# Homework 3

*Kitu Komya*

*November 2, 2016*

```
# I am loading in the packages needed for this lab
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.4.2
```

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.4.2
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

## 1. Tidy Data

### 1.1 *Lord of the Rings*

**1.**

```
# Here I am reading in the data using the package readr
the_fellowship_of_the_ring <- read_csv(file = "The_Fellowship_Of_The_Ring.csv")
```

```
## Parsed with column specification:
## cols(
##   Film = col_character(),
##   Race = col_character(),
##   Female = col_integer(),
##   Male = col_integer()
## )
```

```
the_two_towers <- read_csv(file = "The_Two_Towers.csv")
```

```
## Parsed with column specification:
## cols(
##   Film = col_character(),
##   Race = col_character(),
```

```
##   Female = col_integer(),
##   Male = col_integer()
## )
```

```
the_return_of_the_king <- read_csv(file = "The_Return_Of_The_King.csv")
```

```
## Parsed with column specification:
## cols(
##   Film = col_character(),
##   Race = col_character(),
##   Female = col_integer(),
##   Male = col_integer()
## )
```

**2.**

```
full_dataset <- full_join(x = the_fellowship_of_the_ring, y = the_two_towers) # Using full join, I've c
```

```
## Joining, by = c("Film", "Race", "Female", "Male")
```

```
full_dataset <- full_join(x = full_dataset, y = the_return_of_the_king) # I'm now combining the combine
```

```
## Joining, by = c("Film", "Race", "Female", "Male")
```

We now have one data frame with 9 observations and 4 variables!

**3.**

```
# I have gathered the data so it is now long data instead of wide.
full_dataset <- gather(full_dataset, "Sex", "Words", 3:4)
```

We are left with 18 observations and 4 variables!

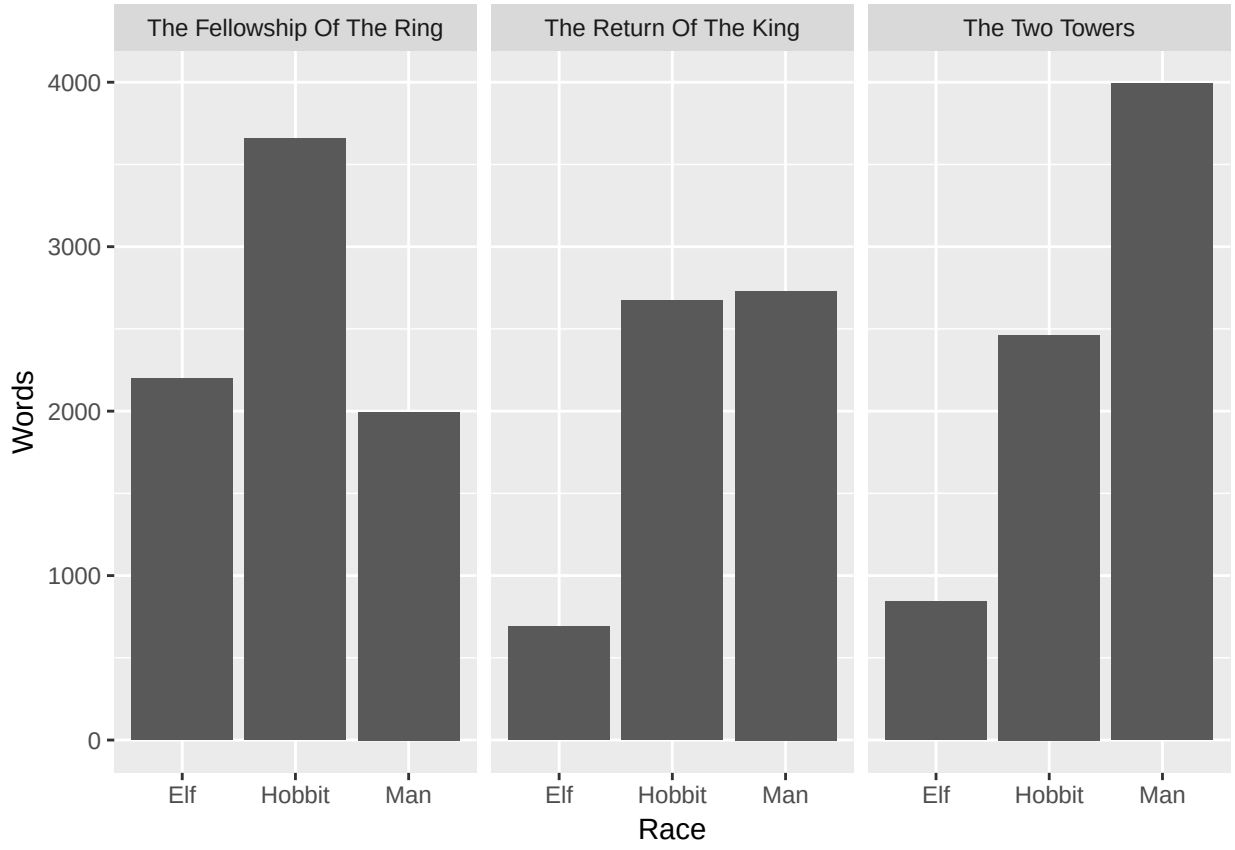
**4.**

```
# I have outputted a table that shows the desired information
full_dataset %>%
  group_by(Race) %>%
  summarize(mean(Words), sd(Words), sum(Words)) %>%
  kable()
```

