Assignment 2

Anh Le

October 1, 2014

```
## knitr configuration: http://yihui.name/knitr/options#chunk_options
opts_chunk$set(comment = "", error= TRUE, warning = FALSE, message = FALSE,
               tidy = FALSE, cache = F, echo = T,
               fig.width = 4, fig.height = 4, fig.align="center")
```

1 Question 1

Let's pursue further the in-class example of ordering / selecting one variable based on another. We have the following mock data.

```
mock_data <- data.frame(country=c("US", "UK", "South Africa", "Liberia"),
                        region=c("America", "Europe", "Africa", "Africa"),
                        gdppc=c(40000, 35000, 25000, 9000),
                        stringsAsFactors=FALSE)
mock_data

       country  region gdppc
1           US America 40000
2           UK  Europe 35000
3 South Africa  Africa 25000
4      Liberia  Africa  9000
```

What I showed you in class is to select the `country` variable based on the `gdppc` variable, like so:

```
# Get countries with GDP per capita > 10000
mock_data$country[mock_data$gdppc > 10000]

[1] "US"           "UK"           "South Africa"

# Get countries with above-average GDP per capita
mock_data$country[mock_data$gdppc > mean(mock_data$gdppc)]

[1] "US" "UK"
```

Now, the question is how to select countries that have `gdppc > 10000` AND belong in Africa? Phrased more generally, how do we subset the data frame using two / multiple conditions? (Google if you don't know how – I have phrased the question in very Google-able terms)

So here's your first assignment.

1. Using the mock data above, select Africans countries that have `gdppc > 10000`.
2. Download real data from package `WDI`, the subset the data according to some conditions that interests you. (E.g. List all African countries that have below / above average GDP per capita; What about other continents? Variables other than GDP, etc.)

Solution

1. Using mock data

```
mock_data$country[(mock_data$gdppc > 10000) & (mock_data$region == "Africa")]

[1] "South Africa"

# Equivalent, less typing way
with(mock_data, country[gdppc > 10000 & region == "Africa"])

[1] "South Africa"
```

2. Using real data

```
library("WDI")
search_res <- WDIsearch(string="health")
world_data <- WDI(country="all", indicator = c("NY.GDP.PCAP.CD", "SH.XPD.PCAP"),
                   start = 2010, end = 2010, extra=TRUE)
world_data <- world_data[world_data$region != "Aggregates", ]
world_data <- world_data[complete.cases(world_data), ]
```

Countries with above average GDP per capita but below average health expenditure per capita

```
with(world_data, country[NY.GDP.PCAP.CD > mean(NY.GDP.PCAP.CD, na.rm=T) &
                           SH.XPD.PCAP < mean(SH.XPD.PCAP, na.rm=T)])

 [1] "Antigua and Barbuda" "Barbados"            "Bahrain"
 [4] "Brunei Darussalam"   "Estonia"             "Equatorial Guinea"
 [7] "St. Kitts and Nevis" "Oman"                "Saudi Arabia"
[10] "Trinidad and Tobago" "Venezuela, RB"
```

Which region do these countries belong?

```
with(world_data, region[NY.GDP.PCAP.CD > mean(NY.GDP.PCAP.CD, na.rm=T) &
                           SH.XPD.PCAP < mean(SH.XPD.PCAP, na.rm=T)])
```

```
 [1] Latin America & Caribbean (all income levels)
 [2] Latin America & Caribbean (all income levels)
 [3] Middle East & North Africa (all income levels)
 [4] East Asia & Pacific (all income levels)
 [5] Europe & Central Asia (all income levels)
 [6] Sub-Saharan Africa (all income levels)
 [7] Latin America & Caribbean (all income levels)
 [8] Middle East & North Africa (all income levels)
 [9] Middle East & North Africa (all income levels)
[10] Latin America & Caribbean (all income levels)
[11] Latin America & Caribbean (all income levels)
8 Levels: Aggregates ... Sub-Saharan Africa (all income levels)
```

