Problem Sheet 2

```julia
begin
    using Pkg; Pkg.add(["Distributions", "LinearAlgebra", "Plots", "PlutoUI"])
    using Distributions
    using LinearAlgebra
    using Plots
    using PlutoUI
    default(;linewidth=3.0, legendfontsize=15.0)
end
```

Table of Contents

1. EM algorithm for a Poisson mixture model

Consider a mixture model for a integer valued random variable $n \in \{0, 1, 2, \ldots\}$ given by the distribution

$$P(n|\boldsymbol{\theta}) = \sum_{j=1}^{M} P(j)\ P(n|\theta_j) = \sum_{j=1}^{M} P(j)\ e^{-\theta_j} \frac{\theta_j^n}{n!}\ ,$$

where the component probabilities $P(n|\theta_j)$ are Poisson distributions. Based on a data set of i.i.d.~samples $D = (n_1, n_2, \ldots, n_N)$ we want to estimate the parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M, P(1), \ldots, P(M))$ of this mixture model.

(a) [MATH] Derive an expression for the Maximum Likelihood estimate of θ_1 for $M = 1$, where all obervations come from the same Poisson distribution.

Fill in your answer here or on paper

(b) [MATH] For $M > 1$ the maximum likelihood estimates of the parameters are to be determined using an EM algorithm. Give explicit formulas for the update of θ_j and $P(j)$.

Hint: For the E-step (see the lecture), compute

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}_t) = -\sum_{i=1}^{N}\sum_{j=1}^{M} P_t(j|n_i, \theta_t) \ln\left(P(n_i|\theta_j)\, P(j)\right),$$

where $P_t(j|n_i)$ is the responsibility of component j for generating data point n_i, computed with the current values of the parameters. For the M-step, minimise $\mathcal{L}$ with respect to θ_j and $P(j)$.

Fill in your answer here or on paper

(c) [CODE] Create a toy dataset with $N = 1000$ samples from a mixture of Poisson with $M = 3$, $\theta_1 = 1.0, \theta_2 = 20.0, \theta_3 = 50.0$ and $P(1) = P(2) = P(3) = 1/3$. Implement you EM algorithm to recover these parameters

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

```julia
function mixpoisson(θ, p) # Return a mixture of Poissons with parameters theta and weights p
    MixtureModel(Poisson.(θ), p)
end
```

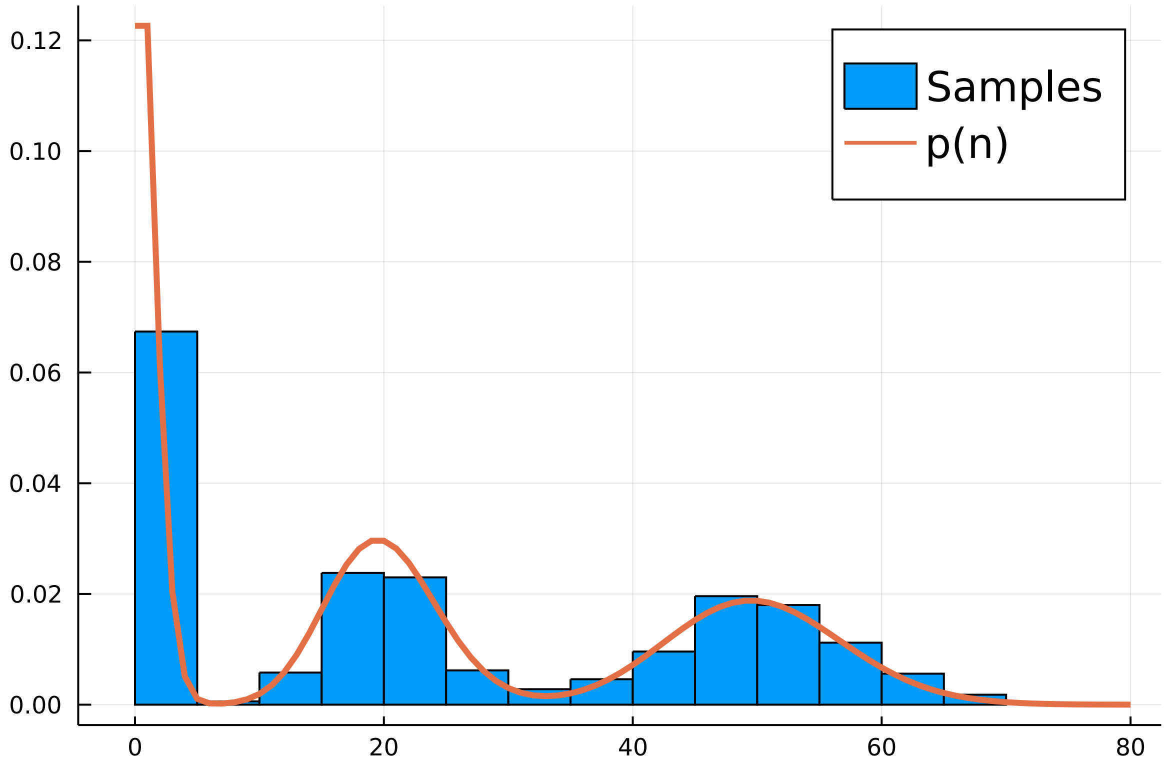
```julia
θ_true =  [1.0, 20.0, 50.0]; # Poisson parameters
```

```julia
p_true =  [1/3, 1/3, 1/3]; # Mixture parameters
```

```julia
d = mixpoisson(θ_true, p_true); # The true Poisson mixture
```

```
N = 1000; # Number of samples
```

```
n = rand(d, N); # Sampled data
```




```
function pt(θ, p, n) # Compute Pt(p | θ, n)
    ## !! CODE MISSING !! ##
    ## Compute here the value of Pt(p | θ, n) for one observation n
    ## !! CODE MISSING !! ##
end;
```

```
function update!(θ, p, n) # Update the parameters
    M = length(p)
    N = length(n)
    pvals = zeros(N, M) # Preallocate the values of pt
    θvals = zeros(N, M) # Preallocate the tmp values for θ
    for i in 1:N # Loop over all the points
        x = pt(θ, p, n[i]) # Compute Pt for each j (x is a vector)
        pvals[i, :] = x # Save Pt value
        θvals[i, :] = n[i] * x # Compute n * Pt
    end
    p .= nothing ## !!CODE MISSING!! Update p given pvals and θvals
    θ .= nothing ## !!CODE MISSING!! Update θ given pvals and θvals
end;
```

