QTM 151 Final Project

Christine Zhou, Echo Sui, Ruby Wu

May 7 2021

Instructions

The class project is split into two parts. This is Part II - the Project. You've already completed Part I - the Proposal.

The due date is May 9th at 6:00pm U.S. ET. This is a **firm, unmoveable deadline** so I have enough time to grade these before grades are due. Please prepare accordingly.

For the project you will work in groups of 3-4 to prepare a brief RMarkdown report using a data set of your choice to answer **at least one research question(s)** you're interested in investigating. The Canvas Project module has an example of a prior project to help guide what would be appropriate research question(s) vs. something that's too broad or too narrow (though note I think it could use better headings, a bit more organization, and your project is more focused on data cleaning), as well as several possible sources for datasets.

The Project must include and will be graded as follows (20 total pts):

1. Make sure to load any packages you may need right at the start. (0 pts)
2. Ensure that no chunks have the `include = FALSE` or `echo = FALSE` option, as I want to be able to see *all* your code and output. (0 pts)
3. Brief but descriptive headings and document organization (answers under headings, text near relevant code, brief explanatory text as necessary to walk the reader through what you did, etc.), as well as **brief bits of narrative text throughout the document to walk me through your cleaning and analytical process**. (5 pts)

   Look to my HW1 and RMarkdown Organization examples from QTM 150 as well as my feedback throughout both courses for how to write good headings, organize your assignment, and how much narrative text (outside of code chunks) I want. A good rule of thumb for narrative text is: explain what *you're* doing, NOT what *your code* is doing. I want to be able to easily track what you're doing as you move through the analysis, and why. I do *not* need to see a repeat, line-by-line narration of what your code does.

   You can use code comments where necessary (complex code lines or blocks every few lines, or at least every chunk) for additional detail. These can not only help the reader but also help future-you when you go back to look at some code you wrote and are trying to figure out what it does and why you even wrote it in the first place.

   Make sure you also provide a correct written answer to your research question(s) and interpretations of all plots and tables included in your project.

   I'm really emphasizing this in the final project because I want you to be able to show this project off to future potential data science employers as part of a "portfolio." Being able not to just promise things, but to *show* your skills is incredibly helpful in getting a job. A *clean, well-organized* analytical document will go a long way in interviews, I assure you.

4. A description and *basic* exploration of the dataset. Hopefully you mostly already did this in the Proposal. (1 pt)

5. At least one research question, clearly articulated. (1 pts)

6. You should have chosen a dataset and question(s) that, before you can answer it/them, requires quite a bit of "real world style" data cleaning. You must demonstrate your skills in **5** of the following: (10 pts)

   i) Data reshaping (changing the number of columns or rows by pivoting)
   ii) Splitting or combining values across columns (separating and uniting)
   iii) Cleaning variable names
   iv) Identifying and cleaning missing observations or values or `NA` and `NaN` and `NULL` values
   v) Re-coding variable values (for example, changing a state abbreviation to the full state name; or changing a "1" for a race variable to "White")
   vi) Cleaning strings (anything from the `stringr` package; if it requires regular expressions, this counts for **2**)
   vii) Cleaning factors (anything from the `forcats` package)
   viii) Cleaning or modifying dates (anything from the `lubridate` package)

7. Once your data is clean, answer your research question(s) with at least 2 visualizations of different types. (3 pts)

   What's a "type?" Different geoms or combinations of geoms.

   Adhere to the layout and graphic design guidance you've received throughout this course and in feedback on your assignments. Everything on the charts should be human-readable, easily separated or laid out to allow the answering of your research questions with a quick visual scan, a good legend position, and there should be no extraneous elements.

   At least one of your visualizations must be of an advanced type we learned in QTM 151 (map or interactive `plotly` plot).

   Make sure to describe and interpret these *briefly* in nearby text.

Length-wise, aim for something a bit longer than the example project (data cleaning often takes up a lot of space).

