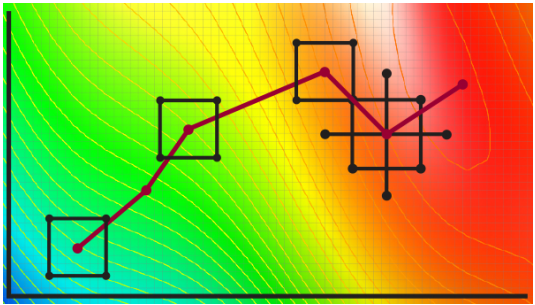
Experimentation for Improvement

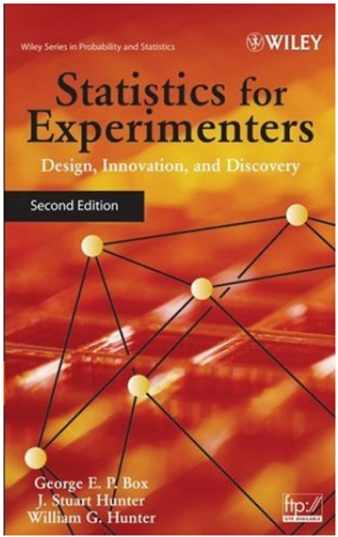


http://learnche.org/

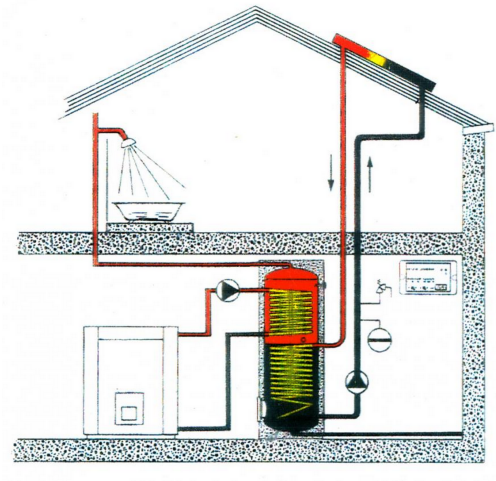
Design and Analysis of Experiments

Copyright, sharing, and attribution notice

Solar panel case study



(p. 230 in Box, Hunter and Hunter, 2nd ed)



[http://yint.org/solar-panel-study]

when experimenting with computer simulations

The same as regular experiments:

- you must follow a systematic method
- don't "play around" with the software: trial-and-error

Many experiments are simulations:

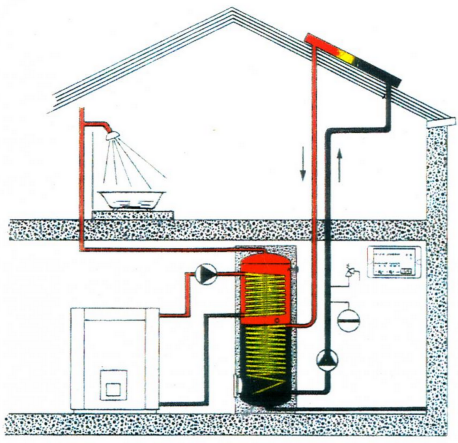
- bridge/building design
- chemical factory design
- improve traffic light timing and queuing
- test a stock market buy/sell strategy

Different to regular experiments:

1. We can often run computer simulations in parallel
2. Computer experiments (mostly*) are deterministic
   - i.e. if you repeat the experiments, you get the identical results
   - this indicates there are no disturbances that affect the outcome
   - this implies you do not need to randomize the order
   - or even repeat experiments!

* *except those that have a random component*

Solar panel case study



http://yint.org/solar-panel-study

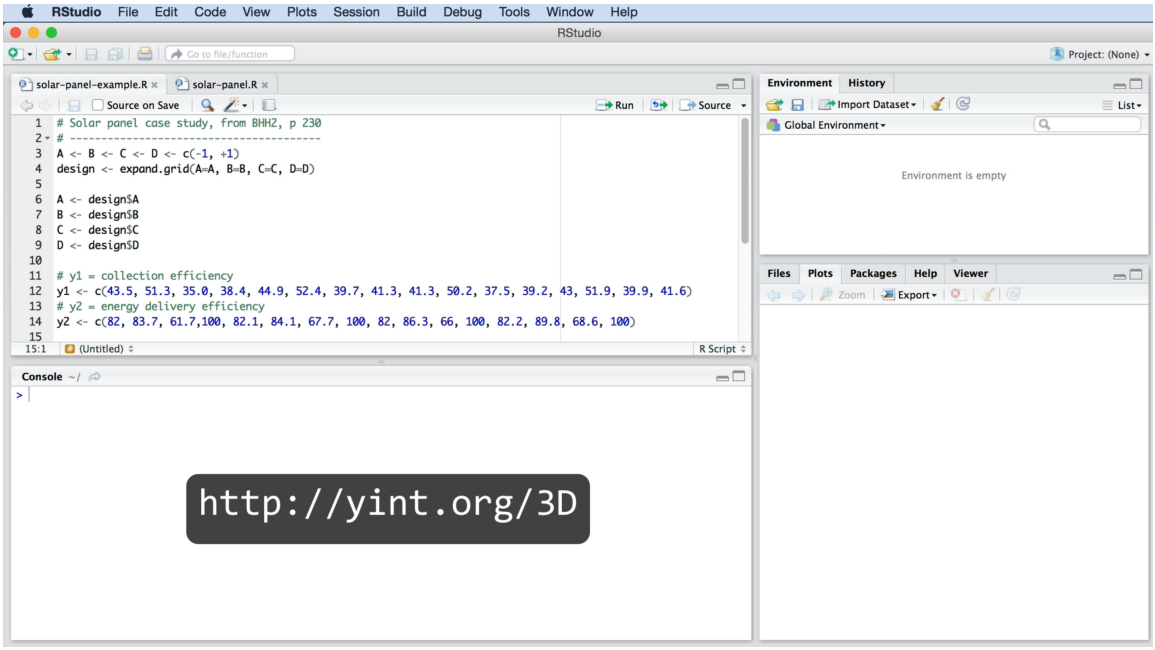
The factors are:

