Intro to R

Part 2: Functions and Objects

Prof. Bisbee

Vanderbilt University

Lecture Date: 2023/01/18

Slides Updated: 2023-01-02

Agenda

1. Recap of last lecture
    - Using packages: `install.packages()` & `require()`
    - Loading and manipulating data: `readRDS()` and `%>%`
2. `tidyverse` functions
    - `filter` and `select`
    - `summarize` and `mutate`
    - `group_by`

Loading Packages & Data

- Create an `.Rmd` file and save to your `code` folder
  - Accept defaults, Save As... (with a good name), then `knit`
- Load the `tidyverse` package

```
require(tidyverse)
```

- Download `sc_debt.Rds` from GitHub and save to your `./data` folder
- Now load the data with `readRDS("[PATH TO DATA]/sc_debt.Rds")`
  - We **create** an "object" to store the data using a left-arrow: `<-`

```
df <- readRDS("../data/sc_debt.Rds")
```

- NB: `../` means "go up one folder"

Looking at Data

- We now have the contents of `sc_debt.Rds` stored in the object `df`
- We can look at this object directly

```
df
```

```
## # A tibble: 2,546 × 16
##    unitid instnm       stabbr grad_…¹ control region preddeg
##     <int> <chr>        <chr>    <int> <chr>   <chr>  <chr>
##  1 100654 Alabama A &… AL       33375 Public  South… Bachel…
##  2 100663 University … AL       22500 Public  South… Bachel…
##  3 100690 Amridge Uni… AL       27334 Private South… Associ…
##  4 100706 University … AL       21607 Public  South… Bachel…
##  5 100724 Alabama Sta… AL       32000 Public  South… Bachel…
##  6 100751 The Univers… AL       23250 Public  South… Bachel…
##  7 100760 Central Ala… AL       12500 Public  South… Associ…
##  8 100812 Athens Stat… AL       19500 Public  South… Bachel…
##  9 100830 Auburn Univ… AL       24826 Public  South… Bachel…
## 10 100858 Auburn Univ… AL       21281 Public  South… Bachel…
## # … with 2,536 more rows, 9 more variables: openadmp <int>,
## #   adm_rate <dbl>, ccbasic <int>, sat_avg <int>,
## #   md_earn_wne_p6 <int>, ugds <int>, costt4_a <int>,
```

Looking at Data

- Or we can look at its columns

```
names(df)
```

```
##  [1] "unitid"          "instnm"          "stabbr"
##  [4] "grad_debt_mdn"   "control"         "region"
##  [7] "preddeg"         "openadmp"        "adm_rate"
## [10] "ccbasic"         "sat_avg"         "md_earn_wne_p6"
## [13] "ugds"            "costt4_a"        "selective"
## [16] "research_u"
```

Good Data has Codebooks!

Name	Definition
unitid	Unit ID
instnm	Institution Name
stabbr	State Abbreviation
grad_debt_mdn	Median Debt of Graduates
control	Control Public or Private
region	Census Region
preddeg	Predominant Degree Offered: Assocates or Bachelors
openadmp	Open Admissions Policy: 1=Yes, 2=No, 3=No 1st time students
adm_rate	Admissions Rate: proportion of applications accepted
ccbasic	Type of institution*
sat_avg	Average SAT scores
md_earn_wne_p6	Average Earnings of Recent Graduates
ugds	Number of undergraduates
costt4_a	Average cost of attendance (tuition-grants)
selective	Institution admits fewer than 10% of applications, 1=Yes, 0=No
research_u	Institution is a research university, 1=Yes, 0=No

Manipulating the Data

- These data are cool!
- But TMI at first
- I want to know...
    1. Where is `Vanderbilt University`?
    2. Which school is the most selective?
    3. Which schools produce the richest grads?

