Linear Models in R

A quadratic model

The `cars` data - contains data on stopping `distances` of cars against `speed` when the driver was first signalled to stop. - Need a linear dependence of `distance` on `speed`, due to reaction time after stop signal - The braking phase contributes a distance proportional to the square of `speed` to the total stopping distance because the kinetic energy of a car is proportional to the square of its `speed`.

$$\texttt{distance}_i = \beta_0 + \beta_1 \texttt{speed}_i + \beta_2 \texttt{speed}_i^2 + \epsilon_i$$

where the ϵ_i are i.i.d. $N(0, \sigma^2)$ random variables.

Physics suggests that $\beta_0 = 0$, but leave it in for the moment to provide a check.

Data

```r
cars[1:10,]  # check what data look like (first 10 observations)
```

```
##    speed dist
## 1      4    2
## 2      4   10
## 3      7    4
## 4      7   22
## 5      8   16
## 6      9   10
## 7     10   18
## 8     10   26
## 9     10   34
## 10    11   17
```

Fit model

```r
stop.model <- lm(dist~speed+I(speed^2),data=cars)
model.matrix(stop.model)[1:5,]
```

```
##   (Intercept) speed I(speed^2)
## 1           1     4         16
## 2           1     4         16
## 3           1     7         49
## 4           1     7         49
## 5           1     8         64
```

```r
summary(stop.model)
```

```
## 
## Call:
## lm(formula = dist ~ speed + I(speed^2), data = cars)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -28.720  -9.184  -3.188   4.628  45.152 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.47014   14.81716   0.167    0.868
## speed        0.91329    2.03422   0.449    0.656
## I(speed^2)   0.09996    0.06597   1.515    0.136
## 
## Residual standard error: 15.18 on 47 degrees of freedom
## Multiple R-squared:  0.6673, Adjusted R-squared:  0.6532 
## F-statistic: 47.14 on 2 and 47 DF,  p-value: 5.852e-12
```

Interpretation

Notice the `Coefficients` section of the table, in particular. It has a row for each model coefficient.

All the p-values for testing the single parameter hypotheses, $\mathrm{H}_0 : \beta_j = 0$, are very high.

Does this mean that the we can accept all such hypotheses, and conclude that all the parameter values could really be 0?

Interpretation

Absolutely not! Each test is only valid if the other parameters are allowed to take non-zero values: this is because the parameter estimators are not independent and so the p-values can not be either.

F-test: Final p-value on the last line of the summary. clearly such a hypothesis has no support from the data.

The high p-values in the summary do suggest that *something* could be removed from our model, and a sensible approach is to try removing the term with the highest p-value. This is the intercept, and on the physical grounds, given in the model motivation, we might expect the intercept to be zero.

```
stop.model2 <- lm(dist~speed+I(speed^2)-1,data=cars)
summary(stop.model2)
```

```
##
## Call:
## lm(formula = dist ~ speed + I(speed^2) - 1, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.836  -9.071  -3.152   4.570  44.986
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## speed       1.23903    0.55997   2.213  0.03171 *
## I(speed^2)  0.09014    0.02939   3.067  0.00355 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.02 on 48 degrees of freedom
## Multiple R-squared:  0.9133, Adjusted R-squared:  0.9097
## F-statistic: 252.8 on 2 and 48 DF,  p-value: < 2.2e-16
```

Now both β_1 and β_2 have much lower associated p-values: there are no grounds for dropping any more model terms. We should, of course, check residual plots for this model (we will do this later).

