

Subway & Taco Bell

Loading our libraries & Reading in our Data

Libraries that are necessary in creating our k-means clusters.

Loading in our data.

```
# Load nutrition info
nutrition_full <- read_csv("data/nutritioninfo.csv", show_col_types = FALSE)
```

SUBWAYS

Determining Number of Cluster for Subways

First, we want to create a separate data set for the nutrition values of Subways items. Then, we want to create a subset of data containing *no characters*, for our clustering algorithm.

```
# Subway info
subway <- nutrition_full %>%
  filter(restaurant == "Subway")

# Remove columns containing characters
subway_nochar <- subway %>%
  select(-c(item,restaurant))
```

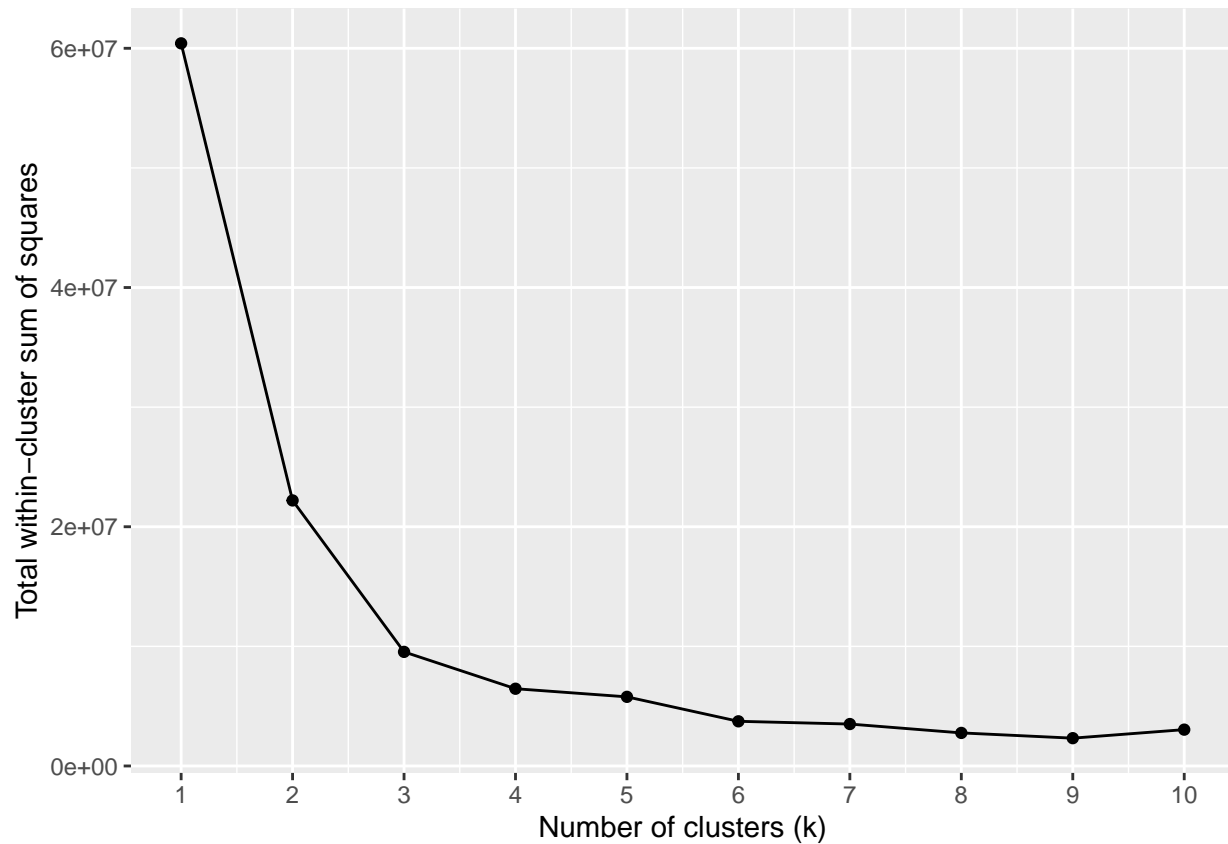
Next, let's make our elbow plot!

```
# Set our seed
set.seed(38)

# Iterate through clustering algorithm for 10 different values of k
subway_elbow <- tibble(k = 1:10) %>%
  mutate(
    # List-column of 10 kmeans objects
    # (apply `kmeans()` to each value of `k`)
    kmeans_results = purrr::map(k, ~kmeans(subway_nochar, .x)),
    glanced = purrr::map(kmeans_results, glance()) %>%
    # Turn `glanced` list-column into regular tibble columns
    unnest(cols = c(glanced))

# Construct elbow plot
ggplot(subway_elbow, 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")
```



```
# Regular k-means, not scaled.
subway_kmeans <- subway_nochar %>%
kmeans(centers = 3, nstart = 20)
```

