# Exercise 3.1

## For loops and sapply()

**useful commands: `for(){ }`, `print`, `paste`, `sample`, `plot`, `unique`, `points`, `rep`, `which`**

**01*. Create a for loop to print "The year is xxxx" for 2010-2016.**

```
for (i in 2010:2016) { # for each year i
  # 'paste' the year into the sentence, then 'print' to console
  print(paste("The year is", i, sep = " "))
}
```

```
## [1] "The year is 2010"
## [1] "The year is 2011"
## [1] "The year is 2012"
## [1] "The year is 2013"
## [1] "The year is 2014"
## [1] "The year is 2015"
## [1] "The year is 2016"
```

**02*. Instead of printing the answer from 01, save it as a character vector called 'my_years'. # You could do this with a for loop, but that would more difficult and less efficient, so try using sapply()**

```
my_years = sapply(2010:2016, FUN = function(i) paste("The year is", i, sep = " "))
```

**03*. Use a for loop to solve: Find the 100th number in the Fibonacci series: 1, 1, 2, 3, 5, 8, 13, 21 . . .**

```
fib = rep(NA, 100) # make a vector length 100 of NA's to store the results
fib[1] = 1 # set the first value of the vector = 1
fib[2] = 1 # set the second value of the vector = 1
# your for loop to fill in the rest of the fib vector
for (i in 3:100){ # starts at 3 because first 2 Fibonacci numbers are given
  # fills in the next number in the series using the Fibonacci pattern
  fib[i] = fib[i-1] + fib[i-2]
}
# the answer
fib[100] # 100th Fibonacci number
```

```
## [1] 3.542248e+20
```

```
# my self-check tests
fib[8] # 8th Fibonacci number
```

```
## [1] 21
```

```
fib[c(1:15)] # first 15 Fibonacci numbers
```

```
##  [1]   1   1   2   3   5   8  13  21  34  55  89 144 233 377 610
```

**BONUS 03b. How many steps into the Fibonacci series until you get the first number over 10,000?**

```
min(which(fib > 10000)) # index of minimum Fibonacci number over 10000
```

```
## [1] 21
```

```
fib[21] # the number
```

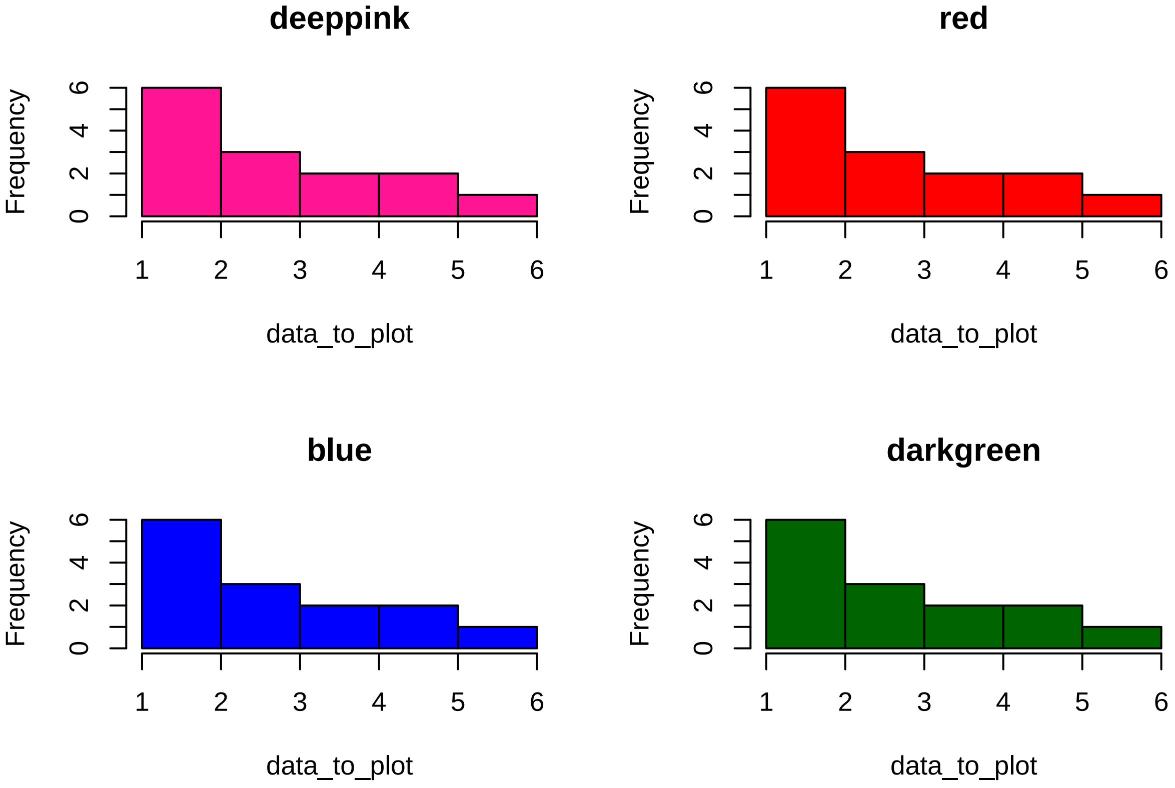
```
## [1] 10946
```

```
fib[min(which(fib > 10000))] # putting it together
```

```
## [1] 10946
```

**BONUS 04. use sapply() to plot a histogram of the data below 4 times, in 4 different colors. For extra style, title the plot by it's color, e.g. the red plot is titled "red"**

```
data_to_plot = c(1,3,4,5,6,3,3,4,5,1,1,1,1,1)
my_colors = c("deeppink", "red", "blue", "darkgreen")
par(mfrow = c(2,2)) # extra styling, plots in a 2x2 grid
sapply(my_colors, FUN = function(i)
  hist(data_to_plot, main = i, col = i))
```



```
##         deeppink    red         blue        darkgreen
## breaks  Integer,6   Integer,6   Integer,6   Integer,6
## counts   Integer,5      Integer,5      Integer,5      Integer,5
## density  Numeric,5      Numeric,5      Numeric,5      Numeric,5
## mids     Numeric,5      Numeric,5      Numeric,5      Numeric,5
## xname    "data_to_plot" "data_to_plot" "data_to_plot" "data_to_plot"
## equidist TRUE           TRUE           TRUE           TRUE
```

**BONUS 05. How would you do question 2 above (print "The year is xxxx" for 2010-2016) without using sapply() or a for loop?**

```r
paste("The year is", 2010:2016, sep = " ") # vector short-hand
```

```
## [1] "The year is 2010" "The year is 2011" "The year is 2012" "The year is 2013"
## [5] "The year is 2014" "The year is 2015" "The year is 2016"
```