Duncan McKinnon

West

HW 10

Load Packages and Data

1).

```
# load tidyverse
suppressPackageStartupMessages({
  library(tidyverse)
})

# load skulls data
skulls <- read_csv("C:/Users/dunca/OneDrive/Desktop/Course Work/Intro to Data Science/HW/HW10/skulls.csv
glimpse(skulls)
```

```
## Observations: 353
## Variables: 12
## $ GOL <dbl> 189, 182, 191, 191, 178, 194, 186, 186, 186, 189, 186, 182...
## $ NOL <dbl> 185, 178, 187, 188, 177, 191, 183, 184, 182, 189, 183, 182...
## $ BNL <dbl> 100, 102, 102, 100, 97, 106, 95, 103, 96, 101, 107, 99, 10...
## $ BBH <dbl> 135, 139, 123, 127, 128, 132, 122, 130, 134, 131, 138, 124...
## $ XCB <dbl> 143, 145, 140, 141, 138, 139, 143, 141, 141, 144, 143, 136...
## $ XFB <dbl> 120, 120, 114, 123, 117, 118, 122, 121, 123, 115, 116, 113...
## $ ZYB <dbl> 133, 137, 134, 135, 129, 136, 128, 128, 135, 130, 136, 125...
## $ AUB <dbl> 119, 125, 125, 127, 121, 128, 119, 121, 123, 125, 129, 120...
## $ WCB <dbl> 70, 66, 74, 71, 69, 76, 71, 72, 71, 63, 71, 69, 71, 70, 72...
## $ ASB <dbl> 112, 113, 112, 113, 111, 112, 108, 109, 111, 108, 113, 111...
## $ BPL <dbl> 96, 108, 102, 95, 90, 102, 95, 100, 87, 95, 96, 98, 96, 98...
## $ NPH <dbl> 66, 64, 67, 76, 67, 69, 66, 64, 72, 63, 70, 69, 70, 73, 67...
```

Data Preprocessing

2, 3).

```
# remove na's and scale skulls data
skulls <- skulls %>% na.omit() %>% lapply(scale) %>% as.data.frame()
dim(skulls)
```

```
## [1] 353  12
```

```
glimpse(skulls)
```

