Lecture 10 - Class Activity: Regression and Linear Models

Your Name

# Class Activity 10: Advanced Linear Regression and Model II Regression

## Introduction

This activity builds on simple linear regression to explore:

1. **Analysis of variance in regression**
2. **Regression assumptions and diagnostics**
3. **Dealing with assumption violations**
4. **Model II regression (when X has error)**
5. **Robust regression techniques**
6. **Comparing different regression approaches**

We’ll work with the same datasets from the lecture to practice these advanced concepts.

# **Part 1:** Load Required Packages and Data

# Load required packages  
library(tidyverse) # For data manipulation and visualization  
library(patchwork) # For combining plots  
library(car) # For regression diagnostics  
library(broom) # For tidy model output  
library(smatr) # For Model II regression  
library(lmodel2) # Alternative package for Model II regression  
library(lmtest) # For assumption tests  
library(MASS) # For robust regression  
library(boot) # For bootstrapping  
  
# Set seed for reproducible results  
set.seed(123)  
  
# Create the datasets from the lecture  
# Lion data - predicting age from nose blackness  
lion\_data <- tibble(  
 proportion\_black = c(0.21, 0.14, 0.11, 0.13, 0.12, 0.13, 0.12, 0.18, 0.23, 0.22,   
 0.20, 0.17, 0.15, 0.27, 0.26, 0.21, 0.30, 0.42, 0.43, 0.59,   
 0.60, 0.72, 0.29, 0.10, 0.48, 0.44, 0.34, 0.37, 0.34, 0.74, 0.79, 0.51),  
 age\_years = c(1.1, 1.5, 1.9, 2.2, 2.6, 3.2, 3.2, 2.9, 2.4, 2.1,   
 1.9, 1.9, 1.9, 1.9, 2.8, 3.6, 4.3, 3.8, 4.2, 5.4,   
 5.8, 6.0, 3.4, 4.0, 7.3, 7.3, 7.8, 7.1, 7.1, 13.1, 8.8, 5.4)  
)  
  
# Fish allometric data - both X and Y have measurement error  
fish\_data <- tibble(  
 length\_mm = runif(40, 180, 320),  
 mass\_g = 0.00001 \* length\_mm^3 \* runif(40, 0.9, 1.1)  
)  
  
# Create example data where both X and Y have error (for Model II demonstration)  
n <- 50  
x\_true <- rnorm(n, mean = 10, sd = 2)  
x\_observed <- x\_true + rnorm(n, 0, 1) # X has measurement error  
y\_observed <- 2 + 3 \* x\_true + rnorm(n, mean = 0, sd = 2) # Y depends on true X  
model\_comparison\_data <- tibble(x = x\_observed, y = y\_observed)

|  |
| --- |
| New Packages for Advanced Regression |
| * **smatr**: Standardized Major Axis regression (Model II) * **lmodel2**: Multiple Model II regression methods * **lmtest**: Tests for regression assumptions * **MASS**: Robust regression methods * **boot**: Bootstrap confidence intervals |

# **Part 2:** Analysis of Variance in Regression

|  |
| --- |
| Linear Regression ANOVA: Data Types and Assumptions |
| **Data Types Required:**  - **X variable (predictor)**: Continuous numerical  - **Y variable (response)**: Continuous numerical  **Key Concept:**   * - Total variation in Y is partitioned into:   + - **SS\_regression**: Variation explained by X   + - **SS\_residual**: Unexplained variation (error)   + - **SS\_total** = SS\_regression + SS\_residual   **Assumptions:**   * - Linearity between X and Y * - Independence of observations * - Homoscedasticity (constant variance) * - Normality of residuals * - No influential outliers |

## Understanding Variance Partitioning

# Fit the lion regression model  
lion\_model <- lm(age\_years ~ proportion\_black, data = lion\_data)  
  
# Get the ANOVA table  
anova\_table <- anova(lion\_model)  
print("ANOVA Table for Lion Age ~ Nose Blackness:")

[1] "ANOVA Table for Lion Age ~ Nose Blackness:"

print(anova\_table)