To submit this assignment:

Ideally, knit straight to PDF by changing `html_document` to `pdf_document` in line 5 above. Otherwise:

1. Knit to HTML. An HTML document should open automatically in another RStudio window.
2. Click "Open in Browser" in that HTML document. It should open as a webpage in your default browser (e.g. Chrome).
3. Click Ctrl+P/Command+P, but instead of printing a hard copy on your printer click "Save as PDF."
4. Save and upload that document to Canvas.

——BEGIN ANSWER BELOW——

Avengers Assemble!

This project is an exploration of the dataset regarding Marvel's Avengers (`Avengers.csv` provided by `fivethirtyeight`). We will explore topics such as how the numbers of appearances and deaths of the Avengers are associated with gender, honorary status and number of years they have joined.

Load packages and import the dataset

The packages `tidyverse`, `skimr`, etc. are loaded to facilitate later data analysis and visualizations.

```
pacman::p_load(tidyverse, skimr, janitor, pander,lubridate,plotly)
avengers <- read_csv("./avengers.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_character(),
##   Appearances = col_double(),
##   Year = col_double(),
##   `Years since joining` = col_double()
## )
## i Use `spec()` for the full column specifications.
```

The csv file `avengers.csv` is obtained from this link. It contains 173 observations of 21 variables, describing the Avengers from the Marvel Comics up until April 30, 2015.

Explore the dataset

Before raising our research question and getting into data cleaning, first we want to take a look at the current structure and variables of the dataset.

```
head(avengers, 10)
```

```
## # A tibble: 10 x 21
##    URL            `Name/Alias`    Appearances `Current?` Gender `Probationary I~
##    <chr>          <chr>                 <dbl> <chr>      <chr>  <chr>
##  1 http://marvel~ "Henry Jonatha~        1269 YES        MALE   <NA>
##  2 http://marvel~ "Janet van Dyn~        1165 YES        FEMALE <NA>
##  3 http://marvel~ "Anthony Edwar~        3068 YES        MALE   <NA>
##  4 http://marvel~ "Robert Bruce ~        2089 YES        MALE   <NA>
##  5 http://marvel~ "Thor Odinson"         2402 YES        MALE   <NA>
##  6 http://marvel~ "Richard Milho~         612 YES        MALE   <NA>
##  7 http://marvel~ "Steven Rogers"        3458 YES        MALE   <NA>
##  8 http://marvel~ "Clinton Franc~        1456 YES        MALE   <NA>
##  9 http://marvel~ "Pietro Maximo~         769 YES        MALE   <NA>
## 10 http://marvel~ "Wanda Maximof~        1214 YES        FEMALE <NA>
## # ... with 15 more variables: Full/Reserve Avengers Intro <chr>, Year <dbl>,
## #   Years since joining <dbl>, Honorary <chr>, Death1 <chr>, Return1 <chr>,
## #   Death2 <chr>, Return2 <chr>, Death3 <chr>, Return3 <chr>, Death4 <chr>,
## #   Return4 <chr>, Death5 <chr>, Return5 <chr>, Notes <chr>
```

From the first 10 rows of the data frame, we can see that each observation in the dataset represents an Avenger from the comic books. The variables describing an observation include the name (or alias), gender, number of comic books that character appeared in as of April 30, 2015, and most importantly, the number of deaths and returns the character experienced from their first appearance to April 30, 2015.

Next, we want to figure out each variable's data type, and have a brief idea about the locations of the missing values in the dataset.

```
skim(avengers)
```

