Lab 7a - K-Means Clustering

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)
```

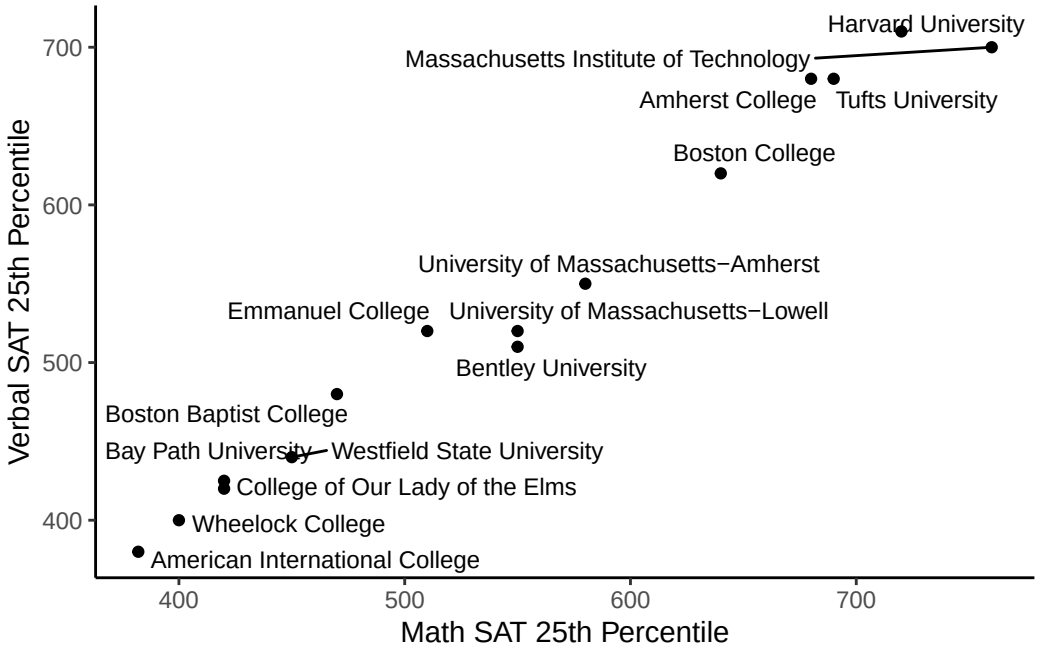
```
# A tibble: 6 x 12
  state city   institution SATMT25 SATVR25 ADM_RATE SAT_AVG GRAD_DEBT_MDN PCIP27
  <chr> <chr>  <chr>       <chr>   <chr>   <chr>    <chr>   <chr>         <chr>
1 AL    Normal Alabama A ~ 370     380     0.8738   849     32750         0.0024
2 AL    Birmi~ University~ 490     480     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")
```



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")
```

```r
ma_km2 <- ma_sample %>%
  select(clustering_vars) %>%
  kmeans(centers = 2, nstart = 20)

# Vector of cluster assignments
ma_km2$cluster
```

