Introduction to statistics: Contrast coding

Shravan Vasishth

Universität Potsdam
vasishth@uni-potsdam.de
http://www.ling.uni-potsdam.de/∼vasishth

April 9, 2020

The goals of this lecture

Goals of this lecture:

- Understand contrast coding for two condition experiments.
- Learn different (custom) contrast codings for more complex designs, such as 2×2 factorial designs.

Typical experiment designs

- Two conditions: 1×2 "factorial design"
- Three conditions: 1×3 "factorial design"
- Four conditions: typically 2×2 factorial design, sometimes 1×4
- More complex designs like 2×3, $2 \times 2 \times 2$, etc.

Typical experiment designs

- We will not go beyond 2×2.
- My advice for your own work is to keep it simple: four conditions maximum.
- Sometimes one has to go beyond such simple designs, but do that only if you really understand that you are introducing possibly intractable complexity in the design (this will be discussed later, lectures on simulating data).

Typical experiment designs

- A classic beginner mistake (I have done this) is to design an experiment without any clear hypotheses in mind.
- After the data come in, we start speculating and trying out different pairwise comparisons.
- A better way: develop an analysis plan in advance using simulated data. This requires some mental discipline.
- Define *a priori* comparisons.
- Pre-registration is a useful tool (to be discussed later).

Generating simulated data

We can generate between-subjects data using a function called mixedDesign.

```
library(dplyr)
# load mixedDesign function for simulating data
source("functions/mixedDesign.v0.6.3.R")
M <- matrix(c(0.8, 0.4), nrow=2, ncol=1, byrow=FALSE)
set.seed(1)
# set seed of random number generator for replicability
```

Generating simulated data

Generate between-subjects data:

```
simdat <- mixedDesign(B=2, W=NULL, n=5,
                      M=M,  SD=.20, long = TRUE)

##
## Attaching package:  'MASS'
## The following object is masked from 'package:dplyr':
##
##     select

t(xtabs(~id+B_A,simdat))

##      id
## B_A   1 2 3 4 5 6 7 8 9 10
##   A1  1 1 1 1 1 0 0 0 0  0
##   A2  0 0 0 0 0 1 1 1 1  1
```

Generating simulated data

We can generate within-subjects data as well, but some changes to the function are needed:

```
simdat_w <- mixedDesign(W=2, B=NULL, n=5,
                        M=t(M),  SD=.20, long = TRUE)
t(xtabs(~id+W_a,simdat_w))

##      id
## W_a   1 2 3 4 5
##   a1  1 1 1 1 1
##   a2  1 1 1 1 1
```

Generating simulated data

Using between-subjects simulated data here, rename the columns:

```
names(simdat)[1] <- "F"  # Rename B_A to F(actor)
levels(simdat$F) <- c("F1", "F2")
head(simdat)

##    F id      DV
## 1 F1  1 0.99664
## 2 F1  2 0.84693
## 3 F1  3 0.71202
## 4 F1  4 0.49940
## 5 F1  5 0.94502
## 6 F2  6 0.18293
```

Generating simulated data

```
str(simdat)

## 'data.frame': 10 obs. of  3 variables:
##  $ F : Factor w/ 2 levels "F1","F2": 1 1 1 1 1 2 2 2 2 2
##  $ id: Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5
##  $ DV: num  0.997 0.847 0.712 0.499 0.945 ...
```

