p8105_hw6_mc5698

2024-11-25

#Problem 1

```
#Read dataset
weather_df =
  rnoaa::meteo_pull_monitors(
    c("USW00094728"),
    var = c("PRCP", "TMIN", "TMAX"),
    date_min = "2017-01-01",
    date_max = "2017-12-31") %>%
  mutate(
    name = recode(id, USW00094728 = "CentralPark_NY"),
    tmin = tmin / 10,
    tmax = tmax / 10) %>%
  select(name, id, everything())
```

```
#Define function
set.seed(1)
bootstrap<-
  map_df(1:5000, function(i) {
  sample_df <- weather_df %>% sample_frac(size = 1, replace = TRUE)
  lm_model <- lm(tmax ~ tmin, data = sample_df)

  glance_metrics <- broom::glance(lm_model)
  tidy_metrics <- broom::tidy(lm_model)

  r_squared <- glance_metrics$r.squared
  log_beta <- log(tidy_metrics$estimate[1] * tidy_metrics$estimate[2])
  tibble(
    r_squared = r_squared,
    log_beta = log_beta)})
```

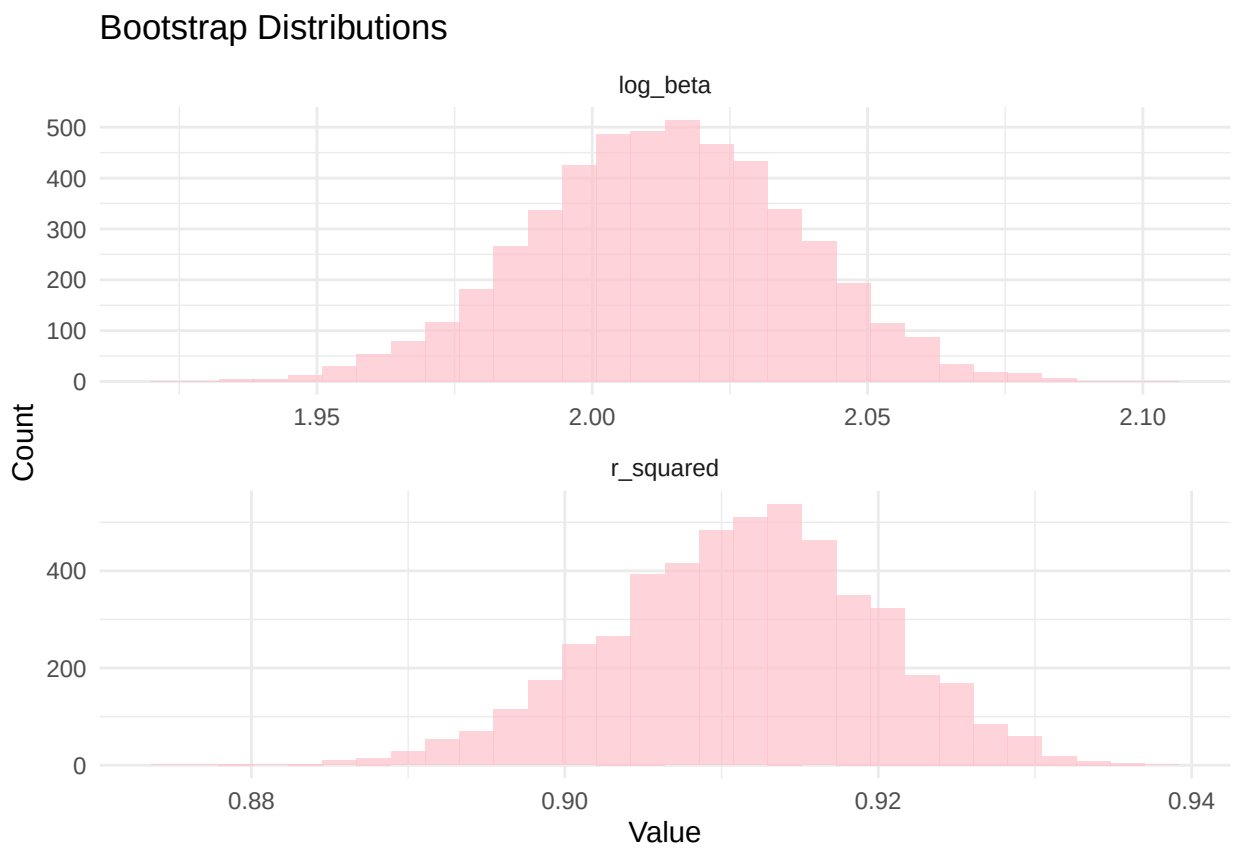
```
# Confidence Intervals
ci_r_squared <- quantile(bootstrap$r_squared, probs = c(0.025, 0.975))
ci_log_beta <- quantile(bootstrap$log_beta, probs = c(0.025, 0.975))

ci_r_squared
```

```
##      2.5%     97.5%
## 0.8936684 0.9271060
```

```
ci_log_beta
```

```
##     2.5%    97.5%
## 1.964949 2.058887
```

```r
# Plot
bootstrap %>%
  pivot_longer(everything(), names_to = "metric", values_to = "value") %>%
