The RMS Titanic

Final Written Exam SCR 2019

Miss Rose Dewitt Bukater

January 09, 2020

Exam Instructions

This exam was created in R version 3.6.1 (2019-07-05). The exam consists of three tasks for a total of 100 points. Behind each (sub)task you will see the number of points that can be earned. The grading model can always be modified in favor of all students.

Unless it is specified differently, during this exam you can only use functions and data from the core packages in `tidyverse`, the packages `bench` and `microbenchmark`, the package `magrittr`, the package `titanic`, or functions coming from packages that are automatically loaded in `RStudio` based on the default settings.

Furthermore, load the variables that are stored in `0_dat/Model_Answer_Variables.RData`. You will need these variables to complete the tasks and you can use these variables to check your answers.

Your style of coding affects the assignment grade. A correct answer is preferred above beautiful code, however, it may cost you points when very complicated code is provided as an answer at a place where simple `R` functions would suffice. Adhere to a consistent and neat programming style, e.g.,
+ https://google.github.io/styleguide/Rguide.xml,
+ http://style.tidyverse.org.

Change the name of the file `Lastname_ULCN.Rmd` accordingly (give it your real last name and ULCN number), and write down your answers in this file. Make sure you write your code in `R` code chunks. When you wish or need to write text to answer one of the questions, don't use `R` comments, but just write your text outside the `R` code chunk.

You are handing in a report of your answers. Thus, make sure that you are able to knit the `.Rmd` report to `.pdf` without any problems in this `Rproject`'s directory, and also make sure that we can obtain all information from the `.pdf` report to grade your answers.

Upload the `.Rmd` file withyour answers as your own clean `Lastname_ULCN.Rmd` file to Blackboard **before** 13.15 hours, on January 9, 2020.

- Go to Statistical Computing with R –> Exams & Assignments –> Exam January 09
- Unless you are granted extra time, every minute later than 13:15 hours will cost you 10 out of 100 points: when you submit your `.Rmd` file at 13:35 hours spot on, it means you already lost 50 points. Files submitted after 13:35 hours will not be graded.
- To be sure, you can also e-mail your `.Rmd` file to `rteam@stat.leidenuniv.nl`.

Last, you are allowed to consult the internet and all files on your computer, or even bring your physical archive. However, keep in mind that any form of direct interaction with another person or person(s) is considered **FRAUD**.

Success!

the R-team.

0. About the Data of the Passengers on the Titanic

There are many (open) datasets available online about the famous RMS Titanic ship. Many of these data sources are about the passengers of the voyage of the RMS Titanic where most passengers and crew members lost their lives. For example see

- https://www.kaggle.com/c/titanic
- http://math.ucdenver.edu/RTutorial/
- https://titanicfacts.net
- https://en.wikipedia.org/wiki/Passengers_of_the_RMS_Titanic#Survivors_and_victims

The data sources on these websites are not completely consistent with each other. For example, the number of Titanic passengers is

- `1309` in the data obtained from *Kaggle*;
- `1313` in the data obtained from UCD;
- `1317` in the data obtained from *https://titanicfacts.net*;
- and `1316` in the data obtained from Wikipedia.

These data sources are stored in the `Model_Answer_Variables.RData` in the variables `kaggletanic`, `ucdtanic`, `factstanic`, and `wikititab`, respectively. From these data sources, the R-team trusts the data provided by Wikipedia the most.

In this exam we are most interested in the following variables of each data set:

- `Name`: complete name of the passenger;
- `PClass`: the 1st, 2nd, or 3rd passengers class of the passenger's ticket;
- `Age`: a numeric variable giving the age of the passenger (in years);
- `Sex`: the sex of the passenger;
- `Survived`: whether the passenger got saved (= 1) or not (= 0).

1. Data Wrangling and Exploratory Data Operations (50 points)

1.1 (5 points)

Check out the following code chunk:

```
MakeKaggleTanic <- function() {
  titanic::titanic_test %>%
    mutate(Survived = NA) %>% # Line 1
    select(names(titanic::titanic_train)) %>% # Line 2
    bind_rows(titanic::titanic_train) %>% # Line 3
    mutate(In_Test_Set = is.na(Survived)) %>% # Line 4
    rename(PClass = Pclass)  %>%
    arrange(PassengerId)
}
```

You could use the function `MakeKaggleTanic()` to create a complete RMS Titanic passengers dataset out of the training and test data from the `titanic` package.

Describe what happens at each line of commented code in a maximum of two sentences per line.

1.2 Which function is faster? (5 points + 2.5 Bonus points)

Apart from `MakeKaggleTanic()`, we could also use these two other functions: `MakeKaggleTanicJoin()`, and `MakeKaggleTanicBase()`. Both functions are stored in `Model_Answer_Variables.RData` in the `0_dat` folder.

The function `MakeKaggleTanicJoin()` is a `tidyverse` based wrapper around the function `full_join()`, while `MakeKaggleTanicBase()` is just based on `rbind()` and code from the `base` package in `R`.

Compare these functions on their speed using `system.time()`, and obtain 2.5 extra bonus points by using `bench::mark()` or `microbenchmark::microbenchmark()` as well.

1.3 Wrangling `Age`, `PClass`, and `Sex` (18 points)

We have stored the data of the RMS Titanic passengers from the Titanic facts website in the variable `factstanic_prep` from the `Model_Answer_Variables.RData`. The data stored in `factstanic_prep` is nearly identical to the data in the `factstanic` variable.

Now, it is up to you in this subtask (and the next) to make sure that the following code

```
all.equal(factstanic_prep, factstanic)
```

```
## [1] "Length mismatch: comparison on first 8 components"
## [2] "Component \"Age\": Modes: character, numeric"
## [3] "Component \"Age\": target is character, current is numeric"
```

eventually evaluates to TRUE. For these two datasetsto be equal the dataset `factstanic_prep` needs two extra variables, `Sex` and `SexCatAge`, and the variable `Age` needs to be of type `integer`.

Hint: for the next tasks the functions `strsplit()`, `grep()`, `gsub()`, `as.integer()` may be of help.

1.3a Creating the `Sex` variable

The first characters of each passenger in the variable `Name` in `factstanic_prep` indicate the title of the passenger. Except for the title Dr, we can use every other title to deduct the sex of a passenger. For example, the female titles are:

```
titles_female <- c("Dona", "Lucy", "Miss", "Mlle", "Mme.", "Mrs", "Ms", "Senora")
```

(For the purpose of this exam, we will assume that 'Lucy' is a title.)

Another piece of information that we can give is that of the eight passengers who were registered with their doctorate title, only Dr Alice May Leader is female.

Use this information to create a `Sex` variable in `factstanic_prep` that is equal to the `Sex` variable in `factstanic`.

1.3b Creating the `Age` variable

The `Age` variable in `factstanic_prep` consists of

- 1304 numeric values that represent the age of each passenger in the number of years;
- 11 alphanumeric characters `paste0(Y,"m")`, where the integer `Y` represents the age of a passenger in months;
- and 2 alpha characters NK that simply stand for not known.

Without obtaining any warning message, change the variable `Age` in `factstanic_prep` into type `integer` such that it represents the number of years and is equal to the `Age` variable in `factstanic`.

1.3c Creating the variable `SexCatAge`

In the `SexCatAge` variable the value child represents a passenger younger than 13 years old, and the female and male values represent the `Sex` of the passengers of 13 years and older.

Recreate the `SexCatAge` variable for your `factstanic_prep` dataset based on its variable `Age` and `Sex`.

Hint: in case you are not sure about your previous answers on the `factstanic_prep`, you may also use the `Age` and `Sex` variables from the `factstanic` data set.

1.4 Which Data Source is Closest to Wikipedia? (7 points)

Put the `kaggletanic`, `ucdtanic`, and `factstanic` datasetsfrom the `Model_Answer_Variables.RData` inside a list.

Use a loop to apply the function `CompareToWikiTiTab()` on each dataset in the list and show that the sum of the absolute number of differences between the counts in `wikititab` and the counts in each dataset is smallest for the `factstanic` dataset.

```
CompareToWikiTiTab <- function(dat, ref = wikititab) {
  # dat = factstanic
  tab <- table(dat$Survived, dat$PClass, dat$SexCatAge)
  out <- rbind(
  child = tab[ , , 1],
  female = tab[ , , 2],
  male = tab[ , , 3]
  )
  rownames(out) <- rownames(ref)
  return(abs(out - wikititab[, -4]))
}
```

1.5 Visualizing `Survival` and `Age` (20 points)

We go further with the `factstanic` data to explore by visualization the relation between the outcome variable `Survived` and the predictor variables `Sex`, `Age`, and `PClass`.

You will need to create two types of visualizations. For any help, check out the `0_img` folder if you are looking for examples of the visualizations that you could plot.

Note that if you wish to use the `ggplot2` package, you are allowed to use any of its extension packages (e.g. the package `cowplot`).

1.5a Proportion of Saved Passengers per `Class` by `Sex`

Create a plot to show the proportion of saved passengers within each passenger class separated for males and females. Interpret the results of your plot.

1.5b Survival and the Distribution of Age

Create another type of plot in which you can obtain an idea of the distribution of Age separatly for each combination of `Survived` by `Sex` by `PClass`.

Hint: the `Survived` variable may need to become of type `factor`.

