SOC6302 Statistics for Sociologists

Monica Alexander

Week 2: Exploratory Data Analysis I

Today

- A note on research questions
- Exploratory Data Analysis: summary measures and tables
- (Next week: data visualization)
- Lab: doing summary stats in R

Assignment 1 is posted!

Research questions

Asking good research questions

A usual first step in social research is formulating your research question. A good research question is

- Clear and focused
  - Well-defined
  - Not too broad or narrow in scope
- Not too easy or hard to answer
  - More than yes/no
  - Is not common knowledge
- Operationalizable
  - The relationships/outcomes/effects/patterns are able to be measured and observed

Research question

Example from last week's slides: "Say we want to study the relationship between hours studied and job placement for all university students in Canada"

Turn this into a research question: *"What is the relationship between hours studied and job placement for Canadian university students?"*

To make this operationalizable, be more specific about what we're measuring:

- Population of interest: all undergraduate students graduating in 2023 enrolled in Canadian universities
- Outcome/dependent variable of interest: "job placement" — got a job requiring degree within 12 months of graduating
- Main independent variable of interest: average hours per week studying during semesters

Research question

"What is the relationship between hours studied and job placement for Canadian university students?"

This is a good start, but to more fully understand the relationship, we may what to consider other *secondary* research questions (or, sub-questions)

Research questions

What is the relationship between hours studied and job placement for Canadian university students?

- *how does the relationship vary across major?*
- *is there still a relationship after taking into account for GPA?*

Note:

1. **bivariate** versus **multivariate** questions
2. **stratification** versus **control** variables

Exploratory Data Analysis (EDA)

What is EDA and why do we do it?

Before we even do any sort of statistical inference, we need to understand the main characteristics of our dataset.

- Helps to identify any potential issues or surprising things about our data
- Helps to check / explore / refine research questions

What is EDA and why do we do it?

EDA is all about asking:

- What types of variables do we have?
- Do we have a complete dataset, or do we have missing data or observations?
- If we have missing data, is it missing equally across observations of different types or concentrated in particular groups?
- Are there any obvious outliers or strange data points?
- What do the data 'look' like?
  - summary measures, measures of centrality, spread
  - Visualizing the data through plots and tables

Steps of EDA

1. Become familiar with size of data set (number of observations and variables available)
2. What kinds of variables are available
3. For the variables that I'm interested in, are there any missing values or other issues?
4. What does the distribution/frequency of observations look like for the variables I'm interested in? (summary measures, tables and graphs)

Summary measures of quantitative data: recap

- **Measures of central tendency**: mean, median, mode
- **Measures of spread**: range, IQR, variance, standard deviation

Correlation between two quantitative variables

Correlation is the statistical measure of the relationship between two variables. **Pearson's correlation coefficient**, r_{xy} summarizes this relationship into one number. For an observation sample of two random variables $x_1, x_2, \dots, x_n$ and $y_1, y_2, \dots, y_n$,

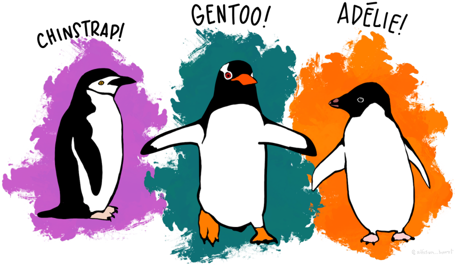
$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Correlation

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Example

Palmer penguins dataset



species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
Adelie	Torgersen	39.1	18.7	181	3750	male
Adelie	Torgersen	39.5	17.4	186	3800	female
Adelie	Torgersen	40.3	18.0	195	3250	female
Adelie	Torgersen	NA	NA	NA	NA	NA
Adelie	Torgersen	36.7	19.3	193	3450	female
Adelie	Torgersen	39.3	20.6	190	3650	male

Penguins

- Mean flipper length: 201
- Median flipper length: 197
- Standard deviation flipper length: 14
- Correlation between flipper length and bill length: 0.65

These summary measures for the whole sample. What would also be interesting to calculate?

