```r
abalone <- read.csv(here::here("data", "abalone_raw.csv"))
source(here::here("R", "utils.R"))
library(ggplot2)
```

The three separate weight variables don't quite sum to total weight, suggesting measurement error. Still, including all three would be a bad idea because it would make the model matrix nearly singular.

```r
summary(with(abalone, Whole.weight - Shucked.weight - Viscera.weight - Shell.weight))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu. 
## -0.44750  0.01800  0.03700  0.04995  0.06800 
##     Max. 
##  0.60800
```

I fit an initial $J - 1$ logits baseline model using sex, dimensions, and whole weight as predictors.

```r
library(nnet)
abalone[["Sex"]] <- factor(abalone[["Sex"]], levels = c("I", "F", "M"))

set.seed(12345)
train_i <- sample(nrow(abalone), floor(nrow(abalone) * 0.8), replace = FALSE)
train <- abalone[train_i, ]
test <- abalone[-train_i, ]
initial_model <- multinom(Sex ~ Whole.weight + Length + Diameter + Height + Rings, data = train)
```

```
## # weights:  21 (12 variable)
## initial  value 3670.463656 
## iter  10 value 2905.758366
## iter  20 value 2882.357358
## final  value 2882.199105 
## converged
```

```r
summary(initial_model)
```

```
## Warning: partial match of 'fitted' to
## 'fitted.values'

## Call:
## multinom(formula = Sex ~ Whole.weight + Length + Diameter + Height + 
##     Rings, data = train)
## 
## Coefficients:
##   (Intercept) Whole.weight    Length  Diameter
## F  -2.9529624     5.393145 -16.57827 10.991562
## M  -0.2375642     6.313870 -17.91517  6.360789
##     Height     Rings
## F 8.401644 0.2278654
## M 5.226298 0.2117678
## 
## Std. Errors:
##   (Intercept) Whole.weight   Length Diameter
## F   0.5507693    0.5022835 3.140566 3.875729
## M   0.4250026    0.4694815 3.004851 3.735638
##     Height      Rings
## F 3.720682 0.02680181
## M 3.619428 0.02618252
##
## Residual Deviance: 5764.398
## AIC: 5788.398
```

```
anova(update(initial_model, . ~ . - .), initial_model)
```

```
## # weights:  6 (2 variable)
## initial  value 3670.463656
## final  value 3662.593993
## converged
```

```
##                                                 Model
## 1                                                   1
## 2 Whole.weight + Length + Diameter + Height + Rings
##   Resid. df Resid. Dev   Test    Df LR stat.
## 1      6680   7325.188           NA       NA
## 2      6670   5764.398 1 vs 2    10  1560.79
##   Pr(Chi)
## 1      NA
## 2       0
```

Most of the estimated coefficients are significant under Wald tests, even after applying the Bonferroni correction to *p*-values.

```
summarized <- broom::tidy(initial_model)
```

```
## Warning: partial match of 'fitted' to
## 'fitted.values'
```

```
summarized[["p.value"]] <- p.adjust(summarized[["p.value"]], method = "bonferroni")
summarized
```

```
## # A tibble: 12 x 6
##    y.level term       estimate std.error statistic
##    <chr>   <chr>         <dbl>     <dbl>     <dbl>
##  1 F       (Intercep~   -2.95     0.551     -5.36
##  2 F       Whole.wei~    5.39     0.502     10.7
##  3 F       Length      -16.6      3.14      -5.28
##  4 F       Diameter     11.0      3.88       2.84
##  5 F       Height        8.40     3.72       2.26
##  6 F       Rings         0.228    0.0268     8.50
##  7 M       (Intercep~   -0.238    0.425     -0.559
##  8 M       Whole.wei~    6.31     0.469     13.4
##  9 M       Length      -17.9      3.00      -5.96
## 10 M       Diameter      6.36     3.74       1.70
## 11 M       Height        5.23     3.62       1.44
## 12 M       Rings         0.212    0.0262     8.09
## # ... with 1 more variable: p.value <dbl>
```

