AV Simulation REAP25

Bijesh Mishra, Ph.D.

2024-12-09

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

```r
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

```r
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
```

```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
```

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 ...
```

```r
head(tomato); tail(tomato)
```

```
  yldvar yield Rev17 Rev18 Rev19 Rev20 Rev21 Rev22 Rev23 Total Cost  rolac17
```

3	2.0	2720	46240	48960	51680	54400	57120	59840	62560	24560.62	21679.38
4	1.9	2584	43928	46512	49096	51680	54264	56848	59432	23862.62	20065.38
5	1.8	2448	41616	44064	46512	48960	51408	53856	56304	23164.62	18451.38
6	1.7	2312	39304	41616	43928	46240	48552	50864	53176	22466.62	16837.38
7	1.6	2176	36992	39168	41344	43520	45696	47872	50048	21768.62	15223.38
8	1.5	2040	34680	36720	38760	40800	42840	44880	46920	21070.62	13609.38

	rolac18	rolac19	rolac20	rolac21	rolac22	rolac23	Operator Cost	rlc17
3	24399.38	27119.38	29839.38	32559.38	35279.38	37999.38	2700	18979.38
4	22649.38	25233.38	27817.38	30401.38	32985.38	35569.38	2565	17500.38
5	20899.38	23347.38	25795.38	28243.38	30691.38	33139.38	2430	16021.38
6	19149.38	21461.38	23773.38	26085.38	28397.38	30709.38	2295	14542.38
7	17399.38	19575.38	21751.38	23927.38	26103.38	28279.38	2160	13063.38
8	15649.38	17689.38	19729.38	21769.38	23809.38	25849.38	2025	11584.38

	rlc18	rlc19	rlc20	rlc21	rlc22	rlc223
3	21699.38	24419.38	27139.38	29859.38	32579.38	35299.38
4	20084.38	22668.38	25252.38	27836.38	30420.38	33004.38
5	18469.38	20917.38	23365.38	25813.38	28261.38	30709.38
6	16854.38	19166.38	21478.38	23790.38	26102.38	28414.38
7	15239.38	17415.38	19591.38	21767.38	23943.38	26119.38
8	13624.38	15664.38	17704.38	19744.38	21784.38	23824.38

	yldvar	yield	Rev17	Rev18	Rev19	Rev20	Rev21	Rev22	Rev23	Total Cost	rolac17
18	0.5	680	11560	12240	12920	13600	14280	14960	15640	14090.62	-2530.617
19	0.4	544	9248	9792	10336	10880	11424	11968	12512	13392.62	-4144.617
20	0.3	408	6936	7344	7752	8160	8568	8976	9384	12694.62	-5758.617
21	0.2	272	4624	4896	5168	5440	5712	5984	6256	11996.62	-7372.617
22	0.1	136	2312	2448	2584	2720	2856	2992	3128	11298.62	-8986.617
23	0.0	0	0	0	0	0	0	0	0	10600.62	-10600.617

	rolac18	rolac19	rolac20	rolac21	rolac22	rolac23
18	-1850.617	-1170.617	-490.6174	189.3826	869.3826	1549.3826
19	-3600.617	-3056.617	-2512.6174	-1968.6174	-1424.6174	-880.6174
20	-5350.617	-4942.617	-4534.6174	-4126.6174	-3718.6174	-3310.6174
21	-7100.617	-6828.617	-6556.6174	-6284.6174	-6012.6174	-5740.6174
22	-8850.617	-8714.617	-8578.6174	-8442.6174	-8306.6174	-8170.6174
23	-10600.617	-10600.617	-10600.6174	-10600.6174	-10600.6174	-10600.6174

	Operator Cost	rlc17	rlc18	rlc19	rlc20	rlc21
18	675	-3205.617	-2525.617	-1845.617	-1165.617	-485.6174
19	540	-4684.617	-4140.617	-3596.617	-3052.617	-2508.6174
20	405	-6163.617	-5755.617	-5347.617	-4939.617	-4531.6174
21	270	-7642.617	-7370.617	-7098.617	-6826.617	-6554.6174
22	135	-9121.617	-8985.617	-8849.617	-8713.617	-8577.6174
23	0	-10600.617	-10600.617	-10600.617	-10600.617	-10600.6174

```
         rlc22      rlc223
18    194.3826    874.3826
19  -1964.6174  -1420.6174
20  -4123.6174  -3715.6174
21  -6282.6174  -6010.6174
22  -8441.6174  -8305.6174
23 -10600.6174 -10600.6174
```

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 ...
```

```
head(strawberry); tail(strawberry)
```

```
  yldvar yield  Rev3  Rev4  Rev5  Rev6  Rev7  Rev8  Rev9 Total Cost    rolac3
3    2.0  6150 18450 24600 30750 36900 43050 49200 55350   17730.79   719.205
4    1.9  5843 17529 23372 29215 35058 40901 46744 52587   17385.71   143.288
5    1.8  5535 16605 22140 27675 33210 38745 44280 49815   17039.50  -434.505
6    1.7  5228 15684 20912 26140 31368 36596 41824 47052   16694.42 -1010.422
7    1.6  4920 14760 19680 24600 29520 34440 39360 44280   16348.21 -1588.215
8    1.5  4613 13839 18452 23065 27678 32291 36904 41517   16003.13 -2164.132
    rolac4    rolac5   rolac6   rolac7   rolac8   rolac9 Operator Cost
3 6869.205 13019.205 19169.21 25319.21 31469.21 37619.21       2700.00
4 5986.288 11829.288 17672.29 23515.29 29358.29 35201.29       2565.22
5 5100.495 10635.495 16170.50 21705.50 27240.50 32775.50       2430.00
6 4217.578  9445.578 14673.58 19901.58 25129.58 30357.58       2295.22
7 3331.785  8251.785 13171.79 18091.79 23011.79 27931.79       2160.00
8 2448.868  7061.868 11674.87 16287.87 20900.87 25513.87       2025.22
       rlc3      rlc4      rlc5      rlc6     rlc7     rlc8     rlc9
3 -1980.795 4169.2050 10319.205 16469.205 22619.21 28769.21 34919.21
4 -2421.932 3421.0685  9264.068 15107.068 20950.07 26793.07 32636.07
5 -2864.505 2670.4950  8205.495 13740.495 19275.50 24810.50 30345.50
6 -3305.642 1922.3585  7150.358 12378.358 17606.36 22834.36 28062.36
7 -3748.215 1171.7850  6091.785 11011.785 15931.79 20851.79 25771.79
8 -4189.352  423.6485  5036.648  9649.648 14262.65 18875.65 23488.65
```

```
   yldvar yield Rev3 Rev4 Rev5 Rev6  Rev7  Rev8  Rev9 Total Cost     rolac3
18    0.5  1538 4614 6152 7690 9228 10766 12304 13842   12546.68  -7932.682
19    0.4  1230 3690 4920 6150 7380  8610  9840 11070   12200.47  -8510.475
20    0.3   923 2769 3692 4615 5538  6461  7384  8307   11855.39  -9086.392
21    0.2   615 1845 2460 3075 3690  4305  4920  5535   11509.18  -9664.185
22    0.1   308  924 1232 1540 1848  2156  2464  2772   11164.10 -10240.102
23    0.0     0    0    0    0    0     0     0     0   10817.89 -10817.895
      rolac4     rolac5     rolac6     rolac7     rolac8     rolac9
```

```
18  -6394.682  -4856.682  -3318.682  -1780.682   -242.682   1295.318
19  -7280.475  -6050.475  -4820.475  -3590.475  -2360.475  -1130.475
20  -8163.392  -7240.392  -6317.392  -5394.392  -4471.392  -3548.392
21  -9049.185  -8434.185  -7819.185  -7204.185  -6589.185  -5974.185
22  -9932.102  -9624.102  -9316.102  -9008.102  -8700.102  -8392.102
23 -10817.895 -10817.895 -10817.895 -10817.895 -10817.895 -10817.895
   Operator Cost       rlc3       rlc4       rlc5       rlc6       rlc7
18      675.2195  -8607.902  -7069.902  -5531.902  -3993.902  -2455.902
19      540.0000  -9050.475  -7820.475  -6590.475  -5360.475  -4130.475
20      405.2195  -9491.612  -8568.612  -7645.612  -6722.612  -5799.612
21      270.0000  -9934.185  -9319.185  -8704.185  -8089.185  -7474.185
22      135.2195 -10375.322 -10067.322  -9759.322  -9451.322  -9143.322
23        0.0000 -10817.895 -10817.895 -10817.895 -10817.895 -10817.895
          rlc8        rlc9
18   -917.9015    620.0985
19  -2900.4750  -1670.4750
20  -4876.6115  -3953.6115
21  -6859.1850  -6244.1850
22  -8835.3215  -8527.3215
23 -10817.8950 -10817.8950
```

2.3 Squash

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

```r
squash <- read_xlsx("Data/Parameters.xlsx",
                    sheet = "Squash",
                    start_row = 2,
                    start_col = 8,
                    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 ...
```

```
head(squash); tail(squash)
```

```
  yldvar yield Rev11 Rev12 Rev13 Rev14 Rev15 Rev16 Rev17 Total Cost   rolac11
3    2.0  2180 23980 26160 28340 30520 32700 34880 37060   13670.88 10309.117
4    1.9  2071 22781 24852 26923 28994 31065 33136 35207   13173.63  9607.367
5    1.8  1962 21582 23544 25506 27468 29430 31392 33354   12676.38  8905.617
6    1.7  1853 20383 22236 24089 25942 27795 29648 31501   12179.13  8203.867
7    1.6  1744 19184 20928 22672 24416 26160 27904 29648   11681.88  7502.117
8    1.5  1635 17985 19620 21255 22890 24525 26160 27795   11184.63  6800.367
    rolac12  rolac13  rolac14  rolac15  rolac16  rolac17 Operator Cost    rlc11
3 12489.117 14669.12 16849.12 19029.12 21209.12 23389.12          2700 7609.117
4 11678.367 13749.37 15820.37 17891.37 19962.37 22033.37          2565 7042.367
5 10867.617 12829.62 14791.62 16753.62 18715.62 20677.62          2430 6475.617
6 10056.867 11909.87 13762.87 15615.87 17468.87 19321.87          2295 5908.867
7  9246.117 10990.12 12734.12 14478.12 16222.12 17966.12          2160 5342.117
8  8435.367 10070.37 11705.37 13340.37 14975.37 16610.37          2025 4775.367
     rlc12     rlc13     rlc14    rlc15    rlc16    rlc17
3 9789.117 11969.117 14149.117 16329.12 18509.12 20689.12
4 9113.367 11184.367 13255.367 15326.37 17397.37 19468.37
```

```
5 8437.617 10399.617 12361.617 14323.62 16285.62 18247.62
6 7761.867  9614.867 11467.867 13320.87 15173.87 17026.87
7 7086.117  8830.117 10574.117 12318.12 14062.12 15806.12
8 6410.367  8045.367  9680.367 11315.37 12950.37 14585.37
```

```
   yldvar yield Rev11 Rev12 Rev13 Rev14 Rev15 Rev16 Rev17 Total Cost   rolac11
18    0.5   545  5995  6540  7085  7630  8175  8720  9265   6212.133  -217.133
19    0.4   436  4796  5232  5668  6104  6540  6976  7412   5714.883  -918.883
20    0.3   327  3597  3924  4251  4578  4905  5232  5559   5217.633 -1620.633
21    0.2   218  2398  2616  2834  3052  3270  3488  3706   4720.383 -2322.383
22    0.1   109  1199  1308  1417  1526  1635  1744  1853   4223.133 -3024.133
23    0.0     0     0     0     0     0     0     0     0   3725.883 -3725.883
     rolac12     rolac13   rolac14   rolac15     rolac16   rolac17
18   327.867   872.86702  1417.867  1962.867  2507.86702  3052.867
19  -482.883   -46.88298   389.117   825.117  1261.11702  1697.117
20 -1293.633  -966.63298  -639.633  -312.633    14.36702   341.367
21 -2104.383 -1886.38298 -1668.383 -1450.383 -1232.38298 -1014.383
22 -2915.133 -2806.13298 -2697.133 -2588.133 -2479.13298 -2370.133
23 -3725.883 -3725.88298 -3725.883 -3725.883 -3725.88298 -3725.883
   Operator Cost     rlc11     rlc12     rlc13     rlc14     rlc15     rlc16
18           675  -892.133  -347.133   197.867   742.867  1287.867  1832.867
19           540 -1458.883 -1022.883  -586.883  -150.883   285.117   721.117
20           405 -2025.633 -1698.633 -1371.633 -1044.633  -717.633  -390.633
21           270 -2592.383 -2374.383 -2156.383 -1938.383 -1720.383 -1502.383
22           135 -3159.133 -3050.133 -2941.133 -2832.133 -2723.133 -2614.133
23             0 -3725.883 -3725.883 -3725.883 -3725.883 -3725.883 -3725.883
         rlc17
18  2377.86702
19  1157.11702
20   -63.63298
21 -1284.38298
22 -2505.13298
23 -3725.88298
```

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)`)
str(elec_price)
```

```
'data.frame':   11 obs. of  1 variable:
 $ epr_kwh: num  0.01 0.015 0.02 0.025 0.03 0.035 0.04 0.045 0.05 0.055 ...
```

```
elec_price
```

```
   epr_kwh
2    0.010
3    0.015
4    0.020
5    0.025
6    0.030
7    0.035
8    0.040
9    0.045
10   0.050
11   0.055
12   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.

```r
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 ...
```

```r
head(pvsc); tail(pvsc)
```

```
            avtyps                     item cost height tcost
2 Typical Fixed PV EPC/Developer Net Profit 0.11    4.6  1.53
3 Typical Fixed PV       Developer Overhead 0.15    4.6  1.53
4 Typical Fixed PV          Contingency(3%) 0.05    4.6  1.53
5 Typical Fixed PV      Interconnection Fee 0.03    4.6  1.53
6 Typical Fixed PV  Permitting Fee (if any) 0.02    4.6  1.53
7 Typical Fixed PV        Sale Tax (if any) 0.05    4.6  1.53
```

```
                                   avtyps                         item cost
104 PV + Crops (Reinforced Regular Mount)                 EPC Overhead 0.25
105 PV + Crops (Reinforced Regular Mount) Installation and Labor Cost 0.32
106 PV + Crops (Reinforced Regular Mount)               Electrical BOS 0.38
107 PV + Crops (Reinforced Regular Mount)               Structural BOS 0.32
108 PV + Crops (Reinforced Regular Mount)                Inverter Only 0.08
109 PV + Crops (Reinforced Regular Mount)                       Module 0.40
    height tcost
104    8.2  2.33
105    8.2  2.33
```

```
106     8.2  2.33
107     8.2  2.33
108     8.2  2.33
109     8.2  2.33
```

2.6 Capex (NREL)

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)
str(capex)
```

```
'data.frame':	6 obs. of  3 variables:
 $ height: num  4.6 4.6 6.4 8.2 8.2 6.4
 $ capex : num  1.59 1.73 1.85 2.33 2.11 ...
 $ array : chr  "Fixed" "Tracking" "Fixed" "Fixed" ...
```

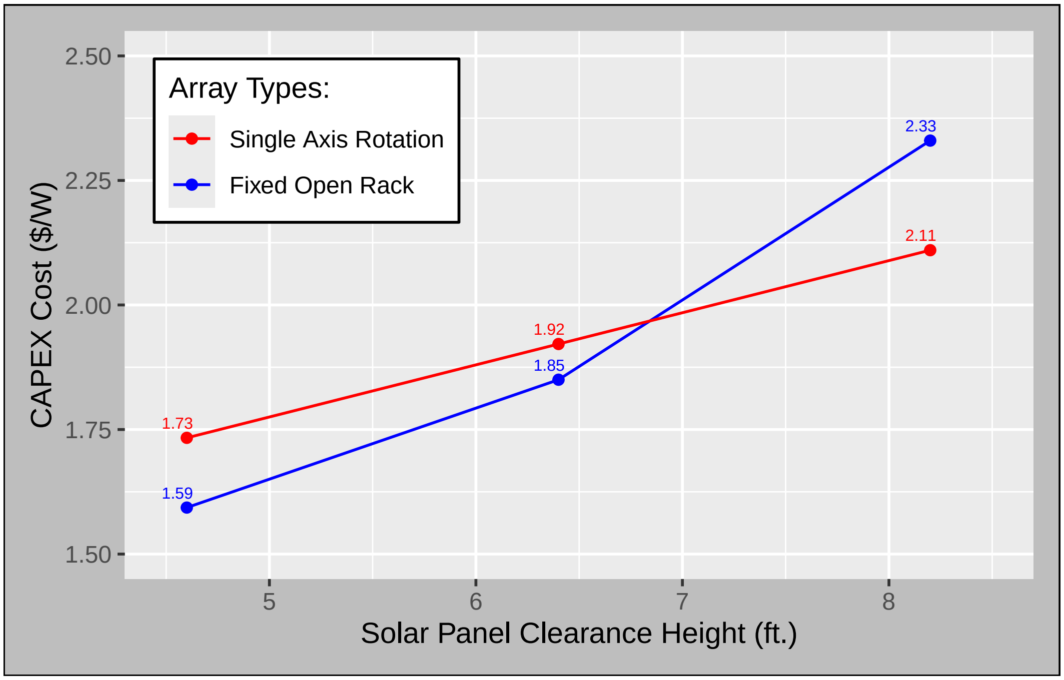
```
capex
```

```
  height    capex    array
1    4.6 1.593333    Fixed
2    4.6 1.733333 Tracking
3    6.4 1.850000    Fixed
4    8.2 2.330000    Fixed
5    8.2 2.110000 Tracking
6    6.4 1.921667 Tracking
```