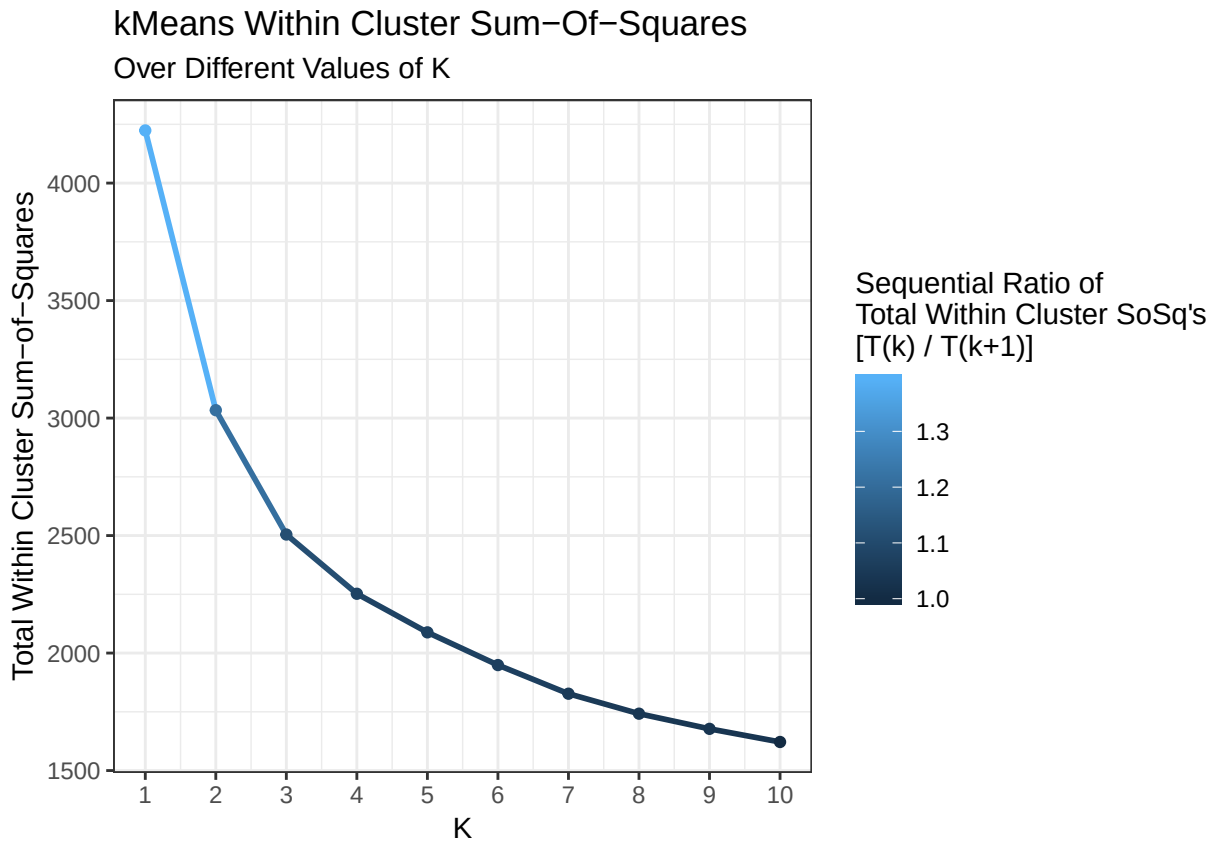
```
## Observations: 353
## Variables: 12
## $ GOL <dbl> 0.60620611, -0.31958827, 0.87071879, 0.87071879, -0.848613...
## $ NOL <dbl> 0.51427756, -0.47158733, 0.79595324, 0.93679108, -0.612425...
## $ BNL <dbl> -0.15068816, 0.24479824, 0.24479824, -0.15068816, -0.74391...
```

```
## $ BBH <dbl> 0.47925862, 1.08836139, -1.34804969, -0.73894692, -0.58667...
## $ XCB <dbl> 0.61399734, 0.89204873, 0.19692025, 0.33594595, -0.0811311...
## $ XFB <dbl> 0.565977013, 0.565977013, -0.289043895, 0.993487467, 0.138...
## $ ZYB <dbl> -0.12145508, 0.52326138, 0.03972403, 0.20090315, -0.766171...
## $ AUB <dbl> -0.47939277, 0.44031184, 0.44031184, 0.74688004, -0.172824...
## $ WCB <dbl> -0.51658989, -1.48656985, 0.45339007, -0.27409490, -0.7590...
## $ ASB <dbl> 0.43563806, 0.62643239, 0.43563806, 0.62643239, 0.24484372...
## $ BPL <dbl> -0.4258429, 1.4590794, 0.5166182, -0.5829198, -1.3683041, ...
## $ NPH <dbl> -0.13216845, -0.48561892, 0.04455679, 1.63508390, 0.044556...
```

