QUESTION 1

# RELATIVE EFFICIENCY OF LSD OVER RBD

Pranav Gopalkrishna, 1940223

2021-11-30

# AIM

Demonstration of whether Latin Square Design (LSD) is more efficient or not as compare to that of RCBD with the same experiment material using a suitable dataset / example and give your conclusion.

#========================

# DATA SET

This data set contains information about crop yield and other parameters affecting it, and is taken from an experiment made to understand the effect of soil density, fertilizer composition and irrigation on the crop yield.

setwd("~/Documents/Study/computerScience/programming/r/data/")  
myData = read.csv("cropYield.csv")  
head(myData)

## X yield block irrigation density fertilizer  
## 1 1 90 A control low N  
## 2 2 95 A control low P  
## 3 3 107 A control low NP  
## 4 4 92 A control medium N  
## 5 5 89 A control medium P  
## 6 6 92 A control medium NP

# DATA USED  
t = myData$fertilizer  
y = myData$yield

The treatment and response variabes will remain the same for both experimental designs. However, depending on the design, we will either choose one or two other factors whose effect on the response we want to correct for, in order to get a more accurate conclusion for the significance of the treatment’s effects on the variation in the response.

#========================

# FUNCTIONS USED TO CALCULATE ERROR MEAN SQUARES

# Correction factor  
cf = sum(y)^2 / length(y)

## Functions to calculate treament sum of squares

Regression sum of squares (for treatment or block) is calculated by summing the means of the squares of the sums of each of the regressor’s level’s replications, and then subtracting the correction factor from this value. So, for treatment mean square, we would first find the sum of the replications for each level at a time, square each of these sums, and divide each one by the number of replications in the level. This process is achieved using the following functions…

# Level mean square  
# (Used to find the mean of the squared sum of replications for each level)  
lms = function(level, regressor)  
{  
 sum = 0  
 n = 0  
 for(i in c(1:(length(y))))  
 {  
 if(regressor[i] == level)  
 {  
 sum = sum + y[i]  
 n = n + 1  
 }  
 }  
 return(sum^2 / n)  
}  
# Regression sum of squares  
rss = function(regressor)  
{  
 sum = 0  
 levels = unique(regressor)  
 for(level in levels)  
 {  
 sum = sum + lms(level, regressor)  
 }  
 return(sum - cf)  
}

## Total sum of squares

tss = 0  
for(i in y)  
{  
 tss = tss + i^2  
}  
tss = tss - cf

**NOTE**: Error sum of squares = Total sum of squares - (Sum of all regression sum of squares)

#=======================

# LATIN SQUARE DESIGN

In this design, we consider two blocking factors that may effect the variance in the response. This way, we can potentially minimise systematic error arising from two sources of variation other than the treatment itself. This tends to make latin square design more efficient than RBD or CRD.

b1 = myData$block  
b2 = myData$density  
data = data.frame(y, t, b1, b2)

Here, plot and soil density are chosen as the blocking factors, since they can potentially cause  
variation in crop yields.

# LINEAR REGRESSION MODEL

data = data.frame(y, t, b1, b2)  
model = lm(y~., data)  
summary(model)

##   
## Call:  
## lm(formula = y ~ ., data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -36.236 -13.260 -1.604 13.979 38.514   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 93.2639 5.8669 15.897 <2e-16 \*\*\*  
## tNP 12.7500 5.0809 2.509 0.0146 \*   
## tP 7.6667 5.0809 1.509 0.1362   
## b1B 4.2222 5.8669 0.720 0.4743   
## b1C 2.8889 5.8669 0.492 0.6241   
## b1D 3.7778 5.8669 0.644 0.5219   
## b2low -10.0417 5.0809 -1.976 0.0524 .   
## b2medium 0.8333 5.0809 0.164 0.8702   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 17.6 on 64 degrees of freedom  
## Multiple R-squared: 0.1654, Adjusted R-squared: 0.07416   
## F-statistic: 1.812 on 7 and 64 DF, p-value: 0.1001

We see that the error degrees of freedom is 65.

error\_df = 65

# ERROR MEAN SQUARE

# Error sum of squares...  
ess = tss - rss(t) - rss(b1) - rss(b2)  
# Error mean square...  
ems\_lsd = ess / error\_df  
ems\_lsd

## [1] 305.0184

#========================

# RANDOMISED BLOCK DESIGN

In this design, we will be choosing only one blocking factor, unlike two chosen for LSD. Hence, we account for only one variable’s effects on the response apart from the treatment, potentially making it less accurate than LSD. We will use the same blocking factors as LSD, but will make separate RBD models for each.

