W7 practice

2023-02-23

1. rats data

```
library(haven); library(psych); library(dplyr);
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(magrittr); library(ggplot2); library(gridExtra)
```

```
##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##     %+%, alpha

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(rstatix); library(multcomp)
```

```
##
## Attaching package: 'rstatix'

## The following object is masked from 'package:stats':
##
##     filter
## Loading required package: mvtnorm

## Loading required package: survival

## Loading required package: TH.data

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:rstatix':
##
##     select

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

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

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

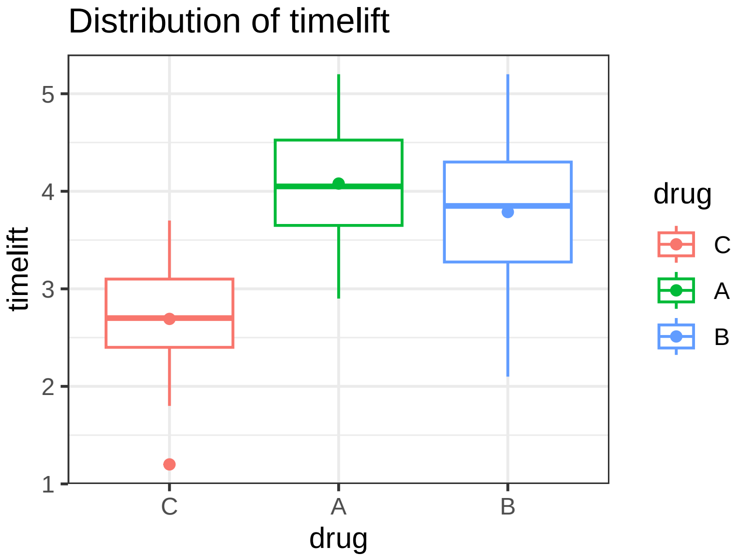
```
  rats =
    data.frame(rat_id = c(1:72),
               drug = c(rep(c("B", "A", "C"), times = 24)),
               timelift = c(c(5.2, 5.1, 2.4, 4.3, 3.1, 2.4,
                              4.2, 5.0, 2.4, 2.4, 4.6, 1.2,
                              5.1, 5.2, 2.9, 3.3, 4.5, 3.2,
                              4.6, 5.0, 2.8, 2.9, 4.1, 2.2,
                              3.8, 4.1, 2.4, 3.9, 3.8, 2.2,
                              4.7, 4.5, 2.6, 3.9, 3.1, 3.4,
                              3.4, 4.5, 3.6, 3.9, 4.2, 2.0,
                              4.3, 3.5, 2.6, 2.8, 2.9, 3.1,
                              5.0, 4.0, 1.8, 3.5, 3.8, 2.9,
                              2.1, 3.1, 2.9, 4.2, 3.7, 3.4,
                              3.7, 3.9, 3.7, 3.6, 3.9, 2.6,
                              3.2, 3.2, 2.8, 2.9, 5.1, 3.1)))
```

1-1. rats ANOVA

```
# making drug C as reference group
  rats =
    rats %>% mutate(drug = factor(rats$drug, levels = c("C", "A", "B")))

# boxplot
```

```
rats %>%
  ggplot(aes(drug, timelift, color = drug, group = drug)) +
  geom_boxplot() +
  stat_summary(fun = mean, geom="point") +
  theme_bw() +
  ggtitle("Distribution of timelift")
```



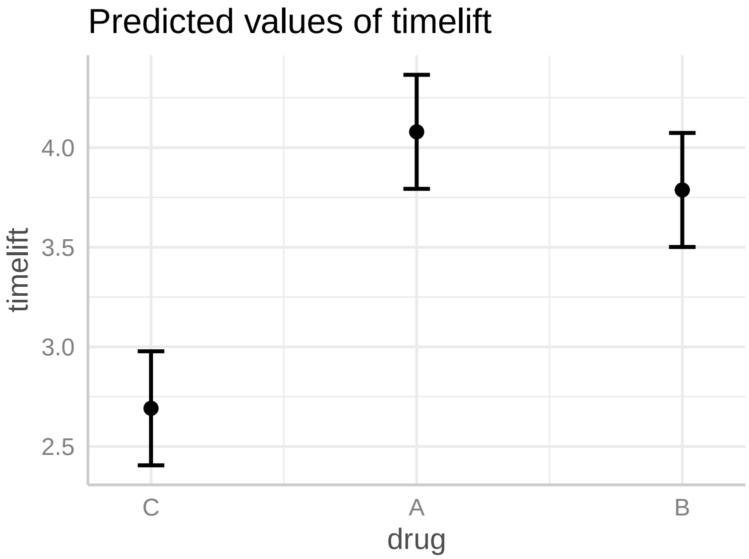
```
# ANOVA
  fit1 = lm(timelift ~ drug, data = rats)
  summary(fit1)
```

```
##
## Call:
## lm(formula = timelift ~ drug, data = rats)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.68750 -0.41250  0.01667  0.44271  1.41250
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.6917     0.1460  18.435  < 2e-16 ***
## drugA         1.3875     0.2065   6.719 4.27e-09 ***
## drugB         1.0958     0.2065   5.307 1.28e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7153 on 69 degrees of freedom
## Multiple R-squared:  0.4212, Adjusted R-squared:  0.4044
## F-statistic:  25.1 on 2 and 69 DF,  p-value: 6.425e-09
```

