ARB,CHI,BUR

2023-04-27

Loading Libraries and Data

First, we load our libraries:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggrepel)
library(broom)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(purrr)
```

Now we need to import our data set, and create data sets containing just data on McDonalds, Sonic and DQ respecively:

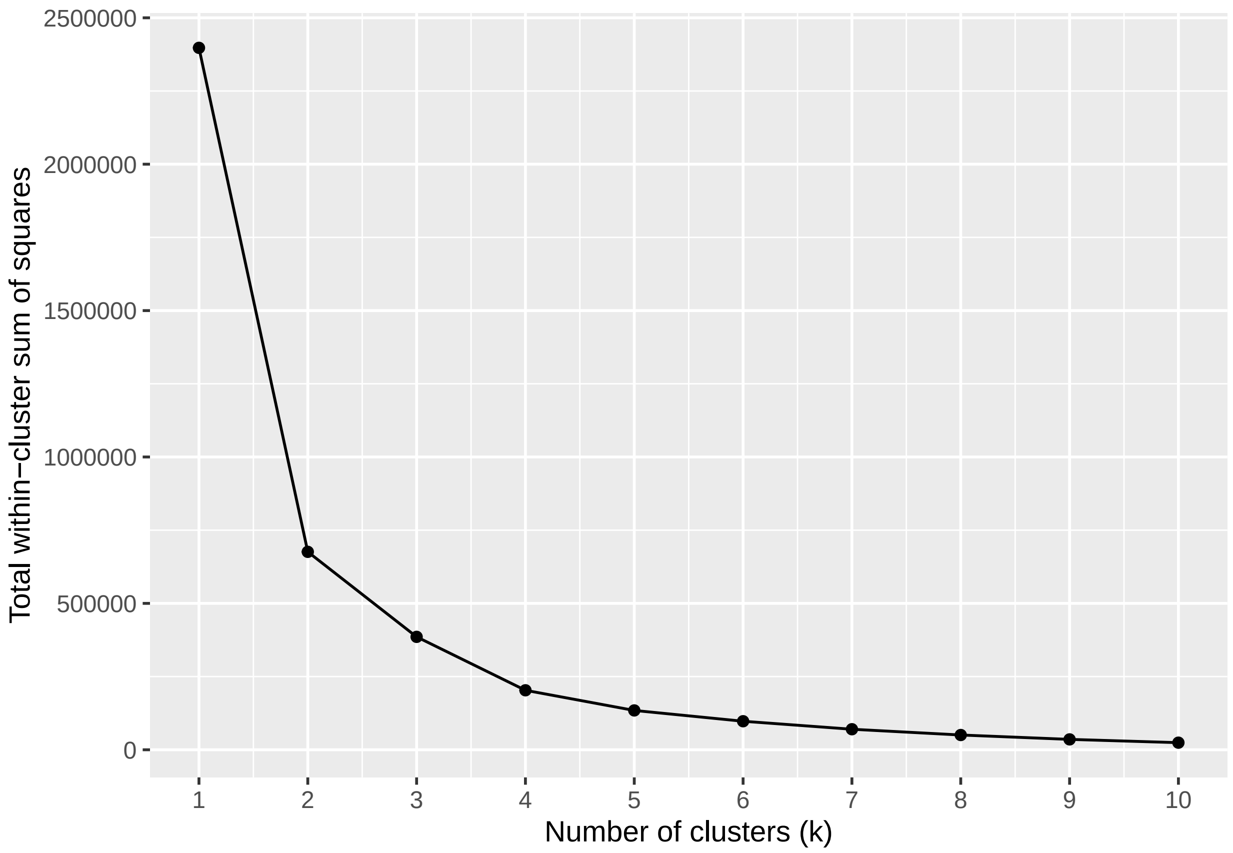
```
#Loading in the CSV
nutritional_data <-  read.csv("data/nutritioninfo.csv")

#Arby's Data Set For 2-way clusters
arb_nutrition <- nutritional_data %>%
  filter(restaurant == "Arbys")%>%
  select(item, protein, calories)

#Arby's Data Set for all variables
arb_nutrition_data <- nutritional_data %>%
  filter(restaurant == "Arbys")

#Burger King Data Set for 2-way clusters
bk_nutrition <- nutritional_data %>%
```

```r
  filter(restaurant == "Burger King")%>%
  select(item, protein, calories)

#Burger King Data Set for all variables
bk_nutrition <- nutritional_data %>%
  filter(restaurant == "Burger King")

#Chick Fila Data set
chf_nutrition <- nutritional_data %>%
   filter(restaurant == "Chick Fil-A")%>%
  select(item, protein, calories)

#Chick Fila Data Set for all variables
chf_nutrition_data <- nutritional_data %>%
   filter(restaurant == "Chick Fil-A")
```

Arby's K-Means Clustering

Now, We are going to make elbow plots to decide how many clusters we want for each Arbys:

```r
arb_clusters_data <- arb_nutrition %>%
  select(calories, protein)%>%
  drop_na

