Lesson 10: Poisson regression

Lesson 10.2

Data

For an example of Poisson regression, we'll use the badhealth data set from the `COUNT` package in `R`.

```
library("COUNT")
```

```
## Loading required package: msme
```

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

```
## Loading required package: lattice
```

```
## Loading required package: sandwich
```

```
data("badhealth")
?badhealth
head(badhealth)
```

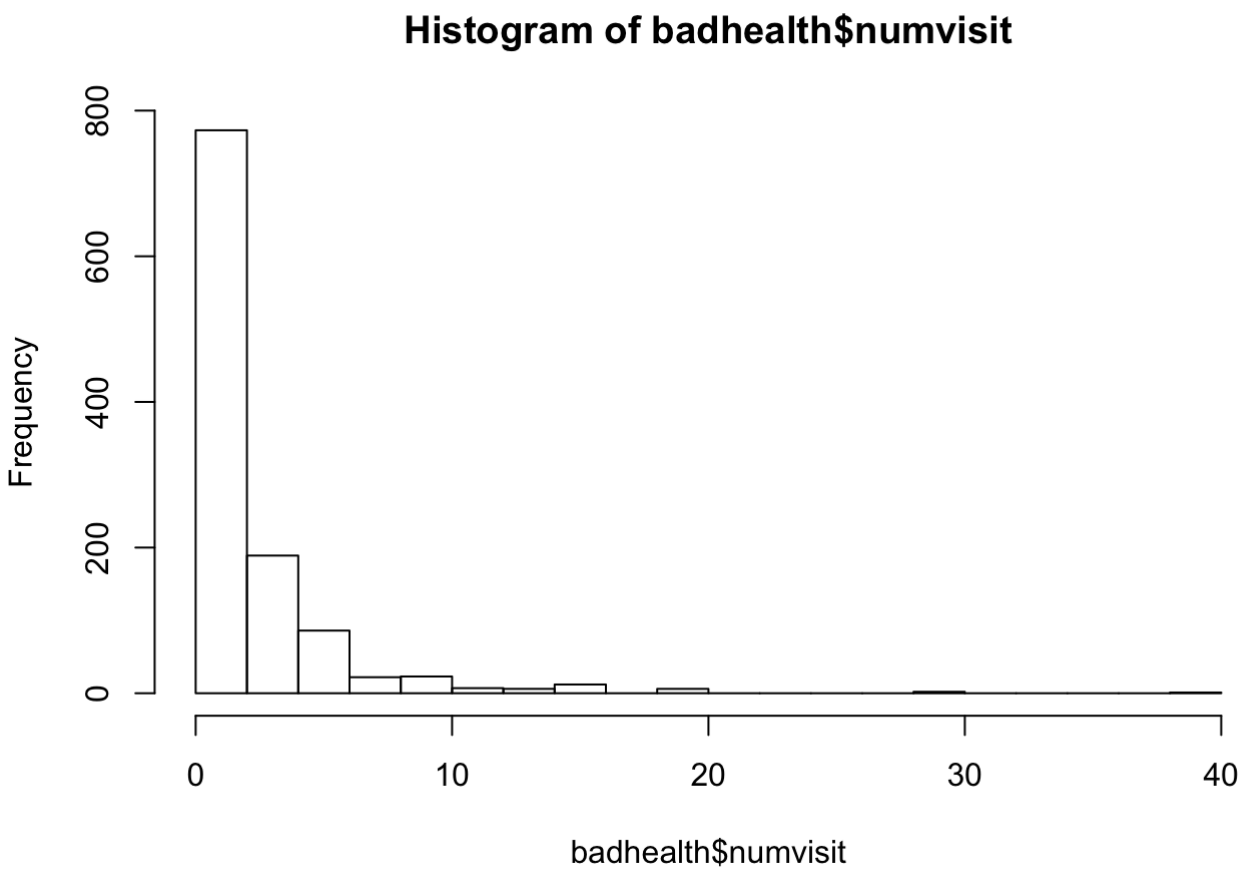
```
##   numvisit badh age
## 1       30    0  58
## 2       20    0  54
## 3       16    0  44
## 4       20    0  57
## 5       15    0  33
## 6       15    0  28
```

```
any(is.na(badhealth))
```

