Linear Models

Aindrila Garai, PhD at University of Bristol

```
library(gamair)
library(ggplot2)

data("hubble")
dim(hubble)
```

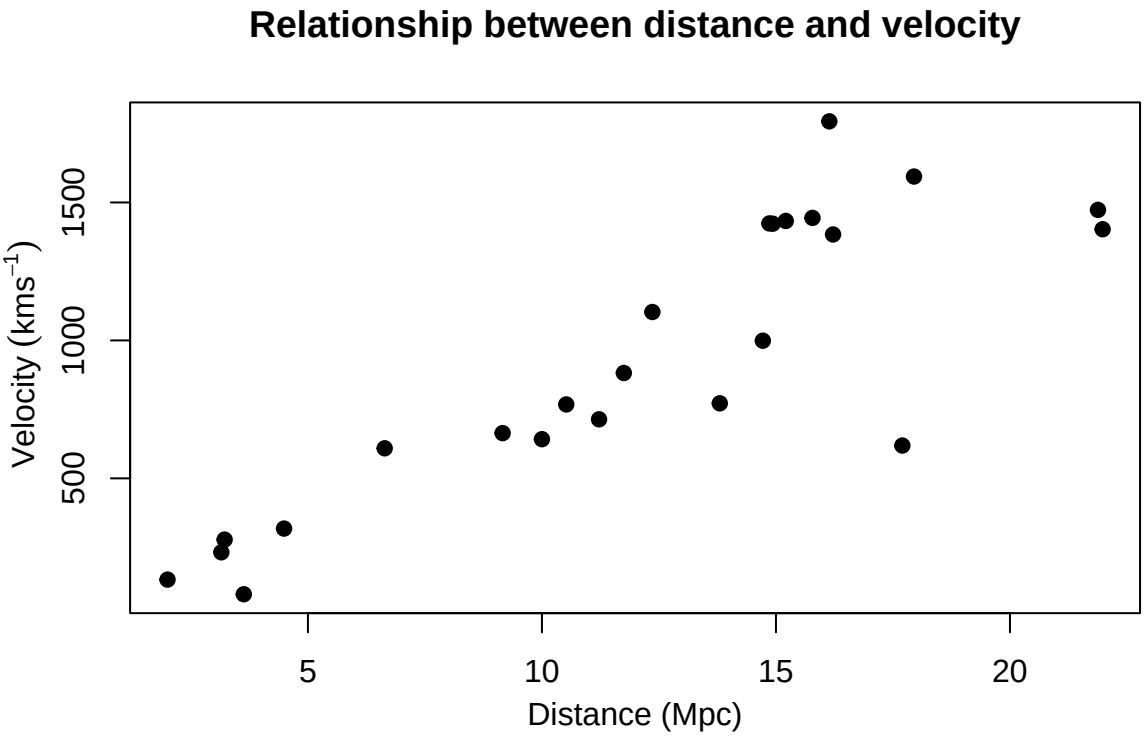
```
## [1] 24  3
```

```
head(hubble)
```

```
##     Galaxy    y     x
## 1  NGC0300  133  2.00
## 2  NGC0925  664  9.16
## 3 NGC1326A 1794 16.14
## 4  NGC1365 1594 17.95
## 5  NGC1425 1473 21.88
## 6  NGC2403  278  3.22
```

According to Hubble's law, $y = \beta x$, where y is the relative velocity of any two galaxies separated by distance x, and β is 'Hubble's constant', gives the approximate age of the universe.

```
par(mgp = c(2.1,1,0), mar = c(5,4,4,2)+0.1)
plot(hubble$x, hubble$y, pch = 19, xlab = 'Distance (Mpc)',
     ylab = expression(~Velocity~(kms^{-1})),
     main = "Relationship between distance and velocity")
```



We have to find β and its consistency with data.

Applying simple linear model, -1 term indicates that the model has no intercept term.

