1).

$S = \{a, b, c\}$

a)

$$p(z) = \prod_{c \in S} \pi_c^{I(z=c)}$$

b)

$$p(x|z) = \prod_{c \in S} \mathcal{N}(x | \mu_c, \Sigma_c)$$

c).

$$p(x) = \sum_{c \in S} p(z)\, p(x|z)$$

$$= \pi_a \mathcal{N}(x | \mu_a, \Sigma_a)$$

$$+ \pi_b \mathcal{N}(x | \mu_b, \Sigma_b)$$

$$+ \pi_c \mathcal{N}(x | \mu_c, \Sigma_c)$$

d).

$$L = \sum_{i=1}^{n} p(x_i).$$

$$= \sum_{i=1}^{n} \pi_a \mathcal{N}(x_i | \mu_a, \Sigma_a) +$$

$$\pi_b \mathcal{N}(x_i | \mu_b, \Sigma_b) +$$

$$\pi_c \mathcal{N}(x_i | \mu_c, \Sigma_c)$$

(a)

$$l = \log \prod_{i=1}^{n} p(x^{(i)})$$

$$= \log \prod_{i=1}^{n} \Big(\pi_a \mathcal{N}(x_i \mid \mu_a, \Sigma_a) + \pi_b \mathcal{N}(x_i \mid \mu_b, \Sigma_b) + \pi_c \mathcal{N}(x_i \mid \mu_c, \Sigma_c) \Big)$$

(b).

$$\gamma(a) = p(z=a \mid x)$$

$$= \frac{p(x \mid z=a)\, p(z=a)}{\sum_{j \in S} p(z=j)\, p(x \mid z=j)}$$

$$= \frac{\pi_a \mathcal{N}(x \mid \mu_a, \Sigma_a)}{\pi_a \mathcal{N}(x \mid \mu_a, \Sigma_a) + \pi_b \mathcal{N}(x \mid \mu_b, \Sigma_b) + \pi_c \mathcal{N}(x \mid \mu_c, \Sigma_c)}$$

$$\gamma(b) = p(z=b \mid x)$$

$$= \frac{p(z=b)\, p(x \mid z=b)}{\sum_{j \in S} p(z=j)\, p(x \mid z=j)}$$

$$= \frac{\pi_b \mathcal{N}(x \mid \mu_b, \Sigma_b)}{\pi_a \mathcal{N}(x \mid \mu_a, \Sigma_a) + \pi_b \mathcal{N}(x \mid \mu_b, \Sigma_b) + \pi_c \mathcal{N}(x \mid \mu_c, \Sigma_c)}$$

$$\gamma(c) = p(z=c \mid x)$$

$$= \frac{p(z=c)\, p(x \mid z=c)}{\sum_{j \in S} p(z=j)\, p(x \mid z=j)}$$

$$= \frac{\pi_c \mathcal{N}(x \mid \mu_c, \Sigma_c)}{\pi_a \mathcal{N}(x \mid \mu_a, \Sigma_a) + \pi_b \mathcal{N}(x \mid \mu_b, \Sigma_b) + \pi_c \mathcal{N}(x \mid \mu_c, \Sigma_c)}$$

a) $$\mu_a = \frac{\sum_{i=1}^{N} \gamma(a)\, x_i}{\sum_{i=1}^{n} \gamma(a)}$$

$$\mu_b = \frac{\sum_{i=1}^{n} \gamma(b)\, x_i}{\sum_{i=1}^{n} \gamma(b)}$$

$$\mu_c = \frac{\sum_{i=1}^{n} \gamma(c)\, x_i}{\sum_{i=1}^{n} \gamma(c)}$$

$$\Sigma_a = \frac{\sum_{i=1}^{n} \gamma(a)(x_i - \mu_a)(x_i - \mu_a)^T}{\sum_{i=1}^{n} \gamma(a)}$$

$$\Sigma_b = \frac{\sum_{i=1}^{n} \gamma(b)(x_i - \mu_b)(x_i - \mu_b)^T}{\sum_{i=1}^{n} \gamma(b)}$$

$$\Sigma_c = \frac{\sum_{i=1}^{n} \gamma(c)(x_i - \mu_c)(x_i - \mu_c)^T}{\sum_{i=1}^{n} \gamma(c)}$$

$$P_a = \frac{\sum_{i=1}^{N} \gamma(a)}{N} \qquad \boxed{n = N}$$

$$P_b = \frac{\sum_{i=1}^{N} \gamma(b)}{N}, \quad P_c = \frac{\sum_{i=1}^{N} \gamma(c)}{N}$$

i)

~~Given a joint distribution~~

~~$p(x, z \mid \theta)$~~

~~Step 1~~

~~Choose initial~~

j).

