Lab 7a - K-Means Clustering - Example Solution

Lab Purpose

This lab is designed to walk you through several *kmeans* clustering examples on a data set. Each row of the data set is a college or university, and the variables are characteristics of an incoming class of first year students.

The variables we have are:

- `STABBR`: State
- `CITY`: City
- `INSTNM`: Institution
- `SATMT25`: 25th percentile SAT MATH score
- `SATVR25`: 25th percentile SAT Verbal score
- `ADM_RATE`: Admission rate
- `SAT_AVG`: Average SAT score
- `GRAD_DEBT_MDN`: Median debt at graduation ($)
- `PCIP27`: % of graduates majoring in Mathematics & Statistics
- `COSTT4_A`: Average cost of attendance

To simplify our exploration of this data, we will only look at a random sample of schools in Massachusetts, and we will start with a focus on SAT scores. The data is slightly out-dated (Amherst, for example, no longer requires SAT/ACT scores), but is still informative for this exercise.

1 - k-means clustering

part a - Run the code below to read in and wrangle the data. Make sure you understand what each line is doing.

```
colleges <- read_csv("data/colleges_subset.csv") %>%
  # Missing scores reported as text "NULL"; make numeric
  mutate(SAT_math25 = as.numeric(SATMT25),
         SAT_verbal25 = as.numeric(SATVR25)) %>%
  rename(state = STABBR,
         city = CITY,
         institution = INSTNM)

dim(colleges)
```

```
[1] 7175   12
```

```
head(colleges)
```

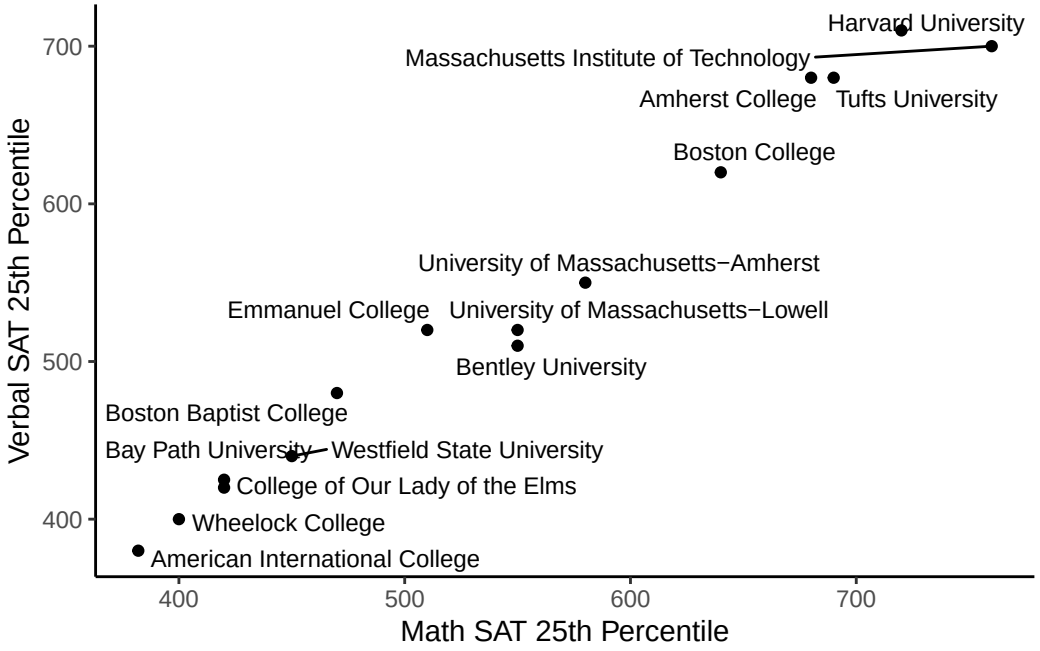
```
# A tibble: 6 x 12
  state city   institution SATMT25 SATVR25 ADM_RATE SAT_AVG GRAD_DEBT_MDN PCIP27
  <chr> <chr>  <chr>       <chr>   <chr>   <chr>    <chr>   <chr>         <chr>
1 AL    Normal Alabama A ~ 370     380     0.8738   849     32750         0.0024
2 AL    Birmi~ University~ 490     480     0.5814   1125    21833         0.009
3 AL    Montg~ Amridge Un~ NULL    NULL    NULL     NULL    22890         0
4 AL    Hunts~ University~ 540     520     0.7628   1257    22647         0.0132
5 AL    Montg~ Alabama St~ 360     370     0.459    825     31500         0.0146
6 AL    Tusca~ The Univer~ 490     490     0.5259   1202    23290         0.009
# i 3 more variables: COSTT4_A <chr>, SAT_math25 <dbl>, SAT_verbal25 <dbl>
```

```
set.seed(231)
ma_sample <- colleges %>%
  # Only keep schools in MA
  filter(state %in% c("MA")) %>%
  # Only keep schools with non-missing SAT scores
  drop_na(SAT_math25, SAT_verbal25) %>%
  select(state, city, institution, SAT_math25, SAT_verbal25) %>%
  # Select a random sample of 15 schools
  sample_n(15)
```

part b - First, let's look at a scatterplot of Math SAT (25th percentile) vs Verbal SAT (25th percentile). Which schools are most similar in terms of these two variables? Do we need to standardize the variables in this case? Why or why not?

Solution:

```
ggplot(data = ma_sample, aes(x = SAT_math25, y = SAT_verbal25)) +
  geom_point() +
  geom_text_repel(aes(label = institution), size = 3) +
  labs(x = "Math SAT 25th Percentile",
       y = "Verbal SAT 25th Percentile")
```



The seed set above will determine which 15 schools you get. A seed of 231 yields a sample that includes Amherst College, so I choose to keep it. Answers may vary slightly based on that.

Based on this plot, there is a strong positive linear association between the verbal and math SAT 25th percentile scores. There appear to be roughly 3 groups of schools. Amherst is with other elite institutions in the upper-right corner. There is a small middle group including both UMass schools in our sample, Bentley, and Emmanuel College. The remaining schools form a group in the lower left.

The two schools most similar in terms of these variables are Bay Path University and College of our Lady of the Elms - their data points are closest to each other out of all the points.

We do not need to standardize the variables here because they are both on the same scale. We would need to standardize if we included any other predictors though.

Note: We usually need to standardize. In fact, it honestly doesn't hurt to standardize, but we'll proceed without for this example and adjust below in the next when we move beyond using just 2 variables to do the clustering. Remember to check and not just jump into clustering when you go to apply k-means to variables in your data set.

