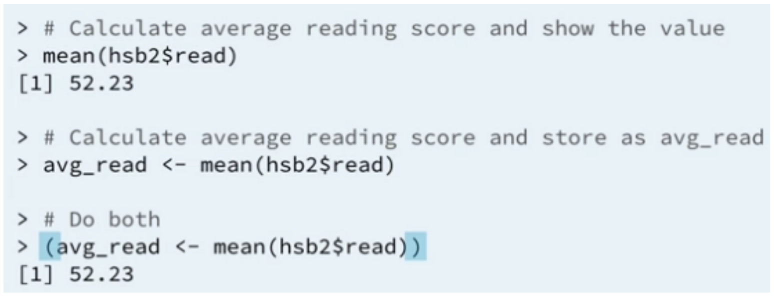
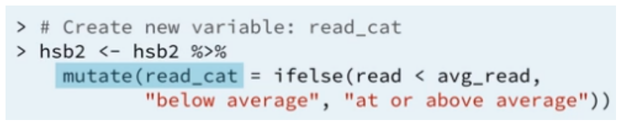
R: Intro to Data - Mine Cetinkaya-Rundel
Note as of July 2019
Note Taker: Paris Zhang

Chapters:

1. Language of data
2. Study types and cautionary tales
3. Sampling strategies and experimental design
4. Case study

Ch 1 - Language of data

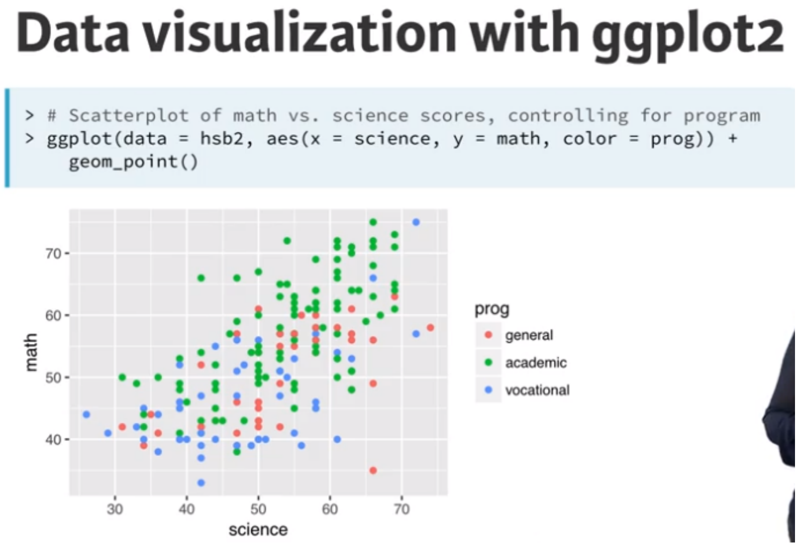
- Welcome information
  - **dplyr**
    - `str()` v.s. `glimpse()` - glimpse tries to show as many data points as fit neatly on the screen
- Type of variables
  - Numerical (quantitative)
    - Continuous - heights
    - Discrete - number of pets in household / test scores
  - Categorical (qualitative) - e.g., gender (no inherent ordering)
    - Ordinal: finite number of values within a given range, often measured - e.g., satisfaction level in a survey (ordered / inherent ordering)
- Categorical data: factors
  - `table()`
  - `filter()`
    `hsb2_public <- hsb2 %>% filter(schtyp == "public")`
    - The pipe operator: `%>%`
      `x %>% f(y)` means f(x, y)
    - `> sum(3, 4)` is the same as
      `> 3 %>% sum(4)`
  - Drop (unused) levels; factor type with 0 observation is dropped
    `hsb2_public$schtyp <- droplevels(hsb2_public$schtyp)`
- Discretize a variable
  (discretize: represent or approximate (a quantity or series) using a discrete quantity or quantities)
  - Assign and print
    