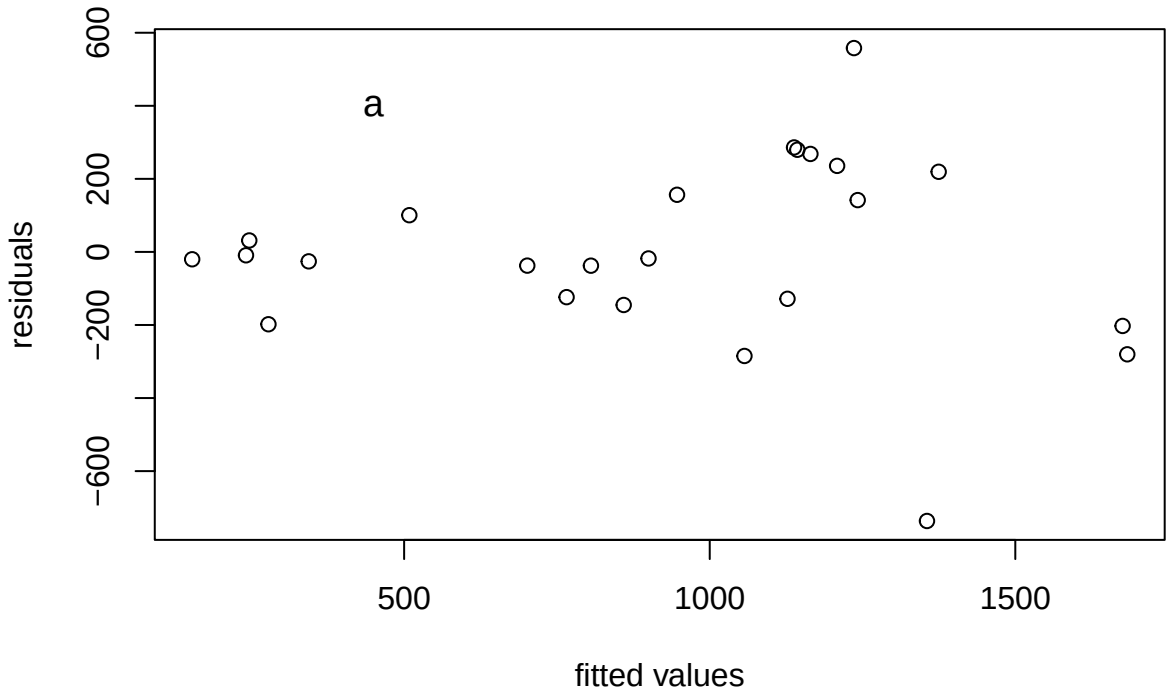
```
library(gamair)
data(hubble)
hub.mod <- lm(y ~ x - 1, data=hubble)
summary(hub.mod) # beta_hat and the estimate of se(beta_hat)
```

```
##
## Call:
## lm(formula = y ~ x - 1, data = hubble)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -736.5 -132.5  -19.0  172.2  558.0
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x   76.581      3.965   19.32 1.03e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 258.9 on 23 degrees of freedom
## Multiple R-squared:  0.9419, Adjusted R-squared:  0.9394
## F-statistic: 373.1 on 1 and 23 DF,  p-value: 1.032e-15
```

Before applying any model, it is important to check the model assumptions errors are independent and all have the same variance. To do this is to examine residual plots.

```
fitted <- hub.mod$coefficients*hubble$x
residuals <- hubble$y - fitted
```

```
plot(fitted(hub.mod),residuals(hub.mod),xlab="fitted values",
     ylab="residuals")
text(450, 400, "a", cex = 1.2)
```



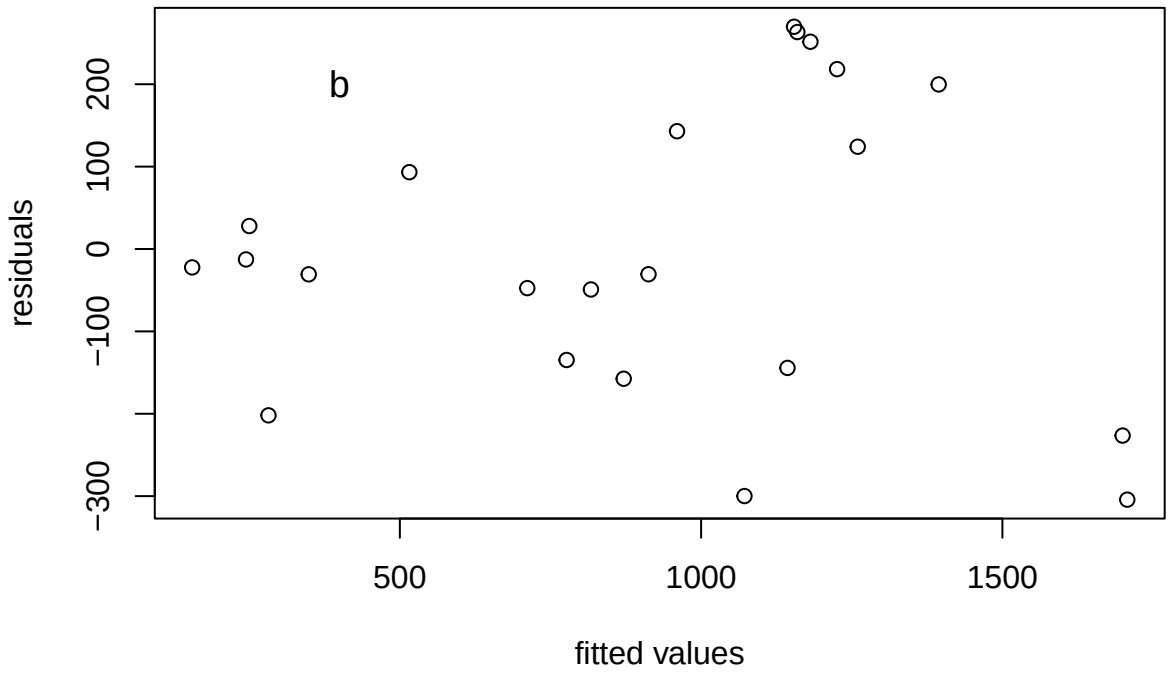
If the model is correct, there would be random scatter of residuals around zero, with no trend in either the mean of the residuals, or their variability, A trend in the mean violates the independence assumption a trend in the variability violates the constant variance assumption.

