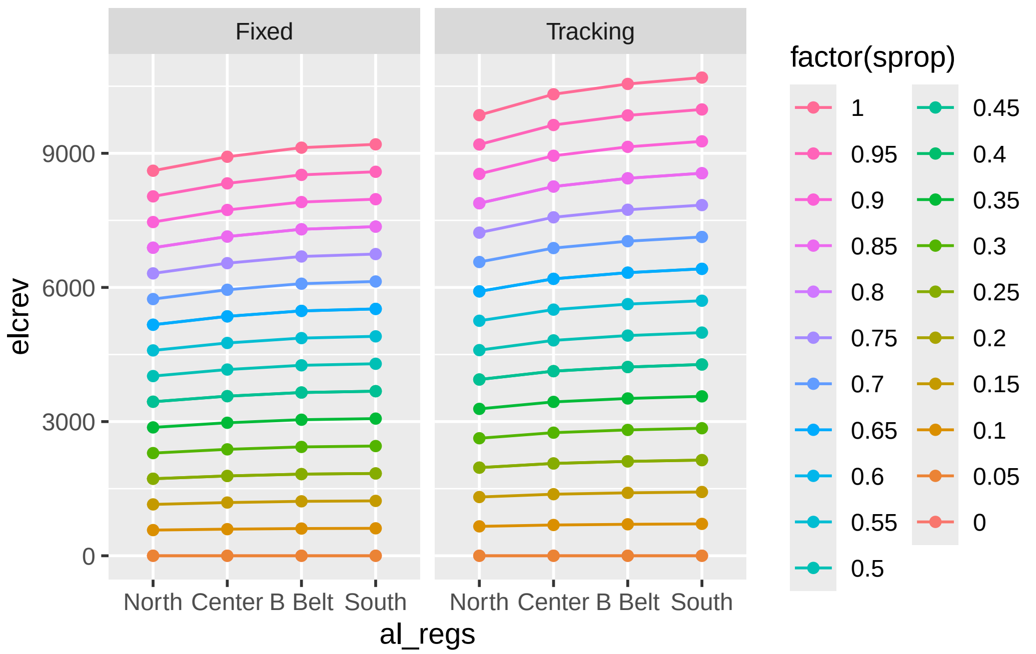
- Two proportions may have same number of solar panels (Eg. 0.80 and 0.85, 0.20 and 0.25). So, total lines in the chart may not match with total number of legend levels. Some proportions are overlapping in the chart. See panel configuration for more detail.

```r
lox <- c("Northern", "Central", "Black Belt", "Southern")
array_levs = c("Single Axis Rotation", "Fixed Open Rack")
datalot_levs = c("Location 1", "Location 2")
for (i in unique(energy_revenue$elcprc)) {
 a = ggplot(data = (energy_revenue %>%
  dplyr::filter(elcprc == i)),
         mapping = aes(x =al_regs,
                       y = elcrev,
                       #fill = energy,
                       color = factor(sprop),
                       group = factor(sprop)))+
  geom_line()+
  geom_point()+
  facet_grid(.~array) +
  scale_x_discrete(limits = lox,
                   labels = c("North", "Center", "B Belt", "South")) +
   guides(color = guide_legend(ncol = 2, reverse = TRUE))
 cat("Electricity Price = ", i)
 print(a)
}
```

