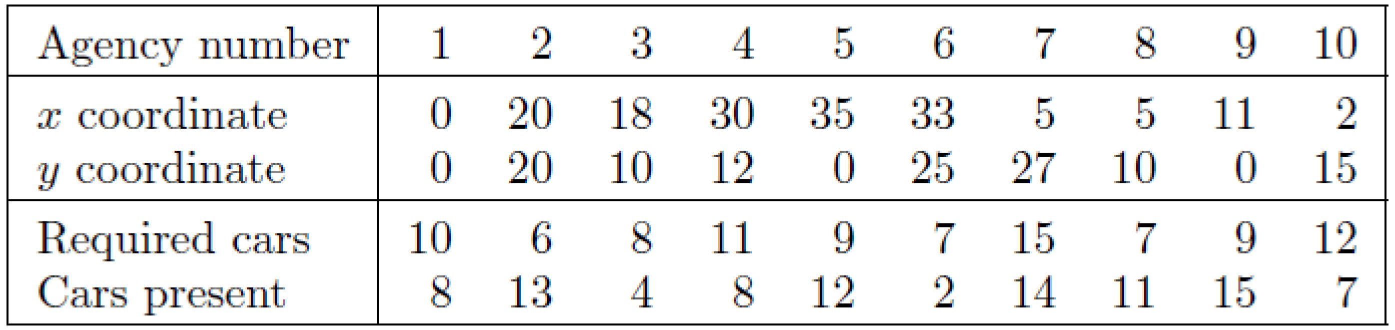
2. Car Rental [10 pts]

by Roumen Guha, on Sunday, February 19th, 2017

A small car rental company has a fleet of 94 vehicles distributed among its 10 agencies. The location of every agency is given by its geographical coordinates x and y in a grid based on miles. We assume that the road distance between agencies is approximately 1.3 times the Euclidean distance (as the crow flies). The following table indicates the coordinates of all agencies, the number of cars required the next morning, and the stock of cars in the evening preceding this day.