```
anova(fit1)
```

```
## Analysis of Variance Table
##
## Response: timelift
##           Df Sum Sq Mean Sq F value    Pr(>F)
## drug       2 25.689 12.8443  25.104 6.425e-09 ***
## Residuals 69 35.304  0.5117
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(ggeffects)
ggpred = ggpredict(fit1, "drug") # get mean and CIs
plot(ggpred)
```



2. rats2

```
rats2 =
  data.frame(rat_id = c(1:72),
             sex = c( "F", "F", "F", "M", "F", "F", "F", "F", "M", "F", "M", "F", "F", "F", "M", "F",
             drug = c(rep(c("B", "A", "C"), times = 24)),
             timelift = c(c(5.2, 5.1, 2.4, 4.3, 3.1, 2.4,
                            4.2, 5.0, 2.4, 2.4, 4.6, 1.2,
                            5.1, 5.2, 2.9, 3.3, 4.5, 3.2,
                            4.6, 5.0, 2.8, 2.9, 4.1, 2.2,
                            3.8, 4.1, 2.4, 3.9, 3.8, 2.2,
                            4.7, 4.5, 2.6, 3.9, 3.1, 3.4,
                            3.4, 4.5, 3.6, 3.9, 4.2, 2.0,
                            4.3, 3.5, 2.6, 2.8, 2.9, 3.1,
                            5.0, 4.0, 1.8, 3.5, 3.8, 2.9,
                            2.1, 3.1, 2.9, 4.2, 3.7, 3.4,
                            3.7, 3.9, 3.7, 3.6, 3.9, 2.6,
                            3.2, 3.2, 2.8, 2.9, 5.1, 3.1)))

 rats2 =
   rats2 %>% mutate(drug = factor(rats2$drug, levels = c("C", "A", "B")))
```

2-1. two-way ANOVA

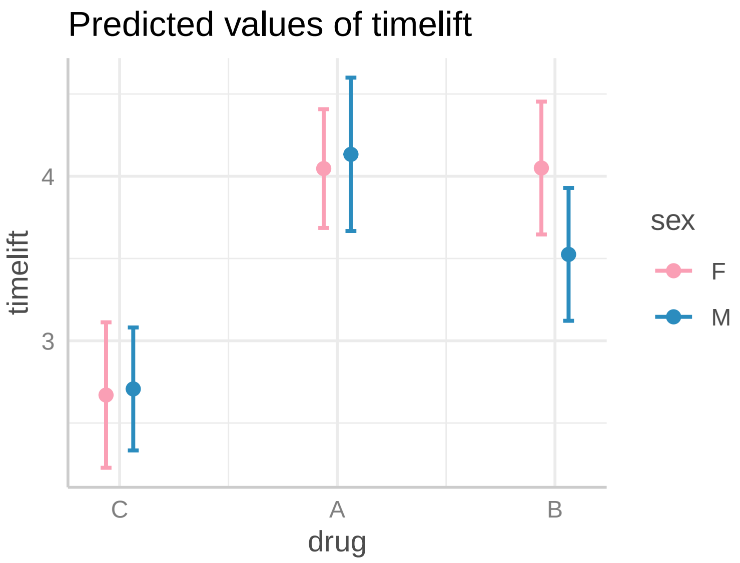
```
fit2 = lm(timelift ~ drug*sex, data =rats2)
summary(fit2)
```

```
##
## Call:
## lm(formula = timelift ~ drug * sex, data = rats2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.65000 -0.45500 -0.02917  0.45667  1.15333
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.67000    0.22563  11.833  < 2e-16 ***
## drugA        1.37667    0.29129   4.726 1.24e-05 ***
## drugB        1.38000    0.30551   4.517 2.66e-05 ***
## sexM         0.03714    0.29542   0.126    0.900
## drugA:sexM   0.04952    0.42164   0.117    0.907
## drugB:sexM  -0.56214    0.41488  -1.355    0.180
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7135 on 66 degrees of freedom
## Multiple R-squared:  0.4491, Adjusted R-squared:  0.4074
## F-statistic: 10.76 on 5 and 66 DF,  p-value: 1.365e-07
```

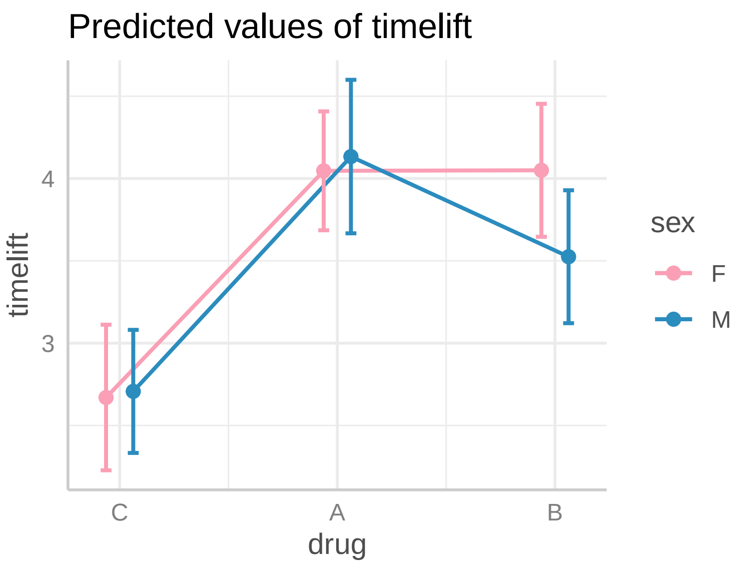
```
anova(fit2)
```