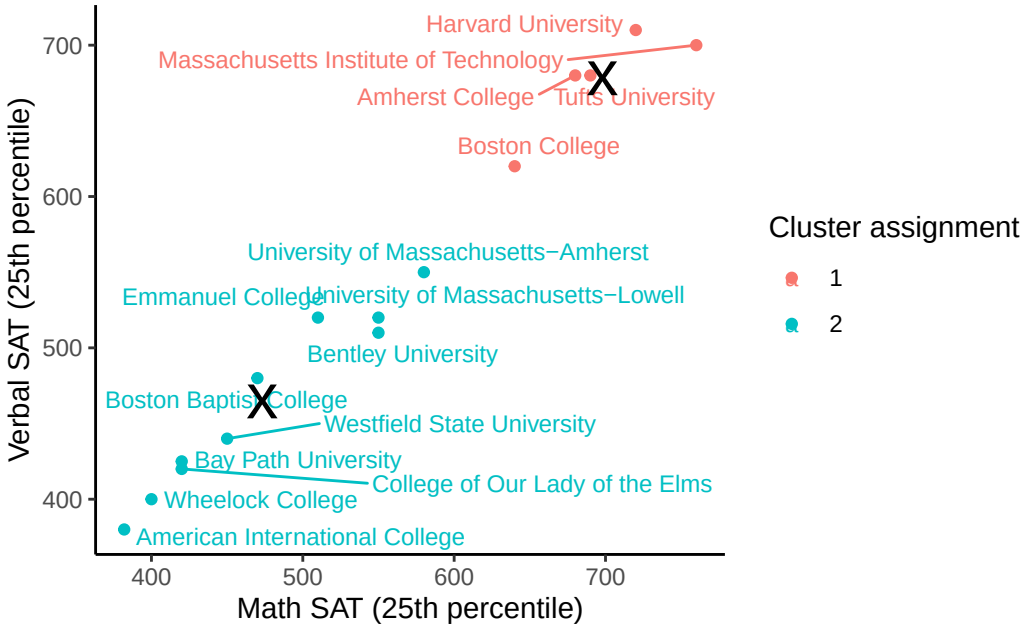
```
 [1] 2 1 2 2 2 2 1 1 1 2 2 2 2 1 2
```

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

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

```r
# 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:

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

Solution:

```
ma_km2$tot.withinss
ma_km5$tot.withinss #name depends on what you called the second solution
```

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?

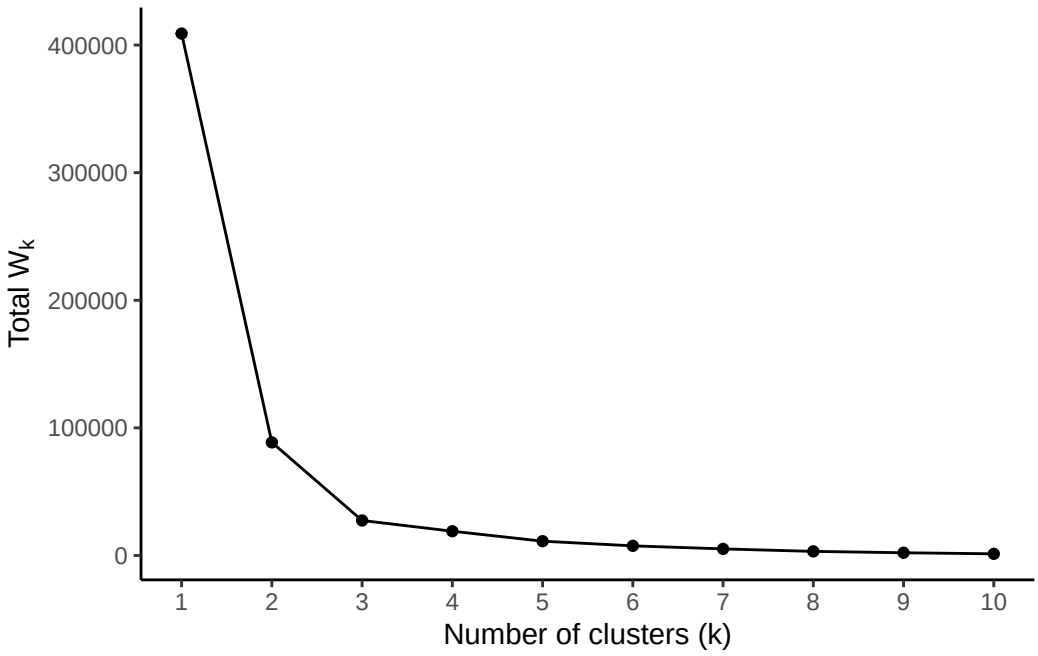
Solution:

```r
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]))
```



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~
```