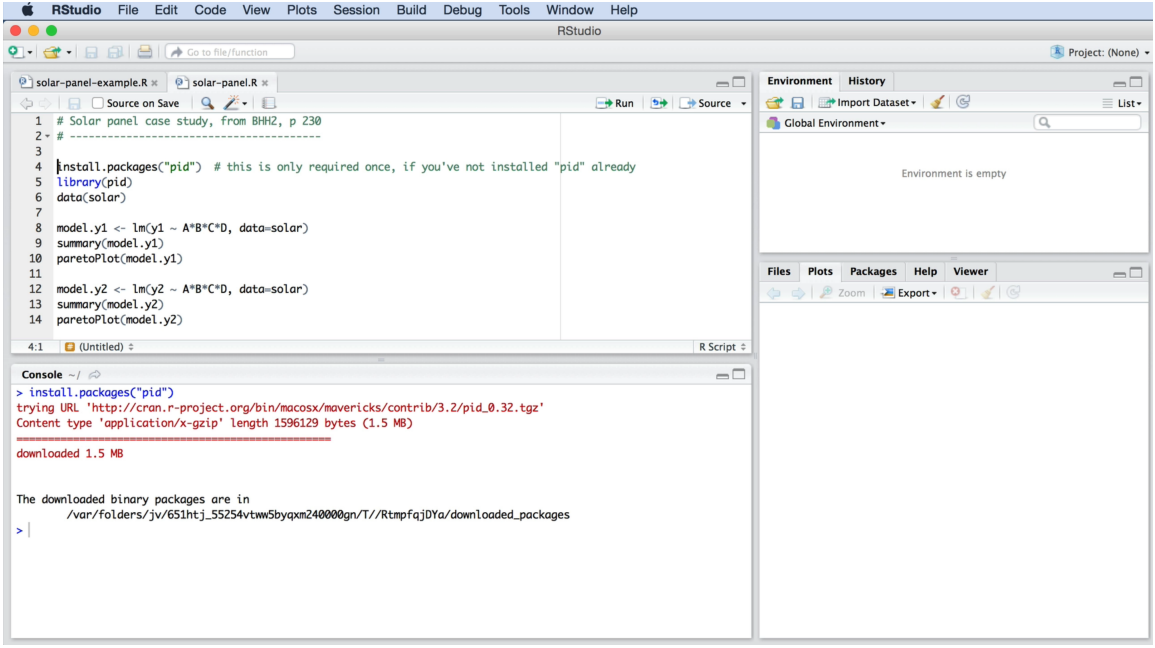
- **A** = total daily insolation (sunlight received)
- **B** = storage tank capacity
- **C** = water flow rate
- **D** = solar intermittency