```
## Analysis of Variance Table
##
## Response: timelift
##           Df Sum Sq Mean Sq F value    Pr(>F)
## drug       2 25.689 12.8443 25.2298 7.264e-09 ***
## sex        1  0.343  0.3427  0.6731    0.4149
## drug:sex   2  1.361  0.6807  1.3371    0.2696
## Residuals 66 33.600  0.5091
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ggpred2 = ggpredict(fit2, c("drug", "sex")) # get mean and CIs
plot(ggpred2, colors = c("#fa9fb5","#2b8cbe"))
```



```
plot(ggpred2, connect.lines = TRUE, colors = c("#fa9fb5","#2b8cbe"))
```


Predicted values of timelift
timelift
4
3
C
A
B
drug
sex
F
M

3. factorial ANOVA

```
# data step
  faex =
    data.frame(hec = factor(rep(c(3,5), times = 10)),
               paa = factor(c(rep(c(1,3), each = 10))),
               zsv = c(1.9, 8.7,
                       2.3, 9.0,
                       2.0, 8.8,
                       2.1, 9.1,
                       2.0, 9.0,
                       4.3, 18.5,
                       4.7, 18.5,
                       4.8, 18.2,
                       4.7, 13.3,
                       4.8, 18.2))

# descriptive stats
  faex %>%
    group_by(hec, paa) %>%
    summarise(N = n(),
              mean = mean(zsv),
              median = median(zsv),
              var = var(zsv),
              sd = sd(zsv))
```

```
## ‘summarise()‘ has grouped output by ’hec’. You can override using the ‘.groups‘
## argument.
```

```
## # A tibble: 4 x 7
## # Groups:   hec [2]
##   hec   paa       N  mean median   var    sd
##   <fct> <fct> <int> <dbl>  <dbl> <dbl> <dbl>
## 1 3     1         5  2.06     2   0.023 0.152
## 2 3     3         5  4.66     4.7 0.043 0.207
## 3 5     1         5  8.92     9   0.027 0.164
## 4 5     3         5 17.3    18.2 5.12  2.26
```

3-1. Run factorial ANOVA

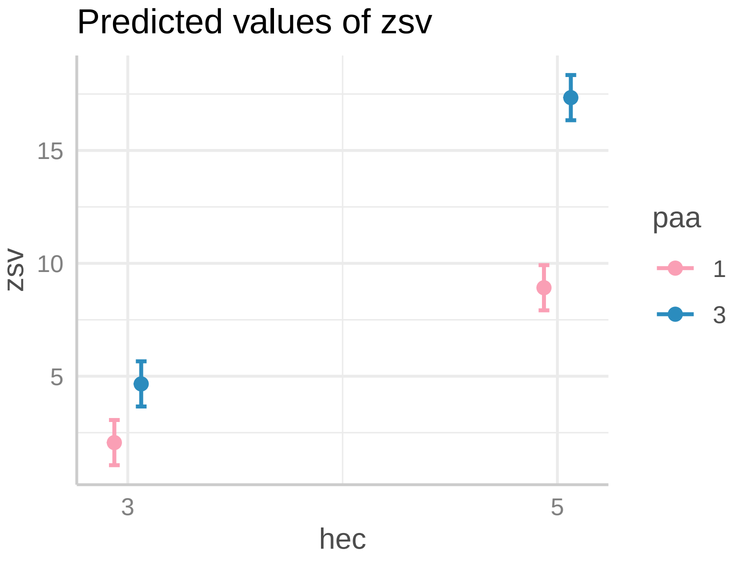
```
  fit3 = lm(zsv ~ hec*paa, data = faex)
  summary(fit3)
```

