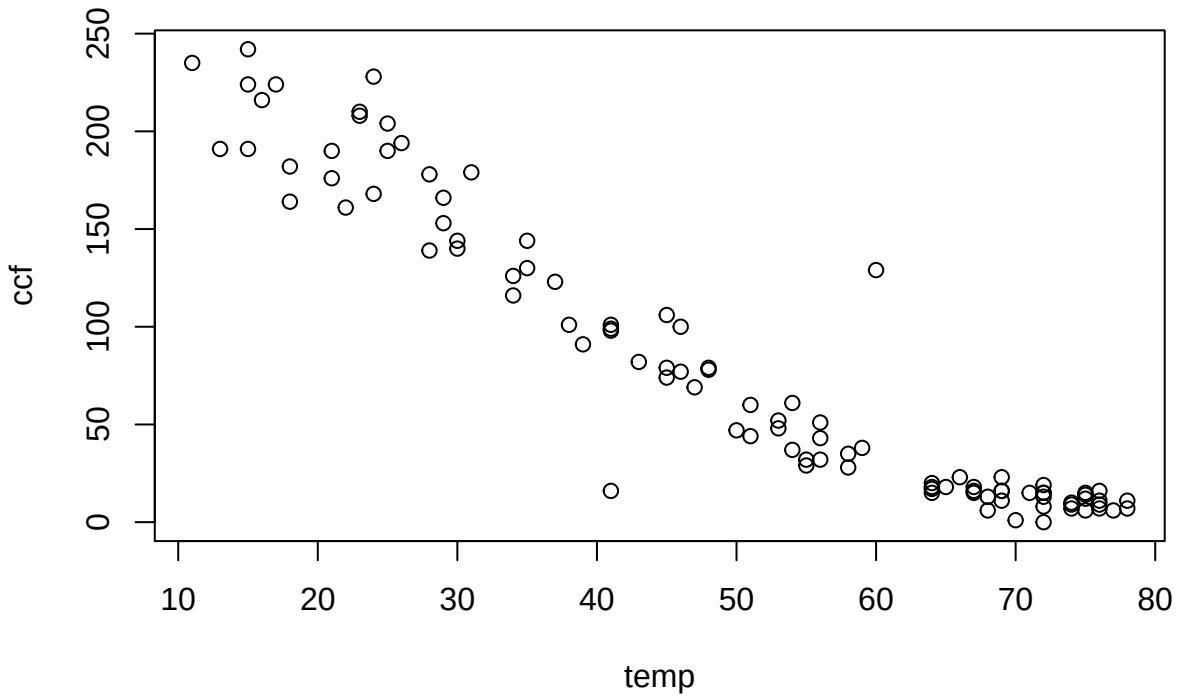
Influence, Leverag, and Outliers Observations

Return to the gas consumption data analysis

Recall the relationship between gas consumptio and outside temperature (average for the month).

```
plot(ccf ~ temp, data = ut)
```



