8 Bivarijantno modeliranje: dvije kontinuirane varijable, 2. dio

Linearna regresija

Zadatak Odredi pravac linearne regresija dobiven pomoću metode najmanjih kvadrata kada prinos slame ovisi o prinosu zrna.

Odredimo model linearne regresije u slučaju kada je $y = ax + b$, koristeći R.

```
x = mhw$grain
y = mhw$straw
model.straw.grain = lm(y ~ x)
model.straw.grain
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##       0.866        1.430
```

Ako nas zanimaju samo koeficijenti:

```
model.straw.grain$coefficients
```

```
## (Intercept)           x
##      0.8663      1.4305
```

ili, na drugačiji način

```
coefficients(model.straw.grain)
```

```
## (Intercept)           x
##      0.8663      1.4305
```

Znači, jednadžba pravca regresije je $y = 1.4305 \cdot x + 0.8663$

Na temelju dobivenog modela `model.straw.grain` možemo, koisteći funkciju `predict()` predviđati vrijednos slame za proizvolju vrijednost zrna. Npr. koliki će biti prinos slame ako je prinos zrna 3.14 ili 4.13?

```
# Pažnja, u data.frame varijabla se mora zvati x, jer je tako definiran
# model.straw.grain
predict(model.straw.grain, data.frame(x = c(3.14, 4.13)))
```

```
##     1     2
## 5.358 6.774
```

Zadatak Iz podataka o prinosima zrnevlja i slame grafički prikažite točke i slici dodajte pravac regresije

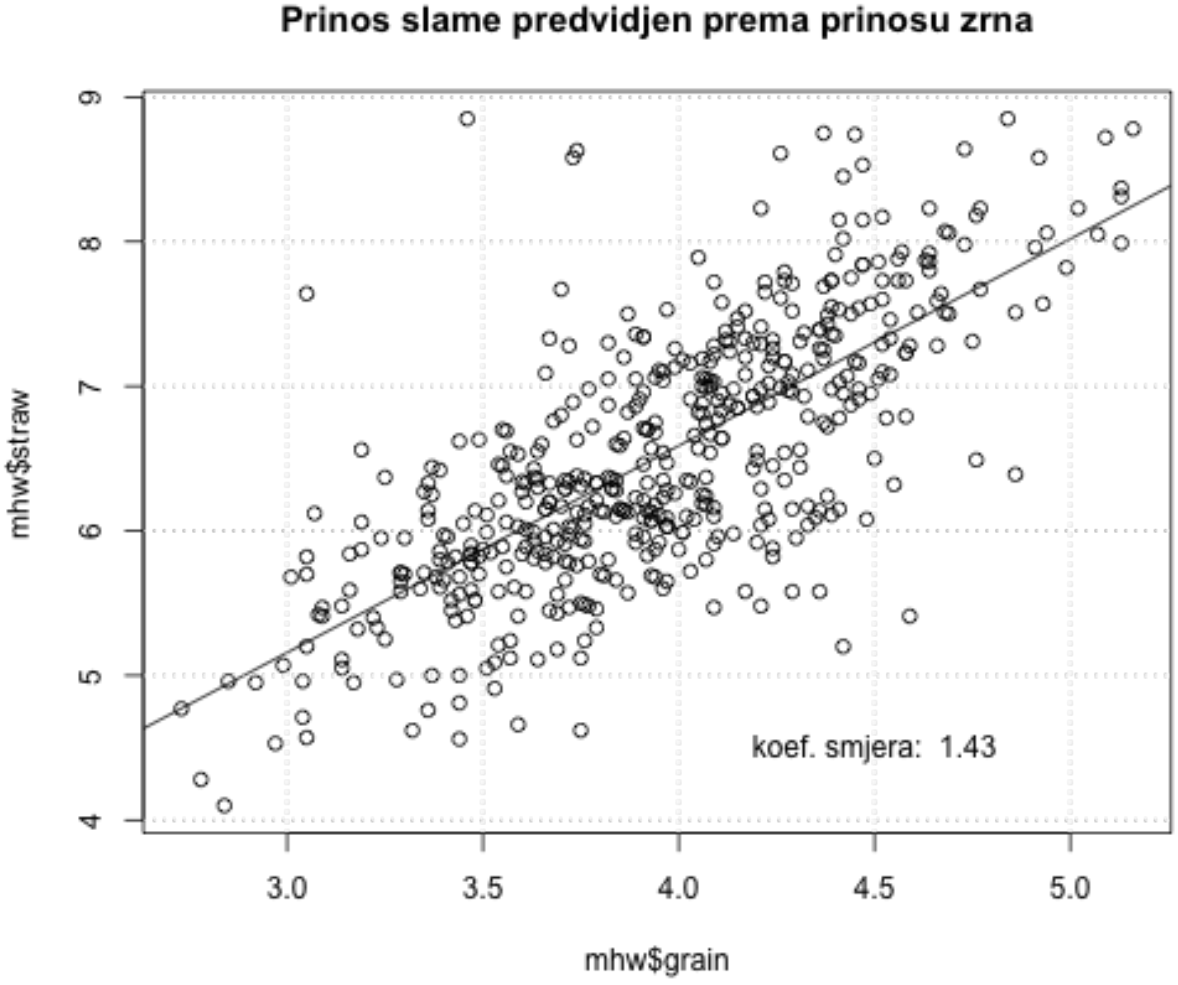


Figure 1: plot of chunk unnamed-chunk-6

```
plot(mhw$straw ~ mhw$grain)
title("Prinos slame predvidjen prema prinosu zrna")
abline(model.straw.grain)
grid()
text(4.5, 4.5, paste("koef. smjera: ",
                     round(coefficients(model.straw.grain)[2], 2)))
```

Ista slika pomocu *ggplot2*.

```
ggplot(mhw, aes(grain, straw)) +
  geom_point(shape=1, size = 3) +    # Use hollow circles
  geom_smooth(method=lm,   # Add linear regression line
              se=FALSE) +   # Don't add shaded confidence region
  labs(title="Prinos slame predvidjen prema prinosu zrna",
       x = "Prinos zrna", y = "Prinos slame") +
  annotate("text", x = 4.5, y = 4.5,
           label = paste("koef. smjera: ",
                     round(coefficients(model.straw.grain)[2], 2))) +
  theme_bw()
```

Pomoću biblioteke *ggplot2* lako je dobiti i grafički prikaz intervala povjerenja za točke na pravcu regresije (podrazumijevanja vrijednost pouzdanosti je 95%).

```
ggplot(mhw, aes(grain, straw)) +
  geom_point(shape=1, size = 3) +    # Use hollow circles
  geom_smooth(method=lm,   # Add linear regression line
              ) +           # Add shaded confidence region
  labs(title="Prinos slame predvidjen prema prinosu zrna",
       x = "Prinos zrna", y = "Prinos slame") +
  annotate("text", x = 4.5, y = 4.5,
           label = paste("koef. smjera: ",
                     round(coefficients(model.straw.grain)[2], 2))) +
  theme_bw()
```

Kako dobiti pdf file

```
list.files()
```

```
system("pandoc --latex-engine=xelatex mhw-08-2.md -o mhw-08-2.pdf -H margins.sty")
```

Napomena. Style file margins.sty ima samo jedan redak i to
`\usepackage[vmargin=1in,hmargin=1in]{geometry}`
radi podrazumijevano prevelikih margina koje radi `pandoc`

Analiza pravca regresije

Podsjetnik: reziduali u regresijskoj analizi određeni su razlikom mjerenih i funkcijskih vrijednosti (modelskih vrijednosti) i trebali bi biti normalno distribuirani.

Zadatak Prikažte histogram i sliku kvantil - kvantil za regresijske reziduale. Sumarno, numerički, prikažite osnovne podatke o rezidualima.

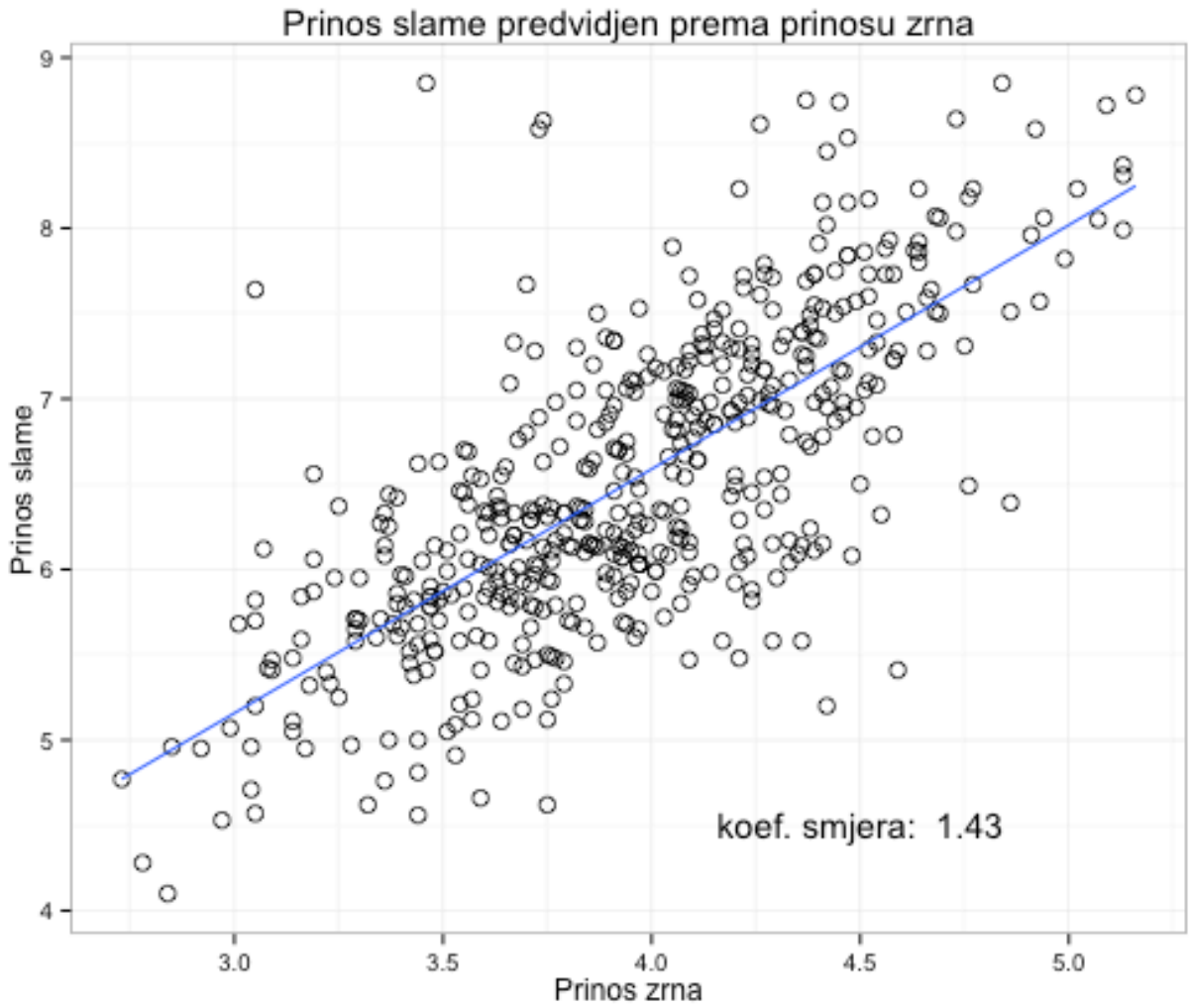


Figure 2: plot of chunk unnamed-chunk-7

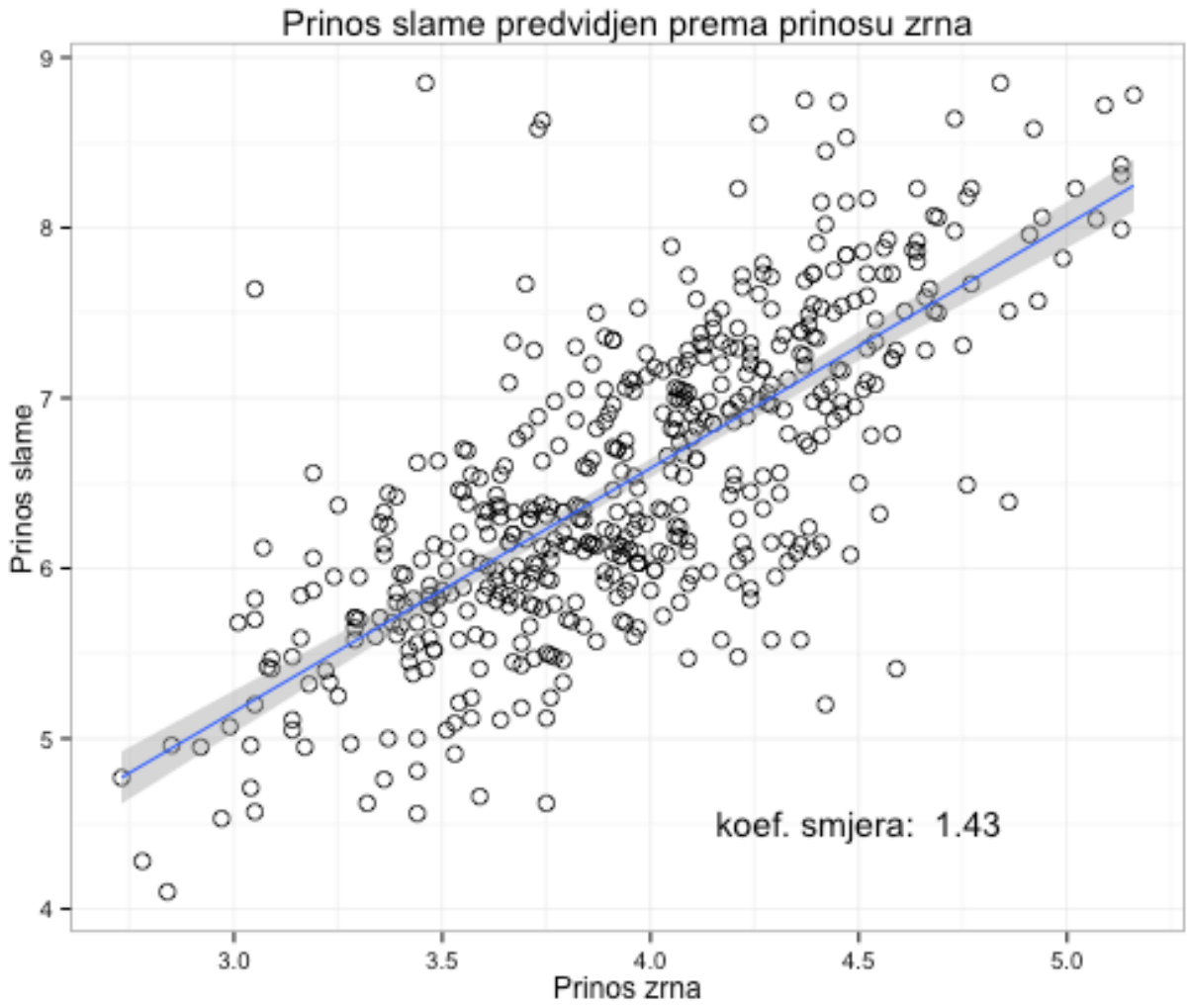


Figure 3: plot of chunk unnamed-chunk-8

```
re = residuals(model.straw.grain)
hist(re, main = "Reziduali iz linearnog modela slama ~ zrna ")
```

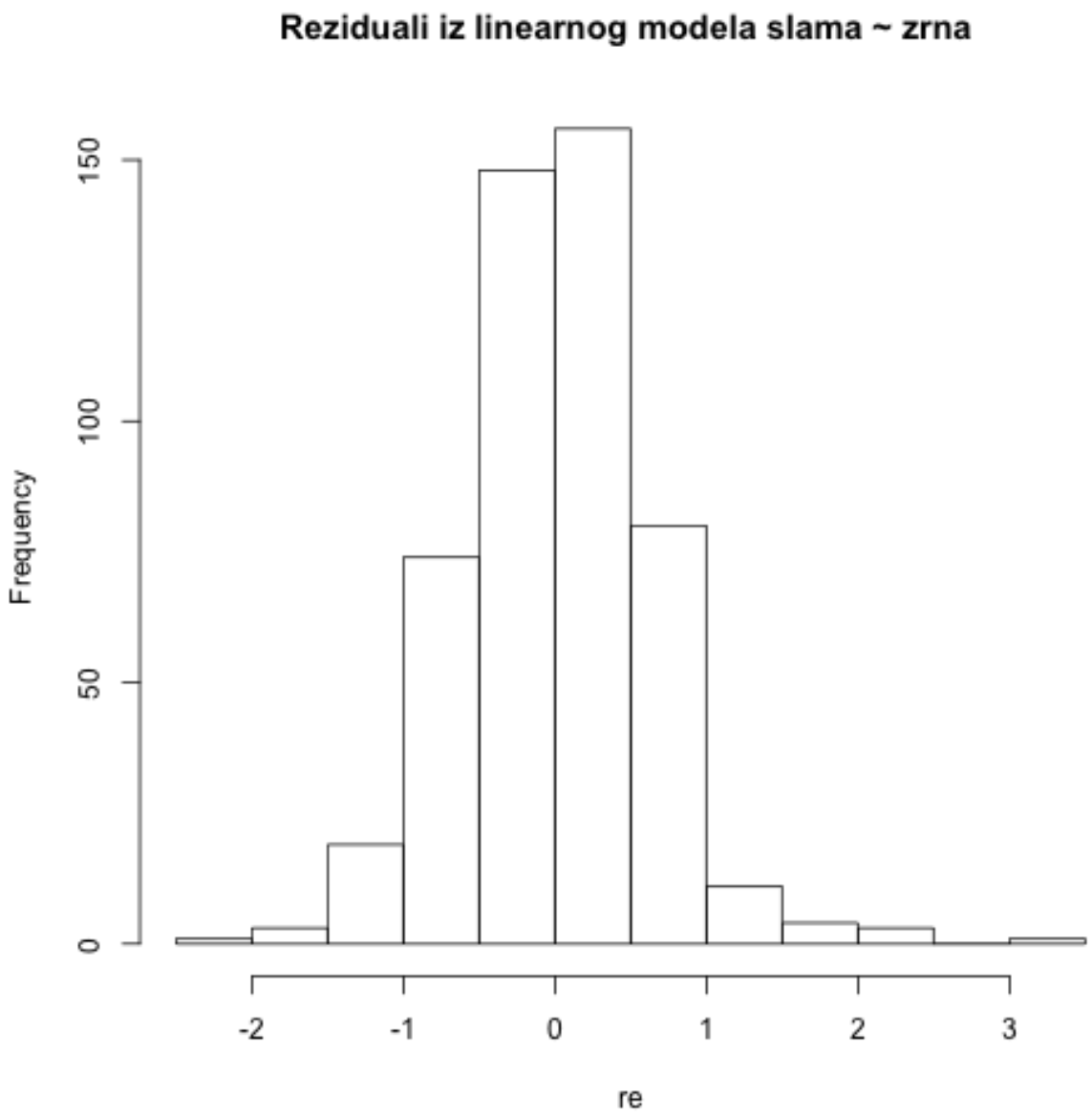


Figure 4: plot of chunk unnamed-chunk-9

Na prvi pogled čini se da reziduali nisu normalno distribuirani, jer je histogram previše strm. Provjerimo dojam na grafičkom prikazu kvantil - kvantil-a.

```
qqnorm(re, main = "Q-Q plot reziduala iz linearnog modela slama ~ zrna")
qqline(re)
```

```
summary(re)
```

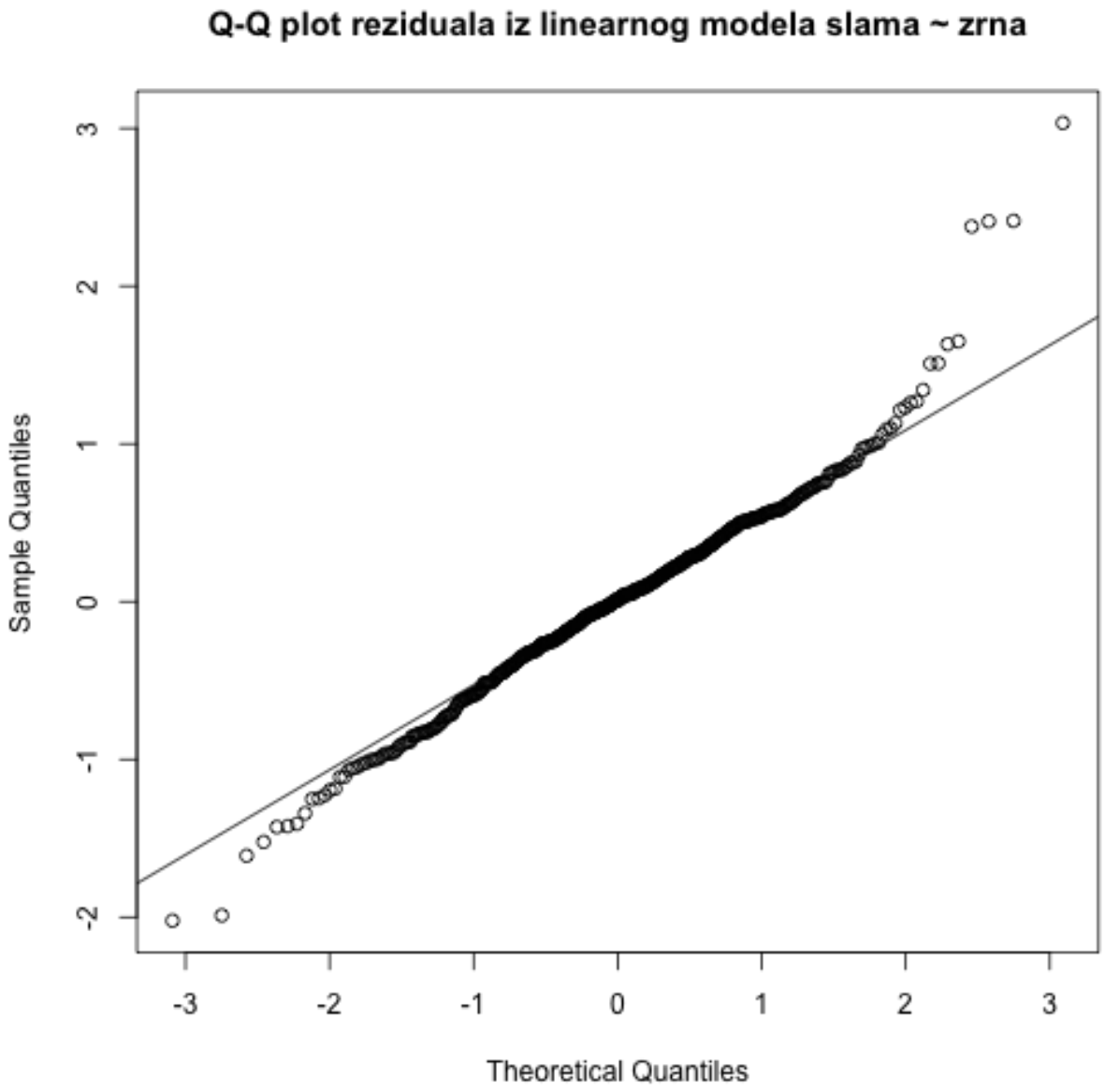


Figure 5: plot of chunk unnamed-chunk-10

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -2.0200 -0.3530  0.0104  0.0000  0.3730  3.0300
```

Iz slike se vidi da su najmanji reziduali premali a najveći preveliki da bi bili distribuirani po normalnoj razdiobi. Isto tako, iz sumarnog se prikaza vidi da je rang veći od 5 što je dosta velik broj.

Zadatak Koristeći test Shapiro-Wilks testirajte normalnost reziduala.

```
shapiro.test(re)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  re
## W = 0.9767, p-value = 3.63e-07
```

"p-value" je manja od 0.05. Na temelju histograma, Q-Q plota i vrijednosti od "p-value" hipotezu da su reziduali distribuirani po normalnoj razdiobi treba odbaciti.

Zadatak Nacrtaj Q-Q plot reziduala iz linearnog modela slama~zrno pomoću biblioteke *gglib2*.

Na adresi http://stackoverflow.com/questions/4357031/qqnorm-and-qqline-in-ggplot2 piše

> The ggplot package doesn't seem to contain code for calculating the parameters of the qqline, so I don't know if it's possible to achieve such a plot in a (comprehensible) one-liner.

Znači, morati ćemo se koristiti posebnom funkcijom. Na istoj adresi je navedena funkcija:

```
qqplot.data <- function (vec) # argument: vector of numbers
{
  # following four lines from base R's qqline()
  y <- quantile(vec[!is.na(vec)], c(0.25, 0.75))
  x <- qnorm(c(0.25, 0.75))
  slope <- diff(y)/diff(x)
  int <- y[1L] - slope * x[1L]

  d <- data.frame(resids = vec)

  ggplot(d, aes(sample = resids)) + stat_qq() + geom_abline(slope = slope, intercept = int)

}
```

Slika bez `qqline()` ne treba posebnu funkciju:

```
redf=data.frame(re)
ggplot(redf, aes(sample=re)) + stat_qq()
```

ali sliku sa `qqline()` trebamo nacrtati sa gore definiranom funkcijom.

```
qqplot.data(re)
```

Zadatak Usporedite modelske vrijednosti (vrijednosti dobivene pomoću pravca regresije) i mjerene vrijednosti slame. Bojom istaknite one točke koje su (vertikalno) od pravca regresije (modelirane vrijednosti) udaljene za više od jedan.

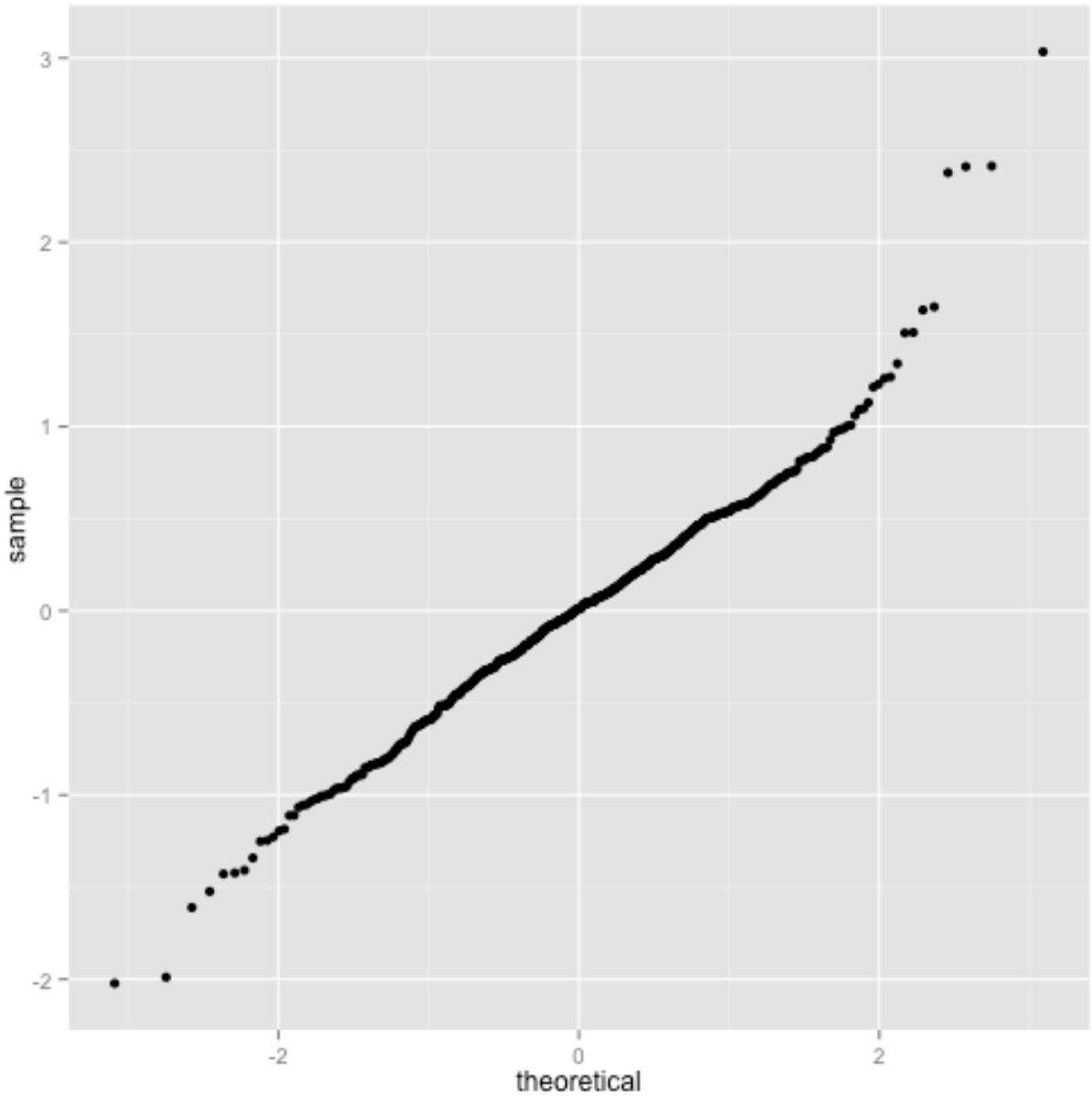


Figure 6: plot of chunk unnamed-chunk-13

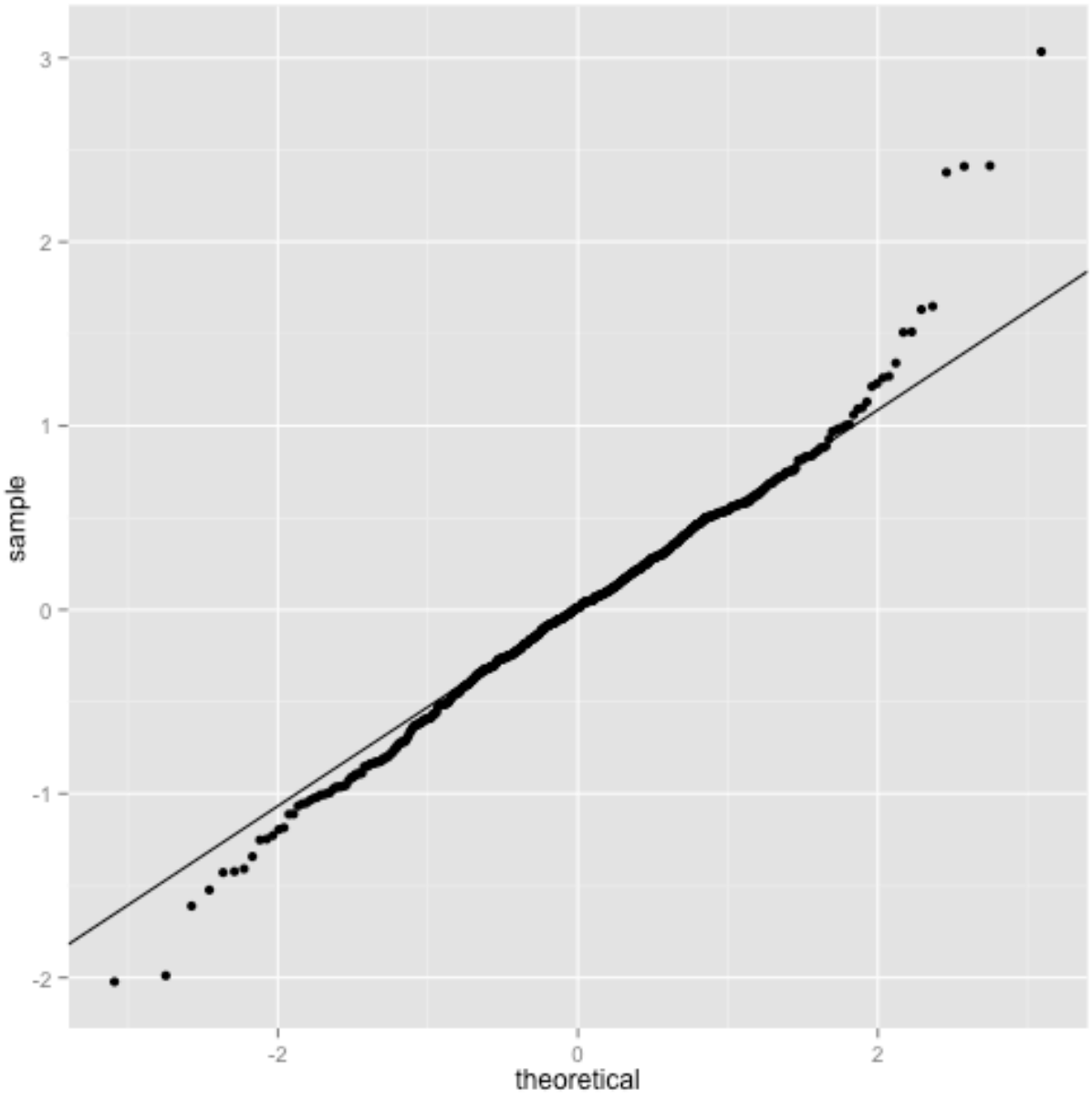


Figure 7: plot of chunk unnamed-chunk-14

```
fmsg = fitted(model.straw.grain)
# lim = c(min(fmsg, mhw$straw), max(fmsg, mhw$straw))
df.fmsg = data.frame(fmsg, mstraw=mhw$straw)
ggplot(df.fmsg, aes(fmsg, mstraw)) +
  geom_point(color = ifelse( (abs(fmsg - mhw$straw)) < 1 , "gray" ,
                      ifelse( (fmsg < mhw$straw), "blue", "red")),
             size = ifelse(abs(fmsg - mhw$straw) < 1, 3, 4)) +
  geom_abline(intercept = 0, slope = 1) +
  labs(title = "Mjerene vrijednosti koje su (vertikalno) za vise od 1 udaljene od pravca regresije",
         x = "Modelske vrijednosti (vrijednosti sa pravca regresije)",
         y = "Slama")
```

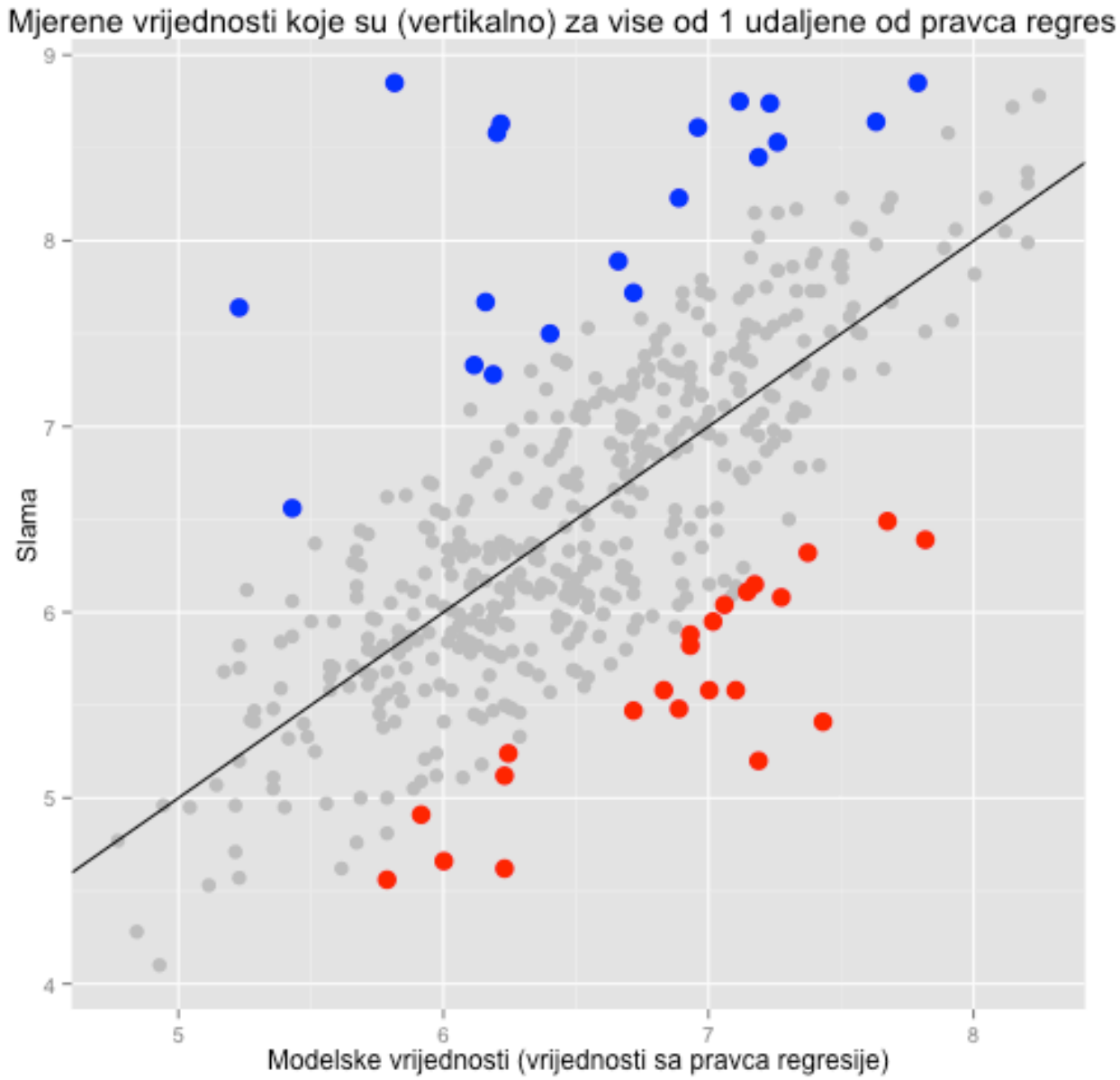


Figure 8: plot of chunk unnamed-chunk-15

Zadatak Neki prinosi slame nisu dobro aproksimirani pomoću linearnog modela. Identificirajte one prinose slame kojima je apsolutna vrijednost reziduala veća od 1.8. Prikažite te retke iz `mhw` sortirane po veličini omjera zrna/slama (grain/straw).

```
re = residuals(model.straw.grain)
which.high.res = which(abs(re) > 1.8)
sort(re[which.high.res])
```

```
##    184     35    295    337    311     15
## -2.022 -1.989  2.378  2.411  2.414  3.034
```

Izdvojimo iz tablice `mhw` retke određene sa `which.high.res` pa toj tablici dodajmo stupac `gsr` omjera `grain/straw`.

```
high.res = mhw[which.high.res, ]
high.res$gsr = high.res$grain/high.res$straw
high.res
```

```
##      r  c grain straw    gsr
## 15  15  1  3.46  8.85 0.3910
## 35  15  2  4.42  5.20 0.8500
## 184  4 10  4.59  5.41 0.8484
## 295 15 15  3.73  8.58 0.4347
## 311 11 16  3.74  8.63 0.4334
## 337 17 17  3.05  7.64 0.3992
```

Prikažimo retke (rekorde) iz data framea `high.res` sortirane po omjeru `grain/straw`.

```
high.res[order(high.res$gsr), ]
```

```
##      r  c grain straw    gsr
## 15  15  1  3.46  8.85 0.3910
## 337 17 17  3.05  7.64 0.3992
## 311 11 16  3.74  8.63 0.4334
## 295 15 15  3.73  8.58 0.4347
## 184  4 10  4.59  5.41 0.8484
## 35  15  2  4.42  5.20 0.8500
```

Daljnja dijagonistika linearne regresije

Ako je model linearne regresije odgovara podacima tada ne bi smjelo biti nikakve relacije (veze) između reziduala i modelskih vrijednosti (vrijednosti na pravcu linearne regresije).

Funkcija `plot` omogućava grafički prikaz, u kojem se prikazuju mjerene točke i krivulja. Ako nema nikakve relacije između reziduala i vrijednosti na pravcu regresije tada bi nacrtana krivulja trebala biti horizontali pravac i trebao bi prolaziti kroz ishodište (tj. trebao bi ležati na x-osi).

Točke na ovakvoj krivulji dobijemo pomoću funkcije `lowess()`.

```
plot(re ~ fmsg)
title("Reziduali vs. modelirane vrijednosti")
tocke = lowess(fmsg, re)
lines(tocke$x, tocke$y, type="l", col="red")
```

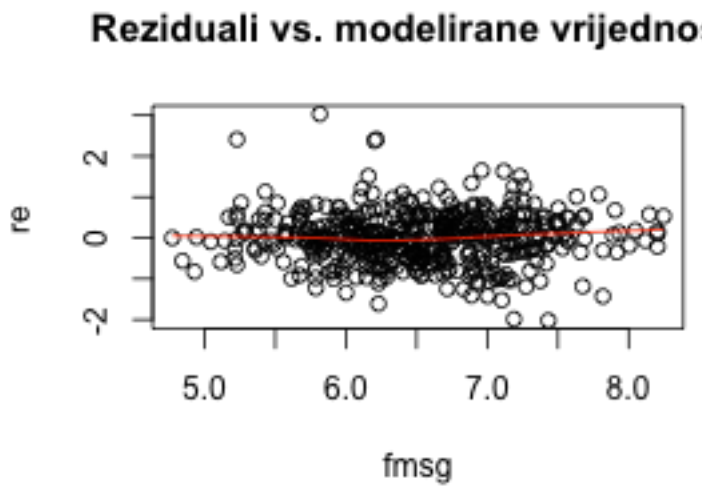


Figure 9: plot of chunk unnamed-chunk-19

Ovakvu sliku možemo dobiti jednostavnije opcijom `which = 1` funkcije `plot()`.

```
plot(model.straw.grain, which = 1)
```

Kako dobivena krivulja nije pravac, to ne možemo, u potpunosti, biti zadovoljni gornjom linearnom regresijom.

Nacrtajmo "istu" sliku pomoću biblioteke `ggplot2`.

```
privdf = data.frame(x=fmsg, y=re)
lowessdf = data.frame(lowess(fmsg, re))
ggplot(data = privdf, aes(x, y)) +
  geom_point(size = 2) +
  geom_line(data=lowessdf, aes(x, y), color = "red") +
  labs( title = "Krivulja dobivena lokalnom tezinskom lin. regresijom",
        x = "Modelirane vrijednosti", y = "Reziduali") +
  theme_bw()
```

Malo o funkciji `lowess()`.

Upotreba

```
lowess(x, y, f=2/3, iter=3, delta=.01*diff(range(x)))
```

Kako rezultat funkcije ovisi o parametru `f`?

- f: the smoother span. This gives the proportion of points in the plot which influence the smooth at each value. Larger values give more smoothness.

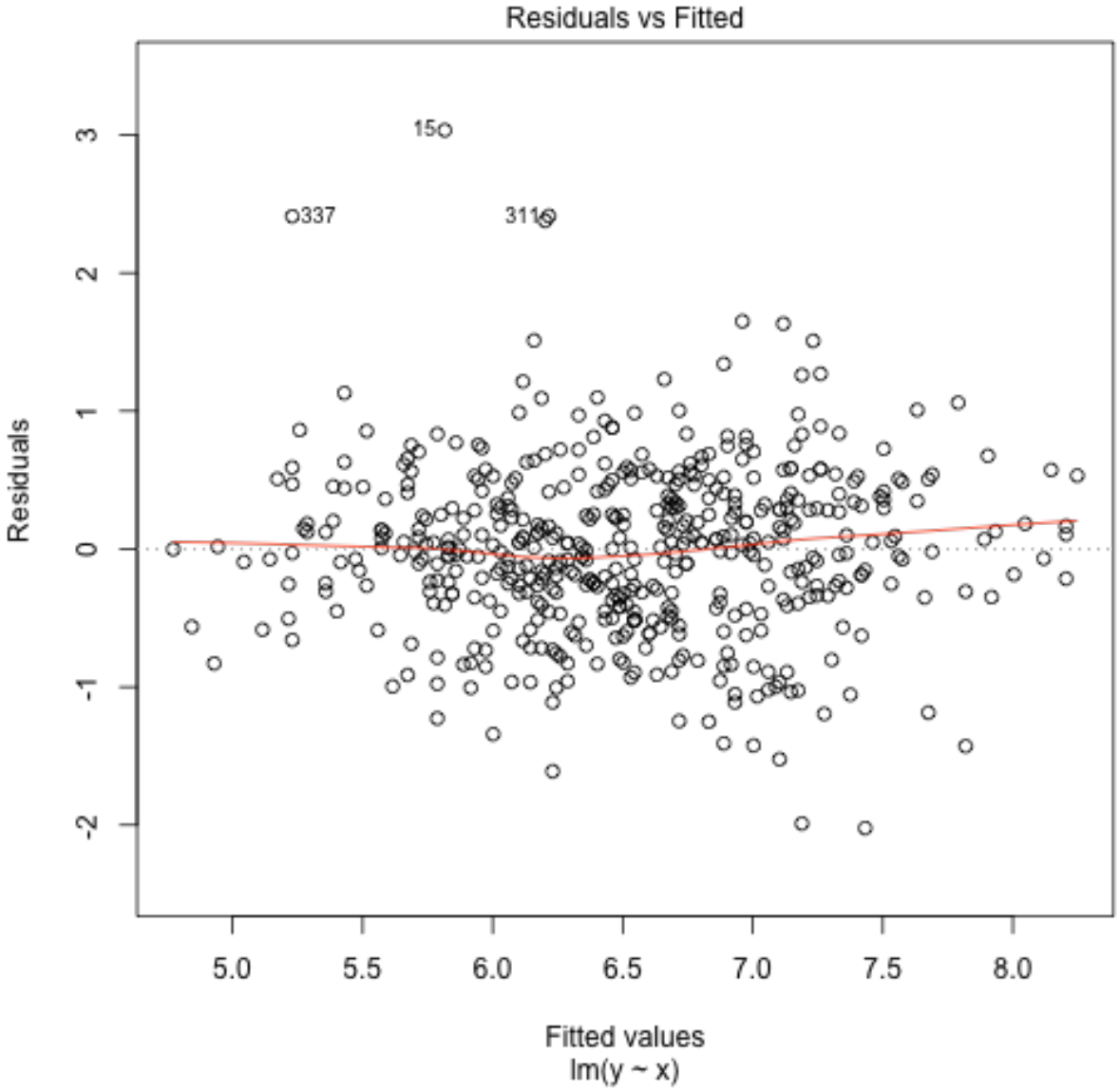


Figure 10: plot of chunk unnamed-chunk-20

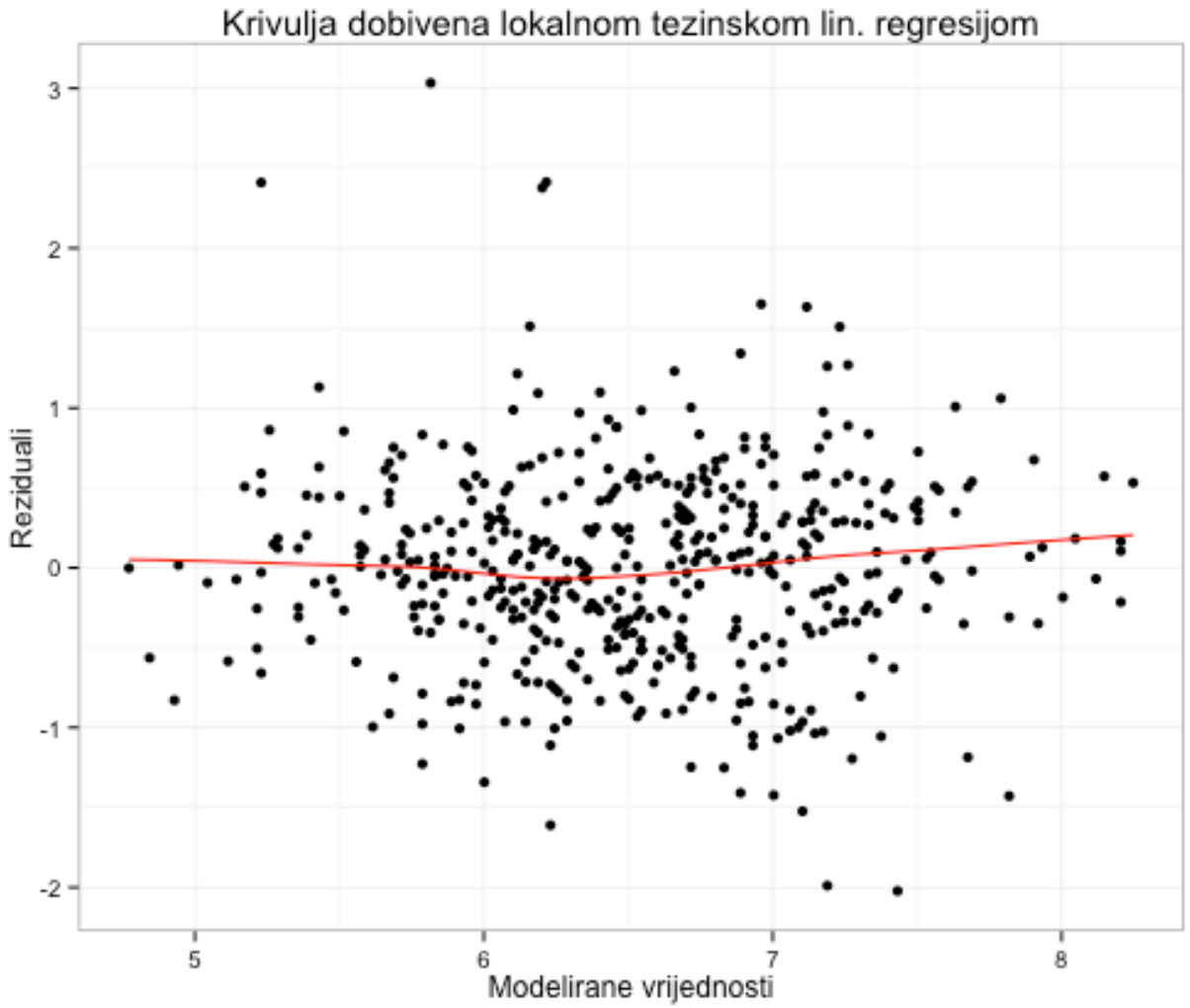


Figure 11: plot of chunk unnamed-chunk-21

