FACTOR PREDICTORS

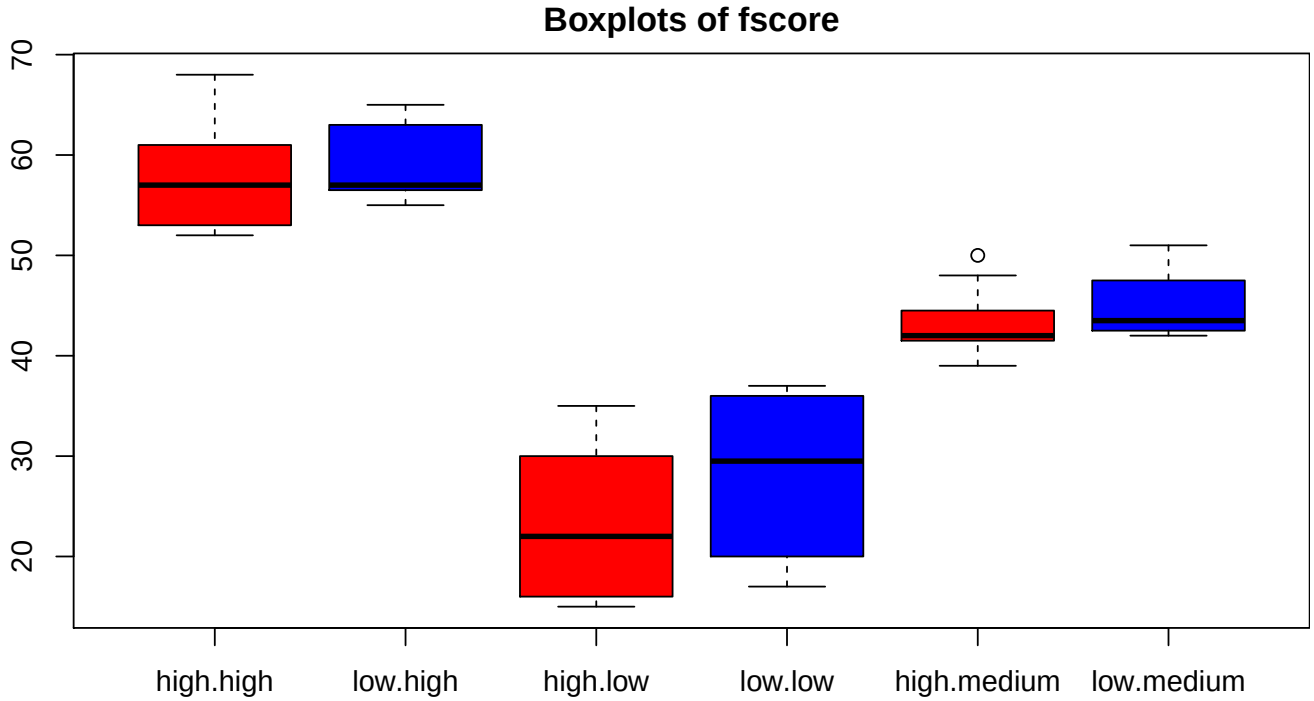
Consider the following data set in R: from the R help file

> *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 disagreement from a partner of either of low or high status. The subjects could either conform to the partner's judgment or stick with their own judgment.*

```
library(car)  #Need to install this library first
data(Moore)   #See help(Moore)
str(Moore)

: 'data.frame': 45 obs. of  4 variables:
:  $ partner.status: Factor w/ 2 levels "high","low": 2 2 2 2 2 2 2 2 2 2 ...
:  $ conformity    : int  8 4 8 7 10 6 12 4 13 12 ...
:  $ fcategory     : Factor w/ 3 levels "high","low","medium": 2 1 1 2 2 2 3 3 2 2 ...
:  $ fscore        : int  37 57 65 20 36 18 51 44 31 36 ...
```

There are two factors: `partner.status` with two levels (`high`, `low`) and `fcategory` with three levels (`high`, `low`, `medium`). There is also a continuous covariate `conformity`. The response variable is `fscore`. The number of observations in the cross-categories are

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

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

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



The red boxes correspond to the partner.status level `high`.

