Lab 10

Adam Shen

November 25, 2020

Packages

```
library(dplyr)
library(broom)
library(ggplot2)
theme_set(theme_bw())
```

Logistic regression with grouped data

Make the data

Normally when working with factors, I would do something similar to what I did in Lab 5 with the `lowwt` data `Hospital` variable. Since we are creating the data from scratch for this example, we will use three separate variables for `Colour` (4 levels) rather than one variable.

```
# n: number of trials (binomial)
n <- c(12, 95, 44, 22)

# s: number of successes (>=1 Satellites)
s <- c(9, 69, 26, 7)

# f: number of failures (<1 Satellites)
f <- n - s

# Colour indicators
c1 <- c(1, 0, 0, 0)
c2 <- c(0, 1, 0, 0)
c3 <- c(0, 0, 1, 0)
```

Fit the model

```
m1 <- glm(cbind(s, f) ~ c1 + c2 + c3, family=binomial)
```

The interpretation of $\hat{\pi}_i$ using the above model is the predicted probability of success, where success is observing at least one satellite.

Suppose instead, we did:

```
m2 <- glm(cbind(f, s) ~ c1 + c2 + c3, family=binomial)
```

Then the interpretation of $\hat{\pi}_i$ using the above model is the predicted probability of success, where success is observing **no** satellites.

Coefficient comparison

```
list(m1 = tidy(m1), m2 = tidy(m2))
```

```
## $m1
## # A tibble: 4 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   -0.762     0.458     -1.67 0.0959
## 2 c1             1.86      0.809      2.30 0.0214
## 3 c2             1.74      0.512      3.39 0.000692
## 4 c3             1.13      0.551      2.05 0.0403
##
## $m2
## # A tibble: 4 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    0.762     0.458      1.67 0.0959
## 2 c1            -1.86      0.809     -2.30 0.0214
## 3 c2            -1.74      0.512     -3.39 0.000692
## 4 c3            -1.13      0.551     -2.05 0.0403
```

Check for model usefulness

```
anova(m1, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: cbind(s, f)
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                     3    13.6977
## c1    1   0.6941         2    13.0035 0.404758
## c2    1   8.5640         1     4.4395 0.003429 **
## c3    1   4.4395         0     0.0000 0.035116 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Similar to the previous multiple regression models, this requires a bit of work to check for model usefulness. Instead, let us fit a null model and compare it to the full model.

```
m0 <- update(m1, . ~ 1)
anova(m0, m1, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(s, f) ~ 1
## Model 2: cbind(s, f) ~ c1 + c2 + c3
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         3     13.698
## 2         0      0.000  3   13.698 0.003347 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The hypotheses are:

$$H_0 : \beta_1 = \beta_2 = \beta_3 = 0 \quad \text{vs} \quad H_A : \text{At least one } \beta_i \text{ non-zero, } i = 1, 2, 3$$

Since the p-value, 0.003347, is less than 0.05, we reject the null hypothesis and conclude that the full model is more useful than the null model.

Variance-covariance matrix

```
vcov(m1)
```

```
##             (Intercept)         c1         c2         c3
## (Intercept)   0.2095238 -0.2095238 -0.2095238 -0.2095238
## c1           -0.2095238  0.6539679  0.2095238  0.2095238
## c2           -0.2095238  0.2095238  0.2624781  0.2095238
## c3           -0.2095238  0.2095238  0.2095238  0.3035409
```

Parameter estimates and CIs for odds ratios

Recall that our fitted model is in terms of log-odds and has equation:

$$\ln\left(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}\right) = \hat{\beta}_0 + \hat{\beta}_1 x_{1i} + \hat{\beta}_2 x_{2i} + \hat{\beta}_3 x_{3i}$$

If we seek parameter estimates and their confidence intervals on the odds ratio scale, we need to exponentiate:

$$\frac{\hat{\pi}_i}{1-\hat{\pi}_i} = e^{\hat{\beta}_0} \cdot e^{\hat{\beta}_1 x_{1i}} \cdot e^{\hat{\beta}_2 x_{2i}} \cdot e^{\hat{\beta}_3 x_{3i}}$$

