Lab 5: Key

Part 1. Logistic Regression with a categorical predictor

Following the thread from the activity on Tuesday, we will consider a case where we have a binary outcome and a categorical predictor with three levels.

There are two formulations of this model, the first is known as the reference case model. In this formulation the β coefficients correspond to differences from the reference case class.

$$
\begin{aligned}
y &\sim Bernoulli(\pi) \\
\pi &= \frac{\exp(\beta_0 + \beta_1 I_{x=1} + \beta_2 I_{x=2})}{1 + \exp(\beta_0 + \beta_1 I_{x=1} + \beta_2 I_{x=2})} \\
\pi &= logit^{-1}(\beta_0 + \beta_1 I_{x=1} + \beta_2 I_{x=2})
\end{aligned}
$$

The second is the cell means model which can more directly estimate probabilities associated with each class.

$$
\begin{aligned}
y &\sim Bernoulli(\pi) \\
\pi &= \frac{\exp(\beta_0 I_{x=0} + \beta_1 I_{x=1} + \beta_2 I_{x=2})}{1 + \exp(\beta_0 I_{x=0} + \beta_1 I_{x=1} + \beta_2 I_{x=2})} \\
\pi &= logit^{-1}(\beta_0 I_{x=0} + \beta_1 I_{x=1} + \beta_2 I_{x=2})
\end{aligned}
$$

t^{-1}(*0* I{x = 0} + *1* I{x =1}+ *2* I{x =2})\ \end{align*}

1. Data Simulation (6 points)

Use the cell means model and simulate data with the following properties

- Let there be a total of 150 observations, 50 from `x = "1"`, 50 from $x =$" 2", and 50 from $x =$" 3".

```
n <- 50
x <- rep(c('1','2','3'), each = n)
```

- Set the coefficient values to the following values:$\beta_0 = 0$, $\beta_1 = 1$, and $\beta_2 = -1$.

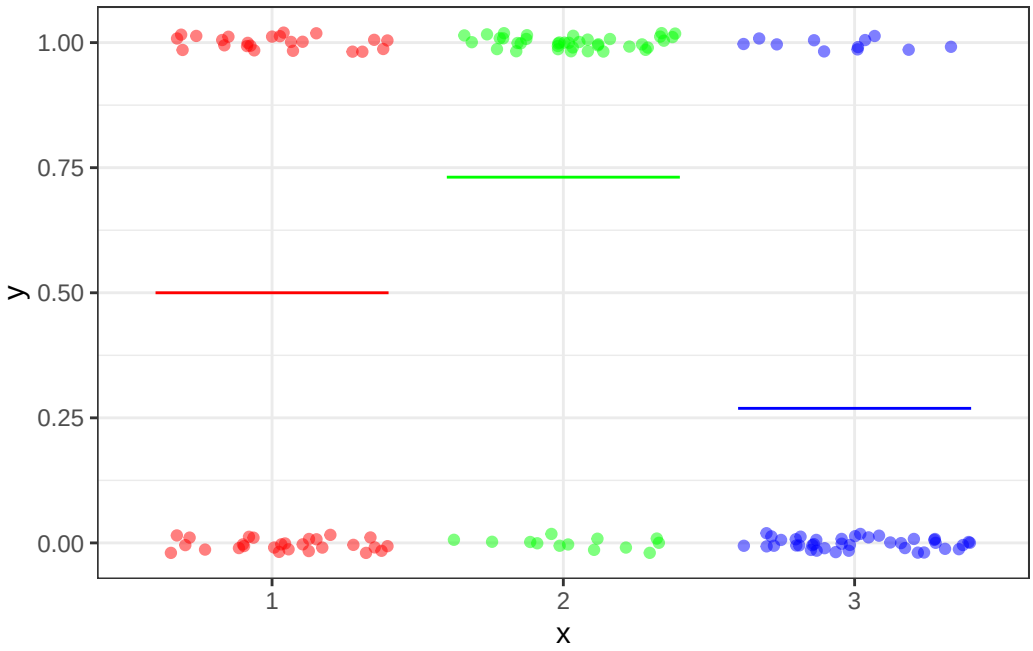
```
beta <- c(0, 1, -1)
```

- Simulate a binary outcome at each x value.

```
pi_values <- invlogit(rep(beta, each = n))
y_values <- rbinom(n * 3, 1, pi_values)
```

- Create a plot that displays the pi values along with observed binary responses.

```
tibble(x = x, pi = pi_values, y = y_values) |>
  ggplot(aes(y = y, x=x, color = x)) +
  geom_jitter(height = .02, alpha = .5) +
  theme_bw() +
  theme(legend.position = 'none') +
  annotate("segment", x = .6, xend = 1.4, y = pi_values[1], yend = pi_values[1], colour = "r
  annotate("segment", x = 1.6, xend = 2.4, y = pi_values[51], yend = pi_values[51], colour =
  annotate("segment", x = 2.6, xend = 3.4, y = pi_values[101], yend = pi_values[101], colour
  scale_color_manual(values = c("1" = "red", "2" = "green", "3" = "blue"))
```



2. Contingency Table (6 points)

Using your simulated data, print a contingency table and determine whether you'd detect any association between the two variables (binary outcome and categorical predictor).

```
cont_table <-  tibble(x = x, pi = pi_values, y = y_values) |>
  group_by(x) |>
  summarize(ones = sum(y ==1), zeros = sum(y ==0))

cont_table |>
  kable()
```

```
Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

