Note:

- Submit a hard copy with a signed, QR-coded coversheet to the SRC and a soft copy of your R code (as Assignment 2) on Canvas. **Ideally**, the soft copy is a .Rmd file (or similar) and the hard copy is produced by "knitting" the .Rmd file (or similar).
- Include everything in the hard copy: R code (tidied up), outputs (including error/warning messages), and your explanations (if any).
- Print some intermediate results to show how your code works step by step, if not obvious.
- Comment your code wherever appropriate, e.g., for functions, blocks of code, and key variables.

1. [10 marks]

This question works with a data set on sodium intake. We can read it into R with the following code ...

```
> sodium <- read.table("sodium.txt", header=TRUE)
```

... and here are the first few rows of data ...

```
> head(sodium)
     Instructor Supplement Sodium
1 Brendon Small          A   1200
2 Brendon Small          A   1400
3 Brendon Small          A   1350
4 Brendon Small          A    950
5 Brendon Small          A   1400
6 Brendon Small          B   1150
```

The `Instructor` is a nutrition advisor and `Supplement` is a nutritional supplement.

Extract just the first observation for each combination of `Instructor` and `Supplement` and create a matrix of the result.

```
> sodiumMat
                Supplement
Instructor          A    B    C    D
  Brendon Small  1200 1150 1250 1300
  Coach McGuirk  1100 1250 1225 1200
  Melissa Robins  900 1150 1125 1100
```

Use `apply` and `sweep` to fit a model of the form ...

$$y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij}$$

... to these data.

```
                 Supplement
Instructor                A         B      C      D
  Brendon Small   70.833333 -95.83333 -12.50  37.50
  Coach McGuirk    2.083333  35.41667  -6.25 -31.25
  Melissa Robins -72.916667  60.41667  18.75  -6.25
```

Does it look like this is an appropriate model ?

2. [10 marks]

This question works with a set of plant weights, measured under two experimental conditions.

```
> ## Annette Dobson (1990) "An Introduction to Generalized Linear Models".
> ## Page 9: Plant Weight Data.
> ## Control = standard conditions
> ## Treatment = nutrient rich
> ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
> trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
> group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
> weight <- c(ctl, trt)
```

We will first assume that all weights are i.i.d. Normal(μ, σ).

We will further assume that σ is the sample standard deviation.

```
> sigma
[1] 0.7040281
```

We are going to estimate the mean, μ, for the plant weights using Maximum Likelihood.

The likelihood function is

$$\prod_{i=1}^{n} f(x_i; \mu)$$

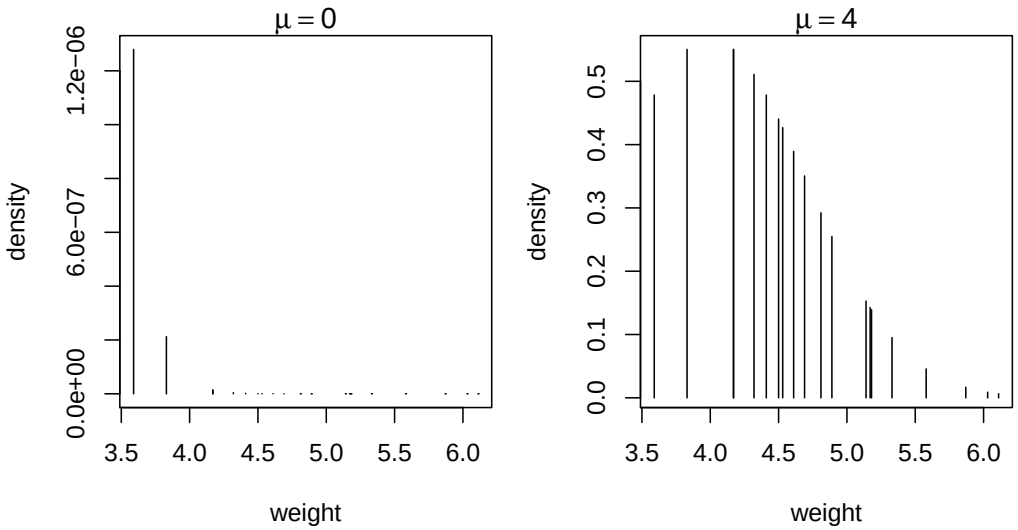
where $f(x_i; \mu)$ is the Normal probability density function

