LP Team Time

Question 1

Fender guitars recently released the American Professional II series of guitars – this represents the mid-tier of the American-made instruments. The lowest tier of the American series is the American Perform and the highest tier is the Ultra.

Average profits are as follows:

Performer: 1200
Professional II: 1300
Ultra: 1400

Given the shocks to global supply chains, materials are in short supply. Every guitar in this series requires the following components:

1 body blank, 1 neck blank, 1 tuner set, 1 electronic set, 1 bridge set

All guitars use the same body and neck blanks. The Professional II and Ultra use the same electronics, bridges, and tuners. The Performer uses different electronics, bridges, and tuners than the others. Currently, Fender has 1500 neck and body blanks each. Fender also has 500 tuner sets, electronic sets, and bridge sets for the Professional II and Ultra series. They also have 1000 tuner, electronic, and bridge sets for the Performer series. Most of the demand occurs at the upper and lower series, but Fender knows that they need at least 200 Professional IIs. How should Fender maximize profits for these 3 different series?

```
library(ROI)
library(ROI.plugin.glpk)

cvec <- c(performer = 1200,
          proII = 1300,
          ultra = 1400)

bvec <- c(1500, 1500, rep(500, 3), rep(1000, 3), 200)

# There will be liberal use of rep throughout. Much easier
# than dealing with typing all of those out.

direction <- c(rep("<=", 8), ">=")

Amat <- rbind(all_neck = c(1, 1, 1), # All require the same neck and body
              all_body = c(1, 1, 1),
              pro_ultra_tuner = c(0, 1, 1), # Only the pro & ultra use these
              pro_ultra_elec = c(0, 1, 1),
              pro_ultra_bridge = c(0, 1, 1),
              performer_tuner = c(1, 0, 0), # Only the performer uses these
              performer_elec = c(1, 0, 0),
              performer_bridge = c(1, 0, 0),
              proII_demand = c(0, 1, 0)) # Pro II demand in action

linprog::solveLP(cvec, bvec, Amat, TRUE, direction)
```

```
Results of Linear Programming / Linear Optimization

Objective function (Maximum): 1880000

Iterations in phase 1: 1
Iterations in phase 2: 3
Solution
           opt
performer 1000
proII      200
ultra      300

Basic Variables
           opt
performer 1000
proII      200
ultra      300
S 2          0
S 4          0
S 5          0
S 6          0
S 7          0
S 8          0

Constraints
  actual dir bvec free dual dual.reg
1   1500  <= 1500    0 1200     1000
2   1500  <= 1500    0    0       NA
3    500  <=  500    0  200      300
4    500  <=  500    0    0       NA
5    500  <=  500    0    0       NA
6   1000  <= 1000    0    0       NA
7   1000  <= 1000    0    0       NA
8   1000  <= 1000    0    0       NA
9    200  >=  200    0  100      300

All Variables (including slack variables)
           opt cvec min.c max.c  marg marg.reg
performer 1000 1200     0  1400    NA       NA
proII      200 1300  -Inf  1400    NA       NA
ultra      300 1400  1300   Inf    NA       NA
S 1          0    0  -Inf  1200 -1200     1000
S 2          0    0  -Inf  1200     0       NA
S 3          0    0  -Inf   200  -200      300
S 4          0    0  -Inf   200     0       NA
S 5          0    0  -Inf   200     0       NA
S 6          0    0  -200  1200     0       NA
S 7          0    0  -200  1200     0       NA
S 8          0    0  -200  1200     0       NA
S 9          0    0  -Inf   100  -100      300
```

```r
OP(cvec, L_constraint(Amat, direction, bvec), maximum = TRUE) |>
  ROI_solve() |>
```

```r
solution()
```

```
performer     proII     ultra 
     1000       200       300 
```

Question 2

You like making money? I thought so! Let's imagine that each of the following investment strategies has some type of yearly ROI percent, assessed risk factor, and terms of investment in years:

Table 1:

	Strategy	ROI	Risk	Term
1	Blue chip stocks	12	2	4
2	Bonds	10	1	8
3	Growth stocks	15	3	2
4	Speculation	25	4	10
5	Cash	0	0	0