2.6.1 Plotting capex

```r
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")
  )
```




```
# Save the plot
ggsave(
  filename = "Plots/CAPEX Solar Panels R25.png",
  width = 8,
  height = 6,
  units = "in"
  )
```

2.7 Panel Configuration

- Panel configuration and DV system output (W).

```
panconf <- wb_read(file = "Data/Parameters.xlsx",
                sheet = "PV Spacing",
                start_row = 2,
```

```r
                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 ...
```

```r
head(panconf); tail(panconf)
```

```
  Total Area (Acre) Total Area (Sq. Ft.) Solar Proportion
3                 1                43560             1.00
4                 1                43560             0.95
5                 1                43560             0.90
6                 1                43560             0.85
7                 1                43560             0.80
8                 1                43560             0.75
  Solar Proportion Area (Sq. Ft.) Solar Proportion Area (Sq.M.)
3                           43560                      4046.856
4                           41382                      3844.513
```

5	39204	3642.170
6	37026	3439.828
7	34848	3237.485
8	32670	3035.142

	Side Length (ft.)	YSide Length (ft.)	XSide length (ft.)	Panel Length (ft.)
3	208.7103	208.7103	208.7103	7.75
4	208.7103	208.7103	198.2748	7.75
5	208.7103	208.7103	187.8393	7.75
6	208.7103	208.7103	177.4038	7.75
7	208.7103	208.7103	166.9683	7.75
8	208.7103	208.7103	156.5327	7.75

	Row Seperator (ft.)	Panel Width(ft.)	Panel Area (Sq. ft.)	Panels/Row
3	6	3.5	27.125	59
4	6	3.5	27.125	59
5	6	3.5	27.125	59
6	6	3.5	27.125	59
7	6	3.5	27.125	59
8	6	3.5	27.125	59

	Total Rows	Total Panels	Array Area (Sq. Ft.)	Array Area (Sq. M.)
3	15	885	24005.62	2230.195
4	14	826	22405.25	2081.516
5	13	767	20804.88	1932.836
6	12	708	19204.50	1784.156
7	12	708	19204.50	1784.156
8	11	649	17604.12	1635.477

	XSide Open Length (ft)	Inter Panel Spacing (ft)	Panel Efficienfy
3	92	6	0.19
4	100	7	0.19
5	107	8	0.19
6	115	10	0.19
7	115	10	0.19
8	123	12	0.19

	DC System Size (kW)
3	423.7371
4	395.4880
5	367.2388
6	338.9897
7	338.9897
8	310.7405

	Total Area (Acre)	Total Area (Sq. Ft.)	Solar Proportion
18	1	43560	0.25

19	1	43560	0.20
20	1	43560	0.15
21	1	43560	0.10
22	1	43560	0.05
23	1	43560	0.00

	Solar Proportion Area (Sq. Ft.)	Solar Proportion Area (Sq.M.)
18	10890	1011.7140
19	8712	809.3712
20	6534	607.0284
21	4356	404.6856
22	2178	202.3428
23	0	0.0000

	Side Length (ft.)	YSide Length (ft.)	XSide length (ft.)	Panel Length (ft.)
18	208.7103	208.7103	52.17758	7.75
19	208.7103	208.7103	41.74207	7.75
20	208.7103	208.7103	31.30655	7.75
21	208.7103	208.7103	20.87103	7.75
22	208.7103	208.7103	10.43552	7.75
23	208.7103	208.7103	0.00000	7.75

	Row Seperator (ft.)	Panel Width(ft.)	Panel Area (Sq. ft.)	Panels/Row
18	6	3.5	27.125	59
19	6	3.5	27.125	59
20	6	3.5	27.125	59
21	6	3.5	27.125	59
22	6	3.5	27.125	59
23	6	3.5	27.125	59

	Total Rows	Total Panels	Array Area (Sq. Ft.)	Array Area (Sq. M.)
18	3	177	4801.125	446.0391
19	3	177	4801.125	446.0391
20	2	118	3200.750	297.3594
21	1	59	1600.375	148.6797
22	0	0	0.000	0.0000
23	0	0	0.000	0.0000

	XSide Open Length (ft)	Inter Panel Spacing (ft)	Panel Efficienfy
18	185	92	0.19
19	185	92	0.19
20	193	193	0.19
21	200	NA	0.19
22	208	NA	0.19
23	208	NA	0.19

	DC System Size (kW)
18	84.74742
19	84.74742

```
20              56.49828
21              28.24914
22               0.00000
23               0.00000
```

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)`) %>%
```

```r
  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 ...
```

```r
head(energy_output); tail(energy_output)
```

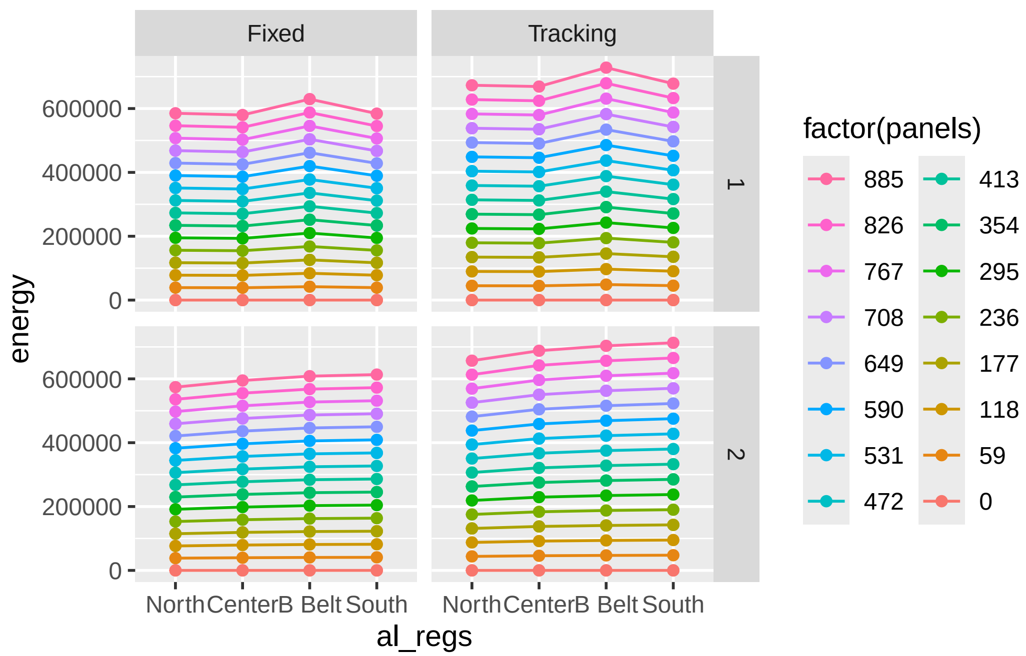
```
  sprop panels datalot    al_regs  zips    array  dc_kw energy
2     1    885       1   Northern 35801 Tracking 423.74 672887
3     1    885       1   Northern 35801    Fixed 423.74 585225
4     1    885       1    Central 35223 Tracking 423.74 668895
5     1    885       1    Central 35223    Fixed 423.74 579758
6     1    885       1 Black Belt 36117 Tracking 423.74 728181
7     1    885       1 Black Belt 36117    Fixed 423.74 629523

    sprop panels datalot    al_regs  zips    array dc_kw energy
332     0      0       2    Central 35136 Tracking     0      0
333     0      0       2    Central 35136    Fixed     0      0
334     0      0       2 Black Belt 36040 Tracking     0      0
335     0      0       2 Black Belt 36040    Fixed     0      0
336     0      0       2   Southern 36507 Tracking     0      0
337     0      0       2   Southern 36507    Fixed     0      0
```

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.

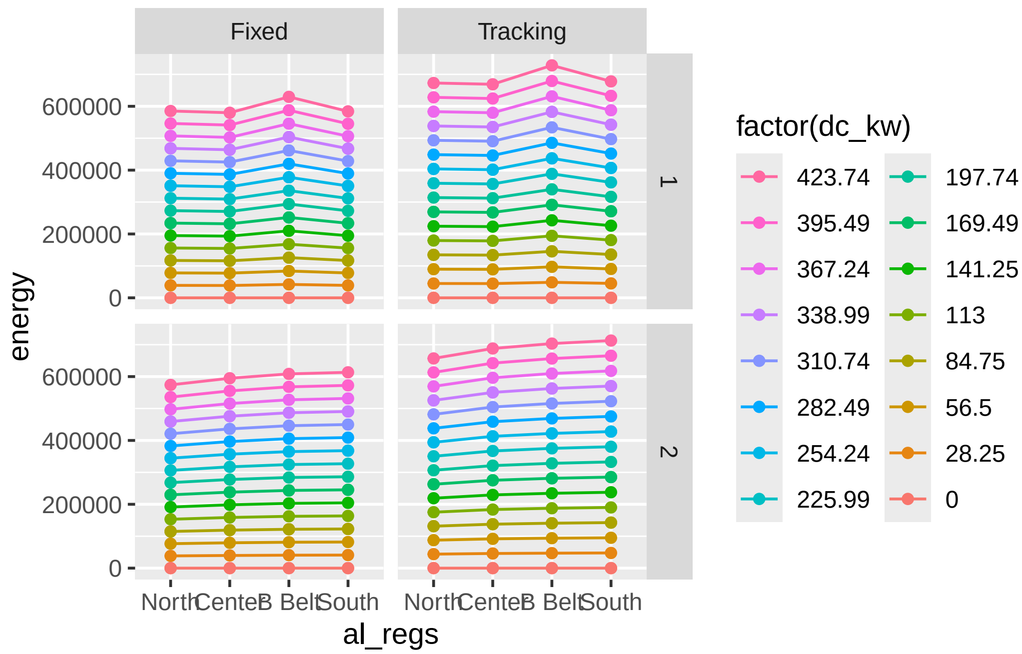
```
lox <- c("Northern", "Central", "Black Belt", "Southern")
array_levs = c("Single Axis Rotation", "Fixed Open Rack")
datalot_levs = c("Location 1", "Location 2")
ggplot(data = energy_output,
         mapping = aes(x = al_regs,
                       y = energy,
                       #fill = energy,
                       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))
```



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.

```r
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))
```



3 Solar Energy

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.

```
# 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
```

```r
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."
```

```r
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 ...
```

```r
head(energy_revenue); tail(energy_revenue)
```

```
  sprop    al_regs    array dc_kw panels energy elcprc elcrev
1     0 Black Belt    Fixed     0      0      0   0.01      0
2     0 Black Belt Tracking     0      0      0   0.01      0
3     0    Central    Fixed     0      0      0   0.01      0
4     0    Central Tracking     0      0      0   0.01      0
5     0   Northern    Fixed     0      0      0   0.01      0
6     0   Northern Tracking     0      0      0   0.01      0
```

```
     sprop  al_regs    array  dc_kw panels energy elcprc   elcrev
1843     1  Central    Fixed 423.74    885 594824   0.06 35689.44
1844     1  Central Tracking 423.74    885 688037   0.06 41282.22
1845     1 Northern    Fixed 423.74    885 574020   0.06 34441.20
1846     1 Northern Tracking 423.74    885 656889   0.06 39413.34
1847     1 Southern    Fixed 423.74    885 613342   0.06 36800.52
1848     1 Southern Tracking 423.74    885 712873   0.06 42772.38
```

3.2 Simulation 2: Energy Revenue

This simulation has same result as above (Cross checking above code and output). Results are suppressed but errors and warnings are not. No error and no warnings means code is working as it should.

```
## | results='hide'
# Sample data
set.seed(123)
matrix1 <- energy_output # dimension of matrix is 176*7
matrix2 <- elec_price # dimension of matrix is 11*1

# Initializing the result matrix
result_matrix <- data.frame(matrix(ncol = nrow(matrix2),
                                   nrow = 0))
colnames(result_matrix) <- c(colnames(matrix1),  "elcrev", "elcprc")

# Loop to multiply first and second matrices
for (i in 1:nrow(matrix2)) {
  temp_matrix <- matrix1
  temp_matrix$E_Prc <- matrix2[i, ]
  temp_matrix$E_Rev <- matrix1$energy[j] * matrix2$epr_kwh[i]
  result_matrix <- rbind(result_matrix, temp_matrix)
}
str(result_matrix)
head(result_matrix); tail(result_matrix)
```

3.3 Plots: Energy Revenue

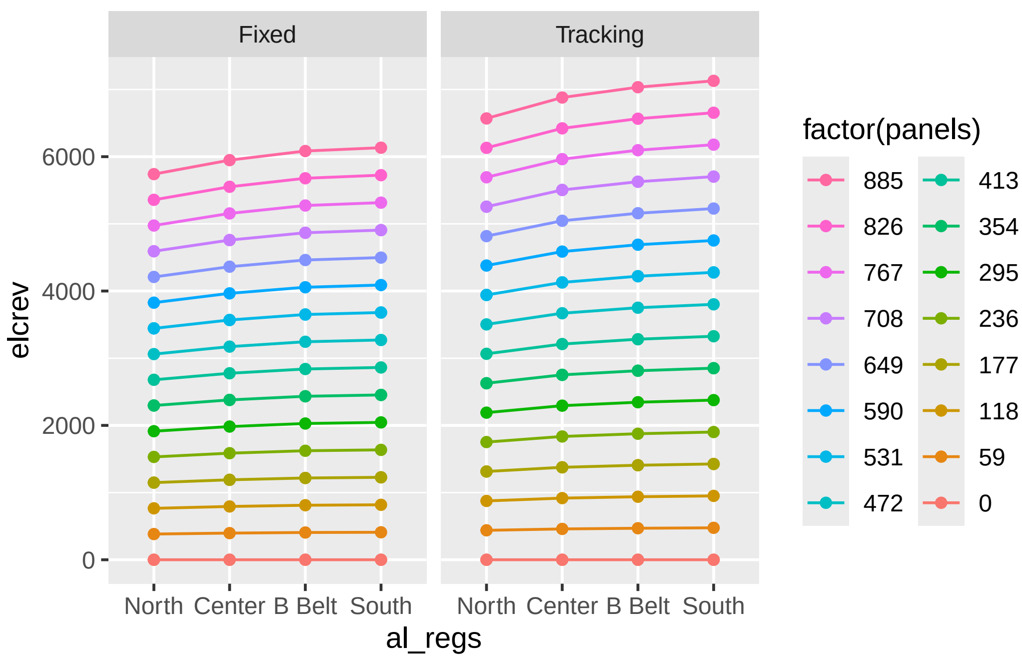
3.3.1 By # of solar panels

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.

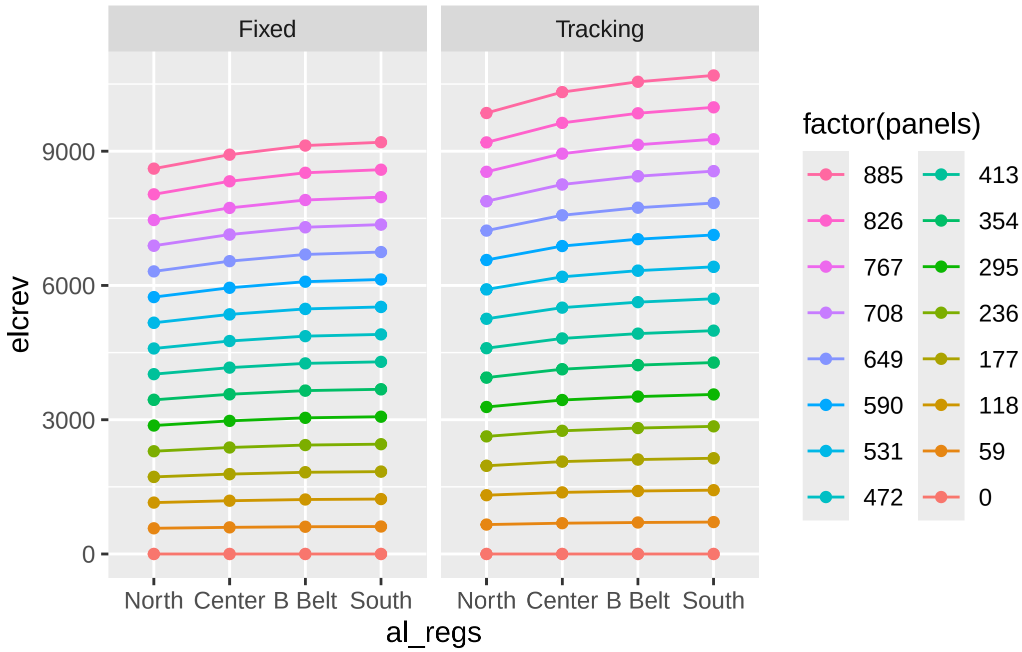
```
lox <- c("Northern", "Central", "Black Belt", "Southern")
array_levs = c("Single Axis Rotation", "Fixed Open Rack")
datalot_levs = c("Location 1", "Location 2")
for (i in unique(energy_revenue$elcprc)) {
 a = ggplot(data = (energy_revenue %>%
  dplyr::filter(elcprc == i)),
```

