GLM exercises

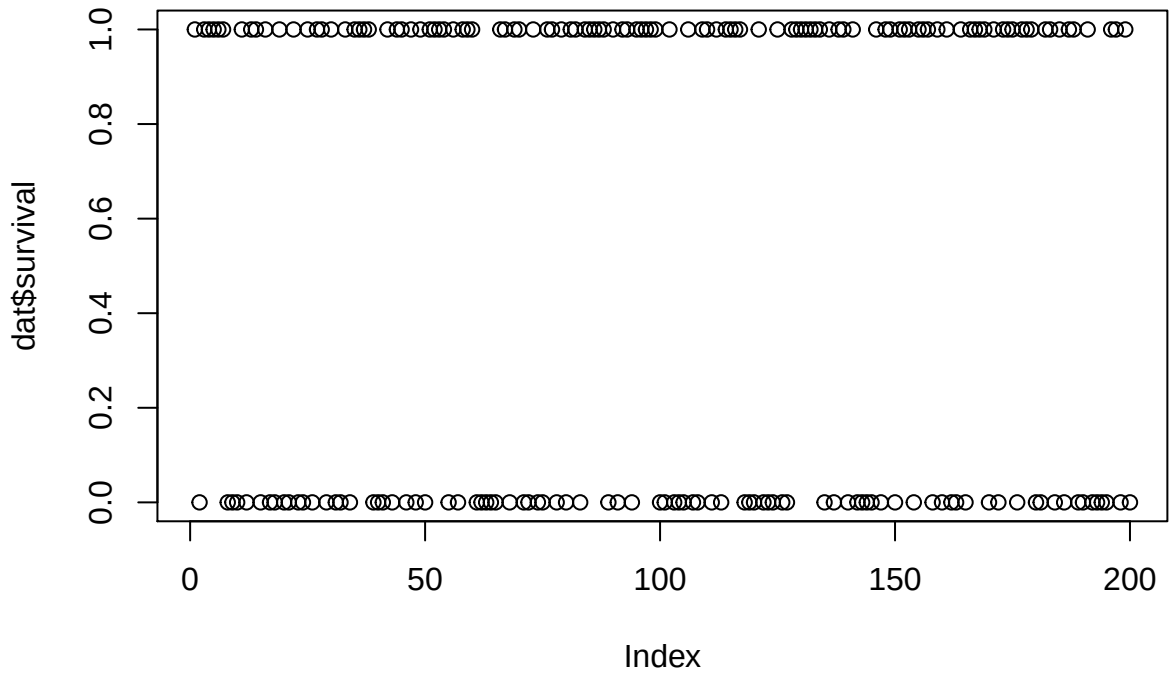
Logistic regression

- Load survivalsize.csv

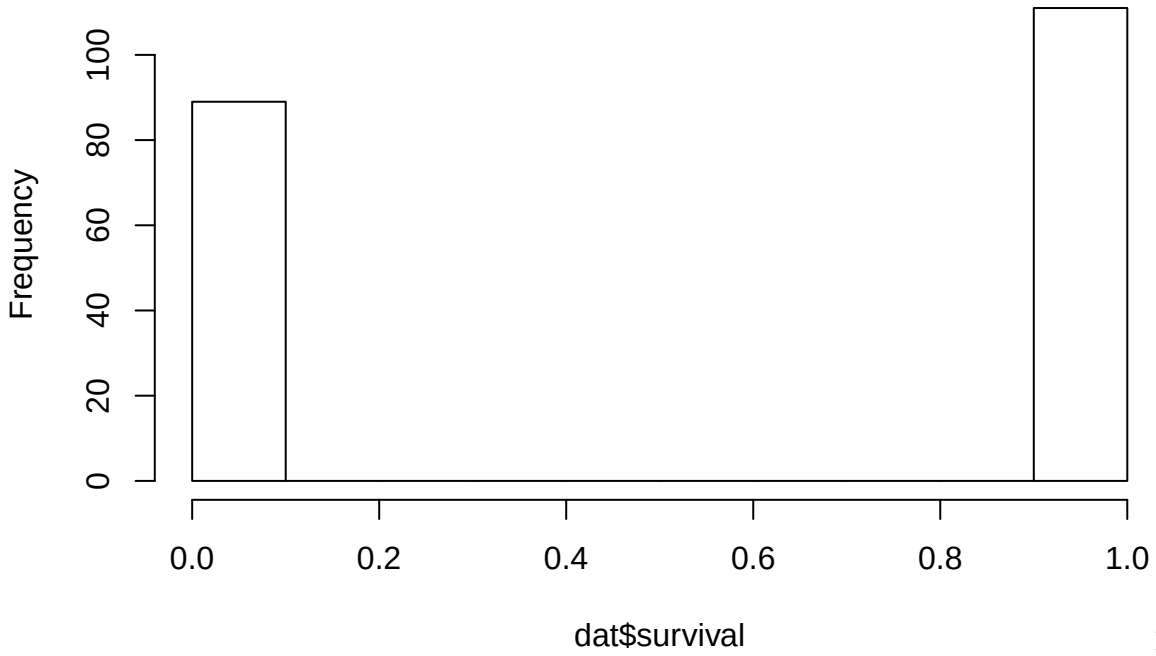
```
dat <- read.csv("survivalsize.csv")
```

- Plot survival data. What kind of distribution is it?

```
plot(dat$survival)
```




```
hist(dat$survival)
```



Bernouilli distribution (= binomial distribution of size 1).

- Fit a linear model and a logistic model with intercept only. How to interpret the estimate?

```
summary(glm(survival ~ 1, data=dat, family = "gaussian"))
```

```
##
## Call:
## glm(formula = survival ~ 1, family = "gaussian", data = dat)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -0.555  -0.555   0.445   0.445   0.445
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.55500    0.03523   15.75   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2482161)
##
##     Null deviance: 49.395  on 199  degrees of freedom
## Residual deviance: 49.395  on 199  degrees of freedom
## AIC: 291.88
##
## Number of Fisher Scoring iterations: 2
```

```
summary(glm(survival ~ 1, data=dat, family = "binomial"))
```

```
##
## Call:
```

```
## glm(formula = survival ~ 1, family = "binomial", data = dat)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.273  -1.273   1.085   1.085   1.085
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.2209     0.1423   1.552    0.121
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 274.83  on 199  degrees of freedom
## Residual deviance: 274.83  on 199  degrees of freedom
## AIC: 276.83
##
## Number of Fisher Scoring iterations: 3
```

```
  1/(1+exp(-0.2209))
```

```
## [1] 0.5550015
```

```
  mean(dat$survival)
```

```
## [1] 0.555
```

The linear model (Gaussian GLM) gives the mean survival as its intercept. The bernouilli model (Gaussian GLM) also gives the mean survival as its intercept, but on a logit scale. You have to back-transform the intercept using 1/(1+exp(- intercept)) to calculate the mean.

- Fit a linear regression and a logistic regression of survival on relative size, compare the output

```
   summary(lm1 <- glm(survival ~ 1 + relative_size, data=dat, family = "gaussian"))
```

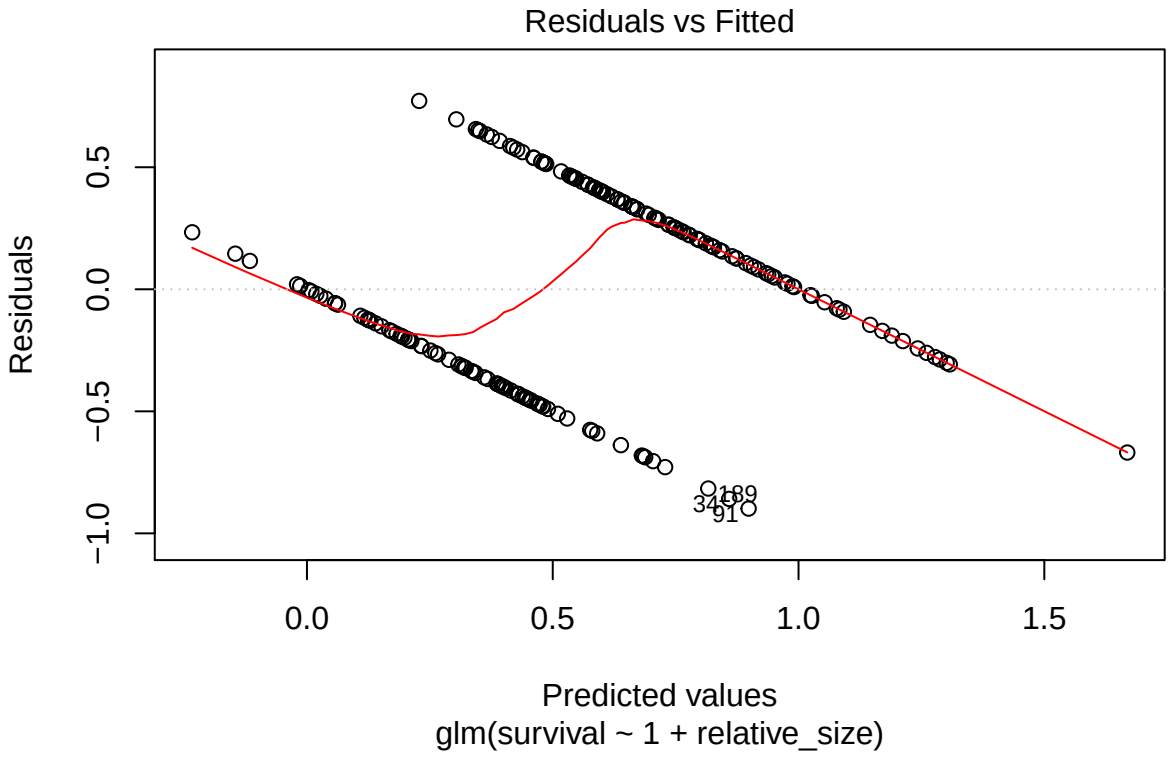
```
##
## Call:
## glm(formula = survival ~ 1 + relative_size, family = "gaussian",
##     data = dat)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.89846  -0.30970  -0.00622   0.33017   0.77174
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.55794    0.02688   20.76   <2e-16 ***
## relative_size  0.34275    0.02857   12.00   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1444421)
##
##     Null deviance: 49.395  on 199  degrees of freedom
## Residual deviance: 28.600  on 198  degrees of freedom
## AIC: 184.59
##
## Number of Fisher Scoring iterations: 2
```

