Problem Sheet 1- With Solutions

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
    using Random
end
```

Present

Table of Contents

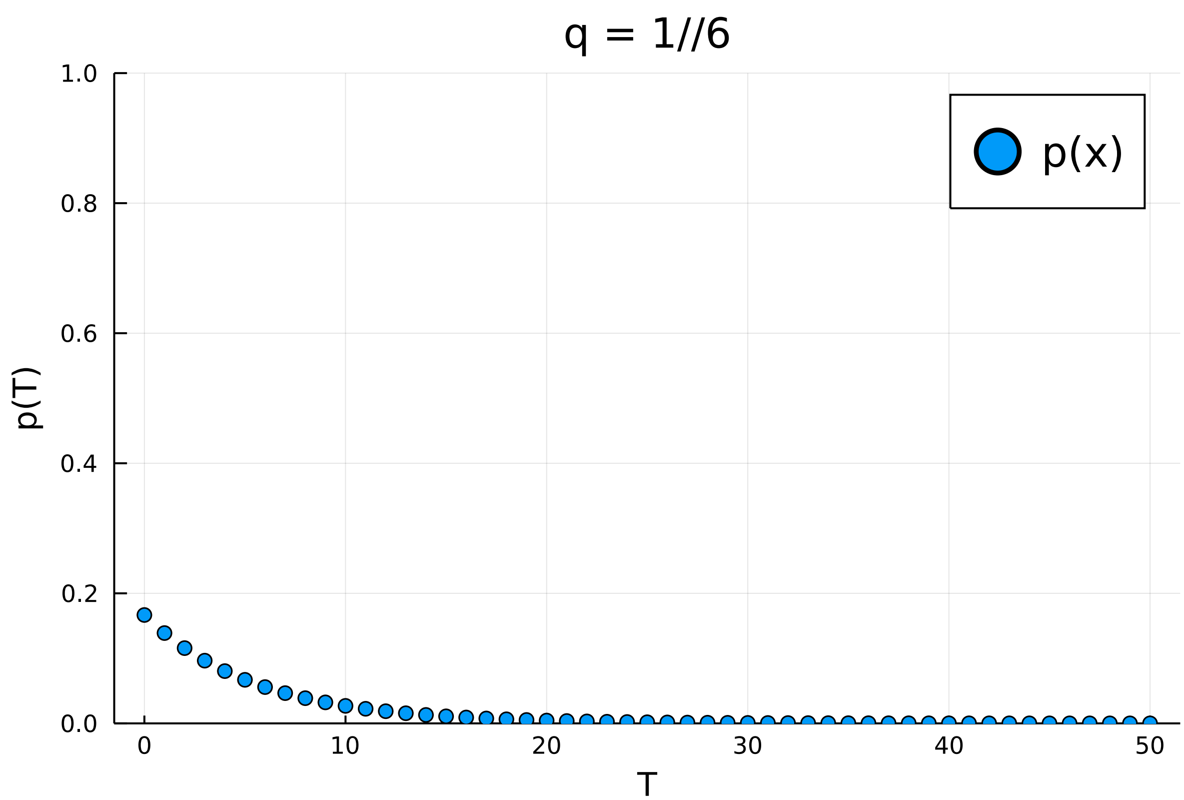
1. Random experiments

A dice is thrown repeatedly until it shows a 6. Let T be the number of throws for this to happen. Obviously, T is a random variable.

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

0.16666666666666667

```
q = 1//6
```



- The probability for t throws is given by the geometric distribution

$$P(T=t) = (1-q)q^{t-1}$$

with parameter $q = 5/6$.