# Iterate through clustering algorithm for 10 different values of k
elbow_plot1 <- tibble(k = 1:10) %>%
  mutate(
    # List-column of 10 kmeans objects
    # (apply `kmeans()` to each value of `k`)
    kmeans_arb = purrr::map(k, ~kmeans(arb_clusters_data, .x, nstart = 20)),
    # List-column of "glanced" model summaries for each kmeans object
    # (apply `glance()` to each corresponding result after running `kmeans()`)
    glanced = purrr::map(kmeans_arb, glance)) %>%
  # Turn `glanced` list-column into regular tibble columns
  unnest(cols = c(glanced))

# Construct elbow plot
ggplot(elbow_plot1, aes(x = k, y = tot.withinss)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = 1:10) +
  labs(x = "Number of clusters (k)",
       y = "Total within-cluster sum of squares")
```

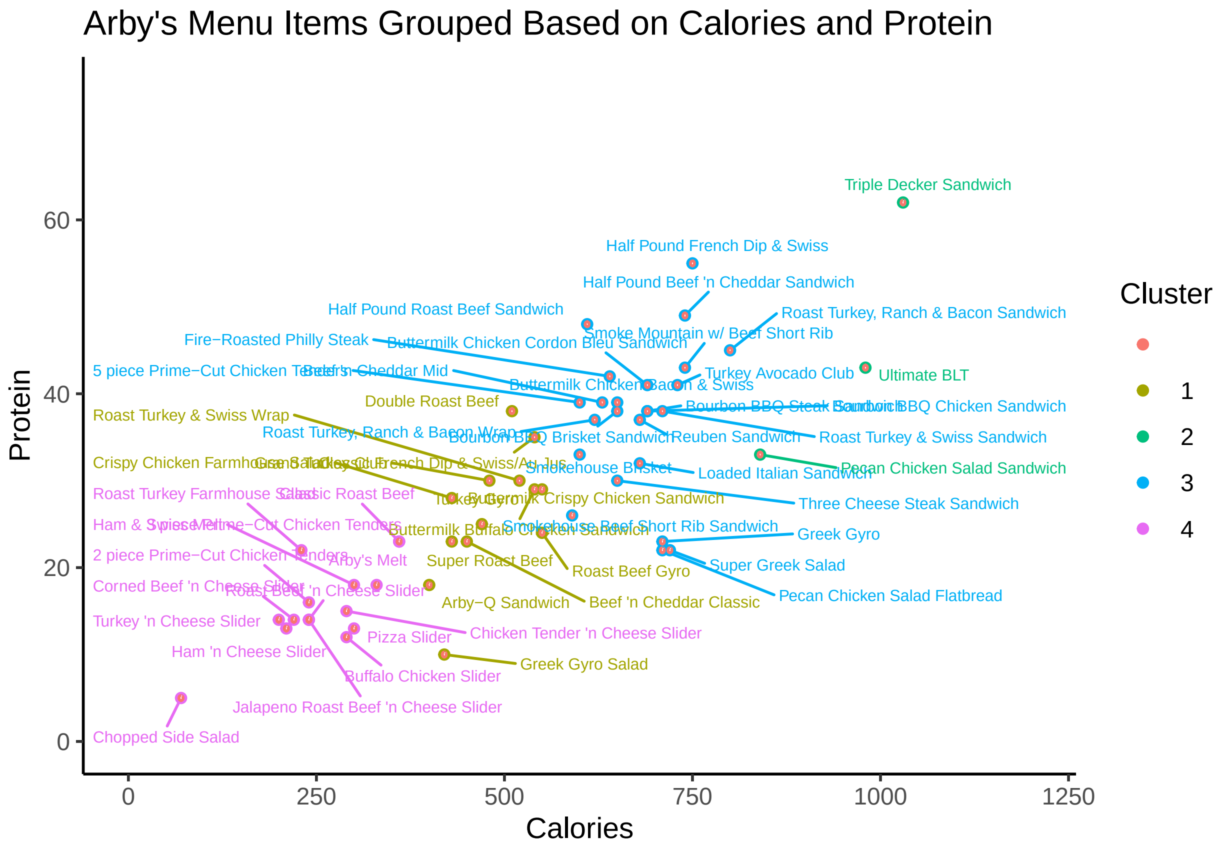


Based on the Elbow Plot Above, I am going to use four clusters, because there is still a large jump from three clusters to four, but then a very small jump between four clusters and five clusters.

Now we are going to cluster based on calories and protein, and create a graph and a list of which items are in which clusters

```
# set the seed for reproducibility
set.seed(23)

