Assignment 2

Group 9. Rinus van Grunsven (10755373), Florens Douwes (11254483), Imad Lotfi (14651610)

TBD 2022

Exercise 2.1

a) The data is read in with the following command:

```
data_sat = read.table("./sat.txt", header=TRUE)
expend = data_sat$expend
ratio = data_sat$ratio
salary = data_sat$salary
takers = data_sat$takers
total = data_sat$total
```

The first technique is the step-up method. This method examines the statistics for each variable and selects the option with the highest r-squared value, given the p-value is significant. This procedure is repeated until the added variable's p-value is no longer statistically significant. The outcomes are displayed below.

The variable 'takers' will be chosen in the first round and the variable 'expend' in the second round. We stop after the second round because the p-values for both the ratio and the salary are not statistically significant. Consequently, the step-up method model will be lm(total~takers+expend). The step-down method does not add variables, but instead begins with a model that contains all variables and then removes options. These are selected using the highest non-significant p-value. The calculation is displayed below.

The variable 'expend' is removed in the first round because it has the highest non-significant p-value.There is no further removal of variables as all variables have significant p-values. The statistics for the step-up method are as follows: multiple r-squared = 0.8195 and p-value = < 2.2e-16. For the step-down method it is: multiple-r^2 = 0.8239 and p-value = < 2.2e-16 Given these numbers, the model with the greatest r-squared multiple would be preferred. However, as the difference is very small, we also consider the model's complexity. The step-up method utilizes only two variables, whereas the step-down method has three. Therefore, the former is easier. In this situation, we would prefer the step-up method due to the fact that their performance is nearly identical.

b) The variable takers2 is the square of the takers values. This is followed by the step-up method.

```
takers2=(data_sat$takers)^2
```

In the first round, the variable 'takers' is selected, followed by the variable 'takers2' and finally the variable 'expend'. We stop after the third round because the variables added in the fourth round do not have a significant p-value.

The step-down method is shown below:

The variable takers2 is included in both the step-up and step-down models, making its addition useful.Both have got three variables so they are equally complex. Since both models are identical, it makes no difference which method is chosen, as both will result in the same model.

c) The model in exercise a has an r-squared multiple of 0.8194 and a p-value less than 2.2e-16. Multiple r-squared value of 0.8859 and a p-value less than 2.2e-16 are the statistical results for the model selected in exercise b. However, the first model uses only two variables, while the second model utilises three. We prefer the multiple r-squared value over the complexity of the model and thus select the second model.

d)

```
summary(lm(total~takers+takers2+expend))
chosen_model = lm(total~takers+takers2+expend)
# model = 1052 - 6.381 * takers + 0.04741 * takers2 + 7.914 * expend
newxdata = data.frame(expend=5, takers2=625, salary=36000, takers=25)
predict(chosen_model,newxdata,interval="prediction",level=0.95)
# Fit = 961.5703; lwr = 907.6003; upr = 1015.54;
```

The fit is 961.57, the lowerbound 907.6 and the upperbound 1015.54.

Exercise 2.2

```
df = read.table("./treeVolume.txt", header=TRUE)
```

a)

We assume that there is no hidden relationship between measurements. We also need to know if the data is normalized.

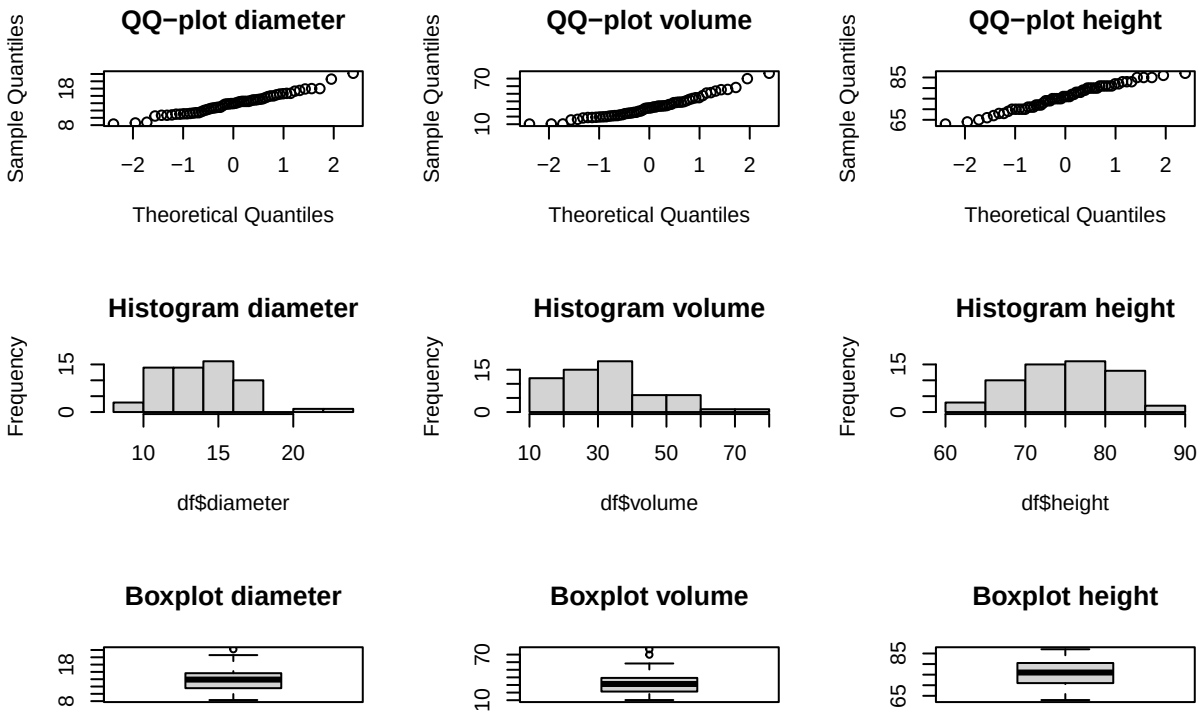
```
shapiro.test(df$diameter);shapiro.test(df$volume);shapiro.test(df$height);
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$diameter
## W = 0.97858, p-value = 0.3825
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$volume
## W = 0.94646, p-value = 0.01156
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$height
## W = 0.97876, p-value = 0.3895
```

```
par(mfrow=c(3,3));
qqnorm(df$diameter, main="QQ-plot diameter");qqnorm(df$volume, main="QQ-plot volume");qqnorm(df
hist(df$diameter, main="Histogram diameter");hist(df$volume, main="Histogram volume");hist(df$
boxplot(df$diameter, main="Boxplot diameter");boxplot(df$volume, main="Boxplot volume");boxplo
```



The data follows a fairly normal distribution, as far as we can know with the limited dataset.

