Assignment 2

Teun Zwier [redacted], Jelle van der Schoot [redacted], Dragos Pop [redacted]
Group 54

October 2021

Exercise 2.1 - Scholastic Aptitude Test

a) **Step down:** start with highest n of variables and eliminate until all are significant

```
> summary(lm(total ~ expend + ratio + salary +takers,data = sat))
[skipped lines]
Coefficients:
           Estimate Std. Error t value Pr(>|t|)
(Intercept) 1045.9715 52.8698  19.784  < 2e-16 ***
expend        4.4626    10.5465   0.423 0.674
ratio      -3.6242    3.2154  -1.127  0.266
salary        1.6379     2.3872   0.686  0.496
takers     -2.9045    0.2313 -12.559 2.61e-16 ***
```

expend is removed from the formula with the highest p-value(= 0.674)

```
> summary(lm(total ~ ratio + salary + takers, data = sat))
[...]
ratio      -4.6394    2.1215  -2.187   0.0339 *
salary        2.5525     1.0045   2.541   0.0145 *
takers     -2.9134    0.2282 -12.764   <2e-16 ***
[...]
Multiple R-squared:  0.8239,    Adjusted R-squared:  0.8124
```

ratio, salary and takers are all significant, so the resulting model is:

Total = 1057.8982 - 4.6394*ratio + 2.5525*salary - 2.9134*takers + error

With Multiple R-squared: 0.8239, Adjusted R-squared: 0.8124

Step – Up: Iterate through the variables to find the model with highest R-squared and check if added variable is significant.Start with 0 variables and add until there are no variables with significant descriptive value left.

```
lm(total ~ expend, data = sat)  R-squared:  0.1448
lm(total ~ ratio, data = sat)   R-squared:  0.006602
lm(total ~ salary, data = sat)  R-squared:  0.1935
lm(total ~ takers, data = sat)  R-squared:  0.787
```

Takers gives highest R-squared(0.79), so is chosen. Taken is significant, so the step is repeated

```
lm(total ~ takers + expend, data = sat)    R-squared:  0.8195
lm(total ~ takers + ratio, data = sat)     R-squared:  0.7991
lm(total ~ takers + salary, data = sat)    R-squared:  0.8056
```

Expend gives highest R-squared(0.82). Expend is significant, so the step is repeated

```
lm(total ~ takers + expend + ratio,data = sat) R-squared:  0.8196
lm(total ~ takers + expend + salary,data = sat)R-squared:  0.8227
```

Salary is chosen. Salary is not significant (p > 0.05), so the final model is:

Total = 993.8317 - 2.8509*takers + 12.2865*expend + error
Multiple R-squared: 0.8195, Adjusted R-squared: 0.8118

Comparing the two models: The model achieved with the step-down method has a slightly better R-Squared and adjusted R-squared, so this model explains more variability of total. However, it uses an additional variable compared to the step-up method, while the R-squared is only 0.8239 - 0.8195 = 0.0044 higher and the adjusted R-squared only 0.8124 - 0.8118 = 0.0006. Therefore the step-up model is chosen, because it describes total almost equally well, while using one less variable.

b) **Step- Down**
First round: salary is removed with p-value = 0.968
Second round: ratio is removed with p-value = 0.2936
Third round: all variables (expend, takers, takers2) are significant (see table below)

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.052e+03  2.082e+01  50.511  < 2e-16 ***
expend       7.914e+00  3.498e+00   2.262   0.0285 *
takers      -6.381e+00  7.036e-01  -9.068 8.30e-12 ***
takers2      4.741e-02  9.161e-03   5.175 4.87e-06 ***
```

Therefore, the formulated model =
Total = 1052 + 7.914*expend - 6.381*takers + 0.0471*takers2 + error
with Multiple R-squared: 0.8859, Adjusted R-squared: 0.8785

Step-up

```
lm(total ~ takers, data = sat) R-squared:  0.787
lm(total ~ takers2, data = sat) R-squared:  0.6578
```

Again, takers is chosen as the first variable, because takers2 does not give a better result.

