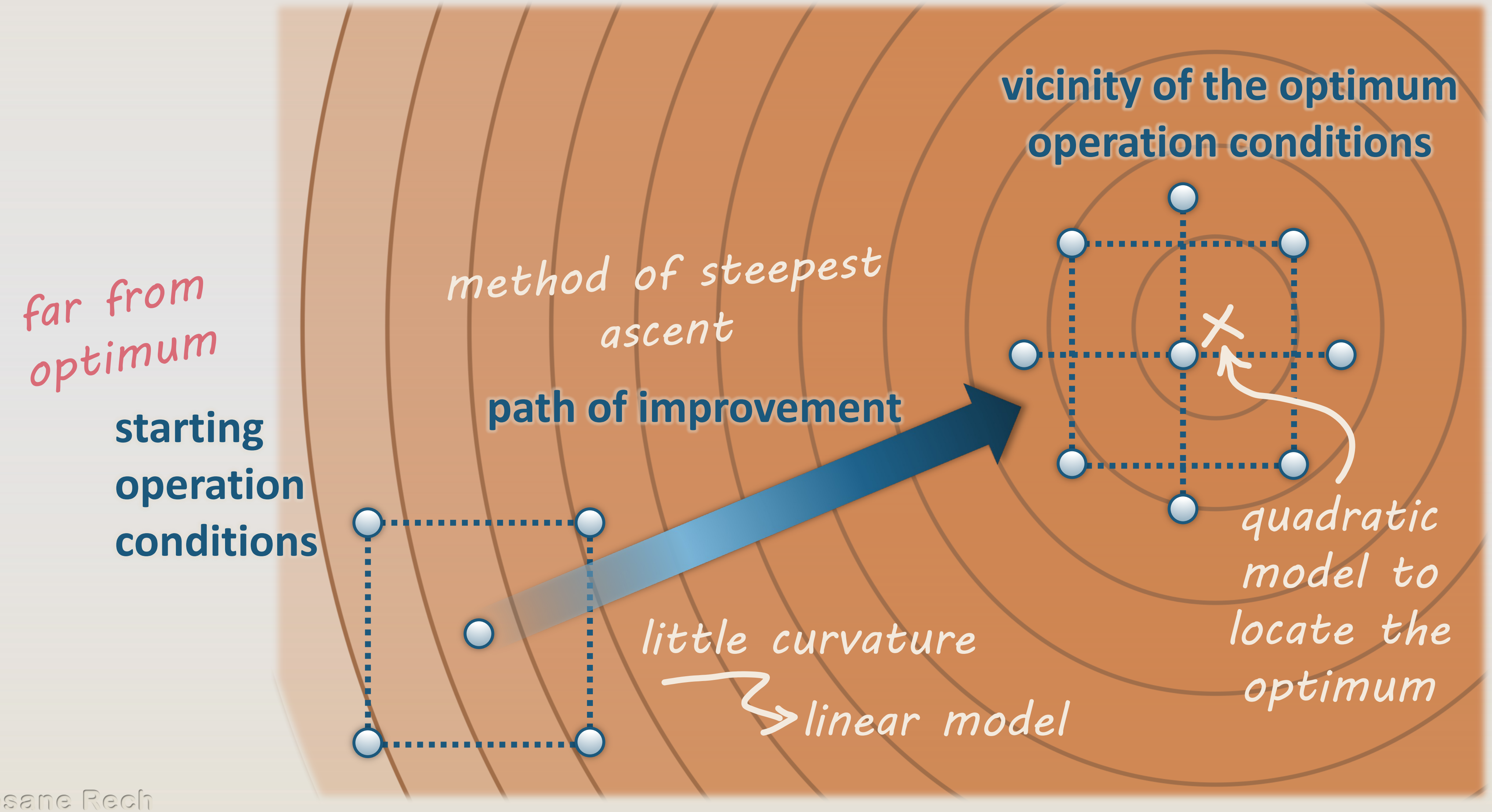
Response Surface Methodology

Introduction to Response Surface Methodology

Response Surface Methodology, or RSM, is a collection of mathematical and statistical techniques useful for the modelling and analysis of problems in which a response of interest is influenced by several variables and the objective is to optimise this response.

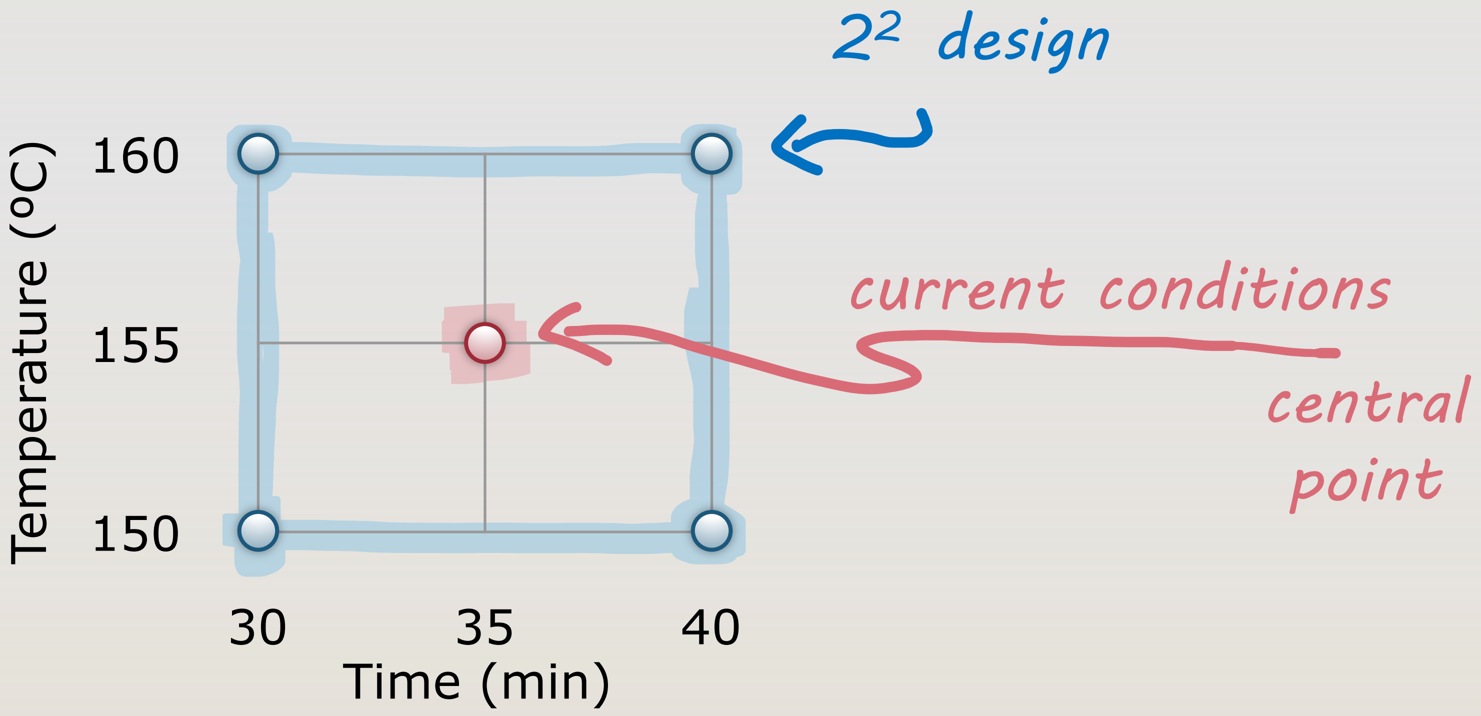
Response Surface Methodology is a Sequential Procedure



The Starting Design

2^2 Design with Central Points

A team of researchers needs to perform experiments to maximise the yield of a chemical reaction. The current conditions are 35 min and 155 ˚C, which results in yield around 40 %. The chemist decides to build a 2^2 design using the current conditions as central point in the design:

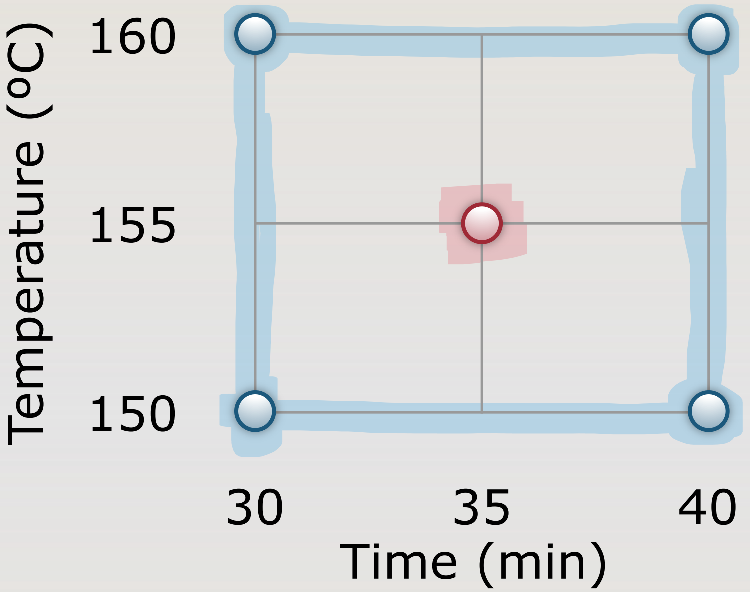


2^2 Design with Central Points

x_1	x_2	Time (min)	Temp (˚C)	Y (%)
−1	−1	30	150	39.3
−1	1	30	160	40.0
1	−1	40	150	40.9
1	1	40	160	41.5
0	0	35	155	40.3
0	0	35	155	40.5
0	0	35	155	40.7
0	0	35	155	40.2
0	0	35	155	40.6

2^k design

central points



$$x_1 = \frac{Time - 35}{5}$$

$$x_2 = \frac{Temp - 155}{5}$$

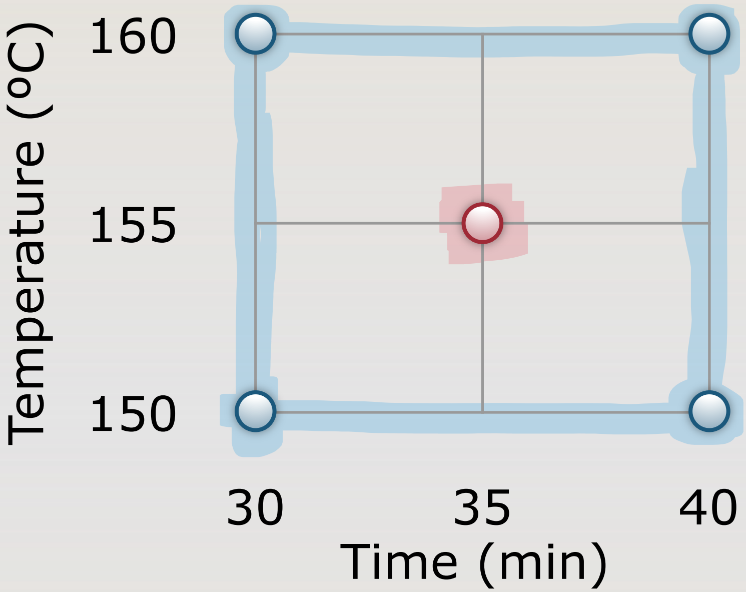
2^2 Design with Central Points

x_1	x_2	Time (min)	Temp (˚C)	Y (%)
-1	-1	30	150	39.3
-1	1	30	160	40.0
1	-1	40	150	40.9
1	1	40	16[illegible]	[illegible]
0	0	35	15[illegible]	[illegible]
0	0	35	15[illegible]	[illegible]
0	0	35	15[illegible]	[illegible]
0	0	35	15[illegible]	[illegible]
0	0	35	155	40.6

2^k design

Using R-Studio:

- Data file: DoEOpt04.csv
- R code file: DoEOpt04.R



$$= \frac{Time - 35}{5}$$

$$x_2 = \frac{Temp - 155}{5}$$

Analysing the Starting Design

```r
DoEOpt04 <- as.coded.data(DoEOpt04,
                x1 ~ (Time-35)/5,
                x2 ~ (Temp-155)/5)
```

relationship between coded and natural variables

```r
model <- rsm(Y ~ FO(x1,x2) + TWI(x1,x2), data = DoEOpt04)
```

response

first order model

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2$$

two-way interaction

$$\beta_{12} x_1 x_2$$

data file

2^2 Design with Central Points

```
Call:
rsm(formula = Y ~ FO(x1, x2) + TWI(x1, x2), data = DoEOpt04)

             Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 40.444444   0.062311 649.0693 1.648e-13 ***
x1           0.775000   0.093467   8.2917 0.0004166 ***
x2           0.325000   0.093467   3.4772 0.0177127 *
x1:x2       -0.025000   0.093467  -0.2675 0.7997870
```

significant

not significant

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Multiple R-squared:  0.9418,    Adjusted R-squared:  0.9069
F-statistic: 26.97 on 3 and 5 DF,  p-value: 0.00163
```

high R^2 and low p-value

```
Analysis of Variance Table

Response: Y
            Df  Sum Sq Mean Sq F value    Pr(>F)
FO(x1, x2)   2 2.82500 1.41250 40.4213 0.0008188
TWI(x1, x2)  1 0.00250 0.00250  0.0715 0.7997870
Residuals    5 0.17472 0.03494
Lack of fit  1 0.00272 0.00272  0.0633 0.8137408
Pure error   4 0.17200 0.04300
```

significant

not significant

not significant

2^2 Design with Central Points

```
Call:
rsm(formula = Y ~ FO(x1, x2), data = DoEOpt04)

              Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 40.444444   0.057288 705.9869 5.451e-16 ***
x1           0.775000   0.085932   9.0188  0.000104 ***
x2           0.325000   0.085932   3.7821  0.009158 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Multiple R-squared:  0.941,     Adjusted R-squared:  0.9213
F-statistic: 47.82 on 2 and 6 DF,  p-value: 0.0002057

Analysis of Variance Table

Response: Y
            Df  Sum Sq Mean Sq F value    Pr(>F)
FO(x1, x2)   2 2.82500 1.41250 47.8213 0.0002057
Residuals    6 0.17722 0.02954
Lack of fit  2 0.00522 0.00261  0.0607 0.9419341
Pure error   4 0.17200 0.04300
```

significant

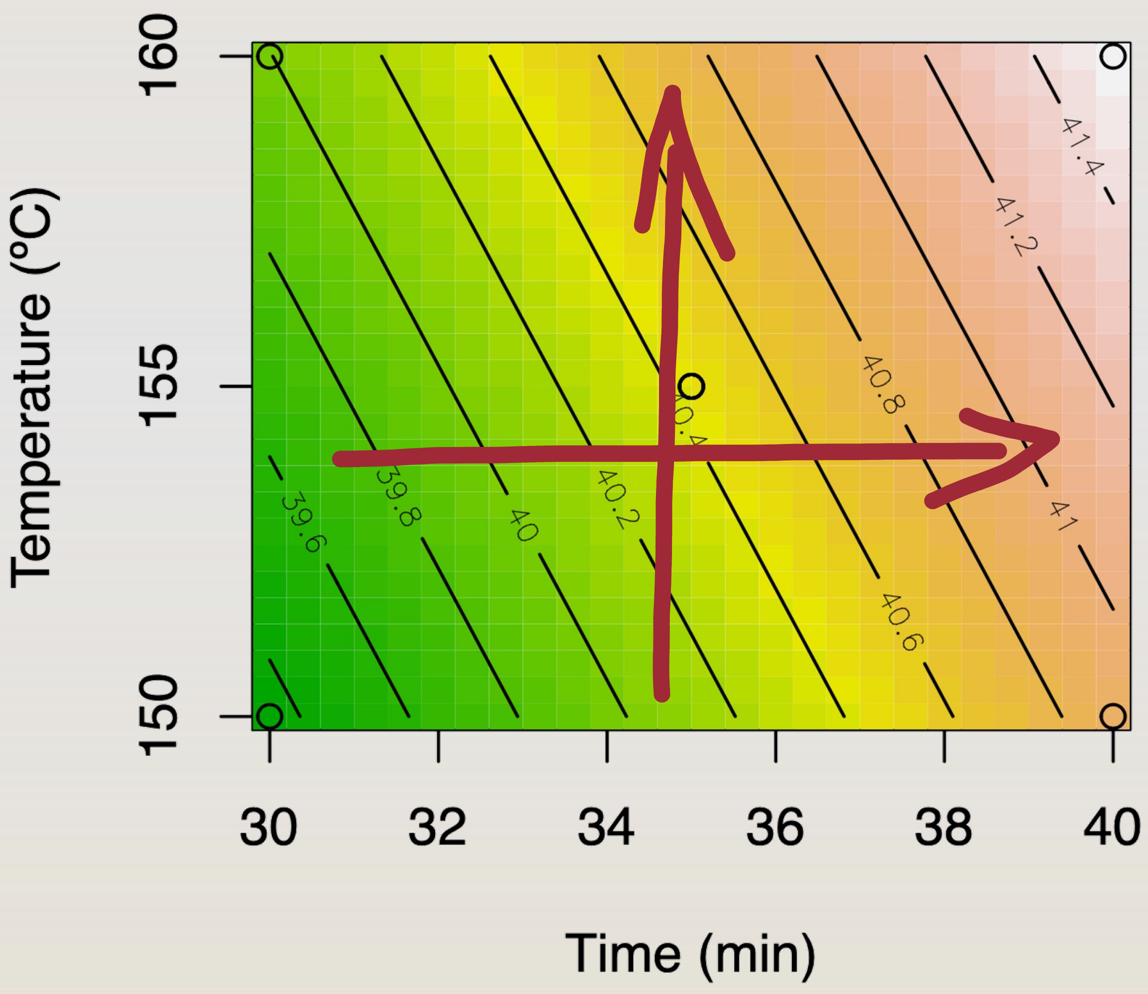
high R^2

higher adj-R^2 and lower p-value

significant

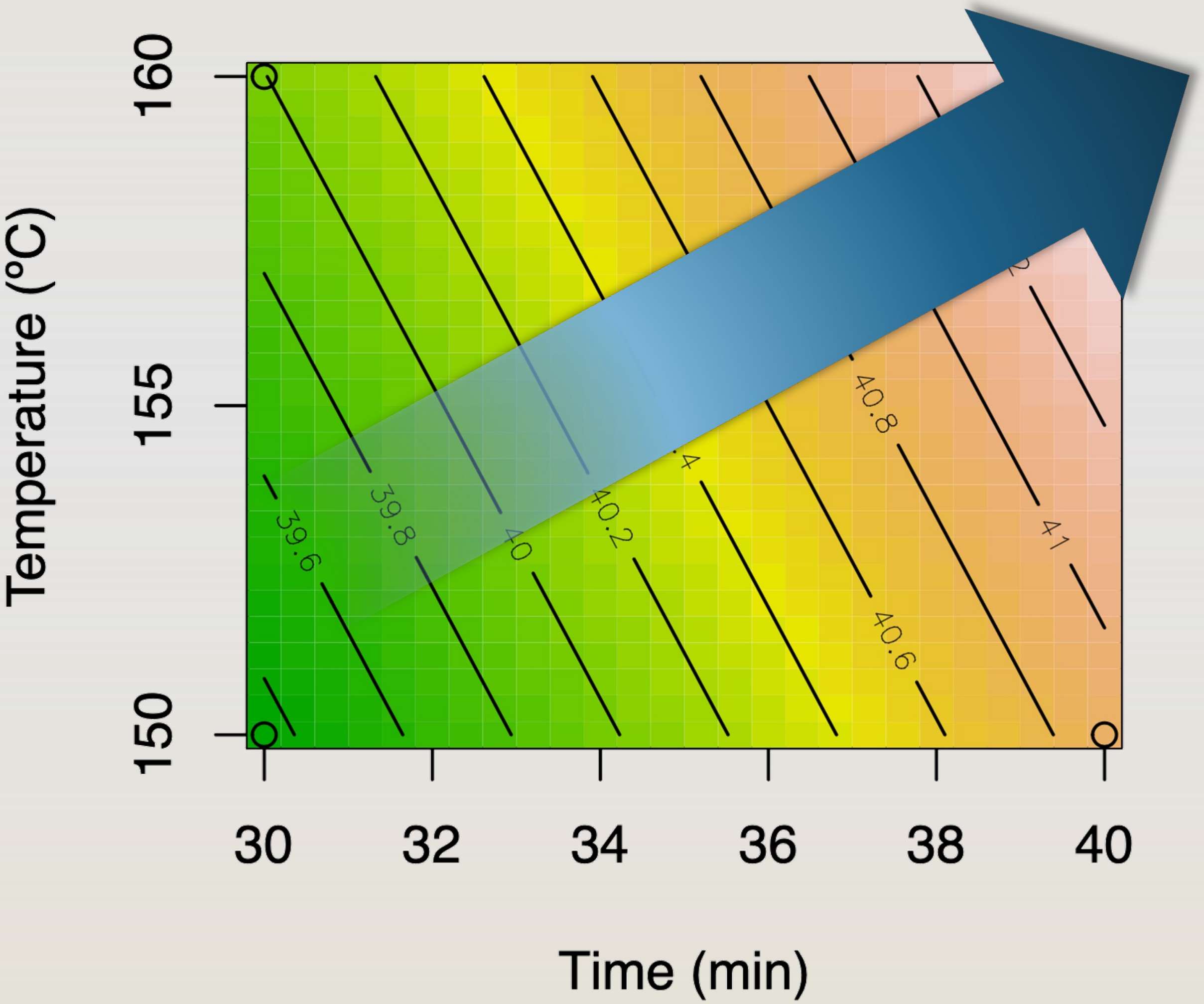
not significant

2^2 Design with Central Points



the yield seems to be far from the optimum!

2^2 Design with Central Points



maximum increase in the yield

direction of the steepest ascent

Method of the Steepest Ascent

Method of Steepest Ascent

```
Call:
rsm(formula = Y ~ FO(x1, x2), data = DoEOpt04)

            Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 40.444444   0.057288 705.9869 5.451e-16
x1           0.775000   0.085932   9.0188  0.000104
x2           0.325000   0.085932   3.7821  0.009158
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’

Multiple R-squared:  0.941,     Adjusted R-squared:
F-statistic: 47.82 on 2 and 6 DF,  p-value: 0.000205

Analysis of Variance Table

Response: Y
            Df  Sum Sq Mean Sq F value    Pr(>F)
FO(x1, x2)   2 2.82500 1.41250 47.8213 0.0002057
Residuals    6 0.17722 0.02954
Lack of fit  2 0.00522 0.00261  0.0607 0.9419341
Pure error   4 0.17200 0.04300

Direction of steepest ascent (at radius 1):
       x1        x2
0.9221944 0.3867267

Corresponding increment in original units:
    Time      Temp
4.610972 1.933633
```


```
            Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 40.444444   0.057288 705.9869 5.451e-16 ***
x1           0.775000   0.085932   9.0188  0.000104 ***
x2           0.325000   0.085932   3.7821  0.009158 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Multiple R-squared:  0.941,     Adjusted R-squared:  0.9213
F-statistic: 47.82 on 2 and 6 DF,  p-value: 0.0002057
```

Method of Steepest Ascent

```
Call:
rsm(formula = Y ~ FO(x1, x2), data = DoEOpt04)

             Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 40.444444   0.057288 705.9869 5.451e-16
x1           0.775000   0.085932   9.0188  0.000104
x2           0.325000   0.085932   3.7821  0.009158
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’

Multiple R-squared:  0.941,     Adjusted R-squared:
F-statistic: 47.82 on 2 and 6 DF,  p-value: 0.000205

Analysis of Variance Table

Response: Y
            Df  Sum Sq Mean Sq F value    Pr(>F)
FO(x1, x2)   2 2.82500 1.41250 47.8213 0.0002057
Residuals    6 0.17722 0.02954
Lack of fit  2 0.00522 0.00261  0.0607 0.9419341
Pure error   4 0.17200 0.04300

Direction of steepest ascent (at radius 1):
       x1        x2
0.9221944 0.3867267

Corresponding increment in original units:
    Time     Temp
4.610972 1.933633
```


