Example of "treatment contrasts" used by R in estimating ANOVA coefficients

The first example shows a simple numerical design matrix in R (no factors) for the groups "1", "a", "b", "ab".

```
> resp <-  rnorm(4,0,1)
> data <-  data.frame(resp=resp,
+                     A=c(0,1,0,1),
+                     B=c(0,0,1,1))
> data
       resp A B
1  0.3112628 0 0
2 -0.7630832 1 0
3 -1.7276357 0 1
4 -0.5787976 1 1
> lm1.out <-  lm(resp~A*B,data=data)
```

Note that the columns of the design matrix have zeros where our effect contrasts had "-1"s. The results (fitted values and tests) are equivalent although this design matrix does not have orthogonal columns and the contrast coefficients do not sum to zero as can be seen from $X^T X$.

```
> X <-  model.matrix(lm1.out)
> X
  (Intercept) A B A:B
1           1 0 0   0
2           1 1 0   0
3           1 0 1   0
4           1 1 1   1
> X%*%t(X)

    1 2 3 4
  1 1 1 1 1
  2 1 2 1 2
  3 1 1 2 2
  4 1 2 2 4
> coef <-  solve(t(X)%*%X)%*%t(X)%*%data$resp
> coef

                   [,1]
  (Intercept)  0.3112628
  A           -1.0743461
  B           -2.0388985
  A:B          2.2231841
> lm1.out$coef
(Intercept)           A           B         A:B
  0.3112628  -1.0743461  -2.0388985   2.2231841
```

An equivalent example with factors rather than numeric design

```
> data <-  data.frame(resp=resp,
+                     A=c("f","m","f","m"),
+                     B=c("f","f","m","m"))
> data
        resp A B
1  0.3112628 f f
2 -0.7630832 m f
3 -1.7276357 f m
4 -0.5787976 m m
```

R drops the first group. Since f comes before m in the alphabet the contrast for "f" is dropped.

```
> contrasts(data$A)
  m
f 0
m 1

> contrasts(data$B)
  m
f 0
m 1

> lm2.out <-  lm(resp~A*B,data=data)
> X <-  as.num(model.matrix(lm2.out))
> X
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    1    1    0    0
[3,]    1    0    1    0
[4,]    1    1    1    1
```

Note that the coefficients calculated using the model matrix and those from the R fit match and match the previous example.

```
> coef <-  solve(t(X)%*%X)%*%t(X)%*%data$resp
> coef
           [,1]
[1,]  0.3112628
[2,] -1.0743461
[3,] -2.0388985
[4,]  2.2231841
> lm2.out$coef
(Intercept)          Am          Bm       Am:Bm
  0.3112628  -1.0743461  -2.0388985   2.2231841
> lm1.out$coef
(Intercept)           A           B         A:B
  0.3112628  -1.0743461  -2.0388985   2.2231841
```

Example of a 3x2 factorial. A has levels "low", "med" and "high" and B has levels "low" and "high".

```
> resp <-  rnorm(6,0,1)
> data <-  data.frame(resp=resp,
+                     A=c("low","med","high",
+                       "low","med","high"),
+                     B=c("low","low","low",
+                       "high","high","high"))
```

