Stat 511, HW 4 answers

1. protein soaking data.

   (a) 3 pts. The sequential (type I) tests are:

Source	df	SS	p-value
time	4	42.08	< 0.0001
solution	1	11.26	< 0.0001
time*solution	3	4.48	0.0024
Error	36	9.23	

   Notes: error included only for completeness, not requested as part of your answer. There are three levels of solution (None, A, B) but only 1 d.f. Definitely a clue something is afoot.

   The answers for the next two questions depend on whether you are using SAS or R. They are given separately. Either was accepted for full credit.

   "R" track

   (b) 3 pts. The partial (type III) tests are:

Source	df	SS	p-value
time	0	0.00	
solution	0	0.00	
time*solution	3	4.48	0.0024

   If you forgot to use an orthogonal set of contrasts, e.g. contr.helmet() and used the default contr.treatment(), you got:

Source	df	SS	p-value
time	3	6.36	??
solution	0	0.00	
time*solution	3	4.48	0.0024

   These aren't (even without the missing cell crud) valid type III tests. This answer was marked wrong.

   (c) 2 pts. The reason for the 0 df and 0 SS is apparent when you look at the names of the coefficients from coef() or the column names in the model matrix, head(model.matrix()). The interaction time.f:solution.f generates $8 = (3\text{-}1)*(5\text{-}1)$ columns in the $\boldsymbol{X}$ matrix. The column space of these 8 columns and the intercept includes the column space of the 5 (or 4 if constrained) columns of time.f and the 3 (or 2 if constrained) columns of solution.f. So if you fit the 8 columns of time.f:solution.f, there is nothing left for time.f or solution.f to "explain". Everything has already been fit by time.f:solution.f. Both df and SS are 0.

   "SAS" track: Note: the sequential SS are the same as for the "R" track.

   (b) 3 pts The partial (type III) tests are:

Source	df	SS	p-value
time	3	26.46	< 0.0001
solution	1	11.26	¡0.001
time*solution	3	4.48	0.0024

(c) 2 pts The key here is to realize that time = 0 is exactly the same as solution = 'None'. One level of Time and one of Solution are confounded. If you fit a model with the terms in the order time solution time*solution, the sequential (type I) df for solution (with 3 lavels) is 1 df because time=0 has already accounted for the difference between solution='None' and the others. If you fit a model with the terms in the order solution time time*solution, the sequential (type I) df for time (with 5 levels) is 3 df because solution='None' has already accounted for the difference between time=0 and the other times. The type III SS are equivalent to adding each term after all the other. Hence, the d.f. are 3,1,3.
Note: SAS is "smart" enough to not try to force-fit all columns of the interaction.

(d) 2 pts estimate= -1.88 se = 0.24

(e) 4 pts For solution A: slope = 0.101, se = 0.014
For solution B: slope = 0.195, se = 0.014

Note: If you divide the linear polynomial coefficients by 50, you do get the slope, as you can check by fitting a simple linear regression to the None and A groups or None and B groups. However, I gave you wrong the reason. It has nothing to do with the number of observations per group. That quantity cancels out of the equation for the slope. The same coefficents would be used for X values that increse by 1, by 5, by 100, or by 0.001. The linear polynomial coefficients always increase by 1. To get a slope with the correct units, you need to divide $\sum l_i^2$ by the spacing of the X's. That just happens to be 5 in this problem. That's the correct reason for dividing by $5\sum l_i^2$.

(f) 2 pts T = -5.99, p < 0.0001

2. Sum-to-zero constraints

(a) 2 pts The cell means are:

$$\begin{aligned}\mu_1 &= \mu + \beta_1 \\ \mu_2 &= \mu + \beta_2 \\ \mu_3 &= \mu + \beta_3 \\ \mu_4 &= \mu + \beta_4 = \mu - \beta_1 - \beta_2 - \beta_3\end{aligned}$$

(b) 2 pts. The estimated coefficients are $\hat{\mu} = 173.75$, $\hat{\beta}_1 = -1.75$, $\hat{\beta}_2 = 11.25$, $\hat{\beta}_3 = 2.25$ so:

$$\begin{aligned}\hat{\mu}_1 &= \hat{\mu} + \hat{\beta}_1 = 173.75 + (-1.75) = 172 \\ \hat{\mu}_2 &= \hat{\mu} + \hat{\beta}_2 = 173.75 + 11.25 = 185 \\ \hat{\mu}_3 &= \hat{\mu} + \hat{\beta}_3 = 173.75 + 2.25 = 176 \\ \hat{\mu}_4 &= \hat{\mu} - \hat{\beta}_1 - \hat{\beta}_2 - \hat{\beta}_3 = 173.75 - (-1.75) - 11.25 - 2.25 = 162\end{aligned}$$

My R code:

```
# R code for HW 4
# Problem 1, parts a-c

prot <- read.csv('data/protein.csv', as.is=T)
prot$time.f <- factor(prot$time)
prot$sol.f <- factor(prot$solution)

options(contrasts=c('contr.helmert', 'contr.poly'))
prot.lm2 <- aov(protein~time.f*sol.f, data=prot)

# type I SS
anova(prot.lm2)

# type III SS
drop1(prot.lm2, ~., test='F')

# Problem 1, parts d-f

# reset to treatment contrasts and fit a cell means model
options(contrasts=c('contr.treatment', 'contr.poly'))
prot.lm1 <- aov(protein~-1+time.f:sol.f, data=prot)

# look at names of coef() to see the order of the groups: 0/None is last
c1 <- c(rep(-1,8),8)/8  # no soak vs ave. of rest
c2 <- c(-1, 0, 1, 2, 0,0,0,0,-2)/50  # slope for A
c3 <- c(0,0,0,0,-1,0,1,2,-2)/50      # slope for B
c4 <- c2 - c3                        # diff in slopes

C <- rbind(c1,c2,c3,c4)
ests <- C %*% coef(prot.lm1)

mse <- anova(prot.lm1)[2,3]
  # location of MSE in anova table depends on model
se <- sqrt(mse*apply(C^2,1,sum)/5)

cbind(ests=ests, se=se)

# calculate T statistic then p-value for 1f:
ests[4]/se[4]
2*pt(-abs(ests[4]/se[4]), 36)
```

```
# Problem 2b
dn <- read.table('data/doughnut2.txt', header=T, as.is=T)

dn.lm <- lm(amount ~ b1+b2+b3, data=dn)

cbind(type=dn$type[seq(1,24,6)],mean=predict(dn.lm)[seq(1,24,6)])
```

My SAS code for problem 1:

```
data protein;
  infile 'protein.csv' dsd firstobs=2;
  input time solution $ protein;
  run;

proc glm;
  class solution time;
  model protein = time solution time*solution;
  title 'Problem 1 a-c';
run;

proc glm;
  class solution time;
  model protein = time*solution;
  lsmeans time*solution;
  estimate 'None - Diff ave of any soak' time*solution -1 -1 -1 -1 -1 -1 -1 -1 8
     /divisor=8;
  estimate 'linear slope for A' time*solution -1 0 1 2 0 0 0 0 -2 /divisor=50;
  estimate 'linear slope for B' time*solution 0 0 0 0 -1 0 1 2 -2 /divisor=50;
  estimate 'diff in slopes' time*solution -1 0 1 2 1 0 -1 -2 0 /divisor=50;
  /* or could have used contrast instead of estimate for test in 1f */
  title 'Problem 1 d-f';
run;
```