

# uebung8.r

Janina

Sat Jun 24 23:45:16 2017

```
# Uebungsblatt 8
# Namen: Janina Schoenberger, Benjamin Weigner
# Tutorin: Gergana Stanilova
# Uebung: Mi 12-14 Uhr

# Aufgabe 19

# A
# Einfaktorielle ANOVA fuer primer und applic

paint = data.frame(adhf = c(4.0,4.5,4.3,5.6,4.9,5.4,3.8,3.7,
                           4.0,5.4,4.9,5.6,5.8,6.1,6.3,5.5,5.0,5.0),
                   primer = factor(rep(rep(1:3,rep(3,3)),2)),
                   applic = factor(rep(c("D","S"),c(9,9))))

head(paint)
```

```
##      adhf primer applic
## 1  4.0      1      D
## 2  4.5      1      D
## 3  4.3      1      D
## 4  5.6      2      D
## 5  4.9      2      D
## 6  5.4      2      D
```

```
paint.primer <- aov(adhf ~ primer, data=paint)
paint.applic <- aov(adhf ~ applic, data=paint)
summary(paint.primer)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## primer          2  4.581   2.2906   5.599 0.0153 *
## Residuals     15  6.137   0.4091
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(paint.applic)
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)
## applic          1  4.909   4.909   13.52 0.00204 **
## Residuals     16  5.809   0.363
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

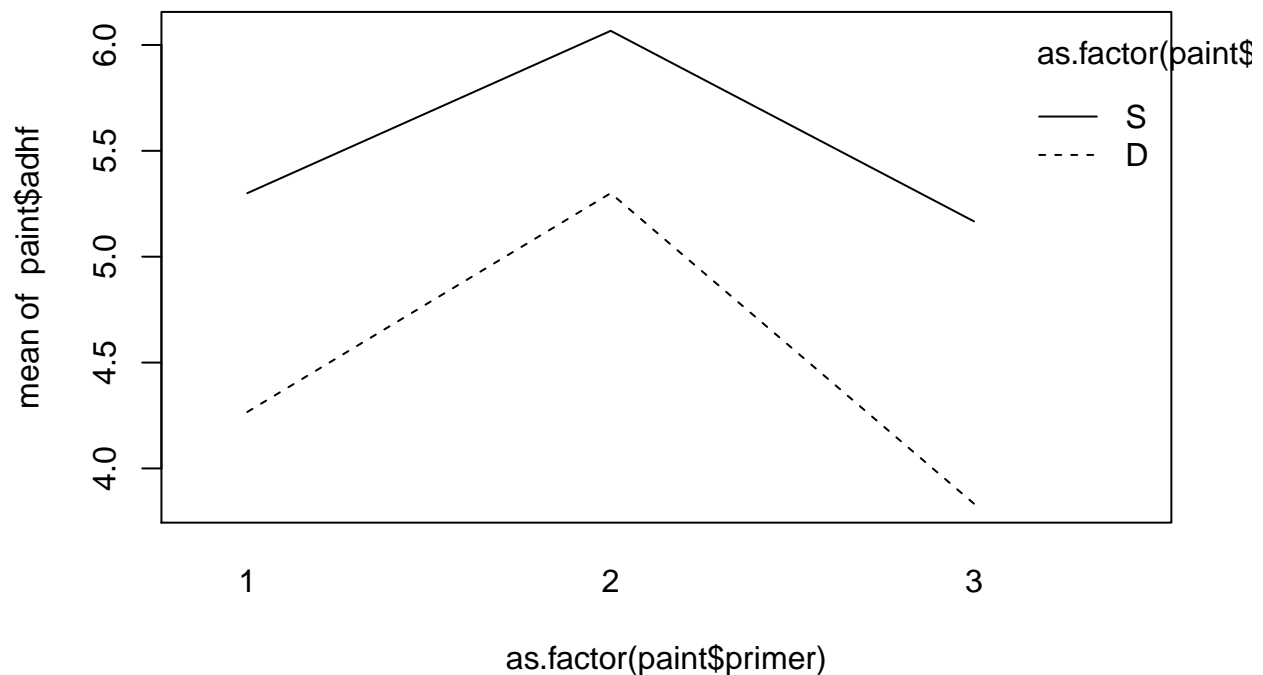
```
# Interpretation
# Die einfaktorielle Varianzanalyse (mit ANOVA) gibt an, ob es einen signifikanten
# Unterschied im Mittelwert von verschiedenen Gruppen gegenueber der abhaengigen Variable
```

```

# gibt.
# Dieser Unterschied ist signifikant, falls der p-Wert des F-Tests unter 0.05 liegt.
# adhf ist die abhaengige Variable.
# Mit primer als unabhaengiger Variable erhaelt man einen p-Wert von 0.0153. Dieser ist signifikant.
# Mit applic als unabhaengiger Variable erhaelt man einen p-Wert von 0.00204. Dieser ist
# signifikant und zeigt einen deutlicheren Unterschied, als bei primer.

# B
# Grafisch: Interaktion zwischen primer und applic im Bezug auf den Endpunkt adhf
if (FALSE){
  # Die Funktionen ggline und ggboxplot haben nicht funktioniert
  library(ggplot2)
  library(gplots)
  ggboxplot(paint,primer,adhf)
  ggline(paint,primer,adhf)
}
interaction.plot(as.factor(paint$primer),as.factor(paint$applic),paint$adhf)

```



```

interaction.plot(as.factor(paint$applic),as.factor(paint$primer),paint$adhf)
# Sowohl der Faktor primer, als auch der Faktor applic scheinen einen Einfluss auf
# die Adhesion Force (paint$adhf) zu haben.
# 1. Plot: Sowohl bei D, als auch bei S Kurve (D/S von applic) ist Ausschlag bei Primer 2
# zu erkennen
# -> primer hat Einfluss auf adhf

```

```

# 2. Plot: Kurven von allen 3 Primer sind bei S hoeher als bei D
# -> applic hat Einfluss auf adhf
# Deshalb fuehren wir eine zweifaktorielle Anova durch.
fita <- aov(adhf~primer*applic, data=paint)
summary(fita)

```

```

##              Df Sum Sq Mean Sq F value    Pr(>F)
## primer         2  4.581    2.291   27.858 3.10e-05 ***
## applic         1  4.909    4.909   59.703 5.36e-06 ***
## primer:applic   2  0.241    0.121    1.466    0.269
## Residuals     12  0.987    0.082
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

# Vergleich zu 19 A: Die zweifaktorielle Anova hat hoehere F-Werte und daher
# noch niedrigere P-Werte. Das Ergebnis ist also mit noch geringerem P-Wert
# signifikant, als die einfaktorielle Anova.

```

```

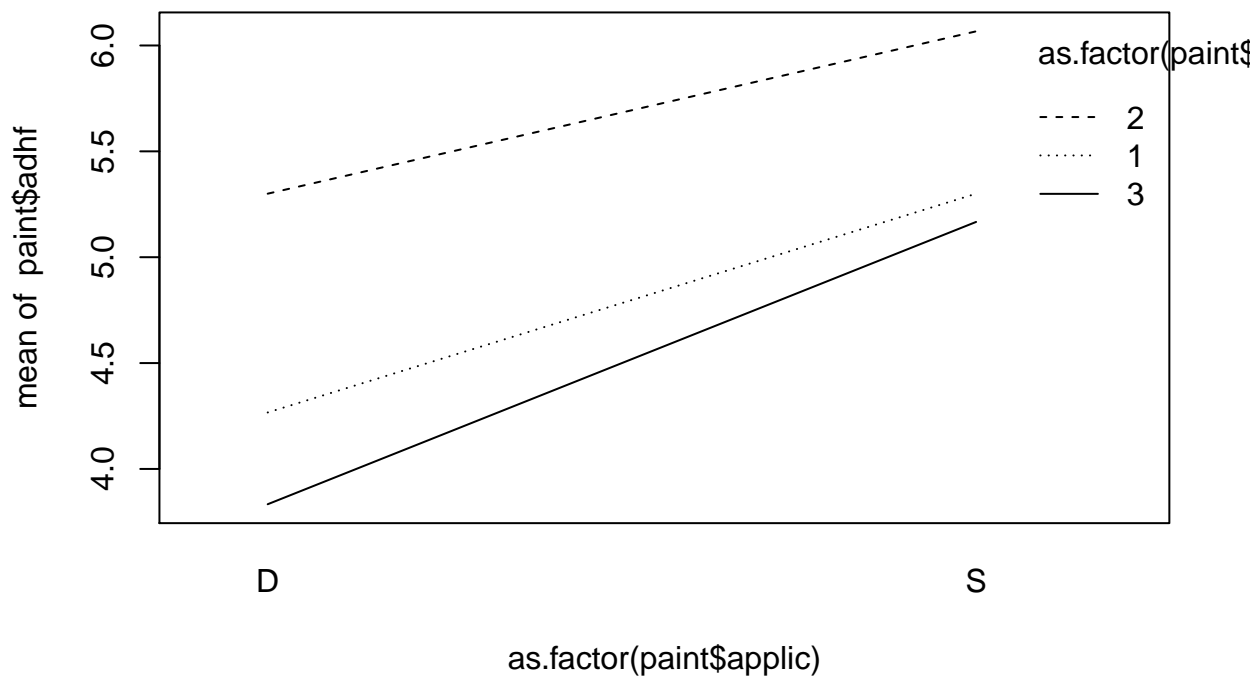
# C
# Kontrasttest = paarweise t-Tests (Vorschalttest)
# install.packages("lsmeans") und ("multcomp")
library(lsmeans)

```

```
## Warning: package 'lsmeans' was built under R version 3.3.3
```

```
## Loading required package: estimability
```

```
## Warning: package 'estimability' was built under R version 3.3.2
```



```
library(multcomp)
```

```
## Warning: package 'multcomp' was built under R version 3.3.3
```

```
## Loading required package: mvtnorm
```

```
## Warning: package 'mvtnorm' was built under R version 3.3.2
```

```
## Loading required package: survival
```

```
## Warning: package 'survival' was built under R version 3.3.3
```

```
## Loading required package: TH.data
```

```
## Warning: package 'TH.data' was built under R version 3.3.3
```

```
## Loading required package: MASS
```

```
## Warning: package 'MASS' was built under R version 3.3.3
```

```
##
```

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

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

```
if (FALSE){
  # stackoverflow Bsp aus Hinweisen
  data(obk.long)
  head(obk.long)
  fit <- aov_car(value~treatment*gender + Error(id),data=obk.long, return = "aov")
  (ref1 <- lsmeans(fit, c("treatment", "gender"))))
  c_list <- list(c1 = c(0, -1, 1, 0, 0, 0),
                c2 = c(0, -0.5, 0.5, 0, -0.5, 0.5))
  summary(contrast(ref1, c_list), adjust = "holm")
  pairs(lsmeans(fit, "treatment"), adjust = "none")
}

ref1 <- lsmeans(fit,c("applic","primer"))
ref1
```

```
## applic primer    lsmean      SE df lower.CL upper.CL
## D      1      4.266667 0.1655518 12 3.905960 4.627373
## S      1      5.300000 0.1655518 12 4.939294 5.660706
## D      2      5.300000 0.1655518 12 4.939294 5.660706
## S      2      6.066667 0.1655518 12 5.705960 6.427373
## D      3      3.833333 0.1655518 12 3.472627 4.194040
## S      3      5.166667 0.1655518 12 4.805960 5.527373
##
## Confidence level used: 0.95
```

```
# c1: primer 2 gegen 1,3 unter applic D
c1 <- c(-0.25,0,0.5,0,-0.25,0)
# c2: applic D gegen S fuer alle primer
c2 <- c(1/6,-1/6,1/6,-1/6,1/6,-1/6)
c_list <- list(c1,c2)
summary(contrast(ref1,c_list,adjust="none")) ##### welches mit tukey?
```

```
## contrast
## c(-0.25, 0, 0.5, 0, -0.25, 0)
## c(0.166666666666667, -0.166666666666667, 0.166666666666667, -0.166666666666667,
## estimate      SE df t.ratio p.value
## 0.6250000 0.10137938 12 6.165 <.0001
## -0.5222222 0.06758625 12 -7.727 <.0001
```

```
pairs(ref1, adjust="none")
```

```
## contrast      estimate      SE df t.ratio p.value
## D,1 - S,1 -1.033333e+00 0.2341256 12 -4.414 0.0008
## D,1 - D,2 -1.033333e+00 0.2341256 12 -4.414 0.0008
## D,1 - S,2 -1.800000e+00 0.2341256 12 -7.688 <.0001
## D,1 - D,3 4.333333e-01 0.2341256 12 1.851 0.0889
## D,1 - S,3 -9.000000e-01 0.2341256 12 -3.844 0.0023
## S,1 - D,2 -4.440892e-16 0.2341256 12 0.000 1.0000
```