The result is a factor vector, let's convert to character for easy reading

```
as.character(with(world_data, region[NY.GDP.PCAP.CD > mean(NY.GDP.PCAP.CD, na.rm=T) &
                           SH.XPD.PCAP < mean(SH.XPD.PCAP, na.rm=T)]))
```

```
 [1] "Latin America & Caribbean (all income levels)"
 [2] "Latin America & Caribbean (all income levels)"
 [3] "Middle East & North Africa (all income levels)"
 [4] "East Asia & Pacific (all income levels)"
 [5] "Europe & Central Asia (all income levels)"
 [6] "Sub-Saharan Africa (all income levels)"
 [7] "Latin America & Caribbean (all income levels)"
 [8] "Middle East & North Africa (all income levels)"
 [9] "Middle East & North Africa (all income levels)"
[10] "Latin America & Caribbean (all income levels)"
[11] "Latin America & Caribbean (all income levels)"
```

Still a bit ugly with the `(all income levels)`. Let's clean up using `strsplit()` we learned in class

```
regions_of_interests <- as.character(with(world_data,
  region[NY.GDP.PCAP.CD > mean(NY.GDP.PCAP.CD, na.rm=T) &
  SH.XPD.PCAP < mean(SH.XPD.PCAP, na.rm=T)]))

unlist(strsplit(regions_of_interests, split=" \\(all income levels)"))
```

```
 [1] "Latin America & Caribbean"  "Latin America & Caribbean"
 [3] "Middle East & North Africa" "East Asia & Pacific"
 [5] "Europe & Central Asia"      "Sub-Saharan Africa"
 [7] "Latin America & Caribbean"  "Middle East & North Africa"
 [9] "Middle East & North Africa" "Latin America & Caribbean"
[11] "Latin America & Caribbean"
```

2 Problem 2 – Problem 1.9.1 in the book

This problem involves data frame – re-read the book chapter on data frame if necessary

```
library(DAAG) # install if you have it yet
```

The following table gives the size of the floor area (ha) and the price ($000), for 15 houses sold in the Canberra (Australia) suburb of Aranda in 1999.

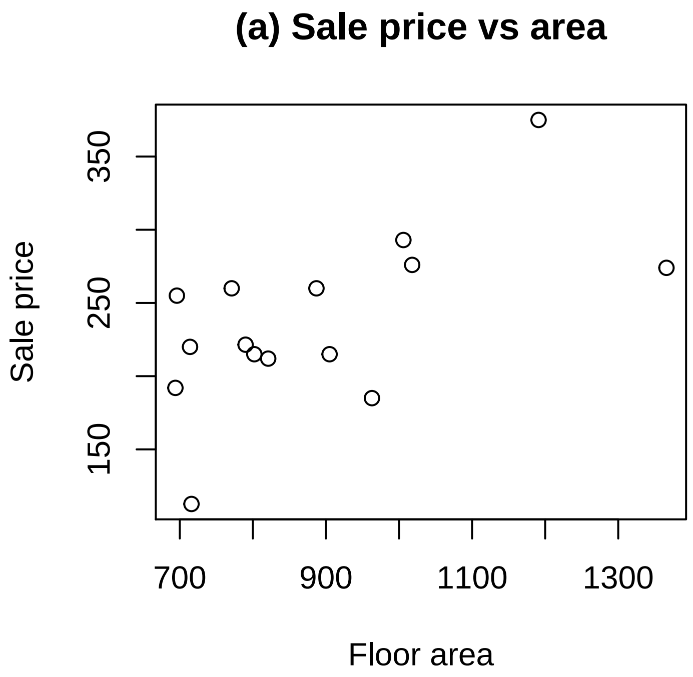
```
houseprices

   area bedrooms sale.price
9   694        4      192.0
10  905        4      215.0
11  802        4      215.0
12 1366        4      274.0
13  716        4      112.7
14  963        4      185.0
15  821        4      212.0
16  714        4      220.0
17 1018        4      276.0
18  887        4      260.0
19  790        4      221.5
20  696        5      255.0
21  771        5      260.0
22 1006        5      293.0
23 1191        6      375.0
```

1. Plot `sale.price` versus `area`.
2. Use the `hist()` command to plot a histogram of the sale prices.
3. Repeat (a) and (b) after taking logarithms of sale prices.
4. The two histograms emphasize different parts of the range of sale prices. Describe the differences.

Solution

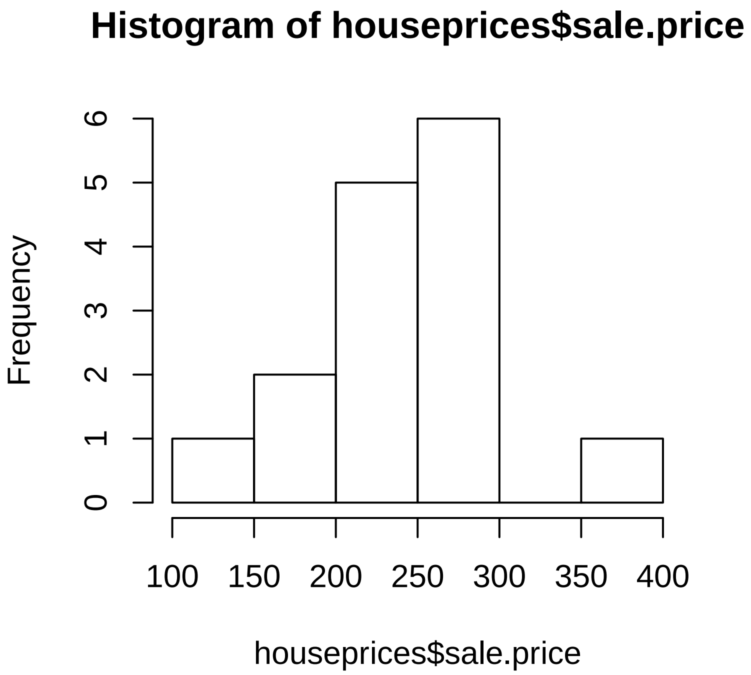
1. Plot `sale.price()` versus `area()`

```r
plot(sale.price ~ area, data=houseprices,
     main="(a) Sale price vs area",
     xlab="Floor area", ylab="Sale price")
```



2. Use the `hist()` command to plot a histogram of the sale prices

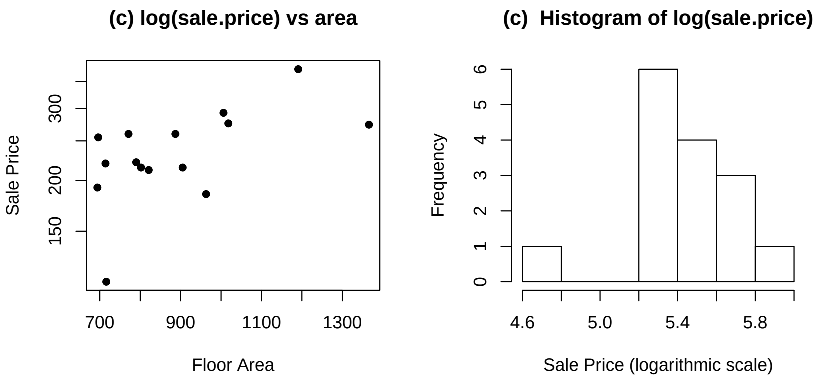
```r
hist(houseprices$sale.price)
```



3. The following code demonstrates the use of the `log="y"` argument to cause `plot` to use a logarithmic scale on the `y` axis, but with axis tick labels that are specified in the original units.

   The next oneputs a logarithmic scale on the x-axis of the histogram.

