

MATH 204: PRINCIPLES OF STATISTICS 2
WINTER 2026
MODEL SELECTION WORKED EXAMPLES

1. **Moore data:** Consider the following data set in R: from the R help file

The Moore data frame has 45 rows and 4 columns. The data are for subjects in a social-psychological experiment, who were faced with manipulated disagreements 40 involving partners of either of low or high status. The subject could either conform to the partner's judgment or stick with their own judgment.

```
> suppressPackageStartupMessages(library(car)) #Install library
> data(Moore) #See help(Moore)
> Moore$fcategory<-factor(Moore$fcategory,
+                           levels=c('high', 'medium', 'low'))
```

There are several variables:

- **partner.status** is a factor predictor with two levels (high, low);
- **fcategory** is a factor predictor with three levels (high, low, medium) measuring the degree of authoritarianism of the study subject,
- **fscore** is a continuous covariate, measuring authoritarianism on a continuous scale.
- **conformity** the response variable, measuring the number of times out of 40 the subject conformed to their partners opinion.

The objective was to see whether a subject's own personal degree of authoritarianism affected the degree of conformity. The number of observations in the cross-categories of the factors are

```
> table(Moore$partner.status, Moore$fcategory)

+
+           high medium low
+   high      7      11  5
+   low       8       4 10

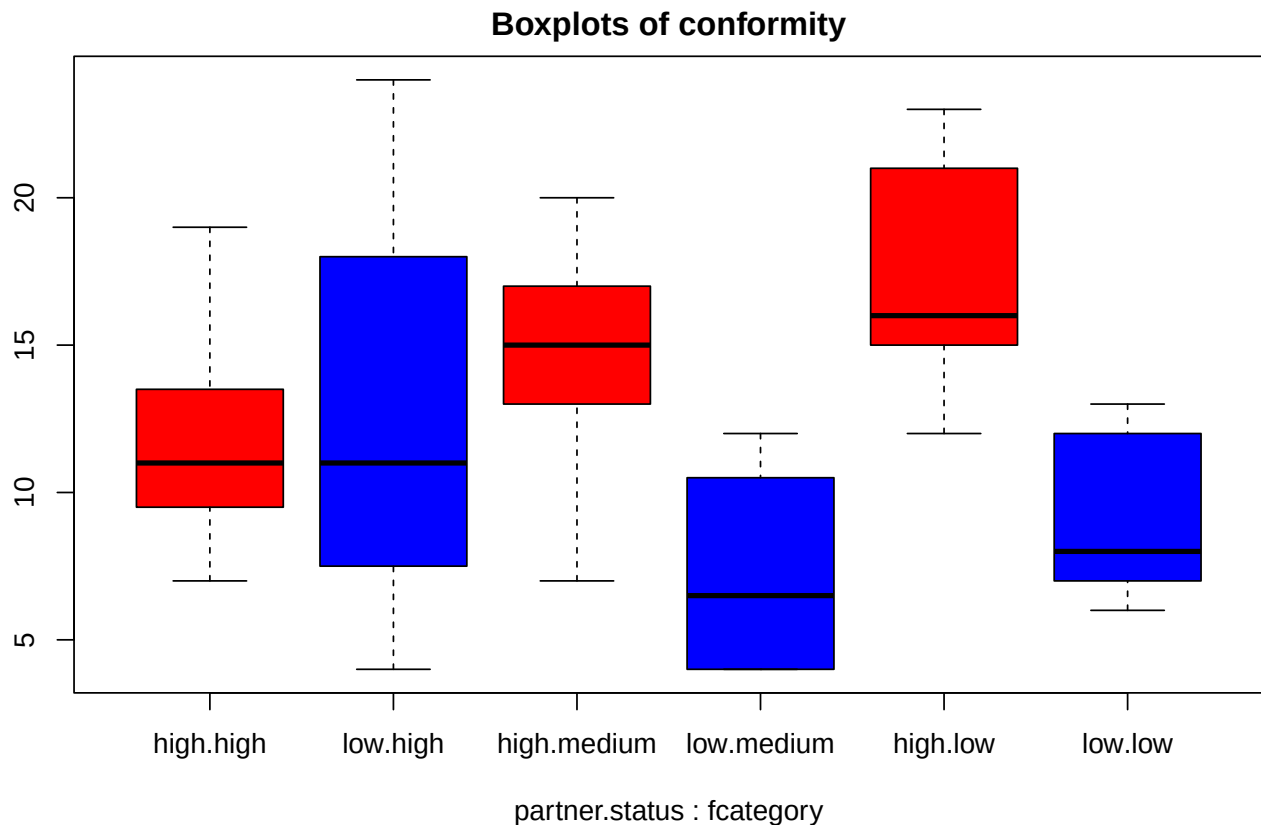
> table(Moore$conformity)

+
+  4  6  7  8  9 10 11 12 13 14 15 16 17 19 20 21 23 24
+  3  2  5  4  3  2  1  5  4  3  2  2  3  1  1  1  2  1
```

so we have an unbalanced design.

In the following boxplot, the red boxes correspond to the partner.status level high.

```
> par(mar=c(4, 2, 2, 0))
> boxplot(conformity ~ partner.status * fcategory, data=Moore,
+          ylab="conformity", main="Boxplots of conformity",
+          col=rep(c('red', 'blue'), 3))
```



There are two ways to analyze these data depending on how the authoritarianism score is treated (either as a factor or as continuous covariate). We begin with the factor predictor analysis.

Backward selection: We begin with the full factorial model that includes all main effects and all two-way interactions.

```
> M.copy<-Moore
> names(M.copy)<-c('status','conf','fcats','fscore')
> fit1<-lm(conf~status+fcats+status:fcats,data=M.copy)
> print(summary(aov(fit1)),digits=6)
```

		Df	Sum Sq	Mean Sq	F value	Pr(>F)	
:	status	1	204.332	204.3324	9.74482	0.0033806 **	
:	fcats	2	11.615	5.8074	0.27696	0.7595645	
:	status:fcats	2	175.489	87.7445	4.18462	0.0225724 *	
:	Residuals	39	817.764	20.9683			
:	---						
:	Signif. codes:	0	'***'	0.001	'**'	0.01	'*' 0.05
:						'.' 0.1	
:						' ' 1	

This analysis implies that the interaction term is significant. We try to drop one term using `drop1` to carry out the F-test: this function carries out 'single term deletions', and in this case the only term that can be legitimately is the interaction.

```

> drop1(fit1, test='F')

+ Single term deletions
+
+ Model:
+ conf ~ status + fcat + status:fcat
+           Df Sum of Sq   RSS   AIC F value Pr(>F)
+ <none>                817.8 142.5
+ status:fcat    2       175.5 993.3 147.2   4.185 0.0226 *
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

This result implies that the term cannot be dropped: the p-value is below the usual significance threshold of $\alpha = 0.05$. The F-statistic computed here matches that in the earlier ANOVA table. Another way to carry out the test is using the `anova` function:

```

> #Remove the 3-way interaction.
> fit1a<-update(fit1, ~.-status:fcat)
> anova(fit1, fit1a, test='F')

+ Analysis of Variance Table
+
+ Model 1: conf ~ status + fcat + status:fcat
+ Model 2: conf ~ status + fcat
+   Res.Df    RSS Df Sum of Sq    F Pr(>F)
+ 1       39 817.8
+ 2       41 993.3 -2     -175.5 4.185 0.0226 *
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Note that

- the terms in the RSS column are the *SSE* quantities
- the `Res.Df` column contains the error degrees of freedom. Note that, in the notation of the course, if k and g are the counts of the number of non-intercept terms in the model, we have

$$k - g = (n - g - 1) - (n - k - 1)$$

so knowing the error degrees of freedom is sufficient.

In the initial ANOVA table gives an indication that the `fcat` variable is not very significant so we attempt to omit it. However, to retain a legal model, we must drop all terms including that variable.

```

> fit2<-lm(conf~status, data=M.copy)
> anova(fit1, fit2, test='F')

+ Analysis of Variance Table
+

```

```

+ Model 1: conf ~ status + fcat + status:fcat
+ Model 2: conf ~ status
+   Res.Df    RSS Df Sum of Sq    F Pr(>F)
+ 1      39  817.8
+ 2      43 1004.9 -4    -187.1 2.231 0.0833 .
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

This simplification is supported. We are left with a model that excludes any reference to the authoritarianism category.

```

> summary(fit2)

+
+ Call:
+ lm(formula = conf ~ status, data = M.copy)
+
+ Residuals:
+   Min       1Q   Median       3Q      Max
+ -7.217 -2.955 -0.217  2.783 14.045
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)    14.22      1.01    14.10  <2e-16 ***
+ statuslow      -4.26      1.44    -2.96   0.005 **
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 4.83 on 43 degrees of freedom
+ Multiple R-squared:  0.169, Adjusted R-squared:  0.15
+ F-statistic: 8.74 on 1 and 43 DF,  p-value: 0.00503

```

Forward selection: We fit the two single variable models:

```

> fit1.only<-lm(conf~status,data=M.copy)
> print(summary(aov(fit1.only)), digits=6)

+              Df    Sum Sq Mean Sq F value    Pr(>F)
+ status         1   204.332  204.332  8.74373 0.0050293 **
+ Residuals     43  1004.868   23.369
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

For the second factor:

```

> fit2.only<-lm(conf~fcat,data=M.copy)
> print(summary(aov(fit2.only)), digits=6)

+              Df    Sum Sq Mean Sq F value    Pr(>F)
+ fcat          2     3.733   1.86667 0.06504 0.93713
+ Residuals    42  1205.467  28.70159

```

We conclude that **status** is the best variable to start with on the forward path. We now consider the “main effects” model

```
partner.status+fcategory
```

```
> fit1.add<-lm(conf~status+fcat,data=M.copy)
> anova(fit1.only, fit1.add, test = 'F')

+ Analysis of Variance Table
+
+ Model 1: conf ~ status
+ Model 2: conf ~ status + fcat
+   Res.Df    RSS Df Sum of Sq   F Pr(>F)
+ 1      43 1004.9
+ 2      41  993.3  2    11.62 0.24  0.788
```

It is clear that adding **fc** does not significantly improve the fit. The forward selection stops at this comparison.

```
> summary(fit1.only)

+
+ Call:
+ lm(formula = conf ~ status, data = M.copy)
+
+ Residuals:
+   Min       1Q   Median       3Q      Max
+ -7.217 -2.955 -0.217  2.783 14.045
+
+ Coefficients:
+               Estimate Std. Error t value Pr(>|t|)
+ (Intercept)    14.22      1.01    14.10  <2e-16 ***
+ statuslow      -4.26      1.44    -2.96   0.005 **
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 4.83 on 43 degrees of freedom
+ Multiple R-squared:  0.169, Adjusted R-squared:  0.15
+ F-statistic: 8.74 on 1 and 43 DF,  p-value: 0.00503
```

We can conclude from this analysis that the status of the partner does affect the conformity score, in that having a lower status partner **decreases** the conformity. We can display the fitted modelled means using the function `lsmeans`:

```
> library(lsmeans) #Make sure this package is loaded
> lsmeans(fit1.add, ~ status)

+   status lsmean    SE df lower.CL upper.CL
```

```

+   high      14.39 1.06 41      12.25      16.5
+   low       9.78 1.08 41       7.59      12.0
+
+ Results are averaged over the levels of: fcat
+ Confidence level used: 0.95

> Moore.fit<-M.copy;Moore.fit$fit<-fitted(fit1.add)
> aggregate(fit~status,data=Moore.fit,mean)      #Check

+   status      fit
+ 1   high 14.21739
+ 2   low  9.95455

```

A second analysis can be carried out treating authoritarianism as a continuous predictor (covariate).

```

> fac8<-lm(conf~fscore+status+fscore:status, data=M.copy)
> print(summary(aov(fac8)),digits=6)

+               Df   Sum Sq   Mean Sq F value    Pr(>F)
+ fscore          1    0.135    0.1345 0.00646 0.9363200
+ status          1 204.827 204.8268 9.84032 0.0031563 **
+ fscore:status    1 150.821 150.8213 7.24578 0.0102414 *
+ Residuals       41 853.417   20.8151
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

This indicates that the interaction is significant, which we can compute equivalently using the single term deletion:

```

> drop1(fac8,test='F')

+ Single term deletions
+
+ Model:
+ conf ~ fscore + status + fscore:status
+               Df Sum of Sq    RSS   AIC F value Pr(>F)
+ <none>                        853.4 140.4
+ fscore:status 1      150.8 1004.2 145.7   7.246 0.0102 *
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Backward selection would therefore stop at this model. For forward selection, we start with the status variable and try to add the other variable as a main effect.

```

> fac9<-lm(conf~status,data=M.copy)
> fac10<-lm(conf~status+fscore,data=M.copy)
> anova(fac10,fac9,test='F')

```

```

+ Analysis of Variance Table
+
+ Model 1: conf ~ status + fscore
+ Model 2: conf ~ status
+   Res.Df  RSS Df Sum of Sq    F Pr(>F)
+ 1      42 1004
+ 2      43 1005 -1    -0.6289 0.026  0.872

```

This indicates a non-significant result, so forward selection stops here. However, note that

```

> anova(fac8, fac9, test='F')

+ Analysis of Variance Table
+
+ Model 1: conf ~ fscore + status + fscore:status
+ Model 2: conf ~ status
+   Res.Df    RSS Df Sum of Sq    F Pr(>F)
+ 1      41  853.4
+ 2      43 1004.9 -2    -151.4 3.638 0.0351 *
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

therefore the addition of the main effect and interaction for **fscore** is supported. If we produce plot of the data, the origin of this result is revealed.

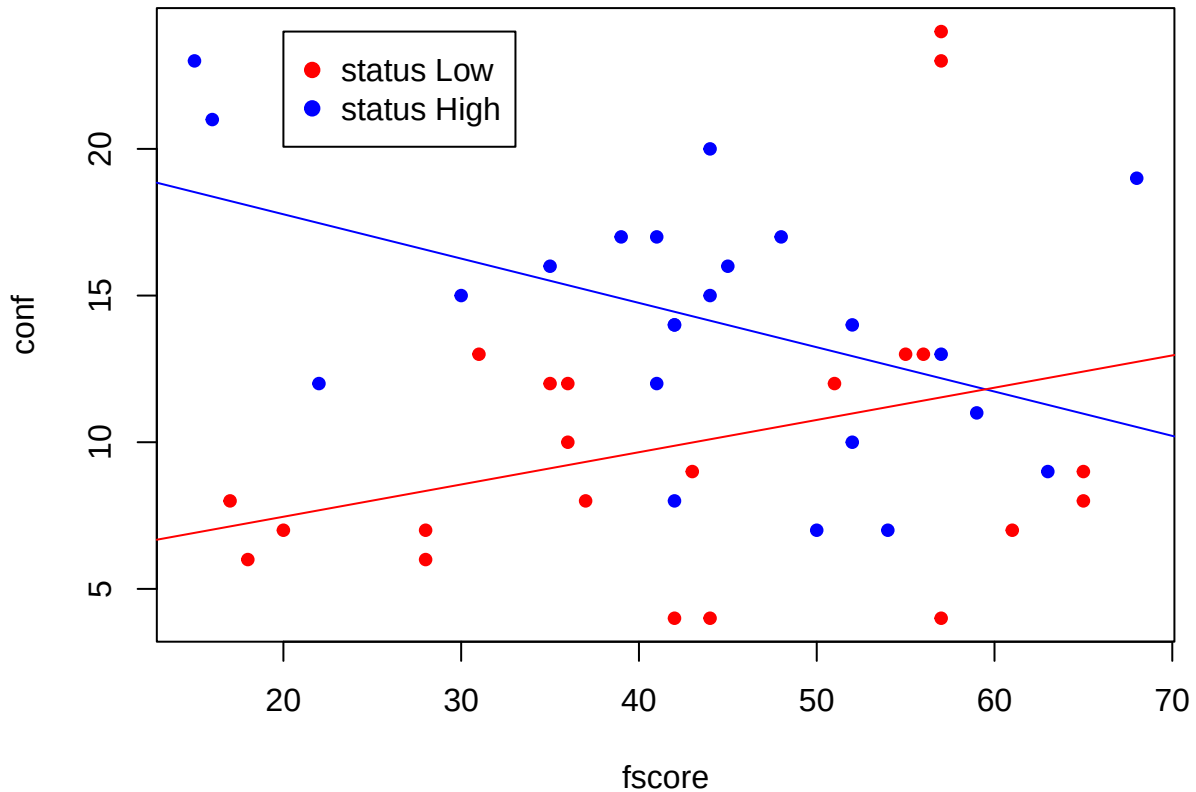
```

> coef(fac8)

+      (Intercept)           fscore      statuslow fscore:statuslow
+      20.793476      -0.151099      -15.534077           0.261102

> par(mar=c(4, 4, 2, 2))
> plot(M.copy$fscore, M.copy$conf, type='n', xlab='fscore', ylab='conf')
> points(M.copy$fscore[M.copy$status=='low'],
+        M.copy$conf[M.copy$status=='low'],
+        pch=19, cex=0.8, col='red')
> points(M.copy$fscore[M.copy$status=='high'],
+        M.copy$conf[M.copy$status=='high'],
+        pch=19, cex=0.8, col='blue')
> bevec<-coef(fac8)
> abline(bevec[1], bevec[2], col='blue')
> abline(bevec[1]+bevec[3], bevec[2]+bevec[4], col='red')
> legend(20, 24, c('status Low', 'status High'), col=c('red', 'blue'), pch=19)

```



This plot implies that the slopes and intercepts are different in the two subgroups identified by the two levels of the status variable. However, the slope in the low status group is not significantly different from zero. We can see that more easily by relabelling the baseline group, which is currently the high status group.

```
> M.copy$status<-relevel(M.copy$status, ref='low')
> summary(lm(conf~fscore+status+fscore:status,data=M.copy))$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.259398	2.9527103	1.78121	0.08228741
fscore	0.110003	0.0653181	1.68412	0.09975820
statushigh	15.534077	4.4004453	3.53011	0.00104156
fscore:statushigh	-0.261102	0.0969992	-2.69180	0.01024137

In summary this result says that

- in the high status group, increasing the **fscore** variable is associated with decreasing conformity;
- in the low status group, there is no significant relationship.

Thus we do need an interaction (as the effect of the continuous covariate is different in the two status groups).

2. **Cigarette brand data:** The data relate to a study of 25 cigarette brands: in the data set, for each brand,

- x_{i1} is the tar content (mg), denoted TAR;
- x_{i2} is the nicotine content (mg), denoted NICOTINE;
- x_{i3} is the weight (g), denoted WEIGHT;
- y_i is the amount of Carbon Monoxide (mg) produced in a standardized volume, denoted CO.

The data can be found at

<https://www.math.mcgill.ca/dstephens/204/Data/DataSets/cigs.csv>

```
> fdir<-'https://www.math.mcgill.ca/dstephens/204/Data/DataSets/'
> fname<-paste(fdir, 'cigs.csv', sep='')
> Cigs<-read.csv(fname, header=T)
> x1<-Cigs$TAR
> x2<-Cigs$NICOTINE
> x3<-Cigs$WEIGHT
> y<-Cigs$CO
> dti<-data.frame(x1, x2, x3, y)
> str(dti)

+ 'data.frame': 25 obs. of 4 variables:
+ $ x1: num 14.1 16 29.8 8 4.1 15 8.8 12.4 16.6 14.9 ...
+ $ x2: num 0.86 1.06 2.03 0.67 0.4 1.04 0.76 0.95 1.12 1.02 ...
+ $ x3: num 0.985 1.094 1.165 0.928 0.946 ...
+ $ y : num 13.6 16.6 23.5 10.2 5.4 15 9 12.3 16.3 15.4 ...

> head(dti)

+      x1    x2      x3    y
+ 1 14.1 0.86 0.9853 13.6
+ 2 16.0 1.06 1.0938 16.6
+ 3 29.8 2.03 1.1650 23.5
+ 4  8.0 0.67 0.9280 10.2
+ 5  4.1 0.40 0.9462  5.4
+ 6 15.0 1.04 0.8885 15.0
```

We begin this analysis from the main effects model

$$X_1 + X_2 + X_3$$

```
> fit1<-lm(y~x1+x2+x3)
> drop1(fit1, test='F')
```

```

+ Single term deletions
+
+ Model:
+ y ~ x1 + x2 + x3
+      Df Sum of Sq    RSS    AIC F value    Pr(>F)
+ <none>                43.89 22.07
+ x1      1      33.00 76.89 34.09  15.789 0.000692 ***
+ x2      1       0.95 44.84 20.61   0.455 0.507234
+ x3      1       0.00 43.89 20.07   0.001 0.973527
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

This analysis reveals that only X_1 seems to be influential in the model, as removing X_1 when X_2 and X_3 are retained leads to a rejection of the null hypothesis that the model is an adequate simplification. Inspecting the estimated coefficients supports this:

```

> summary(fit1)

+
+ Call:
+ lm(formula = y ~ x1 + x2 + x3)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -2.8926 -0.7827  0.0043  0.9289  2.4508
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)    3.202     3.462    0.93  0.36546
+ x1              0.963     0.242    3.97  0.00069 ***
+ x2             -2.632     3.901   -0.67  0.50723
+ x3             -0.130     3.885   -0.03  0.97353
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 1.45 on 21 degrees of freedom
+ Multiple R-squared:  0.919, Adjusted R-squared:  0.907
+ F-statistic: 79 on 3 and 21 DF, p-value: 1.33e-11

```

If we move to the full factorial model, we can attempt backward selection.

```

> fit2<-lm(y~x1*x2*x3)
> drop1(fit2,test='F')

+ Single term deletions
+
+ Model:
+ y ~ x1 * x2 * x3
+      Df Sum of Sq    RSS    AIC F value    Pr(>F)
+ <none>                25.17 16.17
+ x1:x2:x3  1      0.9209 26.09 15.07   0.622  0.441

```

```

> fit3<-update(fit2,~.-x1:x2:x3)
> drop1(fit3, test='F')

+ Single term deletions
+
+ Model:
+ y ~ x1 + x2 + x3 + x1:x2 + x1:x3 + x2:x3
+           Df Sum of Sq    RSS    AIC F value Pr(>F)
+ <none>                26.09 15.07
+ x1:x2      1      0.1635 26.25 13.22   0.113  0.741
+ x1:x3      1      0.9210 27.01 13.93   0.635  0.436
+ x2:x3      1      1.4197 27.51 14.39   0.980  0.335

> fit4<-update(fit3,~.-x2:x3)
> drop1(fit4, test='F')

+ Single term deletions
+
+ Model:
+ y ~ x1 + x2 + x3 + x1:x2 + x1:x3
+           Df Sum of Sq    RSS    AIC F value Pr(>F)
+ <none>                27.51 14.39
+ x1:x2      1      0.3765 27.89 12.73   0.260  0.616
+ x1:x3      1      0.4246 27.93 12.77   0.293  0.594

> fit5<-update(fit4,~.-x1:x3)
> drop1(fit5, test='F')

+ Single term deletions
+
+ Model:
+ y ~ x1 + x2 + x3 + x1:x2
+           Df Sum of Sq    RSS    AIC F value  Pr(>F)
+ <none>                27.93 12.77
+ x3          1      0.0006 27.94 10.78   0.005 0.94635
+ x1:x2       1     15.960 43.89 22.07  11.427 0.00297 **
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> fit6<-update(fit5,~.-x3)
> drop1(fit6, test='F')

+ Single term deletions
+
+ Model:
+ y ~ x1 + x2 + x1:x2
+           Df Sum of Sq    RSS    AIC F value  Pr(>F)
+ <none>                27.94 10.78
+ x1:x2       1     15.96 43.89 20.07   11.99 0.00233 **
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

In this series of fits, guided by the F-test comparisons, we have removed in turn the terms

$$X_1 : X_2 : X_3, X_2 : X_3, X_1 : X_3, X_3$$

and ended up with the model

$$X_1 + X_2 + X_1 : X_2$$

which cannot be simplified further.

```
> summary(fit6)

+
+ Call:
+ lm(formula = y ~ x1 + x2 + x1:x2)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -2.030 -0.746 -0.167  0.934  2.143
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)   0.2756     1.0654   0.26   0.7984
+ x1             1.0713     0.1958   5.47   2e-05 ***
+ x2             2.0617     3.3782   0.61   0.5482
+ x1:x2         -0.2098     0.0606  -3.46   0.0023 **
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 1.15 on 21 degrees of freedom
+ Multiple R-squared:  0.948, Adjusted R-squared:  0.941
+ F-statistic: 128 on 3 and 21 DF,  p-value: 1.17e-13
```

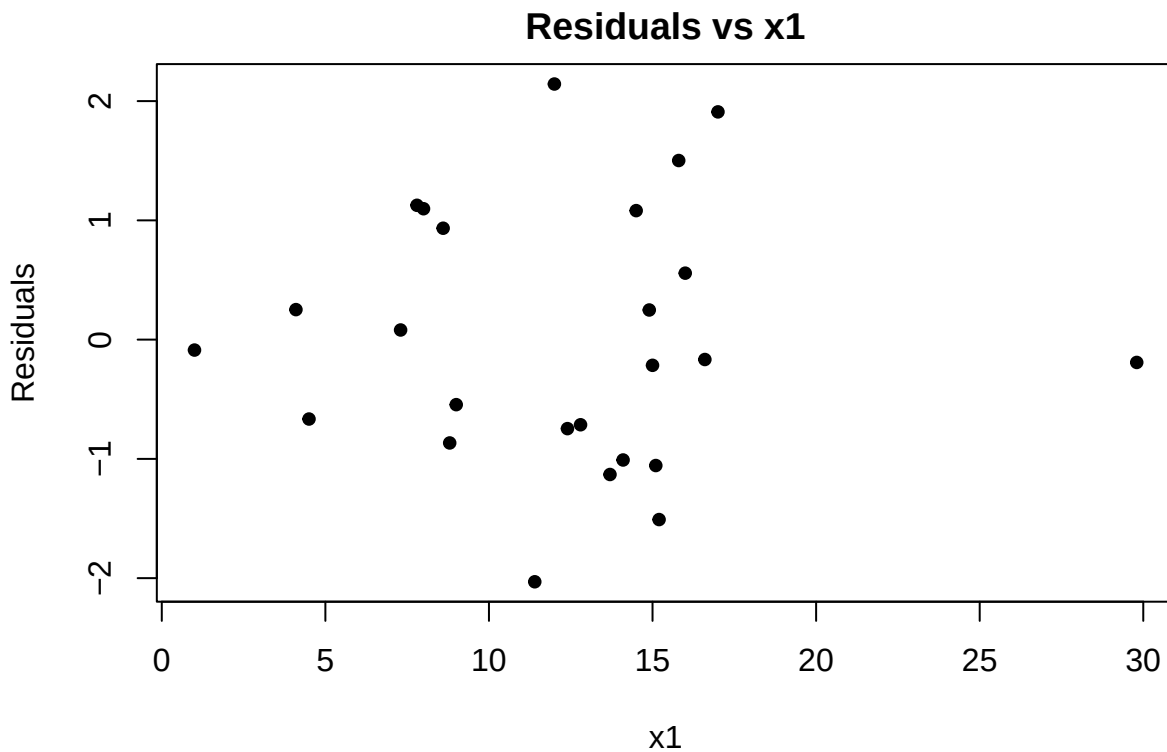
This model appears to have good explanatory power with a high R-squared statistic. The variable X_3 is omitted, even though when fitted **alone** in the model, it appears to be an influential variable.

```
> fit7<-lm(y~x3)
> print(summary(aov(fit7)), digits=5)

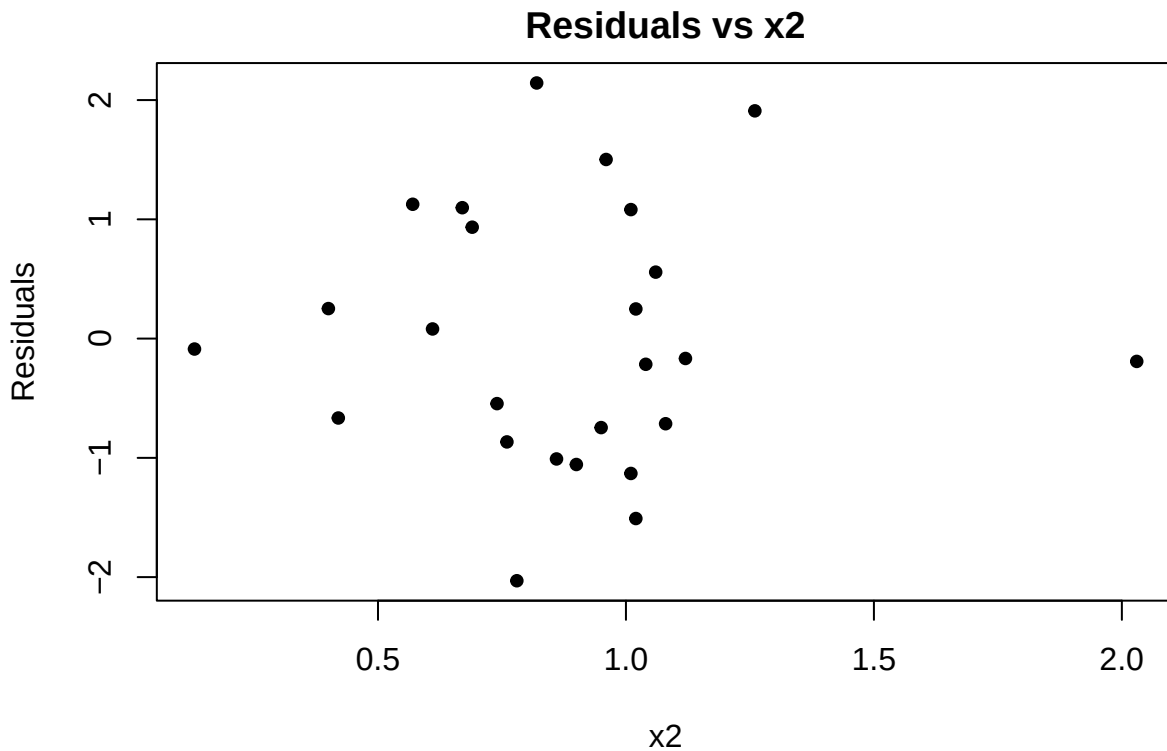
+              Df Sum Sq Mean Sq F value    Pr(>F)
+ x3              1 116.06 116.057    6.309 0.01948 *
+ Residuals     23 423.09  18.395
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To perform model checking, we inspect properties of the residuals: in this case the sample size is quite small, so we have limited capability to draw concrete conclusions.

```
> ehat<-residuals(fit6)
> par(mar=c(4,4,2,2))
> plot(x1,ehat,pch=19,cex=0.8,ylab='Residuals')
> title('Residuals vs x1')
```

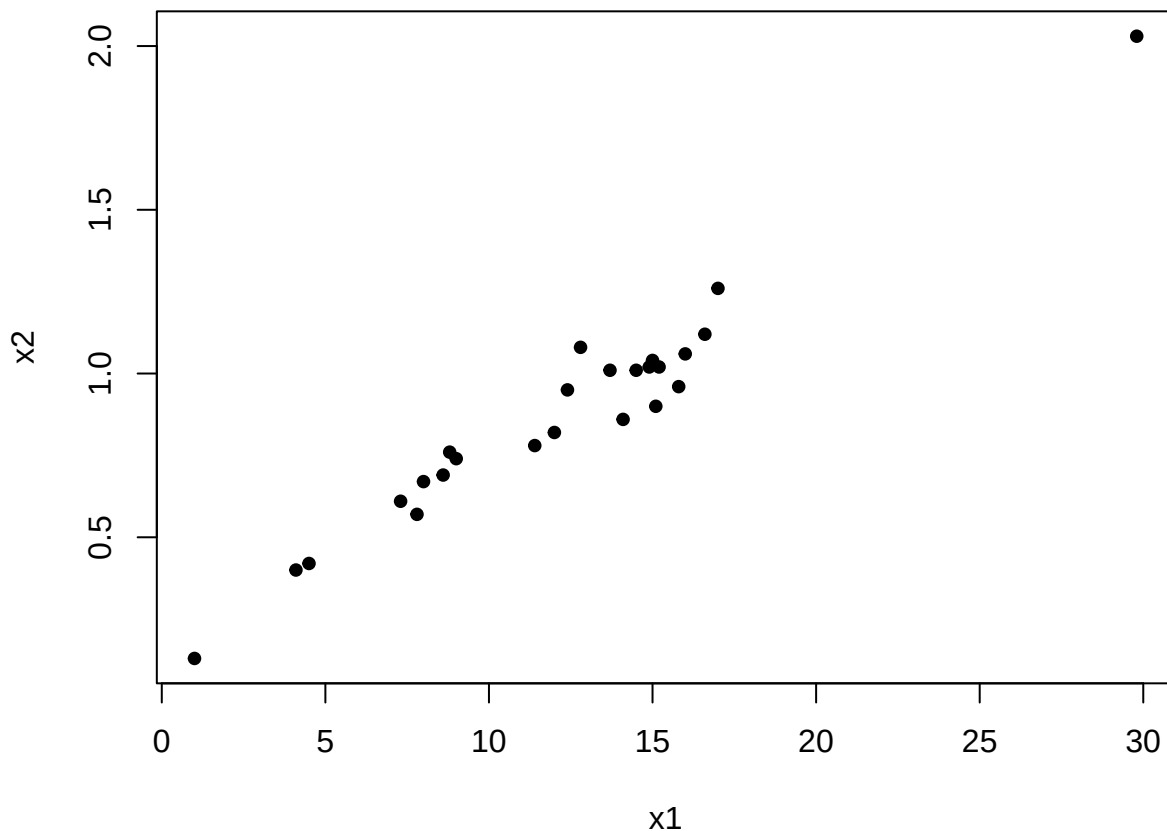


```
> plot(x2, ehat, pch=19, cex=0.8, ylab='Residuals')  
> title('Residuals vs x2')
```



The residuals seem to exhibit increasing variability as the two predictors increase, but always seem to be zero mean. The residual plots seem quite similar, indicating that perhaps the two predictors have a similar impact. We can investigate this further by inspecting a plot of x_1 vs x_2 .

```
> par(mar=c(4, 4, 1, 2))
> plot(x1, x2, pch=19, cex=0.8)
```



This is a clear example of **multicollinearity**; the two variables appear very highly correlated, and this can be confirmed by inspecting the correlation values.

```
> cor(cbind(x1, x2, x3, y))

+           x1           x2           x3           y
+ x1  1.000000  0.976608  0.490765  0.957485
+ x2  0.976608  1.000000  0.500183  0.925947
+ x3  0.490765  0.500183  1.000000  0.463959
+ y   0.957485  0.925947  0.463959  1.000000
```

This reveals that X_1 and X_2 have correlation 0.9766, which is very high, which implies that the two predictors contain very similar and overlapping explanatory power.

3. The data set `Filter.csv` contains data on the noise emission level of 36 cars. The cars are categorized using the `carsize` factor predictor that takes three levels, and two different noise filters are studied – the filter factor predictor `type` therefore takes two levels (normal filter and Octel filter)

```
> fname<-paste(fdir, 'Filter.csv', sep='')
> Filter<-read.csv(fname, header=TRUE)
> table(Filter$carsize, Filter$type)
```

```

+
+           normal filter Octel filter
+   large car           6           6
+   medium car          6           6
+   small car           6           6

```

Here we have a balanced two factor design, so the ANOVA table for a factorial design can be interpreted directly:

```

> print(summary(aov(lm(noise~carsize*type,data=Filter))),digits=6)

+           Df    Sum Sq  Mean Sq  F value      Pr(>F)
+ carsize      2 26051.39 13025.69 199.1189 < 2.22e-16 ***
+ type         1  1056.25  1056.25  16.1465 0.00036314 ***
+ carsize:type  2   804.17   402.08   6.1465 0.00579152 **
+ Residuals    30  1962.50    65.42
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

These results imply that both main effects and the interaction are important in the model. We verify this with model comparison tests. Computing the SSE quantities for the five models identified for a typical subgroup analysis

```

> mod0<-lm(noise~1,data=Filter)
> SSE0<-round(anova(mod0)[1,2]/1000,3)
> mod1<-lm(noise~carsize,data=Filter)
> SSE1<-round(anova(mod1)[2,2]/1000,3)
> mod2<-lm(noise~type,data=Filter)
> SSE2<-round(anova(mod2)[2,2]/1000,3)
> mod3<-lm(noise~carsize+type,data=Filter)
> SSE3<-round(anova(mod3)[3,2]/1000,3)
> mod4<-lm(noise~carsize*type,data=Filter)
> SSE4<-round(anova(mod4)[4,2]/1000,3)
> c(SSE0,SSE1,SSE2,SSE3,SSE4)

+ [1] 29.874  3.823 28.818  2.767  1.963

```

In summary, we have the following results:

Model	$SSE(\times 10^{-3})$	p
1	29.874	1
X_1	3.823	3
X_2	28.818	2
$X_1 + X_2$	2.767	4
$X_1 + X_2 + X_1 : X_2$	1.963	6

We can now proceed with any model comparison between nested models.

Suppose we want to compare the two models

Complete Model: $\text{carsize} + \text{type} + \text{carsize}:\text{type}$

Reduced Model: carsize

Using anova in R:

```
> fit1<-lm(noise~carsize,data=Filter)
> fit2<-lm(noise~carsize*type,data=Filter)
> anova(fit1,fit2,test='F')

+ Analysis of Variance Table
+
+ Model 1: noise ~ carsize
+ Model 2: noise ~ carsize * type
+   Res.Df  RSS Df Sum of Sq    F    Pr(>F)
+ 1      33 3823
+ 2      30 1963  3      1860 9.48 0.000146 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Computing from first principles, we have

```
> n<-nrow(Filter)
> SSER<-sum(residuals(fit1)^2)
> g<-length(coef(fit1))-1
> SSECR<-sum(residuals(fit2)^2)
> k<-length(coef(fit2))-1
> (fstat<-((SSER-SSEC)/(k-g))/(SSEC/(n-k-1)))

+ [1] 9.47983

> (pvalue<-1-pf(fstat,k-g,n-k-1))

+ [1] 0.000146097
```

This model simplification does not produce an adequate reduced model: the null hypothesis is rejected. To try backward selection:

```
> drop1(fit2,test='F')

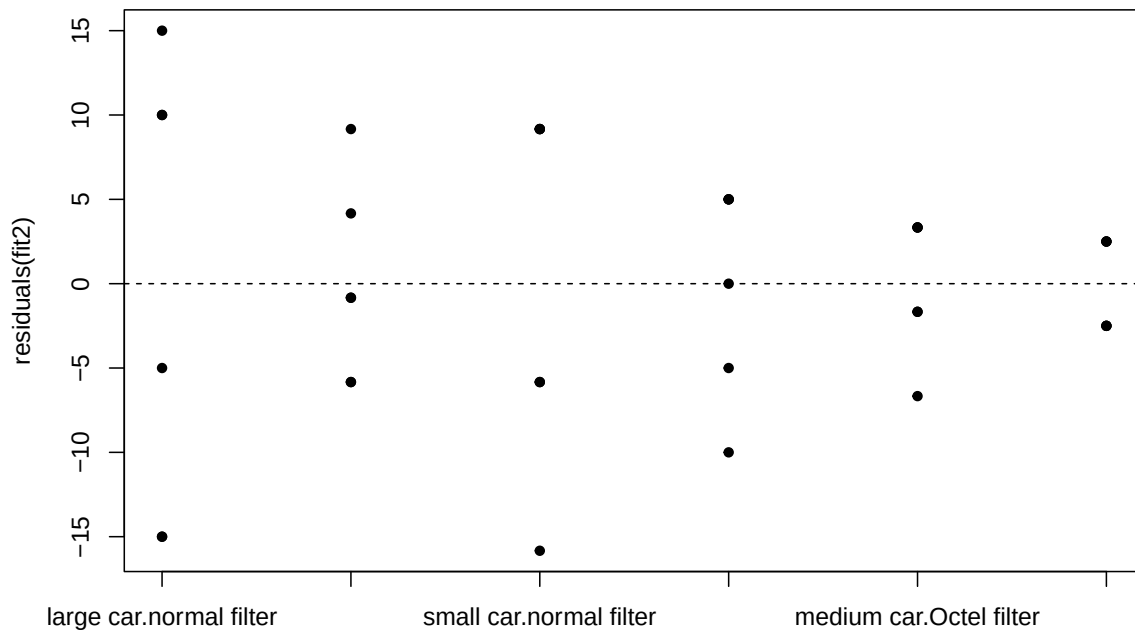
+ Single term deletions
+
+ Model:
+ noise ~ carsize * type
+           Df Sum of Sq  RSS   AIC F value    Pr(>F)
+ <none>             1962 155.9
+ carsize:type      2    804.2 2767 164.3   6.146 0.00579 **
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This is also strongly significant, so no model simplification is possible, we need the interaction term in the model. Note that the F-test here yields an F-statistic and p-value identical to the values in the factorial design ANOVA table. For a final model check

```
> lsmeans(fit2, ~ carsize+ type)

+ carsize      type      lsmean   SE df lower.CL upper.CL
+ large car    normal filter    775  3.3 30      768      782
+ medium car   normal filter    846  3.3 30      839      853
+ small car    normal filter    826  3.3 30      819      833
+ large car    Octel filter     770  3.3 30      763      777
+ medium car   Octel filter     822  3.3 30      815      828
+ small car    Octel filter     822  3.3 30      816      829
+
+ Confidence level used: 0.95

> par(mar=c(4, 4, 2, 4))
> stripchart(residuals(fit2)~ carsize+ type ,data=Filter,
+           pch=19,cex=0.8,vertical=TRUE)
> abline(h=0,lty=2)
```



This residual plot is quite unusual because the outcome variable `noise` has a lot of repeated (integer) values. However, the model assumptions seem to be adequate here: the data are not Normally distributed, but the Normal distribution is a reasonable approximation.

4. The data set `PatSat.csv` contains information on patient satisfaction for 25 patients having undergone treatment at a hospital for the same condition. There are four predictors: `Age` (age of patient in years), `Severity` (severity score for condition) and `Anxiety` (anxiety score for patient) are continuous predictors, whereas `Surgery` is a factor predictor with two levels (No and Yes) recording whether surgery was needed.

The research question here is whether having surgery (as opposed to not having surgery) significantly affected patient satisfaction. A simple comparison of responses for the two surgery groups may not be sufficient to answer the research question if age, severity or anxiety also influence the outcome.

```
> fname<-paste(fdir, 'PatSat.csv', sep='')
>
> PatSat<-read.csv(fname, header=TRUE)
> PatSat$Surgery<-as.factor(PatSat$Surgery)
> str(PatSat)

+ 'data.frame': 25 obs. of 5 variables:
+ $ Satisfaction: int 68 77 96 80 43 44 26 88 75 57 ...
+ $ Age          : int 55 46 30 35 59 61 74 38 27 51 ...
+ $ Severity     : int 50 24 46 48 58 60 65 42 42 50 ...
+ $ Surgery      : Factor w/ 2 levels "No", "Yes": 1 2 2 2 1 1 2 2 1 2 ...
+ $ Anxiety      : num 2.1 2.8 3.3 4.5 2 5.1 5.5 3.2 3.1 2.4 ...
```

In a direct comparison between surgery groups, it seems that there is no effect of surgery on satisfaction level

```
> fit0<-lm(Satisfaction~Surgery, data=PatSat)
> print(summary(fit0), concise=TRUE)

+
+ Call:
+ lm(formula = Satisfaction ~ Surgery, data = PatSat)
+
+ Residuals:
+    Min       1Q   Median       3Q      Max
+ -37.4  -15.0    4.0   17.0   32.6
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)    71.00      6.43   11.04 1.2e-10 ***
+ SurgeryYes     -7.64      8.60   -0.89  0.38
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 21.3 on 23 degrees of freedom
+ Multiple R-squared:  0.0332, Adjusted R-squared:  -0.00881
+ F-statistic: 0.79 on 1 and 23 DF,  p-value: 0.383
```

The task now is to assess whether any effect is being masked by the other variables. We start with the main effects model:

```

> fit1<-lm(Satisfaction~Age+Severity+Surgery+Anxiety,data=PatSat)
> print(summary(fit1), concise=TRUE)

+
+ Call:
+ lm(formula = Satisfaction ~ Age + Severity + Surgery + Anxiety,
+     data = PatSat)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -18.51   -5.10    1.31    4.74   28.72
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)  140.169      8.319   16.85  2.8e-13 ***
+ Age          -1.143      0.190   -6.00  7.2e-06 ***
+ Severity     -0.470      0.187   -2.52   0.02 *
+ SurgeryYes    2.226      4.140    0.54   0.60
+ Anxiety       1.267      1.492    0.85   0.41
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 9.92 on 20 degrees of freedom
+ Multiple R-squared:  0.818, Adjusted R-squared:  0.782
+ F-statistic: 22.5 on 4 and 20 DF,  p-value: 3.61e-07

> drop1(fit1,test='F')

+ Single term deletions

+
+ Model:
+ Satisfaction ~ Age + Severity + Surgery + Anxiety
+      Df Sum of Sq  RSS   AIC F value    Pr(>F)
+ <none>                 1969 119.2
+ Age      1       3545 5514 142.9  36.018 7.22e-06 ***
+ Severity 1        624 2593 124.0   6.341 0.0204 *
+ Surgery  1         28 1997 117.5   0.289 0.5968
+ Anxiety  1         71 2040 118.0   0.721 0.4058
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

It seems we can drop the predictors Surgery and Anxiety: we update the model as follows by omitting these variables, and computing the summaries and comparison statistics.

```

> fit2<-update(fit1,~.-Surgery-Anxiety)
> print(summary(fit2), concise=TRUE)

+
+ Call:
+ lm(formula = Satisfaction ~ Age + Severity, data = PatSat)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -18.369  -5.953   0.297   4.046  29.344
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)   139.923     8.100   17.27 2.8e-14 ***
+ Age           -1.046     0.157   -6.65 1.1e-06 ***
+ Severity      -0.436     0.179   -2.44  0.023 *
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 9.68 on 22 degrees of freedom
+ Multiple R-squared:  0.81, Adjusted R-squared:  0.792
+ F-statistic: 46.8 on 2 and 22 DF,  p-value: 1.19e-08

> anova(fit2,fit1,test='F')

+ Analysis of Variance Table
+
+ Model 1: Satisfaction ~ Age + Severity
+ Model 2: Satisfaction ~ Age + Severity + Surgery + Anxiety
+   Res.Df  RSS Df Sum of Sq    F Pr(>F)
+ 1      22 2062
+ 2      20 1968   2    93.75 0.476  0.628

```

It seems that the two additional predictors do not add anything to the model. We now check whether the introduction of interactions has any effect: we begin by adding in all the interactions to produce a complete model, and then also add only the two-way interactions.

```

> fit3<-lm(Satisfaction~Age*Surgery*Severity*Anxiety, data=PatSat)
> print(summary(fit3)$r.squared) #R-squared value

+ [1] 0.894122

> anova(fit2,fit3,test='F')

+ Analysis of Variance Table
+
+ Model 1: Satisfaction ~ Age + Severity
+ Model 2: Satisfaction ~ Age * Surgery * Severity * Anxiety

```

```

+   Res.Df  RSS Df Sum of Sq    F Pr(>F)
+ 1      22 2062
+ 2       9 1147 13      915.5 0.553  0.84

> fit4<-lm(Satisfaction~(Age+Surgery+Severity+Anxiety)^2, data=PatSat)
> print(summary(fit4)$r.squared)  #R-squared value

+ [1] 0.859657

> anova(fit2,fit4,test='F')

+ Analysis of Variance Table
+
+ Model 1: Satisfaction ~ Age + Severity
+ Model 2: Satisfaction ~ (Age + Surgery + Severity + Anxiety)^2
+   Res.Df  RSS Df Sum of Sq    F Pr(>F)
+ 1      22 2062
+ 2      14 1520  8      542.2 0.624  0.745

```

These models do not improve the fit, it seems, so we attempt other models: first the interaction model with the two predictors

```

> fit5<-update(fit2,~.+Age:Severity)
> anova(fit2,fit5,test='F')

+ Analysis of Variance Table
+
+ Model 1: Satisfaction ~ Age + Severity
+ Model 2: Satisfaction ~ Age + Severity + Age:Severity
+   Res.Df  RSS Df Sum of Sq    F Pr(>F)
+ 1      22 2062
+ 2      21 2033  1      29.55 0.305  0.586

```

This does not lead to a significant improvement. The model Age + Severity still seems to be preferred.

We now attempt a backward selection: we start with the model with up to four-way interaction. For brevity, only the key comparison details are printed.

```

> fit6<-lm(Satisfaction~Age*Severity*Surgery*Anxiety,data=PatSat)
> print(summary(fit6)$r.squared)  #R-squared value

+ [1] 0.894122

```

The R-squared value from the full factorial model has not increased very much compared to the fit of the main effects only model. We proceed with the single term deletions given by the drop function.

```
> dfit6<-drop1(fit6,test='F')
> row.names(dfit6)[-1]

+ [1] "Age:Severity:Surgery:Anxiety"

> as.matrix(dfit6)[-1,-4]

+           Df      Sum of Sq      RSS      F value      Pr(>F)
+ 1.000000    62.846744 1209.612401    0.493231    0.500248
```

```
> fit7<-update(fit6,~.-Age:Severity:Surgery:Anxiety)
> dfit7<-drop1(fit7,test='F')
> as.matrix(dfit7)[-1,-4]

+           Df      Sum of Sq      RSS      F value      Pr(>F)
+ Age:Severity:Surgery      1      4.70687 1214.32 0.0389122 0.847574
+ Age:Severity:Anxiety      1 112.28816 1321.90 0.9282987 0.358025
+ Age:Surgery:Anxiety      1 155.38995 1365.00 1.2846260 0.283482
+ Severity:Surgery:Anxiety  1  49.02255 1258.63 0.4052748 0.538676
```

```
> fit8<-update(fit7,~.-Age:Severity:Surgery)
> as.matrix(drop1(fit8,test='F'))[-1,-4]

+           Df      Sum of Sq      RSS      F value      Pr(>F)
+ Age:Severity:Anxiety      1 117.4479 1331.77 1.063911 0.324471
+ Age:Surgery:Anxiety      1 160.2717 1374.59 1.451833 0.253517
+ Severity:Surgery:Anxiety  1  73.1796 1287.50 0.662903 0.432809
```

```
> fit9<-update(fit8,~.-Severity:Surgery:Anxiety)
> as.matrix(drop1(fit9,test='F'))[-1,-4]

+           Df      Sum of Sq      RSS      F value      Pr(>F)
+ Severity:Surgery      1  309.726 1597.22 2.88677 0.115063
+ Age:Severity:Anxiety  1  159.709 1447.21 1.48855 0.245877
+ Age:Surgery:Anxiety   1  188.385 1475.88 1.75582 0.209824
```

```
> fit10<-update(fit9,~.-Age:Severity:Anxiety)
> as.matrix(drop1(fit10,test='F'))[-1,-4]

+           Df      Sum of Sq      RSS      F value      Pr(>F)
+ Age:Severity      1  41.5062 1488.71 0.372843 0.551985
+ Severity:Surgery  1 277.7283 1724.94 2.494782 0.138239
+ Severity:Anxiety  1  21.9808 1469.19 0.197449 0.664095
+ Age:Surgery:Anxiety 1  72.8474 1520.06 0.654375 0.433109
```

```
> fit11<-update(fit10,~.-Severity:Anxiety)
> as.matrix(drop1(fit11,test='F'))[-1,-4]
```

	Df	Sum of Sq	RSS	F value	Pr(>F)
+ Age:Severity	1	19.5266	1488.72	0.186070	0.672776
+ Severity:Surgery	1	255.8602	1725.05	2.438110	0.140734
+ Age:Surgery:Anxiety	1	55.8698	1525.06	0.532387	0.477636

```
> fit12<-update(fit11,~.-Age:Severity)
> as.matrix(drop1(fit12,test='F'))[-1,-4]
```

	Df	Sum of Sq	RSS	F value	Pr(>F)
+ Severity:Surgery	1	238.7830	1727.50	2.405930	0.141716
+ Age:Surgery:Anxiety	1	45.9943	1534.71	0.463429	0.506399

```
> fit13<-update(fit12,~.-Age:Surgery:Anxiety)
> as.matrix(drop1(fit13,test='F'))[-1,-4]
```

	Df	Sum of Sq	RSS	F value	Pr(>F)
+ Age:Surgery	1	259.87075	1794.58	2.7092632	0.119265
+ Severity:Surgery	1	204.85191	1739.56	2.1356683	0.163269
+ Age:Anxiety	1	3.27691	1537.99	0.0341632	0.855683
+ Surgery:Anxiety	1	119.67522	1654.38	1.2476650	0.280492

```
> fit14<-update(fit13,~.-Age:Anxiety)
> as.matrix(drop1(fit14,test='F'))[-1,]
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
+ Age:Surgery	1	304.921	1842.91	121.506	3.37041	0.0839311
+ Severity:Surgery	1	213.872	1751.86	120.239	2.36402	0.1425659
+ Surgery:Anxiety	1	125.506	1663.49	118.945	1.38727	0.2550935

```
> fit15<-update(fit14,~.-Surgery:Anxiety)
> as.matrix(drop1(fit15,test='F'))[-1,]
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
+ Anxiety	1	30.8179	1694.31	117.404	0.333468	0.5707773
+ Age:Surgery	1	185.8038	1849.30	119.592	2.010510	0.1732929
+ Severity:Surgery	1	279.4193	1942.91	120.827	3.023488	0.0991409

```
> fit16<-update(fit15,~.-Anxiety)
> as.matrix(drop1(fit16,test='F'))[-1,]
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
+ Age:Surgery	1	209.365	1903.68	118.317	2.34782	0.1419399
+ Severity:Surgery	1	314.769	2009.08	119.664	3.52982	0.0757052

```
> fit17<-update(fit16,~.-Age:Surgery)
> as.matrix(drop1(fit17,test='F'))[-1,]
```

		Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
+							
+	Age	1	3790.432	5694.11	143.708	39.8223	3.68631e-06
+	Severity:Surgery	1	135.847	2039.52	118.040	1.4272	2.46197e-01

```
> fit18<-update(fit17,~.-Severity:Surgery)
> as.matrix(drop1(fit18,test='F'))[-1,]
```

		Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
+							
+	Age	1	4121.8348	6161.36	143.679	42.440595	1.87274e-06
+	Severity	1	568.6604	2608.18	122.188	5.855229	2.46865e-02
+	Surgery	1	22.7641	2062.29	116.317	0.234391	6.33298e-01

```
> fit19<-update(fit18,~.-Surgery)
> as.matrix(drop1(fit19,test='F'))[-1,]
```

		Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
+							
+	Age	1	4148.272	6210.56	141.878	44.25283	0.000001092
+	Severity	1	557.449	2619.74	120.299	5.94674	0.023275606

The backward selection ends here as no further simplifications can be made.

```
> print(summary(aov(fit19)),digits=5)
```

		Df	Sum Sq	Mean Sq	F value	Pr(>F)	
+							
+	Age	1	8211.3	8211.3	87.5963	3.971e-09 ***	
+	Severity	1	557.4	557.4	5.9467	0.02328 *	
+	Residuals	22	2062.3	93.7			
+	---						
+	Signif. codes:	0	'***'	0.001	'**'	0.01	'*' 0.05 '.' 0.1 ' ' 1

This is the model also selected by the initial investigation. We examine the fit

```
> summary(fit2)$coef
```

		Estimate	Std. Error	t value	Pr(> t)
+					
+	(Intercept)	139.923348	8.100194	17.27407	2.78000e-14
+	Age	-1.046154	0.157263	-6.65228	1.09200e-06
+	Severity	-0.435907	0.178754	-2.43859	2.32756e-02

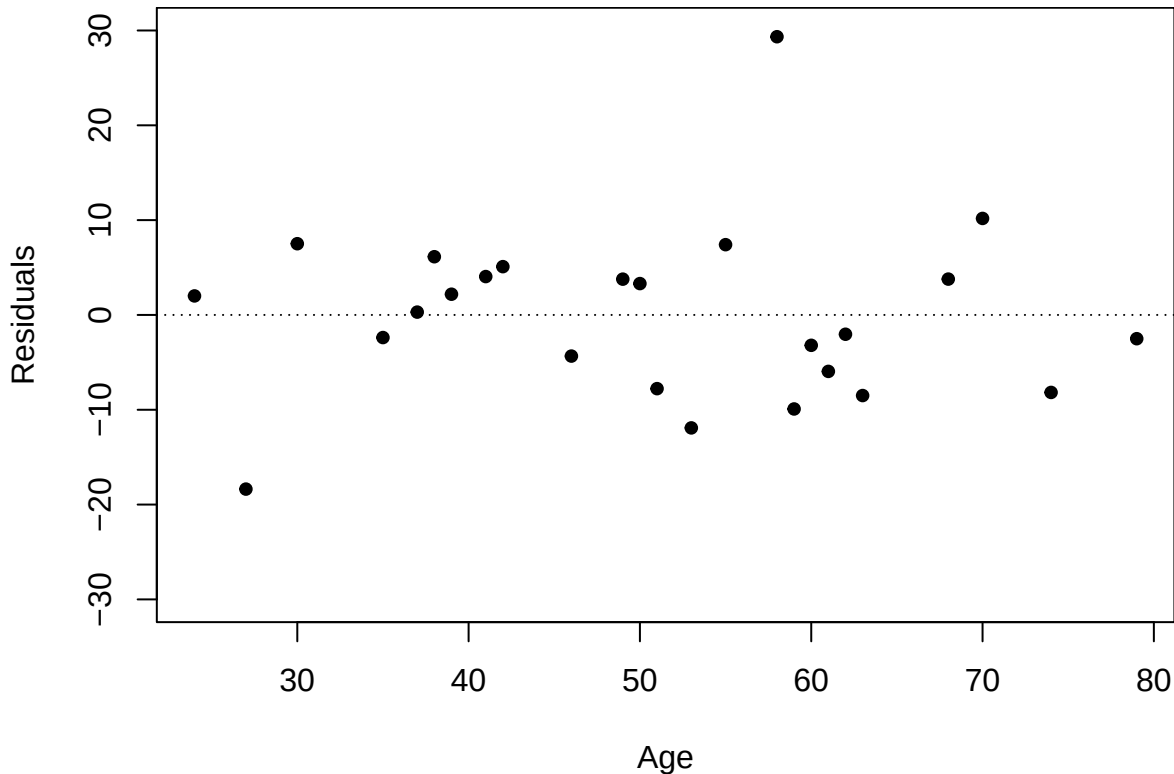
```
> summary(fit2)$r.squared
```

+	[1]	0.809595
---	-----	----------

The negative coefficients for Age and Severity indicate that satisfaction decreases with increasing age and severity, but surgery is not influential in determining satisfaction score. Also, a relatively high R-squared quantity demonstrates good explanatory power.

For the residual plots, the plots seem adequate, although there might be one outlier with a large positive value.

```
> par(mar=c(4, 4, 0, 2))
> plot(PatSat$Age, residuals(fit2), pch=19, cex=0.8, ylim=range(-30, 30),
+      xlab='Age', ylab='Residuals'); abline(h=0, lty=3)
```



```
> plot(PatSat$Severity, residuals(fit2), pch=19, cex=0.8, ylim=range(-30, 30),
+      xlab='Severity', ylab='Residuals'); abline(h=0, lty=3)
```