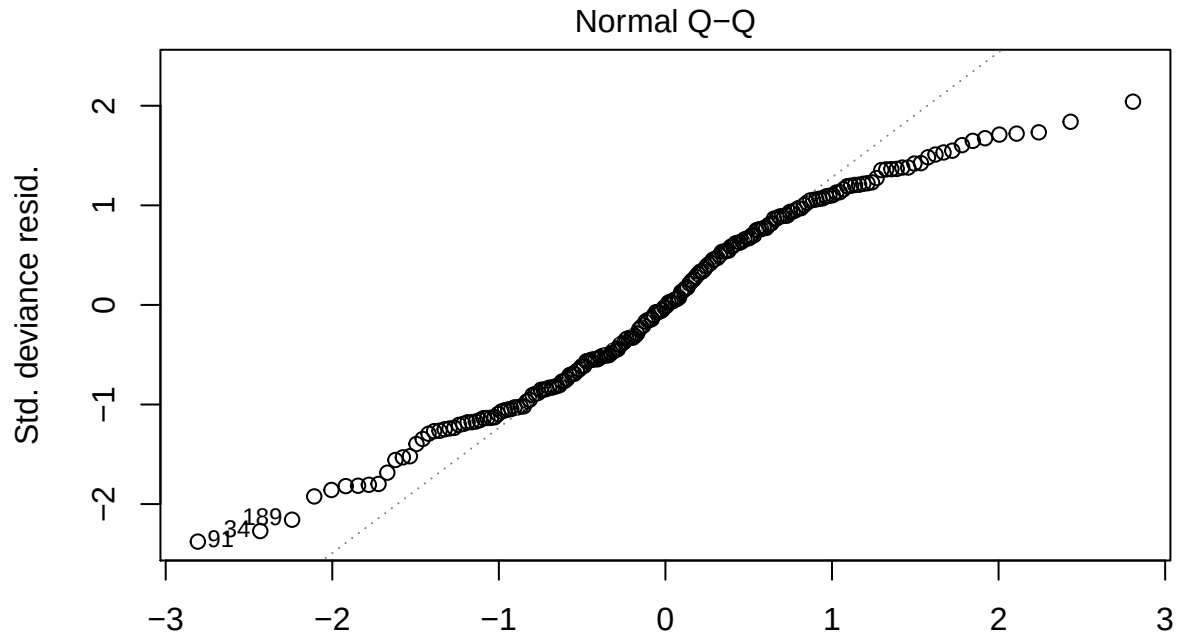
```
summary(glm1 <- glm(survival ~ 1 + relative_size, data=dat, family = "binomial"))
```

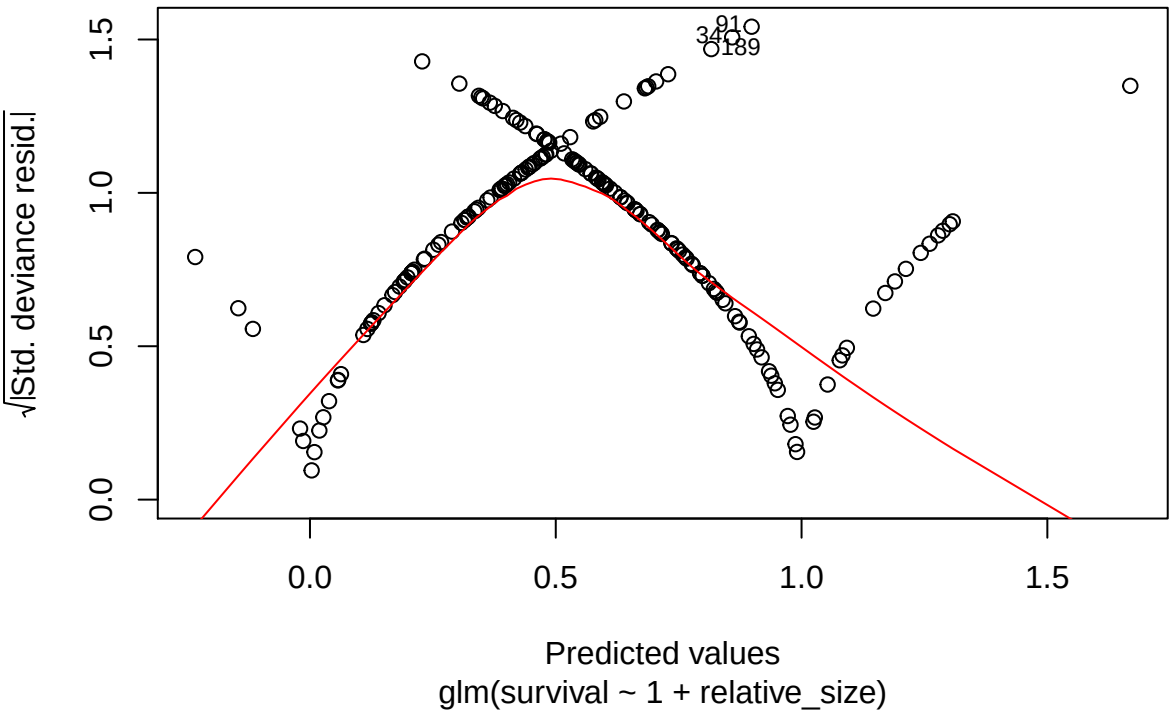
```
##
## Call:
## glm(formula = survival ~ 1 + relative_size, family = "binomial",
##     data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6020  -0.6057   0.1078   0.6412   2.1218
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.5610     0.2092   2.682  0.00731 **
## relative_size   2.8078     0.4015   6.993  2.7e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 274.83  on 199  degrees of freedom
## Residual deviance: 159.02  on 198  degrees of freedom
## AIC: 163.02
##
## Number of Fisher Scoring iterations: 6
```

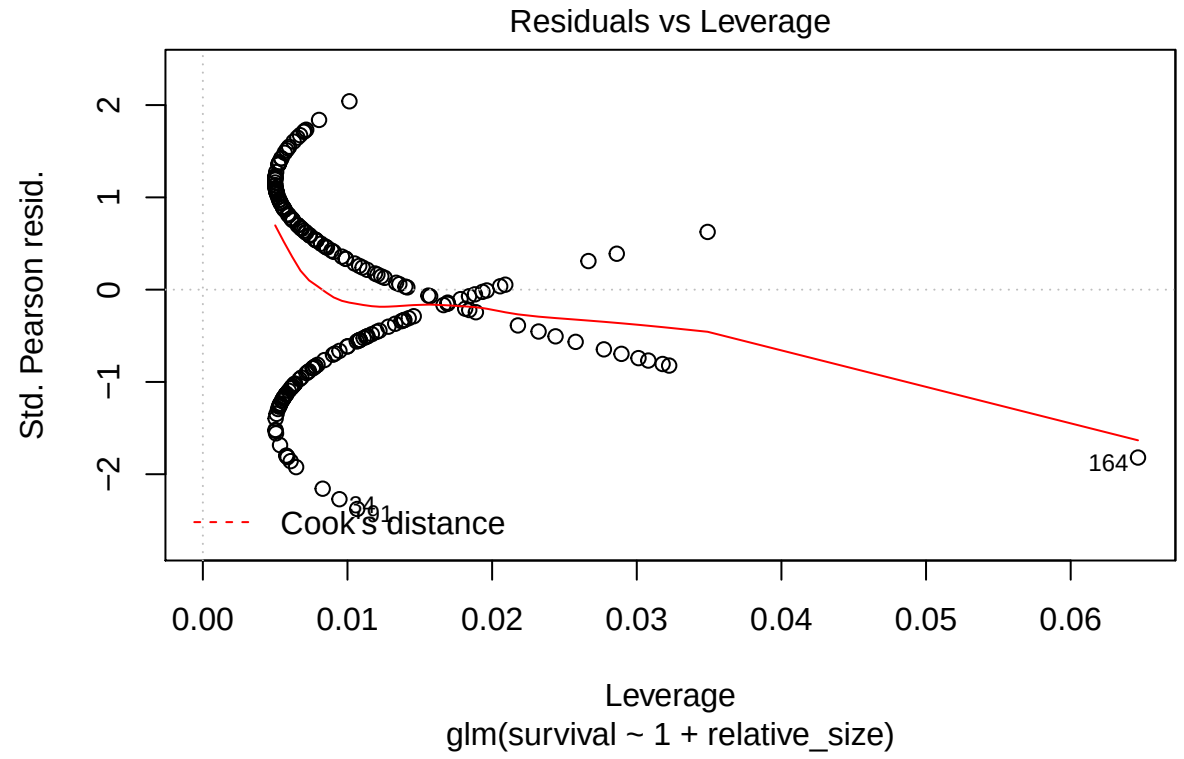
- Check the diagnostic plots for both models. Should you be worried?

```
plot(lm1)
```