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

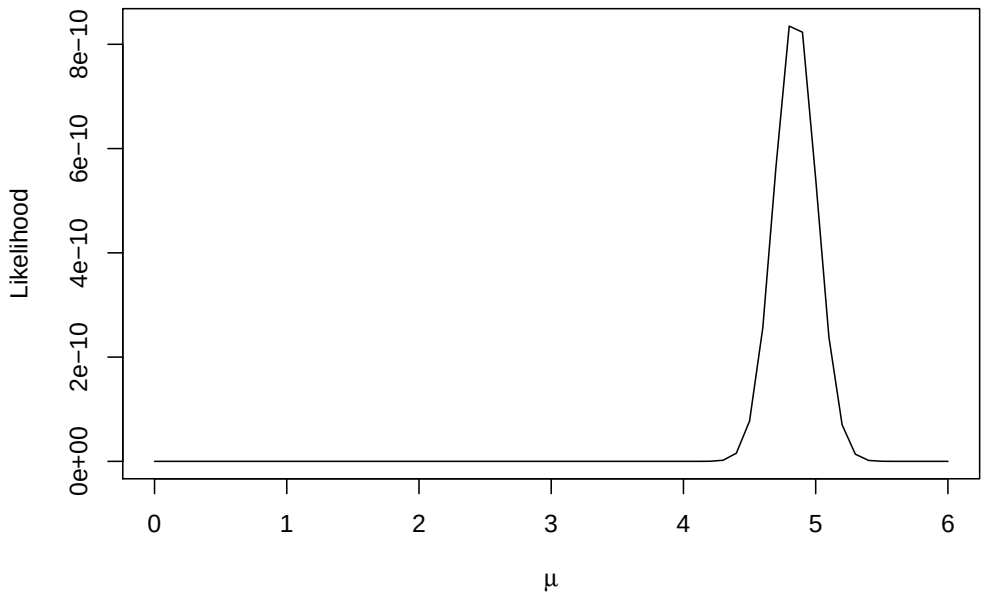
Write an R function called `like` that calculates this likelihood, given a set of data `x` and a mean `mu` (**Hint:** the R function `dnorm` evaluates the Normal probability density given `x`, `mu`, and `sigma`).

```
> like(weight, 0)
[1] 1.359239e-215
> like(weight, 4)
[1] 4.592718e-16
```

Draw a plot of the probability densities for each weight value for both $\mu = 0$ and $\mu = 4$ (and σ equal to the sample standard deviation).



Draw a plot of the likelihood function for μ varying from 0 to 6.