I'm specifying the factor levels here so we get them in order.

```
> data$A <-  factor(data$A,levels=c("low","med","high"))
> data$B <-  factor(data$B,levels=c("low","high"))
> data
       resp    A    B
1 -0.2601737  low  low
2  0.1730960  med  low
3 -0.7590816 high  low
4  0.9682281  low high
5 -0.4736121  med high
6 -0.3177872 high high
>
> contrasts(data$A)
     med high
low    0    0
med    1    0
high   0    1
> contrasts(data$B)
     high
low     0
high    1
>
> lm.out <-  lm(resp~A*B,data=data)
>
> X <-  model.matrix(lm.out)
> X
  (Intercept) Amed Ahigh Bhigh Amed:Bhigh Ahigh:Bhigh
1           1    0     0     0          0           0
2           1    1     0     0          0           0
3           1    0     1     0          0           0
4           1    0     0     1          0           0
5           1    1     0     1          1           0
6           1    0     1     1          0           1
attr(,"contrasts")
attr(,"contrasts")$A
[1] "contr.treatment"

attr(,"contrasts")$B
[1] "contr.treatment"
```

```
> X%*%t(X)

    1 2 3 4 5 6
  1 1 1 1 1 1 1
  2 1 2 1 1 2 1
  3 1 1 2 1 1 2
  4 1 1 1 2 2 2
  5 1 2 1 2 4 2
  6 1 1 2 2 2 4
> coef <-  solve(t(X)%*%X)%*%t(X)%*%data$resp
> coef

                   [,1]
  (Intercept) -0.2601737
  Amed         0.4332697
  Ahigh       -0.4989079
  Bhigh        1.2284018
  Amed:Bhigh  -1.8751099
  Ahigh:Bhigh -0.7871074
> lm.out$coef
(Intercept)        Amed       Ahigh       Bhigh  Amed:Bhigh Ahigh:Bhigh
 -0.2601737   0.4332697  -0.4989079   1.2284018  -1.8751099  -0.7871074
> summary(lm.out)

Residuals:
ALL 6 residuals are 0: no residual degrees of freedom!

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.2602         NA      NA       NA
Amed          0.4333         NA      NA       NA
Ahigh        -0.4989         NA      NA       NA
Bhigh         1.2284         NA      NA       NA
Amed:Bhigh   -1.8751         NA      NA       NA
Ahigh:Bhigh  -0.7871         NA      NA       NA
```

If A is a blocking variable and if we assume no iteration we can do testing.

```
> lm.out <-  lm(resp~A+B,data=data)
> summary(lm.out)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.1835     0.5436   0.338    0.768
Amed         -0.5043     0.6658  -0.757    0.528
Ahigh        -0.8925     0.6658  -1.340    0.312
Bhigh         0.3410     0.5436   0.627    0.595

> anova(lm.out)
Analysis of Variance Table

Response: resp
          Df  Sum Sq Mean Sq F value Pr(>F)
A          2 0.80098 0.40049  0.9035 0.5254
B          1 0.17442 0.17442  0.3935 0.5945
Residuals  2 0.88655 0.44328
```

R can create many kinds of contrasts. Note that the polynomial and Helmert contrasts are orthogonal.

```
> contr.treatment(2)
  2
1 0
2 1

> contr.treatment(5)
  2 3 4 5
1 0 0 0 0
2 1 0 0 0
3 0 1 0 0
4 0 0 1 0
5 0 0 0 1

> con <-  contr.poly(5)
> con
             .L         .Q            .C         ^4
[1,] -0.6324555  0.5345225 -3.162278e-01  0.1195229
[2,] -0.3162278 -0.2672612  6.324555e-01 -0.4780914
[3,]  0.0000000 -0.5345225 -4.095972e-16  0.7171372
[4,]  0.3162278 -0.2672612 -6.324555e-01 -0.4780914
[5,]  0.6324555  0.5345225  3.162278e-01  0.1195229
> round(t(tt)%*%tt,8)
     .L .Q .C ^4
  .L  1  0  0  0
  .Q  0  1  0  0
  .C  0  0  1  0
  ^4  0  0  0  1
>
> contr.sum(5)
  [,1] [,2] [,3] [,4]
1    1    0    0    0
2    0    1    0    0
3    0    0    1    0
4    0    0    0    1
5   -1   -1   -1   -1
>
> con <-  contr.helmert(5)
  [,1] [,2] [,3] [,4]
1   -1   -1   -1   -1
2    1   -1   -1   -1
3    0    2   -1   -1
4    0    0    3   -1
5    0    0    0    4
```

