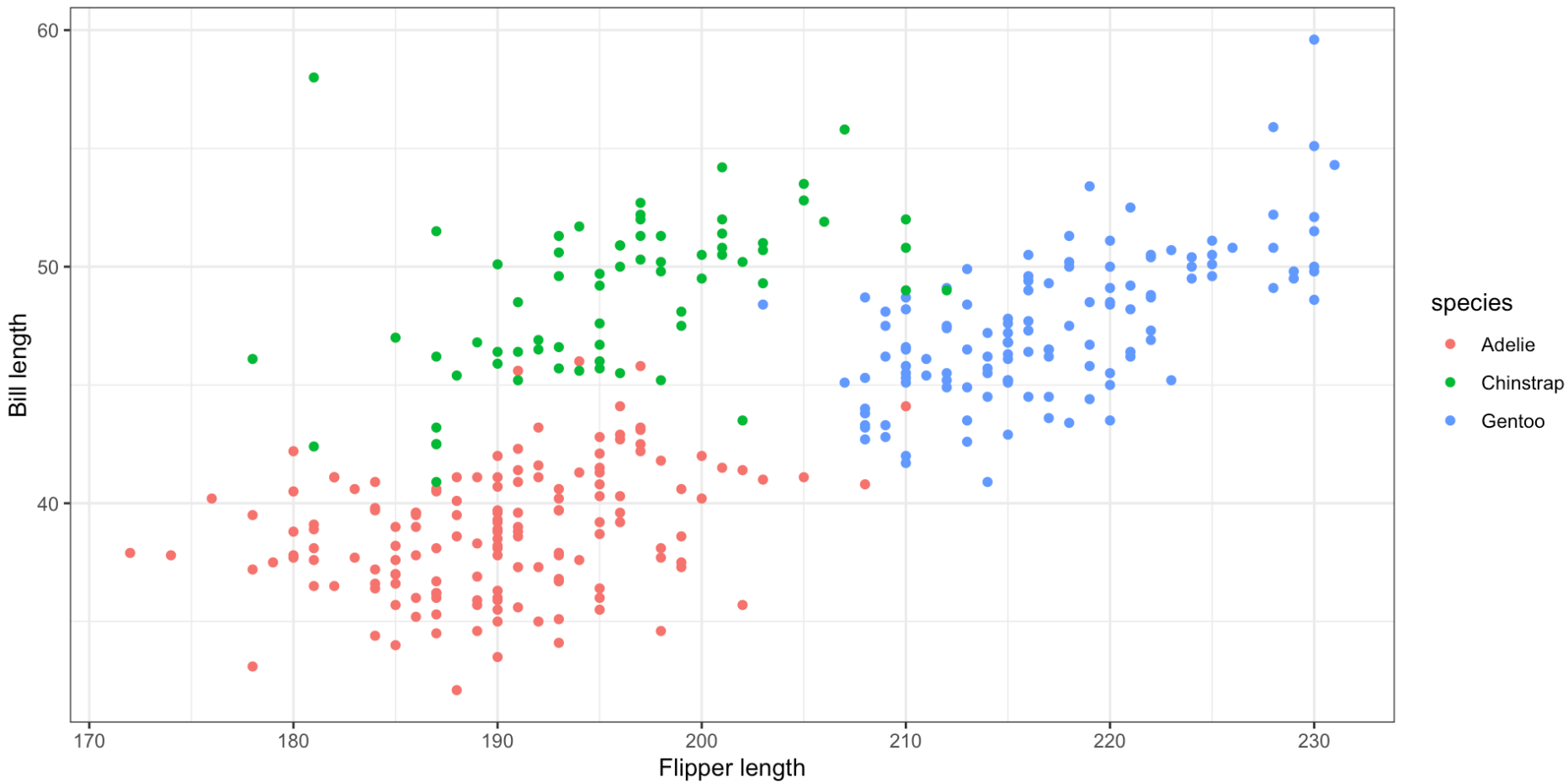
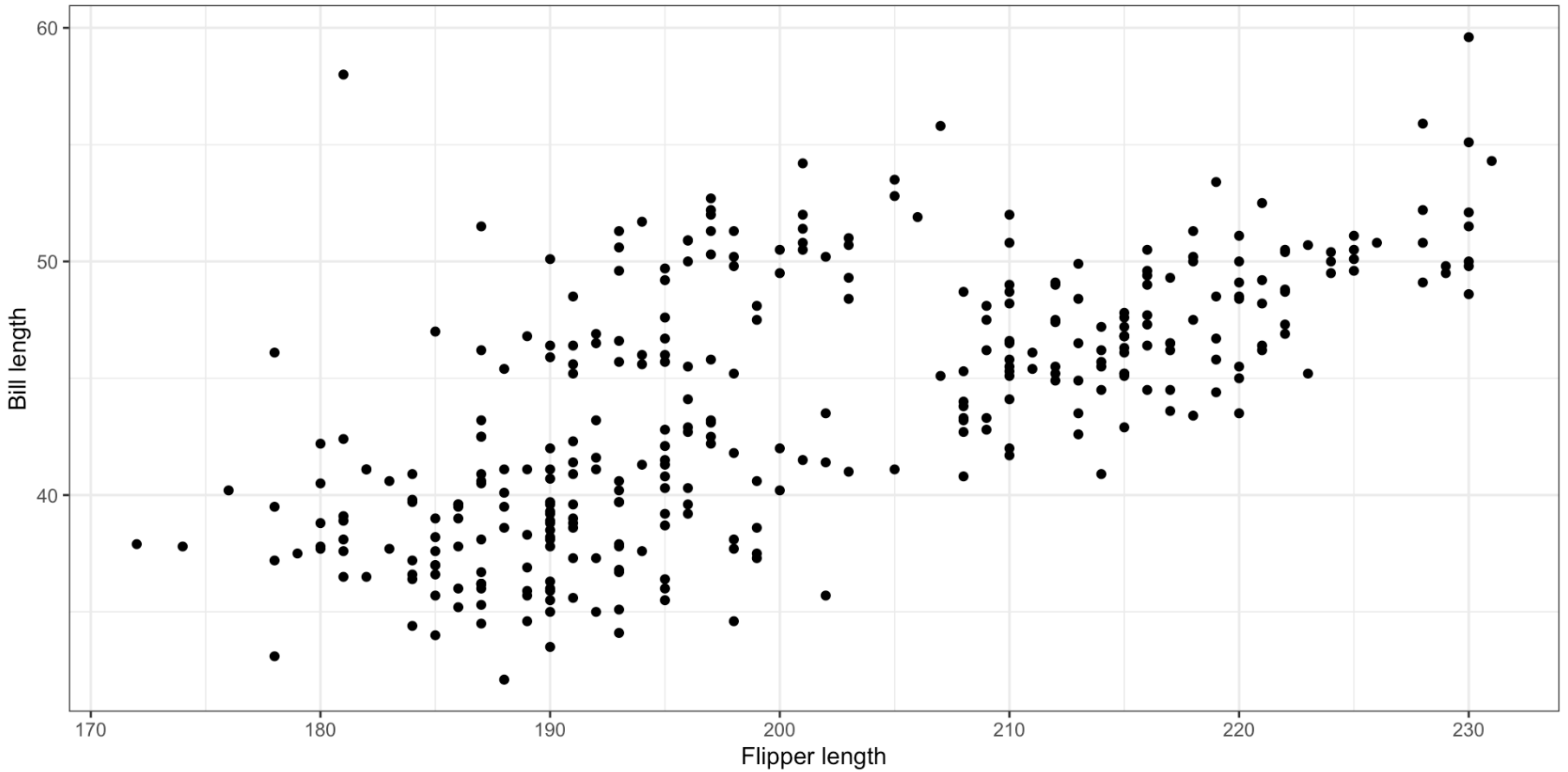
Lecture 27: Implementing statistical methods

