MA214 Applied Statistics - Project 1 Week 3 Activity

Modeling, Model Selection, and Diagnostics

Section C4 | Group 7

2026-02-18

Contents

Overview . . . 1

Part 0: Load Your Data . . . 2

SECTION A: LINEAR REGRESSION **3**

A1: Fitting Linear Regression Models . . . 3

A2: Model Selection (Linear) . . . 9

A3: Model Diagnostics (Linear) . . . 10

A4: Addressing Issues (Linear) . . . 14

A5: Final Linear Model . . . 18

Checklist for Today . . . 20

Resources and Tips . . . 20

Submission . . . 21

Group Information . . . 21

Overview

Today's Focus: Fitting regression models, selecting the best model, and validating model assumptions.

Important: This worksheet covers BOTH linear regression and logistic regression. **Choose the section that matches your project's response variable:**

- **Section A (Linear Regression):** For continuous/numeric response variables (e.g., price, temperature, age)
- **Section B (Logistic Regression):** For binary response variables (e.g., yes/no, success/failure, 0/1)

What you should have ready: - Your cleaned dataset - Summary statistics and EDA from Week 2 - Initial ideas about which variables to include

By the end of today, you should have: - Fitted at least 2-3 regression models - Selected your best model with justification - Checked model diagnostics - Identified any problems and potential solutions

---

Part 0: Load Your Data

```r
# Load dataset
data <- read.csv("ai_dev_productivity.csv")

# Recode task_success as factor (same as Week 2)
data$task_success <- factor(data$task_success, levels = c(0, 1), labels = c("Failure", "Success"))

# Recreate composite productivity score from Week 2
data <- data |>
  mutate(
    productivity = scale(commits) - scale(bugs_reported) + scale(as.numeric(task_success))
  )
data$productivity <- as.numeric(data$productivity)

# Display first few rows
head(data)
```

```
##   hours_coding coffee_intake_mg distractions sleep_hours commits bugs_reported
## 1         5.99              600            1         5.8       2             1
## 2         4.72              568            2         6.9       5             3
## 3         6.30              560            1         8.9       2             0
## 4         8.05              600            7         6.3       9             5
## 5         4.53              421            6         6.9       4             0
## 6         4.53              429            1         7.1       5             0
##   ai_usage_hours cognitive_load task_success productivity
## 1           0.71            5.4      Success   -0.2873806
## 2           1.75            4.7      Success   -0.9918095
## 3           2.27            2.2      Success    0.6193646
## 4           1.40            5.9      Failure   -3.3710217
## 5           1.26            6.3      Success    1.3587390
## 6           3.06            3.9      Success    1.7284261
```

```r
# Check structure
str(data)
```

```
## 'data.frame':    500 obs. of  10 variables:
##  $ hours_coding    : num  5.99 4.72 6.3 8.05 4.53 4.53 8.16 6.53 4.06 6.09 ...
##  $ coffee_intake_mg: int  600 568 560 600 421 429 600 600 409 567 ...
##  $ distractions    : int  1 2 1 7 6 1 1 4 5 5 ...
##  $ sleep_hours     : num  5.8 6.9 8.9 6.3 6.9 7.1 8.3 3.6 6.1 7.3 ...
##  $ commits         : int  2 5 2 9 4 5 6 9 6 7 ...
##  $ bugs_reported   : int  1 3 0 5 0 0 0 3 2 0 ...
##  $ ai_usage_hours  : num  0.71 1.75 2.27 1.4 1.26 3.06 0.3 1.47 2.43 2.11 ...
##  $ cognitive_load  : num  5.4 4.7 2.2 5.9 6.3 3.9 2.2 9.1 7 5.1 ...
##  $ task_success    : Factor w/ 2 levels "Failure","Success": 2 2 2 1 2 2 2 1 1 2 ...
##  $ productivity    : num  -0.287 -0.992 0.619 -3.371 1.359 ...
```

Group Discussion:

- What is your response variable?

- Our response variable is `productivity`, a composite score built from z-scores of commits, bugs_reported, and task_success (from Week 2).

- Is it continuous (use Section A) or binary (use Section B)?

  - It is continuous (numeric), so we use Section A (Linear Regression).

- What predictors are you considering?

  - `hours_coding`, `sleep_hours`, `cognitive_load`, `ai_usage_hours`, `coffee_intake_mg`, and `distractions`. These were identified in Week 2's EDA as having meaningful correlations with productivity.

- Are all variables in the correct format (numeric, factor, etc.)?

  - Yes. All predictors are numeric, which is correct for linear regression. `task_success` was recoded to a factor but is only used inside the productivity score, not as a standalone predictor.

Check here which section you'll use:

☒ Section A: Linear Regression

☐ Section B: Logistic Regression

SECTION A: LINEAR REGRESSION

Use this section if your response variable is continuous (numeric).

Skip to Section B if you have a binary response variable.

A1: Fitting Linear Regression Models

Model 1: Simple Linear Regression

Objective: Start with a simple model using your most important predictor.

```r
# Simple linear regression: productivity predicted by hours_coding
model1 <- lm(productivity ~ hours_coding, data = data)

summary(model1)
```

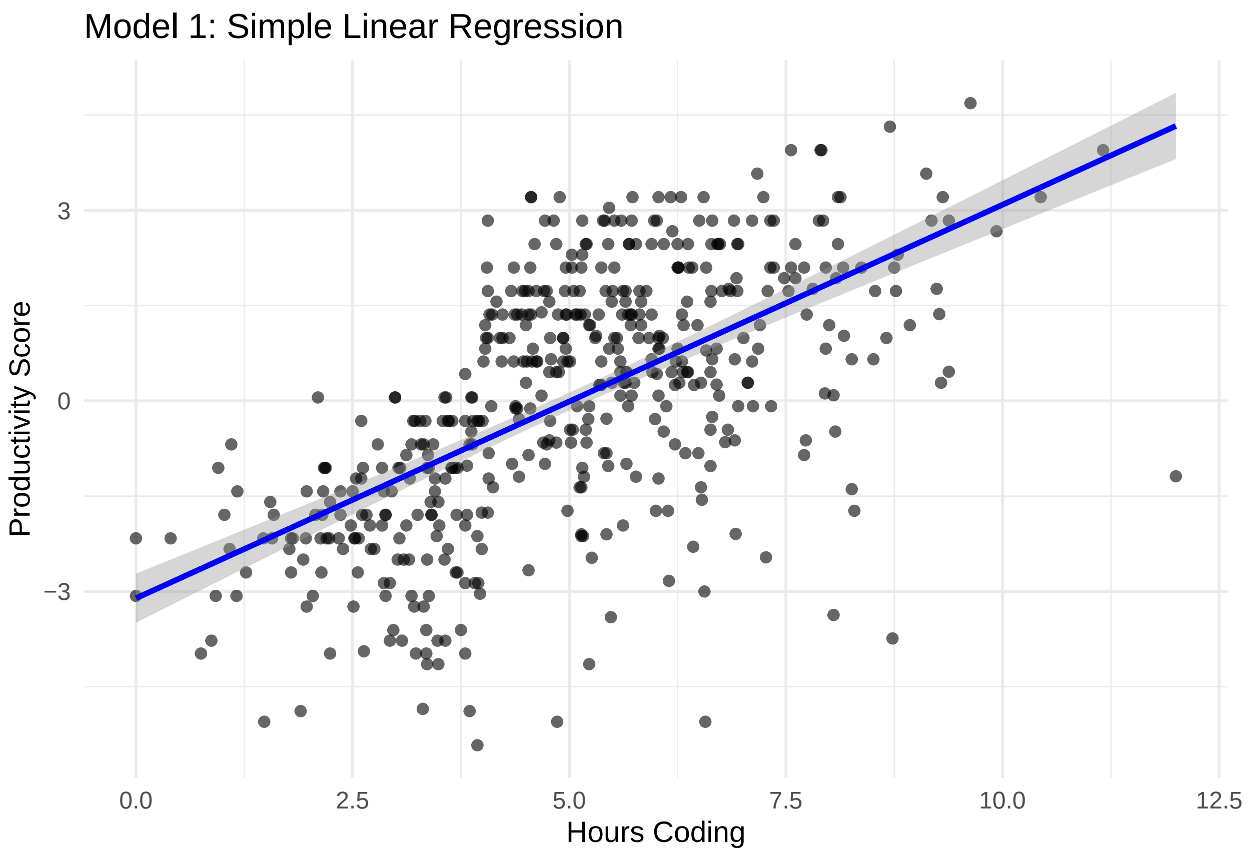
```
##
## Call:
## lm(formula = productivity ~ hours_coding, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.0419 -0.8323  0.1302  1.1164  3.4894
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.10731    0.19677  -15.79   <2e-16 ***
## hours_coding  0.61953    0.03658   16.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.591 on 498 degrees of freedom
## Multiple R-squared:  0.3655, Adjusted R-squared:  0.3642
## F-statistic: 286.9 on 1 and 498 DF,  p-value: < 2.2e-16
```