2 Three Logistic Regression Models and Cross-Validation (25)

2.1 The Three GLM's (5 points)

Take a look at the summaries of the following three models:

```
fitted_glm0 <- glm(
  formula = Survived ~ Sex + Age + PClass,
  family = binomial(link = 'logit'),
  data = factstanic
)
```

```
fitted_glm1 <- glm(
  Survived ~  Sex * Age + PClass,
  family = binomial(link = 'logit'),
  data = factstanic
)
```

```
fitted_glm2 <-  glm(
    formula = Survived ~  Sex + Age + Sex * PClass,
    family = binomial(link = 'logit'),
    data = factstanic
)
```

While taking a look at the summary of `fitted_glm0`, `fitted_glm1`, and `fitted_glm2`, explain which of these three models you would prefer for the purpose of estimation.

2.2 Predict the Probability of Survival (5 points + Spoiler Alert?)

"Mr Jack Dawson was born near Chippewa Falls, Wisconsin in 1892, and boarded the RMS Titanic in on April 10, 1912. He was a poor third-class artist and was able to board the ship only after winning tickets in a lucky game of poker against two Swedish men with tickets."

Could you predict for each of the models in **2.1** the probability of Mr Jack Dawson getting saved?

You are **not** allowed to use the `predict()` function to predict Mr Jack Dawson's probabilities. Instead use the logistic function (e.g. `plogis()` or code it yourself) and the following predictor values for each model:

```
x_mod0 <- c(1, Sexmale = 1, Age = 20, PClass2nd = 0, PClass3rd = 1)
x_mod1 <- c(x_mod0, `SexMale:Age` = 20)
x_mod2 <- c(x_mod0, `SexMale:PClass2nd` = 0, `SexMale:PClass3rd` = 1)
```

Hint: take a quick look at Wikipedia for the definition of the logistic function in logistic regression.

2.3 Cross-validation (15 points)

Use a cross-validation procedure of your own choice to validate the models `fitted_glm0`, `fitted_glm1`, or `fitted_glm2` for prediction purposes.

Which model would you prefer? Add a critical comment on your cross-validation procedure and its relation to some of the previous subtasks in this exam.

Hint: Now, we strongly recommend to use the `predict()` *function with argument* `type = 'response'`.

3 Ticket Prices and the Theil-Sen estimator (25 points)

Through a dubious source, the R-Team has obtained data on the ticket prices charged for transportation and lodging on the RMS Titanic. Our informant claims that the ticket prices can be predicted from the precise age of the passengers, a variable that is present in the `tibble` dataset `titanic_df` (see `Model_Answer_Variables.RData`).

3.1 Simple linear regression (5 points)

Regress `ticket_price` (Y) on `Age` (X) by using a simple linear regression model via `lm()` and the `titanic_df tibble`.

Use your fitted model to create your own plot of the regression line and the data points. Comment on your results that you can see in the plot.

Note: you will have to use your plot in the next subtask again.

3.2 Compute and Visualize the Theil-Sen estimator (12 points)

Given that we only have one predictor, and assuming that our dataset contains outliers, we prefer to use the Theil-Sen estimator of the intercept and slope for our model of interest, because the Theil-Sen estimator is known for its robustness against outliers. The Theil-Sen estimators of the slope and of the intercept are defined as

$$\hat{\beta} = \text{median}\left(\text{slope}\{i,j\}\right), \text{ and}$$
$$\hat{\alpha} = \text{median}\left(Y - \hat{\beta}\right)$$

where slope$\{i,j\}$ are the collections of slopes of the lines going through all possible pairs of points:

$$\{i,j\} \subseteq \{1,\ldots,n\},$$

where $i \neq j$ and n denotes the size of the data set. Thus, there are $\binom{n}{2}$ possible pairs of points.

Write and evaluate your own function that takes at least one argument, the dataset, and returns the Theil-Sen estimate of slope and intercept for the above-mentioned regression problem. Then, return to the plot of the regression line you created above, and add the Theil-Sen regression line. Comment on the differences.

3.3 Empirical Bootstrap and the Percentile Method: 95% confidence (8 points)

The Theil-Sen estimator is (as you may have noticed) non-parametric and it does not come with any "default ways" of estimating the variability of our coefficients. To obtain an idea about the variablility, we could estimate a 95% confidence interval of the Theil-Sen slope by using the empirical bootstap and the percentile method for the confidence interval.

The most blunt way to apply the empirical bootstrap procedure, is to resample each data point $\{X_i, Y_i\}$ to re-create your replicate dataset on which you apply your function to compute your Theil-Sen estimator. Note however, the results would be (approximately) the same if you would resample from all the `choose(nrow(titanic_df), 2)` slopes of each object pair, and it is also faster.

Obtain your 95% confidence interval based on the percentile method by creating at least `B=100` bootstrapped Theil-Sen slopes that are based on resampling form all all the `choose(nrow(titanic_df), 2)` slopes.

*Hint: if you did not succeed in **3.2**, use the model answers variable `slopes_of_all_pairs` for this specific subtask.*