Statistical Modelling 2 Coursework

Jonathan Cheung

March 2021

1 Introduction

We will analyse the results from an experiment that measured the reading accuracy (count of words pronounced correctly before the third error) for children across two year groups, who were each given a score for their attention and their verbal fluency. In this assignment we will fit different models to this data to try and find a relationship between the reading accuracy (count) and the cognitive tests that the children took.

2 Exploratory Analysis

First we will plot some histograms (figure 1) of each of the variables to see the distributions of our variables.

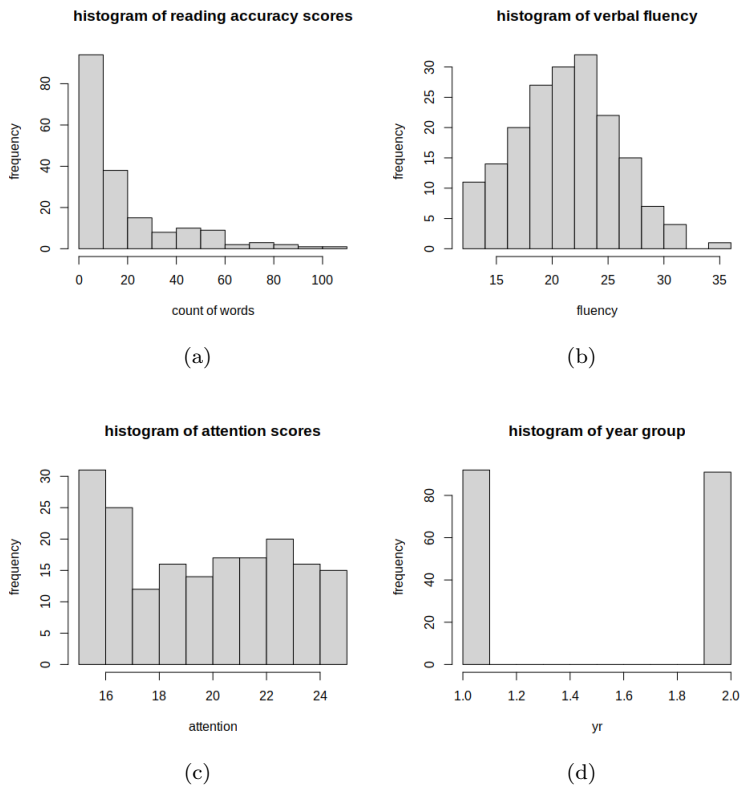


Figure 1: Histograms of the four variables

The attention and year variables as fairly balanced, thought there is a slight skew to the smaller attention scores. Though, count is skewed towards smaller values and fluency has some gaussian shape centred around 23 with some positive skew.

We plot a scatterplot (figure 2) of each variable against the other. There is some correlation (0.64) between attention and fluency. This might suggest some collinearity in the data. The range of count values is much larger (with much larger maximum value) for one year group. This suggests that we might notice some benefits for fitting model that will fit the two groups separately. The variance of count scores is 23.66 and 544.6 for year groups 0 and 1. The mean of count scores is 6.54 and 29.98. There is also a weak correlation between count and attention (0.42) and slightly weaker correlation between count and fluency (0.33).