Looking at All Variables

Now we can compare all the variables (once scaled)!

```
# Scaling our variables
subway_scaled <- subway_nochar %>%
# Standardize all numeric variables (subtract mean and divide by SD)
mutate(across(where(is.numeric), ~scale(.)[,1], .names = "{.col}_scaled")) %>%
select(calories_scaled, total_fat_scaled, protein_scaled, cholesterol_scaled,
       sodium_scaled, total_carb_scaled, sugar_scaled) %>%
drop_na()

# Create k-means
subway_kmeans_scaled <- subway_scaled %>%
kmeans(centers = 3, nstart = 20)

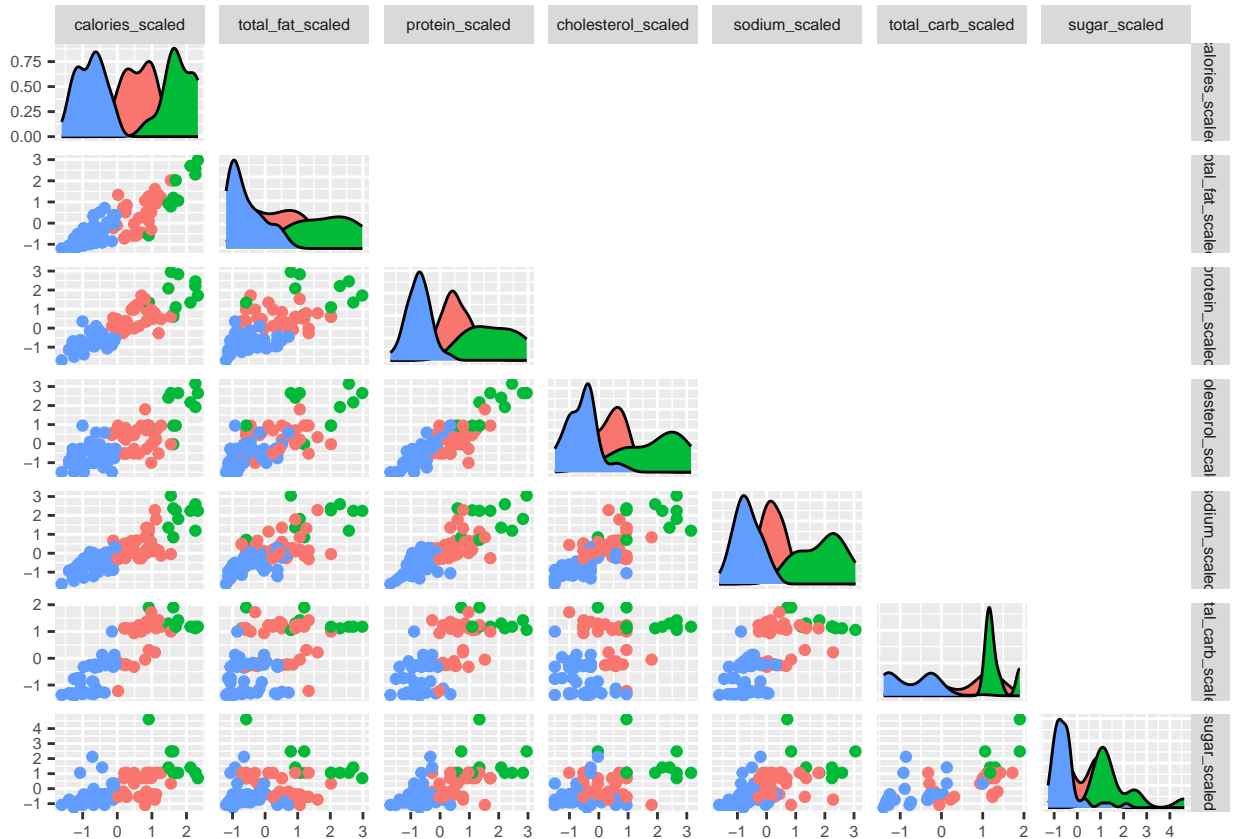
# Scale our clusters
```

```

subway_c_scaled <- augment(subway_kmeans_scaled, subway_scaled) %>%
  rename(cluster_scaled = .cluster)

# Make some visualizations!
ggpairs(subway_c_scaled, aes(color = cluster_scaled),
  columns = c("calories_scaled", "total_fat_scaled", "protein_scaled", "cholesterol_scaled",
    "sodium_scaled", "total_carb_scaled", "sugar_scaled"),
  upper = list(continuous = "blank")) +
  theme(text = element_text(size = 8))

