p8105_hw6_csc2233

Cameron Chesbrough

2024-11-21

Question 2

```
homicide_df = read_csv(file = "./data/homicide-data copy.csv")
```

```
## 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.
```

```
bad = c("Albuquerque, NM", "Dallas, TX", "Phoenix, AZ",
        "Kansas City, MO", "Tulsa, AL")
homicide_df = homicide_df %>%
  mutate(city_state = paste(city, state, sep = ", "),
         victim_age = as.numeric(victim_age),
         resolved = as.numeric(disposition == "Closed by arrest")) %>%
  filter(!city_state %in% bad) %>%
  filter(victim_race %in% c("Black", "White")) %>%
  drop_na()
```

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

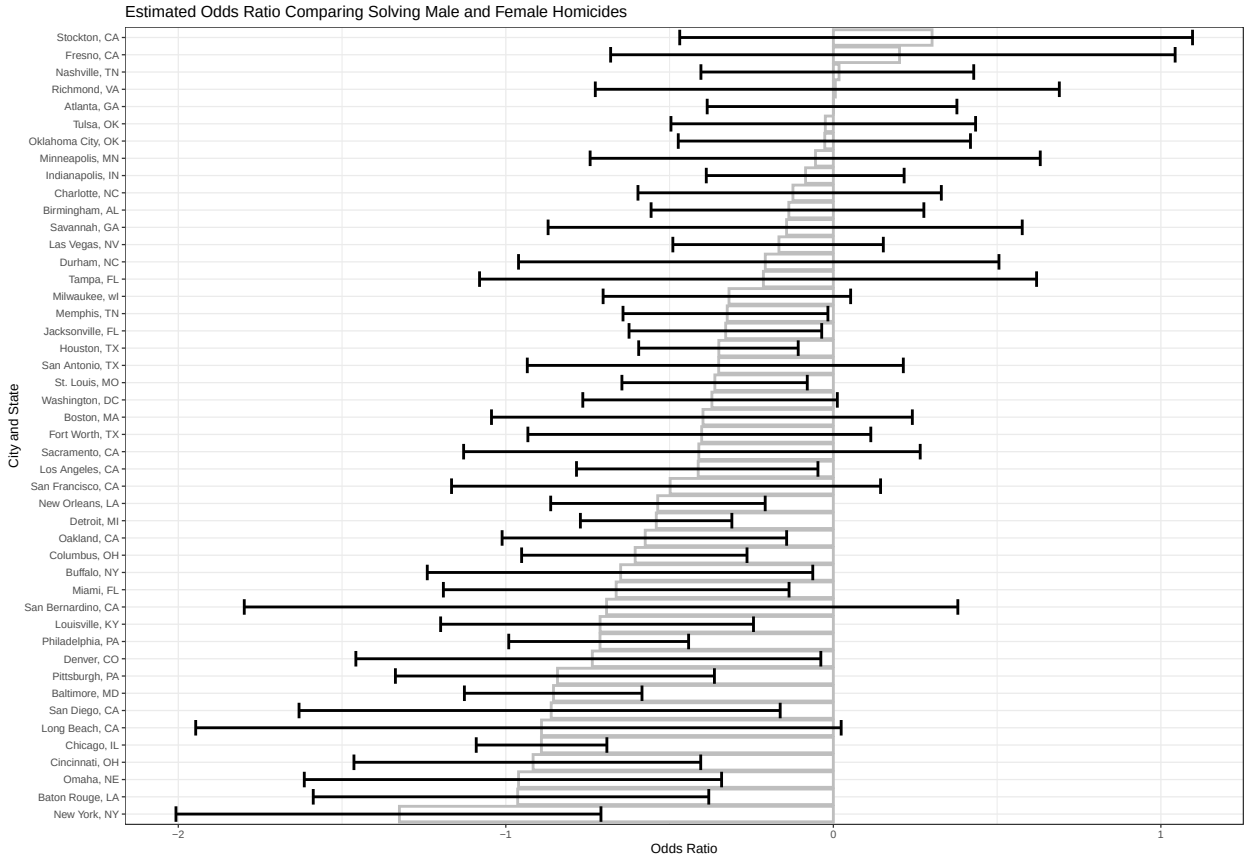
```
bmore = homicide_df %>%
  filter(city_state == "Baltimore, MD")

fit_logistic = bmore |>
  glm(resolved ~ victim_age + victim_race + victim_sex,
      data = _, family = binomial())
fit_logistic |>
  broom::tidy(conf.int = TRUE) |>
  mutate(OR = exp(estimate)) |>
  select(term, log_OR = estimate, conf.low, conf.high)
```

```
## # A tibble: 4 x 4
##   term               log_OR conf.low conf.high
##   <chr>               <dbl>    <dbl>     <dbl>
## 1 (Intercept)       0.310    -0.0245  0.648
## 2 victim_age       -0.00673  -0.0133 -0.000246
## 3 victim_raceWhite  0.842     0.501   1.19
## 4 victim_sexMale   -0.854    -1.13   -0.584
```

```
glm_function = function(x) {
    logis = glm(resolved ~ victim_age + victim_race + victim_sex,
    data = x, family = binomial())
    results = broom::tidy(logis, conf.int = TRUE)
    results[, c("estimate", "conf.low", "conf.high")][4, ]
}

testing = homicide_df %>%
  select(city_state, resolved, victim_age, victim_race, victim_sex)
testing = split(testing, testing$city_state)
testing = map(testing, select, -city_state)
cities = map(testing, glm_function)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
ORs_df = do.call(rbind, cities)
ORs_df = rownames_to_column(ORs_df)

ggplot(ORs_df, aes(reorder(x = rowname, estimate), y = estimate)) +
  geom_col(fill = "white", color = "gray") +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high)) +
  coord_flip() +
  labs(
    title = "Estimated Odds Ratio Comparing Solving Male and Female Homicides",
    y = "Odds Ratio",
    x = "City and State"
  ) +
  theme_bw(base_size = 5)
```



