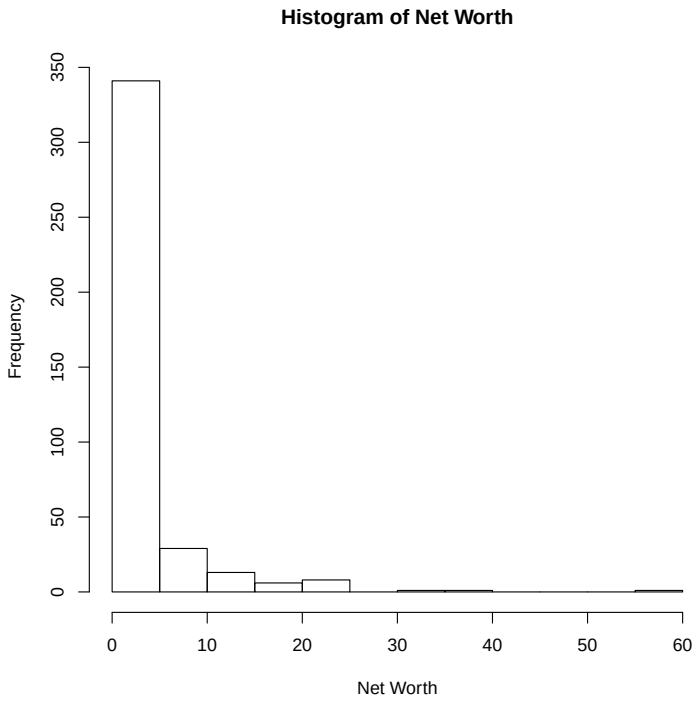
# Homework 9: Solutions

36-350, Fall 2011

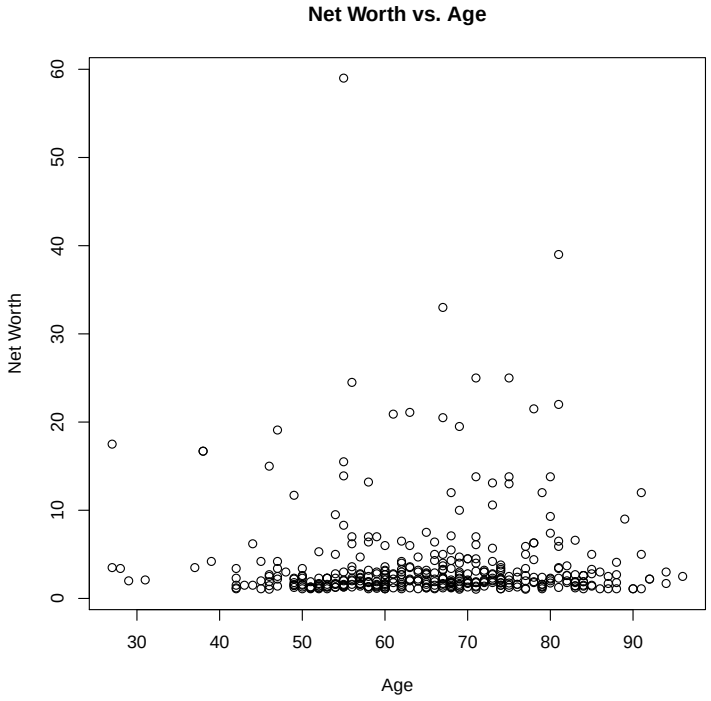
1. (a) Solution

   We provide two different solutions in the addition $R$ file (also on the website).

   (b) Solution



(c) Solution




```
> subset(df, age == min(age, na.rm = T))
   rank             name net.worth age                 residence   source
14   14  Mark Zuckerberg      17.5  27     Palo Alto, California Facebook
92   91 Dustin Moskovitz       3.5  27 San Francisco, California Facebook
> subset(df, age == max(age, na.rm = T))
    rank              name net.worth age               residence                source
163  159 David Rockefeller       2.5  96 Sleepy Hollow, New York Standard Oil, banking
```

The youngest people on the list are Mark Zuckerberg and Dustin Moskovitz, and the oldest is David Rockefeller.

(d) Solution

First you must notice that for each person they may have multiple sources of income, which are separated by commas. Therefore we collapse all the sources together into one long string, separating the sources from different people by ", ". Now we need to just split the string by commas, and count the number of occurrences of strings using `table()`. Building the script from the inside out (starting with `paste()`, then `strsplit()`, etc.) we can eventually do this in one line:

```
> as.matrix(rev(sort(table(strsplit(paste(df$source,collapse = ", "), split = ", "))))[1:10])
                  [,1]
investments         51
real estate         35
hedge funds         26
hotels              17
leveraged buyouts   12
oil                 11
Wal-Mart             7
oil &amp;amp; gas        6
```

```
casinos              6
Cargill Inc.         6
```

We use `rev()` to reverse the vector returned by `sort()` because `sort()` sorts in order of increasing value, and we instead want the ten largest values.

(e) Solution

The resulting column of residences in `df` is of the format "*city_name*, *state_name*", so we need to do some more regular expressions to extract just the states.

```
state.pat <- ", (([[:alnum:]_] ?)+)$"
state.match <- regexec(state.pat, df$residence)
xxx <- regmatches(df$residence, state.match)
```

We use `regexec` to be able to extract the groups of interest, here being all text after the comma and space separating city name from state name. The result is a list of length 400, where each item is a vector of strings: the first element being the value of the entire string matched by the pattern, and the second being the string matched within the first set of parentheses. We can loop through this list and extract the second elements using an `sapply()`.

```
> as.matrix(rev(sort(table(sapply(xxx,function(x){x[2]})))))[1:10])
            [,1]
California    88
New York      64
Texas         47
Florida       28
Illinois      18
Connecticut   11
Wisconsin      9
Michigan       9
Maryland       8
Washington     7
```