Stats Homework 2

ANOVA

As an SE researcher you are evaluating different programming languages. For the next set of questions input the R code and interpret your findings.

a) The results of your first study compares Java, Python, and Ruby code based on the size of the programs in source (i.e. non-blank, non-commented) lines of code. Perform an ANOVA to determine whether there is an effect on size due to programming language. Use `lang-size.csv`.

```
# code goes here.
lang.size <- read.csv("lang-size.csv")
summary(lang.size)
```

```
##      lang                sloc
##  Length:90          Min.   :176.0
##  Class :character   1st Qu.:259.8
##  Mode  :character   Median :306.0
##                     Mean   :308.8
##                     3rd Qu.:366.0
##                     Max.   :455.0
```

```
#lang.size
size <- aov(sloc ~ lang, data = lang.size)
summary(size)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## lang         2 276056  138028    62.6 <2e-16 ***
## Residuals   87 191836    2205
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
#TukeyHSD(size)
```

Report:

1) The degrees of freedom (under column labelled Df) for the variable lang. This is calculated as (# of groups) - 1, so in this case, there were 3 languages(Java, Python, RUuby) and the value is 3-1 = 2.
2) The degrees of freedom for the residuals. This is calculated as (# of total observations) - (# of groups). In this case, there were 90 observations and 3 groups, so this value is 90-3 = 87.
3) The sum of squares (under column labelled Sum Sq) for the variable lang. The sum of squares helps express the total variation that can be attributed to various factors; i.e. sum of squares = treatment sum of squares (SST) + sum of squares of the residual error (SSE). In this case, the value is

276056.

4) The sum of squares of the residual. This value is 191836.

5) The mean square (under column labelled Mean Sq) for the variable lang. This is calculated as (sum of squares of treatment) / (Df of treatment), and allows you to determine whether there is a significant difference due to the treatment. The larger the ratio is, the more the treatments affect the outcome. In this case, it is calculated as 276056/2 = 138028.
6) The mean square of the residuals. This is calculated as (sum of squares of residuals) / (Df of residuals). In this case, it is calculated as 191836/87 = 2205.
7) The overall F-statistic of the ANOVA model (under column labelled F value). This is calculated as (mean square of treatment) / (mean square of residuals). In this case, it is calculated as 138028/2205 = 62.6.
8) The p-value (under column labelled Pr(>F)) associated with the F-statistic with numerator df = 2 and denominator df = 87. In this case, the p-value is <2e-16, which is <0.000000000000000199 and well below either a p<0.05, p<0.01, or even p<0.001 threshold for significance. The *** stars beside this value also indicate where this fits on a significance range from 0 to 1.

- Interpreting the results for our specific test, we see that the p-value in our ANOVA table (<2e-16) is less than p<0.05 and we therefore find sufficient evidence to reject the null hypothesis that there is no effect on size due to programming language. This means that we have sufficient evidence to say that there is an effect on size due to programming language.

- After Post-Hoc test, the pairwise comparisons show that Python has a significantly higher mean than both Java and Ruby, but the difference between the mean of Java and Ruby is not statistically significant.

b) In a subsequent study you measured the programming time (in hours) required to solve a program in Java, Python, and Ruby. This was a within subject study design: each participant solved the problem three times, and all participants solved the problem in the same order (Java, then Python, then Ruby). Perform an ANOVA to determine whether there is an effect due to programming language. Use `lang-time.csv`.

```
# code goes here
lang.time <- read.csv("lang-time.csv")
#(lang.time)
summary(lang.time)
```

```
##      lang           participant            times       
##  Length:72          Length:72          Min.   : 3.100  
##  Class :character   Class :character   1st Qu.: 6.200  
##  Mode  :character   Mode  :character   Median : 7.950  
##                                        Mean   : 7.832  
##                                        3rd Qu.: 9.325  
##                                        Max.   :12.600  
```

```
time <- aov(times ~ lang + participant, data = lang.time)
#time <- aov(times ~ lang + participant + lang:participant, data = lang.time)
summary(time)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)  
## lang         2  33.32  16.661   4.583 0.0153 *
## participant 23 110.62   4.809   1.323 0.2061
## Residuals   46 167.24   3.636
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
#TukeyHSD(time)
```

Report:

- Because this is a two-way ANOVA, the ANOVA table provides results broken out by group (i.e. the independent variables). In this case, we can see that only the Languages factor has a statistically significant effect on the mean number of moths caught. This result leads us to believe that changing the languages will impact significantly the mean of times; and that changing participant would not have such an effect.
- Results: Interpreting the results for our specific test, we see that the p-value in our ANOVA table (0.0153) is less than $p<0.05$ and we therefore find sufficient evidence to reject the null hypothesis that there is no effect due to programming language when each participant solved the problem three times, and all participants solved the problem in the same order (Java, then Python, then Ruby). This means that we have sufficient evidence to say that there is an effect due to programming language.

c) Your realized you should have counterbalanced, so you replicated the study from (b) which uses a crossover design to control for ordering. Each participant solved the problem in all three languages, but in each participant solved them in a different order. Perform an ANOVA to determine whether there is an effect due to programming language. Use `lang-time-crossover.csv`.

```
# code goes here
lang.time.crossover <- read.csv("lang-time-crossover.csv")
summary(lang.time.crossover)
```

```
##  participant         treatment             lang               times       
##  Length:18          Length:18          Length:18          Min.   : 4.000  
##  Class :character   Class :character   Class :character   1st Qu.: 6.550  
##  Mode  :character   Mode  :character   Mode  :character   Median : 7.800  
##                                                           Mean   : 7.500  
##                                                           3rd Qu.: 8.275  
##                                                           Max.   :10.600  
```

```
crossover <- aov(times ~ lang + treatment + participant, data = lang.time.crossover)
summary(crossover)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## lang         2  2.170   1.085   0.377  0.698
## treatment    2  5.623   2.812   0.976  0.418
## participant  5 26.100   5.220   1.812  0.217
## Residuals    8 23.047   2.881
```

Report:

- After counter-balancing, when we performed the ANOVA test, from the p-values we can infer that programming languages($p = 0.698 > 0.05$), treatment ($p = 0.418 > 0.05$) and participant ($p = 0.217 > 0.05$) are not statistically significant. Hence, we can reject null hypothesis that there is effect of programming languages after the counterbalancing. This means that we have sufficient evidence to say that there is no effect of programming languages after the counterbalancing.

d) You have some simulated results from an experiment that compared development time for Java, Python and Ruby, for subjects with low experience and high experience. Perform an ANOVA and identify which factors (language, experience) had a statistically significant effect. Also specify whether the interaction between programming languages and experience was statistically significant or not. Use `lang-time-exp.csv`.

```
# code goes here
lang.time.exp <- read.csv("lang-time-exp.csv")
summary(lang.time.exp)
```

```
##      lang               exp                times       
##  Length:120         Length:120         Min.   : 1.700  
##  Class :character   Class :character   1st Qu.: 5.700  
##  Mode  :character   Mode  :character   Median : 8.100  
##                                        Mean   : 7.848  
##                                        3rd Qu.:10.325  
##                                        Max.   :14.300  
```

```
exp <- aov(times ~ lang + exp + lang:exp, data = lang.time.exp)
#pan <- aov(times ~ lang * exp, data = lang.time.exp) either is same
summary(exp)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)    
## lang          2   11.1     5.6   1.730  0.182    
## exp           1  663.2   663.2 206.137 <2e-16 ***
## lang:exp      2    9.7     4.8   1.502  0.227    
## Residuals   114  366.8     3.2                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Report:

- The p-value in the output of the ANOVA table shows that programming languages($p = 0.182 > 0.05$) has no effect as it was not statistically significant. However, the experience($p < 2e{-16} < 0.05$) is having effect because it is statistically significant. In addition to this, the interaction between programming language and experience ($p = 0.227 > 0.05$) is not statistically significant because of its higher p value.

Part 3: Data analysis of an experiment

In this question, you'll analyze the raw data from an experiment and write up the results (similar to a publication).

The data is from a experiment to test whether statically typed languages (e.g. Java) or dynamically typed languages (e.g. Python) require more programming effort. The study evaluates the languages on two problems, a "small" problem and a "large" problem, to see if the results change based on the size of the problem. The study is a factorial design. The raw data from the experiment is available in this file: `lang-time-size.csv`.

Analyze the data and write up a short "results" section (as if it were a part of a paper) with your analysis of the data. This section should contain: * Analysis of variance tables to determine if there are any interactions * Interaction plot between the 2 factors * Effect sizes for programming language for the "small" problem and for the "large" problem. * I am not looking for a specific format, use your judgement about the best way to present this data to convey the results to a reader.

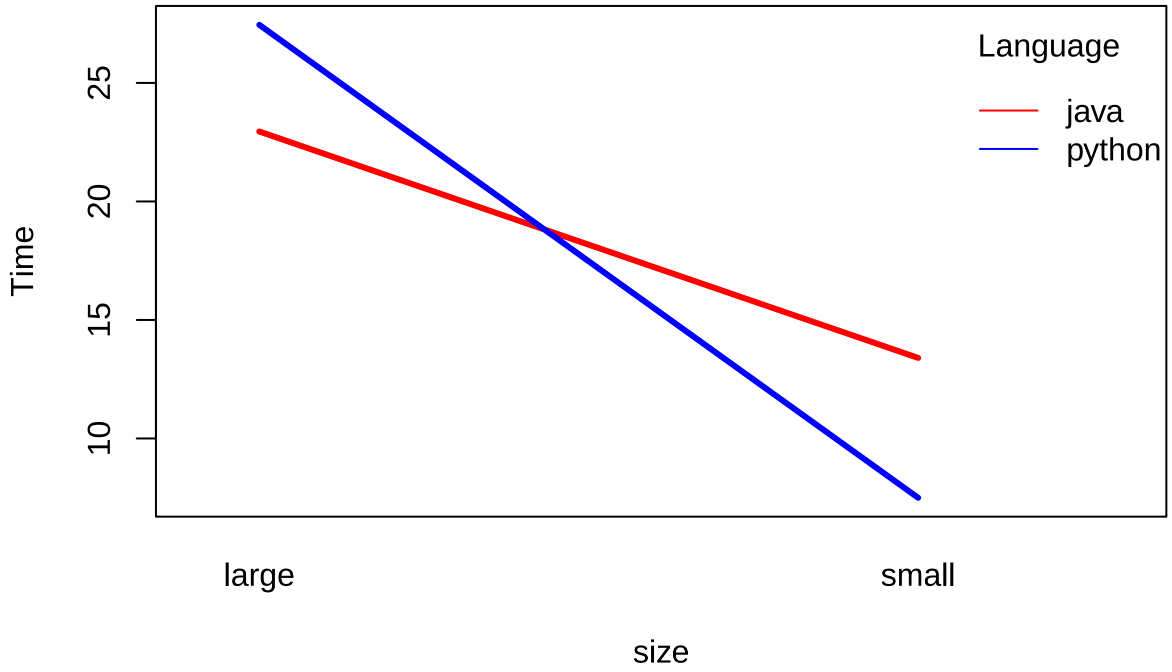
```
# Code for analysis goes here.
lang.time.size <- read.csv("lang-time-size.csv")
summary(lang.time.size)
```

