Table of Contents

1D Gaussians [10 pts]
- Can the probability density function (pdf) of X ever take values greater than 1 ?
- Write the expression for the pdf of a univariate gaussian:
- Write the code for the function that computes the pdf at x .
- Test your implementation against a standard implementation
- What is the value of the pdf at x = 0 ? What is probability that x = 0 ?
- A Write the transformation that takes x ~ N (0. , 1.) to z ~ N (μ , σ 2)
- Write a code to sample from N (μ , σ 2)
- Test your implementation by computing statistics on the samples
- Plot pdf and normalized histogram of samples

CSC 2506 Probabilistic Learning and Reasoning

Assignment I - Q1

Tianyi Liu liuti110 1005820827

1D Gaussians [10 pts]

Let X be a univariate random variable distributed according to a Gaussian distribution with mean μ and variance σ^2

Can the probability density function (pdf) of X ever take values greater than 1?

Answer: Yes. $\int f(x)dx = 1$ does not have any restrictions on the value of pdf for a specific $X = x$.

Write the expression for the pdf of a univariate gaussian:

Answer: $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

Write the code for the function that computes the pdf at x.

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

```
function gaussian_pdf(x; mean=0., variance=0.01)
  #default variables mean and variance
  #set with keyword arguments
  return 1 /√(2 * π * variance) * ℯ ^ (- (x - mean) ^ 2/(2 * variance))
end
```

Test your implementation against a standard implementation

E.g. from a library, e.g. Distributions.jl.

```
using Test
```

```
using Distributions: pdf, Normal
# Note Normal uses N(mean, stddev) for parameters
```

```
Test.DefaultTestSet("Implementation of Gaussian pdf", Any[], 2, false)
```

```
@testset "Implementation of Gaussian pdf" begin
  x = randn()
  @test gaussian_pdf(x) ≈ pdf.(Normal(0.,sqrt(0.01)),x)
  # ≈ is syntax sugar for isapprox, typed with '\approx <TAB>'
  # or use the full function, like below
  @test isapprox(gaussian_pdf(x,mean=10., variance=1) , pdf.(Normal(10.,
sqrt(1)),x))
end
```

What is the value of the pdf at $x = 0$? What is probability that $x = 0$?

Answer: 3.989422804014327, in general $\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\mu^2}{2\sigma^2}}$; $p(x = 0) = 0$.

A Write the transformation that takes $x \sim \mathcal{N}(0., 1.)$ to $z \sim \mathcal{N}(\mu, \sigma^2)$

A Gaussian with mean μ and variance σ^2 can be written as a simple transformation of the standard Gaussian with mean $0.$ and variance $1.$.

Answer: $\sigma x + \mu$

Write a code to sample from $\mathcal{N}(\mu, \sigma^2)$

Implement function returning n independent samples from $\mathcal{N}(\mu, \sigma^2)$ by transforming n samples from $\mathcal{N}(0., 1.)$

```julia
function sample_gaussian(n; mean=0., variance=0.01)
  # n samples from standard gaussian
  x = randn(n)
  # TODO: transform x to sample z from N(mean,variance)
  z = √variance .* x .+ mean
  return z
end;
```

Test your implementation by computing statistics on the samples

```julia
using Statistics: mean, var
```

```
Test.DefaultTestSet("Numerically testing Gaussian Sample Statistics", Any[], 2, fals
```

```julia
@testset "Numerically testing Gaussian Sample Statistics" begin
  #TODO: choose some values of mean and variance to test
  true_mean = 20.
  true_var = 3.
  #TODO: Sample 100000 samples with sample_gaussian
  data = sample_gaussian(100000, mean=true_mean, variance=true_var)
  #TODO: Use `mean` and `var` to compute statistics
  stat_mean = mean(data)
  stat_var = var(data)
  #TODO: test statistics against true values
  @test isapprox(stat_mean, true_mean, atol=1e-2)
  @test isapprox(stat_var, true_var, atol=1e-2)
  # hint: use isapprox with keyword argument atol=1e-2
end
```

Plot pdf and normalized histogram of samples

Sample 10000 samples from a Gaussian with mean $10.$ and variance 2.0.

1. Plot the **normalized** `histogram` of these samples.
2. On the same axes `plot!` the pdf of this distribution.

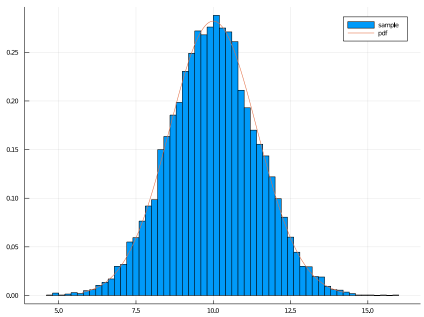
Confirm that the histogram approximates the pdf.

(Note: with `Plots.jl` the function `plot!` will add to the existing axes.)

```julia
using Plots
```

plot_hist_pdf (generic function with 4 methods)

```julia
function plot_hist_pdf(n=10000, mean=10.,variance=2.)
    data = sample_gaussian(n, mean=mean, variance=variance)
    #histogram() #TODO
    histogram(data, norm=true, label="sample")
    #plot!() #TODO
    xs = mean - 3 * √variance : 0.01 : mean + 3 * √variance
    ys = gaussian_pdf.(xs, mean=mean, variance=variance)
    plot!(xs, ys, label="pdf", size=(800,600))
end
```




```julia
plot_hist_pdf()
```

Answer: The histogram approximates the pdf.