Then the exponentiated fitted equation says that for a unit increase in x_{1i}, the odds ratio of observing at least one satellite will increase by a **factor** of $e^{\hat{\beta}_1}$ units.

```
exp(coef(m1))
```

```
## (Intercept)          c1          c2          c3
##   0.4666667   6.4285714   5.6868132   3.0952381
```

```
exp(confint.default(m1)) # Using standard normal quantiles
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.1902741  1.144548
## c1          1.3175357 31.366536
## c2          2.0834167 15.522504
## c3          1.0513043  9.112965
```

Horseshoe crab data: logistic regression with ungrouped data

```
crabs <- read.table("./hcrabs.txt", header=TRUE)
```

Prep the data

We need to create our response variable where $y = 1$ if there is at least one satellite, and $y = 0$ otherwise. Similar to Lab 5, we want to avoid creating additional indicator variables because this makes prediction difficult. Colour has levels `1`, `2`, `3`, `4`. Spine has levels `1`, `2`, `3`. In order to match the output of the code given in the lab instructions, we will make `Colour = 4` and `Spine = 3` the baseline.

```r
crabs <- crabs %>%
  mutate(
    y = ifelse(Satellites >= 1, 1, 0),
    across(c(Colour, Spine), as.factor),
    Colour = relevel(Colour, ref="4"),
    Spine = relevel(Spine, ref="3")
  )

levels(crabs$Colour)
```

```
## [1] "4" "1" "2" "3"
```

```r
levels(crabs$Spine)
```

```
## [1] "3" "1" "2"
```

Fit the model

```r
m <- glm(y ~ Colour + Spine + Width + Weight, family=binomial, data=crabs)
summary(m)
```

```
##
## Call:
## glm(formula = y ~ Colour + Spine + Width + Weight, family = binomial,
##     data = crabs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1977  -0.9424   0.4849   0.8491   2.1198
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -9.2734     3.8378  -2.416  0.01568 *
## Colour1       1.6087     0.9355   1.720  0.08552 .
## Colour2       1.5058     0.5667   2.657  0.00788 **
## Colour3       1.1198     0.5933   1.887  0.05910 .
## Spine1       -0.4003     0.5027  -0.796  0.42588
## Spine2       -0.4963     0.6292  -0.789  0.43024
## Width         0.2631     0.1953   1.347  0.17788
## Weight        0.8258     0.7038   1.173  0.24069
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 185.20  on 165  degrees of freedom
## AIC: 201.2
##
## Number of Fisher Scoring iterations: 4
```

Check for model usefulness

```r
m0 <- update(m, . ~ 1)
anova(m0, m, test="Chisq")
```

```
## Analysis of Deviance Table
```

```
##
## Model 1: y ~ 1
## Model 2: y ~ Colour + Spine + Width + Weight
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       172     225.76
## 2       165     185.20  7   40.557 9.848e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The hypotheses are:

$$H_0 : \text{All } \beta_i = 0,\ i = 1, 2, \ldots, 7 \quad \text{vs} \quad H_A : \text{At least one } \beta_i \text{ non-zero},\ i = 1, 2, \ldots, 7$$

Since the p-value, 9.848e-07, is less than 0.05, we reject the null hypothesis and conclude that the full model is more useful than the null model.

Can `Colour` be removed from the full model?

```
m_noColour <- update(m, . ~ . - Colour)
summary(m_noColour)
```

```
##
## Call:
## glm(formula = y ~ Spine + Width + Weight, family = binomial,
##     data = crabs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0906  -1.0418   0.5251   0.8928   1.7009
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.12885    3.64881  -2.502   0.0124 *
## Spine1      -0.07689    0.45482  -0.169   0.8658
## Spine2      -0.16741    0.60391  -0.277   0.7816
## Width        0.29717    0.18646   1.594   0.1110
## Weight       0.85719    0.67702   1.266   0.2055
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 192.80  on 168  degrees of freedom
## AIC: 202.8
##
## Number of Fisher Scoring iterations: 4
```

```
anova(m_noColour, m, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ Spine + Width + Weight
## Model 2: y ~ Colour + Spine + Width + Weight
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       168      192.8
## 2       165      185.2  3   7.5958  0.05515 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The hypotheses are:

$$H_0 : \beta_1 = \beta_2 = \beta_3 = 0 \quad \text{vs} \quad H_A : \text{At least one } \beta_i \text{ non-zero, } i = 1, 2, 3$$

Since the p-value, 0.05515, is not less than 0.05, we fail to reject the null hypothesis. There is insufficient evidence to support the claim that at least one of the `Colour` parameters is non-zero. Therefore, we can remove `Colour` from the full model.

