# DS-1000 Lecture 2 Notes

2024-01-16

# This is a header

## This is a subheader

### This is a subsubheader

This is plain text.

The first part of this lecture is just about how to format text in an `.Rmd` file.

- Point 1: blah blah blah
- Point 2: Teaching is **boring**
    - Point 2.a: *very* very boring

```
2 + 2
```

```
## [1] 4
```

```
objectText <- "This is a test"

objectText
```

```
## [1] "This is a test"
```

# OOL (Object Oriented Language)

- `R` can save anything (almost) to an **object** using the *object assignment operator* `<-`

```
object1 <- 2+2

object1
```

```
## [1] 4
```

# Different types of objects

- A vector is denoted with the `c()`

```
a_vector <- c(1,5,10,100,1000)
a_vector
```

```
## [1]    1    5   10  100 1000
```

Do some other stuff here…create a gap in the code.

```
a_vector
```

```
## [1]    1    5   10  100 1000
```

```
object1
```

```
## [1] 4
```

- A list is dnoted with the `list()` function

```
a_list <- list(element1 = 2+2,
               element2 = object1,
               vector1 = c(1,100,5,.2,-5),
               vector2 = c("hello world!","My name is Jim."))

a_list
```

```
## $element1
## [1] 4
##
## $element2
## [1] 4
##
## $vector1
## [1]   1.0 100.0   5.0   0.2  -5.0
##
## $vector2
## [1] "hello world!"    "My name is Jim."
```

```
a_list$vector2
```

```
## [1] "hello world!"    "My name is Jim."
```

# Comments

We can take notes out here (i.e., not inside an `R` code chunk).

```
# Or we can write comments in here

a_vector_of_words <- c('This is an','example of a vector','of words') # This creates a n
ew object containing three sentences

a_vector_of_words
```

```
## [1] "This is an"          "example of a vector" "of words"
```

# Slide 19 example

```
# Create object a containing product of 3 and 5
a <- 3*5

# Create object b containing five numbers
b <- c(10,21,43,87,175)

# Create object c which is a minus b
c <- a - b

a / b
```

```
## [1] 1.50000000 0.71428571 0.34883721 0.17241379 0.08571429
```

# Functions and Packages

```
vector_1 <- c(1,2,3)

sum(vector_1)
```

```
## [1] 6
```

```
sum(1,2,3)
```

```
## [1] 6
```

```
mean(vector_1)
```

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

```
median(vector_1)
```

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

```
# Don't do this
# install.packages('tidyverse')

require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
## Warning: package 'tibble' was built under R version 4.2.3
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'purrr' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## Warning: package 'forcats' was built under R version 4.2.3
```

```
## Warning: package 'lubridate' was built under R version 4.2.3
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.3     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.4     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts t
o become errors
```

# Loading data

- We want to load data using some of the new functions in the `tidyverse` package
- In particular, we can load data directly from the internet using the `read_rds()` or `read_csv()` functions
    - These are both **not** included with the base `R` functions, but instead need to be required with `require(tidyverse)`

```
# Loading data from the internet using read_rds()
df <- read_rds('https://github.com/jbisbee1/DS1000_S2024/raw/main/data/sc_debt.Rds')

df
```

```
## # A tibble: 2,546 × 16
##    unitid instnm   stabbr grad_debt_mdn control region preddeg openadmp adm_rate
##     <int> <chr>    <chr>          <int> <chr>   <chr>  <chr>      <int>    <dbl>
##  1 100654 Alabama… AL             33375 Public  South… Bachel…        2    0.918
##  2 100663 Univers… AL             22500 Public  South… Bachel…        2    0.737
##  3 100690 Amridge… AL             27334 Private South… Associ…        1   NA
##  4 100706 Univers… AL             21607 Public  South… Bachel…        2    0.826
##  5 100724 Alabama… AL             32000 Public  South… Bachel…        2    0.969
##  6 100751 The Uni… AL             23250 Public  South… Bachel…        2    0.827
##  7 100760 Central… AL             12500 Public  South… Associ…        1   NA
##  8 100812 Athens … AL             19500 Public  South… Bachel…       NA   NA
##  9 100830 Auburn … AL             24826 Public  South… Bachel…        2    0.904
## 10 100858 Auburn … AL             21281 Public  South… Bachel…        2    0.807
## # ℹ 2,536 more rows
## # ℹ 7 more variables: ccbasic <int>, sat_avg <int>, md_earn_wne_p6 <int>,
## #   ugds <int>, costt4_a <int>, selective <dbl>, research_u <dbl>
```