# Perform k-means clustering with k = 4
arb_clusters_4 <- arb_clusters_data %>%
  kmeans(centers = 4, nstart = 20)

arb_clusters_c4 <- augment(arb_clusters_4, arb_nutrition)

ggplot(arb_clusters_c4, aes(x = calories, y = protein)) +
  geom_point(aes(color = .cluster)) +
  geom_text_repel(aes(label = item, color = .cluster),
                  size = 2, max.overlaps = 200, show.legend = FALSE) +
  scale_x_continuous(breaks = scales::breaks_width(200)) +
  scale_y_continuous(breaks = scales::breaks_width(25)) +
  # Add centroid labels to plot
  geom_label(data = arb_clusters_c4, aes(label = "", color = ""),
             size = 0.1,
             label.r = unit(0.05, "lines"),
             label.size = 0.5,
             label.padding = unit(0.05, "lines"),
             show.legend = FALSE) +
  labs(title = "Arby's Menu Items Grouped Based on Calories and Protein",
       x = "Calories",
       y = "Protein",
       color = "Cluster") +
  theme_classic() +
  xlim(c(0,1200)) +
      ylim(c(0,75))
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```



```
#Get Important Data Points and What items are in which cluster
arb_summaries <- tidy(arb_clusters_4)
arb_summaries
```

```
## # A tibble: 4 x 5
##   calories protein  size withinss cluster
##      <dbl>   <dbl> <int>    <dbl> <fct>
## 1     484.    26.3    13   35928. 1
## 2     950     46       3   19834  2
## 3     678.    37.3    24   71350. 3
## 4     259.    15.6    15   76085. 4
```

```
arb_clusters_4$centers
```

```
##   calories  protein
## 1 483.8462 26.30769
## 2 950.0000 46.00000
## 3 678.3333 37.29167
## 4 258.6667 15.60000
```

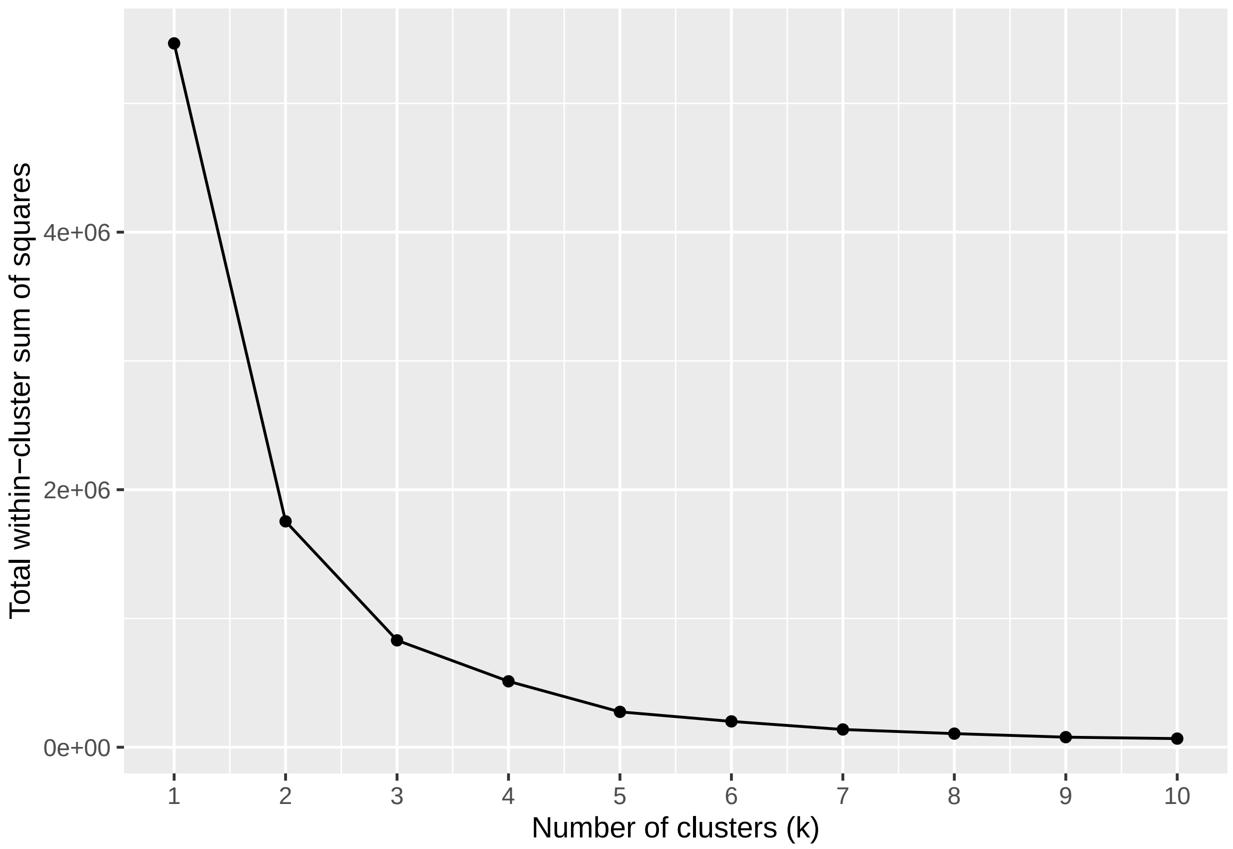
```
cluster_list <- split(arb_nutrition$item, arb_clusters_4$cluster)
cluster_list
```

```
## $‘1‘
##  [1] "Arby-Q Sandwich"                     "Beef ’n Cheddar Classic"
##  [3] "Buttermilk Buffalo Chicken Sandwich" "Buttermilk Crispy Chicken Sandwich"
##  [5] "Classic French Dip & Swiss/Au Jus"   "Double Roast Beef"
##  [7] "Grand Turkey Club"                   "Roast Beef Gyro"
##  [9] "Roast Turkey & Swiss Wrap"           "Super Roast Beef"
## [11] "Turkey Gyro"                         "Crispy Chicken Farmhouse Salad"
## [13] "Greek Gyro Salad"
##
## $‘2‘
## [1] "Pecan Chicken Salad Sandwich" "Triple Decker Sandwich"
## [3] "Ultimate BLT"
##
## $‘3‘
##  [1] "Beef ’n Cheddar Mid"
##  [2] "Bourbon BBQ Brisket Sandwich"
##  [3] "Bourbon BBQ Chicken Sandwich"
##  [4] "Bourbon BBQ Steak Sandwich"
##  [5] "Buttermilk Chicken Bacon & Swiss"
##  [6] "Buttermilk Chicken Cordon Bleu Sandwich"
##  [7] "Fire-Roasted Philly Steak"
##  [8] "Greek Gyro"
##  [9] "Half Pound Beef ’n Cheddar Sandwich"
## [10] "Half Pound French Dip & Swiss"
## [11] "Half Pound Roast Beef Sandwich"
## [12] "Loaded Italian Sandwich"
## [13] "Pecan Chicken Salad Flatbread"
## [14] "5 piece Prime-Cut Chicken Tenders"
## [15] "Reuben Sandwich"
## [16] "Roast Turkey & Swiss Sandwich"
## [17] "Roast Turkey, Ranch & Bacon Sandwich"
## [18] "Roast Turkey, Ranch & Bacon Wrap"
## [19] "Smoke Mountain w/ Beef Short Rib"
## [20] "Smokehouse Beef Short Rib Sandwich"
## [21] "Smokehouse Brisket"
## [22] "Three Cheese Steak Sandwich"
## [23] "Turkey Avocado Club"
## [24] "Super Greek Salad"
##
## $‘4‘
##  [1] "Arby’s Melt"
##  [2] "Classic Roast Beef"
##  [3] "Ham & Swiss Melt"
##  [4] "2 piece Prime-Cut Chicken Tenders"
##  [5] "3 piece Prime-Cut Chicken Tenders"
##  [6] "Buffalo Chicken Slider"
##  [7] "Chicken Tender ’n Cheese Slider"
##  [8] "Corned Beef ’n Cheese Slider"
##  [9] "Ham ’n Cheese Slider"
## [10] "Jalapeno Roast Beef ’n Cheese Slider"
## [11] "Pizza Slider"
## [12] "Roast Beef ’n Cheese Slider"
## [13] "Turkey ’n Cheese Slider"
## [14] "Chopped Side Salad"
## [15] "Roast Turkey Farmhouse Salad"
```