We want to maximize the return, but our risk should not exceed 2.5, we should not exceed 6 years of investment, and we need at least 15% of our portfolio to be in cash. What proportion should be invested in each type?

```r
# We can just pass our ROI values into the cvec as they are.

cvec <- c(bcStocks = 12, bonds = 10, gStocks = 15,  spec = 25, cash = 0)

# Here is where we play with some tricks. We will have the proportion
# constraint set to 1 and hold cash at .15. These two things together
# help to constrain the solution to proportions.

bvec <- c(proportion = 1, risk = 2.5, year = 6, cash = .15)

direction <- c("==", "<=", "<=", ">=")

aMat <- rbind(proportion = c(1, 1, 1, 1, 1),
              # The 1's, coupled with equal to 1, will keep values to proportions.
              risk = c(2, 1, 3, 4, 0),
              year = c(4, 8, 2, 10, 0),
              cash = c(0, 0, 0, 0, 1))

linprog::solveLP(cvec, bvec, aMat, TRUE, direction, lpSolve = TRUE)
```

```

Results of Linear Programming / Linear Optimization
(using lpSolve)

Objective function (Maximum): 15.5286 

Solution
                opt
bcStocks 0.4071429
bonds    0.0285714
```

```
gStocks   0.0000000
spec      0.4142857
cash      0.1500000

Constraints
           actual dir bvec free
proportion   1.00  == 1.00    0
risk         2.50  <= 2.50    0
year         6.00  <= 6.00    0
cash         0.15  >= 0.15    0
```

```
OP(cvec, L_constraint(aMat, direction, bvec), maximum = TRUE) |>
  ROI_solve() |>
  solution()
```

```
  bcStocks      bonds    gStocks       spec       cash
0.40714286 0.02857143 0.00000000 0.41428571 0.15000000
```

Question 3

You think "business" is the only place that uses optimization? Let's check out an example from medicine. When someone is getting radiation, there are some goals:

1. Destroy the tumor

2. Minimize normal tissue damage

3. Avoid organs

While this is done with a few hundred radiation beams, let's try to model it with 2.

We have the following cell types:

- normal tissue (n; what should not be damaged)
- critical tissue (c; think important organ tissue)
- target tissue (t; bad tissue)
- target center (tCenter; the middle of the bad tissue)

Table 2:

	area	beam1	beam2	rules
1	n	0.400	0.500	minimize
2	c	0.300	0.100	<= 2.8
3	t	0.500	0.500	== 6
4	tCenter	0.600	0.500	>= 6

```
cvec <- c(beam1 = .4, beam2 = .5)

bvec <- c(critical = 2.8, tumor = 6, tumor_center = 6)

direction <- c("<=", "==", ">=")

aMat <- rbind(critical = c(.3, .1),
              tumor = c(.5, .5),
```

```
              tumor_center = c(.6, .5))

linprog::solveLP(cvec, bvec, aMat, FALSE, direction, lpSolve = TRUE)
```

```
Results of Linear Programming / Linear Optimization
(using lpSolve)

Objective function (Minimum): 5.2

Solution
      opt
beam1   8
beam2   4

Constraints
             actual dir bvec free
critical        2.8  <=  2.8  0.0
tumor           6.0  ==  6.0  0.0
tumor_center    6.8  >=  6.0  0.8
```

```
OP(cvec, L_constraint(aMat, direction, bvec), maximum = FALSE) |>
  ROI_solve() |>
  solution()
```

```
beam1 beam2
    8     4
```

If you have `lpSolve` set to `FALSE` in the `solveLP` function, you will get a warning and an incorrect answer. Probably safe to just always have it set to TRUE.

Question 4

You need a team and you have a budget of **objective function value from question 1**. You need to get as many people on your team as possible. You need at least 3 stats people and at most 3 utility people. Given their intense dislike for each other, you can't have Seth and Sharif on the same team.

Seth: utility (150K)
Sharif: utility (150K)
Martin: utility (150K)
David: utility (150K)
Brandon: utility (150K)
John: utility (250K)

Francis: stats (150K)
Huy: stats (150K)
Hong: stats (250K)
Zifeng: stats (250K)
Junghee: stats (250K)
Maggie: stats (250K)
Ken: stats (350K)

Who are you taking?

The big question here is what are we doing? If we want to maximize the number of people on the team, then the c vector is going to just be 1's (every person counts as 1). You have a budget constraint, a stats constraint, and a utility constraint. You've also got to incorporate the S^2 constraint (we both have a 1 on that row of the matrix and the margin has to be less than or equal to 1).

When dealing with the stats total, your constraint matrix row will give the stats people a 1 and a 0 to everyone else; the exact same thing holds with utility, but the utility people get a 1 and the stats people get a 0.

```r
cvec <- rep(1, 13)

bvec <- c(budget = 1880000,
          stats = 3,
          utility = 3,
          ss = 1)

directions <- c("<=", ">=", "<=", "<=")

aMat <- rbind(budget = c(rep(150000, 5), 250000, rep(150000, 2),
                rep(250000, 4), 350000),
              stats = c(rep(0, 6), rep(1, 7)),
              utility = c(rep(1, 6), rep(0, 7)),
              ss = c(1, 1, rep(0, 11)))

out <- lpSolve::lp("max", cvec, aMat, directions, bvec, all.bin = TRUE)

people_picked <- out$solution

names(people_picked) <- c("Seth", "Sharif", "Martin", "David",
                          "Brandon", "John", "Francis", "Huy",
                          "Hong", "Zifeng", "Junghee", "Maggie",
                          "Ken")

people_picked
```

```
   Seth  Sharif  Martin   David Brandon    John Francis     Huy    Hong  Zifeng
      0       0       1       1       1       0       1       1       1       1
Junghee  Maggie     Ken
      1       0       1
```

```r
OP(cvec, L_constraint(aMat, directions, bvec),
   types = rep("B", length(cvec)), maximum = TRUE) |>
  ROI_solve() |>
  solution() |>
  setNames(c("Seth", "Sharif", "Martin", "David",
                          "Brandon", "John", "Francis", "Huy",
                          "Hong", "Zifeng", "Junghee", "Maggie",
                          "Ken"))
```

```
   Seth  Sharif  Martin   David Brandon    John Francis     Huy    Hong  Zifeng
      0       1       1       1       0       0       1       1       1       1
Junghee  Maggie     Ken
      0       1       1
```

