STAT40730
Data Programming with R (Online).
Lab 10: Debugging

1. Change the `findwords` function of lecture 3 so that the function reports an error if there is either one word or if it is given an object which isn't a character string. Test your new function with the following three arguments and state whether the function produced an answer.

```
findwords(1:10)
findwords('Thequick brown foxjumpedoverthe lazy dog')
findwords('Ilovedataprogrammingwithr')
```

2. The following function is designed to read in a text string (already in a vector), find the frequency of each letter used, the length of each word, and to produce a bar chart of the results over two panels. Unfortunately it contains two mistakes. In the example paragraph below, the code seems to report that v was used twice, when I can clearly see from the text that it was used 3 times (in vile, victory and prevent). Similarly, it seems to be reporting a bizarre mean word length. Use debugging methods to fix it and report the correct average word length to 2 decimal places.

```
bookdetails <- function(wrds) {
  # Convert words to lower case
  wrds2 <- tolower(wrds)
  # Find word lengths
  wrdlengths <- nchar(wrds2)
  # Set a vector to contain letter frequencies
  letterfreqs <- vector(length = 26)
  # Give it some names
  names(letterfreqs) <- letters
  for(i in 1:26) {
    # Loop through letters to find frequencies
    letterfreqs[i] <- length(grep(letters[i], wrds))
  }

  # Create 2 by 1 plot window
  par(mfrow = c(2,1))
  barplot(table(wrdlengths),main = 'Word Lengths')
  barplot(letterfreqs, main = 'Word frequencies')
  cat('Average word length is', round(mean(letterfreqs), 2),
    'and most popular letter is', letters[which.max(letterfreqs)], '\n')
  # Re-set the plot window
```

```
  par(mfrow = c(1, 1))
}

firstpar <- c("It", "was", "a", "bright", "cold", "day", "in",
  "April", "and", "the", "clocks", "were", "striking", "thirteen",
  "Winston", "Smith", "his", "chin", "nuzzled", "into", "his",
  "breast", "in", "an", "effort", "to", "escape", "the", "vile",
  "wind", "slipped", "quickly", "through", "the", "glass", "doors",
  "of", "Victory", "Mansions", "though", "not", "quickly", "enough",
  "to", "prevent", "a", "swirl", "of", "gritty", "dust", "from",
  "entering", "along", "with", "him")
bookdetails(firstpar)
```

3. The website `www.textfiles.com/etext/AUTHORS` contains text files of books by different authors. The function below will download the text file and turn it into a character vector. Use the fixed version of the function from question 2 to compare some different books. Which has the longest word length - A picture of Dorian Gray by Wilde, Oliver Twist by Dickens or Paradise Lost by Milton?

```
# url needs to be the web address for the text file
readplay <- function(url) {
  # Load in data
  ans <- scan(url, what = '', quiet = TRUE)
  # Remove punctuation
  ans2 <- gsub('[[:punct:]]', '', ans)
  # Remove newlines and tab characters
  ans3 <- gsub('\\s',' ',ans2)
  # Split it all into separate words
  ans4 <- unlist(strsplit(ans3,' '))
  # Get rid of empty strings
  ans5 <- ans4[ans4 != '']
  return(ans5)
}
```

```
wilde <- readplay('http://www.textfiles.com/etext/AUTHORS/WILDE/wilde-picture-615.txt')
dickens <- readplay('http://www.textfiles.com/etext/AUTHORS/DICKENS/dickens-oliver-627.txt')
milton <- readplay('http://www.textfiles.com/etext/AUTHORS/MILTON/milton-paradise-108.txt')
```