HW08

Blake Hammarstrom

Problem 1(a)

```
## # A tibble: 2 x 2
##   wordstart ProportionCorrect
##   <chr>                 <dbl>
## 1 no                    0.484
## 2 yes                   0.541
```

Problem 2(a)

```
##
## Call:
## glm(formula = correct ~ wordstart, family = binomial(), data = data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.06224    0.06670  -0.933   0.3507
## wordstartyes  0.22706    0.09446   2.404   0.0162 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2494.2  on 1799  degrees of freedom
## Residual deviance: 2488.4  on 1798  degrees of freedom
## AIC: 2492.4
##
## Number of Fisher Scoring iterations: 3
```

Problem 2(b)

The coefficient for wordstartyes in the regression model represents the log(odds) of getting the first letter correct when word starts with the sound of its first letter, compared to when it does not. If the coefficient is positive, it means that words starting with the sound of their first letter have higher odds of being spelled correctly at the beginning compared to words that don't. The exponent of this coefficient gives the odds ratio, which is the factor of the odds of a correct first letter increasing or decreasing when the word starts with the sound of its first letter.

Problem 2(c)

```
## (Intercept)
## -0.06224231
```

```
## (Intercept)
## -0.06224231
```

```
## (Intercept)
##   0.9396552
```

```
## (Intercept)
##   0.4844444
```

```
## Estimated log(odds) for no: -0.06224231
```

```
## Odds for 'no': 0.9396552
```

```
## Prob for 'no': 0.4844444
```

Problem 2(d)

```
## wordstartyes
##    0.2270588
```

```
## (Intercept)
##   0.1648165
```

```
## (Intercept)
##    1.179177
```

```
## (Intercept)
##   0.5411111
```

```
## Estimated log(odds) for yes: 0.1648165
```

```
## Odds for 'yes': 1.179177
```

```
## Prob for 'yes': 0.5411111
```

Problem 2(e)

```
## (Intercept)
##    1.254904
```

Problem 2(f)

```
##           1           2
## -0.06224231  0.16481653
```

```
##         1         2
## 0.4844444 0.5411111
```

1 = no , 2 = yes

Problem 3(a)

```
##
## Call:
## glm(formula = correct ~ wordstart + age + order + letterwrite,
##     family = binomial(), data = data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.03063    0.76413  -7.892 2.97e-15 ***
## wordstartyes  0.30940    0.11049   2.800  0.00511 **
## age           0.71536    0.15916   4.495 6.97e-06 ***
## order        -0.03886    0.01601  -2.427  0.01521 *
## letterwrite   3.83995    0.23954  16.031  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2494.2  on 1799  degrees of freedom
## Residual deviance: 1956.0  on 1795  degrees of freedom
## AIC: 1966
##
## Number of Fisher Scoring iterations: 4
```

Problem 3(b)

Magnitude: The coefficient changed signifcantly from -.063 to -6.03 Direction: wordstartyes estimate stayed positive Significance: The p-values decreased signficiantly hinting to a more statistically significant second model

Problem 3(c)

A 1 unit increase in letterwrite is associated with an increase in odds of choosing the correct first letter by a factor of exp(Beta_4) (i.e. letterwrite). This suggests that as letter writing improves, the likelihood of correctly spelling words, in terms of starting with the correct letter, increases times exp(Beta_4).

Problem 3(d)

My interpretation of why letterwrite has a larger coefficient is the logic behind it. Having a better understanding of writing inherently comes with a better understanding of spelling. Therefore, I would attribute a large amount of that number to the idea that letterwriting has a larger effect on spelling.

Problem 4(a)

The assumption of independent errors is not satisfied due to the unnacounted variance between spelling abilities. I would use a random effects model to address this problem where I would add a random intercept for every child.

Appendix

```
library(knitr)
# install the tidyverse library (do this once) install.packages('tidyverse')
library(tidyverse)
knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE, fig.width = 4,
    fig.height = 4, tidy = TRUE)
library(dplyr)

data <- read.csv("spelling_HW8.csv")

proportion_correct <- data %>%
    group_by(wordstart) %>%
    summarise(ProportionCorrect = mean(correct), .groups = "drop")

proportion_correct
data$correct <- as.factor(data$correct)

model <- glm(correct ~ wordstart, data = data, family = binomial())

summary(model)

(intercept <- coef(model)["(Intercept)"])
(log_odds_no <- intercept)
(odds_no <- exp(log_odds_no))
(prob_no <- exp(log_odds_no)/(1 + exp(log_odds_no)))


cat("Estimated log(odds) for no:", log_odds_no, "\n")
cat("Odds for 'no':", odds_no, "\n")
cat("Prob for 'no':", prob_no, "\n")
(beta_wordstart_yes <- coef(model)["wordstartyes"])
(log_odds_yes <- intercept + beta_wordstart_yes)
(odds_yes <- exp(log_odds_yes))
(prob_yes <- exp(log_odds_yes)/(1 + exp(log_odds_yes)))

cat("Estimated log(odds) for yes:", log_odds_yes, "\n")
cat("Odds for 'yes':", odds_yes, "\n")
cat("Prob for 'yes':", prob_yes, "\n")
odds <- odds_yes/odds_no
odds
log_odds_predicted <- predict(model, newdata = data.frame(wordstart = c("no", "yes")))
probabilities_predicted <- predict(model, newdata = data.frame(wordstart = c("no",
    "yes")), type = "response")

log_odds_predicted
```

```
probabilities_predicted
model_full <- glm(correct ~ wordstart + age + order + letterwrite, data = data, family = binomial())

summary(model_full)
```