Remember that whole "no promise of one optimal solution"? Looking us right in the faces here. We can see that different solvers returned different results.

Question 5

Just down the road from Notre Dame is Elkhart – the World's RV Capital! While they might not admit it, most of the manufactures use common components. Thor, Jayco, and Fleetwood all get their engines from 1 of 2 Cummins Engine plants. The following table is on the shipping department supervisor's desk.

Table 3:

	Engine_Plants	Thor_shipping	Jayco_shipping	Fleetwood_shipping	RVs_Needed
1	EP_1	40	30	20	300
2	EP_2	14	25	35	300
3	RVs_Needed	300	35	400	400

Help ship the engines where they are needed, but do it as cheaply as you can.

If specified correctly, this problem should run into trouble: mostly because there is more demand than inventory. If you get answers with the question as written something might be weird. What is even more troubling is the concept of a real NA or a wrong NA (improper specifications with NAs would be wrong).

```r
cvec <- c(ep1_thor = 40, ep1_jay = 30, ep1_fleet = 20,
          ep2_thor = 14, ep2_jay = 25, ep2_fleet = 35)

bvec <- c(ep1_avail = 300, ep2_avail = 300,
          thor_demand = 300, jay_demand = 35, fleet_demand = 400)

directions <- c("<=", "<=", "==", "==", "==")

aMatrix <- rbind(ep1_avail = c(1, 1, 1, 0, 0, 0),
                 ep2_avail = c(0, 0, 0, 1, 1, 1),
                 thor_demand = c(1, 0, 0, 1, 0, 0),
                 jay_demand = c(0, 1, 0, 0, 1, 0),
                 fleet_demand = c(0, 0, 1, 0, 0, 1))

linprog::solveLP(cvec, bvec, aMatrix, FALSE, directions, lpSolve = TRUE)
```

```
Results of Linear Programming / Linear Optimization
(using lpSolve)
lpSolve returned error code '2'
```

With only 600 engines available, this model runs into problems with those demand constraints. To get this problem to solve, something has to change (i.e., demand or capacity).

Now, let's see a more sensible problem and solution. If we bump up the engines available at both plants to 500, and even set our needs a bit higher, we will be in better shape for solving the problem.

Table 4:

	Engine_Plants	Thor_shipping	Jayco_shipping	Fleetwood_shipping	Engines_Avail
1	EP_1	40	30	20	500
2	EP_2	15	25	35	500
3	RVs_Needed	300	300	400	

```r
cvec <- c(ep1_thor = 40, ep1_jay = 30, ep1_fleet = 20,
          ep2_thor = 15, ep2_jay = 25, ep2_fleet = 35)

bvec <- c(ep1_avail = 500, ep2_avail = 500,
          thor_demand = 300, jay_demand = 300, fleet_demand = 400)

directions <- c("<=", "<=", "==", "==", "==")

aMatrix <- rbind(ep1_avail = c(1, 1, 1, 0, 0, 0),
                 ep2_avail = c(0, 0, 0, 1, 1, 1),
                 thor_demand = c(1, 0, 0, 1, 0, 0),
                 jay_demand = c(0, 1, 0, 0, 1, 0),
                 fleet_demand = c(0, 0, 1, 0, 0, 1))

linprog::solveLP(cvec, bvec, aMatrix, FALSE, directions, lpSolve = TRUE)
```

```

Results of Linear Programming / Linear Optimization
(using lpSolve)

Objective function (Minimum): 20500

Solution
          opt
ep1_thor    0
ep1_jay   100
ep1_fleet 400
ep2_thor  300
ep2_jay   200
ep2_fleet   0

Constraints
             actual dir bvec free
ep1_avail       500  <=  500    0
ep2_avail       500  <=  500    0
thor_demand     300  ==  300    0
jay_demand      300  ==  300    0
fleet_demand    400  ==  400    0
```

```r
OP(cvec, L_constraint(aMatrix, directions, bvec), maximum = FALSE) |>
  ROI_solve() |>
  solution()
```

```
 ep1_thor   ep1_jay ep1_fleet  ep2_thor   ep2_jay ep2_fleet 
        0       100       400       300       200         0 
```