- The expectation value of T can be calculated using its definition:

$$E[T] = \sum_{t=1}^{\infty} tP(T=t) = \sum_{t=1}^{\infty} (1-q)tq^{t-1} = \sum_{t=1}^{\infty} (1-q)\frac{d}{dq}q^t$$

- As the geometric series converges absolutely, we can exchange summation and derivation:

$$E[T] = (1-q)\frac{d}{dq}\sum_{t=0}^{\infty} q^t = (1-q)\frac{d}{dq}\frac{1}{1-q} = (1-q)\frac{1}{(1-q)^2} = \frac{1}{1-q}$$

- In order to obtain the variance we need the expectation value of T^2, too:

$$E[T^2] = \sum_{t=1}^{\infty} t^2\, P(T=t) = \sum_{t=1}^{\infty} (1-q)\, t^2\, q^{t-1}$$

- Here $t^2\, q^{t-1}$ is very similar to the second derivative of q^{t+1}:

$$E[T^2] = \sum_{t=1}^{\infty} (1-q)\, t(t+1) q^{t-1} - \sum_{t=1}^{\infty} (1-q)tq^{t-1} = -E[T] + \sum_{t=1}^{\infty} (1-q)\frac{d^2}{dq^2}\, q^{t+1}$$

- Further simplifications

$$E[T^2] = -\frac{1}{1-q} + (1-q)\frac{d^2}{dq^2}\sum_{t=0}^{\infty} q^t = -\frac{1}{1-q} + (1-q)\frac{d^2}{dq^2}\frac{1}{1-q}$$

lead to

$$E[T^2] = -\frac{1}{1-q} + (1-q)\frac{2}{(1-q)^3} = \frac{1+q}{(1-q)^2}$$

so that the variance of T is given by

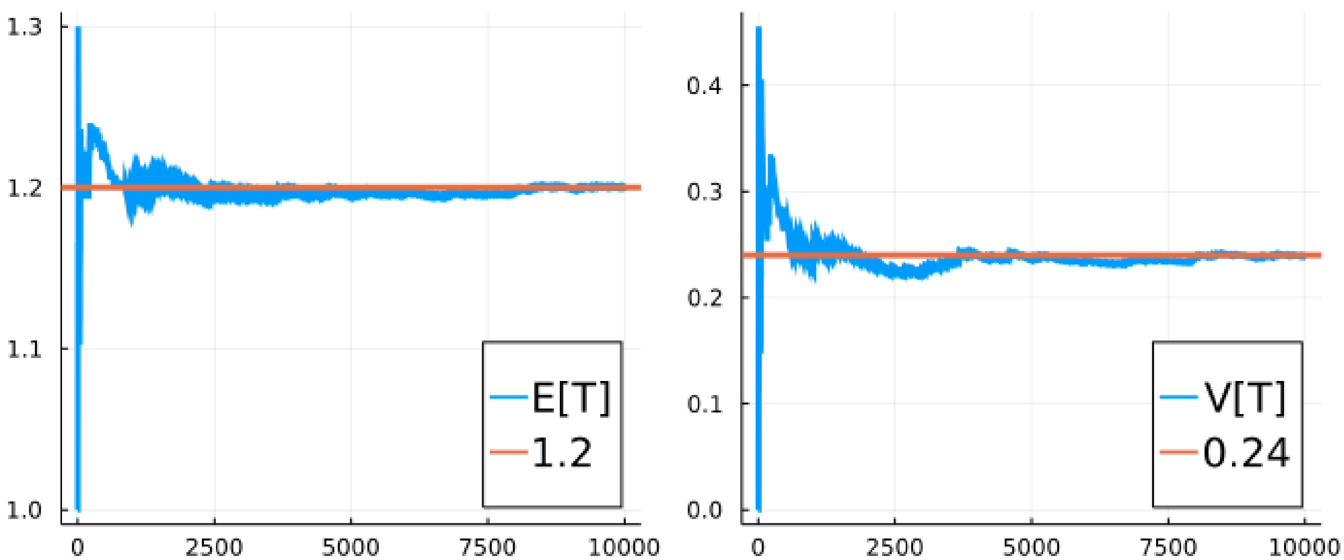
$$V[T] = E[T^2] - E[T]^2 = \frac{1+q}{(1-q)^2} - \frac{1}{(1-q)^2} = \frac{q}{(1-q)^2}$$

- By substituting $q = 5/6$ we finally find $E[T] = 6$ and $V[T] = 30$.