```
Analysis of Variance Table

Response: Y
            Df  Sum Sq Mean Sq F value    Pr(>F)
FO(x1, x2)   2 2.82500 1.41250 47.8213 0.0002057
Residuals    6 0.17722 0.02954
Lack of fit  2 0.00522 0.00261  0.0607 0.9419341
Pure error   4 0.17200 0.04300
```

Method of Steepest Ascent

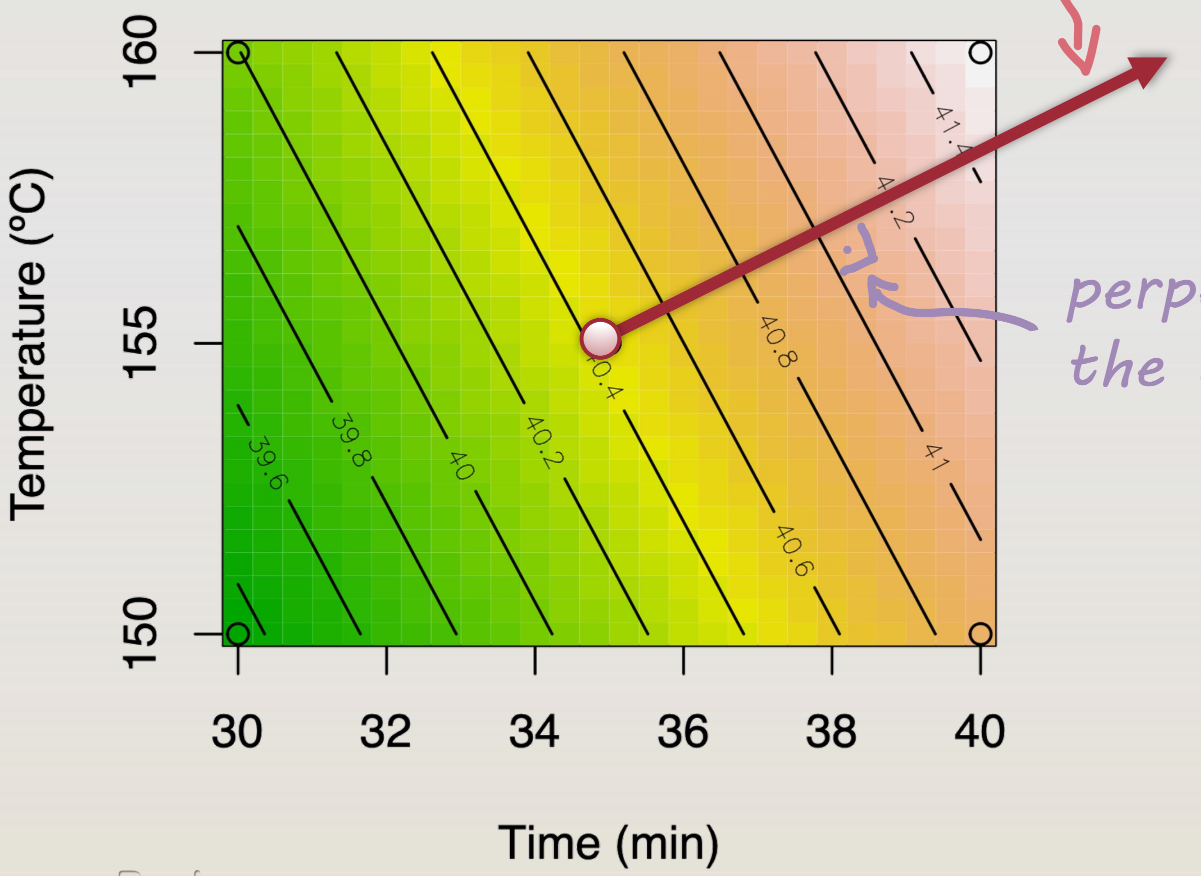
```
Call:
rsm(formula = Y ~ FO(x1, x2), data = DoEOpt04)

             Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 40.444444   0.057288 705.9869 5.451e-16
x1           0.775000   0.085932   9.0188  0.000104
x2           0.325000   0.085932   3.7821  0.009158
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’

Multiple R-squared:  0.941,     Adjusted R-squared:
F-statistic: 47.82 on 2 and 6 DF,  p-value: 0.000205

Analysis of Variance Table

Response: Y
            Df  Sum Sq Mean Sq F value    Pr(>F)
FO(x1, x2)   2 2.82500 1.41250 47.8213 0.0002057
Residuals    6 0.17722 0.02954
Lack of fit  2 0.00522 0.00261  0.0607 0.9419341
Pure error   4 0.17200 0.04300

Direction of steepest ascent (at radius 1):
       x1        x2
0.9221944 0.3867267

Corresponding increment in original units:
    Time      Temp
4.610972 1.933633
```

```
Direction of steepest ascent (at radius 1):
       x1        x2
0.9221944 0.3867267

Corresponding increment in original units:
       Time Temperature
   4.610972    1.933633
```

Method of Steepest Ascent

maximum increase in the response

direction of the steepest accent



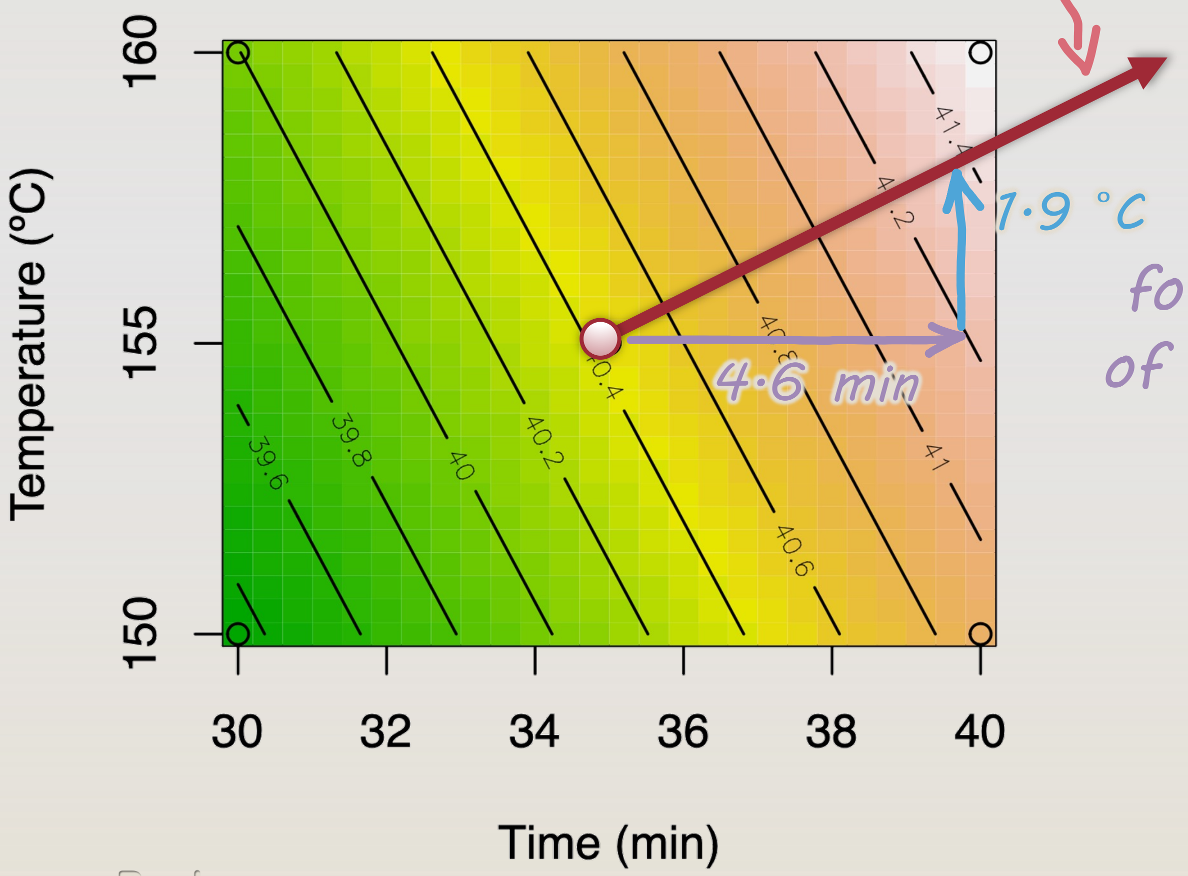
perpendicular to the contour lines

```
Direction of steepest ascent (at radius 1):
       x1        x2
0.9221944 0.3867267

Corresponding increment in original units:
       Time Temperature
   4.610972     1.933633
```

Method of Steepest Ascent

maximum increase
in the response

direction of the
steepest accent




```
Direction of steepest ascent (at radius 1):
       x1        x2
0.9221944 0.3867267

Corresponding increment in original units:
     Time Temperature
 4.610972    1.933633
```

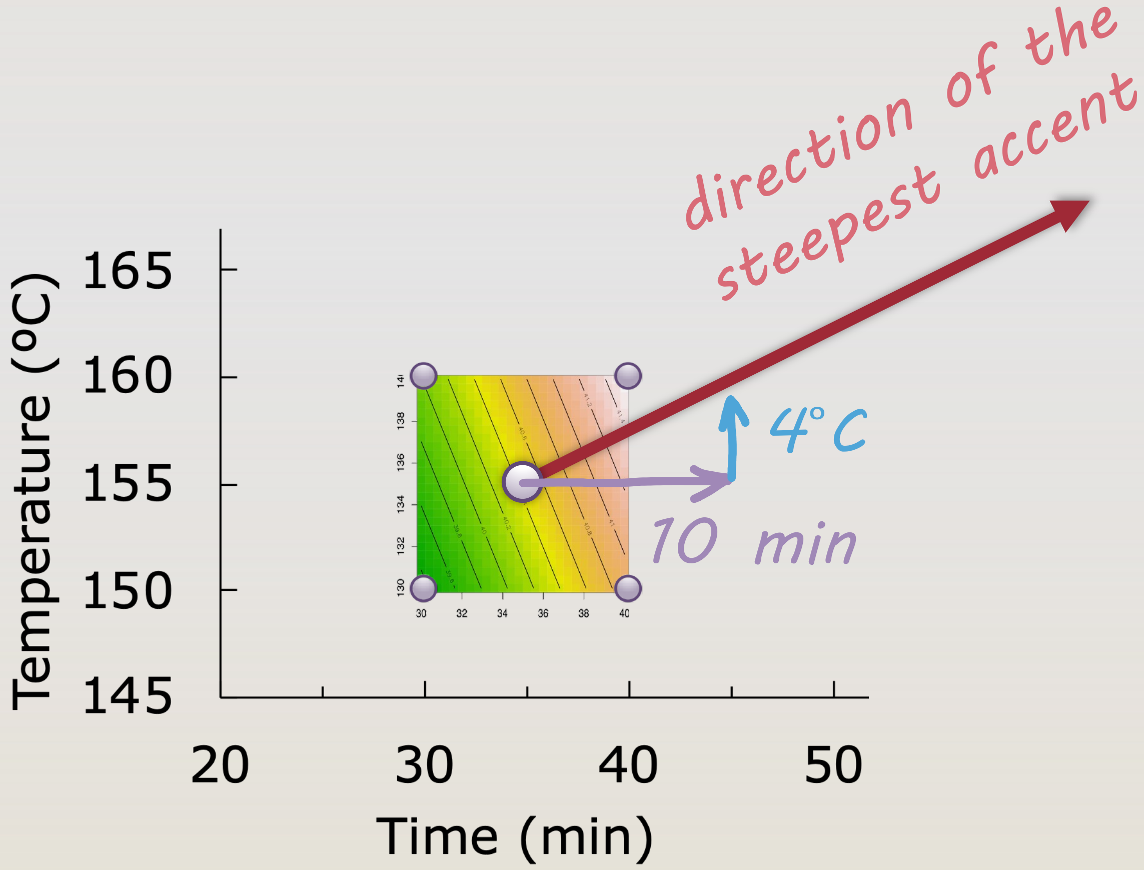
for each 4.6 min
of increase in time

we should increase the
temperature by 1.9°C

Steepest ascent:

$$\Delta T = \frac{1.9}{4.6} \times \Delta t$$

Method of Steepest Ascent

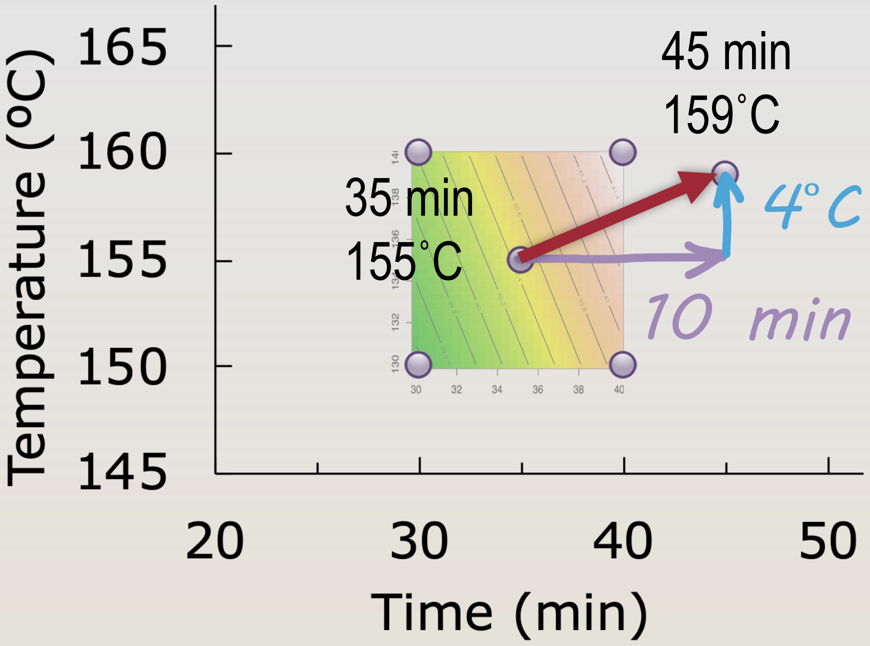



```
Direction of steepest ascent (at radius 1):
        x1        x2
0.9221944 0.3867267

