Factorial_Contrast

Jimmy G. Moore

Sept. 12, 2018

Problem 1

Consider the 2^6 design in eight blocks of eight runs each with ABCD,ACE, and ABEF as the independent effects chosen to be confounded with blocks. Generate the design.

```
#set up matrix of all 64 combinations of levels,
##Also record their mod 2 of the linear contrast of confounding Interactions
orthoMatrix = matrix(0,nrow = 64,ncol = 9)
colnames(orthoMatrix) = c("f","e","d","c","b","a","L1","L2","L3")

rowCount= 1
for(f in c(0,1)){
  for(e in c(0,1)){
    for(d in c(0,1)){
      for(c in c(0,1)){
        for(b in c(0,1)){
          for(a in c(0,1)){
            orthoMatrix[rowCount,1:6] = c(f,e,d,c,b,a)

            #Check linear contrast mod 2
            orthoMatrix[rowCount,7] = (a+b+c+d)%%2
            orthoMatrix[rowCount,8] = (a+c+e)%%2
            orthoMatrix[rowCount,9] = (a+b+e+f)%%2
            rowCount = rowCount +1
          }
        }
      }
    }
  }
}
```

Now that we have our orthogonal matrix set up we will display the blocks as they related to linear contrasts.

```
#block 1
block1 = orthoMatrix[which(orthoMatrix[,7]==0 &
                             orthoMatrix[,8]==0&
                             orthoMatrix[,9]==0),]
block1
```

```
##      f e d c b a L1 L2 L3
## [1,] 0 0 0 0 0 0  0  0  0
## [2,] 0 0 1 1 1 1  0  0  0
## [3,] 0 1 0 1 1 0  0  0  0
## [4,] 0 1 1 0 0 1  0  0  0
## [5,] 1 0 0 1 0 1  0  0  0
## [6,] 1 0 1 0 1 0  0  0  0
## [7,] 1 1 0 0 1 1  0  0  0
## [8,] 1 1 1 1 0 0  0  0  0
```

```
#block 2
block2 = orthoMatrix[which(orthoMatrix[,7]==1 &
                           orthoMatrix[,8]==0&
                           orthoMatrix[,9]==0),]
block2
```

```
##      f e d c b a L1 L2 L3
## [1,] 0 0 0 1 1 1  1  0  0
## [2,] 0 0 1 0 0 0  1  0  0
## [3,] 0 1 0 0 0 1  1  0  0
## [4,] 0 1 1 1 1 0  1  0  0
## [5,] 1 0 0 0 1 0  1  0  0
## [6,] 1 0 1 1 0 1  1  0  0
## [7,] 1 1 0 1 0 0  1  0  0
## [8,] 1 1 1 0 1 1  1  0  0
```

```
#block 3
block3 = orthoMatrix[which(orthoMatrix[,7]==0 &
                           orthoMatrix[,8]==1&
                           orthoMatrix[,9]==0),]
block3
```

```
##      f e d c b a L1 L2 L3
## [1,] 0 0 0 0 1 1  0  1  0
## [2,] 0 0 1 1 0 0  0  1  0
## [3,] 0 1 0 1 0 1  0  1  0
## [4,] 0 1 1 0 1 0  0  1  0
## [5,] 1 0 0 1 1 0  0  1  0
## [6,] 1 0 1 0 0 1  0  1  0
## [7,] 1 1 0 0 0 0  0  1  0
## [8,] 1 1 1 1 1 1  0  1  0
```

