Statistical Modelling 2 Coursework

Hermine Tranié

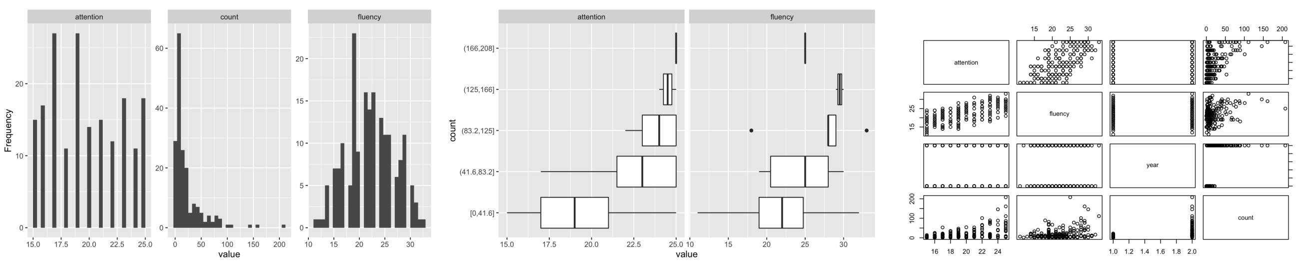
CID: 01400919

1 Introduction

In this report, our goal is to find the best model (relative to the other ones tested) that fits the data collected from a psychological experiment into reading accuracy in children. We will also look at the relationship between the response variable corresponding to the reading accuracy and its predictors: attention span, verbal fluency and year group of the children. We will look at the following models, including their assumptions, explanations and analysis:

- the **initial model** suggested by the team: an additive linear model with 2 predictors: attention span and verbal fluency
- the **alternative model** suggested by the team: a Poisson GLM with log link and 2 predictors: attention span and verbal fluency
- **my proposed model**: a Negative Binomial GLM with log link and 2 predictors: attention span and year group (and some interaction term determined by stepsearch)

2 Exploratory Data Analysis



(a) Histogram (b) Boxplot (c) Pairplot

Figure 1: Exploratory Data Analysis

In order to produce an EDA, we plot the following:

- **Histogram**: to study the distribution of the variables. We see that verbal fluency seems normally distributed, count has a decreasing pattern, while attention span doesn't have any particular behavior.
- **Boxplot**: to study the distribution of the predictors with respect to the response variable. We see that fluency variable exhibits outliers in the interval $(83.2, 125]$ for count and attention span seems to be increasing as the number of words correctly pronounced increases.
- **Pairplot**: to study pairwise relationships in the dataset. We see that (1) year and count shows that in the first year group the count of words correctly pronounced is much lower than in the second year group and that (2) attention and fluency seem to be positively correlated which indicates that these variables provide similar information

Overall, this might give us some hints as to what to include as predictors for later models.

3 Fit initial model

Now, let's conduct an initial fit using an additive linear model as follows: count = attention span + verbal fluency. We have that count is our response variable y and attention span + verbal fluency as our predictors x_i.

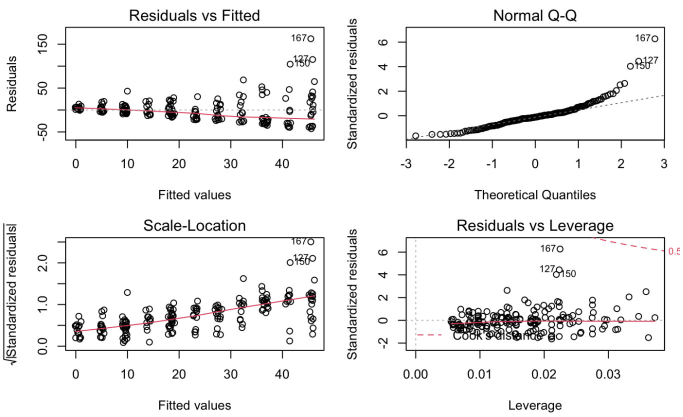
3.1 Assumptions

In order to conduct a linear regression we make the following 4 assumptions: (1) linear relationship between the response variable and each of the predictors, (2) independance of residuals, (3) homoscedasticity (residuals have constant variance at entry level of x). We can detect heteroscedasticity with a fitted values v. residuals plot that has a conic shape for example, if that's the case, use weighted regression instead. And (4) residuals are normally distributed which we can check with the QQ-plot.

3.2 Plots

```
Residuals:
    Min      1Q  Median      3Q     Max
-42.580 -13.772  -3.037   7.155 162.508

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -67.65775   12.52084  -5.404 2.03e-07 ***
attention     4.43896    0.79719   5.568 9.11e-08 ***
fluency       0.08705    0.54127   0.161    0.872
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.22 on 182 degrees of freedom
Multiple R-squared:  0.2273,	Adjusted R-squared:  0.2189
F-statistic: 26.78 on 2 and 182 DF,  p-value: 6.404e-11
```

(a) Model summary



(b) Model plots

Figure 2: Linear model information

3.3 Interpretation

Looking at the summary of the model:

- **Residuals** aren't symmetrically distributed around the mean zero, which means that our model predicts points that are far from the actual ones
- **Coefficients** show that the fluency predictor has a high probability (87%) of getting any value larger or equal than t (= 0.161), while the attention exhibits a very low t value which we can use to reject the null hypothesis, hence there is relationship between count and attention.
- $\mathbf{R^2}$ (or coefficient of determination) has a significantly weak value which indicates that this model isn't the best fit
- **F-statistic** indicates is significantly larger than 1, so we can reject the null hypothesis that there is no relationship between our response variable and our predictors

Looking at the plots of the model:

- **Residuals vs Fitted** plot means that residuals don't have constant variance at entry level of x and hence we might want to have a look at a weighted regression
- **Normal Q-Q** plot indicates that there is a number of points that don't fit correctly
- **Scale-Location** plot shows that residuals are spread out quite evenly but not randomly which emphasizes the heteroscedasticity
- **Residuals vs Leverage** plot shows the 3 points with highest Cook's distance are 127, 150 and 167, they have high residuals but a not so high leverage. So one could try to remove these said outliers to see if it improves the fit (which I have done and it slightly improves the AIC but not the R^2).

In conclusion, this model isn't good because (1) it exhibits heteroscedasticity, (2) the predictors fluency and attention are correlated from the EDA, (3) normal assumption for residuals is violated. Therefore, let's try alternatives.

4 Fit alternative Poisson model

Now, let's conduct an initial fit using a Poisson GLM weighted regression with log link as follows: count = attention span + verbal fluency. We have that count is our response variable y and attention span + verbal fluency as our predictors x_i.

4.1 Assumptions

In order to conduct a Poisson regression we make the following 4 assumptions: (1) response variable follows a Poisson distribution: it is a count, (2) independence of observations, (3) the mean is equal to the variance, and (4) linearity of log of mean rate.

4.2 Explanation

We are going to fit a Poisson GLM with a log link using the IWLS algorithm. First let's write the Poisson distribution in exponential family form:

Suppose you have a sample of $Y_1, ..., Y_n$ where $Y_i \sim Poisson(\lambda_i)$, $f(y_i, \lambda_i) = \frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!} = \frac{1}{y_i}\exp\left(y_i \log(\lambda_i) - \lambda_i\right)$. Then, $\mu_i = E(Y_i) = \lambda_i$, $\theta_i = \log(\lambda_i)$, $a(\phi) = 1$, $b(\theta_i) = \lambda_i = \exp\theta_i$, $b'(\theta_i) = \lambda_i = \exp\theta_i$, $b''(\theta_i) = \lambda_i = \exp\theta_i$, $V(\mu_i) = b''(\theta_i) = \lambda_i = \exp\theta_i = \mu_i$. Using the log link, $\eta_i = \log(\mu_i) \implies \mu_i = \exp\theta_i$, $\frac{\partial\eta_i}{\partial\mu_i} = \frac{1}{\mu_i}$, $z_i = \hat{\eta_i} + (y_i - \hat{\mu_i})\frac{\partial\eta}{\partial\mu}\Big|_{\mu=\hat{\mu_i}} = \hat{\eta_i} + \frac{(y_i - \hat{\mu_i})}{\hat{\mu_i}}$, $w_{ii}^{-1} = \left(\frac{\partial\eta}{\partial\mu}\right)^2 V(\mu)\Big|_{\mu=\hat{\mu_i}} = \frac{1}{\hat{\mu_i}}$.

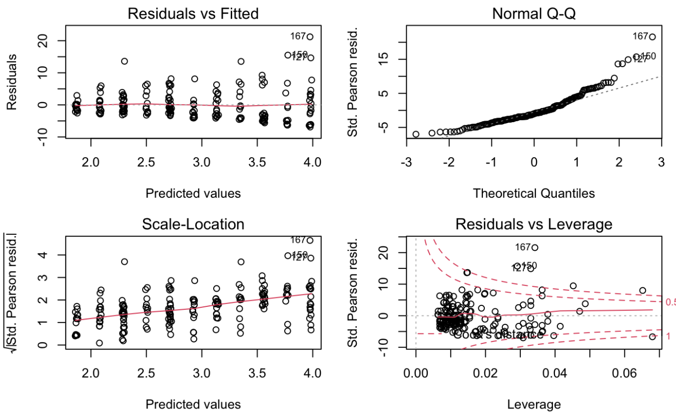
With these values, we can now implement the IWLS algorithm motivated by $\hat{\beta} = (X^TWX)^{-1}X^TWz$ as follows:
1. Form a sensible estimate of β and compute the associated linear predictor $\hat{\eta}$ and fitted values $\hat{\mu}$. To find a sensible estimate, we fit a standard linear model on predictors as in the model before.
2. Form the adjusted dependant variable z_i, as computed above.
3. Form the estimated weights $\tilde{w}_{ii}^{-1}$, as computed above.
4. Regress z_i on x_i with weights $\tilde{w}_{ii}^{-1}$ and obtain the new estimate $\hat{\beta}$.
5. Repeat steps 1 to 4 until convergence. For this algorithm, we use the deviance D for a convergence criterion such that $\frac{|D_{new} - D_{old}|}{|D_{new}| + 0.1} < 1.10^{-8}$.

Now compute important results that we can infer from this model:
- the deviance : $D = 2\phi(l(\hat{\beta}_{saturated}) - l(\hat{\beta})) = 2\sum_{i=1}^{n}\left(y_i \log(\frac{y_i}{\hat{\mu_i}}) - (y_i - \hat{\mu_i})\right)$
- the dispersion parameter, using Pearson's statistic: $\hat{\phi}_p = \frac{X^2}{n-p}$, where X^2 is the Pearson's statistic
- the standard errors: $diag(cov(\hat{\beta})) = diag(\hat{\phi}_p(X^T\tilde{W}X)^{-1}$
- the AIC: $AIC = -2l(\hat{\beta}) + 2p$

4.3 Plots

```
Deviance Residuals:
   Min       1Q   Median       3Q      Max
-9.135   -3.332   -1.042    1.792   16.040

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.298195   0.118969 -10.912   <2e-16 ***
attention    0.209024   0.006830  30.604   <2e-16 ***
fluency      0.001907   0.004392   0.434    0.664
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 5013.2  on 184  degrees of freedom
Residual deviance: 3335.4  on 182  degrees of freedom
AIC: 4124.5

Number of Fisher Scoring iterations: 5
```

(a) Model summary



(b) Model plots

Figure 3: Poisson model information

4.4 Interpretation

Looking at the summary of the model:

- **Residuals** aren't symmetrically distributed around the mean zero, which means that our model predicts points that are far from the actual ones, but it's doing better than the linear model
- **Coefficients** show again that the fluency predictor has a high probability (66%) of getting any value larger or equal than z (= 0.434), while the attention exhibits a very low z value which we can use to reject the null hypothesis, hence there is relationship between count and attention.
- **Deviance** (null/residual) for a well-fitting model, our residual deviance should be close to the degrees of freedom, which isn't the case here, so we are still not looking at the best model

- **AIC** gives a tradeoff between 2 models. It is interesting to compare with another from the same exponential family. On its own, it is not relevant. We can try to compare this Poisson model with another with the year group as predictor and without the fluency as it gives terrible z values. Indeed when doing that, we get an AIC decrease from 4124.5 to 2309. Also by simply removing potential outliers, we can reduce the AIC from 4124.5 to 3481.5.
- **Fisher Scoring iterations** is relatively low, our model is converging, this is satisfying.

Looking at the plots of the model:

- **Residuals vs Fitted** residuals are quite large and it seems that predicted values are still quite packed.
- **Normal Q-Q** plot indicates that there is a number of points that don't fit correctly
- **Scale-Location** plot shows that residuals are spread out quite evenly
- **Residuals vs Leverage** plot shows the 3 points with highest Cook's distance are 127, 150 and 167, they have high residuals but a not so high leverage. So one could try to remove these said outliers to see if it improves the fit (which I have done and it slightly improves the AIC).

Finally our estimate for the dispersion is actually 19, compared to the assumed 1 taken by our model. This means that our data obtained is a bit more variable than expected and all of it can't be captured by the model.

In conclusion, this model isn't good because (1) residuals are quite large, (2) the predictors fluency and attention are correlated from the EDA and (3) there is significant variability that isn't captured by the model. Therefore, let's try alternatives.

5 Fit own model

5.1 Improve Poisson

After trying out step search on Poisson model, I realized that removing the fluency predictor and adding the year group in did improve the fit. But actually a more complex model improves significantly all our diagnostics as follows (which reduces AIC to 2184.5) : count = attention + fluency + year + attention:year + attention:fluency + fluency:year. By inspecting its diagnostic plots and outliers and influential observations, we have managed to improve this model significantly (see code appendix).

However, I tried implementing different exponential families to see if there was anymore improvement possible. Therefore, I will consider the Negative Binomial (as it helps with over-dispersed data) exponential family as follows (as it is a counting process) and perform a weighted regression with log link as follows (after having conducted step search) : count = attention + year + attention:year + fluency:year. We have that count is our response variable y and attention span + year group as our predictors x_i.

Step <S3: AsIs>	Df <dbl>	Deviance <dbl>	Resid. Df <dbl>	Resid. Dev <dbl>	AIC <dbl>
	NA	NA	181	199.7430	1312.993
+ attention:year	-1	0.9079555	180	198.8351	1296.023
+ fluency:year	-1	0.8509229	179	197.9841	1293.552
+ attention:fluency	-1	0.6095962	178	198.5937	1292.906

Figure 4: Steph Search on negative binomial model

5.2 Assumptions on new model: Negative Binomial

In order to conduct a Negative Binomial regression we make the following 4 assumptions: (1) response variable follows a Negative Binomial distribution: it is a count, (2) independence of observations, (3) the mean is not equal to the variance (unlike Poisson); it allows variance to be greater than the mean which helps with fitting, and (4) linearity in model parameters.

5.3 Explanation

We are going to fit a Negative Binomial GLM with a log link using the IWLS algorithm. First let's write the Negative Binomial distribution in exponential family form:

Suppose you have a sample of $Y_1, ..., Y_n$ where $Y_i \sim NegBin(3, p_i)$, p is the fail rate, 3 is the number of words falsely pronounced $f(y_i, p_i) = \binom{3+y_i-1}{y_i}(p_i)^3(1-p_i)^{y_i} \propto \exp\left(3\log(p_i) + y_i\log(1-p_i)\right) = \exp\left(y_i\log(1-p_i) - (-3\log(p_i))\right)$.

Then, $\mu_i = E(Y_i) = \frac{3(1-p_i)}{p_i} \implies p_i = \frac{3}{\mu_i+3}$, $\theta_i = \log(1-p_i)$, $a(\phi) = 1$, $b(\theta_i) = -3\log(p_i) = -3\log(1-\exp\theta_i)$, $b^{'}(\theta_i) = \frac{3e^{\theta_i}}{1-e^{\theta_i}} = \frac{3}{e^{-\theta_i}-1} = \frac{3(1-p_i)}{p_i}$, $b^{''}(\theta_i) = \frac{3e^{\theta_i}}{(1-e^{\theta_i})^2} = \frac{3(1-p_i)}{p_i^2}$, $V(\mu_i) = b^{''}(\theta_i) = \frac{\mu_i}{p_i} = \frac{\mu_i(\mu_i+3)}{3}$. Using the log link,

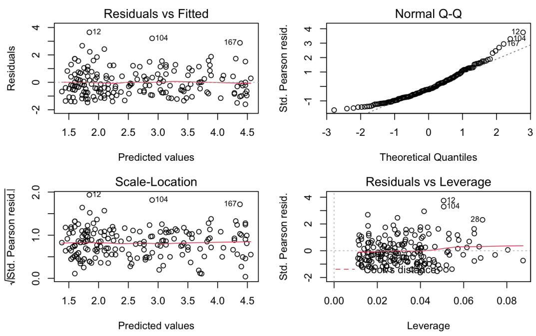
$\eta_i = \log(\mu_i) \implies \mu_i = \exp\eta_i$, $\frac{\partial\eta_i}{\partial\mu_i} = \frac{1}{\mu_i}$, $z_i = \hat{\eta_i} + (y_i - \hat{\mu_i})\frac{\partial\eta}{\partial\mu}\Big|_{\mu=\hat{\mu_i}} = \hat{\eta_i} + \frac{(y_i-\hat{\mu_i})}{\hat{\mu_i}}$, $w_{ii}^{-1} = \left(\frac{\partial\eta}{\partial\mu}\right)^2 V(\mu)\Big|_{\mu=\hat{\mu_i}} = \frac{\hat{\mu_i}+3}{3\hat{\mu_i}}$.