```
> # Calculate average reading score and show the value
> mean(hsb2$read)
[1] 52.23

> # Calculate average reading score and store as avg_read
> avg_read <- mean(hsb2$read)

> # Do both
> (avg_read <- mean(hsb2$read))
[1] 52.23
```
  - Add new variable



```
> # Create new variable: read_cat
> hsb2 <- hsb2 %>%
    mutate(read_cat = ifelse(read < avg_read,
              "below average", "at or above average"))
```

`ifelse`(logical test, if TRUE, if FALSE)

- Visualizing numerical data



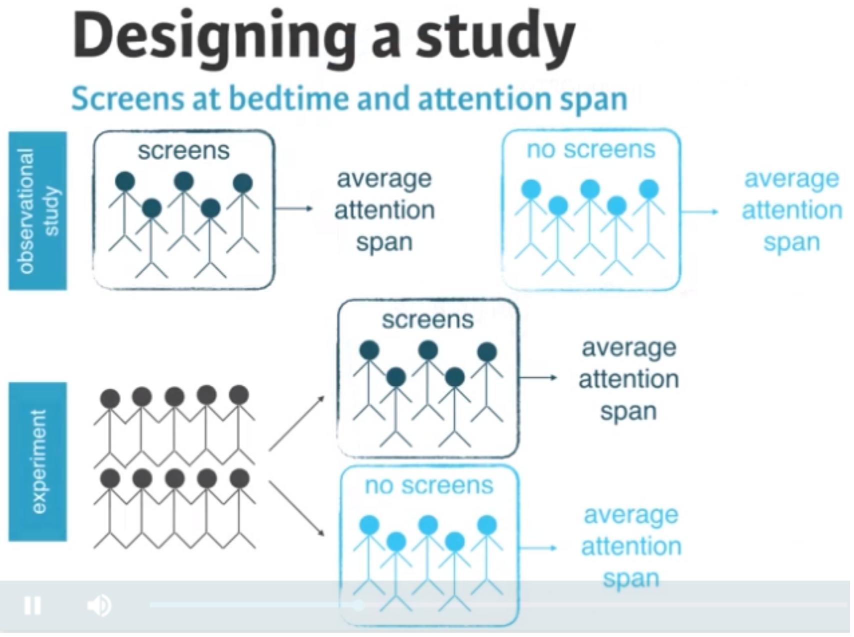
Data visualization with ggplot2

```
> # Scatterplot of math vs. science scores, controlling for program
> ggplot(data = hsb2, aes(x = science, y = math, color = prog)) +
    geom_point()
```

  - `ggplot()`

Ch 2 - Study types and cautionary tales

- Observational studies and experiments



Designing a study

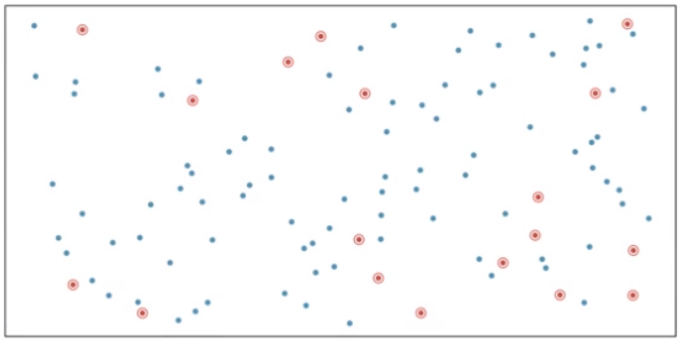
Screens at bedtime and attention span

  - In an experiment, confounding variables are controlled better, so that if difference is found, we might indicate a causal relationship
- Random sampling and random assignment
  - Ramdom sampling - randomly select from population; helps generalizing results
  - Random assignments - randomly assign subjects to various treatments; helps infer causation from results
- Simpson's paradox
  - A third variable that changes the picture (relationship between X and Y): e.g., males are more likely to be admitted to UC Berkeley, but when we **CONTROL** for department, females are more

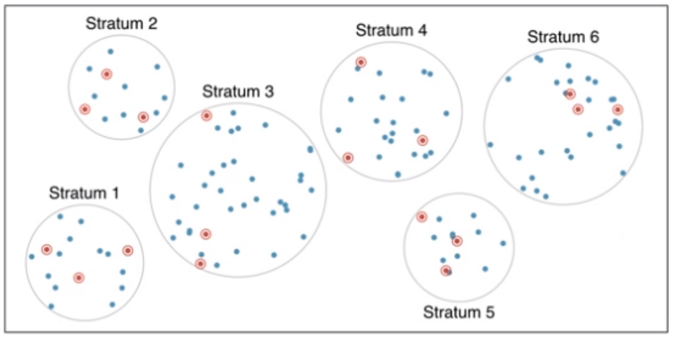
likely to be admitted.

