Part 1

Lecture 1: Linear Regression and Correlation - Key Concepts & Skills

This lecture serves as the essential foundation for the more complex models that follow. While the exam has historically focused on multilevel models, a perfect understanding of linear regression is non-negotiable, as all the concepts (coefficients, p-values, variance) are building blocks.

1. Association and Correlation

- **Core Idea:** The lecture starts by establishing that many scientific questions are about the association between variables. While correlation does not imply causation, causation does imply correlation.
- **Pearson Correlation Coefficient (r_{xy}):** This is the primary metric for quantifying linear association between two continuous variables.
  - **Formula:**
  $$r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{s_x^2 s_y^2}}$$
  - **Range:** It's always between -1 (perfect negative linear correlation) and +1 (perfect positive linear correlation). A value of 0 means no linear correlation.
  - **Critical Point:** A correlation of 0 does not mean the variables are independent. It simply means they are not linearly related. They could have a strong non-linear (e.g., quadratic) relationship. This is a classic statistics question.
- **R Implementation:**
  - `cor(x, y)` calculates the correlation coefficient.
  - `cor.test(x, y)` not only calculates the correlation but also provides a p-value and a confidence interval. The p-value tests the null hypothesis that the true population correlation is zero.

2. Simple Linear Regression (SLR)

This section uses the Kalama Study (relationship between children's age and height) as the main example.

- **The Model:**
$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$
  - **Systematic Part ($E(Y|X) = \beta_0 + \beta_1 X$):** This is the line. It represents the average value of Y for a given value of X.
  - **Random Part (ϵ_i):** The error term. It represents the variability of individual data points around the average line. This variability comes from other factors not in the model and inherent biological variability.
- **Interpretation of Coefficients:**
  - β_0 **(Intercept):** The average value of Y when X is 0. Be careful, this is often not meaningful if $X = 0$ is outside the range of your data or physically impossible (e.g., age = 0).
  - β_1 **(Slope):** The average change in Y for a one-unit increase in X.
- **Estimation (Least Squares):** The goal is to find the line that "best" fits the data. The method of least squares does this by finding the values of $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize the sum of the squared residuals ($SSE = \sum e_i^2$). The residual is the difference between the observed value and the value predicted by the line ($e_i = y_i - \hat{y}_i$).

3. Assessing the SLR Model

- **Decomposition of Variance:** The total variation in Y (Total Sum of Squares, SSTO) can be split into two parts: the variation explained by the model (Regression Sum of Squares, SSR) and the unexplained variation (Error Sum of Squares, SSE).
  -
  $$\text{SSTO} = \text{SSR} + \text{SSE}$$

- **Coefficient of Determination (R^2):**
  - **Formula:**
  $$R^2 = \frac{\text{SSR}}{\text{SSTO}}$$
  - **Interpretation:** It represents the proportion of the total variation in the response variable (Y) that is explained by the model. An R^2 of 0.68 means the model explains 68% of the variability in Y.
  - In Simple Linear Regression, there's a direct relationship: $|r_{xy}| = \sqrt{R^2}$.
- **R Implementation (`lm` function):**
  - You fit the model using `res <- lm(height   age, data = kalama)`.
  - The `summary(res)` command gives the coefficient table.
  - The `anova(res)` command gives the ANOVA table.

4. Multiple Linear Regression (MLR)

This is where the concepts become more powerful and directly relevant to the exam's complexity. This section uses the Patient Satisfaction study.

- **The Model:**
  $$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \epsilon$$
  The line becomes a hyperplane.
- **Key Interpretation of Coefficients:** β_k is the change in the average response Y for a one-unit increase in predictor X_k, while *holding all other predictor variables constant.*
- **Flexibility of MLR:** The model is "linear" because it's linear in the parameters (β's), not necessarily in the variables (X's). This allows for great flexibility:
  - **Categorical Predictors:** A categorical variable with k levels is included by creating k-1 dummy variables.
  - **Interaction Terms:** You can include terms like $\beta_3(X_1 \cdot X_2)$. This means the effect of X_1 on Y depends on the value of X_2. Graphically, this results in non-parallel lines.
  - **Polynomial Regression:** You can include terms like $\beta_2 X_1^2$. This is still a linear model and can capture curved relationships.

CRITICAL EXAM SKILL: `summary()` vs. `anova()` in R

The lecture dedicates significant time to this because it's a common point of confusion. You are very likely to be tested on this, either by interpreting output or explaining the difference. Imagine a model: `lm(satis   age + severity + anxiety)`

- `summary()` **output (The Coefficient Table)**
  - **What it does:** It provides a p-value for each coefficient (e.g., for severity) based on a t-test.
  - **Hypothesis Tested:** The p-value for severity tests whether severity adds significant explanatory power *given that age and anxiety are already in the model.* It compares the full model (age+severity+anxiety) to a model without that specific term (age+anxiety).
  - **Key Feature:** The results do not depend on the order of the variables in the formula.
- `anova()` **output (The ANOVA Table)**
  - **What it does:** It performs a sequential analysis, testing variables one by one in the order they appear in the formula.
  - **Hypothesis Tested:**
    - The p-value for `age` (first term) compares a model with age to a model with only the intercept (`1`).
    - The p-value for `severity` (second term) compares a model with age+severity to a model with just age.
    - The p-value for `anxiety` (third term) compares age+severity+anxiety to age+severity.
  - **Key Feature:** The p-values absolutely *depend on the order of variables in the formula.* Changing the order (`age+anxiety+severity`) will change the p-values for anxiety and severity.

Function	**Test Type**	**Hypothesis for variable X_k**	**Order Dependent?**
`summary()`	Type III SS / Marginal	Does X_k add value, given all other variables are in the model?	No
`anova()`	Type I SS / Sequential	Does X_k add value, given the variables listed *before it* are in the model?	Yes

5. Confidence vs. Prediction Intervals

This is another crucial distinction for the practical part of the exam. After fitting a final model (e.g., `satis age + anxiety`), you might be asked to predict for a new patient.

- **Confidence Interval:**
  - **Question:** "What is the *average* satisfaction for a *group* of patients who are 43 years old with an anxiety level of 2.7?"
  - **Interpretation:** It's an interval for the mean response ($E(Y|X_h)$). It only accounts for the uncertainty in the estimated coefficients ($\hat{\beta}$'s).
  - **R:** `predict(model, newdata, interval = "confidence")`.
- **Prediction Interval:**
  - **Question:** "What is the satisfaction for *one specific, new* patient who is 43 years old with an anxiety level of 2.7?"
  - **Interpretation:** It's an interval for a single observation ($Y_{h,new}$). It must account for the uncertainty in the coefficients *plus* the inherent, unpredictable random error (ϵ).
  - **R:** `predict(model, newdata, interval = "prediction")`.

Key Takeaway: The prediction interval is *always wider* than the confidence interval for the same data point, because it accounts for an additional source of uncertainty.

Lecture 2: Logistic Regression & Model Selection - Key Concepts & Skills

This lecture builds directly on the first one, extending the concept of regression to a new type of outcome variable. The second half introduces a formal framework for choosing between different models, a critical skill for the exam.