Penguins

Mean flipper length by species. (**stratifying** by species)

species	mean_flipper	sd_flipper	corr_flipper_bill
Adelie	190.10	6.52	0.33
Chinstrap	195.82	7.13	0.47
Gentoo	217.24	6.59	0.66

Summary measures of qualitative variables

- If we have qualitative/categorical variables we can't take the mean or calculate standard deviation
- What to do?
- Summary measures are based on counting the number of units/elements of interest in each category

Frequency tables

Counts in each group

species	n
Adelie	152
Chinstrap	68
Gentoo	124

Contingency tables

Two way counts

species	Biscoe	Dream	Torgersen
Adelie	44	56	52
Chinstrap	0	68	0
Gentoo	124	0	0

Counts to proportions

species	n	proportion
Adelie	152	0.442
Chinstrap	68	0.198
Gentoo	124	0.360

Counts to proportions

Row proportions:

species	Biscoe	Dream	Torgersen
Adelie	0.29	0.37	0.34
Chinstrap	0.00	1.00	0.00
Gentoo	1.00	0.00	0.00

Counts to proportions

Column proportions:

species	Biscoe	Dream	Torgersen
Adelie	0.26	0.45	1
Chinstrap	0.00	0.55	0
Gentoo	0.74	0.00	0

A historical note

- Presenting information in tables may seem obvious to us
- Quantitative tables first used in 17th century by William Petty and John Graunt. A new science: ‘Political Arithmetic’
- John Graunt: analysis of the London Bills of Mortality

  “Graunt reduced several great confused Columns into a few perspicuous Tables and underscored the power of tables to bring clarity and conciseness to specific topics”

17. In the next place, whereas many persons live in great fear and apprehension of some of the more formidable and notorious diseases following; I shall only set down how many died of each: that the respective numbers, being compared with the total 229,250, those persons may the better understand the hazard they are in.

Table of notorious diseases

Apoplexy	1,306
Cut of the Stone	38
Falling Sickness	74
Dead in the streets	243
Gowt	134
Head-Ache	51
Jaundice	998
Lethargy	67
Leprosy	6
Lunatick	158
Overlaid, and *Starved*	529
Palsy	423
Rupture	201
Stone and *Strangury*,	863
Sciatica	5
Sodainly	454

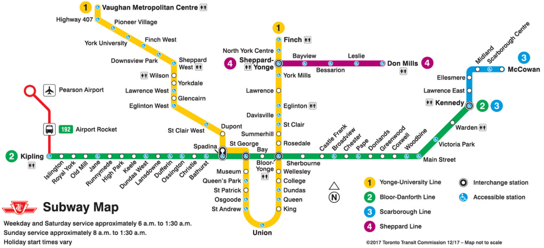
Table of casualties

Bleeding	69
Burnt, and *Scalded*	125
Drowned	829
Excessive drinking	2
Frighted	22
Grief	279
Hanged themselves	222
Killed by several accidents	1,021
Murdered	86
Poisoned	14
Smothered	26
Shot	7
Starved	51
Vomiting	136

EDA Example: TTC subway delays in 2019

Example: TTC subway delays in 2019

- Data on TTC subway delay times by station and day available from the Open Data Toronto website: https://open.toronto.ca/



