Homework #1

- Please produce your assignment as a pdf (knit to pdf). See instructions in announcements if you have not downloaded a LaTeX distributor. If you are still having issues, you may knit to HTML as this is the first assignment. However, future assignments that are not knitted to pdf will be penalized.
- Please also submit your Rmd file (it will not be graded but we want it for reference purposes).
- Please use the provided Rmd file as a basis for your HW (do not submit an R script file), putting your solutions inside the empty blocks that follow "Solution:"
- 10 points per exercise

Ignore this code for now. It is used for reproducibility when using randomly generated numbers.

```
set.seed(1)
```

Exercise 1: Devise an approximate value of pi by generating 1000 x-values on the uniform interval from 0 to 1 and 1000 y-values on the uniform interval from 0 to 1. Use ?runif for help in using the runif() function.

Hint: For a circle of radius 1, pi is equal to the area.

Solution:

```
Ux <- runif(1000, 0, 1)
Uy <- runif(1000, 0, 1)


in_circ <- mean(1*( ((Ux-0.5)^2 + (Uy - 0.5)^2) <= 0.25))
piEst <- in_circ * 4
piEst
```

```
## [1] 3.08
```

Exercise 2: Systematic sample. Given the data:

```
data <- c(23, 89,  1, 34,
          26, 85, 24, 43,
          23, 93, 29, 45,
          32, 42, 43, NA,
          21, 54, 37, 76)
```

get a systematic sample of size `n = 5` from `data` by extracting each value that lies every `K = N/n` elements (where `N` is the total number of elements in `data`).

Your first element needs to be randomly determined as a number between 1 and K.

Note: Your result needs to be a vector containing the 5 elements that are part of your systematic sample. Also, the subsetting from the vector data must be a one liner (you need to extract all elements together).

Solution:

```
n <- 5
k = length(data)/n
```

```
r = sample(1:k, 1)
data[seq(r, r + k*(n-1), k)]
```

```
## [1]  1 24 29 43 37
```

Exercise 3: Run the following code to load the 'babyboom' dataset from the UsingR package. This dataset contains the time of birth, sex, and birth weight for 44 babies born in one 24-hour window at a hospital in Australia.

What were the weights, in grams, of the lightest and heaviest babies recorded?

The "gender" column classifies each baby as a 'boy' or a 'girl'. How many of each gender are there? Plot a histogram of weights for girls and a separate histogram of weights for boys, both using Scott's rule for the number of bins. Which histogram looks more bell-shaped?

Solution:

```
#max weight
max(babyboom$wt)
```

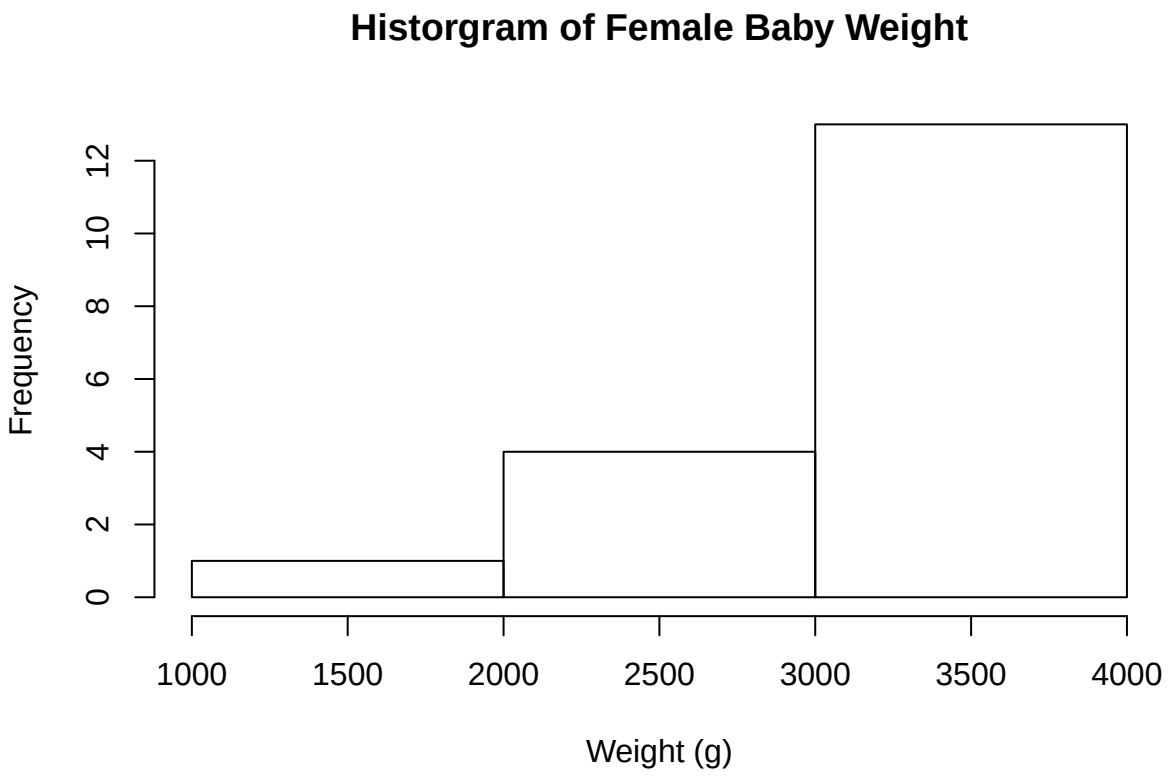
```
## [1] 4162
```

```
#min weight
min(babyboom$wt)
```