Corresponding increment in original units:
        Time Temperature
    4.610972    1.933633
```

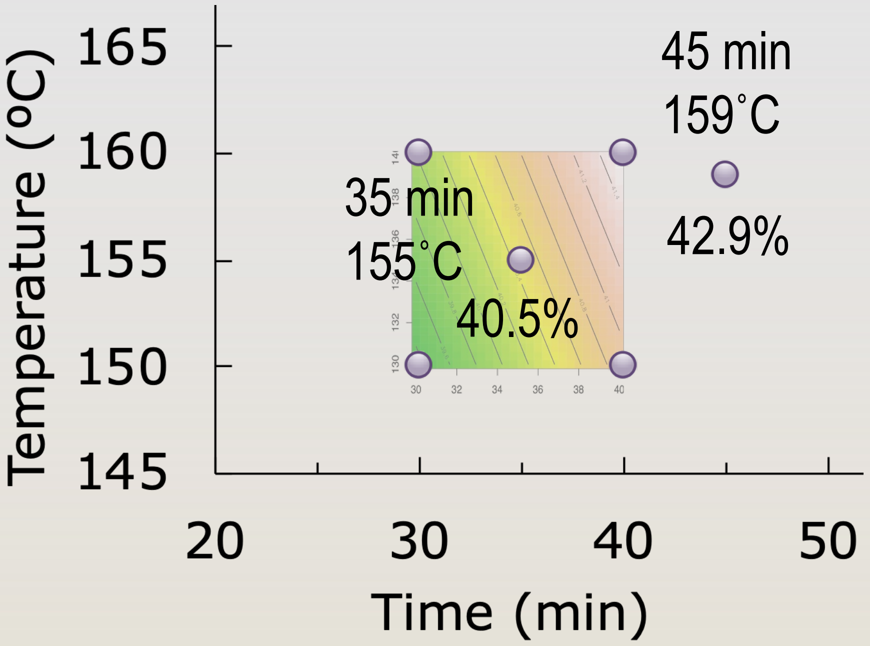
Steepest ascent:

$$\Delta T = \frac{1.9}{4.6} \times \underbrace{\Delta t}_{10\ min} = \frac{1.9}{4.6} \times 10 = \underbrace{4.1}_{4\,^{\circ}C}\,^{\circ}\mathrm{C}$$

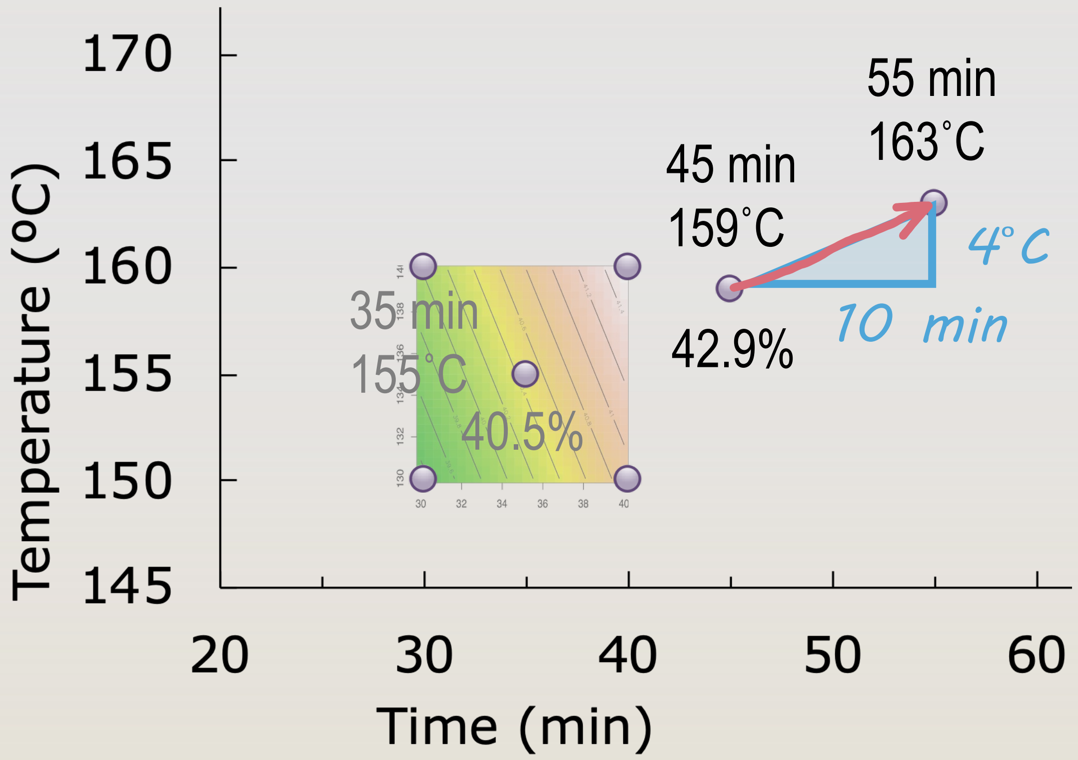
Method of Steepest Ascent



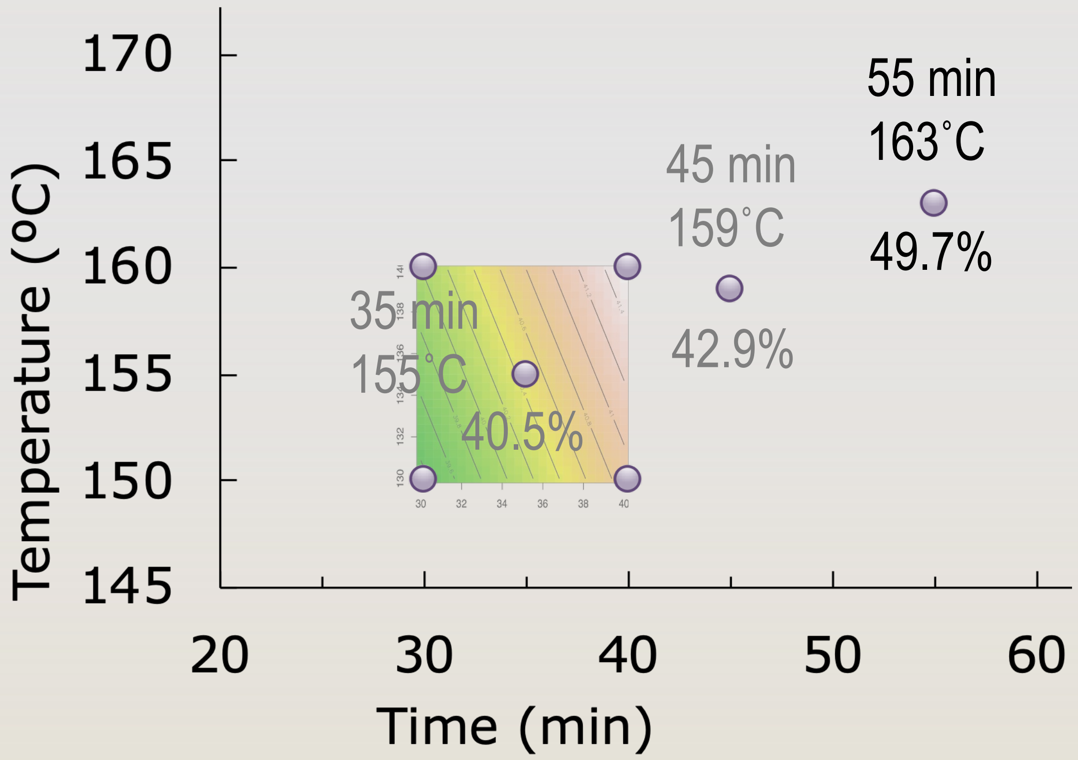
Method of Steepest Ascent



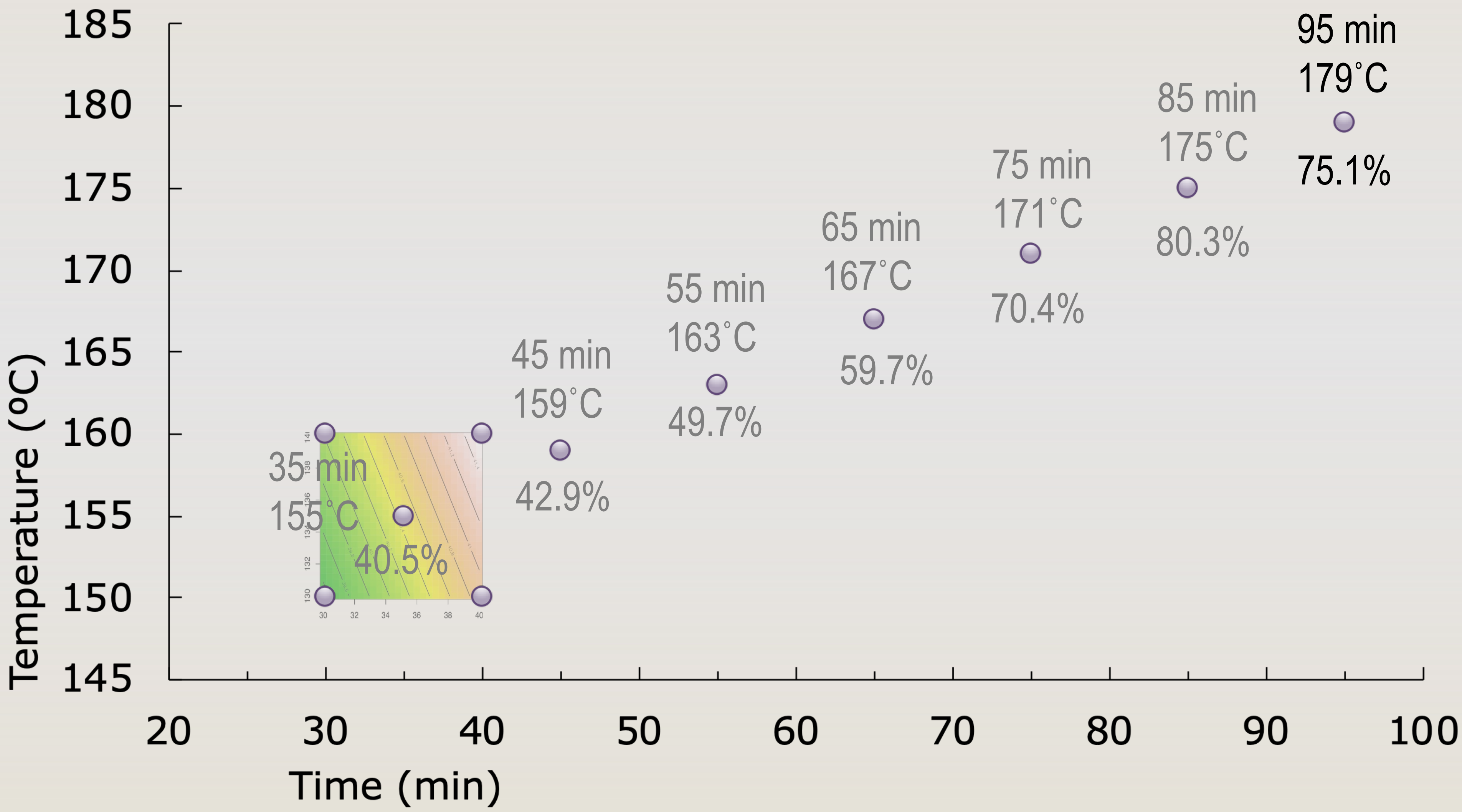
Method of Steepest Ascent



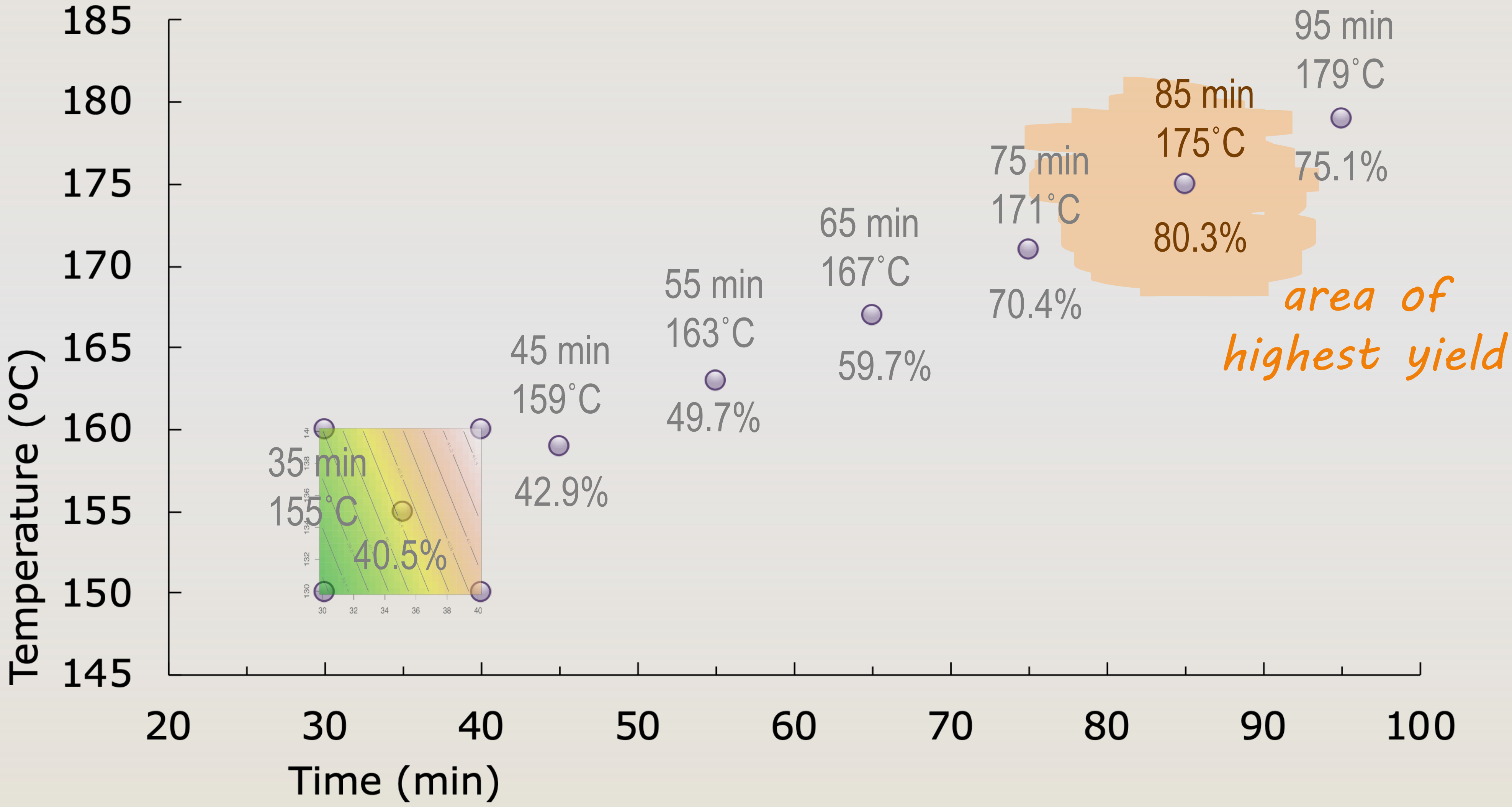
Method of Steepest Ascent



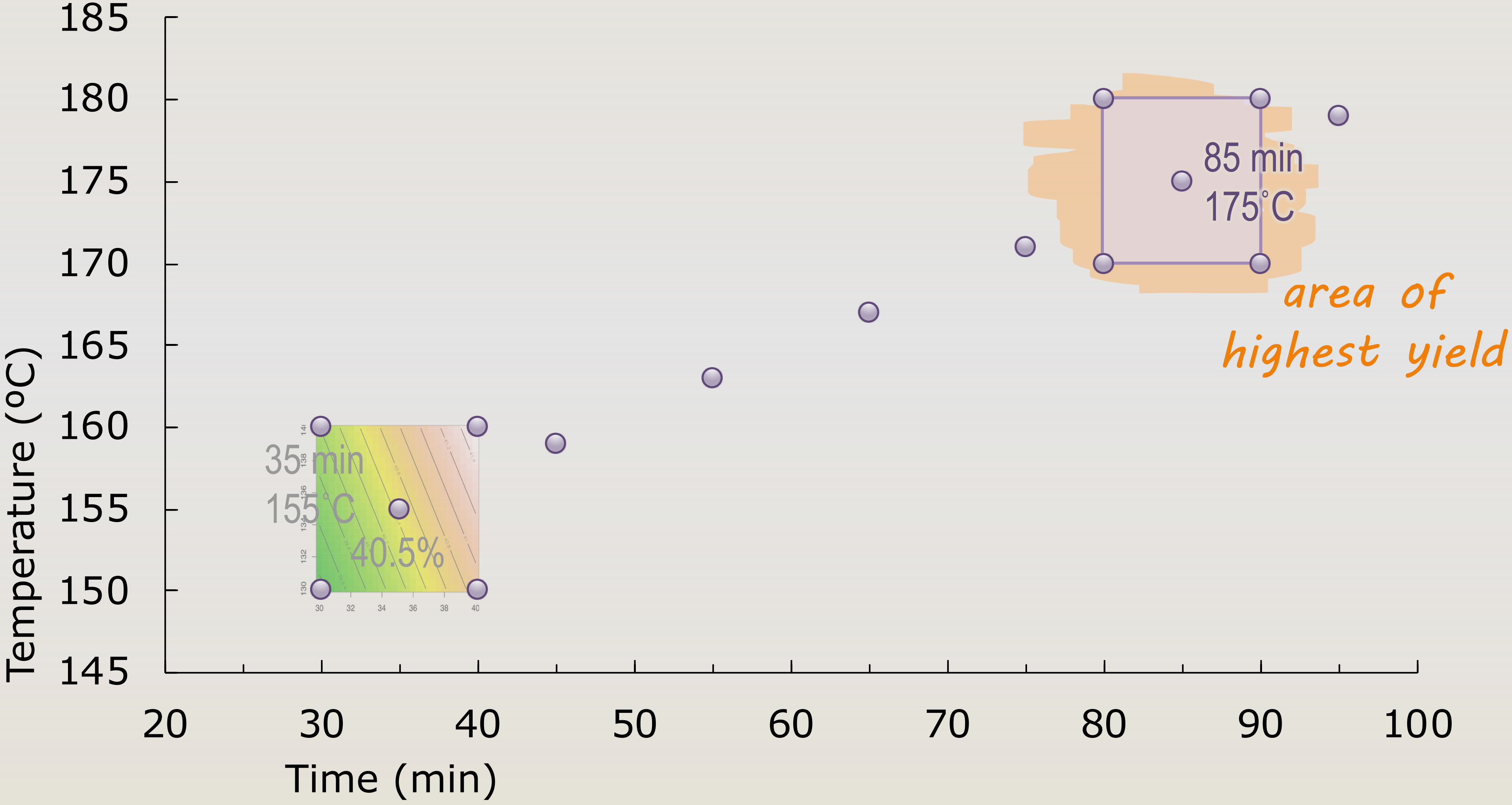
Method of Steepest Ascent



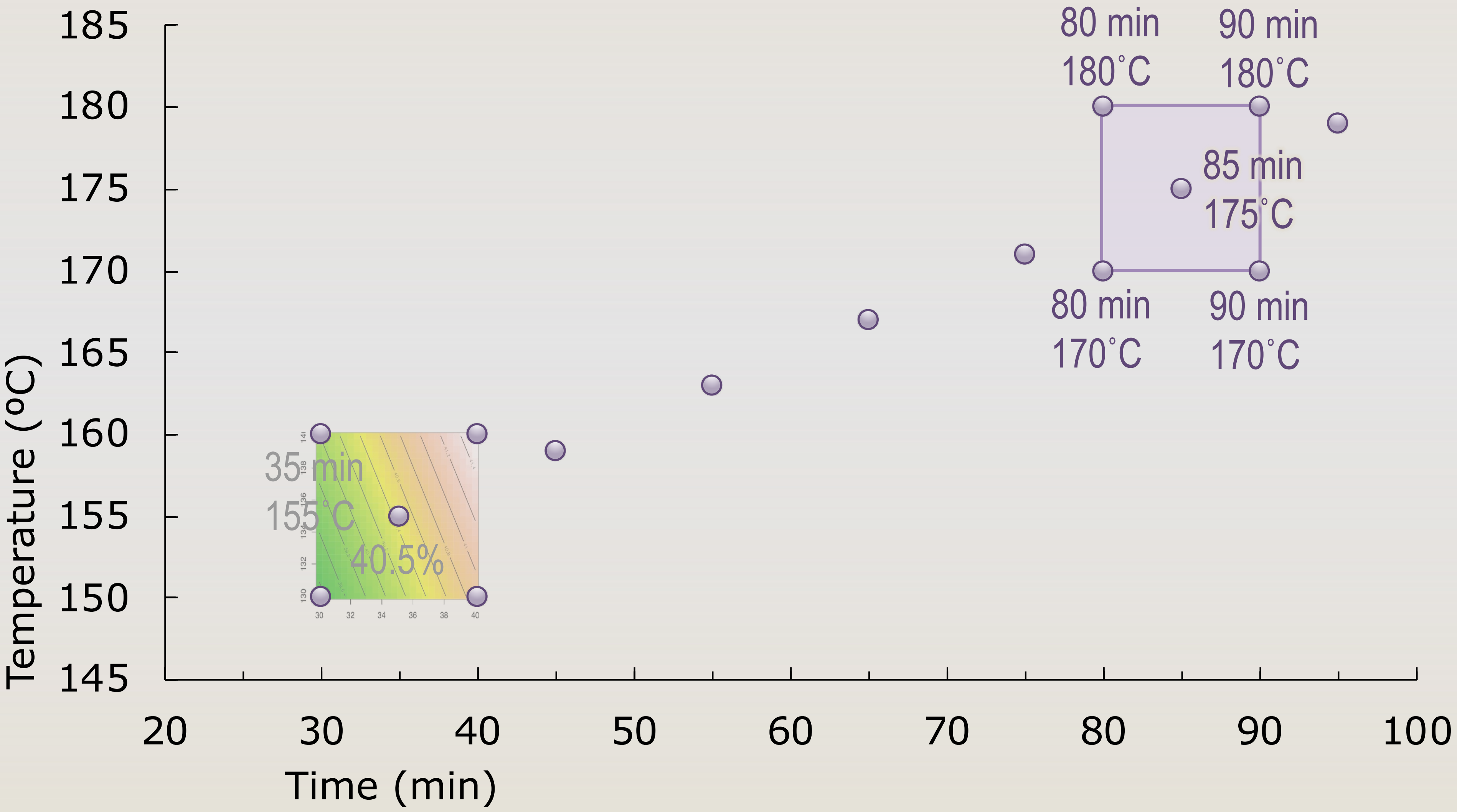
Method of Steepest Ascent



Method of Steepest Ascent



Method of Steepest Ascent

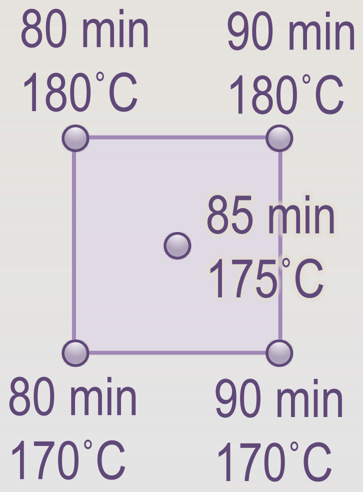


Method of Steepest Ascent

x_1	x_2	Time (min)	Temp (˚C)	Y (%)
-1	-1	80	170	76.5
-1	1	80	180	77.0
1	-1	90	170	78.0
1	1	90	180	79.5
0	0	85	175	79.9
0	0	85	175	80.3
0	0	85	175	80.0
0	0	85	175	79.7
0	0	85	175	79.8

2^k design

central points



$$x_1 = \frac{Time - 85}{5}$$

$$x_2 = \frac{Temp - 175}{5}$$

Method of Steepest Ascent

x_1	x_2	Time (min)	Temp (˚C)	Y (%)
-1	-1	80	170	76.5
-1	1	80	180	[illegible]
1	-1	90	170	[illegible]
1	1	90	180	[illegible]
0	0	85	175	[illegible]
0	0	85	175	[illegible]
0	0	85	175	80.0
0	0	85	175	79.7
0	0	85	175	79.8

Using R-Studio:

- Data file: DoEOpt05.csv
- R code file: DoEOpt05.R

points

$$x_1 = \frac{Time - 85}{5}$$

$$x_2 = \frac{Temp - 175}{5}$$

Interpreting a Lousy Fitting

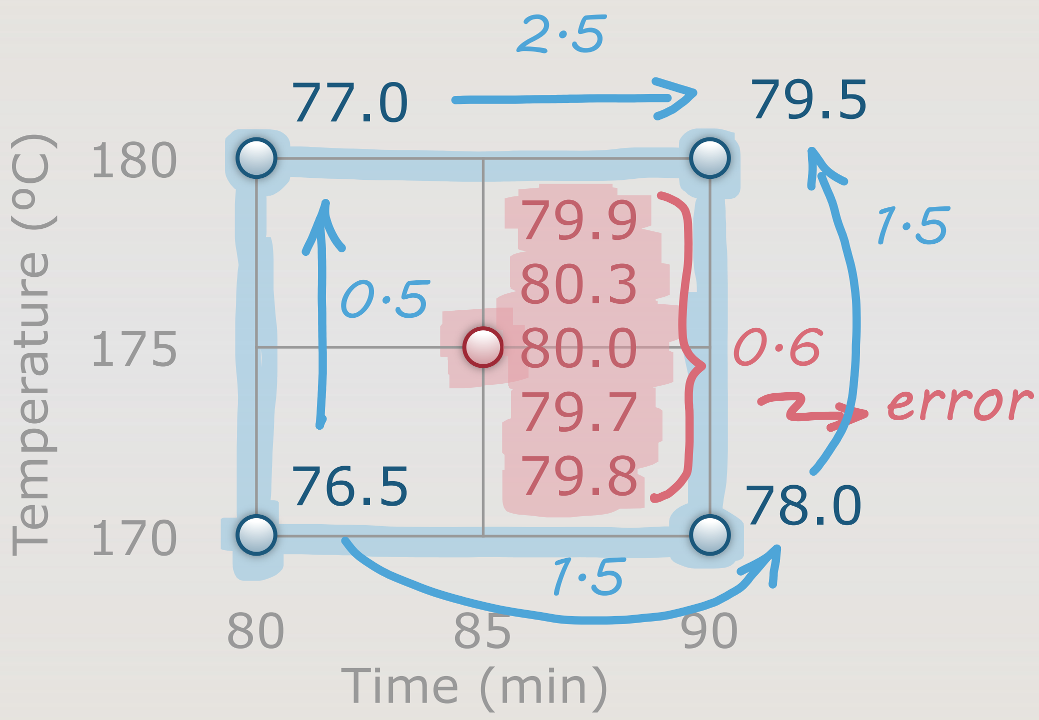
Interpreting a Lousy Fitting

```
Call:
rsm(formula = Y ~ FO(x1, x2) + TWI(x1, x2), data = DoEOpt05)

              Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 78.96667    0.49148 160.6702 1.772e-10 ***
x1           1.00000    0.73722   1.3564    0.2330
x2           0.50000    0.73722   0.6782    0.5277
x1:x2        0.25000    0.73722   0.3391    0.7483
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Multiple R-squared:  0.3257,    Adjusted R-squared:  -0.07891
F-statistic: 0.805 on 3 and 5 DF,  p-value: 0.5426

Analysis of Variance Table

Response: Y
            Df Sum Sq Mean Sq F value    Pr(>F)
FO(x1, x2)   2  5.000   2.500   1.150 0.3882680
TWI(x1, x2)  1  0.250   0.250   0.115 0.7483056
Residuals    5 10.870   2.174
Lack of fit  1 10.658  10.658 201.094 0.0001436
Pure error   4  0.212   0.053
```

Interpreting a Lousy Fitting

```
Call:
rsm(formula = Y ~ FO(x1, x2) + TWI(x1, x2), data = DoEOpt05)

            Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 78.96667    0.49148 160.6702 1.772e-10 ***
x1           1.00000    0.73722   1.3564    0.2330
x2           0.50000    0.73722   0.6782    0.5277
x1:x2        0.25000    0.73722   0.3391    0.7483
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Multiple R-squared:  0.3257,    Adjusted R-squared:  -0.07891
F-statistic: 0.805 on 3 and 5 DF