Number of components M = 3

```
begin
    if pt(0, 0, 0) !== nothing
        nIter = 10 # Number of iterations
        θ = rand(M) * 50 # Random initialization of the pararameters
        p = rand(M); p /= sum(p) # Random initialization of the weights and normalization
```

```
        anim = Animation() # Create an animation
        anim = @animate for i in 1:nIter # Run the algorithm for a few iterations
            d = mixpoisson(θ, p)
            histogram(n, nbins=20, normalize = true, lab = "", lw = 1.0)
            plot!(0:1:80,x->pdf(d, x), lab = "p(n)", title = "i = $(i)")
            update!(θ, p, n)
        end
        gif(anim, fps = 3)
    end
end
```

2. Bayesian estimation for the Poisson distribution

Consider again the Poisson distribution for an integer valued random variable $n \in \{0, 1, 2, \ldots\}$

$$P(n|\theta) = e^{-\theta} \frac{\theta^n}{n!} \,,$$

- **<u>MATH</u> Write the Poisson distribution in the exponential family form :**

$$P(n|\theta) = f(n) \exp\left[\psi(\theta)\phi(n) + g(\theta)\right]$$

Fill in your answer here or on paper

- **(b) [MATH] Use this exponential family representation to show that the conjugate prior for the Poisson distribution is given by the Gamma density**

$$p(\theta|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \theta^{\alpha-1} e^{-\beta\theta}$$

where α, β are hyperparameters.

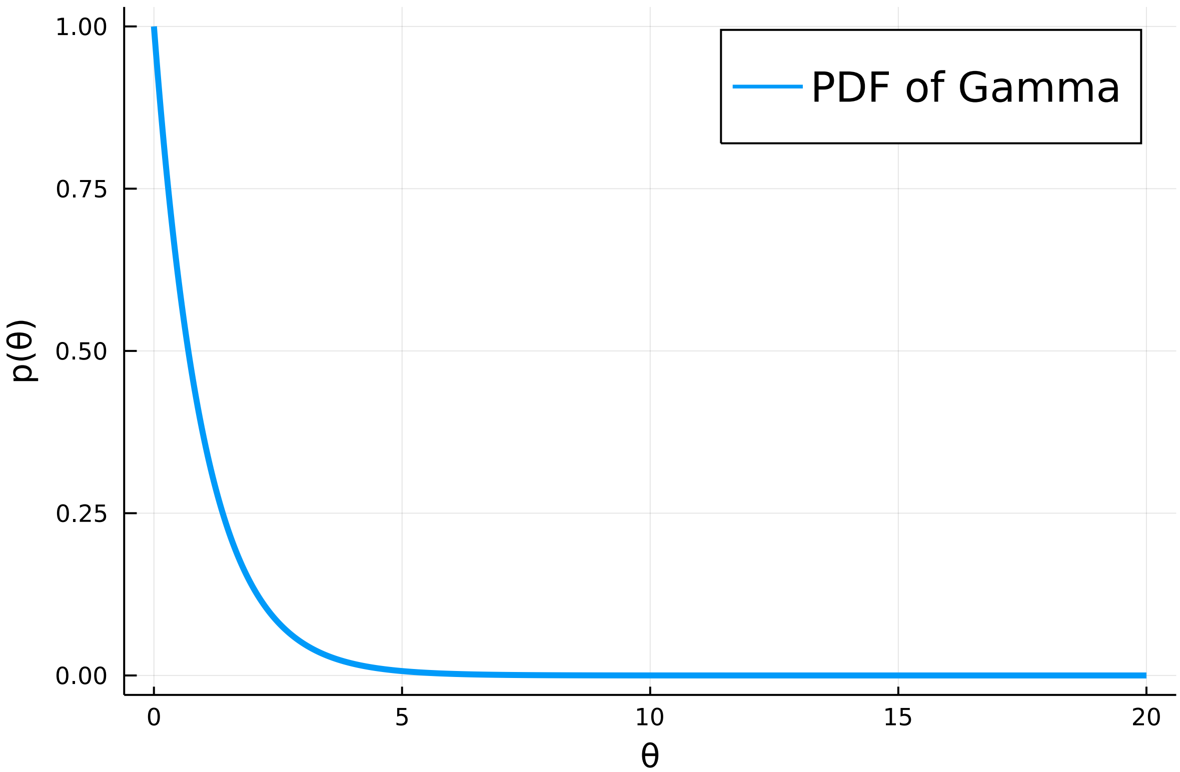
Fill in your answer here or on paper

```
fill_in()
```

