Contents

1. Randomized Block Experimental Design

For this assignment I chose to submit my code for the **randomized block** experiment design.

First, I loaded data from a csv input file into a data frame using the following code:

```
#Read Data

df2=read.table("fastfood-2.csv", header=TRUE, sep=",") #Read data from the file into a data frame

df2
```

It provided the following output in the console window, which showed that the df2 was successfully created and populated with data from the file:

```
Console   Terminal ×
E:/Dropbox/RU DataScience/MSDS650/Week2/Assignment/
>
>
> df2=read.table("fastfood-2.csv", header=TRUE, sep=",") #Read data from the file into a data frame
> df2
  Item1 Item2 Item3
1    31    27    24
2    31    28    31
3    45    29    46
4    21    18    48
5    42    36    46
6    32    17    40
> |
```

In the next step, I created a response variable vector using the following command:

```
#Create Response vector

r = c(t(as.matrix(df2))) # concatenate response data into a single vector

        #as.matrix() function returns a vector of cell values; function t() returns the transpose of the argument

        #and it is concatenated into vector r

r
```

The output is shown below:

```
Console   Terminal ×
E:/Dropbox/RU DataScience/MSDS650/Week2/Assignment/
2    31    28    31
3    45    29    46
4    21    18    48
5    42    36    46
6    32    17    40
> #Create Response vector
> r = c(t(as.matrix(df2))) # concatenate response data into a single vector
>                          #as.matrix() function returns a vector of cell values; function t() returns the transpose of the argument
>                          #and it is concatenated into vector r
> r
 [1] 31 27 24 31 28 31 45 29 46 21 18 48 42 36 46 32 17 40
> |
```

Then I created variables for the treatment factors:

```
#New variables for Treatment factors
f = c("Item1", "Item2", "Item3")  # treatment levels
k = 3                             # number of treatment levels
n = 6                             # number of control blocks
```

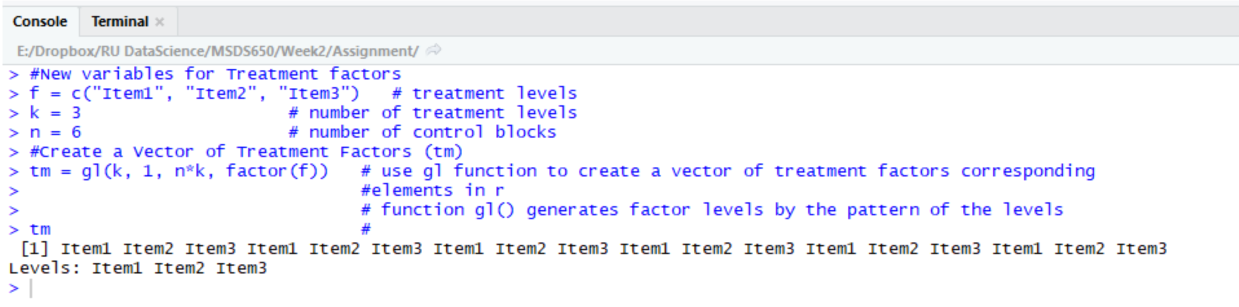
Which had an expected output:

```
Console   Terminal ×
E:/Dropbox/RU DataScience/MSDS650/Week2/Assignment/
6    32    17    40
> #Create Response vector
> r = c(t(as.matrix(df2))) # concatenate response data into a single vector
>                          #as.matrix() function returns a vector of cell values; function t() returns the transpose of the argument
>                          #and it is concatenated into vector r
> r
 [1] 31 27 24 31 28 31 45 29 46 21 18 48 42 36 46 32 17 40
> #New variables for Treatment factors
> f = c("Item1", "Item2", "Item3")    # treatment levels
> k = 3                             # number of treatment levels
> n = 6                             # number of control blocks
> |
```

As a next step, I used the gl() function to create a vector of treatment factors corresponding to responses in r, using the following command:

```
#Create a Vector of Treatment Factors (tm)
tm = gl(k, 1, n*k, factor(f))  # use gl function to create a vector of treatment factors corresponding
                       #elements in r
                       # function gl() generates factor levels by the pattern of the levels
tm                      #
```

The sample output:



```
Console   Terminal ×
E:/Dropbox/RU DataScience/MSDS650/Week2/Assignment/
> #New variables for Treatment factors
> f = c("Item1", "Item2", "Item3")   # treatment levels
> k = 3                          # number of treatment levels
> n = 6                          # number of control blocks
> #Create a Vector of Treatment Factors (tm)
> tm = gl(k, 1, n*k, factor(f))   # use gl function to create a vector of treatment factors corresponding
>                                 #elements in r
>                                 # function gl() generates factor levels by the pattern of the levels
> tm                              #
 [1] Item1 Item2 Item3 Item1 Item2 Item3 Item1 Item2 Item3 Item1 Item2 Item3 Item1 Item2 Item3 Item1 Item2 Item3
Levels: Item1 Item2 Item3
>
```

