Two qualitative predictors

Mauro Gasparini*

MML, April 30, 2025

This is a stand-alone R Markdown file "240525 Two factors.rmd" which produces a "240525 Two factors.pdf" file when "knit" within R, according to the notebook style of R Markdown.

Two factors: a linear model with two qualitative predictors

We recycle a textbook example taken from McClave JT., Benson PG. e Sincich T. (2014). *Statistics for Business and Economics.* Pearson Education Limited.
We want to study the effect on the response (variable *distance*) of 4 different brands of golf ball (A,B,C,D) (variable *brand*) and the club type (DRIVER/IRON) (variable *club*). These two features are qualitative predictors, also called factors. A robot player is used.

Let's build the data in *wide* format, i.e. enter the data online. The data shows 4 replications for each pair of brand and club)

```
golfwide <- read.table(header=T, text='
club    A       B       C     D
DRIVER  226.4   238.3   240.5   219.8
DRIVER  232.6   231.7   246.9   228.7
DRIVER  234.0 227.7 240.3   232.9
DRIVER  220.7   237.2   244.7   237.6
IRON    163.8   184.4   179.0   157.8
IRON    179.4   180.6   168.0   161.8
IRON    168.6   179.5   165.2   162.1
IRON    173.4   186.2   156.5   160.3
')
```

To tranform from wide format to long format, we use the library *tidyr()*.

```
library(tidyr)
golflong <- gather(golfwide, brand, distance, A:D)
golflong
```

```
##      club brand distance
## 1  DRIVER     A    226.4
## 2  DRIVER     A    232.6
## 3  DRIVER     A    234.0
## 4  DRIVER     A    220.7
## 5    IRON     A    163.8
## 6    IRON     A    179.4
## 7    IRON     A    168.6
## 8    IRON     A    173.4
## 9  DRIVER     B    238.3
```

*Politecnico di Torino, mauro.gasparini@polito.it

```
## 10 DRIVER     B    231.7
## 11 DRIVER     B    227.7
## 12 DRIVER     B    237.2
## 13   IRON     B    184.4
## 14   IRON     B    180.6
## 15   IRON     B    179.5
## 16   IRON     B    186.2
## 17 DRIVER     C    240.5
## 18 DRIVER     C    246.9
## 19 DRIVER     C    240.3
## 20 DRIVER     C    244.7
## 21   IRON     C    179.0
## 22   IRON     C    168.0
## 23   IRON     C    165.2
## 24   IRON     C    156.5
## 25 DRIVER     D    219.8
## 26 DRIVER     D    228.7
## 27 DRIVER     D    232.9
## 28 DRIVER     D    237.6
## 29   IRON     D    157.8
## 30   IRON     D    161.8
## 31   IRON     D    162.1
## 32   IRON     D    160.3
```

Let us also use the old-fashioned attach/detach dynamics and study first the effect of **brand** only (a linear model with one factor only)

```
attach(golflong)
summary(lm(distance ~ brand))
```

```
##
## Call:
## lm(formula = distance ~ brand)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.638 -31.703  -0.481  32.947  42.475
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  199.862     12.262  16.299 8.04e-16 ***
## brandB         8.338     17.341   0.481    0.634
## brandC         5.275     17.341   0.304    0.763
## brandD        -4.737     17.341  -0.273    0.787
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.68 on 28 degrees of freedom
## Multiple R-squared:  0.02322,    Adjusted R-squared:  -0.08143
## F-statistic: 0.2219 on 3 and 28 DF,  p-value: 0.8804
```

```
anova(lm(distance ~ brand))
```

```
## Analysis of Variance Table
##
## Response: distance
```