Penguins data

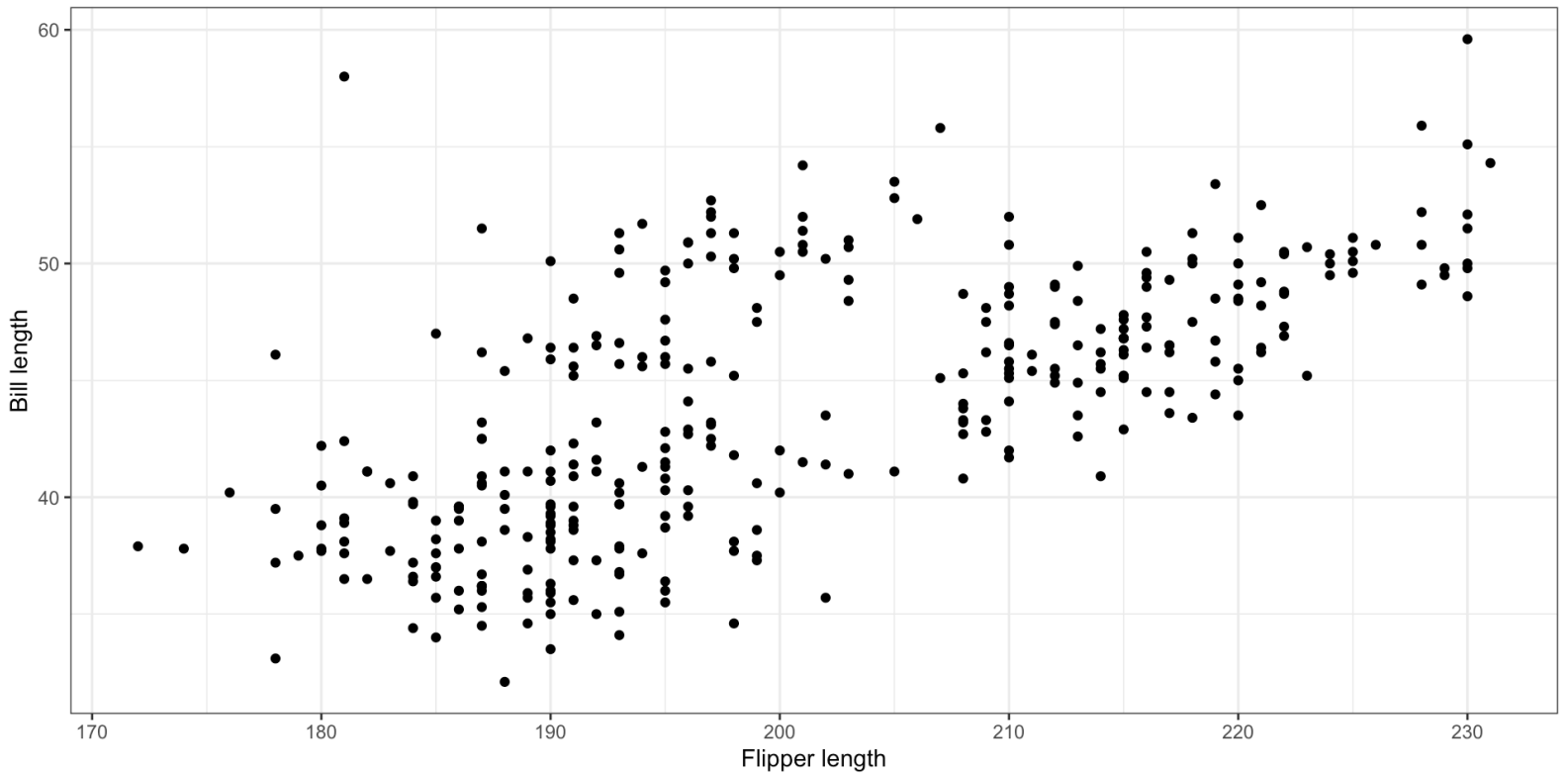


Penguins data

What if we didn't know the species?

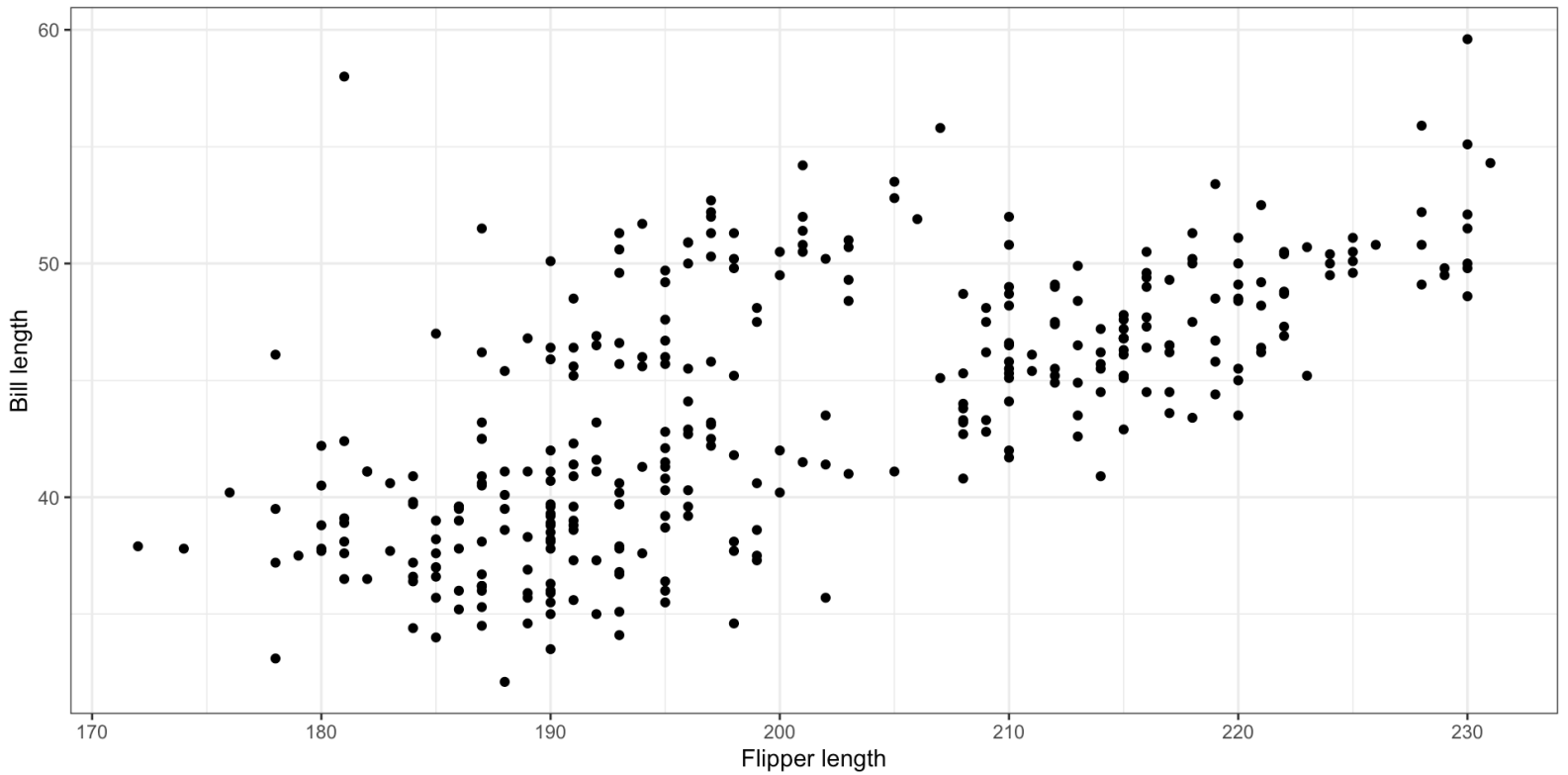


Though excercise



With your neighbor, discuss how you might identify potential groups in this scatterplot, *without* knowing species information.