Step 1 :-

Pick initial values for.

$\mu_a, \mu_b, \mu_c, \Sigma_a, \Sigma_b, \Sigma_c.$

$P_a, P_b, P_c.$

Step 2 :- Maxiter = 1000.

Step 3 for i=1 : maxiter.

Step 4 :-

E step Compute posterior probabilities.

$\gamma(a), \gamma(b), \gamma(c).$

Step 5 :-

M step :- optimize.

μ_a, μ_b, μ_c

$\Sigma_a, \Sigma_b, \Sigma_c$

$P_a, P_b, P_c.$

Step 6 :-

if $|\mu_a^i - \mu_a^{i-1}| < \epsilon$

&& $|\mu_b^i - \mu_b^{i-1}| < \epsilon$

&& $|\mu_c^i - \mu_c^{i-1}| < \epsilon$

&& $|\Sigma_a^i - \Sigma_a^{i-1}| < \epsilon$

&& $|\Sigma_b^i - \Sigma_b^{i-1}| < \epsilon$

&& $|\Sigma_c^i - \Sigma_c^{i-1}| < \epsilon$

terminate; end.

Step 7 :- end for.

2).

$$L(q,\theta) = \sum_z q(z)\log\left\{\frac{p(x,z|\theta)}{q(z)}\right\}$$

(1)

$$KL(q||p) = -\sum_z q(z)\log\left\{\frac{p(z|x,\theta)}{q(z)}\right\}$$

(2)

Adding (1) & (2)

$$\sum_z q(z)\log\left\{\frac{p(x,z|\theta)}{q(z)}\right\} - \sum_z q(z)\log\left\{\frac{p(z|x,\theta)}{q(z)}\right\}$$

$$= \sum_z q(z)\log p(x|\theta)$$

w.k.t

$$\sum_z q(z) = 1$$

$$= \log p(x|\theta)$$

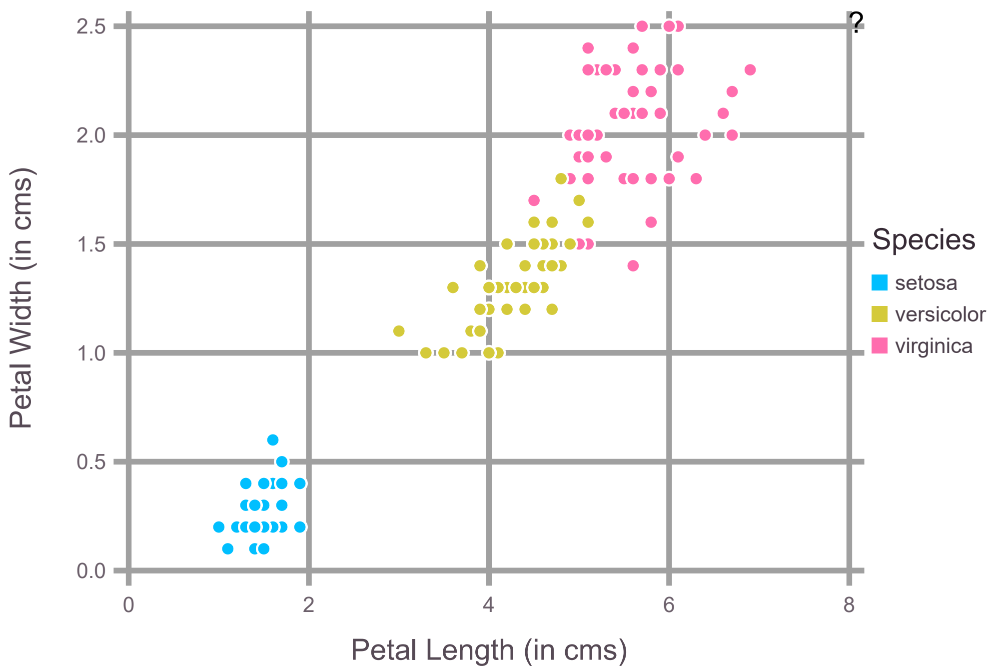
Hence proved.

