Python 最適化第 2 回：最適化による研究室配属

学生の志望を満たしつつ，効果的に授業が行えるような研究室配属を，最適化問題を解くことで実行する方法を学習する．

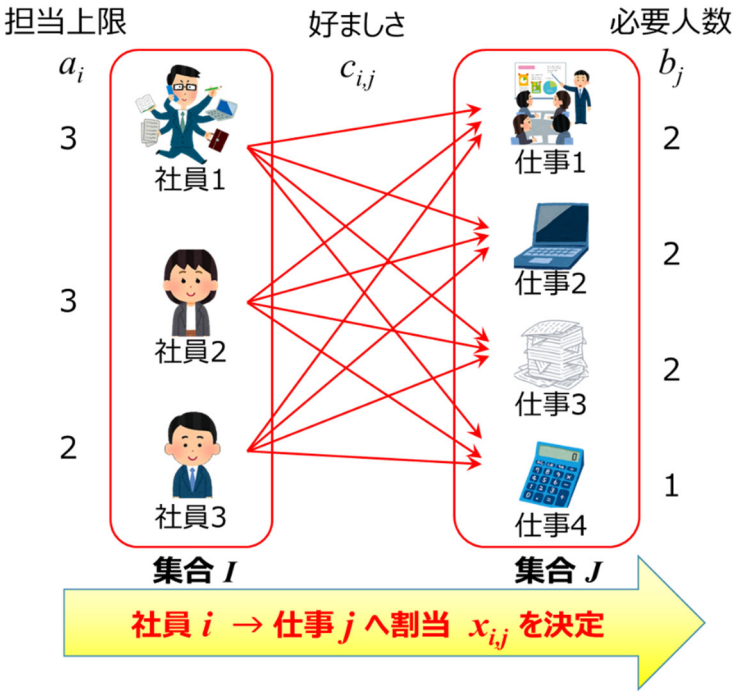
1. 仕事割当問題

1.1 対象の問題（シート「割当」参照）

ある会社のある部署には 3 人の社員が所属し，4 種類の仕事を担当している．どの社員がどの仕事を担当するか，決定したい．

ある社員がある仕事を担当したときの好ましさは，社員と仕事のペアごとに設定されている．また，仕事によって必要とする人数が異なり，必要な人数だけ社員を割り当てたい（余分な割当も行わない）．さらに各社員が担当できる仕事の種類の数（担当上限）も異なり，担当上限までの種類の仕事しか担当できない．

これらの条件を考慮して，好ましさの総和が最も大きくなる割り当て方を決定しよう．



上の図を見てわかるように，割当問題は輸送問題と構造が似ている．ただし，輸送問題では決定変数である輸送量 $x_{i,j}$ は実数であったのに対して，割当問題では決定変数 $x_{i,j}$ は「割り当てる／割り当てない」を表すバイナリ変数（0-1 変数）になる点が異なる．

Excel ファイルのシート「割当」を見て，シート内のデータがどの記号で表されるものか，確認しておこう．

1.2 定数

- a_i　　社員 i が担当する仕事の個数の上限（担当上限）．$i = 1, 2, 3.$
- b_j　　仕事 j が必要とする社員の数（必要人数）．$j = 1, 2, 3, 4.$
- $c_{i,j}$　　社員 i を 仕事 j に割り当てたときの好ましさ．$i = 1, 2, 3.$ $j = 1, 2, 3, 4.$

上の記号はこの後のプログラムでも用いるので，Excel 内の数値と合わせ，一通り確認すること．

1.3 最適化問題の構成要素

① 決定変数

$x_{i,j}$　　社員 i を仕事 j に割り当てる／割り当てないを表す．バイナリ（0-1）型

ここでは，割り当てる場合に 1，割り当てない場合に 0 を取ることとしよう．

② 制約条件

- 社員が担当する仕事の数は，担当上限を超えない
- 仕事には必要とする数以上の社員を割り当てる．
- （決定変数は 0 か 1 しか取らない）・・・決定変数をバイナリ（0-1）型にすることで実現