```
data(cars)
plot(cars, main = "lowess(cars)")
lines(lowess(cars), col = 2)
lines(lowess(cars, f = 0.2), col = 3)
lines(lowess(cars, f = 0.3), col = 4)
legend(5, 120, c(paste("f = ", c("2/3", ".2", ".3"))), lty = 1, col = 2:4)
```

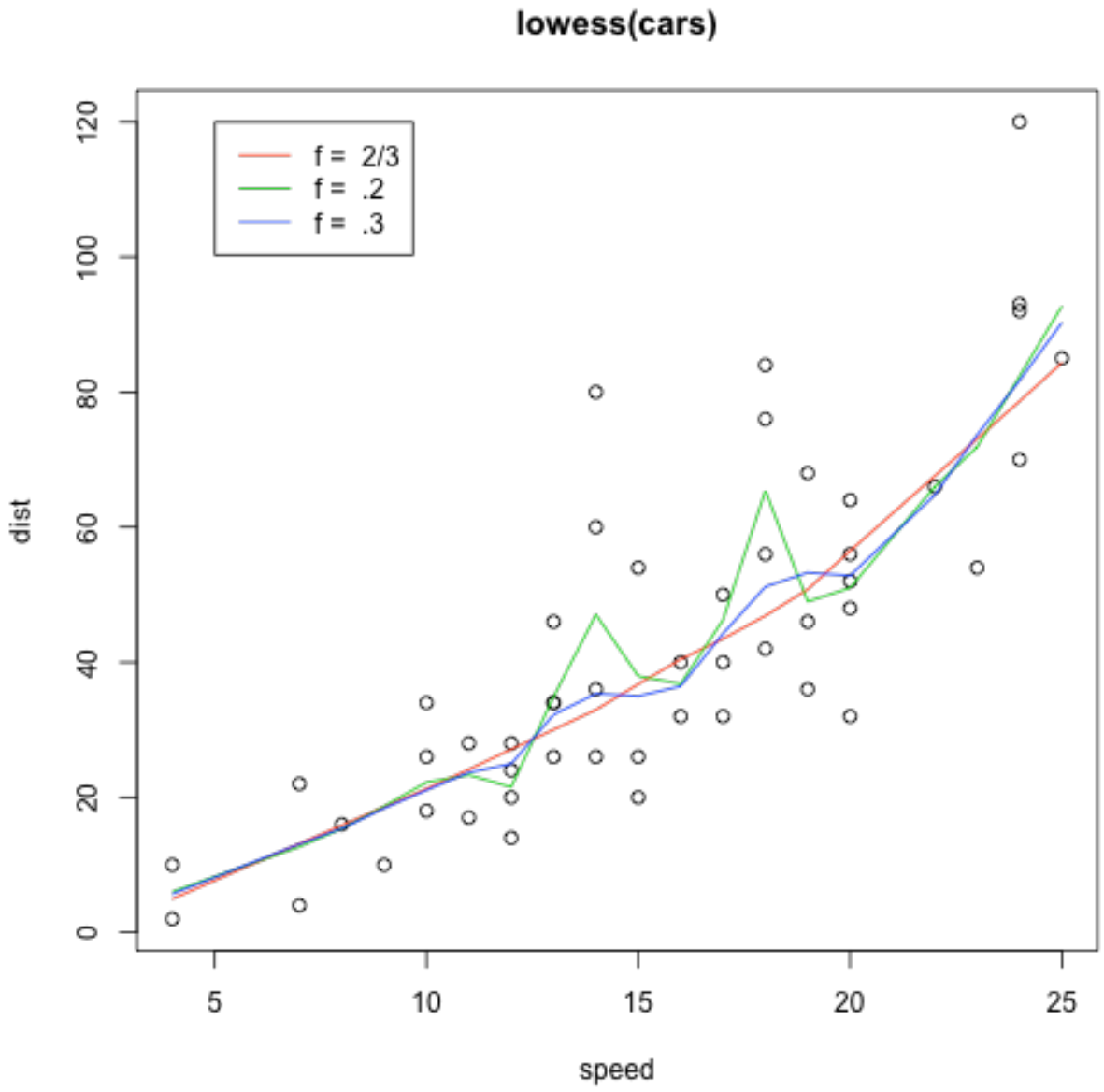


Figure 12: plot of chunk unnamed-chunk-22

Predviđanja vrijednosti

Ako je određen pravac regresije i želimo predvidjeti vrijednost na temelju prediktora moguće su dvije greške:

1. Nepouzdanost u određivanju pravca regresije
2. Nepouzdanost u određivanju vrijednosti iz prediktora.

Stoga je dobro odrediti intervale povjerenja za vrijednosti pravca regresije (confidence interval) i intervale povjerenja za predviđenu vrijednost (prediction interval).

Zadatak Izračunajte vrijednost i 95% confidence interval i prediction interval prinosa slame za srednju vrijednost μ prinosa zrna, te za $\mu \pm k \cdot \sigma$ gdje je $k = 1, 2$.

```
xpotez = mean(mhw$grain);  sdev = sd(mhw$grain)
# Pažnja, u p.frame varijabla se mora zvati x, jer je
# tako definiran model.straw.grain
p.frame = data.frame(x = seq(xpotez - 2*sdev, xpotez + 2*sdev,
                                  by = sdev))
predict(model.straw.grain, newdata = p.frame,
        interval="confidence", level=0.95)
```

```
##     fit   lwr   upr
## 1 5.204 5.083 5.325
## 2 5.859 5.783 5.936
## 3 6.515 6.461 6.569
## 4 7.170 7.094 7.247
## 5 7.826 7.705 7.947
```

```
predict(model.straw.grain, newdata = p.frame,
        interval="prediction", level=0.95)
```

```
##     fit   lwr   upr
## 1 5.204 3.990 6.418
## 2 5.859 4.649 7.070
## 3 6.515 5.306 7.724
## 4 7.170 5.960 8.381
## 5 7.826 6.612 9.040
```

Prediction interval je, prirodno, širi od confidence interval jer je zadnji ocjena procječne vrijednosti a prvi ocjena bilo kakave vrijednosti.

Zadatak Nacrtajte točke koje prikazuju slamu vs. žita, pravac regresije i oba intervala povjerenja za prinos žita od 2 do 6 sa korakom 0.1.