Confidence Intervals

90% CIs for β.

```
beta.hat <- coef(stop.model2)
V.beta <- vcov(stop.model2)
sigma.beta <- sqrt(diag(V.beta))
beta.hat + qt(.95,df=47)*sigma.beta
```

```
##      speed I(speed^2)
##  2.1786198  0.1394517
```

```
beta.hat - qt(.95,df=47)*sigma.beta
```

```
##      speed I(speed^2)
## 0.29944014 0.04082589
```

So, for example, a 90% CI for β_1 is $(0.30, 2.18)$.

anova

Consider testing the null model

$\texttt{distance}_i = \beta\texttt{speed}_i^2 + \epsilon_i$ against the full model
$\texttt{distance}_i = \beta_0 + \beta_1\texttt{speed}_i + \beta_2\texttt{speed}_i^2 + \epsilon_i$

We can use the `anova` function to perform the appropriate F-test:

```r
stop.model0 <-lm(dist~I(speed^2) - 1,data=cars)
anova(stop.model,stop.model0)
```

```
## Analysis of Variance Table
##
## Model 1: dist ~ speed + I(speed^2)
## Model 2: dist ~ I(speed^2) - 1
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     47 10825
## 2     49 11936 -2   -1111.2 2.4123 0.1006
```

Apply anova() to just one model

```r
anova(stop.model)
```

```
## Analysis of Variance Table
##
## Response: dist
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## speed       1 21185.5 21185.5  91.986 1.211e-12 ***
## I(speed^2)  1   528.8   528.8   2.296    0.1364
## Residuals  47 10824.7   230.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpretation anova()

`anova()` with single argument: table based on the sequence of ever simpler models obtained by removing terms sequentially

- Each row in the model tests one model in the sequence against the closest more complicated model in the sequence.
- at each stage testing the null hypothesis that the current version of the model is correct, against the alternative that the previous version was right.
- F-ratio tests are used each time, but $\hat{\sigma}^2$ according to the original model is always used as the denominator of the F ratio statistic.
- The tables have to be read from bottom row up.

drop1()

```r
drop1(stop.model, test="F")
```

```
## Single term deletions
##
## Model:
## dist ~ speed + I(speed^2)
##            Df Sum of Sq   RSS    AIC F value Pr(>F)
## <none>                  10825 274.88
## speed       1     46.42 10871 273.09  0.2016 0.6555
## I(speed^2)  1    528.81 11354 275.26  2.2960 0.1364
```

Produces the table obtained by F-ratio test comparison of the full model with each of the models produced by dropping a single effect from the full model.

Checking model assumptions

Checking model assumptions is necessary because inference (Confidence intervals and hypothesis tests) relies on valid model assumptions.

Examine a variety of diagnostic plots, always looking for evidence of any violation of the linear modelling assumptions:

1. The ϵ_i are independent.
2. The ϵ_i have constant variance.
3. The ϵ_i follow a normal distribution.

Residuals

Model checking is always based on looking at *residuals*. Recall the residuals are

$$\hat{\epsilon} = \mathbf{y} - \hat{\mu} \quad \text{where} \quad \hat{\mu} = \mathbf{X}\hat{\beta}.$$

If the model is adequate then the residuals should behave like observations of the ϵ_i r.v.s.

In other word the ϵ_i should look like a collection of independent observations of an $N(0, \sigma^2)$ r.v.

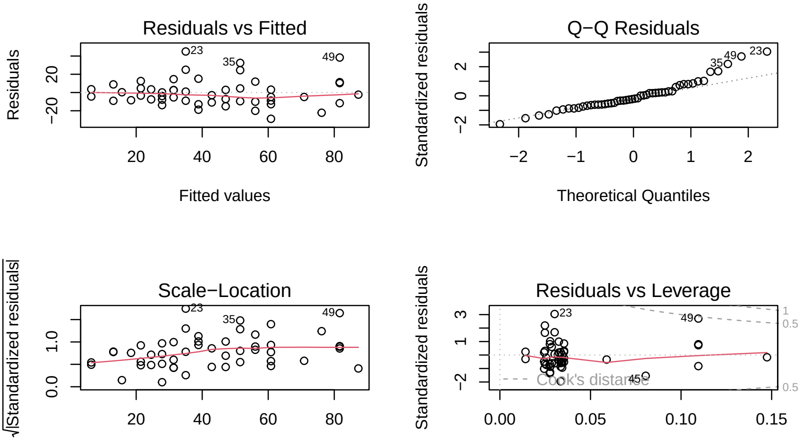
Actually this is only approximately true, but the residuals can be standardized to have constant variance, by defining

$$\hat{\epsilon}_i^p = \hat{\epsilon}_i / \sqrt{1 - A_{ii}}$$

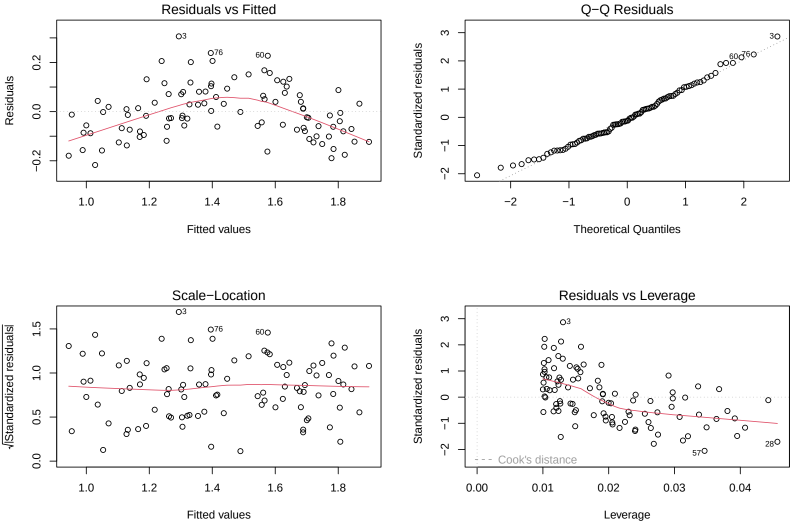
where $\mathbf{A}$ is the model 'influence matrix' and p is a label (rather than a power).

Residual plots

```
par(mfrow=c(2,2))
plot(stop.model2)
```



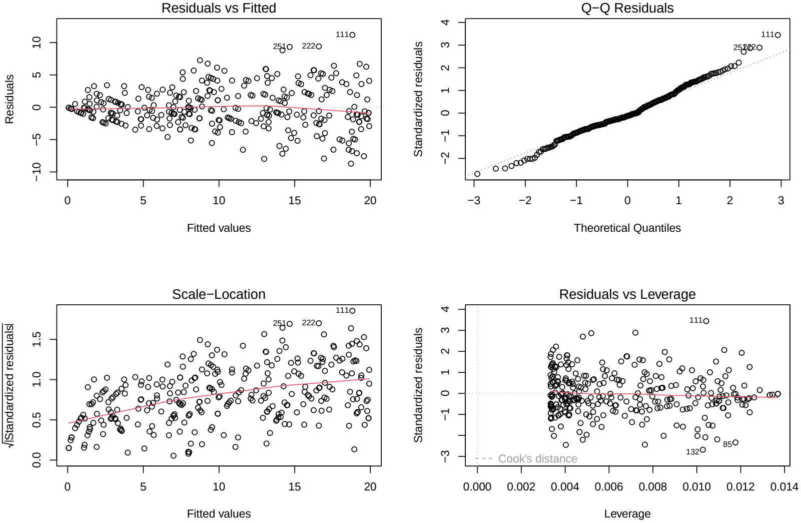
Problematic residual plots



Notice the clear trend in the mean of the residuals plotted against fitted values. This sort of pattern means that the independence assumption is violated, and usually indicates that something is missing from the model. Perhaps a dependence on the square of some predictor variable?

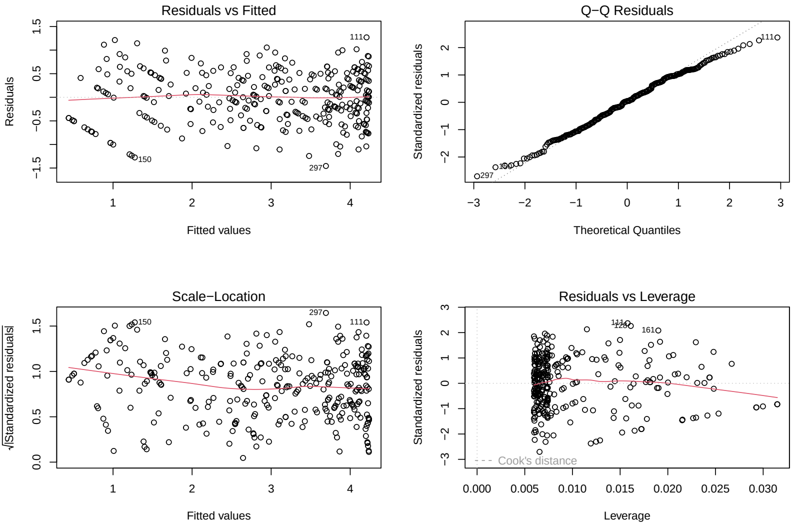
It can be helpful to plot the residuals against the predictor variables as well as against time to investigate violations of the independence assumption.

Problematic residual plots . . .



In this case the mean seems to be more or less zero for all the residuals, as it should be. However the variance of the residuals shows a clear increase with the mean. This pattern of increasing variance with increasing mean is quite a common pattern in practice, and obviously violates the constant variance assumption. In this case it can sometimes help to switch to modelling some transformation of the orginal response variable.

model with transformed response (sqrt)



r^2: How close is the fit?

The r^2 statistic provide a measure of how closely a model fits the response data.

r^2 measures the proportion (or percentage) of the variance in the original data that is ‘explained’ by the fitted model.

The proportion of variance left unexplained, is the observed variance of the residuals, divided by the observed variance of the response data.

The proportion variance explained is hence one minus the same ratio, so

$$r^2 = 1 - \frac{\sum_i \hat{\epsilon}_i^2 / n}{\sum_i (y_i - \bar{y})^2 / n}.$$

Adusted r^2

The conventional definition of r^2 overestimates how well a model is doing because it uses biased variance estimators.

The *adjusted* r^2 avoids this, to some extent, by using unbiased estimators:

$$r^2_{\mathrm{adj}} = 1 - \frac{\sum_i \hat{\epsilon}_i^2/(n-p)}{\sum_i (y_i - \bar{y})^2/(n-1)},$$

where p is the number of model parameters. Note that r^2_{adj} can be negative.

A high r^2 is always better than a low r^2, but a low r^2 should not necessarily be taken as indicating that a model is poor: some response data simply contain a good deal of random variability about which nothing can be done.

(adjusted) r^2 for “ model

```r
names(summary(stop.model2))
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```r
summary(stop.model2)$r.squared
```

```
## [1] 0.9132838
```

```r
summary(stop.model2)$adj.r.squared
```

```
## [1] 0.9096706
```

Prediction

Predict the expected response from the model at new values of the predictor variables.

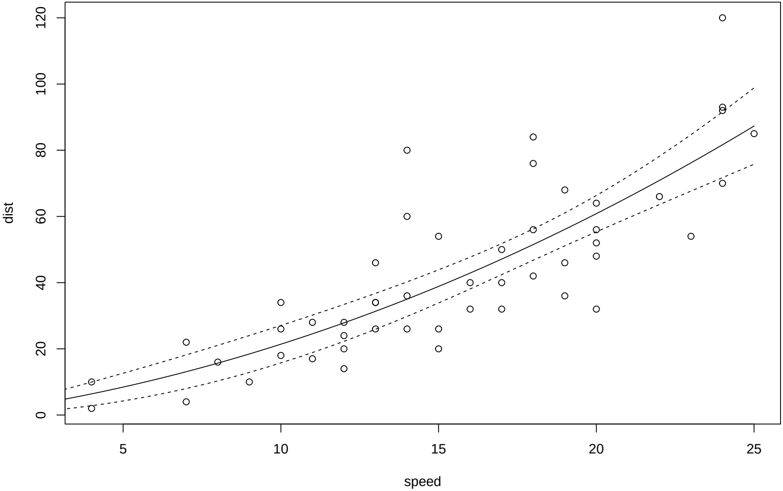
Create predictor matrix $\mathbf{X^p}$, in exactly same way as the original values were used to create $\mathbf{X}$. The predictions are

$$\mu^{\mathbf{p}} = \mathbf{X^p}\widehat{\beta}$$

and

$$\hat{\mu}^{p} \sim N(\mu^{p}, \mathbf{X^p}(\mathbf{X^T X})^{-\mathbf{1}}\mathbf{X^{pT}}\sigma^{\mathbf{2}})$$

