Stat 217: Exam 2 Notecard

ANCOVA

Analysis of Covariance (ANCOVA) is a statistical technique that combines aspects of both ANOVA and regression. It is particularly useful when comparing treatment effects while accounting for the influence of a continuous numerical variable (covariate) that cannot be controlled in the experimental design.

Data Structure: For the simplest ANCOVA with t treatments, we have n_i observations from the i^{th} treatment as pairs (Y_{ij}, X_{ij}), where:

- $j = 1, \ldots, n_i$ (observations within treatment)
- $i = 1, \ldots, t$ (treatments)

Full Model (Unequal Slopes):

$$Y_{ij} = \mu + \tau_i + \delta_i X_{ij} + \epsilon_{ij}$$

Where:

- μ: Overall constant (average Y-intercept across all regression lines)
- τ_i: Adjustment to Y-intercept for the i^{th} treatment's regression line
- δ_i: Slope of the i^{th} treatment's regression line
- X_{ij}: Covariate (measured without error)
- ϵ_{ij}: Independent normally distributed errors with mean 0 and variance σ^2

Equal Slopes Model:

$$Y_{ij} = \mu + \tau_i + \delta X_{ij} + \epsilon_{ij}$$

Key difference: The slope δ is constant across all treatments.

Analysis Procedure:

1. First fit the full (unequal slopes) model
2. Test for equality of slopes ($H_0 : \delta_1 = \delta_2 = \cdots = \delta_t$)
3. If the test is insignificant (slopes can be considered equal), fit the equal slopes model and proceed with comparison of treatment means

Interpretation (ANCOVA):

- The covariate X accounts for variability in Y that would otherwise be attributed to error
- Treatment effects (τ_i) are adjusted for the covariate's effect
- When slopes are equal, treatment comparisons are made at a common value of X

Principal Component Analysis

1. **Normalize Data:** Standardize each variable:

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}$$

where x_{ij} is the value of the j-th variable for the i-th observation, $\bar{x}_j$ is the mean, and s_j is the standard deviation of variable j.

When to Normalize: Normalization is necessary when variables are measured on different scales, as PCA is sensitive to scale differences. If all variables are already on the same scale, normalization may not be required.

2. **Compute Correlation/Covariance Matrix:** Define Σ as the covariance matrix:

$$\Sigma = \frac{1}{n-1} X^T X$$

where X is the standardized data matrix with n observations and p variables. If variables are on different scales, use the correlation matrix instead.

3. **Compute Eigenvalues & Eigenvectors:** Solve

$$(\Sigma - \lambda I)v = 0$$

where λ_k are the eigenvalues representing the variance explained by each principal component (PC), and v_k are the corresponding eigenvectors defining the new basis.

4. **Determine Number of Significant Components:** Choose m such that the cumulative variance explained exceeds a threshold (e.g., 95

$$\frac{\sum_{k=1}^{m} \lambda_k}{\sum_{k=1}^{p} \lambda_k} \geq 0.95$$

Scree plots visualize eigenvalues in descending order, and an "elbow" in the plot indicates the optimal m. The Kaiser criterion suggests retaining components where $\lambda_k > 1$.

Selecting too few components may lose important information, while too many may include noise. Cross-validation can help confirm the optimal number.

5. **Calculate PC Scores:** Project data onto principal components:

$$Z = XV$$

where Z is the matrix of principal component scores, V is the matrix of eigenvectors (principal component loadings), and X is the standardized data matrix. The first m columns of Z serve as reduced-dimension representations.

The transformed data in Z captures the most variance with fewer dimensions, useful for visualization and further modeling. Each row in Z represents an observation in the new principal component space.

