Introduction to Probability Distributions

Sujit Sandipan Chaugule[1*], Dr. Amiya Ranjan Bhowmick[2]

[1*]Department of Pharmaceutical Sciences and Technology, Institute of Chemical Technology, Mumbai

[2]Department of Mathematics, Institute of Chemical Technology, Mumbai

Probability Density Function

A function $f(x)$ is called a probability density function (PDF) if

- $f(x) \geq 0$ for all $x \in (-\infty, \infty)$.
- $\int_{-\infty}^{\infty} f(x)dx = 1$

In the following, let us check whether the following function is a PDF. The function

$$g(x) = x^2 e^{-3x}, \quad 0 < x < \infty,$$

and zero otherwise.

In [1]:
```
using Plots, Statistics, StatsBase, LaTeXStrings # Load the required packages
using QuadGK
```

In [2]:
```
function g(x)
    x^2*exp(-3*x)*(x>0)
end

# we can define the function by another way
g(x) = x^2*exp(-3*x)*(x>0)
```

Out[2]:
```
g (generic function with 1 method)
```

In [3]:
```
plot(g,-1,5, color = "red", xlabel = "x", ylabel = "g(x)",
    lw =2 , label = "")
```

Out[3]:

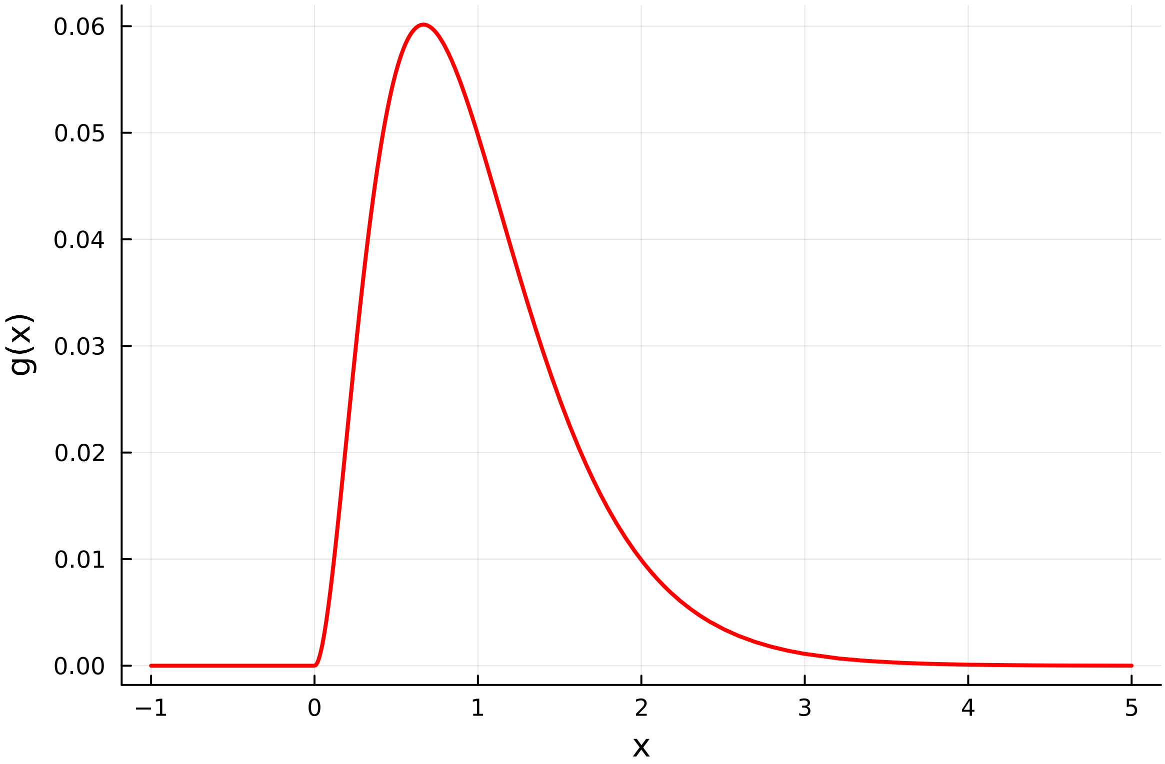

Figure 1: The graph of the function $g(x)$ and we need to check whether this function is a PDF