```
with(cars,plot(speed,dist))
dat <- data.frame(speed=seq(0,25, length=100))
fv <- predict(stop.model2, newdata=dat, se=TRUE)
lines(dat$speed, fv$fit)
lines(dat$speed, fv$fit + 2 * fv$se.fit, lty=2)
lines(dat$speed, fv$fit - 2 * fv$se.fit, lty=2)
```



A model with factors

The data frame PlantGrowth contains data on plant weights after growth under 3 alternative experimental treatments. The data frame contains columns weight and group, the latter indicating which treatment group the weight measurement corresponds to. The groups are ctrl, trt1 and trt2.

```r
summary(PlantGrowth)
```

```
##      weight       group   
##  Min.   :3.590   ctrl:10  
##  1st Qu.:4.550   trt1:10  
##  Median :5.155   trt2:10  
##  Mean   :5.073            
##  3rd Qu.:5.530            
##  Max.   :6.310            
```

Model

$$E(\texttt{weight}_i) = \beta_j \text{ if observation } i \text{ is in group } j,$$

where groups 1, 2 and 3 are `ctrl`, `trt1` and `trt2`, respectively. We have already seen how such a model can be made into a linear model, using a design matrix of zeros and ones, but how can such models be set up in R?

Factor variables in R

In R such categorical variables can be given class `factor` and this indicates that its values are to be taken as identifiers of groups, and that it should be handled specially when used to set up a model matrix.

```r
class(PlantGrowth$group)
```

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

```r
PlantGrowth$group
```

```
##  [1] ctrl ctrl ctrl ctrl ctrl ctrl ctrl ctrl ctrl ctrl trt1 trt1 trt1 trt1 trt1
## [16] trt1 trt1 trt1 trt1 trt1 trt2 trt2 trt2 trt2 trt2 trt2 trt2 trt2 trt2 trt2
## Levels: ctrl trt1 trt2
```

Fit the model

```r
plant.mod <- lm(weight~group-1,data=PlantGrowth)
model.matrix(plant.mod)[1:5,]
```

```
##   groupctrl grouptrt1 grouptrt2
## 1         1         0         0
## 2         1         0         0
## 3         1         0         0
## 4         1         0         0
## 5         1         0         0
```

```r
summary(plant.mod)
```

```
##
## Call:
## lm(formula = weight ~ group - 1, data = PlantGrowth)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0710 -0.4180 -0.0060  0.2627  1.3690
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## groupctrl   5.0320     0.1971   25.53   <2e-16 ***
## grouptrt1   4.6610     0.1971   23.64   <2e-16 ***
## grouptrt2   5.5260     0.1971   28.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6234 on 27 degrees of freedom
## Multiple R-squared:  0.9867, Adjusted R-squared:  0.9852
## F-statistic: 665.5 on 3 and 27 DF,  p-value: < 2.2e-16
```

To test the hypothesis whether $\beta_1{=}\beta_2 = \beta_3 = 0$ we look at the result of the `F-statistic` given at the bottom of `summary`, and the p-value tells us that there is evidence to reject the null hypothesis of all group effects being zero.