Part A: Logistic Regression

The lecture uses the Donner Party dataset (survival outcome) to illustrate why linear regression fails for binary outcomes and how logistic regression solves the problem.

1. The Problem with Linear Regression for Binary Outcomes

- **Core Issue:** If the outcome Y is binary (0 or 1), its average value, $E(Y|X)$, is the probability of the event occurring, $P(Y = 1|X)$.
- A standard linear model, $P(Y = 1|X) = \beta_0 + \beta_1 X$, is problematic because the right side of the equation can produce values outside the valid probability range of $[0, 1]$. For instance, for a very old person, the predicted probability of survival could become negative.

2. The Solution: The Logistic Model

Instead of modeling the probability directly, we model a transformation of it.

- **The Logit Transformation:** The chosen transformation is the logit, which is the natural logarithm of the odds.
  - **Odds:** The ratio of the probability of an event happening to the probability of it not happening.

    $$\text{Odds} = \frac{P(Y = 1|X)}{P(Y = 0|X)} = \frac{p}{1 - p}$$

  - **Logit:**

    $$\text{logit}(p) = \ln\left(\frac{p}{1 - p}\right)$$

    This function takes a probability p (from 0 to 1) and maps it to the entire real number line (from $-\infty$ to $+\infty$).
- **The Logistic Regression Model:** We set the linear combination of predictors equal to the logit of the probability.

  $$\text{logit}[P(Y = 1|X)] = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

- **Back-transformation:** To get the probability back, we use the inverse logit function, which gives the characteristic "S"-shaped curve.

  $$P(Y = 1|X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}$$

  This formula guarantees that the predicted probability will always be between 0 and 1.

3. Interpretation of Coefficients using Odds Ratios (OR)

This is the most critical interpretation skill for logistic regression. A coefficient β_k represents the change in the log-odds for a one-unit change in predictor X_k. To make this intuitive, we exponentiate the coefficient.

- **Key Relationship:**

  $$e^{\beta_k} = \text{Odds Ratio (OR)}$$

- **Interpretation:** For a one-unit increase in the predictor X_k, the odds of the outcome (Y=1) are multiplied by a factor of e^{β_k}.
  - If $\beta_k > 0$, then OR > 1: The predictor increases the odds of the outcome.
  - If $\beta_k < 0$, then OR < 1: The predictor decreases the odds of the outcome.
  - If $\beta_k = 0$, then OR $= 1$: The predictor has no effect on the odds (independence).

- **Example from Donner Party Data:** The model is $\text{logit}[\hat{P}(Y = 1|\text{age}, \text{fem})] = 0.553 - 0.035 \cdot \text{age} + 1.067 \cdot \text{fem}$.
    - **Gender (fem):** The coefficient is 1.067. The OR is $e^{1.067} \approx 3$. *Interpretation:* At any given age, the odds of survival for a woman are 3 times the odds of survival for a man.
    - **Age:** The coefficient is -0.0356. The OR for a one-year increase is $e^{-0.0356} \approx 0.96$. *Interpretation:* Each additional year of age is associated with a 4% decrease in the odds of survival (i.e., the odds are multiplied by 0.96).
    - **For a c-unit change** (e.g., 10 years of age): The OR is $e^{\beta \cdot c}$. For age, this is $e^{-0.0356 \cdot 10} \approx 0.70$. *Interpretation:* A 10-year increase in age is associated with a 30% decrease in the odds of survival.

4. Estimation and R Implementation

- **Estimation Method:** The model is fit using Maximum Likelihood Estimation (MLE), not Ordinary Least Squares (OLS). MLE finds the parameter values that make the observed data most probable.
- **R Implementation:**
    - **Fit the model:** Use `glm()` (generalized linear model). You must specify `family = binomial(link = "logit")`.
      `donner.log <- glm(Outcome   Age + fem, data=donner.na, family=binomial(link="logit"))`
    - **Get Coefficients & Tests:** `summary(donner.log)` gives the coefficients (β), standard errors, and p-values for the null hypothesis that each β is zero.
    - **Get Odds Ratios & CIs:** You must exponentiate the coefficients and their confidence intervals.
      `exp(coef(donner.log))` # Odds Ratios
      `exp(confint(donner.log))` # Confidence Intervals for the ORs

Part B: Model Building and Selection

The lecture emphasizes that in practice, we rarely know the ”true” model and must choose from a set of plausible candidates.

1. The Challenge of Model Selection

- **Goal:** Find a model that fits the data well without unnecessary complexity. This is the principle of parsimony or Occam's Razor.
- **Problems with Automated Procedures:** The lecture cautions against relying solely on automatic procedures like stepwise selection, as they can have statistical issues (e.g., incorrect p-values, instability).

2. Akaike Information Criterion (AIC)

AIC is presented as a powerful tool for comparing a set of candidate models.

- **Formula:**

$$\text{AIC} = -2 \cdot \log(\mathcal{L}_{\max}) + 2 \cdot (\#\text{ of parameters})$$

    - $-2 \cdot \log(\mathcal{L}_{\max})$: A measure of model fit. Lower values mean better fit.
    - $2 \cdot (\#\text{ of parameters})$: A penalty for complexity. More parameters lead to a higher AIC.
- **Interpretation:**
    - Smaller is better.
    - A single AIC value is meaningless; it's only useful for comparing models in a candidate set.
    - The model with the lowest AIC is considered the ”best” in the sense that it is the model that is ”closest” to the true underlying data generating mechanism.

3. Akaike Weights (w_i)

These are derived from the AIC values and are easier to interpret.

- **Calculation:**
  1. Find the model with the minimum AIC (AIC_{min}) in your set.
  2. For each model i, calculate the difference: $\Delta_i = \text{AIC}_i - \text{AIC}_{\text{min}}$.
  3. The weight for model i is calculated as:

$$w_i = \frac{\exp(-\frac{1}{2}\Delta_i)}{\sum_{r=1}^{R} \exp(-\frac{1}{2}\Delta_r)}$$

- **Interpretation:** The Akaike weight, w_i, can be interpreted as the probability that model i is the best model in the set, given the data. All weights in the set sum to 1.

4. Model Uncertainty & Model Averaging

- **Model Selection Uncertainty:** The Akaike weights often show that there isn't one single, clear "winner". For instance, in the Donner Party example, the main effects model has a weight of 0.56, but the interaction model still has a substantial weight of 0.25. This means there is considerable uncertainty as to which model is truly "best".
- **Model Averaging:** Instead of picking one model and ignoring this uncertainty, model averaging bases its conclusions on a weighted average of the most plausible models. The lecture introduces this as a concept, where the final estimate of a parameter, $\tilde{\beta}_i$, is the weighted average of the estimates from each model, using the Akaike weights.

Lecture 3: Multilevel Models for Longitudinal Data - Key Concepts & Skills

This lecture moves beyond standard regression to handle a common and complex data type: longitudinal data, where individuals are measured repeatedly over time.

1. Why We Need Multilevel Models for Longitudinal Data

- **The Problem of Non-Independence:** When you measure the same person multiple times, those measurements are correlated. For instance, a child's IQ at age 1 is a strong predictor of their IQ at age 1.5. Standard linear regression assumes all observations are independent, making it unsuitable for this type of data.
- **The Solution:** Multilevel models (also called hierarchical models, mixed-effects models, or random coefficient models) are specifically designed to handle this nested data structure (measurements nested within individuals).