> part c - Let's use k-means to find two clusters. Run the code below to find the clustering solution. You can set the seed to whatever you would like.

```
set.seed(231)
clustering_vars <- c("SAT_math25", "SAT_verbal25")
ma_km2 <- ma_sample %>%
  select(clustering_vars) %>%
  kmeans(centers = 2, nstart = 20)

# Vector of cluster assignments
ma_km2$cluster
```

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

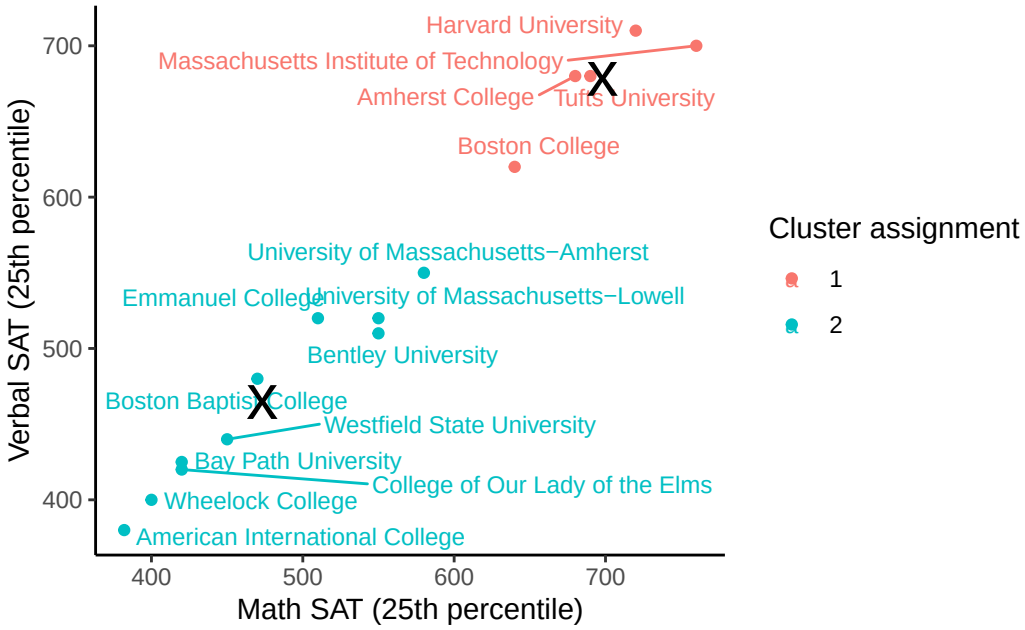
```
# The centroids for the fit
ma_km2$centers
```

```
  SAT_math25 SAT_verbal25
1      698.0        678.0
2      473.2        464.5
```

```
# Add cluster assignment to the data frame
ma_sample <- ma_sample %>%
  mutate(clusters2 = factor(ma_km2$cluster))

# Visualize the cluster assignments and centroids
ggplot(data = ma_sample, aes(x = SAT_math25, y = SAT_verbal25)) +
  geom_point(aes(color = clusters2)) +
  geom_text_repel(aes(label = institution, color = clusters2), size = 3) +
  coord_fixed() +
  geom_point(data = data.frame(ma_km2$centers),
             aes(x = SAT_math25, y = SAT_verbal25),
             pch = "x", size = 8) +
```

```
  labs(x = "Math SAT (25th percentile)",
       y = "Verbal SAT (25th percentile)",
       color = "Cluster assignment")
```



part d - What if we used 5 clusters? Repeat the k-means clustering analysis and visualization above, but with 5 clusters instead of 2.

Important: We want to be able to COMPARE the clustering solutions. Thus, you can't just overwrite the objects and change the centers = 2 to 5 - we'd lose the first solution. Create NEW objects, but update the name. For example, instead of ma_km2, make ma_km5.

Solution:

```
set.seed(231)
ma_km5 <- ma_sample %>%
  select(clustering_vars) %>%
  kmeans(centers = 5, nstart = 20)

# Vector of cluster assignments
ma_km5$cluster
```

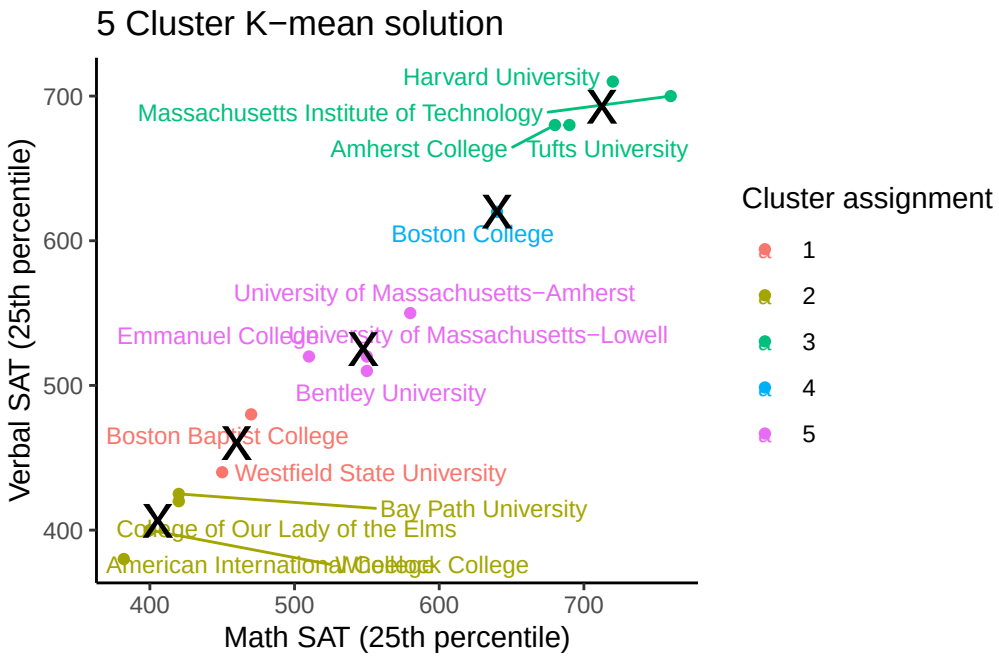
```
 [1] 5 3 2 1 2 5 3 3 4 1 2 2 5 3 5
```

```r
# The centroids for the fit
ma_km5$centers
```

```
  SAT_math25 SAT_verbal25
1      460.0       460.00
2      405.5       406.25
3      712.5       692.50
4      640.0       620.00
5      547.5       525.00
```

```r
# Add cluster assignment to the data frame
ma_sample <- ma_sample %>%
  mutate(clusters5 = factor(ma_km5$cluster))

# Visualize the cluster assignments and centroids
ggplot(data = ma_sample, aes(x = SAT_math25, y = SAT_verbal25)) +
  geom_point(aes(color = clusters5)) +
  geom_text_repel(aes(label = institution, color = clusters5), size = 3) +
  coord_fixed() +
  geom_point(data = data.frame(ma_km5$centers),
             aes(x = SAT_math25, y = SAT_verbal25),
             pch = "x", size = 8) +
  labs(x = "Math SAT (25th percentile)",
       y = "Verbal SAT (25th percentile)",
       color = "Cluster assignment",
       title = "5 Cluster K-mean solution")
```



part e - Which solution will have *smaller* total within-cluster variation? Why? Check your answer using the code below.