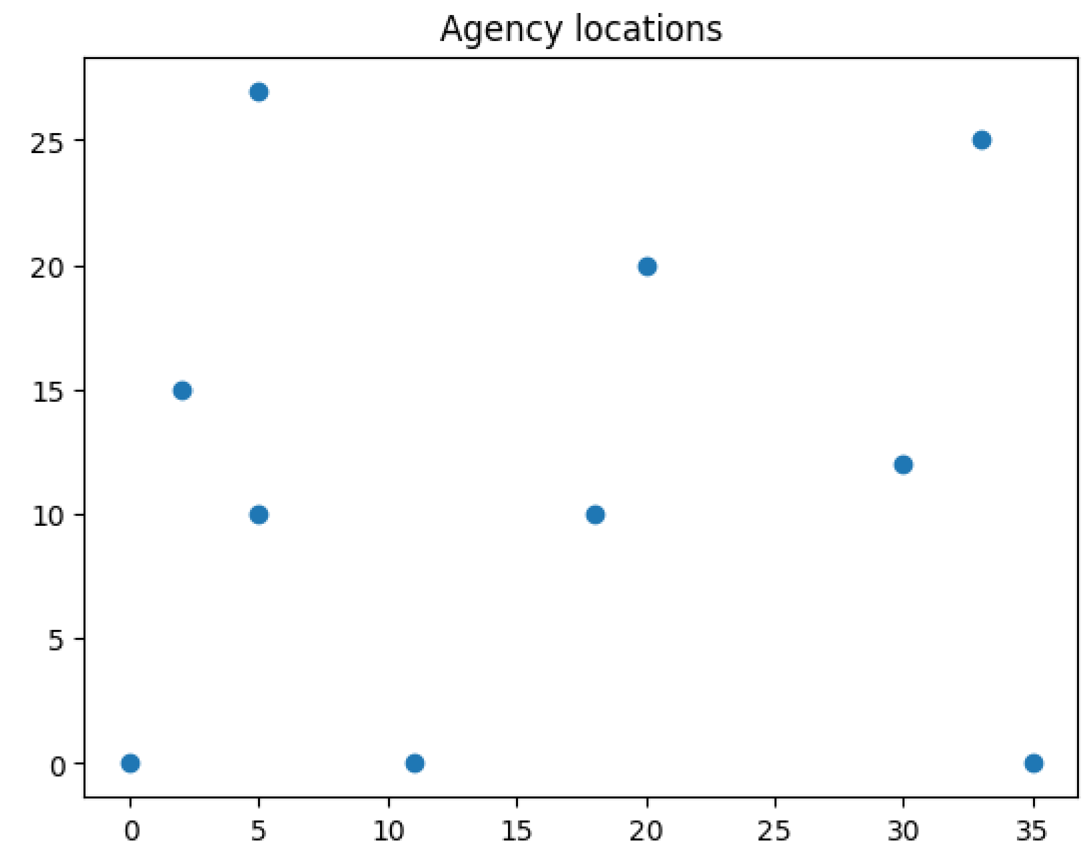


Agency number	1	2	3	4	5	6	7	8	9	10
x coordinate	0	20	18	30	35	33	5	5	11	2
y coordinate	0	20	10	12	0	25	27	10	0	15
Required cars	10	6	8	11	9	7	15	7	9	12
Cars present	8	13	4	8	12	2	14	11	15	7

Supposing the cost for transporting a car is $0.50 per mile, determine the movements of cars that allow the company to re-establish the required numbers of cars at all agencies, minimizing the total cost incurred for transport.

In [6]:

```
using NamedArrays, PyPlot

agencies = [1:10;]

coords =  [ 0 20 18 30 35 33  5  5 11  2
            0 20 10 12  0 25 27 10  0 15 ]

carsReq = [10  6  8 11  9  7 15  7  9 12]

carsPres = [8 13  4  8 12  2 14 11 15  7]

distMultiplier = 1.3
mileCost       = 0.5 # dollars
;

scatter(coords[1,:], coords[2,:])
title("Agency locations")
```



Out[6]:

```
PyObject <matplotlib.text.Text object at 0x0000000028ED4438>
```

In [101]:

```
displacements = zeros(10,10)

for source = 1:10
    for destination = 1:10
        # displacement = sqrt((x2 - x1)^2 + (y2 - y1)^2)
        displacements[source,destination] =  sqrt((coords[1,destination] - coords[1,source])^2 + (coords[2,destination] - coords[2,source])^2)
    end
end

distances = displacements * distMultiplier
costs = distances * mileCost
;
```

In [102]:

```
using JuMP

m = Model()

# movements[i,j] is the number of cars shipped from agency i to agency j
@variable(m, movements[agencies, agencies] >= 0)

# each agency must have as many cars as they require the next moring
@constraint(m, present[j in agencies], sum(movements[i,j] for i in agencies) == carsReq[j])
@constraint(m, required[i in agencies], sum(movements[i,j] for j in agencies) == carsPres[i])

@objective(m, Min, sum(movements[i,j]*costs[i,j] for i in agencies, j in agencies))

status = solve(m)
```

Out[102]:

```
:Optimal
```

In [103]:

```
raw = Int[getvalue(movements[i, j]) for i in agencies, j in agencies]
solution = NamedArray(raw, (agencies, agencies), ("from", "to"))
```

Out[103]:

```
10×10 Named Array{Int64,2}
from ╲ to │  1   2   3   4   5   6   7   8   9  10
──────────┼───────────────────────────────────────
1         │  8   0   0   0   0   0   0   0   0   0
2         │  0   6   1   0   0   5   1   0   0   0
3         │  0   0   4   0   0   0   0   0   0   0
4         │  0   0   0   8   0   0   0   0   0   0
5         │  0   0   0   3   9   0   0   0   0   0
6         │  0   0   0   0   0   2   0   0   0   0
7         │  0   0   0   0   0   0  14   0   0   0
8         │  0   0   0   0   0   0   0   6   0   5
9         │  2   0   3   0   0   0   0   1   9   0
10        │  0   0   0   0   0   0   0   0   0   7
```

In [105]:

```
for j = 1:10
    for i = 1:10
        if movement[i,j] >= 1e-5
            if i != j
                print(Int(movement[i,j]))
                println(" car(s) moved to agency ", j, " (from agency ", i, ")")
            end
        end
    end
end

println()
println("The total cost will be \$", getobjectivevalue(m))
```

```
2 car(s) moved to agency 1 (from agency 9)
1 car(s) moved to agency 3 (from agency 2)
3 car(s) moved to agency 3 (from agency 9)
3 car(s) moved to agency 4 (from agency 5)
5 car(s) moved to agency 6 (from agency 2)
1 car(s) moved to agency 7 (from agency 2)
1 car(s) moved to agency 8 (from agency 9)
5 car(s) moved to agency 10 (from agency 8)

The total cost will be $152.63901632295628
```