2. The Two Levels of Change

The core idea of multilevel modeling is to analyze variation at two different levels simultaneously:

- **Level 1: Within-Individual Change:** This level models how each individual changes over time. We essentially fit a separate growth curve for each person.
- **Level 2: Between-Individual Change:** This level models why the growth curves (i.e., the intercepts and slopes) from Level 1 are different from person to person. We use person-level characteristics (like which experimental group they are in) to explain this variation.

3. Deconstructing the Hierarchical Model

The lecture uses the "Early Dietary Intervention" study to build the model step-by-step. Understanding this formulation is the key to interpreting the R output on the exam.

Level 1 Model: The Individual Growth Model This model describes the trajectory for a single person, i.

$$Y_{ij} = \pi_{0i} + \pi_{1i}(\text{Age}_{ij} - 1) + \epsilon_{ij}$$

- Y_{ij}: The cognitive score (cog) for child i at measurement occasion j.
- π_{0i}: The individual intercept for child i. This is their unique, true cognitive score at the centering point (Age = 1 year).
- π_{1i}: The individual slope for child i. This is their personal rate of change (increase/decrease) in cognitive score per year.
- ϵ_{ij}: The within-individual residual. This is the deviation of child i's observed score at occasion j from their own true trajectory. It represents measurement error and has a variance of σ^2_ϵ.

Level 2 Model: Explaining Between-Individual Differences Now we model the individual intercepts and slopes from Level 1 using a person-level predictor, PROG (1 for intervention, 0 for control).

$$\pi_{0i} = \gamma_{00} + \gamma_{01}\text{PROG}_i + b_{0i}$$
$$\pi_{1i} = \gamma_{10} + \gamma_{11}\text{PROG}_i + b_{1i}$$

- **Fixed Effects ($\gamma_{00}, \gamma_{01}, \gamma_{10}, \gamma_{11}$):** These are the population average parameters. They are "fixed" because they are the same for everyone.
  - γ_{00}: The average intercept for the control group (PROG=0).
  - γ_{01}: The average difference in intercepts between the intervention group and the control group.
  - γ_{10}: The average slope (rate of change) for the control group.
  - γ_{11}: The average difference in slopes between the intervention and control groups. This is often the main parameter of interest, as it tests if the intervention changed the rate of development.
- **Random Effects (b_{0i}, b_{1i}):** These are the individual-specific deviations from the average.
  - b_{0i}: How much child i's intercept deviates from their group's average intercept.
  - b_{1i}: How much child i's slope deviates from their group's average slope.

These random effects are assumed to follow a normal distribution with a mean of 0.

The Combined ”Mixed-Effects” Model

By substituting the Level 2 equations into the Level 1 equation, we get a single, comprehensive model:

$$Y_{ij} = \underbrace{[\gamma_{00} + \gamma_{01}\text{PROG}_i + \gamma_{10}(\text{Age}_{ij} - 1) + \gamma_{11}\text{PROG}_i(\text{Age}_{ij} - 1)]}_{\text{Fixed Effects}} + \underbrace{[b_{0i} + b_{1i}(\text{Age}_{ij} - 1) + \epsilon_{ij}]}_{\text{Random Part}}$$

4. Interpreting Variance Components

The model estimates several variances, which are crucial for understanding the sources of variability.

- σ_ϵ^2 (Residual Variance): The within-individual variance. How much individuals' scores fluctuate around their own trajectory over time.
- σ_0^2 (Random Intercept Variance): The between-individual variance in initial status. How much the starting points (π_{0i}) vary from person to person.
- σ_1^2 (Random Slope Variance): The between-individual variance in the rate of change. How much the slopes (π_{1i}) vary from person to person.
- σ_{01} (Covariance of Intercept and Slope): Measures the association between initial status and the rate of change. A negative covariance (as seen in the example) means that individuals who start with a higher IQ tend to have a more negative (or less positive) slope.

5. Estimation (ML vs. REML) & R Implementation

- **Estimation Methods:**
  - **Maximum Likelihood (ML):** A general method for finding the parameter estimates that make the observed data most likely. However, its estimates for variance components can be biased in small samples.
  - **Restricted Maximum Likelihood (REML):** A modification of ML that produces unbiased estimates of the variance components. It is generally the preferred method.
- **R Implementation (`lme4` package):**
  - The model is fit using the `lmer()` function.
  - The formula syntax is critical: `cog   1 + age0*program + (1 + age0 | id)`
    - `cog   1 + age0*program`: This part specifies the fixed effects. It models `cog` as a function of an intercept, `age0`, `program`, and their interaction (`*` includes both main effects and the interaction).
    - `(1 + age0 | id)`: This part specifies the random effects. It tells R to fit a random intercept (`1`) and a random slope for `age0` for each level of the grouping factor `id`, and to estimate the correlation between them.

Key Takeaways for the Exam

- **Know the Model Inside-Out:** Be able to identify and interpret every parameter from the R output ($\gamma_{00}, \gamma_{01}, \gamma_{10}, \gamma_{11}, \sigma_0^2, \sigma_1^2, \sigma_\epsilon^2, \sigma_{01}$).
- **Understand the Levels:** Be able to explain the difference between Level 1 (within-subject) and Level 2 (between-subject) variance.
- **Hypothesis Testing:** Know what hypothesis is being tested by each fixed effect. For example, testing $H_0 : \gamma_{11} = 0$ is testing whether the effect of the program on the rate of change is significant.
- **ML vs. REML:** This is a key theoretical point.
  - REML is generally preferred for the final model's variance components because it is unbiased.
  - If you are comparing two models with different fixed effects using a likelihood ratio test (`anova()` command), you MUST refit the models using Maximum Likelihood (`REML = FALSE`).
- **Interpreting `lmer` Output:** The exam will likely provide you with the `summary()` output of an `lmer` model. You must be able to extract all the parameter estimates and use them to answer questions about the study's conclusions. For instance, you should be able to calculate the average IQ for a specific group at a specific time point using the fixed-effect estimates.

Lecture 4: Multilevel Models for Clustered Data - Key Concepts & Skills

The focus shifts from data collected over time to data that is naturally grouped or "clustered."

1. Clustered Data vs. Longitudinal Data

- **Core Concept:** The underlying problem is the same: non-independence of observations.
  - In longitudinal data, measurements are nested within individuals.
  - In clustered data, individuals are nested within larger groups (e.g., students in classrooms, patients in hospitals, or in this lecture's example, pups in litters).
- **The Rat Pup Example:** The lecture uses a study where the birth weights (`weight`) of rat pups are measured. Pups are clustered within litters (`litterid`), and each litter belongs to a mother who received a specific treatment (Control, Low, or High dose). The birth weights of pups from the same litter are expected to be correlated.

2. The Hierarchical Model for Clustered Data

The two-level model structure is directly analogous to the one for longitudinal data, but the interpretation of the levels changes.

