Exercise Example: (Basic) One-Factor Repeated Measures

This example is taken from Ott & Longecker. An exercise physiologist designed a study to evaluate the impact of the steepness of running courses on the peak heart rate (Y=PHR). There are 4 courses: flat, slightly steep, moderately steep and very steep. A total of 20 runners will run each of the courses in a randomly assigned order. There will be sufficient time between runs so that there should not be any carryover effect.

This is a one-factor repeated measures analysis. The runners are the subjects and the within-subjects factor is course (Flat, Slight, Moderate or Steep).

Notes:

1. The data is originally in "wide" format, but needs to be "transposed" to "long" format before analysis.
2. Using the "basic" repeated measures analysis is equivalent to analyzing the data as an RCB.
3. We run the analysis using both lme4 (lmer) and lme (nlme). This is just for illustration. The two approaches give identical results.

```
library(tidyverse)
Exercise <- read.csv("C:/hess/STAT512/RNotes/Random3/R3_Exercise.csv")
str(Exercise)
```

```
## 'data.frame':    20 obs. of  5 variables:
##  $ Runner  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Flat    : int  133 138 133 128 130 139 123 128 109 143 ...
##  $ Slight  : int  143 136 149 144 139 152 129 132 137 151 ...
##  $ Moderate: int  155 142 154 143 136 152 131 142 122 161 ...
##  $ Steep   : int  154 154 151 150 145 163 142 148 128 160 ...
```

```
head(Exercise)
```

```
##   Runner Flat Slight Moderate Steep
## 1      1  133    143      155   154
## 2      2  138    136      142   154
## 3      3  133    149      154   151
## 4      4  128    144      143   150
## 5      5  130    139      136   145
## 6      6  139    152      152   163
```

```
#Important: Need to define Runner as.factor!!!
Exercise$Runner <- as.factor(Exercise$Runner)
#Transpose to "Long" form
TrExercise <- gather(data = Exercise, key = "Course", value = "Y", -Runner)
str(TrExercise)
```

```
## 'data.frame':    80 obs. of  3 variables:
##  $ Runner: Factor w/ 20 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Course: chr  "Flat" "Flat" "Flat" "Flat" ...
##  $ Y     : int  133 138 133 128 130 139 123 128 109 143 ...
```

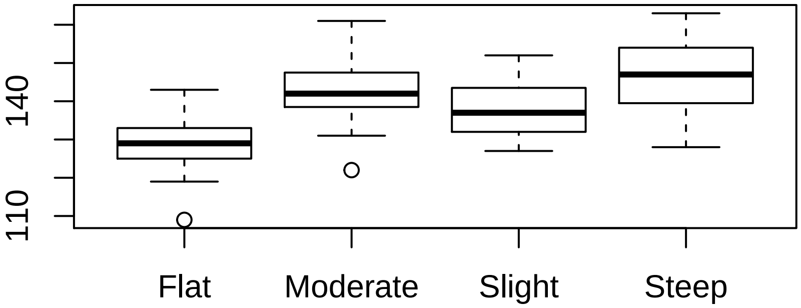
```
head(TrExercise)
```

```
##   Runner Course   Y
## 1      1   Flat 133
## 2      2   Flat 138
## 3      3   Flat 133
## 4      4   Flat 128
## 5      5   Flat 130
## 6      6   Flat 139
```

```
#Summary Statistics
SumStats <- summarize(group_by(TrExercise, Course),
                  n = n(),
                  mean = mean(Y),
                  sd = sd(Y),
                  SE = sd/sqrt(n))
SumStats
```

```
## # A tibble: 4 x 5
##   Course        n  mean    sd    SE
##   <chr>     <int> <dbl> <dbl> <dbl>
## 1 Flat         20  129.  7.79  1.74
## 2 Moderate     20  143.  8.76  1.96
## 3 Slight       20  138.  7.27  1.63
## 4 Steep        20  147.  8.93  2.00
```

```
boxplot(Y ~ Course, data = TrExercise)
```



Approach 1: Using lmer (from lme4)

```
library(lme4)
library(lmerTest)
library(pbkrtest)
library(emmeans)
Model1 <- lmer(Y ~ Course + (1|Runner), data = TrExercise)
summary(Model1)
```

```
## Linear mixed model fit by REML t-tests use Satterthwaite approximations
##   to degrees of freedom [lmerMod]
## Formula: Y ~ Course + (1 | Runner)
##    Data: TrExercise
##
## REML criterion at convergence: 497.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.57194 -0.63464 -0.02782  0.59692  2.90060
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Runner   (Intercept) 48.52    6.965
##  Residual             19.01    4.360
## Number of obs: 80, groups:  Runner, 20
##
## Fixed effects:
##                Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)     129.000      1.837  29.820  70.205  < 2e-16 ***
## CourseModerate   13.800      1.379  57.000  10.009 3.64e-14 ***
## CourseSlight      9.300      1.379  57.000   6.745 8.44e-09 ***
## CourseSteep      18.150      1.379  57.000  13.164  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) CrsMdr CrsSlg
## CourseModrt -0.375
## CourseSlght -0.375  0.500
## CourseSteep -0.375  0.500  0.500
```

```
anova(Model1, ddf="Kenward-Roger")
```

```
## Analysis of Variance Table of type III  with  Kenward-Roger
## approximation for degrees of freedom
##        Sum Sq Mean Sq NumDF DenDF F.value    Pr(>F)
## Course 3619.2  1206.4     3    57  63.465 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
emmeans(Model1, pairwise ~ Course)
```

```
## $emmeans
##  Course   emmean       SE    df lower.CL upper.CL
##  Flat     129.00 1.837466 29.82 125.2464 132.7536
##  Moderate 142.80 1.837466 29.82 139.0464 146.5536
##  Slight   138.30 1.837466 29.82 134.5464 142.0536
##  Steep    147.15 1.837466 29.82 143.3964 150.9036
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
##
## $contrasts
##  contrast          estimate       SE df t.ratio p.value
##  Flat - Moderate     -13.80 1.378731 57 -10.009  <.0001
##  Flat - Slight        -9.30 1.378731 57  -6.745  <.0001
##  Flat - Steep        -18.15 1.378731 57 -13.164  <.0001
##  Moderate - Slight     4.50 1.378731 57   3.264  0.0098
##  Moderate - Steep     -4.35 1.378731 57  -3.155  0.0133
##  Slight - Steep       -8.85 1.378731 57  -6.419  <.0001
##
## P value adjustment: tukey method for comparing a family of 4 estimates
```

```
detach("package:lmerTest")
detach("package:pbkrtest")
detach("package:lme4")
detach("package:emmeans")
```

