Problem Sheet 1

First a couple of packages needs to be used. This will automatically install these packages in a local environment (where the file is currently is. If you would like to do that manually you can open a terminal with Julia and write `] add Distributions` for example

```
begin
    import Pkg # The best package manager in the world
    Pkg.activate(".") # Create a local environment in the current directory
    Pkg.add(["Distributions", "Plots", "Optim", "PlutoUI"]);
    ## Those are needed packages for the different exercises
    using Distributions # Basic library to use probability distributions
    using LinearAlgebra # Standard library for linear algebra operations
    using Plots # Front end for multiple plotting backends, by default it will use GR
    default(linewidth = 3.0, legendfontsize = 15.0) # Some default values for our
plotting
    using Optim # Optimisation library
    using PlutoUI # Some Pluto sugar
end
```

```
# TableOfContents()
```

1. Random experiments

A dice is thrown repeatedly until it shows a 6. Let T be the number of throws for this to happen and q the probability to **not** get a 6. Obviously, T is a random variable.

(a) [MATH] Compute the expectation value $E[T]$ and the variance $V[T]$ of T.

0.8333333333333334

```
q = 5//6
```

You can write your answer here or on paper. For inline LaTeX use `$\alpha$` and for multiline equations use three backticks:

```math

```

(b) Write a program to empirically estimate the mean and the variance of T and compare it to the value you found analytically

0.8333333333333334

```
begin
    N_tries = 10000 # Number of times we run the experiment
    T_vals = zeros(N_tries) # Preallocation of T value at every experiment
    expec_T = zeros(N_tries) # Preallocation of the expectation of T over time
    var_T = zeros(N_tries) # Preallocation of the variance of T over time
    for i in 1:N_tries
        T = 1
        ## !! CODE MISSING !! ##
        ## Write here your code to run a random experiment where T increments
        ## Until a 6 is obtained
        ## Use q the probability of not having a 6
        ## !! CODE MISSING !! ##
        T_vals[i] = T
        expec_T[i] = mean(T_vals[1:i])
        var_T[i] = var(T_vals[1:i])
    end
end;
```

2. Addition of Variances

Let X and Y be independent random variables. Show that:

$$\mathrm{Var}(X+Y) = \mathrm{Var}(X) + \mathrm{Var}(Y),$$

where the variance is defined as :