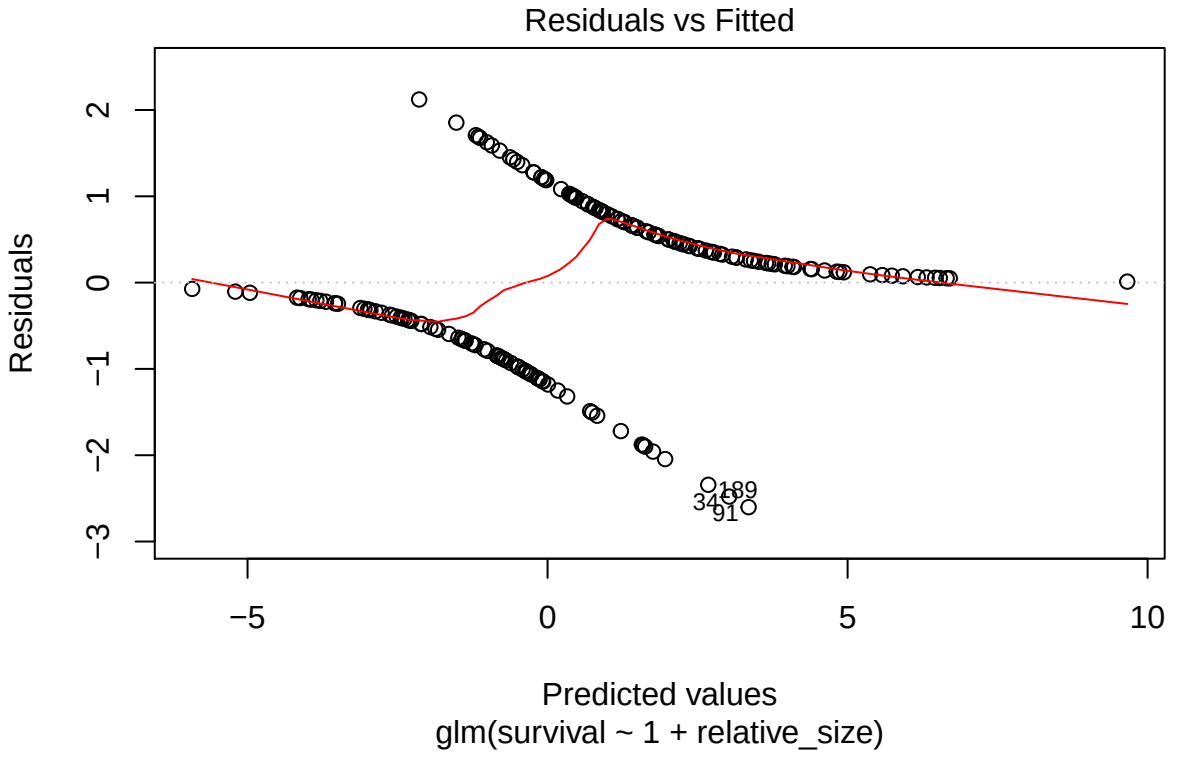


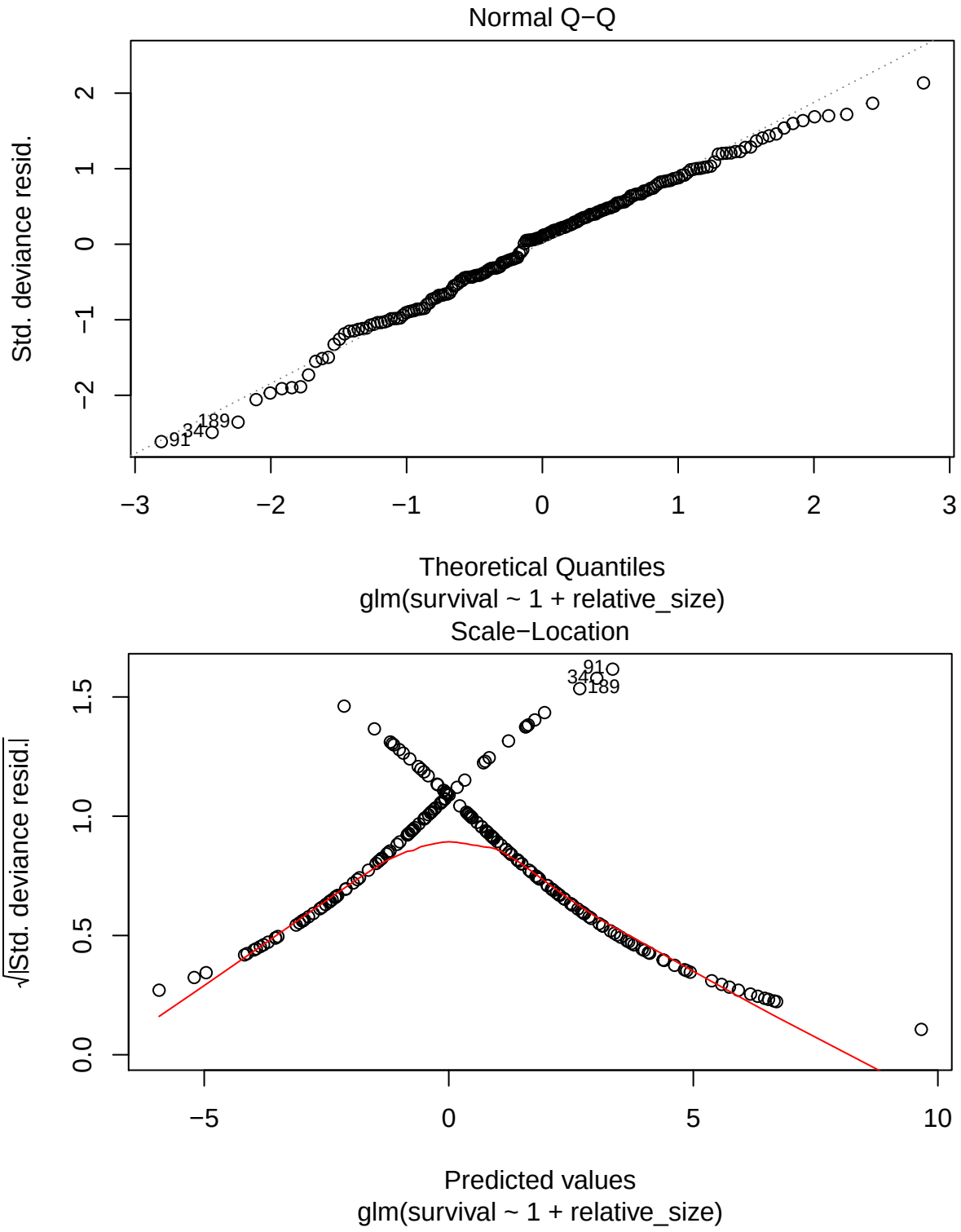

Normal Q−Q
Std. deviance resid.
Theoretical Quantiles
glm(survival ~ 1 + relative_size)
91
34
189

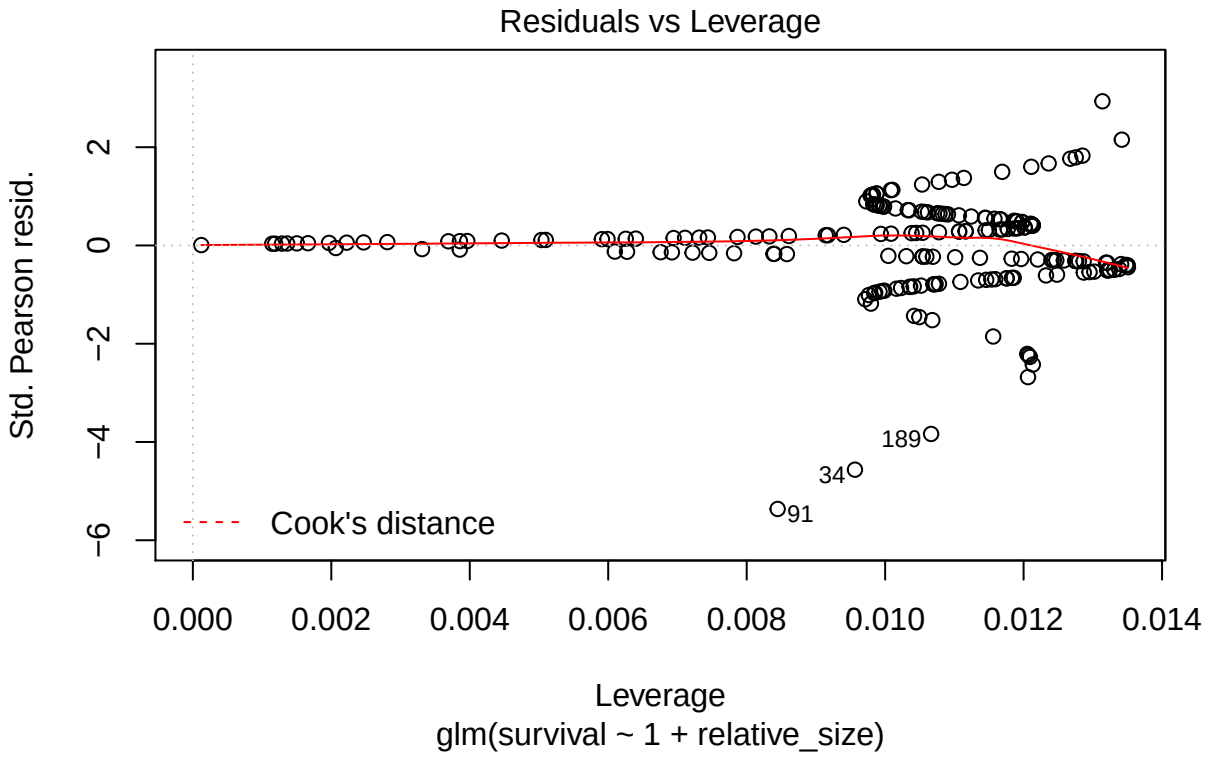

Scale−Location
√|Std. deviance resid.|
Predicted values
glm(survival ~ 1 + relative_size)
91
34
189