Generating simulated data

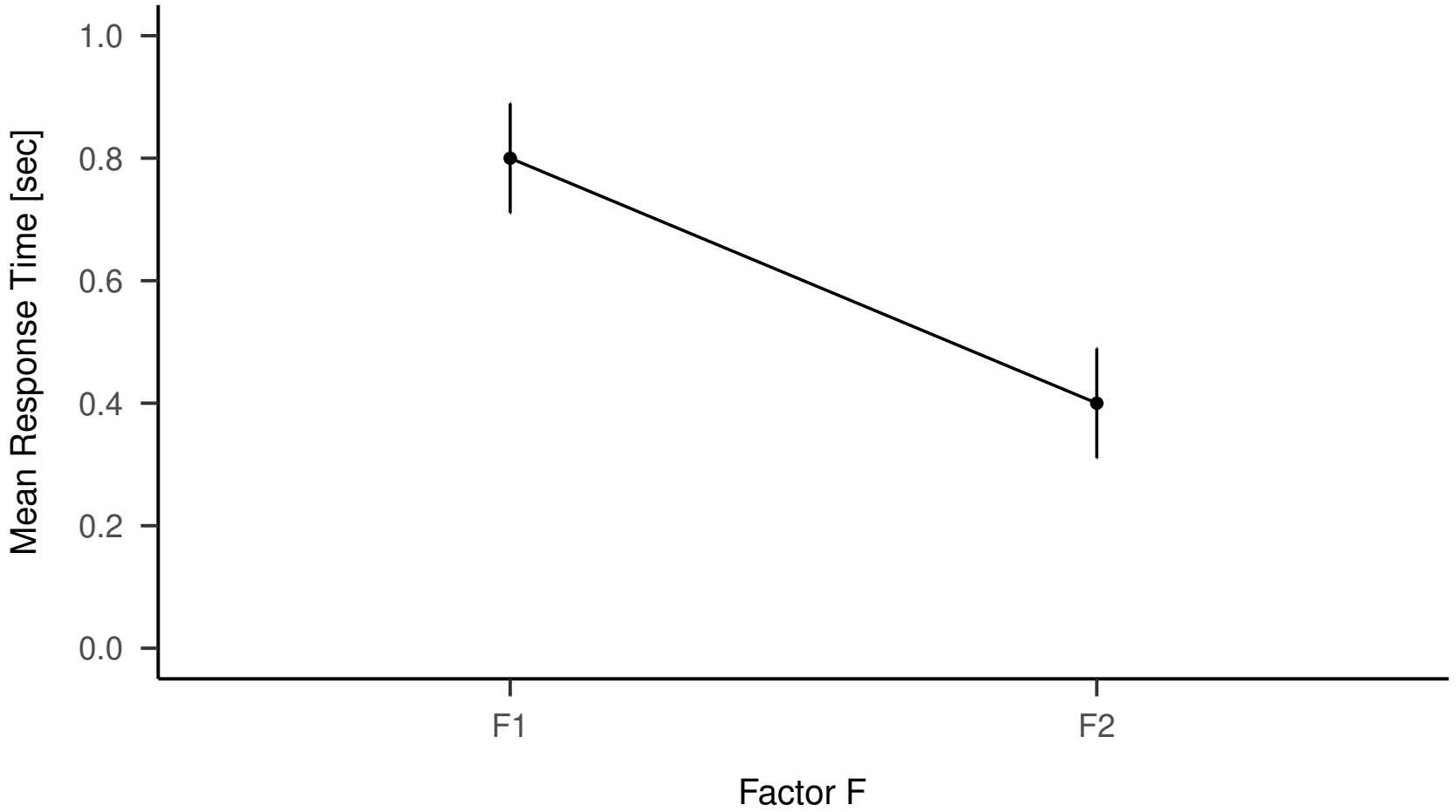
```
table1 <- simdat %>% group_by(F) %>%
  # Table for main effect F
   summarize(N=n(), M=mean(DV),
             SD=sd(DV), SE=SD/sqrt(N) )
table1
```

```
## # A tibble: 2 x 5
##   F          N     M    SD      SE
##   <fct> <int> <dbl> <dbl>  <dbl>
## 1 F1        5   0.8  0.20 0.0894
## 2 F2        5   0.4  0.2  0.0894
```

```
(GM <-  mean(table1$M)) # Grand Mean
```

```
## [1] 0.6
```

Generating simulated data



Treatment contrasts

```
m_F <- lm(DV ~ F, simdat)
round(summary(m_F)$coef,3)

##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.8      0.089   8.944    0.000
## FF2             -0.4      0.126  -3.162    0.013
```

$$\begin{aligned} \text{Intercept} = \quad & \hat{\mu}_1 \quad && = \text{estimated mean for F1} \\ \text{Slope (FF2)} = \quad & \hat{\mu}_2 - \hat{\mu}_1 \quad && = \text{estim. mean for F2} - \text{estim. mean for F1} \end{aligned} \tag{1}$$

Treatment contrasts

```
contrasts(simdat$F)
```

```
##    F2
## F1  0
## F2  1
```

The TREATMENT CONTRAST expresses the null hypothesis that the difference in means between the two levels of the factor F is 0; formally, the null hypothesis H_0 is that $H_0 : \beta_1 = 0$:

$$H_0 : -1 \cdot \mu_{F1} + 1 \cdot \mu_{F2} = 0 \tag{2}$$

or equivalently:

$$H_0 : \mu_{F2} - \mu_{F1} = 0 \tag{3}$$

Treatment contrasts

The intercept in the TREATMENT CONTRAST expresses a null hypothesis that is usually of no interest: that the mean in condition F1 of the factor F is 0. Formally, the null hypothesis is $H_0 : \beta_0 = 0$:

$$H_0 : 1 \cdot \mu_{F1} + 0 \cdot \mu_{F2} = 0 \tag{4}$$

or equivalently:

$$H_0 : \mu_{F1} = 0. \tag{5}$$

Treatment contrasts

Level-ordering is alphabetical and can be changed:

```
simdat$Fb <- factor(simdat$F,  levels = c("F2","F1"))
contrasts(simdat$Fb)

##    F1
## F2  0
## F1  1
```