```
lm( total ~ takers + expend, data = sat) R-squared:  0.8195
lm(total ~ takers + takers2, data = sat) R-squared:  0.8732
```

takers2 gives higher R-squared than expend and is significant, so it's chosen.

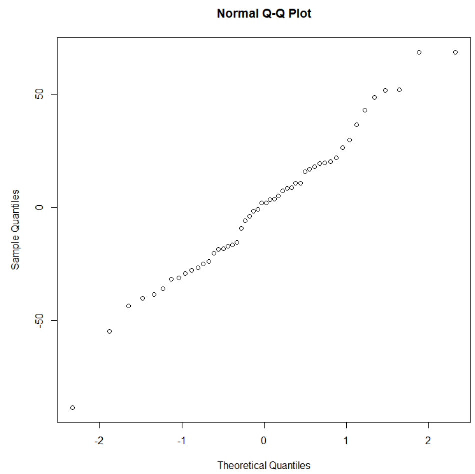
```
lm(total ~ takers + takers2 + expend,data = sat)R-squared:0.8859
lm(total ~ takers + takers2 + ratio,data = sat)R-squared: 0.8738
lm(total ~ takers + takers2 + salary,data = sat)R-squared: 0.8858
```

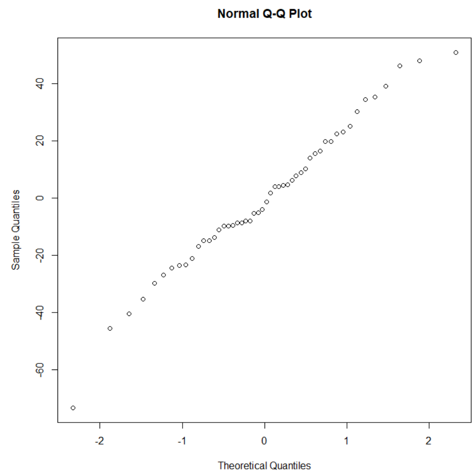
expend has highest R-squared and is significant, so it's chosen

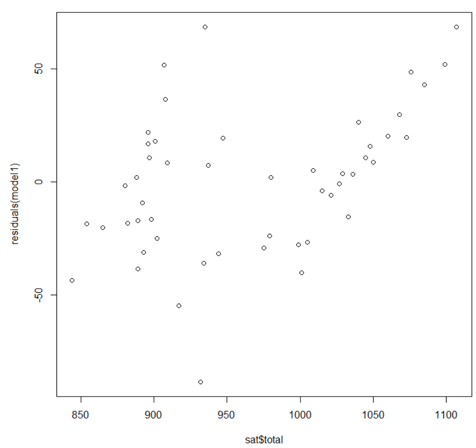
```
lm(total ~ takers + takers2 + expend + ratio, data = sat)
R-squared:  0.8887
lm(total ~ takers + takers2 + expend + salary, data = sat)
R-squared:  0.8873
```

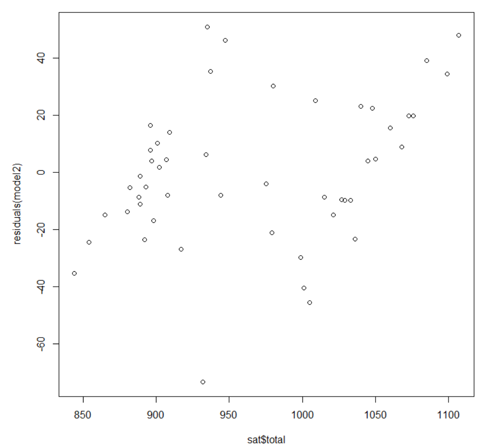
Neither of the variables are significant, so step-up model =
total = 1052 - 6.381*takers + 0.04741*takers2 + 7.914*expend + error
Multiple R-squared: 0.8859, Adjusted R-squared: 0.8785
Chosen model: both methods result in the same model.

c) Model 1 (from a) = Total = 993.8317 - 2.8509*takers + 12.2865*expend + error
Multiple R-squared: 0.8195, Adjusted R-squared: 0.8118
Model 2 (from b) Total = 1052 - 6.381*takers + 0.04741*takers2 + 7.914*expend + error
Multiple R-squared: 0.8859, Adjusted R-squared: 0.8785
Model 2 has the highest R-squared: 0.8859 – 0.8195 = 0.0664 higher than model 1. However, it also uses an additional descriptive variable, takers2. To compare the quality of the models, we look at normality and spread of the residuals (left = model 1, right = model 2).









No big differences can be found in qqnorm plots (residuals seem normally distributed for both models) and the residuals plotted against the total$column show similar structure, so in that aspect the models are similar. Therefore, the question is mainly if the increase in R-squared is worth the additional variable. An increase of 0.066 in R squared, without worsening residuals, is deemed a big enough increase to warrant an additional variable, so the final model is the model from b):
Total = 1052 - 6.381*takers + 0.04741*takers2 + 7.914*expend + error

Concerning the estimated parameters: The parameter for takers2 is small, because the values in takers2 are large due to the quadratic transformation of takers. Furthermore, the parameter for expend is positive (in both models) and takers is negative (in both models), which means the model predicts that higher expenditures will result in higher test scores and a higher percentage of test takers in lower test scores.

d) newxdata=data.frame(expend=5, ratio=20, salary=36.000, takers=25, takers=25**2)

```
> predict(model2,newxdata,interval="predict",level=0.95)
       fit      lwr     upr
1 961.5703 907.6003 1015.54
> predict(model2,newxdata,interval="confidence",level=0.95)
       fit      lwr     upr
1 961.5703 949.0796 974.061
```

Prediction interval = [907.60 1015.54]
Confidence interval = [949.08 974.06]

Exercise 2.2 - Trees

a) For anova, formula = $Y_{ij} = \mu + \alpha_i + e_{ij}$

H0: $\alpha_1 = \alpha_2 = 0$ (no factor effect) H1: at least 1 $\alpha \neq 0$

```
> treelm = lm(volume~type, data=tree); > anova(treelm)
[...]
            Df  Sum Sq Mean Sq F value Pr(>F)
type         1   379.5  379.52  1.8984 0.1736
Residuals 57 11394.8  199.91
```

Type P-value > 0.05 , so the null hypothesis that $\alpha_1 = \alpha_2 = 0$ (i.e. mean diameter is equal between beech and oak trees) cannot be rejected. This means that there is not enough proof to say that an oak is more voluminous than a beech. To get estimates,summary() can be used.