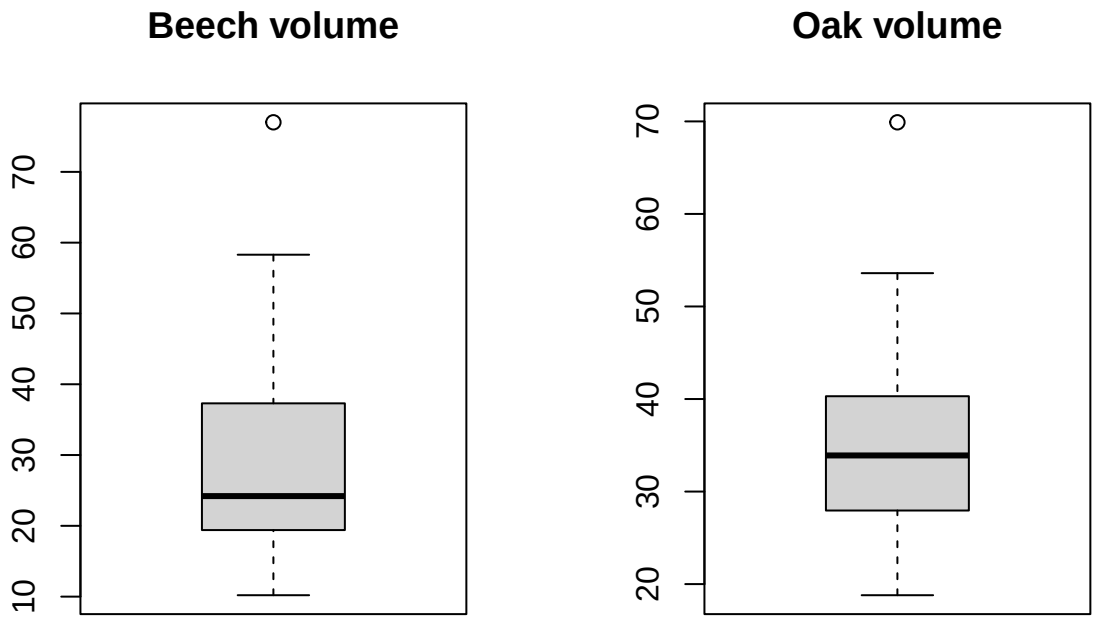
Next, we will model the differences in the mean of the volume, as a function of the type of tree:

```
# Does the type affect the volume?
fit = aov(volume ~ type, data=df)
summary(fit)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## type         1    380   379.5   1.898  0.174
## Residuals   57  11395   199.9
```

The F value is not high, which means there is little variation between groups. Furthermore, the p value is not below 0.05. Therefore it seems (in this test) that the tree type does not have an significant effect on the volume.

```
par(mfrow=c(1,2));
boxplot(df[df$type == "beech",]$volume, main="Beech volume");
boxplot(df[df$type == "oak",]$volume, main="Oak volume")
```



However, graphing the data shows that an oak does in fact seem to have a higher mean volume compared to a beech, just not significant enough. The variance looks about the same.

b) Now, we'll include diameter and height into the analysis, performing an ANCOVA test:

```
lm2 = lm(volume ~ diameter + height + type, data=df)
fit2 = anova(lm2)
fit2
```

```
## Analysis of Variance Table
##
## Response: volume
##           Df  Sum Sq Mean Sq   F value    Pr(>F)
## diameter   1 10826.5 10826.5 1029.5139 < 2.2e-16 ***
## height     1   346.2   346.2   32.9192 4.254e-07 ***
```

```
## type        1    23.2    23.2    2.2083     0.143
## Residuals 55   578.4    10.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With this included, we see that the type is still not a big influence. The p-value is 0.143, indicating that the difference between the means is not significant.

```
lm2
```

```
##
## Call:
## lm(formula = volume ~ diameter + height + type, data = df)
##
## Coefficients:
## (Intercept)     diameter       height      typeoak
##    -63.7814       4.6981       0.4172      -1.3046
```

This shows that the oak type negatively influences the volume (-1.3046).

Estimated volumes for the two tree types:

```
predict(lm2, newdata=data.frame(height=mean(df$height), diameter=mean(df$diameter), type="beech
```

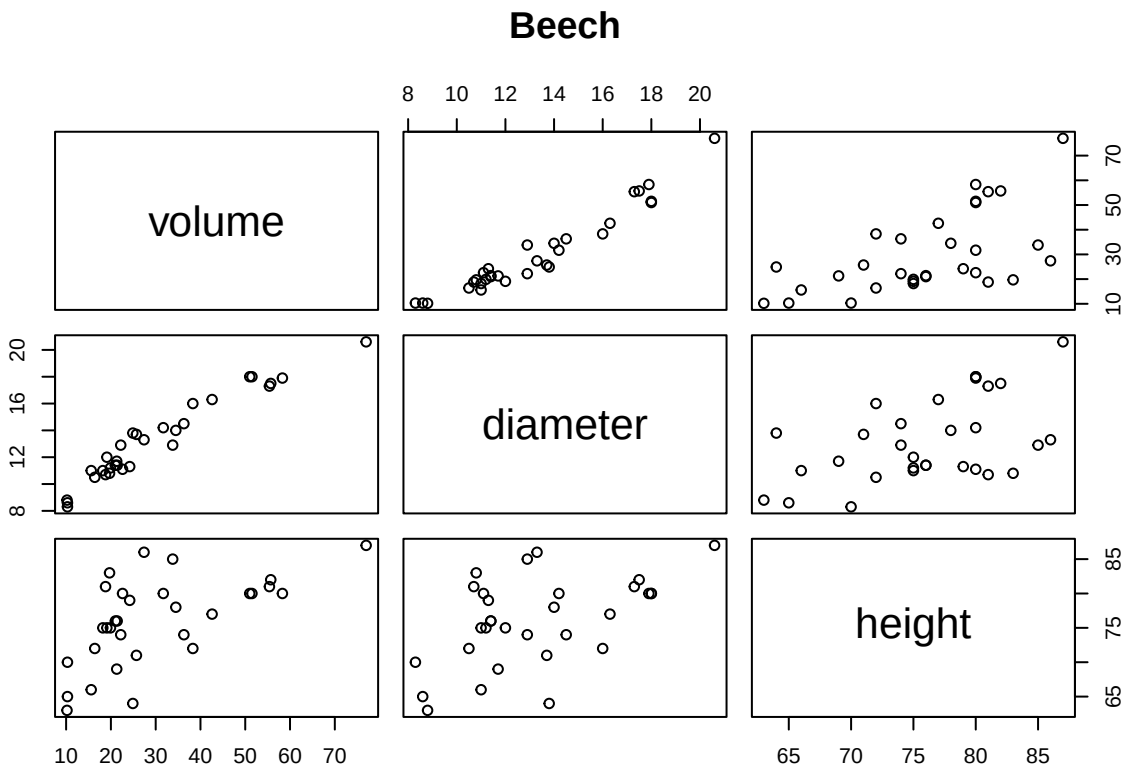
```
##        1
## 33.20049
```