Let's define the likelihood L and loglikelihood l in order to compute our new deviance: $L(\beta) = \prod_{i=1}^n \binom{(3+y_i-1)}{y_i} p_i^3(1-p_i)^{y_i} = \prod_{i=1}^n \binom{(3+y_i-1)}{y_i}\left(\frac{3}{\mu_i+3}\right)^3\left(\frac{\mu_i}{\mu_i+3}\right)^{y_i}$ and $l(\beta) = \log(L(\beta) \propto \sum_{i=1}^n (3\log(3) + y_i\log(\mu_i) - 3\log(\mu_i+3) - y_i\log(\mu_i+3)) = 3n\log(3) + \sum_{i=1}^n(y_i\log(\mu_i)) - \sum_{i=1}^n(3+y_i)\log(\mu_i+3)$. Now the deviance is:

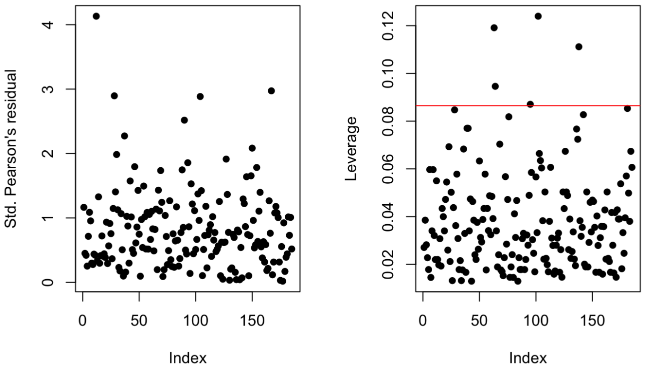
$$D = 2\sum_{i=1}^n\left(y_i\log\left(\frac{y_i}{\hat{\mu}_i}\right) + (3+y_i)\log\left(\frac{\hat{\mu}_i+3}{y_i+3}\right)\right)$$

Now let's implement the same algorithm as described in part 4, with the updated values for the weights and deviance.

5.4 Plots



(a) Model plots



(b) Outlier detection

Figure 5: Negative Binomial model information

5.5 Interpretation

By implementing step search, we have optimized our AIC to 1295.6 (see code appendix for summary of this model), and our residual deviance is now very close to the degrees of freedom of the model which indicates good fit. Before doing step-search and only using the model of count=attention+year, we had an AIC of 1313, so actually the step search isn't actually making such a fascinating improvement. Usually we have to be careful with step search with many predictors as it can induce some overfitting, but here we are using 3 main predictors and an intercept. The deviance residuals are evenly distributed and centered around zero. Looking at the plots, we can observe good randomness of predicted values for the residuals and a good qq-plot fit. We have then looked at potential outliers using Pearson's residuals but there was no overlap between the 5 largest standardised residuals and the 5 largest leverages (as seen in code). Finally we have the following 95% confidence interval for our estimated parameters: $\beta_1 = 0.09320$ and $\beta_2 = -1.49679$: $CI_{\beta_1,95\%} = (0.0348744420.152452048)$ and $CI_{\beta_2,95\%} = (-2.737011768 - 0.256071539)$, which seems reasonable. Finally note that in the in-built function, the dispersion parameter is taken to be 1, however, by using our own coding, we can estimate this dispersion to be around 141 for this model. Our final estimated deviance before convergence (207) matches the one we get from the in-built function (198). Finally our estimate for the dispersion is 1.14 (compared to assumed 1 from inbuilt function) which means that our model now captures all the variability much better than the Poisson model. Note that this has been calculated as follows as mentioned in 4.2, $X^2 = \sum_{i=1}^n \frac{(y_i-\hat{\mu_i})^2}{V(\hat{\mu_i})}$.

6 Conclusion and limitations

To conclude, we have managed to find a relatively good fit with a Negative Binomial weighted regression with log link. It is relative to the results of the other models that have been tried out in this analysis, but it most probably isn't the best possible one. It has proven that this model is the best out of the other ones tested based on dispersion, deviance, residuals and other diagnostics (optimized Poisson, and linear regression). However there are notable limitations in general, first of all, we don't know how the data was collected, and we don't have a very large data set which can skew assumptions of taking number of observations to infinity for example.

Statistical Modelling 2

```r
load("01400919.RData")
```

1. Exploratory Data Analysis

```r
library(knitr)
dat <- data.frame(read$attention, read$fluency, read$yr, read$count)
str(dat)
```

```
## 'data.frame':    185 obs. of  4 variables:
##  $ read.attention: int  17 15 23 20 17 24 19 17 25 23 ...
##  $ read.fluency  : int  23 17 24 26 20 29 23 24 24 23 ...
##  $ read.yr       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ read.count    : int  1 3 6 5 8 2 10 4 8 11 ...
```

```r
colnames(dat)[1] <- "attention"
colnames(dat)[2] <- "fluency"
colnames(dat)[3] <- "year"
colnames(dat)[4] <- "count"
#dat$year <- as.integer(dat$year)
kable(head(dat))
```

attention	fluency	year	count
17	23	0	1
15	17	0	3
23	24	0	6
20	26	0	5
17	20	0	8
24	29	0	2

```r
summary(dat)
```

```
##    attention        fluency      year        count       
##  Min.   :15.00   Min.   :11.00   0:92   Min.   :  0.00  
##  1st Qu.:17.00   1st Qu.:19.00   1:93   1st Qu.:  5.00  
##  Median :19.00   Median :22.00          Median : 10.00  
##  Mean   :19.75   Mean   :22.15          Mean   : 21.95  
##  3rd Qu.:23.00   3rd Qu.:25.00          3rd Qu.: 25.00  
##  Max.   :25.00   Max.   :33.00          Max.   :208.00  
```

```r
library(skimr)
skim(dat)
```

Table 2: Data summary

Name	dat
Number of rows	185

Table 2: Data summary

Number of columns	4
Column type frequency:	
factor	1
numeric	3
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
year	0	1	FALSE	2	1: 93, 0: 92

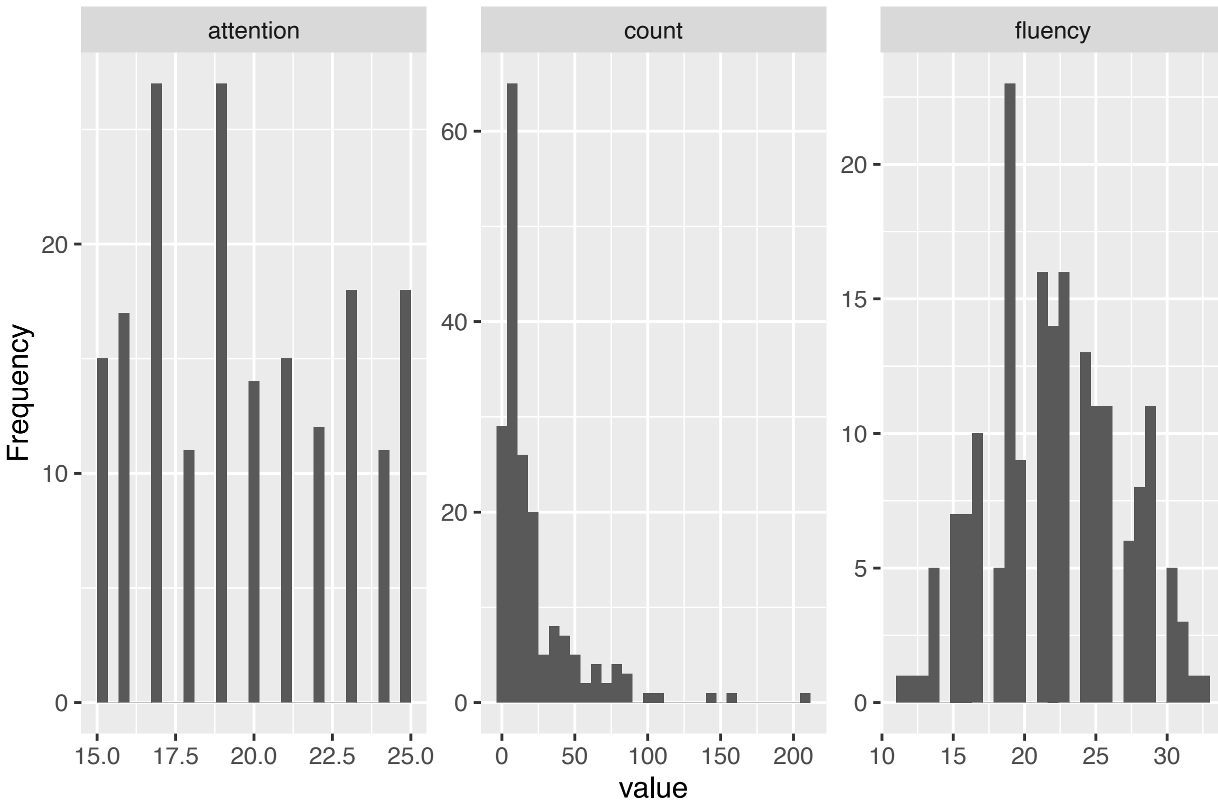
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
attention	0	1	19.75	3.13	15	17	19	23	25	
fluency	0	1	22.15	4.61	11	19	22	25	33	
count	0	1	21.95	29.66	0	5	10	25	208	

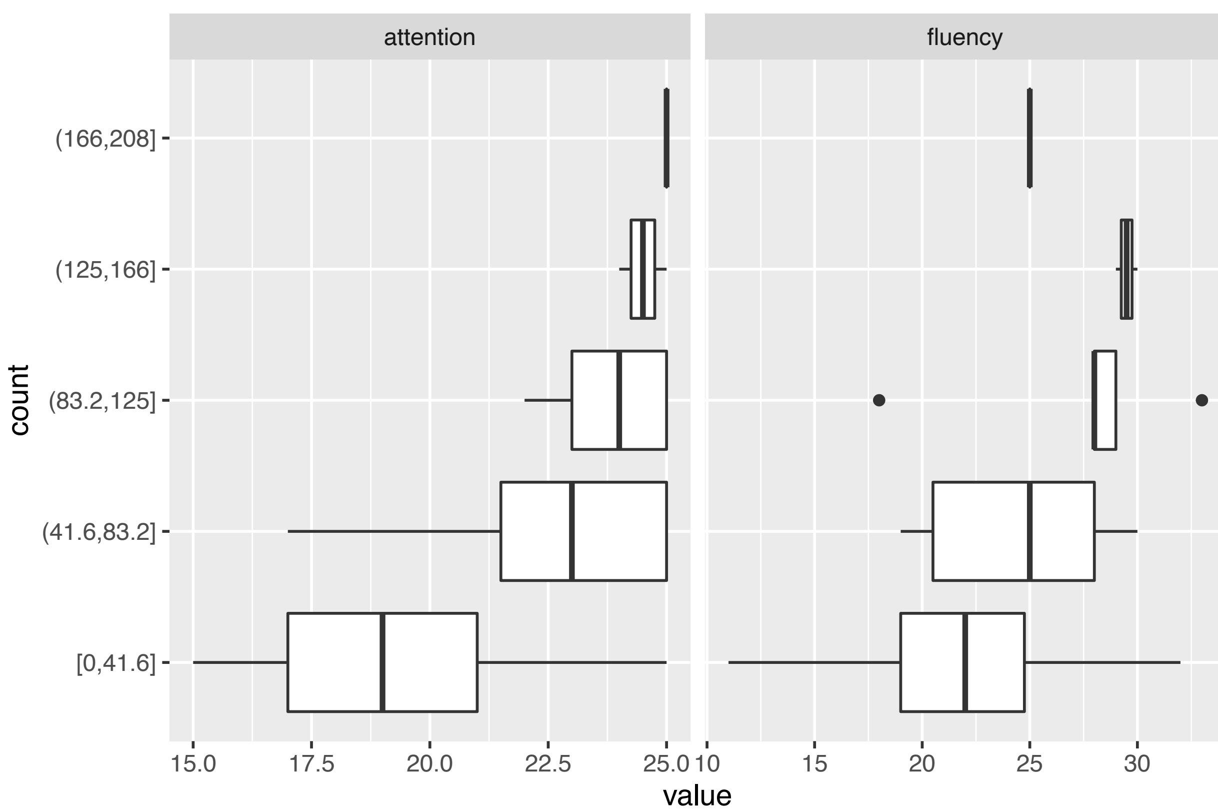
```
with(dat, table(count))
```

```
## count
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
##   4   7   7  11  13   9  10   7  12   6   8   8   5   1   5   1   5   1   1   3
##  20  21  22  23  24  25  27  30  31  32  33  35  36  38  39  40  41  42  43  47
##   5   2   2   2   1   4   2   1   1   1   2   1   1   2   2   1   4   1   1   1
##  48  50  53  54  57  63  66  67  71  72  77  78  82  87  88  89 101 111 146 161
##   1   1   2   1   1   2   1   1   1   1   2   1   1   1   1   1   1   1   1   1
## 208
##   1
```

```
library(DataExplorer)
plot_histogram(dat)
```



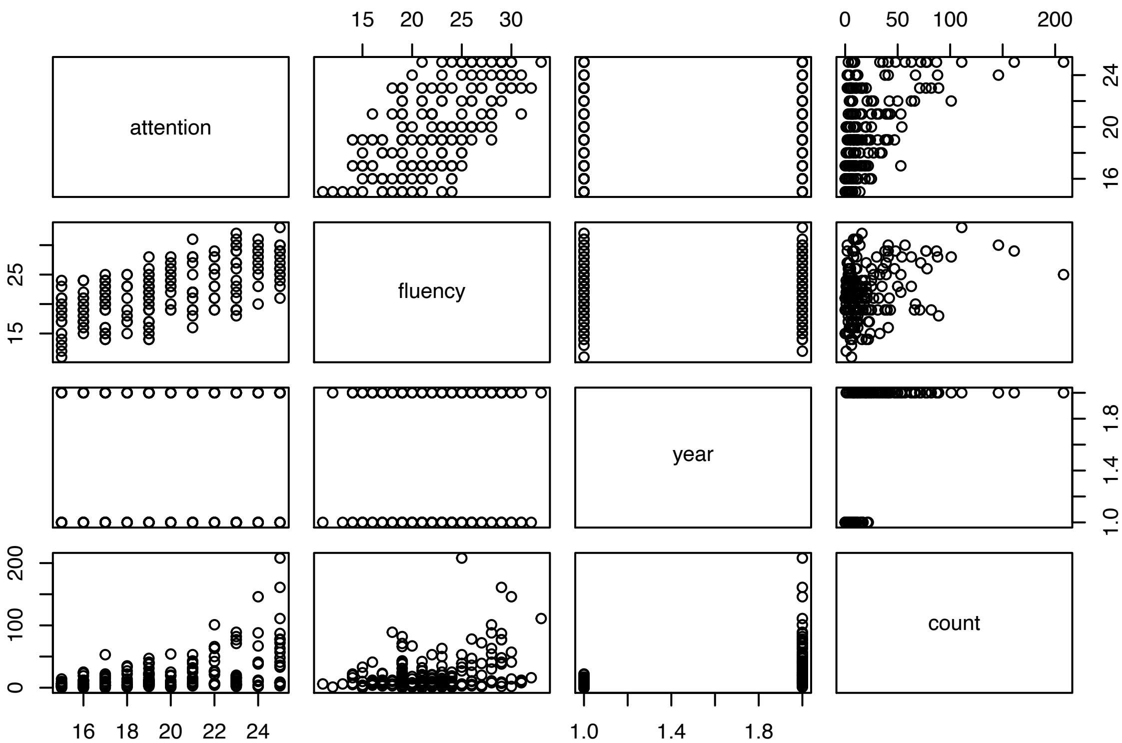

```
plot_boxplot(dat, by= "count")
```



```r
print(dat$year)
```

