Stat_364_HW_Four

Chiayu Tu (Louis Tu)

2022-11-07

Question One

Using the sat dataset, fit a model with the total SAT score as the response and expend, salary, ratio and takers as predictors. Perform regression diagnostics on this model to answer the following questions. Display any plots that are relevant. Do not provide any plots about which you have nothing to say. Suggest possible improvements or corrections to the model where appropriate.

a

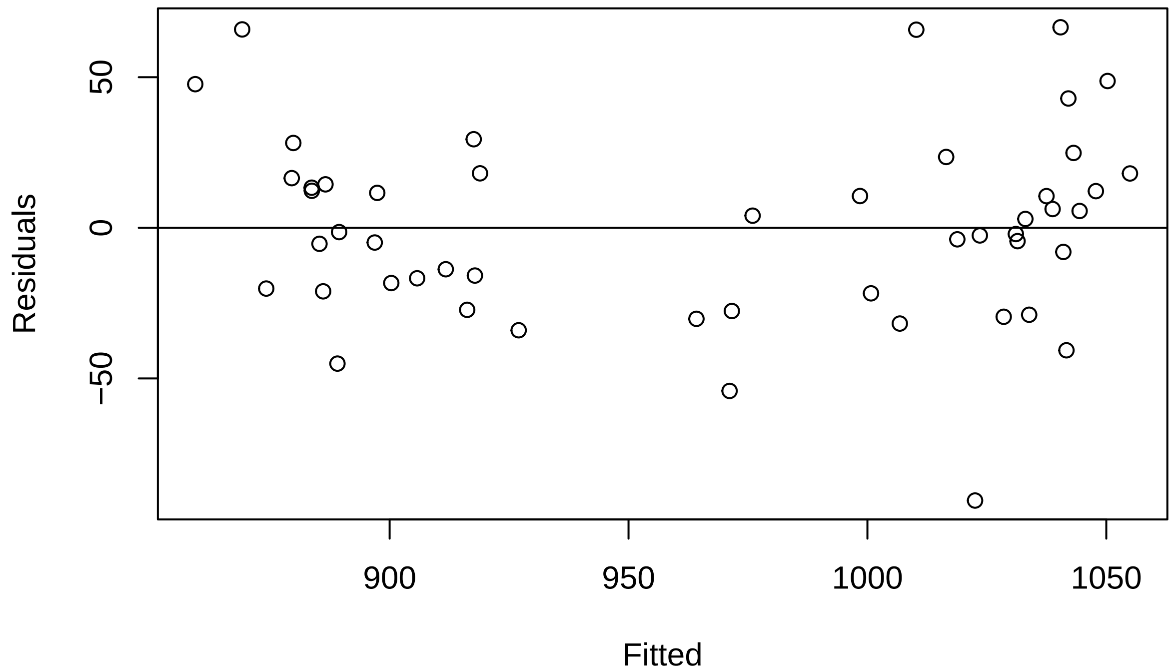
Check the constant variance assumption for the errors.

```
#view(sat)
model_s <- lm(total ~ expend + salary + ratio + takers, data = sat)
summary(model_s)
```

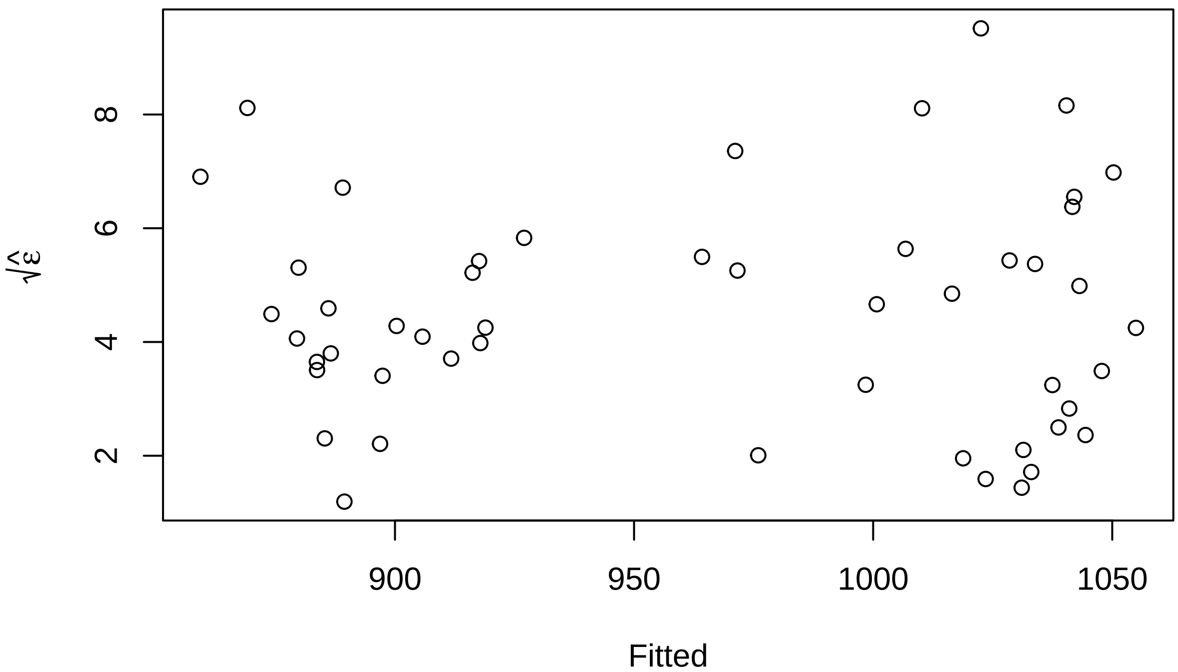
```
##
## Call:
## lm(formula = total ~ expend + salary + ratio + takers, data = sat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.531 -20.855  -1.746  15.979  66.571
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1045.9715    52.8698  19.784  < 2e-16 ***
## expend         4.4626    10.5465   0.423    0.674
## salary         1.6379     2.3872   0.686    0.496
## ratio         -3.6242     3.2154  -1.127    0.266
## takers        -2.9045     0.2313 -12.559 2.61e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.7 on 45 degrees of freedom
## Multiple R-squared:  0.8246, Adjusted R-squared:  0.809
## F-statistic: 52.88 on 4 and 45 DF,  p-value: < 2.2e-16
```

```
plot(fitted(model_s),
     residuals(model_s),
     xlab = "Fitted",
```

```
     ylab = "Residuals")
abline(h=0)
```



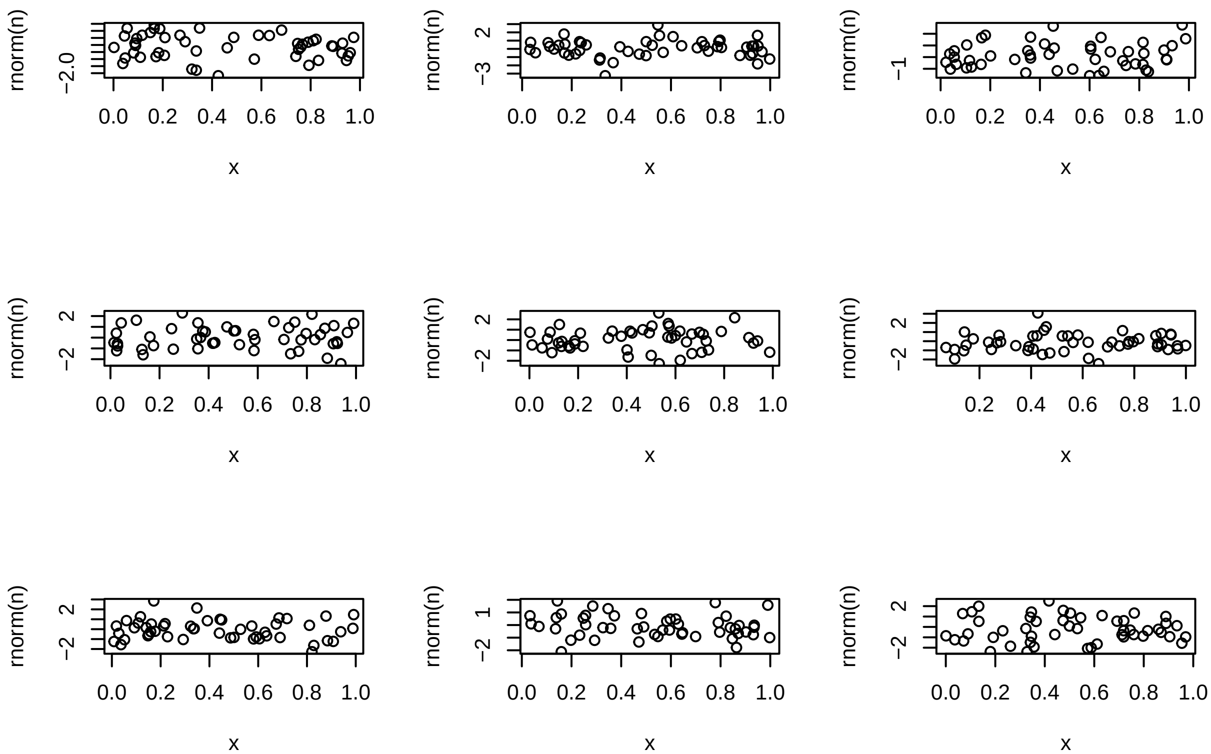

```
plot(fitted(model_s),
     sqrt(abs(residuals(model_s))),
     xlab = "Fitted",
     ylab = expression(sqrt(hat(epsilon))))
```



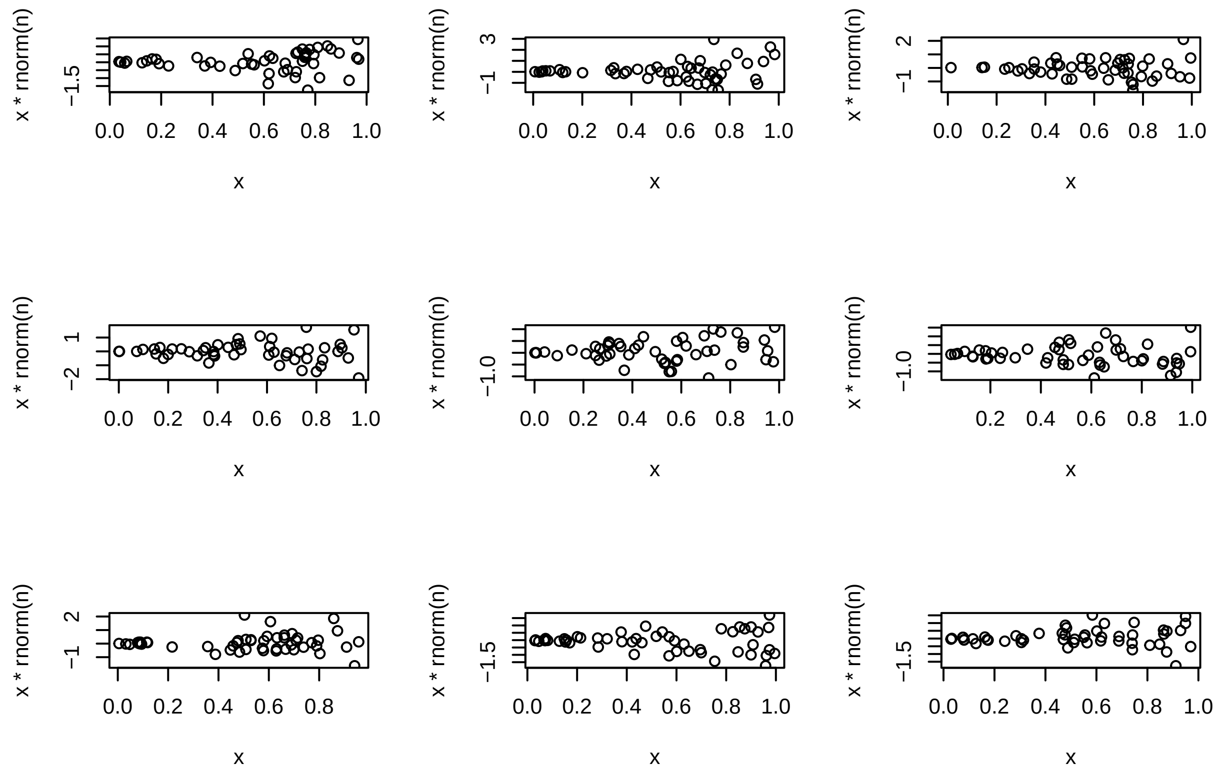

```r
summary(lm(sqrt(abs(residuals(model_s))) ~ fitted(model_s)))
```

```
##
## Call:
## lm(formula = sqrt(abs(residuals(model_s))) ~ fitted(model_s))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3097 -1.2366 -0.2234  0.9929  5.0337
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.6484524  4.0660807   1.143    0.259
## fitted(model_s) -0.0001637  0.0041994  -0.039    0.969
##
## Residual standard error: 1.997 on 48 degrees of freedom
## Multiple R-squared:  3.166e-05,  Adjusted R-squared:  -0.0208
## F-statistic: 0.00152 on 1 and 48 DF,  p-value: 0.9691
```

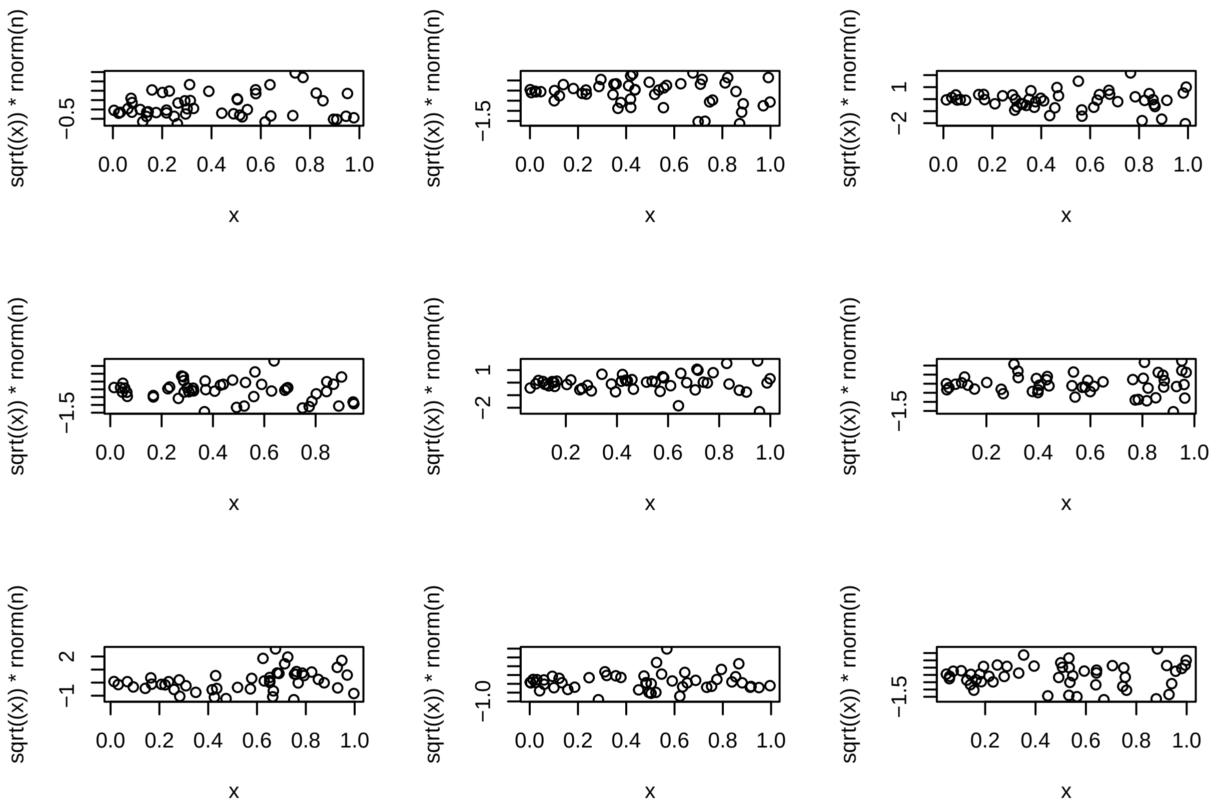
```r
par(mfrow=c(3,3))
n <- 50
for(i in 1:9) {x <- runif(n) ; plot(x, rnorm(n))}
```



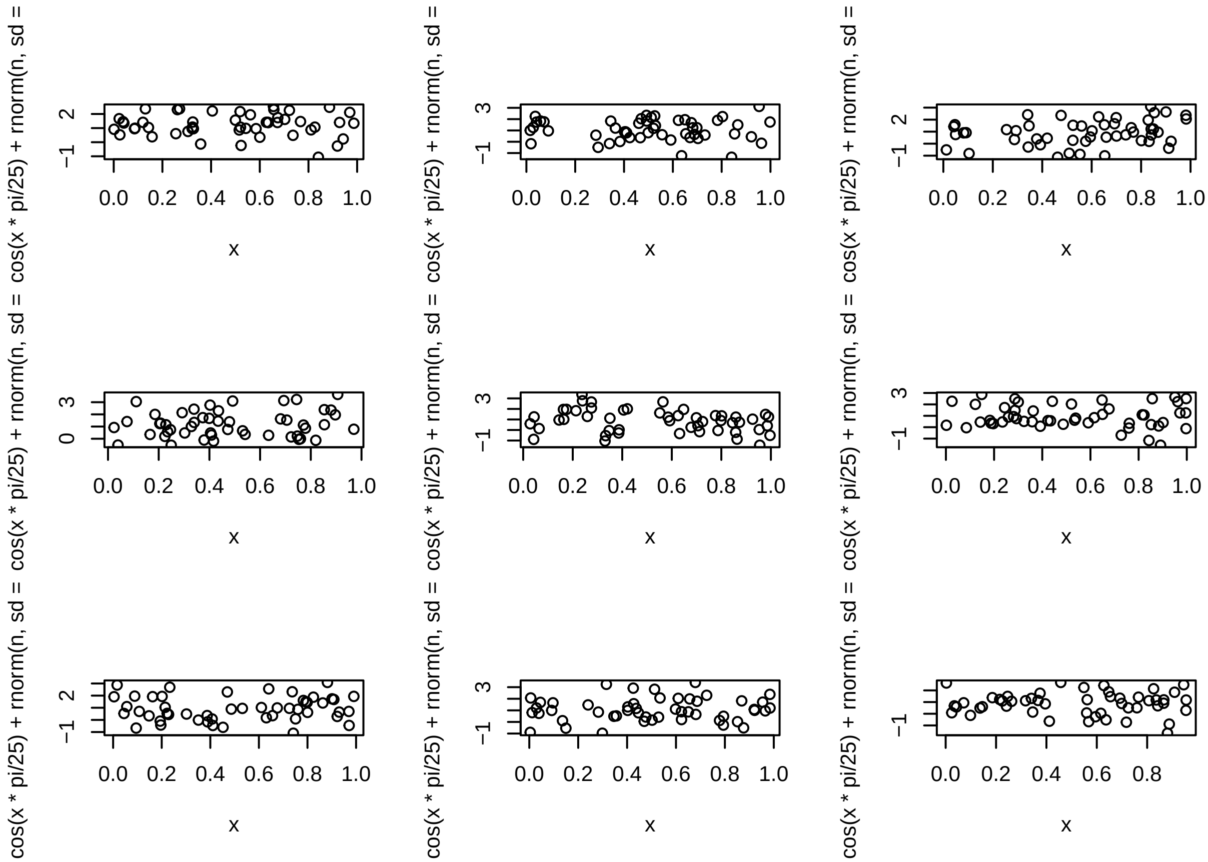

```
for(i in 1:9) {x <- runif(n) ; plot(x, x*rnorm(n))}
```




```
for(i in 1:9) {x <- runif(n) ; plot(x, sqrt ((x)) * rnorm(n))}
```




```
for(i in 1:9) {x <- runif(n) ; plot(x, cos(x*pi/25)+rnorm(n, sd = 1))}
```



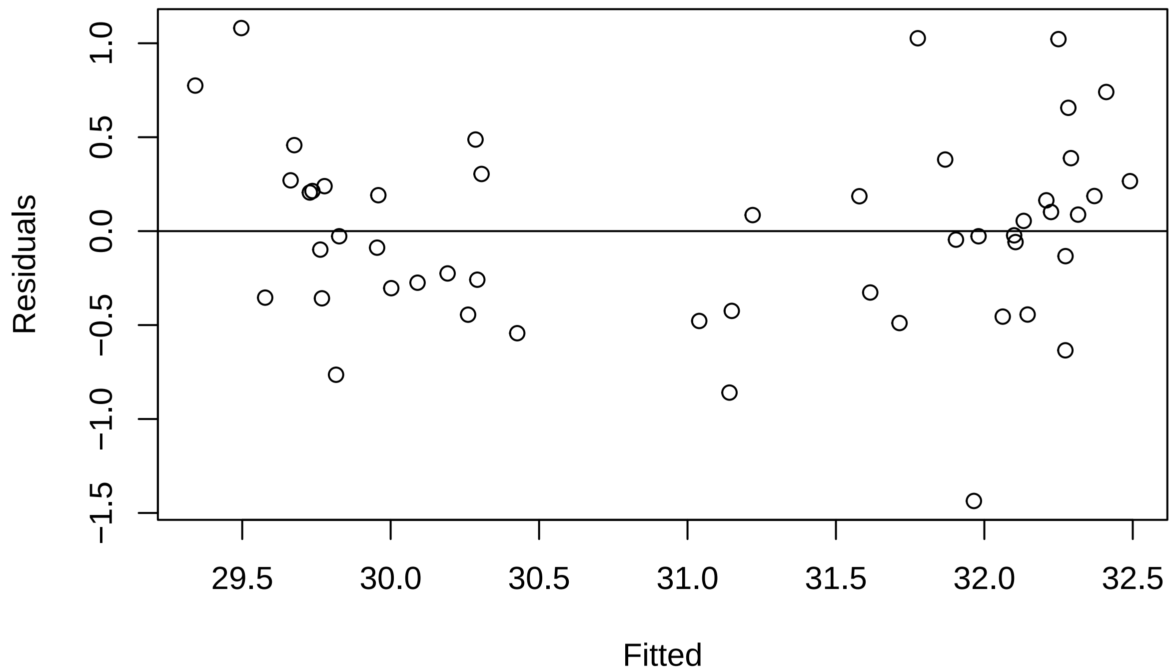
```r
par(mfrow=c(1,1))
```

```r
model_s_1 <- lm(sqrt(total) ~ expend + salary + ratio + takers, data = sat)
summary(model_s_1)
```

```
##
## Call:
## lm(formula = sqrt(total) ~ expend + salary + ratio + takers,
##     data = sat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.43610 -0.34707 -0.02486  0.25943  1.08084
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 32.297536   0.840140  38.443   <2e-16 ***
## expend       0.075430   0.167592   0.450    0.655
## salary       0.026203   0.037935   0.691    0.493
## ratio       -0.056489   0.051095  -1.106    0.275
## takers      -0.046730   0.003675 -12.716   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5197 on 45 degrees of freedom
```

```
## Multiple R-squared:  0.8278, Adjusted R-squared:  0.8125
## F-statistic: 54.08 on 4 and 45 DF,  p-value: < 2.2e-16
```

```
plot(fitted(model_s_1),
     residuals(model_s_1),
     xlab = "Fitted",
     ylab = "Residuals")
abline(h=0)
```




```
mean(sat$expend)
```

```
## [1] 5.90526
```

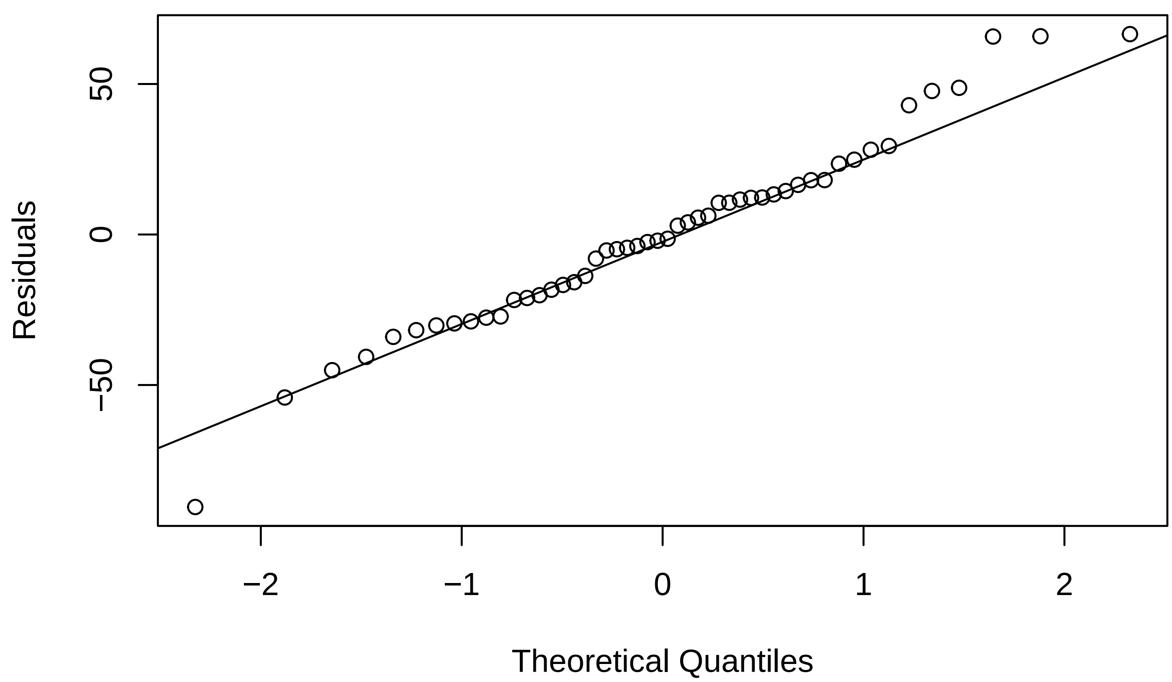
```
var.test(residuals(model_s_1)[sat$expend>5.90526], residuals(model_s_1)[sat$exp<5.90526])
```

```
##
##  F test to compare two variances
##
## data:  residuals(model_s_1)[sat$expend > 5.90526] and residuals(model_s_1)[sat$exp < 5.90526]
## F = 0.89354, num df = 21, denom df = 27, p-value = 0.8004
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.3994815 2.0862745
## sample estimates:
## ratio of variances
##          0.893537
```

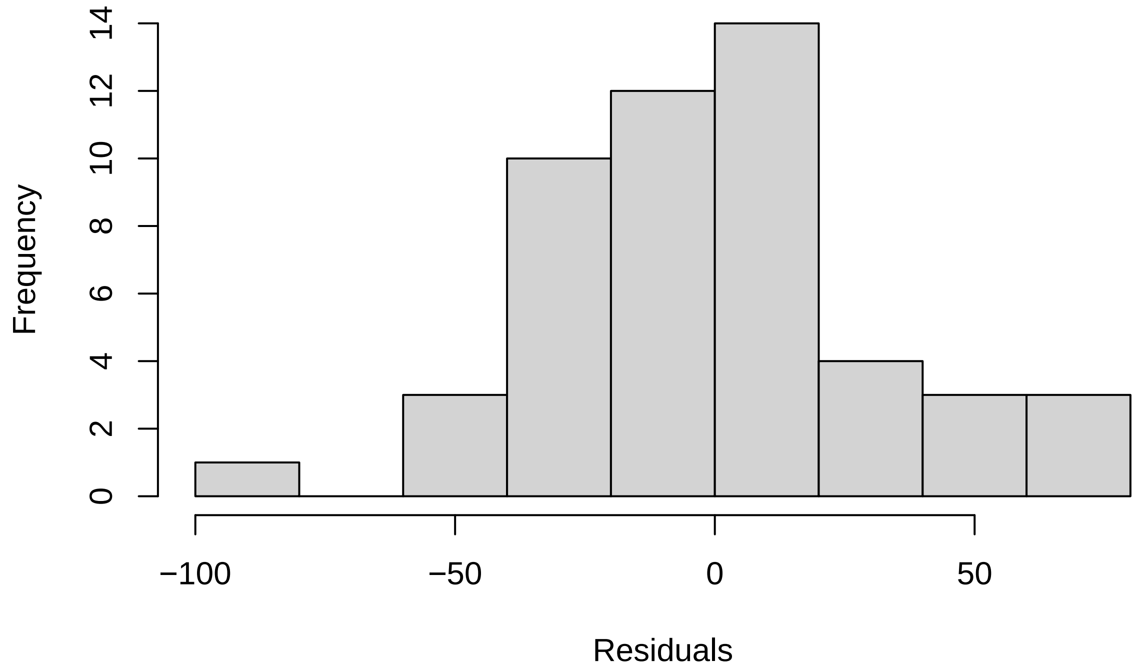
b

Check the normality assumption.

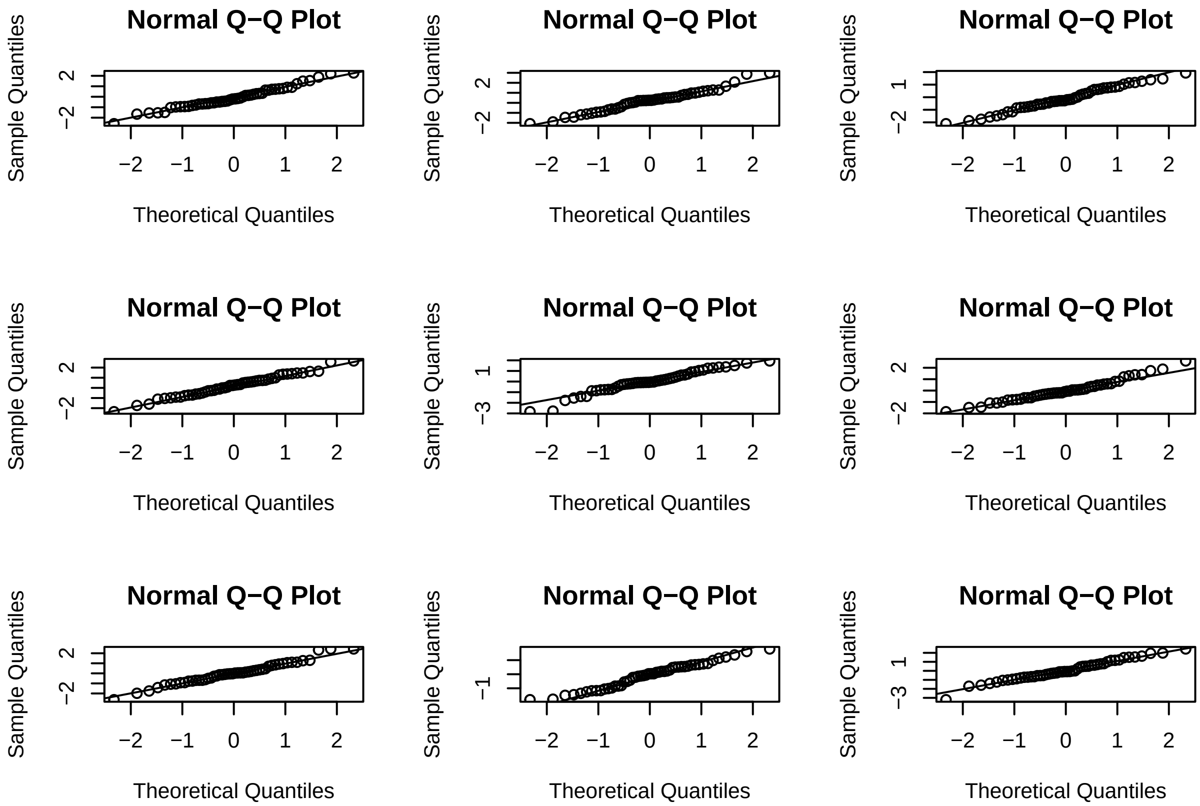
```
qqnorm(residuals(model_s), ylab = "Residuals", main = "")
qqline(residuals(model_s))
```



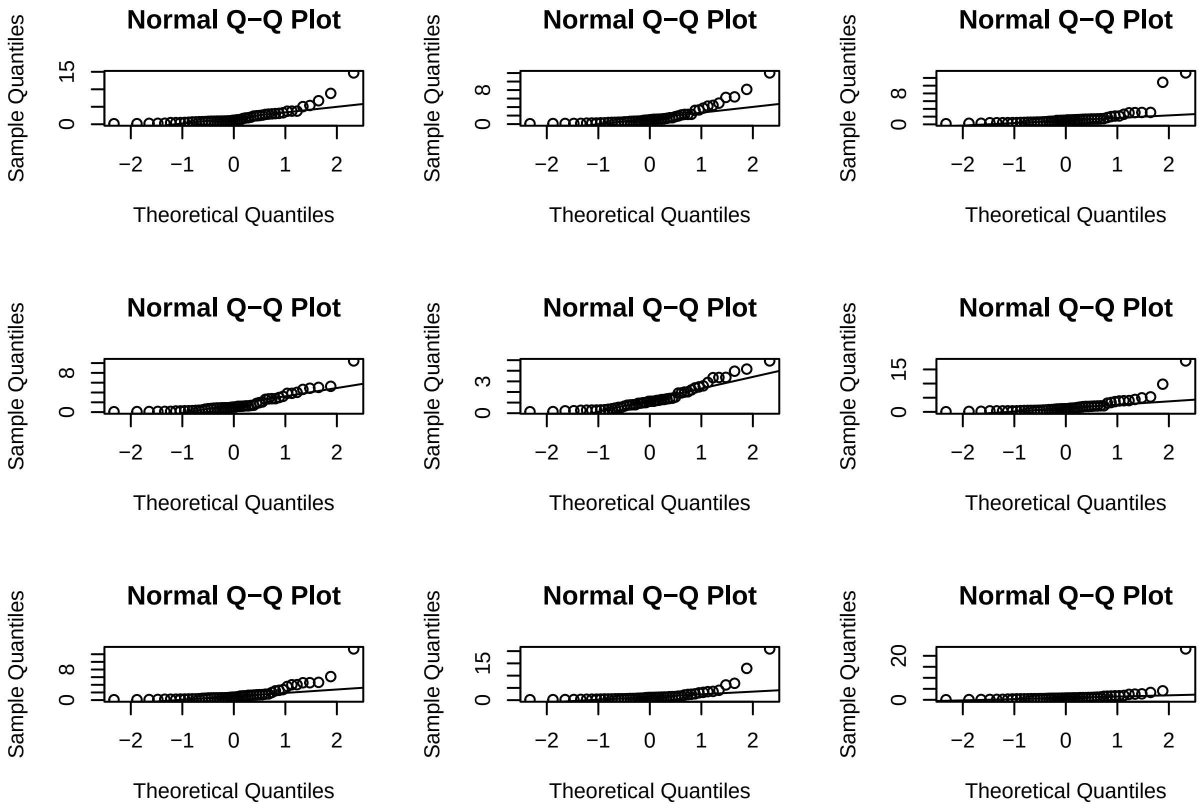

```
hist(residuals(model_s),xlab="Residuals",main="")
```



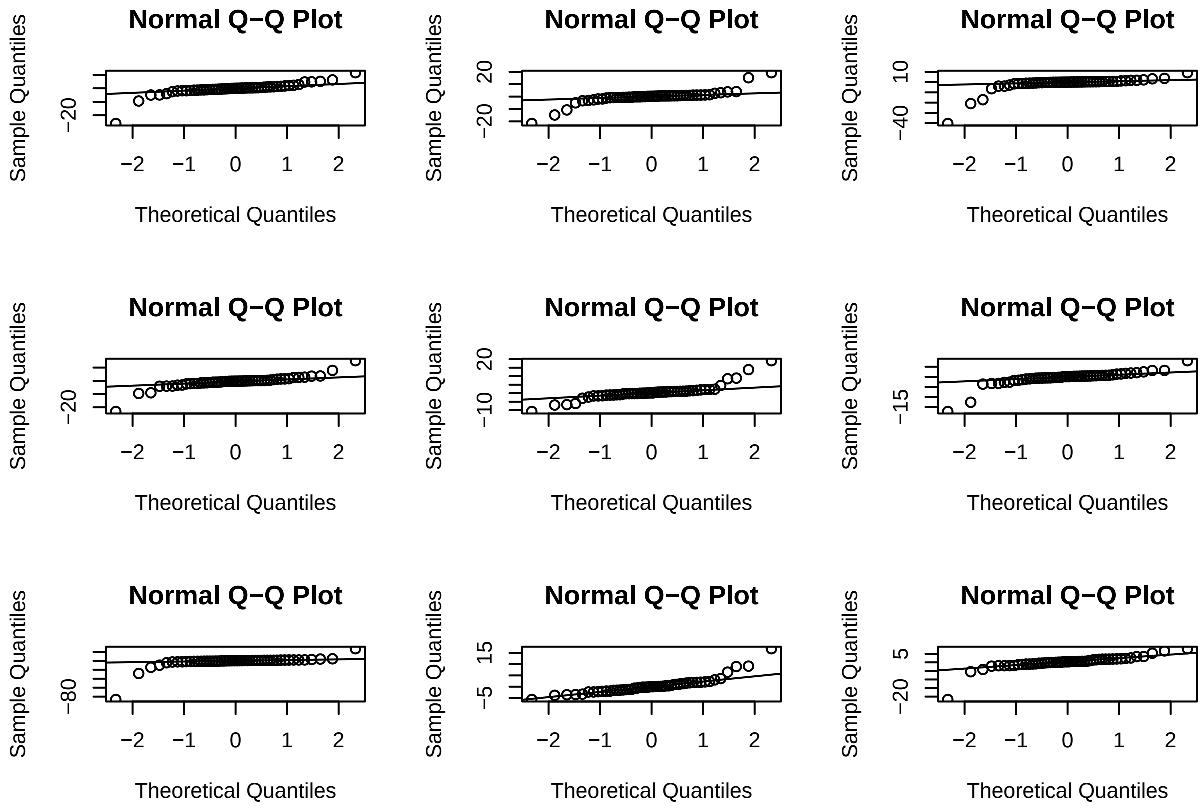

```
par(mfrow=c(3,3))
n <- 50
for(i in 1:9) {x <- rnorm(n) ; qqnorm(x) ; qqline(x)}
```



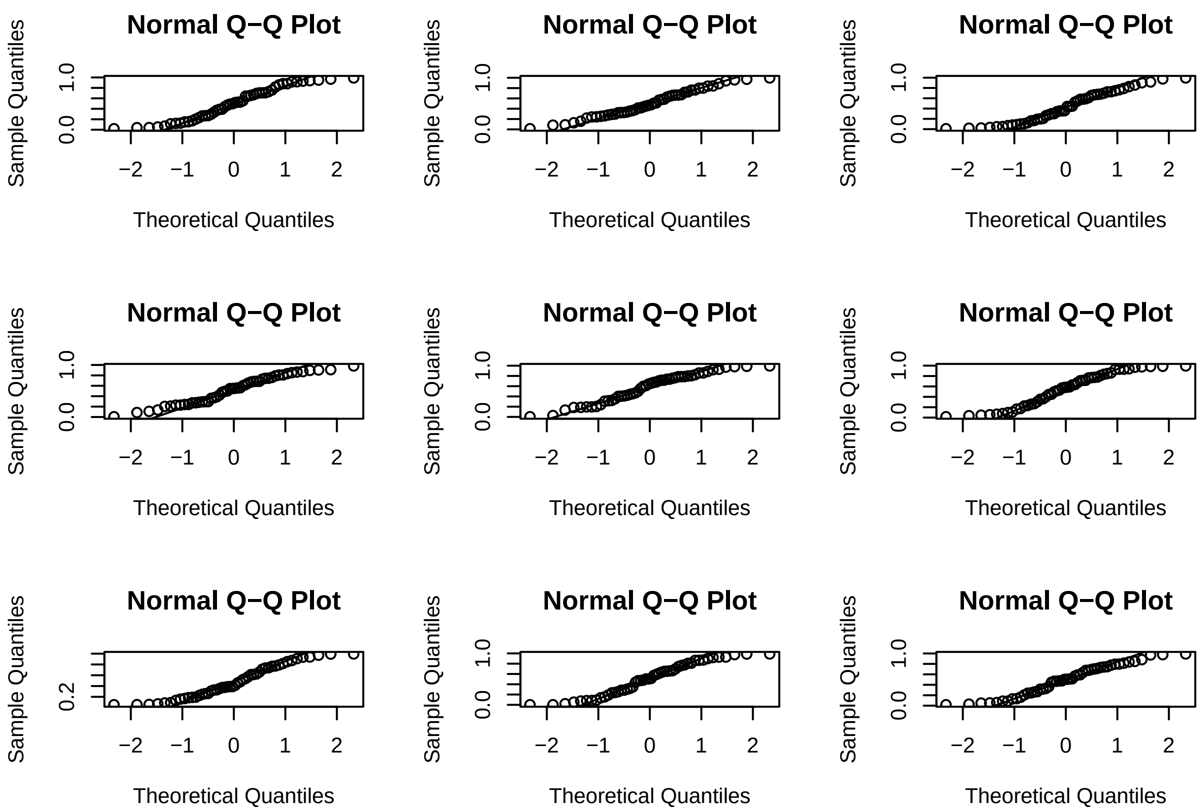

```r
for(i in 1:9) {x <- exp(rnorm(n)); qqnorm(x); qqline(x)}
```




```r
for(i in 1:9) {x <- rcauchy(n); qqnorm(x); qqline(x)}
```




```r
for(i in 1:9) {x <- runif(n); qqnorm(x); qqline(x)}
```




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

shapiro.test(residuals(model_s))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_s)
## W = 0.97691, p-value = 0.4304
```

c

Check for large leverage points.

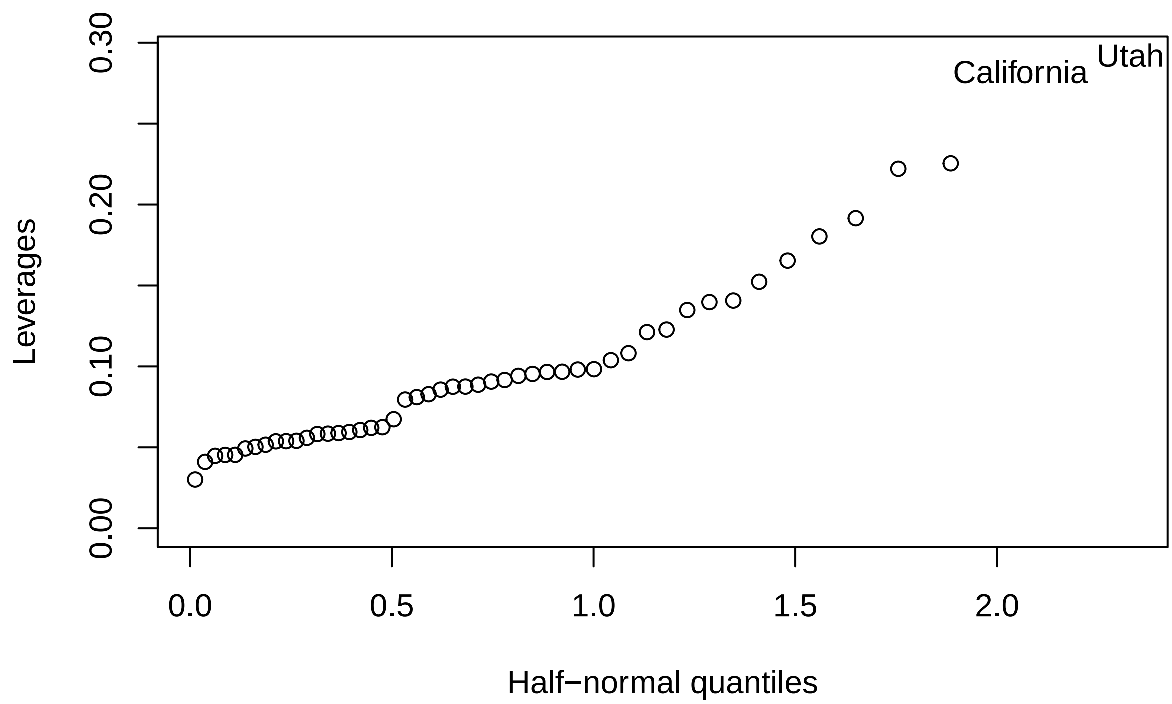
```
hatv <- hatvalues(model_s)
head(hatv)
```

```
##    Alabama     Alaska    Arizona   Arkansas California   Colorado
## 0.09537668 0.18030612 0.04931612 0.05382878 0.28211791 0.03014533
```

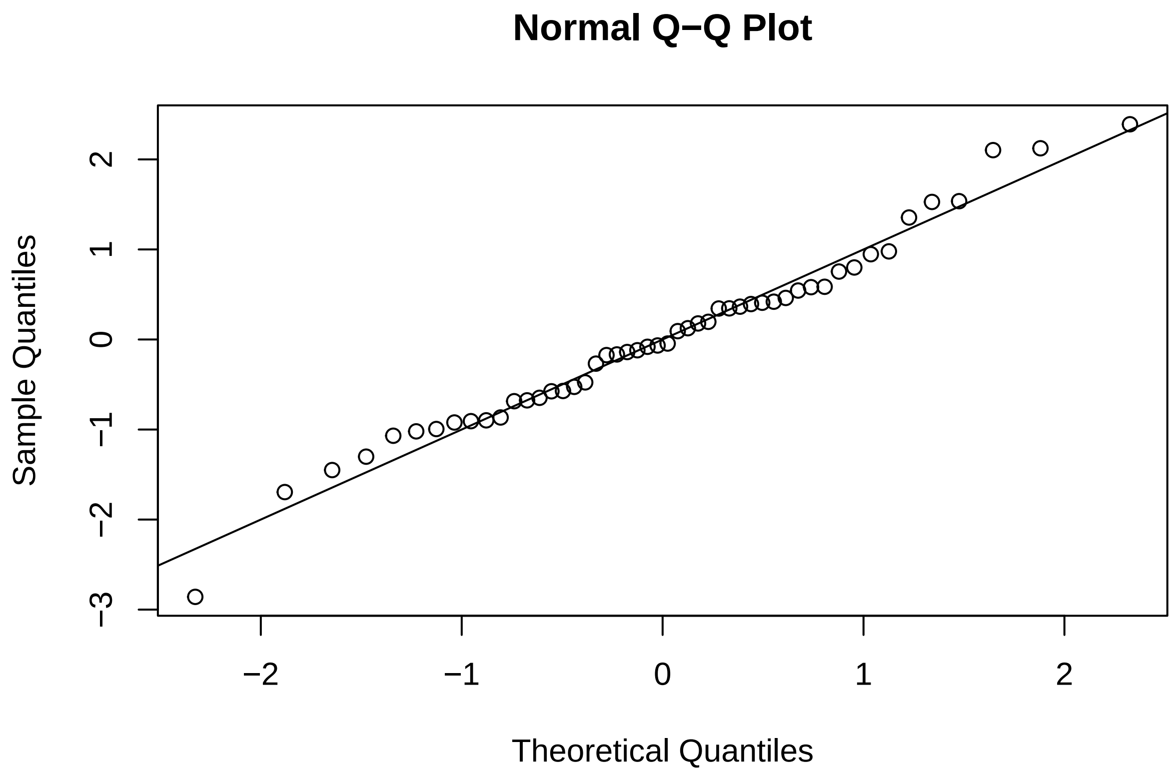
```
sum(hatv)
```

```
## [1] 5
```

```r
states <- row.names(sat)
halfnorm(hatv, labs = states, ylab = "Leverages")
```



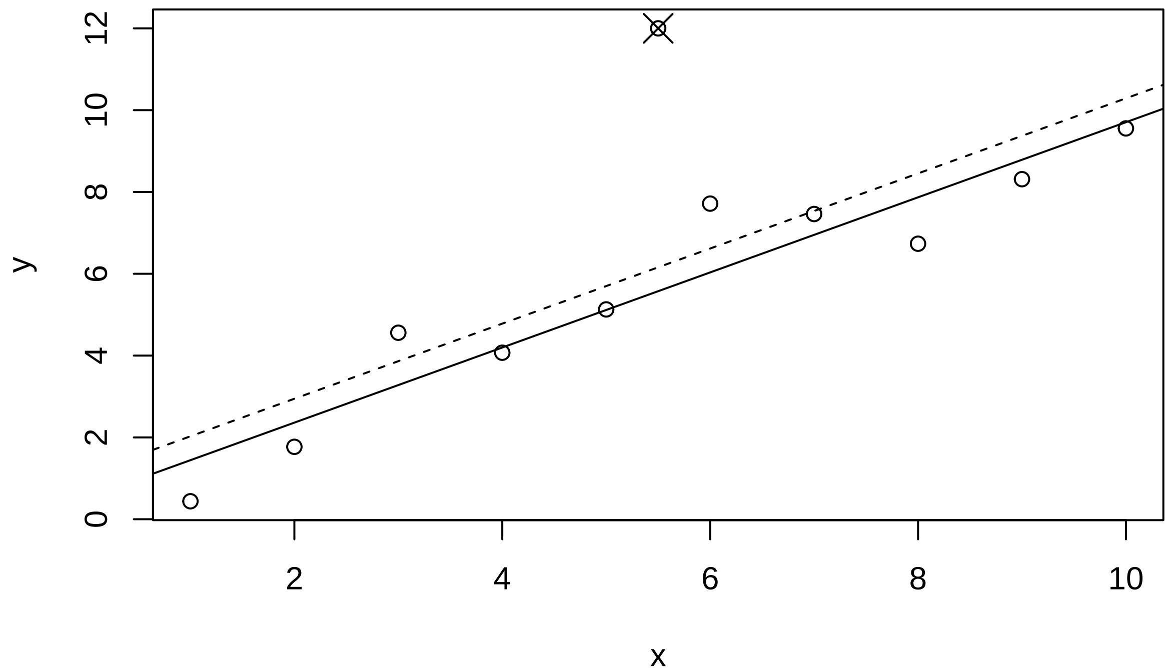

```r
qqnorm(rstandard(model_s))
abline(0,1)
```



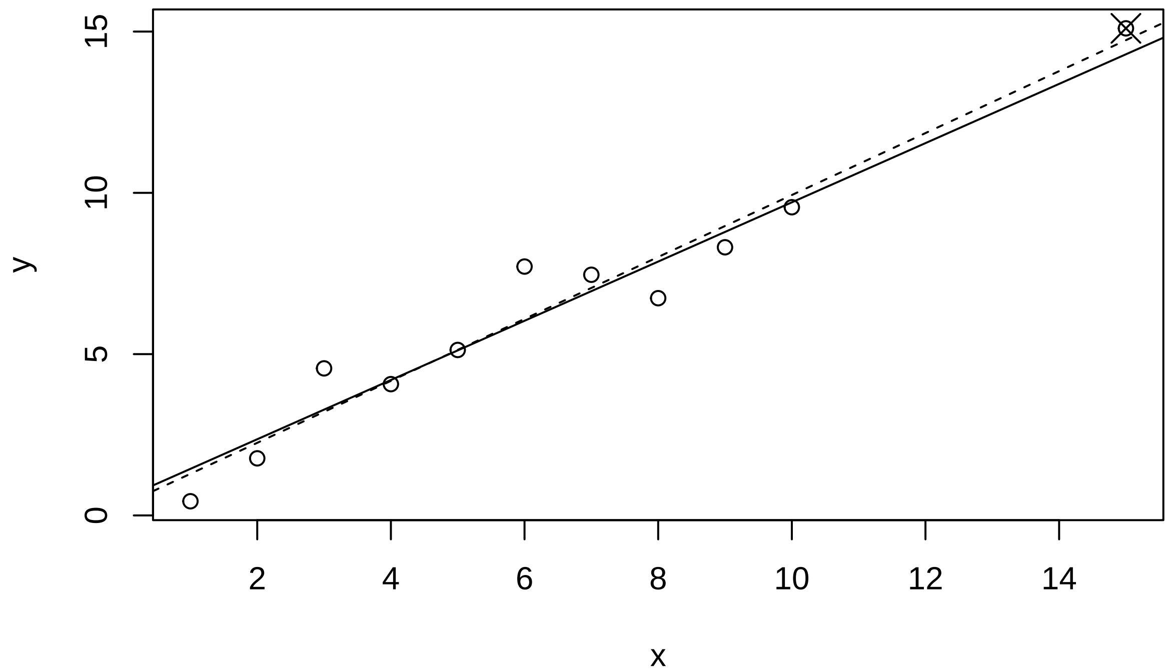
d

Check for outliers.

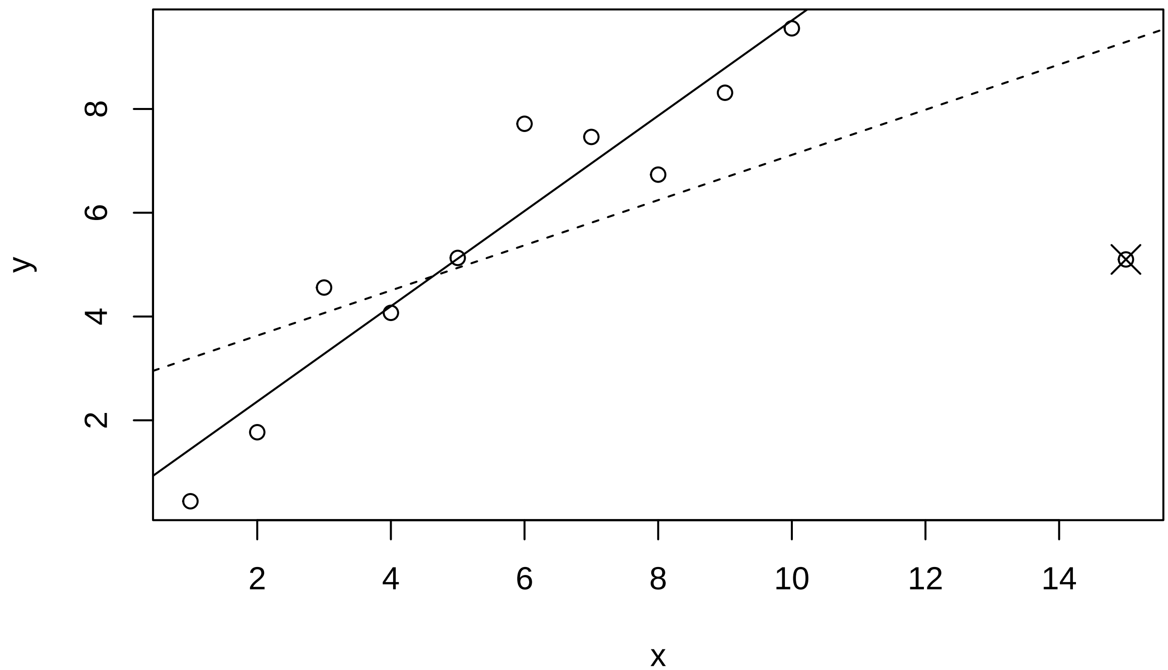
```
set.seed(123)
testdata <- data.frame(x = 1:10,
                       y = 1:10 + rnorm(10))
model_s_2 <- lm(y ~ x, testdata)

p1 <- c(5.5, 12)
model_s_3 <- lm(y ~ x, rbind(testdata, p1))
plot(y ~ x, rbind(testdata, p1))
points(5.5, 12, pch=4, cex=2)
abline(model_s_2)
abline(model_s_3, lty=2)
```




```
p2 <- c(15,15.1)
model_s_4 <- lm(y ~ x, rbind(testdata, p2))
plot(y ~ x, rbind(testdata, p2))
points(15, 15.1, pch=4, cex=2)
abline(model_s_2)
abline(model_s_4, lty=2)
```




```
p3 <- c(15,5.1)
model_s_5 <- lm(y ~ x, rbind(testdata, p3))
plot(y ~ x, rbind(testdata, p3))
points(15, 5.1, pch=4, cex=2)
abline(model_s_2)
abline(model_s_5, lty=2)
```




```
stud <- rstudent(model_s)
stud[which.max(abs(stud))]
```

```
## West Virginia
##     -3.124428
```

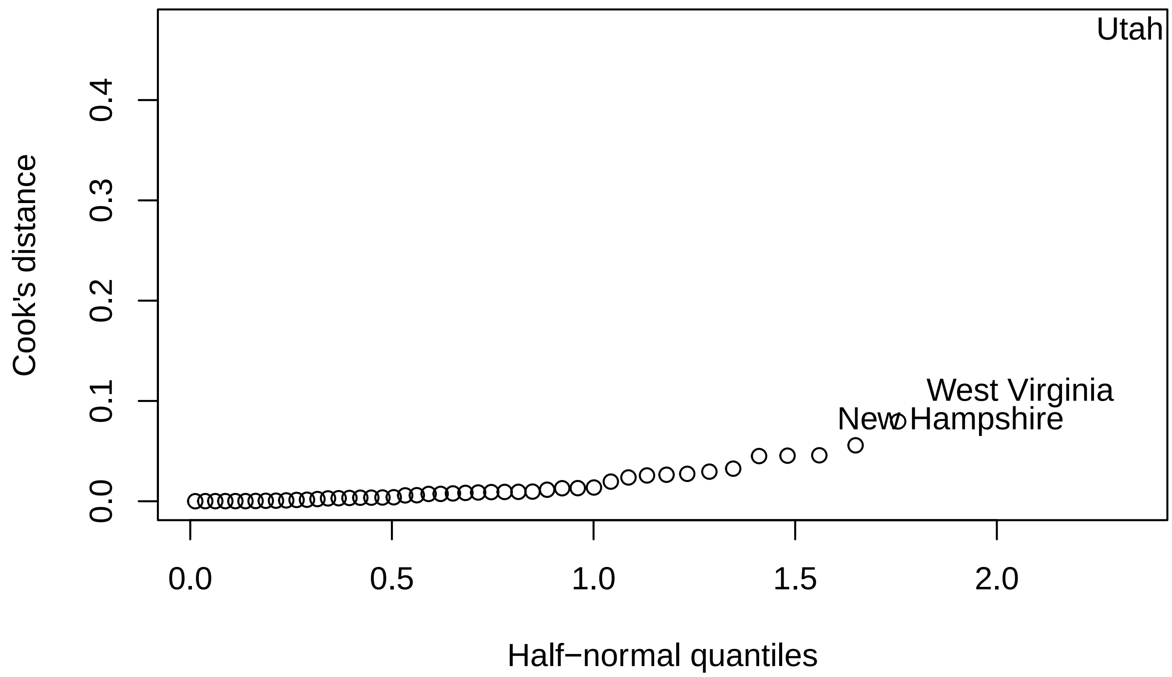
```
qt(0.05/(50*2),44)
```

```
## [1] -3.525801
```

e

Check for influential points.

```
cook <- cooks.distance(model_s)
halfnorm(cook, 3, labs = states, ylab = "Cook's distance")
```



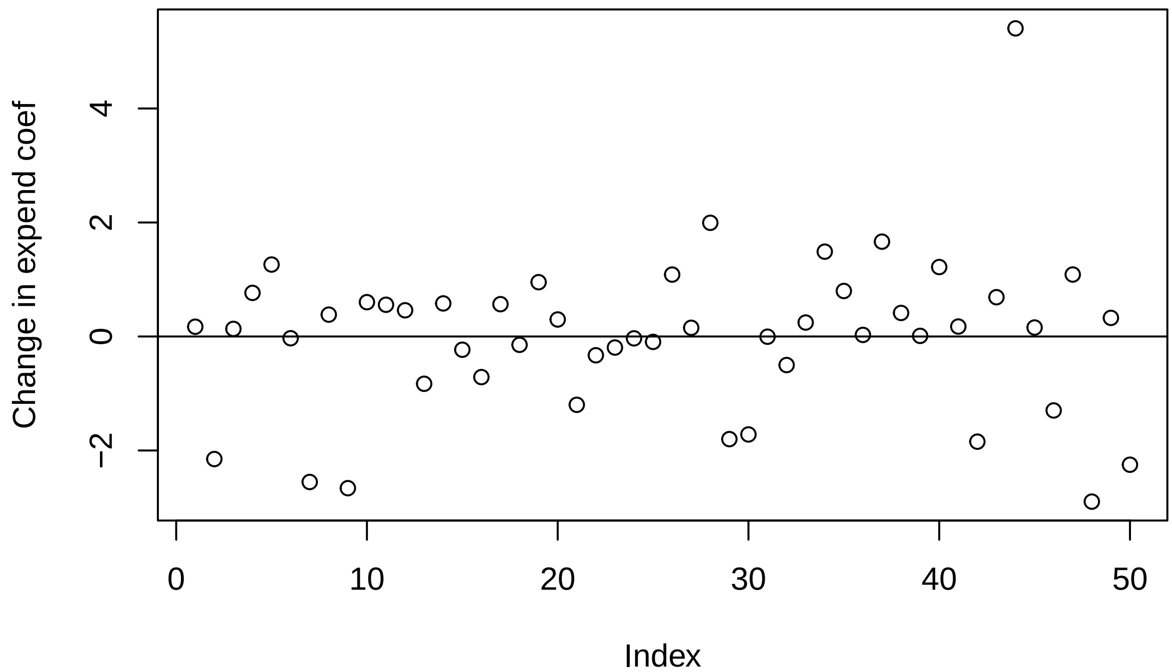

```
model_s_6 <- lm(total ~ expend + salary + ratio + takers, data = sat, subset = (cook < max(cook)))
summary(model_s_6)
```

```
##
## Call:
## lm(formula = total ~ expend + salary + ratio + takers, data = sat,
##     subset = (cook < max(cook)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -92.118 -18.402   1.808  14.890  67.669
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1093.8460    53.4226  20.475   <2e-16 ***
## expend        -0.9427    10.1922  -0.092    0.927
## salary         3.0964     2.3283   1.330    0.190
## ratio         -7.6391     3.4279  -2.229    0.031 *
## takers        -2.9308     0.2188 -13.397   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.9 on 44 degrees of freedom
## Multiple R-squared:  0.8396, Adjusted R-squared:  0.825
## F-statistic: 57.58 on 4 and 44 DF,  p-value: < 2.2e-16
```

```r
summary(model_s)
```

```
##
## Call:
## lm(formula = total ~ expend + salary + ratio + takers, data = sat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.531 -20.855  -1.746  15.979  66.571
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1045.9715    52.8698  19.784  < 2e-16 ***
## expend         4.4626    10.5465   0.423    0.674
## salary         1.6379     2.3872   0.686    0.496
## ratio         -3.6242     3.2154  -1.127    0.266
## takers        -2.9045     0.2313 -12.559 2.61e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.7 on 45 degrees of freedom
## Multiple R-squared:  0.8246, Adjusted R-squared:  0.809
## F-statistic: 52.88 on 4 and 45 DF,  p-value: < 2.2e-16
```

```r
plot(dfbeta(model_s)[,2],
     ylab = "Change in expend coef")
abline(h=0)
```



f

Check the structure of the relationship between the predictors and the response.

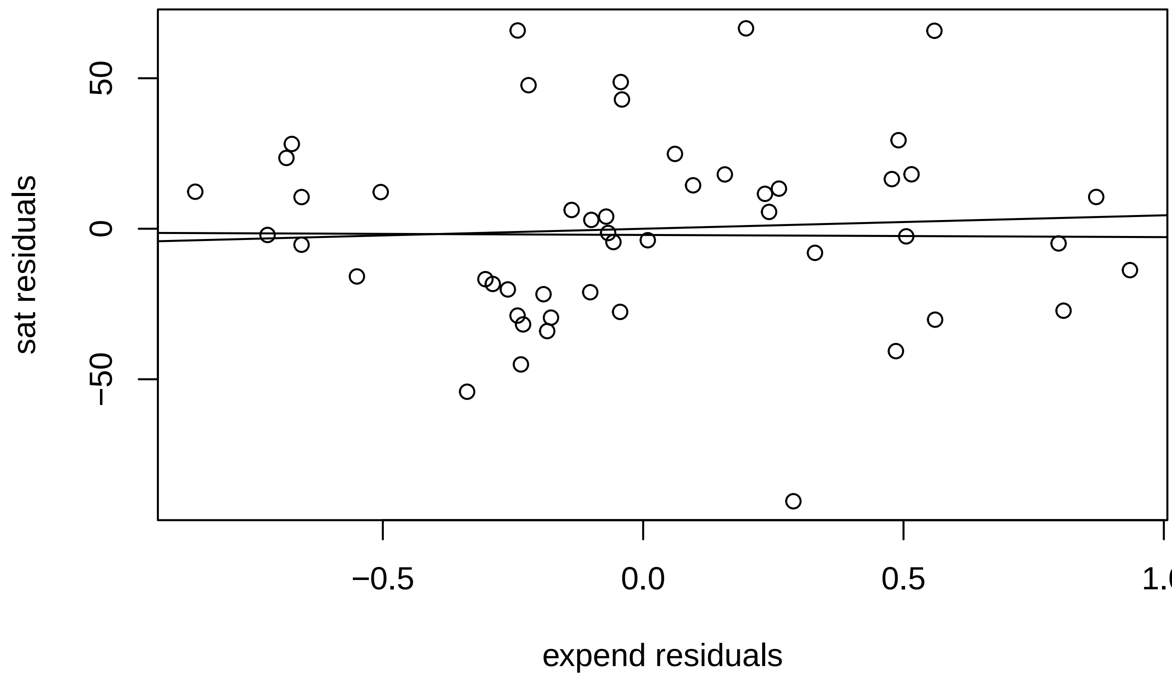
```
model_s_7 <- residuals(lm(total ~ expend + salary + ratio + takers, data = sat))
model_s_8 <- residuals(lm(expend ~ salary + ratio + takers, data = sat))
plot(model_s_8, model_s_7, xlab = "expend residuals", ylab = "sat residuals")
abline(model_s_7, model_s_8)

coef(lm(model_s_7 ~ model_s_8))
```

```
##   (Intercept)     model_s_8
## -6.280370e-16 -1.145765e-15
```

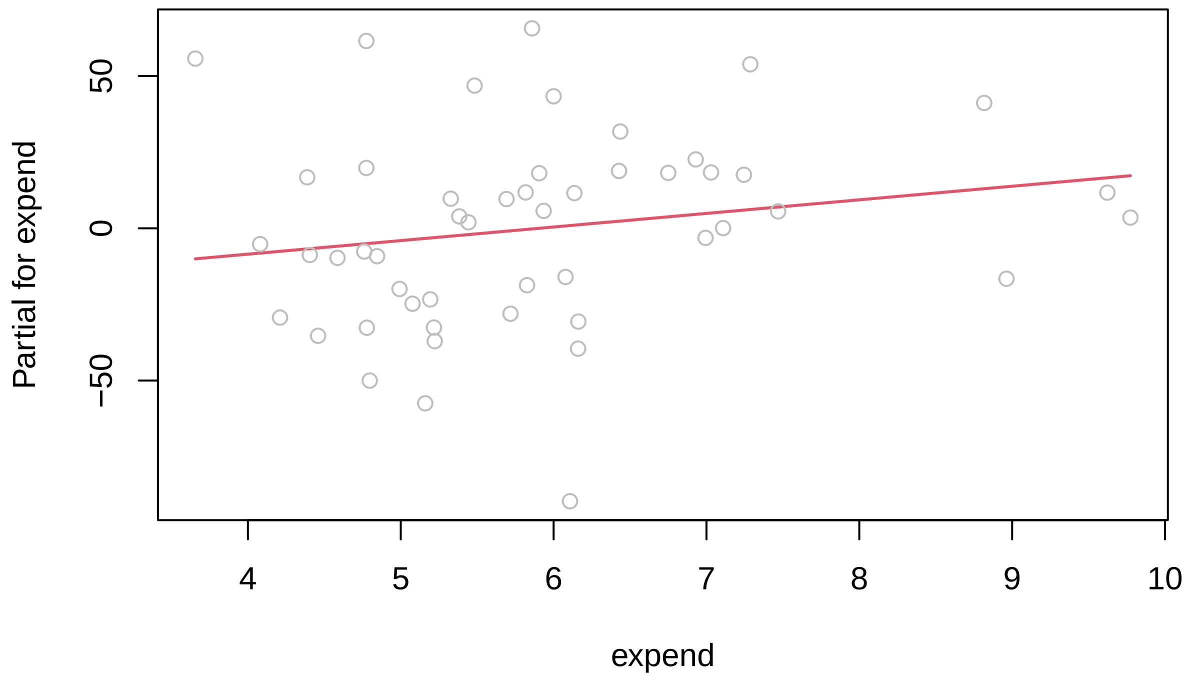
```
coef(model_s)
```

```
## (Intercept)      expend      salary       ratio      takers
## 1045.971536    4.462594    1.637917   -3.624232   -2.904481
```

```
abline(0, coef(model_s)['expend'])
```




```
termplot(model_s,
         partial.resid = T,
         terms = 1)
```



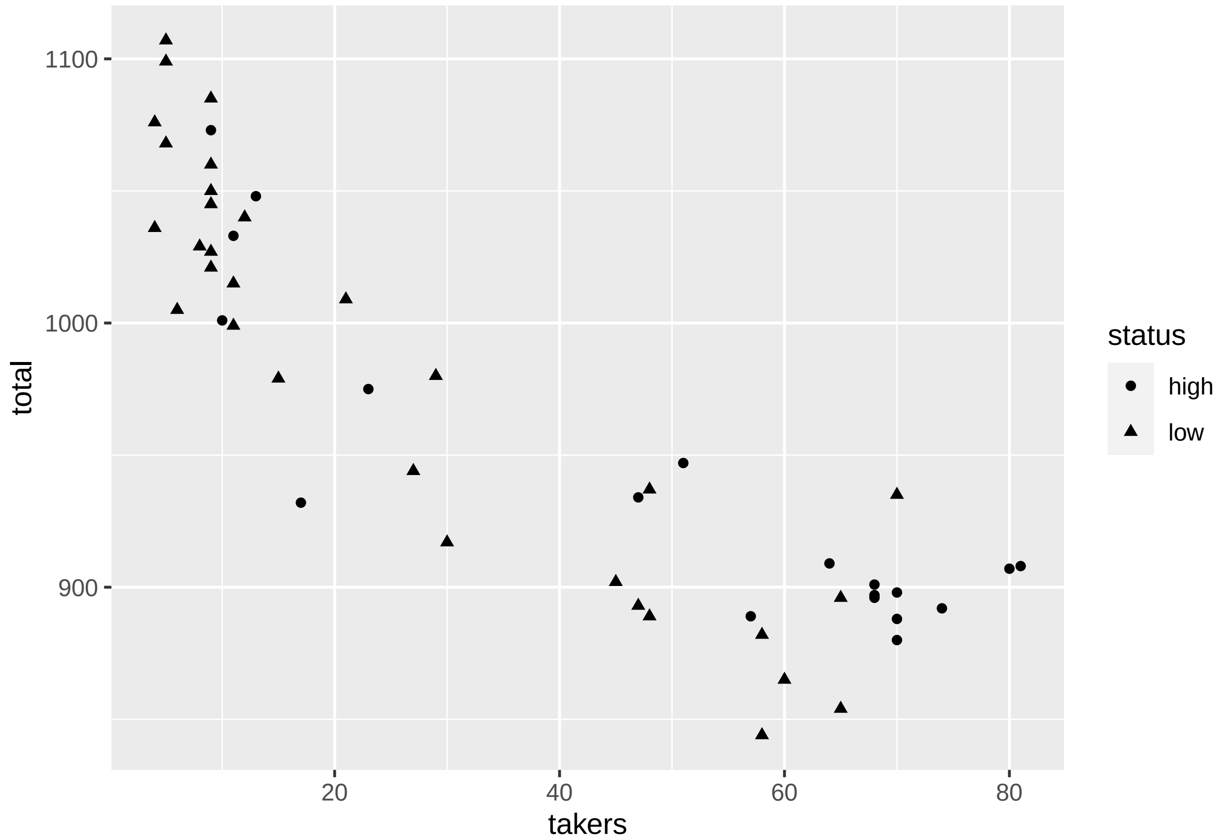

```r
model_s_9 <- lm(total ~ expend + salary + ratio + takers, data = sat, subset = (expend>6))
model_s_10 <- lm(total ~ expend + salary + ratio + takers, data = sat, subset = (expend<6))
summary(model_s_9)
```

```
##
## Call:
## lm(formula = total ~ expend + salary + ratio + takers, data = sat,
##     subset = (expend > 6))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -59.818 -13.456   3.091  15.186  49.887
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 994.7970    87.3748  11.385 1.83e-08 ***
## expend       -8.6936    14.2395  -0.611    0.551
## salary        4.0757     3.2329   1.261    0.228
## ratio        -2.8374     5.4074  -0.525    0.608
## takers       -2.2405     0.3276  -6.838 8.08e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.69 on 14 degrees of freedom
## Multiple R-squared:  0.8209, Adjusted R-squared:  0.7697
## F-statistic: 16.04 on 4 and 14 DF,  p-value: 3.995e-05
```

```r
summary(model_s_10)
```

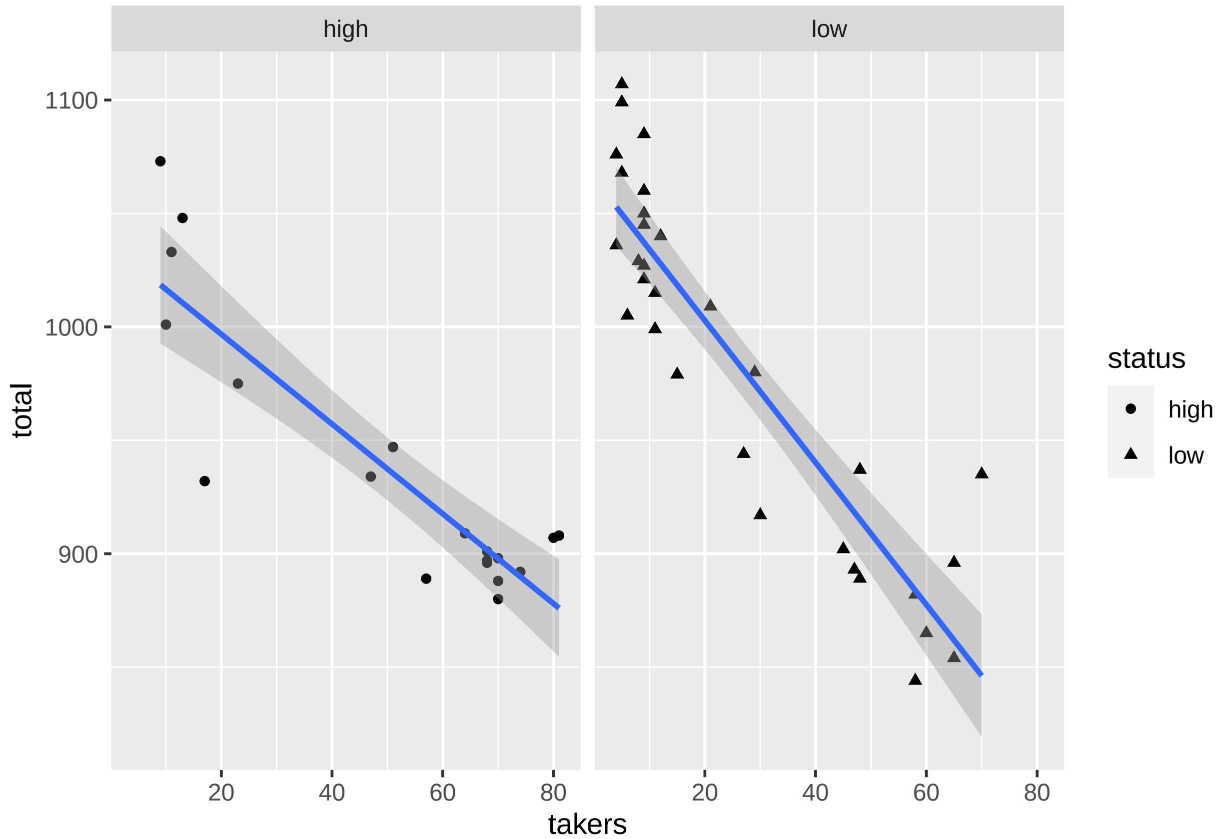
```
##
## Call:
## lm(formula = total ~ expend + salary + ratio + takers, data = sat,
##     subset = (expend < 6))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.233 -19.527  -5.211  12.643  79.212
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1017.9597    90.6421  11.231 2.93e-11 ***
## expend        18.7994    16.7715   1.121    0.273
## salary        -0.5454     3.2032  -0.170    0.866
## ratio         -1.6764     3.8813  -0.432    0.670
## takers        -3.2461     0.3297  -9.846 4.39e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.26 on 25 degrees of freedom
## Multiple R-squared:  0.8506, Adjusted R-squared:  0.8267
## F-statistic: 35.59 on 4 and 25 DF,  p-value: 5.544e-10
```

```r
sat$status <- ifelse(sat$expend > 6, "high", "low")
require(ggplot2)
ggplot(sat, aes(x = takers, y = total, shape = status)) +
  geom_point()
```




```
ggplot(sat, aes(x = takers, y = total, shape = status)) +
  geom_point() +
  facet_grid(~ status) +
  stat_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Question Two

Using the teengamb dataset, fit a model with gamble as the response and the other variables as predictors. Answer the questions posed in the previous question.

a

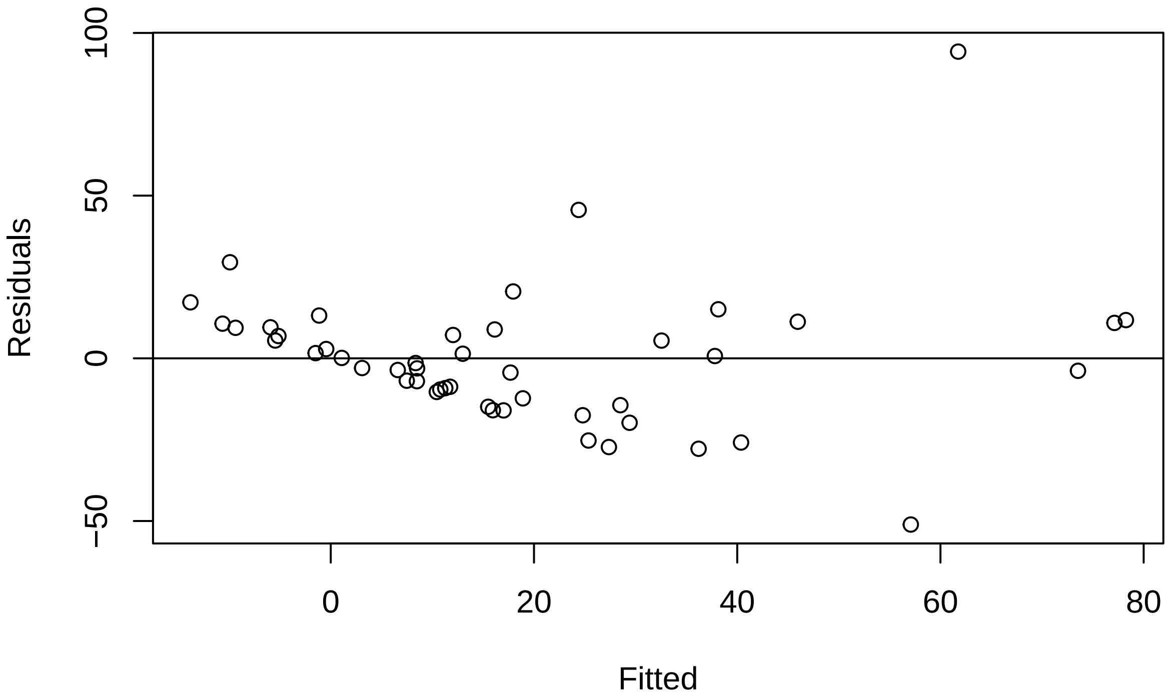
Check the constant variance assumption for the errors.

```
#view(teengamb)
model_t <- lm(gamble ~ ., data = teengamb)
summary(model_t)
```

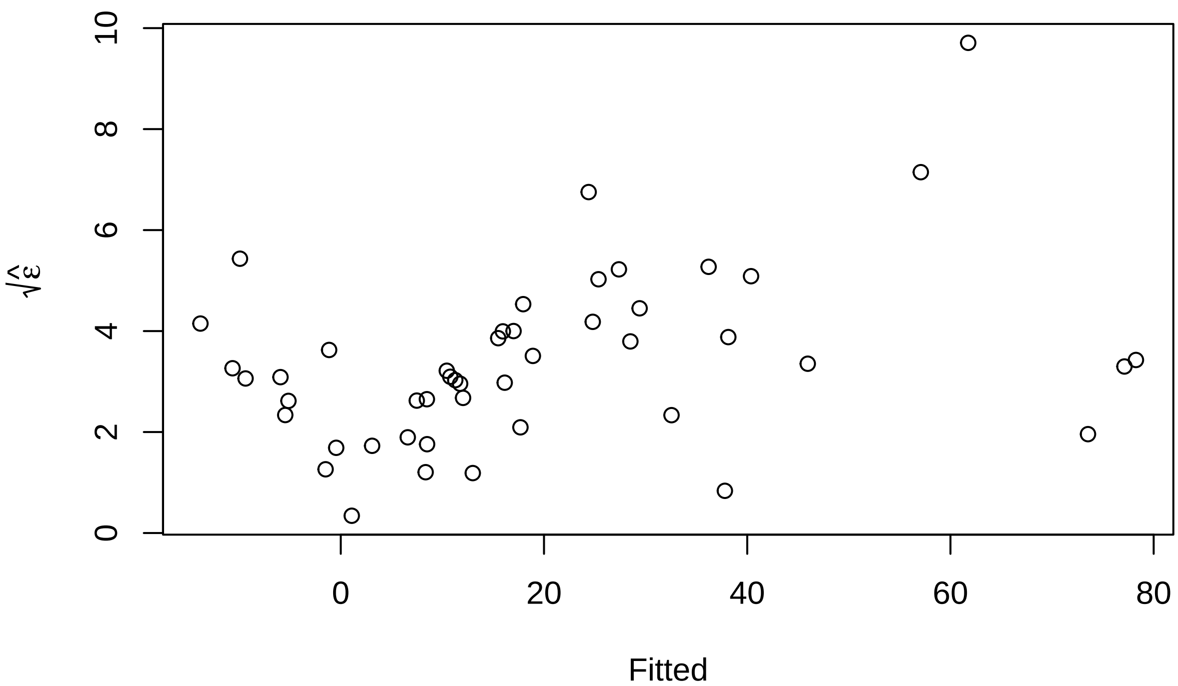
```
##
## Call:
## lm(formula = gamble ~ ., data = teengamb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.082 -11.320  -1.451   9.452  94.252
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.55565   17.19680   1.312   0.1968
```

```
## sex         -22.11833    8.21111  -2.694   0.0101 *
## status        0.05223    0.28111   0.186   0.8535
## income        4.96198    1.02539   4.839 1.79e-05 ***
## verbal       -2.95949    2.17215  -1.362   0.1803
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.69 on 42 degrees of freedom
## Multiple R-squared:  0.5267, Adjusted R-squared:  0.4816
## F-statistic: 11.69 on 4 and 42 DF,  p-value: 1.815e-06
```

```
plot(fitted(model_t), residuals(model_t), xlab = "Fitted", ylab = "Residuals")
abline(h=0)
```



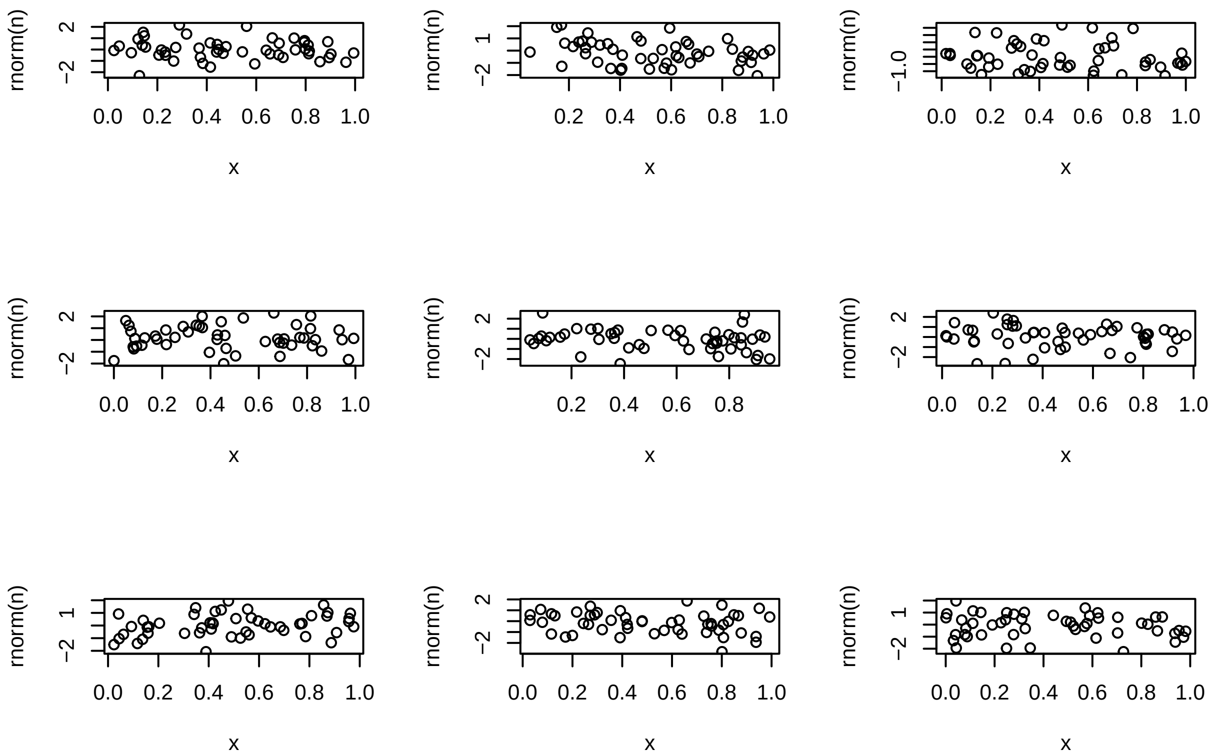

```
plot(fitted(model_t), sqrt(abs(residuals(model_t))), xlab = "Fitted", ylab = expression(sqrt(hat(epsilo
```



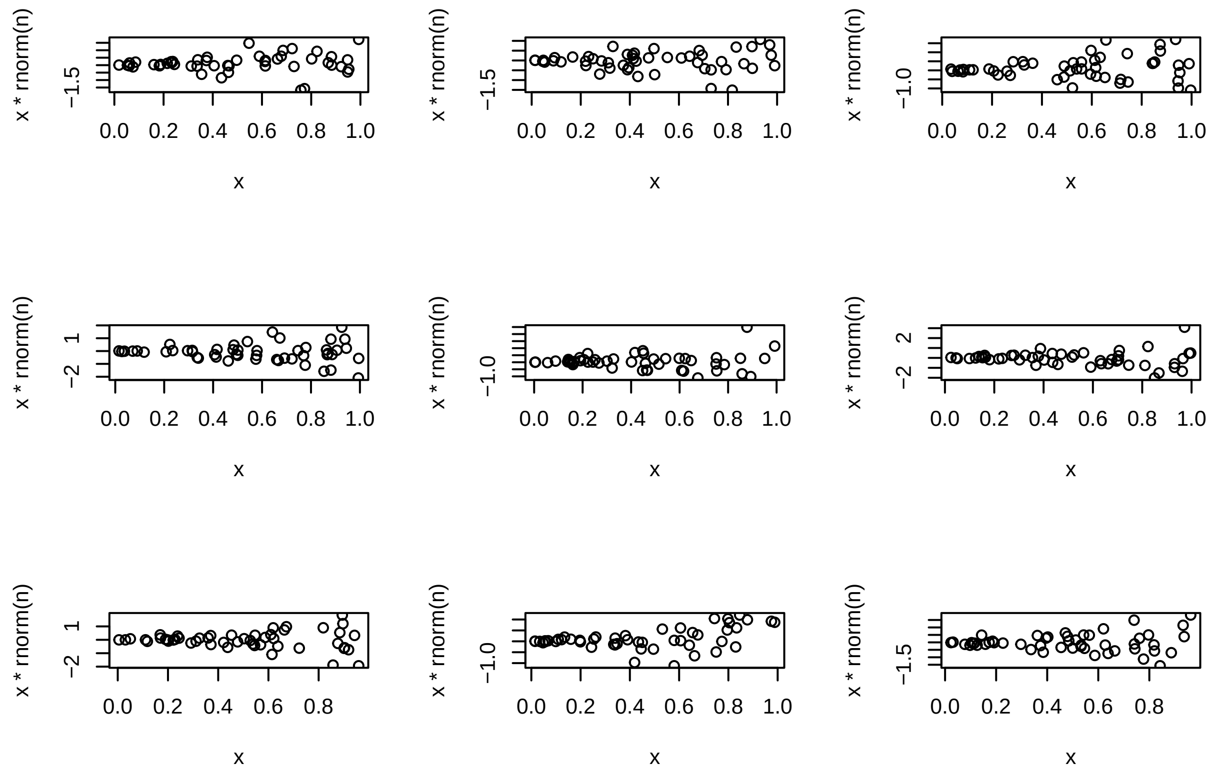

```
summary(lm(sqrt(abs(residuals(model_t))) ~ fitted(model_t)))
```

```
##
## Call:
## lm(formula = sqrt(abs(residuals(model_t))) ~ fitted(model_t))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.055 -1.206 -0.072  0.733  5.176
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.87838    0.31218   9.220 6.21e-12 ***
## fitted(model_t)  0.02679    0.01050   2.552   0.0142 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.628 on 45 degrees of freedom
## Multiple R-squared:  0.1265, Adjusted R-squared:  0.107
## F-statistic: 6.514 on 1 and 45 DF,  p-value: 0.01417
```

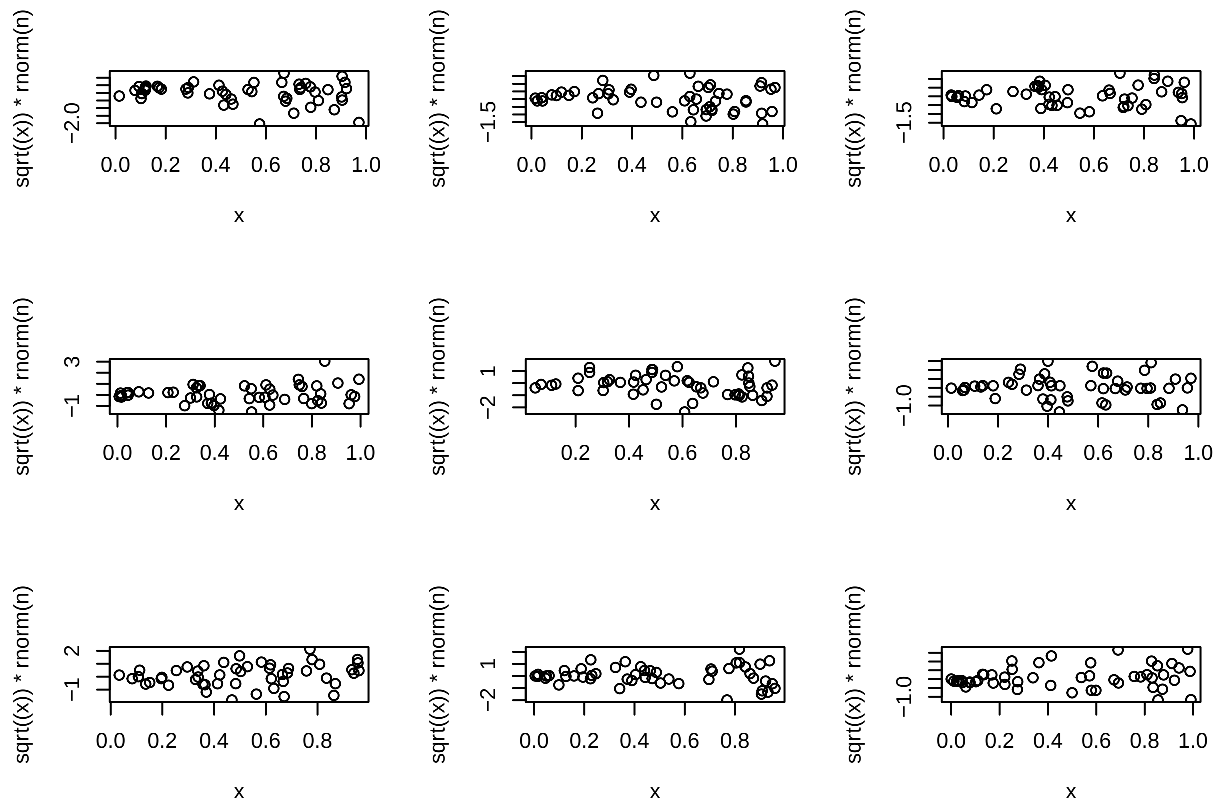
```
par(mfrow=c(3,3))
n <- 50

for(i in 1:9) {x <- runif(n) ; plot(x, rnorm(n))}
```



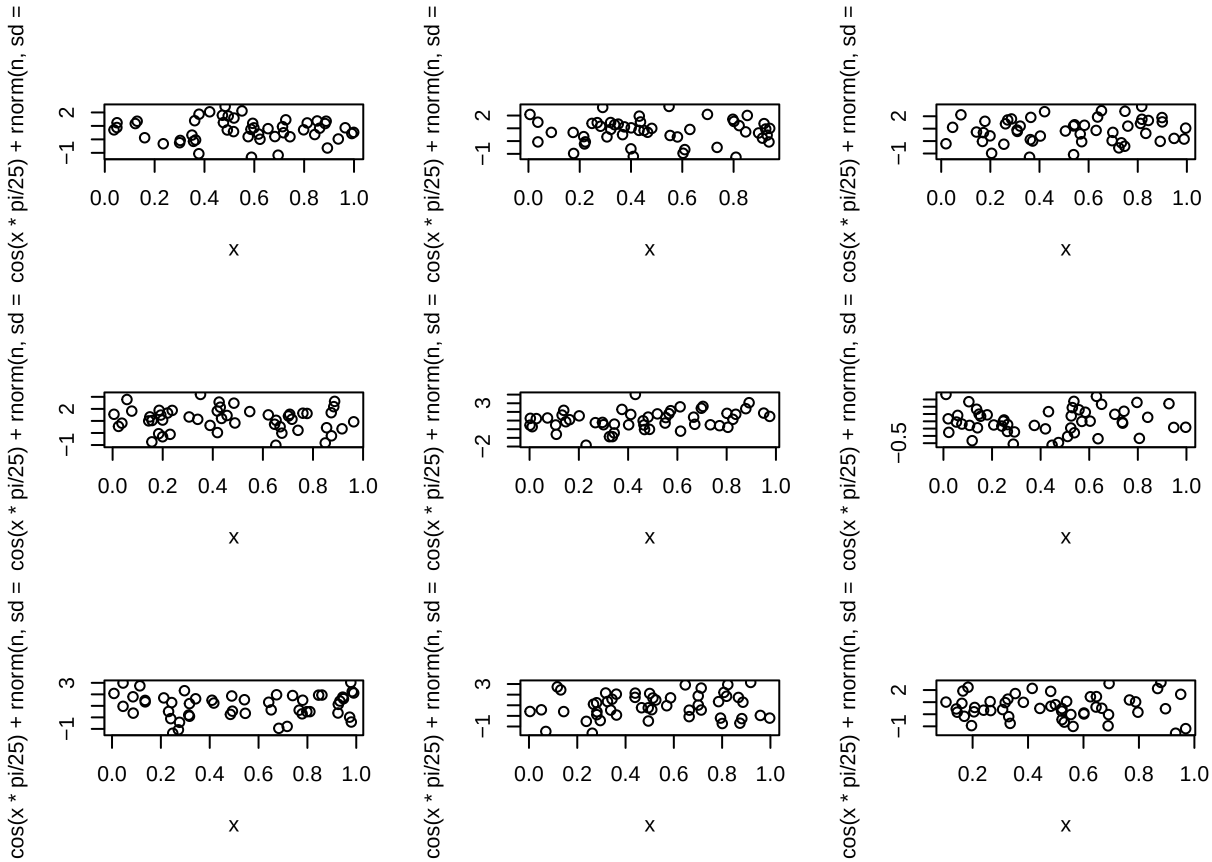

```r
for(i in 1:9) {x <- runif(n) ; plot(x, x*rnorm(n))}
```




```r
for(i in 1:9) {x <- runif(n) ; plot(x, sqrt ((x)) * rnorm(n))}
```



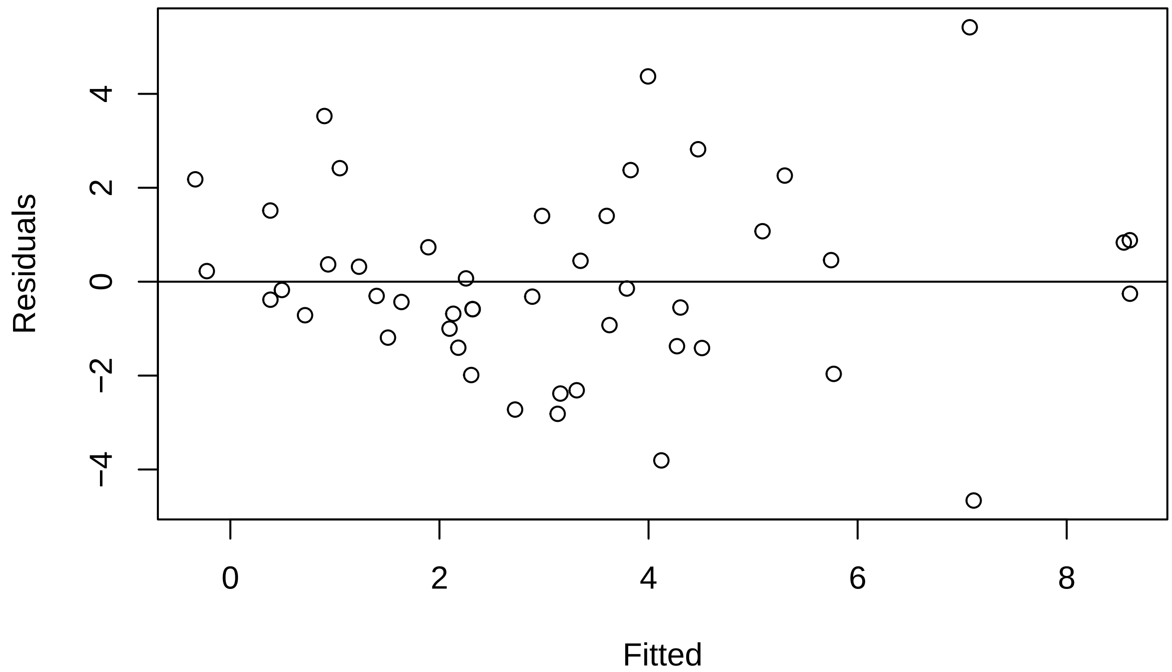

```r
for(i in 1:9) {x <- runif(n) ; plot(x, cos(x*pi/25)+rnorm(n, sd = 1))}
```



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

model_t_1 <- lm(sqrt(gamble) ~ ., data = teengamb)
summary(model_t_1)
```

```
##
## Call:
## lm(formula = sqrt(gamble) ~ ., data = teengamb)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6606 -1.0961 -0.2564  0.9786  5.4178
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.97707    1.57947   1.885  0.06638 .
## sex         -2.04450    0.75416  -2.711  0.00968 **
## status       0.03688    0.02582   1.428  0.16057
## income       0.47938    0.09418   5.090 7.94e-06 ***
## verbal      -0.42360    0.19950  -2.123  0.03967 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.084 on 42 degrees of freedom
## Multiple R-squared:  0.5646, Adjusted R-squared:  0.5231
## F-statistic: 13.61 on 4 and 42 DF,  p-value: 3.362e-07
```

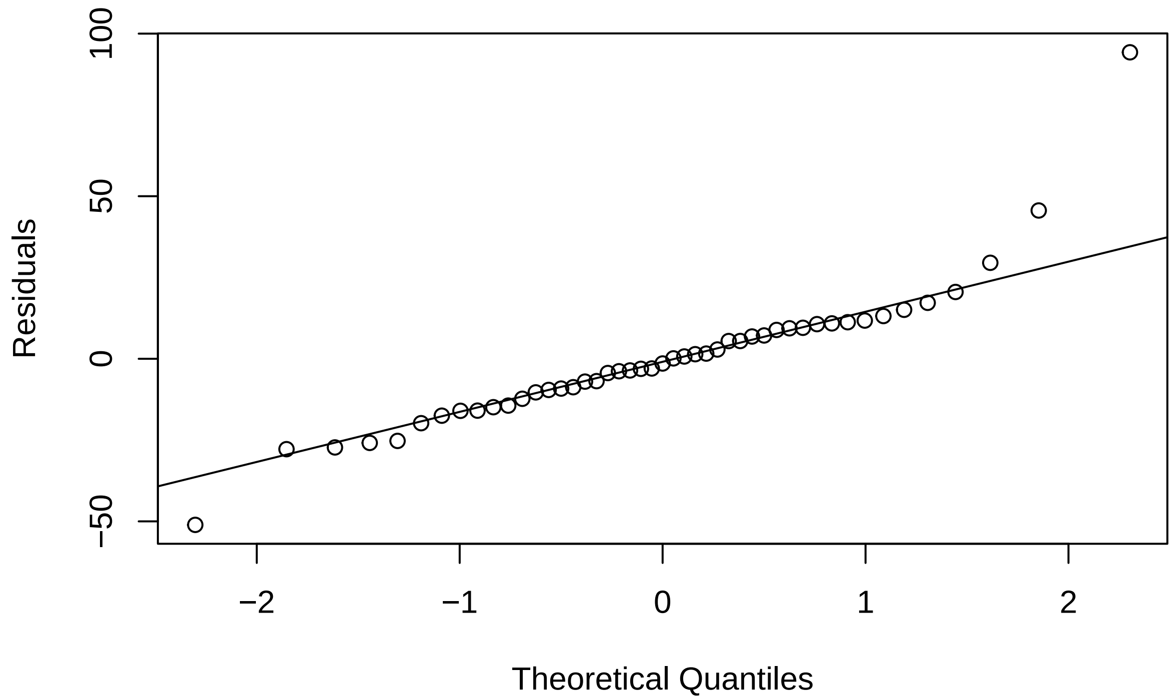
```r
plot(fitted(model_t_1), residuals(model_t_1), xlab = "Fitted", ylab = "Residuals")
abline(h=0)
```



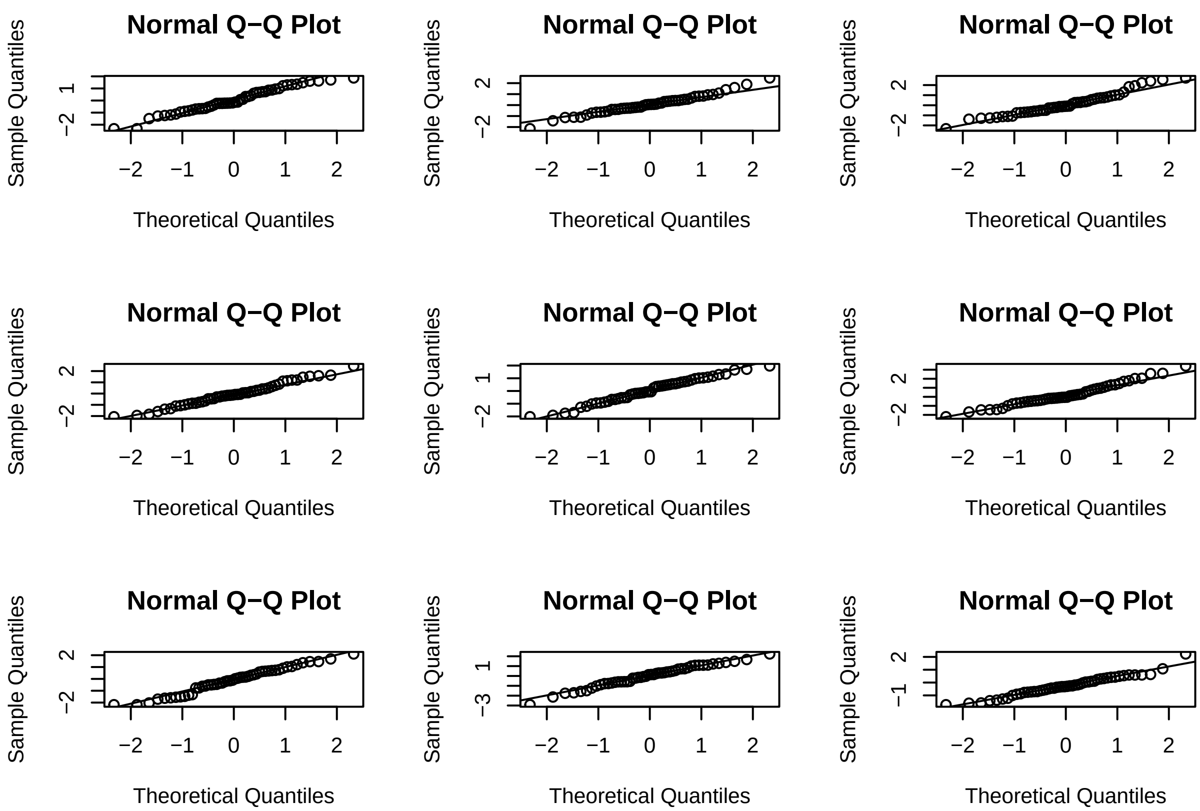
b

Check the normality assumption.

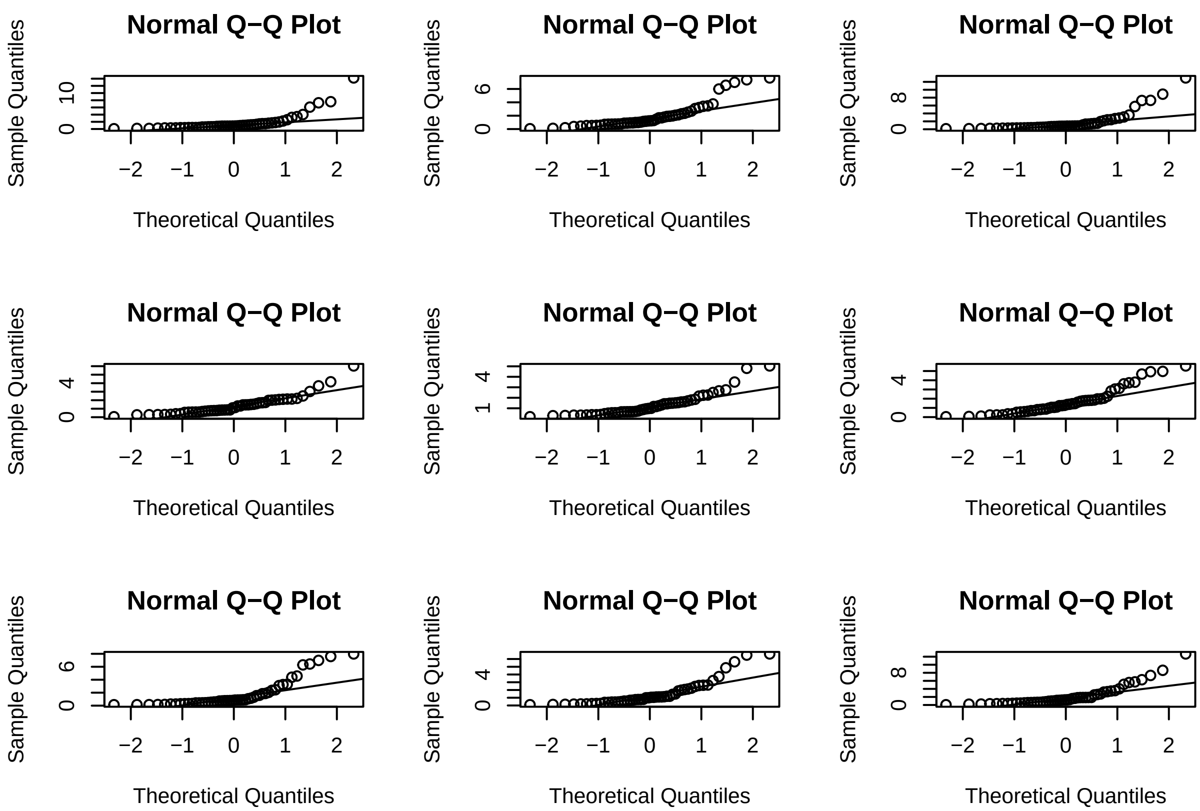
```r
qqnorm(residuals(model_t), ylab = "Residuals", main = "")
qqline(residuals(model_t))
```



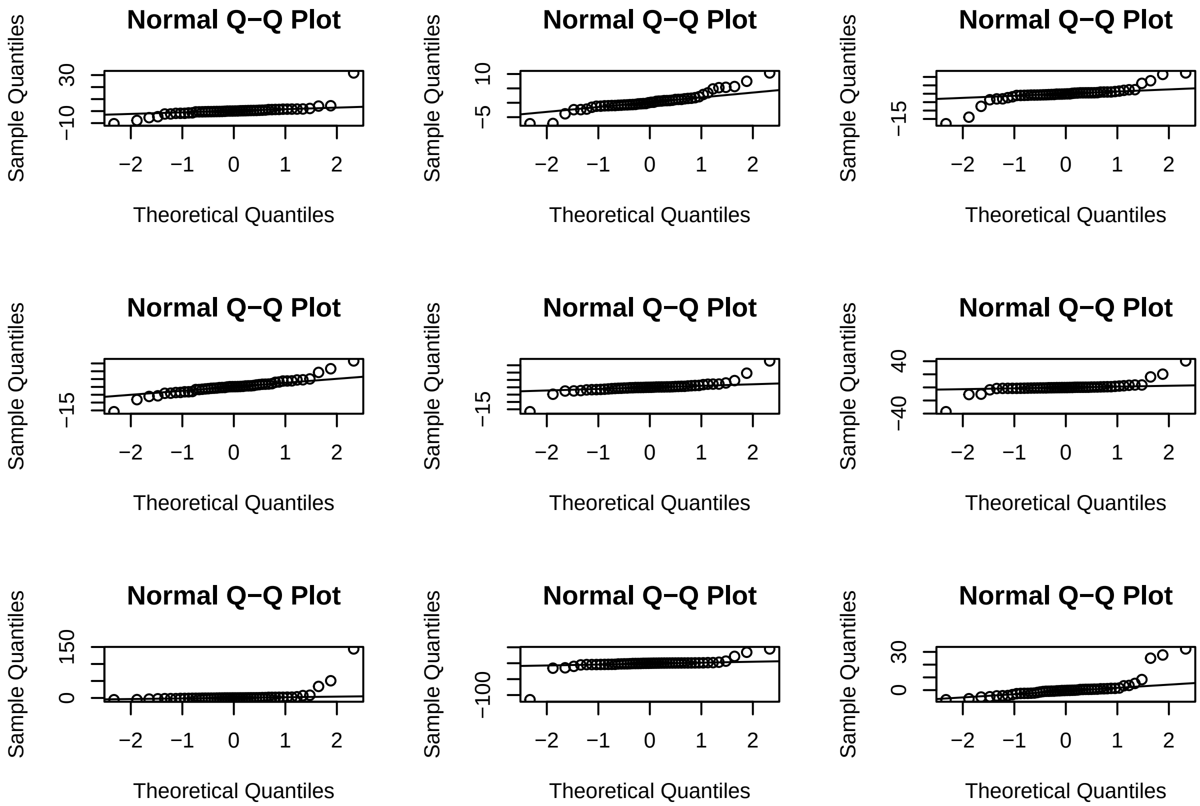