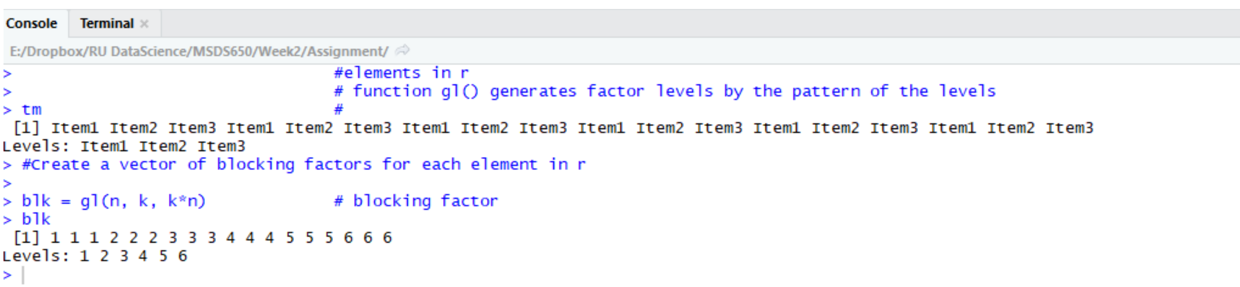
I created the last vector required to analyze variance – vector of blocking factors corresponding each response in r – using the following command:

```
#Create a vector of blocking factors for each element in r

blk = gl(n, k, k*n)          # blocking factor

blk
```

The screenshot below shows that blk was created successfully.



```
Console   Terminal ×
E:/Dropbox/RU DataScience/MSDS650/Week2/Assignment/
>                                 #elements in r
>                                 # function gl() generates factor levels by the pattern of the levels
> tm                              #
 [1] Item1 Item2 Item3 Item1 Item2 Item3 Item1 Item2 Item3 Item1 Item2 Item3 Item1 Item2 Item3 Item1 Item2 Item3
Levels: Item1 Item2 Item3
> #Create a vector of blocking factors for each element in r
>
> blk = gl(n, k, k*n)             # blocking factor
> blk
 [1] 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5 6 6 6
Levels: 1 2 3 4 5 6
>
```

Finally, I used the ***aov()*** function to describe the r response to both treatment factor tm and the block control blk. The following code was used to create the model, and then to display its results:

```
#Use aov function to describe the r response by both treatment factor tm and the block control blk

av= aov(r~ tm+blk)

#Display the results

summary(av)
```

Output:

```
Console   Terminal ×
E:/Dropbox/RU DataScience/MSDS650/Week2/Assignment/
Levels: 1 2 3 4 5 6
> #Use aov function to describe the r response by both treatment factor tm and the block control blk
> av= aov(r~ tm+blk)
>
> #Display the results
>
> summary(av)
            Df Sum Sq Mean Sq F value Pr(>F)
tm           2  538.8  269.39   4.959 0.0319 *
blk          5  559.8  111.96   2.061 0.1547
Residuals   10  543.2   54.32
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

Resulting ANOVA table and its interpretation:

```
            Df Sum Sq Mean Sq F value Pr(>F)
tm           2  538.8  269.39   4.959 0.0319 *
blk          5  559.8  111.96   2.061 0.1547
Residuals   10  543.2   54.32
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The calculations show the variation between the groups are significant and that the p-value for this experiment is 0.0319, which is less that the 0.05 significance level. It means that there is enough evidence to reject the null hypothesis, stating that the mean sales in the fast food restaurant are equal for the new menu items, in favor of the alternative hypothesis (mean sales are different).

Script window:

```r
#MSDS650 Week 2 Assignment1: Experimental Design  Weakly,Natalia

rm(list=ls()) #Clear the environment
setwd("E:/Dropbox/RU DataScience/MSDS650/Week2/Assignment") #Set working directory for the assignemnt
getwd() #Check working directory

########################################
##### Randomized block design ##########
#######################################

#6 restaurants test 3 menu items, one item per week

#Read Data
df2=read.table("fastfood-2.csv", header=TRUE, sep=",") #Read data from the file into a data frame
df2

#Create Response vector
r = c(t(as.matrix(df2))) # concatenate response data into a single vector
                        #as.matrix() function returns a vector of cell values; function t() returns the transpose of the argument
                        #and it is concatenated into vector r
r

#New variables for Treatment factors
f = c("Item1", "Item2", "Item3")   # treatment levels
k = 3                    # number of treatment levels
n = 6                    # number of control blocks


#Create a Vector of Treatment Factors (tm)
tm = gl(k, 1, n*k, factor(f))   # use gl function to create a vector of treatment factors corresponding
                                #elements in r
                                # function gl() generates factor levels by the pattern of the levels
tm                              #

#Create a vector of blocking factors for each element in r

blk = gl(n, k, k*n)             # blocking factor
blk

#Use aov function to describe the r response by both treatment factor tm and the block control blk
av= aov(r~ tm+blk)

#Display the results

summary(av)
```

2. Questions

a) What does an Analysis of Variance tell you? What types of questions does it answer?

Analysis of Variance is a statistical tool that allows researchers to differentiate experimental treatment results by comparing the mean of two or more subsets of data. ANOVA tests general rather than specific differences among means. It tests the null hypothesis that all means are equal using F-test, or a ratio of variances. In case of one-way ANOVA, F = variation between sample means/variation within the samples. When the null hypothesis is rejected it means that at least one mean is different from at least one other mean, and additional tests might be needed to establish which specific means are different.

ANOVA can answer various questions for which we can compare means for subsets of experimental or observational data. Here are some typical examples of such questions:

- Is the new drug/medical procedure working better than an existing treatment protocol?
- What strategy (A, B, C) will make a company more profitable?
- Are there any differences in high school GPA by grade level and gender?
- What is more important for increasing the employees job satisfaction (flexible work schedule, additional vacation time, or additional insurance benefits)?

b) What then is the significance of experimental design?

Overall, experimental design is extremely important for ensuring validity of an experiment. In addition, statistical tools that we can use to analyze the data depend on the experimental design. Depending on the design type (subgroups organization (completely randomized design, randomized block design, or factorial design), within-the-subjects and between-the-subjects design) we can choose between one-way ANOVA (comparison of means of three or more independent groups), one-way repeated measures ANOVA (comparison of means of three or more within-the-subjects variables), factorial ANOVA (examining multiple independent variables), or other mixed models.