Questions to answer:

1. What is the adj R^2 value? What does it mean?
   - **Your answer:** The Adjusted R^2 is 0.3642, meaning hours_coding alone explains about 36.4% of the variance in productivity. This is a moderate fit — hours_coding is a meaningful predictor on its own, but a lot of variance remains unexplained.
2. How do you interpret the slope coefficient?
   - **Your answer:** The slope for hours_coding is 0.6195. This means that for each additional hour spent coding, productivity score increases by about 0.62 points on average. The relationship is statistically significant ($p < 2e\text{-}16$).

Visualize the model:

```r
ggplot(data, aes(x = hours_coding, y = productivity)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = TRUE, color = "blue") +
  labs(title = "Model 1: Simple Linear Regression",
       x = "Hours Coding",
       y = "Productivity Score") +
  theme_minimal()
```



Model 2: Multiple Linear Regression (Main Effects)

Objective: Add more predictors to improve the model.

```r
# Multiple regression with top 4 predictors from Week 2 EDA
model2 <- lm(productivity ~ hours_coding + sleep_hours +
               cognitive_load + ai_usage_hours, data = data)

summary(model2)
```

```
##
## Call:
## lm(formula = productivity ~ hours_coding + sleep_hours + cognitive_load +
##     ai_usage_hours, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9361 -0.8962  0.0958  0.9834  4.1136
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -4.95818    0.68177  -7.273 1.39e-12 ***
## hours_coding    0.69416    0.04159  16.689  < 2e-16 ***
## sleep_hours     0.30061    0.06719   4.474 9.52e-06 ***
## cognitive_load -0.06885    0.05253  -1.311  0.19055
## ai_usage_hours -0.20564    0.07505  -2.740  0.00636 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.484 on 495 degrees of freedom
## Multiple R-squared:  0.4516, Adjusted R-squared:  0.4472
## F-statistic: 101.9 on 4 and 495 DF,  p-value: < 2.2e-16
```

Questions to answer:

1. What is the R^2 value? How does it compare to Model 1?

   - **Your answer:** The R^2 is 0.4516, up from 0.3655 in Model 1. Adding sleep_hours, cognitive_load, and ai_usage_hours explains about 8.6% more variance in productivity.

2. What is the Adjusted R^2 value? Why is this important?

   - **Your answer:** The Adjusted R^2 is 0.4472, up from 0.3642 in Model 1. Adjusted R^2 penalizes for extra predictors, so the fact that it still increased substantially confirms the added variables genuinely improved the model. Notably, hours_coding ($p < 2e-16$) and sleep_hours ($p = 9.52e-06$) are highly significant, ai_usage_hours is significant ($p = 0.006$), while cognitive_load is not significant ($p = 0.19$).

Model 3: Model with Interactions

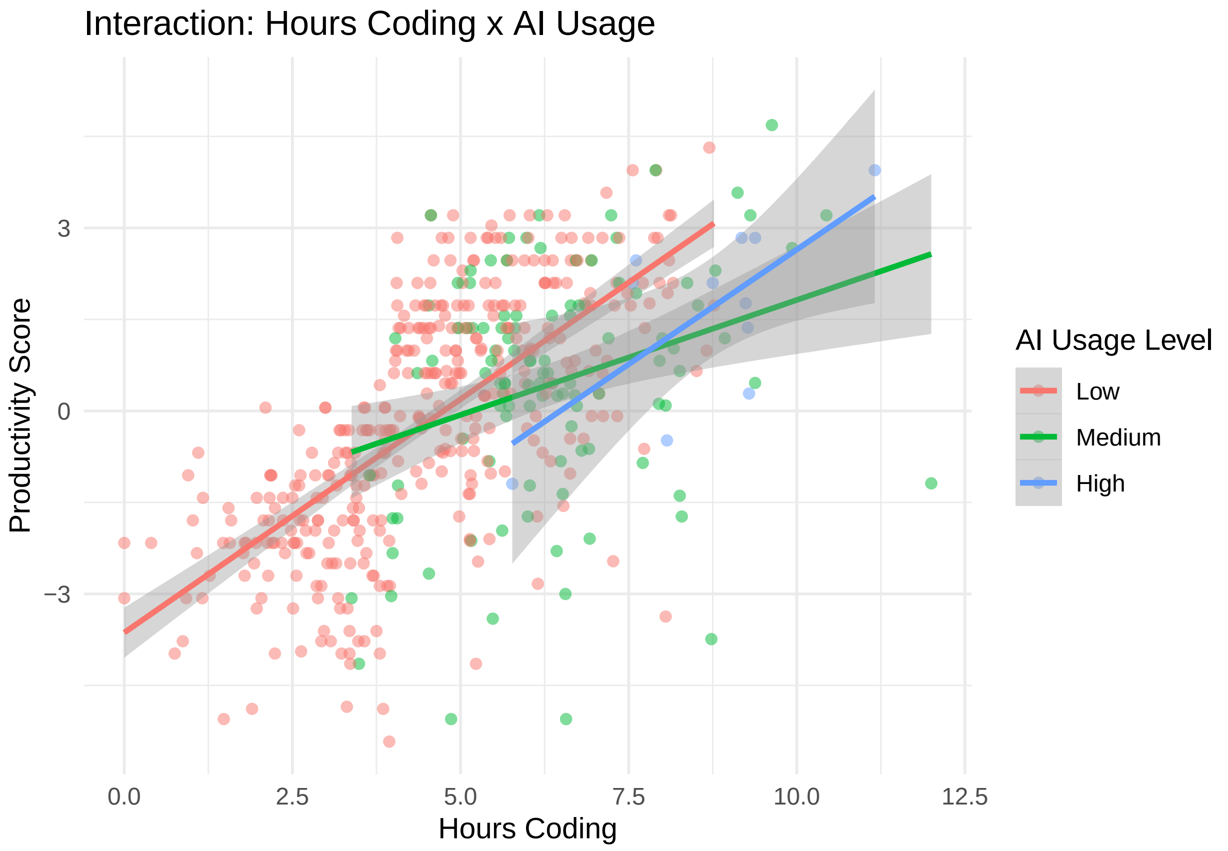
Objective: Test if relationships between variables depend on each other.

```r
# Interaction model: does the effect of hours_coding depend on AI usage?
# Week 2 showed devs who both code more AND use AI more had highest productivity
model3 <- lm(productivity ~ hours_coding * ai_usage_hours +
               sleep_hours + cognitive_load, data = data)

summary(model3)
```

```
##
## Call:
## lm(formula = productivity ~ hours_coding * ai_usage_hours + sleep_hours +
##     cognitive_load, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9738 -0.8874  0.1322  0.9752  3.9556
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -5.43256    0.72008  -7.544 2.20e-13 ***
## hours_coding                 0.76288    0.05390  14.153  < 2e-16 ***
## ai_usage_hours               0.14053    0.18891   0.744   0.4573
## sleep_hours                   0.30619    0.06704   4.567 6.25e-06 ***
## cognitive_load               -0.06191    0.05249  -1.180   0.2387
## hours_coding:ai_usage_hours -0.05276    0.02644  -1.996   0.0465 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.479 on 494 degrees of freedom
## Multiple R-squared:  0.456,  Adjusted R-squared:  0.4505
## F-statistic: 82.81 on 5 and 494 DF,  p-value: < 2.2e-16
```

Visualize the interaction:

```
data %>%
  mutate(ai_group = cut(ai_usage_hours, breaks = 3,
                       labels = c("Low", "Medium", "High"))) %>%
  ggplot(aes(x = hours_coding, y = productivity, color = ai_group)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = TRUE) +
  labs(title = "Interaction: Hours Coding x AI Usage",
       x = "Hours Coding",
       y = "Productivity Score",
       color = "AI Usage Level") +
  theme_minimal()
```



Model 4: Additional model to compare

Objective: Test if additional model can be used

```
# Full model with all 6 predictors (including coffee and distractions)
model4 <- lm(productivity ~ hours_coding + sleep_hours +
               cognitive_load + ai_usage_hours +
               coffee_intake_mg + distractions, data = data)

summary(model4)
```

```
##
## Call:
## lm(formula = productivity ~ hours_coding + sleep_hours + cognitive_load +
##     ai_usage_hours + coffee_intake_mg + distractions, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.0813 -0.8700  0.0604  1.0484  3.6793
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -6.311111   0.763958  -8.261 1.34e-15 ***
## hours_coding      0.383738   0.079737   4.813 1.98e-06 ***
## sleep_hours       0.390075   0.078232   4.986 8.55e-07 ***
## cognitive_load    0.012298   0.066591   0.185   0.8536
## ai_usage_hours   -0.167196   0.073995  -2.260   0.0243 *
## coffee_intake_mg  0.004539   0.001011   4.488 8.97e-06 ***
## distractions     -0.080559   0.050253  -1.603   0.1096
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.453 on 493 degrees of freedom
## Multiple R-squared:  0.4761, Adjusted R-squared:  0.4697
## F-statistic: 74.67 on 6 and 493 DF,  p-value: < 2.2e-16
```

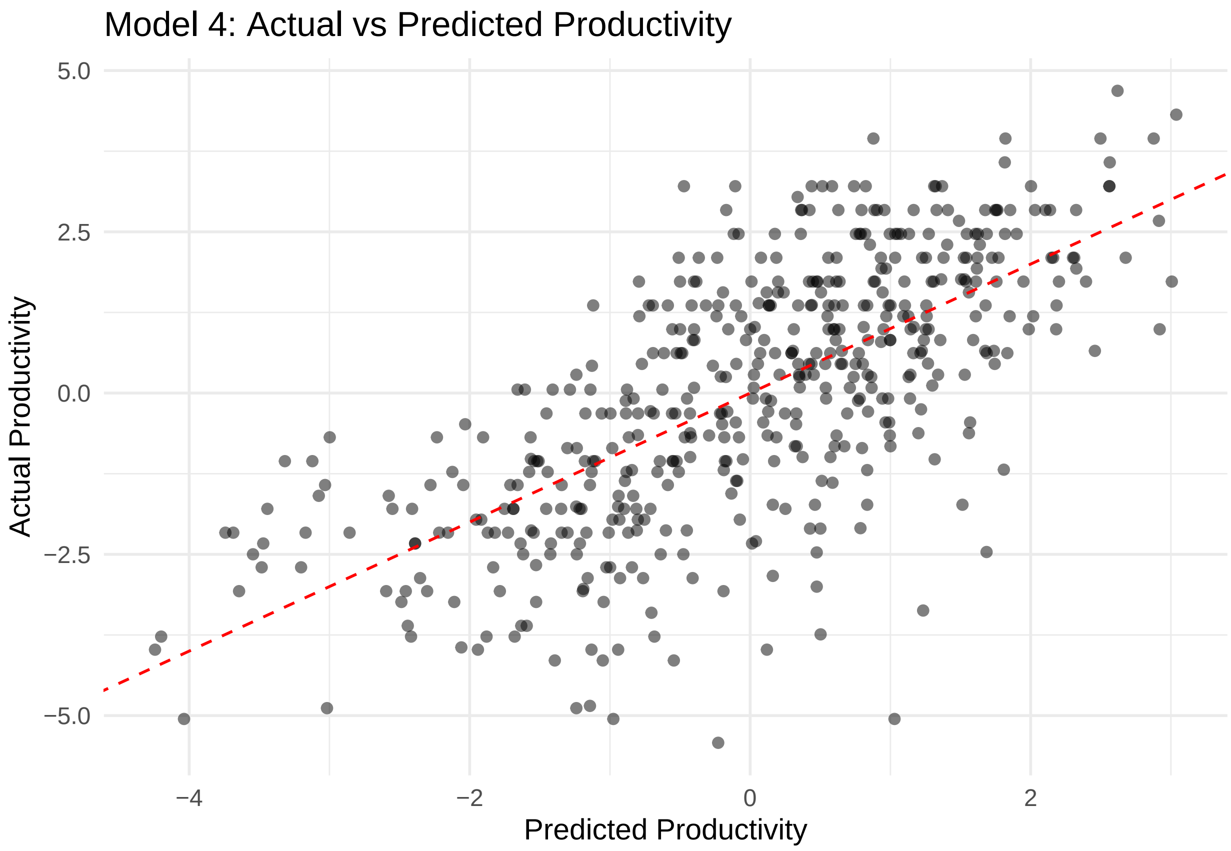
```
# Check multicollinearity with VIF
vif(model4)
```

```
##     hours_coding      sleep_hours   cognitive_load   ai_usage_hours
##         5.697581         3.066172         3.660239         1.524883
## coffee_intake_mg     distractions
##         4.897596         1.681285
```

VIF interpretation: Most VIF values are below 5, suggesting acceptable levels of multicollinearity. However, `hours_coding` (5.70) slightly exceeds the common threshold of 5, and `coffee_intake_mg` (4.90) is close to it. This indicates moderate correlation between these predictors and the others, but since no VIF exceeds 10 (the more conservative threshold), multicollinearity is not severe enough to warrant removing predictors from the model.

Visualize the model:

```r
# Compare actual vs predicted values for the full model
model4_aug <- augment(model4)

ggplot(model4_aug, aes(x = .fitted, y = productivity)) +
  geom_point(alpha = 0.5) +
  geom_abline(slope = 1, intercept = 0,
              color = "red", linetype = "dashed") +
  labs(title = "Model 4: Actual vs Predicted Productivity",
       x = "Predicted Productivity",
       y = "Actual Productivity") +
  theme_minimal()
```



A2: Model Selection (Linear)

```r
# Create comparison using broom::glance
 library(broom)
 bind_rows(
   glance(model1) %>% mutate(model = "Model 1"),
   glance(model2) %>% mutate(model = "Model 2"),
   glance(model3) %>% mutate(model = "Model 3"),
   glance(model4) %>% mutate(model = "Model 4")
 ) %>%
   select(model, r.squared, adj.r.squared, AIC)
```

```
## # A tibble: 4 x 4
##   model   r.squared adj.r.squared   AIC
##   <chr>       <dbl>         <dbl> <dbl>
## 1 Model 1     0.366         0.364 1887.
## 2 Model 2     0.452         0.447 1820.
## 3 Model 3     0.456         0.450 1818.
## 4 Model 4     0.476         0.470 1802.
```

Model Selection Table:

Criterion	Best Model	Value
Highest Adj. R^2	Model 4	0.4697183
R^2	Model 4	0.47609
AIC	Model 4	1801.57

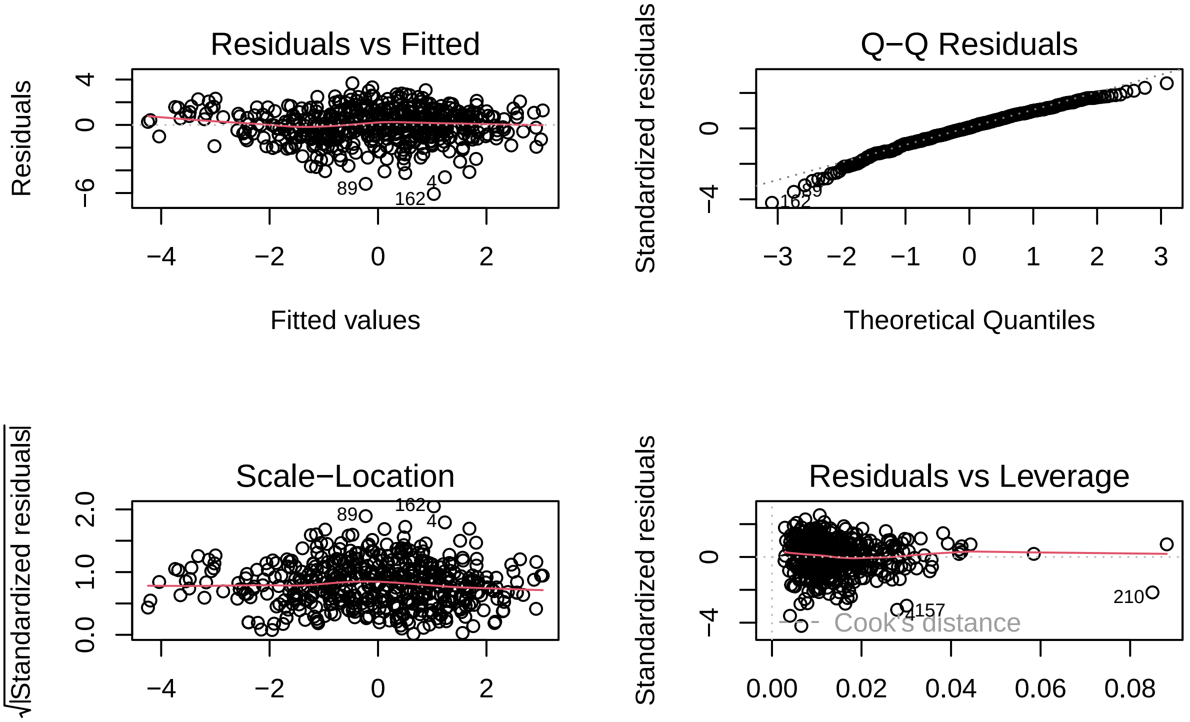
Your chosen model and justification: Model 4 is the preferred model because it has the highest R^2 and adjusted R^2, meaning it explains the greatest proportion of variability in the response variable. Additionally, it has the lowest AIC, indicating the best balance between model fit and complexity. Since it performs best across all comparison metrics, Model 4 is the strongest overall model.

A3: Model Diagnostics (Linear)

The four key assumptions: 1. **Linearity** 2. **Independence** 3. **Normality** of residuals 4. **Equal Variance**

All diagnostic plots at once:

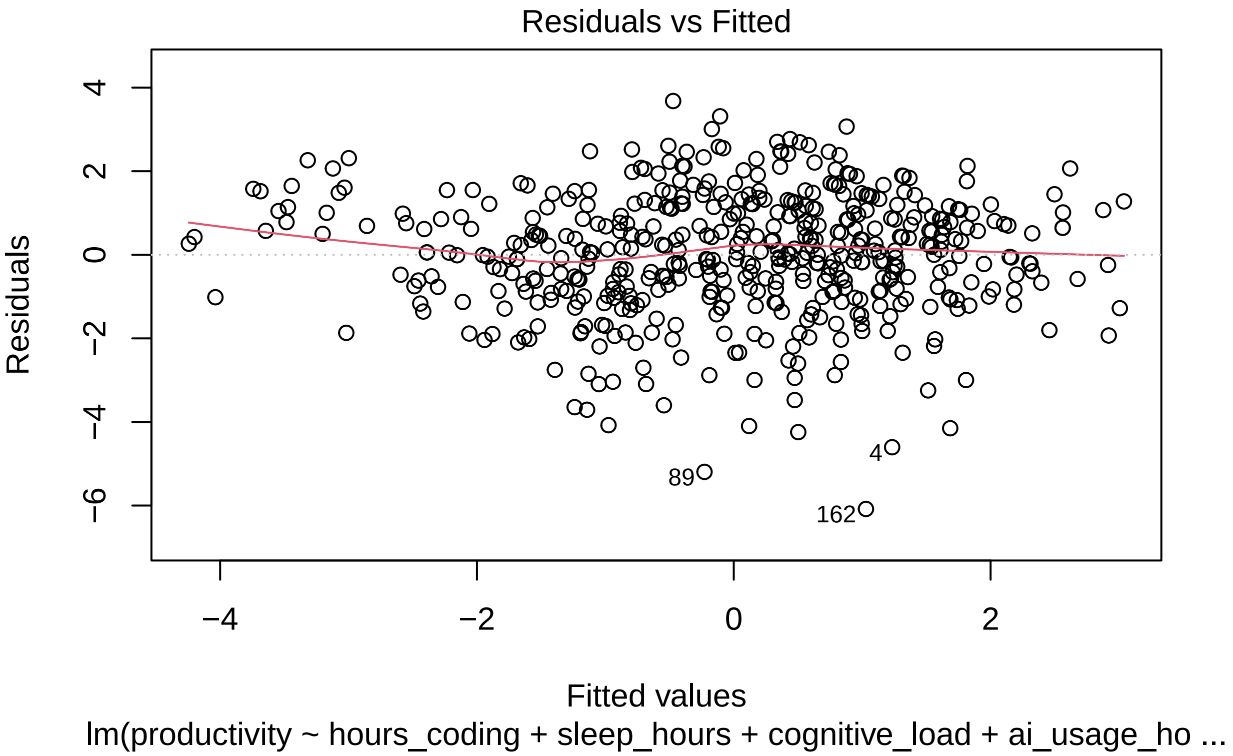
```
# Create all 4 diagnostic plots
par(mfrow = c(2, 2))
plot(model4)  # Replace with your chosen model
```



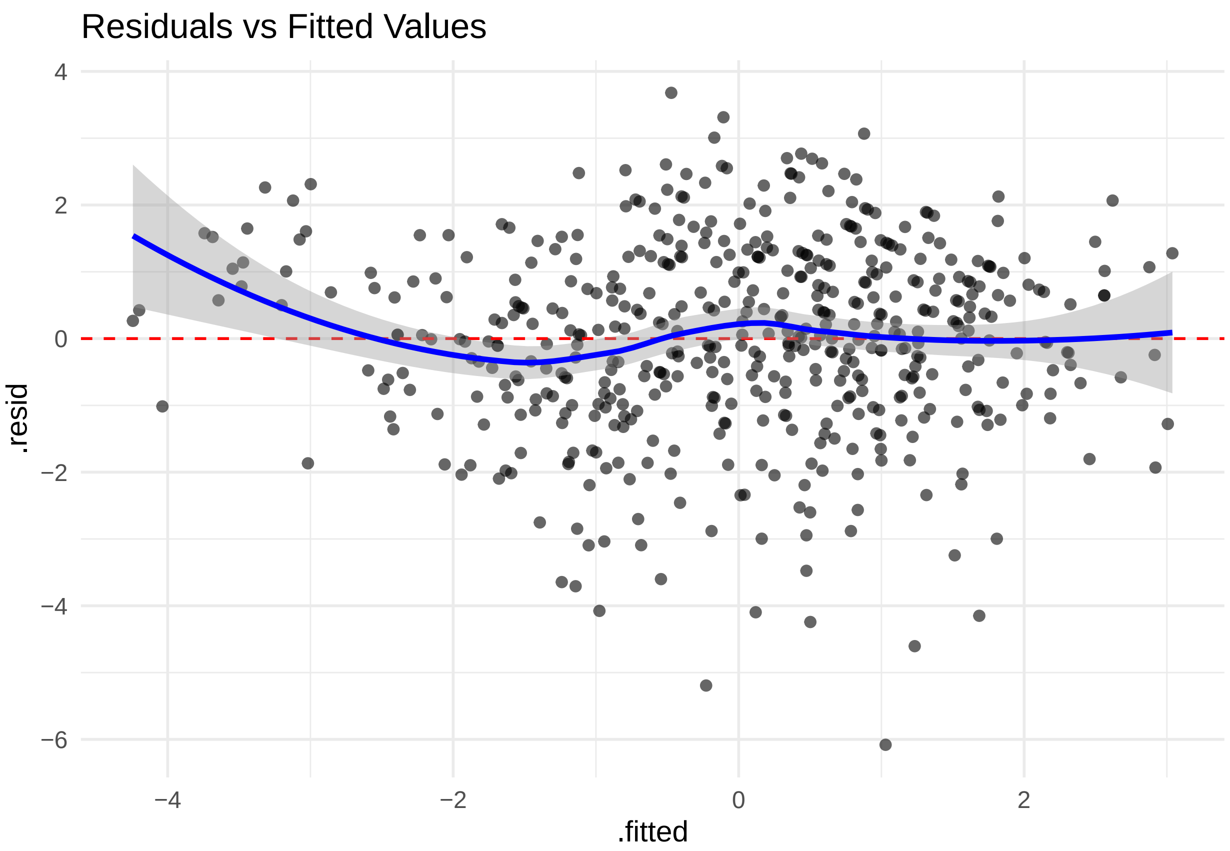
```r
par(mfrow = c(1, 1))
```

Diagnostic Plot 1: Residuals vs Fitted

```r
plot(model4, which = 1)
```



```
# Or using ggplot2
 augment(model4) %>%
   ggplot(aes(x = .fitted, y = .resid)) +
   geom_point(alpha = 0.6) +
   geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
   geom_smooth(se = TRUE, color = "blue") +
   labs(title = "Residuals vs Fitted Values") +
   theme_minimal()
```



What to look for:

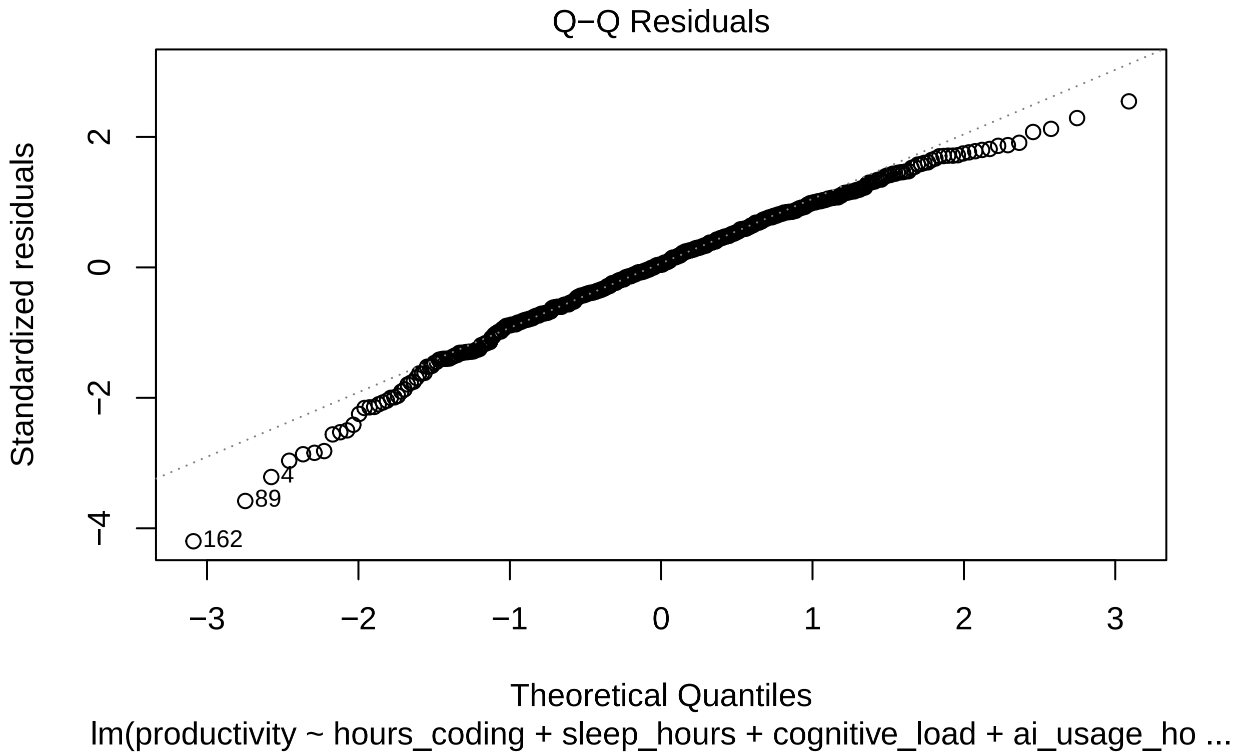
- **Good:** Random scatter around zero

- **Bad:** Curved pattern (non-linearity), funnel shape (unequal variance)

Your observations: The residuals are generally centered around zero, but there is a slight curved pattern in the smooth line. This suggests a slight violation of the linearity assumption. However, there is no strong funnel shape, so the variance of the residuals appears relatively constant. Overall, the model assumptions are mostly reasonable, though there is some slight non-linearity.

Diagnostic Plot 2: Normal Q-Q Plot

```
plot(model4, which = 2)
```



Your observations: **Most of the points closely follow the straight reference line, indicating that the residuals are about normally distributed. However, there are slight deviations in the lower tail, where a few points fall below the line. This suggests mild departures from normality, probably due to a few outliers. Overall, the normality assumption appears reasonably satisfied.**

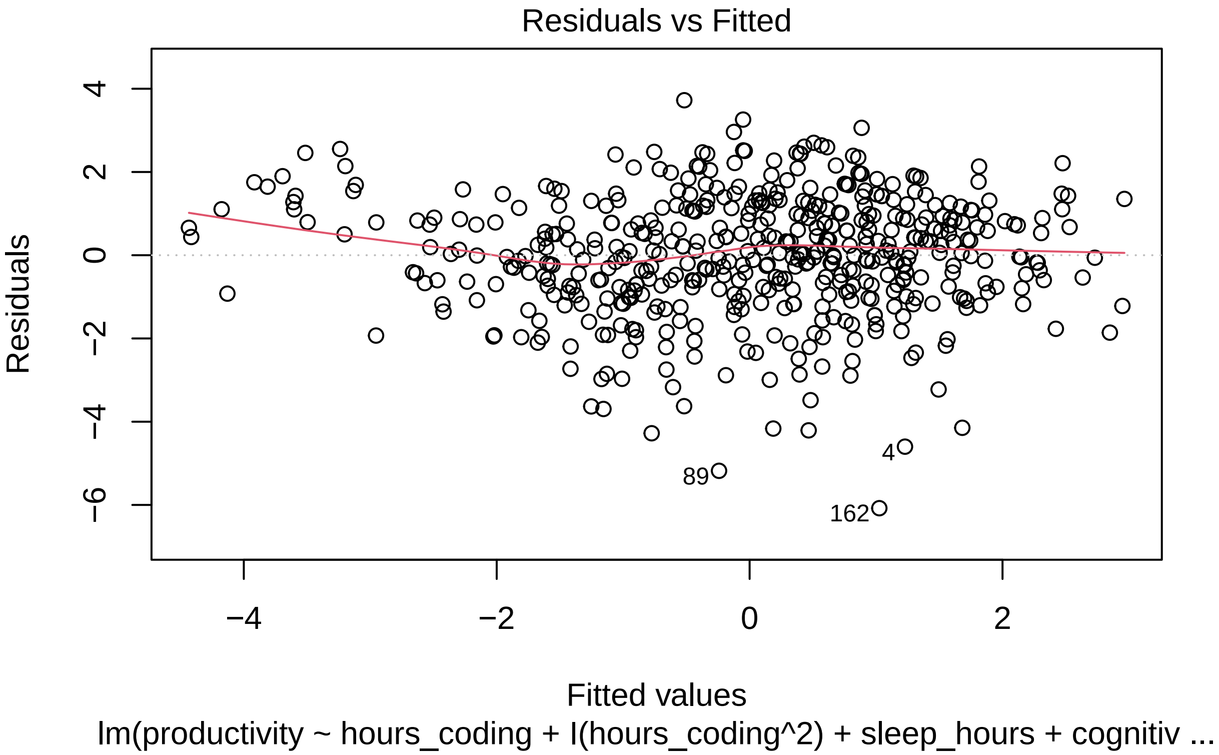
A4: Addressing Issues (Linear)

If you found non-linearity:

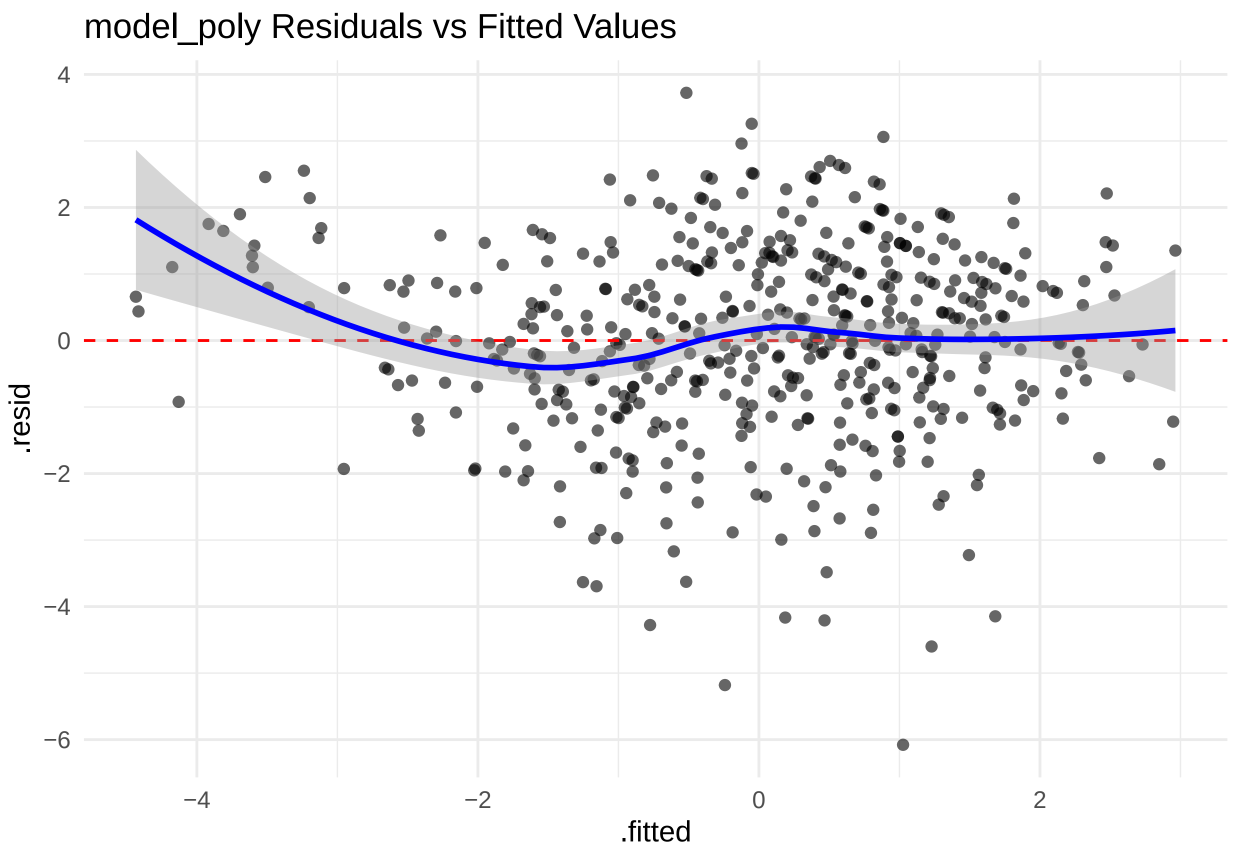
```
# Add polynomial term
model_poly <- lm(productivity ~ hours_coding + I(hours_coding^2) + sleep_hours + cognitive_load + ai_us
# Or log transformation
# model_log <- lm(log(productivity) ~ hours_coding * ai_usage_hours +
#                sleep_hours + cognitive_load, data = data)

# Create comparison using broom::glance
library(broom)
bind_rows(
  glance(model_poly) %>% mutate(model = "Model poly"),
  glance(model4) %>% mutate(model = "Model 4")
) %>%
  select(model, r.squared, adj.r.squared, AIC)
```

```
## # A tibble: 2 x 4
##   model      r.squared adj.r.squared   AIC
##   <chr>          <dbl>         <dbl> <dbl>
## 1 Model poly     0.478         0.470 1802.
## 2 Model 4        0.476         0.470 1802.
```

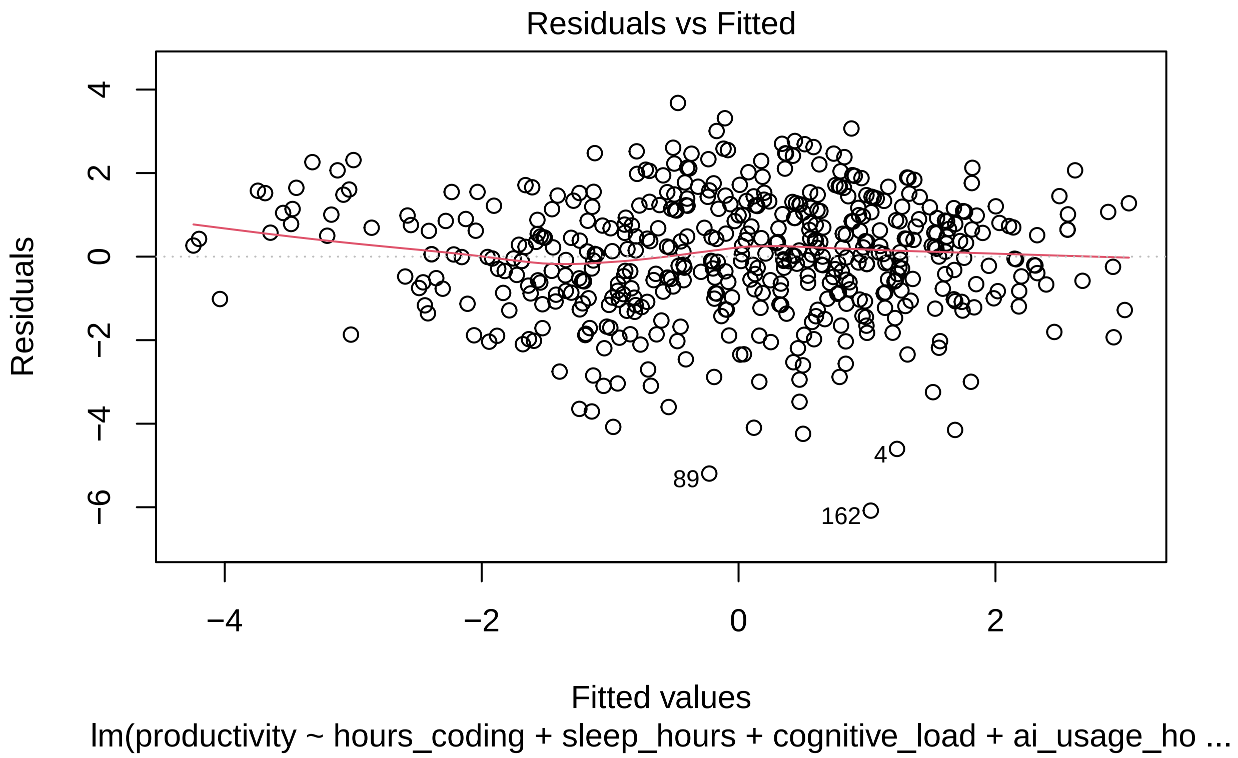
```
# plot model_poly
plot(model_poly, which = 1)
```



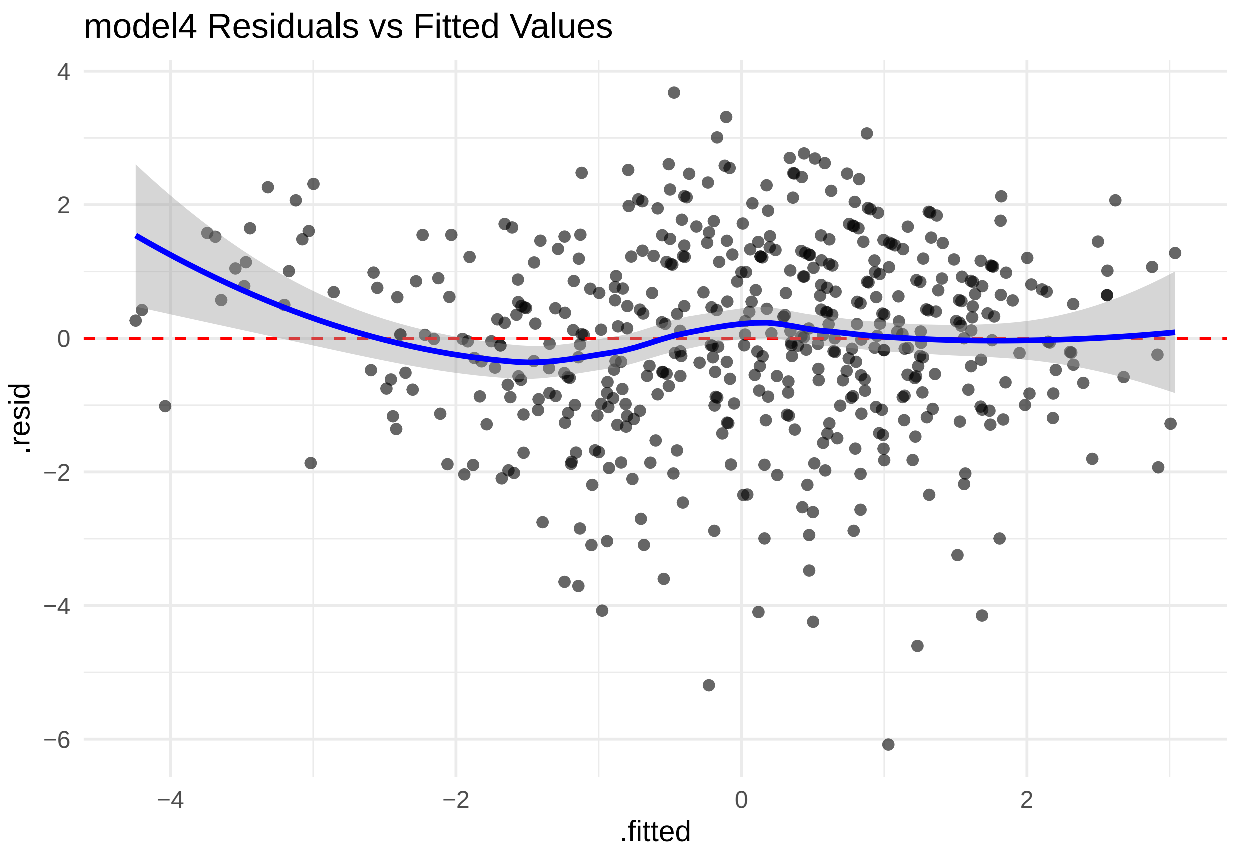
```
augment(model_poly) %>%
  ggplot(aes(x = .fitted, y = .resid)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  geom_smooth(se = TRUE, color = "blue") +
  labs(title = "model_poly Residuals vs Fitted Values") +
  theme_minimal()
```



```r
#model 4
plot(model4, which = 1)
```



```r
augment(model4) %>%
  ggplot(aes(x = .fitted, y = .resid)) +
  geom_point(alpha = 0.6) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  geom_smooth(se = TRUE, color = "blue") +
  labs(title = "model4 Residuals vs Fitted Values") +
  theme_minimal()
```



If you found unequal variance:

```
# Log transformation often helps
# model_log_y <- lm(log(response) ~ predictor1 + predictor2, data = data)
```

A5: Final Linear Model

```
# Your final model
# Model 4 was selected as the final model. While we tested adding a polynomial term for hours_coding to
 final_model <- lm(productivity ~ hours_coding + sleep_hours +
                cognitive_load + ai_usage_hours +
                coffee_intake_mg + distractions, data = data)

 summary(final_model)
```

```
##
## Call:
## lm(formula = productivity ~ hours_coding + sleep_hours + cognitive_load +
##     ai_usage_hours + coffee_intake_mg + distractions, data = data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.0813 -0.8700  0.0604  1.0484  3.6793
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -6.311111   0.763958  -8.261 1.34e-15 ***
## hours_coding      0.383738   0.079737   4.813 1.98e-06 ***
## sleep_hours       0.390075   0.078232   4.986 8.55e-07 ***
## cognitive_load    0.012298   0.066591   0.185   0.8536
## ai_usage_hours   -0.167196   0.073995  -2.260   0.0243 *
## coffee_intake_mg  0.004539   0.001011   4.488 8.97e-06 ***
## distractions     -0.080559   0.050253  -1.603   0.1096
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.453 on 493 degrees of freedom
## Multiple R-squared:  0.4761, Adjusted R-squared:  0.4697
## F-statistic: 74.67 on 6 and 493 DF,  p-value: < 2.2e-16
```

Model equation:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ...$$

Your equation:

$$\widehat{productivity} = -6.311 + 0.384(hours\_coding) + 0.390(sleep\_hours)$$
$$+0.012(cognitive\_load) - 0.167(ai\_usage\_hours)$$
$$+0.00454(coffee\_intake\_mg) - 0.081(distractions)$$

Performance:

- R² = 0.4761

- Adjusted R² = 0.4697

- RMSE = 1.453

- AIC = 1801.57

Diagnostics checklist:

[KINDA] Linearity

I would say the model is only violating the assumption of linearity slightly.

[YES] Normality

[YES] Independence

[YES] Equal variance

Checklist for Today

Before you leave, make sure you have:

[YES] Identified whether you're using linear or logistic regression

[YES] Fitted at least 2-3 different models

[YES] Compared models using appropriate criteria

[YES] Selected your best model with clear justification

[YES] Created all relevant diagnostic plots

[YES] Checked and interpreted diagnostic results

[YES] Calculated performance metrics (R^2/RMSE)

[YES] Addressed any major issues (or documented why you can't)

[YES] Written out your final model equation

[YES] Interpreted at least 2 key coefficients/odds ratios

Resources and Tips

For Linear Regression:

Interpreting Coefficients: - Continuous predictor: "A one-unit increase in X is associated with a -unit change in Y" - Categorical predictor: "Compared to [reference], this category has units higher/lower Y" - Log-transformed Y: "A one-unit increase in X is associated with approximately 100× % change in Y"

Model Selection: - Don't just pick highest R^2! - Use Adjusted R^2 for comparing models with different numbers of predictors - Lower AIC/BIC is better - Consider interpretability

For Logistic Regression:

Interpreting Odds Ratios: - OR = 1: No association - OR > 1: Positive association (predictor increases odds of success) - OR < 1: Negative association (predictor decreases odds of success) - Example: OR = 2.5 means "the odds of success are 2.5 times higher"

Model Selection: - Lower AIC is better

Common Issues: - Separation: Some predictors perfectly predict outcome → use penalized regression

- Non-linearity in logit: Add polynomial terms or categorize continuous predictors

- Poor calibration: Model may discriminate well but probabilities are off

---

Submission

Submit the following to Blackboard:

1. This R Markdown file (`.Rmd`)
2. Knitted PDF
3. Your dataset
4. Any additional R scripts if needed

One submission per group.

Individual submission.

Group Information

Name	BUID	Present/Absent
Hibak Hussen	U15515562	Present
Ian Sabia	U33871576	Present
Muze Ren	U21890514	Present
Yunhao Zhou	U18926707	Present

Which section did you use?

- ☒ Section A: Linear Regression
- ☐ Section B: Logistic Regression

Group Notes:

(Use this space for any additional notes or decisions made during the lab)