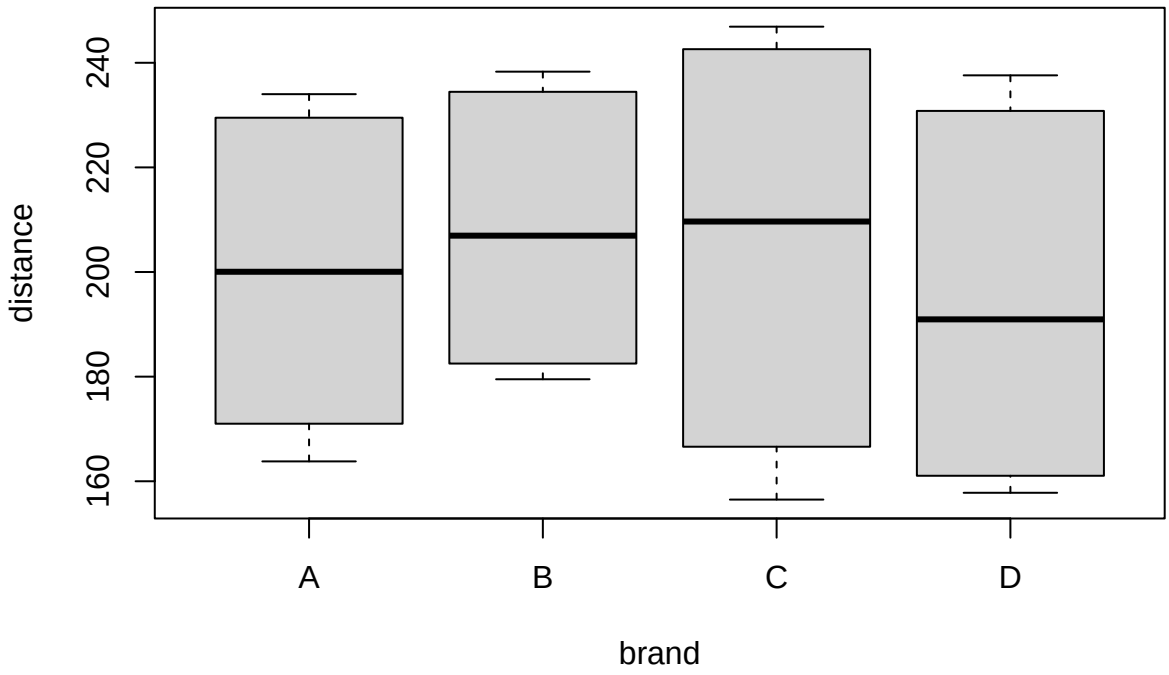
```
##           Df Sum Sq Mean Sq F value Pr(>F)
## brand      3    801  266.91  0.2219 0.8804
## Residuals 28  33681 1202.90
```

```
model.matrix(lm(distance~brand)) ### this is the design matrix
```