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## Levels: 0 1
```

```r
pairs(dat,cex.labels=0.95)
```



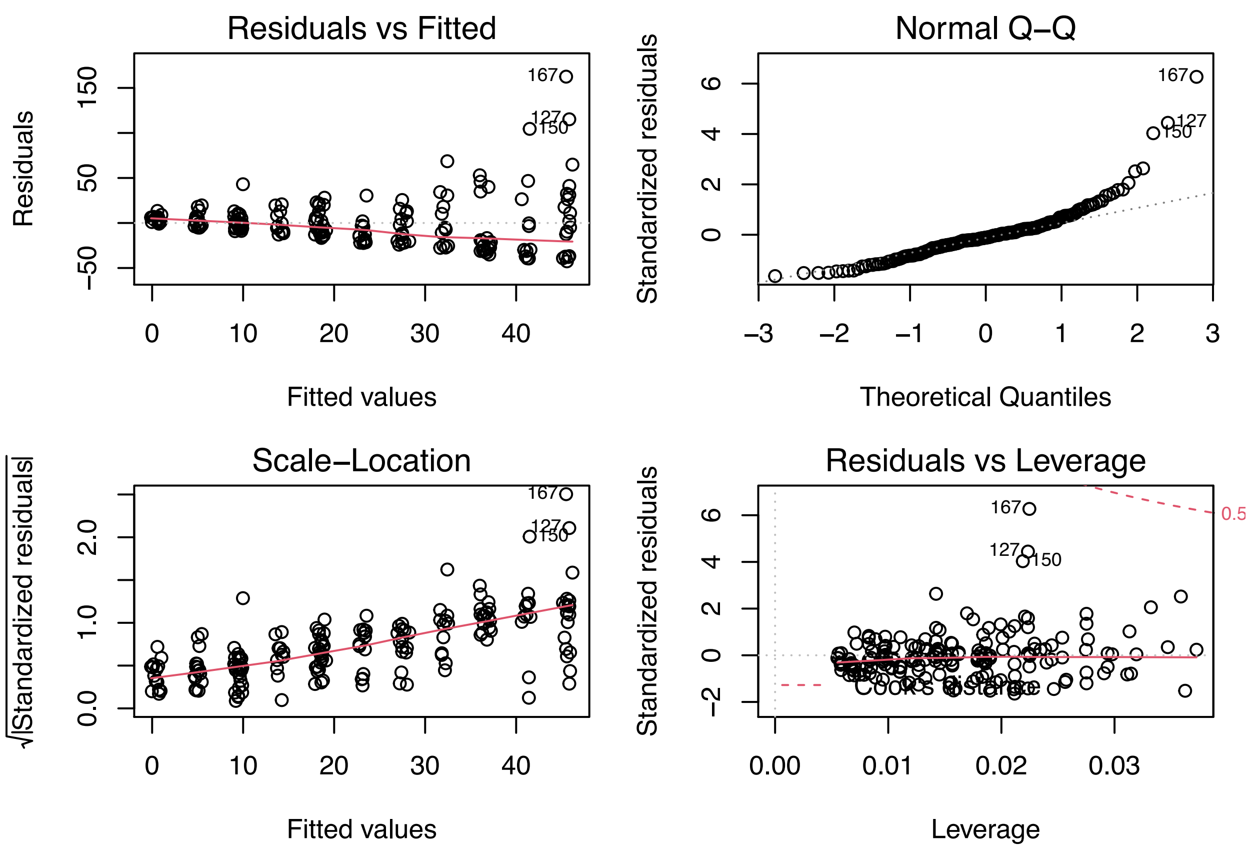
2. Fit initial model suggested by the team (using attention and verbal fluency only)

```r
fit0 <- lm(count~attention+fluency, data=dat)

# plot linear model
summary(fit0)
```

```
##
## Call:
## lm(formula = count ~ attention + fluency, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.580 -13.772  -3.037   7.155 162.508
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -67.65775   12.52084  -5.404 2.03e-07 ***
## attention     4.43896    0.79719   5.568 9.11e-08 ***
```

```
## fluency       0.08705    0.54127   0.161    0.872
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.22 on 182 degrees of freedom
## Multiple R-squared:  0.2273, Adjusted R-squared:  0.2189
## F-statistic: 26.78 on 2 and 182 DF,  p-value: 6.404e-11
```

```r
par(mfrow = c(2, 2), mar = c(4.3, 4.3, 2, 1))
plot(fit0)
```

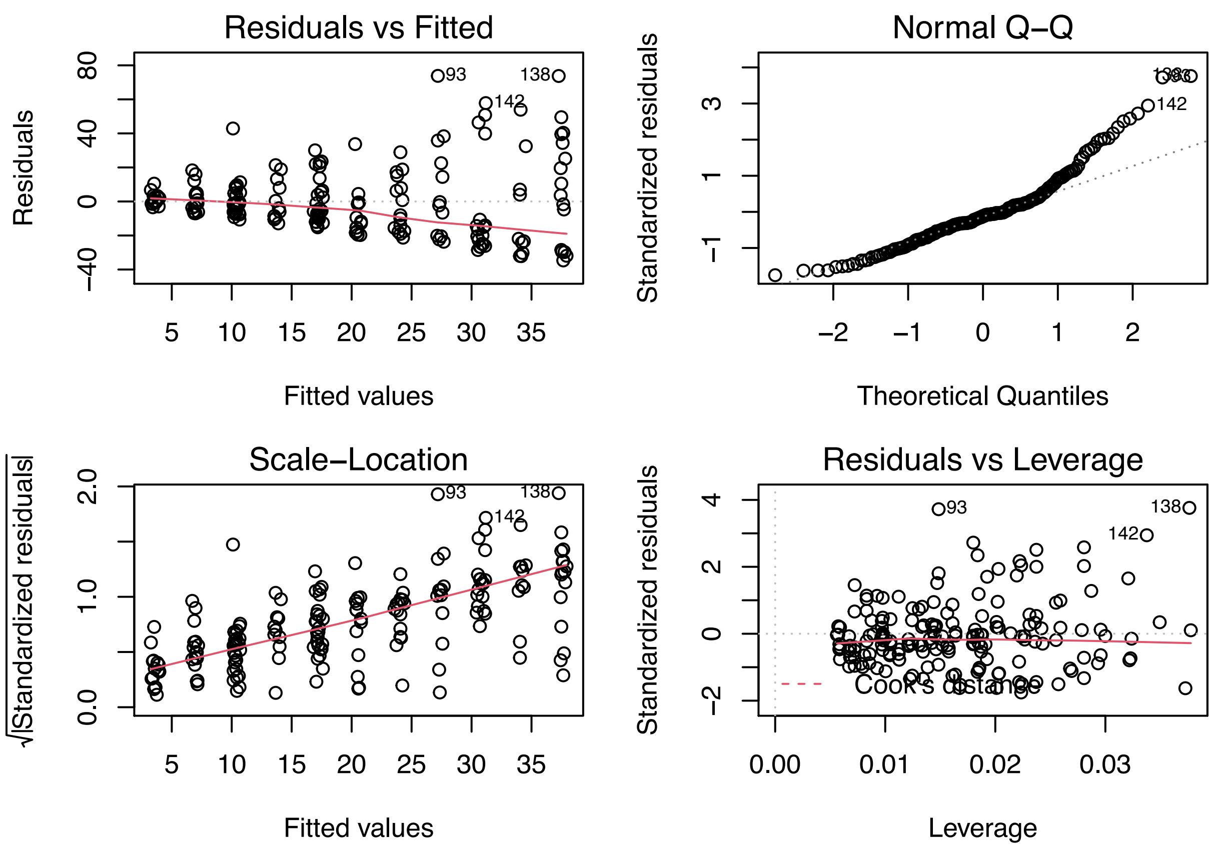



```r
# remove potential outliers and fit model again
dat1 <- dat[c(-127,-150,-167),]
fit01 <- lm(count~attention+fluency, data=dat1)

# summary and diagnostic plots of model without potential outliers
summary(fit01)
```

```
##
## Call:
## lm(formula = count ~ attention + fluency, data = dat1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.650 -12.046  -2.339   6.667  73.808
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -47.18177    9.71350  -4.857 2.59e-06 ***
## attention      3.44986    0.61419   5.617 7.32e-08 ***
## fluency       -0.05441    0.41465  -0.131    0.896
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.98 on 179 degrees of freedom
## Multiple R-squared:  0.2183, Adjusted R-squared:  0.2096
## F-statistic: 24.99 on 2 and 179 DF,  p-value: 2.676e-10
```

```
par(mfrow = c(2, 2), mar = c(4.3, 4.3, 2, 1.2))
plot(fit01)
```




```
AIC(fit0)
```

```
## [1] 1738.529
```

```
AIC(fit01)
```

```
## [1] 1611.591
```

3. Fit alternative Poisson GLM with log link model + evaluate quality of fit

```
# response variable and predictors for Poisson GLM
y <- as.numeric(dat$count)
x1 <- cbind(as.numeric(dat$attention), as.numeric(dat$fluency))
X <- cbind(1,x1)
```

```r
# IWLS
# find initial estimate for beta
fit1 <- lm(y~x1)
beta <- fit1$coefficients

# inverse link function
log.link <- function(u){
  exp(u)
}

# deviance function
D <- function(p){ # p is the estimated mean mu
  a <- y*log(y/p)
  b <- (p-y)
  a[y==0] <- 0
  2*sum(a+b)
}

oldD <- D(log.link(as.numeric(X%*%beta)))
jj <- 0
while(jj==0){
  eta <- X%*%beta # estimated linear predictor
  mu <- log.link(eta) # estimated mean response
  z <- eta + ((y-mu)/mu) # form the adjusted variate
  w <- mu # weights
  lmod <- lm(z~x1, weights=w) # regress z on x with weights w, includes intercept anyway
  beta <- as.vector(lmod$coeff) # newbeta
  newD <- D(log.link(X%*%beta))
  control <- abs(newD-oldD)/(abs(newD)+0.1)
  if(control<1e-8)
    jj <- 1
  oldD <- newD
}
beta # final estimate
```

```
## [1] -1.298195173  0.209023600  0.001907352
```

```r
newD # last deviance calculated
```

```
## [1] 3335.421
```

```r
# Results from IWLS Poisson

# Pearson's statistic
X2 <- 0
for (i in 1:185){
  X2 <- X2 + (y[i]-mu[i])^2/w[i]
}

# dispersion parameter estimate
phi <- X2/(185-3) #n-p, n number of rows, p number of predictors
phi
```

```
## [1] 19.68121
```

```r
# computation of covariance matrix and standard residuals for estimates
J <- t(X)%*%diag(as.vector(w))%*%X
invJ <- solve(J)
cov.beta <- phi*invJ
beta.sd <- sqrt(as.vector(diag(cov.beta)))
beta.sd
```

```
## [1] 0.52778323 0.03030014 0.01948285
```

```r
# computation of deviance residuals
p <- as.vector(log.link(X%*%beta))
a <- y*log(y/p)
b <- (y-p)
a[y==0] <- 0
d <- sign(y-mu)*sqrt(2*(a+b))
```

```
## Warning in sqrt(2 * (a + b)): NaNs produced
```

```r
summary(d)
```

```
##        V1
##  Min.   : 0.2961
##  1st Qu.: 4.2236
##  Median : 7.7757
##  Mean   : 8.6169
##  3rd Qu.:11.3995
##  Max.   :29.6019
##  NA's   :113
```

```r
z <- beta/beta.sd
z # large n makes the student t distribution tend to normal distribution
```

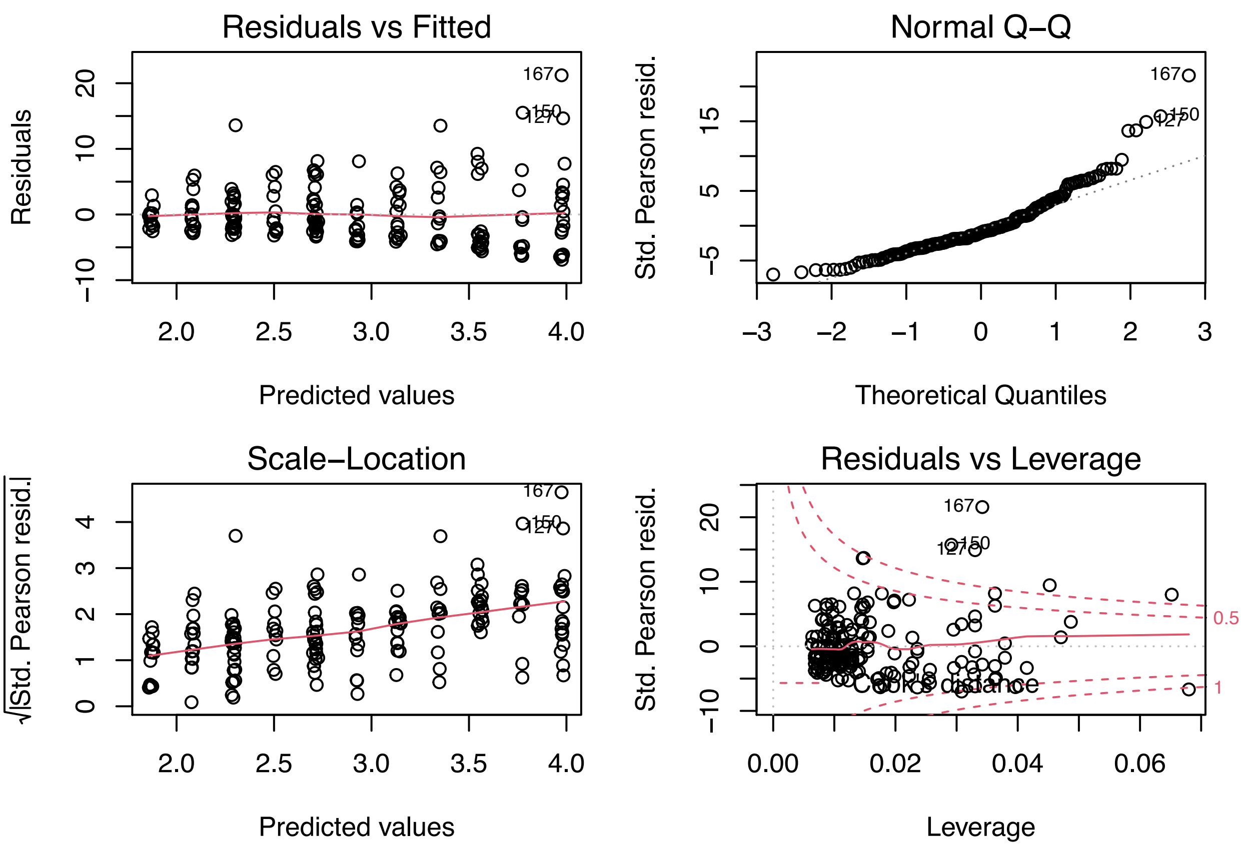
```
## [1] -2.45971282  6.89843771  0.09789899
```

```r
# sanity check
fit10 <- glm(count~attention+fluency, family="poisson", data=dat)
summary(fit10)
```

```
##
## Call:
## glm(formula = count ~ attention + fluency, family = "poisson",
##     data = dat)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -9.135  -3.332  -1.042   1.792  16.040
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.298195   0.118969 -10.912   <2e-16 ***
## attention    0.209024   0.006830  30.604   <2e-16 ***
## fluency      0.001907   0.004392   0.434    0.664
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
```

```
##     Null deviance: 5013.2  on 184  degrees of freedom
## Residual deviance: 3335.4  on 182  degrees of freedom
## AIC: 4124.5
##
## Number of Fisher Scoring iterations: 5
```

```
par(mfrow = c(2, 2), mar = c(4.3, 4.3, 2, 1.2))
plot(fit10)
```




```
confint(fit10)
```

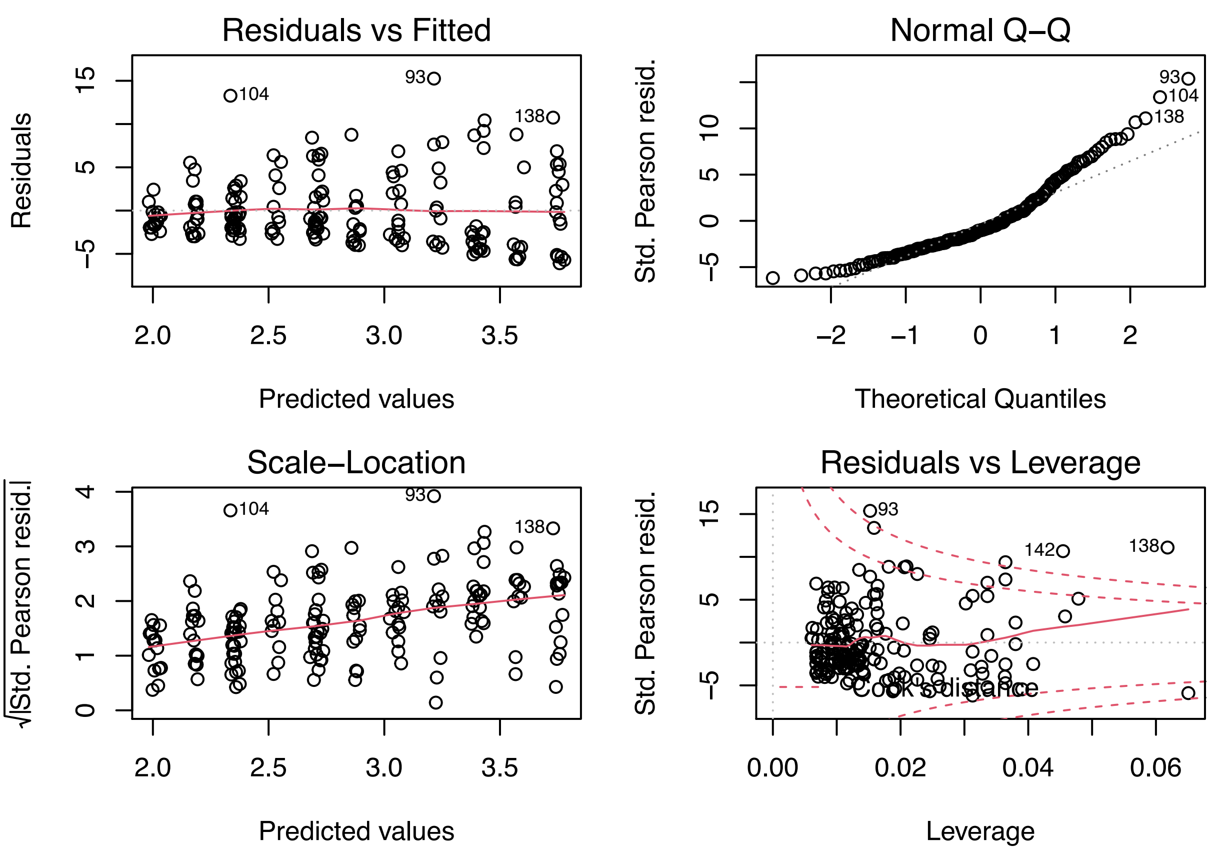
```
## Waiting for profiling to be done...