Table 1: Data summary

Name	avengers
Number of rows	173
Number of columns	21
Column type frequency:	
character	18
numeric	3
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
URL	0	1.00	36	67	0	173	0
Name/Alias	10	0.94	4	35	0	162	0
Current?	0	1.00	2	3	0	2	0
Gender	0	1.00	4	6	0	2	0
Probationary Introl	158	0.09	6	6	0	12	0
Full/Reserve Avengers Intro	14	0.92	5	6	0	93	0
Honorary	0	1.00	4	12	0	4	0
Death1	0	1.00	2	3	0	2	0
Return1	104	0.40	2	3	0	2	0
Death2	156	0.10	2	3	0	2	0
Return2	157	0.09	2	3	0	2	0
Death3	171	0.01	3	3	0	1	0
Return3	171	0.01	2	3	0	2	0
Death4	172	0.01	3	3	0	1	0
Return4	172	0.01	3	3	0	1	0
Death5	172	0.01	3	3	0	1	0
Return5	172	0.01	3	3	0	1	0
Notes	98	0.43	21	255	0	71	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Appearances	0	1	414.05	677.99	2	58	132	491	4333	
Year	0	1	1988.45	30.37	1900	1979	1996	2010	2015	
Years since joining	0	1	26.55	30.37	0	5	19	36	115	

From `skim()`, we notice that except three numerical variables `Appearances`, `Year`, and `Years since joining`, all the other variables are in character type.

Also, it shows that most of the missing values are allocated at the variables describing the number of deaths/returns of the Avenger, and also the variable `Probationary Introl`. This makes sense because not

all Avengers experienced many deaths, recoveries, or were given probationary status. Some names of the Avengers are missing, but we can fill in those values since the names of Avengers are included in their URLs.

Meanwhile,from the max, min, mean, and quartiles of the three numeric variables, it seems like there is no obvious implausible values.

```
get_dupes(avengers)
```

```
## No variable names specified - using all columns.
```

```
## No duplicate combinations found of: URL, Name/Alias, Appearances, Current?, Gender, Probationary Int
```

```
## # A tibble: 0 x 22
## # ... with 22 variables: URL <chr>, Name/Alias <chr>, Appearances <dbl>,
## #   Current? <chr>, Gender <chr>, Probationary Introl <chr>,
## #   Full/Reserve Avengers Intro <chr>, Year <dbl>, Years since joining <dbl>,
## #   Honorary <chr>, Death1 <chr>, Return1 <chr>, Death2 <chr>, Return2 <chr>,
## #   Death3 <chr>, Return3 <chr>, Death4 <chr>, Return4 <chr>, Death5 <chr>,
## #   Return5 <chr>, Notes <chr>, dupe_count <int>
```

Lastly, `get_dupes()` shows that there doesn't seem to be duplicated observations in the `avengers` dataset.

Data Dictionary

To explain the variables more comprehensively, we include a data dictionary of the `avengers` dataset:

Variable	Definition
URL	The URL of the comic character on the Marvel Wikia
Name/Alias	The full name or alias of the character
Appearances	The number of comic books that character appeared in as of April 30, 2015
Current?	Is the member currently active on an avengers affiliated team?
Gender	The recorded gender of the character
Probationary Introl	Sometimes the character was given probationary status as an Avenger, this is the date that happened
Full/Reserve Avengers Intro	The month and year the character was introduced as a full or reserve member of the Avengers
Year	The year the character was introduced as a full or reserve member of the Avengers
Years since joining	2015 minus the year
Honorary	The status of the avenger, if they were given "Honorary" Avenger status, if they are simply in the "Academy," or "Full" otherwise
Death1	Yes if the Avenger died, No if not.
Return1	Yes if the Avenger returned from their first death, No if they did not, blank if not applicable
Death2	Yes if the Avenger died a second time after their revival, No if they did not, blank if not applicable
Return2	Yes if the Avenger returned from their second death, No if they did not, blank if not applicable
Death3	Yes if the Avenger died a third time after their second revival, No if they did not, blank if not applicable

Variable	Definition
Return3	Yes if the Avenger returned from their third death, No if they did not, blank if not applicable
Death4	Yes if the Avenger died a fourth time after their third revival, No if they did not, blank if not applicable
Return4	Yes if the Avenger returned from their fourth death, No if they did not, blank if not applicable
Death5	Yes if the Avenger died a fifth time after their fourth revival, No if they did not, blank if not applicable
Return5	Yes if the Avenger returned from their fifth death, No if they did not, blank if not applicable
Notes	Descriptions of deaths and resurrections.