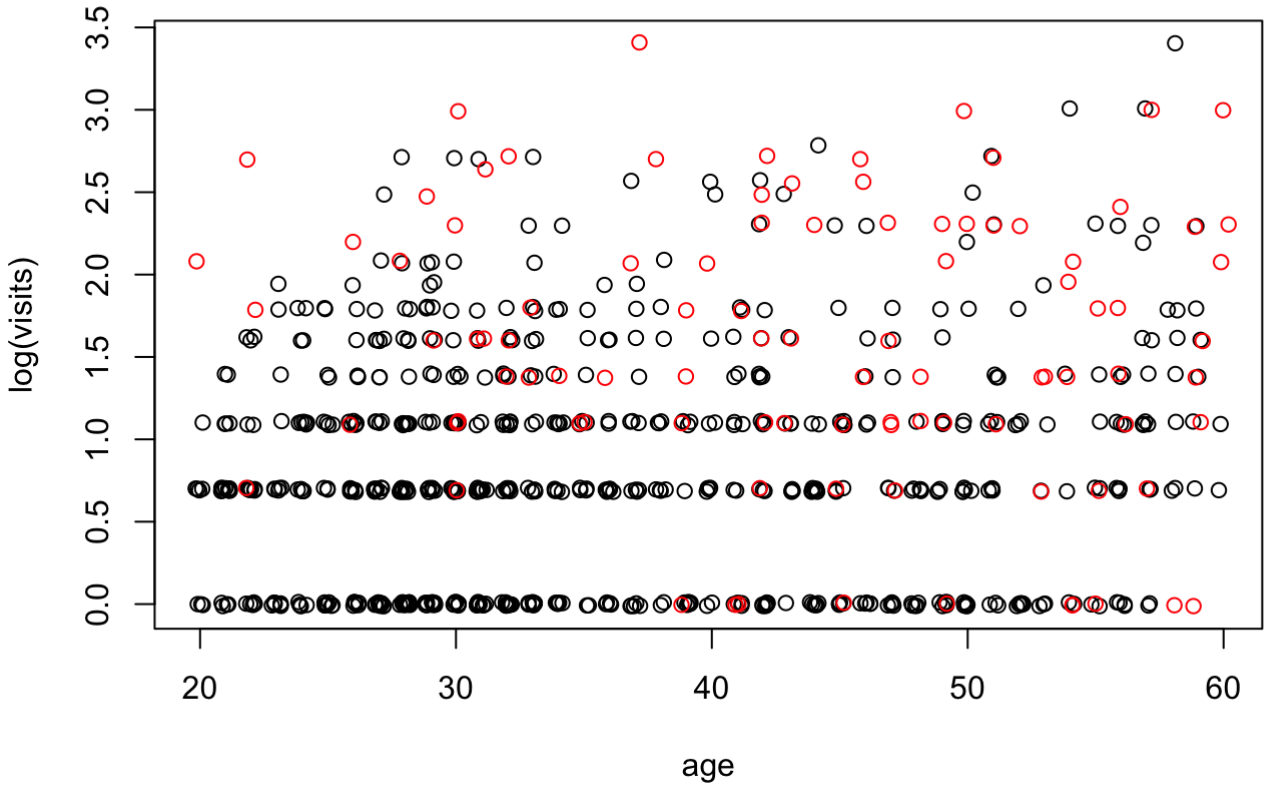
```
## [1] FALSE
```

As usual, let's visualize these data.

```
hist(badhealth$numvisit, breaks=20)
```




```
plot(jitter(log(numvisit)) ~ jitter(age), data=badhealth, subset=badh==0, xlab="age", yl
ab="log(visits)")
points(jitter(log(numvisit)) ~ jitter(age), data=badhealth, subset=badh==1, col="red")
```



