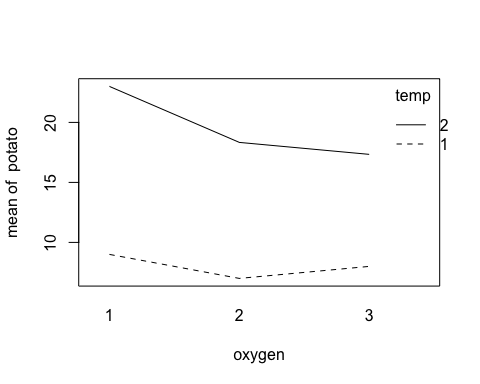
MAST30025 Assignment 3 2020

#Question 2) #Part a: Plot an interaction plot for the data. Does there appear to be interaction? #Attempt 1

potato = data.frame(potato = 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(1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2)))

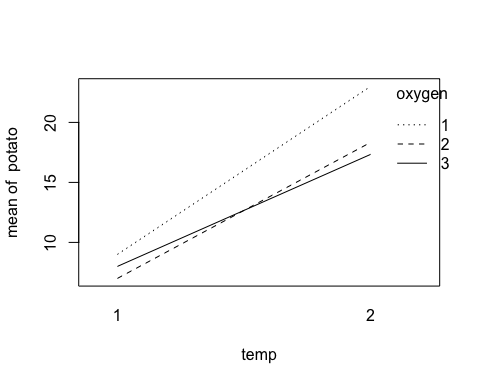
# 

with(potato, interaction.plot(oxygen,temp,potato))



## Actual solution

with(potato, interaction.plot(temp,oxygen,potato))



#DO NOT USE lm in the actual assignment

imodel = lm(potato~temp\*oxygen, data = potato)  
anova(imodel)

## Analysis of Variance Table  
##   
## Response: potato  
## Df Sum Sq Mean Sq F value Pr(>F)   
## temp 1 600.89 600.89 20.6412 0.000674 \*\*\*  
## oxygen 2 44.44 22.22 0.7634 0.487453   
## temp:oxygen 2 16.44 8.22 0.2824 0.758816   
## Residuals 12 349.33 29.11   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#There is clearly no interaction.

#Part b) Fit an additive model, outputting your design matrix. Estimate the common variance. #Attempt 1

library(MASS)  
library(Matrix)  
n = 18  
y = potato$potato  
X = matrix(0,n,6)  
X[,1] = 1  
X[cbind(1:n,as.numeric(potato$temp)+1)] = 1  
X[cbind(1:n,as.numeric(potato$oxygen)+3)] = 1  
X

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 1 1 0 1 0 0  
## [2,] 1 1 0 1 0 0  
## [3,] 1 1 0 1 0 0  
## [4,] 1 1 0 0 1 0  
## [5,] 1 1 0 0 1 0  
## [6,] 1 1 0 0 1 0  
## [7,] 1 1 0 0 0 1  
## [8,] 1 1 0 0 0 1  
## [9,] 1 1 0 0 0 1  
## [10,] 1 0 1 1 0 0  
## [11,] 1 0 1 1 0 0  
## [12,] 1 0 1 1 0 0  
## [13,] 1 0 1 0 1 0  
## [14,] 1 0 1 0 1 0  
## [15,] 1 0 1 0 1 0  
## [16,] 1 0 1 0 0 1  
## [17,] 1 0 1 0 0 1  
## [18,] 1 0 1 0 0 1

XtXc = ginv(t(X)%\*%X)  
b = XtXc%\*%t(X)%\*%y  
r = rankMatrix(X)  
s2 = sum((y - X%\*%b)^2)/(n-r)  
s2

## [1] 26.12698  
## attr(,"method")  
## [1] "tolNorm2"  
## attr(,"useGrad")  
## [1] FALSE  
## attr(,"tol")  
## [1] 3.996803e-15

#Part c) Calculate a 95% confidence interval for the difference between the temperature effects.

#Attempt 1: DO NOT RUN THIS CODE

#C = matrix(c(1,-1,-1,0,0,0),6,1)  
#mu12 = t(C)%\*%b  
#wdth = qt(.975, n-r)\*sqrt(s2\*t(t)%\*%XtXc%\*%t)  
#wdth  
#(mu12-wdth, mu12,mu12 + wdth)

#Actual solution

C = matrix(c(0,1,-1,0,0,0),6,1)  
mu12 = t(C)%\*%b  
wdth = qt(.975, n-r)\*sqrt(s2\*t(C)%\*%XtXc%\*%C)  
wdth

## [,1]  
## [1,] 5.168  
## attr(,"method")  
## [1] "tolNorm2"  
## attr(,"useGrad")  
## [1] FALSE  
## attr(,"tol")  
## [1] 3.996803e-15

c(mu12-wdth, mu12,mu12 + wdth)

## [1] -16.723555 -11.555556 -6.387556

#Part d) Test the hypothesis that oxygen level has no effect on rotting!

C = matrix(c(0,0,0,1,-1,0,0,0,0,0,1,-1),2,6)  
numer = t(C%\*%b)%\*%solve(C%\*%ginv(t(X)%\*%X)%\*%t(C))%\*%C%\*%b  
e = y - X%\*%b  
s2 = sum(e^2)/(n-r)  
Fstat = (numer/2)/s2  
Fstat

## [,1]  
## [1,] 12.45768  
## attr(,"method")  
## [1] "tolNorm2"  
## attr(,"useGrad")  
## [1] FALSE  
## attr(,"tol")  
## [1] 3.996803e-15

pf(Fstat,2,n-r,lower = F)

## [,1]  
## [1,] 0.0007799192  
## attr(,"method")  
## [1] "tolNorm2"  
## attr(,"useGrad")  
## [1] FALSE  
## attr(,"tol")  
## [1] 3.996803e-15

#Actual solution

XtXc = ginv(t(X)%\*%X)  
C = matrix(c(0,0,0,1,-1,0,0,0,0,0,1,-1),2,6,byrow=T)  
numer = t(C%\*%b)%\*%solve(C%\*%XtXc%\*%t(C))%\*%C%\*%b  
Fstat = (numer/2)/s2  
pf(Fstat,2,n-r,lower = F)

## [,1]  
## [1,] 0.4481124  
## attr(,"method")  
## [1] "tolNorm2"  
## attr(,"useGrad")  
## [1] FALSE  
## attr(,"tol")  
## [1] 3.996803e-15

#NOTE/TASK: Recompute your Design Matrix and compute again!

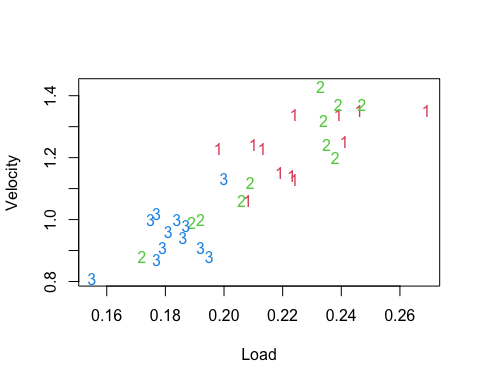
#Question 2e) Suppose we are interested in the effect of oxygen level only, but know that temperature affects the results, so we include it in our model. What type of design would this study be? #A complete complete design (What else?) Refer from slide 36 from Experimental Modulue!

#Question 4) #(a) Plot the data, using different colours and/or symbols for each tree. What do you observe?

setwd("~/Desktop/UNIMELB 2021 Material/UNIMELB S1 2021 (Currently)/MAST30025/Tutorials /Tutorials/Rfile/data")  
heli = read.csv("heli.csv")  
str(heli)

## 'data.frame': 35 obs. of 3 variables:  
## $ Tree : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Load : num 0.239 0.208 0.223 0.224 0.246 0.213 0.198 0.219 0.241 0.21 ...  
## $ Velocity: num 1.34 1.06 1.14 1.13 1.35 1.23 1.23 1.15 1.25 1.24 ...

#Actual solution  
plot(Velocity ~ Load, pch=as.character(Tree), col=Tree+1, data=heli)

 #Demonstrator comment: There appears to be a clear linear relationship between velocity and disk loading. It’s not clear if the tree has an effect, either directly or through interaction.

str(heli)

## 'data.frame': 35 obs. of 3 variables:  
## $ Tree : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Load : num 0.239 0.208 0.223 0.224 0.246 0.213 0.198 0.219 0.241 0.21 ...  
## $ Velocity: num 1.34 1.06 1.14 1.13 1.35 1.23 1.23 1.15 1.25 1.24 ...

