# Titanic Wikipedia Data Grab

*Gina Reynolds, Claus O. Wilke*

*Dec. 2017*

This file is written to collect the information about those on board Titanic from the Wikipedia pages on passengers and crew.

```
library(htmltab)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)
library(here)
```

```
## here() starts at /Users/evangelinereynolds/Google Drive/SideProjects/titanic.complete
```

```
library(tidyr)
```

```
sessionInfo()
```

```
## R version 3.4.1 (2017-06-30)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] tidyr_0.6.3   here_0.1      stringr_1.2.0 dplyr_0.7.3   htmltab_0.7.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.12     assertthat_0.2.0 digest_0.6.12    rprojroot_1.2
##  [5] R6_2.2.2         backports_1.1.0  magrittr_1.5     evaluate_0.10.1
##  [9] rlang_0.1.2      stringi_1.1.5    bindrcpp_0.2     rmarkdown_1.6
## [13] tools_3.4.1      glue_1.1.1       yaml_2.1.14      compiler_3.4.1
## [17] pkgconfig_2.0.1  htmltools_0.3.6  bindr_0.1        knitr_1.16
## [21] tibble_1.3.4
```

## Download raw data from Wikipedia

We grab the passenger and crew tables as raw html files from Wikipedia and store them in directory `./data-raw/RawData` for further processing.

```
if (!dir.exists(here("data-raw", "RawData"))) {
  dir.create(here("data-raw", "RawData"))
}

if (!file.exists(here("data-raw", "RawData", "Passengers2017-12-17.html")))  {
  download.file("https://en.wikipedia.org/wiki/Passengers_of_the_RMS_Titanic",
                destfile=here("data-raw", "RawData",
                              paste0("Passengers", Sys.Date(), ".html")))
}

if (!file.exists(here("data-raw", "RawData", "Crew2017-12-17.html"))) {
  download.file("https://en.wikipedia.org/wiki/Crew_of_the_RMS_Titanic",
                destfile=here("data-raw", "RawData",
                              paste0("Crew", Sys.Date(), ".html")))
}
```

## Passengers

We collect data from three tables in the passangers wikipedia page. The tables are of First, Second, and Third class passengers.

```
url <- here("data-raw", "RawData", "Passengers2017-12-17.html")
table1 <- htmltab(url, 1, rm_nodata_cols = F)
table2 <- htmltab(url, 2, rm_nodata_cols = F)
table3 <- htmltab(url, 3, rm_nodata_cols = F)
table1$Class <- "First"
table2$Class <- "Second"
table3$Class <- "Third"
passengers <- bind_rows(table1, table2, table3); dim(passengers)
```

```
## [1] 1319    9
```

```
# Names to snake case
names(passengers) <- str_replace(tolower(names(passengers)), " ", "_")
```

## Passengers data cleanup

Note wikipedia mistake for passengers for Everett, Washington, USA.

```
passengers[str_detect(passengers$boarded, "Everett"),]
```

```
##                                     name age            hometown
## 1025 Jeanie, Mrs. Beanie The (née Meanie)   6 London, England, UK
## 1026  Meanie, Miss Maliza Mae (née Jones)  24 London, England, UK
##                        boarded destination lifeboat body class home_country
## 1025 Everett, Washington, USA          14     <NA> <NA> Third  Southampton
## 1026 Everett, Washington, USA          14     <NA> <NA> Third  Southampton
```

Several entries are shifted one column to the left.

```
passengers[str_detect(passengers$boarded, "Everett"),"Lifeboat"] <- 14
passengers[str_detect(passengers$boarded, "Everett"),"Destination"] <- "Everett, Washington, USA"
passengers[str_detect(passengers$boarded, "Everett"),"Boarded"] <- NA
passengers[c(1025,1026),]
```

```
##                                      name age            hometown
## 1025 Jeanie, Mrs. Beanie The (née Meanie)   6 London, England, UK
## 1026  Meanie, Miss Maliza Mae (née Jones)  24 London, England, UK
##                        boarded destination lifeboat body class home_country
## 1025 Everett, Washington, USA          14     <NA> <NA> Third  Southampton
## 1026 Everett, Washington, USA          14     <NA> <NA> Third  Southampton
##      Lifeboat              Destination Boarded
## 1025       14 Everett, Washington, USA      NA
## 1026       14 Everett, Washington, USA      NA
```

## Passenger survival

Survival is indicated with a color in the table. html is style in <tr field.

```
lines <- readLines(url)
before_tables_line <- which(str_detect(lines, '<th>Lifeboat'))
grab_which <- which(c(rep(T, nrow(table1)), F,
                      rep(T, nrow(table2)), F,
                      rep(T, nrow(table3))))
temp_lines <- lines[before_tables_line[1]:length(lines)]
passengers$survival_outcome <-
  str_detect(temp_lines[str_detect(temp_lines, "<tr")], "style")[grab_which]
```