```
        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)
}
```

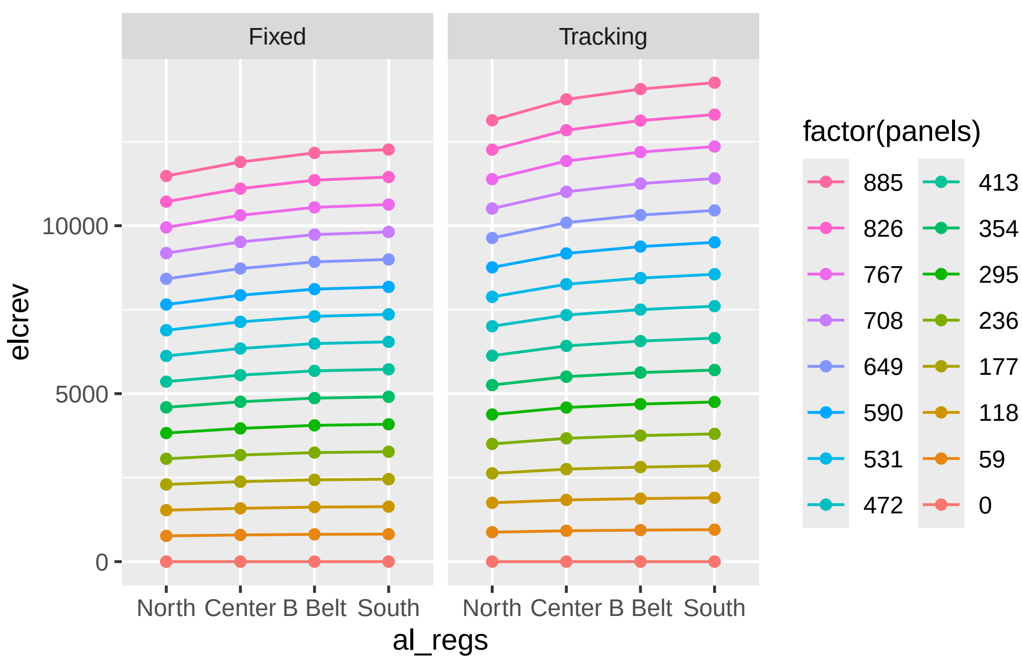
```
Electricity Price =  0.01
```



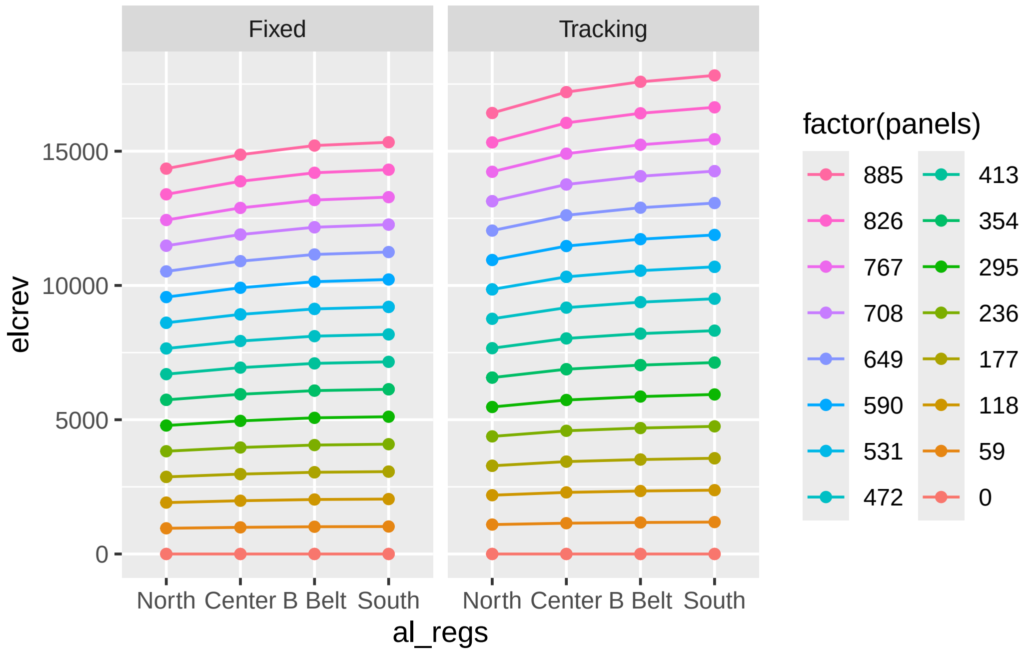

