MAST30025 Linear Statistical Models (Assignment 3)

Brendan Hill - Student 699917 (Prac group - Monday 2:15pm)

August 21, 2018

1. (a) Consider the matrix $A(A^TA)^cA^T$. As proven in lectures, this is symmetric (*). Recall that in general, $A = A(A^TA)^c(A^TA)$ (**).

   Denote two possibly distinct conditional inverses of A^TA as $(A^TA)^{c_1}$ and $(A^TA)^{c_2}$.

   Now consider the difference of $A(A^TA)^{c_1}A^T - A(A^TA)^{c_2}A^T$:

$$\begin{aligned}
A(A^TA)^{c_1}A^T - A(A^TA)^{c_2}A^T &= [A(A^TA)^c(A^TA)](A^TA)^{c_1}A^T - [A(A^TA)^c(A^TA)](A^TA)^{c_2}A^T && (**)\\
&= (A(A^TA)^c(A^TA)(A^TA)^{c_1}A^T)^T - (A(A^TA)^c(A^TA)(A^TA)^{c_2}A^T)^T && (*)\\
&= A((A^TA)^{c_1})^T(A^TA)^T((A^TA)^c)^TA^T - A((A^TA)^{c_2})^T(A^TA)^T((A^TA)^c)^TA^T\\
&= A((A^TA)^T)^{c_1}(A^TA)((A^TA)^T)^cA^T - A((A^TA)^T)^{c_2}(A^TA)((A^TA)^T)^cA^T\\
&= \underbrace{A(A^TA)^{c_1}(A^TA)}_{=A,\text{ by }(**)}(A^TA)^cA^T - \underbrace{A(A^TA)^{c_2}(A^TA)}_{=A,\text{ by }(**)}(A^TA)^cA^T\\
&= A(A^TA)^cA^T - A(A^TA)^cA^T\\
&= 0
\end{aligned}$$

   Hence $A(A^TA)^{c_1}A^T = A(A^TA)^{c_2}A^T$ for arbitrary conditional invoices c_1, c_2, making the expression invariant to the choice of conditional invoice.

   (b) Recall that in general, $r(AB) \leq \min\{r(A), r(B)\} \leq r(A), (*)$, so:

$$r(A(A^TA)^cA^T) \overset{(*)}{\leq} r(A) = r(A(A^TA)^c(A^TA)) \overset{(*)}{\leq} r(A(A^TA)^cA^T)$$

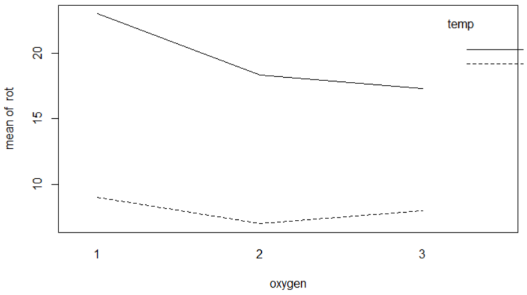
   So $r(A(A^TA)^TA^T) = r(A)$

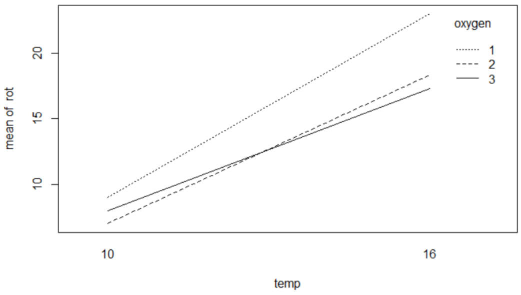
2. (a) The interaction plots are generated with the following code:

```
   data <- data.frame(
  rot =          c(13,11,3, 10,4, 7, 15,2, 7, 26,19,24,15,22,18,20,24,8),
  oxygen = factor(c(1, 1, 1, 2, 2, 2, 3, 3, 3, 1, 1, 1, 2, 2, 2, 3, 3, 3)),
  temp =   factor(c(10,10,10,10,10,10,10,10,10,16,16,16,16,16,16,16,16,16))
)

with(data, interaction.plot(oxygen, temp, rot))
with(data, interaction.plot(temp, oxygen, rot))
```

Which generates the following plots:





In each graph, the lines are approximately parallel. This suggests that the data does not contain strong evidence of interaction between the factors.

