Homework 04

Generalized Linear Models

Yifu Dong

October 8, 2018

Data analysis

Poisson regression:

The folder `risky.behavior` contains data from a randomized trial targeting couples at high risk of HIV infection. The intervention provided counseling sessions regarding practices that could reduce their likelihood of contracting HIV. Couples were randomized either to a control group, a group in which just the woman participated, or a group in which both members of the couple participated. One of the outcomes examined after three months was "number of unprotected sex acts".

1. Model this outcome as a function of treatment assignment using a Poisson regression. Does the model fit well? Is there evidence of overdispersion?

```
#cleaning data
risky_behaviors$fupacts <- round(risky_behaviors$fupacts)

m1 <- glm(fupacts~factor(women_alone)+factor(couples),data=risky_behaviors,family = poisson)
display(m1)
```

```
## glm(formula = fupacts ~ factor(women_alone) + factor(couples),
##     family = poisson, data = risky_behaviors)
##                       coef.est coef.se
## (Intercept)            3.09     0.02
## factor(women_alone)1 -0.57     0.03
## factor(couples)1     -0.32     0.03
## ---
##   n = 434, k = 3
##   residual deviance = 12925.5, null deviance = 13298.6 (difference = 373.1)
```

```
summary.glm(m1)$dispersion
```

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

```
#check for overdispersion
m1_overdispersion <- glm(fupacts~factor(women_alone)+factor(couples),data=risky_behaviors,family = quasi
summary.glm(m1_overdispersion)$dispersion
```

```
## [1] 44.13468
```

We found that the difference is 373.1, much more than 2, so this model fits better than null model.

Then we use quasipoission to check overdisperssion, we found that the dispersion value is 44.13, which is much more than 1, so this is the evidence of overdispersion.

2. Next extend the model to include pre-treatment measures of the outcome and the additional pre-treatment variables included in the dataset. Does the model fit well? Is there evidence of overdispersion?

We now add another predictors in the dataset:

```
m2 <- glm(fupacts ~ factor(women_alone) + factor(couples) + factor(bs_hiv) + factor(sex),data = risky_b
display(m2)
```

```
## glm(formula = fupacts ~ factor(women_alone) + factor(couples) +
##     factor(bs_hiv) + factor(sex), family = poisson, data = risky_behaviors)
##                         coef.est coef.se
## (Intercept)              3.20     0.02
## factor(women_alone)1    -0.54     0.03
## factor(couples)1        -0.25     0.03
## factor(bs_hiv)positive -0.59     0.03
## factor(sex)man          -0.08     0.02
## ---
##   n = 434, k = 5
##   residual deviance = 12589.9, null deviance = 13298.6 (difference = 708.7)
```

```
summary.glm(m2)$dispersion
```

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

```
#check for overdispersion
m2_overdispersion <- glm(fupacts ~ factor(women_alone) + factor(couples) + factor(bs_hiv) + factor(sex)
summary.glm(m2_overdispersion)$dispersion
```

```
## [1] 42.35095
```

We found that after adding another predictors, the residual deviance falls down from 12925 to 12589. So this model fits better than the former model.

Also we use quasipoisson again to check the overdispersion, the overdispersion value is 42.35. So the new model is still overdispersised.

3. Fit an overdispersed Poisson model. What do you conclude regarding effectiveness of the intervention?

```
m3 <- glm(fupacts ~ factor(women_alone) + factor(couples) + factor(bs_hiv) + factor(sex)+bupacts,data =
display(m3)
```

```
## glm(formula = fupacts ~ factor(women_alone) + factor(couples) +
##     factor(bs_hiv) + factor(sex) + bupacts, family = quasipoisson,
##     data = risky_behaviors)
##                         coef.est coef.se
## (Intercept)              2.90     0.13
## factor(women_alone)1    -0.66     0.17
## factor(couples)1        -0.41     0.15
## factor(bs_hiv)positive -0.44     0.19
## factor(sex)man          -0.11     0.13
## bupacts                  0.01     0.00
## ---
##   n = 434, k = 6
##   residual deviance = 10200.4, null deviance = 13298.6 (difference = 3098.2)
##   overdispersion parameter = 30.0
```

The coefficient of bupacts is 0.01 and the std of bupacts is 0.00, so maybe we should scale this predictor:

```
risky_behaviors$bupacts_centered <- (risky_behaviors$bupacts - mean(risky_behaviors$bupacts)) / (2 * sd
m3 <- glm(fupacts ~ factor(women_alone) + factor(couples) + factor(bs_hiv) + factor(sex)+bupacts_center
display(m3)
```

```
## glm(formula = fupacts ~ factor(women_alone) + factor(couples) +
##     factor(bs_hiv) + factor(sex) + bupacts_centered, family = quasipoisson,
```

```
##     data = risky_behaviors)
##                          coef.est coef.se
## (Intercept)               3.18     0.12
## factor(women_alone)1     -0.66     0.17
## factor(couples)1         -0.41     0.15
## factor(bs_hiv)positive   -0.44     0.19
## factor(sex)man           -0.11     0.13
## bupacts_centered          0.69     0.06
## ---
##   n = 434, k = 6
##   residual deviance = 10200.4, null deviance = 13298.6 (difference = 3098.2)
##   overdispersion parameter = 30.0
```

The coefficient of women_alone,bupacts, and couples indicates that the intervention has a impact on unprotected sex acts. Only the woman took part in counseling sessions saw a 48% decrease in unprotected sex acts, and couples who took part in counseling sessions saw a decrease in unprotected sex acts of about 33%. So it's obvious to prove the influence of intervention.

4. These data include responses from both men and women from the participating couples. Does this give you any concern with regard to our modeling assumptions?

Yes, we think it is a problem since the data from couples also contains the data from women. So actually there might be problem of corlinearity. We think there might be extremely high positive corelations some data.

Comparing logit and probit:

Take one of the data examples from Chapter 5. Fit these data using both logit and probit model. Check that the results are essentially the same (after scaling by factor of 1.6)

In this question, we choose the switch data examples from Chapter 5:

First, we fit the model using "logit". We assume that the variables other than switch are all predictors:

```
dist1 <- wells_dt$dist/100
m_compare1 <- glm(switch~ arsenic+dist1+assoc+educ, data=wells_dt, family = binomial(link = "logit"))
display(m_compare1)
```

```
## glm(formula = switch ~ arsenic + dist1 + assoc + educ, family = binomial(link = "logit"),
##     data = wells_dt)
##             coef.est coef.se
## (Intercept) -0.16     0.10
## arsenic      0.47     0.04
## dist1       -0.90     0.10
## assoc       -0.12     0.08
## educ         0.04     0.01
## ---
##   n = 3020, k = 5
##   residual deviance = 3907.8, null deviance = 4118.1 (difference = 210.3)
```

Then let's try the probit model:

```
#probit
dist1 <- wells_dt$dist/100
m_compare2 <- glm(switch~ arsenic+dist1+assoc+educ, data=wells_dt, family = binomial(link = "probit"))
display(m_compare2)
```

```
## glm(formula = switch ~ arsenic + dist1 + assoc + educ, family = binomial(link = "probit"),
##     data = wells_dt)
##             coef.est coef.se
## (Intercept) -0.08     0.06
## arsenic      0.28     0.02
## dist1       -0.55     0.06
## assoc       -0.08     0.05
## educ         0.03     0.01
## ---
##   n = 3020, k = 5
##   residual deviance = 3909.7, null deviance = 4118.1 (difference = 208.4)
```

```
#coefficient of logit model
m_compare1$coefficients
```

```
## (Intercept)     arsenic       dist1       assoc        educ
## -0.15671166  0.46702159 -0.89611018 -0.12429998  0.04244661
```

```
#coefficient of probit model
m_compare2$coefficients
```

```
## (Intercept)     arsenic       dist1       assoc        educ
## -0.08446018  0.27650806 -0.54602184 -0.07964499  0.02658709
```

```
#scaling by factor of 1.6
(m_compare2$coefficients)*1.6
```

```
## (Intercept)     arsenic       dist1       assoc        educ
## -0.13513629  0.44241289 -0.87363495 -0.12743199  0.04253935
```

From the coefficient listed above, we can easily find that coefficients in a probit regression are typically close to logistic regression coefficients divided by 1.6.

Comparing logit and probit:

construct a dataset where the logit and probit models give different estimates.

I think the dataset wells is just what we want. From this dataset and the logit and probit models, the coefficients of logit model are :-0.15671166, 0.46702159, -0.89611018, -0.12429998, 0.04244661. On the other hand, the coefficients of probit model times 1.6 are:-0.13513629, 0.44241289, -0.87363495, -0.12743199 and 0.04253935. So the logit and probit models give different estimates.

Tobit model for mixed discrete/continuous data:

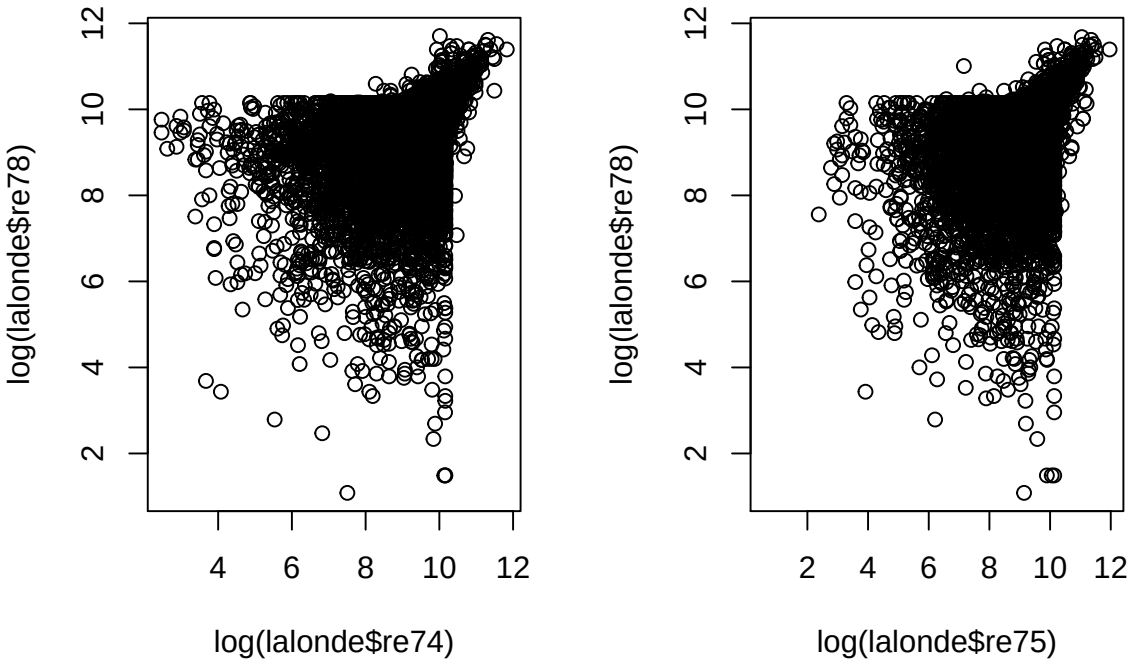
experimental data from the National Supported Work example are available in the folder `lalonde`. Use the treatment indicator and pre-treatment variables to predict post-treatment (1978) earnings using a tobit model. Interpret the model coefficients.

