Data Visualization - Clustering Washington State Trails

Dataset we will be working with:

```
hike_data <- readRDS(url('https://raw.githubusercontent.com/rfordatascience/tidytuesday/
master/data/2020/2020-11-24/hike_data.rds'))
```

```
# Take characters out of "Length" category, leave numerical values
hike_data_new <- hike_data %>%
  separate(length, c("length","Useless Words"), sep = "m") %>%
  select(!c("Useless Words","rating"))
# Change columns to numeric values
fix_cols = c("gain","highpoint","length")
hike_data_new[fix_cols] <- sapply(hike_data_new[fix_cols], as.numeric)
```

Dataset: The dataset used for the below project contains data on all hiking trails found inside of Washington state. Data on hiking trails includes their name (name), their location (location), their length in miles (length), their gain in elevation in feet above sea level (gain), their highest point in feet above sea level (highpoint), a list of interesting features to see (features) and a short description (description). The dataset was scraped from Washington Trails Association's Hiking Guide Webpage which can be found at https://www.wta.org/go-outside/hikes?b_start:int=1 (https://www.wta.org/go-outside/hikes?b_start:int=1). The scraped data set and additional information can be found here: https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-11-24/readme.md#hike_datacsv (https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-11-24/readme.md#hike_datacsv)

Question: What is a suitable number of ratings to classify trail difficulty for the hiking trails in Washington state?

Introduction: The question aims to use the metrics provided in the dataset to categorize the trails into degree of difficulty. We are working with the `hike_data_new` dataset which is a subset of the `hike_data` dataset. In the `hike_data_new` dataset we have 1958 records of individual hikes in Washington state, with each row containing the information of a unique hike, while the columns have information on name, location, length of the trail, elevation gain, max elevation, notable features and a short description.

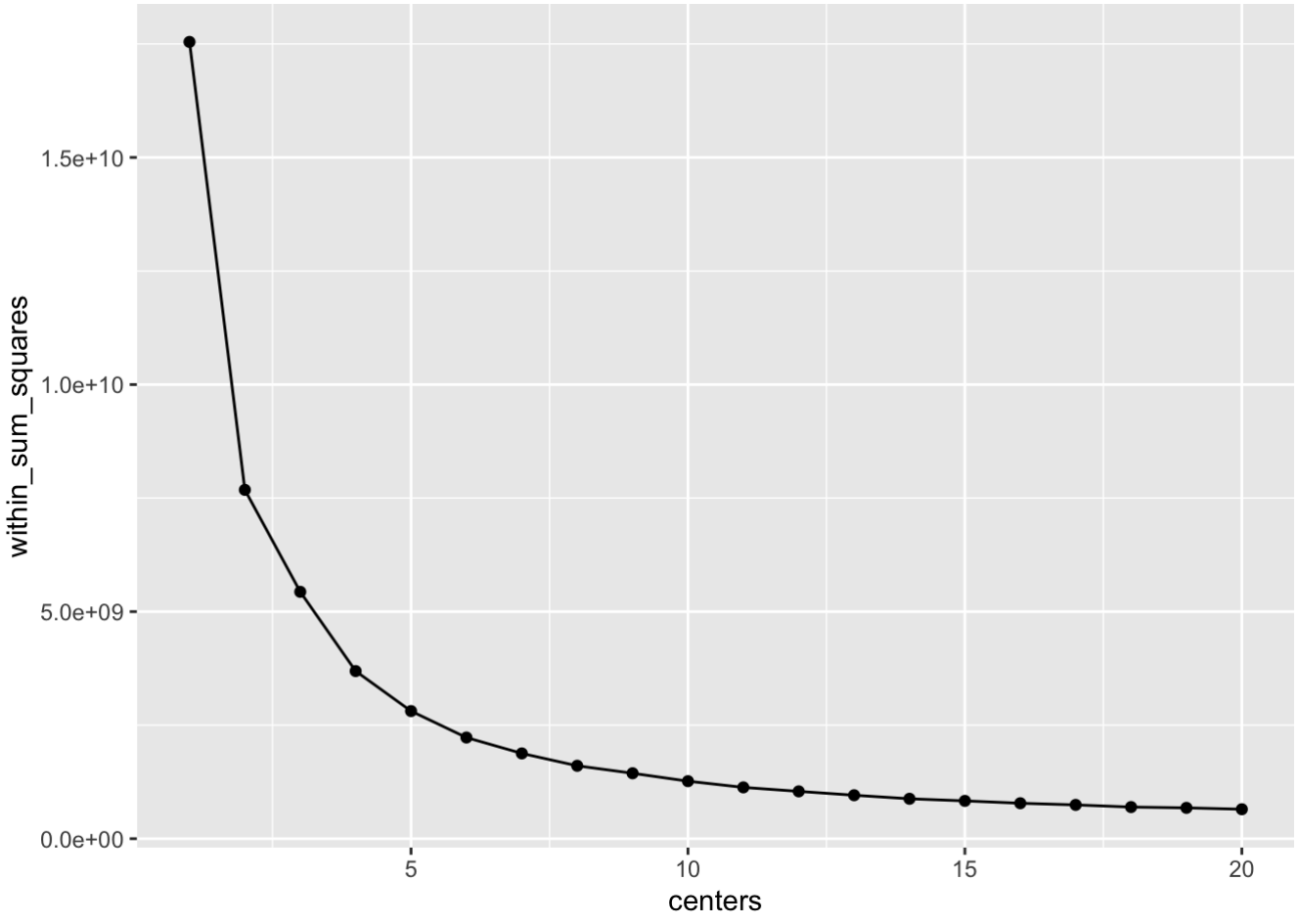
To answer this question, we will work with three variables, the trail's length in miles (column `length`), the elevation gain of the trail in feet above sea level (column `gain`) and the highest point of elevation on the hike (column `highpoint`). All three variables are numeric with length ranging from a short 0.10 miles to 150, elevation gain from flat to 30,000 FT, and highest point raging from sea level to over 12,000 FT.

