8.2.1. Линейное программирование

In [1]:
```
# Подключение пакетов:
import Pkg
Pkg.add("JuMP")
Pkg.add("GLPK")
using JuMP
using GLPK
```

```
   Updating registry at `C:\Users\Admin\.julia\registries\General`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
```

In [2]:
```
# Определение объекта модели с именем model:
model = Model(GLPK.Optimizer)
```

Out[2]:

$$\begin{aligned} & \text{feasibility} \\ & \text{Subject to} \end{aligned}$$

In [3]:
```
# Определение переменных x, y и граничных условий для них:
@variable(model, x >= 0)
@variable(model, y >= 0)
```

Out[3]:

$$y$$

In [4]:
```
# Определение ограничений модели:
@constraint(model, 6x + 8y >= 100)
@constraint(model, 7x + 12y >= 120)
```

Out[4]:

$$7x + 12y \geq 120.0$$

In [5]:
```
# Определение целевой функции:
@objective(model, Min, 12x + 20y)
```

Out[5]:

$$12x + 20y$$

In [6]:
```
# Вызов функции оптимизации:
optimize!(model)
```

In [7]:
```
# Определение причины завершения работы оптимизатора:
termination_status(model)
```

Out[7]:
```
OPTIMAL::TerminationStatusCode = 1
```

In [8]:
```
# Демонстрация первичных результирующих значений переменных x и y:
@show value(x);
@show value(y);
```

```
value(x) = 14.999999999999993
value(y) = 1.2500000000000047
```

In [9]:
```
# Демонстрация результата оптимизации:
@show objective_value(model);
```

```
objective_value(model) = 205.0
```

8.2.2. Векторизованные ограничения и целевая функция оптимизации

In [10]:
```
# Подключение пакетов:
import Pkg
Pkg.add("JuMP")
Pkg.add("GLPK")
using JuMP
using GLPK
```

```
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
```

In [11]:
```
# Определение объекта модели с именем vector_model:
vector_model = Model(GLPK.Optimizer)
```

Out[11]:

$$\begin{aligned} & \text{feasibility} \\ & \text{Subject to} \end{aligned}$$

In [12]:
```
# Определение начальных данных:
A= [ 1 1 9 5;
3 5 0 8;
2 0 6 13]
b = [7; 3; 5]
c = [1; 3; 5; 2]
```

Out[12]:
```
4-element Array{Int64,1}:
 1
 3
 5
 2
```

In [13]:
```
# Определение вектора переменных:
@variable(vector_model, x[1:4] >= 0)
```

Out[13]:
```
4-element Array{VariableRef,1}:
 x[1]
 x[2]
 x[3]
 x[4]
```

In [14]:
```
# Определение ограничений модели:
```

```
@constraint(vector_model, A * x .== b)
```

Out[14]:
```
3-element Array{ConstraintRef{Model,MathOptInterface.ConstraintIndex{MathOptInterface.ScalarAffineFunction{Float64},MathOptInterface.EqualTo{Float64}},ScalarShape},1}:
 x[1] + x[2] + 9 x[3] + 5 x[4] == 7.0
 3 x[1] + 5 x[2] + 8 x[4] == 3.0
 2 x[1] + 6 x[3] + 13 x[4] == 5.0
```

In [15]:
```
# Определение целевой функции:
@objective(vector_model, Min, c' * x)
```

Out[15]:

$$x_1 + 3x_2 + 5x_3 + 2x_4$$

In [16]:
```
# Вызов функции оптимизации:
optimize!(vector_model)
```

In [17]:
```
# Определение причины завершения работы оптимизатора:
termination_status(vector_model)
```

Out[17]:
```
OPTIMAL::TerminationStatusCode = 1
```

In [18]:
```
# Демонстрация результата оптимизации:
@show objective_value(vector_model);
```

```
objective_value(vector_model) = 4.9230769230769225
```

8.2.3. Оптимизация рациона питания

In [19]:
```
# Подключение пакетов:
import Pkg
Pkg.add("JuMP")
Pkg.add("GLPK")
using JuMP
using GLPK
```

```
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
```

In [20]:
```
# Контейнер для хранения данных об ограничениях на количество  потребляемых калори
category_data = JuMP.Containers.DenseAxisArray(
    [1800 2200;
91 Inf;
0 65;
0 1779],
["calories", "protein", "fat", "sodium"],
["min", "max"])
```

Out[20]:
```
2-dimensional DenseAxisArray{Float64,2,...} with index sets:
    Dimension 1, ["calories", "protein", "fat", "sodium"]
    Dimension 2, ["min", "max"]
And data, a 4×2 Array{Float64,2}:
 1800.0  2200.0
   91.0    Inf
    0.0    65.0
    0.0  1779.0
```