| Race | mean(Words) | sd(Words) | sum(Words) |
|---|---|---|---|
| Elf | 622.8333 | 398.0133 | 3737 |
| Hobbit | 1466.0000 | 1648.9571 | 8796 |
| Man | 1452.0000 | 1448.4770 | 8712 |

The table from kable displays the information requested.

```
# I have made a beautiful bargraph using ggplot
ggplot(data = full_dataset) + geom_bar(aes(Race, Words), stat = "identity") + facet_wrap(~Film)
```




In The Fellowship of the Ring, Hobbits spoke the most, while Men and Elves spoke approximately the same, but less. In The Return of the King, the Hobbits and Men spoke the most, approximately tied, while the Elves spoke the least. And finally, in The Two Towers, Men spoke the most, with Hobbit following, and Elves trailing last.

### 1.2 Comic Books

**1.**

```
# I am reading in the data
superheroes <- read_csv(file = "superheroes.csv")
```

```
## Parsed with column specification:
## cols(
##   name = col_character(),
##   alignment = col_character(),
##   sex = col_character(),
##   publisher = col_character()
## )
```

```
publishers <- read_csv(file = "publishers.csv")
```

```
## Parsed with column specification:
## cols(
##   publisher = col_character(),
##   yr_founded = col_integer()
## )
```

**2.**

```
# I am creating an inner join to only include the characters whose publishers are in both sets
characters <- inner_join(publishers, superheroes, by = c("publisher" = "publisher"))
characters
```

```
## # A tibble: 6 x 5
##   publisher yr_founded     name alignment    sex
##       <chr>      <int>    <chr>     <chr>  <chr>
## 1        DC       1934   Batman      good   male
## 2        DC       1934    Joker       bad   male
## 3        DC       1934 Catwoman       bad female
## 4    Marvel       1939  Magneto       bad   male
## 5    Marvel       1939    Storm      good female
## 6    Marvel       1939 Mystique       bad female
```

Hellboy has been dropped, as seen by our new output not containing him!

**3.**

```
# I am creating an anti join to only include publishers that aren't contained in the superheroes data
characters_not <- anti_join(publishers, superheroes, by = c("publisher" = "publisher"))
characters_not
```

```
## # A tibble: 1 x 2
##   publisher yr_founded
##       <chr>      <int>
## 1     Image       1992
```

We see that Image is the only publisher not in the superheroes data!

**4.**

```
# I am creating a left join to append all the superheroes' info appended to the publishers data
all_publishers <- left_join(publishers, superheroes, by = c("publisher" = "publisher"))
all_publishers
```

```
## # A tibble: 7 x 5
##   publisher yr_founded     name alignment    sex
##       <chr>      <int>    <chr>     <chr>  <chr>
## 1        DC       1934   Batman      good   male
## 2        DC       1934    Joker       bad   male
## 3        DC       1934 Catwoman       bad female
## 4    Marvel       1939  Magneto       bad   male
## 5    Marvel       1939    Storm      good female
## 6    Marvel       1939 Mystique       bad female
## 7     Image       1992     <NA>      <NA>   <NA>
```

All of the superheroes' data has been amended to the publishers data.