Solution:

```
ma_km2$tot.withinss
```

```
[1] 88624.1
```

```
ma_km5$tot.withinss #name depends on what you called the second solution
```

```
[1] 11196.75
```

The solution with k=5 will have the lowest WGSS (within group sum of squares). Increasing k tends to decrease WGSS, as it is easier for each observation to be closer to its cluster centroid when there are more clusters.

part f - We can use an *elbow plot* to try to identify an optimal number of clusters k. Run the code below to create the elbow plot and identify an optimal number of clusters based on where the "elbow" bends (the point where the total within-cluster sum of squares starts to decrease linearly). What k would you pick?

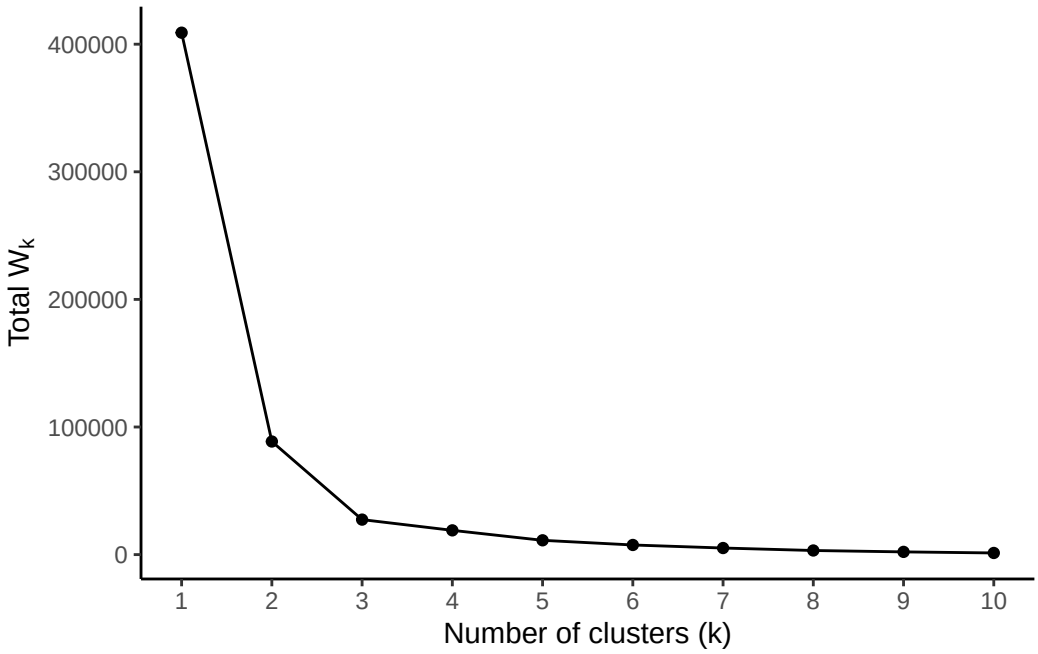
Solution:

```
elbow_plot <- data.frame(clusters = 1:10,
                         within_ss = rep(NA, 10))

set.seed(75)
for (i in 1:10){
  ma_kmi_out <- ma_sample %>%
    select(clustering_vars) %>%
    kmeans(centers = i, nstart = 20)

  elbow_plot$within_ss[i] <- ma_kmi_out$tot.withinss
}

# Construct elbow plot
ggplot(elbow_plot, aes(x = clusters, y = within_ss)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = 1:10) +
  labs(x = "Number of clusters (k)", y = expression("Total W"[k]))
```



With my sample (which could cause your plot to be very different!), it looks like a k of 3 would

be reasonable. $k = 2$ is a bit too few - we lower WGSS a good bit moving to 3. After k=3 though, the decrease in WGSS is tiny compared to adding a cluster. So, I'll go with k=3 as looking the best for my solution here.

2 - Standardization

Let's try clustering colleges by more than two variables. This time, we'll consider admission rate, SAT average, median debt at graduation, percentage of graduates majoring in Mathematics & Statistics, and average cost of attendance. This will be harder to visualize, but since we only have 15 observations, we can list out which colleges end up in each cluster.

part a - Run the code below to get the data ready for analysis. Make sure you understand what each line is doing.

```
# Select the additional variables of interest
ma_sample2 <- colleges %>%
  select(institution,
         state,
         city,
         admit_rate = ADM_RATE,
         SAT_avg = SAT_AVG,
         grad_debt_median = GRAD_DEBT_MDN,
         pct_mathstat = PCIP27,
         cost_avg = COSTT4_A) %>%
  # Use right_join to only keep the colleges in our original sample
  right_join(ma_sample %>% select(institution, state, city)) %>%
  # Additional variables are character but should be numeric
  mutate(across(admit_rate:cost_avg, ~ as.numeric(.)),
         # Standardize or scale() numeric variables (subtract mean and divide by SD)
         across(where(is.numeric),  ~ scale(.)[,1], .names = "{.col}_scaled")) %>%
  # Drop rows with missing values (kmeans breaks with missing values)
  drop_na()

glimpse(ma_sample2)
```

```
Rows: 14
Columns: 13
$ institution            <chr> "American International College", "Amherst Col~
$ state                  <chr> "MA", "MA", "MA", "MA", "MA", "MA", "MA", "MA"~
$ city                   <chr> "Springfield", "Amherst", "Longmeadow", "Walth~
$ admit_rate             <dbl> 0.6931, 0.1369, 0.5992, 0.4632, 0.3114, 0.7140~
$ SAT_avg                <dbl> 881, 1451, 967, 1198, 1378, 1122, 1506, 1172, ~
$ grad_debt_median       <dbl> 27000.0, 13000.0, 24023.0, 26000.0, 19000.0, 2~
$ pct_mathstat           <dbl> 0.0000, 0.0764, 0.0000, 0.0142, 0.0215, 0.0045~
$ cost_avg               <dbl> 45655, 66572, 44015, 60895, 65595, 53199, 6440~
$ admit_rate_scaled      <dbl> 0.6048265, -1.2721514, 0.2879473, -0.1710045, ~
```

```
$ SAT_avg_scaled          <dbl> -1.23943991, 1.16519164, -0.87663585, 0.097872~
$ grad_debt_median_scaled <dbl> 0.9078170, -1.3603658, 0.4255042, 0.7458040, -~
$ pct_mathstat_scaled     <dbl> -0.8394215, 1.5769991, -0.8394215, -0.3902962,~
$ cost_avg_scaled         <dbl> -0.16678983, 1.11448587, -0.26724841, 0.766739~
```

The comments should help explain what is going on. The select is doing both a select and rename operation at the same time. Additionally, with my seed, we lose one college due to lack of information (one variable value is missing).