Here is the same example with polynomial contrasts for B.

```
> contrasts(data$A) <- contr.poly(3)
> contrasts(data$B) <- contr.treatment(2)
> data
        resp    A    B
1 -0.2601737  low  low
2  0.1730960  med  low
3 -0.7590816 high  low
4  0.9682281  low high
5 -0.4736121  med high
6 -0.3177872 high high
> lm.out <-  lm(resp~A+B,data=data)
> round(model.matrix(lm.out),4)
  (Intercept)     A.L     A.Q B2
1           1 -0.7071  0.4082  0
2           1  0.0000 -0.8165  0
3           1  0.7071  0.4082  0
4           1 -0.7071  0.4082  1
5           1  0.0000 -0.8165  1
6           1  0.7071  0.4082  1
attr(,"contrasts")
attr(,"contrasts")$A
                .L         .Q
low  -7.071068e-01  0.4082483
med  -7.850462e-17 -0.8164966
high  7.071068e-01  0.4082483

attr(,"contrasts")$B
     2
low  0
high 1

> summary(lm.out)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.2821     0.3844  -0.734    0.539
A.L          -0.6311     0.4708  -1.340    0.312
A.Q           0.0474     0.4708   0.101    0.929
B2            0.3410     0.5436   0.627    0.595

Residual standard error: 0.6658 on 2 degrees of freedom
Multiple R-Squared: 0.5239,     Adjusted R-squared: -0.1904
F-statistic: 0.7335 on 3 and 2 DF,  p-value: 0.6208

> anova(lm.out)
Analysis of Variance Table

Response: resp
          Df  Sum Sq Mean Sq F value Pr(>F)
A          2 0.80098 0.40049  0.9035 0.5254
B          1 0.17442 0.17442  0.3935 0.5945
Residuals  2 0.88655 0.44328
```

Here is an example with more than one observation per "cell".

```
> resp <-  rnorm(12,0,1)
> data <-  data.frame(resp=resp,
+                     A=c( "low","low"
+                       ,"med","med",
+                       "high","high",
+                       "low","low"
+                       ,"med","med",
+                       "high","high"),
+                     B=c(
+                       "low","low","low",
+                       "low","low","low",
+                       "high","high","high",
+                       "high","high","high"))
>
> data$A <-  factor(data$A,levels=c("low","med","high"))
> data$B <-  factor(data$B,levels=c("low","high"))
> data
          resp    A    B
1   3.33542288  low  low
2  -0.02190004  low  low
3  -0.34680683  med  low
4  -2.11507430  med  low
5   0.13964009 high  low
6   0.79374788 high  low
7   0.29565337  low high
8  -2.28921947  low high
9   1.50376653  med high
10  0.28410049  med high
11  0.57231880 high high
12 -0.76745620 high high
> lm.out <-  lm(resp~A*B,data=data)
> round(model.matrix(lm.out),4)
   (Intercept) Amed Ahigh Bhigh Amed:Bhigh Ahigh:Bhigh
1            1    0     0     0          0           0
2            1    0     0     0          0           0
3            1    1     0     0          0           0
4            1    1     0     0          0           0
5            1    0     1     0          0           0
6            1    0     1     0          0           0
7            1    0     0     1          0           0
8            1    0     0     1          0           0
9            1    1     0     1          1           0
10           1    1     0     1          1           0
11           1    0     1     1          0           1
12           1    0     1     1          0           1
attr(,"contrasts")
attr(,"contrasts")$A
[1] "contr.treatment"

attr(,"contrasts")$B
[1] "contr.treatment"
```

```
> summary(lm.out)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.657      1.016   1.630   0.1542
Amed          -2.888      1.437  -2.009   0.0913 .
Ahigh         -1.190      1.437  -0.828   0.4394
Bhigh         -2.654      1.437  -1.846   0.1144
Amed:Bhigh     4.778      2.033   2.351   0.0570 .
Ahigh:Bhigh    2.089      2.033   1.028   0.3437
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.437 on 6 degrees of freedom
Multiple R-Squared: 0.5001,     Adjusted R-squared: 0.08353
F-statistic: 1.201 on 5 and 6 DF,  p-value: 0.4087

> anova(lm.out)
Analysis of Variance Table

Response: resp
          Df  Sum Sq Mean Sq F value Pr(>F)
A          2  0.5257  0.2629  0.1272 0.8828
B          1  0.3982  0.3982  0.1927 0.6760
A:B        2 11.4766  5.7383  2.7777 0.1400
Residuals  6 12.3952  2.0659
>
```