Clustering

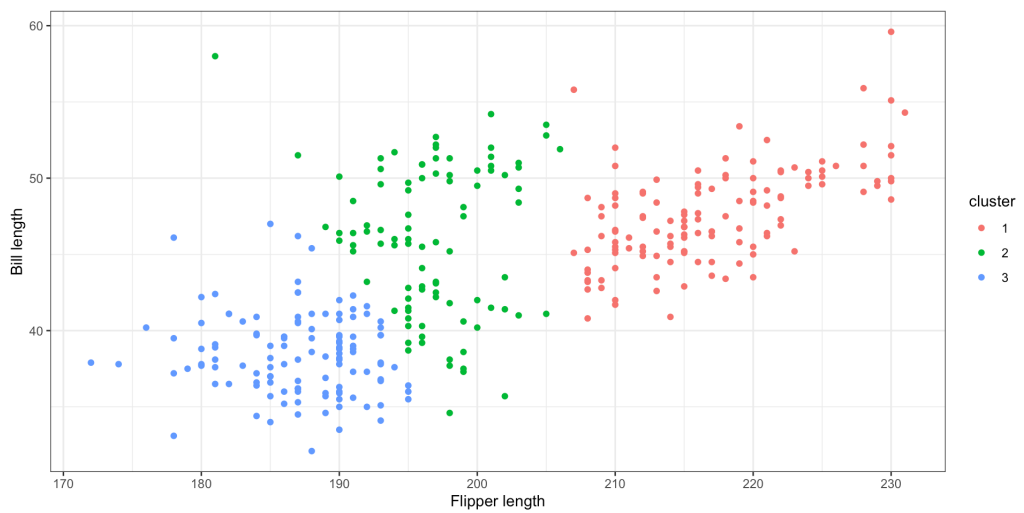


Clustering is the process of dividing data into groups based on the observed variables.

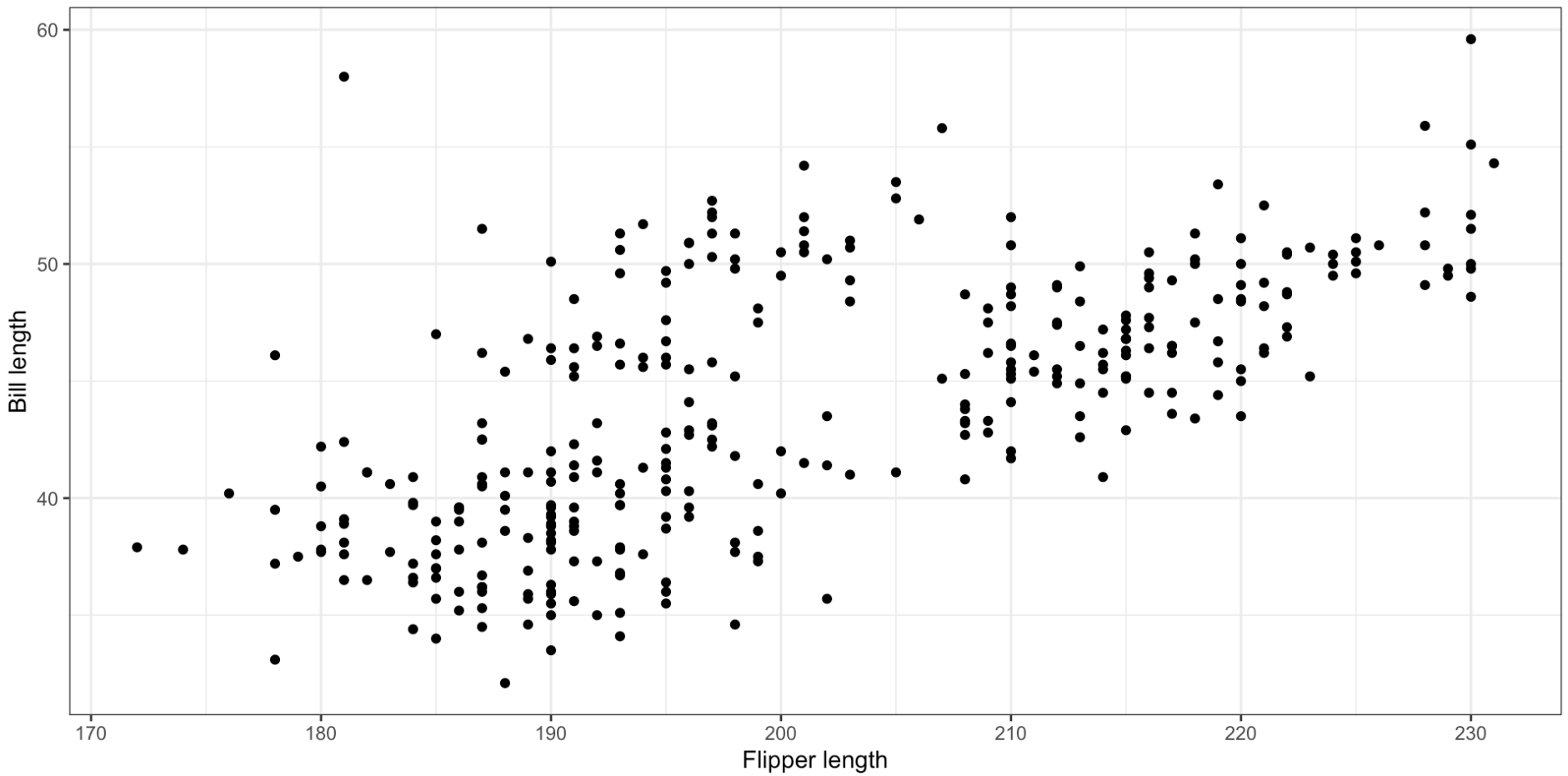
k-means clustering

```r
# cluster with k = 3
clusters <- kmeans(df, 3, algorithm = "Lloyd", iter.max=20)$cluster

df |>
  mutate(cluster = as.factor(clusters)) |>
  ggplot(aes(x = flipper_length_mm, y = bill_length_mm, color = clust
  geom_point() +
  labs(x = "Flipper length", y = "Bill length") +
  theme_bw()
```



