Likelihood and approximate likelihood intervals

```
form <- sale ~ offset(log(normalSale)) + store  + ad +
  ns(discount, knots = c(20, 30, 40), Boundary.knots = c(0, 50)) - 1
vegetablesGlm <- glm(form,
                     family = poisson,
                     data = vegetables)
confint.default(vegetablesGlm, "ad1")

##     2.5 % 97.5 %
## ad1  0.45   0.56
```

```
confint(vegetablesGlm, "ad1")

## Waiting for profiling to be done...

##  2.5 % 97.5 %
##   0.45   0.56
```

Deviance tests

```
x <- factor(rbinom(10, 2, 0.5))
y <- rpois(10, 1 + c(0, 1, 2)[x])
testGlm <- glm(y ~ x, family = poisson)
testGlmNull <- glm(y ~ 1, family = poisson)
anova(testGlmNull, testGlm, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: y ~ 1
## Model 2: y ~ x
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         9       9.75
## 2         8       8.04  1     1.71     0.19
```

Is the deviance test approximately χ^2-distributed?

Sampling under the null

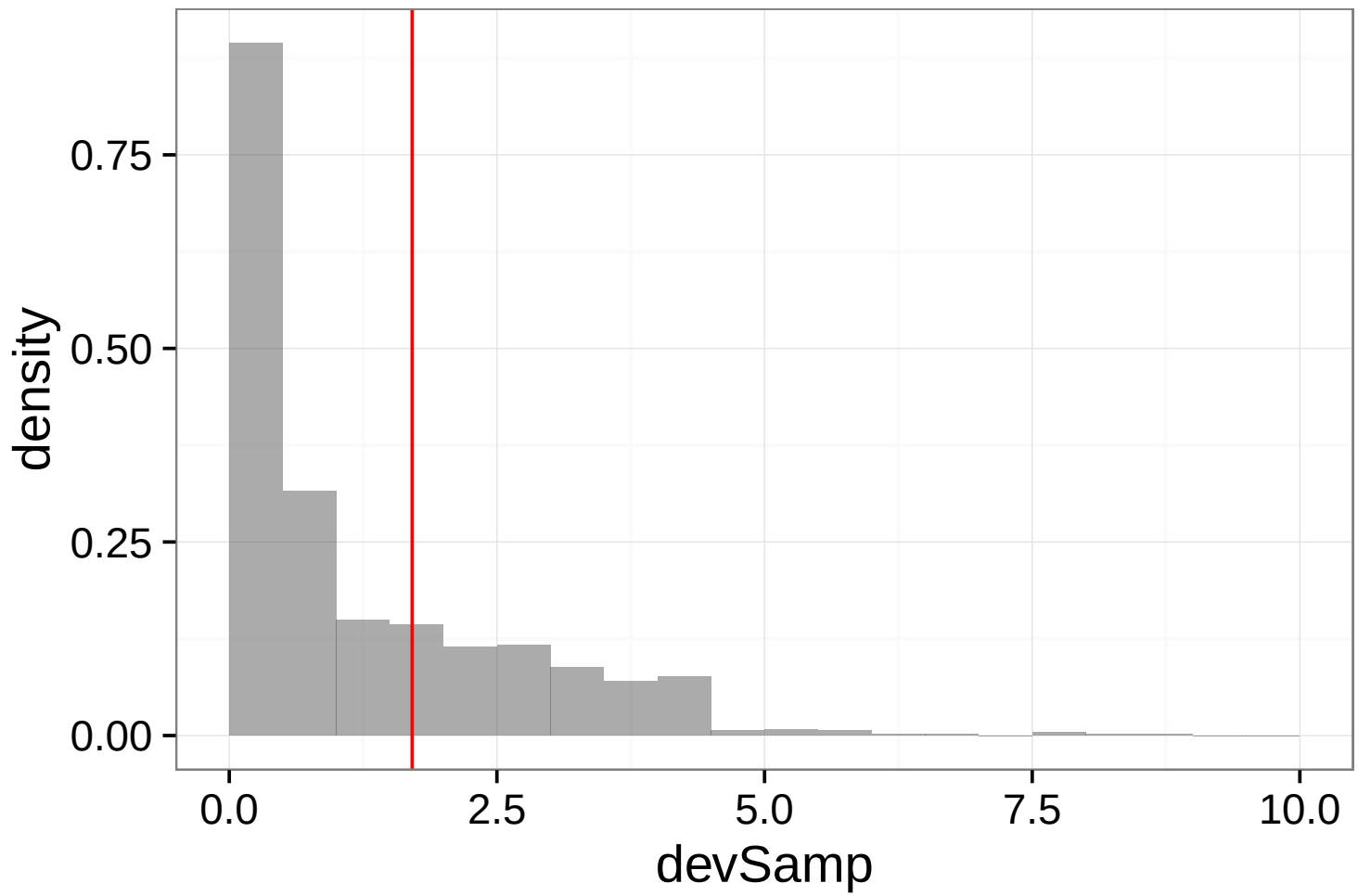
```
B <- 999
devSamp <- numeric(B)
beta0 <- coefficients(testGlmNull)
for (b in 1:B) {
  y <- rpois(10, beta0)
  ## Null model deviance
  dev0 <- glm(y ~ 1, family = poisson)$deviance
  ## Model deviance
  dev <- glm(y ~ x, family = poisson)$deviance
  devSamp[b] <- dev0 - dev
}
```