- sample: 1 = NSW; 2 = CPS; 3 = PSID.
- treat: 1 = experimental treatment group (NSW); 0 = comparison group (either from CPS or PSID) - Treatment took place in 1976/1977.
- age = age in years
- educ = years of schooling
- black: 1 if black; 0 otherwise.
- hisp: 1 if Hispanic; 0 otherwise.

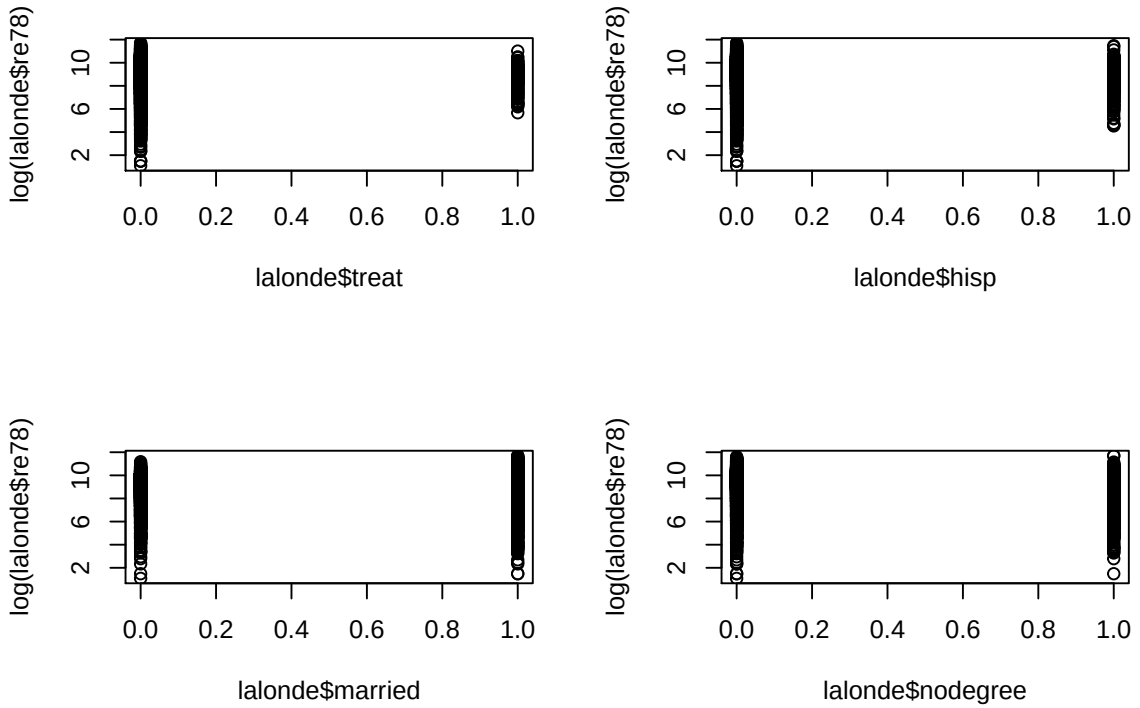
- married: 1 if married; 0 otherwise.
- nodegree: 1 if no high school diploma; 0 otherwise.
- re74, re75, re78: real earnings in 1974, 1975 and 1978
- educ_cat = 4 category education variable (1=<hs, 2=hs, 3=sm college, 4=college)

First, we need to find out the predictors effective for our model. Since age, educ, black, hisp, married, nodegree, and educ_cat are all binary or multilevel. So we first draw the plot of re74 and re75

```
par(mfrow=c(1,2))
plot(log(lalonde$re74),log(lalonde$re78))
plot(log(lalonde$re75),log(lalonde$re78))
```




```
par(mfrow=c(2,2))
plot(lalonde$treat,log(lalonde$re78))
plot(lalonde$hisp,log(lalonde$re78))
plot(lalonde$married,log(lalonde$re78))
plot(lalonde$nodegree,log(lalonde$re78))
```

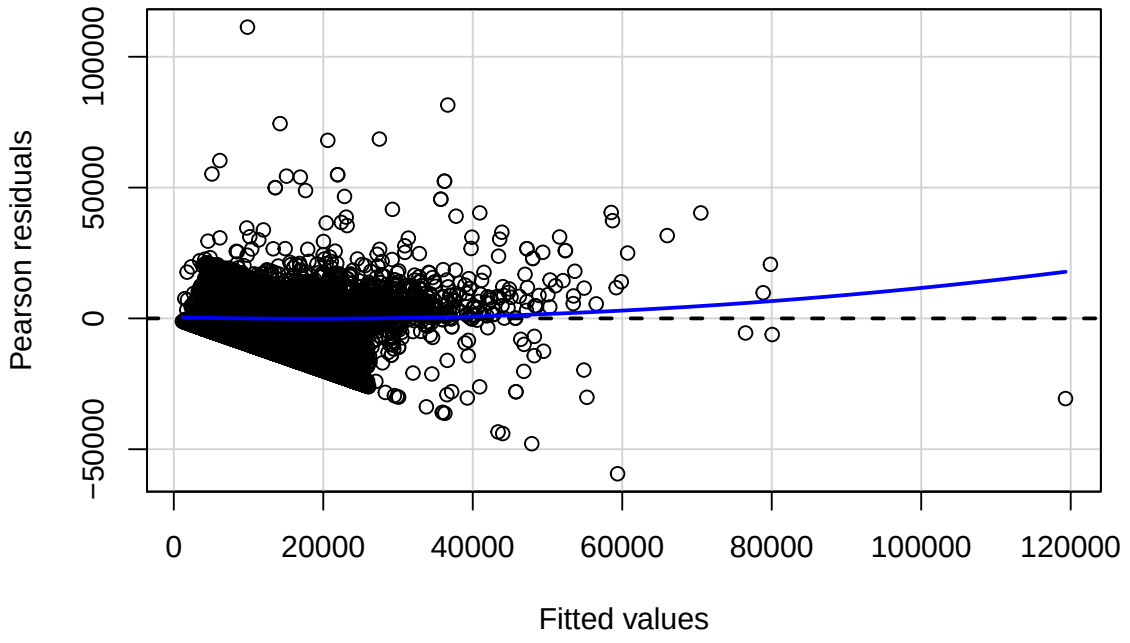


We found that this two variables are both probably related to re78. We don't know why real earning in 1978 can be related with other years, but it's not a bad choice to add them to the model.

As for the next plot, we try to find out the censoring in those predictors. And we can conclude from the plots that there is censoring the our predictors.

Thus we fit the model:

```
lalonde <- na.omit(lalonde)

m_tobitlm <- lm(re78~educ+black+hisp+nodegree+educ_cat4+re74+re75,data = lalonde)
residualPlot(m_tobitlm)
```



The plot shows that there is censoring in the fitted values of 0-40000.

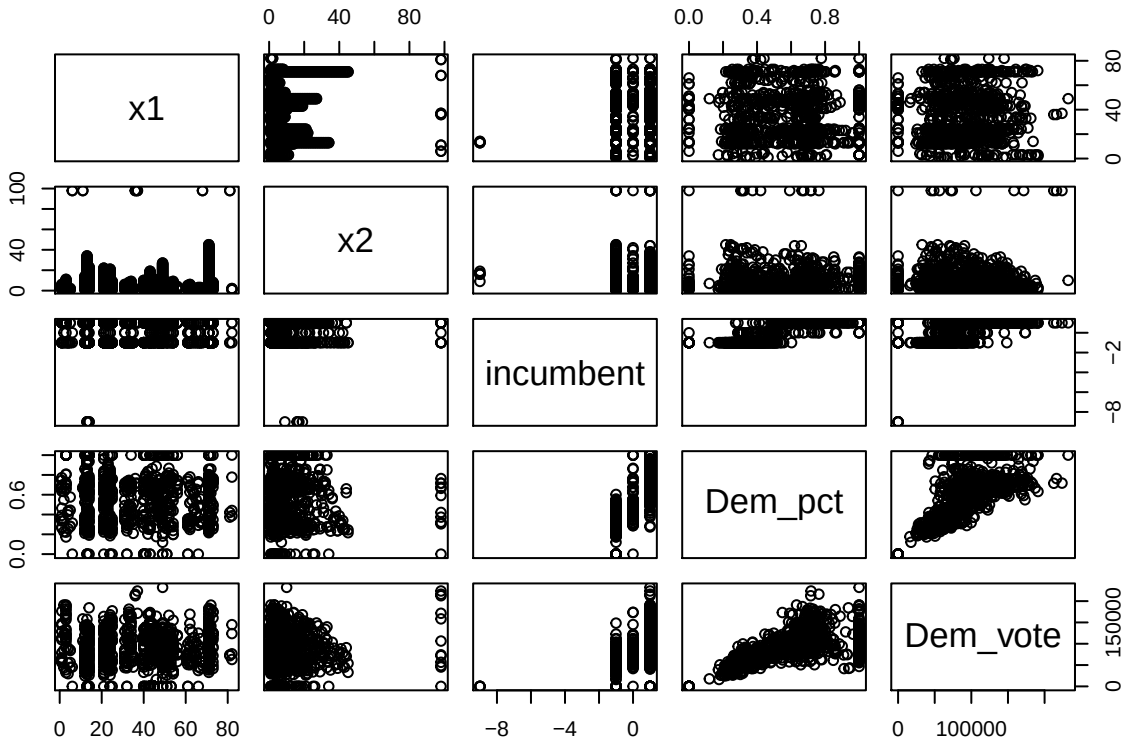
Robust linear regression using the t model:

The csv file `congress` has the votes for the Democratic and Republican candidates in each U.S. congressional district in between 1896 and 1992, along with the parties' vote proportions and an indicator for whether the incumbent was running for reelection. For your analysis, just use the elections in 1986 and 1988 that were contested by both parties in both years.

```
## Parsed with column specification:
## cols(
##   year = col_integer(),
##   x1 = col_integer(),
##   x2 = col_integer(),
##   incumbent = col_integer(),
##   Dem_vote = col_integer(),
##   Rep_vote = col_integer(),
##   Dem_pct = col_double(),
##   contested = col_logical()
## )
```

1. Fit a linear regression (with the usual normal-distribution model for the errors) predicting 1988 Democratic vote share from the other variables and assess model fit.