##                     2.5 %      97.5 %
## (Intercept) -1.532545378 -1.06617199
## attention    0.195658783  0.22243310
## fluency     -0.006690565  0.01052522
```

```
# remove potential outliers
dat2 <- dat[-c(167,127,150),]
fit11 <-  glm(count~attention+fluency, family="poisson", data=dat2)

# model without potential outliers
summary(fit11)
```

```
##
## Call:
## glm(formula = count ~ attention + fluency, family = "poisson",
##     data = dat2)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.985  -3.182  -1.065   1.788  11.431
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.595941   0.122009  -4.884 1.04e-06 ***
## attention    0.178348   0.007125  25.033  < 2e-16 ***
## fluency     -0.004036   0.004649  -0.868    0.385
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3728.4  on 181  degrees of freedom
## Residual deviance: 2713.3  on 179  degrees of freedom
## AIC: 3481.5
##
## Number of Fisher Scoring iterations: 5
```

```
par(mfrow = c(2, 2), mar = c(4.3, 4.3, 2, 1.2))
plot(fit11)
```




```
AIC(fit10)
```

```
## [1] 4124.536
```

```r
AIC(fit11)
```

```
## [1] 3481.466
```

```r
# use step search to compare models
fit12 <- glm(count~attention+fluency+year, family="poisson", data=dat)
stepsearch <- step(fit12,~.^2,test="Chisq")
```

```
## Start:  AIC=2309.11
## count ~ attention + fluency + year
##
##                     Df Deviance    AIC     LRT  Pr(>Chi)
## + attention:year     1   1433.2 2226.3   84.82 < 2.2e-16 ***
## + fluency:year       1   1457.3 2250.4   60.69 6.671e-15 ***
## + attention:fluency  1   1504.0 2297.1   14.00 0.0001828 ***
## - fluency            1   1520.0 2309.1    1.99 0.1585696
## <none>                   1518.0 2309.1
## - attention          1   2126.3 2915.5  608.35 < 2.2e-16 ***
## - year               1   3335.4 4124.5 1817.43 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=2226.3
## count ~ attention + fluency + year + attention:year
##
##                     Df Deviance    AIC    LRT  Pr(>Chi)
## + attention:fluency  1   1405.3 2200.4 27.881  1.29e-07 ***
## + fluency:year       1   1424.5 2219.6  8.724   0.00314 **
## - fluency            1   1433.4 2224.6  0.256   0.61275
## <none>                   1433.2 2226.3
## - attention:year     1   1518.0 2309.1 84.815 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=2200.42
## count ~ attention + fluency + year + attention:year + attention:fluency
##
##                     Df Deviance    AIC    LRT  Pr(>Chi)
## + fluency:year       1   1387.4 2184.5 17.910 2.316e-05 ***
## <none>                   1405.3 2200.4
## - attention:fluency  1   1433.2 2226.3 27.881 1.290e-07 ***
## - attention:year     1   1504.0 2297.1 98.695 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=2184.51
## count ~ attention + fluency + year + attention:year + attention:fluency +
##     fluency:year
##
##                     Df Deviance    AIC    LRT  Pr(>Chi)
## <none>                   1387.4 2184.5
## - fluency:year       1   1405.3 2200.4 17.910 2.316e-05 ***
## - attention:fluency  1   1424.5 2219.6 37.066 1.142e-09 ***
## - attention:year     1   1426.6 2221.8 39.245 3.739e-10 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
stepsearch$anova
```

```
##                  Step Df Deviance Resid. Df Resid. Dev      AIC
## 1                     NA       NA       181   1517.996 2309.111
## 2    + attention:year -1 84.81528       180   1433.181 2226.296
## 3 + attention:fluency -1 27.88051       179   1405.300 2200.416
## 4      + fluency:year -1 17.90965       178   1387.390 2184.506
```

```r
summary(stepsearch)
```

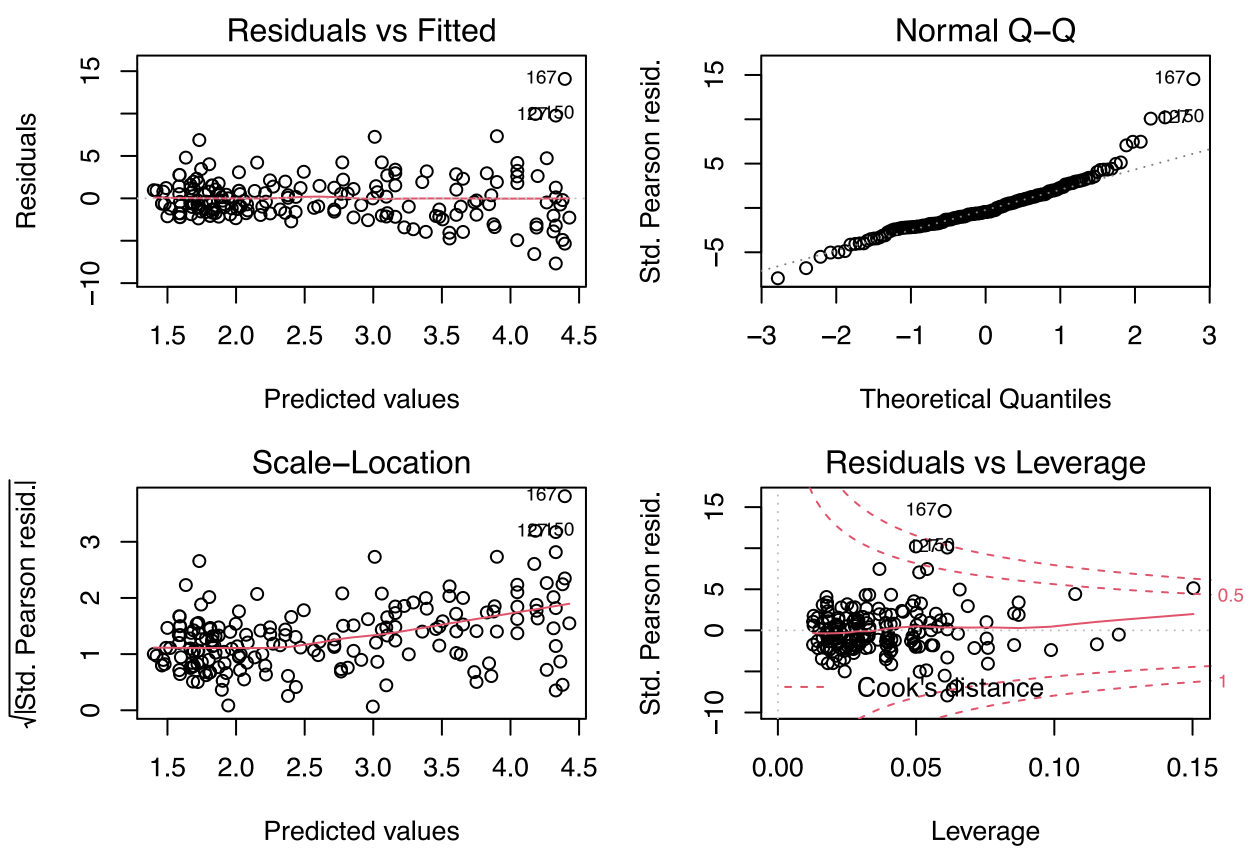
```
## 
## Call:
## glm(formula = count ~ attention + fluency + year + attention:year + 
##     attention:fluency + fluency:year, family = "poisson", data = dat)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -9.7727  -1.9782  -0.4667   1.2392  11.7460  
## 
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)    
## (Intercept)       -2.693020   0.655373  -4.109 3.97e-05 ***
## attention          0.265673   0.034050   7.802 6.08e-15 ***
## fluency            0.132890   0.029745   4.468 7.91e-06 ***
## year1             -1.954138   0.334908  -5.835 5.38e-09 ***
## attention:year1    0.112500   0.018018   6.244 4.27e-10 ***
## attention:fluency -0.008201   0.001371  -5.982 2.21e-09 ***
## fluency:year1      0.055679   0.013157   4.232 2.32e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 5013.2  on 184  degrees of freedom
## Residual deviance: 1387.4  on 178  degrees of freedom
## AIC: 2184.5
## 
## Number of Fisher Scoring iterations: 5
```

```r
fit13 <- glm(formula = count ~ attention + fluency + year + attention:year + 
    attention:fluency + fluency:year, family = "poisson", data = dat)
summary(fit13)
```

```
## 
## Call:
## glm(formula = count ~ attention + fluency + year + attention:year + 
##     attention:fluency + fluency:year, family = "poisson", data = dat)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -9.7727  -1.9782  -0.4667   1.2392  11.7460  
## 
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)    
```

```
## (Intercept)       -2.693020   0.655373  -4.109 3.97e-05 ***
## attention          0.265673   0.034050   7.802 6.08e-15 ***
## fluency            0.132890   0.029745   4.468 7.91e-06 ***
## year1             -1.954138   0.334908  -5.835 5.38e-09 ***
## attention:year1    0.112500   0.018018   6.244 4.27e-10 ***
## attention:fluency -0.008201   0.001371  -5.982 2.21e-09 ***
## fluency:year1      0.055679   0.013157   4.232 2.32e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 5013.2  on 184  degrees of freedom
## Residual deviance: 1387.4  on 178  degrees of freedom
## AIC: 2184.5
##
## Number of Fisher Scoring iterations: 5
```

```
par(mfrow = c(2, 2), mar = c(4.3, 4.3, 2, 1.2))
plot(fit13)
```




```
residuals(stepsearch, type="pearson")
```

```
##            1            2            3            4            5            6
## -1.875537220 -0.629331217 -0.708356461 -0.451770664  1.100037072 -1.694332059
##            7            8            9           10           11           12
##  1.516641307 -0.562914584 -0.391228577  0.872175114 -0.688261676  6.876033886
##           13           14           15           16           17           18
```

```
## -0.589283208 -2.347160541  0.504289732 -0.732259010 -1.103766884  0.893594968
##           19           20           21           22           23           24
## -0.668544100  0.789251349 -1.778847604  0.505191738 -0.956041935  1.527781610
##           25           26           27           28           29           30
## -1.201788832  0.584044690 -1.908724724  4.803883223 -2.116465415  3.455456104
##           31           32           33           34           35           36
## -1.769031151 -0.436944879 -0.298368046  1.759851379  0.893721419  0.425560209
##           37           38           39           40           41           42
##  4.218681861 -0.272409596  1.975314641 -1.714618849 -0.562914584  2.316796802
##           43           44           45           46           47           48
## -1.729450412 -0.873942010 -0.629331217  2.754396221 -1.021019537  1.414035000
##           49           50           51           52           53           54
## -2.217709792  1.175010370 -0.249495163 -1.613828617 -0.985458919 -2.369732537
##           55           56           57           58           59           60
## -1.557989288 -0.926556181  1.850152975  0.791887798 -1.962062372 -1.065126116
##           61           62           63           64           65           66
##  2.126072241 -1.425730544  0.963947024 -1.600458985  1.609898120  0.258183389
##           67           68           69           70           71           72
## -2.213995930  1.968137099  3.078054539 -2.066400412  0.131422526 -0.385463041
##           73           74           75           76           77           78
## -1.756004186 -1.180898477  0.615001632  0.914678705 -2.159611658 -0.602476857
##           79           80           81           82           83           84
## -0.410677437 -1.290534311  0.456857053 -1.095414144 -1.753398257  1.045729527
##           85           86           87           88           89           90
## -0.451770664 -0.615676919 -1.317922885  2.766438081  1.217735802  4.025189692
##           91           92           93           94           95           96
##  0.705925779 -0.664137118  7.331850140  2.848363960  0.007105262 -6.568162490
##           97           98           99          100          101          102
## -4.054038933  1.104191449 -3.963517433  1.215195367  1.922299880 -2.070428608
##          103          104          105          106          107          108
##  2.211313919  7.257498481 -4.735503900  0.368250481  1.256350945 -3.206950478
##          109          110          111          112          113          114
## -4.933444922 -0.974813193  2.310784484 -3.446855980 -3.899924818  1.024358695
##          115          116          117          118          119          120
##  2.705732134 -2.191525659 -3.059259550  1.927745253 -1.943376552  2.593155246
##          121          122          123          124          125          126
##  3.381217837  0.194569275 -2.078639988  0.120716806  1.384906475  1.796653166
##          127          128          129          130          131          132
##  9.759477214 -2.721886115 -2.146088577 -0.200454682  0.532791440  0.568848495
##          133          134          135          136          137          138
## -2.027592985 -1.572966782  0.367170767 -0.485222624 -1.265171389  4.730284364
##          139          140          141          142          143          144
##  0.063824085  2.629393040 -0.248033284  4.172222444 -3.641765704 -3.390186484
##          145          146          147          148          149          150
## -7.682089240 -2.478034072 -0.688141934 -1.312303368  4.245120503  9.954464467
##          151          152          153          154          155          156
##  2.958249297 -1.932994093  1.527842617  4.239760284 -1.747263213  2.969356677
##          157          158          159          160          161          162
##  3.132735474 -1.119488332 -1.314361023  1.473103271  0.728821178 -1.315756080
##          163          164          165          166          167          168
##  1.268188283 -0.471073379 -3.952943615 -2.178907821 14.088500821 -0.452193530
##          169          170          171          172          173          174
## -0.983320613 -5.342254974 -3.423326856  3.184610412 -4.902355879 -0.792579307
##          175          176          177          178          179          180
```

```
## -0.004183694  1.437925829  0.169408926 -2.252558344 -0.732396470 -2.274811798
##          181          182          183          184          185
## -0.940104302 -2.576482154 -3.077589995  3.248688357 -2.048805363
```

```
residuals(stepsearch, type="deviance")
```

```
##            1            2            3            4            5            6
## -2.305067372 -0.666002645 -0.741509893 -0.466690453  1.027107751 -1.978162076
##            7            8            9           10           11           12
##  1.392346117 -0.588580303 -0.400133460  0.833278118 -0.722471554  5.205687051
##           13           14           15           16           17           18
## -0.617365704 -2.958257703  0.486735417 -0.775239861 -1.212276035  0.850731024
##           19           20           21           22           23           24
## -0.704514796  0.751872597 -2.012959806  0.490839082 -1.021046888  1.401644448
##           25           26           27           28           29           30
## -1.353948524  0.563652612 -2.258489998  3.831603276 -2.993134094  2.911363060
##           31           32           33           34           35           36
## -2.159565089 -0.450911514 -0.303919022  1.600137997  0.850845264  0.414769970
##           37           38           39           40           41           42
##  3.556454563 -0.277408204  1.813606496 -1.913271629 -0.588580303  2.039779081
##           43           44           45           46           47           48
## -2.023764249 -0.943324728 -0.666002645  2.363492360 -1.114549309  1.293187139
##           49           50           51           52           53           54
## -3.136315265  1.091733097 -0.254088240 -1.874127026 -1.048720204 -3.351307893
##           55           56           57           58           59           60
## -1.802391530 -0.987721035  1.664897188  0.748353157 -2.276707012 -1.154019786
##           61           62           63           64           65           66
##  1.932791991 -1.580655606  0.899273304 -1.761373664  1.462123845  0.253611130
##           67           68           69           70           71           72
## -3.131063071  1.758454102  2.672882605 -2.567852033  0.130330161 -0.396359086
##           73           74           75           76           77           78
## -2.013649508 -1.289234903  0.595742237  0.856486084 -2.697010045 -0.631807688
##           79           80           81           82           83           84
## -0.424456738 -1.436341077  0.443544353 -1.202367005 -2.138285163  0.983680687
##           85           86           87           88           89           90
## -0.466690453 -0.646282672 -1.469588722  2.372137151  1.122003545  3.336463456
##           91           92           93           94           95           96
##  0.673976922 -0.704871650  6.416469578  2.660168978  0.007102081 -8.082480105
##           97           98           99          100          101          102
## -4.743700909  1.060168878 -4.566086570  1.152467960  1.821743265 -2.573423338
##          103          104          105          106          107          108
##  2.044343608  6.026890727 -5.788741986  0.361672794  1.196443014 -3.441886848
##          109          110          111          112          113          114
## -5.708185227 -1.002359217  2.186086768 -3.747714251 -4.267947521  0.997382408
##          115          116          117          118          119          120
##  2.494379919 -2.360849758 -3.339184090  1.848422903 -2.064117967  2.462995083
##          121          122          123          124          125          126
##  3.072026567  0.193249331 -2.231507422  0.120439709  1.297582466  1.731852864
##          127          128          129          130          131          132
##  8.475406534 -3.355180031 -2.351755505 -0.201211511  0.521534341  0.556354332
##          133          134          135          136          137          138
## -2.212130953 -1.702366906  0.363979940 -0.489991872 -1.388724913  4.367627874
##          139          140          141          142          143          144
##  0.063619060  2.504299691 -0.249629284  3.857195082 -4.286171270 -3.742536000
##          145          146          147          148          149          150
```