```
##
## Call:
## lm(formula = zsv ~ hec * paa, data = faex)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.040 -0.075  0.060  0.195  1.160
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0600     0.5107   4.034 0.000961 ***
## hec5          6.8600     0.7222   9.499 5.59e-08 ***
## paa3          2.6000     0.7222   3.600 0.002399 **
## hec5:paa3     5.8200     1.0214   5.698 3.30e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.142 on 16 degrees of freedom
## Multiple R-squared:  0.9699, Adjusted R-squared:  0.9642
## F-statistic: 171.6 on 3 and 16 DF,  p-value: 2.242e-12
```

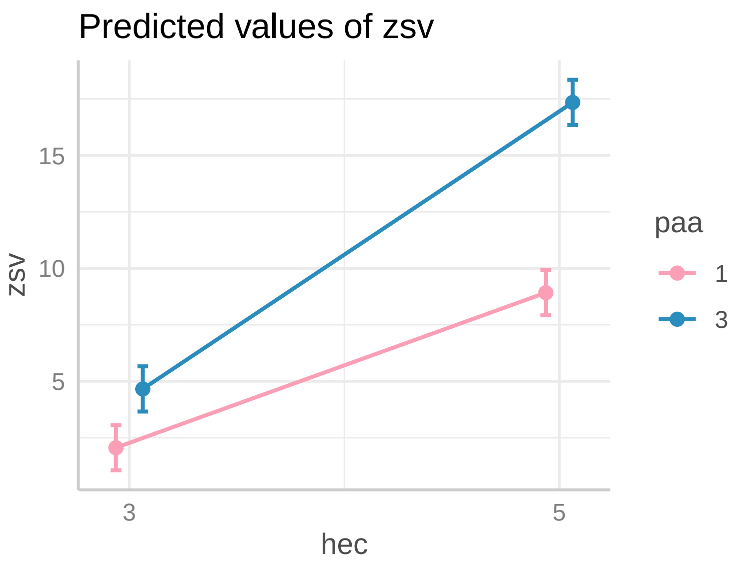
```
anova(fit3)
```

```
## Analysis of Variance Table
##
## Response: zsv
##           Df Sum Sq Mean Sq F value    Pr(>F)
## hec        1 477.26  477.26  366.00 1.896e-12 ***
## paa        1 151.80  151.80  116.41 9.448e-09 ***
## hec:paa    1  42.34   42.34   32.47 3.296e-05 ***
## Residuals 16  20.86    1.30
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ggpred3 = ggpredict(fit3, c("hec", "paa")) # get mean and CIs
plot(ggpred3, colors = c("#fa9fb5","#2b8cbe"))
```



```
plot(ggpred3, connect.lines = TRUE, colors = c("#fa9fb5","#2b8cbe"))
```



```
library(lsmeans)
```

```
## Loading required package: emmeans

## The 'lsmeans' package is now basically a front end for 'emmeans'.
## Users are encouraged to switch the rest of the way.
## See help('transition') for more information, including how to
## convert old 'lsmeans' objects and scripts to work with 'emmeans'.
```

```
lsmeans(fit3, c("hec", "paa")) # get LS menas
```

```
##  hec paa lsmean    SE df lower.CL upper.CL
##  3   1     2.06 0.511 16    0.977     3.14
##  5   1     8.92 0.511 16    7.837    10.00
##  3   3     4.66 0.511 16    3.577     5.74
##  5   3    17.34 0.511 16   16.257    18.42
##
## Confidence level used: 0.95
```

```
TukeyHSD(aov(zsv ~ hec*paa, data = faex), ordered = TRUE) # multiple comparison with Tukey method
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##     factor levels have been ordered
##
## Fit: aov(formula = zsv ~ hec * paa, data = faex)
##
## $hec
##     diff      lwr      upr p adj
## 5-3 9.77 8.687394 10.85261     0
##
## $paa
##     diff      lwr      upr p adj
## 3-1 5.51 4.427394 6.592606     0
##
## $‘hec:paa‘
##          diff        lwr       upr     p adj
## 3:3-3:1  2.60  0.5337177  4.666282 0.0115531
## 5:1-3:1  6.86  4.7937177  8.926282 0.0000003
## 5:3-3:1 15.28 13.2137177 17.346282 0.0000000
## 5:1-3:3  4.26  2.1937177  6.326282 0.0001199
## 5:3-3:3 12.68 10.6137177 14.746282 0.0000000
## 5:3-5:1  8.42  6.3537177 10.486282 0.0000000
```

4. Vision data

```
library(readr)
vision = read_csv("vision.csv")
```

```
## Rows: 313 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): race, gender, agegrp
## dbl (2): acuity, group
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
# I copy paste the data in the word file to excel..

vision %>% names
```

```
## [1] "acuity" "race"   "gender" "group"  "agegrp"
```

```
  vision$race = factor(vision$race, levels = c("Aboriginal","European"))
  vision$gender = factor(vision$gender, levels = c("Male", "Female"))
  vision$group = factor(vision$group, levels = c("1", "2", "3", "4"))
  vision$agegrp = factor(vision$agegrp, levels = c("20-29y"
, "30-39y"))
```

4-1. ANOVA

```
fit4 = lm(acuity ~ race*gender, data = vision)
summary(fit4)
```

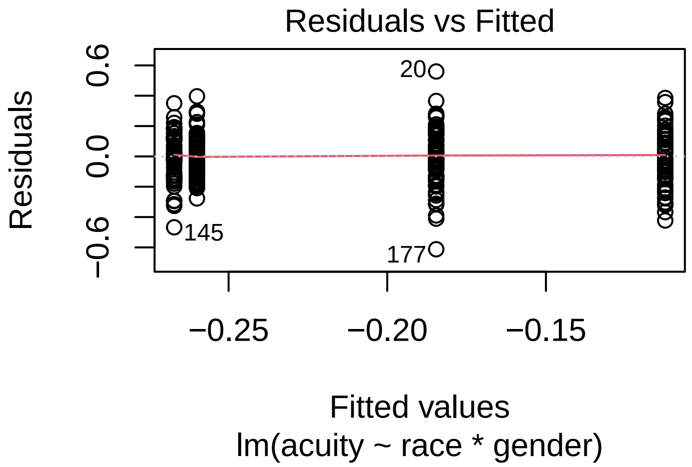
```
##
## Call:
## lm(formula = acuity ~ race * gender, data = vision)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61233 -0.11024  0.00136  0.12014  0.55997
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -0.259958   0.015950 -16.299  < 2e-16 ***
## raceEuropean               0.075391   0.023669   3.185  0.00159 **
## genderFemale              -0.007179   0.027540  -0.261  0.79452
## raceEuropean:genderFemale  0.079337   0.038683   2.051  0.04111 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
```

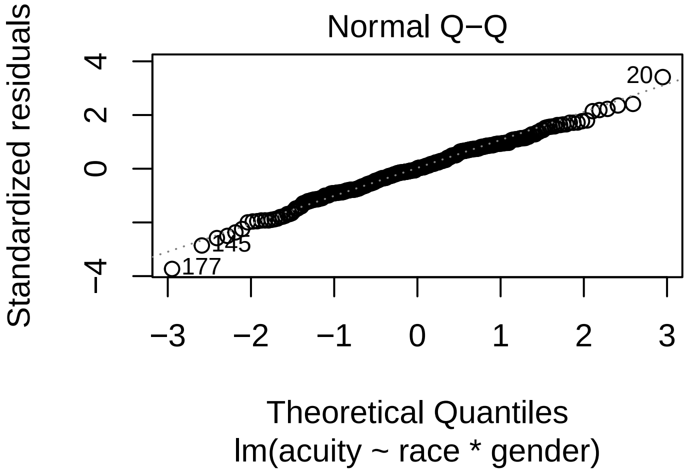
```
## Residual standard error: 0.165 on 309 degrees of freedom
## Multiple R-squared:  0.1157, Adjusted R-squared:  0.1072
## F-statistic: 13.48 on 3 and 309 DF,  p-value: 2.733e-08
```

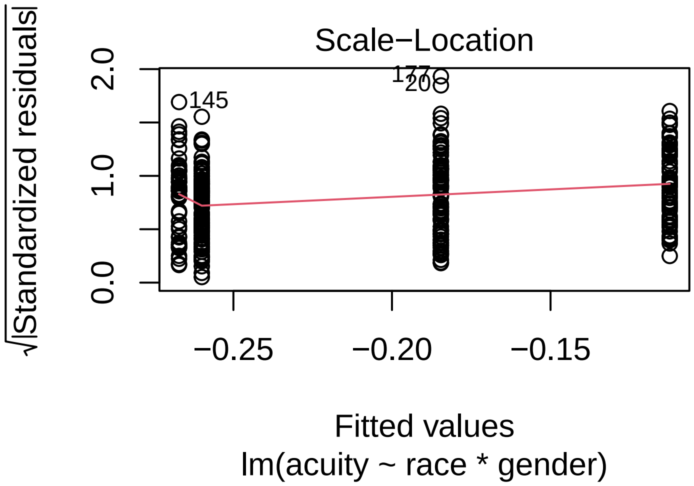
```
anova(fit4)
```

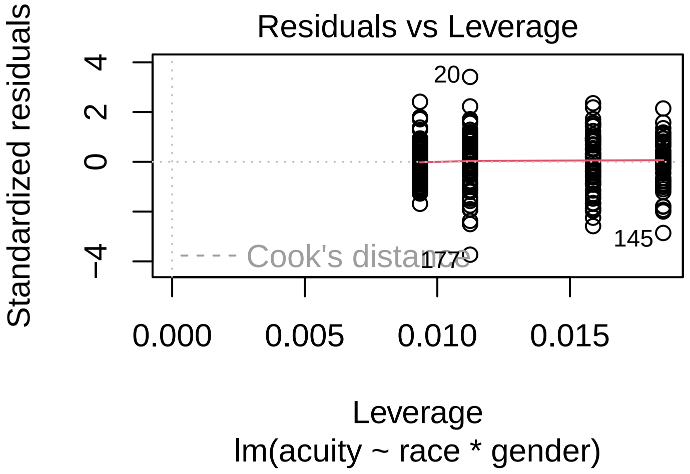
```
## Analysis of Variance Table
##
## Response: acuity
##              Df Sum Sq Mean Sq F value    Pr(>F)
## race          1 0.9070 0.90699 33.3209 1.901e-08 ***
## gender        1 0.0794 0.07942  2.9177   0.08862 .
## race:gender   1 0.1145 0.11450  4.2064   0.04111 *
## Residuals   309 8.4110 0.02722
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# assumption check
plot(fit4)
```

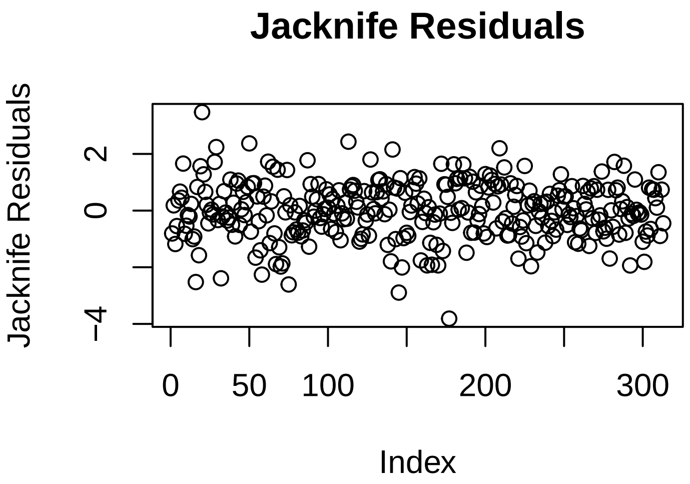



Normal Q−Q
Standardized residuals
4
2
0
−4
20
145
177
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(acuity ~ race * gender)

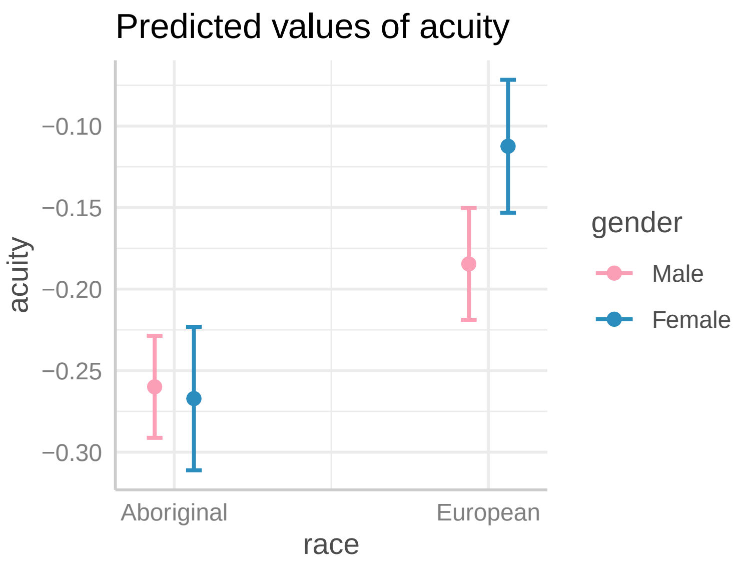

Scale−Location
√|Standardized residuals|
2.0
1.0
0.0
145
177
20
−0.25
−0.20
−0.15
Fitted values
lm(acuity ~ race * gender)



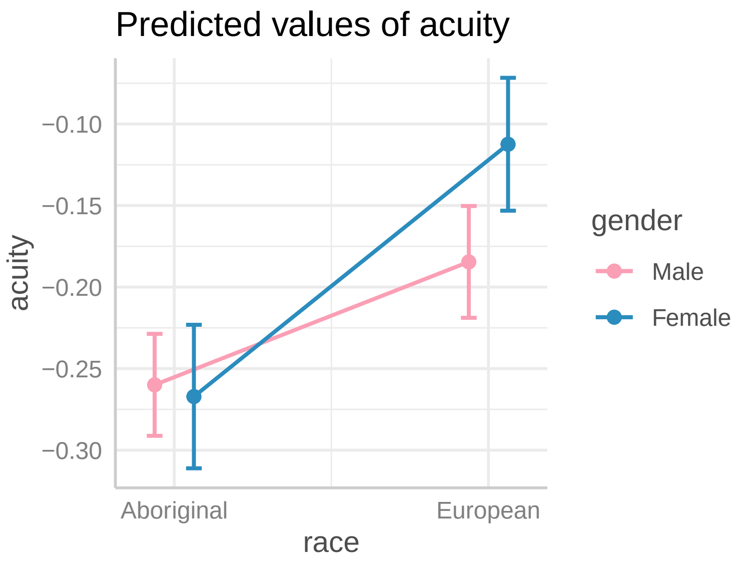
```
jack = rstudent(fit4)
plot(jack, ylab="Jacknife Residuals",main="Jacknife Residuals")
```



```
ggpred4 = ggpredict(fit4, c("race", "gender")) # get mean and CIs
plot(ggpred4, colors = c("#fa9fb5","#2b8cbe"))
```



```
plot(ggpred4, connect.lines = TRUE, colors = c("#fa9fb5","#2b8cbe"))
```



5. Campus data

```
campus = read_csv("campus.csv")
```

```
## Rows: 100 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (3): bacteria, cafeteria, brand
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
campus %>% names
```

```
## [1] "bacteria"  "cafeteria" "brand"
```

```
campus$cafeteria = campus$cafeteria %>% factor()
campus$brand = campus$brand %>% factor()
```

5-1. ANOVA

```
fit5 = lm(bacteria ~ brand*cafeteria, data = campus)
summary(fit5)
```

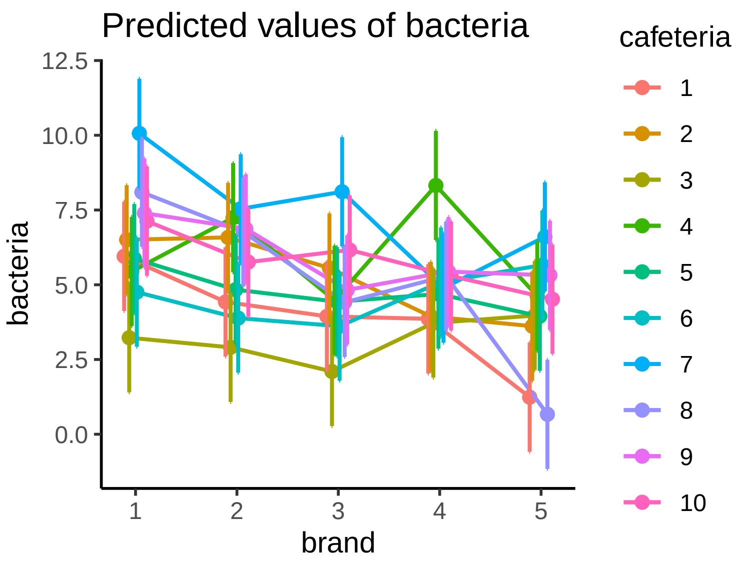
```
##
## Call:
## lm(formula = bacteria ~ brand * cafeteria, data = campus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0505 -0.6874  0.0000  0.6874  2.0505
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            5.9510     0.9320   6.385 5.49e-08 ***
## brand2                -1.5260     1.3180  -1.158 0.252453
## brand3                -2.0080     1.3180  -1.523 0.133940
## brand4                -2.0900     1.3180  -1.586 0.119114
## brand5                -4.7110     1.3180  -3.574 0.000789 ***
## cafeteria2             0.5545     1.3180   0.421 0.675776
## cafeteria3            -2.7215     1.3180  -2.065 0.044146 *
## cafeteria4            -0.5075     1.3180  -0.385 0.701839
## cafeteria5            -0.0780     1.3180  -0.059 0.953045
## cafeteria6            -1.1975     1.3180  -0.909 0.367947
## cafeteria7             4.1125     1.3180   3.120 0.002999 **
## cafeteria8             2.1425     1.3180   1.626 0.110338
## cafeteria9             1.4430     1.3180   1.095 0.278846
## cafeteria10          1.1760     1.3180   0.892 0.376540
## brand2:cafeteria2    1.6040     1.8640   0.861 0.393609
## brand3:cafeteria2    1.0620     1.8640   0.570 0.571400
## brand4:cafeteria2   -0.4755     1.8640  -0.255 0.799693
## brand5:cafeteria2    1.8205     1.8640   0.977 0.333434
## brand2:cafeteria3    1.2025     1.8640   0.645 0.521797
## brand3:cafeteria3    0.8780     1.8640   0.471 0.639665
## brand4:cafeteria3    2.5810     1.8640   1.385 0.172302
## brand5:cafeteria3    5.4530     1.8640   2.925 0.005160 **
## brand2:cafeteria4    3.3255     1.8640   1.784 0.080479 .
## brand3:cafeteria4    1.0485     1.8640   0.563 0.576286
## brand4:cafeteria4    4.9680     1.8640   2.665 0.010331 *
## brand5:cafeteria4    3.8875     1.8640   2.086 0.042140 *
## brand2:cafeteria5    0.4885     1.8640   0.262 0.794342
## brand3:cafeteria5    0.5685     1.8640   0.305 0.761638
## brand4:cafeteria5    0.9080     1.8640   0.487 0.628296
## brand5:cafeteria5    2.7870     1.8640   1.495 0.141152
## brand2:cafeteria6    0.6570     1.8640   0.352 0.725967
## brand3:cafeteria6    0.8695     1.8640   0.466 0.642903
## brand4:cafeteria6    2.4270     1.8640   1.302 0.198866
## brand5:cafeteria6    5.6150     1.8640   3.012 0.004060 **
## brand2:cafeteria7   -0.9965     1.8640  -0.535 0.595291
## brand3:cafeteria7    0.0555     1.8640   0.030 0.976365
## brand4:cafeteria7   -3.0865     1.8640  -1.656 0.104014
## brand5:cafeteria7    1.2520     1.8640   0.672 0.504880
## brand2:cafeteria8    0.2015     1.8640   0.108 0.914348
## brand3:cafeteria8   -1.6745     1.8640  -0.898 0.373308
## brand4:cafeteria8   -0.7125     1.8640  -0.382 0.703899
## brand5:cafeteria8   -2.7155     1.8640  -1.457 0.151417
## brand2:cafeteria9    1.0055     1.8640   0.539 0.591981
## brand3:cafeteria9   -0.5595     1.8640  -0.300 0.765297
## brand4:cafeteria9    0.1355     1.8640   0.073 0.942340
## brand5:cafeteria9    2.6335     1.8640   1.413 0.163900
## brand2:cafeteria10   0.1570     1.8640   0.084 0.933211
## brand3:cafeteria10   1.0440     1.8640   0.560 0.577919
## brand4:cafeteria10   0.2560     1.8640   0.137 0.891314
## brand5:cafeteria10   2.1000     1.8640   1.127 0.265282
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.318 on 50 degrees of freedom
## Multiple R-squared:  0.7843, Adjusted R-squared:  0.573
## F-statistic: 3.711 on 49 and 50 DF,  p-value: 4.293e-06
```