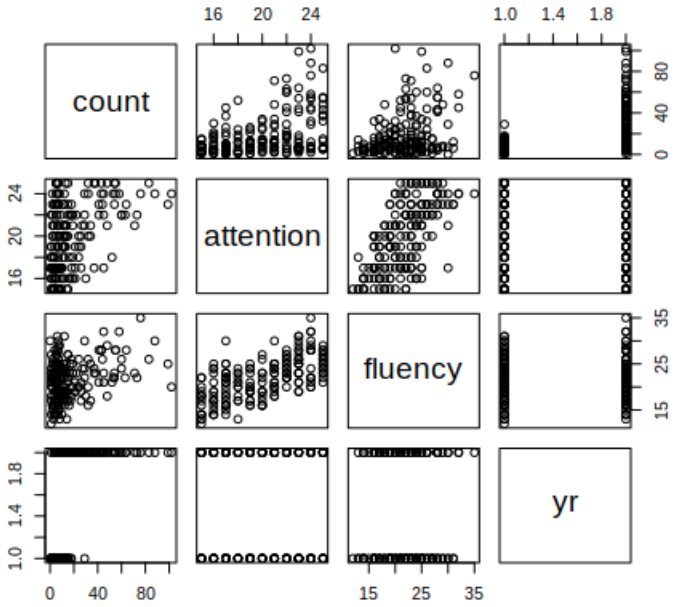


Figure 2: scatterplot matrix

3 The Experimental team's model

Now, we will fit the model suggested by the experimental team. They have gone for a linear model

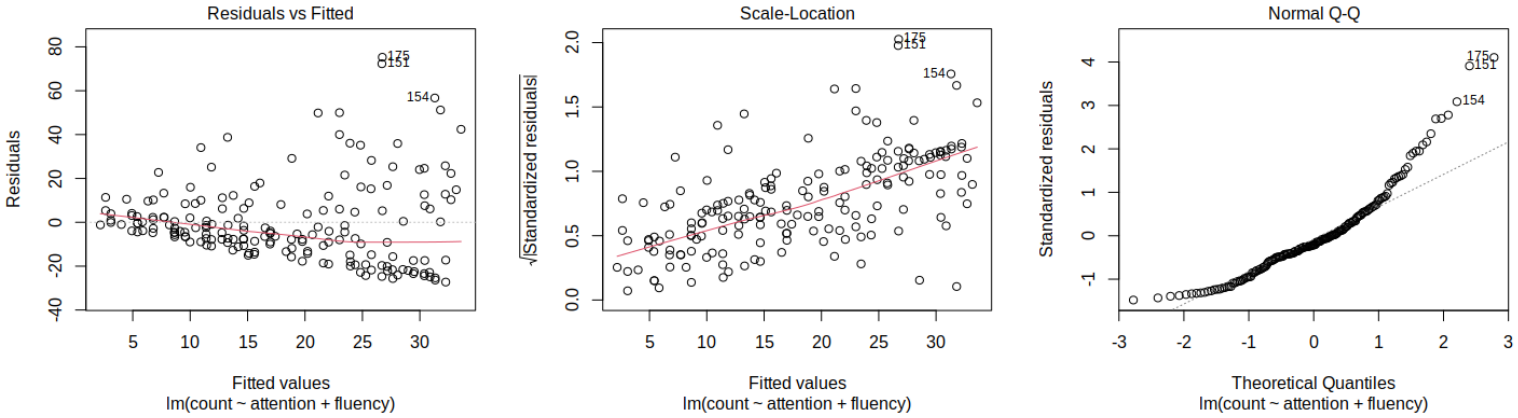
$$y_i = \beta_0 + \beta_1 a_i + \beta_2 f_i$$

Where y_i is the count of words, a_i is the attention score, and f_i is the verbal fluency.

We fit this model using the lm() function in R, which solves the normal equations to find our β coefficients.

Looking at the summary statistics, we have that the median residual is -3.923; the model is on average over-predicting the performance of the childrens' reading accuracy. We also have some very large residuals where we have hugely under-predicted the performance (see points 175, 154 and 151 in figure 3a).