- Let's get to know this dataset

Get familiar with dataset

```
delay_2019
```

```
## # A tibble: 19,222 x 11
##    date                time   day     station  code  min_d~1 min_gap bound line 
##    <dttm>              <time> <chr>   <chr>    <chr>   <dbl>   <dbl> <chr> <chr>
##  1 2019-01-01 00:00:00 01:08  Tuesday YORK MI~ PUSI        0       0 S     YU   
##  2 2019-01-01 00:00:00 02:14  Tuesday ST ANDR~ PUMST       0       0 <NA>  YU   
##  3 2019-01-01 00:00:00 02:16  Tuesday JANE     TUSC        0       0 W     BD   
##  4 2019-01-01 00:00:00 02:27  Tuesday BLOOR    SUO         0       0 N     YU   
##  5 2019-01-01 00:00:00 03:03  Tuesday DUPONT   MUATC      11      16 N     YU   
##  6 2019-01-01 00:00:00 03:08  Tuesday EGLINTO~ EUATC      11      16 S     YU   
##  7 2019-01-01 00:00:00 03:09  Tuesday DUPONT   EUATC       6      11 N     YU   
##  8 2019-01-01 00:00:00 03:26  Tuesday ST CLAI~ EUATC       4       9 N     YU   
##  9 2019-01-01 00:00:00 03:37  Tuesday KENNEDY~ TUMVS       0       0 E     BD   
## 10 2019-01-01 00:00:00 08:04  Tuesday DAVISVI~ MUNOA       5      10 S     YU   
## # ... with 19,212 more rows, 2 more variables: vehicle <dbl>, code_desc <chr>,
## #   and abbreviated variable name 1: min_delay
```

Get familiar with dataset

Dimensions (number of rows x number of columns)

```
dim(delay_2019)
```

```
## [1] 19222    11
```

Variable names

```
colnames(delay_2019)
```

```
##  [1] "date"      "time"      "day"       "station"   "code"      "min_delay"
##  [7] "min_gap"   "bound"     "line"      "vehicle"   "code_desc"
```

Research question?

- What are some good potential research questions with this dataset?

Sanity checks

We need to check variables should be what they say they are. If they aren't, the natural next question is to what to do with issues (recode? remove?)

E.g. check days of week make sense

```
delay_2019 |>
  select(day) |>
  unique()
```

```
## # A tibble: 7 x 1
##   day
##   <chr>
## 1 Tuesday
## 2 Wednesday
## 3 Thursday
## 4 Friday
## 5 Saturday
## 6 Sunday
## 7 Monday
```

Sanity checks

Check lines: oh no. some issues here. Some have obvious recodes, others, not so much.

```r
delay_2019 |>
  select(line) |>
  unique() |>
  pull() # turn into a vector for better display
```

```
##  [1] "YU"                     "BD"                     "YU/BD"
##  [4] "SHP"                    "SRT"                    NA
##  [7] "YUS"                    "B/D"                    "BD LINE"
## [10] "999"                    "YU/ BD"                 "YU & BD"
## [13] "BD/YU"                  "YU\\BD"                 "46 MARTIN GROVE"
## [16] "RT"                     "BLOOR-DANFORTH"         "YU / BD"
## [19] "134 PROGRESS"           "YU - BD"                "985 SHEPPARD EAST EXPR"
## [22] "22 COXWELL"             "100 FLEMINGDON PARK"    "YU LINE"
```

Data issues

How bad is the mislabeling of lines? look at frequency of cases

```
delay_2019 |>
  group_by(line) |> # group by line label
  tally() |> # count the number of occurrences
  arrange(-n) # arrange in descending order
```

```
## # A tibble: 24 x 2
##    line        n
##    <chr>   <int>
##  1 YU       9275
##  2 BD       8200
##  3 SRT       699
##  4 SHP       600
##  5 YU/BD     356
##  6 <NA>       50
##  7 YU / BD    16
##  8 YUS         6
##  9 YU/ BD      3
## 10 999         2
## # ... with 14 more rows
```

Missing values

```
delay_2019  |>
  summarise(across(everything(), ~ sum(is.na(.x))))
```

```
## # A tibble: 1 x 11
##    date  time   day station  code min_delay min_gap bound  line vehicle code_d~1
##   <int> <int> <int>   <int> <int>     <int>   <int> <int> <int>   <int>    <int>
## 1     0     0     0       0     0         0       0  4380    50       0     1001
## # ... with abbreviated variable name 1: code_desc
```