```
Electricity Price =  0.015
```



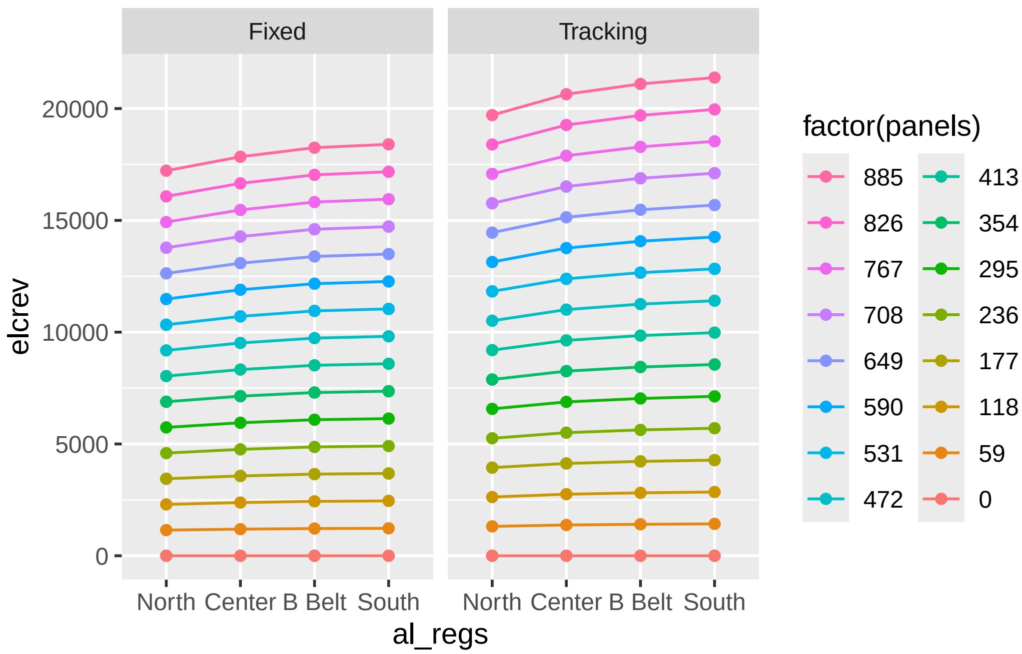

```
Electricity Price =  0.02
```



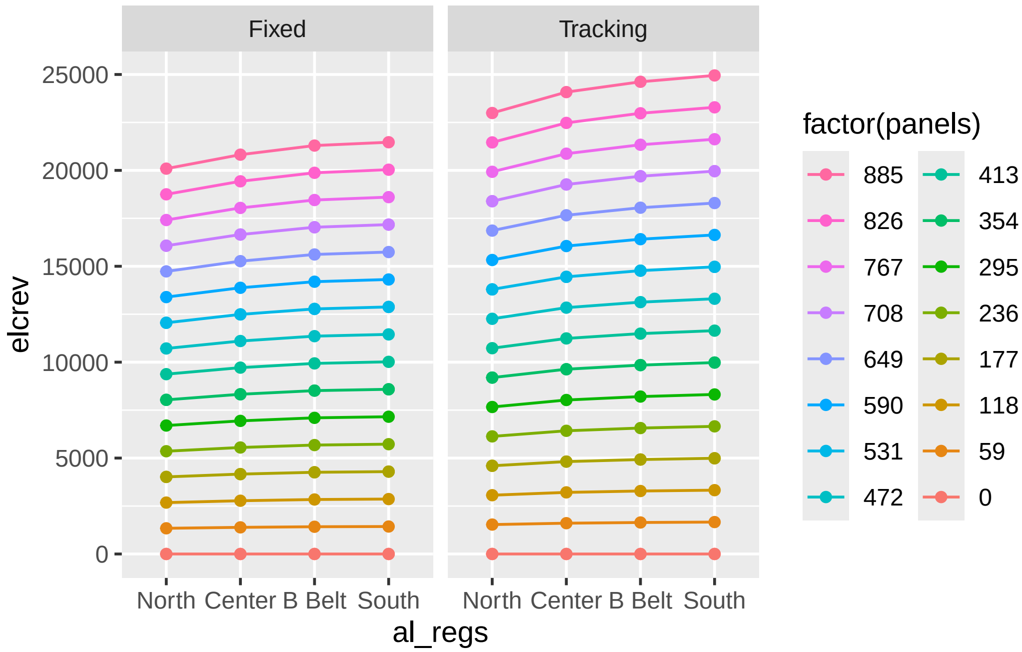

```
Electricity Price =  0.025
```



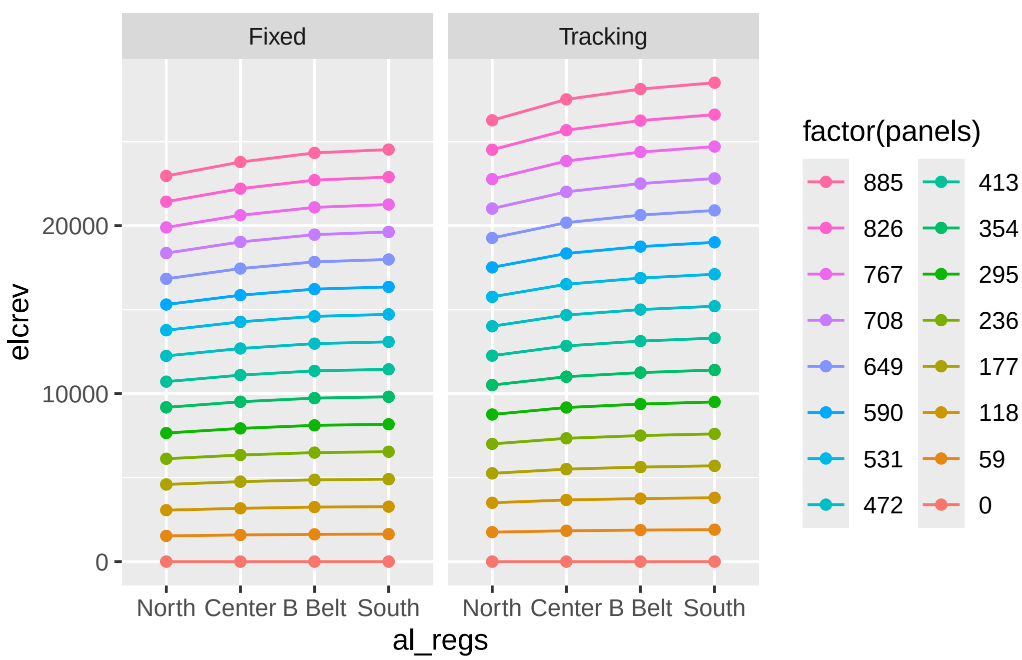

```
Electricity Price =  0.03
```



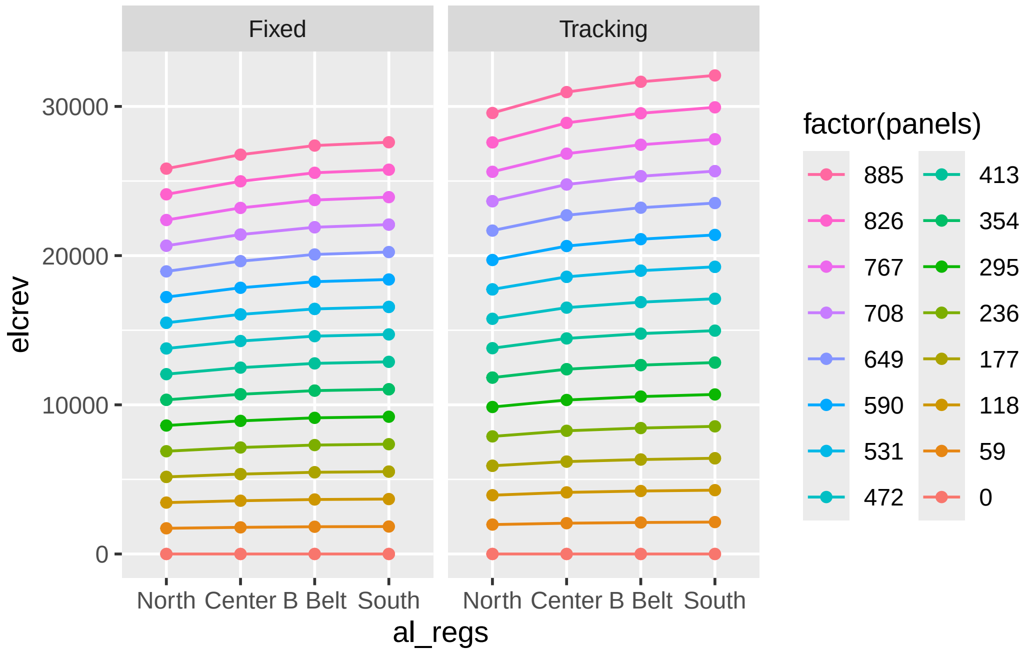

```
Electricity Price =  0.035
```



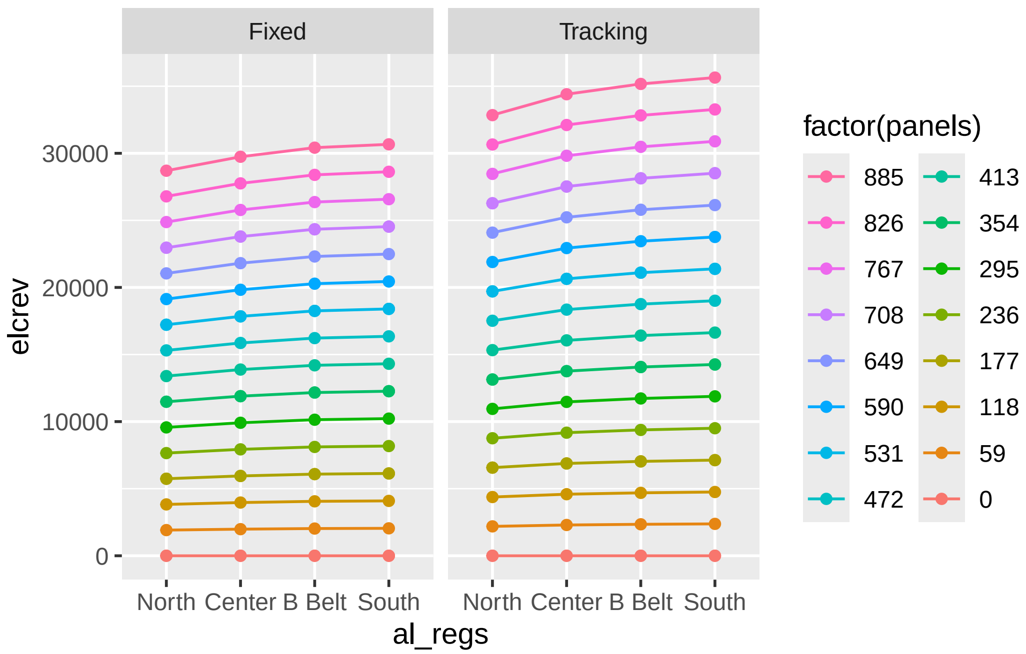
Electricity Price = 0.04



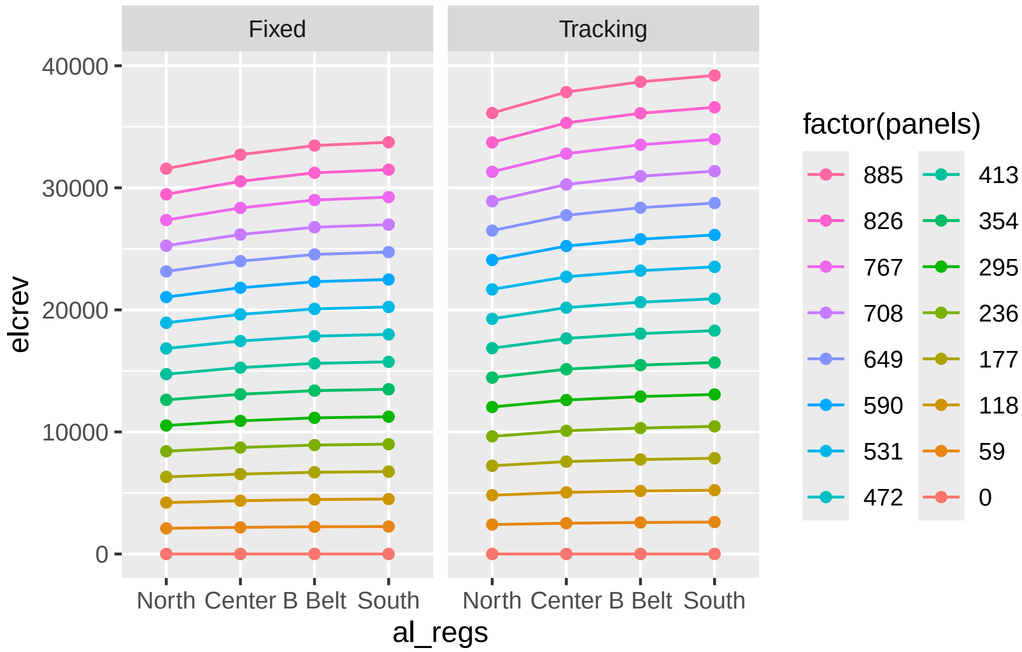
Electricity Price = 0.045



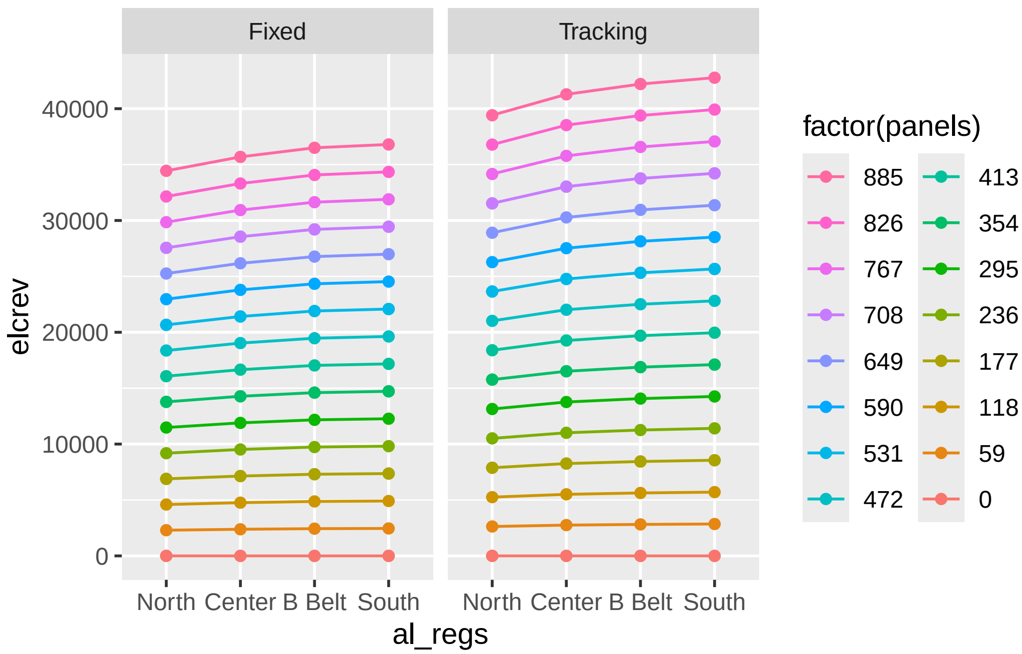

```
Electricity Price =  0.05
```




```
Electricity Price =  0.055
```




```
Electricity Price =  0.06
```

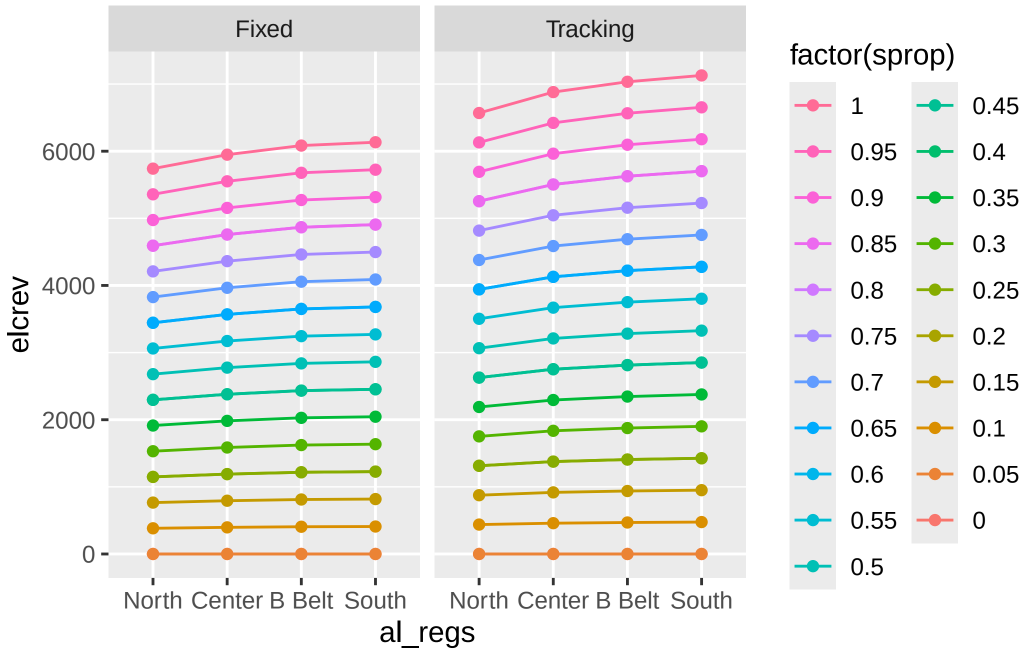


3.3.2 By Land in Solar

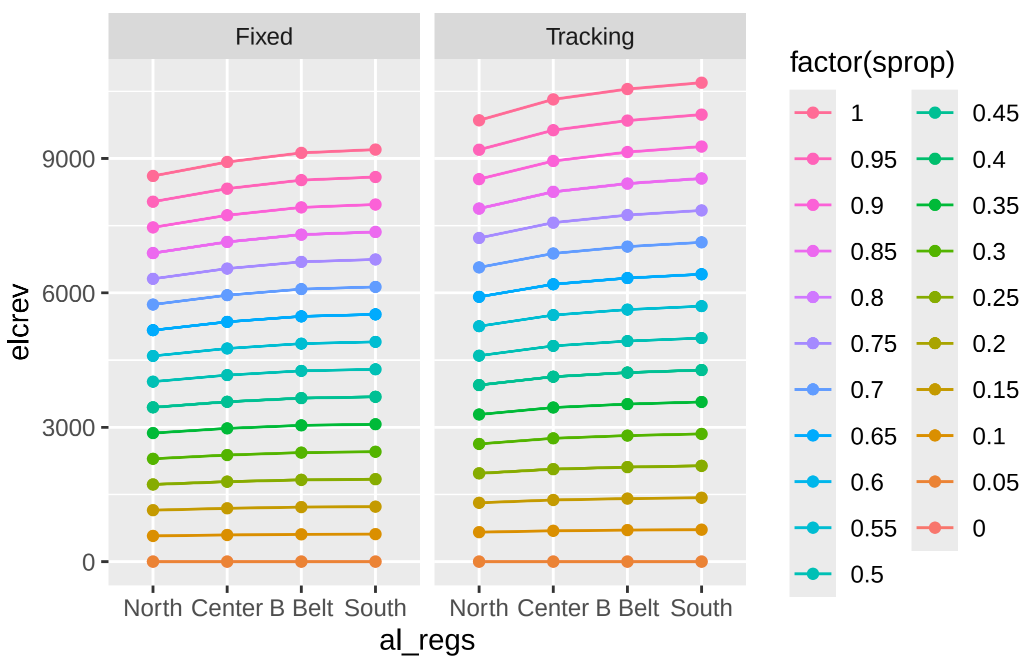
- Two proportions may have same number of solar panels (Eg. 0.80 and 0.85, 0.20 and 0.25). So, total lines in the chart may not match with total number of legend levels. Some proportions are overlapping in the chart. See panel configuration for more detail.

```
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)
}
```

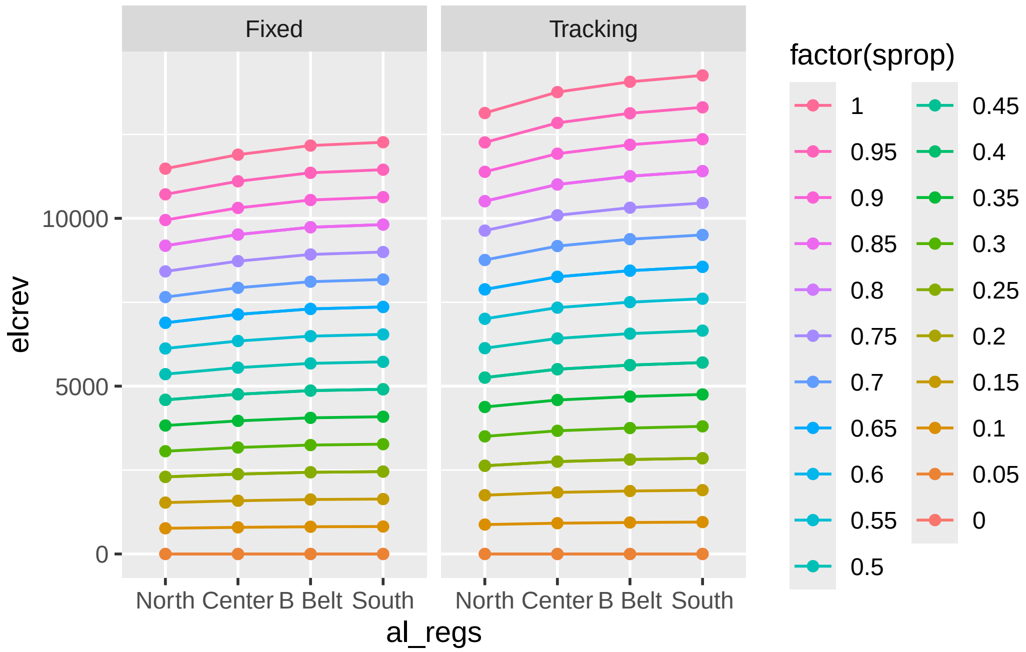
```
Electricity Price =  0.01
```



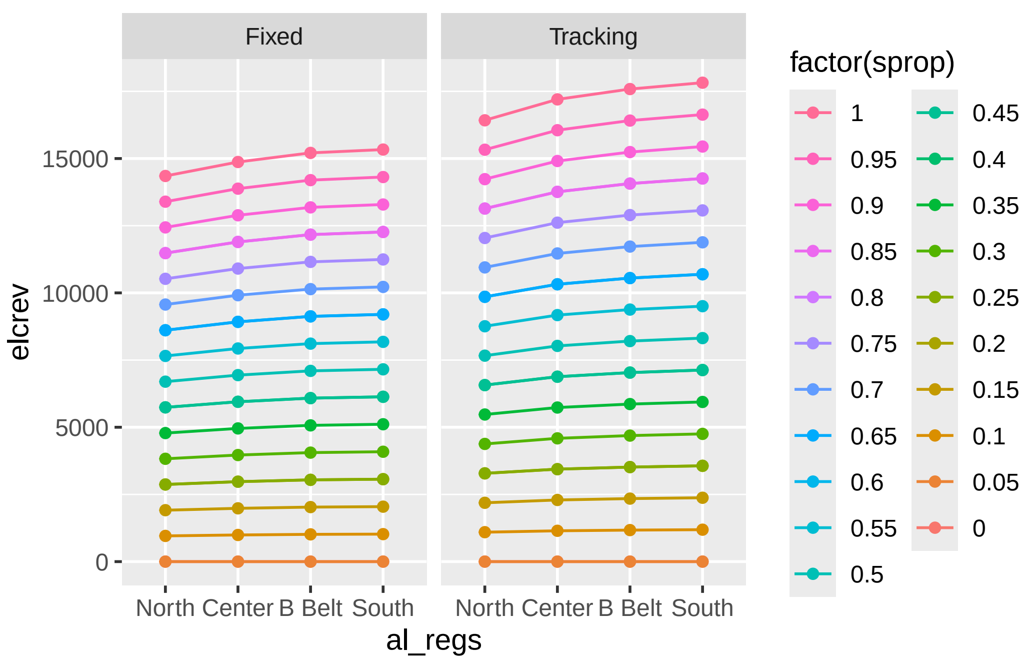