Analysi

Response

FO(x1, x
TWI(x1,                     0.115 0.7483056
Residuals    5 10.870    2.174
Lack of fit  1 10.658   10.658 201.094 0.0001436
Pure error   4  0.212    0.053
```

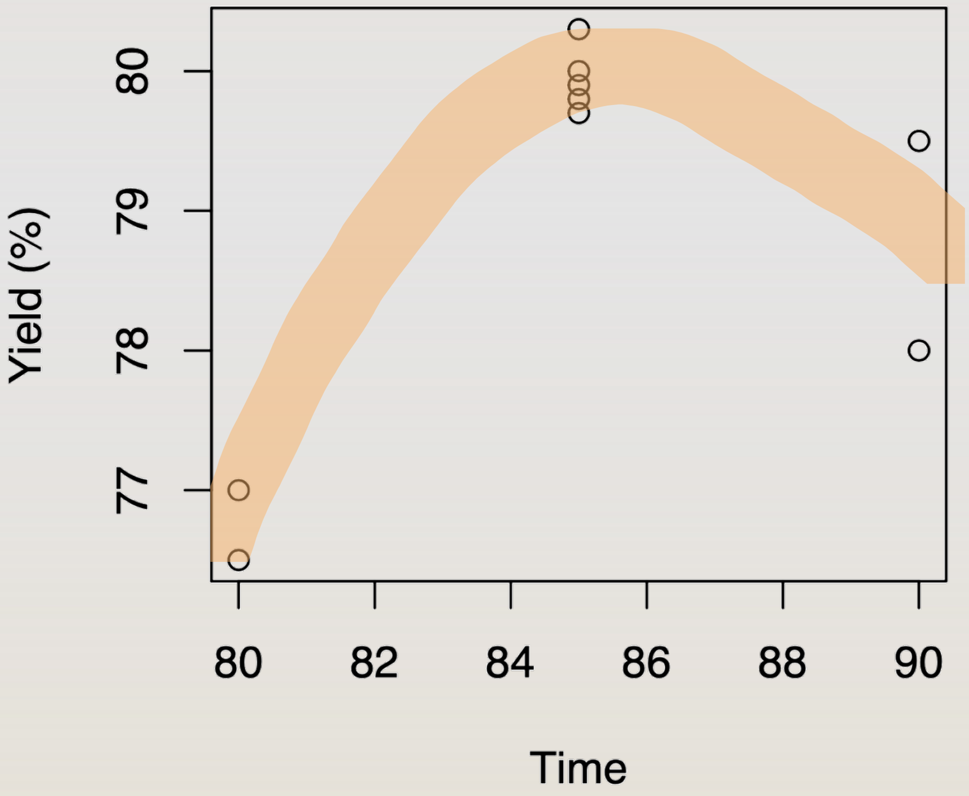
The linear model with interaction is not adequate to describe the behaviour of the data.

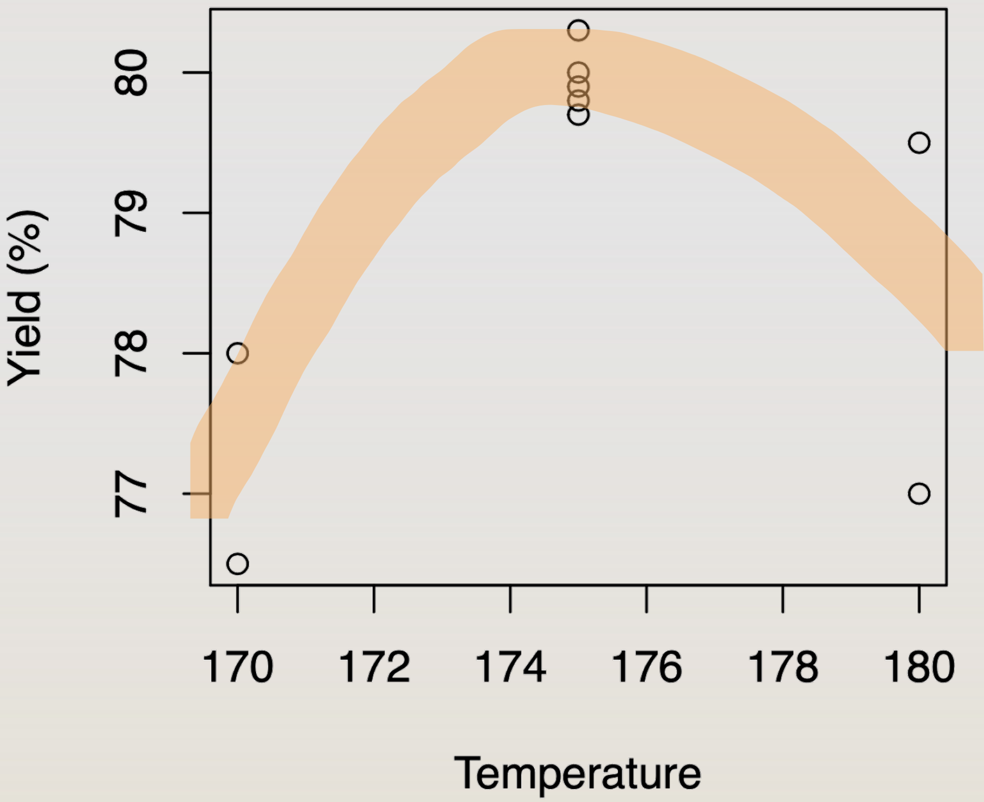


the effects of time and temperature seem to be higher than the intrinsic error

Interpreting a Lousy Fitting

non-linear behaviour between the regression variables and the response



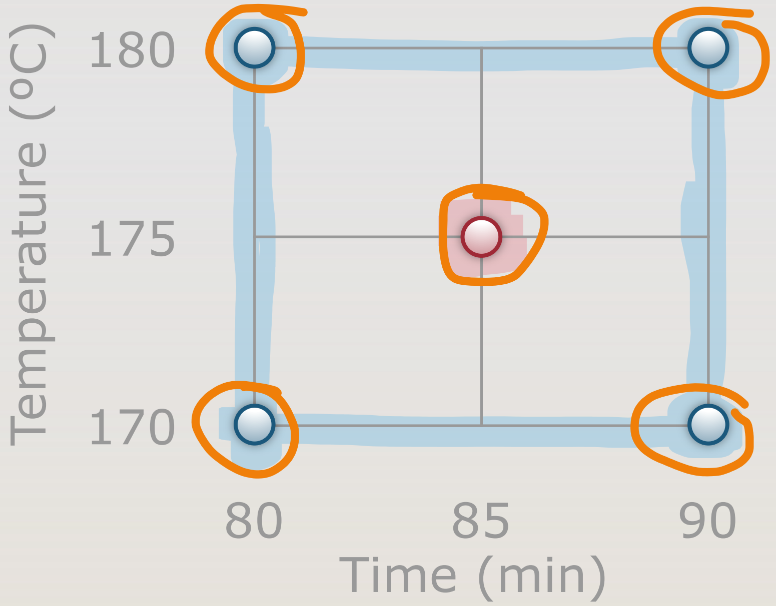


Second-Order Regression Models

Quadradic model with interaction:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2$$

6 regression parameters



5 independent runs

replicates are not independent runs

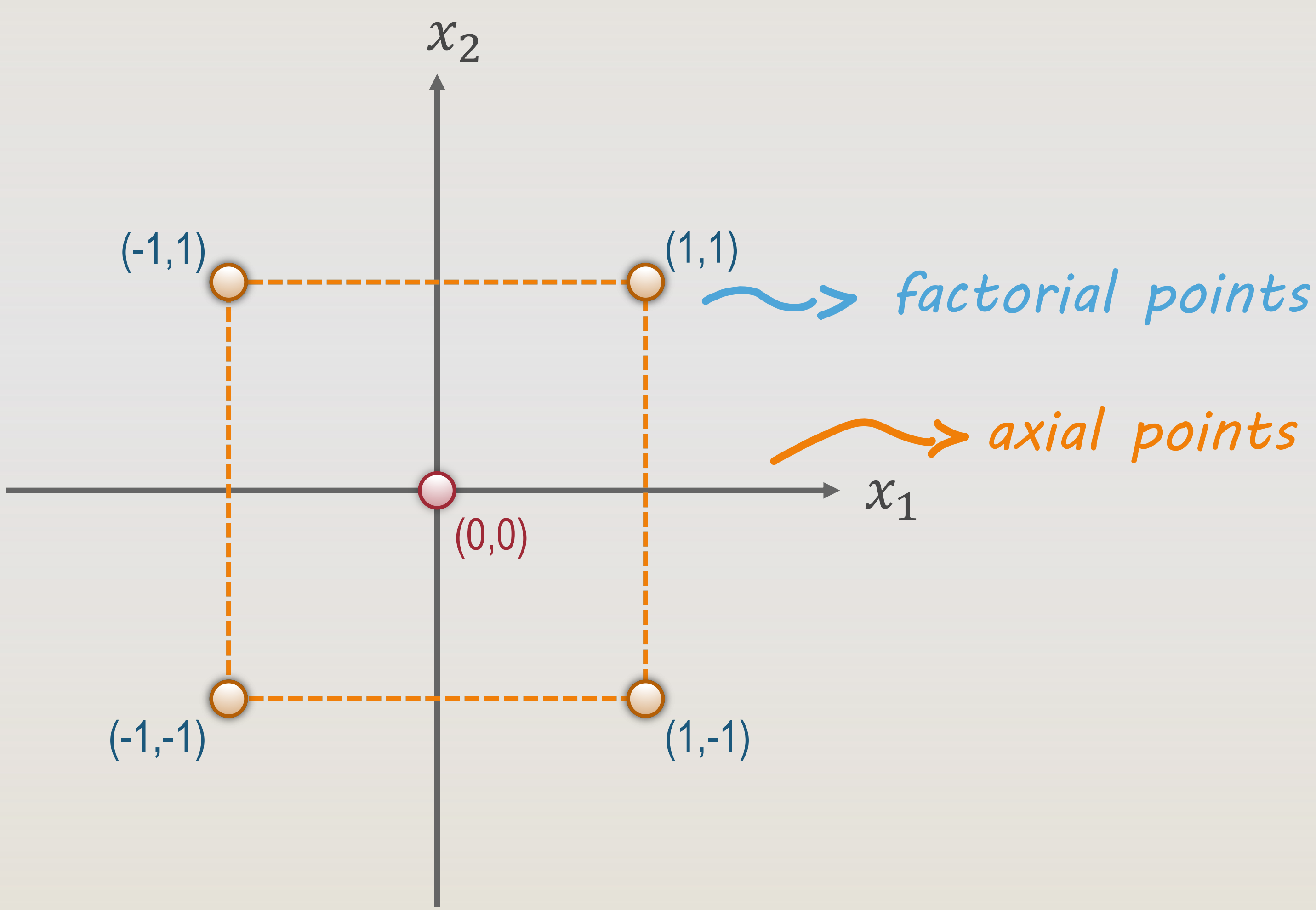
We need at least one independent run for each regression parameter!

The 2^k design with central points can check the model linearity but cannot estimate the quadratic parameters.

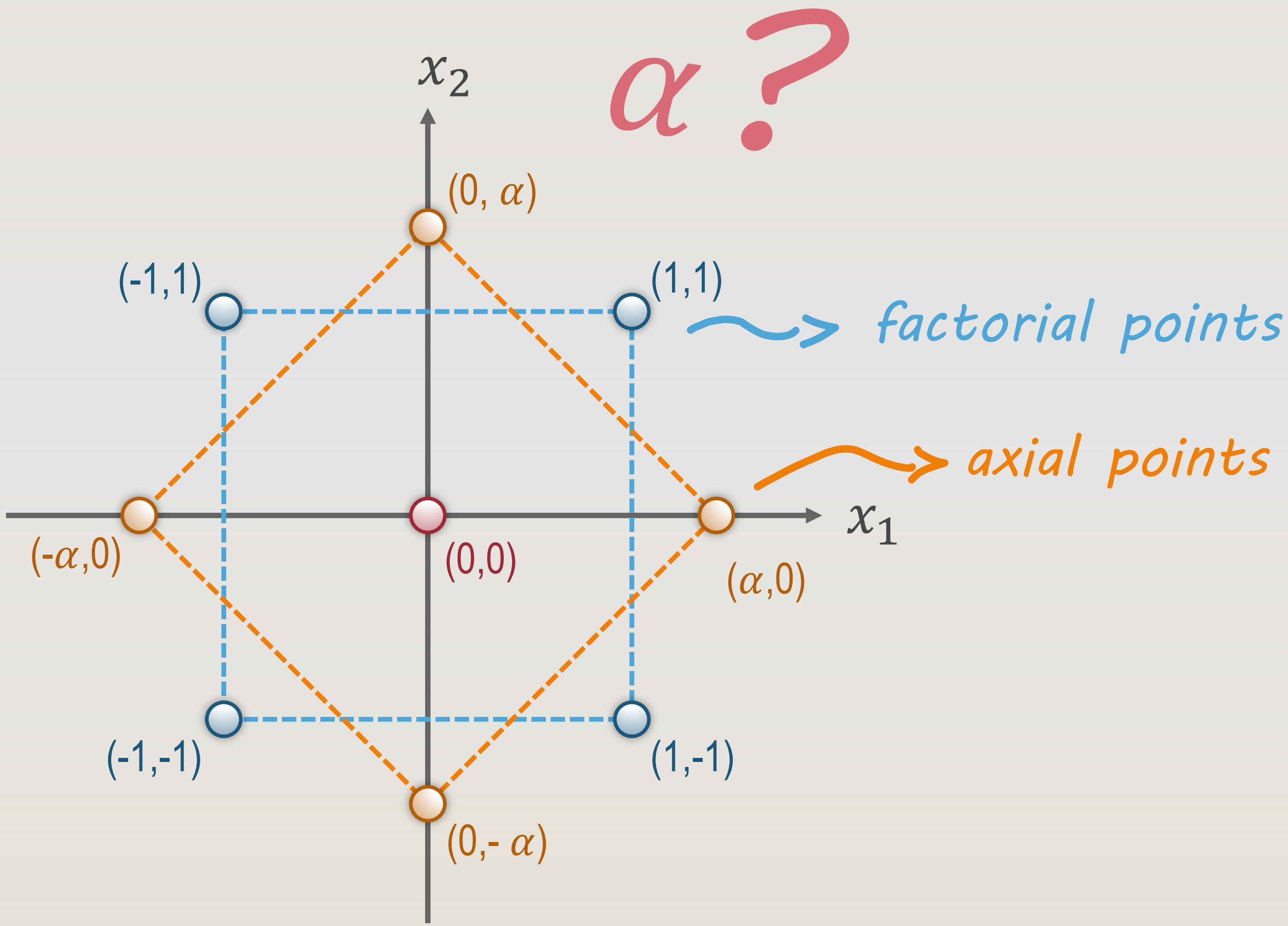
Designs for Second-Order Regression Model

- Central Composite Design -

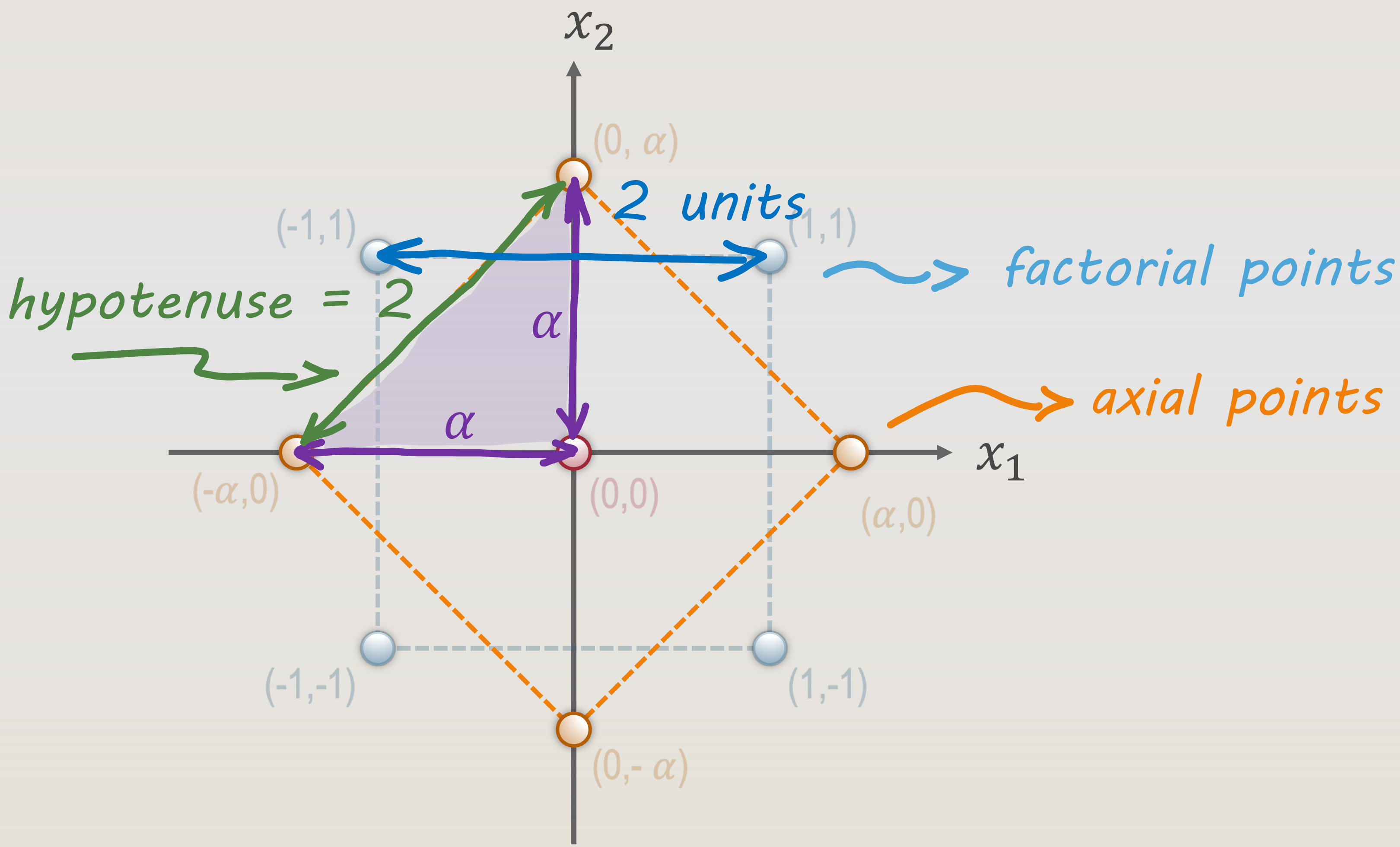
Two-Factor Design for Second-Order Regression Model



Central Composite Design

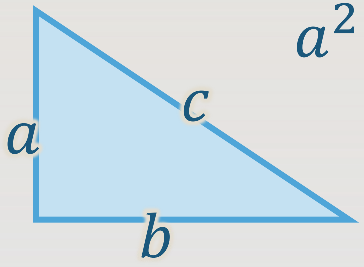


Central Composite Design



Central Composite Design

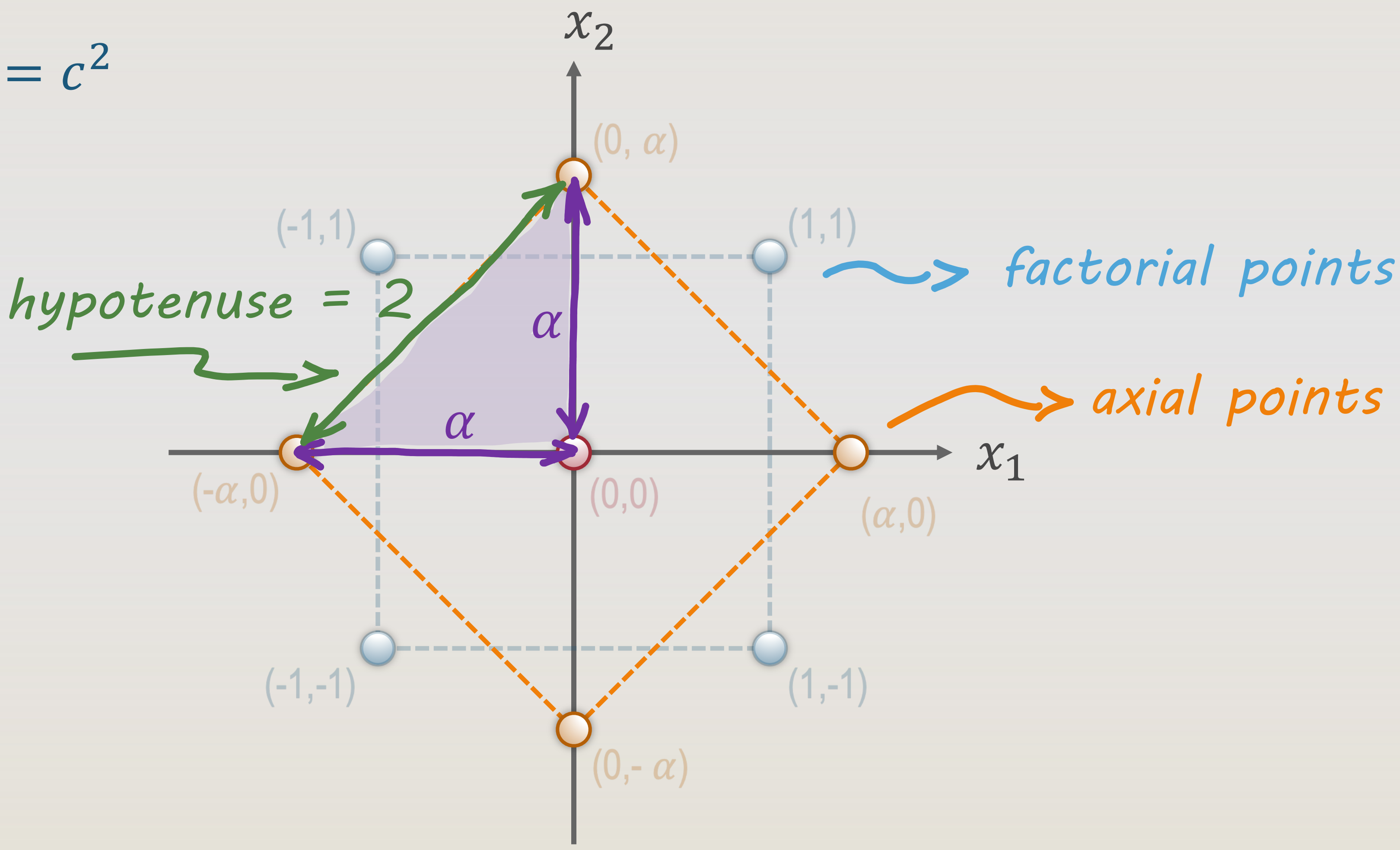
Pythagoras theorem:



$$a^2 + b^2 = c^2$$

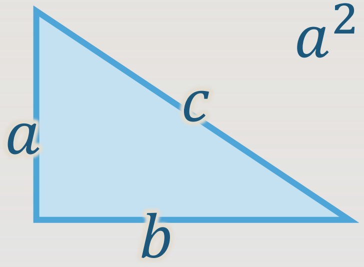
$$\alpha^2 + \alpha^2 = 2^2$$

$$\alpha = \sqrt{2} \cong 1.414$$



Central Composite Design

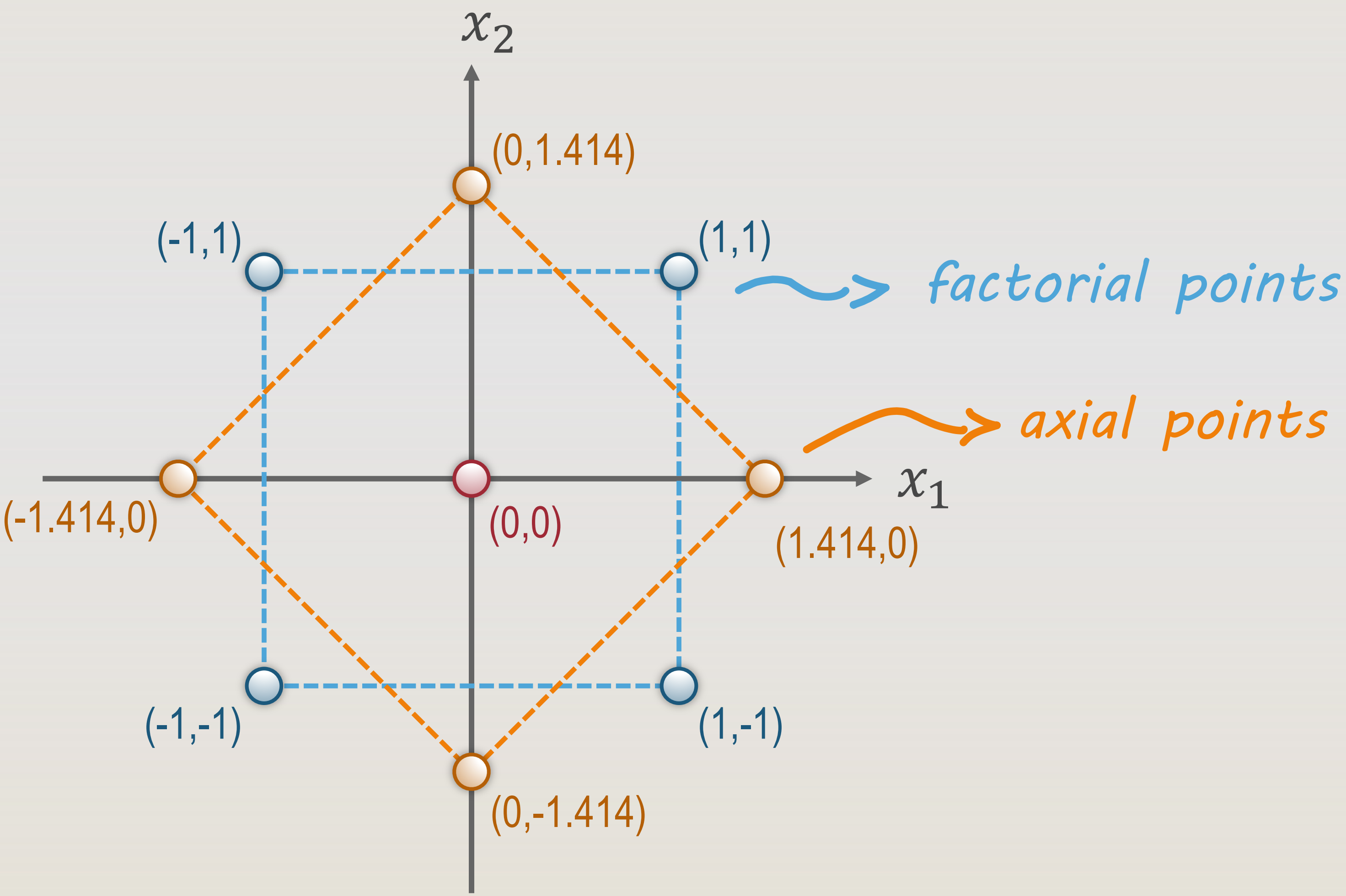
Pythagoras theorem:



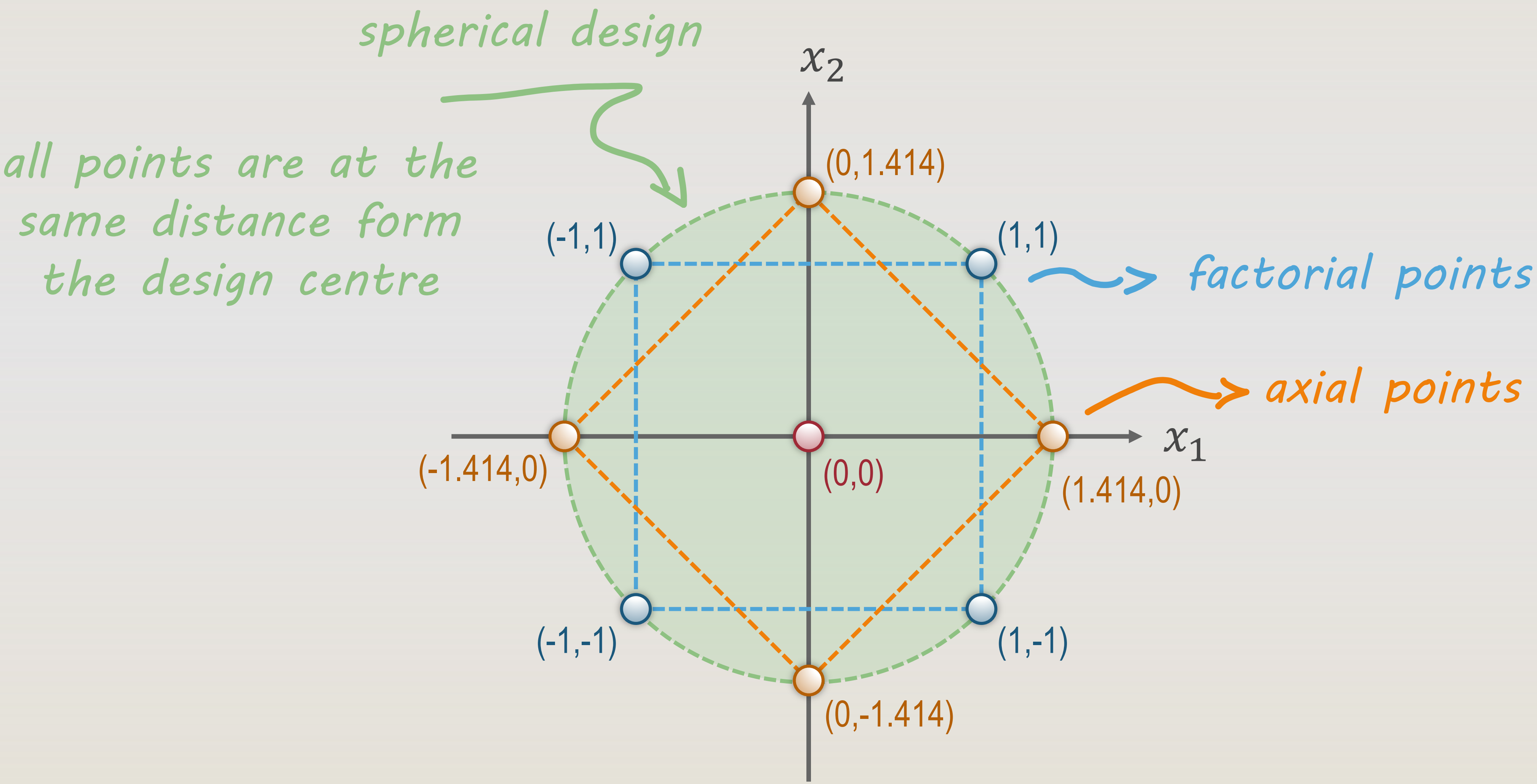
$$a^2 + b^2 = c^2$$

$$\alpha^2 + \alpha^2 = 2^2$$

$$\alpha = \sqrt{2} \cong 1.414$$



Central Composite Design



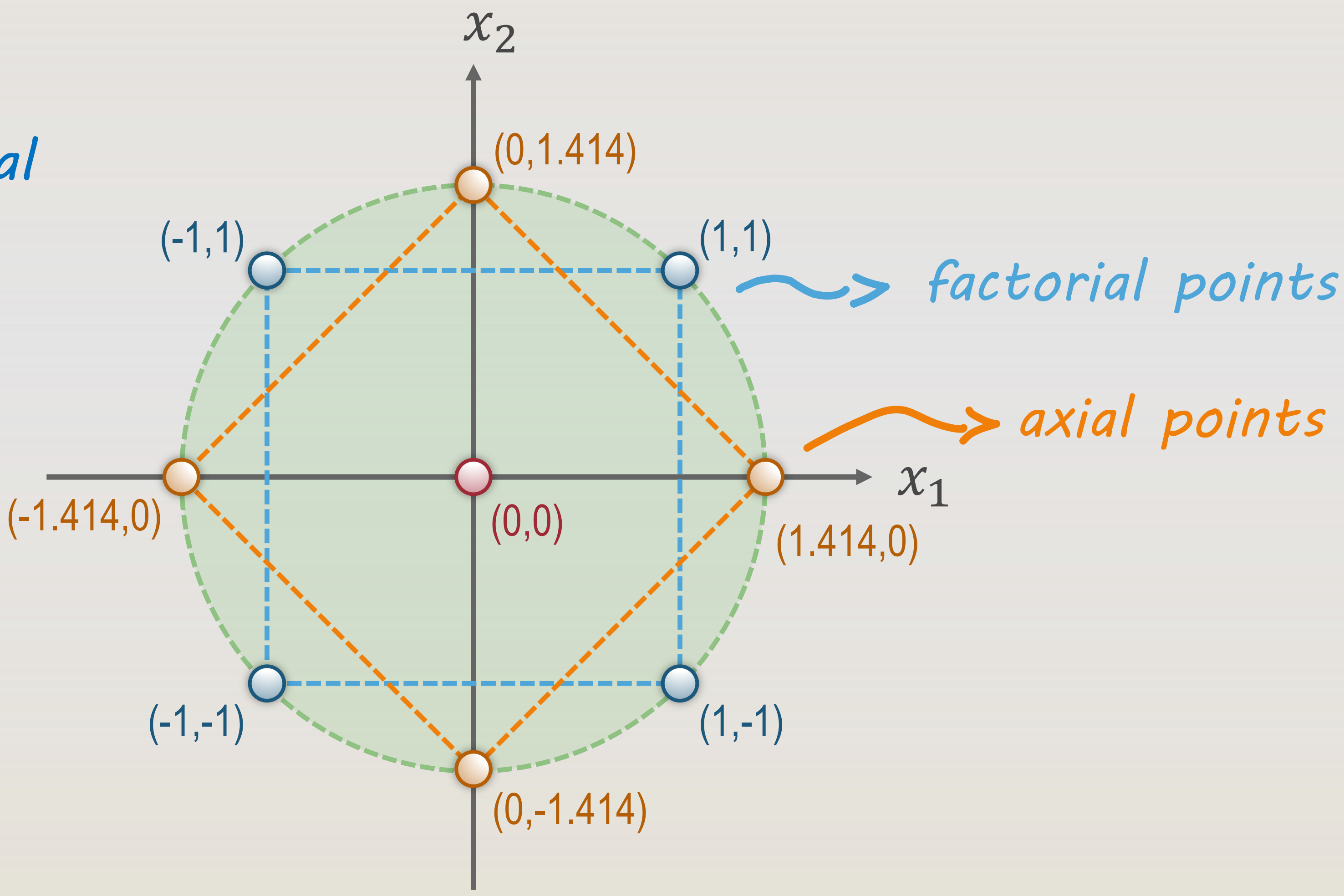
Central Composite Design: Experimental Matrix

x_1	x_2
-1	-1
-1	1
1	-1
1	1
0	0
0	0
0	0
0	0
0	0
1.414	0
-1.414	0
0	1.414
0	-1.414

$n_F = 2^k$ factorial points

n_C central points

$n_A = 2k$ axial points

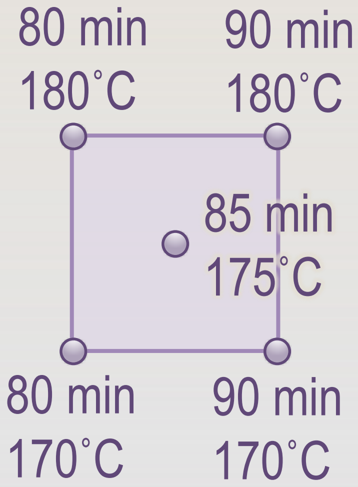


Central Composite Design: Experimental Matrix

x_1	x_2	Time (min)	Temp (˚C)
-1	-1	80	170
-1	1	80	180
1	-1	90	170
1	1	90	180
0	0	85	175
0	0	85	175
0	0	85	175
0	0	85	175
0	0	85	175
1.414	0		
-1.414	0		
0	1.414		
0	-1.414		

factorial points

central points



$$x_1 = \frac{Time - 85}{5}$$

$$x_2 = \frac{Temp - 175}{5}$$

Central Composite Design: Experimental Matrix

x_1	x_2	Time (min)	Temp (˚C)
-1	-1	80	170
-1	1	80	180
1	-1	90	170
1	1	90	180
0	0	85	175
0	0	85	175
0	0	85	175
0	0	85	175
0	0	85	175
1.414	0	92.07	175
-1.414	0	77.93	175
0	1.414	85	182.07
0	-1.414	85	167.93

factorial points

central points

axial points

Relationship between coded and natural variables:

α	Time (min) $x_1 = \frac{Time - 85}{5}$	Temp (˚C) $x_2 = \frac{Temp - 175}{5}$
1.414	92.07	182.07
-1.414	77.93	167.93

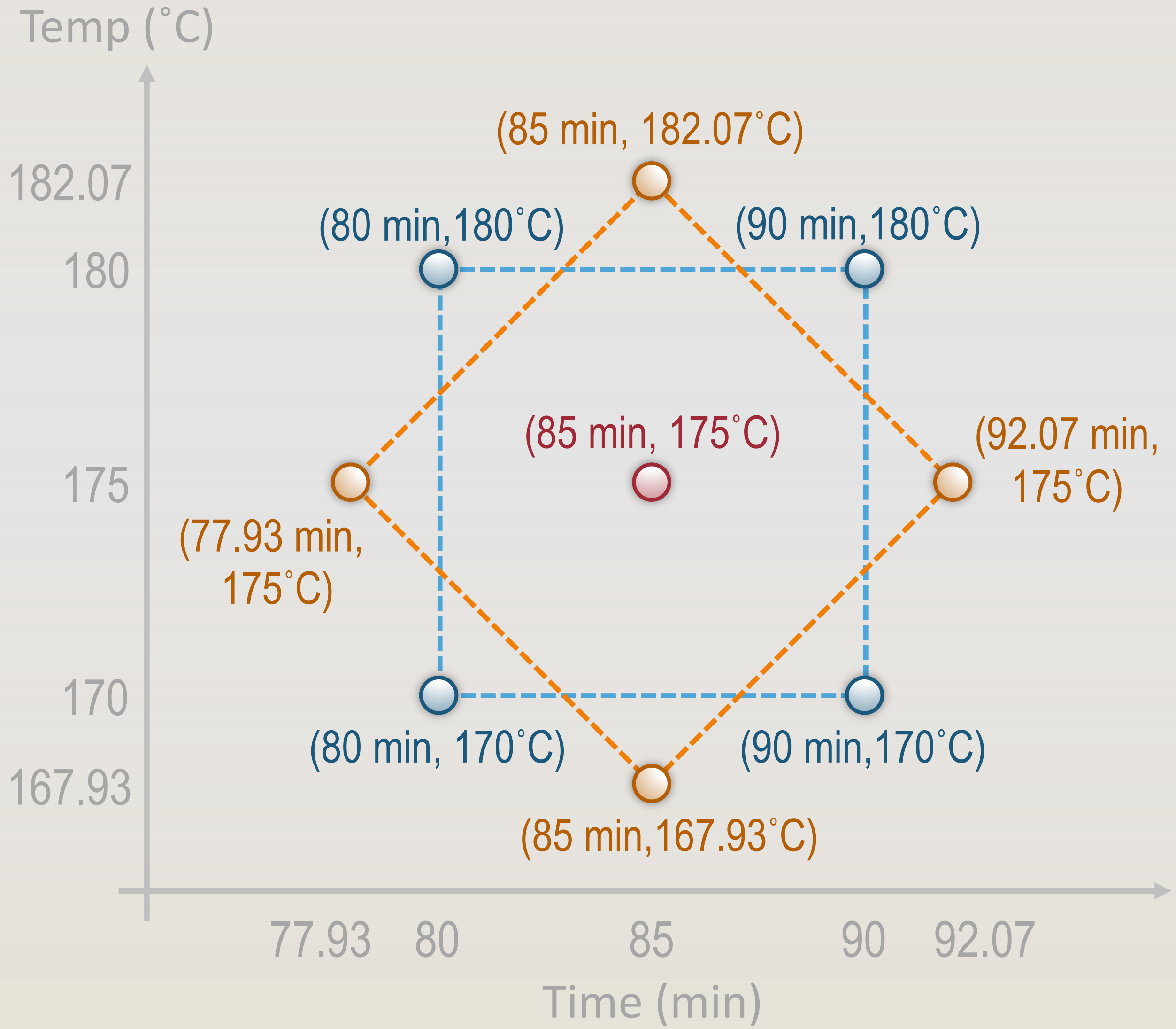
Central Composite Design: Experimental Matrix

x_1	x_2	Time (min)	Temp (˚C)
-1	-1	80	170
-1	1	80	180
1	-1	90	170
1	1	90	180
0	0	85	175
0	0	85	175
0	0	85	175
0	0	85	175
0	0	85	175
1.414	0	92.07	175
-1.414	0	77.93	175
0	1.414	85	182.07
0	-1.414	85	167.93

factorial points

central points

axial points

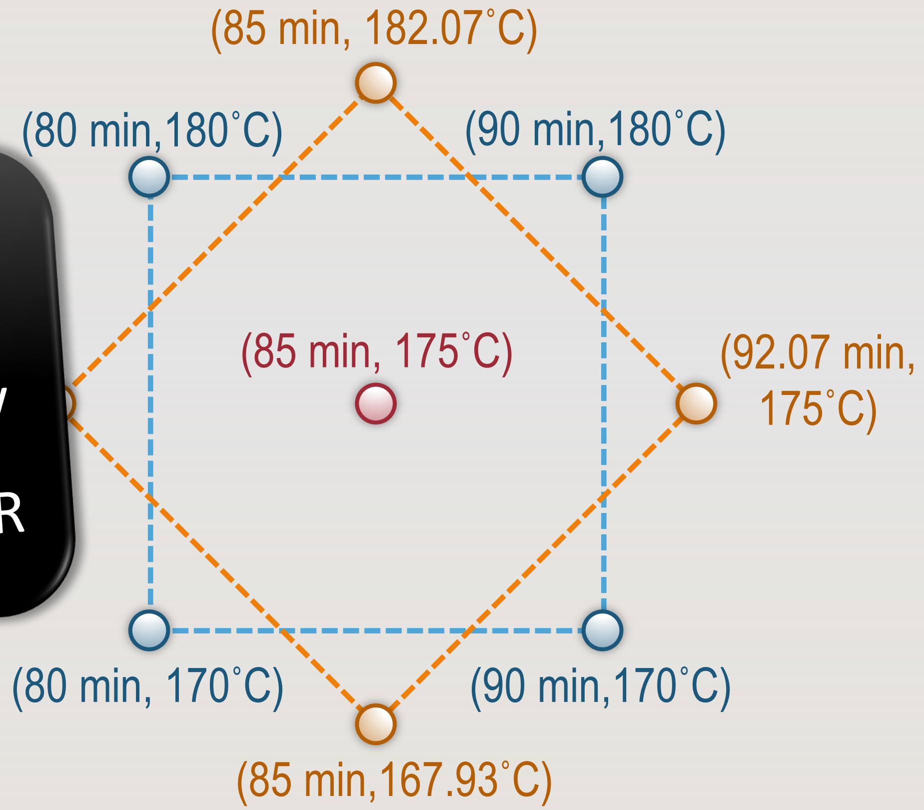


Central Composite Design: Experimental Matrix

x_1	x_2	Time (min)	Temp (˚C)	Yield (%)
−1	−1	80	170	76.5
−1	1	80	180	77.0
1	−1	90	170	78.0
1	1	90	180	[illegible]
0	0	85	175	[illegible]
0	0	85	175	[illegible]
0	0	85	175	[illegible]
0	0	85	175	[illegible]
0	0	85	175	[illegible]
1.414	0	92.07	175	78.4
−1.414	0	77.93	175	75.6
0	1.414	85	182.07	78.5
0	−1.414	85	167.93	77.0

Using R-Studio:

- Data file: DoEOpt06.csv
- R code file: DoEOpt06.R



Building a Central Composite Design in R

Building a Central Composite Design in R