```
predict(lm2, newdata=data.frame(height=mean(df$height), diameter=mean(df$diameter), type="oak")
```

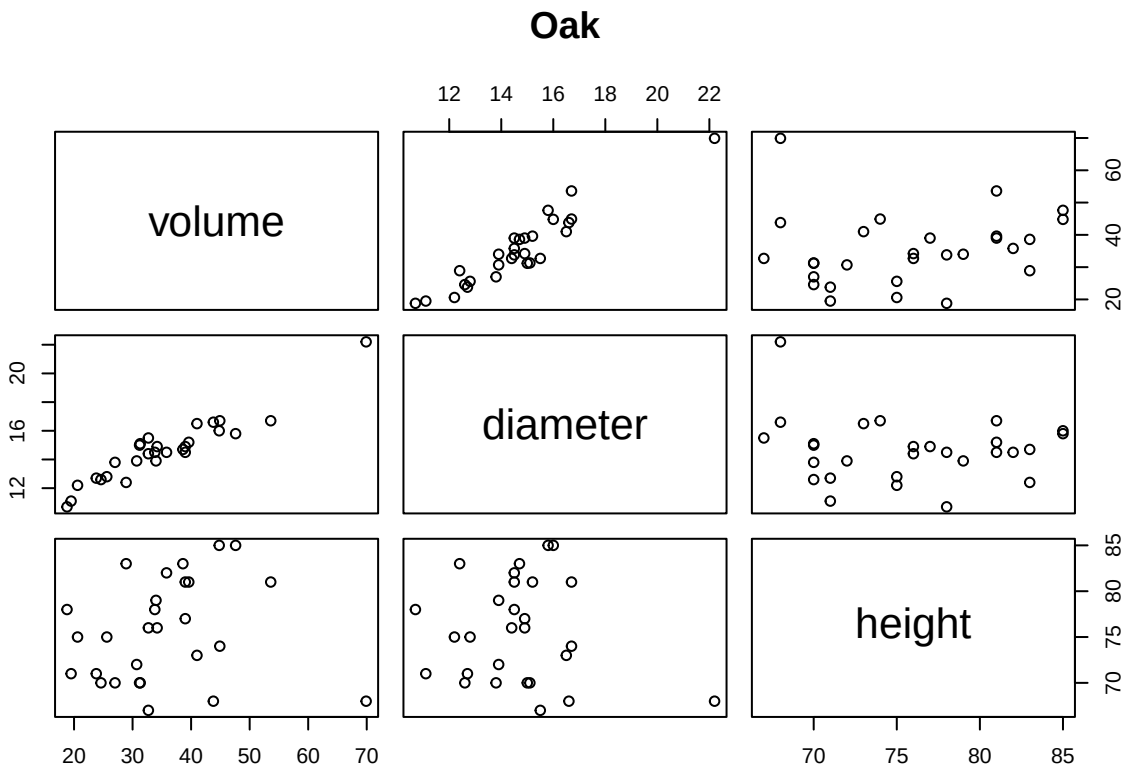
```
##        1
## 31.89589
```

A graphical investigation how the diameter influences the volume:

```
par(mfrow=c(1,2))
pairs(split(df, df$type)$beech[c('volume', 'diameter', 'height')], main="Beech")
```




```
pairs(split(df, df$type)$oak[c('volume', 'diameter', 'height')], main="Oak")
```

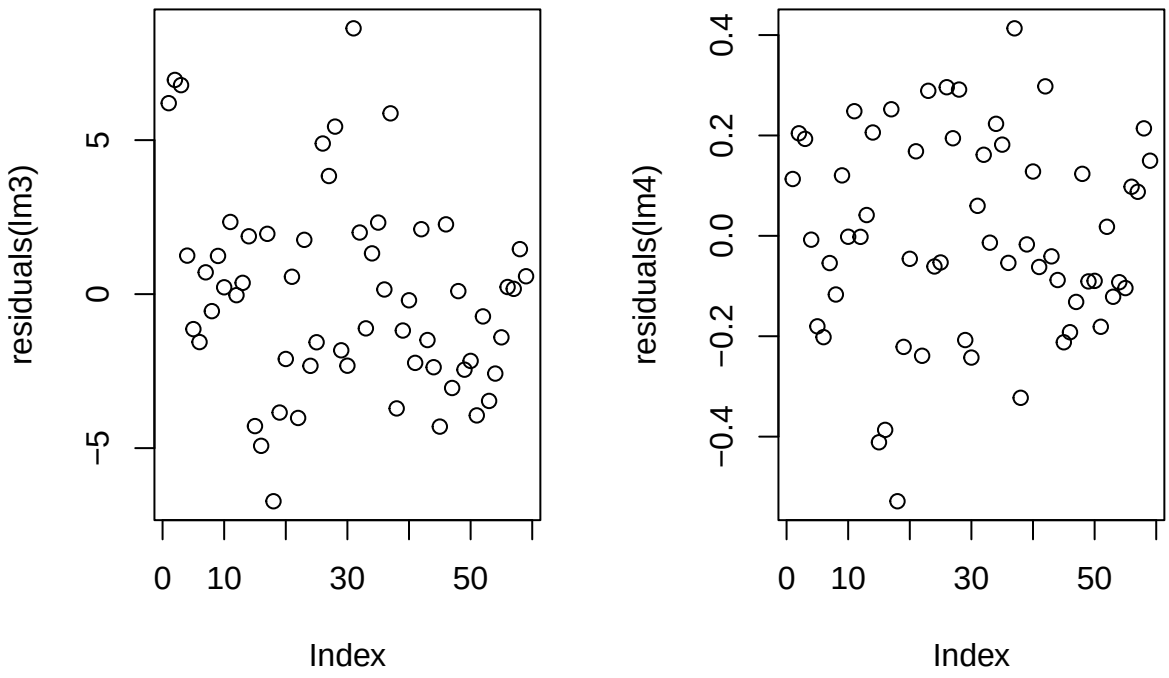


For both tree types, the volume and diameter show a clear relation (straight line). The relation also seems present in both tree types.

c) We can square the volume, as the diameter and height can be seen similar to a multiplication that results in the volume (diameter * height = volume).