$$\mathrm{Var}(X) = E[(X - E[X])^2].$$

Tip

Use the fact that for independent U and V, $E[UV] = E[U]E[V]$

Write your answer here or on paper

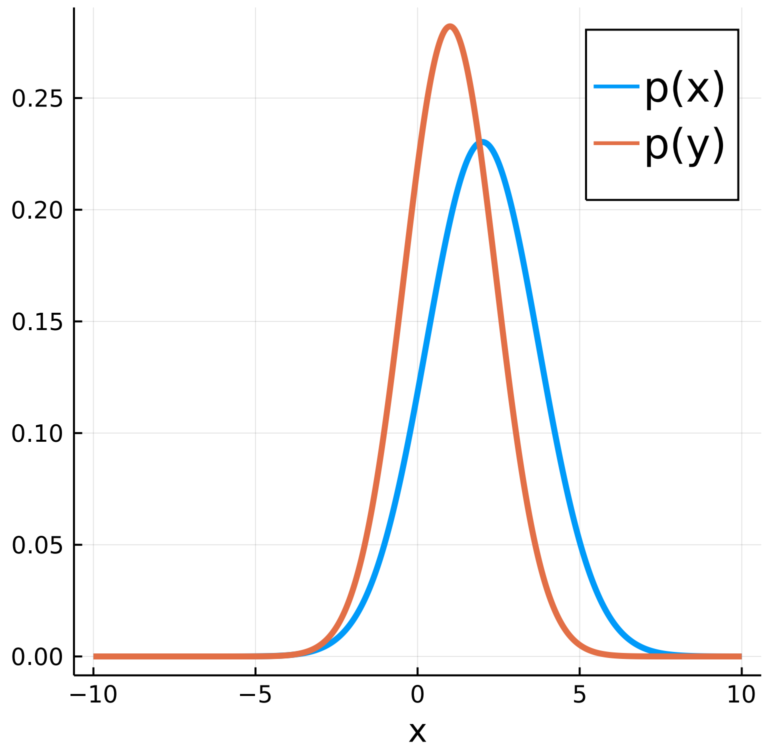
Full covariance ? ☐

E[x] = 2.0 Var[x] = 3.0

E[y] = 1.0 Var[y] = 2.0

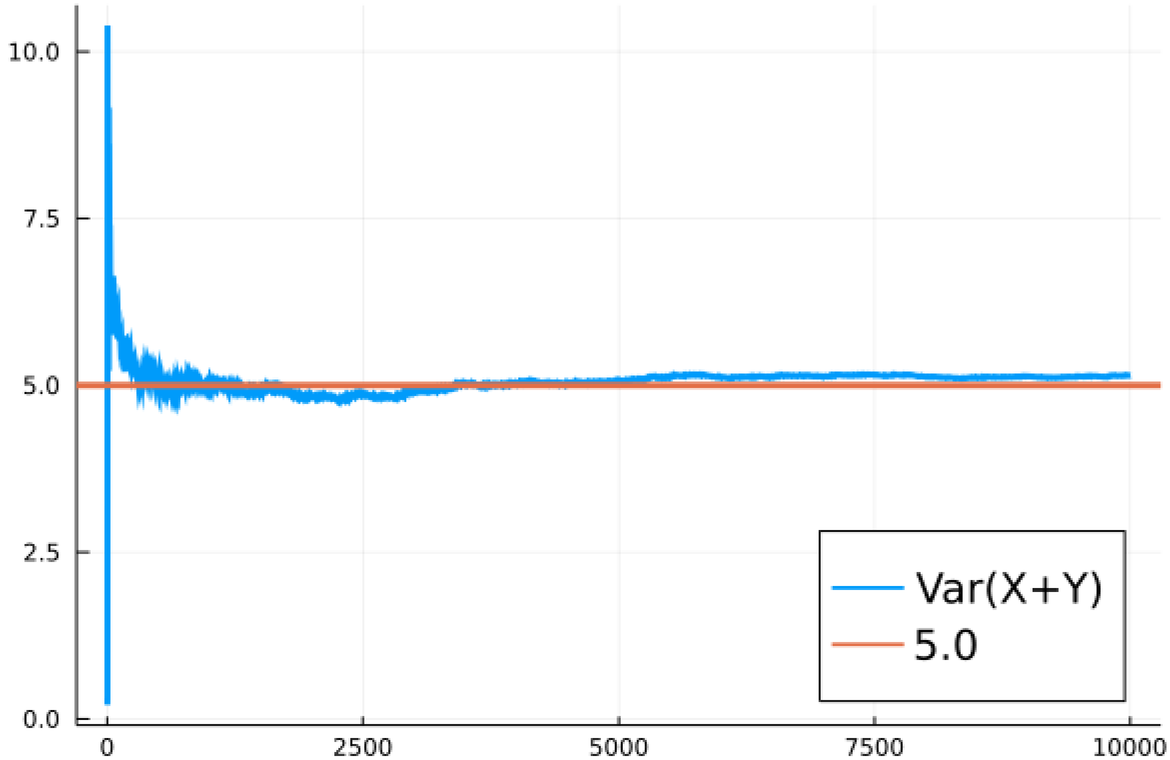
```
dist_x = Distributions.Normal{Float64}(μ=2.0, σ=1.7320508075688772)
```

```
dist_y = Distributions.Normal{Float64}(μ=1.0, σ=1.4142135623730951)
```




```
begin
    nSamples = 10000; # Number of samples we use
    # Preallocation
    xs = zeros(nSamples)
    ys = zeros(nSamples)
    vars = zeros(nSamples)
    for i in 1:nSamples
        if fullcov
            xs[i], ys[i] = rand(dist_xy)
        else
            xs[i] = rand(dist_x)
            ys[i] = rand(dist_y)
        end
        vars[i] = var(xs[1:i] .+ ys[1:i])
    end
end
```