  ggplot(aes(x = value)) +
  geom_histogram(bins = 30, fill = "pink", alpha = 0.7) +
  facet_wrap(~ metric, scales = "free", ncol = 1) +
  theme_minimal() +
  labs(title = "Bootstrap Distributions", x = "Value", y = "Count")
```



#Problem 2

```r
# Load the data
homicide_data <- read_csv("data/homicide-data.csv")
```

```r
# Data Cleaning
omit_cities <- c("Dallas, TX", "Phoenix, AZ", "Kansas City, MO", "Tulsa, AL")

clean_data <- homicide_data %>%
  janitor::clean_names() %>%
  mutate(
    city_state = paste(city, state, sep = ", "),
    solved = ifelse(disposition == "Closed by arrest", 1, 0)
  ) %>%
  filter(!city_state %in% omit_cities,
         victim_race %in% c("White", "Black")) %>%
```

```
  mutate(victim_age = as.numeric(victim_age)) %>%
  filter(!is.na(victim_age))
```

```
# Model for Baltimore
baltimore_data <- clean_data %>%
  filter(city_state == "Baltimore, MD")

baltimore_model <- glm(solved ~ victim_age + victim_sex + victim_race,
                       data = baltimore_data, family = binomial())

baltimore_results <- broom::tidy(baltimore_model, conf.int = TRUE) %>%
  filter(term == "victim_sexMale") %>%
  mutate(
    odds_ratio = exp(estimate),
    ci_lower = exp(conf.low),
    ci_upper = exp(conf.high)) %>%
