Math 420 HW 2

Fall 2023

Washington University in St. Louis

Due date: Saturday, 10/7/2023

Instruction:

Please type your answers clearly and show your work neatly. You are encouraged to use the Rmarkdown version of this assignment as a template to submit your work. Unless stated otherwise, all programming references in the assignment will be in R. For this assignment, problems roughly covers content from Factorial Designs, Random Block Designs, Variance Components and Fractional Factorial Designs

Problem 1

Kenett and Steinberg (1987) described a two-level factorial experiment conducted by students to study the time required to boil 1 qt of water. Factors were A=flame level (low or high), B=pan size (small or large), C=pan cover (none or glass cover), and D=salt added to water (no or yes).

(a) If the standard deviation in boiling time (tested at the same conditions) was found to be $\sigma^2 = 0.236$ minutes, use the shortcut formula to determine how many experiments you will need to perform in order to have power of 0.95 for detecting effects of size $\Delta = 0.50$ minutes. Would this answer change if you decided to only perform an experiment with 3 of the 4 factors?

Ans : The short cut approximation formula for determining the number of runs needed to achieve power equal to 0.95 when the sig. level for a two-level factorial is $\alpha = 0.05$ is the following:

$$r \times 2^k = N = (\frac{8\sigma}{\Delta})^2 \implies r = N/2^k$$

where σ is the standard dev of the experimental error, Δ is the practical size of an effect.

(Version 1): Note - assuming $\sigma = 0.236$, r will be close to 0.9. In the practical context we will then need to round up to $r = 1$ replicate per treatment level combination, resulting in 16 total experiments. If the experiment has 3 factors instead of 4, providing everything else is equal, the total number of experiments would not change (16), however it would imply that $r \times 2^3 = 16$, so each treatment combo will have 2 reps instead of 1.

```
## version 1:
N = (8*0.236/0.5)^2
print(paste('total # of runs N =',N))
```

```
## [1] "total # of runs N = 14.258176"
```

```
r4 = round(N/(2^4),0)
r3 = round(N/(2^3),0)
print(paste('r4 =', r4,' r3 =', r3 ))
```

```
## [1] "r4 = 1  r3 = 2"
```

(Version 2): Note - assuming $\sigma = \sqrt{0}.236$, then $N = 60$, and $r \times 2^4 = 60 = 3.75$ implies we will take 4 replicates per treatment combination.

```
## version 1:
N = (8*sqrt(0.236)/0.5)^2
print(paste('total # of runs N =',N))
```

```
## [1] "total # of runs N = 60.416"
```

```
r4 = round(N/(2^4),0)
r3 = round(N/(2^3),0)
print(paste('r4 =', r4,' r3 =', r3 ))
```

```
## [1] "r4 = 4  r3 = 8"
```

(b) Create a list of experiments in random order for performing these experiments.

Ans: we can use `expand.grid` to create the experiment plan. This example is illustrated for Version 1 using 1 replicate per treatment level combination.

```
D <-expand.grid(A = c("Low", "High"),
                B = c("Small", "Large"),
                C = c("None", "Glass Cover"),
                D = c("No","Yes"))

set.seed(2023)
D<-D[order(sample(1:nrow(D))),]
pander(D)
```

	A	B	C	D
8	High	Large	Glass Cover	No
7	Low	Large	Glass Cover	No
5	Low	Small	Glass Cover	No
14	High	Small	Glass Cover	Yes
1	Low	Small	None	No
12	High	Large	None	Yes
13	Low	Small	Glass Cover	Yes
4	High	Large	None	No
3	Low	Large	None	No
6	High	Small	Glass Cover	No
15	Low	Large	Glass Cover	Yes
16	High	Large	Glass Cover	Yes
11	Low	Large	None	Yes

	A	B	C	D
9	Low	Small	None	Yes
2	High	Small	None	No
10	High	Small	None	Yes

(Version 2)

```
D <-expand.grid(A = c("Low", "High"),
                B = c("Small", "Large"),
                C = c("None", "Glass Cover"),
                D = c("No","Yes"))
## replicate 4 times
D <-rbind(D,D,D,D)
set.seed(2023)
D<-D[order(sample(1:nrow(D))),]
pander(D)
```