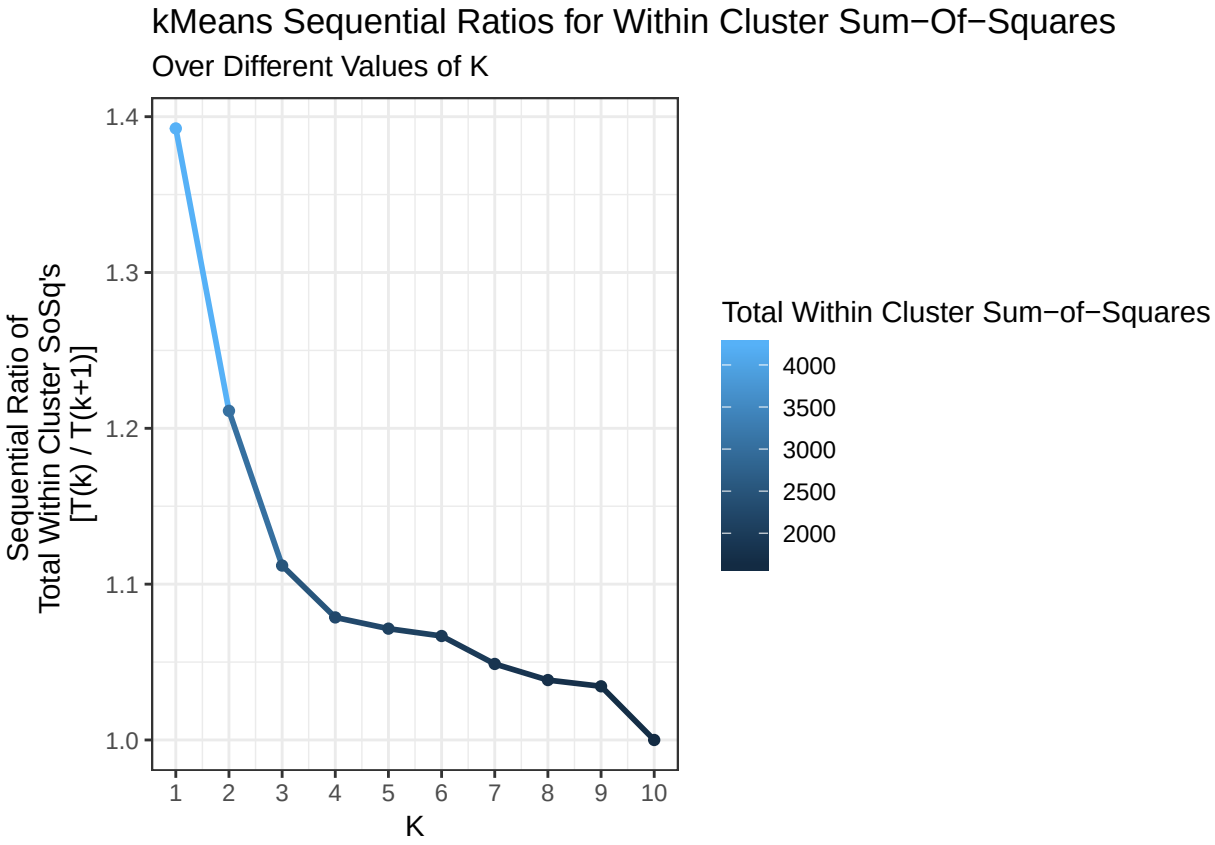
kMeans Clustering Model and Analysis

4).

```
# set random seed
set.seed(1847)

# set up k vals to test
kms <- 1:10

# run kmeans on all k in kms and collect clustering models
kmods <- lapply(kms,
                function(k){
                  return( kmeans(skulls, centers = k, nstart = 20) )
                }
              )

# collect within cluster sum of squares as vector
kss <- kmods %>% lapply('[[', 'tot.withinss') %>% unlist()

# get values of ratio between sequential sum-of-squares results for k to help identify
# inflection point and diminishing returns (divide first nine kss by last nine kss,
# assuming the ratio will approach 1.0 as increasing k has less impact on ss)
rss <- c(kss[1:9] / kss[2:10], 1.0)

# collect results in data frame
kdf <- data.frame('k' = kms, 'ss' = kss, 'rss' = rss)

# set base plot for exploring kdf
p <- ggplot(kdf) + theme_bw()

# plot sum of squares to find inflection point
# where increasing k no longer significantly reduces within cluster sum of squares
p + geom_line(aes(x = k, y = ss, color = rss), size=1) +
  geom_point(aes(x = k, y = ss, color = rss)) +
  labs(title = 'kMeans Within Cluster Sum-Of-Squares',
       subtitle = 'Over Different Values of K',
       x = 'K',
       y = 'Total Within Cluster Sum-of-Squares',
       color = 'Sequential Ratio of \nTotal Within Cluster SoSq\'s \n[T(k) / T(k+1)]') +
  scale_x_continuous(breaks = 1:10, labels = 1:10)
```




```
# Plot rate of change in within cluster sum of squares for sequential values of k
p + geom_line(aes(x = k, y = rss, color = ss), size=1) +
  geom_point(aes(x = k, y = rss, color = ss)) +
  labs(title = 'kMeans Sequential Ratios for Within Cluster Sum-Of-Squares',
       subtitle = 'Over Different Values of K',
       x = 'K',
       y = 'Sequential Ratio of \nTotal Within Cluster SoSq\'s \n[T(k) / T(k+1)]',
       color = 'Total Within Cluster Sum-of-Squares') +
  scale_x_continuous(breaks = 1:10, labels = 1:10)
```