#Without using the lm! #(b) Test for the presence of interaction between disk loading and tree. #Attempt 1

library(MASS)  
library(Matrix)  
y = heli$Velocity  
n = length(heli$Velocity)  
X = matrix(c(rep(1,n),rep(0,n\*7)),n,8)  
X[cbind(1:n,heli$Tree+1)]=1  
X[,5]=heli$Load  
X[cbind(1:n,heli$Tree+5)]=heli$Load  
r = rankMatrix(X)[1]  
XtXc = ginv(t(X)%\*%X)  
XtXc

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 2.6480861 0.3875622 -0.7483181 3.008842 -13.2164161 -0.1136429  
## [2,] 0.3875622 8.7193828 -2.2873302 -6.044490 -0.1136429 -39.8765339  
## [3,] -0.7483181 -2.2873302 6.4476221 -4.908610 4.5917511 8.7383078  
## [4,] 3.0088420 -6.0444903 -4.9086100 13.961942 -17.6945243 31.0245832  
## [5,] -13.2164161 -0.1136429 4.5917511 -17.694524 67.0244300 -8.0853337  
## [6,] -0.1136429 -39.8765339 8.7383078 31.024583 -8.0853337 184.9026225  
## [7,] 4.5917511 8.7383078 -30.4657460 26.319189 -27.3956436 -31.5434527  
## [8,] -17.6945243 31.0245832 26.3191892 -75.038297 102.5054073 -161.4445036  
## [,7] [,8]  
## [1,] 4.591751 -17.69452  
## [2,] 8.738308 31.02458  
## [3,] -30.465746 26.31919  
## [4,] 26.319189 -75.03830  
## [5,] -27.395644 102.50541  
## [6,] -31.543453 -161.44450  
## [7,] 146.282003 -142.13419  
## [8,] -142.134194 406.08410

b = XtXc%\*%t(X)%\*%y  
XtXc

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 2.6480861 0.3875622 -0.7483181 3.008842 -13.2164161 -0.1136429  
## [2,] 0.3875622 8.7193828 -2.2873302 -6.044490 -0.1136429 -39.8765339  
## [3,] -0.7483181 -2.2873302 6.4476221 -4.908610 4.5917511 8.7383078  
## [4,] 3.0088420 -6.0444903 -4.9086100 13.961942 -17.6945243 31.0245832  
## [5,] -13.2164161 -0.1136429 4.5917511 -17.694524 67.0244300 -8.0853337  
## [6,] -0.1136429 -39.8765339 8.7383078 31.024583 -8.0853337 184.9026225  
## [7,] 4.5917511 8.7383078 -30.4657460 26.319189 -27.3956436 -31.5434527  
## [8,] -17.6945243 31.0245832 26.3191892 -75.038297 102.5054073 -161.4445036  
## [,7] [,8]  
## [1,] 4.591751 -17.69452  
## [2,] 8.738308 31.02458  
## [3,] -30.465746 26.31919  
## [4,] 26.319189 -75.03830  
## [5,] -27.395644 102.50541  
## [6,] -31.543453 -161.44450  
## [7,] 146.282003 -142.13419  
## [8,] -142.134194 406.08410

s2 = sum((y-X%\*%b)^2)/(n-r)  
C = matrix(c(0,0,0,0,0,1,-1,0,0,0,0,0,0,1,0,-1),2,8,byrow = T)  
Fstat = t(b)%\*%t(C)%\*%solve(C%\*%XtXc%\*%t(C))%\*%C%\*%b/2/s2  
pf(Fstat,2,n-r,lower=F)

## [,1]  
## [1,] 0.05010679

#The interaction is not signifiancant, a little above at 5%!

#(c) Use backward elimination from the model with interaction to select variables for the data.