In [4]:

```
value, er = quadgk(g, 0, Inf)  # numerical integrtion
println("value of integration : $value")
println("absoute error : $er")
```

```
value of integration : 0.07407407407407407
absoute error : 1.1690346578015613e-10
```

The above function is not a PDF. Can we convert this function to a PDF? Yes. If

$$\int_{-\infty}^{\infty} f(x)dx = M$$

hold, then we obtain PDF, $f(x) = \frac{g(x)}{M}$ is a PDF

In [5]:

```
value, er = quadgk(g, 0, Inf)  # numerical integrtion
print(value)
```

```
0.07407407407407407
```

In [6]:

```
f(x) = g(x)/value # define the functionloc
plot(f, -1, 5, color = "blue", lw = 2, xlabel = "x",
    ylabel  = "f(x)", label = "" )
```

Out[6]:

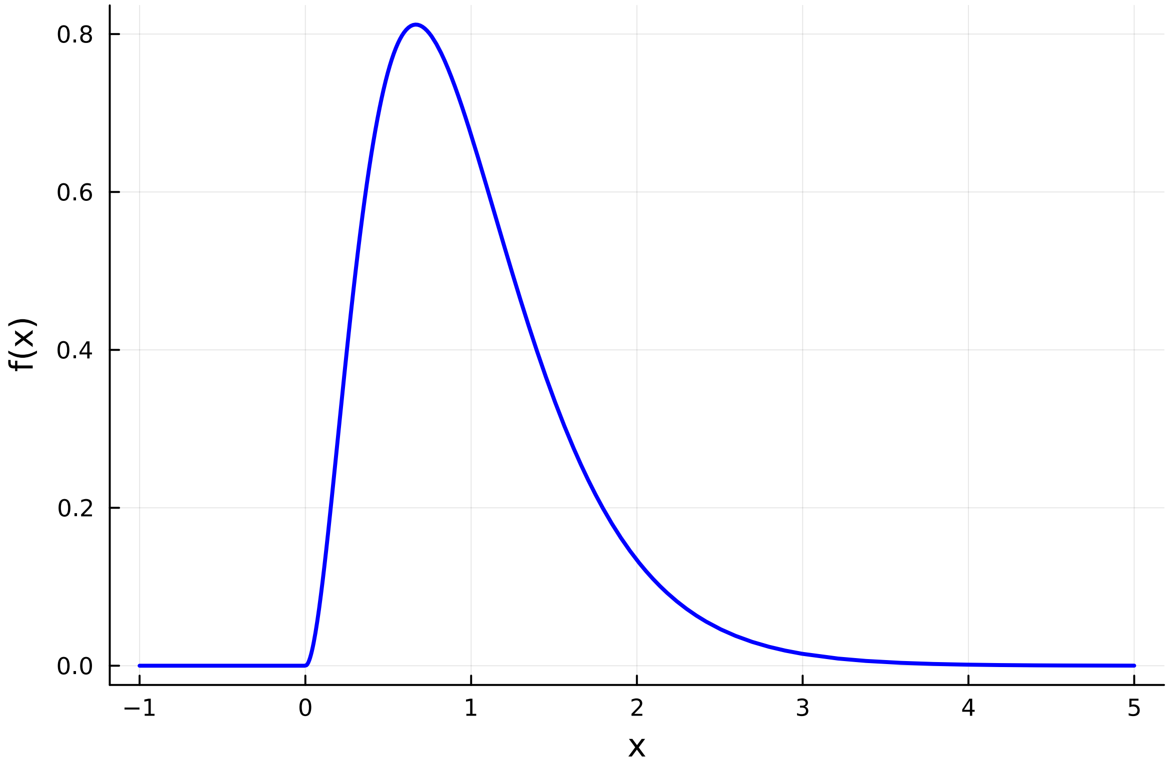

Figure 2: The shape of the PDF $f(x)$ which is obtained from the function $g(x)$, after dividing by the constant M, which is the area under the function $g(x)$. Using the quadgk() function, it is verified that $f(x)$ is indeed a PDF