**5.**

```
# I am creating a full join of everything
full <- full_join(publishers, superheroes, by = c("publisher" ="publisher"))
full
```

```
## # A tibble: 8 x 5
##           publisher yr_founded     name alignment    sex
##               <chr>      <int>    <chr>     <chr>  <chr>
## 1               DC       1934   Batman      good   male
## 2               DC       1934    Joker       bad   male
```

```
## 3                DC       1934 Catwoman       bad female
## 4            Marvel       1939  Magneto       bad   male
## 5            Marvel       1939    Storm      good female
## 6            Marvel       1939 Mystique       bad female
## 7             Image       1992     <NA>      <NA>   <NA>
## 8 Dark Horse Comics         NA  Hellboy      good   male
```

```
dim(full) # I want to look at the dimensions
```

```
## [1] 8 5
```

As seen above, the dimension of this new data frame is 8 by 5!

# 2. Simulations using loops

### 2.1 Central Limit Theorem

**1.**

```
# I am simulating the # of successes from flipping a coin ten times with a 0.2 probability success
rbinom(1, 10, 0.2)
```

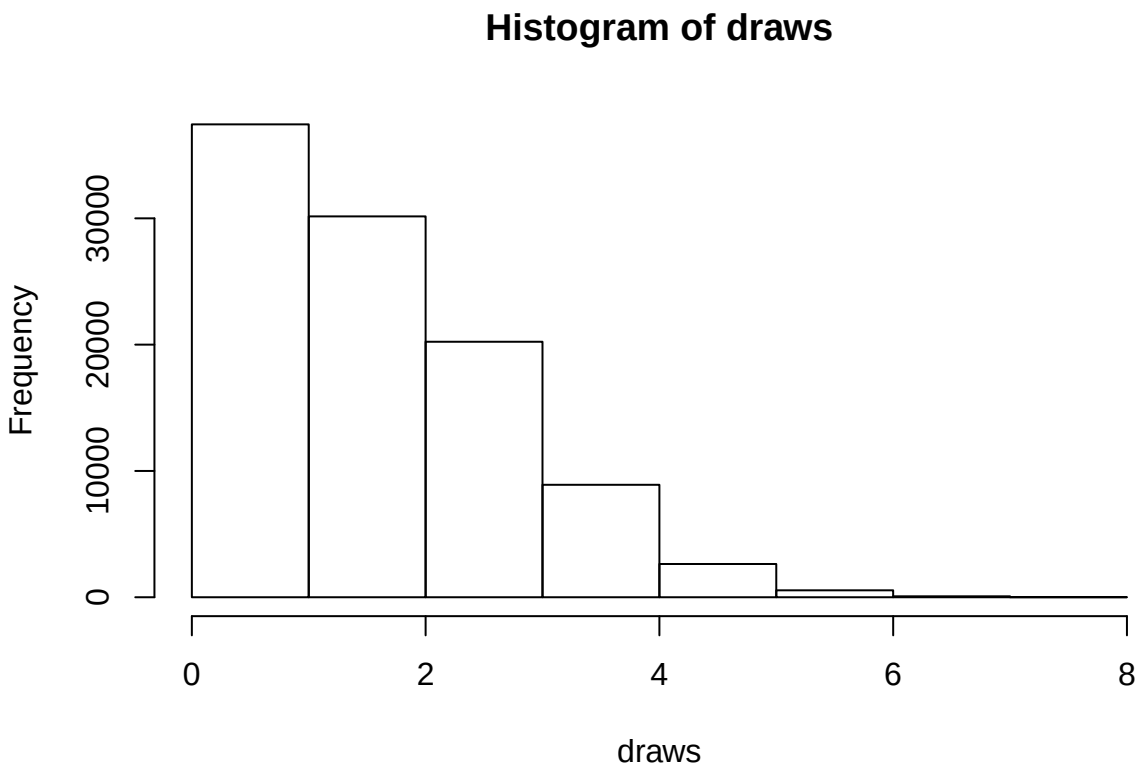
```
## [1] 1
```

**2.**

```
# I am simulating 100,000 draws of the same simulation now
set.seed(2016)
draws <- rbinom(100000, 10, 0.2)
```

**3.**

```
# I am creating a histogram of the draws
hist(draws, breaks = 10)
```




As seen by the histogram, the shape is right skewed. The center is around 2, the shape is non-normal (right-skewed), and the spread is from 0 to 6, but more notably from 0 to 4, which is a large spread, regardless. We are getting such a shape because the probability of success is 0.2.