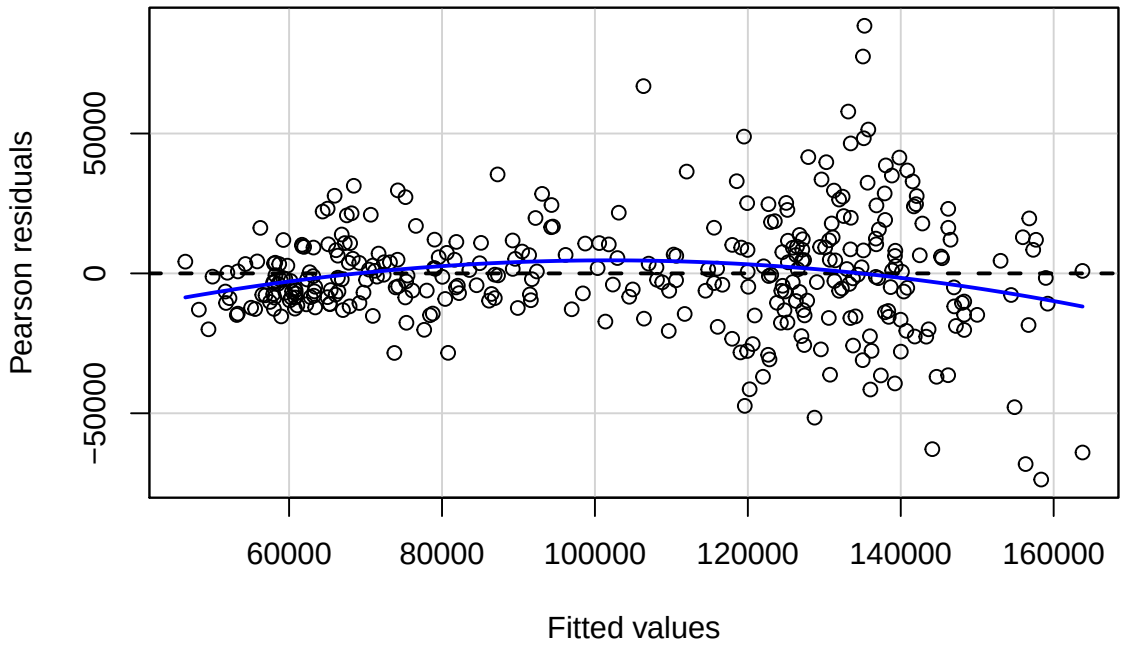
```
#Extract the data in 1986 and 1988

congress <- filter(congress,year==1986|year==1988)
congress1988 <- filter(congress,year==1988)

#cleaning data
congress1988 <- na.omit(congress1988)
congress1988 <- filter(congress1988,contested=="TRUE")

#fit a linear model
#Since this model is used for predicting 1988 democratic vote share, so rep_vote should not be our
#predictor.
pairs(congress[,c("x1","x2","incumbent","Dem_pct","Dem_vote")])
```




```
m_robust <- lm(Dem_vote~log(x1)+log(x2)+incumbent+Dem_pct,data=congress1988)
summary(m_robust)
```

```
##
## Call:
## lm(formula = Dem_vote ~ log(x1) + log(x2) + incumbent + Dem_pct,
##     data = congress1988)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -73688 -10277  -1199   9505  88504
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33020.9     7557.9   4.369 1.66e-05 ***
## log(x1)      -2676.6     1277.3  -2.095   0.0369 *
## log(x2)       -724.6     1014.4  -0.714   0.4755
## incumbent     3651.9     2376.2   1.537   0.1253
## Dem_pct     155526.8    11935.6  13.030  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20160 on 343 degrees of freedom
## Multiple R-squared:  0.7291, Adjusted R-squared:  0.7259
## F-statistic: 230.8 on 4 and 343 DF,  p-value: < 2.2e-16
```

```
residualPlot(m_robust)
```



We found that this model fits very well, the Adjusted R-squared is 0.726. And then we draw the residual plot, it show that the plots are almost normal distributed. However, we need to improve the model since the coefficient of Dem_pct and intercept are too large. Also, the residual still suffer from heteroscedasticity.

Moreover, x1,x2 are both insignificant.

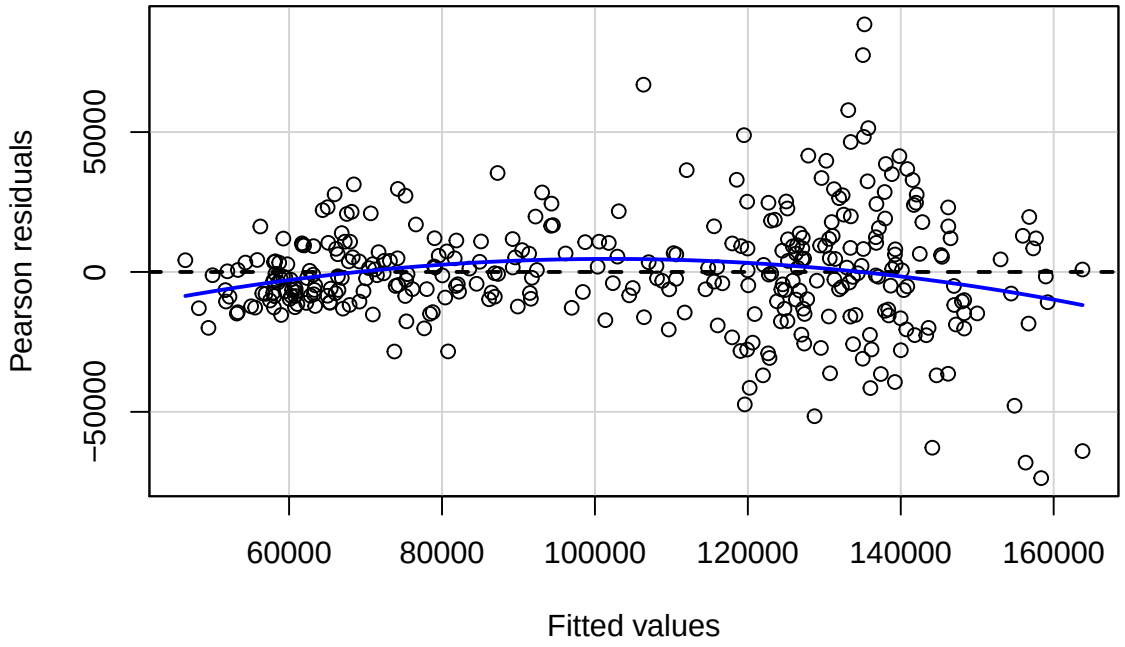
So we fit another model:

```
#scaling
Dem_pct <- (congress1988$Dem_pct - mean(congress1988$Dem_pct)) / (2 * sd(congress1988$Dem_pct))
x1 <- (congress1988$x1 - mean(congress1988$x1)) / (2 * sd(congress1988$x1))
x2 <- (congress1988$x2 - mean(congress1988$x2)) / (2 * sd(congress1988$x2))
Dem_vote <- (congress1988$Dem_vote-mean(congress1988$Dem_vote))/(2 * sd(congress1988$Dem_vote))


#add interaction and reduce x1&x2, trying to reduce heteroscedasticity.
m_robust1 <- lm(Dem_vote~congress1988$incumbent*Dem_pct)
summary(m_robust1)
```

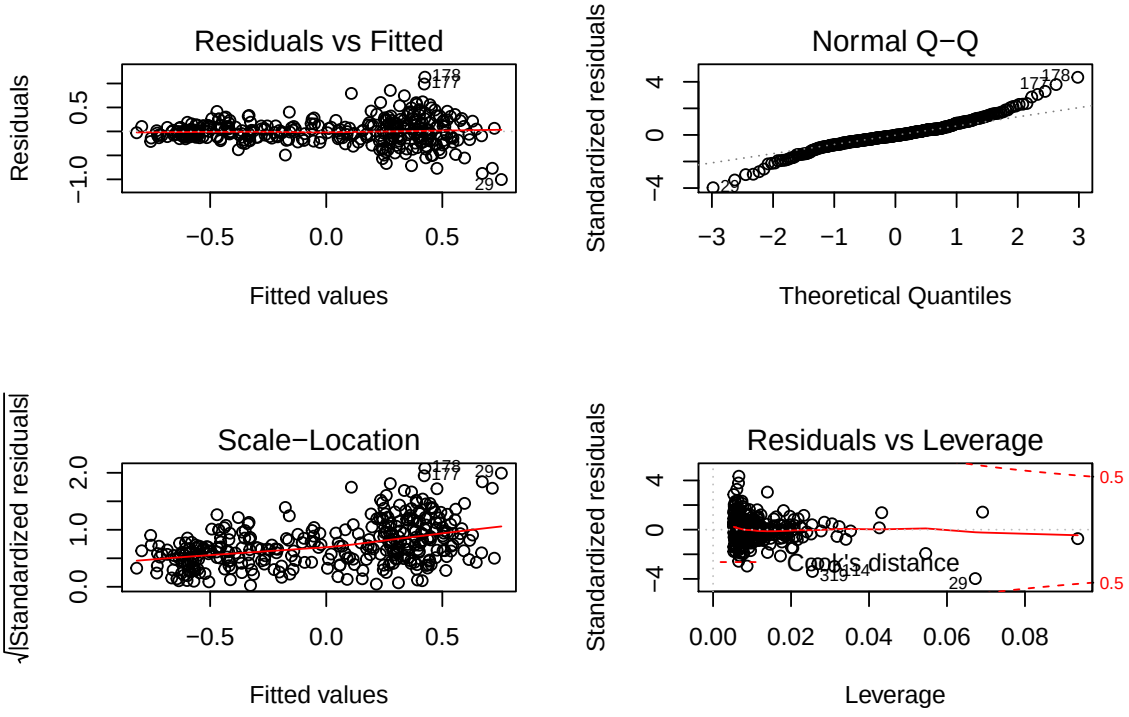
```
##
## Call:
## lm(formula = Dem_vote ~ congress1988$incumbent * Dem_pct)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00393 -0.12771 -0.01001  0.11543  1.13108
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     0.04885    0.02931   1.667   0.0964 .
## congress1988$incumbent          0.03046    0.03192   0.954   0.3406
## Dem_pct                         0.79428    0.06026  13.181   <2e-16 ***
## congress1988$incumbent:Dem_pct -0.12490    0.05817  -2.147   0.0325 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.2615 on 344 degrees of freedom
## Multiple R-squared:  0.7288, Adjusted R-squared:  0.7264
## F-statistic: 308.1 on 3 and 344 DF,  p-value: < 2.2e-16
```

```
residualPlot(m_robust)
```




```
#
par(mfrow=c(2,2))
plot(m_robust1)
```



Then we found that the Adjusted R-Squared is better a little bit, but the residual seems still suffer from heteroscedasticity. But the interaction is significant. Consider the interaction is not bad for our model, so we choose this model.