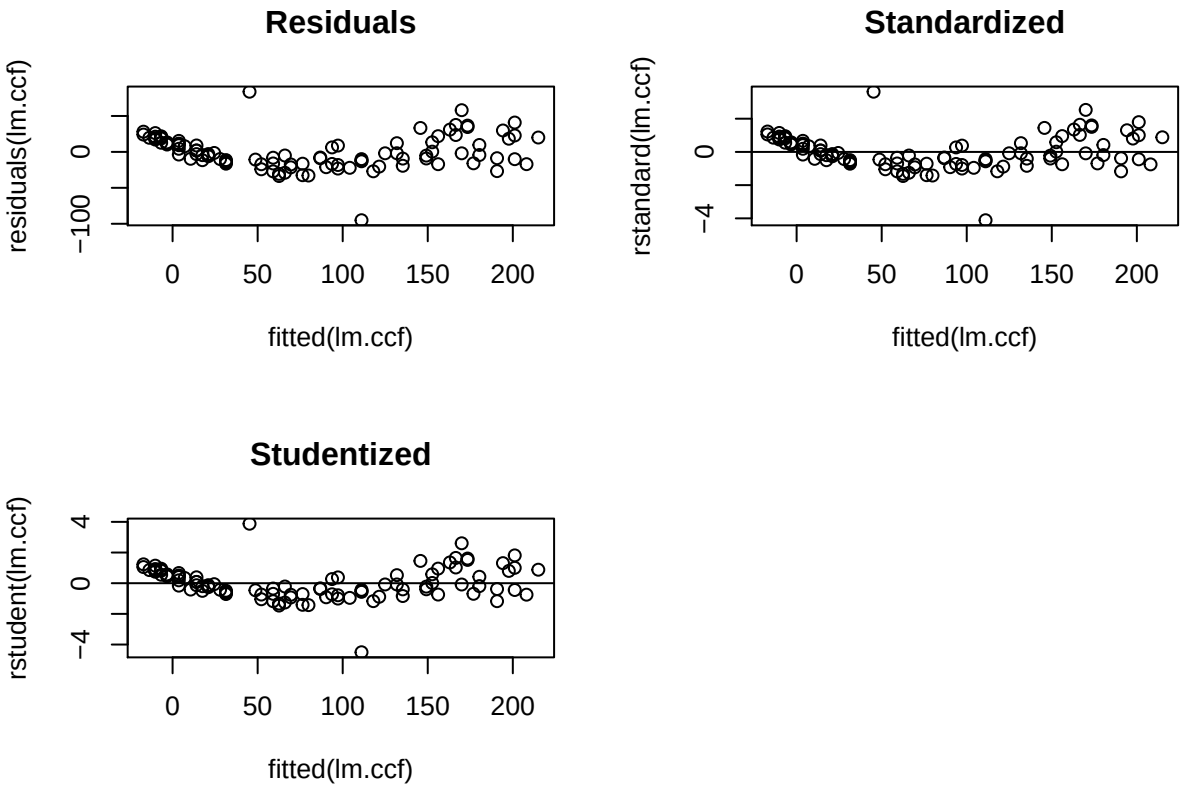
Recall that we first fit gas consumption to temperature with a simple linear regression.

```
lm.ccf = lm(ccf ~ temp, data = ut)
summary(lm.ccf)
```

```
##
## Call:
## lm(formula = ccf ~ temp, data = ut)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -95.06 -15.99  -3.67  14.35  83.76
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  253.098      6.193    40.9   <2e-16 ***
## temp          -3.464      0.115   -30.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.3 on 97 degrees of freedom
## Multiple R-squared:  0.903,  Adjusted R-squared:  0.902
## F-statistic:  900 on 1 and 97 DF,  p-value: <2e-16
```

Let's look at residual plots. We make three plots below, regular residuals, standardized residuals, and studentized residuals.

```
par(mfrow = c(2,2))
plot(residuals(lm.ccf) ~ fitted(lm.ccf), main = "Residuals")

plot(rstandard(lm.ccf) ~ fitted(lm.ccf), main = "Standardized")
abline(h=0)


plot(rstudent(lm.ccf) ~ fitted(lm.ccf), main = "Studentized")
abline(h=0)
par(mfrow = c(1, 1))
```



Which observations are potential outliers.

```
which(abs(rstandard(lm.ccf)) > 3)
```

```
## 53 54
## 53 54
```

```
which(abs(rstudent(lm.ccf)) > 3)
```

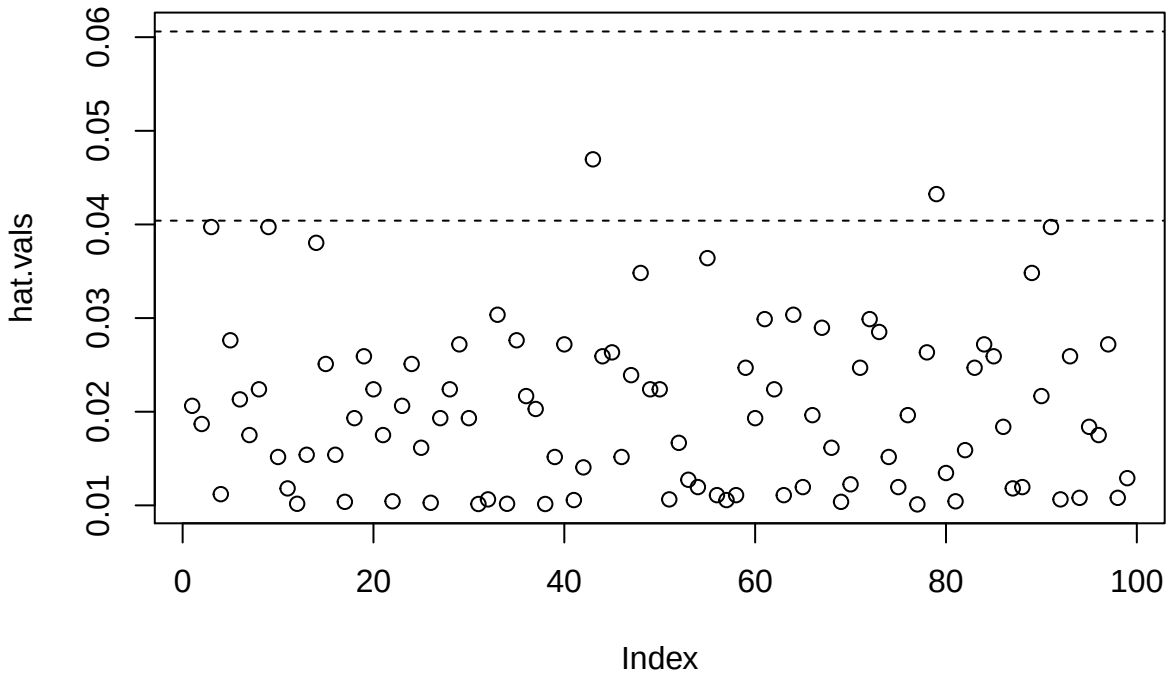
```
## 53 54
## 53 54
```

```
ut$notes[which(abs(rstudent(lm.ccf)) > 3)]
```

```
## [1]                    bad meter reading
## 8 Levels:  5.46 credit for "cost of gas" ... transfer back from England
```

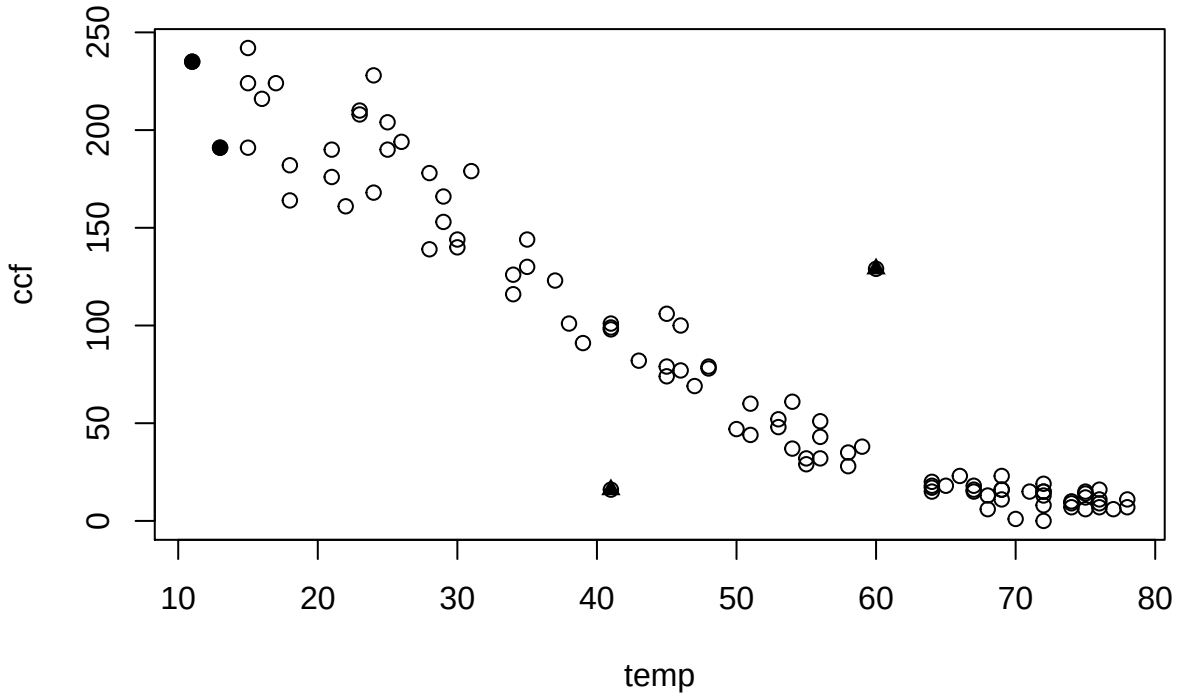
We can also investigate the values along the diagonal of the hat matrix. These can assess leverage. It assesses the contribution of an observation to all of the fitted values. The response variable values are not involved in determining leverage.

```
hat.vals = hatvalues(lm.ccf)

dfModel = 2
dfErr = nrow(ut) - dfModel
hbar = dfModel/nrow(ut)

plot(hat.vals, ylim = c(min(hat.vals), max(3*hbar, hat.vals)))
abline(h = c(2*hbar, 3*hbar), lty = 2)
```



Let's see where these leverage points and outliers are on the original plot.

```
plot(ccf ~ temp, data = ut)
points(ut[which(hat.vals > 2*hbar), c("temp", "ccf")], pch = 19)
points(ut[which(abs(rstudent(lm.ccf)) > 3), c("temp", "ccf")], pch = 17)
```



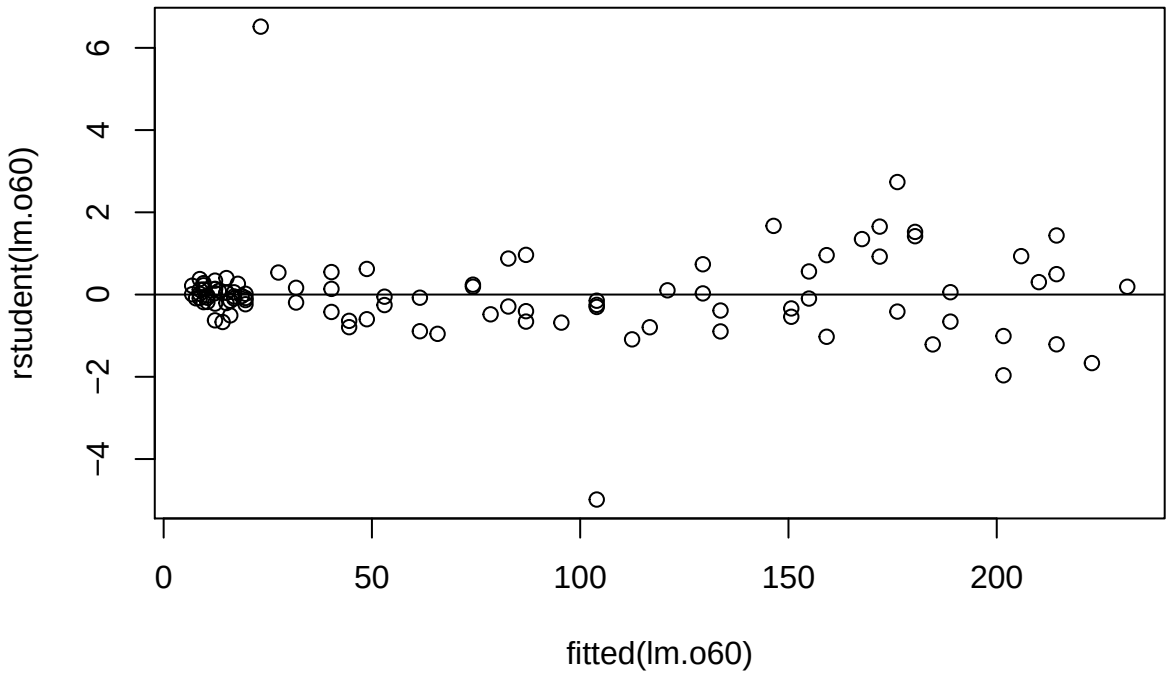
It makes sense to address the structure that is in the residuals. we saw earlier that we can add a variable for the over 60 temperatures.

```
ut$tempOver60 = pmax(0, ut$temp - 60)
lm.o60 = lm(ccf ~ temp + tempOver60, data = ut)
summary(lm.o60)
```

```
##
## Call:
## lm(formula = ccf ~ temp + tempOver60, data = ut)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -87.97  -8.07  -0.91   4.96 105.70
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  278.042      6.699   41.51  < 2e-16 ***
## temp          -4.246      0.162  -26.18  < 2e-16 ***
## tempOver60     3.332      0.549    6.07  2.5e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.9 on 96 degrees of freedom
## Multiple R-squared:  0.93,   Adjusted R-squared:  0.928
## F-statistic:  635 on 2 and 96 DF,  p-value: <2e-16
```

Now let's examine the residuals again:

```
plot(rstudent(lm.o60) ~ fitted(lm.o60))
abline(h=0)
```



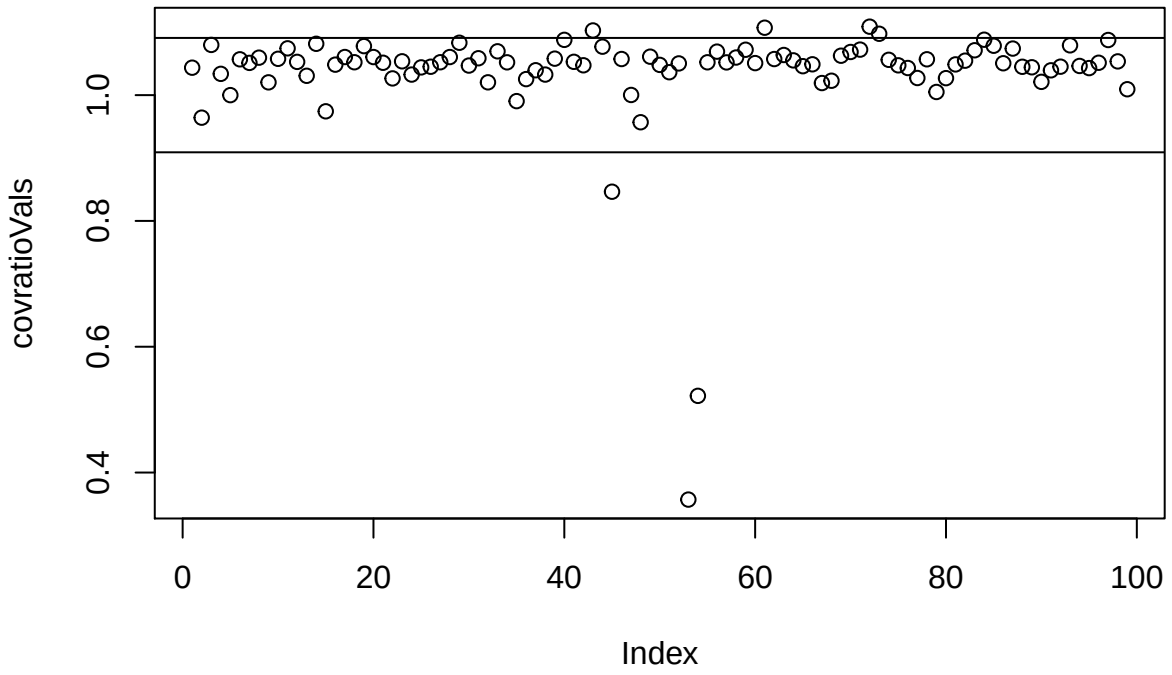

```
checkObs = which(abs(rstudent(lm.o60)) > 3)
checkObs
```

```
## 53 54
## 53 54
```

We still have the same two outliers. Also it appears that the variability changes with the amount of gas consumed. We will address this issue later.

Another measure is the influence on the standard errors of the coefficients.

```
dfModel = 3
covratioVals = covratio(lm.o60)
plot(covratioVals)
covratioCO = (3*dfModel/nrow(ut))
abline(h = c(1 - covratioCO, 1 + covratioCO))
```




```
which(abs(covratioVals - 1) > 2*covratioCO)
```

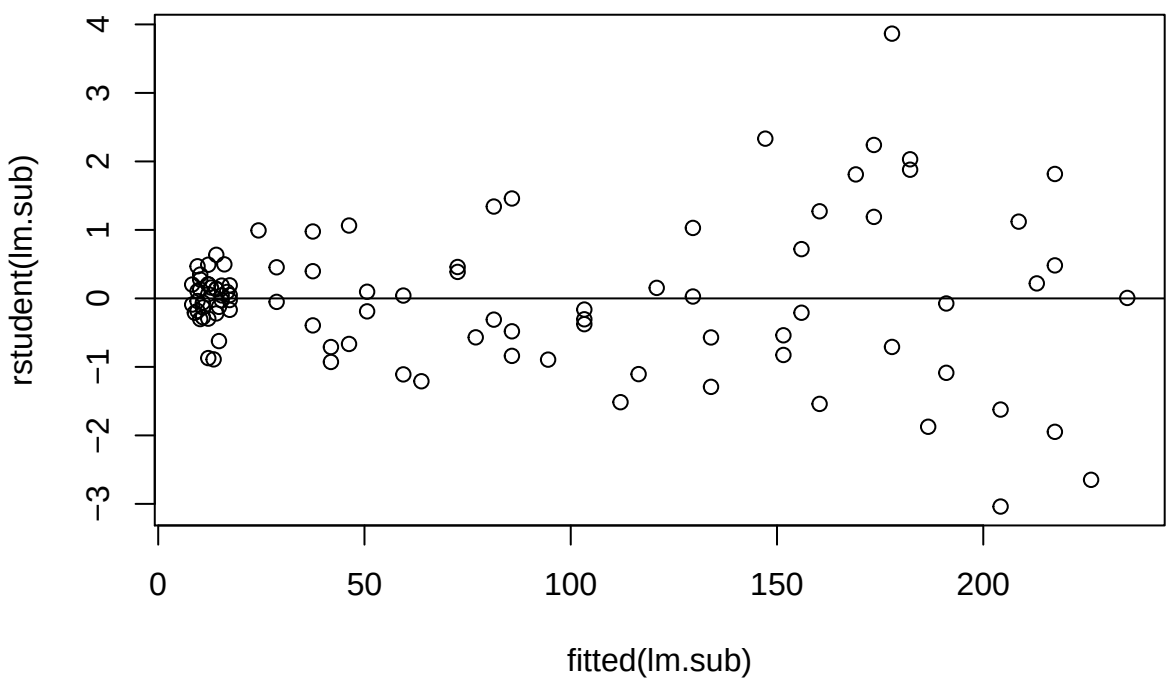
```
## 53 54
## 53 54
```

Next we remove the two outliers, and refit the model.

```
utSub = ut[-checkObs, ]

lm.sub = lm(ccf ~ temp + tempOver60, data = utSub)
summary(lm.sub)
```

```
##
## Call:
## lm(formula = ccf ~ temp + tempOver60, data = utSub)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -40.20  -7.55  -0.33   5.52  50.13
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  283.167      4.791   59.10  < 2e-16 ***
## temp          -4.387      0.117  -37.51  < 2e-16 ***
## tempOver60     3.737      0.397    9.42  3.2e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.1 on 94 degrees of freedom
## Multiple R-squared:  0.965,  Adjusted R-squared:  0.964
## F-statistic: 1.29e+03 on 2 and 94 DF,  p-value: <2e-16
```

```
plot(rstudent(lm.sub) ~ fitted(lm.sub))
abline(h=0)
```




```
par(mfrow = c(1 ,1))
utSub$notes[ which(abs(rstudent(lm.sub)) > 3) ]
```

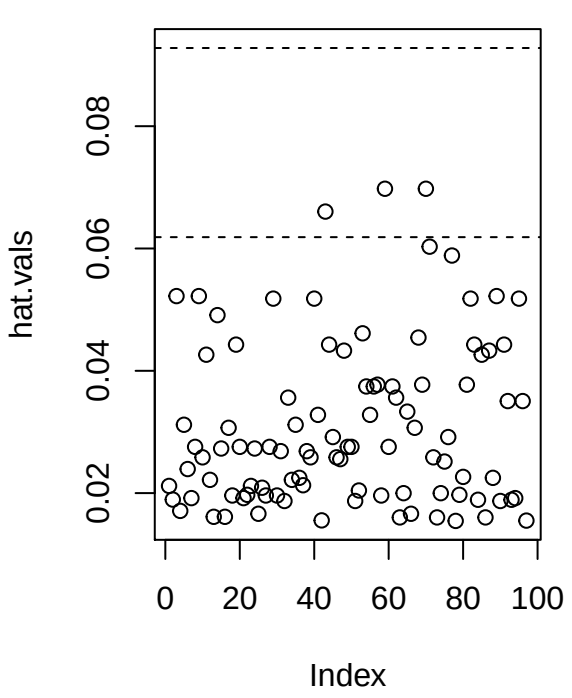
```
## [1]
## 8 Levels:  5.46 credit for "cost of gas" ... transfer back from England
```

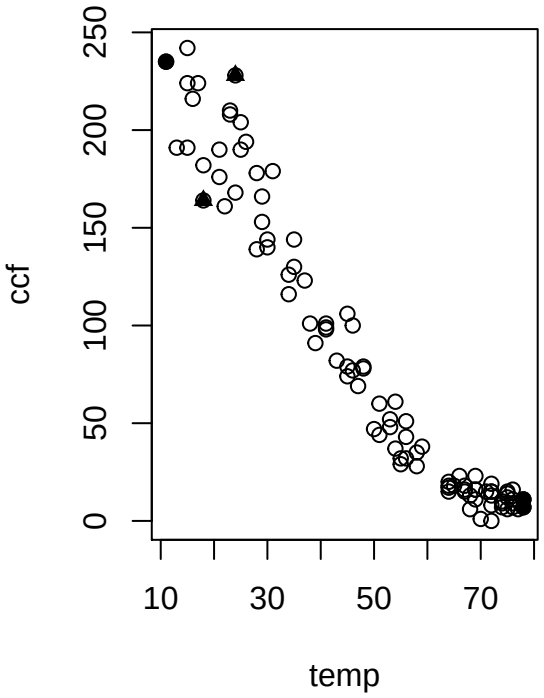
Compare the coefficients for temp before and after the removal of the outliers.

Compare the residual SD before and after the removal of the two outliers.

Let's look again at the hat values.

```
dfModel = 3
dfErr = nrow(utSub) - dfModel

hbar = dfModel/nrow(utSub)
hat.vals = hatvalues(lm.sub)

par(mfrow = c(1,2))
plot(hat.vals, ylim = c(min(hat.vals), max(3*hbar, hat.vals)))
abline(h = c(2*hbar, 3*hbar), lty = 2)

plot(ccf ~ temp, data = utSub)
outPts = which(abs(rstudent(lm.sub)) > 3)
points(utSub[ outPts, c("temp", "ccf")], pch = 17)

hatPts = which(hat.vals > 2*hbar)
points(utSub[ hatPts, c("temp", "ccf")], pch = 19)
```

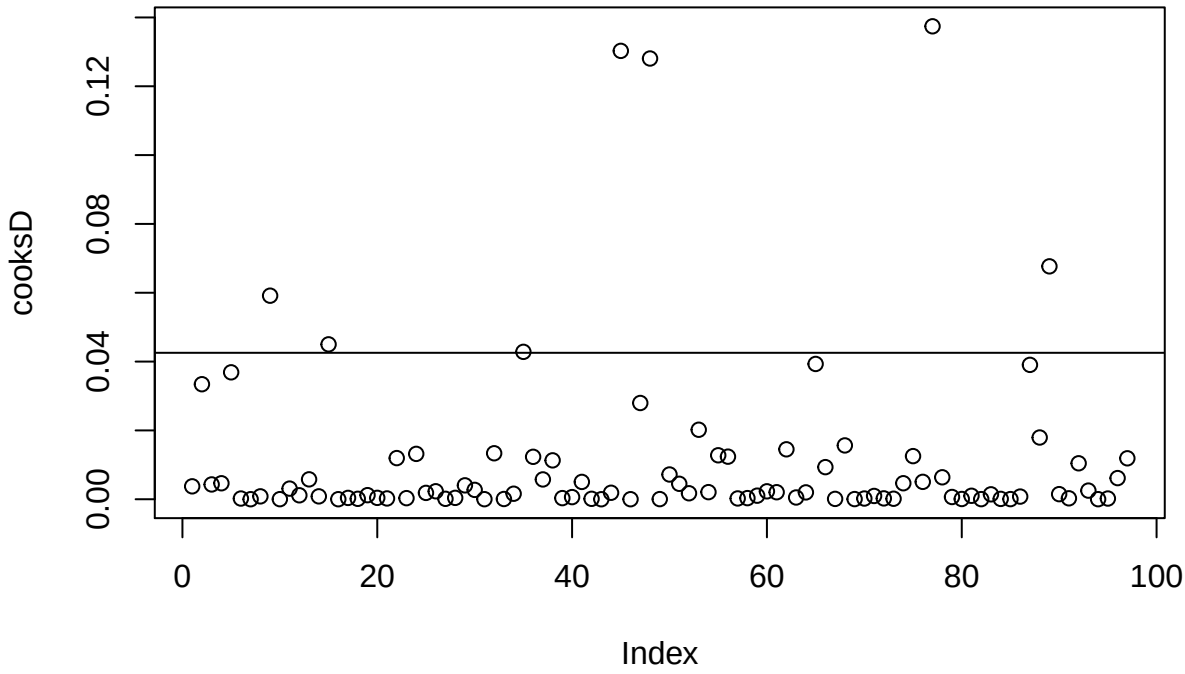





```
par(mfrow = c(1 ,1))
```

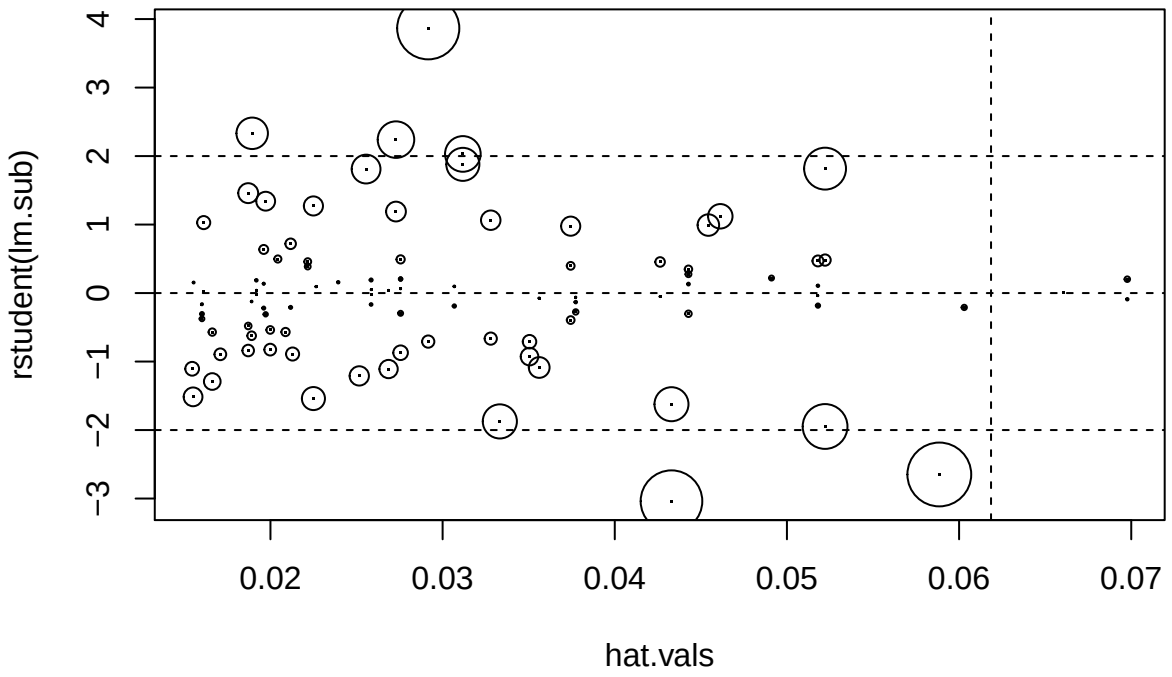
We use Cook's distance to look for influential points. Influence is a combination of leverage and discrepancy (outlyingness). This is influence on the regression coefficients. Let's look for influential observtions in the gas data.

```
cooksD = cooks.distance(lm.sub)
plot(cooksD)
abline(h = 4/dfErr)
```