③ 目的関数

割り当てた好ましさの総和の最大化

1.4 定式化（輸送問題の定式化を参考にしよう）

最？化　　$\sum_{?=?}^{?} \sum_{?=?}^{?} ?$

制約条件　　$\sum_{?=?}^{?} ??? \quad (i = 1, 2, 3)$　　社員 i が担当する仕事数の条件

$\sum_{?=?}^{?} ??? \quad (j = 1, 2, 3, 4)$　　仕事 j への割当人数の条件

$x_{i,j} = 0 \ or \ 1 \quad (i = 1, 2, 3. \ j = 1, 2, 3, 4)$　　各社員－仕事間の割当する／しない

◎ 課題 1

仕事割当問題を定式化し，最適解を求めるプログラムを作成しなさい．

Excel ファイル最適化 2.xlsx のシート「割付」の緑色セルのデータを読み込み，割当結果を黄色セルに出力（割り当てた場合に 1 を出力）して，ファイル最適化 2-1.xlsx として保存すること．（最適値：48）

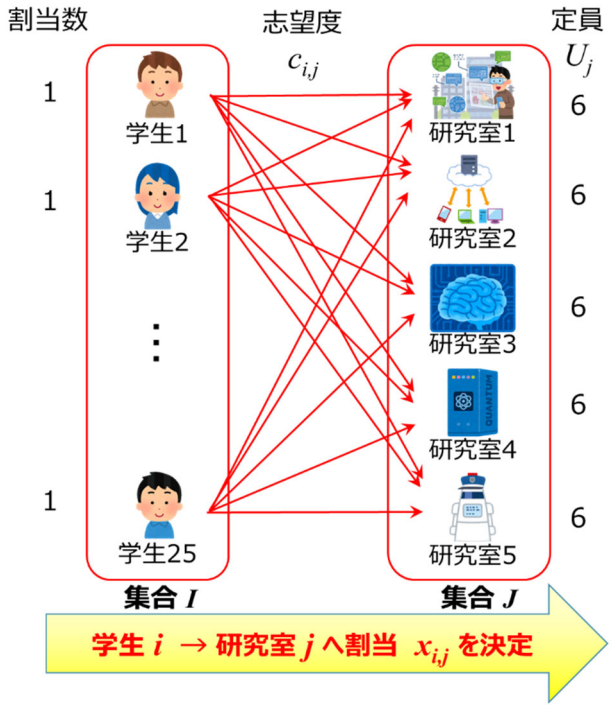
【ヒント】

前回の輸送問題のプログラムの一部をコピペして，仕事割当問題のプログラムになるように追加・変更しよう．

2. 研究室配属問題（その 1）

2.1 対象の問題（Excel シート「配属 1」参照）

- 25 人の学生を 1 つの研究室に配属する．
- 研究室は 5 つ存在する．定員はすべて共通して 6 人．
- 各研究室に対する学生の志望度合いが調査された．学生は少なくとも 1 つの研究室は 100 点満点を付け，志望が高いほど高い点数（以下，「志望度」とする）を付ける．空欄は 0 点．
- 配属された研究室に対する志望度の合計が最大になるような配属を求めたい．



2.2 定数

- $c_{i,j}$　　学生 i を研究室 j に配属したときの志望度．（セル範囲 C4:G28）
- U_j　　研究室 j の定員（配属人数の上限）．（セル範囲 K30:O30）

Excel ファイルのシート「配属 1」を見て，シート内のデータがどの記号で表されるものか，確認しておこう．

2.3 決定変数

- $x_{i,j}$　　学生 i を研究室 j に配属するなら 1，配属しないなら 0．バイナリ（0-1）型．
  例) 学生 1 を研究室 2 に配属する場合，$x_{1,2} = 1$
  学生 3 を研究室 4 に配属しない場合，$x_{3,4} = 0$

2.4 制約条件

- それぞれの学生は，ちょうど 1 つの研究室に配属する．
- それぞれの研究室において，配属される人数は定員以内である．
- （決定変数は 0 か 1 しか取らない）・・・決定変数をバイナリ（0-1）型にすることで実現

