StatR 201 HW 4 Key

Scott Rinnnan and Gregor Thomas

Thursday, Feb 5, 2015

Contents

Problem 1: Offsets and complaints 1
Problem 2: Abalone 8
G&H #6: 12

```
library(faraway)
library(arm)
library(ggplot2)
library(magrittr)
library(hett)
library(metRology)
setwd("~/Dropbox/STATR 201/Week 4/")
```

Problem 1: Offsets and complaints

(a) Let's look at the ratio of complaints per visit.

```
data(esdcomp)

with(esdcomp,mean(complaints/visits))
```

```
## [1] 0.001328877
```

```
with(esdcomp,max(complaints/visits))
```

```
## [1] 0.003017005
```

Indeed, quite low. Even the maximum is only a little more than twice as much as the mean.

(b)

Fitting the model:

```
mod1<-glm(complaints~residency+gender+revenue+hours+offset(log(visits)),
          data=esdcomp,family=poisson)
summary(mod1)
```

```
##
## Call:
## glm(formula = complaints ~ residency + gender + revenue + hours +
##     offset(log(visits)), family = poisson, data = esdcomp)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9434  -0.9490  -0.3130   0.7859   1.8036
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.1202460  0.8502806  -9.550   <2e-16 ***
## residencyY  -0.2090058  0.2011520  -1.039   0.2988
## genderM      0.1954338  0.2181525   0.896   0.3703
## revenue      0.0015761  0.0028294   0.557   0.5775
## hours        0.0007019  0.0003505   2.002   0.0452 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 63.435  on 43  degrees of freedom
## Residual deviance: 54.518  on 39  degrees of freedom
## AIC: 187.3
##
## Number of Fisher Scoring iterations: 5
```

(c)

```r
drop1(mod1)
```

```
## Single term deletions
##
## Model:
## complaints ~ residency + gender + revenue + hours + offset(log(visits))
##           Df Deviance    AIC
## <none>         54.518 187.30
## residency  1   55.610 186.39
## gender     1   55.341 186.12
## revenue    1   54.827 185.61
## hours      1   58.698 189.48
```

It looks like a marginal decrease in AIC can be accomplished by excluding the revenue variable. Probably not justifiable, though, given a decrease in AIC of less than 2.

```r
add1(mod1,scope=~(residency+gender+revenue+hours)^2)
```

```
## Single term additions
##
## Model:
## complaints ~ residency + gender + revenue + hours + offset(log(visits))
##                   Df Deviance    AIC
## <none>                 54.518 187.30
## residency:gender   1   53.975 188.75
## residency:revenue  1   54.016 188.79
## residency:hours    1   48.369 183.15
## gender:revenue     1   54.413 189.19
## gender:hours       1   52.240 187.02
## revenue:hours      1   53.393 188.17
```

It would appear that adding the interaction term `residency:hours` gives us a better model. Let's add it and try again:

```
mod2<-glm(complaints~residency+gender+revenue+hours+offset(log(visits))+residency:hours,
          data=esdcomp,family=poisson)
add1(mod2,scope=~(residency+gender+revenue+hours)^2)
```

```
## Single term additions
##
## Model:
## complaints ~ residency + gender + revenue + hours + offset(log(visits)) +
##     residency:hours
##                    Df Deviance    AIC
## <none>                 48.369 183.15
## residency:gender   1   48.187 184.97
## residency:revenue  1   48.028 184.81
## gender:revenue     1   48.174 184.95
## gender:hours       1   47.164 183.94
## revenue:hours      1   43.853 180.63
```

Again, it looks like adding `revenue:hours` gives us a slightly better model fit. The final model:

```
mod3<-glm(complaints~residency+gender+revenue+hours+offset(log(visits))+
            residency:hours+revenue:hours,
          data=esdcomp,family=poisson)
summary(mod3)
```

```
##
## Call:
## glm(formula = complaints ~ residency + gender + revenue + hours +
##     offset(log(visits)) + residency:hours + revenue:hours, family = poisson,
##     data = esdcomp)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7099  -0.9546  -0.1760   0.6442   2.1444
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.138e+00  3.809e+00  -0.561  0.57452
## residencyY        3.495e+00  1.238e+00   2.823  0.00477 **
## genderM           2.073e-01  2.220e-01   0.934  0.35028
## revenue          -3.158e-02  1.517e-02  -2.081  0.03743 *
## hours            -3.343e-03  2.447e-03  -1.366  0.17196
## residencyY:hours -2.409e-03  7.922e-04  -3.040  0.00236 **
## revenue:hours     2.186e-05  9.869e-06   2.215  0.02677 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 63.435  on 43  degrees of freedom
## Residual deviance: 43.853  on 37  degrees of freedom
```

```
## AIC: 180.63
##
## Number of Fisher Scoring iterations: 5
```

```r
AIC(mod1,mod3)
```

```
##      df      AIC
## mod1  5 187.2973
## mod3  7 180.6326
```