Residuals vs Leverage
Std. Pearson resid.
2
1
0
−1
−2
0.00
0.01
0.02
0.03
0.04
0.05
0.06
164
34
91
Cook's distance
Leverage
glm(survival ~ 1 + relative_size)

```
plot(glm1)
```


Residuals vs Fitted
Residuals
2
1
0
−1
−2
−3
−5
0
5
10
189
34
91
Predicted values
glm(survival ~ 1 + relative_size)

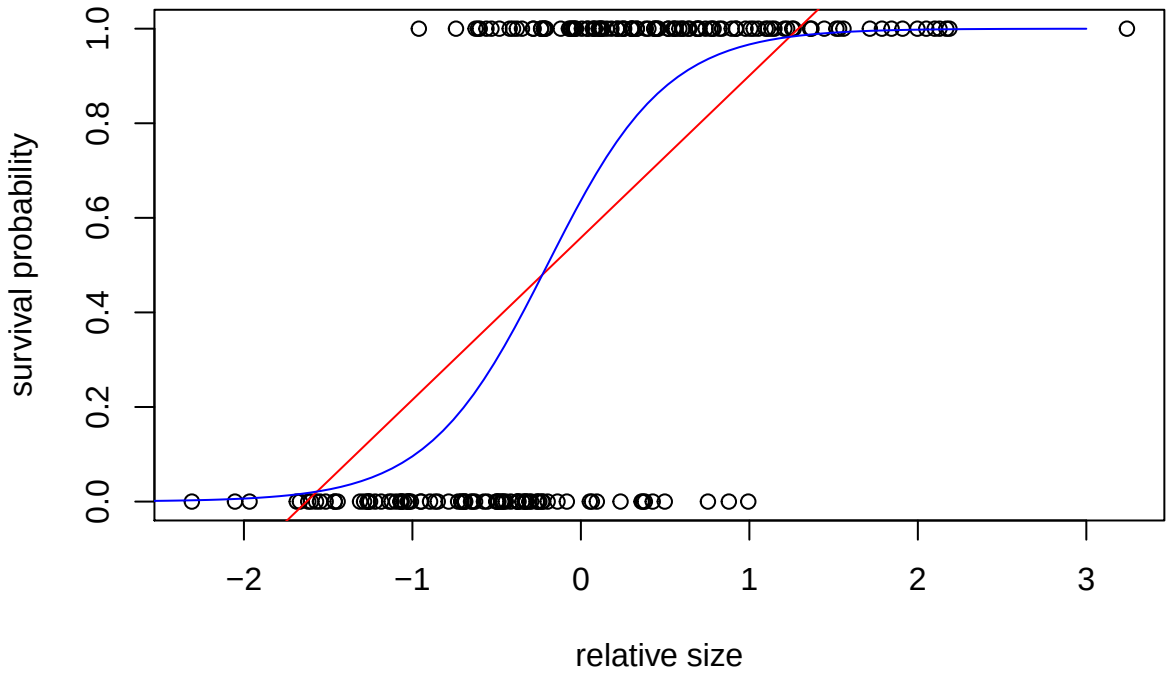

Normal Q−Q
Std. deviance resid.
Theoretical Quantiles
glm(survival ~ 1 + relative_size)
Scale−Location
√|Std. deviance resid.|
Predicted values
glm(survival ~ 1 + relative_size)



The diagnostic for the lm shows violations of the assumpations. The glm diagnostic looks similar, but you should not be worried, because the glm doesn't make the same assumptions. To be specific, the glm does not have defined residuals on the scale of the linear predictor (where distribution assumptions are relevant), but only expected values. The diagnostic are based on the data scale residuals, but it is not the scale on which the glm is fitted. These plots are not useful to check the fit of a glm. Instead, you can check a model works by simulating data from the model estimates and comparing the distribution to that of the initial data (we do not cover that method today).

- Extract and visualize a model prediction from both models (use the function predict, and/or do it by hand to practice link-function back-transformation)

```
plot(dat$relative_size, dat$survival, ylab="survival probability",
     xlab="relative size")
ndat <- data.frame(relative_size = seq(-3,3, length.out = 100))
ndat <- cbind(ndat, predict(lm1, newdata = ndat),
              predict(glm1, newdata = ndat, type = "response"))
lines(ndat[,1], ndat[,2], col="red")
lines(ndat[,1], ndat[,3], col="blue")
```



The linear model makes unreasonable predictions, falling out of the range of possible values. The GLM fits much better.

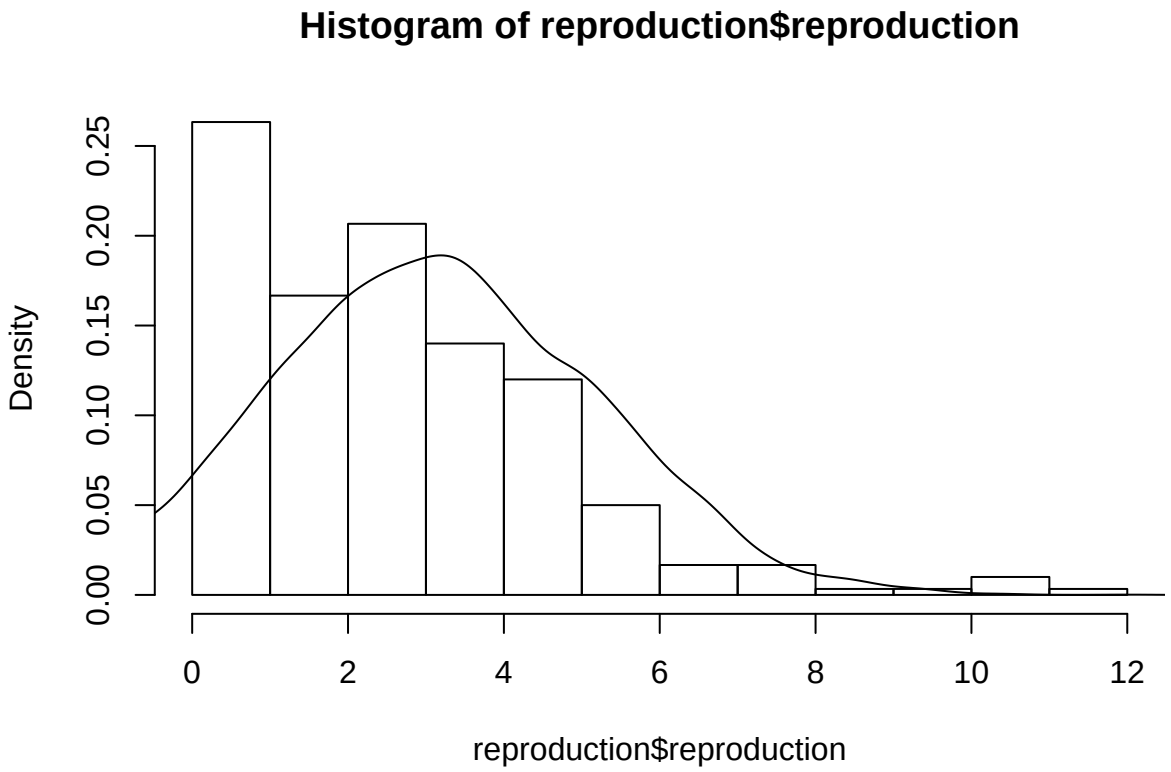
Poisson Regression

- Load the data reproduction.csv

```
reproduction <- read.csv("reproduction.csv")
```

- Plot reproduction data, calculate the mean and variance.
- Overlay a Gaussian distribution of same mean and variance, does it fit?

```
hist(reproduction$reproduction, freq = FALSE)
normsamp <- rnorm(10000, mean(reproduction$reproduction),
  sqrt(var(reproduction$reproduction)))
lines(density(normsamp))
```