Same model fitted to data with two substantial outliers omitted which implies constant variance assumption problem.

```
hub.mod1 <- lm(y ~ x - 1,data=hubble[-c(3,15),])
summary(hub.mod1)
```

```
##
## Call:
## lm(formula = y ~ x - 1, data = hubble[-c(3, 15), ])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -304.3 -141.9  -26.5  138.3  269.8
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x    77.67       2.97   26.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 180.5 on 21 degrees of freedom
## Multiple R-squared:  0.9702, Adjusted R-squared:  0.9688
## F-statistic: 683.8 on 1 and 21 DF,  p-value: < 2.2e-16
```

```
plot(fitted(hub.mod1),residuals(hub.mod1),
       xlab="fitted values",ylab="residuals")
text(400, 200, "b", cex = 1.2)
```



- Estimating age of the universe:

```
hubble.const <- c(coef(hub.mod),coef(hub.mod1))/3.09e19
age <- 1/hubble.const
age/(60^2*24*365)
```

```
##           x           x
## 12794692825 12614854757
```

- Calculating the p-value:

```
cs.hubble <- 163000000
t.stat <- (coef(hub.mod1)-cs.hubble)/vcov(hub.mod1)[1,1]^0.5
pt(t.stat,df=21)*2 # 2 because of |T| in p-value definition
```

```
##             x
## 3.906388e-150
```

- Estimating confidence interval of age:

```
qt(c(0.025,0.975),df=21) # the range of the middle 95% of t_21 random variables
```

```
## [1] -2.079614  2.079614
```

- 95% confidence interval:

```
sigb <- summary(hub.mod1)$coefficients[2]
h.ci <- coef(hub.mod1)+qt(c(0.025,0.975),df=21)*sigb
h.ci
```

```
## [1] 71.49588 83.84995
```

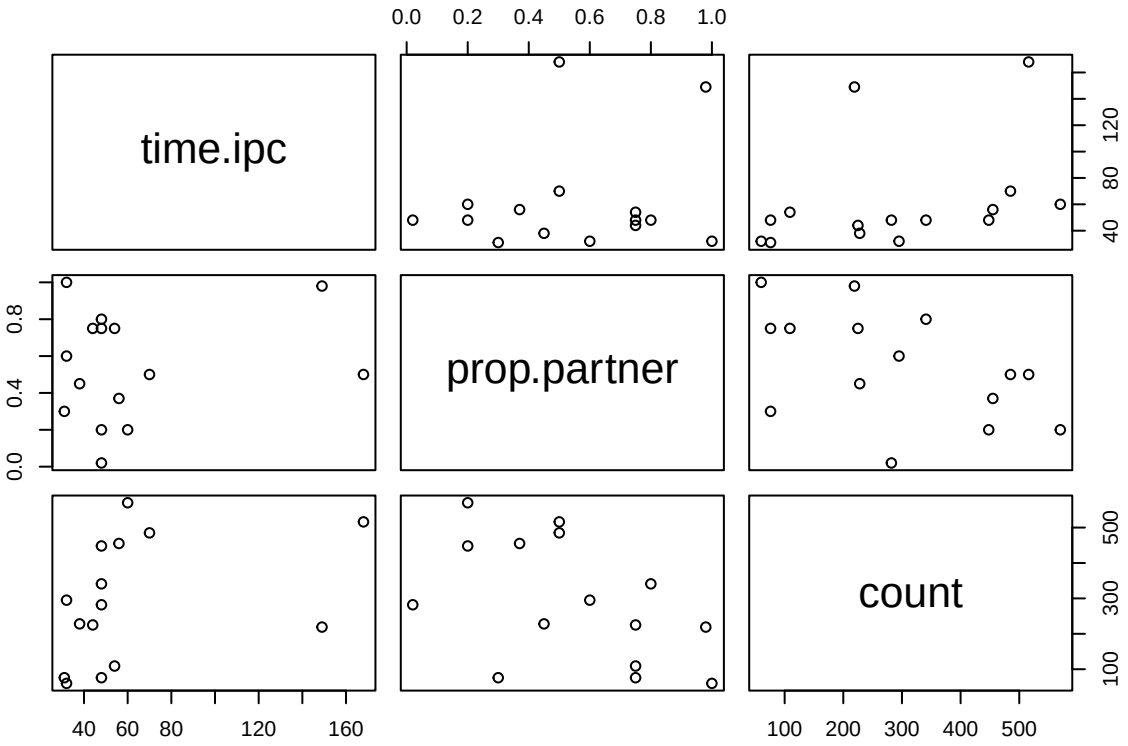
```
# converted to a confidence interval for the age of the universe
h.ci <- h.ci*60^2*24*365.25/3.09e19 # convert to 1/years
sort(1/h.ci)
```

```
## [1] 11677548698 13695361072
```

Data on sperm count, time since last copulation and proportion of that time spent together, for single copulations, from 15 heterosexual couples. The first column has been excluded from the plot as it simply contains subject identification labels.

```
library(gamair)
data("sperm.comp1")

pairs(sperm.comp1[,-1])
```



The clearest pattern seems to be of some decrease in sperm count as the proportion of time spent together increases.

$y_i = \beta_0 + t_i\beta_1 + p_i\beta_2 + \epsilon_i$ - a reasonable model.

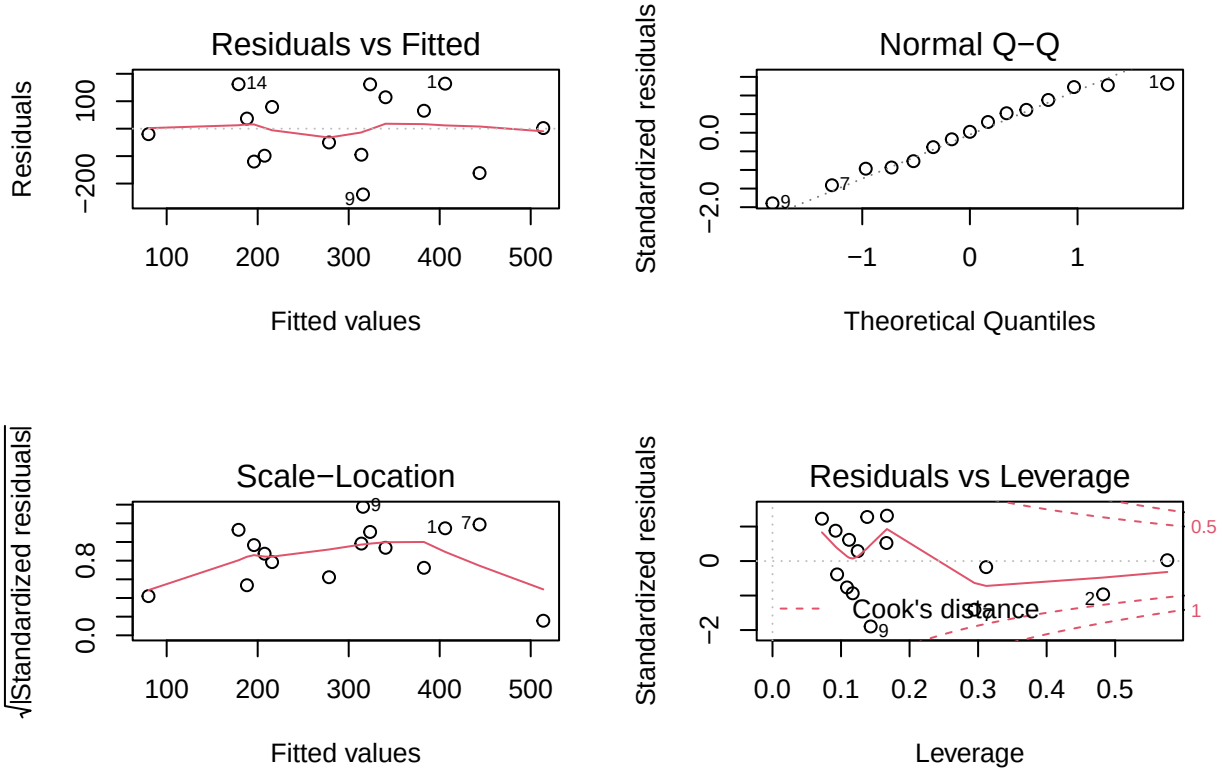
```
sc.mod1 <- lm(count ~ time.ipc + prop.partner, sperm.comp1)
model.matrix(sc.mod1)
```

```
##    (Intercept) time.ipc prop.partner
## 1            1       60         0.20
## 2            1      149         0.98
## 3            1       70         0.50
## 4            1      168         0.50
## 5            1       48         0.20
## 6            1       32         1.00
## 7            1       48         0.02
## 8            1       56         0.37
## 9            1       31         0.30
```

```
## 10           1       38         0.45
## 11           1       48         0.75
## 12           1       54         0.75
## 13           1       32         0.60
## 14           1       48         0.80
## 15           1       44         0.75
## attr(,"assign")
## [1] 0 1 2
```

By default, the model will include an intercept term, unless it is suppressed by a '-1' in the formula. Having fitted the model, it is important to check the plausibility of the assumptions, graphically.

```
par(mfrow=c(2,2)) # split the graphics device into 4 panels
plot(sc.mod1)
```




```
sc.mod1
```

```
##
## Call:
## lm(formula = count ~ time.ipc + prop.partner, data = sperm.comp1)
##
## Coefficients:
##  (Intercept)      time.ipc  prop.partner
##      357.418         1.942      -339.560
```

In two of the plots the residuals have been scaled. If the model assumptions are met, then this standardization should result in residuals that look like N(0,1) random deviates.

- Upper left plot- The residuals should be evenly scattered above and below zero.
- Lower left plot- scale-location plot. The raw residuals are standardized.
- Upper right panel- normal QQ (quantile-quantile) plot. The standardized residuals are sorted and plotted against the quantiles of a standard normal distribution. If the residuals are normally distributed

then the resulting plot should look like a straight line relationship.
- Lower right- the standardized residuals against the leverage of each datum.

The summary of sc.mod1 suggests that there is evidence that the model is better than one including just a constant (p-value = 0.02554). There is quite clear evidence that prop.partner is important in predicting sperm count (p-value = 0.019), but less evidence that time.ipc matters (p-value = 0.053). Finally note that the model leaves most of the variability in count unexplained, since adjusted r-squared is only 37%.

The 9th datum is flagged in all four plots, this subject has quite a low count, but not the lowest in the frame. It is hard to see a good reason to remove it, particularly since, if anything, it is obscuring the relationship, rather than exaggerating it.

```
sperm.comp1[9,]
```

```
##   subject time.ipc prop.partner count
## 9       P       31          0.3    76
```

One point to consider is whether prop.partner is the most appropriate predictor variable. Perhaps the total time spent together (in hours) would be a better predictor.

```
sc.mod2 <- lm(count ~ time.ipc + I(prop.partner*time.ipc), sperm.comp1)
```

To avoid overcomplicated models, dependent on irrelevant predictor variables, for reasons of interpretability and efficiency, we do model selection. For the sperm competition model the p-value for time.ipc is greater than 0.05, so this predictor might be a candidate for dropping.

```
sc.mod3 <- lm(count ~ prop.partner, sperm.comp1)
summary(sc.mod3)
```

```
##
## Call:
## lm(formula = count ~ prop.partner, data = sperm.comp1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -287.83 -111.30   18.84  117.45  210.61
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    451.50      86.23   5.236 0.000161 ***
## prop.partner  -292.23     140.40  -2.081 0.057727 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 154.3 on 13 degrees of freedom
## Multiple R-squared:   0.25,  Adjusted R-squared:  0.1923
## F-statistic: 4.332 on 1 and 13 DF,  p-value: 0.05773
```

Dropping time.ipc has made the estimate of the parameter multiplying prop.partner less precise: indeed this term also has a p-value greater than 0.05 according to this new fit. The new model has a much reduced r^2 , while the model's overall p-value does not give strong evidence that it is better than a model containing only an intercept.

One alternative of model selection is to try and find the model that gets as close as possible to the true model ie, to find the model which can predict the $E(y_i)$. Selecting models in order to minimize Akaike's Information Criterion (AIC).

```
sc.mod4 <- lm(count ~ 1, sperm.comp1) # null model
AIC(sc.mod1,sc.mod3,sc.mod4)
```

```
##         df      AIC
## sc.mod1  4 194.7346
## sc.mod3  3 197.5889
## sc.mod4  2 199.9031
```

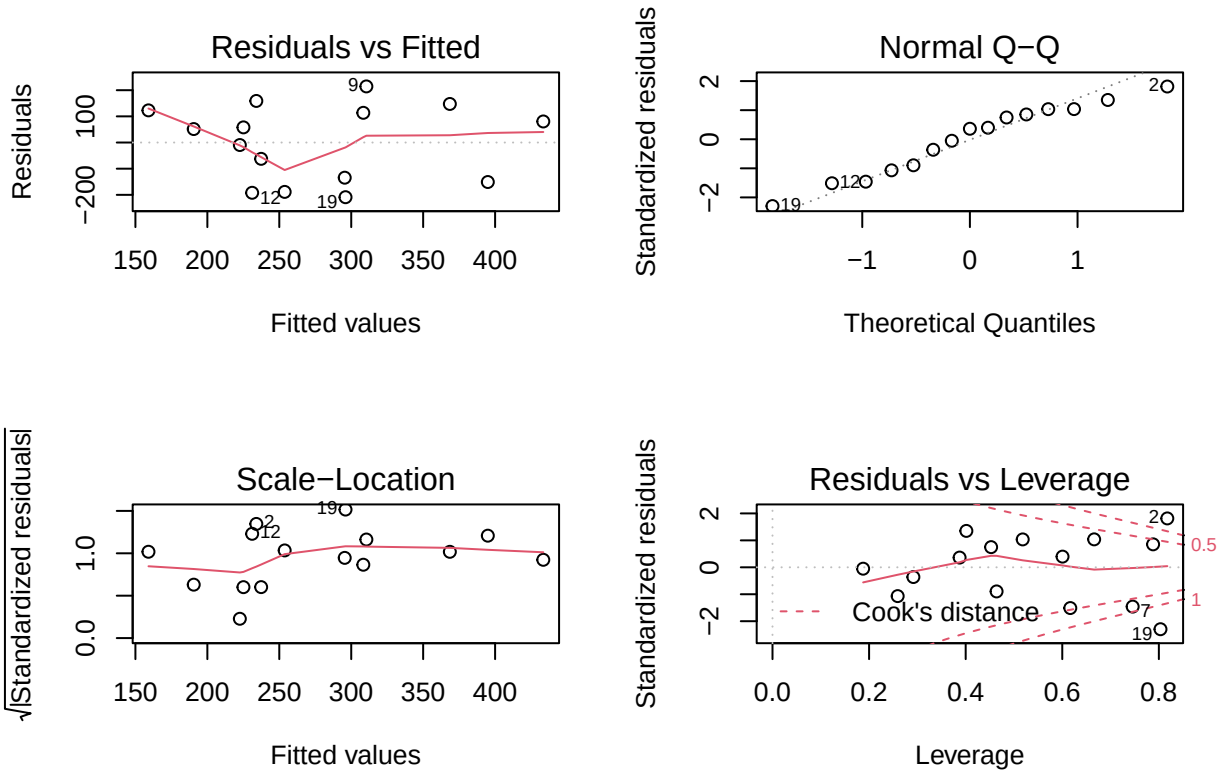
It also suggests that the model with both time.ipc and prop.partner is best. So, on the basis of sperm.comp1, there seems to be reasonable evidence that sperm count increases with time.ipc but decreases with prop.partner.

The 2nd dataset contains data on median sperm count, over multiple copulations, for 24 heterosexual couples, along with the weight, height and age of the male and female of each couple, and the volume of one teste of the male.

```
data("sperm.comp2")
```

A reasonable model including linear effects of all predictors, i.e., $count_i = \beta_0 + \beta_1 f.age_i + \beta_2 f.weight_i + \beta_3 f.height_i + \beta_4 m.age_i + \beta_5 m.weight_i + \beta_6 m.height_i + \beta_7 m.vol + \epsilon_i$

```
par(mfrow=c(2,2))
sc2.mod1 <- lm(count ~ f.age + f.height + f.weight + m.age +
                 m.height + m.weight + m.vol, sperm.comp2)
plot(sc2.mod1)
```




```
summary(sc2.mod1)
```

```
##
## Call:
## lm(formula = count ~ f.age + f.height + f.weight + m.age + m.height +
##     m.weight + m.vol, data = sperm.comp2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -209.03 -142.71   51.46  118.27  214.47
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1098.518   1997.984  -0.550    0.600
## f.age          10.798     22.755   0.475    0.650
## f.height       -4.639     10.910  -0.425    0.683
## f.weight       19.716     35.709   0.552    0.598
## m.age          -1.722     10.219  -0.168    0.871
## m.height        6.009     10.378   0.579    0.581
## m.weight       -4.619     12.655  -0.365    0.726
## m.vol           5.035     17.652   0.285    0.784
##
## Residual standard error: 205.1 on 7 degrees of freedom
##   (9 observations deleted due to missingness)
## Multiple R-squared:  0.2192, Adjusted R-squared:  -0.5616
## F-statistic: 0.2807 on 7 and 7 DF,  p-value: 0.9422
```

Datum 19 (the male of this couple is heavy) appears to produce the most extreme point on all 4 plots. The adjusted r^2 is actually negative, an indication that we have irrelevant predictors in the model.

Model Selection: - 'step' function in R to perform model selection automatically, it takes a fitted model and repeatedly drops the term that leads to the largest decrease in AIC. - 'Backwards model selection', by repeatedly removing the single term with highest p-value, above some threshold, and then refitting the resulting reduced model, until all terms have significant p-values.

```
sc2.mod2 <- lm(count ~ f.age + f.height + f.weight +
                 m.height + m.weight + m.vol,sperm.comp2)
summary(sc2.mod2)
```

```
##
## Call:
## lm(formula = count ~ f.age + f.height + f.weight + m.height +
##     m.weight + m.vol, data = sperm.comp2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -201.98 -144.17   52.52  123.25  219.83
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1054.770   1856.843  -0.568    0.586
## f.age           8.847     18.359   0.482    0.643
## f.height       -5.119      9.871  -0.519    0.618
## f.weight       20.259     33.334   0.608    0.560
## m.height        6.033      9.727   0.620    0.552
## m.weight       -4.473     11.834  -0.378    0.715
## m.vol           4.506     16.281   0.277    0.789
##
## Residual standard error: 192.3 on 8 degrees of freedom
##   (9 observations deleted due to missingness)
## Multiple R-squared:  0.216,  Adjusted R-squared:  -0.372
## F-statistic: 0.3674 on 6 and 8 DF,  p-value: 0.8805
```

The reason for dropping one term at a time: it is quite possible for some remaining terms to have their p-values massively reduced. If two terms are highly correlated it is possible for both to be significant individually, but both to have very high p-values if present together. If we were to drop several terms from a model at once,

we might miss effects.

Proceeding with backwards selection, we would drop m.vol, m.weight, f.height, m.height and finally f.age.

```
sc2.mod7 <- lm(count ~ f.weight,sperm.comp2)
summary(sc2.mod7)
```

```
##
## Call:
## lm(formula = count ~ f.weight, data = sperm.comp2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -209.741 -119.709    7.465   92.913  273.053
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1002.281    489.352  -2.048   0.0539 .
## f.weight       22.397      8.629   2.595   0.0173 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 147.3 on 20 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.252,  Adjusted R-squared:  0.2146
## F-statistic: 6.736 on 1 and 20 DF,  p-value: 0.0173
```

This model does appear to be better than a model containing only a constant, only female weight influences sperm count.

Based on the residual plots we need to re-analyze the data without observation 19, before final conclusions.

```
sc <- sperm.comp2[-19,]
sc3.mod1 <- lm(count ~ f.age + f.height + f.weight + m.age +
                  m.height + m.weight + m.vol, sc)
summary(sc3.mod1)
```

```
##
## Call:
## lm(formula = count ~ f.age + f.height + f.weight + m.age + m.height +
##     m.weight + m.vol, data = sc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -104.779  -57.334    2.707   67.302  101.556
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1687.406   1251.338   1.348   0.2262
## f.age         55.248     15.991   3.455   0.0136 *
## f.height      21.381      8.419   2.540   0.0441 *
## f.weight     -88.992     31.737  -2.804   0.0310 *
## m.age        -17.210      6.555  -2.626   0.0393 *
## m.height     -11.321      6.869  -1.648   0.1504
## m.weight       6.885      7.287   0.945   0.3812
## m.vol         48.996     13.938   3.515   0.0126 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 109.8 on 6 degrees of freedom
##   (9 observations deleted due to missingness)
## Multiple R-squared:  0.7861, Adjusted R-squared:  0.5366
## F-statistic:  3.15 on 7 and 6 DF,  p-value: 0.09167
```

m.vol now has the lowest p-value. Repeating the backwards selection process, every term now drops out except for m.vol, leading to the much less interesting conclusion that the data only really supply evidence that size of testes influences sperm count.

- A follow-up: The same couples feature in both datasets and are identified by label

```
sperm.comp1$m.vol <-
  sperm.comp2$m.vol[sperm.comp2$pair %in% sperm.comp1$subject]

sc1.mod1 <- lm(count ~ m.vol, sperm.comp1)
summary(sc1.mod1)
```

```
##
## Call:
## lm(formula = count ~ m.vol, data = sperm.comp1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -187.236  -55.028   -8.606   75.928  156.257
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -58.694    121.619  -0.483   0.6465
## m.vol         23.247      7.117   3.266   0.0171 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 120.8 on 6 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:   0.64,  Adjusted R-squared:   0.58
## F-statistic: 10.67 on 1 and 6 DF,  p-value: 0.01711
```

Confidence intervals: A 95% confidence interval for the mean increase in count per cm^3 3 increase in m.vol.

```
sc.c <- summary(sc1.mod1)$coefficients
sc.c
```

```
##              Estimate Std. Error    t value   Pr(>|t|)
## (Intercept) -58.69444 121.619433 -0.4826075 0.64647664
## m.vol        23.24653   7.117239  3.2662284 0.01711481
```

```
sc.c[2,1]+qt(c(.025,.975),6)*sc.c[2,2]
```

```
## [1]  5.831271 40.661784
```

- Prediction: The first line is the predictor variable values at which predictions are required. The second line calls predict with the fitted model object and new data from which to predict. se=TRUE tells the function to return standard errors along with the predictions.

```
df <- data.frame(m.vol=c(10,15,20,25))
predict(sc1.mod1,df,se=TRUE)
```

```
## $fit
##        1        2        3        4
## 173.7708 290.0035 406.2361 522.4687
##
## $se.fit
##        1        2        3        4
## 60.39178 43.29247 51.32314 76.98471
##
## $df
## [1] 6
##
## $residual.scale
## [1] 120.7836
```

- Co-linearity, confounding and causation:

Consider, a predictor variable x controls response variable y, but it is also highly correlated with a variable z, which plays no role at all in setting the level of y.

```
set.seed(1); n <- 100; x <- runif(n)
z <- x + rnorm(n)*.05
y <- 2 + 3 * x + rnorm(n)

summary(lm(y~z))
```

```
##
## Call:
## lm(formula = y ~ z)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.78186 -0.70862 -0.02848  0.69218  2.32858
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1628     0.2245   9.632 7.60e-16 ***
## z             2.7342     0.3836   7.127 1.75e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.053 on 98 degrees of freedom
## Multiple R-squared:  0.3414, Adjusted R-squared:  0.3347
## F-statistic: 50.79 on 1 and 98 DF,  p-value: 1.753e-10
```

```
summary(lm(y ~ x + z))
```

```
##
## Call:
## lm(formula = y ~ x + z)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
## x             1.3750     2.3368   0.588    0.558
## z             1.4193     2.2674   0.626    0.533
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 97 degrees of freedom
## Multiple R-squared:  0.3437, Adjusted R-squared:  0.3302
## F-statistic:  25.4 on 2 and 97 DF,  p-value: 1.345e-09
```

Despite z played no role in generating y, the correlation between x and z is high implying a strong evidence that z is predictive of y. Both z and y are being controlled by a confounding variable that is absent from the model.

The linear model fitting has 'shared out' the real dependence on x between x and z and has given x, the true predictor, a higher p-value than z, suggesting that if anything we should drop x! Slope parameter estimates have very high standard errors for not distinguishing which is actually driving y.