> part b - Now implement k-means clustering with $k = 3$ clusters using the standardized variables. What schools are clustered together when using the standardized variables? Are the clusters different if the unstandardized variables are used?

Solution:

```r
# Clustering using standardized variables
set.seed(231)
ma2_km3_out <- ma_sample2 %>%
  select(ends_with("scaled")) %>%
  kmeans(centers = 3, nstart = 20)

# Clustering using unstandardized variables (for comparison)
set.seed(231)
ma2_km3_out_unscaled <- ma_sample2 %>%
  select(admit_rate:cost_avg) %>%
  kmeans(centers = 3, nstart = 20)

# Add cluster assignments to the data frame
ma_sample2 <- ma_sample2 %>%
  mutate(clusters3_scaled = factor(ma2_km3_out$cluster),
         clusters3_unscaled = as.character(ma2_km3_out_unscaled$cluster))

# Clusters based on standardized vars
ma_sample2 %>%
  select(institution, clusters3_scaled) %>%
  arrange(clusters3_scaled)
```

```
# A tibble: 14 x 2
   institution                    clusters3_scaled
   <chr>                          <fct>
 1 Amherst College                1
 2 Harvard University             1
```

```
 3 Massachusetts Institute of Technology 1
 4 American International College        2
 5 Bay Path University                   2
 6 Emmanuel College                      2
 7 University of Massachusetts-Lowell    2
 8 University of Massachusetts-Amherst   2
 9 College of Our Lady of the Elms       2
10 Westfield State University            2
11 Wheelock College                      2
12 Bentley University                    3
13 Boston College                        3
14 Tufts University                      3
```

```
# Clusters based on unstandardized vars
ma_sample2 %>%
  select(institution, clusters3_unscaled) %>%
  arrange(clusters3_unscaled)
```

```
# A tibble: 14 x 2
   institution                          clusters3_unscaled
   <chr>                                <chr>
 1 Amherst College                      1
 2 Bentley University                   1
 3 Boston College                       1
 4 Harvard University                   1
 5 Massachusetts Institute of Technology 1
 6 Tufts University                     1
 7 University of Massachusetts-Lowell   2
 8 University of Massachusetts-Amherst  2
 9 Westfield State University           2
10 American International College       3
11 Bay Path University                  3
12 Emmanuel College                     3
13 College of Our Lady of the Elms      3
14 Wheelock College                     3
```

The code here is all provided to run the analyses.

To compare the solutions, I added a third table and a tally table. The unscaled clusters found nearly match what I described based on the original scatterplot (which used different variables even). We can see the solutions differ somewhat, as what was cluster 1 and 3 in the scaled

solution is cluster 1 in the unscaled, while the cluster 2 and 3 in the unscaled were combined to form cluster 2 in the scaled solution.

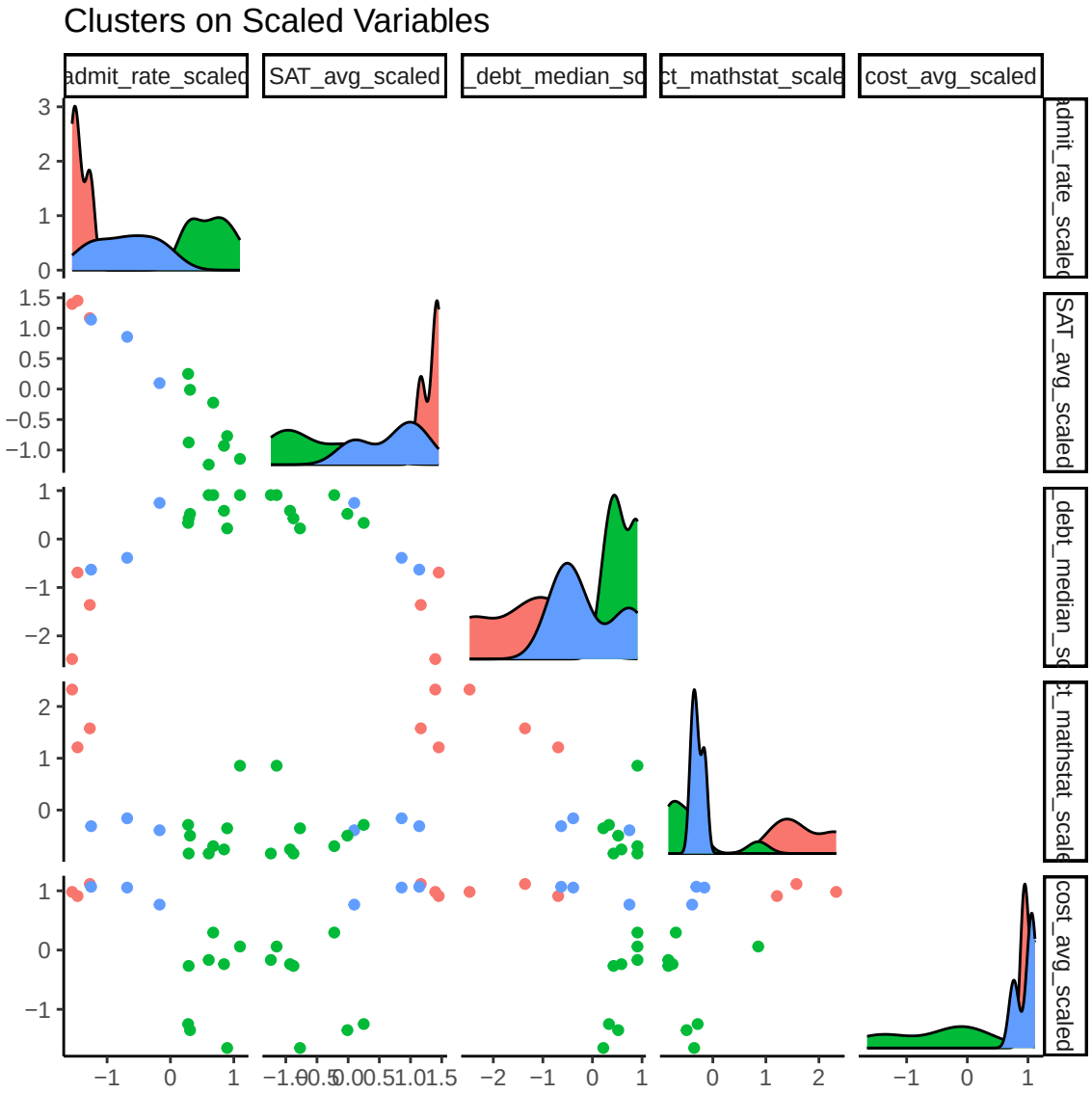
Listing the schools in each cluster is an appropriate way to summarize them, but this will differ based on your initial seed to set your sample.