Finally, a quick note for future work: it is important to understand how we are determining which variables to include or exclude from the model, but iterating the `add1` and `drop1` functions can be tedious and time-consuming. This whole process can be accomplished in one go by using the `stepAIC` function, which can fit the best model by adding and subtracting components:

```r
stepAIC(mod1,scope=~(residency+gender+revenue+hours)^2)
```

```
## Start:  AIC=187.3
## complaints ~ residency + gender + revenue + hours + offset(log(visits))
##
##                     Df Deviance    AIC
## + residency:hours    1   48.369 183.15
## - revenue            1   54.827 185.61
## - gender             1   55.341 186.12
## - residency          1   55.610 186.39
## + gender:hours       1   52.240 187.02
## <none>                   54.518 187.30
## + revenue:hours      1   53.393 188.17
## + residency:gender   1   53.975 188.75
## + residency:revenue  1   54.016 188.79
## + gender:revenue     1   54.413 189.19
## - hours              1   58.698 189.48
##
## Step:  AIC=183.15
## complaints ~ residency + gender + revenue + hours + residency:hours +
##     offset(log(visits))
##
##                     Df Deviance    AIC
## + revenue:hours      1   43.853 180.63
## - revenue            1   48.761 181.54
## - gender             1   50.182 182.96
## <none>                   48.369 183.15
## + gender:hours       1   47.164 183.94
## + residency:revenue  1   48.028 184.81
## + gender:revenue     1   48.174 184.95
## + residency:gender   1   48.187 184.97
## - residency:hours    1   54.518 187.30
##
## Step:  AIC=180.63
## complaints ~ residency + gender + revenue + hours + residency:hours +
##     revenue:hours + offset(log(visits))
##
```

```
##                     Df Deviance    AIC
## - gender             1   44.747 179.53
## <none>                   43.853 180.63
## + gender:hours       1   42.973 181.75
## + residency:revenue  1   43.639 182.42
## + residency:gender   1   43.644 182.42
## + gender:revenue     1   43.734 182.51
## - revenue:hours      1   48.369 183.15
## - residency:hours    1   53.393 188.17
##
## Step:  AIC=179.53
## complaints ~ residency + revenue + hours + residency:hours +
##     revenue:hours + offset(log(visits))
##
##                     Df Deviance    AIC
## <none>                   44.747 179.53
## + gender             1   43.853 180.63
## + residency:revenue  1   44.405 181.19
## - revenue:hours      1   50.182 182.96
## - residency:hours    1   53.789 186.57
```

```
##
## Call:  glm(formula = complaints ~ residency + revenue + hours + residency:hours +
##     revenue:hours + offset(log(visits)), family = poisson, data = esdcomp)
##
## Coefficients:
##      (Intercept)        residencyY           revenue             hours
##       -1.279e+00         3.342e+00        -3.374e-02        -3.930e-03
## residencyY:hours     revenue:hours
##       -2.339e-03         2.381e-05
##
## Degrees of Freedom: 43 Total (i.e. Null);  38 Residual
## Null Deviance:       63.44
## Residual Deviance: 44.75     AIC: 179.5
```

Keep in mind that this ONLY judges which model is best based on AIC. Other methods should also be considered. Fitting models is an art, not an algorithm!

(d)

Quasi-Poisson model:

```r
mod4<-glm(complaints~residency+gender+revenue+hours+offset(log(visits))+
            residency:hours+revenue:hours,
          data=esdcomp,family=quasipoisson)
summary(mod4)
```

```
##
## Call:
## glm(formula = complaints ~ residency + gender + revenue + hours +
##     offset(log(visits)) + residency:hours + revenue:hours, family = quasipoisson,
##     data = esdcomp)
##
## Deviance Residuals:
```

