431 Class 03

Thomas E. Love, Ph.D.

2022-09-06

Today's Agenda

- Work in R with a familiar data set (the 15 question survey from Class 02)
- Open RStudio, load in some data and a template to write R Markdown code
    - We'll do a little typing into the template today, but just a little.
        - We'll then look at the completed R Markdown document.
        - We'll also inspect and knit the R Markdown file after all of the code is included.
    - Then we'll start over again with the slides.
- These slides walk through everything in that R Markdown document

Today's Files

From our 431-data page, or our Class 03 README (data folder), you should find:

- `431-first-r-template.Rmd`
- `quick_survey_2022.csv`

and

- `431-class03-all-code.Rmd`

in addition to the usual slide materials.

Today's Plan

We're using R Markdown to gather together into a single document:

- the code we build,
- text commenting on and reacting to that code, and
- the output of the analyses we build.

Everything in these slides is also going into our R Markdown file.

Load packages and set theme

```r
library(janitor)
library(patchwork)
library(tidyverse)

theme_set(theme_bw())
```

Loading packages in R is like opening up apps on your phone. We need to tell R that, in addition to the base functions available in the software, we also have other functions we want to use.

- Why are we loading these packages, in particular?

On the tidyverse meta-package

- The most important package is actually a series of packages called the `tidyverse`, which we'll use in every R Markdown file we create this semester.
    - The `tidyverse` includes several packages, all developed (in part) by Hadley Wickham, Chief Scientist at RStudio.
    - `dplyr` is our main package for data wrangling, cleaning and transformation
    - `ggplot2` is our main visualization package we'll use for visualization
    - other `tidyverse` help import data, work with factors and other common activities.

More on today’s packages

- The `janitor` package has some tools for examining, cleaning and tabulating data (including `tabyl()` and `clean_names()`) that we’ll use regularly.
- The `patchwork` package will help us show multiple `ggplots` together.
- It’s helpful to load the `tidyverse` package last.

Today’s Data

Our data come from the Quick 15-item Survey we did in Class 02 (pdf in Class 02 README), which we’ve done (in various forms) since 2014.

- A copy of these data (in .csv format) is on our 431-data page, and also linked on our Class 03 README.

We’ll tackle several exploratory questions of interest…

Our Questions of Interest

1. What is the distribution of pulse rates among students in 431 since 2014?
2. Does the distribution of student heights change materially over time?
3. Is a Normal distribution a good model for our data?
4. Do taller people appear to have paid less for their most recent haircut?
5. Do students have a more substantial tobacco history if they prefer to speak English or a language other than English?

Read in data from .csv file

```
quicksur_raw <-
  read_csv("c03/data/quick_survey_2022.csv", show_col_types = FALSE) |>
  clean_names()
```

- Note the <- assignment arrow to create quicksur_raw
- Here, we use read_csv to read in data from the c03/data subfolder of my R project directory which contains the quick_survey_2022.csv file from our 431-data page.
- We use show_col_types = FALSE to suppress some unnecessary output describing the column types
- We use clean_names() from the janitor package
- Note the use of the pipe |> to direct the information flow

What is the result?

```
quicksur_raw
```

```
# A tibble: 494 × 23
   student glasses english statso…¹ love_…² smoke h_left h_right hande…³
statf…⁴
     <dbl> <chr>   <chr>      <dbl>   <dbl> <dbl>  <dbl>   <dbl> <chr>
<dbl>
 1  202201 n       n              5     180     1      1       9 0.8
6
 2  202202 y       y              7     168     1      0      10 1
7
 3  202203 n       y              5     185     2      0      10 1
7
 4  202204 y       y              6     185     1      0      10 1
7
 5  202205 y       y              6     191     1     16       4 1
7
```

A more detailed look?

```
glimpse(quicksur_raw)
```

```
Rows: 494
Columns: 23
$ student     <dbl> 202201, 202202, 202203, 202204, 202205, 202206, 202207,
20…
$ glasses     <chr> "n", "y", "n", "y", "y", "y", "y", "n", "y", "n", "n",
"n"…
$ english     <chr> "n", "y", "y", "y", "y", "y", "y", "n", "y", "y", "n",
"y"…
$ statsofar   <dbl> 5, 7, 5, 6, 6, 6, 7, 7, 7, 6, 5, 3, 5, 6, 4, 5, 3, 1, 5,
5…
$ love_htcm   <dbl> 180, 168, 185, 185, 191, 183, 188, 188, 191, 183, 180,
188…
$ smoke       <dbl> 1, 1, 2, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1…
$ h_left      <dbl> 1, 0, 0, 0, 16, 3, 18, 4, 1, 6, 7, 0, 0, 4, 0, 4, 0, 1,
```