Manipulating the Data

- `filter` will select **rows** of the data based on some criteria

```
df %>%
  filter(instnm == "Vanderbilt University")
```

```
## # A tibble: 1 × 16
##   unitid instnm         stabbr grad_…¹ control region preddeg
##    <int> <chr>          <chr>    <int> <chr>   <chr>  <chr>
## 1 221999 Vanderbilt U… TN       14962 Private South… Bachel…
## # … with 9 more variables: openadmp <int>, adm_rate <dbl>,
## #   ccbasic <int>, sat_avg <int>, md_earn_wne_p6 <int>,
## #   ugds <int>, costt4_a <int>, selective <dbl>,
## #   research_u <dbl>, and abbreviated variable name
## #   ¹grad_debt_mdn
```

Manipulating the Data

- Still TMI!
- `select` will select **columns**

```
df %>%
  filter(instnm == "Vanderbilt University") %>%
  select(instnm,adm_rate,selective,sat_avg,md_earn_wne_p6)
```

```
## # A tibble: 1 × 5
##   instnm                adm_rate selective sat_avg md_earn…¹
##   <chr>                    <dbl>     <dbl>   <int>     <int>
## 1 Vanderbilt University   0.0912         1    1515     53400
## # … with abbreviated variable name ¹md_earn_wne_p6
```

How does Vandy compare?

- `arrange` will sort the data based on a column (ascending!)

```
df %>%
  filter(adm_rate < .1) %>%
  arrange(sat_avg,adm_rate) %>%
  select(instnm,adm_rate,sat_avg)
```

```
## # A tibble: 25 × 3
##    instnm                                    adm_r…¹ sat_avg
##    <chr>                                       <dbl>   <int>
##  1 Colby College                              0.0967    1456
##  2 Swarthmore College                         0.0893    1469
##  3 Pomona College                             0.074     1480
##  4 Dartmouth College                          0.0793    1500
##  5 Stanford University                        0.0434    1503
##  6 Northwestern University                    0.0905    1506
##  7 Columbia University in the City of New Y…  0.0545    1511
##  8 Brown University                           0.0707    1511
##  9 University of Pennsylvania                 0.0766    1511
## 10 Vanderbilt University                      0.0912    1515
## # … with 15 more rows, and abbreviated variable name
## #   ¹adm_rate
```

How does Vandy compare?

- `arrange` in descending order

```
df %>%
  filter(adm_rate < .1) %>%
  arrange(-sat_avg,adm_rate) %>%
  select(instnm,adm_rate,sat_avg)
```

```
## # A tibble: 25 × 3
##    instnm                                  adm_r…¹ sat_avg
##    <chr>                                     <dbl>   <int>
##  1 California Institute of Technology       0.0642    1557
##  2 Massachusetts Institute of Technology    0.067     1547
##  3 University of Chicago                    0.0617    1528
##  4 Duke University                          0.076     1522
##  5 Rice University                          0.0872    1520
##  6 Harvard University                       0.0464    1517
##  7 Princeton University                     0.0578    1517
##  8 Yale University                          0.0608    1517
##  9 Vanderbilt University                    0.0912    1515
## 10 Columbia University in the City of New Y… 0.0545    1511
## # … with 15 more rows, and abbreviated variable name
## #   ¹adm_rate
```

More complicated? More %>%!

- Less selective schools by SAT with debt and state

```
df %>%
  filter(adm_rate > .2 & adm_rate < .3) %>%
  arrange(stabbr,-sat_avg) %>%
  select(instnm,sat_avg,grad_debt_mdn,stabbr)
```

```
## # A tibble: 37 × 4
##    instnm                                sat_avg grad_…¹ stabbr
##    <chr>                                   <int>   <int> <chr>
##  1 Heritage Christian University              NA      NA AL
##  2 University of California-Santa Ba…       1370   15000 CA
##  3 California Polytechnic State Univ…       1342   19501 CA
##  4 University of California-Irvine          1306   15488 CA
##  5 California Institute of the Arts           NA   27000 CA
##  6 University of Miami                      1371   17125 FL
##  7 Georgia Institute of Technology-M…       1418   23000 GA
##  8 Point University                          986   26000 GA
##  9 Grinnell College                         1457   17500 IA
## 10 St Luke's College                          NA   17750 IA
## # … with 27 more rows, and abbreviated variable name
## #   ¹grad_debt_mdn
```

