Homework 8 - AGST 5014

Igor Kuivjogi Fernandes and Ashmita Upadhyay

2023-04-24

1. The following experiment comes from a central composite design with 4 factors in 2 blocks. Conduct the proper analysis (including graphs, interpretation, etc).

```
q1 <- read.csv("HW8_Q1.csv")
q1 <- transform(q1, block = factor(block), logSD = NULL)
str(q1)
```

```
## 'data.frame':    30 obs. of  6 variables:
##  $ block: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ x1   : int  -1 1 -1 1 -1 1 -1 1 -1 1 ...
##  $ x2   : int  -1 -1 1 1 -1 -1 1 1 -1 -1 ...
##  $ x3   : int  -1 -1 -1 -1 1 1 1 1 -1 -1 ...
##  $ x4   : int  -1 -1 -1 -1 -1 -1 -1 -1 1 1 ...
##  $ ave  : int  367 369 374 370 372 355 397 377 350 373 ...
```

```
table(q1$block)
```

```
##
##  1  2
## 18 12
```

We have 18 experimental units for the factorial part, and 12 for the axial part.

Let's fit some Response Surface Methodology models.

First, we can start with a simple first order model:

```
library(rsm)

mod1 <- rsm(ave ~ block + FO(x1, x2, x3, x4), data = q1)
summary(mod1)
```

```
##
## Call:
## rsm(formula = ave ~ block + FO(x1, x2, x3, x4), data = q1)
##
##               Estimate Std. Error  t value  Pr(>|t|)
## (Intercept) 367.111111   1.883588 194.8999 < 2.2e-16 ***
## block2       -1.527778   2.978214  -0.5130  0.612652
## x1           -0.083333   1.631235  -0.0511  0.959680
```

```
## x2            5.083333   1.631235   3.1162  0.004701 **
## x3            0.250000   1.631235   0.1533  0.879476
## x4           -6.083333   1.631235  -3.7293  0.001041 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.499,  Adjusted R-squared:  0.3947
## F-statistic: 4.782 on 5 and 24 DF,  p-value: 0.003577
##
## Analysis of Variance Table
##
## Response: ave
##                     Df  Sum Sq Mean Sq F value   Pr(>F)
## block                1   16.81   16.81  0.2632 0.612652
## FO(x1, x2, x3, x4)   4 1510.00  377.50  5.9112 0.001856
## Residuals           24 1532.69   63.86
## Lack of fit         20 1521.94   76.10 28.3152 0.002594
## Pure error           4   10.75    2.69
##
## Direction of steepest ascent (at radius 1):
##          x1          x2          x3          x4
## -0.01050596  0.64086379  0.03151789 -0.76693536
##
## Corresponding increment in original units:
##          x1          x2          x3          x4
## -0.01050596  0.64086379  0.03151789 -0.76693536
```

The lack of fit is significant (p-value < 0.05), so we should include more complex terms.

