Springboard: Foundations of Data Science

Data Wrangling Exercise 2

Tom Skawski II

July 24-, 2016

Step 00 - Prepare Directory and load packages

```
# Mac:
setwd("~tomskawski/Dropbox/Data Science/Data Wrangling/DW2")
# PC
# setwd("C:/Users/tskawski.WESTERN/Desktop/Dropbox/Data Science/Data Wrangling/DW2")

# install.packages("tidyr")
library("tidyr")

# install.packages("dplyr")
library("dplyr")
```

Step 0: Load the data in RStudio

```
mydata <- read.csv("titanic3.csv", stringsAsFactors = FALSE)
# Save the data set as a CSV file called titanic_original.csv and load it in RStudio into a data frame.
```

```
# Check class of columns. Should not be factors.
sapply(mydata, class)
```

```
##      pclass    survived        name         sex         age       sibsp
##   "integer"   "integer" "character" "character"   "numeric"   "integer"
##       parch      ticket        fare       cabin    embarked        boat
##   "integer" "character"   "numeric" "character" "character" "character"
##        body   home.dest
##   "integer" "character"
```

Step 1 - Port of embarkation

```
# The embarked column has some missing values, which are known to correspond to passengers who actually

mydata %>% group_by(embarked) %>% summarise(name = n())
```

```
## # A tibble: 4 x 2
##   embarked  name
##      <chr> <int>
## 1              3
```

```
## 2        C   270
## 3        Q   123
## 4        S   914
```

```
# REGEX hunting yet again
grep("^$", mydata$embarked)
```

```
## [1]  169  285 1310
```

```
mydata$embarked <- gsub("^$", "S", mydata$embarked)
```

```
# Re-check
mydata %>% group_by(embarked) %>% summarise(name = n())
```

```
## # A tibble: 3 x 2
##   embarked  name
##      <chr> <int>
## 1        C   270
## 2        Q   123
## 3        S   917
```