A quick aside on missingness

- Some rows have `NA` in some columns
  - `NA` is the standard code for **missing data** in `R`
  - Will return to `NA` with **data wrangling**...for now, `filter` with `is.na`

```
df %>%
  filter(is.na(sat_avg)) %>%
  select(instnm,stabbr,sat_avg)
```

```
## # A tibble: 1,317 × 3
##    instnm                                  stabbr sat_avg
##    <chr>                                   <chr>    <int>
##  1 Amridge University                      AL          NA
##  2 Central Alabama Community College       AL          NA
##  3 Athens State University                 AL          NA
##  4 Chattahoochee Valley Community College  AL          NA
##  5 Coastal Alabama Community College       AL          NA
##  6 Gadsden State Community College         AL          NA
##  7 George C Wallace State Community College-… AL       NA
##  8 Heritage Christian University           AL          NA
##  9 Jefferson State Community College       AL          NA
```

Stepping back

- Thus far, lots of data
- Not a lot of science
- RQ: How might admissions and SAT scores be **related**?
  - Theory: selective schools have stricter criteria
  - Hypothesis: admissions and SAT scores should be **negatively** related
- How can we test this hypothesis?

Summarizing Data

- We can combine base `R` functions with `tidyverse` functions!
  - Base `R`: `mean()`
  - `tidyverse`: `summarise()` (aka `summarize()`)
- Overall average SAT scores

```
df %>%
  summarise(mean_sat = mean(sat_avg,na.rm=T))
```

```
## # A tibble: 1 × 1
##   mean_sat
##      <dbl>
## 1    1141.
```

Summarizing Data

- Let's unpack this

```
df %>%
  summarise(mean_sat = mean(sat_avg,na.rm=T))
```

- Create new variable `mean_sat` that contains the `mean()` of every school's average SAT score
- `na.rm=T` means we want to ignore missing data. If not?

```
df %>%
  summarise(mean_sat = mean(sat_avg))
```

```
## # A tibble: 1 × 1
##   mean_sat
##      <dbl>
## 1       NA
```

Summarizing Data

- Recall we want see if more selective schools have higher SAT scores

```
df %>%
  filter(adm_rate < .1) %>%
  summarise(mean_sat_LT10 = mean(sat_avg,na.rm=T))
```

```
## # A tibble: 1 × 1
##   mean_sat_LT10
##           <dbl>
## 1         1510.
```

```
df %>%
  filter(adm_rate > .1 & adm_rate < .2) %>%
  summarise(mean_sat_1020 = mean(sat_avg,na.rm=T))
```

```
## # A tibble: 1 × 1
##   mean_sat_1020
##           <dbl>
## 1         1424.
```

Manipulating the Data: `filter()`

- `filter()` command with other logical operators
  - `>, <`: greater than, less than (`>=, <=`)
  - `!`: not (i.e., `!=` means "not equal to")
  - `&`: and
  - `|`: or

```
df %>%
  filter(instnm != "Vanderbilt University") %>%
  select(instnm,stabbr,adm_rate,sat_avg)
```

```
## # A tibble: 2,545 × 4
##    instnm                                stabbr adm_r…¹ sat_avg
##    <chr>                                 <chr>    <dbl>   <int>
##  1 Alabama A & M University              AL       0.918     939
##  2 University of Alabama at Birmingh…    AL       0.737    1234
##  3 Amridge University                    AL      NA          NA
##  4 University of Alabama in Huntsvil…    AL       0.826    1319
##  5 Alabama State University              AL       0.969     946
##  6 The University of Alabama             AL       0.827    1261
##  7 Central Alabama Community College     AL      NA          NA
##  8 Athens State University               AL      NA          NA
```

