In [1]:

```
# Pkg.add("Cairo");
using Distributions

mu = [10 20 30 40]';
Lambda =
[1.0 0 0
 0 1.0 0
 0 0 1.0
 0.5 0.5 0];
Psi = diagm([0.1, 0.2, 0.3, 0.4]);
d1 = MvNormal([0,0,0],ones(3));
X = zeros(50,4);
for i=1:50
f = rand(d1,1);
d2 = MvNormal(vec(mu+ Lambda*f),Psi);
x = rand(d2,1);
X[i,:] = x';
end
X
```

Out[1]:

```
50×4 Array{Float64,2}:
 10.1719   18.4163  30.6688  38.7584
 10.63     22.1694  29.2598  41.0839
  9.36688  19.5018  31.2387  39.9203
  9.26217  19.3691  30.8179  40.1884
 10.2573   19.1771  28.2707  39.8336
 11.3608   18.7009  32.5208  40.0868
 11.1408   21.2319  28.7629  40.9787
 10.2027   20.3814  30.6974  39.7087
  9.65973  21.1015  31.4924  40.4516
  9.26646  20.4001  28.6904  40.5777
 11.4334   20.6044  31.6017  40.9151
  8.93927  22.0548  30.6187  40.0644
 10.4191   20.8651  31.3216  40.3739
   ⋮                                
  7.33926  18.8322  29.7758  38.7784
  9.93475  21.15    27.7688  40.9617
  8.29452  18.3925  30.9478  40.0767
  9.1321   20.7882  30.3869  39.7228
 11.2158   22.225   30.5573  42.6165
  9.69202  19.094   30.9138  40.1953
 10.3983   22.1094  28.5171  41.5112
 10.5754   19.3998  28.6086  40.4075
 10.5227   20.3018  29.7517  39.9025
 11.3243   19.0142  28.7557  40.8227
  8.65663  20.1532  30.6168  39.1994
 11.193    18.3596  28.9776  38.9614
```

In [2]:

```
function E_Step(X,mu,Lambda,Psi,k)
mu_f_by_x = (X - repmat(mu',size(X,1),1))*(Lambda'*inv(Lambda*Lambda' + Psi))';
Sig_f_by_x = eye(k) - Lambda'*inv(Lambda*Lambda' + Psi)*Lambda;
return mu_f_by_x,Sig_f_by_x;
end
```

Out[2]:

```
E_Step (generic function with 1 method)
```

In [3]:

```
function M_Step(X,mu_f_by_x,Sig_f_by_x,k)
nrows, ncols = size(X);
#Computing mu
mu = mean(X,1)';
#Computing Lambda
Lambda_term1 = zeros(ncols,k);
Lambda_term2 = zeros(k,k);
for i=1:nrows
Lambda_term1 = Lambda_term1 + ((X[i,:] - mu)*mu_f_by_x[i,:]');
Lambda_term2 = Lambda_term2 + (mu_f_by_x[i,:]*mu_f_by_x[i,:]')+Sig_f_by_x;
end
Lambda = Lambda_term1*Lambda_term2;
#Computing Psi
Phi = zeros(ncols,ncols);
for i=1:nrows
Phi = Phi + (X[i,:]*X[i,:]' - X[i,:]*mu_f_by_x[i,:]'*Lambda' - Lambda*mu_f_by_x[i,:
end
Psi = diagm(diag(Phi./nrows));
return mu, Lambda, Psi
end
function compute_llh(X,mu,Lambda,Psi)
llh = 0;
for i=1:size(X,1)
llh = llh + log(pdf(MvNormal(vec(mu),(Lambda*Lambda')+Psi),X[i,:]));
end
return llh;
end
```

Out[3]:

```
compute_llh (generic function with 1 method)
```

In [4]:

```
function fa_em(X,k)
    max_Iter = 100;
    eps = 0.0001;
    llh = -Inf*ones(max_Iter+1);
    mu = mean(X,1)';
    Lambda = rand(size(X,2),k);
    Psi = diagm(rand(size(X,2)));
    print(mu,"\n",Lambda,"\n",Psi,"\n");
    llh[1] = compute_llh(X,mu,Lambda,Psi);
    print(llh[1],"\n")
    for i=1:max_Iter
    print(i,"\n");
    mu_f_by_x,Sig_f_by_x = E_Step(X,mu,Lambda,Psi,k);
    mu_new, Lambda_new, Psi_new = M_Step(X,mu_f_by_x,Sig_f_by_x,k);
    print(mu_new,"\n",Lambda_new,"\n",Psi_new,"\n");
    llh[i+1] = compute_llh(X,mu_new,Lambda_new,Psi_new);
    print(llh[i+1],"\n");
    if(sum(abs.(mu_new-mu))<eps && sum(abs.(Lambda_new-Lambda))<eps && sum(abs.(Psi
        break;
    end
    mu = mu_new;
    Lambda = Lambda_new;
    Psi = Psi_new;
    end
    mu_f_by_x,Sig_f_by_x = E_Step(X,mu,Lambda,Psi,k);
    return (mu, Lambda, Psi, mu_f_by_x, Sig_f_by_x, llh);
end
```

Out[4]:

```
fa_em (generic function with 1 method)
```

In [5]:

```
#Calling the EM approach for dataset X and 3 factors
mu, Lambda, Psi, mu_f_by_x, Sig_f_by_x, llh = fa_em(X,3)
```

```
100
[10.0145; 19.9223; 30.0799; 40.0507]
[19.4681 -49.2282 33.1779; -28.9226 -3.48314 29.8279; 11.2943 24.86
47 9.12157; -9.30443 -18.4574 25.4076]
[2647.89 0.0 0.0 0.0; 0.0 1437.87 0.0 0.0; 0.0 0.0 1467.03 0.0; 0.0
0.0 0.0 2337.95]
-995.147753937666
```

Out[5]:

```
([10.0145; 19.9223; 30.0799; 40.0507], [19.4681 -49.2282 33.1779; -
28.9226 -3.48314 29.8279; 11.2943 24.8647 9.12157; -9.30443 -18.457
4 25.4076], [2647.89 0.0 0.0 0.0; 0.0 1437.87 0.0 0.0; 0.0 0.0 146
7.03 0.0; 0.0 0.0 0.0 2337.95], [0.0198217 0.0050866 -0.0119305; -
0.0239464 -0.0104727 0.0183722; … ; -0.00439356 0.0151278 -0.003682
88; 0.0154488 -0.0164499 -0.0179348], [0.563039 0.0366162 0.1034;
0.0366162 0.444357 0.14396; 0.1034 0.14396 0.484021], [-330.168, -1
846.13, -804.19, -819.022, -965.081, -861.877, -950.156, -857.284,
-970.649, -872.131  …  -864.738, -995.151, -864.739, -995.15, -864.
74, -995.149, -864.74, -995.149, -864.741, -995.148])
```

In [6]:

```
mu
```

Out[6]:

```
4×1 Array{Float64,2}:
 10.0145
 19.9223
 30.0799
 40.0507
```

The results match the parameter mu we used to generate data. Parameter Lambda and Psi did not match the initial parameter as these are the parameters try to estimate using EM algorithm.

In [7]:

```
Lambda
```

Out[7]:

```
4×3 Array{Float64,2}:
  19.4681   -49.2282   33.1779
 -28.9226    -3.48314  29.8279
  11.2943    24.8647    9.12157
  -9.30443  -18.4574   25.4076
```

In [8]:

```
Psi
```

Out[8]:

```
4×4 Array{Float64,2}:
 2647.89     0.0      0.0      0.0
    0.0   1437.87     0.0      0.0
    0.0      0.0   1467.03     0.0
    0.0      0.0      0.0   2337.95
```

In [9]:

```
llh
```

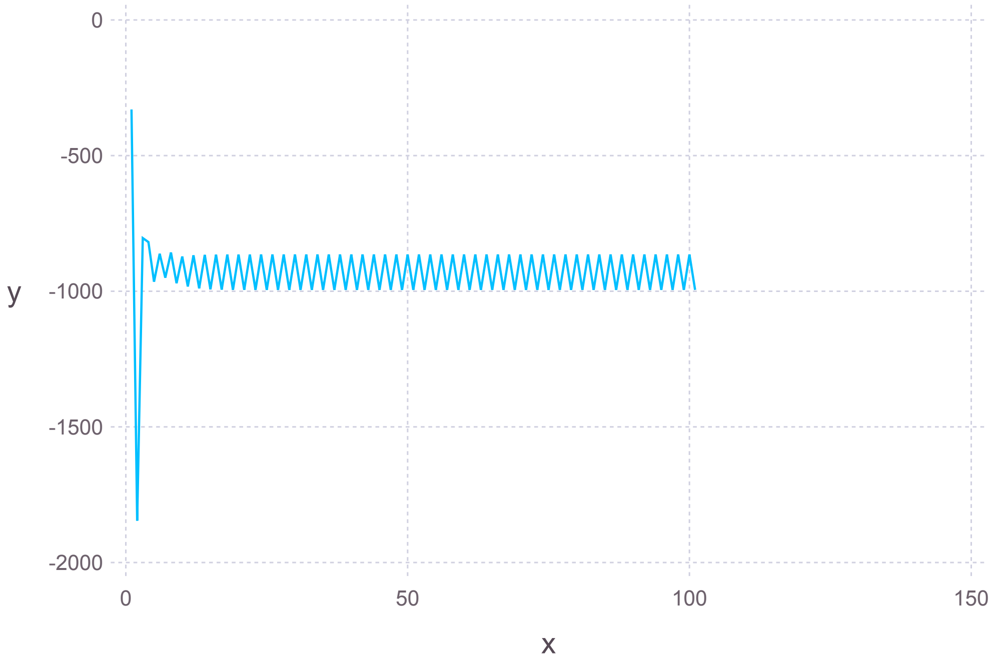
Out[9]:

```
101-element Array{Float64,1}:
  -330.168
 -1846.13
  -804.19
  -819.022
  -965.081
  -861.877
  -950.156
  -857.284
  -970.649
  -872.131
  -982.591
  -867.858
  -989.702
     ⋮
  -864.738
  -995.152
  -864.738
  -995.151
  -864.739
  -995.15
  -864.74
  -995.149
  -864.74
  -995.149
  -864.741
  -995.148
```

In [10]:

```
using Gadfly, Cairo, Fontconfig
#plot the log-likelihood
plot(x=collect(1:1:101), y=llh,Geom.line)
```

Out[10]:



The EM approach converges to the final esitimates.

For k = 2

In [11]:

```
mu, Lambda, Psi, mu_f_by_x, Sig_f_by_x, llh = fa_em(X,2)
```

```
201]
[133.796 0.0 0.0 0.0; 0.0 861.109 0.0 0.0; 0.0 0.0 950.585 0.0; 0.0
0.0 0.0 1695.22]
-862.5250377663626
100
[10.0145; 19.9223; 30.0799; 40.0507]
[49.0707 34.1985; -5.59496 29.8777; -16.8234 -9.55403; 13.2077 27.0
521]
[2523.46 0.0 0.0 0.0; 0.0 914.389 0.0 0.0; 0.0 0.0 1152.85 0.0; 0.0
0.0 0.0 2237.6]
-963.8735017921676
```

Out[11]:

```
([10.0145; 19.9223; 30.0799; 40.0507], [49.0707 34.1985; -5.59496 2
9.8777; -16.8234 -9.55403; 13.2077 27.0521], [2523.46 0.0 0.0 0.0;
0.0 914.389 0.0 0.0; 0.0 0.0 1152.85 0.0; 0.0 0.0 0.0 2237.6], [0.0
069647 -0.025658; -0.00546199 0.0370073; … ; -0.0160432 -0.0045908
8; 0.0252097 -0.0206684], [0.476776 -0.130889; -0.130889 0.38731],
[-305.115, -1625.6, -804.368, -863.689, -883.661, -919.746, -891.78
```

In [12]:

```
mu
```

Out[12]:

```
4×1 Array{Float64,2}:
 10.0145
 19.9223
 30.0799
 40.0507
```

In [13]:

```
Lambda
```

Out[13]:

```
4×2 Array{Float64,2}:
  49.0707   34.1985
  -5.59496  29.8777
 -16.8234   -9.55403
  13.2077   27.0521
```

In [14]:

```
Psi
```

Out[14]:

```
4×4 Array{Float64,2}:
 2523.46    0.0       0.0       0.0
    0.0   914.389     0.0       0.0
    0.0     0.0    1152.85      0.0
    0.0     0.0       0.0    2237.6
```

In [15]:

```
llh
```

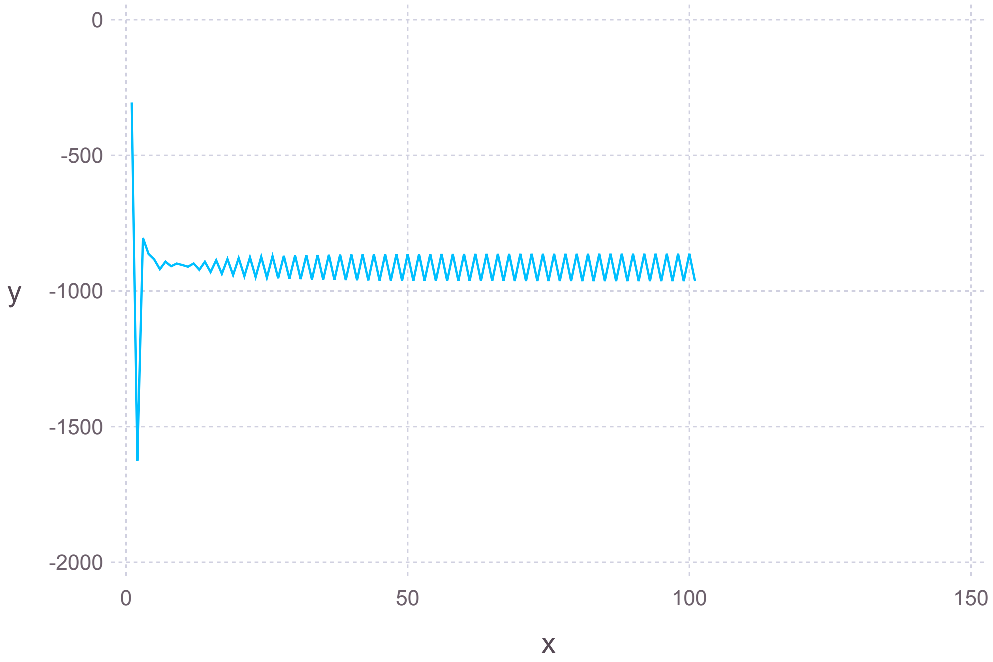
Out[15]:

```
101-element Array{Float64,1}:
  -305.115
 -1625.6
  -804.368
  -863.689
  -883.661
  -919.746
  -891.784
  -908.954
  -898.855
  -904.296
  -910.568
  -898.639
  -921.977
      ⋮
  -862.614
  -963.789
  -862.594
  -963.808
  -862.575
  -963.826
  -862.557
  -963.843
  -862.541
  -963.859
  -862.525
  -963.874
```

In [16]:

```
using Gadfly, Cairo, Fontconfig
#plot the log-likelihood
plot(x=collect(1:1:101), y=llh,Geom.line)
```

Out[16]:



```
##
```

In [ ]:

```
```