Workshop 2: solutions

1. The situation described in the question translates into a model of the form:

$$\mu_i = \begin{cases} kx_i + \alpha x_i^2, & \text{alloy 1} \\ kx_i + \beta x_i^2, & \text{alloy 2} \\ kx_i + \gamma x_i^2, & \text{alloy 3} \end{cases}$$

where $y_i \sim N(\mu_i, \sigma^2)$. Note that we are told that when there is no load, the is no deformation, so the mean function does not include an intercept. Further, we are also told, that the deformation is expected to vary linearly with load, in exactly the same way for all three alloys and hence the term kx_i in the mean function for all three alloys. Finally, because as load increases the three alloys deviate from the linear behaviour in different ways, with the relationship becoming slightly curved, we have three (possibly) different quadratic terms in the mean function. Let the first 6 deformation measurements correspond to the first alloy type, under the six different specified loads, the subsequent six to the second alloy type, and the final six to the third alloy type.

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \\ y_{16} \\ y_{17} \\ y_{18} \end{pmatrix} = \begin{pmatrix} x_1 & x_1^2 & 0 & 0 \\ x_2 & x_2^2 & 0 & 0 \\ x_3 & x_3^2 & 0 & 0 \\ x_4 & x_4^2 & 0 & 0 \\ x_5 & x_5^2 & 0 & 0 \\ x_6 & x_6^2 & 0 & 0 \\ x_1 & 0 & x_1^2 & 0 \\ x_2 & 0 & x_2^2 & 0 \\ x_3 & 0 & x_3^2 & 0 \\ x_4 & 0 & x_4^2 & 0 \\ x_5 & 0 & x_5^2 & 0 \\ x_6 & 0 & x_6^2 & 0 \\ x_1 & 0 & 0 & x_1^2 \\ x_2 & 0 & 0 & x_2^2 \\ x_3 & 0 & 0 & x_3^2 \\ x_4 & 0 & 0 & x_4^2 \\ x_5 & 0 & 0 & x_5^2 \\ x_6 & 0 & 0 & x_6^2 \end{pmatrix} \begin{pmatrix} k \\ \alpha \\ \beta \\ \gamma \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \\ \epsilon_7 \\ \epsilon_8 \\ \epsilon_9 \\ \epsilon_{10} \\ \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{14} \\ \epsilon_{15} \\ \epsilon_{16} \\ \epsilon_{17} \\ \epsilon_{18} \end{pmatrix}.$$

2. The null model is not *nested* within the alternative model: the predictor variable x_i appears in the null model but not the alternative. In the notation of section 4.3.2, the matrix $\mathbf{C}$ plays a key role in the construction of the test statistic. However, there is no matrix $\mathbf{C}$, which would be made up of a selection of rows of the 4×4 identity matrix, that would lead to the model in H_0.

3. We can perform the required hypothesis test, whose null hypothesis is that all γ_j are zero, using an F test. Using the result in the very final part of section 4.3.2 of the notes, we have that the F test statistic can be computed as

$$F = \frac{(\text{RSS}_0 - \text{RSS}_1)/q}{\text{RSS}_1/(n-p)},$$

where RSS_1 denotes the residual sum of squares for the full model and RSS_0 denotes the residual sum of squares for the reduced model in which the H_0 is true, and q is the difference in the number of parameters in the two models.. In the denominator p denotes the number of parameters in the full model and therefore the denominator of the test statistic is nothing more than the estimated variance of the error term in the full model (remember that $\hat{\sigma}^2 = \|\mathbf{r}\|^2/(n-p)$). So, the denominator of the F test statistic is already available to us, it is 0.3031^2. We can work out the residuals sum of squares for the full and reduced model needed for the numerator using the estimated residual standard error and corresponding degrees of freedom (remember that, in this context, these are given by the sample size minus the number of parameters in the corresponding model). So, we have that $\texttt{RSS}_0 = 0.3009^2 \times 98$ and $\texttt{RSS}_1 = 0.3031^2 \times 95$. The only thing that is left to determine is q. We do not know how many levels the variable treatment has from the text (or equivalently, how many γ_j's there are) but we can work out this from the difference in degrees of freedom. The sample size n is the same is the same for the two models and so the difference in the number of parameters, q, should be three (and therefore there are four treatment levels, with one of them being the baseline or reference level). Let's then compute the value of the test statistic:

$$F = \frac{(((0.3009^2) \times 98) - ((0.3031^2) \times 95))/3}{0.3031^2} = 0.52751$$

We know that under the null hypothesis this test statistic follows a F distribution with degrees of freedom given by q and $n-p$. So, in this case the test statistics follows a $F_{3,95}$ distribution. Let's then compute the p-value.

```
pval <- pf(0.52751, 3, 95, lower.tail = F)
pval
```

```
## [1] 0.6644552
```

The p value is 0.66 and so there is no reason to doubt the null model.

