# Wrangling Data

Dominguez Center for Data Science Workshop

2025-03-05

## Recap from last time

- Does anyone want to share a graph they made?
- Does anyone have questions?

## Plan for today

- Cover a little more `ggplot2`.
- Start going through data wrangling in `R`.
- Remember that you can either fill in the "wrangling.qmd" file or follow along with the "wrangling_key.qmd" file.

## Load Packages

The packages we need for our explorations today (`readr` for reading in data, `ggplot2` for graphing data, and `dplyr` for wrangling/summarizing the data) are part of a popular suite of packages called the `tidyverse`.

```
library(tidyverse)
library(ggrepel)
```

## Data Background

We will return to the same dataset we saw last time. Here's the background and description of the variables.

In 2013, the government decided to make data about colleges more accessible so that students and parents could more easily compare schools. These data are called the "College Scorecard" data and the 2024 dataset contains 3,305 variables on 6,484 universities in the US!

I have filtered that 2024 dataset to only include schools which confer majority baccalaureate degrees and where the majority of those degrees are in the arts and sciences based on the Carnegie Classification system. In other words, I filtered the data down to the schools which are "similar" to Bucknell (including Bucknell itself) and picked out some variables for us to explore.

### Data Dictionary

Below are the code names and descriptions of the variables in our dataset.

- `UNITID`: Unique identifier
- `INSTNM`: Name of institution
- `CITY`: City
- `STABBR`: State
- `HIGHDEG`: Highest degree awarded (0 = Non-degree grants, 1 = Certificate degree, 2 = Associate degree, 3 = Bachelor's degree, 4 = Graduate degree)
- `PREDDEG`: Predominant undergraduate degree awarded (0 = Not classified, 1 = Predominantly certificate-degree granting, 2 = Predominantly associate's-degree granting, 3 = Predominantly bachelor's-degree granting, 4 = Entirely graduate-degree granting)
- `CONTROL`: Ownership (1 = Public, 2 = Private non-profit, 3 = Private for-profit)
- `HBCU`: Flag for Historically Black College and University
- `TUITFTE`: Net tuition revenue per full-time equivalent student
- `AVGFACSAL`: Average faculty salary
- `ADM_RATE`: Admission rate
- `SATVR75`: 75th percentile of SAT scores at the institution (critical reading)
- `SATMT75`: 75th percentile of SAT scores at the institution (math)
- `ACTCM75`: 75th percentile of the ACT cumulative score

- `COSTT4_A`: The average annual total cost of attendance, including tuition and fees, books and supplies, and living expenses for all full-time, first-time, degree/certificate-seeking undergraduates who receive Title IV aid.
- `NPT4_PRIV`: The average annual total cost of attendance, including tuition and fees, books and supplies, and living expenses, minus the average grant/scholarship aid
- `UGDS`: Enrollment of undergraduate certificate/degree-seeking students
- `UG25ABV`: Percentage of undergraduates aged 25 and above
- `PCTFLOAN_DCS`: Percentage of degree/certificate-seeking undergraduate students awarded a federal loan
- `PCTPELL_DCS`: Percentage of degree/certificate-seeking undergraduate students awarded a Pell Grant
- `DEBT_MDN`: The median original amount of the loan principal upon entering repayment
- `C100_4`: Completion rate for first-time, full-time students at four-year institutions (100% of expected time to completion)
- `RET_FT4`: First-time, full-time student retention rate at four-year institutions
- `MD_EARN_WNE_5YR`: Median earnings of graduates working and not enrolled 5 years after completing

## Load the Data

Run the following code to load and inspect the data.

```
# Load the data
colleges <- read_csv("data/ccbasic21.csv")


# Wrangling data (Will talk through soon!)
colleges <- colleges %>%
  mutate(Location = if_else(STABBR == "PA", "PA", "Not PA"),
         CONTROL_CAT = case_match(CONTROL,
                                  1 ~ "Public",
                                  2 ~ "Private non-profit"),
         HIGHDEG_CAT = case_match(HIGHDEG,
                                  3 ~ "Bachelor's degree",
                                  4 ~ "Graduate degree"))
```

## Graphs Recap

**Guiding Principle**: We will map variables from the **data** to the **aes**thetic attributes (e.g. location, size, shape, color) of **geom**etric objects (e.g. points, lines, bars).

```
ggplot(data = ---, mapping = aes(---)) +
  geom_---(---)
```

## What is data wrangling??


Data wrangling = any transformations done on the data.

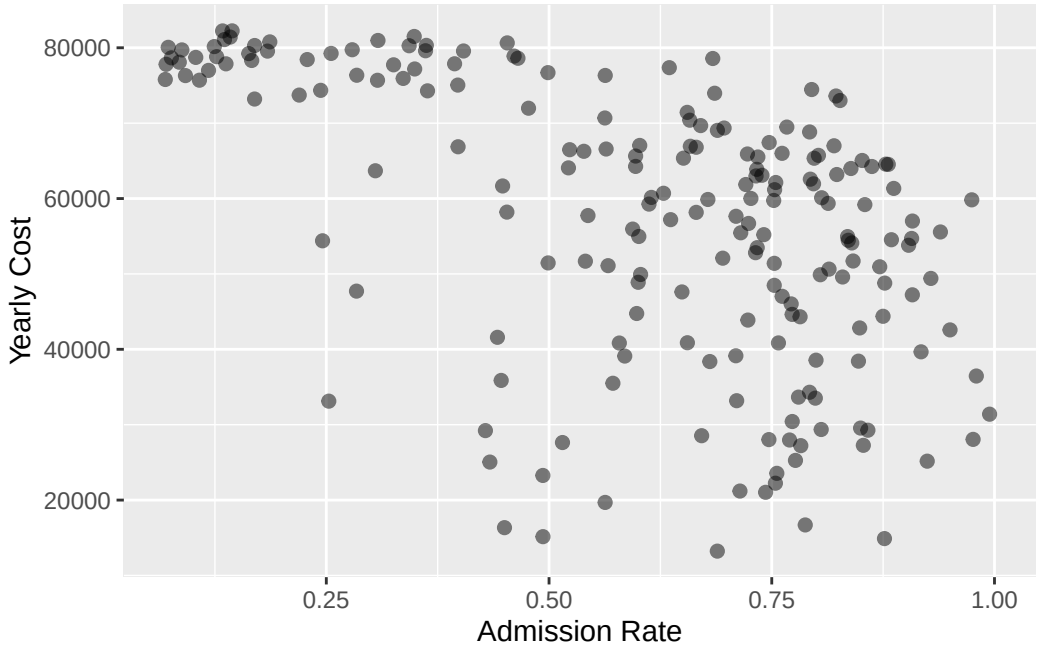
Examples:

- Summarizing the data by computing the mean of a variable.
- Counting the number of observations in the categories of a set of variables. (Excel users: Think pivot tables.)
- Dropping rows of the dataset that have missing values.
- Filtering down to just a subset of the data.
- Collapsing a categorical variable into fewer categories.
- Fixing how `R` stores a variable.
- Sorting the data by one of the variables.
- Joining multiple datasets together.
    - Won't see joins today but can learn about them here

## Motivating Examples: Sprucing up our Graphs

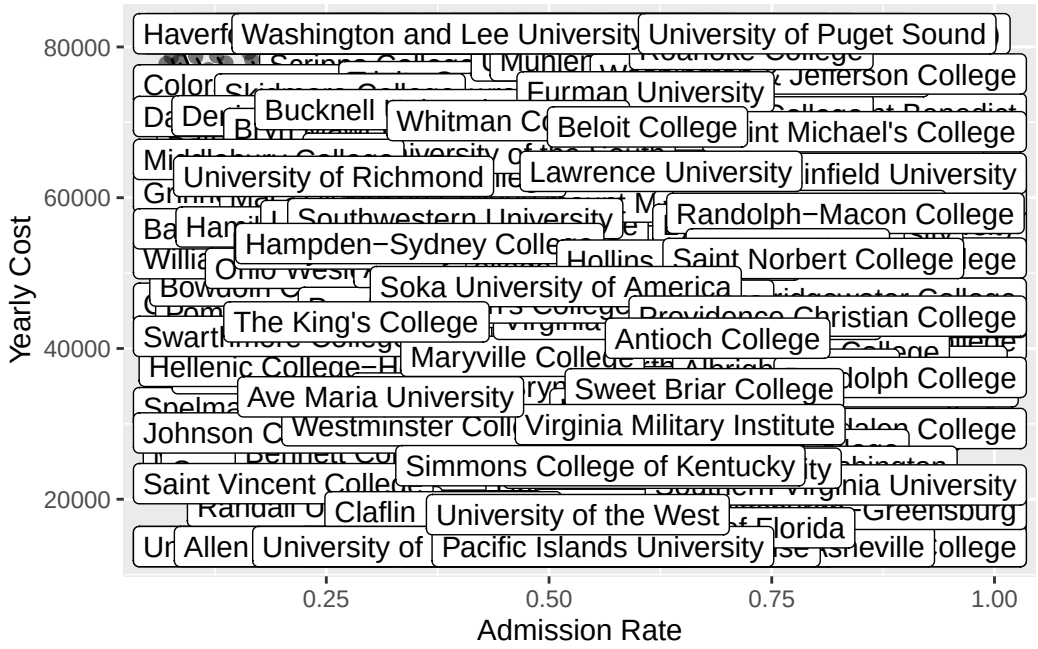
### How do I add Bucknell to my graph?

```
ggplot(data = colleges,
       mapping = aes(x = ADM_RATE,
                     y = COSTT4_A)) +
  geom_point(size = 2, alpha = .5) +
  labs(x = "Admission Rate", y = "Yearly Cost")
```



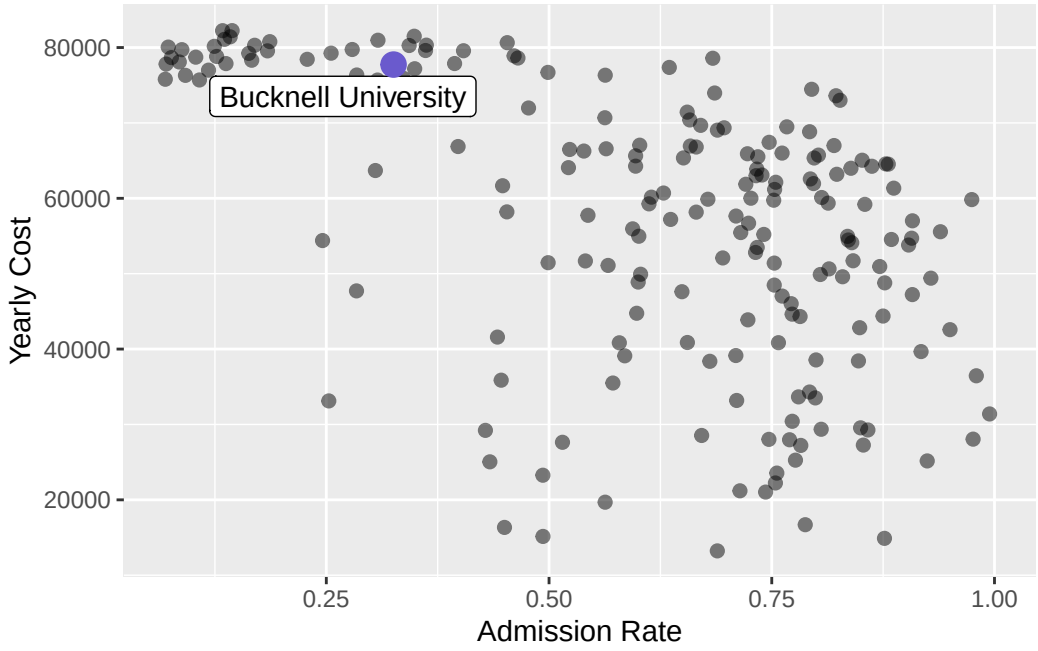
Option 1: Label all the schools.

```
ggplot(data = colleges,
       mapping = aes(x = ADM_RATE,
                     y = COSTT4_A)) +
  geom_point(size = 2, alpha = .5) +
  labs(x = "Admission Rate", y = "Yearly Cost") +
  geom_label_repel(mapping = aes(label = INSTNM), max.overlaps = 150)
```