Sampling under the null

```
obsDev <- anova(testGlmNull, testGlm, test = "Chisq")[2, "Deviance"]
p <- qplot(devSamp, ..density..,
           geom = "histogram", bins = 20,
           alpha = I(0.5)) +
  xlim(0, 10)
```

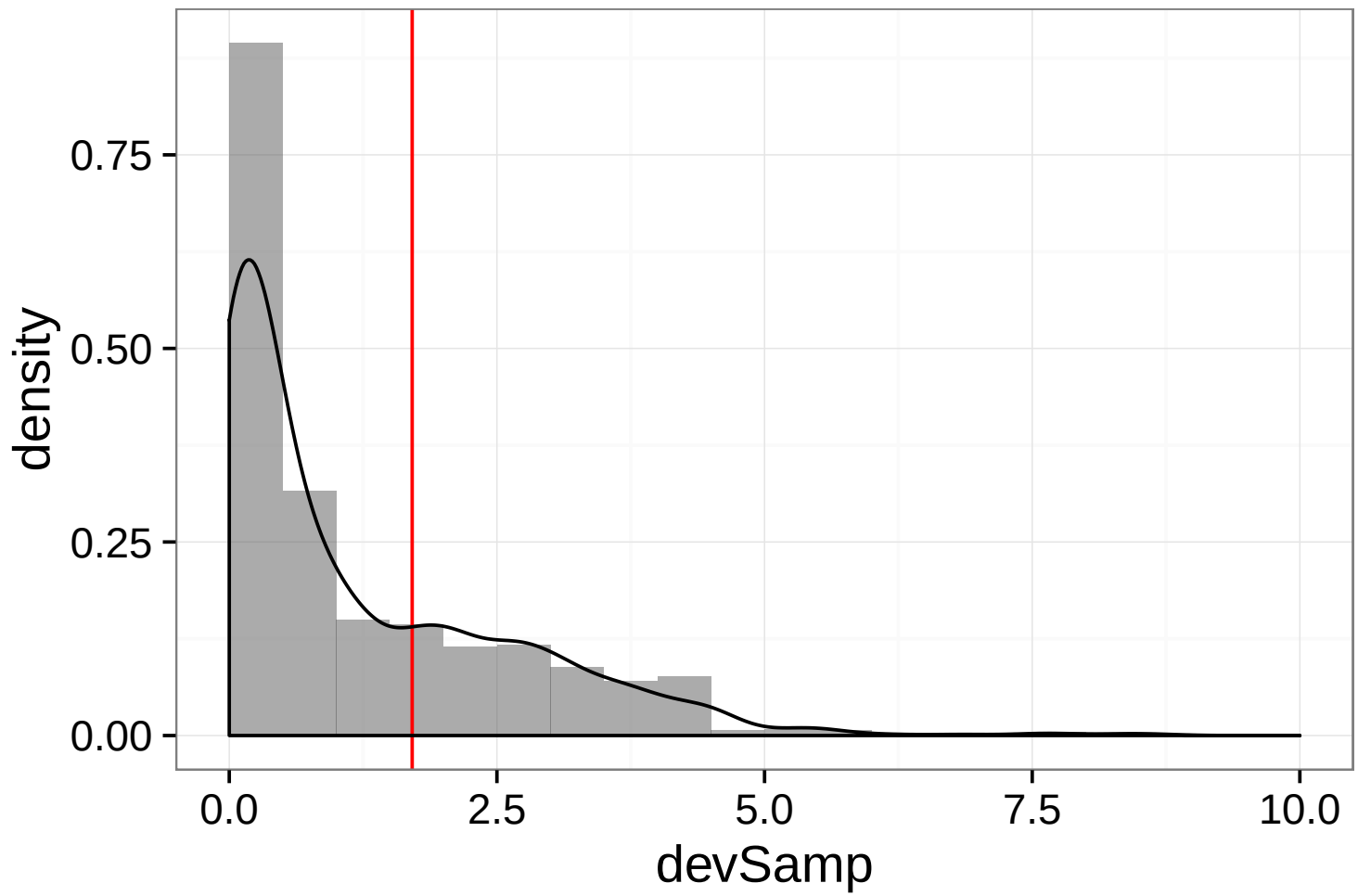
Sampling under the null

```
p <- p + geom_vline(xintercept = obsDev, color = "red")
p
```