Partial Least Squares Regression (PLSR) Procedure

1. **Initialization**:
   - Standardize predictors $E_0 = (X - \bar{X})/S_X$ and responses $F_0 = (Y - \bar{Y})/S_Y$
   - Initialize Y-score $\mathbf{u}_0$ as first column of F_0
2. **Component Extraction** (repeat for $h = 1, ..., k$):
   (a) X-weights: $\mathbf{w}_h = E_{h-1}^T \mathbf{u}_{h-1} / \|E_{h-1}^T \mathbf{u}_{h-1}\|$ *(Maximizes covariance with Y)*
   (b) X-scores: $\mathbf{t}_h = E_{h-1} \mathbf{w}_h$
   (c) Y-weights: $\mathbf{c}_h = F_{h-1}^T \mathbf{t}_h / \|\mathbf{t}_h\|^2$ *(Regression weights for prediction)*
   (d) Y-scores: $\mathbf{u}_h = F_{h-1} \mathbf{c}_h$
   (e) Check convergence: $\|\mathbf{t}_h^{new} - \mathbf{t}_h^{old}\| < \epsilon$
3. **Store Parameters**:
   - X-loadings: $\mathbf{p}_h = E_{h-1}^T \mathbf{t}_h / \|\mathbf{t}_h\|^2$
   - Regression coefficient: $b_h = \mathbf{u}_h^T \mathbf{t}_h / \|\mathbf{t}_h\|^2$
4. **Deflation**:
   - $E_h = E_{h-1} - \mathbf{t}_h \mathbf{p}_h^T$
   - $F_h = F_{h-1} - b_h \mathbf{t}_h \mathbf{c}_h^T$
5. **Prediction Equations**:
   - Multivariate: $\hat{Y} = T B_k C^T$ where $T = XW(P^T W)^{-1}$
   - Univariate: $\hat{y} = XW_k(P_k^T W_k)^{-1} \mathbf{b}_k$

Key Features(PLS):

- *Components*: Choose k via cross-validation (minimize MSE)
- *VIP Scores*:
  $\text{VIP}_j = \sqrt{p \sum_{h=1}^{k} (b_h^2 w_{hj}^2 \|\mathbf{t}_h\|^2) / \sum_{h=1}^{k} b_h^2 \|\mathbf{t}_h\|^2}$
  (Variables with VIP > 1 are predictive)
- *Advantages*:
  - Handles multicollinearity and high-dimensional data ($p \gg n$)
  - Focuses on Y-relevant X-variance
  - More parsimonious than PCR for prediction

Key Properties

- PLS maximizes $\text{cov}(X\mathbf{w}, Y\mathbf{c})$ (covariance between components)
- Handles multicollinearity better than OLS
- Useful when $p \gg n$ (more predictors than observations)
- Components are orthogonal (uncorrelated)

Bass Diffusion Model

The Bass Diffusion Model describes the adoption process of new products in a market, analogous to epidemiological models of disease spread. It has wide applications across retail services, industrial technology, agriculture, education, pharmaceuticals, and consumer durable goods markets.

Model Foundation: The model is based on a hazard function representing the probability of adoption at time t given it hasn't occurred yet:

$$\frac{f(t)}{1-F(t)} = p + qF(t)$$

where:

- $f(t)$ is the density function of time to adoption
- $F(t)$ is the cumulative fraction of adopters at time t

Differential Equation Form: The adoption process is described by:

$$\frac{dN(t)}{dt} = p[m - N(t)] + \frac{q}{m}N(t)[m - N(t)]$$

with initial condition $N(0) = 0$, where:

- $N(t)$ = cumulative number of adopters at time t
- $m > 0$ = total market potential (saturation point)
- $p > 0$ = coefficient of innovation (external influence)
- $q \geq 0$ = coefficient of imitation (internal influence)

Solution: The closed-form solution to the differential equation is:

$$N(t; m, p, q) = m\left(\frac{1 - e^{-(p+q)t}}{1 + \frac{q}{p}e^{-(p+q)t}}\right)$$

Key Parameters:

- **m**: Market size parameter determining the scale of demand
- **p**: Innovation coefficient representing adoption due to external influences (e.g., advertising)
- **q**: Imitation coefficient representing adoption through word-of-mouth and social contagion

Interpretation (BASS):

- The first term $p[m - N(t)]$ represents adoptions by innovators
- The second term $\frac{q}{m}N(t)[m - N(t)]$ represents adoptions by imitators
- Products with high p (e.g., 0.5) adopt quickly initially, even with low q
- Products with low p (e.g., 0.0001) start slowly but may accelerate with high q
- The relative values of p and q determine the shape of the adoption curve