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

0.8333333333333333

```
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
        while !rand(Bernoulli(1-q)) || T > 10000 # Sample from a Bernoulli with prob q until we get a 6
            T += 1
        end
        T_vals[i] = T
        expec_T[i] = mean(T_vals[1:i])
        var_T[i] = var(T_vals[1:i])
    end
end;
```




```
plot(p1, p2, size = (700,300))
```

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 :

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

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

$$
\begin{aligned}
\mathrm{Var}(X+Y) =& E[(X+Y-E[(X+Y)])^2] = E[(X-E[X]+Y-E[Y])^2] \\
=& E[(X-E[X])^2] + 2E[(X-E[X])(Y-E[X])] + E[(Y-E[Y])^2] \\
=& \mathrm{Var}(X) + \mathrm{Var}(Y) + 2(E[XY] - 2E[X]E[Y] + E[X]E[Y]) \\
=& \mathrm{Var}(X) + \mathrm{Var}(Y)
\end{aligned}
$$

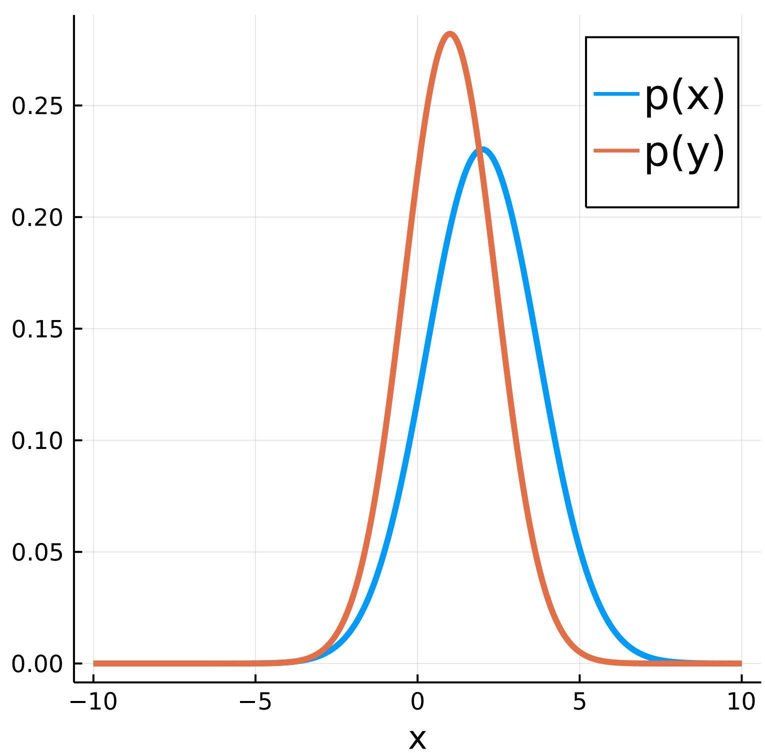
Full covariance ? ☐

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)
```

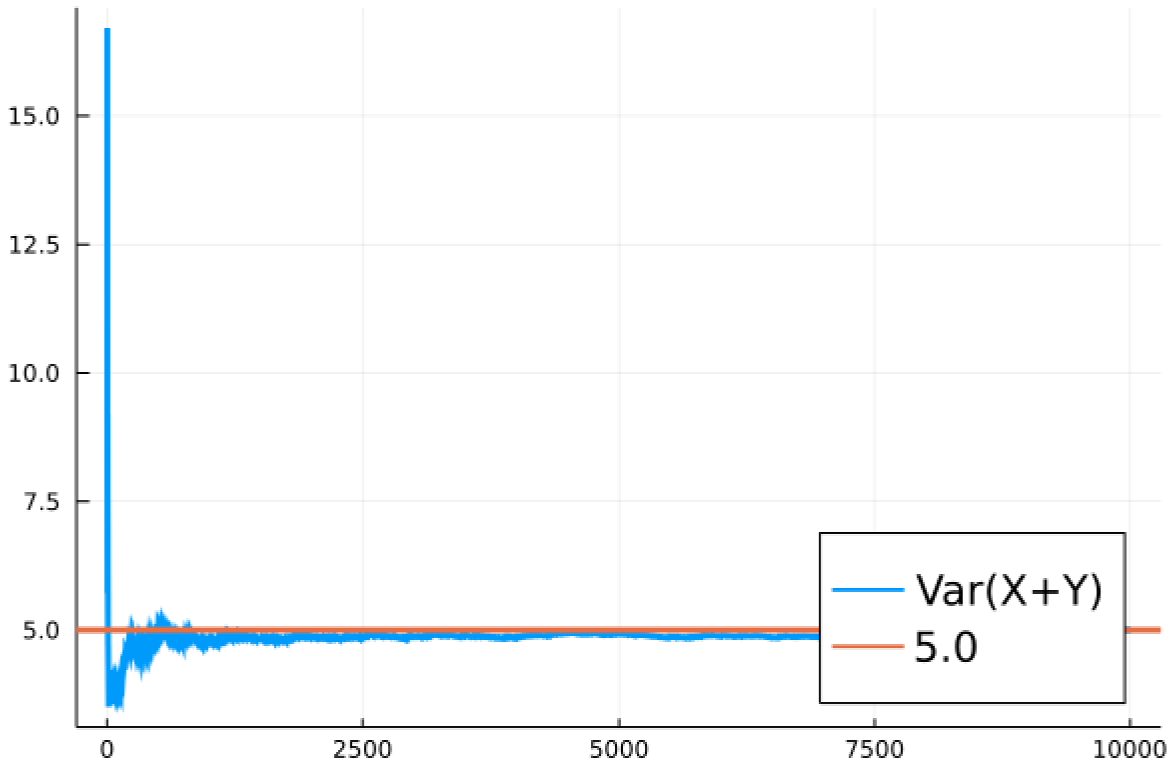
```
dist_y = Distributions.Normal{Float64}(μ=1.0, σ=1.4142135623730951)