```
Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

x	ones	zeros
1	23	27
2	37	13
3	11	39

```
chisq.test(cont_table[,-1])
```

```
    Pearson's Chi-squared test

data:  cont_table[, -1]
X-squared = 27.171, df = 2, p-value = 1.259e-06
```

3. Model fitting (6 points)

Fit your data using either an MLE or Bayesian approach, report point estimates and uncertainty intervals for the beta values. Hint, you can get the cell means coding with `y ~ x - 1`

```
log_data <- tibble(x = x, pi = pi_values, y = y_values)

model.fit <- glm(y ~ x - 1, data = log_data, family = binomial)
summary(model.fit)
```

```
Call:
glm(formula = y ~ x - 1, family = binomial, data = log_data)

Coefficients:
   Estimate Std. Error z value Pr(>|z|)
x1  -0.1603     0.2838  -0.565 0.572019
x2   1.0460     0.3224   3.244 0.001178 **
x3  -1.2657     0.3414  -3.707 0.000209 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 207.94  on 150  degrees of freedom
```

```
Residual deviance: 178.99  on 147  degrees of freedom
AIC: 184.99

Number of Fisher Scoring iterations: 4
```

```
confint(model.fit)
```

```
Waiting for profiling to be done...

        2.5 %     97.5 %
x1 -0.7242625  0.3952540
x2  0.4406363  1.7158481
x3 -1.9840045 -0.6318829
```

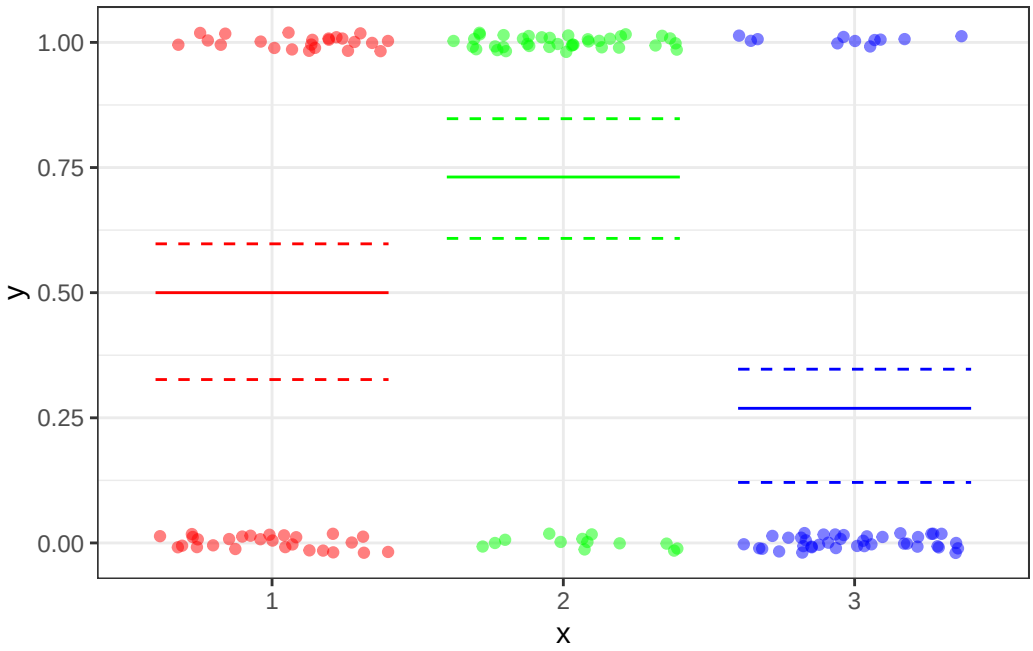
4. Probability Scale (6 points)

With this model framework, we can also directly estimate uncertainty intervals for the probabilities using $\pi = logit^{-1}(\beta)$. Compute uncertainty intervals for the probability of success for each class and add them to an updated version of the figure from part 1.

```
probs <- invlogit(confint(model.fit))
```

```
Waiting for profiling to be done...
```

```
tibble(x = x, pi = pi_values, y = y_values) |>
  ggplot(aes(y = y, x=x, color = x)) +
  geom_jitter(height = .02, alpha = .5) +
  theme_bw() +
  theme(legend.position = 'none') +
  annotate("segment", x = .6, xend = 1.4, y = pi_values[1], yend = pi_values[1], colour = "re
  annotate("segment", x = .6, xend = 1.4, y = probs[1,1], yend = probs[1,1], colour = "red",
  annotate("segment", x = .6, xend = 1.4, y = probs[1,2], yend = probs[1,2], colour = "red",
  annotate("segment", x = 1.6, xend = 2.4, y = pi_values[51], yend = pi_values[51], colour =
  annotate("segment", x = 1.6, xend = 2.4, y = probs[2,1], yend = probs[2,1], colour = "green
  annotate("segment", x = 1.6, xend = 2.4, y = probs[2,2], yend = probs[2,2], colour = "green
  annotate("segment", x = 2.6, xend = 3.4, y = pi_values[101], yend = pi_values[101], colour
  annotate("segment", x = 2.6, xend = 3.4, y = probs[3,1], yend = probs[3,1], colour = "blue"
  annotate("segment", x = 2.6, xend = 3.4, y = probs[3,2], yend = probs[3,2], colour = "blue"
  scale_color_manual(values = c("1" = "red", "2" = "green", "3" = "blue"))
```



5. Alternative Models (6 points)

Why do the two models below give you incorrect results?

```
model.fit <- glm(y ~ as.numeric(x) - 1, data = log_data, family = binomial)
model.fit <- glm(y ~ x - 1, data = log_data)
```