We see signs of heteroscedastic behaviour; the standardised residuals vary more as the fitted values increase (fig 3b). This suggests that we have not modelled the mean variance relationship well. There is also some non linear behaviour; the line of best fit is not horizontal. This linear model assumes residuals are normally distributed. In the qq-plot (fig 3c), it is clear our residuals do not appear to be normally distributed



(a) (b) (c)

Figure 3: Diagnostic plots for the linear model

4 An alternative model

We have been suggested to use a poisson glm with log link function as an alternative model. This might work better than the last linear model since this model will not assume that the errors are normally distribute

There is code in the appendix for this. Our method will be the iterated weighted least squares algorithm. This method makes an initial guess of the $\hat{\boldsymbol{\beta}}^{(\mathbf{0})}$. Then, it forms the linear predictor and applies the link function to estimate $\boldsymbol{\mu}^{(\mathbf{0})}$. Then we calculate the adjusted dependent variable and weights. Using these weights, we solve the maximum likelihood equation for this weighted least squares problem to give us new $\hat{\boldsymbol{\beta}}^{(\boldsymbol{i})}$ coefficients, which can help us make new predictions for $\boldsymbol{\mu}^{(\boldsymbol{i})}$. A new deviance score is calculated, which we compare to the old deviance to monitor how well our deviance is improving. We iterate these steps of finding new $\hat{\boldsymbol{\beta}}^{(\boldsymbol{i})}$ and $\boldsymbol{\mu}^{(\boldsymbol{i})}$ until our change $\frac{|D_new - D_old|}{|D_new + 0.1|}$ is less than 1e-8.

We also derive the parts of this algorithm that are specific to this poisson exponential family and log link function:

From the exponential family of the poisson distribution, we know. $\theta = log(\lambda_i), b(\theta_i) = exp(\theta_i)$

Our link function is the log function, so $\eta_i = log(\mu_i), \frac{\delta \eta_i}{\delta \mu_i} = \frac{1}{\mu_i}, \mu_i = E(Y_i) = \lambda_i$

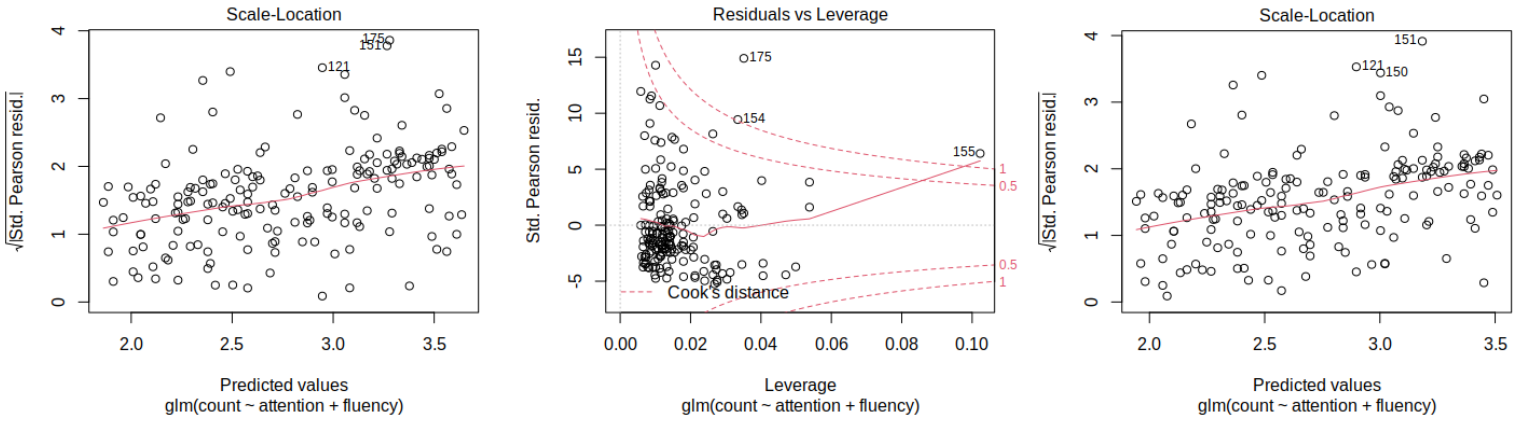
Using this, we derive $b'(\theta_i) = b''(\theta_i) = exp(\theta_i)$