In the majority of cities, male homicides are less likely to be resolved than female homicides. New York is the city with the lowest estimate of the odds ratio. Interestingly, the two cities with positive odds ratios (that are more than a smidge above zero) are both in the central valley of California. Many of the confidence intervals are very wide.

Question 3

```
birthweight_df = read_csv(file = "./data/birthweight.csv")
```

```
## 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.
```

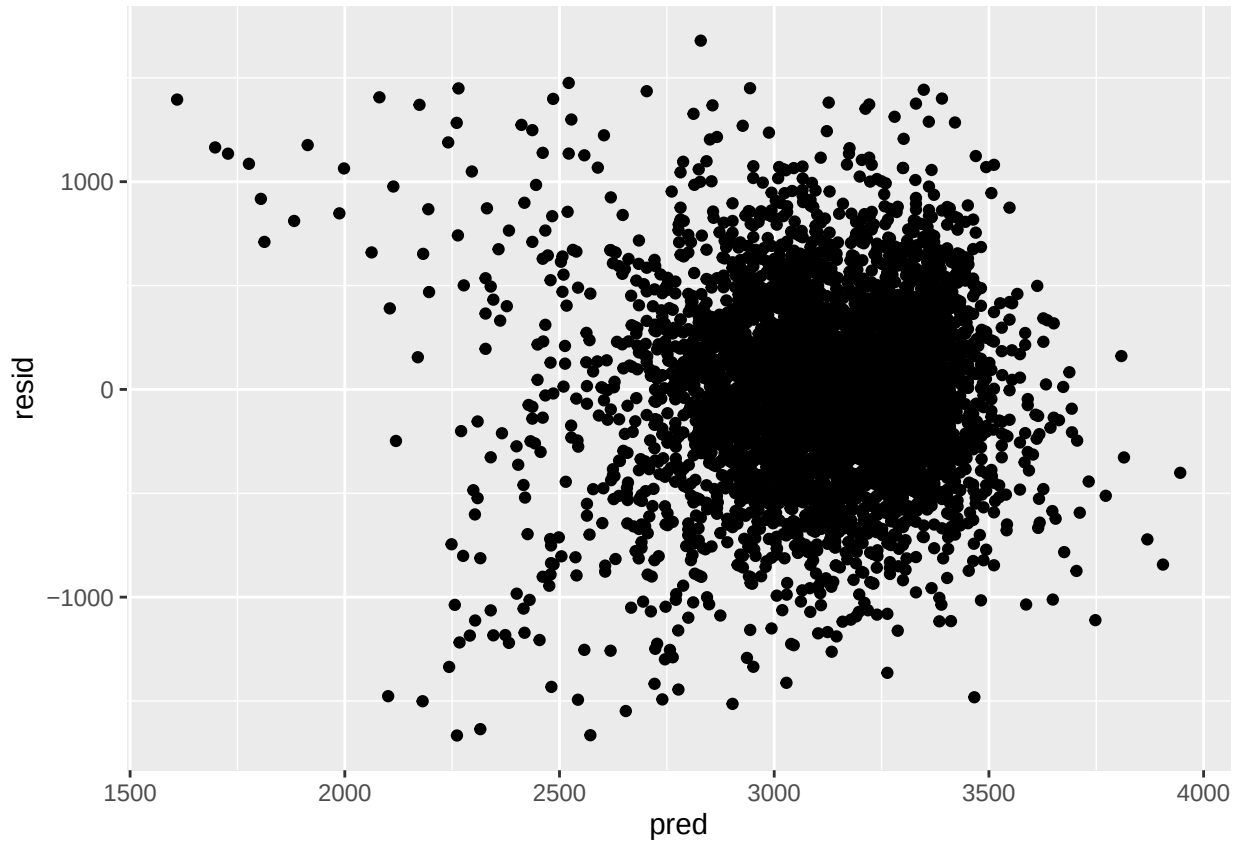
```
birthweight_df = birthweight_df %>%
  mutate(
    babysex = as.factor(babysex),
    frace = as.factor(frace),
    mrace = as.factor(mrace),
    malform = as.factor(malform)
  ) %>%
  select(-c(pnumlbw, pnumsga)) %>%
  mutate(low_age = momage < 15,
         low_age = as.factor(as.numeric(low_age)),
         mother_aa = mrace == 2,
         mother_aa = as.factor(as.numeric(mother_aa))) %>%
  drop_na()


propose_model = lm(bwt ~ gaweeks + low_age + mother_aa + smoken,
                   data = birthweight_df)
summary(propose_model)
```

```
##
## Call:
## lm(formula = bwt ~ gaweeks + low_age + mother_aa + smoken, data = birthweight_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1666.2  -278.0     0.0   283.6  1679.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  883.9648    87.9927  10.046   <2e-16 ***
## gaweeks       60.5484     2.1880  27.673   <2e-16 ***
## low_age1     -50.8383    43.7350  -1.162    0.245
## mother_aa1  -257.6224    14.4488 -17.830   <2e-16 ***
## smoken       -10.1787     0.9424 -10.800   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 447.6 on 4337 degrees of freedom
## Multiple R-squared:  0.2369, Adjusted R-squared:  0.2362
## F-statistic: 336.6 on 4 and 4337 DF,  p-value: < 2.2e-16
```

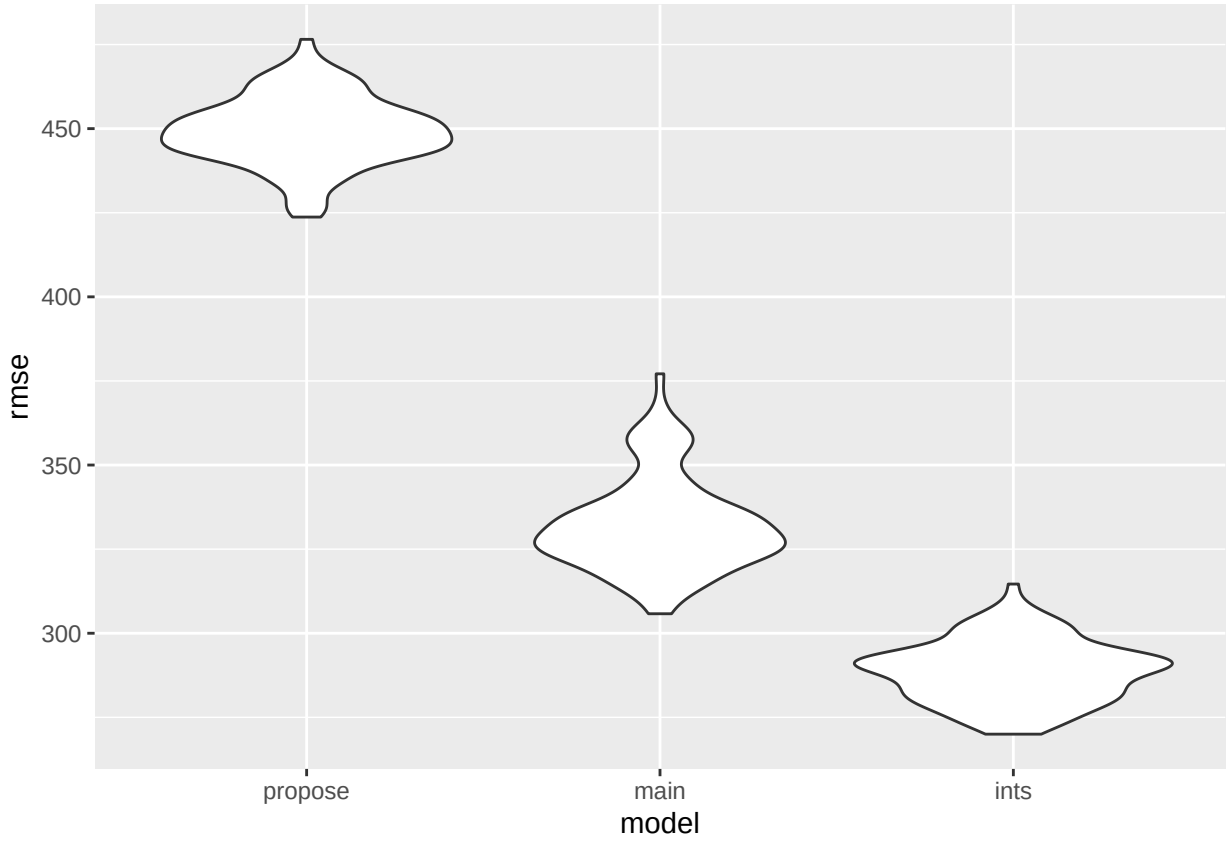
```
propose_model2 = lm(bwt ~ gaweeks + mother_aa + smoken,
                   data = birthweight_df)
summary(propose_model2)
```

```
##
## Call:
## lm(formula = bwt ~ gaweeks + mother_aa + smoken, data = birthweight_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1662.57  -278.64     0.68   282.88  1681.45
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  881.601     87.973   10.02   <2e-16 ***
## gaweeks       60.601      2.188   27.70   <2e-16 ***
## mother_aa1  -260.264     14.270  -18.24   <2e-16 ***
## smoken       -10.146      0.942  -10.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 447.6 on 4338 degrees of freedom
## Multiple R-squared:  0.2366, Adjusted R-squared:  0.2361
## F-statistic: 448.3 on 3 and 4338 DF,  p-value: < 2.2e-16
```

```
birthweight_df %>%
  modelr::add_residuals(propose_model) %>%
  modelr::add_predictions(propose_model) %>%
  ggplot(aes(x = pred, y = resid)) + geom_point()
```



```r
main_effects_model = lm(bwt ~ blength + gaweeks, data = birthweight_df)
ints_model = lm(bwt ~ bhead + blength + babysex +
                bhead*blength + bhead*babysex + blength*babysex +
                bhead*blength*babysex,
                data = birthweight_df)

cv_df = crossv_mc(birthweight_df, 100) %>%
  mutate(
    train = map(train, as_tibble),
    test = map(test, as_tibble))

cv_df = cv_df %>%
  mutate(
    propose_model  = map(train, \(df) lm(bwt ~ gaweeks + low_age + mother_aa + smoken,
                                         data = df)),
    main_effects_model = map(train, \(df) lm(bwt ~ blength + gaweeks,
                                         data = df)),
    ints_model  = map(train, \(df) lm(bwt ~ bhead + blength + babysex +
                                      bhead*blength + bhead*babysex + blength*babysex +
                                      bhead*blength*babysex,
                                      data = as_tibble(df)))) %>%
    mutate(
    rmse_propose = map2_dbl(propose_model, test,
                            \(mod, df) rmse(model = mod, data = df)),
    rmse_main = map2_dbl(main_effects_model, test,
                            \(mod, df) rmse(model = mod, data = df)),
    rmse_ints = map2_dbl(ints_model, 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()
```



To construct a simple model I began by doing some brief research into what causes abnormal birthweights (https://www.chop.edu/conditions-diseases/low-birthweight). While age and race of the mother are already variables, in particular mothers under the age of 15 and african american mothers are at risk; to address this I coded a binary variable for if the mother is under 15, and a variable for if the mother is or is not african american. Because premature births and smoking are also causes, my model was constructed with those variables as well. Comparing my model to the two other models, my constructed model had signifigantly more error. Body length of the baby appears to be a strong predictor for birthweight. The strongest model appears to be the one with head circumference, length, and sex, as well as interactions.