Assignment 1

This assignment is an individual assignment and should be made individually. Do not copy or share code from/to others. The same holds for your report. Make it on an individual basis.

Hand in a .pdf on Canvas containing your answers to the questions and submit your R-code. Do not put them all together in a zip-file but upload the documents separately. The Deadline is June 17th, 2022. To pass this course you need at least a grade of 4.5 for this assignment. Thus, if your grade is lower than 4.5, you must go for the improvement option to pass the course. If your grade for this assignment is lower than 5.5 (but higher than 4.5), you have an option to improve your assignment. The deadline to hand in the improvement option is 7 days after the grades of this assignment are released.

This assignment sums up to 100 point. Your grade can be computed by dividing your total points by 10. Note that, following the grading guidelines, the grade will be rounded to one decimal.

Question 1: Project management

In knowledge clip 2.6 of week 2 a project management example is introduced. The video started with: "what do you need to do?".

1. (35 points) You are the project manager of this project, write a brief note for you manager, in which you indicate which decisions need to be taken and why. Please answer this question by solving this problem with (Integer) Linear programming in R. *Note that you are assumed to program it in R in an efficient way (e.g. using loops), manually typing in all constraints can lead to a maximum reduction of 15 points.*

In the video, this problem was introduced as an integer linear program. We can also assume that all decision variables are continuous, especially in case the fine of 70 euros per day is paid by the second (e.g. if the project is ready in 6.2 days, the fine is 6.2x70).

2. (10 points) Solve the problem as a (continuous) linear program, does the solution change? Could you explain why (not)?
3. (10 points) Can you come up with situations in which you expect a difference between the linear and integer linear program of this problem?
4. (20 points) Your manager is a bit worried due to some uncertainties which may arise in the coming days. Perform a sensitivity analysis. Report all the information, in easy language, that you can get from this sensitivity analyses.

Hints:

To make your computations more straightforward, you can leave the "fixed" cost $\sum_{i=1}^{6} c_i^{min}$ which is equal to 250 out of the LP and just add them in the last line of your code to the objective function value of the LP.

In the video we start to count from i=0 to 7, for convenience we have changed that in the assignment to i=1 to 8 (also in my explanations below).

You have a total of 16 variables (8 x variables and 8 y variables). Let us from now on consider that the first 8 values relate to the x variables and the second 8 variables to the y variables. The LP solver requires you to give a parameter for each of these 16 variables for each constraint and for the objective function.

For example, for the objective function you must insert a vector 'v' with 16 numbers. You have to put a 0 for each variable which is not present in your objective. E.g. v[1]<-0, as x_1 is not present in the objective and v[10]<- a_2=40 as y_2 (the 10th variable refers to y_2) is multiplied in the objective with a_2.

As another example, one of the constraints for the problem is:

$$x_3 \geq x_2 + r_2 - y_2$$

As said in one of the clips, we like to put all variables in the left-hand side part of the constraint and all parameters in the right-hand side. Most solvers, including the one we use in R have a hard requirement on this. Therefore, we need to transform this constraint into:

$$-x_2 + x_3 + y_2 \geq r_2$$

Then the coefficient vector of this constraint is:

[0, -1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0] and this should be added as constraint with right hand side r_2 and sign >=.

So, for each constraint you add, you must perform the following steps:

- Add (use the combine function) the coefficient vector of this constraint to the matrix with all coefficient vectors. Which you add, once completed, as constraint matrix (f.con).
- Add (use the combine function) the sign of the constraint (>= or <=, or ==) to the vector of all signs. Which you add, once completed, as sign vector (f.dir).
- Add (use the combine function) the value of the right-hand side to the vector of all right-hand side values. Which you add, once completed, as RHS vector (f.rhs).

Last hint on the logic of adding the predecessor constraints:

For each activity j, check for each other activity i whether it is a predecessor of j (you can use the last 8 columns for this in the dataset where row j column i indicates whether i is a predecessor of j) if so, add the constraint with the coefficient vector as discussed above. You can get the location of the y_i variable of activity i by v[i+8].

Question 2: Using travel duration in transportation problems.

Consider that you are a manager of the planning department of a transportation company. Your employee is struggling with the transportation plan for next week. So far, your company has used a strategy to determine routes based on shortest travel distance. It seems that these routes do not always perform well in terms of travel time. Therefore, you decide that from now on your department will make use of travel times (instead of distances) to determine the routes. A disadvantage of this is that it is sometime worthwhile to consider a detour between customers, meaning that you must evaluate the travel times on all possible detours.

You have 30 customers and you know all direct travel times between each pair of customers as indicated in the attached table as Duration_Matrix.csv. Element (i, j) in this table refers to the direct travel time in minutes from customer i (row i) to customer j (column j). Note that the duration can be different depending on the direction a truck is travelling, for example due to one-way roads, different traffic conditions etc. Therefore, in general, element (i, j) is not equal to element (j, i).

As you can see in the table, sometimes a detour is worthwhile. The direct travel time from customer 5 to customer 4 is equal to 248 minutes. However, going from customer 5 first to customer 2 (91 minutes) and then from customer 2 to customer 4 (137 min) is equal to a total of 228 minutes and thus shorter. A tour from customer 5 via customer 3 to customer 4 is even 220 minutes. Maybe a detour via 2 other customers is even shorter etc. Finding this out by hand for all possible pairs of customers will be impossible.

Luckily, you know someone who has studied dynamic programming in her prescriptive analytics course and she came up with the following idea:

Let $D_{i,j,k}$ be the shortest (in duration) path between customer i and j, using only nodes 0 to k as intermediaries.

Observation: the shortest path from customer i to j, using only nodes 0 to k as intermediates should either not contain k as intermediate node or should contain k as intermediate node.

If the shortest path using only nodes 0 to k as intermediates does not contain k as intermediate node, the shortest path is the same as the shortest path from customer i to j, using only nodes 0 to $k-1$ as intermediates, so in that case $D_{i,j,k} = D_{i,j,k-1}$

If the shortest path does contain k, then the sum of $D_{i,k,k-1}$ is the duration of the shortest path from i to k using only nodes 0 to $k-1$ as intermediates (k cannot be an intermediate as it is the end node) and the duration $D_{k,j,k-1}$ of the shortest path from k to j using only nodes 0 to $k-1$ as intermediates was lower than $D_{i,j,k-1}$ (not including k).

$$\text{Summarized } D_{i,j,k} = \min\{D_{i,j,k-1},\ D_{i,k,k-1} + D_{k,j,k-1}\}.$$

5. (35 points) It is now your task to use this dynamic program to transform the table with direct durations between each pair of customers into a table with the shortest duration via any possible subtour between each pair of customers.

Hints:

$D_{i,j,0}$ should be equal to the table of direct durations, which is provided, as no intermediate nodes are allowed here.

In the end we are interested in the table $D_{i,j,30}$

Start this exercise small, e.g. try to solve it first for 5 customers and once your code works, scale it up to the 30 customer dataset.