2.5 目的関数

学生の配属による志望度の総和の最大化

2.6 定式化

最？化　$\sum_{?=?}^{?}\sum_{?=?}^{?} ?$

条件　$\sum_{?=?}^{?} ? ? ?$　$(i = 1, 2, \dots, 25)$　学生 i に対する条件

$\sum_{?=?}^{?} ? ? ?$　$(j = 1, 2, \dots, 5)$　研究室 j に対する条件

$x_{i,j} = 0 \ or \ 1 \quad (i = 1, 2, \dots, 25, \ j = 1, 2, \dots, 5)$　割当する／しない

2.7 Python プログラム（途中から）：定数用データの作成

???の部分は，自分で考えてプログラムを作成する．

```
I = [i+1 for i in range(25)]    #学生番号の集合（リスト）
J = [i+1 for i in range(5)]     #研究室番号の集合（リスト）
c = {}  #志望度点数
```

```
for i in I:
    for j in J:
        v = sheet.cell(row=3+i, column=2+j).value   #セルの読込
        if type(v) is int:
            c[i, j] = ???      #整数が入力されていれば，点数はその整数値
        else:
            c[i, j] = ???      #整数が入力されていない場合，点数は 0
U = {}
for j in J:
    U[j] = sheet.cell(row=30, column=10+j).value          #定員 Uj の読込
```

- セルに代入された値を変数 v に格納する．
  【余裕のある人向け解説】セルが未入力の場合，None というオブジェクトが v に格納される．None は「非存在」を表すもので，C 言語でいう null と同じ．
- Python は型宣言することなく変数を使うことができ，同じ変数の中に様々な型の値を代入することができる．ある変数 A の中に現在格納されている値の型は，type(A)で調べる．「変数 A の中に整数型の値が入っているなら」という条件は type(A) is int と書く．
  type(A) == int とは書かない．== は値が等しいときに使う．

2.8 Python プログラム（途中から）：最適化問題の作成

```
model = LpProblem('Lab', sense=Lp???)

x = {}
for i in I:
    for j in J:
        x[i, j] = LpVariable(f'x{i},{j}', cat=LpBinary)

for i in I:
    model += lpSum(??? for ??? in ???) ??? ???
for j in J:
    model += lpSum(??? for ??? in ???) ??? ???

model += lpSum(??? for i in I for j in J)
```

- 最大化なら LpMaximize，最小化なら LpMinimize
- 決定変数はバイナリ型（0-1 型）なので，cat=LpBinary で指定する．
- 【余裕がある人向けの解説】目的関数の設定行における for i in I for j in J は for (i, j) in x でも OK.「辞書 x にキーとして登録されている全ての(i, j)の組に対して」という意味になる．

2.9 Python プログラム（途中から）：最適化の結果出力

```
    for i in I:
        for j in J:
            if value(x[i, j]) > 0.01:
                sheet.cell(row=3+i, column=10+j).value = 1
```

- x[i, j] はバイナリ型なので 0 か 1 しか取らないが，計算誤差が生じると，0 が 0.00000001 のように極小の正の値になる可能性もある．そのケースに備えて value(x[i, j]) ==1 と記す代わりに value(x[i, j]) > 0.01 と記している．
- どの変数がどのセルに出力されるのか，よく確認しておこう
  （後のプログラムでは，row と column の値を自分で指定します）．

◎ 課題 2

研究室配属問題（その 1）を定式化しなさい．記号で与えられている定数・係数は記号を用いること（具体的な数値は用いない）．また，配属を出力するプログラムを作成しなさい．

Excel ファイル最適化 2-1.xlsx のシート「配属 1」から必要なデータを読み込み，求めた結果を同シート内に出力して，ファイル最適化 2-2.xlsx として保存すること．（最適値：2460）

