# Untitled

*Andrew Liu*

*December 6, 2018*

## 4a

```
run=1:15
fac.a<- c(rep(c(4,8),4),rep(6,7))
fac.b<-c(10,10,20,20,15,15,15,15,
         10,20,10,20,15,15,15)
fac.c<-c(rep(2.5,4),2,2,3,3,2,2,3,3,rep(2.5,3))
y<-c(33,85,86,113,75,105,40,
         89,83,108,49,101,88,91,91)

df.4a<-data.frame(cbind(fac.a,fac.b,fac.c,y))
mean.fac.a<-mean(fac.a)
mean.fac.b<-mean(fac.b)
mean.fac.c<-mean(fac.c)
```

For factor A:

$X_{A1} = \frac{4-6}{.5*(8-4)} = -1$, $X_{A2} = \frac{6-6}{.5*(8-4)} = 0$, $X_{A3} = \frac{8-6}{.5*(8-4)} = -1$

For factor B:

$X_{B1} = \frac{10-15}{.5*(10)} = -1$, $X_{B2} = \frac{15-15}{.5*(10)} = 0$, $X_{B3} = \frac{20-15}{.5*(10)} = 1$

For factor C:

$X_{A1} = \frac{2-2.5}{.5*(1)} = -1$, $X_{A2} = \frac{2.5-2.5}{.5*(1)} = 0$, $X_{A3} = \frac{3-2.5}{.5*(1)} = 1$

Then,

```
coded.fac.a<-(fac.a-mean.fac.a)/2
coded.fac.b<-(fac.b-mean.fac.b)/5
coded.fac.c<-(fac.c-mean.fac.c)/0.5
```

## 4b

```
library(rsm)
```

```
## Warning: package 'rsm' was built under R version 3.4.4
```

```
library(plotrix)
```

```
## Warning: package 'plotrix' was built under R version 3.4.4
```

```
library(alr3)
```

```
## Warning: package 'alr3' was built under R version 3.4.4

## Loading required package: car

## Warning: package 'car' was built under R version 3.4.4
```

```
## Loading required package: carData

## Warning: package 'carData' was built under R version 3.4.4
```

```
coded.4b <- coded.data(df.4a,fac.codeda ~ (fac.a-6)/2,
    fac.codedb ~ (fac.b-15)/5, fac.codedc~(fac.c-2.5)/.5)
out <- rsm(y~SO(fac.codeda,fac.codedb,fac.codedc),data=coded.4b)
summary(out)
```

```
##
## Call:
## rsm(formula = y ~ SO(fac.codeda, fac.codedb, fac.codedc), data = coded.4b)
##
##                         Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)             90.00000    1.16905  76.9859 7.006e-09 ***
## fac.codeda              19.75000    0.71589  27.5880 1.171e-06 ***
## fac.codedb              19.75000    0.71589  27.5880 1.171e-06 ***
## fac.codedc             -11.50000    0.71589 -16.0639 1.703e-05 ***
## fac.codeda:fac.codedb   -6.25000    1.01242  -6.1733 0.0016247 **
## fac.codeda:fac.codedc    4.75000    1.01242   4.6917 0.0053768 **
## fac.codedb:fac.codedc    6.75000    1.01242   6.6672 0.0011461 **
## fac.codeda^2            -9.37500    1.05376  -8.8967 0.0002986 ***
## fac.codedb^2            -1.37500    1.05376  -1.3048 0.2487686
## fac.codedc^2            -3.37500    1.05376  -3.2028 0.0239200 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Multiple R-squared:  0.9975, Adjusted R-squared:  0.9929
## F-statistic: 218.9 on 9 and 5 DF,  p-value: 5.964e-06
##
## Analysis of Variance Table
##
## Response: y
##                                          Df Sum Sq Mean Sq  F value
## FO(fac.codeda, fac.codedb, fac.codedc)   3 7299.0 2433.00 593.4146
## TWI(fac.codeda, fac.codedb, fac.codedc)  3  428.8  142.92  34.8577
## PQ(fac.codeda, fac.codedb, fac.codedc)   3  351.5  117.16  28.5759
## Residuals                                5   20.5    4.10
## Lack of fit                              3   14.5    4.83   1.6111
## Pure error                               2    6.0    