2. Fit a t-regression model predicting 1988 Democratic vote share from the other variables and assess model fit; to fit this model in R you can use the `vglm()` function in the VGLM package or `tlm()` function in the hett package.

```
#robust model
m_robust2 <- tlm(Dem_vote~congress1988$incumbent*Dem_pct)
summary(m_robust2)
```

```
## Location model :
##
## Call:
## tlm(lform = Dem_vote ~ congress1988$incumbent * Dem_pct)
##
## Residuals:
##       Min         1Q     Median         3Q        Max
## -1.081846  -0.120357  -0.002717   0.127687   1.124831
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    0.052210   0.023260   2.245  0.02543 *
## congress1988$incumbent        -0.007376   0.025334  -0.291  0.77112
## Dem_pct                        0.878235   0.047830  18.362  < 2e-16 ***
## congress1988$incumbent:Dem_pct -0.129994   0.046166  -2.816  0.00515 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Scale parameter(s) as estimated below)
##
##
## Scale Model :
##
## Call:
## tlm(lform = Dem_vote ~ congress1988$incumbent * Dem_pct)
##
## Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0000  -1.7280  -0.8025   1.3557   5.4898
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.5499     0.1072  -33.11   <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 : 13 in 0.037
## Heteroscedastic t Likelihood : 0.7472359
```

Surprisingly we found that Heteroscedastic t Likelihood of this model is 0.747, 0.747<1.96, so this model isn't influenced much by heteroscedasticity, but it cannot solve this problem totally. Also, this model fits well. So we choose this model.

3. Which model do you prefer?

I prefer the t-regression model, because this model is influenced less by heteroscedasticity than the linear regression model. And this model is also less influenced by outlying data points. So in this case, t-regression model is the better one.

Robust regression for binary data using the robit model:

Use the same data as the previous example with the goal instead of predicting for each district whether it was won by the Democratic or Republican candidate.

1. Fit a standard logistic or probit regression and assess model fit.

As is mentioned, we first create a viriable to represent whether it was won by the Democratic or Republican.

```
congress <- read_csv("~/dongyifu/Desktop/congress.csv")  ##this is the absolute path
```

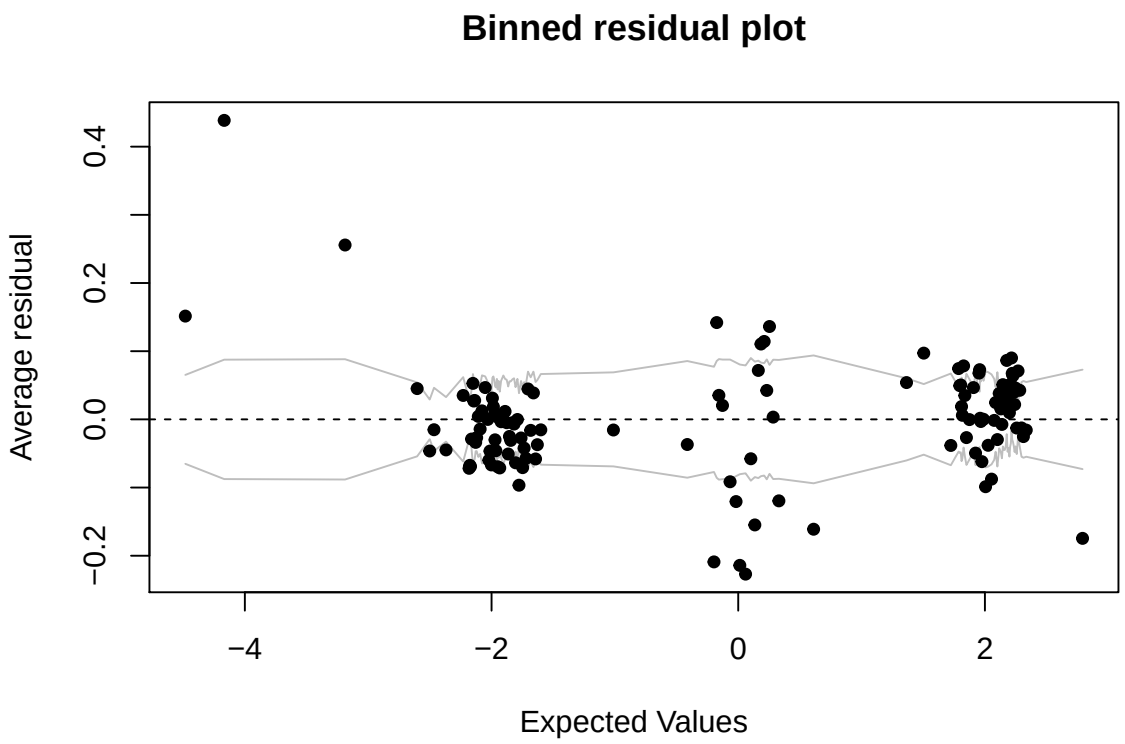
```
## Parsed with column specification:
## cols(
##   year = col_integer(),
##   x1 = col_integer(),
##   x2 = col_integer(),
##   incumbent = col_integer(),
##   Dem_vote = col_integer(),
##   Rep_vote = col_integer(),
##   Dem_pct = col_double(),
##   contested = col_logical()
## )
```

```
#cleaning data
congress <- na.omit(congress)
congress <- filter(congress,contested==TRUE)


congressDem <- filter(congress,congress$Dem_vote>congress$Rep_vote)
congressRep <- filter(congress,congress$Dem_vote<congress$Rep_vote)
#create a binary variable to represent whether it was won by the Democratic or Republican.

congress$won <- ifelse(congress$Dem_vote>congress$Rep_vote,1,0)


#fit a logistic model

m_won <- glm(won~log(x1)+log(x2)+invlogit(incumbent),data=congress,family = binomial(link = "logit"))
summary(m_won)
```

```
##
## Call:
## glm(formula = won ~ log(x1) + log(x2) + invlogit(incumbent),
##     family = binomial(link = "logit"), data = congress)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -2.9055  -0.5336  -0.1743   0.5104   3.0978
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -4.97114    0.11052  -44.98   <2e-16 ***
## log(x1)              0.25679    0.02564   10.02   <2e-16 ***
## log(x2)             -0.04637    0.02079   -2.23   0.0257 *
## invlogit(incumbent)  8.54909    0.11311   75.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22992  on 16588  degrees of freedom
## Residual deviance: 13494  on 16585  degrees of freedom
## AIC: 13502
##
## Number of Fisher Scoring iterations: 4
```

```
#residual plot
binnedplot(predict(m_won),resid(m_won,type="response"))
```



While I don't add invlogit transformation for incumbent, the residual deviance is 15983, now while I use invlogit(incumbent) as our predictor, this model fits much better. The residual deviance falls down to 13494.

Also, the residual plot shows above. Most of the plots are in the interval. So we can say this model is not bad.

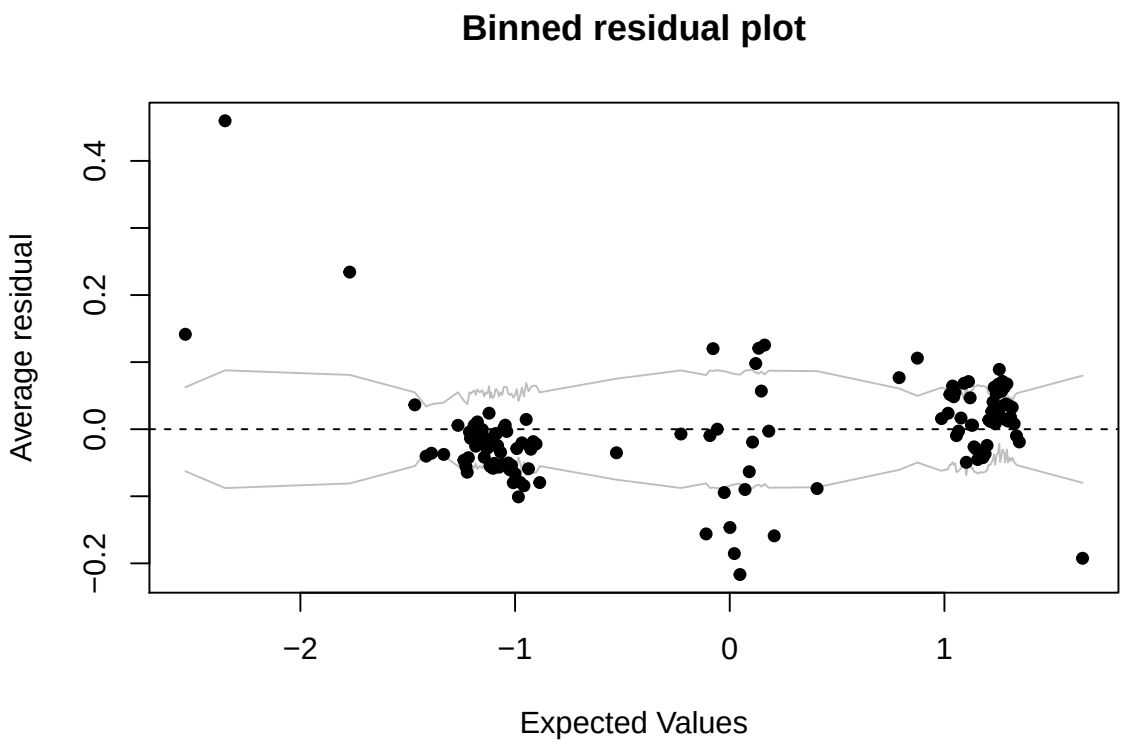
2. Fit a robit regression and assess model fit.

```
#robit regression
```

```r
#first we create a latent error e. From our textbook, we know that robit model is similar with a logit
e2 <- rt.scaled(nrow(congress),df=4,mean = 0,sd=1)
m_robit <- glm(won~log(x1)+log(x2)+invlogit(incumbent)+e2,data=congress,family = binomial(link = "probi
summary(m_robit)
```

```
##
## Call:
## glm(formula = won ~ log(x1) + log(x2) + invlogit(incumbent) +
##     e2, family = binomial(link = "probit"), data = congress)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1148  -0.5618  -0.1340   0.5237   3.3840
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -2.81775    0.05885 -47.884   <2e-16 ***
## log(x1)              0.14934    0.01405  10.631   <2e-16 ***
## log(x2)             -0.02766    0.01138  -2.431    0.015 *
## invlogit(incumbent)  4.85286    0.05869  82.694   <2e-16 ***
## e2                  -0.01117    0.00823  -1.358    0.175
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22992  on 16588  degrees of freedom
## Residual deviance: 13666  on 16584  degrees of freedom
## AIC: 13676
##
## Number of Fisher Scoring iterations: 4
```

```r
binnedplot(predict(m_robit),resid(m_robit,type="response"))
```



3. Which model do you prefer?

I don't know whether I create the robit model correctly. What I get by creating the robit model fits not better than the logit or probit model. However, I would say I prefer the robit model since it can downweights the discordant data so that the model better fits the main part of the data.