```
Electricity Price =  0.015
```



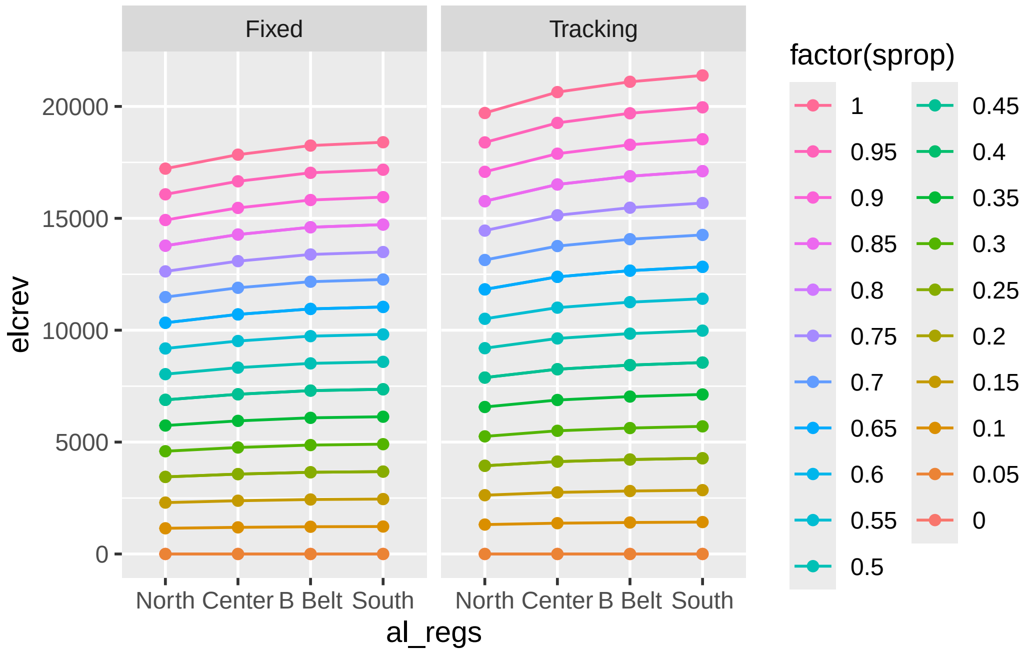

```
Electricity Price =  0.02
```



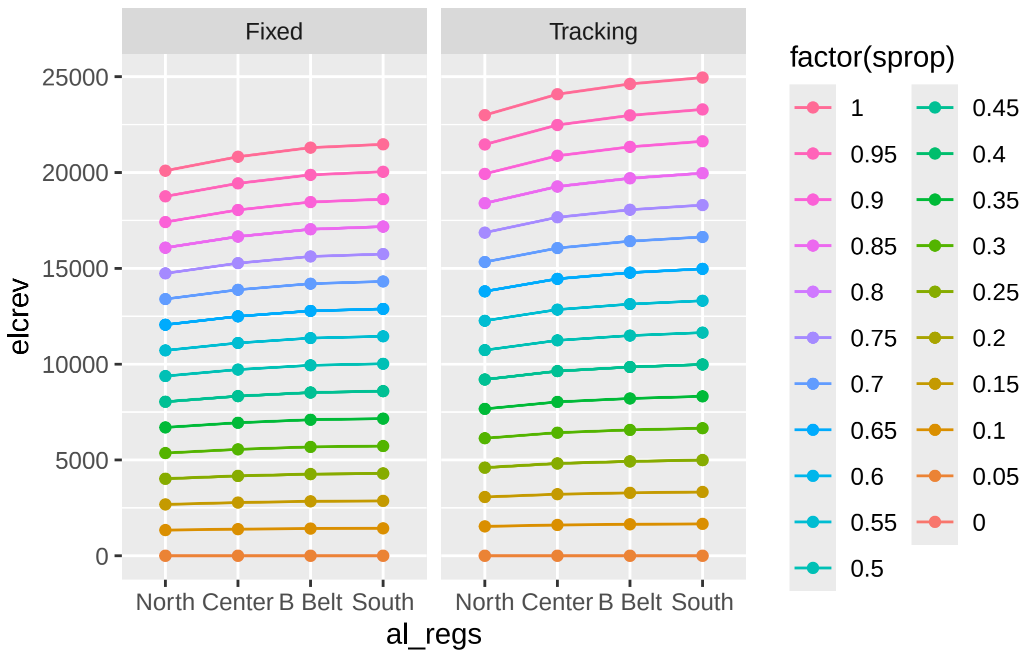
Electricity Price = 0.025



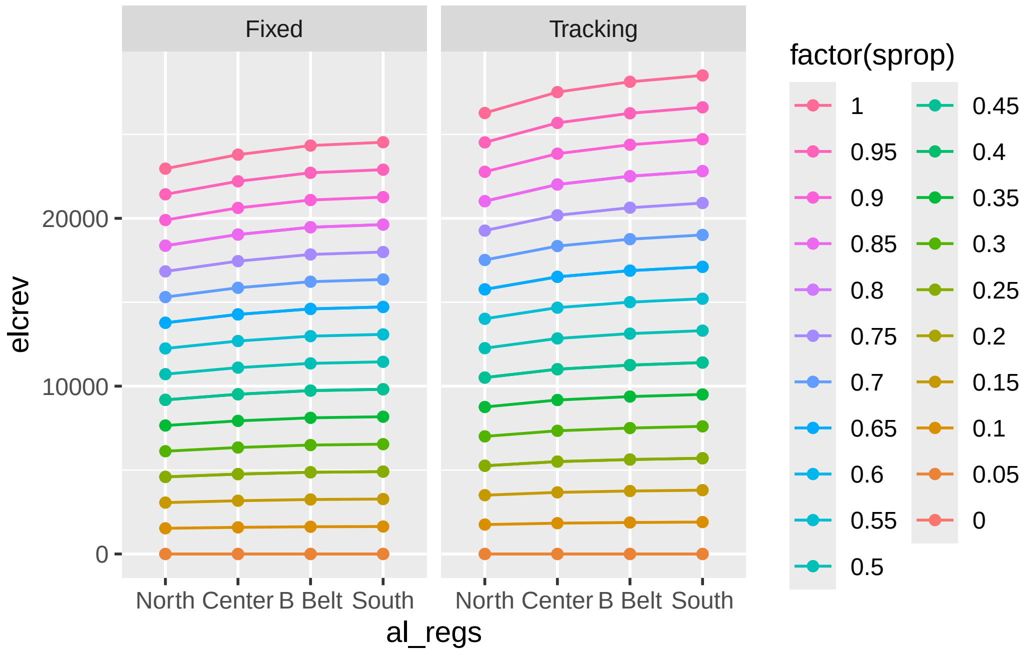
Electricity Price = 0.03



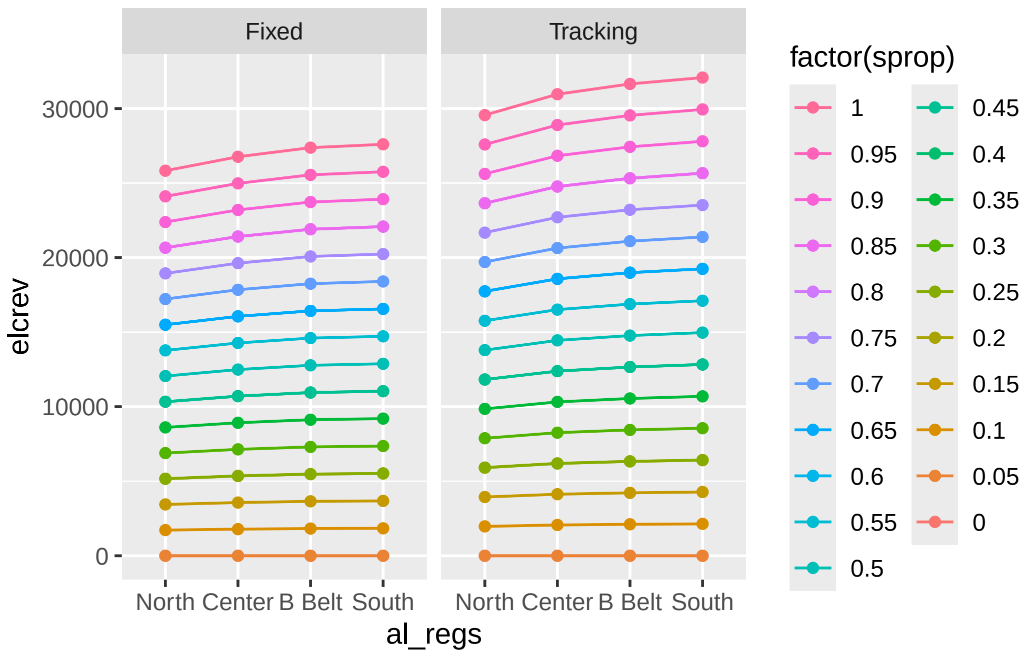
Electricity Price = 0.035



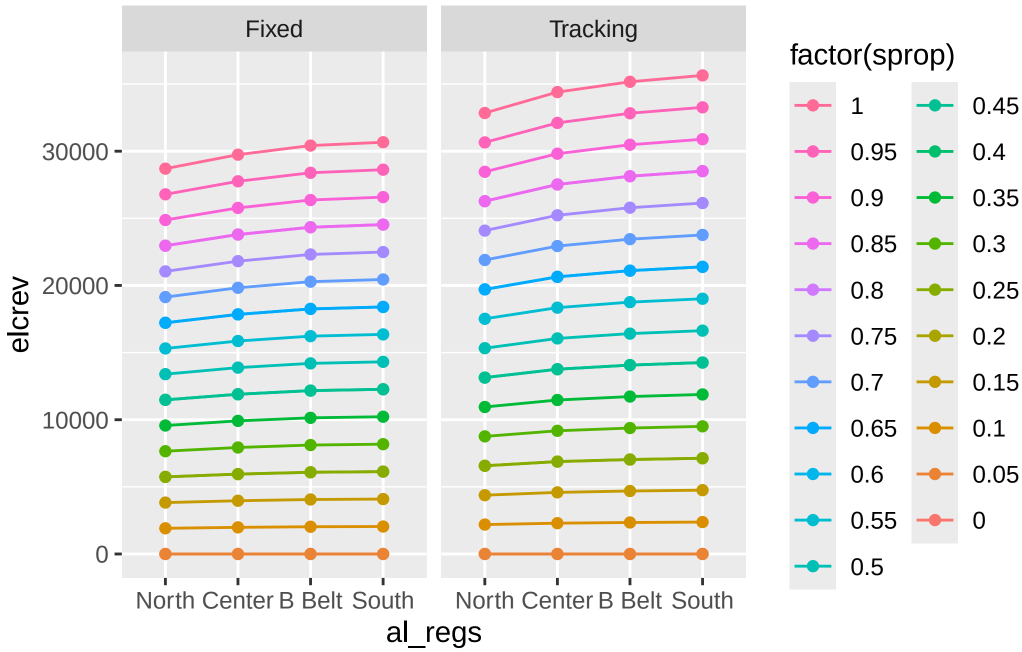
Electricity Price = 0.04



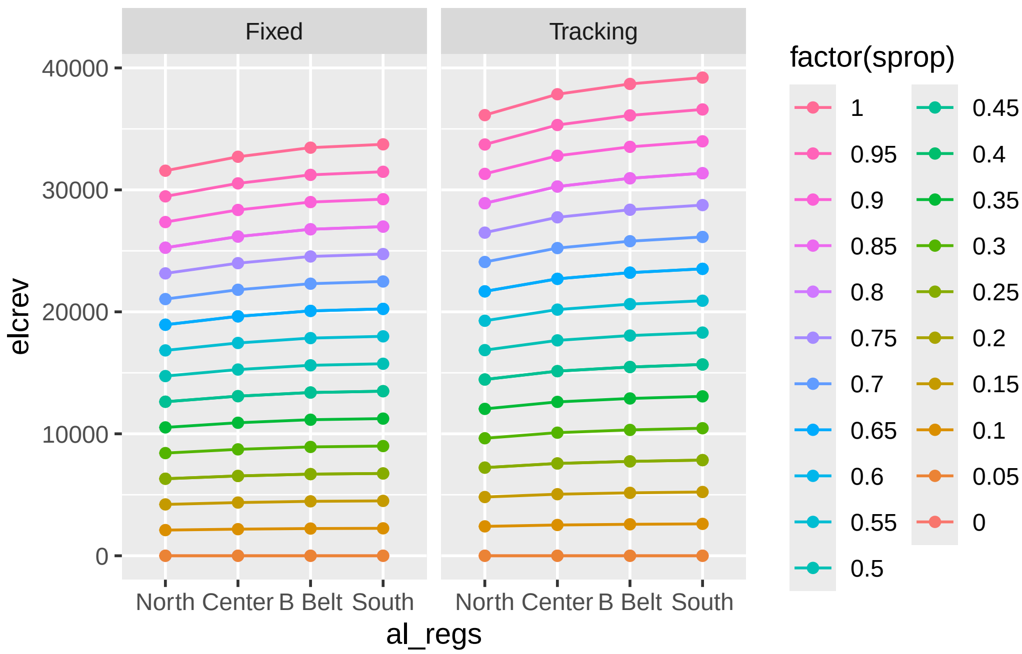

```
Electricity Price =  0.045
```



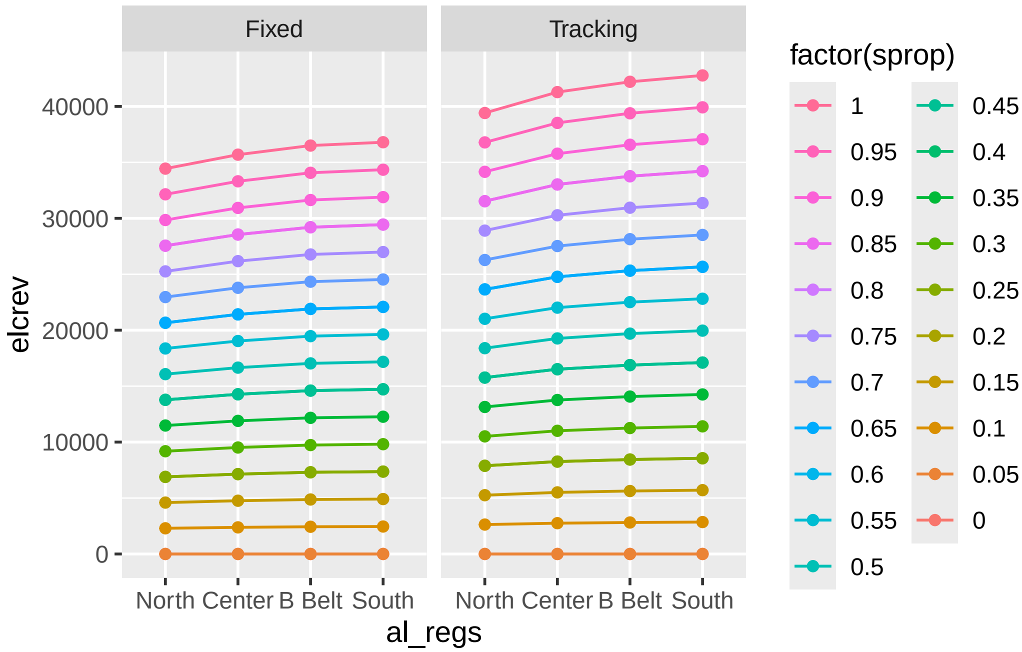

```
Electricity Price =  0.05
```




```
Electricity Price =  0.055
```




```
Electricity Price =  0.06
```



3.4 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 = 25/100  # 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
    )

write_xlsx(file = "Results/Solar Profit R25.xlsx",
```

```r
           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
head(solar_profit); tail(solar_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
  landlease ttlcost inscst recredit reap annlzcost annoftotcost monthlycost
```

```
1     1000       0      0        0    0         0            0           0
2     1000       0      0        0    0         0            0           0
3     1000       0      0        0    0         0            0           0
4     1000       0      0        0    0         0            0           0
5     1000       0      0        0    0         0            0           0
6     1000       0      0        0    0         0            0           0
  opex taxcr anncost eannprof eannprofworeap eannprofwoincentives
1    0     0       0        0              0                    0
2    0     0       0        0              0                    0
3    0     0       0        0              0                    0
4    0     0       0        0              0                    0
5    0     0       0        0              0                    0
6    0     0       0        0              0                    0

     sprop  al_regs    array  dc_kw panels energy elcprc   elcrev height
5539     1 Southern    Fixed 423.74    885 613342   0.06 36800.52    4.6
5540     1 Southern    Fixed 423.74    885 613342   0.06 36800.52    6.4
5541     1 Southern    Fixed 423.74    885 613342   0.06 36800.52    8.2
5542     1 Southern Tracking 423.74    885 712873   0.06 42772.38    4.6
5543     1 Southern Tracking 423.74    885 712873   0.06 42772.38    6.4
5544     1 Southern Tracking 423.74    885 712873   0.06 42772.38    8.2
        capex landlease   ttlcost   inscst recredit     reap annlzcost
5539 1.593333      1000  768504.5 3842.523 4048.057 192058.9  49794.83
5540 1.850000      1000  892301.2 4461.506 4048.057 222997.2  57816.17
5541 2.330000      1000 1123817.3 5619.086 4048.057 280856.0  72817.12
5542 1.733333      1000  836030.0 4180.150 4704.962 208934.3  54170.10
5543 1.921667      1000  926867.9 4634.339 4704.962 231635.9  60055.89
5544 2.110000      1000 1017705.8 5088.529 4704.962 254337.4  65941.68
     annoftotcost monthlycost     opex    taxcr  anncost  eannprof
5539     65945.77    2194.929 1978.373 19783.73 51773.20  8859.109
5540     76568.83    2548.506 2297.065 22970.65 60113.23  3705.994
5541     96435.34    3209.740 2893.060 28930.60 75710.18 -5931.001
5542     71740.17    2387.789 2152.205 21522.05 56322.31 12677.084
5543     79535.01    2647.232 2386.050 23860.50 62441.94  8895.902
5544     87329.86    2906.674 2619.896 26198.96 68561.58  5114.720
     eannprofworeap eannprofwoincentives
5539      -5313.461            -29145.25
5540     -12749.605            -39768.31
5541     -26656.160            -59634.82
5542      -2740.775            -28967.79
5543      -8197.166            -36762.63
5544     -13653.558            -44557.48
```

3.5 Profit from Solar

```
pf_solar_r25 <- solar_profit %>%
  filter(sprop == 1,
         elcprc == 0.04) %>%
  select(al_regs, array, height, eannprof, eannprofworeap)
cat("Maximum profit from solar at 100% PVD at 25% REAP = ",
    max(pf_solar_r25$eannprof),
    fill = TRUE)
```

```
Maximum profit from solar at 100% PVD at 25% REAP =  -1580.376
```

```
pf_solar_r25[which.max(pf_solar_r25$eannprof),]
```

```
    al_regs    array height  eannprof eannprofworeap
22 Southern Tracking    4.6 -1580.376      -16998.23
```

```
cat("Minimum profit from solar at 100% PVD at 25% REAP = ",
    min(pf_solar_r25$eannprof),
    fill = TRUE)
```

```
Minimum profit from solar at 100% PVD at 25% REAP =  -20030.25
```

```
pf_solar_r25[which.min(pf_solar_r25$eannprof),]
```

```
    al_regs array height  eannprof eannprofworeap
15 Northern Fixed    8.2 -20030.25      -40755.41
```

