Homework 3

Ericka Smith

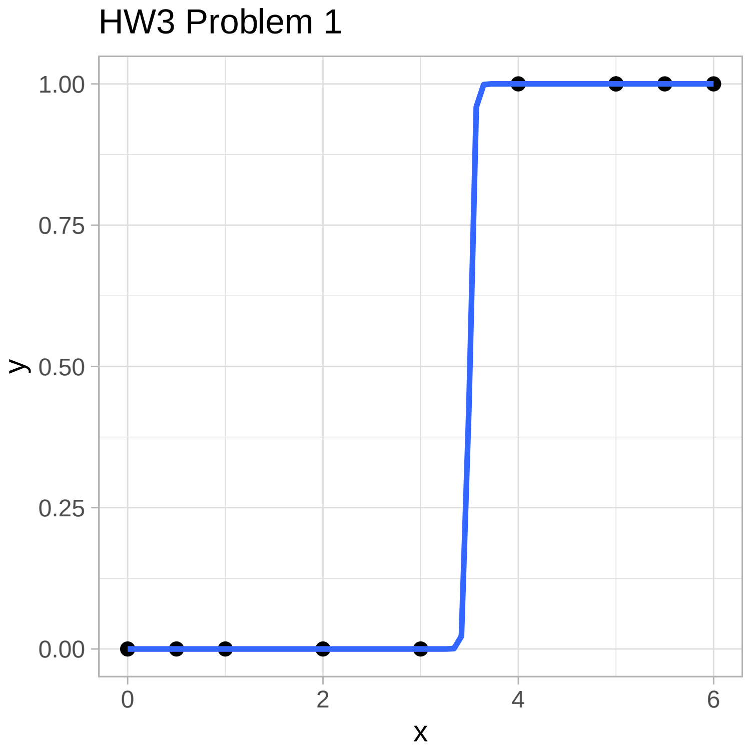
10/14/2020

Problem 1

```r
dat <- data.frame(
  x = c(0, 0.5, 1, 2, 3, 4, 5, 5, 5.5, 6),
  y = c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1))

ggplot(data = dat, aes(x = x, y=y))+
  geom_point(size = 2)+
  geom_smooth(method = "glm", method.args = list(family=binomial), se=F)+
  theme_light()+
  ggtitle("HW3 Problem 1")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
summary(glm(y~x, family = binomial, data=dat))
```

```
##
## Call:
## glm(formula = y ~ x, family = binomial, data = dat)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -1.575e-05  -2.110e-08   0.000e+00   2.110e-08   1.607e-05
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -159.55  270391.50  -0.001        1
## x               45.58   76267.22   0.001        1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.3863e+01  on 9  degrees of freedom
## Residual deviance: 5.0616e-10  on 8  degrees of freedom
## AIC: 4
##
## Number of Fisher Scoring iterations: 25
```

For both the intercept and the coefficient for x we get p-values equal to 1, so neither estimate is significant and we can't interpret their values in a very meaningful way. That being said, in a logistic regression output such as this the intercept signifies the log odds of being in the $p(x) = 0$ group for y, and the coefficient of x signifies the change in the log odds of being in the $p(x) = 1$ group as compared to the $p(x) = 0$ group.

Problem 2

Part A They are equivalent as parameter points because the constant c falls away in the model, because $\alpha_i = log(w_i)$ and and $+c$ is the only difference between $\alpha_1, ..., \alpha_8$ as compared to $\alpha_1 + c, ..., \alpha_8 + c$. We can still estimate the weights $w_1, ..., w_8$ because we have the relationship between these already.

```
X <- read_csv("p2.csv", col_names = FALSE)
```

Part B

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   X2 = col_double(),
##   X3 = col_double(),
##   X4 = col_double(),
##   X5 = col_double(),
##   X6 = col_double(),
##   X7 = col_double()
## )
```

```r
(xmat <- as.matrix(X))
```

```
##       X1 X2 X3 X4 X5 X6 X7
##  [1,]  1 -1  0  0  0  0  0
##  [2,]  1  0 -1  0  0  0  0
##  [3,]  1  0  0 -1  0  0  0
##  [4,]  1  0  0  0 -1  0  0
##  [5,]  1  0  0  0  0 -1  0
##  [6,]  1  0  0  0  0  0 -1
##  [7,]  1  0  0  0  0  0  0
##  [8,]  0  1 -1  0  0  0  0
##  [9,]  0  1  0 -1  0  0  0
## [10,]  0  1  0  0 -1  0  0
## [11,]  0  1  0  0  0 -1  0
## [12,]  0  1  0  0  0  0 -1
## [13,]  0  1  0  0  0  0  0
## [14,]  0  0  1 -1  0  0  0
## [15,]  0  0  1  0 -1  0  0
## [16,]  0  0  1  0  0 -1  0
## [17,]  0  0  1  0  0  0 -1
## [18,]  0  0  1  0  0  0  0
## [19,]  0  0  0  1 -1  0  0
## [20,]  0  0  0  1  0 -1  0
## [21,]  0  0  0  1  0  0 -1
## [22,]  0  0  0  1  0  0  0
## [23,]  0  0  0  0  1 -1  0
## [24,]  0  0  0  0  1  0 -1
## [25,]  0  0  0  0  1  0  0
## [26,]  0  0  0  0  0  1 -1
## [27,]  0  0  0  0  0  1  0
## [28,]  0  0  0  0  0  0  1
```

```r
y <- c(5,3,7,6,6,1/3,1/4,1/3,5,3,3,1/5,1/7,6,3,4,6,1/5,1/3,1/4,1/7,1/8,1/2,1/5,1/6,1/5,1/6,1/2)
logy <- map_dbl(y, ~log(.x))
model0 <- lm(y~xmat)
summary(model0)
```

```
## 
## Call:
## lm(formula = y ~ xmat)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -2.7216 -0.7734  0.0876  0.7992  3.1946 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   2.4300     0.5442   4.465 0.000237 ***
## xmatX1       -0.6107     1.1938  -0.512 0.614550    
## xmatX2       -2.6166     1.0884  -2.404 0.026020 *  
## xmatX3       -0.8603     0.9904  -0.869 0.395366    
## xmatX4       -4.3797     0.9024  -4.853 9.64e-05 ***
## xmatX5       -3.0619     0.8275  -3.700 0.001417 ** 
## xmatX6       -2.6940     0.7696  -3.501 0.002252 **
## xmatX7       -1.2356     0.7326  -1.687 0.107222
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.44 on 20 degrees of freedom
## Multiple R-squared:  0.7459, Adjusted R-squared:  0.6569
## F-statistic: 8.386 on 7 and 20 DF,  p-value: 8.272e-05
```

Here are the estimates I obtain:

$\alpha_1 - \alpha_2 = -0.6107$

$\alpha_1 - \alpha_3 = -2.6166$

$\alpha_1 - \alpha_4 = -0.8603$

$\alpha_1 - \alpha_5 = -4.3797$

$\alpha_1 - \alpha_6 = -3.0619$

$\alpha_1 - \alpha_7 = -2.6940$

$\alpha_1 - \alpha_8 = -1.2356$

```
exp_a <- unname(exp(model0$coefficients))
w <- rbind(diag(-1,7,7), rep(1,7))
(w <- cbind(w, as.vector(exp_a)))
```

Part C

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]        [,8]
## [1,]   -1    0    0    0    0    0    0 11.35907526
## [2,]    0   -1    0    0    0    0    0  0.54296290
## [3,]    0    0   -1    0    0    0    0  0.07305060
## [4,]    0    0    0   -1    0    0    0  0.42303976
## [5,]    0    0    0    0   -1    0    0  0.01252887
## [6,]    0    0    0    0    0   -1    0  0.04679648
## [7,]    0    0    0    0    0    0   -1  0.06760818
## [8,]    1    1    1    1    1    1    1  0.29065181
```

```
(z <- c(rep(0,7), 1))
```

```
## [1] 0 0 0 0 0 0 0 1
```

```
solve(w,z)
```

```
## [1] 0.8863396439 0.0423669649 0.0057000807 0.0330094571 0.0009776181
## [6] 0.0036514923 0.0052754124 0.0780292078
```

The estimates of the weights are the following:

$w_1 = 0.8863396439$

$w_2 = 0.0423669649$

$w_3 = 0.0057000807$

$w_4 = 0.0330094571$

$w_5 = 0.0009776181$

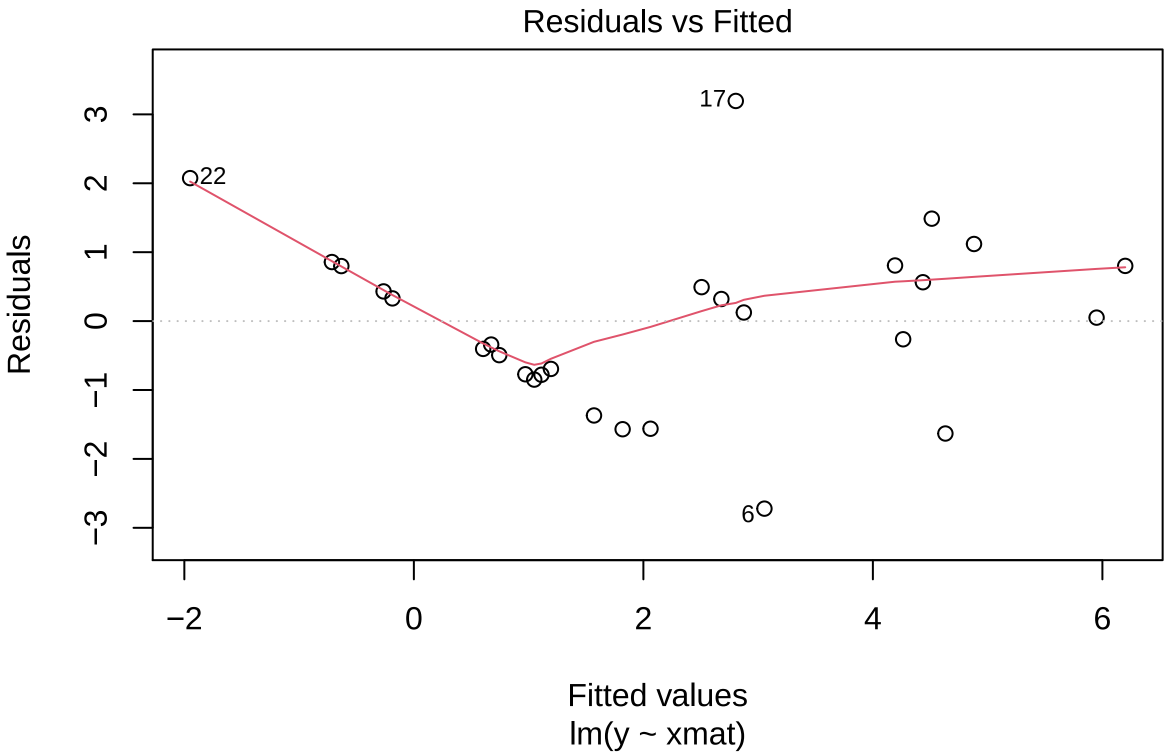
$w_6 = 0.0036514923$

$w_7 = 0.0052754124$

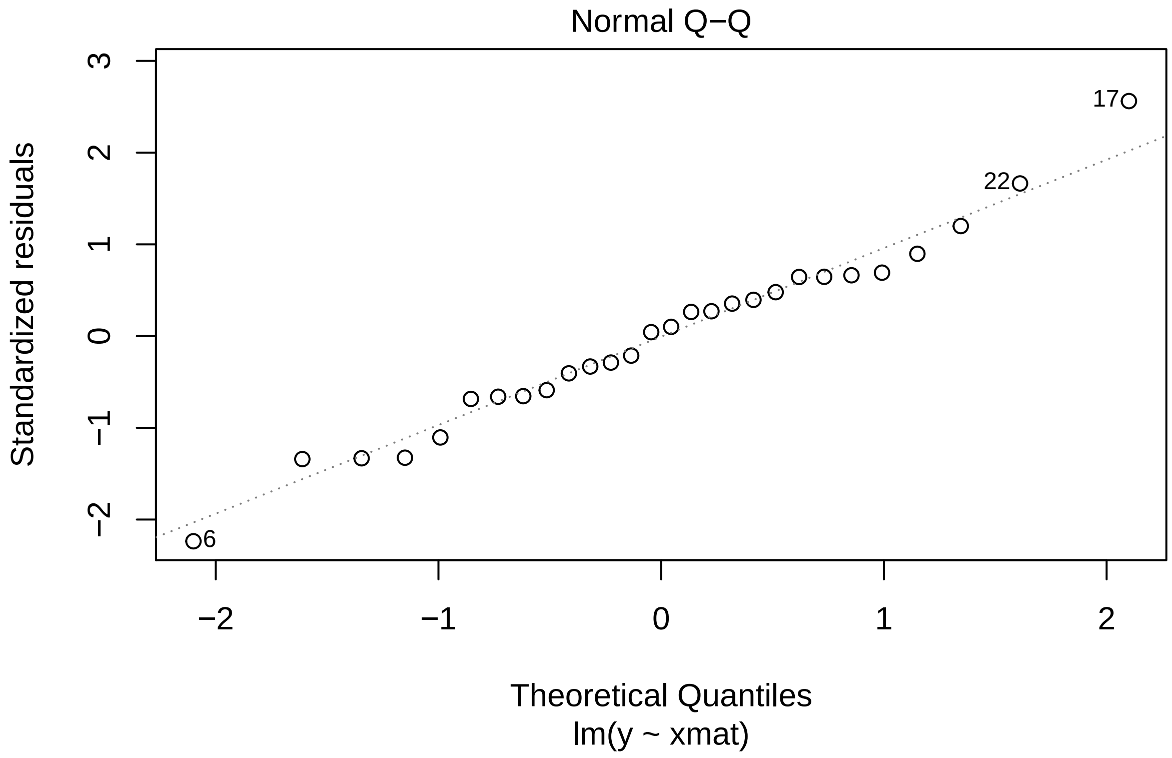
$w_8 = 0.0780292078$

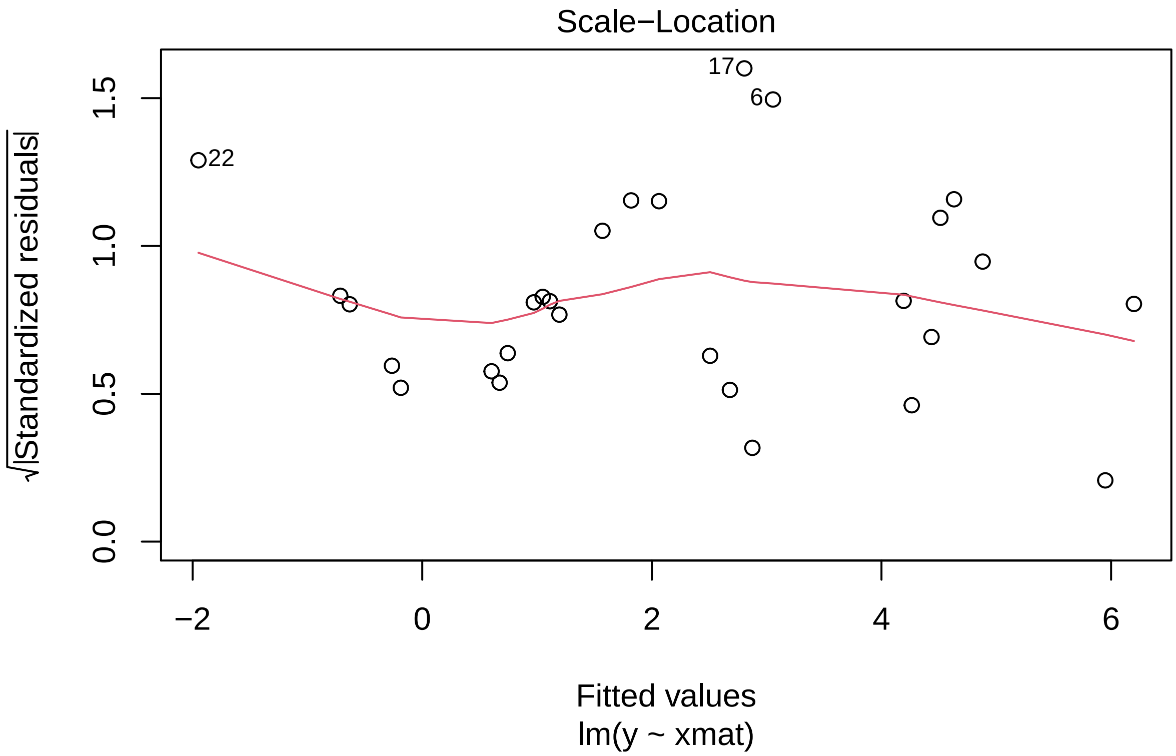
Part D Since only certain α estimates were significant that calls into question the hypothesis that all 8 criteria are equally important. The varying magnitudes of the weights also do. For example, w_5 is a LOT smaller than the other weights, which leads me to believe that yard space (5) is not actually as important as some of the other criteria.

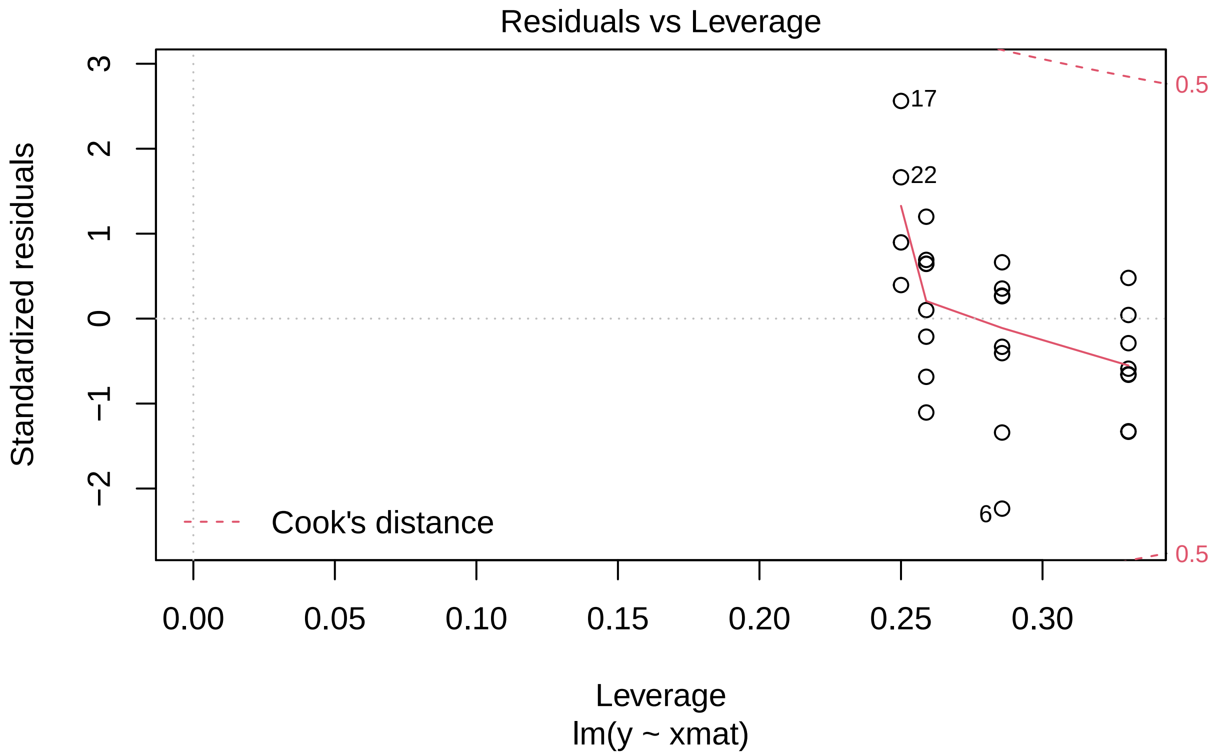
```
plot(model0)
```


Residuals vs Fitted
Residuals
Fitted values
lm(y ~ xmat)
−3
−2
−1
0
1
2
3
−2
0
2
4
6
17
22
6

Part E


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(y ~ xmat)
−2
−1
0
1
2
3
−2
−1
0
1
2
17
22
6


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
−2
0
2
4
6
Fitted values
lm(y ~ xmat)
17
6
22



Based on the plots the model assumptions are not quite met. In the Residuals vs. Fitted plot you can see that there is nonconstant variance and on the Q-Q plot there is a bit of tailing off at the upper right hand corner of the plot.

Problem 3

```
# dat <- data.frame(
#   insecticide = c(2, 2.64, 3.48, 4.59, 6.06, 8),
#   ddt = c(3/50, 5/49, 19/47, 19/38, 24/49, 35/50),
#   bhc = c(2/50, 14/49, 20/50, 27/50, 41/50, 40/50),
#   both = c(28/50, 37/50, 46/50, 48/50, 48/50, 50/50))
df <- data.frame(
  dose <- c(2, 2.64, 3.48, 4.59, 6.06, 8, 2, 2.64, 3.48, 4.59, 6.06, 8, 2, 2.64, 3.48, 4.59, 6.06, 8),
  kr <- c(3/50, 5/49, 19/47, 19/38, 24/49, 35/50, 2/50, 14/49, 20/50, 27/50, 41/50, 40/50, 28/50, 37/50
  insecticide <- c("DDT", "DDT", "DDT", "DDT", "DDT", "DDT", "BHC", "BHC", "BHC", "BHC", "BHC", "BHC",

ggplot(data = df)+
  geom_smooth(aes(x = dose, y=kr, color=insecticide), method = "glm", method.args = list(family=binomial
  geom_point(aes(x = dose, y=kr, color=insecticide))+
  theme_light()+
  ggtitle("Toxicity of Insecticides")+
  xlab(as.expression(bquote("Insecticide Dose (mg/10" ~ cm^2 ~ ")")))+
  ylab("Kill Rate")
```

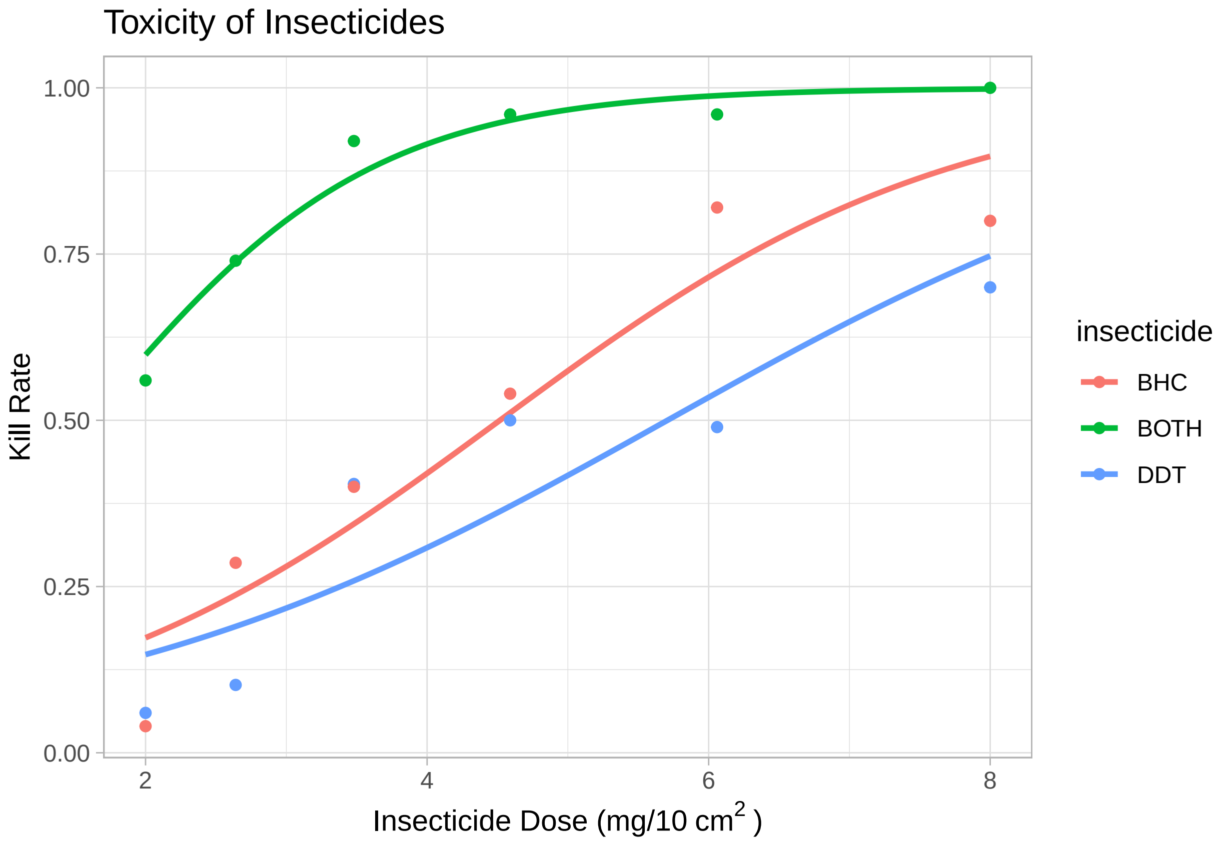
Parts A and B

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```



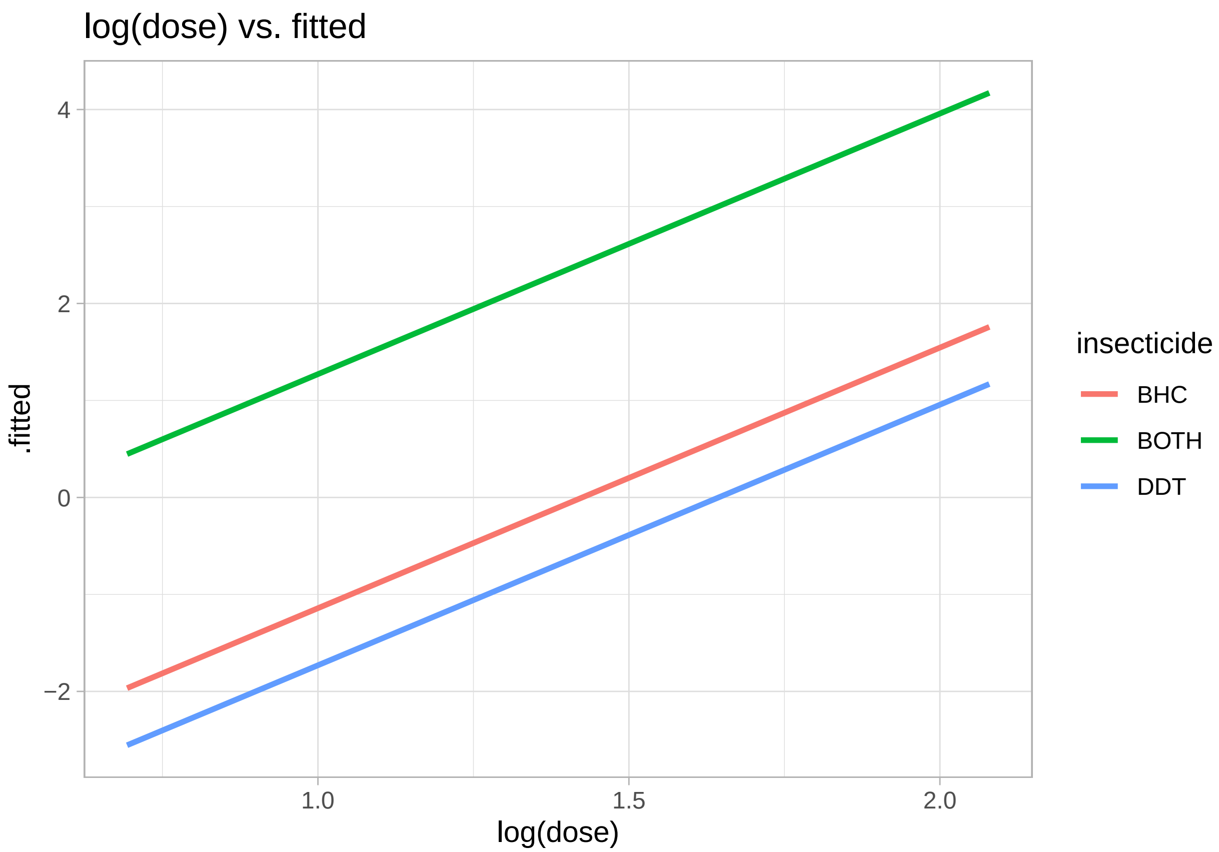
```
mod <- glm(kr ~ log(dose) + insecticide, family="binomial")
```

Part C

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
ggplot(mod)+
  geom_smooth(aes(log(dose), .fitted, color = insecticide), se = F)+
  theme_light()+
  ggtitle("log(dose) vs. fitted")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Looking at the plot of log(dose) vs. fitted values we see that these lines are parallel. This gives evidence to the hypothesis of parallelism.

Part D In the formula chem+ldose there is an intercept term and in the formula chem+ldose-1 there is not. Since this is the only difference the covariance matrices will be the same.

```
a = mod$coefficients[1]
b = mod$coefficients[2]
V=vcov(mod)
z = 1.645;
k = 1.645^2*V[2,2]/b^2
(tau = polyroot(c(a^2 - V[1,1]*z*z,2*a*b - 2*V[1,2]*z*z,b^2 - V[2,2]*z*z)))
```

Part E

```
## [1] -0.0048049+0i  3.0788096+0i
```

The 90% Confidence Interval that I obtain is (-0.0048049, 3.0788096)

```
mod_probit <- glm(kr ~ log(dose) + insecticide, family=binomial(link = "probit"))
```

Part F

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
mod_cloglog <- glm(kr ~ log(dose) + insecticide, family=binomial(link = "cloglog"))
```

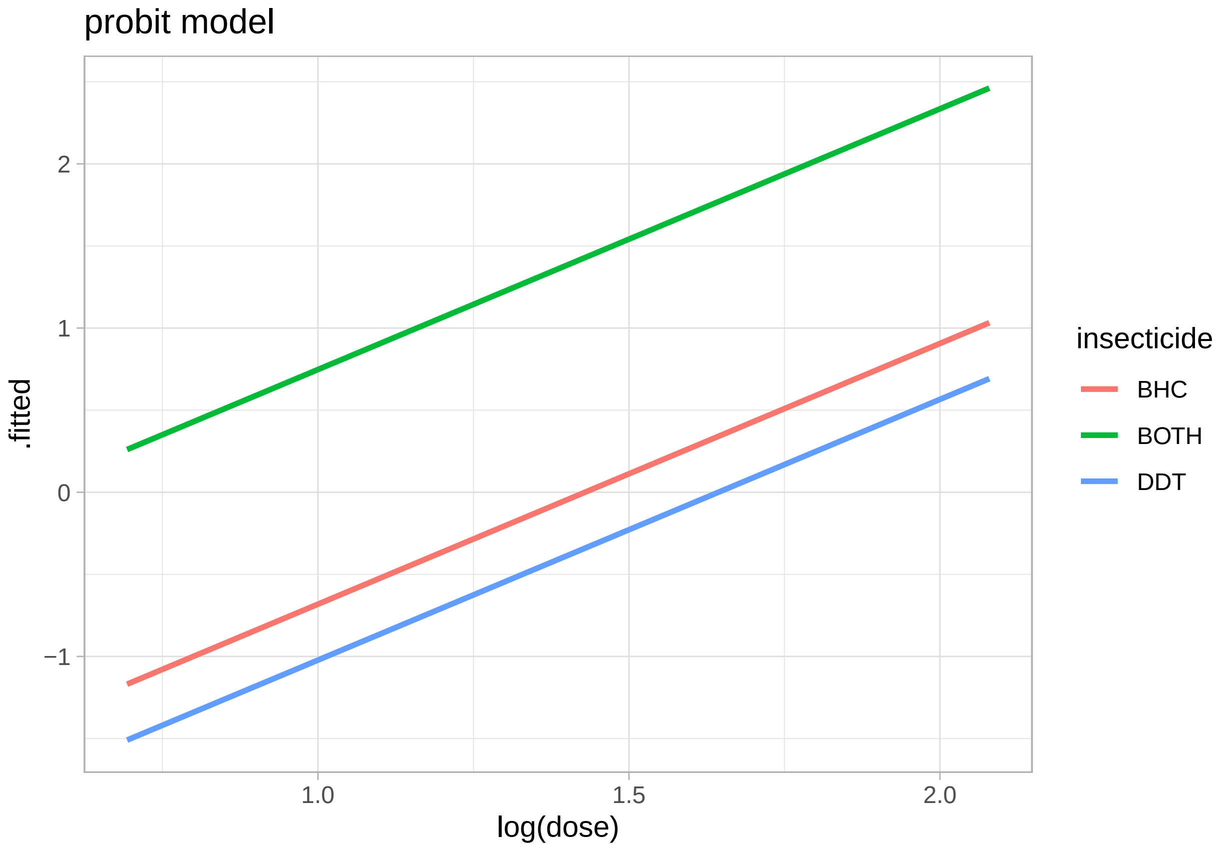
```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
mod_loglog <- glm((1-kr) ~ log(dose) + insecticide, family=binomial(link="cloglog"))
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

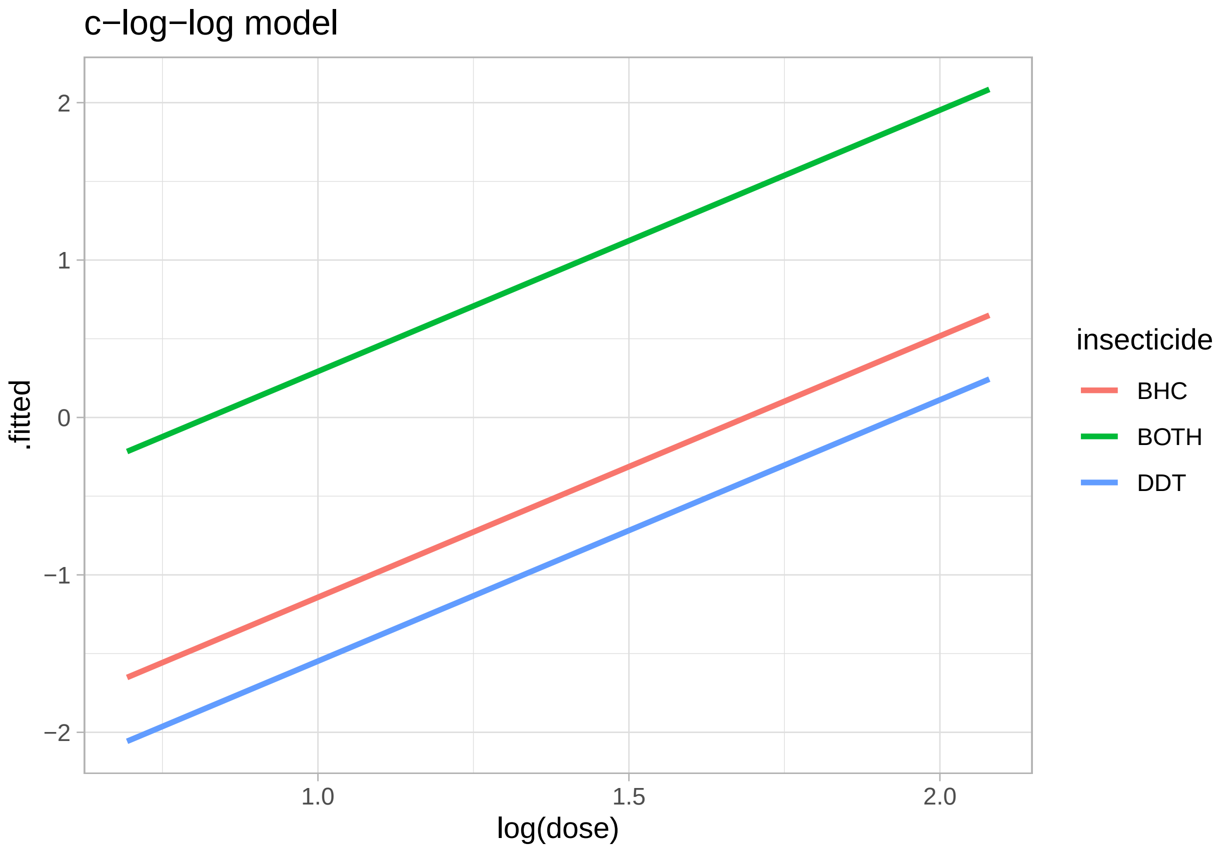
```
ggplot(mod_probit)+
  geom_smooth(aes(log(dose), .fitted, color = insecticide), se = F)+
  theme_light()+
  ggtitle("probit model")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



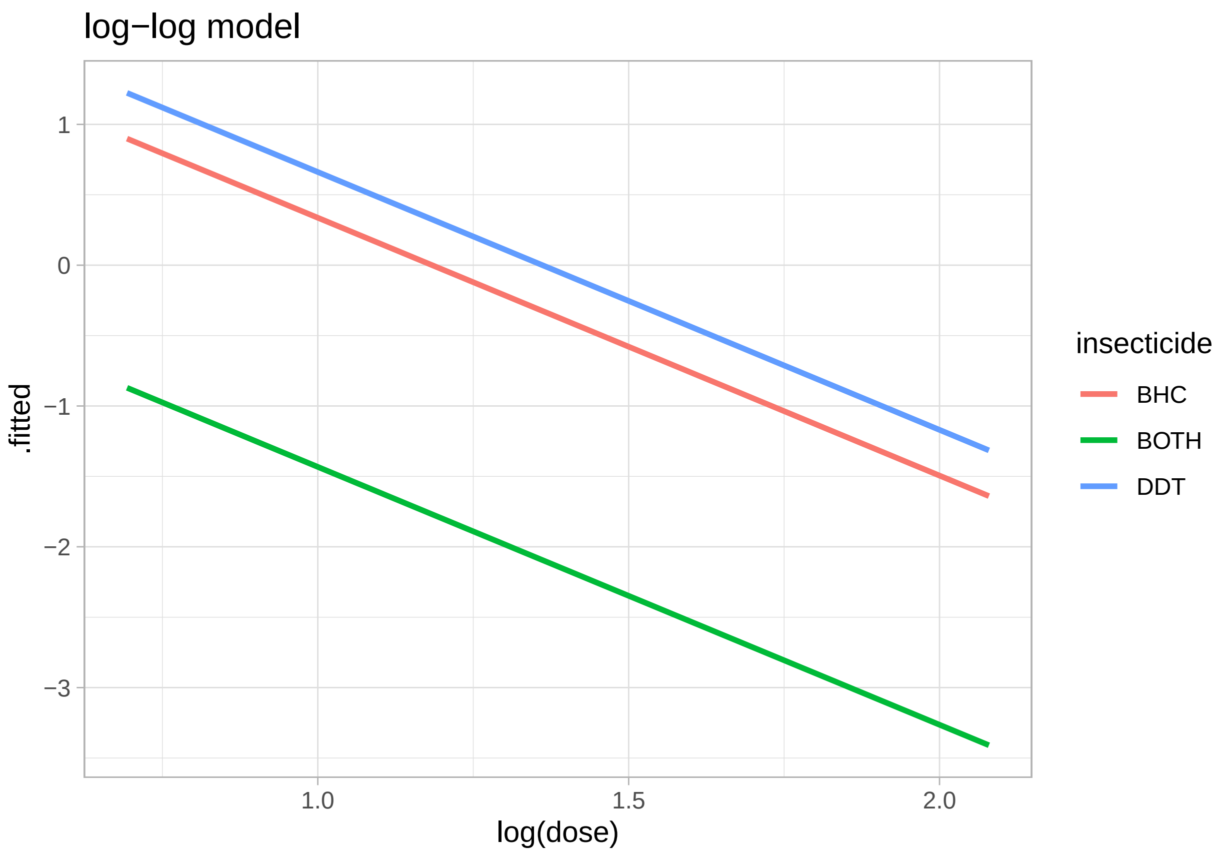
```
ggplot(mod_cloglog)+
  geom_smooth(aes(log(dose), .fitted, color = insecticide), se = F)+
  theme_light()+
  ggtitle("c-log-log model")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
ggplot(mod_loglog)+
  geom_smooth(aes(log(dose), .fitted, color = insecticide), se = F)+
  theme_light()+
  ggtitle("log-log model")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
a = mod_cloglog$coefficients[1]
b = mod_cloglog$coefficients[2]
V=vcov(mod_cloglog)
z = 1.645;
k = 1.645^2*V[2,2]/b^2
(tau = polyroot(c(a^2 - V[1,1]*z*z,2*a*b - 2*V[1,2]*z*z,b^2 - V[2,2]*z*z)))
```

```
## [1] 0.532220-0i 4.451669+0i
```

It really doesn't seem to me that any of them gives an appreciably better fit. For my 90% confidence interval for the c-log-log model I get (0.532220,4.451669).

```
predict(mod, type = "response")[18]
```

Part G

```
##        18
## 0.9848035
```

```
exp(predict(mod)[18])
```

```
##       18
## 64.80447
```

The closest we can reliably get to 0.99 is 0.98480346, and this occurs at a dose of $64.8mg/10cm^2$

Part H It seems pretty clear to me based upon the prior plots and models that both insecticides together are the most effective. DDT appears to be less effective than gamma-BHC but the two are fairly similar overall.