First model Fractional Factorial, remove unneeded factors, those with a high probability factor.

Call:

lm(formula = yield ~ nifB + nifN + nifE + nifM + nifU + nifH +

nifD + nifS + nifK, data = transformData)

Residuals:

Min 1Q Median 3Q Max

-0.038764 -0.015173 0.003924 0.015259 0.043629

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.862016 0.103736 8.310 3.12e-08 \*\*\*

nifB 0.067303 0.009856 6.828 7.36e-07 \*\*\*

nifN 0.066313 0.011333 5.851 6.90e-06 \*\*\*

nifE 0.055111 0.011446 4.815 8.25e-05 \*\*\*

nifM 0.079585 0.011902 6.686 1.01e-06 \*\*\*

nifU 0.009574 0.009838 0.973 0.341

nifH -0.001689 0.009967 -0.169 0.867

nifD -0.012816 0.009348 -1.371 0.184

nifS 0.057756 0.010440 5.532 1.47e-05 \*\*\*

nifK 0.152725 0.008723 17.507 2.11e-14 \*\*\*

---

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

Residual standard error: 0.02653 on 22 degrees of freedom

Multiple R-squared: 0.9569, Adjusted R-squared: 0.9393

F-statistic: 54.3 on 9 and 22 DF, p-value: 5.934e-13

The factors we removed were nifU, nifH, and nifD.

all:

lm(formula = yield ~ nifB + nifN + nifE + nifM + nifS + nifK,

data = transformData)

Residuals:

Min 1Q Median 3Q Max

-0.049671 -0.016736 0.005668 0.015469 0.040969

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.846986 0.087068 9.728 5.58e-10 \*\*\*

nifB 0.067303 0.009828 6.848 3.54e-07 \*\*\*

nifN 0.066313 0.011301 5.868 4.03e-06 \*\*\*

nifE 0.055111 0.011412 4.829 5.80e-05 \*\*\*

nifM 0.079585 0.011868 6.706 5.00e-07 \*\*\*

nifS 0.057756 0.010409 5.548 9.08e-06 \*\*\*

nifK 0.152725 0.008698 17.558 1.42e-15 \*\*\*

---

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

Residual standard error: 0.02645 on 25 degrees of freedom

Multiple R-squared: 0.9513, Adjusted R-squared: 0.9397

F-statistic: 81.45 on 6 and 25 DF, p-value: 3.475e-15

> newData <- data.frame(nifB = log10(723), nifN = log10(941), nifE = log10(946), nifM = log10(839), nifS = log10(838), nifK = log10(396))

> predictedYield <- 10^predict(fit, newData)

Predicted yield = 158

newData <- data.frame(nifB = log10(6468), nifN = log10(6327), nifE = log10(6243), nifM = log10(5150), nifS = log10(6633), nifK = log10(396))

predictedYield <- 10^predict(fit, newData)

predicted yield = 300;

# Uses regression model to predict yield for a new set of gene expression levels

I changed these values by adding the percentage in the estimate column to the values and then recalculating a yield.

newData <- data.frame(nifB = log10(6903), nifN = log10(6746), nifE = log10(6587), nifM = log10(5560), nifS = log10(7016), nifK = log10(456))

predictedYield <- 10^predict(fit, newData)

the predicted yield is 313

# Uses regression model to predict yield for a new set of gene expression levels

newData <- data.frame(nifB = log10(7140), nifN = log10(6917), nifE = log10(6662), nifM = log10(5950), nifS = log10(16500), nifK = log10(1428))

predictedYield <- 10^predict(fit, newData)

The predicted yield is:

For the second model: We set thresholds on the values left in our model. Then got the yields from these thresholds. The summary of the fit is this.

Call:

lm(formula = yield ~ nifB + nifN + nifE + nifM + nifS + nifK,

data = transformData)

Residuals:

Min 1Q Median 3Q Max

-0.030507 -0.010743 0.001472 0.010300 0.029521

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 1.697024 0.698652 2.429 0.0227 \*

nifB 0.002901 0.038258 0.076 0.9402

nifN 0.050842 0.060560 0.840 0.4091

nifE -0.008544 0.153247 -0.056 0.9560

nifM -0.057750 0.070771 -0.816 0.4222

nifS -0.001932 0.014472 -0.133 0.8949

nifK 0.314426 0.012298 25.568 <2e-16 \*\*\*

---

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

Residual standard error: 0.01631 on 25 degrees of freedom

Multiple R-squared: 0.9632, Adjusted R-squared: 0.9544

F-statistic: 109.2 on 6 and 25 DF, p-value: < 2.2e-16

From this we know that the only factor left that will make a change to our model is nifK.

So we set an upper threshold for nifK of 5000 and submitted our third model using full factorial.

newData <- data.frame(nifB = log10(7140), nifN = log10(6917), nifE = log10(6662), nifM = log10(5950), nifS = log10(16500), nifK = log10(5000))

predictedYield <- 10^predict(fit, newData)

predictedYield = 642

The only factor left to optimize is nifK.

Model with the following thresholds:

nifB: 4500 – 8000

nifE: 5000 – 7000

nifK: 3000-5000

nifM: 4000 – 7000

nifN: 4000 – 7000

nifS: 6000-16000

Call:

lm(formula = yield ~ nifB + nifN + nifE + nifM + nifS + nifK,

data = transformData)

Residuals:

Min 1Q Median 3Q Max

-0.0262904 -0.0074146 -0.0008126 0.0079143 0.0196133

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.297486 0.216290 10.622 3.93e-15 \*\*\*

nifB 0.025238 0.015556 1.622 0.1103

nifN 0.038613 0.022639 1.706 0.0935 .

nifE 0.008970 0.038142 0.235 0.8149

nifM 0.017500 0.029030 0.603 0.5490

nifS 0.007619 0.007380 1.032 0.3063

nifK -0.003219 0.014421 -0.223 0.8241

---

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

Residual standard error: 0.01132 on 57 degrees of freedom

Multiple R-squared: 0.1104, Adjusted R-squared: 0.01678

F-statistic: 1.179 on 6 and 57 DF, p-value: 0.3303

Average yield approximately: 450

Fractional Factorial Model, max synthesis:

Expression thresholds:

nifB: 6000-8000

nifE: 5500 – 7500

nifK: 2000- 4000

nifM: 5000-8000

nifN: 5000 – 8000

nifS: 8000 – 12000

I made the model above because I realized that the R-squared value of my former model was rather low, which indicates to me that what I thought to be my final model may not be a good one. Idk. Anyways its always best to have another model.

Important things to look at in the fit summaries:

The probability values, anything less than 0.05 is negligible. The estimates, a positive estimate means you can increase that values. A negative estimate means you can decrease that value. And the R-squared values, which tells you how well your statistical model is fitting the data.