```
> summary(treelm)
[...]
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   30.171      2.539  11.881   <2e-16 ***
typeoak        5.079      3.686   1.378    0.174
```

Estimated volumes: Beech = 30.171. oak = 30.171 + 5.079 = 35.250

b) **Influence of type**

```
> anova(lm(volume~diameter + height + type, data=tree));
[...]
            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
```

The significance of type is again > 0.05, so it's the same conclusion as a): there isn't enough evidence to conclude that type has significant influence on volume. Looking at the summary of the model, there is a negative coefficient for typeoak (= -1.30), meaning with the same height and diameter, the total volume is predicted to be lower for oak trees.

Estimated volumes:

```
> newdata_beech = data.frame(diameter= mean(tree$diameter),
height= mean(tree$height), type='beech')
> newdata_oak = data.frame(diameter= mean(tree$diameter), height=
mean(tree$height), type='oak')
> predict(treelm2,newdata_beech,interval="predict",level=0.95)
       fit      lwr      upr
1 33.20049 26.59383 39.80715
```

Beech estimation = 33.20

```
> predict(treelm2,newdata_oak,interval="predict",level=0.95)
       fit      lwr      upr
1 31.89589 25.27733 38.51445
```

Oak estimation = 31.90

Influence of diameter

```
> anova(lm(volume~height + type + diameter, data=tree))
[...]
           Df Sum Sq Mean Sq F value  Pr(>F)
height      1 2187.6  2187.6 208.025 < 2.2e-16 ***
type        1  431.2   431.2  41.005 3.562e-08 ***
diameter    1 8577.1  8577.1 815.611 < 2.2e-16 ***
Residuals 55  578.4    10.5
```

Diameter has p-value < 0.05, so the null hypothesis that there is no influence is rejected. There is a positive coefficient (= 4.70) for diameter in the model, so the model estimates that volume will be larger if diameter is larger. To investigate if this dependence is similar for both types of trees, anova can be utilized to test interaction between diameter and type in regards to volume.

```
> anova(lm(volume~type*diameter, data=tree))
[…]
type             1   379.5   379.5  23.3742 1.112e-05 ***
diameter         1 10492.3 10492.3 646.2145 < 2.2e-16 ***
type:diameter  1     9.5    9.5    0.5874    0.4467
```

p-value of type:diameter is > 0.05, meaning there is not enough evidence to prove interaction between type and diameter in modelling volume. So therefore it can be assumed that the diameter volume dependence is similar for both tree types.

c) Tree trunks are roughly cylinder-shaped. The volume of a cylinder can be computed with height and radius with the formula $\pi r^2 * h$. diameter/2 = r, so an additional variable was computed like this:

```
> tree$cyl_volume = pi*((tree$diameter/2)**2)*tree$height
```

Then, a model could be created with said variable.

