Solution to Series 5

1. a) From the three R formulae we can derive the following:

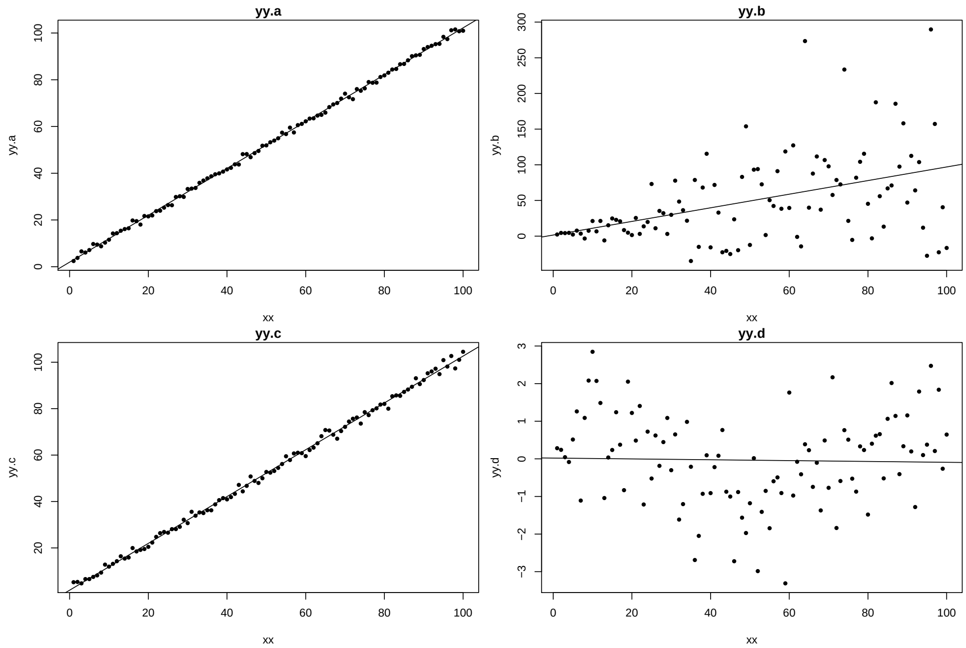
.a Model assumptions valid.

.b Model contains strong non-constant variance.

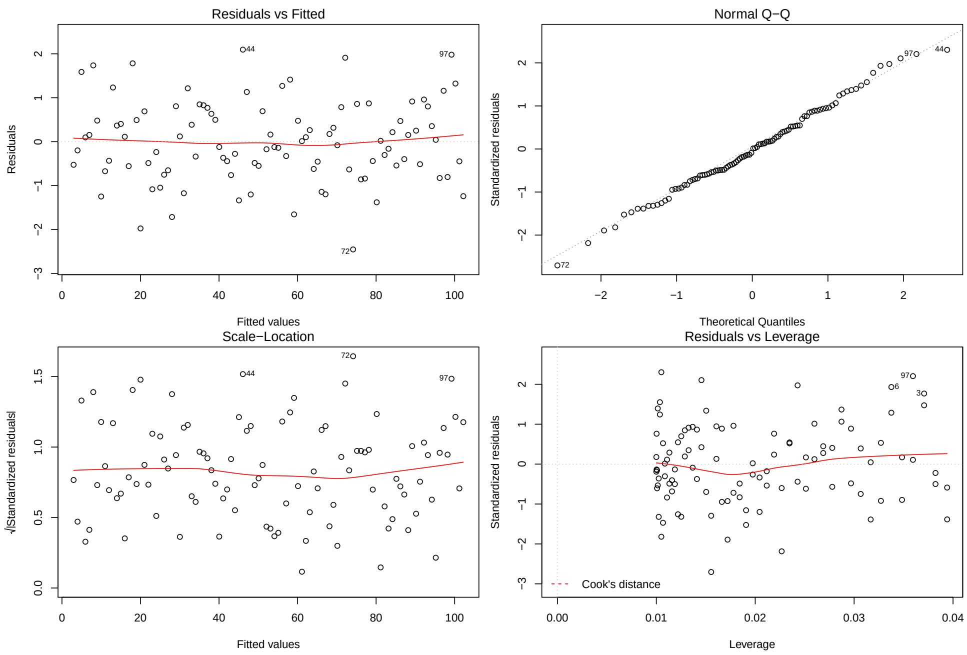
.c Variance slightly non-constant.

.d Non-linear model.

b)
```
> set.seed(123) #To make data reproducible.
> n <- 100
> xx <- 1:n
> yy.a <- 2+1*xx+rnorm(n)
> yy.b <- 2+1*xx+rnorm(n)*(xx)
> yy.c <- 2+1*xx+rnorm(n)*(1+xx/n)
> yy.d <- cos(xx*pi/(n/2)) + rnorm(n)
> par(mfrow=c(2,2))
> fit.a <- lm(yy.a ~ xx)
> plot(xx, yy.a, main="yy.a", pch=20)
> abline(fit.a)
> fit.b <- lm(yy.b ~ xx)
> plot(xx, yy.b, main="yy.b", pch=20)
> abline(fit.b)
> fit.c <- lm(yy.c ~ xx)
> plot(xx, yy.c, main="yy.c", pch=20)
> abline(fit.c)
> fit.d <- lm(yy.d ~ xx)
> plot(xx, yy.d, main="yy.d", pch=20)
> abline(fit.d)
```



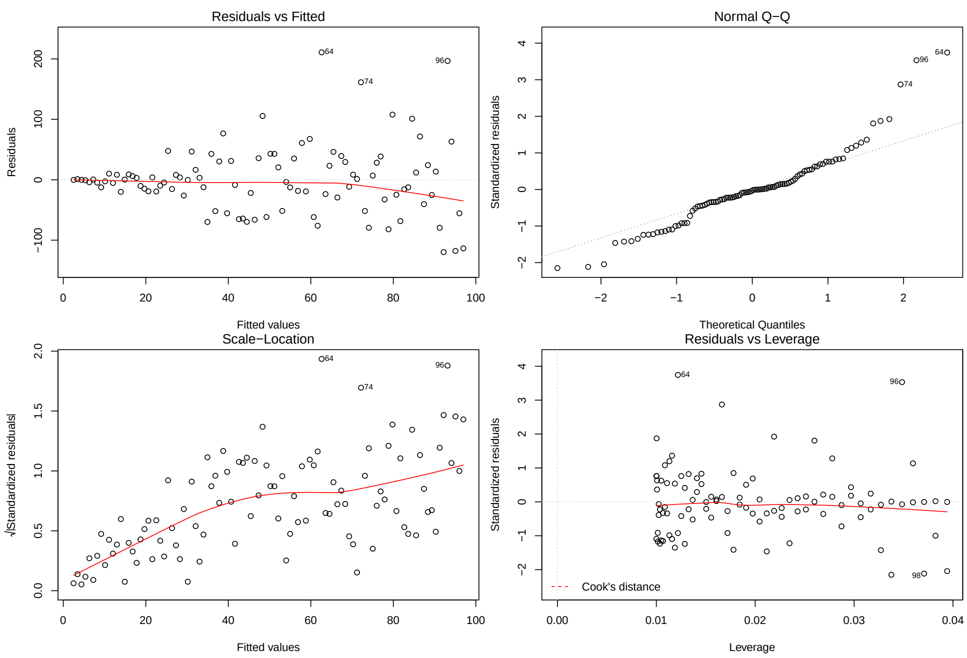
c) `Model diagnostics yy.a`
```
> par(mfrow=c(2,2))
> plot(fit.a)
```



`yy.a`: From the Residuals vs Fitted (Tukey-Anscombe) and Scale-Location plots we conclude that the constant variance of the errors assumption is satisfied. Moreover, looking at the Tukey-Anscombe plot, we see that neither the zero-expectation of the errors nor the uncorrelated errors assumptions are violated (the red line seems to be close to the x-axis and we cannot identify a non-random structure in the data). Furthermore, the Q-Q plot does not show strong evidence against the normality assumption.

`Model diagnostics yy.b`

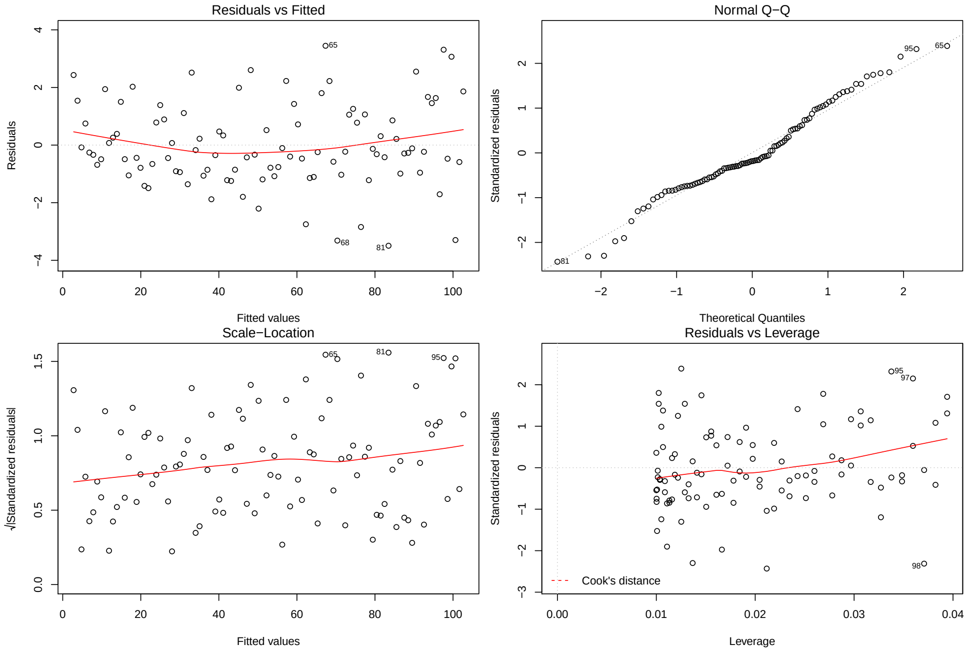
```
> par(mfrow=c(2,2))
> plot(fit.b)
```



`yy.b`: The Tukey-Anscombe and Scale-Location plots show residuals with strong non-constant variance: the residuals are bigger for larger fitted values. From the Tukey-Anscombe plot, we conclude that the zero-expectation and uncorrelated errors assumption are satisfied. The Q-Q plot provide evidence against the normality assumption, which is what we would expect if we look at the model equation.

`Model diagnostics yy.c`

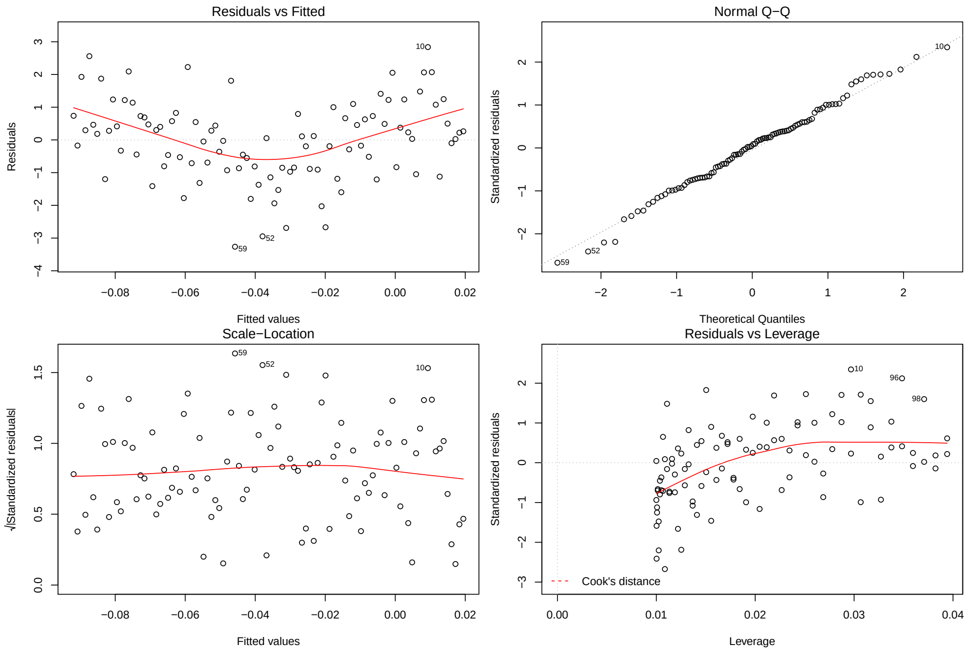
```
> par(mfrow=c(2,2))
> plot(fit.c)
```



yy.c: The Tukey-Anscombe plot again show evidence against the non-constant variance assumption. However, it is less accentuated than in the previous example because the residuals have smaller values than in fit.b. From this plot, however, we can see that the zero-expectation and uncorrelated errors assumption are satisfied. From the Q-Q plot, we conclude that the normality assumption is slightly violated as we would expect by looking at the model equation.