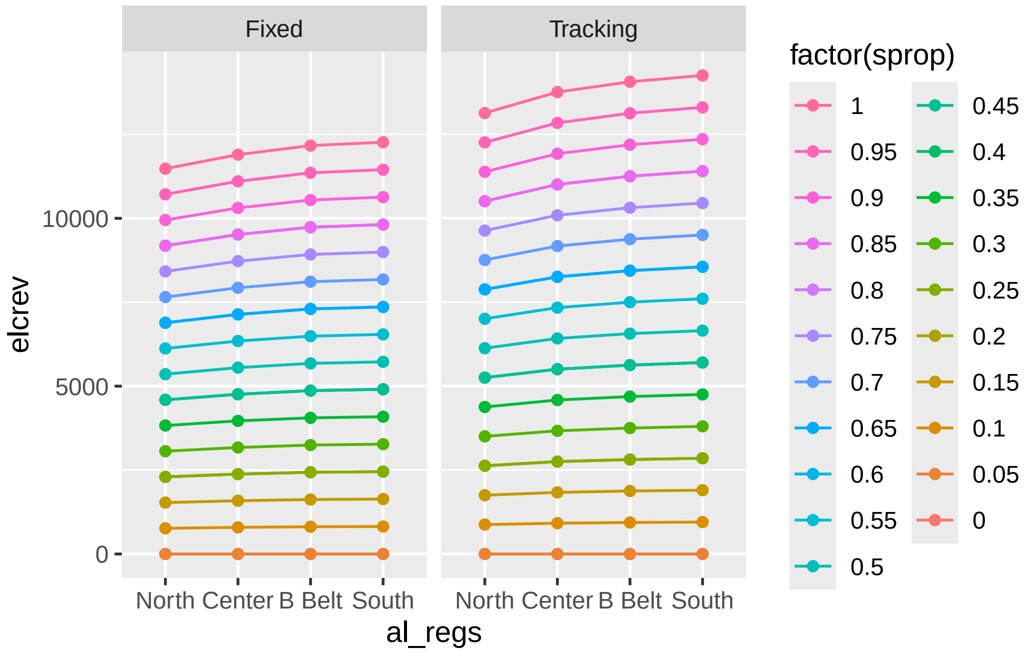
```
Electricity Price =  0.01
```



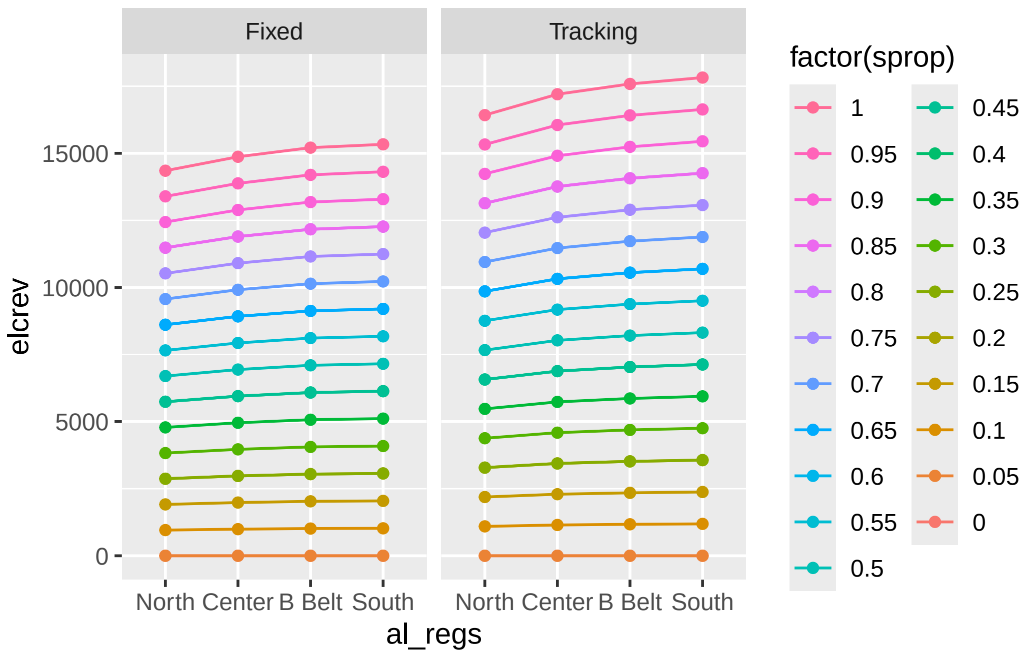

```
Electricity Price =  0.015
```



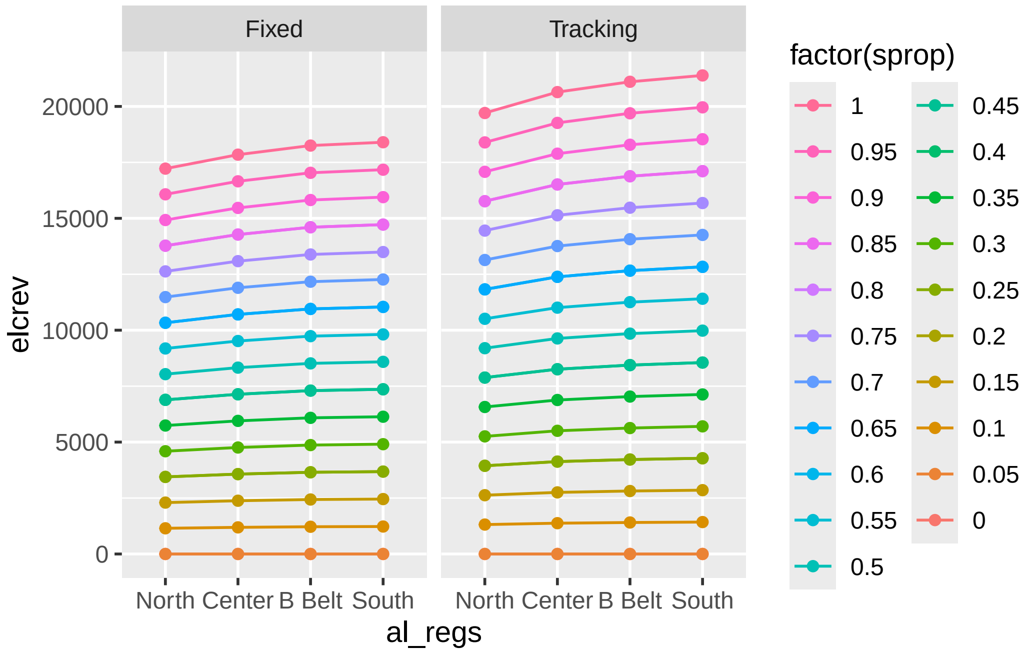

```
Electricity Price =  0.02
```



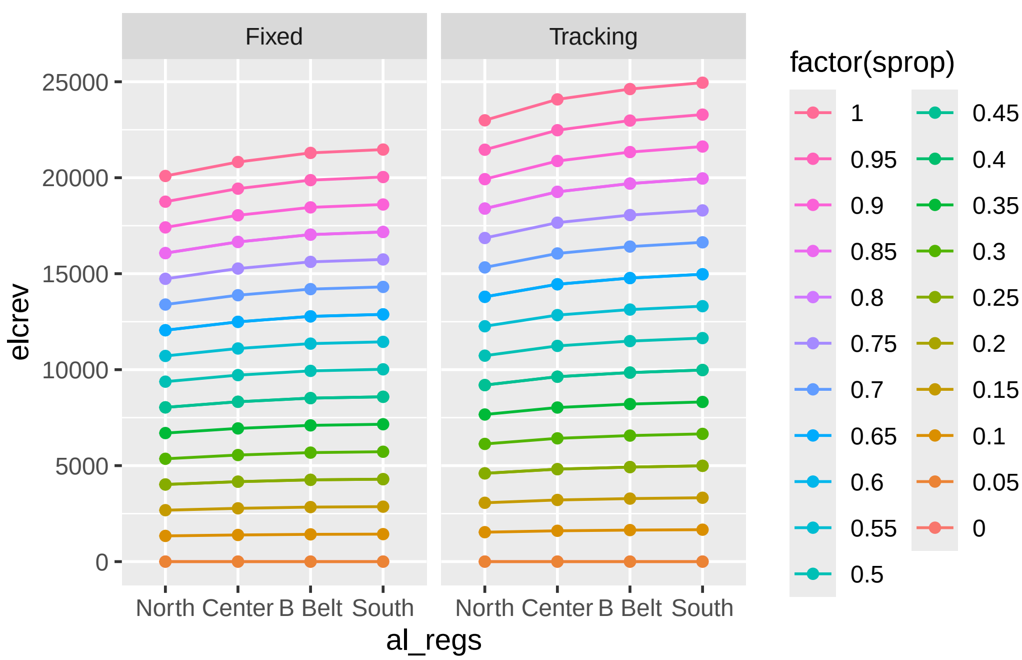

```
Electricity Price =  0.025
```



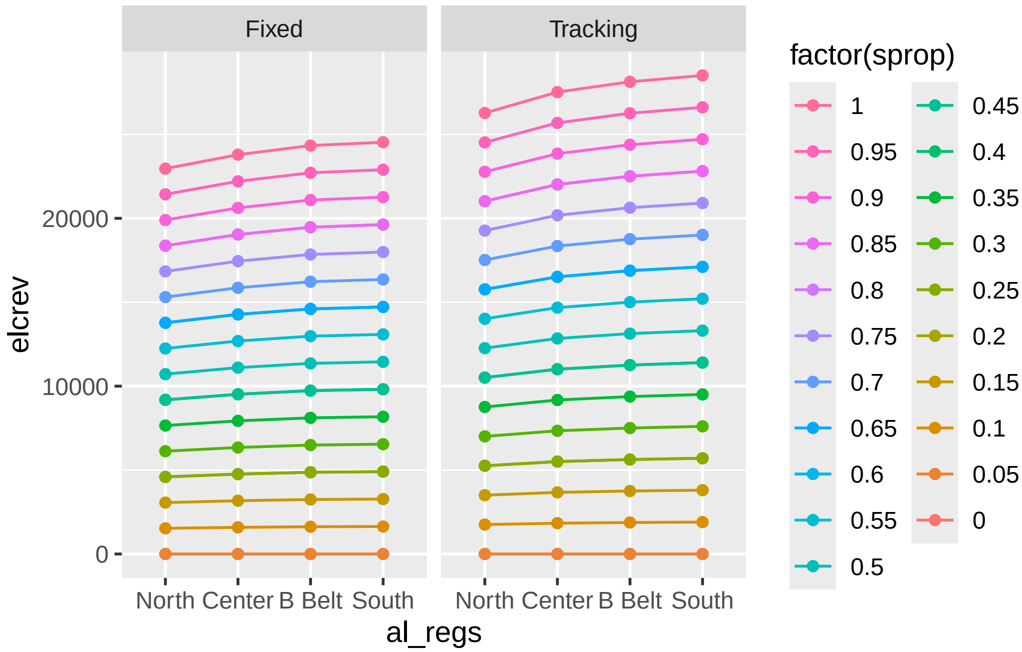

```
Electricity Price =  0.03
```



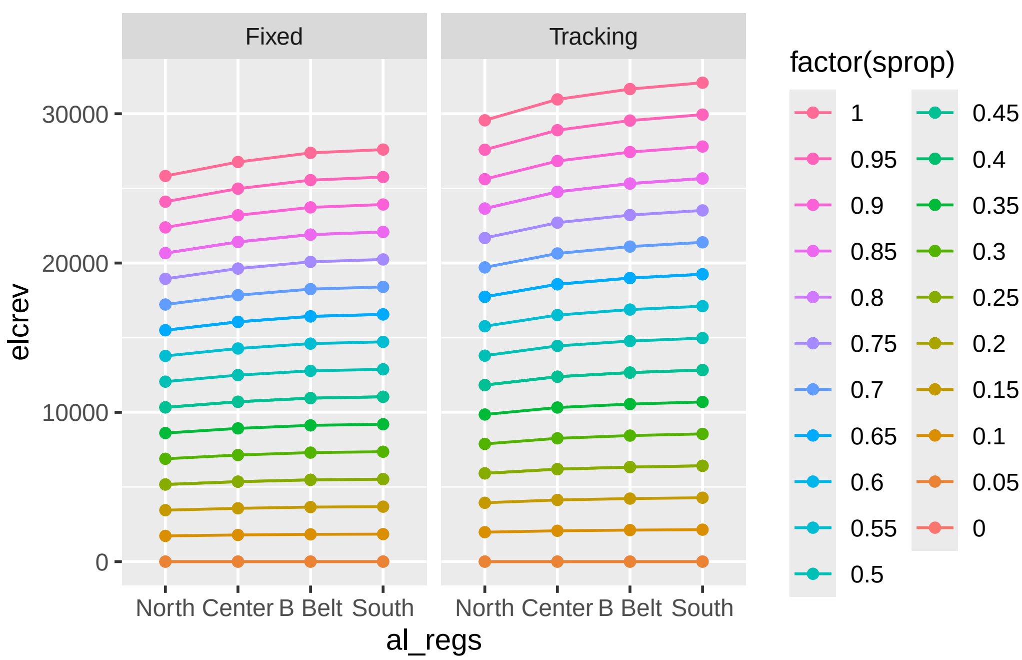

```
Electricity Price =  0.035
```



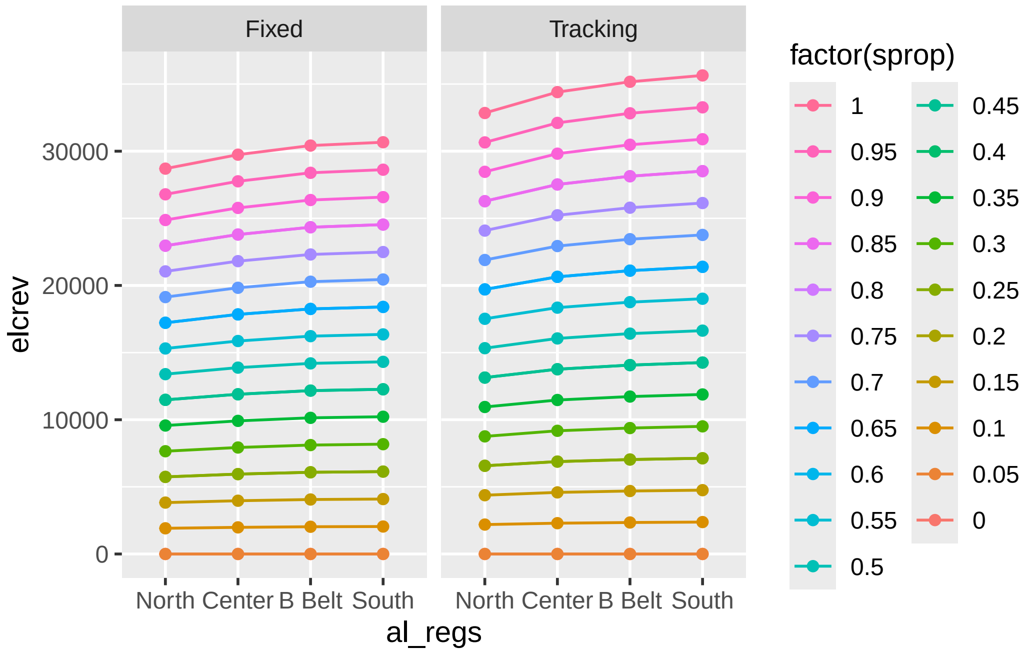

```
Electricity Price =  0.04
```



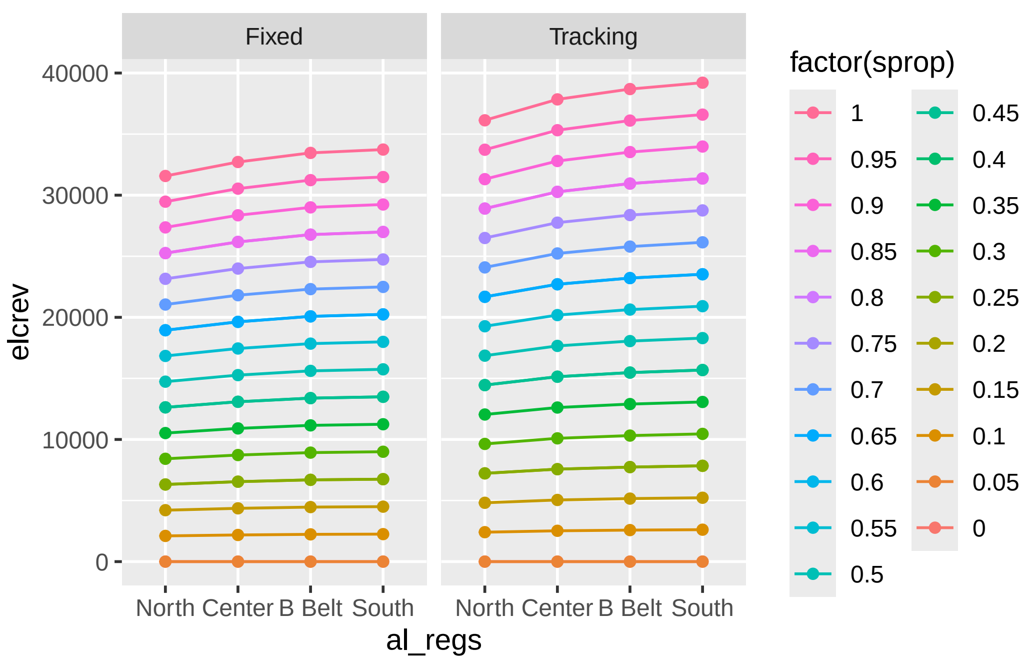

```
Electricity Price =  0.045
```



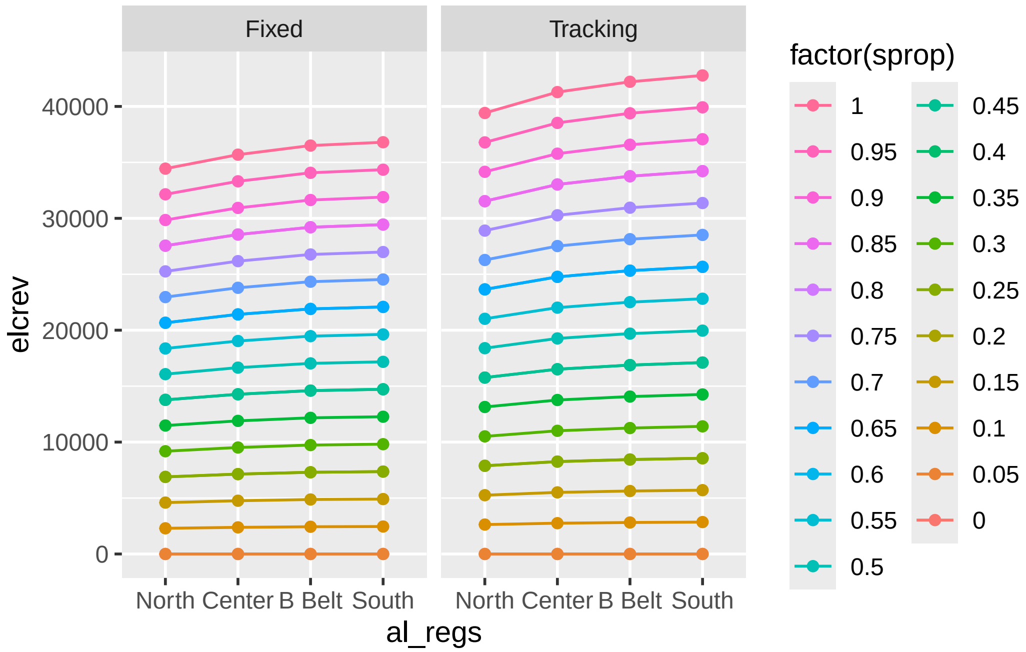

```
Electricity Price =  0.05
```




```
Electricity Price =  0.055
```




```
Electricity Price =  0.06
```




```
rm(lox, array_levs, datalot_levs, i, a)
```

3.3 Cost and Profit from solar

- Cost of solar energy system in agrivoltaic setting.
- I used energy output per 7.75 ft.*3.5 ft. panel (545 w), capex ($/w), and total number of panels to get total cost for each height and panel tracking system.
- height = height of solar panels; see capex dataset for details.
- capex = capex from capex table; see capex dataset for details.
- opex = Operational cost ($15/kW/Year) Source: Ramasamy, 2022. PV Cost Benchmark (This is revised to 3% of annual capex based on Dennis Brother's suggestion).
- ttlcost = Total cost for given DC system size.
- anncost = Annual payment to repay loan (P_{ann}) = $\frac{P_o(i(1+i)^t)}{(1+i)^t-1)}$, where P_o = CAPEX loan burrowed to repay in t years; $t = 25$, and i = annual interest rate at 5%.
- moncost = Monthly payment to repay loan (P_{mon}) = $\frac{P_o((i/12)(1+(i/12))^{t*12})}{(1+(i/12))^{t*12}-1)}$, where P_o = CAPEX loan burrowed to repay in t years; $t = 25$, and i = annual interest rate at 5%.
- inscst = insurance cost. $5 per $1000 capex.

- eprofit = profit from electricity after subtracting total cost (ttlcost) from total revenue (elcrev).
- eannprof = annual profit from solar after subtracting annual loan repayment distributed over 25 years.
- emonprof = monthly profit from solar after subtracting monthly loan repayment distributed over 25 years.
- eannprofworeap = annual profit without REAP benefit.
- eannprofwoincentives = Annual profit without incentives.

Policy Components:

- taxcr = 30% tax credit of annual cost covered through federal tax exemption (Investment tax credit).
- reap = Rural Energy for America Program reimburses 50% of capex (ttlcost) upfront. The waiting time for reimbursement is about 6 months. So, 50% of ttlcost acquire simple interest for six months. This is changed to 25% and 50%.
- recredit = renewable energy credit ($6.60/MWh).

```
i = 0.07 # Discount/interest Rate
n = 25 # Life Span of solar panels (Years)
reapprop = 50/100  # Percentage of CAPEX covered by REAP program.

expanded_data <- energy_revenue %>%
  slice(rep(1:n(),
            each = 3))
capex_height <- rep(unique(capex$height),
                    length.out = nrow(energy_revenue))

energy_cost = cbind(expanded_data, capex_height) %>%
  rename(height = capex_height)

energy_cost <- left_join(energy_cost,
                         capex,
                         by = c("array", "height")) %>%
  mutate(
    # 7.75*3.5 sq.ft. panel energy output = 545 W.
    # Operational cost (OPEX) = $15/kW-yr; 1 kW = 1,000W.
    # Opex = 545*15/1000*panels,

    # Land lease cost Per acre.
```

```
    landlease = 1000,

    # Total Capex
    ttlcost = capex*545*panels,

    # Cost of Insurance = $5/$1000/Yr Total capex
    inscst = ttlcost*5/1000, #Cost

    # Renewable energy credit 6.60 $/MWh