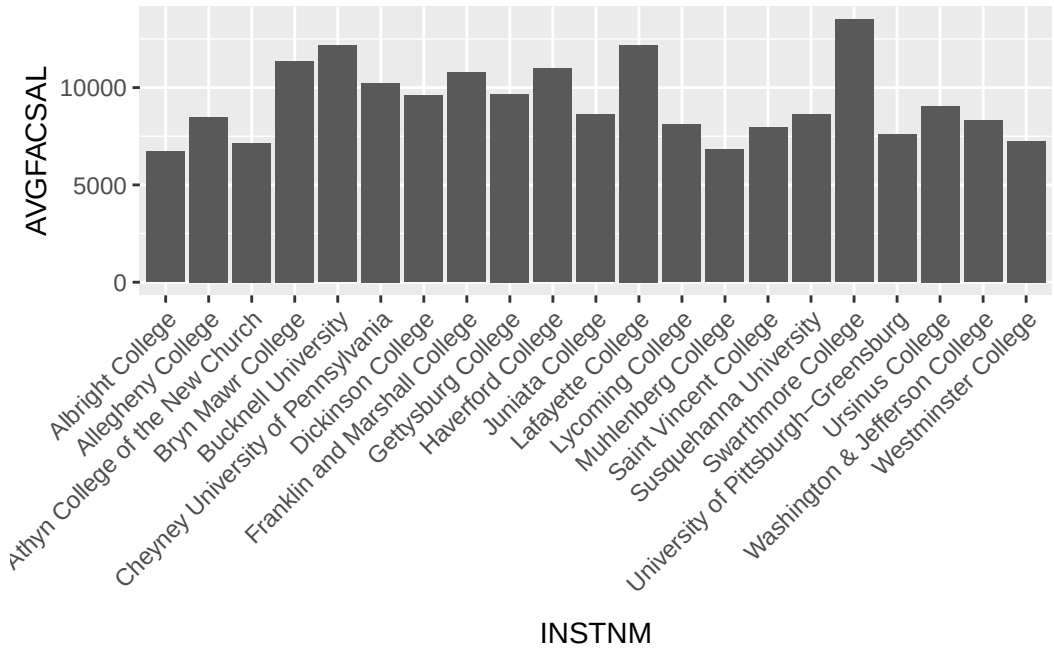

Option 2: Create a new dataset that contains only Bucknell and then add that to the graph.

```
# Create a dataset that just has Bucknell's info
Bucknell <- filter(colleges, INSTNM == "Bucknell University")

# Just label Bucknell
ggplot(data = colleges,
       mapping = aes(x = ADM_RATE,
                     y = COSTT4_A)) +
  geom_point(size = 2, alpha = .5) +
  labs(x = "Admission Rate", y = "Yearly Cost") +
  geom_label_repel(data = Bucknell, mapping = aes(label = INSTNM), max.overlaps = 20) +
  geom_point(data = Bucknell, size = 4, color = "slateblue")
```



