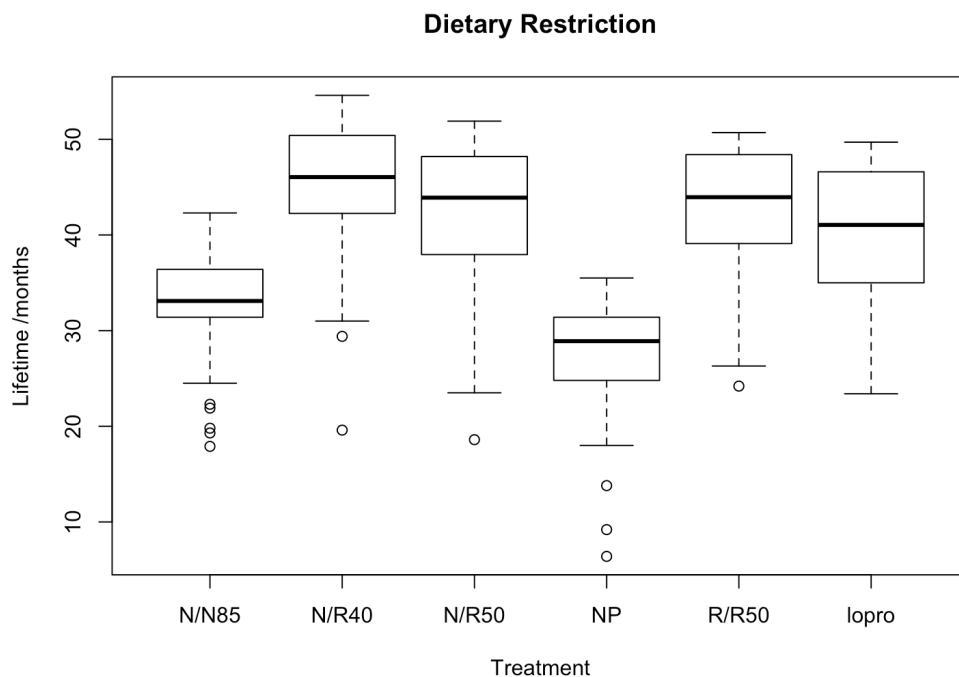


1. Case Study 5.1.1 from *The Statistical Sleuth*

```
> install.packages("Sleuth3")
> library(Sleuth3)
```

- (a) Create side-by-side boxplots of the response for this dataset, with one boxplot for each treatment group.

```
> boxplot(Lifetime~Diet,case0501, main = "Dietary Restriction",
          xlab="Treatment", ylab = "Lifetime /months")
```



- (b) Find the SSE (sum of squared errors) for the full model with one unrestricted mean for each of the six treatment groups.

From the code and output below,

$$SSE_{Full} = 15297.42.$$

```
> fit <- lm(Lifetime~Diet,case0501)
> aov(fit)
```

Call:

```
aov(formula = fit)
```

Terms:

		Diet	Residuals
Sum of Squares	12733.94	15297.42	
Deg. of Freedom	5	343	

- (c) Compute  $\hat{\sigma}^2$  for the full model with 6 means.

$$\hat{\sigma}^2 = \frac{SSE_{Full}}{df_{Full}} = \frac{15297.42}{343} = 44.6.$$

```
> anova(fit)
Analysis of Variance Table

Response: Lifetime
          Df Sum Sq Mean Sq F value    Pr(>F)
Diet         5  12734   2546.8   57.104 < 2.2e-16 ***
Residuals  343  15297     44.6
```

- (d) Find the SSE for a reduced model that has one common mean for the N/R50 and N/R50 *lopro* treatment groups and an unrestricted mean for each of the other four treatment groups.