```
# building a central composite design:

dsg <- ccd(2, alpha = “spherical”)
```

central composite design function

axial points -> $\alpha = \sqrt{k}$

number of factors -> 2

```r
# Way 1: Build a 2^k + central points than add the axial (star) points:

dsg <- cube(2, n0 = 5)
dsg <- djoin(dsg, star(alpha = "spherical"))
```

builds a 2^2 design with 5 central points

adds the star (axial) points

```r
# Way 2: Build a central composite design

dsg <- ccd(2, n0=3, alpha = "spherical")
```

builds a complete central composite design

Analysing a Central Composite Design

R Tutorial

second-order model

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2$$

```
# regression model for Yield

model_Y <- rsm(Y ~ SO(x1,x2), data = DoEOpt06)

# or

model_Y <- rsm(Y ~ FO(x1,x2) + TWI(x1,x2) + PQ(x1,x2), data = DoEOpt06)
```

first order model — $\beta_0 + \beta_1 x_1 + \beta_2 x_2$

two-way interaction — $\beta_{12} x_1 x_2$

pure quadratic — $\beta_{11} x_1^2 + \beta_{22} x_2^2$

remove the interaction

```
rsm(formula = Y ~ FO(x1, x2) + TWI(x1, x2) + PQ(x1, x2), data = CCD1)

             Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 79.939955   0.119089 671.2644 < 2.2e-16 ***
x1           0.995050   0.094155  10.5682 1.484e-05 ***
x2           0.515203   0.094155   5.4719  0.000934 ***
```

first-order (linear)

```
x1:x2        0.250000   0.133145   1.8777  0.102519
x1^2        -1.376449   0.100984 -13.6303 2.693e-06 ***
x2^2        -1.001336   0.100984  -9.9158 2.262e-05 ***
```

second-order (quadratic)

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Multiple R-squared:  0.9827,     Adjusted R-squared:  0.9704
F-statistic: 79.67 on 5 and 7 DF,  p-value: 5.147e-06
```

high R^2 and low p-value

```
Analysis of Variance Table

Response: Y
            Df  Sum Sq Mean Sq  F value    Pr(>F)
FO(x1, x2)   2 10.0430  5.0215  70.8143 2.267e-05
```

first-order (linear)

```
TWI(x1, x2)  1  0.2500  0.2500   3.5256    0.1025
PQ(x1, x2)   2 17.9537  8.9769 126.5944 3.194e-06
```

second-order (quadratic)

```
Residuals    7  0.4964  0.0709
Lack of fit  3  0.2844  0.0948   1.7885    0.2886
Pure error   4  0.2120  0.0530
```

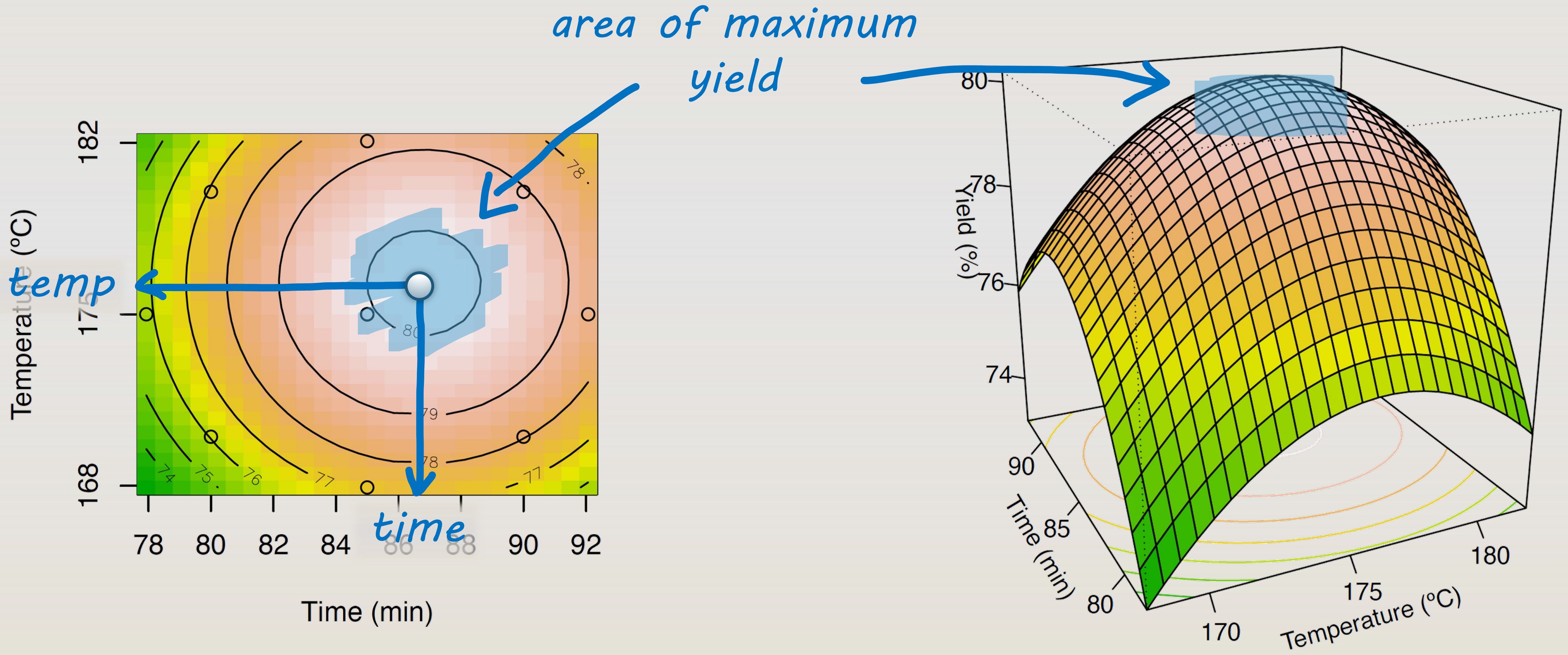
LOF not significant

```
rsm(formula = Y ~ FO(x1, x2) + PQ(x1, x2), data = CCD1)

             Estimate Std. Error  t value  Pr(>|t|)    
(Intercept) 79.93995    0.13660 585.2155 < 2.2e-16 ***
x1           0.99505    0.10800   9.2135 1.559e-05 ***
x2           0.51520    0.10800   4.7704  0.001408 ** 
x1^2        -1.37645    0.11583 -11.8831 2.310e-06 ***
x2^2        -1.00134    0.11583  -8.6447 2.489e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Multiple R-squared:  0.974,     Adjusted R-squared:  0.961 
F-statistic: 75.02 on 4 and 8 DF,  p-value: 2.226e-06

Analysis of Variance Table

Response: Y
            Df  Sum Sq Mean Sq F value    Pr(>F)
FO(x1, x2)   2 10.0430  5.0215 53.8227 2.290e-05
PQ(x1, x2)   2 17.9537  8.9769 96.2186 2.538e-06
Residuals    8  0.7464  0.0933                  
Lack of fit  4  0.5344  0.1336  2.5206    0.1962
Pure error   4  0.2120  0.0530                  
```

all regression coefficients are significant

high R^2 and the lowest p-value

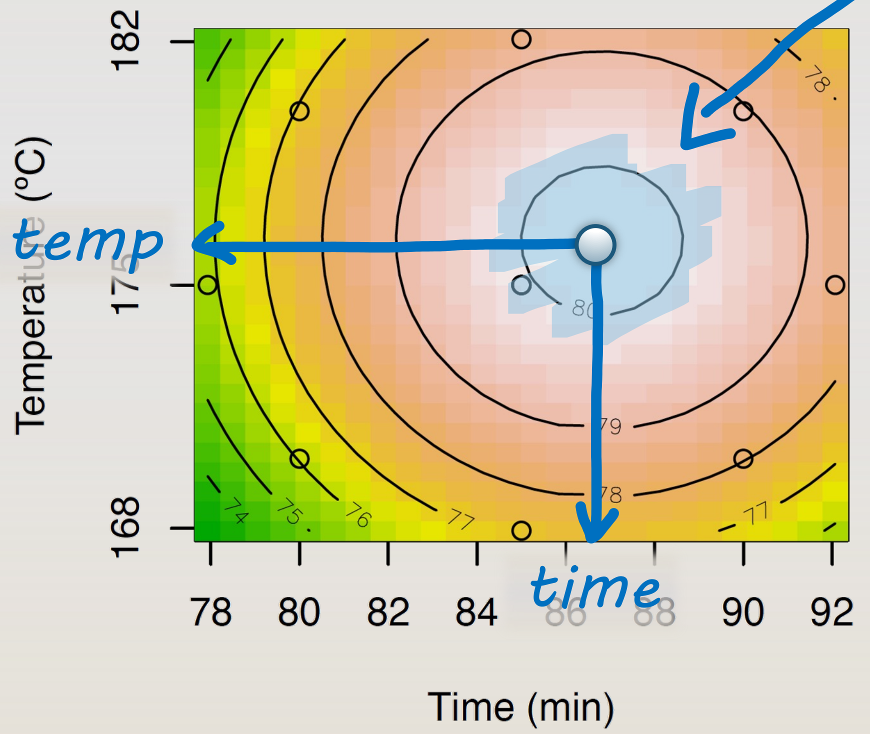
LOF not significant


area of maximum yield
temp
time
Temperature (°C)
Time (min)
182
175
168
78
80
82
84
86
88
90
92
74
75
76
77
78
79
80
Yield (%)
Time (min)
Temperature (°C)
170
175
180
85
90

$$y = 79.94 + 0.995x_1 + 0.51x_2 - 1.38x_1^2 - x_2^2$$

The maximum response,
or stationary point (if exists): $\dfrac{\delta y}{\delta x_1} = \dfrac{\delta y}{\delta x_2} = 0$

area of maximum yield




```
rsm(formula = Y ~ FO(x1, x2) + PQ(x1, x2), data = CCD1)

             Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 79.93995    0.13660 585.2155 < 2.2e-16 ***
x1           0.99505    0.10800   9.2135 1.559e-05 ***
x2           0.51520    0.10800   4.7704  0.001408 **
x1^2        -1.37645    0.11583 -11.8831 2.310e-06 ***
x2^2        -1.00134    0.11583  -8.6447 2.489e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

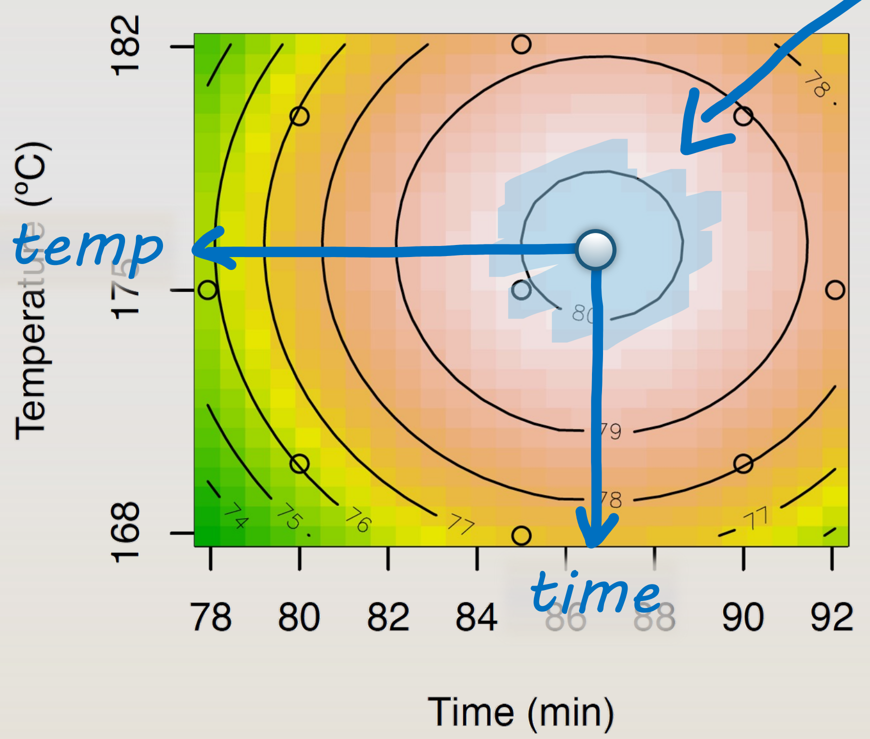
```
             Estimate
(Intercept) 79.93995
x1           0.99505
x2           0.51520
x1^2        -1.37645
x2^2        -1.00134
```