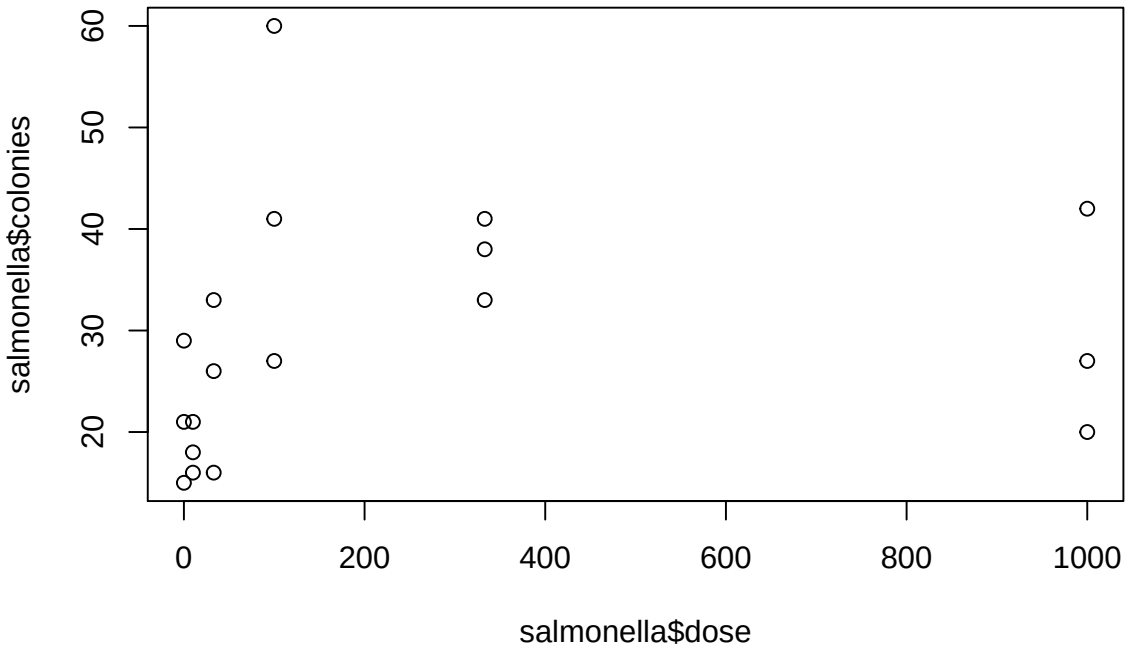
Salmonellla

The `salmonella` data was collected in a salmonella reverse mutagenicity assay. The predictor is the dose level of quinoline and the response is the numbers of revertant colonies of TA98 salmonella observed on each of three replicate plates. Show that a Poisson GLM is inadequate and that some overdispersion must be allowed for. Do not forget to check out other reasons for a high deviance.

```
data(salmonella)
?salmonella
```

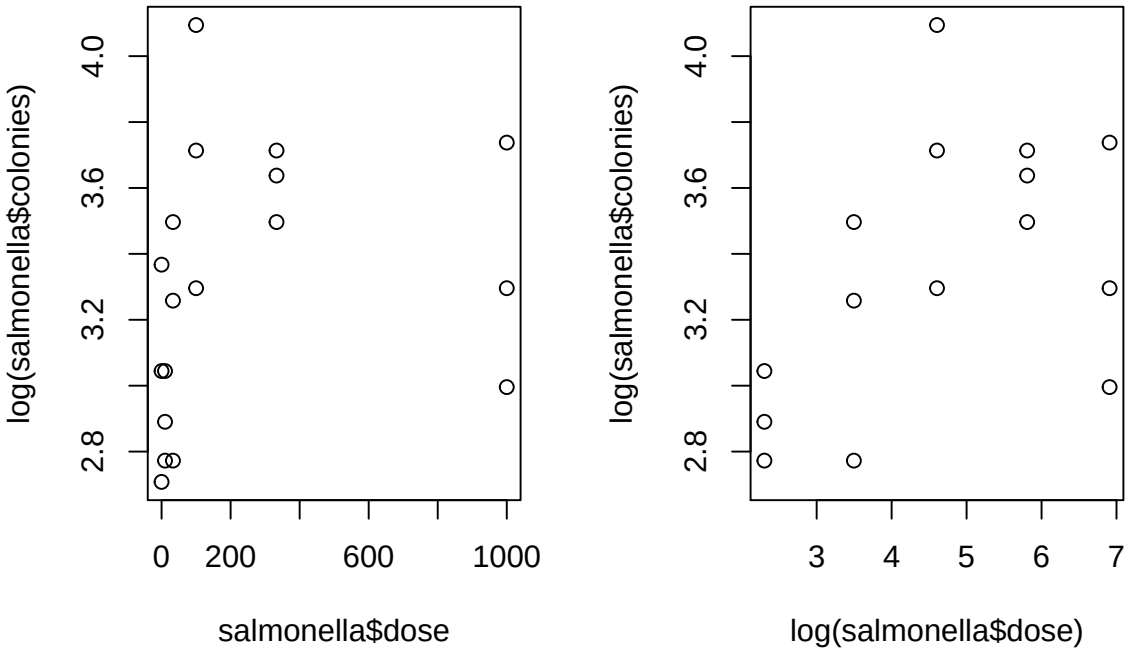
When you plot the data you see that the number of colonies as a function of dose is not monotonic especially around the dose of 1000.

```
plot(x=salmonella$dose,y=salmonella$colonies)
```



Since we are fitting log linear model we should look at the data on log scale. Also becase the dose is not equally spaced on the raw scale it may be better to plot it on the log scale as well.

```
#look at the data on log scale
par(mfrow=c(1,2))
plot(x=salmonella$dose,y=log(salmonella$colonies))
plot(x=log(salmonella$dose),y=log(salmonella$colonies))
```



This shows that the trend is not monotonic. Hence when you fit the model and look at the residual you will see a trend.

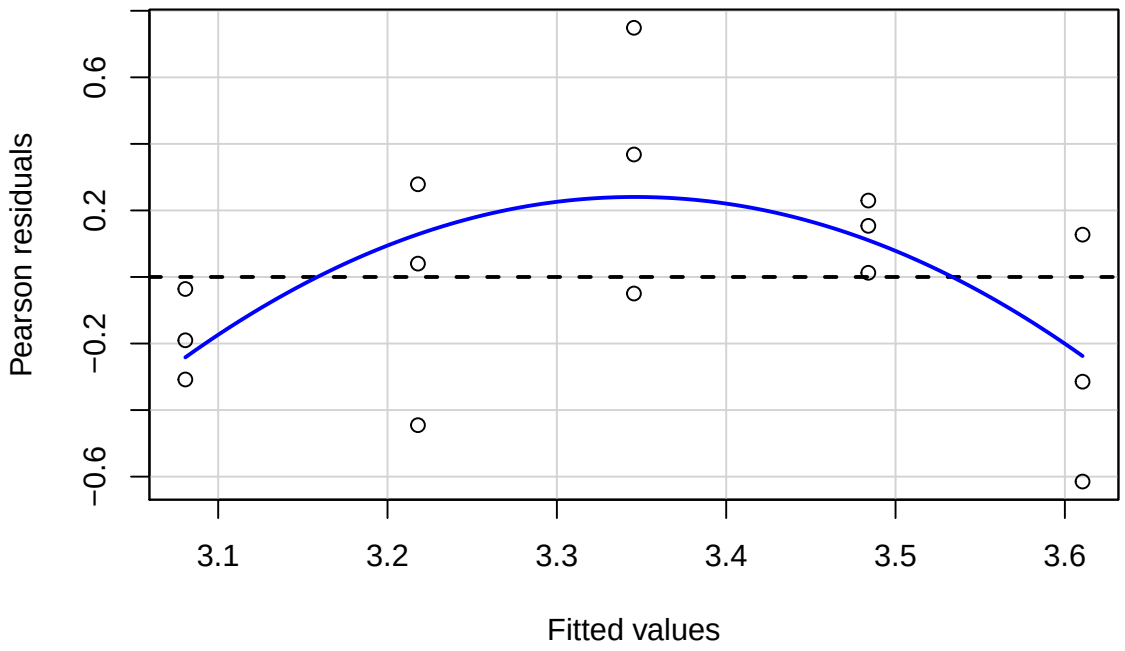
```
#fit the model
salmonella1 <- salmonella[4:18,]
salmonella1
```

```
##    colonies dose
## 4        16   10
## 5        18   10
## 6        21   10
## 7        16   33
## 8        26   33
## 9        33   33
## 10       27  100
## 11       41  100
## 12       60  100
## 13       33  333
## 14       38  333
## 15       41  333
## 16       20 1000
## 17       27 1000
## 18       42 1000
```

```
lm_salmon <- lm(log(colonies)~log(dose),data=salmonella1)
summary(lm_salmon)
```

```
##
## Call:
## lm(formula = log(colonies) ~ log(dose), data = salmonella1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61472 -0.24911  0.01258  0.19165  0.74883
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.81566    0.27909  10.089 1.62e-07 ***
## log(dose)    0.11506    0.05693   2.021   0.0643 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3593 on 13 degrees of freedom
## Multiple R-squared:  0.2391, Adjusted R-squared:  0.1806
## F-statistic: 4.085 on 1 and 13 DF,  p-value: 0.06435
```

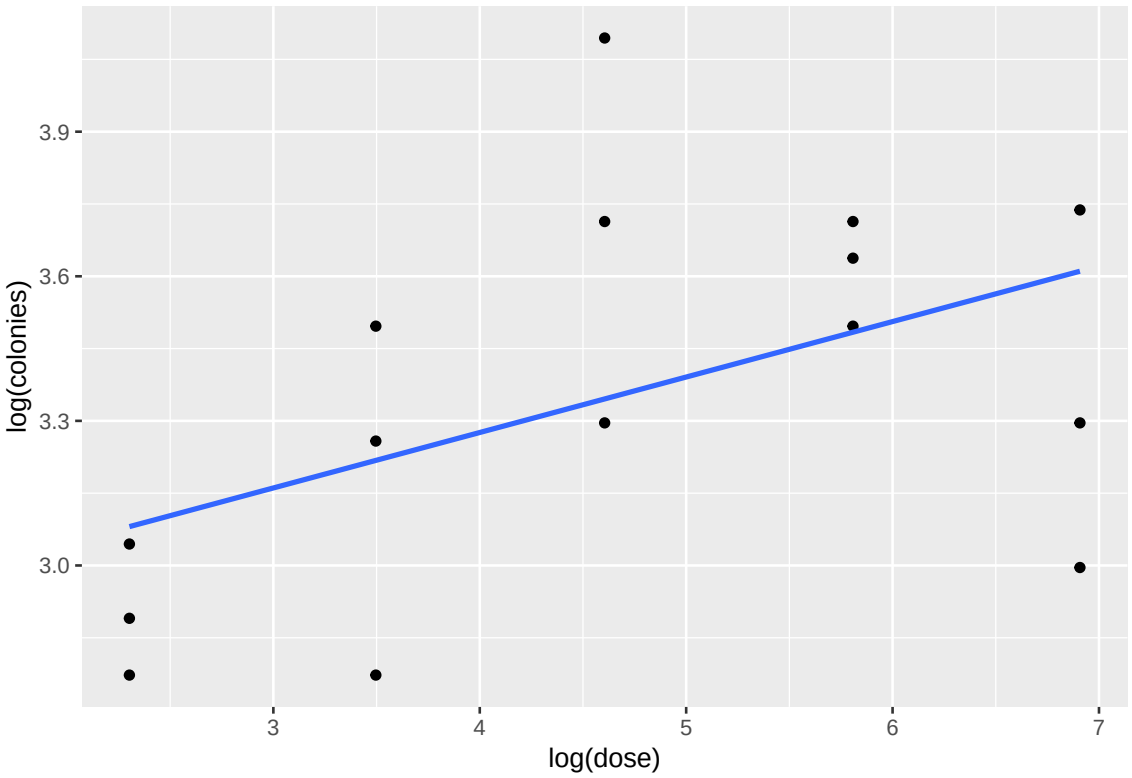
```
residualPlot(lm_salmon)
```



The residual plot does have a trend.