```




```
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?

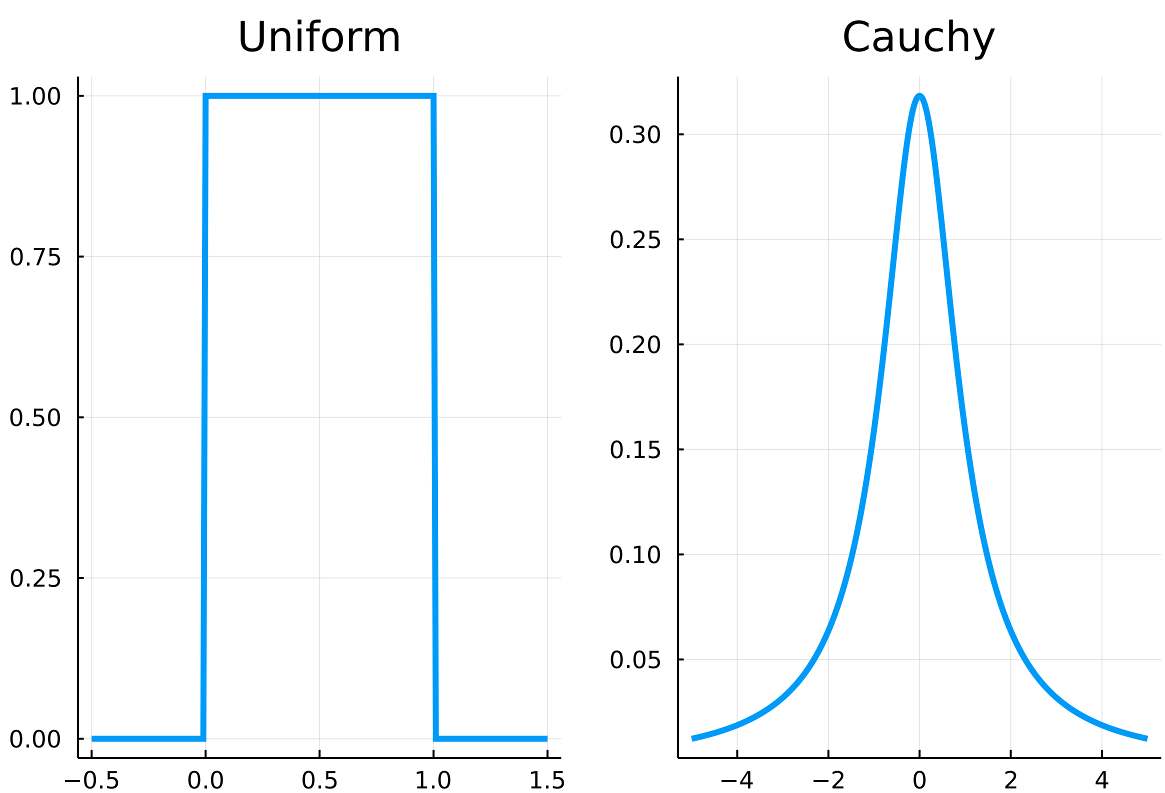
- Inverse function:

$$\begin{aligned} y = \tan(\pi(x - 1/2)) &\iff \arctan y = \pi(x - 1/2) \\ &\iff x = \frac{1}{\pi}\arctan y + \frac{1}{2} \end{aligned}$$

- Transformation of probability densities:

$$q(y) = p(x) \cdot \frac{dx}{dy} = p(x) \cdot \frac{1}{\pi}\frac{1}{1+y^2} = \frac{1}{\pi}\frac{1}{1+y^2}$$

- This transformation together with a (pseudo-)random number generator can be used to generate (pseudo-)random numbers with a standard Cauchy distribution.



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 $E[\nu^2] = 1$.

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

$$p(V,Y) = \frac{1}{2\pi\sqrt{\det(\mathbf{S})}} \exp\left\{-\frac{1}{2}(V,\ Y)^\top \mathbf{S}^{-1}(V,\ Y)\right\}$$

Note $(V,\ Y)$ is a two dimensional vector and the covariance matrix is given by

$$\mathbf{S} = \begin{pmatrix} E[V^2] & E[VY] \\ E[VY] & E[Y^2] \end{pmatrix}$$

The expectations are

$$\begin{aligned} E[V_1Y] &= E[V_1V_2] \\ E[V_2Y] &= E[V_2^2] \\ E[Y^2] &= E[V_2^2] + E[\nu^2] \end{aligned}$$