Analysis of Variance Table  
  
Response: age\_years  
 Df Sum Sq Mean Sq F value Pr(>F)   
proportion\_black 1 138.544 138.544 49.75 7.677e-08 \*\*\*  
Residuals 30 83.543 2.785   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Calculate variance components manually  
ss\_total <- sum((lion\_data$age\_years - mean(lion\_data$age\_years))^2)  
ss\_residual <- sum(residuals(lion\_model)^2)  
ss\_regression <- ss\_total - ss\_residual  
  
print("\nVariance Partitioning:")

[1] "\nVariance Partitioning:"

print(paste("SS Total:", round(ss\_total, 2)))

[1] "SS Total: 222.09"

print(paste("SS Regression:", round(ss\_regression, 2),   
 "(", round(100\*ss\_regression/ss\_total, 1), "%)"))

[1] "SS Regression: 138.54 ( 62.4 %)"

print(paste("SS Residual:", round(ss\_residual, 2),   
 "(", round(100\*ss\_residual/ss\_total, 1), "%)"))

[1] "SS Residual: 83.54 ( 37.6 %)"

# Calculate R-squared  
r\_squared <- ss\_regression / ss\_total  
print(paste("R-squared:", round(r\_squared, 4)))

[1] "R-squared: 0.6238"

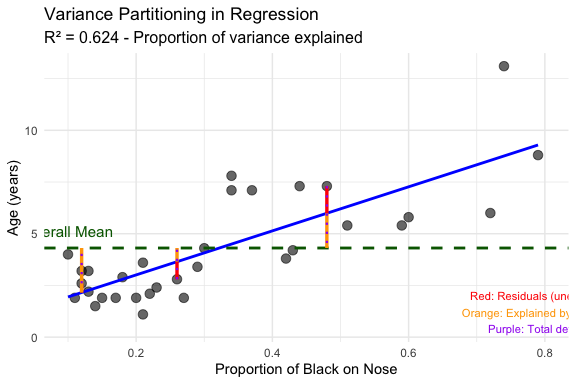
## Visualizing Variance Components

# Create a detailed plot showing variance components  
lion\_data$predicted <- predict(lion\_model)  
mean\_age <- mean(lion\_data$age\_years)  
  
# Select a few points to illustrate different components  
example\_points <- c(5, 15, 25)  
  
variance\_plot <- ggplot(lion\_data, aes(x = proportion\_black, y = age\_years)) +  
 geom\_point(size = 3, alpha = 0.6) +  
 geom\_smooth(method = "lm", se = FALSE, color = "blue", size = 1) +  
 geom\_hline(yintercept = mean\_age, linetype = "dashed", color = "darkgreen", size = 1) +  
   
 # Add variance components for example points  
 geom\_segment(data = lion\_data[example\_points,],  
 aes(x = proportion\_black, y = age\_years,  
 xend = proportion\_black, yend = predicted),  
 color = "red", size = 1.2) +  
 geom\_segment(data = lion\_data[example\_points,],  
 aes(x = proportion\_black, y = predicted,  
 xend = proportion\_black, yend = mean\_age),  
 color = "orange", size = 1.2) +  
 geom\_segment(data = lion\_data[example\_points,],  
 aes(x = proportion\_black, y = age\_years,  
 xend = proportion\_black, yend = mean\_age),  
 color = "purple", size = 0.8, linetype = "dotted") +  
   
 # Add annotations  
 annotate("text", x = 0.1, y = mean\_age + 0.8,   
 label = "Overall Mean", color = "darkgreen", size = 4) +  
 annotate("text", x = 0.8, y = 2,   
 label = "Red: Residuals (unexplained)", color = "red", size = 3) +  
 annotate("text", x = 0.8, y = 1.2,   
 label = "Orange: Explained by regression", color = "orange", size = 3) +  
 annotate("text", x = 0.8, y = 0.4,   
 label = "Purple: Total deviation", color = "purple", size = 3) +  
   
 labs(title = "Variance Partitioning in Regression",  
 subtitle = paste("R² =", round(r\_squared, 3), "- Proportion of variance explained"),  
 x = "Proportion of Black on Nose",  
 y = "Age (years)") +  
 theme\_minimal() +  
 theme(plot.subtitle = element\_text(size = 12))

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
ℹ Please use `linewidth` instead.