	A	B	C	D
11	Low	Large	None	Yes
38	High	Small	Glass Cover	No
18	High	Small	None	No
21	Low	Small	Glass Cover	No
15	Low	Large	Glass Cover	Yes
33	Low	Small	None	No
42	High	Small	None	Yes
5	Low	Small	Glass Cover	No
25	Low	Small	None	Yes
47	Low	Large	Glass Cover	Yes
62	High	Small	Glass Cover	Yes
48	High	Large	Glass Cover	Yes
45	Low	Small	Glass Cover	Yes
43	Low	Large	None	Yes
19	Low	Large	None	No
2	High	Small	None	No
14	High	Small	Glass Cover	Yes
37	Low	Small	Glass Cover	No
56	High	Large	Glass Cover	No
41	Low	Small	None	Yes
28	High	Large	None	Yes
46	High	Small	Glass Cover	Yes
59	Low	Large	None	Yes
24	High	Large	Glass Cover	No
64	High	Large	Glass Cover	Yes
7	Low	Large	Glass Cover	No
57	Low	Small	None	Yes
52	High	Large	None	No
12	High	Large	None	Yes
29	Low	Small	Glass Cover	Yes
30	High	Small	Glass Cover	Yes
22	High	Small	Glass Cover	No
32	High	Large	Glass Cover	Yes

	A	B	C	D
9	Low	Small	None	Yes
49	Low	Small	None	No
31	Low	Large	Glass Cover	Yes
50	High	Small	None	No
27	Low	Large	None	Yes
36	High	Large	None	No
63	Low	Large	Glass Cover	Yes
4	High	Large	None	No
51	Low	Large	None	No
35	Low	Large	None	No
8	High	Large	Glass Cover	No
20	High	Large	None	No
54	High	Small	Glass Cover	No
3	Low	Large	None	No
26	High	Small	None	Yes
10	High	Small	None	Yes
23	Low	Large	Glass Cover	No
6	High	Small	Glass Cover	No
40	High	Large	Glass Cover	No
1	Low	Small	None	No
44	High	Large	None	Yes
13	Low	Small	Glass Cover	Yes
17	Low	Small	None	No
34	High	Small	None	No
53	Low	Small	Glass Cover	No
61	Low	Small	Glass Cover	Yes
16	High	Large	Glass Cover	Yes
55	Low	Large	Glass Cover	No
39	Low	Large	Glass Cover	No
58	High	Small	None	Yes
60	High	Large	None	Yes

Problem 2

Lew (2007) presents the data from an experiment to determine whether cultured cells respond to two drugs. The experiment was conducted using a stable cell line plated onto Petri dishes, with each experimental run involving assays of responses in three Petri dishes: one treated with drug 1, one treated with drug 2, and one untreated serving as a control. The data are shown in the table below:

	Control	Drug1	Drug2
Exp1	1147	1169	1009
Exp2	1283	1323	1260
Exp3	1216	1276	1143
Exp4	1046	1240	1099
Exp5	1108	1432	1385
Exp6	1265	1562	1164

(a) Analyze the data as if it came from a completely randomized design using the model $y_{ij} = \mu + \tau_i + \epsilon_{ij}$. Is there a significant difference between the treatment groups?

Ans: The results produced an F ratio of 3.34 (where $F = MS_T/MS_E = 50510/15105$) and its p value $= 0.063$. Assuming $\alpha = 0.05$ to be the significance level, we therefore would fail to reject the null hypothesis that there is no significant difference between treatment groups.

```
## setting the variables
t1 <- rep('Control',6)
t2 <- rep('Drug1',6)
t3 <- rep('Drug2',6)
treatment<-as.factor(c(t1,t2,t3))
## experiment block
exp_levels<-c('Exp1','Exp2','Exp3','Exp4','Exp5','Exp6')
exp<-as.factor(rep(exp_levels,3))
## response variables
resp<-c(1147,1273,1216,1046,1108,1265,
        1169,1323,1276,1249,1432,1562,
        1009,1260,1143,1099,1385,1164)

lew<-data.frame(exp = exp,trt = treatment, resp = resp)

## CRD design without blocking
mod1 <- aov( resp ~ trt, data = lew)
summary(mod1)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## trt          2 101020   50510   3.344  0.063 .
## Residuals   15 226595   15106
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(b) Analyze the data as an RCB design, where experiment number represents a blocking factor.

Ans: The RCB design can be written as $y_{ijk} = b_i + \tau_j + \epsilon_{ijk}$, where b_i is the blocking factor and τ_j is the treatment effect.