3.00
##                                            Pr(>F)
## FO(fac.codeda, fac.codedb, fac.codedc)  8.448e-07
## TWI(fac.codeda, fac.codedb, fac.codedc) 0.0008912
## PQ(fac.codeda, fac.codedb, fac.codedc)  0.0014236
## Residuals
## Lack of fit                             0.4051312
## Pure error
##
## Stationary point of response surface:
## fac.codeda fac.codedb fac.codedc
##  0.9236846 -1.7161183 -2.7698217
##
## Stationary point in original units:
##    fac.a    fac.b    fac.c
## 7.847369 6.419409 1.115089
```

```
##
## Eigenanalysis:
## eigen() decomposition
## $values
## [1]   1.280298  -3.551452 -11.853845
##
## $vectors
##                   [,1]       [,2]       [,3]
## fac.codeda -0.1236692  0.5238084  0.8428112
## fac.codedb  0.8323200 -0.4077092  0.3755217
## fac.codedc  0.5403233  0.7479291 -0.3855551
```

```
pureErrorAnova(out)
```

```
## Analysis of Variance Table
##
## Response: y
##                                          Df Sum Sq Mean Sq  F value
## FO(fac.codeda, fac.codedb, fac.codedc)   3 7299.0 2433.00 811.0000
## TWI(fac.codeda, fac.codedb, fac.codedc)  3  428.8  142.92  47.6389
## PQ(fac.codeda, fac.codedb, fac.codedc)   3  351.5  117.16  39.0537
## Residuals                                5   20.5    4.10
##  Lack of fit                             3   14.5    4.83   1.6111
##  Pure Error                              2    6.0    3.00
##                                            Pr(>F)
## FO(fac.codeda, fac.codedb, fac.codedc)  0.001232 **
## TWI(fac.codeda, fac.codedb, fac.codedc) 0.020630 *
## PQ(fac.codeda, fac.codedb, fac.codedc)  0.025070 *
## Residuals
##  Lack of fit                            0.405131
##  Pure Error
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The pvalue for lack of fit is .40, so there's not a statistically significant lack of fit for the optimization model.

## 4c

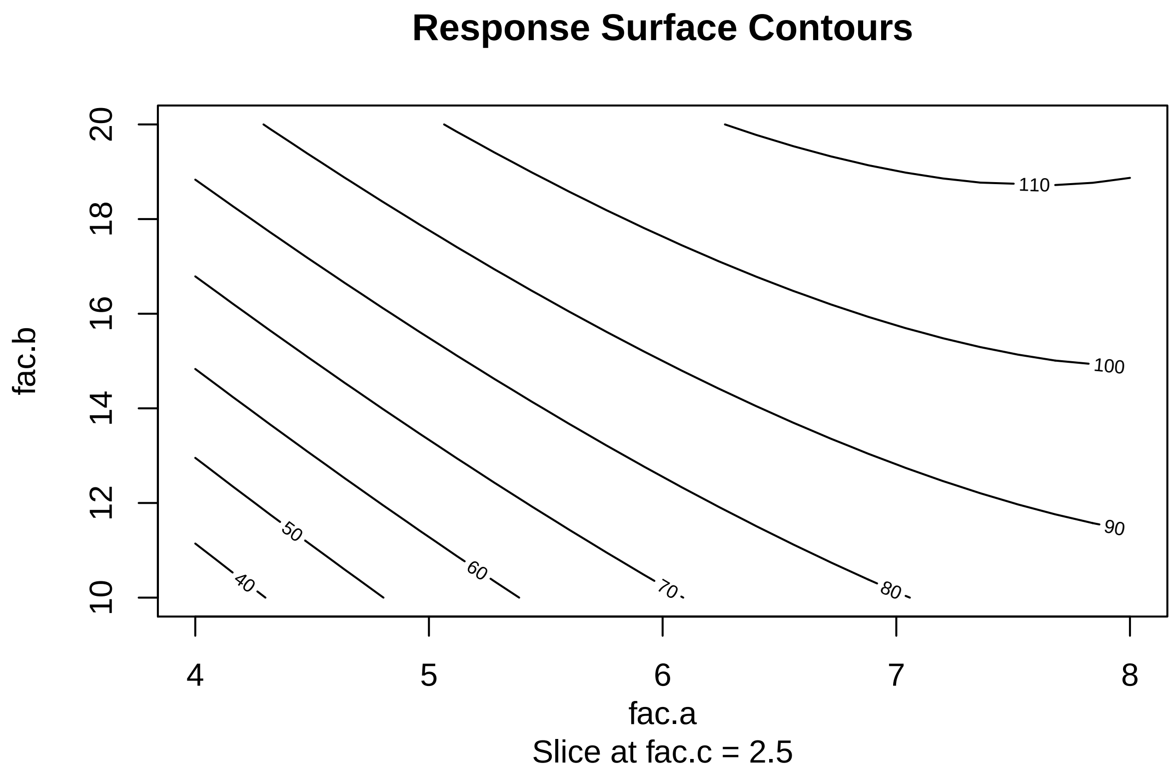
If we let $x_1, x_2, x_3$ be the variables representing the coded factors A,B,C respectively, then

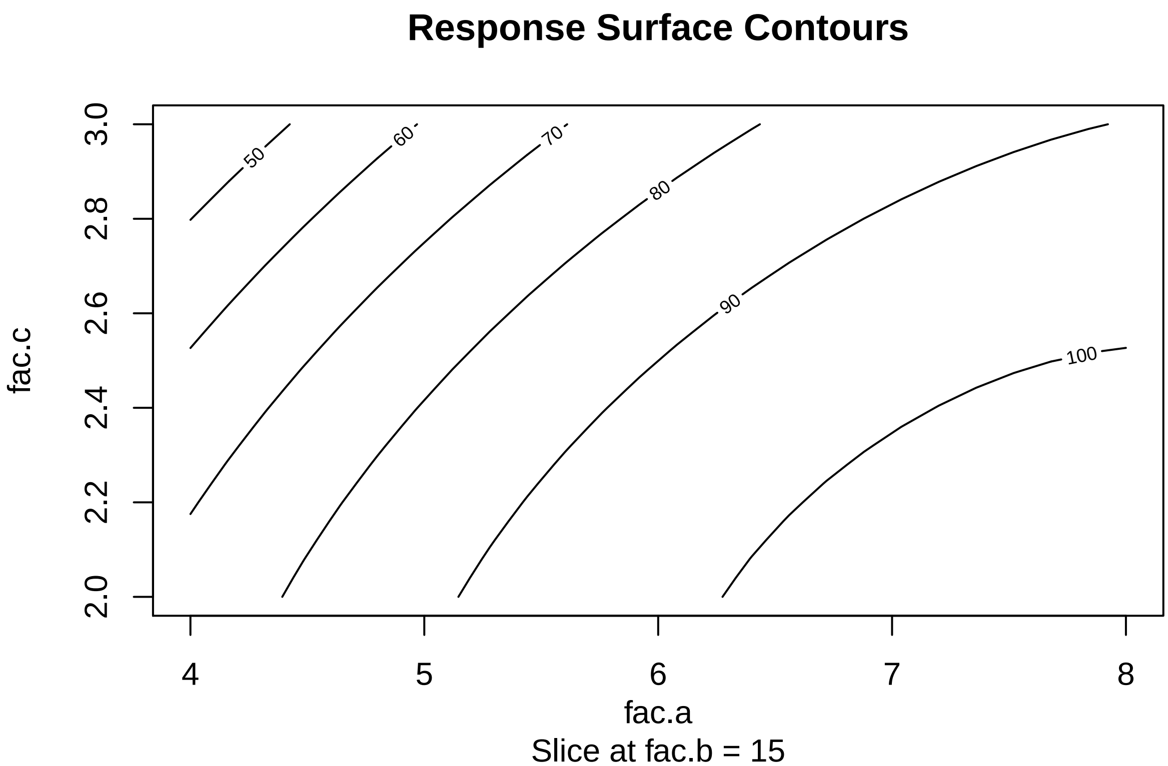
$\hat{y} = 90 + 19.75x_1 + 19.75x_2 - 11.5x_3 - 6.25x_1x_2 + 4.75x_1x_3 + 6.75x_2x_3 - 9.375x_1^2 - 1.375x_2^2 - 3.375x_3^2$

is our model.

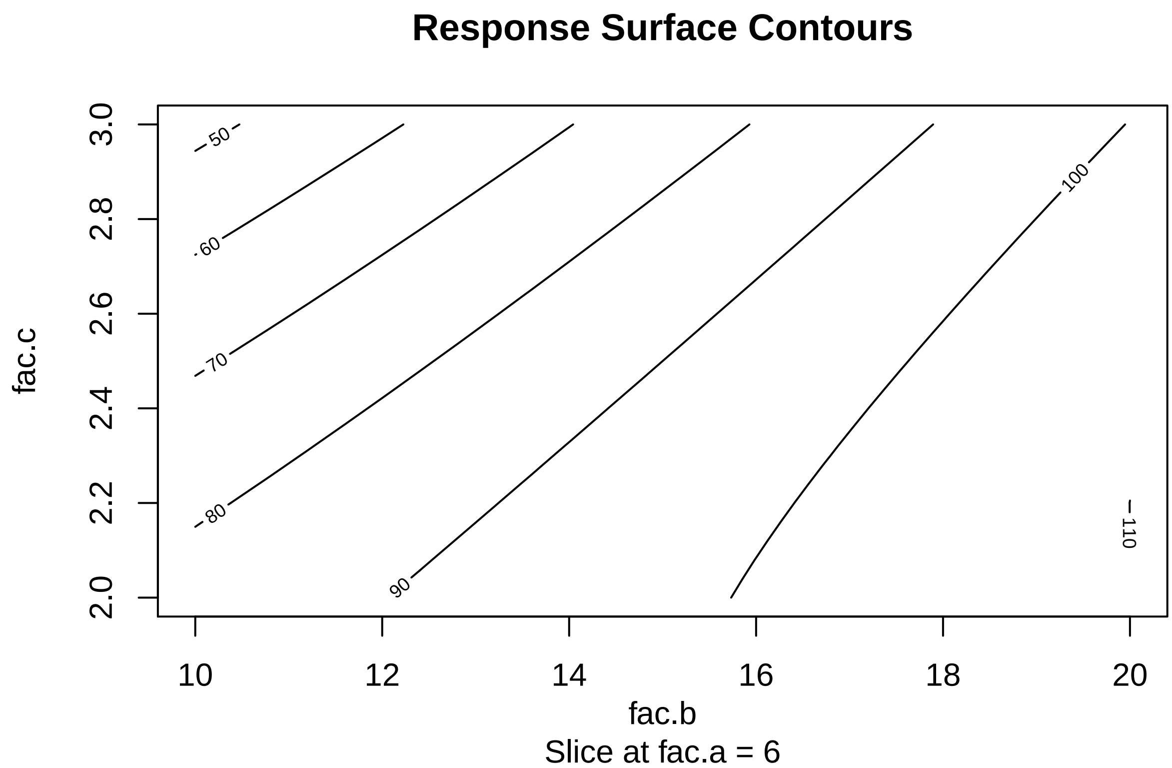
## 4d