Behavioral Implications:

- When $q > p$, the adoption curve has an S-shape characteristic
- When $p > q$, the curve resembles exponential decay
- The model captures both external and internal influences on adoption

James-Stein Estimator

The James-Stein estimator is a method in statistical decision theory that improves upon traditional estimators, particularly when dealing with multiple parameters. Shrinkage estimation reduces variance by pulling individual estimates toward the mean, which leads to lower overall mean squared error compared to traditional estimation methods. In the context of baseball, James-Stein estimation is applied to batting averages, demonstrating that individual player estimates can be improved by incorporating information from the entire dataset.

The James-Stein estimator is given by:

$$\hat{\theta}_i^{JS} = \bar{\theta} + \left(1 - \frac{(p-2)\sigma^2}{\sum_{i=1}^{p}(\theta_i - \bar{\theta})^2}\right)(\theta_i - \bar{\theta}) \tag{1}$$

where θ_i are the individual estimates, $\bar{\theta}$ is the overall mean, and p is the number of parameters.

Interpretation
The estimator reduces variance by shrinking extreme estimates toward the group mean, leading to more stable predictions.

When is Time for James-Stein?
The James-Stein estimator has better predictive accuracy than traditional methods when:

- There are at least three parameters being estimated.
- Individual estimates have high variability, making shrinkage beneficial.
- The parameters being estimated are related (such as batting averages across players in a season), allowing for shared information.
- The normality assumption holds, which underlies the theoretical justification for the estimator.

Limitations and Assumptions
The estimator assumes normality and is applicable when estimating three or more parameters.

Residuals & Leverage

Residuals represent the differences between observed and predicted values in a regression model:

$$e_i = y_i - \hat{y}_i \tag{2}$$

where e_i is the residual, y_i is the observed value, and $\hat{y}_i$ is the predicted value.

Standardized Residuals: Residuals adjusted for variability:

$$r_i = \frac{e_i}{s_e\sqrt{1 - h_{ii}}} \tag{3}$$

where s_e is the standard error of residuals.

Studentized Residuals: Further accounts for variance inflation:

$$t_i = \frac{e_i}{s_{(i)}\sqrt{1 - h_{ii}}} \tag{4}$$

where $s_{(i)}$ is the standard error excluding the i-th observation.

Leverage measures how far an observation's predictor values are from the mean predictor values:

$$h_{ii} = X_i(X^TX)^{-1}X_i^T \tag{5}$$

where h_{ii} is the leverage score of observation i.

Observations with high leverage ($h_{ii} > \frac{2p}{n}$) can disproportionately affect the regression model.

Cook's Distance: Quantifies how much an observation affects regression estimates:

$$D_i = \frac{r_i^2 h_{ii}}{p(1 - h_{ii})} \tag{6}$$

where p is the number of predictors.

A Cook's Distance greater than 1 suggests high influence.

DFITS:

- DFITS measures the effect each observation has on the fitted values in a linear model. DFITS represents approximately the number of standard deviations that the fitted value changes when each observation is removed from the data set and the model is refit.
- **Interpretation (DFITS)**:
  - Observations that have a large DFITS value may be influential.
  - A commonly used criterion for a large DFITS value is if DFITS is greater than $2\sqrt{p/n}$.

Diagnostic Classification:

Stat.	Unusual	Very Unusual
h_i	$> \frac{2p}{n}$	$> \frac{3p}{n}$
r	$\lvert r\rvert > 2$	$\lvert r\rvert > 3$
t	$\lvert t\rvert > 2$	$\lvert t\rvert > 3$
D	> 0.5	> 1.0

Diagnostic Procedures:

- Identify high-leverage points using h_{ii} values.
- Detect outliers using standardized residuals ($|r_i| > 2$ suggests an outlier).
- Evaluate influence with Cook's Distance.

- Use multiple diagnostics together for a comprehensive analysis.

Common Errors:

- Ignoring high-leverage points can distort model accuracy.
- Confusing residuals with standardized residuals in model assessment.
- Relying solely on one diagnostic measure instead of considering multiple checks.