Model

It appears that both age and bad health are related to the number of doctor visits. We should include model terms for both variables. If we believe the age/visits relationship is different between healthy and non-healthy populations, we should also include an interaction term. We will fit the full model here and leave it to you to compare it with the simpler additive model.

```
library("rjags")
```

```
## Loading required package: coda
```

```
## Linked to JAGS 4.2.0
```

```
## Loaded modules: basemod,bugs
```

```
mod_string = " model {
    for (i in 1:length(numvisit)) {
        numvisit[i] ~ dpois(lam[i])
        log(lam[i]) = int + b_badh*badh[i] + b_age*age[i] + b_intx*age[i]*badh[i]
    }

    int ~ dnorm(0.0, 1.0/1e6)
    b_badh ~ dnorm(0.0, 1.0/1e4)
    b_age ~ dnorm(0.0, 1.0/1e4)
    b_intx ~ dnorm(0.0, 1.0/1e4)
} "

set.seed(102)

data_jags = as.list(badhealth)

params = c("int", "b_badh", "b_age", "b_intx")

mod = jags.model(textConnection(mod_string), data=data_jags, n.chains=3)
update(mod, 1e3)

mod_sim = coda.samples(model=mod,
                       variable.names=params,
                       n.iter=5e3)
mod_csim = as.mcmc(do.call(rbind, mod_sim))

## convergence diagnostics
plot(mod_sim)

gelman.diag(mod_sim)
autocorr.diag(mod_sim)
autocorr.plot(mod_sim)
effectiveSize(mod_sim)

## compute DIC
dic = dic.samples(mod, n.iter=1e3)
```

Model checking

To get a general idea of the model's performance, we can look at predicted values and residuals as usual. Don't forget that we must apply the inverse of the link function to get predictions for λ.

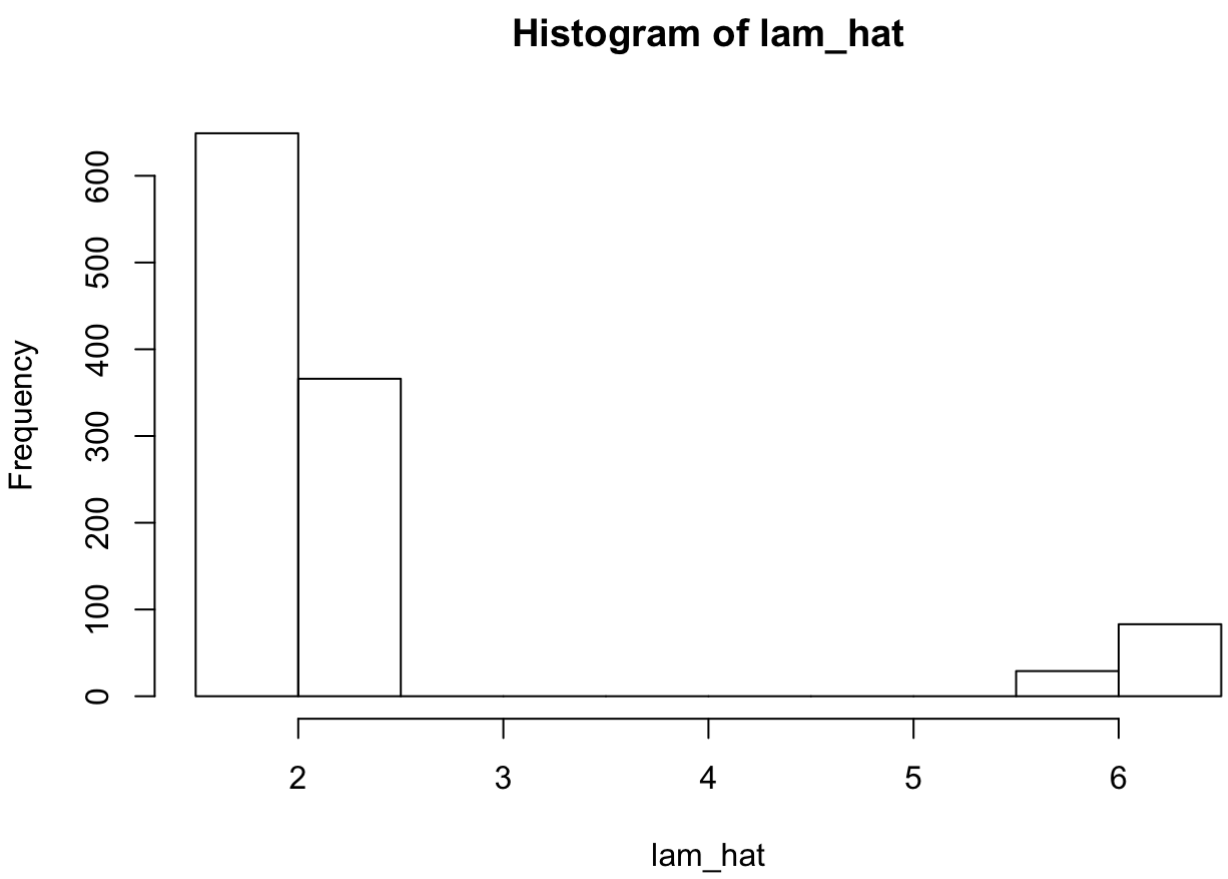
```
X = as.matrix(badhealth[,-1])
X = cbind(X, with(badhealth, badh*age))
head(X)
```

```
##      badh age
## [1,]    0  58 0
## [2,]    0  54 0
## [3,]    0  44 0
## [4,]    0  57 0
## [5,]    0  33 0
## [6,]    0  28 0
```