```
## -9.772672058 -2.697639847 -0.700204054 -1.422429598  3.763323653  8.552850387
##          151          152          153          154          155          156
##  2.774780496 -2.042542241  1.456701497  3.704300073 -1.876323647  2.724934550
##          157          158          159          160          161          162
##  2.782418318 -1.186400021 -1.370453236  1.388394515  0.710550965 -1.402951833
##          163          164          165          166          167          168
##  1.239157751 -0.480844968 -4.695959221 -2.383069285 11.745952575 -0.457616014
##          169          170          171          172          173          174
## -1.017763227 -6.072168381 -4.001302253  2.931908862 -5.502483362 -0.819084337
##          175          176          177          178          179          180
## -0.004184346  1.370397963  0.168022702 -2.522274174 -0.742851940 -2.380360913
##          181          182          183          184          185
## -0.985293224 -2.922460582 -3.311120532  3.050220819 -2.138978534
```

```
cooks.distance(stepsearch)
```

```
##            1            2            3            4            5            6
## 1.332418e-02 1.875239e-03 2.143197e-03 7.090186e-04 3.256251e-03 2.248201e-02
##            7            8            9           10           11           12
## 4.845809e-03 1.484482e-03 1.629685e-03 3.730846e-03 1.588900e-03 3.836258e-01
##           13           14           15           16           17           18
## 1.097554e-03 1.519872e-02 1.069888e-03 1.489026e-03 7.770326e-03 4.813569e-03
##           19           20           21           22           23           24
## 2.242709e-03 4.083729e-03 2.853456e-02 9.630650e-04 1.153753e-02 7.628753e-03
##           25           26           27           28           29           30
## 9.521855e-03 6.473803e-04 2.300687e-02 2.484604e-01 2.169342e-02 9.290008e-02
##           31           32           33           34           35           36
## 1.331517e-02 4.963918e-04 2.735741e-04 5.877861e-03 1.912847e-03 1.682839e-03
##           37           38           39           40           41           42
## 8.609784e-02 1.873805e-04 5.696230e-02 4.291905e-02 1.484482e-03 3.771457e-02
##           43           44           45           46           47           48
## 5.587775e-03 4.735494e-03 1.875239e-03 2.606705e-02 3.294928e-03 9.996013e-03
##           49           50           51           52           53           54
## 1.826927e-02 1.308517e-02 2.748389e-04 1.170128e-02 5.906813e-03 3.581650e-02
##           55           56           57           58           59           60
## 1.888575e-02 2.055982e-03 2.183232e-02 2.740300e-03 3.036265e-02 5.408799e-03
##           61           62           63           64           65           66
## 3.590561e-02 1.248265e-02 1.171996e-02 5.395660e-02 1.214190e-02 1.863641e-04
##           67           68           69           70           71           72
## 1.684196e-02 3.560324e-02 2.268969e-02 1.113027e-02 3.826665e-05 4.962769e-04
##           73           74           75           76           77           78
## 2.740104e-02 5.950426e-03 3.008745e-03 7.378648e-03 9.999037e-03 1.014814e-03
##           79           80           81           82           83           84
## 5.808390e-04 3.508646e-03 1.268791e-03 5.297984e-03 1.869412e-02 2.042967e-03
##           85           86           87           88           89           90
## 7.090186e-04 1.020021e-03 4.668697e-03 3.219733e-02 6.547874e-03 4.212539e-02
##           91           92           93           94           95           96
## 1.575056e-03 1.879721e-03 3.040947e-01 9.180947e-02 4.483643e-07 4.537480e-01
##           97           98           99          100          101          102
## 1.046780e-01 5.023138e-03 4.202963e-02 1.210374e-02 2.231798e-02 5.044858e-02
##          103          104          105          106          107          108
## 3.474796e-02 4.537474e-01 1.920375e-01 6.862026e-04 5.180442e-03 1.021959e-01
##          109          110          111          112          113          114
## 8.830242e-02 2.793394e-03 1.428607e-02 7.184526e-02 1.933682e-01 4.531012e-03
##          115          116          117          118          119          120
```

```
## 2.946595e-02 1.363870e-02 2.944486e-02 2.102234e-02 1.630816e-02 3.769955e-02
##          121          122          123          124          125          126
## 3.010721e-02 1.513717e-04 1.046823e-02 1.448053e-04 8.074556e-03 4.819340e-02
##          127          128          129          130          131          132
## 9.464595e-01 3.068356e-02 1.841584e-02 3.273149e-04 8.717408e-04 1.035258e-03
##          133          134          135          136          137          138
## 2.075643e-02 8.120523e-03 4.409593e-04 5.397090e-03 1.216007e-02 6.654016e-01
##          139          140          141          142          143          144
## 2.061270e-05 3.375661e-02 4.985390e-04 3.358416e-01 4.443677e-02 4.139713e-02
##          145          146          147          148          149          150
## 5.864184e-01 2.212473e-02 2.247791e-03 1.043345e-02 9.098530e-02 7.829299e-01
##          151          152          153          154          155          156
## 6.703462e-02 3.027891e-02 6.147258e-03 1.143430e-01 1.228761e-02 2.321929e-02
##          157          158          159          160          161          162
## 3.702538e-02 6.972225e-03 4.905794e-03 8.033945e-03 3.364554e-03 5.681927e-03
##          163          164          165          166          167          168
## 1.598146e-02 6.814752e-04 4.617599e-02 1.250265e-02 1.937970e+00 1.264784e-03
##          169          170          171          172          173          174
## 2.579739e-03 2.786549e-01 2.749865e-02 8.339191e-02 1.957686e-01 2.967760e-03
##          175          176          177          178          179          180
## 1.374133e-07 5.595506e-03 1.230307e-04 2.090444e-02 4.120777e-03 8.994111e-02
##          181          182          183          184          185
## 6.848064e-03 4.185313e-02 5.727609e-02 1.575706e-01 5.336714e-02
```

```r
rstandard(stepsearch, type="pearson")
```

```
##            1            2            3            4            5            6
## -1.899620801 -0.639355960 -0.718573358 -0.457104701  1.110162788 -1.737912937
##            7            8            9           10           11           12
##  1.527624143 -0.571790262 -0.404628860  0.886546942 -0.696115563  7.058887986
##           13           14           15           16           17           18
## -0.595629689 -2.369298378  0.511457624 -0.739209940 -1.127147072  0.911533750
##           19           20           21           22           23           24
## -0.679804673  0.806411874 -1.831068065  0.511655063 -0.994323781  1.544781237
##           25           26           27           28           29           30
## -1.228058744  0.587861554 -1.948775653  4.970122367 -2.150919321  3.543791970
##           31           32           33           34           35           36
## -1.794451287 -0.440833905 -0.301494605  1.771352406  0.901060852  0.438407434
##           37           38           39           40           41           42
##  4.287286743 -0.274765812  2.065556668 -1.792951018 -0.571790262  2.370591410
##           43           44           45           46           47           48
## -1.740578811 -0.891962562 -0.639355960  2.786571098 -1.032015746  1.437767758
##           49           50           51           52           53           54
## -2.245653303  1.211140828 -0.253210707 -1.638269223 -1.005411437 -2.419928637
##           55           56           57           58           59           60
## -1.597817990 -0.934165260  1.889343247  0.803563926 -2.012867047 -1.082205885
##           61           62           63           64           65           66
##  2.181493534 -1.454861504  1.002517928 -1.701641457  1.635281965  0.260650405
##           67           68           69           70           71           72
## -2.239858240  2.026951677  3.103332188 -2.084838259  0.132422493 -0.389843637
##           73           74           75           76           77           78
## -1.806852049 -1.197914546  0.631054914  0.940978639 -2.175521949 -0.608233693
##           79           80           81           82           83           84
## -0.415485584 -1.299879782  0.466102002 -1.111727326 -1.788888894  1.052458425
##           85           86           87           88           89           90
```

```
## -0.457104701 -0.621344155 -1.330040906  2.805759780  1.235872248  4.061015876
##           91           92           93           94           95           96
##  0.713528482 -0.673695363  7.470373347  2.951557363  0.007310897 -6.790604739
##           97           98           99          100          101          102
## -4.139798651  1.119571911 -3.999795635  1.247819520  1.960959839 -2.148188001
##          103          104          105          106          107          108
##  2.263209667  7.461641680 -4.867966451  0.374503369  1.270387298 -3.309993303
##          109          110          111          112          113          114
## -4.994201428 -0.984595301  2.331935169 -3.516258811 -4.057109683  1.039289678
##          115          116          117          118          119          120
##  2.742579600 -2.212787773 -3.092059723  1.964166818 -1.971703077  2.641729635
##          121          122          123          124          125          126
##  3.411691149  0.197202179 -2.095905422  0.124595569  1.404604443  1.880407852
##          127          128          129          130          131          132
## 10.073059887 -2.759992560 -2.175129371 -0.205805013  0.538370765  0.575047834
##          133          134          135          136          137          138
## -2.061947410 -1.590540468  0.371259311 -0.518229134 -1.296795367  5.131603694
##          139          140          141          142          143          144
##  0.064907817  2.672533800 -0.254621348  4.416499417 -3.683278798 -3.431644627
##          145          146          147          148          149          150
## -7.928923165 -2.508347136 -0.699130274 -1.338773611  4.317047501 10.212652331
##          151          152          153          154          155          156
##  3.032771638 -1.984336753  1.541612338  4.329340416 -1.771057964  2.996118080
##          157          158          159          160          161          162
##  3.172806940 -1.140304386 -1.327112913  1.491604642  0.744158176 -1.330456126
##          163          164          165          166          167          168
##  1.308936559 -0.476006401 -3.992815037 -2.198545284 14.533856998 -0.461496621
##          169          170          171          172          173          174
## -0.992296521 -5.511130732 -3.450883273  3.270363183 -5.033204535 -0.805177783
##          175          176          177          178          179          180
## -0.004291555  1.451235364  0.171860989 -2.283934594 -0.750896906 -2.396277795
##          181          182          183          184          185
## -0.964044248 -2.630462193 -3.139557614  3.400132653 -2.131381620
```

```
rstandard(stepsearch, type="deviance")
```

```
##             1             2             3             4             5
##  -2.334666506  -0.676611536  -0.752204975  -0.472200647   1.036562160
##             6             7             8             9            10
##  -2.029043508   1.402428864  -0.597860661  -0.413838751   0.847008996
##            11            12            13            14            15
##  -0.730715816   5.344121683  -0.624014629  -2.986159257   0.493653795
##            16            17            18            19            20
##  -0.782598785  -1.237954683   0.867809319  -0.716381239   0.768220403
##            21            22            23            24            25
##  -2.072052945   0.497118782  -1.061931664   1.417240547  -1.383544496
##            26            27            28            29            30
##   0.567336210  -2.305880081   3.964196518  -3.041859275   2.985789639
##            31            32            33            34            35
##  -2.190597012  -0.454924850  -0.307103750   1.610595261   0.857832589
##            36            37            38            39            40
##   0.427291448   3.614290199  -0.279807657   1.896460905  -2.000679227
##            41            42            43            44            45
##  -0.597860661   2.087141506  -2.036786453  -0.962775942  -0.676611536
##            46            47            48            49            50
```

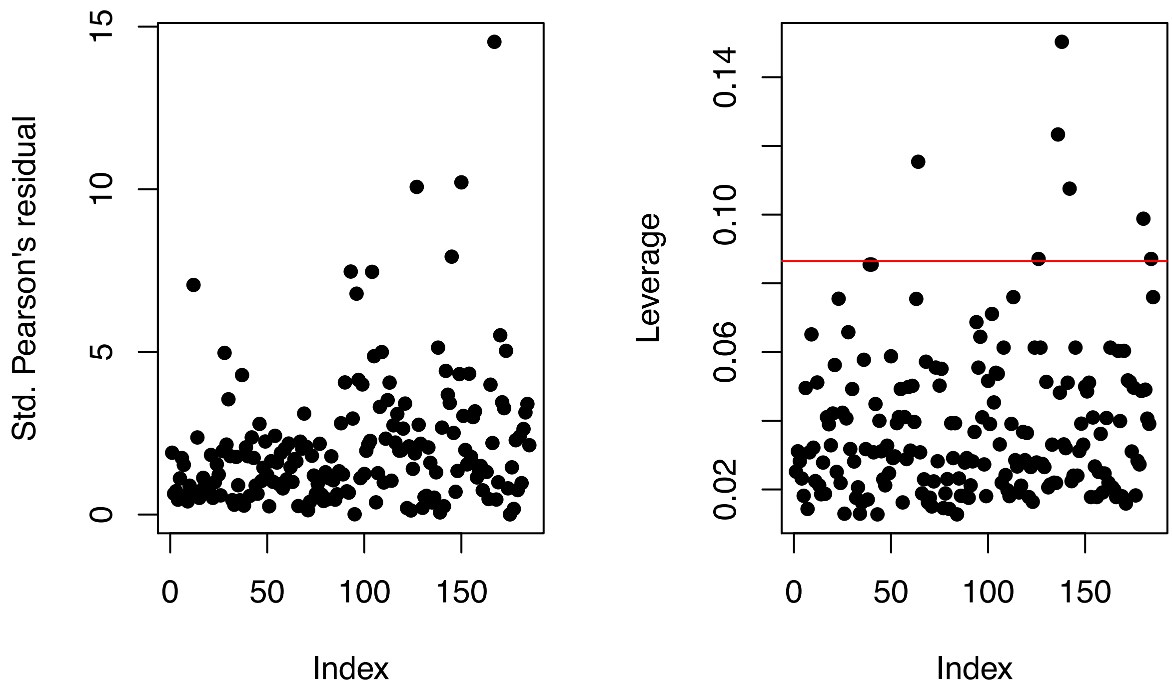
```
##    2.391100979  -1.126552818   1.314891622  -3.175833357   1.125302857
##             51            52            53            54            55
##   -0.257872185  -1.902509717  -1.069953569  -3.422295898  -1.848468172
##             56            57            58            59            60
##   -0.995832413   1.700163339   0.759387381  -2.335658940  -1.172525004
##             61            62            63            64            65
##    1.983174960  -1.612952042   0.935256385  -1.872729308   1.485177679
##             66            67            68            69            70
##    0.256034456  -3.167637900   1.811002645   2.694832894  -2.590764177
##             71            72            73            74            75
##    0.131321816  -0.400863509  -2.071957895  -1.307812038   0.611292794
##             76            77            78            79            80
##    0.881112794  -2.716879457  -0.637844789  -0.429426212  -1.446742415
##             81            82            83            84            85
##    0.452519907  -1.220272955  -2.181566318   0.990010323  -0.472200647
##             86            87            88            89            90
##   -0.652231631  -1.483101278   2.405854322   1.138714195   3.366159635
##             91            92            93            94            95
##    0.681235540  -0.715016145   6.537698180   2.756544263   0.007307624
##             96            97            98            99           100
##   -8.356207354  -4.844049835   1.074936142  -4.607880106   1.183408080
##            101           102           103           104           105
##    1.858380899  -2.670073778   2.092320849   6.196418664  -5.950665944
##            106           107           108           109           110
##    0.367813993   1.209810056  -3.552478435  -5.778482837  -1.012417745
##            111           112           113           114           115
##    2.206096090  -3.823174898  -4.439965390   1.011920186   2.528349127
##            116           117           118           119           120
##   -2.383754650  -3.374985503   1.883345803  -2.094204410   2.509131342
##            121           122           123           124           125
##    3.099713285   0.195864374  -2.250042591   0.124309568   1.316038396
##            126           127           128           129           130
##    1.812586750   8.747730612  -3.402152599  -2.383579377  -0.206582043
##            131           132           133           134           135
##    0.526995783   0.562417509  -2.249612089  -1.721386292   0.368032953
##            136           137           138           139           140
##   -0.523322803  -1.423437211   4.738179273   0.064699310   2.545388029
##            141           142           143           144           145
##   -0.256259739   4.083027706  -4.335030051  -3.788302978 -10.086678696
##            146           147           148           149           150
##   -2.730639285  -0.711385003  -1.451121179   3.827087348   8.774684739
##            151           152           153           154           155
##    2.844681007  -2.096794633   1.469830057   3.782566689  -1.901875982
##            156           157           158           159           160
##    2.749493092   2.818008805  -1.208460248  -1.383749331   1.405831990
##            161           162           163           164           165
##    0.725503492  -1.418626058   1.278973245  -0.485880317  -4.743325081
##            166           167           168           169           170
##   -2.404546758  12.117257698  -0.467030663  -1.027053534  -6.264117669
##            171           172           173           174           175
##   -4.033511142   3.010857078  -5.649350006  -0.832104125  -0.004292224
##            176           177           178           179           180
##    1.383082456   0.170454701  -2.557407340  -0.761616483  -2.507462816
##            181           182           183           184           185
```