(b) The model is $y = X\beta + \epsilon$ where $\beta = \begin{bmatrix}\mu & \tau_1 & \tau_2 & \tau_3 & \beta_1 & \beta_2\end{bmatrix}^T$.
The model is loaded and estimated using the following code:

```
   > y <- data$rot
> n = length(data$rot)
> p =  1 + nlevels(data$oxygen) + nlevels(data$temp)
> X = matrix(0, n, p)
> X[,1] <- 1
> X[,2] <- data$oxygen == 1
> X[,3] <- data$oxygen == 2
> X[,4] <- data$oxygen == 3
> X[,5] <- data$temp == 10
> X[,6] <- data$temp == 16
> X
      [,1] [,2] [,3] [,4] [,5] [,6]
 [1,]    1    1    0    0    1    0
 [2,]    1    1    0    0    1    0
 [3,]    1    1    0    0    1    0
 [4,]    1    0    1    0    1    0
 [5,]    1    0    1    0    1    0
 [6,]    1    0    1    0    1    0
 [7,]    1    0    0    1    1    0
 [8,]    1    0    0    1    1    0
 [9,]    1    0    0    1    1    0
[10,]    1    1    0    0    0    1
[11,]    1    1    0    0    0    1
[12,]    1    1    0    0    0    1
[13,]    1    0    1    0    0    1
[14,]    1    0    1    0    0    1
[15,]    1    0    1    0    0    1
[16,]    1    0    0    1    0    1
[17,]    1    0    0    1    0    1
[18,]    1    0    0    1    0    1
>
> library(Matrix)
> r = rankMatrix(X)[1]
> r
```

```
[1] 4
>
> XTX = t(X) %*% X
> M = XTX[3:6, 3:6]
> XTXc = matrix(0, dim(XTX)[1], dim(XTX)[2])
> XTXc[3:6,3:6] <- t(solve(M))
> XTXc <- t(XTXc)
> b = XTXc %*% t(X) %*% y
> b
          [,1]
[1,]  7.515152
[2,]  4.727273
[3,]  1.393939
[4,]  1.393939
[5,] -2.020202
[6,]  9.535354
>
> SSRes = sum((y - X %*% b)^2)
> SSRes
[1] 365.7778
>
> s2 = SSRes / (n-r)
> s2
[1] 26.12698
```

So the design matrix, estimated b and estimated common variance s^2 are:

$$X = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \qquad b = \begin{bmatrix} 7.515152 \\ 4.727273 \\ 1.393939 \\ 1.393939 \\ -2.020202 \\ 9.535354 \end{bmatrix} \qquad s^2 = 26.12698$$

(c) The difference between the temperature effects is $(\beta_1 + \mu) - (\beta_2 + \mu) = \beta_1 - \beta_2$.
This following code confirms that this quantity is estimable (since $t = [0, 0, 0, 0, 1, -1] = t(X^TX)^tX^TX$):

```
> t = c(0,0,0,0,1,-1)
> round(t - t %*% XTXc %*% t(X) %*% X, 5)
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    0    0    0    0    0
```

Thus a 95% confidence interval for t is:

```
> c(t(t) %*% b - qt(0.975, n-r) * sqrt(s2) * sqrt(t(t) %*% XTXc %*% t),
+   t(t) %*% b + qt(0.975, n-r) * sqrt(s2) * sqrt(t(t) %*% XTXc %*% t))
[1] -16.723555  -6.387556
```

That is, we are 95% confident that the true difference between the temperatures $(\beta_1 - \beta_2)$ is in the interval $[-16.723555, -6.387556]$

(d) The hypothesis that oxygen level has no effect on rotting can be expressed as the hypothesis $H_0 : \tau_1 = \tau_2 = \tau_3$.
The hypothesis matrix is confirmed to be made up of estimable quantities:

```
> C = matrix(c(0,1,-1,0,0,0,0,1,0,-1,0,0),2,6, byrow=TRUE)
> C
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    1   -1    0    0    0
[2,]    0    1    0   -1    0    0
> round(C - C %*% XTXc %*% XTX, 5)
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    0    0    0    0    0
[2,]    0    0    0    0    0    0
```

The F statistic and critical value are found as follows:

```
> m = rankMatrix(C)[1]
> m
[1] 2
> FStat = (t(C %*% b) %*% solve(C %*% XtXc %*% t(C)) %*% C %*% b) / (s2 * m)
> criticalFvalue = qf(0.95, m, n-r)
> FStat
          [,1]
[1,] 0.8505468
> criticalFvalue
[1] 3.738892
```

Since the F statistic $0.8505468 < 3.738892$, we do not reject H_0 at the 95% confidence level.