Approach: We approach the question by first performing K - means on a range of 1 cluster point to 20 cluster points. We will then graph a scree plot and look for an elbow in the resultant line. This elbow represents the number where signicant reduction in sum of squares error has occurred while the addition of subsequent clusters provides less error reduction.

Finally, we will scatterplot with elevation gain on the x-axis and trail length on the y-axis with points colored by cluster to inspect the results (the number of clusters determined by the "elbow point" as determined from the previous scree plot). We should see distinct groupings of largely the same colors. This inspection should allow us to answer the question.

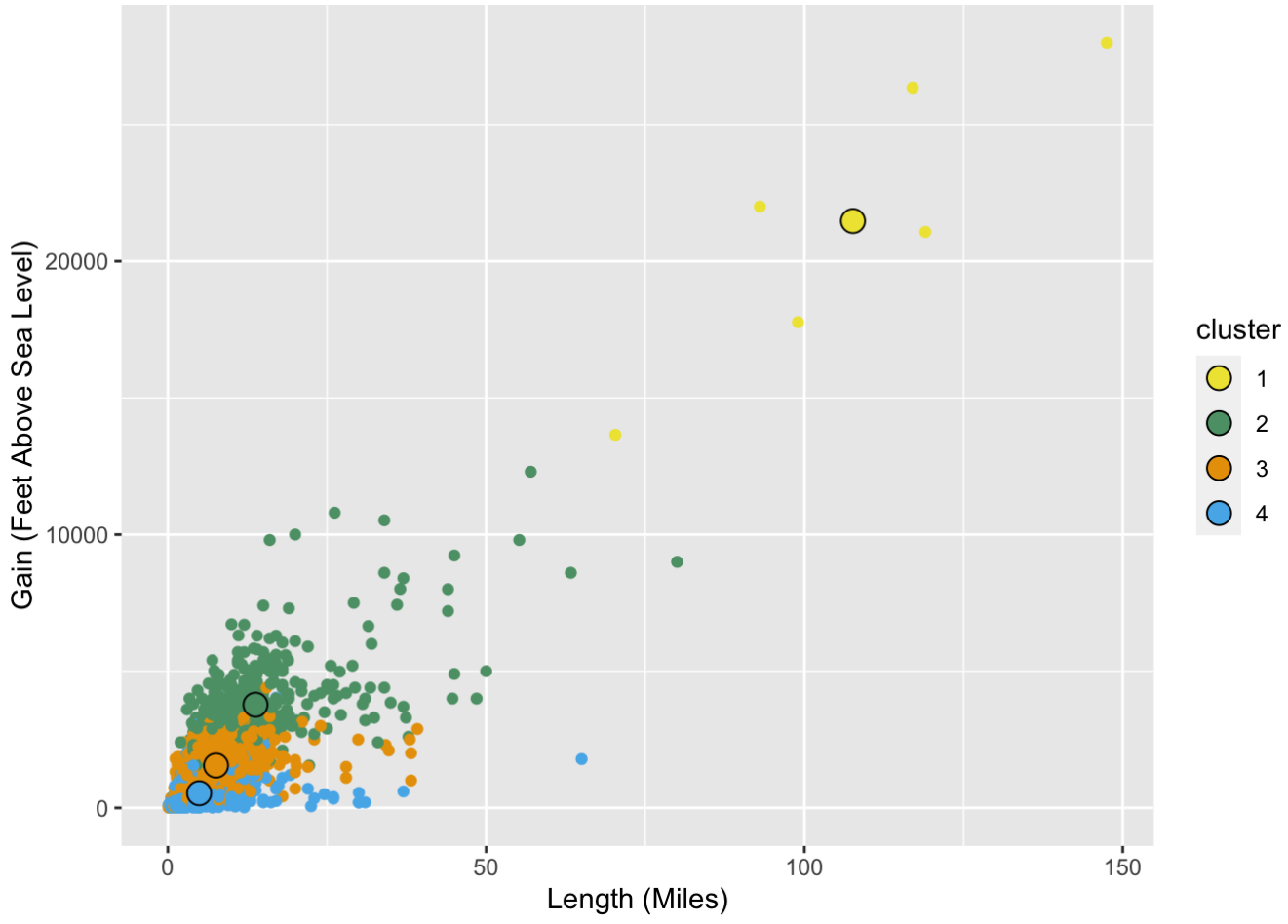
Analysis: We first perform the k-means clustering at for number of clusters 1 through 20 and plot the resultant scree plot.

```
# function to calculate within sum squares
calc_withinss <- function(data, centers) {
  km_fit <- select(data, where(is.numeric)) %>%
    kmeans(centers = centers, iter.max = 20, nstart = 10)
  km_fit$tot.withinss
}
tibble(centers = 1:20) %>%
  mutate(
    within_sum_squares = map_dbl(
      centers, ~calc_withinss(hike_data_new, .x)
    )
  ) %>%
  ggplot() +
  aes(centers, within_sum_squares) +
  geom_point() +
  geom_line()
```



