In [1]:
```
#Pkg.add("Distributions")
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}:
 11.8342   20.5978  30.8086  41.4171
 10.6987   21.8422  32.0769  41.2695
 10.4924   21.202   30.6792  40.4241
  9.69625  18.7289  30.7595  39.9493
 10.1585   20.7698  28.6737  39.08
  9.02372  20.8617  29.1288  39.8561
 10.224    18.4453  29.5575  39.1574
  9.35707  20.2546  29.7534  39.2309
 10.7427   19.7685  29.9169  39.9085
  8.5487   21.6946  29.4852  40.004
  9.9524   20.6222  29.6129  40.5888
  8.22279  20.9229  28.7771  38.1357
  7.89777  17.8073  29.6034  38.4133
  ⋮
  9.96123  18.7745  29.9192  39.9712
 11.505    20.0853  30.0778  40.7837
  9.88228  18.4037  32.2841  39.8371
  8.90957  19.8485  31.9808  40.5351
  8.33182  20.5547  30.3923  40.3823
 10.4465   20.2861  30.0579  39.5676
  8.35544  20.8412  30.7752  39.534
 11.3365   21.5372  30.7286  41.0307
  9.5099   18.3416  31.5037  37.9532
 10.4488   19.2088  28.358   39.17
  8.03035  21.5109  32.2283  38.5372
 10.1601   20.2442  29.8635  40.0345
```

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 + inv((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,:]*X[i,:]' + Lambda*(m
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_new-Psi))<eps)
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 2 factors
mu, Lambda, Psi, mu_f_by_x, Sig_f_by_x, llh = fa_em(X,3)
```

```
[7490.33 0.0 0.0 0.0; 0.0 3178.56 0.0 0.0; 0.0 0.0 3885.31 0.0; 0.0 0.0 0.0 4066.43]
-1894.8260378320422
98
[9.8195; 19.9924; 30.3568; 39.8279]
[2.20751e5 52446.0 1.11643e5; -1.30429e5 10192.5 1.24847e5; -2.37056e5 -16755.1 63249.5; 50581.7 3
1875.7 1.17565e5]
[1848.77 0.0 0.0 0.0; 0.0 2131.45 0.0 0.0; 0.0 0.0 2569.83 0.0; 0.0 0.0 0.0 2642.46]
-1832.8104157485127
99
[9.8195; 19.9924; 30.3568; 39.8279]
[4.57838e5 1.12881e5 2.5058e5; -1.80713e5 16553.1 1.84228e5; -3.22851e5 -21227.5 93527.5; 126084.0
60895.7 2.07084e5]
[6945.94 0.0 0.0 0.0; 0.0 3041.88 0.0 0.0; 0.0 0.0 3782.58 0.0; 0.0 0.0 0.0 3925.08]
-1912.2181422398894
100
[9.8195; 19.9924; 30.3568; 39.8279]
[395083.0 98182.4 2.19832e5; -1.29613e5 24922.9 1.92598e5; -3.01318e5 -21607.1 78984.8; 1.43069e5
62898.0 2.06267e5]
[2876.8 0.0 0.0 0.0; 0.0 2157.05 0.0 0.0; 0.0 0.0 2811.98 0.0; 0.0 0.0 0.0 2876.37]
-1883.1491581931339
```

In [7]:
```
mu
```

Out[7]:
```
4×1 Array{Float64,2}:
  9.8195
 19.9924
 30.3568
 39.8279
```

In [9]:
```
Lambda
```

Out[9]:
```
4×3 Array{Float64,2}:
 395083.0         98182.4      2.19832e5
     -1.29613e5   24922.9      1.92598e5
     -3.01318e5  -21607.1  78984.8
      1.43069e5   62898.0      2.06267e5
```

In [10]:
```
Psi
```

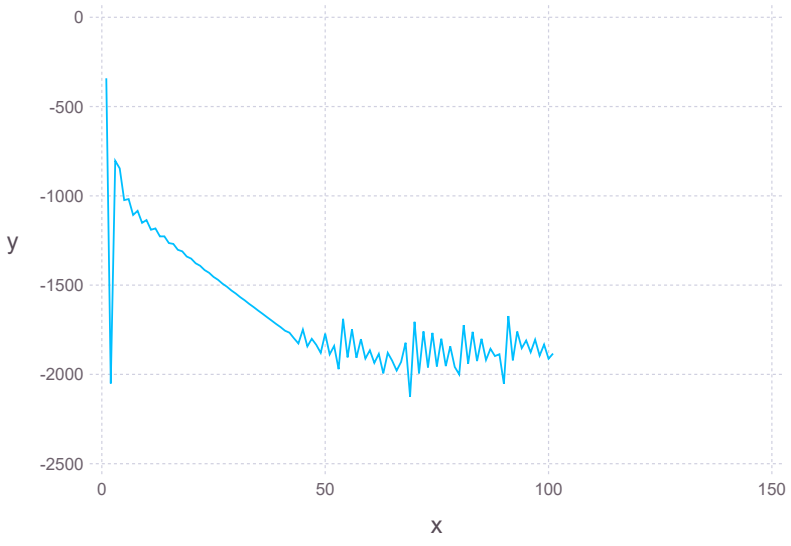
Out[10]:
```
4×4 Array{Float64,2}:
 2876.8     0.0      0.0      0.0
    0.0  2157.05     0.0      0.0
    0.0     0.0   2811.98     0.0
    0.0     0.0      0.0   2876.37
```

In [17]:
```
llh
```

Out[17]:
```
101-element Array{Float64,1}:
  -341.68
 -2050.76
  -803.721
  -846.328
 -1023.66
 -1017.97
 -1107.07
 -1083.68
 -1151.13
 -1135.71
 -1189.8
 -1182.54
 -1227.18
     ⋮
 -2052.9
 -1673.1
 -1921.42
 -1759.88
 -1853.01
 -1809.96
 -1875.81
 -1805.51
 -1894.83
 -1832.81
 -1912.22
 -1883.15
```

In [6]:
```
using Gadfly, Cairo, Fontconfig
#plot the log-likelihood
plot(x=collect(1:1:101), y=llh,Geom.line)
```

Out[6]:



In [ ]:
```
### For k = 2 ###
```

In [11]:
```
mu, Lambda, Psi, mu_f_by_x, Sig_f_by_x, llh = fa_em(X,2)
```