Horseshoe crab data: residual analysis

```
crabs <- read.table("./hcrabs.txt", header=TRUE)
```

Prep data

```
crabs <- crabs %>%
  mutate(
    y = ifelse(Satellites >= 1, 1, 0),
    NotDark = ifelse(Colour != 4, 1, 0)
  )
```

Fit the model

```
m <- glm(y ~ NotDark + Width, family=binomial, data=crabs)
summary(m)
```

```
##
## Call:
## glm(formula = y ~ NotDark + Width, family = binomial, data = crabs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0821  -0.9932   0.5274   0.8606   2.1553
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -12.9795     2.7272  -4.759 1.94e-06 ***
## NotDark       1.3005     0.5259   2.473   0.0134 *
## Width         0.4782     0.1041   4.592 4.39e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.76  on 172  degrees of freedom
## Residual deviance: 187.96  on 170  degrees of freedom
## AIC: 193.96
##
## Number of Fisher Scoring iterations: 4
```

Check for model usefulness

```
m0 <- update(m, . ~ 1)
anova(m0, m, test="Chisq")
```

```
## Analysis of Deviance Table
```

```
##
## Model 1: y ~ 1
## Model 2: y ~ NotDark + Width
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       172     225.76
## 2       170     187.96  2   37.801 6.19e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The hypotheses are:

$$H_0 : \beta_1 = \beta_2 = 0 \quad \text{vs} \quad H_A : \text{At least one } \beta_i \text{ non-zero, } i = 1, 2$$

Since the p-value, 6.19e-09, is less than 0.05, we reject the null hypothesis. We conclude that this model is more useful than the null model.

Additional model information

The sum of the squared deviance residuals is equal to the deviance of the model:

```
sum(resid(m, type="deviance")^2)
```

```
## [1] 187.9579
```

```
m$deviance
```

```
## [1] 187.9579
```

Obtaining the additional model info:

```
model_info <- data.frame(
  .fitted = fitted(m),
  p.resid = resid(m, type="pearson"),
  d.resid = resid(m, type="deviance"),
  .hat = hatvalues(m)
) %>%
  transform(std.p.resid = p.resid / sqrt(1 - .hat)) %>%
  round(digits=4)
head(model_info)
```