Manipulating the Data: `str_detect()`

- `filter()` command with other functions
  - `str_detect([VAR],[PATTERN])`: detect a string
  - `grepl([PATTERN],[VAR])`: also detects a string

```
df %>%
  filter(str_detect(instnm,"Vanderbilt")) %>%
  select(instnm,stabbr,adm_rate,sat_avg)
```

```
## # A tibble: 1 × 4
##   instnm                stabbr adm_rate sat_avg
##   <chr>                 <chr>     <dbl>   <int>
## 1 Vanderbilt University TN       0.0912    1515
```

Manipulating the Data

- String detection is case sensitive!

```
df %>%
  filter(str_detect(instnm,"VAND")) %>%
  select(instnm,stabbr,adm_rate,sat_avg)
```

```
## # A tibble: 0 × 4
## # … with 4 variables: instnm <chr>, stabbr <chr>,
## #   adm_rate <dbl>, sat_avg <int>
```

```
df %>%
  filter(str_detect(instnm,"anderbil")) %>%
  select(instnm,stabbr,adm_rate,sat_avg)
```

```
## # A tibble: 1 × 4
##   instnm                stabbr adm_rate sat_avg
##   <chr>                 <chr>     <dbl>   <int>
## 1 Vanderbilt University TN       0.0912    1515
```

Manipulating the Data

- & (and) and | (or) examples

```
df %>%
  filter(str_detect(instnm,"Colorado")) %>%
  select(instnm,stabbr,adm_rate,sat_avg)
```

```
## # A tibble: 12 × 4
##    instnm                                stabbr adm_r…¹ sat_avg
##    <chr>                                 <chr>    <dbl>   <int>
##  1 University of Colorado Denver/Ans…    CO       0.673    1124
##  2 University of Colorado Colorado S…    CO       0.872    1136
##  3 University of Colorado Boulder        CO       0.784    1276
##  4 Colorado Christian University         CO      NA          NA
##  5 Colorado College                      CO       0.135      NA
##  6 Colorado School of Mines              CO       0.531    1342
##  7 Colorado State University-Fort Co…    CO       0.814    1204
##  8 Colorado Mesa University              CO       0.782    1063
##  9 University of Northern Colorado       CO       0.908    1096
## 10 Colorado State University Pueblo      CO       0.930    1047
## 11 Western Colorado University           CO       0.842    1114
## 12 Colorado State University-Global …    CO       0.986    1048
## # … with abbreviated variable name ¹adm_rate
```

Manipulating the Data

- & (and) and | (or) examples

```
df %>%
  filter(grepl("Colorado",instnm) & grepl(' of ',instnm)) %>%
  select(instnm,stabbr,adm_rate,sat_avg)
```

```
## # A tibble: 5 × 4
##   instnm                                stabbr adm_r…¹ sat_avg
##   <chr>                                 <chr>    <dbl>   <int>
## 1 University of Colorado Denver/Ansc… CO       0.673    1124
## 2 University of Colorado Colorado Sp… CO       0.872    1136
## 3 University of Colorado Boulder      CO       0.784    1276
## 4 Colorado School of Mines            CO       0.531    1342
## 5 University of Northern Colorado     CO       0.908    1096
## # … with abbreviated variable name ¹adm_rate
```

Manipulating the Data

- & (and) and | (or) examples

```
df %>%
  filter(grepl("Colorado",instnm) | grepl('Vermont',instnm)) %>%
  select(instnm,stabbr,adm_rate,sat_avg)
```

```
## # A tibble: 16 × 4
##    instnm                                stabbr adm_r…¹ sat_avg
##    <chr>                                 <chr>    <dbl>   <int>
##  1 University of Colorado Denver/Ans…    CO       0.673    1124
##  2 University of Colorado Colorado S…    CO       0.872    1136
##  3 University of Colorado Boulder        CO       0.784    1276
##  4 Colorado Christian University         CO      NA          NA
##  5 Colorado College                      CO       0.135      NA
##  6 Colorado School of Mines              CO       0.531    1342
##  7 Colorado State University-Fort Co…    CO       0.814    1204
##  8 Colorado Mesa University              CO       0.782    1063
##  9 University of Northern Colorado       CO       0.908    1096
## 10 Colorado State University Pueblo      CO       0.930    1047
## 11 Western Colorado University           CO       0.842    1114
## 12 Community College of Vermont          VT      NA          NA
## 13 Northern Vermont University           VT       0.778      NA
```