- Fit an compare a lm and a Poisson glm of reproduction on size

```
summary(glm3 <- glm(reproduction ~ size , family=poisson, data=reproduction))
```

```
##
## Call:
## glm(formula = reproduction ~ size, family = poisson, data = reproduction)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.03254  -0.81019  -0.05926   0.57879   2.60137
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.62044    0.05737   10.81   <2e-16 ***
## size         0.20350    0.01653   12.31   <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: 467.68  on 299  degrees of freedom
## Residual deviance: 321.02  on 298  degrees of freedom
## AIC: 1132.5
##
## Number of Fisher Scoring iterations: 5
```

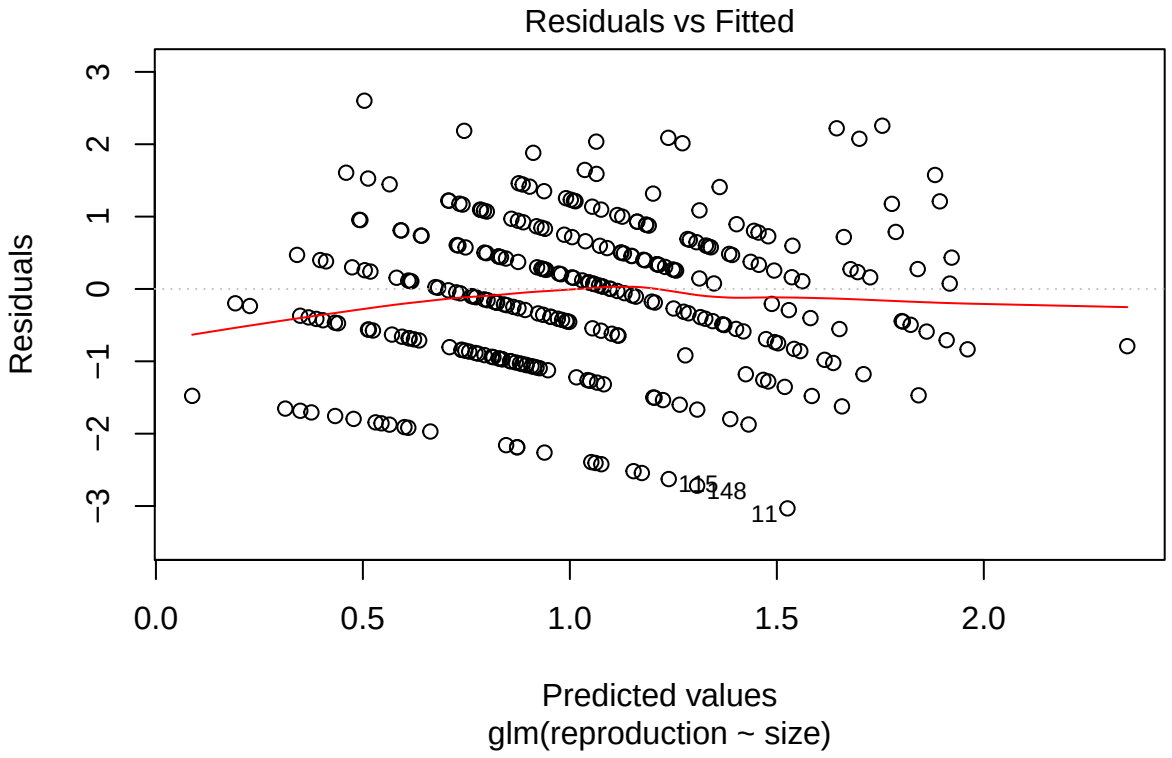
```
summary(lm3 <- lm(reproduction ~ size,data=reproduction))
```

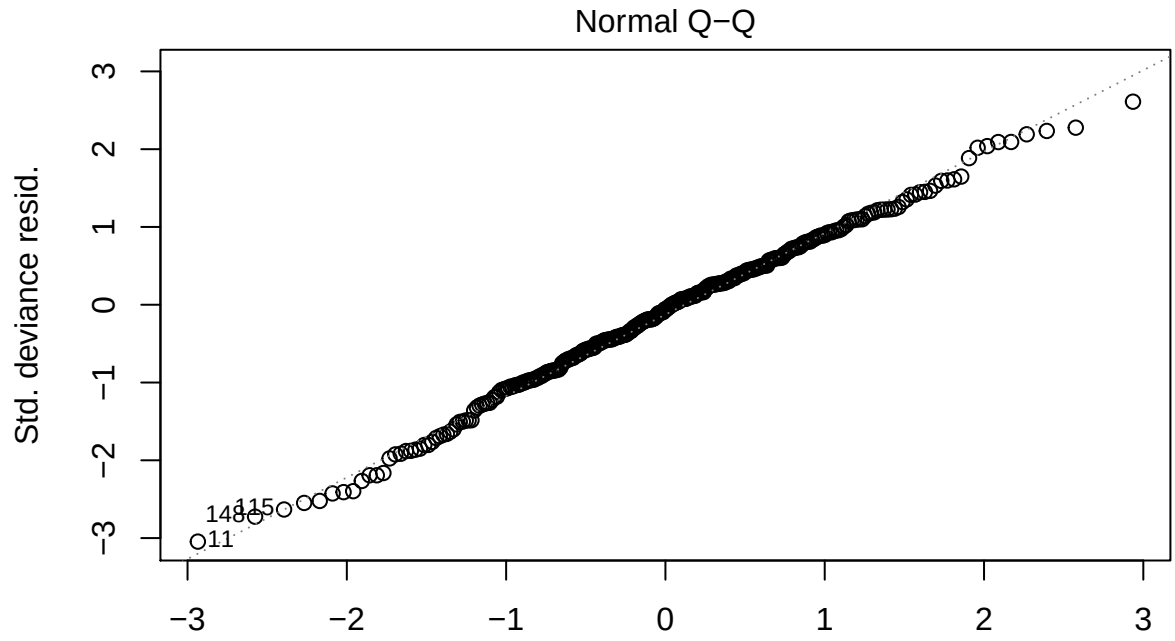
```
##
## Call:
```

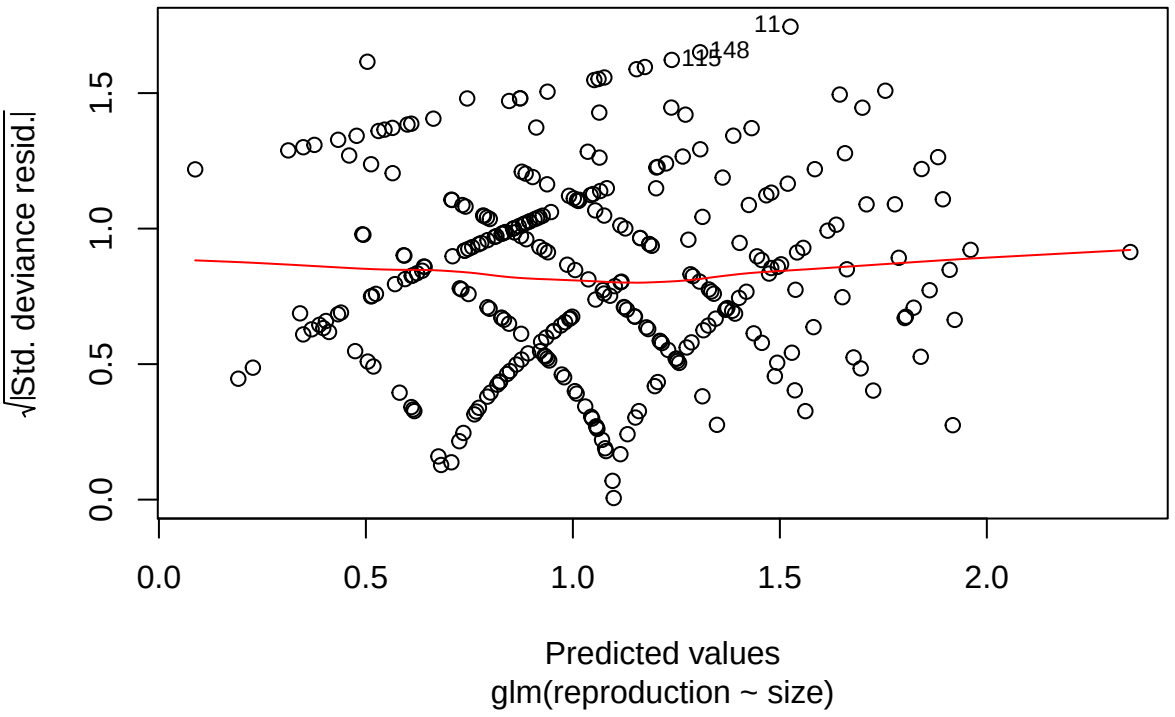
```
## lm(formula = reproduction ~ size, data = reproduction)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6334 -1.1360 -0.0929  0.9934  6.6222
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.69184    0.14892   11.36   <2e-16 ***
## size         0.66130    0.05316   12.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.739 on 298 degrees of freedom
## Multiple R-squared:  0.3418, Adjusted R-squared:  0.3396
## F-statistic: 154.7 on 1 and 298 DF,  p-value: < 2.2e-16
```