In [7]:
```
val, abs_er = quadgk(f, 0, Inf)
println("value of integration : $val")
println("absoute error : $abs_er")
```

```
value of integration : 1.0
absoute error : 1.5781968840653722e-9
```

Let us expand the scope of this problem, Consider the choice of following $g(x)$

$$g(x) = e^{-\lambda x} x^{a-1}, \quad 0 < x < \infty,$$

and zero otherwise

In [8]:
```
using Plots, Statistics, StatsBase, LaTeXStrings # Load the required packages
using SpecialFunctions
```

In [9]:
```
using Plots, SpecialFunctions

a = 3
lambda = 3
M = gamma(a) / lambda^a

g(x, a, lambda) = exp(-lambda * x) * x^(a - 1) * (x > 0)

f(x, a, lambda) = g(x, a, lambda) / M  # Define the PDF

plot(layout=(1, 3), size=(800, 400))

a_vals = [1, 3, 5]  # Filling up the first row
lambda = 3
x_range = 0:0.01:8
for (idx, a) in enumerate(a_vals)
    plot!(x_range,x -> g(x, a, lambda),lw=2, color="red",
```

```
          xlabel = L"x", ylabel = L"g(x)", label="a = $a",
          subplot = idx )
end

display(plot!())
```

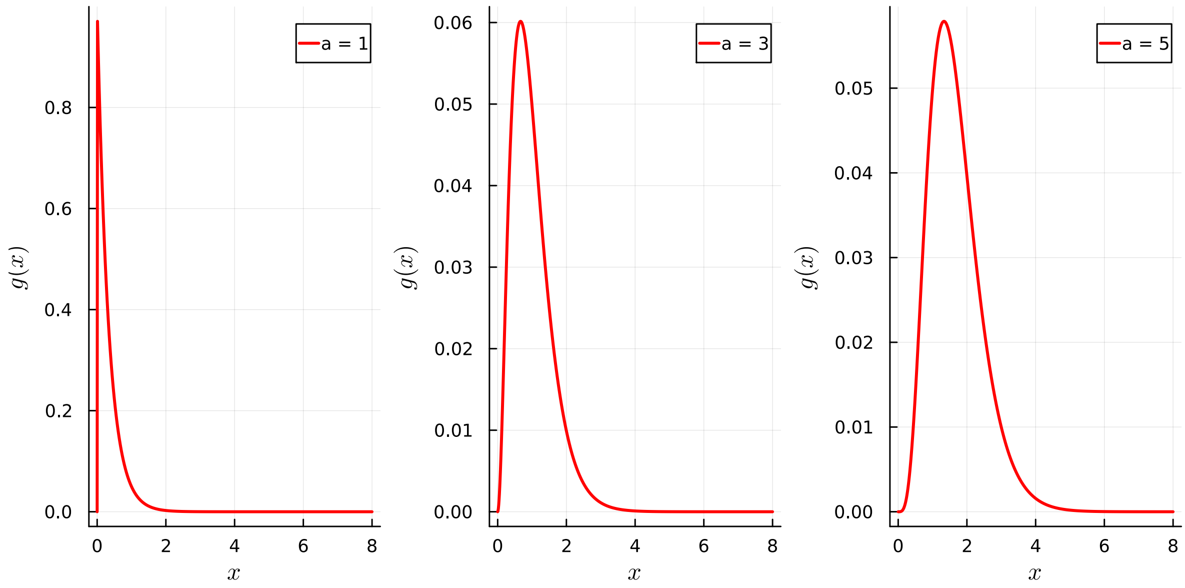

Figure 3: The shape of the fucntion $g(x)$ for different choices of a

In [10]:
```
plot(layout=(1, 3), size=(800, 400))
lambda_vals = [1,5,7]
a = 3
x_range = 0:0.01:8
for (idx, lambda) in enumerate(lambda_vals)
    plot!(x_range,x -> g(x, a, lambda),lw=2, color="red",
          xlabel = L"x", ylabel = L"g(x)", label=L"\lambda = %$lambda",
          subplot = idx )
end

display(plot!())
```

