handout7.R

spinoza

2021-12-08

```
#We begin by calling the data for handout 7
library("readxl")

h7.data = read_excel("handout7data.xlsx")
```

```
## New names:
## * `` -> ...17
## * `` -> ...19
```

```
str(h7.data)
```

```
## tibble [24 x 20] (S3: tbl_df/tbl/data.frame)
##  $ tip1      : num [1:24] 7 3 3 4 8 3 2 9 5 4 ...
##  $ tip2      : num [1:24] 6 3 5 3 8 2 4 9 4 5 ...
##  $ t         : num [1:24] 1 1 1 1 1 1 1 1 1 1 ...
##  $ s         : num [1:24] 1 2 3 4 5 6 7 8 9 10 ...
##  $ h         : num [1:24] 7 3 3 4 8 3 2 9 5 4 ...
##  $ pressure  : num [1:24] 1 1 1 1 1 1 2 2 2 2 ...
##  $ batch     : num [1:24] 1 2 3 4 5 6 1 2 3 4 ...
##  $ yield     : num [1:24] 90.3 89.2 98.2 93.9 87.4 97.9 92.5 89.5 90.6 94.7 ...
##  $ operator 1: num [1:24] 4.85 4.93 4.75 4.77 4.67 4.87 4.67 4.94 4.85 4.75 ...
##  $ operator 2: num [1:24] 5.09 5.04 4.95 5.02 4.9 5.05 4.9 5.15 5.08 4.98 ...
##  $ o         : num [1:24] 1 1 1 1 1 1 1 1 1 1 ...
##  $ fuse      : num [1:24] 1 2 3 4 5 6 7 8 9 10 ...
##  $ time      : num [1:24] 4.85 4.93 4.75 4.77 4.67 4.87 4.67 4.94 4.85 4.75 ...
##  $ tip       : num [1:24] 1 1 1 1 2 2 2 2 3 3 ...
##  $ specimen  : num [1:24] 1 2 3 4 1 2 3 4 1 2 ...
##  $ hardness  : num [1:24] 49.3 49.4 49.6 50 49.4 49.3 49.8 49.9 49.2 49.4 ...
##  $ ...17     : logi [1:24] NA NA NA NA NA NA ...
##  $ o1        : num [1:24] 4.85 4.93 4.75 4.77 4.67 4.87 4.67 4.94 4.85 4.75 ...
##  $ ...19     : logi [1:24] NA NA NA NA NA NA ...
##  $ o2        : num [1:24] 5.09 5.04 4.95 5.02 4.9 5.05 4.9 5.15 5.08 4.98 ...
```

```
#We will need these packages to work with blocks as random effects
library("lme4")
```

```
## Loading required package: Matrix
```

```
library("lmerTest")
```

```
##
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':
##
##     lmer
## The following object is masked from 'package:stats':
##
##     step
```

```
#We will need this package for performing Fisher comparisons and groupings
library("multcomp")
```

```
## Loading required package: mvtnorm

## Loading required package: survival

## Loading required package: TH.data

## Loading required package: MASS

##
## Attaching package: 'TH.data'

## The following object is masked from 'package:MASS':
##
##     geyser
```

```
#Example 7.1
#An experiment is conducted to compare two tips used on a hardness testing machine
#The experimental units are the metal specimens. Each tip gives a hardness measurement for each specime
#Thus, the data is from a paired comparisons design
tip1 = na.omit(h7.data$tip1)
tip2 = na.omit(h7.data$tip2)

#The built-in function t.test can be used to compute a paired comparisons analysis.
t.test(tip1,tip2,paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  tip1 and tip2
## t = -0.26414, df = 9, p-value = 0.7976
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.9564389  0.7564389
## sample estimates:
## mean of the differences
##                    -0.1
```

```
#Below is code for our own function to compute a paired comparisons analysis.
#Note that the data must be unstacked. (Each group has its own column of measurements.)
paired.test = function(y1,y2,alpha=.05)
{
  d = y1 - y2

  n = length(d)
  d.bar = mean(d)
  s.d = sd(d)
  SE = s.d/sqrt(n)

  t.0 = d.bar / SE
  p.value = 2*pt(abs(t.0),df=n-1,lower.tail = FALSE)

  t.mult = qt(alpha/2,lower.tail = FALSE, df=n-1)
  lower.est = d.bar - t.mult*SE
  upper.est = d.bar + t.mult*SE

  table1 = matrix(c(n,d.bar,s.d),nrow = 1)
  dimnames(table1) = list(c(""),c("sample.size","mean.diff","sd.diff"))
  print(table1)

  table2 = matrix(c(t.0,p.value),nrow = 1)
  dimnames(table2) = list(c(""),c("test statistic","p-value"))
  print(table2)

  table3 = matrix(c(d.bar,lower.est,upper.est),nrow = 1)
  dimnames(table3) = list(c(""),c("estimated difference","lower limit","upper limit"))
  print(table3,digits = 3)

}

paired.test(tip1,tip2)
```