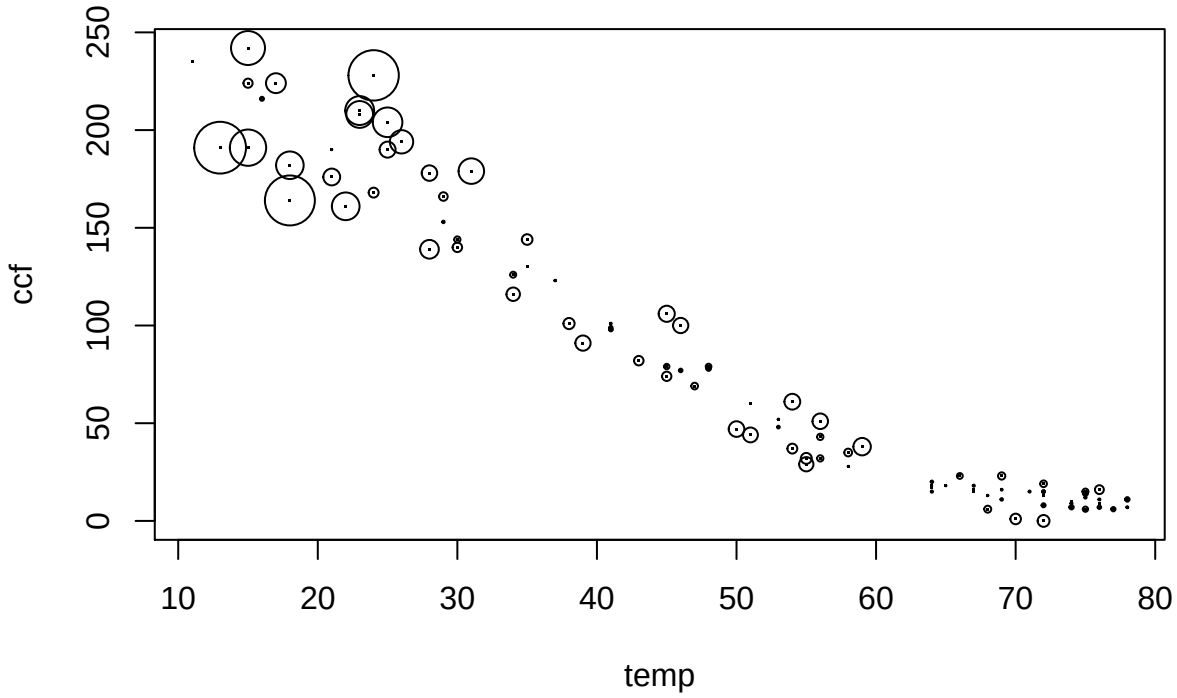


Another plot that people find useful is the bubble plot.

```
plot(rstudent(lm.sub) ~ hat.vals, pch = ".")
abline(v = c(2*hbar, 3*hbar), lty = 2)
abline(h = c(-2, 0, 2), lty = 2)
symbols(y = rstudent(lm.sub), x = hat.vals,
        circles = sqrt(cooksD)/200, inches = FALSE,
        add = TRUE)
```




```
plot(ccf ~ temp, data = utSub, pch = ".")
symbols(y = utSub$ccf, x = utSub$temp,
        circles = 5*sqrt(cooksD), inches = FALSE,
        add = TRUE)
```


ccf
0
50
100
150
200
250
10
20
30
40
50
60
70
80
temp