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)
```

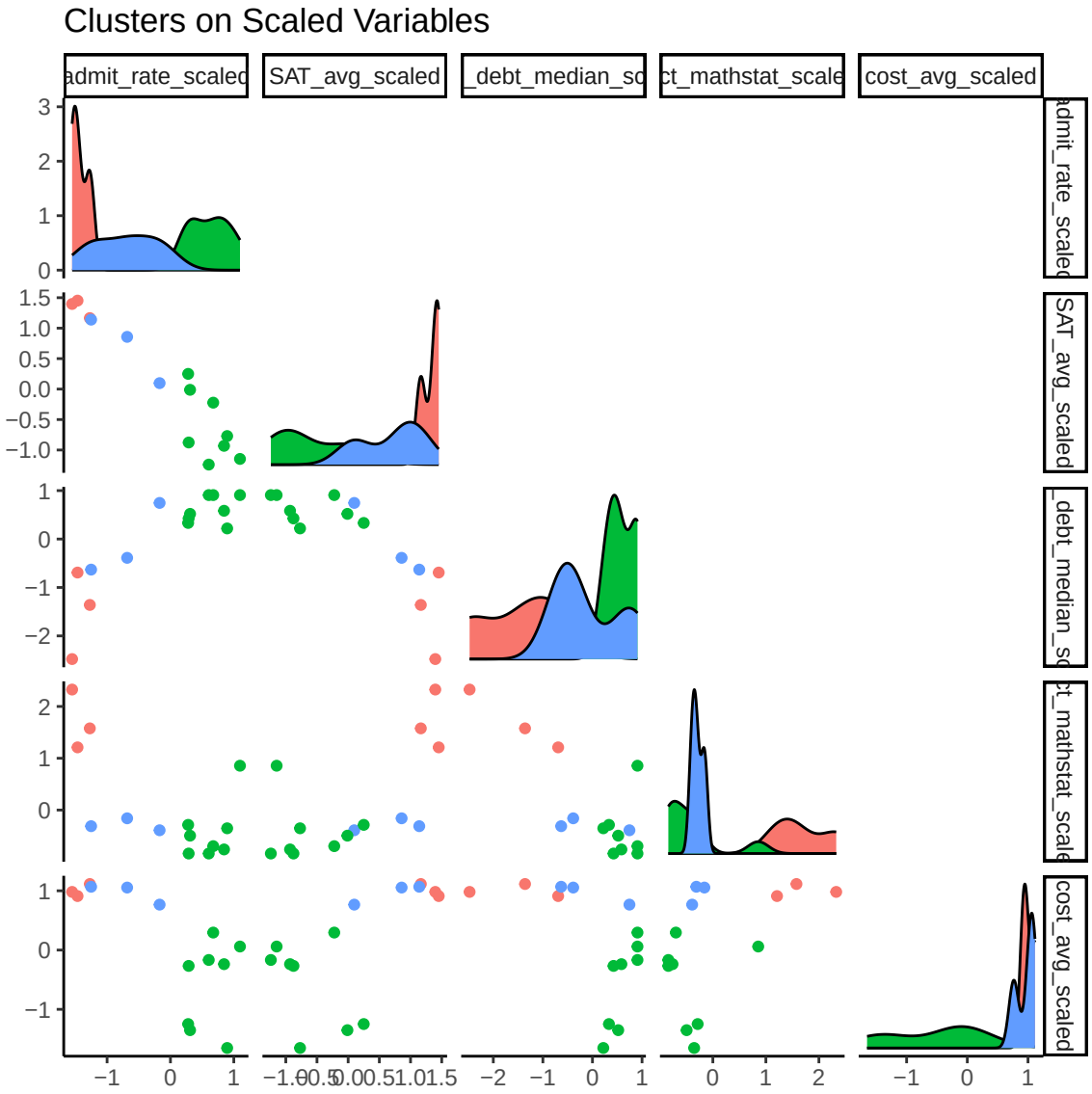
```
# A tibble: 14 x 2
   institution                          clusters3_unscaled
   <chr>                                <chr>
 1 Amherst College                      1
 2 Bentley University                   1
 3 Boston College                       1
 4 Harvard University                   1
 5 Massachusetts 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
```

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?

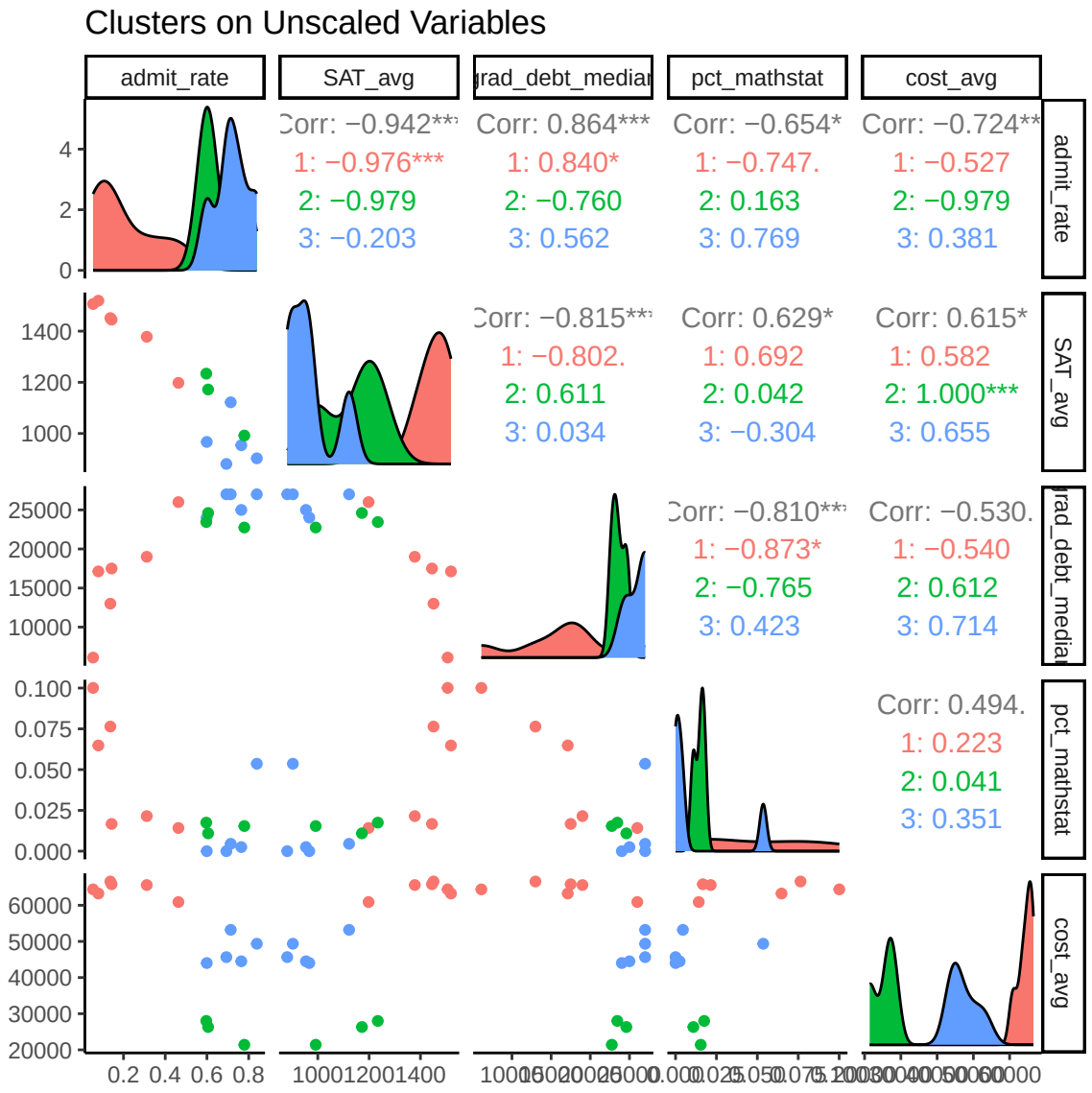
Solution:

```
# 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")
```



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")
```

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:

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

Solution:

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:

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:

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:

References

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