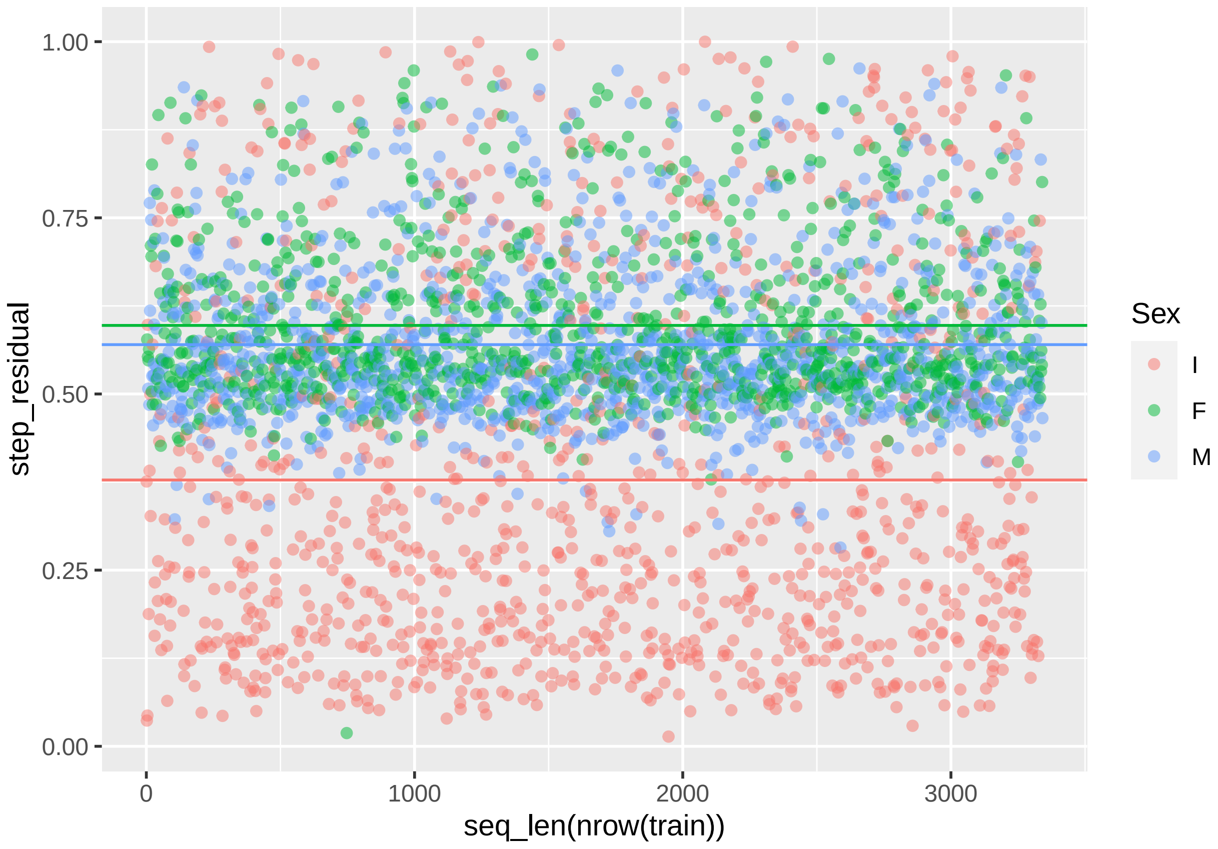
```
rich <- generate_rich_model(initial_model,
  model_data = train[, colnames(coef(initial_model))[-1]],
  model_formula = Sex ~ Whole.weight + Length + Diameter + Height + Rings
)
```

Stepwise selection chooses a model with a few interactions. Note that using stepwise selection violates some of the assumptions behind inferential model statistics.

```
step_model <- suppressMessages(step(initial_model, scope = list(upper = rich, lower = Sex ~ Whole.weigh
step_residual <- extract_by_level(residuals(step_model, "std.res"), as.integer(train[["Sex"]]))
step_fitted <- extract_by_level(step_model[["fitted.values"]], as.integer(train[["Sex"]]))
step_model
```