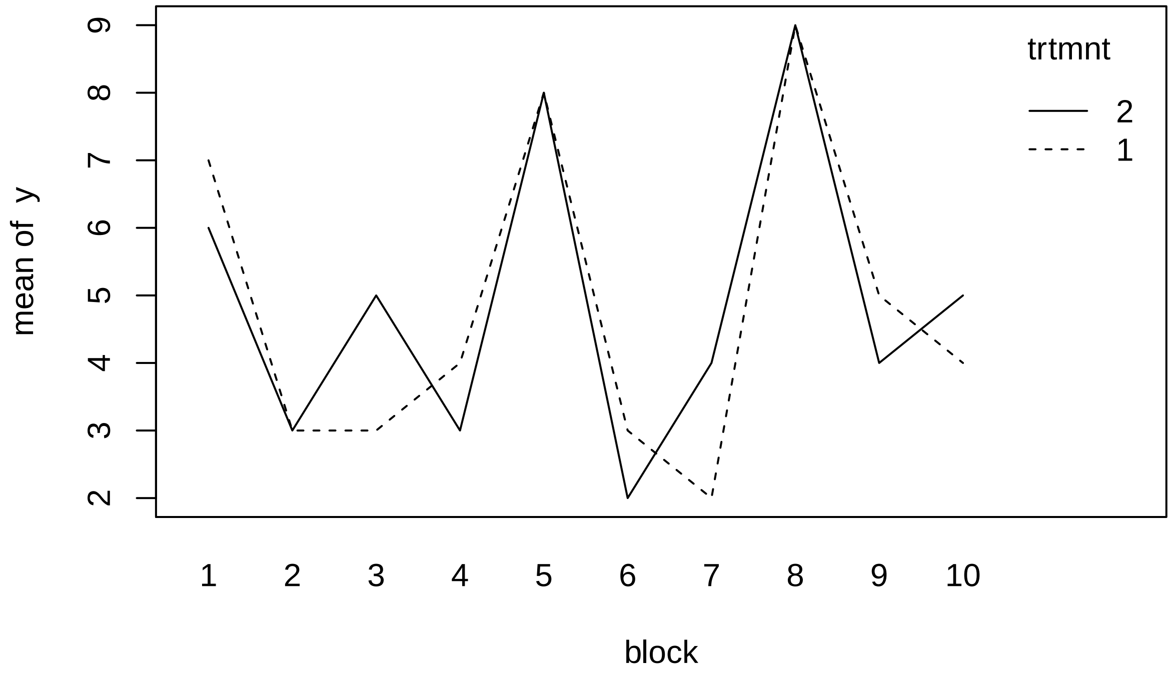
```
##  sample.size mean.diff  sd.diff
##           10      -0.1 1.197219
##  test statistic   p-value
##      -0.2641353 0.7976245
##  estimated difference lower limit upper limit
##                  -0.1      -0.956       0.756
```

```
#The variables below define the same data, only with responses stacked in one column,
#identified by treatment (machine tip) and block (specimen).
trtmnt = as.factor(na.omit(h7.data$t))
block = as.factor(na.omit(h7.data$s))
y = na.omit(h7.data$h)

#We could use aov to compute F for a randomized block design instead of computing t for a paired compar
#Note that t^2=F, so the p-values are the same.
rcbd.mod = aov(y ~ block + trtmnt)
summary(rcbd.mod)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## block        9  90.05  10.006   13.96 0.000281 ***
## trtmnt       1   0.05   0.050    0.07 0.797625    
## Residuals    9   6.45   0.717                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

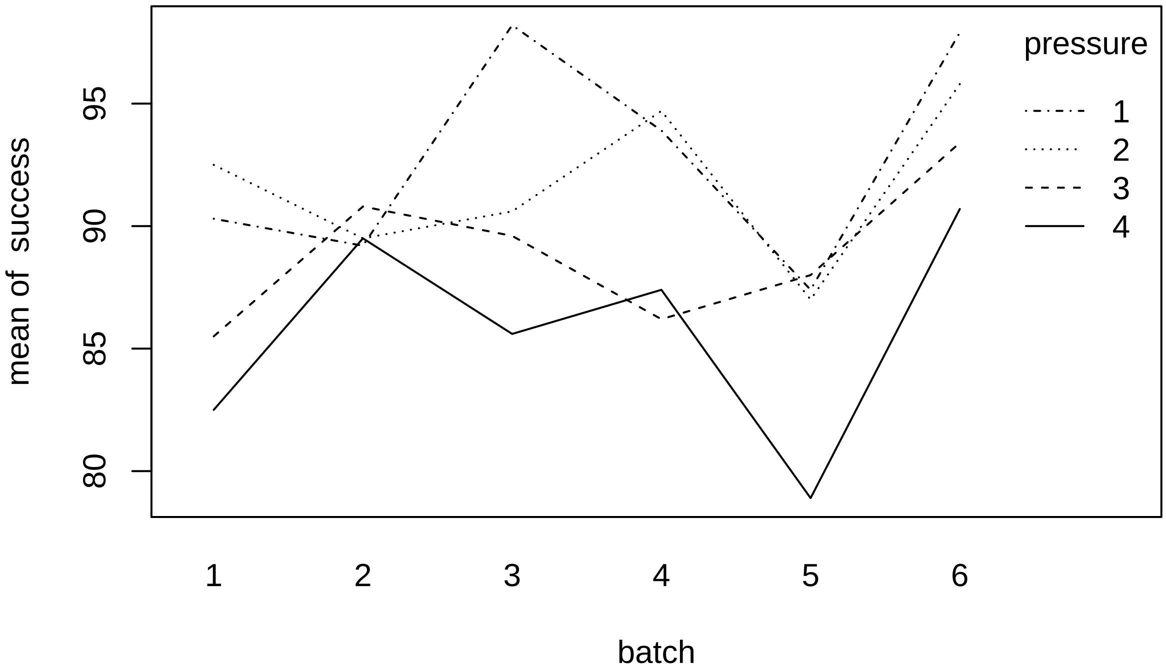
```
interaction.plot(block,trtmnt,y)
```



```
#Example 5.2
#A medical device manufacturer produces vascular grafts (artificial veins).
#An experiment is planned to investigate the effect of extrusion pressure on yield (proportion of accep
#The resin used in the production is from an external supplier and may differ from batch to batch.

#Each batch is tested at each of the extrusion pressures.
#The batches are randomly selected, and have no identifiable features that can be used for modeling pur
pressure = as.factor(na.omit(h7.data$pressure))
batch = as.factor(na.omit(h7.data$batch))
success = na.omit(h7.data$yield)

#Graph the data.
#Remember that we are testing whether pressure differences are generalizable to a larger population of
#Thus, we are testing how the pressure effect depends on the batch.
interaction.plot(batch,pressure,success)
```



```
#Use contrasts to define parameter restrictions for the fixed effect in the model
contrasts(pressure)=contr.sum

#This is how the lme4 package defines a model with random effects.
#The 1 in front of batch signifies that batch levels are randomly selected from a common distribution.
random.mod = lmer(success ~ (1|batch) + pressure)

#The anova command is used to compute the test for fixed effects.
anova(random.mod)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF DenDF F value   Pr(>F)
## pressure 178.17   59.39     3    15  8.1071 0.001916 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

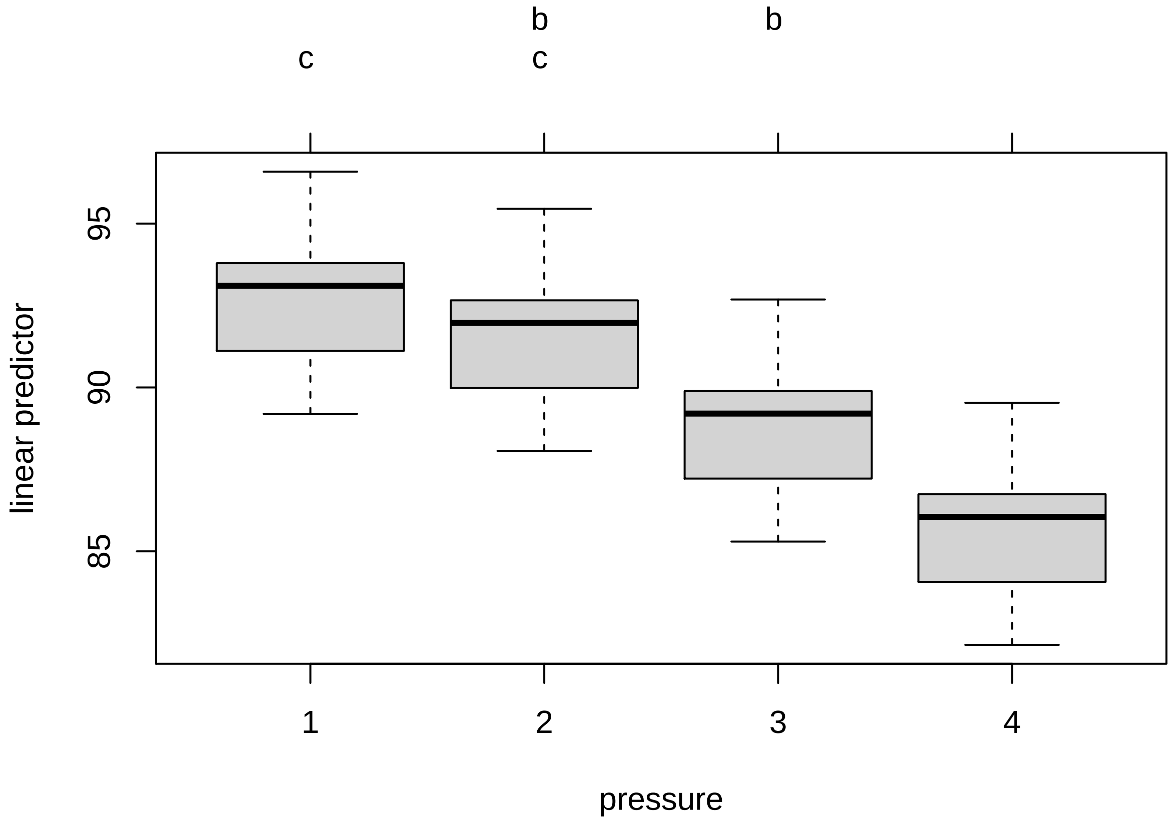
```
#We can perform pairwise comparisons, here using the Fisher LSD method
comps = glht(random.mod,linfct = mcp(pressure="Tukey"))
summary(comps,test=univariate())
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lmer(formula = success ~ (1 | batch) + pressure)