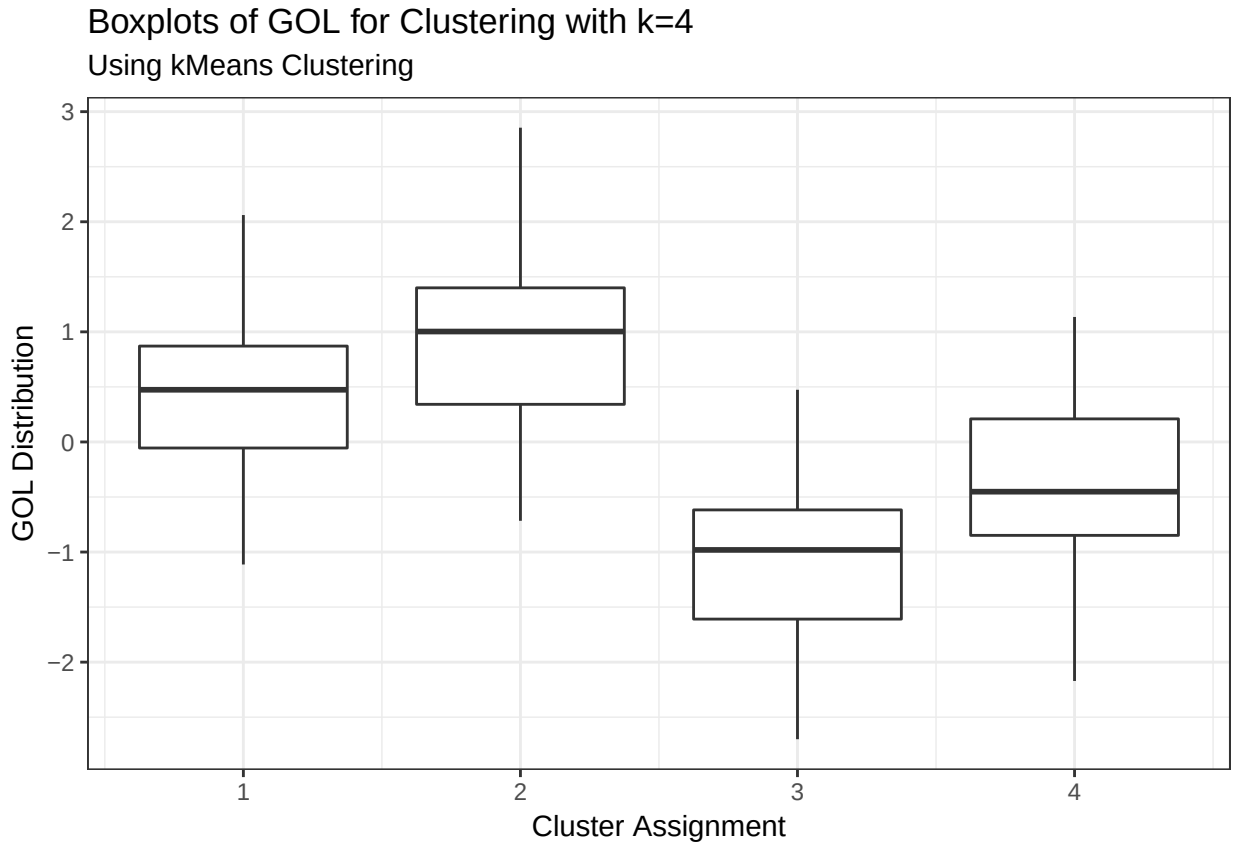


Looking at the graph of the within cluster sum-of-squares by k value and the difference in sequential sum-of-squares by k-values, the first thing I notice is that the difference between sequential k values approach 1.0, but it never gets very close. Increasing k continues to have a relatively linear impact on the decrease in within cluster sum-of-squares right up to k = 10. While the k's never truly converge, the decrease in within cluster sum-of-squares is clearly super linear below k = 4, and subsequently becomes pretty linear. This indicates that the highest value of k that is still having a significant impact is around 3 or 4, so the best kmeans model for this data would probably be the one for k = 4.