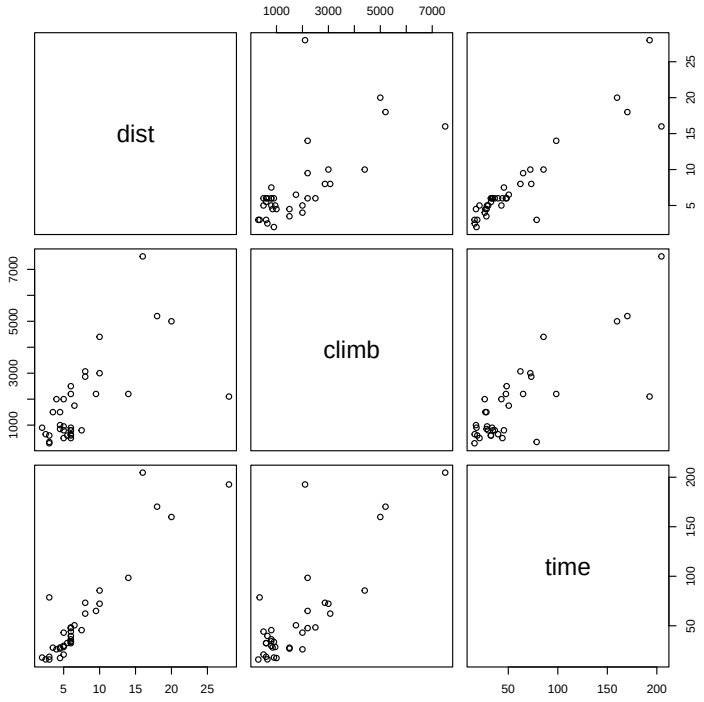
4. The ϵ_i will almost certainly be correlated in time, violating the independence assumption of the linear model, meaning that it will not be possible to accurately assess the uncertainty of model predictions (at least not using linear model theory). A plot of residuals against t_i should show this up or, alternatively, a plot successive pairs of residuals, say, $\hat{\epsilon}_{i+1}$ against $\hat{\epsilon}_i$, should also show the hypothesised correlation between errors (or, a bit beyond the scope of this course, one could use the R function `acf` on the residuals).

5. We first need to load the library `MASS`. We can then start by quickly inspecting the data. Let's do that.

```
library(MASS)
head(hills)
```

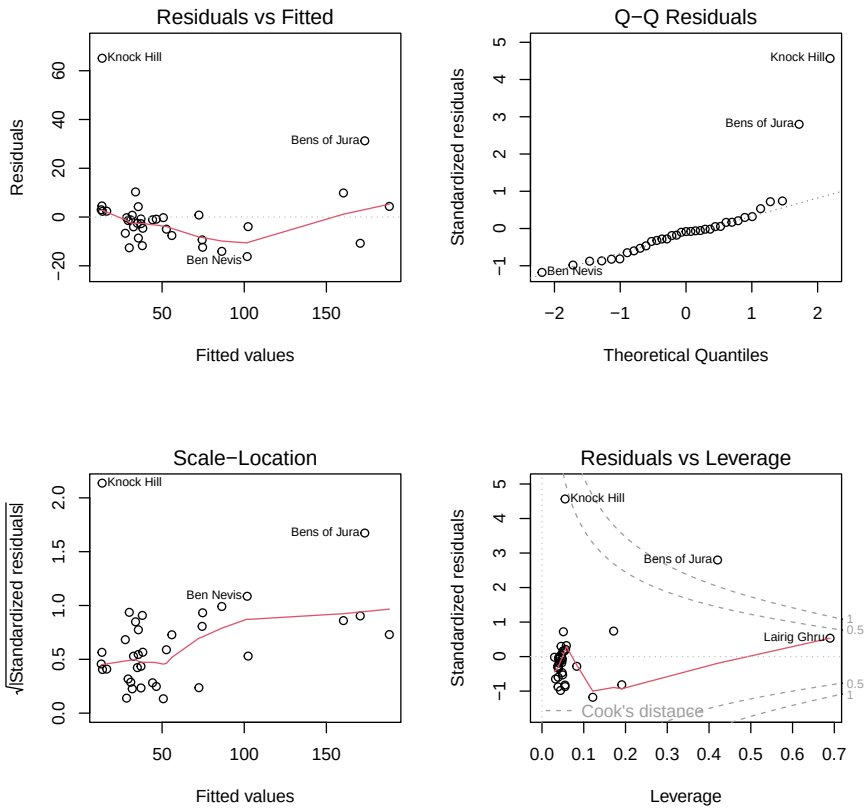
```
##              dist climb   time
## Greenmantle   2.5   650 16.083
## Carnethy      6.0  2500 48.350
## Craig Dunain  6.0   900 33.650
```

```
## Ben Rha        7.5   800 45.600
## Ben Lomond     8.0  3070 62.267
## Goatfell       8.0  2866 73.217

plot(hills)
```



It seems natural that longer races would tend to have greater record times per mile, so we might expect the record time to be a convex increasing function of distance. However, the scatterplot relating these variables reveals a quite strong linear trend. The scatterplot of record time by climb also shows linearity. Let us now start by modeling the winning times as a function of the climb (total height gained during the route) and distance and inspect the residual plots.

```
hm <- lm(time ~ dist + climb, data = hills)
par(mfrow = c(2, 2))
plot(hm)
```



A few things to remark about the residual plots. Knock Hill has a very large standardised residual, Bens of Jura also has a high standardised residual but it can be regarded still borderline under the standard normal distribution. We can see that Bens of Jura has a Cook distance greater than one and so it is safer to re-do all the analyses to be conducted with and without this observation and see how much, if anything, results change. Let us inspect these two observations.

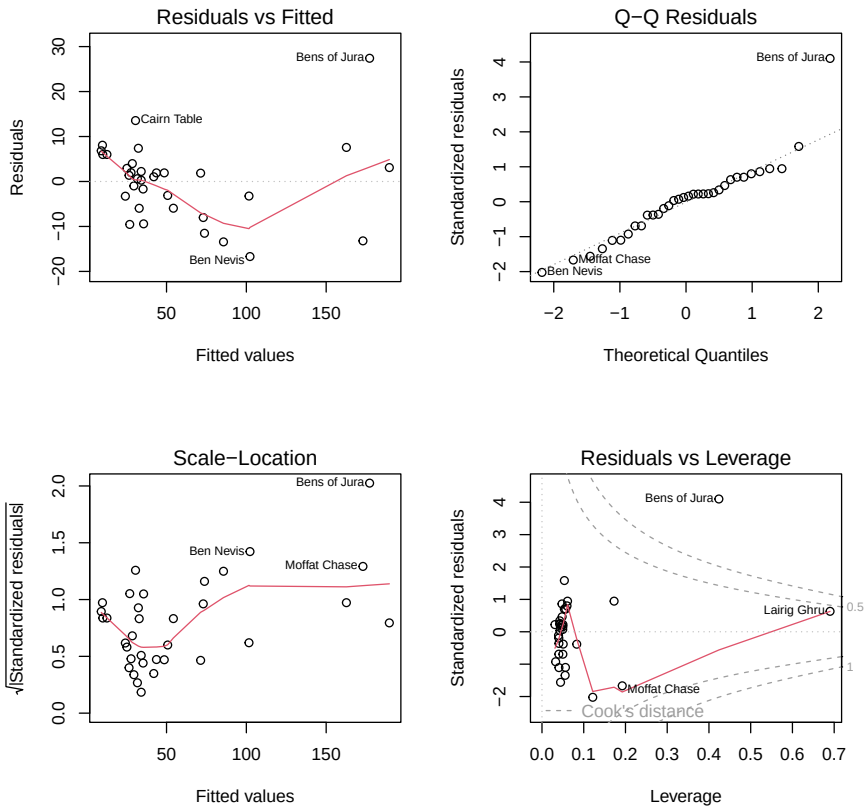
```
hills[which(hm$residuals > 20), ]

##              dist climb    time
## Bens of Jura   16  7500 204.617
## Knock Hill      3   350  78.650
```

Something cannot be right with Knock Hill, this corresponds to a slow walking speed. For this reason, we will remove this observation from the dataset as it appears to be a recording error .

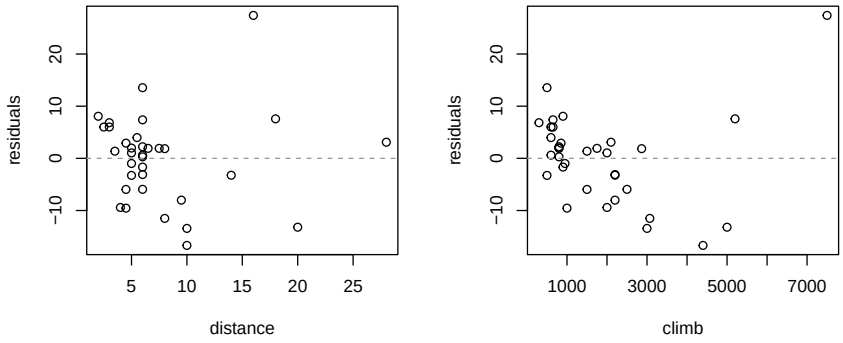
```
hills1 <- hills[- which(hm$residuals > 60), ]
```

There seem to be a trend in the residuals versus fitted values plot (although the sample size is limited (35) and for fitted values of 100 or above there are only 5 observations). To better inspect let us plot the standardised residuals against each of the predictor variables. We will use the version of the dataset without the big outlier (Knock Hill). We also check for this version of the data set the usual residual plots.

```r
hm <- lm(time ~ dist + climb, data = hills1)

summary(hm)

##
## Call:
## lm(formula = time ~ dist + climb, data = hills1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.694  -5.276   1.210   3.758  27.403
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -13.530352   2.649100  -5.108 1.58e-05 ***
## dist          6.364562   0.361130  17.624  < 2e-16 ***
## climb         0.011855   0.001235   9.600 8.41e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.804 on 31 degrees of freedom
## Multiple R-squared:  0.9716,Adjusted R-squared:  0.9698
## F-statistic: 530.9 on 2 and 31 DF,  p-value: < 2.2e-16

par(mfrow = c(2, 2))
plot(hm)
```



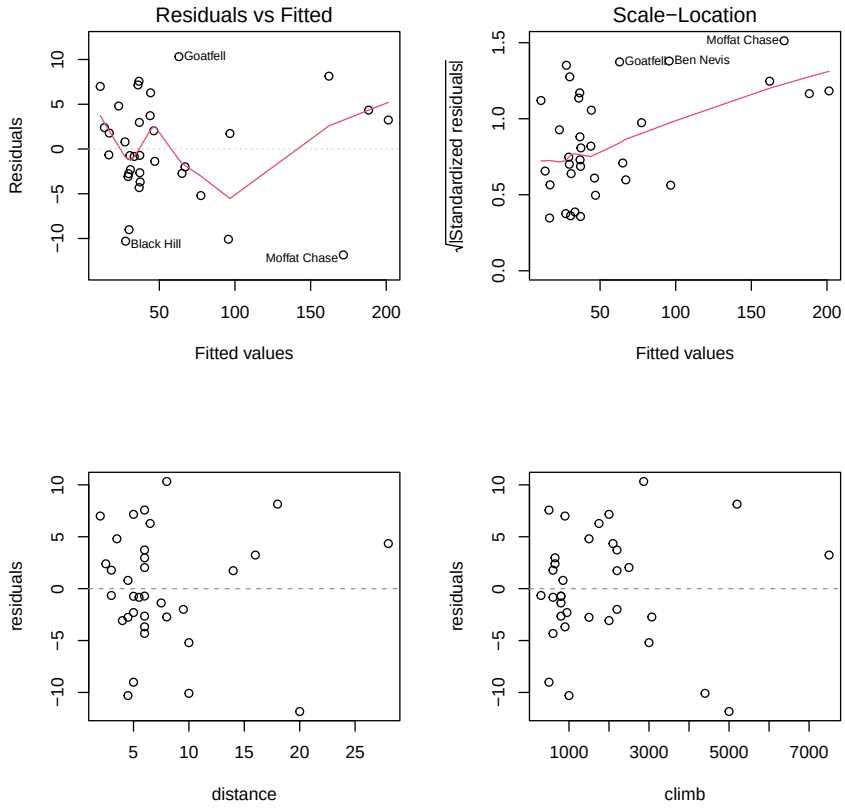

```r
par(mfrow = c(2, 2))
plot(hills1$dist, residuals(hm),
     ylab = "residuals", xlab = "distance")
abline(h = 0, lty = 2, col = "gray60")

plot(hills1$climb, residuals(hm),
     ylab = "residuals", xlab = "climb")
abline(h = 0, lty = 2, col = "gray60")
```



Bens of Jura has now a higher standardised residual and Cook's distance is very large. However, it does not seem to be an erroneous observation so we will proceed and come back to this issue at the end. The plot of the residuals against climb seem to indicate a very slight quadratic trend (but, again, data is very sparse for climb values above 3000). Let us then include a quadratic effect of this variable in the model and see what happens.

```
hm1 <- lm(time ~ dist + climb + I(climb^2), data = hills1)
summary(hm1)

##
## Call:
## lm(formula = time ~ dist + climb + I(climb^2), data = hills1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.8350  -2.7525  -0.6867   3.6026  10.3224
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.149e+00  2.354e+00  -1.338    0.191
## dist         6.597e+00  2.389e-01  27.616  < 2e-16 ***
## climb       -6.709e-04  2.085e-03  -0.322    0.750
## I(climb^2)   1.849e-06  2.838e-07   6.516 3.33e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.758 on 30 degrees of freedom
## Multiple R-squared:  0.9883,Adjusted R-squared:  0.9871
## F-statistic: 841.4 on 3 and 30 DF,  p-value: < 2.2e-16

par(mfrow = c(2, 2))
plot(hm1, which = 1)

plot(hm1, which = 3)

plot(hills1$dist, residuals(hm1),
     ylab = "residuals", xlab = "distance")
```

```r
abline(h = 0, lty = 2, col = "gray60")

plot(hills1$climb, residuals(hm1),
     ylab = "residuals", xlab = "climb")
abline(h = 0, lty = 2, col = "gray60")
```



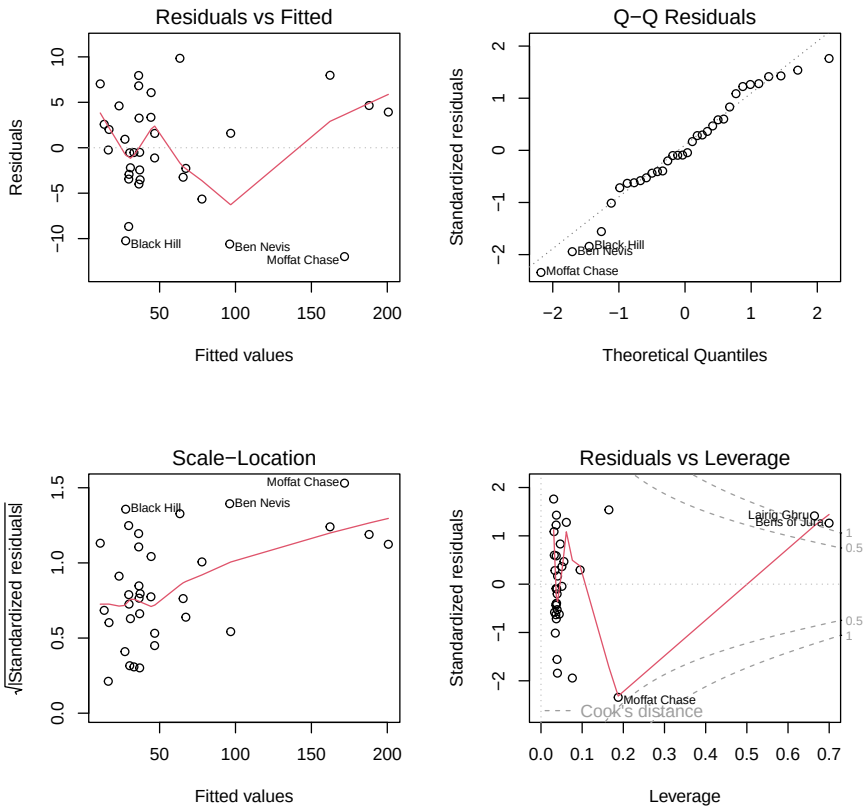
Residual plots seem better and the adjusted R squared has also increased. However, the effect of climb is no longer significant. Let us drop it and check the results.

```r
hm2 <- lm(time ~ dist + I(climb^2), data = hills1)
summary(hm2)
```

```
##
## Call:
## lm(formula = time ~ dist + I(climb^2), data = hills1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.9794  -3.1662  -0.3776   3.7822   9.8419
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.663e+00  1.703e+00  -2.151   0.0393 *
```

```
## dist         6.568e+00  2.173e-01  30.225   <2e-16 ***
## I(climb^2)   1.765e-06  1.083e-07  16.293   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.674 on 31 degrees of freedom
## Multiple R-squared:  0.9882,Adjusted R-squared:  0.9875
## F-statistic:  1300 on 2 and 31 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(hm2)
```



All terms are now significant and the adjusted R squared is comparable (even slightly higher). Let us now, based on this model, predict the winning time for a 7 mile race with 2400 feet of ascent.

```
predict(hm2, newdata = data.frame(dist = 7, climb = 2400),
        interval = "prediction")
```

```
##        fit      lwr      upr
## 1 52.47657 40.73149 64.22165
```

Let us see how this result changes if the two observations with a Cook's distance above one are removed from the dataset. These are observations number 7 (Bens of Jura) and 11 (Lairig Ghru).

```
hills2 <- hills1[-c(7,11), ]
hm3 <- lm(time ~ dist + I(climb^2), data = hills2)
predict(hm3, newdata = data.frame(dist = 7, climb = 2400),
        interval = "prediction")

##        fit      lwr      upr
## 1 52.13239 40.40349 63.86128
```

As can be observed, results are basically the same. So, a possible short report would be as follows.