Approach 2: Using lme (from nlme)

Use anova.lme(, type = “marginal”) to calculate the Type 3 tests for lme models.

```
library(nlme)
library(emmeans)
Model2 <- lme(Y ~ Course, random = ~1|Runner, data = TrExercise)
summary(Model2)
```

```
## Linear mixed-effects model fit by REML
##  Data: TrExercise
##        AIC      BIC    logLik
##   509.3929 523.3773 -248.6964
##
## Random effects:
##  Formula: ~1 | Runner
##         (Intercept) Residual
## StdDev:    6.965391  4.35993
##
## Fixed effects: Y ~ Course
##                 Value Std.Error DF  t-value p-value
## (Intercept)    129.00  1.837466 57 70.20536       0
## CourseModerate  13.80  1.378731 57 10.00920       0
## CourseSlight     9.30  1.378731 57  6.74533       0
## CourseSteep     18.15  1.378731 57 13.16428       0
##  Correlation:
##                (Intr) CrsMdr CrsSlg
## CourseModerate -0.375
## CourseSlight   -0.375  0.500
## CourseSteep    -0.375  0.500  0.500
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
## -1.5719437 -0.6346395 -0.0278183  0.5969242  2.9006050
##
## Number of Observations: 80
## Number of Groups: 20
```

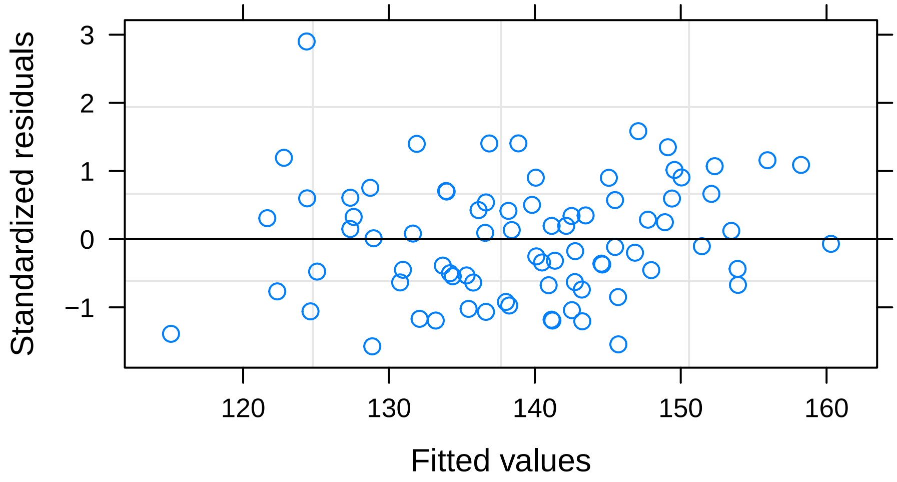
```
anova.lme(Model2, type = "marginal")
```

```
##             numDF denDF  F-value p-value
## (Intercept)     1    57 4928.793  <.0001
## Course          3    57   63.465  <.0001
```

```r
emmeans(Model2, pairwise ~ Course)
```

```
## $emmeans
##  Course   emmean       SE df lower.CL upper.CL
##  Flat     129.00 1.837466 19 125.1541 132.8459
##  Moderate 142.80 1.837466 19 138.9541 146.6459
##  Slight   138.30 1.837466 19 134.4541 142.1459
##  Steep    147.15 1.837466 19 143.3041 150.9959
##
## Confidence level used: 0.95
##
## $contrasts
##  contrast          estimate       SE df t.ratio p.value
##  Flat - Moderate     -13.80 1.378731 57 -10.009  <.0001
##  Flat - Slight        -9.30 1.378731 57  -6.745  <.0001
##  Flat - Steep        -18.15 1.378731 57 -13.164  <.0001
##  Moderate - Slight     4.50 1.378731 57   3.264  0.0098
##  Moderate - Steep     -4.35 1.378731 57  -3.155  0.0133
##  Slight - Steep       -8.85 1.378731 57  -6.419  <.0001
##
## P value adjustment: tukey method for comparing a family of 4 estimates
```

```r
plot(Model2)
```


Standardized residuals
Fitted values