5).

```
# change k to test different models (best chosen k = 4)
k_km <- 4

# change skulls_fact_km to test different factors (chosen GOL)
skulls_fact_km <- 'GOL'

# select kmeans model (based on k_km)
km_mod <- kmods[[k_km]]

# add cluster labels to data
skulls_km <- skulls %>% mutate(cluster = km_mod$cluster)

# plot skulls_fact_km distribution box plots for k_km clusters
ggplot(skulls_km) +
  geom_boxplot(aes_string(y = skulls_fact_km, x = 'cluster', group = 'cluster')) +
```

```
  labs(title = paste('Boxplots of ', skulls_fact_km,' for Clustering with k=', k_km, sep=''),
       subtitle = 'Using kMeans Clustering',
       x = 'Cluster Assignment',
       y = paste(skulls_fact_km, 'Distribution')) +
  scale_x_continuous(breaks = 1:k_km, labels = 1:k_km) +
  theme_bw()
```



6).

```
# get total within group sum of squares for clustering with k
km_mod$tot.withinss
```

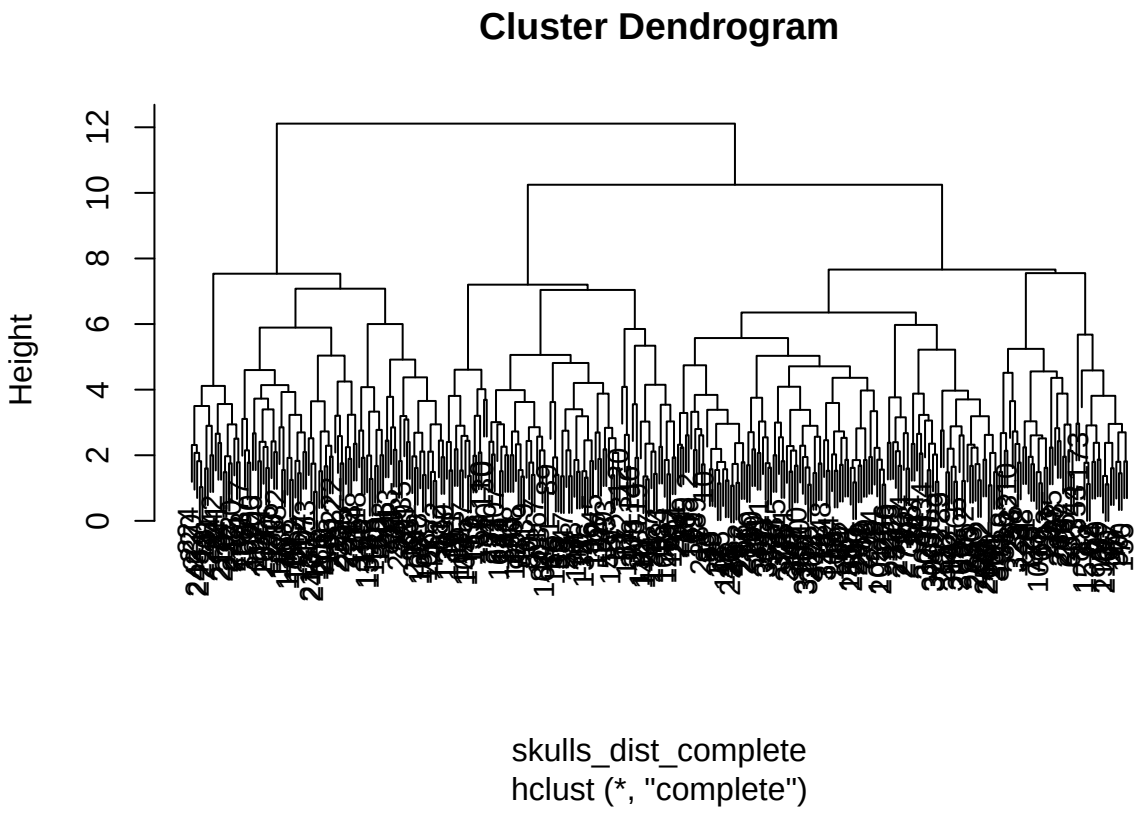
```
## [1] 2252.374
```

Heirarchical Clustering Model and Analysis

7).

```
# set random seed
set.seed(1842)

# heirarchical clustering on skull data using complete linkage
skulls_dist_complete <- dist(skulls)
```