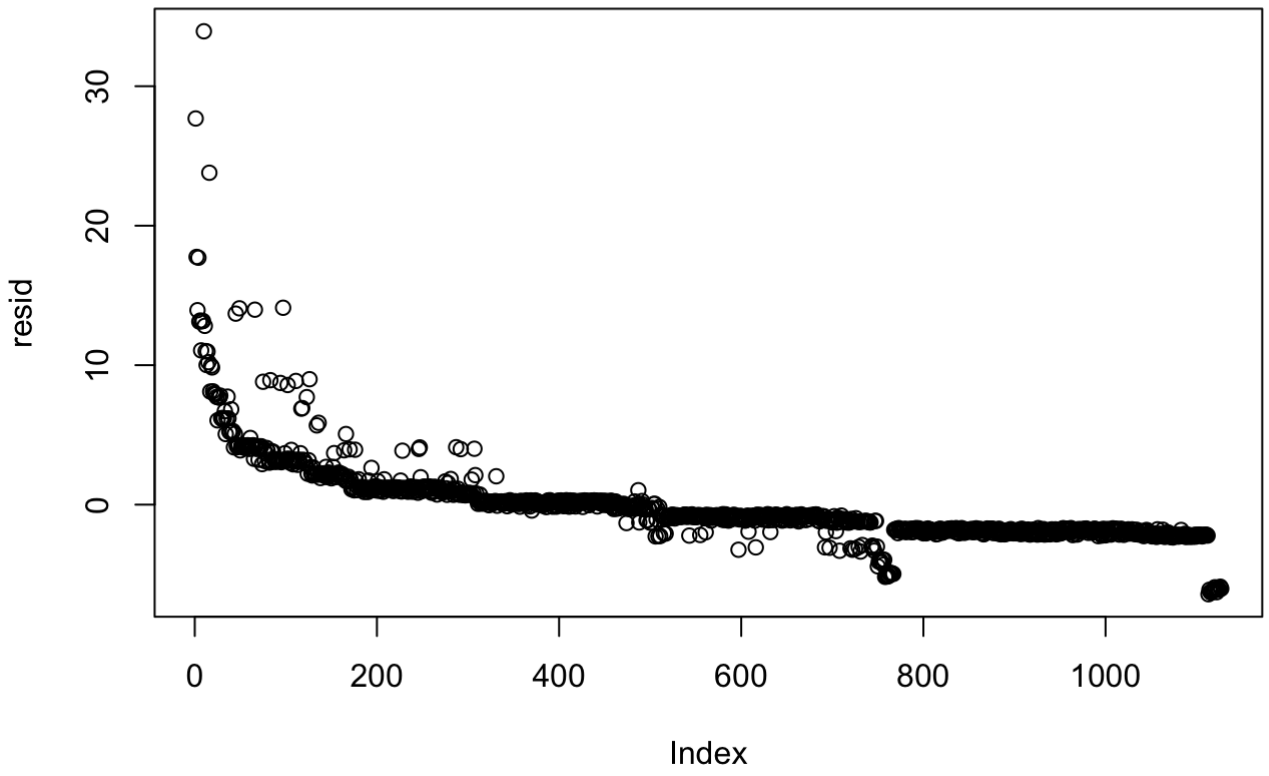
```
(pmed_coef = apply(mod_csim, 2, median))
```

```
##        b_age       b_badh       b_intx          int
##  0.008375506  1.557530512 -0.010684466  0.353067292
```