Level 1: Within-Cluster Variation

This model describes the variation among pups within the same litter.

$$w_{ij} = \pi_{0i} + \pi_{1i}\text{sex}_{ij} + \epsilon_{ij}$$

- w_{ij}: The birth weight of pup j from litter i.
- π_{0i}: The cluster-specific intercept for litter i. This represents the average weight of the reference group (male pups, where sex=0) in that specific litter.
- π_{1i}: The cluster-specific slope for litter i. This is the average difference in weight between female and male pups in that specific litter.
- ϵ_{ij}: The within-cluster residual for pup j. It represents the variability among pups of the same sex within the same litter. Its variance is σ^2_ϵ.

Level 2: Between-Cluster Variation

This model explains why the litter-specific intercepts (π_{0i}) and slopes (π_{1i}) vary across different litters. This variation is modeled using cluster-level predictors like treatment and `litsize`.

$$\begin{aligned}\pi_{0i} &= \gamma_{00} + \gamma_{01}\text{treat1}_i + \gamma_{02}\text{treat2}_i + \gamma_{03}\text{ls}_i + b_{0i}\\ \pi_{1i} &= \gamma_{10} + \ldots\end{aligned}$$

- **Fixed Effects (γ):** These describe the population-average relationships. For example, γ_{01} would be the average difference in pup weight between the 'High' treatment group and the 'Control' group, holding other factors constant.
- **Random Effects (b_{0i}):** The random intercept for litter i. It represents how the average weight of pups in litter i deviates from the overall average of its treatment group, after accounting for litter size. Its variance, σ^2_b, is the between-litter variance.

3. Modeling Heterogeneous Variances (Key New Concept)

A standard mixed model assumes that the residual variance (σ^2_ϵ) is the same across all groups (homoscedasticity).

- **The Problem:** The exploratory data analysis of the rat pup data suggests this assumption is false. The variability (spread) of birth weights appears smaller in the High and Low dose groups compared to the Control group.
- **The Solution:** We can fit a more flexible heteroscedastic model that allows the residual variance to be different for different treatment groups.
- **R Implementation (`nlme` package):** The lecture notes that `lmer()` is not ideal for this, so it uses the `lme()` function from the `nlme` package.

  - **Homoscedastic Model:** `lme(weight   ..., random =  1 | litterid, ...)`
  - **Heteroscedastic Model:** An additional `weights` argument is used.
    `lme(..., weights = varIdent(form =  1 | treatment))`

- **Model Comparison:** To formally test if the variances are different, you can compare the two models (homoscedastic vs. heteroscedastic) using a Likelihood Ratio Test (`anova()` in R). Because the models have the same fixed effects, this comparison can and should be done using models fit with REML (the default in `lme`).

4. Testing for the Significance of Random Effects

A crucial question is whether the clustering even matters. Is there significant variability between litters?

- **The Hypothesis:** To test this, we test if the variance of the random effect is zero. For the random intercept model, this is $H_0 : \sigma_b^2 = 0$.

- **The "Boundary" Problem:** Standard likelihood ratio tests assume the null value is not on the boundary of the parameter space. Since variance cannot be negative, $\sigma_b^2 = 0$ is on the boundary.

- **The Consequence:** The usual χ^2 distribution for the test statistic is incorrect. The true null distribution is a 50:50 mixture of a χ_0^2 (a point mass at 0) and a χ_1^2 distribution.

Key Takeaways for the Exam

- **Recognize the Data Structure:** Be able to identify if a problem involves longitudinal or clustered data. The key is understanding the source of non-independence (repeated measures vs. natural groupings).

- **Apply the Two-Level Framework:** Even if the data isn't longitudinal, you should be able to define the Level 1 (within-cluster) and Level 2 (between-cluster) models and interpret the coefficients and variance components accordingly.

- **Check for Heteroscedasticity:** A potential practical question could involve checking whether the residual variance is constant across groups and knowing how to fit a model that accounts for this using the `weights` argument in `lme()`.

- **Know How to Test Variance Components:**

  - Remember that testing if a random effect is needed involves testing if its variance is zero ($H_0 : \sigma_b^2 = 0$).
  - Be aware of the boundary problem: know that the standard p-value from an LRT is conservative (too large) by a factor of up to 2. To get a more accurate test, you can divide the p-value from the standard χ_1^2 test by two.

- **Distinguish `lme4` and `nlme`:** For the exam, know that `lmer()` (from `lme4`) is the go-to for standard mixed models, but if you need to model different residual variances (heteroscedasticity), `lme()` (from `nlme`) is the tool demonstrated in the lecture.

Lecture 5: Missing Data - Key Concepts & Skills

This lecture explains why missing data is a serious problem, classifies the different reasons why data might be missing, and presents modern, valid methods to handle it.

1. The Problem: "The Unknown Unknowns"

- **Ubiquitous in Science:** Missing data is an extremely common problem that occurs for many reasons, such as lost questionnaires, equipment failure, or subjects declining to answer certain questions.
- **The Danger of Defaults:** Standard software packages like R often handle missing data by default using Complete Case (CC) analysis (also called listwise deletion). This means any subject with even a single missing value is completely removed from the analysis. As the lecture demonstrates, this often leads to biased and incorrect conclusions.

2. Missing Data Mechanisms: Why Are the Data Missing?

The validity of any method depends on the underlying reason for the missingness. There are three formal classifications.

- **Missing Completely at Random (MCAR)**
  - **Definition:** The probability of a value being missing is completely random and does not depend on any observed or unobserved data. Think of a researcher accidentally dropping a tray of test tubes.
  - **Implication:** This is the most benign scenario, but often unrealistic. Under MCAR, a Complete Case analysis gives unbiased (but inefficient) estimates.
- **Missing at Random (MAR)**
  - **Definition:** The probability of a value being missing depends only on other observed data, not on the missing value itself.
  - **Example:** In the Titanic dataset, the probability that a passenger's age is missing might depend on their class or survived status (both of which are fully observed). We saw that people in 3rd class and those who died were more likely to have a missing age.
  - **Implication:** This is a more realistic assumption than MCAR. Complete Case analysis is biased under MAR. However, principled methods like Multiple Imputation and IPW can provide unbiased results.
- **Missing Not at Random (MNAR)**
  - **Definition:** The probability of a value being missing depends on the missing value itself (or other unobserved variables).
  - **Example:** If people with very high incomes are the most likely to refuse to report their income, the missingness in the income variable depends on the value of income itself.
  - **Implication:** This is the most difficult scenario. There are no simple fixes, and even advanced methods require making strong, untestable assumptions about the missing data mechanism.

3. Principled Methods to Handle Missing Data (Under MAR)

The lecture focuses on two main approaches that are valid under the MAR assumption.

A) Multiple Imputation (MI)

- **Core Idea:** Instead of throwing data away (like CC) or guessing a single value (like mean substitution), MI creates multiple (e.g., $m = 5, 10, 100$) plausible versions of the complete dataset. This process explicitly accounts for the uncertainty about the true missing values.
- **The Three Steps of MI:**
  1. **Impute:** An "imputation model" is built using the observed data to predict the missing values. This model is then used to "fill in" the missing holes m times, creating m different completed datasets. Each imputation is a random draw from a predictive distribution, reflecting the uncertainty.