$\mathrm{V}(\mu_i) = b''(\theta_i) = exp(\theta_i) = \lambda_i$ so $w_i i^- 1 = \frac{\delta \eta_i}{\delta \mu_i}^2 V(\mu_i) = \mu_i \cdot \frac{1}{\mu_i^2} = \frac{1}{\mu_i}$

The model still does not capture the mean-variance relationship; we can still see heteroscedascity and non linearity (fig 4a). We even refitted the model by removing the points 154, 155, 175 since fig 4b shows that these have a Cook's distance all over one and so we believe they could be having a detrimental effect on the fit. The newly fitted poisson GLM still did not capture the mean variance relationship (fig 4c). We can see that the standardised pearson residual is increasing as the predicted values increases. A χ^2_{n-p} significance test also returns 0 (we believe the value was smaller than machine precision hence this number) and suggests that the model is not fitting well at all, assuming the poisson response distribution is correct.

The poisson GLM, while being a slight improvement over the linear model, is still too restrictive. It only has one parameter and this forces the mean to be equal to the variance. We even tried fitting poisson GLMs with a multiplicative year group term(count yr*attention + yr*fluency). While this showed an improvement in the non-linearity, these models still showed some signs of heteroscedascity. In the next section, we will

use a different exponential family.



(a) (b) (c)

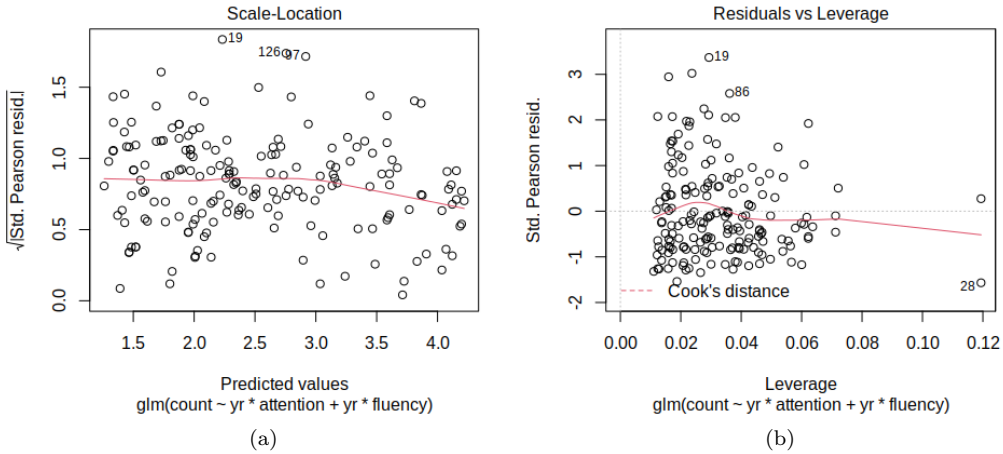
Figure 4: Effect of removing outliers in poisson generalised linear model

5 Our proposed general linear model

Since the poisson distribution's single parameter appeared to be too restrictive, the negative binomial distribution is a sensible exponential family to try. This has an additional parameter and so allows more flexibility for the model and the mean variance relationship.

Unlike the last two models, we will also try and make use of 'yr' the year group variable since we noticed a difference in the distribution of count across the two groups.

Fitting the negative binomial GLM with log link function onto the whole dataset using the formula count yr*attention + yr*fluency will allow us to plot figure 5. Fig 5a shows an improvement upon the poisson GLM. The residuals look like noise now. Fig 5b shows a couple of points that have high leverage. However their residuals are quite small so they will not affect the fit that much. There are also some points with larger residuals (e.g 19), however this point has very small leverage. Overall, our data points have small Cooke's distances; the 0.5 line does not even show up on this scale. So there are no points that we believe to be outliers.



(a) (b)

Figure 5: Negative binomial GLM with log link on full data

We used the ANOVA test to compare this model to one where count yr + attention + fluency. This gave us a p value of 0.00557 (log likelihood test) for our more complex model with the multiplicative terms. Thus, we conclude that utilising the year group variable was a good idea.