    recredit = 6.60/1000*energy, #Return

    # REAP Program = 50% of Capex - Simple interest rmbrst delay
    reap = reapprop*ttlcost - (reapprop*ttlcost)*i*0.5/100, #Return

    # Annualized cost - reap:
    annlzcost = (ttlcost - reap + inscst)*(i*(1+i)^n)/((1+i)^n-1),

    # Annualized Cost of total cost:
    annoftotcost = ttlcost*(i*(1+i)^n)/((1+i)^n-1),

    # Monthalized using monthly discount rate:
    monthlycost = ttlcost*
      ((i/12)*(1+(i/n))^(n*12))/((1+(i/12))^(n*12)-1),

    # Operational cost = 3% of annualized total capex
    opex = 3*annoftotcost/100, #Cost

    # Tax credit = 30% of annualized capex
    taxcr = 30*annoftotcost/100, #Return

    # Annualized using annual discount rate:
    anncost = annlzcost + opex
    )

solar_profit <- energy_cost %>%
  mutate(
    # Annualized Profit
    eannprof = elcrev + recredit + taxcr - anncost,

    eannprofworeap = elcrev + recredit + taxcr - annoftotcost,
    eannprofwoincentives = elcrev - annoftotcost
    )
```

```r
write_xlsx(file = "Results/Solar Profit R50.xlsx",
           x = solar_profit,
           overwrite = TRUE,
           as_table = TRUE)
str(solar_profit)
```

```
'data.frame':   5544 obs. of  24 variables:
 $ sprop              : num  0 0 0 0 0 0 0 0 0 0 ...
 $ al_regs            : chr  "Black Belt" "Black Belt" "Black Belt" "Black Belt" ...
 $ array              : chr  "Fixed" "Fixed" "Fixed" "Tracking" ...
 $ dc_kw              : num  0 0 0 0 0 0 0 0 0 0 ...
 $ panels             : num  0 0 0 0 0 0 0 0 0 0 ...
 $ energy             : num  0 0 0 0 0 0 0 0 0 0 ...
 $ elcprc             : num  0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
 $ elcrev             : num  0 0 0 0 0 0 0 0 0 0 ...
 $ height             : num  4.6 6.4 8.2 4.6 6.4 8.2 4.6 6.4 8.2 4.6 ...
 $ capex              : num  1.59 1.85 2.33 1.73 1.92 ...
 $ landlease          : num  1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 ...
 $ ttlcost            : num  0 0 0 0 0 0 0 0 0 0 ...
 $ inscst             : num  0 0 0 0 0 0 0 0 0 0 ...
 $ recredit           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ reap               : num  0 0 0 0 0 0 0 0 0 0 ...
 $ annlzcost          : num  0 0 0 0 0 0 0 0 0 0 ...
 $ annoftotcost       : num  0 0 0 0 0 0 0 0 0 0 ...
 $ monthlycost        : num  0 0 0 0 0 0 0 0 0 0 ...
 $ opex               : num  0 0 0 0 0 0 0 0 0 0 ...
 $ taxcr              : num  0 0 0 0 0 0 0 0 0 0 ...
 $ anncost            : num  0 0 0 0 0 0 0 0 0 0 ...
 $ eannprof           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ eannprofworeap     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ eannprofwoincentives: num  0 0 0 0 0 0 0 0 0 0 ...
```

```r
rm(capex_height, i, n, reapprop, expanded_data, energy_cost)
```

3.4 Profit from Solar

Profit from solar alone at 100% PVD

```
pf_solar <- solar_profit %>%
  filter(sprop == 1,
         elcprc == 0.04) %>%
  select(al_regs, array, height, eannprof, eannprofworeap)
cat("Maximum profit from solar at 100% PVD at 50% REAP = ",
    max(pf_solar$eannprof),
    fill = TRUE)
```

```
Maximum profit from solar at 100% PVD at 50% REAP =  16348.39
```

```
pf_solar[which.max(pf_solar$eannprof),]
```

```
    al_regs    array height eannprof eannprofworeap
22 Southern Tracking    4.6 16348.39      -16998.23
```

```
cat("Minimum profit from solar at 100% PVD at 50% REAP = ",
    min(pf_solar$eannprof),
    fill = TRUE)
```

```
Minimum profit from solar at 100% PVD at 50% REAP =  4070.151
```

```
pf_solar[which.min(pf_solar$eannprof),]
```

```
    al_regs array height eannprof eannprofworeap
15 Northern Fixed    8.2 4070.151      -40755.41
```

```
rm(pf_solar)
```