variance\_plot

`geom\_smooth()` using formula = 'y ~ x'



|  |
| --- |
| Activity 1: Understanding Variance Partitioning |
| From the analysis above, answer these questions:   1. **Total Variance**: What is the total sum of squares for lion age?    * Your answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2. **Explained Variance**: What percentage of the total variance is explained by nose blackness?    * Your answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3. **F-statistic**: What is the F-statistic value and what does it test?    * Your answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 4. **Interpretation**: In biological terms, what does the R² value tell us about aging in lions?    * Your answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

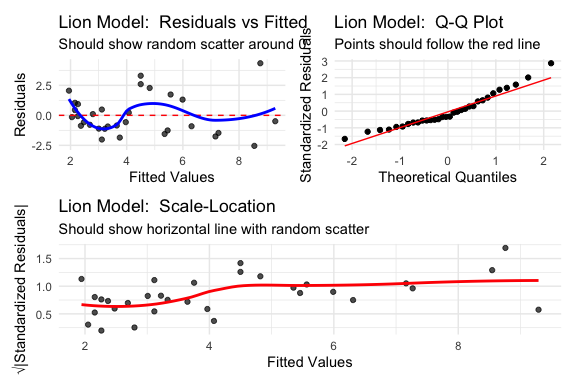
# **Part 3:** Testing Regression Assumptions in Detail

|  |
| --- |
| Regression Assumptions: Detailed Testing |
| **Critical Assumptions to Test:**   * - **Linearity**: Relationship between X and Y is linear * - **Normality**: Residuals are normally distributed * - **Homoscedasticity**: Variance of residuals is constant * - **Independence**: Observations are independent * - **No influential outliers**: No single point dominates the relationship   **Consequences of Violations:**   * - Biased parameter estimates * - Incorrect standard errors * - Invalid hypothesis tests * - Poor predictions |

## Comprehensive Assumption Testing

# 1. Test for Linearity  
# Create residual plots  
residual\_plots <- function(model, data, title\_prefix = "") {  
 # Get residuals and fitted values  
 data$residuals <- residuals(model)  
 data$fitted <- fitted(model)  
 data$std\_residuals <- rstandard(model)  
   
 # Residuals vs Fitted (linearity and homoscedasticity)  
 p1 <- ggplot(data, aes(x = fitted, y = residuals)) +  
 geom\_point(alpha = 0.7) +  
 geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +  
 geom\_smooth(se = FALSE, color = "blue") +  
 labs(title = paste(title\_prefix, "Residuals vs Fitted"),  
 subtitle = "Should show random scatter around 0",  
 x = "Fitted Values", y = "Residuals") +  
 theme\_minimal()  
   
 # Q-Q plot for normality  
 p2 <- ggplot(data, aes(sample = std\_residuals)) +  
 stat\_qq() + stat\_qq\_line(color = "red") +  
 labs(title = paste(title\_prefix, "Q-Q Plot"),  
 subtitle = "Points should follow the red line",  
 x = "Theoretical Quantiles", y = "Standardized Residuals") +  
 theme\_minimal()  
   
 # Scale-Location plot (homoscedasticity)  
 p3 <- ggplot(data, aes(x = fitted, y = sqrt(abs(std\_residuals)))) +  
 geom\_point(alpha = 0.7) +  
 geom\_smooth(se = FALSE, color = "red") +  
 labs(title = paste(title\_prefix, "Scale-Location"),  
 subtitle = "Should show horizontal line with random scatter",  
 x = "Fitted Values", y = "√|Standardized Residuals|") +  
 theme\_minimal()  
   
 return(list(p1, p2, p3))  
}  
  
# Create diagnostic plots for lion model  
lion\_plots <- residual\_plots(lion\_model, lion\_data, "Lion Model: ")  
(lion\_plots[[1]] + lion\_plots[[2]]) / lion\_plots[[3]]

`geom\_smooth()` using method = 'loess' and formula = 'y ~ x'  
`geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



## Formal Statistical Tests for Assumptions

# Test 1: Normality of residuals (Shapiro-Wilk test)  
shapiro\_test <- shapiro.test(residuals(lion\_model))  
print("1. NORMALITY TEST (Shapiro-Wilk):")