Now we repeat the same process with Burger King:

First an elbow plot to find the best number of clusters to use

```
bk_clusters_data <- bk_nutrition %>%
  select(calories, protein)%>%
  drop_na

# Iterate through clustering algorithm for 10 different values of k
elbow_plot2 <- tibble(k = 1:10) %>%
  mutate(
    # List-column of 10 kmeans objects
    # (apply `kmeans()` to each value of `k`)
    kmeans_bk = purrr::map(k, ~kmeans(bk_clusters_data, .x, nstart = 20)),
    # List-column of "glanced" model summaries for each kmeans object
    # (apply `glance()` to each corresponding result after running `kmeans()`)
    glanced = purrr::map(kmeans_bk, glance)) %>%
  # Turn `glanced` list-column into regular tibble columns
  unnest(cols = c(glanced))

# Construct elbow plot
ggplot(elbow_plot2, aes(x = k, y = tot.withinss)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = 1:10) +
  labs(x = "Number of clusters (k)",
       y = "Total within-cluster sum of squares")
```

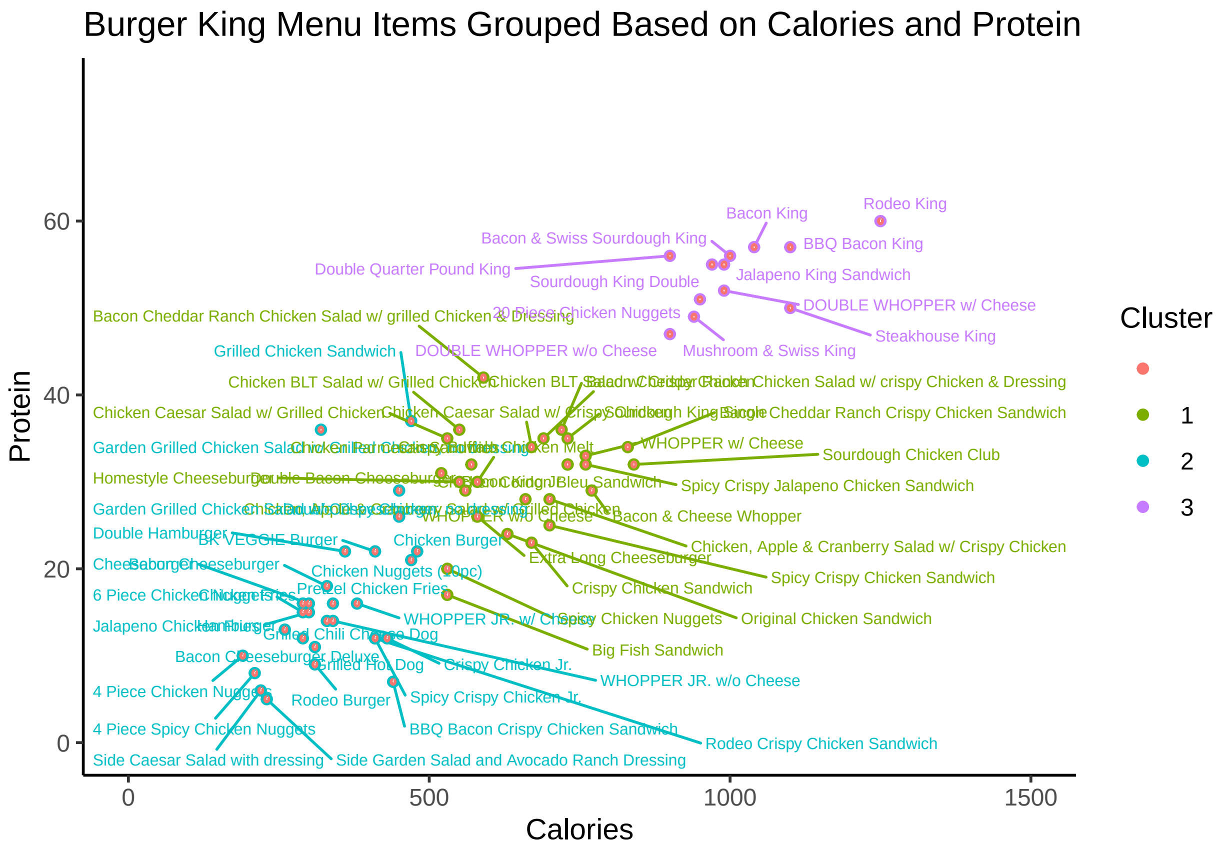


Based on this elbow plot I am going to use three clusters because again there is a large drop in within-cluster sum of squares, and not a very large drop from 3 to four.

Now we are going to cluster based on calories and protein, and create a graph and a list of which items are in which clusters

```
# set the seed for reproducibility
set.seed(23)

# Perform k-means clustering with k = 3
bk_clusters_3 <- bk_clusters_data %>%
  kmeans(centers = 3, nstart = 20)

bk_nutrition <- bk_nutrition %>%
drop_na

bk_clusters_c3 <- augment(bk_clusters_3, bk_nutrition)

ggplot(bk_clusters_c3, aes(x = calories, y = protein)) +
  geom_point(aes(color = .cluster)) +
  geom_text_repel(aes(label = item, color = .cluster),
                  size = 2, max.overlaps = 200, show.legend = FALSE) +
  scale_x_continuous(breaks = scales::breaks_width(200)) +
  scale_y_continuous(breaks = scales::breaks_width(25)) +
  # Add centroid labels to plot
  geom_label(data = bk_clusters_c3, aes(label = "", color = ""),
             size = 0.1,
             label.r = unit(0.05, "lines"),
             label.size = 0.5,
             label.padding = unit(0.05, "lines"),
             show.legend = FALSE) +
  labs(title = "Burger King Menu Items Grouped Based on Calories and Protein",
       x = "Calories",
       y = "Protein",
       color = "Cluster") +
  theme_classic() +
  xlim(c(0,1500)) +
      ylim(c(0,75))
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```



```
#Get Important Data Points and What items are in which cluster
bk_summaries <- tidy(bk_clusters_3)
bk_summaries
```

```
## # A tibble: 3 x 5
##   calories protein  size withinss cluster
##      <dbl>   <dbl> <int>    <dbl> <fct>
## 1     655.    30.4    27  247228. 1
## 2     346.    16.3    29  198721. 2
## 3    1052.    59.9    13  384342. 3
```