From the code and output below,

$$SSE_{Reduced} = 15510.92.$$

```
> ## merge levels for N/R50 and lopro
> diet1 <- case0501$Diet
> levels(diet1)[c(3,6)] <- rep("N/R50+lopro",2)
> newcase <- data.frame(Lifetime = case0501$Lifetime,diet1)

> fit1 <- lm(Lifetime~diet1,newcase)
> aov(fit1)
```

```
Call:
aov(formula = fit1)
```

```
Terms:
          diet1 Residuals
Sum of Squares 12520.43  15510.92
Deg. of Freedom      4      344
```

- (e) Use the answers from parts (b) through (d) to compute an  $F$  statistic.

$$H_0 : E(\mathbf{y}) \in \mathcal{C}(\mathbf{X}_0)$$

$$H_a : E(\mathbf{y}) \in \mathcal{C}(\mathbf{X}) \setminus \mathcal{C}(\mathbf{X}_0)$$

From the code and output below,

$$F = \frac{(SSE_{Reduced} - SSE_{Full}) / (df_{Reduced} - df_{Full})}{SSE_{Full} / df_{Full}} = 4.7873$$

```
> anova(fit1, fit)
```

# Analysis of Variance Table

Model 1: Lifetime ~ diet1

Model 2: Lifetime ~ Diet

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	344	15511				
2	343	15297	1	213.51	4.7873	0.02935 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

- (f) Explain to the scientists conducting this study what the  $F$  statistic in part (e) can be used to test. Consider the context of the study and use terms non-statistician scientists will understand.

$F$  statistic in part (e) indicates the variation contributed by the variable which is eliminated from the full model. It can be used to test whether the full model is significantly better than the reduced model, that is whether there is significant difference between the N/R50 and N/R50 *lopro* treatment groups. In the context of this problem,  $F$  statistic can be used to test whether the expected lifetime is affected by reduced protein among mice who are treated with normal diet before weaning and reduced calorie (50 kcal/week) after weaning.

In this study,  $F$  statistic is 4.7873 with p-value 0.0293. We will reject the reduced model in favor of the full model at 0.05 level and conclude that there is significant difference between the N/R50 and N/R50 *lopro* treatment groups.

- (g) Consider an  $F$  statistic of the form given on slide 18 of the notes entitled *A Review of Some Key Linear Model Results*. Provide the  $\mathbf{C}$  matrix and  $\mathbf{d}$  vector and compute the  $F$  statistic corresponding to the test of the hypotheses in part (e).

$$H_0 : \mathbf{C}\boldsymbol{\beta} = \mathbf{d}$$

$$H_a : \mathbf{C}\boldsymbol{\beta} \neq \mathbf{d}$$

where  $\mathbf{C} = (0, 0, 1, 0, 0, -1)$  according to the order of the treatments in dataset,  $q = 1$  and  $\mathbf{d} = 0$ .

$$F = \frac{(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d})'(\mathbf{C}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{C}')^{-1}(\mathbf{C}\hat{\boldsymbol{\beta}} - \mathbf{d})/q}{\hat{\sigma}^2}$$

and

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}, \hat{\sigma}^2 = \frac{\mathbf{y}'(\mathbf{I} - \mathbf{P}_\mathbf{X})\mathbf{y}}{n - r}$$

.

From the code and output below,  $F = 4.787275$  the same result as in part (e).

```
> y=case0501$Lifetime
> I=diag(1, length(y))
> r=length(levels(case0501$Diet)) ## rank(X)
```

```

> xmat=model.matrix(~0+case0501$Diet)
> proj=function(x){x %*% MASS::ginv(t(x)%*%x) %*% t(x)}
> hat.sig2 = t(y) %*% (I-proj(xmat)) %*% y / (length(y)-r) ##hat.sigma^2
> hat.b=solve(t(xmat)%*%xmat) %*% t(xmat) %*% y ##hat.beta

> levels(case0501$Diet) # order of treatments
[1] "N/N85" "N/R40" "N/R50" "NP"      "R/R50" "lopro"
> C=t(c(0,0,1,0,0,-1))

> Fstat=t(C %*% hat.b) %*% solve(C %*% solve(t(xmat)%*%xmat) %*% t(C))
      %*% (C %*% hat.b)/1/hat.sig2
      [,1]
[1,] 4.787275

```

2. There are infinitely many possible examples. One example can be given using

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{G} = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}.$$

We can verify that  $\mathbf{G}$  meets the definition of a generalized inverse of  $\mathbf{A}$ :

$$\mathbf{AGA} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \mathbf{A}.$$

Clearly  $\mathbf{A}$  is symmetric (i.e.,  $\mathbf{A} = \mathbf{A}'$ ), yet  $\mathbf{G}$  is *not* symmetric:

$$\mathbf{G} = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix} \neq \begin{bmatrix} 1 & 3 \\ 2 & 0 \end{bmatrix} = \mathbf{G}'.$$

Hence, a generalized inverse of a symmetric matrix need not be symmetric.

