Exercise 1. You know that $R_0$ for a simple SIR epidemic is given by $R_0 = \frac{\beta}{\gamma}$. Calculate $R_0$ for an epidemic where $\beta = 1.2$ and $\gamma = 0.6$.

```r
> beta <- 1.2
> gamma <- 0.6
> beta/gamma
[1] 2
```

Exercise 2. Create two vectors: (1) eight different measurements of $\beta$, (2) eight different measurements of $\gamma$. Obtain a vector of $R_0$ by element-wise division of these two vectors.

```r
> beta <- c(1, 3.2, 4.5, 1.05, 1.8, 2.02, 2.8, 3.1)
> gamma <- c(0.8, 3.4, 3.2, 1.04, 1.2, 2, 1.2, 1.5)
> beta/gamma
[1] 1.2500000 0.9411765 1.4062500 1.0096154 1.5000000 1.0100000 2.3333333 2.0666667
```

Exercise 3. Look at the help for the function `lm` and answer the following questions.

- What are the function inputs called in R terminology?
- What are the function outputs called in R terminology?
- What else does the help file tell you about `lm`?

The inputs are called *arguments*.
The outputs are called *value*.
The help file also tells you:

- What the function is for (fitting linear models)
• The format the arguments are expected to take
• Optional specification for interactions
• Optional fitting of weighted models
• How to use this function with time series data
• The authors and provenance of the code
• References to the primary literature
• Additional functions that can be used to process the output of this model
• Pointers to alternative functions that perform the same or similar functions
• Examples of usage

Exercise 4. Copy and paste the first seven lines from the example in the lm help file. Type the following commands into the console and view the output: lm.D9, summary(lm.D9), attributes(lm.D9). What is being returned in each case? What is lm.D9?

```r
> ## Annette Dobson (1990) "An Introduction to Generalized Linear Models".
> ## Page 9: Plant Weight Data.
> 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)
> lm.D9 <- lm(weight ~ group)
> lm.D90 <- lm(weight ~ group - 1) # omitting intercept
> lm.D9

Call:
lm(formula = weight ~ group)
Coefficients:
(Intercept) groupTrt
  5.032   -0.371

> summary(lm.D9)

Call:
lm(formula = weight ~ group)
Residuals:
   Min     1Q Median     3Q    Max
-1.0710 -0.4938  0.0685  0.2462  1.3690
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.0320     0.2202  22.850 9.55e-15 ***
groupTrt    -0.3710     0.3114  -1.191  0.249
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

> ```
Residual standard error: 0.6964 on 18 degrees of freedom
Multiple R-squared: 0.07308, Adjusted R-squared: 0.02158
F-statistic: 1.419 on 1 and 18 DF, p-value: 0.249

> attributes(lm.D9)

$names
[1] "coefficients"  "residuals"  "effects"  "rank"
[5] "fitted.values" "assign"  "qr"  "df.residual"
[9] "contrasts"  "xlevels"  "call"  "terms"
[13] "model"

$class
[1] "lm"

The object lm.D9 is the fit regression model for the effect of control and treatment groups on weight.

Performing summary on this object returns the anova table for the regression model and other summary statistics.

Performing attributes on this object returns the different parts of the object, e.g., coefficients, residuals, etc.

Exercise 5. The plots generated above consist of 'b'oth points and lines, hence the argument type='b'. Retrieve the help for the plot function and try some of the other plot types, e.g., line plot and point plot.

> ?plot
> par(mfrow=c(2,2))
> plot(flu$day,flu$flu, type='h',main='Type = h', xlab='Day',ylab='New cases')
> plot(flu$day,flu$flu, type='l',main='Type = l', xlab='Day',ylab='New cases')
> plot(flu$day,flu$flu, type='p',main='Type = p', xlab='Day',ylab='New cases')
> plot(flu$day,flu$flu, type='s',main='Type = s', xlab='Day',ylab='New cases')
Exercise 6. Experiment with the functions `plot`, and `points` by plotting the measles incidence data from three different sites in the city of Niamey, Niger in different colors on the same figure.

```r
> plot(seq(0, 30, by=2), niamey[,1], type='b', lwd=3, col='red', xlab='Weeks', ylab='Incidence')
> lines(seq(0, 30, by=2), niamey[,2], type='b', lwd=3, col='blue')
> lines(seq(0, 30, by=2), niamey[,3], type='b', lwd=3, col='green')
> legend('topleft', legend=c('Site 1', 'Site 2', 'Site 3'),
+ col=c('red', 'blue', 'green'), bty='n', lty=1, lwd=2)
```
Exercise 7. Generate one hundred random numbers and plot a histogram.

```r
> Z <- rnorm(100)
> hist(Z)
```