In [21]:
```
# массив данных с наименованиями продуктов:
foods = ["hamburger", "chicken", "hot dog", "fries", "macaroni",
 "pizza","salad", "milk", "ice cream"]
```

Out[21]:
```
9-element Array{String,1}:
 "hamburger"
 "chicken"
 "hot dog"
 "fries"
 "macaroni"
 "pizza"
 "salad"
 "milk"
 "ice cream"
```

In [22]:
```
# Массив стоимости продуктов:
cost = JuMP.Containers.DenseAxisArray(
[2.49, 2.89, 1.50, 1.89, 2.09, 1.99, 2.49, 0.89, 1.59],
foods)
```

Out[22]:
```
1-dimensional DenseAxisArray{Float64,1,...} with index sets:
    Dimension 1, ["hamburger", "chicken", "hot dog", "fries", "macaroni", "pizza",
"salad", "milk", "ice cream"]
And data, a 9-element Array{Float64,1}:
 2.49
 2.89
 1.5
 1.89
 2.09
 1.99
 2.49
 0.89
 1.59
```

In [23]:
```
# Массив данных о содержании калорий, белков, жиров и соли в продуктах питания:
food_data = JuMP.Containers.DenseAxisArray(
[410 24 26 730;
420 32 10 1190;
560 20 32 1800;
380 4 19 270;
320 12 10 930;
320 15 12 820;
320 31 12 1230;
100 8 2.5 125;
330 8 10 180],
foods,
["calories", "protein", "fat", "sodium"])
```

Out[23]:
```
2-dimensional DenseAxisArray{Float64,2,...} with index sets:
    Dimension 1, ["hamburger", "chicken", "hot dog", "fries", "macaroni", "pizza",
"salad", "milk", "ice cream"]
    Dimension 2, ["calories", "protein", "fat", "sodium"]
And data, a 9×4 Array{Float64,2}:
 410.0  24.0  26.0   730.0
 420.0  32.0  10.0  1190.0
 560.0  20.0  32.0  1800.0
 380.0   4.0  19.0   270.0
 320.0  12.0  10.0   930.0
 320.0  15.0  12.0   820.0
 320.0  31.0  12.0  1230.0
 100.0   8.0   2.5   125.0
 330.0   8.0  10.0   180.0
```

In [24]:
```
# Определение объекта модели с именем model:
model = Model(GLPK.Optimizer)
```

Out[24]:

$$\begin{aligned} & \text{feasibility} \\ & \text{Subject to} \end{aligned}$$

In [25]:
```
# Определим массив:
categories = ["calories", "protein", "fat", "sodium"]
```

Out[25]:
```
4-element Array{String,1}:
 "calories"
 "protein"
 "fat"
 "sodium"
```

In [26]:
```
# Определение переменных:
@variables(model, begin
category_data[c, "min"] <= nutrition[c = categories] <= category_data[c, "max"]
# Сколько покупать продуктов:
buy[foods] >= 0
end)
```

In [27]:
```
# Определение целевой функции:
@objective(model, Min, sum(cost[f] * buy[f] for f in foods))
```

Out[27]:

$$2.49buy_{hamburger} + 2.89buy_{chicken} + 1.5buy_{hotdog} + 1.89buy_{fries} + 2.09buy_{macaroni} + 1.99bu$$

In [28]:
```
# Определение ограничений модели:
@constraint(model, [c in categories],
sum(food_data[f, c] * buy[f] for f in foods) == nutrition[c])
```

Out[28]:
```
1-dimensional DenseAxisArray{ConstraintRef{Model,MathOptInterface.ConstraintIndex
{MathOptInterface.ScalarAffineFunction{Float64},MathOptInterface.EqualTo{Float6
4}},ScalarShape},1,...} with index sets:
    Dimension 1, ["calories", "protein", "fat", "sodium"]
And data, a 4-element Array{ConstraintRef{Model,MathOptInterface.ConstraintIndex{M
athOptInterface.ScalarAffineFunction{Float64},MathOptInterface.EqualTo{Float64}},S
calarShape},1}:
 -nutrition[calories] + 410 buy[hamburger] + 420 buy[chicken] + 560 buy[hot dog] +
380 buy[fries] + 320 buy[macaroni] + 320 buy[pizza] + 320 buy[salad] + 100 buy[mil
k] + 330 buy[ice cream] == 0.0
 -nutrition[protein] + 24 buy[hamburger] + 32 buy[chicken] + 20 buy[hot dog] + 4 b
uy[fries] + 12 buy[macaroni] + 15 buy[pizza] + 31 buy[salad] + 8 buy[milk] + 8 buy
[ice cream] == 0.0
 -nutrition[fat] + 26 buy[hamburger] + 10 buy[chicken] + 32 buy[hot dog] + 19 buy
[fries] + 10 buy[macaroni] + 12 buy[pizza] + 12 buy[salad] + 2.5 buy[milk] + 10 bu
y[ice cream] == 0.0
 -nutrition[sodium] + 730 buy[hamburger] + 1190 buy[chicken] + 1800 buy[hot dog] +
270 buy[fries] + 930 buy[macaroni] + 820 buy[pizza] + 1230 buy[salad] + 125 buy[mi
lk] + 180 buy[ice cream] == 0.0
```