```
##   .fitted p.resid d.resid   .hat std.p.resid
## 1  0.8647  0.3956  0.5393 0.0123      0.3981
## 2  0.2852 -0.6316 -0.8194 0.0313     -0.6417
## 3  0.6802  0.6857  0.8779 0.0080      0.6884
## 4  0.5451 -1.0947 -1.2551 0.0116     -1.1011
## 5  0.6802  0.6857  0.8779 0.0080      0.6884
## 6  0.4262 -0.8619 -1.0540 0.0194     -0.8703
```

We can do something similar with `broom::augment.glm`. What we will need to do:

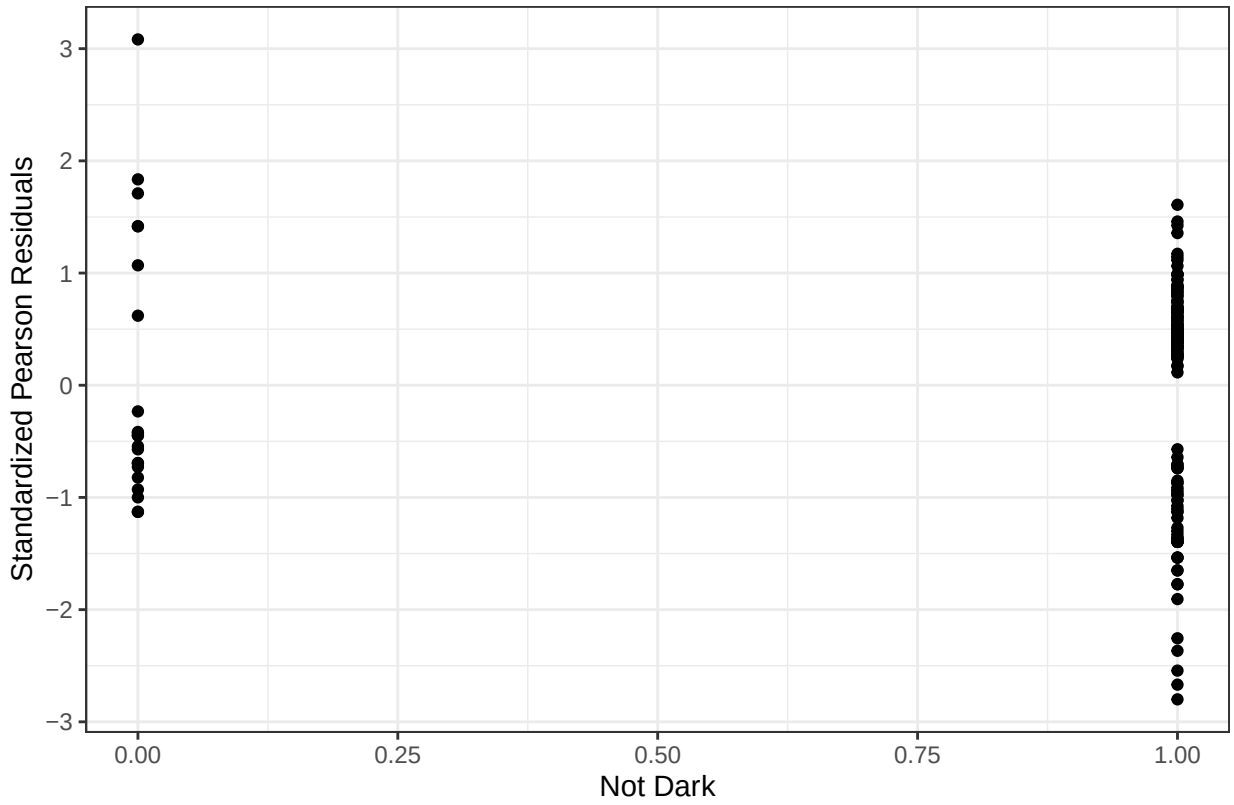
- By default, the `.fitted` values are on the predictor scale so we need to specify `type.predict="response"` to get fitted values from the response scale
- The leverage values will be found under the `.hat` column
- By default, `augment.glm` returns deviance residuals under the `.resid` column, so we will add the Pearson residuals manually
- Create the standardized Pearson residuals manually
- Drop unneeded columns