The lack of fit is also evident if we plot the fitted line onto the data.

```
b <- lm_salmon$coefficients

#y value
yhat=exp(b[1]+b[2]*log(salmonella1$dose))

#plot fitted line
ggplot(salmonella1,mapping = aes(log(dose),log(colonies)))+
  geom_point()+
  geom_smooth(method = "lm", se=FALSE)
```



How do we adress this problem? The serious problem to address is the nonlinear trend of dose ranther than the overdispersion since the line is missing the points. Let's add a beny line with 4th order polynomial.

For this part, we'd better not use linear model since we have proved that linear trend doesn't work well for our data.

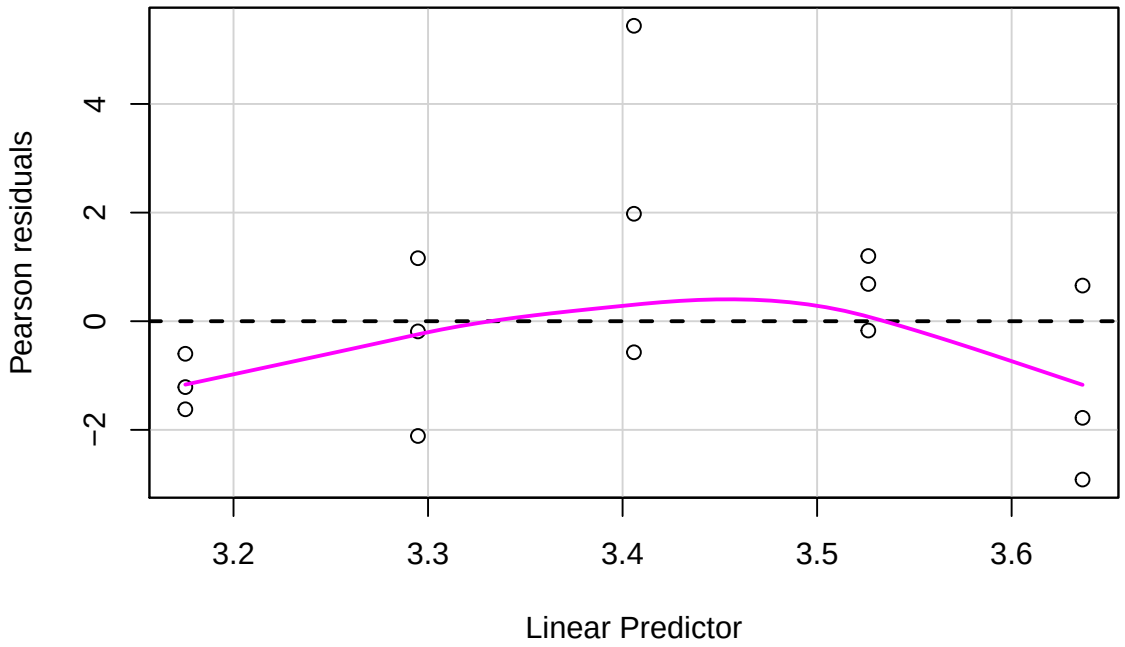
```
poisson_salmon <- glm(salmonella1$colonies~log(salmonella1$dose),family = poisson)
display(poisson_salmon)
```

```
## glm(formula = salmonella1$colonies ~ log(salmonella1$dose), family = poisson)
##                       coef.est coef.se
## (Intercept)           2.94     0.15
## log(salmonella1$dose) 0.10     0.03
## ---
##   n = 15, k = 2
##   residual deviance = 54.3, null deviance = 66.4 (difference = 12.1)
```

The residual deviance is 54.3, while null deviance is 66.4. Thus this model look good.

The resulting residual looks nice and if you plot it on the raw data. Whether the trend makes real contextual sense will need to be validated but for the given data it looks feasible.

```
residualPlot(poisson_salmon)
```



Dispite the fit, the overdispersion still exists so we'd be better off using the quasi Poisson model.

```
quasipoisson_salmon <- glm(salmonella1$colonies~log(salmonella1$dose),family = quasipoisson)
summary(poisson_salmon)
```

```
##
## Call:
## glm(formula = salmonella1$colonies ~ log(salmonella1$dose), family = poisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2071  -1.4975  -0.1883   0.8972   4.7854
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            2.94461    0.14892  19.773  < 2e-16 ***
## log(salmonella1$dose)  0.10016    0.02893   3.462 0.000537 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 66.399  on 14  degrees of freedom
## Residual deviance: 54.299  on 13  degrees of freedom
## AIC: 136.18
##
## Number of Fisher Scoring iterations: 4
```

We notice that the dispersion parameter for this model is 1, which means that the overdispersion of this model doesn't exist.

Ships

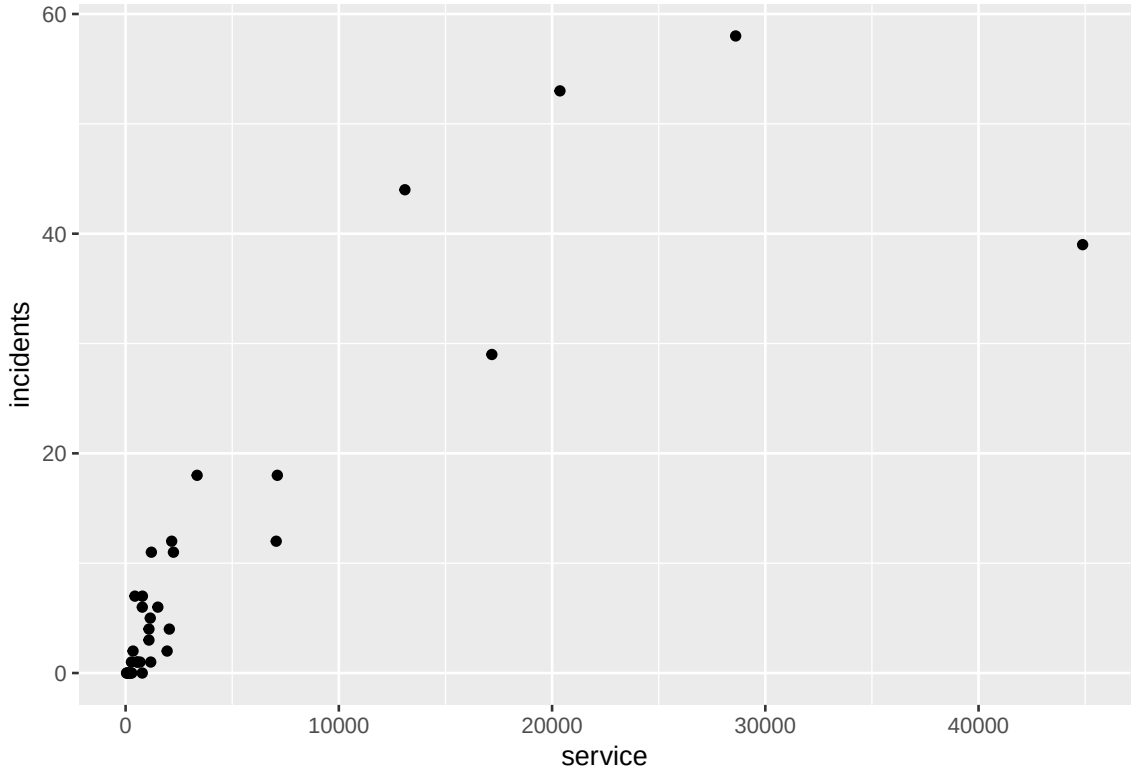
The `ships` dataset found in the MASS package gives the number of damage incidents and aggregate months of service for different types of ships broken down by year of construction and period of operation.

```
data(ships)
?ships
```

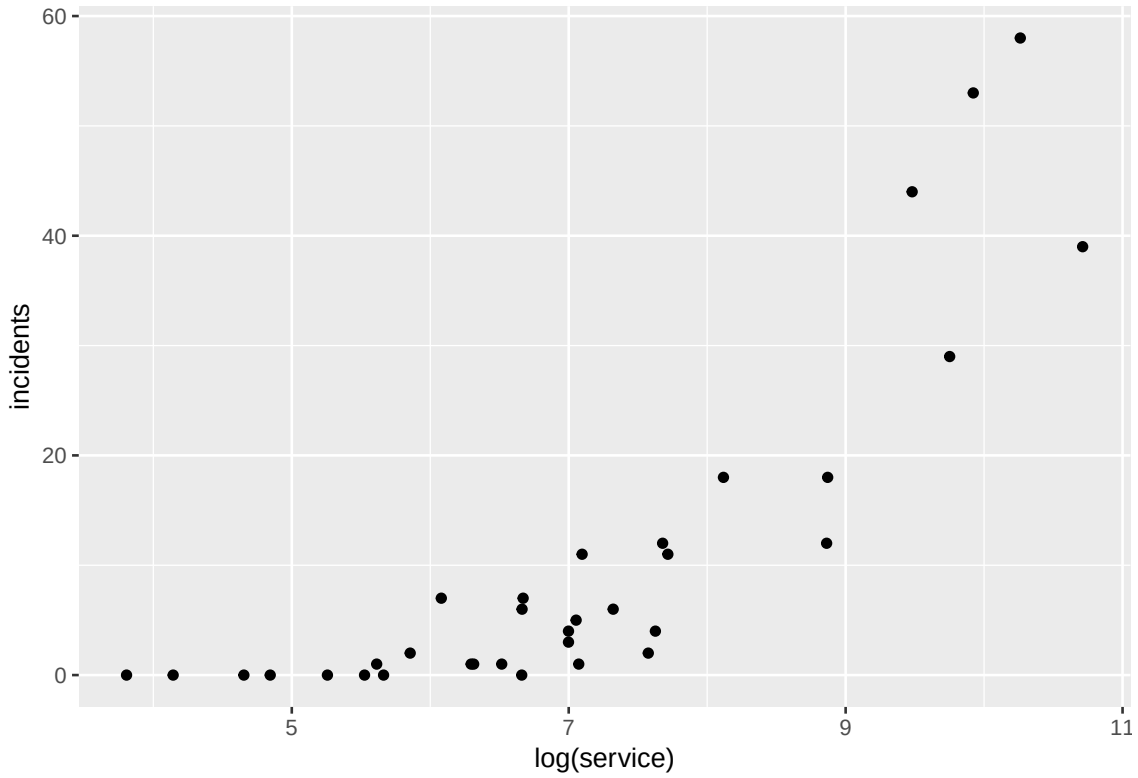
Develop a model for the rate of incidents, describing the effect of the important predictors.