Average standardized residuals are notably lower for infants than for adults

```
ggplot(
  data.frame(step_fitted, step_residual,
    Sex = train[["Sex"]],
    yintercept = ave(step_residual, train[["Sex"]], FUN = mean)
  ),
  aes(x = seq_len(nrow(train)), y = step_residual, color = Sex)
) +
  geom_point(alpha = .5, size = 1.5) +
  geom_hline(aes(yintercept = yintercept, color = Sex), show.legend = FALSE)
```



```
## Model Testing
```

A likelihood-ratio test of the step-selected model over the initial model is highly significant. A goodness-of-fit test for the step-selected model has a p-value that is computationally 1, providing no evidence against the null of a good fit

```
lr_test(initial_model, step_model)
```

```
## Statistic: 112.255
## Degrees of freedom: 6
## Threshhold: 12.59159
## p = 6.86951e-22
```

```
pchisq(sum(extract_by_level(
  residuals(step_model, "pearson"),
  as.integer(train[["Sex"]])
)^2),
df = nrow(train) - length(coef(step_model)), lower.tail = FALSE
)
```

```
## [1] 1
```

A type II ANOVA test shows that all predictors aside from `diameter` significantly improve the fit when included. `Whole.weight` is by far the most important, followed by `Rings` and the interactions.

```
car::Anova(step_model)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Sex
##                        LR Chisq Df Pr(>Chisq)
## Whole.weight            288.530  2  < 2.2e-16 ***
## Length                   52.765  2  3.485e-12 ***
## Diameter                  4.480  2  0.1064769
## Height                    5.602  2  0.0607488 .
## Rings                    80.745  2  < 2.2e-16 ***
## Whole.weight:Rings       16.054  2  0.0003266 ***
## Height:Rings             12.566  2  0.0018676 **
## Whole.weight:Diameter     4.609  2  0.0998088 .
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next I try refitting the model with all four available weight variables (including replacing the interaction) and comparing AIC. It turns out whole weight has the lowest, but the differences are minor.

```
weight_vars <- c("Whole.weight", "Shucked.weight", "Viscera.weight", "Shell.weight")
names(weight_vars) <- weight_vars
weight_vars <- lapply(weight_vars, as.symbol)
```

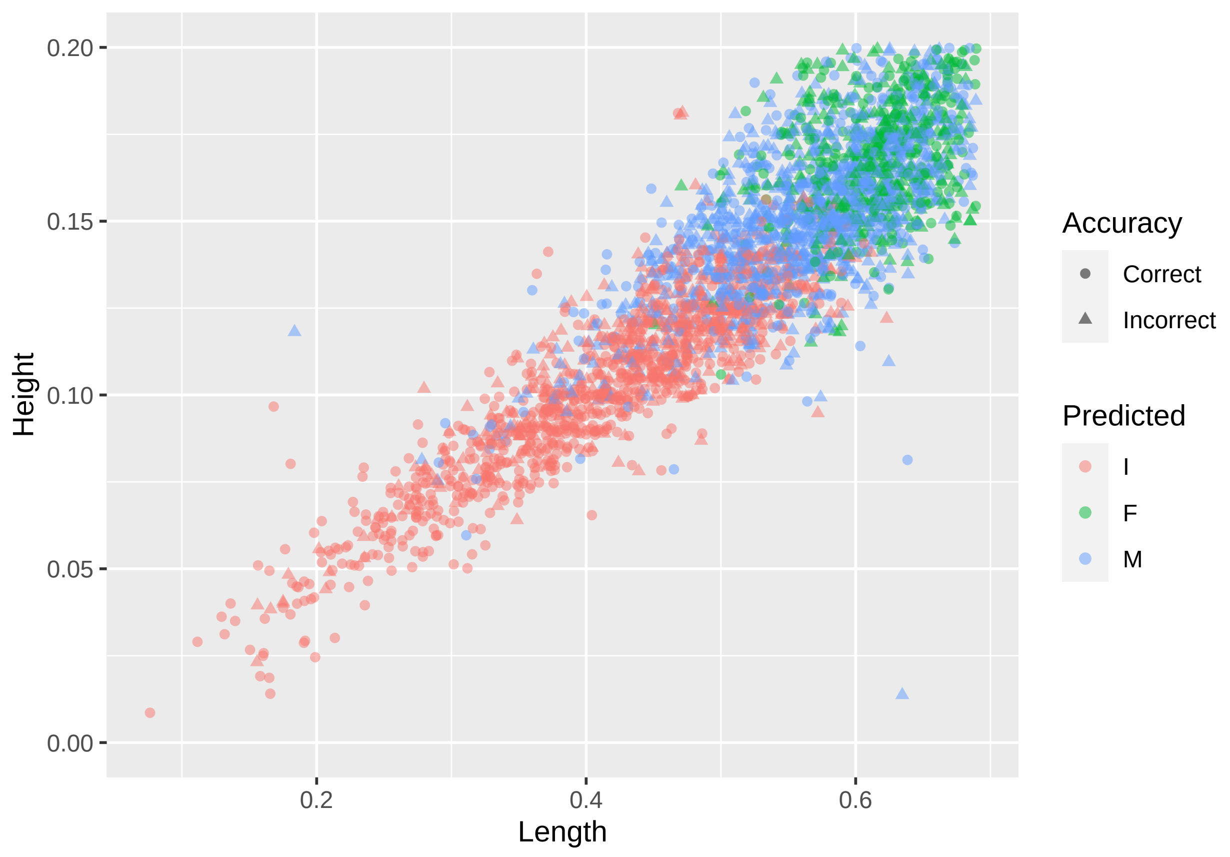
```r
lapply(weight_vars, function(x) {
  update_formula(step_model, Whole.weight, x, indirect = TRUE)[["AIC"]]
})