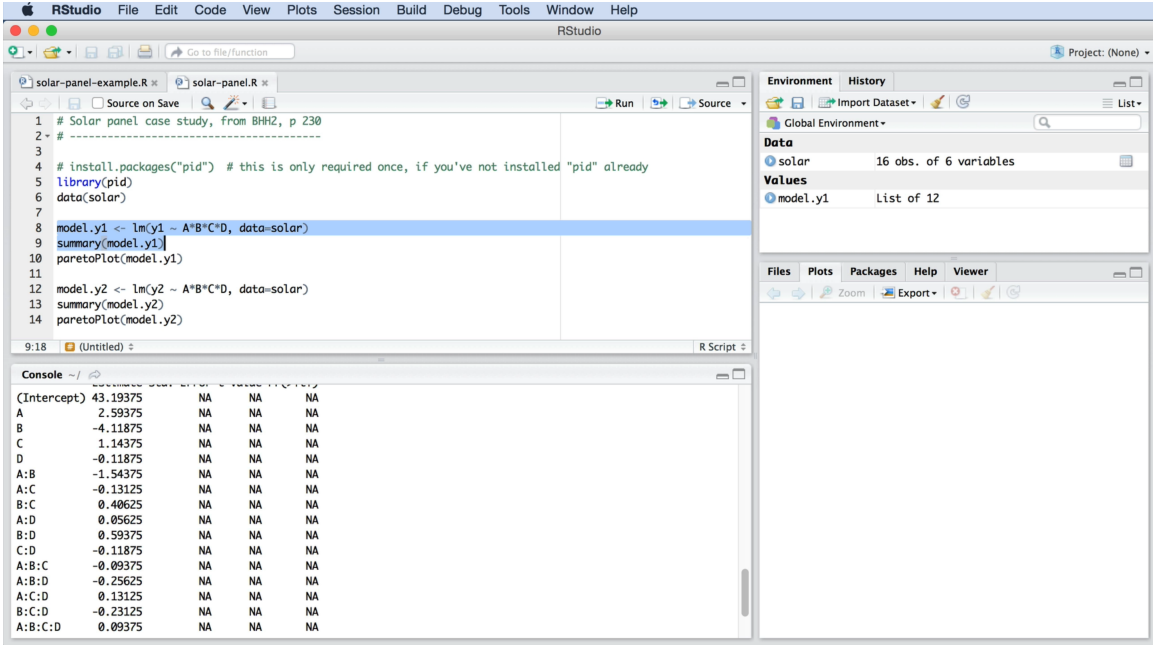
The outcome variables were:

- y_1 = collection efficiency
- y_2 = energy delivery efficiency

Total experiments required $= 2^4 = 16$


RStudio File Edit Code View Plots Session Build Debug Tools Window Help
RStudio
Go to file/function
Project: (None)
solar-panel-example.R
solar-panel.R
Source on Save
Run
Source
1 # Solar panel case study, from BHH2, p 230
2 # ----------
3 A <- B <- C <- D <- c(-1, +1)
4 design <- expand.grid(A=A, B=B, C=C, D=D)
5
6 A <- design$A
7 B <- design$B
8 C <- design$C
9 D <- design$D
10
11 # y1 = collection efficiency
12 y1 <- c(43.5, 51.3, 35.0, 38.4, 44.9, 52.4, 39.7, 41.3, 41.3, 50.2, 37.5, 39.2, 43, 51.9, 39.9, 41.6)
13 # y2 = energy delivery efficiency
14 y2 <- c(82, 83.7, 61.7,100, 82.1, 84.1, 67.7, 100, 82, 86.3, 66, 100, 82.2, 89.8, 68.6, 100)
15
15:1
(Untitled)
R Script
Console ~/
>
Environment
History
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
Zoom
Export
http://yint.org/3D


RStudio File Edit Code View Plots Session Build Debug Tools Window Help
RStudio
Go to file/function
Project: (None)
solar-panel-example.R
solar-panel.R
Source on Save
Run
Source
1 # Solar panel case study, from BHH2, p 230
2 # --
3
4 install.packages("pid") # this is only required once, if you've not installed "pid" already
5 library(pid)
6 data(solar)
7
8 model.y1 <- lm(y1 ~ A*B*C*D, data=solar)
9 summary(model.y1)
10 paretoPlot(model.y1)
11
12 model.y2 <- lm(y2 ~ A*B*C*D, data=solar)
13 summary(model.y2)
14 paretoPlot(model.y2)
4:1
(Untitled)
R Script
Console ~/
> install.packages("pid")
trying URL 'http://cran.r-project.org/bin/macosx/mavericks/contrib/3.2/pid_0.32.tgz'
Content type 'application/x-gzip' length 1596129 bytes (1.5 MB)
downloaded 1.5 MB
The downloaded binary packages are in
/var/folders/jv/651htj_55254vtww5byqxm240000gn/T//RtmpfqjDYa/downloaded_packages
>
Environment
History
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
Zoom
Export