Data on winning times for 34 Scottish hill races were analysed by linear modelling (a 35th time was excluded from the original dataset as it appears to represent a recording error). In particular the following simple model was used:

$$\texttt{time}_i = \beta_0 + \beta_1 \texttt{dist}_i + \beta_2 \texttt{climb}_i^2 + \epsilon_i \tag{1}$$

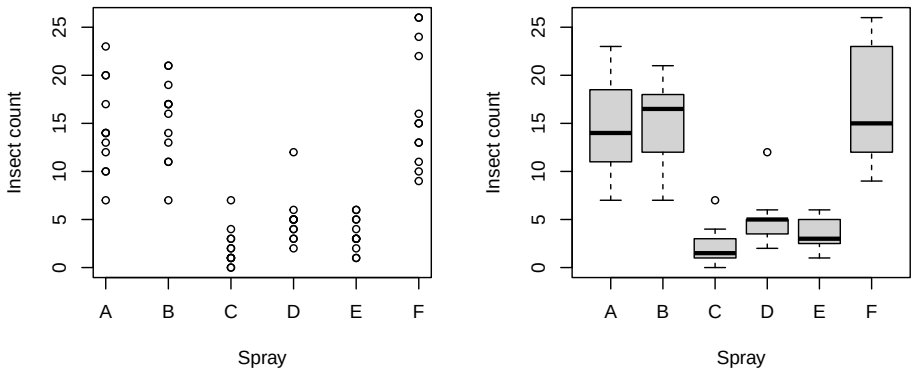
where `time` was winning time in minutes, `dist` was overall distance of race and `climb` was total ascent in the race. ϵ_i represents random variability in the times unexplained by the distance and height variables. The β_j are parameters of the model, which were estimated by fitting the model to the 34 data, by least squares. The data set analysed contained races with a range of distances and heights covering those proposed for the Whisky challenge, suggesting that the model should be reasonably reliable for predictive purposes.

The model structure was arrived at by starting with a model with linear dependence on `dist` and `climb`, but this was (slightly) improved by the model structure in (1). This model would predict a winning time of 52.5 minutes for the Whisky Challenge, with the 95% interval estimate for the wining time being being between 40.7and 64.2 minutes. However you should be aware that the act of offering substantial prizes for achieving particular times will in itself make the Whisky Challenge somewhat different to the other races used to estimate the model.

Note: This may not be the only possible correct solution.

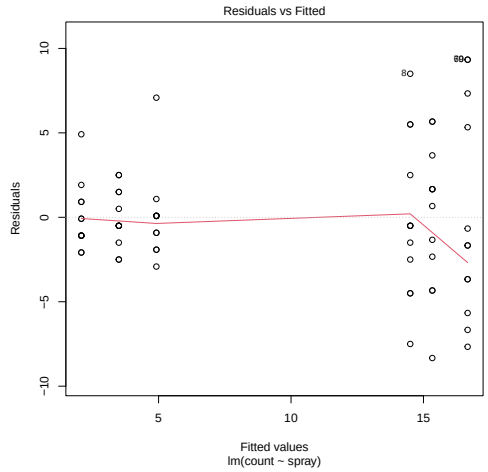
6. (a) As we have seen in the lecture notes, two possible options to plot the insect counts against the spray type are as follows.

```
par(mfrow = c(2, 2))
stripchart(count ~ spray, data = InsectSprays, vertical = TRUE, pch = 1,
           ylab = "Insect count", xlab = "Spray")
boxplot(count ~ spray, data = InsectSprays,
        ylab = "Insect count", xlab = "Spray")
```

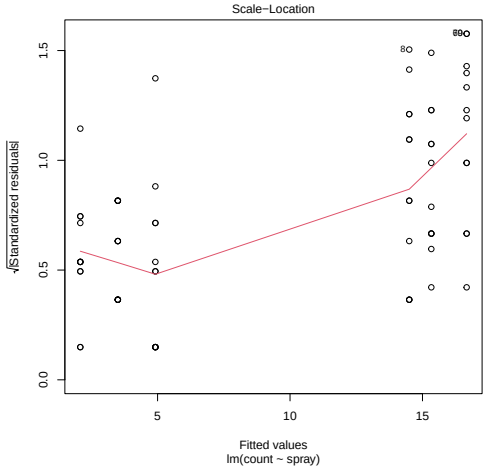


The plot suggests that sprays C, D, and E are most effective (lead to lower insects count). Also not that the variance of the response variable, insect counts, is not constant across the different types of sprays (there is a relationship between the variance and the size of the response).