We set

$$\mathbf{S}^{-1} = \begin{pmatrix} (\mathbf{S}^{-1})_{vv} & (\mathbf{S}^{-1})_{vy} \\ (\mathbf{S}^{-1})_{vy} & (\mathbf{S}^{-1})_{yy} \end{pmatrix}$$

Then, from the joint density, we can write the conditional density as

$$p(V|Y) \propto \exp\left(-\frac{V^2}{2}(\mathbf{S}^{-1})_{vv} - V(\mathbf{S}^{-1})_{vy}Y\right)$$

- This can be written in the standard notation as

$$p(V|Y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(V-\mu)^2}{2\sigma^2}}$$

where

$$\mu = E[V|Y] = -\frac{(\mathbf{S}^{-1})_{vy}Y}{(\mathbf{S}^{-1})_{vv}}$$
$$\sigma^2 = \mathrm{VAR}[V|Y] = \frac{1}{(\mathbf{S}^{-1})_{vv}}$$

are the conditional mean and variance. We can use $E[V|Y]$ for prediction. $\mathrm{VAR}[V|Y]$ would give us a measure for the error of such a prediction.

(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.

- For $p(V_1|Y)$ we have

$$\mathbf{S} = \begin{pmatrix} 16.6 & 6.4 \\ 6.4 & 7.8 \end{pmatrix}$$

and

$$\mathbf{S}^{-1} = \begin{pmatrix} 0.0881 & -0.0723 \\ -0.0723 & 0.1875 \end{pmatrix}$$

Hence $E[V_1|Y] = 0.8207$ and $\mathrm{VAR}[V_1|Y] = 11.3507$.