Full covariance? ☐



3. Transformation of probability densities

Let X be uniformly distributed in $(0,1)$:

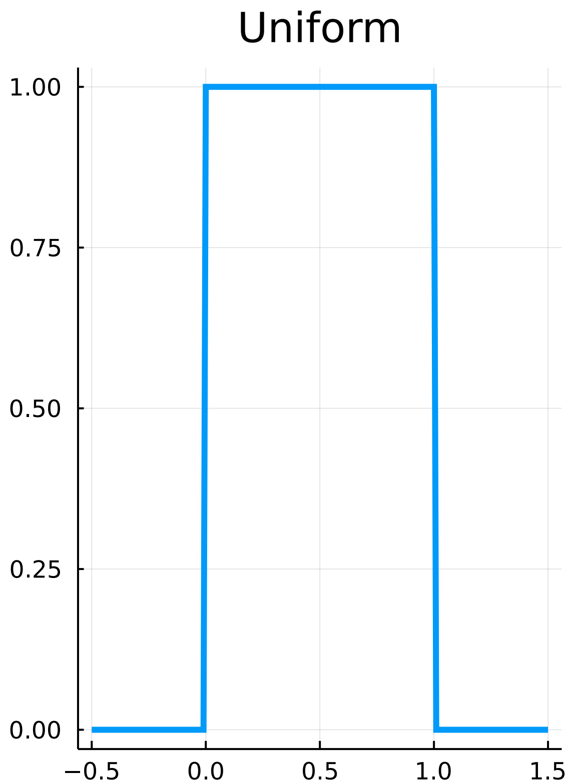
$$p(x) = \begin{cases} 1 & \text{for } 0 < x < 1, \\ 0 & \text{otherwise.} \end{cases}$$

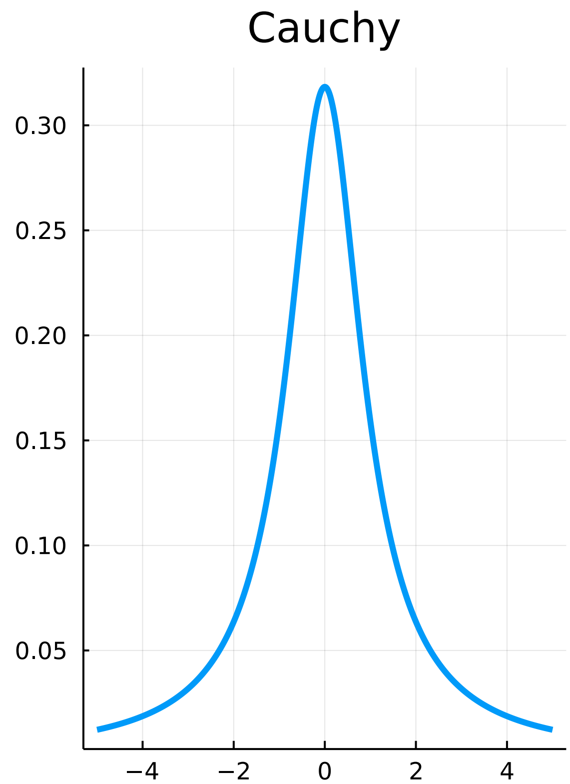
A second random variable Y is defined as

$$Y = \tan\left(\pi(X - 1/2)\right).$$

What is the probability density $q(y)$ of Y?

Write your answer here or on paper





4. Gaussian Inference

Suppose we have two random variables V_1 and V_2 which are **jointly Gaussian** distributed with zero means $E[V_1] = E[V_2] = 0$ and variances $E[V_1^2] = 16.6$ and $E[V_2^2] = 6.8$. The covariance is $E[V_1\ V_2] = 6.4$.

Assume that we observe a noisy estimate $Y = V_2 + \nu$ of V_2 where ν is a Gaussian noise variable independent of V_1 and of V_2 with $E[\nu] = 0$ and $E[\nu^2] = 1$.