Manipulating the Data

- & (and) and | (or) examples

```
df %>%
  filter((grepl("Colorado",instnm) | grepl('Vermont',instnm)) &
grepl(' of ',instnm)) %>%
  select(instnm,stabbr,adm_rate,sat_avg)
```

```
## # A tibble: 7 × 4
##   instnm                                stabbr adm_r…¹ sat_avg
##   <chr>                                 <chr>    <dbl>   <int>
## 1 University of Colorado Denver/Ansc… CO       0.673    1124
## 2 University of Colorado Colorado Sp… CO       0.872    1136
## 3 University of Colorado Boulder      CO       0.784    1276
## 4 Colorado School of Mines            CO       0.531    1342
## 5 University of Northern Colorado     CO       0.908    1096
## 6 Community College of Vermont        VT      NA          NA
## 7 University of Vermont               VT       0.673    1287
## # … with abbreviated variable name ¹adm_rate
```

Manipulating the Data

- `&` can be separated into multiple `filter()` commands

```
df %>%
  filter((grepl("Colorado",instnm) | grepl('Vermont',instnm))) %>%
  filter(grepl(' of ',instnm)) %>%
  select(instnm,stabbr,adm_rate,sat_avg)
```

```
## # A tibble: 7 × 4
##   instnm                                stabbr adm_r…¹ sat_avg
##   <chr>                                 <chr>    <dbl>   <int>
## 1 University of Colorado Denver/Ansc… CO       0.673    1124
## 2 University of Colorado Colorado Sp… CO       0.872    1136
## 3 University of Colorado Boulder      CO       0.784    1276
## 4 Colorado School of Mines            CO       0.531    1342
## 5 University of Northern Colorado     CO       0.908    1096
## 6 Community College of Vermont        VT      NA          NA
## 7 University of Vermont               VT       0.673    1287
## # … with abbreviated variable name ¹adm_rate
```

Manipulating the Data

- `|` can be moved into the `str_detect()` or `grepl()` commands

```
df %>%
  filter(grepl("Colorado|Vermont",instnm)) %>%
  filter(grepl(' of ',instnm)) %>%
  select(instnm,stabbr,adm_rate,sat_avg)
```

```
## # A tibble: 7 × 4
##   instnm                                stabbr adm_r…¹ sat_avg
##   <chr>                                 <chr>    <dbl>   <int>
## 1 University of Colorado Denver/Ansc… CO       0.673    1124
## 2 University of Colorado Colorado Sp… CO       0.872    1136
## 3 University of Colorado Boulder      CO       0.784    1276
## 4 Colorado School of Mines            CO       0.531    1342
## 5 University of Northern Colorado     CO       0.908    1096
## 6 Community College of Vermont        VT      NA          NA
## 7 University of Vermont               VT       0.673    1287
## # … with abbreviated variable name ¹adm_rate
```

Quick Test

- Filter schools from Texas with the word "community" in their name

```
# INSERT CODE HERE
```

Manipulating the Data: `contains()` + `matches()`

- `select` can be paired with `matches` or `contains` for similar flexibility

```
df %>%
  select(contains('inst'))
```

```
## # A tibble: 2,546 × 1
##    instnm
##    <chr>
##  1 Alabama A & M University
##  2 University of Alabama at Birmingham
##  3 Amridge University
##  4 University of Alabama in Huntsville
##  5 Alabama State University
##  6 The University of Alabama
##  7 Central Alabama Community College
##  8 Athens State University
##  9 Auburn University at Montgomery
## 10 Auburn University
## # … with 2,536 more rows
```