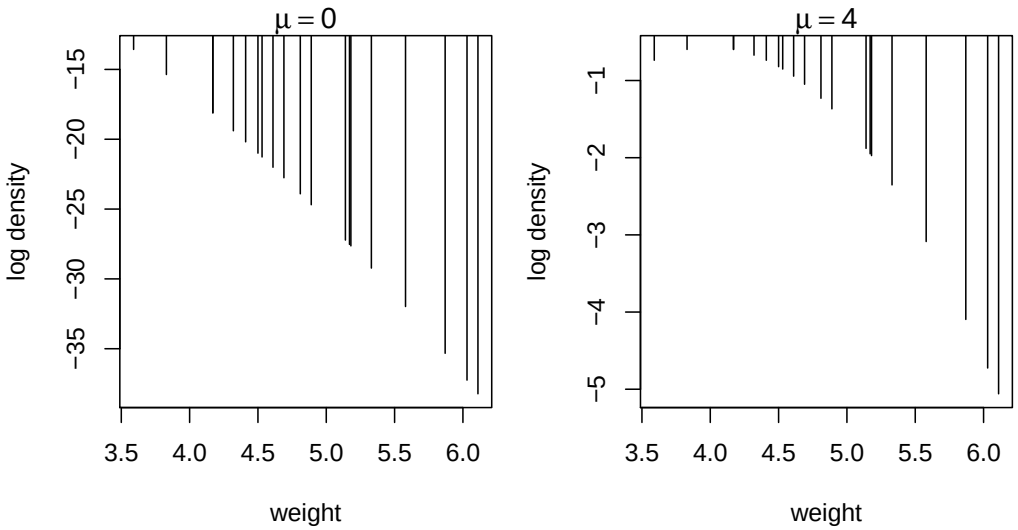
The log-likelihood is

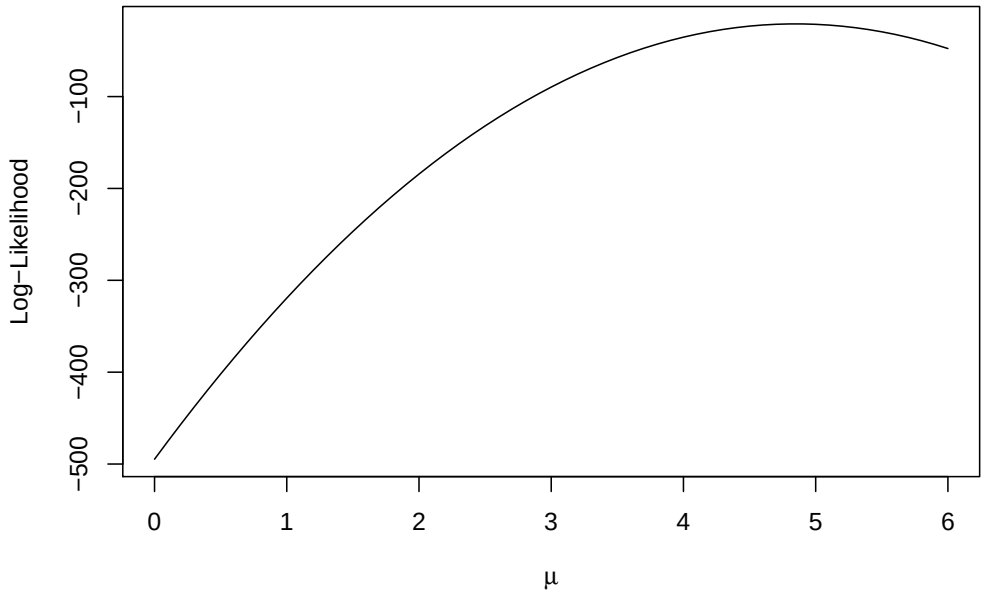
$$\sum_{i=1}^{n} \log(f(x_i; \mu))$$

Write a function `loglike` to calculate the log-likelihood (**Hint:** the `dnorm` function has an argument `log`).

```
> loglike(weight, 0)
[1] -494.7489
> loglike(weight, 4)
[1] -35.31689
```

Plot log probability densities for the weight data given $\mu = 0$ and $\mu = 4$ and plot the log-likelihood curve for μ between 0 and 6.





Use the `optimise` function to find the maximum likelihood estimate of μ (find the value of μ that maximises the log-likelihood function).

```
> muMLE
[1] 4.8465
```

This should equal the sample mean.

```
> mean(weight)
[1] 4.8465
```

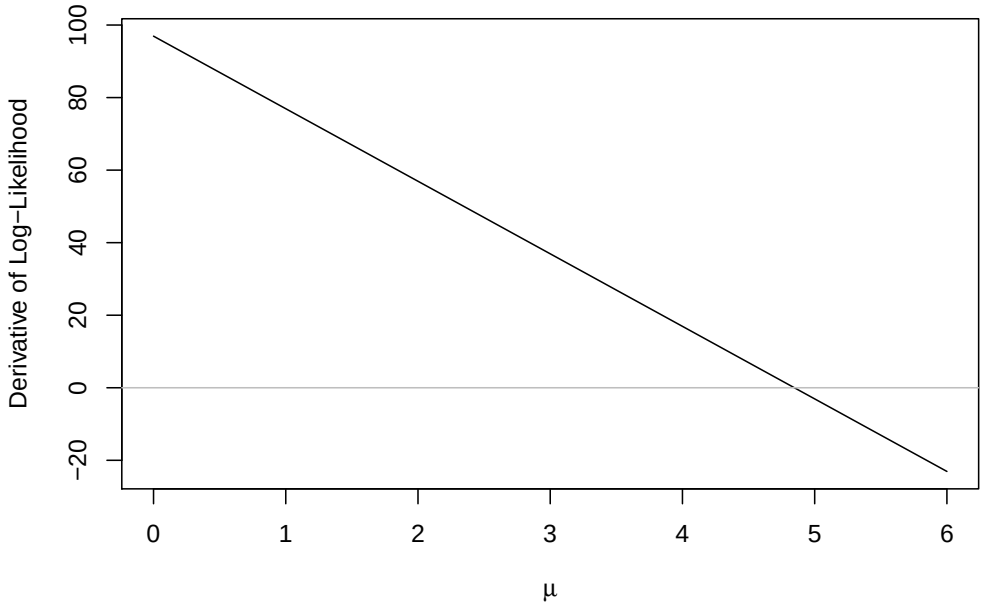
NOTE: using maximum likelihood estimation is NOT how we would normally estimate this parameter (or the parameters in the next two questions), but it can be a useful exercise to help understand how maximum likelihood works.