##
## Linear Hypotheses:
##            Estimate Std. Error z value Pr(>|z|)
## 2 - 1 == 0   -1.133      1.563  -0.725 0.468294
## 3 - 1 == 0   -3.900      1.563  -2.496 0.012570 *
## 4 - 1 == 0   -7.050      1.563  -4.512 6.44e-06 ***
## 3 - 2 == 0   -2.767      1.563  -1.770 0.076647 .
## 4 - 2 == 0   -5.917      1.563  -3.786 0.000153 ***
## 4 - 3 == 0   -3.150      1.563  -2.016 0.043822 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Univariate p values reported)
```

```
cld(summary(comps,test=univariate()))
```

```
##    1    2    3    4
##  "c" "bc"  "b"  "a"
```

```
plot(cld(summary(comps,test=univariate())))
```



```
#The following code is used for computing the variance components estimates, and fixed effect parameter
summary(random.mod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: success ~ (1 | batch) + pressure
##
## REML criterion at convergence: 114.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.32253 -0.64269  0.01068  0.54202  1.62882
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  batch    (Intercept) 7.781    2.789
##  Residual             7.326    2.707
## Number of obs: 24, groups:  batch, 6
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept)  89.7958     1.2657  5.0000  70.943 1.05e-08 ***
## pressure1     3.0208     0.9569 15.0000   3.157  0.00652 ** 
## pressure2     1.8875     0.9569 15.0000   1.972  0.06728 .  
## pressure3    -0.8792     0.9569 15.0000  -0.919  0.37277    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) prssr1 prssr2
## pressure1  0.000              
## pressure2  0.000 -0.333       
## pressure3  0.000 -0.333 -0.333
```

```
#Since the fixed effect estimates must sum to 0, the estimate at level a=4 is the negative of the sum of
#fixed effect estimates at levels 1 through a-1. (The first parameter estimate is for the overall mean.)
estimates = summary(random.mod)
estimates.pressure = c(estimates$coefficients[1:4,1],0-sum(estimates$coefficients[2:4,1]))
estimates.pressure
```

```
## (Intercept)   pressure1   pressure2   pressure3             
##  89.7958333   3.0208333   1.8875000  -0.8791667  -4.0291667
```