Manipulating the Data

- `select` can be paired with `matches` or `contains` for similar flexibility
  - `matches` can work with `|`

```
df %>%
  select(!matches('_|inst'))
```

```
## # A tibble: 2,546 × 9
##    unitid stabbr control region    preddeg   opena…¹ ccbasic
##     <int> <chr>  <chr>   <chr>     <chr>       <int>   <int>
##  1 100654 AL     Public  Southeast Bachelor…       2      18
##  2 100663 AL     Public  Southeast Bachelor…       2      15
##  3 100690 AL     Private Southeast Associate       1      20
##  4 100706 AL     Public  Southeast Bachelor…       2      16
##  5 100724 AL     Public  Southeast Bachelor…       2      19
##  6 100751 AL     Public  Southeast Bachelor…       2      15
##  7 100760 AL     Public  Southeast Associate       1       2
##  8 100812 AL     Public  Southeast Bachelor…      NA      22
##  9 100830 AL     Public  Southeast Bachelor…       2      18
## 10 100858 AL     Public  Southeast Bachelor…       2      15
## # … with 2,536 more rows, 2 more variables: ugds <int>,
## #   selective <dbl>, and abbreviated variable name
```

Manipulating the Data

- `select` can also work with `where` to find classes

```
df %>%
  select(where(is.numeric))
```

```
## # A tibble: 2,546 × 11
##    unitid grad_deb…¹ opena…² adm_r…³ ccbasic sat_avg md_ea…⁴
##     <int>      <int>   <int>   <dbl>   <int>   <int>   <int>
##  1 100654      33375       2   0.918      18     939   25200
##  2 100663      22500       2   0.737      15    1234   35100
##  3 100690      27334       1  NA          20      NA   30700
##  4 100706      21607       2   0.826      16    1319   36200
##  5 100724      32000       2   0.969      19     946   22600
##  6 100751      23250       2   0.827      15    1261   37400
##  7 100760      12500       1  NA           2      NA   23100
##  8 100812      19500      NA  NA          22      NA   33400
##  9 100830      24826       2   0.904      18    1082   30100
## 10 100858      21281       2   0.807      15    1300   39500
## # … with 2,536 more rows, 4 more variables: ugds <int>,
## #   costt4_a <int>, selective <dbl>, research_u <dbl>, and
## #   abbreviated variable names ¹grad_debt_mdn, ²openadmp,
## #   ³adm_rate, ⁴md_earn_wne_p6
```

Quick Test

- Filter to only schools in California and select only character columns

```
# INSERT CODE HERE
```

Summarizing Data: `filter()` + `summarise()`

- What is the average SAT score for schools in California?

```
df %>%
  filter(stabbr == "CA") %>%
  summarise(mean_sat_CA = mean(sat_avg,na.rm=T))
```

```
## # A tibble: 1 × 1
##   mean_sat_CA
##         <dbl>
## 1       1183.
```

Quick Test

- Calculate average earnings for schools where SAT scores are higher than 1200 and the admissions rate is between 10 and 20 percent

```
# INSERT CODE HERE
```

Adding / changing variables: `mutate()`

- `mutate()` creates a new variable

```
df %>%
  mutate(newvar = 1) %>%
  select(matches('instnm|newvar'))
```

```
## # A tibble: 2,546 × 2
##    instnm                                newvar
##    <chr>                                  <dbl>
##  1 Alabama A & M University                   1
##  2 University of Alabama at Birmingham        1
##  3 Amridge University                         1
##  4 University of Alabama in Huntsville        1
##  5 Alabama State University                   1
##  6 The University of Alabama                  1
##  7 Central Alabama Community College          1
##  8 Athens State University                    1
##  9 Auburn University at Montgomery            1
## 10 Auburn University                          1
```