◎ 課題 3（Excel シート「配属 1-2」参照）

学生の志望度を重視して割り当てた結果，配属人数が少ない研究室が存在したため，各研究室に配属する最低人数も考慮することになった．新たに 31 行目に入力された研究室 j への配属人数の下限 L_j を読み込み，課題 2 の問題に「研究室 j への配属人数を L_j 以上にする」という条件を追加した研究室配属を実行するプログラムを作成しなさい．

Excel ファイル**最適化 2-2.xlsx** のシート「配属 1-2」から必要なデータを読み込み，求めた結果を同シート内に出力して，ファイル**最適化 2-3.xlsx** として保存すること．（最適値：2430）

【余裕のある人向けのコメント】

課題 2 と比較すると，学生にとって不要な条件が追加されるため，学生の志望度合計は低下する．ただ，この条件追加が全ての学生にとって不利益をもたらす訳ではない．課題 2 での配属結果と比較して，志望度が高い研究室に配属される学生もいる．（シート「配属 1」と「配属 1-2」を比較するとよい．配属された研究室の志望度の値は白色で表示されている．）全体的に何か改悪があっても，一部の人には改善になったり，全体的に何か改善しても，一部の人には改悪になることも往々にして起こる（全体を良くするために，一部の人が犠牲になることもあるし，その逆もありうる，ということですね）．

3. 研究室配属問題（その２）

3.1 対象の問題（Excel シート「配属２」参照）

- 118 人の学生を 3 つの研究室に配属する．研究室数は 18.
- 学生の志望情報が，割り当ての点数ではなく，志望順位に変更．順位が入力されていない研究室は志望外
- 研究室の定員が設定済み（123 行目）．
- それ以外は，配属問題（その１）と同様．

3.2 志望順位の扱いに対するアプローチ

- （その１）の最適化問題を最小化に変更する方法

研究室配属問題（その１）の志望度を志望順位に変更して，目的関数を志望順位の合計の最小化に変更する方法では正しい配属結果にならない．なぜなら，志望外の研究室が「0 位」として扱われるので，「最小化」＝「なるべく志望外の研究室に割り当てる」ことになるからである．

- 上の方法＋志望外を第 10 希望として扱う方法

空白のセルを 0 ではなく 10 に変更して，目的関数を最小化にする方法であれば，なるべく志望順位の高い研究室に割り当てることになる．

ただ，目的関数は志望順位の合計なので，少し問題が生じる．例えば(1 位，2 位，6 位)の 3 つの研究室への割当と，(2 位，3 位，4 位)の割当を考えると，志望順位の合計は 2 つとも 9 となり，「志望順位の合計の最小化」という観点では，同じ「良さ」を持つ割当となる．

実際には，1 位に希望する研究室への志望度は他の順位の研究室への志望度より群を抜いて大きいことが多く，(2 位，3 位，4 位)のように比較的上位志望の 3 研究室への割当よりも，第 1 希望への配属がある(1 位，2 位，6 位)の割当の方を重視することが多い.

この問題点は，各順位の差が均一であることである．1 位と 2 位との差が，8 位と 9 位との差と等しくなっている.1 位と 2 位との差と比較すると,8 位と 9 位の差はあまりないことが多い.

- 志望順位の高い研究室ほど指数関数的に大きな志望度（点数）を付ける方法

上の方法の問題点が解決できるように，1 位と 2 位との差が，8 位と 9 位との差よりも大きくなるように，順位に応じた志望度（点数）を付ける方法である.

例えば，次の表のように志望順位に対する志望度を付ける.

順位	1	2	3	4	5	6	7	8	9	志望外
志望度	20	15	10	8	6	4	3	2	1	0

このように志望度を付けて，目的関数を志望度の合計の最大化とすれば，より志望順位の高い研究室に割り当てることができる．また，目的関数が「志望度の合計の最大化」なので，研究室配属問題（その 1）の最適化問題をベースに利用することができる.

（注：上の志望度（点数）は例であり，他の数値の与え方をしても構わない).