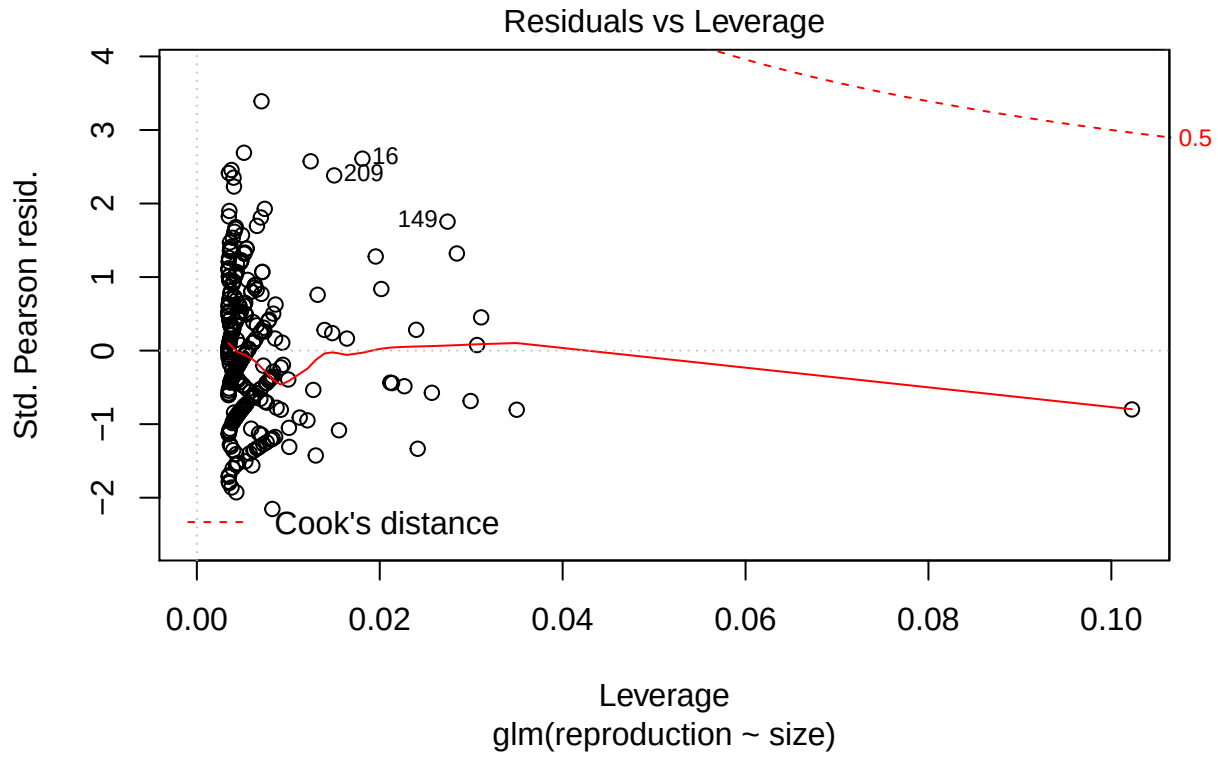
- Check the diagnostic plots for both models. Should you be worried?

```
plot(glm3)
```

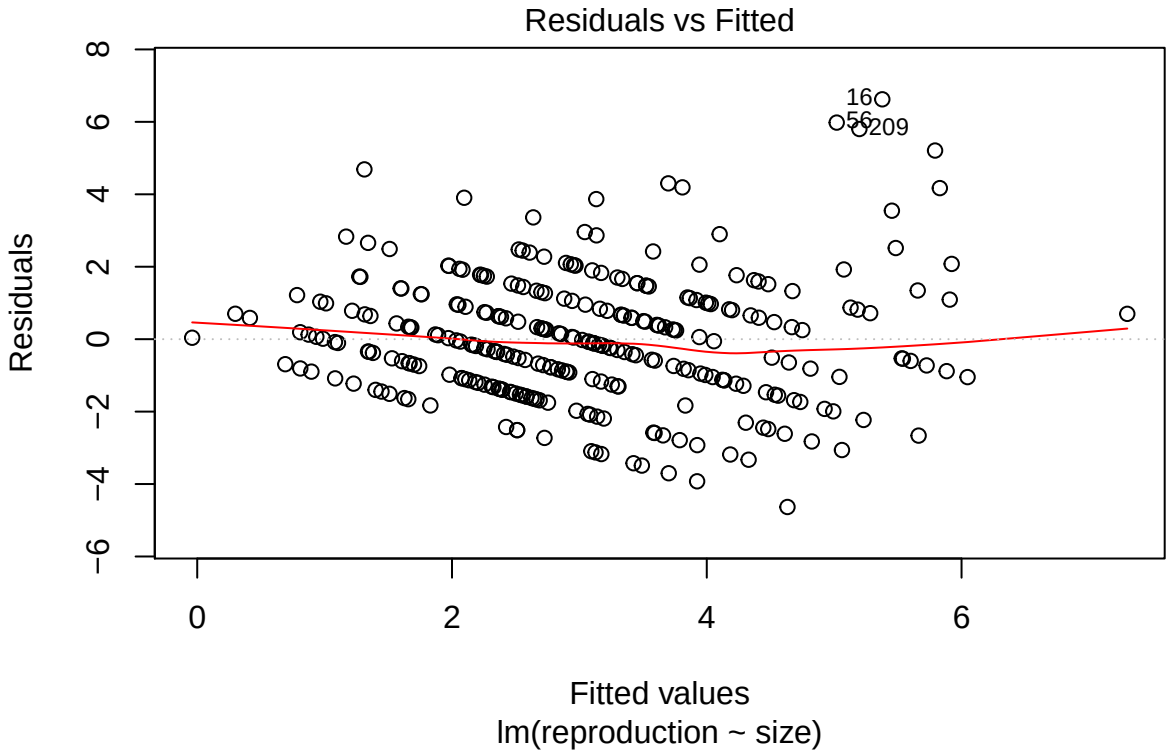


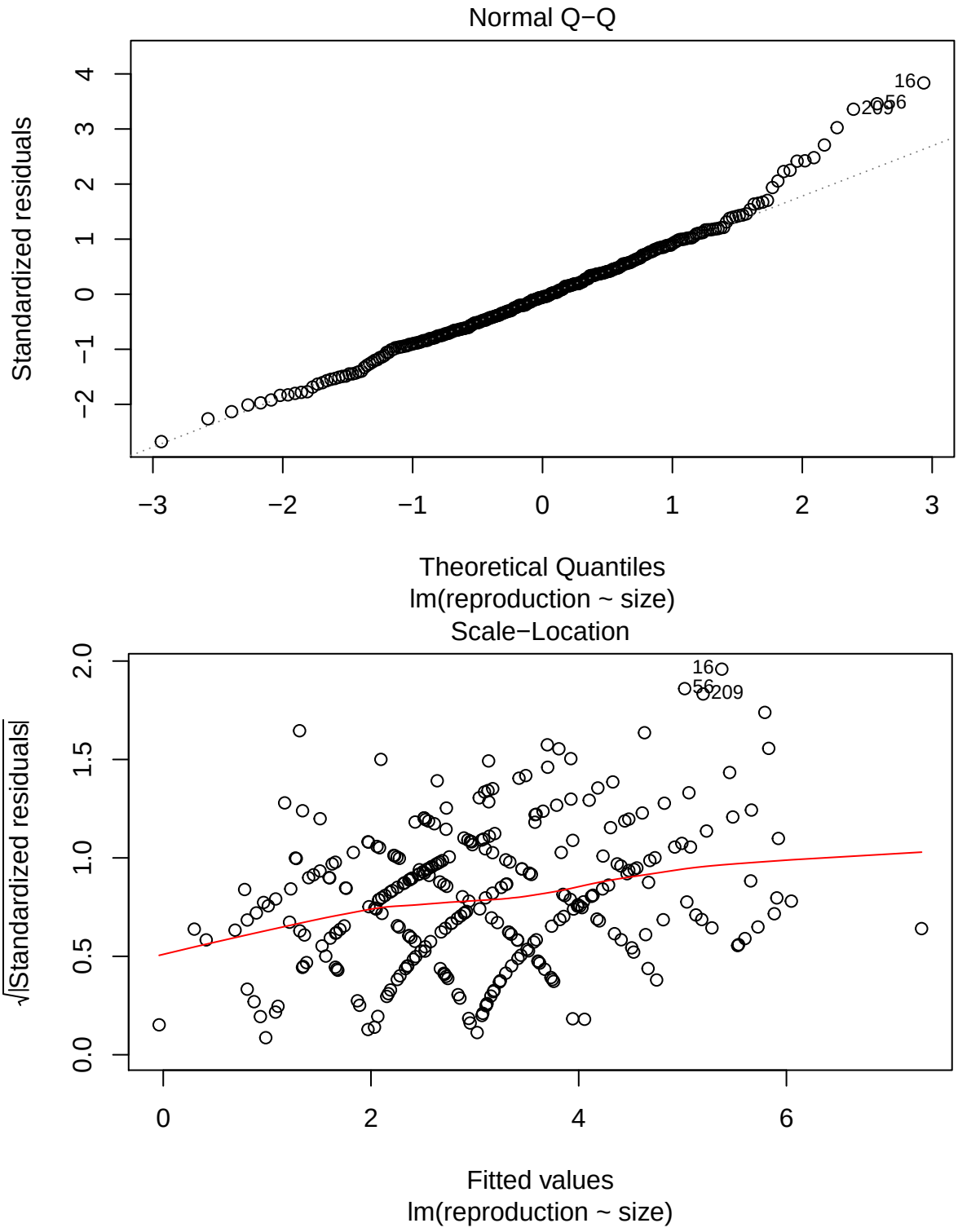

Normal Q−Q
Std. deviance resid.
−3
−2
−1
0
1
2
3
148
115
11
Theoretical Quantiles
glm(reproduction ~ size)