Lasso/Ridge Regression

Regularization techniques prevent overfitting by imposing a penalty on coefficient magnitudes. Ridge regression (Tikhonov regularization) adds an ℓ_2 penalty, minimizing:

$$\sum_{i=1}^{n}(Y_i - X_i^\top \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2.$$

Lasso regression adds an ℓ_1 penalty, minimizing:

$$\sum_{i=1}^{n}(Y_i - X_i^\top \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|,$$

which encourages sparsity by setting some coefficients to zero.

1. **Standardization**: Regularization methods are sensitive to scale, so we standardize predictors and center the response:

$$X_{ij} \leftarrow \frac{X_{ij} - \bar{X}_j}{\sigma_{X_j}}, \quad \begin{matrix} \bar{X}_j = \frac{1}{n}\sum_{i=1}^{n} X_{ij}, \\ \sigma^2_{X_j} = \frac{1}{n}\sum_{i=1}^{n}(X_{ij} - \bar{X}_j)^2, \end{matrix}$$

$$Y_i \leftarrow Y_i - \bar{Y}, \quad \bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i.$$

   This ensures that the intercept is not penalized and facilitates numerical stability.

2. λ **Range Estimation**: The regularization parameter λ controls the penalty strength. We estimate a suitable range:
   - Compute eigenvalues $d_1, \ldots, d_p$ of $X^\top X$.
   - Set λ values on a logarithmic scale:

$$\lambda \in [0.01 d_{\min}, 10 d_{\max}].$$

   - For Lasso, the maximum λ is the smallest value where all $\hat{\beta} = 0$.

3. **Cross-Validation**: We use k-fold cross-validation (typically $k = 10$) to select the optimal λ:
   - Partition data into k folds.
   - For each λ, fit models on $k-1$ folds and validate on the remaining fold.
   - Ridge regression estimates coefficients via:

$$\hat{\beta}_{-k} = (X_{-k}^\top X_{-k} + \lambda I)^{-1} X_{-k}^\top Y_{-k}.$$

   - Predict on validation set:

$$\hat{Y}_k = X_k \hat{\beta}_{-k}.$$

   - Compute mean squared error (MSE):

$$\text{MSE}_k(\lambda) = \frac{1}{n_k}\sum_{i=1}^{n_k}(Y_{k,i} - \hat{Y}_{k,i})^2.$$

   - Compute cross-validation MSE:

$$\text{CV-MSE}(\lambda) = \frac{1}{k}\sum_{j=1}^{k}\text{MSE}_j(\lambda).$$

   - Choose $\lambda_{\min}$ minimizing CV-MSE.

   Lasso regression uses coordinate descent for coefficient estimation.

4. **Final Model**: Using $\lambda_{\min}$, refit the model on the full dataset:

$$\hat{\beta} = (X^\top X + \lambda_{\min} I)^{-1} X^\top Y.$$

   Lasso coefficients are obtained via iterative soft-thresholding.

5. **Output**: The final model contains:
   - Coefficients $\hat{\beta}$.
   - Residuals $Y - X\hat{\beta}$.
   - Fitted values $X\hat{\beta}$.
   - Model diagnostics (e.g., R^2, residual plots).

   Lasso may produce sparse $\hat{\beta}$, performing feature selection.

Prediction

Best Approaches:

- Use Multiple Linear Regression (MLR) or Ridge/Lasso Regression for many predictors.
- Apply Partial Least Squares (PLS) if predictors are highly collinear.
- Consider Polynomial Regression if nonlinearity is suspected.
- Split data into training and testing sets (e.g., 80%-20%) for model evaluation.

Best Fit Metrics:

- **Root Mean Squared Error (RMSE)**:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{7}$$

  RMSE measures the average magnitude of prediction errors. It provides an indication of how well the model predicts the dependent variable. Lower RMSE values indicate better model performance.

- **R-squared (R^2)**:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{8}$$

  R^2 represents the proportion of variance in the dependent variable explained by the model. A value closer to 1 indicates a strong relationship between predictors and response, whereas a value near 0 suggests a poor fit.