# Looking at Vandy

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

```
## # A tibble: 1 × 4
##   instnm                sat_avg adm_rate md_earn_wne_p6
##   <chr>                   <int>    <dbl>          <int>
## 1 Vanderbilt University    1515   0.0912          53400
```

# Looking at schools with a 1515 SAT score

```
df %>%
  filter(sat_avg == 1515)
```

```
## # A tibble: 1 × 16
##   unitid instnm    stabbr grad_debt_mdn control region preddeg openadmp adm_rate
##    <int> <chr>     <chr>          <int> <chr>   <chr>  <chr>      <int>    <dbl>
## 1 221999 Vanderbi… TN             14962 Private South… Bachel…        2   0.0912
## # ℹ 7 more variables: ccbasic <int>, sat_avg <int>, md_earn_wne_p6 <int>,
## #   ugds <int>, costt4_a <int>, selective <dbl>, research_u <dbl>
```

# `arrange()` example

```
df %>%
  arrange(sat_avg) %>% # Sort data by SAT scores
  select(instnm,sat_avg) # Only look at name and SAT scores
```

```
## # A tibble: 2,546 × 2
##    instnm                         sat_avg
##    <chr>                            <int>
##  1 Morgan State University            737
##  2 Saint Augustine's University       847
##  3 Albany State University            849
##  4 Holy Names University              851
##  5 Livingstone College                854
##  6 Virginia Union University          855
##  7 Manor College                      861
##  8 Saint Louis Christian College      865
##  9 Bacone College                     875
## 10 Paine College                      876
## # ℹ 2,536 more rows
```

```
df %>%
  arrange(desc(sat_avg)) %>% # Sort data by SAT scores (descending order with desc())
  select(instnm,sat_avg)  %>%# Only look at name and SAT scores
  print(n = 20)
```

```
## # A tibble: 2,546 × 2
##    instnm                                        sat_avg
##    <chr>                                           <int>
##  1 California Institute of Technology               1557
##  2 Massachusetts Institute of Technology            1547
##  3 University of Chicago                            1528
##  4 Harvey Mudd College                              1526
##  5 Duke University                                  1522
##  6 Franklin W Olin College of Engineering           1522
##  7 Washington University in St Louis                1520
##  8 Rice University                                  1520
##  9 Yale University                                  1517
## 10 Harvard University                               1517
## 11 Princeton University                             1517
## 12 Vanderbilt University                            1515
## 13 Johns Hopkins University                         1514
## 14 Carnegie Mellon University                       1513
## 15 Columbia University in the City of New York      1511
## 16 University of Pennsylvania                       1511
## 17 Brown University                                 1511
## 18 Northwestern University                          1506
## 19 Stanford University                              1503
## 20 Dartmouth College                                1500
## # ℹ 2,526 more rows
```

# Combining functions

```
df %>%
  filter(adm_rate < .1) %>%
  arrange(sat_avg,adm_rate) %>%
  select(instnm,sat_avg,adm_rate) %>%
  print(n = 25)
```

```
## # A tibble: 25 × 3
##    instnm                                    sat_avg adm_rate
##    <chr>                                       <int>    <dbl>
##  1 Colby College                                1456   0.0967
##  2 Swarthmore College                           1469   0.0893
##  3 Pomona College                               1480   0.074
##  4 Dartmouth College                            1500   0.0793
##  5 Stanford University                          1503   0.0434
##  6 Northwestern University                      1506   0.0905
##  7 Columbia University in the City of New York  1511   0.0545
##  8 Brown University                             1511   0.0707
##  9 University of Pennsylvania                   1511   0.0766
## 10 Vanderbilt University                        1515   0.0912
## 11 Harvard University                           1517   0.0464
## 12 Princeton University                         1517   0.0578
## 13 Yale University                              1517   0.0608
## 14 Rice University                              1520   0.0872
## 15 Duke University                              1522   0.076
## 16 University of Chicago                        1528   0.0617
## 17 Massachusetts Institute of Technology        1547   0.067
## 18 California Institute of Technology           1557   0.0642
## 19 Saint Elizabeth College of Nursing             NA   0
## 20 Yeshivat Hechal Shemuel                        NA   0
## 21 Hampshire College                              NA   0.0197
## 22 Curtis Institute of Music                      NA   0.0393
## 23 Pacific Oaks College                           NA   0.0511
## 24 The Juilliard School                           NA   0.0688
## 25 Bowdoin College                                NA   0.0905
```