RStudio File Edit Code View Plots Session Build Debug Tools Window Help
RStudio
Go to file/function
Project: (None)
solar-panel-example.R
solar-panel.R
Source on Save
Run
Source
1 # Solar panel case study, from BHH2, p 230
2 # ----------
3
4 # install.packages("pid") # this is only required once, if you've not installed "pid" already
5 library(pid)
6 data(solar)
7
8 model.y1 <- lm(y1 ~ A*B*C*D, data=solar)
9 summary(model.y1)
10 paretoPlot(model.y1)
11
12 model.y2 <- lm(y2 ~ A*B*C*D, data=solar)
13 summary(model.y2)
14 paretoPlot(model.y2)
9:18
(Untitled)
R Script
Console ~/
(Intercept) 43.19375 NA NA NA
A 2.59375 NA NA NA
B -4.11875 NA NA NA
C 1.14375 NA NA NA
D -0.11875 NA NA NA
A:B -1.54375 NA NA NA
A:C -0.13125 NA NA NA
B:C 0.40625 NA NA NA
A:D 0.05625 NA NA NA
B:D 0.59375 NA NA NA
C:D -0.11875 NA NA NA
A:B:C -0.09375 NA NA NA
A:B:D -0.25625 NA NA NA
A:C:D 0.13125 NA NA NA
B:C:D -0.23125 NA NA NA
A:B:C:D 0.09375 NA NA NA
Environment History
Import Dataset
List
Global Environment
Data
solar 16 obs. of 6 variables
Values
model.y1 List of 12
Files Plots Packages Help Viewer
Zoom
Export

```
 8  model.y1 <- lm(y1 ~ A*B*C*D, data=solar)
 9  summary(model.y1)
10  paretoPlot(model.y1)
11
12  model.y2 <- lm(y2 ~ A*B*C*D, data=solar)
13  summary(model.y2)
14  paretoPlot(model.y2)
```

8:42 (Untitled) R Script

Console ~/

```
> lm(y1 ~ A*B)

Call:
lm.default(formula = y1 ~ A * B)

Coefficients:
(Intercept)            A            B          A:B
     43.194        2.594       -4.119       -1.544

>
```

`lm(y ~ A*B)`

expands into: `lm(y ~ A + B + A*B)`

```
8  model.y1 <- lm(y1 ~ A*B*C*D, data=solar)
9  summary(model.y1)
10 paretoPlot(model.y1)
11
12 model.y2 <- lm(y2 ~ A*B*C*D, data=solar)
13 summary(model.y2)
14 paretoPlot(model.y2)
```

8:42 (Untitled) R Script

Console ~/

`lm(y ~ A*B*C)`
ultimately expands into: **`lm(y ~ A + B + C + A*B + A*C + B*C + A*B*C)`**

```
> lm(y1 ~ A*B*C)

Call:
lm.default(formula = y1 ~ A * B * C)

Coefficients:
(Intercept)            A            B            C          A:B          A:C          B:C        A:B:C
   43.19375      2.59375     -4.11875      1.14375     -1.54375     -0.13125      0.40625     -0.09375

>
```

```
 8  model.y1 <- lm(y1 ~ A*B*C*D, data=solar)
 9  summary(model.y1)
10  paretoPlot(model.y1)
11
12  model.y2 <- lm(y2 ~ A*B*C*D, data=solar)
13  summary(model.y2)
14  paretoPlot(model.y2)
```

8:42 (Untitled) R Script

Console ~/

```
Coefficients:
(Intercept)            A            B            C          A:B          A:C          B:C        A:B:C
   43.19375      2.59375     -4.11875      1.14375     -1.54375     -0.13125      0.40625     -0.09375

> lm(y1 ~ A*B*C*D)

Call:
lm.default(formula = y1 ~ A * B * C * D)

Coefficients:
(Intercept)            A            B            C            D          A:B          A:C          B:C
   43.19375      2.59375     -4.11875      1.14375     -0.11875     -1.54375     -0.13125      0.40625
        A:D          B:D          C:D        A:B:C        A:B:D        A:C:D        B:C:D      A:B:C:D
    0.05625      0.59375     -0.11875     -0.09375     -0.25625      0.13125     -0.23125      0.09375

>
```

```
 8  model.y1 <- lm(y1 ~ A*B*C*D, data=solar)
 9  summary(model.y1)
10  paretoPlot(model.y1)
11
12  model.y2 <- lm(y2 ~ A*B*C*D, data=solar)
13  summary(model.y2)
14  paretoPlot(model.y2)
```

8:42 (Untitled) R Script

Console ~/

```
Coefficients:
(Intercept)            A            B            C          A:B          A:C          B:C          A:B:C
   43.19375      2.59375     -4.11875      1.14375     -1.54375     -0.13125      0.40625     -0.09375

> lm(y1 ~ A*B*C*D)

Call:
lm.default(formula = y1 ~ A * B * C * D)

Coefficients:
(Intercept)            A            B            C            D          A:B          A:C          B:C
   43.19375      2.59375     -4.11875      1.14375     -0.11875     -1.54375     -0.13125      0.40625
        A:D          B:D          C:D        A:B:C        A:B:D        A:C:D        B:C:D        A:B:C:D
    0.05625      0.59375     -0.11875     -0.09375     -0.25625      0.13125     -0.23125      0.09375

> model.y1 <- lm(y1 ~ A + B + C + D + A*B + A*C + B*C + A*D + B*D + C*D + A*B*C + A*B*D + A*C*D + B*C*D + A*B*C*D, data=solar)
```

More details appear in the PDF slides

R's automatic expansion of model terms

`lm(y ~ A*B)`

expands into: `lm(y ~ A + B + A*B)`

`lm(y ~ A*B*C)`

ultimately expands into: `lm(y ~ A + B + C + A*B + A*C + B*C + A*B*C)`

This is because: `A*B*C` can be considered to be `(A*B)*C`
which expands into `(A + B + A*B)*C = A*C + B*C + A*B*C`

but `A*C` expands into `A + C + A*C`
and `B*C` expands into `B + C + B*C`
which when all collected together gives the full model expansion above.