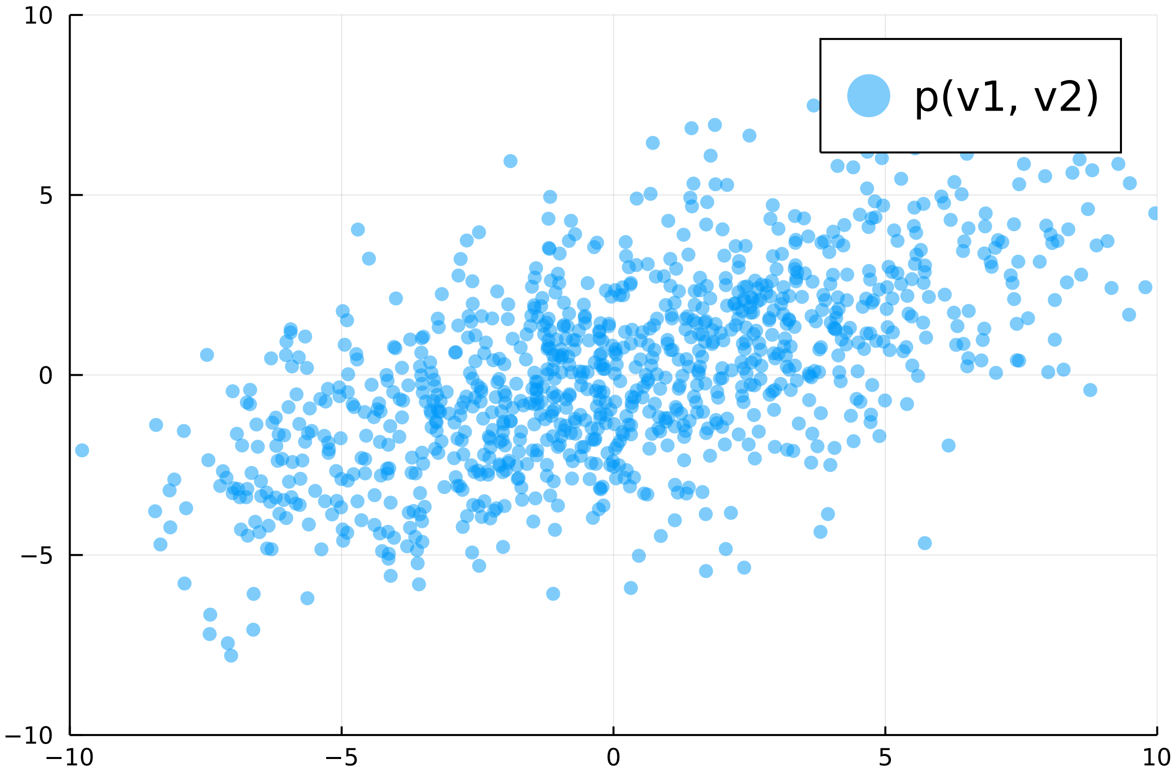
- For $p(V_2|Y)$ we have

$$\mathbf{S} = \begin{pmatrix} 6.8 & 6.8 \\ 6.8 & 7.8 \end{pmatrix}$$

and

$$\mathbf{S}^{-1} = \begin{pmatrix} 1.1471 & -1.0000 \\ -1.0000 & 1.0000 \end{pmatrix}$$

Hence $E[V_2|Y] = 0.8718$ and $\mathrm{VAR}[V_2|Y] = 0.8718$.