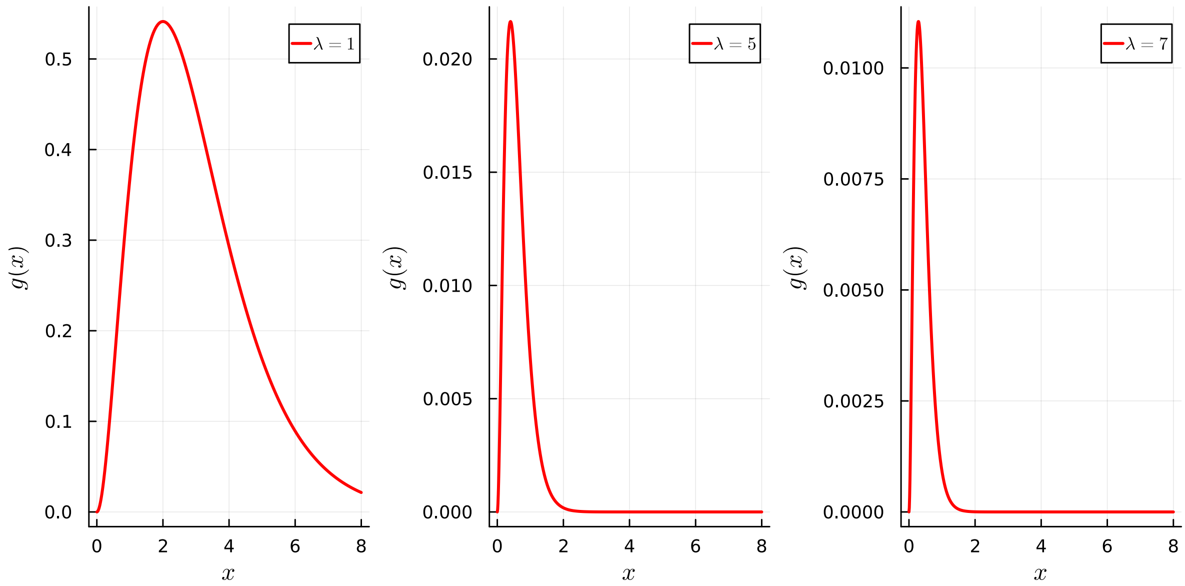

Figure 4: The shape of the fucntion $g(x)$ for different choices of λ

In the following code, we convert the function $g(x)$ to a PDF $f(x)$. Students can identify that the integral can be converted to a gamma integral and

$$\int_0^\infty g(x)dx = \frac{\Gamma(a)}{\lambda^a}.$$

Therefore, the PDF obtained from $g(x)$ is given by

$$f(x) = \begin{cases} \frac{e^{-\lambda x} x^{\backslash a-1} \lambda^a}{\Gamma(a)}, & 0 < x < \infty, \\ 0, & \text{otherwise} \end{cases}$$

In [11]:
```
plot(layout=(1, 3), size=(800, 400))
a_vals = [1, 3, 5]  # Filling up the first row
lambda = 3
x_range = 0:0.01:8
for (idx, a) in enumerate(a_vals)
    plot!(x_range,x -> f(x, a, lambda),lw=2, color="red", label="a = $a",
           xlabel = L"x", ylabel = L"f(x)", subplot = idx )
end

display(plot!())
```