RStudio File Edit Code View Plots Session Build Debug Tools Window Help

RStudio

Go to file/function

solar-panel-example.R | solar-panel.R

Source on Save

```
1  # Solar panel case study, from BHH2, p 230
2  # ----------------------------------------
3
4  # install.packages("pid")  # this is only required onc
5  library(pid)
6  data(solar)
7
8  model.y1 <- lm(y1 ~ A*B*C*D, data=solar)
9  summary(model.y1)
10 paretoPlot(model.y1)
11
12 model.y2 <- lm(y2 ~ A*B*C*D, data=solar)
13 summary(model.y2)
14 paretoPlot(model.y2)
```

10:21 (Untitled)

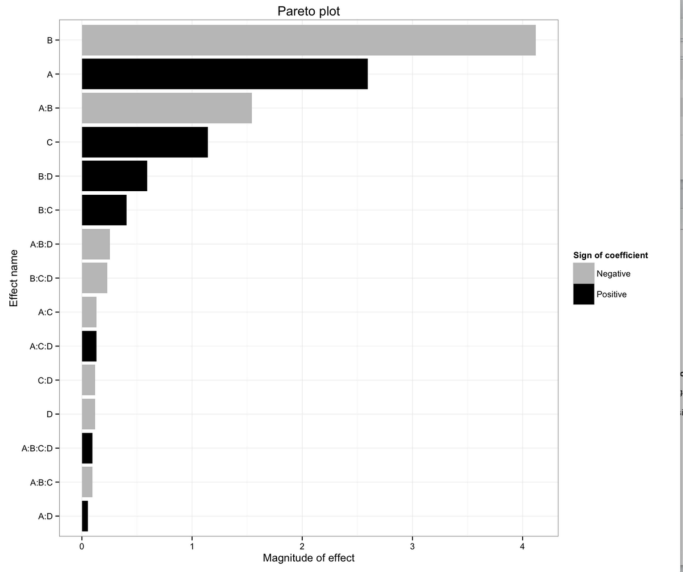
Console ~/

```
B:C         0.40625         NA      NA      NA
A:D         0.05625         NA      NA      NA
B:D         0.59375         NA      NA      NA
C:D        -0.11875         NA      NA      NA
A:B:C      -0.09375         NA      NA      NA
A:B:D      -0.25625         NA      NA      NA
A:C:D       0.13125         NA      NA      NA
B:C:D      -0.23125         NA      NA      NA
A:B:C:D     0.09375         NA      NA      NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:     NaN
F-statistic:   NaN on 15 and 0 DF,  p-value: NA

> paretoPlot(model.y1)
>
```

Plot Zoom



RStudio File Edit Code View Plots Session Build Debug Tools Window Help

RStudio

Go to file/function

solar-panel-example.R | solar-panel.R

Source on Save

```
1  # Solar panel case study, from BHH2, p 230
2  # ----------------------------------------
3
4  # install.packages("pid")  # this is only required onc
5  library(pid)
6  data(solar)
7
8  model.y1 <- lm(y1 ~ A*B*C*D, data=solar)
9  summary(model.y1)
10 paretoPlot(model.y1)
11
12 model.y2 <- lm(y2 ~ A*B*C*D, data=solar)
13 summary(model.y2)
14 paretoPlot(model.y2)
```

10:21 (Untitled)