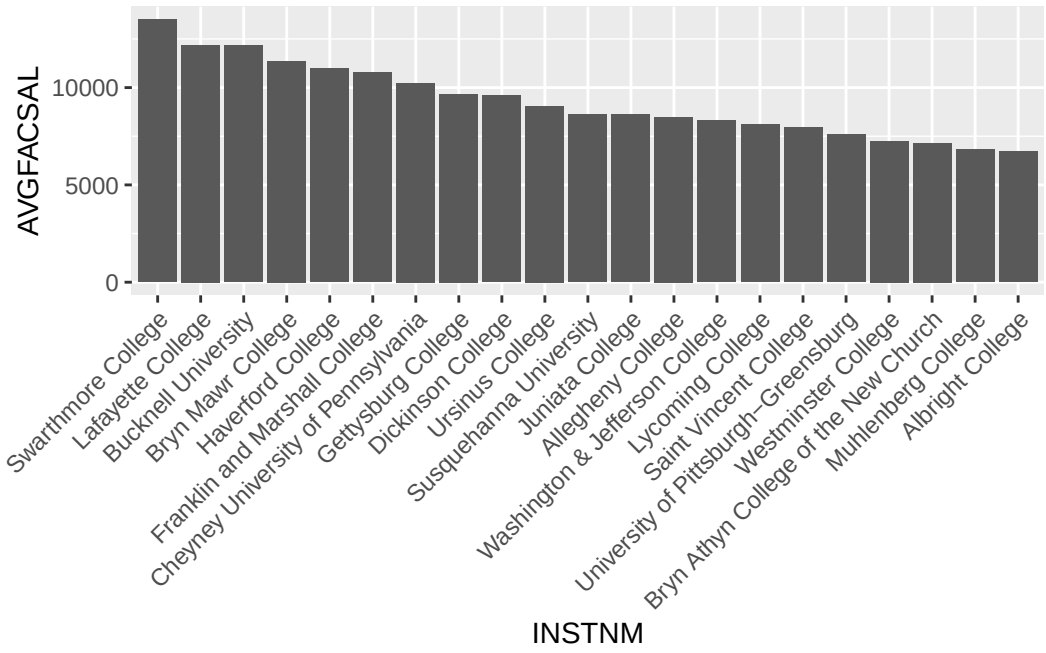

**How do I reorder the bars of my bar graph?**

```
# Let's focus on PA schools
pa_schools <- filter(colleges, STABBR == "PA")

# Look at average faculty salaries by school
ggplot(data = pa_schools, mapping = aes(x = INSTNM,
                                        y = AVGFACSAL)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




What is the current order for `INSTNM`? What is a better order for the bars?

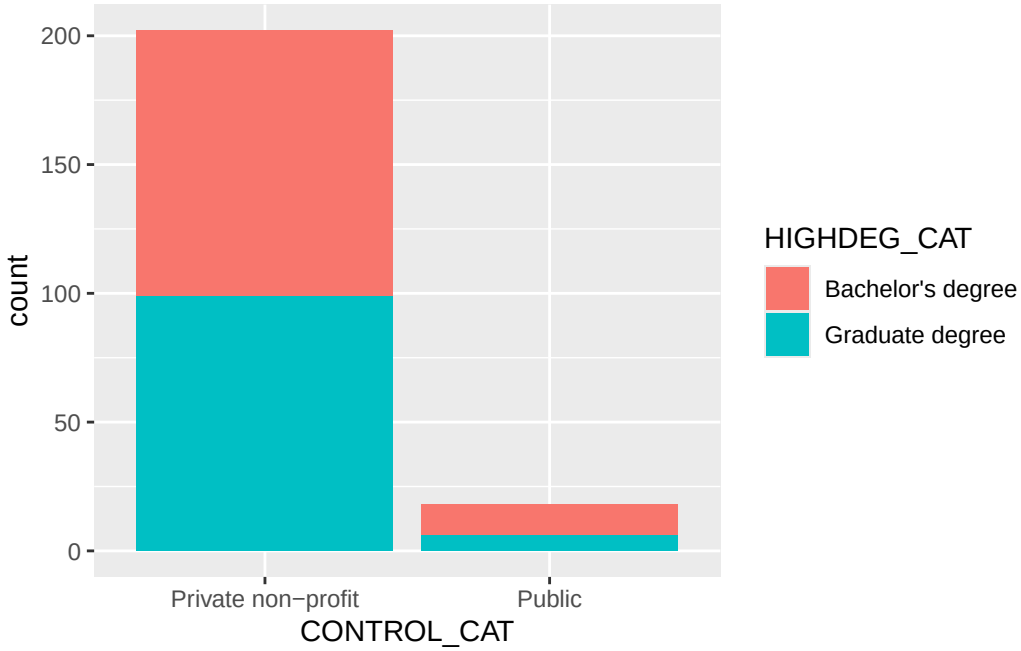
```
# Reorder institution name by the average faculty salaries
pa_schools <- mutate(pa_schools,
                     INSTNM = fct_reorder(INSTNM,
                                          -AVGFACSAL))

# Look at average faculty salaries by school
ggplot(data = pa_schools, mapping = aes(x = INSTNM,
                                        y = AVGFACSAL)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