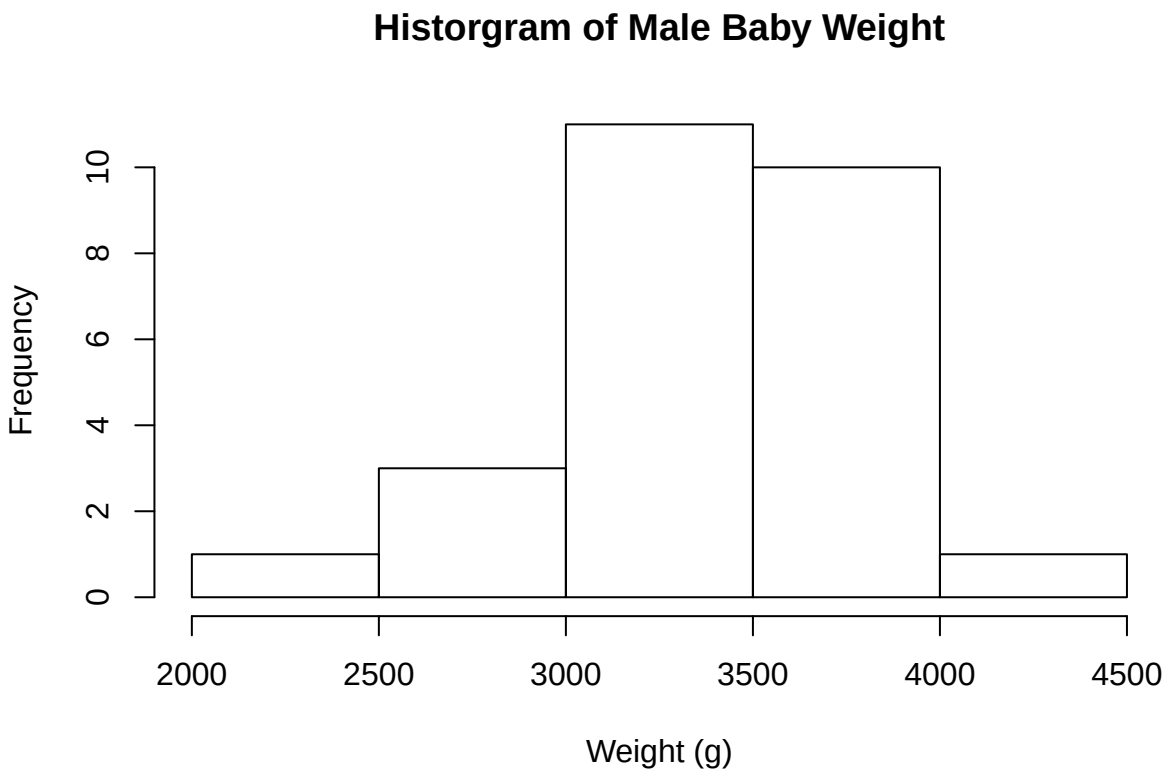
```
## [1] 1745
```

```
babyboom_f <- subset(babyboom, gender == "girl")
babyboom_m <- subset(babyboom, gender == "boy")

hist(babyboom_f$wt, "Scott", main = "Historgram of Female Baby Weight", xlab = "Weight (g)")
```




```
hist(babyboom_m$wt, "Scott", main = "Historgram of Male Baby Weight", xlab = "Weight (g)")
```



Exercise 4: Define the vector as follows:

```
data <- c(21, 16, 12, 3)
```

Generate both a simple random sample of size 3 and a random sample of size 3 from this vector. Are "simple random sample" and "random sample" the same? If not, what is the difference?