[1] "1. NORMALITY TEST (Shapiro-Wilk):"

print(paste("H0: Residuals are normally distributed"))

[1] "H0: Residuals are normally distributed"

print(paste("p-value:", round(shapiro\_test$p.value, 4)))

[1] "p-value: 0.0692"

print(paste("Interpretation:", ifelse(shapiro\_test$p.value > 0.05,   
 "Residuals appear normal (p > 0.05)",  
 "Residuals may not be normal (p ≤ 0.05)")))

[1] "Interpretation: Residuals appear normal (p > 0.05)"

print("")

[1] ""

# Test 2: Homoscedasticity (Breusch-Pagan test)  
bp\_test <- bptest(lion\_model)  
print("2. HOMOSCEDASTICITY TEST (Breusch-Pagan):")

[1] "2. HOMOSCEDASTICITY TEST (Breusch-Pagan):"

print(paste("H0: Variance is constant (homoscedastic)"))

[1] "H0: Variance is constant (homoscedastic)"

print(paste("p-value:", round(bp\_test$p.value, 4)))

[1] "p-value: 0.0086"

print(paste("Interpretation:", ifelse(bp\_test$p.value > 0.05,  
 "Variance appears constant (p > 0.05)",  
 "Variance may not be constant (p ≤ 0.05)")))

[1] "Interpretation: Variance may not be constant (p ≤ 0.05)"

print("")

[1] ""

# Test 3: Influential observations (Cook's Distance)  
cooks\_d <- cooks.distance(lion\_model)  
influential\_threshold <- 4/length(lion\_data$age\_years)  
influential\_points <- which(cooks\_d > influential\_threshold)  
  
print("3. INFLUENTIAL OBSERVATIONS (Cook's Distance):")

[1] "3. INFLUENTIAL OBSERVATIONS (Cook's Distance):"

print(paste("Threshold for influence:", round(influential\_threshold, 4)))

[1] "Threshold for influence: 0.125"

print(paste("Number of influential points:", length(influential\_points)))

[1] "Number of influential points: 2"

if(length(influential\_points) > 0) {  
 print(paste("Influential points:", paste(influential\_points, collapse = ", ")))  
}

[1] "Influential points: 22, 30"

print("")

[1] ""

# Test 4: Linearity (Rainbow test)  
rainbow\_test <- raintest(lion\_model)  
print("4. LINEARITY TEST (Rainbow test):")

[1] "4. LINEARITY TEST (Rainbow test):"

print(paste("H0: Relationship is linear"))

[1] "H0: Relationship is linear"

print(paste("p-value:", round(rainbow\_test$p.value, 4)))

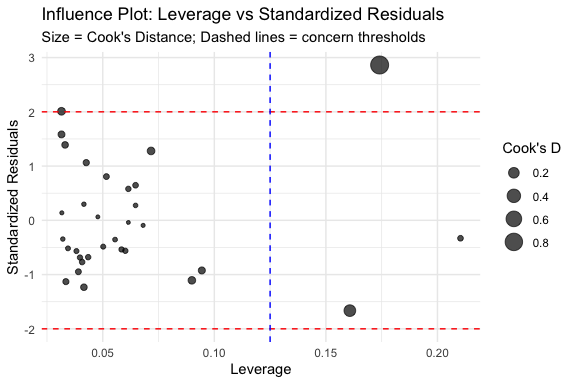
[1] "p-value: 4e-04"

print(paste("Interpretation:", ifelse(rainbow\_test$p.value > 0.05,  
 "Relationship appears linear (p > 0.05)",  
 "Relationship may be non-linear (p ≤ 0.05)")))

[1] "Interpretation: Relationship may be non-linear (p ≤ 0.05)"