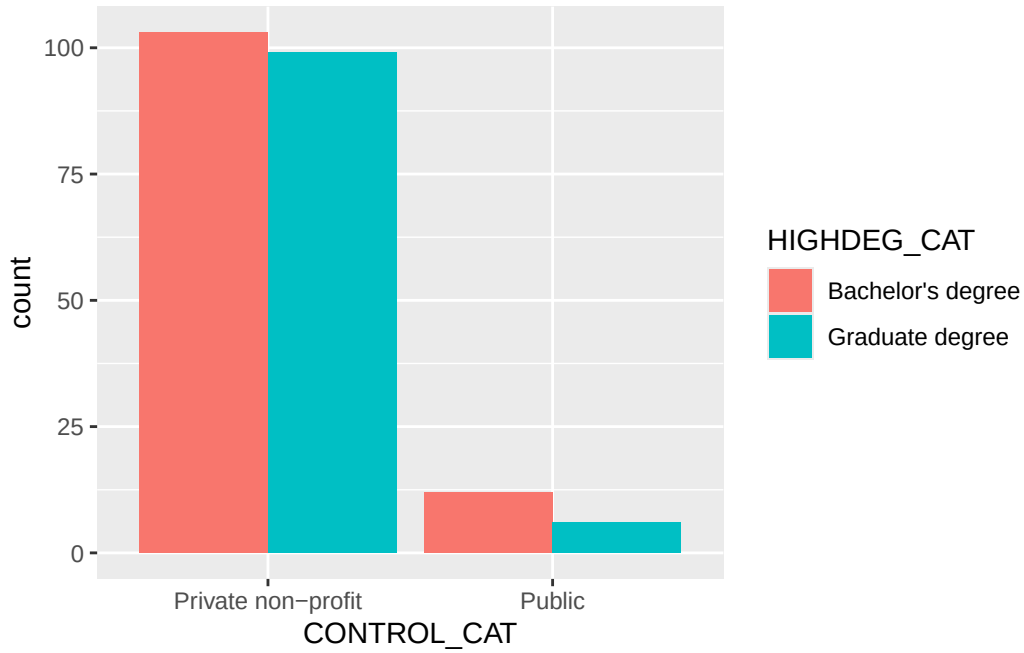

**One more `ggplot` thing: `geom_bar()` versus `geom_col()`**

Let's create bar graphs that compare the number of public and private schools by highest degree awarded.

```
# Graph of counts
ggplot(data = colleges, mapping = aes(x = CONTROL_CAT,
                                      fill = HIGHDEG_CAT)) +
  geom_bar()
```




```
# Graph of dodged counts
ggplot(data = colleges, mapping = aes(x = CONTROL_CAT,
                                      fill = HIGHDEG_CAT)) +
  geom_bar(position = "dodge")
```




```
# Graph of proportions
ggplot(data = colleges, mapping = aes(x = CONTROL_CAT,
                                      fill = HIGHDEG_CAT)) +
  geom_bar(position = "fill")
```


## Main Data Wrangling Operations in `dplyr`

### `summarize()`: Summarize variable(s)

What is the average admission rate? What is the lowest admission rate?

```
# Summarize
summarize(colleges, mean(ADM_RATE))
```

```
# A tibble: 1 x 1
  `mean(ADM_RATE)`
             <dbl>
1               NA
```

```
summarize(colleges, mean(ADM_RATE, na.rm = TRUE))
```

```
# A tibble: 1 x 1
  `mean(ADM_RATE, na.rm = TRUE)`
                           <dbl>
1                          0.601
```

```r
summarize(colleges, mean_admit = mean(ADM_RATE, na.rm = TRUE),
          lowest_admit = min(ADM_RATE, na.rm = TRUE) )
```

```
# A tibble: 1 x 2
  mean_admit lowest_admit
       <dbl>        <dbl>
1      0.601       0.0693
```

```r
# Save summary to new dataset
colleges_summary <- summarize(colleges,
                              mean_admit = mean(ADM_RATE, na.rm = TRUE),
                              lowest_admit = min(ADM_RATE, na.rm = TRUE) )
colleges_summary
```

```
# A tibble: 1 x 2
  mean_admit lowest_admit
       <dbl>        <dbl>
1      0.601       0.0693
```

### `count()`: Add up number of rows for each category

How many historically black colleges and universities are in the dataset? Of those, how many award graduate degrees?

```r
count(colleges, HBCU)
```

```
# A tibble: 2 x 2
   HBCU     n
  <dbl> <int>
1     0   203
2     1    17
```

```r
count(colleges, HBCU, HIGHDEG)
```

```
# A tibble: 4 x 3
   HBCU HIGHDEG     n
  <dbl>   <dbl> <int>
1     0       3   105
2     0       4    98
3     1       3    10
4     1       4     7
```

### `mutate()`: Modify an existing variable or add new variables

Let's re-create the `Location` variable that indicates whether or not a college is in PA. What happened to the dimensions of `colleges` once we made this change?

```
colleges <- mutate(colleges,
                   Location = if_else(STABBR == "PA", "PA", "NOT PA"))

# Check work with count()
count(colleges, Location)
```

```
# A tibble: 2 x 2
  Location     n
  <chr>    <int>
1 NOT PA     199
2 PA          21
```