#Attempt 1

fullmodel = lm(y~Load + factor(Tree)+ Load\*factor(Tree),data = heli)  
drop1(fullmodel, scope = ~., test = "F")

## Single term deletions  
##   
## Model:  
## y ~ Load + factor(Tree) + Load \* factor(Tree)  
## 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 \*  
## factor(Tree) 2 0.037553 0.20305 -172.24 3.2903 0.05154 .  
## Load:factor(Tree) 2 0.037949 0.20344 -172.17 3.3250 0.05011 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

model2 = lm(y~Load + factor(Tree),data = heli)  
drop1(model2, scope = ~., test = "F")

## Single term deletions  
##   
## Model:  
## y ~ Load + factor(Tree)  
## Df Sum of Sq RSS AIC F value Pr(>F)   
## <none> 0.20344 -172.17   
## Load 1 0.315542 0.51898 -141.39 48.0817 8.884e-08 \*\*\*  
## factor(Tree) 2 0.011322 0.21476 -174.28 0.8626 0.4319   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

model3 = lm(y~Load,data = heli)  
drop1(model3, scope = ~., test = "F")

## Single term deletions  
##   
## Model:  
## y ~ Load  
## Df Sum of Sq RSS AIC F value Pr(>F)   
## <none> 0.21476 -174.28   
## Load 1 0.84364 1.05840 -120.45 129.63 5.704e-13 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

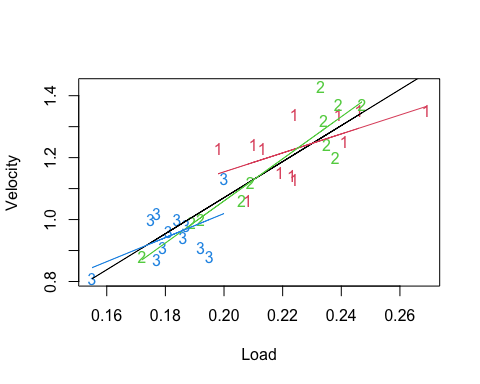
#We include Load in our final model! Despite the Tree is dropped during the elimation which also mean the interaction term has droppped!

#Part d: Add lines corresponding to model from part (c), and the full model with interaction, to the plot from question (a).

setwd("~/Desktop/UNIMELB 2021 Material/UNIMELB S1 2021 (Currently)/MAST30025/Tutorials /Tutorials/Rfile/data")  
heli = read.csv("heli.csv")  
str(heli)

## 'data.frame': 35 obs. of 3 variables:  
## $ Tree : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Load : num 0.239 0.208 0.223 0.224 0.246 0.213 0.198 0.219 0.241 0.21 ...  
## $ Velocity: num 1.34 1.06 1.14 1.13 1.35 1.23 1.23 1.15 1.25 1.24 ...

plot(Velocity ~ Load, pch=as.character(Tree), col=Tree+1, data=heli)  
#Actual solution  
with(heli, lines(Load, fitted(model3)))  
for (i in 1:3) {with(heli, lines(Load[Tree==i], fitted(fullmodel)[Tree==i], col=i+1))}



#Part e: In the full model with interaction, test the hypothesis that a samara from tree 2 with a disk loading of 0.2 has an average descent velocity of 1.

#My Method  
library(car)

## Loading required package: carData

heli$Tree = factor(heli$Tree)  
imodel = lm(Velocity~Tree\*Load, data = heli)  
linearHypothesis(imodel, c(1,1,0,0.2,0.2,0),1)

## Linear hypothesis test  
##   
## Hypothesis:  
## (Intercept) + Tree2 + 0.2 Load + 0.2 Tree2:Load = 1  
##   
## Model 1: restricted model  
## Model 2: Velocity ~ Tree \* Load  
##   
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 30 0.19127   
## 2 29 0.16549 1 0.025781 4.5177 0.04219 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

#Question 5 (JUST TRY TO UNDERSTAND IT) #Part a

#Attempt 1  
#We use a completely randomised design. The plots can be oriented horizontally or vertically — we choose horizontally. Here is one possible allocation:  
alloc = sample(16,16) #plot allocation, ( a rondom allocation.)  
alloc

## [1] 4 2 11 9 10 6 8 16 15 14 5 7 12 1 13 3

talloc = as.vector(rbind(2\*alloc-1,2\*alloc))  
talloc

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

talloc2 = c()  
for (i in 1:4){talloc2[talloc[(8\*i-7):(8\*i)]] = i}  
matrix(talloc2,4,8,byrow = T) #tree allocation

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]  
## [1,] 4 4 1 1 4 4 1 1  
## [2,] 3 3 2 2 3 3 2 2  
## [3,] 1 1 2 2 1 1 4 4  
## [4,] 4 4 3 3 3 3 2 2