Research Questions

Based on the dataset, here are some research questions we want to investigate:

1. Is a male Avenger more likely to appear more frequently than a female Avenger? How about the appearance frequency for different Honorary levels?
2. Are the number of deaths of Avengers related to the years since they've been introduced?

Data Cleaning

However, before we step into problem solving and data visualization, some clean steps need to be done to the original dataset to create a "tidy" data frame for us to work on.

```
avengers_org<-avengers #save a copy of the original dataset

avengers<-avengers %>%
  clean_names() %>%    #clean variable names
  mutate(gender=as.factor(gender),  #change gender to factor type (unordered)
         honorary=factor(honorary,  #change honorary to factor type (ordered)
                         ordered = TRUE,
                        levels = c("Full","Honorary","Academy","Probationary")),

         month=as.character(str_extract_all(full_reserve_avengers_intro,"[A-Za-z]+")),
         month=match(month,month.abb),
         #split full_reserve_avengers_intro to single out the month part and
         #discard the 2 digit year value

         joining_time = case_when(is.na(month) ~ as.character(year),
                                  TRUE ~paste(year,month,sep="-")),
         #rejoin month with a complete, 4 digit year value

         #Identifying missing names from character URLs
         name_alias=case_when(is.na(name_alias) ~
         str_match(url,"http\\:\\/\\/marvel\\.wikia\\.com\\/([^\\()]+)")[,2] %>%
                             str_replace_all("_$","") %>%
                             str_replace_all("_"," "),
                           TRUE ~ name_alias),
```

```
        #summarize columns deaths1-death5, return1-return 5 to 2 columns
        across(death1:return5, ~case_when(.x == "YES" ~ TRUE,
                                          TRUE ~ FALSE)),
        total_death = death1+death2+death3+death4+death5,
        total_return = return1+return2+return3+return4+return5,

        #change current to boolean type
        current=case_when(current=="YES"~TRUE,
                  TRUE~FALSE))
```

```
## Warning in FUN(X[[i]], ...): strings not representable in native encoding will
## be translated to UTF-8

## Warning in FUN(X[[i]], ...): unable to translate '<U+00C4>' to native encoding

## Warning in FUN(X[[i]], ...): unable to translate '<U+00D6>' to native encoding

## Warning in FUN(X[[i]], ...): unable to translate '<U+00E4>' to native encoding

## Warning in FUN(X[[i]], ...): unable to translate '<U+00F6>' to native encoding

## Warning in FUN(X[[i]], ...): unable to translate '<U+00DF>' to native encoding

## Warning in FUN(X[[i]], ...): unable to translate '<U+00C6>' to native encoding

## Warning in FUN(X[[i]], ...): unable to translate '<U+00E6>' to native encoding

## Warning in FUN(X[[i]], ...): unable to translate '<U+00D8>' to native encoding

## Warning in FUN(X[[i]], ...): unable to translate '<U+00F8>' to native encoding

## Warning in FUN(X[[i]], ...): unable to translate '<U+00C5>' to native encoding

## Warning in FUN(X[[i]], ...): unable to translate '<U+00E5>' to native encoding
```

```
avengers<-avengers %>%
  select(name_alias,appearances,current,gender,joining_time,honorary,total_death
         ,total_return,years_since_joining,year)%>%  #select variables needed
  rename(name=name_alias)

skim(avengers)
```

Table 5: Data summary

Name	avengers
Number of rows	173
Number of columns	10

Column type frequency:	
character	2
factor	2
logical	1
numeric	5
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
name	0	1	4	35	0	172	0
joining_time	0	1	4	7	0	94	0

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
gender	0	1	FALSE	2	MAL: 115, FEM: 58
honorary	0	1	TRUE	4	Ful: 138, Aca: 17, Hon: 16, Pro: 2

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
current	0	1	0.47	FAL: 91, TRU: 82