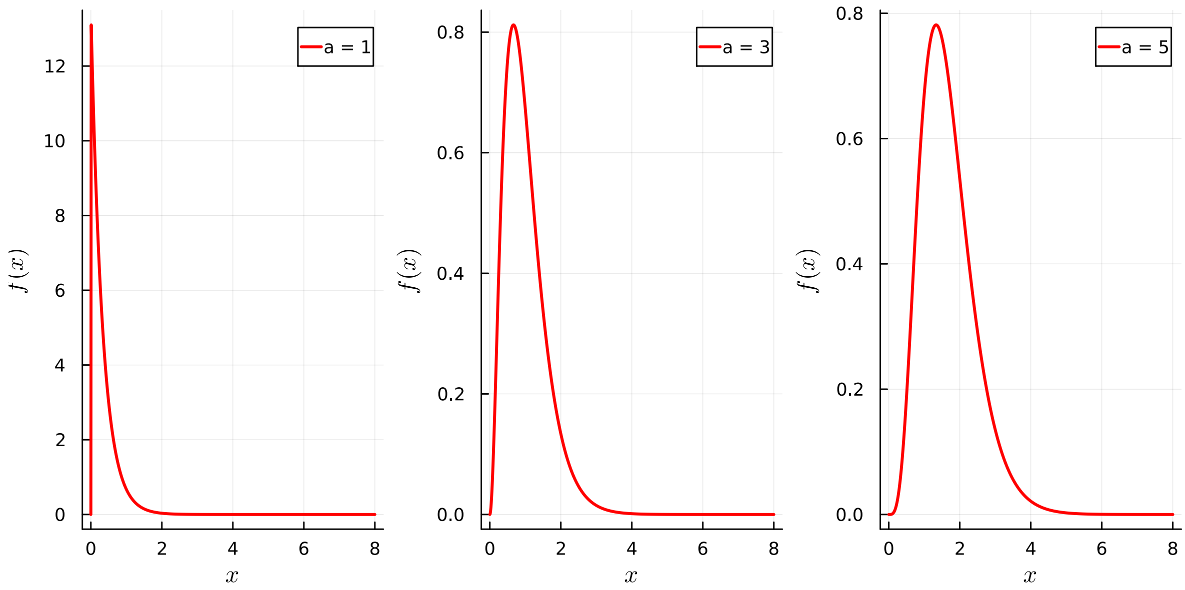

Figure 5: The shape of the fucntion $f(x)$ for different choices of a

In [12]:
```
plot(layout=(1, 3), size=(800, 400))
lambda_vals = [1,5,7]
a = 3
x_range = 0:0.01:8
for (idx, lambda) in enumerate(lambda_vals)
    plot!( x_range,x -> f(x, a, lambda), lw=2, color="red",
        xlabel = L"x", ylabel = L"f(x)",
        label=L"\lambda = %$lambda", subplot = idx )
end

display(plot!())
```