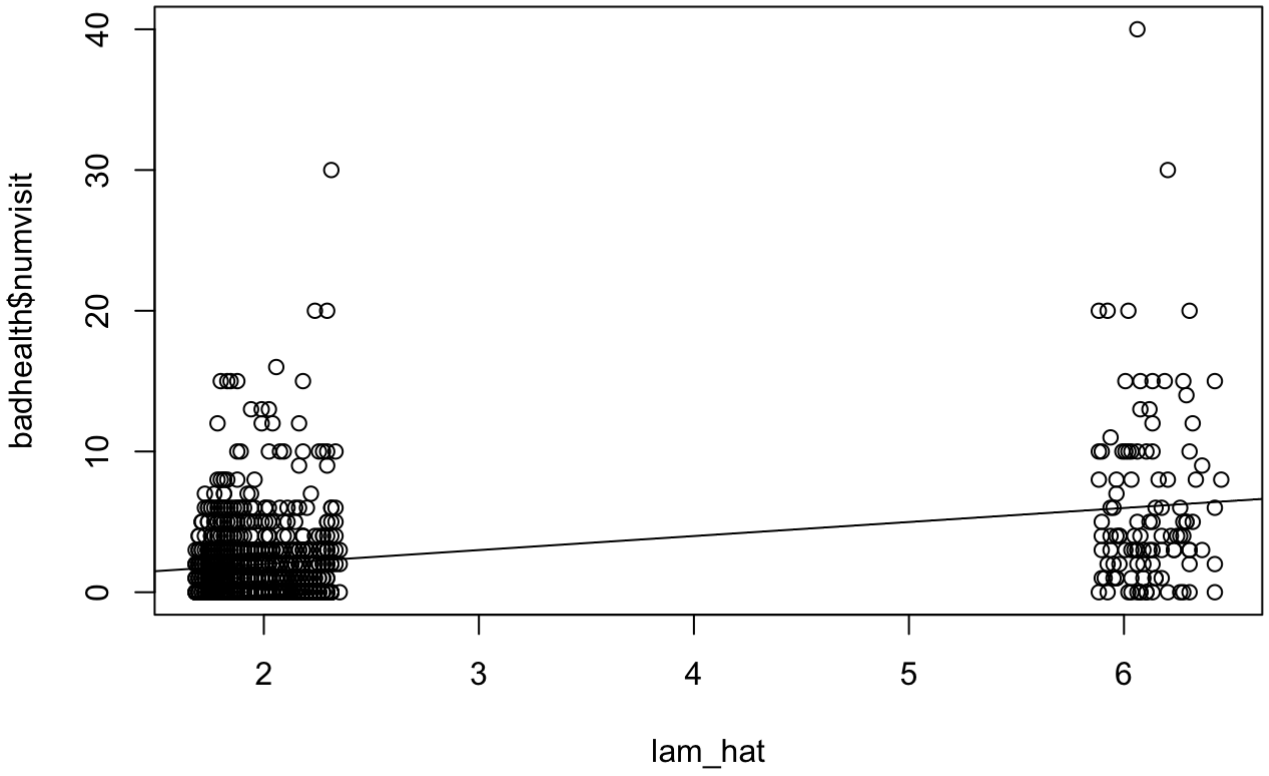
```
llam_hat = pmed_coef["int"] + X %*% pmed_coef[c("b_badh", "b_age", "b_intx")]
lam_hat = exp(llam_hat)

hist(lam_hat)
```