Step 2: Age

```
# You'll notice that a lot of the values in the Age column are missing. While there are many ways to fi
# Calculate the mean of the Age column and use that value to populate the missing values
# Think about other ways you could have populated the missing values in the age column. Why would you p

mydata %>% group_by(age) %>% summarise(name = n())
```

```
## # A tibble: 99 x 2
##       age  name
##     <dbl> <int>
## 1  0.1667     1
## 2  0.3333     1
## 3  0.4167     1
## 4  0.6667     1
## 5  0.7500     3
## 6  0.8333     3
## 7  0.9167     2
## 8  1.0000    10
## 9  2.0000    12
## 10 3.0000     7
## # ... with 89 more rows
```

```
mydata$age[is.na(mydata$age)] <- mean(mydata$age, na.rm = TRUE)

mydata$age
```

```
##    [1] 29.00000  0.91670  2.00000 30.00000 25.00000 48.00000 63.00000
##    [8] 39.00000 53.00000 71.00000 47.00000 18.00000 24.00000 26.00000
##   [15] 80.00000 29.88113 24.00000 50.00000 32.00000 36.00000 37.00000
##   [22] 47.00000 26.00000 42.00000 29.00000 25.00000 25.00000 19.00000
##   [29] 35.00000 28.00000 45.00000 40.00000 30.00000 58.00000 42.00000
##   [36] 45.00000 22.00000 29.88113 41.00000 48.00000 29.88113 44.00000
##   [43] 59.00000 60.00000 41.00000 45.00000 29.88113 42.00000 53.00000
##   [50] 36.00000 58.00000 33.00000 28.00000 17.00000 11.00000 14.00000
##   [57] 36.00000 36.00000 49.00000 29.88113 36.00000 76.00000 46.00000
##   [64] 47.00000 27.00000 33.00000 36.00000 30.00000 45.00000 29.88113
##   [71] 29.88113 27.00000 26.00000 22.00000 29.88113 47.00000 39.00000
##   [78] 37.00000 64.00000 55.00000 29.88113 70.00000 36.00000 64.00000
##   [85] 39.00000 38.00000 51.00000 27.00000 33.00000 31.00000 27.00000
##   [92] 31.00000 17.00000 53.00000  4.00000 54.00000 50.00000 27.00000
##   [99] 48.00000 48.00000 49.00000 39.00000 23.00000 38.00000 54.00000
##  [106] 36.00000 29.88113 29.88113 29.88113 36.00000 30.00000 24.00000
##  [113] 28.00000 23.00000 19.00000 64.00000 60.00000 30.00000 29.88113
##  [120] 50.00000 43.00000 29.88113 22.00000 60.00000 48.00000 29.88113
##  [127] 37.00000 35.00000 47.00000 35.00000 22.00000 45.00000 24.00000
##  [134] 49.00000 29.88113 71.00000 53.00000 19.00000 38.00000 58.00000
##  [141] 23.00000 45.00000 46.00000 25.00000 25.00000 48.00000 49.00000
##  [148] 29.88113 45.00000 35.00000 40.00000 27.00000 29.88113 24.00000
##  [155] 55.00000 52.00000 42.00000 29.88113 55.00000 16.00000 44.00000
##  [162] 51.00000 42.00000 35.00000 35.00000 38.00000 29.88113 35.00000
##  [169] 38.00000 50.00000 49.00000 46.00000 50.00000 32.50000 58.00000
##  [176] 41.00000 29.88113 42.00000 45.00000 29.88113 39.00000 49.00000
##  [183] 30.00000 35.00000 29.88113 42.00000 55.00000 16.00000 51.00000
##  [190] 29.00000 21.00000 30.00000 58.00000 15.00000 30.00000 16.00000
##  [197] 29.88113 19.00000 18.00000 24.00000 46.00000 54.00000 36.00000
##  [204] 28.00000 29.88113 65.00000 44.00000 33.00000 37.00000 30.00000
##  [211] 55.00000 47.00000 37.00000 31.00000 23.00000 58.00000 19.00000
##  [218] 64.00000 39.00000 29.88113 22.00000 65.00000 28.50000 29.88113
##  [225] 45.50000 23.00000 29.00000 22.00000 18.00000 17.00000 30.00000
##  [232] 52.00000 47.00000 56.00000 38.00000 29.88113 22.00000 29.88113
##  [239] 43.00000 31.00000 45.00000 29.88113 33.00000 46.00000 36.00000
##  [246] 33.00000 55.00000 54.00000 33.00000 13.00000 18.00000 21.00000
##  [253] 61.00000 48.00000 29.88113 24.00000 29.88113 35.00000 30.00000
##  [260] 34.00000 40.00000 35.00000 50.00000 39.00000 56.00000 28.00000
##  [267] 56.00000 56.00000 24.00000 29.88113 18.00000 24.00000 23.00000
##  [274]  6.00000 45.00000 40.00000 57.00000 29.88113 32.00000 62.00000
##  [281] 54.00000 43.00000 52.00000 29.88113 62.00000 67.00000 63.00000
##  [288] 61.00000 48.00000 18.00000 52.00000 39.00000 48.00000 29.88113
##  [295] 49.00000 17.00000 39.00000 29.88113 31.00000 40.00000 61.00000
##  [302] 47.00000 35.00000 64.00000 60.00000 60.00000 54.00000 21.00000
##  [309] 55.00000 31.00000 57.00000 45.00000 50.00000 27.00000 50.00000
##  [316] 21.00000 51.00000 21.00000 29.88113 31.00000 29.88113 62.00000
##  [323] 36.00000 30.00000 28.00000 30.00000 18.00000 25.00000 34.00000
##  [330] 36.00000 57.00000 18.00000 23.00000 36.00000 28.00000 51.00000
##  [337] 32.00000 19.00000 28.00000  1.00000  4.00000 12.00000 36.00000
##  [344] 34.00000 19.00000 23.00000 26.00000 42.00000 27.00000 24.00000
##  [351] 15.00000 60.00000 40.00000 20.00000 25.00000 36.00000 25.00000
##  [358] 42.00000 42.00000  0.83330 26.00000 22.00000 35.00000 29.88113
##  [365] 19.00000 44.00000 54.00000 52.00000 37.00000 29.00000 25.00000
##  [372] 45.00000 29.00000 28.00000 29.00000 28.00000 24.00000  8.00000
```

```
##  [379] 31.00000 31.00000 22.00000 30.00000 29.88113 21.00000 29.88113
##  [386]  8.00000 18.00000 48.00000 28.00000 32.00000 17.00000 29.00000
##  [393] 24.00000 25.00000 18.00000 18.00000 34.00000 54.00000  8.00000
##  [400] 42.00000 34.00000 27.00000 30.00000 23.00000 21.00000 18.00000
##  [407] 40.00000 29.00000 18.00000 36.00000 29.88113 38.00000 35.00000
##  [414] 38.00000 34.00000 34.00000 16.00000 26.00000 47.00000 21.00000
##  [421] 21.00000 24.00000 24.00000 34.00000 30.00000 52.00000 30.00000
##  [428]  0.66670 24.00000 44.00000  6.00000 28.00000 62.00000 30.00000
##  [435]  7.00000 43.00000 45.00000 24.00000 24.00000 49.00000 48.00000
##  [442] 55.00000 24.00000 32.00000 21.00000 18.00000 20.00000 23.00000
##  [449] 36.00000 54.00000 50.00000 44.00000 29.00000 21.00000 42.00000
##  [456] 63.00000 60.00000 33.00000 17.00000 42.00000 24.00000 47.00000
##  [463] 24.00000 22.00000 32.00000 23.00000 34.00000 24.00000 22.00000
##  [470] 29.88113 35.00000 45.00000 57.00000 29.88113 31.00000 26.00000
##  [477] 30.00000 29.88113  1.00000  3.00000 25.00000 22.00000 17.00000
##  [484] 29.88113 34.00000 36.00000 24.00000 61.00000 50.00000 42.00000
##  [491] 57.00000 29.88113  1.00000 31.00000 24.00000 29.88113 30.00000
##  [498] 40.00000 32.00000 30.00000 46.00000 13.00000 41.00000 19.00000
##  [505] 39.00000 48.00000 70.00000 27.00000 54.00000 39.00000 16.00000
##  [512] 62.00000 32.50000 14.00000  2.00000  3.00000 36.50000 26.00000
##  [519] 19.00000 28.00000 20.00000 29.00000 39.00000 22.00000 29.88113
##  [526] 23.00000 29.00000 28.00000 29.88113 50.00000 19.00000 29.88113
##  [533] 41.00000 21.00000 19.00000 43.00000 32.00000 34.00000 30.00000
##  [540] 27.00000  2.00000  8.00000 33.00000 36.00000 34.00000 30.00000
##  [547] 28.00000 23.00000  0.83330  3.00000 24.00000 50.00000 19.00000
##  [554] 21.00000 26.00000 25.00000 27.00000 25.00000 18.00000 20.00000
##  [561] 30.00000 59.00000 30.00000 35.00000 40.00000 25.00000 41.00000
##  [568] 25.00000 18.50000 14.00000 50.00000 23.00000 28.00000 27.00000
##  [575] 29.00000 27.00000 40.00000 31.00000 30.00000 23.00000 31.00000
##  [582] 29.88113 12.00000 40.00000 32.50000 27.00000 29.00000  2.00000
##  [589]  4.00000 29.00000  0.91670  5.00000 36.00000 33.00000 66.00000
##  [596] 29.88113 31.00000 29.88113 26.00000 24.00000 42.00000 13.00000
##  [603] 16.00000 35.00000 16.00000 25.00000 20.00000 18.00000 30.00000
##  [610] 26.00000 40.00000  0.83330 18.00000 26.00000 26.00000 20.00000
##  [617] 24.00000 25.00000 35.00000 18.00000 