Coefficients are:

intercept	yr	attention	fluency	yr*attention	yr*fluency
0.00743702	-0.44664986	0.10981312	-0.01734761	0.05243734	0.03805635

Confidence intervals for these coefficients are:

	2.5%	97.5%
(Intercept)	-0.91450492	0.92691423
yr	-1.64552415	0.75466860
attention	0.05296151	0.16720903
fluency	-0.05735601	0.02269664
yr*attention	-0.02274478	0.12746751
yr*fluency	-0.01506261	0.09134005

6 Standalone report for the data collection team

The linear model suggested by the experimental team was not very suitable for this data. It assumed that the difference between the predicted values and true values would be normally distributed, which they not. Using a poisson GLM was a bit better because our data was a count and a poisson distribution would fit this better than a normal distribution. However, the poisson GLM had an issue in that it assumed the variance is equal to the mean. The best model we found used the negative binomial distribution. This distribution represents the number of successes until a number of failures, which is precisely what the reading accuracy test was. Just like the poisson GLM, it allowed the variance to be non constant, but it goes one step further and allows the variance to take on values other than the mean. In short, this is a very flexible model that deals with changing variance very well. For our model we also utilised the year group data since we found that there was a significant different between the two year groups and this had to be factored in. Year group 1 appears to be an older group that would achieve much higher scores and a wider range of scores.

It appears that there are some children who scored very highly on the reading test and perhaps additional information such as another test could be collected in addition to fluency and attention so they can be modelled better.

To conclude, we found a relationship between our variables and the reading accuracy (Y). Since we modelled using a negative binomial with log link, our coefficients can be interpreted as: the mean of count increases by a multiple of $\exp(\beta)$, when the respective X variable increases by 1. Looking at the coefficients from our best model, we have that a higher attention score will result in a higher reading accuracy, and especially so if the child is in year group 1. We have a negative coefficient for fluency, but positive for yr*fluency. So our model claims that children in year group 0 on average do not read more accurately if they have a higher fluency. However in year group 1, an increase in fluency will lead to an overall increase in reading accuracy.

7 Mastery material: Dilution Assay

Pages 11-12 in Generalized Linear Models by McCullagh and Nelder describes a use of a GLM. A solution with organisms is diluted multiple times and at each dilution agar plates are streaked and we record the proportion that were infected. Note that the volume of solution used for the streak is negligible compared to the change in volume when we dilute.

We can fit a binomial GLM with a complementary log log link function to this data. This GLM will estimate two coefficients and it is the intercept α we are interested in. $\alpha = log(v)+log(p_0)$ (see aforementioned book for more details) and so we can estimate the initial density of organisms as follows.