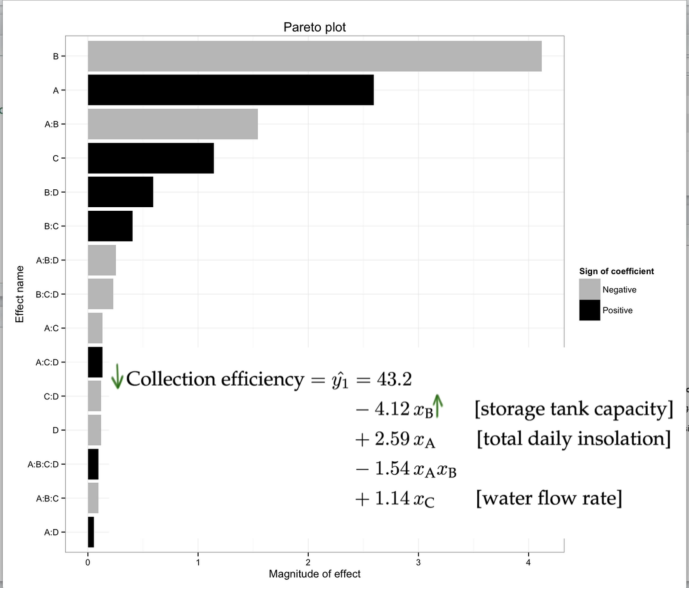
Console ~/

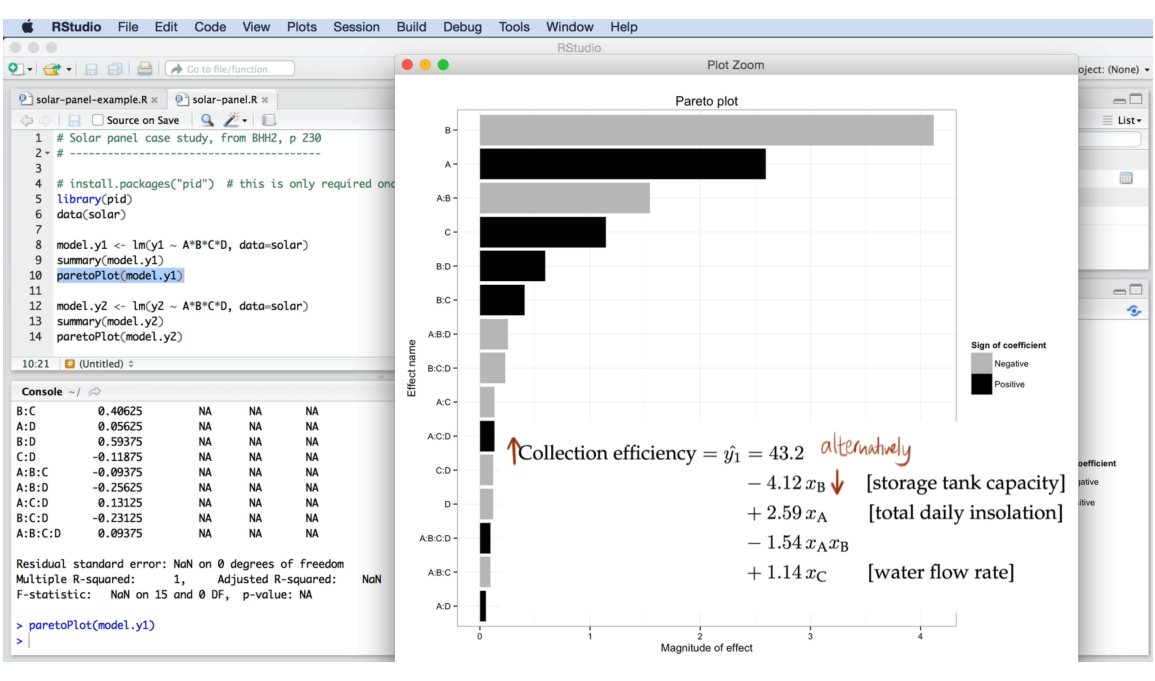
```
B:C         0.40625         NA      NA      NA
A:D         0.05625         NA      NA      NA
B:D         0.59375         NA      NA      NA
C:D        -0.11875         NA      NA      NA
A:B:C      -0.09375         NA      NA      NA
A:B:D      -0.25625         NA      NA      NA
A:C:D       0.13125         NA      NA      NA
B:C:D      -0.23125         NA      NA      NA
A:B:C:D     0.09375         NA      NA      NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:     NaN
F-statistic:   NaN on 15 and 0 DF,  p-value: NA

> paretoPlot(model.y1)
>
```

Plot Zoom




RStudio File Edit Code View Plots Session Build Debug Tools Window Help
RStudio
Go to file/function
solar-panel-example.R
solar-panel.R
Source on Save
1 # Solar panel case study, from BHH2, p 230
2 # ----------
3
4 # install.packages("pid") # this is only required onc
5 library(pid)
6 data(solar)
7
8 model.y1 <- lm(y1 ~ A*B*C*D, data=solar)
9 summary(model.y1)
10 paretoPlot(model.y1)
11
12 model.y2 <- lm(y2 ~ A*B*C*D, data=solar)
13 summary(model.y2)
14 paretoPlot(model.y2)
10:21 (Untitled)
Console ~/
B:C 0.40625 NA NA NA
A:D 0.05625 NA NA NA
B:D 0.59375 NA NA NA
C:D -0.11875 NA NA NA
A:B:C -0.09375 NA NA NA
A:B:D -0.25625 NA NA NA
A:C:D 0.13125 NA NA NA
B:C:D -0.23125 NA NA NA
A:B:C:D 0.09375 NA NA NA
Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: NaN
F-statistic: NaN on 15 and 0 DF, p-value: NA
> paretoPlot(model.y1)
>
Plot Zoom
Pareto plot
B
A
A:B
C
B:D
B:C
A:B:D
B:C:D
A:C
A:C:D
C:D
D
A:B:C:D
A:B:C
A:D
Effect name
0
1
2
3
4
Magnitude of effect
Sign of coefficient
Negative
Positive
↑Collection efficiency = $\hat{y_1} = 43.2$ alternatively
$- 4.12\, x_B$ ↓ [storage tank capacity]
$+ 2.59\, x_A$ [total daily insolation]
$- 1.54\, x_A x_B$
$+ 1.14\, x_C$ [water flow rate]