3.3 定数

- $r_{i,j}$　学生 i の研究室 j に対する志望順位．1～9 の整数，（セル範囲 C4:T121).
  空白のセルは志望順位外を表す
- p_k　第 k 志望の研究室に対する志望度．（AS 列）
- U_j　研究室 j の定員（123 行目）
- $c_{i,j}$　学生 i を研究室 j に配属したときの志望度. 研究室配属問題（その 1）と同じ意味.
  $r_{i,j}$とp_kを組み合わせて，値を設定される.
  例えば，学生 i の研究室 j に対する志望順位が第 2 位の場合，$c_{i,j} = p_2 = 15$.
  学生 i の研究室 j に対する志望順位が空白の場合は，$c_{i,j} = 0$.

3.4 決定変数・制約条件・目的関数

配属する研究室の個数が 1 から 3 に増加した点を除けば，研究室配属問題（その 1）と同じ

3.5 Python プログラム（途中から）: 定数用データの作成

```
I = [i+1 for i in range(118)]
J = [i+1 for i in range(18)]
K = [i+1 for i in range(9)]
```

- 記入された志望順位を要素とするリスト K を定義する．K = [1, 2, ..., 9]

3.6 Python プログラム（途中から）：定数用データの作成

```
c = {}
for i in I:
    for j in J:
        if r[i, j] in K:
            c[i, j] = ???
        else:
            c[i, j] = ???
```

- $c_{i,j}$ の定義（前頁）をよく読んで，適切な値を設定する．

3.7 Python プログラム：制約条件の追加・目的関数の設定

（その１）との違いは，配属する研究室が 3 つになったことのみ

◎ 課題 4

研究室配属問題（その２）を実行するプログラムを作成しなさい．

Excel ファイル最適化 2-3.xlsx のシート「配属 2」から必要なデータを読み込み，求めた結果を同シート内に出力して，ファイル最適化 2-4.xlsx として保存すること．（最適値：5247）

4. 研究室配属問題（その３）

4.1 対象の問題（Excel シート「配属３」参照）

- 配属の期間を 3 つのサイクルに分ける．
- 118 人の学生を各サイクルに 1 つずつ，異なる 3 つの研究室に配属する．
- 学生はなるべく志望順位の高い研究室に配属する．
- 配属問題（その２）の条件に加え，各サイクルでどの研究室に配属するかまで決定する．
- 研究室には 1 サイクル当たりの定員が設定済み．
- いずれのサイクルでも定員を超える配属は禁止

4.2 定数（その２と異なる点のみ）

- U_j　　1 サイクル当たりの研究室 j の定員（配属人数の上限）．

4.3 決定変数

- $x_{i,j,t}$　　学生 i を研究室 j の第 t サイクル配属するなら 1，配属しないなら 0．

  例) 学生 1 を研究室 2 の第 3 サイクルに配属する場合，$x_{1,2,3} = 1$

  学生 3 を研究室 4 に配属しない場合，$x_{3,4,1} = x_{3,4,2} = x_{3,4,3} = 0$

4.4 制約条件

- それぞれの学生は，各サイクルにちょうど 1 つ，異なる 3 つの研究室に配属される．
- それぞれの研究室において，各サイクルに配属される人数は定員以内である．
- （決定変数は 0 か 1 しか取らない）

4.5 目的関数

配属された研究室に対する志望度の点数の合計　→　最大化

4.6 定式化のヒント

研究室配属問題（その２）と同じように，次の制約条件を考えたとする．

$$\sum_{j=1}^{18}\sum_{t=1}^{3} x_{i,j,t} = 3 \quad (i = 1, 2, \dots, 118)$$

この制約条件は「各学生を 3 回配属する」を表す．この制約条件と研究室定員に関する制約条件で最適解を求めると，正しい結果が得られない．なぜなら，志望度の高い第 1 志望の研究室に 3 回割り当てるからである．

例えば，学生 1 が研究室 2 を第 1 志望とした場合，$x_{1,2,1} = x_{1,2,2} = x_{1,2,3} = 1$としてしまい，3 サイクルとも第 1 志望の研究室 2 に割り当てられる．学生は異なる 3 つの研究室に割り当てる必要がある．したがって，「各学生は各サイクルに 1 つの研究室に配属される」制約条件と，「各学生は各研究室に 1 回までしか配属されない」制約条件が必要となる．