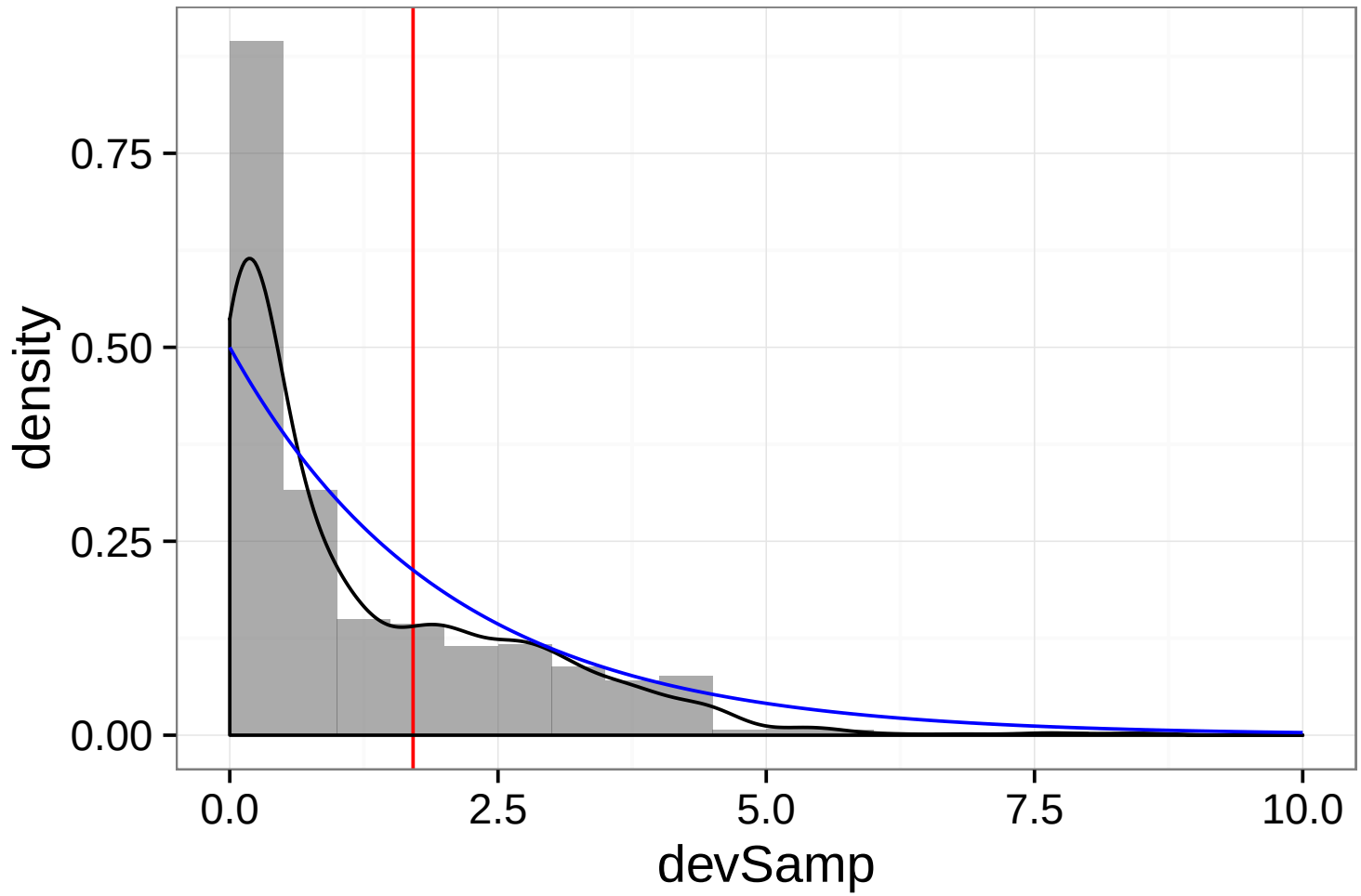
Sampling under the null

```
p <- p + geom_density()
p
```