k-means clustering



k-means clustering

Recommendations

- Before putting code in a function, write it for an example dataset. *Then* generalize
- Start small (one row, one iteration, etc.), then build up
- Work step by step

Step 1: initialization

Initialization: Randomly select k observations from the dataset to be the *initial* group means

```
1 head(df)
```

```
# A tibble: 6 × 2
  flipper_length_mm bill_length_mm
              <int>          <dbl>
1               181           39.1
2               186           39.5
3               195           40.3
4               193           36.7
5               190           39.3
6               181           38.9
```

Question: How would we randomly choose $k = 3$ rows from our dataset?

Step 1: initialization

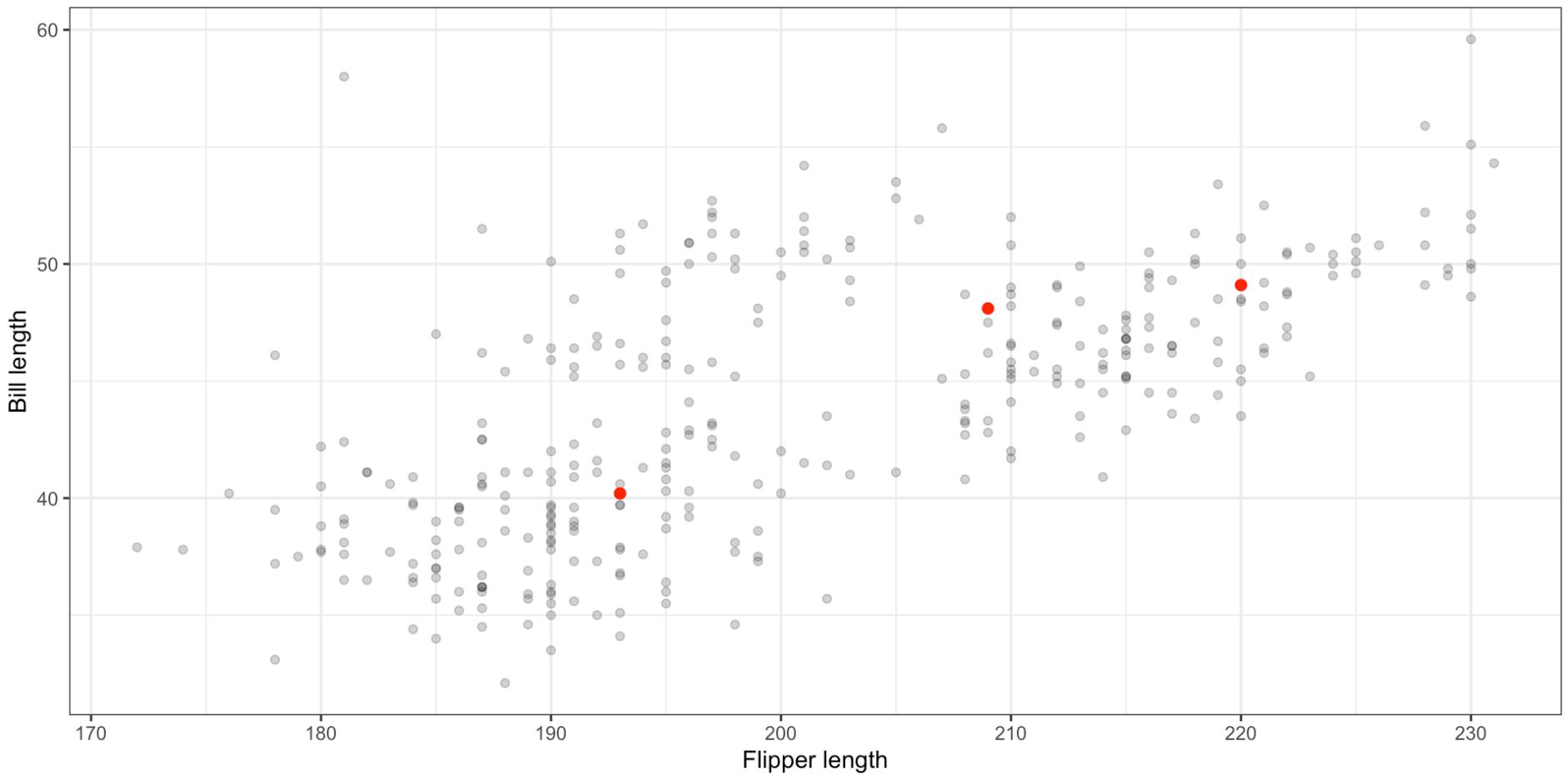
Initialization: Randomly select k observations from the dataset to be the *initial* group means

```r
k <- 3
n <- nrow(df)

# choose the initial means
means <- df[sample(1:n, k),]
means
```

```
# A tibble: 3 × 2
  flipper_length_mm bill_length_mm
              <int>          <dbl>
1               209           48.1
2               220           49.1
3               193           40.2
```