```
> summary(lm(volume~cyl_volume + type, data=tree))
[...]
Multiple R-squared:  0.975,     Adjusted R-squared:  0.9741
```

This model yields a better R-squared score than a model using type, diameter and height (R-squared = 0.95).

Exercise 2.3 - Optimal Product Mix

a) To find the optimal product mix, we define x_i as the number of servings of product i, where i is raw carrots (1), baked potatoes (2), wheat bread (3), cheddar cheese (4) or peanut butter (5), and set up the following LO model:

Objective: cheapest diet = min (0,14*x_1 + 0,12*x_2 + 0,2*x_3 + 0,75*x_4 + 0,15*x_5)

Constraints:

23*x_1 + 171*x_2 + 65*x_3 +112* x_4 + 188*x_5 ≥ 2000 (plan has at least 2000 calories)

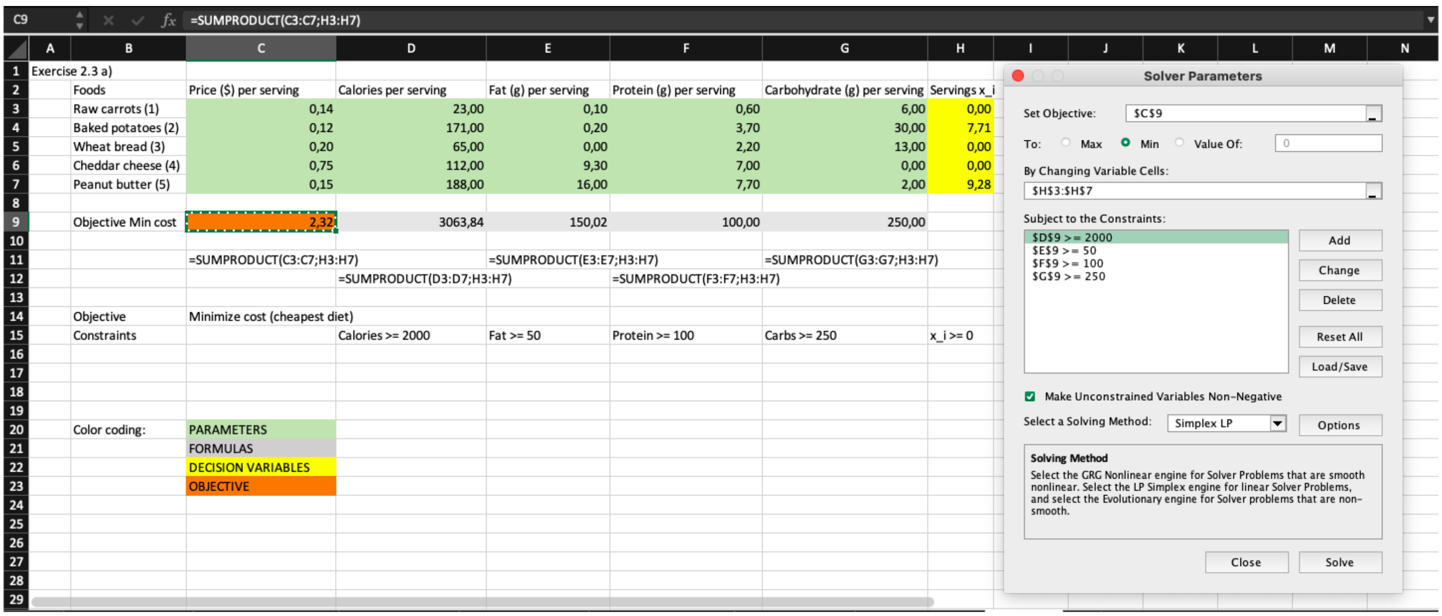
0,1*x_1 + 0,2*x_2 +9,3* x_4 + 16*x_5 ≥ 50 (plan has at least 50g of fat)

0,6*x_1 + 3,7*x_2 + 2,2*x_3 +7* x_4 + 7,7*x_5 ≥ 100 (plan has at least 100g of protein)

6*x_1 + 30*x_2 + 13*x_3 + 2*x_5 ≥ 250 (plan has at least 250g of carbohydrates)

x_1,x_2,x_3,x_4,x_5 ≥ 0 (servings have to be non-negative)

In order to solve the model, Excel solver was used as one can see in the image below, and returned the following servings $x_1 = 0$, $x_2 = 7{,}71$, $x_3 = 0$, $x_4 = 0$, $x_5 = 9{,}28$, meaning that the optimal product mix is composed of 7,71 portions of baked potatoes (2) and 9,28 servings of peanut butter (5) which gives a minimum cost of $2,32. The cost was calculated by multiplying the price of each product with the number of servings and summed together for all the five products. Similarly, the total calories, fats, proteins, and carbohydrates were calculated for the optimal mix using the SUMPRODUCT function of Excel with the number of servings and the respective given variable. Additionally, one can see that these values match the constraints, as well as the number of servings is non-negative, constraint added by checking the box "Make Unconstrained Variables Non-Negative" in the Solver window.



b) Considering from the previous point that over 5 units of peanut butter are added to the optimal product mix, and that the nutritional values of the product will stay the same regardless of the price, the new LO model will include another decision variable x_6 that defines the number of peanut butter servings at $0,25 as well as a constraint that limits x_5 (the number of peanut butter servings at $0,15) to maximum 5. The reason why this works is that after the 5 units of cheap peanut butter will be added to the servings (since at point a) over 5 units were added), the model will have to choose between the expensive peanut butter with the same nutritional values as the cheap version and the rest of the products. Consequently, the new model will be:

Objective: cheapest diet = min $(0{,}14^*x_1 + 0{,}12^*x_2 + 0{,}2^*x_3 + 0{,}75^*x_4 + 0{,}15^*x_5 + 0.25^*x_6)$

Constraints:

$23^*x_1 + 171^*x_2 + 65^*x_3 + 112^*x_4 + 188^*x_5 + 188^*x_6 \geq 2000$ (plan has at least 2000 calories)

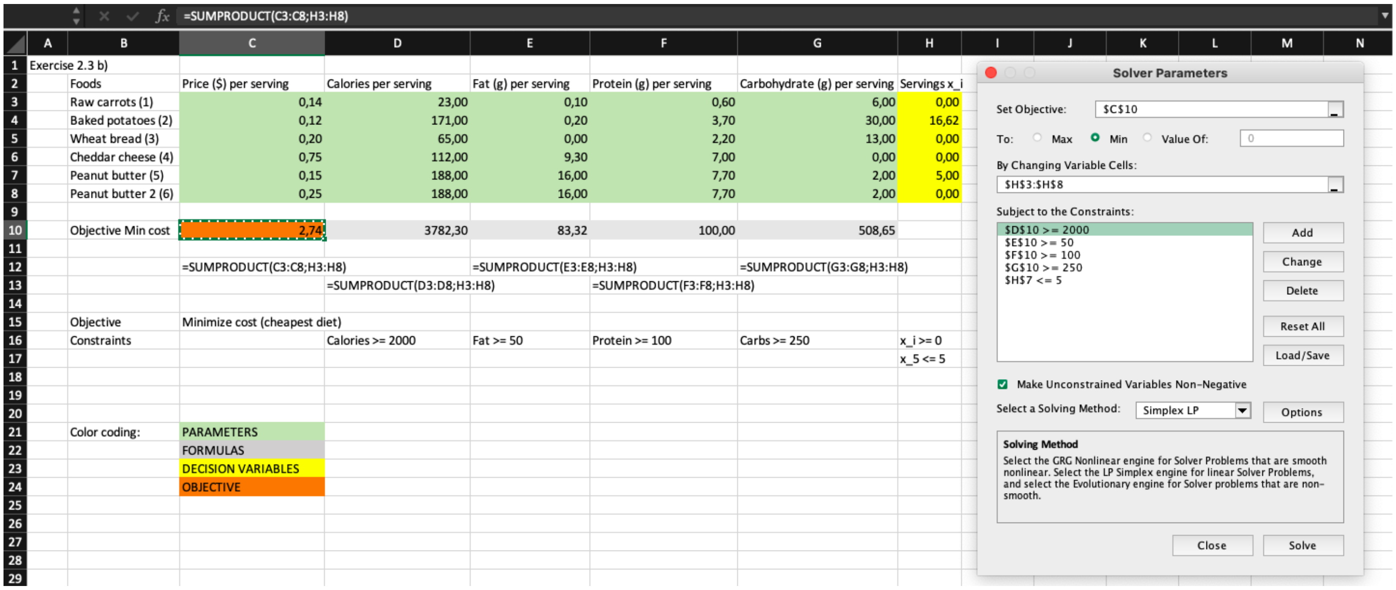
$0{,}1^*x_1 + 0{,}2^*x_2 + 9{,}3^*x_4 + 16^*x_5 + 16^*x_6 \geq 50$ (plan has at least 50g of fat)