Let's fix the class of `DEBT_MDN` and `HIGHDEG`. You can use `glimpse()` to see the classes of each variable. What happened to the dimensions of `colleges` once we made these changes?

```
glimpse(colleges)
```

```
Rows: 220
Columns: 27
$ UNITID          <dbl> 100937, 101912, 106342, 107080, 107512, 112260, 115409~
$ INSTNM          <chr> "Birmingham-Southern College", "Oakwood University", "~
$ CITY            <chr> "Birmingham", "Huntsville", "Batesville", "Conway", "A~
$ STABBR          <chr> "AL", "AL", "AR", "AR", "AR", "CA", "CA", "CA", "CA", ~
$ HIGHDEG         <dbl> 3, 4, 3, 4, 4, 4, 3, 4, 3, 3, 3, 4, 4, 3, 3, 4, 4, 4, ~
$ PREDDEG         <dbl> 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ~
$ CONTROL         <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, ~
$ HBCU            <dbl> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ TUITFTE         <dbl> 10340, 12279, 10188, 9685, 10749, 35957, 36122, 26331,~
$ AVGFACSAL       <dbl> 7029, 4842, 5817, 7889, 6735, 15333, 14478, 11309, 117~
$ ADM_RATE        <dbl> 0.5717, 0.6805, 0.5984, 0.6028, 0.7232, 0.1035, 0.1336~
$ SATVR75         <dbl> 670, NA, NA, 680, 610, 760, 770, NA, 750, NA, 770, 760~
$ SATMT75         <dbl> 610, NA, NA, 648, 590, 790, 790, NA, 760, NA, 790, 750~
$ ACTCM75         <dbl> 29, NA, NA, 30, 28, 35, 36, NA, 34, NA, 35, 34, NA, 32~
$ COSTT4_A        <dbl> 35495, 38377, 44749, 49928, 43878, 78723, 82236, NA, 7~
$ NPT4_PRIV       <dbl> 19723, 19686, 25183, 22780, 23086, 19489, 39671, NA, 3~
$ UGDS            <dbl> 968, 1378, 489, 1127, 1587, 1383, 906, 15, 1935, 1212,~
```

```
$ UG25ABV         <dbl> 0.0170, 0.1284, 0.0276, 0.0054, 0.0140, 0.0021, 0.0011~
$ PCTFLOAN_DCS    <dbl> 0.6452, 0.6477, 0.5934, 0.4483, 0.6109, 0.1627, 0.3646~
$ PCTPELL_DCS     <dbl> 0.2277, 0.4906, 0.3702, 0.2543, 0.2486, 0.2008, 0.1293~
$ DEBT_MDN        <chr> "16000", "21500", "10699", "19500", "15000", "11948", ~
$ C100_4          <dbl> 0.5854, 0.3351, 0.3085, 0.6743, 0.6174, 0.8318, 0.8826~
$ RET_FT4         <dbl> 0.7746, 0.7706, 0.5072, 0.7905, 0.7897, 0.9579, 0.9733~
$ MD_EARN_WNE_5YR <dbl> 56625, 51429, 45744, 49579, 48168, 108186, 154095, 418~
$ Location        <chr> "NOT PA", "NOT PA", "NOT PA", "NOT PA", "NOT PA", "NOT~
$ CONTROL_CAT     <chr> "Private non-profit", "Private non-profit", "Private n~
$ HIGHDEG_CAT     <chr> "Bachelor's degree", "Graduate degree", "Bachelor's de~
```

```
colleges <- mutate(colleges, DEBT_MDN = as.numeric(DEBT_MDN))

# Check work with glimpse()
glimpse(colleges)
```

```
Rows: 220
Columns: 27
$ UNITID          <dbl> 100937, 101912, 106342, 107080, 107512, 112260, 115409~
$ INSTNM          <chr> "Birmingham-Southern College", "Oakwood University", "~
$ CITY            <chr> "Birmingham", "Huntsville", "Batesville", "Conway", "A~
$ STABBR          <chr> "AL", "AL", "AR", "AR", "AR", "CA", "CA", "CA", "CA", ~
$ HIGHDEG         <dbl> 3, 4, 3, 4, 4, 4, 3, 4, 3, 3, 3, 4, 4, 3, 3, 4, 4, 4, ~
$ PREDDEG         <dbl> 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, ~
$ CONTROL         <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, ~
$ HBCU            <dbl> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ TUITFTE         <dbl> 10340, 12279, 10188, 9685, 10749, 35957, 36122, 26331,~
$ AVGFACSAL       <dbl> 7029, 4842, 5817, 7889, 6735, 15333, 14478, 11309, 117~
$ ADM_RATE        <dbl> 0.5717, 0.6805, 0.5984, 0.6028, 0.7232, 0.1035, 0.1336~
$ SATVR75         <dbl> 670, NA, NA, 680, 610, 760, 770, NA, 750, NA, 770, 760~
$ SATMT75         <dbl> 610, NA, NA, 648, 590, 790, 790, NA, 760, NA, 790, 750~
$ ACTCM75         <dbl> 29, NA, NA, 30, 28, 35, 36, NA, 34, NA, 35, 34, NA, 32~
$ COSTT4_A        <dbl> 35495, 38377, 44749, 49928, 43878, 78723, 82236, NA, 7~
$ NPT4_PRIV       <dbl> 19723, 19686, 25183, 22780, 23086, 19489, 39671, NA, 3~
$ UGDS            <dbl> 968, 1378, 489, 1127, 1587, 1383, 906, 15, 1935, 1212,~
$ UG25ABV         <dbl> 0.0170, 0.1284, 0.0276, 0.0054, 0.0140, 0.0021, 0.0011~
$ PCTFLOAN_DCS    <dbl> 0.6452, 0.6477, 0.5934, 0.4483, 0.6109, 0.1627, 0.3646~
$ PCTPELL_DCS     <dbl> 0.2277, 0.4906, 0.3702, 0.2543, 0.2486, 0.2008, 0.1293~
$ DEBT_MDN        <dbl> 16000, 21500, 10699, 19500, 15000, 11948, 19500, 18667~
$ C100_4          <dbl> 0.5854, 0.3351, 0.3085, 0.6743, 0.6174, 0.8318, 0.8826~
$ RET_FT4         <dbl> 0.7746, 0.7706, 0.5072, 0.7905, 0.7897, 0.9579, 0.9733~
$ MD_EARN_WNE_5YR <dbl> 56625, 51429, 45744, 49579, 48168, 108186, 154095, 418~
```

```
$ Location        <chr> "NOT PA", "NOT PA", "NOT PA", "NOT PA", "NOT PA", "NOT~
$ CONTROL_CAT     <chr> "Private non-profit", "Private non-profit", "Private n~
$ HIGHDEG_CAT     <chr> "Bachelor's degree", "Graduate degree", "Bachelor's de~
```

### `select()`: Extract variables

Let's create a new dataset that only has the school name and location.

```
colleges2 <- select(colleges, INSTNM, Location)
```

### `filter()`: Extract cases

Let's filter down to schools that are:

- In the mid-atlantic: PA, NJ, VA, MD, DE, WV, DC
- Have undergraduate enrollments over 1000 students
- Don't have grad students

```
colleges3 <- filter(colleges,
                    STABBR %in% c("PA", "NJ", "VA", "MD", "DE", "WV", "DC"),
       UGDS > 1000,
       HIGHDEG != 4)
```

Let's filter down to just Bucknell.

```
bucknell <- filter(colleges, INSTNM == "Bucknell University")
```