Step 1: initialization



Step 2: calculate distances

Calculate distances: For every observation in the dataset, calculate the distance to each of the k group means

```
1 means
```

```
# A tibble: 3 × 2
  flipper_length_mm bill_length_mm
              <int>           <dbl>
1               209            48.1
2               220            49.1
3               193            40.2
```

```
1 head(df)
```

```
# A tibble: 6 × 2
  flipper_length_mm bill_length_mm
              <int>           <dbl>
1               181            39.1
2               186            39.5
3               195            40.3
4               193            36.7
5               190            39.3
6               181            38.9
```

Step 2: calculate distances

Question: How could we calculate these distances in R?

```
1 means
```

```
# A tibble: 3 × 2
  flipper_length_mm bill_length_mm
              <int>           <dbl>
1               209            48.1
2               220            49.1
3               193            40.2
```

```
1 head(df)
```

```
# A tibble: 6 × 2
  flipper_length_mm bill_length_mm
              <int>           <dbl>
1               181            39.1
2               186            39.5
3               195            40.3
4               193            36.7
5               190            39.3
6               181            38.9
```

Step 2: calculate distances

```
means[1,]
```

```
# A tibble: 1 × 2
  flipper_length_mm bill_length_mm
              <int>          <dbl>
1               209           48.1
```

```
head(df)
```

```
# A tibble: 6 × 2
  flipper_length_mm bill_length_mm
              <int>          <dbl>
1               181           39.1
2               186           39.5
3               195           40.3
4               193           36.7
5               190           39.3
6               181           38.9
```

```
sweep(df, 2, means[1,])
```

```
     flipper_length_mm bill_length_mm
[1,]               -28           -9.0
[2,]               -23           -8.6
[3,]               -14           -7.8
[4,]               -16          -11.4
[5,]               -19           -8.8
[6,]               -28           -9.2
```

```
 [7,]          -14          -8.9
 [8,]          -16         -14.0
 [9,]          -19          -6.1
[10,]          -23         -10.3
[11,]          -29         -10.3
[12,]          -27          -7.0
[13,]          -18          -9.5
```

Step 2: calculate distances

```
1  sweep(df, 2, means[1,])
```

```
     flipper_length_mm bill_length_mm
[1,]               -28            -9.0
[2,]               -23            -8.6
[3,]               -14            -7.8
```

Question: Now how do I get the distances?

Step 2: calculate distances

```
1 sweep(df, 2, means[1,])
```

```
     flipper_length_mm bill_length_mm
[1,]               -28            -9.0
[2,]               -23            -8.6
[3,]               -14            -7.8
```

```
1 rowSums(sweep(df, 2, means[1,])^2)
```

```
[1] 865.00 602.96 256.84
```

So far

```
k <- 3
n <- nrow(df)

# choose the initial means
means <- df[sample(1:n, k),]

# calculate distances for the first group mean
rowSums(sweep(df, 2, means[1,])^2)
```

Question: How would we modify this code to get distances for *each* group mean?