2. **Analyze:** The desired statistical analysis (e.g., a logistic regression) is performed separately on each of the m completed datasets. This results in m different sets of parameter estimates and standard errors.
3. **Pool:** The m results are combined into a single, final set of estimates and standard errors using specific formulas called Rubin's Rules. This pooling step correctly incorporates both the regular sampling variance and the extra variance that comes from the uncertainty of the imputation.

- **R Implementation:** This is done using the `mice` package.

```
# 1. Impute the data 100 times
imp <- mice(titanic.missing, m=100)

# 2. Fit the analysis model to each of the 100 datasets
fit <- with(data=imp, exp=glm(survived ~ pclass + sex + age,
           family=binomial))

# 3. Pool the results using Rubin's Rules
est <- pool(fit)
summary(est)
```

B) Inverse Probability Weighting (IPW)

- **Core Idea:** IPW is a clever form of weighted complete case analysis. It gives more weight to the observed individuals who are ”similar” to the missing individuals, thereby correcting for the bias introduced by throwing them away.
- **The Two Steps of IPW:**
  1. **Model the Missingness & Calculate Weights:**
     - Create a new variable r that is 1 if a case is complete and 0 otherwise.
     - Fit a logistic regression to model the probability of being complete, using variables that are fully observed: e.g., `r   class + survived`.
     - The weight for each complete case is the inverse of its predicted probability of being complete:

$$w_i = \frac{1}{\hat{P}(r_i = 1)}$$

  2. **Weighted Analysis:** Perform the standard analysis (e.g., logistic regression) on the complete cases only, but use the calculated weights in the `weights` argument of the `glm()` function.

Key Takeaways for the Exam

- **Never Use Simple Methods Blindly:** Methods like Listwise Deletion (the default in R), Mean Imputation, or Last Observation Carried Forward are generally flawed and will likely lead to incorrect conclusions, especially if the data are not MCAR.
- **The Mechanism is Key:** The first step is to think about why the data might be missing. Is it MCAR, MAR, or MNAR? Your choice of method depends entirely on this assumption.
- **MI and IPW are the Go-To Solutions for MAR:** These two methods are the main tools presented for providing valid statistical inference under the more realistic MAR assumption.
- **Know How MI and IPW Work:**
  - **MI:** Understand the ”Impute, Analyze, Pool” process and the idea that it accounts for uncertainty by creating multiple datasets.
  - **IPW:** Understand the concept of up-weighting complete cases that are representative of the missing cases.
- **Be Ready for Practical Application:** You may be given a dataset with missing values and asked to perform one of these analyses in R, likely using the `mice` package for MI or by manually calculating weights for IPW.

Part 2

This section will switch focus to Part 2, taught by Professor Jelier.

Lecture 1: Bias-Variance Trade-off, Cross-Validation & Bootstrap

This lecture establishes the fundamental challenge in all statistical modeling—finding a model that is both accurate and generalizable—and introduces the primary tools used to navigate this challenge.

1. The Philosophy of Statistical Modeling

- **Science and Falsifiability:** The lecture begins by framing statistical modeling within the philosophy of science. A scientific statement or model must be falsifiable—that is, it must make predictions that can be tested against real-world data. Statistical tests are the tools we use to decide if a hypothesis is supported by the data.
- **A General Approach:** A typical modeling workflow involves:
  - Surveying the data: Understanding variable types and distributions.
  - Choosing a model: Selecting a method with an appropriate level of flexibility and interpretability.
  - Fitting parameters: Using methods like Maximum Likelihood or Least Squares.
  - Checking the model: Assessing the fit and residuals.

2. The Bias-Variance Trade-off: The Core Concept

This is the most critical concept for this part of the course. It addresses the challenge of building a model that captures the underlying patterns in the training data (low bias) without being overly sensitive to its specific noise (low variance). The total expected error of a model can be decomposed into three parts:

$$\text{Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

- **Bias:**
  - **Definition:** Bias is the error introduced by approximating a very complex real-world problem with a simpler model. A model with high bias pays very little attention to the training data and tends to underfit.
  - **Example:** Using a simple linear model to describe a complex, non-linear relationship.
- **Variance:**
  - **Definition:** Variance refers to the amount by which our model fit ($\hat{f}$) would change if we estimated it on a different training dataset. A model with high variance is overly sensitive to the training data and tends to overfit.
  - **Example:** A highly flexible, "wiggly" model that fits the noise in the training data perfectly but fails to generalize to new data.
- **The Trade-off:**
  - Increasing model flexibility or complexity (e.g., adding more predictors or polynomial terms) will decrease bias but increase variance.
  - The goal is to find the optimal level of complexity that minimizes the total error, as shown in the classic U-shaped curve.

3. The Curse of Dimensionality

Modern biological datasets are often "high-dimensional," meaning the number of predictors (p) is large, sometimes even larger than the number of observations (n). This poses specific challenges:

- It becomes very easy to find models that perfectly fit the training data but fail to generalize (overfitting).
- In high-dimensional space, observations are sparse, making it difficult to find reliable local patterns. The volume of the space grows exponentially, requiring an exponentially larger amount of data to maintain the same data density.

4. Resampling Methods: Tools for Model Assessment

To estimate a model's true performance on unseen data (the test error) and find the optimal complexity, we use resampling methods.

- **Validation Set Approach:**
  - **Method:** Randomly split the data into a training set (to build the model) and a validation/test set (to evaluate it).
  - **Drawbacks:** The test error estimate can be highly variable depending on the random split. The model is trained on a smaller dataset, which can cause it to perform worse and thus overestimate the test error (this is a source of bias).
- **Leave-One-Out Cross-Validation (LOOCV):**
  - **Method:** For a dataset of size n, you perform n iterations. In each iteration i, you train the model on all data points except for point i, and then test the model on point i. The final LOOCV error is the average of these n individual errors.
  - **Advantages:** It has much less bias than the validation set approach because the training sets are of size $n-1$, which is almost the full dataset. The error estimate is also more stable.
  - **Disadvantage:** It can be extremely computationally expensive, as it requires fitting the model n times.
- **k-Fold Cross-Validation:**
  - **Method:** A compromise between the two extremes. The data is randomly split into k groups (or "folds"), typically 5 or 10. You then perform k iterations. In each iteration, one fold is held out as the test set, and the model is trained on the other $k-1$ folds. The final error is the average of the k test errors.
  - **Bias-Variance Trade-off for CV:** LOOCV has low bias but can have high variance because the n training sets are almost identical to each other and thus the outputs are highly correlated. k-fold CV has slightly more bias but lower variance. A k of 5 or 10 is empirically shown to be a good trade-off.
- **The Bootstrap:**
  - **Method:** A powerful resampling technique where one generates new datasets by repeatedly sampling with replacement from the original dataset.
  - **Primary Use:** The lecture emphasizes its use for estimating the uncertainty (standard error) of a parameter estimate. By fitting a model on many bootstrap samples and collecting the parameter estimates each time, the standard deviation of these collected estimates approximates the standard error of the original estimate. This is extremely useful when the assumptions for standard statistical formulas break down.