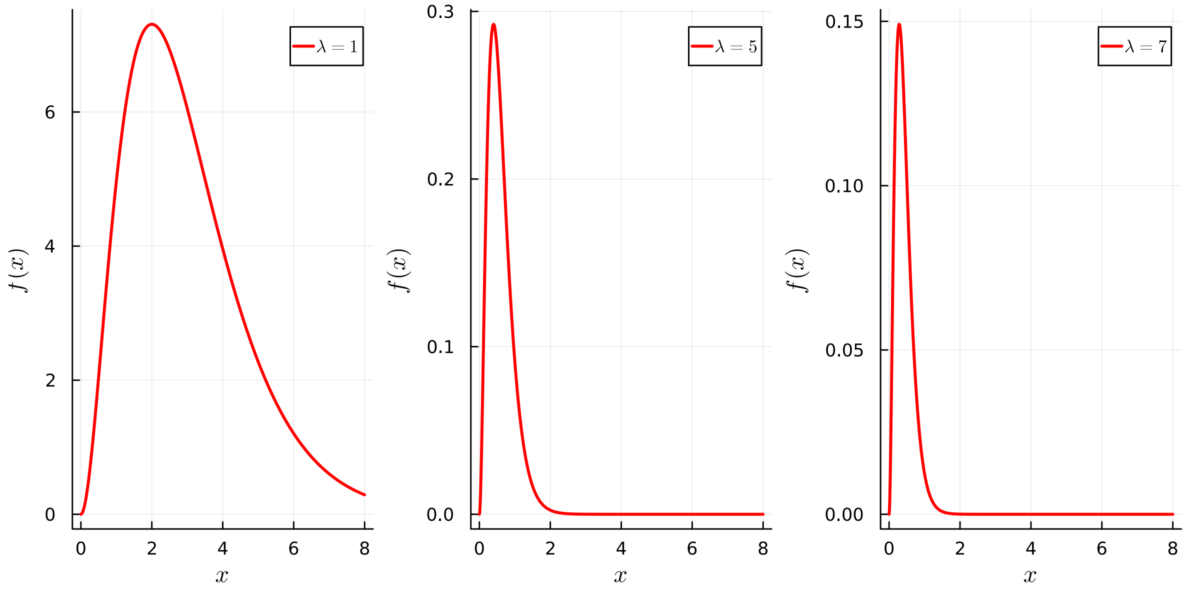

Figure 6: The shape of the function $f(x)$ for different choices of λ

Let us plot f and g together

In [13]:
```
plot(layout=(1, 3), size=(800, 400))
a_vals = [1, 3, 5]
lambda = 3
x_range = 0:0.01:8
for (idx, a) in enumerate(a_vals)
    plot!(x_range,x -> f(x, a, lambda),lw=2, color="red",
          xlabel = L"x",label=L"f(x)", title = "a = $a",
           subplot = idx, )
    plot!(x_range,x -> g(x, a, lambda),lw=2, color="blue",
           label=L"g(x)", subplot = idx )
end

display(plot!())
```