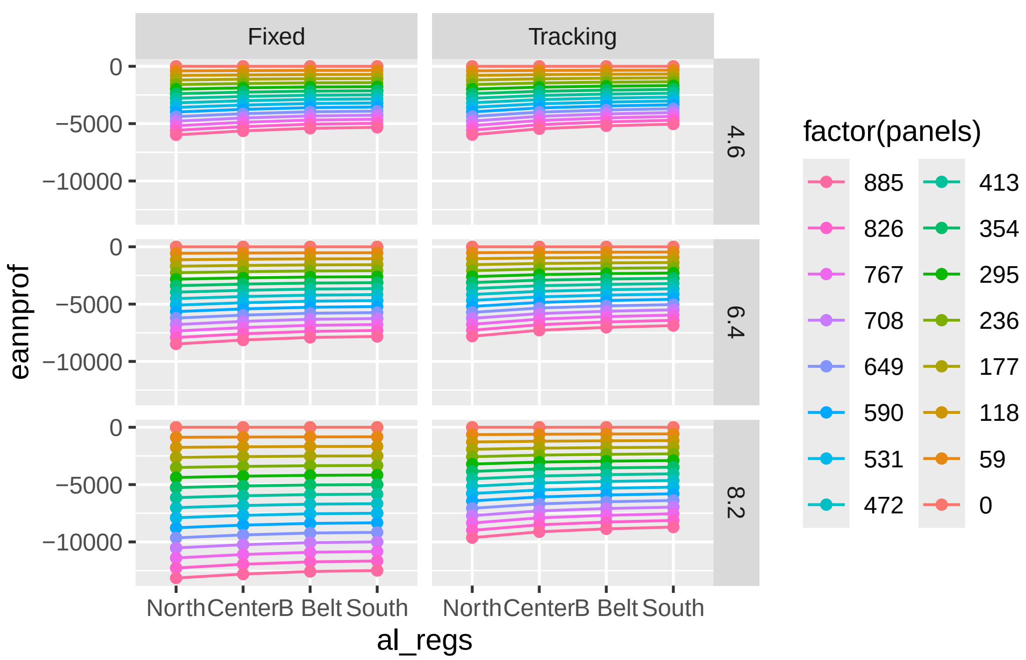
3.4.1 Plot Solar profit

Solar annual profit by number of solar panels

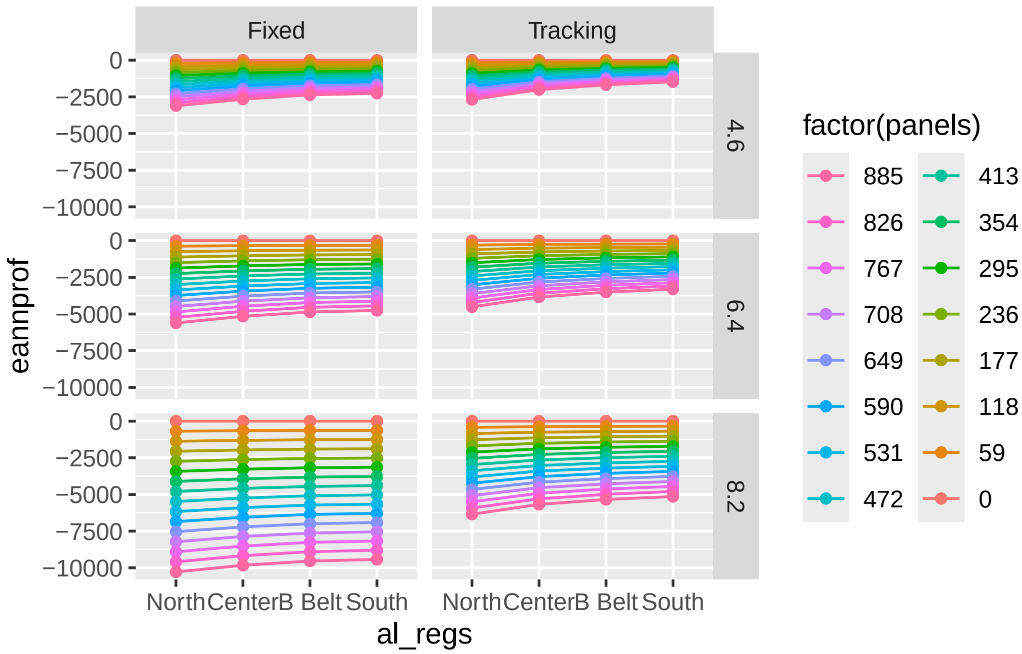
```
lox <- c("Northern", "Central", "Black Belt", "Southern")
array_levs = c("Single Axis Rotation", "Fixed Open Rack")
datalot_levs = c("Location 1", "Location 2")
  for (i in unique(solar_profit$elcprc)) {
    b = ggplot(
      data = (solar_profit %>%
```

```
                dplyr::filter(elcprc == i)),
      mapping = aes(
        x = al_regs,
        y = eannprof, #Annual Profit
        #fill = energy,
        color = factor(panels),
        group = factor(panels)
      )
    ) +
      geom_line() +
      geom_point() +
      facet_grid(height ~ array) +
      scale_x_discrete(limits = lox,
                       labels = c("North", "Center",
                                  "B Belt", "South")) +
      guides(color = guide_legend(ncol = 2,
                                  reverse = TRUE))
    cat("Electricity Price = ", i)
    print(b)
  }
```