We can fit the model using binomial regression model:

```
#For the rate of incidents, we need to add exposure to the poisson function
#We add offset = log(ships$service) to our function


#clean data
ships <- filter(ships,service>0)
ships <- na.omit(ships)

period <- (ships$period - mean(ships$period)) / (2 * sd(ships$period))
year <- (ships$year - mean(ships$year)) / (2 * sd(ships$year))
service <- (ships$service - mean(ships$service)) / (2 * sd(ships$service))

par(mfrow=c(1,2))
ggplot(data = ships)+
  geom_point(mapping = aes(x=service,y=incidents))
```




```
ggplot(data = ships,mapping = aes(x=log(service),y=incidents))+
  geom_point(mapping = aes(x=log(service),y=incidents))
```



Notice that after adding log function to servcice, the positive relation between service and incidents is much clear, so we decide to use log(service) as one of our predictors.

Also, we can add the quadratic form of log(service) since the ggplot shown above indicates a quadratic relation.

```
m_ships <- glm(ships$incidents~period+ships$type+log(ships$service)^2+year, family= poisson, offset = lo
summary(m_ships)
```

```
##
## Call:
## glm(formula = ships$incidents ~ period + ships$type + log(ships$service)^2 +
##     year, family = poisson, offset = log(ships$service))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2355  -1.0345  -0.4454   0.6005   2.8353
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -4.7443     0.7989  -5.939 2.87e-09 ***
## period                0.3338     0.1227   2.721   0.0065 **
## ships$typeB          -0.3302     0.2613  -1.264   0.2063
## ships$typeC          -0.7363     0.3413  -2.157   0.0310 *
## ships$typeD          -0.2842     0.2920  -0.973   0.3304
## ships$typeE           0.3359     0.2427   1.384   0.1662
## log(ships$service)^2 -0.1135     0.0993  -1.143   0.2529
## year                  0.3676     0.1430   2.570   0.0102 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 146.328  on 33  degrees of freedom
## Residual deviance:  58.114  on 26  degrees of freedom
## AIC: 171.98
##
## Number of Fisher Scoring iterations: 5
```

We found this model fit very well, althoug we cannot compare AIC with other model, but the residual deviance is 58, whereas the null deviance is 614.

So this is the model which describes the effect of important indicators. The more service a ship has, the more possible the ship breaks down or has incidents. This is the most important indicator. Also, the the type of ship, year and period are also important for the rate of incidents.

Australian Health Survey

The `dvisits` data comes from the Australian Health Survey of 1977-78 and consist of 5190 single adults where young and old have been oversampled.

```
data(dvisits)
?dvisits
```

1. Build a Poisson regression model with doctorco as the response and sex, age, agesq, income, levyplus, freepoor, freerepa, illness, actdays, hscore, chcond1 and chcond2 as possible predictor variables. Considering the deviance of this model, does this model fit the data?

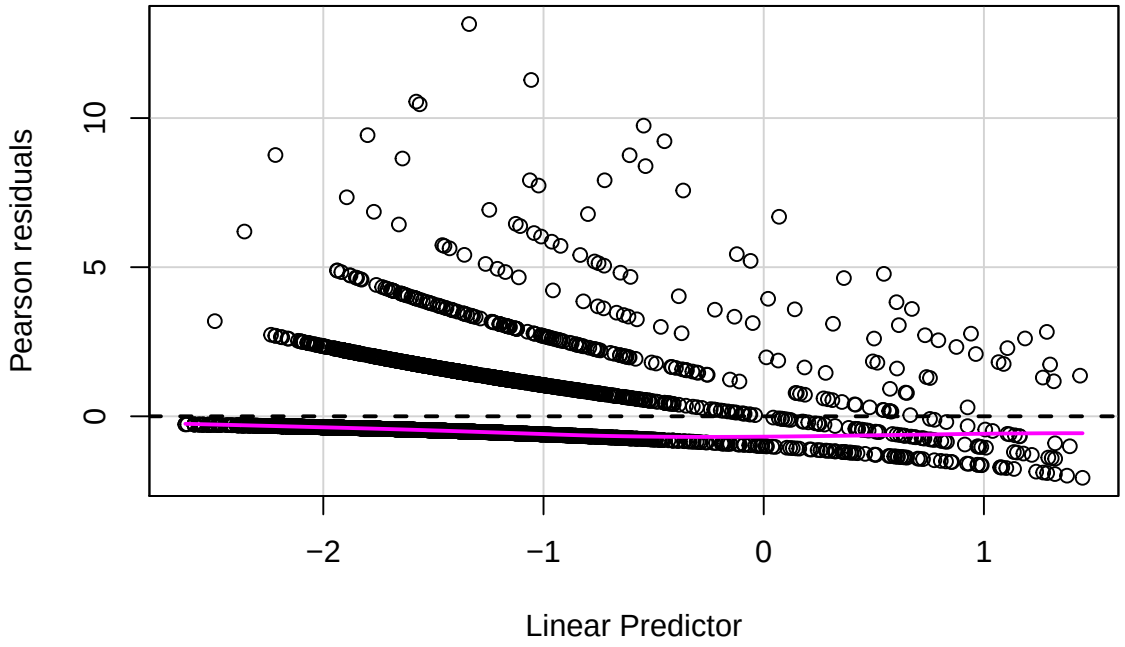
First let's simply check which variables are related to doctorco. We use pairs() to check the relation.

```
AUSHealth <- glm(doctorco ~ sex + age + agesq + income + levyplus + freepoor + freerepa + illness + acto
summary(AUSHealth)
```

```
##
## Call:
## glm(formula = doctorco ~ sex + age + agesq + income + levyplus +
##     freepoor + freerepa + illness + actdays + hscore + chcond1 +
##     chcond2, family = poisson(link = log), data = dvisits)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9170  -0.6862  -0.5743  -0.4839   5.7005
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.223848   0.189816 -11.716   <2e-16 ***
## sex          0.156882   0.056137   2.795   0.0052 **
## age          1.056299   1.000780   1.055   0.2912
## agesq       -0.848704   1.077784  -0.787   0.4310
## income      -0.205321   0.088379  -2.323   0.0202 *
## levyplus     0.123185   0.071640   1.720   0.0855 .
## freepoor    -0.440061   0.179811  -2.447   0.0144 *
## freerepa     0.079798   0.092060   0.867   0.3860
## illness      0.186948   0.018281  10.227   <2e-16 ***
## actdays      0.126846   0.005034  25.198   <2e-16 ***
```

```
## hscore       0.030081   0.010099   2.979   0.0029 **
## chcond1      0.114085   0.066640   1.712   0.0869 .
## chcond2      0.141158   0.083145   1.698   0.0896 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 5634.8  on 5189  degrees of freedom
## Residual deviance: 4379.5  on 5177  degrees of freedom
## AIC: 6737.1
##
## Number of Fisher Scoring iterations: 6
```

```
residualPlot(AUSHealth)
```



We found that the residual deviance is 4379, while the null deviance is 5634. So generally this model is good. But the residual plot show that this model fits not very well.

Now let's check the overdispersion. We use quasipoission function to deal with it:

```
AUSHealth1 <- glm(doctorco ~ sex + age + agesq + income + levyplus + freepoor + freerepa + illness + ac
summary(AUSHealth1)
```

```
##
## Call:
## glm(formula = doctorco ~ sex + age + agesq + income + levyplus +
##     freepoor + freerepa + illness + actdays + hscore + chcond1 +
##     chcond2, family = quasipoisson, data = dvisits)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9170  -0.6862  -0.5743  -0.4839   5.7005
##
## Coefficients:
```