  select(odds_ratio, ci_lower, ci_upper)

baltimore_results
```

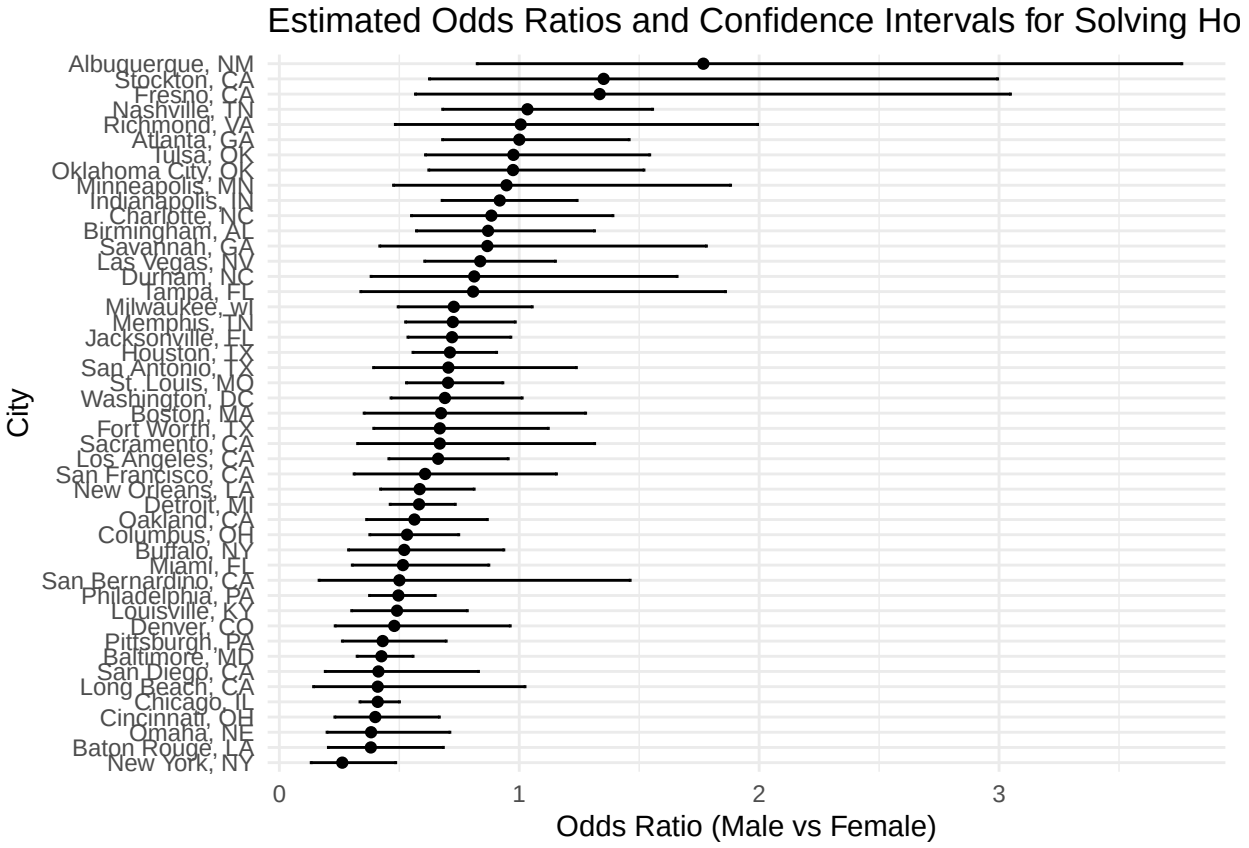
```
## # A tibble: 1 x 3
##   odds_ratio ci_lower ci_upper
##        <dbl>    <dbl>    <dbl>
## 1      0.426    0.324    0.558
```

```
#For All Cities
city_results <- clean_data %>%
  group_by(city_state) %>%
  nest() %>%
  mutate(
    model = map(data, ~ glm(solved ~ victim_age + victim_sex + victim_race,
                            data = ., family = binomial)),
    tidy_model = purrr::map(model, ~ tryCatch(tidy(.x, conf.int = TRUE),
                                      error = function(e) NULL))) %>%
  unnest(tidy_model) %>%
  filter(term == "victim_sexMale") %>%
  mutate(
    odds_ratio = exp(estimate),
    ci_lower = exp(conf.low),
    ci_upper = exp(conf.high)
  ) %>%
  select(city_state, odds_ratio, ci_lower, ci_upper)
city_results
```

```
## # A tibble: 47 x 4
## # Groups:   city_state [47]
##    city_state      odds_ratio ci_lower ci_upper
##    <chr>                <dbl>    <dbl>    <dbl>
##  1 Albuquerque, NM      1.77     0.825    3.76
##  2 Atlanta, GA          1.00     0.680    1.46
##  3 Baltimore, MD        0.426    0.324    0.558
##  4 Baton Rouge, LA      0.381    0.204    0.684
```

```