```
lm3 = lm(volume ~ diameter + height, data=df)
lm4 = lm(sqrt(volume) ~ diameter + height, data=df)

par(mfrow=c(1,2))
plot(residuals(lm3))
plot(residuals(lm4))
```



By doing so we can see the residuals going down a lot, which would improve the fit.

Exercise 2.3

a) The calculation made in cell C23 is '=SUMPRODUCT(C19:G19;C20:G20)'. For cell I7 it is '=SUMPRODUCT(C7:G7;C19:G19)', cell I8 '=SUMPRODUCT(C8:G8;C19:G19)', cell I9 '=SUMPRODUCT(C9:G9;C19:G19)' and cell I10 '=SUMPRODUCT(C10:G10;C19:G19)'. The image depicts the calculations performed in Excel solver.
The products and nutrients are swapped because it was simpler to construct the Excel solver in this manner.The solution is 0 raw carrots, 7,71 baked potatoes, 0 wheat bread, 0 cheddar cheese, and 9,28 peanut butter.

b) The same calculations in cells I7, I8, I9, and I10 are made as was done in exercise a), but cell C23 is now '=SUMPRODUCT(C19:H19;C20:H20)'. The calculation has been modified slightly because the price of peanut butter increases after the fifth unit. Therefore, a new column ('Peanut butter expensive') containing the same nutrients but at a higher price is added. Additionally, a new constraint has been added so that the quantity of inexpensive peanut butter units cannot exceed five units.The solution is 0 raw carrots, 16,62 baked potatoes, 0 wheat bread, 0 cheddar cheese, 5 cheap peanut butter, and 0 expensive peanut butter.

c) The data used in this exercise is the same as the one in exercise a). The calculations made in this exercise are also the same, but with the additional constraint that the decision variables must be integers.The solution is 0 raw carrots, 9 baked potatoes, 0 wheat bread, 0 cheddar cheese, and 9 peanut butter.

Exercise 2.4

a)

The transportation problem is a simple constraint problem. Letting the constraint to both that the demand should at least match the given demand, while the supply may at most be the given supply, the solver will solve the problem.

b) This model is an extension to the one in a). To incorporate the extra cost, we will add a binary variable matrix to act as a usage constraint. This constraint will enable a large value (100000 in our case) in a matrix of parameters when enabled, and zero when disabled. Then, another constraint is added to let the path decision matrix be at most the value in this matrix. This means that, when the solver wants to use a path, the binary variable has to be set to one, so that the path limit is unlimited. If the path is not used, the maximum value allowed is zero, effectively disabling the path. And finally, in the objective the sum is taken from the using path matrix, multiplied by the path usage cost (100). So when a path is used, this usage cost is added to the objective.