3. We are given that for any two matrices  $\mathbf{U}$  and  $\mathbf{V}$  that allow for the product matrix  $\mathbf{UV}$ ,

$$\text{rank}(\mathbf{UV}) \leq \min \{ \text{rank}(\mathbf{U}), \text{rank}(\mathbf{V}) \} \quad (1)$$

This says  $\text{rank}(\mathbf{UV})$  is no larger than the smaller of the two quantities  $\text{rank}(\mathbf{U})$  and  $\text{rank}(\mathbf{V})$ , which implies

$$\text{rank}(\mathbf{UV}) \leq \text{rank}(\mathbf{U}) \quad \text{and} \quad \text{rank}(\mathbf{UV}) \leq \text{rank}(\mathbf{V}).$$

(a) Let  $\mathbf{X}$  be any matrix. Then,

$$\begin{aligned}
\text{rank}(\mathbf{X}'\mathbf{X}) &\leq \min \{ \text{rank}(\mathbf{X}'), \text{rank}(\mathbf{X}) \} && \text{by (??)} \\
&\leq \text{rank}(\mathbf{X}) \\
&= \text{rank}(\mathbf{P}_\mathbf{X}\mathbf{X}) && \text{since } \mathbf{P}_\mathbf{X}\mathbf{X} = \mathbf{X} \\
&= \text{rank}(\mathbf{X}(\mathbf{X}'\mathbf{X})^-\mathbf{X}'\mathbf{X}) && \text{since } \mathbf{P}_\mathbf{X} = \mathbf{X}(\mathbf{X}'\mathbf{X})^-\mathbf{X}' \\
&\leq \min \{ \text{rank}(\mathbf{X}(\mathbf{X}'\mathbf{X})^-), \text{rank}(\mathbf{X}'\mathbf{X}) \} && \text{by (??)} \\
&\leq \text{rank}(\mathbf{X}'\mathbf{X}).
\end{aligned}$$

The above says that  $\text{rank}(\mathbf{X}'\mathbf{X})$  is simultaneously no larger and no smaller than  $\text{rank}(\mathbf{X})$ , that is,

$$\text{rank}(\mathbf{X}) \leq \text{rank}(\mathbf{X}'\mathbf{X}) \quad \text{and} \quad \text{rank}(\mathbf{X}) \geq \text{rank}(\mathbf{X}'\mathbf{X}),$$

which implies that

$$\text{rank}(\mathbf{X}) = \text{rank}(\mathbf{X}'\mathbf{X}).$$

Comments: An alternative solution can be given by appealing to some linear algebra facts. Some students approached this problem by stating that  $\mathbf{X}$  and  $\mathbf{X}'\mathbf{X}$  must have the same null space because

$$\mathbf{X}\mathbf{z} = \mathbf{0} \iff \mathbf{X}'\mathbf{X}\mathbf{z} = \mathbf{0}. \quad (2)$$

However, a few additional steps are needed for this to establish the desired result. First, why does (??) imply that  $\mathbf{X}$  and  $\mathbf{X}'\mathbf{X}$  have the same null space? For an  $m \times n$  real-valued matrix  $\mathbf{A}$ , its null space,  $\text{Null}(\mathbf{A})$ , is defined as

$$\text{Null}(\mathbf{A}) = \{ \mathbf{z} \in \mathbb{R}^n : \mathbf{A}\mathbf{z} = \mathbf{0} \}.$$