**4.**

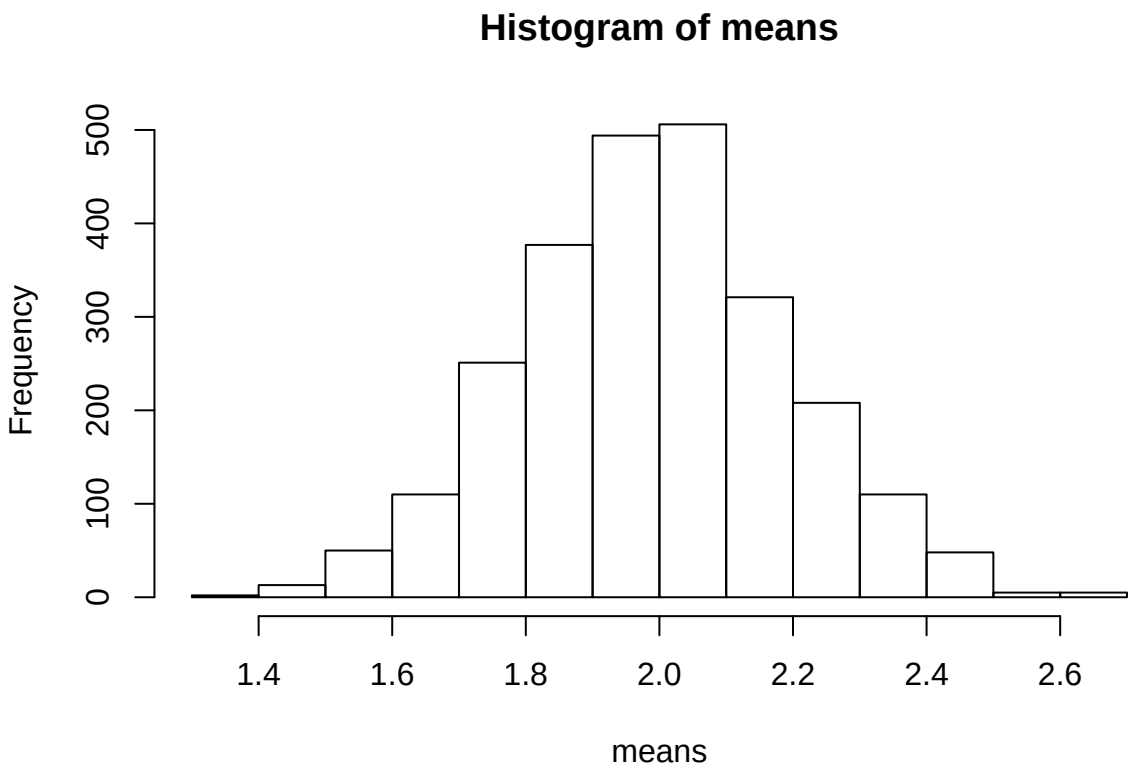
```
# I have matrixified my data!
matrix <- matrix(data = draws, nrow = 40, ncol = 2500, byrow = F)
```

**5.**

```
# I have created a for loop to calculate the column means of my matrix. I can't believe I did this on m
n = ncol(matrix) # aka n = 2500
means <- vector(length = 2500) # I have initialized a new vector
for(i in 1:n) {
  col_mean <- mean(matrix[, i]) # This will calculate the mean for each column
  means[i] <- col_mean # Now each mean will be stored in its respective column
}
```

**6.**

```
# I have created a histogram of the means!
hist(means)
```




```
mean(means) # I am finding the mean of means!
```

```
## [1] 2.00293
```

```
sd(means) # I am finding the standard deviation of means
```

```
## [1] 0.2004563
```

What a beautiful graph. The graph is normally distributed, symmetric around its mean of 2. The spread is from 1.4 to 2.6, which is a much smaller spread than before. Using the mean and sd commands, we see that more precisely the mean is centered around 2.00293 and the standard deviation is 0.2004563, which is a small number that represents that 68% of the data fall within +/- standard deviation (aka the number 0.2004563) from the mean. The mean value says that the average value landed on is 2.00293. Compared from the distribution before, this is drastically different. Here we have a normal, symmetric distribution with a much smaller spread than before where we had a right-skewed graph with a larger spread. That is because the Central Limit Theorem states that with larger sample sizes, it gets closer to the true mean of the sample (variability decreases), and the sampling distribution of the means is normal, despite the initial distribution it's taken from. It's pretty cool to see this happen visually in action!