- **Predictive R^2**, PRESS Statistic – Evaluates predictive capability:

$$\text{PRESS} = \sum_{i=1}^{n}(y_i - \hat{y}_{(i)})^2 \tag{9}$$

  The Predicted Residual Sum of Squares (PRESS) statistic measures how well a regression model predicts new observations. Lower PRESS values indicate a better model.

- **PRESS R^2**:

$$R^2_{\text{PRESS}} = 1 - \frac{\text{PRESS}}{\sum(y_i - \bar{y})^2} \tag{10}$$

  PRESS R^2 evaluates model predictiveness using leave-one-out residuals. A higher PRESS R^2 value indicates stronger predictive performance.

- **Cross-Validation Mean Squared Error (CV MSE)**:

$$\text{CV MSE} = \frac{1}{k}\sum_{j=1}^{k}\text{MSE}_j \tag{11}$$

  CV MSE is an average of mean squared errors obtained from k-fold cross-validation. It helps in assessing model generalization to unseen data. Lower values indicate better predictive performance.

- **Mallow's C_p Statistic**:

$$C_p = \frac{SS_{\text{res}}}{\sigma^2} - (n - 2p) \tag{12}$$

  Mallow's C_p helps in model selection by balancing goodness-of-fit and model complexity. A smaller C_p value close to the number of predictors p suggests a well-fitting model.

Choosing the Best k for Cross-Validation:

- Small k (e.g., 5 or 10): Less computationally expensive but results in higher variance in the model's performance estimates. Suitable for large datasets.
- Large k (e.g., $k = n$, Leave-One-Out Cross-Validation - LOOCV): Provides a more precise estimate of model performance but is computationally expensive and can have higher variance in some cases. Suitable for small datasets where every observation matters.

Understanding Relationships

Best Approaches:

- Use Ordinary Least Squares (OLS) Regression.
- Check p-values of coefficients (H_0: coefficient = 0).
- Include interaction terms if relationships depend on other variables.

Best Fit Metrics:

- **Adjusted R^2**: Accounts for number of predictors, preventing overfitting.
- **p-values, Confidence Intervals (CIs)**: Assess predictor significance.
- **Variance Inflation Factor (VIF)**:

$$\text{VIF}_j = \frac{1}{1 - R_j^2} \tag{13}$$

  Detects multicollinearity; VIF $>$ 10 suggests high collinearity.

Causal Inference

Best Approaches:

- Use regression with a designed experiment.
- Apply Instrumental Variable (IV) Regression to address endogeneity.
- Consider Difference-in-Differences (DiD) or Propensity Score Matching (PSM).

Best Fit Metrics:

- **Coefficient Interpretability**: Theoretical validity of estimates.
- **R^2 and Adjusted R^2**: Measure explanatory power.
- **Causal Tests**: Check for confounders, robustness checks.

Feature Selection/Importance

Best Approaches:

- Use Lasso Regression for automatic feature selection.
- Try Stepwise Regression (Forward/Backward selection).
- Apply tree-based models (Random Forest, XGBoost) for ranking importance.

Best Fit Metrics:

- **Feature Importance Scores** (from tree-based models).
- **Akaike Information Criterion (AIC)**:

$$AIC = 2k - 2\ln(L) \tag{14}$$

  Lower is better; penalizes complexity.
- **Bayesian Information Criterion (BIC)**: Similar to AIC but penalizes complexity more strongly.
- **Adjusted R^2**: Helps avoid overfitting.

Common Errors to Avoid:

- Ignoring multicollinearity (use VIF checks).
- Overfitting with too many predictors (use AIC/BIC, cross-validation).
- Assuming correlation implies causation (use causal inference methods).
- Using RMSE alone for model evaluation (also consider Adjusted R^2 and predictive performance).

Robust Regression

Problems Addressed by Robust Regression:

- Non-constant variance (heteroscedasticity)
- Correlated errors (autocorrelation)
- Non-normality of errors
- Overfitting and multicollinearity
- Protect against influential outliers
- Useful for detecting outliers
- Check results against a least squares fit

Definition: Robust regression methods provide an alternative to ordinary least squares (OLS) regression by requiring less restrictive assumptions. They aim to reduce the influence of outliers, providing a better fit for the majority of the data.