From the definition, clearly (??) implies  $\text{Null}(\mathbf{X}) = \text{Null}(\mathbf{X}'\mathbf{X})$ . Now, how does this imply equality of ranks? The rank-nullity theorem says that for any  $m \times n$  matrix  $\mathbf{A}$ ,

$$\text{rank}(\mathbf{A}) + \dim(\text{Null}(\mathbf{A})) = n.$$

Noting that if  $\mathbf{X}$  is  $m \times n$ , then  $\mathbf{X}'\mathbf{X}$  is  $n \times n$ , so the rank-nullity theorem says

$$\text{rank}(\mathbf{X}) + \dim(\text{Null}(\mathbf{X})) = n = \text{rank}(\mathbf{X}'\mathbf{X}) + \dim(\text{Null}(\mathbf{X}'\mathbf{X})).$$

Since  $\text{Null}(\mathbf{X}) = \text{Null}(\mathbf{X}'\mathbf{X})$ , the two sets must have the same dimension, implying that  $\text{rank}(\mathbf{X}) = \text{rank}(\mathbf{X}'\mathbf{X})$ .

(b) Let  $\mathbf{X}$  be any matrix. Then,

$$\begin{aligned} \text{rank}(\mathbf{X}) &= \text{rank}(\mathbf{P}_\mathbf{X}\mathbf{X}) && \text{since } \mathbf{P}_\mathbf{X}\mathbf{X} = \mathbf{X} \\ &\leq \min \{ \text{rank}(\mathbf{P}_\mathbf{X}), \text{rank}(\mathbf{X}) \} && \text{by (??)} \\ &\leq \text{rank}(\mathbf{P}_\mathbf{X}) \\ &= \text{rank}(\mathbf{X}(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}') && \text{since } \mathbf{P}_\mathbf{X} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}' \\ &\leq \min \{ \text{rank}(\mathbf{X}), \text{rank}((\mathbf{X}'\mathbf{X})^{-}\mathbf{X}') \} && \text{by (??)} \\ &\leq \text{rank}(\mathbf{X}). \end{aligned}$$

Inequality in both directions implies equality; therefore,

$$\text{rank}(\mathbf{X}) = \text{rank}(\mathbf{P}_\mathbf{X}).$$

(c) Let  $\mathbf{X}$  be an  $n \times p$  matrix. Suppose  $\mathbf{C}$  is a  $q \times p$  matrix of rank  $q$  and that there exists a matrix  $\mathbf{A}$  such that  $\mathbf{C} = \mathbf{A}\mathbf{X}$ .

First, let us verify that  $\mathbf{C}(\mathbf{X}'\mathbf{X})^{-}\mathbf{C}'$  is  $q \times q$ . Because  $\mathbf{X}$  is a  $n \times p$  matrix,  $\mathbf{X}'\mathbf{X}$  is  $p \times p$  matrix. Consequently, any generalized inverse  $(\mathbf{X}'\mathbf{X})^{-}$  of  $\mathbf{X}'\mathbf{X}$  is also  $p \times p$ . Then,

$$\underbrace{\mathbf{C}(\mathbf{X}'\mathbf{X})^{-}\mathbf{C}'}_{p \times p}$$

is clearly a  $q \times q$  matrix.

Next, we will show that  $\mathbf{C}(\mathbf{X}'\mathbf{X})^{-}\mathbf{C}'$  has rank  $q$ . In order to do so, I will use following fact: for any matrix  $\mathbf{X}$ ,

$$\text{rank}(\mathbf{X}) = \text{rank}(\mathbf{X}'), \quad (3)$$

because the rank of a matrix can be defined as the number of linearly independent rows or columns.