32.00000 19.00000  4.00000
##  [624]  6.00000  2.00000 17.00000 38.00000  9.00000 11.00000 39.00000
##  [631] 27.00000 26.00000 39.00000 20.00000 26.00000 25.00000 18.00000
##  [638] 24.00000 35.00000  5.00000  9.00000  3.00000 13.00000  5.00000
##  [645] 40.00000 23.00000 38.00000 45.00000 21.00000 23.00000 17.00000
##  [652] 30.00000 23.00000 13.00000 20.00000 32.00000 33.00000  0.75000
##  [659]  0.75000  5.00000 24.00000 18.00000 40.00000 26.00000 20.00000
##  [666] 18.00000 45.00000 27.00000 22.00000 19.00000 26.00000 22.00000
##  [673] 29.88113 20.00000 32.00000 21.00000 18.00000 26.00000  6.00000
##  [680]  9.00000 29.88113 29.88113 29.88113 40.00000 32.00000 21.00000
##  [687] 22.00000 20.00000 29.00000 22.00000 22.00000 35.00000 18.50000
##  [694] 21.00000 19.00000 18.00000 21.00000 30.00000 18.00000 38.00000
##  [701] 17.00000 17.00000 21.00000 21.00000 21.00000 29.88113 29.88113
##  [708] 28.00000 24.00000 16.00000 37.00000 28.00000 24.00000 21.00000
##  [715] 32.00000 29.00000 26.00000 18.00000 20.00000 18.00000 24.00000
##  [722] 36.00000 24.00000 31.00000 31.00000 22.00000 30.00000 70.50000
##  [729] 43.00000 35.00000 27.00000 19.00000 30.00000  9.00000  3.00000
##  [736] 36.00000 59.00000 19.00000 17.00000 44.00000 17.00000 22.50000
##  [743] 45.00000 22.00000 19.00000 30.00000 29.00000  0.33330 34.00000
##  [750] 28.00000 27.00000 25.00000 24.00000 22.00000 21.00000 17.00000
```

```
##  [757] 29.88113 29.88113 36.50000 36.00000 30.00000 16.00000  1.00000
##  [764]  0.16670 26.00000 33.00000 25.00000 29.88113 29.88113 22.00000
##  [771] 36.00000 19.00000 17.00000 42.00000 43.00000 29.88113 32.00000
##  [778] 19.00000 30.00000 24.00000 23.00000 33.00000 65.00000 24.00000
##  [785] 23.00000 22.00000 18.00000 16.00000 45.00000 29.88113 39.00000
##  [792] 17.00000 15.00000 47.00000  5.00000 29.88113 40.50000 40.50000
##  [799] 29.88113 18.00000 29.88113 29.88113 29.88113 26.00000 29.88113
##  [806] 29.88113 21.00000  9.00000 29.88113 18.00000 16.00000 48.00000
##  [813] 29.88113 29.88113 25.00000 29.88113 29.88113 22.00000 16.00000
##  [820] 29.88113  9.00000 33.00000 41.00000 31.00000 38.00000  9.00000
##  [827]  1.00000 11.00000 10.00000 16.00000 14.00000 40.00000 43.00000
##  [834] 51.00000 32.00000 29.88113 20.00000 37.00000 28.00000 19.00000
##  [841] 24.00000 17.00000 29.88113 29.88113 28.00000 24.00000 20.00000
##  [848] 23.50000 41.00000 26.00000 21.00000 45.00000 29.88113 25.00000
##  [855] 29.88113 11.00000 29.88113 27.00000 29.88113 18.00000 26.00000
##  [862] 23.00000 22.00000 28.00000 28.00000 29.88113  2.00000 22.00000
##  [869] 43.00000 28.00000 27.00000 29.88113 29.88113 42.00000 29.88113
##  [876] 30.00000 29.88113 27.00000 25.00000 29.88113 29.00000 21.00000
##  [883] 29.88113 20.00000 48.00000 17.00000 29.88113 29.88113 34.00000
##  [890] 26.00000 22.00000 33.00000 31.00000 29.00000  4.00000  1.00000
##  [897] 49.00000 33.00000 19.00000 27.00000 29.88113 29.88113 29.88113
##  [904] 29.88113 23.00000 32.00000 27.00000 20.00000 21.00000 32.00000
##  [911] 17.00000 21.00000 30.00000 21.00000 33.00000 22.00000  4.00000
##  [918] 39.00000 29.88113 18.50000 29.88113 29.88113 29.88113 29.88113
##  [925] 34.50000 44.00000 29.88113 29.88113 29.88113 29.88113 29.88113
##  [932] 29.88113 22.00000 26.00000  4.00000 29.00000 26.00000  1.00000
##  [939] 18.00000 36.00000 29.88113 25.00000 29.88113 37.00000 29.88113
##  [946] 29.88113 29.88113 22.00000 29.88113 26.00000 29.00000 29.00000
##  [953] 22.00000 22.00000 29.88113 29.88113 29.88113 29.88113 29.88113
##  [960] 32.00000 34.50000 29.88113 29.88113 36.00000 39.00000 24.00000