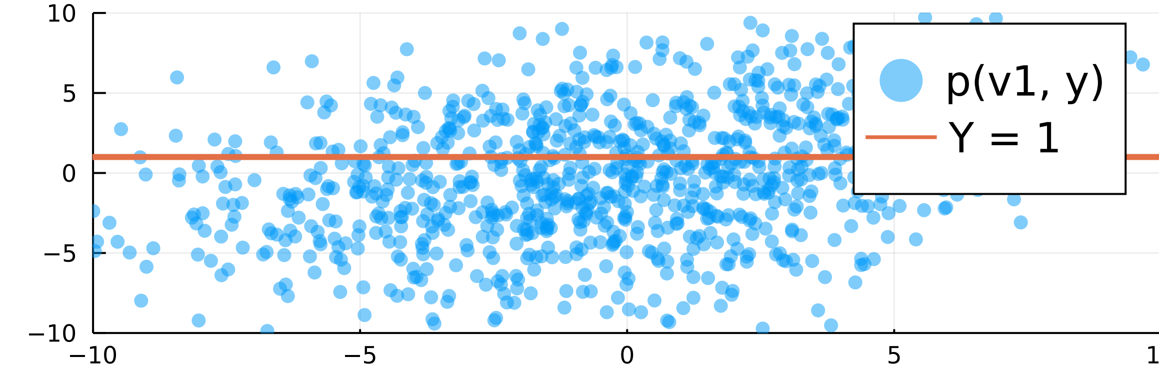

```
begin
    lim = 10.0
    S_V1V2 = [16.6 6.4
              6.4 6.8]
    dV1V2 = MvNormal(S_V1V2)
    scatter(eachrow(rand(dV1V2, 1000))..., msw = 0.0, alpha = 0.5, lab = "p(v1, v2)",
xlims = (-lim, lim), ylims = (-lim, lim))
end
```

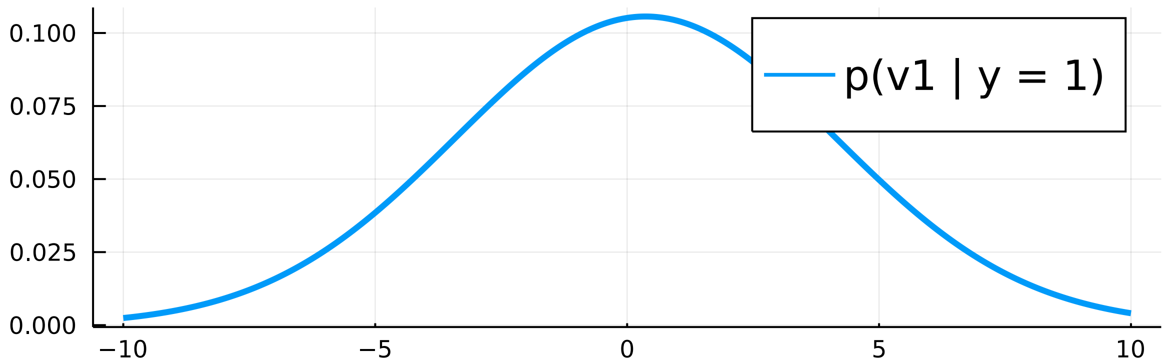
ν = 0.7

```
ZeroMeanFullNormal(
dim: 2
μ: 2-element Zeros{Float64}
Σ: [16.6 6.4; 6.4 17.5]
)
```

```
begin
    S_V1Y = [16.6 6.4
             6.4 16.8 + ν]
    dV1Y = MvNormal(S_V1Y)
end
```

y = 1





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 $\theta \neq 0$.

