DATA 607—Homework No. 1

Ben Horvath

August 28, 2018

Load libraries:

```
library(RCurl)
```

First, let's load the data directly from the source (though a copy is saved in the `./data/` directory):

```
data_url <- 'https://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/agaricu
original <- getURL(data_url)
df <- read.csv(text=original, header=FALSE, stringsAsFactors=FALSE)
head(df)
```

```
##   V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20
## 1  p  x  s  n  t  p  f  c  n   k   e   e   s   s   w   w   p   w   o   p
## 2  e  x  s  y  t  a  f  c  b   k   e   c   s   s   w   w   p   w   o   p
## 3  e  b  s  w  t  l  f  c  b   n   e   c   s   s   w   w   p   w   o   p
## 4  p  x  y  w  t  p  f  c  n   n   e   e   s   s   w   w   p   w   o   p
## 5  e  x  s  g  f  n  f  w  b   k   t   e   s   s   w   w   p   w   o   e
## 6  e  x  y  y  t  a  f  c  b   n   e   c   s   s   w   w   p   w   o   p
##   V21 V22 V23
## 1   k   s   u
## 2   n   n   g
## 3   n   n   m
## 4   k   s   u
## 5   n   a   g
## 6   k   n   g
```

Fill in the column names and subset just a handful:

```
colnames(df) <- c('poisonous',
                  'cap_shape',
                  'cap_surface',
                  'cap_color',
                  'bruises',
```

```
                  'odor',
                  'gill_attachment',
                  'gill_spacing',
                  'gill_size',
                  'gill_color',
                  'stalk_shape',
                  'stalk_root',
                  'stalk_surface_above_ring',
                  'stalk_surface_below_ring',
                  'stalk_color_above_ring',
                  'stalk_color_below_ring',
                  'veil_type',
                  'veil_color',
                  'ring_number',
                  'ring_type',
                  'spore_print_color',
                  'population',
                  'habitat')

cols <- c('poisonous', 'bruises', 'gill_size', 'ring_number')
df <- df[cols]
head(df)
```

```
##   poisonous bruises gill_size ring_number
## 1         p       t         n           o
## 2         e       t         b           o
## 3         e       t         b           o
## 4         p       t         n           o
## 5         e       f         b           o
## 6         e       t         b           o
```

The remaining task is the de-abbreviate the data, converting each entry to a meaningful designation.

One way to do this would be to use many `gsub()` commands. However, a custom function that makes multiple substitutions at one go might make the job a little cleaner and easier to read.

The function `gsub_map()` accepts a string and a mapping (named list) of

pattern-replacements, performing multiple `gsub()` operations together:

```
gsub_map <- function(s, mapping) {
    # Accepts a mapping of pattern-replacements on a string s, allowing more
    # compact operations involving mutliple substitutions on the same string
    # sequentially
    for (i in 1:length(mapping)) {
        pattern <- names(mapping[i])
        replacement <- mapping[i]
        s <- gsub(pattern, replacement, s)
    }
    return(s)
}

# Example
gsub_map('foo bar', list(foo='foo1', bar='bar1'))
```

```
## [1] "foo1 bar1"
```

Using `sapply()` to apply to function to each row of the columns:

```
df$poisonous <- sapply(df$poisonous, gsub_map, list(e='edible', p='poisonous'))
df$bruises <- sapply(df$bruises, gsub_map, list(t='TRUE', f='FALSE'))
df$gill_size <- sapply(df$gill_size, gsub_map, list(b='broad', n='narrow'))
df$ring_number <- sapply(df$ring_number, gsub_map, list(n='0', 'o'='1', 't'='2'))

head(df)
```

```
##   poisonous bruises gill_size ring_number
## 1 poisonous    TRUE    narrow           1
## 2    edible    TRUE     broad           1
## 3    edible    TRUE     broad           1
## 4 poisonous    TRUE    narrow           1
## 5    edible   FALSE     broad           1
## 6    edible    TRUE     broad           1
```

Finally, convert to proper R data types:

```
df$bruises <- as.logical(df$bruises)
df$ring_number <- as.integer(df$ring_number)
```

```r
head(df)
```

```
##   poisonous bruises gill_size ring_number
## 1 poisonous    TRUE    narrow           1
## 2    edible    TRUE     broad           1
## 3    edible    TRUE     broad           1
## 4 poisonous    TRUE    narrow           1
## 5    edible   FALSE     broad           1
## 6    edible    TRUE     broad           1
```

```r
sapply(df, class)
```

```
##   poisonous     bruises   gill_size ring_number
## "character"   "logical" "character"   "integer"
```