Step 2: calculate distances

```r
dists <- matrix(nrow = n, ncol = k)
for(i in 1:k){
  dists[,i] <- rowSums((sweep(df, 2, means[i,]))^2)
}

head(dists)
```

```
       [,1]    [,2]   [,3]
[1,] 865.00 1621.00 145.21
[2,] 602.96 1248.16  49.49
[3,] 256.84  702.44   4.01
[4,] 385.96  882.76  12.25
[5,] 438.44  996.04   9.81
[6,] 868.64 1625.04 145.69
```

Step 3: Assign points to the nearest group mean

```
1 head(dists)
```

```
       [,1]    [,2]   [,3]
[1,] 865.00 1621.00 145.21
[2,] 602.96 1248.16  49.49
[3,] 256.84  702.44   4.01
[4,] 385.96  882.76  12.25
[5,] 438.44  996.04   9.81
[6,] 868.64 1625.04 145.69
```

Question: Which group mean does each row get assigned to?

Step 3: Assign points to the nearest group mean

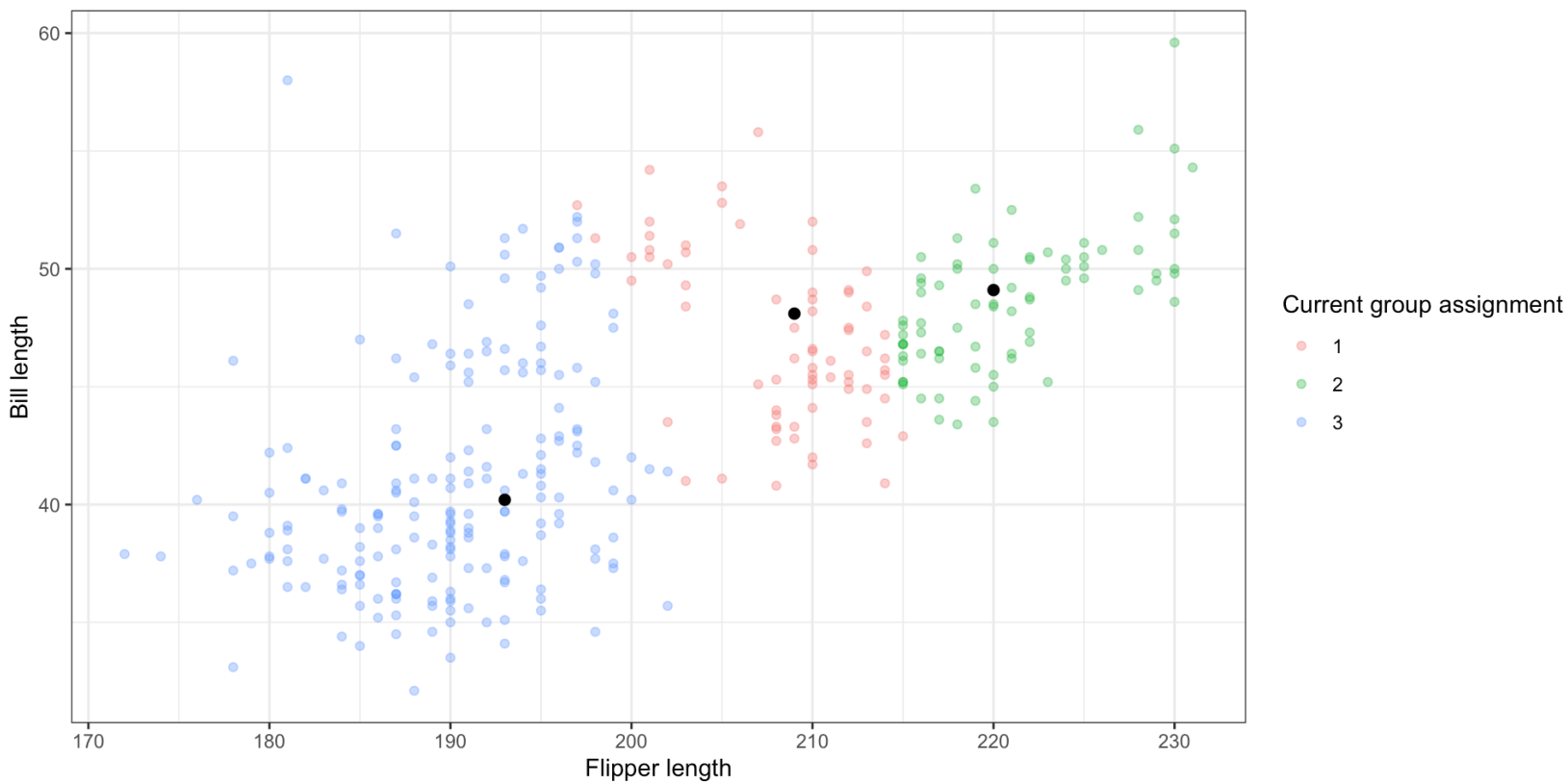
```
1 head(dists)
```

```
       [,1]    [,2]   [,3]
[1,] 865.00 1621.00 145.21
[2,] 602.96 1248.16  49.49
[3,] 256.84  702.44   4.01
[4,] 385.96  882.76  12.25
[5,] 438.44  996.04   9.81
[6,] 868.64 1625.04 145.69
```

Question: How could I write code to assign each row?