Now a second-order model:

```r
mod2 <- rsm(ave ~ block + SO(x1, x2, x3, x4), data = q1)
summary(mod2)
```

```
##
## Call:
## rsm(formula = ave ~ block + SO(x1, x2, x3, x4), data = q1)
##
##               Estimate Std. Error  t value  Pr(>|t|)
## (Intercept) 372.800000   1.506375 247.4815 < 2.2e-16 ***
## block2       -2.950000   1.207787  -2.4425 0.0284522 *
## x1           -0.083333   0.636560  -0.1309 0.8977075
## x2            5.083333   0.636560   7.9856 1.398e-06 ***
## x3            0.250000   0.636560   0.3927 0.7004292
## x4           -6.083333   0.636560  -9.5566 1.633e-07 ***
## x1:x2        -2.875000   0.779623  -3.6877 0.0024360 **
## x1:x3        -3.750000   0.779623  -4.8100 0.0002773 ***
## x1:x4         4.375000   0.779623   5.6117 6.412e-05 ***
## x2:x3         4.625000   0.779623   5.9324 3.657e-05 ***
## x2:x4        -1.500000   0.779623  -1.9240 0.0749257 .
## x3:x4        -2.125000   0.779623  -2.7257 0.0164099 *
## x1^2         -2.037500   0.603894  -3.3739 0.0045424 **
## x2^2         -1.662500   0.603894  -2.7530 0.0155541 *
## x3^2         -2.537500   0.603894  -4.2019 0.0008873 ***
```

```
## x4^2          -0.162500   0.603894  -0.2691 0.7917877
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.9555, Adjusted R-squared:  0.9078
## F-statistic: 20.04 on 15 and 14 DF,  p-value: 6.54e-07
##
## Analysis of Variance Table
##
## Response: ave
##                     Df  Sum Sq Mean Sq F value    Pr(>F)
## block                1   16.81   16.81  1.7281  0.209786
## FO(x1, x2, x3, x4)   4 1510.00  377.50 38.8175 1.965e-07
## TWI(x1, x2, x3, x4)  6 1114.00  185.67 19.0917 5.355e-06
## PQ(x1, x2, x3, x4)   4  282.54   70.64  7.2634  0.002201
## Residuals           14  136.15    9.72
## Lack of fit         10  125.40   12.54  4.6660  0.075500
## Pure error           4   10.75    2.69
##
## Stationary point of response surface:
##         x1         x2         x3         x4
##  0.8607107 -0.3307115 -0.8394866 -0.1161465
##
## Eigenanalysis:
## eigen() decomposition
## $values
## [1]  3.258222 -1.198324 -3.807935 -4.651963
##
## $vectors
##          [,1]       [,2]       [,3]        [,4]
## x1  0.5177048 0.04099358  0.7608371 -0.38913772
## x2 -0.4504231 0.58176202  0.5056034  0.45059647
## x3 -0.4517232 0.37582195 -0.1219894 -0.79988915
## x4  0.5701289 0.72015994 -0.3880860  0.07557783
```

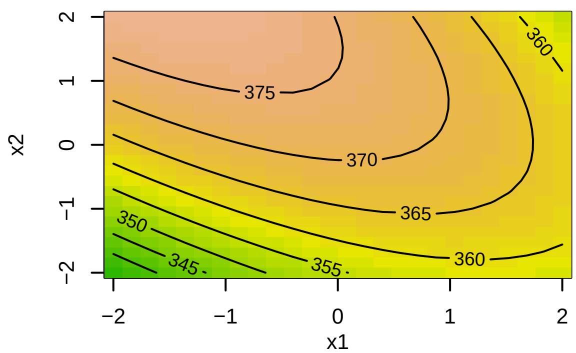
There are positive and negatives eigenvalues, which means we have a saddle point.
The adj.$R^2 = 0.9078$, and the lack of fit is not significant, so we can stick with this model.

The optimal experimental points are:

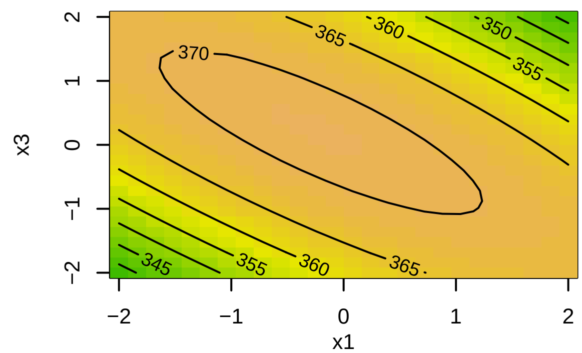
```
summary(mod2)$canonical$xs
```

```
##         x1         x2         x3         x4
##  0.8607107 -0.3307115 -0.8394866 -0.1161465
```

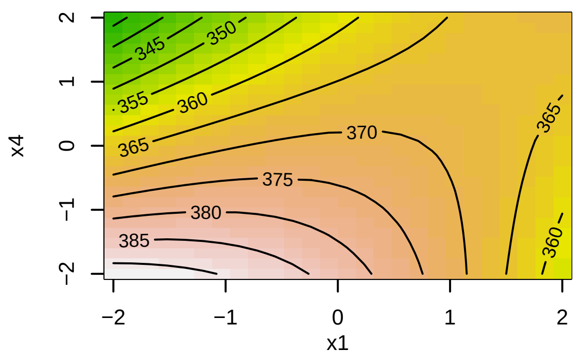
```
par(mfrow = c(3, 2))
contour(
  mod2,
  ~ x1 + x2 + x3 + x4,
  image = TRUE
)
```