### `drop_na()`: Remove rows that have missing values for certain variables

Let's remove rows that are missing an admissions rate.

```
drop_na(colleges, ADM_RATE)
```

```
# A tibble: 208 x 27
   UNITID INSTNM    CITY  STABBR HIGHDEG PREDDEG CONTROL  HBCU TUITFTE AVGFACSAL
    <dbl> <chr>     <chr> <chr>    <dbl>   <dbl>   <dbl> <dbl>   <dbl>     <dbl>
 1 100937 Birmingh~ Birm~ AL           3       3       2     0   10340      7029
 2 101912 Oakwood ~ Hunt~ AL           4       3       2     1   12279      4842
 3 106342 Lyon Col~ Bate~ AR           3       3       2     0   10188      5817
```

```
 4 107080 Hendrix ~ Conw~ AR           4       3       2     0    9685      7889
 5 107512 Ouachita~ Arka~ AR           4       3       2     0   10749      6735
 6 112260 Claremon~ Clar~ CA           4       3       2     0   35957     15333
 7 115409 Harvey M~ Clar~ CA           3       3       2     0   36122     14478
 8 120254 Occident~ Los ~ CA           3       3       2     0   32778     11782
 9 121257 Pitzer C~ Clar~ CA           3       3       2     0   57556     12023
10 121345 Pomona C~ Clar~ CA           3       3       2     0   23672     14220
# i 198 more rows
# i 17 more variables: ADM_RATE <dbl>, SATVR75 <dbl>, SATMT75 <dbl>,
#   ACTCM75 <dbl>, COSTT4_A <dbl>, NPT4_PRIV <dbl>, UGDS <dbl>, UG25ABV <dbl>,
#   PCTFLOAN_DCS <dbl>, PCTPELL_DCS <dbl>, DEBT_MDN <dbl>, C100_4 <dbl>,
#   RET_FT4 <dbl>, MD_EARN_WNE_5YR <dbl>, Location <chr>, CONTROL_CAT <chr>,
#   HIGHDEG_CAT <chr>
```

### `arrange()`: Sort the cases

Let's sort rows by their admissions rate. Which schools has the lowest admissions rate? Which has the highest?

```
arrange(colleges, ADM_RATE)
```

```
# A tibble: 220 x 27
   UNITID INSTNM    CITY  STABBR HIGHDEG PREDDEG CONTROL  HBCU TUITFTE AVGFACSAL
    <dbl> <chr>     <chr> <chr>    <dbl>   <dbl>   <dbl> <dbl>   <dbl>     <dbl>
 1 216287 Swarthmo~ Swar~ PA           3       3       2     0   29620     13487
 2 121345 Pomona C~ Clar~ CA           3       3       2     0   23672     14220
 3 164465 Amherst ~ Amhe~ MA           3       3       2     0   30616     14046
 4 161086 Colby Co~ Wate~ ME           3       3       2     0   34919     11925
 5 168342 Williams~ Will~ MA           4       3       2     0   35531     14484
 6 189097 Barnard ~ New ~ NY           3       3       2     0   42671     14635
 7 161004 Bowdoin ~ Brun~ ME           3       3       2     0   34579     13417
 8 112260 Claremon~ Clar~ CA           4       3       2     0   35957     15333
 9 164155 United S~ Anna~ MD           3       3       1     0       0     12920
10 153384 Grinnell~ Grin~ IA           3       3       2     0   19898     11658
# i 210 more rows
# i 17 more variables: ADM_RATE <dbl>, SATVR75 <dbl>, SATMT75 <dbl>,
#   ACTCM75 <dbl>, COSTT4_A <dbl>, NPT4_PRIV <dbl>, UGDS <dbl>, UG25ABV <dbl>,
#   PCTFLOAN_DCS <dbl>, PCTPELL_DCS <dbl>, DEBT_MDN <dbl>, C100_4 <dbl>,
#   RET_FT4 <dbl>, MD_EARN_WNE_5YR <dbl>, Location <chr>, CONTROL_CAT <chr>,
#   HIGHDEG_CAT <chr>
```

```
arrange(colleges, desc(ADM_RATE))
```

```
# A tibble: 220 x 27
   UNITID INSTNM    CITY  STABBR HIGHDEG PREDDEG CONTROL  HBCU TUITFTE AVGFACSAL
    <dbl> <chr>     <chr> <chr>    <dbl>   <dbl>   <dbl> <dbl>   <dbl>     <dbl>
 1 172033 Sacred H~ Detr~ MI           4       1       2     0   16565      6336
 2 233611 Southern~ Buen~ VA           3       3       2     0   12302      6313
 3 182917 Magdalen~ Warn~ NH           3       3       2     0   12194      4361
 4 215275 Universi~ Gree~ PA           3       3       1     0   13107      7602
 5 154527 Wartburg~ Wave~ IA           3       3       2     0   16505      6994
 6 233301 Randolph~ Lync~ VA           4       3       2     0   12760      8249
 7 206525 Wittenbe~ Spri~ OH           4       3       2     0   12968      7836
 8 150604 Franklin~ Fran~ IN           4       3       2     0   12805      6801
 9 167288 Massachu~ Nort~ MA           4       3       1     0    6841      9334
10 165936 Gordon C~ Wenh~ MA           4       3       2     0   14956      6992
# i 210 more rows
# i 17 more variables: ADM_RATE <dbl>, SATVR75 <dbl>, SATMT75 <dbl>,
#   ACTCM75 <dbl>, COSTT4_A <dbl>, NPT4_PRIV <dbl>, UGDS <dbl>, UG25ABV <dbl>,
#   PCTFLOAN_DCS <dbl>, PCTPELL_DCS <dbl>, DEBT_MDN <dbl>, C100_4 <dbl>,
#   RET_FT4 <dbl>, MD_EARN_WNE_5YR <dbl>, Location <chr>, CONTROL_CAT <chr>,
#   HIGHDEG_CAT <chr>
```

### The pipe: %>% or |> for chaining together multiple wranglings

If you want to do multiple operations at once, you should use the pipe.

