Data Cleaning Tutorials

Eilif Mikkelsen

Introduction

In an academic setting data sets are often presented as clean and orderly sets of information. For most real-world applications this couldn't be farther from the truth. In this tutorial we will cover some core data cleaning methods. This is by no means exhaustive however it should leave the reader with a strong set of data manipulation skills. For context, I wrote this document to hone my data manipulation skills in R after spending years working with the Python Pandas data library which provides data structures and features similar to the R `data.frame` object.

Setup

Here is the output information on the operating system and R version used to generate this tutorial. The tools used in this tutorial are elemental functions of R as such is it not expected that this tutorial will become out-of-date. If you find issues, please feel free to edit this file and submit a pull request to this repository. Where specific packages are required, additional notes will be made.

```
version
```

```
##                _
## platform       x86_64-apple-darwin13.4.0
## arch           x86_64
## os             darwin13.4.0
## system         x86_64, darwin13.4.0
## status
## major          3
## minor          3.3
## year           2017
## month          03
## day            06
## svn rev        72310
## language       R
## version.string R version 3.3.3 (2017-03-06)
## nickname       Another Canoe
```

Topics Covered

- Cleaning string categories
- Context aware unit standardization
- Datetime parsing

Let's start by loading and looking at the first few rows of the data. The ix columns is an arbitrary row number that was created along with the dataset. We will use this to uniquely identify rows. Additionally, we want all string columns to be treated as characters rather than "factors", a special R data type.

```
# Loading the CSV
in_data = read.csv('./horrible_data.csv', stringsAsFactors=FALSE)

# Setting the row names to the `ix` column
```

```
row.names(in_data) = in_data$ix

head(in_data)
```

```
##   ix        BAC     age            collection_date             drink_type
## 0  0 0.05725438  1/7/66 2018-01-29T12:21:46.232643                  Merlot
## 1  1 0.15269702 8/19/68 2018-01-29T12:21:46.233977 Franziskaner Weissbier
## 2  2 0.05564855      50 2018-01-29T12:21:46.234572                    Wine
## 3  3 0.03288376     671 2018-01-29T12:21:46.235095                    Wine
## 4  4 0.03005566  4/2/60 2018-01-29T12:21:46.235582                     Vin
## 5  5 0.02840692 9/14/74 2018-01-29T12:21:46.236121                 whiskey
##   num_drinks    sex  units volume_consumed   weight
## 0        2.4      M                12.0000 177.8200
## 1        4.6      F metric       1634.5534 237.3310
## 2        2.5   Male                12.5000 185.4400
## 3        0.9 female metric        134.6799 265.0237
## 4        1.7   meal     SI        253.6016 315.3754
## 5        0.6 Female metric         28.0873 225.4100
```

Standardizing Text (string) Columns

Immediately upon observing the first 6 rows of the data we see that there are several variations of the same category for both the `units` and `sex` column. The goal is to identify what values in the column correspond to what categories.

Commands Used:

- `unique`
- `which`
- []

```
sex_vals = unique(in_data$sex)
sex_vals
```

```
## [1] "M"      "F"      "Male"   "female" "meal"   "Female" "femeal" ""
## [9] "male"
```

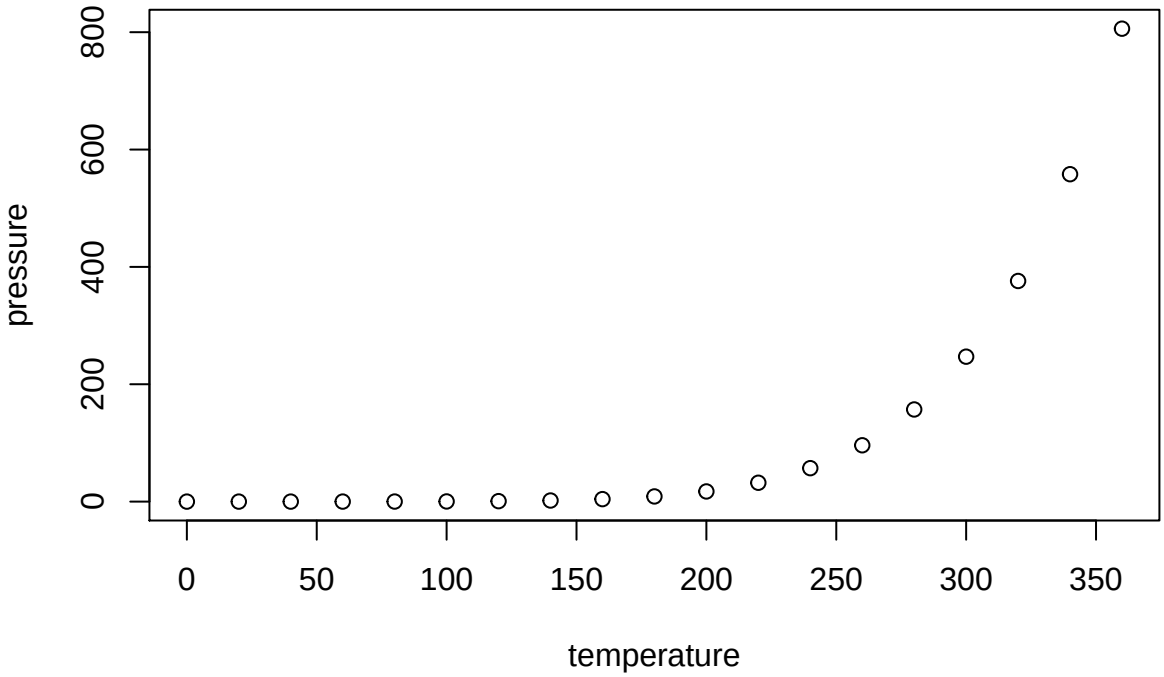
We see from the output that this dataset has two correct options for `sex`, Male and Female. We also observe that the dataset contains a variety of abbreviations and typos for the two categories. To standardize the values such that all observations that should be recorded as Male are updated accordingly. We will perform the same operation for Female.

```
units_vales = unique(in_data$units)
units_vales
```

```
## [1] ""       "metric" "SI"
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.