Final Assessment: Comprehensive Assessment: Puerto Rico Hurricane Mortality

Hector Antigua

Last compiled on March, 2022

**Project introduction**

On September 20, 2017, Hurricane María made landfall on Puerto Rico. It was the worst natural disaster on record in Puerto Rico and the deadliest Atlantic hurricane since 2004. However, Puerto Rico’s official death statistics only tallied 64 deaths caused directly by the hurricane (due to structural collapse, debris, floods and drownings), an undercount that slowed disaster recovery funding. The majority of the deaths resulted from infrastructure damage that made it difficult to access resources like clean food, water, power, healthcare and communications in the months after the disaster, and although these deaths were due to effects of the hurricane, they were not initially counted.

In order to correct the misconception that few lives were lost in Hurricane María, statisticians analyzed how death rates in Puerto Rico changed after the hurricane and estimated the excess number of deaths likely caused by the storm. [This analysis](https://drive.google.com/file/d/16X9qtnPaD--2dPhpcwu7S53esafH59i9/preview) suggested that the actual number of deaths in Puerto Rico was 2,975 (95% CI: 2,658-3,290) over the 4 months following the hurricane, much higher than the original count.

We will use your new data wrangling skills to extract actual daily mortality data from Puerto Rico and investigate whether the Hurricane María had an immediate effect on daily mortality compared to unaffected days in September 2015-2017.

Note: This project is only available for verified learners. If you have not paid for a certificate in this course, you will be offered an opportunity to upgrade instead of seeing the problems.

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.4 v purrr 0.3.4  
## v tibble 3.1.2 v dplyr 1.0.6  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(pdftools)

## Warning: package 'pdftools' was built under R version 4.1.2

## Using poppler version 21.04.0

options(digits = 3) # report 3 significant digits

## Puerto Rico Hurricane Mortality: Part 1

### **Question 1**

In the extdata directory of the **dslabs** package, you will find a PDF file containing daily mortality data for Puerto Rico from Jan 1, 2015 to May 31, 2018. You can find the file like this:

fn <- system.file("extdata", "RD-Mortality-Report\_2015-18-180531.pdf", package="dslabs")

Find and open the file or open it directly from RStudio. On a Mac, you can type: and on Windows, you can type:

#system2("open", args = fn) for MAC  
system("cmd.exe", input = paste("start", fn))

## [1] 0

Which of the following best describes this file?

**Answer:** It is a report combining graphs and tables. Extracting the data seems possible.

### **Question 2**

We are going to create a tidy dataset with each row representing one observation. The variables in this dataset will be year, month, day and deaths.

Use the **pdftools** package to read in fn using the pdf\_text() function. Store the results in an object called txt.

Describe what you see in txt.

**Answer:** A character string of length 12. Each entry represents the text in each page. The mortality data is in there somewhere.

txt <- pdf\_text(fn)

### **Question 3**

Extract the ninth page of the PDF file from the object txt, then use the str\_split() function from the stringr package so that you have each line in a different entry. The new line character is \n. Call this string vector x.

Look at x. What best describes what you see?

x <- txt[[9]] %>% str\_split(pattern = "\\n")

**Answer:** I can see the table! But there is a bunch of other stuff we need to get rid of.

What kind of object is x? How many entries does x have? **Answer:**

class(x)

## [1] "list"

length(x)

## [1] 1

### **Question 4**

Define s to be the first entry of the x object.

What kind of object is s? How many entries does s have? **Answer:**

s <- x[[1]]  
class(s)

## [1] "character"

length(s)

## [1] 41

### **Question 5**

When inspecting the string we obtained above, we see a common problem: white space before and after the other characters. Trimming is a common first step in string processing. These extra spaces will eventually make splitting the strings hard so we start by removing them.

We learned about the command str\_trim() that removes spaces at the start or end of the strings. Use this function to trim s and assign the result to s again.

After trimming, what single character is the last character of element 1 of s? **Answer:**

s <- str\_trim(s)  
s[[1]]

## [1] "6/4/2018 Departamento de Salud - Registro Demográfico - División de Calidad y Estadísticas Vitales"

### **Question 6**

We want to extract the numbers from the strings stored in s. However, there are a lot of non-numeric characters that will get in the way. We can remove these, but before doing this we want to preserve the string with the column header, which includes the month abbreviation.

Use the str\_which() function to find the row with the header. Save this result to header\_index. Hint: find the first string that matches the pattern “2015” using the str\_which() function.

What is the value of header\_index? **Answer:**

header\_index <- s %>% str\_which("2015") %>% first()  
header\_index

## [1] 3

### **Question 7**

We want to extract two objects from the header row: month will store the month and header will store the column names.