3 - Extracting meaning

Clustering methods don't just determine clusters based on individual variable values, but rather how these p variables relate in p-dimensional space (e.g. clusters in $\mathbb{R}^5$ for this example). We are limited for now to 2D visualizations, however. We can examine a *scatterplot matrix* using the ggplot "add on" package, **GGally**. Run the code below to see the `ggpairs()` function in action. Consider both the density plots and the scatterplots. What are the defining characteristics of each cluster?

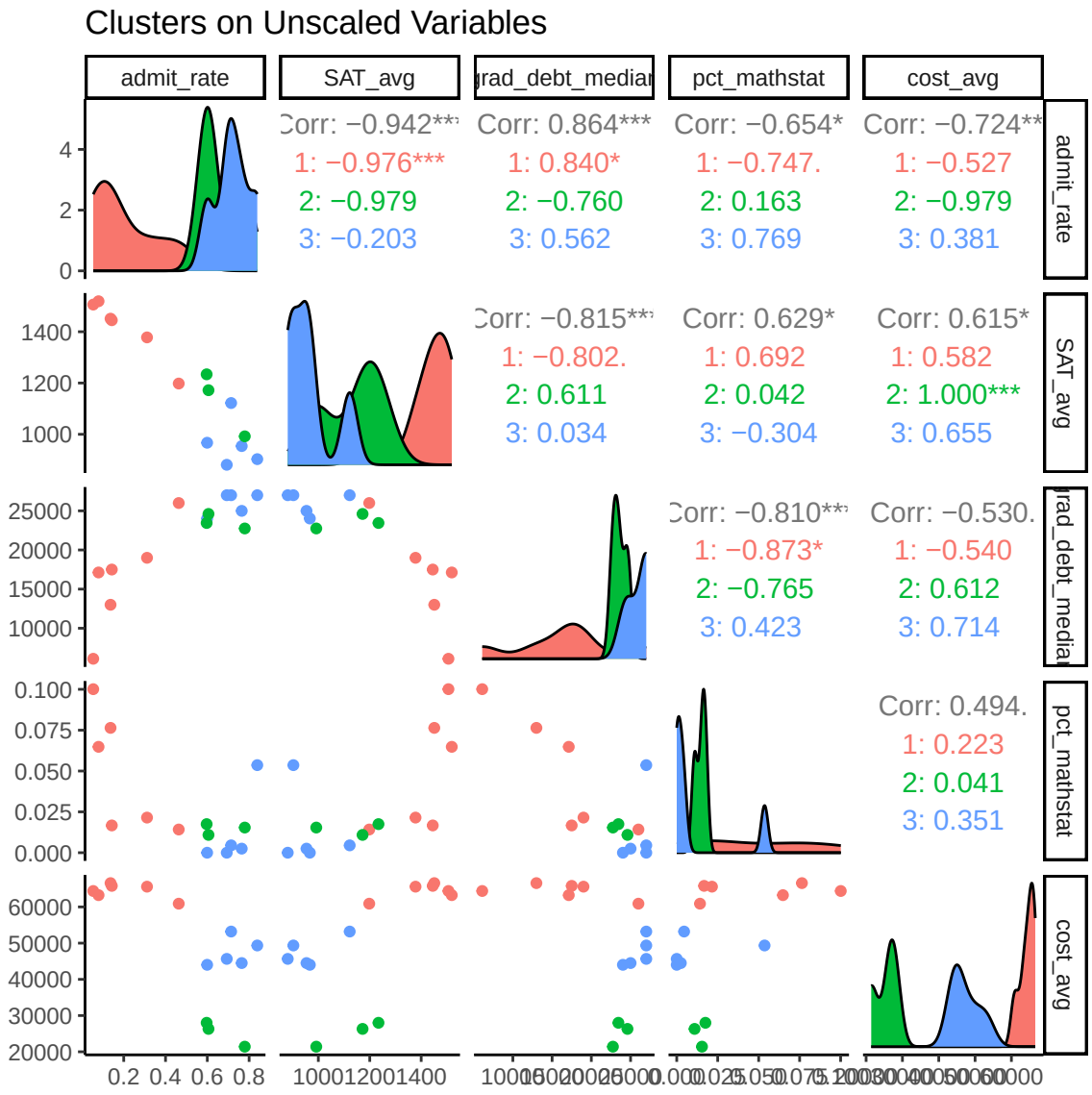
Solution:

```
# Scatterplot matrix of clusters based on scaled variables
GGally::ggpairs(data = ma_sample2,
                aes(color = clusters3_scaled),
                columns = c("admit_rate_scaled",
                            "SAT_avg_scaled",
                            "grad_debt_median_scaled",
                            "pct_mathstat_scaled",
                            "cost_avg_scaled"),
                upper = list(continuous = "blank")) +
  labs(title = "Clusters on Scaled Variables")
```




```
# Scatterplot matrix of clusters based on unscaled variables
GGally::ggpairs(data = ma_sample2,
                aes(color = clusters3_unscaled),
                columns = c("admit_rate",
                            "SAT_avg",
                            "grad_debt_median",
                            "pct_mathstat",
                            "cost_avg")) +
```

```
labs(title = "Clusters on Unscaled Variables")
```



Answers will differ again based on your sample selected.

For my 3 cluster scaled variable solution, the first cluster (colored red), which includes Amherst, has low admittance rates, lower debt, and higher scores and cost than the other clusters. Cluster 2 (green) is the opposite. Cluster 3 (blue) is right in between those.