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
appearances	0	1	414.05	677.99	2	58	132	491	4333	
total_death	0	1	0.51	0.75	0	0	0	1	5	
total_return	0	1	0.33	0.65	0	0	0	1	5	
years_since_joining	0	1	26.55	30.37	0	5	19	36	115	
year	0	1	1988.45	30.37	1900	1979	1996	2010	2015	

```
head(avengers)
```

```
## # A tibble: 6 x 10
##   name               appearances current gender joining_time honorary total_death
##   <chr>                    <dbl> <lgl>   <fct>  <chr>        <ord>          <int>
## 1 "Henry Jonathan ~         1269 TRUE    MALE   1963-9       Full               1
## 2 "Janet van Dyne"          1165 TRUE    FEMALE 1963-9       Full               1
## 3 "Anthony Edward ~         3068 TRUE    MALE   1963-9       Full               1
## 4 "Robert Bruce Ba~         2089 TRUE    MALE   1963-9       Full               1
## 5 "Thor Odinson"            2402 TRUE    MALE   1963-9       Full               2
```

```
## 6 "Richard Milhous~          612 TRUE    MALE   1963-9       Honorary           0
## # ... with 3 more variables: total_return <int>, years_since_joining <dbl>,
## #   year <dbl>
```

In order to analyze the data, we do the following data cleaning:

1. Use `clean_names( )` to change all variable names to snack_case.
2. Change `gender` and `honorary` from character type to factor type. `gender` is an unordered factor, and `honorary` is an ordered factor with levels "Full," "Honorary," "Academy," and then "Probationary."
3. Use regular expression to extract `month` from `full_reserve_avengers_intro,` change the month value from abbreviation to numeric type, and then combined with `year.` For 14 observations that have a joining year but not a joining month, we copy `year` to `joining_time`.
4. For the observations that have a NA value for `name_alies,` we use regular expression to extract the name part from the `url` and fill that into `name_alies.`
5. Variable `death1-5`, `return1-5` and `current` has character value "YES" and missing value NA. We convert 'YES' to TRUE and NA to FALSE.
6. Compute the `total_death` by plus death1 to death5. Same for `total_return`.
7. Select the variable we might use in research questions and do the final variable name cleaning.

After the data cleaning, the data contains 173 observations of 10 variables. There are 2 character type (name, joining_time), 2 factor type (gender, honorary), 1 logical type (current) and 5 numeric type (appearances, total_death, total_return, years_since_joining, year). The data contains no missing value, and the only unusual data appears in `joining_time`. There are 14 observations with `joining_time` of 1900 and don't have a joining month, while all other avengers joined after 1963. Since Marvel's first comic book was published in 1939, we infer that the joining year for these 14 Avengers might be missing, and so `fivethirtyeight` decided to fill in these impossible years instead.

Question 1: Appearance Frequency Across Genders and Honorary Statuses

Based on the dataset, we are interested in knowing if the gender and honorary status of an avenger have any relationship with their appearance frequency in the comics.

To achieve that, we first create the `appearance_frequency` column by calculating `appearance/years_since_joining` and applying `mutate`. Before the visualization, let's first look over the numerical values.

```
freq_difference <- avengers%>%
  filter(years_since_joining != 0)%>%
  #removes observations with years_since_joining = 0 to
  #avoid dividing by 0 in the next step
  mutate(appearance_frequency = appearances / years_since_joining)%>%
  group_by(gender, honorary)%>%
  summarise(mean_freq = mean(appearance_frequency))%>%
  ungroup(gender, honorary)%>% #ungroup to add row
  #extra row added manually since there is no probationary state female avenger
  add_row(gender = "FEMALE", honorary = "Probationary", mean_freq = 0, .before = 4) %>%
  mutate(honorary=factor(honorary,   #change honorary to factor again
                         ordered=TRUE,
                         levels =c("Full","Honorary","Academy","Probationary")))
```

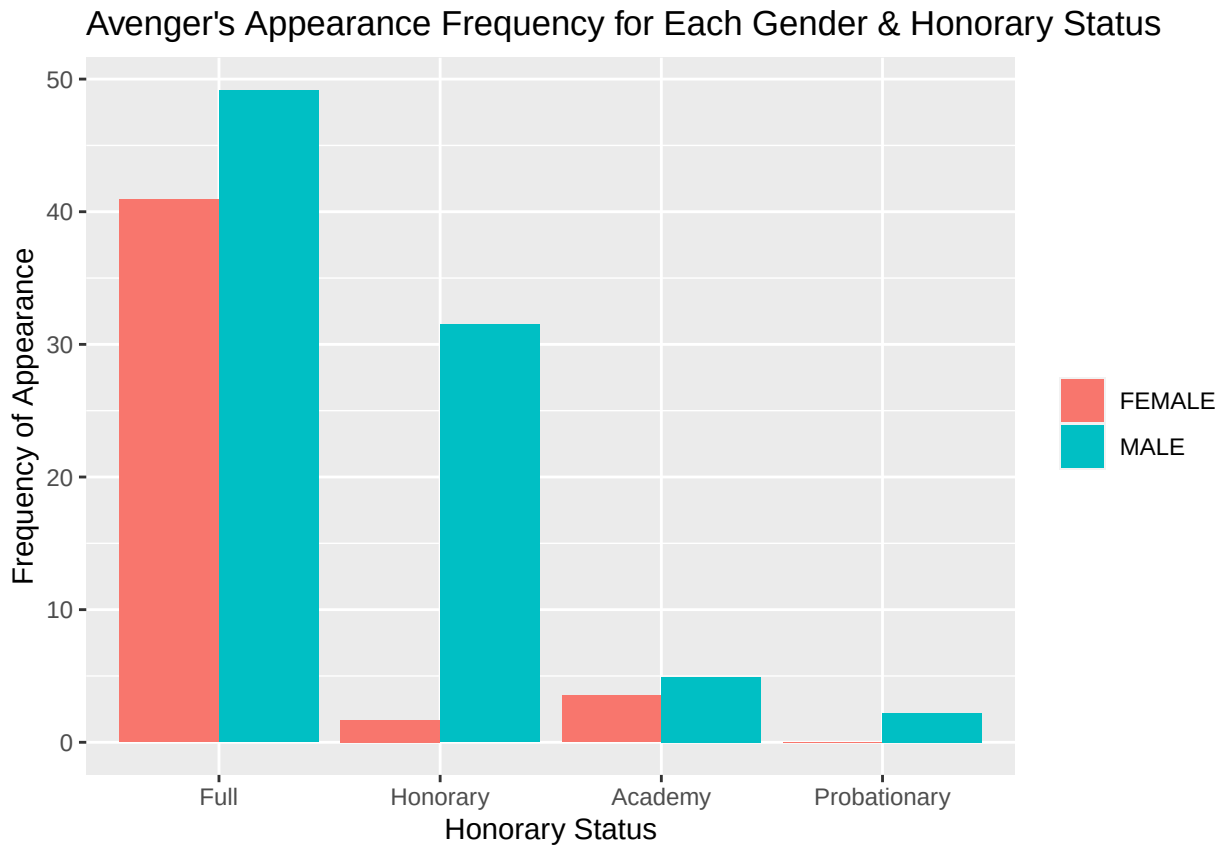
```
## `summarise()` has grouped output by 'gender'. You can override using the `.groups` argument.
```

```
freq_difference
```

```
## # A tibble: 8 x 3
##   gender honorary     mean_freq
##   <chr>  <ord>            <dbl>
## 1 FEMALE Full             40.9
## 2 FEMALE Honorary          1.67
## 3 FEMALE Academy           3.51
## 4 FEMALE Probationary      0
## 5 MALE   Full             49.2
## 6 MALE   Honorary         31.5
## 7 MALE   Academy           4.93
## 8 MALE   Probationary      2.23
```

Then a bar chart is plotted to display the frequency of appearance for Avengers of all possible combinations of genders and honorary statuses.

```
freq_difference%>%
  ggplot()+
  geom_bar(mapping = aes(x = honorary, y = mean_freq, fill = gender),
           stat = "identity",
           position = "dodge")+
  labs(title = "Avenger's Appearance Frequency for Each Gender & Honorary Status",
       x = "Honorary Status",
       y = "Frequency of Appearance")+
  theme(legend.position = "right",
        legend.title = element_blank())
```



The result reflects that within all honorary statuses, on average, male Avengers appear more frequently that female avengers. This gap is especially significant for Avengers in the Honorary status, with a appearance frequency difference of almost 30. The bar for female Avengers with Probationary status is empty because there is no female Avenger with such honorary status.