Ch 3 - Sampling strategies and experimental design

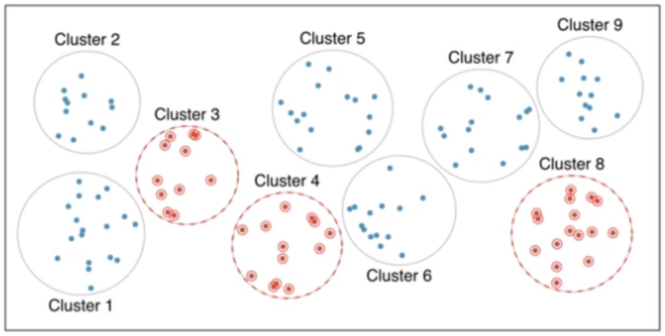
- Sampling strategies
  - Simple random sample: each element is equally likely to be sampled. (e.g., randomly drawing names from a hat)



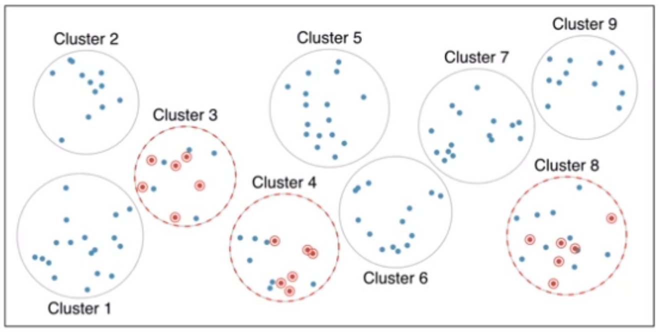
  - Stratified sample: homogenious stratum first, and then randomly sample each stratum. (e.g., low, medium, high social economical status)



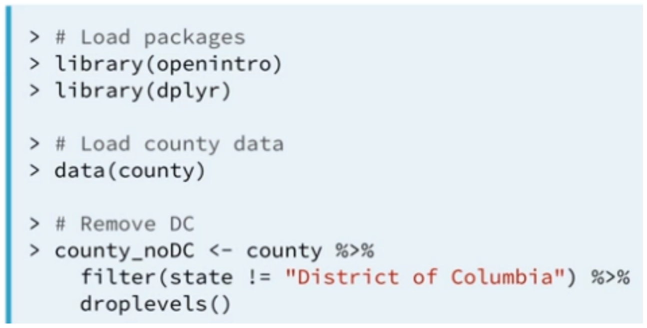
  - Cluster sample: divide into different heterogeneous clusters, randomly select a few clusters, and then randomly sample all observations from these few clusters.



  - Multistage sample: cluster, randomly select a few clusters, and then randomly select observations from those clusters. (e.g., in socialeconomic studies, divide a city into different regions, and then sample a few regions to avoid going to all regions)