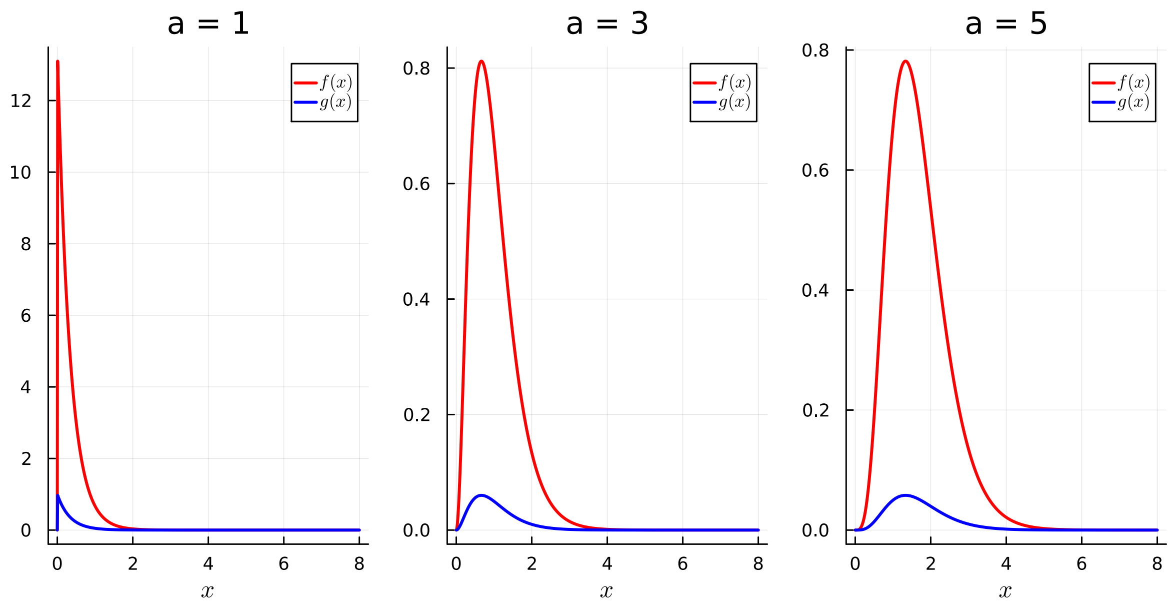

Figure 7: The shape of the function $f(x)$ and $g(x)$ for different choices of a

In [14]:
```
plot(layout=(1, 3), size=(800, 400))
lambda_vals = [1,5,7]
a = 3
```

```julia
x_range = 0:0.01:8
for (idx, lambda) in enumerate(lambda_vals)
    plot!(x_range,x -> f(x, a, lambda), lw=2, color="red",
        label=L"f(x)", subplot = idx, xlabel = L"x",
        title = L"\lambda = %$lambda")
    plot!(x_range,x -> g(x, a, lambda),lw=2, color="blue",
        label=L"g(x)", subplot = idx )
end

display(plot!())
```

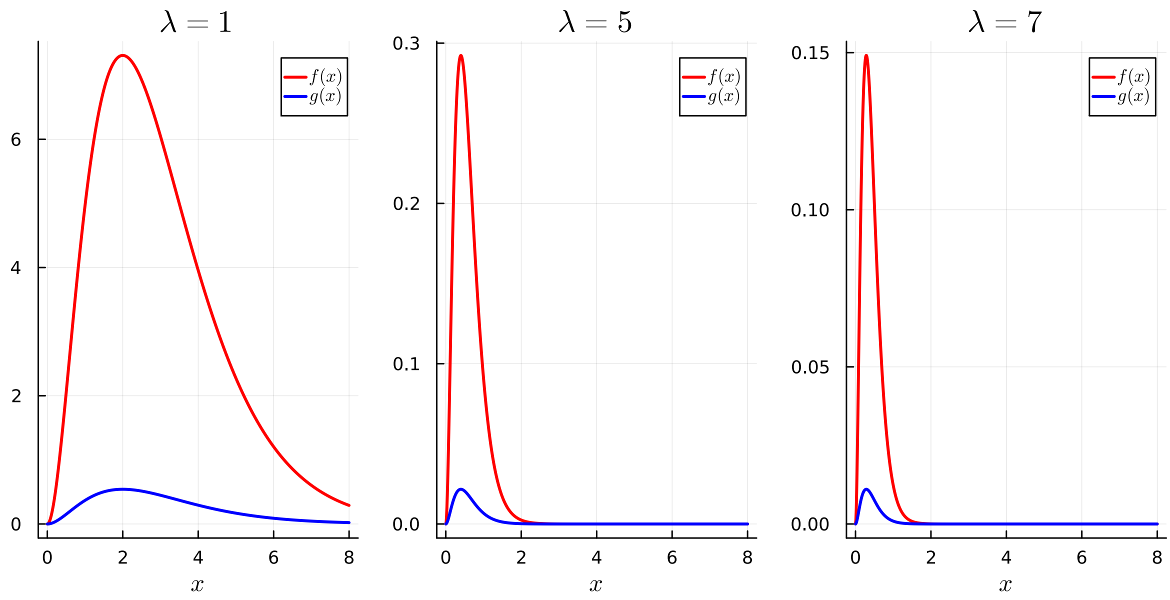

Figure 8: The shape of the function $f(x)$ and $g(x)$ for different choices of λ

In [15]:
```julia
mu = 1
sigma = 1

function f(x)
    (1/(sigma*sqrt(2*pi)))*exp(-(x-mu)^2/(2*sigma^2))
end
```

Out[15]:
```
f (generic function with 2 methods)
```

In [16]:
```julia
val, er = quadgk(f, -Inf, Inf)
println("Intgral value : ", round(val))
```

```
Intgral value : 1.0
```

In [17]:
```julia
f1(x) = x*f(x)
f1_val, er = quadgk(f1, -Inf, Inf)
println("Intgral value : ", round(f1_val))
```

```
Intgral value : 1.0
```

In [18]:
```julia
f2(x) = x^2*f(x)
f2_val, er = quadgk(f2, -Inf, Inf)
println("Intgral value : ", round(f2_val))
```

```
Intgral value : 2.0
```

In [19]:
```julia
M = f2_val - f1_val^2
round(M)
```

Out[19]:
```
1.0
```