The appearance frequencies for different honorary status also have clear patterns. Avengers with Full and Honorary statuses appear much more frequent than those with Academy and Probationary statuses.

Question 2: Avengers vs Death

A special set of magical variables in this dataset includes information about the number of deaths and returns experienced by each Avenger. As a result, we are interested about if the number of deaths of Avengers are related to the years since they've been introduced.

First, we create a table to see the distribution of deaths among Avengers.

```
avengers %>%
  select(total_death, year) %>%
  group_by(total_death) %>%
  summarise(count = n())
```

```
## # A tibble: 5 x 2
##   total_death count
##         <int> <int>
## 1           0   104
## 2           1    53
```

```
## 3           2    14
## 4           3     1
## 5           5     1
```

Clearly (and luckily), most Avengers have died zero times. The number of Avengers decreases as the number of death increments, which is similar to what we would have expected.

We then create an interactive plot, where the readers can hover over the points to see which Avenger the data pertains to, if they are interested.

```
avengers %>%
  filter(years_since_joining < 115)%>%#remove outliers with joining year of 1900
  plot_ly(x = ~ years_since_joining,
          y = ~ jitter(total_death),  #jitter to avoid points overlapping
          color = ~ total_death,
          colors = "Set2",
          text = ~ paste0('</br>Name/Alias: ', name,  #add hover text
                          '</br>Time Joined: ', joining_time,
                          '</br>Number of Deaths: ', total_death),
        hoverinfo = "text") %>%
  add_markers() %>%
  layout(title = "Number of Deaths vs Number of Years Joined",
         xaxis = list(title = "Number of Years Joined"),
         yaxis = list(title = "Number of Deaths Since Joining")) %>%
  hide_colorbar()
```