```
(model_aug <- m %>%
   augment(type.predict = "response") %>%
   rename(d.resid = .resid) %>%
   mutate(
     p.resid = resid(m, type = "pearson"),
     std.p.resid = p.resid / sqrt(1 - .hat)
   ) %>%
   select(-.std.resid, -.sigma, -.cooksd))
```

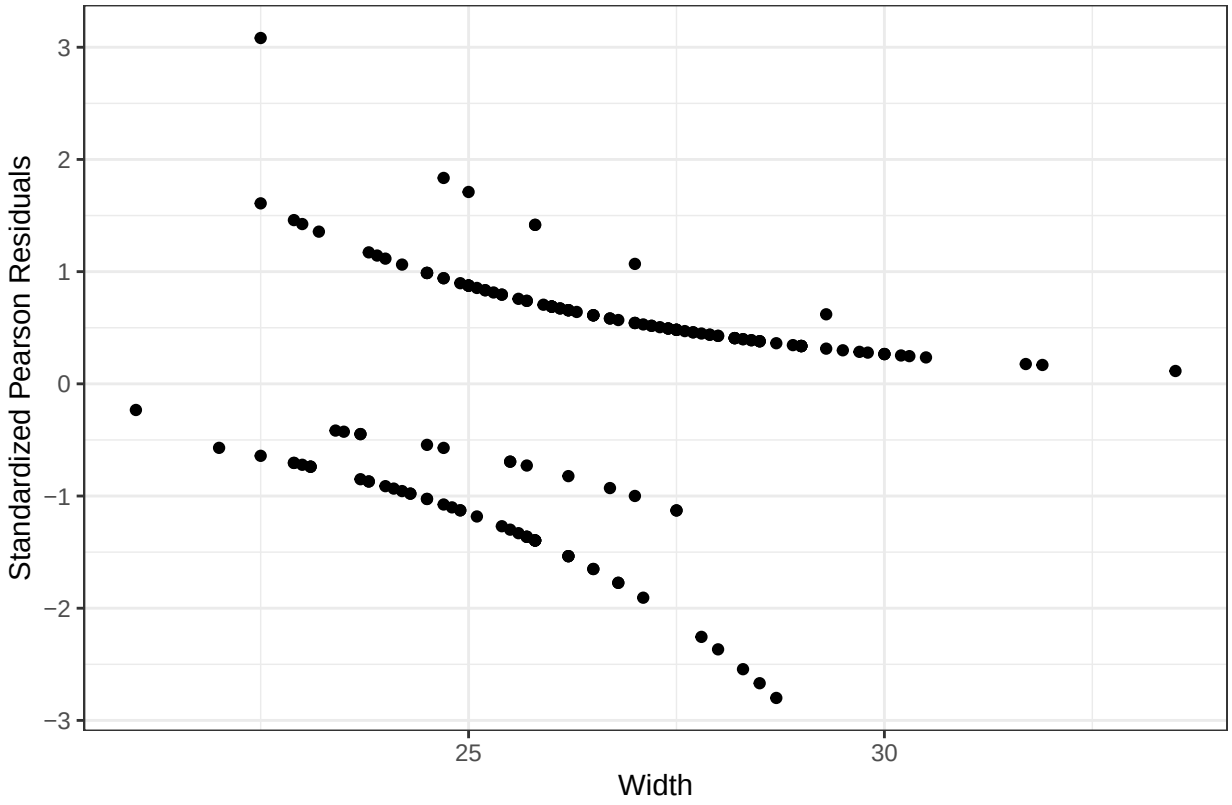
```
## # A tibble: 173 x 8
##        y NotDark Width .fitted d.resid    .hat p.resid std.p.resid
##    <dbl>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>       <dbl>
##  1     1       1     1  28.3    0.865    0.539 0.0123    0.396       0.398
##  2     0       1     1  22.5    0.285   -0.819 0.0313   -0.632      -0.642
##  3     1       1     1  26      0.680    0.878 0.00802   0.686       0.688
##  4     0       1     1  24.8    0.545   -1.26  0.0116   -1.09       -1.10
##  5     1       1     1  26      0.680    0.878 0.00802   0.686       0.688
##  6     0       1     1  23.8    0.426   -1.05  0.0194   -0.862      -0.870
##  7     0       1     1  26.5    0.730   -1.62  0.00826  -1.64       -1.65
##  8     0       1     1  24.7    0.533   -1.23  0.0123   -1.07       -1.08
##  9     0       1     1  23.7    0.415   -1.03  0.0203   -0.841      -0.850
## 10     0       1     1  25.6    0.637   -1.42  0.00850  -1.33       -1.33
## # ... with 163 more rows
```

Plots of standardized Pearson residuals

```
ggplot(model_aug, aes(x=NotDark, y=std.p.resid))+
  geom_point()+
  labs(x="Not Dark", y="Standardized Pearson Residuals")
```



```
ggplot(model_aug, aes(x=Width, y=std.p.resid))+
  geom_point()+
  labs(y="Standardized Pearson Residuals")
```



From *Module 10.8*, when n_i is not small, a standardized Pearson residual exceeding 2 in absolute value indicates a lack of fit for observation i. We can't really do much with these plots here since the data is ungrouped, so $n_i = 1$ for all i.

Correlation

```
with(model_aug, cor(y, .fitted))
```

```
## [1] 0.4469688
```

Grouped horseshoe crab data: residual and influence analysis

```
crabsg <- read.table("./hcrabgp.txt", header=TRUE)
```

Data prep

```
crabsg <- crabsg %>%
  mutate(Nosat = Total - Withsat)
```

Fit the model

```
m <- glm(cbind(Withsat, Nosat) ~ NotDark + Iwidth, family=binomial, data=crabsg)
summary(m)
```

```
##
## Call:
## glm(formula = cbind(Withsat, Nosat) ~ NotDark + Iwidth, family = binomial,
##     data = crabsg)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4856  -0.4884   0.1861   0.7461   2.1987
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -13.4208     2.6948  -4.980 6.35e-07 ***
## NotDark       1.2748     0.5331   2.391   0.0168 *
## Iwidth        0.5044     0.1046   4.821 1.43e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 58.579  on 18  degrees of freedom
## Residual deviance: 17.366  on 16  degrees of freedom
## AIC: 50.362
##
## Number of Fisher Scoring iterations: 4
```

Check for model usefulness

```
m0 <- update(m, . ~ 1)
anova(m0, m, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(Withsat, Nosat) ~ 1
## Model 2: cbind(Withsat, Nosat) ~ NotDark + Iwidth
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        18     58.579
## 2        16     17.366  2   41.213 1.124e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The hypotheses are:

$$H_0 : \beta_1 = \beta_2 = 0 \quad \text{vs} \quad H_A : \text{At least one } \beta_i \text{ non-zero, } i = 1, 2$$

Since the p-value, 1.124e-09, is less than 0.05, we reject the null hypothesis. We conclude that the current model is more useful than the null model.

Variance-covariance matrix

```
vcov(m)
```

```
##             (Intercept)       NotDark        Iwidth
## (Intercept)   7.2618273 -0.2383070199 -0.2771377066
## NotDark      -0.2383070  0.2841469345 -0.0003425998
## Iwidth       -0.2771377 -0.0003425998  0.0109489620
```

Parameter estimates and CIs for odds ratios

```
exp(coef(m))
```

```
##  (Intercept)      NotDark       Iwidth
## 1.484026e-06 3.577963e+00 1.656002e+00
```

```
exp(confint.default(m))
```

```
##                    2.5 %       97.5 %
## (Intercept) 7.544711e-09 2.919044e-04
## NotDark     1.258633e+00 1.017121e+01
## Iwidth      1.348943e+00 2.032958e+00
```

Additional model information

We're going to augment it manually here because `augment` gets a bit weird when the response variable involves `cbind`.

```
model_aug <- crabsg %>%
  as_tibble() %>%
  select(Withsat, Nosat, NotDark, Iwidth) %>%
  mutate(
    .fitted = fitted(m),
    .hat = hatvalues(m),
    d.resid = resid(m, type = "deviance"),
    p.resid = resid(m, type = "pearson"),
    std.p.resid = p.resid / sqrt(1 - .hat),
    ci = (p.resid^2) * .hat / ((1 - .hat)^2)
  )
```

Hide the `.fitted` column for now, otherwise `ci` gets cut off.

```
model_aug %>% select(-.fitted)
```