Model diagnostics yy.d

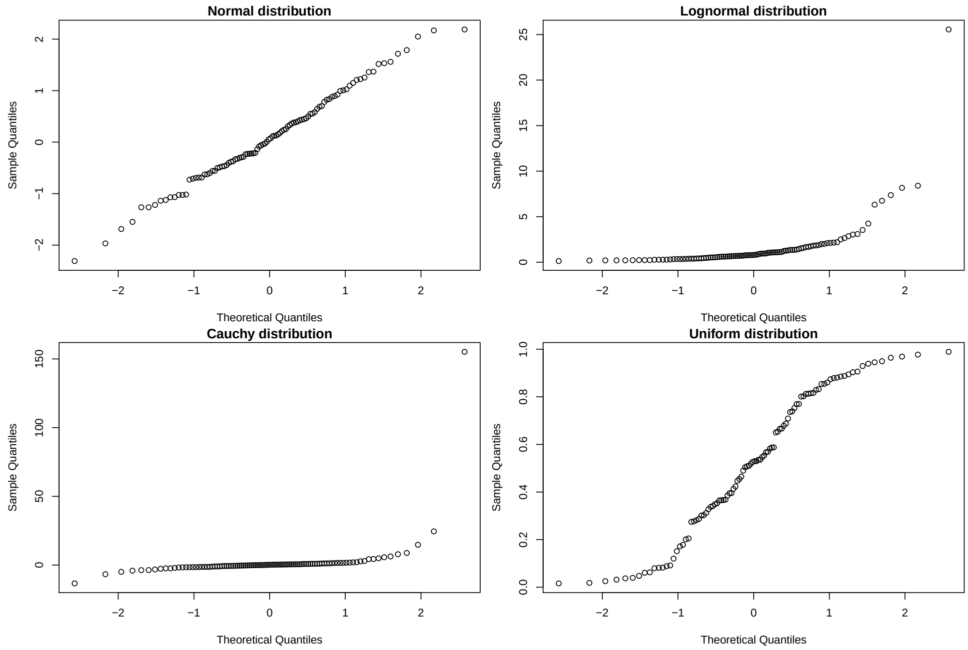
```
> par(mfrow=c(2,2))
> plot(fit.d)
```



yy.d: From the Tukey-Anscombe plot we can see that this model is clearly non-linear since it exhibits a U-shaped pattern. Therefore, we can conclude the existence of a non-linear relation between response and predictor. From the Scale-Location, Tukey-Anscombe, and Q-Q plots , we cannot see strong evidence against the assumptions of constant variance, normality and uncorrelated errors.

d) The exercise should be repeated generating new random numbers (remember to change the argument of `set.seed` or just eliminate it). Manipulating the number of observations is also instructive. However, the above described structures are of general nature and will largely remain on the repetitions.

e)
```
> par(mfrow=c(2,2))
> set.seed(123)
> qqnorm(rnorm(n), main=c("Normal distribution"))
> qqnorm(exp(rnorm(n)), main=c("Lognormal distribution"))
> qqnorm(rcauchy(n), main=c("Cauchy distribution"))
> qqnorm(runif(n), main=c("Uniform distribution"))
```



`Normal distribution` The sample quantiles fit nicely to the theoretical quantiles of a normal distribution. Deviations from the diagonal line are to be expected due to randomness.
`Lognormal distribution` The curve is bent upwards. This indicates a positively skewed distribution of the sample points.
`Cauchy distribution` The distribution of the data seems to be fairly symmetric. However, the curve has the shape of an inverted S which indicates that this distribution has heavier tails than those of a Normal distribution.
`Uniform distribution` We have the opposite case of the Cauchy distribution. Here, the curve is S-shaped and we conclude that the distribution of this sample has shorter tails than those of a normal distribution.

f) Repeat the exercise generating new random numbers (remember to change the argument of `set.seed` or eliminate it) and varying the number of observations as well.

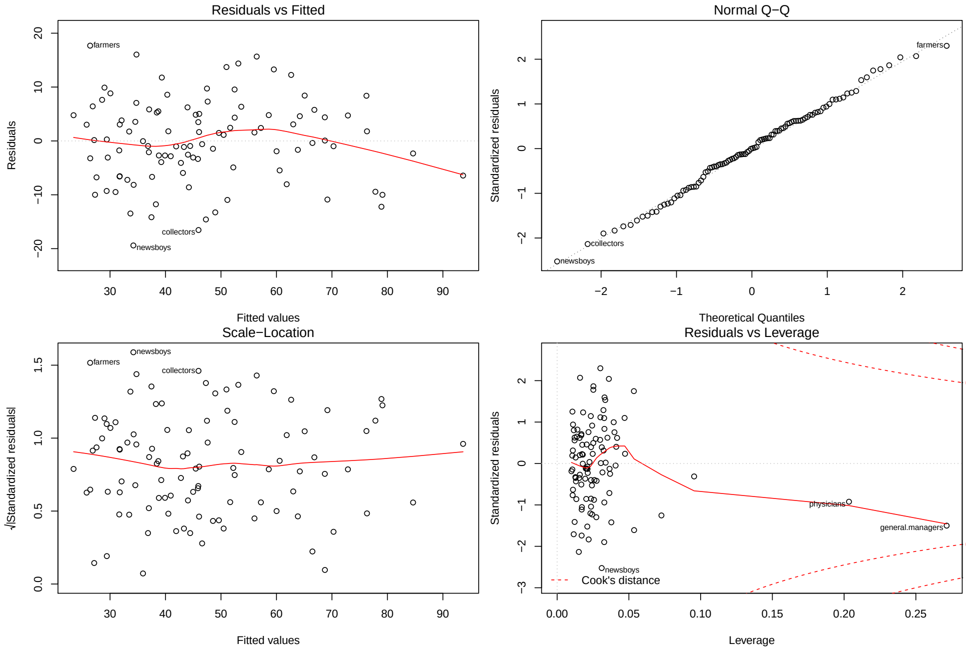
2. a) Partial residual plots

```
> library(car)
> data(Prestige)
> fit00 <- lm(prestige ~ income + education, data=Prestige)
> summary(fit00)
```

```
Call:
lm(formula = prestige ~ income + education, data = Prestige)

Residuals:
     Min       1Q   Median       3Q      Max
-19.4040  -5.3308   0.0154   4.9803  17.6889

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) -6.8477787  3.2189771  -2.127   0.0359 *  
income       0.0013612  0.0002242   6.071 2.36e-08 ***
education    4.1374444  0.3489120  11.858  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.81 on 99 degrees of freedom
Multiple R-squared: 0.798,	Adjusted R-squared: 0.7939 
F-statistic: 195.6 on 2 and 99 DF,  p-value: < 2.2e-16
```

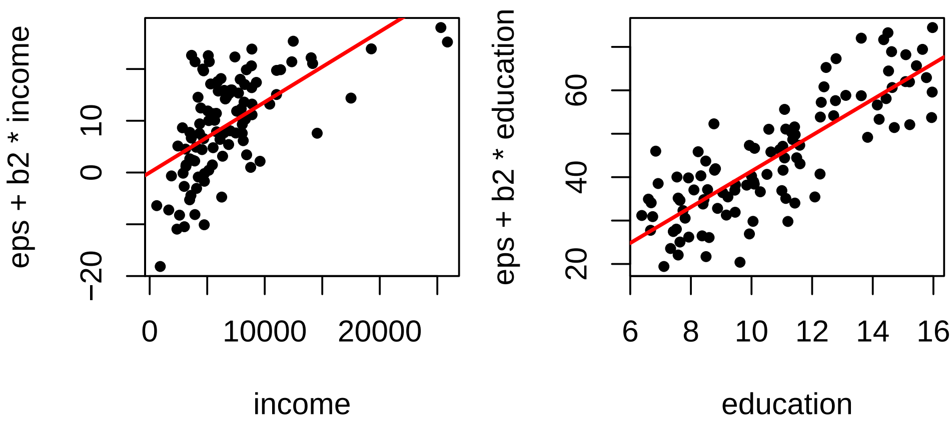
```
> par(mfrow=c(2,2))
> plot(fit00)
```



Already, this model fits well. Global F-test and the two predictors are highly significant. Diagnostic plots look reasonable. We can see some deviation of the smoother from the x-axis in the Tukey-Anscombe plot. Physicians and General Managers seem to be leverage points. However, since both do not have large residuals nor Cook's distances there is no reason to worry.

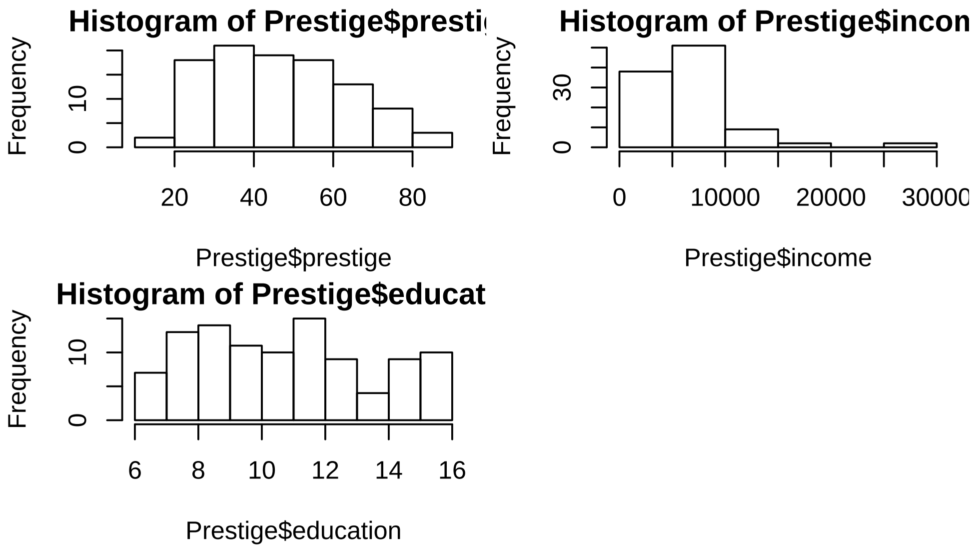
We now look at the partial residual plots:

```
> par(mfrow = c(1,2))
> plot(Prestige$income, resid(fit00)+coef(fit00)[2]*Prestige$income,
       xlab="income", ylab="eps + b2 * income", pch=20)
> abline(0, coef(fit00)[2], lwd=2, col="red")
> plot(Prestige$education, resid(fit00)+coef(fit00)[3]*Prestige$education,
       xlab="education", ylab="eps + b2 * education", pch=20)
> abline(0, coef(fit00)[3], lwd=2, col="red")
```



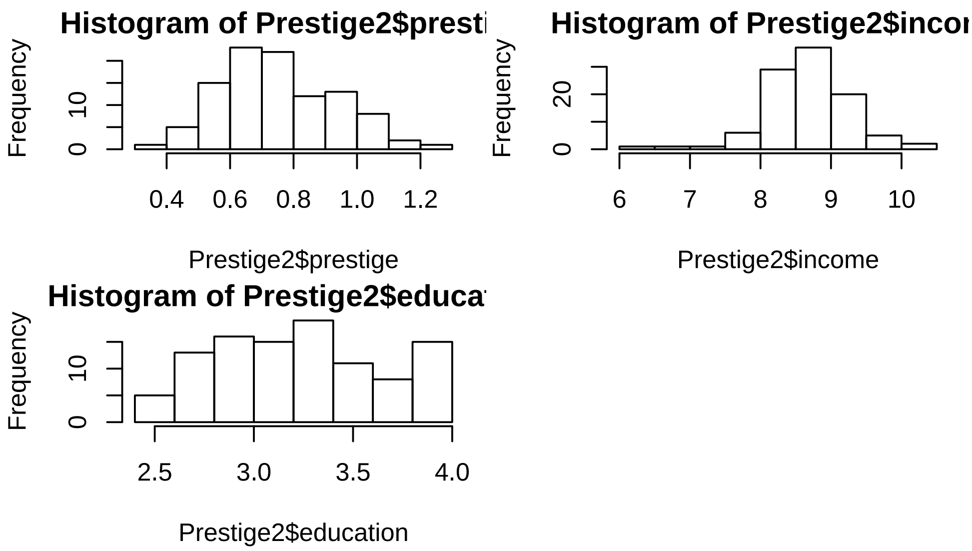
From these plots we see the influence of each predictor on the response in the presence of the other predictors. The dependence is clearly visible in both plots, even though a nonlinear (e.g. logarithmic) relation might be more appropriate for `income`.
To decide on transformations we first look at the histograms.

```
> par(mfrow=c(2,2))
> hist(Prestige$prestige)
> hist(Prestige$income)
> hist(Prestige$education)
```



We apply an arcsin transformation to `prestige`, as it is a proportion, a log transformation to `income`, as it is right-skewed, and a square-root transformation to `education`, as it is a count (number of years).

```
> Prestige2 <- Prestige
> Prestige2$prestige <- asin(sqrt(Prestige$prestige/100))
> Prestige2$income <- log(Prestige$income)
> Prestige2$education <- sqrt(Prestige$education)
> par(mfrow=c(2,2))
> hist(Prestige2$prestige)
> hist(Prestige2$income)
> hist(Prestige2$education)
```

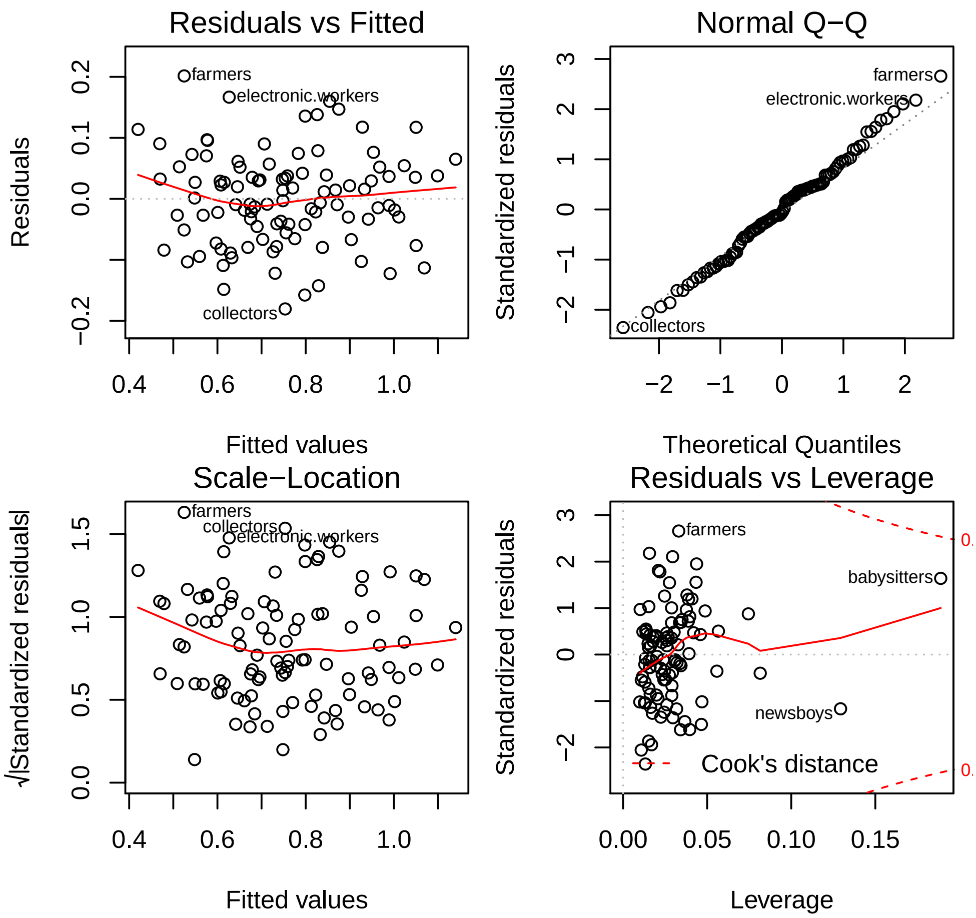


Now the model looks like this:

```
> fit01 <- lm(prestige ~ income + education, data=Prestige2)
> summary(fit01)
```

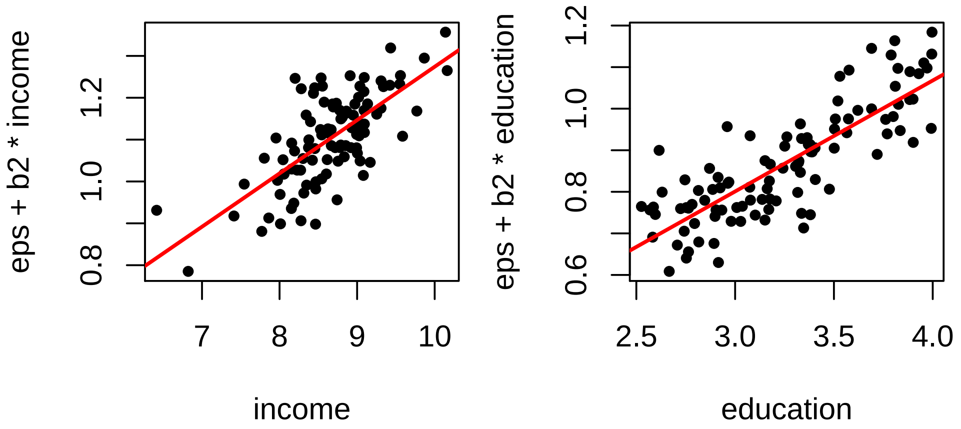
```
Call:
lm(formula = prestige ~ income + education, data = Prestige2)

Residuals:
      Min        1Q    Median        3Q       Max
-0.180399 -0.049532 -0.004739  0.041214  0.201576

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.21832    0.11258 -10.822  < 2e-16 ***
income       0.12741    0.01538   8.286 5.82e-13 ***
education    0.26695    0.02193  12.175  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.07709 on 99 degrees of freedom
Multiple R-squared: 0.822,        Adjusted R-squared: 0.8184
F-statistic: 228.6 on 2 and 99 DF,  p-value: < 2.2e-16
```

```
> par(mfrow=c(2,2))
> plot(fit01)
```



And the partial residual plots are:

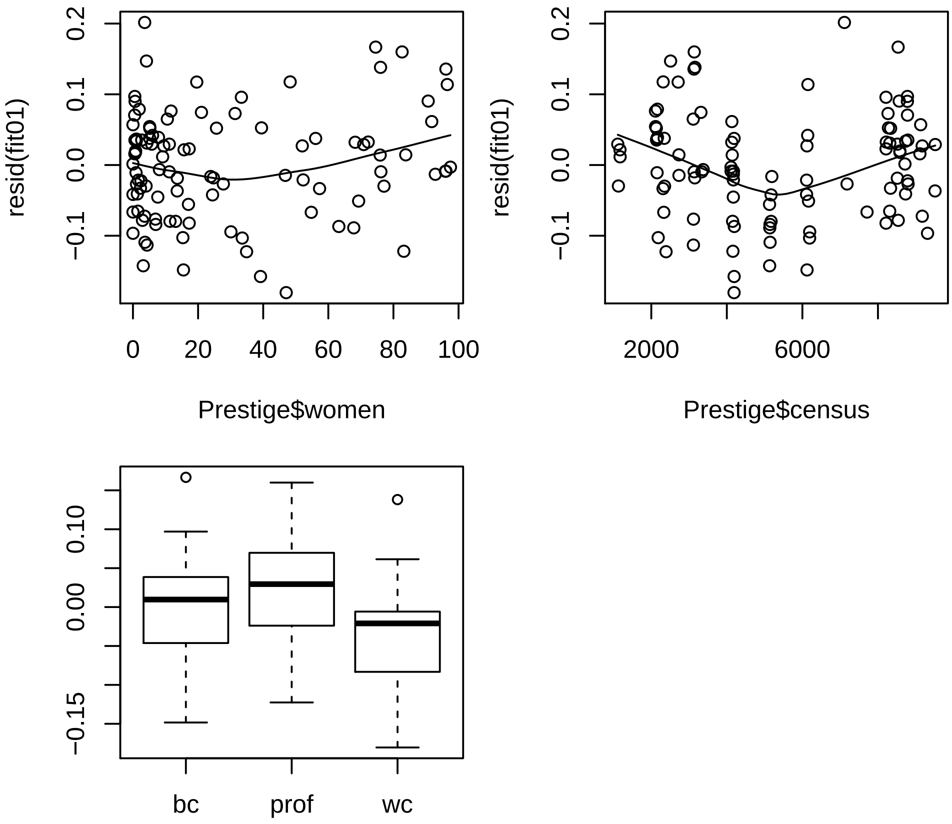
```
> par(mfrow = c(1,2))
> plot(Prestige2$income, resid(fit01)+coef(fit01)[2]*Prestige2$income,
      xlab="income", ylab="eps + b2 * income", pch=20)
> abline(0, coef(fit01)[2], lwd=2, col="red")
> plot(Prestige2$education, resid(fit01)+coef(fit01)[3]*Prestige2$education,
      xlab="education", ylab="eps + b2 * education", pch=20)
> abline(0, coef(fit01)[3], lwd=2, col="red")
```



The adjusted R-squared has increased slightly and the diagnostic plots still look fine. From the partial residual plots we see that a linear relation looks alot more plausible now.

We now see which of the other variables in the data set could explain the remaining variance, so we plot them against the residuals:

```
> par(mfrow=c(2,2))
> scatter.smooth(resid(fit01) ~ Prestige$women)
> scatter.smooth(resid(fit01) ~ Prestige$census)
> boxplot(resid(fit01) ~ Prestige$type)
```

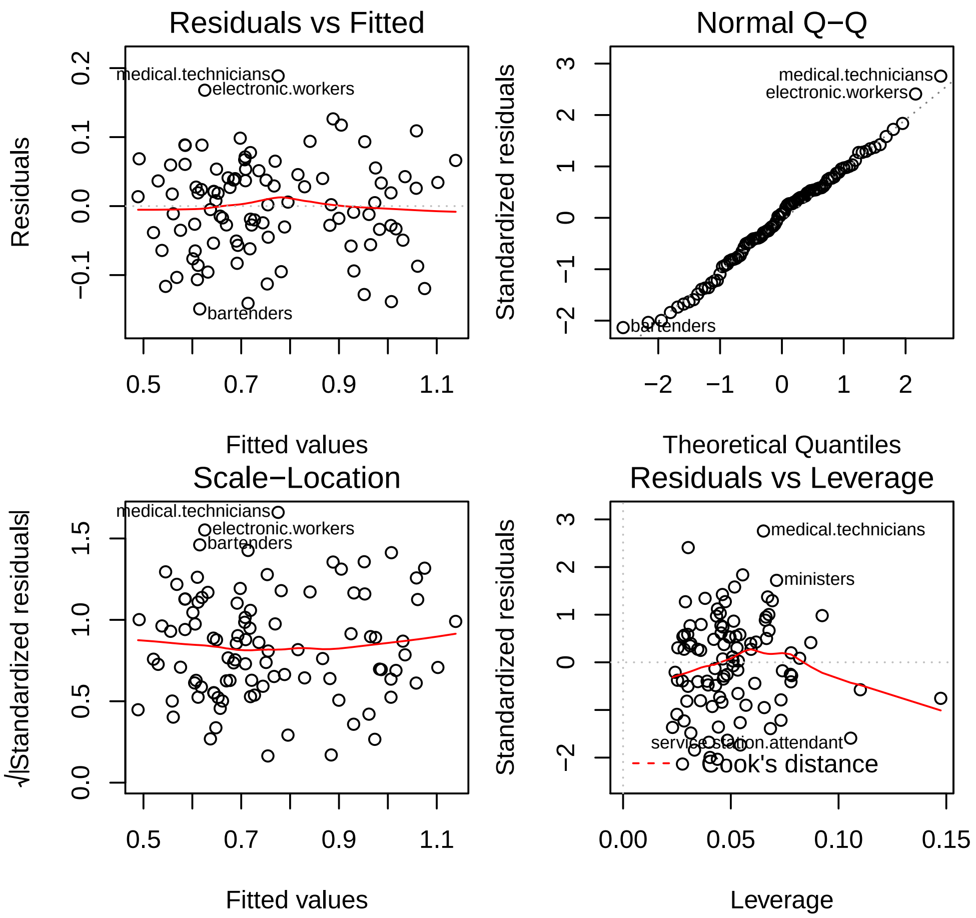


The predictor `women` doesn't seem to add much information, so we ignore it. We first add the factor type.

```
> fit02 <- lm(prestige ~ income + education + type, data=Prestige2)
> summary(fit02)
Call:
lm(formula = prestige ~ income + education + type, data = Prestige2)
```

```
Residuals:
      Min        1Q    Median        3Q       Max
-0.149034 -0.048319  0.005344  0.042236  0.188723

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.98026    0.16134  -6.076 2.68e-08 ***
income       0.11391    0.01831   6.222 1.39e-08 ***
education    0.22387    0.04204   5.325 6.99e-07 ***
typeprof     0.06973    0.03791   1.839   0.0691 .
typewc      -0.01911    0.02601  -0.735   0.4644
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0708 on 93 degrees of freedom
  (4 observations deleted due to missingness)
Multiple R-squared: 0.8506,	Adjusted R-squared: 0.8442
F-statistic: 132.4 on 4 and 93 DF,  p-value: < 2.2e-16
```

```
> par(mfrow=c(2,2))
> plot(fit02)
```



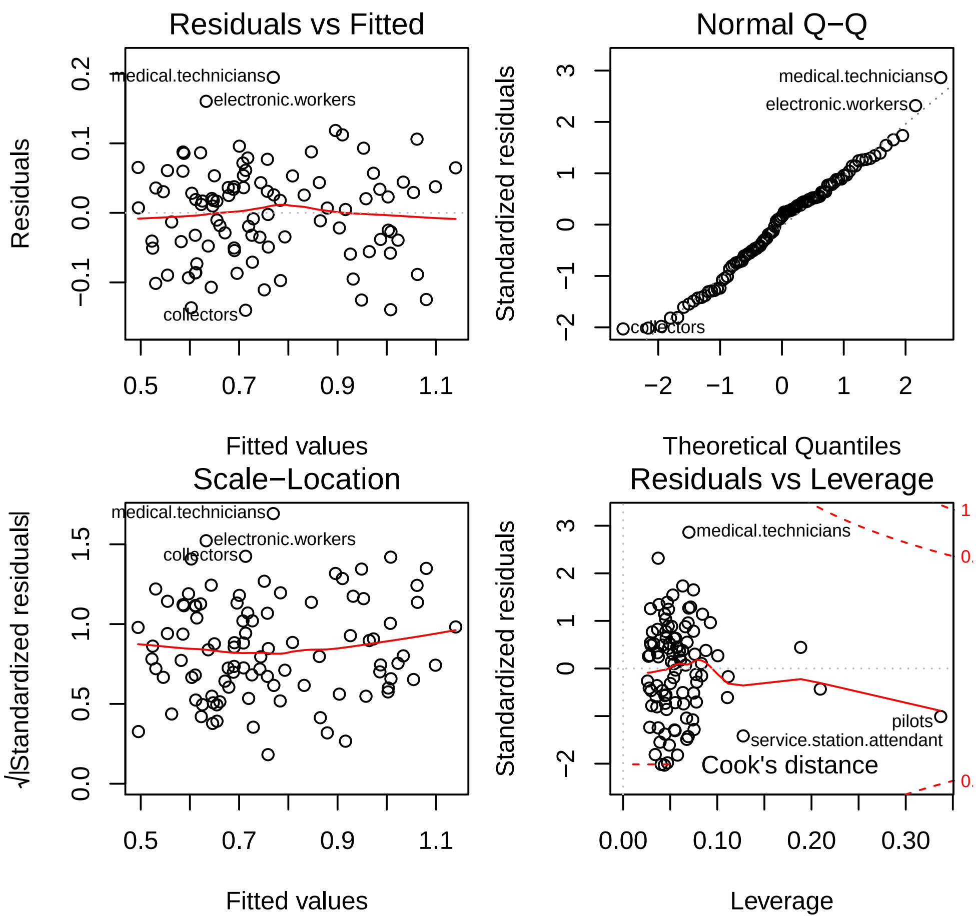
The coefficients for type don't seem to be significant, but since this is a factor, we need to do a partial F-test. Note that from the summary output we see that 4 observations were deleted because of missing values (apparently not all occupations were assigned a type). However, to do the partial F-test we need to fit both models on the same data, so we take out the missing observations and re-fit the old model before doing the F-test:

```
> Prestige2 <- na.omit(Prestige2)
> fit01 <- lm(prestige ~ income + education, data=Prestige2)
> fit02 <- lm(prestige ~ income + education + type, data=Prestige2)
> anova(fit01, fit02)
```

```
Analysis of Variance Table

Model 1: prestige ~ income + education
Model 2: prestige ~ income + education + type
  Res.Df     RSS Df Sum of Sq      F   Pr(>F)
1     95 0.52537
2     93 0.46618  2  0.059193 5.9044 0.003855 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

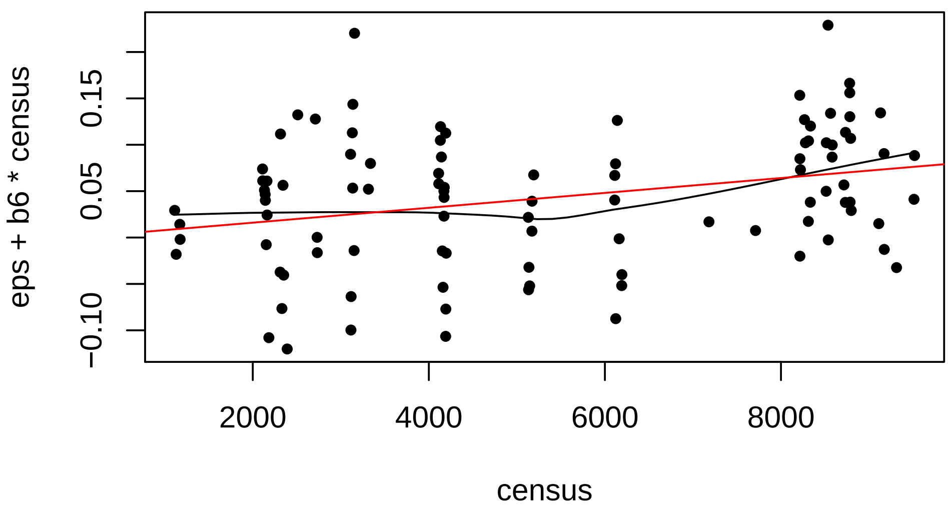
We see that the p-value is quite low, so we can reject the null hypothesis that the two models were equal (e.g. on the 5% level). Hence `type` is significant, and since the diagnostic plots also look fine, we leave it in the model.
Next we add the variable census:

```
> fit03 <- lm(prestige ~ income + education + type + census, data=Prestige2)
> summary(fit03)
Call:
lm(formula = prestige ~ income + education + type + census, data = Prestige2)

Residuals:
     Min       1Q   Median       3Q      Max
-0.14008 -0.04881  0.01096  0.04354  0.19490

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.030e+00  1.655e-01  -6.223 1.43e-08 ***
income       1.070e-01  1.904e-02   5.622 2.01e-07 ***
education    2.392e-01  4.361e-02   5.485 3.62e-07 ***
typeprof     1.040e-01  4.643e-02   2.239   0.0276 *
typewc       2.752e-03  3.113e-02   0.088   0.9297
census       8.016e-06  6.318e-06   1.269   0.2077
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.07057 on 92 degrees of freedom
Multiple R-squared: 0.8532,       Adjusted R-squared: 0.8452
F-statistic: 106.9 on 5 and 92 DF,  p-value: < 2.2e-16
> par(mfrow=c(2,2))
> plot(fit03)
```



We see from the summary output, that census is not significant. However, from the plot against the residuals it looks like there is a nonlinear relation (V-shaped). We look at the partial residual plot to confirm this:

```
> scatter.smooth(Prestige2$census, resid(fit03)+coef(fit03)[6]*Prestige2$census,
                 xlab="census", ylab="eps + b6 * census", pch=20)
> abline(0, coef(fit03)[6], lwd=1, col="red")
```



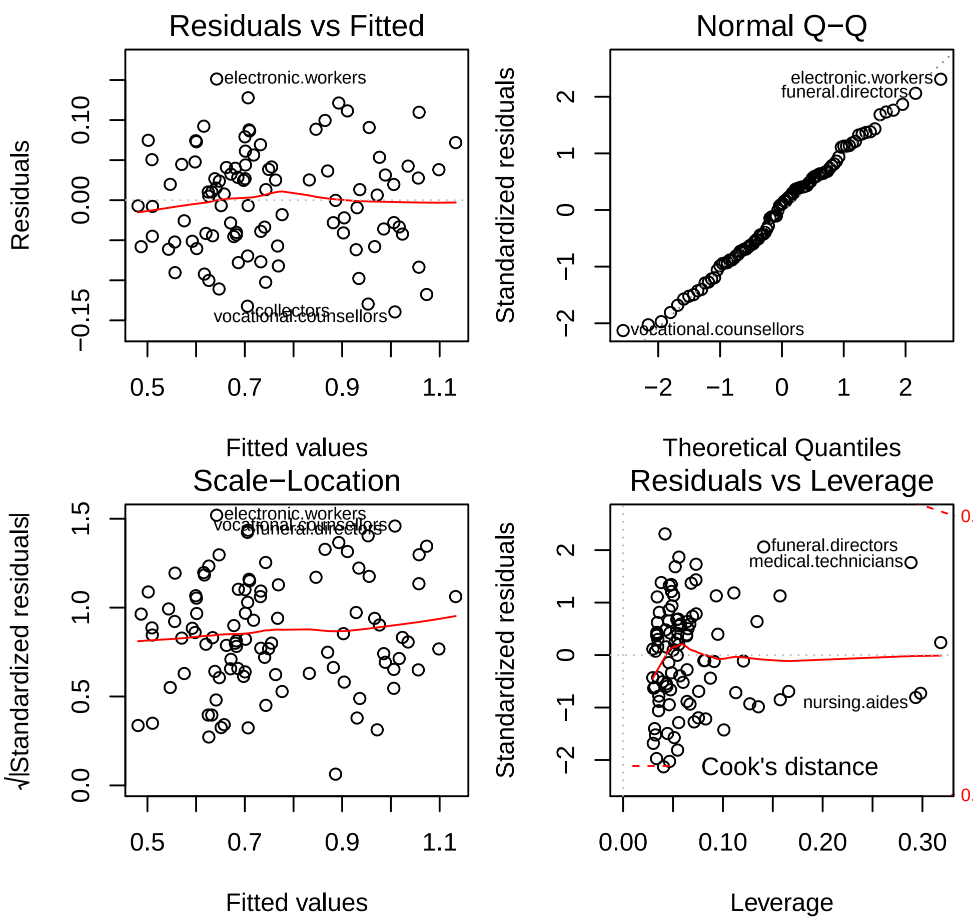
What we can try to keep it in the model is to categorize it:

```
> Prestige2$census.cat <- cut(Prestige2$census, c(0,4000,7000,10000))
> fit04 <- lm(prestige ~ income + education + type + census.cat, data=Prestige2)
> summary(fit04)
```

```
Call:
lm(formula = prestige ~ income + education + type + census.cat,
```

```
    data = Prestige2)

Residuals:
      Min        1Q    Median        3Q       Max
-0.139485 -0.044921  0.007032  0.042246  0.151240

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)             -0.88873    0.16106  -5.518 3.20e-07 ***
income                   0.10673    0.01785   5.981 4.31e-08 ***
education                0.22630    0.04248   5.327 7.18e-07 ***
typeprof                 0.03554    0.04387   0.810    0.420
typewc                   0.03127    0.02870   1.090    0.279
census.cat(4e+03,7e+03] -0.09779    0.03719  -2.630    0.010 *
census.cat(7e+03,1e+04] -0.02258    0.03930  -0.574    0.567
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0669 on 91 degrees of freedom
Multiple R-squared: 0.8695,     Adjusted R-squared: 0.8608
F-statistic:   101 on 6 and 91 DF,  p-value: < 2.2e-16
```

```
> par(mfrow=c(2,2))
> plot(fit04)
```



Again, the diagnostic plots look OK, but we need to do a partial F-test to decide whether census is significant:

```
> anova(fit02, fit04)
```

```
Analysis of Variance Table

Model 1: prestige ~ income + education + type
```

```
Model 2: prestige ~ income + education + type + census.cat
  Res.Df     RSS Df Sum of Sq      F   Pr(>F)
1     93 0.46618
2     91 0.40733  2  0.058847 6.5734 0.002155 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

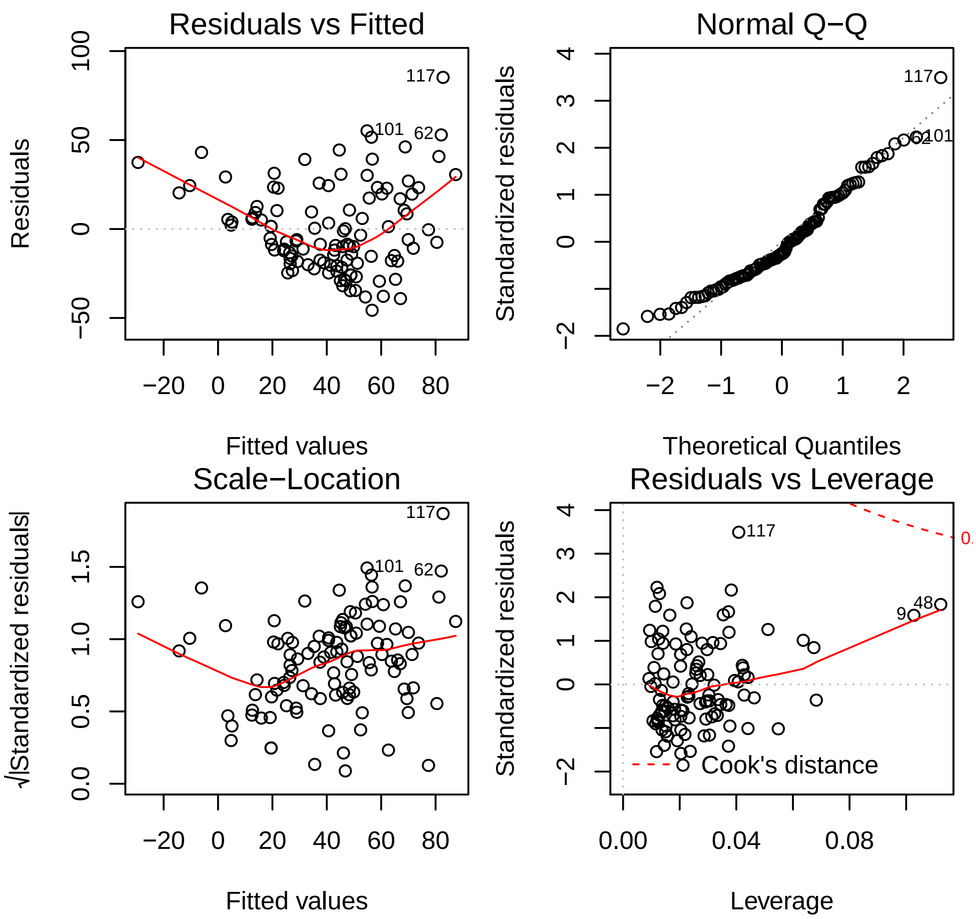
Again, we see that the categorized form of census is significant.