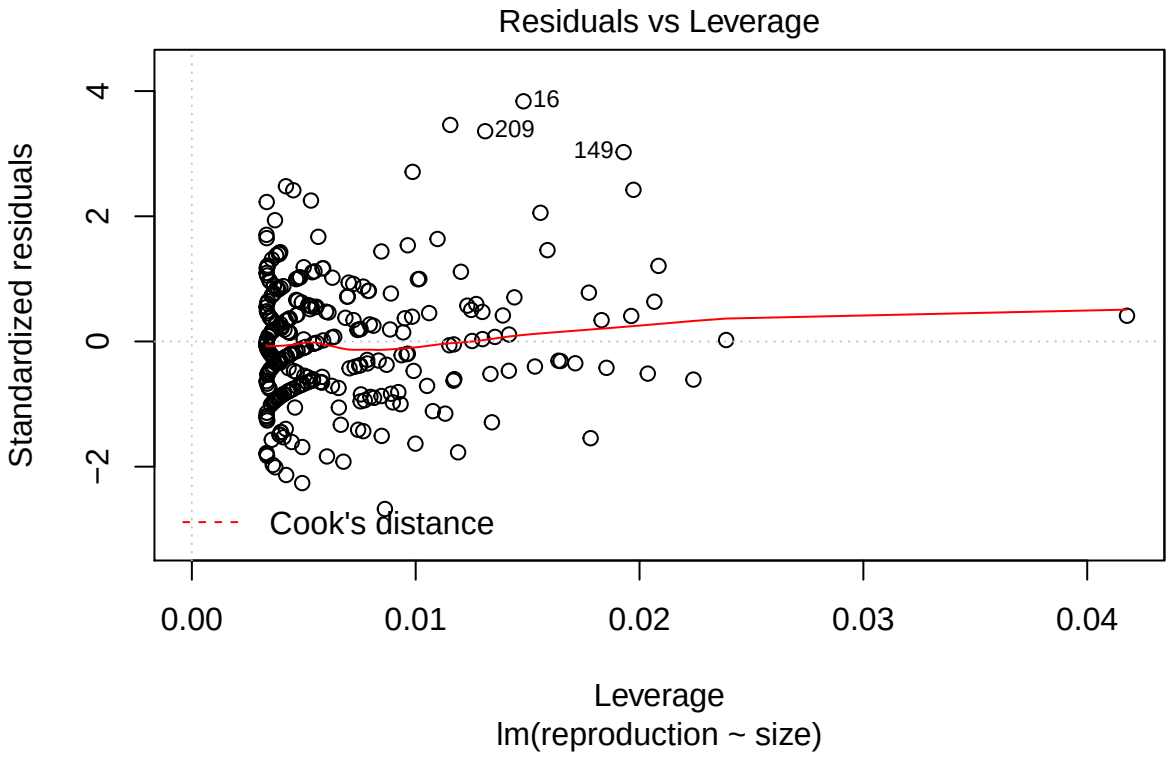

Scale−Location
√|Std. deviance resid.|
0.0
0.5
1.0
1.5
2.0
11
148
115
Predicted values
glm(reproduction ~ size)


Residuals vs Leverage
Std. Pearson resid.
4
3
2
1
0
−1
−2
16
209
149
0.5
Cook's distance
0.00
0.02
0.04
0.06
0.08
0.10
Leverage
glm(reproduction ~ size)

```
plot(lm3)
```


Residuals vs Fitted
Residuals
8
6
4
2
0
−2
−4
−6
16
56
209
0
2
4
6
Fitted values
lm(reproduction ~ size)

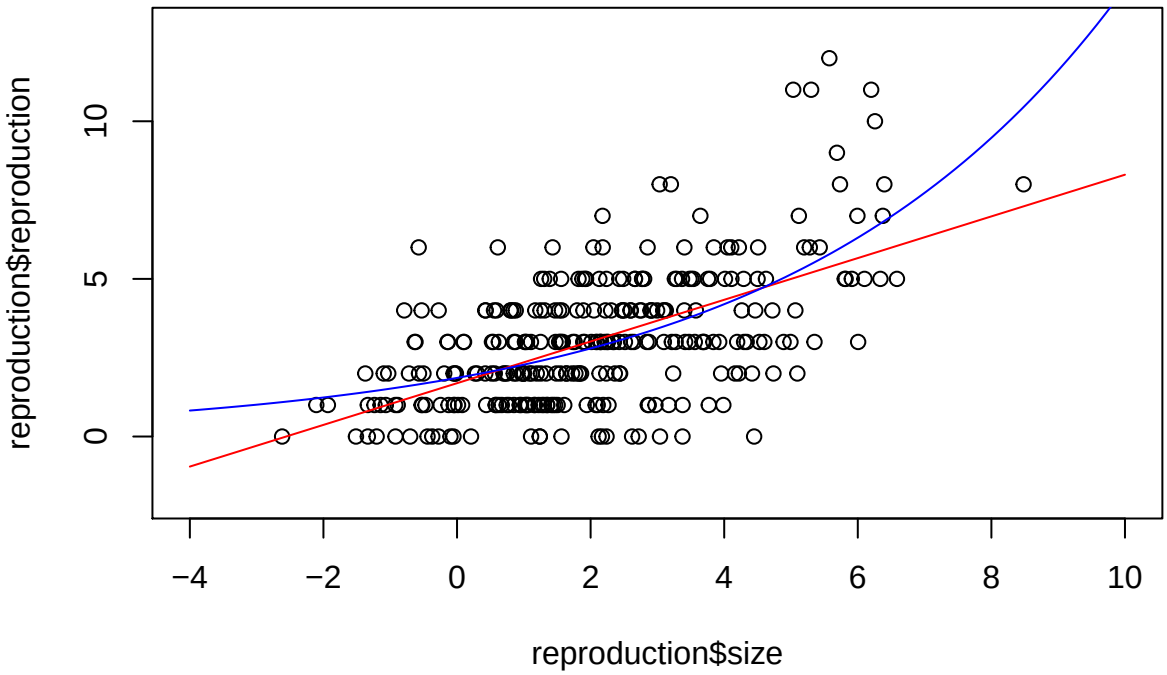

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(reproduction ~ size)
Scale−Location
√|Standardized residuals|
Fitted values
lm(reproduction ~ size)
16
56
209



The lm is wrong. Diagnostic looks similar for the glm, but assumptions are not the same, so it is okay.

- Extract and visualize a model prediction from both models (use the function predict, and/or do it by hand to practice link-function back-transformation)

```
plot(reproduction$reproduction, x=reproduction$size, xlim=c(-4,10),ylim=c(-2,13))
ndat <- data.frame(size=seq(-4,10,length.out = 100))
ndat <- cbind(ndat, predict(lm3, newdata = ndat),
              predict(glm3, newdata = ndat, type = "response"))
lines(ndat[,1], ndat[,2], col="red")
lines(ndat[,1], ndat[,3], col="blue")
```

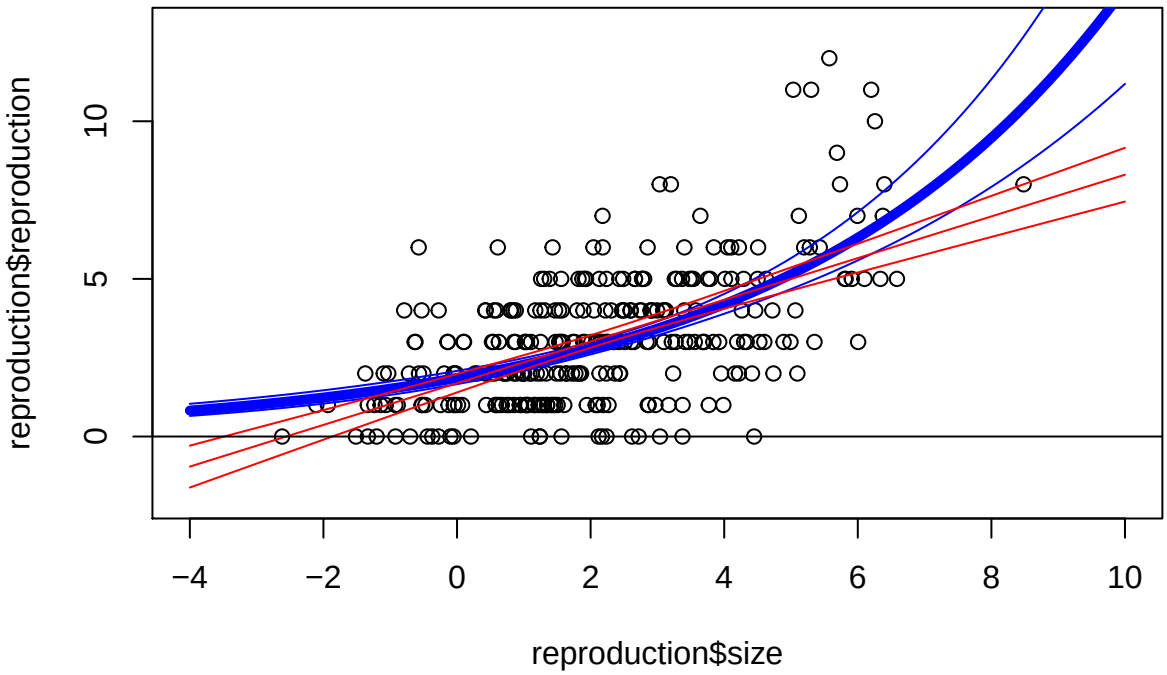


The GLM fits better, predicts correctly that negative values are impossible, and that the values tend to increase exponentially.