In [29]:
```
# Вызов функции оптимизации:
JuMP.optimize!(model)
term_status = JuMP.termination_status(model)
```

Out[29]:
```
OPTIMAL::TerminationStatusCode = 1
```

In [30]:
```
#Для просмотра результата решения модно вывести значение переменной buy:
hcat(buy.data,JuMP.value.(buy.data))
```

Out[30]:
```
9×2 Array{GenericAffExpr{Float64,VariableRef},2}:
 buy[hamburger]  0.6045138888888888
 buy[chicken]    0
 buy[hot dog]    0
 buy[fries]      0
 buy[macaroni]   0
 buy[pizza]      0
 buy[salad]      0
 buy[milk]       6.9701388888888935
 buy[ice cream]  2.591319444444441
```

8.2.4. Путешествие по миру

In [31]:
```
# Скачиваем данные с ресурса на git:
```

In [32]:
```
# Подключение пакетов:
import Pkg
Pkg.add("DelimitedFiles")
Pkg.add("CSV")
using DelimitedFiles
using CSV
```

```
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
```

In [33]:
```
# Считывание данных:
```

```
passportdata = readdlm("passport-index-matrix.csv",',')
```

Out[33]:

```
200×200 Array{Any,2}:
 "Passport"                "Albania"          …    "Afghanistan"
 "Afghanistan"             "visa required"         -1
 "Albania"                 -1                      "visa required"
 "Algeria"                 "visa required"         "visa required"
 "Andorra"                 90                      "visa required"
 "Angola"                  "visa required"    …    "visa required"
 "Antigua and Barbuda"     90                      "visa required"
 "Argentina"               90                      "visa required"
 "Armenia"                 90                      "visa required"
 "Australia"               90                      "visa required"
 "Austria"                 90                 …    "visa required"
 "Azerbaijan"              90                      "visa required"
 "Bahamas"                 90                      "visa required"
 ⋮                                            ⋱
 "United Arab Emirates"    90                      "visa required"
 "United Kingdom"          90                      "visa required"
 "United States"           90                 …    "visa required"
 "Uruguay"                 90                      "visa required"
 "Uzbekistan"              "visa required"         "visa required"
 "Vanuatu"                 "visa required"         "visa required"
 "Vatican"                 90                      "visa required"
 "Venezuela"               90                 …    "visa required"
 "Vietnam"                 "visa required"         "visa required"
 "Yemen"                   "visa required"         "visa required"
 "Zambia"                  "visa required"         "visa required"
 "Zimbabwe"                "visa required"         "visa required"
```

In [34]:

```
# Подключение пакетов:
Pkg.add("JuMP")
Pkg.add("GLPK")
using JuMP
using GLPK
```

```
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
```

In [35]:

```
# Задаём переменные:
cntr = passportdata[2:end,1]
vf = (x -> typeof(x)==Int64 || x == "VF" || x == "VOA" ? 1 : 0).(passportdata[2:en
```

In [36]:

```
# Определение объекта модели с именем model:
model = Model(GLPK.Optimizer)
```

Out[36]:

feasibility

Subject to

In [37]:

```
# Переменные, ограничения и целевая функция:
@variable(model, pass[1:length(cntr)], Bin)
@constraint(model, [j=1:length(cntr)], sum( vf[i,j]*pass[i] for i in 1:length(cntr
@objective(model, Min, sum(pass))
```

