Math 328 Chapter 6 HW Part A

David Oniani

March 02, 2021

## Setup

library(Stat2Data)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library (tidyr)  
library(emmeans)  
library(ggplot2)  
  
# I like the minimal theme  
theme\_set(theme\_minimal())  
  
# Disable warnings (they clutter the document)  
options(warn = -1)

## Exercise 6.10

1. 3 - 1 = 2. Hence, we have 2 degrees of freedom for factor A.
2. 3 - 1 = 2. Hence, we have 2 degrees of freedom for factor B.
3. (3 - 1) \* (3 - 1) = 2 \* 2 = 4. Hence, we have 4 degrees of freedom for Error.

## Exercise 6.26

1. This is an observational study as the study did not affect or control the subjects in any way.
   1. Factors of interest: word lists (note that the study suggested that within each word list, words are of equal hearing difficulty, but we do not know if this is the case across the word lists/groups).
   2. Nuisance factors: volume and background noise
2. Subjects with normal hearing.
3. We know that the variability within blocks is less than the variability between blocks (since within blocks/word lists, words are of equal hearing difficulty). Hence, this is a randomized block design with 4 levels. Since there are 24 subjects in the study, the block size is 24.

NOTE: I just recreated the dataset presented in the book (the problem did not state to use it however).

data(HearingTest)  
  
# Data processing  
df1 = HearingTest  
  
l1 = df1[df1$List == "L1",]$Percent  
l2 = df1[df1$List == "L2",]$Percent  
l3 = df1[df1$List == "L3",]$Percent  
l4 = df1[df1$List == "L4",]$Percent  
  
df2 = data.frame(c(1:24), l1, l2, l3, l4)  
df2$Mean = rowMeans(df2[,-1])  
colnames(df2) = c("Sub", "L1", "L2", "L3", "L4", "Mean")  
  
mean\_l1 = round(mean(df2$L1))  
mean\_l2 = round(mean(df2$L2))  
mean\_l3 = round(mean(df2$L3))  
mean\_l4 = round(mean(df2$L4))  
mean\_mean = round(mean(df2$Mean), 1)  
  
tmp = data.frame(nrow(df2) + 1, mean\_l1, mean\_l2, mean\_l3, mean\_l4, mean\_mean)  
colnames(tmp) = c("Sub", "L1", "L2", "L3", "L4", "Mean")  
df3 = rbind(df2, tmp)  
rownames(df3) = c(c(1:24), "Mean")

## Exercise 6.34

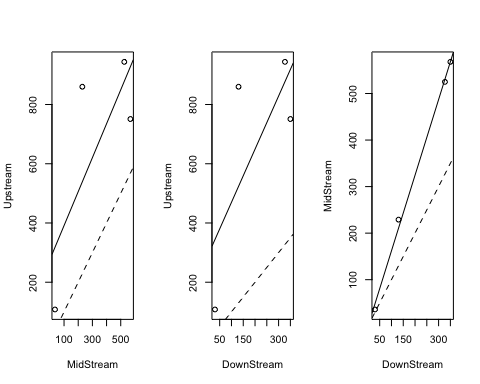
data("RiverIron")  
  
Iron = pivot\_wider (RiverIron,  
 id\_cols = "River",  
 names\_from = "Site",  
 values\_from = "Iron")  
  
anscombe.plot = function (yvar, xvar) {  
 yname = deparse (substitute (yvar))  
 xname = deparse (substitute (xvar))  
 plot (yvar ~ xvar, ylab=yname, xlab=xname)  
 apfit = lm (yvar ~ xvar)  
 apfit  
 abline (apfit)  
 abline (0, 1, lty=2)  
 return (apfit)  
}  
  
par (mfrow=c(1,3))  
with (Iron, anscombe.plot (Upstream, MidStream))

##   
## Call:  
## lm(formula = yvar ~ xvar)  
##   
## Coefficients:  
## (Intercept) xvar   
## 276.533 1.146

with (Iron, anscombe.plot (Upstream, DownStream))

##   
## Call:  
## lm(formula = yvar ~ xvar)  
##   
## Coefficients:  
## (Intercept) xvar   
## 290.293 1.794

with (Iron, anscombe.plot (MidStream, DownStream))



##   
## Call:  
## lm(formula = yvar ~ xvar)  
##   
## Coefficients:  
## (Intercept) xvar   
## -0.5967 1.6253

1. Above find the plots.
2. The slopes are 1.146 (for UpStream vs MidStream), 1.146 (for UpStream vs DownStream), and 1.6253 (for MidStream vs DownStream). In general, slopes in the Anscombe plot alert to the possibility that a model with additive main effects may offer a bad fit to the data, or at least that one might need to reexpress the response to get a good fit. In this case, we got that all three slopes are “close” to 1 which allows for the application of the randomized complete block additive model (RCB additive model).