# p8105_hw6_al4925

Angela Lin

2025-12-03

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   4.0.0     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(modelr)
library(ggplot2)
```

### Problem 1

```
df = read_csv("homicide-data.csv") |>
  janitor::clean_names() |>
  mutate(city_state = paste(city, state, sep = ", "),
         solved = ifelse(disposition == "Closed by arrest", 1, 0),
         victim_age = as.numeric(victim_age)) |>
  filter(!city_state %in% c("Dallas, TX", "Phoenix, AZ", "Kansas City, MO", "Tulsa, AL"),
         victim_race %in% c("White", "Black"))
```

```
## Rows: 52179 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): uid, victim_last, victim_first, victim_race, victim_age, victim_sex...
## dbl (3): reported_date, lat, lon
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `victim_age = as.numeric(victim_age)`.
## Caused by warning:
## ! NAs introduced by coercion
```

```r
md_df = df |>
  filter(city_state == "Baltimore, MD")
```

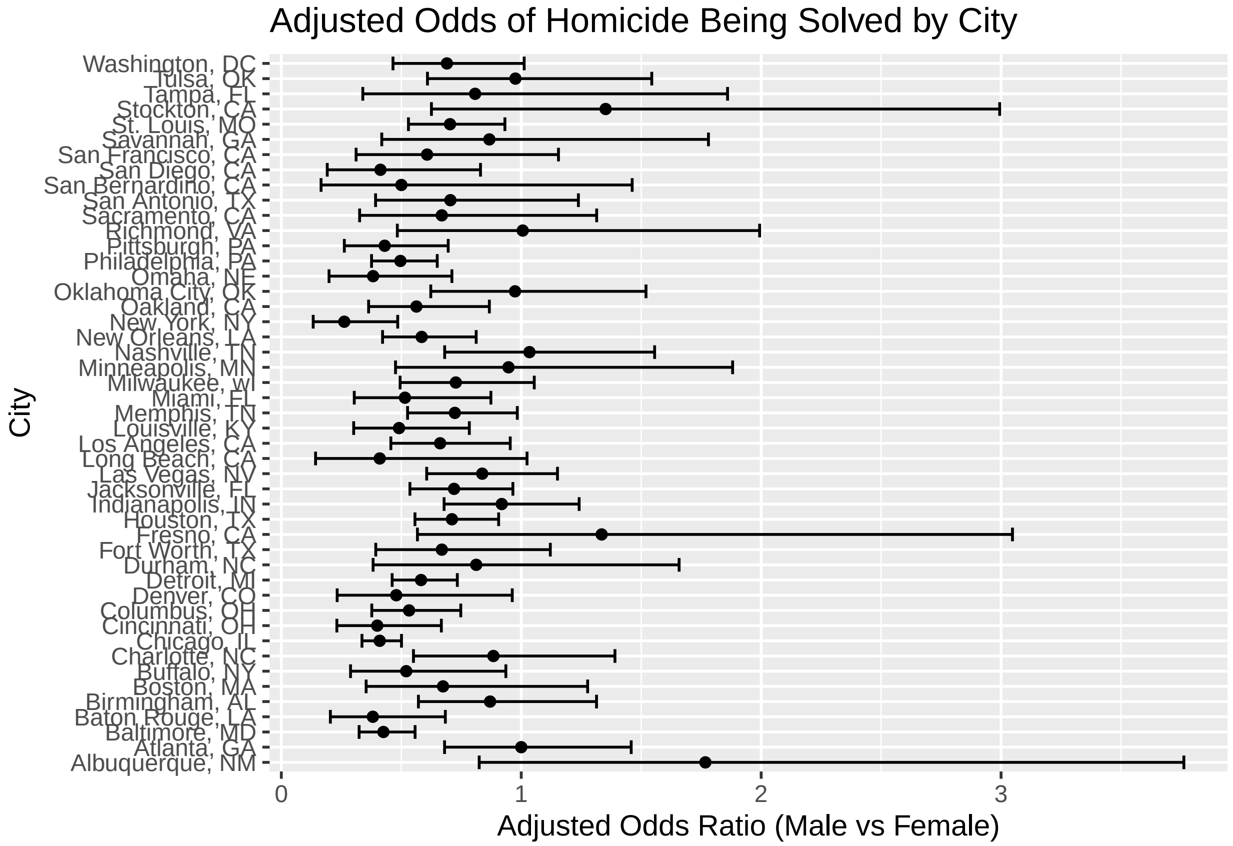
```r
fit_md = glm(solved ~ victim_age + victim_sex + victim_race,
  data = md_df,
  family = binomial())

fit_md |>
  broom::tidy() |>
  mutate(OR = exp(estimate)) |>
  select(term, log_OR = estimate, OR, p.value) |>
  knitr::kable(digits = 3)
```

| term | log_OR | OR | p.value |
|---|---:|---:|---:|
| (Intercept) | 0.310 | 1.363 | 0.070 |
| victim_age | -0.007 | 0.993 | 0.043 |
| victim_sexMale | -0.854 | 0.426 | 0.000 |
| victim_raceWhite | 0.842 | 2.320 | 0.000 |

```r
nest_city_df = df |>
  group_by(city_state) |>
  nest() |>
  mutate(fit = purrr::map(data,
      ~ glm(
        solved ~ victim_age + victim_sex + victim_race,
        data = .x,
        family = binomial())),
      tidy_fit = purrr::map(
      fit,
      ~ broom::tidy(.x, conf.int = TRUE, exponentiate = TRUE)
    )
  ) |>
  select(city_state, tidy_fit) |>
  unnest(tidy_fit) |>
   filter(term == "victim_sexMale") |>
  transmute(
    city_state,
    OR = estimate,
    CI_low = conf.low,
    CI_high = conf.high,
    p.value
  )
```

```
## Warning: There were 43 warnings in `mutate()`.
## The first warning was:
## i In argument: `tidy_fit = purrr::map(fit, ~broom::tidy(.x, conf.int = TRUE,
##   exponentiate = TRUE))`.
## i In group 1: `city_state = "Albuquerque, NM"`.
## Caused by warning:
## ! glm.fit: fitted probabilities numerically 0 or 1 occurred
## i Run `dplyr::last_dplyr_warnings()` to see the 42 remaining warnings.
```

```
ggplot(nest_city_df, aes(x = city_state, y = OR)) +
  geom_point() +
  coord_flip() +
  geom_errorbar(aes(ymin = CI_low, ymax = CI_high)) +
  labs(
    x = "City",
    y = "Adjusted Odds Ratio (Male vs Female)",
    title = "Adjusted Odds of Homicide Being Solved by City"
  )
```



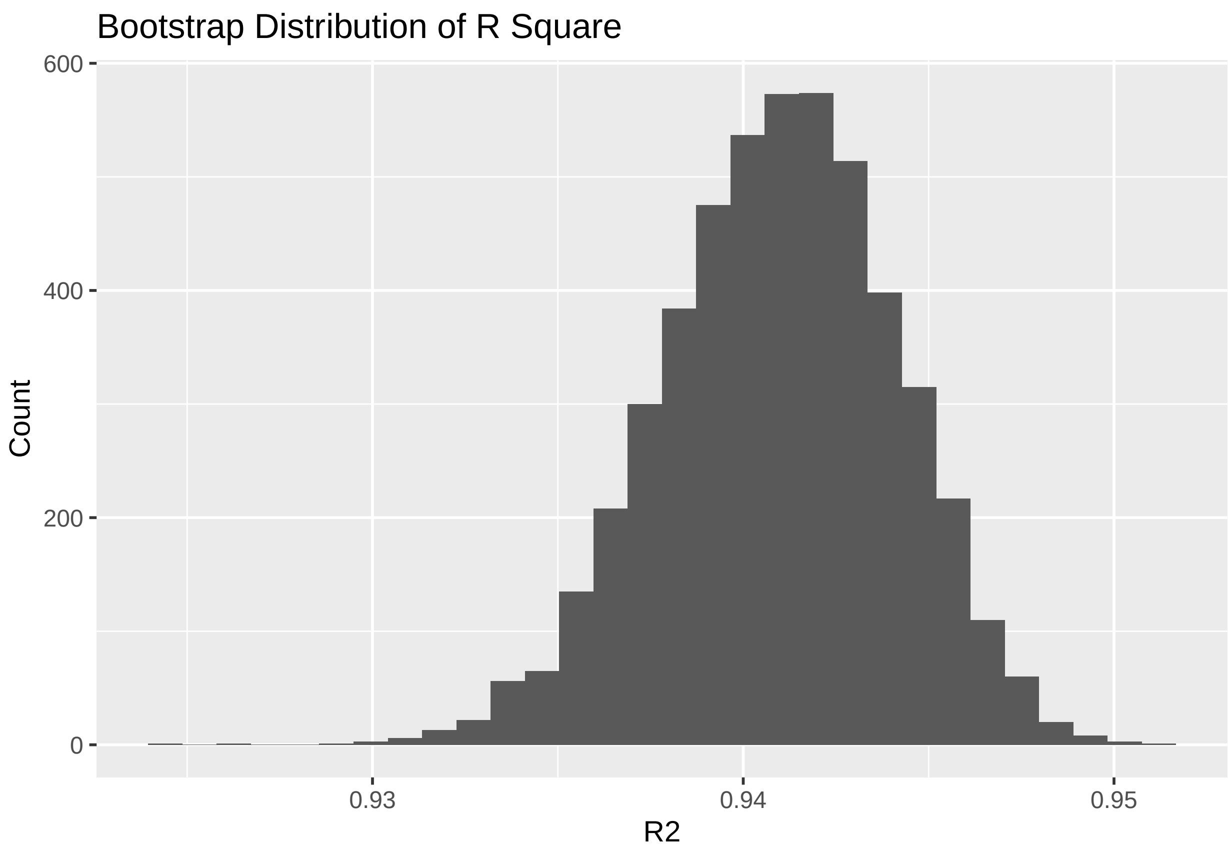
The plot shows substantial variation in the adjusted odds ratio for solving homicides comparing male to female victims across cities. In some cities, the OR is above 1, indicating that homicides involving male victims are more likely to be solved after adjusting for age and race. In others, the OR is below 1, suggesting relatively higher solve rates for cases involving female victims. The cities with wider CIs likely have fewer cases (smaller sample sizes), resulting in greater uncertainty.

**Problem 2**

```
library(p8105.datasets)
data("weather_df")
```

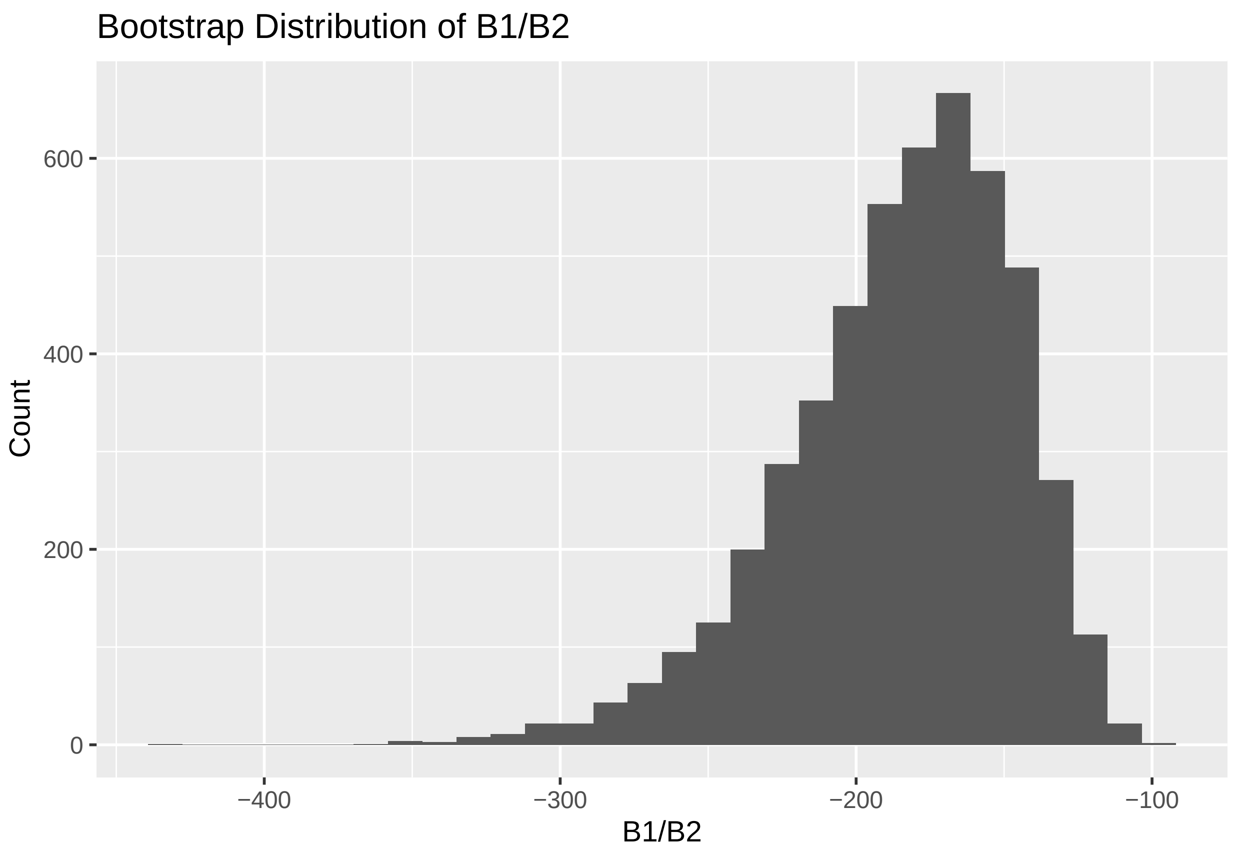
```
weather = weather_df |>
  select(tmax, tmin, prcp) |>
  drop_na()

boot_est = function(df) {
   fit = lm(tmax ~ tmin + prcp, data = df)
   r2 = broom::glance(fit)$r.squared
   coefs = broom::tidy(fit)
   b1 = coefs$estimate[coefs$term == "tmin"]
   b2 = coefs$estimate[coefs$term == "prcp"]
   ratio = b1 / b2
   tibble(
    r2 = r2,
    beta_ratio = ratio
  )
}

set.seed(1)

boot_straps =
  tibble(id = 1:5000) |>
  mutate(
    strap = map(id, ~ sample_frac(weather, replace = TRUE))
  )

bootstrap_results =
  boot_straps |>
  mutate(
    estimates = map(strap, boot_est)
  ) |>
  select(-strap) |>
  unnest(estimates)

bootstrap_results |>
  ggplot(aes(x = r2)) +
  geom_histogram() +
  labs(
    title = "Bootstrap Distribution of R Square",
    x = "R2",
    y = "Count"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



```
bootstrap_results |>
  ggplot(aes(x = beta_ratio)) +
  geom_histogram() +
  labs(
    title = "Bootstrap Distribution of B1/B2",
    x = "B1/B2",
    y = "Count"
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```



```
bootstrap_results |>
  summarize(
    r2_lower = quantile(r2, 0.025),
    r2_upper = quantile(r2, 0.975),
    beta_lower = quantile(beta_ratio, 0.025),
    beta_upper = quantile(beta_ratio, 0.975)
  )
```

```
## # A tibble: 1 x 4
##   r2_lower r2_upper beta_lower beta_upper
##      <dbl>    <dbl>      <dbl>      <dbl>
## 1    0.934    0.947      -275.      -125.
```

**Problem 3**

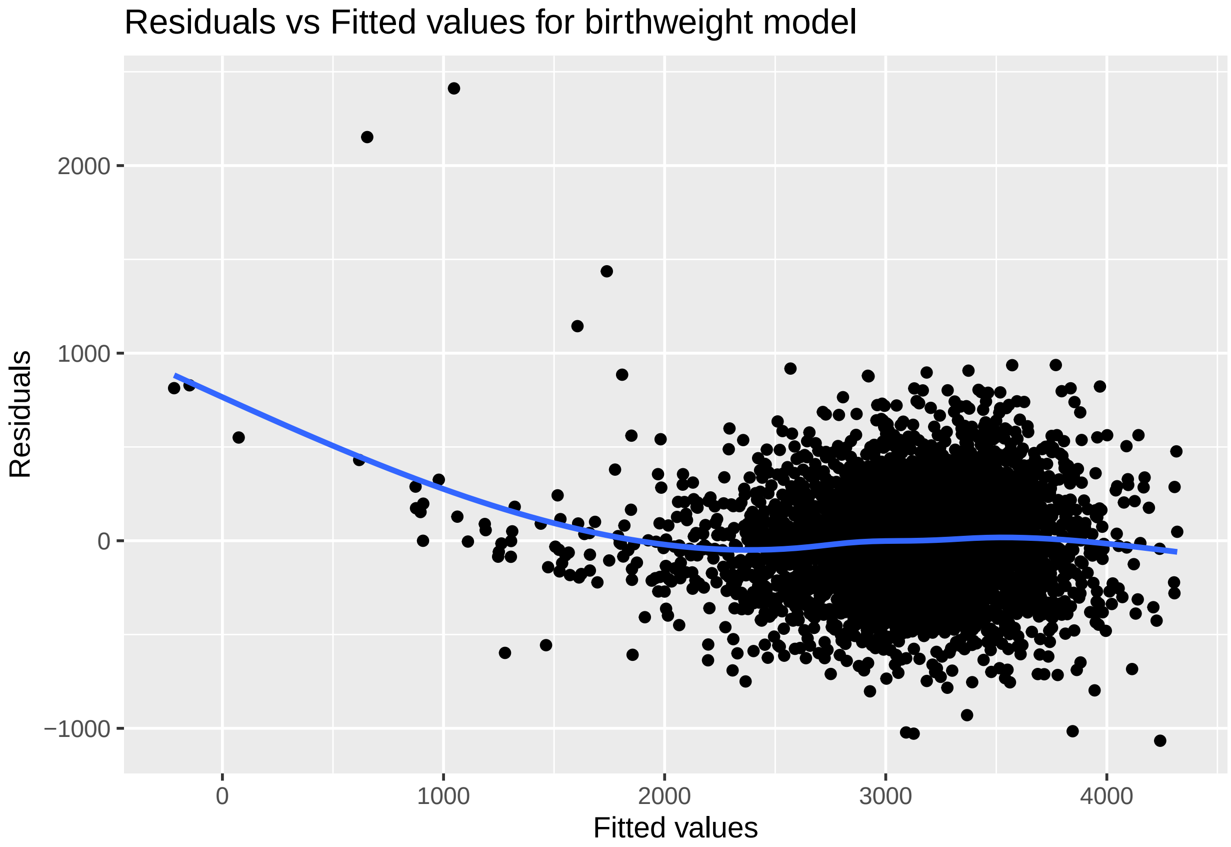
```
birthweight <- read_csv("birthweight.csv") |>
  janitor::clean_names() |>
  mutate(
    babysex = factor(babysex, levels = c(1, 2), labels = c("male", "female")),
    malform = factor(malform, levels = c(0, 1), labels = c("absent", "present")),
    mrace   = factor(mrace),
    frace   = factor(frace)
  )
```

```
## Rows: 4342 Columns: 20
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (20): babysex, bhead, blength, bwt, delwt, fincome, frace, gaweeks, malf...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

I began by cleaning the dataset and converting several categorical variables into factors, such as the baby's sex (babysex), the presence of malformations (malform), and parental race (mrace, frace). After confirming that the dataset contained no missing values, I selected a set of predictors based on both clinical intuition and prior knowledge of factors known to influence fetal growth.

```
bwt_model = lm(bwt ~ bhead + blength + gaweeks + ppbmi + wtgain +
         smoken + momage + fincome + babysex + mrace, data = birthweight)

birthweight |>
  add_predictions(bwt_model) |>
  add_residuals(bwt_model) |>
  ggplot(aes(x = pred, y = resid)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(
    x = "Fitted values",
    y = "Residuals",
    title = "Residuals vs Fitted values for birthweight model"
  )
```

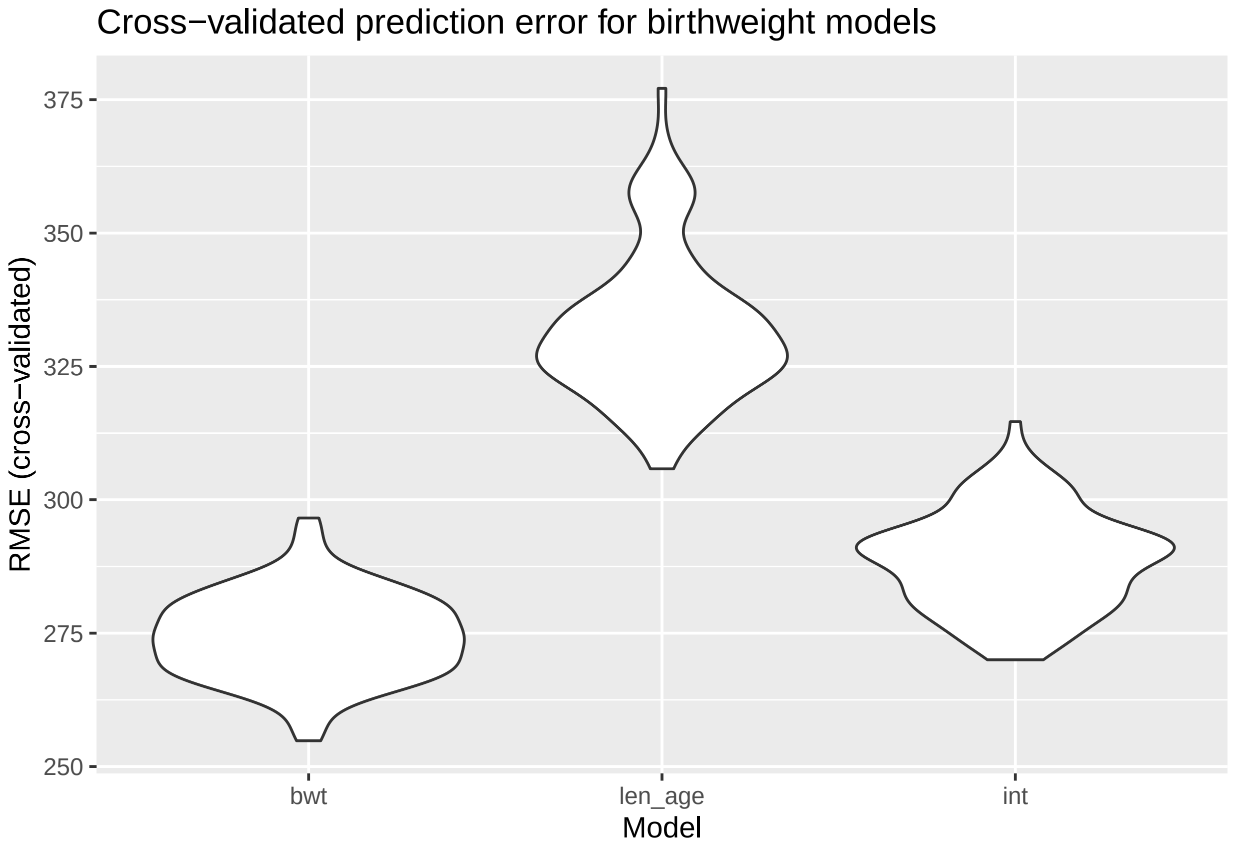
```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



My proposed model includes the every variables as my predictors. This combination reflects a biologically plausible structure: infant size, gestational duration, and maternal physiological characteristics are direct determinants of birthweight, while smoking, income, and demographic factors play indirect but well-documented roles. After fitting the model, I evaluated its assumptions by examining a residuals-versus-fitted plot.

The residuals–versus–fitted plot shows that residuals are generally centered around zero, but there is some curvature at the lower fitted values, where the model tends to underestimate birthweight. Overall, the plot indicates a mostly acceptable fit with some mild nonlinearity at the low end.

```
bwt_model_len_age =
  lm(bwt ~ blength + gaweeks, data = birthweight)
bwt_model_int =
  lm(bwt ~ bhead * blength * babysex, data = birthweight)

set.seed(1)
cv_df =
  crossv_mc(birthweight, n = 100) |>
  mutate(
    train = map(train, as_tibble),
    test  = map(test, as_tibble))

cv_df =
  cv_df |>
  mutate(
    mod_bwt = map(train,\(df) lm(bwt ~ bhead + blength + gaweeks + ppbmi + wtgain +
               smoken + momage + fincome + babysex + mrace, data = df)),
    mod_len_age = map(train, \(df) lm(bwt ~ blength + gaweeks, data = df)),
    mod_int = map(train, \(df) lm(bwt ~ bhead * blength * babysex, data = df))
    )

cv_df =
  cv_df |>
  mutate(
    rmse_bwt = map2_dbl(mod_bwt, test, \(mod, df) rmse(model = mod, data = df)),
    rmse_len_age = map2_dbl(mod_len_age, test, \(mod, df) rmse(model = mod, data = df)),
    rmse_int = map2_dbl(mod_int, test, \(mod, df) rmse(model = mod, data = df))
  )


cv_df |>
  select(starts_with("rmse_")) |>
  pivot_longer(
    everything(),
    names_to   = "model",
    values_to  = "rmse",
    names_prefix = "rmse_"
  ) |>
  mutate(model = fct_inorder(model)) |>
  ggplot(aes(x = model, y = rmse)) +
  geom_violin() +
  labs(
    x = "Model",
    y = "RMSE (cross-validated)",
    title = "Cross-validated prediction error for birthweight models"
  )
```



To assess predictive performance, I compared my model to two alternatives. The cross-validated RMSE results show clear differences in predictive performance across the three models. My proposed multivariable model (“bwt”) has the lowest RMSE values overall, indicating the best out-of-sample predictive accuracy. The simple model using only birth length and gestational age (“len_age”) performs noticeably worse, with higher and more variable RMSE. The interaction model (“int”) performs better than the simple model but still slightly worse than my proposed model, suggesting that although interactions add flexibility, they do not outperform a more comprehensive set of clinically relevant predictors. Overall, the multivariable model provides the best balance of accuracy and stability.