```
bk_clusters_3$centers
```

```
##    calories  protein
## 1  654.8148 30.37037
## 2  345.5172 16.27586
## 3 1052.3077 59.92308
```

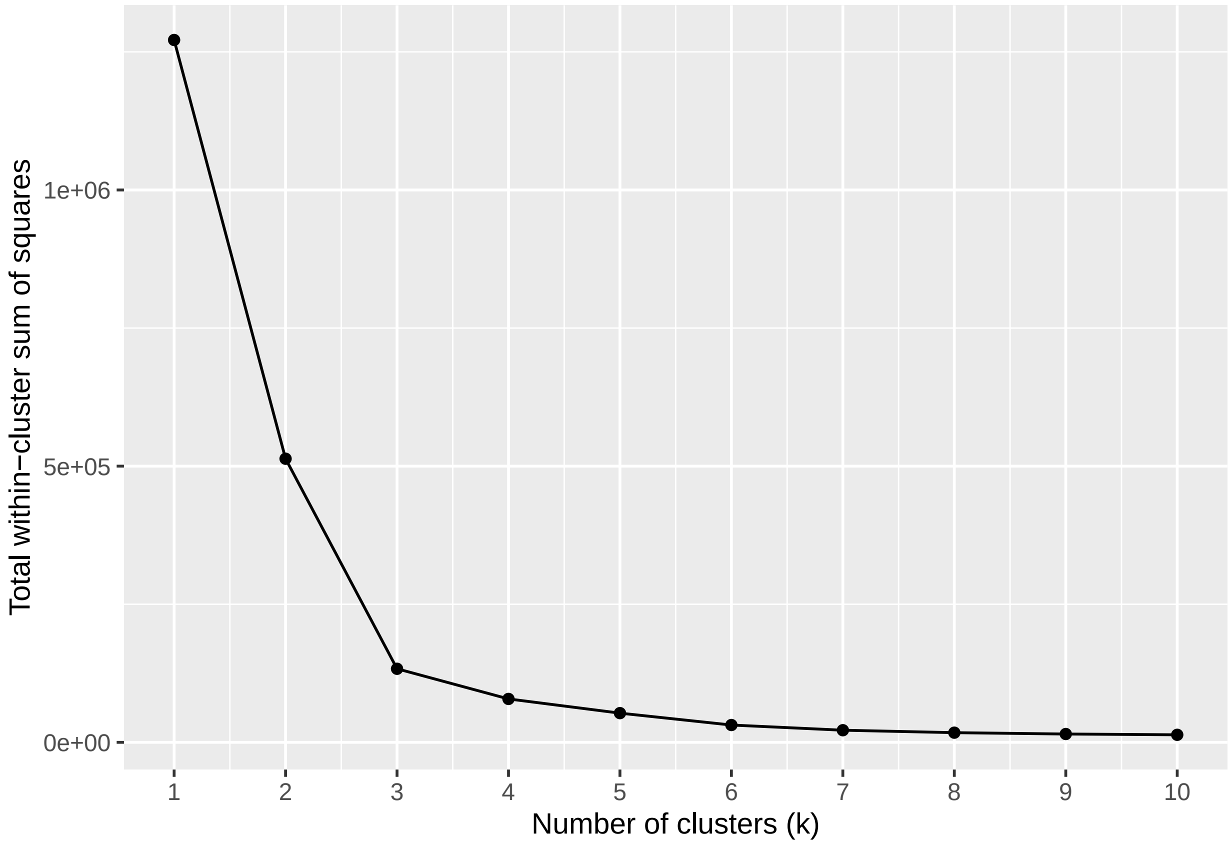
```
cluster_list <- split(bk_nutrition$item, bk_clusters_3$cluster)
cluster_list
```