```
Stationary point in original units:
     Time      Temp
 86.80728 176.28629

Eigenanalysis:
eigen() decomposition
$values
[1] -1.001336 -1.376449

$vectors
   [,1] [,2]
x1    0   -1
x2   -1    0
```

$$y = 79.94 + 0.995x_1 + 0.51x_2 - 1.38x_1^2 - x_2^2$$

The maximum response,
or stationary point (if exists): $\dfrac{\delta y}{\delta x_1} = \dfrac{\delta y}{\delta x_2} = 0$

area of maximum yield




```
rsm(formula = Y ~ FO(x1, x2) + PQ(x1, x2), data = CCD1)

             Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 79.93995    0.13660 585.2155 < 2.2e-16 ***
x1           0.99505    0.10800   9.2135 1.559e-05 ***
```

```
Stationary point of response surface:
       x1        x2
0.3614555 0.2572577

Stationary point in original units:
     Time      Temp
 86.80728 176.28629
```

```
Pure error   4  0.2120  0.0530
```

Stationary point of response surface:
x1 x2
0.3614555 0.2572577

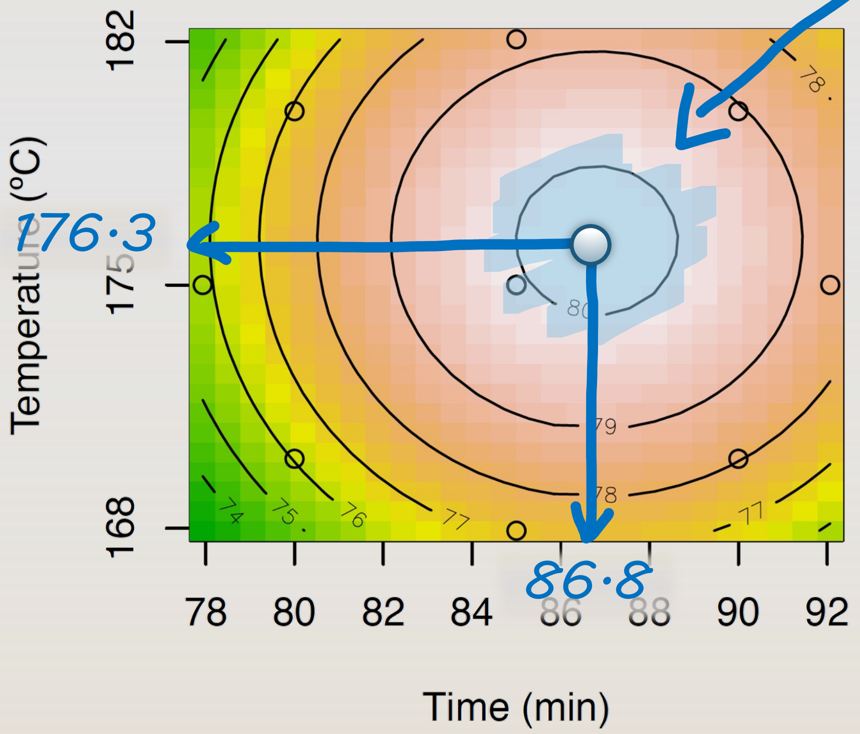
Stationary point in original units:
Time Temp
86.80728 176.28629

```
Eigenanalysis:
eigen() decomposition
$values
[1] -1.001336 -1.376449

$vectors
   [,1] [,2]
x1    0   -1
x2   -1    0
```

$$y = 79.94 + 0.995x_1 + 0.51x_2 - 1.38x_1^2 - x_2^2$$

The maximum response, or stationary point (if exists): $\dfrac{\delta y}{\delta x_1} = \dfrac{\delta y}{\delta x_2} = 0$

area of maximum yield




```
rsm(formula = Y ~ FO(x1, x2) + PQ(x1, x2), data = CCD1)

             Estimate Std. Error  t value  Pr(>|t|)
(Intercept) 79.93995    0.13660 585.2155 < 2.2e-16 ***
x1           0.99505    0.10800   9.2135 1.559e-05 ***
```

```
Pure error    4  0.2120  0.0530

Stationary point of response surface:
       x1        x2
0.3614555 0.2572577

Stationary point in original units:
     Time      Temp
 86.80728 176.28629

Eigenanalysis:
eigen() decomposition
$values
[1] -1.001336 -1.376449

$vectors
   [,1] [,2]
x1    0   -1
x2   -1    0
```


Stationary point of response surface:
x1 x2
0.3614555 0.2572577

Stationary point in original units:
Time Temp
86.80728 176.28629

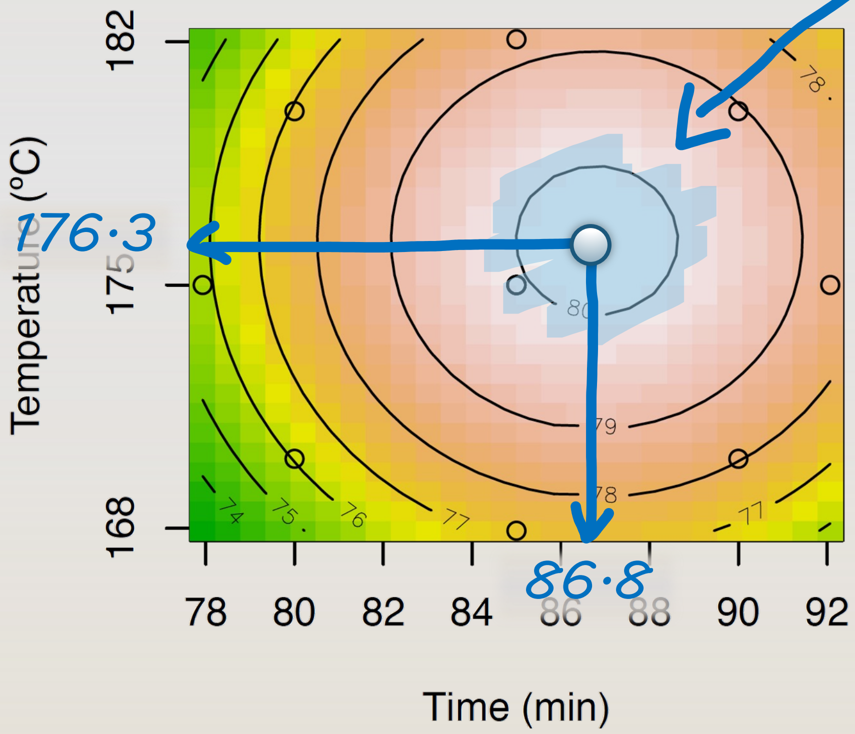
$$y = 79.94 + 0.995x_1 + 0.51x_2 - 1.38x_1^2 \ldots x_2^2$$

The ...

or s...

How much is the maximum predicted yield?

area of maximum yield



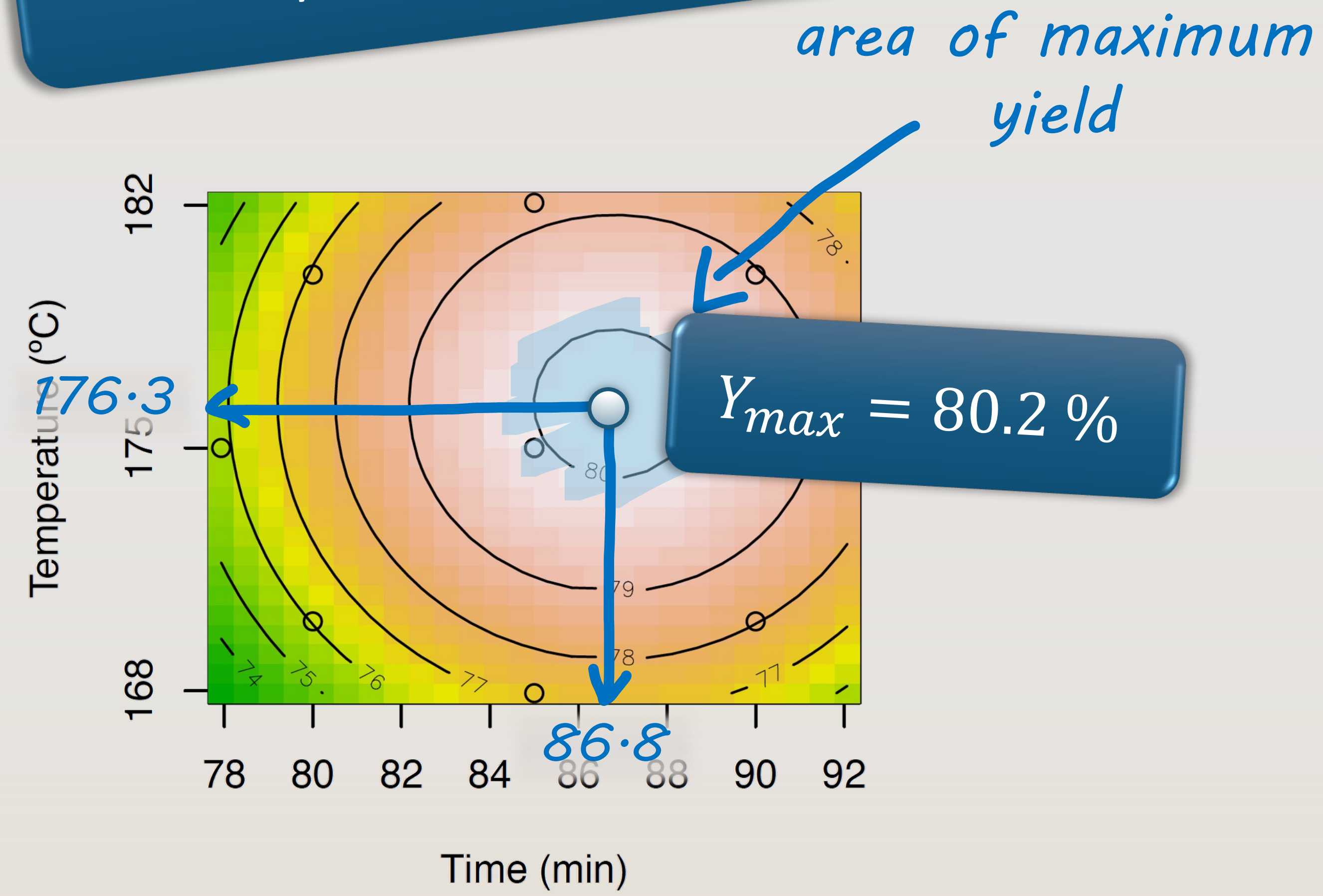

```
Stationary point of response surface:
       x1        x2
0.3614555 0.2572577