Solution:

```
sample(data, 3)
```

```
## [1] 12  3 21
```

```
sample(data, 3, replace = TRUE)
```

```
## [1] 12 16 16
```

The two are not the same as the simple random allows for replacement.

Exercise 5: Consider the following string a.

```
a <- c("group1", "group2", "group3")
```

Take a random sample of size 20 from a. Convert this random sample to a factor. Sort this factor in decreasing order. Use plot() to construct a bar chart of the frequencies of each level of the factor.

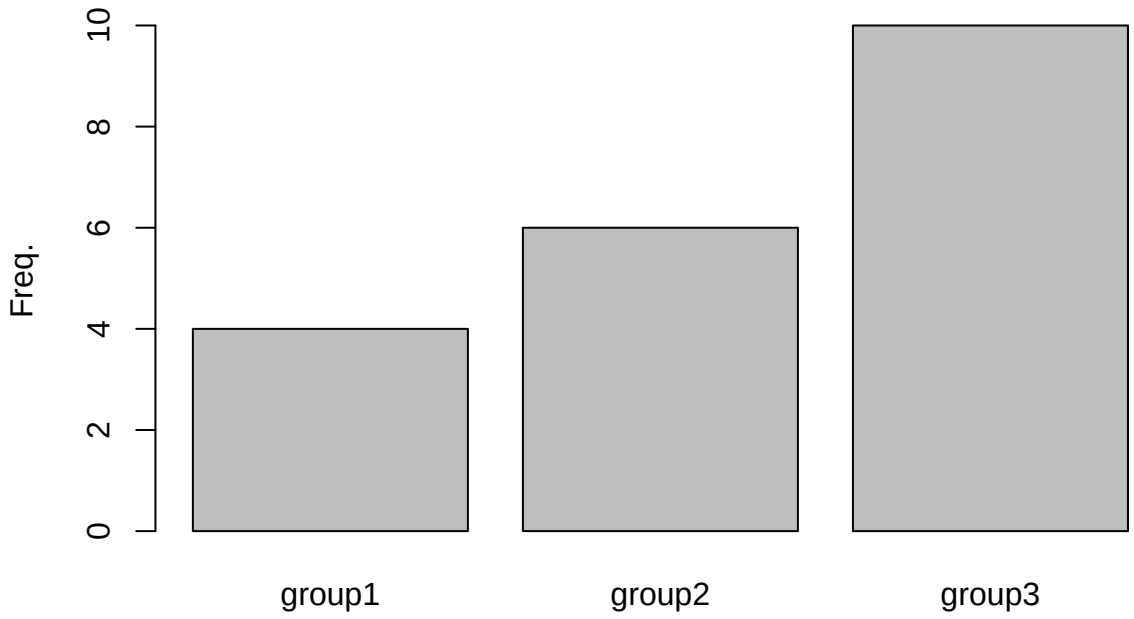
Solution:

```
a_fac <- sample(a, 20, replace = TRUE)
```

```r
a_fac <- as.factor(a_fac)

a_fac <- sort(a_fac, decreasing = T)

plot(a_fac, ylab = "Freq.")
```



Exercise 6: Load the 'snacks' dataset using the following code.

Using logical indexes, how many snacks have an NA value for sugar? How many have more protein than sugar (confirm that your answer is not "NA")? What is the average amount of sodium in snacks with over 500 calories? What about in snacks with less than 300 calories? How many snacks have either 40 carbohydrates or fewer OR 350 or more calories? Finally, write a line of code to see how many snacks satisfy ALL of the following criteria: less than 400 calories, less than 20 sugar, 12 or more protein.

Solution:

```r
sum(is.na(snacks$sugar))
```

```
## [1] 11
```

11 snacks have NA for sugar

```r
dif <- snacks$protein - snacks$sugar

dif <- dif > 0
sum(dif, na.rm = T)
```

```
## [1] 35
```

35 snacks have more protein than sugar

```
mean(snacks$sodium[snacks$calories > 500])
```

```
## [1] 307.9706
```

```
mean(snacks$sodium[snacks$calories < 300])
```

```
## [1] 491.5
```

```
carb <- snacks$carbohydrates <= 40
cal <- snacks$calories >= 350

sum(carb | cal)
```

```
## [1] 114
```

```
sum(snacks$calories < 400 & snacks$sugar < 20 & snacks$protein >= 12)
```

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