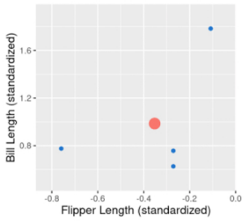
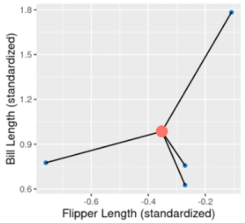
Clustering

WWSD/S^2





$$\mu_x = \frac{1}{4}(x_1 + x_2 + x_3 + x_4) \quad \mu_y = \frac{1}{4}(y_1 + y_2 + y_3 + y_4).$$

$$S^2 = ((x_1 - \mu_x)^2 + (y_1 - \mu_y)^2) + ((x_2 - \mu_x)^2 + (y_2 - \mu_y)^2) + ((x_3 - \mu_x)^2 + (y_3 - \mu_y)^2) + ((x_4 - \mu_x)^2 + (y_4 - \mu_y)^2).$$

Total WSSD - Sum of all the WSSD

Clustering Algorithm Steps:

1. **Center Update** - Compute the center of each cluster.
2. **Label Update** - Reassign each data point to the cluster with the nearest center.

Scaling Data

```
standardized_data <- not_standardized_data %>%
                        mutate(across(everything(), scale))
```

Finding K

```
penguin_clust_ks <- tibble(k = 1:9)

penguin_clust_ks <- tibble(k = 1:9) %>%
                        rowwise() %>%
                        mutate(penguin_clusts = list(kmeans(standardized_data, nstart = 10, k)),
                                glanced = list(glance(penguin_clusts)))

clustering_statistics <- penguin_clust_ks %>%
                          unnest(glanced)

elbow_plot <- ggplot(clustering_statistics, aes(x = k, y = tot.withinss)) +
                geom_point() +
                geom_line() +
                xlab("K") +
                ylab("Total within-cluster sum of squares") +
                scale_x_continuous(breaks = 1:9) +
                theme(text = element_text(size = 12))
```

Performing Kmeans

```
set.seed(100)
penguin_clust <- kmeans(standardized_data, centers = 3)

clusted_data <- augment(penguin_clust, standardized_data)

cluster_plot <- cluster_data %>%
          ggplot(aes(x = flipper_length_mm, y = bill_length_mm, color = .cluster), size = 2) +
             geom_point() +
             labs(x = "Flipper Length (standardized)",
                  y = "Bill Length (standardized)",
                  color = "Cluster") +
          scale_color_manual(values = c("dodgerblue3", "darkorange3", "goldenrod1")) +
          theme(text = element_text(size = 12))
```

Useful functions

pull() - Pulls out a single column from the dataframe.
pluck() - Allows an easiwer way of selecting elements in a list.