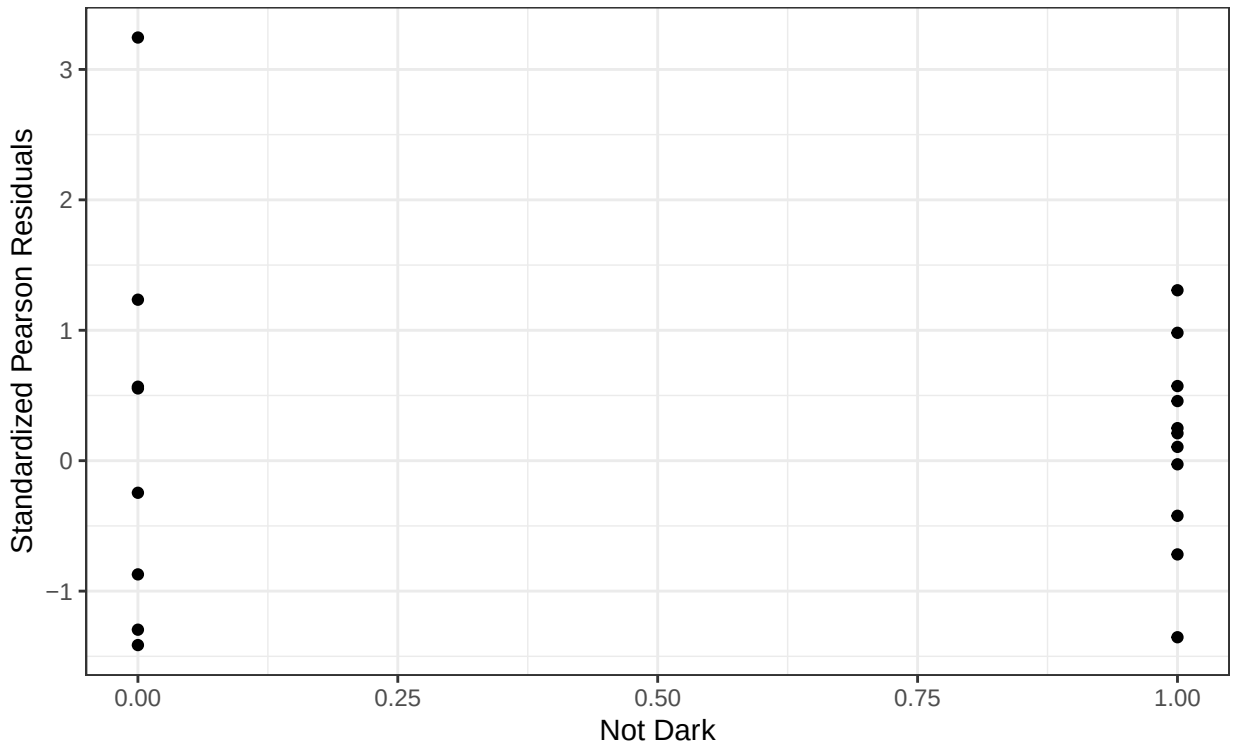
```
## # A tibble: 19 x 9
##    Withsat Nosat NotDark Iwidth    .hat d.resid p.resid std.p.resid        ci
##      <int> <int>   <int>  <int>   <dbl>   <dbl>   <dbl>       <dbl>     <dbl>
##  1       0     1       0     21 0.0237  -0.339  -0.243      -0.246  0.00147
##  2       1     0       0     22 0.0298   2.20    3.20        3.24   0.324
##  3       2     4       1     22 0.184    0.402   0.413       0.457  0.0471
##  4       0     4       0     23 0.147   -1.10   -0.805      -0.872  0.131
##  5       4     7       1     23 0.249   -0.0240 -0.0239     -0.0276 0.000252
##  6       1     2       0     24 0.133    0.488   0.516       0.554  0.0471
##  7       9    11       1     24 0.284   -0.358  -0.357      -0.423  0.0710
##  8       3     3       0     25 0.317    0.980   1.02        1.23   0.708
##  9      15    12       1     25 0.246   -0.619  -0.624      -0.719  0.168
## 10       0     2       0     26 0.123   -1.49   -1.21       -1.30   0.236
## 11      20     7       1     26 0.226    0.186   0.185       0.210  0.0129
## 12       1     3       0     27 0.275   -1.22   -1.20       -1.41   0.760
## 13      20     2       1     27 0.224    1.25    1.15        1.31   0.494
## 14      15     4       1     28 0.232   -1.09   -1.19       -1.35   0.554
## 15       1     0       0     29 0.0702   0.723   0.547       0.567  0.0243
## 16      10     0       1     29 0.131    1.27    0.914       0.981  0.145
## 17       6     0       1     30 0.0754   0.769   0.550       0.572  0.0267
## 18       2     0       1     31 0.0223   0.347   0.247       0.250  0.00142
## 19       1     0       1     33 0.00738  0.149   0.105       0.106  0.0000833
```

```
(model_dfbetas <- dfbetas(m) %>%
   as_tibble())
```