OLS vs. Robust Regression:

- OLS minimizes the sum of squared residuals, making it sensitive to outliers:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - X_i\beta)^2$$

- Robust regression uses alternative norms or weighting schemes to down-weight influential points.

Least Absolute Deviation (L1-Norm) Regression:

$$\min_{\beta} \sum_{i=1}^{n} |y_i - X_i\beta|$$

Least Median of Squares (LMS) Regression:

$$\min_{\beta} \text{median}_i (y_i - X_i\beta)^2$$

Iteratively Reweighted Least Squares (IRLS):

- IRLS is a common robust regression technique using weighted least squares.
- Weights are determined based on residuals and updated iteratively:

$$\min_{\beta} \sum_{i=1}^{n} w_i (y_i - X_i\beta)^2$$

- The weights are updated using a robust function, e.g., Huber or Tukey's biweight function.

This is equivalent to solving an iterated reweighted least-squares problem:

$$\min \sum_i w\left(r_i^{k-1}\right) r_i^2,$$

where k denotes the iteration number, and weights are updated iteratively.

The influence function $\psi(x)$ quantifies the effect of an observation on the parameter estimate. For least-squares ($\rho(x) = x^2/2$), $\psi(x) = x$, meaning the influence grows linearly with the residual, making it non-robust. A robust M-estimator must satisfy:

1. A bounded influence function to limit outlier effects.
2. A unique solution to ensure estimator stability.

Convergence:

- Choose initial weights.
- Perform weighted least squares.
- Update weights using residuals.
- Repeat until convergence.

Interpretation: (Robust regression) particularly useful when data contains outliers or violates the assumptions of normality and constant variance. By reducing the influence of these points, robust regression provides more reliable coefficient estimates and improved model performance.

Type	$\rho(x)$	$\psi(x)$	$w(x)$
L_2	$x^2/2$	x	1
L_1	$\lvert x\rvert$	$\text{sgn}(x)$	$\frac{1}{\lvert x\rvert}$
$L_1 - L_2$	$2(\sqrt{1 + x^2/2} - 1)$	$\frac{x}{\sqrt{1+x^2/2}}$	$\frac{1}{\sqrt{1+x^2/2}}$
L_p	$\frac{\lvert x\rvert^\nu}{\nu}$	$\text{sgn}(x)\lvert x\rvert^{\nu-1}$	$\lvert x\rvert^{\nu-2}$
"Fair"	$c^2\left[\frac{\lvert x\rvert}{c} - \log\left(1 + \frac{\lvert x\rvert}{c}\right)\right]$	$\frac{x}{1+\lvert x\rvert/c}$	$\frac{1}{1+\lvert x\rvert/c}$
Huber $\begin{cases}\text{if } \lvert x\rvert \le k \\ \text{if } \lvert x\rvert > k\end{cases}$	$\begin{cases}x^2/2 \\ k(\lvert x\rvert - k/2)\end{cases}$	$\begin{cases}x \\ k\,\text{sgn}(x)\end{cases}$	$\begin{cases}1 \\ k/\lvert x\rvert\end{cases}$
Cauchy	$\frac{c^2}{2}\log(1 + (x/c)^2)$	$\frac{x}{1+(x/c)^2}$	$\frac{1}{1+(x/c)^2}$
German-McClure	$\frac{x^2/2}{1+x^2}$	$\frac{x}{(1+x^2)^2}$	$\frac{1}{(1+x^2)^2}$
Welsch	$\frac{c^2}{2}\left[1 - \exp\left(-(x/c)^2\right)\right]$	$x\exp\left(-(x/c)^2\right)$	$\exp\left(-(x/c)^2\right)$
Tukey $\begin{cases}\text{if } \lvert x\rvert \le c \\ \text{if } \lvert x\rvert > c\end{cases}$	$\begin{cases}\frac{c^2}{6}\left(1 - \left[1 - (x/c)^2\right]^3\right) \\ c^2/6\end{cases}$	$\begin{cases}x\left[1 - (x/c)^2\right]^2 \\ 0\end{cases}$	$\begin{cases}\left[1 - (x/c)^2\right]^2 \\ 0\end{cases}$