Lecture 2: High Dimensionality & Subset Selection

This lecture addresses the challenges of building models when the number of predictors is large and introduces methods to select a smaller subset of valuable predictors.

1. The Challenge of High Dimensionality ($p \gg n$)

- **Problem Statement:** Many modern datasets, especially in bioinformatics, are high-dimensional, meaning the number of features or predictors (p) is large, often much larger than the number of samples (n).
- **Consequences:**
  - **Overfitting:** When p is large, it becomes easy to find models that perfectly fit the noise in the training data but fail to generalize to new, unseen data. The variance of the model becomes too high.
  - **Noise and Spurious Correlations:** With many predictors, there's a high chance that some are correlated with the response purely by chance, leading to incorrect conclusions.
  - **Computational Cost:** Fitting models with a vast number of predictors can be computationally intensive or, in some cases, impossible.
  - **Interpretability:** Models with hundreds of predictors are difficult or impossible to interpret.

2. Subset Selection: Finding the Best Predictors

To address these challenges, we can use subset selection methods to identify a smaller set of predictors that are truly related to the response. The lecture covers three main approaches to find the best model for each possible subset size k.

A) Best Subset Selection

- **How it Works:** This is the most thorough approach. It considers every possible combination of predictors for each subset size k, from $k = 1$ to $k = p$.
  - For $k = 1$, fit all p models with a single predictor. The best one is chosen based on the highest R^2 or lowest Residual Sum of Squares (RSS).
  - For $k = 2$, fit all $\binom{p}{k}$ models with two predictors and choose the best one.
  - Continue this process up to the full model with p predictors.
- **Result:** This process yields the best possible model for each subset size $(M_1, M_2, \ldots, M_p)$.
- **Major Drawback:** This method is computationally exhaustive. For p predictors, there are 2^p possible models to evaluate, making it infeasible for even moderate p (e.g., $p = 40$ is computationally impossible).

B) Forward Stepwise Selection

- **How it Works:** This is a "greedy," computationally efficient alternative. It starts with a model containing no predictors and adds them one at a time.
  - Start with the null model (intercept only).
  - Fit p simple linear regressions and add the single variable that results in the lowest RSS.
  - Add the next variable that gives the greatest additional improvement (lowest RSS) when added to the current model.
  - Continue until a stopping rule is reached or all p predictors are in the model.
- **Advantage:** Much faster than best subset, as it only fits $1 + \sum_{k=0}^{p-1}(p - k)$ models instead of 2^p.
- **Disadvantage:** It's a greedy approach and is not guaranteed to find the true best model for each subset size. An initial variable might seem important, but become redundant when other variables are added later.

C) Backward Stepwise Selection

- **How it Works:** The opposite of forward selection. It starts with the full model (all p predictors) and removes the least useful predictor one at a time.
  - Start with the full model containing all p predictors.
  - Remove the variable that results in the smallest increase in RSS.
  - Continue removing variables one by one until a stopping rule is reached.
- **Advantage:** Like forward selection, it's computationally efficient.
- **Disadvantage:** It's also a greedy approach. It also has the requirement that the number of samples n must be larger than the number of predictors p to be able to fit the initial full model.

3. Choosing the Optimal Model Size k

After running one of the selection methods above, you will have a set of "best" models, one for each size k. The final crucial step is to select the single best model from this set.

- **The Problem:** You cannot use metrics like training RSS or R^2 to choose among models of different sizes. These metrics will always improve as you add more predictors, which would always lead you to select the full model and overfit the data.
- **The Solution:** You need a method that estimates the test error. The lecture presents two ways to do this.

Approach 1: Adjusting Training Error with a Penalty These methods add a penalty to the training error for model complexity. A model is chosen that optimizes this adjusted metric. The goal is to select the model with the lowest value (except for Adjusted R^2, which should be maximized).

- **Mallow's C_p:**
  $$C_p = \frac{1}{n}(\text{RSS} + 2k\hat{\sigma}^2)$$
  This is an estimate of the test MSE.
- **AIC (Akaike Information Criterion):** Proportional to C_p for linear models. It is a more general criterion that can be used for other models as well.
- **BIC (Bayesian Information Criterion):**
  $$\text{BIC} \propto \text{RSS} + \log(n)k\hat{\sigma}^2$$
  BIC uses a larger penalty term than AIC when n is large. It is more stringent and tends to select smaller, more parsimonious models than AIC/C_p.
- **Adjusted R^2:**
  $$R^2_{\text{adj}} = 1 - \frac{\text{RSS}/(n-k-1)}{\text{TSS}/(n-1)}$$
  This metric adjusts R^2 downwards for having more predictors. The goal is to maximize Adjusted R^2.

Approach 2: Directly Estimating Test Error with Cross-Validation

- **Method:** This is the most direct and generally preferred method. You use k-fold cross-validation (as discussed in Lecture 1) to get a direct estimate of the test error for each model size k.
- **Process:**
  1. Divide the data into k folds.
  2. For each fold i:
     - Perform the entire subset selection procedure (e.g., forward stepwise) on the other $k-1$ folds.
     - For each model size k, calculate the test error on the held-out fold i.
  3. Average the test errors across all k folds for each model size.
  4. Select the model size k that has the lowest average cross-validated error.
- **The "One-Standard-Error" Rule:** Often, a plot of CV error vs. model complexity shows a flat "valley." The one-standard-error rule suggests choosing the simplest model whose CV error is within one standard error of the absolute best model. This favors parsimony when multiple models have similar performance.

Lecture 3: Penalized Regression & Dimensionality Reduction

This lecture presents two families of methods for building models in high-dimensional settings: Shrinkage Methods (which penalize large coefficients) and Dimensionality Reduction Methods (which create new, fewer predictors).

1. Shrinkage (Regularization) Methods

Instead of explicitly selecting a subset of predictors, shrinkage methods fit a model containing all p predictors. However, they constrain or regularize the coefficient estimates, shrinking them towards zero. This reduces model variance and can significantly improve prediction accuracy.

A) Ridge Regression

- **Core Idea:** Ridge regression modifies the standard least squares loss function by adding a penalty term that is proportional to the sum of the squares of the coefficients.
- **Objective Function:** Minimize
$$\text{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$
The term $\lambda \sum \beta_j^2$ is called the L2 penalty.
- **The Tuning Parameter (λ):** λ controls the strength of the penalty. It is always non-negative.
  - When $\lambda = 0$, the penalty has no effect, and Ridge regression produces the standard least squares estimates.
  - As $\lambda \to \infty$, the penalty becomes more influential, and the coefficients are shrunk closer and closer to zero.
- **Key Property:** Ridge regression will always keep all p predictors in the model. It shrinks the coefficients towards zero but will never set any of them exactly to zero (unless $\lambda = \infty$). Therefore, Ridge is good for improving prediction accuracy but does not perform automatic variable selection.
- **Standardization:** It is essential to standardize the predictors (scale them to have a standard deviation of one) before fitting a Ridge model. This ensures that the penalty is applied fairly, as the scale of the coefficients depends on the scale of the predictors themselves.
- **Choosing λ:** The optimal value of the tuning parameter λ is determined using cross-validation to find the value that yields the lowest test error.