Sum contrasts

```
(contrasts(simdat$F) <- c(-0.5,+0.5))

## [1] -0.5  0.5

m1_mr <- lm(DV ~ F, simdat)
```

Sum contrasts

```
summary(m1_mr)

##
## Call:
## lm(formula = DV ~ F, data = simdat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3006 -0.1755  0.0524  0.1530  0.2084
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.6000     0.0632    9.49  1.3e-05
## F1           -0.4000     0.1265   -3.16    0.013
##
```

Sum contrasts

$$\begin{aligned} \text{Intercept} = & \quad (\hat{\mu}_1 + \hat{\mu}_2)/2 & = \text{estimated mean of } \texttt{F1} \text{ and } \texttt{F2} \\ \text{Slope } (\texttt{F1}) = & \quad \hat{\mu}_2 - \hat{\mu}_1 & = \text{est. mean for } \texttt{F2} - \text{estim. mean for } \texttt{F1} \end{aligned} \tag{6}$$

Sum contrasts

Sum contrasts express the null hypothesis that the difference in means between the two levels of factor F is 0; formally, the null hypothesis H_0 is that

$$H_0 : -1 \cdot \mu_{F1} + 1 \cdot \mu_{F2} = 0 \tag{7}$$

This is the same hypothesis that was also tested by the slope in the treatment contrast.

Sum contrasts

The intercept, however, now expresses a different hypothesis about the data: it expresses the null hypothesis that the average of the two conditions F1 and F2 is 0:

$$H_0 : 1/2 \cdot \mu_{F1} + 1/2 \cdot \mu_{F2} = \frac{\mu_{F1} + \mu_{F2}}{2} = 0 \tag{8}$$

Summary: Treatment and sum contrasts

- To summarize, TREATMENT CONTRASTS and SUM CONTRASTS are two possible ways to parameterize the difference between two groups; they test different hypotheses.
- TREATMENT CONTRASTS compare one or more means against a baseline condition, whereas SUM CONTRASTS allow us to determine whether we can reject the null hypothesis that a condition's mean is the same as the GM (in the two-group case, this also implies a hypothesis test that the two group means are the same).

Example: simulated data with a three-level factor

```
M <- matrix(c(500, 450, 400),
            nrow=3, ncol=1, byrow=FALSE)
set.seed(1)
simdat2 <- mixedDesign(B=3, W=NULL,
                       n=4, M=M,  SD=20, long = TRUE)
names(simdat2)[1] <- "F"  # Rename B_A to F(actor)/F(reque
levels(simdat2$F) <- c("low", "medium", "high")
simdat2$DV <- round(simdat2$DV)
```

Example: simulated data with a three-level factor

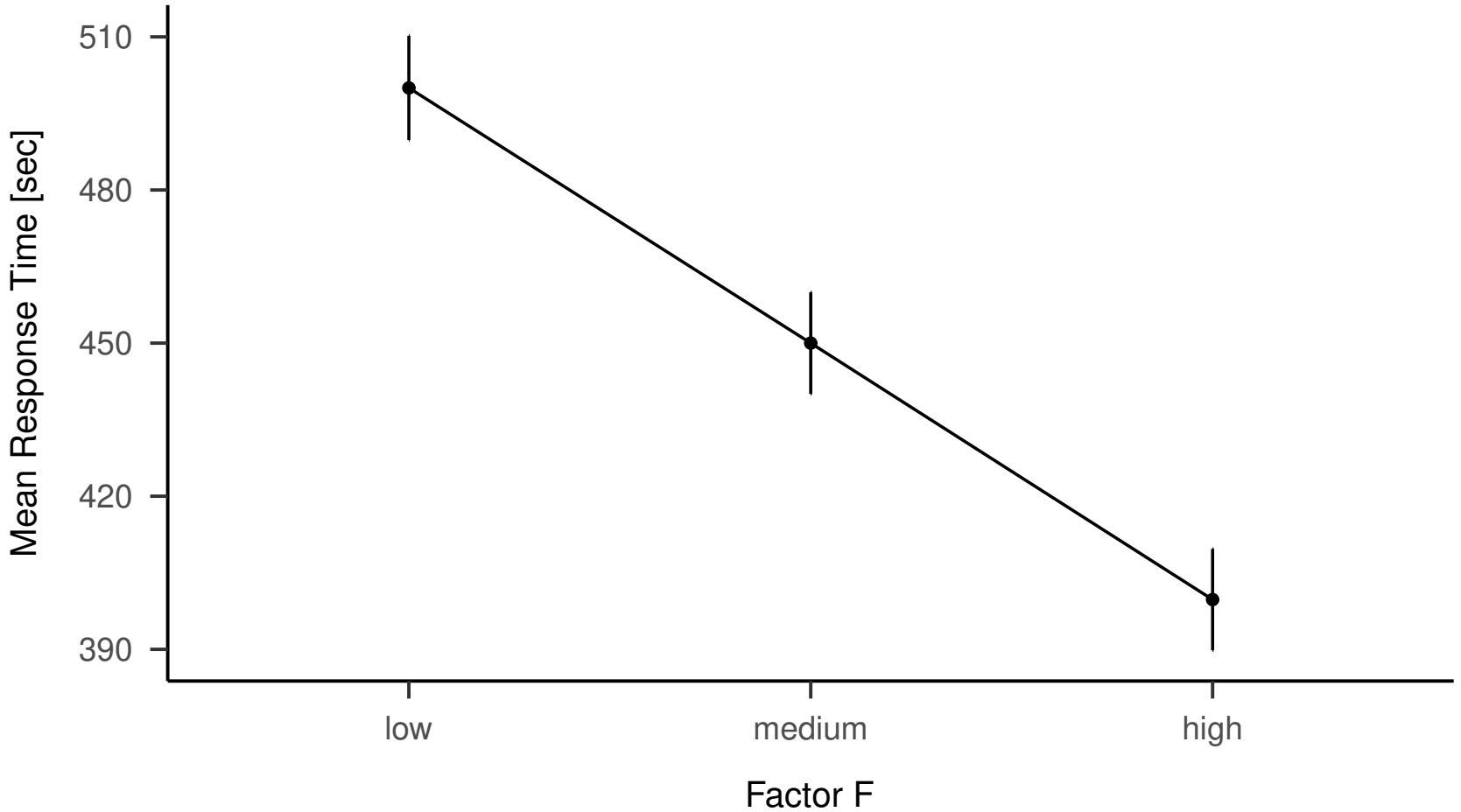
```
head(simdat2)

##        F id  DV
## 1    low  1 497
## 2    low  2 474
## 3    low  3 523
## 4    low  4 506
## 5 medium  5 422
## 6 medium  6 467

table.word <- simdat2 %>% group_by(F) %>%
  summarise(N = length(DV), M = mean(DV),
            SD = sd(DV), SE = sd(DV)/sqrt(N))
```

Example: simulated data with a three-level factor

```
table.word1 <- table.word
names(table.word1) <- c("Factor F","N data points","Estimated means",
                        "Standard deviations","Standard errors")
table.word1
```

```
## # A tibble: 3 x 5
##   `Factor F` `N data points` `Estimated mean... `Standard devia...
##   <fct>                <int>              <dbl>             <dbl>
## 1 low                      4                500              20.4
## 2 medium                   4                450              19.9
## 3 high                     4                400.             19.9
## # ... with 1 more variable: `Standard errors` <dbl>
```

Generating simulated data



Example: simulated data with a three-level factor

The estimated means reflect our assumptions about the true means in the data simulation: Response times decrease with increasing word frequency.

```
aovF <- aov(DV ~ F + Error(id), data=simdat2)
summary(aovF)

##
## Error: id
##           Df Sum Sq Mean Sq F value  Pr(>F)
## F          2  20100   10050    24.9 0.00021
## Residuals  9   3629     403
```

Example: simulated data with a three-level factor

- Notice that the ANOVA only tells that there is some difference, not which conditions are different from which.
- People often do an ANOVA and only if they find some difference do they do pairwise comparisons.
- But this two-step procedure is unnecessary; you can simply specify your hypotheses directly through contrast coding.

Example: simulated data with a three-level factor

Sum contrasts

Define three hypotheses to test:

$$H_{0_0} : \frac{\mu_1 + \mu_2 + \mu_3}{3} = 0 \tag{9}$$

$$H_{0_1} : \mu_1 = \frac{\mu_1 + \mu_2 + \mu_3}{3} = GM \tag{10}$$

and

$$H_{0_2} : \mu_2 = \frac{\mu_1 + \mu_2 + \mu_3}{3} = GM \tag{11}$$

Example: simulated data with a three-level factor

Sum contrasts

H_{0_0} can be written as:

$$H_{0_0} : \frac{1}{3}\mu_1 + \frac{1}{3}\mu_2 + \frac{1}{3}\mu_3 = 0 \tag{12}$$

Example: simulated data with a three-level factor

Sum contrasts

H_{0_1} can be written as:

$$\mu_1 = \frac{\mu_1 + \mu_2 + \mu_3}{3} \tag{13}$$

$$\Leftrightarrow \mu_1 - \frac{\mu_1 + \mu_2 + \mu_3}{3} = 0 \tag{14}$$

$$\Leftrightarrow \frac{2}{3}\mu_1 - \frac{1}{3}\mu_2 - \frac{1}{3}\mu_3 = 0 \tag{15}$$

Here, the weights $2/3, -1/3, -1/3$ are informative about how to combine the condition means to define the null hypothesis.

Example: simulated data with a three-level factor

Sum contrasts

H_{0_2} can be rewritten as:

$$\mu_2 = \frac{\mu_1 + \mu_2 + \mu_3}{3} \tag{16}$$

$$\Leftrightarrow \mu_2 - \frac{\mu_1 + \mu_2 + \mu_3}{3} = 0 \tag{17}$$

$$\Leftrightarrow -\frac{1}{3}\mu_1 + \frac{2}{3}\mu_2 - \frac{1}{3}\mu_3 = 0 \tag{18}$$

Here, the weights are $-1/3, 2/3, -1/3$, and they again indicate how to combine the condition means for defining the null hypothesis.

The hypothesis matrix

We can write the weights for the hypotheses in two columns of a matrix:

```
HcSum <- rbind(cH00=c(low= 1/3, med= 1/3, hi= 1/3),
               cH01=c(low=+2/3, med=-1/3, hi=-1/3),
               cH02=c(low=-1/3, med=+2/3, hi=-1/3))
fractions(t(HcSum))

##      cH00 cH01 cH02
## low   1/3  2/3 -1/3
## med   1/3 -1/3  2/3
## hi    1/3 -1/3 -1/3
```

The hypothesis matrix

The generalized inverse (details require matrix algebra, which we will just turn to):

```
ginv2 <- function(x)
  # define a function to make the output nicer
  fractions(provideDimnames(ginv(x),
                            base=dimnames(x)[2:1]))
```

The hypothesis matrix

Take the generalized inverse of the hypothesis matrix:

```
(XcSum <- ginv2(HcSum))

##      cH00 cH01 cH02
## low     1    1    0
## med     1    0    1
## hi      1   -1   -1
```

Compare the second and third columns of the inverse of the hypothesis matrix with the sum contrast matrix:

```
contr.sum(3)

##   [,1] [,2]
## 1    1    0
## 2    0    1
## 3   -1   -1
```

The hypothesis matrix

```
contrasts(simdat2$F) <- XcSum[,2:3]
## same as:
#contrasts(simdat2£F) <- contr.sum(3)
m1_mr <- lm(DV ~ F, data=simdat2)
```

The hypothesis matrix

Recall that in matrix form, the linear model is:
$DV = X\beta + \varepsilon$, where X is:

```
head(round(model.matrix(m1_mr)))

##   (Intercept) FcH01 FcH02
## 1           1     1     0
## 2           1     1     0
## 3           1     1     0
## 4           1     1     0
## 5           1     0     1
## 6           1     0     1
```

You can see that the model or design matrix X is based on the generalized inverse of the hypothesis matrix.

The hypothesis matrix

```
summary(m1_mr)

##
## Call:
## lm(formula = DV ~ F, data = simdat2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -28.75  -8.75   4.12  12.75  23.00
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 449.9167     5.7965   77.62  4.9e-14
## FcH01        50.0833     8.1975    6.11  0.00018
## FcH02         0.0833     8.1975    0.01  0.99211
##
## Residual standard error: 20.1 on 9 degrees of freedom
## Multiple R-squared:  0.847,Adjusted R-squared:  0.813
## F-statistic: 24.9 on 2 and 9 DF,  p-value: 0.000214
```

The procedure

From the hypothesis matrix to the contrast matrix:

- Write down the hypotheses
- Extract the weights and write them into a **hypothesis matrix** as shown above
- Apply the **generalized matrix inverse** to the hypothesis matrix to create the contrast matrix
- Assign the contrast matrix (here, contr.sum(3)) to the factor and run the linear model

Repeated difference contrasts

```
fractions(contr.sdif(3))
```

```
##   2-1  3-2
## 1 -2/3 -1/3
## 2  1/3 -1/3
## 3  1/3  2/3
```

Repeated difference contrasts

Compares med to low, and high to low frequency.

```
## med vs low
table.word$M[2]-table.word$M[1]

## [1] -50

## high vs med
table.word$M[3]-table.word$M[2]

## [1] -50.25
```

Repeated difference contrasts

```
contrasts(simdat2$F) <- contr.sdif(3)
m1_mr <- lm(DV ~ F, data=simdat2)
summary(m1_mr)$coefficients

##             Estimate Std. Error t value   Pr(>|t|)
## (Intercept)   449.92     5.7965 77.6185 4.9491e-14
## F2-1          -50.00    14.1985 -3.5215 6.5001e-03
## F3-2          -50.25    14.1985 -3.5391 6.3229e-03
```

Two-by-two factorial designs

```
set.seed(1)
M <- matrix(c(10, 20, 10, 40), nrow=4, ncol=1, byrow=FALSE)
simdat4 <- mixedDesign(B=c(2,2), W=NULL,
                       n=5, M=M,  SD=10,
                       long = TRUE)
names(simdat4)[1:2] <- c("A","B")
```

Two-by-two factorial designs

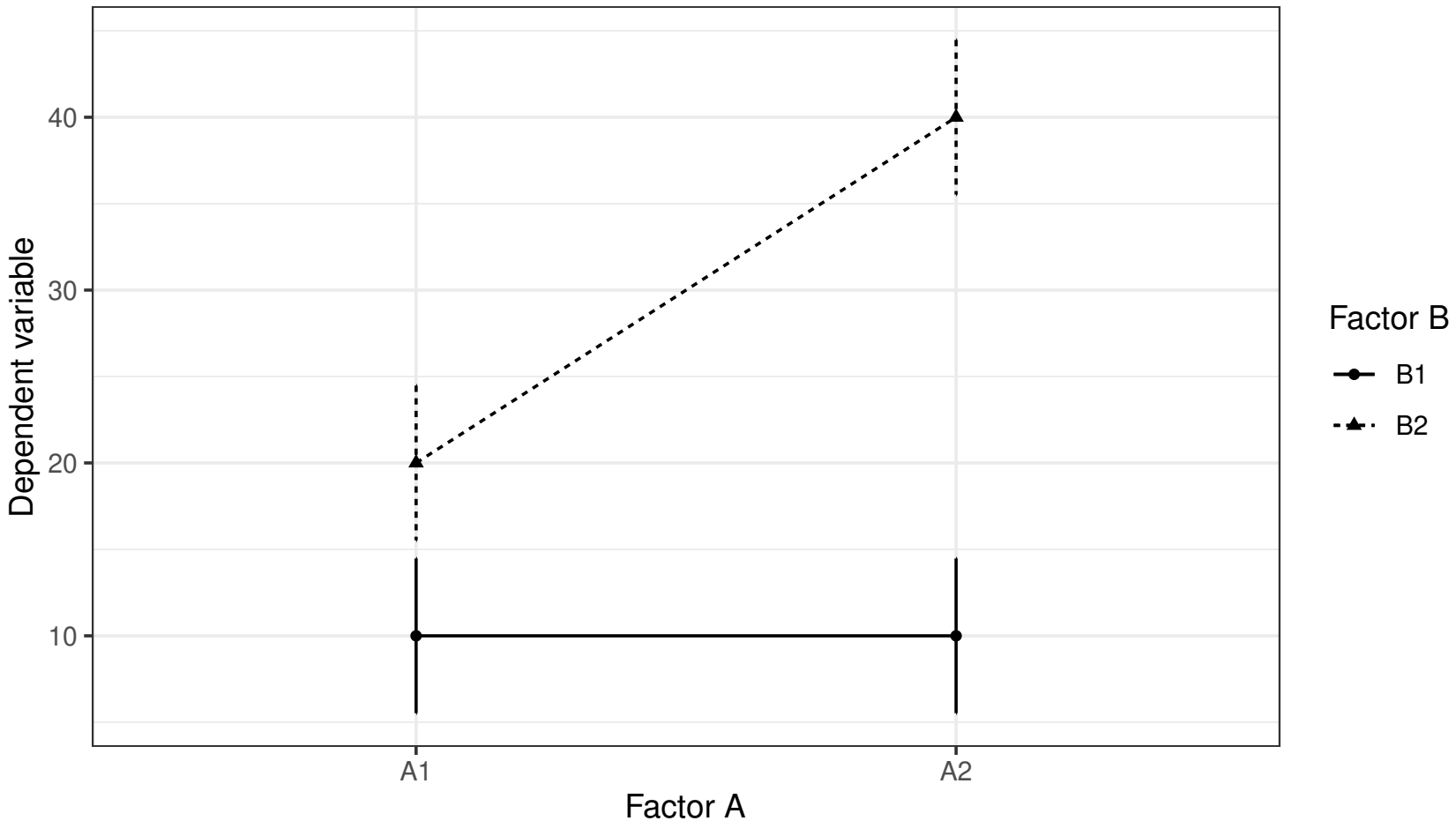
```
table4 <- simdat4 %>% group_by(A, B) %>% # plot interactio
    summarize(N=length(DV),
              M=mean(DV), SD=sd(DV), SE=SD/sqrt(N))
GM <-  mean(table4$M) # Grand Mean

table4a <- table4
names(table4a) <- c("Factor A","Factor B",
                    "N data","Means",
                    "Std. dev.","Std. errors")
```

Two-by-two factorial designs

```
table4a
```

```
## # A tibble: 4 x 6
## # Groups:   A [2]
##   `Factor A` `Factor B` `N data` Means `Std. dev.` `Std. errors`
##   <fct>      <fct>         <int> <dbl>       <dbl>          <dbl>
## 1 A1         B1                5    10        10             4.47
## 2 A1         B2                5    20        10             4.47
## 3 A2         B1                5    10        10.0           4.47
## 4 A2         B2                5    40        10             4.47
```

Two-by-two factorial designs



Two-by-two factorial designs

```
# ANOVA: B_A(2) times B_B(2)
m2_aov <- aov(DV ~ A*B + Error(id), data=simdat4)
summary(m2_aov)

##
## Error: id
##           Df Sum Sq Mean Sq F value  Pr(>F)
## A          1    500     500       5 0.03994
## B          1   2000    2000      20 0.00039
## A:B        1    500     500       5 0.03994
## Residuals 16   1600     100
```

Two-by-two factorial designs

```
# MR: B_A(2) times B_B(2)
m2_mr <- lm(DV ~ A*B, data=simdat4)
summary(m2_mr)$coefficients

##                Estimate Std. Error    t value Pr(>|t|)
## (Intercept) 1.0000e+01     4.4721 2.2361e+00 0.039945
## AA2         5.7595e-15     6.3246 9.1065e-16 1.000000
## BB2         1.0000e+01     6.3246 1.5811e+00 0.133410
## AA2:BB2     2.0000e+01     8.9443 2.2361e+00 0.039945
```

Two-by-two factorial designs

The discrepancy between ANOVA and lm arises from the default treatment contrast coding in R, which lm uses but aov does not:

```
contrasts(simdat4$A)

##    A2
## A1  0
## A2  1

contrasts(simdat4$B)

##    B2
## B1  0
## B2  1
```

Two-by-two factorial designs

ANOVA uses sum contrasts, and we can force lm to do this too:

```
# define sum contrasts:
contrasts(simdat4$A) <- contr.sum(2)
contrasts(simdat4$B) <- contr.sum(2)
m2_mr.sum <- lm(DV ~ A*B, data=simdat4)
summary(m2_mr.sum)$coefficients

##             Estimate Std. Error t value   Pr(>|t|)
## (Intercept)       20     2.2361  8.9443 1.2670e-07
## A1                -5     2.2361 -2.2361 3.9945e-02
## B1               -10     2.2361 -4.4721 3.8511e-04
## A1:B1              5     2.2361  2.2361 3.9945e-02
```

Two-by-two factorial designs

Read the hypr package vignette for examples on how to use it:

```
library(hypr)
vignette("hypr-intro", package = "hypr")
vignette("hypr-regression", package = "hypr")
```

This package allows you to easy go from the hypothesis to contrast matrix and back.

How I use contrast coding

- In my own work, I like experiment designs to be kept simple. $2 \times 2 \times 2$ is too complex for my taste.
- However, sometimes complex designs are unavoidable. In that case, I generally set up sum contrasts for ANOVA-style analyses, or nested contrasts (or both).
- Example of $2 \times 2 \times 2$ design:
  Lena A. Jäger, Daniela Mertzen, Julie A. Van Dyke, and Shravan Vasishth. Interference patterns in subject-verb agreement and reflexives revisited: A large-sample study. Journal of Memory and Language, 111, 2020. https://osf.io/reavs/

How I use contrast coding

- I never use the contr.sum() etc. functions. I define my own contrast coding columns in the design matrix.
- This becomes relevant in linear mixed models.