Out[37]:

$$pass_1 + pass_2 + pass_3 + pass_4 + pass_5 + pass_6 + pass_7 + pass_8 + pass_9 + pass_{10} + pa$$
$$+ pass_{25} + pass_{26} + pass_{27} + pass_{28} + pass_{29} + pass_{30} + pass_{31} + pass_{32} + pass_{33} + pa$$
$$+ pass_{48} + pass_{49} + pass_{50} + pass_{51} + pass_{52} + pass_{53} + pass_{54} + pass_{55} + pass_{56} + pa$$
$$+ pass_{71} + pass_{72} + pass_{73} + pass_{74} + pass_{75} + pass_{76} + pass_{77} + pass_{78} + pass_{79} + pa$$
$$+ pass_{94} + pass_{95} + pass_{96} + pass_{97} + pass_{98} + pass_{99} + pass_{100} + pass_{101} + pass_{10}$$
$$+ pass_{115} + pass_{116} + pass_{117} + pass_{118} + pass_{119} + pass_{120} + pass_{121} + pass_{122} + pass$$
$$+ pass_{136} + pass_{137} + pass_{138} + pass_{139} + pass_{140} + pass_{141} + pass_{142} + pass_{143} + pass$$
$$+ pass_{157} + pass_{158} + pass_{159} + pass_{160} + pass_{161} + pass_{162} + pass_{163} + pass_{164} + pass$$
$$+ pass_{178} + pass_{179} + pass_{180} + pass_{181} + pass_{182} + pass_{183} + pass_{184} + pass_{185} + pass$$

In [38]:

```
# Вызов функции оптимизации:
JuMP.optimize!(model)
termination_status(model)
```

Out[38]:

```
OPTIMAL::TerminationStatusCode = 1
```

In [39]:

```
# Просмотр результата:
print(JuMP.objective_value(model)," passports:",join(cntr[findall(JuMP.value.(pass
```

```
63.0 passports:Afghanistan, Andorra, Argentina, Australia, Azerbaijan, Bahrain, Br
unei, Cambodia, Cameroon, Canada, Chile, Colombia, Comoros, DR Congo, Djibouti, Eq
uatorial Guinea, Eritrea, Fiji, Gabon, Georgia, Guinea, Guinea-Bissau, Hong Kong,
Hungary, Indonesia, Iraq, Ireland, Israel, Jamaica, Japan, Kuwait, Laos, Liberia,
Libya, Macao, Madagascar, Malaysia, Maldives, Marshall Islands, Mauritania, Maurit
ius, Mongolia, Mozambique, Nauru, Nepal, New Zealand, North Korea, Palestine, Papu
a New Guinea, Qatar, Saudi Arabia, Solomon Islands, Somalia, South Sudan, Sri Lank
a, Syria, Taiwan, Timor-Leste, Togo, Turkmenistan, United States, Uruguay, Vietnam
```

8.2.5. Портфельные инвестиции

In [40]:

```
# Подключение необходимых пакетов:
import Pkg
Pkg.add("DataFrames")
Pkg.add("XLSX")
Pkg.add("Plots")
Pkg.add("PyPlot")
Pkg.add("Convex")
Pkg.add("SCS")
Pkg.add("Statistics")
using DataFrames
using XLSX
using Plots
pyplot()
using Convex
using SCS
using Statistics
```

```
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
```

In [41]:

```
# Считываем данные и размещаем их во фрейм:
T = DataFrame(XLSX.readtable("stock_prices.xlsx","Sheet2")...)
```