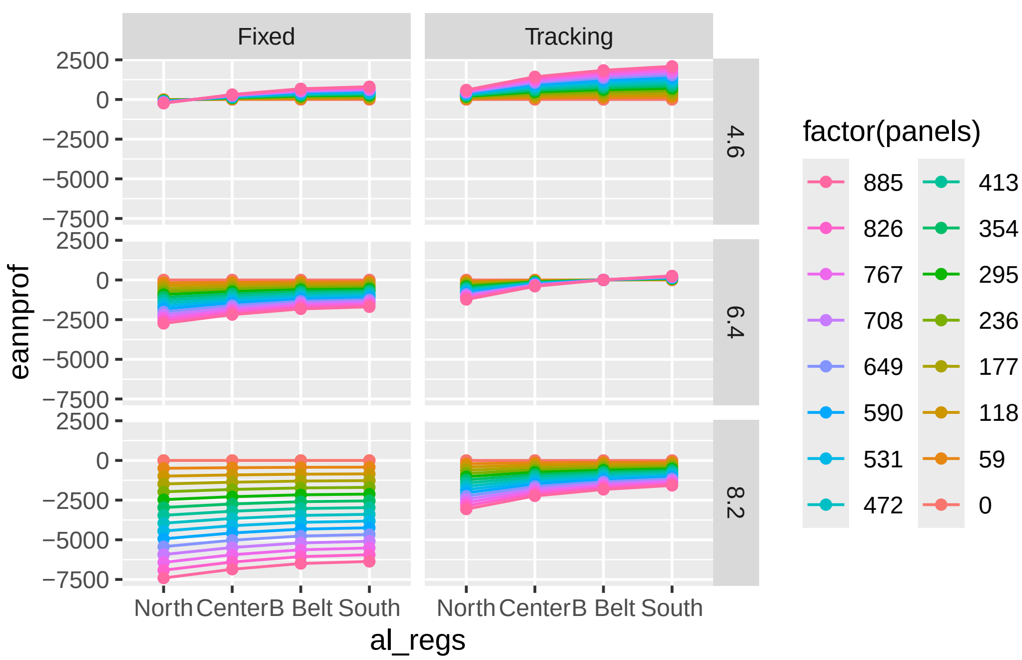
```
Electricity Price =  0.01
```



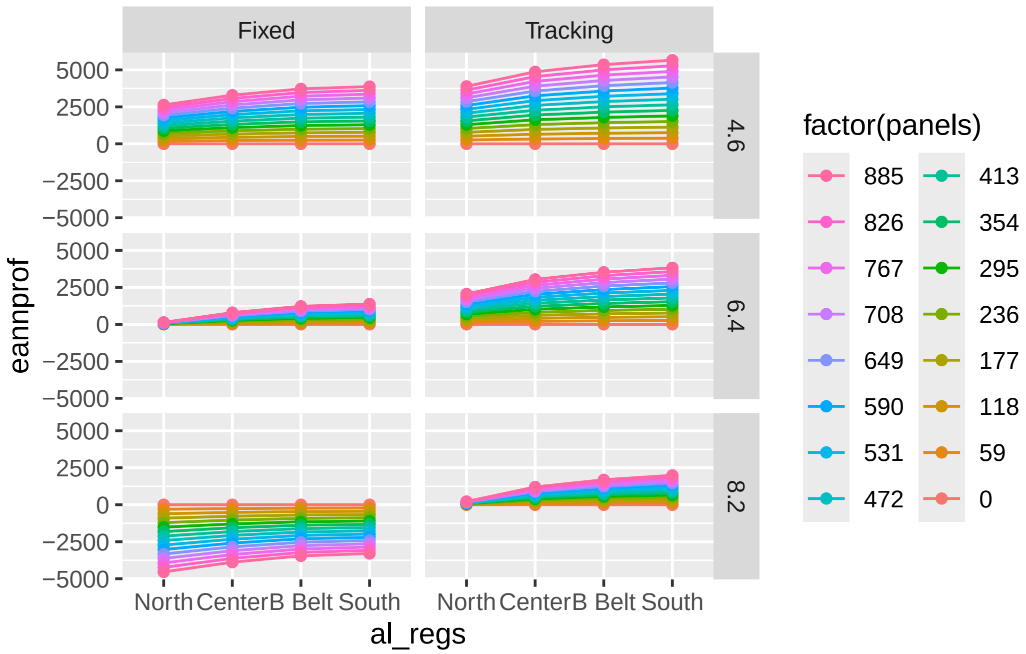
```
Electricity Price =  0.015
```



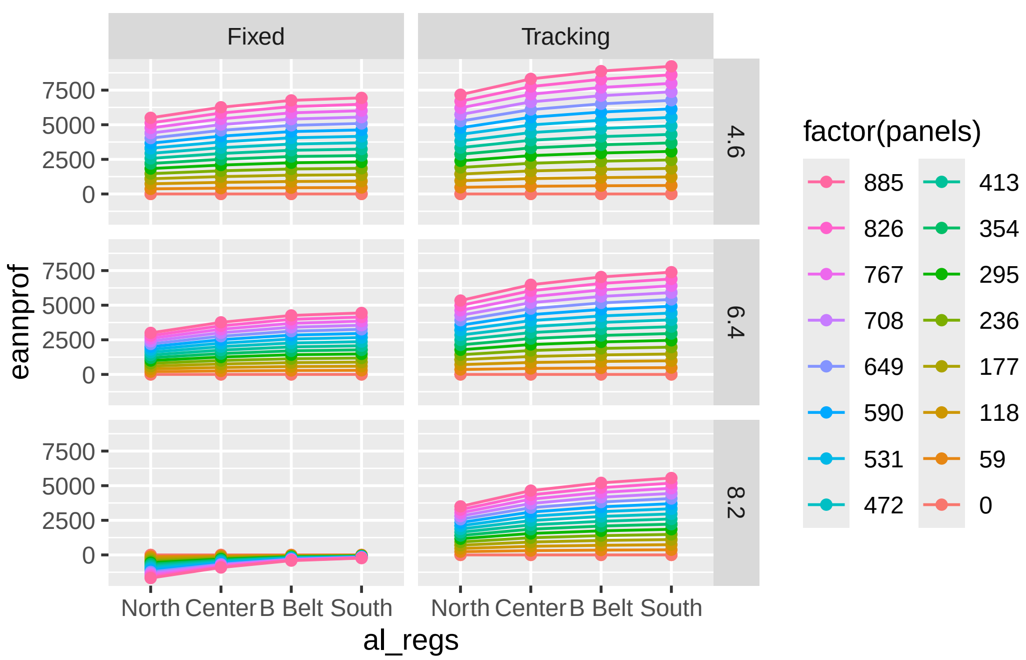

```
Electricity Price =  0.02
```



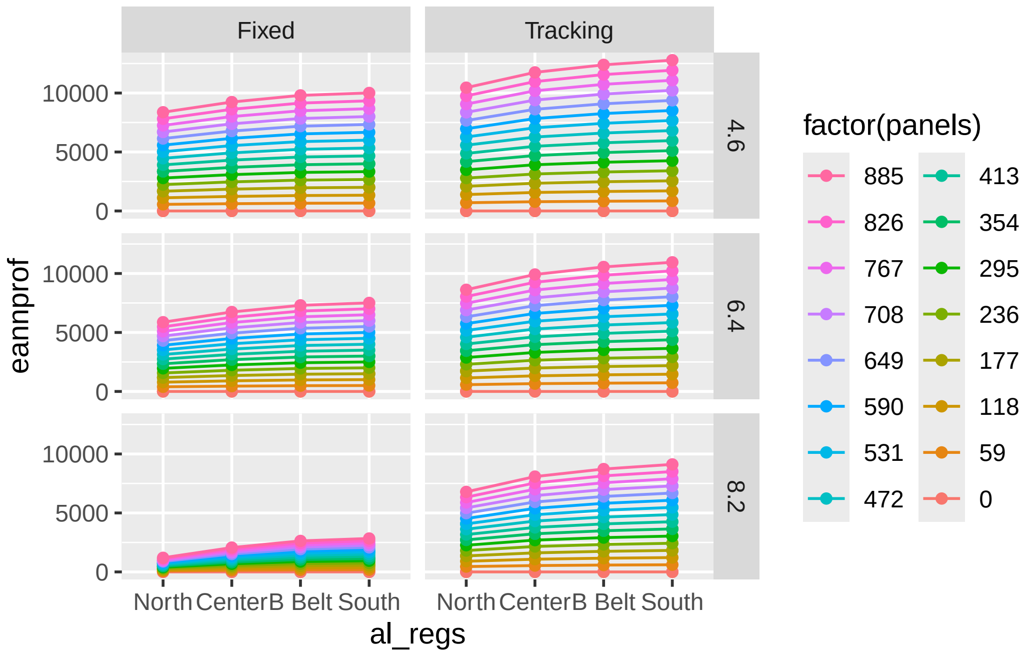
Electricity Price = 0.025



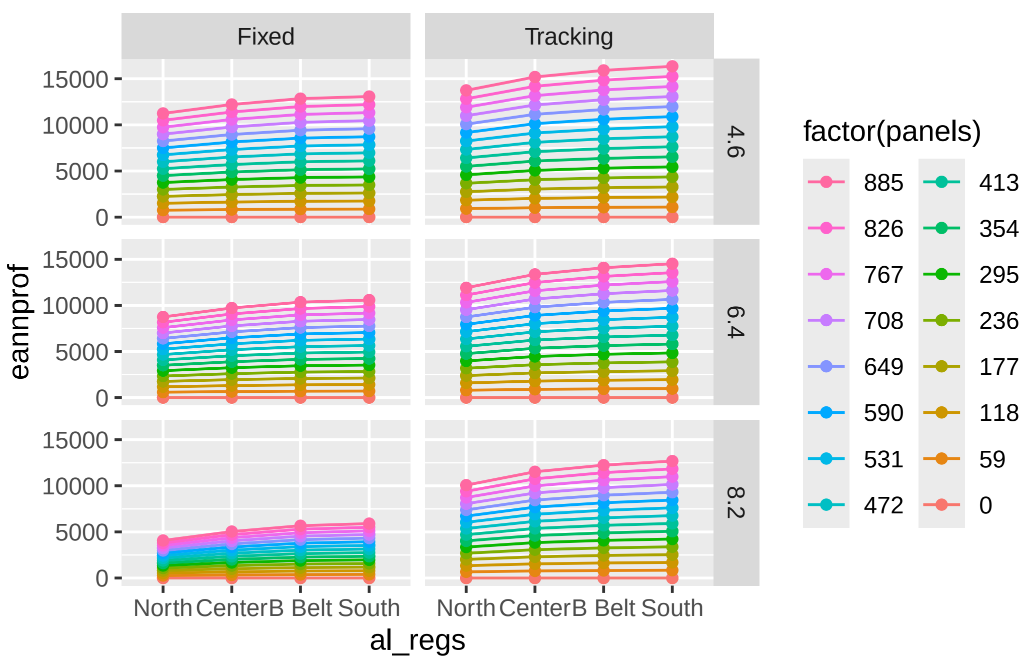
Electricity Price = 0.03



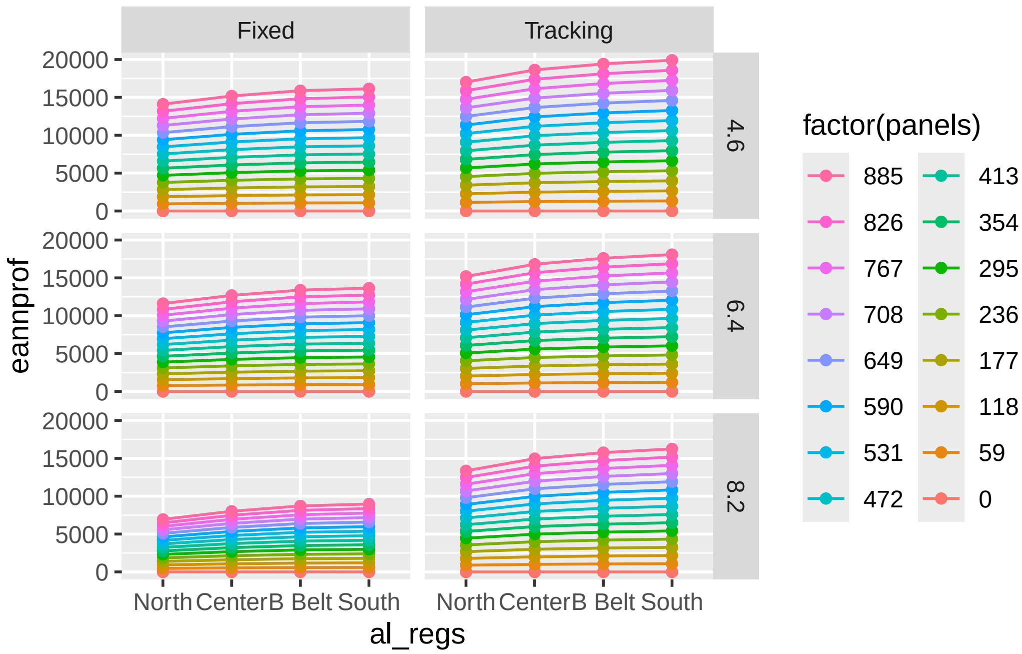
```
Electricity Price =  0.035
```



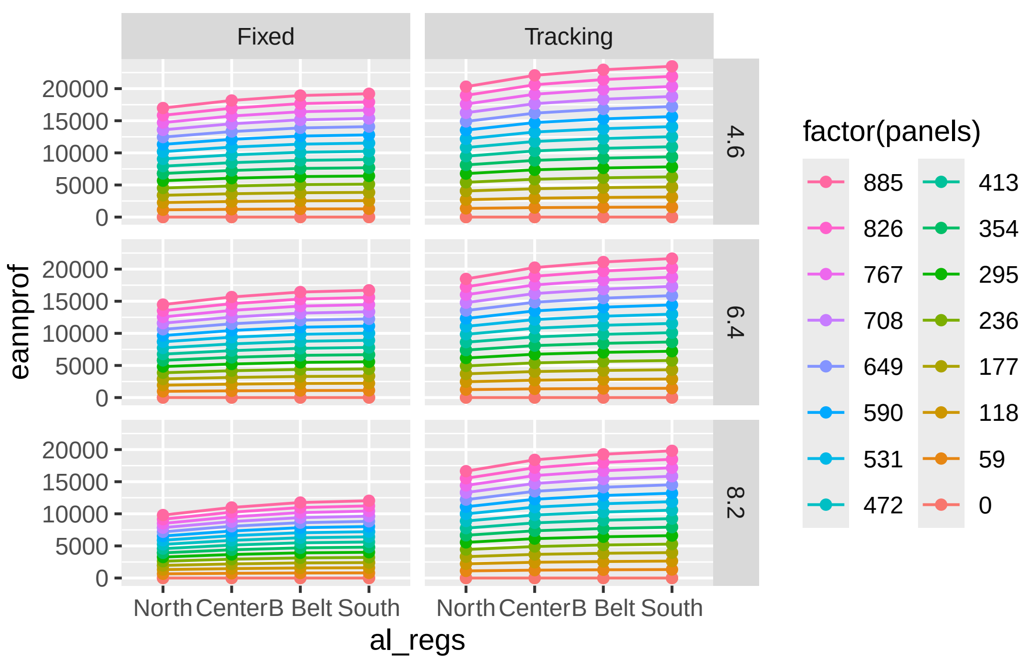

```
Electricity Price =  0.04
```



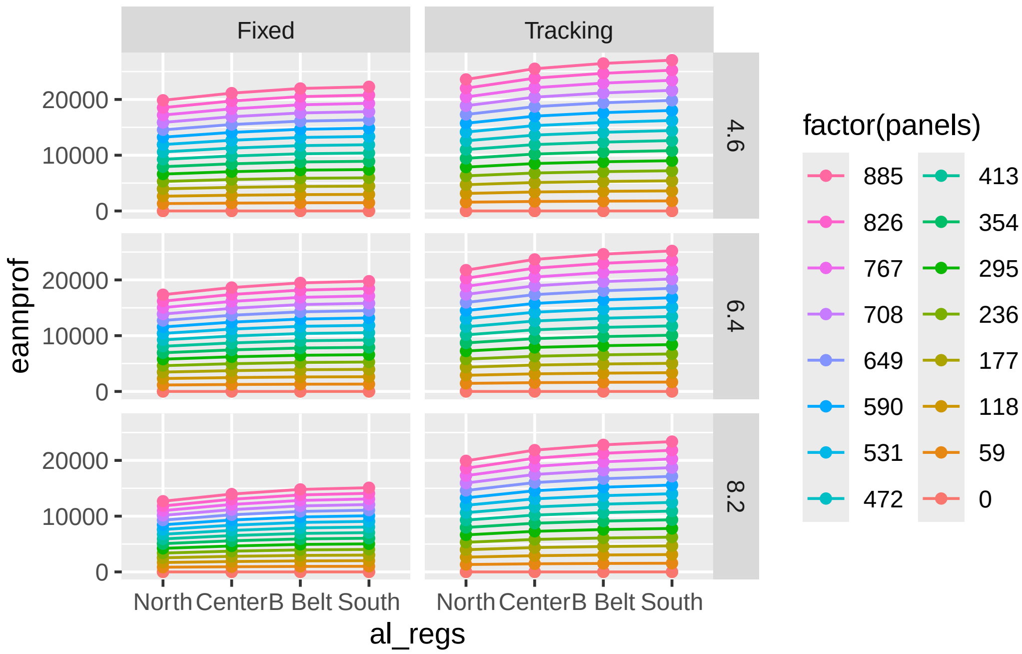
```
Electricity Price =  0.045
```



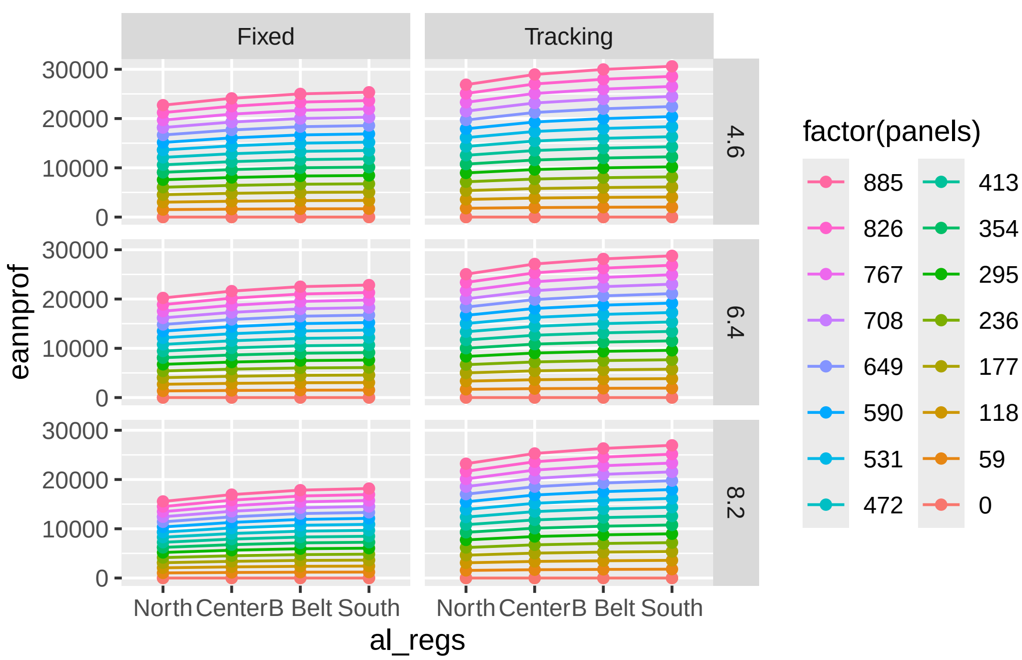

```
Electricity Price =  0.05
```



Electricity Price = 0.055



Electricity Price = 0.06



```r
rm(lox, array_levs, datalot_levs, b, i)
```

4 Profit from crops

4.1 Tomato

Filter return to operator, land and capital profit from Tomato:

```r
tomato_profit = tomato %>%
  select(yldvar, yield,
         rolac17, rolac18, rolac19, rolac20,
         rolac21, rolac22, rolac23)
dim(tomato_profit)
```

```
[1] 21  9
```

```r
tomato_profit
```

```
   yldvar yield    rolac17    rolac18    rolac19    rolac20    rolac21
3     2.0  2720 21679.3826 24399.3826 27119.3826 29839.3826 32559.3826
4     1.9  2584 20065.3826 22649.3826 25233.3826 27817.3826 30401.3826
5     1.8  2448 18451.3826 20899.3826 23347.3826 25795.3826 28243.3826
6     1.7  2312 16837.3826 19149.3826 21461.3826 23773.3826 26085.3826
7     1.6  2176 15223.3826 17399.3826 19575.3826 21751.3826 23927.3826
8     1.5  2040 13609.3826 15649.3826 17689.3826 19729.3826 21769.3826
9     1.4  1904 11995.3826 13899.3826 15803.3826 17707.3826 19611.3826
10    1.3  1768 10381.3826 12149.3826 13917.3826 15685.3826 17453.3826
11    1.2  1632  8767.3826 10399.3826 12031.3826 13663.3826 15295.3826
12    1.1  1496  7153.3826  8649.3826 10145.3826 11641.3826 13137.3826
13    1.0  1360  5539.3826  6899.3826  8259.3826  9619.3826 10979.3826
14    0.9  1224  3925.3826  5149.3826  6373.3826  7597.3826  8821.3826
15    0.8  1088  2311.3826  3399.3826  4487.3826  5575.3826  6663.3826