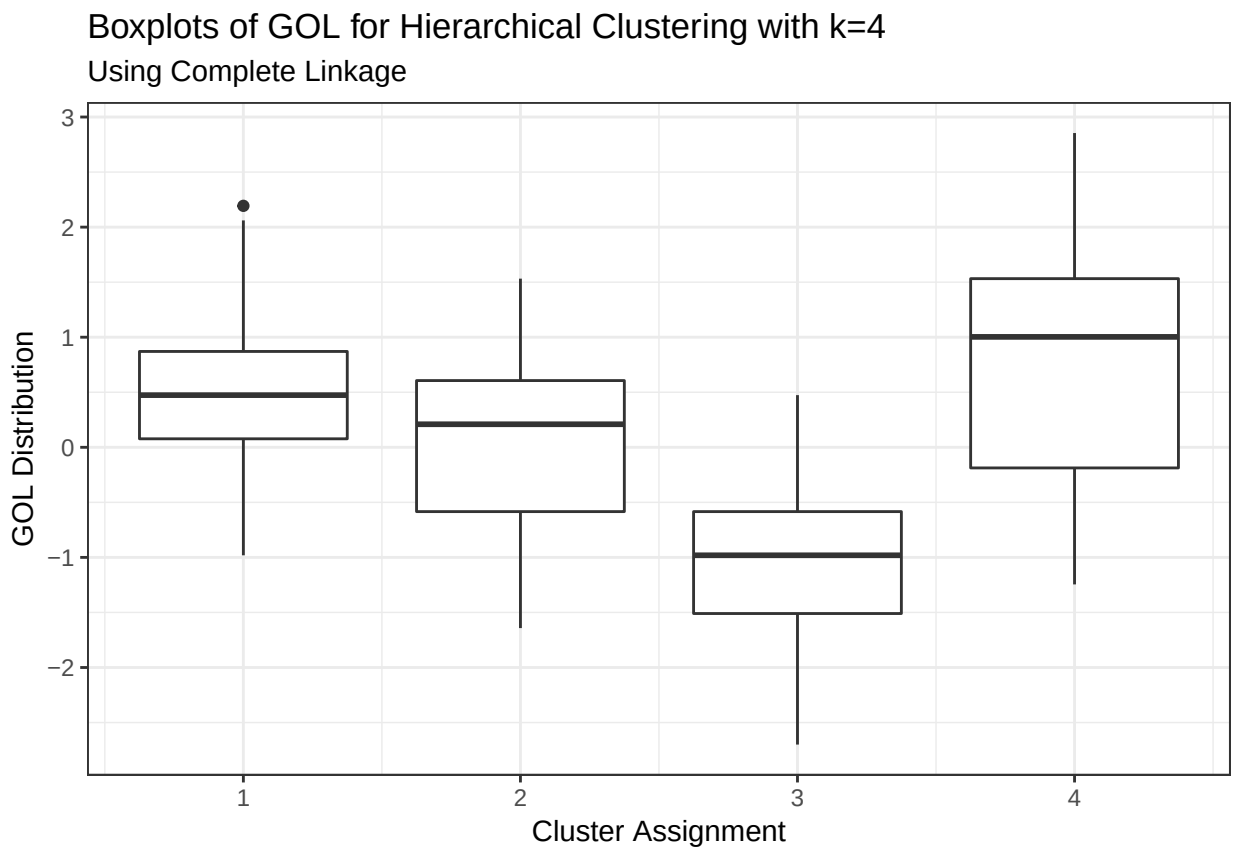
```r
skulls_hclust_complete <- hclust(skulls_dist_complete, method = "complete")

plot(skulls_hclust_complete, cex = 0.9)
```



8).