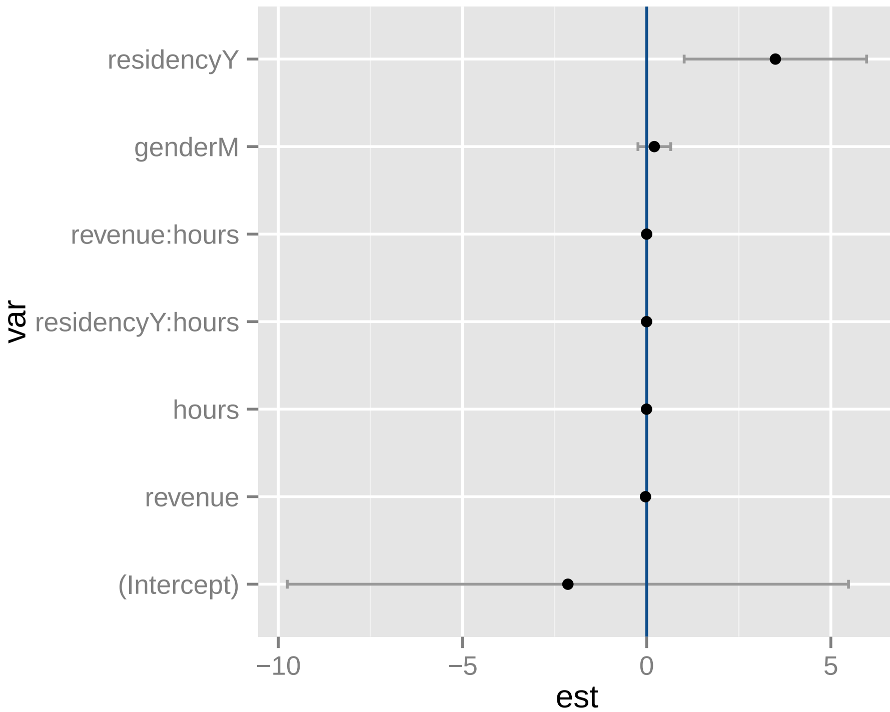
```
##     Min       1Q   Median       3Q      Max
## -1.7099  -0.9546  -0.1760   0.6442   2.1444
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -2.138e+00  4.175e+00  -0.512  0.61160
## residencyY        3.495e+00  1.357e+00   2.575  0.01417 *
## genderM           2.073e-01  2.433e-01   0.852  0.39966
## revenue          -3.158e-02  1.663e-02  -1.898  0.06546 .
## hours            -3.343e-03  2.683e-03  -1.246  0.22058
## residencyY:hours -2.409e-03  8.684e-04  -2.773  0.00864 **
## revenue:hours     2.186e-05  1.082e-05   2.020  0.05062 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.201698)
##
##     Null deviance: 63.435  on 43  degrees of freedom
## Residual deviance: 43.853  on 37  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

The dispersion parameter is only 1.2. Probably not different enough from 1 to justify adding an extra parameter to our model.

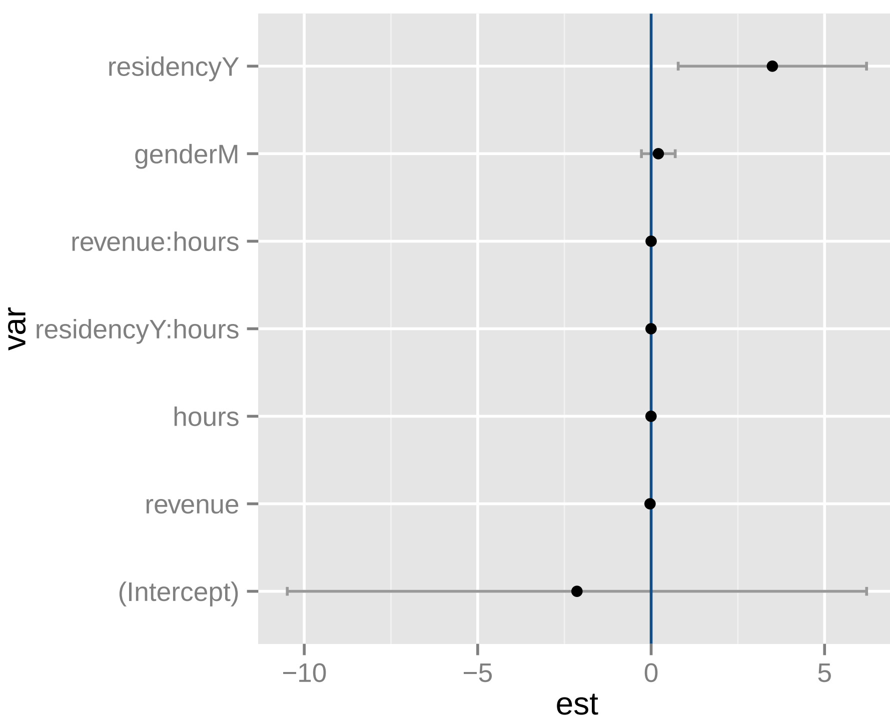
(e)

Ladder plots:

```
poiscoef<-data.frame(var = names(coef(mod3)), est = coef(mod3), se = se.coef(mod3)) #arm::se.coef
poiscoef$var %<>% reorder(poiscoef$est)
p <- ggplot(poiscoef, aes(x = est, y = var))
p + geom_errorbarh(aes(xmin = est - 2 * se, xmax = est + 2 * se), height = 0.1, color = "gray60") +
  geom_vline(xintercept = 0, color = "dodgerblue4") +
  geom_point()
```




```
qpoiscoef<-data.frame(var = names(coef(mod4)), est = coef(mod4), se = se.coef(mod4))
qpoiscoef$var %<>% reorder(qpoiscoef$est)
p <- ggplot(qpoiscoef, aes(x = est, y = var))
p + geom_errorbarh(aes(xmin = est - 2 * se, xmax = est + 2 * se), height = 0.1, color = "gray60") +
  geom_vline(xintercept = 0, color = "dodgerblue4") +
  geom_point()
```



The relatively small differences in the ladder plots confirm there is little difference between fitting a Poisson model and a quasi-Poisson model.

Problem 2: Abalone

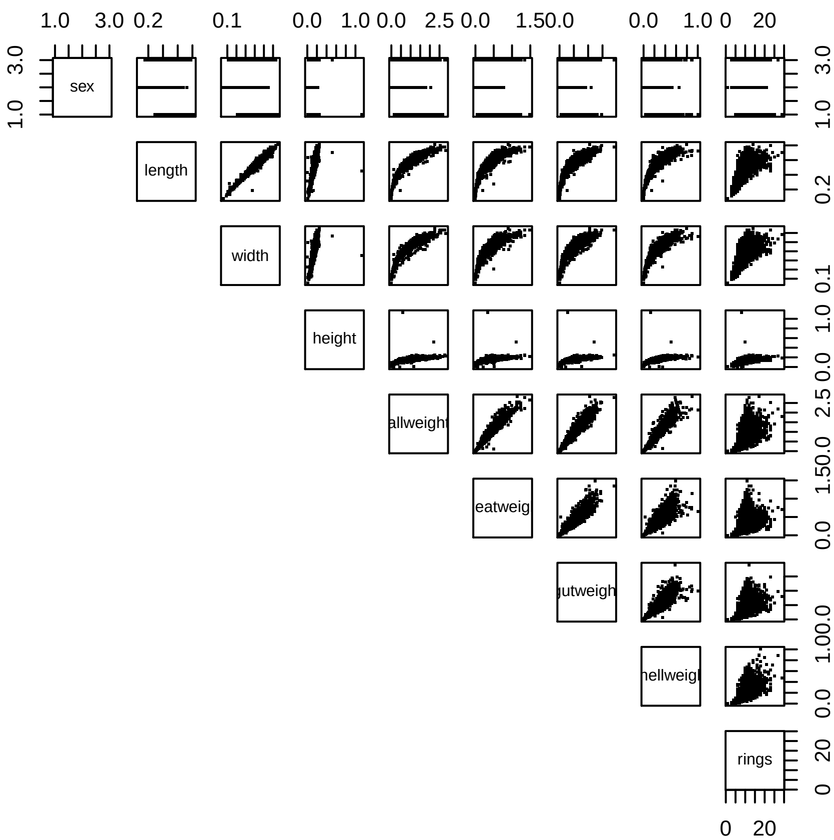
(a)

Read in the data, and do some initial exploration:

```r
abalone<-read.csv("abaloneTrain.csv")
head(abalone)
```

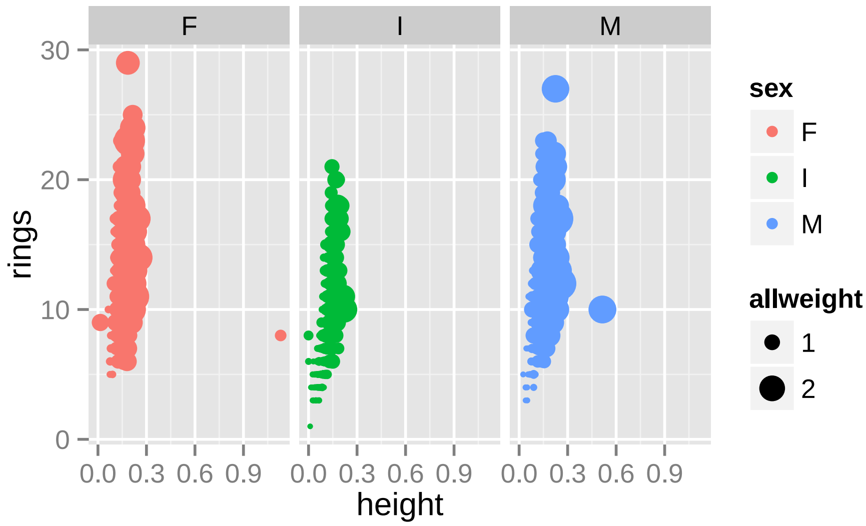
```
##   sex length width height allweight meatweight gutweight shellweight rings
## 1   I  0.430 0.320  0.110    0.3675     0.1675    0.1020       0.105     8
## 2   M  0.485 0.395  0.140    0.6295     0.2285    0.1270       0.225    14
## 3   I  0.655 0.515  0.145    1.2500     0.5265    0.2830       0.315    15
## 4   F  0.575 0.470  0.165    0.8690     0.4350    0.1970       0.238     9
## 5   I  0.500 0.375  0.145    0.5795     0.2390    0.1375       0.185     9
## 6   F  0.595 0.470  0.155    1.1775     0.5420    0.2690       0.310     9
```

```r
pairs(abalone,pch=".",lower.panel = NULL)
```



We can see right away that several of our variables our heavily correlated. We can use this information to inform our model selection. It probably won't make a lot of sense, for example, to include both length and width in a model, since the second variable doesn't offer much new information, given the first.

```r
ggplot(abalone,aes(x=height,y=rings,colour=sex,size=allweight))+
  geom_point() +
  facet_grid(~sex)
```



We can also see there are a couple of data points that we will probably want to consider outliers.

(b)

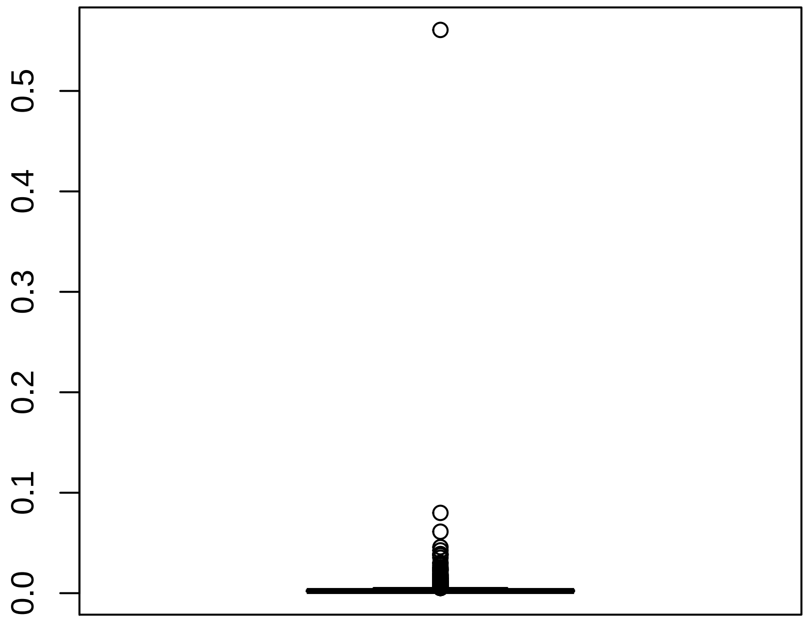
The full linear model:

```r
mod1<-lm(rings~.,data=abalone)
summary(mod1)
```

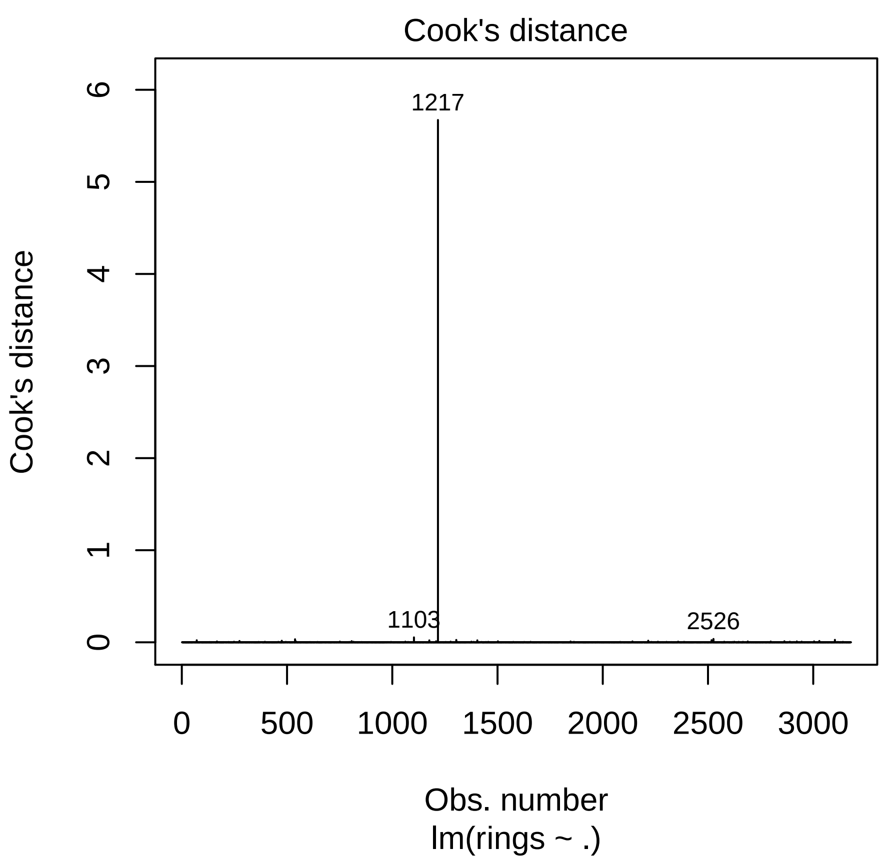
```
##
## Call:
## lm(formula = rings ~ ., data = abalone)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7191 -1.3108 -0.3275  0.8779 13.9763
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.95409    0.33578  11.776  < 2e-16 ***
## sexI         -0.69996    0.11864  -5.900 4.03e-09 ***
## sexM          0.17671    0.09623   1.836 0.066400 .
## length        0.15763    2.07183   0.076 0.939359
## width         9.75763    2.54985   3.827 0.000132 ***
## height       10.11446    1.62689   6.217 5.73e-10 ***
## allweight     9.61250    0.81764  11.756  < 2e-16 ***
## meatweight  -20.16914    0.92692 -21.759  < 2e-16 ***
## gutweight   -10.90474    1.48704  -7.333 2.84e-13 ***
## shellweight   7.87468    1.26626   6.219 5.66e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.2 on 3167 degrees of freedom
```

```
## Multiple R-squared:  0.5277, Adjusted R-squared:  0.5264
## F-statistic: 393.2 on 9 and 3167 DF,  p-value: < 2.2e-16
```

```r
boxplot(hatvalues(mod1))
```




```r
plot(mod1,which=4)
```

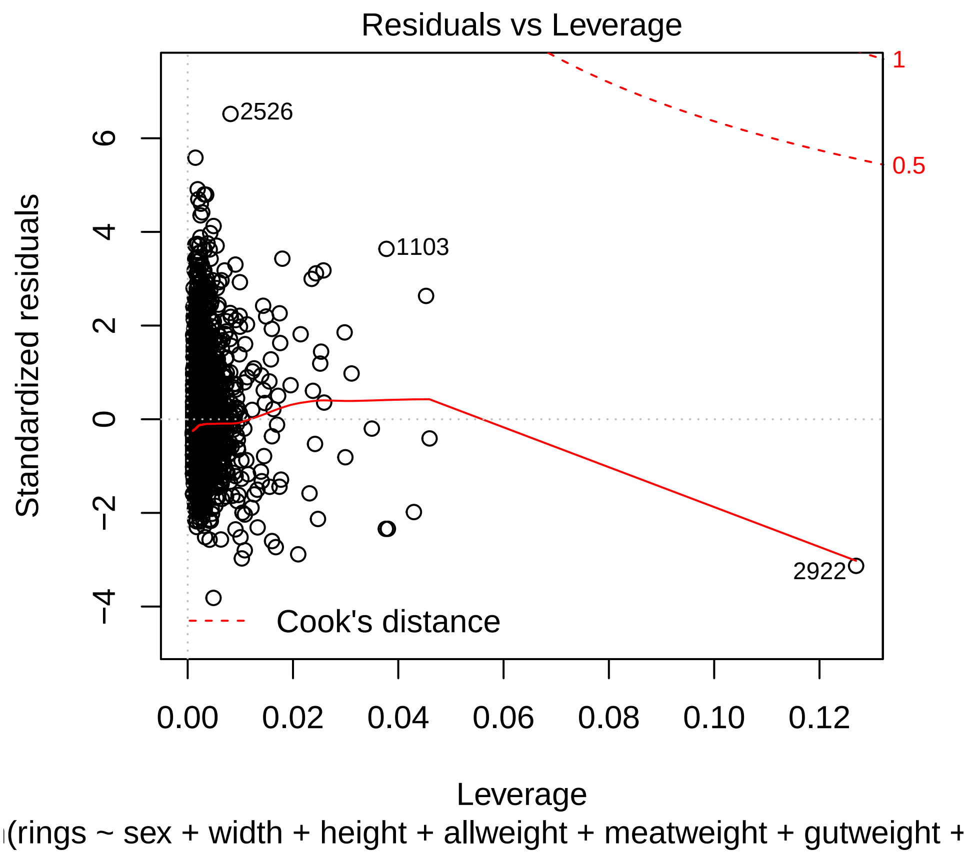


As we can see, observation 1217 has substantially more leverage than the other points. If our point is to come up with an equation to predict the number of rings an abalone shell will have based on its other characteristics, then it is probably a good idea to ignore this outlier and refit the model.

