36-315: Statistical Graphics & Visualization

- course objectives
  - learn useful principles for making appropriate statistical graphics.
  - critique existing graphs and remake better ones.
  - visualize statistical analyses to facilitate communication.
  - pinpoint the statistical claims you can/cannot make from graphics.
  - write and speak publicly about statistical graphics.
  - practice tidy data manipulation in R using the tidyverse
  - practice reproducible workflows with Quarto
- grammar of graphics defined and used in ggplot2
  - see `01lec.pdf`
- goal of data visualization: show data, communicate a story
  - induce viewer to think about substance, not graphical methodology
  - make large, complex datasets more coherent
  - encourage comparison of different pieces of data
  - describe, explore, and identify relationships
  - avoid data distortion and data decoration
  - use consistent graph design
  - avoid graphs that lead to misleading conclusions!
- data types
  - quantitative
    - discrete
    - continuous
  - categorical (`factor`)
    - nominal
      - no order
      - e.g. race, species
    - ordinal
      - ordered!
      - ranking
      - DEFAULT IN R! manually define `factor` levels, or alpha. default
- area plots
  - pie chart (BAD!!!)
  - bar chart
  - stacked bar/spine chart (for variable comparison)
  - waffle charts????
  - rose diagrams (temporal or directional context can justify usage)
- something something "geom bar stat=identity" to "take y as is"
- α level CI is $\hat{x} \pm z_{1-\frac{\alpha}{2}} \cdot \mathrm{SE}(\hat{x})$
- 1d chisq test: H_A is "at least one category differs"

  ```
  chisq.test(table(penguins$species))
  ```
- CI interpretation
  - If CIs don't overlap $\longrightarrow$ significant difference
  - If CIs overlap $\longrightarrow$ a little ambiguous
  - If CIs overlap $\longrightarrow$ a lot no significant difference

- multiple testing:
  - have multiple pairwise comparisons via CI eyeballing? Type 1 error is now above 5%!!!
  - correct by inflating p values
  - Bonferroni Correction:
    - making K comparisons $\longrightarrow$ reject iff $p \leq \frac{\alpha}{K}$.
    - easy to impl and popular but inflates p the most
    - CIs: plot $(1 - \alpha)\%$ CIs $\longrightarrow$ plot $(1 - \frac{\alpha}{K})\%$ CIs
- 2d chisq test: H_A is "A, B independent"

  ```
  chisq.test(table(penguins$species, penguins$island))
  ```
  - visualize this with mosaic plots
- mosaic plots: can shade by Pearson residuals
  - more positive p.r $\longrightarrow$ more counts than expected, more neg is vice versa
  - we might reject null for the global chisq test but see all white residuals: can't reject null for individual local tests.
- 1d quant
  - boxplots: only summary stats: bad!
  - hist: see dist, bin width matters.
  - density curves: conditional dists
- estimation schools of thought:
  - parametric: assume dist, est params (eg MLE, 3623X)
  - nonparametric: make few assumptions, use whole dataset (density curves, regression lines??)
- kernel density estimation
  - place lil dist on every x_i
  - usually normal, but many exist (fuck it triangle. things can help maintain strict left right dist bounds if needed)
  - bandwidth (higher is more smooth dist): ggplot alr uses Gaussian reference rule of thumb, set to $1.06 \cdot \sigma_{\text{sample}} \cdot n^{-1/5}$.
  - adjust bandwith via `geom_density(adjust = <multiplier>)`
- Kolmogorov-Smirnov (KS) Test
  - H_A :distributions different
  - stat: largest gap
  - 1 sample: compare ECDF to theoretical distribution

    ```
    ks.test(
      x = penguins$flipper_length_mm,
      y = "pnorm",
      mean = flipper_length_mean,
      sd = flipper_length_sd
    )
    ```
  - 2 sample: compare two ECDFs

    ```
    ks.test(rap_duration, y = rock_duration) # both straight up vec[int]
    ```
- Power: prob of reject when you're supposed to (null is false) increased by:
  - sample size
  - reduce variance/error
  - increase differences/effects
  - choose right test! i.e. KS is underpowered compared to t.test/Barlett

```
t.test(sample_rap_duration, sample_pop_duration) # for mean
bartlett.test(list(sample_rap_duration, sample_pop_duration)) # for variance/
stddev
```