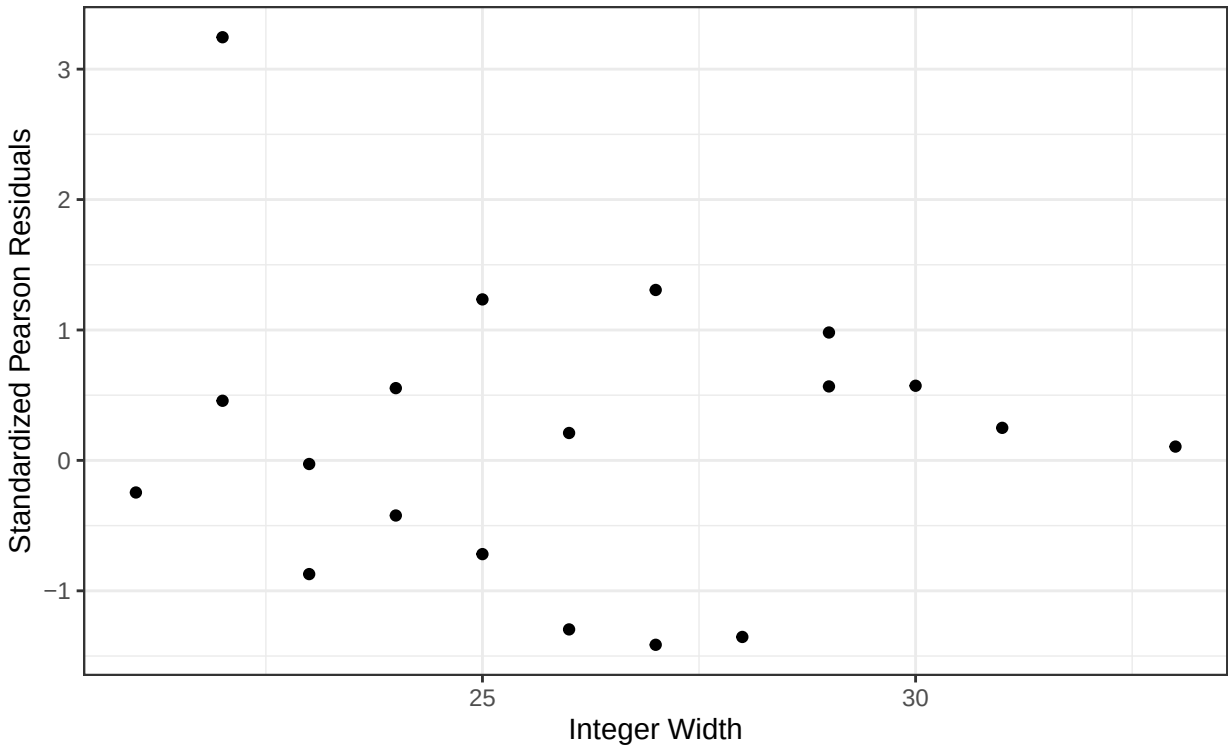
```
## # A tibble: 19 x 3
##    `(Intercept)`   NotDark  Iwidth
##            <dbl>     <dbl>   <dbl>
##  1       -0.0398  0.0342    0.0335
##  2        0.307  -0.328    -0.246
##  3        0.170   0.0355   -0.173
##  4       -0.284   0.399     0.209
##  5       -0.0114 -0.00337   0.0116
##  6        0.0845 -0.172    -0.0512
##  7       -0.144  -0.0737    0.147
##  8        0.195  -0.728    -0.0513
##  9       -0.0691 -0.137     0.0703
## 10       -0.0248  0.552    -0.0857
## 11       -0.0351  0.0360    0.0357
## 12        0.136   0.768    -0.292
## 13       -0.497   0.200     0.506
## 14        0.538  -0.134    -0.548
## 15       -0.0890 -0.144     0.119
## 16       -0.455   0.0814    0.463
## 17       -0.198   0.0275    0.201
## 18       -0.0461  0.00523   0.0469
## 19       -0.0114  0.000941  0.0116
```

Plots of standardized Pearson residuals

```
ggplot(model_aug, aes(x=NotDark, y=std.p.resid))+
  geom_point()+
  labs(x="Not Dark", y="Standardized Pearson Residuals")
```



```
ggplot(model_aug, aes(x=Iwidth, y=std.p.resid))+
  geom_point()+
  labs(x="Integer Width", y="Standardized Pearson Residuals")
```



Observation 2 is the only observation with a standardized Pearson residual exceeding 2 in absolute value. This observation is quite unusual since this binomial experiment only contained one trial and had satellites despite its low shell width. However, this observation actually does not have a large influence when compared to other observations such as 8 and 12.

The `ci` value for observation 2 is not as large as the `ci` value for observations 8 and 12, and the absolute value of the `dfbetas` values for observation 2 are not as large in absolute value compared to the `dfbetas` values for observations 8 and 12.