(b)
```r
im0 <- lm(count ~ spray, data = InsectSprays)
plot(im0, which = 1)
```

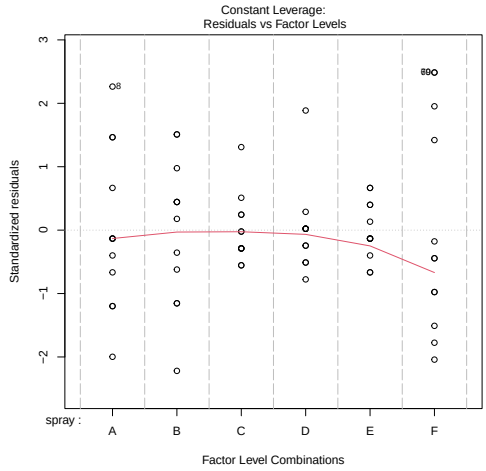



```r
plot(im0, which = 3)
```



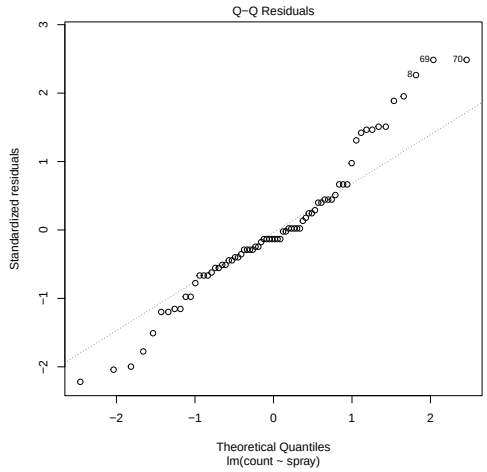
These plots confirm what we have already observed in (a): the constant variance assumption is not met. In particular, larger fitted values tend to have larger residuals. This further plot, shown below, confirms also what we have seen in (a).

```
plot(im0, which = 5)
```

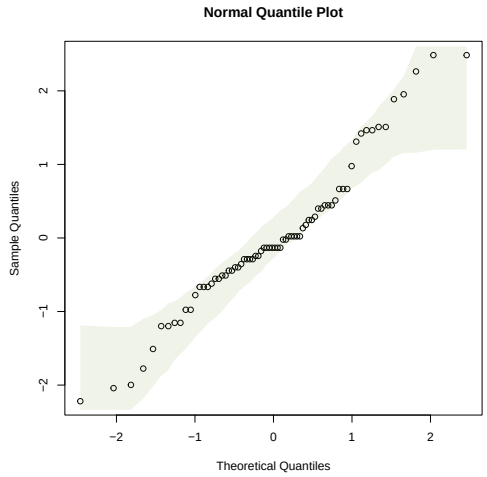


(c) Let us look at the qqplot.

```
plot(im0, which = 2)
library(ecostats)
```



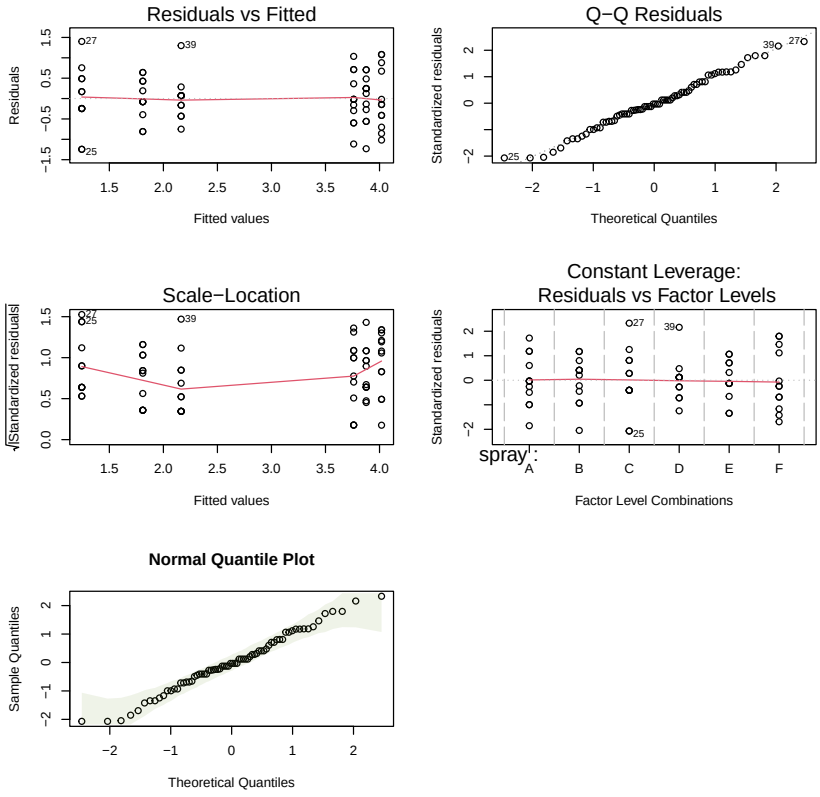

```
qqenvelope(im0)
```



The plots seem to indicate that the normality assumption is not blatantly violated, although there are a number of points close to the bounds of the simulation envelope.