$0{,}6^*x_1 + 3{,}7^*x_2 + 2{,}2^*x_3 + 7^*x_4 + 7{,}7^*x_5 + 7{,}7^*x_6 \geq 100$ (plan has at least 100g of protein)

$6^*x_1 + 30^*x_2 + 13^*x_3 + 2^*x_5 + 2^*x_6 \geq 250$ (plan has at least 250g of carbohydrates)

$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$ (servings have to be non-negative)

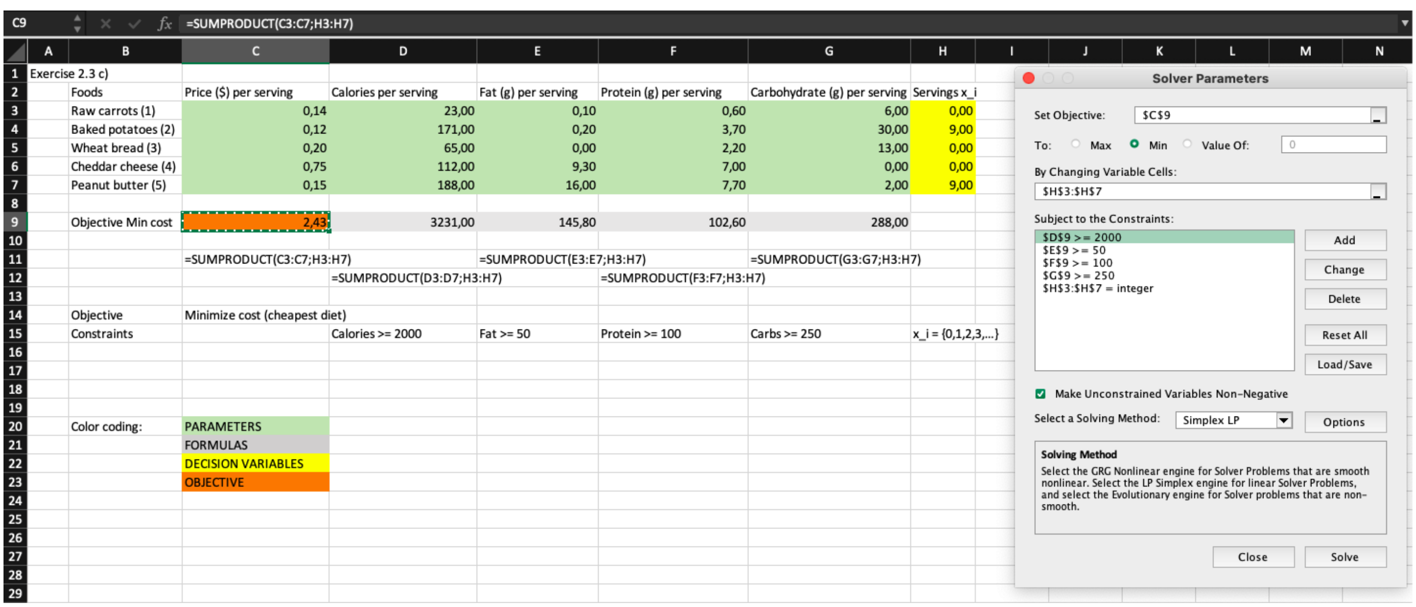
The result is shown in the next image. As expected, the minimum cost increase to \$2,74 because there are more constraints in place, hence, the feasibility region decreases, as well as the 5 servings of cheap peanut butter are selected in the optimal plan. Next, the other ingredient picked is 16,62 portions of baked potatoes, which the model prefers over the peanut butter at \$0,25, which is not selected at all.



=SUMPRODUCT(C3:C8;H3:H8)

	A	B	C	D	E	F	G	H
1	Exercise 2.3 b)							
2		Foods	Price (\$) per serving	Calories per serving	Fat (g) per serving	Protein (g) per serving	Carbohydrate (g) per serving	Servings x_i
3		Raw carrots (1)	0,14	23,00	0,10	0,60	6,00	0,00
4		Baked potatoes (2)	0,12	171,00	0,20	3,70	30,00	16,62
5		Wheat bread (3)	0,20	65,00	0,00	2,20	13,00	0,00
6		Cheddar cheese (4)	0,75	112,00	9,30	7,00	0,00	0,00
7		Peanut butter (5)	0,15	188,00	16,00	7,70	2,00	5,00
8		Peanut butter 2 (6)	0,25	188,00	16,00	7,70	2,00	0,00
10		Objective Min cost	2,74	3782,30	83,32	100,00	508,65	
12			=SUMPRODUCT(C3:C8;H3:H8)		=SUMPRODUCT(E3:E8;H3:H8)		=SUMPRODUCT(G3:G8;H3:H8)	
13				=SUMPRODUCT(D3:D8;H3:H8)		=SUMPRODUCT(F3:F8;H3:H8)		
15		Objective	Minimize cost (cheapest diet)					
16		Constraints		Calories >= 2000	Fat >= 50	Protein >= 100	Carbs >= 250	x_i >= 0
17								x_5 <= 5
21		Color coding:	PARAMETERS					
22			FORMULAS					
23			DECISION VARIABLES					
24			OBJECTIVE					

c) A very similar model to the model described point a) was used to deal with integer servings, the only distinction being the last constraint that was changed from $x_1, x_2, x_3, x_4, x_5 \geq 0$ to $x_1, x_2, x_3, x_4, x_5 \in \{0, 1, 2, 3, \ldots\}$. Other than that, the model was exactly the same as the one at point a). As one can see in the under the paragraph, the number of servings (yellow cells H3:H7) were constrained to be integer, resulting in a minimum cost of \$2,43 composed from 9 portions of baked potatoes and 9 of peanut butter. Accordingly, the cheapest diet is slightly more expensive than that of a) (\$0,09), which makes sense because the integer constraint applied at c) is more restrictive compared to the non-negative one applied at a).



C9 =SUMPRODUCT(C3:C7;H3:H7)

	A	B	C	D	E	F	G	H
1	Exercise 2.3 c)							
2		Foods	Price (\$) per serving	Calories per serving	Fat (g) per serving	Protein (g) per serving	Carbohydrate (g) per serving	Servings x_i
3		Raw carrots (1)	0,14	23,00	0,10	0,60	6,00	0,00
4		Baked potatoes (2)	0,12	171,00	0,20	3,70	30,00	9,00
5		Wheat bread (3)	0,20	65,00	0,00	2,20	13,00	0,00
6		Cheddar cheese (4)	0,75	112,00	9,30	7,00	0,00	0,00
7		Peanut butter (5)	0,15	188,00	16,00	7,70	2,00	9,00