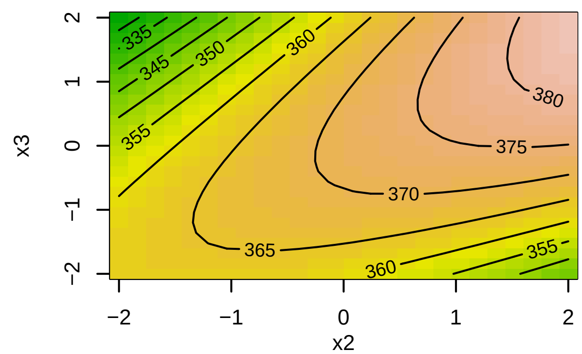
Slice at x3 = 0, x4 = 0



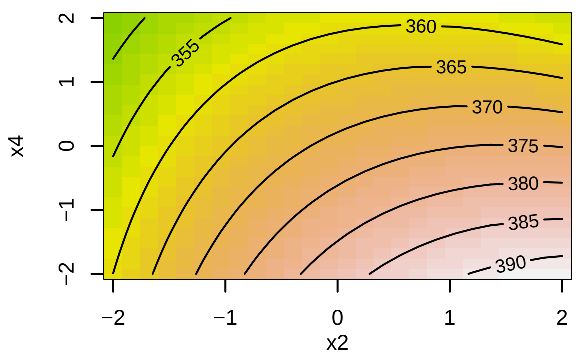
Slice at x2 = 0, x4 = 0



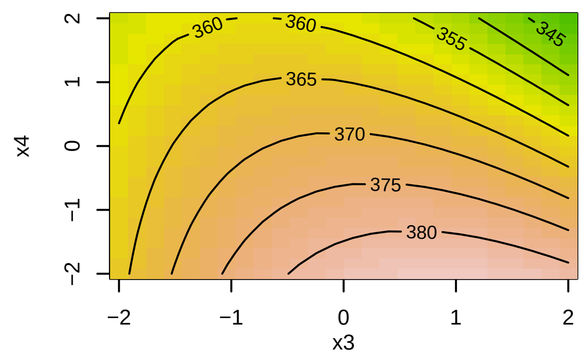
Slice at x2 = 0, x3 = 0



Slice at x1 = 0, x4 = 0



Slice at x1 = 0, x3 = 0



Slice at x1 = 0, x2 = 0

We have 6 plots because there are 4 two-way interactions (`x1:x2`, `x1:x3`, `x1:x4`, `x2:x3`, `x2:x4`, and `x3:x4`). From the plots we can see that the maximum point reached lies roughly in the interval 380-390.
We can check the distribution of the fitted values:

```
summary(mod2$fitted.values)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   346.8   359.4   368.7   366.5   372.2   397.1
```

The lowest point was 346.8, whereas the maximum point was 397.1.

Now we can run a steepest-ascent algorithm to search for a better solution:

```r
steep <- steepest(mod2)
```

```
## Path of steepest ascent from ridge analysis:
```

```r
steep
```

```
##    dist     x1    x2    x3     x4 |    yhat
## 1   0.0  0.000 0.000 0.000  0.000 | 372.800
## 2   0.5 -0.127 0.288 0.116 -0.371 | 377.106
## 3   1.0 -0.351 0.538 0.312 -0.700 | 382.675
## 4   1.5 -0.595 0.775 0.526 -1.009 | 389.783
## 5   2.0 -0.846 1.007 0.745 -1.309 | 398.485
## 6   2.5 -1.101 1.237 0.966 -1.605 | 408.819
## 7   3.0 -1.356 1.465 1.189 -1.897 | 420.740
## 8   3.5 -1.613 1.693 1.413 -2.188 | 434.322
## 9   4.0 -1.870 1.920 1.637 -2.477 | 449.497
## 10  4.5 -2.127 2.147 1.862 -2.766 | 466.323
## 11  5.0 -2.385 2.373 2.086 -3.054 | 484.750
```