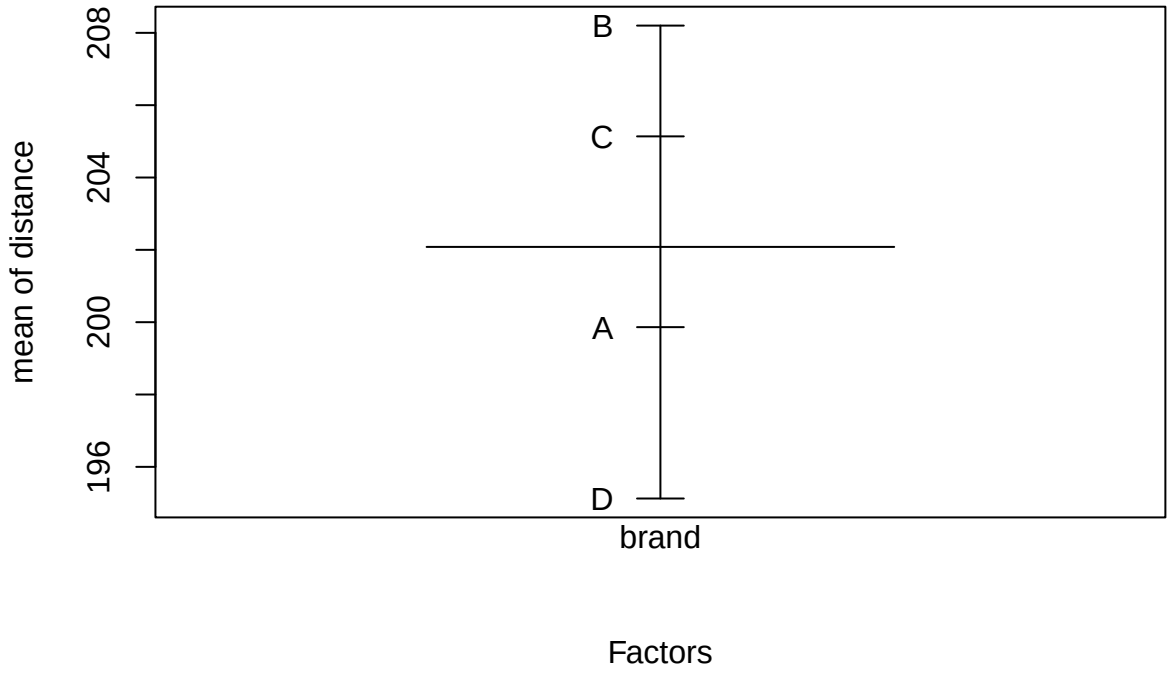
```
##    (Intercept) brandB brandC brandD
## 1            1      0      0      0
## 2            1      0      0      0
## 3            1      0      0      0
## 4            1      0      0      0
## 5            1      0      0      0
## 6            1      0      0      0
## 7            1      0      0      0
## 8            1      0      0      0
## 9            1      1      0      0
## 10           1      1      0      0
## 11           1      1      0      0
## 12           1      1      0      0
## 13           1      1      0      0
## 14           1      1      0      0
## 15           1      1      0      0
## 16           1      1      0      0
## 17           1      0      1      0
## 18           1      0      1      0
## 19           1      0      1      0
## 20           1      0      1      0
## 21           1      0      1      0
## 22           1      0      1      0
## 23           1      0      1      0
## 24           1      0      1      0
## 25           1      0      0      1
## 26           1      0      0      1
## 27           1      0      0      1
## 28           1      0      0      1
## 29           1      0      0      1
## 30           1      0      0      1
## 31           1      0      0      1
## 32           1      0      0      1
## attr(,"assign")
## [1] 0 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$brand
## [1] "contr.treatment"
```

To visualize dta, we can use boxplots or more specialized graphics for qualitative predictors (for the latter, we have to define the predictors explicitely as factor objects in R). Recall that prettier graphics can always be produced using *ggplot2()*.

```
boxplot(distance ~ brand)
```



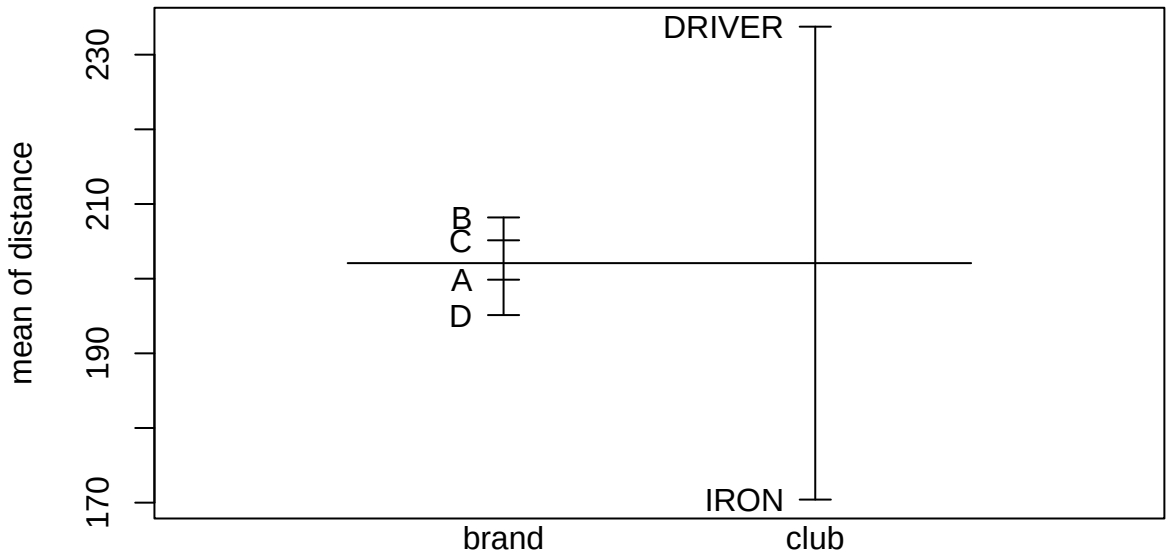

```
brand <- as.factor(brand)
club <- as.factor(club)
plot.design(distance ~ brand)
```



Additive and non-additive models

Now let us add the second factor *club* and build a 'small' additive model ...

```
plot.design(distance ~ brand  + club)
```



Factors

```
small <- lm(distance ~ brand  + club)
summary(small)
```

```
##
## Call:
## lm(formula = distance ~ brand + club)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.9688  -5.2156   0.7375   5.2875  11.2063
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  231.531      3.032  76.371   <2e-16 ***
## brandB         8.338      3.835   2.174   0.0386 *
## brandC         5.275      3.835   1.376   0.1803
## brandD        -4.737      3.835  -1.235   0.2273
## clubIRON     -63.337      2.712 -23.358   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.67 on 27 degrees of freedom
## Multiple R-squared:  0.9539, Adjusted R-squared:  0.9471
## F-statistic: 139.8 on 4 and 27 DF,  p-value: < 2.2e-16
```

```
anova(small)
```

```
## Analysis of Variance Table
##
## Response: distance
##           Df Sum Sq Mean Sq  F value  Pr(>F)
## brand      3    801     267   4.5376 0.01061 *
## club       1  32093   32093 545.5946 < 2e-16 ***
## Residuals 27   1588      59
```

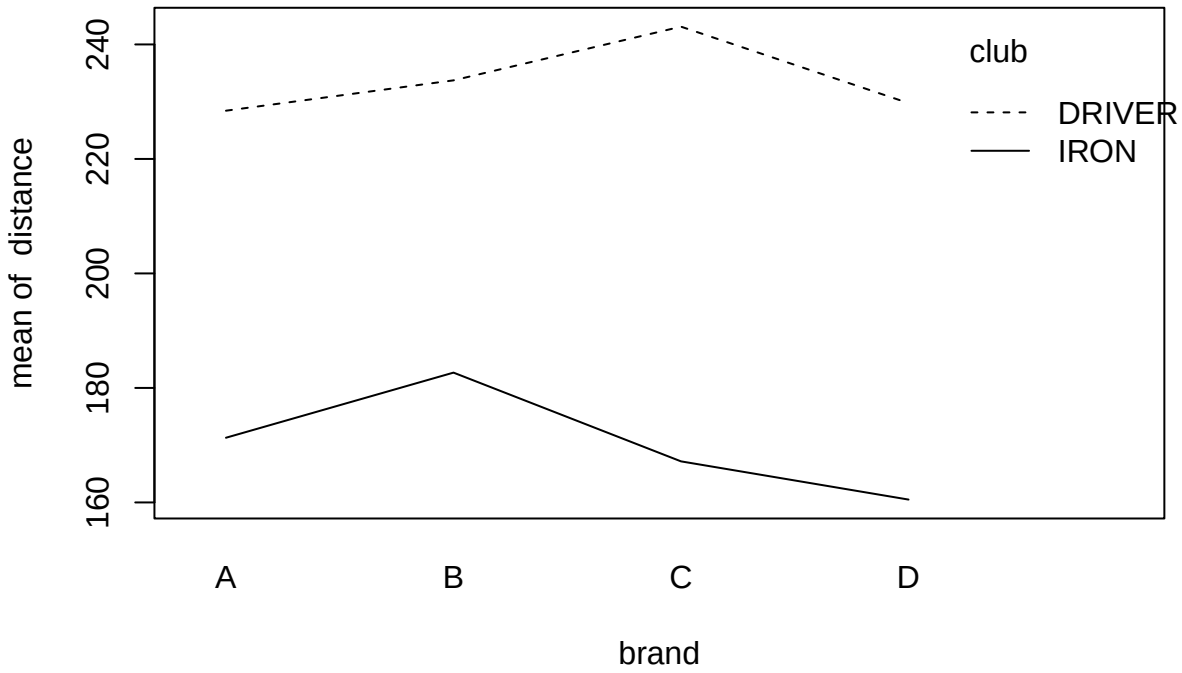
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model.matrix(small)
```

```
##    (Intercept) brandB brandC brandD clubIRON
## 1            1      0      0      0        0
## 2            1      0      0      0        0
## 3            1      0      0      0        0
## 4            1      0      0      0        0
## 5            1      0      0      0        1
## 6            1      0      0      0        1
## 7            1      0      0      0        1
## 8            1      0      0      0        1
## 9            1      1      0      0        0
## 10           1      1      0      0        0
## 11           1      1      0      0        0
## 12           1      1      0      0        0
## 13           1      1      0      0        1
## 14           1      1      0      0        1
## 15           1      1      0      0        1
## 16           1      1      0      0        1
## 17           1      0      1      0        0
## 18           1      0      1      0        0
## 19           1      0      1      0        0
## 20           1      0      1      0        0
## 21           1      0      1      0        1
## 22           1      0      1      0        1
## 23           1      0      1      0        1
## 24           1      0      1      0        1
## 25           1      0      0      1        0
## 26           1      0      0      1        0
## 27           1      0      0      1        0
## 28           1      0      0      1        0
## 29           1      0      0      1        1
## 30           1      0      0      1        1
## 31           1      0      0      1        1
## 32           1      0      0      1        1
## attr(,"assign")
## [1] 0 1 1 1 2
## attr(,"contrasts")
## attr(,"contrasts")$brand
## [1] "contr.treatment"
##
## attr(,"contrasts")$club
## [1] "contr.treatment"
```

. . . and a larger model with interaction. An interaction plot suggests interactions may be significant, but we have to test for it since interaction plots are subject to sampling variation.

```
#plot.design(distance ~ brand*club)
interaction.plot(brand, club, distance)
```




```
large <- lm(distance ~ brand*club)
summary(large)
```

```
##
## Call:
## lm(formula = distance ~ brand * club)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.6750  -2.7000   0.3125   3.4875  11.8250
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      228.425      2.927  78.051  < 2e-16 ***
## brandB             5.300      4.139   1.281  0.21259
## brandC            14.675      4.139   3.546  0.00165 **
## brandD             1.325      4.139   0.320  0.75163
## clubIRON         -57.125      4.139 -13.802 6.55e-13 ***
## brandB:clubIRON    6.075      5.853   1.038  0.30966
## brandC:clubIRON  -18.800      5.853  -3.212  0.00373 **
## brandD:clubIRON  -12.125      5.853  -2.072  0.04923 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.853 on 24 degrees of freedom
## Multiple R-squared:  0.9762, Adjusted R-squared:  0.9692
## F-statistic: 140.4 on 7 and 24 DF,  p-value: < 2.2e-16
```

```
model.matrix(large)
```

```
##    (Intercept) brandB brandC brandD clubIRON brandB:clubIRON brandC:clubIRON
## 1            1      0      0      0        0               0               0
## 2            1      0      0      0        0               0               0
```