```
## -1.010383915 -2.983689238 -3.377790315  3.192413142 -2.225189184
```

```
par(mfrow=c(1,2))
plot(abs(rstandard(stepsearch, type="pearson")), xlab="Index", ylab="Std. Pearson's residual", pch=16)
plot(influence(stepsearch)$hat, xlab="Index", ylab="Leverage", pch=16)
l_threshold <- 2*8 / 185
l_threshold
```

```
## [1] 0.08648649
```

```
abline(h=l_threshold, col="red")
```




```
order(abs(rstandard(stepsearch, type="pearson")), decreasing = TRUE)[1:5]
```

```
## [1] 167 150 127 145  93
```

```
order(influence(stepsearch)$hat, decreasing=TRUE)[1:5]
```

```
## [1] 138 136  64 142 180
```

There is no reoccuring indices.

4. Fit own models + evaluate quality of fit

```
# fit defined model, try a few
require(MASS)
```

```
## Loading required package: MASS
```

```
lin <- glm(count~attention+fluency+year, data=dat)
pois <- glm(count~attention+fluency+year, family= poisson, data=dat)
qpois <- glm(count~attention+fluency+year, family= quasipoisson, data=dat)
nbin <- glm.nb(count~attention+fluency+year, data=dat)
```

```
summary(lin)
```

```
##
## Call:
## glm(formula = count ~ attention + fluency + year, data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -48.466  -12.973   -2.458    8.049  152.211
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -71.6017    10.6103  -6.748 1.96e-10 ***
## attention     3.5482     0.6829   5.196 5.46e-07 ***
## fluency       0.4193     0.4599   0.912    0.363
## year1        28.2040     3.3027   8.540 5.39e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 492.5757)
##
##     Null deviance: 161881  on 184  degrees of freedom
## Residual deviance:  89156  on 181  degrees of freedom
## AIC: 1677.9
##
## Number of Fisher Scoring iterations: 2
```

```r
list(residual.deviance           = deviance(lin),
     residual.degrees.of.freedom = df.residual(lin),
     chisq.p.value               = pchisq(deviance(lin), df.residual(lin), lower = F)
     )
```

```
## $residual.deviance
## [1] 89156.2
##
## $residual.degrees.of.freedom
## [1] 181
##
## $chisq.p.value
## [1] 0
```

```r
summary(pois)
```

```
##
## Call:
## glm(formula = count ~ attention + fluency + year, family = poisson,
##     data = dat)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -9.851  -2.102  -0.342   1.544  12.594
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.775411   0.118473 -14.986   <2e-16 ***
## attention    0.172228   0.007092  24.285   <2e-16 ***
## fluency      0.006242   0.004432   1.408    0.159
```

```
## year1         1.630112   0.045119  36.130   <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: 5013.2  on 184  degrees of freedom
## Residual deviance: 1518.0  on 181  degrees of freedom
## AIC: 2309.1
##
## Number of Fisher Scoring iterations: 5
```

```
list(residual.deviance           = deviance(pois),
     residual.degrees.of.freedom = df.residual(pois),
     chisq.p.value               = pchisq(deviance(pois), df.residual(pois), lower = F)
     )
```

```
## $residual.deviance
## [1] 1517.996
##
## $residual.degrees.of.freedom
## [1] 181
##
## $chisq.p.value
## [1] 1.030629e-209
```

```
summary(qpois)
```

```
##
## Call:
## glm(formula = count ~ attention + fluency + year, family = quasipoisson,
##     data = dat)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -9.851  -2.102  -0.342   1.544  12.594
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.775411   0.354490  -5.008 1.30e-06 ***
## attention    0.172228   0.021221   8.116 7.15e-14 ***
## fluency      0.006242   0.013262   0.471    0.638
## year1        1.630112   0.135002  12.075  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 8.953056)
##
##     Null deviance: 5013.2  on 184  degrees of freedom
## Residual deviance: 1518.0  on 181  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```
list(residual.deviance           = deviance(pois),
     residual.degrees.of.freedom = df.residual(pois),
```

```r
     chisq.p.value                 = pchisq(deviance(pois), df.residual(pois), lower = F)
     )
```

```
## $residual.deviance
## [1] 1517.996
##
## $residual.degrees.of.freedom
## [1] 181
##
## $chisq.p.value
## [1] 1.030629e-209
```

```r
summary(nbin)
```

```
##
## Call:
## glm.nb(formula = count ~ attention + fluency + year, data = dat,
##     init.theta = 2.845857478, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4870  -0.9291  -0.2117   0.6367   3.1446
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.089727   0.324394  -3.359 0.000781 ***
## attention    0.148918   0.020461   7.278 3.39e-13 ***
## fluency      0.001269   0.013711   0.093 0.926250
## year1        1.502919   0.099641  15.083  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(2.8459) family taken to be 1)
##
##     Null deviance: 599.21  on 184  degrees of freedom
## Residual deviance: 199.74  on 181  degrees of freedom
## AIC: 1315
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  2.846
##           Std. Err.:  0.365
##
##  2 x log-likelihood:  -1304.993
```

```r
list(residual.deviance             = deviance(nbin),
     residual.degrees.of.freedom = df.residual(nbin),
     chisq.p.value                 = pchisq(deviance(nbin), df.residual(nbin), lower = F)
     )
```

```
## $residual.deviance
## [1] 199.743
##
## $residual.degrees.of.freedom
```

```
## [1] 181
##
## $chisq.p.value
## [1] 0.1616301
```

From now, on let's use the negbinomial model:

```
# response variable and predictors for NegBin GLM
y <- as.numeric(dat$count)
x2 <- cbind(as.numeric(dat$attention), as.numeric(dat$year))
X2 <- cbind(1,x2)
```

```
# IWLS
# find initial beta
fit3 <- lm(y~x2)
beta <- fit3$coefficients

# inverse link function
log.link <- function(u){
  exp(u)
}

# deviance function
D <- function(p){ # p is the estimated mean mu
  a <- y*log(y/p)
  b <- (3+y)*log((p+3)/(y+3))
  a[y==0] <- 0
  2*sum(a+b)
}

oldD <- D(log.link(as.numeric(X2%*%beta)))
jj <- 0
while(jj==0){
  eta <- X2%*%beta # estimated linear predictor
  mu <- log.link(eta) # estimated mean response
  z <- eta + ((y-mu)/mu) # form the adjusted variate
  w <- 3*mu/(mu+3) # weights
  lmod <- lm(z~x2, weights=w) # regress z on x with weights w, includes intercept anyway
  beta <- as.vector(lmod$coeff) # newbeta
  newD <- D(log.link(X2%*%beta))
  control <- abs(newD-oldD)/(abs(newD)+0.1)
  if(control<1e-8)
    jj <- 1
  oldD <- newD
}
beta # final estimate
```

```
## [1] -2.5960792  0.1504124  1.5039373
```

```
newD # last deviance calculated
```

```
## [1] 207.579
```

```
# Diagnostics

# Pearson's statistic for negbin
X3 <- 0
```

```r
for (i in 1:185){
  X3 <- X3 + (y[i]-mu[i])^2/(mu[i]*(mu[i]+3)/3)
}

# dispersion parameter estimate
phi <- X3/(185-3) #n-p, n number of rows, p number of predictors
phi
```

```
## [1] 1.139763
```

```r
# computation of covariance matrix and standard residuals for estimates
J <- t(X2)%*%diag(as.vector(w))%*%X2
invJ <- solve(J)
cov.beta <- phi*invJ
beta.sd <- sqrt(as.vector(diag(cov.beta)))
beta.sd
```

```
## [1] 0.36100436 0.01646965 0.10388851
```

```r
# confidence intervals for estimates of model parameters
beta1.CI = c(beta[1]-qt(0.975, 182)*beta.sd[1],beta[1]+qt(0.975, 182)*beta.sd[1])
beta1.CI
```

```
## [1] -3.308371 -1.883787
```

```r
beta2.CI = c(beta[2]-qt(0.975, 182)*beta.sd[2],beta[2]+qt(0.975, 182)*beta.sd[2])
beta2.CI
```

```
## [1] 0.1179164 0.1829084
```

```r
beta3.CI = c(beta[3]-qt(0.975, 182)*beta.sd[3],beta[3]+qt(0.975, 182)*beta.sd[3])
beta3.CI
```

```
## [1] 1.298956 1.708918
```

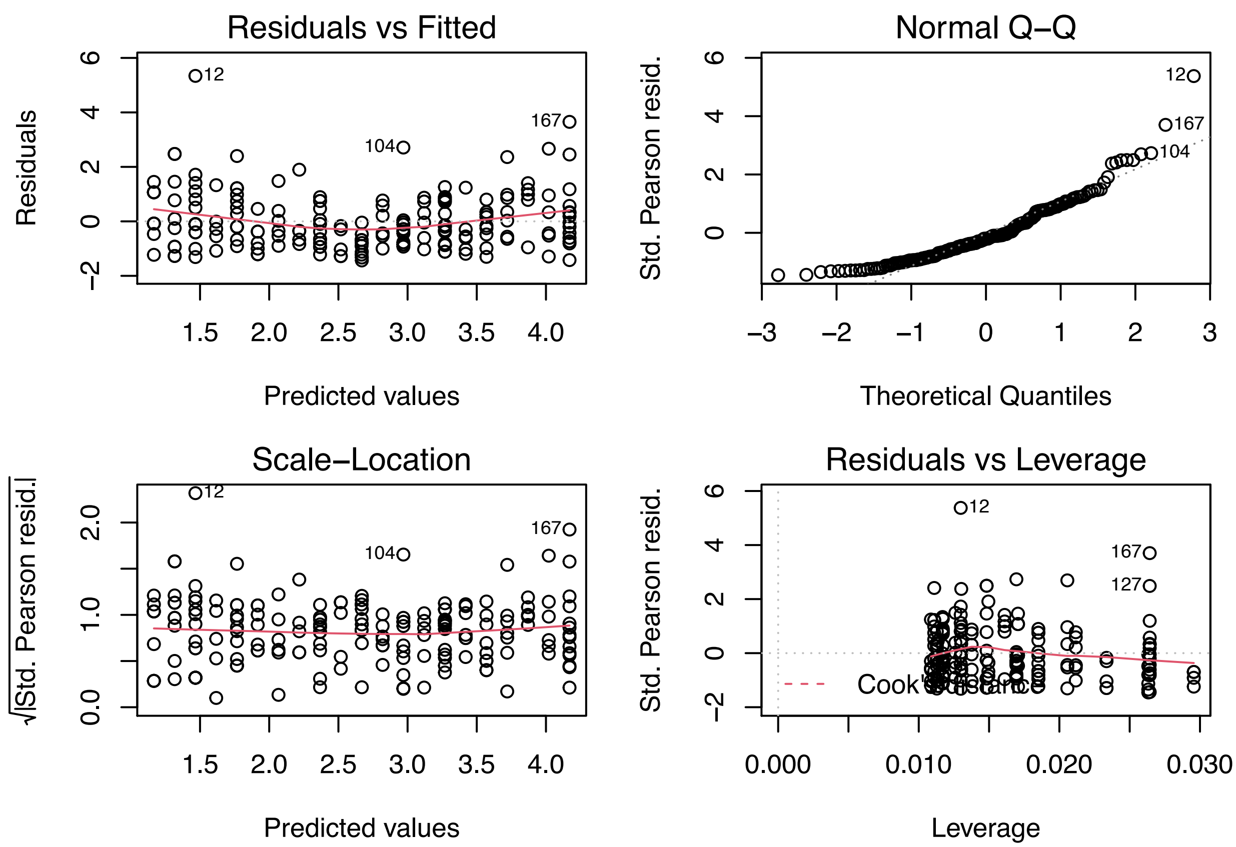
```r
# sanity check
fit30 <- glm.nb(count~attention+year, data=dat)
```

```r
# summary and plots of sanity check
summary(fit30)
```

```
##
## Call:
## glm.nb(formula = count ~ attention + year, data = dat, init.theta = 2.844392692,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4930  -0.9298  -0.2105   0.6310   3.1280
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.08488    0.32062  -3.384 0.000715 ***
## attention    0.15010    0.01576   9.522  < 2e-16 ***
## year1        1.50284    0.09935  15.127  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for Negative Binomial(2.8444) family taken to be 1)
##
##     Null deviance: 598.96  on 184  degrees of freedom
## Residual deviance: 199.68  on 182  degrees of freedom
## AIC: 1313
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  2.844
##           Std. Err.:  0.365
##
##  2 x log-likelihood:  -1305.002
```

```
par(mfrow = c(2, 2), mar = c(4.3, 4.3, 2, 1.2))
plot(fit30)
```

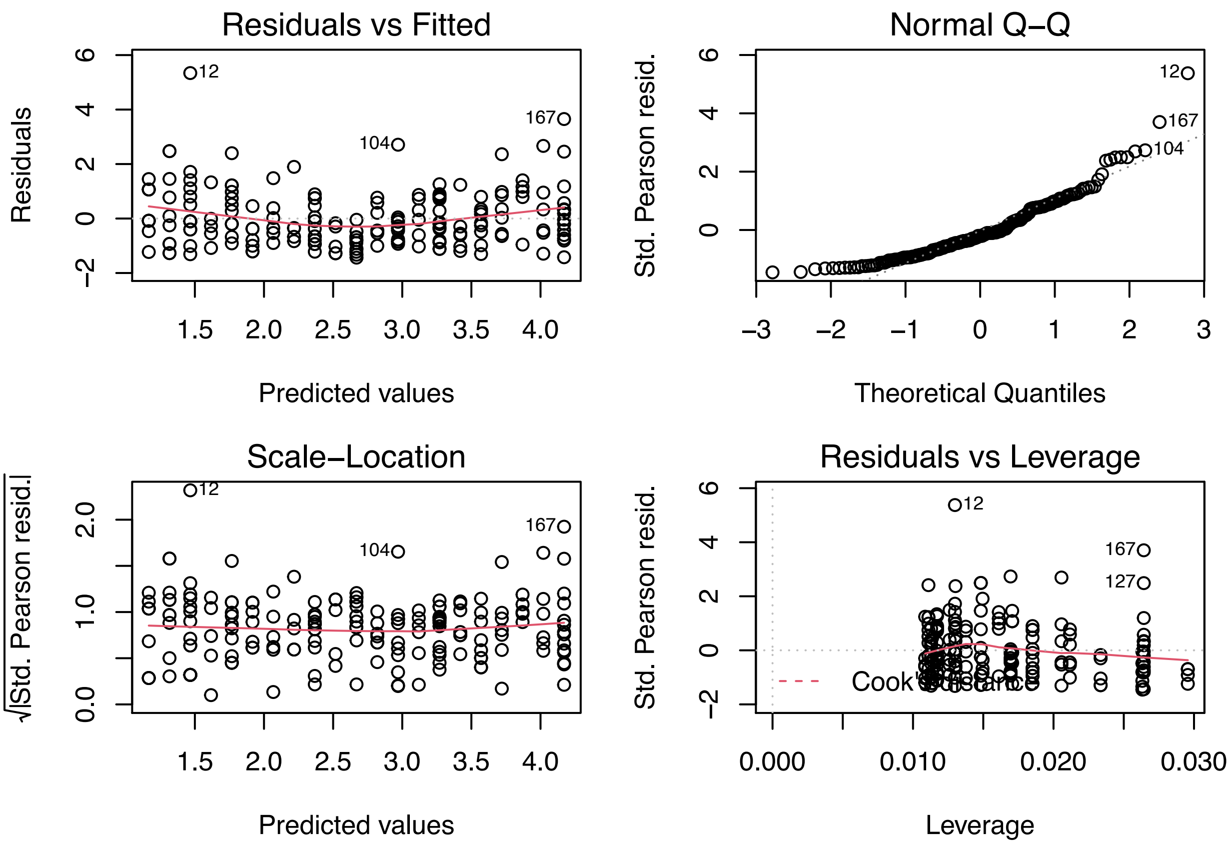



```
# spotting potential outliers
dat2 <- dat[-c(12,167,127),]
fit31 <- glm.nb(count~attention+year, data=dat)

# summary and diagnostic plots without suspicious points
summary(fit31)
```

```
##
## Call:
## glm.nb(formula = count ~ attention + year, data = dat, init.theta = 2.844392692,
##     link = log)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4930  -0.9298  -0.2105   0.6310   3.1280
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.08488    0.32062  -3.384 0.000715 ***
## attention    0.15010    0.01576   9.522  < 2e-16 ***
## year1        1.50284    0.09935  15.127  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(2.8444) family taken to be 1)
##
##     Null deviance: 598.96  on 184  degrees of freedom
## Residual deviance: 199.68  on 182  degrees of freedom
## AIC: 1313
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  2.844
##           Std. Err.:  0.365
##
##  2 x log-likelihood:  -1305.002
```

