An Optimal Breakfast

ECE367 PS05 Problem 5.5 -- Aman Bhargava

There are $n = 3$ food types, each with certain nutritional characteristics.

GOAL: Find optimal composition of breakfast as follows:

- Minimizing cost ($c^T = [0.15, 0.25, 0.05]$)
- Number of calories between 2000 and 2250 ($[70, 121, 65]x < 2250$, $[70, 121, 65]x > 2000$)
- Vitamin between 5000 and 10000 ($[107, 500, 0]x < 10000$, $[107, 500, 0]x > 5000$)
- Sugar no larger than 1000 ($[45, 40, 60]x < 1000$)
- Maximum number of servings total is 10 ($x_i < 10$ for $i \in [3]$).

Results

As calculated below:

$p^* = 3.7411764705882358$

$$x^* = \begin{bmatrix} 6.58823529411765 \\ 10.0 \\ 5.058823529411761 \end{bmatrix}$$

In [1]:
```
using JuMP
import GLPK
```

In [2]:
```
model = Model(GLPK.Optimizer)

@variable(model, 0 <= x1 <= 10)
@variable(model, 0 <= x2 <= 10)
@variable(model, 0 <= x3 <= 10)

@objective(model, Min, 0.15x1 + 0.25x2 + 0.05x3)

@constraint(model, 70x1+121x2+65x3 <= 2250)
@constraint(model, 70x1+121x2+65x3 >= 2000)

@constraint(model, 107x1+500x2 <= 10000)
@constraint(model, 107x1+500x2 >= 5000)

@constraint(model, 45x1+40x2+60x3 <= 1000)

print(model)
```

```
Min 0.15 x1 + 0.25 x2 + 0.05 x3
Subject to
 70 x1 + 121 x2 + 65 x3 ≥ 2000.0
 107 x1 + 500 x2 ≥ 5000.0
 70 x1 + 121 x2 + 65 x3 ≤ 2250.0
```

```
107 x1 + 500 x2 ≤ 10000.0
45 x1 + 40 x2 + 60 x3 ≤ 1000.0
x1 ≥ 0.0
x2 ≥ 0.0
x3 ≥ 0.0
x1 ≤ 10.0
x2 ≤ 10.0
x3 ≤ 10.0
```

In [3]:
```
optimize!(model)
println("Termination status : ", termination_status(model))
println("Primal status      : ", primal_status(model))
```

```
Termination status : OPTIMAL
Primal status      : FEASIBLE_POINT
```

In [4]:
```
obj_value = objective_value(model)
corn = value(x1)
milk = value(x2)
bread = value(x3)

println("Objective value : ", obj_value)
println("corn            : ", corn)
println("milk            : ", milk)
println("bread           : ", bread)
```

```
Objective value : 3.7411764705882358
corn            : 6.58823529411765
milk            : 10.0
bread           : 5.058823529411761
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: