Day 10: Example

Data set: hormone

Example 9.3 in Efron and Tibshirani

This is a small data set which is a good candidate for regression analysis. A medical device for continuously delivering an anti-inflammatory hormone has been tested on $n = 27$ subjects.

- y_i : amount of hormone remaining in device i after wearing, $i = 1, 2, \ldots, 27$.
- z_i : number of hours the ith device was worn.
- L_i : manufacturing lot of device i.

The devices tested were randomly selected from three different manufacturing lots, called A, B, and C.

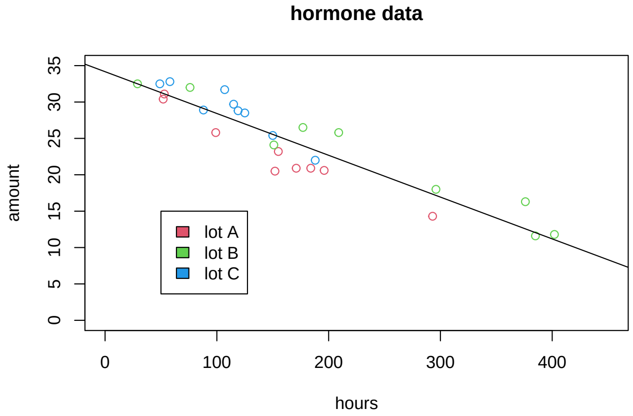
```
library(bootstrap) # hormone data set
attach(hormone)
str(hormone)
```

```
## 'data.frame':    27 obs. of  3 variables:
##  $ Lot   : chr  "A" "A" "A" "A" ...
##  $ hrs   : num  99 152 293 155 196 53 184 171 52 376 ...
##  $ amount: num  25.8 20.5 14.3 23.2 20.6 31.1 20.9 20.9 30.4 16.3 ...
```

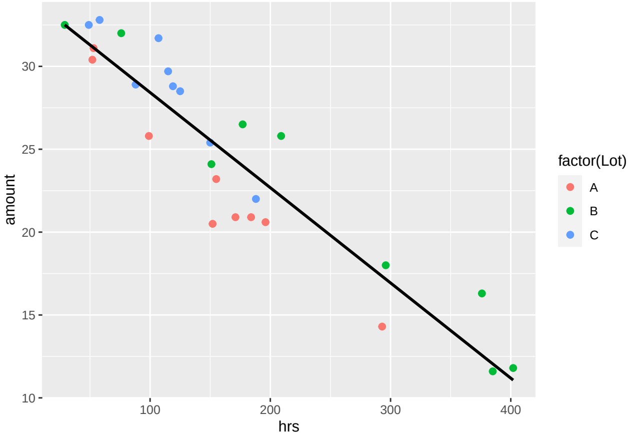
Scatter plot [Figure 9.1]

```
# scatter plot
plot(hrs[1:9], amount[1:9], xlim=c(0,450), ylim=c(0,35), xlab="hours", ylab="amount", col=2)
points(hrs[10:18], amount[10:18], col=3)
points(hrs[19:27], amount[19:27], col=4)
legend(50, 15, legend=c("lot A", "lot B", "lot C"), fill = c(2,3,4))
title("hormone data")
simfit <- lm(amount ~ hrs)
abline(simfit, col= "black")

library(ggplot2)
```



```
ggplot(hormone, aes(hrs, amount)) +
  geom_point(aes(colour = factor(Lot)), size = 2) +
   geom_smooth(method = "lm", se = FALSE, color='black')
```



Estimate coefficients and their standard errors

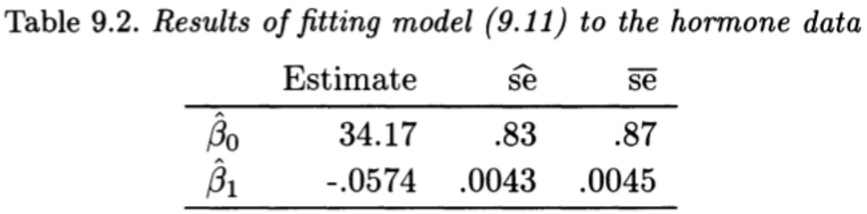


Table 9.2. *Results of fitting model (9.11) to the hormone data*

	Estimate	$\widehat{\text{se}}$	$\overline{\text{se}}$
$\hat{\beta}_0$	34.17	.83	.87
$\hat{\beta}_1$	-.0574	.0043	.0045

```
summary(simfit)
```

```
##
## Call:
## lm(formula = amount ~ hrs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9357 -1.7282 -0.0229  1.7388  3.7323
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.167528   0.867197   39.40  < 2e-16 ***
## hrs         -0.057446   0.004464  -12.87 1.58e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.378 on 25 degrees of freedom
## Multiple R-squared:  0.8688, Adjusted R-squared:  0.8636
## F-statistic: 165.6 on 1 and 25 DF,  p-value: 1.584e-12
```

Residual resampling

```
library(boot)
fits <- fitted(simfit)
e <- residuals(simfit)
X <- model.matrix(simfit)

boot.sim = function(data, indices) {
  y_b <-  fits + e[indices]
  mod <- lm(y_b ~ X - 1)
  coefficients(mod)
}

sim.boot <- boot(hormone, boot.sim, 800)
sim.boot
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = hormone, statistic = boot.sim, R = 800)
##
##
## Bootstrap Statistics :
##       original        bias    std. error
## t1* 34.1675282  0.0417120810 0.837122892
## t2* -0.0574463 -0.0002296116 0.004249073
```

Observation resampling

```
set.seed(123)
boot.sim0 <- function(data, indices){
  # select obs. in bootstrap sample
  data <- data[indices,]
  mod <- lm(amount ~ hrs, data = data)
  # return coefficient vector
  coefficients(mod)
}

sim0.boot <- boot(hormone, boot.sim0, 800)
sim0.boot # the result (9.33), not much different than Table 9.2
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = hormone, statistic = boot.sim0, R = 800)
##
##
## Bootstrap Statistics :
##       original        bias    std. error
## t1* 34.1675282  0.0514136225 0.731098155
## t2* -0.0574463 -0.0004888482 0.004365065
```

An analysis of covariance model with adding lot as a factor

The coefficients for the different levels of the factor can be thought of as intercepts for three different regression lines (all having common slope) — one for each lot.

```
lot <- factor(Lot)
aoc <- lm(amount ~ - 1 + hrs + lot) # a different intercept for each lot
summary(aoc)
```

```
##
## Call:
## lm(formula = amount ~ -1 + hrs + lot)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9245 -1.0626 -0.1304  0.8544  2.8061
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## hrs  -0.060136   0.003474  -17.31  1.1e-14 ***
## lotA 32.131595   0.748277   42.94  < 2e-16 ***
## lotB 36.105095   0.971643   37.16  < 2e-16 ***
## lotC 35.597324   0.659579   53.97  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.605 on 23 degrees of freedom
## Multiple R-squared:  0.9966, Adjusted R-squared:  0.996
## F-statistic:  1695 on 4 and 23 DF,  p-value: < 2.2e-16
```

```
detach(hormone)
```

Data set: cell (Cell Survival data)

Example 9.6 in Efron and Tibshirani

There are regression situations where the covariates are more nat- urally considered fixed rather than random. A radiologist has run an experiment involving 14 bacterial plates. The plates were exposed to various doses of radiation, and the proportion of the surviving cells measured. Greater doses lead to smaller survival proportions, as would be expected. The question mark after the response for plate 13 reflects some uncertainty in that result expressed by the investigator.

- y_i : log.surv i.e. log of survival proportion.
- z_i : dose

```
library(bootstrap) # hormone data set
library(MASS) # Use lmsreg
attach(cell)
```

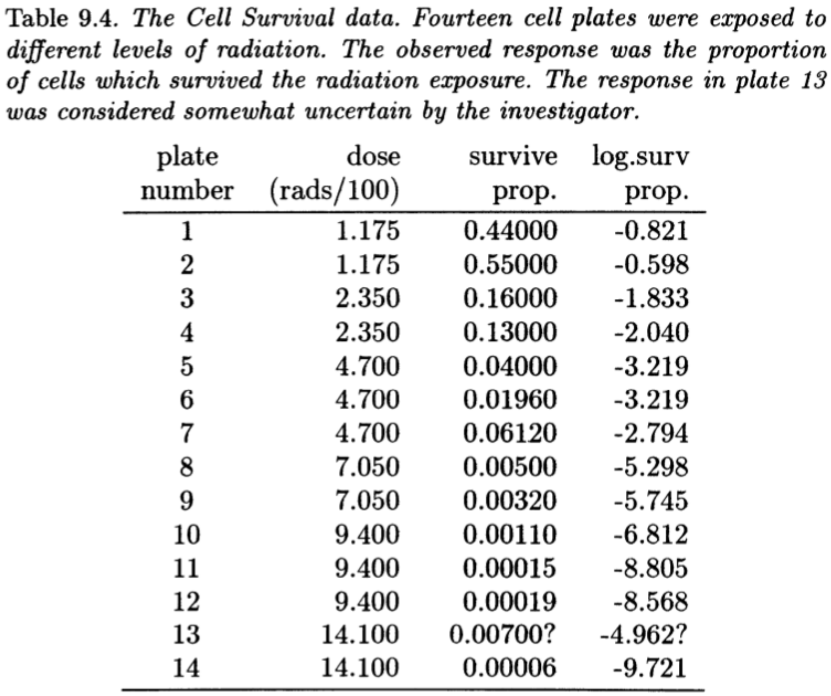


Table 9.4. *The Cell Survival data. Fourteen cell plates were exposed to different levels of radiation. The observed response was the proportion of cells which survived the radiation exposure. The response in plate 13 was considered somewhat uncertain by the investigator.*

plate number	dose (rads/100)	survive prop.	log.surv prop.
1	1.175	0.44000	-0.821
2	1.175	0.55000	-0.598
3	2.350	0.16000	-1.833
4	2.350	0.13000	-2.040
5	4.700	0.04000	-3.219
6	4.700	0.01960	-3.219
7	4.700	0.06120	-2.794
8	7.050	0.00500	-5.298
9	7.050	0.00320	-5.745
10	9.400	0.00110	-6.812
11	9.400	0.00015	-8.805
12	9.400	0.00019	-8.568
13	14.100	0.00700?	-4.962?
14	14.100	0.00006	-9.721

Estimate coefficients and their standard errors

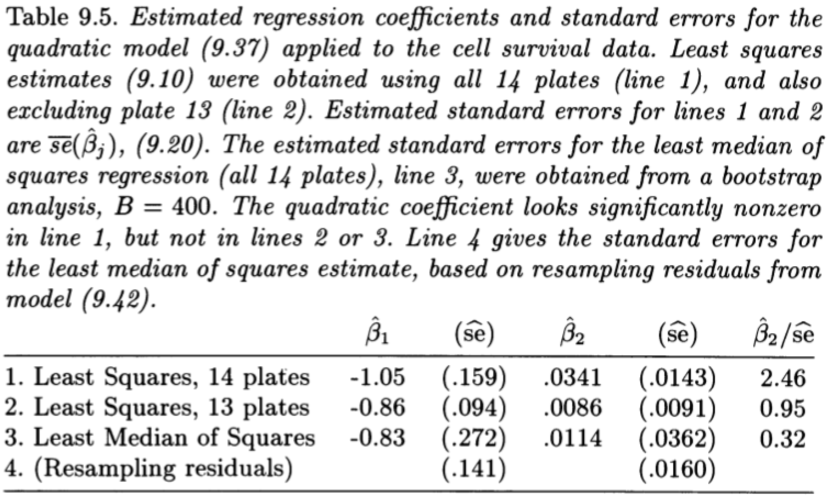


Table 9.5. *Estimated regression coefficients and standard errors for the quadratic model (9.37) applied to the cell survival data. Least squares estimates (9.10) were obtained using all 14 plates (line 1), and also excluding plate 13 (line 2). Estimated standard errors for lines 1 and 2 are $\overline{se}(\hat{\beta}_j)$, (9.20). The estimated standard errors for the least median of squares regression (all 14 plates), line 3, were obtained from a bootstrap analysis, $B = 400$. The quadratic coefficient looks significantly nonzero in line 1, but not in lines 2 or 3. Line 4 gives the standard errors for the least median of squares estimate, based on resampling residuals from model (9.42).*

	$\hat{\beta}_1$	$(\widehat{\text{se}})$	$\hat{\beta}_2$	$(\widehat{\text{se}})$	$\hat{\beta}_2/\widehat{\text{se}}$
1. Least Squares, 14 plates	-1.05	(.159)	.0341	(.0143)	2.46
2. Least Squares, 13 plates	-0.86	(.094)	.0086	(.0091)	0.95
3. Least Median of Squares	-0.83	(.272)	.0114	(.0362)	0.32
4. (Resampling residuals)		(.141)		(.0160)	

plot for comparisons

```
attach(cell)
y <- log.surv
z <- dose
plot(z[-13], y[-13], xlim=c(0.5,14.5), ylim=c(-10,0), xlab="dose", ylab="log proportion alive")
points(z[13], y[13], pch=18, col="hotpink")
text(z[13], y[13], pos = 1, "?")
legend(1, -6, legend=c("fit with 13 points", "fit with 14 points" ,"LMS"), fill=c(4,2,3))
```

```r
title("cell survival data")

# Using all 14 points
ls1 <- lm(y ~ -1 + z) # no intercept
summary(ls1)
```

```
##
## Call:
## lm(formula = y ~ -1 + z)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4459 -0.6965 -0.3467 -0.0295  4.5766
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## z -0.67650    0.05597  -12.09 1.92e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.624 on 13 degrees of freedom
## Multiple R-squared:  0.9183, Adjusted R-squared:  0.912
## F-statistic: 146.1 on 1 and 13 DF,  p-value: 1.916e-08
```

```r
ls2 <- lm(y ~ -1 + z + I(z^2))
summary(ls2) # Table 9.5 first line
```

```
##
## Call:
## lm(formula = y ~ -1 + z + I(z^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.97720 -0.03754  0.30231  0.55116  3.00441
##
## Coefficients:
##        Estimate Std. Error t value Pr(>|t|)
## z      -1.04910    0.15871   -6.61  2.5e-05 ***
## I(z^2)  0.03433    0.01395    2.46     0.03 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.378 on 12 degrees of freedom
## Multiple R-squared:  0.9457, Adjusted R-squared:  0.9366
## F-statistic: 104.5 on 2 and 12 DF,  p-value: 2.566e-08
```

```r
lines(z, ls2$fit, col = 2)

# except 13th point
new.z <- z[-13]
ls3 <- lm(y[-13] ~ -1 + new.z  + I(new.z^2))
summary(ls3) # Table 9.5 second line
```

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

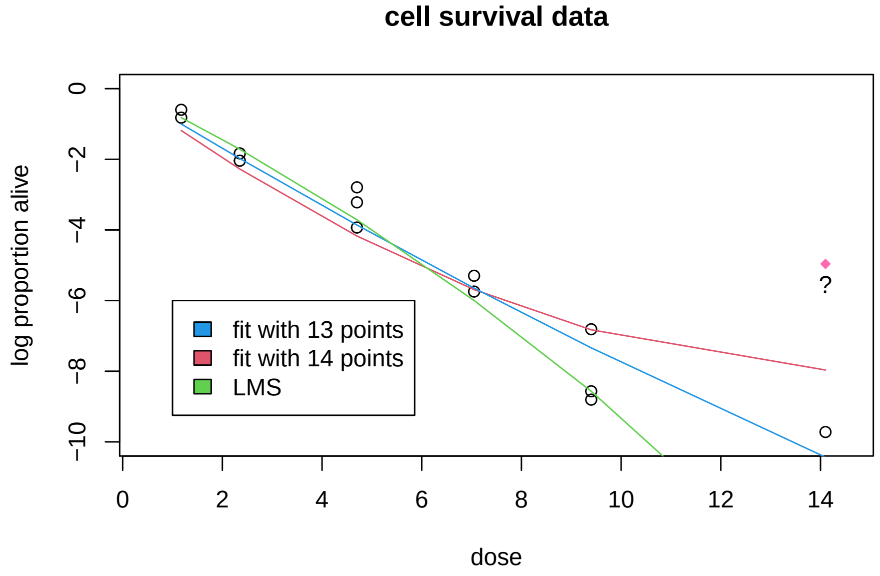
```
## lm(formula = y[-13] ~ -1 + new.z + I(new.z^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.46663 -0.07183  0.17985  0.52637  1.06617
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## new.z      -0.861950   0.094280  -9.142  1.8e-06 ***
## I(new.z^2)  0.008646   0.009076   0.953    0.361
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.76 on 11 degrees of freedom
## Multiple R-squared:  0.9839, Adjusted R-squared:  0.981
## F-statistic: 336.5 on 2 and 11 DF,  p-value: 1.363e-10
```

```
lines(z[-13], ls3$fit, col = 4)

# Using Least Median of Squares  (LMS) estimation
lms <- lmsreg(y ~ -1 + z + I( z^2 ))
lms
```

```
## Call:
## lqs.formula(formula = y ~ -1 + z + I(z^2), method = "lms")
##
## Coefficients:
##        z    I(z^2)
## -0.66833  -0.02587
##
## Scale estimates 0.3743 0.4613
```

```
lines(z, lms$fit, col=3)
```



```
# Delete 13th point
lms2 <- lmsreg( y[-13] ~ -1 + new.z  + I(new.z^2))
lms2
```

```
## Call:
## lqs.formula(formula = y[-13] ~ -1 + new.z + I(new.z^2), method = "lms")
##
## Coefficients:
##      new.z  I(new.z^2)
##   -0.66833    -0.02587
##
## Scale estimates 0.3428 0.3660
```

Plate 13, marked “?” in the plot, has a large effect on the fitted least-squares curve. The questionable point has no effect on the LMS curve. Sample means are sensitive to influential values, but medians are not.

The LMS estimate of $\boldsymbol{\beta}$ is the value $\hat{\boldsymbol{\beta}}$ minimizing the median squared residual (MSR),

$$MSR(\hat{\boldsymbol{\beta}}) = \min_{\boldsymbol{b}}[\mathrm{MSR}(\boldsymbol{b})]$$

where $\mathrm{MSR}(\boldsymbol{b}) = \mathrm{median}(y_i - \boldsymbol{c}_i\boldsymbol{b})^2$.

How accurate are the LMS estimates $\hat{\beta}_1, \hat{\beta}_2$?

There is no neat formula for LMS standard errors. (There is no neat formula for the LMS estimates themselves. They are calculated using a sampling algorithm: see Problem 9.8.) The standard errors in Table 9.5 were obtained by bootstrap methods. The standard errors in line 3 are based on resampling observations.

Observation resampling

```r
boot.lms <- function(data, indices){
  # select obs. in bootstrap sample
  data <- data[indices,]
  mod <- lmsreg(log.surv ~ -1 + dose + I(dose^2) , data = data)
  # return coefficient vector
  return(mod$coefficients)
}

lms.boot <- boot(cell, boot.lms, 500)
lms.boot # similar to  Table 9.5 third line
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = cell, statistic = boot.lms, R = 500)
##
##
## Bootstrap Statistics :
##       original      bias    std. error
## t1* -0.6683283 -0.04172152  0.24093709
## t2* -0.0258682  0.01619735  0.03668268
```

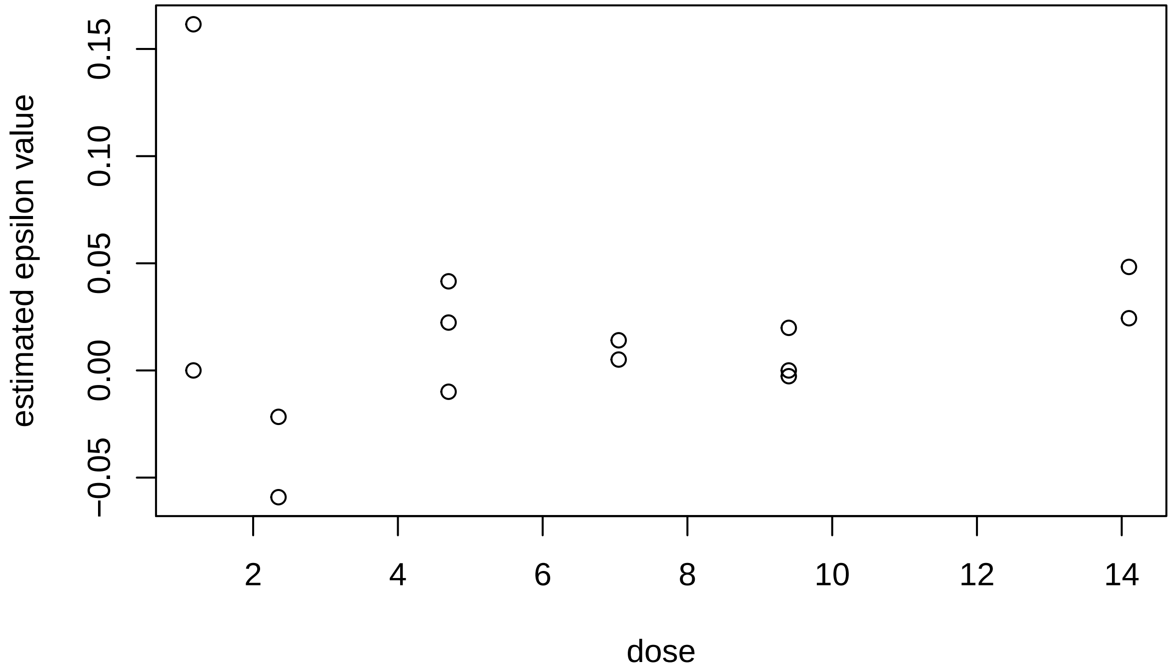
Note that how large the estimated of standard error for the $\hat{\beta}_2$ is relative to the magnitude of the estimate.

Residual resampling with transformed residuals

Looking at the scatter plot, we can see that the response y_i are more dispersed for larger values of z. As a roughly appropriate model, we will assume that the errors from the linear model increase lin- early with the dose z.

$$y_i = \boldsymbol{c_i}\boldsymbol{\beta} + z_i\varepsilon_i \text{ for } i = 1, 2, \ldots, 14.$$

```
new.res <- lms$res/z^2
plot(z, new.res, xlab="dose", ylab="estimated epsilon value")
```



```
boot.lms2 = function(data, indices) {
  y_b <-  lms$fit + z*new.res[indices]
  mod <- lm(y_b ~ z)
  return(mod$coefficients)
}

lms2.boot <- boot(cell, boot.lms2, 200)
lms2.boot
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = cell, statistic = boot.lms2, R = 200)
##
##
## Bootstrap Statistics :
##       original      bias    std. error
## t1*  0.8870745  0.09573535  0.13050079
## t2* -1.0243528 -0.01436623  0.03069656
```

It would be good to make more observations at large dose values in order to better assess the variance structure, and to better assess whether or not the questionable value should be used.

Example of Generalized Linear regression

For generalized linear models one can use various definitions of raw residuals. The raw residuals usually do not have constant variance. Let's consider the modified residuals, $r_i = \frac{e_i}{(1-h_{ii})^{1/2}}$. Approximate leverages are used for generalized linear and nonlinear models, and the modified residuals usually do not have mean zero, so need to be adjusted.

data: leukemiaFZ

Survival times of 33 patients with leukemia (Feigl and Zeelen, 1965). Times are measured in weeks from diagnosis. Reported covariates are white blood cell counts (wbc) and a binary variable AG that indicates a positive or negative test related to the white blood cell characteristics. Three of the observations were censored. The data was taken from Lawless (2003).

```
library(boot)
library(BGPhazard) # data set leukemiaFZ
```

```
##
## Attaching package: 'BGPhazard'

## The following object is masked from 'package:MASS':
##
##     gehan
```

```
fz <- leukemiaFZ
head(fz)
```

```
##   time delta AG   wbc
## 1   65     1  1  2.30
## 2  140     0  1  0.75
## 3  100     1  1  4.30
## 4  134     1  1  2.60
## 5   16     1  1  6.00
## 6  106     0  1 10.50
```