```r
par(mfrow = c(2, 2), mar = c(4.3, 4.3, 2, 1.2))
plot(fit31)
```




```
# computation of deviance residuals
a <- y*log(y/p)
b <- (3+y)*log((p+3)/(y+3))
a[y==0] <- 0
d <- sign(y-mu)*sqrt(2*(a+b))
summary(d)
```

```
##        V1
##  Min.   :-3.1602
##  1st Qu.:-1.4254
##  Median :-0.4360
##  Mean   :-0.3626
##  3rd Qu.: 0.7421
##  Max.   : 3.6340
```

```
z <- beta/beta.sd
z # large n makes the student t distribution tend to normal distribution
```

```
## [1] -7.191268  9.132703 14.476455
```

```
# use step search to compare models
fit32 <- glm.nb(count~attention+fluency+year, data=dat)
stepsearch <- step(fit32,~.^2,test="Chisq")
```

```
## Start:  AIC=1312.99
## count ~ attention + fluency + year
##
##                      Df Deviance    AIC    LRT  Pr(>Chi)
## + attention:year      1   181.67 1296.9 18.070 2.129e-05 ***
```

```
## + fluency:year       1   182.48 1297.7  17.263 3.255e-05 ***
## - fluency            1   199.75 1311.0   0.009    0.9234
## <none>                   199.74 1313.0
## + attention:fluency  1   199.73 1315.0   0.011    0.9176
## - attention          1   253.82 1365.1  54.075 1.930e-13 ***
## - year               1   406.34 1517.6 206.600 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=1296.02
## count ~ attention + fluency + year + attention:year
##
##                     Df Deviance    AIC     LRT  Pr(>Chi)
## + fluency:year       1   194.40 1293.6  4.4327   0.03526 *
## - fluency            1   198.84 1294.0  0.0100   0.92023
## <none>                   198.84 1296.0
## + attention:fluency  1   197.31 1296.5  1.5249   0.21687
## - attention:year     1   218.76 1314.0 19.9290 8.037e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=1293.55
## count ~ attention + fluency + year + attention:year + fluency:year
##
##                     Df Deviance    AIC    LRT Pr(>Chi)
## + attention:fluency  1   195.37 1292.9 2.6164  0.10576
## <none>                   197.98 1293.5
## - fluency:year       1   202.49 1296.1 4.5082  0.03373 *
## - attention:year     1   203.53 1297.1 5.5413  0.01857 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step:  AIC=1292.91
## count ~ attention + fluency + year + attention:year + fluency:year +
##     attention:fluency
##
##                     Df Deviance    AIC    LRT Pr(>Chi)
## <none>                   198.59 1292.9
## - attention:fluency  1   201.27 1293.6 2.6777  0.10176
## - fluency:year       1   204.24 1296.5 5.6466  0.01749 *
## - attention:year     1   204.78 1297.1 6.1902  0.01285 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
stepsearch$anova
```

```
##                  Step Df  Deviance Resid. Df Resid. Dev      AIC
## 1                     NA        NA       181   199.7430 1312.993
## 2    + attention:year -1 0.9079555       180   198.8351 1296.023
## 3      + fluency:year -1 0.8509229       179   197.9841 1293.552
## 4 + attention:fluency -1 0.6095962       178   198.5937 1292.906
```

```
summary(stepsearch)
```

```
##
```

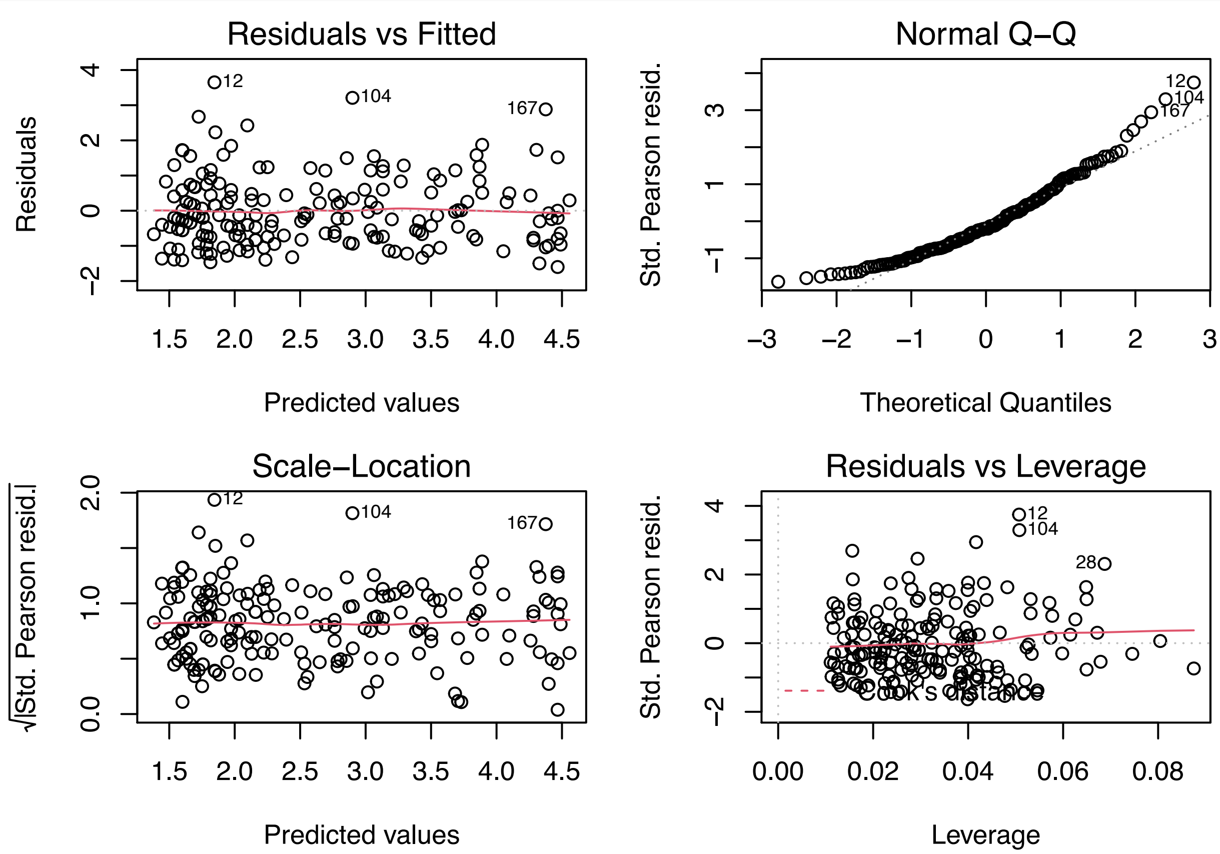
```
## Call:
## glm.nb(formula = count ~ attention + fluency + year + attention:year +
##     fluency:year + attention:fluency, data = dat, init.theta = 3.388864932,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8027  -0.8656  -0.2227   0.5087   2.6237
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.740362   1.570415  -1.108  0.26777
## attention          0.219360   0.082994   2.643  0.00822 **
## fluency            0.082527   0.071721   1.151  0.24987
## year1             -1.754953   0.634029  -2.768  0.00564 **
## attention:year1    0.096853   0.038469   2.518  0.01181 *
## fluency:year1      0.060243   0.026551   2.269  0.02327 *
## attention:fluency -0.005764   0.003522  -1.637  0.10166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(3.3889) family taken to be 1)
##
##     Null deviance: 690.22  on 184  degrees of freedom
## Residual deviance: 198.59  on 178  degrees of freedom
## AIC: 1294.9
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  3.389
##           Std. Err.:  0.452
##
##  2 x log-likelihood:  -1278.906
```

```r
fit33 <- glm.nb(formula = count ~ attention + year + attention:year +
    fluency:year, data = dat, init.theta = 3.388864932,
    link = log)
summary(fit33)
```

```
##
## Call:
## glm.nb(formula = count ~ attention + year + attention:year +
##     fluency:year, data = dat, init.theta = 3.3109128, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8695  -0.8606  -0.2068   0.5286   2.4397
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.69026    0.47541   1.452  0.14652
## attention         0.09320    0.02890   3.225  0.00126 **
## year1            -1.49679    0.61974  -2.415  0.01573 *
## attention:year1   0.09134    0.03887   2.350  0.01878 *
```

```
## year0:fluency   -0.03068    0.02030  -1.511  0.13080
## year1:fluency    0.02275    0.01686   1.349  0.17727
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(3.3109) family taken to be 1)
##
##     Null deviance: 677.48  on 184  degrees of freedom
## Residual deviance: 197.98  on 179  degrees of freedom
## AIC: 1295.6
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  3.311
##           Std. Err.:  0.438
##
##  2 x log-likelihood:  -1281.552
```

```
par(mfrow = c(2, 2), mar = c(4.3, 4.3, 2, 1.2))
plot(fit33)
```




```
#Confidence interval:
confint(fit33)
```

```
## Waiting for profiling to be done...

##                     2.5 %       97.5 %
## (Intercept)     -0.255941141  1.636021008
```

```
## attention        0.034874442  0.152452048
## year1           -2.737011768 -0.256071539
## attention:year1  0.015350528  0.166855793
## year0:fluency   -0.069025923  0.007502342
## year1:fluency   -0.008496873  0.053852162
```

```r
residuals(stepsearch, type="pearson")
```

```
##           1           2           3           4           5           6
## -1.14900486 -0.44176484 -0.41481563 -0.24943073  0.70883080 -1.05449149
##           7           8           9          10          11          12
##  0.94759778 -0.29387504 -0.26899068  0.43145534 -0.41800017  4.01659478
##          13          14          15          16          17          18
## -0.33031666 -1.31447966  0.29250152 -0.41179478 -0.70506224  0.39784936
##          19          20          21          22          23          24
## -0.43567854  0.37747242 -0.90928589  0.28871139 -0.54682398  0.90418393
##          25          26          27          28          29          30
## -0.73701632  0.36451983 -1.12169551  2.77032690 -1.37930614  1.92684862
##          31          32          33          34          35          36
## -1.10909688 -0.22969336 -0.17546122  1.05981433  0.50198743  0.09478347
##          37          38          39          40          41          42
##  2.23519314 -0.15802243  0.98469045 -0.84057929 -0.29387504  1.53594031
##          43          44          45          46          47          48
## -1.00863145 -0.49644425 -0.44176484  1.77389304 -0.60582049  0.84286544
##          49          50          51          52          53          54
## -1.40853090  0.72396732 -0.09450991 -0.96122589 -0.53024257 -1.46224545
##          55          56          57          58          59          60
## -0.98873939 -0.51334829  1.06169858  0.50753344 -1.02956845 -0.65488696
##          61          62          63          64          65          66
##  1.07513684 -0.81444350  0.45884346 -0.78123015  1.11819884  0.19591255
##          67          68          69          70          71          72
## -1.41417636  1.06990421  1.72039031 -1.23236685  0.08995651 -0.19482978
##          73          74          75          76          77          78
## -1.00884336 -0.70236683  0.26913990  0.47267539 -1.25784368 -0.34828817
##          79          80          81          82          83          84
## -0.24308874 -0.74859530  0.22963746 -0.63288008 -1.14434936  0.65030009
##          85          86          87          88          89          90
## -0.24943073 -0.36609433 -0.76677102  1.71940260  0.83860248  2.49539380
##          91          92          93          94          95          96
##  0.49619484 -0.43356663  1.83120425  0.84139753 -0.13225215 -1.48361021
##          97          98          99         100         101         102
## -1.19808931  0.43847656 -1.10379939  0.49536130  0.65532134 -1.28560427
##         103         104         105         106         107         108
##  0.92939938  2.79179092 -1.37951953  0.10263150  0.50488774 -0.70923964
##         109         110         111         112         113         114
## -1.16727408 -0.22317419  0.73212436 -0.79529103 -0.86742456  0.39654081
##         115         116         117         118         119         120
##  1.00535727 -0.65114134 -0.76702792  0.47205203 -0.50533707  0.55659136
##         121         122         123         124         125         126
##  1.24444869  0.10145987 -0.61969077 -0.04861215  0.60419603  0.60404396
##         127         128         129         130         131         132
##  1.86492952 -1.33124762 -0.76590484 -0.03296666  0.20303581  0.22139390
##         133         134         135         136         137         138
## -0.73224908 -0.68916732  0.15515828  0.04067933 -0.78291270  0.73943124
##         139         140         141         142         143         144
```

```
## -0.04044959  0.53267394 -0.06892392  1.23993895 -1.21025119 -0.85722616
##         145         146         147         148         149         150
## -1.59766970 -0.73535127 -0.10120170 -0.70713163  1.62636508  2.03425211
##         151         152         153         154         155         156
##  0.93921032 -0.52580754  0.57753677  1.74695627 -0.62622148  1.09624604
##         157         158         159         160         161         162
##  1.37550344 -0.54171859 -0.37639371  0.61985822  0.30899017 -0.58061686
##         163         164         165         166         167         168
##  0.17919043 -0.21380730 -1.25247785 -0.75628708  2.91057987 -0.11633616
##         169         170         171         172         173         174
## -0.31309238 -1.05507989 -1.16973322  1.05408086 -0.98597571 -0.30948612
##         175         176         177         178         179         180
##  0.03009480  0.55105663  0.01811632 -0.91258194 -0.16691451 -0.37909601
##         181         182         183         184         185
## -0.42227088 -0.99003027 -0.71622672  0.97414036 -0.50246152
```

```
residuals(stepsearch, type="deviance")
```

```
##           1           2           3           4           5           6
## -1.59509853 -0.48538238 -0.45156451 -0.26204122  0.63245368 -1.39054737
##           7           8           9          10          11          12
##  0.81920987 -0.31185242 -0.28353113  0.40139137 -0.45560617  2.62365739
##          13          14          15          16          17          18
## -0.35331649 -1.96278204  0.27789395 -0.44859554 -0.82830824  0.37200828
##          19          20          21          22          23          24
## -0.47723571  0.35394761 -1.13194363  0.27466161 -0.61448213  0.78615952
##          25          26          27          28          29          30
## -0.87784352  0.34250627 -1.51717170  1.98516822 -2.36344617  1.49153801
##          31          32          33          34          35          36
## -1.51581868 -0.24031593 -0.18147330  0.90346395  0.46194693  0.09317804
##          37          38          39          40          41          42
##  1.68245667 -0.16289527  0.84850370 -1.02292638 -0.31185242  1.23742839
##          43          44          45          46          47          48
## -1.30845137 -0.55264360 -0.48538238  1.39264424 -0.69313109  0.73815277
##          49          50          51          52          53          54
## -2.44294082  0.64457588 -0.09623098 -1.22685843 -0.59305257 -2.59922435
##          55          56          57          58          59          60
## -1.27382737 -0.57218367  0.90446539  0.46592675 -1.33814450 -0.75733651
##          61          62          63          64          65          66
##  0.91583842 -0.98478541  0.42442063 -0.93347272  0.94528935  0.18918360
##          67          68          69          70          71          72
## -2.45871455  0.91050966  1.36131301 -1.77155542  0.08850253 -0.20238409
##          73          74          75          76          77          78
## -1.30166744 -0.82266812  0.25691591  0.43627276 -1.82883393 -0.37402035
##          79          80          81          82          83          84
## -0.25518075 -0.89020928  0.22053443 -0.72921575 -1.58568975  0.58535657
##          85          86          87          88          89          90
## -0.26204122 -0.39470468 -0.91656331  1.35771697  0.73428128  1.83159497
##          91          92          93          94          95          96
##  0.45658580 -0.47545660  1.43584868  0.73922489 -0.13561934 -2.37883017
##          97          98          99         100         101         102
## -1.65101566  0.40772512 -1.46374434  0.45659618  0.59047925 -1.89328451
##         103         104         105         106         107         108
##  0.80695199  2.00496387 -2.07807635  0.10076891  0.46479407 -0.82865927
##         109         110         111         112         113         114
```

```
## -1.58685066 -0.23290078  0.65273637 -0.95182895 -1.06061873  0.37116306
##         115         116         117         118         119         120
##  0.86469104 -0.74932242 -0.91070427  0.43682174 -0.56085715  0.50868098
##         121         122         123         124         125         126
##  1.03975696  0.09964914 -0.70738650 -0.04904687  0.54800567  0.54828783
##         127         128         129         130         131         132
##  1.45740920 -1.97592634 -0.90961824 -0.03316558  0.19598885  0.21306246
##         133         134         135         136         137         138
## -0.86151908 -0.80178064  0.15099924  0.04038340 -0.93797695  0.65863382
##         139         140         141         142         143         144
## -0.04075237  0.48852551 -0.06980407  1.03675507 -1.67776987 -1.04503482
##         145         146         147         148         149         150
## -2.80266296 -0.86554646 -0.10311942 -0.82762332  1.30215188  1.56350913
##         151         152         153         154         155         156
##  0.81475487 -0.58639852  0.52615320  1.38086535 -0.71620725  0.93232591
##         157         158         159         160         161         162
##  1.13137650 -0.60699760 -0.40567017  0.56113891  0.29317995 -0.65672086
##         163         164         165         166         167         168
##  0.17368457 -0.22274196 -1.76925303 -0.89567752  2.07101969 -0.11888340
##         169         170         171         172         173         174
## -0.33288945 -1.37323186 -1.59407559  0.90120623 -1.25250339 -0.32884326
##         175         176         177         178         179         180
##  0.02993185  0.50393669  0.01805677 -1.13344113 -0.17224545 -0.40878439
##         181         182         183         184         185
## -0.46003105 -1.26141917 -0.83842606  0.84132536 -0.55722103
```