Summary statistics

Most interested in delay minutes, which is the `min_delay` variable

```
delay_2019 |>
  summarize(n_obs = n(),
            mean_delay = mean(min_delay),
            median_delay = median(min_delay),
            range_delay = max(min_delay) - min(min_delay),
            iqr_delay = IQR(min_delay))
```

```
## # A tibble: 1 x 5
##   n_obs mean_delay median_delay range_delay iqr_delay
##   <int>      <dbl>        <dbl>       <dbl>     <dbl>
## 1 18697       2.43            0         455         3
```

Summary statistics

Probably more interesting to do these summaries by line (**stratify** by line)

```
delay_2019 |>
  group_by(line) |>
  summarize(n_obs = n(),
            mean_delay = mean(min_delay),
            median_delay = median(min_delay),
            range_delay = max(min_delay) - min(min_delay),
            iqr_delay = IQR(min_delay))
```

```
## # A tibble: 4 x 6
##   line  n_obs mean_delay median_delay range_delay iqr_delay
##   <chr> <int>      <dbl>        <dbl>       <dbl>     <dbl>
## 1 BD     8197       2.11            0         180       3  
## 2 SHP     598       2.20            0         165       3  
## 3 SRT     631       5.79            3         284       5.5
## 4 YU     9271       2.50            0         455       3  
```

Summaries

Could also stratify by reason for delay

```
delay_2019 |>
  group_by(code_desc) |>
  summarize(n_obs = n(),
            mean_delay = mean(min_delay),
            median_delay = median(min_delay),
            range_delay = max(min_delay) - min(min_delay),
            iqr_delay = IQR(min_delay)) |>
  arrange(-n_obs)
```

```
## # A tibble: 119 x 6
##    code_desc                               n_obs mean_~1 media~2 range~3 iqr_d~4
##    <chr>                                   <int>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Miscellaneous Speed Control              1997 0.186         0      19       0
##  2 Injured or ill Customer (In Station) -~  1747 0.151         0      54       0
##  3 Operator Overspeeding                    1379 0.114         0       8       0
##  4 Passenger Assistance Alarm Activated -~  1353 0.800         0      12       0
##  5 Disorderly Patron                        1147 3.02          3      23       4
##  6 <NA>                                      931 4.19          0     284       5
##  7 Injured or ill Customer (On Train) - M~   671 3.92          3      50       5
##  8 Escalator/Elevator Incident               605 0.00826       0       5       0
##  9 Speed Control Equipment                   527 0.436         0      30       0
## 10 ATC Project                               514 3.88          3      28       5
## # ... with 109 more rows, and abbreviated variable names 1: mean_delay,
## #   2: median_delay, 3: range_delay, 4: iqr_delay
```

Summaries

Arrange by mean delay time

```
delay_2019 |>
  group_by(code_desc) |>
  summarize(n_obs = n(),
            mean_delay = mean(min_delay),
            median_delay = median(min_delay),
            range_delay = max(min_delay) - min(min_delay),
            iqr_delay = IQR(min_delay)) |>
  arrange(-mean_delay)
```

```
## # A tibble: 119 x 6
##    code_desc                               n_obs mean_~1 media~2 range~3 iqr_d~4
##    <chr>                                   <int>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Traction Power Rail Related                 1   145     145         0     0
##  2 Priority One - Train in Contact With P~    24    78.8    80       193    70.2
##  3 Structure Related Problem                   4    70.5    27       228    97.5
##  4 Rail Related Problem                        8    58.6     3       455     4
##  5 Fire/Smoke Plan A                           6    50      11.5     250    17.5
##  6 Bomb Threat                                12    36.7    20       130    32
##  7 Fire/Smoke Plan B - Source TTC             84    19.4    11       180    16.2
##  8 Doors Open in Error                        11    18.7    16        40     7.5
##  9 Fire/Smoke Plan B - Source External to~     2    13.5    13.5      19     9.5
## 10 Suspicious Package                         14    13       3.5      67    22
## # ... with 109 more rows, and abbreviated variable names 1: mean_delay,
## #   2: median_delay, 3: range_delay, 4: iqr_delay
```