- Factor Variables:

```
(z <- c(1,1,1,2,2,1,3,3,3,3,4))
```

```
##  [1] 1 1 1 2 2 1 3 3 3 3 4
```

```
(z <- as.factor(z))
```

```
##  [1] 1 1 1 2 2 1 3 3 3 3 4
## Levels: 1 2 3 4
```

```
(x <- c("A","A","C","C","C","er","er"))
```

```
## [1] "A"  "A"  "C"  "C"  "C"  "er" "er"
```

```
(x <- factor(x))
```

```
## [1] A  A  C  C  C  er er
## Levels: A C er
```

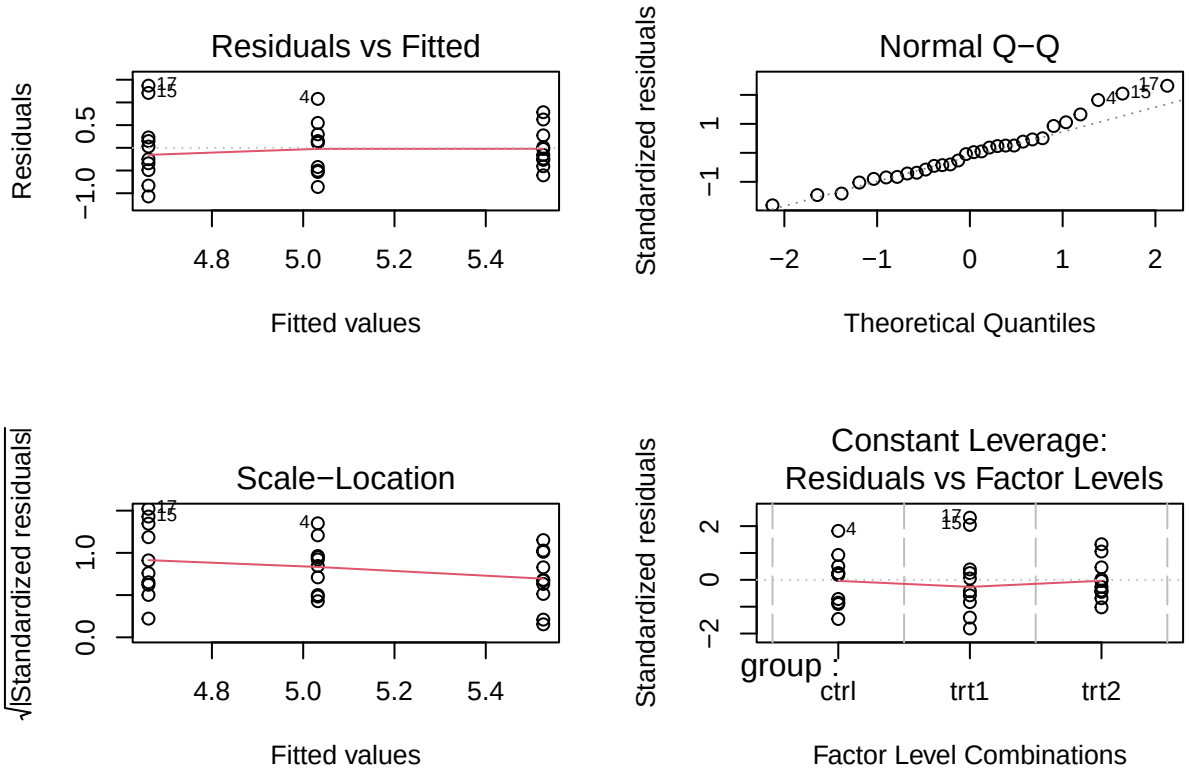
```
PlantGrowth$group # three levels
```

```
##  [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
```

```
PlantGrowth$group <- as.factor(PlantGrowth$group)
```

The response variable for these data is weight of the plants at some set time after planting, and the aim is to investigate whether the group factor controls this, and if so to what extent.

```
par(mfrow=c(2,2))
pgm.1 <- lm(weight ~ group, data=PlantGrowth)
plot(pgm.1)
```




```
summary(pgm.1)
```

```
##
## Call:
## lm(formula = weight ~ group, 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|)
## (Intercept)   5.0320     0.1971  25.527   <2e-16 ***
## grouptrt1    -0.3710     0.2788  -1.331   0.1944
## grouptrt2     0.4940     0.2788   1.772   0.0877 .
## ---
## 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:  0.2641, Adjusted R-squared:  0.2096
## F-statistic: 4.846 on 2 and 27 DF,  p-value: 0.01591
```

R reports an intercept parameter and parameters for the two treatment levels, but, in order to obtain an identifiable model, it has not included a parameter for the control level of the group factor. We would compare a model in which the expected response is given by a single parameter that does not depend on group.

```
pgm.0 <- lm(weight ~ 1, data=PlantGrowth)
anova(pgm.0,pgm.1)
```

```
## Analysis of Variance Table
##
```

```
## Model 1: weight ~ 1
## Model 2: weight ~ group
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     29 14.258
## 2     27 10.492  2    3.7663 4.8461 0.01591 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output gives the F-ratio statistic used to test the null hypothesis that the simpler model is correct.