##  5 Birmingham, AL       0.870    0.571    1.31
##  6 Boston, MA           0.674    0.353    1.28
##  7 Buffalo, NY          0.521    0.288    0.936
##  8 Charlotte, NC        0.884    0.551    1.39
##  9 Chicago, IL          0.410    0.336    0.501
## 10 Cincinnati, OH       0.400    0.231    0.667
## # i 37 more rows
```

```r
# plot
city_results %>%
  arrange(odds_ratio) %>%
  ggplot(aes(x = reorder(city_state, odds_ratio), y = odds_ratio)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_lower, ymax = ci_upper), width = 0.2) +
  coord_flip() +
  labs(
    title = "Estimated Odds Ratios and Confidence Intervals for Solving Homicides by City",
    x = "City",
    y = "Odds Ratio (Male vs Female)"
  ) +
  theme_minimal()
```



The graph shows the estimated odds ratios and 95% confidence intervals for solving homicides comparing male to female victims across various U.S. cities. In most cities, such as Baltimore and New York, the ORs are below 1, suggesting that homicides involving male victims are less likely to be solved compared to female victims. Moreover, some cities like Fresno and Boston, have confidence intervals that cross 1, implying that

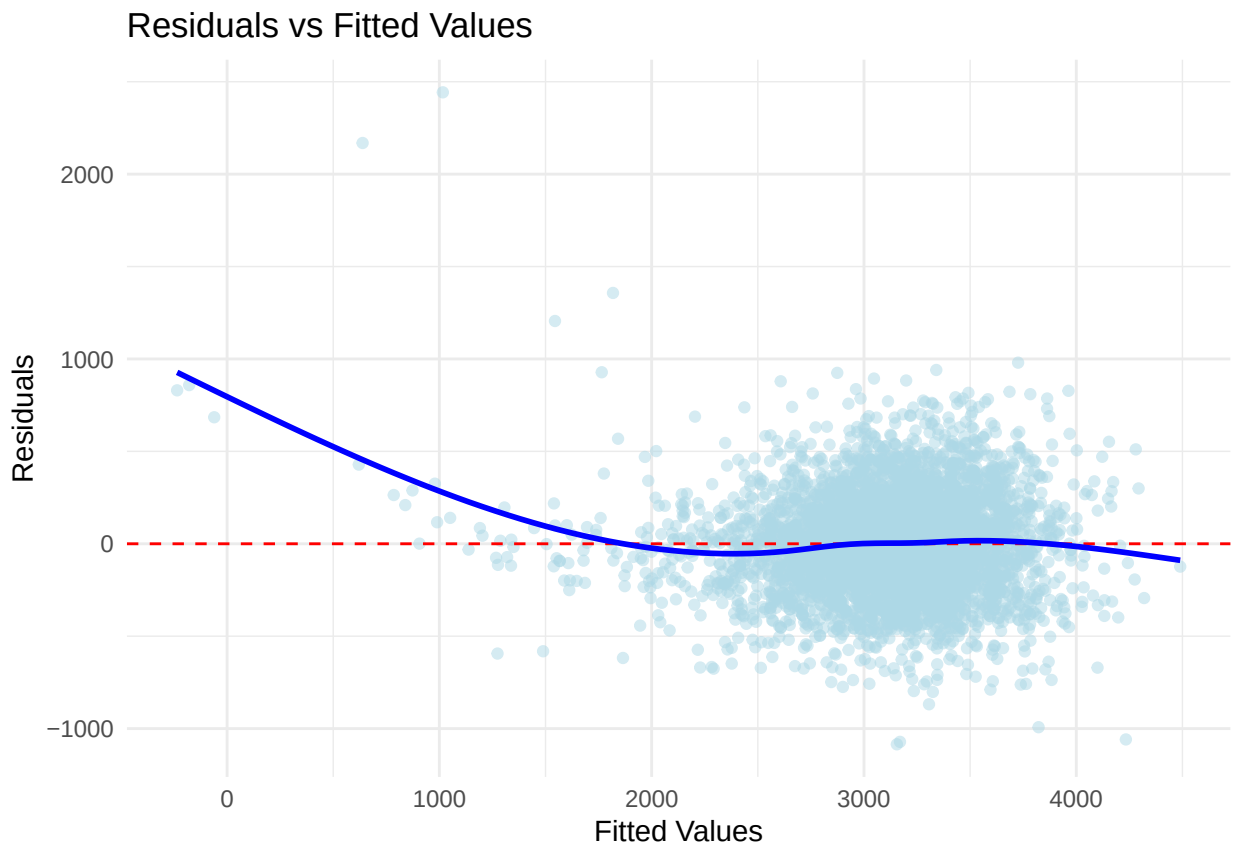
the difference in solving rates between male and female victims may not be statistically significant in those locations.