(e) A design which blocks on temperature would achieve this, eg. a CBD.

3. (a) Let $A_{a,b}$ be the set of equations $\xi_{ij} - \xi_{1j} - \xi_{i1} + \xi_{11} = 0, i = 2, \ldots, a, j = 2, \ldots, b$.

Clearly each ξ_{ij} for $i = 2, \ldots, a, j = 2, \ldots, b$ appears in exactly one equation in $A_{a,b}$.

Hence if $A_{a,b}$ is in matrix form, the columns relating to $\xi_{ij}, i = 2, \ldots, a, j = 2, \ldots, b$ form an identity matrix.

THis means no row in $A_{a,b}$ can be a linear combination of the other rows.

Hence none of the equations in $A_{a,b}$ is redundant.

(b) Let $A_{a,b}$ be the set of equations:

$$\xi_{ij} - \xi_{1j} - \xi_{i1} + \xi_{11} = 0, \qquad i = 2, \ldots, a, \qquad j = 2, \ldots, b$$

Let $B_{a,b}$ be the full set of equations in the hypothesis of no interaction:

$$\xi_{ij} - \xi_{i'j} - \xi_{ij'} + \xi_{i'j'} = 0, \qquad i, i' = 1, \ldots, a, i \neq i', \qquad j, j' = 2, \ldots, b, j \neq j'$$

Consider an arbitrary equation $S \in B_{a,b} \setminus A_{a,b}$. Note that for the i, i', j, j' specified in S, $i \neq 1, i' \neq 1, i \neq 1, j' \neq 1$, and by construction $A_{a,b}$ contains the following equations:

$$\begin{aligned}
S_1 &: \xi_{ij} - \xi_{1j} - \xi_{i1} + \xi_{11} = 0 \\
S_2 &: \xi_{ij'} - \xi_{1j'} - \xi_{i1} + \xi_{11} = 0 \\
S_3 &: \xi_{i'j} - \xi_{1j} - \xi_{i'1} + \xi_{11} = 0 \\
S_4 &: \xi_{i'j'} - \xi_{1j'} - \xi_{i'1} + \xi_{11} = 0
\end{aligned}$$

However the equation formed by $S_1 - S_2 - S_3 + S_4$:

$$(\xi_{ij} - \xi_{1j} - \xi_{i1} + \xi_{11}) - (\xi_{ij'} - \xi_{1j'} - \xi_{i1} + \xi_{11}) - (\xi_{i'j} - \xi_{1j} - \xi_{i'1} + \xi_{11}) + (\xi_{i'j'} - \xi_{1j'} - \xi_{i'1} + \xi_{11}) = 0$$

is precisely S:

$$\xi_{ij} - \xi_{i'j} - \xi_{ij'} + \xi_{i'j'} = 0$$

Hence, S is implied by $A_{a,b}$.

Since all equations in $B_{a,b}$ are either contained in or implied by $A_{a,b}$, $A_{a,b}$ is sufficient to define the hypothesis of no interaction.