Counting Categories

```
quicksur_raw |> count(glasses)
```

```
# A tibble: 3 × 2
  glasses     n
  <chr>   <int>
1 n          78
2 y         162
3 <NA>      254
```

```
quicksur_raw |> count(glasses, english)
```

```
# A tibble: 8 × 3
  glasses english     n
  <chr>   <chr>   <int>
1 n       n          20
2 n       y          58
3 y       n          34
4 y       y         127
5 y       <NA>        1
6 <NA>    n          47
7 <NA>    y         205
8 <NA>    <NA>        2
```

Favorite Color in 2022?

```r
quicksur_raw |>
    filter(year == "2022") |>
    tabyl(favcolor) |>
    adorn_pct_formatting()
```

```
      favcolor  n percent valid_percent
   all of them  1    1.9%          1.9%
         black  2    3.7%          3.8%
          blue 24   44.4%         45.3%
         green  8   14.8%         15.1%
        maroon  1    1.9%          1.9%
        orange  4    7.4%          7.5%
          pink  1    1.9%          1.9%
        purple  2    3.7%          3.8%
           red  6   11.1%         11.3%
  royal purple  1    1.9%          1.9%
 seafoam green  1    1.9%          1.9%
         white  2    3.7%          3.8%
          <NA>  1    1.9%             -
```

Using `summary()` on Quantities

```r
quicksur_raw |>
  select(love_htcm, haircut, height_in, lastsleep) |>
  summary()
```

```
   love_htcm        haircut          height_in       lastsleep    
 Min.   :165.0   Min.   :  0.00   Min.   :57.00   Min.   : 2.00  
 1st Qu.:178.0   1st Qu.: 14.00   1st Qu.:64.00   1st Qu.: 6.00  
 Median :183.0   Median : 20.00   Median :67.00   Median : 7.00  
 Mean   :182.5   Mean   : 30.17   Mean   :67.33   Mean   : 6.94  
 3rd Qu.:188.0   3rd Qu.: 40.00   3rd Qu.:70.00   3rd Qu.: 8.00  
 Max.   :191.0   Max.   :250.00   Max.   :77.50   Max.   :12.00  
 NA's   :385     NA's   :9        NA's   :7       NA's   :6      
```

Manage the data into `qsdat`

Recall our Questions of Interest

1. What is the distribution of pulse rates among students in 431 since 2014?
2. Does the distribution of student heights change materially over time?
3. Is the Normal distribution a good model for our data?
4. Do taller people appear to have paid less for their most recent haircut?
5. Do students have a more substantial tobacco history if they prefer to speak English or a language other than English?

Variables we'll look at closely today

To address our Questions of Interest, we need these seven variables in our analytic data frame (tibble.)

- `student`: student identification (numerical code)
- `year`: indicates year when survey was taken (August)
- `english`: y = prefers to speak English, else n
- `smoke`: 1 = never smoker, 2 = quit, 3 = current
- `pulse`: pulse rate (beats per minute)
- `height_in`: student's height (in inches)
- `haircut`: price of student's last haircut (in $)

Select our variables

```
qsdat <- quicksur_raw |>
    select(student, year, english, smoke,
           pulse, height_in, haircut)
```

- The `select()` function chooses the variables (columns) we want to keep in our new tibble called `qsdat`.
- What should the result of this code look like?

What do we have now?

```r
qsdat
```

```
# A tibble: 494 × 7
   student  year english smoke pulse height_in haircut
     <dbl> <dbl> <chr>   <dbl> <dbl>     <dbl>   <dbl>
 1  202201  2022 n           1    80      69.5       2
 2  202202  2022 y           1    64      63        50
 3  202203  2022 y           2    68      73         0
 4  202204  2022 y           1    88      70        38
 5  202205  2022 y           1    60      59        25
 6  202206  2022 y           2    72      68        15
 7  202207  2022 y           1    68      71         0
 8  202208  2022 n           1    68      70        32
 9  202209  2022 y           2    96      69        60
10  202210  2022 y           1    66      76        40
# … with 484 more rows
```

