**What Needs To Happen in our Simulation**

Before writing code, let us first understand what needs to happen in our simulation code. There are two tables that will be used for this simulation study. The Explanatory table that provides the demographic or background information about each study participate and the Response table that provides the outcome for each study participate.

|  |  |  |
| --- | --- | --- |
| Explanatory Table |  | Treatments that must be assigned completely at random to each of the study participants. |

The first step is to randomly assign the various treatments to each of the study participants.

|  |  |  |
| --- | --- | --- |
| The random assignment |  | Randomly assign treatments by “pulling from hat” without replacement |

The next step is to “run the experiment”. For our simulation study, this means we will use a Lookup table to assign the appropriate response for each person in our study.

|  |  |
| --- | --- |
| Experiment Data | Lookup Tables for “Running The Experiment” |

After all lookups have been completed, merge the response data to the Experiment Data as is shown here.

|  |  |
| --- | --- |
| Merge outcomes to Experiment Data | Final Experiment Data for analysis |

The appropriate analysis here is a t-test or equivalently an analysis of variance (ANOVA). The ANOVA approach will be used here because eventually a blocking variable will be included in our simulation study.

**Getting the Analysis Done in R**

The general syntax for fitting a model in R has the following form.

Response ~ Treatment

The following R code will be used to run the ANOVA model. The dataset being used by this function is called ExperimentData. The summary() function will produce the standard analysis of variance table.

outcomes <- aov(Improvement ~ Trt, data=ExperimentData)

summary(outcomes)



The following null and alternative hypothesis are being tested by this ANOVA analysis.

Hypotheses

HO : The Average Improvement is the SAME  
HA : The Average Improvement is DIFFERENT

P-Value: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

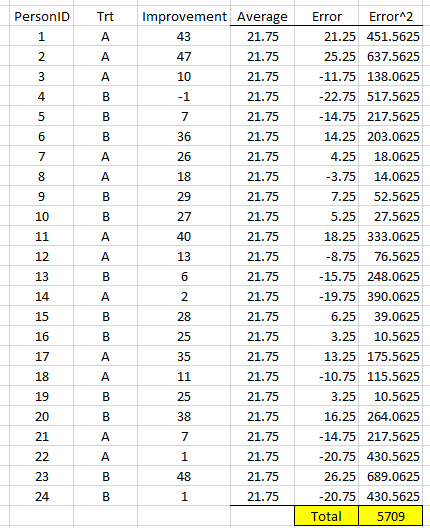
Decision: If p-value < 0.05, then have enough evidence to reject HO. Write a conclusion.

**Understanding the Decomposition of Error in ANOVA.**

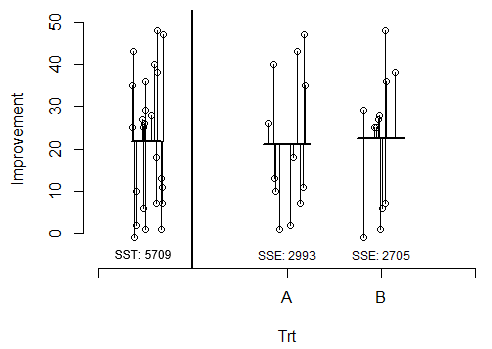
Consider the following discussion centered on the decomposition of error terms in a standard analysis of variance.

|  |  |
| --- | --- |
|  | Treatment Averages |

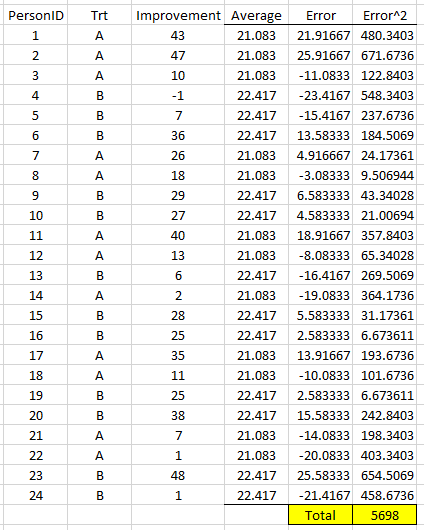
The “Sums of Squares Error - Total” is computed using the overall average. This calculation is being done ***assuming the treatments have no effect***.