X = matrix(0,32,5)  
X[,1] = 1  
X[cbind(1:32,talloc2+1)] = 1  
X

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1 0 0 0 1  
## [2,] 1 0 0 0 1  
## [3,] 1 1 0 0 0  
## [4,] 1 1 0 0 0  
## [5,] 1 0 0 0 1  
## [6,] 1 0 0 0 1  
## [7,] 1 1 0 0 0  
## [8,] 1 1 0 0 0  
## [9,] 1 0 0 1 0  
## [10,] 1 0 0 1 0  
## [11,] 1 0 1 0 0  
## [12,] 1 0 1 0 0  
## [13,] 1 0 0 1 0  
## [14,] 1 0 0 1 0  
## [15,] 1 0 1 0 0  
## [16,] 1 0 1 0 0  
## [17,] 1 1 0 0 0  
## [18,] 1 1 0 0 0  
## [19,] 1 0 1 0 0  
## [20,] 1 0 1 0 0  
## [21,] 1 1 0 0 0  
## [22,] 1 1 0 0 0  
## [23,] 1 0 0 0 1  
## [24,] 1 0 0 0 1  
## [25,] 1 0 0 0 1  
## [26,] 1 0 0 0 1  
## [27,] 1 0 0 1 0  
## [28,] 1 0 0 1 0  
## [29,] 1 0 0 1 0  
## [30,] 1 0 0 1 0  
## [31,] 1 0 1 0 0  
## [32,] 1 0 1 0 0

#Part b

alloc = c()  
for (i in 1:4) { alloc = c(alloc,sample(4,4)) }  
malloc = matrix(alloc,4,4) #plot allocation  
 #tree allocation

talloc = rep(as.vector(t(malloc)),each=2)  
matrix(talloc, 4,8, byrow=T) #tree allocation

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]  
## [1,] 4 4 3 3 3 3 2 2  
## [2,] 2 2 1 1 2 2 3 3  
## [3,] 3 3 4 4 4 4 1 1  
## [4,] 1 1 2 2 1 1 4 4

X = matrix(0,32,9)  
X[,1] = 1  
X[cbind(1:32,rep(1:4,4,each=2)+1)] = 1   
X[cbind(1:32,talloc+5)] = 1  
X

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]  
## [1,] 1 1 0 0 0 0 0 0 1  
## [2,] 1 1 0 0 0 0 0 0 1  
## [3,] 1 0 1 0 0 0 0 1 0  
## [4,] 1 0 1 0 0 0 0 1 0  
## [5,] 1 0 0 1 0 0 0 1 0  
## [6,] 1 0 0 1 0 0 0 1 0  
## [7,] 1 0 0 0 1 0 1 0 0  
## [8,] 1 0 0 0 1 0 1 0 0  
## [9,] 1 1 0 0 0 0 1 0 0  
## [10,] 1 1 0 0 0 0 1 0 0  
## [11,] 1 0 1 0 0 1 0 0 0  
## [12,] 1 0 1 0 0 1 0 0 0  
## [13,] 1 0 0 1 0 0 1 0 0  
## [14,] 1 0 0 1 0 0 1 0 0  
## [15,] 1 0 0 0 1 0 0 1 0  
## [16,] 1 0 0 0 1 0 0 1 0  
## [17,] 1 1 0 0 0 0 0 1 0  
## [18,] 1 1 0 0 0 0 0 1 0  
## [19,] 1 0 1 0 0 0 0 0 1  
## [20,] 1 0 1 0 0 0 0 0 1  
## [21,] 1 0 0 1 0 0 0 0 1  
## [22,] 1 0 0 1 0 0 0 0 1  
## [23,] 1 0 0 0 1 1 0 0 0  
## [24,] 1 0 0 0 1 1 0 0 0  
## [25,] 1 1 0 0 0 1 0 0 0  
## [26,] 1 1 0 0 0 1 0 0 0  
## [27,] 1 0 1 0 0 0 1 0 0  
## [28,] 1 0 1 0 0 0 1 0 0  
## [29,] 1 0 0 1 0 1 0 0 0  
## [30,] 1 0 0 1 0 1 0 0 0  
## [31,] 1 0 0 0 1 0 0 0 1  
## [32,] 1 0 0 0 1 0 0 0 1

#An alternate formulation would be to have 8 block effects.