For the unscaled variables, cluster 1 (which includes Amherst, and is red) is again characterized

by high scores and cost, with lower admittance rates and debt. Cluster 3 (blue) has some bimodal distributions and is harder to describe, but seems to have the highest admittance rate and debt values, with moderate cost. Cluster 2 (green) tends to be in between, except that it has the lowest cost average.

4 - Your Turn! Customer segmentation

Suppose you're working as a data scientist for a credit card company. The company wants to divide their customers into groups for targeted marketing. That is, your tasked with grouping the credit card holders based on their credit card behavior; then, the marketing team at the company will use the information you provide them to help inform their marketing stategy. You're given a dataset with information on 8,950 credit card holders, with the following variables:

- `CUST_ID`: Identification of Credit Card holder
- `BALANCE`: Balance amount left in their account to make purchases
- `BALANCE_FREQUENCY`: How frequently the Balance is updated, score between 0 and 1 (1 = frequently updated, 0 = not frequently updated)
- `PURCHASES`: Amount of purchases made from account
- `PURCHASES_FREQUENCY`: How frequently the Purchases are being made, score between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased)
- `ONEOFF_PURCHASES`: Maximum purchase amount done in one-go
- `ONEOFFPURCHASESFREQUENCY`: How frequently purchases are happening in one-go (1 = frequently purchased, 0 = not frequently purchased)
- `PRC_FULL_PAYMENT`: Percent of full payment paid by user
- `INSTALLMENTS_PURCHASES`: Amount of purchase done in installment
- `PURCHASES_INSTALLMENTS_FREQUENCY`: How frequently purchases in installments are being done (1 = frequently done, 0 = not frequently done)
- `CASH_ADVANCE`: Cash in advance given by the user
- `CASH_ADVANCE_FREQUENCY`: How frequently the cash in advance being paid
- `CASH_ADVANCE_TRX`: Number of Transactions made with "Cash in Advanced"
- `PURCHASES_TRX`: Number of purchase transactions made
- `CREDIT_LIMIT`: Limit of Credit Card for user
- `PAYMENTS`: Amount of Payment done by user
- `MINIMUM_PAYMENTS`: Minimum amount of payments made by user
- `PRC_FULL_PAYMENT`: Percent of full payment paid by user
- `TENURE`: Tenure of credit card service for user

```
# loads in the data
ccdata <- read_csv("data/cc-general.csv")
```

Answers to the below will vary based on choices made. This is an example solution.

> part a - Apply k-means clustering to identify 3 clusters. Don't forget to remove any rows with missing values and to standardize the variables first. How many customers are in each cluster?

Solution:

k=3 has been chosen for us, so luckily, we don't have that issue. We see there are 18 variables, but at least one, the first one, cust_id is an identifier, so it shouldn't be used in the clustering. We keep it in the data set, in case someone wants to use it to label observations in the cluster.

We also see that there are some NAs (use summary(ccdata)), so we remove those. Note: If you decide to select a subset of variables, do that FIRST, then drop the NAs. We also see that the variables are on VERY different scales, so we must standardize to use them.

```
summary(ccdata)
```

```
  CUST_ID             BALANCE        BALANCE_FREQUENCY   PURCHASES
Length:8950        Min.   :    0.0   Min.   :0.0000    Min.   :    0.00
Class :character   1st Qu.:  128.3   1st Qu.:0.8889    1st Qu.:   39.63
Mode  :character   Median :  873.4   Median :1.0000    Median :  361.28
                   Mean   : 1564.5   Mean   :0.8773    Mean   : 1003.20
                   3rd Qu.: 2054.1   3rd Qu.:1.0000    3rd Qu.: 1110.13
                   Max.   :19043.1   Max.   :1.0000    Max.   :49039.57

ONEOFF_PURCHASES  INSTALLMENTS_PURCHASES  CASH_ADVANCE     PURCHASES_FREQUENCY
Min.   :    0.0   Min.   :    0.0        Min.   :    0.0   Min.   :0.00000
1st Qu.:    0.0   1st Qu.:    0.0        1st Qu.:    0.0   1st Qu.:0.08333
Median :   38.0   Median :   89.0        Median :    0.0   Median :0.50000
Mean   :  592.4   Mean   :  411.1        Mean   :  978.9   Mean   :0.49035
3rd Qu.:  577.4   3rd Qu.:  468.6        3rd Qu.: 1113.8   3rd Qu.:0.91667
Max.   :40761.2   Max.   :22500.0        Max.   :47137.2   Max.   :1.00000

ONEOFF_PURCHASES_FREQUENCY PURCHASES_INSTALLMENTS_FREQUENCY
Min.   :0.00000            Min.   :0.0000
1st Qu.:0.00000            1st Qu.:0.0000
Median :0.08333            Median :0.1667
Mean   :0.20246            Mean   :0.3644
3rd Qu.:0.30000            3rd Qu.:0.7500
Max.   :1.00000            Max.   :1.0000

CASH_ADVANCE_FREQUENCY CASH_ADVANCE_TRX  PURCHASES_TRX     CREDIT_LIMIT
Min.   :0.0000         Min.   :  0.000   Min.   :  0.00   Min.   :   50
1st Qu.:0.0000         1st Qu.:  0.000   1st Qu.:  1.00   1st Qu.: 1600
Median :0.0000         Median :  0.000   Median :  7.00   Median : 3000
Mean   :0.1351         Mean   :  3.249   Mean   : 14.71   Mean   : 4494
3rd Qu.:0.2222         3rd Qu.:  4.000   3rd Qu.: 17.00   3rd Qu.: 6500
Max.   :1.5000         Max.   :123.000   Max.   :358.00   Max.   :30000
                                                          NA's   :1
```

```
    PAYMENTS       MINIMUM_PAYMENTS   PRC_FULL_PAYMENT     TENURE
 Min.   :    0.0   Min.   :    0.02   Min.   :0.0000   Min.   : 6.00
 1st Qu.:  383.3   1st Qu.:  169.12   1st Qu.:0.0000   1st Qu.:12.00
 Median :  856.9   Median :  312.34   Median :0.0000   Median :12.00
 Mean   : 1733.1   Mean   :  864.21   Mean   :0.1537   Mean   :11.52
 3rd Qu.: 1901.1   3rd Qu.:  825.49   3rd Qu.:0.1429   3rd Qu.:12.00
 Max.   :50721.5   Max.   :76406.21   Max.   :1.0000   Max.   :12.00
                   NA's   :313
```

```
#set up data
mydata <- ccdata %>%
  drop_na() %>%
  mutate(across(where(is.numeric),  ~ scale(.)[,1], .names = "{.col}_scaled"))

#run clustering
set.seed(231)
km3_out <- mydata %>%
  select(ends_with("scaled")) %>%
  kmeans(centers = 3, nstart = 20)

km3_out$size
```

```
[1] 1211 5874 1551
```

With our seed, we see the clusters have sizes 1211, 5874, and 1551 respectively.

