STA 4320 CHAP 3.3.3

Prof. He Jiang

Oct 2024

Sec 3.3.3

Advertising dataset

```
fpath = getwd()
Advertising = read.csv(paste0(fpath, "/Advertising.csv"))
```

Credit and Auto dataset

```
require(ISLR2)
```

```
## Loading required package: ISLR2
```

```
require(car) # residual vs x plot
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
require(MASS) # mammals dataset
```

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

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:ISLR2':
##
##     Boston
```

Non Linearity of data

Auto dataset, mpg as Y, horsepower as X

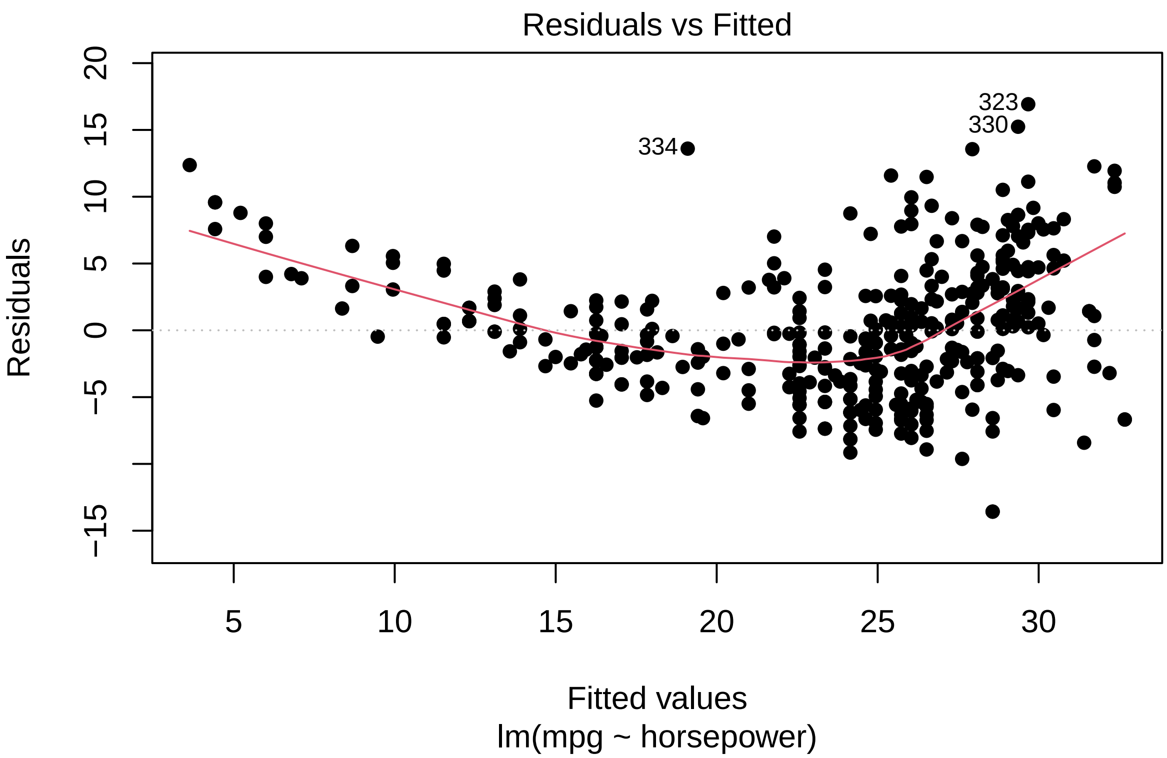
First, we do simple linear regression

```
reg_1 = lm(mpg ~ horsepower, data = Auto)
# summary(reg_1)
```

Residual vs fitted value plot

Note that the curve is a "smooth fit"(summary) to the dots, intended to give a trend.

```
plot(reg_1, which = 1, pch = 16)
```

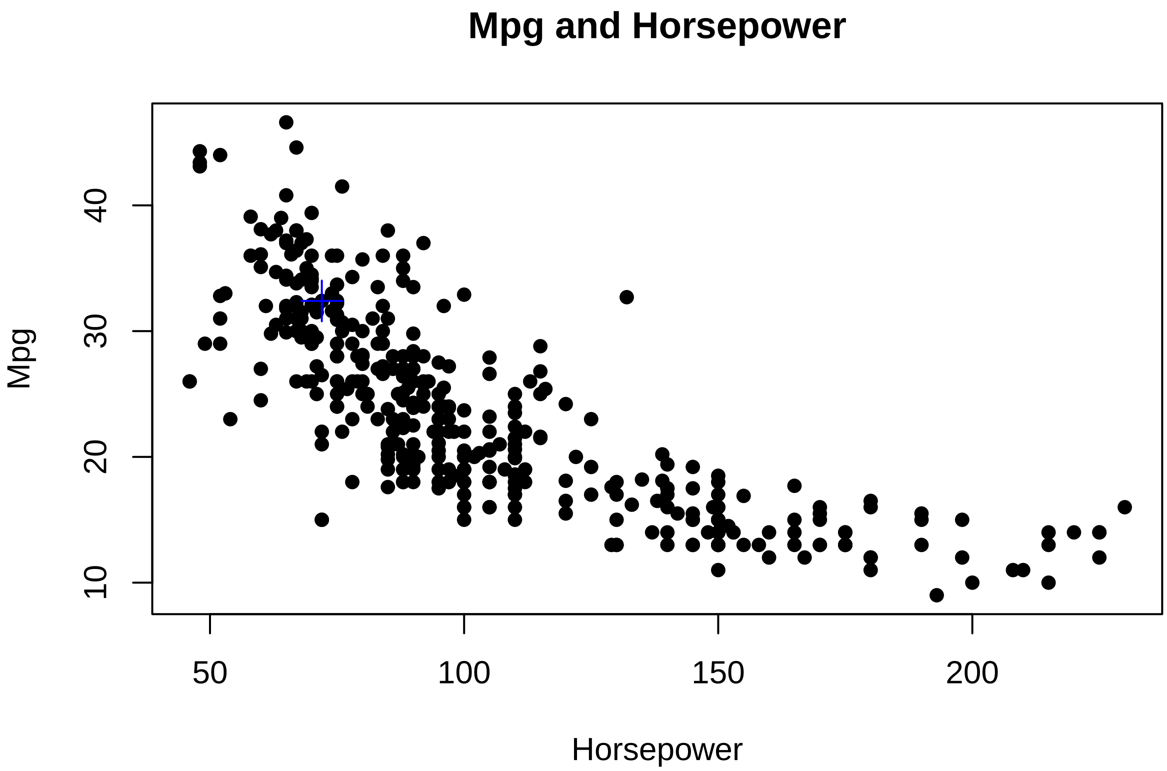


Question: in car 334, what should the sum of the x and y coordinates be?

Scatterplot

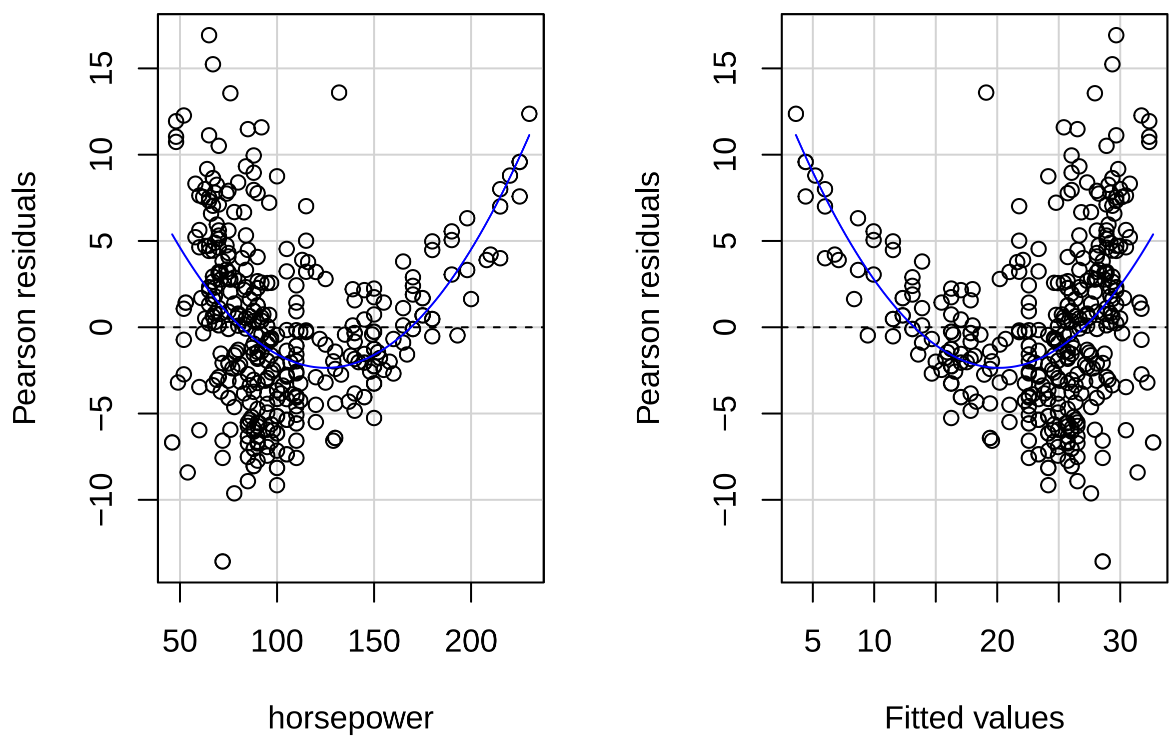
```r
y = Auto$mpg
x = Auto$horsepower

plot(x, y,
     main = "Mpg and Horsepower",
     pch = 16,
     xlab = "Horsepower",
     ylab = "Mpg")
points(x[334], y[334], col = "blue", pch = 3, cex = 2)
```



See residual vs x in simple linear regression (in car package)

```
residualPlots(reg_1)
```




```
##                Test stat Pr(>|Test stat|)
## horsepower         10.08        < 2.2e-16 ***
## Tukey test         10.08        < 2.2e-16 ***
```

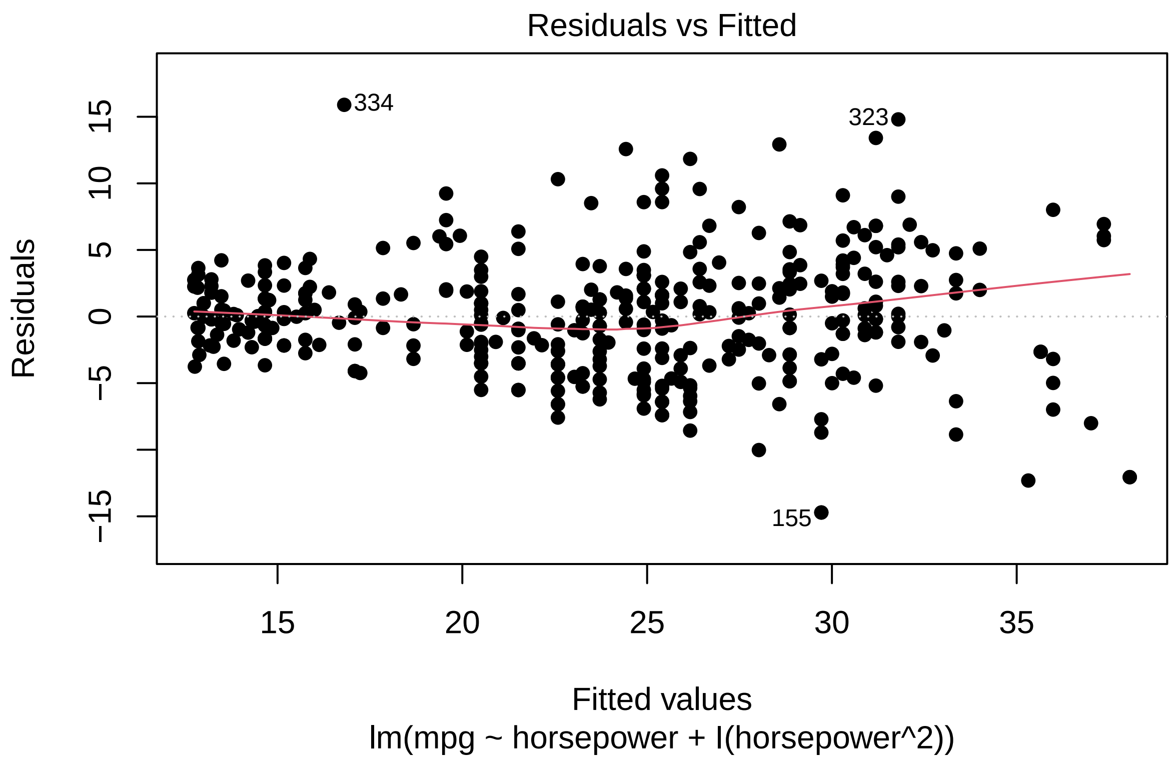
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next, we consider polynomial regression with degree 2

```r
reg_2 = lm(mpg ~ horsepower + I(horsepower^2), data = Auto)
# summary(reg_2)
```

Residual vs fitted value plot

```r
plot(reg_2, which = 1, pch = 16)
```



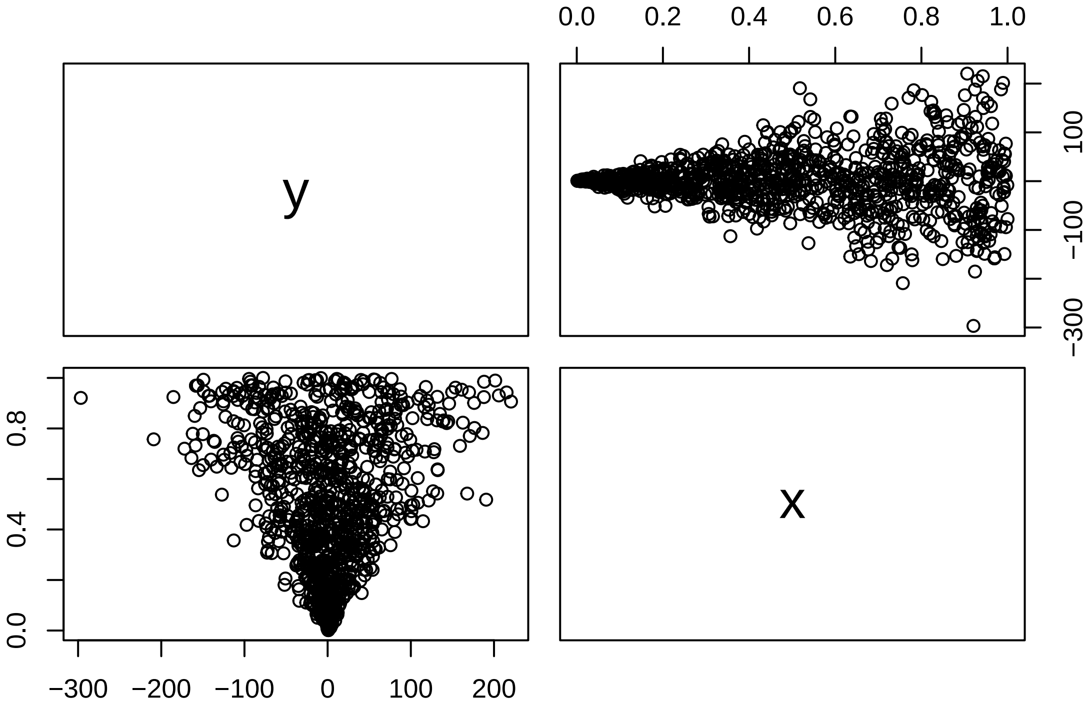
Non Constant Variance

Example with error terms being generally larger for larger fitted values

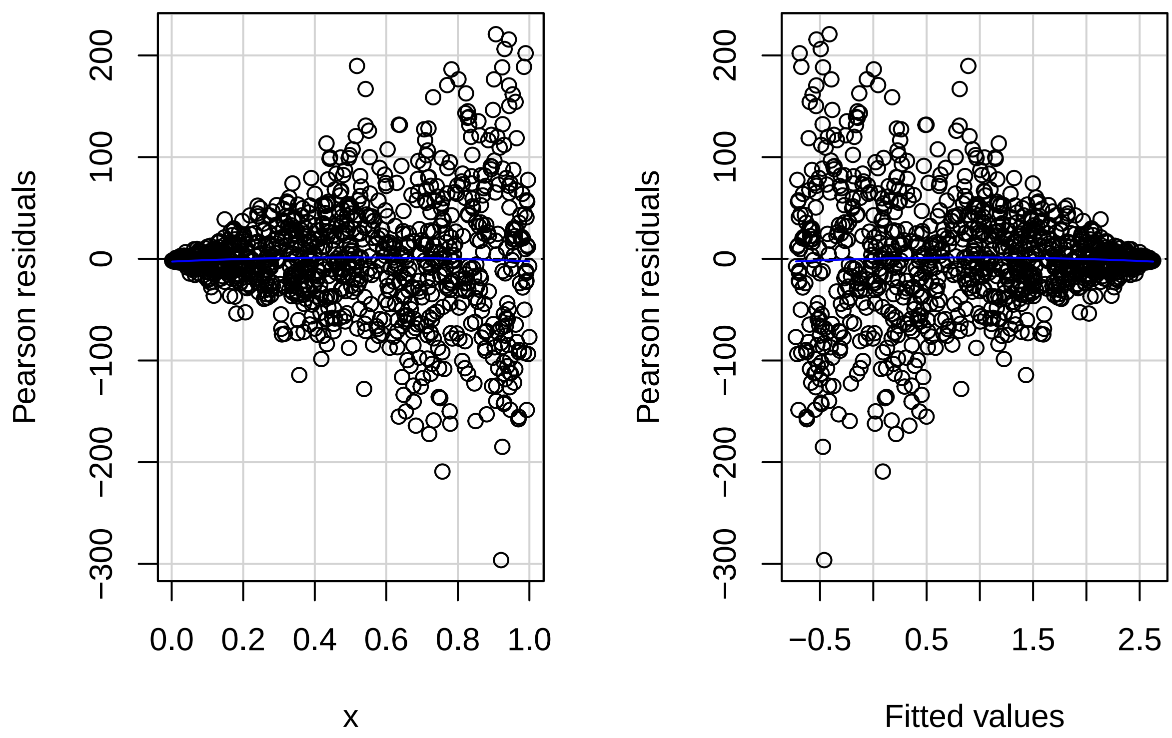
```r
n = 1000

set.seed(1)
x = runif(n, 0, 1)
epsilon = 100 * (x) * rnorm(n, 0, 1)
y = 1 + 2 * x + epsilon
reg = lm(y ~ x)

# pairwise plots will show funnel shape
pairs(y ~ x)
```




```r
# residual plots will show funnel shape
residualPlots(reg)
```



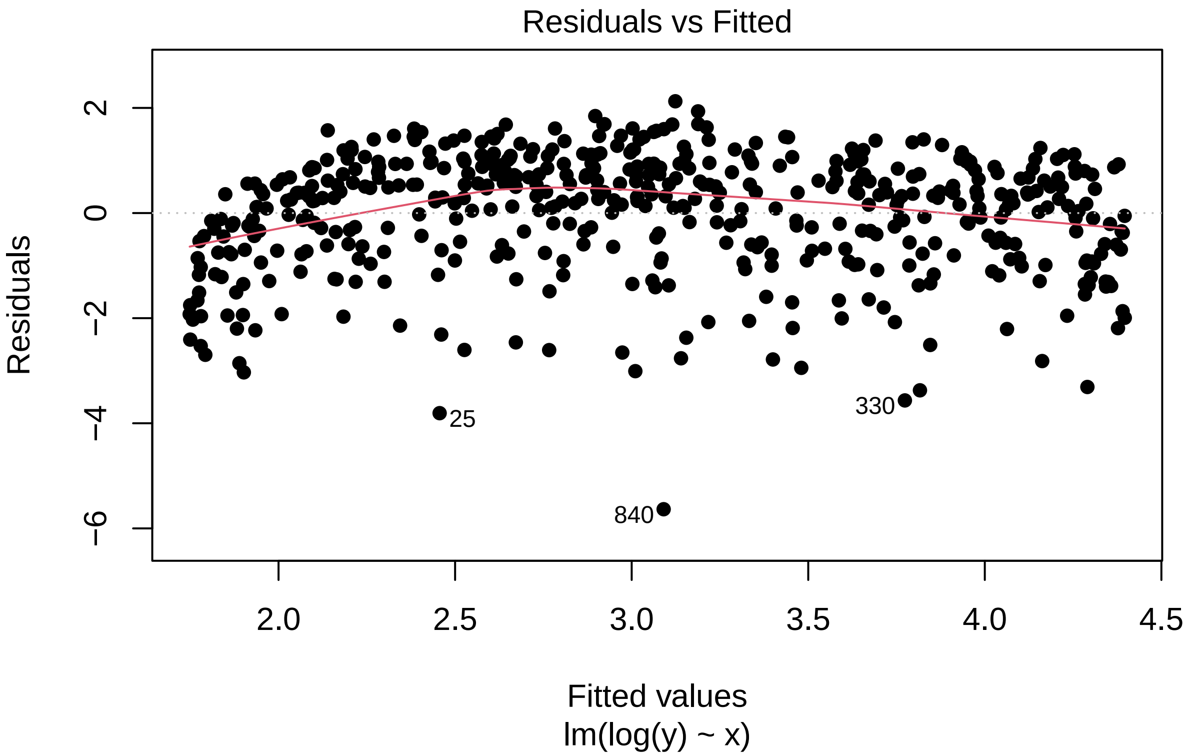

```
##            Test stat Pr(>|Test stat|)
## x            -0.6361           0.5248
## Tukey test   -0.6361           0.5247
```

Taking the log term

```r
reg_log = lm(log(y) ~ x)
```

```
## Warning in log(y): NaNs produced
```

```r
# taking the log improves the residuals
plot(reg_log, which = 1, pch = 16)
```



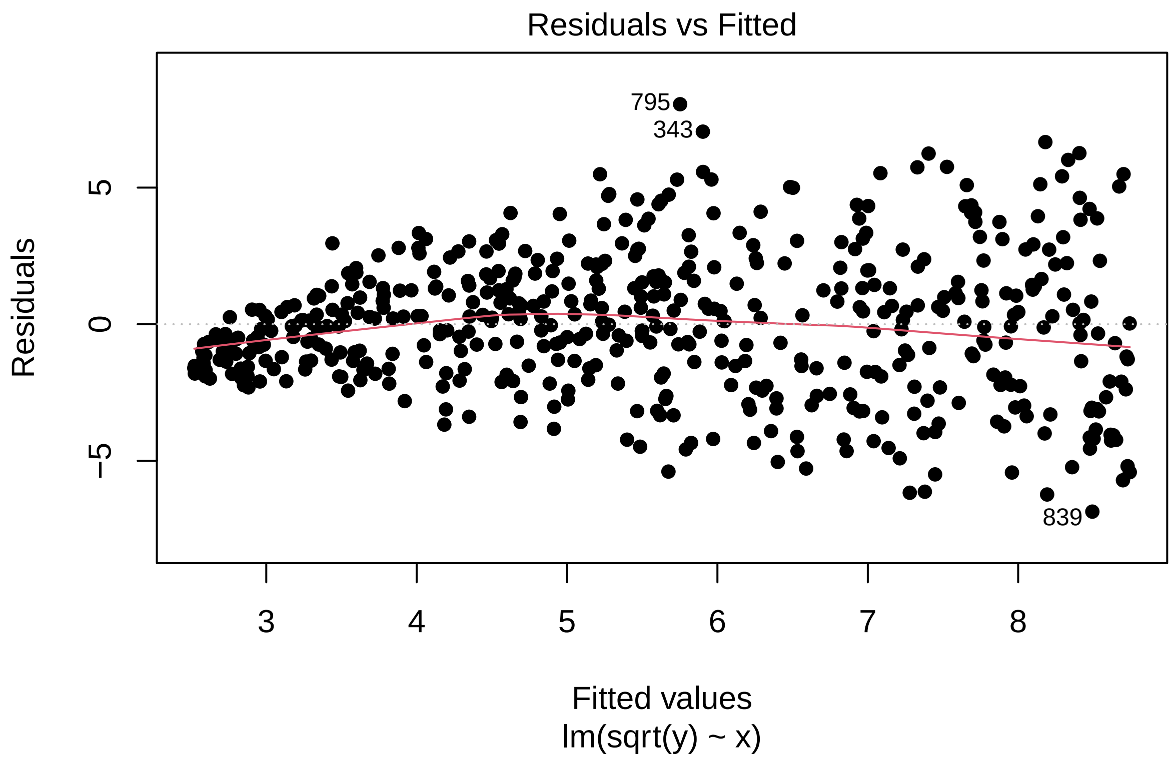

```r
# for relationship with x also use the following
# residualPlots(reg_log)
```

Taking the square root term

```r
reg_sqrt = lm(sqrt(y) ~ x)
```

```
## Warning in sqrt(y): NaNs produced
```

```r
# taking the sqrt does not show significant improvements
plot(reg_sqrt, which = 1, pch = 16)
```



Weighted least squares

```
reg = lm(y ~ x)
reg_weighted = lm(y ~ x, weights = x^(-2))

coef(reg)
```

```
## (Intercept)           x
##    2.630844   -3.357237
```

```
coef(reg_weighted)
```

```
## (Intercept)           x
##   0.8246334   0.7737004
```

Outliers

Outlier simulation

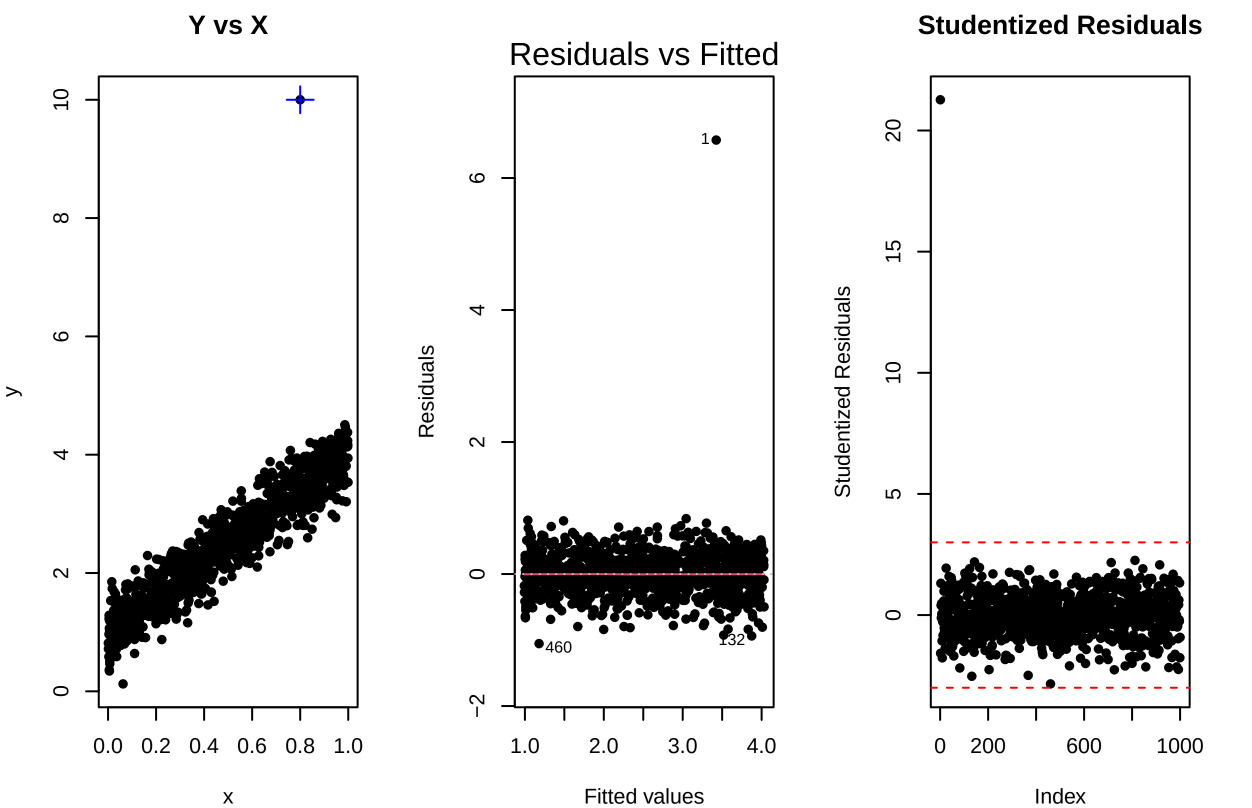
```
n = 1000
x = runif(n)
y = 1 + 3*x + 0.3*rnorm(n)
x[1] = 0.8; y[1] = 10 # outlier in predictor
reg = lm(y ~ x)

par(mfrow = c(1, 3))

# plot visualizing the outlier
plot(x, y, pch=16,
     main = "Y vs X")
points(x[1], y[1], col = "blue", pch = 3, cex = 2)
```

```r
# residuals
plot(reg, which = 1, pch = 16)

# visualizing the studentized residuals
plot((rstudent(reg)),
     main = "Studentized Residuals", pch = 16,
     ylab = "Studentized Residuals")
abline(h = -3, col = "red", lty = 2)
abline(h = 3, col = "red", lty = 2)
```




```r
# as.numeric( scale(residuals(reg)) ) could also give the studentized residuals
```

Mammals dataset (MASS)

```r
head(mammals)
```

```
##                     body brain
## Arctic fox         3.385  44.5
## Owl monkey         0.480  15.5
## Mountain beaver    1.350   8.1
## Cow              465.000 423.0
## Grey wolf         36.330 119.5
## Goat              27.660 115.0
```

```r
y = mammals$brain
x = mammals$body

reg = lm(brain ~ body, data = mammals)
```

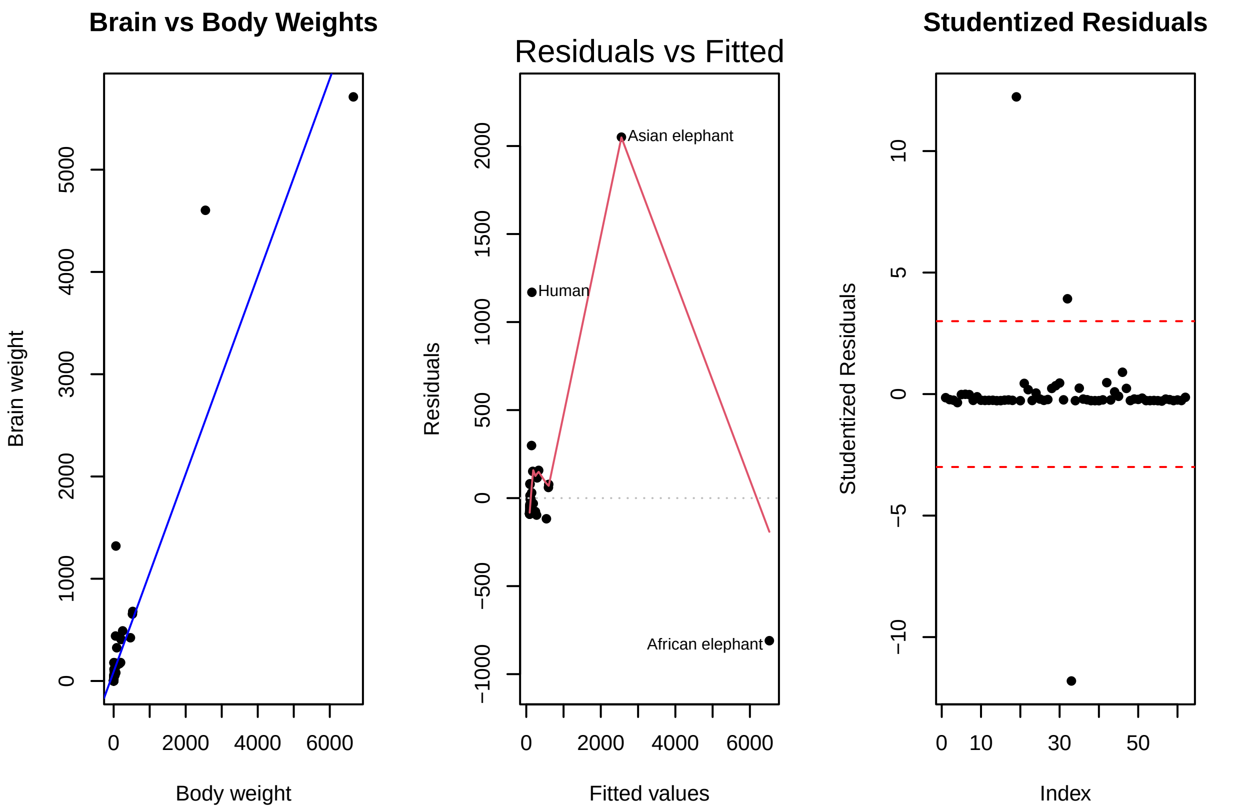
Mammals plots

```r
par(mfrow = c(1, 3))

# plot visualizing the outlier
plot(x, y, pch = 16, main = "Brain vs Body Weights",
     xlab = "Body weight", ylab = "Brain weight")
abline(reg, col = "blue")

# residuals
plot(reg, which = 1, pch = 16)

# visualizing the studentized residuals
plot((rstudent(reg)),
     main = "Studentized Residuals", pch = 16,
     ylab = "Studentized Residuals")
abline(h = -3, col = "red", lty = 2)
abline(h = 3, col = "red", lty = 2)
```



To find possible outliers

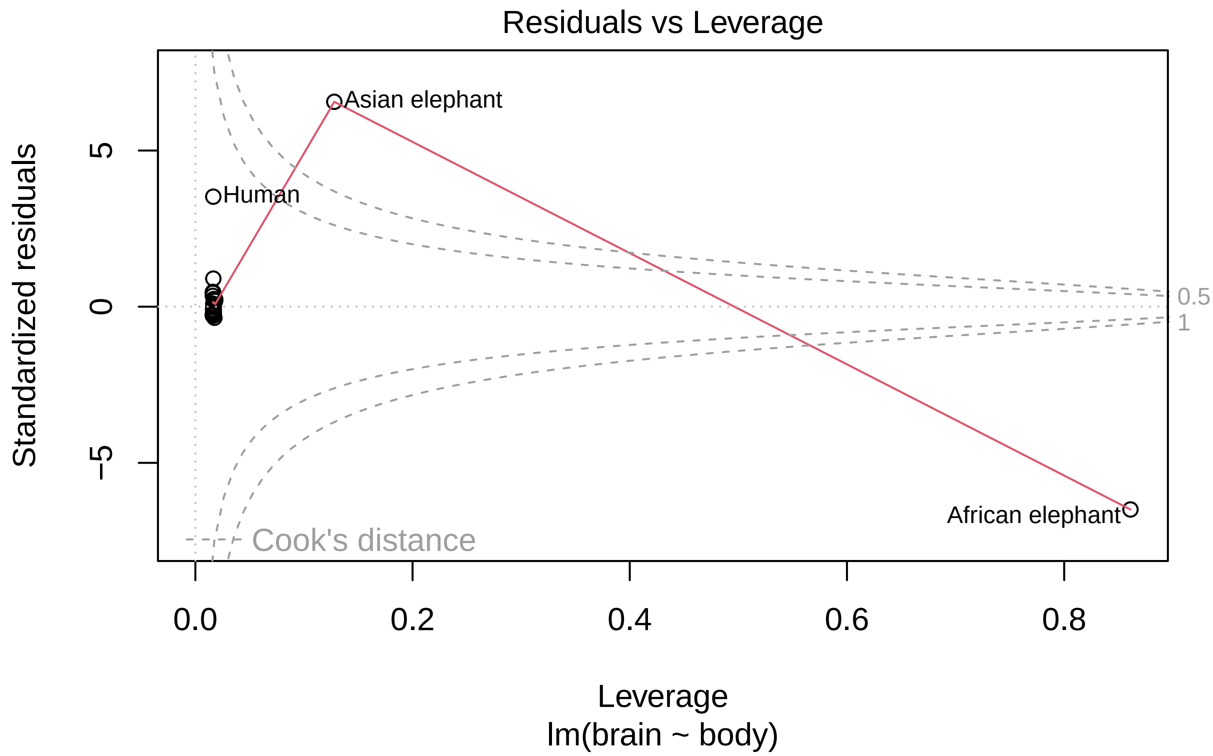
```r
which(abs( rstudent(reg) ) >= 3)
```

```
##    Asian elephant             Human African elephant
##                19                32               33
```

Leverage

Residual vs leverage plot

```r
plot(reg, 5)
```



Colinearity

Age, Rating, Limit in Credit dataset

```r
reg = lm(Balance ~ Age + Rating + Limit, data = Credit)
summary(reg)
```

```
##
## Call:
## lm(formula = Balance ~ Age + Rating + Limit, data = Credit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -729.67 -135.82   -8.58  127.29  827.65
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -259.51752   55.88219  -4.644 4.66e-06 ***
## Age           -2.34575    0.66861  -3.508 0.000503 ***
## Rating         2.31046    0.93953   2.459 0.014352 *
## Limit          0.01901    0.06296   0.302 0.762830
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 229.1 on 396 degrees of freedom
## Multiple R-squared:  0.7536, Adjusted R-squared:  0.7517
## F-statistic: 403.7 on 3 and 396 DF,  p-value: < 2.2e-16
```

Correlation Matrix (for pair-colinearity)

```r
round(cor(Credit[,c("Age", "Rating", "Limit")]), 4)
```

```
##           Age Rating  Limit
## Age    1.0000 0.1032 0.1009
## Rating 0.1032 1.0000 0.9969
## Limit  0.1009 0.9969 1.0000
```

Marginal effect of the Limit variable

```r
reg_age_limit = lm(Balance ~ Age + Limit, data = Credit)
summary(reg_age_limit)
```

```
##
## Call:
## lm(formula = Balance ~ Age + Limit, data = Credit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -696.84 -150.78  -13.01  126.68  755.56
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.734e+02  4.383e+01  -3.957 9.01e-05 ***
## Age         -2.291e+00  6.725e-01  -3.407 0.000723 ***
## Limit        1.734e-01  5.026e-03  34.496  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 230.5 on 397 degrees of freedom
## Multiple R-squared:  0.7498, Adjusted R-squared:  0.7486
## F-statistic:   595 on 2 and 397 DF,  p-value: < 2.2e-16
```

```r
reg_rating_limit = lm(Balance ~ Rating + Limit, data = Credit)
summary(reg_rating_limit)
```

```
##
## Call:
## lm(formula = Balance ~ Rating + Limit, data = Credit)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -707.8 -135.9   -9.5  124.0  817.6
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -377.53680   45.25418  -8.343 1.21e-15 ***
## Rating         2.20167    0.95229   2.312   0.0213 *
## Limit          0.02451    0.06383   0.384   0.7012
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 232.3 on 397 degrees of freedom
## Multiple R-squared:  0.7459, Adjusted R-squared:  0.7447
## F-statistic: 582.8 on 2 and 397 DF,  p-value: < 2.2e-16
```

VIF (for general colinearity)

```
vif(reg)
```

```
##        Age    Rating      Limit
##   1.011385 160.668301 160.592880
```

```
# check one value, with Rating as y, and Age and Limit as Xs
r_sq_vif = summary( lm(Rating ~ Age + Limit, data = Credit) )$r.sq
vif_rating = 1 / (1 - r_sq_vif)
vif_rating
```

```
## [1] 160.6683
```

Balance as Y and age and limit as Xs (without rating)

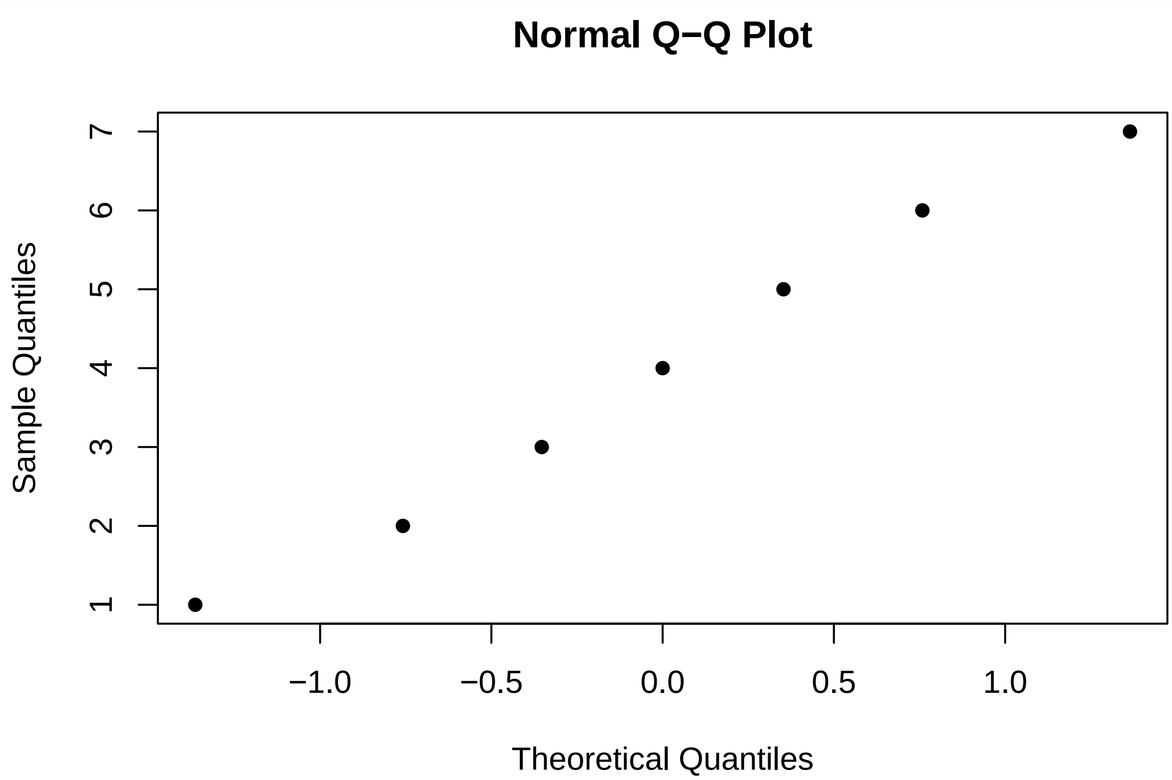
```
reg_0 = lm(Balance ~ Age + Rating, data = Credit)
vif(reg_0)
```

```
##      Age   Rating
## 1.010758 1.010758
```

Normality of error terms

qqnorm plots

```
qqnorm(1:7, pch = 16)
```

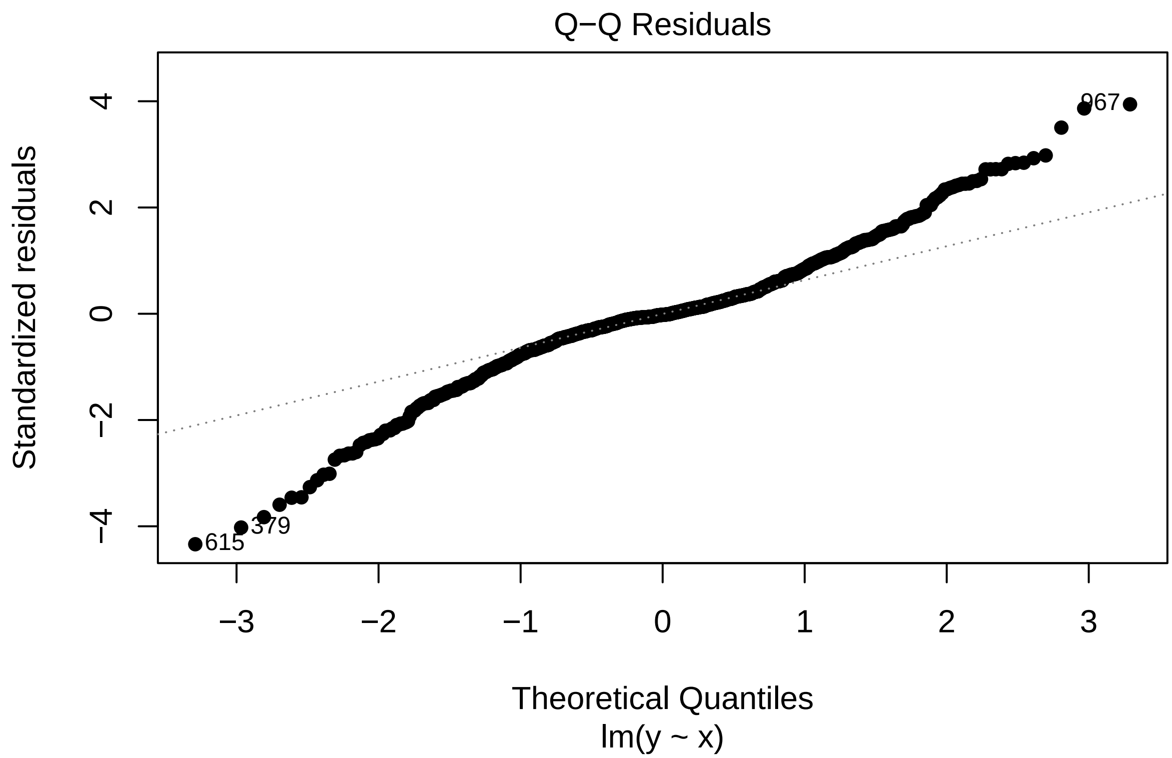


Example with error terms being clearly non-normal

```
n = 1000
x = runif(n, 0, 1)
epsilon = 100 * (x) * rnorm(n, 0, 1)
```

```r
y = 1 + 2 * x + epsilon
reg = lm(y ~ x)

plot(reg, which = 2, pch = 16)
```

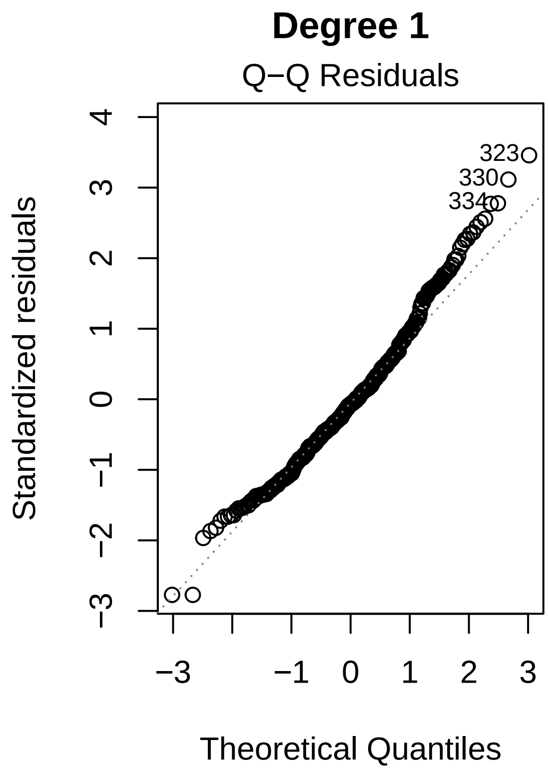


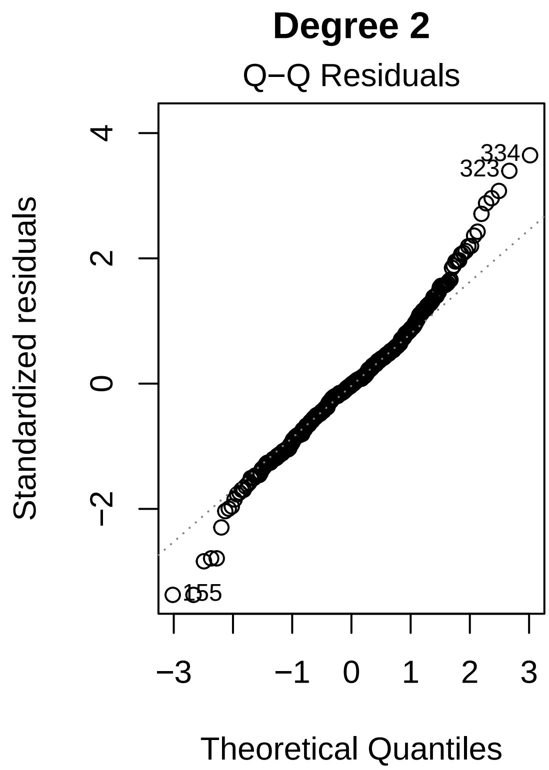
Mpg vs horsepower residual plots

```r
par(mfrow = c(1, 2))

reg_1 = lm(mpg ~ horsepower, data = Auto)
plot(reg_1, which = 2, main = "Degree 1")

reg_2 = lm(mpg ~ horsepower + I(horsepower^2), data = Auto)
plot(reg_2, which = 2, main = "Degree 2")
```


Degree 1
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
323
330
334
−3
−1
0
1
2
3
4


Degree 2
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
334
323
155
−3
−1
0
1
2
3
−2
4