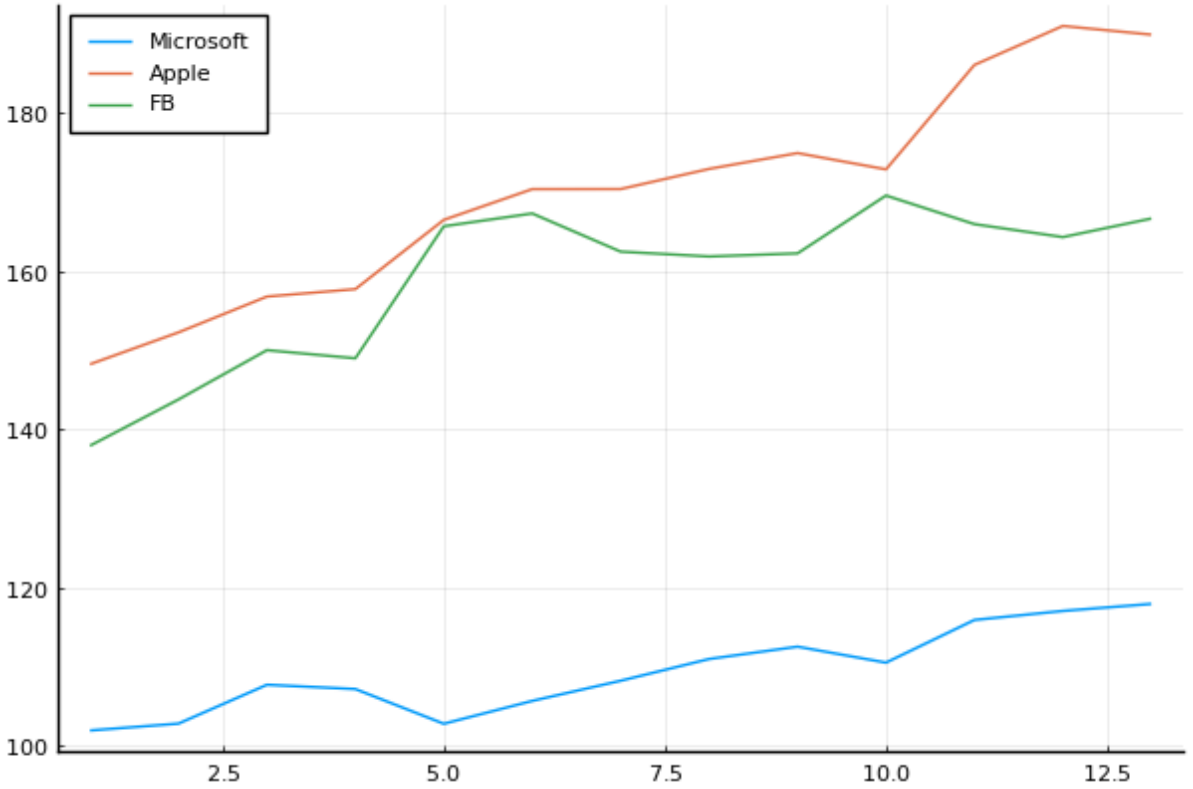
Out[41]: 13 rows × 3 columns

	MSFT	FB	AAPL
	Any	Any	Any
1	101.93	137.95	148.26
2	102.8	143.8	152.29
3	107.71	150.04	156.82
4	107.17	149.01	157.76
5	102.78	165.71	166.52
6	105.67	167.33	170.41
7	108.22	162.5	170.42
8	110.97	161.89	172.97
9	112.53	162.28	174.97
10	110.51	169.6	172.91
11	115.91	165.98	186.12
12	117.05	164.34	191.05
13	117.94	166.69	189.95

In [42]:

```
# Построение графика:
plot(T[!,:MSFT],label="Microsoft")
plot!(T[!,:AAPL],label="Apple")
plot!(T[!,:FB],label="FB")
```

Out[42]:



In [43]:

```
# Данные о ценах на акции размещаем в матрице:
prices_matrix = Matrix(T)
```

Out[43]:

```
13×3 Array{Any,2}:
 101.93  137.95  148.26
 102.8   143.8   152.29
 107.71  150.04  156.82
 107.17  149.01  157.76
 102.78  165.71  166.52
 105.67  167.33  170.41
 108.22  162.5   170.42
 110.97  161.89  172.97
 112.53  162.28  174.97
 110.51  169.6   172.91
 115.91  165.98  186.12
 117.05  164.34  191.05
 117.94  166.69  189.95
```

In [44]:

```
# Вычисление матрицы доходности за период времени:
M1 = prices_matrix[1:end-1,:]
M2 = prices_matrix[2:end,:]
```

Out[44]:

```
12×3 Array{Any,2}:
 102.8   143.8   152.29
 107.71  150.04  156.82
 107.17  149.01  157.76
 102.78  165.71  166.52
 105.67  167.33  170.41
 108.22  162.5   170.42
 110.97  161.89  172.97
 112.53  162.28  174.97
 110.51  169.6   172.91
 115.91  165.98  186.12
 117.05  164.34  191.05
 117.94  166.69  189.95
```

In [45]:

```
# Матрица доходности:
R = (M2.-M1)./M1
```

Out[45]:
```
12×3 Array{Float64,2}:
  0.00853527   0.0424067    0.027182
  0.0477626    0.0433936    0.0297459
 -0.00501346  -0.00686484   0.00599413
 -0.040963     0.112073     0.0555274
  0.0281183    0.00977611   0.0233606
  0.0241317   -0.0288651    5.8682e-5
  0.0254112   -0.00375385   0.014963
  0.0140579    0.00240904   0.0115627
 -0.0179508    0.0451072   -0.0117734
  0.0488644   -0.0213443    0.0763981
  0.00983522  -0.00988071   0.0264883
  0.00760359   0.0142996   -0.00575766
```

In [46]:
```
# Матрица рисков:
risk_matrix = cov(R)
```

Out[46]:
```
3×3 Array{Float64,2}:
  0.000659383  -0.000630653  0.000139112
 -0.000630653   0.00152162   0.000192288
  0.000139112   0.000192288  0.000635503
```