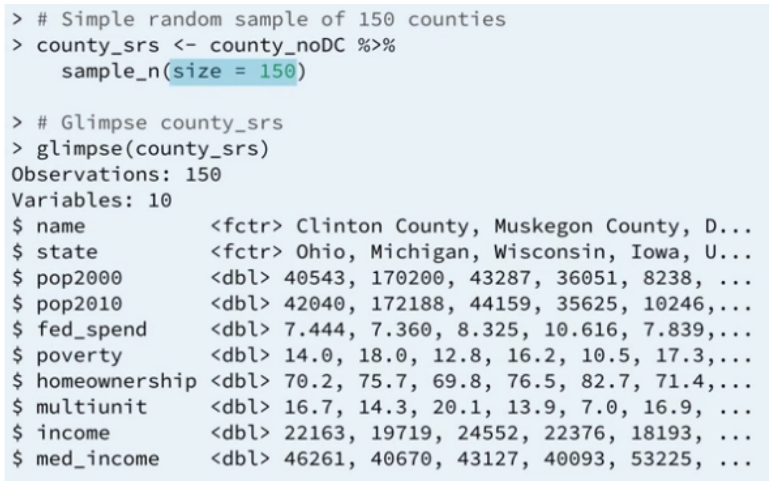


- Sampling in R
  - Setup



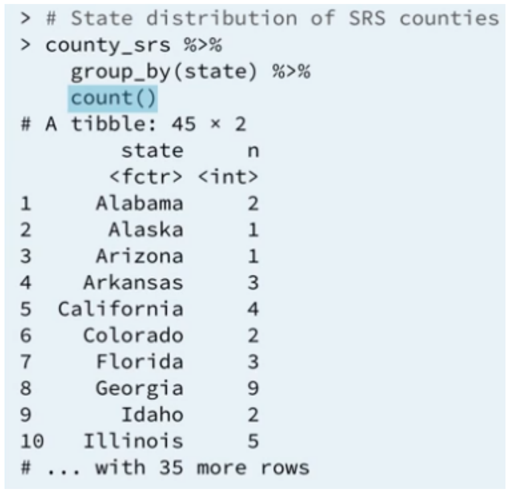
```
> # Load packages
> library(openintro)
> library(dplyr)

> # Load county data
> data(county)

> # Remove DC
> county_noDC <- county %>%
    filter(state != "District of Columbia") %>%
    droplevels()
```

  - **Simple random sample**: select only 150 out of 3,000 available counties.



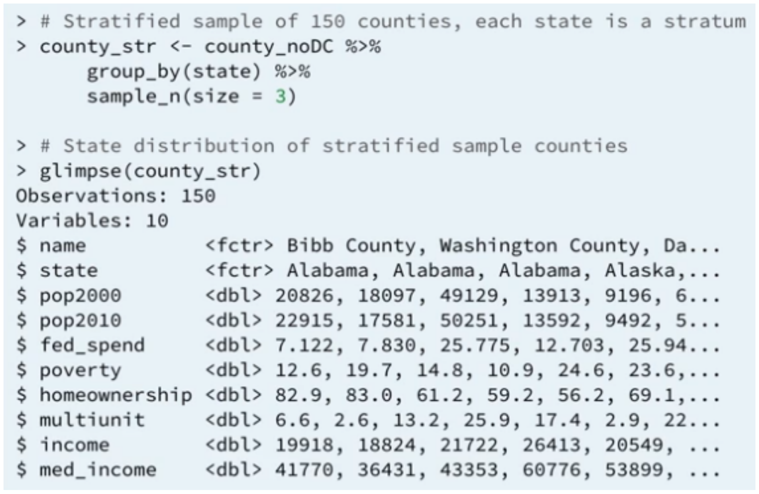
```
> # Simple random sample of 150 counties
> county_srs <- county_noDC %>%
    sample_n(size = 150)

> # Glimpse county_srs
> glimpse(county_srs)
Observations: 150
Variables: 10
$ name          <fctr> Clinton County, Muskegon County, D...
$ state         <fctr> Ohio, Michigan, Wisconsin, Iowa, U...
$ pop2000       <dbl> 40543, 170200, 43287, 36051, 8238, ...
$ pop2010       <dbl> 42040, 172188, 44159, 35625, 10246,...
$ fed_spend     <dbl> 7.444, 7.360, 8.325, 10.616, 7.839,...
$ poverty       <dbl> 14.0, 18.0, 12.8, 16.2, 10.5, 17.3,...
$ homeownership <dbl> 70.2, 75.7, 69.8, 76.5, 82.7, 71.4,...
$ multiunit     <dbl> 16.7, 14.3, 20.1, 13.9, 7.0, 16.9, ...
$ income        <dbl> 22163, 19719, 24552, 22376, 18193, ...
$ med_income    <dbl> 46261, 40670, 43127, 40093, 53225, ...
```

