hw3

Problem 1

Here we read in the `states` data frame, directly from a URL (a cool feature of `read.table()`):

```
states <- read.table("http://library.open.oregonstate.edu/doc/computationalbiology/states.txt",
                     header = TRUE,
                     sep = "\t",
                     comment.char = "#",
                     stringsAsFactors = FALSE)

# your code, using subset()


## uncomment to test:
# print(states_gradincome_high)
```

Using `subset()`, extract a sub-dataframe into a variable called `states_gradincome_high`, which should include all columns, but only rows where `income` is greater than the median of `income`, *or* where `hs_grad` is greater than the median of `hs_grad` (there are 35 such rows).

Next, do the same thing using `[row_selector, col_selector]` syntax.

```
# your code, using [row_selector, col_selector] syntax


## uncomment to test:
# print(states_gradincome_high)
```

Problem 2

Extract all of the *other* rows into a sub-dataframe called `states_gradincome_low`. You can use either `subset()` or `[row_selector, col_selector]` syntax.

```
# your code


## uncomment to test:
# print(states_gradincome_low)
```

Problem 3

For this problem, we're going to write a function that takes the `states` data frame as a parameter called `states_param`, and compares the murder rate amongst states with income less than or equal to the median income, to the murder rate for all the others (where income is greater than the median) using a basic non-paired `t.test()`. The function should return just the p-value, but, that p-value should be stored in a dataframe with a single row and a single column called `income_murder_pvalue`. (I.e., the function will take a data frame as a parameter, and return a data frame as well.)

(You may need to set the column name to `"income_murder_pvalue"` after creating the data frame inside the function. If it helps, you can try solving the problem without a function first, and then 'wrap' your code in the function, being sure to have it take a parameter, only use local variables, and return the result.)

I've written the function skeleton and test code, you just have to write the function contents :)

```
# this function takes a data frame with columns for 'income' (numeric)
# and 'murder' (numeric). It compares murder rates for rows where income is
# less than or equal to the median, to murder rates where income is greater than the median.
# returned is a data frame with a single row, and a single column called `income_murder_pvalue`.
test_murder_by_income <- function(states_param) {
  # your code here
}

## uncomment to test
# answer <- test_murder_by_income(states)
# print(answer)
```

The output should be:

```
  income_murder_pvalue
1             0.222537
```