The optimal solution points now are:

```r
opt_points <- steep[which.max(steep$yhat), ]
opt_points
```

```
##    dist     x1    x2    x3     x4 |   yhat
## 11    5 -2.385 2.373 2.086 -3.054 | 484.75
```

These are the new points we would use in the process to get the higher response values.
If we do predictions using these new coefficients, we get the maximum predicted response found by the steepest-ascent algorithm above:

```r
grid <- expand.grid(
  block = unique(q1$block),
  x1 = opt_points$x1,
  x2 = opt_points$x2,
  x3 = opt_points$x3,
  x4 = opt_points$x4
)
predict(mod2, grid)
```

```
##        1        2 
## 484.7497 481.7997
```

As we have 2 blocks, the model did two predictions. The first matches with the steepest-ascent algorithm.

2. The design below presents the yield of different common bean cultivars. There was a variable stand count in each plot. Conduct the proper analysis.

```
q2 <- read.csv("HW8_Q2.csv")
q2 <- transform(q2, block = factor(block), cv = factor(cv))
str(q2)
```

```
## 'data.frame':    28 obs. of  4 variables:
##  $ block: Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 2 2 2 ...
##  $ cv   : Factor w/ 7 levels "CNFP8018","CNFP8019",..: 1 2 3 4 5 6 7 1 2 3 ...
##  $ stand: int  95 124 106 92 101 98 103 103 114 116 ...
##  $ yield: num  1588 2012 1975 1838 1825 ...
```

```
table(q2$block)
```

```
##
## 1 2 3 4
## 7 7 7 7
```

The blocks are equally frequent.

The `stand` variable is a covariable, hence we can use ANCOVA to analyse the yield of different cultivars.

First step is to check whether `stand` is independent of the treatment `cv`.

```
check <- aov(stand ~ block + cv, data = q2)
summary(check)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## block        3 1066.0   355.3   4.082 0.0225 *
## cv           6  790.7   131.8   1.514 0.2298
## Residuals   18 1567.0    87.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The `cv` is not significant, so we expect `stand` and `cv` to not be related.

Next, we run ANCOVA with interaction. For ANCOVA, we should use Type III sum of squares.

```
check_inter <- lm(
  yield ~ block + cv * stand,
  contrasts = list(cv = contr.sum),
  data = q2
)
car::Anova(check_inter, type = 'III')
```

```
## Anova Table (Type III tests)
##
## Response: yield
##             Sum Sq Df F value Pr(>F)
## (Intercept)  11688  1  0.2633 0.6180
```