J)

In [3]:
```
using RDatasets, Gadfly, Distributions;
```

In [6]:
```
data = dataset("datasets","iris");
plot(data,x=:PetalLength,y=:PetalWidth, color=:Species,Geom.point,
Guide.xlabel("Petal Length (in cms)"),
Guide.ylabel("Petal Width (in cms)"),
Coord.Cartesian(xmin=0, xmax=8))
```

Out[6]:



In [47]:
```
function E_step(x,mu_a,mu_b, mu_c,sigma_a,sigma_b,sigma_c,p_a, p_b, p_c)
    numerator_a = zeros(size(x,1));
    numerator_b = zeros(size(x,1));
    numerator_c = zeros(size(x,1));
    denominator = zeros(size(x,1));
    post_a = zeros(size(x,1));
    post_b = zeros(size(x,1));
    post_c = zeros(size(x,1));
    for i=1:size(x,1)
        numerator_a[i] = p_a.*pdf(MvNormal(mu_a,sigma_a),x[i,:]);
        numerator_b[i] = p_b.*pdf(MvNormal(mu_b,sigma_b),x[i,:]);
        numerator_c[i] = p_c.*pdf(MvNormal(mu_c,sigma_c),x[i,:]);
        denominator[i] = numerator_a[i] + numerator_b[i] + numerator_c[i];
        post_a[i] = numerator_a[i] ./denominator[i];
        post_b[i] = numerator_b[i] ./denominator[i];
        post_c[i] = numerator_c[i] ./denominator[i];
    end
    return post_a, post_b, post_c;
end
```

Out[47]: E_step (generic function with 2 methods)

In [20]:
```
function M_step(x,post_a,post_b,post_c)
    mu_a = sum(post_a.*x,1)./sum(post_a);
    mu_a = Vector(mu_a[:]);
    mu_b = sum(post_b.*x,1)./sum(post_b);
    mu_b = Vector(mu_b[:]);
    mu_c = sum(post_c.*x,1)./sum(post_c);
    mu_c = Vector(mu_c[:]);
    sigma_a = round.((post_a.*(x.-mu_a'))'*(x.-mu_a')/sum(post_a),5);
    sigma_b = round.((post_b.*(x.-mu_b'))'*(x.-mu_b')/sum(post_b),5);
    sigma_c = round.((post_c.*(x.-mu_c'))'*(x.-mu_c')/sum(post_c),5);
    p_a = sum(post_a)/size(x,1);
    p_b = sum(post_b)/size(x,1);
    p_c = sum(post_c)/size(x,1);
    return mu_a, mu_b, mu_c, sigma_a, sigma_b, sigma_c, p_a, p_b, p_c;
end
```

Out[20]: M_step (generic function with 1 method)

In [41]:
```
function EM(x,mu_a,mu_b,mu_c,sigma_a,sigma_b,sigma_c,p_a,p_b,p_c)
    maxIter = 1000;
    for i=1:maxIter
        post_a, post_b, post_c = E_step(x,mu_a,mu_b,mu_c,sigma_a,sigma_b,sigma
_c,p_a,p_b,p_c);
        mu_a_new, mu_b_new, mu_c_new, sigma_a_new, sigma_b_new, sigma_c_new, p
_a_new, p_b_new, p_c_new = M_step(x,post_a,post_b,post_c)
        if(sum(abs.(mu_a-mu_a_new))<0.001 && sum(abs.(mu_b-mu_b_new))<0.001 &&
sum(abs.(mu_c-mu_c_new))<0.001
        && sum(abs.(sigma_a-sigma_a_new))<0.001 && sum(abs.(sigma_b-sigma_b_ne
w))<0.001 && sum(abs.(sigma_c-sigma_c_new))<0.001)
            break;
        end;
        mu_a = mu_a_new; mu_b = mu_b_new; mu_c = mu_c_new;
        sigma_a = sigma_a_new; sigma_b = sigma_b_new; sigma_c = sigma_c_new;
        p_a = p_a_new; p_b = p_b_new; p_c = p_c_new;
    end
    return mu_a, mu_b, mu_c, sigma_a, sigma_b, sigma_c, p_a, p_b, p_c;
end
```