Tip

The following formula could be helpful: The inverse of the matrix

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

is given by

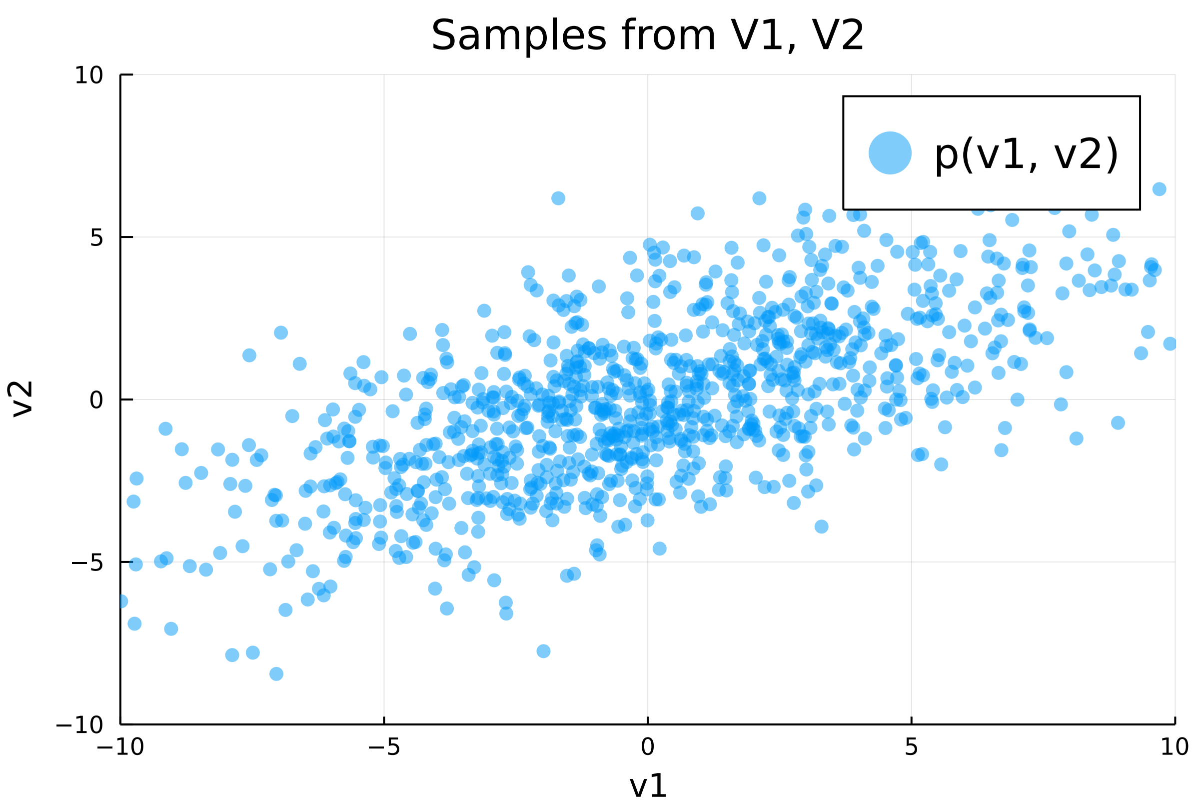
$$\mathbf{A}^{-1} = \frac{1}{\det \mathbf{A}} \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix}$$

(a) Obtain the conditional densities $p(V|Y)$ from the joint densities $p(V, Y)$. (Here V can be either V_1 or V_2) !

Write your answers here or on paper

(b) What are the posterior mean predictions of V_1 and V_2 for an observation $Y = 1$ and what are the posterior uncertainties of these predictions.