EDA: summary so far

- There's no one checklist of things to looks at, depends on your data and research question
- Get familiar with your dataset
- Check for missing values, and that existing values make sense
- Summary statistics depend on your research question of interest
  - quantitative versus qualitative summary measures
  - stratifying by important characteristics often useful

Preview: visualization

It’s so important to plot your data!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Imagine we have the following sets of datasets of (x,y) pairs

```
library(tidyverse)
library(datasauRus)
head(datasaurus_dozen)
```

```
## # A tibble: 6 x 3
##   dataset     x     y
##   <chr>   <dbl> <dbl>
## 1 dino     55.4  97.2
## 2 dino     51.5  96.0
## 3 dino     46.2  94.5
## 4 dino     42.8  91.4
## 5 dino     40.8  88.3
## 6 dino     38.7  84.9
```

How many observations?

```
datasaurus_dozen  %>%  count(dataset)
```

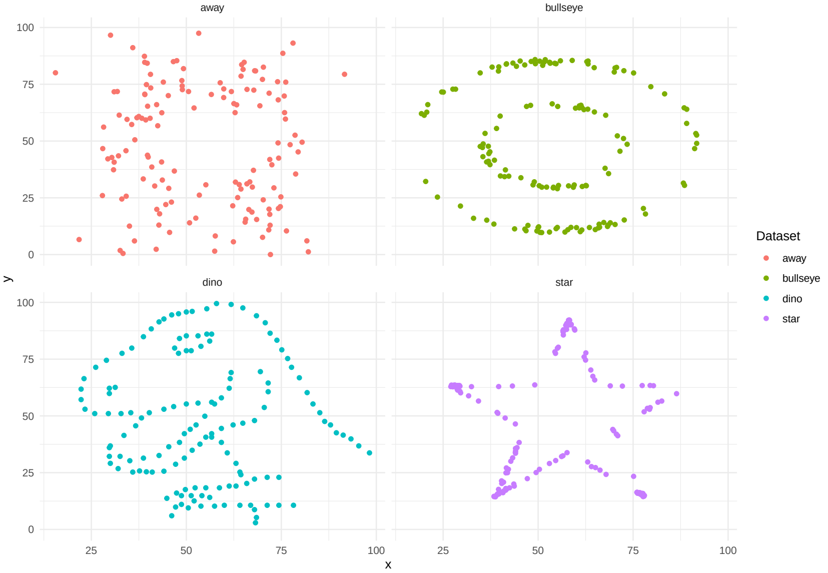
```
## # A tibble: 13 x 2
##    dataset        n
##    <chr>      <int>
##  1 away         142
##  2 bullseye     142
##  3 circle       142
##  4 dino         142
##  5 dots         142
##  6 h_lines      142
##  7 high_lines   142
##  8 slant_down   142
##  9 slant_up     142
## 10 star         142
## 11 v_lines      142
## 12 wide_lines   142
## 13 x_shape      142
```

Do some summaries for each dataset

```
datasaurus_dozen %>%
  group_by(dataset) %>%
  summarise(mean_x = mean(x),
            mean_y = mean(y),
            correlation = cor(x,y))
```

```
## # A tibble: 13 x 4
##    dataset    mean_x mean_y correlation
##    <chr>       <dbl>  <dbl>       <dbl>
##  1 away         54.3   47.8     -0.0641
##  2 bullseye     54.3   47.8     -0.0686
##  3 circle       54.3   47.8     -0.0683
##  4 dino         54.3   47.8     -0.0645
##  5 dots         54.3   47.8     -0.0603
##  6 h_lines      54.3   47.8     -0.0617
##  7 high_lines   54.3   47.8     -0.0685
##  8 slant_down   54.3   47.8     -0.0690
##  9 slant_up     54.3   47.8     -0.0686
## 10 star         54.3   47.8     -0.0630
## 11 v_lines      54.3   47.8     -0.0694
## 12 wide_lines   54.3   47.8     -0.0666
## 13 x_shape      54.3   47.8     -0.0656
```

But now let's plot


away
bullseye
dino
star
x
y
Dataset
away
bullseye
dino
star