Exercise 2.5

a) The problem is formulated as an Integer Linear Programming (ILP) problem, using the following setup:

In this setup (see appendix 2.5a), all possible shifts (8-hour and 4-hour shifts starting between mentioned times) are presented in a matrix format. The cost per shift (# hours * hourly wage) and the required number of workers can also be found in this setup. The amount of shifts of a specific type can be found in the column "amount". The number of workers per interval is corresponding to the amount of shifts per type. Lastly, our objective is the total costs (sum of (amount per shift type* cost per shift)). By adjusting the column "amount" (decision variables) with the constraints that the amounts should be integers and that the number of workers satisfy the minimum requirement, we are minimizing our objective (total costs). This is done by using the integrated solver of Excel, as is shown in appendix 2.5a.

The resulting objective value is 3296. This denotes that the minimum cost to cover all intervals with the required number of workers is equal to 3296 euro.

b) To answer this question, the same formulation as above is used with some adjustments (see appendix 2.5b):

- 4-hour shifts are removed.
- Instead of a required number of workers, there is a demanded number of workers. This changes the problem in a way that it is also possible that in some cases the demanded number of workers is not reached (understaffed). In part a, it was only possible to have more scheduled workers than required (overstaffed), because the required number was a hard minimum. Now both situations are possible.
- The objective value is now the total absolute difference between the demanded number of workers and the scheduled number of workers. By solving this ILP, we are minimizing this difference. A low objective value corresponds to reliability in scheduling the demanded amount of workers.

As before, the integrated solver of excel (with the adjustments above) is used to solve this problem (see appendix 2.5b for settings of the solver). As can be seen, the resulting objective value is 63. So the schedule that minimizes the sum of absolute differences between the demanded and scheduled amount of workers has an outcome of 63.

Appendix

Table 1: Round 1

Model	R.2	P
lm(total~expend)	0.145	0.006
lm(total~ratio)	0.007	0.575
lm(total~salary)	0.194	0.001
lm(total~takers)	0.787	0.000