```
## 3            1      0      0      0        0                0                0
## 4            1      0      0      0        0                0                0
## 5            1      0      0      0        1                0                0
## 6            1      0      0      0        1                0                0
## 7            1      0      0      0        1                0                0
## 8            1      0      0      0        1                0                0
## 9            1      1      0      0        0                0                0
## 10           1      1      0      0        0                0                0
## 11           1      1      0      0        0                0                0
## 12           1      1      0      0        0                0                0
## 13           1      1      0      0        1                1                0
## 14           1      1      0      0        1                1                0
## 15           1      1      0      0        1                1                0
## 16           1      1      0      0        1                1                0
## 17           1      0      1      0        0                0                0
## 18           1      0      1      0        0                0                0
## 19           1      0      1      0        0                0                0
## 20           1      0      1      0        0                0                0
## 21           1      0      1      0        1                0                1
## 22           1      0      1      0        1                0                1
## 23           1      0      1      0        1                0                1
## 24           1      0      1      0        1                0                1
## 25           1      0      0      1        0                0                0
## 26           1      0      0      1        0                0                0
## 27           1      0      0      1        0                0                0
## 28           1      0      0      1        0                0                0
## 29           1      0      0      1        1                0                0
## 30           1      0      0      1        1                0                0
## 31           1      0      0      1        1                0                0
## 32           1      0      0      1        1                0                0
##    brandD:clubIRON
## 1                0
## 2                0
## 3                0
## 4                0
## 5                0
## 6                0
## 7                0
## 8                0
## 9                0
## 10               0
## 11               0
## 12               0
## 13               0
## 14               0
## 15               0
## 16               0
## 17               0
## 18               0
## 19               0
## 20               0
## 21               0
## 22               0
## 23               0
```

```
## 24               0
## 25               0
## 26               0
## 27               0
## 28               0
## 29               1
## 30               1
## 31               1
## 32               1
## attr(,"assign")
## [1] 0 1 1 1 2 3 3 3
## attr(,"contrasts")
## attr(,"contrasts")$brand
## [1] "contr.treatment"
##
## attr(,"contrasts")$club
## [1] "contr.treatment"
```

```
anova(large)
```

```
## Analysis of Variance Table
##
## Response: distance
##            Df Sum Sq Mean Sq  F value    Pr(>F)
## brand       3    801     267   7.7908 0.0008401 ***
## club        1  32093   32093 936.7516 < 2.2e-16 ***
## brand:club  3    766     255   7.4524 0.0010789 **
## Residuals  24    822      34
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With the last *anova()* command we have build a more traditional anova table. We can compare the two models also with the *anova* command.

```
anova(small, large)
```

```
## Analysis of Variance Table
##
## Model 1: distance ~ brand + club
## Model 2: distance ~ brand * club
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1     27 1588.20
## 2     24  822.24  3    765.96 7.4524 0.001079 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interactions are significant, meaning each combination of the two factors tells a different story.

```
fitted(large)
```

```
##       1       2       3       4       5       6       7       8       9      10
## 228.425 228.425 228.425 228.425 171.300 171.300 171.300 171.300 233.725 233.725
##      11      12      13      14      15      16      17      18      19      20
## 233.725 233.725 182.675 182.675 182.675 182.675 243.100 243.100 243.100 243.100
##      21      22      23      24      25      26      27      28      29      30
## 167.175 167.175 167.175 167.175 229.750 229.750 229.750 229.750 160.500 160.500
##      31      32
```

```
## 160.500 160.500
```

Fitted and predicted value, with confidence and prediction intervals

The following code is an application of the concepts of fitted values, confidence intervals, predictive intervals.

```
predict.lm(large, newdata=data.frame(brand="A",club="DRIVER"))
```

```
##       1
## 228.425
```

```
predict.lm(large, newdata=data.frame(brand="A",club="DRIVER"),
           interval="confidence",
           level=.99)
```

```
##       fit      lwr      upr
## 1 228.425 220.2395 236.6105
```

```
predict.lm(large, newdata=data.frame(brand="A",club="DRIVER"),
           interval="prediction",
           level=.99)
```

```
##       fit      lwr      upr
## 1 228.425 210.1216 246.7284
```

```
detach(golflong)
```