INF 511 Assignment 6

Natasha Wesely

2022-11-21

The data for this homework were obtained from a randomized experiment to estimate the effect of a diet (Diet) on the concentration of a substance (Conc) in the blood of participants (no units given). (This is similar to the diet example in our notes.) The data are contained in the file hw6.rds in BbLearn and are shown in the table, below. We will conduct analyses of these data using the cell means model and the factor effects models with reference treatment coding and sum-to-zero coding. Use readRDS(file='hw6.rds') to read the data frame into R.

Question 1

What is the factor in this experiment? (1 point)

Diet

Question 2

Give at least one other name for a factor. (1 point)

Categorical variable

Question 3

What are the levels of the factor? (1 point)

A, B, C, D, and E

The levels of a factor are the different versions/types of the categorical variable.

Question 4

What are the treatments? (1 point)

In this experiment, the treatments are Diet A, Diet B, Diet C, Diet D, and Diet E. Treatments are the different unique levels of the factor.

Question 5

Make sure the Diet variable is seen as a factor variable by R. (2 points)

```
# read in the data
df = readRDS(file='hw_assignmentInstructions/hw6.rds')

# check to make sure the Diet variable is seen as a factor
str(df)
```

```
## 'data.frame':    30 obs. of  2 variables:
##  $ Conc: num  66 90 87 76 106 73 69 71 36 74 ...
##  $ Diet: Factor w/ 6 levels "A","B","C","D",..: 4 4 4 1 1 1 2 2 2 3 ...
```

Diet is recognized by R as a factor with 6 levels.

Question 6

It seems reasonable to have the Control level be the reference level, at least for the factor effects model with reference treatment coding. Use the relevel function in R to make the Control level the reference level. Make this change to the Diet factor, keeping remaining levels in their same relative order. (We will use treatment coding, shortly.) Be sure to use the so releveled Diet factor throughout the remainder of this homework; replacing the existing Diet factor in the data frame will help to ensure this. Show your code and output to convince me that you've reordered the levels as requested. (3 points)

```
# Use the relevel function in R to make the Control level the reference level
df$Diet <- relevel(df$Diet, ref = "Control")

# convince me that you've reordered the levels as requested
contrasts(df$Diet)
```

```
##         A B C D E
## Control 0 0 0 0 0
## A       1 0 0 0 0
## B       0 1 0 0 0
## C       0 0 1 0 0
## D       0 0 0 1 0
## E       0 0 0 0 1
```

You can tell that the reordering was successful because when I run the contrasts() function, it shows "Control" as the top level (row).

Question 7

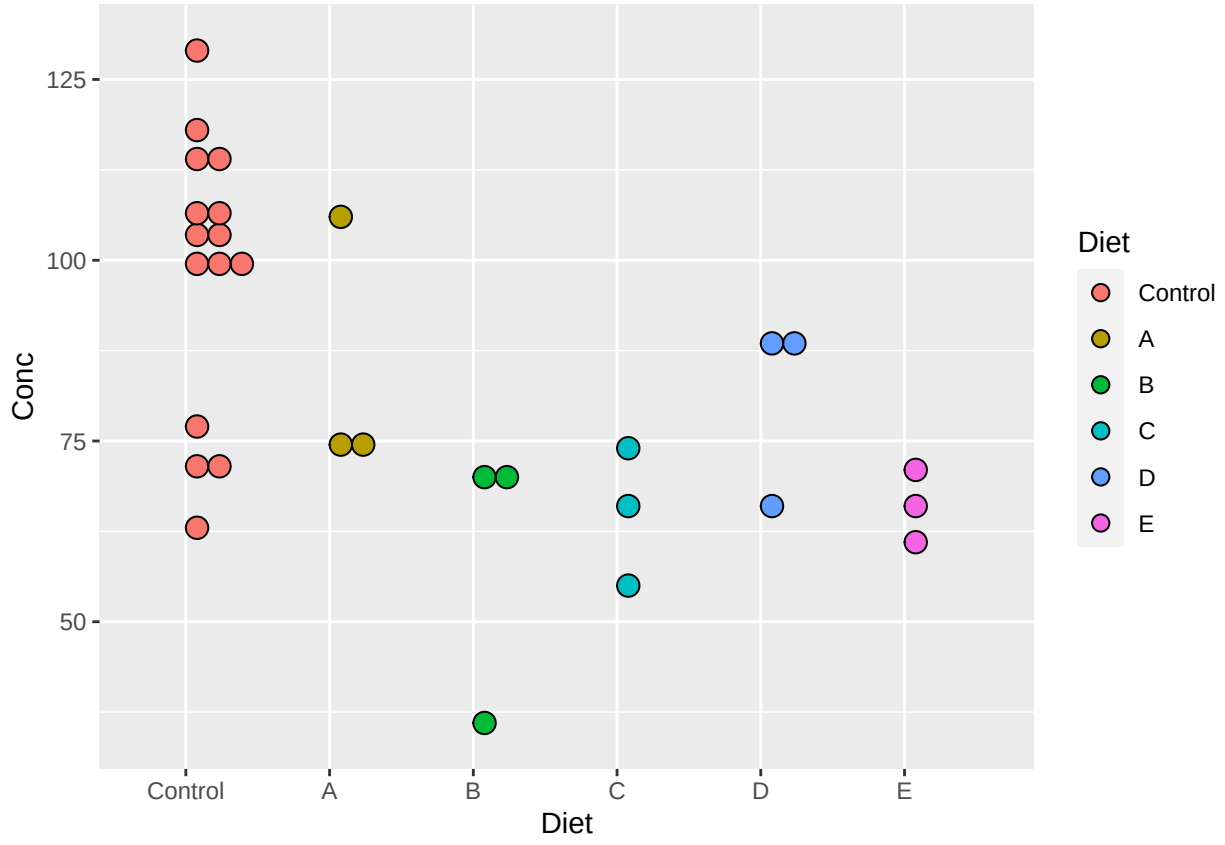
Create a scatter plot (dot plot) of concentration level (vertical) by factor level. We did this in our notes with a convenient function in R. Show your code and plot. (5 points)

```
library(tidyverse, quietly = T)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
df %>%
  ggplot()+
  geom_dotplot(aes(
    x = Diet,
    y = Conc,
    fill = Diet),
    binaxis = "y")
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with `binwidth`.
```



Question 8

8.a

Compute a one-way ANOVA (i.e., one-factor linear model) using the cell means model and test for ("overall") equality of factor level means (alpha = 0.05). Be sure to state null and alternative hypotheses, report a test statistic, p-value and state your conclusions. In particular, what are the null and alternative hypotheses in terms of CBeta? Etc. You may use stats::anova or gmodels::glh.test, as in our notes. (Remember, by default, R reports results that are not likely of interest when using the cell means model. You'll have to fix this as we did in our notes.) Also, show your code and output. (10 points)

```
# create the model
# add the "- 1" to the formula to ensure I am doing the cell means model
model_cellMn = lm(Conc ~ Diet - 1, data = df)

summary(model_cellMn)
```

```
##
## Call:
## lm(formula = Conc ~ Diet - 1, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.467  -9.750   3.033   9.000  30.533
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## DietControl   98.467      4.467  22.045  < 2e-16 ***
## DietA         85.000      9.988   8.510 1.04e-08 ***
## DietB         58.667      9.988   5.874 4.65e-06 ***
## DietC         65.000      9.988   6.508 9.91e-07 ***
## DietD         81.000      9.988   8.110 2.48e-08 ***
## DietE         66.000      9.988   6.608 7.79e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.3 on 24 degrees of freedom
## Multiple R-squared:  0.9688, Adjusted R-squared:  0.961
## F-statistic: 124.1 on 6 and 24 DF,  p-value: < 2.2e-16
```

```
# get an overall F test stat by explicitly fitting a contant
# mean model that does not restrict that constant to be zero
# then use or Full vs Reduced (F v R) approach

modelReduced = lm(Conc ~ 1, data = df)

anova(modelReduced, model_cellMn)
```

```
## Analysis of Variance Table
##
## Model 1: Conc ~ 1
```

```
## Model 2: Conc ~ Diet - 1
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1     29 14312.8
## 2     24  7182.4  5    7130.4 4.7652 0.003643 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

H_0: The mean substance concentration in the patients' blood of the diets are all equal.

$\mu_A = \mu_B = \mu_C = \mu_D = \mu_E$

H_a: The mean concentration of the different diets are not all equal.

Not all μ_j are equal

Test Statistic: $F = 4.7652$

P-value: 0.003643

Conclusions: Because the p-value (0.003643) is less than alpha (0.05), I can reject the null hypothesis. Therefore, the mean concentration of all the different diets are NOT the same.

8.b

Report the factor level (i.e., treatment) estimated mean concentration $\hat{\mu_i}$, $i = 1, ..., a$ in $\hat{\beta}$ (5 points)

```
coef(model_cellMn)
```

```
## DietControl       DietA       DietB       DietC       DietD       DietE
##    98.46667    85.00000    58.66667    65.00000    81.00000    66.00000
```

8.c

It seems plausible that our fellow researchers may have wanted to compare the control group to the remaining groups. Using the cell means model, compare the mean of the control group with the average of the means in the remaining groups. In particular, construct a 95% confidence interval for the difference between the mean concentration of the control group and the average of the mean concentrations of the remaining groups. Report your code/output and summarize your interval very briefly. (10 points)

```
# construct a 95% confidence interval for the difference between the mean
# concentration of the control group and the average of the mean
# concentrations of the remaining groups.
TukeyHSD(aov(model_cellMn))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = model_cellMn)
##
## $Diet
##                 diff       lwr        upr     p adj
## A-Control -13.466667 -47.29565 20.3623186 0.8176120
```

```
## B-Control -39.800000 -73.62899 -5.9710148 0.0146361
## C-Control -33.466667 -67.29565  0.3623186 0.0536821
## D-Control -17.466667 -51.29565 16.3623186 0.6085596
## E-Control -32.466667 -66.29565  1.3623186 0.0651483
## B-A       -26.333333 -70.00637 17.3396988 0.4465444
## C-A       -20.000000 -63.67303 23.6730322 0.7174122
## D-A        -4.000000 -47.67303 39.6730322 0.9997197
## E-A       -19.000000 -62.67303 24.6730322 0.7576325
## C-B         6.333333 -37.33970 50.0063655 0.9974238
## D-B        22.333333 -21.33970 66.0063655 0.6179742
## E-B         7.333333 -36.33970 51.0063655 0.9948716
## D-C        16.000000 -27.67303 59.6730322 0.8628878
## E-C         1.000000 -42.67303 44.6730322 0.9999997
## E-D       -15.000000 -58.67303 28.6730322 0.8914436
```

I am 95% confident the mean difference in substance concentration between Diet A group and the control is somewhere between -47.29565 and 20.3623186.

Similarly, I am 95% confident the mean difference between Diet B group and the control is between -73.62899 and -5.9710148, meaning the people in the Diet B group have 6-74 units less of the concentration of a substance (Conc) in their blood compared to the control group.

I am 95% confident the mean difference between Diet C group and the control is between -67.2956 and 0.3623186.

I am 95% confident the mean difference between Diet D group and the control is between -51.29565 and 16.3623186.

I am 95% confident the mean difference between Diet E group and the control is between -66.29565 and 1.3623186.

Question 9

9.a

Change to the factor effects model with sum-to-zero coding and report code/output to convince me that you have done this. (5 points)

```
# Any contrasts set for the particular diet factor?
attr(df$Diet, which='contrasts') # no
```

```
## NULL
```

```
# shows _global_ setting if no local  contrast attribute
contrasts(df$Diet)
```

```
##         A B C D E
## Control 0 0 0 0 0
## A       1 0 0 0 0
## B       0 1 0 0 0
## C       0 0 1 0 0
## D       0 0 0 1 0
## E       0 0 0 0 1
```

```r
# change coding/contraints/contrasts
contrasts(df$Diet) <- contr.sum(levels(df$Diet))
# check to make sure the local settings changed
attr(df$Diet, which='contrasts')
```

```
##         [,1] [,2] [,3] [,4] [,5]
## Control    1    0    0    0    0
## A          0    1    0    0    0
## B          0    0    1    0    0
## C          0    0    0    1    0
## D          0    0    0    0    1
## E         -1   -1   -1   -1   -1
```

9.b

Repeat 8a, above, now using the factor effects model with sum-to-zero coding. Be sure to report your (new) C matrix. (Note: R's default behavior should be what we want now that we do not omit the overall constant ("intercept") from our model. That is, its overall F test and R2 are more likely to be of interest. Still, report a C matrix and use stats::anova or gmodels::glh.test as in 8a, above.) (10 points)

```r
model_effectSum0 = lm(Conc ~ Diet, data = df)
summary(model_effectSum0)
```

```
##
## Call:
## lm(formula = Conc ~ Diet, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.467  -9.750   3.033   9.000  30.533
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   75.689      3.796  19.939  < 2e-16 ***
## Diet1         22.778      5.264   4.327  0.00023 ***
## Diet2          9.311      8.995   1.035  0.31093
## Diet3        -17.022      8.995  -1.892  0.07056 .
## Diet4        -10.689      8.995  -1.188  0.24634
## Diet5          5.311      8.995   0.590  0.56041
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.3 on 24 degrees of freedom
## Multiple R-squared:  0.4982, Adjusted R-squared:  0.3936
## F-statistic: 4.765 on 5 and 24 DF,  p-value: 0.003643
```

```r
# compare the reduced & full model (the sum to zero constrained effects model)
anova(modelReduced, model_effectSum0)
```

```
## Analysis of Variance Table
```

```
##
## Model 1: Conc ~ 1
## Model 2: Conc ~ Diet
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1     29 14312.8
## 2     24  7182.4  5    7130.4 4.7652 0.003643 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# report a C matrix
(Cmat = matrix(c(0, 1, 0, 0, 0, 0,
                 0, 0, 1, 0, 0, 0,
                 0, 0, 0, 1, 0, 0,
                 0, 0, 0, 0, 1, 0,
                 0, 0, 0, 0, 0, 1)
               , ncol=6, byrow=TRUE))
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    0    1    0    0    0    0
## [2,]    0    0    1    0    0    0
## [3,]    0    0    0    1    0    0
## [4,]    0    0    0    0    1    0
## [5,]    0    0    0    0    0    1
```

```
# use gmodels with C matrix to make sure you get the same results
d<- rep(0,nrow(Cmat))
gmodels::glh.test(model_effectSum0, cm=Cmat, d=d)
```

```
##
##   Test of General Linear Hypothesis
## Call:
## gmodels::glh.test(reg = model_effectSum0, cm = Cmat, d = d)
## F = 4.7652, df1 =  5, df2 = 24, p-value = 0.003643
```

H_0: The effect of the diets are all equal.

$\alpha_A = \alpha_B = \alpha_C = \alpha_D = \alpha_E$

H_a: The effects of the different diets are not all equal.

Not all α_j are equal

C Matrix:

```
 [,1] [,2] [,3] [,4] [,5] [,6]
```

[1,] 0 1 0 0 0 0

[2,] 0 0 1 0 0 0

[3,] 0 0 0 1 0 0

[4,] 0 0 0 0 1 0

[5,] 0 0 0 0 0 1

Test Statistic: $F = 4.7652$

P-value: 0.003643

Conclusions: Because the p-value (0.003643) is less than alpha (0.05), I can reject the null hypothesis. Therefore, the effects (alphas) of all the different diets are NOT the same.

9.c

Repeat 8b, above, now using the factor effects model with sum-to-zero coding. In particular, I want beta hats for the factor effects model with sum-to-zero coding, of course! (Be sure to compute alpha hats, too!) ALSO, compute mu + alpha_i hats and compare these a values with the estimated means computed using the cell means model in 8b, above. (5 points)

```
# beta hats for effects model with sum to zero constraints
(bHats = coef(model_effectSum0))
```

```
## (Intercept)       Diet1       Diet2       Diet3       Diet4       Diet5
##   75.688889   22.777778    9.311111  -17.022222  -10.688889    5.311111
```

```
# calculate the alpha hats for effect model with sum to zero contraint
# alpha_i = mu_i - mu

# calc mu
# mu is the overall mean for this model
mu = mean(df$Conc)

# calc means for each group
muControl = mean(df$Conc[which(df$Diet == "Control")])
muA = mean(df$Conc[which(df$Diet == "A")])
muB = mean(df$Conc[which(df$Diet == "B")])
muC = mean(df$Conc[which(df$Diet == "C")])
muD = mean(df$Conc[which(df$Diet == "D")])
muE = mean(df$Conc[which(df$Diet == "E")])

# calc the alphas
alphaControl = muControl - mu
alphaA = muA - mu
alphaB = muB - mu
alphaC = muC - mu
alphaD = muD - mu
alphaE = muE - mu

# print all the alphas
alphas = c(alphaControl, alphaA, alphaB, alphaC, alphaD, alphaE)
names(alphas) = c("alphaControl",'alphaA', 'alphaB', 'alphaC', 'alphaD', 'alphaE')
alphas
```

```
## alphaControl       alphaA       alphaB       alphaC       alphaD       alphaE
##     13.66667      0.20000    -26.13333    -19.80000     -3.80000    -18.80000
```

```
# compute mu + alpha_i hats
muAlphaHats = mu + alphas
names(muAlphaHats) = c('muAlphaControl_hat','muAlphaA_hat', 'muAlphaB_hat', 'muAlphaC_hat', 'muAlphaD_ha

# print the mu + alpha_i hats
muAlphaHats
```

```
## muAlphaControl_hat       muAlphaA_hat       muAlphaB_hat       muAlphaC_hat
##           98.46667           85.00000           58.66667           65.00000
##       muAlphaD_hat         alphaE_hat
##           81.00000           66.00000
```

```
# compare mu + alpha_i hats to cell means model betas
coef(model_cellMn)
```

```
## DietControl       DietA       DietB       DietC       DietD       DietE
##    98.46667    85.00000    58.66667    65.00000    81.00000    66.00000
```

My $\mu \hat{+} \alpha_i$ from my effect model (with sum to zero constraint) match nicely with my $\hat{\beta}$ from my cell mean model.

9.d

Repeat 8c, above, now using the factor effects model with sum-to-zero coding. (10 points)

ASK!!!!!!!!!!!!!!

```
# compute 95% CIs for the difference in effects between the concentration
# of the control group and the concentrations of the remaining groups.
TukeyHSD(aov(model_effectSum0))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = model_effectSum0)
##
## $Diet
##                 diff       lwr        upr     p adj
## A-Control -13.466667 -47.29565 20.3623186 0.8176120
## B-Control -39.800000 -73.62899 -5.9710148 0.0146361
## C-Control -33.466667 -67.29565  0.3623186 0.0536821
## D-Control -17.466667 -51.29565 16.3623186 0.6085596
## E-Control -32.466667 -66.29565  1.3623186 0.0651483
## B-A       -26.333333 -70.00637 17.3396988 0.4465444
## C-A       -20.000000 -63.67303 23.6730322 0.7174122
## D-A        -4.000000 -47.67303 39.6730322 0.9997197
## E-A       -19.000000 -62.67303 24.6730322 0.7576325
## C-B         6.333333 -37.33970 50.0063655 0.9974238
## D-B        22.333333 -21.33970 66.0063655 0.6179742
## E-B         7.333333 -36.33970 51.0063655 0.9948716
## D-C        16.000000 -27.67303 59.6730322 0.8628878
## E-C         1.000000 -42.67303 44.6730322 0.9999997
## E-D       -15.000000 -58.67303 28.6730322 0.8914436
```

Question 10

10.a

Change to the factor effects model with (reference) treatment coding, with the control group as the reference level. Report code/output to convince me that you have done this correctly. (5 points)

ASK!!!!!!!!!!

so is the alpha1 = 0 for control? or diet A?

```
# re-read in the data
df = readRDS(file='hw_assignmentInstructions/hw6.rds')

# Use the relevel function in R to make the Control level the reference level
df$Diet <- relevel(df$Diet, ref = "Control")

# convince me that you've reordered the levels as requested
contrasts(df$Diet)
```

```
##         A B C D E
## Control 0 0 0 0 0
## A       1 0 0 0 0
## B       0 1 0 0 0
## C       0 0 1 0 0
## D       0 0 0 1 0
## E       0 0 0 0 1
```

10.b

Repeat 8a, above, now using the factor effects model with treatment coding. Be sure to report your (new) C matrix. (Note: R's default behavior should be what we want now that we do not omit the overall constant from our model. Still, report a C matrix and use stats::anova or gmodels::glh.test as in 8a, above.) (10 points)

```
# fit a new effect model with treatment group coding
model_effectTreatment = lm(Conc ~ Diet, data = df)

# compare the reduced & full model
anova(modelReduced, model_effectTreatment)
```

```
## Analysis of Variance Table
##
## Model 1: Conc ~ 1
## Model 2: Conc ~ Diet
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1     29 14312.8
## 2     24  7182.4  5    7130.4 4.7652 0.003643 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

H_0: The effect of diets are all equal.

$\alpha_A = \alpha_B = \alpha_C = \alpha_D = \alpha_E$

H_a: The effects of the different diets are not all equal.

Not all α_j are equal

C Matrix: ?????????????????????????????? ASK !!!!!!!!!!!!!!

```
 [,1] [,2] [,3] [,4] [,5] [,6]
```

[1,] 0 1 0 0 0 0

[2,] 0 0 1 0 0 0

[3,] 0 0 0 1 0 0

[4,] 0 0 0 0 1 0

[5,] 0 0 0 0 0 1

Test Statistic: $F = 4.7652$

P-value: 0.003643

Conclusions: Because the p-value (0.003643) is less than alpha (0.05), I can reject the null hypothesis. Therefore, the effects (alphas) of all the different diets are NOT the same.

10.c

Repeat 8b, above, now using the factor effects model with treatment coding. In particular, I want beta hats for the factor effects model with treatment coding, of course! (Be sure to give alpha1 hat, too!) ALSO, compute mu + alpha_i hats and compare these alpha values with the estimated means computed using the cell means model in 8b, above. (5 points)

```
# beta hats for effect model with treatment coding
coef(model_effectTreatment)
```

```
## (Intercept)       DietA       DietB       DietC       DietD       DietE
##    98.46667   -13.46667   -39.80000   -33.46667   -17.46667   -32.46667
```

```
# calc alpha hats

# as part of the treatment constraint, alpha1 = 0
alphaControl = 0

# the rest of the alphas should be mu_i - mu

# calc mu
# mu is now the mean of the observations associated with the first
# ("reference/baseline") factor level
mu = mean(df$Conc[which(df$Diet == "Control")])

# calc means for each group
muA = mean(df$Conc[which(df$Diet == "A")])
muB = mean(df$Conc[which(df$Diet == "B")])
```

```
muC = mean(df$Conc[which(df$Diet == "C")])
muD = mean(df$Conc[which(df$Diet == "D")])
muE = mean(df$Conc[which(df$Diet == "E")])

# calc the alphas
# mu_i - mu
alphaA = muA - mu
alphaB = muB - mu
alphaC = muC - mu
alphaD = muD - mu
alphaE = muE - mu

# print all the alphas
alphas = c(alphaControl, alphaA, alphaB, alphaC, alphaD, alphaE)
names(alphas) = c('alphaControl','alphaA', 'alphaB', 'alphaC', 'alphaD', 'alphaE')
alphas
```

```
## alphaControl       alphaA       alphaB       alphaC       alphaD       alphaE
##      0.00000    -13.46667    -39.80000    -33.46667    -17.46667    -32.46667
```

```
# compute mu + alpha_i hats
muAlphaHats = mu + alphas
names(muAlphaHats) = c("muAlphaControl_hat",'muAlphaA_hat', 'muAlphaB_hat', 'muAlphaC_hat', 'muAlphaD_ha

# print the mu + alpha_i hats
muAlphaHats
```

```
## muAlphaControl_hat       muAlphaA_hat       muAlphaB_hat       muAlphaC_hat
##           98.46667           85.00000           58.66667           65.00000
##       muAlphaD_hat         alphaE_hat
##           81.00000           66.00000
```

```
# compare mu + alpha_i hats to cell means model betas
coef(model_cellMn)
```

```
## DietControl       DietA       DietB       DietC       DietD       DietE
##    98.46667    85.00000    58.66667    65.00000    81.00000    66.00000
```

My $\mu \hat{+} \alpha_i$ from my effect model (with treatment constraint) match nicely with my $\hat{\beta}_i$ from my cell mean model.

10.d

Repeat 8c, above, now using the factor effects model with treatment coding. (10 points)

ASK !!!!!!!!!!!!

```
# compute 95% CIs for the difference in effects between the concentration
# of the control group and the concentrations of the remaining groups.
```