$$p_0 = exp(\alpha - log(v))$$

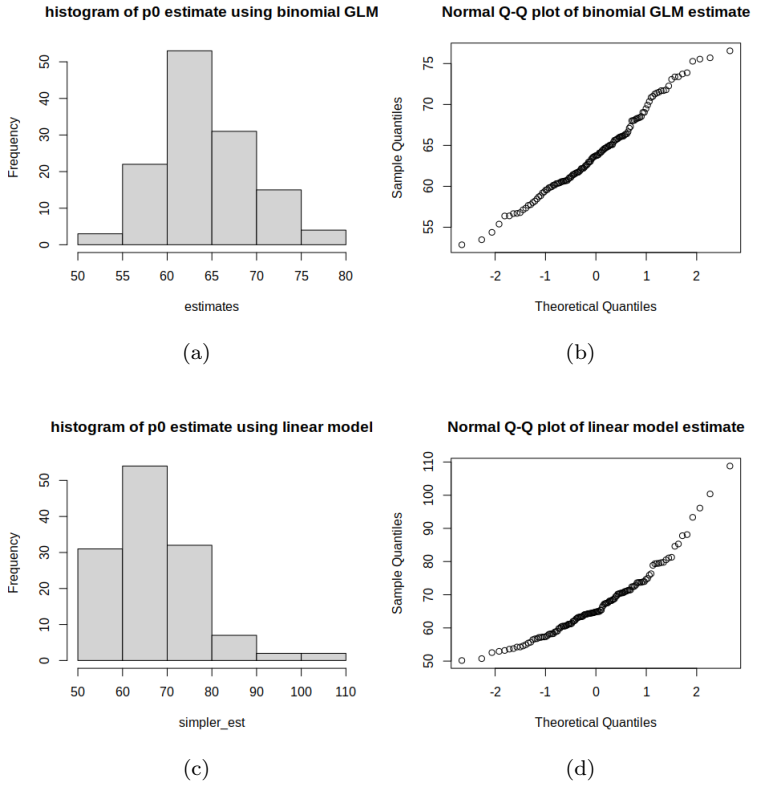
. The R code simulates data with starting density $p_0 = 64$, streak volume $v = 0.1$, number of dilutions $x = 16$, and number of agar plates per dilution $n = 256$.

We simulate data 256 times and fit the GLM and use the above equation to estimate p_0 each time. The distribution of the estimates of p_0 can be visualised (figure 6). We have also fitted a simple linear regression on $log(-log(1 - y_x))$ against x, where y_x is the proportion of infected plates at dilution x.

The histogram for the binomial method is a bit more symmetric than the linear model method. The qq plots show that the estimates from the binomial GLM are closer to being normally distributed. Both methods appear to produce accurate estimates close to 64 (table 1). The binomial has a marginally better mean estimate and a significantly smaller variance.

	mean	median	variance
Binomial GLM	64.12615	63.74591	25.88536
Simple LM	66.96122	64.81027	101.3878

Table 1: Comparison of estimates.



(a) (b)

(c) (d)