```
## CRBD design (with blocking)
mod2 <- aov( resp ~ exp + trt, data = lew)
summary(mod2)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## exp          5 132405   26481   2.811 0.0772 .
## trt          2 101020   50510   5.363 0.0262 *
## Residuals   10  94190    9419
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(c) Is there any difference in the results you obtain in (a) and (b)? If so explain what may be the cause of the difference in the results and which method would you recommend?

Ans: By addressing experiment as a blocking factor, we can control the homogeneity of the observations to better isolate the treatment effect. As a result, we can see that there is indeed significant difference between treatment group in the RCB design. In this regard, design (b) is a better option as it better models the sources of variation introduced by the experiments.

Problem 3

Consider the data in Table 5.20 (p.216 in DAE with R book) from Smith and Beverly (1981) taken from a staggered nested design to investigate the sources of variability in impurities in raw materials received at a plant in trailer loads. Two samples of material were taken from each of nine trailer loads of pellets. Two measurements of impurities were made on the first sample from each trailer but only one measurement for the second sample from each trailer.

(a) Write the model for the data.

Ans. This is a 3-stage staggered nested model which can be written as

$$y_{ijk} = \mu + a_i + b_{(i)j} + \epsilon_{ijk}$$

where μ is the main impurity effect, a_i is the random effect from the i^{th} trailer, $b_{(i)j}$ is the random sample effect, and ϵ_{ijk} is the model error which also represents the random measurement effect nested within the sample and trailer level

(b) Analyze the data and estimate the three variance components using the method of moments.

Ans: first, let's create the dataset. Be sure the assignment of the observations correspond to the right hierarchical structure of the trailer, sample and measurement. Below is one way to do it :

```
## first create the dataset according to the nested structure
## create trailer structure
trailer <-NULL

for(i in 1:10){
  trailer <-c(trailer,rep(i,3))
}
## create sample structure
sample <- rep(c(1,1,2), 10)
## create measurement structure
msmt <- rep(c(1,2,1), 10)
## assign response variable
value <- c(47.06,44.37,49.3,
           47.43,50.35,50.42,
           48.9,48.05,50.64,
           52.32,52.26,53.47,
           46.53,45.60,53.98,
           46.99,50.87,51.87,
           47.49,51.55,58.57,
           47.41,47.63,48.63,
           48.37,51.03,50.15,
           54.8,51.57,54.52)
raw_material = data.frame(trailer = factor(trailer),
                          sample = factor(sample),
                          msmt = factor(msmt),
                          value = value)
pander(head(raw_material))
```

trailer	sample	msmt	value
1	1	1	47.06
1	1	2	44.37
1	2	1	49.3
2	1	1	47.43
2	1	2	50.35
2	2	1	50.42

To create the nested model, we can use the `aov` function in R with the right set of nested effects. Note that by design of the dataset, we don't have enough sample to make the measurement its own 'level', therefore the random effect is combined with the model error ϵ here. From the `mod3_aov` object we can then find the estimated mean square errors corresponding to each source of variation.

```
## part b nested model design implemented using aov
mod3_aov <- aov( value ~ trailer +
                 trailer:sample, data = raw_material)
summary(mod3_aov)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## trailer         9 130.44  14.493   4.362 0.0155 *
## trailer:sample 10 118.46  11.846   3.565 0.0286 *
## Residuals      10  33.22   3.322
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Using the table (Table 5.12) shown on the book, we can proceed to calculate the variance component estimation using method of moment. Here B, A stand for $b_{(i)j}, a_i$ in the model respectively:

```
var_est_3_stage<-function(msA,msB,msC){
  ## input variance mean square error estimate from ANOVA table
  ## formula taken from the inverse of Table 5.12 in book
  sig2C <-msC
  sig2B <-(msB - sig2C)*(3/4)
  sig2A <- (msA - sig2C - (5/3)*sig2B)*(1/3)
  sig2_est <- data.frame(B=sig2B,A=sig2A,Residual=sig2C)
  return(sig2_est)
}

s3 <- summary(mod3_aov)
ms_est<-s3[[1]]$`Mean Sq`
sig2_est <- var_est_3_stage(msA = ms_est[1],
                            msB = ms_est[2],
                            msC = ms_est[3])
pander(sig2_est)
```

B	A	Residual
6.392	0.1722	3.322

(c) Analyze the data using REML and check to see if your estimates remain the same.

Ans. REML estimates can be obtained when using the `lmer()` function to implement the model. Compared with the estimates from part b, we can see that the variance components are generallly similar with a noticeable exception of σ_a^2, which is the variance component corresponding to the random trailer effect (0.7056 vs 0.172). This could be due to several reasons. 1) MoM vs REML are two different estimate methods and 2) there could be atypical values in the data.

```
## part c using REML:

mod3_reml <-lmer(value ~ 1 + (1|trailer)
                 + (1|trailer:sample), data = raw_material)
print(VarCorr(mod3_reml),comp="Variance")
```

```
##  Groups         Name        Variance
##  trailer:sample (Intercept) 6.1401
##  trailer        (Intercept) 0.7056
##  Residual                   3.3987
```

Another technique to check for questions in consisteny between two estimation methods is to utilize confidence interval, a quick call to `confint()` function for the lmer model provides the following, where .sig01,.sig02, are σ_a, σ_b respectively, and .sigma is σ. We can see the range of values, especially for the factor a and b are quite large.

```
pander(confint(mod3_reml))
```

```
## Computing profile confidence intervals ...
```

	2.5 %	97.5 %
.sig01	0	3.924
.sig02	0	2.934
.sigma	1.251	3.121
(Intercept)	48.98	51.9

(d) Make half-normal plots of the square root of the variances pooled to get the mean squares for sample(trailer) and measurement(sample). Does the assumption of homogeneous variances appear reasonable?

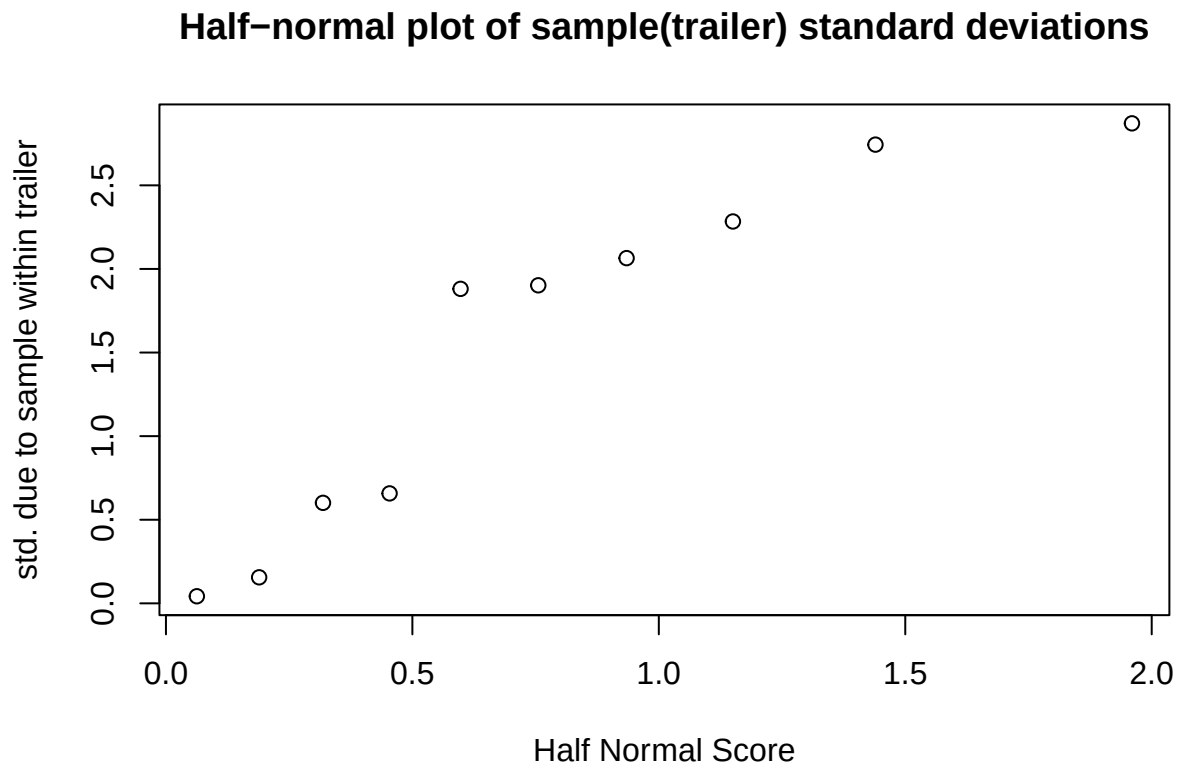
Ans: First, create an array to represent all observations at the same level pivoted by trailer and calculate the pooled standard deviations

```
y<-array(raw_material$value,c(3,10))
## pooled sd at sample level
sd1 <- sqrt((y[2,]-y[1,])**2/2)
## pooled sd at trailer level
sd2 <- sqrt(2/3*(y[3,]-(y[2,]+y[1,])/2)**2)
pooled_data<-data.frame(trailer = c(1:10),
                        y1=y[1,],
                        y2=y[2,],
                        y3=y[3,],
                        sd1=sd1,sd2=sd2)
pander(pooled_data)
```

trailer	y1	y2	y3	sd1	sd2
1	47.06	44.37	49.3	1.902	2.927
2	47.43	50.35	50.42	2.065	1.249
3	48.9	48.05	50.64	0.601	1.768
4	52.32	52.26	53.47	0.04243	0.9635
5	46.53	45.6	53.98	0.6576	6.463
6	46.99	50.87	51.87	2.744	2.4
7	47.49	51.55	58.57	2.871	7.389
8	47.41	47.63	48.63	0.1556	0.9063
9	48.37	51.03	50.15	1.881	0.3674
10	54.8	51.57	54.52	2.284	1.09

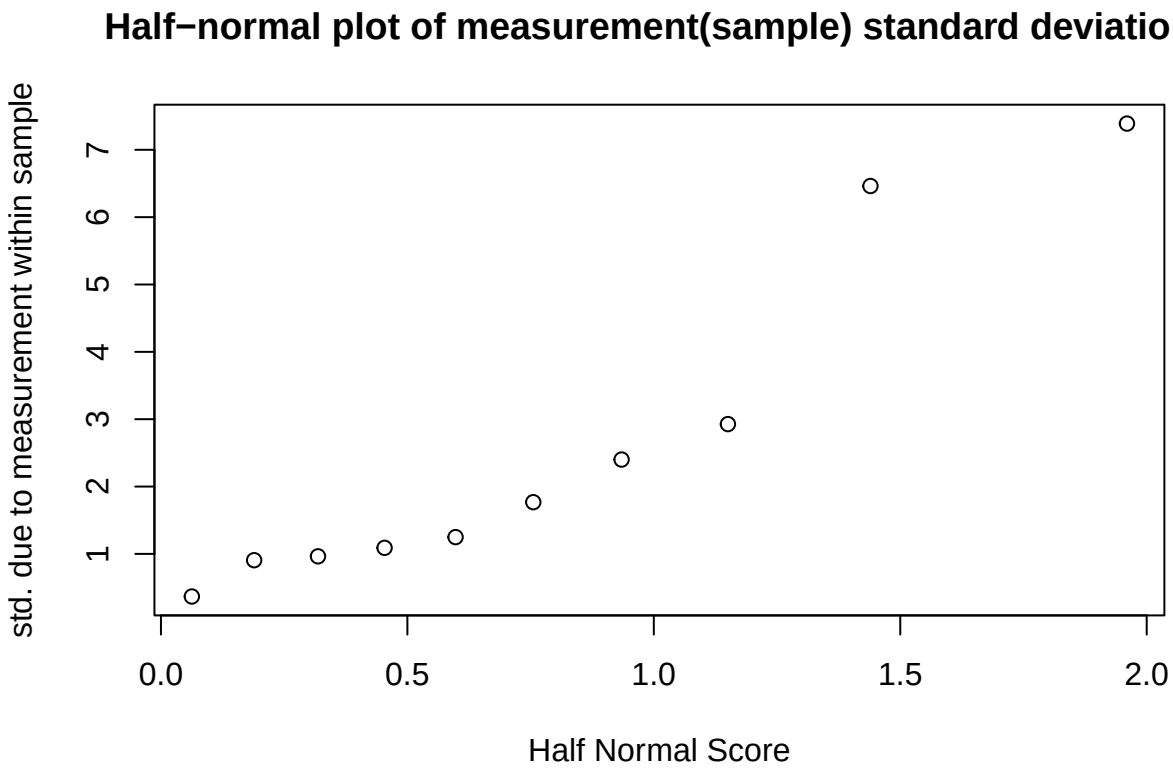
```
## half normal plot of sd1 (sample(trailer)) and sd2 (measurement(sample))

osd1 <- sort(sd1)
r <- c( 1: length(sd1))
zscore <- qnorm( ( ( r - .5 ) / length(sd1) +1 )/ 2)
plot( zscore, osd1,
      main = "Half-normal plot of sample(trailer) standard deviations",
      xlab = "Half Normal Score",
      ylab ="std. due to sample within trailer")
```




```
osd2 <- sort(sd2)
r <- c( 1: length(sd2))
zscore <- qnorm( ( ( r - .5 ) / length(sd2) +1 )/ 2)
plot( zscore, osd2,
      main = "Half-normal plot of measurement(sample) standard deviations",
```

```
      xlab = "Half Normal Score",
      ylab ="std. due to measurement within sample")
```



(Note:This is a subjective evaluate) From the half normal plots above, we can see that not all the observations align on a straight line, in particular trailer 9 (which is the 5^{th} observation when sorted by sd1) and trailer 5 (which is the 9^{th} observation when sorted by sd2). We can try removing them and repeat the analysis, which is implemented below. This data cleaning procedure appears to have fixed some of the inconsistency issue we saw earlier, especially between σ_a^2 estimated using MoM and REML. However, one thing to note is that the variance components do fluctuate quite a bit based on changes in the data, this suggest that we may need a larger sample size in order to get more consistent estimates of the variance components.

```
## observation 5 or 9 could be problematic based on the half normal plot, what happens if we remove it?

mod3_reml_adj <-lmer(value ~ 1 + (1|trailer)
                 + (1|trailer:sample),
                 data = raw_material[raw_material$trailer!=c(5,9),])
mod3_aov_adj <- aov( value ~ trailer +
                   trailer:sample,
                 data = raw_material[raw_material$trailer!=c(5,9),])

s3_adj <- summary(mod3_aov_adj)
ms_est<-s3_adj[[1]]$`Mean Sq`
sig2_est_adj <- var_est_3_stage(msA = ms_est[1],
                            msB = ms_est[2],
                            msC = ms_est[3])