```
mod2<-lm(rings~sex+width+height+allweight+meatweight+gutweight+shellweight,
         data=abalone[-1217,])
summary(mod2)
```

```
##
## Call:
## lm(formula = rings ~ sex + width + height + allweight + meatweight +
##     gutweight + shellweight, data = abalone[-1217, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3163 -1.3111 -0.3215  0.8737 14.1862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.58744    0.32051  11.193  < 2e-16 ***
## sexI         -0.67658    0.11752  -5.757 9.37e-09 ***
## sexM          0.16724    0.09556   1.750   0.0802 .
## width         7.52387    1.17522   6.402 1.76e-10 ***
## height       22.31472    2.43134   9.178  < 2e-16 ***
## allweight     9.44291    0.81216  11.627  < 2e-16 ***
## meatweight  -19.81539    0.91922 -21.557  < 2e-16 ***
## gutweight   -11.48309    1.47117  -7.805 7.99e-15 ***
## shellweight   6.94778    1.26456   5.494 4.23e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.185 on 3167 degrees of freedom
## Multiple R-squared:  0.5343, Adjusted R-squared:  0.5331
## F-statistic: 454.2 on 8 and 3167 DF,  p-value: < 2.2e-16
```

```
plot(mod2,which=5)
```



One could persuasively argue that observation 2922 should be left out of the model as well. That exercise is left up to you!

G&H #6:

(a)

```
dat<-read.table("congress.txt")
mod1<-lm(dem_prop_88~.,data=dat)
summary(mod1)
```