##  [967] 25.00000 45.00000 36.00000 30.00000 20.00000 29.88113 28.00000
##  [974] 29.88113 30.00000 26.00000 29.88113 20.50000 27.00000 51.00000
##  [981] 23.00000 32.00000 29.88113 29.88113 29.88113 24.00000 22.00000
##  [988] 29.88113 29.88113 29.88113 29.00000 29.88113 30.50000 29.88113
##  [995] 29.88113 35.00000 33.00000 29.88113 29.88113 29.88113 29.88113
## [1002] 29.88113 29.88113 29.88113 29.88113 29.88113 29.88113 15.00000
## [1009] 35.00000 29.88113 24.00000 19.00000 29.88113 29.88113 29.88113
## [1016] 55.50000 29.88113 21.00000 29.88113 24.00000 21.00000 28.00000
## [1023] 29.88113 29.88113 25.00000  6.00000 27.00000 29.88113 29.88113
## [1030] 29.88113 29.88113 34.00000 29.88113 29.88113 29.88113 29.88113
## [1037] 29.88113 29.88113 29.88113 29.88113 24.00000 29.88113 29.88113
## [1044] 29.88113 29.88113 18.00000 22.00000 15.00000  1.00000 20.00000
## [1051] 19.00000 33.00000 29.88113 29.88113 29.88113 29.88113 12.00000
## [1058] 14.00000 29.00000 28.00000 18.00000 26.00000 21.00000 41.00000
## [1065] 39.00000 21.00000 28.50000 22.00000 61.00000 29.88113 29.88113
## [1072] 29.88113 29.88113 29.88113 29.88113 23.00000 29.88113 29.88113
## [1079] 29.88113 22.00000 29.88113 29.88113  9.00000 28.00000 42.00000
## [1086] 29.88113 31.00000 28.00000 32.00000 20.00000 23.00000 20.00000
## [1093] 20.00000 16.00000 31.00000 29.88113  2.00000  6.00000  3.00000
## [1100]  8.00000 29.00000  1.00000  7.00000  2.00000 16.00000 14.00000
## [1107] 41.00000 21.00000 19.00000 29.88113 32.00000  0.75000  3.00000
## [1114] 26.00000 29.88113 29.88113 29.88113 21.00000 25.00000 22.00000
## [1121] 25.00000 29.88113 29.88113 29.88113 29.88113 24.00000 28.00000
## [1128] 19.00000 29.88113 25.00000 18.00000 32.00000 29.88113 17.00000
```

```
## [1135] 24.00000 29.88113 29.88113 29.88113 29.88113 38.00000 21.00000
## [1142] 10.00000  4.00000  7.00000  2.00000  8.00000 39.00000 22.00000
## [1149] 35.00000 29.88113 29.88113 29.88113 50.00000 47.00000 29.88113
## [1156] 29.88113  2.00000 18.00000 41.00000 29.88113 50.00000 16.00000
## [1163] 29.88113 29.88113 29.88113 25.00000 29.88113 29.88113 29.88113
## [1170] 38.50000 29.88113 14.50000 29.88113 29.88113 29.88113 29.88113
## [1177] 29.88113 29.88113 29.88113 29.88113 29.88113 24.00000 21.00000
## [1184] 39.00000 29.88113 29.88113 29.88113  1.00000 24.00000  4.00000
## [1191] 25.00000 20.00000 24.50000 29.88113 29.88113 29.88113 29.00000
## [1198] 29.88113 29.88113 29.88113 29.88113 22.00000 29.88113 40.00000
## [1205] 21.00000 18.00000  4.00000 10.00000  9.00000  2.00000 40.00000
## [1212] 45.00000 29.88113 29.88113 29.88113 29.88113 29.88113 19.00000
## [1219] 30.00000 29.88113 32.00000 29.88113 33.00000 23.00000 21.00000
## [1226] 60.50000 19.00000 22.00000 31.00000 27.00000  2.00000 29.00000
## [1233] 16.00000 44.00000 25.00000 74.00000 14.00000 24.00000 25.00000
## [1240] 34.00000  0.41670 29.88113 29.88113 29.88113 16.00000 29.88113
## [1247] 29.88113 29.88113 32.00000 29.88113 29.88113 30.50000 44.00000
## [1254] 29.88113 25.00000 29.88113  7.00000  9.00000 29.00000 36.00000
## [1261] 18.00000 63.00000 29.88113 11.50000 40.50000 10.00000 36.00000
## [1268] 30.00000 29.88113 33.00000 28.00000 28.00000 47.00000 18.00000
## [1275] 31.00000 16.00000 31.00000 22.00000 20.00000 14.00000 22.00000
## [1282] 22.00000 29.88113 29.88113 29.88113 32.50000 38.00000 51.00000
## [1289] 18.00000 21.00000 47.00000 29.88113 29.88113 29.88113 28.50000
## [1296] 21.00000 27.00000 29.88113 36.00000 27.00000 15.00000 45.50000
## [1303] 29.88113 29.88113 14.50000 29.88113 26.50000 27.00000 29.00000
## [1310] 29.88113
```