Sampling under the null

```
xx <- seq(0, 10, 0.01)
p + geom_line(aes(xx, dchisq(xx, 2)), color = "blue")
```



Sampling under the null

When sampling the distribution of a test statistic we sample from the model fitted under the null.

If the test statistic is (approximately) pivot it does not matter (that much) if we don't sample from the "true" model.

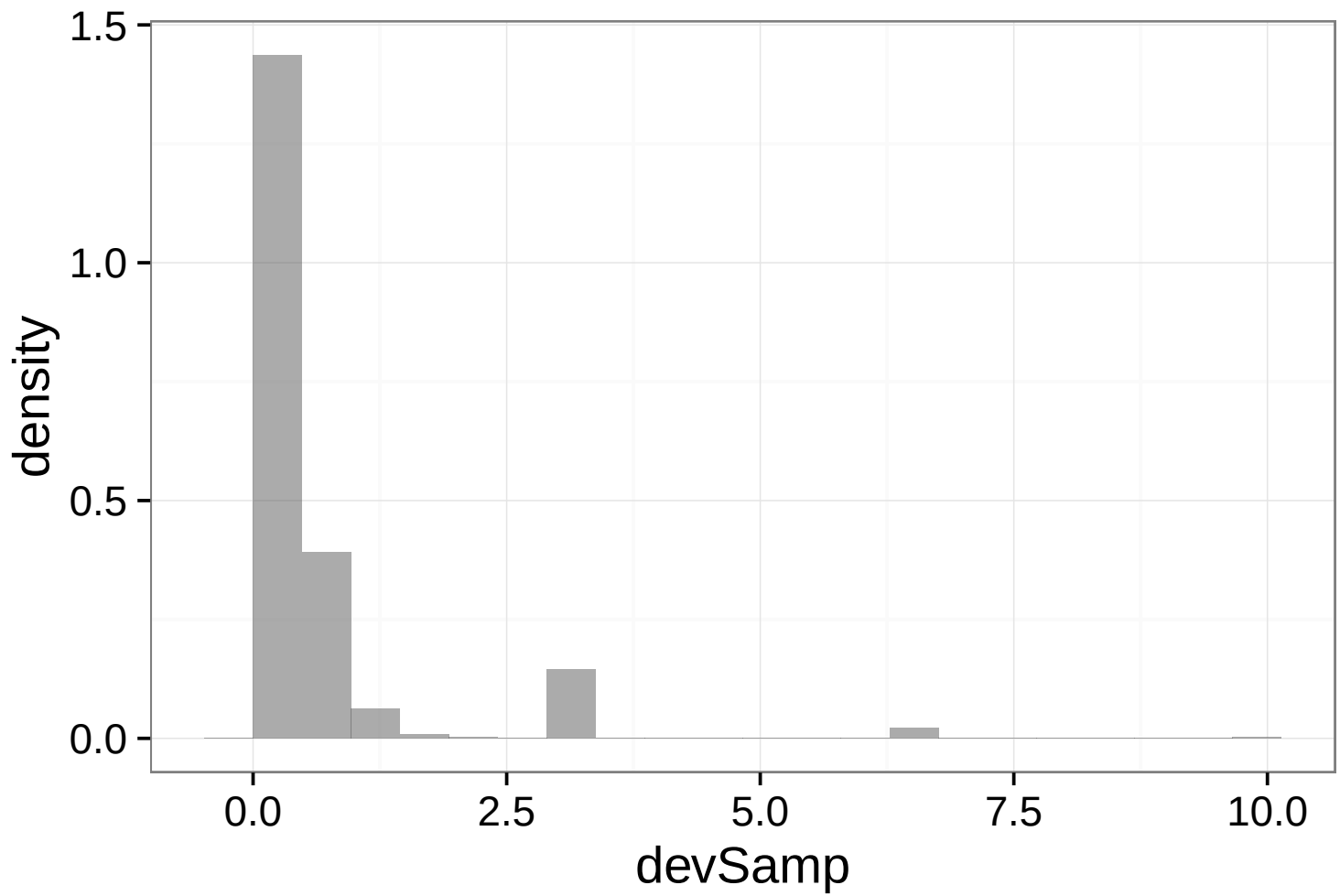
Technically, if the null is wrong, rejecting the test is correct. However, the mean could be correct under the null, while it is the Poisson assumption or the variance assumption that is wrong.

