# Biostat 200C Homework 4

Due May 24 @ 11:59PM

```
library(faraway)
library(ggplot2)
library(MASS)
```

## Q1. ELMR Excercise 7.5 (p150)

### (a) Answer:

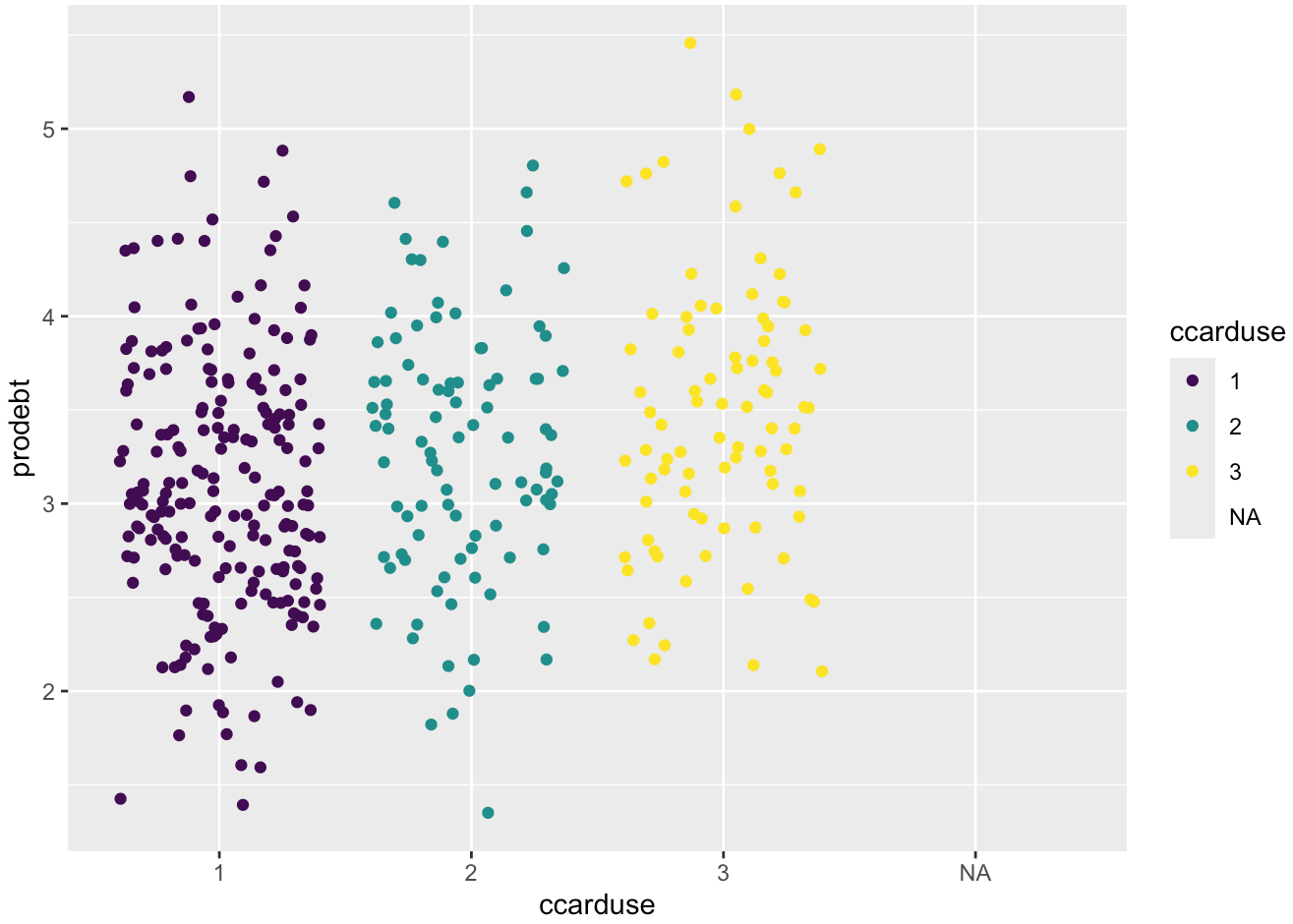
```
data(debt)
```

```
debt$ccarduse <- factor(debt$ccarduse, ordered = TRUE)
```

```
#debt$incomegp <- factor(debt$incomegp, ordered = TRUE)
```

```
ggplot(debt, aes(x = ccarduse, y = prodebt, color = ccarduse)) + geom_jitter()
```

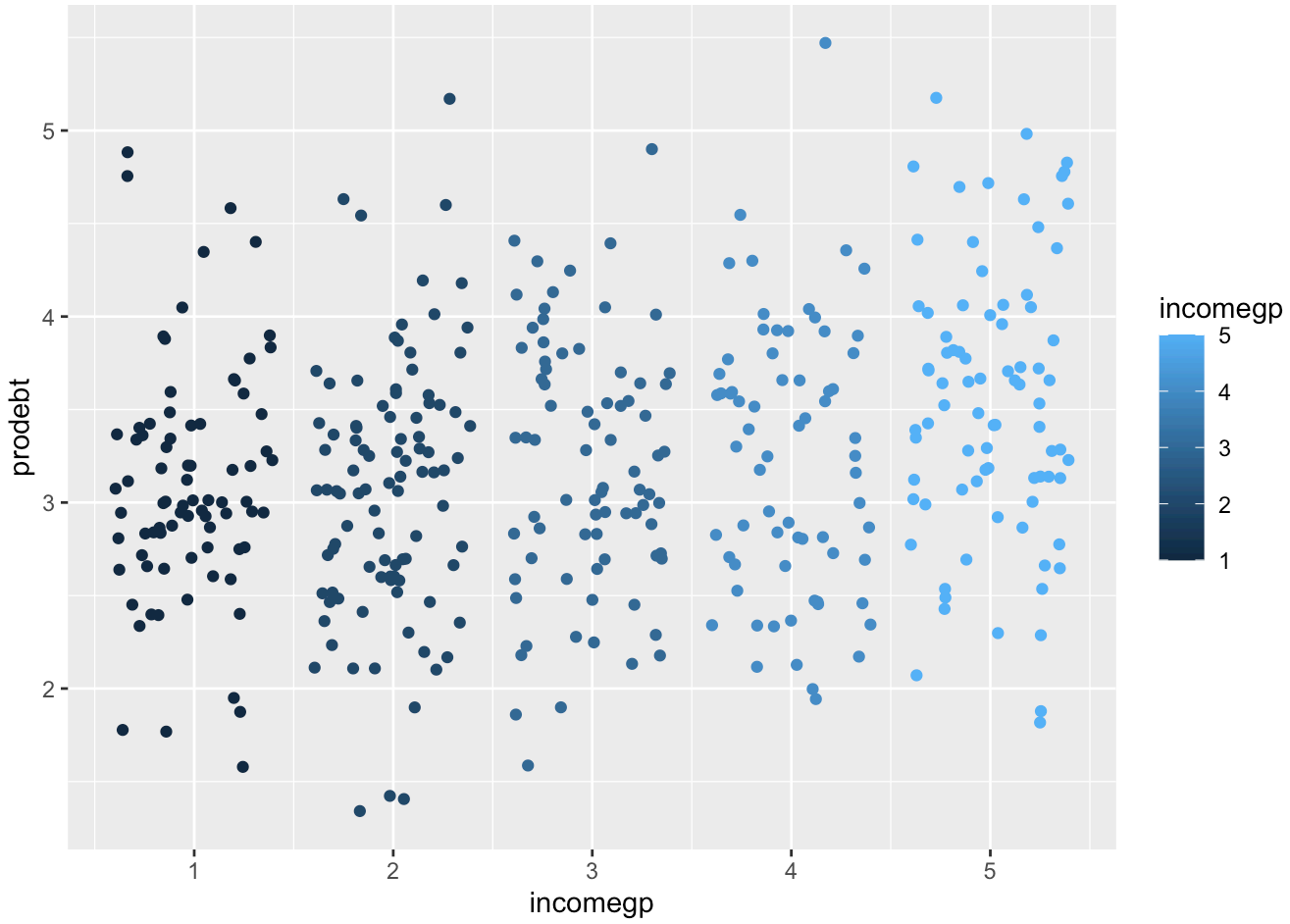
```
Warning: Removed 73 rows containing missing values or values outside the scale range
(`geom_point()`).
```




Observation with high frequency of credit card usage are more favorable to debt.

```
ggplot(debt, aes(x = incomegp, y = prodebt, color = incomegp)) + geom_jitter()
```

```
Warning: Removed 61 rows containing missing values or values outside the scale range
(`geom_point()`).
```




The level of income does not seem to have a strong relationship with the preference to debt.

**(b) Answer:**

```
pomod <- polr(ccarduse ~ ., data = debt)
summary(pomod)
```

```
Re-fitting to get Hessian

Call:
polr(formula = ccarduse ~ ., data = debt)

Coefficients:
            Value Std. Error t value
incomegp  0.47131     0.1061  4.4423
house     0.11600     0.2324  0.4992
children -0.07872     0.1250 -0.6296
singpar   0.88172     0.5971  1.4766
agegp     0.20568     0.1576  1.3050
bankacc   2.10270     0.5934  3.5435
bsocacc   0.47322     0.2671  1.7715
manage    0.18179     0.1653  1.0998
cigbuy   -0.73546     0.2981 -2.4674
```

```
xmasbuy   0.47014     0.4130  1.1385
locintrn  0.11881     0.1424  0.8344
prodebt   0.61046     0.1822  3.3497

Intercepts:
    Value   Std. Error t value
1|2  7.9694  1.4752     5.4023
2|3  9.3944  1.5051     6.2417

Residual Deviance: 511.673
AIC: 539.673
(160 observations deleted due to missingness)
```

The 2 most significant predictors are `incomegp` and `bankacc`.

- `incomegp`: The odds of moving from credit card usage level 1 to credit card usage level 2 or from credit card usage level 2 to credit card usage level 3 increases by a factor of 1.602096 as income increases by one unit.
- `bankacc`: The odds of moving from credit card usage level 1 to credit card usage level 2 or from credit card usage level 2 to credit card usage level 3 increases by a factor of 1.605149 as having bank account compared to not having bank account.

The 2 least significant predictors are `house` and `children`.

**(c) Answer:**

```
pomod_least <- polr(ccarduse ~ house, data = debt)
summary(pomod_least)
```

```
Re-fitting to get Hessian

Call:
polr(formula = ccarduse ~ house, data = debt)

Coefficients:
      Value Std. Error t value
house 0.558     0.1433   3.895

Intercepts:
    Value  Std. Error t value
1|2 1.3000 0.3118     4.1698
2|3 2.4670 0.3274     7.5344

Residual Deviance: 847.2661
AIC: 853.2661
(36 observations deleted due to missingness)
```

```
0.558 - 1.96*0.1433
```

```
[1] 0.277132
```

The predictor seems to be significant in this model. There is contradiction on conclusion between the two models.

**(d) Answer:**

```
#drop missing values

debt_clean <- na.omit(debt)
```

```
pomod_clean <- polr(ccarduse ~ ., data = debt_clean)
```

```
pomod_clean_i <- step(pomod_clean)
```

```
Start:  AIC=539.67
ccarduse ~ incomegp + house + children + singpar + agegp + bankacc +
    bsocacc + manage + cigbuy + xmasbuy + locintrn + prodebt

           Df    AIC
- house     1 537.92
- children  1 538.07
- locintrn  1 538.37
- manage    1 538.89
- xmasbuy   1 539.00
- agegp     1 539.38
<none>        539.67
- singpar   1 539.75
- bsocacc   1 540.83
- cigbuy    1 543.94
- prodebt   1 549.30
- bankacc   1 554.83
- incomegp  1 558.37

Step:  AIC=537.92
ccarduse ~ incomegp + children + singpar + agegp + bankacc +
    bsocacc + manage + cigbuy + xmasbuy + locintrn + prodebt

           Df    AIC
- children  1 536.32
- locintrn  1 536.57
- xmasbuy   1 537.19
- manage    1 537.23
<none>        537.92
- singpar   1 538.01
- agegp     1 538.54
- bsocacc   1 539.14
- cigbuy    1 542.55
- prodebt   1 547.61
- bankacc   1 553.79
```

```
- incomegp  1 557.55

Step:  AIC=536.32
ccarduse ~ incomegp + singpar + agegp + bankacc + bsocacc + manage +
    cigbuy + xmasbuy + locintrn + prodebt

           Df    AIC
- locintrn  1 535.01
- xmasbuy   1 535.34
- manage    1 535.71
- singpar   1 536.23
<none>        536.32
- bsocacc   1 537.47
- agegp     1 538.12
- cigbuy    1 541.09
- prodebt   1 545.83
- bankacc   1 551.97
- incomegp  1 556.19

Step:  AIC=535.01
ccarduse ~ incomegp + singpar + agegp + bankacc + bsocacc + manage +
    cigbuy + xmasbuy + prodebt

           Df    AIC
- xmasbuy   1 534.19
- manage    1 534.58
- singpar   1 534.90
<none>        535.01
- bsocacc   1 536.40
- agegp     1 536.66
- cigbuy    1 539.71
- prodebt   1 543.93
- bankacc   1 551.87
- incomegp  1 555.76

Step:  AIC=534.19
ccarduse ~ incomegp + singpar + agegp + bankacc + bsocacc + manage +
    cigbuy + prodebt

           Df    AIC
- manage    1 533.90
<none>        534.19
- singpar   1 534.32
- bsocacc   1 535.32
- agegp     1 536.11
- cigbuy    1 538.62
- prodebt   1 543.71
- bankacc   1 550.24
- incomegp  1 556.78

Step:  AIC=533.9
```

```
ccarduse ~ incomegp + singpar + agegp + bankacc + bsocacc + cigbuy +
    prodebt

           Df    AIC
- singpar   1 533.59
<none>        533.90
- bsocacc   1 536.04
- agegp     1 536.27
- cigbuy    1 539.16
- prodebt   1 542.16
- bankacc   1 551.27
- incomegp  1 555.32

Step:  AIC=533.59
ccarduse ~ incomegp + agegp + bankacc + bsocacc + cigbuy + prodebt

           Df    AIC
<none>        533.59
- bsocacc   1 535.42
- agegp     1 535.60
- cigbuy    1 538.72
- prodebt   1 542.25
- bankacc   1 549.99
- incomegp  1 553.43
```

```
final_model <- polr(ccarduse ~ incomegp + agegp + bankacc +
                        bsocacc + cigbuy + prodebt, data = debt_clean)
```

```
summary(final_model)
```

```
Re-fitting to get Hessian

Call:
polr(formula = ccarduse ~ incomegp + agegp + bankacc + bsocacc +
    cigbuy + prodebt, data = debt_clean)

Coefficients:
           Value Std. Error t value
incomegp  0.4589     0.1007   4.555
agegp     0.2696     0.1352   1.993
bankacc   2.0816     0.5753   3.618
bsocacc   0.5048     0.2591   1.949
cigbuy   -0.7677     0.2922  -2.627
prodebt   0.5635     0.1755   3.211

Intercepts:
    Value   Std. Error t value
1|2  5.9944  0.9961     6.0178
```

```
2|3  7.3948  1.0276     7.1961

Residual Deviance: 517.5895
AIC: 533.5895
```

```
exp(final_model$coef[1:6])
```

```
 incomegp     agegp   bankacc   bsocacc    cigbuy   prodebt
1.5822669 1.3093912 8.0170914 1.6566115 0.4640734 1.7568894
```

- `incomegp`: The odds of moving from credit card usage level 1 to credit card usage level 2 or from credit card usage level 2 to credit card usage level 3 increases by a factor of 1.582267 as income increases by one unit.
- `agegp`: The odds of moving from credit card usage level 1 to credit card usage level 2 or from credit card usage level 2 to credit card usage level 3 increases by a factor of 1.309391 as age increases by one unit.
- `bankacc`: The odds of moving from credit card usage level 1 to credit card usage level 2 or from credit card usage level 2 to credit card usage level 3 increases by a factor of 8.0170914 as having bank account compared to not having bank account.
- `bsocacc`: The odds of moving from credit card usage level 1 to credit card usage level 2 or from credit card usage level 2 to credit card usage level 3 increases by a factor of 1.6566115 as having building society account compared to not having building society account.
- `cigbuy`: The odds of moving from credit card usage level 1 to credit card usage level 2 or from credit card usage level 2 to credit card usage level 3 increases by a factor of 0.4640734 as the observation buys cigarettes compared to not buying cigarettes.
- `prodebt`: The odds of moving from credit card usage level 1 to credit card usage level 2 or from credit card usage level 2 to credit card usage level 3 increases by a factor of 1.7568894 as prodebt increases by one unit.

We cannot conclude that dropped predictors have no relation to the response. The dropped predictors may have interaction with the remained predictors in the model to affect the response.

**(e) Answer:**

```
l1 = median(debt_clean$incomegp)
l2 = median(debt_clean$agegp)
l3 = median(debt_clean$bankacc)
l4 = median(debt_clean$bsocacc)
l5 = median(debt_clean$prodebt)

predict(final_model, data.frame(incomegp = l1, agegp = l2, bankacc = l3, bsocacc
```

```
        1         2         3
0.6149076 0.2513666 0.1337258
```

```
predict(final_model, data.frame(incomegp = l1, agegp = l2, bankacc = l3, bsocacc
```

```
        1         2         3
0.4256250 0.3247658 0.2496092
```

Row one in the output is the probability for smoker and row 2 is the probability for non-smoker.

**(f) Answer:**

```
mod_hazard = polr(ccarduse ~ incomegp + agegp + bankacc + bsocacc + cigbuy + proc
     method = "cloglog", data = debt_clean)
```

```
predict(mod_hazard, data.frame(incomegp = l1, agegp = l2, bankacc = l3, bsocacc =
```

```
        1         2         3
0.5571469 0.2872181 0.1556350
```

```
predict(mod_hazard, data.frame(incomegp = l1, agegp = l2, bankacc = l3, bsocacc =
```

```
        1         2         3
0.4491074 0.2946605 0.2562321
```

Row one in the output is the probability for smoker and row 2 is the probability for non-smoker.

The general trend when comparing row 1 and row2 in the proportional hazards model is unchanged compared to the proportional odd smodel. Therefore, it does not seem to make a difference to use this type of model

## Q2. Moments of exponential family distributions

Show that the exponential family distributions have moments

$$\mathbb{E}Y = \mu = b'(\theta)$$
$$\operatorname{Var} Y = \sigma^2 = b''(\theta)a(\phi).$$

Denote $f_y = f(y;\theta,\phi) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right)$

$l(\theta) = \log(f_y) = \frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)$

$l'(\theta) = \frac{y - b'(\theta)}{a(\phi)}$

Since true $\theta$ must maximize $l(\theta)$, we have $E(l'(\theta)) = 0$

$E(\frac{y-b'(\theta)}{a(\phi)}) = 0$

$E(Y) = b'(\theta) = \mu$

$l''(\theta) = -\frac{b''(\theta)}{a(\phi)}$

$E(\frac{d^2l}{d\theta^2}) = -E[(\frac{dl}{d\theta})^2]$

$E[-\frac{b''(\theta)}{a(\phi)}] = \frac{-b''(\theta)}{a(\phi)} = -\frac{Var(Y)}{[a(\phi)]^2}$

$Var(Y) = b''(\theta)a(\phi) = \sigma^2$

## Q3. Score and information matrix of GLM

Derive the gradient (score), negative Hessian, and Fisher information matrix (expected negative Hessian) of GLM.

$l(\beta) = \sum \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)$

Assume g is the canonical link function, we have $\theta_i = g(\mu_i) = \eta_i = X^T\beta$

$\nabla l(\beta) = \sum \frac{y_i\frac{d\theta_i}{d\beta} - \frac{db(\theta_i)}{d\beta}\frac{d\theta_i}{d\beta}}{a(\phi)} = \sum \frac{y_i\frac{d\theta_i}{d\beta} - b'(\theta_i)\frac{d\theta_i}{d\beta}}{a(\phi)} = \sum \frac{(y_i - b'(\theta_i))\frac{d\theta_i}{d\beta}}{a(\phi)} = \sum \frac{(y_i-\mu)\mu'}{\sigma^2}x_i$

$-\nabla^2 l(\beta) = \sum \frac{[\mu_i'(\eta_i)]^2}{\sigma_i^2}x_i x_i^T - \frac{(y_i-\mu_i)u_i''(\eta_i)}{\sigma^2} + \frac{(y_i-\mu_i)[\mu_i'(\eta_i)]^2(d\sigma_i^2/d\mu_i)}{\sigma_i^4}x_i x_i^T$

Since $E(y_i) = \mu_i$

$E[-\nabla^2 l(\beta)] = \sum \frac{[\mu_i'(\eta_i)]^2}{\sigma_i^2}x_i x_i^T$

## Q4. ELMR Exercise 8.1 (p171)

**(a) Answer:**

We first rewrite the function.

$f(y) = \lambda e^{-\lambda y} = e^{\log(\lambda) - \lambda y} = e^{-\lambda y + \log(\lambda)}$

$\theta = -\lambda$

$\phi = 1$

$a(\phi) = 1$

$b(\theta) = -\log(-\theta)$

$c(y, \phi) = 0$

**(b) Answer:**

$\mu = b'(\theta) = \frac{1}{\lambda}$

$g(\mu) = g(b'(\theta)) = g(\frac{1}{-\theta}) = \theta = -\frac{1}{\mu} = \eta$

$Var(\mu) = b''(\theta)a(\phi) = \frac{1}{\theta^2} = \mu^2$

**(c) Answer:**

We can end up with negative value for $\lambda$.

**(d) Answer:**

When comparing nested model, a likelihood ratio test should be used which assumed $\chi^2$ distribution. F test should only be used when the models assumed normal assumption.

**(e) Answer:**

$D(y, \hat{\mu}) = 2\sum_{i=1}^{n}\left[y_i(\mu_i - \hat{\mu}_i) - \log(y_i) + b(\hat{\mu}_i)\right]$

## Q5. ELMR Exercise 8.4 (p172)

**(a) Answer:**

```
data(gala,package="faraway")
```

```
mod <- glm(Species ~ . -Endemics, data = gala, family = poisson)
```

```
summary(mod)
```

```
Call:
glm(formula = Species ~ . - Endemics, family = poisson, data = gala)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.155e+00  5.175e-02  60.963  < 2e-16 ***
Area        -5.799e-04  2.627e-05 -22.074  < 2e-16 ***
Elevation    3.541e-03  8.741e-05  40.507  < 2e-16 ***
Nearest      8.826e-03  1.821e-03   4.846 1.26e-06 ***
Scruz       -5.709e-03  6.256e-04  -9.126  < 2e-16 ***
Adjacent    -6.630e-04  2.933e-05 -22.608  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)
```

```
    Null deviance: 3510.73  on 29  degrees of freedom
Residual deviance:  716.85  on 24  degrees of freedom
AIC: 889.68

Number of Fisher Scoring iterations: 5
```

The values of coefficients and deviances are given in the output above.

**(b) Answer:**

$\eta = \log(\mu)$, $\frac{d\eta}{d\mu} = \frac{1}{\mu}$, $V(\mu) = \mu$, $w_i = \mu_i$

$z_i = \eta_i + \frac{y_i - \mu_i}{w_i} = \log(\mu_i) + \frac{y_i - \mu_i}{\mu_i}$

**(c) Answer:**

```
y <- gala$Species
mu <- y
eta <- log(mu)
w <- mu
z <- eta + (y-mu)/mu
lmod <- lm(z  ~ . -Species -Endemics, weights=w, gala)
coef(lmod)
```

```
  (Intercept)          Area     Elevation       Nearest         Scruz
 3.5191545412 -0.0005298484  0.0031643557  0.0025188990 -0.0037899780
     Adjacent
-0.0006623523
```

```
coef(mod)
```

```
  (Intercept)          Area     Elevation       Nearest         Scruz
 3.1548078779 -0.0005799429  0.0035405940  0.0088255719 -0.0057094223
     Adjacent
-0.0006630311
```

The coefficients are quite close.

**(d) Answer:**

```
y <- gala$Species
eta <- lmod$fit
mu <- exp(eta)

w <- mu

z <- eta + (y-mu)/mu

lmod <- lm(z  ~ . -Species -Endemics, weights=w, gala)
```

```
        2 * sum(y * log(y / mu) - (y - mu), na.rm = TRUE)
```

```
[1] 828.0096
```

```
        summary(mod)
```

```
Call:
glm(formula = Species ~ . - Endemics, family = poisson, data = gala)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.155e+00  5.175e-02  60.963  < 2e-16 ***
Area        -5.799e-04  2.627e-05 -22.074  < 2e-16 ***
Elevation    3.541e-03  8.741e-05  40.507  < 2e-16 ***
Nearest      8.826e-03  1.821e-03   4.846 1.26e-06 ***
Scruz       -5.709e-03  6.256e-04  -9.126  < 2e-16 ***
Adjacent    -6.630e-04  2.933e-05 -22.608  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3510.73  on 29  degrees of freedom
Residual deviance:  716.85  on 24  degrees of freedom
AIC: 889.68

Number of Fisher Scoring iterations: 5
```

The deviance after the first iteration is 828.0096 which is larger than 716.85 which is the deviance of the GLM.

**(e) Answer:**

```
        y <- gala$Species
        eta <- lmod$fit
        mu <- exp(eta)

        w <- mu

        z <- eta + (y - mu)/mu

        lmod <- lm(z ~ . -Species -Endemics, weights=w, gala)
```

```
        lmod$coef
```

```
  (Intercept)          Area     Elevation       Nearest         Scruz
3.1562582546 -0.0005793855  0.0035379237  0.0087861184 -0.0056868875
```

```
    Adjacent
-0.0006630167
```

```
        2 * sum(y * log(y / mu) - (y - mu), na.rm = TRUE)
```

```
[1] 719.4158
```

The deviance after this iteration is 719.4158 which is more close to the deviance of the GLM. The coefficient given in this iteration also gets closer to the coefficients of GLM.

**(f) Answer:**

```
        y <- gala$Species

        deviance = 2 * sum(y * log(y / mu) - (y - mu), na.rm = TRUE)

        for (iter in 1:10) {
          eta <- lmod$fit
          mu <- exp(eta)
          w <- mu
          z <- eta + (y - mu)/mu
          lmod <- lm(z ~ . -Species -Endemics, weights=w, gala)
          curr_deviance <- 2 * sum(y * log(y / mu) - (y - mu), na.rm = TRUE)
          print(curr_deviance)
          if (abs(deviance - curr_deviance) < 0.0001) {
            break
          }
          deviance = curr_deviance
        }
```

```
[1] 716.8488
[1] 716.8458
[1] 716.8458
```

```
        lmod$coef
```

```
  (Intercept)          Area     Elevation       Nearest         Scruz
 3.1548078779 -0.0005799429  0.0035405940  0.0088255719 -0.0057094223
    Adjacent
-0.0006630311
```

```
        mod$coef
```

```
  (Intercept)          Area     Elevation       Nearest         Scruz
 3.1548078779 -0.0005799429  0.0035405940  0.0088255719 -0.0057094223
    Adjacent
-0.0006630311
```

They are exactly the same.

**(g) Answer:**

```
        xm <- model.matrix(lmod)
        wm <- diag(w)

        #Standard error
        sqrt(diag(solve(t(xm) %*% wm %*% xm)))
```

```
 (Intercept)         Area    Elevation      Nearest        Scruz     Adjacent 
5.174955e-02 2.627299e-05 8.740709e-05 1.821261e-03 6.256214e-04 2.932754e-05 
```

```
        summary(mod)
```

```
Call:
glm(formula = Species ~ . - Endemics, family = poisson, data = gala)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept)  3.155e+00  5.175e-02  60.963  < 2e-16 ***
Area        -5.799e-04  2.627e-05 -22.074  < 2e-16 ***
Elevation    3.541e-03  8.741e-05  40.507  < 2e-16 ***
Nearest      8.826e-03  1.821e-03   4.846 1.26e-06 ***
Scruz       -5.709e-03  6.256e-04  -9.126  < 2e-16 ***
Adjacent    -6.630e-04  2.933e-05 -22.608  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3510.73  on 29  degrees of freedom
Residual deviance:  716.85  on 24  degrees of freedom
AIC: 889.68

Number of Fisher Scoring iterations: 5
```

They are exactly the same.

# Q6. ELMR Exercise 8.5 (p172)

**(a) Answer:**

```
        mod <- glm(Species ~ . -Endemics, data = gala, family = poisson)
```

```
        summary(mod)
```

```
Call:
glm(formula = Species ~ . - Endemics, family = poisson, data = gala)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.155e+00  5.175e-02  60.963  < 2e-16 ***
Area        -5.799e-04  2.627e-05 -22.074  < 2e-16 ***
Elevation    3.541e-03  8.741e-05  40.507  < 2e-16 ***
Nearest      8.826e-03  1.821e-03   4.846 1.26e-06 ***
Scruz       -5.709e-03  6.256e-04  -9.126  < 2e-16 ***
Adjacent    -6.630e-04  2.933e-05 -22.608  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3510.73  on 29  degrees of freedom
Residual deviance:  716.85  on 24  degrees of freedom
AIC: 889.68

Number of Fisher Scoring iterations: 5
```

p value of elevation is < 2e-16.

**(b) Answer:**

```
mod2 <- glm(Species ~ . -Endemics -Elevation, data = gala, family = poisson)
```

```
deviance_dff <- mod2$deviance - mod$deviance
```

```
pchisq(deviance_dff, 1, lower = FALSE)
```

```
[1] 0
```

p value is 0.

**(c) Answer:**

```
px <- sum(residuals(mod2, type = "pearson")^2)
```

```
pchisq(px, 1, lower = FALSE)
```

```
[1] 0
```

p value is 0.

**(d) Answer:**

```
(dp <- sum(residuals(mod, type="pearson")^2)/mod$df.res)
```

```
[1] 31.74914
```

```
summary(mod,dispersion=dp)
```

```
Call:
glm(formula = Species ~ . - Endemics, family = poisson, data = gala)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.1548079  0.2915897  10.819  < 2e-16 ***
Area        -0.0005799  0.0001480  -3.918 8.95e-05 ***
Elevation    0.0035406  0.0004925   7.189 6.53e-13 ***
Nearest      0.0088256  0.0102621   0.860    0.390
Scruz       -0.0057094  0.0035251  -1.620    0.105
Adjacent    -0.0006630  0.0001653  -4.012 6.01e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 31.74914)

    Null deviance: 3510.73  on 29  degrees of freedom
Residual deviance:  716.85  on 24  degrees of freedom
AIC: 889.68

Number of Fisher Scoring iterations: 5
```

p value is 6.53e-13

**(e) Answer:**

```
library(sandwich)

se <- mod |>
  vcovHC() |>
  diag() |>
  sqrt()
```

```
z <- mod$coef['Elevation'] / se['Elevation']
z
```

```
Elevation
 2.965378
```

```
2 * (1 - pnorm(abs(z)))
```

```
  Elevation
0.003023114
```

p value is 0.003023114

**(f) Answer:**

```
library(robust)
```

```
Loading required package: fit.models
```

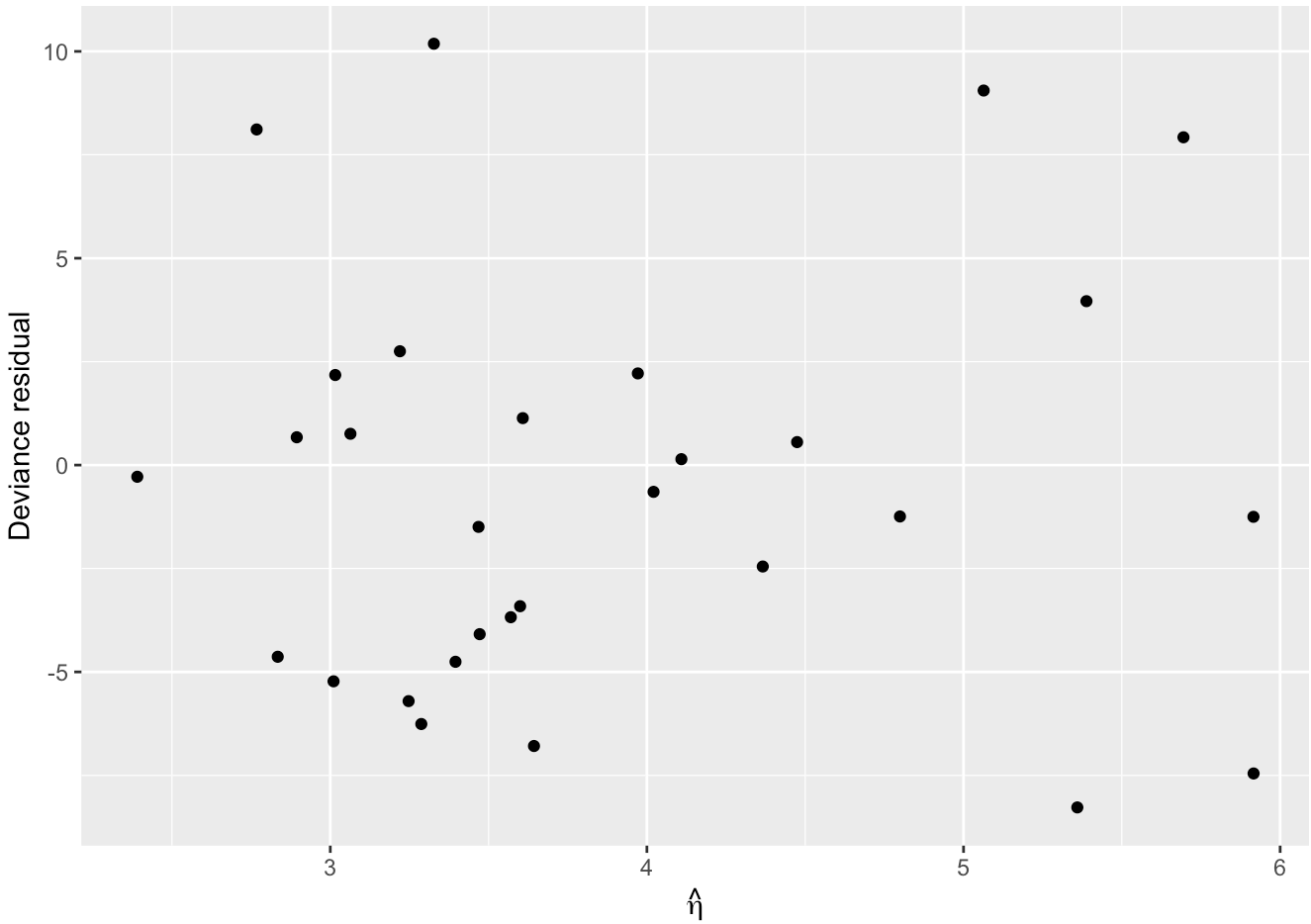
```
set.seed(300)

glmRob(Species ~ . -Endemics, data = gala, family = poisson) |>
  summary()
```

```
Call: glmRob(formula = Species ~ . - Endemics, family = poisson, data = gala)
Deviance Residuals:
     Min       1Q   Median       3Q      Max
  0.7319  65.2010  87.6696 144.7517 191.1331

Coefficients:
            Estimate Std. Error    z value Pr(>|z|)
(Intercept)  -103363   0.429314 -2.408e+05        0
Area         -142913   0.001843 -7.754e+07        0
Elevation       1650   4.607764  3.581e+02        0
Nearest        11234   0.608225  1.847e+04        0
Scruz            569   8.459854  6.726e+01        0
Adjacent        3454   0.018431  1.874e+05        0

(Dispersion Parameter for poisson family taken to be 1 )

    Null Deviance: 21190 on 29 degrees of freedom

Residual Deviance: NaN on 24 degrees of freedom

Number of Iterations: 50

Correlation of Coefficients:
          (Intercept) Area   Elevation Nearest Scruz
Area      2.3293
Elevation 0.4293      1.0000
Nearest   0.4293      1.0000 1.0000
Scruz     0.4293      1.0000 1.0000    1.0000
Adjacent  0.4293      1.0000 1.0000    1.0000  1.0000
```

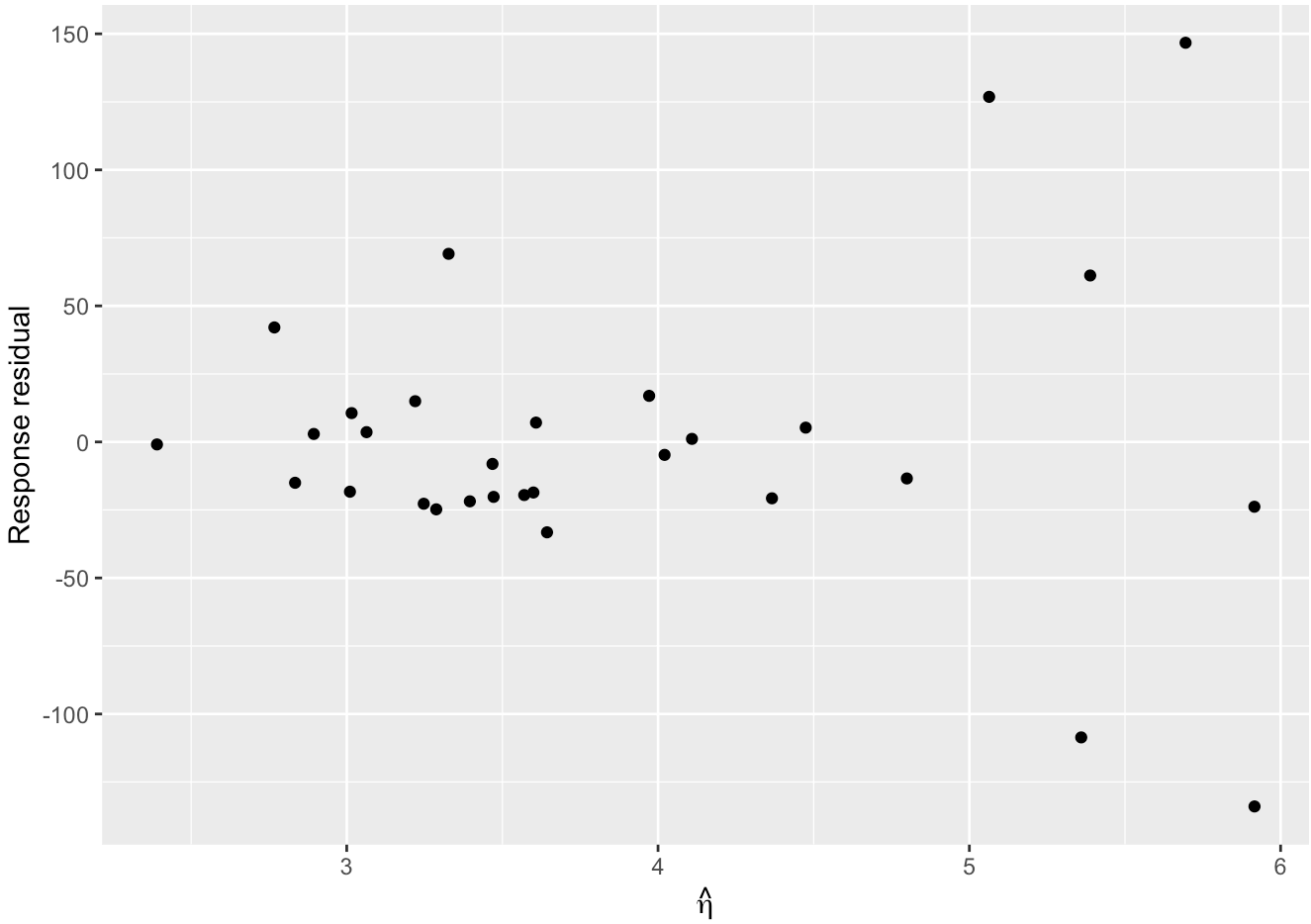
p value is 0

**(g) Answer:**

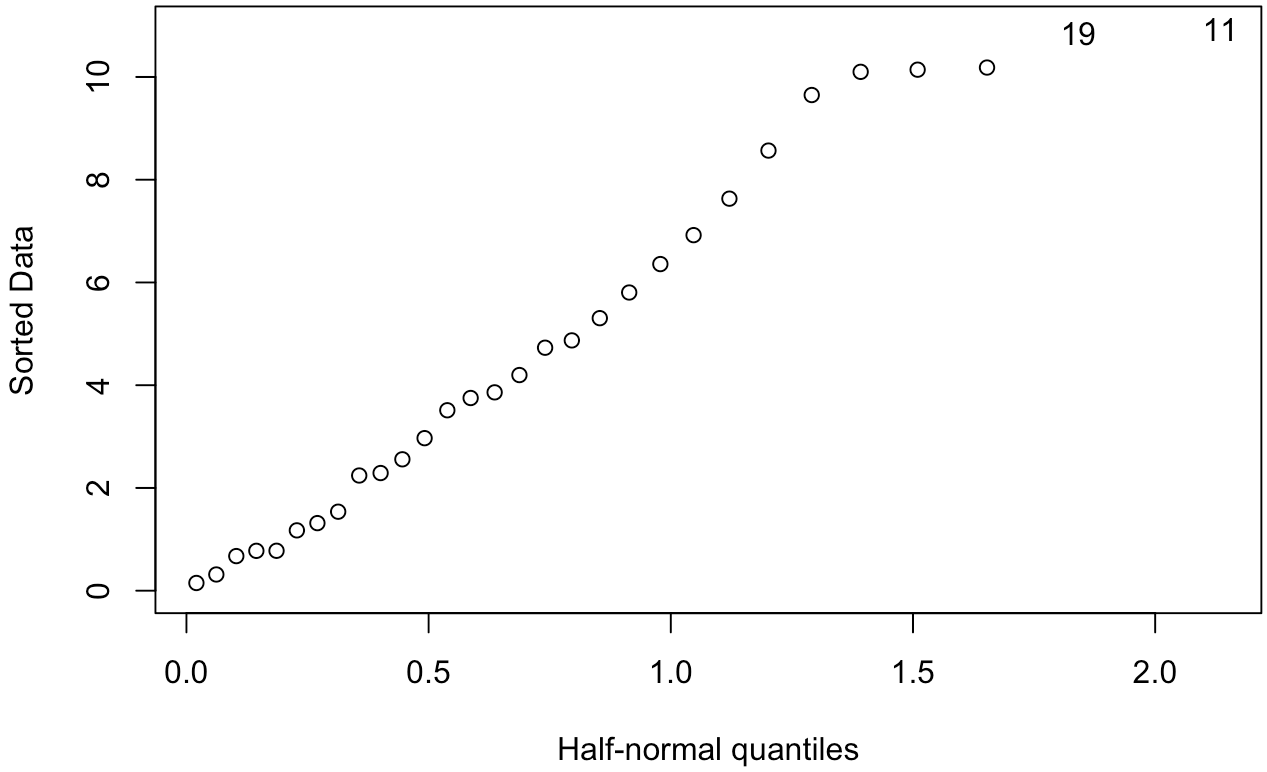
```
gala |>
  dplyr::mutate(devres  = residuals(mod, type = "deviance"),
         linpred = predict(mod, type = "link")) |>
  ggplot() +
  geom_point(mapping = aes(x = linpred, y = devres)) +
  labs(x = expression(hat(eta)), y = "Deviance residual")
```



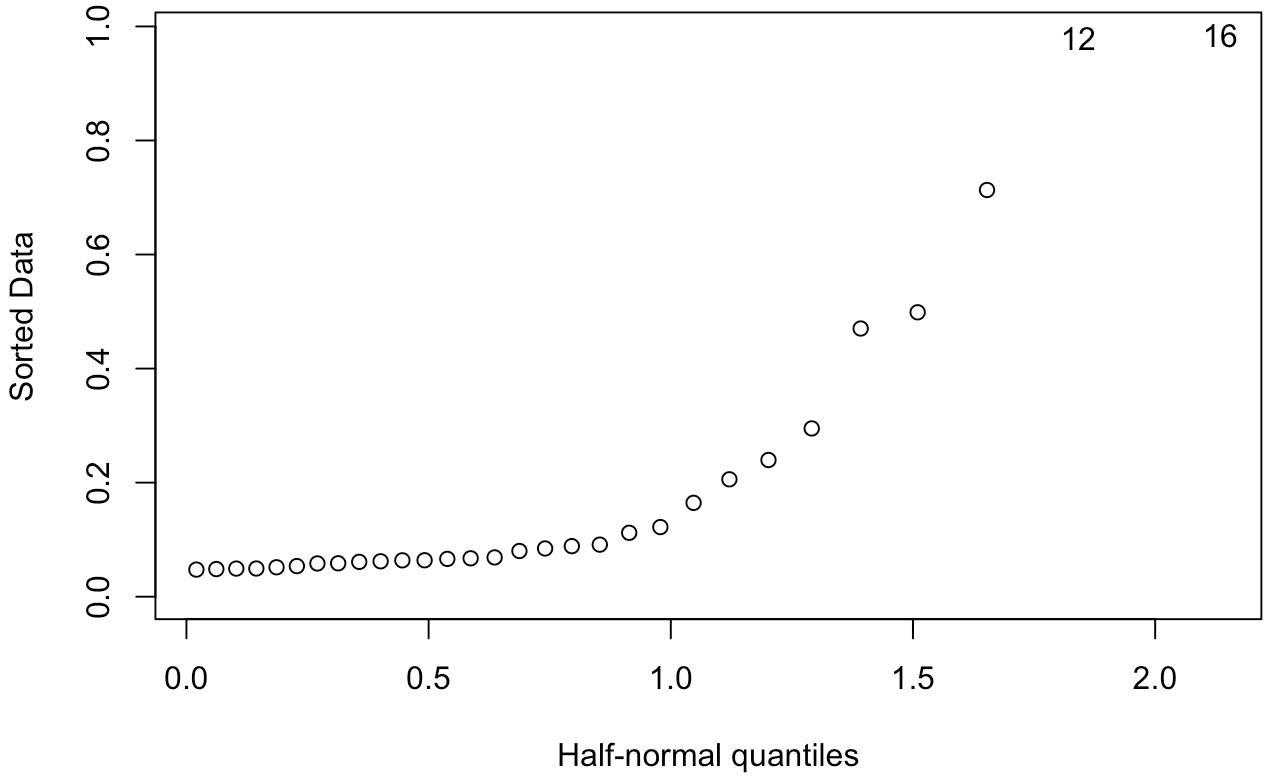

```
gala |>
  dplyr::mutate(resres  = residuals(mod, type = "response"),
         linpred = predict(mod, type = "link")) |>
  ggplot() +
  geom_point(mapping = aes(x = linpred, y = resres)) +
  labs(x = expression(hat(eta)), y = "Response residual")
```




```
halfnorm(rstudent(mod))
```




```
gali <- influence(mod)
halfnorm(gali$hat)
```




All six results show `elevation` is a significant predictor. The p values are 0, 0, 0, 6.53e-13, 0.003023114, 0. Although all of them have the same inference, we should use robust estimation since there is overdispersion and outliers.