Now, consider

$$\begin{aligned}
\text{rank}(\mathbf{C}(\mathbf{X}'\mathbf{X})^{-}\mathbf{C}') &\leq \min \{ \text{rank}(\mathbf{C}), \text{rank}((\mathbf{X}'\mathbf{X})^{-}\mathbf{C}') \} \quad \text{by (??)} \\
&\leq \text{rank}(\mathbf{C}) \\
&= q \quad \text{by assumption} \\
&= \text{rank}(\mathbf{C}) \\
&= \text{rank}(\mathbf{A}\mathbf{X}) \\
&= \text{rank}(\mathbf{A}\mathbf{P}_\mathbf{X}\mathbf{X}) \quad \text{since } \mathbf{P}_\mathbf{X}\mathbf{X} = \mathbf{X} \\
&\leq \min \{ \text{rank}(\mathbf{A}\mathbf{P}_\mathbf{X}), \text{rank}(\mathbf{X}) \} \quad \text{by (??)} \\
&\leq \text{rank}(\mathbf{A}\mathbf{P}_\mathbf{X}) \\
&= \text{rank}([\mathbf{A}\mathbf{P}_\mathbf{X}]') \quad \text{by (??)} \\
&= \text{rank}(\mathbf{P}'_\mathbf{X}\mathbf{A}') \\
&= \text{rank}([\mathbf{P}'_\mathbf{X}\mathbf{A}']'\mathbf{P}'_\mathbf{X}\mathbf{A}') \quad \text{by part (a) using } \mathbf{X} = \mathbf{P}'_\mathbf{X}\mathbf{A}' \\
&= \text{rank}(\mathbf{A}\mathbf{P}_\mathbf{X}\mathbf{P}'_\mathbf{X}\mathbf{A}') \\
&= \text{rank}(\mathbf{A}\mathbf{P}_\mathbf{X}\mathbf{P}_\mathbf{X}\mathbf{A}') \quad \mathbf{P}_\mathbf{X} \text{ is symmetric} \\
&= \text{rank}(\mathbf{A}\mathbf{P}_\mathbf{X}\mathbf{A}') \quad \mathbf{P}_\mathbf{X} \text{ is idempotent} \\
&= \text{rank}(\mathbf{A}[\mathbf{X}(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}']\mathbf{A}') \quad \text{since } \mathbf{P}_\mathbf{X} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}' \\
&= \text{rank}([\mathbf{A}\mathbf{X}](\mathbf{X}'\mathbf{X})^{-}[\mathbf{X}'\mathbf{A}']) \\
&= \text{rank}([\mathbf{A}\mathbf{X}](\mathbf{X}'\mathbf{X})^{-}[\mathbf{A}\mathbf{X}]') \\
&= \text{rank}(\mathbf{C}(\mathbf{X}'\mathbf{X})^{-}\mathbf{C}').
\end{aligned}$$

Therefore,  $\mathbf{C}(\mathbf{X}'\mathbf{X})^{-}\mathbf{C}'$  is a  $q \times q$  matrix of rank  $q$ .

Comments: In this proof, you need to be careful to avoid claiming  $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-} = (\mathbf{X}'\mathbf{X})^{-}$  or  $\mathbf{G}\mathbf{A}\mathbf{G} = \mathbf{G}$ . This is not nessessary true, for example,  $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$  and one of its generalized inverse

is  $\mathbf{G} = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$ , we can verify that  $\mathbf{G}\mathbf{A}\mathbf{G} = \begin{bmatrix} 1 & 2 \\ 3 & 6 \end{bmatrix} \neq \mathbf{G}$ .

I will recommod to use the idempotent property of projection matrix  $\mathbf{P}_\mathbf{X}\mathbf{P}_\mathbf{X} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}' = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}' = \mathbf{P}_\mathbf{X}$  or  $\mathbf{P}_\mathbf{X}\mathbf{X} = \mathbf{X}$

- (d) Suppose  $\mathbf{X}$  is an  $n \times p$  matrix and  $\mathbf{A}$  is a matrix with  $n$  columns satisfying  $\mathbf{A}\mathbf{P}_\mathbf{X} = \mathbf{A}$ . Then,

$$\begin{aligned}
 \text{rank}(\mathbf{A}) &= \text{rank}(\mathbf{A}\mathbf{P}_\mathbf{X}) \\
 &= \text{rank}(\mathbf{A}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') && \text{(def. of } \mathbf{P}_\mathbf{X}) \\
 &\leq \min \{ \text{rank}(\mathbf{A}\mathbf{X}), \text{rank}((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') \} && \text{by (??)} \\
 &\leq \text{rank}(\mathbf{A}\mathbf{X}) \\
 &\leq \min \{ \text{rank}(\mathbf{A}), \text{rank}(\mathbf{X}) \} && \text{by (??)} \\
 &\leq \text{rank}(\mathbf{A}).
 \end{aligned}$$

Inequality in both directions implies equality; therefore,

$$\text{rank}(\mathbf{A}\mathbf{X}) = \text{rank}(\mathbf{A}).$$

4. For this problem, let  $\mu_{ijk}$  denote the mean for filler type  $i = 1, 2$ , surface treatment  $j = 1, 2$ , and filler proportion  $k = 1, 2, 3$  (corresponding to 25%, 50%, and 75%, respectively).