Gamma distribution visualization

α_gamma = 1

β_gamma = 1



- **(c) [MATH] Assume that we observe Poisson data $D = (n_1, n_2, \ldots, n_N)$. Write down the posterior distribution $p(\theta|D)$ assuming the Gamma prior. What are the posterior mean and MAP estimators for θ ?**

Fill in your answer here or on paper

- **(d) [MATH] Compute the posterior variance for large N and compare your result with the asymptotic frequentist error of the maximum likelihood estimator.**

> Tip
>
> For the computation of the frequentist error use the **Fisher Information** $J(\theta) \doteq E[(\frac{d \ln P(n|\theta)}{d\theta})^2]$ where the expectation is over the probability distribution $P(n|\theta)$.

Fill in your answer here or on paper

- **(e) [CODE] Estimate the posterior distribution by continuously sampling from a Poisson distribution and compare with the Maximum likelihood estimator.**

θ_🐟= 10.0 : True Poisson parameter

```
d_poisson = Poisson(θ_poisson); # True Poisson distribution
```

alpha (generic function with 1 method)

```
alpha(n, α) = nothing # ## !! CODE MISSING !! give here the posterior parameter alpha
```

beta (generic function with 1 method)

```
beta(N, β) = nothing # ## !! CODE MISSING !! give here the posterior parameter beta
```

mapestimator (generic function with 1 method)

```
mapestimator(n, α, β) = nothing # ## !! CODE MISSING !! compute the MAP estimator of θ
```

mlestimator (generic function with 1 method)

```
mlestimator(n) = nothing # ## !! CODE MISSING !! compute the Maximum Likelihood estimator of θ
```

α = 2.0

β = 3.0

```
d_prior = Gamma(α, 1/β); # Prior distribution
```

```
begin # Elements for plotting
    nrange = 0:1:30
    xrange = 0:0.01:30
    Nmax = 50
    n_samples_per_step = 10
end;
```

MethodError: no method matching /(::Int64, ::Nothing)

```
Closest candidates are:
/(::Union{Int128, Int16, Int32, Int64, Int8, UInt128, UInt16, UInt32, UInt64, UInt8}, !M
/(::Union{Integer, Complex{var"#s79"} where var"#s79"<:Union{Integer, Rational}}, !Matche
/(::Union{Int16, Int32, Int64, Int8, UInt16, UInt32, UInt64, UInt8}, !Matched::BigInt) a
...
```

1. **macro expansion** @ ***Local: 9*** [inlined]
2. **macro expansion** @ *animation.jl:183* [inlined]
3. **top-level scope** @ ***Local: 3***

```
begin
    n_model = Int[]
    anim_2 = @animate for i in 1:Nmax
        for _ in 1:n_samples_per_step
            push!(n_model, rand(d_poisson)) # Add n new samples
        end
        p1 = histogram(n_model; nbins=length(nrange), normalize=true, linewidth=0.0,
title="N = $(i *  n_samples_per_step)", label="")
        plot!(nrange, x -> pdf(d_poisson, x), label="p(D)", ylims=(0, 0.35))
        d_posterior = Gamma(alpha(n_model, α),  1 / beta(length(n_model), β)) #
Distributions.jl uses a different parametrization
        p2 = plot(xrange, x -> pdf(d_posterior, x), label="p(θ|D)")
        plot!(xrange, x -> pdf(d_prior, x); label="p(θ)")
        vline!([mapestimator(n_model, α, β)]; label="MAP", ylims=(0, 1.4))
        vline!([mlestimator(n_model)]; label="ML")
        vline!([θ_poisson]; label="θ_poisson")
        plot(p1, p2; size=(800, 300))
    end
end;
```

UndefVarError: anim_2 not defined

1. **top-level scope** @ ***Local: 1***

```
gif(anim_2, fps = 5)
```

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