RStudio File Edit Code View Plots Session Build Debug Tools Window Help

RStudio

Go to file/function

solar-panel-example.R | solar-panel.R

Source on Save

```
1  # Solar panel case study, from BHH2, p 230
2  # ----------------------------------------
3
4  # install.packages("pid")  # this is only required onc
5  library(pid)
6  data(solar)
7
8  model.y1 <- lm(y1 ~ A*B*C*D, data=solar)
9  summary(model.y1)
10 paretoPlot(model.y1)
11
12 model.y2 <- lm(y2 ~ A*B*C*D, data=solar)
13 summary(model.y2)
14 paretoPlot(model.y2)
```

10:21 (Untitled)

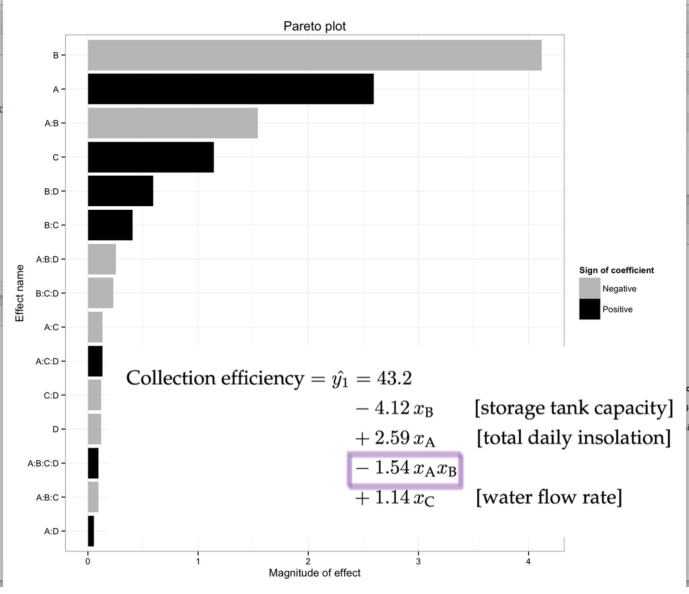
Console ~/

```
B:C          0.40625         NA      NA      NA
A:D          0.05625         NA      NA      NA
B:D          0.59375         NA      NA      NA
C:D         -0.11875         NA      NA      NA
A:B:C       -0.09375         NA      NA      NA
A:B:D       -0.25625         NA      NA      NA
A:C:D        0.13125         NA      NA      NA
B:C:D       -0.23125         NA      NA      NA
A:B:C:D      0.09375         NA      NA      NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:     NaN
F-statistic:   NaN on 15 and 0 DF,  p-value: NA

> paretoPlot(model.y1)
>
```

Plot Zoom



RStudio File Edit Code View Plots Session Build Debug Tools Window Help

RStudio

Go to file/function

solar-panel-example.R | solar-panel.R

Source on Save

```
1  # Solar panel case study, from BHH2, p 230
2  # ----------------------------------------
3
4  # install.packages("pid")  # this is only required onc
5  library(pid)
6  data(solar)
7
8  model.y1 <- lm(y1 ~ A*B*C*D, data=solar)
9  summary(model.y1)
10 paretoPlot(model.y1)
11
12 model.y2 <- lm(y2 ~ A*B*C*D, data=solar)
13 summary(model.y2)
14 paretoPlot(model.y2)
```

14:1 (Untitled)

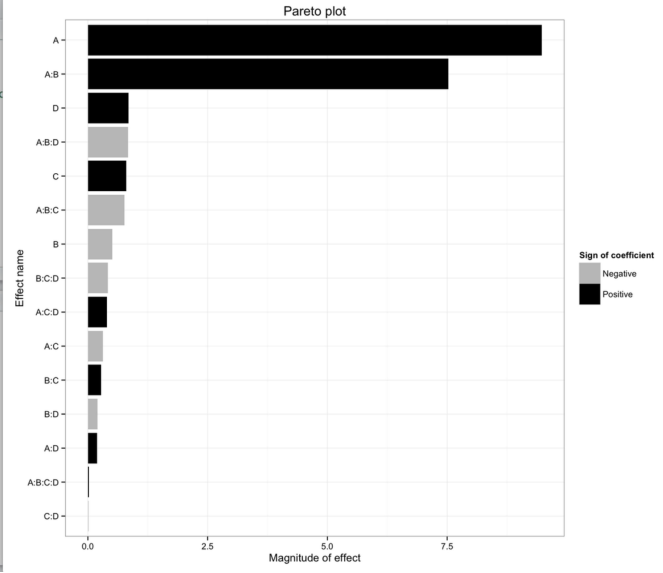
Console ~/

```
B:C          0.2750         NA       NA       NA
A:D          0.1875         NA       NA       NA
B:D         -0.2000         NA       NA       NA
C:D         -0.0125         NA       NA       NA
A:B:C       -0.7625         NA       NA       NA
A:B:D       -0.8375         NA       NA       NA
A:C:D        0.4000         NA       NA       NA
B:C:D       -0.4125         NA       NA       NA
A:B:C:D      0.0250         NA       NA       NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:     NaN
F-statistic:   NaN on 15 and 0 DF,  p-value: NA

> paretoPlot(model.y2)
>
```