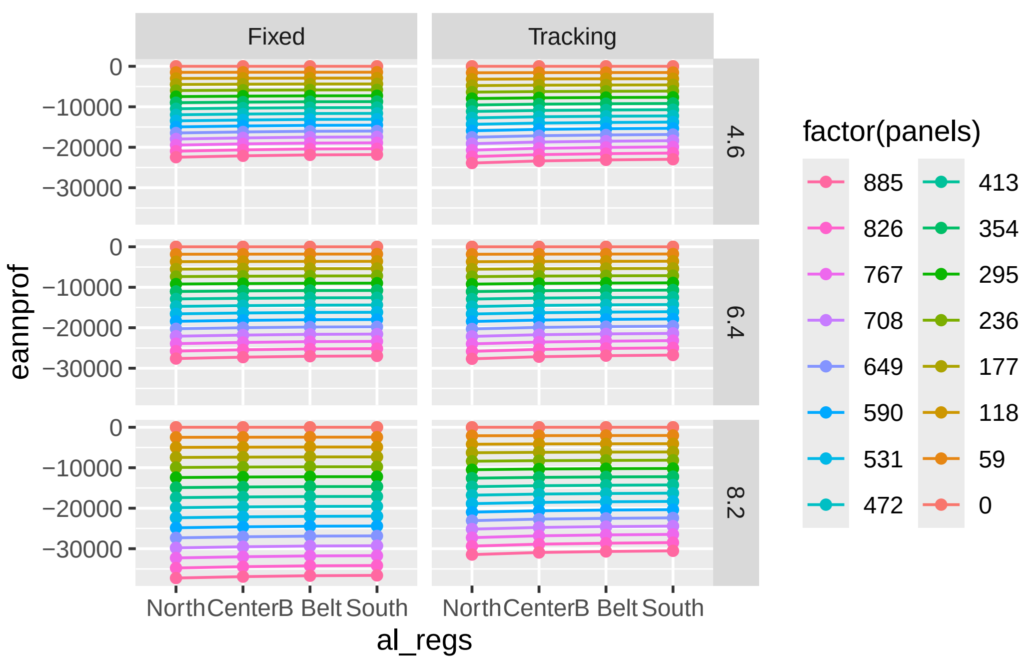
3.5.1 Plot Solar profit

Solar annual profit by number of solar panels

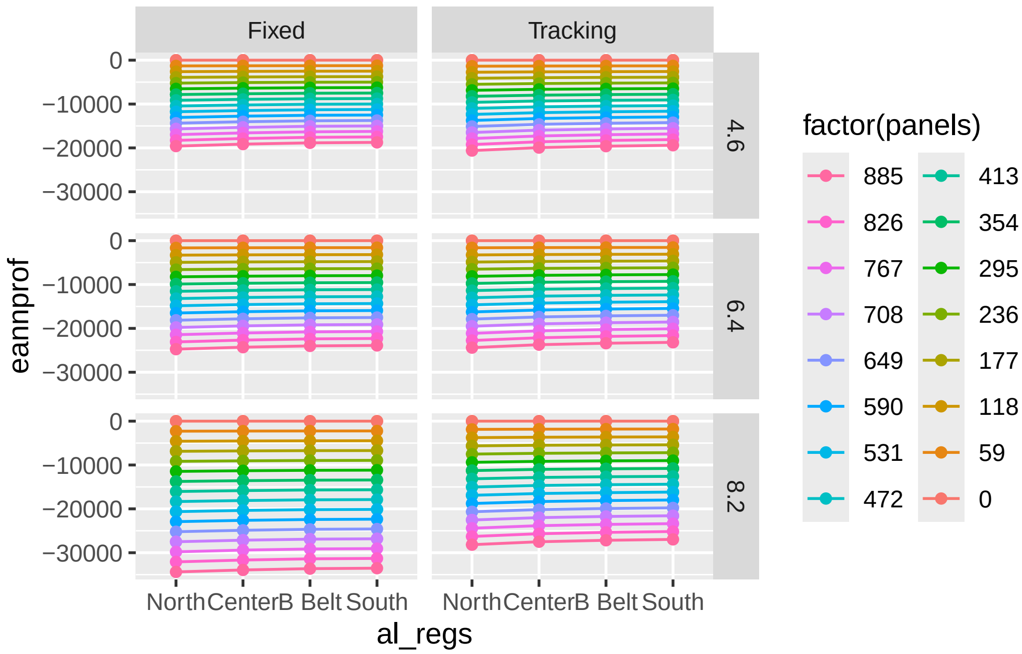
```
lox <- c("Northern", "Central", "Black Belt", "Southern")
array_levs = c("Single Axis Rotation", "Fixed Open Rack")
datalot_levs = c("Location 1", "Location 2")
  for (i in unique(solar_profit$elcprc)) {
    b = ggplot(
      data = (solar_profit %>%
```

```
                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)
  }
```