```
anova(fit5)
```

```
## Analysis of Variance Table
##
## Response: bacteria
##                 Df  Sum Sq Mean Sq F value    Pr(>F)
## brand            4  67.831 16.9577  9.7614 6.483e-06 ***
## cafeteria        9 129.912 14.4347  8.3091 1.946e-07 ***
## brand:cafeteria 36 118.159  3.2822  1.8893    0.0187 *
## Residuals       50  86.861  1.7372
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ggpred5 = ggpredict(fit5, c("brand", "cafeteria")) # get mean and CIs
plot(ggpred5, connect.lines = TRUE, jitter = 0.3) +
  scale_fill_brewer(palette="Spectral") +
  theme_classic()
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```



```
library(lsmeans)
lsmeans(fit5, c("brand", "cafeteria")) # get LS menas
```

```
##  brand cafeteria lsmean    SE df lower.CL upper.CL
##  1     1          5.951 0.932 50    4.079     7.82
##  2     1          4.425 0.932 50    2.553     6.30
##  3     1          3.943 0.932 50    2.071     5.81
##  4     1          3.861 0.932 50    1.989     5.73
##  5     1          1.240 0.932 50   -0.632     3.11
##  1     2          6.505 0.932 50    4.634     8.38
##  2     2          6.583 0.932 50    4.712     8.46
##  3     2          5.559 0.932 50    3.688     7.43
##  4     2          3.940 0.932 50    2.068     5.81
##  5     2          3.615 0.932 50    1.743     5.49
##  1     3          3.229 0.932 50    1.358     5.10
##  2     3          2.906 0.932 50    1.034     4.78
##  3     3          2.099 0.932 50    0.228     3.97
##  4     3          3.720 0.932 50    1.849     5.59
##  5     3          3.971 0.932 50    2.100     5.84
##  1     4          5.444 0.932 50    3.572     7.32
##  2     4          7.243 0.932 50    5.371     9.11
##  3     4          4.484 0.932 50    2.612     6.36
##  4     4          8.322 0.932 50    6.450    10.19
##  5     4          4.620 0.932 50    2.748     6.49
##  1     5          5.873 0.932 50    4.001     7.74
##  2     5          4.835 0.932 50    2.964     6.71
##  3     5          4.434 0.932 50    2.562     6.31
##  4     5          4.691 0.932 50    2.819     6.56
##  5     5          3.949 0.932 50    2.077     5.82
##  1     6          4.753 0.932 50    2.882     6.63
##  2     6          3.885 0.932 50    2.013     5.76
##  3     6          3.615 0.932 50    1.743     5.49
##  4     6          5.090 0.932 50    3.219     6.96
##  5     6          5.657 0.932 50    3.786     7.53
##  1     7         10.063 0.932 50    8.192    11.94
##  2     7          7.541 0.932 50    5.669     9.41
##  3     7          8.111 0.932 50    6.239     9.98
##  4     7          4.887 0.932 50    3.015     6.76
##  5     7          6.604 0.932 50    4.733     8.48
##  1     8          8.094 0.932 50    6.222     9.97
##  2     8          6.769 0.932 50    4.897     8.64
##  3     8          4.411 0.932 50    2.539     6.28
##  4     8          5.291 0.932 50    3.419     7.16
##  5     8          0.667 0.932 50   -1.205     2.54
##  1     9          7.394 0.932 50    5.522     9.27
##  2     9          6.873 0.932 50    5.002     8.75
##  3     9          4.827 0.932 50    2.955     6.70
##  4     9          5.439 0.932 50    3.568     7.31
##  5     9          5.316 0.932 50    3.445     7.19
##  1     10         7.127 0.932 50    5.255     9.00
##  2     10         5.758 0.932 50    3.886     7.63
##  3     10         6.163 0.932 50    4.291     8.03
##  4     10         5.293 0.932 50    3.421     7.16
##  5     10         4.516 0.932 50    2.644     6.39
##
## Confidence level used: 0.95
```

6. irradiate

```
irradiate = read_csv("irradiate.csv")
```

```
## Rows: 100 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (3): bacteria, batch, duration
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
irradiate %>% names
```

```
## [1] "bacteria" "batch"    "duration"
```

```
irradiate$batch = factor(irradiate$batch)
irradiate$duration = factor(irradiate$duration)
```

6-1. linear mixed effect model

```
library(lme4); library(lmerTest)
```

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

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

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

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

```
fit6  = lmer(bacteria ~ duration + (1 + duration | batch), data = irradiate) # mixed-effect model wit
```

```
## boundary (singular) fit: see help('isSingular')
```

```
# Calculate LS means with Tukey's adjustment
lsmeans(fit6, "duration", adjust = "tukey")
```

```
## Note: adjust = "tukey" was changed to "sidak"
## because "tukey" is only appropriate for one set of pairwise comparisons
##  duration lsmean    SE df lower.CL upper.CL
##  1          6.44 0.615  9     4.45     8.43
##  2          5.68 0.527  9     3.98     7.39
##  3          4.76 0.538  9     3.02     6.50
##  4          5.05 0.430  9     3.66     6.45
##  5          4.02 0.587  9     2.11     5.92
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
## Conf-level adjustment: sidak method for 5 estimates
```