Quantifying prediction accuracy
  Context: Using leave-one-out (loo) cross-validation
  PRESS statistic: Prediction Residual Error Sums-of-Squares
    loo idea, quantifying overall accuracy predicting new observations
    like Sum-of-squared errors but quantifies prediction accuracy
  PRESS = $\Sigma_{obs}(Y_i - \hat{Y}_{-i})^2$
    $\hat{Y}_{-i}$ is prediction of Y_i from model fit without Y_i
    Almost always larger than SSE = $\Sigma_{obs}(Y_i - \hat{Y}_i)^2$
    Because PRESS prediction of Y_i not based on Y_i

Things we haven't covered that are of general interest
  Paired yes/no data
  Regression with count responses
  Flexible regressions (Splines, CART, Random Forests)
  Two-way factorial ANOVA
  Randomized Complete Block Designs (RCBD)   Quick introduction to these topics
    Goal is that you know some names if you want to pursue any topic
    Stat 5710 (Intro to Expt. Design) covers factorial ANOVA and RCBD in great depth

Paired yes/no data:
  Vit C study: what if conducted differently?
    Find households with 2 adults.
    Within each household, one adult gets Vit C, other gets Placebo
  Data are paired (yes/no response doesn't change that aspect of the design)
  My experience is that this pairing often gets forgotten
    Chi-square test is wrong (obs. are not independent)
    se of log odds ratio is wrong
  There are methods that explicitly account for pairing
    analog of the paired t-test for continuous responses
    one simple one is McNemar's test

Regression with count responses
  Two types:
    Fixed maximum: Binomial distribution
    unlimited maximum: Poisson distribution
  Example: Case study 22.1: African elephant matings
    Q: Do older elephants have more successful matings
    Does success of elderly elephants "slow down"?

Poisson distribution:
  Statistical model for non-negative counts
  One parameter: mean. Variance is the same as the mean
    So, samples from a Poisson distribution with larger mean have larger standard deviation
  λ = mean ≥ 0, doesn't have to be an integer (2.2 children in the average family)

Relate λ to covariates by:

$$\log \lambda_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} \cdots$$

Increasing X_{1i} by 1 multiplies λ by $\exp \beta_1$

Elephant example: plots linked to graphs page
Data exploration:
Mean # matings increases with age,
Pattern seems linear when Y = log(# matings)
SD # matings also increases with age
Q: Do older elephants have more successful matings?
Fit a Poisson regression model with linear predictor, log link:

$$\begin{aligned} \log \lambda_i &= \beta_0 + \beta_1 Age_i \\ Y_i &\sim Poisson(\lambda_i) \end{aligned}$$

Estimated coefficients (se):
Intercept: -1.582 (0.545)
Slope: 0.0687 (0.0138)
Interpretation of slope:
Mean # matings for an elephant that is 1 year older is 7.1% larger than that for the younger elephant
$\exp 0.0687 = 1.071$
Interpretation of slope - 2nd version with 10 year change:
Mean # matings for elephant approximately doubles when comparing an individual to one that is 10 years younger.
$\exp(10 \times 0.0687) = 1.988$
Q: Does success of elderly elephants "slow down"?
Write a model that allows that, e.g., quadratic

$$\begin{aligned} \log \lambda_i &= \beta_0 + \beta_1 Age_i + \beta_2 (Age_i)^2 \\ Y_i &\sim Poisson(\lambda_i) \end{aligned}$$

$\hat{\beta}_2$ = -0.00086, se = 0.002, p = 0.67
No evidence that increase in success "slows down" in elderly elephants

Comments about Poisson regression
Why is conclusion about mean instead of median?
Model is a log transformation of the mean (λ_i) not of the data values (Y_i)
Why not just log transform # matings and use SLR? Various reasons
Poisson regression accommodates 0 values
$\log 0$ undefined $\rightarrow$ problem when $Y_i = 0$
0 values are no problem so long as $\lambda > 0$, $\lambda = 0.000001$ is just fine