16    0.7   952   697.3826  1649.3826  2601.3826  3553.3826  4505.3826
17    0.6   816  -916.6174  -100.6174   715.3826  1531.3826  2347.3826
18    0.5   680 -2530.6174 -1850.6174 -1170.6174  -490.6174   189.3826
19    0.4   544 -4144.6174 -3600.6174 -3056.6174 -2512.6174 -1968.6174
20    0.3   408 -5758.6174 -5350.6174 -4942.6174 -4534.6174 -4126.6174
21    0.2   272 -7372.6174 -7100.6174 -6828.6174 -6556.6174 -6284.6174
22    0.1   136 -8986.6174 -8850.6174 -8714.6174 -8578.6174 -8442.6174
```

```
23    0.0     0 -10600.6174 -10600.6174 -10600.6174 -10600.6174 -10600.6174
       rolac22     rolac23
3   35279.3826  37999.3826
4   32985.3826  35569.3826
5   30691.3826  33139.3826
6   28397.3826  30709.3826
7   26103.3826  28279.3826
8   23809.3826  25849.3826
9   21515.3826  23419.3826
10  19221.3826  20989.3826
11  16927.3826  18559.3826
12  14633.3826  16129.3826
13  12339.3826  13699.3826
14  10045.3826  11269.3826
15   7751.3826   8839.3826
16   5457.3826   6409.3826
17   3163.3826   3979.3826
18    869.3826   1549.3826
19  -1424.6174   -880.6174
20  -3718.6174  -3310.6174
21  -6012.6174  -5740.6174
22  -8306.6174  -8170.6174
23 -10600.6174 -10600.6174
```

Convert data to long format:

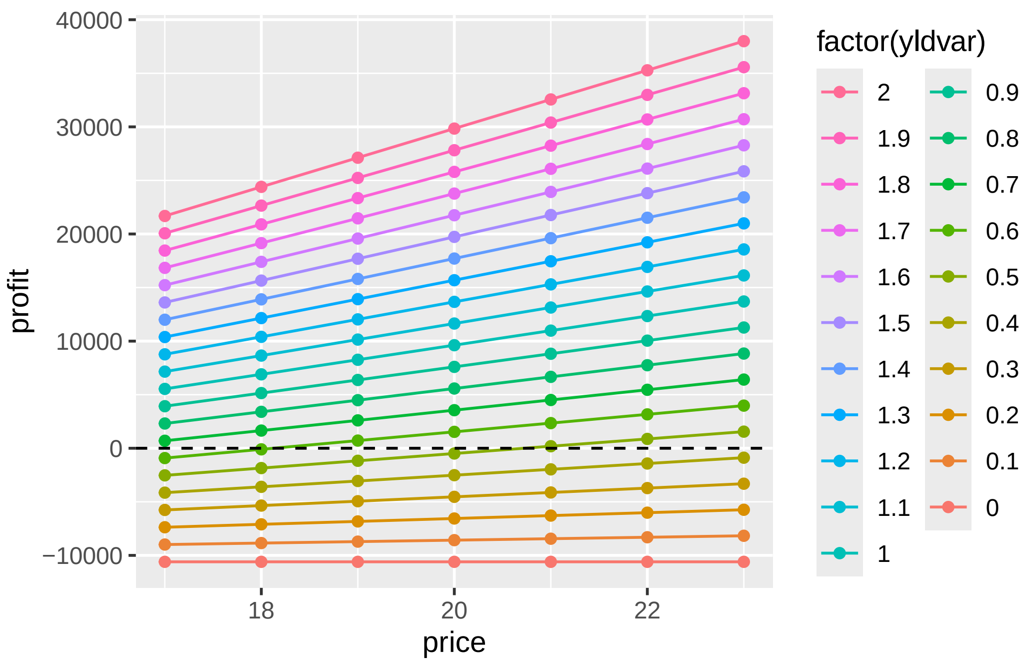
```r
# Assign column names for clarity
colnames(tomato_profit) <- c("yldvar", "yield",
                  "rolac17", "rolac18", "rolac19",
                  "rolac20", "rolac21", "rolac22",
                  "rolac23")

# Reshape the data frame from wide to long format
tomato_long <- melt(tomato_profit,
                id.vars = c("yldvar", "yield"),
                measure.vars = c("rolac17", "rolac18",
                                 "rolac19", "rolac20",
                                 "rolac21", "rolac22",
                                 "rolac23"),
                variable.name = "price",
                value.name = "profit")
```

```r
# Convert the 'Price' column to numeric by extracting the number
tomato_long$price <- as.numeric(gsub("rolac", "", tomato_long$price))
str(tomato_long)
```