4.7 定式化

最？化　$\sum_{?=?}^{?}\sum_{?=?}^{?}\sum_{?=?}^{?} ?$

条件　$\sum_{?=?}^{?} ? \; ? \; ? \quad (i = 1, 2, \dots, 118,\ t = 1, 2, 3)$　（学生iの第tサイクルに対する条件）

$\sum_{?=?}^{?} ? \; ? \; ? \quad (j = 1, 2, \dots, 18,\ t = 1, 2, 3)$　（研究室jの第tサイクルに対する条件）

$\sum_{?=?}^{?} ? \; ? \; ? \quad (?= 1, \dots, ?,\ ?= 1, \dots, ?)$　（？？？）

$x_{ij} = 0 \ or \ 1 \quad (i = 1, 2, \dots, 118,\ j = 1, 2, \dots, 18)$

4.8 Python プログラム（途中から）：定数用データの作成

```
I = [i+1 for i in range(118)]
J = [i+1 for i in range(18)]
K = [i+1 for i in range(9)]
T = [i+1 for i in range(3)]
```

- サイクルの番号 1, 2, 3 を要素とするリスト T を定義する．T = [1, 2, 3]

4.9 Python プログラム（途中から）：決定変数の作成

```
for i in I:
    for j in J:
        for t in T:
            x[i, j, t] = LpVariable(f'x{i},{j},{t}', cat=LpBinary)
```

- 決定変数 $x_{i,j,t}$ の添え字が 3 種になったので，3 重の for ループになる．
  変数名の添字も 1 つ増える．

4.10 Python プログラム：制約条件の追加・目的関数の設定

- 決定変数 $x_{i,j,t}$ の添え字が 3 種になったので，（その２）と同じ意味の制約条件や目的関数でも修正が必要

◎ 課題 5

研究室配属問題（その３）を定式化し，配属を出力するプログラムを作成しなさい．

Excel ファイル最適化 2-4.xlsx のシート「配属 3」から必要なデータを読み込み，求めた結果を同シート内に出力して，ファイル最適化 2-5.xlsx として保存すること．（最適値：5247）

【余裕のある人向けのコメント】

課題 4 と課題 5 の結果を比較すると，最適値はどちらも 5247 で同じであるが，Excel ファイルで結果を確認すると，最適解，すなわち個々の学生が配属される研究室は少し異なることが分かる．

課題 4 に対して「各サイクルにおいて定員を守る」という条件を追加しているので，課題 5 の方が条件が厳しく，課題 5 の最適解は課題 4 の最適解でもある．したがって，課題 4 の最適解は少なくとも 2 つ以上存在することが分かる（「最適な配属方法」が何種類もある）．

また，課題 4 と課題 5 の最適解が異なった理由は，以下の 2 つが考えられる．

- （サイクルごとの割当は考慮していない）課題 4 の最適配属方法では「各サイクルにおいて定員を守る」ことができないので，課題 5 では別の配属が見つかった．
- 課題 4 の最適配属方法でも「各サイクルにおいて定員を守る」ように，サイクルごとに学生を割り当てることができるが，課題 5 ではたまたま別の配属が見つかった．

最適値が同じ最適解が複数ある場合，最適化ソルバーがどの最適解を出力するかは，残念ながら分からない．

◎ 課題 6

課題 5 の出力では，学生が受講する 3 つの研究室は分かるが，どのサイクルにどの研究室に配属されたのかが分からない，という問題がある．課題 5 のプログラムを修正して，各学生がどのサイクルにどの研究室に配属になったのかを出力するようにしよう．

Excel ファイル最適化 2-5.xlsx のシート「配属 3-2」から必要なデータを読み込み，各サイクルに配属される研究室番号を黄色のセル内（X：Z 列）に出力して，ファイル kada3-6.xlsx として保存すること．