#Problem 3

```
#Read data
birthweight_data <- read_csv("data/birthweight.csv")

# Data Cleaning
birthweight_clean <- birthweight_data %>%
  janitor::clean_names() %>%
  mutate(babysex = factor(babysex, levels = c(1, 2), labels =    c("Male","Female")),
         frace = factor(frace, levels = c(1, 2, 3, 4, 8, 9), labels = c("White", "Black", "Asian", "Pue
         mrace = factor(mrace, levels = c(1, 2, 3, 4, 8), labels = c("White", "Black", "Asian", "Puerto
         malform = factor(malform, levels = c(0, 1), labels = c("Absent", "Present"))) %>%
  drop_na()
```

```
# Fit a model
birthweight_model <- lm(bwt ~ gaweeks + wtgain + blength + bhead + babysex + delwt + momage + frace + s

# Plot residuals against fitted values
birthweight_data <- birthweight_data %>%
  add_predictions(birthweight_model) %>%
  add_residuals(birthweight_model)

ggplot(birthweight_data, aes(x = pred, y = resid)) +
  geom_point(alpha = 0.5, color = "lightblue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  geom_smooth(se = FALSE, color = "blue") +
  labs(title = "Residuals vs Fitted Values",
       x = "Fitted Values",
       y = "Residuals") +
  theme_minimal()
```

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




```
# Model 1
model1 <- lm(bwt ~ blength + gaweeks, data = birthweight_data)

# Model 2
model2 <- lm(bwt ~ bhead * blength * babysex, data = birthweight_data)
```

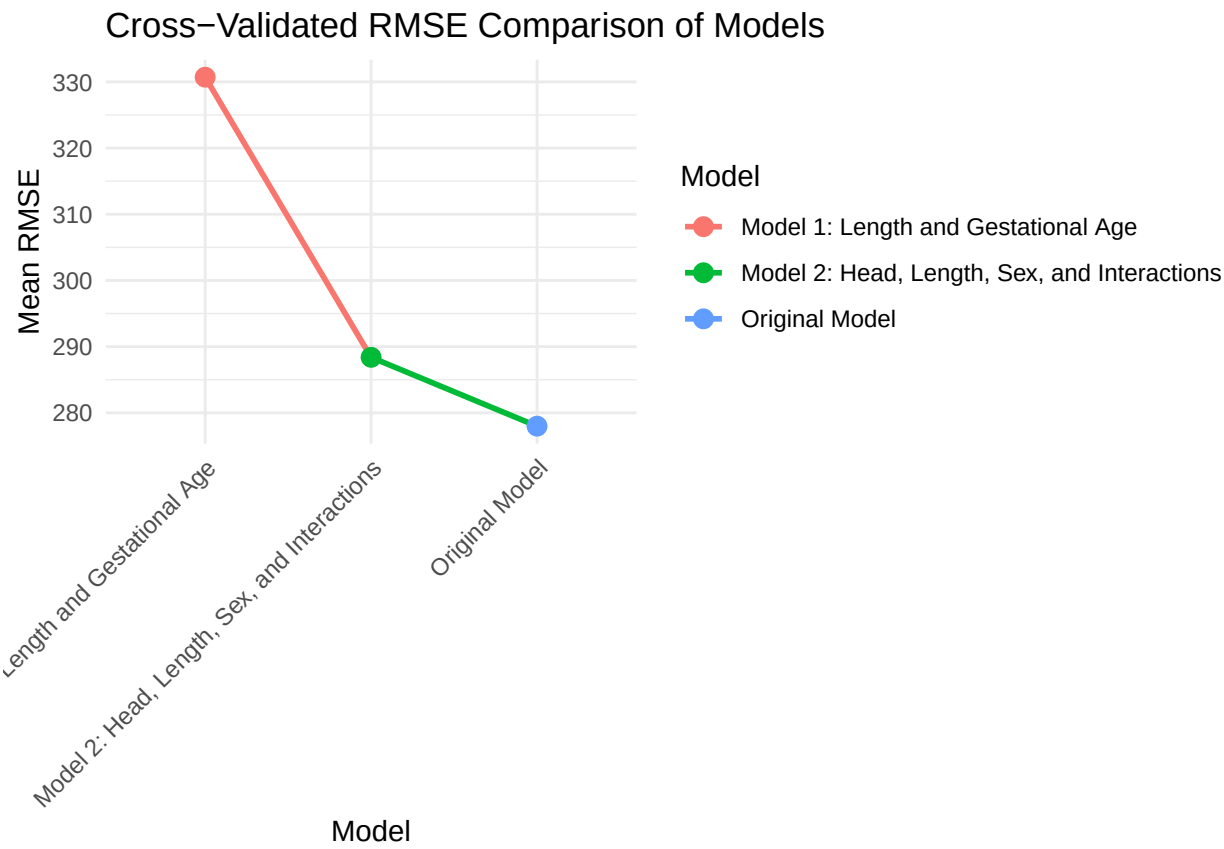
```
#Cross-Validation
set.seed(123)
cv<- crossv_mc(birthweight_data, 100)

cv <- cv %>%
  mutate(model_orig = map(train, ~lm(bwt ~ gaweeks + wtgain + blength + bhead + babysex + delwt + momag
    model1 = map(train, ~lm(bwt ~ blength + gaweeks, data = .x)),
    model2 = map(train, ~lm(bwt ~ bhead * blength * babysex, data = .x)),