### 2.2 Random Walk

**1.**

```
# I am creating these objects, as asked
place <- 0
zeros <- 0
path <- c()
```

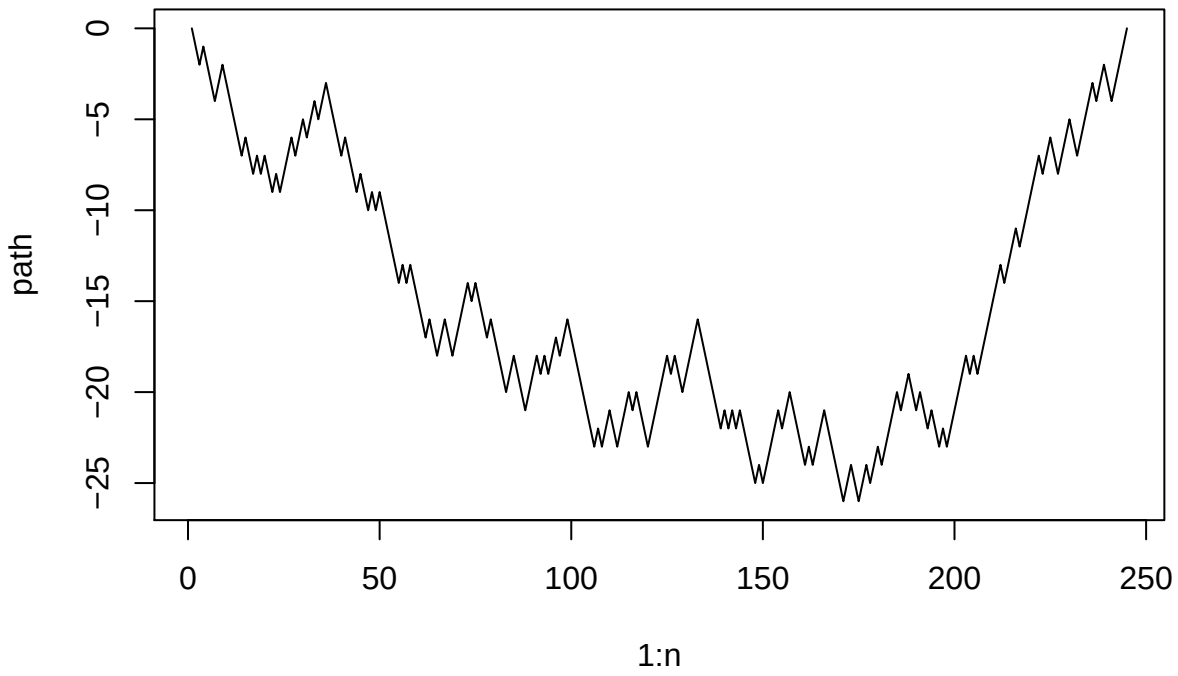
```
iteration <- c()
```

**2.**

```
# I am filling out the template of code here
# This will ensure my code matches everyone else's
set.seed(2016)

# The loop will run until zeros is not zero
while (zeros == 0) {
  iteration <- c(iteration, length(iteration)) # To each iteration, its current length will be added to
  path <- c(path, place) # To each path, its current place will be added too
  draw <- runif(n = 1, min = 0, max = 1) # This will draw a number from a random, uniform distribution

  # We are deciding what to do when draw is a certain value
  if (draw >= 0.5) {
    place <- place + 1 # We will add to place if draw is >= 0.5
  } else {
    place <- place - 1 # We will subtract from place if draw is < 0.5
  }

  # Once place finally reaches 0
  if (place == 0) {
    iteration <- c(iteration, length(iteration)) # To each iteration, its current length will be added
    path <- c(path, place) # To ecah path, its current place will be added again too
    zeros <- zeros + 1  # Zeroes will change its value, thereby falsifying the while loop condition
    break # We will break out of the loop once this condition is met
  }
}
```

**3.**

```
# This will return the length of the iteration vector, aka the number of iterations it took to return t
n <- length(iteration)
plot(x = 1:n, y = path, type = "l", main = "Random Walk: the path taken until the initial position is re
```




Wow, the line graph is so pretty.

**4.**

```
# This will provide how many times the while loop run, aka the length it took to return to the starting
length(iteration)
```

```
## [1] 245
```

It took 245 iterations to return to our starting place! To be honest, I did not expect it to take so long for us to return back to 0.