Save the content of the header row into an object called header, then use str\_split() to help define the two objects we need.

What is the value of month? **Answer:**

month <- s[[header\_index]] %>% str\_split(pattern = "\\s+", n = 2) %>% .[[1]] %>% .[1]  
header <- s[[header\_index]] %>% str\_split(pattern = "\\s+", n = 2) %>% .[[1]] %>% .[2]

## Puerto Rico Hurricane Mortality: Part 2

This assessment continues from the previous page and assumes that you have defined all of the variables from those questions. In particular, make sure s is defined as in the previous exercises.

### **Question 8**

Notice that towards the end of the page defined by s you see a “Total” row followed by rows with other summary statistics. Create an object called tail\_index with the index of the “Total” entry.

What is the value of tail\_index? **Answer:**

tail\_index <- s %>% str\_which("Total")   
tail\_index

## [1] 36

### **Question 9**

Because our PDF page includes graphs with numbers, some of our rows have just one number (from the y-axis of the plot). Use the str\_count() function to create an object n with the count of numbers in each row.

How many rows have a single number in them? **Answer:**

n <- s %>% str\_count(pattern = "\\d+") %>% str\_which("1") %>% length()  
n

## [1] 2

### **Question 10**

We are now ready to remove entries from rows that we know we don’t need. The entry header\_index and everything before it should be removed. Entries for which n is 1 should also be removed, and the entry tail\_index and everything that comes after it should be removed as well.

How many entries remain in s? **Answer:**

s[(header\_index+1):tail\_index] %>% str\_count(pattern = "\\d+") %>% str\_which("1")

## [1] 4 7

s <- s[(header\_index+1):(tail\_index-1)][c(-4,-7)]  
length(s)

## [1] 30

### **Question 11**

Now we are ready to remove all text that is not a digit or space. Do this using regular expressions (regex) and the str\_remove\_all() function.

In regex, using the ^ inside the square brackets [] means not, like the ! means not in !=. To define the regex pattern to catch all non-numbers, you can type [^\\d]. But remember you also want to keep spaces.

Which of these commands produces the correct output?

**Answer:**

s <- str\_remove\_all(s, "[^\\d\\s]")

### **Question 12**

Use the str\_split\_fixed function to convert s into a data matrix with just the day and death count data:

s <- str\_split\_fixed(s, "\\s+", n = 6)[,1:5]

Now you are almost ready to finish. Add column names to the matrix: the first column should be day and the next columns should be the header. Convert all values to numeric. Also, add a column with the month. Call the resulting object tab. **Answer:**

cnames <- c("day",str\_split\_fixed(header, "\\s+", n = 4))  
colnames(s) <- cnames  
tab <- s %>% as\_tibble() %>% mutate\_all(~as.numeric(as.character(.))) %>% mutate(month=month)

What was the mean number of deaths per day in September 2015? What was the mean number of deaths per day in September 2016? Hurricane María hit Puerto Rico on September 20, 2017. What was the mean number of deaths per day from September 1-19, 2017, before the hurricane hit? What was the mean number of deaths per day from September 20-30, 2017, after the hurricane hit? **Answer:**

tab[2] %>% colMeans(.)

## 2015   
## 75.3

tab[3] %>% colMeans(.)

## 2016   
## 78.9

tab[1:19,4] %>% colMeans(.)

## 2017   
## 83.7

tab[20:30,4] %>% colMeans(.)

## 2017   
## 122

### **Question 13**

Finish it up by changing tab to a tidy format, starting from this code outline:

tab\_TB <- tab %>% gather(year, deaths, -day) %>%  
 mutate(deaths = as.numeric(deaths))

## Warning in mask$eval\_all\_mutate(quo): NAs introduced by coercion

tab\_TB

## # A tibble: 150 x 3  
## day year deaths  
## <dbl> <chr> <dbl>  
## 1 1 2015 75  
## 2 2 2015 77  
## 3 3 2015 67  
## 4 4 2015 71  
## 5 5 2015 62  
## 6 6 2015 77  
## 7 7 2015 85  
## 8 8 2015 84  
## 9 9 2015 79  
## 10 10 2015 66  
## # ... with 140 more rows

**Answer:** gather

### **Question 14**

Make a plot of deaths versus day with color to denote year. Exclude 2018 since we have no data. Add a vertical line at day 20, the day that Hurricane María hit in 2017.

Which of the following are TRUE?

**Answer:** September 2015 and 2016 deaths by day are roughly equal to each other. After the hurricane in September 2017, there were over 100 deaths per day every day for the rest of the month. No days before September 20, 2017 have over 100 deaths per day.

ggplot(tab\_TB[1:90,]) + geom\_line(aes(day,deaths,color=year)) + geom\_vline(xintercept = 20)