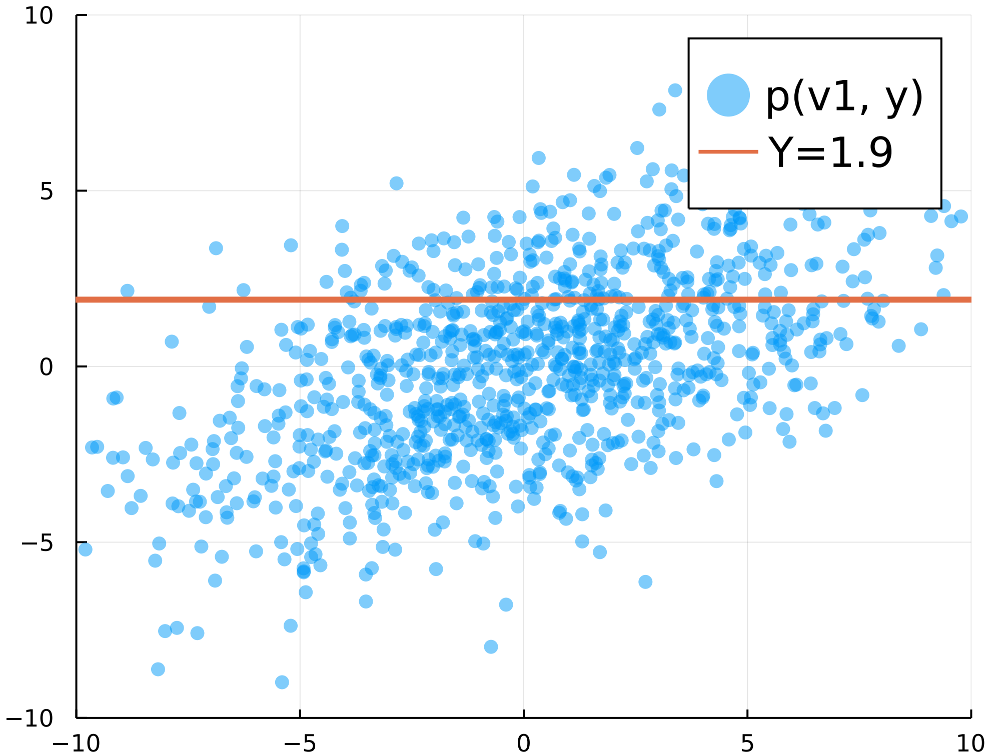
Write your answers here or on paper

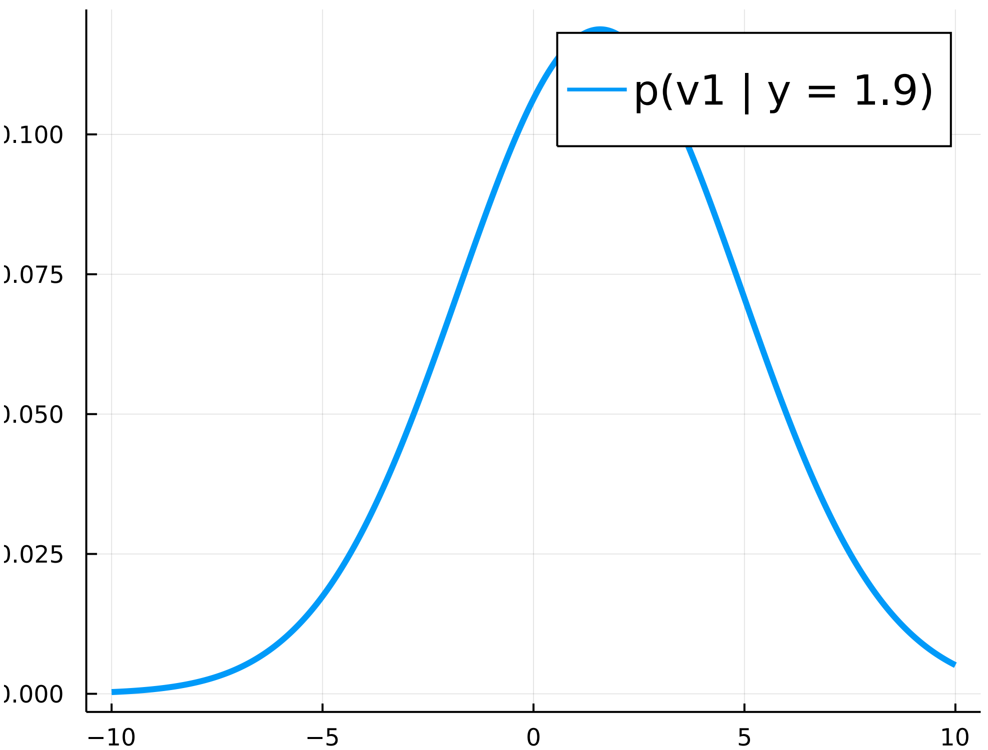


v = 0.9

y = 1.9

```
ZeroMeanFullNormal(
dim: 2
μ: 2-element Zeros{Float64}
Σ: [16.6 6.4; 6.4 7.7]
)
```





5. Maximum Likelihood

- **(a) How can you use the results of problem 3 to generate a dataset of $n = 1000$ independent random numbers $D = (x_1, \ldots, x_n)$ from a Cauchy density**

$$p(x|\theta) = \frac{1}{\pi} \frac{1}{1 + (x - \theta)^2}$$

when $\theta \neq 0$.

Write your answers here or on paper

- **(b) Write down an expression for the log–likelihood $\ln p(D|\theta)$ for independent Cauchy data.**

Write your answers here or on paper

- **(c) Set $\theta = 1$, generate a Cauchy dataset D and use numerical optimisation to find the maximum likelihood estimator $\hat{\theta}_{ML}(D)$.**

```julia
# Generate a dataset D of Cauchy variables
function generate_D(N, θ)
    ## !! CODE MISSING !! ##
    ## The function should return a vector of random Cauchy variables of size N
    ## !! CODE MISSING !! ##
end;
```

θ = 1.0

```julia
D = generate_D(1000, θ) # Generate the dataset;
```

```julia
function log_likelihood(ys, θ) # Compute the loglikelihood
    ## !! CODE MISSING !! ##
    ## The function should return the total loglikelihood for the observations ys given the parameter θ
    ## !! CODE MISSING !! ##
end;
```

```
# We call optimize, from Optim.jl. Since we want to maximize
# but optimize minimizes we give the negative value
if D !== nothing
    θML = optimize(x -> -log_likelihood(D, first(x)), [0.5], BFGS()).minimizer[1]
end
```