9		Objective Min cost	2,43	3231,00	145,80	102,60	288,00	
11			=SUMPRODUCT(C3:C7;H3:H7)		=SUMPRODUCT(E3:E7;H3:H7)		=SUMPRODUCT(G3:G7;H3:H7)	
12				=SUMPRODUCT(D3:D7;H3:H7)		=SUMPRODUCT(F3:F7;H3:H7)		
14		Objective	Minimize cost (cheapest diet)					
15		Constraints		Calories >= 2000	Fat >= 50	Protein >= 100	Carbs >= 250	x_i = {0,1,2,3,...}
20		Color coding:	PARAMETERS					
21			FORMULAS					
22			DECISION VARIABLES					
23			OBJECTIVE					

Exercise 2.4 - Transportation Problem

a) To solve the transportation problem, we define the decision variables: x_{ij} = quantity transported from i to j and set up the following LO model:

Objective: cheapest transportation plan = min $(\sum_{i,j} c_{ij} x_{i,j})$, where c_{ij} is the given transportation cost of one unit from i $\in \{1,2,3\}$ to j $\in \{1,2,3,4\}$

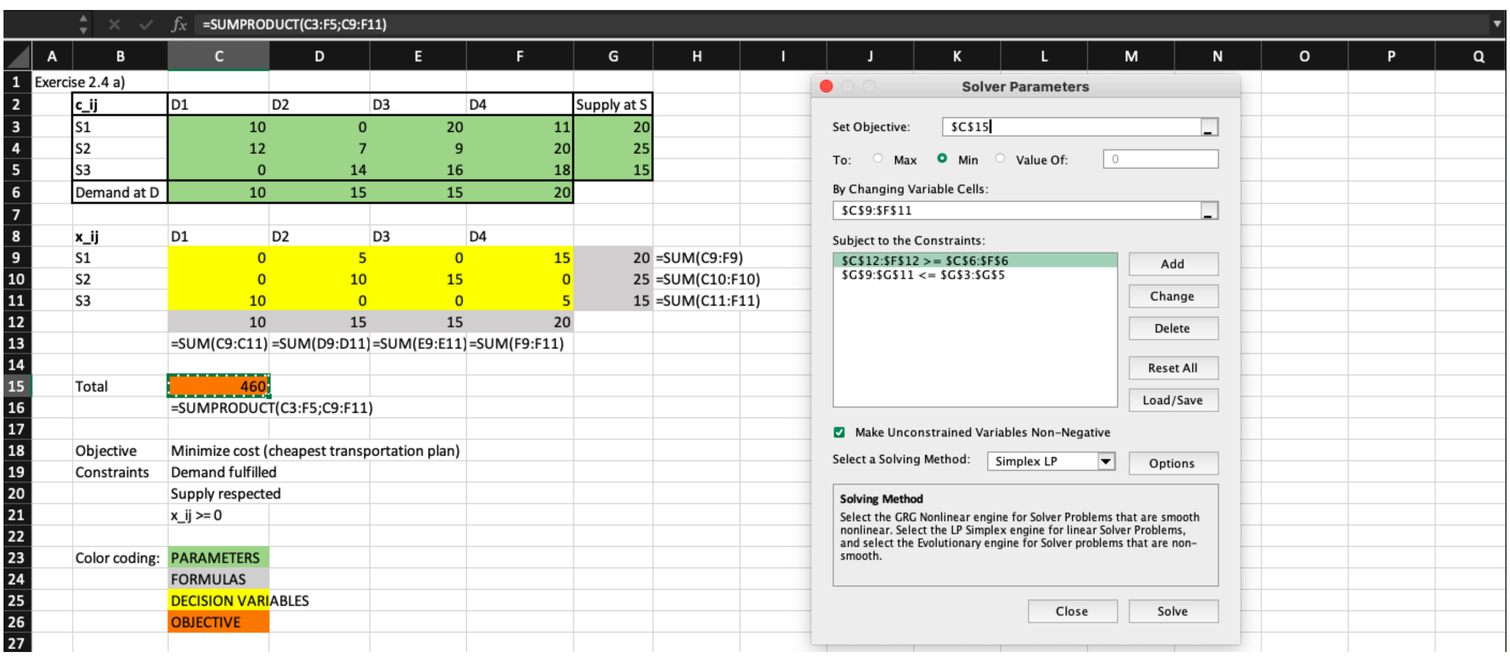
Constraints:

$\sum_j x_{ij} \leq a_i,\ i \in \{1, 2, 3\}$ (the source a does not deliver more supplies than it has available)

$\sum_i x_{ij} \geq b_j,\ j \in \{1, 2, 3, 4\}$ (the demand at destination b is fulfilled)

$x_{ij} \geq 0$ (quantities transported are non-negative)

In order to solve the model, Excel Solver was used as one can see in the picture below. This found a minimum transportation cost of 460 euro by computing the product between the cells of the given green table with the yellow cells, namely the cost of transportation of one unit times the number of units transported via that particular route. Additionally, it is visible in the picture that the demand (grey cells C12:F12), computed as the sum of the column, is satisfied, while the supplies delivered from each source (grey cells G9:G11), computed as the row sum, is not larger that the given available supply at the respective source. Lastly, the non-negativity constraint is added by checking the box "Make Unconstrained Variables Non-Negative" in the Solver window, and the resulting quantities between source and destination can be seen in the yellow table.



b) To model this problem, another binary decision variable y_{ij} has to be added to the model:

y_{ij}={$0\ if\ x_{ij} = 0$ (when route i to j is not used)

{$1\ if\ x_{ij} \geq 0$ (when route j to j is used)

Objective: cheapest transportation plan = min ($\sum_{i,j} c_{ij} x_{i,j}$ + 100*$\sum_{i,j} y_{i,j}$) (adds 100 for every used edge)

Constraints:

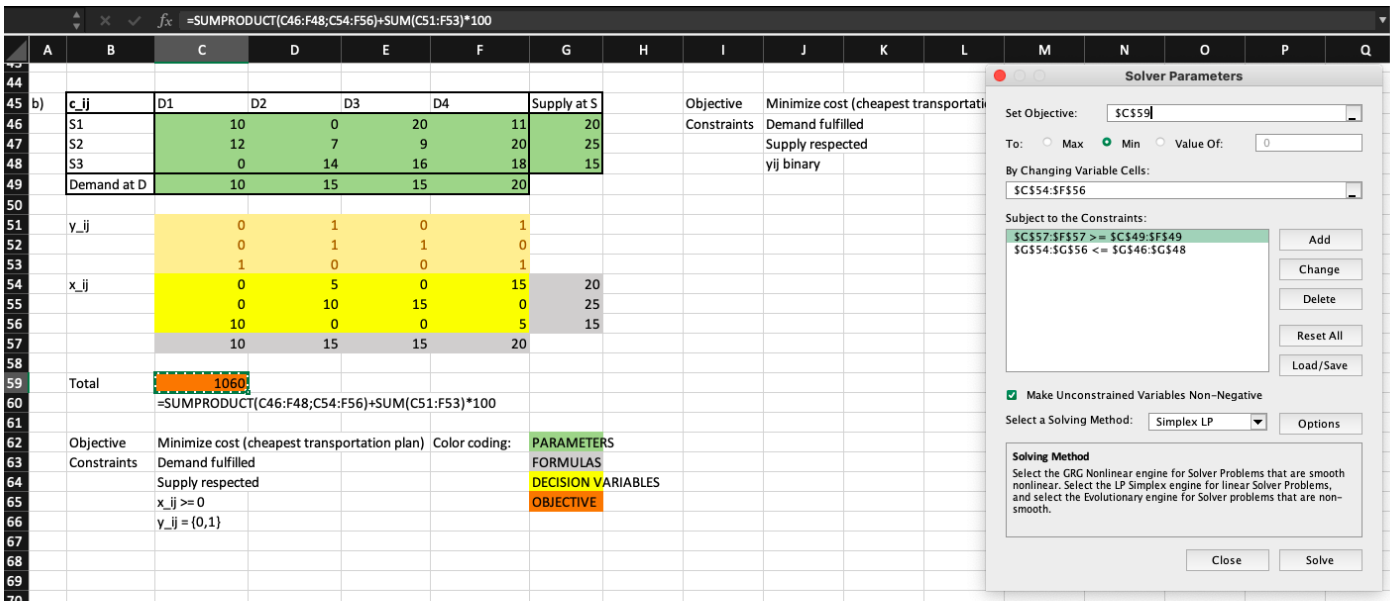
$\sum_{j} x_{ij} \leq a_i,\ i \in \{1, 2, 3\}$ (the source a does not deliver more supplies than it has available)

$\sum_{i} x_{ij} \geq b_j,\ j \in \{1, 2, 3, 4\}$ (the demand at destination b is fulfilled)

x_{ij} ≥0 (quantities transported are non-negative)

y_{ij} = {0,1}

As expected, the cheapest transportation cost increases drastically to 1060 euro since now every edge used adds 100 euro to the transportation cost. However, the same number of units are transported via a route as in the previous point. This means that the number of used edges could not be reduced without resulting in a larger transportation cost due to the limited room associated with the supply and demand constraints.