First we fit the two single factor models; first the model which may be written

`1+partner.status`

that uses partner status only.

```
fit1.only<-lm(fscore~partner.status,data=Moore);summary(fit1.only)

:
: Call:
: lm(formula = fscore ~ partner.status, data = Moore)
:
: Residuals:
:      Min       1Q   Median       3Q      Max
: -28.5217  -7.6818   0.3182  12.3182  24.4783
:
: Coefficients:
:                   Estimate Std. Error t value Pr(>|t|)
: (Intercept)        43.5217     3.0043  14.487   <2e-16 ***
: partner.statuslow  -0.8399     4.2967  -0.195    0.846
: ---
: Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
:
: Residual standard error: 14.41 on 43 degrees of freedom
: Multiple R-squared:  0.0008879,Adjusted R-squared:  -0.02235
: F-statistic: 0.03821 on 1 and 43 DF,  p-value: 0.8459
```

This result implies that partner status has no influence on outcome. The two levels of partner.status give two parameter estimates: the baseline group is factor level `high` (`R` chooses the baseline by alphabetical ordering), and the estimated contrast between `high` and `low` is

$$\beta^{C}_{\text{low}} = -0.84$$

For the second factor:

```
fit2.only<-lm(fscore~fcategory,data=Moore);summary(fit2.only)

:
: Call:
: lm(formula = fscore ~ fcategory, data = Moore)
:
: Residuals:
:      Min       1Q   Median       3Q      Max
: -11.9333  -2.8667  -0.8667   4.1333  10.0667
:
: Coefficients:
:                 Estimate Std. Error t value Pr(>|t|)
: (Intercept)       58.533      1.508  38.819  < 2e-16 ***
: fcategorylow     -31.600      2.132 -14.819  < 2e-16 ***
: fcategorymedium  -14.667      2.132  -6.878 2.17e-08 ***
: ---
: Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
:
: Residual standard error: 5.84 on 42 degrees of freedom
: Multiple R-squared:  0.8397,Adjusted R-squared:  0.832
: F-statistic:   110 on 2 and 42 DF,  p-value: < 2.2e-16
```

This result implies that `fcategory` does have an influence on outcome. The two levels of partner.status give three parameter estimates: the baseline group is factor level `high`, and the estimated contrasts between `high` and `low`, and `high` and `medium` are

$$\beta^{C}_{\text{low}} = -31.600 \qquad \beta^{C}_{\text{medium}} = -14.667$$

To use different baseline group, say `low`, use the following commands, in particular, the function `relevel()`:

```
Moore2 <- within(Moore,  fcategory<- relevel(fcategory, ref = 'low'))
fit2.only2<-lm(fscore~fcategory,data=Moore2);summary(fit2.only2)

:
: Call:
: lm(formula = fscore ~ fcategory, data = Moore2)
:
: Residuals:
:      Min       1Q   Median       3Q      Max
: -11.9333  -2.8667  -0.8667   4.1333  10.0667
:
: Coefficients:
:                 Estimate Std. Error t value Pr(>|t|)
: (Intercept)       26.933      1.508  17.862  < 2e-16 ***
: fcategoryhigh     31.600      2.132  14.819  < 2e-16 ***
: fcategorymedium   16.933      2.132   7.941 6.77e-10 ***
: ---
: Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
:
: Residual standard error: 5.84 on 42 degrees of freedom
: Multiple R-squared:  0.8397,Adjusted R-squared:  0.832
: F-statistic:   110 on 2 and 42 DF,  p-value: < 2.2e-16
```

Notice that many of the details of the fit do not change.

We now consider the "main effects only" model

`1+partner.status+fcategory`

```
fit3.add<-lm(fscore~partner.status+fcategory,data=Moore);summary(fit3.add)

:
: Call:
: lm(formula = fscore ~ partner.status + fcategory, data = Moore)
:
: Residuals:
:     Min      1Q  Median      3Q     Max
: -10.798  -3.204  -1.175   4.825  10.850
:
: Coefficients:
:                   Estimate Std. Error t value Pr(>|t|)
: (Intercept)         57.150      1.780  32.115  < 2e-16 ***
: partner.statuslow    2.593      1.825   1.421    0.163
: fcategorylow       -31.946      2.121 -15.062  < 2e-16 ***
: fcategorymedium    -13.975      2.162  -6.463 9.51e-08 ***
: ---
: Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
:
: Residual standard error: 5.77 on 41 degrees of freedom
: Multiple R-squared:  0.8472,Adjusted R-squared:  0.836
: F-statistic: 75.78 on 3 and 41 DF,  p-value: < 2.2e-16
```

We can display the fitted modelled means using the function `lsmeans`:

```
library(lsmeans)   #Make sure this package is loaded

:  Loading required package:  estimability
:  Loading required package:  methods

lsmeans(fit3.add, ~ partner.status * fcategory)
```

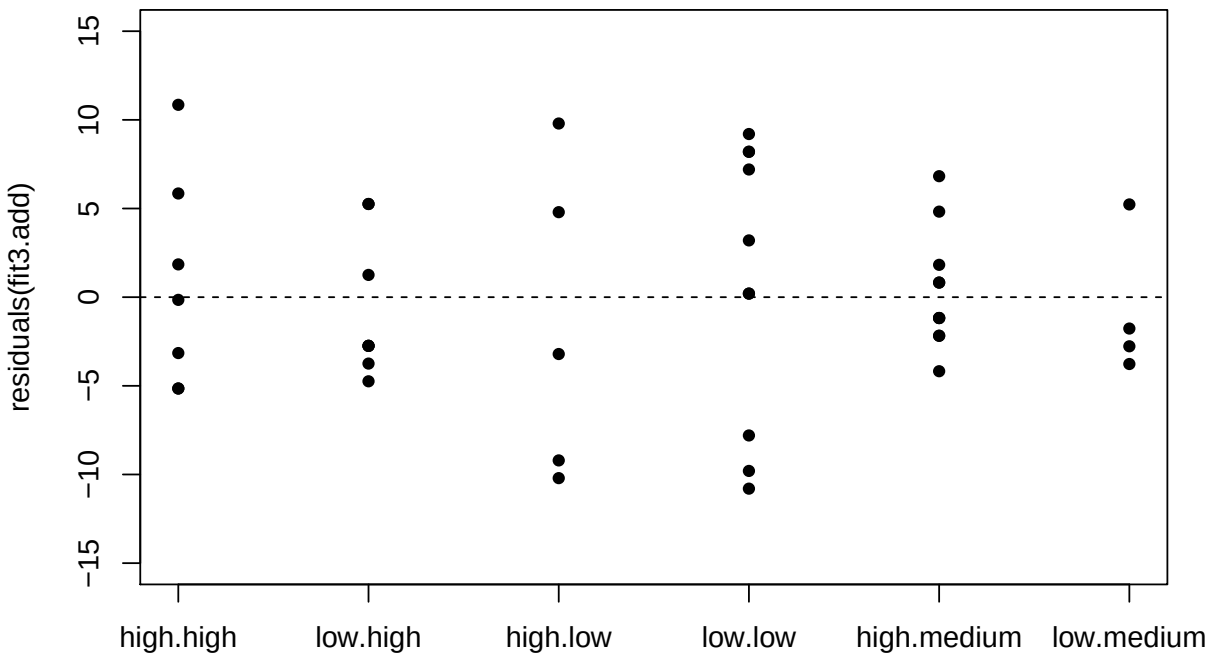
```
:  partner.status fcategory   lsmean       SE df lower.CL upper.CL
:  high           high      57.15022 1.779563 41 53.55632 60.74412
:  low            high      59.74356 1.716062 41 56.27790 63.20921
:  high           low       25.20444 1.923430 41 21.32000 29.08889
:  low            low       27.79778 1.609257 41 24.54782 31.04774
:  high           medium    43.17511 1.567330 41 40.00982 46.34040
:  low            medium    45.76844 2.002585 41 41.72414 49.81275
:
: Confidence level used: 0.95

Moore.fit<-Moore;Moore.fit$fit<-fitted(fit3.add)
aggregate(fit~partner.status+fcategory,data=Moore.fit,mean)    #Check

:   partner.status fcategory      fit
: 1           high      high 57.15022
: 2            low      high 59.74356
: 3           high       low 25.20444
: 4            low       low 27.79778
: 5           high    medium 43.17511
: 6            low    medium 45.76844
```