Regression problems: λ never = 0
ANOVA problems: Can get $\lambda = 0$ when all observations in a group are 0
Detail: different relationship between mean and sd
Detail: $\log Y_i$ is not normally distributed, e.g., values are $\log 1$ or $\log 2$

Overdispersion in models for count data:
Both Binomial and Poisson distributions: Var Y_i depends on mean Y_i i.e., π_i or λ_i
Sometimes the data are more variable than they "should" be
This is known as overdispersion
Account for it by using a more complicated distribution for the data
Fixed maximum: Beta binomial distribution instead of Binomial
Unlimited maximum: Negative binomial distribution instead of Poisson
My experience is that most ag/bio count data is overdispersed
Analysis of these data must account for overdispersion
Only exception is bird eggs/clutch, which are less variable than expected

Smoothing splines:
Goal: model relationship between Y and X without specifying the details
We've seen linear: E $Y_i = \beta_0 + \beta_1 X_i$ and quadratic E $Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2$ models
If curve needs to bend in a different way, could use higher order polynomial models
Adding more terms allows curve to "wiggle" more.
But only in the ways "allowed" by some specific polynomial
Smoothing splines let the data tell the model where/how much to bend
Model is

$$Y_i = f(X_i) + \varepsilon_i$$

where $f(X_i)$ is an arbitrary function estimated from the data
Simple model is a series of line segments joined together
bends where data bents, straight where data straight
Continuous, but not smooth (1st and 2nd derivatives not continuous)
More useful: join cubic polynomials that are continuous
with continuous 1st and 2nd derivatives
Looks smooth, most common choice for splines
Practical issue: how wiggly is the fitted curve?
A model selection issue: want to fit the data, but not overfit
A curve that wiggles a lot probably overfits the data
Common solution: borrow a model selection idea
combine Fit + penalty for wiggliness (= complexity)
What can this be used for?
Learning: understand relationship between Y and X
Interpolation: predict Y for X's inside the data
Splines do not extrapolate well (no data to estimate the curve)
Evaluate a specific model
Overlay predicted values from the model and from a spline

CART models: Classification and Regression Trees
  Really useful for multiple X variables with complicated interaction effects
  Predictor is a dichotomous key like classifying species
  Model (for Gaussian data): $Y_i = f(X_i) + \varepsilon_i$, $\varepsilon_i \sim N(0, \sigma^2)$
  Like splines: estimate $f(X)$ from data
    but prediction is the average of a group of observations
    Example from SAT data: if ltakers ≥ 3.205, predict 877.4; if not, predict 1002
    Those values are the means of the groups with ltakers ≥ 3.205 and ltakers < 3.205
  Algorithm:
    Consider all variables, and all possible split points
    Find the variable and split point that best separates two groups
      details of "best" depend on nature of the data (continuous, count, yes/no)
    Split the data, consider each subgroup separately
    Find the best split for subgroup 1, result is two sub-sub-groups
    And for subgroup 2, giving 2 more sub-sub-groups
    Keep splitting until:
      no effective split
      or groups too small to split further (user specified limit)
  Practical experience is that a tree tends to overfit the data
    "prune" the tree by removing lowest branches
    decision often made using cross-validation
  Make prediction by working down the tree
    at each split, decide which way the observation goes
    when get to end, prediction is the average for that leaf
  CART models:
    require quite a bit of data ($>$ 100 observations, $>$250 is better)
    are really effective with contingent relationships
      SAT: rank only matters when ltakers < 3.205
    not as useful when a simple model (e.g., linear) is sufficient

Random Forest:
  Extension of a CART model
  Idea is to create many trees by resampling the original data
    500 trees is common, often more.
  Prediction: have covariate values
    Use covariates to make a prediction from each model, so 500 predictions
    report their average as the prediction for that observation
  An example of "ensemble" prediction: using many models
  Seems weird, but works extremely well
  Random Forests are the best "out of the box" method to make predictions
    my experience and opinion, shared by lots of others
    "out of the box" means they work on lots of different types of problems
    without requiring a lot of problem-specific tweaking.