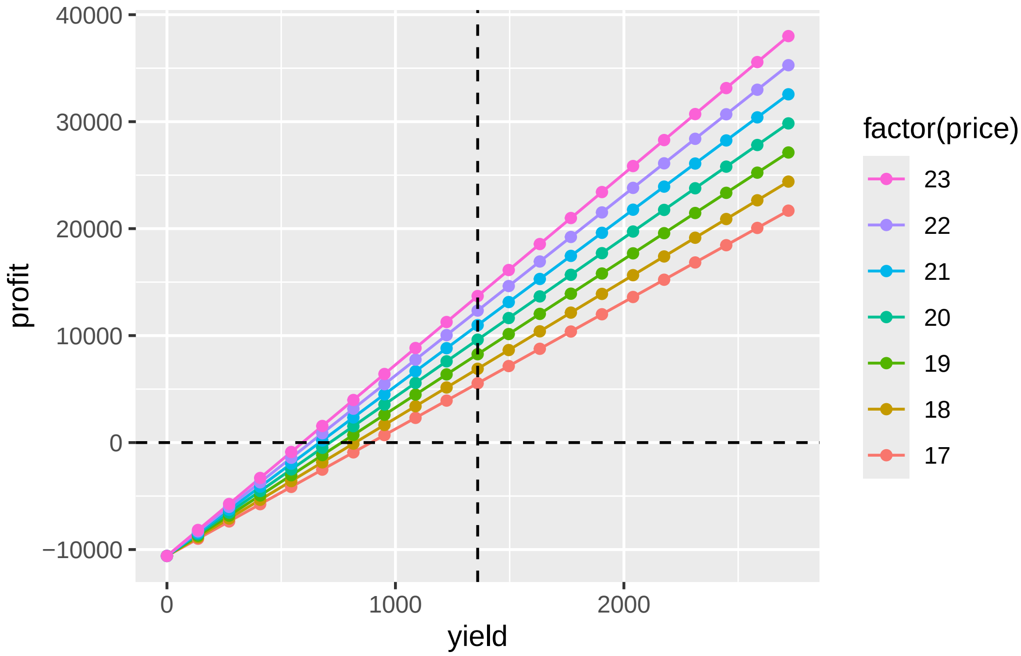
```
'data.frame':	147 obs. of  4 variables:
 $ yldvar: num  2 1.9 1.8 1.7 1.6 1.5 1.4 1.3 1.2 1.1 ...
 $ yield : num  2720 2584 2448 2312 2176 ...
 $ price : num  17 17 17 17 17 17 17 17 17 17 ...
 $ profit: num  21679 20065 18451 16837 15223 ...
```

4.1.1 Plot Tomato Profit

```r
ggplot(data = tomato_long,
       mapping = aes(x = price,
                     y = profit,
                     color = factor(yldvar),
                     group = factor(yield))) +
  geom_line() +
  geom_point() +
  geom_hline(yintercept = 0,
             linetype = "dashed",
             color = "black") +
  guides(color = guide_legend(ncol = 2,
                              reverse = TRUE))
```




```r
ggplot(data = tomato_long,
       mapping = aes(x = yield,
                     y = profit,
                     #fill = yield,
                     color = factor(price),
                     group = factor(price))) +
  geom_line() +
  geom_point() +
  geom_hline(yintercept = 0,
             linetype = "dashed",
             color = "black") +
  # Vertical dashed line is 100% yield
  geom_vline(xintercept = tomato_long$yield[11],
             linetype = "dashed",
             color = "black") +
guides(color = guide_legend(reverse = TRUE))
```



4.2 Strawberry

Filter return to operator, land and capital profit from strawberry

```
strawberry_profit = strawberry %>%
  select(yldvar, yield,
         rolac3, rolac4, rolac5, rolac6,
         rolac7, rolac8, rolac9)
str(strawberry_profit)
```

```
'data.frame':   21 obs. of  9 variables:
 $ yldvar: num  2 1.9 1.8 1.7 1.6 1.5 1.4 1.3 1.2 1.1 ...
 $ yield : num  6150 5843 5535 5228 4920 ...
 $ rolac3: num  719 143 -435 -1010 -1588 ...
 $ rolac4: num  6869 5986 5100 4218 3332 ...
 $ rolac5: num  13019 11829 10635 9446 8252 ...
 $ rolac6: num  19169 17672 16170 14674 13172 ...
 $ rolac7: num  25319 23515 21705 19902 18092 ...
 $ rolac8: num  31469 29358 27240 25130 23012 ...
 $ rolac9: num  37619 35201 32775 30358 27932 ...
```

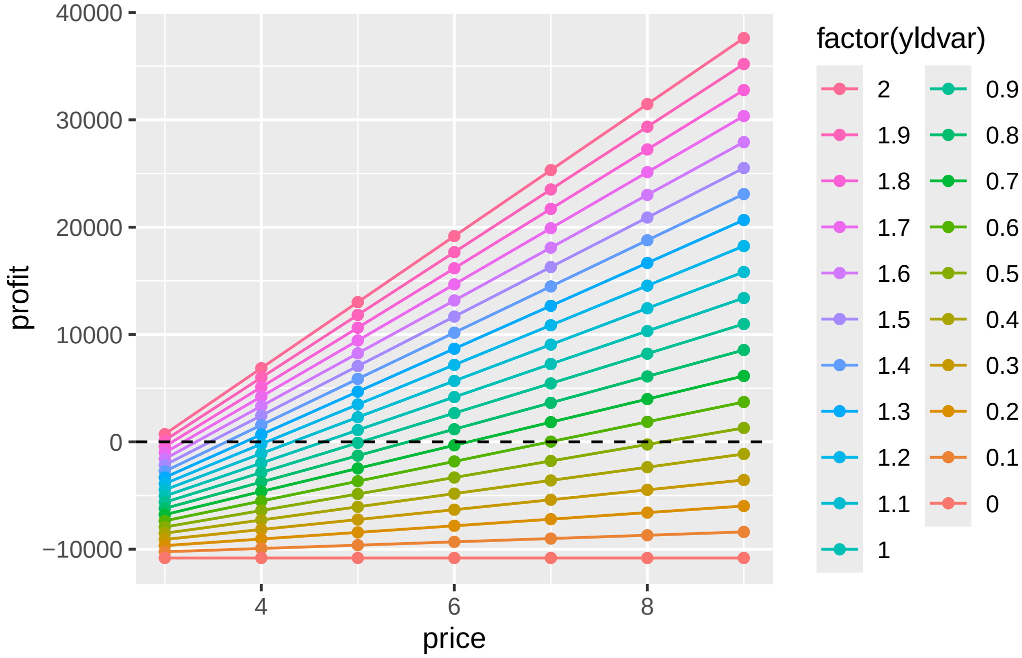
Convert data to long format:

```r
# Assign column names for clarity
colnames(strawberry_profit) <- c("yldvar", "yield",
                  "rolac3", "rolac4", "rolac5",
                  "rolac6", "rolac7", "rolac8",
                  "rolac9")
# Reshape the data frame from wide to long format
stberry_long <- melt(strawberry_profit,
                id.vars = c("yldvar", "yield"),
                measure.vars = c("rolac3", "rolac4",
                                 "rolac5", "rolac6",
                                 "rolac7", "rolac8",
                                 "rolac9"),
                variable.name = "price",
                value.name = "profit")

# Convert the 'Price' column to numeric by extracting the number
stberry_long$price <- as.numeric(gsub("rolac", "", stberry_long$price))
str(stberry_long)
```

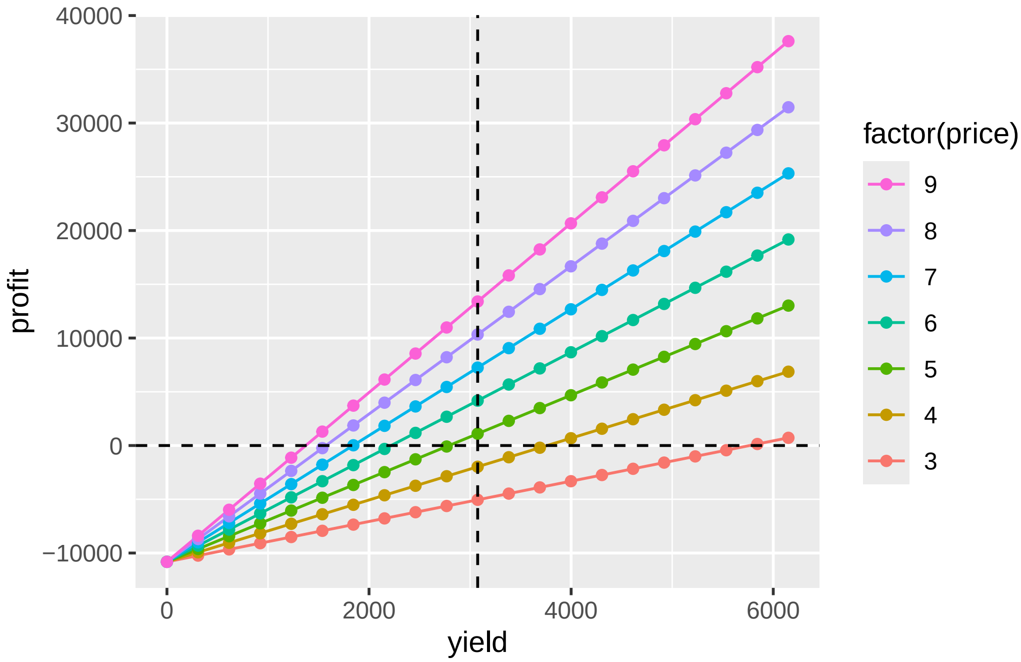
```
'data.frame':	147 obs. of  4 variables:
 $ yldvar: num  2 1.9 1.8 1.7 1.6 1.5 1.4 1.3 1.2 1.1 ...
 $ yield : num  6150 5843 5535 5228 4920 ...
 $ price : num  3 3 3 3 3 3 3 3 3 3 ...
 $ profit: num  719 143 -435 -1010 -1588 ...
```

4.2.1 Plot Strawberry Profit

```r
ggplot(data = stberry_long,
       mapping = aes(x = price,
                     y = profit,
                     color = factor(yldvar),
                     group = factor(yield))) +
  geom_line() +
  geom_point() +
  geom_hline(yintercept = 0,
             linetype = "dashed",
             color = "black") +
  guides(color = guide_legend(ncol = 2,
                              reverse = TRUE))
```




```r
ggplot(data = stberry_long,
       mapping = aes(x = yield,
                     y = profit,
                     color = factor(price),
                     group = factor(price))) +
  geom_line() +
  geom_point() +
  geom_hline(yintercept = 0,
             linetype = "dashed",
             color = "black") +
  #Vertical dashed line is 100% yield
  geom_vline(xintercept = stberry_long$yield[11],
             linetype = "dashed",
             color = "black") +
  guides(color = guide_legend(reverse = TRUE))
```



4.3 Squash

```
squash_profit = squash %>%
  dplyr::select(yldvar, yield,
         rolac11, rolac12, rolac13, rolac14,
         rolac15, rolac16, rolac17)
# Reshape the data frame from wide to long format
squash_long <- reshape2::melt(squash_profit,
                id.vars = c("yldvar", "yield"),
                measure.vars = c("rolac11", "rolac12",
                                 "rolac13", "rolac14",
                                 "rolac15", "rolac16",
                                 "rolac17"),
                variable.name = "price",
                value.name = "profit")

# Convert the 'Price' column to numeric by extracting the number
squash_long$price <- as.numeric(gsub("rolac", "", squash_long$price))
```

5 Profit from agrivoltaics

Total profit from solar and crops for all combinations of AVs simulated.