Found this [here] (https://stackoverflow.com/questions/25835643/replacing-missing-values-in-r-with-column-mean)

```
# Used 'separate' function, once I found it.
# mydata <- separate(mydata, Product.code...number, c("Prod.code", "Prod.number"), sep = "-")
# mydata %>% group_by(Prod.code) %>% summarise(Prod.number = n())
```

Step 3 - 3: Lifeboat

You're interested in looking at the distribution of passengers in different lifeboats, but as we know, many passengers did not make it to a boat :-(This means that there are a lot of missing values in the boat column. Fill these empty slots with a dummy value e.g. the string 'None' or 'NA'

```
# kept new data with same/desired names as result columns.
# Prod.code <- c("p", "q", "v", "x")
# Prod.cat <- c("Smartphone", "Tablet", "TV", "Laptop")

# merge = data_frame(Prod.code, Prod.cat)

# mydata <- left_join(mydata, merge)

# MM: (made a vector, not a df, b/c vectors are smaller than dfs)
# categories = c(p = "smartphone", v = "tv", x = "laptop", q = "tablet")
# df <- rename(df, c("product code" = "product.code"))
# product_category <- categories[df$product.code]
# df <- mutate(df, product.cat = product_category)
```

Step 4 - 4: Cabin

You notice that many passengers don't have a cabin number associated with them. Does it make sense to fill missing cabin numbers with a value? What does a missing value here mean? You have a hunch that the fact that the cabin number is missing might be a useful indicator of survival. Create a new column has_cabin_number which has 1 if there is a cabin number, and 0 otherwise.

```
# mydata <- unite(mydata, full_address, address:country, sep = ",")
# not sure if this should be a new, combined column, or an additional column

# MM: (keeps original three fields, which I did not do)
#df <- mutate(df, full.address = paste(df$address, df$city, df$country, sep = ","))
```

Step 5 - Create dummy variables [No Step 5 in DW2]

```
# mydata$company_philips <- as.numeric(mydata$company == "philips")
# mydata$company_akzo <- as.numeric(mydata$company == "akzo")
# mydata$company_unilever <- as.numeric(mydata$company == "unilever")
# mydata$company_van_houten <- as.numeric(mydata$company == "van houten")
# mydata$product_smartphone <- as.numeric(mydata$Prod.cat == "Smartphone")
# mydata$product_tv <- as.numeric(mydata$Prod.cat == "TV")
# mydata$product_laptop <- as.numeric(mydata$Prod.cat == "Laptop")
# mydata$product_tablet <- as.numeric(mydata$Prod.cat == "Tablet")
```

Step 6 - 6: Submit the project on Github

Include your code, the original data as a CSV file titanic_original.csv, and the cleaned up data as a CSV file called titanic_clean.csv.

```
# Make file and submit
write.csv(mydata, file = "refine_clean.csv")

# Used github desktop, drag and drop
```