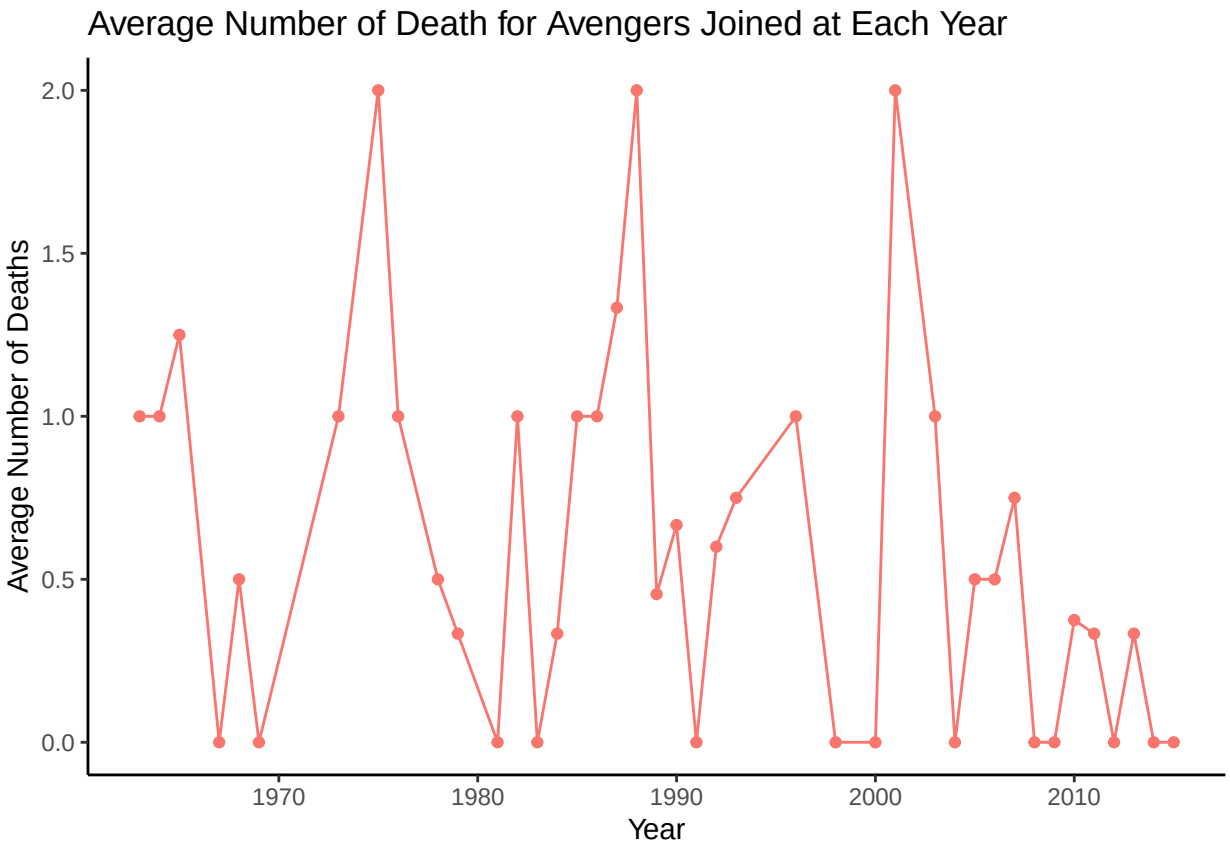
```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, pleas
```

This is a scatter plot with the function `jitter` so that none of the data points overlap. From this scatter plot, it can be noticed that except a few outliers, most Avengers have died zero to two times.

If we go through the plot from right to left along the x-axis, it can be found that a larger proportion of the scattered points are crowed near the x-axis. Meanwhile, the proportion of Avengers with 2 deaths is also decreasing over time. We can conclude that in general, the newer members of the Avengers are more likely to have experienced fewer deaths.

Let's investigate it further with another static line graph.

```
avengers %>%
  filter(years_since_joining < 115)%>%#remove outliers with joining year of 1900
  select(total_death, year) %>%
  group_by(year) %>%
   #calculate averge number of deaths for avengers joined at each year
  summarise(avg_death = mean(total_death)) %>%
  ggplot(aes(year,avg_death, colour = "RdYlBu")) +
           geom_line() +
           geom_point() +
  labs(title = "Average Number of Death for Avengers Joined at Each Year",
       x = "Year", y = "Average Number of Deaths") +
    theme_classic() +
  theme(legend.position = "none")
```



We are now exploring the relationship between the number of deaths a Avenger has experienced and the year the Avenger joined the team. This plot shows that the risk of deaths is not linearly related to the year joined. We would expect that the line is downward sloping, where higher number of deaths would pertain to earlier year joined. The truth is, however, that there are random years that are "safer" to have joined in, where the average number of deaths in that year is 0.

In conclusion, while the number of deaths is related to the number of years one have joined the Avengers, there is not a linear relationship between the two variables. Further investigation could be warranted for the number of deaths versus appearances, and see whether those two are related. We would also assume that this would be close to a linear relationship, but this is to be seen!

Conclusion

In this project, we take a close look at the dataset `Avengers.csv` provided by `fivethirtyeight`. To answer our research questions, we employ various data cleaning skills to modify the data type, shape, variable names, and some contents of the dataset.

For Question 1, the table and bar chart display some hidden preferences of gender and honorary status by the producers of the comics, showing that overall, male Avengers seem to appear more than their female counterparts.

In Question 2, the relationship between the number of years in the Avengers and the risk of experiencing deaths is studied to reveal the interesting pattern that death risk actually fluctuates drastically depending on the year the Avenger joined the team.