Out[41]:
```
EM (generic function with 1 method)
```

In [24]:
```
data_mat_a = data[find(data[:Species].=="setosa"),[:PetalLength,:PetalWidth]];
data_mat_b = data[find(data[:Species].=="versicolor"),[:PetalLength,:PetalWidt
h]];
data_mat_c = data[find(data[:Species].=="virginica"),[:PetalLength,:PetalWidth
]];
nrows_a = size(data_mat_a,1);
nrows_b = size(data_mat_b,1);
nrows_c = size(data_mat_c,1);
#Estimate these using EM for MV Gaussian approach
mean_vec_a = vec([1.5 0.25]);
mean_vec_b = vec([4.2 1.3]);
mean_vec_c = vec([5.6 2.0]);
cov_mat_a = [0.031 0.0061; 0.0061 0.0109];
cov_mat_b = [0.22 0.0732; 0.0732 0.039];
cov_mat_c = [0.3008 0.0466; 0.0466 0.0746];
d_a = MvNormal(mean_vec_a,cov_mat_a);
d_b = MvNormal(mean_vec_b,cov_mat_b);
d_c = MvNormal(mean_vec_c,cov_mat_c);
```

In [50]:
```
x = convert(Array,data[:,[:PetalLength,:PetalWidth]]);
mu_a=[1, 0.5];
mu_b=[3.5, 1.2];
mu_c = [4.5, 1.7];
sigma_a = [0.028 0.0061; 0.0061 0.0107];
sigma_b = [0.23 0.0732; 0.0732 0.035];
sigma_c = [0.25 0.0732; 0.0732 0.037];
p_a = 0.34;
p_b = 0.33;
p_c = 0.33;

mu_a,mu_b,mu_c,sigma_a,sigma_b,sigma_c,p_a,p_b,p_c = EM(x,mu_a,mu_b,mu_c,sigma
_a,sigma_b,sigma_c,p_a,p_b,p_c);
print(mu_a,"\n");
print(mu_b,"\n");
print(mu_c,"\n");
print(sigma_a,"\n");
print(sigma_b,"\n");
print(sigma_c,"\n");
print(p_a,"\n");
print(p_b,"\n");
print(p_c,"\n");
```

```
[1.462, 0.246]
[4.28275, 1.33199]
[5.54699, 2.0298]
[0.02956 0.00595; 0.00595 0.01088]
[0.23996 0.0781; 0.0781 0.04043]
[0.31159 0.05222; 0.05222 0.07387]
0.3333329714767985
0.338012498002689
0.3286545305205126
```

L) Yes it's a good approximation of true distributions

In [51]:
```
data = dataset("datasets","iris");
data_mat_a = data[find(data[:Species].=="setosa"),[:PetalLength,:PetalWidth]];
data_mat_b = data[find(data[:Species].=="versicolor"),[:PetalLength,:PetalWidt
h]];
data_mat_c = data[find(data[:Species].=="virginica"),[:PetalLength,:PetalWidth
]];
nrows_a = size(data_mat_a,1);
nrows_b = size(data_mat_b,1);
nrows_c = size(data_mat_c,1);
#Estimate these using EM for MV Gaussian approach
d_a = MvNormal(vec(mu_a),sigma_a);
d_b = MvNormal(vec(mu_b),sigma_b);
d_c = MvNormal(vec(mu_c),sigma_c);
```

In [53]:
```
a = collect(0:0.05:8);
b = collect(0:0.05:2.5);
pdf_mv = zeros(length(a),length(b));
for i=1:length(a)
    for j=1:length(b)
        pdf_mv[i,j] = maximum([pdf(d_a,[a[i],b[j]]),pdf(d_b,[a[i],b[j]]),pdf(d
_c,[a[i],b[j]])]);
    end
end
plot(layer(x=data_mat_a[:,1],y=data_mat_a[:,2],
Geom.point,Theme(default_color=colorant"red")),layer(x=data_mat_b[:,1],y=data_
mat_b[:,2],
Geom.point,Theme(default_color=colorant"blue")),layer(x=data_mat_c[:,1],y=data
_mat_c[:,2],
Geom.point,Theme(default_color=colorant"green")),layer(z=pdf_mv,x=a,y=b, Geom.
contour(levels=80)),
Coord.Cartesian(xmin=0, xmax=8,ymin=0,ymax=2.55))
```

Out[53]: