

Department of Statistics

COURSE STATS 330/762

Model answers for Assignment 4, 2010

1. *Read the data into R, and make a data frame containing the continuous variables `wt.d` and `wt.n`, plus the factors `n.treat` and `current`. Note that only the values for `wt.d` and `wt.n` are given in the data file. You will need to create the factors. The data are in the following order:*

*8 observations with current = "galvanic" and n.treat=1,
8 observations with current = "galvanic" and n.treat=3,
8 observations with current = "galvanic" and n.treat=6,
8 observations with current = "faradic" and n.treat=1,
8 observations with current = "faradic" and n.treat=3,
8 observations with current = "faradic" and n.treat=6,
8 observations with current = "60.cycles" and n.treat=1,
8 observations with current = "60.cycles" and n.treat=3,
8 observations with current = "60.cycles" and n.treat=6,
8 observations with current = "25.cycles" and n.treat=1,
8 observations with current = "25.cycles" and n.treat=3,
8 observations with current = "25.cycles" and n.treat=6,*

for a total of 96 observations. Check for gross errors. Print out the first 16 lines. [5 marks]

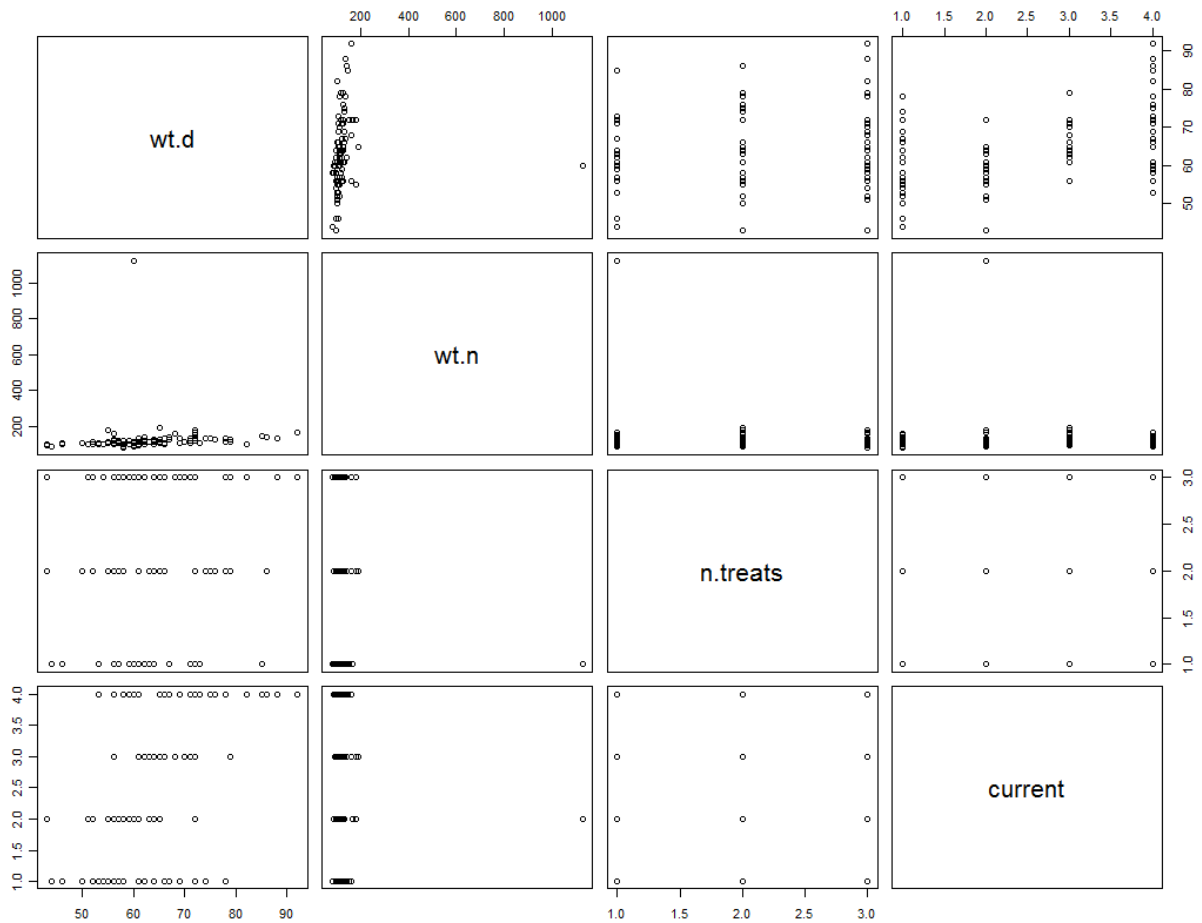
The following code reads in the data frame and adds the two factors. Note the use of `rep`.

```
temp.df =  
read.table("http://www.stat.auckland.ac.nz/~lee/330/datasets.dir/muscle.txt",  
header=TRUE)  
current.levels = c("galvanic", "faradic", "60.cycle", "25.cycle")  
rats.df = data.frame(temp.df,  
n.treats = factor(rep(rep(c(1,3,6), c(8,8,8)),4)),  
current = factor(rep(current.levels, c(24,24,24,24))))
```

Note that we have set the levels of `current` as "galvanic", "faradic", "60.cycle", "25.cycle".

To check the data, we can draw a pairs plot:

```
pairs(rats.df)
```



Looks like there is a big value for wt.n in level 1 of n.treats and level 2 (“faradic”) of current. Inspecting the data reveals that observation 29 has a typo, the value for wt.n should be 126. We correct this:

```
rats.df[29, 2]=126
```

The first 16 lines are

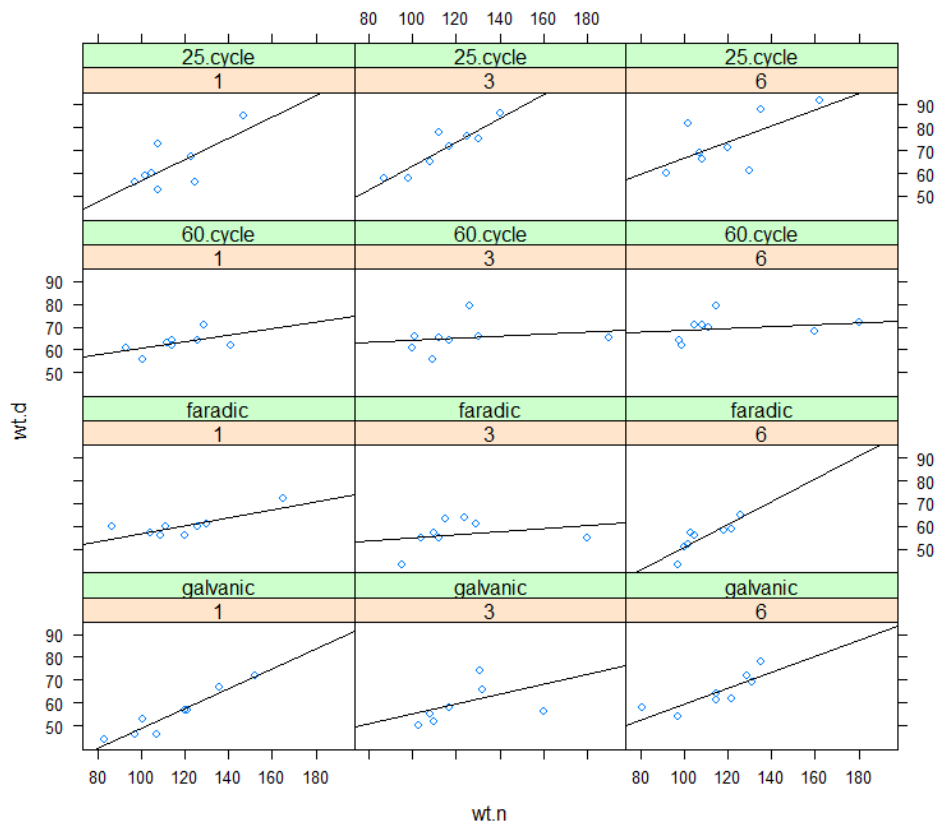
```
rats.df[1:16, ]
  wt.d wt.n n.treats  current
1   72  152         1 galvanic
2   67  136         1 galvanic
3   57  120         1 galvanic
4   57  121         1 galvanic
5   46   97         1 galvanic
6   44   83         1 galvanic
7   53  101         1 galvanic
```

8	46	107	1	galvanic
9	74	131	3	galvanic
10	52	110	3	galvanic
11	66	132	3	galvanic
12	56	160	3	galvanic
13	74	131	3	galvanic
14	58	117	3	galvanic
15	50	103	3	galvanic
16	55	108	3	galvanic

2. Make a suitable plot that will let you see how the number of treatments and the type of current affect the relationship between the variables wt.d and wt.n. What are these effects? [5 marks]

Let's draw a trellis plot, conditioning on the two factors. We also add a least squares line to each panel

```
library(lattice)
xyplot(wt.d~wt.n|n.treats*current, data=rats.df,
panel=function(x,y){
panel.xyplot(x,y)
panel.lmline(x,y)
})
```



From the plot, it seems that the slopes for the 60cycles currents are different from those of the others. The slopes for the other currents are reasonably similar. The number of daily treatments does not seem to have much effect on the slopes. The intercepts seem to have no systematic pattern.

3. *Fit the following models to the data, and choose the one you think fits best. You must provide a justification for your choice. [10 marks]*
- A model with a common slope and intercept for all 12 factor level combinations.*
 - A model with a common slope and but different intercepts depending on the current only*
 - A model with a common slope and **but different** intercepts for all 12 factor level combinations.*
 - A model with different intercepts for all 12 factor level combinations, but the slopes depending on current only.*
 - A model with different intercepts and slopes for all 12 factor level combinations.*

The models are described by the following formulas:

- $wt.d \sim wt.n$
- $wt.d \sim current + wt.n$
- $wt.d \sim ntreat * current + wt.n$
- $wt.d \sim ntreat * current + wt.n * current$
- $wt.d \sim ntreat * current * wt.n$

We can fit the full model d) and see which terms are significant in the anove table:

```
> rats.lm = lm(wt.d~n.treats*current*wt.n, data=rats.df)
> anova(rats.lm)
Analysis of Variance Table
Response: wt.d
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
n.treats	2	447.44	223.72	5.6620	0.0052029	**
current	3	2145.45	715.15	18.0994	7.465e-09	***
wt.n	1	1890.66	1890.66	47.8499	1.569e-09	***
n.treats:current	6	487.55	81.26	2.0566	0.0690436	.
n.treats:wt.n	2	165.24	82.62	2.0910	0.1309998	
current:wt.n	3	794.27	264.76	6.7006	0.0004727	***
n.treats:current:wt.n	6	206.92	34.49	0.8728	0.5192596	
Residuals	72	2844.88	39.51			

This suggests that models (c) or (d) might be appropriate, as indicated by the trellis plot. We can also try a stepwise approach:

```
step(lm(wt.d~1, data=rats.df), formula(rats.lm), direction="both")
```

This results in model (d).

Call:

```
lm(formula = wt.d ~ wt.n + current + n.treats + wt.n:current +  
current:n.treats, data = rats.df)
```

The AIC values are

```
> AIC(rats.lm)  
[1] 647.7733  
> AIC(lm(wt.d~n.treats*current + wt.n*current, data=rats.df))  
[1] 640.4349  
> AIC(lm(wt.d~n.treats*current + wt.n, data=rats.df))  
[1] 658.7586  
> AIC(lm(wt.d~current + wt.n, data=rats.df))  
[1] 662.7128  
> AIC(lm(wt.d~wt.n, data=rats.df))  
[1] 690.916
```

Accordingly, we will go with model (d). Diagnostic plots do not indicate any serious problems. There are a couple of outliers but they do not seem to have an appreciable effect on the fitted model.

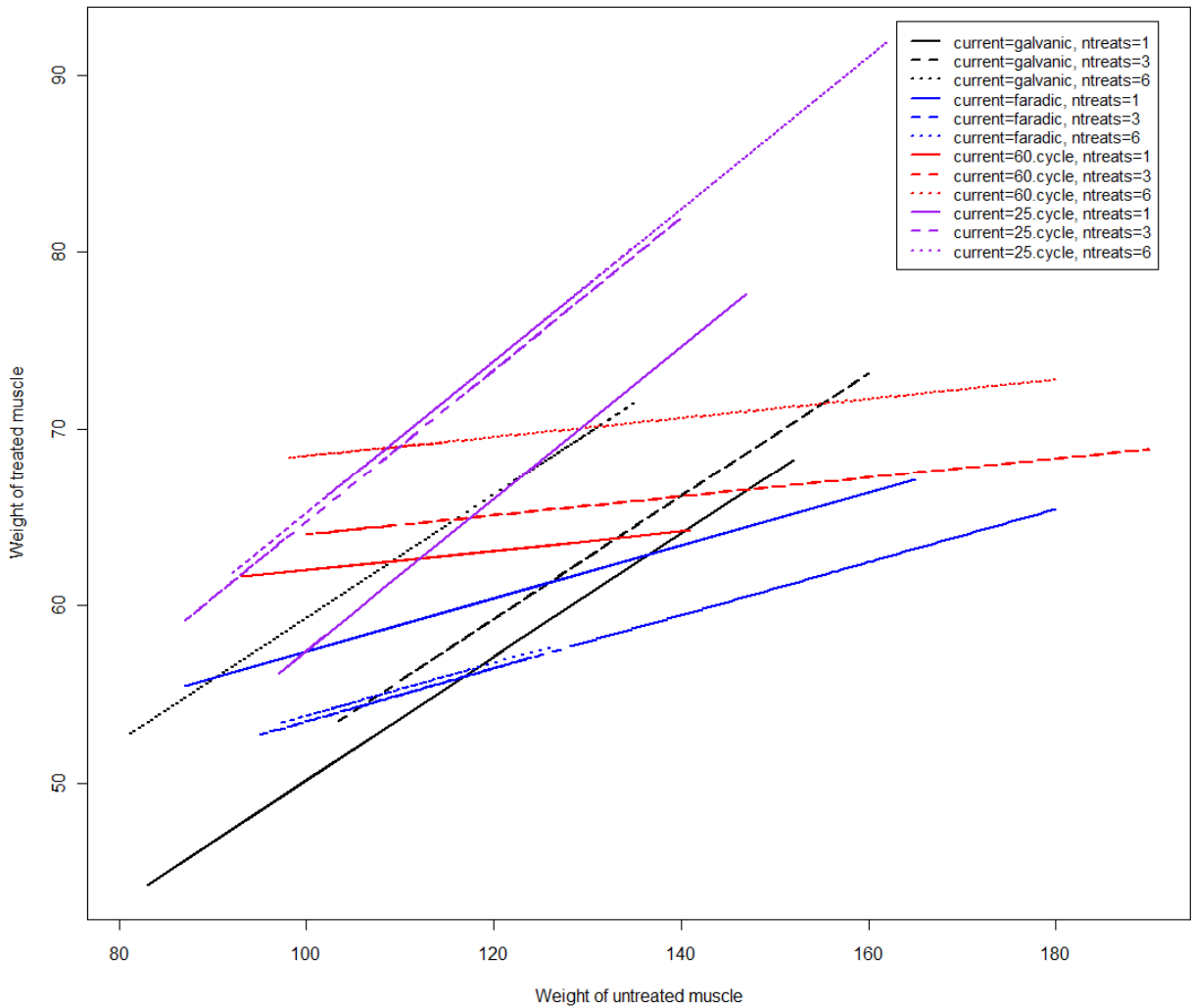
4. *For the model you have chosen, draw a plot showing the 12 fitted lines (which could coincide). Comment on the effect of the factors on the weight of the treated muscle. [10 marks]*

The following code will draw the plot:

```
rats2.lm = lm(wt.d~ n.treats*current+ wt.n*current, data=rats.df)  
  
y.pred=predict(rats2.lm)  
plot(rats.df$wt.n, y.pred, type="n", xlab="Weight of untreated  
muscle",  
ylab = "Weight of treated muscle")  
plot.range = 1:8; my.col = c("black","blue","red", "purple")  
  
for(j in 1:4){  
  for(i in 1:3){  
    lines(rats.df$wt.n[plot.range], y.pred[plot.range], lty=i,  
          col=my.col[j], lwd=2)  
    plot.range = plot.range+8  
  }  
}  
  
legend.text = c("current=galvanic, ntreats=1",
```

```
"current=galvanic, ntreats=3",
"current=galvanic, ntreats=6",
"current=faradic, ntreats=1",
"current=faradic, ntreats=3",
"current=faradic, ntreats=6",
"current=60.cycle, ntreats=1",
"current=60.cycle, ntreats=3",
"current=60.cycle, ntreats=6",
"current=25.cycle, ntreats=1",
"current=25.cycle, ntreats=3",
"current=25.cycle, ntreats=6")
```

```
legend(163,93, legend.text, col=rep(my.col,c(3,3,3,3)),
lty=rep(1:3,4), lwd=2, cex=0.7)
```



The plot shows that the weight of the treated muscle increases as the untreated weight increases more slowly for the 60 cycle current, followed by the faradic current. The other two currents seem alike in their effects. For a fixed untreated weight, the treated weight

goes up with the number of treatments. Which current leads to the largest treated muscle weight depends on the current: for large untreated weights, the 25-cycle current gives the largest treated weights, while for small untreated weights, the treated weights for the 60-cycle current gives similar treated weights to the 25-cycle current.

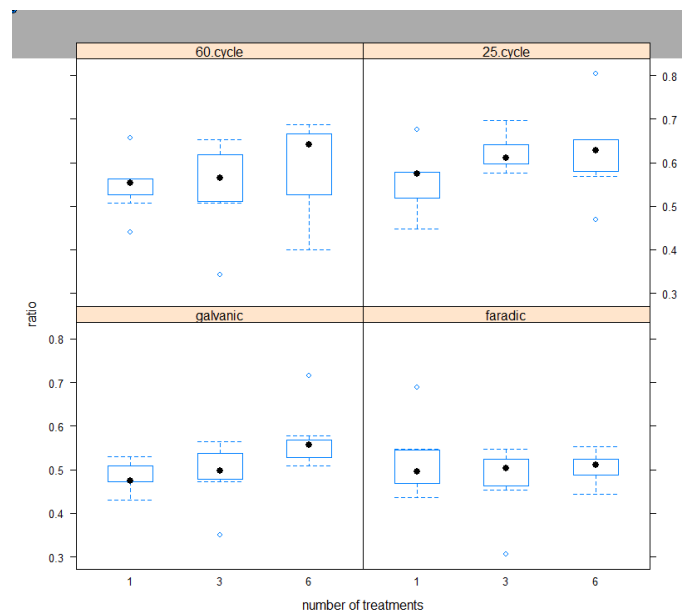
- Now make a new variable which is the ratio of the weights of the treated and untreated muscles. Draw a graph or graphs that show the effect of the factors on the ratio. Fit a two-way anova model to these ratios. Do the factors interact? What is the effect of the factors on the ratios? [10 marks]

The following code makes the new variable and adds it to the data frame:

```
ratio = rats.df$wt.d/rats.df$wt.n
newrats.df = data.frame(rats.df, ratio)
```

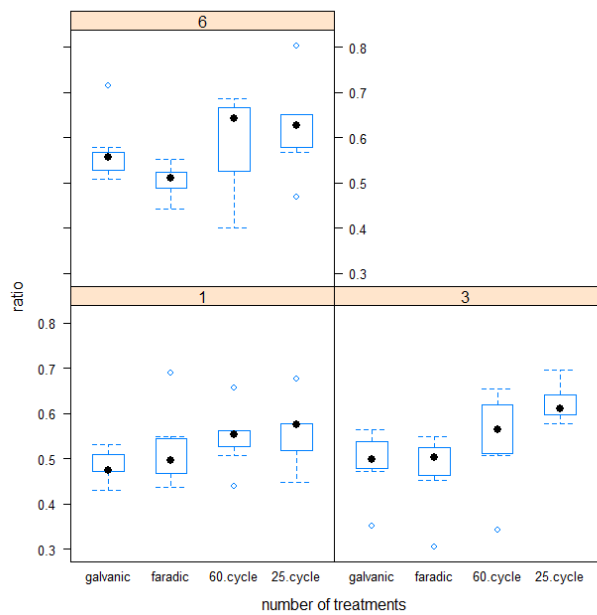
We can draw trellis plots, conditioning on each factor in turn:

```
bwplot(ratio~n.treatments|current, data=newrats.df, xlab="number of treatments")
```



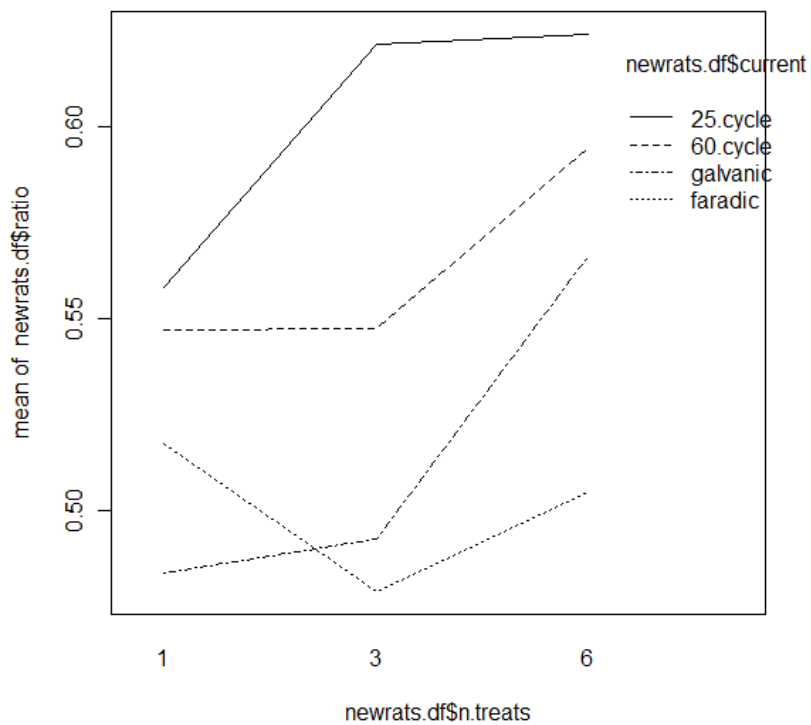
and

```
bwplot(ratio~current|n.treatments, data=newrats.df, xlab="number of treatments")
```



From these we see that the ratio is going up as the current changes from galvanic through to 25.cycle, and as the number of treatment increases.

The interaction plot is



Which suggests that some interaction might be present, as the lines are not particularly parallel. However, in the the anova table (obtained using the code)

```
> newrats.lm = lm(ratio~ n.treats*current, data=newrats.df)
> anova(newrats.lm)
Analysis of Variance Table

Response: ratio
      Df Sum Sq Mean Sq F value    Pr(>F)
n.treats      2  0.03797  0.018987   3.5338  0.03361 *
current       3  0.15345  0.051150   9.5199 1.762e-05 ***
n.treats:current  6  0.03503  0.005838   1.0866  0.37715
Residuals    84  0.45133  0.005373
```

the main effects are significant but there seems to be no evidence of interaction. Thus, the non-parallel appearance of the interaction plot could be due to sampling variation. The summary for the no-interaction model is

```
> newrats2.lm = lm(ratio~ n.treats+current, data=newrats.df)
> summary(newrats2.lm) Call:
lm(formula = ratio ~ n.treats + current, data = newrats.df)
```

```
Residuals:
      Min       1Q   Median       3Q      Max
-0.2114085 -0.0231852 -0.0005157  0.0319604  0.2072064
```

```
Coefficients:
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.496217   0.018378  27.001 < 2e-16 ***
n.treats3    0.008622   0.018378   0.469  0.6401
n.treats6    0.045836   0.018378   2.494  0.0145 *
currentfaradic -0.013768   0.021221  -0.649  0.5181
current60.cycle  0.048674   0.021221   2.294  0.0241 *
current25.cycle  0.086834   0.021221   4.092 9.31e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.07351 on 90 degrees of freedom
Multiple R-squared:  0.2824,    Adjusted R-squared:  0.2426
F-statistic: 7.085 on 5 and 90 DF,  p-value: 1.258e-05
```

The summary confirms the story in the plots: 6 treatments are significantly better than 1 or 3, and 60.cycles or 25.cycles are better than galvanic or faradic.

762 question

For the setup in Q5, calculate confidence intervals for the means of each of the 12 factor combinations, based on the model indicated by the result of Q5 above. Hint: use the function `test.lc` in the “330.functions”.

The means are

For "current=galvanic, ntreats=1", the intercept

For "current=galvanic, ntreats=3", the intercept + the main effect
for ntreats=3

For "current=galvanic, ntreats=6", the intercept + the main effect
for ntreats=6

For "current=faradic, ntreats=1", the intercept + the main effect for
current=faradic

For "current=faradic, ntreats=3", the intercept + the main effect
for current=faradic+ the main effect for ntreats=3

For "current=faradic, ntreats=6", the intercept + the main effect
for current=faradic + the main effect for ntreats=6

For "current=60.cycle, ntreats=1", the intercept + the main effect
for current=60.cycle

For "current=60.cycle, ntreats=3", the intercept + the main effect
for current=60.cycle + the main effect for ntreats=3

For "current=60.cycle, ntreats=6", the intercept + the main effect
for current=60.cycle + the main effect for ntreats=6

For "current=25.cycle, ntreats=1", the intercept + the main effect
for current=25.cycle

For "current=25.cycle, ntreats=3", the intercept + the main effect
for current=25.cycle + the main effect for ntreats=3

For "current=25.cycle, ntreats=6", the intercept + the main effect
for current=25.cycle + the main effect for ntreats=6

In terms of the parameters in the model, these are following linear combinations (see overleaf)

	Intercept	n.treats3	ntreats6	currentfaradic	current 60.cycle	current25.cycle
mean						
1	1	0	0	0	0	0
2	1	1	0	0	0	0
3	1	0	1	0	0	0
4	1	0	0	1	0	0
5	1	1	0	1	0	0
6	1	0	1	1	0	0
7	1	0	0	0	1	0
8	1	1	0	0	1	0
9	1	0	1	0	1	0
10	1	0	0	0	0	1
11	1	1	0	0	0	1
12	1	0	1	0	0	1

First, make a matrix containing the coefficients:

```
> coef.mat=matrix(scan(), ncol=6, byrow=TRUE)
```

```
1: 1 0 0 0 0 0
```

```
7: 1 1 0 0 0 0
```

```
13: 1 0 1 0 0 0
```

```
19: 1 0 0 1 0 0
```

```
25: 1 1 0 1 0 0
```

```
31: 1 0 1 1 0 0
```

```
37: 1 0 0 0 1 0
```

```
43: 1 1 0 0 1 0
```

```
49: 1 0 1 0 1 0
```

```
55: 1 0 0 0 0 1
```

```
61: 1 1 0 0 0 1
```

```
67: 1 0 1 0 0 1
```

```
73:
```

```
Read 72 items
```

```
> coef.mat
```

```
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    0    0    0    0    0
[2,]    1    1    0    0    0    0
[3,]    1    0    1    0    0    0
[4,]    1    0    0    1    0    0
[5,]    1    1    0    1    0    0
[6,]    1    0    1    1    0    0
[7,]    1    0    0    0    1    0
[8,]    1    1    0    0    1    0
[9,]    1    0    1    0    1    0
[10,]   1    0    0    0    0    1
[11,]   1    1    0    0    0    1
[12,]   1    0    1    0    0    1
```

Then, calculate the intervals using the test.lc function:

```
> std.errs=numeric(12)
```

```
> est =numeric(12)
```

```

> for(i in 1:12) {
+ stuff = test.lc(coef.mat[i,], 0, newrats2.lm)
+ std.errs[i] = stuff$std.err
+ est[i]=stuff$est
+ }
> t.val = qt(0.975,90)
> intervals = cbind(est - t.val*std.errs, est + t.val*std.errs)
> dimnames(intervals) = list(paste("Mean",1:12), c("LCL", "UCL"))
> intervals

```

	LCL	UCL
Mean 1	0.4597059	0.5327280
Mean 2	0.4683284	0.5413504
Mean 3	0.5055415	0.5785636
Mean 4	0.4459377	0.5189598
Mean 5	0.4545602	0.5275822
Mean 6	0.4917733	0.5647954
Mean 7	0.5083803	0.5814023
Mean 8	0.5170027	0.5900248
Mean 9	0.5542159	0.6272380
Mean 10	0.5465399	0.6195620
Mean 11	0.5551624	0.6281844
Mean 12	0.5923755	0.6653976