```
##
## Call:
## lm(formula = dem_prop_88 ~ ., data = dat)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.113640 -0.016542 -0.003689  0.013293  0.148921
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.048e-01  1.357e-02  37.201  < 2e-16 ***
## state_id     2.013e-04  7.884e-05   2.553   0.0111 *
## district    -1.377e-04  1.156e-04  -1.192   0.2341
## incumbent    9.583e-03  3.723e-03   2.574   0.0105 *
```

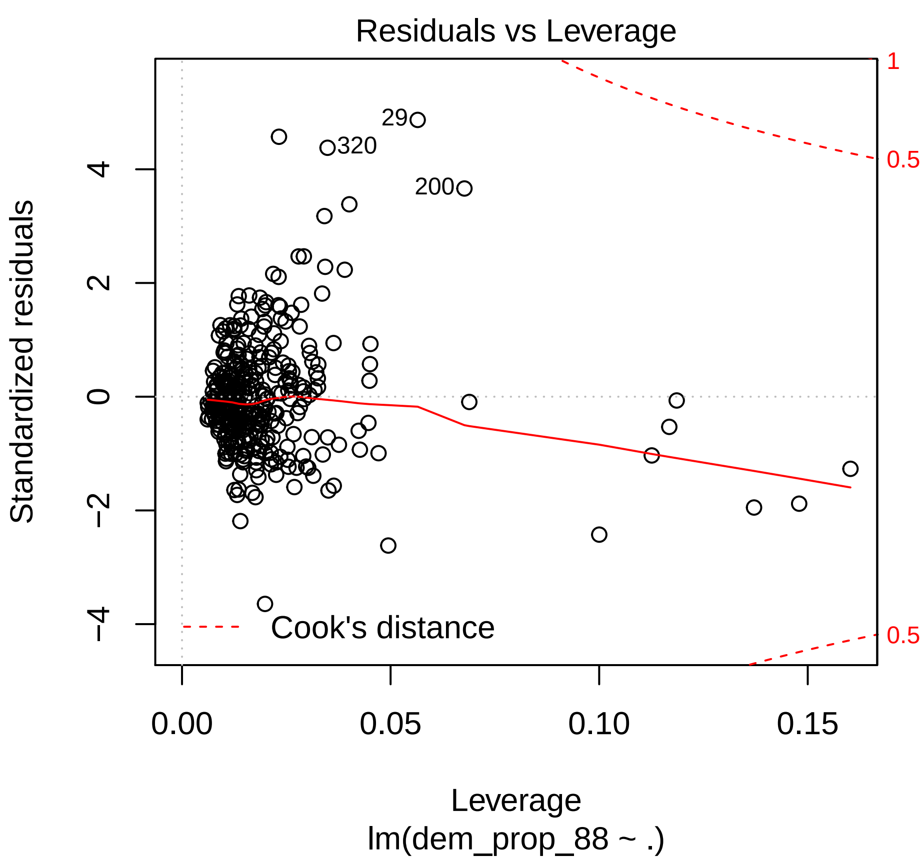
```
## dem_votes     1.877e-06  6.855e-08  27.378  < 2e-16 ***
## rep_votes    -2.292e-06  7.136e-08 -32.120  < 2e-16 ***
## dem_prop_86   7.752e-02  1.360e-02   5.701 2.57e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03149 on 342 degrees of freedom
## Multiple R-squared:  0.9732, Adjusted R-squared:  0.9728
## F-statistic:  2073 on 6 and 342 DF,  p-value: < 2.2e-16
```

A model using all of the provided variables explains an astonishing 97% of the variance in the data! Unsurprisingly, voter constituency, incumbency of the candidate, and previoues election results are all good predictors of the outcome of a current election. In this case, a voter's district doesn't seem to matter so much.

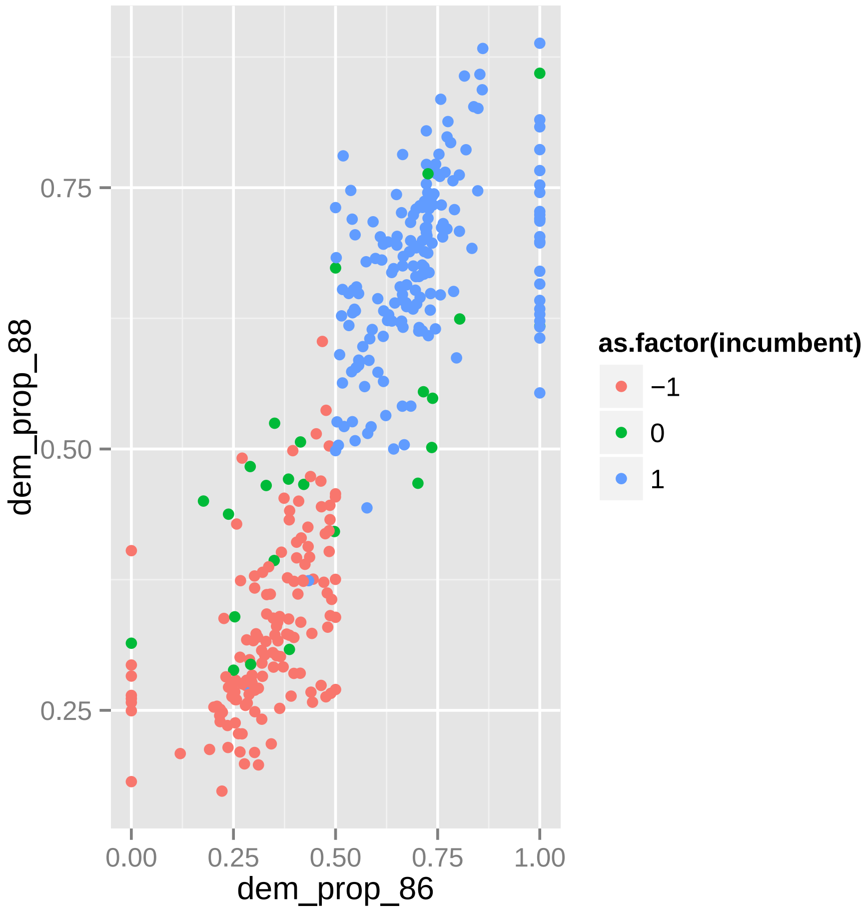
(b)

We note that there do seem to be some outliers in the dataset, and intuitively, I think it makes sense that some elections just wouldn't follow the trends, and would more be a function of specific political events or candidates. With this dataset in particular, we see that some of the proportions for votes in 1986 were either entirely Democratic or entirely Republican. This is probably because those were the only candidates in that particular election. In this case, that leads to fatter tails in the distribution of residuals than we would expect to see if they were normally distributed.

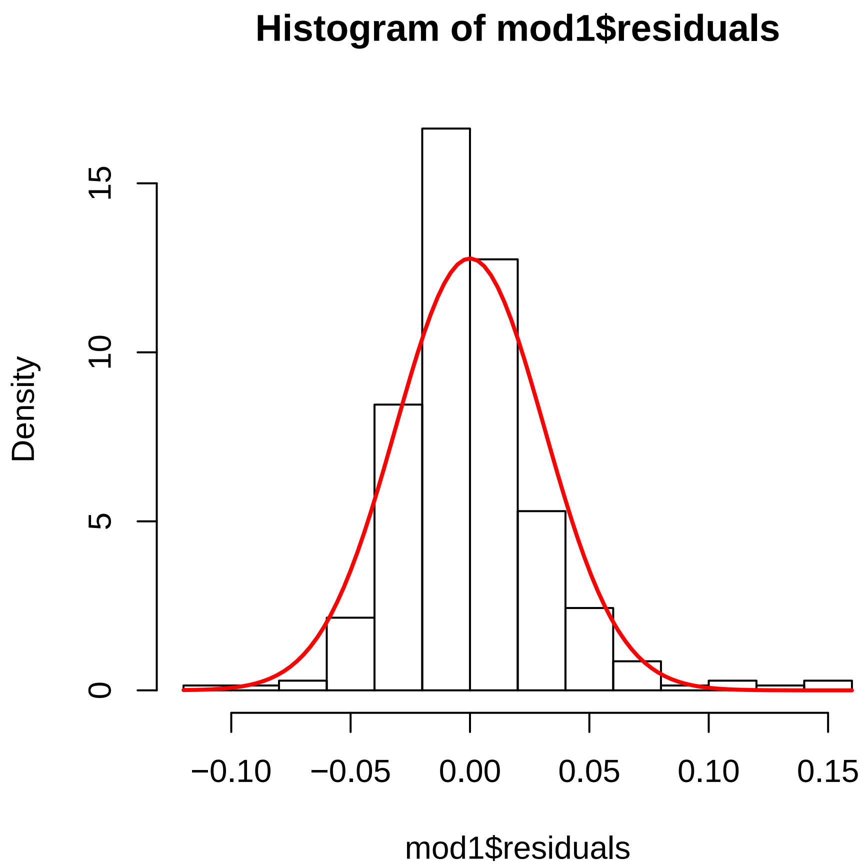
```r
plot(mod1,which=5)
```



