Homework 5

Ericka Smith

10/27/2020

Problem 1

Part A load data:

```
frogs <- data.frame(
  num_toes_removed = c(rep(1, 4), rep(2, 51), rep(3, 150), rep(4, 245), rep(5, 190), rep(6, 80),
                       rep(7, 12), 8),
  size = as.factor(c(rep("less",44), rep("greater", 11), rep("less", 120), rep("greater", 30),
                     rep("less", 180), rep("greater", 65), rep("less", 120), rep("greater", 70),
                     rep("less", 40), rep("greater", 40), rep("less", 4), rep("greater", 9))),
  recaptured = as.factor(c(rep("y", 3), "n", rep("y", 26), rep("n", 14), rep("y", 7),
                           rep("n",4), rep("y", 80), rep("n", 40), rep("y", 9), rep("n",21),
                           rep("y", 110), rep('n', 70), rep("y", 17), rep("n", 48),
                           rep("y", 68), rep("n", 52), rep("y", 17), rep("n", 53),
                           rep("y", 20),rep("n", 20), rep("y", 9), rep("n", 31), "y",
                           rep("n", 3), "y", rep("n",8))))
```

fit model:

```
fit <- glm(recaptured ~size, data=frogs, family=binomial)
summary(fit)
```

```
##
## Call:
## glm(formula = recaptured ~ size, family = binomial, data = frogs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.365  -1.365   1.000   1.000   1.626
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.0116     0.1508  -6.710 1.94e-11 ***
## sizeless      1.4434     0.1760   8.201 2.38e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1016.14  on 732  degrees of freedom
```

```
## Residual deviance:  942.06  on 731  degrees of freedom
## AIC: 946.06
##
## Number of Fisher Scoring iterations: 4
```

Based on the output there is an effect of size on return rate. For the coefficient for size, $p = 2.38 * 10^{-16}$, which is significant at $\alpha = 0.05$

Part B fit model:

```
fitb <- glm(recaptured ~ num_toes_removed, data=frogs, family=binomial)
summary(fitb)
```

```
##
## Call:
## glm(formula = recaptured ~ num_toes_removed, family = binomial,
##     data = frogs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6344  -1.2046   0.7812   1.1505   1.5769
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        1.35275    0.28624   4.726 2.29e-06 ***
## num_toes_removed -0.32225    0.06619  -4.868 1.13e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1016.14  on 732  degrees of freedom
## Residual deviance:  991.32  on 731  degrees of freedom
## AIC: 995.32
##
## Number of Fisher Scoring iterations: 4
```

Based on our model output there is an effect of toe clipping. The coefficient for number of toes removed is significant at $\alpha = 0.05$, with $p = 1.13 * 10^{-6}$

In order to estimate and interpret this I consider the coefficient itself. The estimate for the coefficient is -0.32225, which can be interpreted as the expected change in log odds for a one-unit increase in number of toes removed. The odds ratio then can be calculated by exponentiating this value to get 0.724517, and taking $1 - 0.724517 = 0.275483 \approx 27.5\%$, which means that **we expect to see about 27.5% decrease in the odds of recapture for each additional toe clipped.**

Part C To fit this model to the data set I will have $\beta_0 = log(R(0))$ as my intercept and $\beta_1 = log(1 + m)$ as a coefficient for n the covariate.

Part D First change "y"s and "n"s to 0s and 1s

```r
frogs01 <- frogs %>%
  mutate(recaptured = recode(recaptured,
                             "y"="1",
                             "n"="0"))
```

Now fit the model

```r
fitd <- glm(as.integer(recaptured) ~ num_toes_removed, data=frogs01, family=poisson(link="log"))
summary(fitd)
```

```
##
## Call:
## glm(formula = as.integer(recaptured) ~ num_toes_removed, family = poisson(link = "log"),
##     data = frogs01)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6297  -0.4446   0.1706   0.3773   0.5740
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.62204    0.11001   5.655 1.56e-08 ***
## num_toes_removed -0.05203    0.02580  -2.017   0.0437 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 124.48  on 732  degrees of freedom
## Residual deviance: 120.41  on 731  degrees of freedom
## AIC: 1816.2
##
## Number of Fisher Scoring iterations: 4
```

$$\beta_0 = 0.62204 = e^{log(R(n))} \implies R(n) = e^{0.62204} = 1.862724$$

$$\beta_1 = -0.05203 = log(1+m) \implies m = e^{-0.05203} - 1 = -0.0506996$$

The estimate for the effect of toe clipping that I get is a reduction in return rate by a constant proportion of 0.0506996, significant at $\alpha = 0.05$

Part E I will choose the second model to describe the variations in the data because, despite it's complexity, the interpretation is much easier to explain as compared to the first model. The idea of a constant change in proportion is much more clearly understandable by the average person.

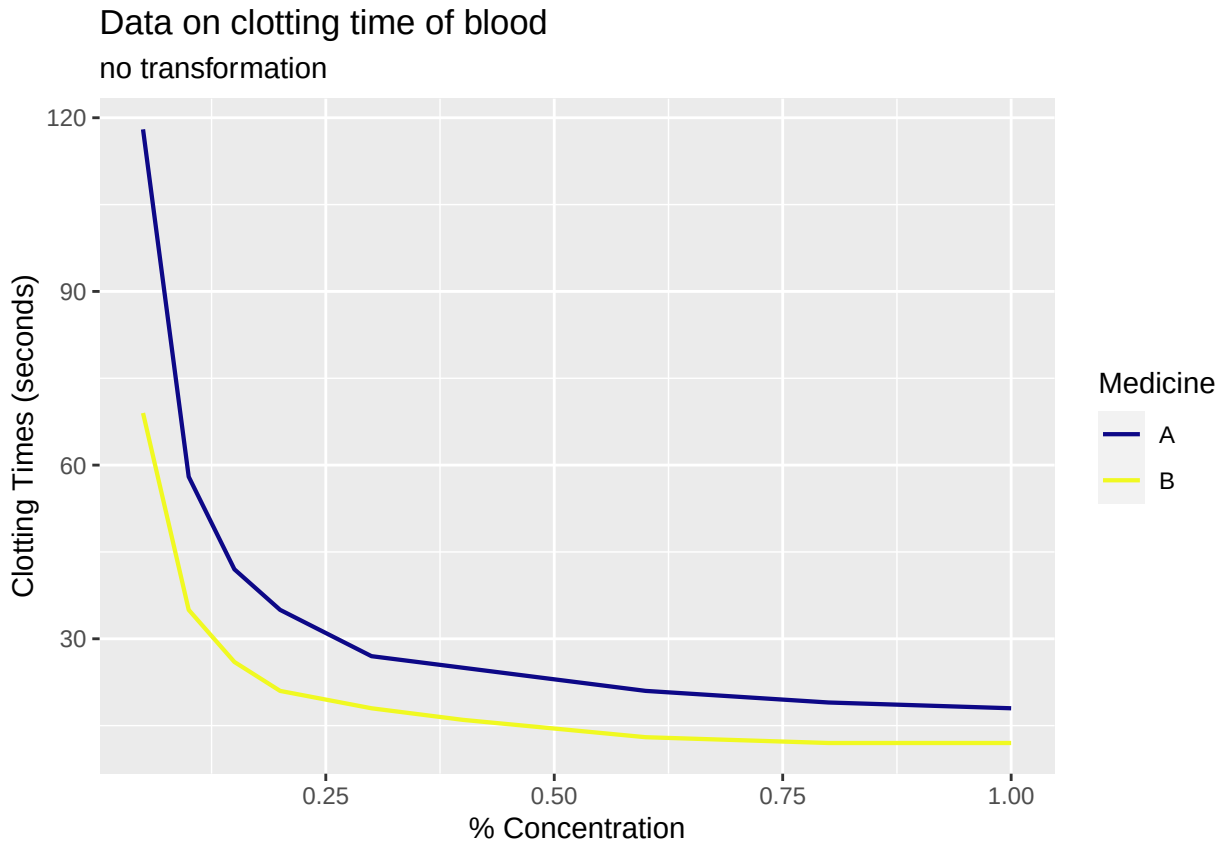
Problem 2

Part A Load data:

```
blood <- data.frame(plasma_pct = rep(c(.05, .10, .15, .20,.30, .40, .60, .80, 1), 2),
                        Medicine = as.factor(rep(c("A", "B"), each = 9)),
                        clot_time = c(118, 58, 42, 35, 27, 25, 21, 19, 18,69, 35, 26, 21, 18, 16, 13, 12
```

Create Plots:

```
ggplot(data=blood)+
  geom_line(aes(x=plasma_pct, y=clot_time, color=Medicine), size=0.75)+
  theme_grey()+
  labs(title="Data on clotting time of blood",
       subtitle = "no transformation")+
  xlab("% Concentration")+
  ylab("Clotting Times (seconds)")+
  scale_color_viridis(discrete = TRUE, option="plasma")
```




```
ggplot(data=blood)+
  geom_line(aes(x=log(plasma_pct), y=(1/clot_time), color=Medicine), size=0.75)+
  theme_grey()+
  labs(title="Data on clotting time of blood",
       subtitle = "with transformation")+
  xlab("log(% Concentration)")+
  ylab("1/(Clotting Times)")+
  scale_color_viridis(discrete = TRUE, option = "plasma")
```

In the plot with no transformation I see that Medicine B has lower clotting times for all of the 9 percentage concentrations.

In the plot with both percentage concentration and clotting times transformed I see that Medicine B changes more steeply than Medicine A does between the different concentrations.