With a significant elbow occurring in the scree plot at K = 4, we now plot our scatterplot with data colored by the 4 cluster points.

```
colors <- c("#EFE441", "#5C9E76", "#E8A003", "#56B5EA")
# run kmeans clustering
km_fit <- hike_data_new %>%
  select(where(is.numeric)) %>%
  kmeans(centers = 4, iter.max = 20, nstart = 10)
# plot
plot1 <- km_fit %>%
  # combine with original data
  augment(hike_data_new) %>%
  ggplot() +
  aes(x = length, y = gain, color = colors) +
  geom_point(
    aes(color = .cluster)
  ) +
  geom_point(
    data = tidy(km_fit),
    aes(fill = cluster),
    shape = 21, color = "black", size = 4
  ) +
  scale_color_manual(values = colors) +
  scale_fill_manual(values = colors) +
  labs(x = "Length (Miles)", y = "Gain (Feet Above Sea Level)") +
  guides(color = "none")
plot1
```



Discussion: Looping through the values of k for k means clustering performed as expected. The resultant scree graph showed significant improvement in mean squared error from K = 1 through K = 4 and began to level off after K = 4. Thus we can infer that the number of clusters belongs at 4. Moving on to the resultant graph we see all data points plotted length vs elevation gain. The most extreme trails, the significant outliers greater than 80 miles in length and elevation gain above 10,000 FT are all clustered together. Like wise the scattered data with elevation climb over 5,000 FT are all clustered together. The easiest two levels are all inside of 35 miles long and under 4,000 FT of elevation gain with the exception of one outlier.

As a trail rating system a degree of 4 would suffice. Most importantly, the last category is saved exclusively for the 5-10 extreme trails and the second to last category warns against both long trails over 30 miles and climbs with over 3,000 FT elevation gain. The state of Washington is likely most concerned with individuals using those trails unprepared and wants to ensure proper labeling. The bottom two categories differentiate between the vast number of shorter/closer to sea levels trails.

Keep in mind there are 3 dimensions to the clustering. Length of the trail, elevation gain AND elevation of highest point. If we could graph the resultant clustering on 3 deminsions the result would be even more clear. I did not graph on 3-D due to the need for additional R-packages that could result in the markdown not running when graded.