# LINEAR REGRESSION MODEL

# Multiple linear regression model for two regressors...  
model = lm(y~t + b1)  
summary(model)

##   
## Call:  
## lm(formula = y ~ t + b1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -33.167 -12.174 -2.903 13.028 41.583   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 90.194 5.220 17.277 <2e-16 \*\*\*  
## tNP 12.750 5.220 2.442 0.0173 \*   
## tP 7.667 5.220 1.469 0.1467   
## b1B 4.222 6.028 0.700 0.4861   
## b1C 2.889 6.028 0.479 0.6334   
## b1D 3.778 6.028 0.627 0.5330   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 18.08 on 66 degrees of freedom  
## Multiple R-squared: 0.09142, Adjusted R-squared: 0.02259   
## F-statistic: 1.328 on 5 and 66 DF, p-value: 0.2632

# Here, we see error DF as 67.  
error\_df = 67

## ERROR MEAN SQUARE FOR RBD WITH BLOCKING FACTOR 1 (b1)

# Error sum of squares...  
ess = tss - rss(t) - rss(b1)  
# Error mean square...  
ems\_rbd\_b1 = ess / error\_df  
ems\_rbd\_b1

## [1] 322.1575

## ERROR MEAN SQUARE FOR RBD WITH BLOCKING FACTOR 2 (b2)

# Error sum of squares...  
ess = tss - rss(t) - rss(b2)  
# Error mean square...  
ems\_rbd\_b2 = ess / error\_df  
ems\_rbd\_b2

## [1] 298.8155

#========================

# RELATIVE EFFICIENCY

This compares the precision of LSD to the precision of RBD. In other words, it compares how much lesser or greater the variation in experimental error is in LSD compared to RBD.

# LSD VS. RBD WITH BLOCKING FACTOR 1  
re\_b1 = 100\*(1/ems\_lsd)/(1/ems\_rbd\_b1)  
# Increase / descrease  
re\_b1 - 100

## [1] **5.619061**

# LSD VS. RBD WITH BLOCKING FACTOR 2  
re\_b2 = 100\*(1/ems\_lsd)/(1/ems\_rbd\_b2)  
# Increase / descrease  
re\_b2 - 100

## [1] **-2.033605**

As we can see, LSD is 5.6161% a more efficient than the RBD design using the 1st blocking factor. However, we see that LSD is 2.033605% less efficient than the RBD design using the 2nd blocking factor. This suggests that the first blocking factor is an ineffective blocking factor, since it seems to have a adverse effect on the variation in experimental error, increasing it rather tha minimising it. Note that greater efficiency in a model means that it takes smaller samples to achieve more accurate predictions and estimations (predictions about the significance of effects of the factors, and estimations about the linear regression model that gets created before the ANOVA test).

QUESTION 2

# TWO FACTOR DESIGN AND TWO-WAY ANOVA

Pranav Gopalkrishna

2021-11-30

# AIM

Explain how does the factorial experiment is different from two way ANOVA with respect to all the aspects using a suitable dataset(s) and draw your conclusion.

#========================

# DEFINITIONS

## TWO FACTOR DESIGN

This is an experimental design i.e. a method using which data is collected and organised. Two-factor design is a design wherein data is collected on two interacting treatments, and is collected not only for every level of each treatment, but for every possible combination of treatment levels among the two treatments. i.e. every possible interaction.

## TWO-WAY ANOVA

This is a statistical test of significance, that estimates the significance of the effect of two variables (separately, and if possible, with interaction) on the response variable. It involves a null hypothesis for each treatment separately and their interactions (if possible), and states that the mean responses for ever level of the treatment or treatment interaction is equal. To estimate significance, we use the F-statistic. A significant result means the null hypothesis is rejected.