B) The LASSO (Least Absolute Shrinkage and Selection Operator)

- **Core Idea:** The LASSO is a popular alternative to Ridge that overcomes its main disadvantage. It uses a different penalty that can force coefficients to be exactly zero.
- **Objective Function:** Minimize
$$\text{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|$$
The term $\lambda \sum |\beta_j|$ is called the L1 penalty.
- **Key Property:** Due to the nature of the L1 penalty, as λ increases, it has the effect of forcing some coefficient estimates to be exactly equal to zero.
- **Advantage:** Because it can produce zero coefficients, the LASSO performs automatic variable selection, yielding a ”sparse” model that is often easier to interpret.
- **Choosing λ:** Similar to Ridge, the optimal value for λ is found using cross-validation.

2. Dimensionality Reduction Methods

This approach reduces the problem of high dimensionality by transforming the original predictors into a smaller set of new, uncorrelated variables, and then using these new variables to fit a model.

Feature	Ridge Regression	The LASSO
Penalty	L2 Norm ($\sum \beta_j^2$)	L1 Norm ($\sum \lvert\beta_j\rvert$)
Coefficients	Shrinks towards zero, but never exactly zero.	Can be shrunk to be exactly zero.
Variable Selection	No. All p predictors remain in the model.	Yes. Produces a sparse model.
Best Use Case	When most predictors are truly related to the response.	When a relatively small subset of predictors are truly related to the response.

A) Principal Components Regression (PCR)

- **How it Works:**
  1. **Derive Principal Components:** First, perform Principal Component Analysis (PCA) on the p predictors to create a new set of variables, $Z_1, Z_2, \ldots, Z_p$. These "principal components" are linear combinations of the original predictors and are all uncorrelated with each other.
  2. **Reduce Dimensions:** Select the first M components (where $M < p$).
  3. **Fit Model:** Use these M principal components as predictors in a standard least squares linear regression model.
- **The Tuning Parameter (M):** The number of components to use, M, is a tuning parameter that is selected via cross-validation.
- **Key Drawback:** The principal components are identified in an *unsupervised* manner. This means they are derived by only looking at the predictors X (to explain their variance) and completely ignoring the response variable Y. There is no guarantee that the components that explain the most variance in the predictors are the ones that are most predictive of the response.

B) Partial Least Squares (PLS)

- **How it Works:** PLS is a supervised alternative to PCR. Like PCR, it creates a smaller set of new components to use as predictors. However, it creates these components in a *supervised* way.
- **Supervised Dimension Reduction:** When constructing the PLS components ($Z_1, \ldots, Z_M$), the algorithm gives higher weight to the original predictors that are most strongly related to the response variable Y.
- **Advantage:** PLS attempts to find components that explain both the variance in the predictors and the relationship with the response. Because it uses information from Y, it often leads to models that perform better than PCR.

Key Takeaways for the Exam

- **Ridge vs. LASSO:** Understand the fundamental difference between the L2 (Ridge) and L1 (LASSO) penalties. The key exam point is that LASSO performs variable selection, while Ridge does not.
- **The Role of λ:** For both Ridge and LASSO, you must be able to explain that λ is a tuning parameter that controls the amount of shrinkage and is chosen via cross-validation.
- **PCR vs. PLS:** The critical distinction is that PCR is *unsupervised* (it ignores the response Y when creating components), while PLS is *supervised* (it uses Y to create the components). This often makes PLS a better choice for prediction.
- **High-Dimensionality Context:** These four methods are presented as solutions to the problems that arise when the number of predictors p is large. They are computationally superior and often more accurate than the stepwise selection methods from the previous lecture.

Lecture 4: Beyond Linearity

The standard linear model assumes a simple, additive, and linear relationship. This lecture explores methods to relax the "linearity" assumption, allowing for more flexible and accurate models.

1. The Need for Non-Linear Models

Often, the true relationship between a predictor and the response is not a straight line. Forcing a linear fit in such cases can lead to high bias and poor predictive performance. The methods in this lecture provide ways to automatically find and fit non-linear relationships.

2. Polynomial Regression

- **How it Works:** This is the simplest extension of a linear model. Instead of just fitting $y = \beta_0 + \beta_1 x$, you add higher-order polynomial terms:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d + \epsilon_i$$

- **Advantages:**
  - It's still a linear model in its coefficients, so it can be fit easily using standard least squares.
  - It can model curved relationships.
- **Disadvantages:**
  - **Global Nature:** The fit in one region of the predictor's range can be heavily influenced by data points far away.
  - **Erratic Fits:** Polynomials often become erratic and produce wild fits at the boundaries (tails) of the data.
- **Choosing the Degree d:** The degree d is a tuning parameter. If d is too low, the model underfits. If d is too high, it overfits and becomes excessively "wiggly." The optimal degree is typically chosen using cross-validation.

3. Regression Splines (A More Local and Flexible Approach)

Instead of fitting one high-degree polynomial over the entire range of X, splines fit separate low-degree polynomials over different regions of X.

- **Knots:** The points where the different polynomial pieces connect are called knots.
- **Piecewise Polynomials:** The general idea of fitting separate functions in different regions. A simple piecewise cubic spline with K knots results in fitting $K + 1$ different cubic polynomials.
- **The Smoothness Constraint:** To avoid a "jumpy" and disconnected fit, we enforce constraints at each knot. For a smooth cubic spline, we require that the function, its first derivative, and its second derivative are all continuous at each knot. This ensures a smooth curve.
- **Choosing the Knots:**
  - The number of knots (K) determines the flexibility of the spline. More knots lead to a more flexible, higher-variance fit.
  - The location of the knots is also important. A common and effective strategy is to place them at uniform quantiles (e.g., percentiles) of the observed predictor data.
  - The optimal number of knots is chosen using cross-validation.
- **Natural Splines:** This is a special type of regression spline that is constrained to be linear at the boundaries (i.e., in the regions beyond the first and last knot). This helps to reduce the high variance and erratic behavior that can occur at the tails of the data range.

4. Smoothing Splines (An Alternative to Choosing Knots)

- **Core Idea:** Instead of the user choosing the number and location of knots, a smoothing spline finds a function $g(x)$ that fits the data well but is also smooth. It does this by minimizing a modified loss function:

$$\text{Minimize: RSS} + \lambda \int g''(t)^2 dt$$

  - **RSS:** The standard residual sum of squares, which encourages the function to fit the data.
  - $\lambda \int g''(t)^2 dt$**:** A roughness penalty. The second derivative, $g''(t)$, measures the "wiggliness" of the function. This penalty term punishes functions that are too rough or wiggly.

- **The Tuning Parameter (λ):** This parameter controls the bias-variance trade-off.
  - If $\lambda = 0$, there is no penalty, and $g(x)$ will be a very wiggly function that perfectly interpolates the data points (high variance).
  - As $\lambda \to \infty$, the penalty dominates, forcing $g(x)$ to become a simple straight line (high bias).

- **Choosing λ:** The effective degrees of freedom of the spline are a function of λ. The optimal value for λ is typically found efficiently using Leave-One-Out Cross-Validation (LOOCV).

5. Generalized Additive Models (GAMs)

GAMs provide a general framework for extending these non-linear methods to models with multiple predictors.