```
##      times           lang               size          
##  Min.   : 1.50   Length:120         Length:120        
##  1st Qu.:10.15   Class :character   Class :character  
##  Median :16.35   Mode  :character   Mode  :character  
##  Mean   :17.69                                        
##  3rd Qu.:25.05                                        
##  Max.   :43.80
```

```
interactions <- aov(times ~ lang + size + lang:size,  data = lang.time.size)
#int <- aov(times ~ lang*size, data = lang.time.size) either is same
summary(interactions)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## lang          1      3       3   0.085    0.772    
## size          1   5410    5410 133.246  < 2e-16 ***
## lang:size     1    811     811  19.968 1.84e-05 ***
## Residuals   116   4709      41                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
interaction.plot(x.factor = lang.time.size$size, #x-axis variable
                 trace.factor = lang.time.size$lang, #variable for lines
                 response = lang.time.size$times, #y-axis variable
                 fun = median, #metric to plot
                 ylab = "Time",
                 xlab = "size",
                 col = c("red", "blue"),
                 lty = 1, #line type
                 lwd = 3, #line width
                 trace.label = "Language")
```



```
TukeyHSD(interactions)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = times ~ lang + size + lang:size, data = lang.time.size)
##
## $lang
##                   diff       lwr      upr     p adj
## python-java -0.3383333 -2.642417 1.965751 0.7716957
##
## $size
##                  diff       lwr       upr p adj
## small-large -13.42833 -15.73242 -11.12425     0
##
## $‘lang:size‘
##                                 diff         lwr        upr     p adj
## python:large-java:large     4.860000   0.5715926   9.148407 0.0195978
## java:small-java:large      -8.230000 -12.5184074  -3.941593 0.0000120
## python:small-java:large   -13.766667 -18.0550741  -9.478259 0.0000000
## java:small-python:large   -13.090000 -17.3784074  -8.801593 0.0000000
## python:small-python:large -18.626667 -22.9150741 -14.338259 0.0000000
## python:small-java:small    -5.536667  -9.8250741  -1.248259 0.0056466
```

```
#TukeyHSD(interactions, which = "lang")
#TukeyHSD(interactions, which = "size")

#boxplot(times ~ lang * size, data=lang.time.size) --> this not relevant to
#running and interpreting ANOVA tests
```

Report:

- The p-value in the output of the ANOVA table shows that programming languages ($0.772 > 0.05$) has no effect as it was not statistically significant. However, size($p <$ 2e-16 <0.05) does have effect because it is statistically significant. Also, the programming languages and size have interaction between themselves which can be inferred from its p value which is 1.84e-05 < 0.05 and is statistically significant.
- In general, if the two lines on the interaction plot are parallel then there is no interaction effect. However, if the lines intersect then there is likely an interaction effect. We can see in this plot that the lines for Python and Java do intersect, which indicates that there is likely an interaction effect between the variables of size and programming language. This matches the fact that the p-value in the output of the ANOVA table was statistically significant for the interaction term in the ANOVA model.
- The effect sizes for programming language for the "small" problem and for the "large" problem is clearly shown using the Post-Hoc method. We can interpret that all the adjusted p values for python-java with large and small are statistically. If we scan down the p-values in the p adj column, we can quickly see that the only adjusted p-values that are less than 0.05 python and java pairing along with large and small. Therefore, we can conclude that there is a significant difference in mean percent of small and large problems on python and java. Therefore, we can say that there is effect sizes for programming language for small problem and large problem. Also, we can notice that, when the programming languages were given with both large or both small there is a higher adjusted p-value for it when compared to that of combination of large and small problems. This means that there is a smaller significant difference when the problems are similar.