- **(d) Repeat the estimation for $M = 100$ independent data sets $(D_1, \ldots, D_{100})$ and report the empirical mean and variance of the ML estimators.**

```
begin
    N = 1000 # Size of dataset
    M = 100 # Number of tries
end;
```

```
begin
    if D !== nothing
        θs_ML = [ # Repeat the ML estimator M times
            begin # This is a comprehension
                ## !! CODE MISSING !! ##
                ## Write here a function generating a random dataset and evaluating the ML estimator
                ## !! CODE MISSING !! ##
            end
        for _ in 1:M]
        (mean = mean(θs_ML), variance = var(θs_ML))
    end
end;
```

UndefVarError: θs_ML not defined

1. top-level scope @ *Local: 2*

```
begin
    histogram(θs_ML; title="Histogram of estimators", bins=20, lw=0.0, label="", xlabel="θₘₗ")
    vline!([θ], label="True value")
end
```

- **(e) Report mean and variance of a naive estimator $\hat{\theta}_{naive}(D) \doteq \frac{1}{n}\sum_{i=1}^{n} x_i$ on the same datasets.**

```
begin
    N_naive = 10000
    M_naive = 10000
    if D !== nothing
        θs_naive = map(1:M) do _
            ## !! CODE MISSING !! ##
```

```
•              ## Fill in here code generating a random dataset and evaluating the naive
  estimator
•              ## !! CODE MISSING !! ##
•          end
•          (mean = mean(θs_naive), variance = var(θs_naive))
•      end
• end
```

UndefVarError: θs_naive not defined

1. **top-level scope** @ *Local: 2*

```
• begin
•     histogram(θs_naive; title="Histogram of estimators", bins=20, lw=0.0, label="",
  xlabel="θₘₗ")
•     vline!([θ], label="True value")
• end
```