#========================

# NOTES

Two-factor design uses two-way ANOVA as a statistical test for significance of factors. The only point to remember is that the interactions are always available and are exhaustive, when using this design.

QUESTION 3

# LSD MISSING VALUE ESTIMATION

Pranav Gopalkrishna, 1940223

2021-11-30

# AIM

Demonstrate the concept of missing plot technique of Latin Square Design (LSD) using a suitable example and give your interpretation. (Only one missing value case)

#========================

# FORMULA

X = [t(R’ + C’ + T’) - 2G’]/[(t - 2)(t - 2)]

# where...  
# X is the missing value's estimate  
# R' is the total of the available values in the row blocking factor.  
# C' is the total of the available values in the column blocking factor.  
# T' is the total of the available values in the treatment.  
# t is the number of levels in the treatment.  
# G' is the total of the available values in the data set.  
#========================

# DATA SET

This data set contains archived information about the weather in certain cities in South Africa, and is used to help understand weather patterns and effects of past and potential future rains on current rainfall.

setwd("/Users/pranav/Documents/Study/computerScience/programming/r/data/")  
data = read.csv("weatherSouthAfrica.csv")[c(2, 5, 22, 23)]  
head(data)

## Location Rainfall RainToday RainTomorrow  
## 1 Cape Town 0.6 High Low  
## 2 Cape Town 2.7 Medium Low  
## 3 Cape Town 4.0 Low Medium  
## 4 Cape Town 4.3 Low Low  
## 5 Cape Town 1.0 Medium Medium  
## 6 Cape Town 0.2 Low Low

tr = data$Location  
r = data$RainToday  
c = data$RainTomorrow  
y = data$Rainfall  
  
# Summary about the data  
summary(data.frame(as.factor(tr), as.factor(r), as.factor(c), y))

## as.factor.tr. as.factor.r. as.factor.c. y   
## Cape Town :10 High : 8 High : 8 Min. : 0.000   
## Durban :11 Low :13 Low :12 1st Qu.: 1.100   
## Johannesberg:11 Medium:11 Medium:12 Median : 2.700   
## Mean : 3.748   
## 3rd Qu.: 4.950   
## Max. :12.000   
## NA's :1

#========================

# MISSING VALUE ESTIMATION

## Checking how many missing values are there in the response

sum(is.na(y))

## [1] 1

## Locating the missing value

for(i in c(1:length(y)))  
{  
 if(is.na(y[i]))  
 {  
 r\_at\_X = r[i]  
 c\_at\_X = c[i]  
 tr\_at\_X = tr[i]  
 i\_at\_X = i  
 }  
}

## Calculating R’

R\_prime = 0  
for(i in c(1:length(y)))  
{  
 if(r[i] == r\_at\_X & i != i\_at\_X){R\_prime = R\_prime + y[i]}  
}  
R\_prime

## [1] 29.2

## Calculating C’

C\_prime = 0  
for(i in c(1:length(y)))  
{  
 if(c[i] == c\_at\_X & i != i\_at\_X){C\_prime = C\_prime + y[i]}  
}  
C\_prime

## [1] 33.3

## Calculating T’

T\_prime = 0  
for(i in c(1:length(y)))  
{  
 if(tr[i] == tr\_at\_X & i != i\_at\_X){T\_prime = T\_prime + y[i]}  
}  
T\_prime

## [1] 48.5

## Calculating G’

G\_prime = 0  
for(i in c(1:length(y)))  
{  
 if(i != i\_at\_X){G\_prime = G\_prime + y[i]}  
}  
G\_prime

## [1] 116.2

## Number of treatments

t = length(unique(tr))  
t

## [1] 3

## Estimating X (missing value)

X = (t\*(R\_prime + C\_prime + T\_prime) - 2\*G\_prime)/(t\*t)  
# t\*t was chosen instead of (t-1)(t-2) since t is too small, and was yielding grossly deviating esimations  
X

## [1] 11.17778

#========================

# CONCLUSIONS

We get the missing value esimate as 11.17778, which is much higher than the average, and close to the maximum. This may be due to the following factors:

# - Error in my calculations  
# - Small and potentially biased sample