train_preds <- predict(step_model, newdata = train, type = "class")
```

```
## Warning in model.matrix.default(Terms, m,
## contrasts = object$contrasts): partial argument
## match of 'contrasts' to 'contrasts.arg'
```

Plotting predicted class by height and width shows again that infants are well separated from adults.

```r
ggplot(
  data.frame(train,
    Predicted = train_preds,
    Accuracy = ifelse(train_preds == train[["Sex"]],
      "Correct", "Incorrect"
    )
  ),
  aes(x = Length, y = Height, color = Predicted, shape = Accuracy)
) +
  lims(
    x = quantile(probs = c(0, .95), train[["Length"]]),
    y = quantile(probs = c(0, .95), train[["Height"]])
  ) +
  geom_jitter(alpha = .5)
```

```
## Warning: Removed 258 rows containing missing values
## (geom_point).
```



```
## Model Validation
```

On both training and testing sets, sensitivity, specificity, and precision are much higher for the infant than the adult classes. However, test error was only a little higher than train error. Still,

```
mean(train_preds == train[["Sex"]])
```

```
## [1] 0.5636037
```

```
train_cm <- confusion_matrix(train[["Sex"]], train_preds)
train_cm
```

```
##      predicted
## truth   I   F   M
##     I 859  48 162
##     F 154 330 566
##     M 240 288 694
```

```
analyze_cm(train_cm)
```

```
##                     I         F         M
## sensitivity 0.8035547 0.3142857 0.5679214
## specificity 0.8298217 0.6832380 0.6924869
## precision   0.6855547 0.4954955 0.4880450
```

```r
test_preds <- predict(step_model, newdata = test, type = "class")
```

```
## Warning in model.matrix.default(Terms, m,
## contrasts = object$contrasts): partial argument
## match of 'contrasts' to 'contrasts.arg'
```

```r
mean(test_preds == test[["Sex"]])
```

```
## [1] 0.5681818
```

```r
test_cm <- confusion_matrix(test[["Sex"]], test_preds)
test_cm
```

```
##      predicted
## truth   I   F   M
##     I 223  14  36
##     F  36  82 139
##     M  67  69 170
```

```r
analyze_cm(test_cm)
```

```
##                     I         F         M
## sensitivity 0.8168498 0.3190661 0.5555556
## specificity 0.8344371 0.6919014 0.6916100
## precision   0.6840491 0.4969697 0.4927536
```

```r
analyze_cm(train_cm) - analyze_cm(test_cm)
```

```
##                        I            F
## sensitivity -0.013295093 -0.004780434
## specificity -0.004615368 -0.008663397
## precision    0.001505589 -0.001474201
##                         M
## sensitivity  0.0123658847
## specificity  0.0008769184
## precision   -0.0047086162
```