For example:

How I use contrast coding

```
persian<-read.table("data/Persiane1crit.txt")
xtabs(~dist+distance,persian)

##      distance
## dist long short
##   -1    0   756
##   1   756     0

xtabs(~pred+predability,persian)

##      predability
## pred predictable unpredictable
##   -1         756             0
##   1            0           756
```

How I use contrast coding

```
contrasts(persian$distance)<-contr.sum(2)
contrasts(persian$predability)<-contr.sum(2)
m<-lmer(rt~distance*predability+(1+distance*predability||su
          (1+distance*predability||item),persian)

## boundary (singular) fit:  see ?isSingular
```

[Ignore the singularity warning for now]

How I use contrast coding

Correlations should not have been computed:

```
summary(m)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## rt ~ distance * predability + ((1 | subj) + (0 + distance | subj) +
##     (0 + predability | subj) + (0 + distance:predability | subj)) +
##     ((1 | item) + (0 + distance | item) + (0 + predability |
##         item) + (0 + distance:predability | item))
##    Data: persian
##
## REML criterion at convergence: 21880
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -2.104 -0.454 -0.175  0.160 11.210
##
## Random effects:
##  Groups   Name                                  Variance Std.Dev. Corr
##  subj     (Intercept)                           4.67e+00   2.16
##  subj.1   distancelong                          1.01e+04 100.29
##           distanceshort                         6.49e+03  80.59   1.00
##  subj.2   predabilitypredictable                1.24e+04 111.47
##           predabilityunpredictable              2.17e+04 147.45   1.00
##  subj.3   distancelong:predabilitypredictable   2.43e+03  49.34
##           distanceshort:predabilitypredictable  2.48e+03  49.81   0.83
##           distancelong:predabilityunpredictable 5.55e+02  23.56   1.00
##           distanceshort:predabilityunpredictable 1.58e+04 125.57   0.50
##  item     (Intercept)                           6.23e-02   0.25
```

How I use contrast coding

Using my own contrast coding vectors:

```
head(persian)

##      subj item   rt dist distance pred   predability
## 60      4    6  568   -1    short   -1   predictable
## 94      4   17  517    1     long    1 unpredictable
## 146     4   22  675   -1    short   -1   predictable
## 185     4    5  575    1     long    1 unpredictable
## 215     4    3  581    1     long   -1   predictable
## 285     4    7 1171    1     long   -1   predictable

m2<-lmer(rt~dist*pred+(1+dist*pred||subj)+
           (1+dist*pred||item),persian)

## boundary (singular) fit:  see ?isSingular
```

How I use contrast coding

Correlations are not computed, as expected:

```
summary(m2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rt ~ dist * pred + ((1 | subj) + (0 + dist | subj) + (0 + pred |
##     subj) + (0 + dist:pred | subj)) + ((1 | item) + (0 + dist |
##     item) + (0 + pred | item) + (0 + dist:pred | item))
##    Data: persian
##
## REML criterion at convergence: 21897
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -1.884 -0.452 -0.175  0.159 11.018
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subj     (Intercept) 2.79e+04 167.15
##  subj.1   dist        7.81e+02  27.94
##  subj.2   pred        5.51e+02  23.48
##  subj.3   dist:pred   2.50e-03   0.05
##  item     (Intercept) 1.97e+03  44.41
##  item.1   dist        0.00e+00   0.00
##  item.2   pred        6.20e+02  24.90
##  item.3   dist:pred   0.00e+00   0.00
##  Residual             1.06e+05 325.40
## Number of obs: 1512, groups:  subj, 42; item, 36
##
```

How I use contrast coding

- I prefer to define my own contrasts by hand because then I don't get spurious correlations in the random effects, as illustrated above.
- As far as I know, it's not harmful to have those correlations. So maybe this is just a minor technical point.

To learn more

Work through:
Daniel J. Schad, Shravan Vasishth, Sven Hohenstein, and Reinhold Kliegl. How to capitalize on a priori contrasts in linear (mixed) models: A tutorial. Journal of Memory and Language, 110, 2020.

- This is a long and detailed paper, and should be read patiently, working through the examples.
- To fully understand contrasts, you need to become familiar with the matrix formulation of the linear model (some passive matrix algebra knowledge is needed).