Stat 462: Lab 1

For this lab, do the following exercises from Data Analysis and Graphics Using R (DAAG): 1.2, 1.3, 1.10, 1.11, 1.13, 1.16, 1.19. Students in Stat 862 should also do 1.21. For this week, I will include the problems so that you can get started in case you don't have the textbook yet. If you don't yet have the text, usingR.pdf (see the course syllabus) provides a very similar introduction.

For all questions, provide the code you used as well as any output asked for and clear, thorough explanations where necessary. Do not provide output that was not requested in the question.

1.2) The `orings` data frame gives data on the damage that had occurred in US space shuttle launches prior to the disastrous Challenger launch of 28 January 1986. The observations in rows 1, 2, 4, 11, 13, and 18 were included in the pre-launch charts used in deciding whether to proceed with the launch, while the remaining rows were omitted.

Create a new data frame by extracting these rows from `orings`, and plot `total` incidents against `temperature` for this new data frame. Obtain a similar plot for the full data set.

1.3) For the data frame `possum` (DAAG package)

(a) Use the function `str()` to get information on each of the columns

(b) Using the function `complete.cases()`, determine the rows in which one or more values is missing. Print those rows. In which columns do the missing values appear?

1.10) The ^ symbol denotes exponentiation. Consider the following:

```
1000*((1+0.075)^5 - 1)    # Interest on $1000, compounded
                          # annually at 7.5% p.a. for five
                          # years
```

(a) Evaluate the above expression.
(b) Modify the expression to determine the amount of interest paid if the rate is 3.5% p.a.
(c) Explain the result obtained when the exponent 5 is changed to `seq(1,10)`

1.11) Run the following code:

```
gender <- factor( c( rep("female", 91), rep("male", 92) ) )
table( gender )
gender <- factor( gender, levels = c( "male", "female" ) )
table( gender )
gender <- factor( gender, levels = c("Male", "female" ) )
                # Note the mistake: "Male" should be "male"
table( gender )
table( gender, exclude = NULL )
```

```
     rm( gender )
```

Explain the output from the successive uses of `table()`.

1.13) The following plots four different transformations of the `Animals` data from the MASS package. What different aspects of the data do these different graphs emphasize? Consider the effect on low values of the variables, as contrasted with the effect on high values.

```
     par( mfrow=c(2,2) ) # 2 by 2 layout on the page
     library( MASS )     # Animals is in the MASS package
     plot( brain ~ body, data = Animals )
     plot( sqrt( brain ) ~ sqrt( body ), data = Animals )
     plot( I( brain^0.1 ) ~ I( body^0.1 ), data = Animals )
          # I() forces its argument to be treated “as is”
     plot( log( brain ) ~ log( body ), data = Animals )
     par( mfrow=c(1,1) ) # Restore to 1 figure per page
```

1.16) Functions that can be useful for labeling points on graphs are `abbreviate()` (create abbreviated names) and `paste()` (create composite labels). A composite label might, for the data from `socsupport`, give information about `gender`, `country`, and row number. Try the following:

```
     gender1 <- with( socsupport, abbreviate( gender, 1 ) )
     table( gender1 )    # Examine the result
     country3 <- with( socsupport, abbreviate( country, 3 ) )
     table( country3 )   # Examine the result
```

Now use the following to create a label that can be used with `text()` or with `identify()`:

```
     num <- with( socsupport, seq( along=gender )
                         # Generate row numbers
     lab <- paste( gender1, country3, num, sep = “:” )
```

Use `identify()` to place labels on all the points that the boxplots have identified as “outliers”.

Note: You can create the boxplot using `plot( BDI ~ age, data = socsupport )`

1.19) The data frame `vlt` (DAAG) consists of observations taken on a video lottery terminal during a two-day period. Eight different objects can appear in each of the three windows. Here, they are coded from 0 to 7. Different combinations of the objects give prizes (although with small probability). The first four rows are:

```
> head( vlt, 4 )
  window1 window2 window3 prize night
1       2       0       0     0     1
2       0       5       1     0     1
3       0       0       0     0     1
4       2       0       0     0     1
```

Use `stack()` to convert the first three columns of this data set to a case-by-variable format, then creating a tabular summary of the results, broken down by window.

```
vltcv <- stack( vlt[, 1:3] )
head( vltcv )       # first few rows of vltcv
table( vltcv$values, vltcv$ind )
     # More cryptically, table( vltcv ) gives the same result.
```

Does any window stand out as different?

For Stat 862 students (or ambitious 462 students):

1.21) The following uses the `for()` looping function to plot graphs that compare the relative population growth (here, by the use of a logarithmic scale) for the Australian states and territories.

```
oldpar <- par( mfrow=c(2,4) )
for( i in 2:9 ){
     plot( austpop[,1], log( austpop[,i] ), xlab = "Year",
          ylab = names(austpop)[i], pch=16, ylim=c(0,10) )
}
par( oldpar )
```

Find a way to do this without explicit looping. Hint: Use the function `sapply()`, with `austpop[,2:9]` as the first argument.

Note: The hint in the book is a little off… Use the function `sapply()`, with `2:9` as the first argument.