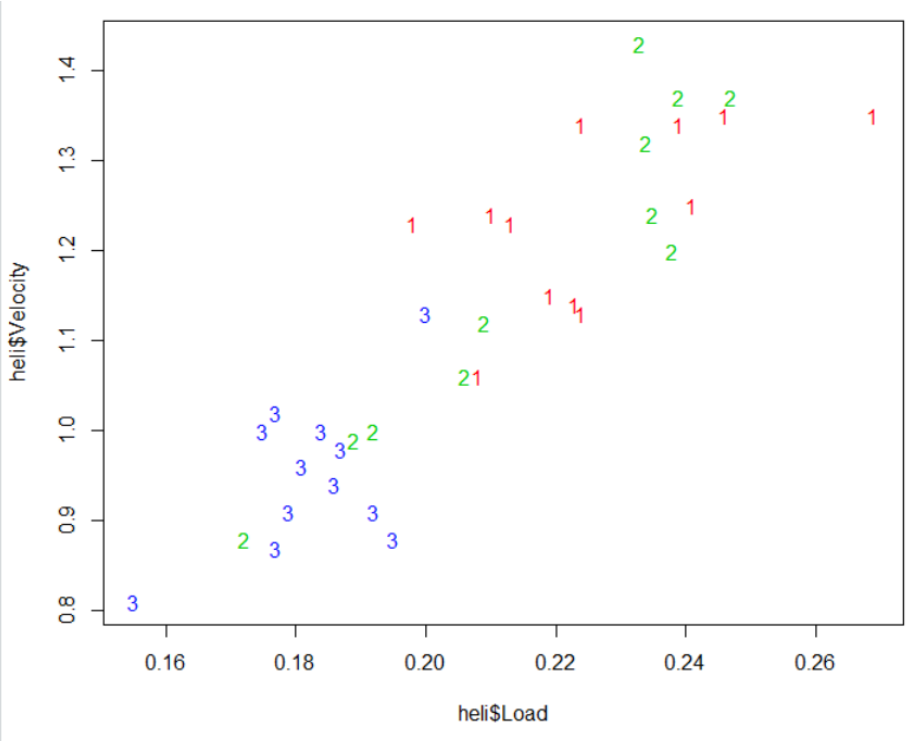
(c) Since $A_{a,b}$ is of full rank and defines the hypothesis of no interaction, the rank of this hypothesis is the number of equations in $A_{a,b}$ which is $(a-1)(b-1)$.

(d) This one looked terribly difficult and I ran out of time and also the cat ate my homework.

4. (a) The plot is generate using the following code:

```
    heli = read.csv(”./heli.csv”)
heli$Tree.f <- factor(heli$Tree)
plot(heli$Load, heli$Velocity, pch=array(heli$Tree.f), col=(heli$Tree+1))
```

Plot of Velocity vs. Load, per tree:



The data overall suggests a linear relationship between Velocity and Load. Tree 1 appears to have higher load/velocity, Tree 2 is spread out, and Tree 3 has lower load/velocity.

(b) The presence of interaction is test for with this script:

```
> imodel <- lm(Velocity ~ Load + Tree.f + Load : Tree.f, data=heli)
> anova(imodel)
Analysis of Variance Table

Response: Velocity
            Df  Sum Sq Mean Sq F value    Pr(>F)
Load         1 0.84364 0.84364 147.835 6.58e-13 ***
Tree.f       2 0.01132 0.00566   0.992  0.38306
Load:Tree.f  2 0.03795 0.01897   3.325  0.05011 .
Residuals   29 0.16549 0.00571
---
Signif. codes:  0   ***   0.001   **   0.01   *   0.05   .   0.1       1
```

The interaction term has F statistic 0.05011 which is just above the threshold for affirming interaction at the 95% confidence level. However, the presence of interaction would be accepted at the 90% confidence level.

(c) Backwards elimination:

```
> reducedmodel <- lm(Velocity ~ Load + Tree.f + Load : Tree.f, data=heli)
> drop1(reducedmodel, scope=~., test=”F”)
Single term deletions

Model:
Velocity ~ Load + Tree.f + Load:Tree.f
            Df Sum of Sq     RSS     AIC F value   Pr(>F)
<none>                   0.16549 -175.40
Load         1  0.039792 0.20528 -169.85  6.9729 0.01319 *
Tree.f       2  0.037553 0.20305 -172.24  3.2903 0.05154 .
```

```
Load:Tree.f   2   0.037949  0.20344  -172.17   3.3250  0.05011  .
---
Signif. codes:  0  ***  0.001  **  0.01  *  0.05  .  0.1  1
> reducedmodel <- lm(Velocity ~ Tree.f + Tree.f : Load, data=heli)
> drop1(reducedmodel, scope=~., test="F")
Single term deletions

Model:
Velocity ~ Tree.f + Tree.f:Load
             Df Sum of Sq      RSS      AIC F value     Pr(>F)
<none>                     0.16549  -175.40
Tree.f        2   0.03755  0.20305  -172.24   3.2903   0.05154 .
Tree.f:Load   3   0.35349  0.51898  -141.39  20.6480  2.343e-07 ***
---
Signif. codes:  0  ***  0.001  **  0.01  *  0.05  .  0.1  1
> reducedmodel <- lm(Velocity ~ Tree.f : Load, data=heli)
> drop1(reducedmodel, scope=~., test="F")
Single term deletions

Model:
Velocity ~ Tree.f:Load
             Df Sum of Sq      RSS      AIC F value     Pr(>F)
<none>                     0.20305  -172.24
Tree.f:Load   3   0.85535  1.05840  -120.45   43.531  3.165e-11 ***
---
Signif. codes:  0  ***  0.001  **  0.01  *  0.05  .  0.1  1
```

Hence in the final model, only the interaction term of Tree:Load is preserved.

(d) The factored regression lines from the full interaction model is generated with this script:

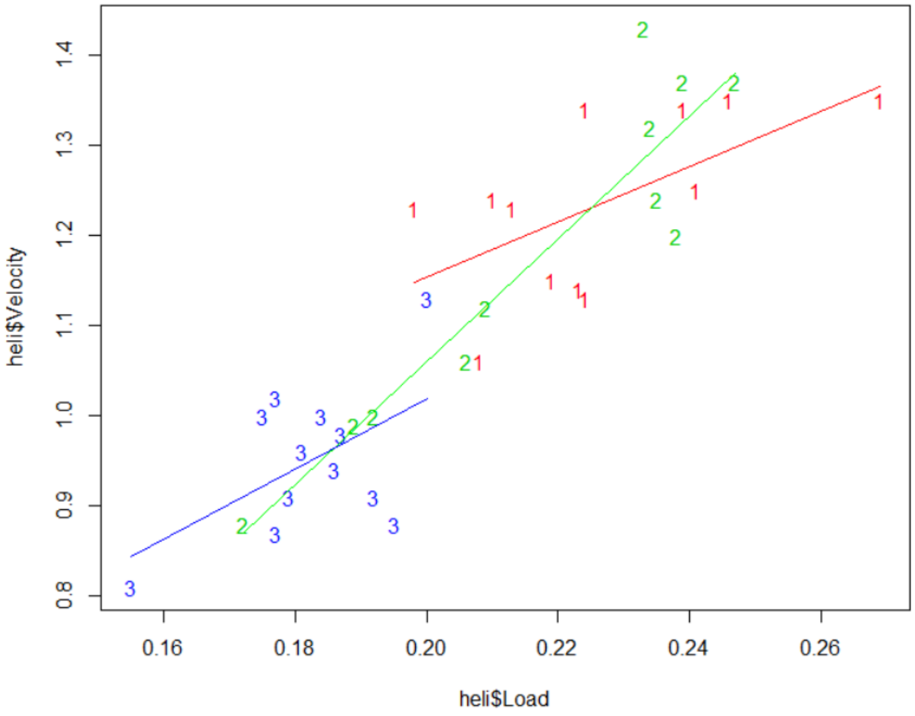
```
#FULL MODEL
plot(heli$Load, heli$Velocity, pch=array(heli$Tree.f), col=(heli$Tree+1))
b = imodel$coefficients

Tree1FromX = min(heli[heli$Tree==1,]$Load)
Tree1ToX = max(heli[heli$Tree==1,]$Load)
Tree1FromY = b[1] + b[2] * Tree1FromX
Tree1ToY = b[1] + b[2] * Tree1ToX
segments(Tree1FromX, Tree1FromY, Tree1ToX, Tree1ToY, col = "red")

Tree2FromX = min(heli[heli$Tree==2,]$Load)
Tree2ToX = max(heli[heli$Tree==2,]$Load)
Tree2FromY = b[1] + b[2] * Tree2FromX + b[3] + b[5] * Tree2FromX
Tree2ToY = b[1] + b[2] * Tree2ToX + b[3] + b[5] * Tree2ToX
segments(Tree2FromX, Tree2FromY, Tree2ToX, Tree2ToY, col = "green")

Tree3FromX = min(heli[heli$Tree==3,]$Load)
Tree3ToX = max(heli[heli$Tree==3,]$Load)
Tree3FromY =  b[1] + b[2] * Tree3FromX + b[4] + b[6] * Tree3FromX
Tree3ToY = b[1] + b[2] * Tree3ToX + b[4] + b[6] * Tree3ToX
segments(Tree3FromX, Tree3FromY, Tree3ToX, Tree3ToY, col = "blue")
```

Resulting in this graph:



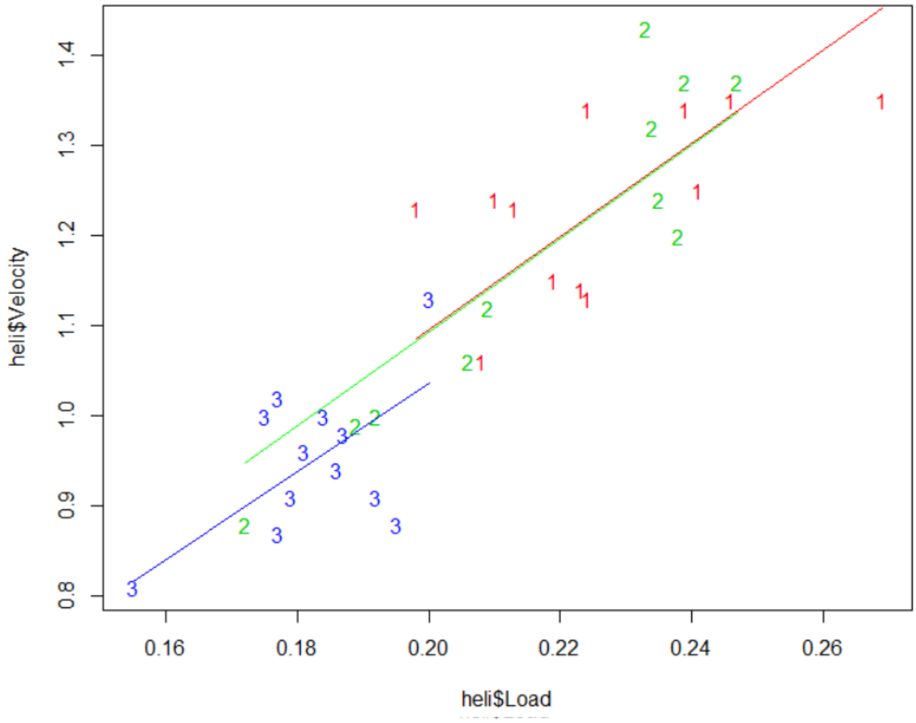
The factored regression lines from the reduced interaction model is generated with this script:

```
# REDUCED MODEL
plot(heli$Load, heli$Velocity, pch=array(heli$Tree.f), col=(heli$Tree+1))
b2 = reducedmodel$coefficients
Tree1FromX = min(heli[heli$Tree==1,]$Load)
Tree1ToX = max(heli[heli$Tree==1,]$Load)
Tree1FromY = b2[1] + b2[2] * Tree1FromX
Tree1ToY = b2[1] + b2[2] * Tree1ToX
segments(Tree1FromX, Tree1FromY, Tree1ToX, Tree1ToY, col = "red")

Tree2FromX = min(heli[heli$Tree==2,]$Load)
Tree2ToX = max(heli[heli$Tree==2,]$Load)
Tree2FromY = b2[1] +  b2[3] * Tree2FromX
Tree2ToY = b2[1] + b2[3] * Tree2ToX
segments(Tree2FromX, Tree2FromY, Tree2ToX, Tree2ToY, col = "green")

Tree3FromX = min(heli[heli$Tree==3,]$Load)
Tree3ToX = max(heli[heli$Tree==3,]$Load)
Tree3FromY =  b2[1] + b2[4] * Tree3FromX
Tree3ToY = b2[1] + b2[4] * Tree3ToX
segments(Tree3FromX, Tree3FromY, Tree3ToX, Tree3ToY, col = "blue")
```

Resulting in this graph:



(e) A 95% confidence interval for the average descent velocity of a disk with loading 0.2 from tree 2 is found from the full interaction model with this code:

```
> xnew = data.frame(Load=0.2, Tree =2, Tree.f = factor(2))
> predict(imodel, xnew, interval="confidence", level=0.95)
       fit       lwr       upr
1 1.060123 1.00227 1.117976
```

So the confidence interval is [1.00227, 1.117976].

So we can reject the hypothesis that the average descent velocity is 1.0 at the 95% confidence level. (But only just!)

5. For both problems I block off the trees into 16 horizontal plots each of size 2:

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

(a) The 16 plots are allocated to the treatments in a complete randomized design (CRD) with the following code:

```
> TREES=32
> PLOTS=16
> TREATMENTS = 4
> s = ceiling(sample(PLOTS, PLOTS)/TREATMENTS)
> s
 [1] 2 3 2 4 3 1 4 2 4 1 4 3 3 1 1 2
> treatments = c(rbind(s,s)) # duplicates each element in s
> treatments
 [1] 2 2 3 3 2 2 4 4 3 3 1 1 4 4 2 2 4 4 1 1 4 4 3 3 3 3 1 1 1 1 2 2
> X = cbind(1, treatments == 1, treatments == 2, treatments == 3, treatments == 4)
> X
      [,1] [,2] [,3] [,4] [,5]
 [1,]    1    0    1    0    0
 [2,]    1    0    1    0    0
 [3,]    1    0    0    1    0
 [4,]    1    0    0    1    0
 [5,]    1    0    1    0    0
 [6,]    1    0    1    0    0
 [7,]    1    0    0    0    1
 [8,]    1    0    0    0    1
 [9,]    1    0    0    1    0
[10,]    1    0    0    1    0
[11,]    1    1    0    0    0
[12,]    1    1    0    0    0
[13,]    1    0    0    0    1
[14,]    1    0    0    0    1
[15,]    1    0    1    0    0
[16,]    1    0    1    0    0
[17,]    1    0    0    0    1
[18,]    1    0    0    0    1
[19,]    1    1    0    0    0
[20,]    1    1    0    0    0
[21,]    1    0    0    0    1
[22,]    1    0    0    0    1
[23,]    1    0    0    1    0
[24,]    1    0    0    1    0
[25,]    1    0    0    1    0
[26,]    1    0    0    1    0
[27,]    1    1    0    0    0
[28,]    1    1    0    0    0
[29,]    1    1    0    0    0
[30,]    1    1    0    0    0
[31,]    1    0    1    0    0
[32,]    1    0    1    0    0
```

So the resulting model is

$$y = X\beta + \epsilon$$

where y_i contains the individual growth result of tree i, ϵ_i is the error term of the i'th tree, and

$$\beta = \begin{bmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix}$$

$$X = \begin{pmatrix}
1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0
\end{pmatrix}$$

(b) Suppose the ground slopes down from left to right. Then, since this may be a confounding factor, we can introduce a block per column for a total of 8 blocks.

The random allocation to treatments is restricted by the requirement that each horizontal plot have the same treatment, hence the randomization among blocks is not completely independent.

The random allocation is now performed by this script:

```
> ta = cbind(ceiling(sample(PLOTS/TREATMENTS,PLOTS/TREATMENTS)),
+          ceiling(sample(PLOTS/TREATMENTS,PLOTS/TREATMENTS)),
+          ceiling(sample(PLOTS/TREATMENTS,PLOTS/TREATMENTS)),
+          ceiling(sample(PLOTS/TREATMENTS,PLOTS/TREATMENTS))
+ )
> treatmentallocations = cbind(ta[,1],ta[,1],ta[,2],ta[,2],ta[,3],ta[,3],ta[,4],ta[
> treatmentallocations
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]    3    3    3    3    2    2    1    1
[2,]    2    2    4    4    1    1    4    4
[3,]    4    4    1    1    4    4    2    2
[4,]    1    1    2    2    3    3    3    3
> treatments = as.vector(t(treatmentallocations))
> blocks = rep(1:8,4)
>
> X = cbind(1,
+          blocks==1, blocks==2, blocks==3, blocks==4, blocks==5, blocks==6, blocks
+          treatments == 1, treatments == 2, treatments == 3, treatments == 4)
> X
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
 [1,]    1    1    0    0    0    0    0    0    0     0     0     1     0
 [2,]    1    0    1    0    0    0    0    0    0     0     0     1     0
 [3,]    1    0    0    1    0    0    0    0    0     0     0     1     0
 [4,]    1    0    0    0    1    0    0    0    0     0     0     1     0
 [5,]    1    0    0    0    0    1    0    0    0     0     1     0     0
 [6,]    1    0    0    0    0    0    1    0    0     0     1     0     0
 [7,]    1    0    0    0    0    0    0    1    0     1     0     0     0
 [8,]    1    0    0    0    0    0    0    0    1     1     0     0     0
 [9,]    1    1    0    0    0    0    0    0    0     0     1     0     0
[10,]    1    0    1    0    0    0    0    0    0     0     1     0     0
[11,]    1    0    0    1    0    0    0    0    0     0     0     0     1
[12,]    1    0    0    0    1    0    0    0    0     0     0     0     1
[13,]    1    0    0    0    0    1    0    0    0     1     0     0     0
[14,]    1    0    0    0    0    0    1    0    0     1     0     0     0
```

```
[15,]   1   0   0   0   0   0   0   1   0   0   0   0   1
[16,]   1   0   0   0   0   0   0   0   1   0   0   0   1
[17,]   1   1   0   0   0   0   0   0   0   0   0   0   1
[18,]   1   0   1   0   0   0   0   0   0   0   0   0   1
[19,]   1   0   0   1   0   0   0   0   0   1   0   0   0
[20,]   1   0   0   0   1   0   0   0   0   1   0   0   0
[21,]   1   0   0   0   0   1   0   0   0   0   0   0   1
[22,]   1   0   0   0   0   0   1   0   0   0   0   0   1
[23,]   1   0   0   0   0   0   0   1   0   0   1   0   0
[24,]   1   0   0   0   0   0   0   0   1   0   1   0   0
[25,]   1   1   0   0   0   0   0   0   0   1   0   0   0
[26,]   1   0   1   0   0   0   0   0   0   1   0   0   0
[27,]   1   0   0   1   0   0   0   0   0   0   1   0   0
[28,]   1   0   0   0   1   0   0   0   0   0   1   0   0
[29,]   1   0   0   0   0   1   0   0   0   0   0   1   0
[30,]   1   0   0   0   0   0   1   0   0   0   0   1   0
[31,]   1   0   0   0   0   0   0   1   0   0   0   1   0
[32,]   1   0   0   0   0   0   0   0   1   0   0   1   0
```