Suppose we want to look at `INSTNM`, `Location`, `ADM_RATE`, `UGDS`, `RET_FT4`, and `MD_EARN_WNE_5YR` for schools in PA that reported an admissions rate and we want to arrange the schools from largest undergraduate class to smallest undergraduate class.

```
PA_colleges <- colleges %>%
  mutate(Location = if_else(STABBR == "PA", "PA", "Not PA")) %>%
  select(INSTNM, Location, ADM_RATE, UGDS, RET_FT4, MD_EARN_WNE_5YR) %>%
  filter(Location == "PA") %>%
  drop_na(ADM_RATE) %>%
  arrange(desc(UGDS))
PA_colleges
```

```
# A tibble: 20 x 6
   INSTNM                        Location ADM_RATE  UGDS RET_FT4 MD_EARN_WNE_5YR
   <chr>                         <chr>       <dbl> <dbl>   <dbl>           <dbl>
```

```
 1 Bucknell University           PA         0.326   3732   0.906          90297
 2 Lafayette College             PA         0.336   2725   0.899          86844
 3 Gettysburg College            PA         0.563   2236   0.886          71373
 4 Susquehanna University        PA         0.767   2139   0.854          59913
 5 Dickinson College             PA         0.349   2083   0.888          71404
 6 Franklin and Marshall College PA         0.362   1986   0.878          68877
 7 Muhlenberg College            PA         0.655   1933   0.909          67290
 8 Swarthmore College            PA         0.0693  1619   0.960          73588
 9 Ursinus College               PA         0.822   1505   0.824          61871
10 Haverford College             PA         0.142   1417   0.961          69576
11 Bryn Mawr College             PA         0.308   1402   0.903          57709
12 Saint Vincent College         PA         0.734   1335   0.84           56756
13 Allegheny College             PA         0.696   1324   0.789          58614
14 University of Pittsburgh-Gre~ PA         0.976   1323   0.633          69754
15 Albright College              PA         0.849   1276   0.640          59794
16 Washington & Jefferson Colle~ PA         0.881   1139   0.829          65052
17 Juniata College               PA         0.762   1116   0.807          53474
18 Lycoming College              PA         0.752   1046   0.721          53116
19 Westminster College           PA         0.753   1023   0.822          53025
20 Bryn Athyn College of the Ne~ PA         0.800    271   0.776          38029
```

### `group_by()`: Perform actions by certain groups

For each of the Mid-Atlantic states, what is the average admission rate and how many schools are in each state?

```r
filter(colleges, STABBR %in% c("PA", "NJ", "VA", "MD", "DE", "WV", "DC")) %>%
  drop_na(ADM_RATE) %>%
  group_by(STABBR) %>%
  summarize(mean_admit = mean(ADM_RATE), count = n())
```

```
# A tibble: 5 x 3
  STABBR mean_admit count
  <chr>       <dbl> <int>
1 MD          0.586     5
2 NJ          0.754     2
3 PA          0.595    20
4 VA          0.718    16
5 WV          0.649     1
```

### Closing Thoughts on Wrangling

- Data are messy. Be prepared to wrangle.
- Before you start writing code ask yourself, what do I expect the wrangled data to look like? How many rows do I expect? How many columns?
- Don't try to wrangle all at once.
  - Write one line of code. Run it. And then keep going.
- Give the wrangled dataset a new name if you are removing rows or changing the structure drastically.

### Your Optional Homework

If using your own data, do some wrangling that help answer questions of interest to you.

For the provided data, try to complete the following tasks.

a. How many schools are in each of the categories of `PREDDEG`?

```
count(colleges, PREDDEG)
```

```
# A tibble: 2 x 2
  PREDDEG     n
    <dbl> <int>
1       1     2
2       3   218
```

b. Create a dataset that only contains schools that are predominantly bachelor's degree granting. Use this dataset for the following questions.

```
colleges_b <- filter(colleges, PREDDEG == 3)
```

c. Compute the minimum, maximum, and median values of the median earnings of graduates working and not enrolled 5 years after completing. Useful `R` functions here are: `min()`, `max()`, `median()`.

```
colleges_b %>%
  drop_na(MD_EARN_WNE_5YR) %>%
  summarize(min_earn = min(MD_EARN_WNE_5YR), max_earn = max(MD_EARN_WNE_5YR),
            med_earn = median(MD_EARN_WNE_5YR))
```

```
# A tibble: 1 x 3
  min_earn max_earn med_earn
     <dbl>    <dbl>    <dbl>
1    29334   154095   54626.
```

d. Repeat part (c) but this time compute the summary statistics for both HBCUs and non-HBCUs.

```
colleges_b %>%
  group_by(HBCU) %>%
  drop_na(MD_EARN_WNE_5YR) %>%
  summarize(min_earn = min(MD_EARN_WNE_5YR), max_earn = max(MD_EARN_WNE_5YR),
            med_earn = median(MD_EARN_WNE_5YR))
```

```
# A tibble: 2 x 4
   HBCU min_earn max_earn med_earn
  <dbl>    <dbl>    <dbl>    <dbl>
1     0    29334   154095    55755
2     1    35387    62234    44269
```

e. Ask some of your own questions of the data and then wrangle the data in order to answer them.

## Resources for Learning More about Data Wrangling with `dplyr`

- Modern Dive's chapter on Data Wrangling
- R for Data Science's chapter on Data Transformation
- `dplyr` cheatsheet: https://raw.githubusercontent.com/rstudio/cheatsheets/main/data-transformation.pdf