```
## S,1 - S,2 -7.666667e-01 0.2341256 12 -3.275 0.0066
## S,1 - D,3 1.466667e+00 0.2341256 12 6.264 <.0001
## S,1 - S,3 1.333333e-01 0.2341256 12 0.569 0.5795
## D,2 - S,2 -7.666667e-01 0.2341256 12 -3.275 0.0066
## D,2 - D,3 1.466667e+00 0.2341256 12 6.264 <.0001
## D,2 - S,3 1.333333e-01 0.2341256 12 0.569 0.5795
## S,2 - D,3 2.233333e+00 0.2341256 12 9.539 <.0001
## S,2 - S,3 9.000000e-01 0.2341256 12 3.844 0.0023
## D,3 - S,3 -1.333333e+00 0.2341256 12 -5.695 0.0001
```

```
pairs(ref1, adjust="tukey")
```

```
## contrast estimate SE df t.ratio p.value
## D,1 - S,1 -1.033333e+00 0.2341256 12 -4.414 0.0085
## D,1 - D,2 -1.033333e+00 0.2341256 12 -4.414 0.0085
## D,1 - S,2 -1.800000e+00 0.2341256 12 -7.688 0.0001
## D,1 - D,3 4.333333e-01 0.2341256 12 1.851 0.4724
## D,1 - S,3 -9.000000e-01 0.2341256 12 -3.844 0.0221
## S,1 - D,2 -4.440892e-16 0.2341256 12 0.000 1.0000
## S,1 - S,2 -7.666667e-01 0.2341256 12 -3.275 0.0576
## S,1 - D,3 1.466667e+00 0.2341256 12 6.264 0.0005
## S,1 - S,3 1.333333e-01 0.2341256 12 0.569 0.9913
## D,2 - S,2 -7.666667e-01 0.2341256 12 -3.275 0.0576
## D,2 - D,3 1.466667e+00 0.2341256 12 6.264 0.0005
## D,2 - S,3 1.333333e-01 0.2341256 12 0.569 0.9913
## S,2 - D,3 2.233333e+00 0.2341256 12 9.539 <.0001
## S,2 - S,3 9.000000e-01 0.2341256 12 3.844 0.0221
## D,3 - S,3 -1.333333e+00 0.2341256 12 -5.695 0.0011
##
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```
# Interpretation:
# Tukey kontrolliert alle  $h=k(k-1)/2$  Nullhypothesen und beruecksichtigt die Korrelation
# zwischen den Gruppen. Bei abhaengigen Vergleichen besser als Bonferroni
# Die p-Werte mit Tukey Adjustierung sind deutlich hoeher. Da wir wissen, dass die
# Variablen korrelieren, macht das Sinn.
```

```
# D
# Vergleich Haupteffekt von primer ohne Adjustierung mit Bonferroni und Bonferroni Holm
pairs(lsmmeans(fita,"primer"),adjust="none")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
## contrast estimate SE df t.ratio p.value
## 1 - 2 -0.9000000 0.1655518 12 -5.436 0.0002
## 1 - 3 0.2833333 0.1655518 12 1.711 0.1127
## 2 - 3 1.1833333 0.1655518 12 7.148 <.0001
##
## Results are averaged over the levels of: applic
```

```
pairs(lsmmeans(fita,"primer"),adjust="bonferroni")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
## contrast estimate SE df t.ratio p.value
## 1 - 2 -0.9000000 0.1655518 12 -5.436 0.0005
## 1 - 3 0.2833333 0.1655518 12 1.711 0.3381
## 2 - 3 1.1833333 0.1655518 12 7.148 <.0001
##
## Results are averaged over the levels of: applic
## P value adjustment: bonferroni method for 3 tests
```

```
pairs(lsmmeans(fita,"primer"),adjust="holm")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
## contrast estimate SE df t.ratio p.value
## 1 - 2 -0.9000000 0.1655518 12 -5.436 0.0003
## 1 - 3 0.2833333 0.1655518 12 1.711 0.1127
## 2 - 3 1.1833333 0.1655518 12 7.148 <.0001
##
## Results are averaged over the levels of: applic
## P value adjustment: holm method for 3 tests
```

```
# Es faellt auf, dass die t-Ratios alle gleich sind, waehrend sich die p-Werte
# durch die Adjustierungsmethoden unterscheiden.
# Die p-Werte von Holm und Ohne Adjustierung sind gleich,
# waehrend die Bonferroni Adjustierung hoehere p-Werte vergiebt,
# sodass die Ergebnisse schneller nicht signifikant werden.
```