```
#block 4
block4 = orthoMatrix[which(orthoMatrix[,7]==0 &
                           orthoMatrix[,8]==0&
                           orthoMatrix[,9]==1),]
block4
```

```
##      f e d c b a L1 L2 L3
## [1,] 0 0 0 1 0 1  0  0  1
## [2,] 0 0 1 0 1 0  0  0  1
## [3,] 0 1 0 0 1 1  0  0  1
## [4,] 0 1 1 1 0 0  0  0  1
## [5,] 1 0 0 0 0 0  0  0  1
## [6,] 1 0 1 1 1 1  0  0  1
## [7,] 1 1 0 1 1 0  0  0  1
## [8,] 1 1 1 0 0 1  0  0  1
```

```
#block 5
block5 = orthoMatrix[which(orthoMatrix[,7]==1 &
                           orthoMatrix[,8]==1&
                           orthoMatrix[,9]==0),]
block5
```

```
##      f e d c b a L1 L2 L3
## [1,] 0 0 0 1 0 0  1  1  0
## [2,] 0 0 1 0 1 1  1  1  0
## [3,] 0 1 0 0 1 0  1  1  0
## [4,] 0 1 1 1 0 1  1  1  0
## [5,] 1 0 0 0 0 1  1  1  0
## [6,] 1 0 1 1 1 0  1  1  0
## [7,] 1 1 0 1 1 1  1  1  0
## [8,] 1 1 1 0 0 0  1  1  0
```

```
#block 6
block6 = orthoMatrix[which(orthoMatrix[,7]==1 &
                           orthoMatrix[,8]==0&
                           orthoMatrix[,9]==1),]
block6
```

```
##      f e d c b a L1 L2 L3
## [1,] 0 0 0 0 1 0  1  0  1
## [2,] 0 0 1 1 0 1  1  0  1
## [3,] 0 1 0 1 0 0  1  0  1
## [4,] 0 1 1 0 1 1  1  0  1
## [5,] 1 0 0 1 1 1  1  0  1
## [6,] 1 0 1 0 0 0  1  0  1
## [7,] 1 1 0 0 0 1  1  0  1
## [8,] 1 1 1 1 1 0  1  0  1
```

```
#block 7
block7 = orthoMatrix[which(orthoMatrix[,7]==0 &
                            orthoMatrix[,8]==1&
                            orthoMatrix[,9]==1),]
block7
```

```
##      f e d c b a L1 L2 L3
## [1,] 0 0 0 1 1 0  0  1  1
## [2,] 0 0 1 0 0 1  0  1  1
## [3,] 0 1 0 0 0 0  0  1  1
## [4,] 0 1 1 1 1 1  0  1  1
## [5,] 1 0 0 0 1 1  0  1  1
## [6,] 1 0 1 1 0 0  0  1  1
## [7,] 1 1 0 1 0 1  0  1  1
## [8,] 1 1 1 0 1 0  0  1  1
```

```
#block 8
block8 = orthoMatrix[which(orthoMatrix[,7]==1 &
                           orthoMatrix[,8]==1&
                           orthoMatrix[,9]==1),]
block8
```

```
##      f e d c b a L1 L2 L3
## [1,] 0 0 0 0 0 1  1  1  1
## [2,] 0 0 1 1 1 0  1  1  1
## [3,] 0 1 0 1 1 1  1  1  1
## [4,] 0 1 1 0 0 0  1  1  1
## [5,] 1 0 0 1 0 0  1  1  1
## [6,] 1 0 1 0 1 1  1  1  1
## [7,] 1 1 0 0 1 0  1  1  1
## [8,] 1 1 1 1 0 1  1  1  1
```