結果の出力枠の大きさが減少したことに伴って，志望度 p_k，および定員U_j の入力位置が変わったので，プログラムも修正すること．（最適値：5247）

◎ 課題 7（挑戦問題）：研究室配属問題（その４）

課題 6 での研究室・サイクルごとの配属数を確認すると，うまく配属すれば，学生の志望度を下げることなく，研究室が開講するサイクル数を減らせることが分かる．例えば定員が 10 の研究室の場合, 各サイクルの人数が 7, 7, 6 ではなく 10, 10, 0 とすることで, 合計配属数を変えることなく，開講するサイクルを減らすことができる．

そこで，課題 6 の最適化で求めた最大の学生の志望度を保ったまま，研究室が開講するサイクル数をなるべく少なくする問題を考える．課題 6 の最適値（志望度合計の最大値）を V とおいて，学生の志望度合計が V であるという制約条件の下で, 研究室が開講するサイクル数を最小化する研究室配属問題（その４）を定式化し，最適解を求めるプログラムを作成しなさい．

課題 6 の結果から得た V の値である 5247 をプログラムに直接記入するのではなく，まずプログラムで課題 6 の最適化を実行して最適値 V の値を保持して，その後研究室配属問題（その４）の最適解を得ること（1 つのプログラムで 2 つの最適化問題を解く）．

Excel ファイル最適化 2-6.xlsx のシート「配属 3-3」から必要なデータを読み込み，求めた結果を同シート内に出力して，ファイル最適化 2-7.xlsx として保存すること．（2 回目の最適値：44）

【注意１】

2 回目の最適化問題は，これまでより多くの実行時間がかかります（これまでの数～10 倍程度）．実行前に保存することを推奨します．

プログラムを記入したセルの左側が In[*] となっているときは，プログラム実行中です．しばらく待ちましょう．

実行を中断するには，Run ボタン右の停止ボタン（Interrupt the kernel）を押す．それでも止まらない時は，Kernel メニュー内の Restart，または Shutdown で強制終了しましょう（プログラム実行前に保存していないと，プログラムは残っていないかもしれません）．

【注意２】

一つのプログラムの中で複数回目的関数の設定をすると警告（warning）が表示されるが，今回のように意図的に行っている場合は，気にする必要はない．

以上

【ヒント】（課題が解けず，ヒントが欲しい人向け）

◎ 課題 1

エラーが出てプログラムが動かない人は，まずプログラムが動くようにデバッグしましょう．プログラムの入出力ファイルである Excel ファイルが開いたままだと，PermissionError というファイルアクセス時のエラーが出るので，Excel ファイルを閉じてから実行しましょう．

エラーは出ないけど，最適解が 48 にならない人は，間違っている箇所を特定しましょう．モデルを作った後で print(model) を実行すれば，作成した最適化モデルを確認できます．最適化を実行した後で value(x[i, j]) を表示するプログラムを追加すれば，最適解の値を確認することができます．

◎ 課題 2

条件 if type(v) is int を満たすのは，変数 v に整数値が入っているときです．そのときには，その整数値を c[i, j] に代入します．そうでないときには，代わりに 0 を代入します．

◎ 課題 3

配属人数の下限 L_j を読み込む処理が必要です．Excel ファイルを開いて，何行目から読み込むのか確認しましょう．

◎ 課題 4

- 学生 i の研究室 j に対する志望順位 $r_{i,j}$ が 1～9 の整数のとき，言い換えると r_{ij} が K = [1, 2, ..., 9] のいずれかのときは，$c_{i,j}$ はその順位に応じた志望度（点数）になります．志望順位が第 k 位の研究室の志望度はp_kなので，c[i, j]は r と p を組み合わせて表現できます．
- $r_{i,j}$ が 1～9 の整数でないときは？

◎ 課題 5

- 決定変数が$x_{i,j}$から $x_{i,j,t}$ になったので，プログラムも x[i, j]から x[i, j, t]に変更しよう．
- 1 種類の制約条件に対して，必ず i, j, t の 3 種の変数に関する for ループが存在します．
  lpSum の内側に入る for ループは， Σ の添え字に相当します．
  lpSum の外側に位置する for ループは，制約条件式の後にある(i = 1, 2, ..., 118)に相当します．
- 3 番目の制約条件によって，学生が同じ研究室に複数回配属されないようにします．