```
## block         75844  3  0.5694 0.6466
## cv           488399  6  1.8334 0.1816
## stand        140453  1  3.1636 0.1029
## cv:stand     441062  6  1.6557 0.2217
## Residuals    488368 11
```

The interaction `cv:stand` is not significant, so we can go further and fit ANCOVA without the interaction:

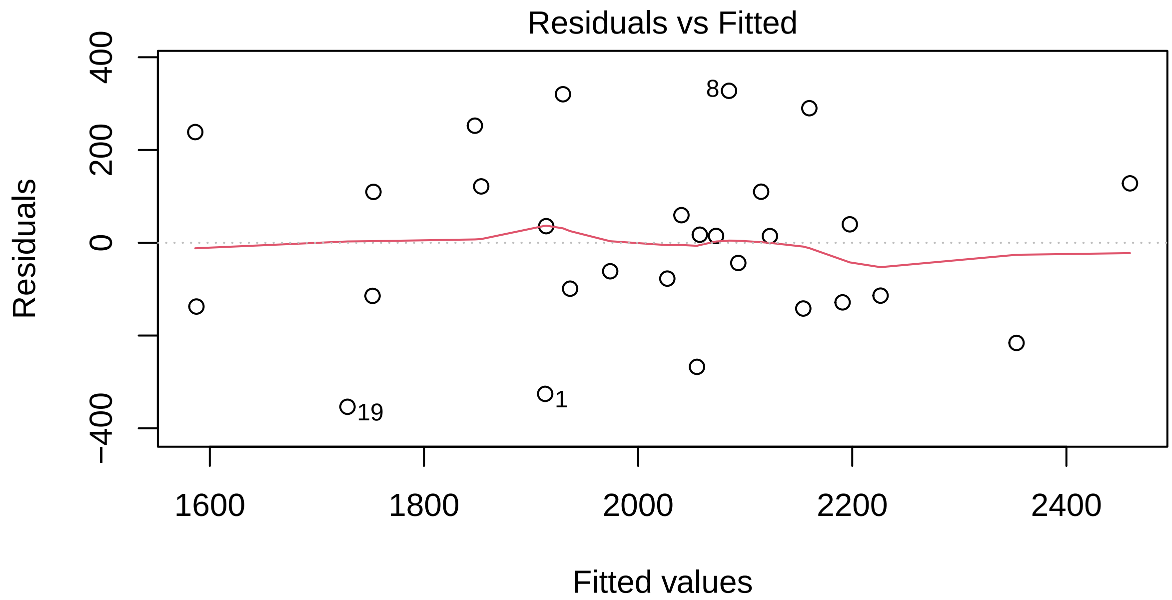
```
ancova <- lm(
  yield ~ block + cv + stand,
  contrasts = list(cv = contr.sum),
  data = q2
)
car::Anova(ancova, type = 'III')
```

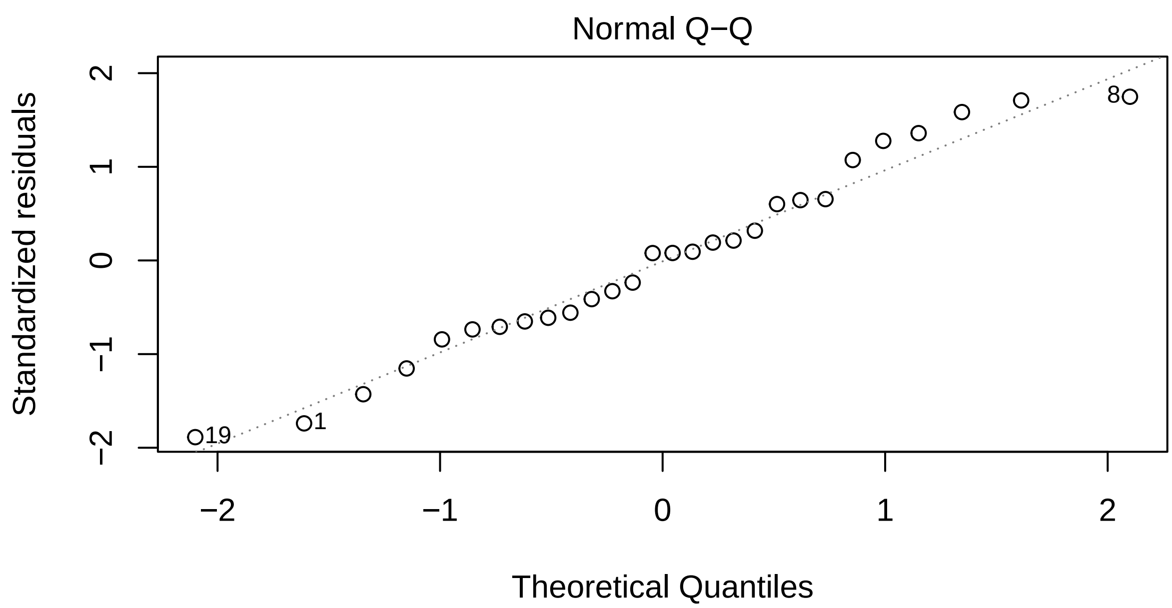
```
## Anova Table (Type III tests)
##
## Response: yield
##             Sum Sq Df F value  Pr(>F)
## (Intercept)  24622  1  0.4504 0.51118
## block       116917  3  0.7128 0.55769
## cv          899421  6  2.7419 0.04741 *
## stand       348326  1  6.3712 0.02184 *
## Residuals   929430 17
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The `cv` is indeed significant, using a significance level of $\alpha = 0.05$.