In [47]:
```
# Проверка положительной определённости матрицы рисков:
isposdef(risk_matrix)
```

Out[47]:
```
true
```

In [48]:
```
# Доход от каждой из компаний:
r = mean(R,dims=1)[:]
```

Out[48]:
```
3-element Array{Float64,1}:
 0.012532748705136572
 0.016563036855293173
 0.02114580465503291
```

In [49]:
```
# Вектор инвестиций:
x = Variable(length(r))
```

Out[49]:
```
Variable
size: (3, 1)
sign: real
vexity: affine
id: 111…765
```

In [50]:
```
# Объект модели:
problem = minimize(Convex.quadform(x,risk_matrix),[sum(x)==1;r'*x>=0.02;x.>=0])
```

Out[50]:
```
minimize
└─ * (convex; positive)
   ├─ 1
   └─ qol_elem (convex; positive)
      ├─ norm2 (convex; positive)
      │  └─ …
      └─ [1.0]
subject to
├─ == constraint (affine)
│  ├─ sum (affine; real)
│  │  └─ 3-element real variable (id: 111…765)
│  └─ 1
├─ >= constraint (affine)
│  ├─ * (affine; real)
│  │  ├─ [0.0125327 0.016563 0.0211458]
│  │  └─ 3-element real variable (id: 111…765)
│  └─ 0.02
├─ >= constraint (affine)
│  ├─ index (affine; real)
│  │  └─ 3-element real variable (id: 111…765)
│  └─ 0
├─ >= constraint (affine)
│  ├─ index (affine; real)
│  │  └─ 3-element real variable (id: 111…765)
│  └─ 0
└─ >= constraint (affine)
   ├─ index (affine; real)
   │  └─ 3-element real variable (id: 111…765)
   └─ 0

status: `solve!` not called yet
```

In [51]:
```
# Находим решение:
solve!(problem, SCS.Optimizer)
```

```
------------------------------------------------------------------------------
        SCS v2.1.2 - Splitting Conic Solver
        (c) Brendan O'Donoghue, Stanford University, 2012
------------------------------------------------------------------------------
Lin-sys: sparse-indirect, nnz in A = 24, CG tol ~ 1/iter^(2.00)
eps = 1.00e-005, alpha = 1.50, max_iters = 5000, normalize = 1, scale = 1.00
acceleration_lookback = 10, rho_x = 1.00e-003
Variables n = 6, constraints m = 14
Cones:  primal zero / dual free vars: 2
        linear vars: 5
        soc vars: 7, soc blks: 2
Setup time: 4.75e-005s
------------------------------------------------------------------------------
 Iter | pri res | dua res | rel gap | pri obj | dua obj | kap/tau | time (s)
------------------------------------------------------------------------------
     0|1.71e+019 2.86e+019 1.00e+000 -3.40e+019 5.17e+018 5.49e+019 1.99e-005
    97|4.78e-011 1.28e-010 1.08e-010 4.85e-004 4.85e-004 3.76e-017 3.34e-003
------------------------------------------------------------------------------
Status: Solved
Timing: Solve time: 3.35e-003s
        Lin-sys: avg # CG iterations: 3.43, avg solve time: 6.04e-007s
        Cones: avg projection time: 7.86e-008s
        Acceleration: avg step time: 3.27e-005s
------------------------------------------------------------------------------
Error metrics:
dist(s, K) = 6.5489e-018, dist(y, K*) = 0.0000e+000, s'y/|s||y| = 1.6666e-017
primal res: |Ax + s - b|_2 / (1 + |b|_2) = 4.7779e-011
dual res:   |A'y + c|_2 / (1 + |c|_2) = 1.2839e-010
rel gap:    |c'x + b'y| / (1 + |c'x| + |b'y|) = 1.0795e-010
------------------------------------------------------------------------------
c'x = 0.0005, -b'y = 0.0005
==============================================================================
```

In [52]:

```
#Проверяем выполнение условия
sum(x.value)
```

Out[52]:

```
1.0000000000510323
```

In [53]:

```
#Проверяем выполнение условия на уровень доходности от 2%:
r'*x.value
```

Out[53]:

```
1×1 LinearAlgebra.Adjoint{Float64,Array{Float64,1}}:
 0.020000000000662013
```

In [54]:

```
#Переводим процентные значения компонент вектора инвестиций в фактические денежные
x.value .* 1000
```

Out[54]:

```
3×1 Array{Float64,2}:
  67.95414742252918
 122.30857118794069
 809.7372814405625
```