```
[9.8195; 19.9924; 30.3568; 39.8279]
[0.947504 0.203326; 0.748321 0.335209; 0.738288 0.44953; 0.0648133 0.755094]
[0.914696 0.0 0.0 0.0; 0.0 0.985151 0.0 0.0; 0.0 0.0 0.886963 0.0; 0.0 0.0 0.0 0.835965]
-309.1604388661933
1
[9.8195; 19.9924; 30.3568; 39.8279]
[2318.62 1158.02; 1307.44 1005.58; 1643.32 1288.18; 1472.62 1583.49]
[3.94538e6 0.0 0.0 0.0; 0.0 1.68578e6 0.0 0.0; 0.0 0.0 2.70946e6 0.0; 0.0 0.0 0.0 3.05421e6]
-1717.884195111248
2
[9.8195; 19.9924; 30.3568; 39.8279]
[2.51062 1.57771; 2.39875 1.96796; 2.09053 1.73111; 2.40416 1.87644]
[99.101 0.0 0.0 0.0; 0.0 402.11 0.0 0.0; 0.0 0.0 923.875 0.0; 0.0 0.0 0.0 1588.22]
-807.0076306791051
3
[9.8195; 19.9924; 30.3568; 39.8279]
[91.2607 57.3607; 13.1381 11.5892; 1.50793 2.6518; 44.465 29.2261]
[10421.9 0.0 0.0 0.0; 0.0 673.985 0.0 0.0; 0.0 0.0 931.328 0.0; 0.0 0.0 0.0 4102.41]
-986.6876187965773
```

In [12]:
```
mu
```

Out[12]:
```
4×1 Array{Float64,2}:
  9.8195
 19.9924
 30.3568
 39.8279
```

In [13]:
```
Lambda
```

Out[13]:
```
4×3 Array{Float64,2}:
 395083.0         98182.4      2.19832e5
     -1.29613e5   24922.9      1.92598e5
     -3.01318e5  -21607.1  78984.8
      1.43069e5   62898.0      2.06267e5
```

In [14]:
```
Psi
```

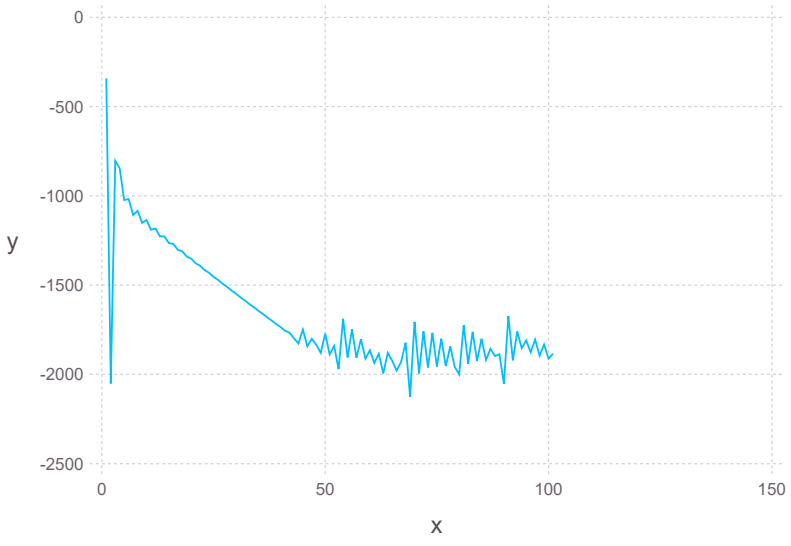
Out[14]:
```
4×4 Array{Float64,2}:
 2876.8     0.0      0.0      0.0
    0.0  2157.05     0.0      0.0
    0.0     0.0   2811.98     0.0
    0.0     0.0      0.0   2876.37
```

In [16]:
```
llh
```

Out[16]:
```
101-element Array{Float64,1}:
  -341.68
 -2050.76
  -803.721
  -846.328
 -1023.66
 -1017.97
 -1107.07
 -1083.68
 -1151.13
 -1135.71
 -1189.8
 -1182.54
 -1227.18
     ⋮
 -2052.9
 -1673.1
 -1921.42
 -1759.88
 -1853.01
 -1809.96
 -1875.81
 -1805.51
 -1894.83
 -1832.81
 -1912.22
 -1883.15
```

In [15]:
```
using Gadfly, Cairo, Fontconfig
#plot the log-likelihood
plot(x=collect(1:1:101), y=llh,Geom.line)
```

Out[15]:



In [ ]:
```

```