◎ 課題 6

最適化問題は課題 5 から変更なし．志望度 p_kおよび定員U_j の読込位置を変更し，結果の出力について出力位置と出力内容を変更します．

◎ 課題 7

- 2 回目の最適化では，新たに以下の決定変数を追加する方法が良いでしょう．
  - $y_{j,t}$　　研究室 j の第 t サイクルを開講するなら 1，開講しないなら 0

- 次の制約条件を追加します.
  - 配属された研究室に対する志望度の点数の合計が V に等しい.
- 制約条件「各研究室において，各サイクルに配属される人数は定員以内である」を「それぞれの研究室において，各サイクルに配属される人数は，開講すれば定員以内，開講しなければ 0 である」に変更します．この条件を，2 種類の決定変数 x_{ijt} と y_{it} を用いて表しましょう．具体的には，次のように考えます.
  - ✧ 研究室 j の第 t サイクルを開講する（$y_{i,t} = 1$）なら，最大で U_j 人まで配属
  - ✧ 研究室 j の第 t サイクルを開講しない（$y_{i,t} = 0$）なら，最大で 0 人まで配属
  - ✧ 上の 2 つをまとめると，研究室 j へ第 t サイクルに配属する学生数の上限が，$y_{i,t}$（と U_j）に関する 1 つの不等式で表現できる（前回課題 11 のトラックの容量と同じ考え方）.
- 2 回目の最適化における目的関数は，開講サイクル数の最小化．$y_{j,t}$ で表現します.
  最小化問題に変わるので，model.sense = LpMinimize の一行が必要です.

【大ヒント】（【ヒント】を見ても，どうしても課題が解けない人向け！）

◎ 課題 1

輸送問題との違いは，決定変数の型と目的関数の向き（最大化・最小化）です．指定方法を忘れた人は，第 2 回の資料，または先に今回の課題 2 の解説を読みましょう．

◎ 課題 2

各学生が配属される研究室の数は必ず 1 です．

各研究室に配属される学生の数は U[j] までです．

◎ 課題 3

課題 2 の定式化との違いは，下限に関する制約条件が追加されるだけです．

◎ 課題 4

例えば r[i, j] = 2 は学生 i の研究室 j に対する志望順位が第 2 位であることを表します．この場合の志望度 c[i, j] は第 2 志望の研究室に対する志望度なので，c[i, j] = p[2] = 15 となります．r[i, j] = 2 と c[i, j] = p[2] を組み合わせれば，他の志望順位でも表現できるようになります．

最後に，課題 3 とは違い，配属研究室の数が 1 つではなく 3 つであることに注意して下さい．

◎ 課題 5

- 制約条件 1 つ目：
  各学生 i と各サイクル t に関する条件（外側の for ループ）で，「各学生 i は各サイクル t で 1 つの研究室に配属される」＝「各学生 i が各サイクル t で配属される研究室の数の合計（内側の for ループは研究室 j）は 1 に等しい」
- 制約条件 2 つ目：
  各研究室 j と各サイクル t に関する条件（外側の for ループ）で，「各研究室 j は各サイクル t で定員までしか配属を受け入れない」＝「各研究室 j に各サイクル t で配属される学生の数の合計（内側の for ループは学生 i）は研究室の定員 U[j]以下」
- 制約条件 3 つ目：
  各学生 i と各研究室 j に関する条件（外側の for ループ）で，「各学生 i は各研究室 j に複数回配属されない」＝「各学生 i が各研究室 j に配属される回数の合計（内側の for ループはサイクル t）は 1 回以下」

◎ 課題 6

出力するのは，研究室番号なので j．出力枠は，縦方向に学生，横方向にサイクルが並ぶので，行（row）は学生番号 i によって変化し，列（column）はサイクル番号 t によって変化します．