So the resulting model is

$$y = X\beta + \epsilon$$

where y_i contains the individual growth result of tree i, ϵ_i is the error term of the i'th tree, and

$$\beta = \begin{bmatrix} \mu \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_8 \\ \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix}$$

$$X = \begin{pmatrix}
1&1&0&0&0&0&0&0&0&0&0&1&0 \\
1&0&1&0&0&0&0&0&0&0&0&1&0 \\
1&0&0&1&0&0&0&0&0&0&0&1&0 \\
1&0&0&0&1&0&0&0&0&0&0&1&0 \\
1&0&0&0&0&1&0&0&0&0&1&0&0 \\
1&0&0&0&0&0&1&0&0&0&1&0&0 \\
1&0&0&0&0&0&0&1&0&1&0&0&0 \\
1&0&0&0&0&0&0&0&1&1&0&0&0 \\
1&1&0&0&0&0&0&0&0&0&1&0&0 \\
1&0&1&0&0&0&0&0&0&0&1&0&0 \\
1&0&0&1&0&0&0&0&0&0&0&0&1 \\
1&0&0&0&1&0&0&0&0&0&0&0&1 \\
1&0&0&0&0&1&0&0&0&1&0&0&0 \\
1&0&0&0&0&0&1&0&0&1&0&0&0 \\
1&0&0&0&0&0&0&1&0&0&0&0&1 \\
1&0&0&0&0&0&0&0&1&0&0&0&1 \\
1&1&0&0&0&0&0&0&0&0&0&0&1 \\
1&0&1&0&0&0&0&0&0&0&0&0&1 \\
1&0&0&1&0&0&0&0&0&1&0&0&0 \\
1&0&0&0&1&0&0&0&0&1&0&0&0 \\
1&0&0&0&0&1&0&0&0&0&0&0&1 \\
1&0&0&0&0&0&1&0&0&0&0&0&1 \\
1&0&0&0&0&0&0&1&0&0&1&0&0 \\
1&0&0&0&0&0&0&0&1&0&1&0&0 \\
1&1&0&0&0&0&0&0&0&1&0&0&0 \\
1&0&1&0&0&0&0&0&0&1&0&0&0 \\
1&0&0&1&0&0&0&0&0&0&1&0&0 \\
1&0&0&0&1&0&0&0&0&0&1&0&0 \\
1&0&0&0&0&1&0&0&0&0&0&1&0 \\
1&0&0&0&0&0&1&0&0&0&0&1&0 \\
1&0&0&0&0&0&0&1&0&0&0&1&0 \\
1&0&0&0&0&0&0&0&1&0&0&1&0
\end{pmatrix}$$