Now let's check the usual ANOVA assumptions:

```
par(mfrow= c(2, 1))
plot(ancova, which = 1)
plot(ancova, which = 2)
```



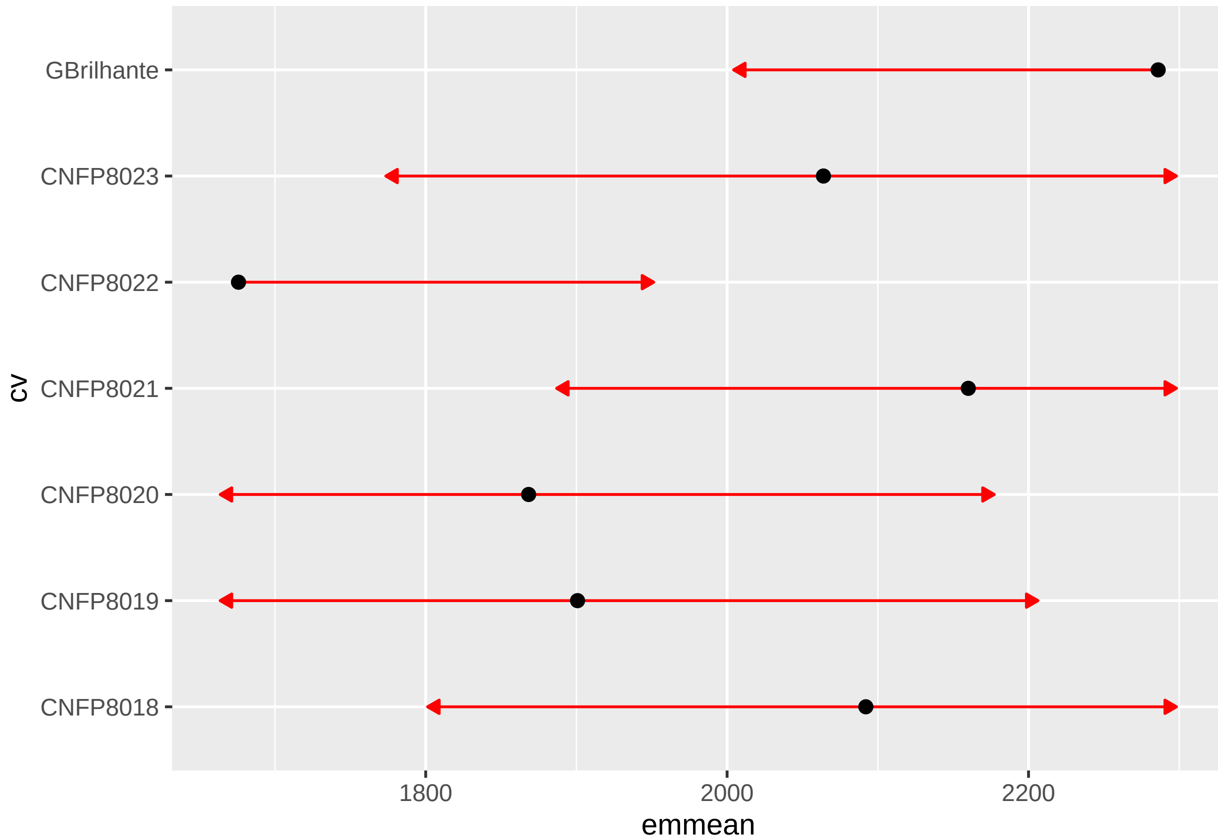


We have homogeneous variance across the fitted values and the residuals seems to be normally distributed.

Which cultivar was the best?

```
plot(
  emmeans::emmeans(ancova, pairwise ~ cv, adjust = 'tukey'),
  interval = F, comparisons = T
```

```
)
```



We can see that cultivar `GBrilhante` had better performance than `CNFP8022`, but `GBrilhante` is not different from others.

3. Design a proper experiment to identify the best dose of Nitrogen and amount of water to maximize yield (choose what values you would use).

Let's suppose there are 3 different doses of nitrogen and 2 different levels of water.
We can design a factorial CRD:

```
q3 <- expand.grid(
  nitrogen = factor(c('10', '50', '150')),
  water = factor(c('0', '20'))
)
q3
```

```
##   nitrogen water
## 1       10     0
## 2       50     0
## 3      150     0
## 4       10    20
## 5       50    20
## 6      150    20
```