b) Correlated errors

```
> library(faraway)
> data(airquality)
> fit00 <- lm(Ozone ~ Solar.R + Wind, data=airquality)
> summary(fit00)
```

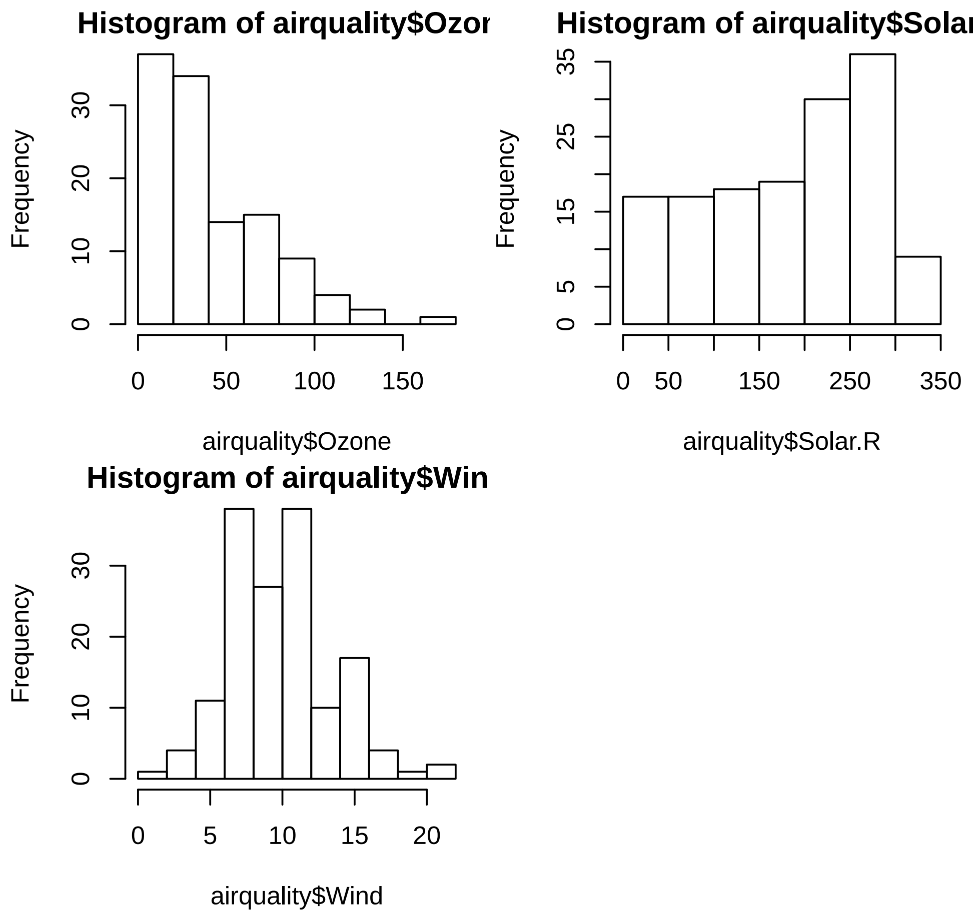
```
Call:
lm(formula = Ozone ~ Solar.R + Wind, data = airquality)

Residuals:
    Min      1Q  Median      3Q     Max
-45.651 -18.164  -5.959  18.514  85.237

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 77.24604    9.06751   8.519 1.05e-13 ***
Solar.R      0.10035    0.02628   3.819 0.000224 ***
Wind        -5.40180    0.67324  -8.024 1.34e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 24.92 on 108 degrees of freedom
  (42 observations deleted due to missingness)
Multiple R-squared: 0.4495,	Adjusted R-squared: 0.4393
F-statistic: 44.09 on 2 and 108 DF,  p-value: 1.003e-14
```

```
> par(mfrow=c(2,2))
> plot(fit00)
```



This model does not fit at all. We can see a massive systematic error in the Tukey-Anscombe plot which makes this initial model unacceptable. Additionally, several observations were removed due to missing values. First, we check for transformations:

```
> par(mfrow=c(2,2))
> hist(airquality$Ozone)
> hist(airquality$Solar.R)
> hist(airquality$Wind)
```

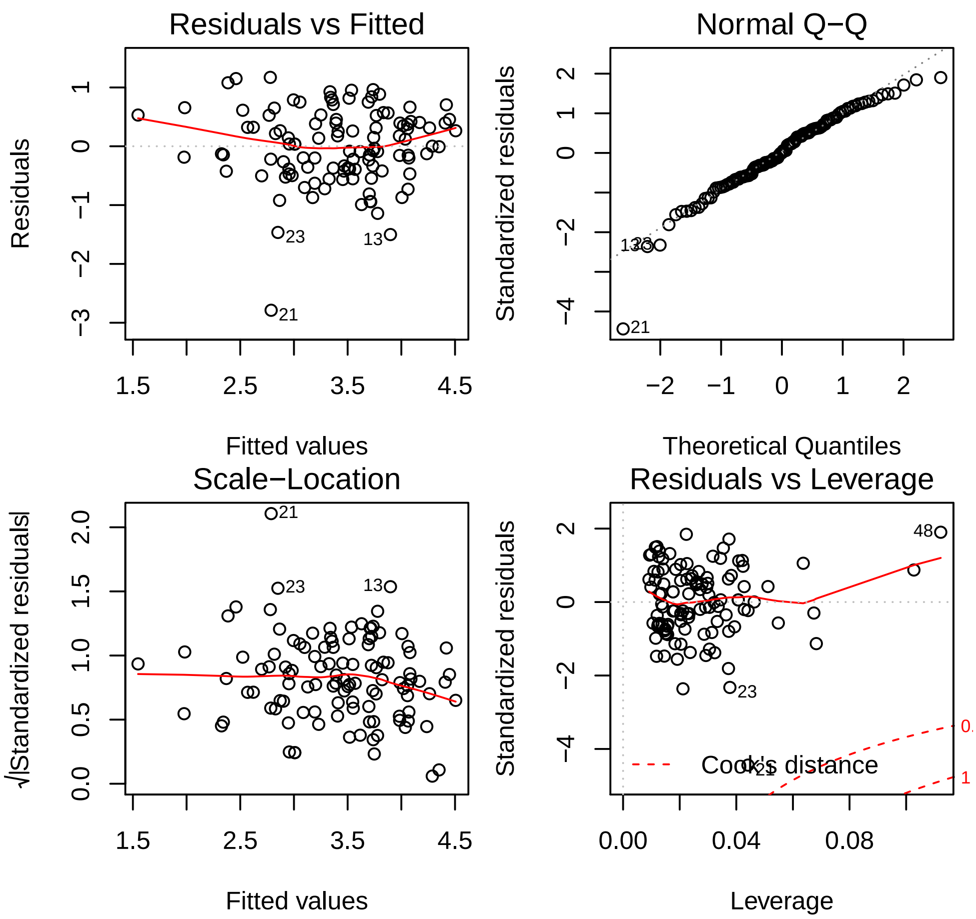


Since Ozone is heavily right-skewed, we do a log-transformation on it.

```
> fit01 <- lm(log(Ozone) ~ Solar.R + Wind, data=airquality)
> summary(fit01)
```

```
Call:
lm(formula = log(Ozone) ~ Solar.R + Wind, data = airquality)

Residuals:
     Min       1Q   Median       3Q      Max
-2.78747 -0.38971  0.00222  0.43882  1.17156

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.9519449  0.2337241  16.909  < 2e-16 ***
Solar.R      0.0037215  0.0006773   5.494 2.63e-07 ***
Wind        -0.1231183  0.0173535  -7.095 1.42e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6423 on 108 degrees of freedom
  (42 observations deleted due to missingness)
Multiple R-squared: 0.4598,       Adjusted R-squared: 0.4498
F-statistic: 45.96 on 2 and 108 DF,  p-value: 3.612e-15
```

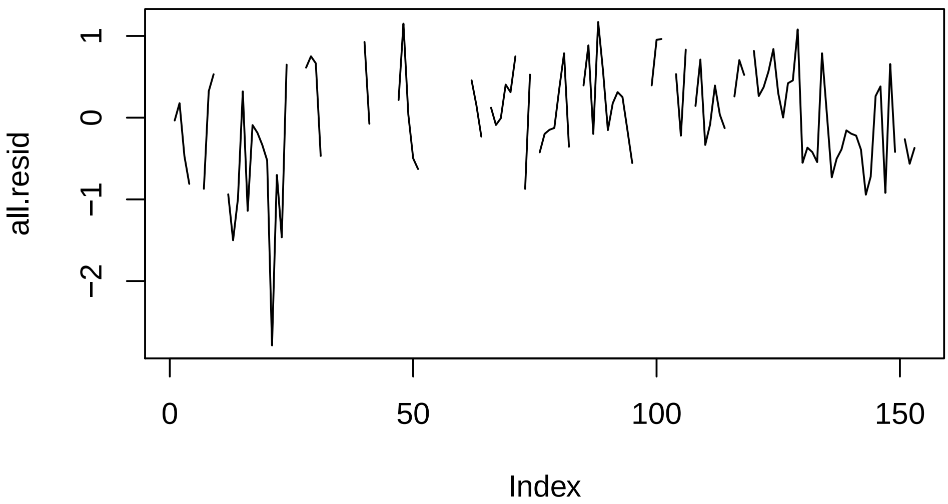
```
> par(mfrow=c(2,2))
> plot(fit01)
```



This improves the situation but the fit is still far from perfect.

We now check for correlated residuals:

```
> # We want to get the full vector of residuals, including missing values, so we
> # first fill up a vector with NA's and then fill in the values we have.
>
> all.resid <- rep(NA, 153)
> all.resid[as.numeric(names(resid(fit01)))] <- resid(fit01)
> plot(all.resid, type="l")
```



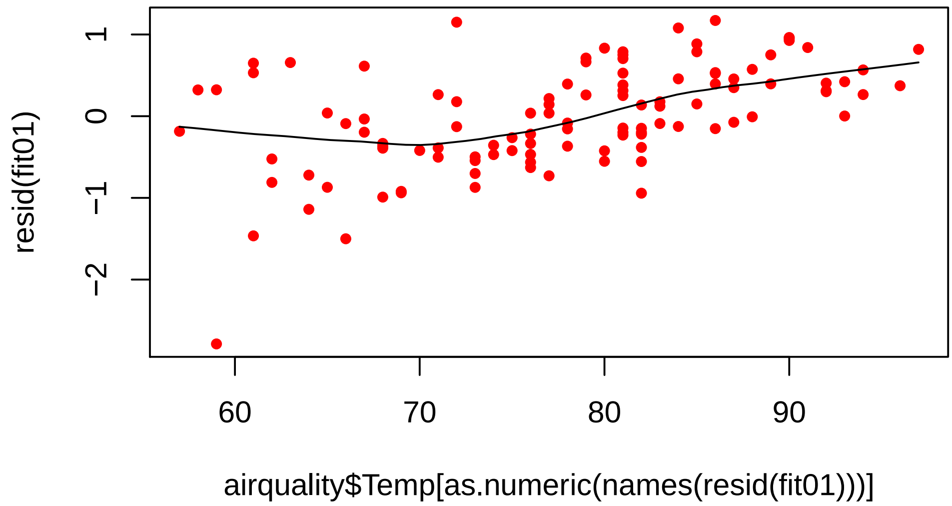
It is difficult to check for correlation just from this plot. We get more insight from a Durbin-Watson test for autocorrelation:

```
> library(lmtest)
> dwtest(fit01, alternative="two.sided")
```

```
        Durbin-Watson test

data:  fit01
DW = 1.4551, p-value = 0.003467
alternative hypothesis: true autocorrelation is not 0
```

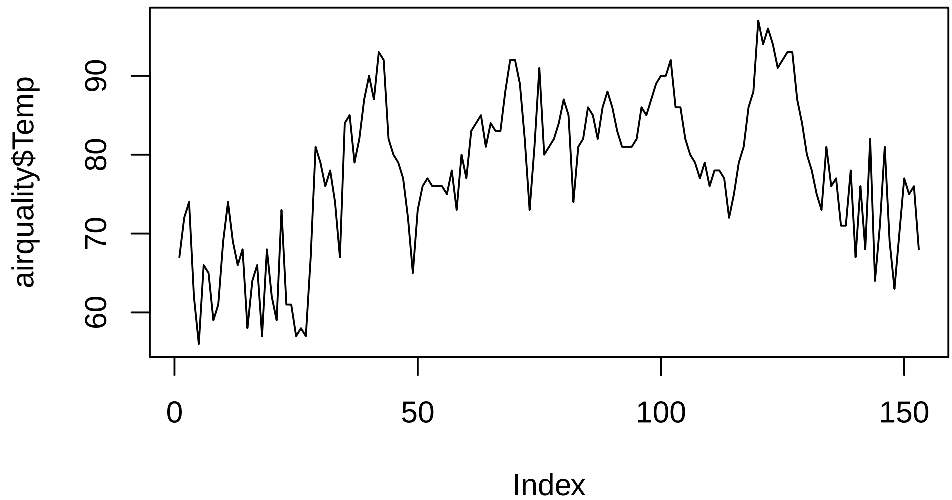
We see that the Durbin-Watson test is significant, rejecting the null-hypothesis of uncorrelated residuals. This autocorrelation probably originates from time-dependent changes in `Ozone`. If the variable `Temp` exhibits the same time-dependence and autocorrelation structure, we could improve the situation by adding it into the model.
We first look at the plot of residuals againt `Temp`:

```
> scatter.smooth(airquality$Temp[as.numeric(names(resid(fit01)))],
                 resid(fit01), pch=20, col="red")
```