In [106]:

```
println(m)
```

```
Min 18.38477631085024 movements[1,2] + 13.384319183283102 movements[1,3] + 21.002142747824564 movements[1,4] + 22.75 movements[1,5] + 26.91031400782979 movements[1,6] + 17.84838928
306978 movements[1,7] + 7.267220926874317 movements[1,8] + 7.15 movements[1,9] + 9.836284867774012 movements[1,10] + 18.38477631085024 movements[2,1] + 6.62872536767062 movements
[2,3] + 8.324061508662703 movements[2,4] + 16.25 movements[2,5] + 9.053452380169677 movements[2,6] + 10.759414482210452 movements[2,7] + 11.718041645257966 movements[2,8] + 14.2556
12929649851 movements[2,9] + 12.143002099975114 movements[2,10] + 13.384319183283102 movements[3,1] + 6.62872536767062 movements[3,2] + 7.907591289387685 movements[3,4] + 12.820003
900155413 movements[3,5] + 13.788582233137678 movements[3,6] + 13.910607463371253 movements[3,7] + 8.450000000000001 movements[3,8] + 7.934261150226907 movements[3,9] + 10.89598549
9256138 movements[3,10] + 21.002142747824564 movements[4,1] + 8.324061508662703 movements[4,2] + 7.907591289387685 movements[4,3] + 8.450000000000001 movements[4,5] + 8.67208164168
2118 movements[4,6] + 18.950593658247225 movements[4,7] + 16.30191706517979 movements[4,8] + 14.606933285258751 movements[4,9] + 18.304166192427342 movements[4,10] + 22.75 movement
s[5,1] + 16.25 movements[5,2] + 12.820003900155413 movements[5,3] + 8.450000000000001 movements[5,4] + 16.30191706517979 movements[5,6] + 26.234566891793737 movements[5,7] + 20.554
804791094465 movements[5,8] + 15.600000000000001 movements[5,9] + 23.561939648509416 movements[5,10] + 26.91031400782979 movements[6,1] + 9.053452380169677 movements[6,2] + 13.7885
82233137678 movements[6,3] + 8.672081641682118 movements[6,4] + 16.30191706517979 movements[6,5] + 18.246369501903658 movements[6,7] + 20.647094226549168 movements[6,8] + 21.646073
546950728 movements[6,9] + 21.17244671737303 movements[6,10] + 17.84838928306978 movements[7,1] + 10.759414482210452 movements[7,2] + 13.910607463371253 movements[7,3] + 18.9505936
58247225 movements[7,4] + 26.234566891793737 movements[7,5] + 18.246369501903658 movements[7,6] + 11.05 movements[7,8] + 17.978111691721132 movements[7,9] + 8.040055969954437 movem
ents[7,10] + 7.267220926874317 movements[8,1] + 11.718041645257966 movements[8,2] + 8.450000000000001 movements[8,3] + 16.30191706517979 movements[8,4] + 20.554804791094465 movemen
ts[8,5] + 20.647094226549168 movements[8,6] + 11.05 movements[8,7] + 7.580237463298891 movements[8,9] + 3.7901187316494456 movements[8,10] + 7.15 movements[9,1] + 14.2556129296498
1 movements[9,2] + 7.934261150226907 movements[9,3] + 14.606933285258751 movements[9,4] + 15.600000000000001 movements[9,5] + 21.646073546950728 movements[9,6] + 17.978111691721132
movements[9,7] + 7.580237463298891 movements[9,8] + 11.370356194948336 movements[9,10] + 9.836284867774012 movements[10,1] + 12.143002099975114 movements[10,2] + 10.895985499256138
movements[10,3] + 18.304166192427342 movements[10,4] + 23.561939648509416 movements[10,5] + 21.17244671737303 movements[10,6] + 8.040055969954437 movements[10,7] + 3.79011873164944
56 movements[10,8] + 11.370356194948336 movements[10,9]
Subject to
 movements[1,1] + movements[2,1] + movements[3,1] + movements[4,1] + movements[5,1] + movements[6,1] + movements[7,1] + movements[8,1] + movements[9,1] + movements[10,1] == 10