print(sig2_est_adj)
```

```
##         B         A Residual
```

```
## 1 3.76908 2.091553 3.656744
```

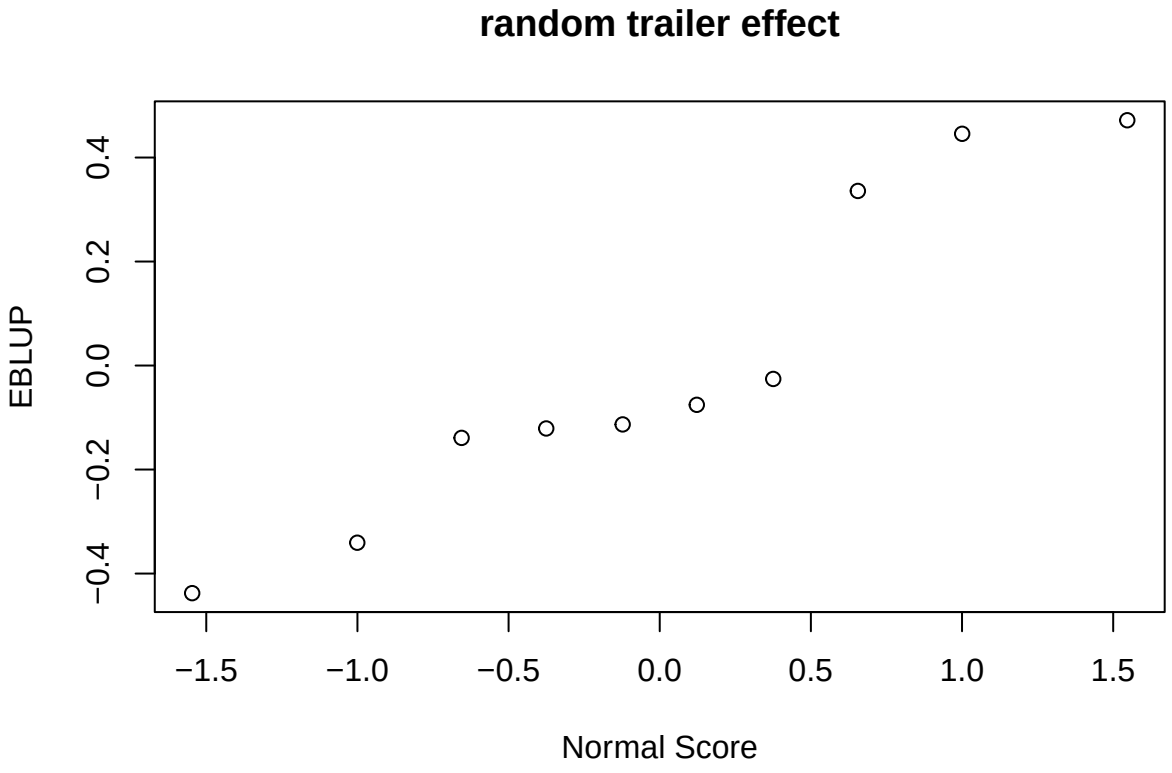
```
print(VarCorr(mod3_reml_adj),comp="Variance")
```

```
##  Groups         Name        Variance
##  trailer:sample (Intercept) 3.5069
##  trailer        (Intercept) 2.9831
##  Residual                   3.8969
```

(e) Calculate the EBLUPs for the random trailer effect and make a normal plot to check the normality assumption. What is your conclusion?

Estimated BLUP can be computed using the `ranef()` function in R. Applying this to the model with the full dataset, we don't see a fully straight line. This again could be due to the small sample size (10 trailers).

```
ranef_mod3<-ranef(mod3_reml)
qqnorm( ranef_mod3$trailer[[1]],
        main="random trailer effect",
        ylab="EBLUP",xlab ="Normal Score" )
```



Problem 4

Reanalyze the data from the golf experiment, presented in the Appendix of Chapter 4 (or dataset `rcb` in the `daewr` R package) using the `lmer` function. Check to see if you get the same P-values and conclusions shown in Section 4.7.

Ans: This is a mixed effects model which can be modeled by

$$y_{ijk} = \mu + \alpha_i + b_j + \epsilon_{ijk}$$

where μ is the overall effect, α_i is the fixed effect for tee height, and b_j is the random golfer effect. This can be implemented using `lmer` as :

```
library(lmerTest)
```

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

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

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

```
mod4 <- lmer(cdistance ~ 1+teehgt+(1|id)+(1|id:teehgt),
             data = rcb)
pander(anova(mod4))
```

Table 8: Type III Analysis of Variance Table with Satterthwaite's method

	Sum Sq	Mean Sq	NumDF	DenDF	F value	Pr(>F)
teehgt	795.7	397.9	2	16	5.854	0.01237

Note, there is something strange happening with R markdown that makes it not print the full result of the ANOVA table. Loading the `lmerTest` package seems to have fixed that.

Comparing this implementation with the one showed in Chapter 4, we can see that the conclusion of the results remain the same, that there is significant effect due to changes in tee height, when the F ratio is calculated with denominator as MS_{AB} instead of MS_E. In the summary table below, the result is shared under the section `Error: Id:teehgt`

```
mod4_aov <- aov(cdistance ~ teehgt + Error(id/teehgt), data = rcb)
summary(mod4_aov)
```

```
##
## Error: id
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  8 124741   15593
##
## Error: id:teehgt
##           Df Sum Sq Mean Sq F value Pr(>F)
## teehgt     2   1724   862.0   5.854 0.0124 *
## Residuals 16   2356   147.3
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Within
##            Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 108   7341   67.97
```