Sampling under the null

```
B <- 999
devSamp <- numeric(B)
## Different null model!
beta0 <- 0.1
for (b in 1:B) {
  y <- rpois(10, beta0)
  ## Null model deviance
  dev0 <- glm(y ~ 1, family = poisson)$deviance
  ## Model deviance
  dev <- glm(y ~ x, family = poisson)$deviance
  devSamp[b] <- dev0 - dev
}
```

The deviance test is not a pivot

```
qplot(devSamp, ..density..,
          geom = "histogram", bins = 20,
          alpha = I(0.5))
```



Bootstrapping

Roughly, bootstrapping replaces analytic approximations by simulations – specifically by resampling new datasets.

- Sample B new data sets $\mathbf{Y}_1^*, \ldots, \mathbf{Y}_B^*$.
- Compute a combinant of interest from each resampled data set.

Different choices of combinants can produce estimates of

- standard errors,
- bias,
- quantiles and confidence intervals.

Estimating the standard error

From the resampled data $\mathbf{Y}_1^*, \ldots, \mathbf{Y}_B^*$ we compute reestimates

$$\hat{\gamma}_b = \hat{\gamma}(\mathbf{Y}_b^*).$$

Then we can estimate the standard error of $\hat{\gamma}$ as

$$\hat{\text{se}} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\gamma}_b - \overline{\gamma})^2}$$

with $\overline{\gamma} = \frac{1}{B}\sum_{b=1}^{B}\hat{\gamma}_b$.

We can use $\hat{\text{se}}$ to construct standard confidence intervals.
Purpose: Can be applied when we don't have an analytic formula or approximation of the standard error. Here, $B = 200$ is usually enough for acceptable precision.