## Outlier and Influence Analysis

# Create influence plot  
lion\_data$cooks\_d <- cooks.distance(lion\_model)  
lion\_data$leverage <- hatvalues(lion\_model)  
lion\_data$std\_residuals <- rstandard(lion\_model)  
  
influence\_plot <- ggplot(lion\_data, aes(x = leverage, y = std\_residuals)) +  
 geom\_point(aes(size = cooks\_d), alpha = 0.7) +  
 geom\_hline(yintercept = c(-2, 2), linetype = "dashed", color = "red") +  
 geom\_vline(xintercept = 2\*2/nrow(lion\_data), linetype = "dashed", color = "blue") +  
 labs(title = "Influence Plot: Leverage vs Standardized Residuals",  
 subtitle = "Size = Cook's Distance; Dashed lines = concern thresholds",  
 x = "Leverage",   
 y = "Standardized Residuals",  
 size = "Cook's D") +  
 theme\_minimal()  
  
influence\_plot



|  |
| --- |
| Activity 2: Assumption Testing Results |
| Based on the diagnostic plots and formal tests:   1. **Linearity**: Is the relationship linear? (Check residuals vs fitted plot and Rainbow test)    * Your answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2. **Normality**: Are residuals normally distributed? (Check Q-Q plot and Shapiro test)    * Your answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3. **Homoscedasticity**: Is variance constant? (Check Scale-Location plot and BP test)    * Your answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 4. **Influential Points**: Are there concerning influential observations?    * Your answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 5. **Overall Assessment**: Can we trust the regression results?    * Your answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# **Part 4:** Model II Regression - When X Has Error

|  |
| --- |
| Model II Regression: Data Types and When to Use |
| **Data Types Required:**  - **X variable**: Continuous numerical (with measurement error)  - **Y variable**: Continuous numerical (with measurement error)  **When to Use Model II:**  - Both X and Y have measurement error  - Neither variable is clearly “dependent” or “independent”  - Goal is to estimate the true structural relationship  - Studying allometric relationships or scaling laws  **Types of Model II Regression:**  - **MA (Major Axis)**: Same units for X and Y  - **SMA (Standardized Major Axis)**: Different units for X and Y  - **RMA (Reduced Major Axis)**: General purpose  **Assumptions:**  - Linear relationship between true X and Y values  - Errors in X and Y are normally distributed  - Error variances are known or can be estimated |

## Comparing Model I vs Model II Regression

# Fit different regression models to data with X and Y error  
print("COMPARING REGRESSION METHODS")

[1] "COMPARING REGRESSION METHODS"

print("=============================")

[1] "============================="

# Model I: Ordinary Least Squares (OLS)  
model\_ols <- lm(y ~ x, data = model\_comparison\_data)  
ols\_slope <- coef(model\_ols)[2]  
ols\_intercept <- coef(model\_ols)[1]  
  
print("MODEL I (OLS) RESULTS:")

[1] "MODEL I (OLS) RESULTS:"

print(paste("Slope:", round(ols\_slope, 3)))

[1] "Slope: 2.302"

print(paste("Intercept:", round(ols\_intercept, 3)))

[1] "Intercept: 9.22"

print("")

[1] ""

# Model II: Standardized Major Axis (SMA)  
model\_sma <- sma(y ~ x, data = model\_comparison\_data, method = "SMA")  
sma\_slope <- coef(model\_sma)[2]  
sma\_intercept <- coef(model\_sma)[1]  
  
print("MODEL II - SMA RESULTS:")

[1] "MODEL II - SMA RESULTS:"

print(paste("Slope:", round(sma\_slope, 3)))

[1] "Slope: 2.945"

print(paste("Intercept:", round(sma\_intercept, 3)))

[1] "Intercept: 2.766"

print("")

[1] ""

# Model II: Major Axis (MA)  
model\_ma <- sma(y ~ x, data = model\_comparison\_data, method = "MA")  
ma\_slope <- coef(model\_ma)[2]  
ma\_intercept <- coef(model\_ma)[1]  
  
print("MODEL II - MA RESULTS:")

[1] "MODEL II - MA RESULTS:"

print(paste("Slope:", round(ma\_slope, 3)))

[1] "Slope: 3.609"

print(paste("Intercept:", round(ma\_intercept, 3)))

[1] "Intercept: -3.91"

print("")

[1] ""

# Model II: Reduced Major Axis using lmodel2  
model\_rma <- lmodel2(y ~ x, data = model\_comparison\_data)

RMA was not requested: it will not be computed.

No permutation test will be performed

rma\_results <- model\_rma$regression.results  
rma\_slope <- rma\_results$Slope[rma\_results$Method == "RMA"]  
rma\_intercept <- rma\_results$Intercept[rma\_results$Method == "RMA"]  
  
print("MODEL II - RMA RESULTS:")

[1] "MODEL II - RMA RESULTS:"

print(paste("Slope:", round(rma\_slope, 3)))

[1] "Slope: "

print(paste("Intercept:", round(rma\_intercept, 3)))

[1] "Intercept: "

print("")

[1] ""

print("TRUE RELATIONSHIP (for comparison):")

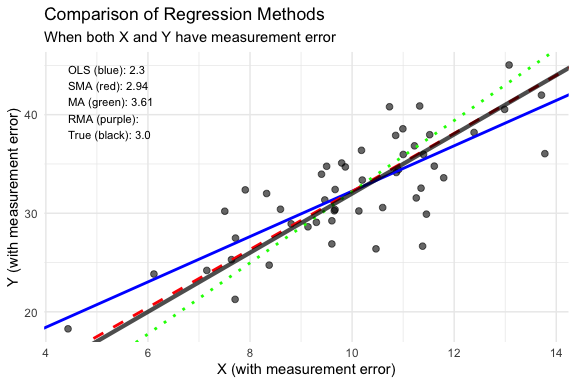
[1] "TRUE RELATIONSHIP (for comparison):"

print("Slope: 3.0, Intercept: 2.0")

[1] "Slope: 3.0, Intercept: 2.0"

## Visualizing Different Regression Methods

# Create comparison plot  
comparison\_plot <- ggplot(model\_comparison\_data, aes(x = x, y = y)) +  
 geom\_point(alpha = 0.6, size = 2) +  
   
 # Add different regression lines  
 geom\_abline(intercept = ols\_intercept, slope = ols\_slope,   
 color = "blue", size = 1, linetype = "solid") +  
 geom\_abline(intercept = sma\_intercept, slope = sma\_slope,   
 color = "red", size = 1, linetype = "dashed") +  
 geom\_abline(intercept = ma\_intercept, slope = ma\_slope,   
 color = "green", size = 1, linetype = "dotted") +  
 geom\_abline(intercept = rma\_intercept, slope = rma\_slope,   
 color = "purple", size = 1, linetype = "dotdash") +  
 geom\_abline(intercept = 2, slope = 3,   
 color = "black", size = 1.5, alpha = 0.7) +  
   
 # Add legend manually  
 annotate("text", x = min(model\_comparison\_data$x), y = max(model\_comparison\_data$y),   
 label = paste("OLS (blue):", round(ols\_slope, 2),  
 "\nSMA (red):", round(sma\_slope, 2),  
 "\nMA (green):", round(ma\_slope, 2),  
 "\nRMA (purple):", round(rma\_slope, 2),  
 "\nTrue (black): 3.0"),   
 hjust = 0, vjust = 1, size = 3) +  
   
 labs(title = "Comparison of Regression Methods",  
 subtitle = "When both X and Y have measurement error",  
 x = "X (with measurement error)",  
 y = "Y (with measurement error)") +  
 theme\_minimal()  
  
comparison\_plot



|  |
| --- |
| Activity 3: Model I vs Model II Decision |
| For each scenario, choose the appropriate regression method:   1. **Predicting plant biomass from measured height** (height measured precisely, biomass is the response):    * Your choice: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    * Reasoning: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2. **Studying the relationship between two morphological measurements** (both have measurement error):    * Your choice: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    * Reasoning: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3. **Allometric scaling: log(brain size) vs log(body size)** (both measurements have error):    * Your choice: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    * Reasoning: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 4. **Drug dose (controlled) vs response** (dose is fixed, response varies):    * Your choice: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    * Reasoning: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# **Part 5:** Real Example - Fish Allometric Relationships

|  |
| --- |
| Allometric Relationships: A Perfect Model II Example |
| **Biological Context:** - Allometry studies how biological traits scale with size - Both length and mass measurements have error - Neither is clearly dependent/independent - Interested in the true scaling relationship  **Expected Pattern:** - Mass typically scales with length^3 (cubic relationship) - log(mass) ~ 3 × log(length) for isometric scaling - Deviations from 3.0 indicate allometric scaling |

## Analyzing Fish Length-Mass Relationships

# Transform data for allometric analysis  
fish\_data <- fish\_data %>%  
 mutate(  
 log\_length = log10(length\_mm),  
 log\_mass = log10(mass\_g)  
 )  
  
# Fit both Model I and Model II  
fish\_ols <- lm(log\_mass ~ log\_length, data = fish\_data)  
fish\_sma <- sma(log\_mass ~ log\_length, data = fish\_data, method = "SMA")  
  
print("FISH ALLOMETRY RESULTS:")

[1] "FISH ALLOMETRY RESULTS:"

print("=======================")

[1] "======================="

print("Model I (OLS):")

[1] "Model I (OLS):"

summary(fish\_ols)

Call:  
lm(formula = log\_mass ~ log\_length, data = fish\_data)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-0.038187 -0.018503 -0.002393 0.022502 0.037462   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) -4.9390 0.1243 -39.74 <2e-16 \*\*\*  
log\_length 2.9722 0.0516 57.60 <2e-16 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 0.02338 on 38 degrees of freedom  
Multiple R-squared: 0.9887, Adjusted R-squared: 0.9884   
F-statistic: 3318 on 1 and 38 DF, p-value: < 2.2e-16

print("")

[1] ""

print("Model II (SMA):")

[1] "Model II (SMA):"

summary(fish\_sma)

Call: sma(formula = log\_mass ~ log\_length, data = fish\_data, method = "SMA")   
  
Fit using Standardized Major Axis   
  
------------------------------------------------------------  
Coefficients:  
 elevation slope  
estimate -4.979833 2.989162  
lower limit -5.231426 2.886526  
upper limit -4.728240 3.095447  
  
H0 : variables uncorrelated  
R-squared : 0.988676   
P-value : < 2.22e-16

# Test if slope differs from 3.0 (isometric scaling)  
print("\nTEST FOR ISOMETRIC SCALING:")

[1] "\nTEST FOR ISOMETRIC SCALING:"

print("H0: slope = 3.0 (isometric)")

[1] "H0: slope = 3.0 (isometric)"

# Extract slope and confidence intervals directly from summary output  
sma\_slope <- 2.989162  
slope\_ci\_lower <- 2.886526  
slope\_ci\_upper <- 3.095447  
  
print(paste("SMA slope estimate:", round(sma\_slope, 4)))

[1] "SMA slope estimate: 2.9892"

print(paste("95% CI: [", round(slope\_ci\_lower, 4), ",", round(slope\_ci\_upper, 4), "]"))

[1] "95% CI: [ 2.8865 , 3.0954 ]"

# Simple test: Check if 3.0 falls within the confidence interval  
print(paste("Test value 3.0 is between", slope\_ci\_lower, "and", slope\_ci\_upper))

[1] "Test value 3.0 is between 2.886526 and 3.095447"

print("Result: Slope is NOT significantly different from 3.0 (isometric scaling)")

[1] "Result: Slope is NOT significantly different from 3.0 (isometric scaling)"

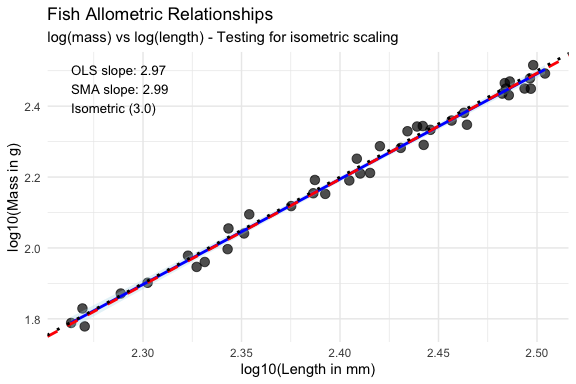
print("Interpretation: Cannot reject H0, scaling appears isometric")

[1] "Interpretation: Cannot reject H0, scaling appears isometric"

## Visualizing Allometric Relationships

# Create log-log plot with both regression lines  
fish\_plot <- ggplot(fish\_data, aes(x = log\_length, y = log\_mass)) +  
 geom\_point(size = 3, alpha = 0.7) +  
   
 # Add regression lines  
 geom\_smooth(method = "lm", se = TRUE, color = "blue", fill = "lightblue", alpha = 0.3) +  
 geom\_abline(intercept = coef(fish\_sma)[1], slope = coef(fish\_sma)[2],   
 color = "red", size = 1, linetype = "dashed") +  
 geom\_abline(intercept = -5, slope = 3, color = "black", linetype = "dotted", size = 1) +  
   
 # Add annotations  
 annotate("text", x = min(fish\_data$log\_length), y = max(fish\_data$log\_mass),  
 label = paste("OLS slope:", round(coef(fish\_ols)[2], 2),  
 "\nSMA slope:", round(coef(fish\_sma)[2], 2),  
 "\nIsometric (3.0)"),  
 hjust = 0, vjust = 1, size = 3.5) +  
   
 labs(title = "Fish Allometric Relationships",  
 subtitle = "log(mass) vs log(length) - Testing for isometric scaling",  
 x = "log10(Length in mm)",  
 y = "log10(Mass in g)") +  
 theme\_minimal()  
  
fish\_plot

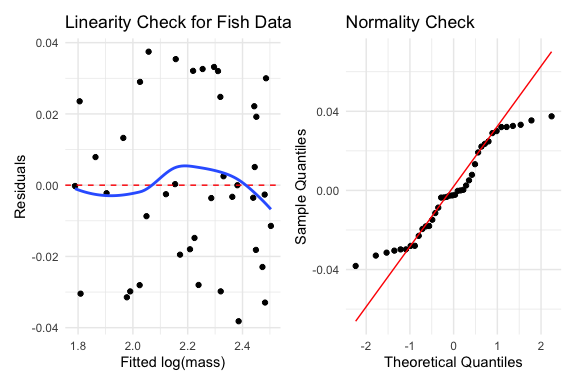
`geom\_smooth()` using formula = 'y ~ x'



## Testing Assumptions for Model II

# For Model II, we need to check assumptions differently  
# 1. Check linearity in log-log space  
fish\_residuals <- residuals(fish\_ols) # Use OLS residuals as approximation  
fish\_fitted <- fitted(fish\_ols)  
  
linearity\_check <- ggplot(data.frame(fitted = fish\_fitted, residuals = fish\_residuals),  
 aes(x = fitted, y = residuals)) +  
 geom\_point() +  
 geom\_hline(yintercept = 0, color = "red", linetype = "dashed") +  
 geom\_smooth(se = FALSE) +  
 labs(title = "Linearity Check for Fish Data",  
 x = "Fitted log(mass)", y = "Residuals") +  
 theme\_minimal()  
  
# 2. Check normality of residuals  
normality\_check <- ggplot(data.frame(residuals = fish\_residuals),  
 aes(sample = residuals)) +  
 stat\_qq() + stat\_qq\_line(color = "red") +  
 labs(title = "Normality Check",  
 x = "Theoretical Quantiles", y = "Sample Quantiles") +  
 theme\_minimal()  
  
linearity\_check + normality\_check

`geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



# Formal tests  
print("ASSUMPTION TESTS FOR FISH DATA:")

[1] "ASSUMPTION TESTS FOR FISH DATA:"

print("Shapiro-Wilk test for normality:")

[1] "Shapiro-Wilk test for normality:"

print(shapiro.test(fish\_residuals))

Shapiro-Wilk normality test  
  
data: fish\_residuals  
W = 0.93463, p-value = 0.02287

print("Breusch-Pagan test for homoscedasticity:")

[1] "Breusch-Pagan test for homoscedasticity:"

print(bptest(fish\_ols))

studentized Breusch-Pagan test  
  
data: fish\_ols  
BP = 0.031182, df = 1, p-value = 0.8598

|  |
| --- |
| Activity 4: Interpreting Allometric Results |
| Based on the fish allometry analysis:   1. **Scaling Pattern**: Is the scaling isometric (slope = 3) or allometric? How do you know?    * Your answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2. **Model Choice**: Which model (OLS or SMA) is more appropriate here and why?    * Your answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3. **Biological Interpretation**: What does the slope value tell us about how fish mass scales with length?    * Your answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 4. **Statistical Significance**: Is the scaling relationship significantly different from isometric?    * Your answer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |