DA 2: Characterizing Multivariate Data

Anish Yakkala

Part Two: Numerical Summaries and Characterizations

Suppose through the miracle of time travel, you now work at US News and World Report in 1995. You would like to come up with a way to rank all the colleges, based only on the following variables: Average SAT Score, Average ACT Score, Acceptance Rate, Out-of-State Tuition, and Spending per student.

Load and edit the data, if you have not already:

```
# Read Data
colleges = read.csv('http://kbodwin.web.unc.edu/files/2016/09/tuition_final.csv')

# Adjust labels for later

colleges <- colleges %>% mutate(
  Name = gsub("California State Univ. at", "CSU", Name),
  Name = gsub("California State University at", "CSU", Name),
  Name = gsub("California Polytechnic", "Cal Poly", Name),
  Name = gsub("California Poly", "Cal Poly", Name),
  Name = gsub("University of California at", "UC", Name)
)
```

First, create a data matrix called `Y` which contains only the information we are interested in.

```
### EDIT THIS CODE###

Y <- colleges %>%
      mutate(Acc.Rate = Accepted/Applied)%>%
      select(Avg.SAT,Avg.ACT,Acc.Rate,Out.Tuition,Spending) %>%
      na.omit()

Y <- as.matrix(Y)
```

- What is the dimension of `Y`?

```
470 x 11
```

Next, find the mean and variances of each variable in `Y`, and save them as vectors. You may wish to use the function `apply()` for this task.

```
### EDIT THIS CODE###

means_Y <- colMeans(Y, na.rm = TRUE, dims = 1)

vars_Y <- apply(Y,2, var, na.rm = TRUE)
```

Now calculate the full covariance matrix of `Y`. Do this in two ways: First, use the function `cov()` to automatically find the matrix. Then, use what you know about matrix multiplication to construct the covariance from scratch. Check that your two answers are the same.

```r
# Find Covariance matrix
S <- cov(Y)

# Find it again, without using cov()
j <- matrix(1,nrow(Y),1)
Y_c <- Y - j %*% t(means_Y)

S_new <- (t(Y_c) %*% (Y_c))/(nrow(Y) - 1)
```

You are considering three possible schemes for scoring the schools:

1. *Equal weights*: You will weigh all 5 variables as equally important.
2. *Scores only*: You will only consider the SAT and ACT scores in your ranking.
3. *Complex formula*: 20% SAT Score, 20% ACT Score, 10% Acceptance Rate, 25% Tuition, 25% Spending. (This is approximately the actual weights that US News gives to these variables; however, they also include many more variables that we do not have access to data for.)

Make a matrix `A`, which should have dimension 3 by 5, that contains the weights for the three ranking schemes proposed.

```r
###  EDIT THIS CODE ###
A <- t(matrix(c(0.2,0.2,0.2,0.2,0.2,
                0.5,0.5,0,0,0,
                0.2,0.2,0.1,.25,.25), 5, 3))
```

Note that our variables of interest are on different scales; an average SAT score is around 1000, while an average acceptance rate is around 0.75. To combine these, we want our measurements to be "unitless". Make a new data matrix where each variable has been standardized by its sample mean and variance. Then calculate the mean vector and covariance matrix for this adjusted data.

```r
### EDIT THIS CODE ###
Y_std <- t(t(t(t(Y) - means_Y))/(sqrt(vars_Y)))

y_bar <- colMeans(Y_std)

S <- cov(Y_std)
```

- What values are in y-bar? What values are on the diagonal of S? Why do these make sense?

`0 since the values have been standardized. And the diagonal is 1 since we standardized them and so the s`

Use an appropriate matrix operation to find the mean score for each of the three ranking schemes.

```r
### EDIT THIS CODE ###
mean_scores <- (t(j) %*% (Y_std %*% t(A)))/nrow(Y_std)
```

Interpret these mean scores.

`The mean scores is the average score of the weighted standaridized scores.`

Use an appropriate matrix operation to find the variances of the three ranking schemes.

```r
### EDIT THIS CODE ###
vars_scores <- diag(A %*% S %*% t(A))
```

Use an appropriate matrix operation to find the scores in each of the three schemes for all the colleges in our dataset (except those with missing data). Note that this should be an n by 3 matrix, where columns

represent the three scores.

```
### EDIT THIS CODE ###

college_mat <- as.matrix(colleges %>%
                           mutate(Acc.Rate = Accepted/Applied)%>%
                           select(Avg.SAT,Avg.ACT,Acc.Rate,Out.Tuition,Spending) %>%
                           na.omit())
scores <- (Y_std %*% t(A))
```

Convert your scores matrix to a matrix of ranks; that is, assign each college a number from 1 to n based on their score, where 1 is the most "desirable" college.

```
### EDIT THIS CODE ###

ranks <- cbind(rank(scores[,1]),rank(scores[,2]),rank(scores[,3]))


sort(ranks[,1])[c(1:5)]
```

```
## 1156 1070  492 1066 1130
##    1    2    3    4    5
```

```
sort(ranks[,2])[c(1:5)]
```

```
## 1156 1070   31 1066  226
##  1.0  2.0  3.0  4.0  5.5
```

```
sort(ranks[,3])[c(1:5)]
```

```
## 1156 1070   31 1066  226
##    1    2    3    4    5
```

What are the top 10 ranked colleges for each scheme? Which scheme do you like best, based on this?

I like Model 2 since it takes into account the two most important features. A great school is made up o

```
### Make use of this code to find your answers ###

#Scheme 1
colleges %>%
  mutate(Acc.Rate = Accepted/Applied)%>%
  select(Name, Avg.SAT,Avg.ACT,Acc.Rate,Out.Tuition,Spending) %>%
  na.omit() %>%
  filter(ranks[,1] <= 10) %>%
  select(Name)
```

```
##                                   Name
## 1                       Oakwood College
## 2  University of Arkansas at Pine Bluff
## 3               Bethune Cookman College
## 4                        Webber College
## 5                Savannah State College
## 6                  Coppin State College
## 7       South Carolina State University
## 8                       Voorhees College
## 9              Huston-Tillotson College
## 10                        Texas College
```

```r
#Scheme 2
colleges %>%
  mutate(Acc.Rate = Accepted/Applied)%>%
  select(Name, Avg.SAT,Avg.ACT,Acc.Rate,Out.Tuition,Spending) %>%
  na.omit() %>%
  filter(ranks[,2] <= 10) %>%
  select(Name)
```

```
##                                   Name
## 1  University of Arkansas at Pine Bluff
## 2               Bethune Cookman College
## 3                Savannah State College
## 4                       Hilbert College
## 5       South Carolina State University
## 6                      Voorhees College
## 7              Huston-Tillotson College
## 8     Prairie View A. and M. University
## 9                        Texas College
## 10        Eastern Washington University
```

```r
#Scheme 3
colleges %>%
  mutate(Acc.Rate = Accepted/Applied)%>%
  select(Name, Avg.SAT,Avg.ACT,Acc.Rate,Out.Tuition,Spending) %>%
  na.omit() %>%
  filter(ranks[,3] <= 10) %>%
  select(Name)
```

```
##                                   Name
## 1  University of Arkansas at Pine Bluff
## 2               Bethune Cookman College
## 3                Savannah State College
## 4                  Coppin State College
## 5       South Carolina State University
## 6                      Voorhees College
## 7              Huston-Tillotson College
## 8     Prairie View A. and M. University
## 9                        Texas College
## 10             Bluefield State College
```

Where does Cal Poly rank for each scheme?

```r
### YOUR CODE HERE ###
cal_id <- colleges %>%
          filter(Name == "Cal Poly-San Luis") %>%
          pull(ID)

ranks[row.names(ranks) == cal_id]
```

```
## [1] 154 150 181
```

Your turn

Choose one or more of the ranking schemes. Find an interesting insight into college rankings based on this scheme, by studying the relationship between the ranks and at least one of the variables that we did *not* include in the ranking scheme. Support your observation with plots and numerical summaries.

```
library(tidyverse)
library(usmap)
library(broom)
```

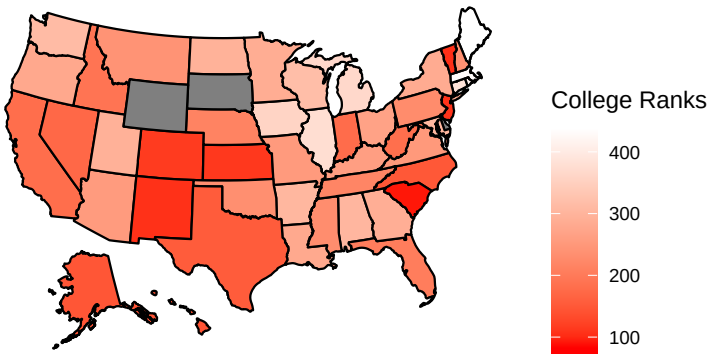
```
scheme_ranks <- tidy(ranks) %>%
                rename(index = .rownames,s1 = X1, s2 = X2, s3 = X3) %>%
                select(index,s2)

colleges %>%
    mutate(Acc.Rate = Accepted/Applied)%>%
    rownames_to_column() %>%
    na.omit() %>%
    rename(index = rowname) %>%
    inner_join(scheme_ranks,by = "index") %>%
    group_by(State) %>%
    summarize(average_rank = median(s2)) %>%
    rename(state = State) %>%
    top_n(-10,average_rank) %>%
    arrange(average_rank)
```

```
## # A tibble: 10 x 2
##    state average_rank
##    <fct>        <dbl>
##  1 DC            82.5
##  2 SC            89.5
##  3 VT           104.
##  4 NM           106.
##  5 NJ           108
##  6 KS           112
##  7 CO           115
##  8 HI           144.
##  9 AK           147
## 10 NC           148.
```

```
rank_colleges <- colleges %>%
                      mutate(Acc.Rate = Accepted/Applied)%>%
                      rownames_to_column() %>%
                      na.omit() %>%
                      rename(index = rowname) %>%
                      inner_join(scheme_ranks,by = "index") %>%
                      group_by(State) %>%
                      summarize(average_rank = median(s2)) %>%
                      rename(state = State)

plot_usmap(data = rank_colleges, values = "average_rank") +
  scale_fill_continuous(name = "College Ranks", label = scales::comma,
                        low = "red", high = "white") +
  theme(legend.position = "right")
```


College Ranks
400
300
200
100