    rmse_orig = map2_dbl(model_orig, test, ~rmse(.x, .y)),
    rmse_model1 = map2_dbl(model1, test, ~rmse(.x, .y)),
    rmse_model2 = map2_dbl(model2, test, ~rmse(.x, .y))
  )
```

```
summary <- cv %>%
  summarise(
    mean_rmse_orig = mean(rmse_orig),
    mean_rmse_model1 = mean(rmse_model1),
    mean_rmse_model2 = mean(rmse_model2)
  )
summary
```

```
## # A tibble: 1 x 3
##   mean_rmse_orig mean_rmse_model1 mean_rmse_model2
##            <dbl>            <dbl>            <dbl>
## 1           278.             331.             288.
```

```
rmse_summary_long = summary %>%
  pivot_longer(cols = everything(), names_to = "Model", values_to = "RMSE") %>%
  mutate(Model = recode(Model,
                        "mean_rmse_orig" = "Original Model",
                        "mean_rmse_model1" = "Model 1: Length and Gestational Age",
                        "mean_rmse_model2" = "Model 2: Head, Length, Sex, and Interactions"))

# Plot
ggplot(rmse_summary_long, aes(x = Model, y = RMSE, group = 1)) +
  geom_line(aes(color = Model), size = 1) +
  geom_point(aes(color = Model), size = 3) +
  labs(
    title = "Cross-Validated RMSE Comparison of Models",
    x = "Model",
    y = "Mean RMSE"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Based on the results, the original model has the lowest mean RMSE, indicating its best predictive performance among others. Model 1, which only uses length and gestational age, has the highest RMSE, suggesting it is less effective at predicting birth weight accurately. While Model 2 including interactions between head circumference, length, and sex performs better than Model 1, but it still has a higher RMSE compared to the original model.