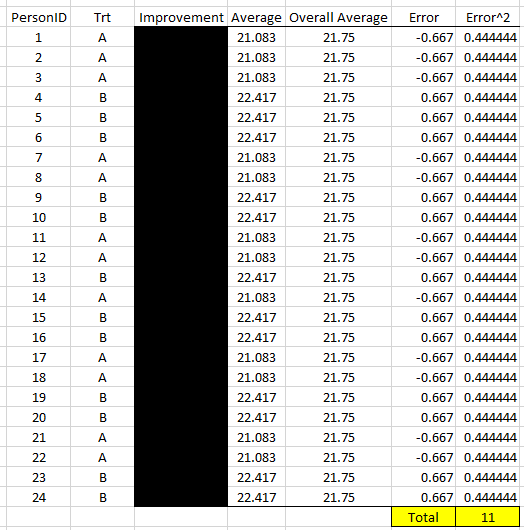
Visually, the sums of squares error – total is shown on the left. The sums of squares error is computing in a similar way except using the individual treatment means.



The “Sums of Squares Error” is computed using the individual treatment averages. This calculation is being done ***assuming the treatments have some effect***.



This Sums of Squares Error measures the amount of variation in the outcomes that *cannot* be explained by the treatments. This is in contrast to the “Sums of Squares Treatment” which is the amount of variation that *can* be explained by the treatments.



**Building the Simulation Study in R**

Reading the two data tables in R can be done using the following code. The file.choose() version below uses the Explorer/Finder window to locate the file to be read in.

Explanatory <- read.csv( file.choose() )

An alternative to using file.choose() is to provide the exact system path to the data to be read into R, e.g. “C:/Deskstop/Data.csv”.

Explanatory <- read.csv("*<exact system path to file>*")

View(Explanatory)

Response<- read.csv("*<exact system path to file>*")

View(Response)

The following can be used to make the random assignment of the treatments to each person in our study.

RandomAssignments <- sample(c(“A”,”A”,"A","A","A","A","A","A","A","A","A","A","B","B","B","B","B","B","B","B","B","B","B","B"),replace=FALSE)

Building a data.frame in R to merge the PersonID and the random assignments obtained above.

ExperimentData <- data.frame(‘PersonID’ = Explanatory$PersonID, ‘Treatment’ = RandomAssignments)

|  |  |
| --- | --- |
| Gather PersonID and RandomAssignment and merge together | Experiment Data |

Next, we will use the left\_join() function in the dplyr package in R to conduct the necessary lookup to attach the appropriate responses to each person in the study. If you have not previously done so, you will need to install the dplyr package. This can be done using RStudio’s Package installer.

library(dplyr)

ExperimentData <- left\_join(ExperimentData,Response,

by=c(‘PersonID’ = ‘PersonID’,

‘Treatment’ = ‘Treatment’)

)

|  |  |  |  |
| --- | --- | --- | --- |
| Experiment Data  *before lookup* | Response |  | Experiment Data  *after lookup* |

The ANOVA table for this analysis can be obtained as follows.

outcomes <- aov(Improvement ~ Treatment, data=ExperimentData)

summary(outcomes)



A simulation study will require us to repeatedly run the code above. The most succinct way of doing this in R is to place the code into a function as shown below. The outcome that will be obtained on each iteration is the “Sums of Squares Error” term from the ANOVA table.

MySimulation <- function(){

RandomAssignments <- sample(c("A","A","A","A","A","A","A","A","A","A","A","A","B","B","B","B","B","B","B","B","B","B","B","B"),replace=FALSE)

ExperimentData <- data.frame(‘PersonID’ = Explanatory$PersonID, ‘Treatment’= RandomAssignments)

ExperimentData <- left\_join(ExperimentData,Response,by=c(‘PersonID’ = ‘PersonID’,’Treatment’ = ‘Treatment’))

outcomes <- aov(Improvement ~ Treatment, data=ExperimentData)

sse.value <- summary(outcomes)[[1]][2,2]

return(sse.value)

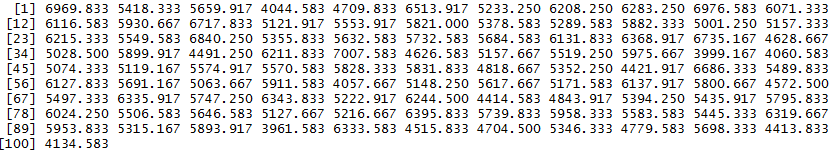
}

The replicate() function in R can be used to run repeated instances of a function. This will work well for a simulation study as the function written above needs to be run repeatedly. This will allow us to understand how much the random assignment influences the “Sums of Squares Error” term from each ANOVA analysis. A total of 100 repeated outcomes are obtained here.

SimulationOutcomes<-data.frame(‘SSEValues’ = replicate( 100, MySimulation() ) )

To see the “Sums of Squared Error” values from each iteration of your simulation, you can simply type the object’s name in R.

SimulationOutcomes

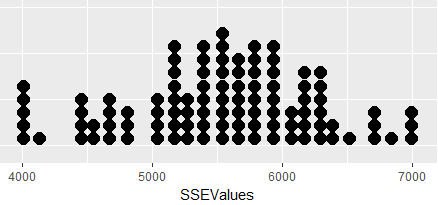


Basic summaries, e.g. graphs, summary statistics, etc., can be obtained easily in R. Here the ggplot() function (from the ggplot2 package) is being used to create a dotplot and the summarize() function (from the dplyr package) is being used to obtain the mean and standard deviation of the distribution of “Sums of Squares Error” values from the 100 simulated iterations.

The following ggplot() function can be used to obtain a simple dotplot of these 100 simulated outcomes.

library(ggplot2)

ggplot(SimulationOutcomes,aes(x=SSEValues)) + geom\_dotplot()



The summarize() function which is part of the dplyr library can be used to obtain basic summary statistics of your simulated outcomes.

> summarize(SimulationOutcomes,’Mean’ = mean(SSEValues))

Mean

1 5532.738

> summarize(SimulationOutcomes,’SD’ = sd(SSEValues))

SD

1 716.0141

**Modification for Blocking**

When an experiment includes a blocking variable, the randomization of treatments to subjects will be modified. In particular, the randomization is restricted in the sense that random assignment is done within each block.

|  |  |
| --- | --- |
| Define blocking variable | Form the blocks |

#Sorting by Initial Cholesterol

Explanatory\_ByCholesterol<-arrange(Explanatory, InitialCholesterol)

#Adding in a variable to identify the block

Explanatory\_ByCholesterol<-mutate(Explanatory\_ByCholesterol,’Block’=rep(1:12,each=2))

The random assignment of treatments to subjects must take place within each block in a randomized block design; this is in contrast to the assignment of treatments in a completely randomized design.

|  |  |
| --- | --- |
| Treatment assignments in a block design | Completely randomized design |

The following code can be used to do the random assignment of treatments within each block, merge the random assignments with PersonID and then synthetically run the experiment using the lookup table.

#Getting random assignments under blocking, i.e. 12 blocks with two treatments in each block

RandomAssignments <- unlist(replicate(12,sample(c("A","B"), replace=FALSE), simplify = FALSE))

#Merging assignments to each person

ExperimentData <- data.frame(‘PersonID’ = Explanatory\_ByCholesterol$PersonID, ‘Block’ = Explanatory\_ByCholesterol$Block,’Treatment’= RandomAssignments)

#Running the Experiment

ExperimentData <- left\_join(ExperimentData,Response,by=c(‘PersonID’ = ‘PersonID’,’Treatment’ = ‘Treatment’))

The code above creates the ExperiementData which will be used to conduce the analysis.

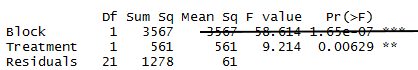
|  |  |  |
| --- | --- | --- |
| Experiment Data  *before lookup* | Lookup table to run the experiment | Experiment Data  *after lookup* |

The following code conducts a one-way analysis of variance with the inclusion of the blocking variable.

#Anayzing the Data

outcomes <- aov(Improvement ~ Block + Treatment, data=ExperimentData)

summary(outcomes)



The following null and alternative hypothesis are being tested by this ANOVA analysis.