Adding the confidence interval is a bit annoying, but can be done:

```
plot(reproduction$reproduction, x=reproduction$size, xlim=c(-4,10),ylim=c(-2,13))
ndat <- data.frame(size=seq(-4,10,length.out = 100))
ndatp <- cbind(ndat,predict(glm3, newdata = ndat, se.fit = TRUE))
ndatp$plci <- ndatp$fit -1.96*ndatp$se.fit
ndatp$phci <- ndatp$fit +1.96*ndatp$se.fit
lines(ndatp$size, exp(ndatp$fit), col="blue", lwd=5)
lines(ndatp$size, exp(ndatp$plci), col="blue")
lines(ndatp$size, exp(ndatp$phci), col="blue")

lm3 <- lm(reproduction ~ size,data=reproduction)
ndatg <- cbind(ndat,predict(lm3, newdata = ndat, interval = "confidence"))
lines(ndatg$size, ndatg[,2], col="red")
lines(ndatg$size, ndatg[,3], col="red")
  lines(ndatg$size, ndatg[,4], col="red")
  abline(h=0)
```



- Before GLMs, researchers used to log-transform the data and fit linear models. What are the problems with this approach?

```
lm(log(reproduction) ~size, data=reproduction)
lm(log(reproduction + 0.01) ~size, data=reproduction)
lm(log(reproduction + 0.0001) ~size, data=reproduction)
lm(log(reproduction + 0.1) ~size, data=reproduction)
```

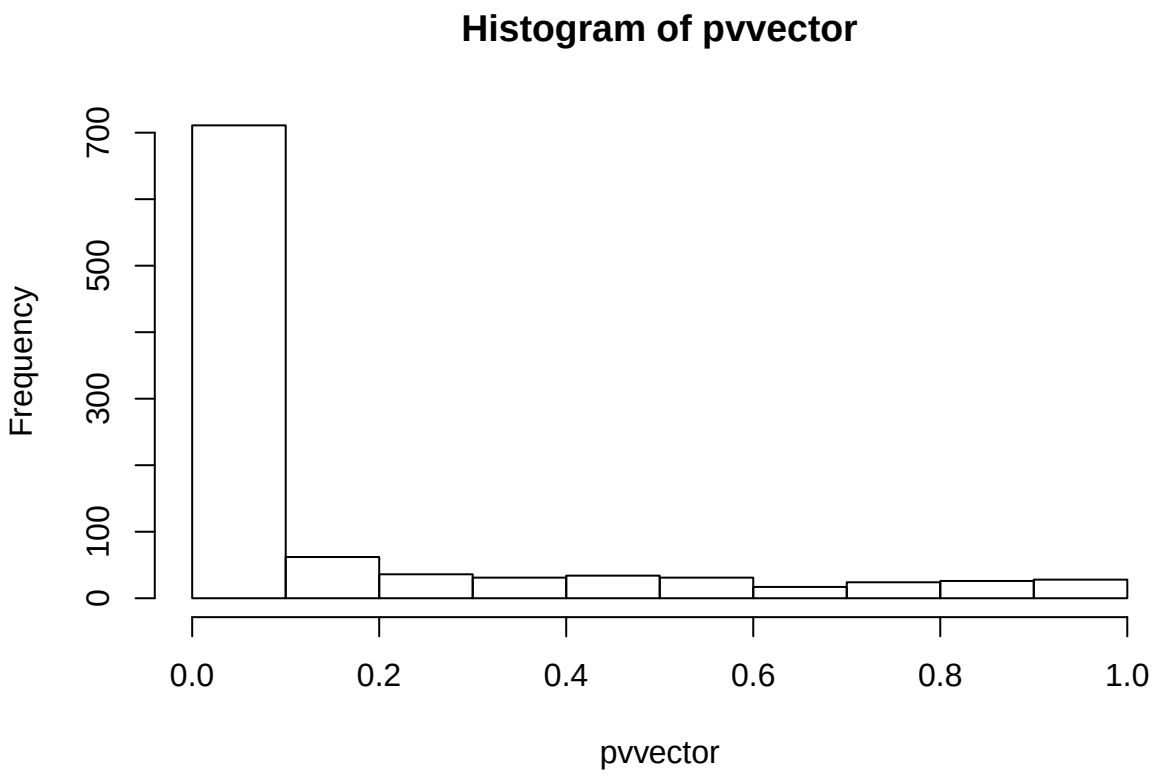
Either you cannot fit the model because log(0) is not defined, or you have to add an arbitrary quantity to the zeros. The choice of the arbitrary quantity changes model estimates, so it is difficult to interpret them.

Over-dispersion

Write a for loop to look at the distribution of p-values for a Poisson GLM and a quasi-Poisson GLM.

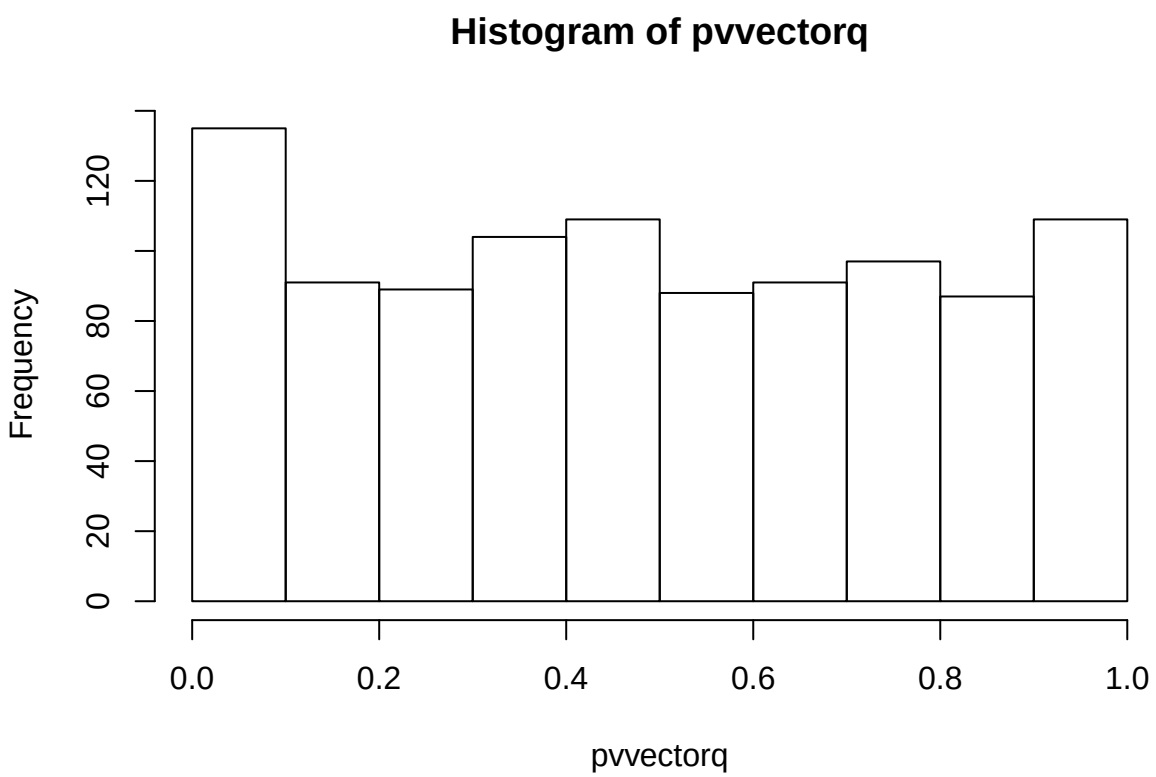
```
   set.seed(123)
   pvvector <- vector(length = 1000)
pvvectorq <- vector(length = 1000)
for (i in 1:1000)
    {
      x <- rnorm(100)
       y <- exp(-1 + rnorm(100, 0, 2))
       obs <- sapply(y, FUN = function(x){rpois(n = 1, lambda = x)})
       glm2 <- glm(obs ~ x, family = "poisson")
       sglm2 <- summary(glm2)
       pvvector[i] <- sglm2$coefficients[2,4]

       glm2q <- glm(obs ~ x, family = "quasipoisson")
       sglm2q <- summary(glm2q)
      pvvectorq[i] <- sglm2q$coefficients[2,4]
      }
   hist(pvvector); mean(pvvector<0.05)
```




```
## [1] 0.663
```

```
    hist(pvvectorq) ; mean(pvvectorq<0.05)
```




```
## [1] 0.086
```

The Poisson GLM finds significant effect 66.3% of the time, while we simulated no effect. The quasi-Poisson GLM finds significant effects only 8.6% of the time (which is a bit more than the 5% we should get, but not by much). The quasi-Poisson is much more reliable than the Poisson.

Never use a simple Poisson GLM, it makes unreasonable and unecessary assumptions.