(d) Let us fit the model to the square root of the inssect counts.

```
im1 <- lm(sqrt(count) ~ spray, data = InsectSprays)
par(mfrow = c(3, 2))
plot(im1)
qqenvelope(im1)
```



Much improved residual plots: nothing to suggest a problem with the assumptions.

(e) Let us look now at the summary of the model in fitted (d).

```
summary(im1)

##
## Call:
## lm(formula = sqrt(count) ~ spray, data = InsectSprays)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24486 -0.39970 -0.01902  0.42661  1.40089
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7607     0.1814  20.733  < 2e-16 ***
## sprayB        0.1160     0.2565   0.452    0.653
## sprayC       -2.5158     0.2565  -9.807 1.64e-14 ***
## sprayD       -1.5963     0.2565  -6.223 3.80e-08 ***
## sprayE       -1.9512     0.2565  -7.606 1.34e-10 ***
## sprayF        0.2579     0.2565   1.006    0.318
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6283 on 66 degrees of freedom
```

```
## Multiple R-squared:  0.7724,Adjusted R-squared:  0.7552
## F-statistic:  44.8 on 5 and 66 DF,  p-value: < 2.2e-16

head(model.matrix(im1))

##   (Intercept) sprayB sprayC sprayD sprayE sprayF
## 1           1      0      0      0      0      0
## 2           1      0      0      0      0      0
## 3           1      0      0      0      0      0
## 4           1      0      0      0      0      0
## 5           1      0      0      0      0      0
## 6           1      0      0      0      0      0
```

Note that we can write the model as

$$\sqrt{\text{count}}_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \beta_4 x_{4i} + \beta_5 x_{5i} + \epsilon_i, \quad \epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2),$$

where $x_{1i} = 1$ if spray B was used and $x_{1i} = 0$ otherwise, $x_{2i} = 1$ if spray C was used and $x_{2i} = 0$ otherwise, $x_{3i} = 1$ if spray D was used and $x_{3i} = 0$ otherwise, $x_{4i} = 1$ if spray E was used and $x_{4i} = 0$ otherwise, $x_{5i} = 1$ if spray F was used and $x_{5i} = 0$ otherwise. Spray A is the baseline/reference level, for which $x_{1i} = x_{2i} = x_{3i} = x_{4i} = x_{5i} = 0$. Therefore, for spray A, we have

$$\mathbb{E}(\sqrt{\text{count}}_i) = \beta_0,$$

and so the intercept is the expected square root of the insect count for spray A and this is estimated to be 3.76. .When $x_{1i} = 1$ and $x_{2i} = x_{3i} = x_{4i} = x_{5i} = 0$, we have that

$$\mathbb{E}(\sqrt{\text{count}}_i) = \beta_0 + \beta_1,$$

and so β_1 can be interpreted as the expected difference in square root counts between spray B and spray A and this is estimated to be $\hat{\beta}_1 = 0.1160$. Note that the estimated expected square root insect count for spray B is $\hat{\beta}_0 + \hat{\beta}_1 = 3.7607 + 0.1160 = 3.8767$. The remaining parameters have a similar interpretation.

(f) The previous summary suggested no real differences between sprays A and B and A and F, with C, D and E all significantly lower than A. Spray C gives the lowest expected square root count, so it might help interpretation to make this the reference level. Let's do that. Note that there is no need to re-check model as it's identical to model `im1`, just with a different parameterization, so residuals and fitted values are the same as `im1`.

```
InsectSprays$spray <- relevel(InsectSprays$spray, "C")
im2 <- lm(sqrt(count) ~ spray, data = InsectSprays)
summary(im2)

##
## Call:
## lm(formula = sqrt(count) ~ spray, data = InsectSprays)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1.24486 -0.39970 -0.01902  0.42661  1.40089
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.2449     0.1814   6.863 2.84e-09 ***
## sprayA        2.5158     0.2565   9.807 1.64e-14 ***
## sprayB        2.6318     0.2565  10.259 2.67e-15 ***
## sprayD        0.9195     0.2565   3.584 0.000641 ***
## sprayE        0.5646     0.2565   2.201 0.031238 *
## sprayF        2.7738     0.2565  10.813 2.98e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6283 on 66 degrees of freedom
## Multiple R-squared:  0.7724,Adjusted R-squared:  0.7552
## F-statistic:  44.8 on 5 and 66 DF,  p-value: < 2.2e-16
```

All other sprays lead to a significant higher number of insect counts than spray C, although D and E are not so very different and at 0.03 the p-value for E is a bit marginal. Given given that we are conducting 6 tests at once here. (e.g. if we conduct 20 tests of null hypotheses that are all true we still expect to see one ($= 20 \times 0.05$) p-value below 0.05).