> part b - Compute the centroids for each cluster. Can you identify any distinguishing characteristics about the clusters from these centroid values?

Solution:

```
km3_out$centers
```

```
  BALANCE_scaled BALANCE_FREQUENCY_scaled PURCHASES_scaled
1      0.3033055                0.4198473        1.5115954
2     -0.3678418               -0.1690527       -0.2338323
3      1.1562862                0.3124311       -0.2946557
  ONEOFF_PURCHASES_scaled INSTALLMENTS_PURCHASES_scaled CASH_ADVANCE_scaled
1               1.2636438                     1.2511481          -0.2490798
2              -0.2047530                    -0.1764519          -0.3113077
3              -0.2111886                    -0.3086149           1.3734733
```

```
  PURCHASES_FREQUENCY_scaled ONEOFF_PURCHASES_FREQUENCY_scaled
1                 1.13331325                          1.5390745
2                -0.06113643                         -0.2339758
3                -0.65333782                         -0.3155675
  PURCHASES_INSTALLMENTS_FREQUENCY_scaled CASH_ADVANCE_FREQUENCY_scaled
1                              0.95729600                   -0.3644758
2                             -0.04953831                   -0.3341472
3                             -0.55983069                    1.5500713
  CASH_ADVANCE_TRX_scaled PURCHASES_TRX_scaled CREDIT_LIMIT_scaled
1              -0.2545865            1.6675907           0.8888954
2              -0.2999197           -0.2453532          -0.3426309
3               1.3346438           -0.3728224           0.6035860
  PAYMENTS_scaled MINIMUM_PAYMENTS_scaled PRC_FULL_PAYMENT_scaled TENURE_scaled
1       0.8243600               0.1579333              0.47227073     0.2953214
2      -0.2874439              -0.1365652              0.01458008    -0.0244149
3       0.4449682               0.3938922             -0.42396082    -0.1381180
```

Remember - these are the scaled variables. You could also look at centers for the unstandardized variables if that would help.

Cluster 3 individuals tend to have high balances and the most cash_advanced. They also have the highest minimum payments and lowest percent full payment.

Cluster 2 individuals have the lowest balances, lowest cash_advanced, the lowest credit limits and lowest payments scaled.

Cluster 1 individuals have modest balances, the highest purchases, the highest percent full payment, highest payments, and modest minimum payments. These customers have the longest tenure as well.

> part c - In 1-3 sentences, explain how you would expect the results to be different had you forgotten to standardize the variables prior to clustering.

Solution:

In the original data set, some variables were scaled to be a score between 0 and 1, while others were in the thousands. If we didn't scale, the variables with the scales in the thousands would dominate the solution, as the most a difference could be in terms of some of the scaled variables would be 1, compared to thousands. So, we would expect the clusters we found to be derived mostly from Balance, Purchases, OneOff_Purchases, Installments_Purchases, Cash_Advance, Credit_Limit, and Minimum_Payments, if variables were not standardized.

> part d - Identify some primary defining characteristics of each of the three clusters. Come up with a short name for each cluster based on their defining characteristics.

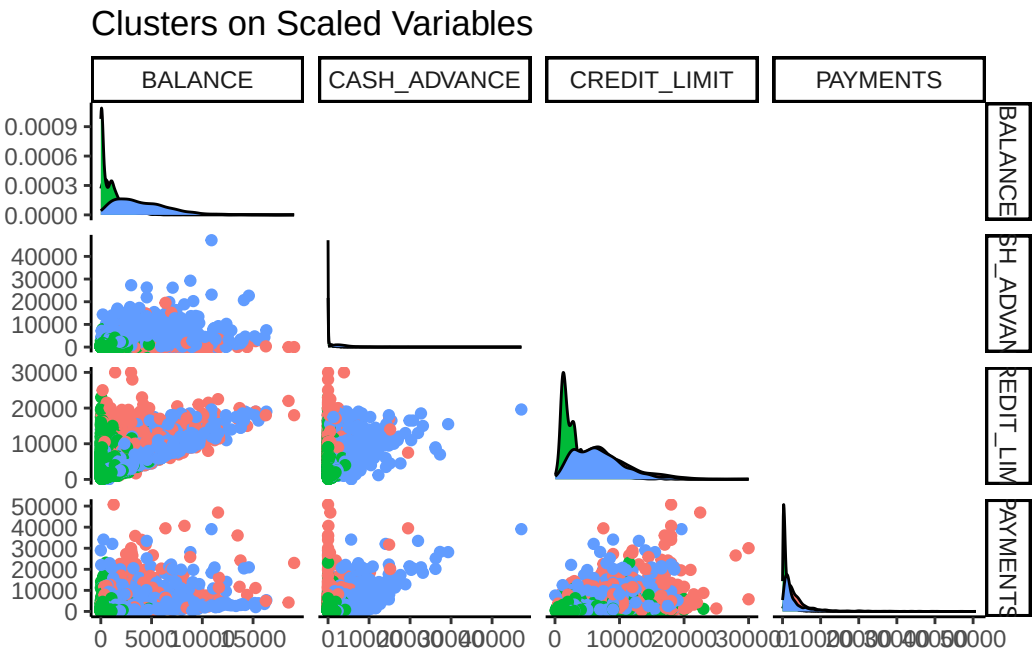
Note: There are so many variables that using `ggpairs()` to visualize them all at once produces a figure that isn't legible. You can try running `ggpairs()` separately for 3-4 variables at a time and/or considering the centroids of the clusters (as you computed above). This is a reason that we use dimension reduction techniques to help visualize our solutions (more in Chapter 12, if you want to look!).