```
ggplot(dat,aes(x=dem_prop_86,y=dem_prop_88,colour=as.factor(incumbent)))+geom_point()
```




```
hist(mod1$residuals,prob=T)
curve(dnorm(x,mean(mod1$residuals),sd(mod1$residuals)),add=T,lwd=2,col=2)
```



In order to minimize the effects of data outliers, we turn to robust regression. We will fit a t distribution to our model, i.e., we assume that the error follows a t distribution, i.e., fatter tails:

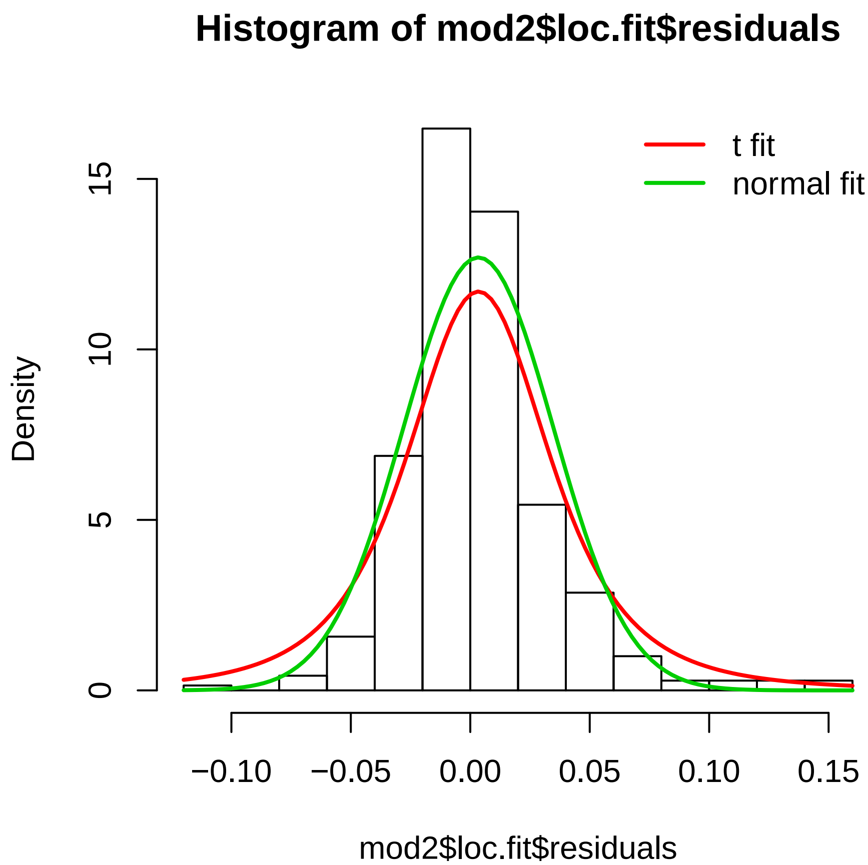
```
mod2<-tlm(dem_prop_88~.,data=dat)
summary(mod2)
```

```
## Location model :
##
## Call:
## tlm(lform = dem_prop_88 ~ ., data = dat)
##
## Residuals:
##        Min          1Q      Median          3Q         Max
## -0.1128550  -0.0130910  -0.0004776   0.0153691   0.1531407
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.234e-01  1.055e-02  49.619  < 2e-16 ***
## state_id     2.102e-04  6.129e-05   3.430 0.000678 ***
## district    -1.361e-04  8.984e-05  -1.515 0.130726
## incumbent    7.801e-03  2.895e-03   2.695 0.007384 **
## dem_votes    1.911e-06  5.329e-08  35.863  < 2e-16 ***
## rep_votes   -2.419e-06  5.547e-08 -43.607  < 2e-16 ***
## dem_prop_86  5.339e-02  1.057e-02   5.051 7.16e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Scale parameter(s) as estimated below)
##
##
## Scale Model :
##
## Call:
## tlm(lform = dem_prop_88 ~ ., data = dat)
##
## Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9999  -1.6907  -0.8433   1.4029   5.6108
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -7.8250     0.1071  -73.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Scale parameter taken to be  2 )
##
##
## Est. degrees of freedom parameter:  3
## Standard error for d.o.f:  NA
## No. of iterations of model : 22 in 0.014
## Heteroscedastic t Likelihood : 746.4485
```

We can see that the t distribution does a bit better job accommodating the tails of our error.

```r
hist(mod2$loc.fit$residuals,prob=T)
curve(dt.scaled(x,3,mean(mod2$loc.fit$residuals),sd(mod2$loc.fit$residuals)),
      add=T,col=2,lwd=2)  #metRology::dt.scaled
curve(dnorm(x,mean(mod2$loc.fit$residuals),sd(mod2$loc.fit$residuals)),add=T,col=3,lwd=2)
legend("topright",legend=c("t fit","normal fit"),col=2:3,lwd=2,bty="n")
```



(c)

Which model you prefer is really a matter of personal preference. I generally like simplicity in models, so unless the outliers are really causing problems, I would tend toward a simple linear model.

(d)

I think it makes a lot more sense to consider incumbency as a factor. This will fit a separate line for each incumbency category.