There indeed seems to be a relation between `Temp` and the residuals. We can also check its autocorrelation visually:

```
> plot(airquality$Temp, type="l")
```

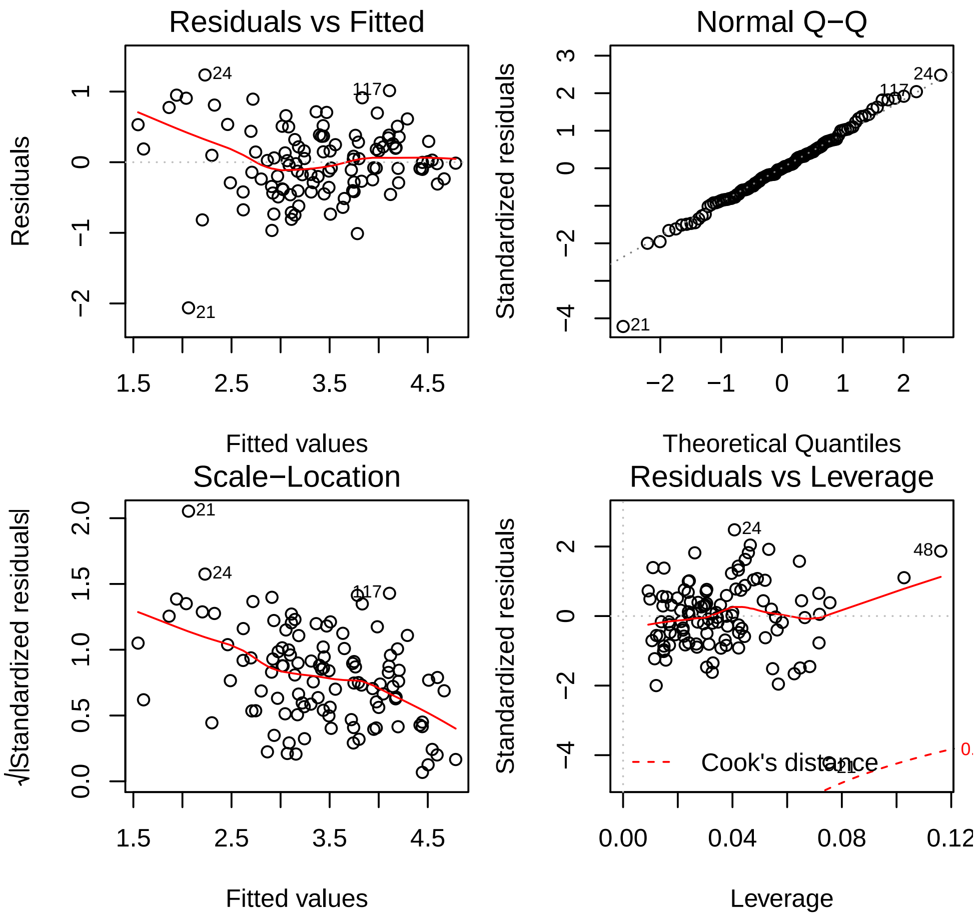


Days with high temperature are generally followed by days with high temperature, and the same holds for cold days.
We now add `Temp` into the model:

```
> fit02 <- lm(log(Ozone) ~ Solar.R + Wind + Temp, data=airquality)
> summary(fit02)
```

```
Call:
lm(formula = log(Ozone) ~ Solar.R + Wind + Temp, data = airquality)
```

```
Residuals:
     Min       1Q   Median       3Q      Max 
-2.06193 -0.29970 -0.00231  0.30756  1.23578 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.2621323  0.5535669  -0.474 0.636798    
Solar.R      0.0025152  0.0005567   4.518 1.62e-05 ***
Wind        -0.0615625  0.0157130  -3.918 0.000158 ***
Temp         0.0491711  0.0060875   8.077 1.07e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 0.5086 on 107 degrees of freedom
  (42 observations deleted due to missingness)
Multiple R-squared: 0.6644,	Adjusted R-squared: 0.655 
F-statistic: 70.62 on 3 and 107 DF,  p-value: < 2.2e-16
> par(mfrow=c(2,2))
> plot(fit02)
> dwtest(fit02, alternative="two.sided")
	Durbin-Watson test

data:  fit02 
DW = 1.8068, p-value = 0.2668
alternative hypothesis: true autocorrelation is not 0 
```

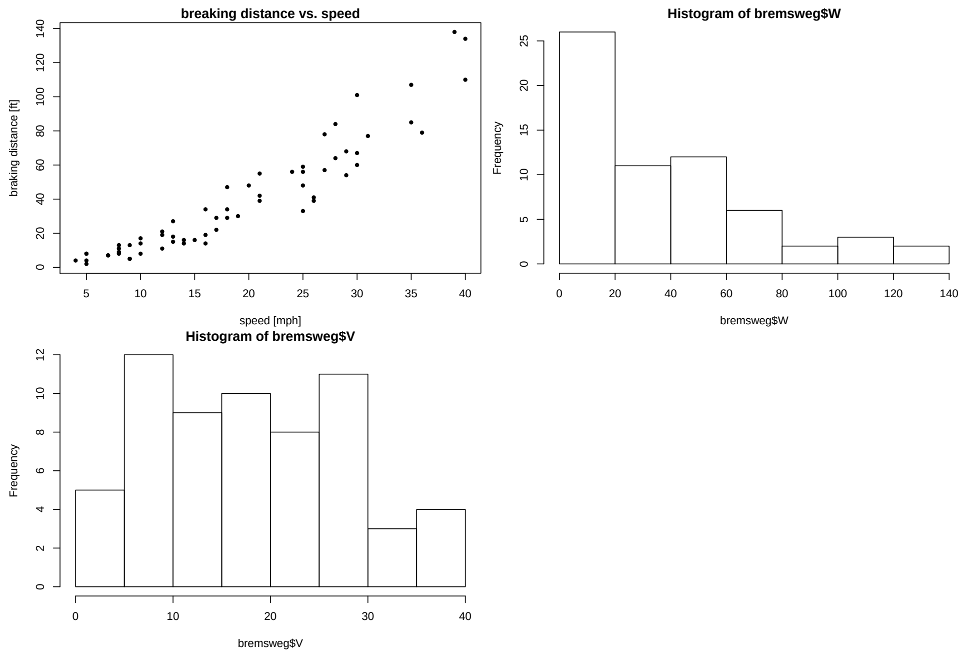


Now the residuals are not correlated anymore. There are still other problems like non-constant variance and non-zero expectation. For the non-constant variance we could do a weighted regression. The cause for the trend in the residuals is probably a nonlinear relation with the predictors. We could try to improve this by either categorizing (like in part a) or by using more advanced techniques like Generalized Additive Model (GAM) regression.

3. Braking distance

a) We first look at the scatter plot and histograms:

```
> load("bremsweg.rda")
> par(mfrow=c(2,2))
> plot(W ~ V, data=bremsweg, xlab= "speed [mph]", ylab="braking distance [ft]",
      main="breaking distance vs. speed", pch=20)
> hist(bremsweg$W)
> hist(bremsweg$V)
```



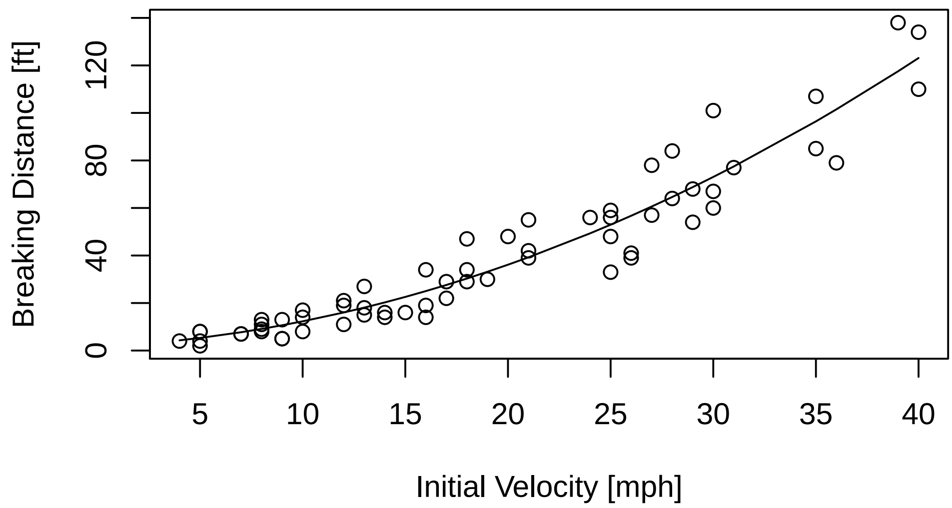
Braking distances are quite strongly right-skewed distributed. However, we know from simple physics that the breaking distance is proportional to the square of the initial velocity (for constant acceleration). Therefore we do not log-transform in this case.

b) As noted in the previous part, we expect a quadratic relation, so we fit a second order polynomial:

```
> fit <- lm(W ~ V + I(V^2), data=bremsweg)
> summary(fit)
```

```
Call:
lm(formula = W ~ V + I(V^2), data = bremsweg)

Residuals:
     Min       1Q   Median       3Q      Max
-22.5192  -5.4527  -0.5519   3.8442  27.9373

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.58036    5.10266   0.310    0.758
V            0.41607    0.55641   0.748    0.458
I(V^2)       0.06556    0.01303   5.033 4.83e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.927 on 59 degrees of freedom
Multiple R-squared: 0.9144,       Adjusted R-squared: 0.9115
F-statistic: 315.3 on 2 and 59 DF,  p-value: < 2.2e-16
```

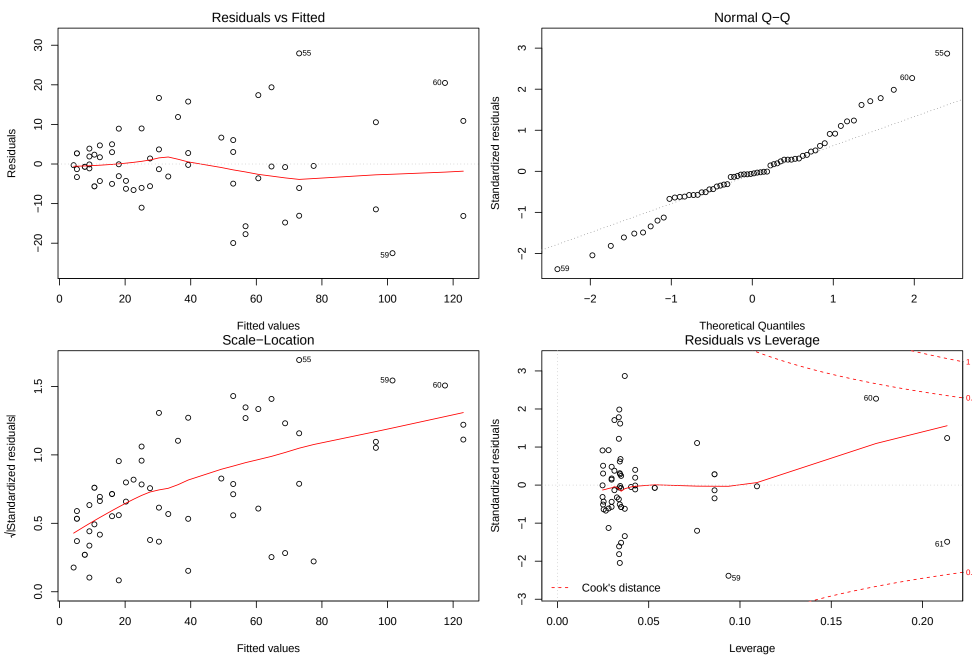
```
> plot(W ~ V, data=bremsweg, xlab="Initial Velocity [mph]",
       ylab="Breaking Distance [ft]")
> lines(bremsweg$V, predict(fit, data=bremsweg))
```



We see that the quadratic term is highly significant. Even though the linear term seems not significant, we don't exclude it. In polynomial regression, we always include all lower-order terms, because otherwise the model formulation would not be stable under linear transformations of the predictors.

c) If we brake with constant acceleration $\alpha < 0$ from initial velocity v_0, our velocity will be $v(t) = v_0 + \alpha t$ and it takes $t = -v_0/\alpha$ units of time to stop. Integrating this up we get the breaking distance $d = \int_0^{-v_0/\alpha} v(t)\mathrm{d}t = -v_0^2/\alpha$. If braking is only initiated after a reaction time t_r, we get $d = t_r v_0 - v_0^2/\alpha$. Looking at the regression output from the previous part, we can read off the values for t_r and $-1/\alpha$ – they are the coefficients of `V` and `I(V^2)`. Thus, transforming to standard units, we get $t_r = 0.28s$ and $\alpha = -9.97m/s^2$, which seem fairly reasonable.

d)
```
> par(mfrow=c(2,2))
> plot(fit)
```

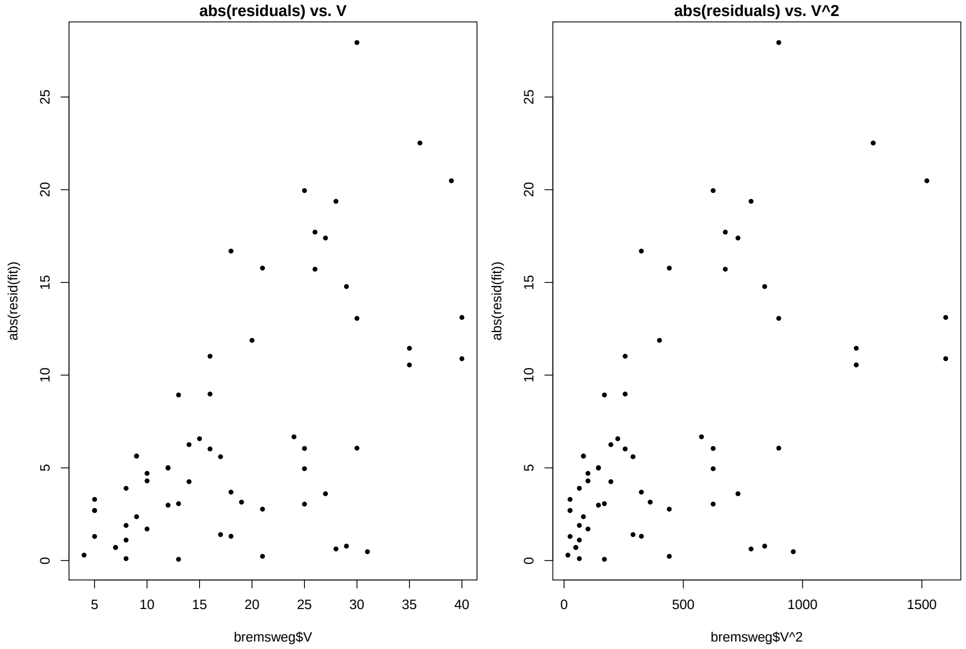


There are two main problems: non-constant variance of the residuals and a long-tailed error distribution.

e) Weighted regression

First, we need to choose suitable weights. To this end, we plot the absolute values of the residuals against the predictors.

```
> par(mfrow=c(1,2))
> plot(bremsweg$V, abs(resid(fit)), pch=20, main="abs(residuals) vs. V")
> plot(bremsweg$V^2, abs(resid(fit)), pch=20, main="abs(residuals) vs. V^2")
```

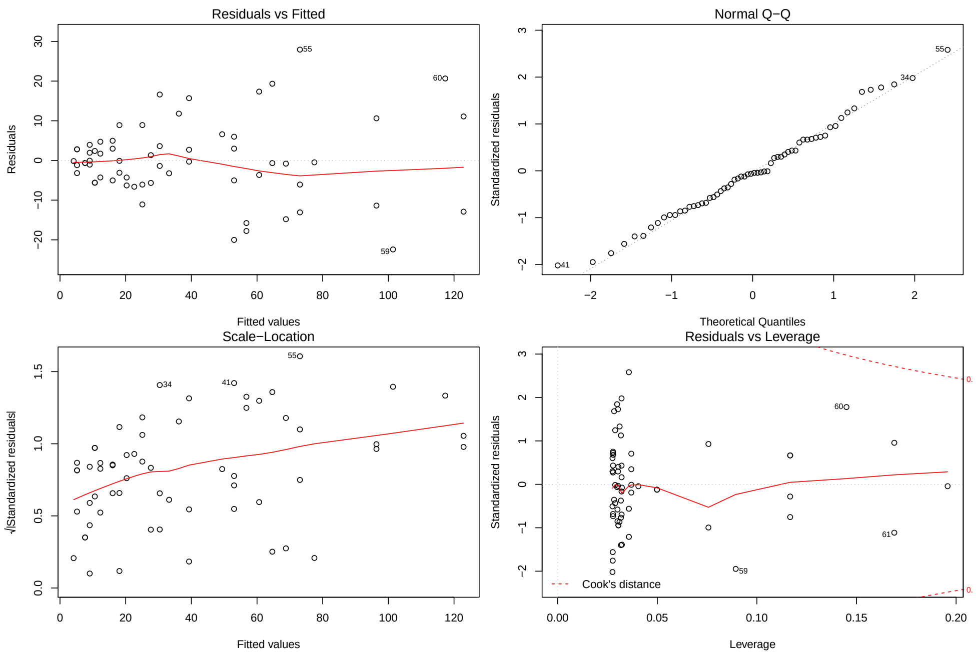


We can see that the variance is roughly proportional to V and not V^2. Therefore, we choose the weights as 1/V.

```
> fit.weight <- lm(W ~ V + I(V^2), weights=1/V, data=bremsweg)
> summary(fit.weight)
```

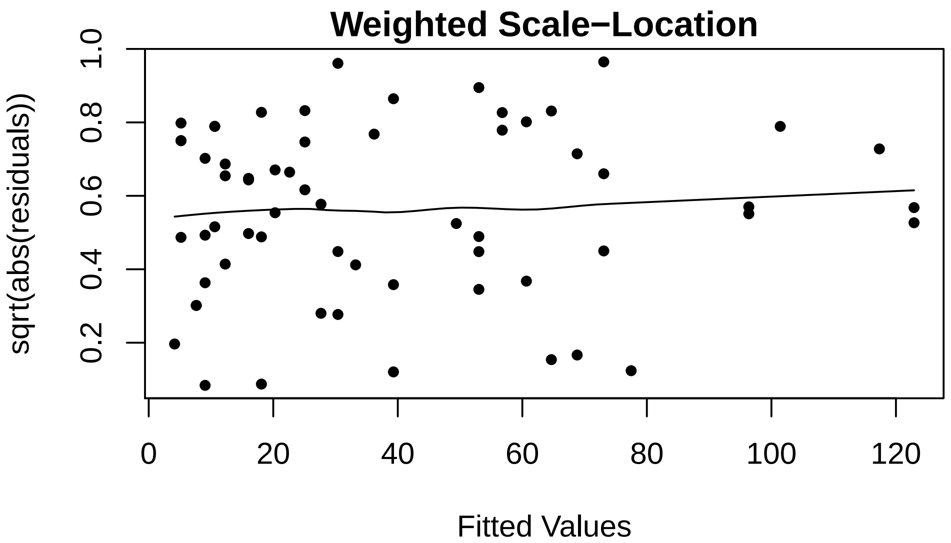
```
Call:
lm(formula = W ~ V + I(V^2), data = bremsweg, weights = 1/V)

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-4.0037 -1.4120 -0.1054  1.2586  5.0984

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.32590    3.09898   0.428    0.670
V            0.44801    0.42065   1.065    0.291
I(V^2)       0.06479    0.01122   5.777 3.03e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.011 on 59 degrees of freedom
Multiple R-squared: 0.923,       Adjusted R-squared: 0.9204
F-statistic: 353.8 on 2 and 59 DF,  p-value: < 2.2e-16
```

```
> par(mfrow=c(2,2))
> plot(fit.weight)
```



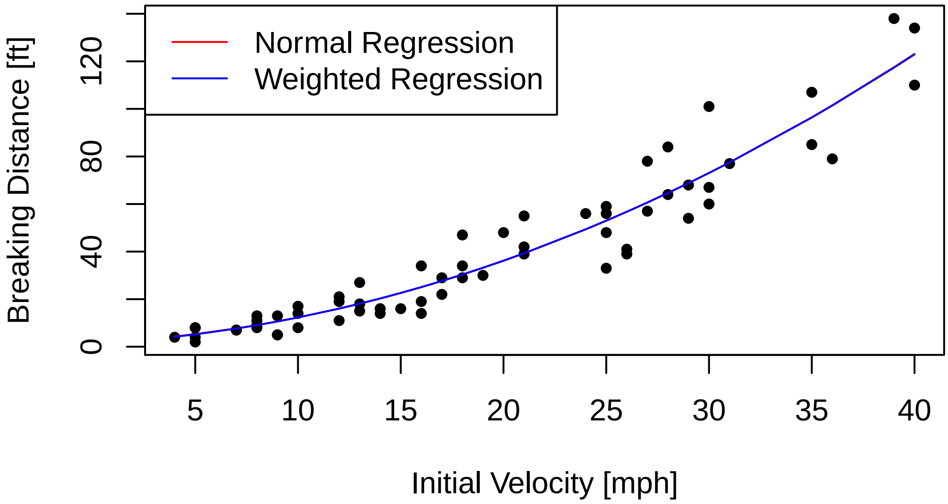
The regression summary looks similar as before. Note that the residuals used in the diagnostic plots are the unweighted residuals. To check whether the weighting has stabilized the variance sufficiently, we look at the weighted residuals. For this we generate a scale-location plot 'by hand':

```
> scatter.smooth(fitted(fit.weight), sqrt(abs(resid(fit.weight)/bremsweg$V)),
                 xlab="Fitted Values", ylab="sqrt(abs(residuals))",
                 main="Weighted Scale-Location", pch=20)
```



Weighting has improved the situation considerably. We note that also the quantiles of the error distribution look a lot more normal now. Our final result looks like this:

```
> plot(W ~ V, data=bremsweg, xlab="Initial Velocity [mph]",
       ylab="Breaking Distance [ft]", pch=20)
> lines(bremsweg$V, predict(fit, data=bremsweg), col="red")
> lines(bremsweg$V, predict(fit.weight, data=bremsweg), col="blue")
> legend("topleft", legend=c("Normal Regression", "Weighted Regression"),
         col=c("red", "blue"), lwd=1)
```



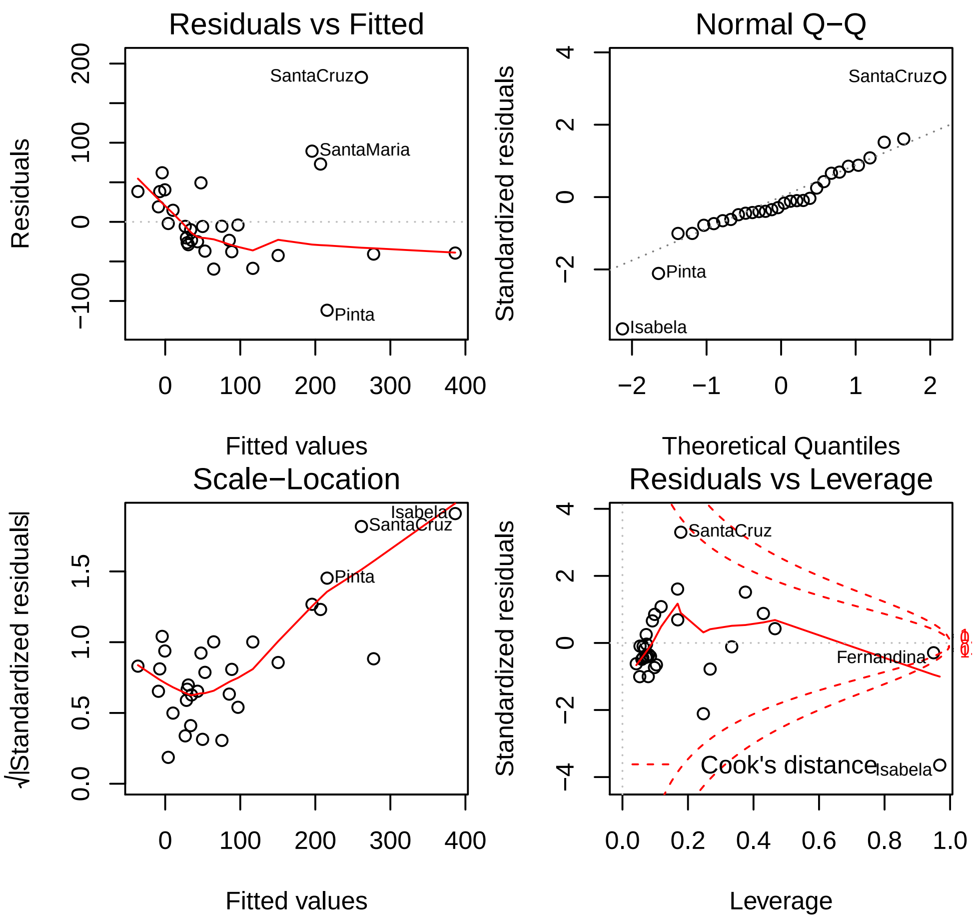
There is no visible difference between the two fits in this case.

f) Robust regression

```
> data(gala, package="faraway")
> fit0 <- lm(Species ~ Area + Elevation + Scruz + Nearest + Adjacent, data=gala)
> summary(fit0)
```

```
Call:
lm(formula = Species ~ Area + Elevation + Scruz + Nearest + Adjacent,
    data = gala)

Residuals:
     Min       1Q   Median       3Q      Max
-111.679  -34.898   -7.862   33.460  182.584

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.068221  19.154198   0.369 0.715351
Area        -0.023938   0.022422  -1.068 0.296318
Elevation    0.319465   0.053663   5.953 3.82e-06 ***
Scruz       -0.240524   0.215402  -1.117 0.275208
Nearest      0.009144   1.054136   0.009 0.993151
Adjacent    -0.074805   0.017700  -4.226 0.000297 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 60.98 on 24 degrees of freedom
Multiple R-squared: 0.7658,        Adjusted R-squared: 0.7171
F-statistic:  15.7 on 5 and 24 DF,  p-value: 6.838e-07
```

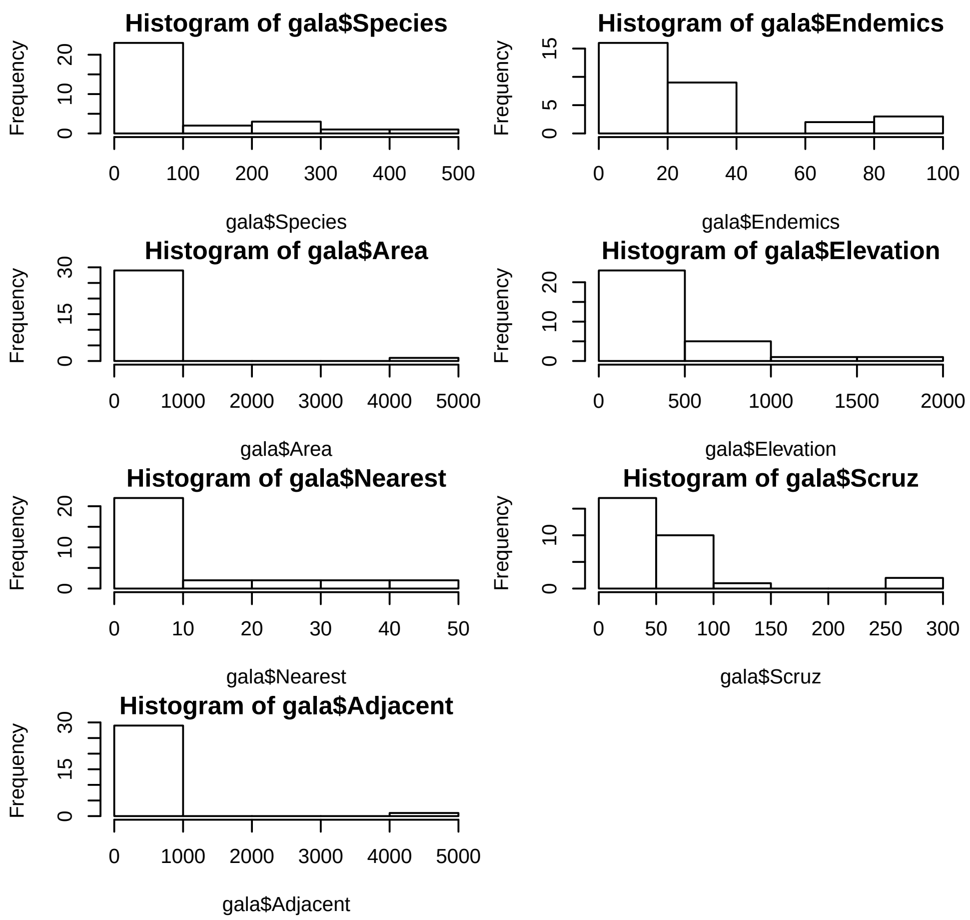
```
> par(mfrow=c(2,2))
> plot(fit0)
```



We can see from the summary output and the residual plots that the present model is not suitable for describing the given data. Only `Elevation` and `Adjacent` seem to have a significant influence on `Species`. The diagnostic plots show a strong violation of constant variance and normality. Additionally, Cook's distances shwo two observations being quite influential and one observation (`Isabela`) being a leverage point.

These structural deficiencies suggest the necessity of transformations. We have a look at the histograms of the variables to decide which transformations to apply.

```
> par(mfrow=c(4,2))
> hist(gala$Species)
> hist(gala$Endemics)
> hist(gala$Area)
> hist(gala$Elevation)
> hist(gala$Nearest)
> hist(gala$Scruz)
> hist(gala$Adjacent)
```

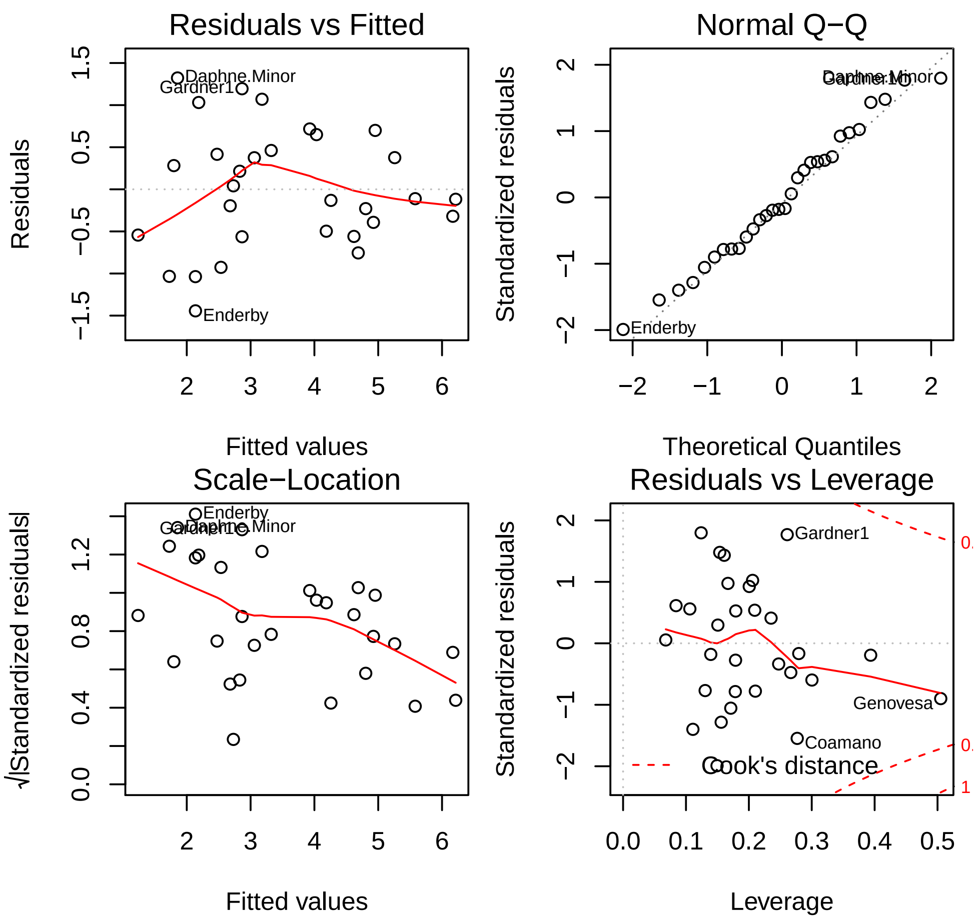


Since they are all heavily right-skewed, we apply log transformations. Note that `Endemics` and `Scruz` contain zero values, so we add the smallest positive value to all the entries.

```
> which(gala$Species <= 0)
integer(0)
> which(gala$Endemics <= 0)
[1] 7
> which(gala$Area <= 0)
integer(0)
> which(gala$Elevation <= 0)
integer(0)
> which(gala$Nearest <= 0)
integer(0)
> which(gala$Scruz <= 0)
[1] 25
> which(gala$Adjacent <= 0)
integer(0)
> gala.log <- gala
> gala.log$Species <- log(gala$Species)
> gala.log$Endemics <- log(gala$Endemics + min(gala$Endemics[-7]))
> gala.log$Area <- log(gala$Area)
> gala.log$Elevation <- log(gala$Elevation)
> gala.log$Nearest <- log(gala$Nearest)
> gala.log$Scruz <- log(gala$Scruz + min(gala$Scruz[-25]))
> gala.log$Adjacent <- log(gala$Adjacent)
```

We now fit a model with the transformed variables:

```
> fit1 <- lm(Species ~ Area + Elevation + Scruz + Nearest + Adjacent, data=gala.log)
> summary(fit1)
Call:
lm(formula = Species ~ Area + Elevation + Scruz + Nearest + Adjacent,
    data = gala.log)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4429 -0.5317 -0.1144  0.4500  1.3229

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.20280    1.66715   3.121  0.00465 **
Area         0.50769    0.09982   5.086 3.34e-05 ***
Elevation   -0.38217    0.32261  -1.185  0.24777
Scruz       -0.10039    0.10781  -0.931  0.36105
Nearest     -0.06017    0.11533  -0.522  0.60663
Adjacent    -0.02543    0.04578  -0.555  0.58370
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7859 on 24 degrees of freedom
Multiple R-squared: 0.7909,     Adjusted R-squared: 0.7473
F-statistic: 18.15 on 5 and 24 DF,  p-value: 1.839e-07
> par(mfrow=c(2,2))
> plot(fit1)
```

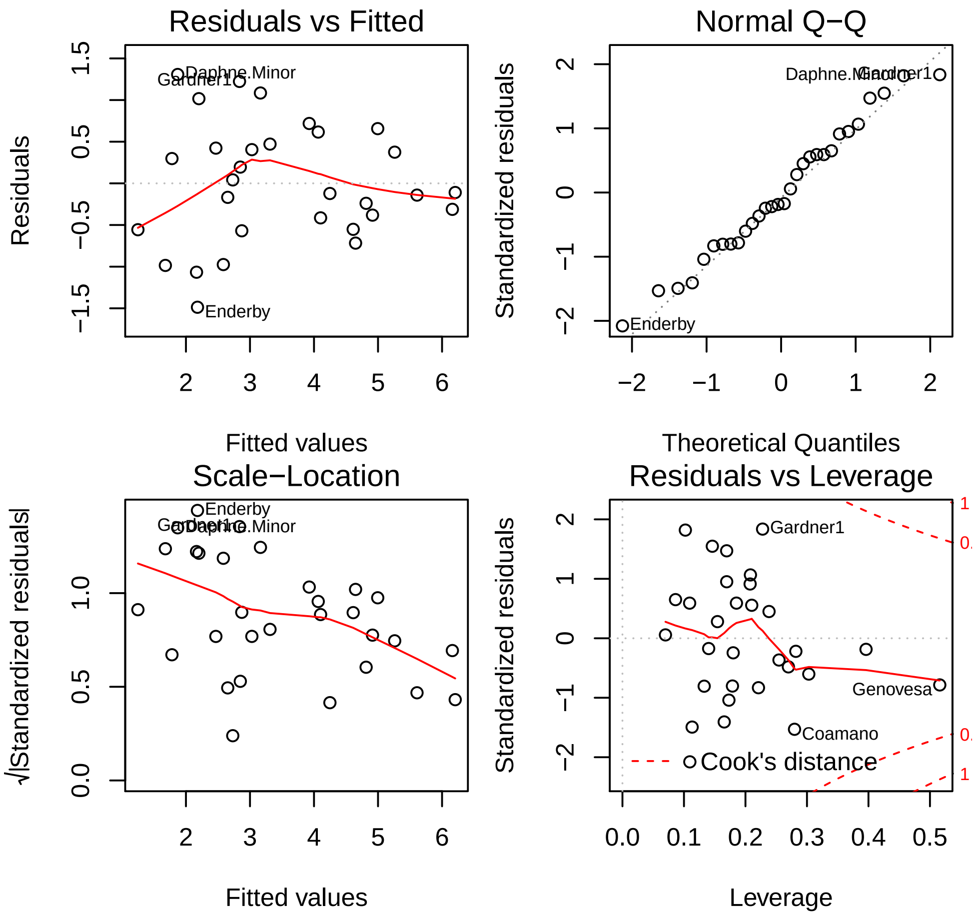


The residual plots still indicate non-constant variance and non-zero expectation. However, at least the normality assumption seems satisfied now.
We now estimate the parameters in a robust fashion and see whether we can improve our model.

```
> library(MASS)
> fit2 <- rlm(Species ~ Area + Elevation + Scruz + Nearest + Adjacent, data=gala.log)
> summary(fit2)
```

```
Call: rlm(formula = Species ~ Area + Elevation + Scruz + Nearest +
    Adjacent, data = gala.log)
Residuals:
    Min      1Q  Median      3Q     Max
-1.4868 -0.5172 -0.1155  0.4594  1.3074

Coefficients:
            Value   Std. Error t value
(Intercept)  5.0084  1.8992     2.6372
Area         0.4980  0.1137     4.3800
Elevation   -0.3449  0.3675    -0.9386
Scruz       -0.0907  0.1228    -0.7383
Nearest     -0.0687  0.1314    -0.5228
Adjacent    -0.0324  0.0522    -0.6208

Residual standard error: 0.7584 on 24 degrees of freedom
```

```
> par(mfrow=c(2,2))
> plot(fit2)
```



Using a robust estimation of the parameters did not improve the model fit. Probably because the residuals are sufficiently normal distributed after the transformation.