Part B Update data:

```
blood_b <- blood %>%
  mutate(yinv = (1/clot_time),
         logpct = log(plasma_pct))
```

Fit model:

```
fit2b <- glm(yinv ~ logpct*Medicine-1, family = Gamma(link = "inverse"), data = blood_b)
summary(fit2b)
```

```
##
## Call:
## glm(formula = yinv ~ logpct * Medicine - 1, family = Gamma(link = "inverse"),
##     data = blood_b)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.56521  -0.10273   0.07791   0.15479   0.24602
```

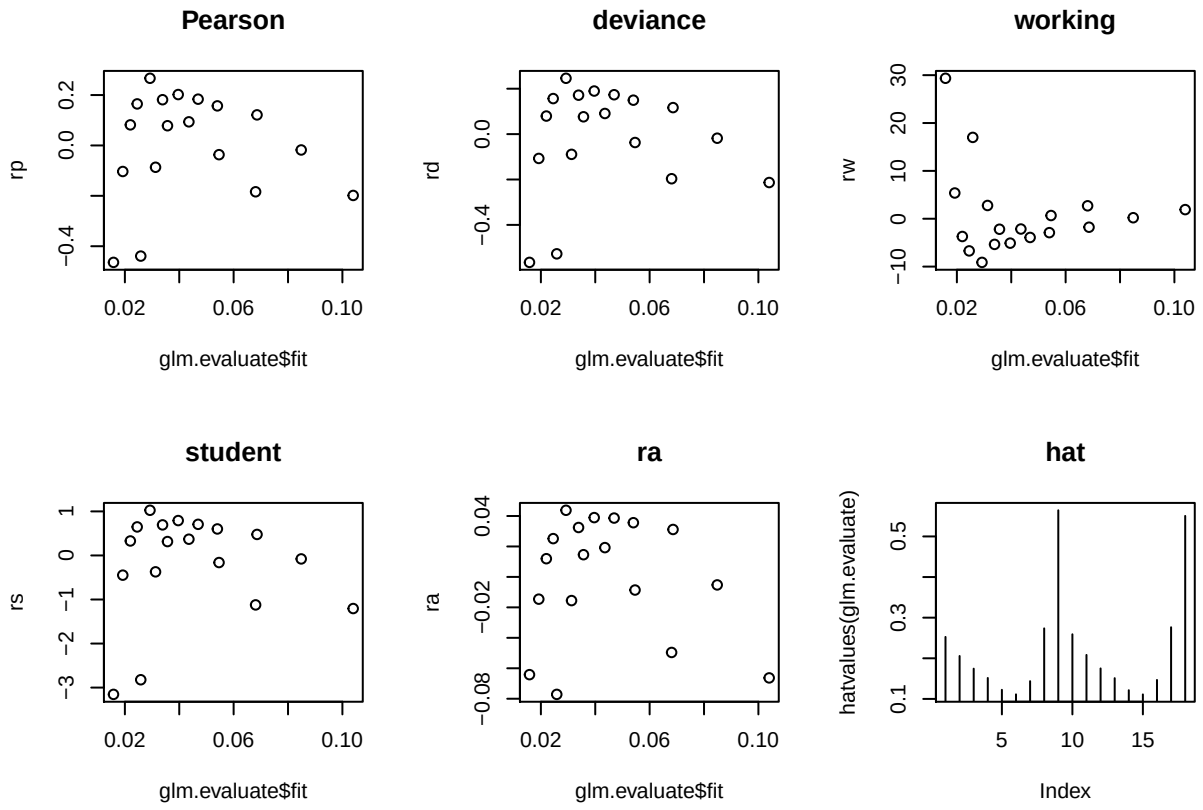
```
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## logpct            -16.205      2.921  -5.548 7.18e-05 ***
## MedicineA          14.688      2.585   5.683 5.65e-05 ***
## MedicineB           9.618      1.672   5.754 4.99e-05 ***
## logpct:MedicineB    6.498      3.445   1.886   0.0802 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.05485759)
##
##     Null deviance:     NaN  on 18  degrees of freedom
## Residual deviance: 0.94007  on 14  degrees of freedom
## AIC: -109.52
##
## Number of Fisher Scoring iterations: 4
```

Residual Analysis:

```
glm.evaluate=fit2b

rp=resid(glm.evaluate, "pearson")
rd=resid(glm.evaluate, "deviance")
rw=resid(glm.evaluate, "working")
rs=rstudent(glm.evaluate)
ra=3*(glm.evaluate$y^{2/3}-glm.evaluate$fit^{2/3})/glm.evaluate$fit^{1/6}/2

par(mfrow=c(2,3))
plot(glm.evaluate$fit,rp, main="Pearson")
plot(glm.evaluate$fit,rd, main = "deviance")
plot(glm.evaluate$fit,rw, main="working")
plot(glm.evaluate$fit,rs, main="student")
plot(glm.evaluate$fit,ra, main="ra")
plot(hatvalues(glm.evaluate), type="h", main="hat")
```



Based on the residual analysis plots it seems that the proposed model is a decent fit. The deviance residuals are small and that indicates a good fit. The working residuals are a bit concerning but I still think it's overall a good fit.

Part C Based on the model output the coefficients for Medicine A and for Medicine B are both significant at $\alpha = 0.05$ ($p = 5.65 * 10^{-5}$ and $p = 4.99 * 10^{-5}$, respectively), so I can interpret the coefficients.

Since this is a log-linked gamma GLM I first exponentiate the coefficients, to get $e^{\beta_A} = e^{14.688} = 2392860$ and $e^{\beta_B} = e^{9.618} = 15032.95$ Now taking $2392860/15032.95 = 159.1743 \approx 159$ we get that Medicine A is approximately 159 times more potent than medicine B.