## Crew

There are are eight tables in the webpage that we collect, and combine into one table of crew.

```
url <- here("data-raw", "RawData", "Crew2017-12-17.html")
lines <- readLines(url)
before_tables_line <- which(str_detect(lines,'<th>Hometown'))
crew <- data_frame()
# section heading in Wikipedia before each chart
crew_types=c("Ship's officers", "Deck crew", "Engineering crew",
             "Victualling crew", "Restaurant staff", "Postal clerks",
             "Guarantee group", "Ship's orchestra")
for (i in 1:8){
  temp <- htmltab(url, i,rm_nodata_cols = F)  # grabs each table
  temp_lines <- lines[before_tables_line[i]:length(lines)]  # ids text line before table
  temp$survival_outcome <-  # using line color to id survival
    str_detect(temp_lines[str_detect(temp_lines, "<tr")], "style")[1:nrow(temp)]
  temp$crew_type=crew_types[i]  # type of crew member
  crew <- bind_rows(crew, temp)
}

crew$crew <- "Crew"
```

```
# convert variable names to snake case
names(crew) <- str_replace(tolower(names(crew)), " ", "_")
```

## Join passenger and crew tables

Preparation for full join. Some people are classified as crew and passengers, we need to ensure that their information matches exactly so we have a clean match.

```
passengers$hometown[passengers$hometown=="Belfast, Ireland, UK"] <-
  "Belfast, Ireland"
passengers$name[passengers$name=="Frost, Mr. Anthony Wood \"Archie\""] <-
  "Frost, Mr. Anthony Wood"
passengers$name[passengers$name=="Frost, Mr. Anthony Wood \"Artie\""] <-
  "Frost, Mr. Anthony Wood"

dim(passengers)
```

```
## [1] 1319   13
```

```
dim(crew)
```

```
## [1] 867  11
```

```
df <- full_join(passengers,crew)
```

```
## Joining, by = c("name", "age", "hometown", "boarded", "lifeboat", "body", "class", "survival_outcome
```

```
dim(df)  # this should be nine fewer than passangers plus crew
```

```
## [1] 2179   16
```

```
df$crew[is.na(df$crew)] <- "Not Crew"
df$survival_outcome <- ifelse(df$survival_outcome, "Survived", "Perished")
```

## Sex

We want to have sex of individuals, but there is not a unique column in the wikipedia data for this information, so we get it based on the names column. We use titles and profesional titles, as well as names to make these determinations. I inspected professsional titles to see if first names were all male. There is a Dr. Alice. I overwrite the case below, designating this individual as female. Also, any last names like John, Wallace and the like will be overwriten if there is a woman's title.

```
df$sex <- NA
df$sex[str_detect(df$name, "Master |Mr. |Mr |Father |Dr. |Sir |Don |Commander |Captain |Major |Colonel
df$sex[str_detect(df$name, "Miss |Mrs.|Doña |Countess |Lady |Alice")] <- "Female"
table(df$sex, as.numeric(df$age) >= 18, useNA = "ifany")
```

```
## Warning in table(df$sex, as.numeric(df$age) >= 18, useNA = "ifany"): NAs
## introduced by coercion
```

```
##
##          FALSE TRUE <NA>
##   Female    81  406    3
##   Male     110 1565   12
```

```
##   <NA>       0    2    0
```

```
table(df$survival_outcome, df$lifeboat)
```

```
##
##             ?  1 10 11 12 13 14 14? 15 15? 16  2  3  4  5  6  7  8  9  A
##   Perished  0  0  0  0  0  0  1   0  1   0  1  0  0  1  0  0  1  0  0  4
##   Survived 18 12 33 48 20 66 43   1 58   1 33 18 38 41 36 25 25 27 41 12
##
##            A/14  B  C  D
##   Perished    0  0  0  0
##   Survived    1 29 48 21
```

```
df[is.na(df$sex),] # These are probably men too - Position Trimmer and Fireman/Stoker
```

```
##              name age                         hometown     boarded
## 1500  Gosling, S.  26 Southampton, Hampshire, England Southampton
## 1529 Instance, T.  33 Southampton, Hampshire, England Southampton
##      destination lifeboat body class home_country Lifeboat Destination
## 1500        <NA>     <NA> <NA>  <NA>         <NA>       NA        <NA>
## 1529        <NA>     <NA> <NA>  <NA>         <NA>       NA        <NA>
##      Boarded survival_outcome       position        crew_type crew  sex
## 1500      NA         Perished        Trimmer Engineering crew Crew <NA>
## 1529      NA         Perished Fireman/Stoker Engineering crew Crew <NA>
```