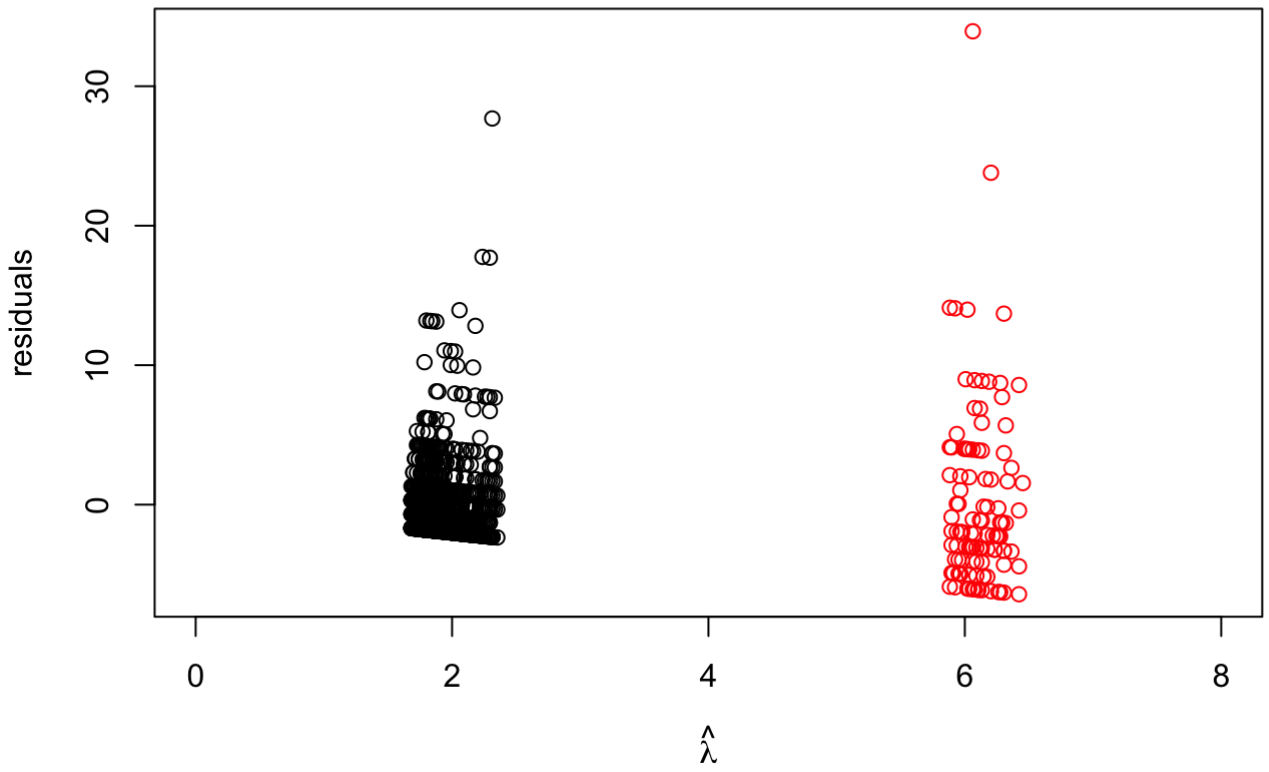

```
resid = badhealth$numvisit - lam_hat
plot(resid) # the data were ordered
```




```
plot(lam_hat, badhealth$numvisit)
abline(0.0, 1.0)
```




```
plot(lam_hat[which(badhealth$badh==0)], resid[which(badhealth$badh==0)], xlim=c(0, 8), y
lab="residuals", xlab=expression(hat(lambda)), ylim=range(resid))
points(lam_hat[which(badhealth$badh==1)], resid[which(badhealth$badh==1)], col="red")
```