Step 3: Assign points to the nearest group mean

```
1  groups <- apply(dists, 1, which.min)
```



Step 4: Re-calculate group means

Update means: For each group, re-calculate the new group mean as the mean of the points assigned to that group

```
1 head(df)
2
3 head(groups)
```

```
     flipper_length_mm bill_length_mm
[1,]               190            38.8
[2,]               220            49.1
[3,]               196            50.0
[4,]               197            52.2
[5,]               209            42.8
[6,]               210            42.0
```

```
[1] 3 2 3 3 1 1
```

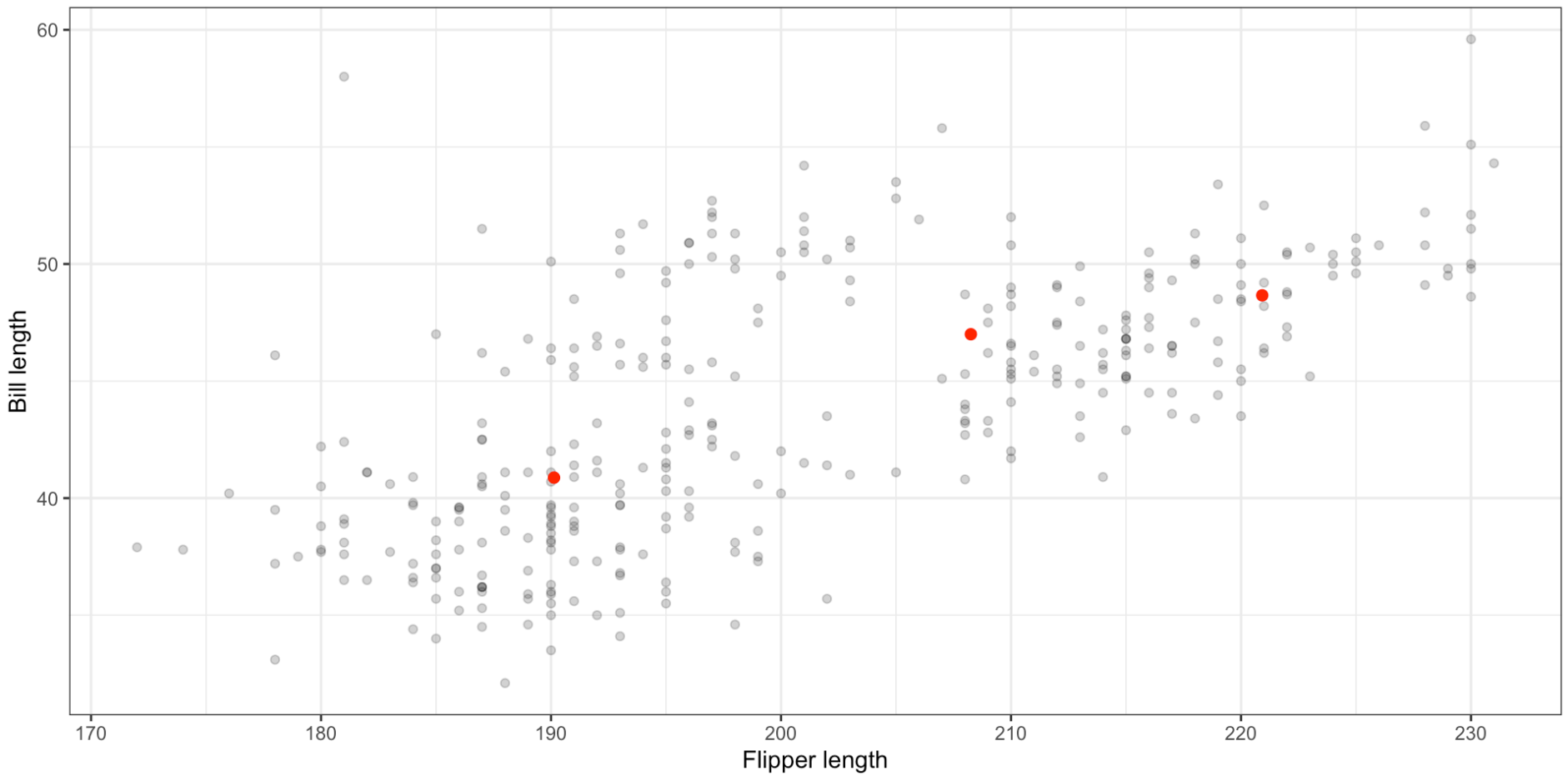
Step 4: Re-calculate the group means

```r
for(i in 1:k){
  means[i,] <- colMeans(df[groups == i,])
}

means
```

```
     flipper_length_mm bill_length_mm
[1,]          208.2535       47.00000
[2,]          220.9231       48.66026
[3,]          190.1295       40.87461
```

Step 4: Re-calculate the group means



So far:

```r
# choose the initial means
means <- df[sample(1:n, k),]

# calculate distances
dists <- matrix(nrow = n, ncol = k)
for(i in 1:k){
  dists[,i] <- rowSums((sweep(df, 2, means[i,]))^2)
}

# assign groups
groups <- apply(dists, 1, which.min)

# update means
for(i in 1:k){
  means[i,] <- colMeans(df[groups == i,])
}
```

Question: What should we do next?