Variance Stabilizing Example

A semiconductor manufacturer has developed three different methods for reducing particle counts on wafers. All three methods are tested on five different wafers and the after treatment paricle count obtained. The data is shown below:

```
#Enter data
trt1 = c(31,10,21,4,1)
trt2 = c(62,40,24,30,35)
trt3 = c(53,27,120,97,68)
semiCondData = data.frame(method = factor(c(rep(1,5),rep(2,5),rep(3,5))),
                          count = c(trt1,trt2,trt3))
```

Part A) Do all Methods have the same effect on mean particle count?

Essentially this question is asking us to do a global F test on the data to test our global hypothesis. Denoting mean partical count for a specific method as μ_i,our null and alternative hypothesis in this case is:

$$H_0 : \mu_1 = \mu_2 = \mu_3$$

$$H_a : u_i \neq \mu_j \; for \; at \; least \; one \; pair \; (i, j)$$

We will now generate our test statistic for the hypothesis of no differences in method means using the following equation

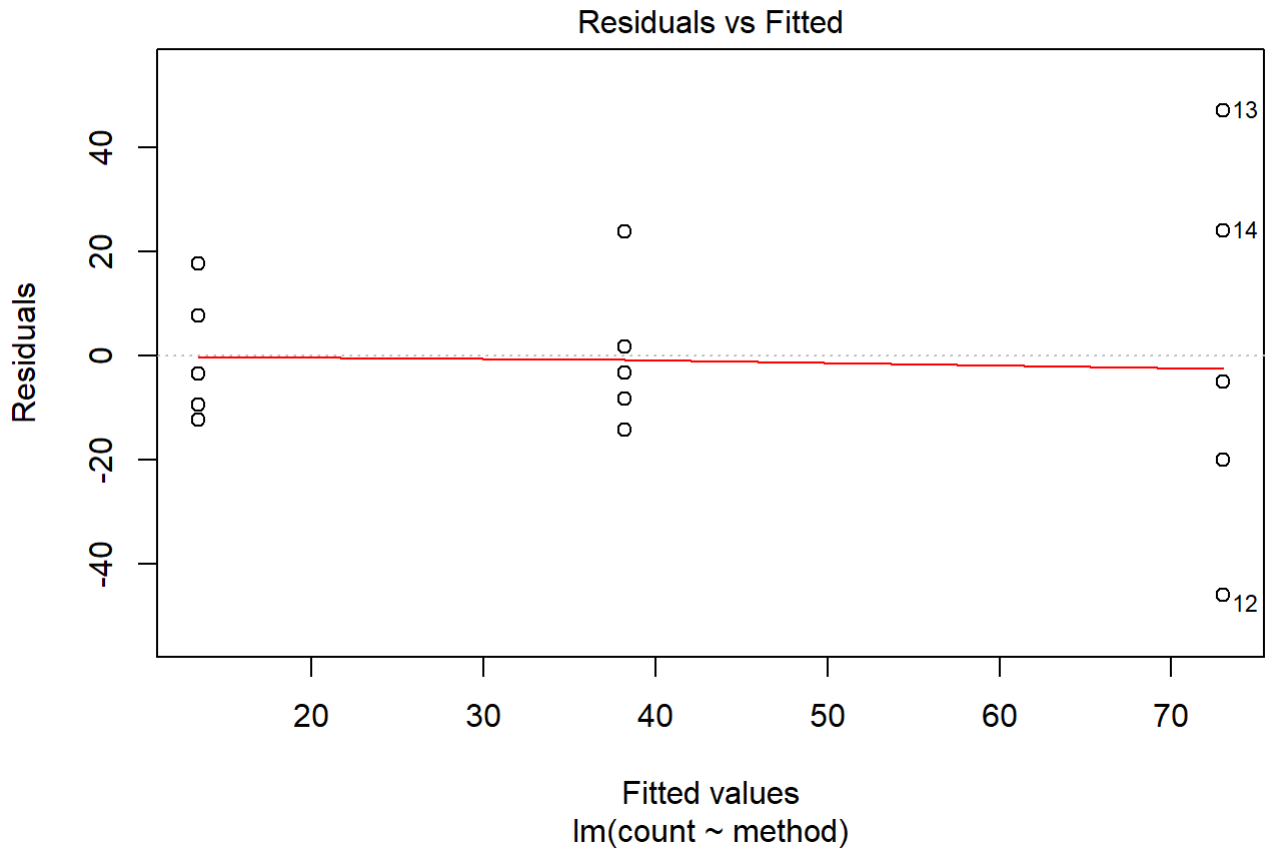
$$F_0 = \frac{MS_{trt}}{MS_e}$$

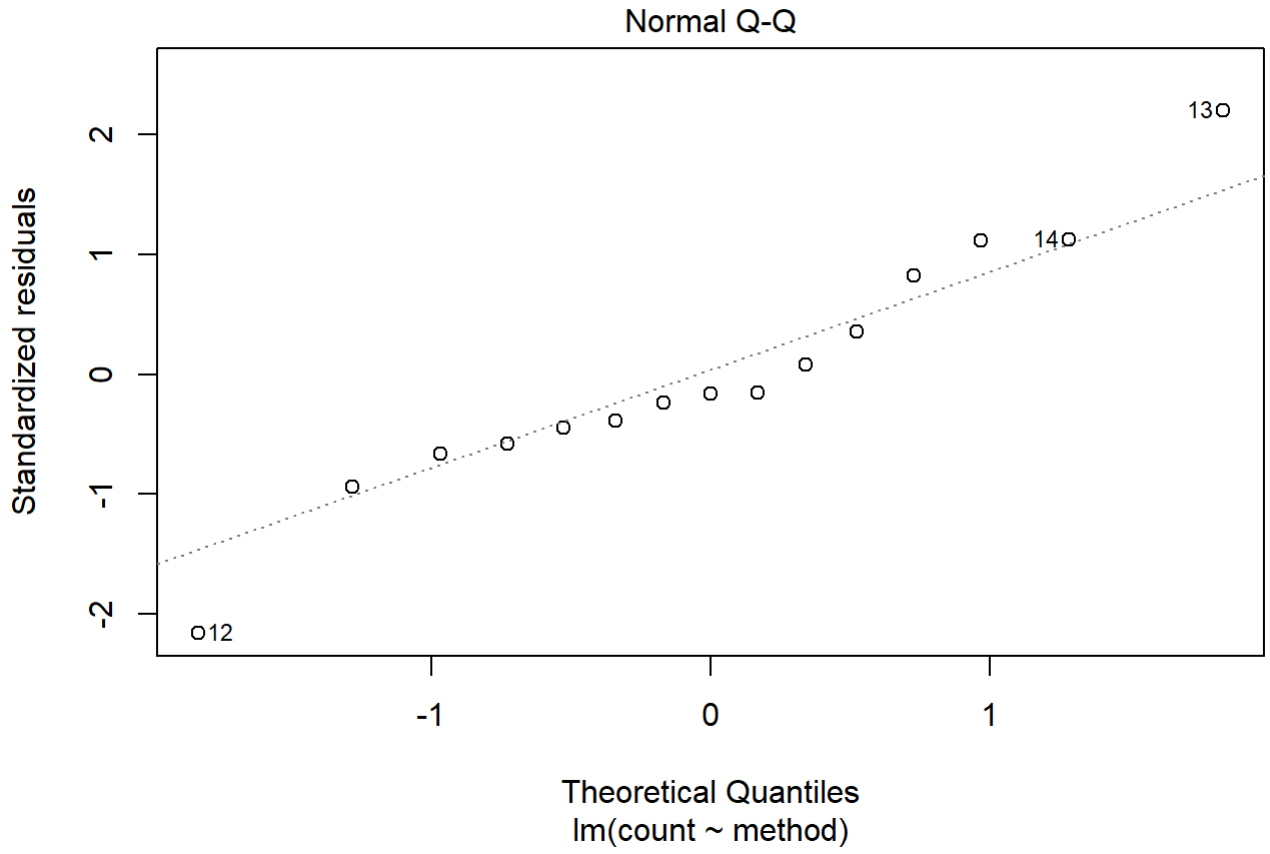
```
anova(aov(count~method,data = semiCondData))
```

```
## Analysis of Variance Table
##
## Response: count
##           Df Sum Sq Mean Sq F value  Pr(>F)
## method     2 8963.7  4481.9  7.9138 0.00643 **
## Residuals 12 6796.0   566.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here we can see that we observe a test statistic of 7.9138 and a P-value of 0.006. Therefore, at the α = 0.05 level there is sufficient evidence to reject the null hypothesis and conclude that not all methods have the same effect on mean particle count.

Part B)


Residuals vs Fitted
Residuals
Fitted values
lm(count ~ method)
13
14
12


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(count ~ method)
13
14
12

The first plot shows us the relationship between residuals and predicted response. We can see the the the residuals and the fitted values are likely independent which is good. However, there is concern about the heteroskedasticity in our residuals as our fitted values increase. Also our QQ plot indicates that there may be some moderate departures from normality, however this assumption is rather flexible so I would say based on my naked eye, the assumption of normality is upheld. A Kolmogorov-Smirnov test could be conducted to validate or refute my judgement.

Part C)

Looking at the diagnostic plots in Part B, it is easy to conclude that our assumption of homoskedasticity is violated in our original ANOVA model. Therefore, we want to find a transformation on `count` that yields a constant variance.

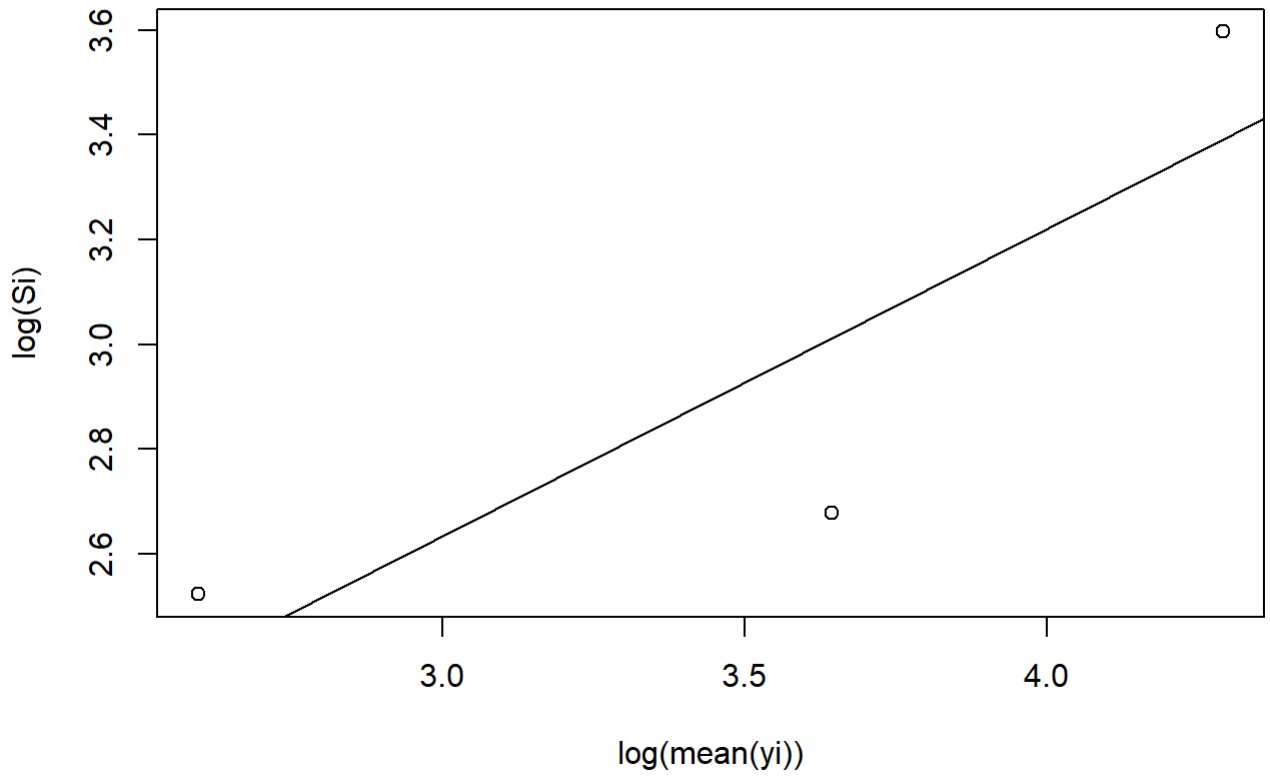
The standard deviation of our response is proportional to a power of the mean of y:

$$\sigma_y \propto \mu^{\alpha}$$

Therefore, our transformed variable ($y*$) will be derived by $y* = y^{\lambda}$ such that $\lambda = 1 - \alpha$.

Generally we want to keep the strength of our transformation (i.e. α value) as low as possible to minimize the effect on our analysis.

To determine the appropriate α we plot $\log(S_i)$ on $log(y_i)$ the slope of this plot will be the approximate transformation value of α



We see in the above plot that the slope is approximately 1/2 so we will assign α = 0.5 and thus conduct a square root transformation. This is done by simply taking the square root of the count data because:

$$y* = y^{1-\alpha}$$

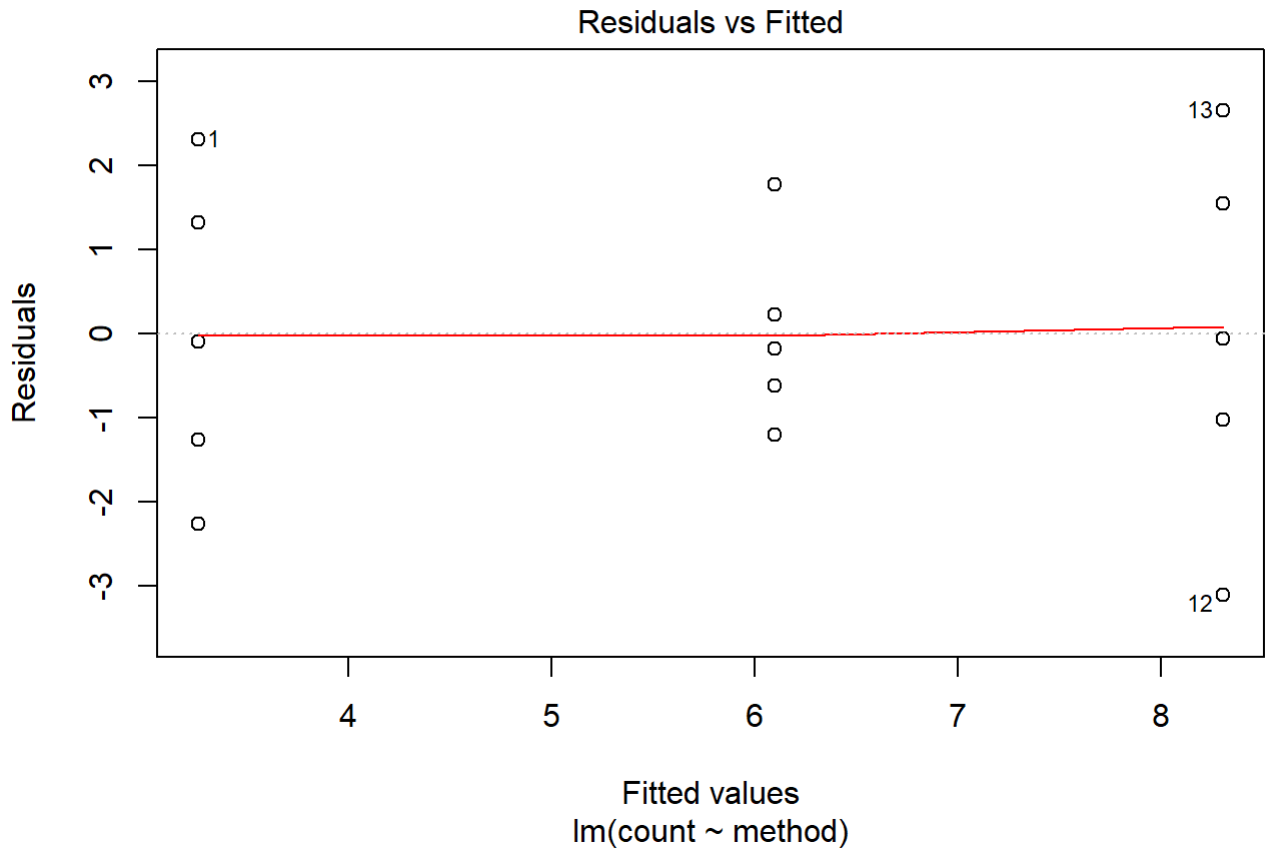
```
#Transform
transSemi = semiCondData
transSemi$count = sqrt(transSemi$count)
```

Now that the data has been transformed we can conduct a Tukey HSD test on our data. We choose the Tukey test because this test controls the overall error rate.

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = count ~ method, data = transSemi)
##
## $method
##         diff        lwr      upr     p adj
## 2-1 2.835646 -0.2047516 5.876044 0.0684023
## 3-1 5.042633  2.0022353 8.083031 0.0022071
## 3-2 2.206987 -0.8334108 5.247385 0.1709289
```

Based on the output from our Tukey test there is significant evidence to reject the null hypothesis for the comparison between Method 1 and Method 3 and conclude that Method 1 has a significantly different effect on the mean particle count when compared to Method 3. Our Tukey test also indicates that at the 0.05 significance level we fail to reject the null hypothesis for comparisons between method 1 and method 2, as well as the comparison between method 2 and method 3.

Finally we show the improved constant variance by showing the residual on fitted values plot with our transformed data



This plot is much better than our original diagnostic plot and we can say that homoskedasticity is preserved