Figure 6: Plots of sampling distribution of initial density estimate

8 Appendix

```
load("01063446.RData")
library(MASS)
#histograms
hist(as.numeric(read[[1]]), ylab = "frequency", xlab="count of words")
hist(as.numeric(read[[2]]), ylab = "frequency", xlab="attention")
hist(as.numeric(read[[3]]), ylab = "frequency", xlab="fluency")
hist(as.numeric(read[[4]]), ylab = "frequency", xlab="yr")

pairs(read) #correlation plot
cor(read$count,read$attention) #calculate correlations between variables
cor(read$count,read$fluency)
cor(read$fluency,read$attention)


#q3 fit linear model
fit3 = lm(count ~ attention + fluency, data=read)
summary(fit3)
plot(fit3)

#poisson with log link q4
fit4 = glm(count ~ attention + fluency, family='poisson', data = read)
summary(fit4)
plot(fit4)
pchisq(deviance(fit4b), df.residual(fit4b),lower=FALSE) #tiny p value
#remove outliers
fit4b = glm(count ~attention+fluency, family='poisson', data = read[-c(154,155,175),])
plot(fit4b)

#poisson with split on year group
fit_yg = glm(count ~ yr*attention + yr*fluency, family='poisson', data=read)
summary(fit_yg)
plot(fit_yg)
pchisq(deviance(fit_yg), df.residual(fit_yg),lower=FALSE) #tiny p value

#negative binomial
fit_bin = glm.nb(count~ yr+attention+fluency+yr*attention +yr*fluency, data=read)
summary(fit_bin)
plot(fit_bin)
pchisq(deviance(fit_bin), df.residual(fit_bin),lower=FALSE)

drop1(fit_bin, test="Chisq")
fit_bin2 = glm.nb(count~., data=read) #no yr group multiply
plot(fit_bin2)
anova(fit_bin,fit_bin2) #compare models

confint(fit_bin) #confidence intervals

read0 = subset(read, yr==0)
read1 = subset(read, yr==1)
read1drop = read1[-c(154,175),]
```

```r
summary(read0)
summary(read1)
var(read0$count)
var(read1$count)
mean(read0$count)
mean(read1$count)
#split data and fit
fit_nb0 = glm.nb(count~attention +fluency,data=read0)
summary(fit_nb0)
plot(fit_nb0)
fit_nb1 = glm.nb(count~attention +fluency,data=read1)
summary(fit_nb1)
plot(fit_nb1)

#WORKING NUMERICAL ALGORITHM TO FIT GLM~~~~~~~~~~~~~~~~~~~~
#inverse of log link
inv.link <- function(u){
  return(exp(u))
}
#deviance for poisson
D <- function(y, mu){
  d = 2*(y*log(y/mu) - y + mu)
  return(d)
}
#derivative of log link
detadmu <- function(mu){
  return(1/mu)
}

iwls <- function(X,y,b0, tol=1e-8 ){ #iterated weighted least squares
  beta=b0 #initialise beta

  X = as.matrix(X)
  eta = X %*% beta #linear predictor
  mu = inv.link(eta)

  oldD = D(inv.link(X%*%beta),mu) #old deviance

  jj = 0 #initialise loop start
  while(jj==0){
    eta = X %*% beta #linear predictor
    mu = inv.link(eta) #inverse log link function
    z = eta + (y-mu)*detadmu(mu) #adjusted dep var
    w = mu #weights
    lmod = lm(z~ -1+X, weights=w) #regress z on x with w
    beta = as.vector(lmod$coeff) #new beta
    newD = D(inv.link(X%*%beta),mu)
    control = abs(newD-oldD)/(abs(newD)+0.1) #improvement,controls convergence crit
    if(control<1e-8) #stopping condition
      jj = 1
    oldD = newD #set new deviance to old deviance for next loop
  }
  return(beta)
}
```

```r
X = cbind(1 ,read[,2:3])
y = read[,1]
b0 = c(0.5,0,0)

iwls(X,y,b0) #returns same output as glm


#mastery~~~~~~~~~
dilu_ass <- function(p0,x,v,n,dilute=2){ #simulate dilution assay

  y= matrix(,x) #to save y variable results
  px = p0 #initialise
  for (d in seq(x)){ #for each dilution up to x
    px = px/dilute #dilute density
    pi = 1 - exp(-v*px) #calculate probability
    r= rbinom(1,n,pi) #r is number of infected, use binomial to generate
    y[d] = r/n #this ratio is our y variable
  }
  return(y)
}

estimates = matrix(,128) #preallocate space
simpler_est = matrix(,128)

v=0.1 #initialise parameters
p0=64
x=16
for (i in seq(128)){ #run simulation
  y = dilu_ass(p0,x,v,256)
  mas_data = as.data.frame(cbind(y, seq(x))) #make df with y variable and dilution x
  fit_mas = glm(V1~V2,family = 'binomial'(link=cloglog), data=mas_data ) #fit glm
  p0_hat = exp(fit_mas$coefficients[1] - log(v)) #estimate
  estimates[i] = p0_hat #save result
  #have to cut off at 8th dilution because 0s fed into log
  mas_data2 = as.data.frame(cbind(log(-log(1-y[1:8])),seq(8)))
  fit_mas2 = lm(V1~V2, data=mas_data2) #lin regression
  simpler_est[i] = exp(fit_mas2$coefficients[1] - log(v)) #estimate
}

hist(estimates, main='histogram of p0 estimate using binomial GLM')
qqnorm(estimates, main='Normal Q-Q plot of binomial GLM estimate')
mean(estimates)
median(estimates)
var(estimates)

hist(simpler_est, main='histogram of p0 estimate using linear model')
qqnorm(simpler_est,main='Normal Q-Q plot of linear model estimate')
mean(simpler_est)
median(simpler_est)
var(simpler_est)
```