Simple confidence intervals

For confidence intervals a simple combinant is

$$R(\mathbf{Y}, \gamma) = \hat{\gamma} - \gamma.$$

With w_α and z_α the α and $1 - \alpha$ quantiles for R the nominal level $(1 - 2\alpha)$-confidence interval is

$$\{\gamma \in \mathbb{R} \mid \hat{\gamma} - \gamma \in [w_\alpha, z_\alpha]\} = [\hat{\gamma} - z_\alpha, \hat{\gamma} - w_\alpha]$$

We compute estimates, $\hat{w}_\alpha$ and $\hat{z}_\alpha$, as empirical quantiles for the bootstrapped sample of the combinant

$$\hat{\gamma}(\mathbf{Y}_1^*) - \hat{\gamma}, \ldots, \hat{\gamma}(\mathbf{Y}_B^*) - \hat{\gamma}.$$

This is simple, but approximate pivotality of R is doubtful.

Bootstrapping other combinants

We can bootstrap the combinant

$$R(\mathbf{Y}, \gamma) = \frac{(\hat{\gamma} - \gamma)^2}{\hat{\text{se}}(\mathbf{Y})^2}$$

and get the bootstrapped sample

$$\frac{(\hat{\gamma}_1^* - \hat{\gamma})^2}{\hat{\text{se}}(\mathbf{Y}_1^*)^2}, \ldots, \frac{(\hat{\gamma}_B^* - \hat{\gamma})^2}{\hat{\text{se}}(\mathbf{Y}_B^*)^2}.$$

From this we compute the estimate $\hat{c}_\alpha$ as the empirical $(1-\alpha)$-quantile and the correponding confidence interval.

The quality of the confidence interval still depends upon approximate pivotality, but not on an approximate χ^2-distribution.

Parametric bootstrapping

We have a parameter γ of interest and possibly additional parameters θ and their estimates $\hat{\gamma}$ and $\hat{\theta}$ based on the dataset

$$\mathbf{Y} = (Y_1, \ldots, Y_n)$$

with predictors $X_1, \ldots, X_n$.

Parametric bootstrapping: We simulate B new datasets, each from the model with density

$$f(\mathbf{Y}^*; \hat{\gamma}, \hat{\theta}) = \prod_{i=1}^{n} f(Y_i^*; X_i, \hat{\gamma}, \hat{\theta}).$$

The predictors $X_1, \ldots, X_n$ are held fixed in the simulation above – X_i enters in the distribution of Y_i^*.

Parametric bootstrapping

```
B <- 999
n <- nrow(vegetables)
parbeta <- numeric(B)
vegetablesSamp <- vegetables
for(b in 1:B) {
  vegetablesSamp$sale <- simulate(vegetablesGlm)[, 1]
  bootGlm <- glm(form,
                 family = poisson,
                 data = vegetablesSamp)
  parbeta[b] <- coefficients(bootGlm)["ad1"]
}
```

Pair sampling – non-parametric bootstrapping

Consider the collection of observations, responses and predictors,

$$(Y_1, X_1), \ldots, (Y_n, X_n).$$

Non-parametric bootstrapping: Simulate B new datasets, each as

$$(Y_1^*, X_1^*), \ldots, (Y_n^*, X_n^*)$$

with (Y_i^*, X_i^*) chosen uniformly at random with replacement from the original dataset.

This version of non-parametric bootstrapping consists of B samples of size n from the pair empirical measure

$$\sum_{i=1}^{n} \epsilon_{(Y_i, X_i)}.$$

It can be implemented by sampling indices $1, \ldots, n$ uniformly.

Pair sampling – non-parametric bootstrapping

```
beta <- numeric(B)
for(b in 1:B) {
  i <- sample(n, n, replace = TRUE)
  bootGlm <- glm(form,
                 family = poisson,
                 data = vegetables[i, ])
  beta[b] <- coefficients(bootGlm)["ad1"]
}
```

Standard errors

```
sebeta <- sd(beta)
separbeta <- sd(parbeta)
## Standard error based on nonparametric bootstrapping
sebeta

## [1] 0.17

## Standard error based on parametric bootstrapping
separbeta

## [1] 0.029

## Standard error based on ordinary analytic approximations
coef(summary(vegetablesGlm))["ad1", 2]

## [1] 0.03
```