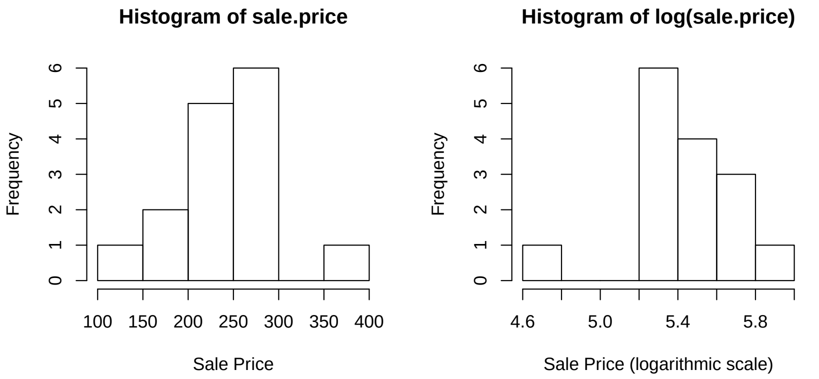
```
par(mfrow=c(1, 2))
plot(sale.price ~ area, data=houseprices, log="y",
     pch=16, xlab="Floor Area", ylab="Sale Price",
     main="(c) log(sale.price) vs area")
hist(log(houseprices$sale.price),
     xlab="Sale Price (logarithmic scale)",
     main="(c)  Histogram of log(sale.price)")
```



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

4. Compare the two histograms

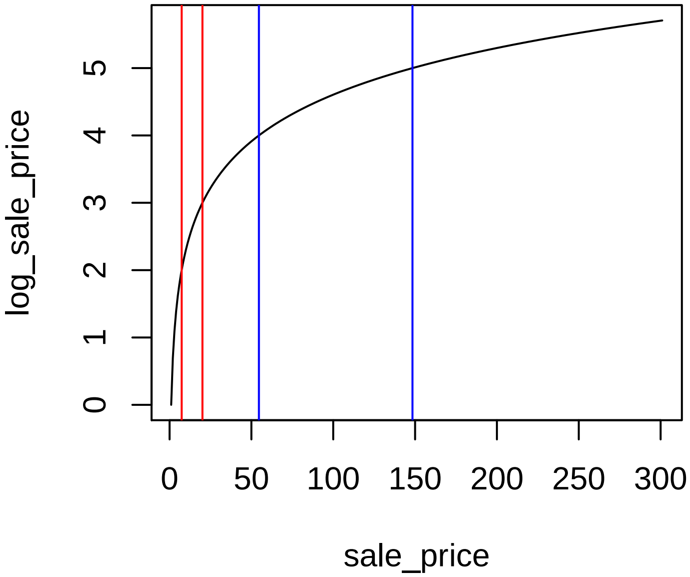
```r
par(mfrow=c(1, 2))
hist(houseprices$sale.price,
     xlab="Sale Price",
     main="Histogram of sale.price")
hist(log(houseprices$sale.price),
     xlab="Sale Price (logarithmic scale)",
     main="Histogram of log(sale.price)")
```



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

At log scale, the small difference (i.e. to the left of the axis) is indistinguisable. The log scale emphasizes the larger spread. To see this more clearly, consider the following plot:

```
plot(log(seq(100:400)), type="l", ylab="log_sale_price", xlab="sale_price")
abline(v=exp(2), col="red")
abline(v=exp(3), col="red")
abline(v=exp(4), col="blue")
abline(v=exp(5), col="blue")
```



Notice that one unit increase in `log_sale_price` does not correspond with the same increase in `sale_price`. For example, `log_sale_price` increases from 2 to 3 between the red lines, and from 4 to 5 between the blue lines (an increase of 1 in both cases). However, the change of `sale_price` between the blue lines is much greater than between the red lines. More generally, the farther we are to the right, the larger 1 unit in log scale becomes.

Back to our original question, this means that to the right of our log scale histogram, one unit in the x axis covers a lot more houses. Thus, the histogram is higher to the right.