Object Assignment Operator: `<-`

- Thus far, nothing we have done has changed `df`
- `<-` is like "Save As..."
- `[name of object] <- [things you want saved]`

```
df <- df %>%
  mutate(adm_rate_pct = adm_rate*100)
```

- Did it work?

```
df %>%
  summarise(adm_rate_pct = mean(adm_rate_pct,na.rm=T),
            adm_rate = mean(adm_rate,na.rm=T))
```

```
## # A tibble: 1 × 2
##   adm_rate_pct adm_rate
##          <dbl>    <dbl>
## 1         67.9    0.679
```

Logic: `ifelse()`

- 3 inputs:
  - Logical statement
  - Value if the logic is `TRUE`
  - Value if the logic is `FALSE`
- `ifelse([LOGIC],[VALUE IF TRUE],[VALUE IF FALSE])`

Logic: `ifelse()`

- Say it out loud: "Create a new variable called `selective` that records if the school is selective or not. If the admissions rate is less than 10% (0.1), record the school as `selective = 1`. Otherwise, record the school as `selective = 0`."

```
df %>%
  mutate(selective = ifelse([LOGIC],
                            [VALUE IF TRUE],
                            [VALUE IF FALSE]))
```

Logic: `ifelse()`

- Say it out loud: "Create a new variable called `selective` that records if the school is selective or not. **If the admissions rate is less than 10% (0.1)**, record the school as `selective = 1`. Otherwise, record the school as `selective = 0`."

```
df %>%
  mutate(selective = ifelse(adm_rate < 0.1,
                            [VALUE IF TRUE],
                            [VALUE IF FALSE]))
```

Logic: `ifelse()`

- Say it out loud: "Create a new variable called `selective` that records if the school is selective or not. If the admissions rate is less than 10% (0.1), **record the school as `selective = 1`**. Otherwise, record the school as `selective = 0`."

```
df %>%
  mutate(selective = ifelse(adm_rate < 0.1,
                            1,
                            [VALUE IF FALSE]))
```

Logic: `ifelse()`

- Say it out loud: "Create a new variable called `selective` that records if the school is selective or not. If the admissions rate is less than 10% (0.1), record the school as `selective = 1`. **Otherwise, record the school as `selective = 0`.**"

```
df %>%
  mutate(selective = ifelse(adm_rate < 0.1,
                            1,
                            0))
```

Logic: `ifelse()` + `mutate()`

- Remember that if we want to keep this, we need the **assignment operator** `<-`

```
df <- df %>%
  mutate(selective = ifelse(adm_rate < 0.1,
                            1,
                            0))
```

Quick Test

- Create a new variable `big` that is `1` if a school has more than 10,000 undergrads and `0` otherwise

```
# INSERT CODE HERE
```

Summarizing Data

- Remember the **hypothesis** from above?
  - Schools with lower admissions rates will have higher SAT scores
- Why is this a sensible hypothesis?
  - Selective schools evaluate applicants based on SAT scores

Summarizing Data

- One final `tidyverse` function: `group_by()`

```
df %>%
  group_by(selective) %>%
  summarise(mean_sat = mean(sat_avg,na.rm=T))
```

```
## # A tibble: 3 × 2
##   selective mean_sat
##       <dbl>    <dbl>
## 1         0    1135.
## 2         1    1510.
## 3        NA     NaN
```

Conclusion

- What we've done today is a microcosm of data science
    1. Opened data (`readRDS`)
    2. Looked at data (`tidyverse` + `select()`, `filter()`, `arrange()`)
    3. Generated hypotheses (Admissions versus SAT scores)
    4. Tested hypotheses (`summarise()` + `mean()`)
- Next lecture reviews these skills and introduces **visualization**

Quiz & Homework

- Go to Brightspace and take the **3rd** quiz
  - The password to take the quiz is 3326
- **Homework:**
  1. Work through Intro_to_R_Part2_hw.Rmd