```



TACO BELL

Determining Number of Cluster for Taco Bell

First, we want to create a separate data set for the nutrition values of Taco Bell items. Then, we want to create a subset of data containing *no characters*, for our clustering algorithm.

```

# Taco Bell info
tacobell <- nutrition_full %>%
  filter(restaurant == "Taco Bell")

# Remove columns containing characters
tacobell_nochar <- tacobell %>%
  select(-c(item, restaurant))

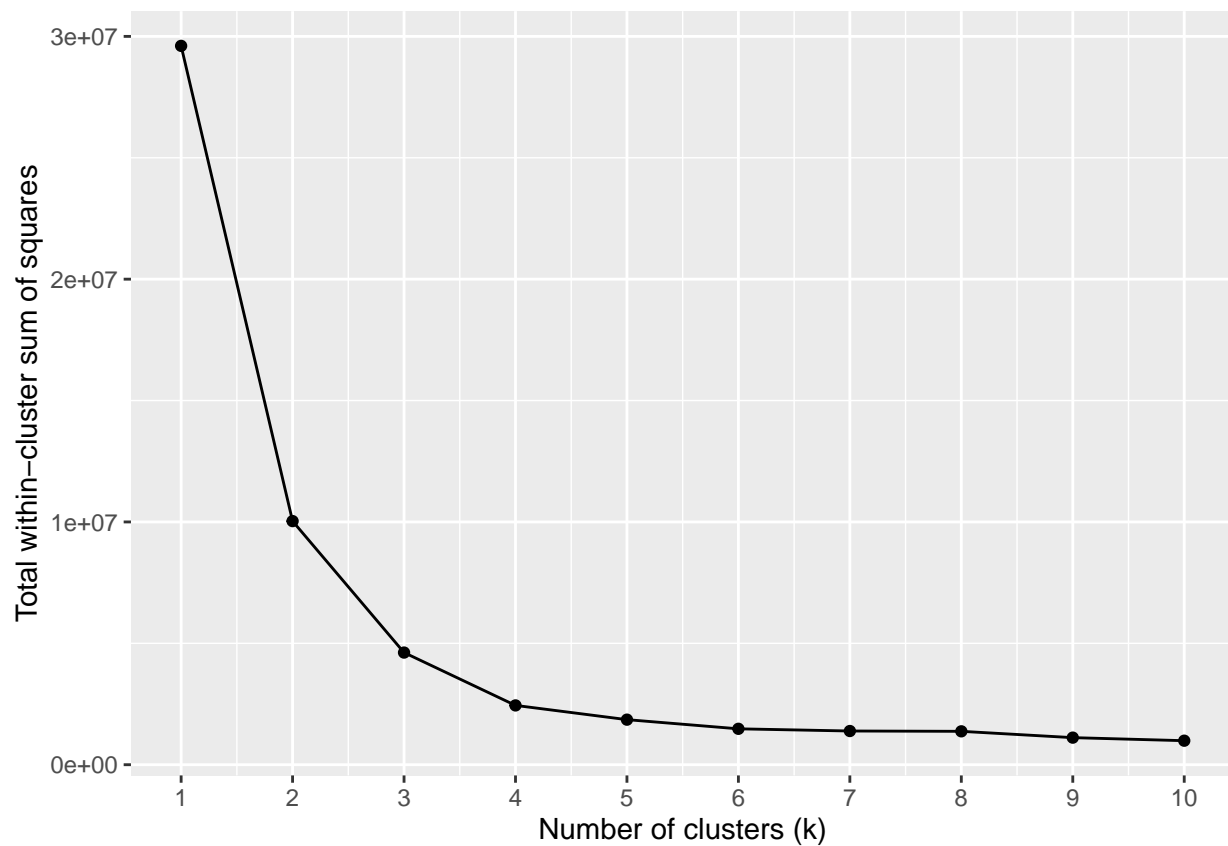
```

Next, let's make our elbow plot!

```
# Set our seed
set.seed(38)

# Iterate through clustering algorithm for 10 different values of k
tacobell_elbow <- tibble(k = 1:10) %>%
  mutate(
    # List-column of 10 kmeans objects
    # (apply `kmeans()` to each value of `k`)
    kmeans_results = purrr::map(k, ~kmeans(tacobell_nochar, .x)),
    glanced = purrr::map(kmeans_results, glance()) %>%
    # Turn `glanced` list-column into regular tibble columns
    unnest(cols = c(glanced))

# Construct elbow plot
ggplot(tacobell_elbow, 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")
```



```
# Regular k-means, not scaled.
tacobell_kmeans <- tacobell_nochar %>%
  kmeans(centers = 3, nstart = 20)
```

Looking at All Variables

Now we can compare all the variables (once scaled)!

```
# Scaling our variables
tacobell_scaled <- tacobell_nochar %>%
# Standardize all numeric variables (subtract mean and divide by SD)
mutate(across(where(is.numeric), ~scale(.)[,1], .names = "{.col}_scaled")) %>%
select(calories_scaled, total_fat_scaled, protein_scaled, cholesterol_scaled,
       sodium_scaled, total_carb_scaled, sugar_scaled) %>%
drop_na()

# Create k-means
tacobell_kmeans_scaled <- tacobell_scaled %>%
kmeans(centers = 3, nstart = 20)

# Scale our clusters
tacobell_c_scaled <- augment(tacobell_kmeans_scaled, tacobell_scaled) %>%
rename(cluster_scaled = .cluster)

# Make some visualizations!
ggpairs(tacobell_c_scaled, aes(color = cluster_scaled),
columns = c("calories_scaled", "total_fat_scaled", "protein_scaled", "cholesterol_scaled",
            "sodium_scaled", "total_carb_scaled", "sugar_scaled"),
upper = list(continuous = "blank")) +
theme(text = element_text(size = 8))
```