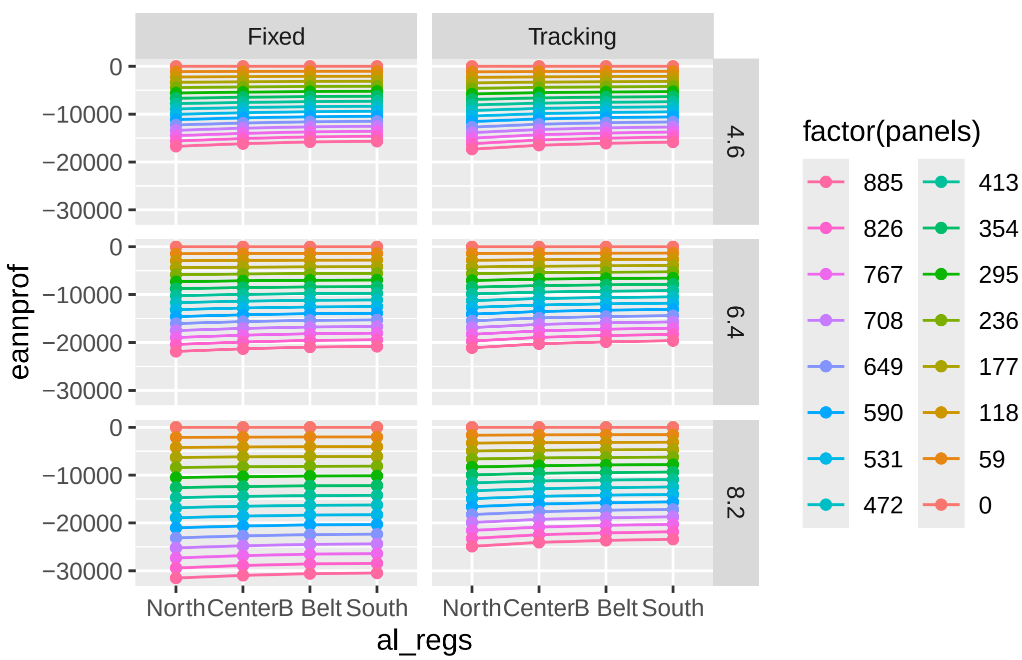
```
Electricity Price =  0.01
```



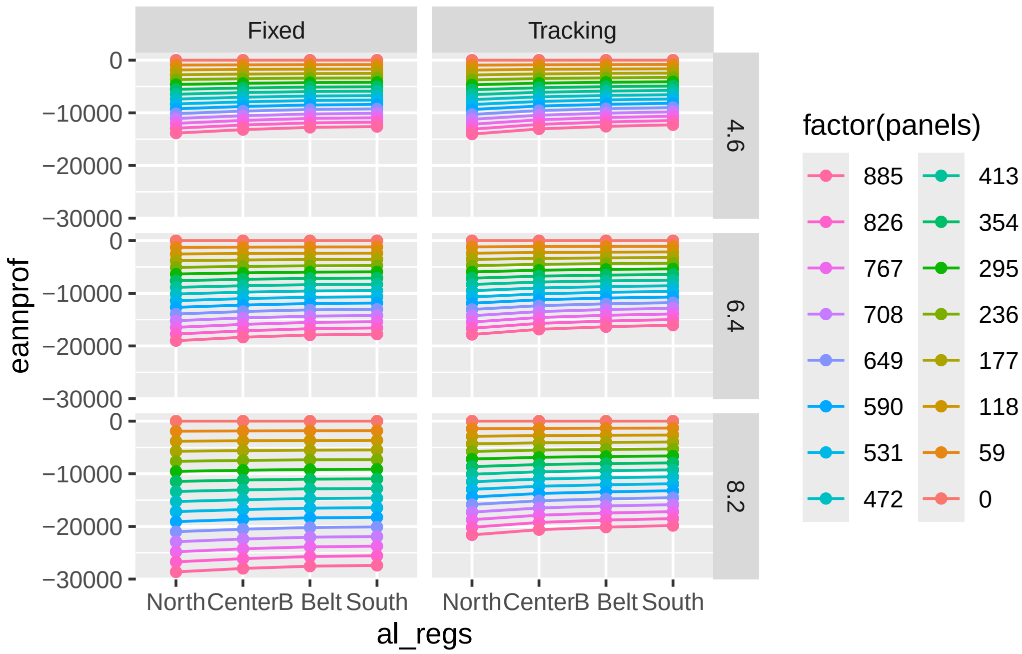
```
Electricity Price =  0.015
```



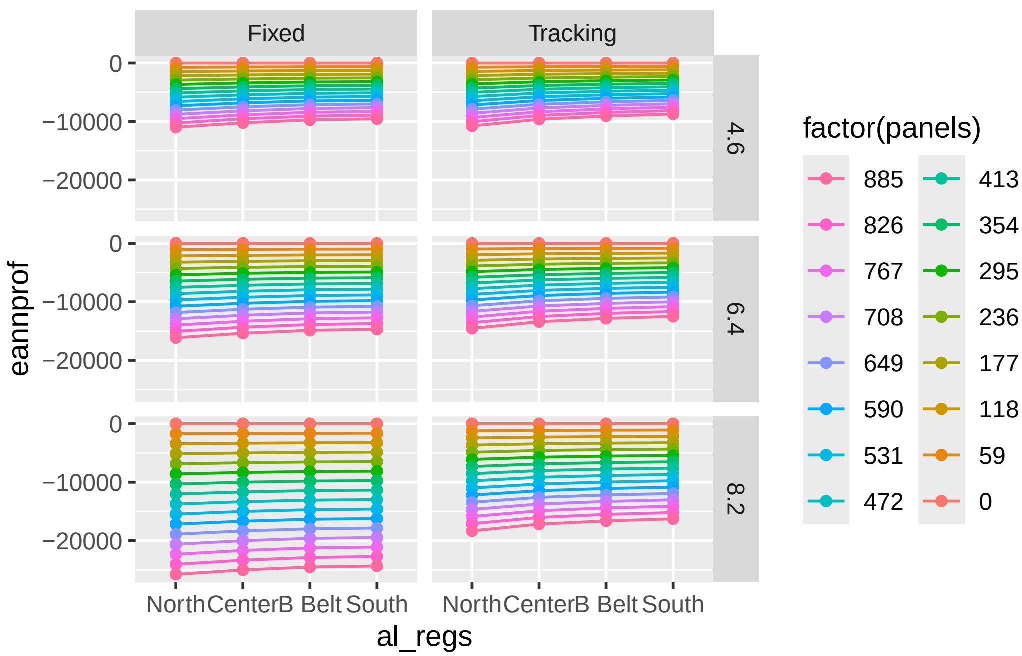

```
Electricity Price =  0.02
```



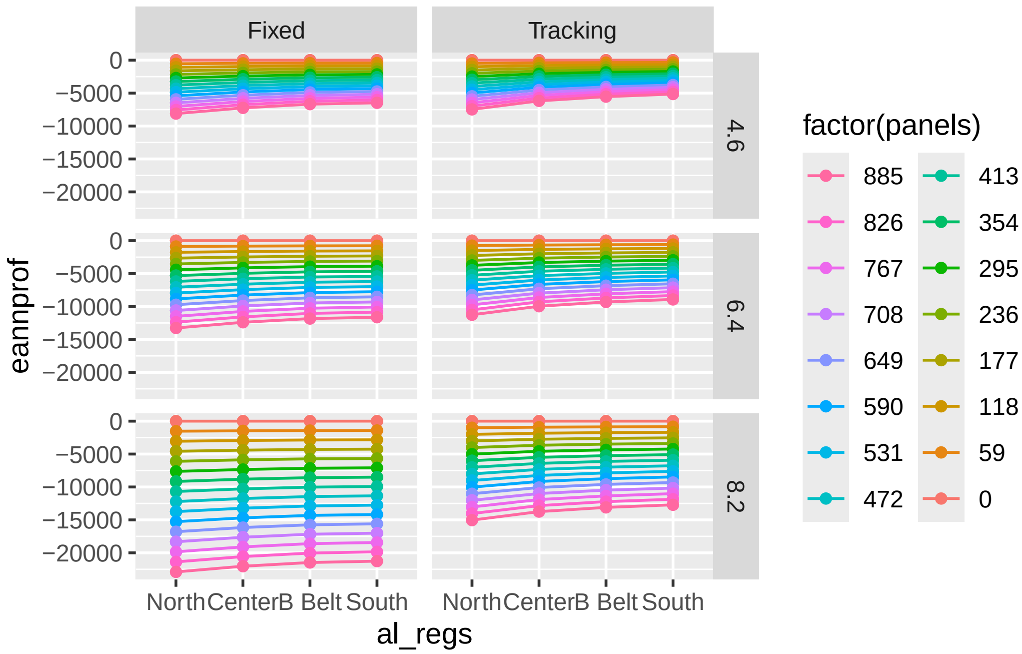
Electricity Price = 0.025



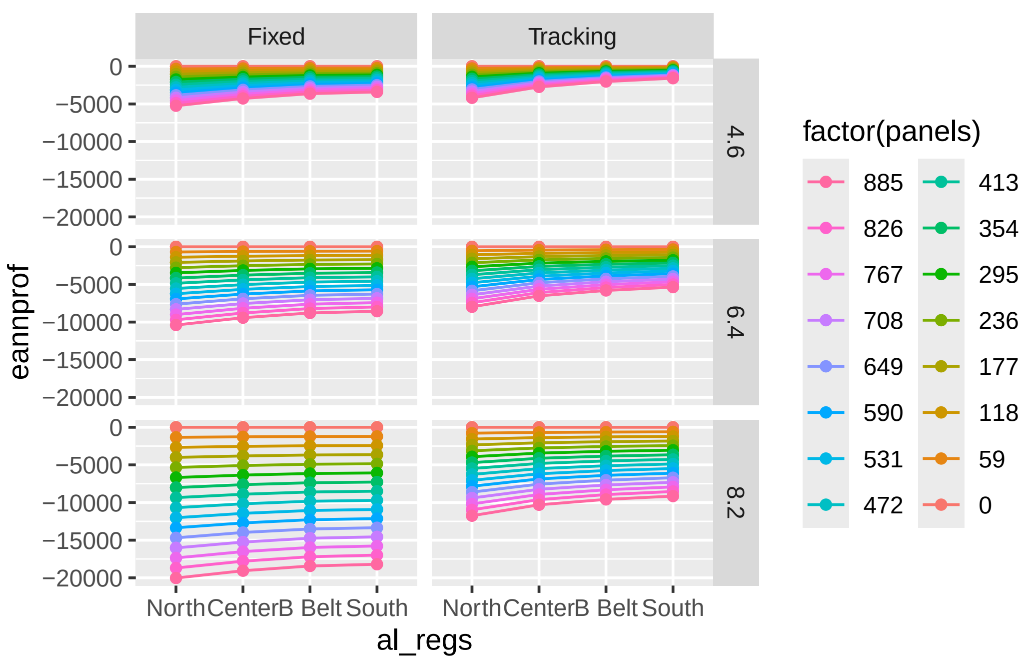
Electricity Price = 0.03



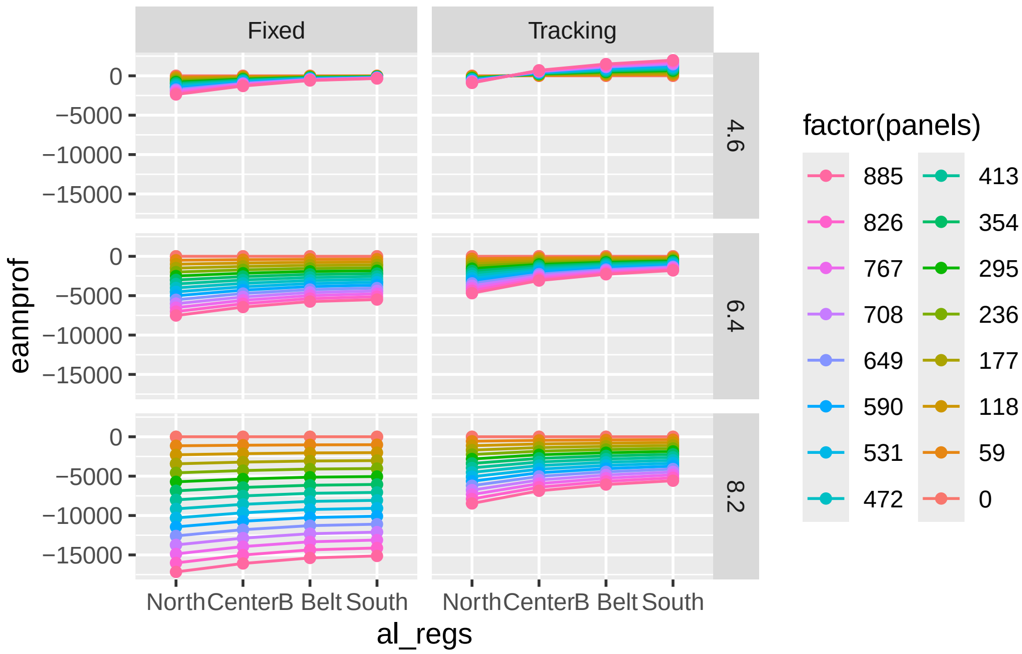
```
Electricity Price =  0.035
```



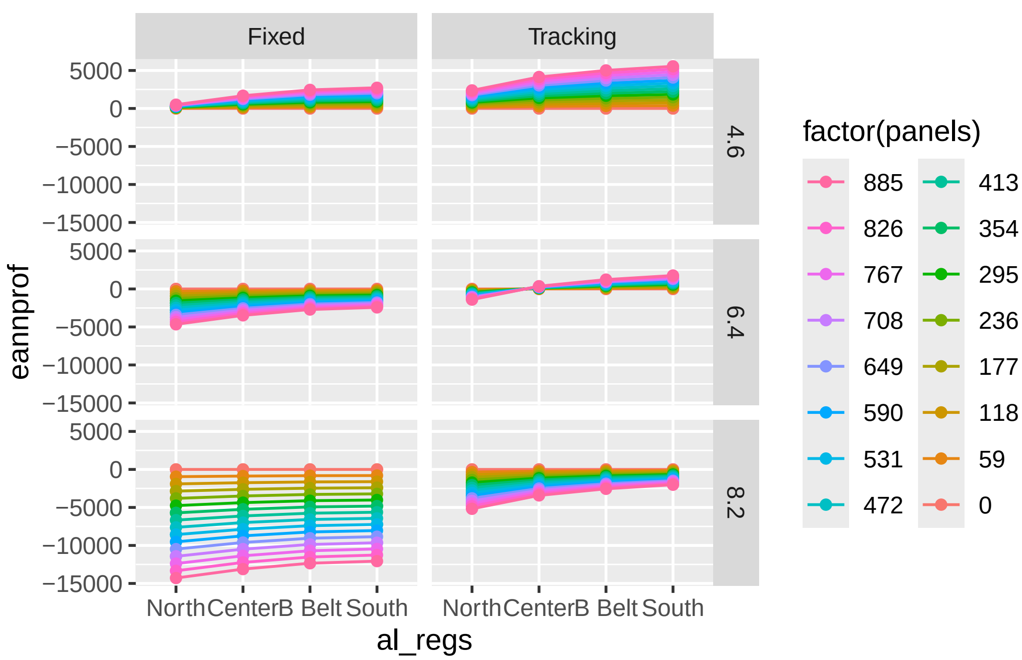

```
Electricity Price =  0.04
```



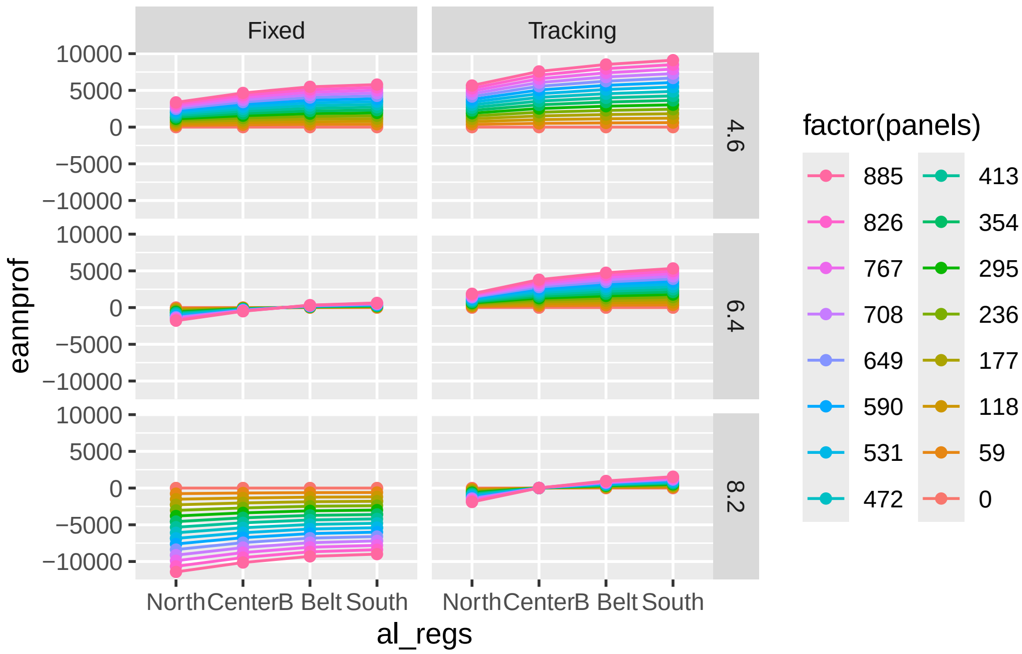
Electricity Price = 0.045



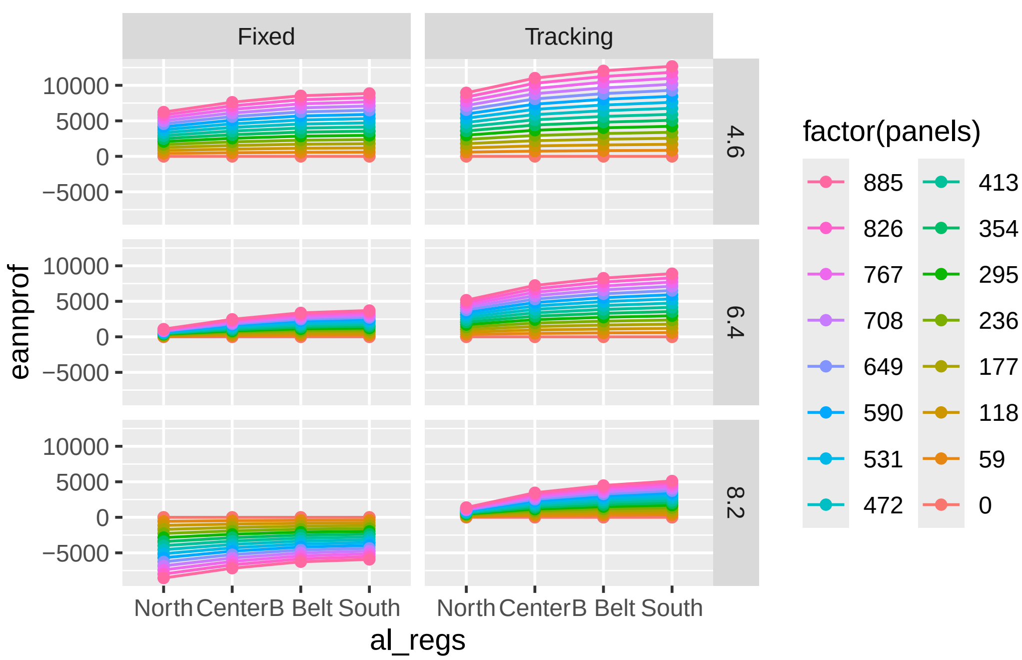
Electricity Price = 0.05



Electricity Price = 0.055



Electricity Price = 0.06



4 Profit from crops

4.1 Tomato

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

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

```
[1] 21  9
```

```
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")

# 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 ...
```

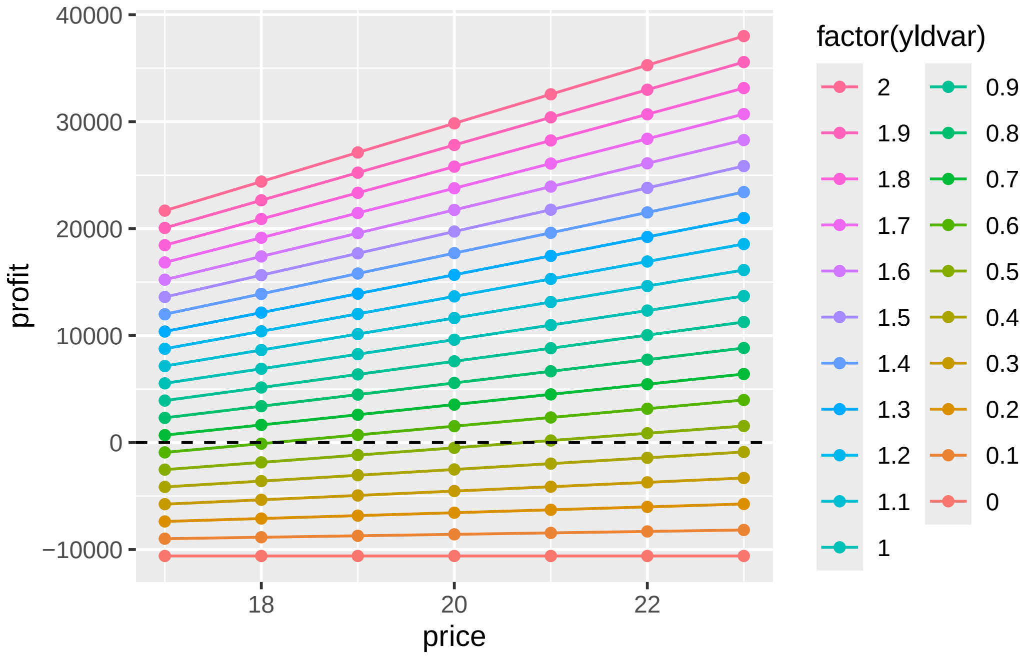
```
head(tomato_long); tail(tomato_long)
```

```
  yldvar yield price   profit
1    2.0  2720    17 21679.38
2    1.9  2584    17 20065.38
3    1.8  2448    17 18451.38
4    1.7  2312    17 16837.38
5    1.6  2176    17 15223.38
6    1.5  2040    17 13609.38
```

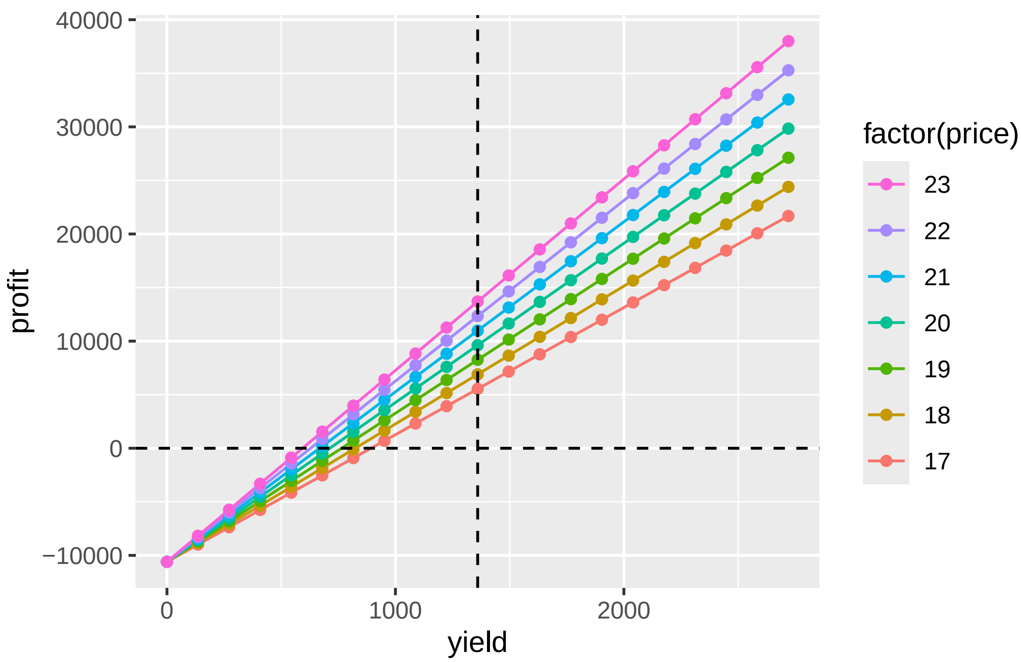
```
    yldvar yield price      profit
142    0.5   680    23   1549.3826
143    0.4   544    23   -880.6174
144    0.3   408    23  -3310.6174
145    0.2   272    23  -5740.6174
146    0.1   136    23  -8170.6174
147    0.0     0    23 -10600.6174
```

4.1.1 Plot Tomato Profit

```
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))
```




```
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 ...
```

```r
strawberry_profit
```

```
   yldvar yield      rolac3     rolac4       rolac5     rolac6       rolac7
3     2.0  6150     719.205   6869.205  13019.20503  19169.205  25319.20503
4     1.9  5843     143.288   5986.288  11829.28801  17672.288  23515.28801
5     1.8  5535    -434.505   5100.495  10635.49503  16170.495  21705.49503
6     1.7  5228   -1010.422   4217.578   9445.57801  14673.578  19901.57801
7     1.6  4920   -1588.215   3331.785   8251.78503  13171.785  18091.78503
8     1.5  4613   -2164.132   2448.868   7061.86801  11674.868  16287.86801
9     1.4  4305   -2741.925   1563.075   5868.07503  10173.075  14478.07503
10    1.3  3998   -3317.842    680.158   4678.15801   8676.158  12674.15801
11    1.2  3690   -3895.635   -205.635   3484.36503   7174.365  10864.36503
12    1.1  3383   -4471.552  -1088.552   2294.44801   5677.448   9060.44801
13    1.0  3075   -5049.345  -1974.345   1100.65503   4175.655   7250.65503
14    0.9  2768   -5625.262  -2857.262    -89.26199   2678.738   5446.73801
15    0.8  2460   -6203.055  -3743.055  -1283.05497   1176.945   3636.94503
16    0.7  2153   -6778.972  -4625.972  -2472.97199   -319.972   1833.02801
17    0.6  1845   -7356.765  -5511.765  -3666.76497  -1821.765     23.23503
18    0.5  1538   -7932.682  -6394.682  -4856.68199  -3318.682  -1780.68199
19    0.4  1230   -8510.475  -7280.475  -6050.47497  -4820.475  -3590.47497
20    0.3   923   -9086.392  -8163.392  -7240.39199  -6317.392  -5394.39199
21    0.2   615   -9664.185  -9049.185  -8434.18497  -7819.185  -7204.18497
22    0.1   308  -10240.102  -9932.102  -9624.10199  -9316.102  -9008.10199
23    0.0     0  -10817.895 -10817.895 -10817.89497 -10817.895 -10817.89497
       rolac8     rolac9
3   31469.205  37619.205
4   29358.288  35201.288
5   27240.495  32775.495
6   25129.578  30357.578
7   23011.785  27931.785
8   20900.868  25513.868
9   18783.075  23088.075
10  16672.158  20670.158
11  14554.365  18244.365
12  12443.448  15826.448
13  10325.655  13400.655
14   8214.738  10982.738
15   6096.945   8556.945
16   3986.028   6139.028
17   1868.235   3713.235
18   -242.682   1295.318
19  -2360.475  -1130.475
```

```
20  -4471.392  -3548.392
21  -6589.185  -5974.185
22  -8700.102  -8392.102
23 -10817.895 -10817.895
```

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 ...
```

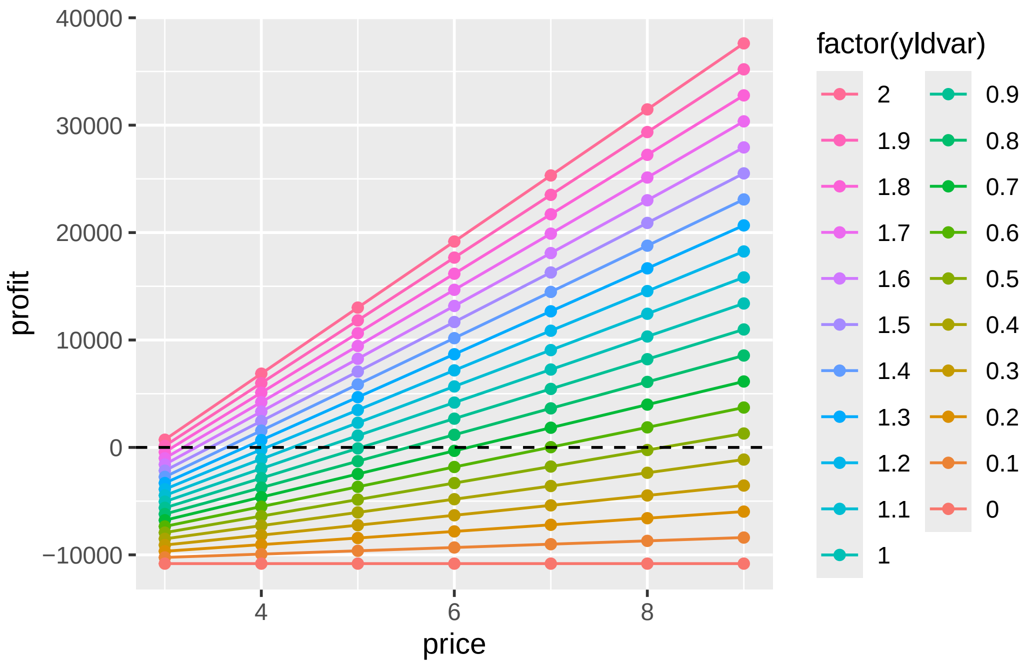
```r
head(stberry_long); tail(stberry_long)
```