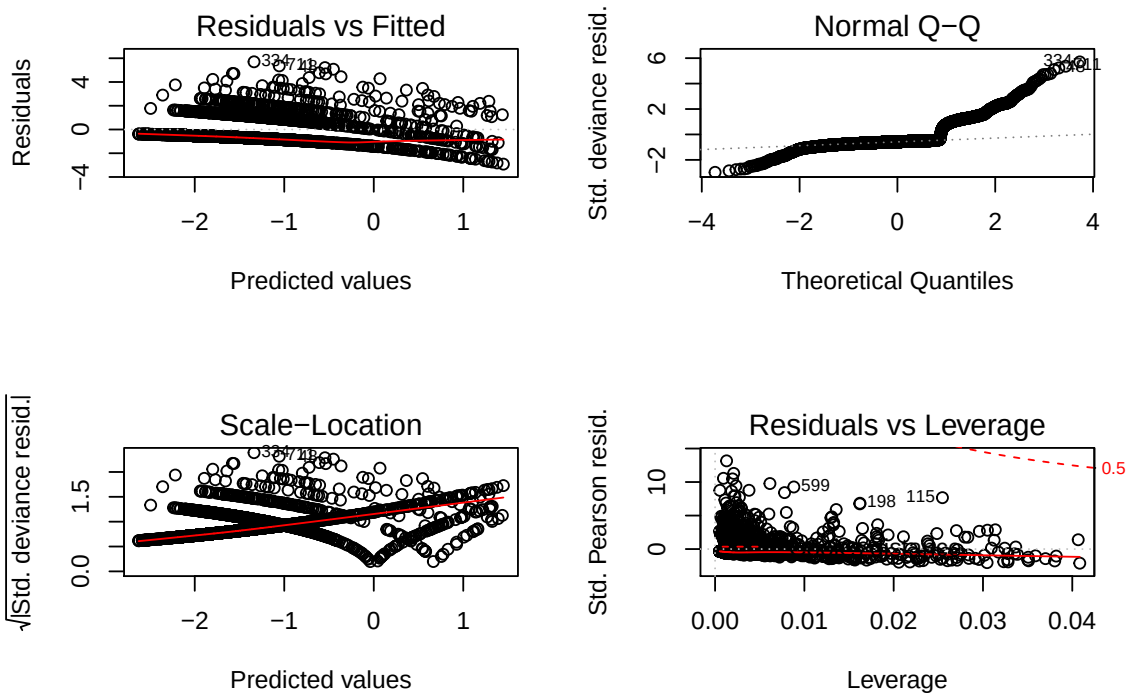
```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.223848   0.218725 -10.167  < 2e-16 ***
## sex          0.156882   0.064686   2.425  0.01533 *
## age          1.056299   1.153198   0.916  0.35972
## agesq       -0.848704   1.241930  -0.683  0.49440
## income      -0.205321   0.101839  -2.016  0.04384 *
## levyplus     0.123185   0.082551   1.492  0.13570
## freepoor    -0.440061   0.207197  -2.124  0.03373 *
## freerepa     0.079798   0.106081   0.752  0.45194
## illness      0.186948   0.021065   8.875  < 2e-16 ***
## actdays      0.126846   0.005801  21.868  < 2e-16 ***
## hscore       0.030081   0.011637   2.585  0.00977 **
## chcond1      0.114085   0.076789   1.486  0.13742
## chcond2      0.141158   0.095808   1.473  0.14072
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.327793)
##
##     Null deviance: 5634.8  on 5189  degrees of freedom
## Residual deviance: 4379.5  on 5177  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

The dispersion parameter for quasipoisson family taken to be 1.327793. Thus, the basic correction for overdispersion is to multiply all regression std by $\sqrt{1.327793} = 1.15$. So it's reasonable to say the formal model is not seriously affected by overdispersion.

Overall, we would say that our model doesn't suffer from overdispersion that much. The model fits not very well.

2. Plot the residuals and the fitted values-why are there lines of observations on the plot?

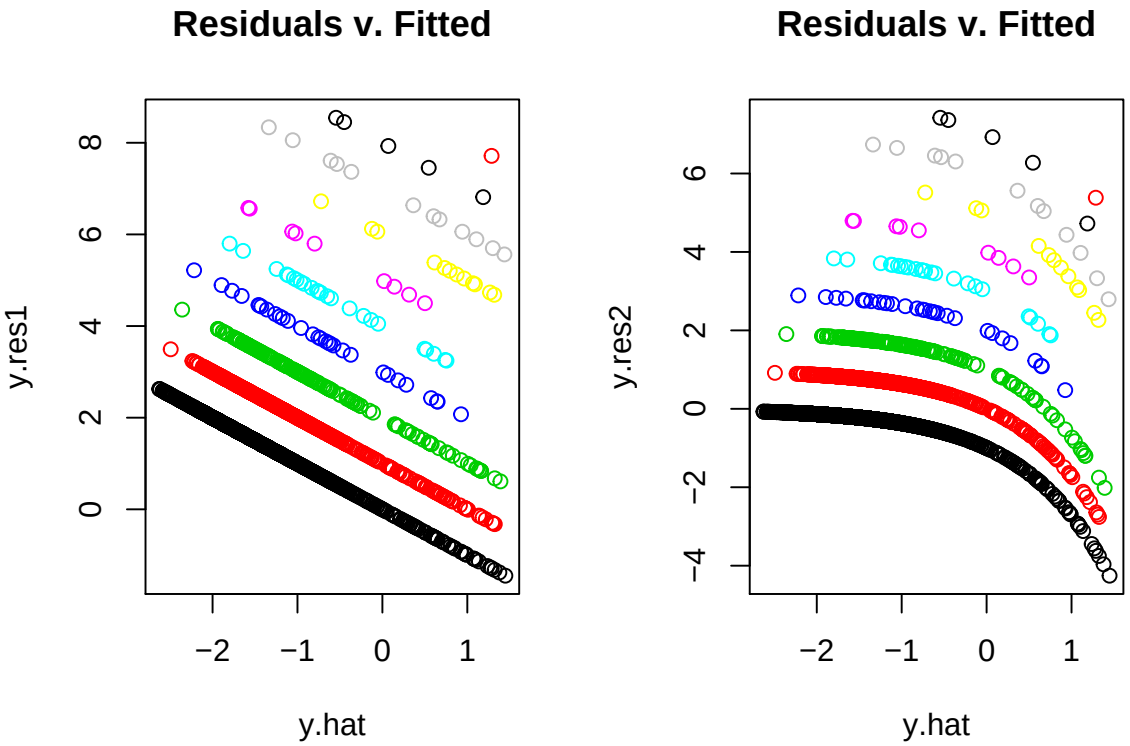
```
par(mfrow=c(2,2))
plot(AUSHealth)
```



Because Poisson model is a count model. Each curvilinear trace of points on the plot corresponds to a fixed value k of the dependent variable y. Every case where y=k has a prediction y_hat. Its residual equals k-y_hat. The plot of k-y_hat versus y_hat is a line with slope of -1. In Poisson regression, the x axis is shown on a log scale, it is log(y_hat). The curves now bend down exponentially.

We can draw the residual line more clearly.

```
b <- coefficients(AUSHealth)

y.hat <- b[1]+b[2]*dvisits$sex + b[3]*dvisits$age + b[4]*dvisits$agesq + b[5]*dvisits$income + b[6]*dvis

y.res1 <- dvisits$doctorco - y.hat
y.res2 <- dvisits$doctorco-exp(y.hat)# Residuals
colors <- 1:(max(dvisits$doctorco)+1)
par(mfrow=c(1,2))
plot(y.hat, y.res1, col=colors[dvisits$doctorco+1], main="Residuals v. Fitted")
plot(y.hat, y.res2, col=colors[dvisits$doctorco+1], main="Residuals v. Fitted")
```



After swtiching the x-axis from log(y) to y, we can see that each line is a straight line.

3. What sort of person would be predicted to visit the doctor the most under your selected model?

For this question, we need to find out which predictors are sigfinicat in our model. We find that sex, imcome, freepoor, illness, actdays and hscore are statistcally significant. Notice that the coefficients of freepoor and income are negative, so in this question, female with less income, and not covered by the government and with more illness and higher number of days of reduced activity due to illness are predicted to visit the doctor the most.

4. For the last person in the dataset, compute the predicted probability distribution for their visits to the doctor, i.e., give the probability they visit 0,1,2, etc. times.

```
predict(AUSHealth, dvisits[5190,], type="response")
```

```
##      5190
## 0.1533837
```

So let $\lambda = 0.1533837$:

```
#Probabiity they visit 0 time
p0 <- dpois(0,lambda = 0.1533837)
p0
```

```
## [1] 0.8578005
```

```
#Probabiity they visit 1 time
p1 <- dpois(1,lambda = 0.1533837)
p1
```

```
## [1] 0.1315726
```

```
#Probabiity they visit 2 time
p2 <- dpois(2,lambda = 0.1533837)
p2
```

```
## [1] 0.01009055
```

```
#Probabiity they visit 3 time
p3 <- dpois(3,lambda = 0.1533837)
p3
```

```
## [1] 0.0005159085
```

```
#Probabiity they visit 3 time
p4 <- dpois(4,lambda = 0.1533837)
p4
```

```
## [1] 1.978299e-05
```

```
p0+p1+p2+p3+p4
```

```
## [1] 0.9999994
```

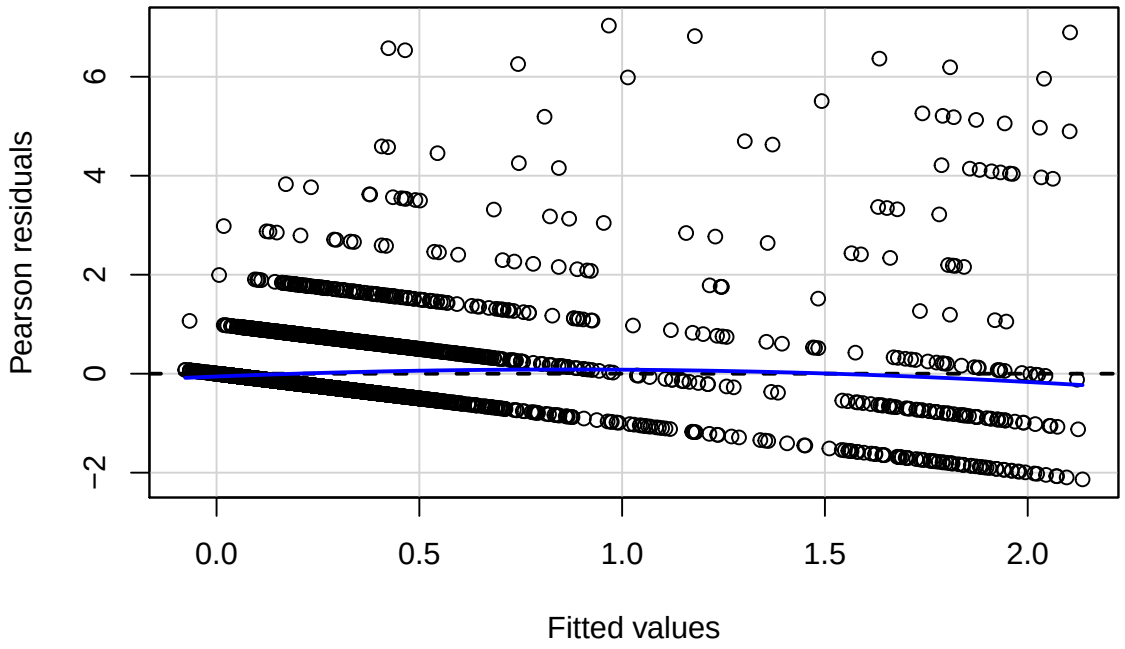
Since $p0 + p1 + p2 + p3 + p4 = 0.9999994$, there is no need to calculate the probability they visit 5 or more times.

5. Fit a comparable (Gaussian) linear model and graphically compare the fits. Describe how they differ.

```
AUSlm <- lm(doctorco ~ sex + age + agesq + income + levyplus + freepoor + freerepa + illness + actdays +
summary(AUSlm)
```

```
##
## Call:
## lm(formula = doctorco ~ sex + age + agesq + income + levyplus +
##     freepoor + freerepa + illness + actdays + hscore + chcond1 +
##     chcond2, data = dvisits)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1352 -0.2588 -0.1435 -0.0433  7.0327
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.027632   0.072220   0.383  0.70202
## sex          0.033811   0.021604   1.565  0.11764
## age          0.203201   0.410016   0.496  0.62020
## agesq       -0.062103   0.458716  -0.135  0.89231
## income      -0.057323   0.033089  -1.732  0.08326 .
## levyplus     0.035179   0.024882   1.414  0.15748
## freepoor    -0.103314   0.052471  -1.969  0.04901 *
## freerepa     0.033241   0.038157   0.871  0.38371
## illness      0.059946   0.008357   7.173 8.39e-13 ***
## actdays      0.103192   0.003657  28.216  < 2e-16 ***
## hscore       0.016976   0.005190   3.271  0.00108 **
## chcond1      0.004384   0.023740   0.185  0.85349
## chcond2      0.041617   0.035863   1.160  0.24592
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7139 on 5177 degrees of freedom
## Multiple R-squared:  0.2018, Adjusted R-squared:    0.2
## F-statistic: 109.1 on 12 and 5177 DF,  p-value: < 2.2e-16
```

```
residualPlot(AUSlm)
```



It seems that the linear model also fits not so well. The adjusted R-Squared is only 0.2. And the residual plot shows that the residual and fitted value is similar with the Poisson model.

Now let's look at the predcited λ

```
predict(AUSlm,dvisits[5190,])
```

```
##      5190
## 0.1606531
```

Generally, this two models are similar. The link function is the difference.