- **The Model:** A GAM assumes an additive structure but allows for non-linear functions for each predictor.

$$y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i$$

- **The Functions (f_j):** Each f_j is a function that captures the relationship between predictor x_j and the response. These are typically fit using splines (e.g., smoothing splines).

- **Advantages:**
  - **Flexibility:** Can model complex, non-linear relationships for each predictor individually.
  - **Interpretability:** Because the model is additive, we can examine the effect of each predictor on the response by plotting its individual function f_j. This allows us to see if the relationship is linear, U-shaped, etc.
  - **Avoids the Curse of Dimensionality:** By not considering complex interactions between predictors, GAMs can be effectively fit even with a moderate number of predictors.

- **GAMs vs. GLMs:** A Generalized Linear Model (GLM) assumes a linear relationship: $\beta_1 x_1 + \beta_2 x_2 + \ldots$. A GAM replaces each linear term $\beta_j x_j$ with a flexible, non-linear function $f_j(x_j)$.

Key Takeaways for the Exam

- **Splines over Polynomials:** Understand that splines are generally superior to high-degree polynomials because their effect is more local, which provides more stability and avoids erratic fits at the boundaries.

- **The Role of Cross-Validation:** CV is the primary tool used to select the tuning parameters that control the flexibility of these models (e.g., the number of knots for a regression spline, or the smoothing parameter λ for a smoothing spline).

- **GAMs are the Workhorse:** For the practical exam, GAMs are a powerful and likely tool. Be prepared to fit a GAM and, most importantly, interpret the plots of the component functions (f_j) to determine if there is evidence for a non-linear relationship for any given predictor.

- **GAMs and Interpretability:** The key benefit of a GAM is the balance it strikes. It is more flexible than a standard linear model but more interpretable than a complex, fully non-parametric "black box" model like a random forest or boosting.

Lecture 5: Tree-Based Methods

This lecture explores an alternative to the linear and additive models discussed previously. Tree-based methods work by partitioning the predictor space into distinct regions to make predictions.

1. The Basics of Decision Trees

- **Core Idea:** Decision trees mimic human decision-making by creating a set of sequential, hierarchical rules. The model segments the predictor space into a number of simple, non-overlapping regions.
- **Interpretation:** Single decision trees are very easy to interpret and visualize. They are often called "white-box" models because their prediction logic is transparent.
- **Building a Regression Tree:**
  - **Partitioning:** The goal is to divide the predictor space into distinct regions $(R_1, R_2, \ldots, R_J)$ that minimize the Residual Sum of Squares (RSS).
  - **Prediction:** For any new observation that falls into a region R_j, the prediction is simply the mean of the training observations in that region.
  - **Recursive Binary Splitting:** Since considering all possible partitions is computationally impossible, trees are built using a "greedy" algorithm called recursive binary splitting.
    - It starts at the top and finds the single best predictor and cutpoint that splits the data into two regions and leads to the greatest reduction in RSS.
    - This process is then repeated on each of the two new regions, and so on, until a stopping criterion is met.
- **Pruning a Tree:** A tree that is grown too deep will overfit the training data. To counter this, a technique called cost complexity pruning is used.
  - A very large tree, T_0, is grown.
  - A sequence of best subtrees is considered, indexed by a non-negative tuning parameter, α. For each value of α, there is a corresponding subtree that minimizes a penalized RSS.
  - The optimal subtree (i.e., the best value of α) is chosen using k-fold cross-validation.
- **Classification Trees:** The process is very similar, but instead of minimizing RSS, the goal is to create splits that result in nodes that are as "pure" as possible (i.e., containing observations from a single class). Purity is measured using metrics like the Gini Index or Entropy.

2. Ensemble Methods: The Power of Many Trees

While single decision trees are interpretable, they often suffer from high variance and are not very accurate. Ensemble methods improve accuracy and stability by combining the results of many trees.

A) Bagging (Bootstrap Aggregating)

- **Core Idea:** Bagging reduces the variance of a statistical learning method by averaging predictions from many models fit on different bootstrap samples.
- **How it Works:**
  1. Generate B different training datasets by sampling with replacement from the original dataset (bootstrapping).
  2. Grow a full, unpruned decision tree on each of the B bootstrapped datasets.
  3. To make a prediction for a new observation, average the predictions from all B trees.
- **Effect:** This averaging process dramatically reduces the variance of the final model without increasing the bias, leading to a substantial improvement in prediction accuracy over a single tree.

B) Random Forests

- **Core Idea:** A powerful enhancement of bagging that improves performance by de-correlating the trees.
- **The Problem with Bagging:** If there is one very strong predictor in the dataset, most of the bagged trees will use it as the top split, causing the trees to be highly correlated. Averaging correlated trees does not lead to as large a reduction in variance.
- **How it Works:** Random forests are built just like bagged trees, with one key difference:
  - At each split in the tree-building process, a random subset of m predictors is chosen as split candidates. Typically, $m \approx \sqrt{p}$.
- **Effect:** By forcing each split to consider only a random subset of predictors, this method prevents a few strong predictors from dominating all the trees. The resulting trees are less correlated with each other, and averaging them achieves a greater reduction in variance.

C) Boosting

- **Core Idea:** A completely different ensemble strategy where trees are grown sequentially, with each new tree helping to improve the performance of the previously grown trees.
- **How it Works:**
  1. Fit a small decision tree to the data (more specifically, to the residuals).
  2. The model is updated by adding a shrunken version of this new tree.
  3. The residuals are recalculated based on the current model.
  4. A new tree is fit to these new residuals. This process is repeated for a specified number of iterations, B.
- **Effect:** Boosting is a slow-learning approach that sequentially improves the model in areas where it does not perform well. It is a powerful method that primarily reduces bias.
- **Key Tuning Parameters:**
  - B **(Number of Trees):** Unlike bagging/random forests, boosting can overfit if B is too large. B is chosen using cross-validation.
  - λ **(Shrinkage/Learning Rate):** A small positive number (e.g., 0.01) that controls how slowly the model learns.
  - d **(Interaction Depth):** The maximum depth of each tree. $d = 1$ results in an additive model, while larger values allow for interactions.

Key Takeaways for the Exam

- **Trees vs. Ensembles:** Understand that single decision trees are highly interpretable but often inaccurate and high-variance. Ensembles (Bagging, Random Forests, Boosting) improve accuracy by combining many trees but at the cost of interpretability.
- **Bagging vs. Random Forests:** This is a classic comparison. Both are variance-reduction techniques. Random Forests improves on Bagging by adding another layer of randomness (selecting a random subset of predictors at each split) to de-correlate the trees, which generally leads to better performance.
- **Boosting is Different:** Recognize that Boosting builds trees sequentially to reduce bias, whereas Bagging/Random Forests build them independently in parallel to reduce variance. Boosting is powerful but has more tuning parameters and can overfit.
- **Tuning Parameters:** Know the key tuning parameters for each method and that they are almost always selected using cross-validation:
  - **Pruned Tree:** The complexity parameter (α).
  - **Random Forest:** The number of predictors to consider at each split (m).
  - **Boosting:** The number of trees (B), the learning rate (λ), and the interaction depth (d).