For the ANOVA using partial F-tests:

```
anova(fit3.add)

: Analysis of Variance Table
:
: Response: fscore
:                Df Sum Sq Mean Sq  F value Pr(>F)
: partner.status  1    7.9     7.9   0.2382 0.6281
: fcategory       2 7561.4  3780.7 113.5468 <2e-16 ***
: Residuals      41 1365.1    33.3
: ---
: Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For a residual plot:

```
par(mar=c(4,4,2,4))
stripchart(residuals(fit3.add)~partner.status+fcategory,data=Moore,
           pch=19,cex=0.8,ylim=range(-15,15),vertical=TRUE)
abline(h=0,lty=2)
```



Finally we now consider the "main effects plus interaction" model

`1+partner.status+fcategory+partner.status:fcategory`

```
fit4.add<-lm(fscore~partner.status*fcategory,data=Moore);summary(fit4.add)

:
: Call:
: lm(formula = fscore ~ partner.status * fcategory, data = Moore)
:
: Residuals:
:     Min      1Q  Median      3Q     Max
: -11.600  -3.000  -1.000   5.143  11.400
:
: Coefficients:
:                                 Estimate Std. Error t value Pr(>|t|)
: (Intercept)                      57.8571     2.2122  26.153  < 2e-16 ***
: partner.statuslow                 1.2679     3.0292   0.419    0.678
: fcategorylow                    -34.2571     3.4272  -9.996 2.59e-12 ***
: fcategorymedium                 -14.4026     2.8299  -5.089 9.44e-06 ***
: partner.statuslow:fcategorylow    3.7321     4.4106   0.846    0.403
: partner.statuslow:fcategorymedium 0.2776     4.5667   0.061    0.952
: ---
: Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
:
: Residual standard error: 5.853 on 39 degrees of freedom
: Multiple R-squared:  0.8505,Adjusted R-squared:  0.8313
: F-statistic: 44.36 on 5 and 39 DF,  p-value: 4.567e-15
```

We can display the fitted modelled means using the function `lsmeans`:

```
lsmeans(fit4.add, ~ partner.status * fcategory)

:  partner.status fcategory   lsmean       SE df lower.CL upper.CL
:  high           high      57.85714 2.212237 39 53.38247 62.33181
:  low            high      59.12500 2.069358 39 54.93933 63.31067
:  high           low       23.60000 2.617554 39 18.30550 28.89450
:  low            low       28.60000 1.850890 39 24.85622 32.34378
:  high           medium    43.45455 1.764754 39 39.88499 47.02410
:  low            medium    45.00000 2.926514 39 39.08057 50.91943
:
: Confidence level used: 0.95
```

For the ANOVA using partial F-tests:

```
anova(fit4.add)

: Analysis of Variance Table
:
: Response: fscore
:                          Df Sum Sq Mean Sq  F value Pr(>F)
: partner.status            1    7.9     7.9   0.2316 0.6331
: fcategory                 2 7561.4  3780.7 110.3593 <2e-16 ***
: partner.status:fcategory  2   29.1    14.5   0.4245 0.6571
: Residuals                39 1336.1    34.3
: ---
: Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For a residual plot:

```
par(mar=c(4,4,2,4))
stripchart(residuals(fit4.add)~partner.status+fcategory,data=Moore,
           pch=19,cex=0.8,ylim=range(-15,15),vertical=TRUE)
abline(h=0,lty=2)
```