Initial Numeric Summaries

- Is everything the “type” of variable it should be?
- Are we getting the summaries we want?

```r
summary(qsdat)
```

```
    student             year        english               smoke      
 Min.   :201401   Min.   :2014   Length:494         Min.   :1.000  
 1st Qu.:201633   1st Qu.:2016   Class :character   1st Qu.:1.000  
 Median :201845   Median :2018   Mode  :character   Median :1.000  
 Mean   :201848   Mean   :2018                      Mean   :1.089  
 3rd Qu.:202056   3rd Qu.:2020                      3rd Qu.:1.000  
 Max.   :202254   Max.   :2022                      Max.   :3.000  
                                                    NA's   :2      
     pulse          height_in        haircut      
 Min.   : 30.00   Min.   :57.00   Min.   :  0.00  
 1st Qu.: 65.00   1st Qu.:64.00   1st Qu.: 14.00  
 Median : 72.00   Median :67.00   Median : 20.00  
 Mean   : 73.57   Mean   :67.33   Mean   : 30.17  
 3rd Qu.: 80.00   3rd Qu.:70.00   3rd Qu.: 40.00  
 Max.   :110.00   Max.   :77.50   Max.   :250.00  
 NA's   :75       NA's   :1       NA's   :9       
```

What should we be seeing?

- Categorical variables should list the categories, with associated counts.
  - To accomplish this, the variable needs to be represented in R with a `factor`, rather than as a `character` or `numeric` variable.
- Quantitative variables should show the minimum, median, mean, maximum, etc.

```r
names(qsdat)
```

```
[1] "student"   "year"      "english"   "smoke"     "pulse"     "height_in"
[7] "haircut"
```

Change categorical variables to factors

We want the `year` and `smoke` information treated as categorical, rather than as quantitative, and the `english` information as a factor, too. Also, do we want to summarize the student ID codes?

- We use the `mutate()` function to help with this.

```r
qsdat <- qsdat |>
    mutate(year = as_factor(year),
           smoke = as_factor(smoke),
           english = as_factor(english),
           student = as.character(student))
```

- Note that it’s `as_factor()` but `as.character()`. Sigh.

Next step: Recheck the summaries and do range checks

- Do these summaries make sense?
- Are the minimum and maximum values appropriate?
- How much missingness are we to deal with?

Now, how’s our summary?

```
summary(qsdat)
```

```
   student              year        english     smoke          pulse       
 Length:494         2020   : 67   n   :101   1   :456   Min.   : 30.00  
 Class :character   2016   : 64   y   :390   2   : 28   1st Qu.: 65.00  
 Mode  :character   2019   : 61   NA's:  3   3   :  8   Median : 72.00  
                    2021   : 58              NA's:  2   Mean   : 73.57  
                    2022   : 54                         3rd Qu.: 80.00  
                    2018   : 51                         Max.   :110.00  
                    (Other):139                         NA's   :75      
   height_in          haircut      
 Min.   :57.00   Min.   :  0.00  
 1st Qu.:64.00   1st Qu.: 14.00  
 Median :67.00   Median : 20.00  
 Mean   :67.33   Mean   : 30.17  
 3rd Qu.:70.00   3rd Qu.: 40.00  
 Max.   :77.50   Max.   :250.00  
 NA's   :7       NA's   :0       
```

- Some things to look for appear on the next slide.

What to look for...

- Are we getting counts for all variables that are categorical?
    - Do the category levels make sense?
- Are we getting means and medians for all variables that are quantities?
    - Do the minimum and maximum values make sense for each of these quantities?
- Which variables have missing data, as indicated by `NA's`?

The summary for `year` is an issue

- Just to fill in the gap left by the `summary()` result, how many students responded each year?

```
qsdat |> tabyl(year) |> adorn_totals() |> adorn_pct_formatting()
```

```
  year   n percent
  2014  42    8.5%
  2015  49    9.9%
  2016  64   13.0%
  2017  48    9.7%
  2018  51   10.3%
  2019  61   12.3%
  2020  67   13.6%
  2021  58   11.7%
  2022  54   10.9%
 Total 494  100.0%
```

This is how far we got in Class 03.

See the Class 04 slides for the remainder of the materials originally posted here.

Session Information

Don’t forget to close your file with the session information.

```
sessionInfo()
```

```
R version 4.2.1 (2022-06-23 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 22000)

Matrix products: default

locale:
[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base
```