```
## $‘1‘
##  [1] "Bacon King Jr"
##  [2] "Double Bacon Cheeseburger"
##  [3] "Extra Long Cheeseburger"
##  [4] "Homestyle Cheeseburger"
##  [5] "Sourdough King Single"
##  [6] "Bacon & Cheese Whopper"
##  [7] "WHOPPER w/o Cheese"
##  [8] "WHOPPER w/ Cheese"
##  [9] "Bacon Cheddar Ranch Chicken Salad w/ grilled Chicken & Dressing"
## [10] "Bacon Cheddar Ranch Chicken Salad w/ crispy Chicken & Dressing"
## [11] "Chicken BLT Salad w/ Grilled Chicken"
## [12] "Chicken BLT Salad w/ Crispy Chicken"
## [13] "Chicken Caesar Salad w/ Grilled Chicken"
## [14] "Chicken Caesar Salad w/ Crispy Chicken"
## [15] "Chicken, Apple & Cranberry Salad w/ Grilled Chicken"
## [16] "Chicken, Apple & Cranberry Salad w/ Crispy Chicken"
## [17] "Bacon Cheddar Ranch Crispy Chicken Sandwich"
## [18] "Big Fish Sandwich"
## [19] "Chicken Cordon Bleu Sandwich"
## [20] "Chicken Parmesan Sandwich"
## [21] "Crispy Buffalo Chicken Melt"
## [22] "Crispy Chicken Sandwich"
## [23] "Original Chicken Sandwich"
## [24] "Sourdough Chicken Club"
## [25] "Spicy Chicken Nuggets"
## [26] "Spicy Crispy Chicken Sandwich"
## [27] "Spicy Crispy Jalapeno Chicken Sandwich"
##
## $‘2‘
##  [1] "Bacon Cheeseburger"
##  [2] "Bacon Cheeseburger Deluxe"
##  [3] "Cheeseburger"
##  [4] "Double Cheeseburger"
##  [5] "Double Hamburger"
##  [6] "Hamburger"
##  [7] "Rodeo Burger"
##  [8] "WHOPPER JR. w/o Cheese"
##  [9] "WHOPPER JR. w/ Cheese"
## [10] "Garden Grilled Chicken Salad w/ Grilled Chicken, no dressing"
## [11] "Garden Grilled Chicken Salad w/ Crispy Chicken, no dressing"
## [12] "Side Caesar Salad with dressing"
## [13] "Side Garden Salad and Avocado Ranch Dressing"
## [14] "BBQ Bacon Crispy Chicken Sandwich"
## [15] "BK VEGGIE Burger"
## [16] "Chicken Burger"
## [17] "Chicken Fries"
## [18] "4 Piece Chicken Nuggets"
## [19] "6 Piece Chicken Nuggets"
## [20] "Chicken Nuggets (10pc)"
## [21] "Crispy Chicken Jr."