```
cooks.distance(stepsearch)
```

```
##            1            2            3            4            5            6
## 5.394586e-03 1.161619e-03 7.341794e-04 2.121207e-04 1.326503e-03 1.072670e-02
##            7            8            9           10           11           12
## 1.916657e-03 4.500492e-04 6.975863e-04 9.108582e-04 5.740074e-04 1.419194e-01
##           13           14           15           16           17           18
## 3.589967e-04 5.084574e-03 4.000796e-04 4.859426e-04 3.368494e-03 9.807536e-04
##           19           20           21           22           23           24
## 9.886912e-04 1.058323e-03 7.197925e-03 3.258396e-04 3.416048e-03 2.582352e-03
##           25           26           27           28           29           30
## 4.317908e-03 2.569880e-04 8.604615e-03 1.108627e-01 1.059952e-02 3.453164e-02
##           31           32           33           34           35           36
## 5.767724e-03 1.391871e-04 9.905349e-05 2.172352e-03 6.588042e-04 1.009927e-04
##           37           38           39           40           41           42
## 2.514460e-02 6.149059e-05 1.252162e-02 9.124698e-03 4.500492e-04 1.760087e-02
##           43           44           45           46           47           48
## 1.931771e-03 1.793228e-03 1.161619e-03 1.133597e-02 1.207584e-03 4.117329e-03
##           49           50           51           52           53           54
## 7.875289e-03 5.401975e-03 4.252146e-05 4.117381e-03 1.687022e-03 1.448843e-02
##           55           56           57           58           59           60
## 9.092549e-03 6.889614e-04 7.638065e-03 1.346366e-03 8.157499e-03 2.121215e-03
##           61           62           63           64           65           66
## 8.838945e-03 4.020946e-03 4.617087e-03 1.005974e-02 6.515874e-03 1.078601e-04
##           67           68           69           70           71           72
## 7.204621e-03 1.331065e-02 7.737927e-03 3.814796e-03 1.819682e-05 1.295276e-04
##           73           74           75           76           77           78
## 9.271279e-03 2.127162e-03 5.377026e-04 3.097801e-03 3.391610e-03 3.408895e-04
##           79           80           81           82           83           84
```

```
## 2.132756e-04 1.196163e-03 3.374194e-04 1.906761e-03 9.624082e-03 8.030048e-04
##           85           86           87           88           89           90
## 2.121207e-04 3.538416e-04 1.552475e-03 1.382437e-02 3.545400e-03 1.642784e-02
##           91           92           93           94           95           96
## 8.100912e-04 9.261795e-04 1.421173e-02 5.468458e-03 2.609865e-04 2.074585e-02
##           97           98           99          100          101          102
## 6.613965e-03 7.116873e-04 3.040318e-03 2.230156e-03 2.185460e-03 3.813737e-02
##          103          104          105          106          107          108
## 9.405443e-03 8.058202e-02 1.859545e-02 6.227245e-05 8.449893e-04 4.010728e-03
##          109          110          111          112          113          114
## 4.587870e-03 1.472158e-04 1.337545e-03 3.712104e-03 7.387920e-03 7.417056e-04
##          115          116          117          118          119          120
## 4.263496e-03 9.940337e-04 1.498934e-03 9.443939e-04 1.204531e-03 1.566931e-03
##          121          122          123          124          125          126
## 3.820805e-03 4.057576e-05 8.258774e-04 1.884200e-05 2.471082e-03 4.036689e-03
##          127          128          129          130          131          132
## 2.773078e-02 1.367098e-02 2.312212e-03 6.553068e-06 1.387246e-04 1.871029e-04
##          133          134          135          136          137          138
## 2.756021e-03 1.574385e-03 6.959979e-05 2.126580e-05 7.368912e-03 1.098975e-02
##          139          140          141          142          143          144
## 9.673038e-06 1.372993e-03 2.590201e-05 2.158905e-02 4.375780e-03 2.065010e-03
##          145          146          147          148          149          150
## 2.035219e-02 1.519215e-03 4.529834e-05 3.610415e-03 1.359568e-02 3.054182e-02
##          151          152          153          154          155          156
## 6.420914e-03 1.507465e-03 8.229248e-04 1.905381e-02 1.654173e-03 2.964947e-03
##          157          158          159          160          161          162
## 8.970451e-03 1.619645e-03 3.321508e-04 2.113776e-03 6.371917e-04 1.117483e-03
##          163          164          165          166          167          168
## 2.560158e-04 1.538343e-04 4.958566e-03 1.411153e-03 5.498997e-02 5.730635e-05
##          169          170          171          172          173          174
## 2.371929e-04 7.225953e-03 2.940453e-03 7.423145e-03 5.861749e-03 5.779437e-04
##          175          176          177          178          179          180
## 7.765397e-06 8.172498e-04 1.670317e-06 3.082764e-03 1.704180e-04 1.316268e-03
##          181          182          183          184          185
## 2.597505e-03 7.733864e-03 3.010711e-03 1.049860e-02 2.478931e-03
```

```r
rstandard(stepsearch, type="pearson")
```

```
##           1           2           3           4           5           6
## -1.16488277 -0.45052670 -0.42079247 -0.25232260  0.71523499 -1.08745399
##           7           8           9          10          11          12
##  0.95454882 -0.29900778 -0.27739448  0.43854889 -0.42267459  4.13180823
##          13          14          15          16          17          18
## -0.33401604 -1.32768384  0.29710533 -0.41582558 -0.72088055  0.40604797
##          19          20          21          22          23          24
## -0.44328452  0.38670921 -0.93511564  0.29253364 -0.56680887  0.91391582
##          25          26          27          28          29          30
## -0.75624147  0.36694674 -1.14708177  2.89568842 -1.40498905  1.98506830
##          31          32          33          34          35          36
## -1.12659897 -0.23176710 -0.17738394  1.06687037  0.50647959  0.09819656
##          37          38          39          40          41          42
##  2.27294997 -0.15935604  1.02494708 -0.87494429 -0.29900778  1.57368347
##          43          44          45          46          47          48
## -1.01522642 -0.50835815 -0.45052670  1.79558967 -0.61260538  0.85916263
##          49          50          51          52          53          54
```

```
## -1.42745724  0.74802998 -0.09602326 -0.97566894 -0.54084016 -1.49505141
##          55          56          57          58          59          60
## -1.01861413 -0.51794211  1.08551815  0.51642335 -1.05561839 -0.66576582
##          61          62          63          64          65          66
##  1.10217658 -0.83088039  0.48888354 -0.82102268  1.13772901  0.19779397
##          67          68          69          70          71          72
## -1.43147327  1.10964632  1.73578562 -1.24297141  0.09065102 -0.19709048
##          73          74          75          76          77          78
## -1.03874048 -0.71259042  0.27572203  0.49328726 -1.26710933 -0.35163290
##          79          80          81          82          83          84
## -0.24606735 -0.75408657  0.23451661 -0.64301409 -1.17207284  0.65455210
##          85          86          87          88          89          90
## -0.25232260 -0.36940194 -0.77369979  1.74646503  0.85279126  2.51792304
##          91          92          93          94          95          96
##  0.50175197 -0.44074243  1.85741835  0.86276111 -0.13841421 -1.52899542
##          97          98          99         100         101         102
## -1.21668060  0.44398278 -1.11323673  0.51000982  0.66650956 -1.37355282
##         103         104         105         106         107         108
##  0.96189794  2.88485106 -1.42315907  0.10465394  0.51058332 -0.72779309
##         109         110         111         112         113         114
## -1.18064415 -0.22542571  0.73838393 -0.81085426 -0.89498849  0.40283445
##         115         116         117         118         119         120
##  1.01968376 -0.65637846 -0.77372064  0.47880956 -0.51335744  0.56603833
##         121         122         123         124         125         126
##  1.25497075  0.10281393 -0.62427020 -0.04988383  0.61773808  0.62547782
##         127         128         129         130         131         132
##  1.91371540 -1.36500628 -0.77612640 -0.03362862  0.20536007  0.22425819
##         133         134         135         136         137         138
## -0.74487085 -0.69694174  0.15669017  0.04233500 -0.81289587  0.78430585
##         139         140         141         142         143         144
## -0.04124669  0.54133842 -0.07018108  1.29463233 -1.22258878 -0.86545831
##         145         146         147         148         149         150
## -1.63946416 -0.74241141 -0.10271135 -0.72397894  1.65439873  2.08373257
##         151         152         153         154         155         156
##  0.96175860 -0.53539813  0.58241995  1.78321724 -0.63514523  1.10551501
##         157         158         159         160         161         162
##  1.39744291 -0.55171498 -0.37942104  0.63126136  0.31582329 -0.58716674
##         163         164         165         166         167         168
##  0.18387798 -0.21625486 -1.26596798 -0.76268163  2.97327152 -0.11800006
##         169         170         171         172         173         174
## -0.31568970 -1.07780550 -1.17837104  1.07741456 -1.00577376 -0.31570470
##         175         176         177         178         179         180
##  0.03093758  0.55612972  0.01842563 -0.92404178 -0.17031224 -0.39038747
##         181         182         183         184         185
## -0.44152465 -1.01567593 -0.73024273  1.00870670 -0.51842810
```

```
rstandard(stepsearch, type="deviance")
```

```
##           1           2           3           4           5           6
## -1.61714093 -0.49500934 -0.45807085 -0.26507929  0.63816781 -1.43401470
##           7           8           9          10          11          12
##  0.82521914 -0.31729916 -0.29238920  0.40799063 -0.46070114  2.69891532
##          13          14          15          16          17          18
## -0.35727346 -1.98249853  0.28226784 -0.45298656 -0.84689161  0.37967437
##          19          20          21          22          23          24
```

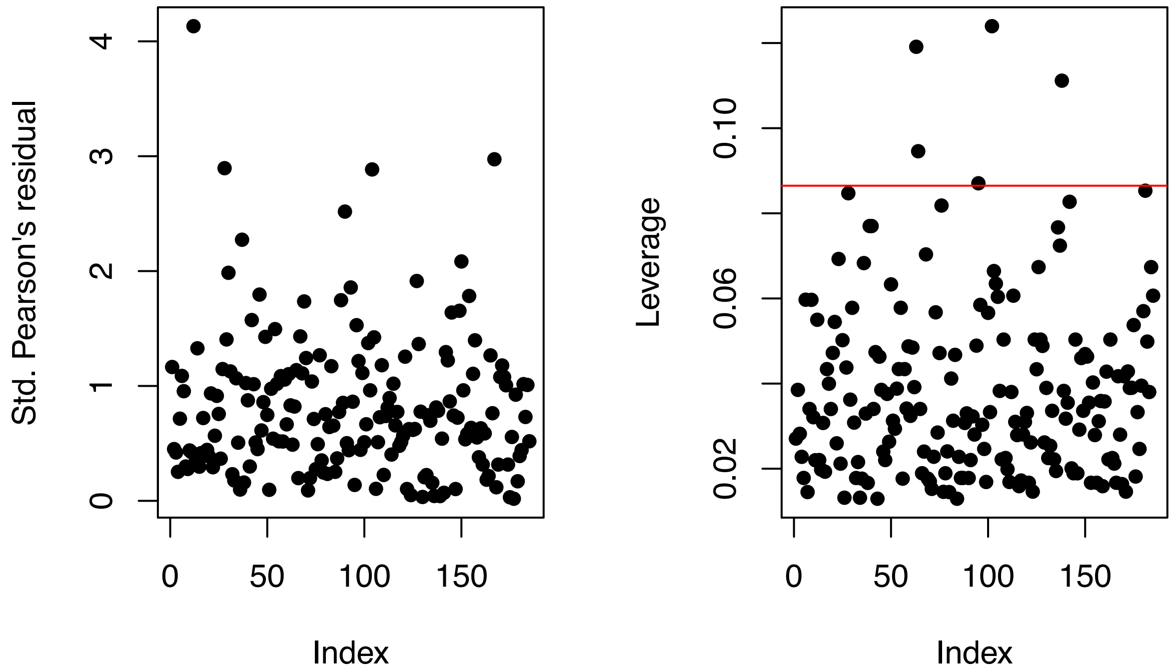
```
## -0.48556719  0.36260874 -1.16409833  0.27829786 -0.63693974  0.79462108
##          25          26          27          28          29          30
## -0.90074217  0.34478662 -1.55150840  2.07500011 -2.40745394  1.53660479
##          31          32          33          34          35          36
## -1.53973904 -0.24248558 -0.18346190  0.90947904  0.46608077  0.09653333
##          37          38          39          40          41          42
##  1.71087669 -0.16427001  0.88319267 -1.06474620 -0.31729916  1.26783612
##          43          44          45          46          47          48
## -1.31700672 -0.56590621 -0.49500934  1.40967778 -0.70089381  0.75242528
##          49          50          51          52          53          54
## -2.47576647  0.66599979 -0.09777189 -1.24529279 -0.60490550 -2.65753882
##          55          56          57          58          59          60
## -1.31231604 -0.57730399  0.92475738  0.47408787 -1.37200197 -0.76991725
##          61          62          63          64          65          66
##  0.93887180 -1.00466010  0.45220707 -0.98101984  0.96179953  0.19100040
##          67          68          69          70          71          72
## -2.48878729  0.94433098  1.37349503 -1.78679972  0.08918582 -0.20473244
##          73          74          75          76          77          78
## -1.34024242 -0.83464280  0.26319909  0.45529722 -1.84230566 -0.37761220
##          79          80          81          82          83          84
## -0.25830752 -0.89673936  0.22522017 -0.74089234 -1.62410533  0.58918394
##          85          86          87          88          89          90
## -0.26507929 -0.39827078 -0.92484564  1.37908667  0.74670500  1.84813122
##          91          92          93          94          95          96
##  0.46169933 -0.48332571  1.45640318  0.75799425 -0.14193829 -2.45160111
##          97          98          99         100         101         102
## -1.67663520  0.41284517 -1.47625916  0.47009836  0.60056042 -2.02280463
##         103         104         105         106         107         108
##  0.83516891  2.07179632 -2.14381394  0.10275464  0.47003736 -0.85033670
##         109         110         111         112         113         114
## -1.60502661 -0.23525042  0.65831718 -0.97045551 -1.09432174  0.37705391
##         115         116         117         118         119         120
##  0.87701302 -0.75534921 -0.91865064  0.44307494 -0.56975871  0.51731477
##         121         122         123         124         125         126
##  1.04854830  0.10097903 -0.71261399 -0.05032991  0.56028831  0.56774325
##         127         128         129         130         131         132
##  1.49553449 -2.02603319 -0.92175776 -0.03383153  0.19823244  0.21581896
##         133         134         135         136         137         138
## -0.87636909 -0.81082544  0.15249007  0.04202703 -0.97389860  0.69860500
##         139         140         141         142         143         144
## -0.04155543  0.49647187 -0.07107727  1.08248607 -1.69487345 -1.05507053
##         145         146         147         148         149         150
## -2.87597960 -0.87385660 -0.10465767 -0.84734133  1.32459708  1.60153940
##         151         152         153         154         155         156
##  0.83431526 -0.59709428  0.53060193  1.40952750 -0.72641331  0.94020891
##         157         158         159         160         161         162
##  1.14942211 -0.61819860 -0.40893297  0.57146183  0.29966343 -0.66412925
##         163         164         165         166         167         168
##  0.17822810 -0.22529180 -1.78830922 -0.90325065  2.11562786 -0.12058373
##         169         170         171         172         173         174
## -0.33565099 -1.40281022 -1.60584694  0.92115583 -1.27765322 -0.33545078
##         175         176         177         178         179         180
##  0.03077006  0.50857599  0.01836506 -1.14767443 -0.17575169 -0.42096012
##         181         182         183         184         185
```

```
## -0.48100653 -1.29409487 -0.85483342  0.87117890 -0.57492770
```

```
par(mfrow=c(1,2))
plot(abs(rstandard(stepsearch, type="pearson")), xlab="Index", ylab="Std. Pearson's residual", pch=16)
plot(influence(stepsearch)$hat, xlab="Index", ylab="Leverage", pch=16)
l_threshold <- 2*8 / 185
l_threshold
```

```
## [1] 0.08648649
```

```
abline(h=l_threshold, col="red")
```




```
order(abs(rstandard(stepsearch, type="pearson")), decreasing = TRUE)[1:5]
```

```
## [1]  12 167  28 104  90
```

```
order(influence(stepsearch)$hat, decreasing=TRUE)[1:5]
```

```
## [1] 102  63 138  64  95
```

5. Analysis of best model

- estimates of model parameters
- confidence intervals
- limitations
- experimental design