Solution:

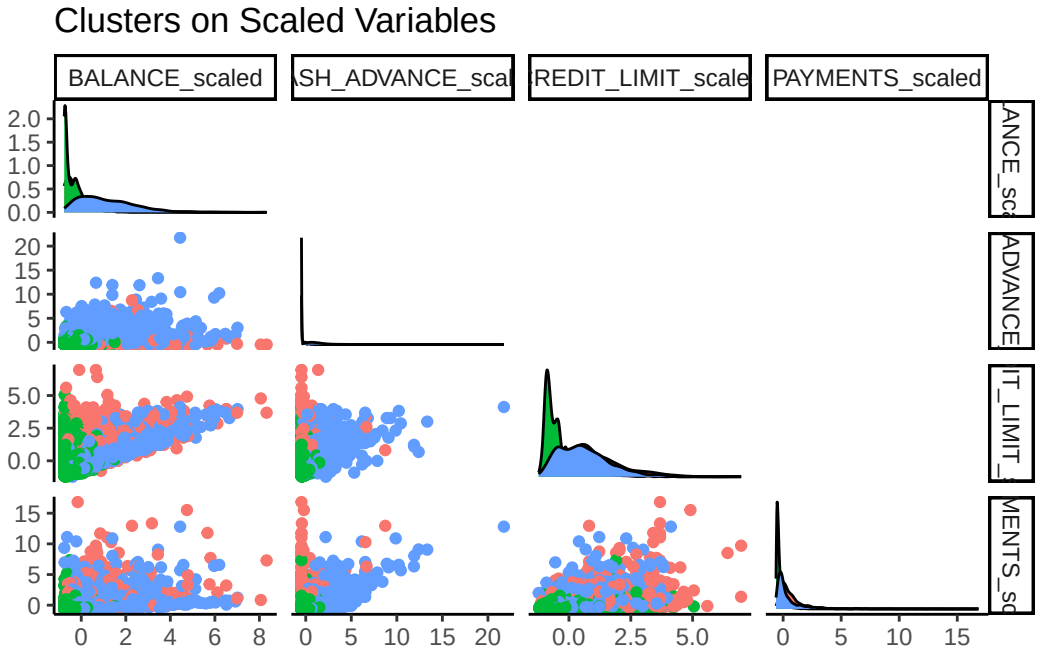
Answers will vary. We have decent descriptions up above - Cluster 1 is the longer term good customers. Cluster 3 is individuals who use cash_advance a lot. Cluster 2 is individuals with less credit available.

I'll run ggpairs on a subset of the variables to see if this helps with thinking about the clusters.

```
# Scatterplot matrix of clusters based on scaled variables
# Plotted on 4 chosen unscaled variables
GGally::ggpairs(data = select(mydata, BALANCE, CASH_ADVANCE, CREDIT_LIMIT, PAYMENTS),
                aes(color = factor(km3_out$cluster)),
                columns = c("BALANCE",
                            "CASH_ADVANCE",
                            "CREDIT_LIMIT",
                            "PAYMENTS"),
                upper = list(continuous = "blank")) +
  labs(title = "Clusters on Scaled Variables")
```



```
# Scatterplot matrix of clusters based on scaled variables
# Plotted on 4 chosen scaled variables
GGally::ggpairs(data = select(mydata, BALANCE_scaled, CASH_ADVANCE_scaled, CREDIT_LIMIT_sc
                aes(color = factor(km3_out$cluster)),
                columns = c("BALANCE_scaled",
                            "CASH_ADVANCE_scaled",
                            "CREDIT_LIMIT_scaled",
                            "PAYMENTS_scaled"),
                upper = list(continuous = "blank")) +
  labs(title = "Clusters on Scaled Variables")
```



part e - Create an elbow plot to help identify an appropriate number of clusters to create in this analysis. How many clusters seem reasonable such that there's enough of a decrease in the total within cluster variability to warrant that many clusters but not too many clusters to complicate the analysis. How might you proceed?

Solution:

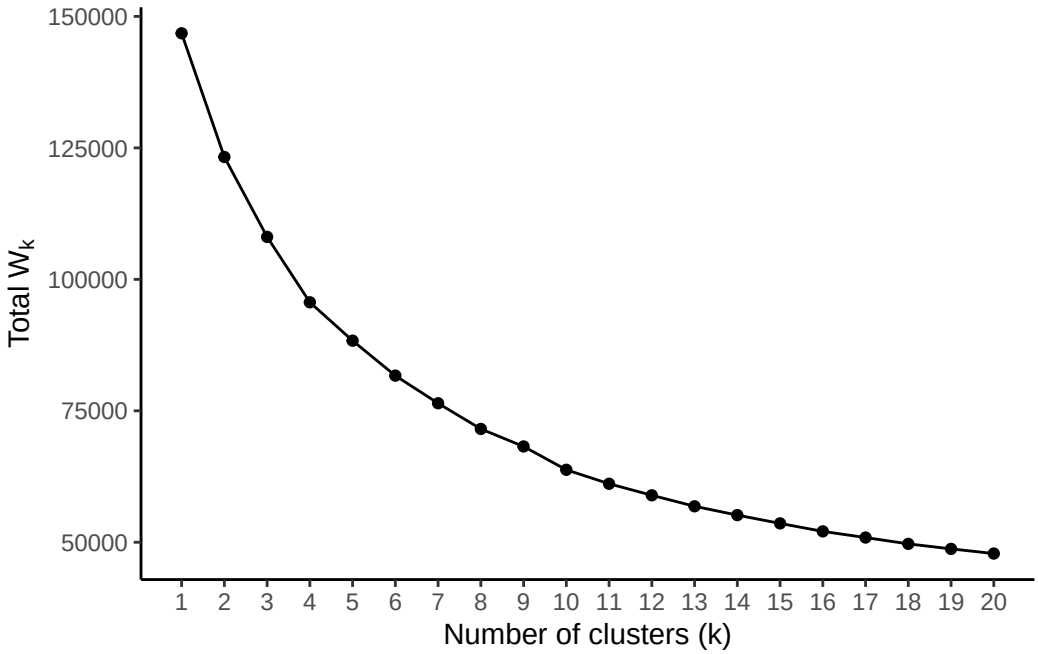
n is large here, so I chose to examine options for k from 1 to 20.

```r
elbow_plot <- data.frame(clusters = 1:20,
                         within_ss = rep(NA, 20))

set.seed(231)
for (i in 1:20){
   kmi_out <- mydata %>%
    select(ends_with("scaled")) %>%
    kmeans(centers = i, nstart = 20)

  elbow_plot$within_ss[i] <- kmi_out$tot.withinss
}

# Construct elbow plot
ggplot(elbow_plot, aes(x = clusters, y = within_ss)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = 1:20) +
  labs(x = "Number of clusters (k)", y = expression("Total W"[k]))
```



There is no clear elbow, but the drops start to diminish in size around k = 4. After k = 10, it's fairly plateaued. So, I'd choose something between those - maybe try 4 and see if it makes more sense than 3, and if not, go a bit larger. We need to refit the solution above with different

k to proceed.

References

College data is from: https://collegescorecard.ed.gov/data/