Exercise 2.5 - Call Center Staffing

a) We have a so-called universe U = {1,...,m}, and sets S1,...,Sn, with Si ⊂ U. We are looking for the smallest selection of sets that covers U. In other words: what is the minimum number of shifts to schedule to meet the requirements?

Introduced variables:

- a_{iu} = 1 if shift i works at time u, 0 else
- c_u = cost of shift i (160 for shift i = {1,...,8} and 96 for shift i = {9,...,25})
- x_i = number of workers scheduled with shift i

In Excel, the following matrix was created with every shift and the a_{iu} value to determine whether a certain shift works at a certain time. Note that the 8 hour shifts include a (unpaid) 30 minute break after 4 hours; this has been incorporated into the matrix.

	Time interval	09:00 - 09:30	09:30 - 10:00	10:00 - 10:30	10:30 - 11:00	11:00 - 11:30	11:30 - 12:00	12:00 - 12:30	12:30 - 13:00	13:00 - 13:30	13:30 - 14:00	14:00 - 14:30	14:30 - 15:00	15:00 - 15:30	15:30 - 16:00	16:00 - 16:30	16:30 - 17:00	17:00 - 17:30	17:30 - 18:00	18:00 - 18:30	18:30 - 19:00	19:00 - 19:30	19:30 - 20:00	20:00 - 20:30	20:30 - 21:00					
Shift time	Shift i / Hour u	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Shift i	Cost		Shift i	Schedule
09:00 - 17:30	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	160		1	7
09:30 - 18:00	2	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	2	160		2	0
10:00 - 18:30	3	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	3	160		3	0
10:30 - 19:00	4	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	4	160		4	0
11:00 - 19:30	5	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	5	160		5	0
11:30 - 20:00	6	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	6	160		6	0
12:00 - 20:30	7	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	7	160		7	0
12:30 - 21:00	8	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	8	160		8	6
09:00 - 13:00	9	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	96		9	4
09:30 - 13:30	10	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	96		10	0
10:00 - 14:00	11	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	96		11	5
10:30 - 14:30	12	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	12	96		12	0
11:00 - 15:00	13	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	13	96		13	0
11:30 - 15:30	14	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	14	96		14	0
12:00 - 16:00	15	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	15	96		15	0
12:30 - 16:30	16	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	16	96		16	0
13:00 - 17:00	17	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	17	96		17	0
13:30 - 17:30	18	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	18	96		18	0
14:00 - 18:30	19	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	19	96		19	0
14:30 - 18:30	20	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	20	96		20	0
15:00 - 19:00	21	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	21	96		21	1
15:30 - 19:30	22	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	22	96		22	0