Confidence intervals

```
confint.default(vegetablesGlm, "ad1")
```

```
##     2.5 % 97.5 %
## ad1  0.45   0.56
```

```
betahat <- coefficients(vegetablesGlm)["ad1"]
betahat + 1.96 * sebeta * c(-1, 1)
```

```
## [1] 0.17 0.85
```

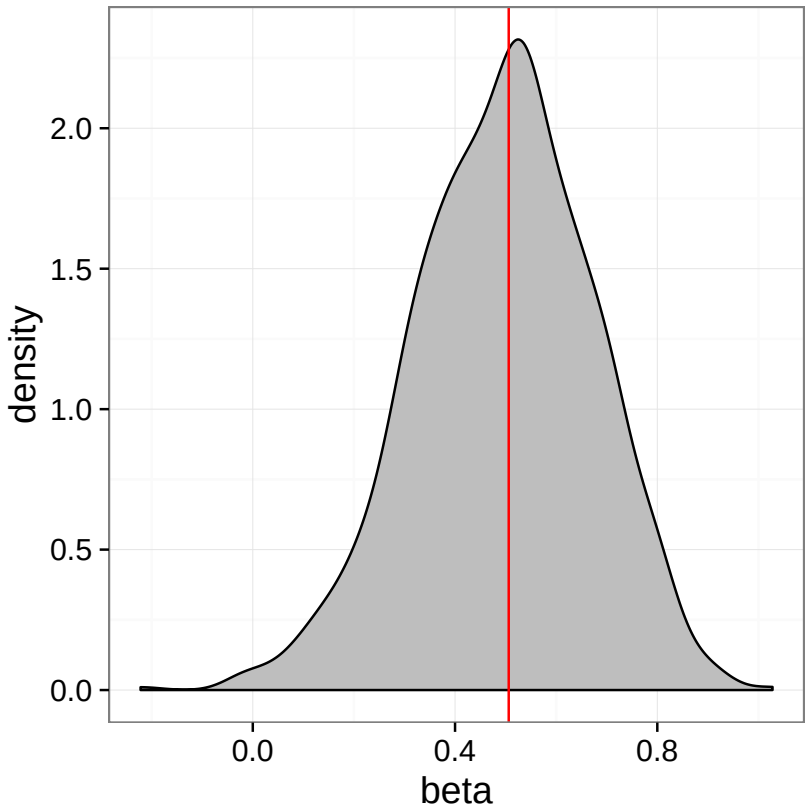
Confidence intervals

```
betastar <- beta - betahat
qbeta <- quantile(betastar, probs = c(0.975, 0.025), type = 1)
betahat - qbeta

##  98% 2.5%
## 0.20 0.88
```

Distribution of bootstrap samples

```
qplot(beta, fill = I('gray'), geom = "density") +
  geom_vline(aes(xintercept=betahat), color = "red")
```



Quasi Poisson intervals

```
vegetablesGlm2 <- glm(form,
                      family = quasipoisson,
                      data = vegetables)
confint(vegetablesGlm2, "ad1")

## Waiting for profiling to be done...

##  2.5 % 97.5 %
##   0.31   0.71
```

Gamma model intervals

```
vegetablesGlm3 <- glm(form,
                      family = Gamma("log"),
                      data = vegetables)
confint(vegetablesGlm3, "ad1")

## Waiting for profiling to be done...

##  2.5 % 97.5 %
##   0.11   0.58
```

Residual sampling

Consider

$$Y_i = \mu(X_i^T \gamma) + \varepsilon_i$$

with $\varepsilon_1, \ldots, \varepsilon_n$ fulfilling distributional assumptions A1, A2 and A4. The raw residuals are

$$e_i = y_i - \mu(X_i^T \hat{\gamma}) = y_i - \hat{\mu}_i.$$

Non-parametric bootstrapping of residuals: A single new dataset is constructed as follows:

- Sample n residuals $e_1^*, \ldots, e_n^*$ from $\{e_1, \ldots, e_n\}$ with replacement.
- Construct the new dataset with

$$Y_i^* = \hat{\mu}_i + e_i^*.$$

The resampling is repeated B times. The predictors are held fixed as for parametric bootstrapping.