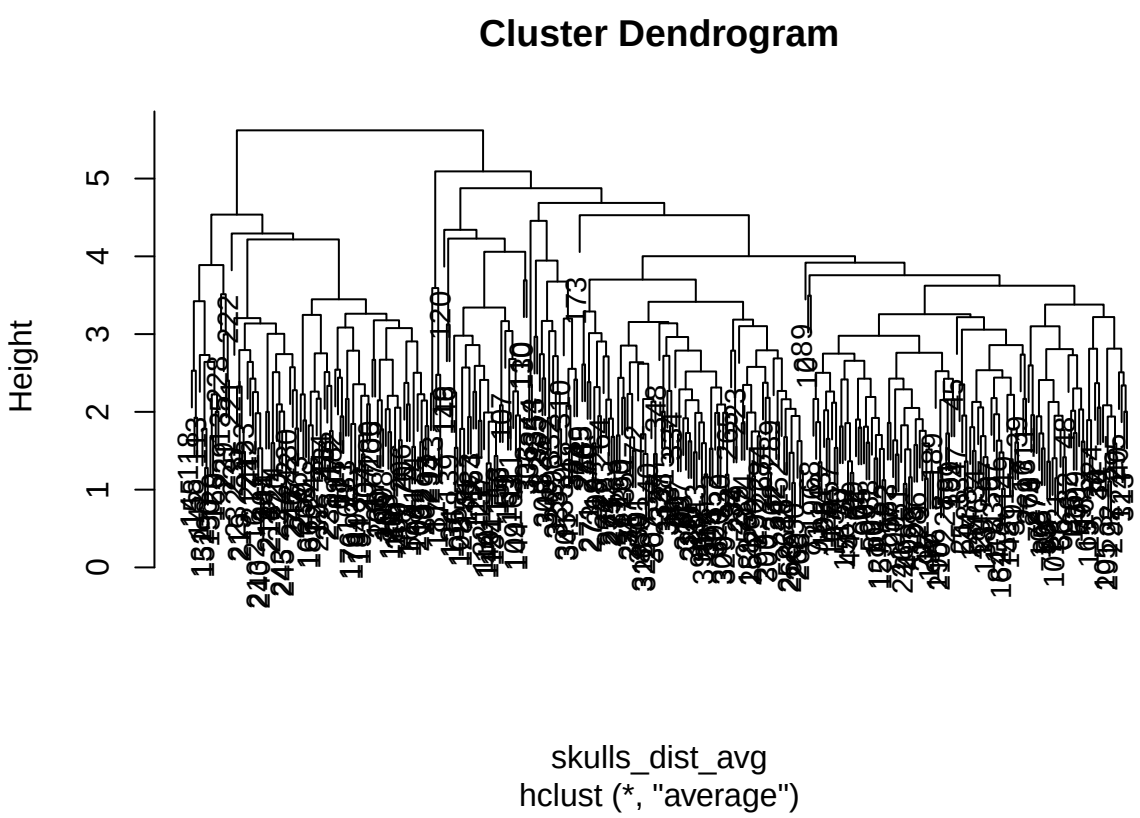
```r
# change k to test different models (default chosen k = 4)
k_h_complete <- 4

# change skull_fact_h_complete to test different factors (chosen GOL)
skulls_fact_h_complete <- 'GOL'

# cut tree down to k_h_complete
skulls_cut_complete <- cutree(skulls_hclust_complete, k_h_complete)

# add cluster labels to data
skulls_h_complete <- skulls %>% mutate(cluster = skulls_cut_complete)

# plot skulls_fact_h_complete distribution box plots for k_h_complete clusters
ggplot(skulls_h_complete) +
  geom_boxplot(aes_string(y = skulls_fact_h_complete, x = 'cluster', group = 'cluster')) +
  labs(title = paste('Boxplots of ', skulls_fact_h_complete,' for Hierarchical Clustering with k=', k_h
       subtitle = 'Using Complete Linkage',
       x = 'Cluster Assignment',
```

```
      y = paste(skulls_fact_h_complete, 'Distribution')) +
  scale_x_continuous(breaks = 1:k_h_complete, labels = 1:k_h_complete) +
  theme_bw()
```