We will fit a model

$$\log \mathrm{time}_i = \beta_0 + \beta_1 \log(\mathrm{wbc}_i/1000) + \beta_2 \mathrm{AG}_i + \varepsilon_i$$

where the ε_i are assumed to have zero mean and constant variance.

```
fz.lm <- glm(log(fz$time) ~ log(fz$wbc/10000)  + fz$AG)
summary(fz.lm)
```

```
##
## Call:
## glm(formula = log(fz$time) ~ log(fz$wbc/10000) + fz$AG)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1707  -0.9300   0.1879   0.6694   2.5628
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -1.3982     1.1088  -1.261  0.21705
## log(fz$wbc/10000)   -0.5638     0.1639  -3.441  0.00173 **
## fz$AG                 0.9723     0.4339   2.241  0.03261 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for gaussian family taken to be 1.544063)
##
##     Null deviance: 74.217  on 32  degrees of freedom
## Residual deviance: 46.322  on 30  degrees of freedom
## AIC: 112.84
##
## Number of Fisher Scoring iterations: 2
```

```r
fz.diag <- glm.diag(fz.lm)
names(fz.diag)
```

```
## [1] "res"  "rd"   "rp"   "cook" "h"    "sd"
```

Observation resampling

```r
set.seed(123)
fz.obs <- function(data, i){
  data <- data[i,]
  mod <- glm(log(time) ~ log(wbc/10000) + AG, data = data)
  return(coef(mod))
}
fz.boot1 <- boot(fz, fz.obs, R=499)
mean(fz.boot1$t[,3])
```

```
## [1] 0.974047
```

```r
sd(fz.boot1$t[,3])
```

```
## [1] 0.4456532
```

Residual resampling

```r
fz.res <- residuals(fz.lm) / sqrt(1 - fz.diag$h)
fz.res <- fz.res - mean(fz.res)
fz.df <- data.frame(fz, res = fz.res, fit = fitted(fz.lm))

fz.model <- function(data, i) {
data$time <- exp(data$fit + data$res[i])
mod <- glm(log(time) ~ log(wbc/10000) + AG, data = data)
return(coef(mod))
}

fz.boot2 <- boot(fz.df, fz.model, R=499)
mean(fz.boot2$t[,3])
```

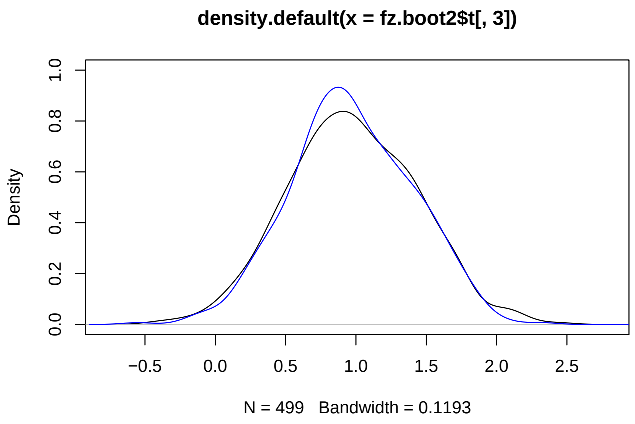
```
## [1] 0.9767127
```

```r
sd(fz.boot2$t[,3])
```

```
## [1] 0.4592704
```

Comparison of the bootstrap distributions for $\hat{\beta}_2$

```r
plot(density(fz.boot2$t[,3]), ylim = c(0, 1))
lines(density(fz.boot1$t[,3]),col="blue")
```


density.default(x = fz.boot2$t[, 3])
Density
0.0
0.2
0.4
0.6
0.8
1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
N = 499   Bandwidth = 0.1193