The first derivative of the log-likelihood function (w.r.t. μ, assuming σ known constant, and data x fixed) is

$$\text{constant} * \sum_{i=1}^{n} x_i - \mu$$

Write a function `dllike` that calculates this first derivative and use `uniroot` to find where this function is zero (a plot of the function is shown below). This should produce the same answer as above.

```
[1] 4.8465
```



3. [10 marks]

This question also works with the set of plant weights and we will still assume that all weights are i.i.d. Normal(μ, σ).

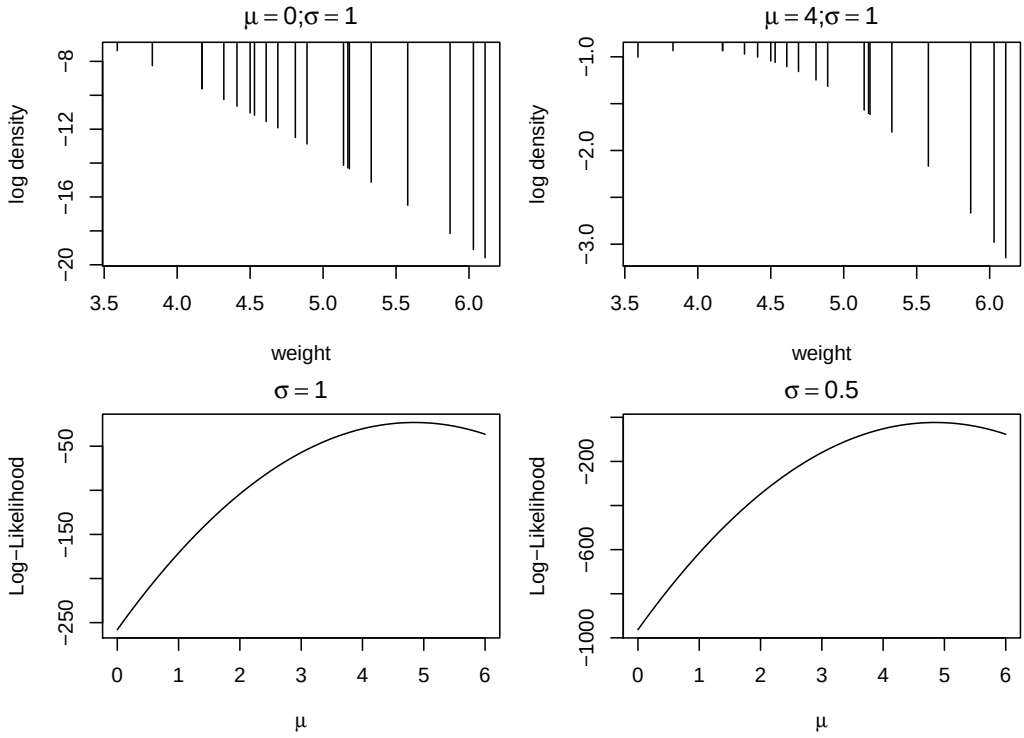
However, we will now estimate both μ and σ using maximum likelihood.