It is not surprising that the variability increases for values predicted at higher values since the mean is also the variance in the Poisson distribution. However, observations predicted to have about two visits should have variance about two, and observations predicted to have about six visits should have variance about six.

```
var(resid[which(badhealth$badh==0)])
```

```
## [1] 7.022589
```

```
var(resid[which(badhealth$badh==1)])
```

```
## [1] 41.19617
```

Clearly this is not the case with these data. This indicates that either the model fits poorly (meaning the covariates don't explain enough of the variability in the data), or the data are "overdispersed" for the Poisson likelihood we have chosen. This is a common issue with count data. If the data are more variable than the Poisson likelihood would suggest, a good alternative is the negative binomial distribution, which we will not pursue here.

Lesson 10.3

Results

Assuming the model fit is adequate, we can interpret the results.

```
summary(mod_sim)
```

```
##
## Iterations = 2001:7000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean       SD  Naive SE Time-series SE
## b_age    0.008296 0.002030 1.657e-05      0.0001304
## b_badh   1.555741 0.178620 1.458e-03      0.0126779
## b_intx -0.010619 0.004133 3.374e-05      0.0002945
## int     0.354550 0.078629 6.420e-04      0.0049361
##
## 2. Quantiles for each variable:
##
##             2.5%       25%       50%       75%    97.5%
## b_age    0.00399  0.006994  0.008376  0.009664  0.01215
## b_badh   1.19779  1.438963  1.557531  1.674431  1.89903
## b_intx -0.01861 -0.013398 -0.010684 -0.007903 -0.00243
## int     0.20326  0.302091  0.353067  0.405186  0.51951
```

The intercept is not necessarily interpretable here because it corresponds to a healthy 0-year-old, whereas the youngest person in the data set is 20 years old.

For healthy individuals, it appears that age has a positive association with number of doctor visits. Clearly, bad health is associated with an increase in expected number of visits. The interaction coefficient is interpreted as an adjustment to the age coefficient for people in bad health. Hence, for people with bad health, age is essentially unassociated with number of visits.

Predictive distributions

Let's say we have two people aged 35, one in good health and the other in poor health. What is the posterior probability that the individual with poor health will have more doctor visits? This goes beyond the posterior probabilities we have calculated comparing expected responses in previous lessons. Here we will create Monte Carlo samples for the responses themselves. This is done by taking the Monte Carlo samples of the model parameters, and for each of those, drawing a sample from the likelihood. Let's walk through this.

First, we need the x values for each individual. We'll say the healthy one is Person 1 and the unhealthy one is Person 2. Their x values are:

```
x1 = c(0, 35, 0) # good health
x2 = c(1, 35, 35) # bad health
```

The posterior samples of the model parameters are stored in `mod_csim`:

```
head(mod_csim)
```

```
## Markov Chain Monte Carlo (MCMC) output:
## Start = 1
## End = 7
## Thinning interval = 1
##           b_age   b_badh      b_intx       int
## [1,] 0.01068516 1.720881 -0.01379178 0.2356173
## [2,] 0.01065240 1.716152 -0.01350241 0.2603712
## [3,] 0.01083776 1.654726 -0.01325925 0.2914893
## [4,] 0.01106869 1.613979 -0.01214153 0.2695708
## [5,] 0.01014217 1.603212 -0.01026622 0.2562006
## [6,] 0.01003312 1.585700 -0.01155382 0.2651380
## [7,] 0.01098848 1.596662 -0.01126325 0.2625073
```

First, we'll compute the linear part of the predictor:

```
loglam1 = mod_csim[,"int"] + mod_csim[,c(2,1,3)] %*% x1
loglam2 = mod_csim[,"int"] + mod_csim[,c(2,1,3)] %*% x2
```

Next we'll apply the inverse link:

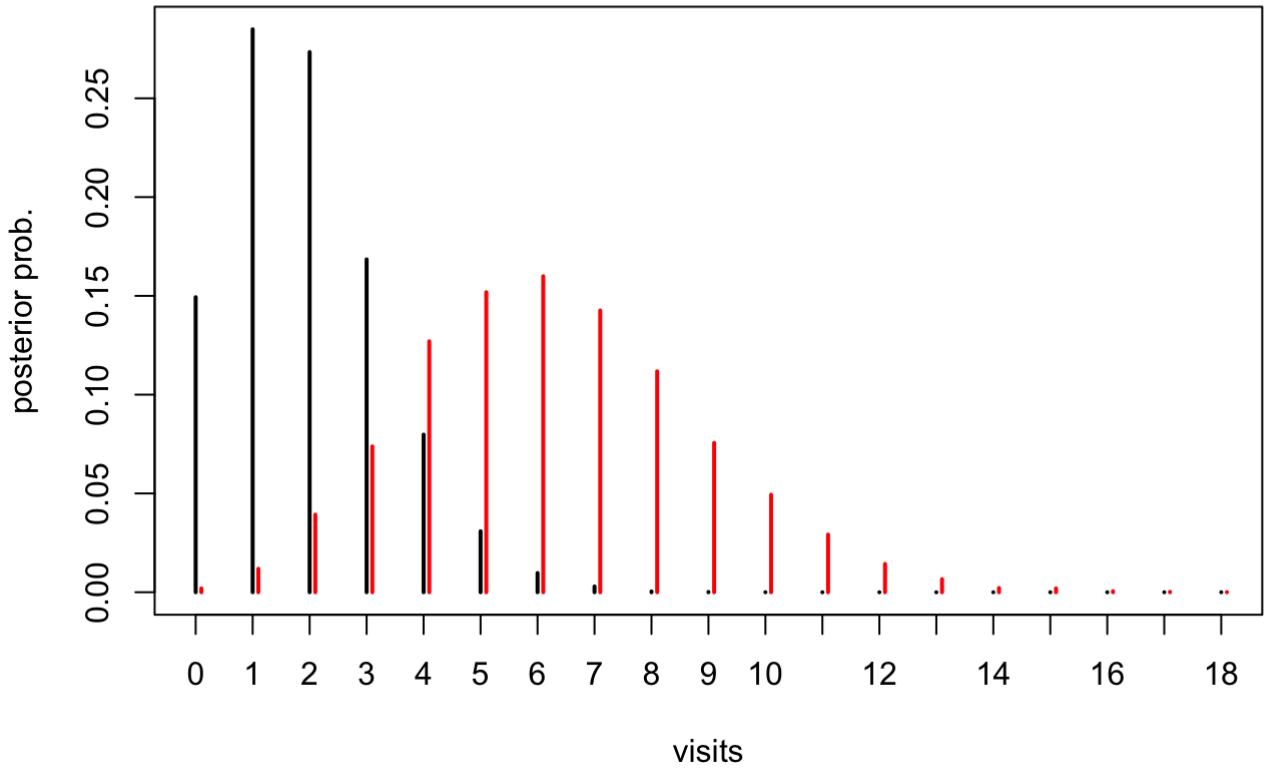
```
lam1 = exp(loglam1)
lam2 = exp(loglam2)
```

The final step is to use these samples for the λ parameter for each individual and simulate actual number of doctor visits using the likelihood:

```
(n_sim = length(lam1))
```

```
## [1] 15000
```

```
y1 = rpois(n=n_sim, lambda=lam1)
y2 = rpois(n=n_sim, lambda=lam2)

plot(table(factor(y1, levels=0:18))/n_sim, pch=2, ylab="posterior prob.", xlab="visits")
points(table(y2+0.1)/n_sim, col="red")
```



Finally, we can answer the original question: What is the probability that the person with poor health will have more doctor visits than the person with good health?

```
mean(y2 > y1)
```

```
## [1] 0.9202
```

Because we used our posterior samples for the model parameters in our simulation, this posterior predictive distribution on the number of visits for these two new individuals naturally takes into account our uncertainty in the model estimates. This is a more honest/realistic distribution than we would get if we had fixed the model parameters at their MLE or posterior means and simulated data for the new individuals.