5.1 Profit from TAV

- Joint profit from tomato (tomato_long) and solar energy production (solar_profit) from 1 acre of land.
- The last variable (tav_profit) is the final profit from tomato agrivoltaic system which is the result of our interest.

```
# Calculate all combinations of rows from both matrices in a vectorized way
solar_expanded <- solar_profit[rep(1:nrow(solar_profit),
                                   each = nrow(tomato_long)), ]
tomato_expanded <- tomato_long[rep(1:nrow(tomato_long),
                                   times = nrow(solar_profit)), ]

# Calculate the new column for tav_profit directly
tav_profit_values <- solar_expanded$eannprof + tomato_expanded$profit

# Combine the matrices and the calculated tav_profit column
tav_profit <- cbind(solar_expanded,
                    tomato_expanded,
                    tav_profit = tav_profit_values)

# Convert to a data frame and ensure the correct format
tav_profit <- as.data.frame(tav_profit)
tav_profit <- data.frame(lapply(tav_profit, unlist))
str(tav_profit)
```

```
'data.frame':   814968 obs. of  29 variables:
 $ sprop                  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ al_regs                : chr  "Black Belt" "Black Belt" "Black Belt" "Black Belt" ...
 $ array                  : chr  "Fixed" "Fixed" "Fixed" "Fixed" ...
 $ dc_kw                  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ panels                 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ energy                 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ elcprc                 : num  0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
 $ elcrev                 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ height                 : num  4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 ...
 $ capex                  : num  1.59 1.59 1.59 1.59 1.59 ...
```

```
 $ landlease           : num  1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 ...
 $ ttlcost             : num  0 0 0 0 0 0 0 0 0 0 ...
 $ inscst              : num  0 0 0 0 0 0 0 0 0 0 ...
 $ recredit            : num  0 0 0 0 0 0 0 0 0 0 ...
 $ reap                : num  0 0 0 0 0 0 0 0 0 0 ...
 $ annlzcost           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ annoftotcost        : num  0 0 0 0 0 0 0 0 0 0 ...
 $ monthlycost         : num  0 0 0 0 0 0 0 0 0 0 ...
 $ opex                : num  0 0 0 0 0 0 0 0 0 0 ...
 $ taxcr               : num  0 0 0 0 0 0 0 0 0 0 ...
 $ anncost             : num  0 0 0 0 0 0 0 0 0 0 ...
 $ eannprof            : num  0 0 0 0 0 0 0 0 0 0 ...
 $ eannprofworeap      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ eannprofwoincentives: num  0 0 0 0 0 0 0 0 0 0 ...
 $ yldvar              : num  2 1.9 1.8 1.7 1.6 1.5 1.4 1.3 1.2 1.1 ...
 $ yield               : num  2720 2584 2448 2312 2176 ...
 $ price               : num  17 17 17 17 17 17 17 17 17 17 ...
 $ profit              : num  21679 20065 18451 16837 15223 ...
 $ tav_profit          : num  21679 20065 18451 16837 15223 ...
```

```r
rm(solar_expanded, tomato_expanded, tav_profit_values)
```

5.1.1 Saving results locally

```r
write_feather(tav_profit,
  sink = "Data/tav_profit R50.feather",
  version = 2,
  chunk_size = 65536L,
  compression = c("default"),
  #compression = c("default", "lz4", "lz4_frame", "uncompressed", "zstd"),
  compression_level = NULL
)
tictoc::tic("Using Dplyr:")
write_xlsx(x = tav_profit %>%
             dplyr::sample_n(100),
           file = "Results/TAV Profit Sample R50.xlsx",
           as_table = TRUE)
tictoc::toc()
```

```
Using Dplyr:: 0.08 sec elapsed
```

```r
write_xlsx(x = tav_profit %>%
             filter(sprop %in% c(0, 0.25, 0.50, 0.75, 1),
                    yldvar == 1,
                    price == 20,
                    elcprc == 0.04)%>%
             dplyr::select(sprop, panels, height, array,
                           al_regs, yldvar, yield, price,
                           elcprc, tav_profit) %>%
             mutate(al_regs1 = case_when(
               al_regs == "Northern" ~ 1,
    al_regs == "Central" ~ 2,
    al_regs == "Black Belt" ~ 3,
    al_regs == "Southern" ~ 4,
    TRUE ~ NA_real_)),
           file = "Results/TAV Profit WriteUp R50.xlsx",
           as_table = TRUE)
```

5.2 Profit from SBAV

- Joint profit from strawberry (stberry_long) and solar energy production (solar_profit) from 1 acre of land.
- The last variable (sbav_profit) is the final profit from strawberry agrivoltaic system which is the result of our interest.

```r
# Generate all combinations of rows from both matrices in a vectorized way
solar_expanded <- solar_profit[rep(1:nrow(solar_profit),
                                   each = nrow(stberry_long)), ]
stberry_expanded <- stberry_long[rep(1:nrow(stberry_long),
                                     times = nrow(solar_profit)), ]

# Calculate the new column for sbav_profit directly
sbav_profit_values <- solar_expanded$eannprof + stberry_expanded$profit

# Combine the matrices and the calculated sbav_profit column
sbav_profit <- cbind(solar_expanded,
                     stberry_expanded,
                     sbav_profit = sbav_profit_values)

# Convert to a data frame and ensure the correct format
sbav_profit <- as.data.frame(sbav_profit)
```

```r
sbav_profit <- data.frame(lapply(sbav_profit, unlist))
rm(solar_expanded, stberry_expanded, sbav_profit_values)
```

5.3 Filtering Strawberry Results

```r
sbav_profit %>%
  dplyr::filter(height == 8.2,
         array == "Tracking",
         elcprc == 0.04,
         sprop == 0.70,
         yldvar == 1,
         price == 3) %>%
  dplyr::select(height, array, elcprc, al_regs,
                yldvar, price, sprop, sbav_profit)
```

```
  height    array elcprc    al_regs yldvar price sprop sbav_profit
1    8.2 Tracking   0.04 Black Belt      1     3   0.7    3111.020
2    8.2 Tracking   0.04    Central      1     3   0.7    2633.929
3    8.2 Tracking   0.04   Northern      1     3   0.7    1666.234
4    8.2 Tracking   0.04   Southern      1     3   0.7    3405.579
```

5.3.1 Saving results locally

```r
#write_csv(sbav_profit, "tav_profit.csv")
write_feather(sbav_profit,
  sink = "Data/sbav_profit R50.feather",
  version = 2,
  chunk_size = 65536L,
  compression = c("default"),
  #compression = c("default", "lz4", "lz4_frame", "uncompressed", "zstd"),
  compression_level = NULL
)
tictoc::tic("Using Base R Matrix:")
write_xlsx(x = sbav_profit[sample(nrow(sbav_profit), 100),],
           file = "Results/SBAV Profit Sample R50.xlsx",
           as_table = TRUE)
tictoc::toc()
```

Using Base R Matrix:: 0.47 sec elapsed

```
write_xlsx(x = sbav_profit %>%
             filter(sprop %in% c(0, 0.25, 0.50, 0.75, 1),
                    yldvar == 1,
                    price == 6,
                    elcprc == 0.04)%>%
             dplyr::select(sprop, panels, height, array, al_regs,
                           #price, elcprc, yldvar, yield,
                           sbav_profit) %>%
             mutate(al_regs1 = case_when(
               al_regs == "Northern" ~ 1,
    al_regs == "Central" ~ 2,
    al_regs == "Black Belt" ~ 3,
    al_regs == "Southern" ~ 4,
    TRUE ~ NA_real_)),
           file = "Results/SBAV Profit WriteUp R50.xlsx",
           as_table = TRUE)
```

5.4 Profit from SQAV

```
solar_expanded <- solar_profit[rep(1:nrow(solar_profit),
                                   each = nrow(squash_long)), ]
squash_expanded <- squash_long[rep(1:nrow(squash_long),
                                   times = nrow(solar_profit)), ]

# Calculate the new column for tav_profit directly
sqav_profit_values <- solar_expanded$eannprof + squash_expanded$profit

# Combine the matrices and the calculated tav_profit column
sqav_profit <- cbind(solar_expanded,
                     squash_expanded,
                     sqav_profit = sqav_profit_values)

# Convert to a data frame and ensure the correct format
sqav_profit <- as.data.frame(sqav_profit)
sqav_profit <- data.frame(lapply(sqav_profit, unlist))

# Create a new variable
sqav_profit <- sqav_profit %>%
```

```r
  group_by(price) %>%  # Control for unique prices
  mutate(
    sqavp_ge_sq = if_else(yldvar == 1 &
                            sqav_profit >= profit, 1, 0)
  ) %>%
  ungroup()

# SQAV Profit Greater or Equal to Squash
sqavp_ge_squash = sqav_profit %>% filter(sqavp_ge_sq == 1)

write_feather(sqav_profit,
  sink = "Data/sqav_profit R50.feather",
  version = 2,
  chunk_size = 65536L,
  compression = c("default"),
  compression_level = NULL
)

write_xlsx(x = sqav_profit[sample(nrow(sqav_profit), 100),],
           file = "Results/SQAV Profit Sample R50.xlsx",
           as_table = TRUE)

write_xlsx(x = sqav_profit %>%
             filter(sprop %in% c(0, 0.25, 0.50, 0.75, 1),
                    yldvar == 1,
                    price == 14,
                    elcprc == 0.04)%>%
             dplyr::select(sprop, panels, height, array,
                           al_regs, yldvar, yield, price,
                           elcprc, sqav_profit) %>%
             mutate(al_regs1 = case_when(
               al_regs == "Northern" ~ 1,
    al_regs == "Central" ~ 2,
    al_regs == "Black Belt" ~ 3,
    al_regs == "Southern" ~ 4,
    TRUE ~ NA_real_)),
           file = "Results/SQAV Profit WriteUp R50.xlsx",
           as_table = TRUE)

write_xlsx(
  x = sqavp_ge_squash %>%
    dplyr::filter(sqavp_ge_sq == 1) %>%
```

```r
    dplyr::select(
      sprop, panels, height, array, al_regs,
      yldvar, yield, price, elcprc, sqav_profit
    ) %>%
    mutate(al_regs1 = case_when(
      al_regs == "Northern" ~ 1,
      al_regs == "Central" ~ 2,
      al_regs == "Black Belt" ~ 3,
      al_regs == "Southern" ~ 4,
      TRUE ~ NA_real_
    )),
  file = "Results/SQAV Profit GE Squash R50.xlsx",
  as_table = TRUE
)
rm(solar_expanded, squash_expanded, squash_long, sqav_profit_values)
```