## [22] "Grilled Chicken Sandwich"
## [23] "Grilled Chili Cheese Dog"
## [24] "Grilled Hot Dog"
## [25] "Jalapeno Chicken Fries"
## [26] "Pretzel Chicken Fries"
## [27] "Rodeo Crispy Chicken Sandwich"
## [28] "4 Piece Spicy Chicken Nuggets"
## [29] "Spicy Crispy Chicken Jr."
##
## $‘3‘
##  [1] "American Brewhouse King"      "Bacon & Swiss Sourdough King"
##  [3] "Bacon King"                   "BBQ Bacon King"
##  [5] "Double Quarter Pound King"    "Jalapeno King Sandwich"
##  [7] "Mushroom & Swiss King"        "Rodeo King"
##  [9] "Sourdough King Double"        "Steakhouse King"
## [11] "DOUBLE WHOPPER w/o Cheese"    "DOUBLE WHOPPER w/ Cheese"
## [13] "20 Piece Chicken Nuggets"
```

Now we are going to do the same process but with Chick Fila:

First an elbow plot to determin number of clusters:

```
chf_clusters_data <- chf_nutrition %>%
  select(calories, protein)%>%
  drop_na

# Iterate through clustering algorithm for 10 different values of k
elbow_plot3 <- tibble(k = 1:10) %>%
  mutate(
    # List-column of 10 kmeans objects
    # (apply `kmeans()` to each value of `k`)
    kmeans_chf = purrr::map(k, ~kmeans(chf_clusters_data, .x, nstart = 20)),
    # List-column of "glanced" model summaries for each kmeans object
    # (apply `glance()` to each corresponding result after running `kmeans()`)
    glanced = purrr::map(kmeans_chf, glance)) %>%
  # Turn `glanced` list-column into regular tibble columns
  unnest(cols = c(glanced))

# Construct elbow plot
ggplot(elbow_plot3, aes(x = k, y = tot.withinss)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = 1:10) +
  labs(x = "Number of clusters (k)",
       y = "Total within-cluster sum of squares")
```



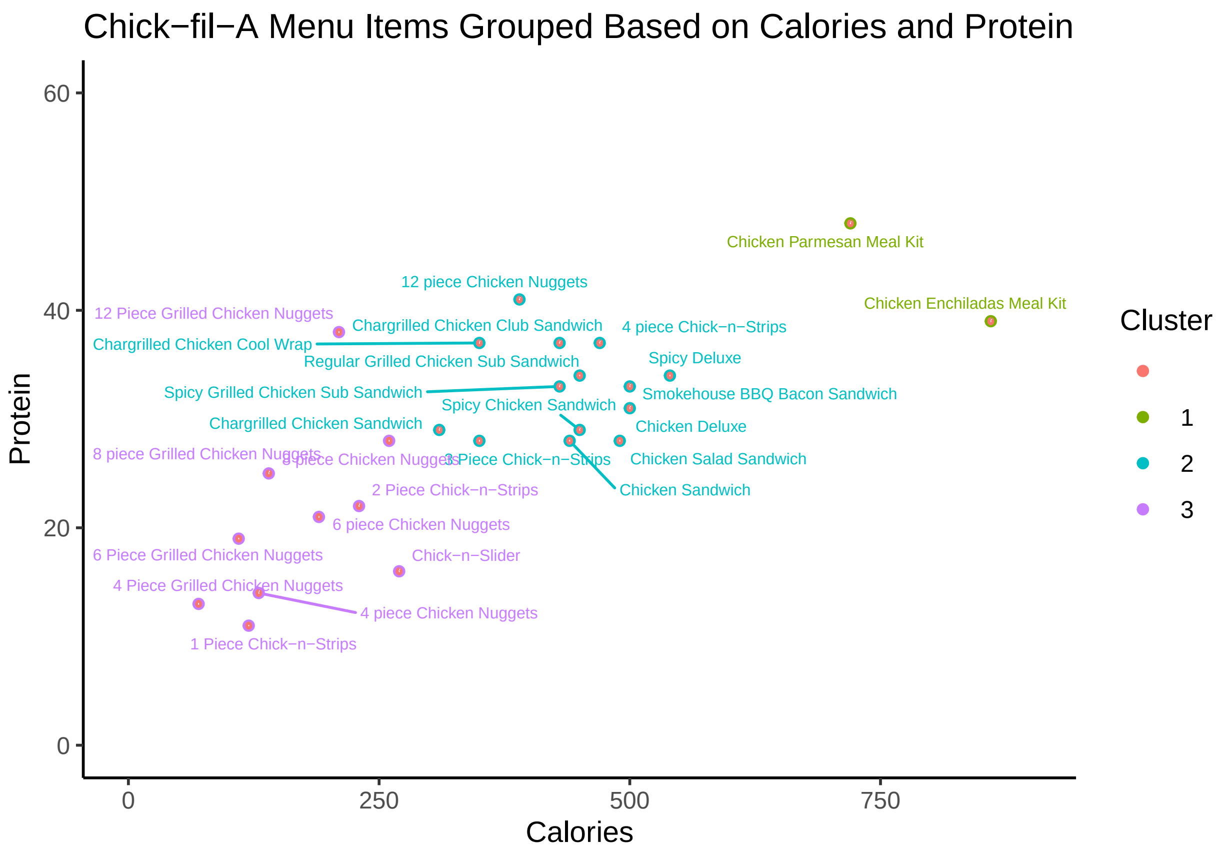
For Chick-Fila I am going to use 3 clusters because while there is not a large jump between 2 and 3 clusters there is a relatively small jump from 3 to 4.

Now we are going to cluster based on calories and protein, and create a graph and a list of which items are in which clusters