```
contour(out, ~fac.codeda+fac.codedb+fac.codedc, main="Response Surface Contours")
```


Response Surface Contours
fac.b
20
18
16
14
12
10
4
5
6
7
8
fac.a
Slice at fac.c = 2.5
110
100
90
80
70
60
50
40



Response Surface Contours
fac.c
3.0
2.8
2.6
2.4
2.2
2.0
4
5
6
7
8
fac.a
Slice at fac.b = 15
50
60
70
80
90
100

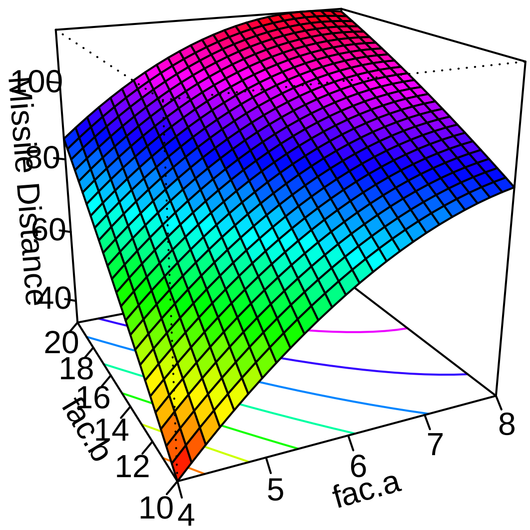



All three of these contours are rising ridges( first and third) and sloping ridge(second), so we conclude that the max/min region is outside of the design space.

## 4e

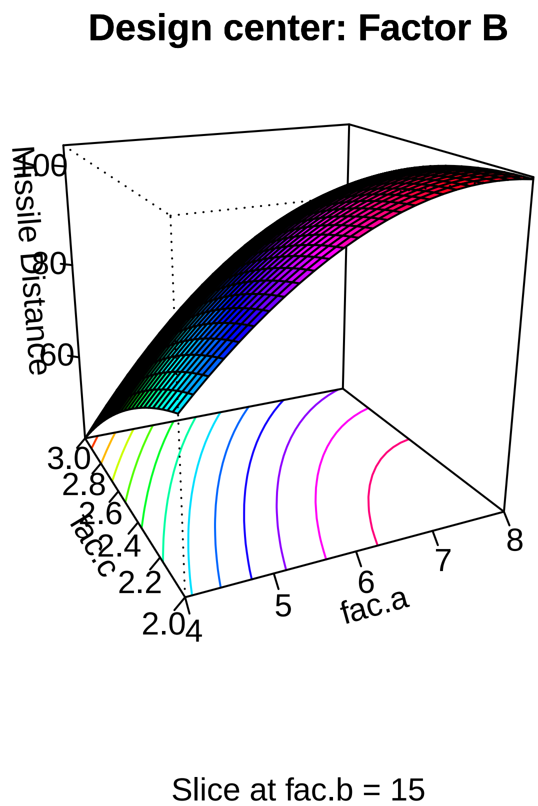
```
persp(out,~fac.codeda+fac.codedb, at = list(fac.codedc=0),col=rainbow(50),contours="colors",zlab="Missil
```

**Design center: Factor C**



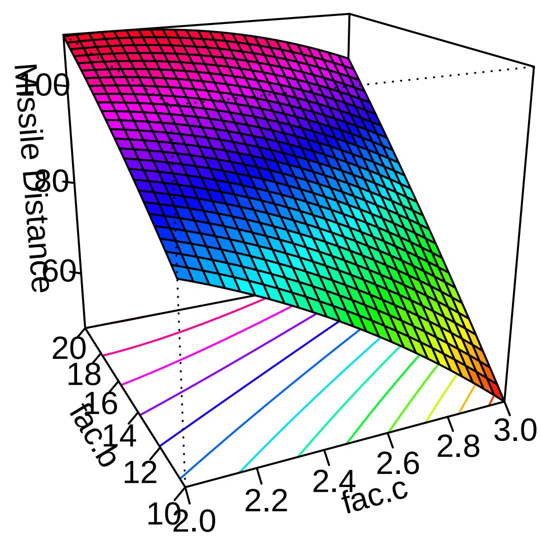

Slice at fac.c = 2.5

```
persp(out,~fac.codeda+fac.codedc, at = list(fac.codedb=0),col=rainbow(50),contours="colors",zlab="Missil
```




```
persp(out,~fac.codedc+fac.codedb, at = list(fac.codeda=0),col=rainbow(50),contours="colors",zlab="Missil
```

## Design center: Factor A




Slice at fac.a = 6

Similar to the contour plots from 4d, we conclude that the stationary point lies outside the design space for factors A,B,C.

## 4f

From part 4a we have :

$\hat{y} = 90 + 19.75x_1 + 19.75x_2 - 11.5x_3 - 6.25x_1x_2 + 4.75x_1x_3 + 6.75x_2x_3 - 9.375x_1^2 - 1.375x_2^2 - 3.375x_3^2$

Then:

$\frac{d\hat{y}}{dx_1} = 0 \implies 19.75 - 18.75x_1 - 6.25x_2 + 4.75x_3 = 0$

$\frac{d\hat{y}}{dx_2} = 0 \implies 19.75 - -6.25x_1 + 6.75x_3 - 2.75x_2 = 0$

$\frac{d\hat{y}}{dx_3} = 0 \implies -11.5 + 4.75x_1 + 6.75x_2 - 6.75x_3 = 0$

Solving yields: $x_{1(SP)} = .923, x_{2(SP)} = -1.716, x_{3(SP)} = -2.77$

```
x1=.923
x2=-1.716
x3=-2.77
y.hat=90+19.75*x1+19.75*x2-11.5*x3-6.25*x1*x2+4.75*x1*x3+6.75*x2*x3-9.375*x1^2-1.375*x2^2-3.375*x3^2
```

and $\hat{y}_{SP} = 98.1$

Since $x_1 = \frac{A-6}{2}, x_2 = \frac{B-15}{5}, x_3 = \frac{C-2.5}{.5}$

we have that $A = 7.846, B = 6.42, C = 1.12$ as the original units for factors A,B,C.

## 5a

```r
library(HoRM)
```

```
## Warning: package 'HoRM' was built under R version 3.4.4
```

```r
library(mixexp)
```

```
## Warning: package 'mixexp' was built under R version 3.4.4

## Loading required package: gdata

## Warning: package 'gdata' was built under R version 3.4.4

## gdata: Unable to locate valid perl interpreter
## gdata:
## gdata: read.xls() will be unable to read Excel XLS and XLSX files
## gdata: unless the 'perl=' argument is used to specify the location
## gdata: of a valid perl intrpreter.
## gdata:
## gdata: (To avoid display of this message in the future, please
## gdata: ensure perl is installed and available on the executable
## gdata: search path.)

## gdata: Unable to load perl libaries needed by read.xls()
## gdata: to support 'XLX' (Excel 97-2004) files.

##

## gdata: Unable to load perl libaries needed by read.xls()
## gdata: to support 'XLSX' (Excel 2007+) files.

##

## gdata: Run the function 'installXLSXsupport()'
## gdata: to automatically download and install the perl
## gdata: libaries needed to support Excel XLS and XLSX formats.

##
## Attaching package: 'gdata'

## The following object is masked from 'package:stats':
##
##     nobs

## The following object is masked from 'package:utils':
##
##     object.size

## The following object is masked from 'package:base':
##
##     startsWith

## Loading required package: lattice

## Loading required package: grid

## Loading required package: daewr

## Warning: package 'daewr' was built under R version 3.4.4
```

```r
fuel1<-c(1,1,0,0,0,0,
         .5,.5,0,
         1/3,1/3,2/3,
         1/6,1/6)

fuel2<-c(0,0,1,1,0,0,
         .5,0,.5,
         1/3,1/3,1/6,2/3,1/6)

fuel3<-c(0,0,0,0,1,1,
         0,0.5,0.5,
         1/3,1/3,1/6,1/6,2/3)
mpg<-c(24.5,25.1,24.8,23.9,22.7,23.6,
       25.1,24.3,23.5,
       24.8,24.1,24.2,23.9,23.7)

df.5<-data.frame(fuel1,fuel2,fuel3,mpg)
```

```r
mix.5a <- lm(mpg ~ fuel1 + fuel2 + fuel3 - 1, data = df.5)
summary(mix.5a)
```

```
##
## Call:
## lm(formula = mpg ~ fuel1 + fuel2 + fuel3 - 1, data = df.5)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49048 -0.35131 -0.01548  0.36702  0.64286
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## fuel1  24.9305     0.2498   99.82   <2e-16 ***
## fuel2  24.3505     0.2498   97.49   <2e-16 ***
## fuel3  23.1905     0.2498   92.85   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4295 on 11 degrees of freedom
## Multiple R-squared:  0.9998, Adjusted R-squared:  0.9997
## F-statistic: 1.477e+04 on 3 and 11 DF,  p-value: < 2.2e-16
```

The estimates for fuel 1,2,3 are 24.93,24.35,23.19 respectively. Each of the t-tests for the parameter coefficients of fuel1, 2, and 3 show that there is statistical significance in that the coefficients are nonzero.

# 5b

```r
mix.5b <- lm(mpg ~ fuel1 + fuel2 + fuel3 +fuel1:fuel2+fuel2:fuel3+fuel3:fuel1- 1, data = df.5)
summary(mix.5b)
```