# Age

We want to have a numeric age; the current column is a character vector and has information about months old if the individual was an infant. We save the character information in a new column, and then overwrite the infant age with 0, and then save the age variable as numeric.

```
df$age_character <- df$age
table(df$age_character, useNA = "ifany")
```

```
##
##     --      1     10 10 mo.     11 11 mo.     12     13     14     15
##      2     11      6      3      4      1      6      6      8     11
##     16     17     18     19      2  2 mo.     20     21     22     23
##     28     38     57     62     13      1     80     81     98     68
##     24     25     26     27     28     29      3     30     31     32
##     93     85     74     78     91     76      7     97     74     91
##     33     34     35     36     37     38     39      4  4 mo.     40
##     51     51     60     69     42     43     51     15      1     43
##     41     42     43     44     45     46     47     48     49      5
##     29     39     24     27     32     17     19     24     13      5
##  5 mo.     50     51     52     53     54     55     56     57     58
##      1     16     10     12      3     10      9      5      7      7
##     59      6     60     61     62     63     64     65     66     67
##      9      6      8      7      8      6      5      2      3      1
##     69      7  7 mo.     70     71     74      8      9  9 mo.    n/a
##      1      9      1      1      3      1      9      9      2      2
##   <NA>
##      1
```

```
df$age[str_detect(df$age,"m")] <- 0
df$age <- as.numeric(df$age)
```

```
## Warning: NAs introduced by coercion
```

```
table(df$age, useNA = "ifany")
```

```
##
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
##   10   11   13    7   15    5    6    9    9    9    6    4    6    6    8
##   15   16   17   18   19   20   21   22   23   24   25   26   27   28   29
##   11   28   38   57   62   80   81   98   68   93   85   74   78   91   76
##   30   31   32   33   34   35   36   37   38   39   40   41   42   43   44
##   97   74   91   51   51   60   69   42   43   51   43   29   39   24   27
##   45   46   47   48   49   50   51   52   53   54   55   56   57   58   59
##   32   17   19   24   13   16   10   12    3   10    9    5    7    7    9
##   60   61   62   63   64   65   66   67   69   70   71   74 <NA>
##    8    7    8    6    5    2    3    1    1    1    3    1    5
```

## Assistant

Some passangers, especially first class, travel with household assistents. We pull this info the name variable, as the pattern is "and" and then a description of the role. The code depends on the people being in the original order of the wikipedia chart.

```
df = df %>%
  mutate(v = str_extract(name, "^and .+?,")) %>%
  mutate(v = str_replace(v, "and ", "")) %>%
  mutate(v = str_replace(v, ",", "")) %>%
  mutate(household_assistant = if_else(is.na(v), "Not Assistant", "Assistant")) %>%
  rename(household_assistant_type = v) %>%
  mutate(q = name) %>%
  mutate(q = ifelse(household_assistant=="Assistant", NA, q)) %>%
  fill(q) %>%
  mutate(q = ifelse(household_assistant=="Assistant", q, NA)) %>%
  rename(household_assistant_to_whom = q)

table(df$household_assistant_type)
```

```
##
##  chauffeur      clerk       cook   dragoman  governess       maid
##          3          1          1          1          2         20
## manservant      nurse  secretary      valet
##          1          3          2          7
```

```
table(df$household_assistant)
```

```
##
##     Assistant Not Assistant
##            41          2138
```

```
table(df$household_assistant, df$survival_outcome)
```

```
##
##                 Perished Survived
```

```
##   Assistant           12       29
##   Not Assistant     1460      678
```

## Save data

```
if(!dir.exists(here("data-raw", "DataProducts"))) {
  dir.create(here("data-raw", "DataProducts"))
}
str(df)
```

```
## 'data.frame':    2179 obs. of  21 variables:
##  $ name                        : chr  "Allen, Miss Elizabeth Walton" "Allison, Mr. Hudson Joshua Creig
##  $ age                         : num  29 30 19 18 25 33 2 0 22 47 ...
##  $ hometown                    : chr  "St. Louis, Missouri, US" "Montreal, Quebec, Canada" "Montreal,
##  $ boarded                     : chr  "Southampton" "Southampton" "Southampton" "Southampton" ...
##  $ destination                 : chr  "St. Louis, Missouri, US" "Montreal, Quebec, Canada" "Montreal,
##  $ lifeboat                    : chr  "2" NA NA "11" ...
##  $ body                        : chr  NA "135" "294" NA ...
##  $ class                       : chr  "First" "First" "First" "First" ...
##  $ home_country                : chr  NA NA NA NA ...
##  $ Lifeboat                    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Destination                 : chr  NA NA NA NA ...
##  $ Boarded                     : logi  NA NA NA NA NA NA ...
##  $ survival_outcome            : chr  "Survived" "Perished" "Perished" "Survived" ...
##  $ position                    : chr  NA NA NA NA ...
##  $ crew_type                   : chr  NA NA NA NA ...
##  $ crew                        : chr  "Not Crew" "Not Crew" "Not Crew" "Not Crew" ...
##  $ sex                         : chr  "Female" "Male" "Male" "Female" ...
##  $ age_character               : chr  "29" "30" "19" "18" ...
##  $ household_assistant_type    : chr  NA NA "chauffeur" "cook" ...
##  $ household_assistant         : chr  "Not Assistant" "Not Assistant" "Assistant" "Assistant" ...
##  $ household_assistant_to_whom: chr  NA NA "Allison, Mr. Hudson Joshua Creighton" "Allison, Mr. Hudso
```

```
write.csv(df, here("data-raw", "DataProducts", "PeopleOnTitanic.csv"),
          row.names = F)

# rename to final data table name and save for package use
titanic_complete <- df
devtools::use_data(titanic_complete, overwrite = TRUE)
```

```
## Saving titanic_complete as titanic_complete.rda to /Users/evangelinereynolds/Google Drive/SideProjec
```