8.2.6. Восстановление изображения

In [55]:

```
# Подключение необходимых пакетов:
import Pkg
Pkg.add("ImageMagick")
Pkg.add("Convex")
Pkg.add("SCS")
using Images
```

```
using Convex
using SCS
```

```
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
  Resolving package versions...
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Project.toml`
No Changes to `C:\Users\Admin\.julia\environments\v1.5\Manifest.toml`
ArgumentError: Package Images not found in current path:
- Run `import Pkg; Pkg.add("Images")` to install the Images package.


Stacktrace:
 [1] require(::Module, ::Symbol) at .\loading.jl:893
 [2] include_string(::Function, ::Module, ::String, ::String) at .\loading.jl:1091
 [3] execute_code(::String, ::String) at C:\Users\Admin\.julia\packages\IJulia\rWZ
9e\src\execute_request.jl:27
 [4] execute_request(::ZMQ.Socket, ::IJulia.Msg) at C:\Users\Admin\.julia\packages
\IJulia\rWZ9e\src\execute_request.jl:86
 [5] #invokelatest#1 at .\essentials.jl:710 [inlined]
 [6] invokelatest at .\essentials.jl:709 [inlined]
 [7] eventloop(::ZMQ.Socket) at C:\Users\Admin\.julia\packages\IJulia\rWZ9e\src\ev
entloop.jl:8
 [8] (::IJulia.var"#15#18")() at .\task.jl:356
```

In [56]:

```
# Считывание исходного изображения:
Kref = load("khiam-small.jpg")
```

```
UndefVarError: load not defined

Stacktrace:
 [1] top-level scope at In[56]:2
 [2] include_string(::Function, ::Module, ::String, ::String) at .\loading.jl:1091
 [3] execute_code(::String, ::String) at C:\Users\Admin\.julia\packages\IJulia\rWZ
9e\src\execute_request.jl:27
 [4] execute_request(::ZMQ.Socket, ::IJulia.Msg) at C:\Users\Admin\.julia\packages
\IJulia\rWZ9e\src\execute_request.jl:86
 [5] #invokelatest#1 at .\essentials.jl:710 [inlined]
 [6] invokelatest at .\essentials.jl:709 [inlined]
 [7] eventloop(::ZMQ.Socket) at C:\Users\Admin\.julia\packages\IJulia\rWZ9e\src\ev
entloop.jl:8
 [8] (::IJulia.var"#15#18")() at .\task.jl:356
```

In [57]:

```
#Преобразуем изображение в оттенки серого и испортим некоторые пиксели:
K = copy(Kref)
p = prod(size(K))
missingids = rand(1:p,400)
K[missingids] .= RGBX{N0f8}(0.0,0.0,0.0)
K
Gray.(K)
```

```
UndefVarError: Kref not defined

Stacktrace:
 [1] top-level scope at In[57]:2
 [2] include_string(::Function, ::Module, ::String, ::String) at .\loading.jl:1091
 [3] execute_code(::String, ::String) at C:\Users\Admin\.julia\packages\IJulia\rWZ
9e\src\execute_request.jl:27
 [4] execute_request(::ZMQ.Socket, ::IJulia.Msg) at C:\Users\Admin\.julia\packages
\IJulia\rWZ9e\src\execute_request.jl:86
 [5] #invokelatest#1 at .\essentials.jl:710 [inlined]
 [6] invokelatest at .\essentials.jl:709 [inlined]
 [7] eventloop(::ZMQ.Socket) at C:\Users\Admin\.julia\packages\IJulia\rWZ9e\src\ev
entloop.jl:8
 [8] (::IJulia.var"#15#18")() at .\task.jl:356
```

In [58]:

```
# Матрица цветов:
Y = Float64.(Gray.(K));
```

```
UndefVarError: K not defined

Stacktrace:
 [1] top-level scope at In[58]:2
 [2] include_string(::Function, ::Module, ::String, ::String) at .\loading.jl:1091
 [3] execute_code(::String, ::String) at C:\Users\Admin\.julia\packages\IJulia\rWZ
9e\src\execute_request.jl:27
 [4] execute_request(::ZMQ.Socket, ::IJulia.Msg) at C:\Users\Admin\.julia\packages
\IJulia\rWZ9e\src\execute_request.jl:86
 [5] #invokelatest#1 at .\essentials.jl:710 [inlined]
 [6] invokelatest at .\essentials.jl:709 [inlined]
 [7] eventloop(::ZMQ.Socket) at C:\Users\Admin\.julia\packages\IJulia\rWZ9e\src\ev
entloop.jl:8
 [8] (::IJulia.var"#15#18")() at .\task.jl:356
```