The R code below fits the cell means model to these data:

```

> # Read in data and set covariates as factors
> dat <- read.table("https://dnett.github.io/S510/FabricLoss.txt", header = T,
+                   col.names=c("S", "F", "p", "y"),
+                   colClasses = c("factor", "factor", "factor", "numeric"))

> # Fit cell means model.
> fit <- lm(y ~ F:S:p + 0, data = dat)

```

From the output below, notice that R is using the parameterization

$$\boldsymbol{\beta} = [\mu_{111}, \mu_{211}, \mu_{121}, \mu_{221}, \mu_{112}, \mu_{212}, \mu_{122}, \mu_{222}, \mu_{113}, \mu_{213}, \mu_{123}, \mu_{223}]'. \quad (4)$$

```

> # Look at the vector of estimates.
> coef(fit)
F1:S1:p25 F2:S1:p25 F1:S2:p25 F2:S2:p25 F1:S1:p50 F2:S1:p50
  201.0    213.0    164.0    148.5    237.0    233.5
F1:S2:p50 F2:S2:p50 F1:S1:p75 F2:S1:p75 F1:S2:p75 F2:S2:p75
  187.5    113.5    267.0    234.5    232.0    143.5

```

Depending how you called `lm()`, you may not have the same parameterization of  $\boldsymbol{\beta}$  as in (4), and hence may require different  $\mathbf{C}$  matrices in the following problems.

(a) From the output below,

$$\hat{\mu}_{212} = 233.50 \quad \text{and} \quad \text{se}(\hat{\mu}_{212}) = 11.59.$$

```
> # 4a: mean and se for F2, S1, 50% filler.
> summary(fit)

Call:
lm(formula = y ~ F:S:p + 0, data = dat)

Residuals:
    Min       1Q   Median       3Q      Max
-26.000  -9.125   0.000   9.125  26.000

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
F1:S1:p25      201.00      11.59  17.340 7.33e-10 ***
F2:S1:p25      213.00      11.59  18.375 3.74e-10 ***
F1:S2:p25      164.00      11.59  14.148 7.57e-09 ***
F2:S2:p25      148.50      11.59  12.811 2.33e-08 ***
F1:S1:p50      237.00      11.59  20.445 1.08e-10 ***
F2:S1:p50      233.50      11.59  20.143 1.28e-10 ***
F1:S2:p50      187.50      11.59  16.175 1.64e-09 ***
F2:S2:p50      113.50      11.59   9.791 4.50e-07 ***
F1:S1:p75      267.00      11.59  23.033 2.67e-11 ***
F2:S1:p75      234.50      11.59  20.229 1.22e-10 ***
F1:S2:p75      232.00      11.59  20.014 1.38e-10 ***
F2:S2:p75      143.50      11.59  12.379 3.42e-08 ***
```

(b) From the code and output below (which uses Dr. Nettleton's `estimate()` function),

$$\hat{\mu}_{1..} = 214.75 \quad \text{and} \quad \hat{\mu}_{2..} = 181.08.$$

```
> # 4b: LSMEANS for F1 and F2.
> C.b <- c(rep(c(1,0), 6), # F1
+         rep(c(0,1), 6)) # F2
> C.b <- matrix(C.b / 6, nrow = 2, byrow = TRUE)
> estimate(fit, C.b)

      c1      c2      c3      c4      c5      c6      c7      c8
[1,] 0.1666667 0.0000000 0.1666667 0.0000000 0.1666667 0.0000000 0.1666667 0.0000000
[2,] 0.0000000 0.1666667 0.0000000 0.1666667 0.0000000 0.1666667 0.0000000 0.1666667
      c9      c10     c11     c12 estimate      se 95% Conf.  limits
[1,] 0.1666667 0.0000000 0.1666667 0.0000000 214.7500 4.732424  204.4389 225.0611
[2,] 0.0000000 0.1666667 0.0000000 0.1666667 181.0833 4.732424  170.7723 191.3944
```