```
##
## Call:
## lm(formula = mpg ~ fuel1 + fuel2 + fuel3 + fuel1:fuel2 + fuel2:fuel3 +
##     fuel3:fuel1 - 1, data = df.5)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47765 -0.34063  0.02727  0.31525  0.55114
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## fuel1        24.7443     0.3225  76.734 9.27e-13 ***
## fuel2        24.3110     0.3225  75.391 1.07e-12 ***
## fuel3        23.1777     0.3225  71.876 1.56e-12 ***
## fuel1:fuel2   1.5136     1.8168   0.833    0.429
## fuel2:fuel3  -1.0864     1.8168  -0.598    0.566
## fuel1:fuel3   1.1136     1.8168   0.613    0.557
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4653 on 8 degrees of freedom
## Multiple R-squared:  0.9998, Adjusted R-squared:  0.9996
## F-statistic:  6293 on 6 and 8 DF,  p-value: 3.019e-14
```

The coefficents are

$\beta_1 = 24.74, \beta_2 = 24.31, \beta_3 = 23.18, \beta_{1,2} = 1.514, \beta_{2,3} = -1.0864, \beta_{1,3} = 1.11$

As for the parameter coefficients, we determine that there is evidence to show that coefficients for fuel 1,2,3 coefficients (i.e. $\beta_1, \beta_2, \beta_3$) are nonzero, as the pvalues are less than 0.05. For the two-way interactions, we see that because the pvalues are all above 0.05 for $\beta_{1,2}, \beta_{1,3}, \beta_{3,2}$, we conclude that there is not enough evidence to show that those coefficients are nonzero.

## 5c

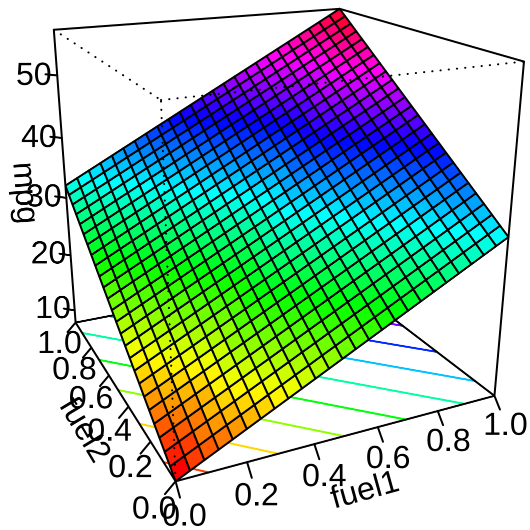
```
anova(mix.5a,mix.5b)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ fuel1 + fuel2 + fuel3 - 1
## Model 2: mpg ~ fuel1 + fuel2 + fuel3 + fuel1:fuel2 + fuel2:fuel3 + fuel3:fuel1 -
##     1
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     11 2.0296
## 2      8 1.7319  3   0.29776 0.4585 0.7188
```

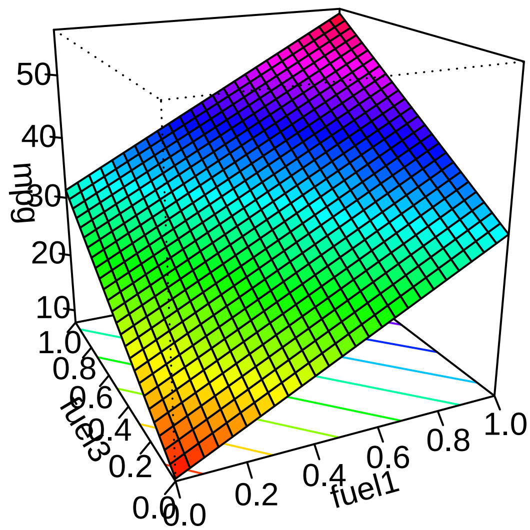
We compute that the F-statistic for comparsion between the two models is .46. This yields a pvalue of .72, so we conclude that there is no significant difference between the linear and quadratic models and so there is no need to use the quadratic model over the linear one.

## 5d

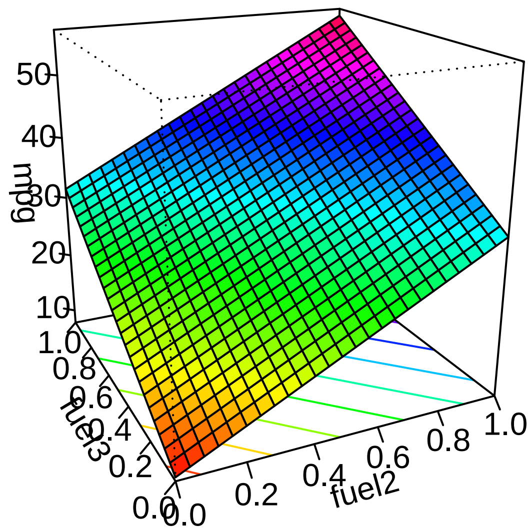
```
persp(mix.5a,~fuel1+fuel2+fuel3, col=rainbow(50),contours="colors",zlab="mpg")
```




Slice at fuel3 = 0.33




Slice at fuel2 = 0.33




Slice at fuel1 = 0.33

For this model, fuel 3 contributes the least to the mpg as its coefficient is the smallest.