In [59]:

```
#матрица X, в которой минимизируется норма ядра матрицы
correctids = findall(Y[:].!=0)
X = Convex.Variable(size(Y))
problem = minimize(nuclearnorm(X))
problem.constraints += X[correctids]==Y[correctids]
```

```
UndefVarError: Y not defined

Stacktrace:
 [1] top-level scope at In[59]:2
 [2] include_string(::Function, ::Module, ::String, ::String) at .\loading.jl:1091
 [3] execute_code(::String, ::String) at C:\Users\Admin\.julia\packages\IJulia\rWZ
9e\src\execute_request.jl:27
 [4] execute_request(::ZMQ.Socket, ::IJulia.Msg) at C:\Users\Admin\.julia\packages
\IJulia\rWZ9e\src\execute_request.jl:86
 [5] #invokelatest#1 at .\essentials.jl:710 [inlined]
 [6] invokelatest at .\essentials.jl:709 [inlined]
 [7] eventloop(::ZMQ.Socket) at C:\Users\Admin\.julia\packages\IJulia\rWZ9e\src\ev
entloop.jl:8
 [8] (::IJulia.var"#15#18")() at .\task.jl:356
```

In [60]:

```
# Находим решение:
solve!(problem, SCS.Optimizer(eps=1e-3, alpha=1.5))
```

```
------------------------------------------------------------------------------
        SCS v2.1.2 - Splitting Conic Solver
        (c) Brendan O'Donoghue, Stanford University, 2012
------------------------------------------------------------------------------
Lin-sys: sparse-indirect, nnz in A = 24, CG tol ~ 1/iter^(2.00)
eps = 1.00e-003, alpha = 1.50, max_iters = 5000, normalize = 1, scale = 1.00
acceleration_lookback = 10, rho_x = 1.00e-003
Variables n = 6, constraints m = 14
Cones:  primal zero / dual free vars: 2
        linear vars: 5
        soc vars: 7, soc blks: 2
Setup time: 3.98e-005s
------------------------------------------------------------------------------
 Iter | pri res | dua res | rel gap | pri obj | dua obj | kap/tau | time (s)
------------------------------------------------------------------------------
     0|1.71e+019 2.86e+019 1.00e+000 -3.40e+019 5.17e+018 5.49e+019 2.18e-005
    80|2.04e-006 1.50e-004 1.31e-006 4.84e-004 4.85e-004 7.41e-017 4.02e-004
------------------------------------------------------------------------------
Status: Solved
Timing: Solve time: 4.03e-004s
        Lin-sys: avg # CG iterations: 3.94, avg solve time: 7.36e-007s
        Cones: avg projection time: 8.52e-008s
        Acceleration: avg step time: 3.39e-006s
------------------------------------------------------------------------------
Error metrics:
dist(s, K) = 1.3491e-016, dist(y, K*) = 0.0000e+000, s'y/|s||y| = 6.5145e-017
primal res: |Ax + s - b|_2 / (1 + |b|_2) = 2.0405e-006
dual res:   |A'y + c|_2 / (1 + |c|_2) = 1.5009e-004
rel gap:    |c'x + b'y| / (1 + |c'x| + |b'y|) = 1.3147e-006
------------------------------------------------------------------------------
c'x = 0.0005, -b'y = 0.0005
==============================================================================
```

In [61]:

```
#Выводим значение нормы и исправленное изображение:
@show norm(float.(Gray.(Kref))-X.value)
@show norm(-X.value)
colorview(Gray, X.value)
```

```
UndefVarError: Kref not defined

Stacktrace:
 [1] top-level scope at show.jl:641
 [2] include_string(::Function, ::Module, ::String, ::String) at .\loading.jl:1091
 [3] execute_code(::String, ::String) at C:\Users\Admin\.julia\packages\IJulia\rWZ
9e\src\execute_request.jl:27
 [4] execute_request(::ZMQ.Socket, ::IJulia.Msg) at C:\Users\Admin\.julia\packages
\IJulia\rWZ9e\src\execute_request.jl:86
 [5] #invokelatest#1 at .\essentials.jl:710 [inlined]
 [6] invokelatest at .\essentials.jl:709 [inlined]
 [7] eventloop(::ZMQ.Socket) at C:\Users\Admin\.julia\packages\IJulia\rWZ9e\src\ev
entloop.jl:8
 [8] (::IJulia.var"#15#18")() at .\task.jl:356
```

In [ ]:

In [ ]:

In [ ]: