MA 576 HW 6

Benjamin Draves

```r
#load necessary packages
library(plyr)
library(tidyr)
library(ggplot2)
```

Exercise 3

Part a

```r
#Read in data
trucks = read.table("~/Desktop/Courses/MA 576/data/trucks.dat", header = T)
head(trucks)
```

```
##   B C D E O height
## 1 - - - - -   7.78
## 2 + - - + -   8.15
## 3 - + - + -   7.50
## 4 + + - - -   7.59
## 5 - - + + -   7.94
## 6 + - + - -   7.69
```

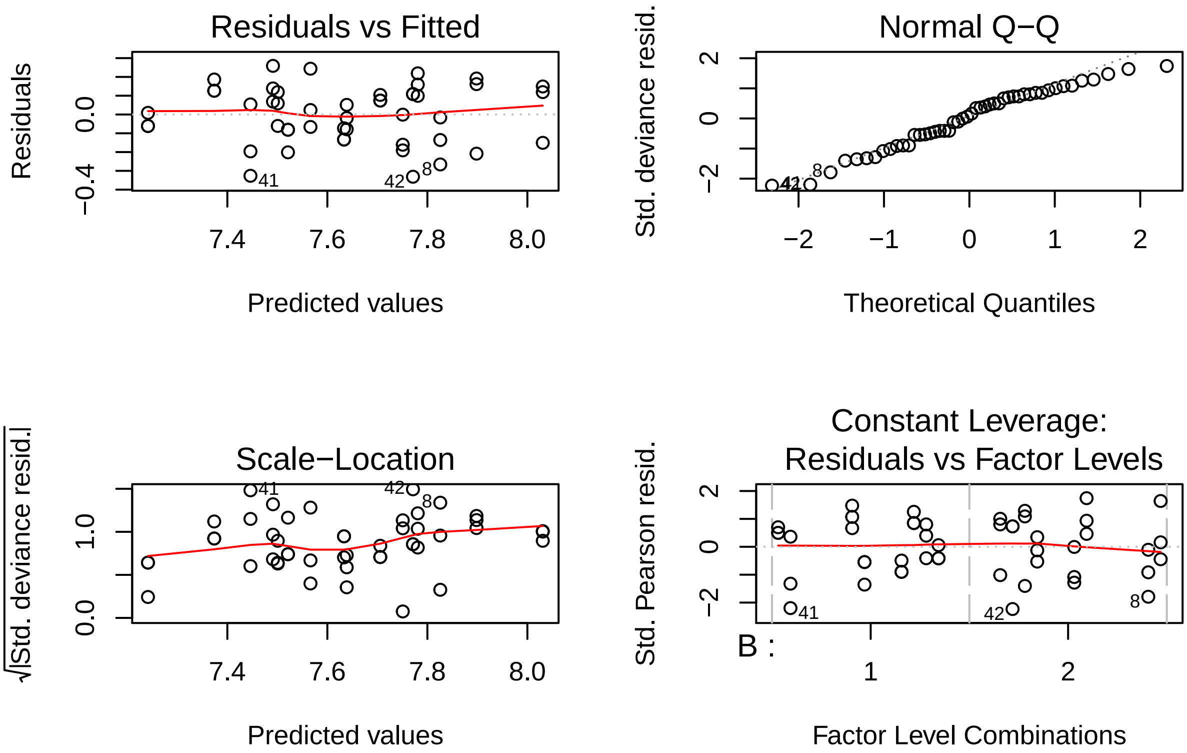
```r
#rename levels
levels(trucks[,1])[levels(trucks[,1])=="-"] = "1"
levels(trucks[,2])[levels(trucks[,2])=="-"] = "1"
levels(trucks[,3])[levels(trucks[,3])=="-"] = "1"
levels(trucks[,4])[levels(trucks[,4])=="-"] = "1"
levels(trucks[,5])[levels(trucks[,5])=="-"] = "1"

levels(trucks[,1])[levels(trucks[,1])=="+"] = "2"
levels(trucks[,2])[levels(trucks[,2])=="+"] = "2"
levels(trucks[,3])[levels(trucks[,3])=="+"] = "2"
levels(trucks[,4])[levels(trucks[,4])=="+"] = "2"
levels(trucks[,5])[levels(trucks[,5])=="+"] = "2"

#fit linear regression
m1 = glm(height~., data = trucks, family = gaussian)
summary(m1)
```

```
##
## Call:
## glm(formula = height ~ ., family = gaussian, data = trucks)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.33125  -0.09427   0.01625   0.11917   0.25875
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.70583    0.05612 137.319  < 2e-16 ***
## B2           0.22125    0.04582   4.829 1.85e-05 ***
## C2          -0.17625    0.04582  -3.847   0.0004 ***
## D2          -0.02875    0.04582  -0.627   0.5337
## E2           0.10375    0.04582   2.264   0.0288 *
## O2          -0.25958    0.04582  -5.665 1.20e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.02519216)
##
##     Null deviance: 2.9659  on 47  degrees of freedom
## Residual deviance: 1.0581  on 42  degrees of freedom
## AIC: -32.89
##
## Number of Fisher Scoring iterations: 2
```

```r
#find combination of factors that is closest to 8 inches
possible = expand.grid(rep(list(1:2), 5))
possible = as.data.frame(lapply(possible,factor))
colnames(possible) = colnames(trucks[,-6])
pred = as.vector(predict(m1, newdata = possible))
opt = which.min(abs(8-pred))
possible[opt,]
```

```
##    B C D E O
## 14 2 1 2 2 1
```

```r
#plot residuals
par(mfrow = c(2,2))
plot(m1)
```



It appears that the furnance temperature (B+), heating time (C+), and quench oil temperature (O+) all significantly affect height. Moreover, hold-down time (E+) is significantly affects the height at the $p = 0.05$ level. The only covariate that does not play a signficant role is transfer time. From this model fit, we expect that the vector $(B, C, D, E, O) = (+, -, +, +, -)$ makes the leaf spring free height as close to 8 inches as possible (8.002083 inches). It appears that this model is quite underdispersed. While the residuals show no clear behavior and are relatively normally the variance appears to be smaller than expected.

Part b

Now suppose that the model variance is a function of the covariates. Then we may wish to fit the model $\sigma^2 = \sum_{j=1}^{p} \gamma_j X_j$ and use this model estimate in a weighted least square regression. Here, as $\sigma^2 > 0$ we will fit a Gamma GLM to the data with a log-link to arrive at estimates of $\hat{\sigma^2}$ as a function of the covariates.

```r
#get residuals
res = residuals(m1, type = "response")^2

#fit a Gamma GLM
df2 = cbind(trucks[,-6], res)
m2 = glm(res~., df2, family = Gamma(link = "log"))
summary(m2)
```

```
##
## Call:
## glm(formula = res ~ ., family = Gamma(link = "log"), data = df2)
##
```

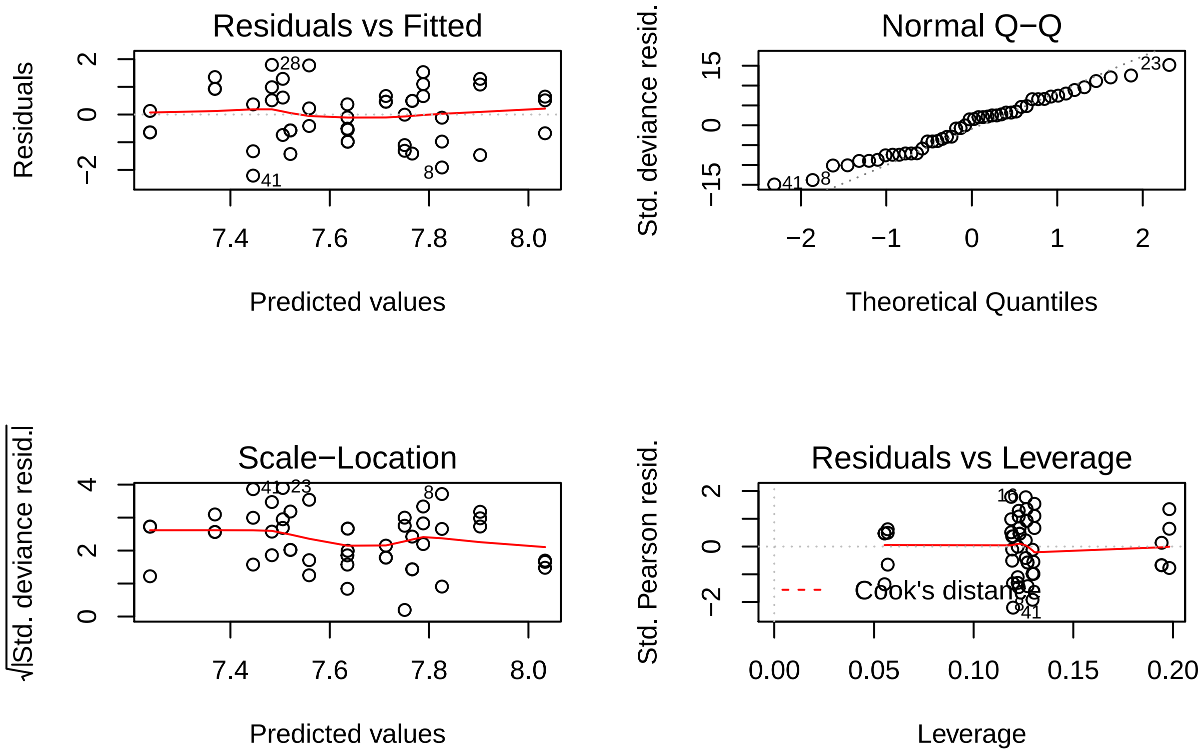
```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3190  -1.0957  -0.6343   0.5974   2.1450
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.86502    0.39934  -9.678 2.95e-12 ***
## B2           0.49818    0.32606   1.528    0.134
## C2          -0.48640    0.32606  -1.492    0.143
## D2          -0.49281    0.32606  -1.511    0.138
## E2           0.39977    0.32606   1.226    0.227
## O2           0.03824    0.32606   0.117    0.907
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.27579)
##
##     Null deviance: 89.281  on 47  degrees of freedom
## Residual deviance: 82.390  on 42  degrees of freedom
## AIC: -266.49
##
## Number of Fisher Scoring iterations: 12
```

We note that while this model suggests that the covariates are relatively unrelated, we now have estimates for the model variance at each given datapoint. With this we can complete a weighted regression based on these estimates as follows.

```
#fit joint model
sigma_weights = 1/m2$fitted.values
m3 = glm(height~., data = trucks, family = gaussian, weights = sigma_weights)

#summary statistics
summary(m3)
```

```
##
## Call:
## glm(formula = height ~ ., family = gaussian, data = trucks, weights = sigma_weights)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2076  -0.6919   0.1754   0.6660   1.7948
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.71307    0.05426 142.150  < 2e-16 ***
## B2           0.21750    0.04332   5.021 9.93e-06 ***
## C2          -0.17973    0.04326  -4.155 0.000156 ***
## D2          -0.02776    0.04329  -0.641 0.524803
## E2           0.10282    0.04290   2.397 0.021055 *
## O2          -0.26727    0.04176  -6.400 1.06e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.14154)
##
##     Null deviance: 156.345  on 47  degrees of freedom
## Residual deviance:  47.945  on 42  degrees of freedom
## AIC: -36.391
##
## Number of Fisher Scoring iterations: 2
```

```
par(mfrow = c(2,2))
plot(m3)
```



Under this new joint model, we see that the model estimates are altered, but only slightly. Moreover, the standard errors seem to go unchanged. We notice however, as the weight matrix change (from I to $W = diag(1/\hat{\sigma^2})$) that our residual deviance is much closer to what we expect. That is, the model does not appear underdispersed any longer.

Exercise 4

Part a

```
#Read in data
gala = read.table("~/Desktop/Courses/MA 576/data/galapagos.txt", header = T)
head(gala)
```

```
##         Island Plants PlantEnd Finches FinchEnd FinchGenera  Area
## 1       Baltra     58       23      NA       NA           4 25.09
## 2    Bartolome     31       21      NA       NA          NA  1.24
## 3     Caldwell      3        3      NA       NA          NA  0.21
## 4     Champion     25        9      NA       NA          NA  0.10
## 5      Coamano      2        1      NA       NA          NA  0.05
## 6 Daphne.Major     18       11      NA       NA          NA  0.34
##   Elevation Nearest StCruz Adjacent
## 1       100     0.6    0.6     1.84
## 2       109     0.6   26.3   572.33
## 3       114     2.8   58.7     0.78
## 4        46     1.9   47.4     0.18
## 5        25     1.9    1.9   903.82
## 6        50     8.0    8.0     1.84
```

```
#fit binomial GLM
m4 = glm(cbind(PlantEnd,Plants-PlantEnd)~ Elevation + Adjacent, data  = gala, family = binomial

#get covariance
S = summary(m4)$cov.unscaled
S
```

```
##               (Intercept)     Elevation      Adjacent
## (Intercept)  6.530774e-03 -6.795758e-06  9.070762e-07
## Elevation   -6.795758e-06  1.037065e-08 -2.362671e-09
## Adjacent     9.070762e-07 -2.362671e-09  2.637772e-09
```

```
sigma = 1

#get point estimate
nd = data.frame(Elevation = 100, Adjacent = 100)
linear_pred = predict(m4, nd)
data_pred = exp(linear_pred)/(1+exp(linear_pred))
```

Notice to get a standard error on our prediction we have

$$Var(y*) = Var(\hat{\beta}_0 + 100\hat{\beta}_1 + 100\hat{\beta}_2)$$

which we can compute via the covariance matrix Σ.

```
#get standard error
var = S[1,1] + 100^2*S[2,2] + 100^2*S[3,3] + 2*100*S[1,2] + 2*100*S[1,3] + 2*100*100*S[2,3]
se = sigma*sqrt(var)

#get linear space PI
linear_left = linear_pred - qnorm(.975)*se
linear_right = linear_pred + qnorm(.975)*se

#get data space PI
data_left = exp(linear_left)/(1+exp(linear_left))
```

```
data_right = exp(linear_right)/(1+exp(linear_right))
c(data_left, data_right)
```

```
##         1         1 
## 0.3613681 0.4303505
```

Part b

```
#fit quasi-binomial GLM
m5 = glm(cbind(PlantEnd,Plants-PlantEnd)~ Elevation + Adjacent, data  = gala, family = quasibi

#get scaled covariance
S = summary(m5)$cov.unscaled
S
```

```
##               (Intercept)     Elevation      Adjacent
## (Intercept)  6.530774e-03 -6.795758e-06  9.070762e-07
## Elevation   -6.795758e-06  1.037065e-08 -2.362671e-09
## Adjacent     9.070762e-07 -2.362671e-09  2.637772e-09
```

```
sigma2 = sum(resid(m5, type = "pearson")^2)/(m5$df.residual)

#get standard error
var = S[1,1] + 100^2*S[2,2] + 100^2*S[3,3] + 200*S[1,2] + 200*S[1,3] + 2*100*100*S[2,3]
se = sqrt(sigma2)*sqrt(var)

#get linear space PI
linear_left = linear_pred - qnorm(.975)*se
linear_right = linear_pred + qnorm(.975)*se

#get data space PI
data_left = exp(linear_left)/(1+exp(linear_left))
data_right = exp(linear_right)/(1+exp(linear_right))
c(data_left, data_right)
```

```
##         1         1 
## 0.3333835 0.4608475
```

Part c

```
#Sandwich estimator for quasi-binomial
Sinv = summary(m5)$cov.unscaled
A = crossprod(model.matrix(m5), model.matrix(m5) * m5$weights * resid(m5, type="pearson")^2)
S = Sinv %*% A %*% Sinv
S
```

```
##               (Intercept)     Elevation      Adjacent
## (Intercept)  2.042720e-02 -2.296332e-05  2.492196e-06
## Elevation   -2.296332e-05  4.549238e-08 -9.763027e-09
## Adjacent     2.492196e-06 -9.763027e-09  4.511121e-09
```

```
sigma2 = sum(resid(m5, type = "pearson")^2)/(m5$df.residual)


#get standard error
var = S[1,1] + 100^2*S[2,2] + 100^2*S[3,3] + 200*S[1,2] + 200*S[1,3] + 2*100*100*S[2,3]
se = sqrt(sigma2)*sqrt(var)

#get linear space PI
linear_left = linear_pred - qnorm(.975)*se
linear_right = linear_pred + qnorm(.975)*se

#get data space PI
data_left = exp(linear_left)/(1+exp(linear_left))
data_right = exp(linear_right)/(1+exp(linear_right))
c(data_left, data_right)
```

```
##         1         1
## 0.2903298 0.5109808
```

Here we notice that by using the sandwich estimator for the covariance that the estimates on teh variance remain relataively similar (except the intercept term). The covariance components, however, increase considerably. As a result we see that when we estimate this variance that our prediction interval increases sizeably. The prediction interval for the binomial-glm is the smallest. The prediction interval for the quasi-binomial is larger as we estimate the dispersion to be $\hat{\sigma^2} = 3.439451$. Lastly, we see the sandwich estimator has the largest interval as it detects covariance amoungst the parameters of the model.