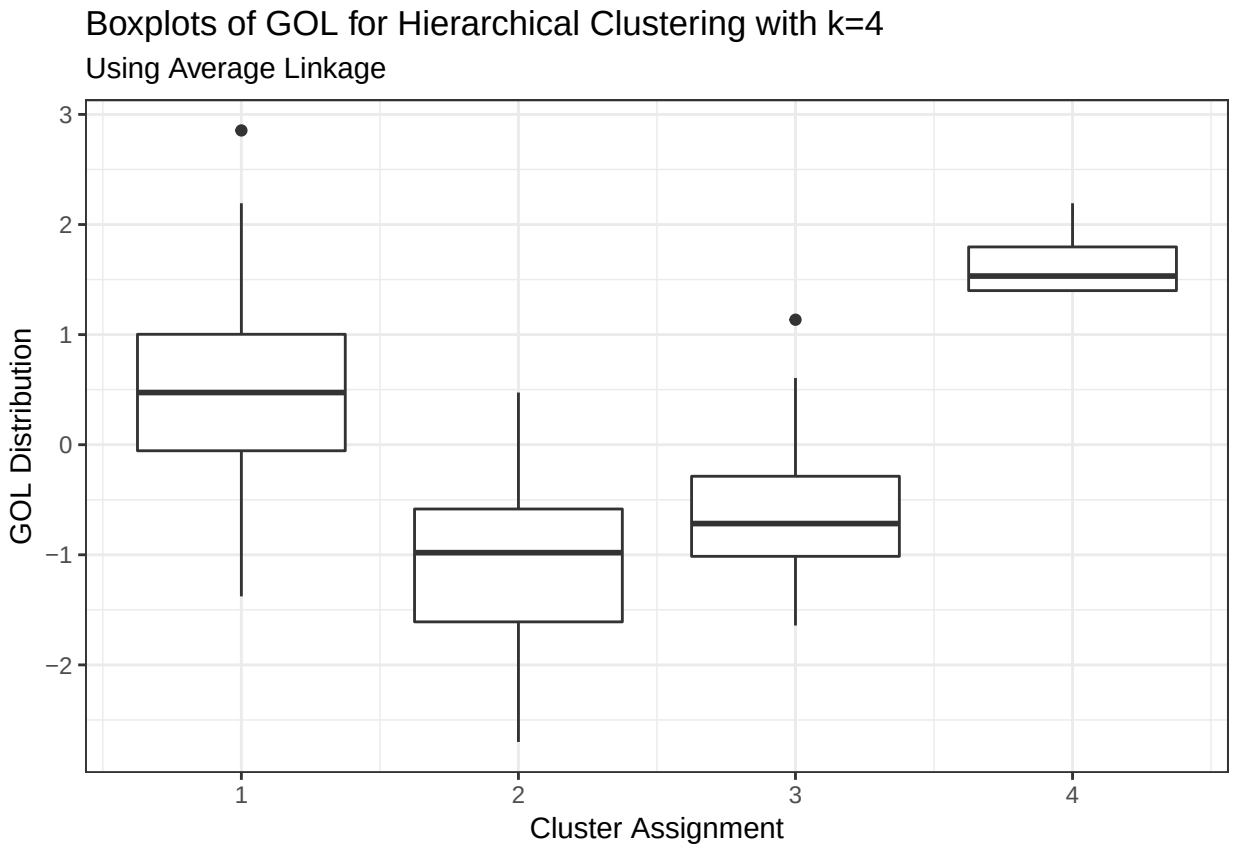
9).

```
# set random seed
set.seed(1842)

# heirarchical clustering on skull data using average linkage
skulls_dist_avg <- dist(skulls)
skulls_hclust_avg <- hclust(skulls_dist_avg, method = "average")

plot(skulls_hclust_avg, cex = 0.9)
```



10).

```
# change k to test different models (default chosen k = 4)
k_h_avg <- 4

# change skulls_fact_h_avg to test different factors (chosen GOL)
skulls_fact_h_avg <- 'GOL'

# cut tree down to k_h_avg
skulls_cut_avg <- cutree(skulls_hclust_avg, k_h_avg)

# add cluster labels to data
skulls_h_avg <- skulls %>% mutate(cluster = skulls_cut_avg)

# plot skulls_fact_h_avg distribution box plots for k_h_avg hierarchical clusters
ggplot(skulls_h_avg) +
  geom_boxplot(aes_string(y = skulls_fact_h_avg, x = 'cluster', group = 'cluster')) +
  labs(title = paste('Boxplots of ', skulls_fact_h_avg,' for Hierarchical Clustering with k=', k_h_avg,
       subtitle = 'Using Average Linkage',
       x = 'Cluster Assignment',
       y = paste(skulls_fact_h_avg, 'Distribution')) +
  scale_x_continuous(breaks = 1:k_h_avg, labels = 1:k_h_avg) +
  theme_bw()
```


Boxplots of GOL for Hierarchical Clustering with k=4
Using Average Linkage
GOL Distribution
3
2
1
0
−1
−2
1
2
3
4
Cluster Assignment