```r
par(mfrow=c(3,3))
n <- 50

for(i in 1:9) {x <- rnorm(n) ; qqnorm(x) ; qqline(x)}
```



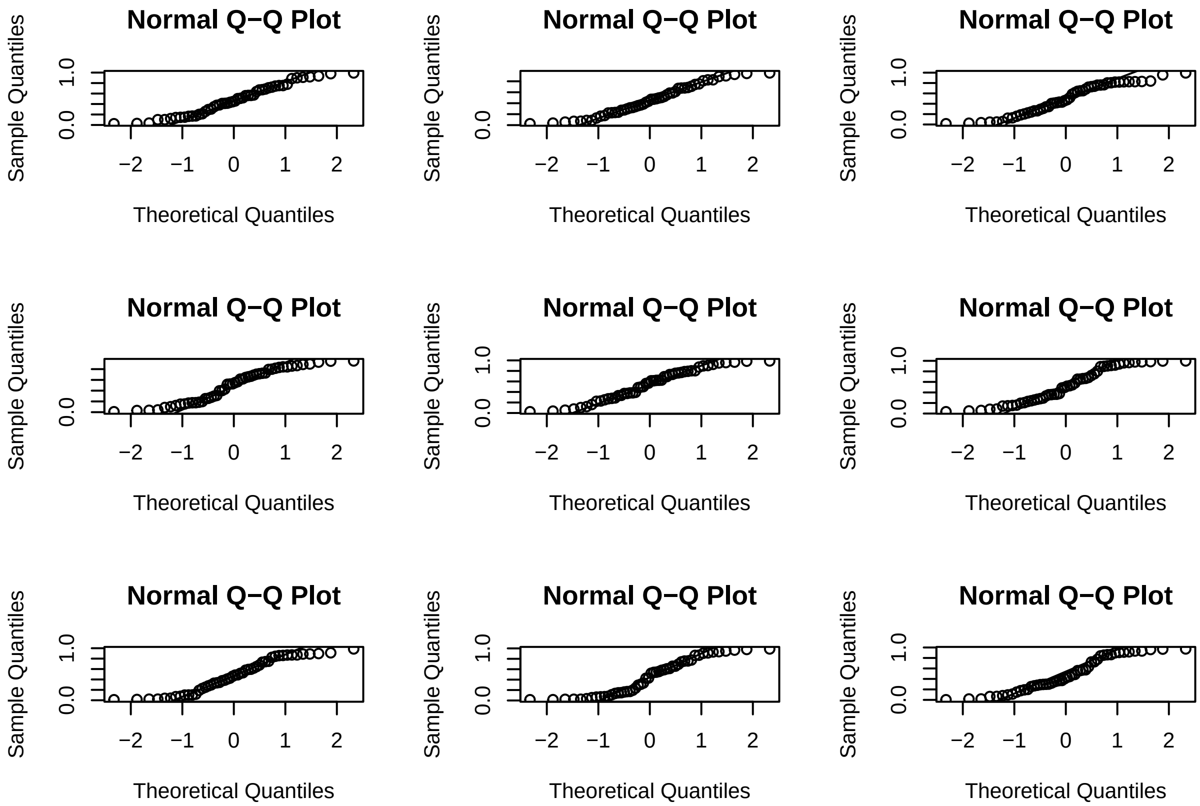

```r
for(i in 1:9) {x <- exp(rnorm(n)); qqnorm(x); qqline(x)}
```




```r
for(i in 1:9) {x <- rcauchy(n); qqnorm(x); qqline(x)}
```




```
for(i in 1:9) {x <- runif(n); qqnorm(x); qqline(x)}
```




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

shapiro.test(residuals(model_t))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_t)
## W = 0.86839, p-value = 8.16e-05
```

c

Check for large leverage points.

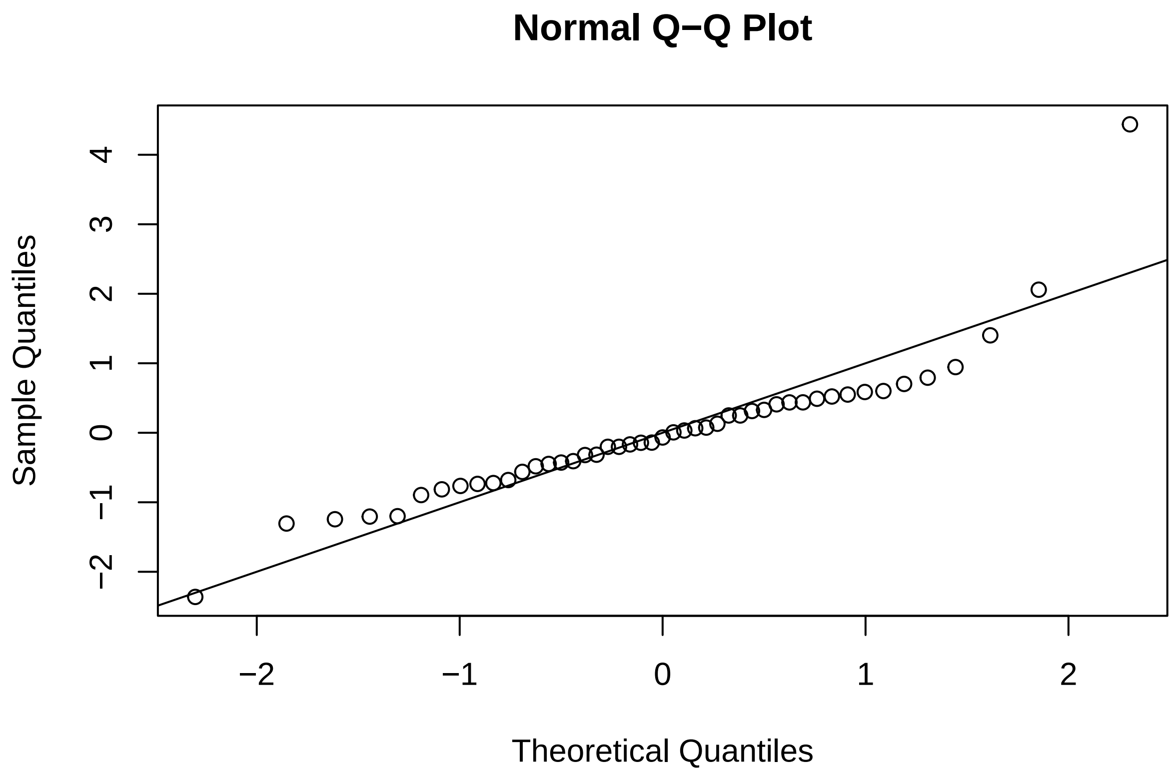
```r
hatv <- hatvalues(model_t)
head(hatv)
```

```
##          1          2          3          4          5          6
## 0.07988226 0.10851291 0.06347643 0.10273955 0.13866946 0.16378563
```

```r
sum(hatv)
```

```
## [1] 5
```

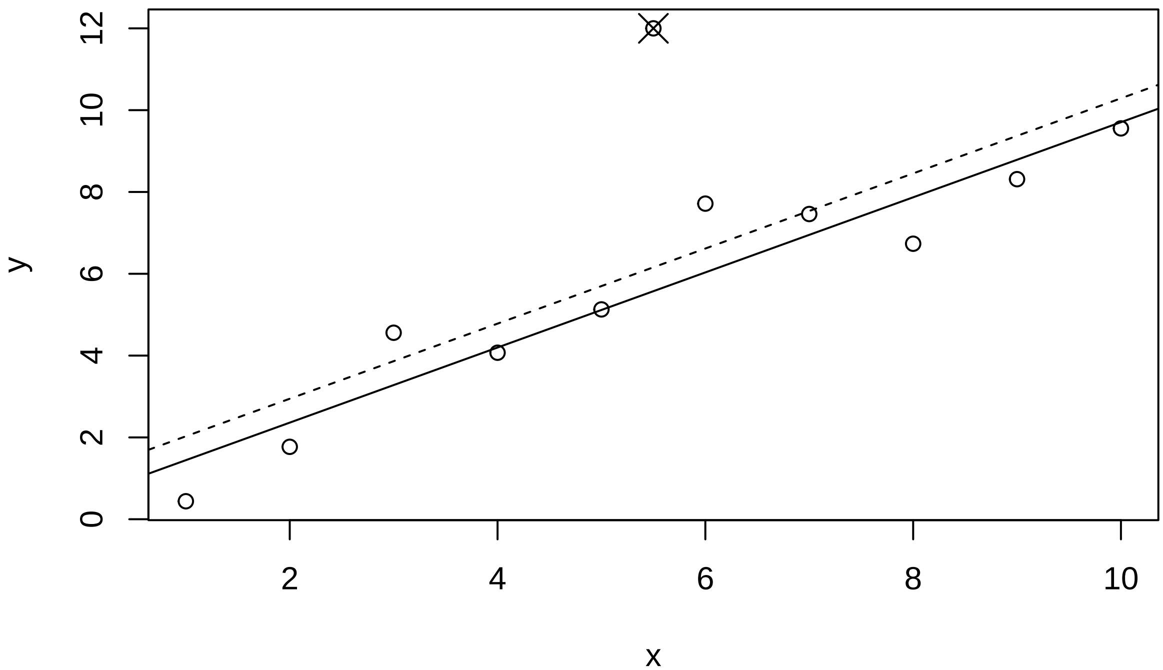
```r
#states <- row.names(gamble)
#halfnorm(hatv, labs = states, ylab = "Leverages")

qqnorm(rstandard(model_t))
abline(0,1)
```



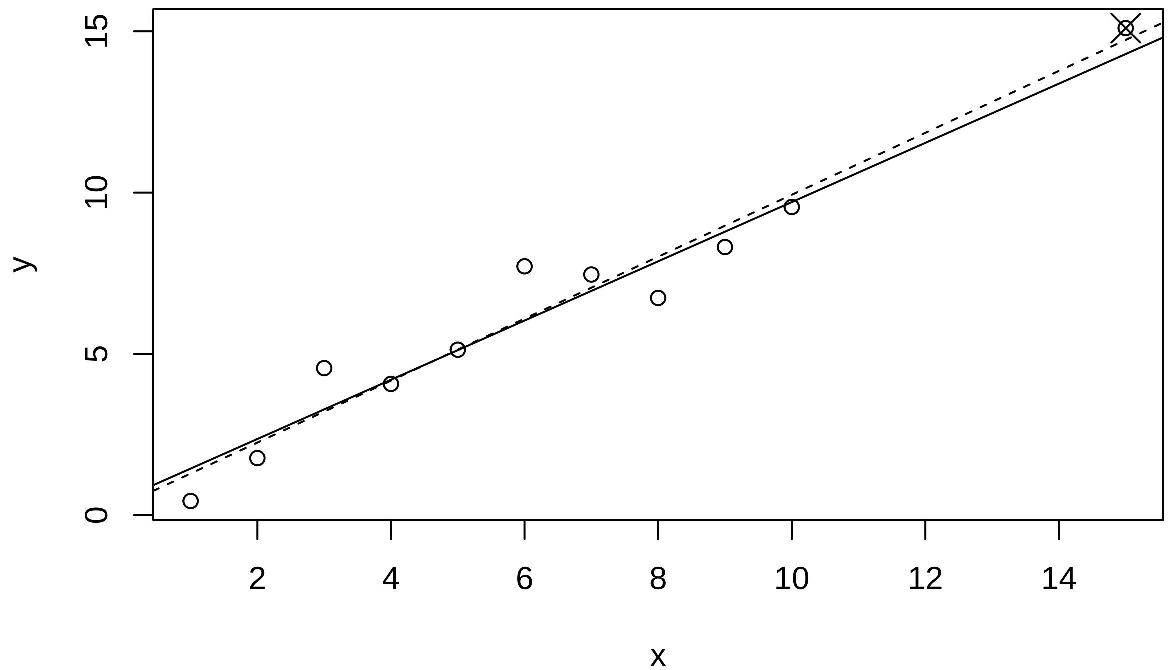
d

Check for outliers.

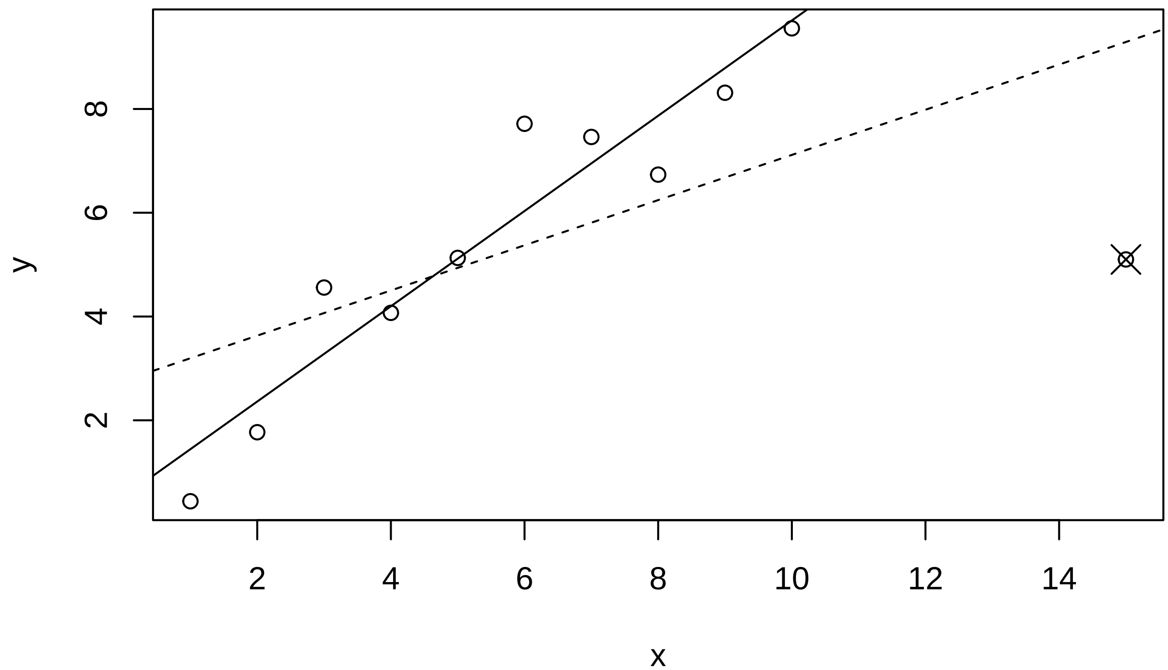
```r
set.seed(123)
testdata <- data.frame(x=1:10,y=1:10+rnorm(10))
model_t_2 <- lm(y ~ x, testdata)

p1 <- c(5.5,12)
model_t_3 <- lm(y ~ x, rbind(testdata, p1))
plot(y ~ x, rbind(testdata, p1))
points(5.5, 12, pch=4, cex=2)
abline(model_t_2)
abline(model_t_3, lty=2)
```




```
p2 <- c(15,15.1)
model_t_4 <- lm(y ~ x, rbind(testdata, p2))
plot(y ~ x, rbind(testdata, p2))
points(15, 15.1, pch=4, cex=2)
abline(model_t_2)
abline(model_t_4, lty=2)
```




```
p3 <- c(15,5.1)
model_t_5 <- lm(y ~ x, rbind(testdata, p3))
plot(y ~ x, rbind(testdata, p3))
points(15, 5.1, pch=4, cex=2)
abline(model_t_2)
abline(model_t_5, lty=2)
```




```
stud <- rstudent(model_t)
stud[which.max(abs(stud))]
```

```
##       24
## 6.016116
```

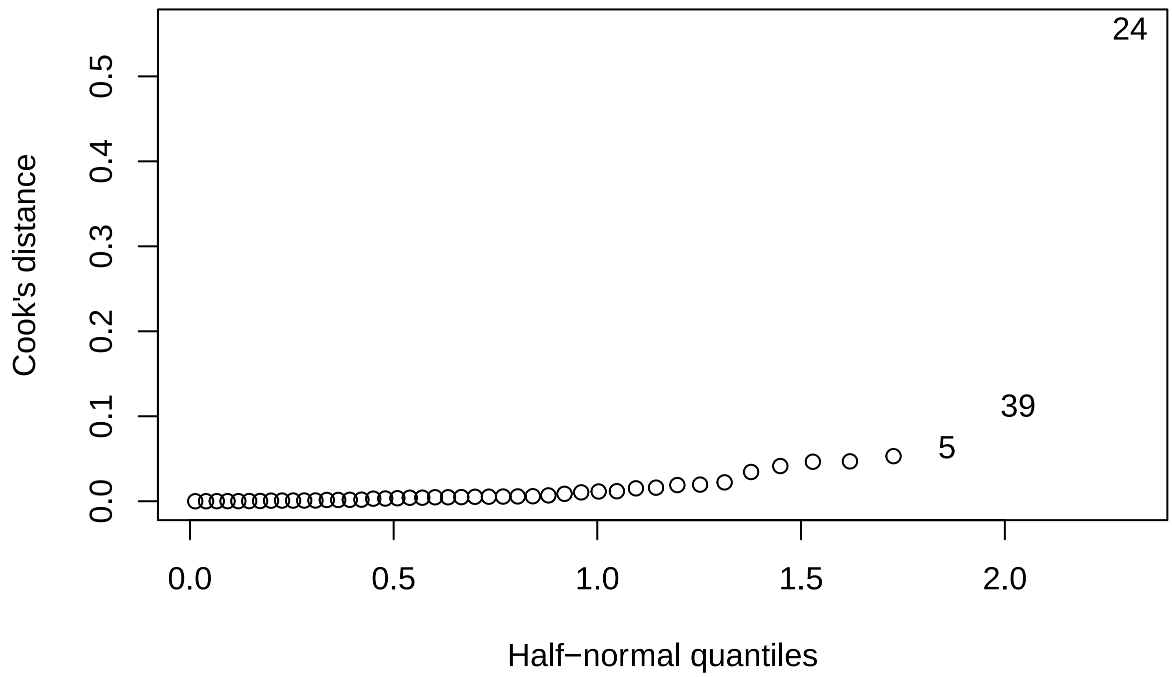
```
qt(0.05/(50*2),44)
```

```
## [1] -3.525801
```

e

Check for influential points.

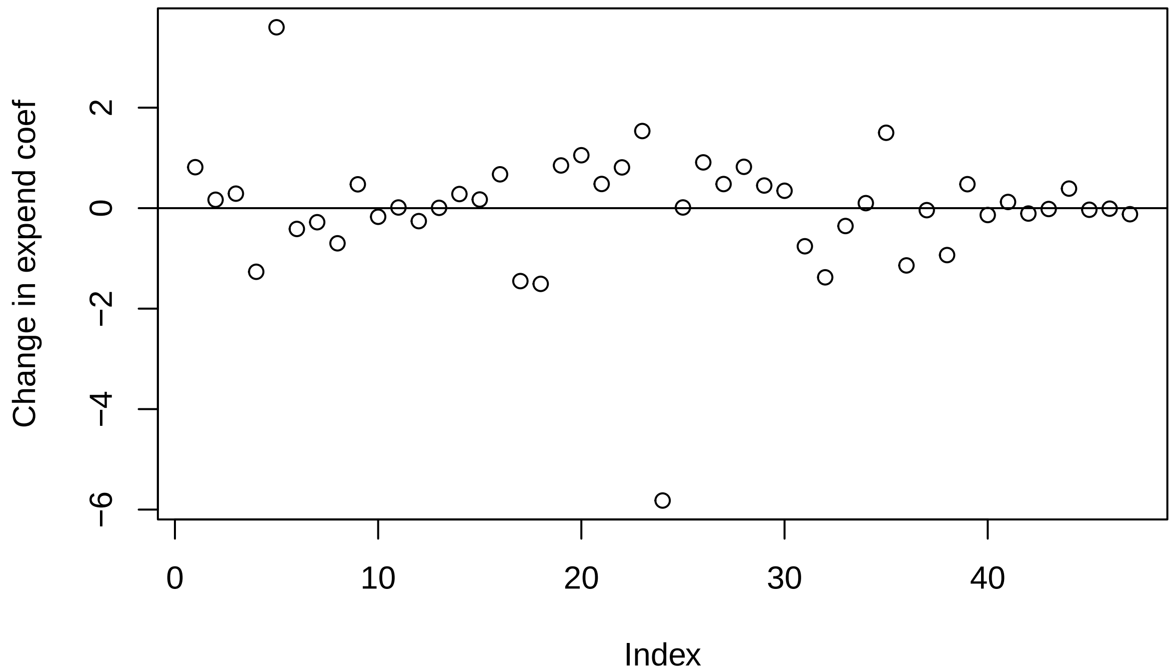
```
cook <- cooks.distance(model_t)
halfnorm(cook, 3, ylab = "Cook's distance")
```




```r
model_t_6 <- lm(gamble ~ ., data = teengamb, subset = (cook < max(cook)))
summary(model_t_6)
```

```
##
## Call:
## lm(formula = gamble ~ ., data = teengamb, subset = (cook < max(cook)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.999  -8.102  -0.491   8.600  46.688
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.6306    12.9251   0.590   0.5582
## sex         -16.2986     6.1335  -2.657   0.0112 *
## status        0.1739     0.2083   0.835   0.4088
## income        4.3312     0.7636   5.672 1.26e-06 ***
## verbal       -1.8019     1.6137  -1.117   0.2707
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.74 on 41 degrees of freedom
## Multiple R-squared:  0.5682, Adjusted R-squared:  0.526
## F-statistic: 13.49 on 4 and 41 DF,  p-value: 4.225e-07
```

```
plot(dfbeta(model_t)[,2],ylab = "Change in expend coef")
abline(h=0)
```



f

Check the structure of the relationship between the predictors and the response.

Question Three

Using the divusa data:

a

Fit a regression model with divorce as the response and unemployed, femlab, marriage, birth and military as predictors. Compute the condition numbers and interpret their meanings.

```
model_d <- lm(divorce ~ unemployed + femlab + marriage + birth + military, data = divusa)

sumary(model_d)
```

```
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  2.487845   3.393779  0.7331   0.46594
```

```
## unemployed  -0.111252   0.055925 -1.9893   0.05052
## femlab       0.383649   0.030587 12.5430 < 2.2e-16
## marriage     0.118674   0.024414  4.8609 6.772e-06
## birth       -0.129959   0.015595 -8.3334 4.027e-12
## military    -0.026734   0.014247 -1.8764   0.06471
##
## n = 77, p = 6, Residual SE = 1.65042, R-Squared = 0.92
```

```
summary(model_d)
```

```
##
## Call:
## lm(formula = divorce ~ unemployed + femlab + marriage + birth +
##     military, data = divusa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8611 -0.8916 -0.0496  0.8650  3.8300
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.48784    3.39378   0.733   0.4659
## unemployed  -0.11125    0.05592  -1.989   0.0505 .
## femlab       0.38365    0.03059  12.543  < 2e-16 ***
## marriage     0.11867    0.02441   4.861 6.77e-06 ***
## birth       -0.12996    0.01560  -8.333 4.03e-12 ***
## military    -0.02673    0.01425  -1.876   0.0647 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.65 on 71 degrees of freedom
## Multiple R-squared:  0.9208, Adjusted R-squared:  0.9152
## F-statistic: 165.1 on 5 and 71 DF,  p-value: < 2.2e-16
```

```
print("")
```

```
## [1] ""
```

b

For the same model, compute the VIFs. Is there evidence that collinearity causes some predictors not to be significant? Explain.

c

Does the removal of insignificant predictors from the model reduce the collinearity? Investigate.

Question Four

Use the fat data, fitting the model described in Section 4.2.

a

Compute the condition numbers and variance inflation factors. Comment on the degree of collinearity observed in the data.

```r
#view(fat)
model_f <- lm(brozek ~ age + weight + height + neck + chest + abdom + hip + thigh + knee + ankle + bicep
summary(model_f)
```

```
##
## Call:
## lm(formula = brozek ~ age + weight + height + neck + chest +
##     abdom + hip + thigh + knee + ankle + biceps + forearm + wrist,
##     data = fat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.264  -2.572  -0.097   2.898   9.327
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -15.29255   16.06992  -0.952  0.34225
## age           0.05679    0.02996   1.895  0.05929 .
## weight       -0.08031    0.04958  -1.620  0.10660
## height       -0.06460    0.08893  -0.726  0.46830
## neck         -0.43754    0.21533  -2.032  0.04327 *
## chest        -0.02360    0.09184  -0.257  0.79740
## abdom         0.88543    0.08008  11.057  < 2e-16 ***
## hip          -0.19842    0.13516  -1.468  0.14341
## thigh         0.23190    0.13372   1.734  0.08418 .
## knee         -0.01168    0.22414  -0.052  0.95850
## ankle         0.16354    0.20514   0.797  0.42614
## biceps        0.15280    0.15851   0.964  0.33605
## forearm       0.43049    0.18445   2.334  0.02044 *
## wrist        -1.47654    0.49552  -2.980  0.00318 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.988 on 238 degrees of freedom
## Multiple R-squared:  0.749,  Adjusted R-squared:  0.7353
## F-statistic: 54.63 on 13 and 238 DF,  p-value: < 2.2e-16
```

```r
x <- model.matrix(model_f)[,-1]
e <- eigen(t(x) %*% x)
e$val
```

```
##  [1] 1.959256e+07 6.418499e+04 3.059739e+04 5.704341e+03 2.803947e+03
##  [6] 1.934715e+03 1.030340e+03 6.376692e+02 5.280964e+02 4.318186e+02
## [11] 3.763758e+02 2.723663e+02 6.345357e+01
```

```r
sqrt(e$val[1]/e$val)
```

```
##  [1]   1.00000  17.47144  25.30482  58.60610  83.59121 100.63222 137.89717
##  [8] 175.28623 192.61449 213.00748 228.15747 268.20620 555.67072
```

```r
vif(x)
```

```
##       age    weight    height      neck     chest     abdom       hip     thigh
##  2.250450 33.509320  1.674591  4.324463  9.460877 11.767073 14.796520  7.777865
##      knee     ankle    biceps   forearm     wrist
##  4.612147  1.907961  3.619744  2.192492  3.377515
```

b

Cases 39 and 42 are unusual. Refit the model without these two cases and recompute the collinearity diagnostics. Comment on the differences observed from the full data fit.

```r
fat_1 <- fat[-c(39,42),]
model_f_1 <- lm(brozek ~ age + weight + height + neck + chest + abdom + hip + thigh + knee + ankle + bic
summary(model_f_1)
```

```
##
## Call:
## lm(formula = brozek ~ age + weight + height + neck + chest +
##     abdom + hip + thigh + knee + ankle + biceps + forearm + wrist,
##     data = fat_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0975  -2.8719  -0.2185   2.7420   9.2030
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.62251   21.65491   0.121  0.90371
## age          0.06583    0.02981   2.209  0.02817 *
## weight      -0.01596    0.06220  -0.257  0.79771
## height      -0.22355    0.17708  -1.262  0.20806
## neck        -0.35926    0.21759  -1.651  0.10004
## chest       -0.11059    0.10029  -1.103  0.27127
## abdom        0.83988    0.08468   9.919  < 2e-16 ***
## hip         -0.15313    0.13513  -1.133  0.25830
## thigh        0.17447    0.13603   1.283  0.20090
## knee        -0.06941    0.22754  -0.305  0.76060
## ankle        0.17426    0.20368   0.856  0.39310
## biceps       0.15160    0.15786   0.960  0.33785
## forearm      0.26827    0.19170   1.399  0.16300
## wrist       -1.64288    0.49384  -3.327  0.00102 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.942 on 236 degrees of freedom
## Multiple R-squared:  0.7503, Adjusted R-squared:  0.7366
## F-statistic: 54.56 on 13 and 236 DF,  p-value: < 2.2e-16
```

```r
x_1 <- model.matrix(model_f_1)[,-1]
e_1 <- eigen(t(x_1) %*% x_1)
e_1$val
```

```
##  [1] 1.929452e+07 5.700314e+04 2.807492e+04 5.095990e+03 2.326071e+03
##  [6] 1.473052e+03 8.722982e+02 6.034713e+02 4.724511e+02 4.301792e+02
## [11] 3.330435e+02 2.526472e+02 6.268514e+01
```

```r
sqrt(e_1$val[1] / e_1$val)
```

```
##  [1]   1.00000  18.39787  26.21547  61.53224  91.07633 114.44792 148.72518
##  [8] 178.80871 202.08708 211.78359 240.69468 276.35018 554.79777
```

```r
vif(x_1)
```

```
##       age    weight    height      neck     chest     abdom       hip     thigh
##  2.278191 45.298843  3.439587  3.978898 10.712505 11.967580 12.146249  7.153711
##      knee     ankle    biceps   forearm     wrist
##  4.441752  1.810253  3.409524  2.422878  3.263677
```

c

Fit a model with brozek as the response and just age, weight and height as predictors. Compute the collinearity diagnostics and compare to the full data fit

```r
model_f_2 <- lm(brozek ~ age + weight + height, fat)
summary(model_f_2)
```

```
##
## Call:
## lm(formula = brozek ~ age + weight + height, data = fat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.0023  -4.1099  -0.0371   3.4873  14.4576
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.72142    6.92955   2.557   0.0111 *
## age          0.15583    0.02739   5.690 3.57e-08 ***
## weight       0.18373    0.01216  15.107  < 2e-16 ***
## height      -0.55099    0.09904  -5.563 6.85e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.382 on 248 degrees of freedom
## Multiple R-squared:  0.5236, Adjusted R-squared:  0.5179
## F-statistic: 90.87 on 3 and 248 DF,  p-value: < 2.2e-16
```

```r
x_2 <- model.matrix(model_f_2)[,-1]
e_2 <- eigen(t(x_2) %*% x_2)
e_2$val
```

```
## [1] 10001051.85    54778.52    19455.70
```

```r
sqrt(e_2$val[1] / e_2$val)
```

```
## [1]  1.00000 13.51194 22.67250
```

```r
vif(x_2)
```

```
##      age   weight   height
## 1.032253 1.107050 1.140470
```

d

Compute a 95% prediction interval for brozek for the median values of age, weight and height

```r
x <- model.matrix(model_f_2)
median <- apply(x, 2, median)
median
```

```
## (Intercept)         age      weight      height
##         1.0        43.0       176.5        70.0
```

```r
pred_d <- predict(model_f_2, data.frame(t(median)), interval="prediction")
pred_d
```

```
##        fit      lwr      upr
## 1 18.28132 7.659609 28.90304
```

```r
pred_d_width <- pred_d[3]-pred_d[2]
pred_d_width
```

```
## [1] 21.24343
```

e

Compute a 95% prediction interval for brozek for age=40, weight=200 and height=73. How does the interval compare to the previous prediction?

```r
new_data <- data.frame(age = 40, weight = 200, height = 73)
pred_e <- predict(model_f_2, data.frame(new_data), interval = "prediction")
pred_e
```

```
##        fit      lwr      upr
## 1 20.47854 9.837784 31.11929
```

f

Compute a 95% prediction interval for brozek for age=40, weight=130 and height=73. Are the values of predictors unusual? Comment on how the interval compares to the previous two answers.

```
new_data_1 <- data.frame(age = 40, weight = 130, height = 73)
pred_f <- predict(model_f_2, data.frame(new_data_1), interval = "prediction")
pred_f
```

```
##        fit       lwr     upr
## 1 7.617419 -3.101062 18.3359
```