Table 2: Round 2

Model	R.2	P
lm(total~takers+expend)	0.820	0.006
lm(total~takers+ratio)	0.800	0.098
lm(total~takers+salary)	0.806	0.039

Table 3: Round 3

Model	R.2	P
lm(total~takers+expend+ratio)	0.823	0.363
lm(total~takers+ratio)	0.820	0.853

Table 4: Round 1

Variables	P
takers	0.000
expend	0.674
ratio	0.266
salary	0.496

Table 5: Round 2

Variables	P
takers	0.000
ratio	0.034

Variables	P
salary	0.015

Table 6: Round 1

Model	R.2	P
lm(total~expend)	0.145	0.006
lm(total~ratio)	0.007	0.575
lm(total~salary)	0.194	0.001
lm(total~takers)	0.787	0.000
lm(total~takers2)	0.658	0.000

Table 7: Round 2

Model	R.2	P
lm(total~takers+expend)	0.820	0.006
lm(total~takers+ratio)	0.800	0.098
lm(total~takers+salary)	0.806	0.039
lm(total~takers+takers2)	0.873	0.000

Table 8: Round 3

Model	R.2	P
lm(total~takers+takers2+expend)	0.886	0.029
lm(total~takers+takers2+ratio)	0.874	0.634
lm(total~takers+takers2+ratio)	0.886	0.029

Table 9: Round 4

Model	R.2	P
lm(total~takers+takers2+expend+ratio)	0.889	0.294
lm(total~takers+takers2+expend+salary)	0.877	0.466

Table 10: Round 1

Variables	P
takers	0.000
expend	0.292
ratio	0.454
salary	0.968

Variables	P
takers2	0.000

Table 11: Round 2

Variables	P
takers	0.000
expend	0.018
ratio	0.294
takers2	0.000

Table 12: Round 3

Variables	P
takers	0.000
expend	0.028
takers2	0.000

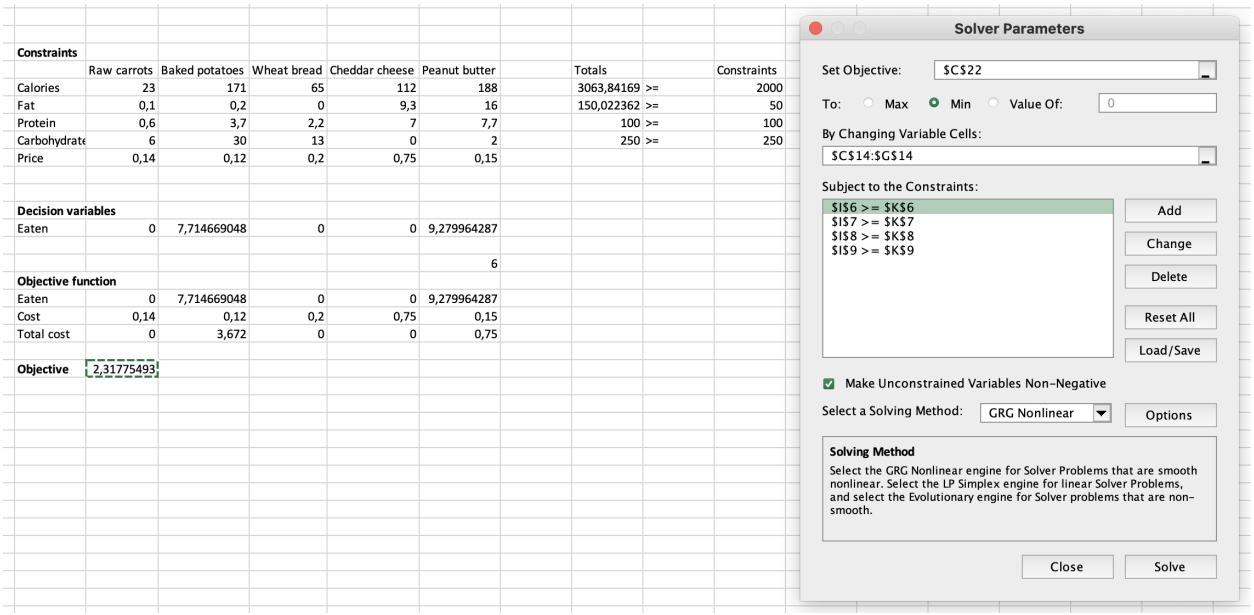
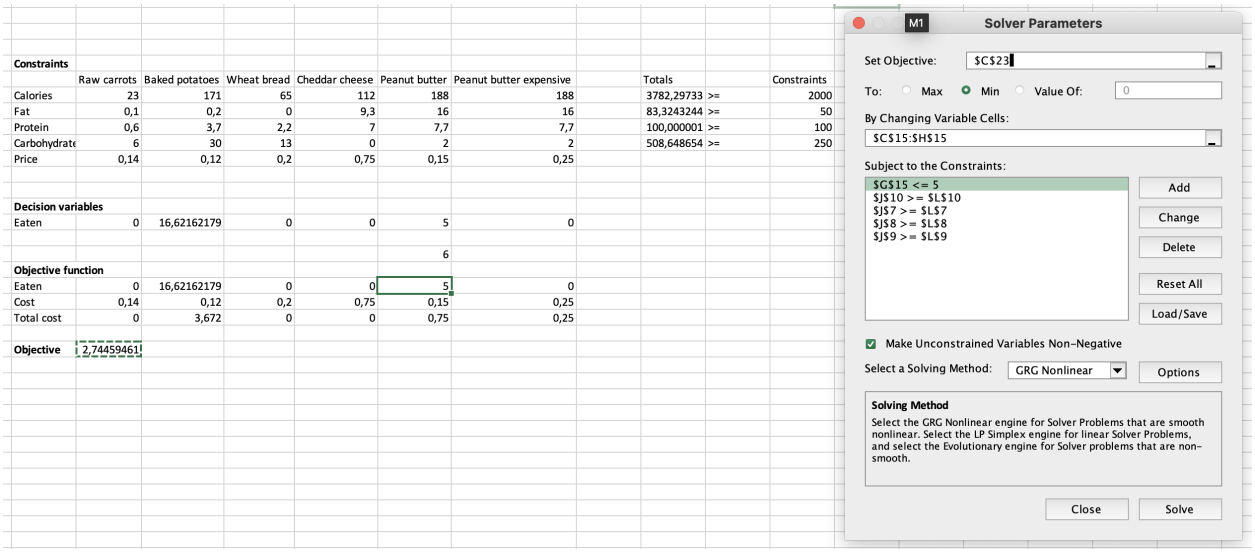
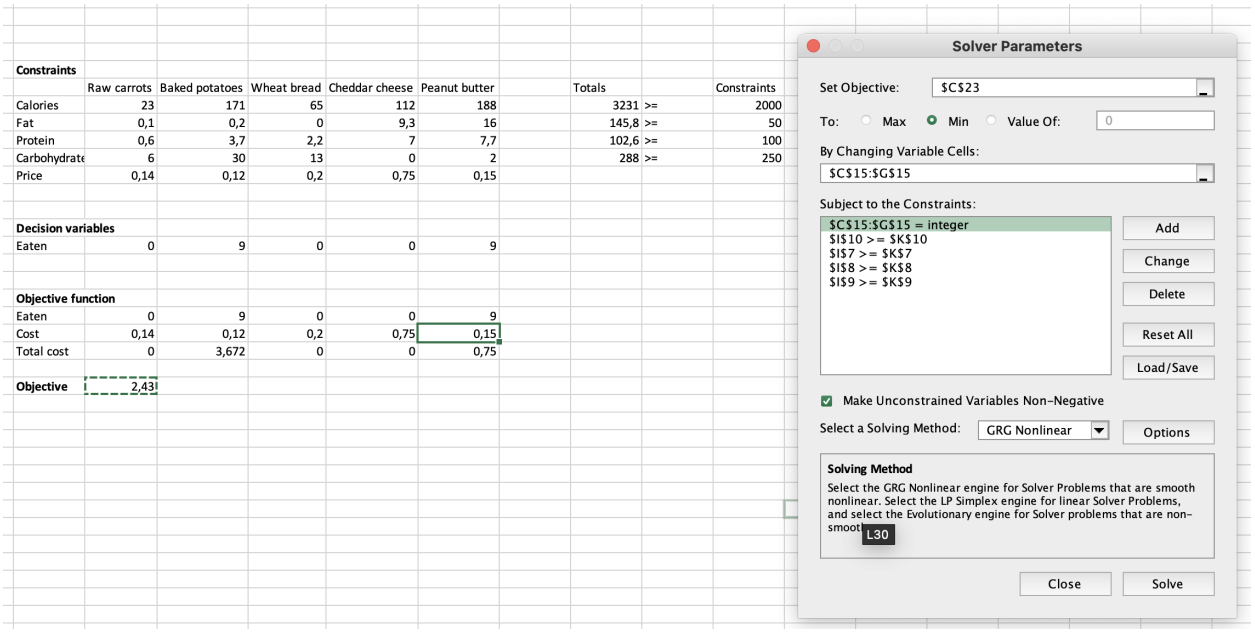


Figure 1: screenshot 2.3a



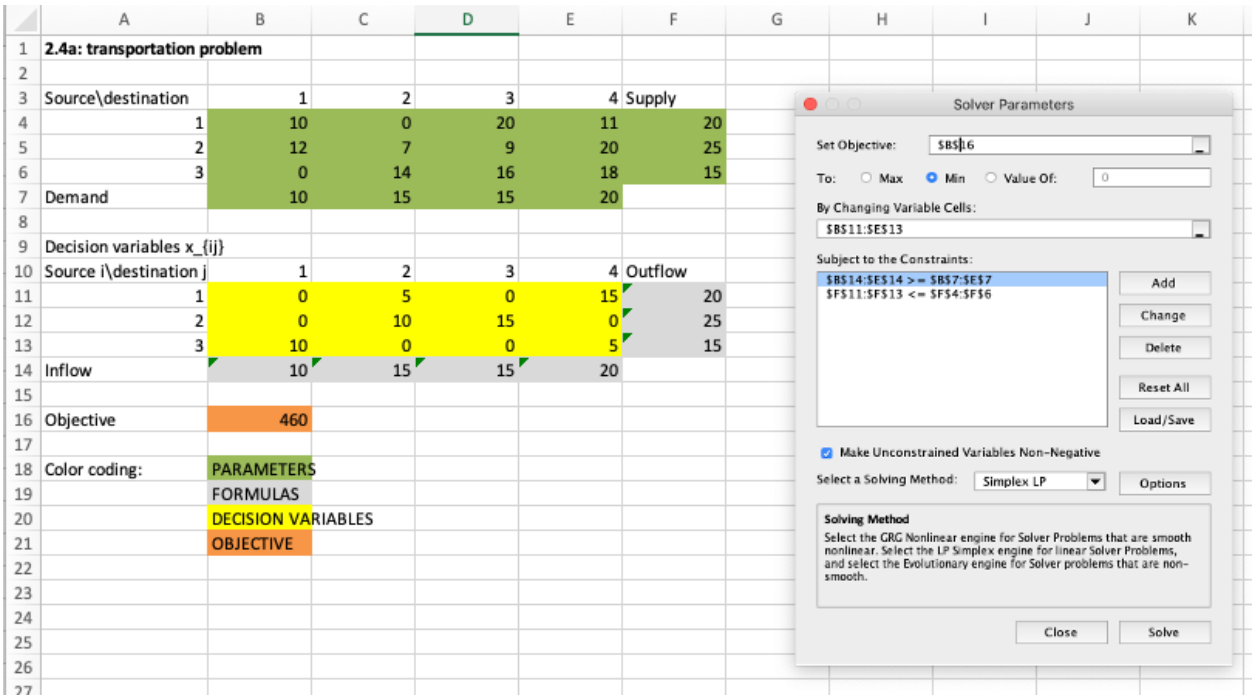
Constraints									
	Raw carrots	Baked potatoes	Wheat bread	Cheddar cheese	Peanut butter	Peanut butter expensive	Totals		Constraints
Calories	23	171	65	112	188	188	3782,29733	>=	2000
Fat	0,1	0,2	0	9,3	16	16	83,3243244	>=	50
Protein	0,6	3,7	2,2	7	7,7	7,7	100,000001	>=	100
Carbohydrate	6	30	13	0	2	2	508,648654	>=	250
Price	0,14	0,12	0,2	0,75	0,15	0,25			
Decision variables									
Eaten	0	16,62162179	0	0	5	0			
					6				
Objective function									
Eaten	0	16,62162179	0	0	5	0			
Cost	0,14	0,12	0,2	0,75	0,15	0,25			
Total cost	0	3,672	0	0	0,75	0,25			
Objective	2,74459461								

Figure 2: screenshot 2.3b



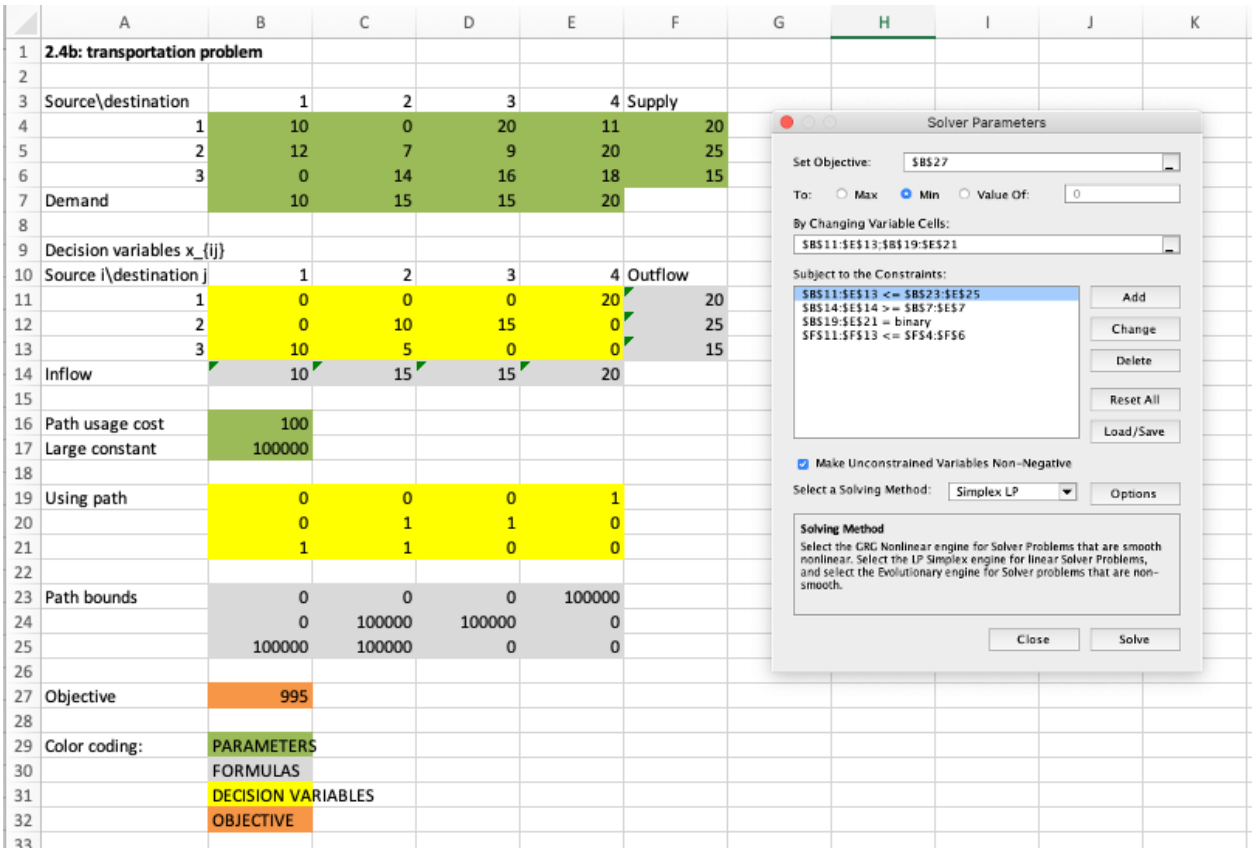
Constraints								
	Raw carrots	Baked potatoes	Wheat bread	Cheddar cheese	Peanut butter	Totals		Constraints
Calories	23	171	65	112	188	3231	>=	2000
Fat	0,1	0,2	0	9,3	16	145,8	>=	50
Protein	0,6	3,7	2,2	7	7,7	102,6	>=	100
Carbohydrate	6	30	13	0	2	288	>=	250
Price	0,14	0,12	0,2	0,75	0,15			
Decision variables								
Eaten	0	9	0	0	9			
Objective function								
Eaten	0	9	0	0	9			
Cost	0,14	0,12	0,2	0,75	0,15			