The log-likelihood function is now

$$\sum_{i=1}^{n} \log(f(x_i; \mu, \sigma))$$

Write a function `loglike2` to evaluate the log-likelihood, plot probability densities values for the weight data for both $\mu = 0; \sigma = 1$ and $\mu = 4, \sigma = 1$, and plot the log-likelihood function for μ between 0 and 6, with $\sigma = 1$ and with $\sigma = .5$.

```
> loglike2(weight, 0, 1)
[1] -257.9731
> loglike2(weight, 4, 1)
[1] -30.25312
```



Use the `optim` function to find the maximum likelihood estimates for μ and σ.

These should correspond to the sample mean and (almost) the sample standard deviation.

```
> muSigmaMLE
[1] 4.846455 0.686466
```

```
> mean(weight)
[1] 4.8465
> sd(weight)
[1] 0.7040281
> sd(weight)*sqrt((length(weight) - 1)/length(weight))
[1] 0.6862017
```

4. [20 marks]

This question also works with the set of plant weights, but now we will allow there to be a separate mean for the treatment and control groups.

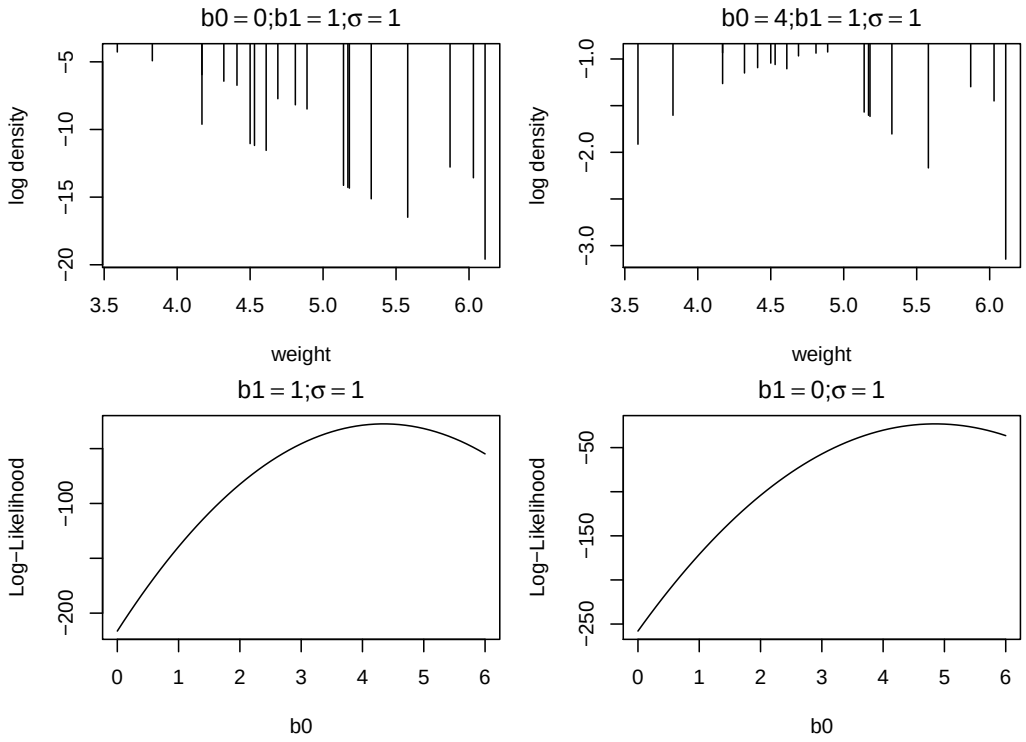
The log-likelihood function now looks like this

$$\sum_{i=1}^{n} \log(f(x_i; \beta_0 + g * \beta_1, \sigma)$$

where g is 0 for control weights and 1 for treatment weights.

Find the maximum likelihood estimates for β_0, β_1, and σ.

```
> gp <- as.numeric(group) - 1
> loglike3(weight, gp, 0, 1, 1)
[1] -216.3631
> loglike3(weight, gp, 4, 1, 1)
[1] -28.64312
```




```
> params
[1]  5.0319929 -0.3709581  0.6604996
```

The corresponding answer from `lm` is shown below.

```
> lm.D9 <- lm(weight ~ group)
> coef(lm.D9)
(Intercept)    groupTrt
      5.032      -0.371
```