```
  yldvar yield price    profit
1    2.0  6150     3   719.205
2    1.9  5843     3   143.288
3    1.8  5535     3  -434.505
4    1.7  5228     3 -1010.422
5    1.6  4920     3 -1588.215
6    1.5  4613     3 -2164.132
```

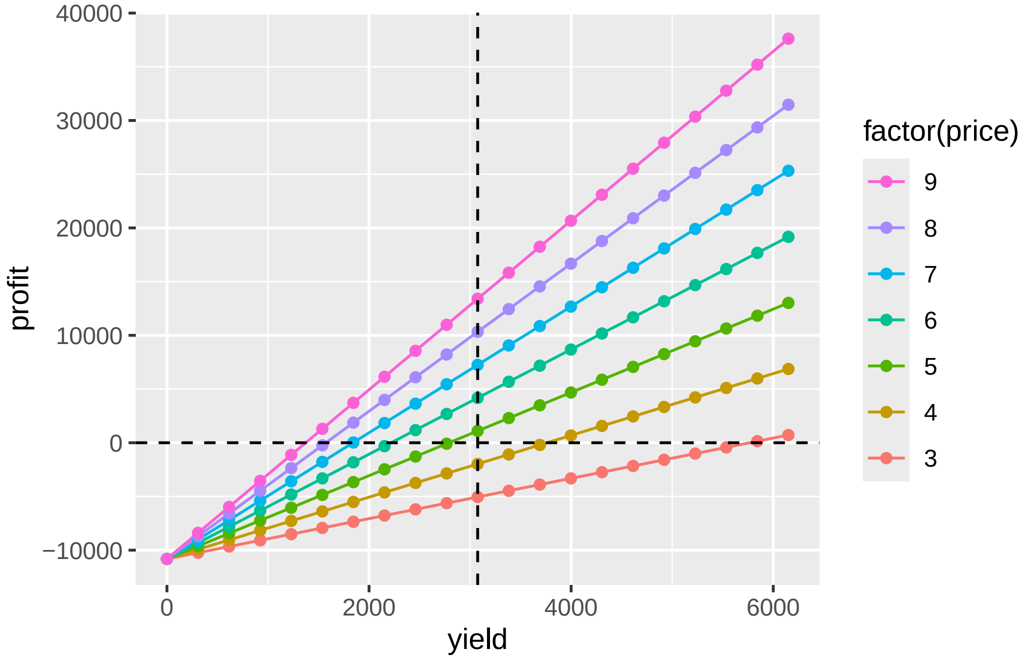
```
    yldvar yield price      profit
142    0.5  1538     9   1295.318
143    0.4  1230     9  -1130.475
144    0.3   923     9  -3548.392
145    0.2   615     9  -5974.185
146    0.1   308     9  -8392.102
147    0.0     0     9 -10817.895
```

4.2.1 Plot Strawberry Profit

```
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))
```



```
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 <- 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))
head(squash_profit); tail(squash_profit)
```

```
  yldvar yield   rolac11   rolac12  rolac13  rolac14  rolac15  rolac16  rolac17
3    2.0  2180 10309.117 12489.117 14669.12 16849.12 19029.12 21209.12 23389.12
4    1.9  2071  9607.367 11678.367 13749.37 15820.37 17891.37 19962.37 22033.37
5    1.8  1962  8905.617 10867.617 12829.62 14791.62 16753.62 18715.62 20677.62
6    1.7  1853  8203.867 10056.867 11909.87 13762.87 15615.87 17468.87 19321.87
7    1.6  1744  7502.117  9246.117 10990.12 12734.12 14478.12 16222.12 17966.12
8    1.5  1635  6800.367  8435.367 10070.37 11705.37 13340.37 14975.37 16610.37

   yldvar yield   rolac11   rolac12     rolac13   rolac14   rolac15     rolac16
18    0.5   545  -217.133   327.867   872.86702  1417.867  1962.867  2507.86702
19    0.4   436  -918.883  -482.883   -46.88298   389.117   825.117  1261.11702
20    0.3   327 -1620.633 -1293.633  -966.63298  -639.633  -312.633    14.36702
21    0.2   218 -2322.383 -2104.383 -1886.38298 -1668.383 -1450.383 -1232.38298
22    0.1   109 -3024.133 -2915.133 -2806.13298 -2697.133 -2588.133 -2479.13298
23    0.0     0 -3725.883 -3725.883 -3725.88298 -3725.883 -3725.883 -3725.88298
     rolac17
18  3052.867
19  1697.117
20   341.367
21 -1014.383
22 -2370.133
23 -3725.883
```

```
head(squash_long); tail(squash_long)
```

```
  yldvar yield price    profit
```

```
1    2.0  2180     11 10309.117
2    1.9  2071     11  9607.367
3    1.8  1962     11  8905.617
4    1.7  1853     11  8203.867
5    1.6  1744     11  7502.117
6    1.5  1635     11  6800.367

    yldvar yield price    profit
142    0.5   545    17  3052.867
143    0.4   436    17  1697.117
144    0.3   327    17   341.367
145    0.2   218    17 -1014.383
146    0.1   109    17 -2370.133
147    0.0     0    17 -3725.883
```

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)
```

```r
# 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))

# Create a new variable
tav_profit <- tav_profit %>%
  group_by(price) %>%  # Control for unique prices
  mutate(
    tavp_ge_t = if_else(yldvar == 1 & tav_profit >= profit, 1, 0)
  ) %>%
  ungroup()

# Check the result
str(tav_profit)
```

```
tibble [814,968 x 30] (S3: tbl_df/tbl/data.frame)
 $ sprop                : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ al_regs              : chr [1:814968] "Black Belt" "Black Belt" "Black Belt" "Black Belt"
 $ array                : chr [1:814968] "Fixed" "Fixed" "Fixed" "Fixed" ...
 $ dc_kw                : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ panels               : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ energy               : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ elcprc               : num [1:814968] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ..
 $ elcrev               : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ height               : num [1:814968] 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 ...
 $ capex                : num [1:814968] 1.59 1.59 1.59 1.59 1.59 ...
 $ landlease            : num [1:814968] 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 ..
 $ ttlcost              : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ inscst               : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ recredit             : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ reap                 : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ annlzcost            : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ annoftotcost         : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ monthlycost          : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ opex                 : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ taxcr                : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ anncost              : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ eannprof             : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ eannprofworeap       : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ eannprofwoincentives: num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ yldvar               : num [1:814968] 2 1.9 1.8 1.7 1.6 1.5 1.4 1.3 1.2 1.1 ...
 $ yield                : num [1:814968] 2720 2584 2448 2312 2176 ...
```

```
 $ price               : num [1:814968] 17 17 17 17 17 17 17 17 17 17 ...
 $ profit              : num [1:814968] 21679 20065 18451 16837 15223 ...
 $ tav_profit          : num [1:814968] 21679 20065 18451 16837 15223 ...
 $ tavp_ge_t           : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
```

```
head(tav_profit); tail(tav_profit)
```

```
# A tibble: 6 x 30
  sprop al_regs   array dc_kw panels energy elcprc elcrev height capex landlease
  <dbl> <chr>     <chr> <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>     <dbl>
1     0 Black Be~ Fixed     0      0      0   0.01      0    4.6  1.59      1000
2     0 Black Be~ Fixed     0      0      0   0.01      0    4.6  1.59      1000
3     0 Black Be~ Fixed     0      0      0   0.01      0    4.6  1.59      1000
4     0 Black Be~ Fixed     0      0      0   0.01      0    4.6  1.59      1000
5     0 Black Be~ Fixed     0      0      0   0.01      0    4.6  1.59      1000
6     0 Black Be~ Fixed     0      0      0   0.01      0    4.6  1.59      1000
# i 19 more variables: ttlcost <dbl>, inscst <dbl>, recredit <dbl>, reap <dbl>,
#   annlzcost <dbl>, annoftotcost <dbl>, monthlycost <dbl>, opex <dbl>,
#   taxcr <dbl>, anncost <dbl>, eannprof <dbl>, eannprofworeap <dbl>,
#   eannprofwoincentives <dbl>, yldvar <dbl>, yield <dbl>, price <dbl>,
#   profit <dbl>, tav_profit <dbl>, tavp_ge_t <dbl>

# A tibble: 6 x 30
  sprop al_regs  array  dc_kw panels energy elcprc elcrev height capex landlease
  <dbl> <chr>    <chr>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>     <dbl>
1     1 Southern Track~  424.    885 712873   0.06 42772.    8.2  2.11      1000
2     1 Southern Track~  424.    885 712873   0.06 42772.    8.2  2.11      1000
3     1 Southern Track~  424.    885 712873   0.06 42772.    8.2  2.11      1000
4     1 Southern Track~  424.    885 712873   0.06 42772.    8.2  2.11      1000
5     1 Southern Track~  424.    885 712873   0.06 42772.    8.2  2.11      1000
6     1 Southern Track~  424.    885 712873   0.06 42772.    8.2  2.11      1000
# i 19 more variables: ttlcost <dbl>, inscst <dbl>, recredit <dbl>, reap <dbl>,
#   annlzcost <dbl>, annoftotcost <dbl>, monthlycost <dbl>, opex <dbl>,
#   taxcr <dbl>, anncost <dbl>, eannprof <dbl>, eannprofworeap <dbl>,
#   eannprofwoincentives <dbl>, yldvar <dbl>, yield <dbl>, price <dbl>,
#   profit <dbl>, tav_profit <dbl>, tavp_ge_t <dbl>
```

```
# TAV Profit Greater or Equal to Tomato
tavp_ge_tomato = tav_profit %>% filter(tavp_ge_t == 1)
```

5.1.1 Saving results locally

```
write_feather(tav_profit,
  sink = "Data/tav_profit R25.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 R25.xlsx",
           as_table = TRUE)
tictoc::toc()
```

```
Using Dplyr:: 0.07 sec elapsed
```

```
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/Profit TAV WriteUp R25.xlsx",
           as_table = TRUE)
```

```
write_xlsx(
  x = tavp_ge_tomato %>%
    dplyr::filter(tavp_ge_t == 1) %>%
    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/Profit TAV GE Tomato R25.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.

```
# 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)
sbav_profit <- data.frame(lapply(sbav_profit, unlist))
# Create the new variable

sbav_profit <- sbav_profit %>%
```

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

str(sbav_profit)
```

```
tibble [814,968 x 30] (S3: tbl_df/tbl/data.frame)
 $ sprop               : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ al_regs             : chr [1:814968] "Black Belt" "Black Belt" "Black Belt" "Black Belt"
 $ array               : chr [1:814968] "Fixed" "Fixed" "Fixed" "Fixed" ...
 $ dc_kw               : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ panels              : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ energy              : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ elcprc              : num [1:814968] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ..
 $ elcrev              : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ height              : num [1:814968] 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 4.6 ...
 $ capex               : num [1:814968] 1.59 1.59 1.59 1.59 1.59 ...
 $ landlease           : num [1:814968] 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 ..
 $ ttlcost             : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ inscst              : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ recredit            : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ reap                : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ annlzcost           : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ annoftotcost        : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ monthlycost         : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ opex                : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ taxcr               : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ anncost             : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ eannprof            : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ eannprofworeap      : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ eannprofwoincentives: num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
 $ yldvar              : num [1:814968] 2 1.9 1.8 1.7 1.6 1.5 1.4 1.3 1.2 1.1 ...
 $ yield               : num [1:814968] 6150 5843 5535 5228 4920 ...
 $ price               : num [1:814968] 3 3 3 3 3 3 3 3 3 3 ...
 $ profit              : num [1:814968] 719 143 -435 -1010 -1588 ...
 $ sbav_profit         : num [1:814968] 719 143 -435 -1010 -1588 ...
 $ sbavp_ge_sb         : num [1:814968] 0 0 0 0 0 0 0 0 0 0 ...
```

```r
head(sbav_profit); tail(sbav_profit)
```

```
# A tibble: 6 x 30
  sprop al_regs   array dc_kw panels energy elcprc elcrev height capex landlease
  <dbl> <chr>     <chr> <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>     <dbl>
1     0 Black Be~ Fixed     0      0      0   0.01      0    4.6  1.59      1000
2     0 Black Be~ Fixed     0      0      0   0.01      0    4.6  1.59      1000
3     0 Black Be~ Fixed     0      0      0   0.01      0    4.6  1.59      1000
4     0 Black Be~ Fixed     0      0      0   0.01      0    4.6  1.59      1000
5     0 Black Be~ Fixed     0      0      0   0.01      0    4.6  1.59      1000
6     0 Black Be~ Fixed     0      0      0   0.01      0    4.6  1.59      1000
# i 19 more variables: ttlcost <dbl>, inscst <dbl>, recredit <dbl>, reap <dbl>,
#   annlzcost <dbl>, annoftotcost <dbl>, monthlycost <dbl>, opex <dbl>,
#   taxcr <dbl>, anncost <dbl>, eannprof <dbl>, eannprofworeap <dbl>,
#   eannprofwoincentives <dbl>, yldvar <dbl>, yield <dbl>, price <dbl>,
#   profit <dbl>, sbav_profit <dbl>, sbavp_ge_sb <dbl>

# A tibble: 6 x 30
  sprop al_regs  array  dc_kw panels energy elcprc elcrev height capex landlease
  <dbl> <chr>    <chr>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>     <dbl>
1     1 Southern Track~  424.    885 712873   0.06 42772.    8.2  2.11      1000
2     1 Southern Track~  424.    885 712873   0.06 42772.    8.2  2.11      1000
3     1 Southern Track~  424.    885 712873   0.06 42772.    8.2  2.11      1000
4     1 Southern Track~  424.    885 712873   0.06 42772.    8.2  2.11      1000
5     1 Southern Track~  424.    885 712873   0.06 42772.    8.2  2.11      1000
6     1 Southern Track~  424.    885 712873   0.06 42772.    8.2  2.11      1000
# i 19 more variables: ttlcost <dbl>, inscst <dbl>, recredit <dbl>, reap <dbl>,
#   annlzcost <dbl>, annoftotcost <dbl>, monthlycost <dbl>, opex <dbl>,
#   taxcr <dbl>, anncost <dbl>, eannprof <dbl>, eannprofworeap <dbl>,
#   eannprofwoincentives <dbl>, yldvar <dbl>, yield <dbl>, price <dbl>,
#   profit <dbl>, sbav_profit <dbl>, sbavp_ge_sb <dbl>
```

```r
# TAV Profit Greater or Equal to Tomato
sbavp_ge_sberry = sbav_profit %>% filter(sbavp_ge_sb == 1)
```

5.2.1 Saving results locally

```r
#write_csv(sbav_profit, "tav_profit.csv")
write_feather(sbav_profit,
```

```
  sink = "Data/sbav_profit R25.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(tav_profit), 100),],
           file = "Results/SBAV Profit Sample R25.xlsx",
           as_table = TRUE)
tictoc::toc()
```

```
Using Base R Matrix:: 0.09 sec elapsed
```

```
write_xlsx(x = sbav_profit %>%
             filter(sprop %in% c(0, 0.25, 0.50, 0.75, 1),
                    yldvar == 1,
                    price == 9,
                    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/Profit SBAV WriteUp R25.xlsx",
           as_table = TRUE)
```

```
write_xlsx(
  x = sbavp_ge_sberry %>%
    dplyr::filter(sbavp_ge_sb == 1) %>%
    dplyr::select(
      sprop, panels, height, array, al_regs,
      yldvar, yield, price, elcprc, 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/Profit SBAV GE Strawberry R25.xlsx",
  as_table = TRUE
)
```

5.3 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 %>%
  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,
```

```r
  sink = "Data/sqav_profit R25.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 R25.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/Profit SQAV WriteUp R25.xlsx",
           as_table = TRUE)
write_xlsx(
  x = sqavp_ge_squash %>%
    dplyr::filter(sqavp_ge_sq == 1) %>%
    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/Profit SQAV GE Squash R25.xlsx",
```

```
  as_table = TRUE
)
```