Total cost	0	3,672	0	0	0,75			
Objective	2,43							

Figure 3: screenshot 2.3c



	A	B	C	D	E	F
1	**2.4a: transportation problem**					
2						
3	Source\destination	1	2	3	4	Supply
4	1	10	0	20	11	20
5	2	12	7	9	20	25
6	3	0	14	16	18	15
7	Demand	10	15	15	20	
8						
9	Decision variables x_{ij}					
10	Source i\destination j	1	2	3	4	Outflow
11	1	0	5	0	15	20
12	2	0	10	15	0	25
13	3	10	0	0	5	15
14	Inflow	10	15	15	20	
15						
16	Objective	460				
17						
18	Color coding:	PARAMETERS				
19		FORMULAS				
20		DECISION VARIABLES				
21		OBJECTIVE				

Figure 4: screenshot 2.4a



	A	B	C	D	E	F
1	**2.4b: transportation problem**					
2						
3	Source\destination	1	2	3	4	Supply
4	1	10	0	20	11	20
5	2	12	7	9	20	25
6	3	0	14	16	18	15
7	Demand	10	15	15	20	
8						
9	Decision variables x_{ij}					
10	Source i\destination j	1	2	3	4	Outflow
11	1	0	0	0	20	20
12	2	0	10	15	0	25
13	3	10	5	0	0	15
14	Inflow	10	15	15	20	
15						
16	Path usage cost	100				
17	Large constant	100000				
18						
19	Using path	0	0	0	1	
20		0	1	1	0	
21		1	1	0	0	
22						
23	Path bounds	0	0	0	100000	
24		0	100000	100000	0	
25		100000	100000	0	0	
26						
27	Objective	995				
28						
29	Color coding:	PARAMETERS				
30		FORMULAS				
31		DECISION VARIABLES				
32		OBJECTIVE				

Figure 5: screenshot 2.4b

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
3																													
4	Time intervals (each 30 min long, starting from 09:00)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Cost per shift	Shift number	Amount
5		1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1								160	S1	9
6			1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1								160	S2	0
7				1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1							160	S3	0
8					1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1						160	S4	1
9						1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1					160	S5	0
10							1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1				160	S6	0
11								1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1			160	S7	0
12									1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1		160	S8	4
13		1	1	1	1	1	1	1	1																		96	S9	1
14			1	1	1	1	1	1	1	1																	96	S10	5
15				1	1	1	1	1	1	1	1																96	S11	0
16					1	1	1	1	1	1	1	1															96	S12	0
17	8-hour shift					1	1	1	1	1	1	1	1														96	S13	0
18	4-hour shift						1	1	1	1	1	1	1	1													96	S14	0
19								1	1	1	1	1	1	1	1												96	S15	0
20									1	1	1	1	1	1	1	1											96	S16	0
21										1	1	1	1	1	1	1	1										96	S17	0
22											1	1	1	1	1	1	1	1									96	S18	0
23												1	1	1	1	1	1	1	1								96	S19	0
24													1	1	1	1	1	1	1	1							96	S20	0
25														1	1	1	1	1	1	1	1						96	S21	0
26															1	1	1	1	1	1	1	1					96	S22	0
27																1	1	1	1	1	1	1	1				96	S23	1
28																	1	1	1	1	1	1	1	1			96	S24	0
29																		1	1	1	1	1	1	1	1		96	S25	4
30																													
31	Required number of workers	10	11	13	16	16	13	11	10	10	11	12	13	14	14	13	11	10	9	9	10	9	8	8	8				
32																													
33																													
34	Number of workers	10	15	15	16	16	16	16	20	10	14	14	13	14	14	15	11	19	10	10	10	9	9	8	8				
35																													
36																													
37	Objective	3296																											

Figure 6: screenshot 2.5a

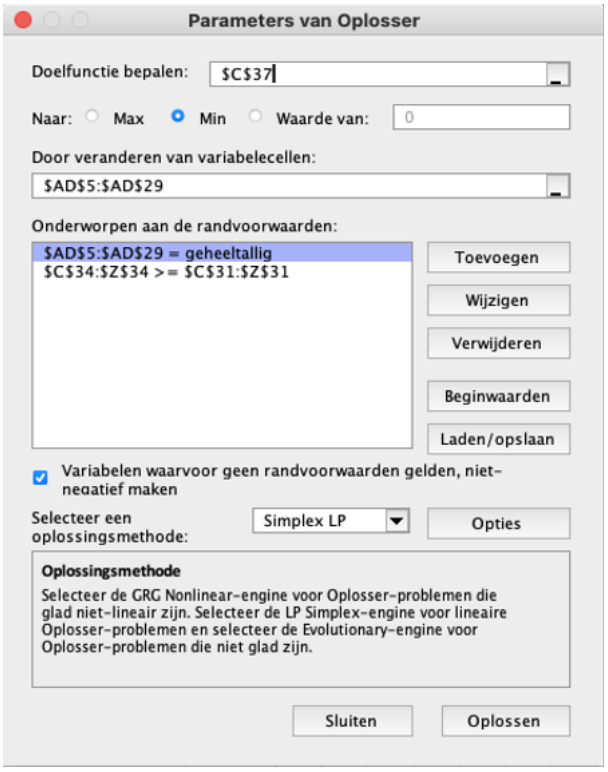


Figure 7: screenshot 2.5a2

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
1																													
2																													
3																													
4	Time intervals (each 30 min long, starting from 09:00)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Cost per shift	Shift number	Amount	
5		1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1								160	S1	5	
6			1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1								160	S2	3	
7				1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1							160	S3	2	
8					1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1						160	S4	1	
9						1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1					160	S5	0	
10							1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1				160	S6	0	
11	8-hour shift							1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1			160	S7	0	
12									1	1	1	1	1	1	1	1		1	1	1	1	1	1	1	1		160	S8	3	
13																													
14	Demanded number of workers	10	11	13	16	16	13	11	10	10	11	12	13	14	14	13	11	10	9	9	10	9	8	8	8				
15																													
16																													
17	Scheduled number of workers	5	8	10	11	11	11	11	14	9	11	12	13	14	14	14	11	14	9	6	4	3	3	3	3				
18	Absolute difference	5	3	3	5	5	2	0	4	1	0	0	0	0	0	1	0	4	0	3	6	6	5	5	5				
19	over/understaffed	-5	-3	-3	-5	-5	-2	0	4	-1	0	0	0	0	0	1	0	4	0	-3	-6	-6	-5	-5	-5				
20	Objective (total absolute difference)	63																											
21																													
22																													
23	positive	0	0	0	0	0	0	0	4	0	0	0	0	0	0	1	0	4	0	0	0	0	0	0	0				
24	negative	5	3	3	5	5	2	0	0	1	0	0	0	0	0	0	0	0	0	3	6	6	5	5	5				
25																													
26																													

Figure 8: screenshot 2.5b

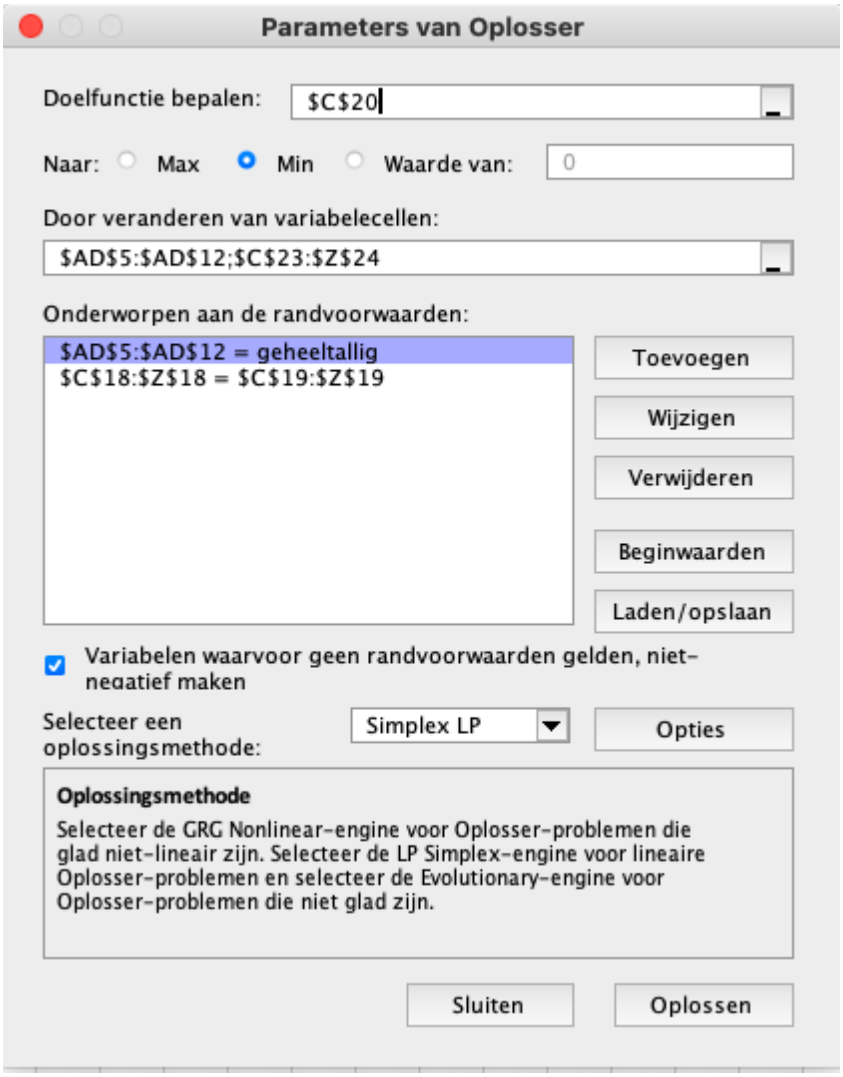


Figure 9: screenshot 2.5b2