STAT 511 Homework 5
Due Date: 11:00 A.M., Friday, February 17

1. Suppose $\boldsymbol{X}$ is an $n \times p$ design matrix and $\boldsymbol{B}$ is a $p \times p$ non-singular matrix. Prove that

$$C(\boldsymbol{X}) = C\left(\boldsymbol{X}\boldsymbol{B}^{-1}\right)$$

Solution: It's obvious that $C\left(\boldsymbol{X}\boldsymbol{B}^{-1}\right) \subseteq C(\boldsymbol{X})$ because each column of $\boldsymbol{X}\boldsymbol{B}^{-1}$ is a linear combination of the columns of $\boldsymbol{X}$. Also, we have $\boldsymbol{X} = \boldsymbol{X}\boldsymbol{B}^{-1}\boldsymbol{B}$, so the columns of $\boldsymbol{X}$ are each linear combinations of the columns of $\boldsymbol{X}\boldsymbol{B}^{-1}$. It follows that $C(\boldsymbol{X}) \subseteq C\left(\boldsymbol{X}\boldsymbol{B}^{-1}\right)$. Thus $C(\boldsymbol{X}) = C\left(\boldsymbol{X}\boldsymbol{B}^{-1}\right)$.

2. Consider the simple linear regression model

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \quad i = 1, \ldots, n$$

where $\epsilon_1, \ldots, \epsilon_n$ are iid and have a normal distribution $N\left(0, \sigma^2\right)$ and β_0,β_1, and $\sigma^2 > 0$ are unknown parameters. The design matrix for this Normal theory Gauss-Markov linear model can be written as $\boldsymbol{X} = [\mathbf{1}, \boldsymbol{x}]$, where $\mathbf{1}$ is an $n \times 1$ vector of ones and $\boldsymbol{x} = (x_1, \ldots, x_n)'$.

(a) We have learned that the least squares estimator of $\boldsymbol{\beta}$ in a Normal theory Gauss-Markov linear model with a full-rank design matrix is given by $\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{y}$. Simplify this expression for the special case of simple linear regression to obtain expression for the least squares estimators of β_0 and β_1. Express your final answers using summation notation.

Solution:

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{y} = \left(\begin{pmatrix} \mathbf{1}' \\ \boldsymbol{x}' \end{pmatrix}(\mathbf{1}, \boldsymbol{x})\right)^{-1}\begin{pmatrix} \mathbf{1}' \\ \boldsymbol{x}' \end{pmatrix}\boldsymbol{y} = \begin{pmatrix} n & \mathbf{1}'\boldsymbol{x} \\ \boldsymbol{x}'\mathbf{1} & \boldsymbol{x}'\boldsymbol{x} \end{pmatrix}^{-1}\begin{pmatrix} \mathbf{1}'\boldsymbol{y} \\ \boldsymbol{x}'\boldsymbol{y} \end{pmatrix} \\
&= \begin{pmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{pmatrix}^{-1}\begin{pmatrix} \sum y_i \\ \sum x_i y_i \end{pmatrix} \\
&= \frac{1}{n\sum x_i^2 - \left(\sum x_i\right)^2}\begin{pmatrix} \sum x_i^2 & -\sum x_i \\ -\sum x_i & n \end{pmatrix}\begin{pmatrix} \sum y_i \\ \sum x_i y_i \end{pmatrix} \\
&= \frac{1}{n\sum x_i^2 - \left(\sum x_i\right)^2}\begin{pmatrix} \sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i \\ n\sum x_i y_i - \sum x_i \sum y_i \end{pmatrix}
\end{aligned}$$

So

$$\hat{\beta}_0 = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2}$$

$$\hat{\beta}_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2}$$

(b) There are some computational advantages to working with a design matrix whose columns are orthogonal. For the simple linear regression problem consider the design matrix

$$\boldsymbol{W} = [\mathbf{1}, \boldsymbol{x} - \bar{x}\mathbf{1}]$$

This design matrix is obtained by "centering" the explanatory variable around its mean $\bar{x}$. Find a matrix $\boldsymbol{B}$ so that

$$\boldsymbol{X\beta} = \boldsymbol{XB}^{-1}\boldsymbol{B\beta} = \boldsymbol{W\alpha}$$

where $\boldsymbol{XB}^{-1} = \boldsymbol{W}$ and $\boldsymbol{B\beta} = \boldsymbol{\alpha}$.

Solution: Let

$$\boldsymbol{B}^{-1} = \begin{pmatrix} 1 & -\bar{x} \\ 0 & 1 \end{pmatrix} \text{ which implies } \begin{pmatrix} 1 & \bar{x} \\ 0 & 1 \end{pmatrix}.$$

Then $\boldsymbol{XB}^{-1} = \boldsymbol{W}$.

(c) Derive expressions for the least squares estimators of α_0 and α_1 (where $\boldsymbol{\alpha} = (\alpha_0, \alpha_1)'$ from part (b)) using $\hat{\boldsymbol{\alpha}} = (\boldsymbol{W'W})^{-1}\boldsymbol{W'y}$.

Solution: Let's denote $\boldsymbol{W}$ as $(\mathbf{1}, \boldsymbol{w})$. Similar to what we've done part (a), we have

$$\begin{aligned}
\hat{\boldsymbol{\alpha}} &= (\boldsymbol{W'W})^{-1}\boldsymbol{W'y} = \left(\begin{pmatrix} \mathbf{1}' \\ \boldsymbol{w}' \end{pmatrix}(\mathbf{1}, \boldsymbol{w})\right)^{-1}\begin{pmatrix} \mathbf{1}' \\ \boldsymbol{w}' \end{pmatrix}\boldsymbol{y} = \begin{pmatrix} n & 0 \\ 0 & \boldsymbol{w'w} \end{pmatrix}^{-1}\begin{pmatrix} \mathbf{1'y} \\ \boldsymbol{w'y} \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{n} & 0 \\ 0 & \frac{1}{\boldsymbol{w'w}} \end{pmatrix}\begin{pmatrix} \mathbf{1'y} \\ \boldsymbol{w'y} \end{pmatrix} = \begin{pmatrix} \frac{\mathbf{1'y}}{n} \\ \frac{\boldsymbol{w'y}}{\boldsymbol{w'w}} \end{pmatrix} \\
&= \begin{pmatrix} \sum y_i/n \\ \frac{\sum(x_i-\bar{x})y_i}{\sum(x_i-\bar{x})^2} \end{pmatrix}
\end{aligned}$$

(d) Multiply $\hat{\boldsymbol{\alpha}}$ from part (c) by $\boldsymbol{B}^{-1}$ from part (b) to obtain expressions for $\hat{\beta}_0$ and $\hat{\beta}_1$.

Solution:

$$\boldsymbol{B}^{-1}\hat{\boldsymbol{\alpha}} = \begin{pmatrix} 1 & -\bar{x} \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \sum y_i/n \\ \frac{\sum(x_i-\bar{x})y_i}{\sum(x_i-\bar{x})^2} \end{pmatrix} = \begin{pmatrix} \frac{\sum y_i}{n} - \bar{x}\frac{\sum(x_i-\bar{x})y_i}{\sum(x_i-\bar{x})^2} \\ \frac{\sum(x_i-\bar{x})y_i}{\sum(x_i-\bar{x})^2} \end{pmatrix}$$

(e) Show that your answer to part (a) matches your answer to part (d).

Solution: Notice that $\sum(x_i - \bar{x})^2 = \sum x_i^2 - n\bar{x}^2 = \sum x_i^2 - \frac{1}{n}\left(\sum x_i\right)^2$, we have

$$\begin{aligned}
\tilde{\beta}_0 &\equiv \frac{\sum y_i}{n} - \bar{x}\frac{\sum(x_i-\bar{x})y_i}{\sum(x_i-\bar{x})^2} = \frac{\sum y_i\left(\sum x_i^2 - \frac{1}{n}\left(\sum x_i\right)^2\right) - n\bar{x}\sum(x_i-\bar{x})y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2} \\
&= \frac{\sum y_i \sum x_i^2 - \frac{1}{n}\sum y_i\left(\sum x_i\right)^2 - n\bar{x}\sum x_i y_i - n\bar{x}^2\sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2} \\
&= \frac{\sum y_i \sum x_i^2 - \sum x_i \sum x_i y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2} = \hat{\beta}_0
\end{aligned}$$

$$\begin{aligned}\tilde{\beta}_1 &\equiv \frac{\sum(x_i-\bar{x})y_i}{\sum(x_i-\bar{x})^2} = \frac{\sum x_i y_i - \bar{x}\sum y_i}{\sum x_i^2 - \frac{1}{n}\left(\sum x_i\right)^2} = \frac{n\sum x_i y_i - n\bar{x}\sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2} \\ &= \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2} = \hat{\beta}_1\end{aligned}$$

3. An experiment was conducted to study the durability of coated fabric subjected to abrasive tests. Three factors were considered. One factor was filler type with two levels (F1 and F2). Another was surface treatment with two levels (S1 and S2). The third factor was proportion of filler with three levels (25%, 50%, and 75%). Using a completely randomized design with two fabric samples per treatment, the amount of fabric lost in milligrams for each fabric sample was recorded following testing. Data are available in a tab delimited text file at http://www.public.iastate.edu/∼dnett/S511/FabricLoss.txt.

Solution:

```
> d=read.delim("http://www.public.iastate.edu/~dnett/S511/FabricLoss.txt")
> d
   surface filler  p   y
1        1      1 25 194
2        1      1 25 208
3        1      1 50 233
4        1      1 50 241
5        1      1 75 265
6        1      1 75 269
7        1      2 25 239
8        1      2 25 187
9        1      2 50 224
10       1      2 50 243
11       1      2 75 243
12       1      2 75 226
13       2      1 25 155
14       2      1 25 173
15       2      1 50 198
16       2      1 50 177
17       2      1 75 235
18       2      1 75 229
19       2      2 25 137
20       2      2 25 160
21       2      2 50 129
22       2      2 50  98
23       2      2 75 155
24       2      2 75 132
>
> s=factor(d$surface)
```

```
> f=factor(d$filler)
> p=factor(d$p)
> y=d$y
>
> #########Part (a)#########
>
> #There are several different ways to fit a cell means model.
>
> #The long way is
>
> o=lm(y~s+f+p+s:f+s:p+f:p+s:f:p)
>
> summary(o)

Call:
lm(formula = y ~ s + f + p + s:f + s:p + f:p + s:f:p)

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

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   201.00      11.59  17.340 7.33e-10 ***
s2            -37.00      16.39  -2.257  0.04345 *
f2             12.00      16.39   0.732  0.47823
p50            36.00      16.39   2.196  0.04849 *
p75            66.00      16.39   4.026  0.00168 **
s2:f2         -27.50      23.18  -1.186  0.25851
s2:p50        -12.50      23.18  -0.539  0.59963
s2:p75          2.00      23.18   0.086  0.93268
f2:p50        -15.50      23.18  -0.669  0.51643
f2:p75        -44.50      23.18  -1.919  0.07902 .
s2:f2:p50     -43.00      32.79  -1.311  0.21423
s2:f2:p75     -28.50      32.79  -0.869  0.40177
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 16.39 on 12 degrees of freedom
Multiple R-squared: 0.9373,     Adjusted R-squared: 0.8798
F-statistic:  16.3 on 11 and 12 DF,  p-value: 1.502e-05

>
> #This model includes all main effects and interactions and
> #is equivalent to the cell means model.
>
```

```
> #The same design matrix can be obtained with the short cut
>
> o=lm(y~s*f*p)
>
> #Using this parameterization, the mean for S1, F2, 25% filler
> #is intercept+f2
>
> summary(o)

Call:
lm(formula = y ~ s * f * p)

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

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   201.00      11.59  17.340 7.33e-10 ***
s2            -37.00      16.39  -2.257  0.04345 *
f2             12.00      16.39   0.732  0.47823
p50            36.00      16.39   2.196  0.04849 *
p75            66.00      16.39   4.026  0.00168 **
s2:f2         -27.50      23.18  -1.186  0.25851
s2:p50        -12.50      23.18  -0.539  0.59963
s2:p75          2.00      23.18   0.086  0.93268
f2:p50        -15.50      23.18  -0.669  0.51643
f2:p75        -44.50      23.18  -1.919  0.07902 .
s2:f2:p50     -43.00      32.79  -1.311  0.21423
s2:f2:p75     -28.50      32.79  -0.869  0.40177
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 16.39 on 12 degrees of freedom
Multiple R-squared: 0.9373,     Adjusted R-squared: 0.8798
F-statistic:  16.3 on 11 and 12 DF,  p-value: 1.502e-05

> b=coef(o)
> v=vcov(o)
>
> #Thus, the estimate of the mean can be computed as follows.
>
> C=matrix(c(1,0,1,0,0,0,0,0,0,0,0,0),nrow=1)
> C%*%b
     [,1]
[1,]  213
```

```
>
> #The standard error is
>
> sqrt(C%*%v%*%t(C))
         [,1]
[1,] 11.59202
>
> #Note that this is the same as the standard error of the
> #intercept.  That makes sense in this case because we have
> #balanced data and the intercept is the mean for a treatment
> #group, namely, S1, F1, 25% filler.
>
> #All this can be accomplished more easily with the
> #following commands
>
> o=lm(y~s:f:p-1)
> summary(o)

Call:
lm(formula = y ~ s:f:p - 1)

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

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
s1:f1:p25   201.00      11.59  17.340 7.33e-10 ***
s2:f1:p25   164.00      11.59  14.148 7.57e-09 ***
s1:f2:p25   213.00      11.59  18.375 3.74e-10 ***
s2:f2:p25   148.50      11.59  12.811 2.33e-08 ***
s1:f1:p50   237.00      11.59  20.445 1.08e-10 ***
s2:f1:p50   187.50      11.59  16.175 1.64e-09 ***
s1:f2:p50   233.50      11.59  20.143 1.28e-10 ***
s2:f2:p50   113.50      11.59   9.791 4.50e-07 ***
s1:f1:p75   267.00      11.59  23.033 2.67e-11 ***
s2:f1:p75   232.00      11.59  20.014 1.38e-10 ***
s1:f2:p75   234.50      11.59  20.229 1.22e-10 ***
s2:f2:p75   143.50      11.59  12.379 3.42e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 16.39 on 12 degrees of freedom
Multiple R-squared: 0.9967,     Adjusted R-squared: 0.9935
F-statistic: 306.4 on 12 and 12 DF,  p-value: 5.395e-13
```

```
>
> #The estimate and standard error are in the third
> #line of the "Coefficients:" part of the output.
>
> #########Part (b) and (c)#########
>
> #We need to estimate the average of the treatment mean
> #parameters for each level of filler type.
> #This can be done with any of the design matrices
> #previously considered, but it will be easiest with
> #the last design matrix.
>
> #The lsmean for filler type 1 is
>
> b=coef(o)
> v=vcov(o)
> C=matrix(c(1,1,0,0,1,1,0,0,1,1,0,0)/6,nrow=1)
> C%*%b
       [,1]
[1,] 214.75
>
> #The standard error for the lsmean is
>
> sqrt(C%*%v%*%t(C))
         [,1]
[1,] 4.732424
>
> #The lsmean for filler type 2 is
>
> C=matrix(c(0,0,1,1,0,0,1,1,0,0,1,1)/6,nrow=1)
> C%*%b
         [,1]
[1,] 181.0833
>
> #The standard error for the lsmean is
>
> sqrt(C%*%v%*%t(C))
         [,1]
[1,] 4.732424
>
> #########Part (d)#########
>
> #We start by creating a quantitative variable
> #that contains the information about filler proportion.
>
> xp=d$p
```

```
>
> #Although not necessary in this case, we can use the function
> #as.numeric to be sure that
> #R will consider a variable as quantitative.
>
> xp=as.numeric(d$p)
>
> #Now lets fit a reduced model.
>
> red=lm(y~s*f*xp)
>
> #Check the design matrix to make sure we are getting what
> #we want.
>
> model.matrix(red)
   (Intercept) s2 f2 xp s2:f2 s2:xp f2:xp s2:f2:xp
1            1  0  0 25     0     0     0        0
2            1  0  0 25     0     0     0        0
3            1  0  0 50     0     0     0        0
4            1  0  0 50     0     0     0        0
5            1  0  0 75     0     0     0        0
6            1  0  0 75     0     0     0        0
7            1  0  1 25     0     0    25        0
8            1  0  1 25     0     0    25        0
9            1  0  1 50     0     0    50        0
10           1  0  1 50     0     0    50        0
11           1  0  1 75     0     0    75        0
12           1  0  1 75     0     0    75        0
13           1  1  0 25     0    25     0        0
14           1  1  0 25     0    25     0        0
15           1  1  0 50     0    50     0        0
16           1  1  0 50     0    50     0        0
17           1  1  0 75     0    75     0        0
18           1  1  0 75     0    75     0        0
19           1  1  1 25     1    25    25       25
20           1  1  1 25     1    25    25       25
21           1  1  1 50     1    50    50       50
22           1  1  1 50     1    50    50       50
23           1  1  1 75     1    75    75       75
24           1  1  1 75     1    75    75       75
attr(,"assign")
[1] 0 1 2 3 4 5 6 7
attr(,"contrasts")
attr(,"contrasts")$s
[1] "contr.treatment"
```

```
attr(,"contrasts")$f
[1] "contr.treatment"

>
> #Now conduct the lack of fit test.
>
> anova(red,o)
Analysis of Variance Table

Model 1: y ~ s * f * xp
Model 2: y ~ s:f:p - 1
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     16 4919.1
2     12 3225.0  4    1694.1 1.5759 0.2435
>
> #The p-value 0.2435, so there is no significant evidence of
> #lack of fit.  Treating proportion as quantitative with a
> #linear effect on the response seems OK.
>
> #########Part (e)#########
>
> #I would begin by seeing if the most complex model
> #term is needed.  Thus, test for significance of the
> #three way interaction.
>
> anova(red)
Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq F value    Pr(>F)
s          1 26268.2 26268.2 85.4409 8.114e-08 ***
f          1  6800.7  6800.7 22.1201 0.0002393 ***
xp         1  5662.6  5662.6 18.4183 0.0005603 ***
s:f        1  3952.7  3952.7 12.8566 0.0024733 **
s:xp       1   150.1   150.1  0.4881 0.4948071
f:xp       1  3451.6  3451.6 11.2267 0.0040619 **
s:f:xp     1   203.1   203.1  0.6605 0.4283150
Residuals 16  4919.1   307.4
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
>
> #The p-value is 0.4283150.
> #Thus, we can do without the three-way interaction.
>
> red2=lm(y~s+f+xp+s:f+s:xp+f:xp)
>
```

```
> #Next, let's examine each two-way interaction.
>
> drop1(red2,test="F")
Single term deletions

Model:
y ~ s + f + xp + s:f + s:xp + f:xp
       Df Sum of Sq    RSS    AIC F value    Pr(F)
<none>              5122.1 142.72
s:f     1    3952.7 9074.8 154.44  13.119 0.002106 **
s:xp    1     150.1 5272.2 141.41   0.498 0.489919
f:xp    1    3451.6 8573.7 153.08  11.456 0.003523 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
>
> #We see that surface by proportion filler
> #interaction can be removed.
>
> red3=lm(y~s+f+xp+s:f+f:xp)
> drop1(red3,test="F")
Single term deletions

Model:
y ~ s + f + xp + s:f + f:xp
       Df Sum of Sq    RSS    AIC F value    Pr(F)
<none>              5272.2 141.41
s:f     1    3952.7 9224.9 152.84  13.495 0.001738 **
f:xp    1    3451.6 8723.8 151.50  11.784 0.002968 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
>
> #The remaining interactions are significant.
> #Let's stop reducing the model here.
>
> summary(red3)

Call:
lm(formula = y ~ s + f + xp + s:f + f:xp)

Residuals:
    Min      1Q  Median      3Q     Max
-37.167  -6.042   0.750   8.219  28.958

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 168.0000    13.9738  12.023 4.90e-10 ***
```

```
s2          -40.5000     9.8810  -4.099 0.000674 ***
f2           50.7500    19.7619   2.568 0.019354 *
xp            1.3400     0.2420   5.536 2.95e-05 ***
s2:f2       -51.3333    13.9738  -3.674 0.001738 **
f2:xp        -1.1750     0.3423  -3.433 0.002968 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 17.11 on 18 degrees of freedom
Multiple R-squared: 0.8974,     Adjusted R-squared: 0.869
F-statistic:  31.5 on 5 and 18 DF,  p-value: 2.677e-08

>
> #Estimates of the slope and intercept for each combination
> #of surface and filler are obtained from the coefficients
> #as follows:
>
> #S1,F1:  168+1.34*xp
>
> #S1,F2:  (168+50.75)+(1.34-1.175)*xp
> #        =218.75+0.165*xp
>
> #S2,F1:  (168-40.5)+1.34*xp
> #        =127.5+1.34*xp
>
> #S2,F2:  (168-40.5+50.75-51.333)+(1.34-1.175)*xp
> #        =126.92+0.165*xp
>
> #########Part (f)#########
>
> #We can see that the line corresponding to S2F2
> #is below all other lines.  Thus, S2F2 is the
> #best combination for preventing fabric loss.
> #Because the slope on xp is positive, proportion
> #filler 25% may be preferred.
>
> #The slope of that line is close to zero,
> #and we cant test the null hypothesis that
> #the slope for that line is zero using the
> #function "test" as follows.
>
> test=function(lmout,C,d=0){
+   b=coef(lmout)
+   V=vcov(lmout)
+   dfn=nrow(C)
+   dfd=lmout$df
```

```
+   Cb.d=C%*%b-d
+   Fstat=drop(t(Cb.d)%*%solve(C%*%V%*%t(C))%*%Cb.d/dfn)
+   pvalue=1-pf(Fstat,dfn,dfd)
+   list(Fstat=Fstat,pvalue=pvalue)
+ }
>
> test(red3,t(c(0,0,0,1,0,1)))
$Fstat
[1] 0.4647483

$pvalue
[1] 0.5040911

>
> #From the above result, we could conclude that there are
> #no significant differences among the proportions of filler
> #for the S2 F2 combination because the slope 0.165 is not
> #significantly different from zero (p-value 0.5041).
> #Thus, S2, F2, and any proportion (25%, 50%, 75%) could be
> #a reasonable answer, although 25% proportion may be best
> #given the positive slope estimate.
```