(c) From the code and output below (which uses Dr. Nettleton's `estimate()` function),

$$\hat{\mu}_{.21} = 156.25.$$

```
> # 4c-d. LSMEAN and se for S2&p25.
> C.c <- c(0,0, 1, 1, rep(0, 8))
> C.c <- matrix(C.c / 2, nrow = 1)
> estimate(fit, C.c)

      c1 c2  c3  c4 c5 c6 c7 c8 c9 c10 c11 c12 estimate      se 95% Conf.  limits
[1,]  0  0 0.5 0.5  0  0  0  0  0  0  0  0  156.25 8.196798 138.3907 174.1093
```



(d) From the output in part (c),

$$\text{se}(\hat{\mu}_{.12}) = 8.197.$$

(e) The test for filler type main effects is given by

$$H_0 : \bar{\mu}_{1..} - \bar{\mu}_{2..} = 0.$$

The code below (which uses Dr. Nettleton's `test()` function) gives  $F = 25.30$  and  $p = 0.0003$ . These data provide strong evidence that there are filler type main effects. That is, averaging over surface treatment and proportion of filler, there is strong evidence that filler type F1 differs from F2 at preventing fabric loss in abrasive tests.

```
> # 4e. Filler type main effects.
> C.e <- matrix(rep(c(1, -1), 6), nrow = 1)
> test(fit, C.e) # Note d = 0 by default.
$Fstat
[1] 25.30481
$pvalue
[1] 0.0002939847
```

Comments: we are specifically interested in main effects here, not any kind of effect. Hence, comparing a reduced cell-means model with only the factors surface treatment and filler proportion to a full cell-means model with all three factors does not test for filler type main effects.

(f) The test for three-way interactions is given by

$$H_0 : \begin{bmatrix} \mu_{111} - \mu_{121} - \mu_{211} + \mu_{221} - \mu_{112} + \mu_{122} + \mu_{212} - \mu_{222} \\ \mu_{111} - \mu_{121} - \mu_{211} + \mu_{221} - \mu_{113} + \mu_{123} + \mu_{213} - \mu_{223} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The code below (which uses Dr. Nettleton's `test()` function) gives  $F = 0.89$  and  $p = 0.44$ . These data do not provide evidence of three-way interactions among filler type, surface treatment, and filler proportion.

```
> # 4f. Three-factor interaction.
> C.f <- c(1, -1, -1, 1, -1, 1, 1, -1, 0, 0, 0, 0,
+         1, -1, -1, 1, 0, 0, 0, 0, -1, 1, 1, -1)
> C.f <- matrix(C.f, nrow = 2, byrow = TRUE)
> test(fit, C.f)
$Fstat
[1] 0.8903876
$pvalue
[1] 0.4359589
```

Comments: a large  $p$ -value (i.e.,  $p > \alpha$ ) suggests that it *possible* that there are no three-way interactions, but it does not say there are no three-way interactions. Why? There could be three-way interactions, but they are too small (relative to the variability in the study) to detect.

(g) The test for two-way interactions between filler type and filler proportion is given by

$$H_0 : \begin{bmatrix} \bar{\mu}_{1.1} - \bar{\mu}_{1.2} - \bar{\mu}_{2.1} + \bar{\mu}_{2.2} \\ \bar{\mu}_{1.1} - \bar{\mu}_{1.3} - \bar{\mu}_{2.1} + \bar{\mu}_{2.3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The code below (which uses Dr. Nettleton's `test()` function) gives  $F = 6.57$  and  $p = 0.012$ . These data provide evidence that there are two-way interactions between filler type and filler proportion.

```

> # 4g. Two-factor interaction between filler type and proportion.
> C.g <- c(1, -1, 1, -1, -1, 1, -1, 1, 0, 0, 0, 0,
+         1, -1, 1, -1, 0, 0, 0, 0, -1, 1, -1, 1)
> C.g <- matrix(C.g, nrow = 2, byrow = TRUE)
> test(fit, C.g)
$Fstat
[1] 6.565736
$pvalue
[1] 0.01185168

```