Plot Zoom



oject: (None)

List

oefficient
ative
itive

RStudio File Edit Code View Plots Session Build Debug Tools Window Help

RStudio

Go to file/function

solar-panel-example.R | solar-panel.R

Source on Save

```
1  # Solar panel case study, from BHH2, p 230
2  # ----------------------------------------
3
4  # install.packages("pid")  # this is only required onc
5  library(pid)
6  data(solar)
7
8  model.y1 <- lm(y1 ~ A*B*C*D, data=solar)
9  summary(model.y1)
10 paretoPlot(model.y1)
11
12 model.y2 <- lm(y2 ~ A*B*C*D, data=solar)
13 summary(model.y2)
14 paretoPlot(model.y2)
```

14:1 (Untitled)

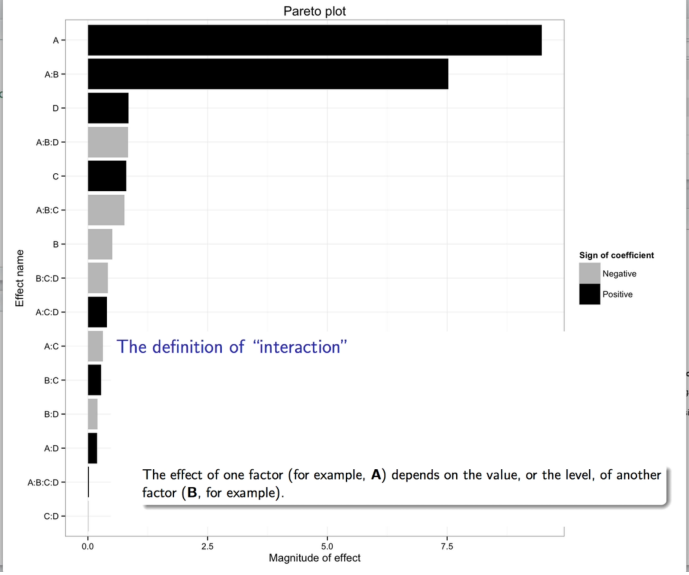
Console ~/

```
B:C          0.2750         NA      NA      NA
A:D          0.1875         NA      NA      NA
B:D         -0.2000         NA      NA      NA
C:D         -0.0125         NA      NA      NA
A:B:C       -0.7625         NA      NA      NA
A:B:D       -0.8375         NA      NA      NA
A:C:D        0.4000         NA      NA      NA
B:C:D       -0.4125         NA      NA      NA
A:B:C:D      0.0250         NA      NA      NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:     NaN
F-statistic:   NaN on 15 and 0 DF,  p-value: NA

> paretoPlot(model.y2)
>
```

Plot Zoom



Object: (None)

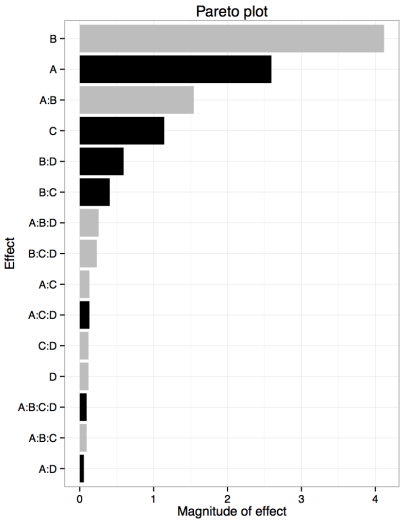
List

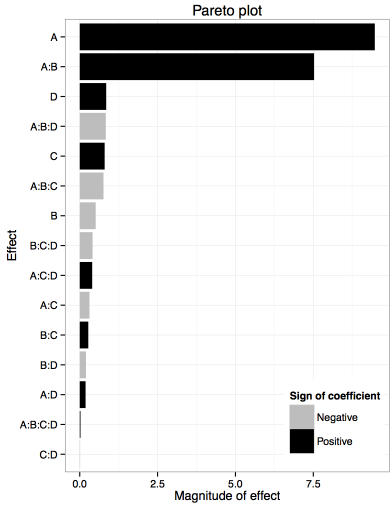
oefficient
ative
itive

Advanced thinking: optimizing multiple objectives

Consider the case where the aim is to maximize **both** y_1 and y_2.

y_1 = "collection efficiency" y_2 = "energy delivery efficiency"





Specify at what levels (high or low) should we set the factors **A**, **B**, **C**, and **D** to achieve a maximum in both y_1 *and* in y_2.