16:00 - 20:00	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	23	96		23	1
16:30 - 20:30	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	24	96		24	0
17:00 - 21:00	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	25	96		25	2
	Required	10	11	13	16	16	13	11	10	10	11	12	13	14	14	13	11	10	9	9	10	9	8	8	8				Cost	€ 3.328
	Scheduled	11	11	16	16	16	16	16	22	18	11	13	13	14	14	15	15	11	10	10	10	9	9	8	8					

The number of workers scheduled at a certain time can be computed using the following formula:
`SUMPRODUCT(time intervals;scheduled shifts)`
The optimal solution can be computed after adding the Solver constraints:

1. For each interval u, the number of scheduled workers must be greater than the number of required workers
2. Planned shifts must be integers

Solution: [7, 0, 0, 0, 0, 0, 0, 6, 4, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 2] with cost: € 3.328.

b) The 4h shift is dropped and a new variariable is introduced: z_i. This is necessary because it is impossible to solve this ILP that minimizes the absolute difference between the required and scheduled workers, because the abs() is not linear. The new variable works around this by adding two constraints for each value of z_i: it must be bigger or equal to the positive and negative differences.

$$min \sum_i z_i$$

s.t.:

- $z_i \geq required - scheduled$ for i = {1, ..., 8}
- $z_i \geq -(required - scheduled)$ for i = {1, ..., 8}
- z_i and *schedule*-values are integers

	Time interval	09:00 - 09:30	09:30 - 10:00	10:00 - 10:30	10:30 - 11:00	11:00 - 11:30	11:30 - 12:00	12:00 - 12:30	12:30 - 13:00	13:00 - 13:30	13:30 - 14:00	14:00 - 14:30	14:30 - 15:00	15:00 - 15:30	15:30 - 16:00	16:00 - 16:30	16:30 - 17:00	17:00 - 17:30	17:30 - 18:00	18:00 - 18:30	18:30 - 19:00	19:00 - 19:30	19:30 - 20:00	20:00 - 20:30	20:30 - 21:00		
Shift time	Shift i / Hour u	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Shift i	Schedule
09:00 - 17:30	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	13
09:30 - 18:00	2	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	2	0
10:00 - 18:30	3	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	3	0
10:30 - 19:00	4	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	0	4	0
11:00 - 19:30	5	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0	5	3
11:30 - 20:00	6	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	6	0
12:00 - 20:30	7	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	7	0
12:30 - 21:00	8	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	8	8
																										Min	121
	Required	10	11	13	16	16	13	11	10	10	11	12	13	14	14	13	11	10	9	9	10	9	8	8	8		
	Scheduled	13	13	13	13	16	16	16	24	24	11	24	24	24	21	24	24	16	11	11	11	11	8	8	8		
	Difference	-3	-2	0	3	0	-3	-5	-14	-14	0	-12	-11	-10	-7	-11	-13	-6	-2	-2	-1	-2	0	0	0		
	z_i	3	2	0	3	0	3	5	14	14	0	12	11	10	7	11	13	6	2	2	1	2	0	0	0		
	Neg. Difference	3	2	0	-3	0	3	5	14	14	0	12	11	10	7	11	13	6	2	2	1	2	0	0	0		

Solution:

- Schedule: [13,0,0,0,3,0,8]
- z_i: [3, 2, 0, 3, 0, 3, 5, 14, 14, 0, 12, 11, 10, 7, 11, 13, 6, 2, 2, 1, 2, 0, 0, 0]
- Difference: 121.