Stationary point in original units:
     Time      Temp
 86.80728 176.28629
```

```
# predicting the Yield at the
                    stationary point

max <- data.frame(x1 = 0.361, x2 = 0.257)
predict(model_Y, max)
```

$y = 79.94 + 0.995x_1 + 0.51x_2 - 1.38x_1^2 \ldots x_2^2$

The ...
or s...

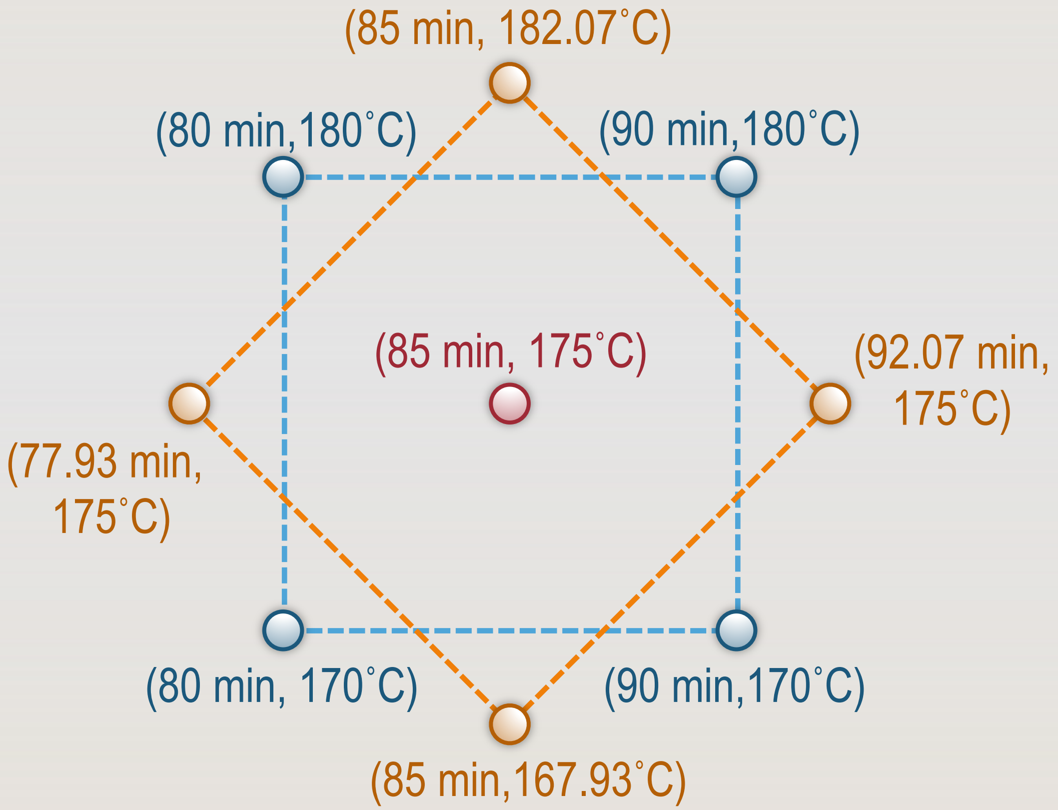
How much is the maximum predicted yield?



One interesting follow up is to run the experiment using the predicted optimised conditions to verify the model's prediction.

Central Composite Design: Multiple Responses

x_1	x_2	Time (s)	Temp (˚C)	Yield (%)
−1	−1	80	170	76.5
−1	1	80	180	77.0
1	−1	90	170	78.0
1	1	90	180	79.5
0	0	85	175	79.9
0	0	85	175	80.3
0	0	85	175	80.0
0	0	85	175	79.7
0	0	85	175	79.8
1.414	0	92.07	175	78.4
−1.414	0	77.93	175	75.6
0	1.414	85	182.07	78.5
0	−1.414	85	167.93	77.0



Central Composite Design: Multiple Responses

molecular weight

x_1	x_2	Time (s)	Temp (˚C)	Yield (%)	Viscosity	M_w
-1	-1	80	170	76.5	62	2940
-1	1	80	180	77.0	60	3470
1	-1	90	170	78.0	66	3680
1	1	90	180	79.5	59	3890
0	0	85	175	79.9	72	3480
0	0	85	175	80.3	69	3200
0	0	85	175	80.0	68	3410
0	0	85	175	79.7	70	3290
0	0	85	175	79.8	71	3500
1.414	0	92.07	175	78.4	68	3360
-1.414	0	77.93	175	75.6	71	3020
0	1.414	85	182.07	78.5	58	3630
0	-1.414	85	167.93	77.0	57	3150

Central Composite Design: Multiple Responses

x_1	x_2	Time (s)	Temp (˚C)	Yield (%)	Viscosity	M_w
−1	−1	80	170	76.5	62	2940
−1	1	80	180	77.0	60	3470
1	−1	90	170	78.0	66	3680
1	1	90	180	79.5	59	3890
0	0	85	175	79.9	72	3480
0	0	85	175	80.3	69	3200
0	0	85	175	80.0	68	3410
0	0	85	175	79.7	70	3290
0	0	85	175	79.8	71	3500
1.414	0	92.07	175	78.4	68	3360
−1.414	0	77.93	175	75.6	71	3020
0	1.414	85	182.07	78.5	58	3630
0	−1.414	85	167.93	77.0	57	3150

Objectives:

- Yield higher than 78.5 %
- Viscosity between 62 and 68 mPa s
- Molecular weight lower than 3400 g/mol

Central Composite Design: Multiple Responses

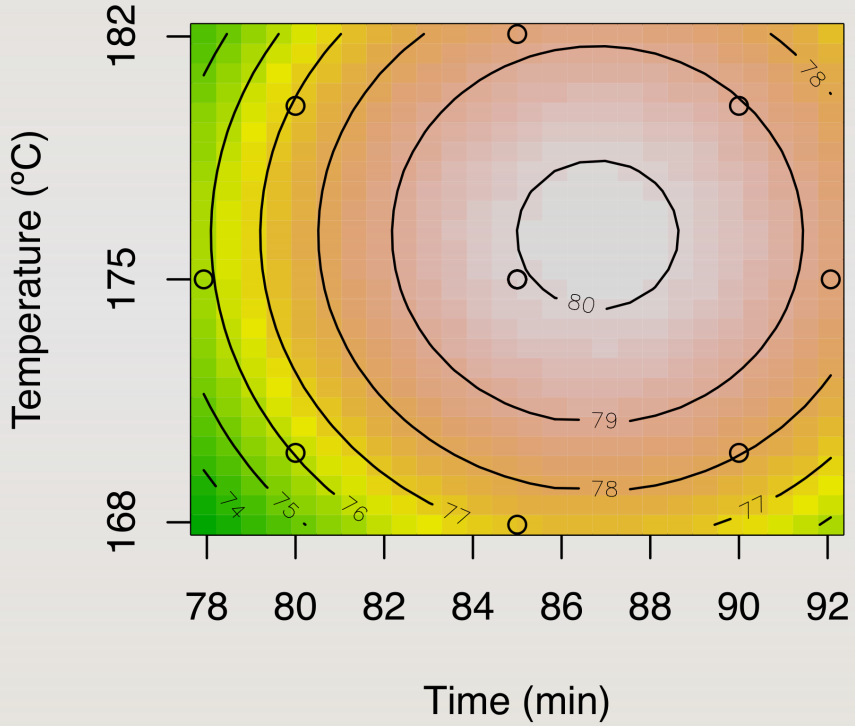
x_1	x_2	Time (s)	Temp (˚C)	Yield (%)	Viscosity	M_w
-1	-1	80	170	76.5	62	2940
-1	1	80	180	77.0	60	3470
1	-1	90	170	78.0	66	3680
1	1	90	180	79.5	59	3890
0	0	85	175	79.9	72	3480
0	0	85	175	80.3	69	3200
0	0	85	175	80.0	68	3410
0	0	85	175	79.7	70	3290
0	0	85	175	79.8	71	3500
1.414	0	92.07	175	78.4	68	3360
-1.414	0	77.93	175	75.6	71	3020
0	1.414	85	182.07	78.5	58	3630
0	-1.414	85	167.93	77.0	57	3150

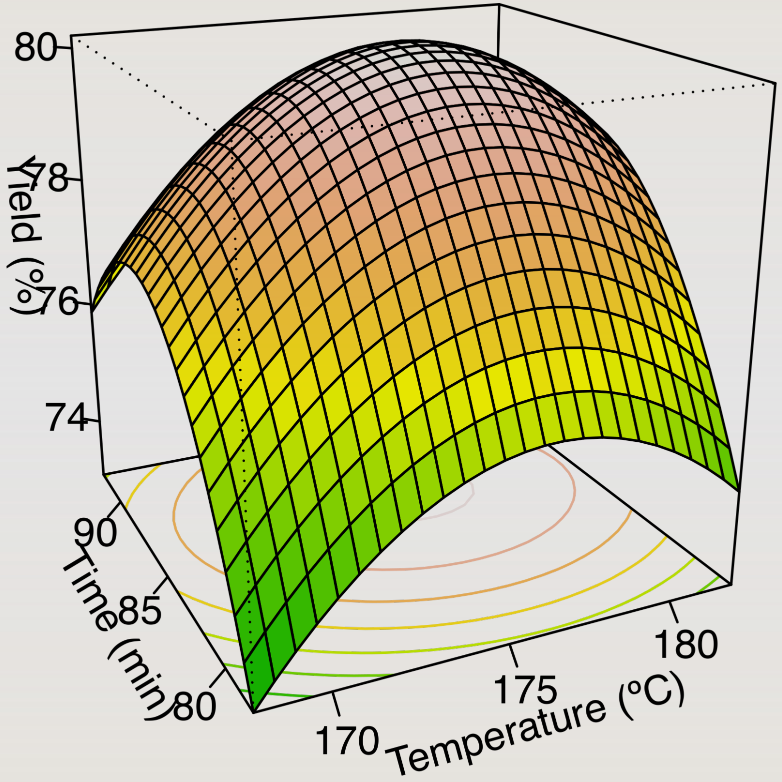
Before starting to watch the next video analyse the **viscosity** and the **molecular weight** data.

To keep in mind:
the coefficients of the first-order (or **linear**) model **cannot be removed** from the regression.

Analysing Multiple Responses

Analysing Multiple Responses: Yield

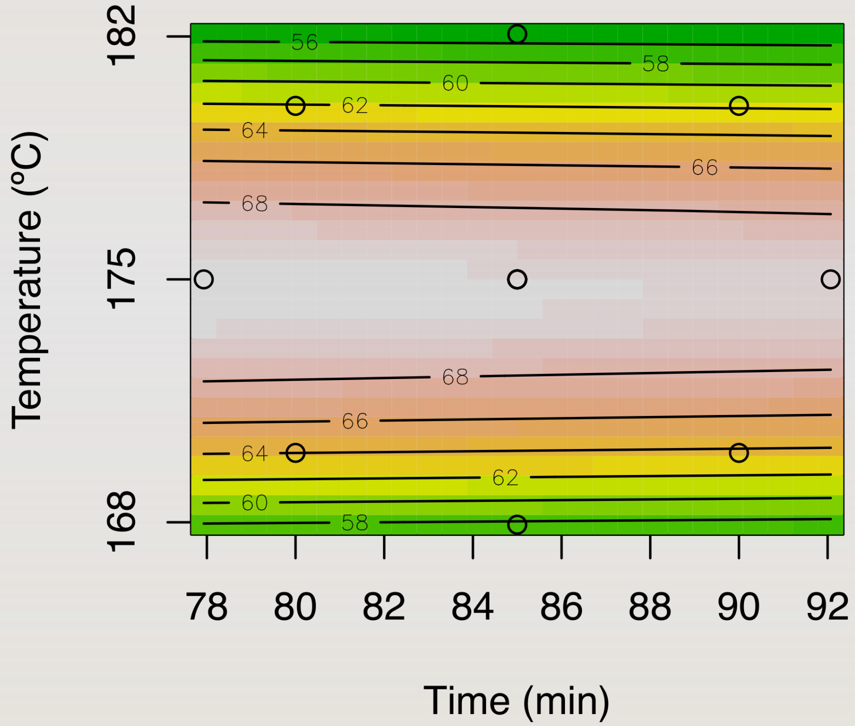


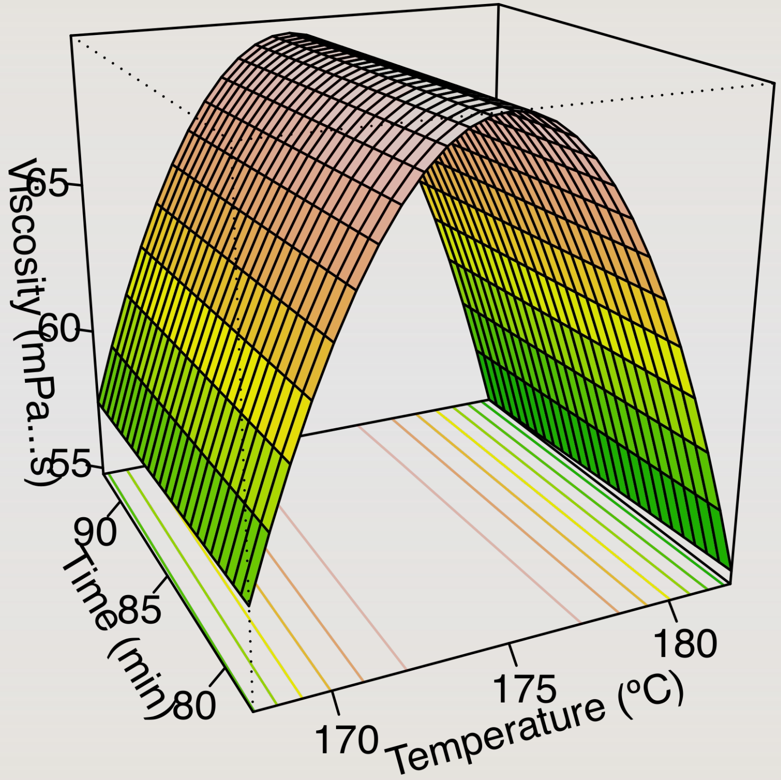


$$Y(\%) = 79.94 + 0.995x_1 + 0.51x_2 - 1.38x_1^2 - x_2^2$$

$$R^2 = 0.974,\ p\text{-value} = 2{\times}10^{-6}$$

Analysing Multiple Responses: Viscosity

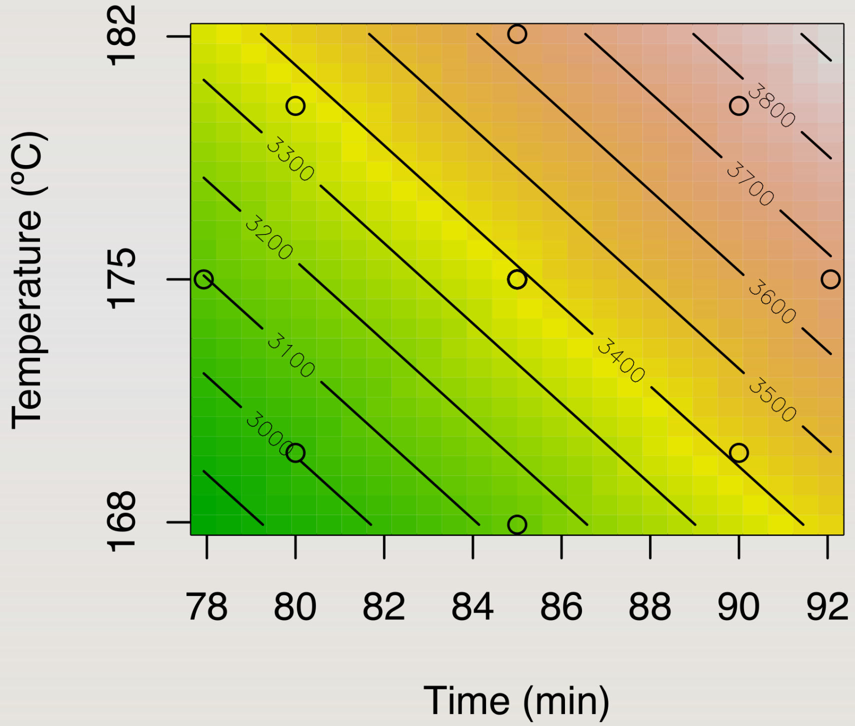


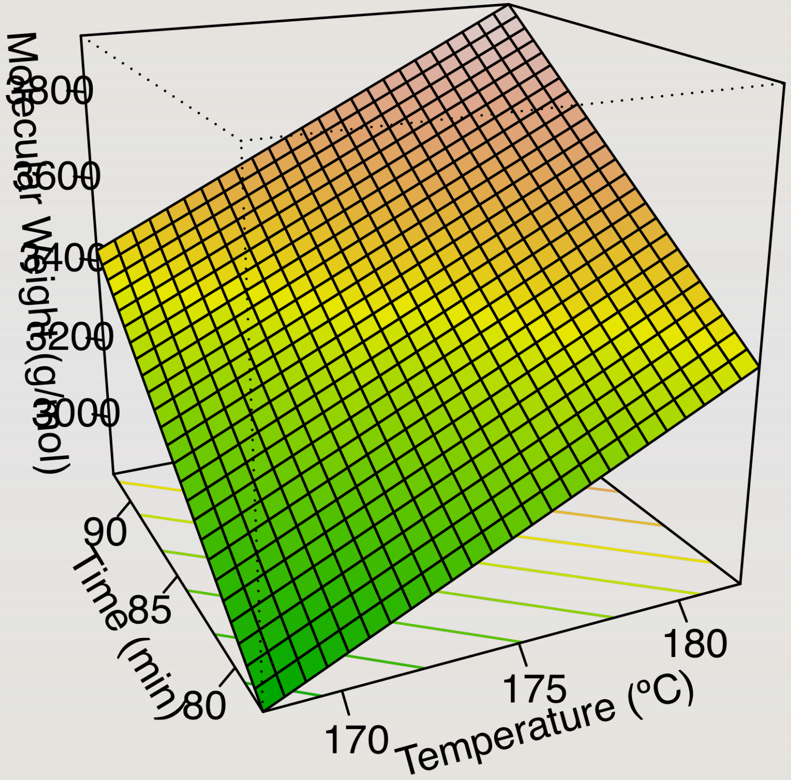


$$Visc = 69.52 - 155x_1 - 0.948x_2 - 6.599x_2^2$$

$$R^2 = 0.873, p\text{-value} = 2{\times}10^{-4}$$

Analysing Multiple Responses: Molecular Weight

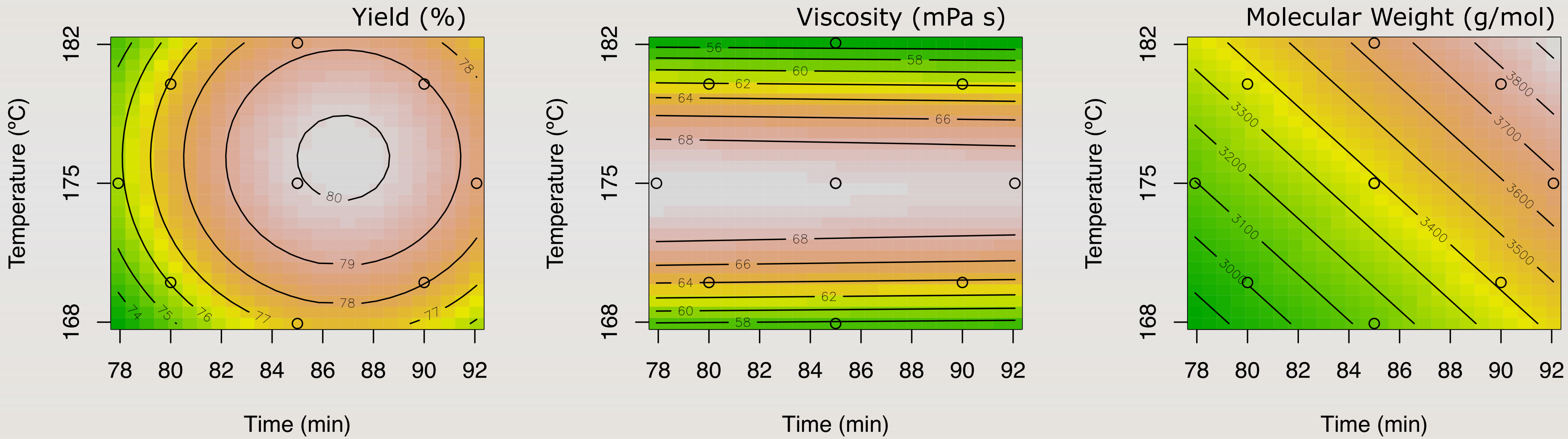




$$M_W = 3386 + 205x_1 + 117x_2$$

$$R^2 = 0.682, p\text{-value} = 0.0032$$

Analysing Multiple Responses

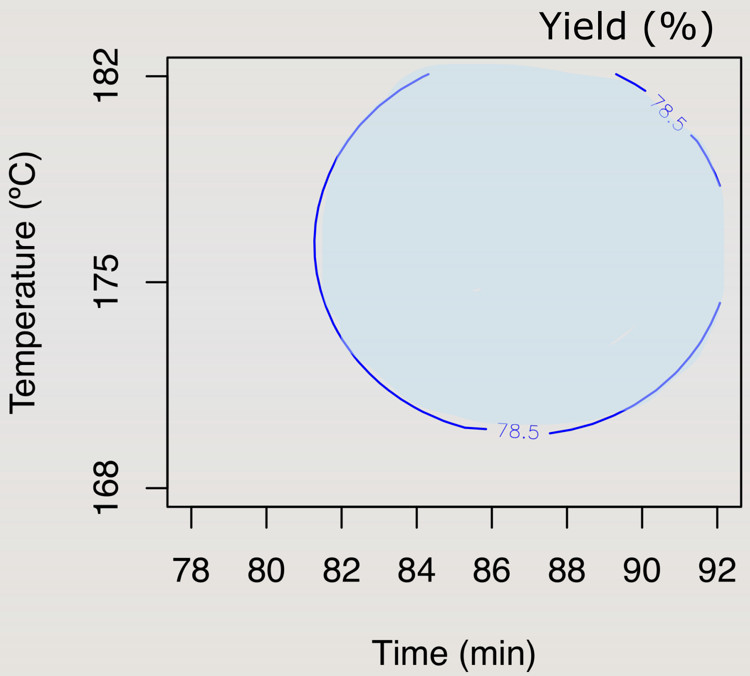


Yield higher than 78.5 %

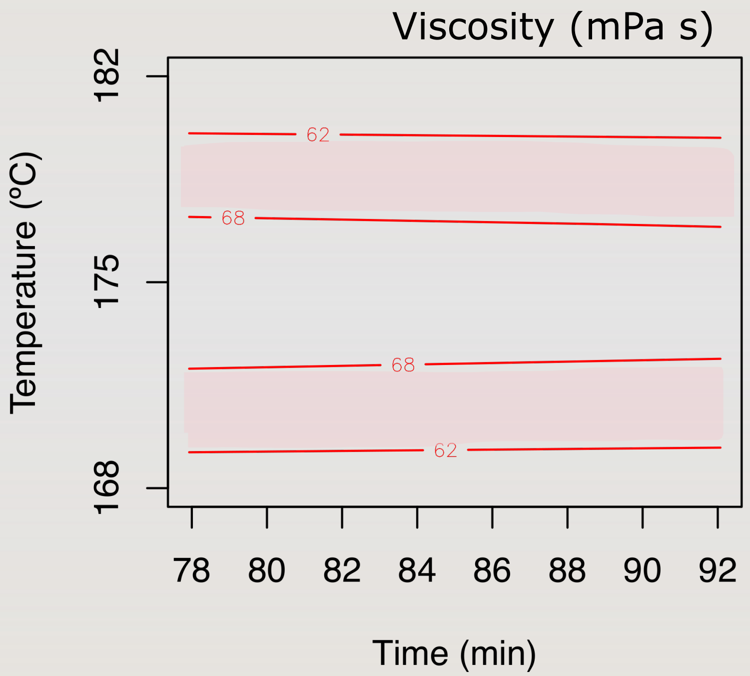
Viscosity between 62 and 68 mPa s

Molecular weight lower than 3400 g/mol

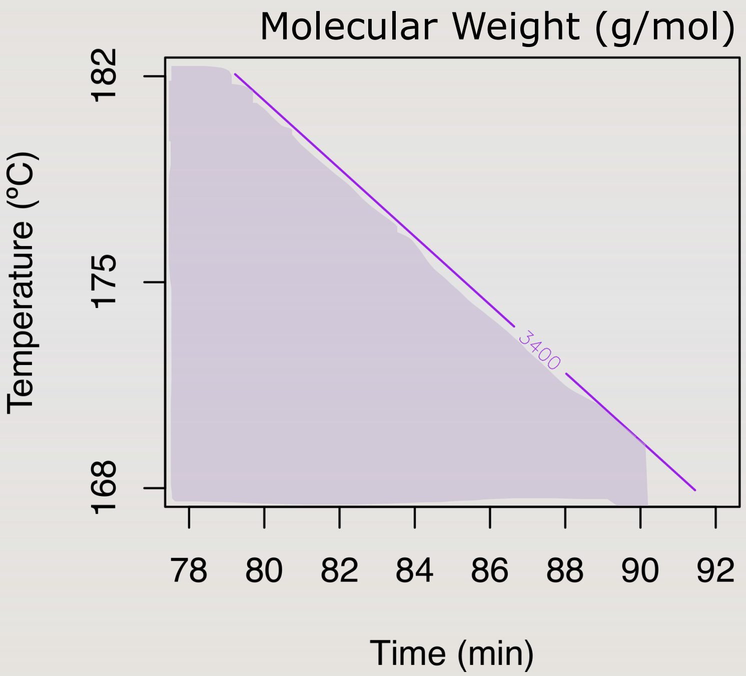
Analysing Multiple Responses



Yield higher than 78.5 %

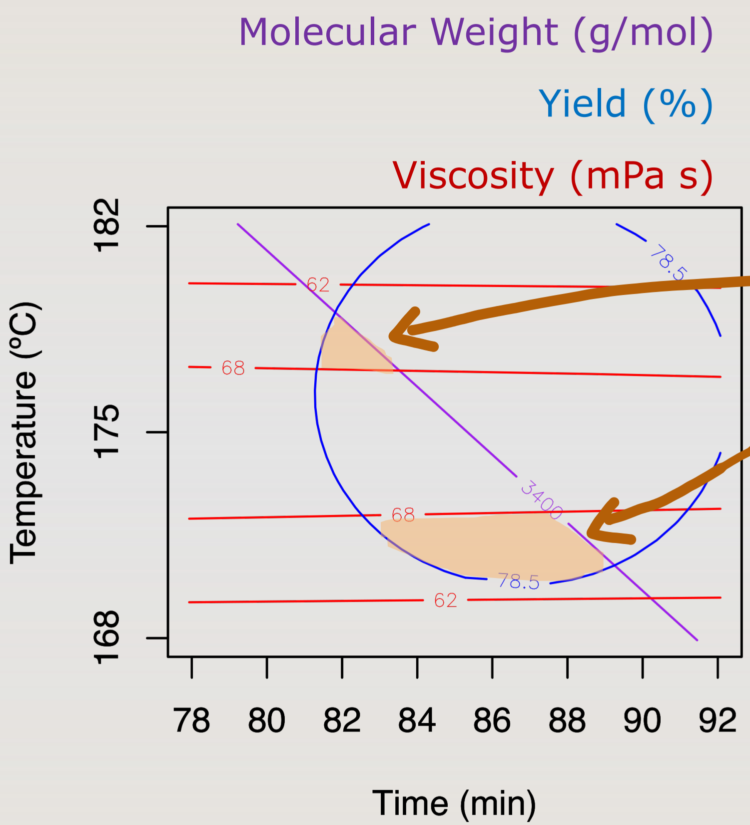


Viscosity between 62 and 68 mPa s



Molecular weight lower than 3400 g/mol

Analysing Multiple Responses



Combination of temperature and time that satisfies simultaneously the goals for yield, viscosity and molecular weight.

Yield higher than 78.5 %

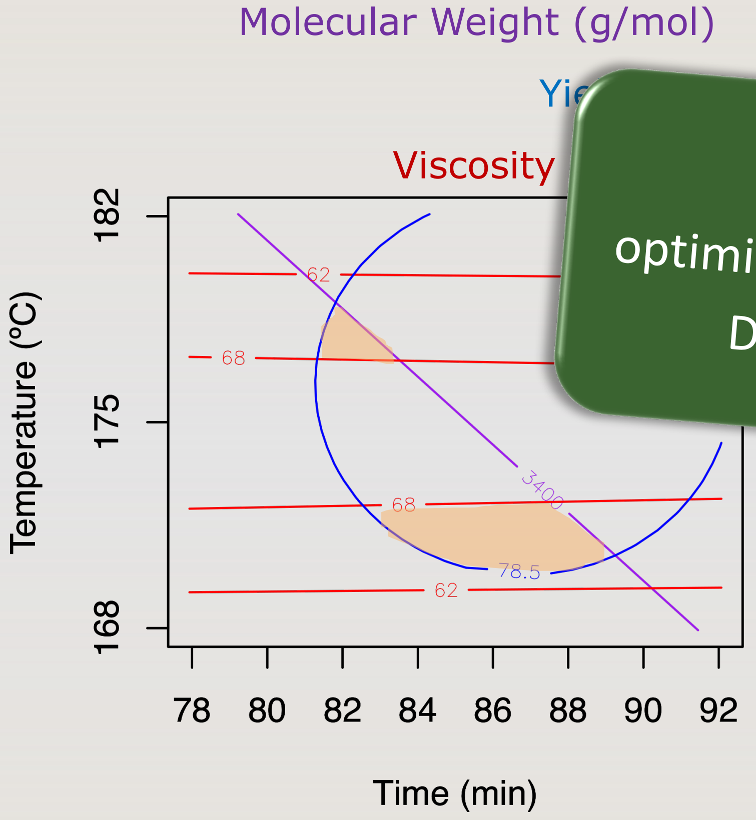
Viscosity between 62 and 68 mPa s

Molecular weight lower than 3400 g/mol

Desirability Functions for Multiple Responses

Analysing Multiple Responses

Is it possible to have mathematical functions instead of overlapping plots manually?



Desirability functions: optimization technique popularized by Derringer and Suich (1980).

Yield higher than 78.5 %

Viscosity between 62 and 68 mPa s

Molecular weight lower than 3400 g/mol

Desirability Function for Multiple Responses

The general approach is to first convert each response y_i into an individual desirability function d_i that varies over the range 0 to 1:

$$y_i \rightarrow 0 \leq d_i \leq 1$$

response variable

yield, viscosity, and molecular weight

y_i is at its goal or target $\rightarrow d_i = 1$

y_i is outside an acceptable region $\rightarrow d_i = 0$

the individual desirability functions can be structured to maximise, minimise or for a target value of the response variable

Desirability Function for Multiple Responses

The general approach is to first convert each response y_i into an individual desirability function d_i that varies over the range 0 to 1:

$$y_i \rightarrow 0 \leq d_i \leq 1$$

y_i is at its goal or target $\rightarrow d_i = 1$

y_i is outside an acceptable region $\rightarrow d_i = 0$

The objective is to maximize the overall desirability D:

$$D = (d_1 \cdot d_2 \cdot \cdots \cdot d_m)^{\frac{1}{m}}$$

m: responses

The overall desirability will be zero if any of the individual responses is undesirable.

Desirability Function for Multiple Responses

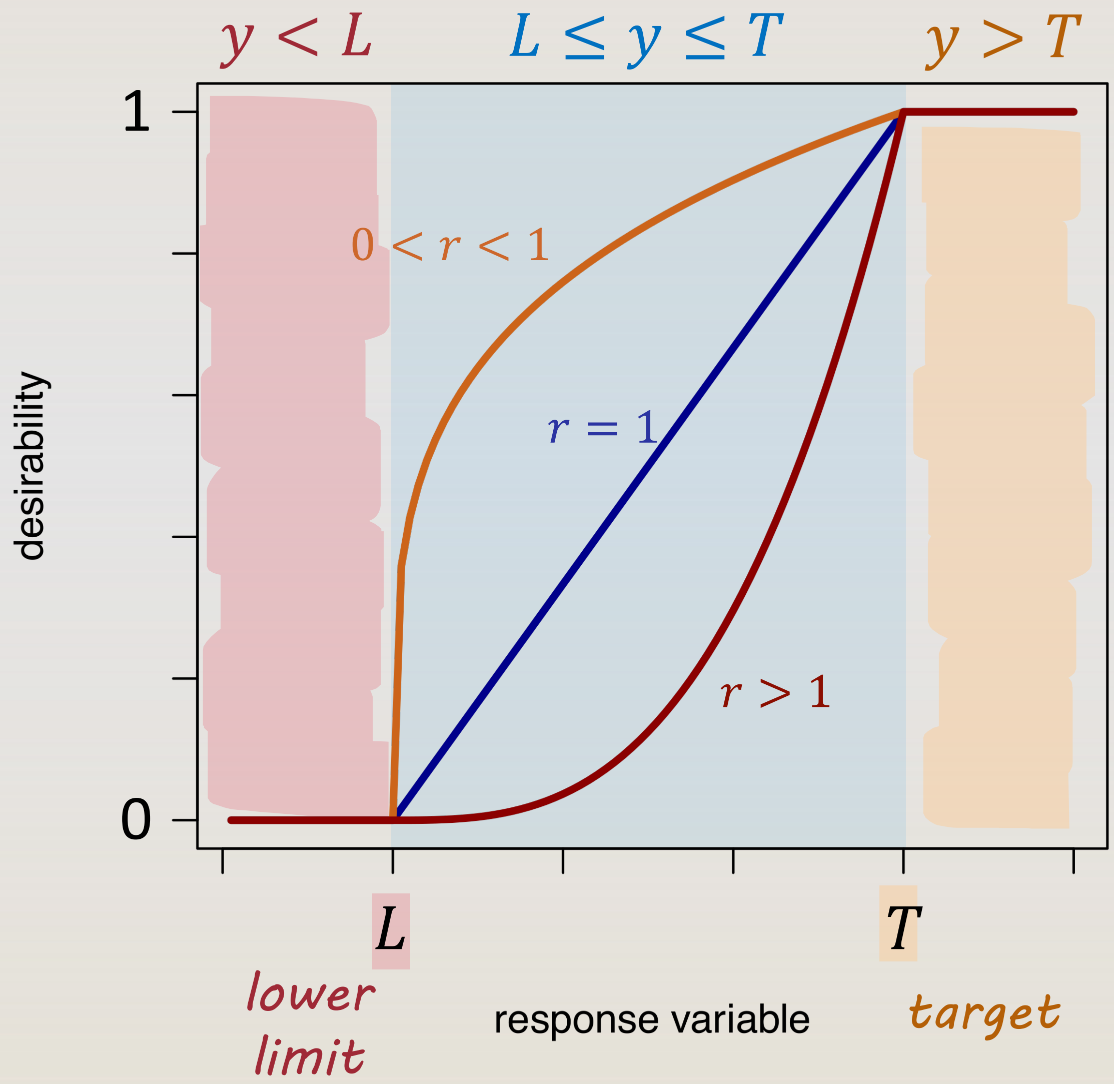
Maximise y:

weight

$$d = \begin{cases} 0 & y < L \\ \left(\frac{y-L}{T-L}\right)^{r} & L \le y \le T \\ 1 & y > T \end{cases}$$

lower limit

target



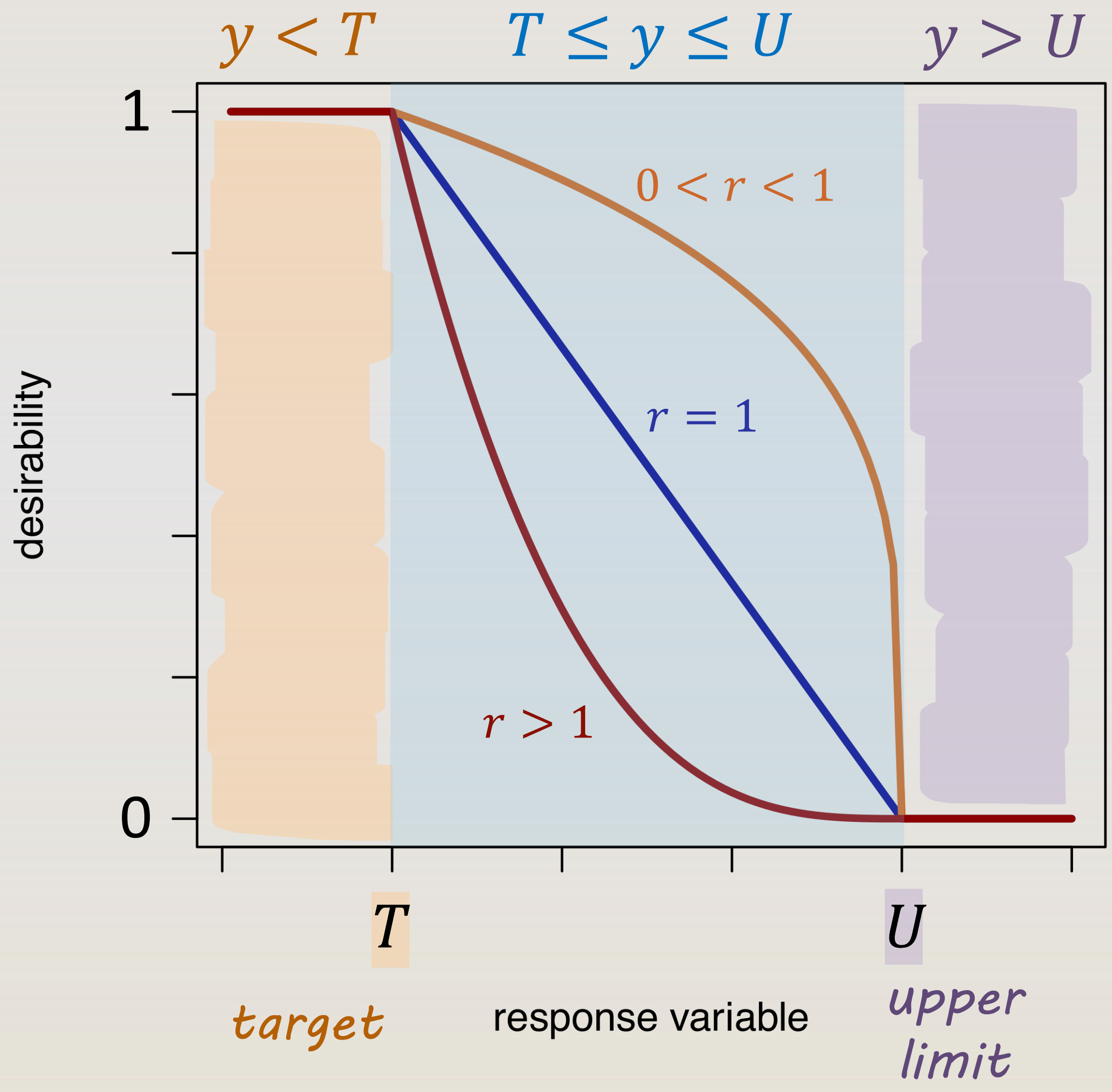
Desirability Function for Multiple Responses

Minimise y:

$$d = \begin{cases} 1 & y < T \\ \left(\dfrac{U-y}{U-T}\right)^r & T \le y \le U \\ 0 & y > U \end{cases}$$

target

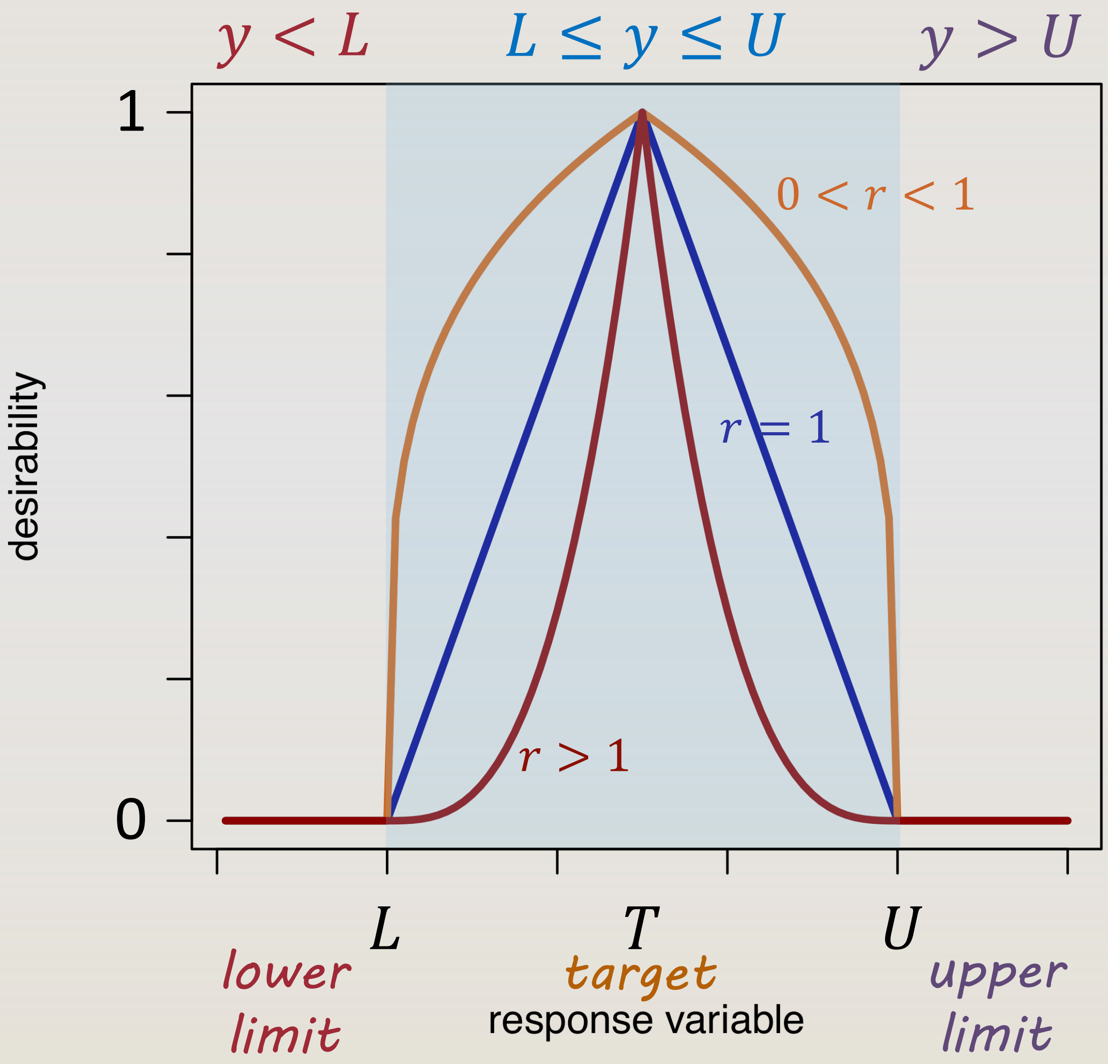
upper limit



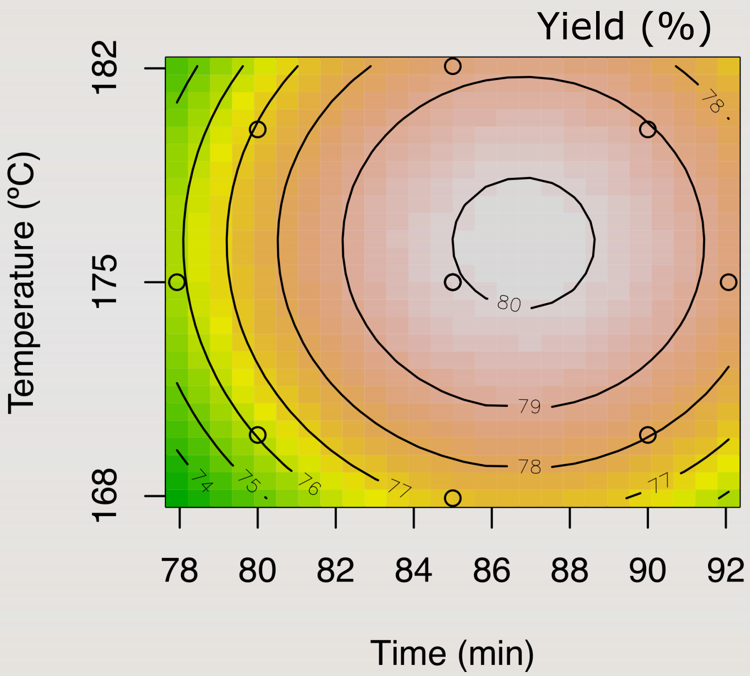
Desirability Function for Multiple Responses

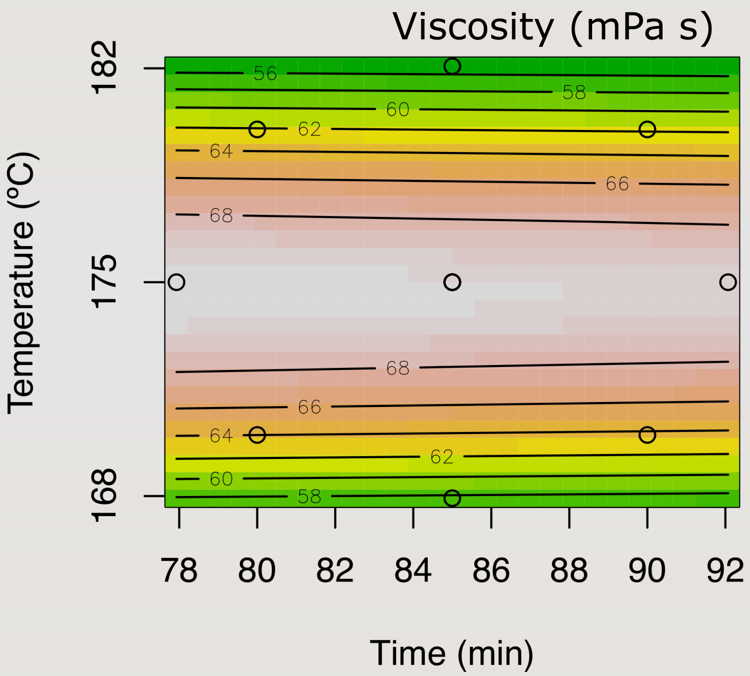
Target y:

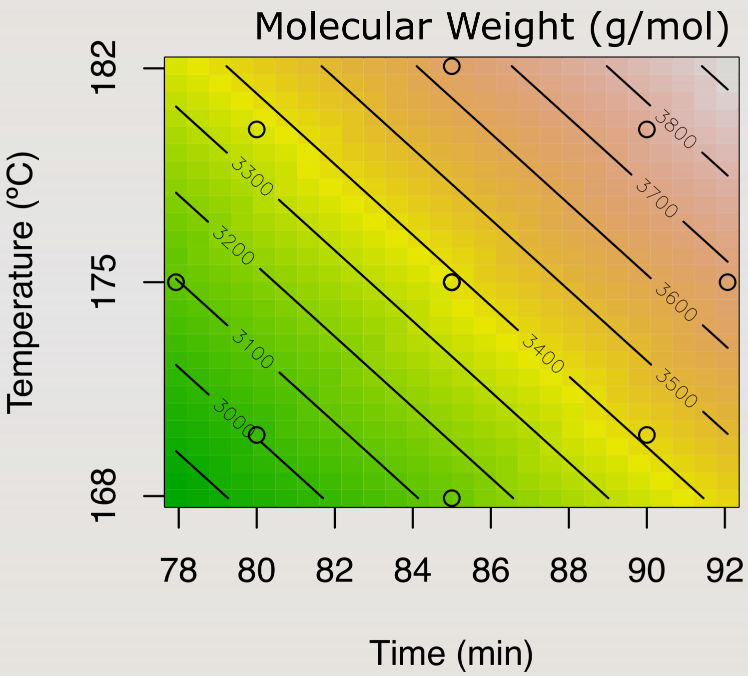
$$d = \begin{cases} 0 & y < L \text{ lower limit} \\ \left(\dfrac{y-L}{T-L}\right)^{r_1} & L \le y \le T \\ \left(\dfrac{U-y}{U-T}\right)^{r_2} & T \le y \le U \\ 0 & y > U \text{ upper limit} \end{cases}$$



Desirability Function for Multiple Responses





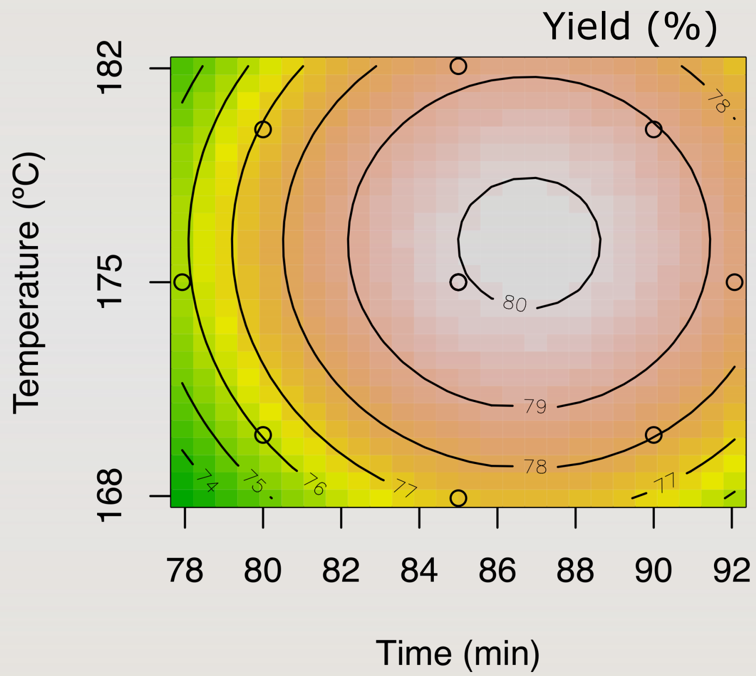


Yield higher than 78.5 %

Viscosity between 62 and 68 mPa s

Molecular weight lower than 3400 g/mol

Desirability Function for Multiple Responses



Maximise y:

$$d = \begin{cases} 0 & y < L \\ \left(\dfrac{y-L}{T-L}\right)^r & L \leq y \leq T \\ 1 & y > T \end{cases}$$

Maximise Yield:

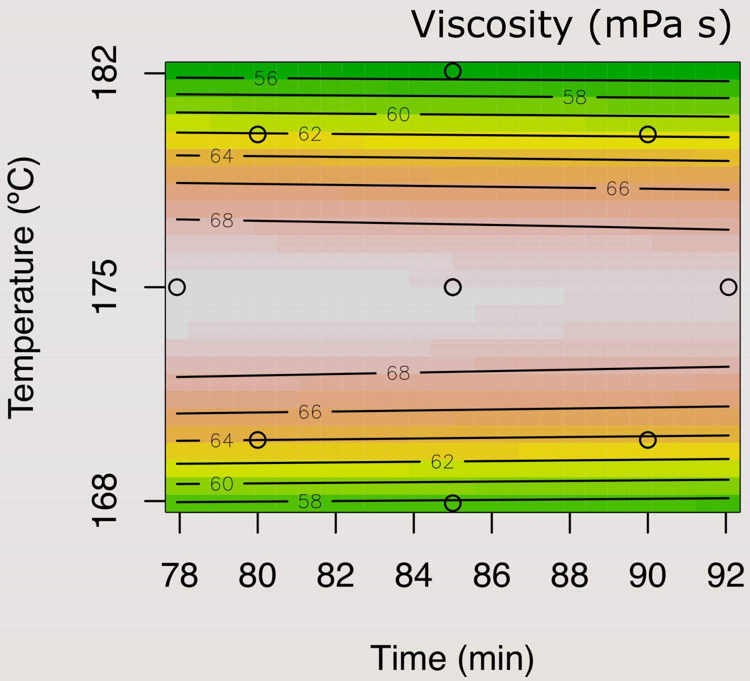
$L = 78.5$

$T = 81$

$r = 1$

Yield higher than 78.5 %

Desirability Function for Multiple Responses



Viscosity between 62 and 68 mPa s

Target y:

$$d = \begin{cases} 0 & y < L \\ \left(\dfrac{y - L}{T - L}\right)^{r_1} & L \le y \le T \\ \left(\dfrac{U - y}{U - T}\right)^{r_2} & T \le y \le U \\ 1 & y > U \end{cases}$$

Target Viscosity:

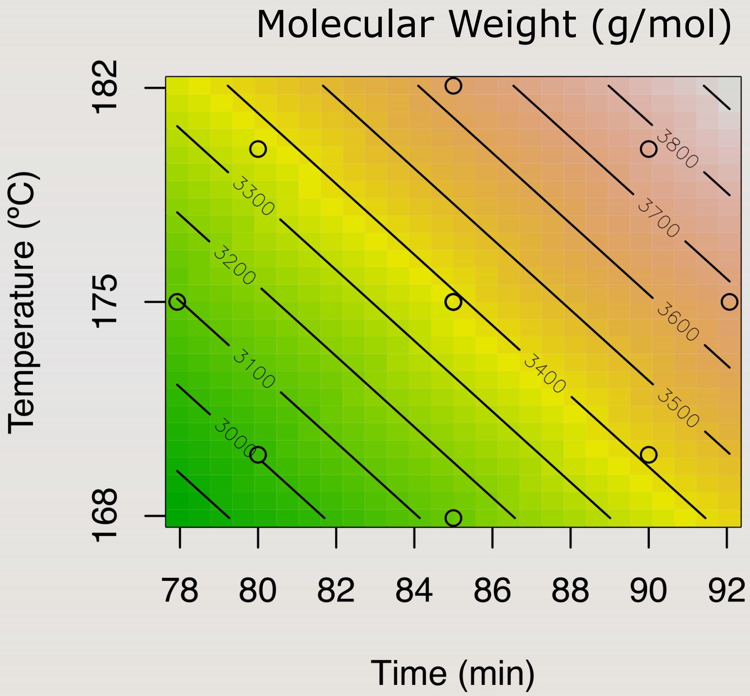
$L = 62$

$U = 68$

$T = 65$

$r_1 = r_2 = 1$

Desirability Function for Multiple Responses



Molecular weight lower than 3400 g/mol

Minimise y

$$d = \begin{cases} 1 & y < T \\ \left(\frac{U - y}{U - T}\right)^r & T \leq y \leq U \\ 0 & y > U \end{cases}$$

Minimise M_W:

$U = 3400$

$T = 3000$

$r = 1$

Desirability Functions for Multiple Responses

Desirability Function for Multiple Responses

Build and run the functions for desirability

```r
# Target
targetD <- function(y, L, T, U, r1, r2) {
  if (y < L){d <- 0}
  else if ( L<=y & y<=T ){
    d <- ((y - L) / (T - L))^r1}
  else if( T<=y & y<=U ){
    d <- ((U - y) / (U - T))^r2}
  else {
    d <- 0}
  return(d)}
```

```r
# Maximum
maxD <- function(y, L, T, r) {
  if (y < L){d <- 0}
  else if (y > T){d <- 1}
  else{d <- ((y - L) / (T - L))^r}
  return(d) }

# Minimum
minD <- function(y, U, T, r) {
  if (y < T){d <- 1}
  else if (y > U){d <- 0}
  else{d <- ((U - y) / (U - T))^r}
  return(d) }
```

Desirability Function for Multiple Responses

Build and run the functions for desirability

Create a grid with the coded or natural variables for the desirability determination

```
# create a grid for the desirability determination
dataD <- expand.grid(seq(-1.5, 1.5, by=0.05), seq(-1.5, 1.5, by=0.05))
colnames(dataD) <- c("x1", "x2")
```

coded variables

same denomination as the original data file

Desirability Function for Multiple Responses

Build and run the functions for desirability

Create a grid with the coded or natural variables for the desirability determination

Add the natural or coded variables and add the predicted responses

relationship between coded and natural variables

```
# add the natural variables

dataD$Time <- dataD$x1*5 + 85

dataD$Temp <- dataD$x2*5 + 175
```

predicted responses using the models built during the analysis

```
# add the predicted responses
dataD$Y <- predict(model_Y, newdata = dataD)
dataD$v <- predict(model_v, newdata = dataD)
dataD$Mw <- predict(model_Mw, newdata = dataD
```

Desirability Function for Multiple Responses

1. Build and run the functions for desirability
2. Create a grid with the coded or natural variables for the desirability determination
3. Add the natural or coded variables and add the predicted responses
4. Calculate individual and overall desirability for each value

Desirability Function for Multiple Responses

Build and run the

Create a grid
variables for

Add the
the pred

Calc

```
# For each data value, calculate desirability
for (i in 1:nrow(dataD)) {
  d1 <- maxD(dataD$Y[i]
                , L = 78.5, T = 81, r = 1)
  d2 <- targetD(dataD$v[i]
                , L = 62, T = 65, U = 68, r1 = 1, r2 = 1)
  d3 <- minD(dataD$Mw[i]
                , U = 3400, T = 3000, r = 1)
  D <- (d1 * d2 * d3)^(1/3)
  dataD[i, c("d1", "d2", "d3", "D")] <- c(d1, d2, d3, D)}
```

Desirability Function for Multiple Responses

Build and run the functions for desirability

Create a grid with the coded or natural variables for the desirability determination

Add the natural or coded variables and add the predicted responses

Calculate desirability for each value

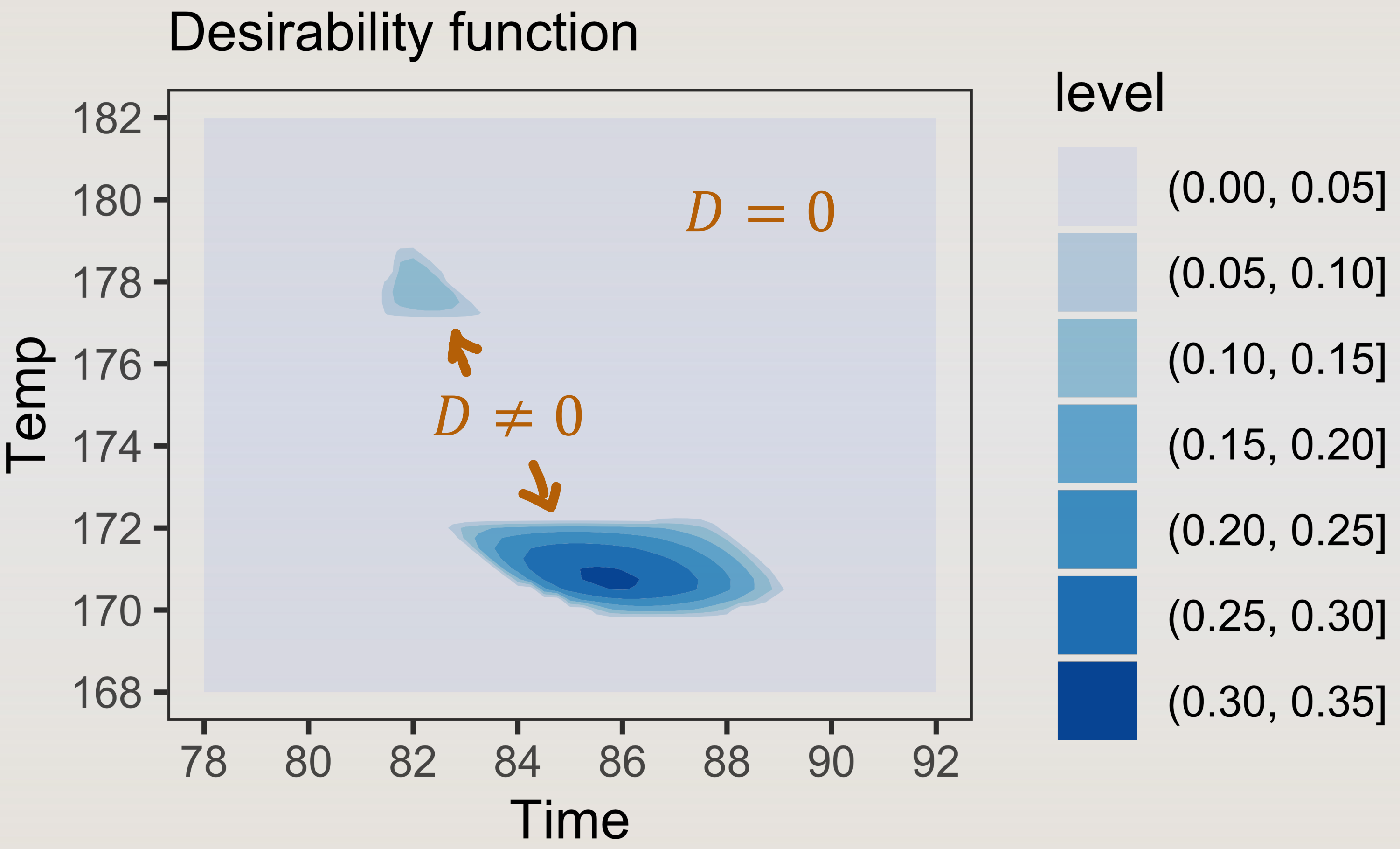
Plot the desirability function

Desirability Function for Multiple Responses

1. Build and run the functions for desirability
2. Create a grid with the coded or natural variables for the desirability determination
3. Add the natural or coded variables and add the predicted responses
4. Calculate desirability for each value
5. Plot the desirability function

```
# plot the desirability results
ggplot(data = dataD,
        aes(x = x1, y = x2, z = D)) +
  geom_contour_filled(alpha = 0.75) +
  scale_fill_brewer() +
  theme_bw()
```

Desirability Function for Multiple Responses



Desirability Function for Multiple Responses