# Digging into missingness

- First, I want to see ONLY the schools who don't report SAT scores

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

```
## # A tibble: 1,317 × 2
##    instnm                                          stabbr
##    <chr>                                           <chr>
##  1 Amridge University                              AL
##  2 Central Alabama Community College               AL
##  3 Athens State University                         AL
##  4 Chattahoochee Valley Community College          AL
##  5 Coastal Alabama Community College               AL
##  6 Gadsden State Community College                 AL
##  7 George C Wallace State Community College-Selma AL
##  8 Heritage Christian University                   AL
##  9 Jefferson State Community College               AL
## 10 Lurleen B Wallace Community College             AL
## # ℹ 1,307 more rows
```

- Second, I want to see only the schools who **DO** report SAT scores

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

```
## # A tibble: 1,229 × 2
##    instnm                              stabbr
##    <chr>                               <chr>
##  1 Alabama A & M University            AL
##  2 University of Alabama at Birmingham AL
##  3 University of Alabama in Huntsville AL
##  4 Alabama State University            AL
##  5 The University of Alabama           AL
##  6 Auburn University at Montgomery     AL
##  7 Auburn University                   AL
##  8 Birmingham-Southern College         AL
##  9 Faulkner University                 AL
## 10 Huntingdon College                  AL
## # ℹ 1,219 more rows
```

# Relationship between admissions rate and SAT scores

- Theory: selective schools have higher criteria for SAT scores
- Hypothesis: admissions rates and SAT scores should be **negatively** related

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

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

```
# Compare selective versus non-selective schools
df %>%
  filter(adm_rate < .1) %>% # Selective schools only
  summarise(overall_avg_sat = mean(sat_avg,na.rm=T))
```

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

```
df %>%
  filter(adm_rate > .1) %>% # Non-Selective schools only
  summarise(overall_avg_sat = mean(sat_avg,na.rm=T))
```

```
## # A tibble: 1 × 1
##   overall_avg_sat
##             <dbl>
## 1           1135.
```

- More efficient coding with the `group_by()`

```
df %>%
  select(selective,adm_rate) %>%
  arrange(adm_rate) %>%
  print(n = 50)
```

```
## # A tibble: 2,546 × 2
##    selective adm_rate
##        <dbl>    <dbl>
##  1         0   0
##  2         0   0
##  3         1   0.0197
##  4         1   0.0393
##  5         1   0.0434
##  6         1   0.0464
##  7         1   0.0511
##  8         1   0.0545
##  9         1   0.0578
## 10         1   0.0608
## 11         1   0.0617
## 12         1   0.0642
## 13         1   0.067
## 14         1   0.0688
## 15         1   0.0707
## 16         1   0.074
## 17         1   0.076
## 18         1   0.0766
## 19         1   0.0793
## 20         1   0.0872
## 21         1   0.0893
## 22         1   0.0905
## 23         1   0.0905
## 24         1   0.0912
## 25         1   0.0967
## 26         0   0.103
## 27         0   0.103
## 28         0   0.108
## 29         0   0.112
## 30         0   0.113
## 31         0   0.114
## 32         0   0.118
## 33         0   0.121
## 34         0   0.123
## 35         0   0.126
## 36         0   0.129
## 37         0   0.130
## 38         0   0.130
## 39         0   0.135
## 40         0   0.137
## 41         0   0.137
## 42         0   0.138
## 43         0   0.144
## 44         0   0.145
## 45         0   0.150
## 46         0   0.154
## 47         0   0.154
## 48         0   0.154
## 49         0   0.156
```

```
## 50         0   0.157
## # ℹ 2,496 more rows
```

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

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