Homework 1

Question 1: I don't know

1. Logic questions
    1. `FALSE & NA` returns `FALSE` since the statement `FALSE` is unambiguous
    2. `TRUE & NA` returns `NA` since we don't know if the former statement is true or false. It is ambiguous
    3. `TRUE | NA` returns `TRUE` since R just needs to evaluate the first term (i.e. `TRUE`) to return `TRUE`
    4. `mean(c(3,5,7,NA))` returns `NA` since `NA` is one of the argument, making it ambiguous.

Plot Function

```
function (x, ...) # This starts the function
    {
    plot2 <- function(x, xlab = names(x)[1L], ylab = names(x)[2L], )
    plot(x[[1L]], x[[2L]], xlab = xlab, ylab = ylab, ...) # Creates a function that uses
    the generic 'plot' function to plot a x-y graph or boxplot.
    The labels are generated by the names of the columns.
    if (!is.data.frame(x)) # If this logic doesn't work, no plot will be produced
        stop("'plot.data.frame' applied to non data frame")
    if (ncol(x) == 1) { # If the dataframe is only 1 column long
        x1 <- x[[1L]] # Changes the dataframe into a vector
        cl <- class(x1)
        if (cl %in% c("integer", "numeric"))
            stripchart(x1, ...) # If the class of the vector is numeric, a 1D scatterplot is produced
        else plot(x1, ...) # If not, then a bar graph is produced
    }
    else if (ncol(x) == 2) { # Uses the plot2 function if the number of columns = 2
        plot2(x, ...)
    }
    else {
        pairs(data.matrix(x), ...) # Makes a matrix of variable pairs. There are no graphs in the
        diagonal because the x and y are the same variables
    }
}
```

Global Environment

1.

```
##  [1] ".GlobalEnv"        "package:MASS"      "package:stats"
##  [4] "package:graphics"  "package:grDevices" "package:utils"
##  [7] "package:datasets"  "package:methods"   "Autoloads"
## [10] "package:base"
```

2. `mean(height)` takes the original `height` variable that is multiplied by 2.54 `mean(women$height)` takes the original `height` is not modified by 2.54

Complex Numbers

```
exp(1i*pi) + 1
```

```
## [1] 0+1.224647e-16i
```

```
sin(pi)
```

```
## [1] 1.224647e-16
```

It gives an answer close to 0 but not quite zero; there's a very small imaginary number component. Euler's identity expands into cos(pi) + isin(pi). sin(pi) is not zero, but is actually a very small positive number. This is a result of floating-point arithmetic in R. Computers use a binary floating-point format that don't accurately represent numbers (but comes close to it).