 movements[1,2] + movements[2,2] + movements[3,2] + movements[4,2] + movements[5,2] + movements[6,2] + movements[7,2] + movements[8,2] + movements[9,2] + movements[10,2] == 6
 movements[1,3] + movements[2,3] + movements[3,3] + movements[4,3] + movements[5,3] + movements[6,3] + movements[7,3] + movements[8,3] + movements[9,3] + movements[10,3] == 8
 movements[1,4] + movements[2,4] + movements[3,4] + movements[4,4] + movements[5,4] + movements[6,4] + movements[7,4] + movements[8,4] + movements[9,4] + movements[10,4] == 11
 movements[1,5] + movements[2,5] + movements[3,5] + movements[4,5] + movements[5,5] + movements[6,5] + movements[7,5] + movements[8,5] + movements[9,5] + movements[10,5] == 9
 movements[1,6] + movements[2,6] + movements[3,6] + movements[4,6] + movements[5,6] + movements[6,6] + movements[7,6] + movements[8,6] + movements[9,6] + movements[10,6] == 7
 movements[1,7] + movements[2,7] + movements[3,7] + movements[4,7] + movements[5,7] + movements[6,7] + movements[7,7] + movements[8,7] + movements[9,7] + movements[10,7] == 15
 movements[1,8] + movements[2,8] + movements[3,8] + movements[4,8] + movements[5,8] + movements[6,8] + movements[7,8] + movements[8,8] + movements[9,8] + movements[10,8] == 7
 movements[1,9] + movements[2,9] + movements[3,9] + movements[4,9] + movements[5,9] + movements[6,9] + movements[7,9] + movements[8,9] + movements[9,9] + movements[10,9] == 9
 movements[1,10] + movements[2,10] + movements[3,10] + movements[4,10] + movements[5,10] + movements[6,10] + movements[7,10] + movements[8,10] + movements[9,10] + movements[10,10]
 == 12
 movements[1,1] + movements[1,2] + movements[1,3] + movements[1,4] + movements[1,5] + movements[1,6] + movements[1,7] + movements[1,8] + movements[1,9] + movements[1,10] == 8
 movements[2,1] + movements[2,2] + movements[2,3] + movements[2,4] + movements[2,5] + movements[2,6] + movements[2,7] + movements[2,8] + movements[2,9] + movements[2,10] == 13
 movements[3,1] + movements[3,2] + movements[3,3] + movements[3,4] + movements[3,5] + movements[3,6] + movements[3,7] + movements[3,8] + movements[3,9] + movements[3,10] == 4
 movements[4,1] + movements[4,2] + movements[4,3] + movements[4,4] + movements[4,5] + movements[4,6] + movements[4,7] + movements[4,8] + movements[4,9] + movements[4,10] == 8
 movements[5,1] + movements[5,2] + movements[5,3] + movements[5,4] + movements[5,5] + movements[5,6] + movements[5,7] + movements[5,8] + movements[5,9] + movements[5,10] == 12
 movements[6,1] + movements[6,2] + movements[6,3] + movements[6,4] + movements[6,5] + movements[6,6] + movements[6,7] + movements[6,8] + movements[6,9] + movements[6,10] == 2
 movements[7,1] + movements[7,2] + movements[7,3] + movements[7,4] + movements[7,5] + movements[7,6] + movements[7,7] + movements[7,8] + movements[7,9] + movements[7,10] == 14
 movements[8,1] + movements[8,2] + movements[8,3] + movements[8,4] + movements[8,5] + movements[8,6] + movements[8,7] + movements[8,8] + movements[8,9] + movements[8,10] == 11
 movements[9,1] + movements[9,2] + movements[9,3] + movements[9,4] + movements[9,5] + movements[9,6] + movements[9,7] + movements[9,8] + movements[9,9] + movements[9,10] == 15
 movements[10,1] + movements[10,2] + movements[10,3] + movements[10,4] + movements[10,5] + movements[10,6] + movements[10,7] + movements[10,8] + movements[10,9] + movements[10,10]
 == 7
 movements[i,j] >= 0 for all i in {1,2,..,9,10}, j in {1,2,..,9,10}
```