```
p.frame = data.frame(x = seq(from = 2, to = 6, by = 0.1))
pred.c = predict(model.straw.grain, newdata = p.frame,
                 interval = "confidence", level = 0.95)
pred.p = predict(model.straw.grain, newdata = p.frame,
                 interval = "prediction", level = 0.95)
plot (straw ~ grain, data = mhw, pch = 20)
title(main = "Prinos slame prema predvidjanju prinosa zita",
      sub = "Prediction (blue) i confidence (red) intervali")
abline(model.straw.grain)
grid()
lines(p.frame$x, pred.c[, "lwr"], col = "red", lwd = 1.5)
lines(p.frame$x, pred.c[, "upr"], col = "red", lwd = 1.5)
lines(p.frame$x, pred.p[, "lwr"], col = "blue", lwd = 1.5)
```

```
lines(p.frame$x, pred.p[, "upr"], col = "blue", lwd = 1.5)
points(mean(x), mean(y), pch=23, cex=2, bg="red")
abline(h = mean(y), lty = 2)
abline(v = mean(x), lty = 2)
```

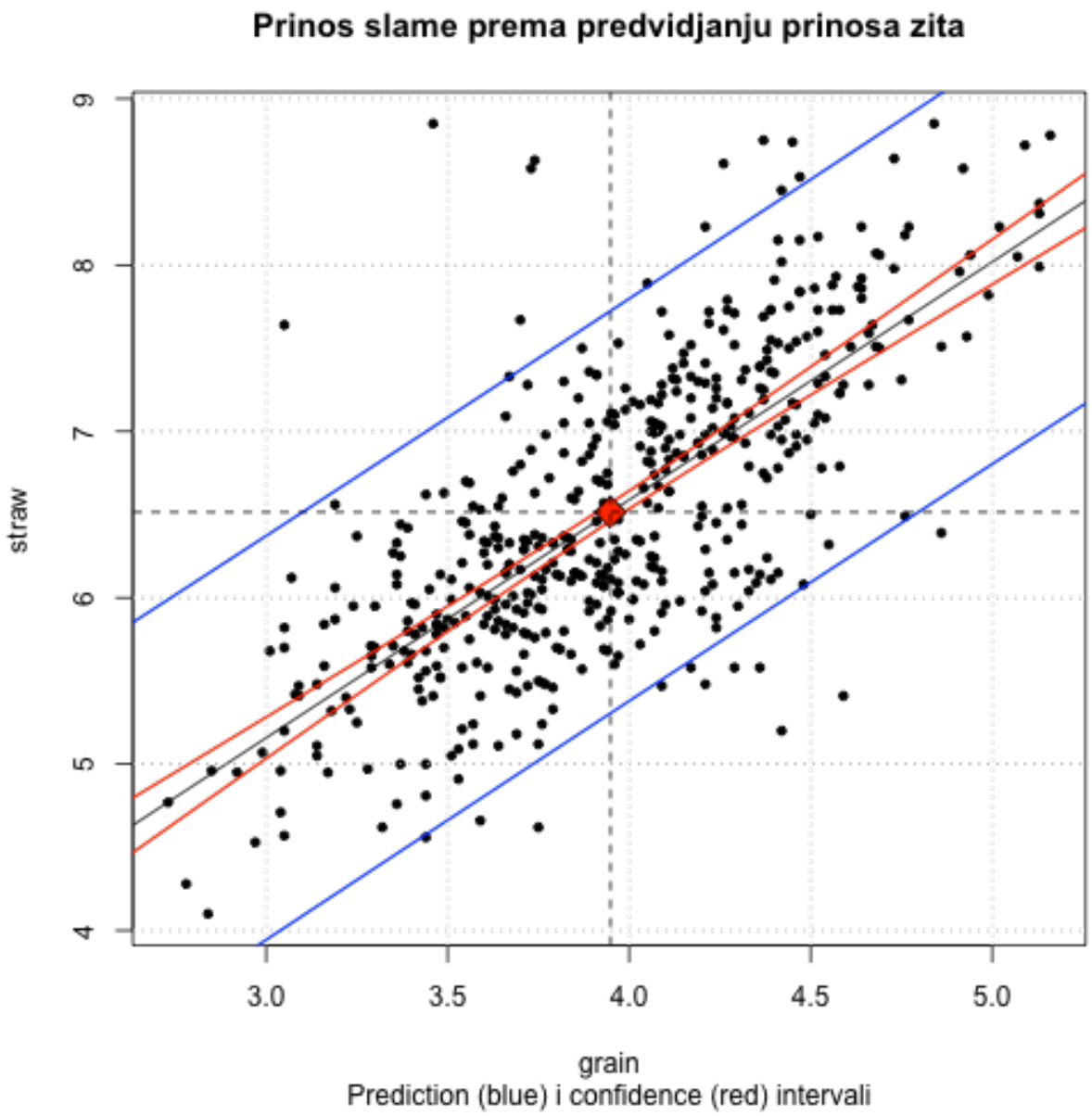


Figure 13: plot of chunk unnamed-chunk-25

Zadatak Usporedite regresiju slama vs. žito sa regresijom žito vs. slama.

```
model.grain.straw = lm(grain ~ straw, data = mhw)
summary(model.grain.straw)
```

```
##
```

```
## Call:
## lm(formula = grain ~ straw, data = mhw)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3580 -0.2011  0.0004  0.1918  1.0527
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.5231     0.1028    14.8   <2e-16 ***
## straw         0.3723     0.0156    23.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.314 on 498 degrees of freedom
## Multiple R-squared:  0.533,  Adjusted R-squared:  0.532
## F-statistic:  567 on 1 and 498 DF,  p-value: <2e-16
```

```
summary(model.straw.grain)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0223 -0.3529  0.0104  0.3734  3.0342
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8663     0.2387    3.63  0.00031 ***
## x             1.4305     0.0601   23.82  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.615 on 498 degrees of freedom
## Multiple R-squared:  0.533,  Adjusted R-squared:  0.532
## F-statistic:  567 on 1 and 498 DF,  p-value: <2e-16
```

U rezultatima se navodi "multiple R" (točnije njegov kvadrat). Značenje tog koeficijenta (bez kvadrata) je koeficijent korelacije između predviđene vrijednosti(na pravcu regresije) i izmjerene vrijednosti.

Odredite koeficijent korelacije R za obje regresije

```
sqrt(summary(model.grain.straw)$r.squared)
```

```
## [1] 0.7298
```

```
sqrt(summary(model.straw.grain)$r.squared)
```

```
## [1] 0.7298
```

Dobiveni rezultati su isti. Koja je posljedica te činjenice?

Neka Y ovisi o X (tj. model je Y ~ X); slučajna vrijednost Y se sastoji od vrijednosti pravca regresije (a vrijednost na pravcu regresije je određena sa X) i slučajnog odstupanja od pravca regresije. Statistička ovisnost odstupanje od pravca regresije je opisano koeficijentom korelacije R.

Kako su u oba modela koeficijenti isti to slijedi da poznavanje slame u prvom modelu (žito ~ slama) jednako određuje žito kao što poznavanje žita u drugom modelu (slama ~ žito) određuje slamu.

Zadatak Može se činiti da je pravac regresije u modelu (žito ~ slama) inverzna funkcija pravca regresije u modelu (slama ~ žito). Provjerite da li je to istina?

Ako je zadan pravac $y = a \cdot x + b$, njemu inverzan se dobije ako se po y riješi izraz $x = a \cdot y + b$, tj. dobije se

$$y = \frac{1}{a}x - \frac{b}{a}$$

Pravac regresije u modelu (slama ~ žito) je $y = 1.431 \cdot x + 0.866$. Inverzni pravac je

$$y = \frac{1}{1.431}x - \frac{0.866}{1.431}$$
$$y = 0.699 \cdot x - 0.605$$

Budući je jednadžba pravca regresije u modelu (žito ~ slama) jednaka $y = 0.3723 \cdot x + 1.523$ to inverzni pravac prvog modela nema niti jedan koeficijent zajednički sa pravcem drugog modela.

Zaljučujemo da regresija između dvije varijable nije simetrična već rezultat ovisi o tome koja je varijabla input a koja output (koja je nezavisna a koja zavisna).