Hypotheses

HO : The Average Improvement is the SAME  
HA : The Average Improvement is DIFFERENT

P-Value: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Decision: If p-value < 0.05, then have enough evidence to reject HO. Write a conclusion.

|  |
| --- |
| What happens when the analysis ignores the blocking variable? |
| #Anayzing the Data  outcomes <- aov(Improvement ~ Treatment, data=ExperimentData)  summary(outcomes)    Notice that the Sums of Squares for the blocking variable have been dumped into the Sums of Squares for Residuals, i.e. the unexplained variation. The result is that the test of significance no longer can detect the difference between Treatment A and B. |

Next, we will place the code for the block simulation into a function so that this code can be easily run on repeated iterations.

MyBlockSimulation <- function(){

RandomAssignments <- unlist(replicate(12,sample(c("A","B"), replace=FALSE), simplify = FALSE))

ExperimentData <- data.frame(‘PersonID’ = Explanatory\_ByCholesterol$PersonID, ‘Block’ = Explanatory\_ByCholesterol$Block,’Treatment’= RandomAssignments)

ExperimentData <- left\_join(ExperimentData,Response,by=c(‘PersonID’ = ‘PersonID’,’Treatment’ = ‘Treatment’))

outcomes <- aov(Improvement ~ Block + Treatment, data=ExperimentData)

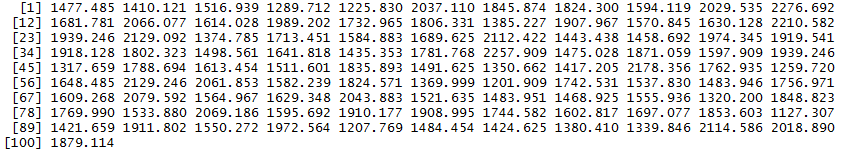
sse.value <- summary(outcomes)[[1]][3,2]

return(sse.value)

}

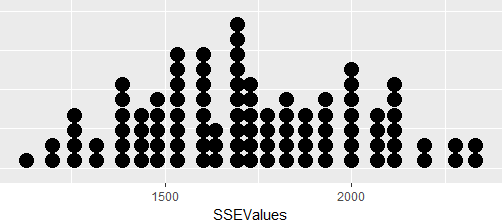
The replicate function is again used to run 100 repeated iterations.

BlockSimulationOutcomes<-data.frame(‘SSEValues’ = replicate( 100, MyBlockSimulation() ) )



Plotting these outcomes and obtaining basic summary statistics for these outcomes.

ggplot(BlockSimulationOutcomes,aes(x=SSEValues)) + geom\_dotplot()



> summarize(BlockSimulationOutcomes,’Mean’ = mean(SSEValues))

Mean

1 1719.937

> summarize(BlockSimulationOutcomes,’SD’ = sd(SSEValues))

SD

1 283.9971

The following can be used to plot all the outcomes at once for comparison purposes.

#Putting the two simulation outcomes together

SimulationOutcomes<-mutate(SimulationOutcomes,’Type’ = "CRD")

BlockSimulationOutcomes<-mutate(BlockSimulationOutcomes,’Type’ = "Block")

AllSimulationOutcomes <- bind\_rows(SimulationOutcomes, BlockSimulationOutcomes)

#ggplot to compare

ggplot(AllSimulationOutcomes, aes(x=SSEValues)) +

geom\_dotplot(binwidth=100) +

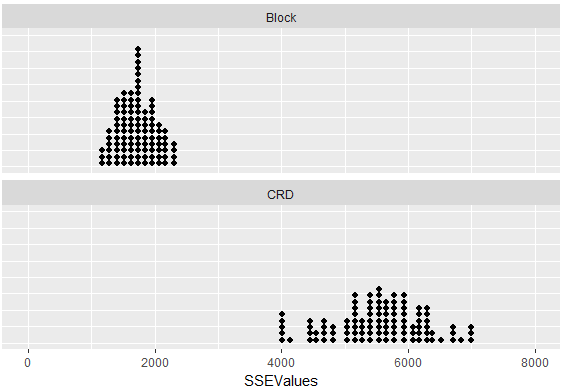
theme(axis.title.y=element\_blank(),

axis.text.y=element\_blank(),

axis.ticks.y=element\_blank()) +

scale\_x\_continuous(limits=c(0,8000)) +

facet\_wrap(~ Type, ncol=1)



Task: Set up a simulation study to investigate the effect of blocking by Gender. Obtain a distribution SSE Values and compare against the values obtained here.