NCSU ST 503 Discussion 7

Probems 6.2,6.3,6.4,6.5 Parts c,d,e,f Faraway, Julian J. Linear Models with R CRC Press.

Bruce Campbell

6.2 Using the teengamb dataset, fit a model with gamble as the response and the other variables as predictors.

(c) Check for large leverage points.

```
rm(list = ls())
data(teengamb, package = "faraway")
```

```
df <- teengamb
numPredictors <- (ncol(df) - 1)
lm.fit <- lm(gamble ~ ., data = df)
hatv <- hatvalues(lm.fit)
lev.cut <- (numPredictors + 1) * 2 * 1/nrow(df)
high.leverage <- df[hatv > lev.cut, ]
pander(high.leverage, caption = "High Leverage Data Elements")
```

Table 1: High Leverage Data Elements

	sex	status	income	verbal	gamble
31	0	18	12	2	88
33	0	38	15	7	90
35	0	28	1.5	1	14.1
42	0	61	15	9	69.7

We've used the rule of thumb that points with a leverage greater than $\frac{2p}{n}$ should be looked at.

(d) Check for outliers.

```
studentized.residuals <- rstudent(lm.fit)
max.residual <- studentized.residuals[which.max(abs(studentized.residuals))]
range.residuals <- range(studentized.residuals)
```

```
names(range.residuals) <- c("left", "right")
pander(data.frame(range.residuals = t(range.residuals)), caption = "Range of Studentized
```

Table 2: Range of Studentized residuals

range.residuals.left	range.residuals.right
-2.506	6.016

```
p <- numPredictors + 1
n <- nrow(df)
t.val.alpha <- qt(0.05/(n * 2), n - p - 1)
pander(data.frame(t.val.alpha = t.val.alpha), caption = "Bonferroni corrected t-value")
```

Table 3: Bonferroni corrected t-value

t.val.alpha
-3.523

```
outlier.index <- abs(studentized.residuals) > abs(t.val.alpha)

outliers <- df[outlier.index == TRUE, ]

if (nrow(outliers) >= 1) {
    pander(outliers, caption = "outliers")
}
```

Table 4: outliers

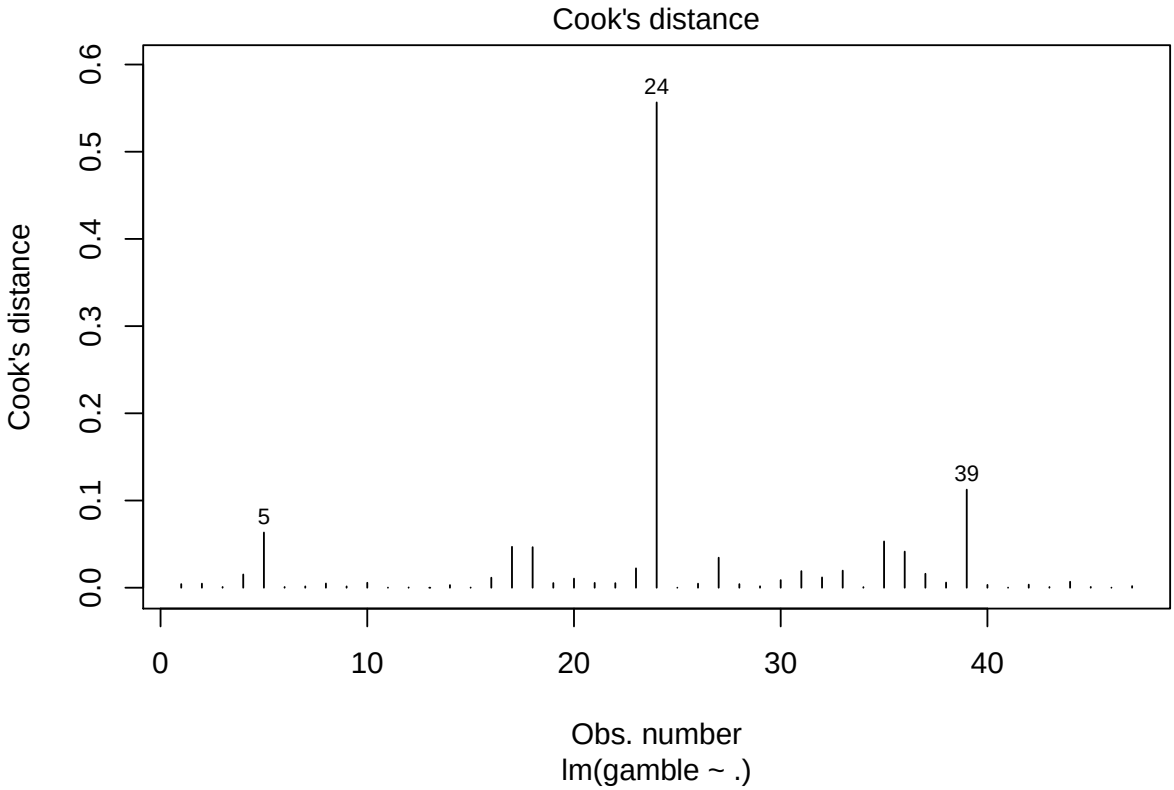
	sex	status	income	verbal	gamble
24	0	27	10	4	156

Here we look for studentized residuals that fall outside the interval given by the Bonferroni corrected t-values.

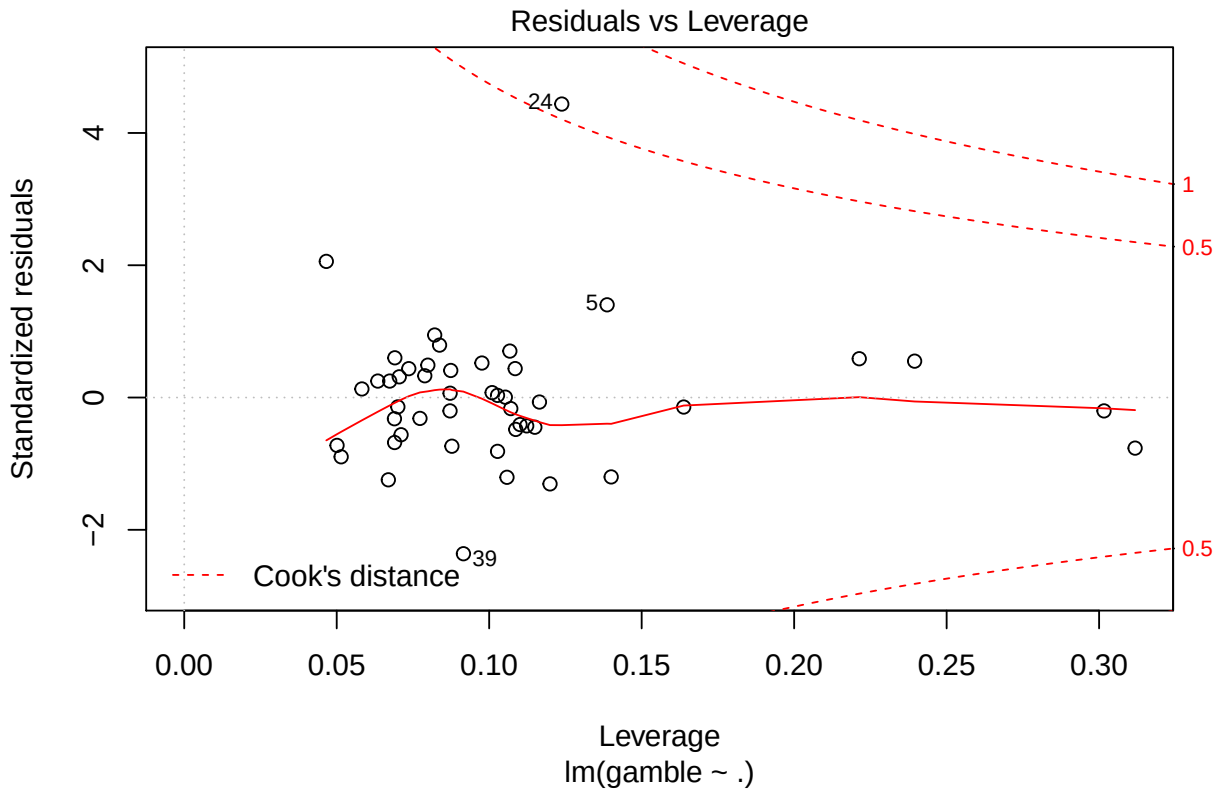
(e) Check for influential points.

We plot the Cook's distances and the residual-leverage plot with level set contours of the Cook distance.

```
plot(lm.fit, which = 4)
```



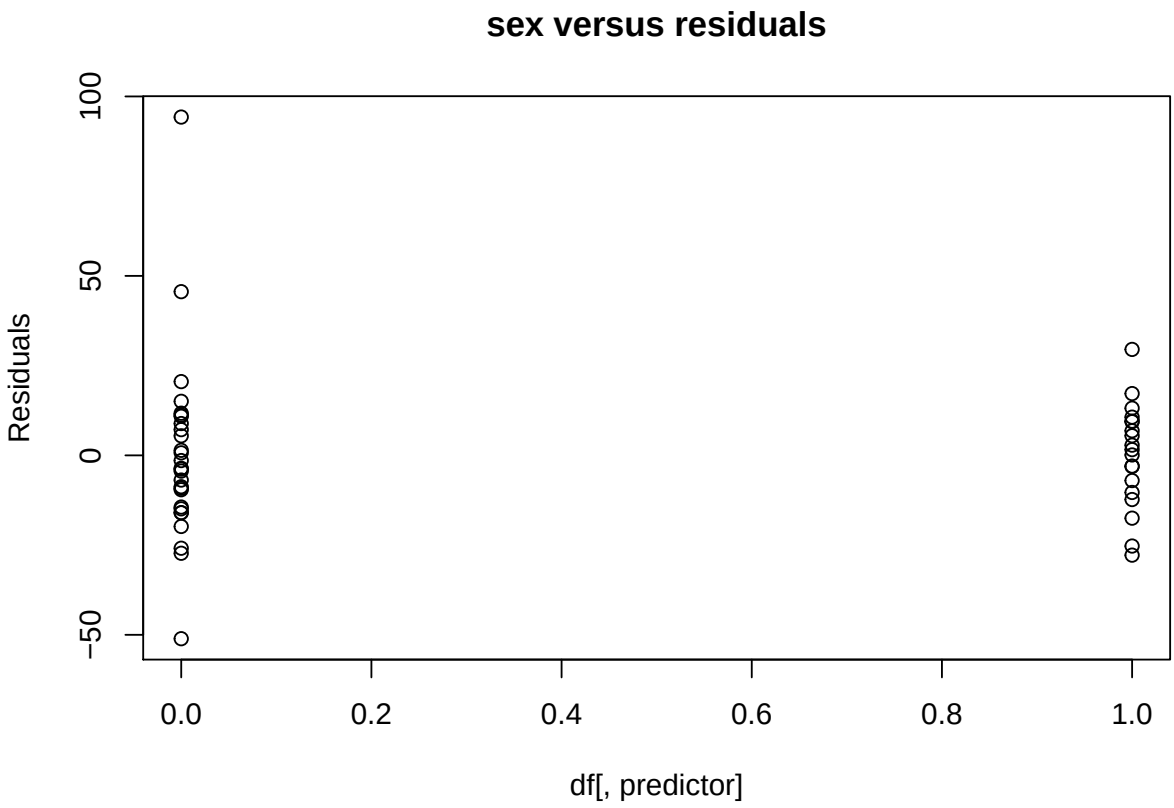

```
plot(lm.fit, which = 5)
```

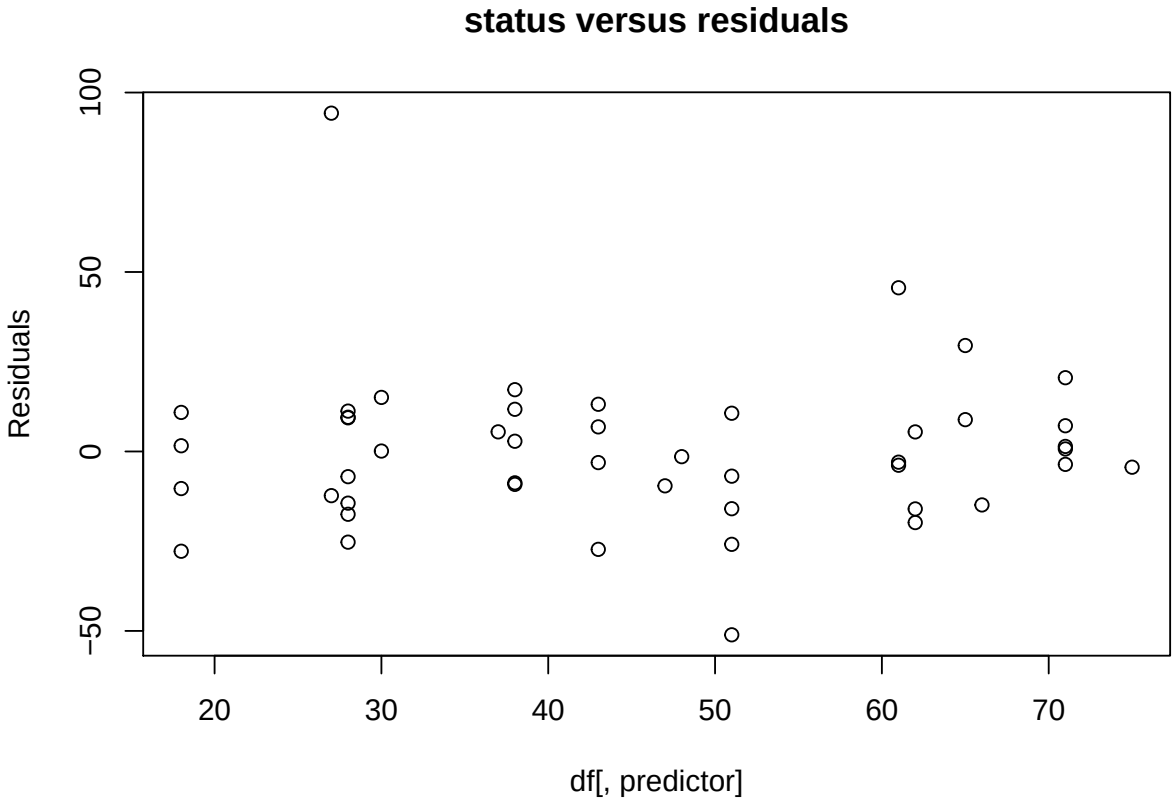


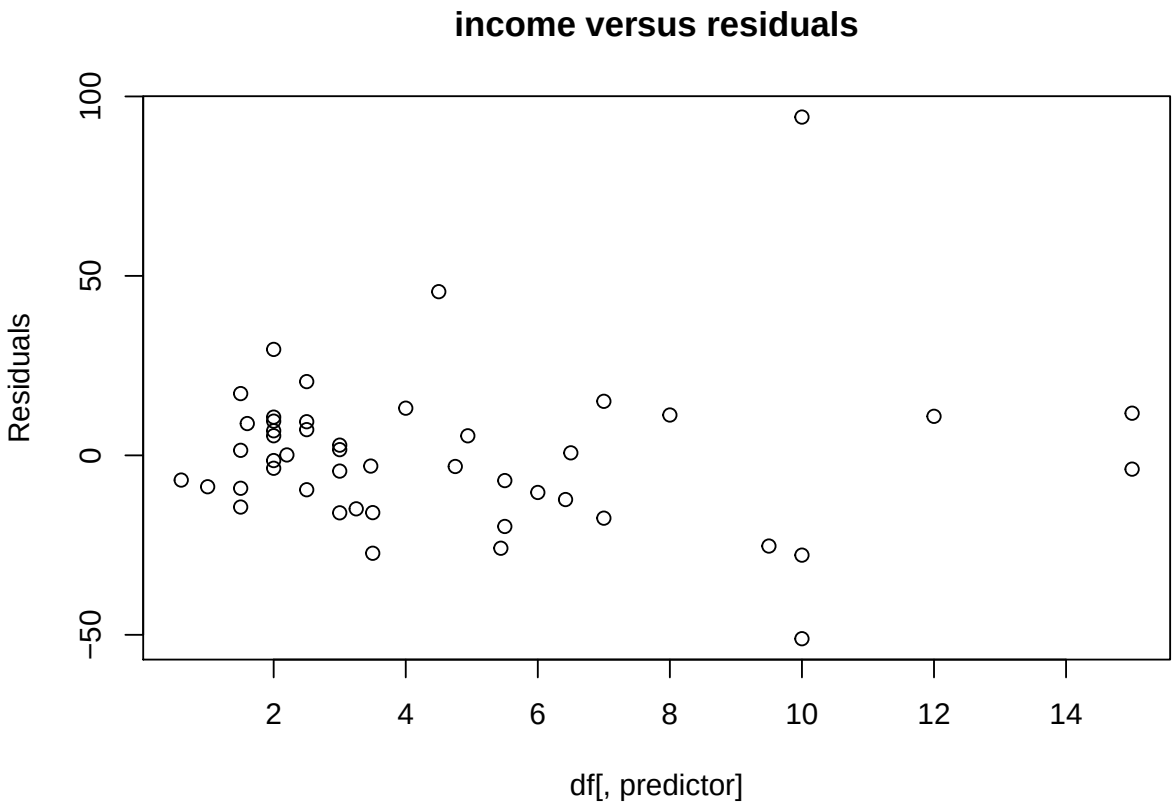
(f) Check for structure in the model.

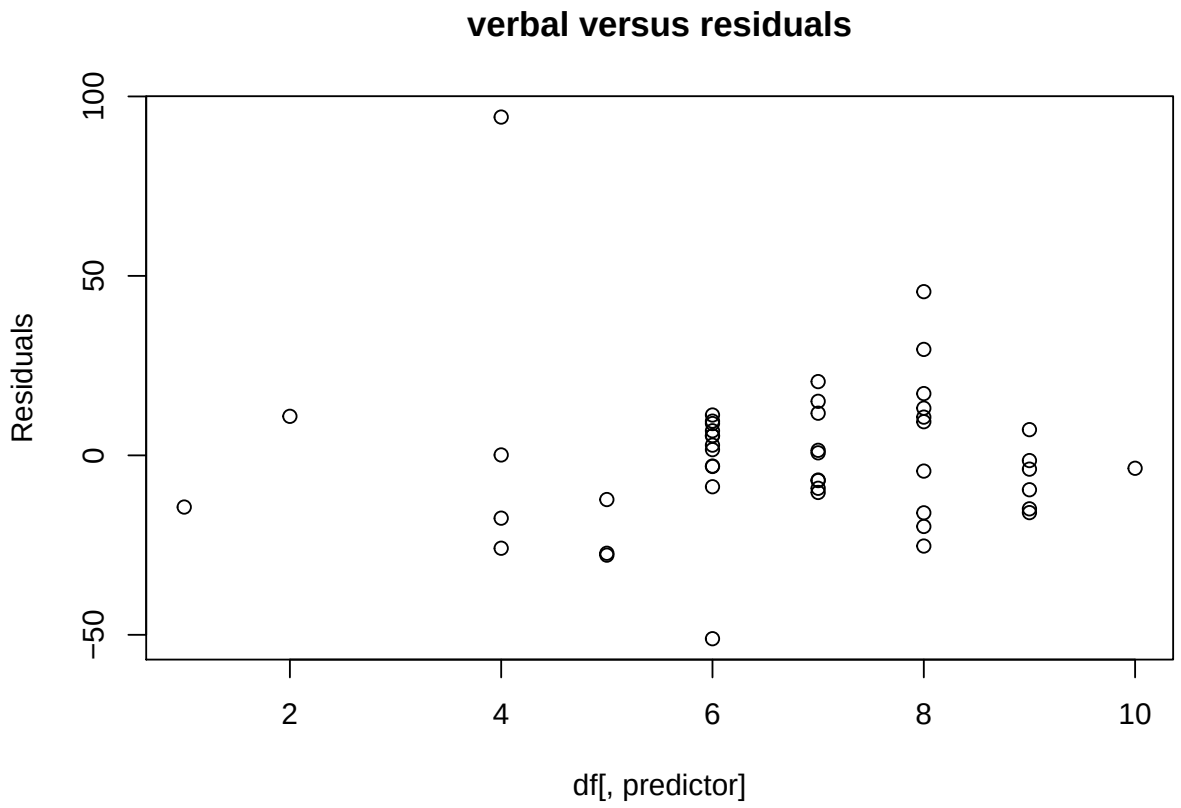
Plot residuals versus predictors

```
predictors <- names(lm.fit$coefficients)
predictors <- predictors[2:length(predictors)]

for (i in 1:length(predictors)) {
    predictor <- predictors[i]

    plot(df[, predictor], residuals(lm.fit), xlab = , ylab = "Residuals", main = paste(p
        " versus residuals", sep = ""))

}
```


sex versus residuals
Residuals
100
50
0
−50
0.0
0.2
0.4
0.6
0.8
1.0
df[, predictor]


status versus residuals
Residuals
100
50
0
−50
20
30
40
50
60
70
df[, predictor]

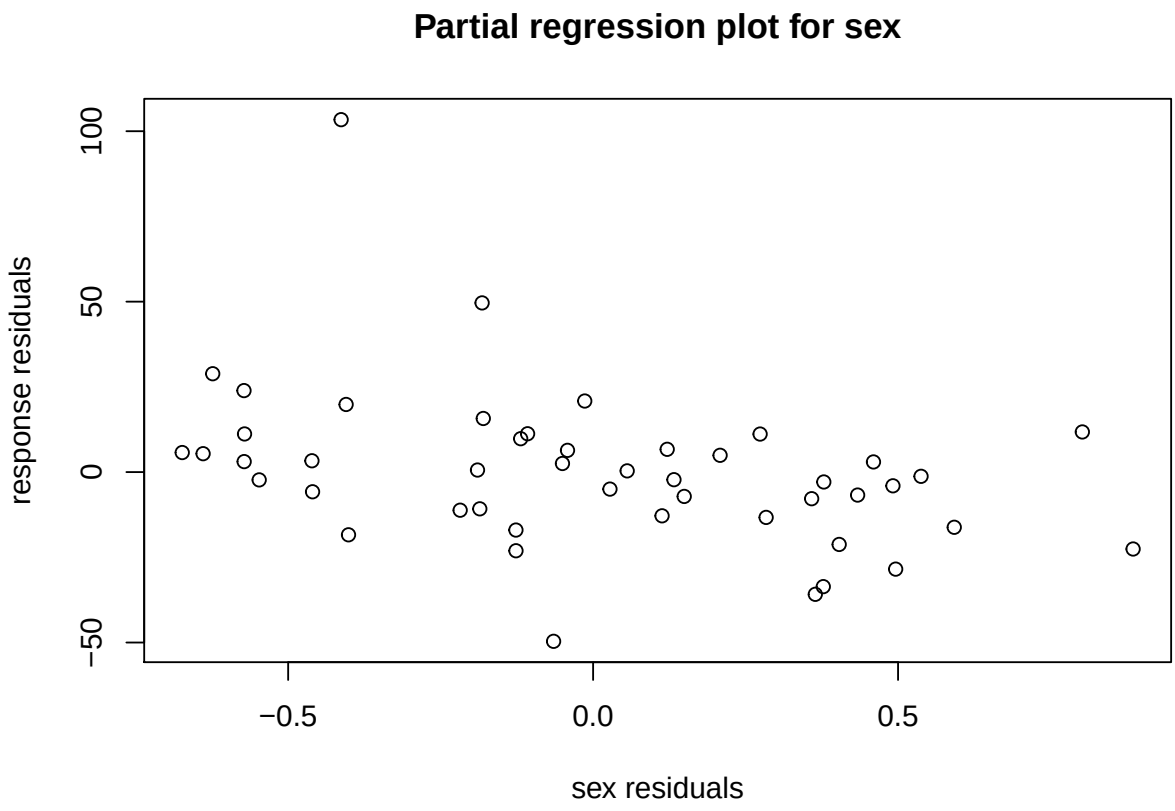

income versus residuals
Residuals
100
50
0
−50
2
4
6
8
10
12
14
df[, predictor]

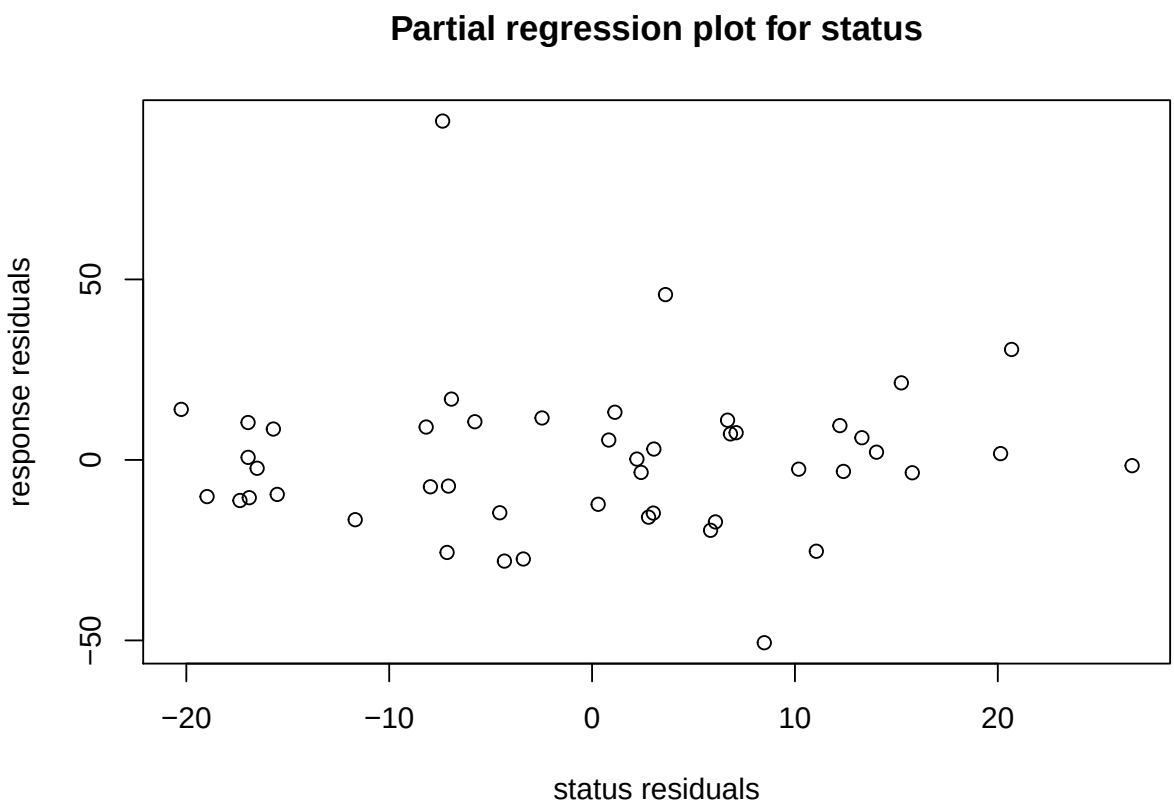


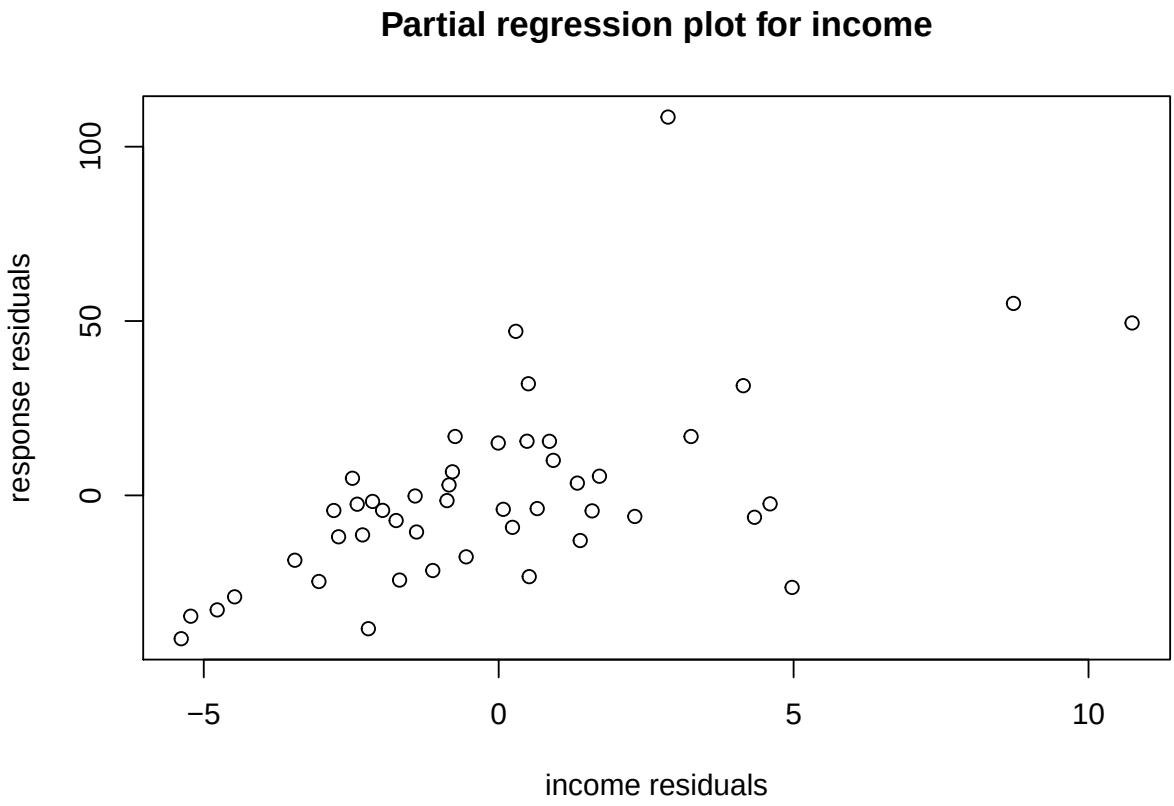
Perform partial regression

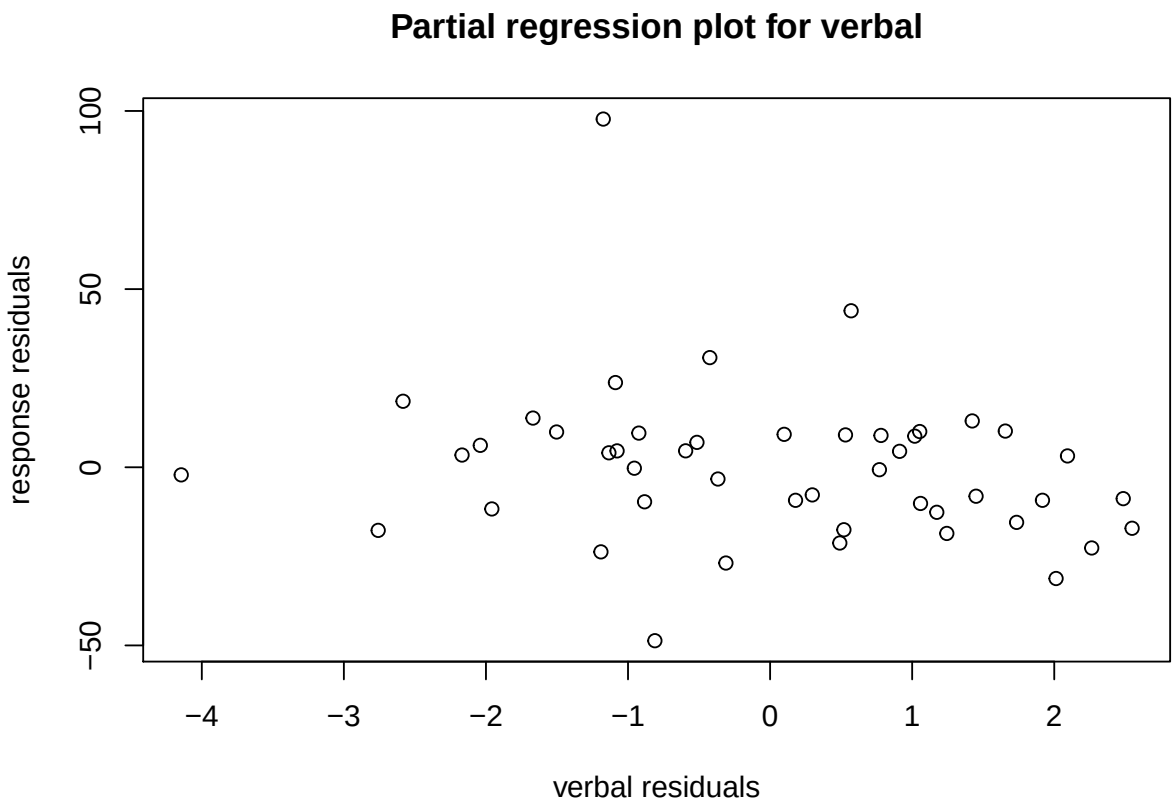
```
predictors <- names(lm.fit$coefficients)
predictors <- predictors[2:length(predictors)]

lm.formula <- formula(lm.fit)
response <- lm.formula[[2]]

for (i in 1:length(predictors)) {
    predictor <- predictors[i]
    others <- predictors[which(predictors != predictor)]
    d.formula <- paste(response, " ~ ", sep = "")
    m.formula <- paste(predictor, " ~ ", sep = "")

    for (j in 1:(length(others) - 1)) {
        d.formula <- paste(d.formula, others[j], " + ", sep = "")
        m.formula <- paste(m.formula, others[j], " + ", sep = "")
    }
    d.formula <- paste(d.formula, others[length(others)], sep = "")
    d.formula <- formula(d.formula)
```

```
    m.formula <- paste(m.formula, others[length(others)], sep = "")
    m.formula <- formula(m.formula)

    d <- residuals(lm(d.formula, df))

    m <- residuals(lm(m.formula, df))

    plot(m, d, xlab = paste(predictor, " residuals", sep = ""), ylab = "response residua
        main = paste("Partial regression plot for ", predictor, sep = ""))

}
```




Partial regression plot for status
response residuals
50
0
−50
−20
−10
0
10
20
status residuals


Partial regression plot for income
response residuals
100
50
0
−5
0
5
10
income residuals



6.3 For the prostate data, fit a model with lpsa as the response and the other variables as predictors.

```
rm(list = ls())
data(prostate, package = "faraway")
lm.fit <- lm(lpsa ~ ., data = prostate)
```

```
df <- prostate
numPredictors <- (ncol(df) - 1)
hatv <- hatvalues(lm.fit)
lev.cut <- (numPredictors + 1) * 2 * 1/nrow(df)
high.leverage <- df[hatv > lev.cut, ]
pander(high.leverage, caption = "High Leverage Data Elements")
```

Table 5: High Leverage Data Elements

	lcavol	lweight	age	lbph	svi	lcp	gleason	pgg45	lpsa
32	0.1823	6.108	65	1.705	0	-1.386	6	0	2.008

	lcavol	lweight	age	lbph	svi	lcp	gleason	pgg45	lpsa
37	1.423	3.657	73	-0.5798	0	1.658	8	15	2.158
41	0.6206	3.142	60	-1.386	0	-1.386	9	80	2.298
74	1.839	3.237	60	0.4383	1	1.179	9	90	3.075
92	2.533	3.678	61	1.348	1	-1.386	7	15	4.13

We've used the rule of thumb that points with a leverage greater than $\frac{2p}{n}$ should be looked at.

(d) Check for outliers.

```
studentized.residuals <- rstudent(lm.fit)
max.residual <- studentized.residuals[which.max(abs(studentized.residuals))]
range.residuals <- range(studentized.residuals)
names(range.residuals) <- c("left", "right")
pander(data.frame(range.residuals = t(range.residuals)), caption = "Range of Studentized
```

Table 6: Range of Studentized residuals

range.residuals.left	range.residuals.right
-2.617	2.554

```
p <- numPredictors + 1
n <- nrow(df)
t.val.alpha <- qt(0.05/(n * 2), n - p - 1)
pander(data.frame(t.val.alpha = t.val.alpha), caption = "Bonferroni corrected t-value")
```

Table 7: Bonferroni corrected t-value

t.val.alpha
-3.607

```
outlier.index <- abs(studentized.residuals) > abs(t.val.alpha)

outliers <- df[outlier.index == TRUE, ]

if (nrow(outliers) >= 1) {
    pander(outliers, caption = "outliers")
```

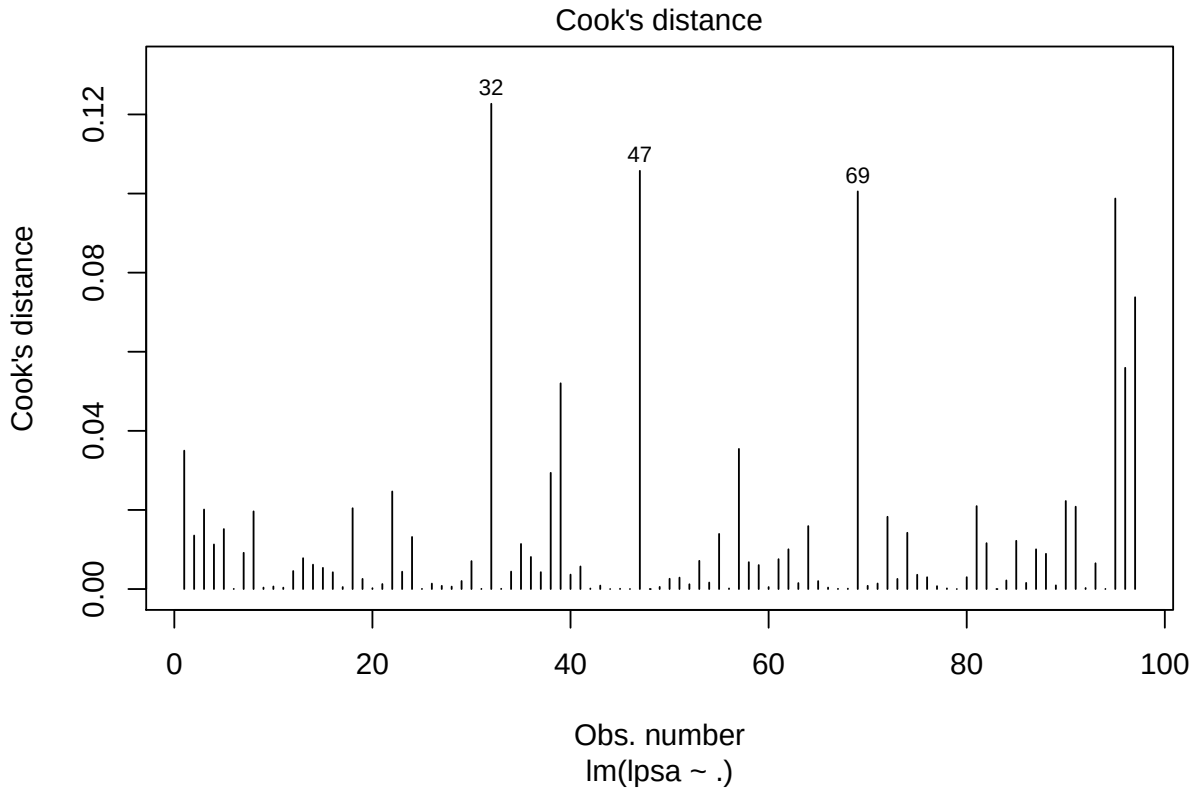
```
}
```

Here we look for studentized residuals that fall outside the interval given by the Bonferroni corrected t-values.

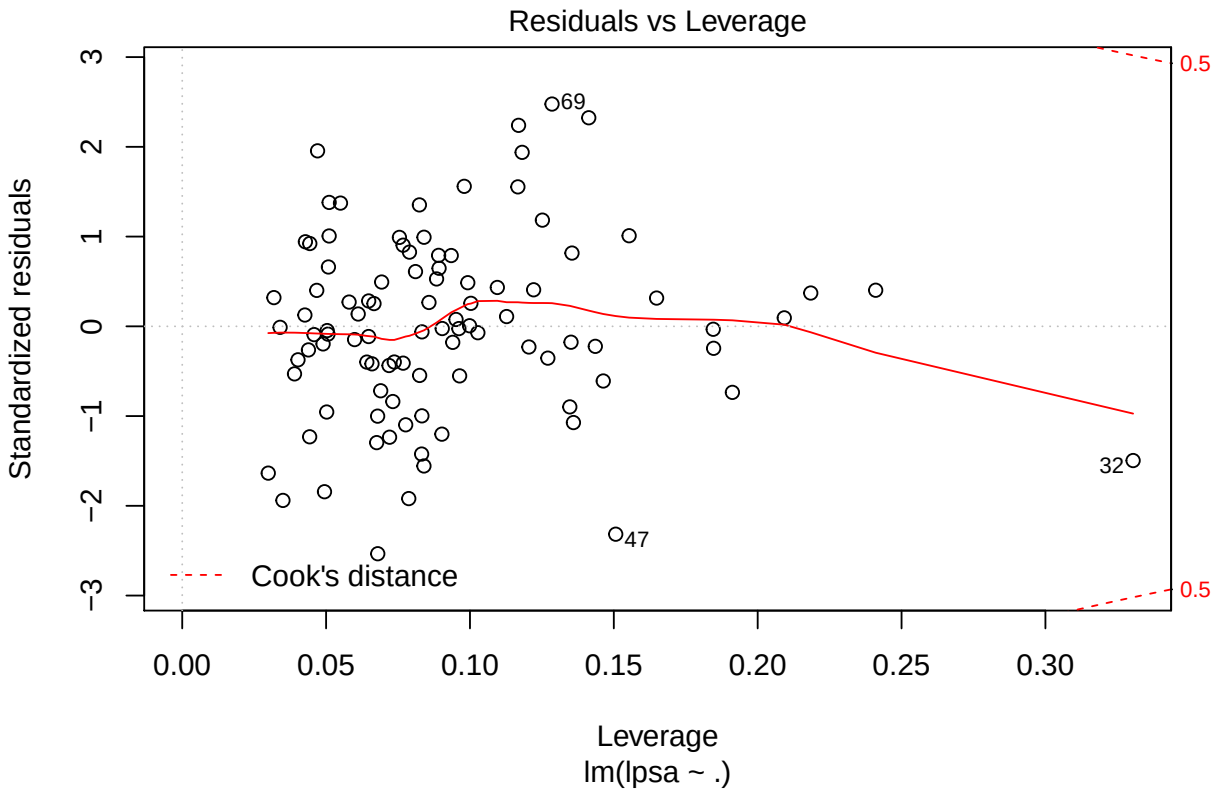
(e) Check for influential points.

We plot the Cook's distances and the residual-leverage plot with level set contours of the Cook distance.

```
plot(lm.fit, which = 4)
```



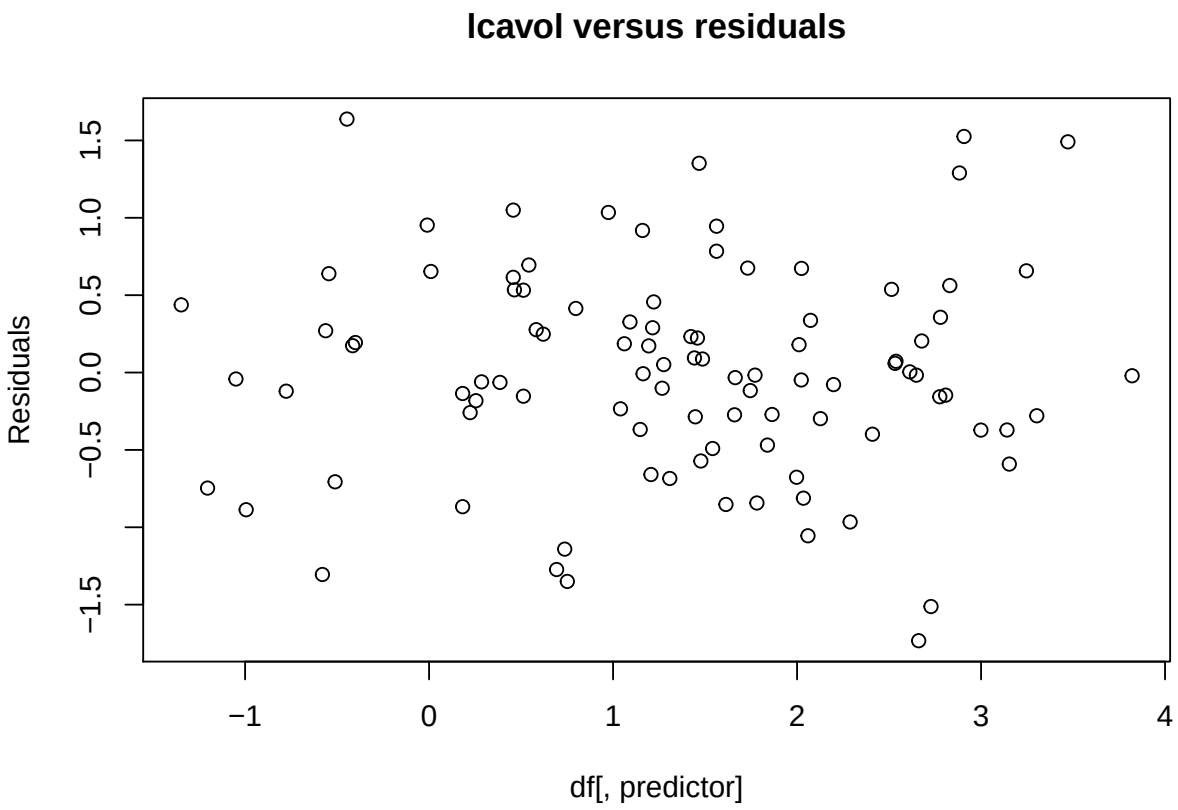

```
plot(lm.fit, which = 5)
```

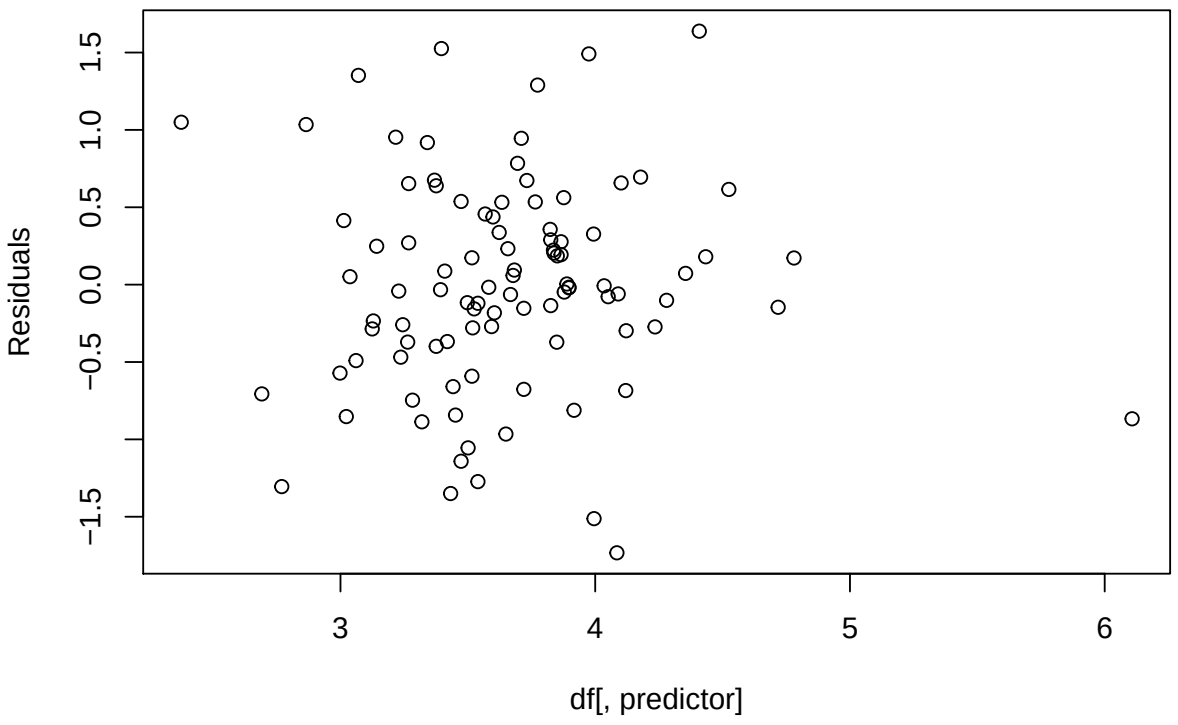


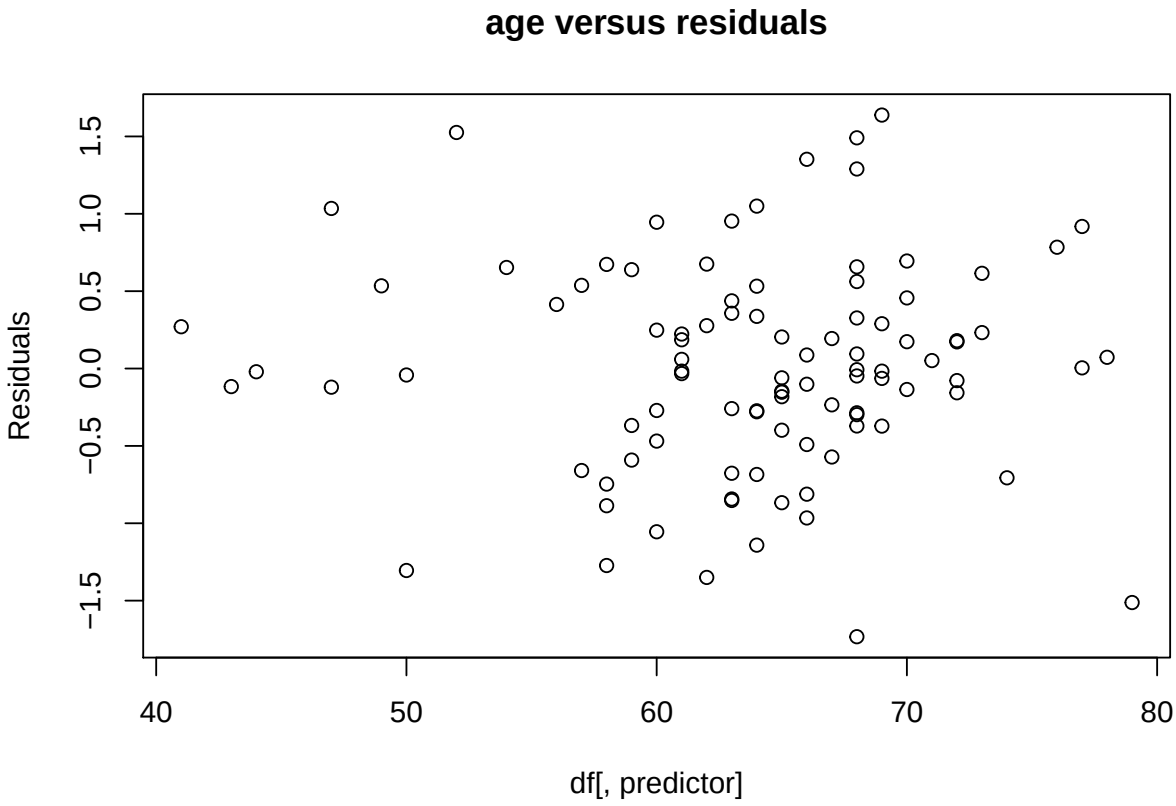
(f) Check for structure in the model.

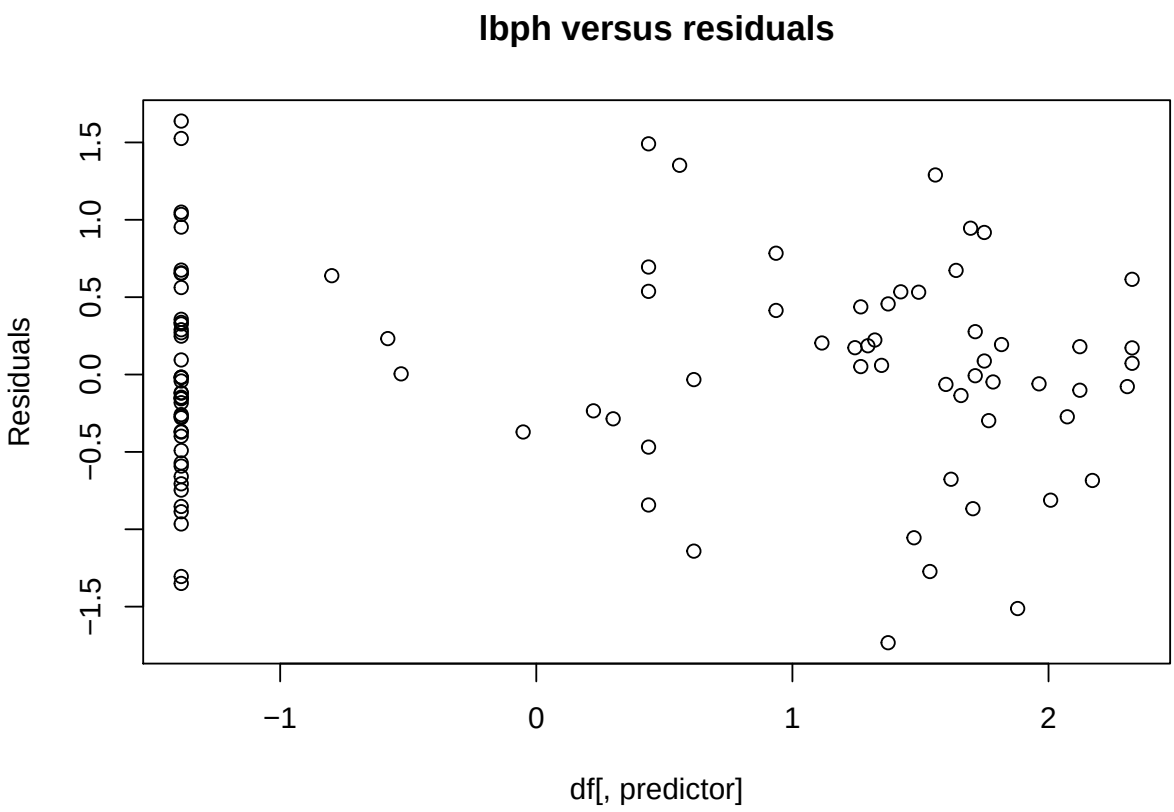
Plot residuals versus predictors

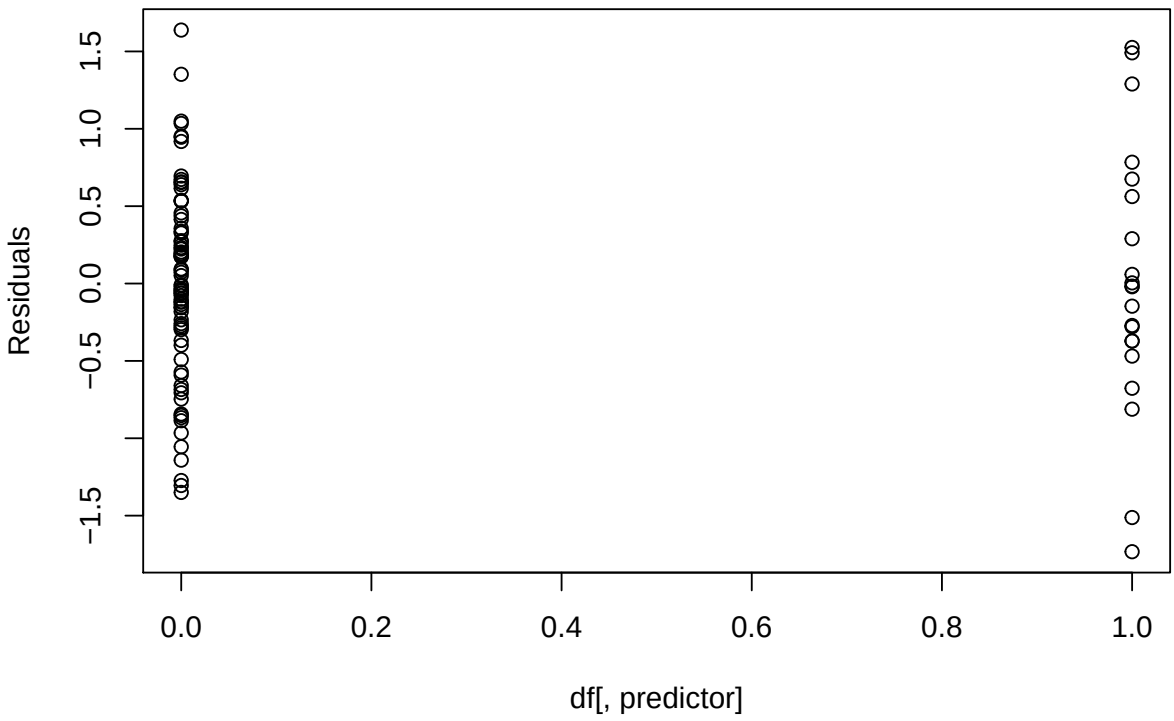
```
predictors <- names(lm.fit$coefficients)
predictors <- predictors[2:length(predictors)]

for (i in 1:length(predictors)) {
    predictor <- predictors[i]

    plot(df[, predictor], residuals(lm.fit), xlab = , ylab = "Residuals", main = paste(p
        " versus residuals", sep = ""))

}
```

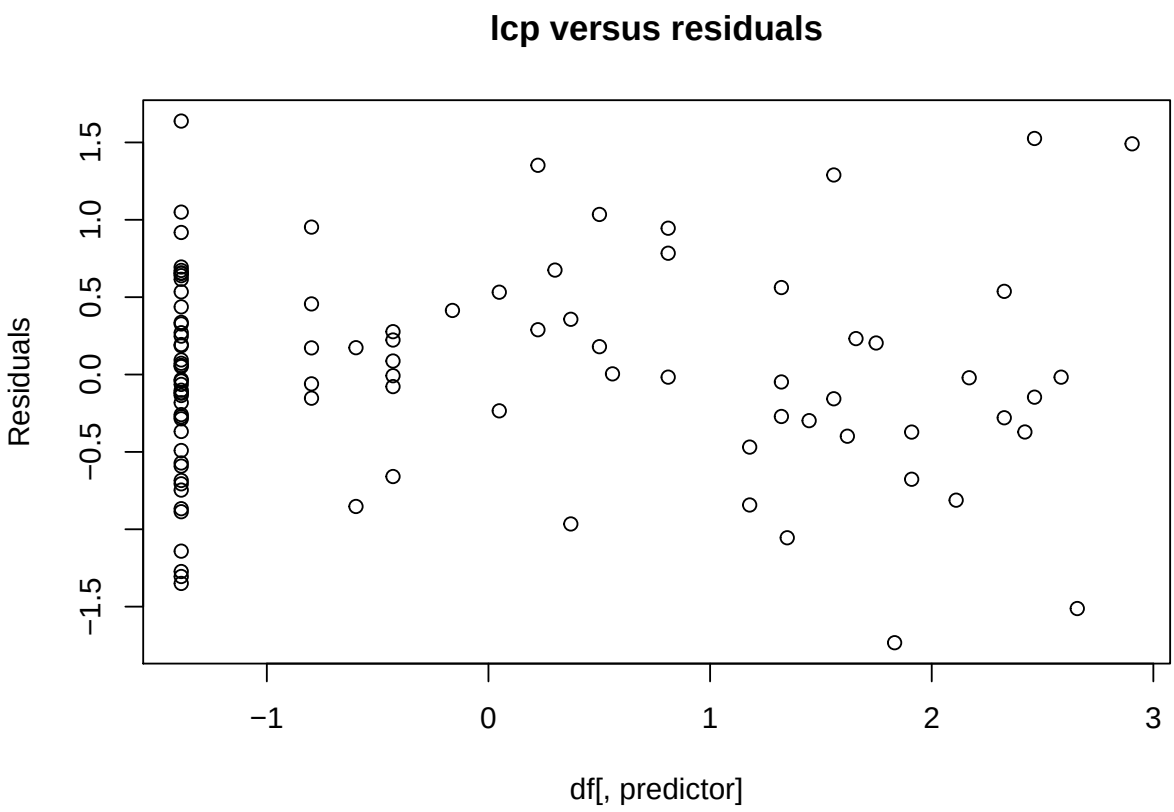

lcavol versus residuals
Residuals
1.5
1.0
0.5
0.0
−0.5
−1.5
−1
0
1
2
3
4
df[, predictor]

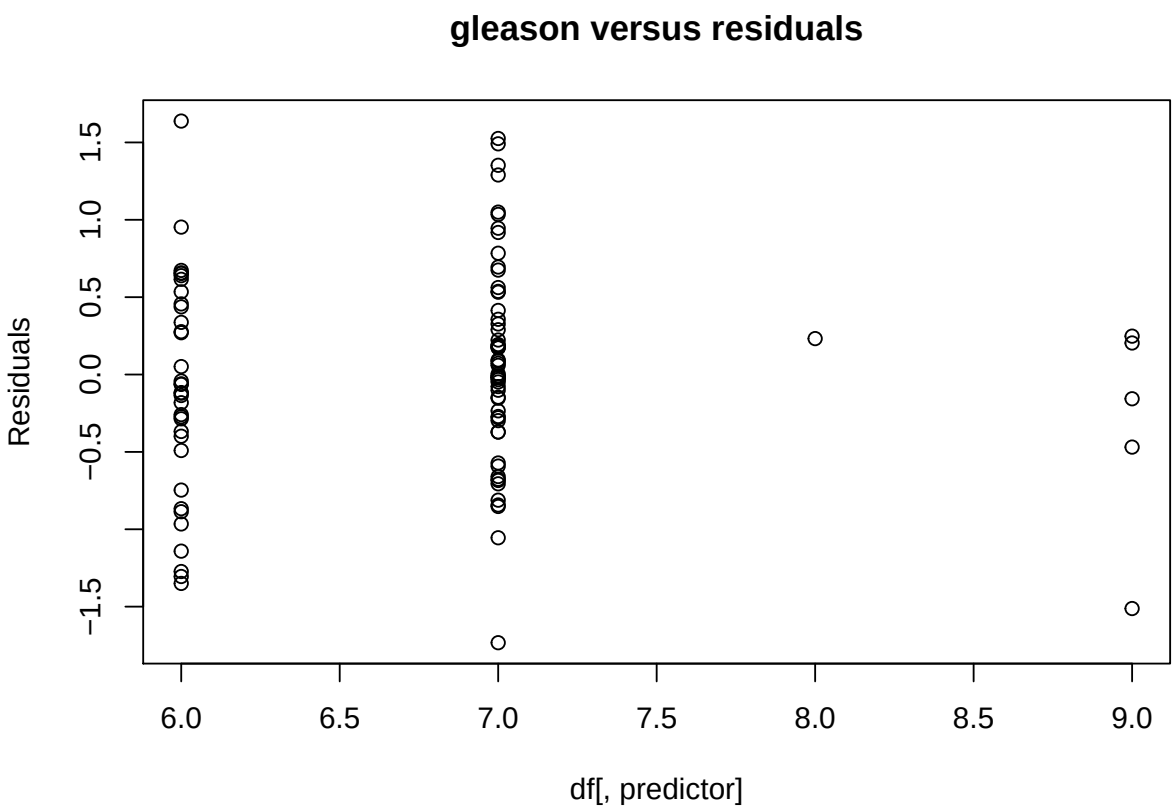

lweight versus residuals
Residuals
1.5
1.0
0.5
0.0
−0.5
−1.5
3
4
5
6
df[, predictor]

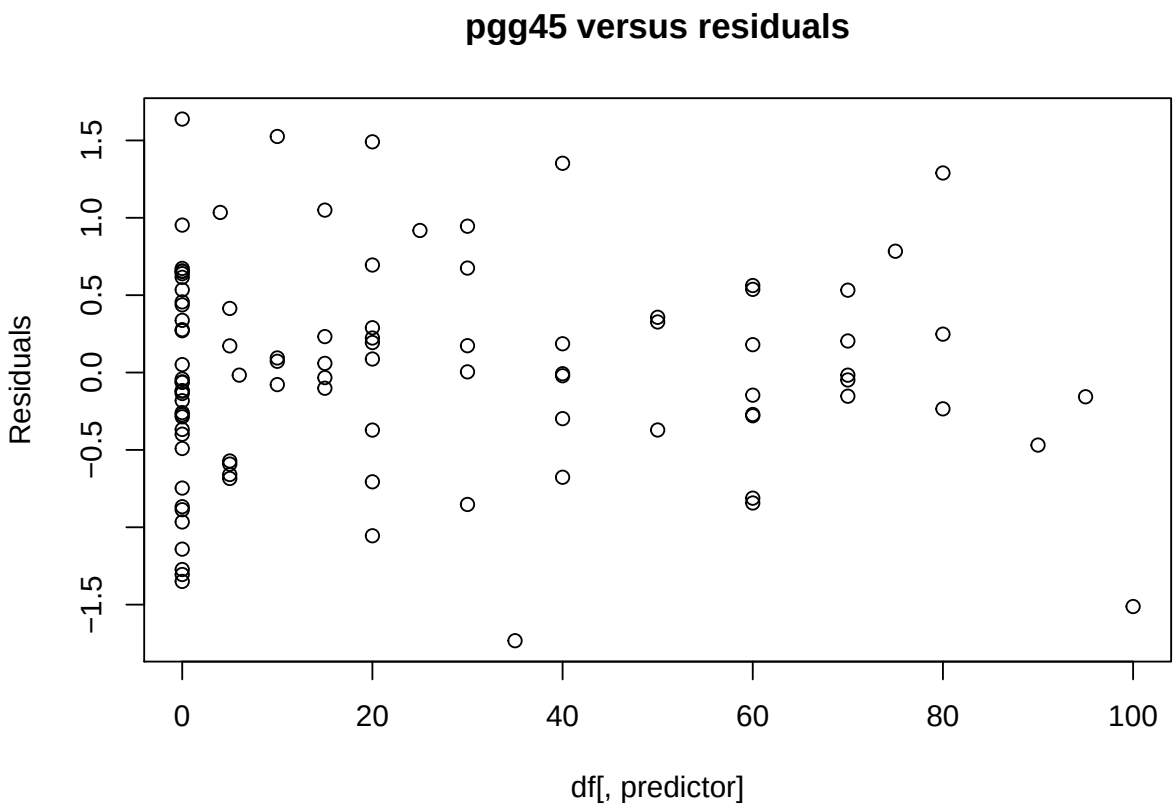

age versus residuals
Residuals
1.5
1.0
0.5
0.0
−0.5
−1.5
40
50
60
70
80
df[, predictor]


lbph versus residuals
Residuals
1.5
1.0
0.5
0.0
−0.5
−1.5
−1
0
1
2
df[, predictor]


svi versus residuals
Residuals
−1.5
−0.5
0.0
0.5
1.0
1.5
0.0
0.2
0.4
0.6
0.8
1.0
df[, predictor]


lcp versus residuals
Residuals
−1.5
−0.5
0.0
0.5
1.0
1.5
−1
0
1
2
3
df[, predictor]

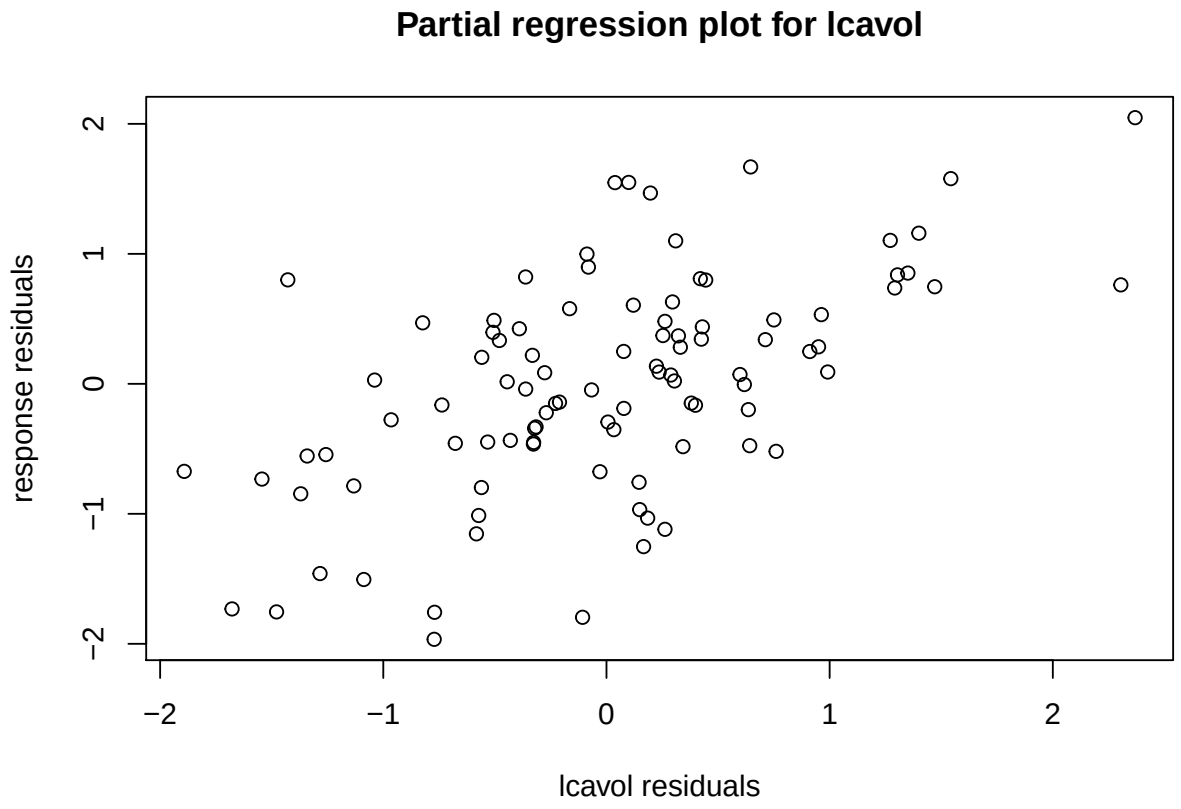

gleason versus residuals
Residuals
1.5
1.0
0.5
0.0
−0.5
−1.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0
df[, predictor]

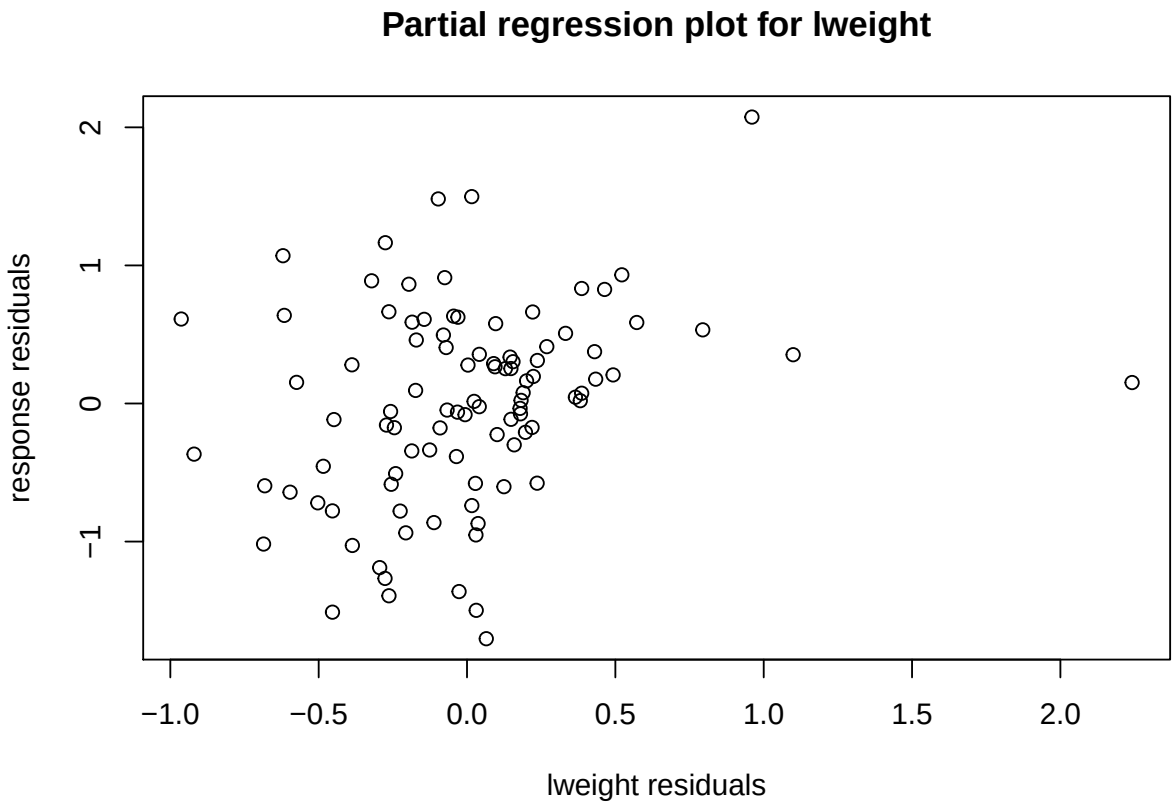


Perform partial regression

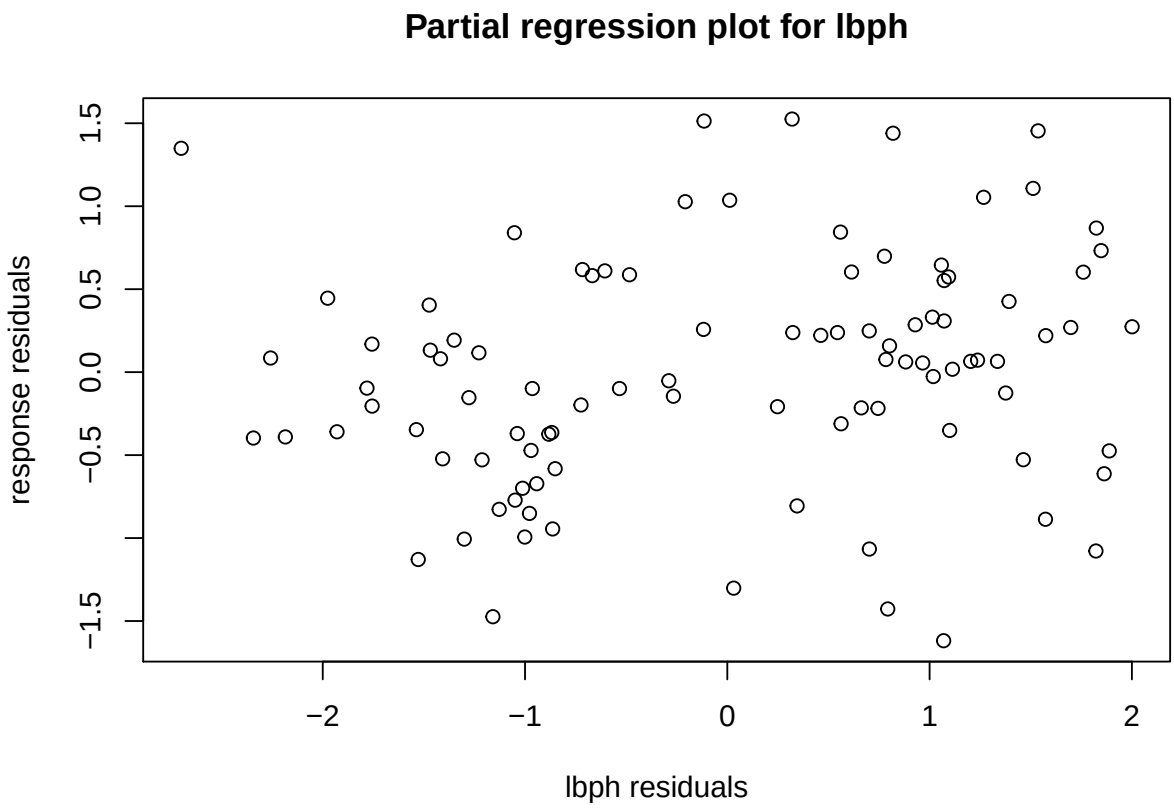
```
predictors <- names(lm.fit$coefficients)
predictors <- predictors[2:length(predictors)]

lm.formula <- formula(lm.fit)
response <- lm.formula[[2]]

for (i in 1:length(predictors)) {
    predictor <- predictors[i]
    others <- predictors[which(predictors != predictor)]
    d.formula <- paste(response, " ~ ", sep = "")
    m.formula <- paste(predictor, " ~ ", sep = "")

    for (j in 1:(length(others) - 1)) {
        d.formula <- paste(d.formula, others[j], " + ", sep = "")
        m.formula <- paste(m.formula, others[j], " + ", sep = "")
    }
    d.formula <- paste(d.formula, others[length(others)], sep = "")
    d.formula <- formula(d.formula)
```

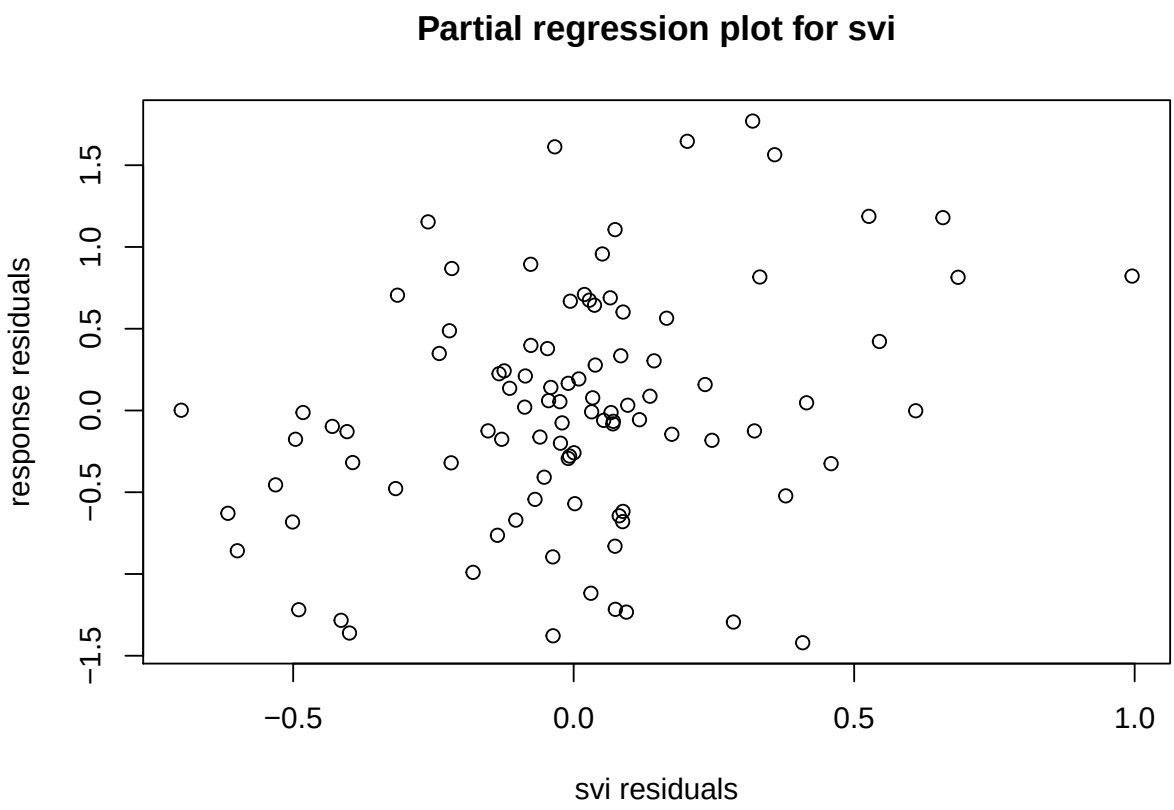
```
    m.formula <- paste(m.formula, others[length(others)], sep = "")
    m.formula <- formula(m.formula)

    d <- residuals(lm(d.formula, df))

    m <- residuals(lm(m.formula, df))

    plot(m, d, xlab = paste(predictor, " residuals", sep = ""), ylab = "response residua
        main = paste("Partial regression plot for ", predictor, sep = ""))

}
```

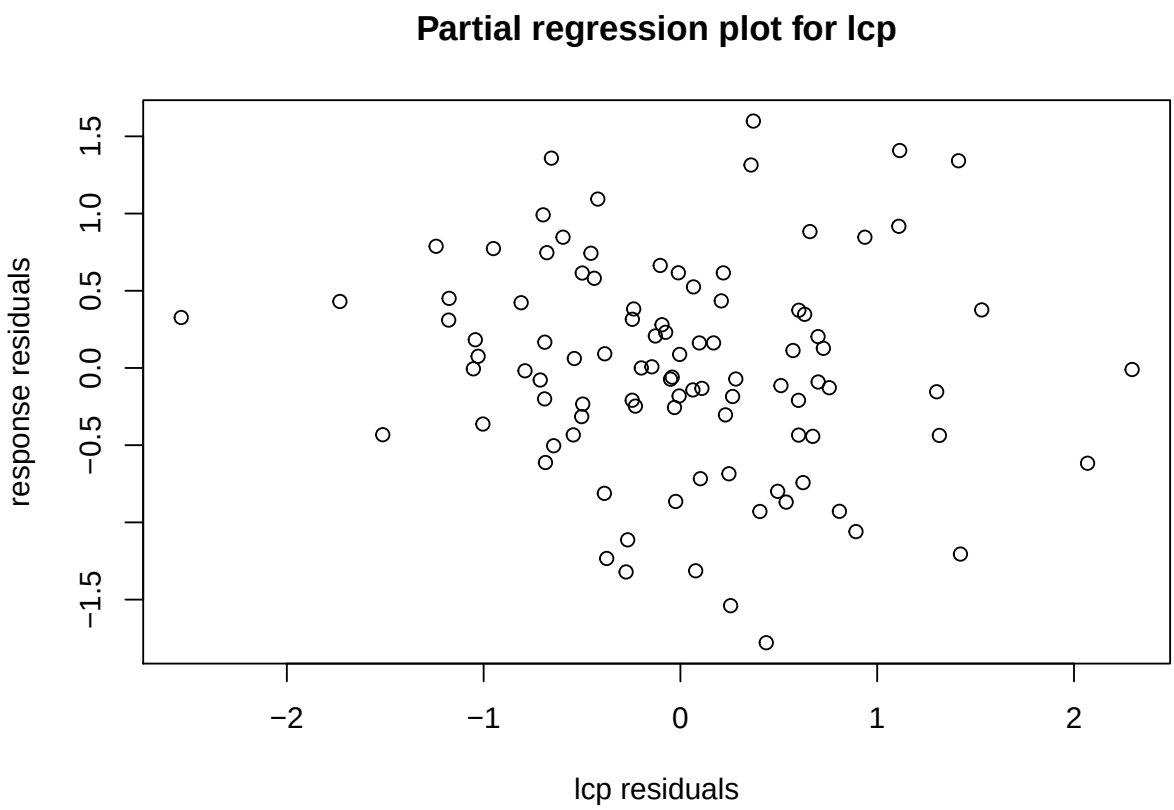


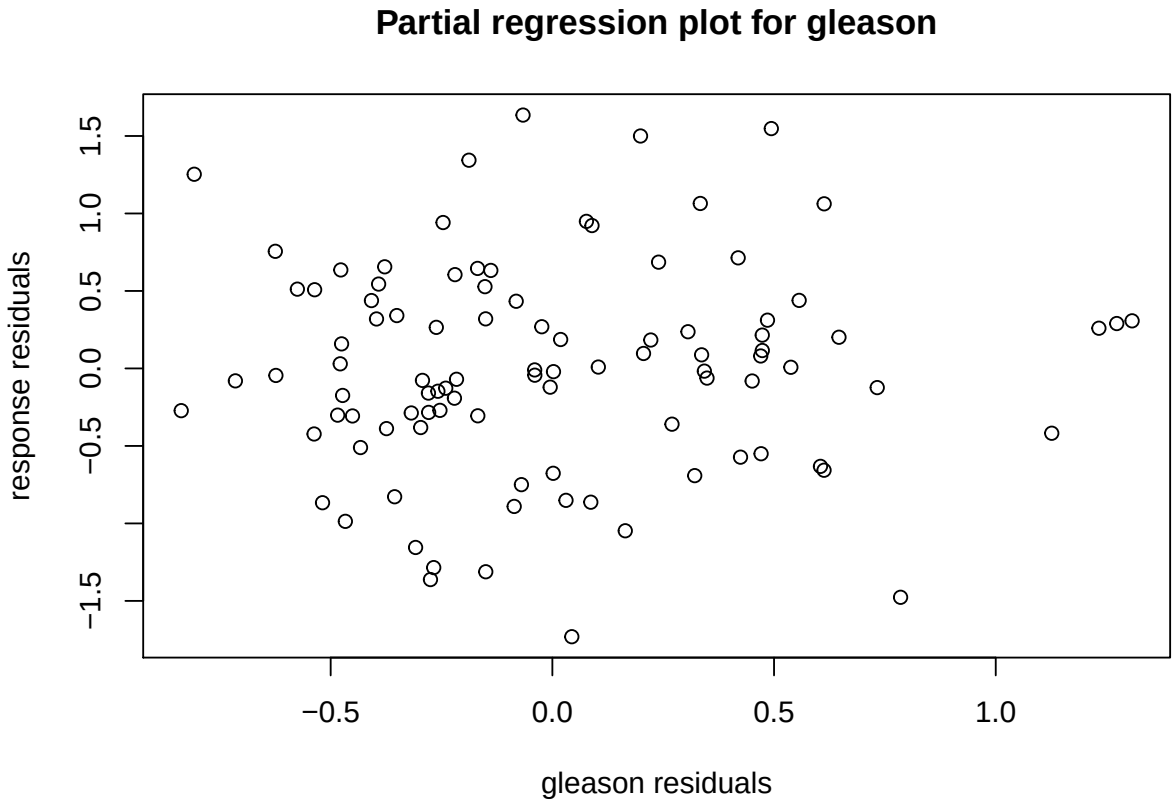

Partial regression plot for lweight
response residuals
2
1
0
−1
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
lweight residuals

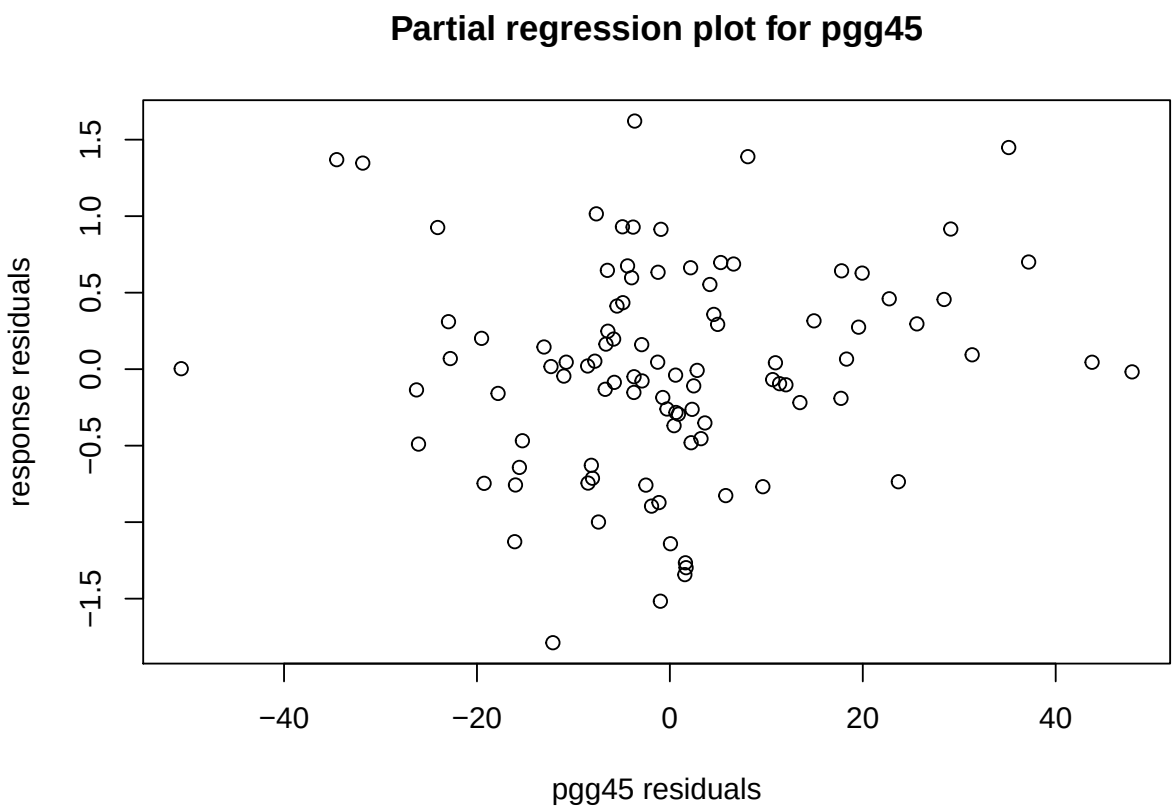

Partial regression plot for age
response residuals
age residuals
−1.5
−0.5
0.0
0.5
1.0
1.5
−20
−15
−10
−5
0
5
10
15


Partial regression plot for lbph
response residuals
lbph residuals
1.5
1.0
0.5
0.0
−0.5
−1.5
−2
−1
0
1
2


Partial regression plot for svi
response residuals
svi residuals
−1.5
−0.5
0.0
0.5
1.0
1.5
−0.5
0.0
0.5
1.0


Partial regression plot for lcp
response residuals
−1.5
−0.5
0.0
0.5
1.0
1.5
−2
−1
0
1
2
lcp residuals


Partial regression plot for gleason
response residuals
1.5
1.0
0.5
0.0
−0.5
−1.5
−0.5
0.0
0.5
1.0
gleason residuals



6.4 For the swiss data, fit a model with Fertility as the response and the other variables as predictors.

```
rm(list = ls())
data(swiss, package = "faraway")
lm.fit <- lm(Fertility ~ ., data = swiss)
```

```
df <- swiss
numPredictors <- (ncol(df) - 1)
hatv <- hatvalues(lm.fit)
lev.cut <- (numPredictors + 1) * 2 * 1/nrow(df)
high.leverage <- df[hatv > lev.cut, ]
pander(high.leverage, caption = "High Leverage Data Elements")
```

Table 8: High Leverage Data Elements (continued below)

	Fertility	Agriculture	Examination	Education
La Vallee	54.3	15.2	31	20
V. De Geneve	35	1.2	37	53

	Fertility	Agriculture	Examination	Education

	Catholic	Infant.Mortality
La Vallee	2.15	10.8
V. De Geneve	42.34	18

We've used the rule of thumb that points with a leverage greater than $\frac{2p}{n}$ should be looked at.

(d) Check for outliers.

```
studentized.residuals <- rstudent(lm.fit)
max.residual <- studentized.residuals[which.max(abs(studentized.residuals))]
range.residuals <- range(studentized.residuals)
names(range.residuals) <- c("left", "right")
pander(data.frame(range.residuals = t(range.residuals)), caption = "Range of Studentized
```

Table 10: Range of Studentized residuals

range.residuals.left	range.residuals.right
-2.394	2.445

```
p <- numPredictors + 1
n <- nrow(df)
t.val.alpha <- qt(0.05/(n * 2), n - p - 1)
pander(data.frame(t.val.alpha = t.val.alpha), caption = "Bonferroni corrected t-value")
```

Table 11: Bonferroni corrected t-value

t.val.alpha
-3.529

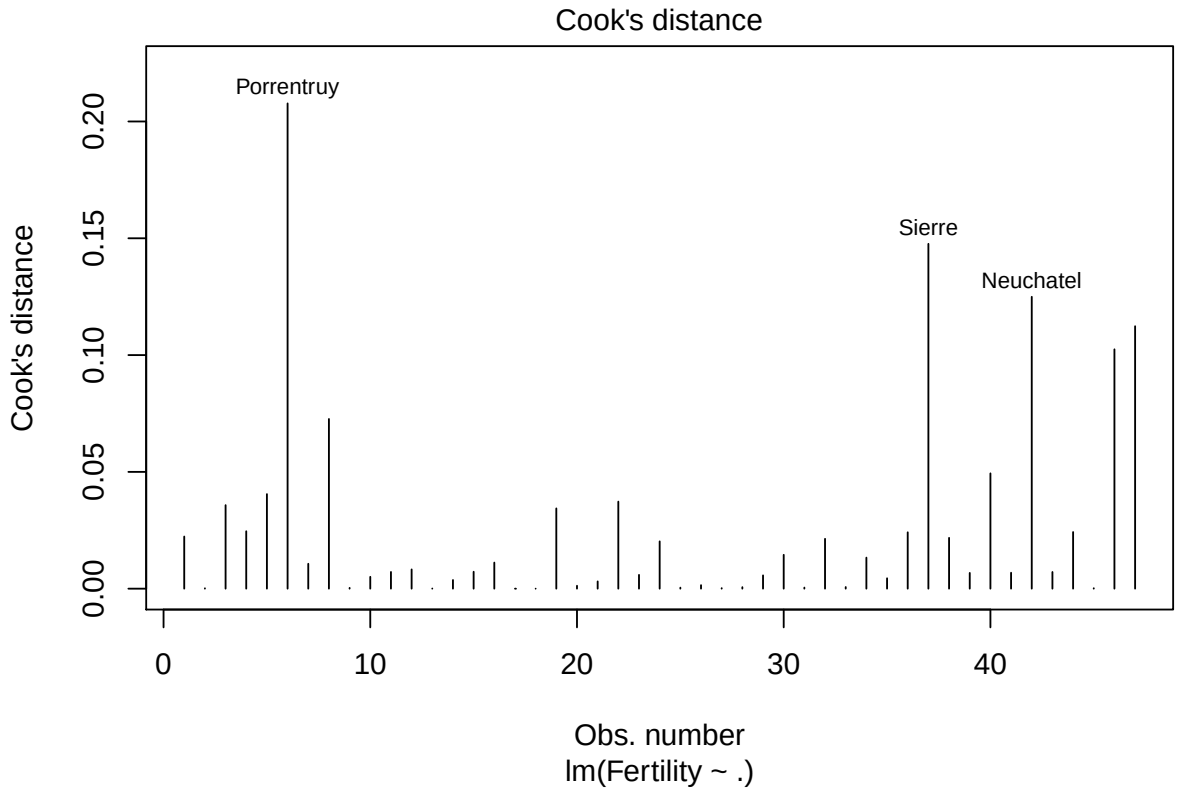
```
outlier.index <- abs(studentized.residuals) > abs(t.val.alpha)

outliers <- df[outlier.index == TRUE, ]

if (nrow(outliers) >= 1) {
    pander(outliers, caption = "outliers")
}
```

Here we look for studentized residuals that fall outside the interval given by the Bonferroni corrected t-values.

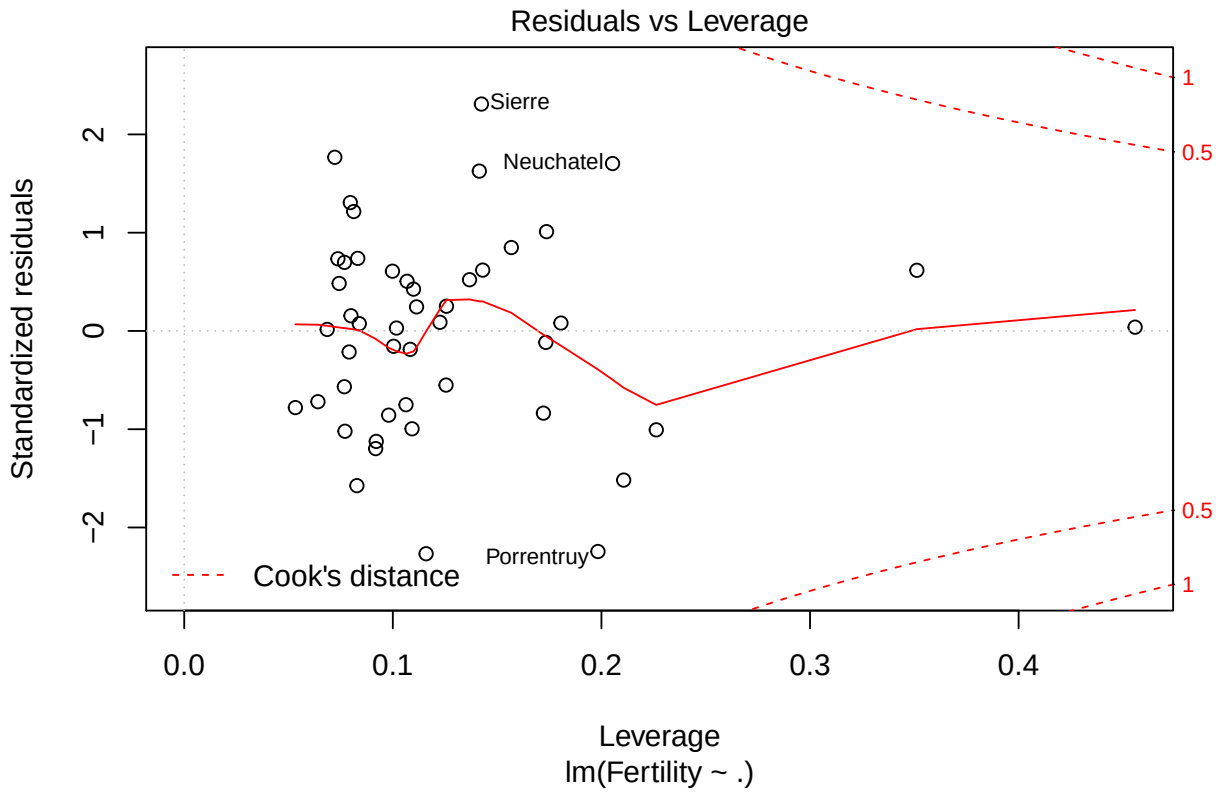
(e) Check for influential points.

We plot the Cook's distances and the residual-leverage plot with level set contours of the Cook distance.

```
plot(lm.fit, which = 4)
```



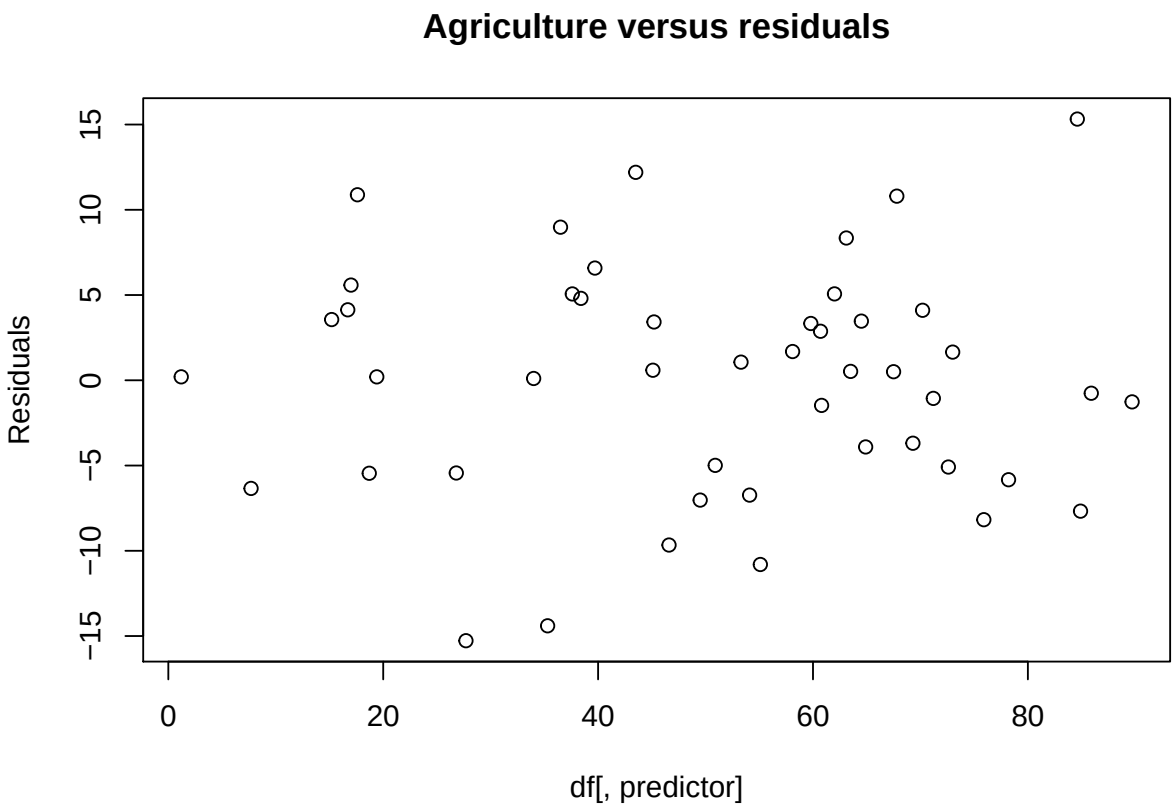

```
plot(lm.fit, which = 5)
```

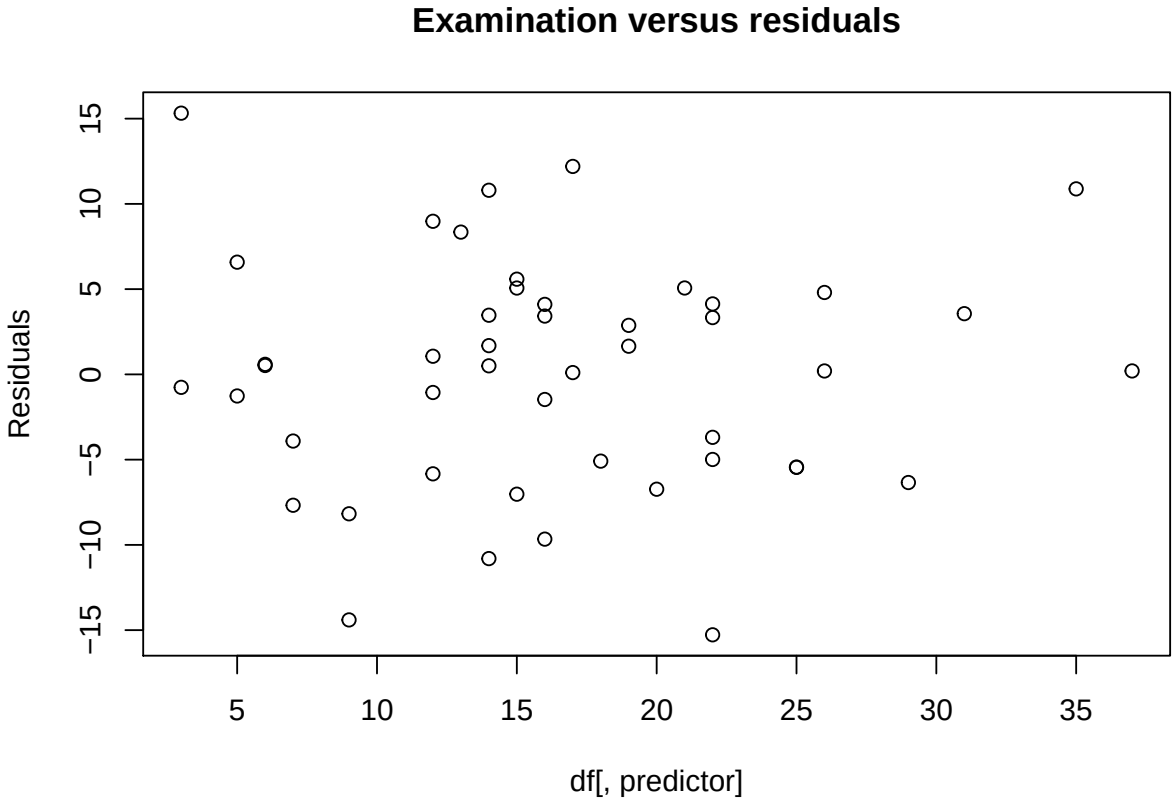


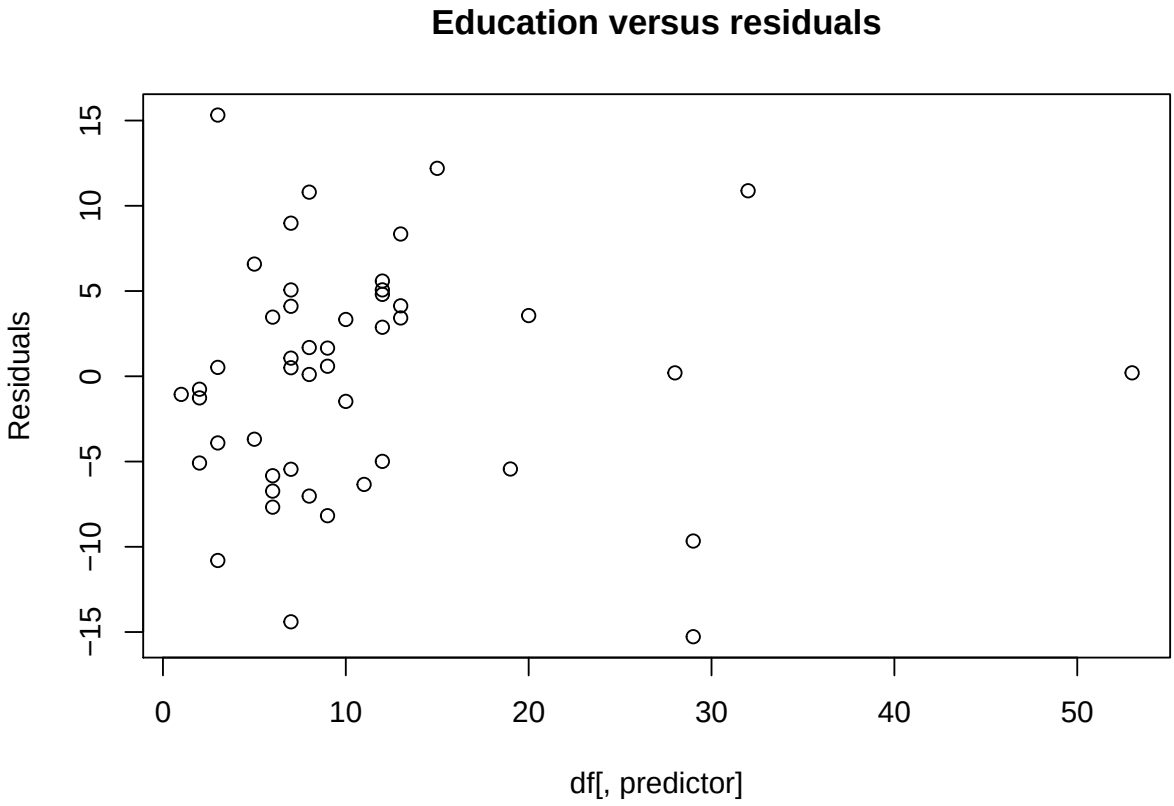
(f) Check for structure in the model.

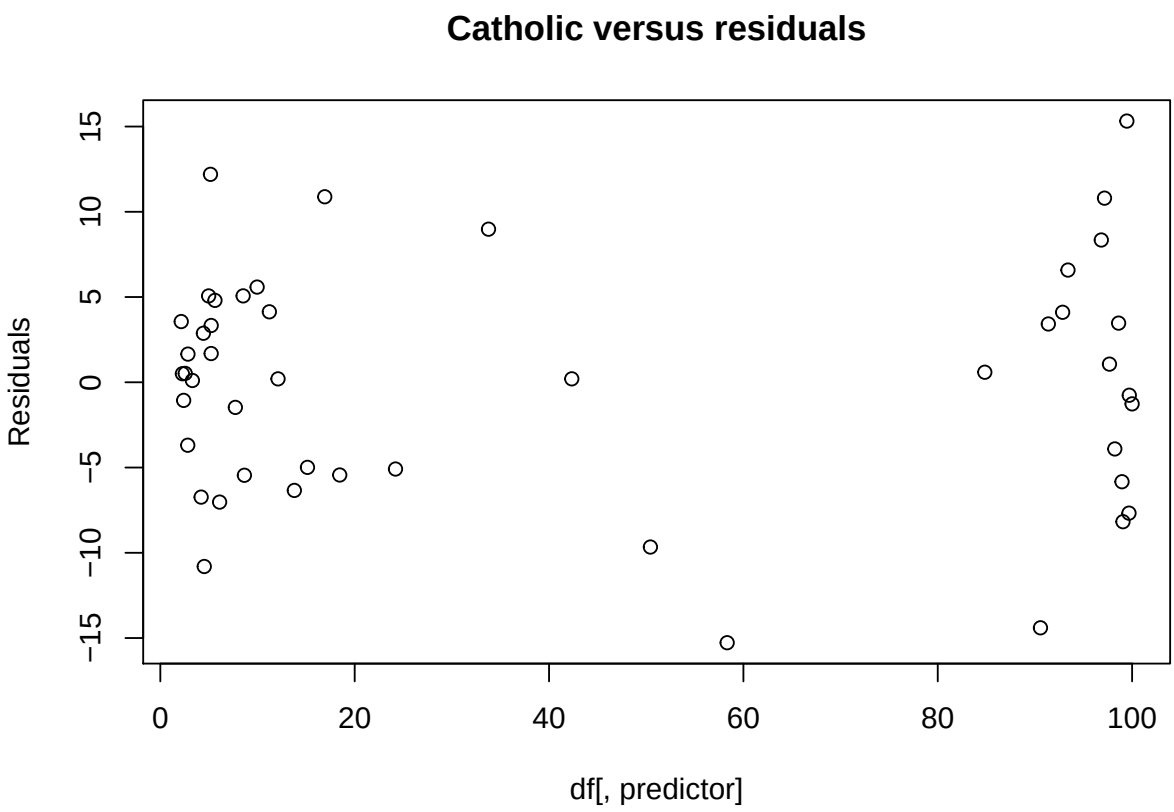
Plot residuals versus predictors

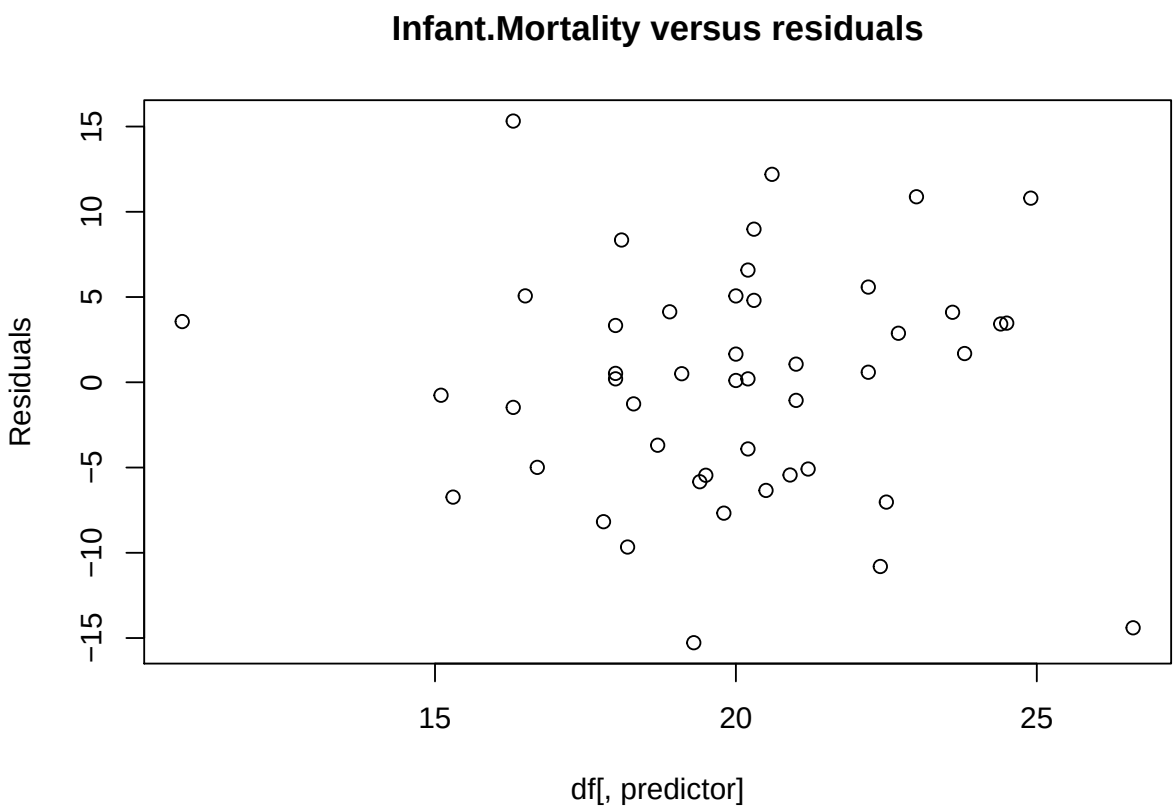
```
predictors <- names(lm.fit$coefficients)
predictors <- predictors[2:length(predictors)]

for (i in 1:length(predictors)) {
    predictor <- predictors[i]

    plot(df[, predictor], residuals(lm.fit), xlab = , ylab = "Residuals", main = paste(p
        " versus residuals", sep = ""))

}
```


Agriculture versus residuals
Residuals
15
10
5
0
−5
−10
−15
0
20
40
60
80
df[, predictor]


Examination versus residuals
Residuals
15
10
5
0
−5
−10
−15
5
10
15
20
25
30
35
df[, predictor]


Education versus residuals
Residuals
15
10
5
0
−5
−10
−15
0
10
20
30
40
50
df[, predictor]

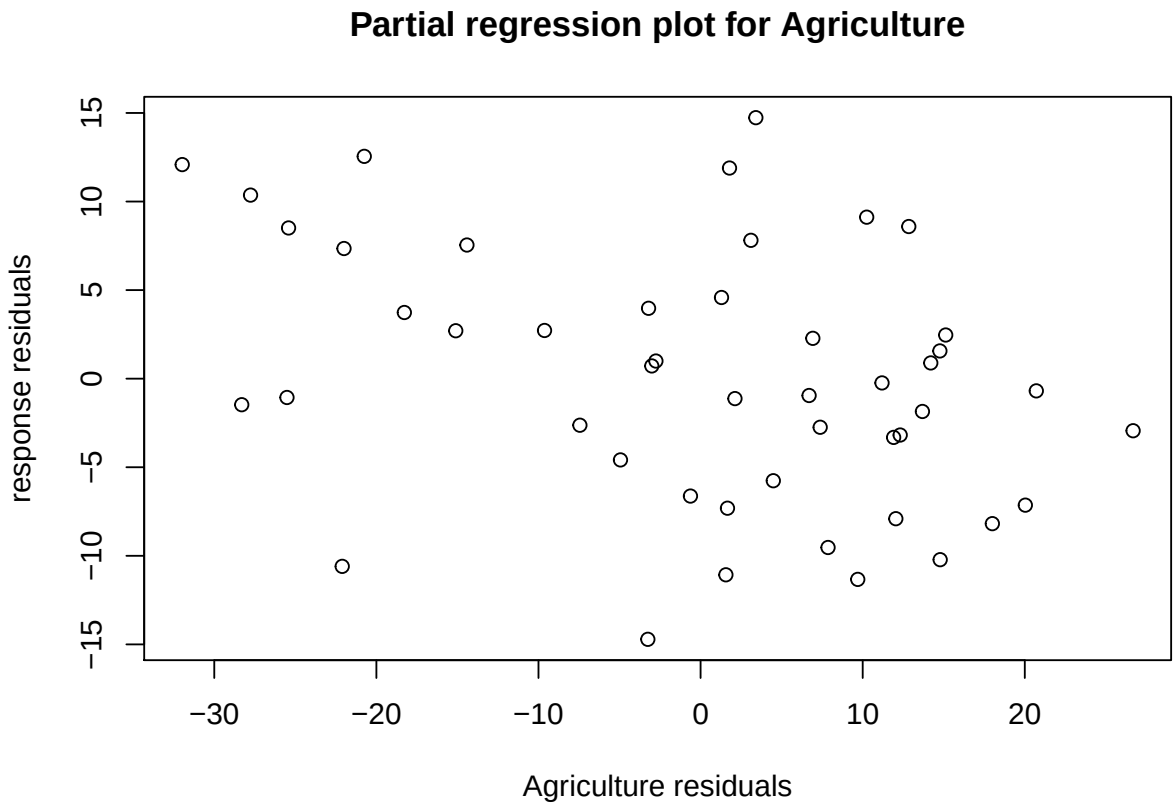

Catholic versus residuals
Residuals
15
10
5
0
−5
−10
−15
0
20
40
60
80
100
df[, predictor]

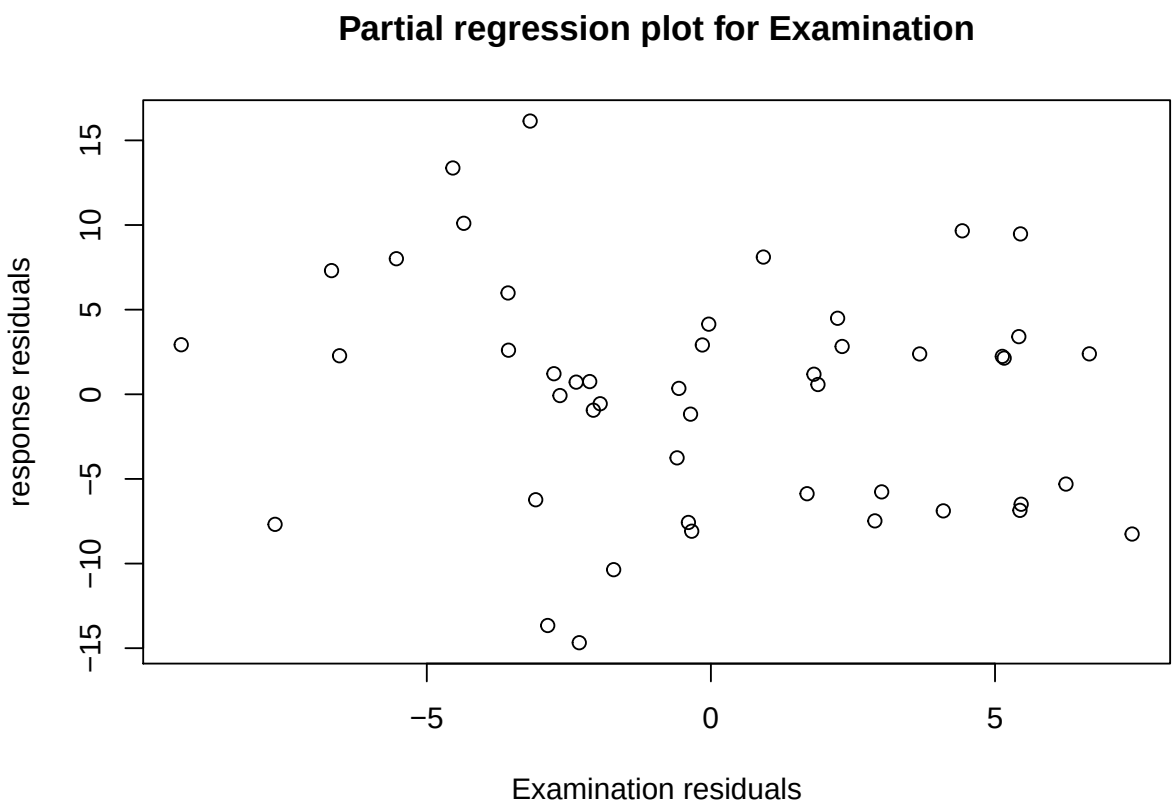


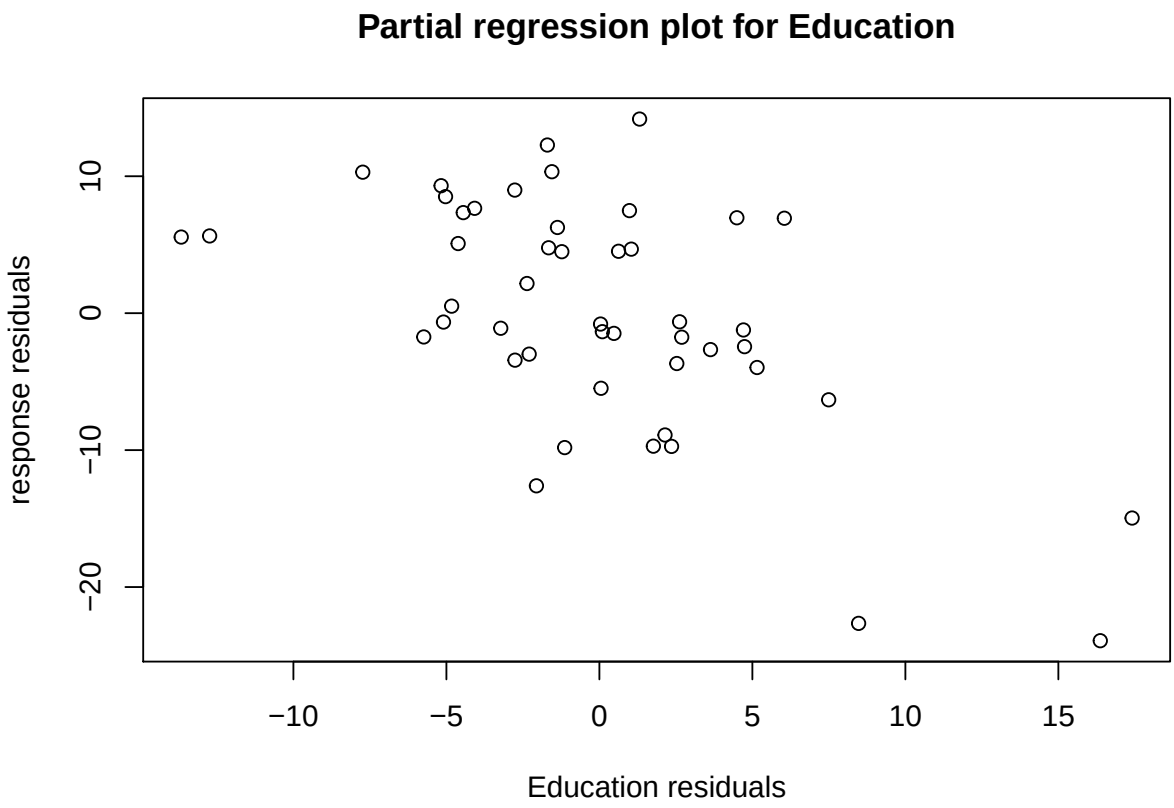
Perform partial regression

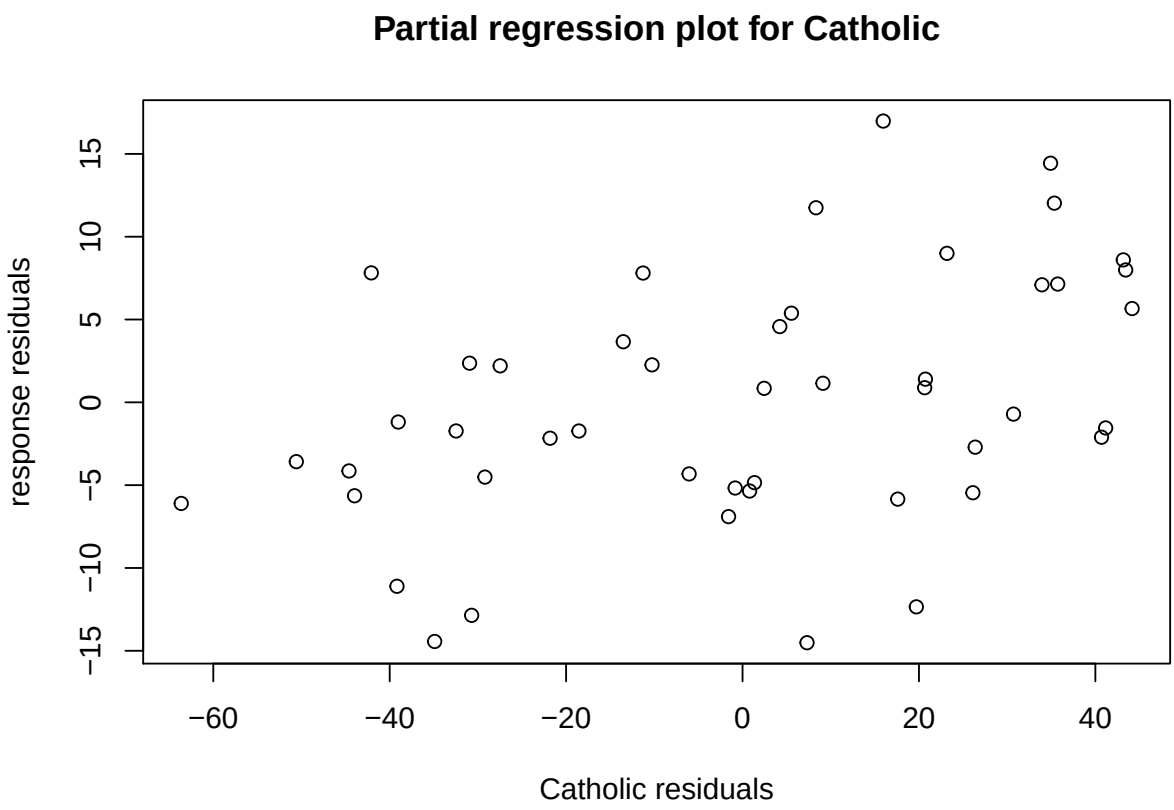
```
predictors <- names(lm.fit$coefficients)
predictors <- predictors[2:length(predictors)]

lm.formula <- formula(lm.fit)
response <- lm.formula[[2]]

for (i in 1:length(predictors)) {
    predictor <- predictors[i]
    others <- predictors[which(predictors != predictor)]
    d.formula <- paste(response, " ~ ", sep = "")
    m.formula <- paste(predictor, " ~ ", sep = "")

    for (j in 1:(length(others) - 1)) {
        d.formula <- paste(d.formula, others[j], " + ", sep = "")
        m.formula <- paste(m.formula, others[j], " + ", sep = "")
    }
    d.formula <- paste(d.formula, others[length(others)], sep = "")
    d.formula <- formula(d.formula)
```

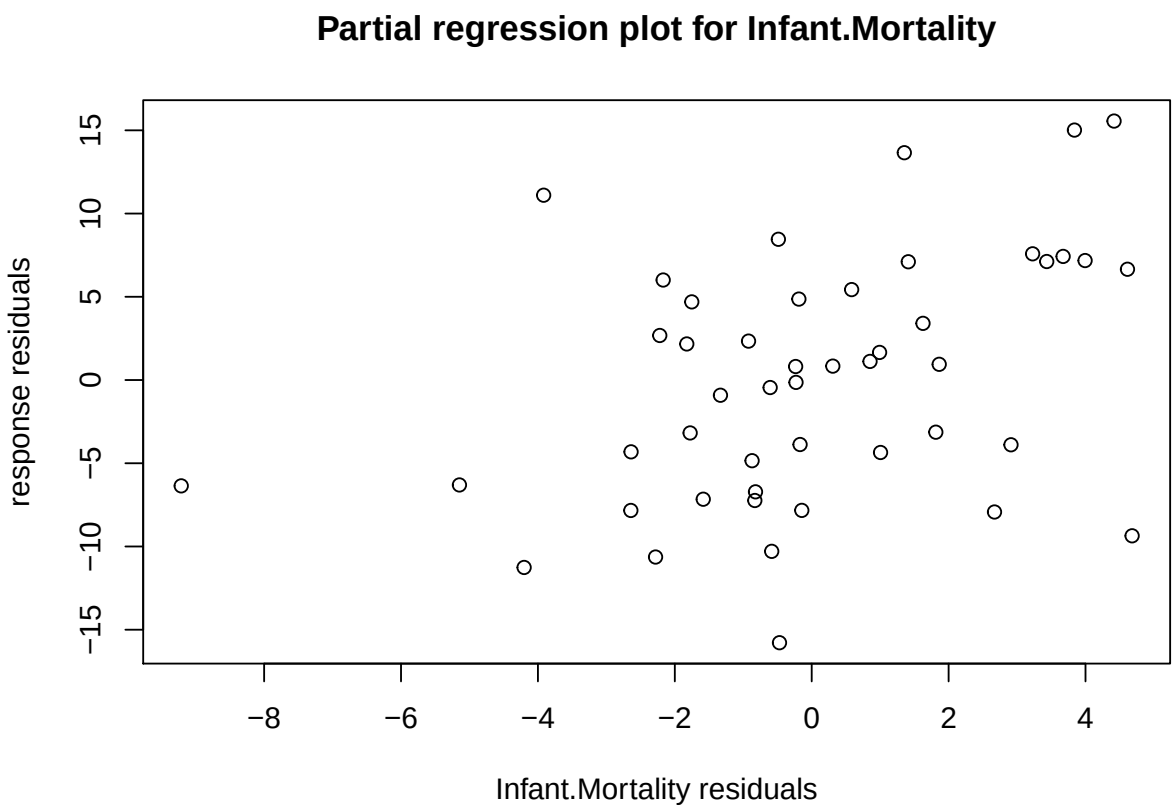
```
    m.formula <- paste(m.formula, others[length(others)], sep = "")
    m.formula <- formula(m.formula)

    d <- residuals(lm(d.formula, df))

    m <- residuals(lm(m.formula, df))

    plot(m, d, xlab = paste(predictor, " residuals", sep = ""), ylab = "response residua
        main = paste("Partial regression plot for ", predictor, sep = ""))

}
```




Partial regression plot for Examination
response residuals
15
10
5
0
−5
−10
−15
−5
0
5
Examination residuals


Partial regression plot for Education
response residuals
10
0
−10
−20
−10
−5
0
5
10
15
Education residuals


Partial regression plot for Catholic
response residuals
Catholic residuals
15
10
5
0
−5
−10
−15
−60
−40
−20
0
20
40



6.5 Using the cheddar data, fit a model with taste as the response and the other three variables as predictors.

```
rm(list = ls())
data(cheddar, package = "faraway")
lm.fit <- lm(taste ~ ., data = cheddar)
```

```
df <- cheddar
numPredictors <- (ncol(df) - 1)
hatv <- hatvalues(lm.fit)
lev.cut <- (numPredictors + 1) * 2 * 1/nrow(df)
high.leverage <- df[hatv > lev.cut, ]
pander(high.leverage, caption = "High Leverage Data Elements")
```

Table 12: High Leverage Data Elements

taste	Acetic	H2S	Lactic

We've used the rule of thumb that points with a leverage greater than $\frac{2p}{n}$ should be looked at.

(d) Check for outliers.

```
studentized.residuals <- rstudent(lm.fit)
max.residual <- studentized.residuals[which.max(abs(studentized.residuals))]
range.residuals <- range(studentized.residuals)
names(range.residuals) <- c("left", "right")
pander(data.frame(range.residuals = t(range.residuals)), caption = "Range of Studentized
```

Table 13: Range of Studentized residuals

range.residuals.left	range.residuals.right
-1.878	3.015

```
p <- numPredictors + 1
n <- nrow(df)
t.val.alpha <- qt(0.05/(n * 2), n - p - 1)
pander(data.frame(t.val.alpha = t.val.alpha), caption = "Bonferroni corrected t-value")
```

Table 14: Bonferroni corrected t-value

t.val.alpha
-3.523

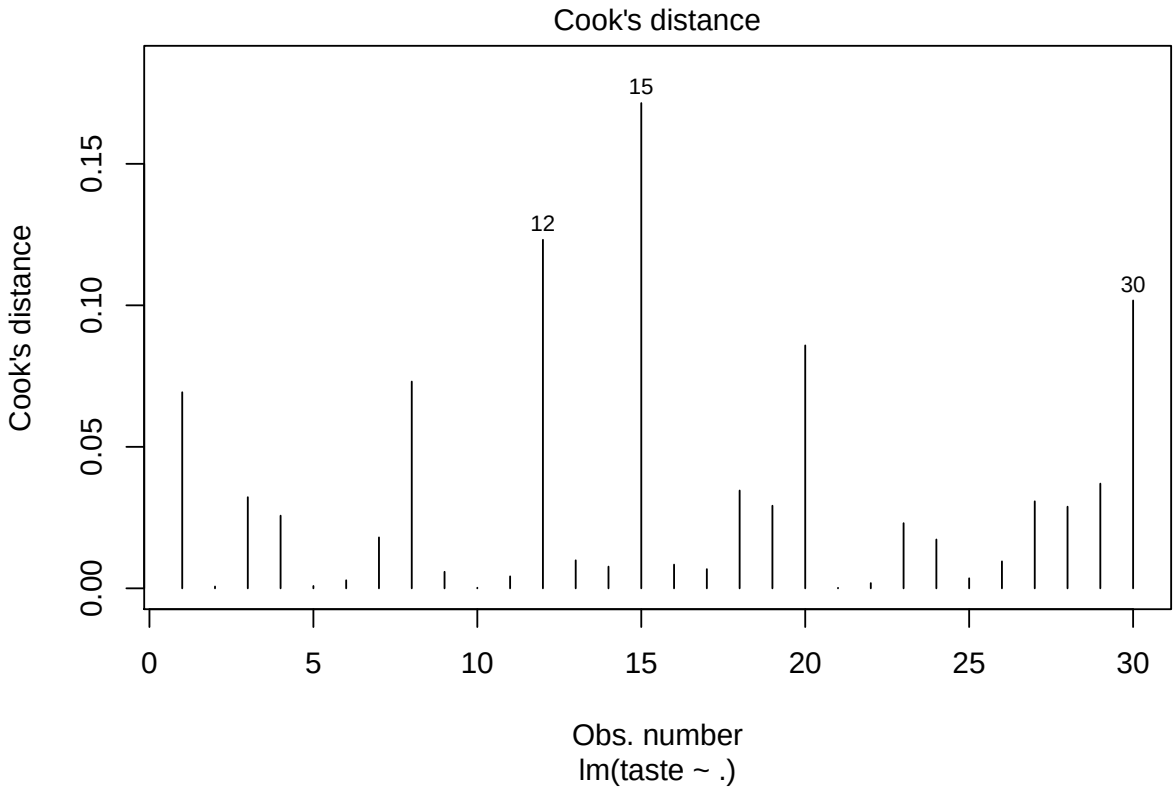
```
outlier.index <- abs(studentized.residuals) > abs(t.val.alpha)

outliers <- df[outlier.index == TRUE, ]

if (nrow(outliers) >= 1) {
    pander(outliers, caption = "outliers")
}
```

Here we look for studentized residuals that fall outside the interval given by the Bonferroni corrected t-values.

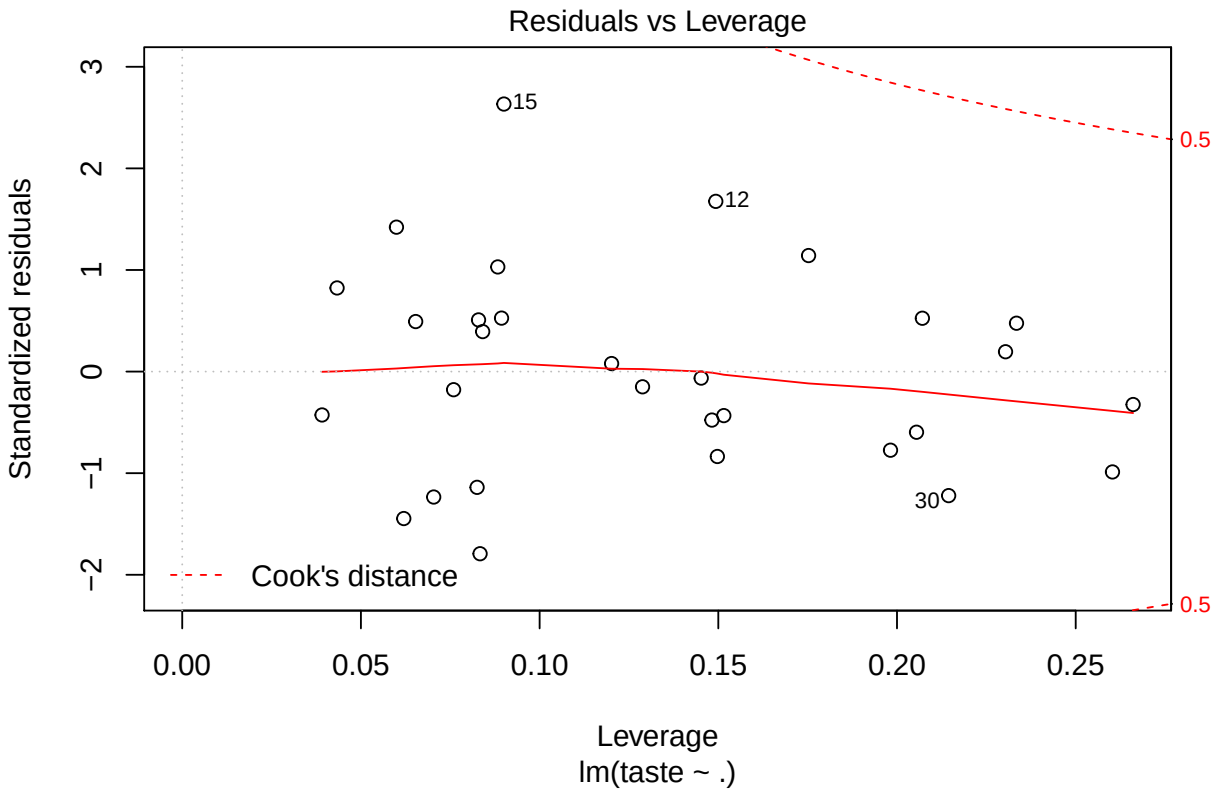
(e) Check for influential points.

We plot the Cook's distances and the residual-leverage plot with level set contours of the Cook distance.

```
plot(lm.fit, which = 4)
```



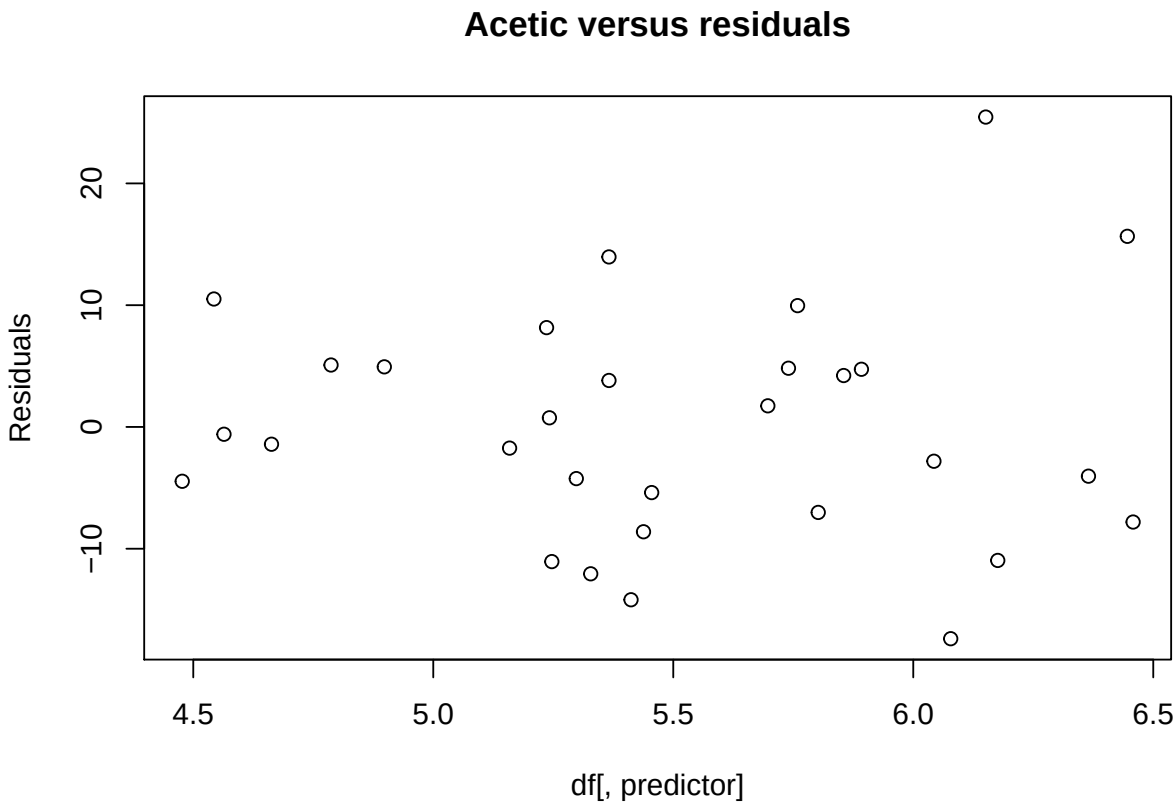

```
plot(lm.fit, which = 5)
```

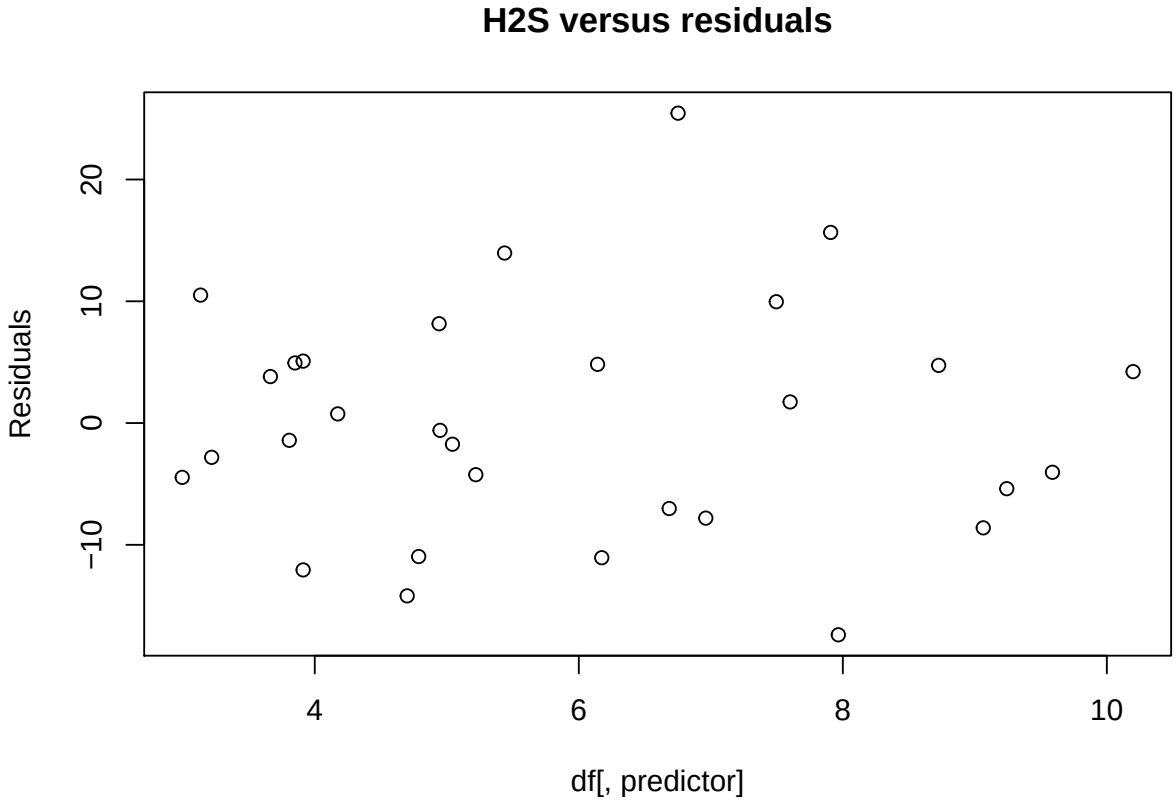


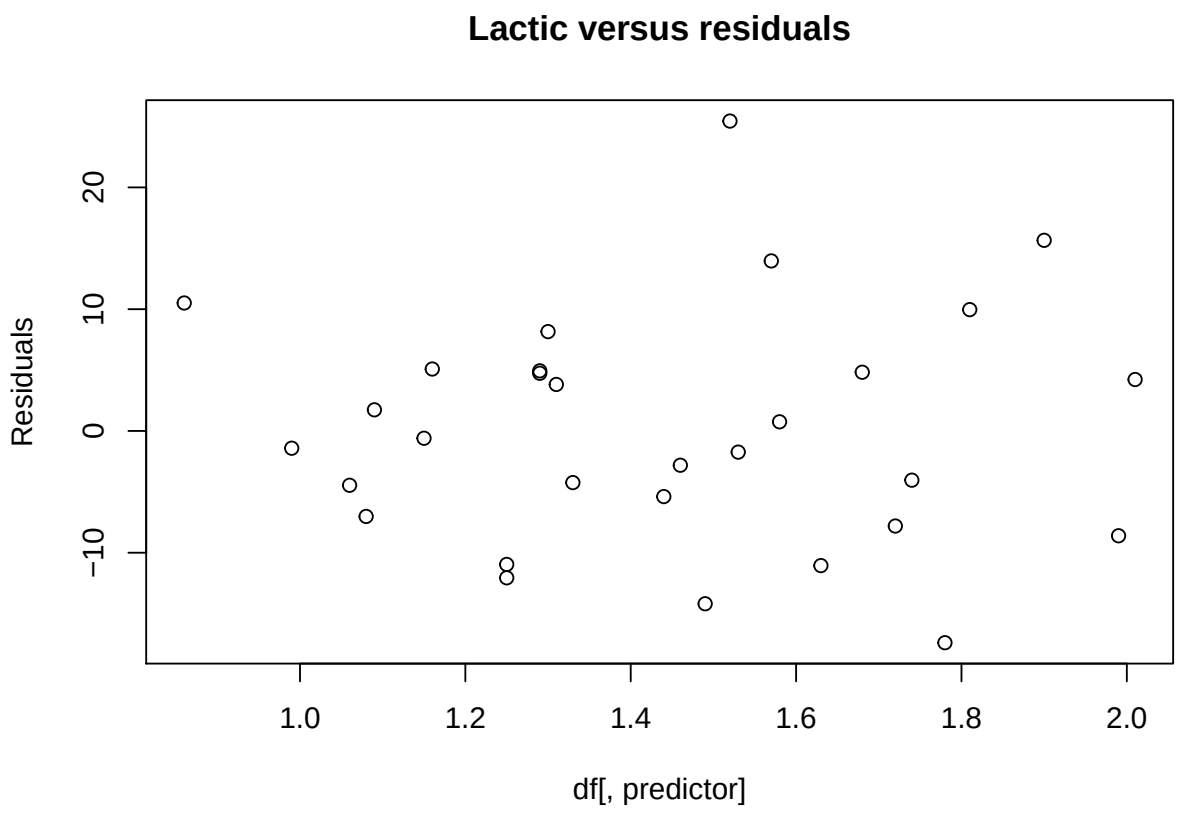
(f) Check for structure in the model.

Plot residuals versus predictors

```
predictors <- names(lm.fit$coefficients)
predictors <- predictors[2:length(predictors)]

for (i in 1:length(predictors)) {
    predictor <- predictors[i]

    plot(df[, predictor], residuals(lm.fit), xlab = , ylab = "Residuals", main = paste(p
        " versus residuals", sep = ""))

}
```


Acetic versus residuals
Residuals
20
10
0
−10
4.5
5.0
5.5
6.0
6.5
df[, predictor]

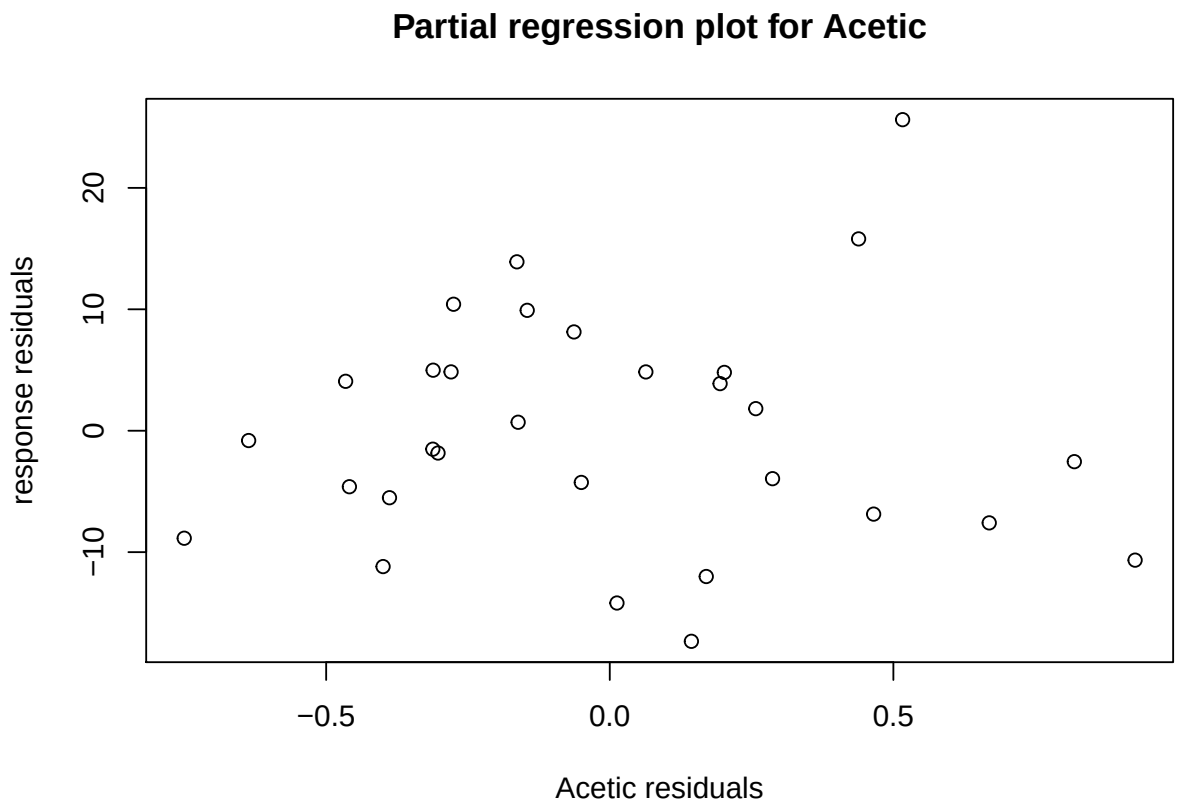

H2S versus residuals
Residuals
20
10
0
−10
4
6
8
10
df[, predictor]

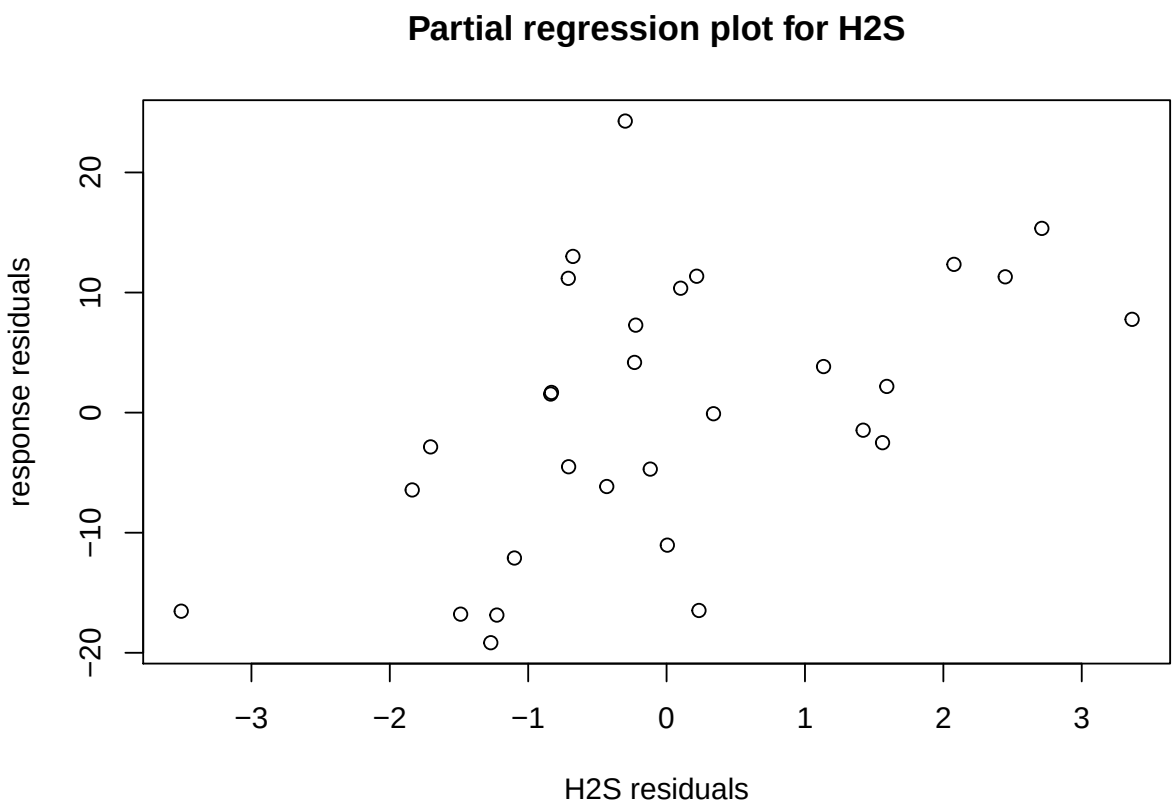


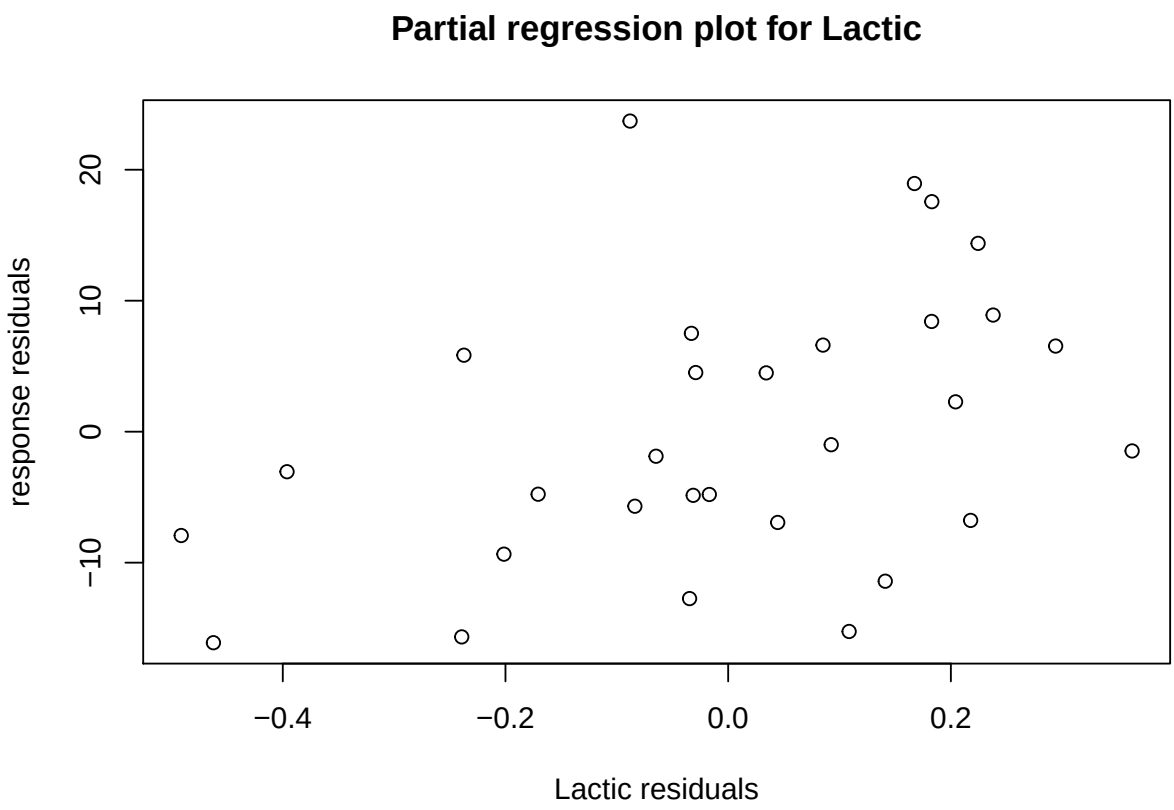
Perform partial regression

```
predictors <- names(lm.fit$coefficients)
predictors <- predictors[2:length(predictors)]

lm.formula <- formula(lm.fit)
response <- lm.formula[[2]]

for (i in 1:length(predictors)) {
    predictor <- predictors[i]
    others <- predictors[which(predictors != predictor)]
    d.formula <- paste(response, " ~ ", sep = "")
    m.formula <- paste(predictor, " ~ ", sep = "")

    for (j in 1:(length(others) - 1)) {
        d.formula <- paste(d.formula, others[j], " + ", sep = "")
        m.formula <- paste(m.formula, others[j], " + ", sep = "")
    }
    d.formula <- paste(d.formula, others[length(others)], sep = "")
    d.formula <- formula(d.formula)
```

```
    m.formula <- paste(m.formula, others[length(others)], sep = "")
    m.formula <- formula(m.formula)

    d <- residuals(lm(d.formula, df))

    m <- residuals(lm(m.formula, df))

    plot(m, d, xlab = paste(predictor, " residuals", sep = ""), ylab = "response residua
        main = paste("Partial regression plot for ", predictor, sep = ""))

}
```




Partial regression plot for H2S
response residuals
H2S residuals
20
10
0
−10
−20
−3
−2
−1
0
1
2
3


Partial regression plot for Lactic
response residuals
20
10
0
−10
−0.4
−0.2
0.0
0.2
Lactic residuals