But it fails if we want counties to be evenly distributed in each state:



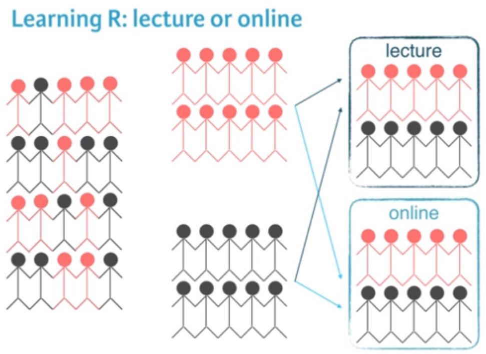
```
> # State distribution of SRS counties
> county_srs %>%
    group_by(state) %>%
    count()
# A tibble: 45 × 2
        state     n
       <fctr> <int>
1     Alabama     2
2      Alaska     1
3     Arizona     1
4    Arkansas     3
5  California     4
6    Colorado     2
7     Florida     3
8     Georgia     9
9       Idaho     2
10   Illinois     5
# ... with 35 more rows
```

  - **Stratified sample**



```
> # Stratified sample of 150 counties, each state is a stratum
> county_str <- county_noDC %>%
      group_by(state) %>%
      sample_n(size = 3)

> # State distribution of stratified sample counties
> glimpse(county_str)
Observations: 150
Variables: 10
$ name          <fctr> Bibb County, Washington County, Da...
$ state         <fctr> Alabama, Alabama, Alabama, Alaska,...
$ pop2000       <dbl> 20826, 18097, 49129, 13913, 9196, 6...
$ pop2010       <dbl> 22915, 17581, 50251, 13592, 9492, 5...
$ fed_spend     <dbl> 7.122, 7.830, 25.775, 12.703, 25.94...
$ poverty       <dbl> 12.6, 19.7, 14.8, 10.9, 24.6, 23.6,...
$ homeownership <dbl> 82.9, 83.0, 61.2, 59.2, 56.2, 69.1,...
$ multiunit     <dbl> 6.6, 2.6, 13.2, 25.9, 17.4, 2.9, 22...
$ income        <dbl> 19918, 18824, 21722, 26413, 20549, ...
$ med_income    <dbl> 41770, 36431, 43353, 60776, 53899, ...
```

- Principles of experimental design
  - Principles
    - **Control**: compare treatment of interest to a control group
    - **Randomize**: randomly assign subjects to treatments
    - **Replicate**: collect a sufficiently large sample within a study, or replicate the entire study
    - **Block**: account for the potential effect of confounding variabls
      1. Group subjects into blocks based on these variables
      2. Randomize within each block to treatment groups
  - Design a study, with blocking (Learnig R: lecture or online)