One can redo the same derivation by adding θ. This will lead to

$$Y = \theta + \tan(\pi(X - \frac{1}{2}))$$

One can generate uniform samples, using for instance a pseudo-random generator `rand()` in most programming languages. Then applying the transform from problem 3

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

The log likelihood for a dataset of N independent points y_i drawn from a cauchy distribution is given by :

$$\log p(D|\theta) = \sum_{i=1}^{N} \log p(y_i|\theta) = -N \log \pi - \sum \log(1 + (y_i - \theta)^2)$$

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

```
# Generate a dataset D of Cauchy variables
function generate_D(N, θ)
    u = rand(N)
    ys = θ .- tan.(π * (u .- 0.5))
end;
```

θ = 1.0

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

```
function log_likelihood(ys, θ) # Compute the loglikelihood
    - length(ys) * log(π) - sum(log(1.0 + (y - θ)^2) for y in ys)
end;
```

θML = 1.033558382065018

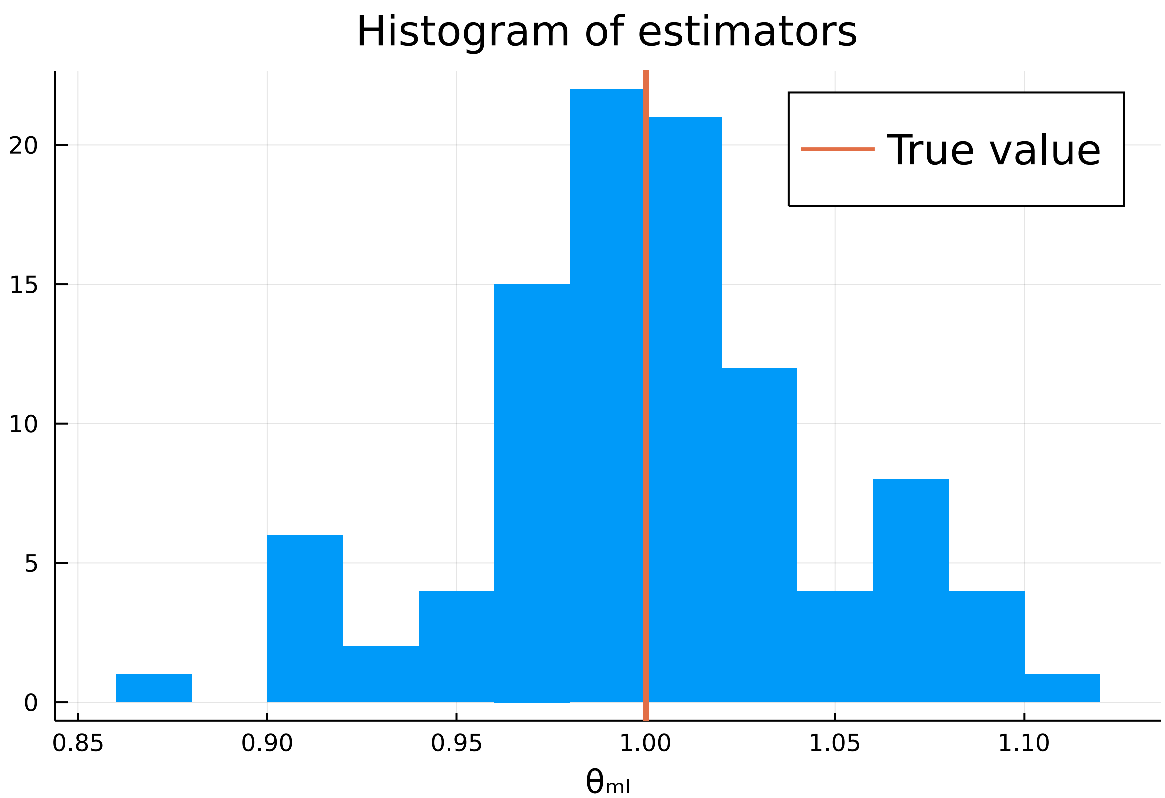
```
# We call optimize, from Optim.jl. Since we want to maximize
# but optimize minimizes we give the negative value
θML = optimize(x -> -log_likelihood(D, first(x)), [0.5], BFGS()).minimizer[1]
```

- ### (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
    θs_ML = [ # Repeat the ML estimator M times
        begin # This is a comprehension
            ys = generate_D(N, θ)
            optimize(x -> -log_likelihood(ys, first(x)), [0.0],
BFGS()).minimizer[1]
        end
    for _ in 1:M]
    (mean = mean(θs_ML), variance = var(θs_ML))
end;
```




```
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.**

```
(mean = -0.566375,  variance = 274.239)
```

```
begin
    N_naive = 10000
    M_naive = 10000
    θs_naive = map(1:M) do _
```

```
        ys = generate_D(1000, θ)
        return sum(ys)/N
    end
    (mean = mean(θs_naive), variance = var(θs_naive))
end
```