```
# set the seed for reproducibility
set.seed(23)

# Perform k-means clustering with k = 3
chf_clusters_3 <- chf_clusters_data %>%
  kmeans(centers = 3, nstart = 20)
```

```r
chf_clusters_c3 <- augment(chf_clusters_3, chf_nutrition)

ggplot(chf_clusters_c3, aes(x = calories, y = protein)) +
  geom_point(aes(color = .cluster)) +
  geom_text_repel(aes(label = item, color = .cluster),
                  size = 2, max.overlaps = 200, show.legend = FALSE) +
  scale_x_continuous(breaks = scales::breaks_width(200)) +
  scale_y_continuous(breaks = scales::breaks_width(25)) +
  # Add centroid labels to plot
  geom_label(data = chf_clusters_c3, aes(label = "", color = ""),
             size = 0.1,
             label.r = unit(0.05, "lines"),
             label.size = 0.5,
             label.padding = unit(0.05, "lines"),
             show.legend = FALSE) +
  labs(title = "Chick-fil-A Menu Items Grouped Based on Calories and Protein",
       x = "Calories",
       y = "Protein",
       color = "Cluster") +
  theme_classic() +
  xlim(c(0,900)) +
      ylim(c(0,60))
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```



```
#Get Important Data Points and What items are in which cluster
chf_summaries <- tidy(chf_clusters_3)
chf_summaries
```

```
## # A tibble: 3 x 5
##   calories protein  size withinss cluster
##      <dbl>   <dbl> <int>    <dbl> <fct>
## 1     850     63.3     3   33801. 1
## 2     436.    32.8    14   56567. 2
## 3     173     20.7    10   42806. 3
```

```
chf_clusters_3$centers
```

```
##   calories  protein
## 1 850.0000 63.33333
## 2 435.7143 32.78571
## 3 173.0000 20.70000
```

```
cluster_list <- split(chf_nutrition$item, chf_clusters_3$cluster)
cluster_list
```

```
## $‘1‘
## [1] "30 piece Chicken Nuggets"    "Chicken Enchiladas Meal Kit"
## [3] "Chicken Parmesan Meal Kit"
```

```
##
## $‘2‘
##  [1] "Chargrilled Chicken Club Sandwich"
##  [2] "Chargrilled Chicken Sandwich"
##  [3] "3 Piece Chick-n-Strips"
##  [4] "4 piece Chick-n-Strips"
##  [5] "Chicken Deluxe"
##  [6] "12 piece Chicken Nuggets"
##  [7] "Chicken Salad Sandwich"
##  [8] "Chicken Sandwich"
##  [9] "Spicy Grilled Chicken Sub Sandwich"
## [10] "Regular Grilled Chicken Sub Sandwich"
## [11] "Smokehouse BBQ Bacon Sandwich"
## [12] "Spicy Chicken Sandwich"
## [13] "Spicy Deluxe"
## [14] "Chargrilled Chicken Cool Wrap"
##
## $‘3‘
##  [1] "Chick-n-Slider"                   "1 Piece Chick-n-Strips"
##  [3] "2 Piece Chick-n-Strips"           "4 piece Chicken Nuggets"
##  [5] "6 piece Chicken Nuggets"          "8 piece Chicken Nuggets"
##  [7] "4 Piece Grilled Chicken Nuggets"  "6 Piece Grilled Chicken Nuggets"
##  [9] "8 piece Grilled Chicken Nuggets"  "12 Piece Grilled Chicken Nuggets"
```