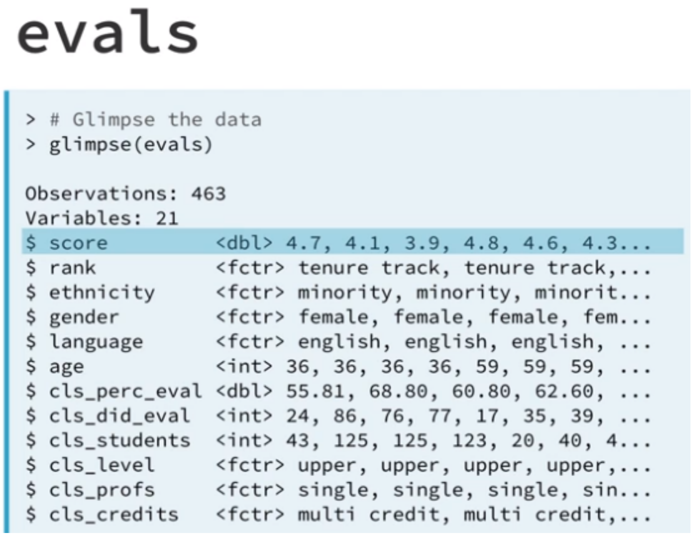


    - Goal: evaluate the effect of lecture or online course for learning R
    - Problem: some students have previous programming experience and we suspect this will affect the results
    - Blocking: Divide the students into two groups based on past programming experience (confounding variable)
    - Random sampling: randomly assign students from each block to the treatments, lecture or online (explanatory variable)
    - Outcome: to make sure that previous coding experience does not interfere with our conclusion of whether oneline course or lecture is better for learning R.
  - Example: A researcher designs a study to test the effect of light and noise levels on exam performance of students. The researcher also believes that light and noise levels might have different effects on males and females, so she wants to make sure both genders are represented equally under different conditions.
    There are 2 **explanatory variables** (light and noise), 1 **blocking variable** (gender), and 1 **response variable** (exam performance).
  - Experimental design terminology

- **Explanatory variables** are conditions you can impose on the experimental units, while **blocking variables** are characteristics that the experimental units come with that you would like to control for.
- In random sampling, we use **stratifying** to control for a variable. In random assignment, we use **blocking** to achieve the same goal.

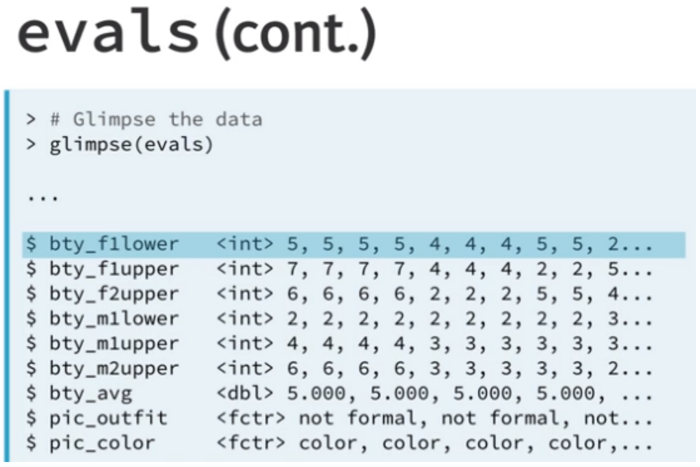
Ch 4 - Case Study

- Beauty in the classroom: to see if a professor's rating is correlated with his/her physical attractiveness.
- Variables in the data: `evals`



evals

```
> # Glimpse the data
> glimpse(evals)

Observations: 463
Variables: 21
$ score         <dbl> 4.7, 4.1, 3.9, 4.8, 4.6, 4.3...
$ rank          <fctr> tenure track, tenure track,...
$ ethnicity     <fctr> minority, minority, minorit...
$ gender        <fctr> female, female, female, fem...
$ language      <fctr> english, english, english, ...
$ age           <int> 36, 36, 36, 36, 59, 59, 59, ...
$ cls_perc_eval <dbl> 55.81, 68.80, 60.80, 62.60, ...
$ cls_did_eval  <int> 24, 86, 76, 77, 17, 35, 39, ...
$ cls_students  <int> 43, 125, 125, 123, 20, 40, 4...
$ cls_level     <fctr> upper, upper, upper, upper,...
$ cls_profs     <fctr> single, single, single, sin...
$ cls_credits   <fctr> multi credit, multi credit,...
```



evals (cont.)

```
> # Glimpse the data
> glimpse(evals)

...

$ bty_f1lower   <int> 5, 5, 5, 5, 4, 4, 4, 5, 5, 2...
$ bty_f1upper   <int> 7, 7, 7, 7, 4, 4, 4, 2, 2, 5...
$ bty_f2upper   <int> 6, 6, 6, 6, 2, 2, 2, 5, 5, 4...
$ bty_m1lower   <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 3...
$ bty_m1upper   <int> 4, 4, 4, 4, 3, 3, 3, 3, 3, 3...
$ bty_m2upper   <int> 6, 6, 6, 6, 3, 3, 3, 3, 3, 2...
$ bty_avg       <dbl> 5.000, 5.000, 5.000, 5.000, ...
$ pic_outfit    <fctr> not formal, not formal, not...
$ pic_color     <fctr> color, color, color, color,...
```