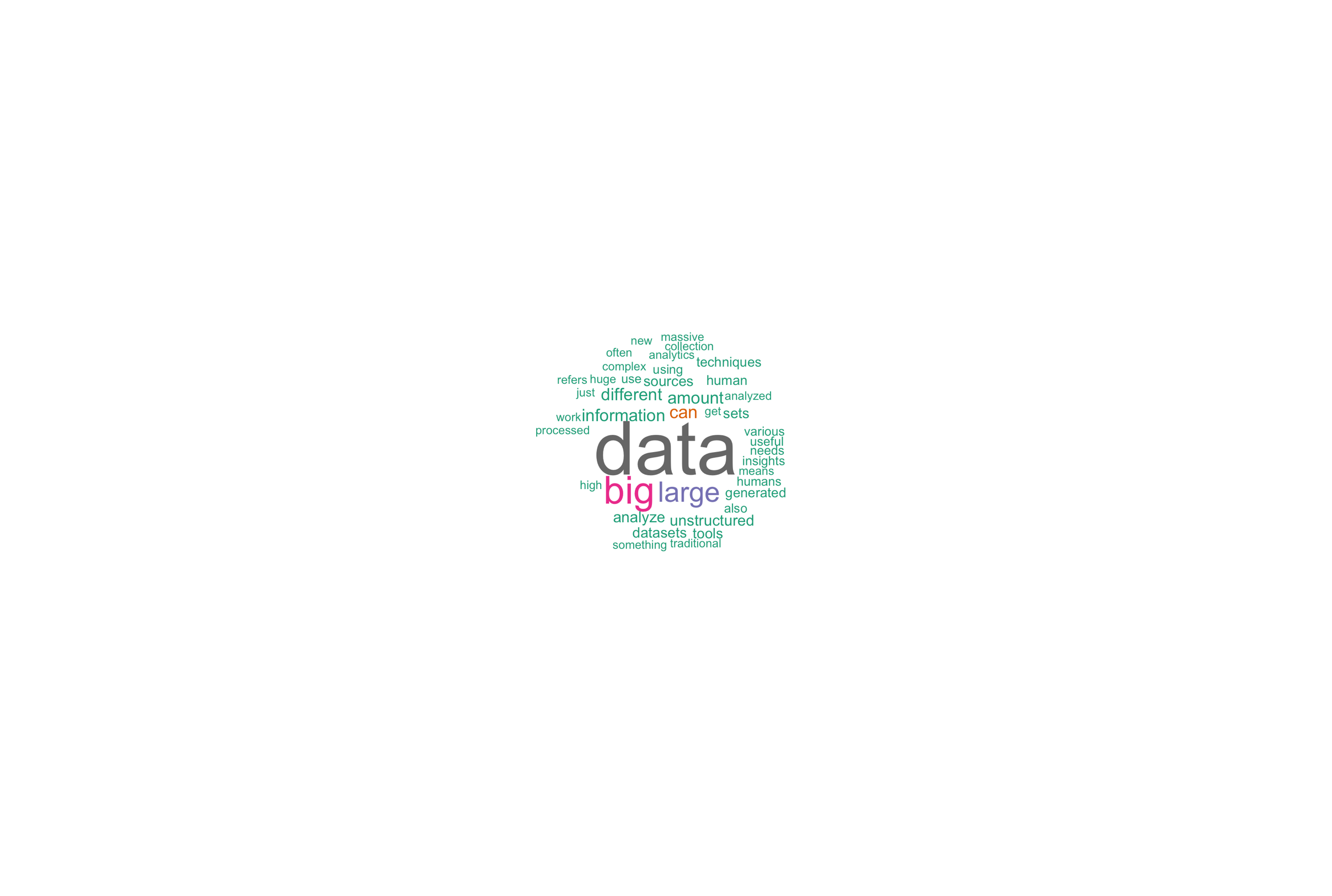
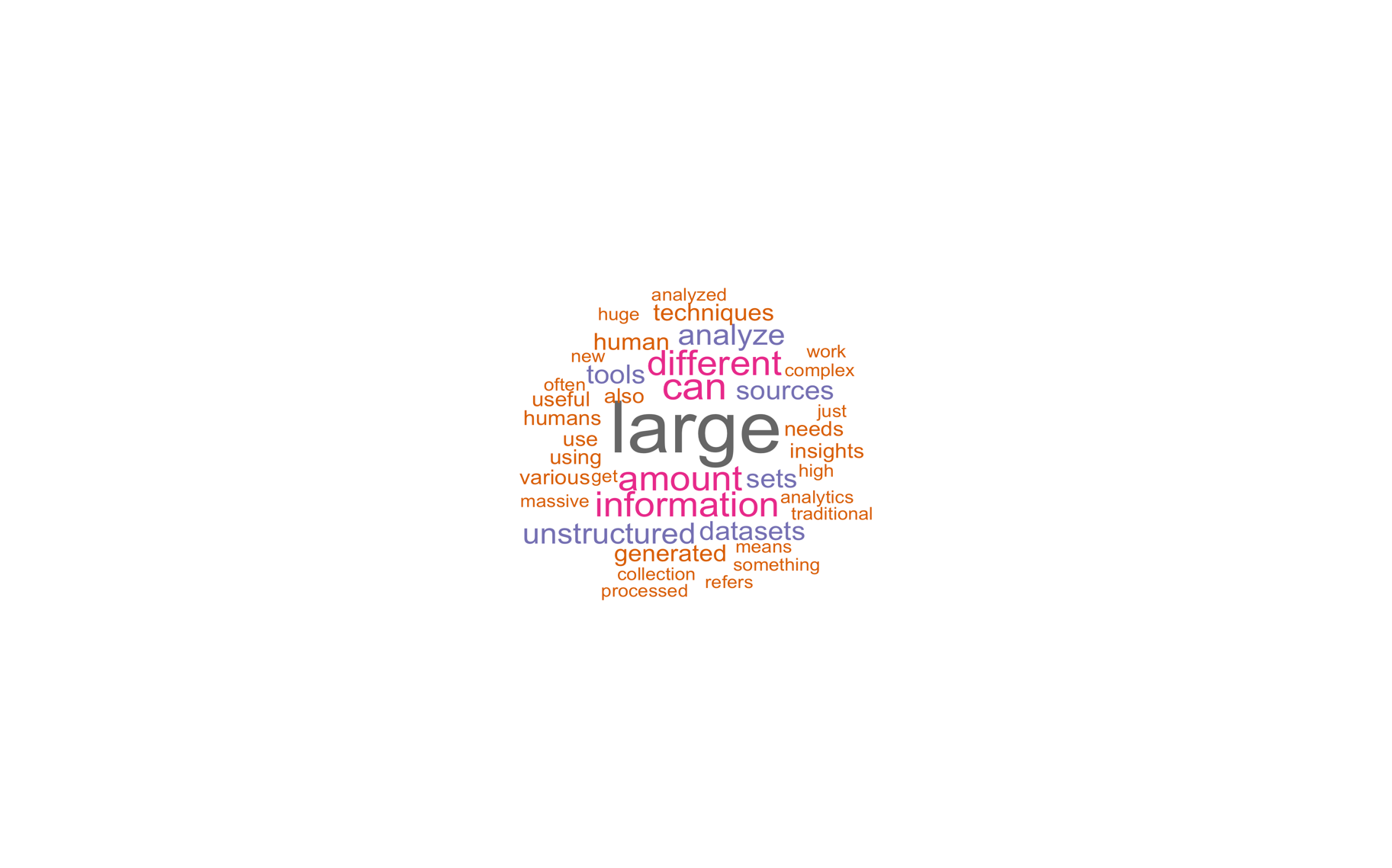
**Web Exercise 2: R and R-Studio**

1. Added the files ([BigData-Definition-V1.png](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/BigData-Definition-V1.png), [BigData-Definition-V2.png](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/BigData-Definition-V2.png))

**1.**  **2.** 

1. For this task I have chosen [Medical Examiner Cases](https://data.sandiegocounty.gov/Safety/Medical-Examiner-Cases/jkvb-n4p7/about_data) dataset from San Diego County Data Catalogue. This dataset contains all accidental deaths, homicides, suicides, in-custody deaths, sudden or unexpected natural deaths which occurred since year 1997. One more thing about this dataset is that all these patients did not see any health consultants during their last 20 days of living. The code for this task can be found at – [data\_analysis.r](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/data_analysis.r)

I have developed 4 different statistical plots from the underlying data, and they are as follows –

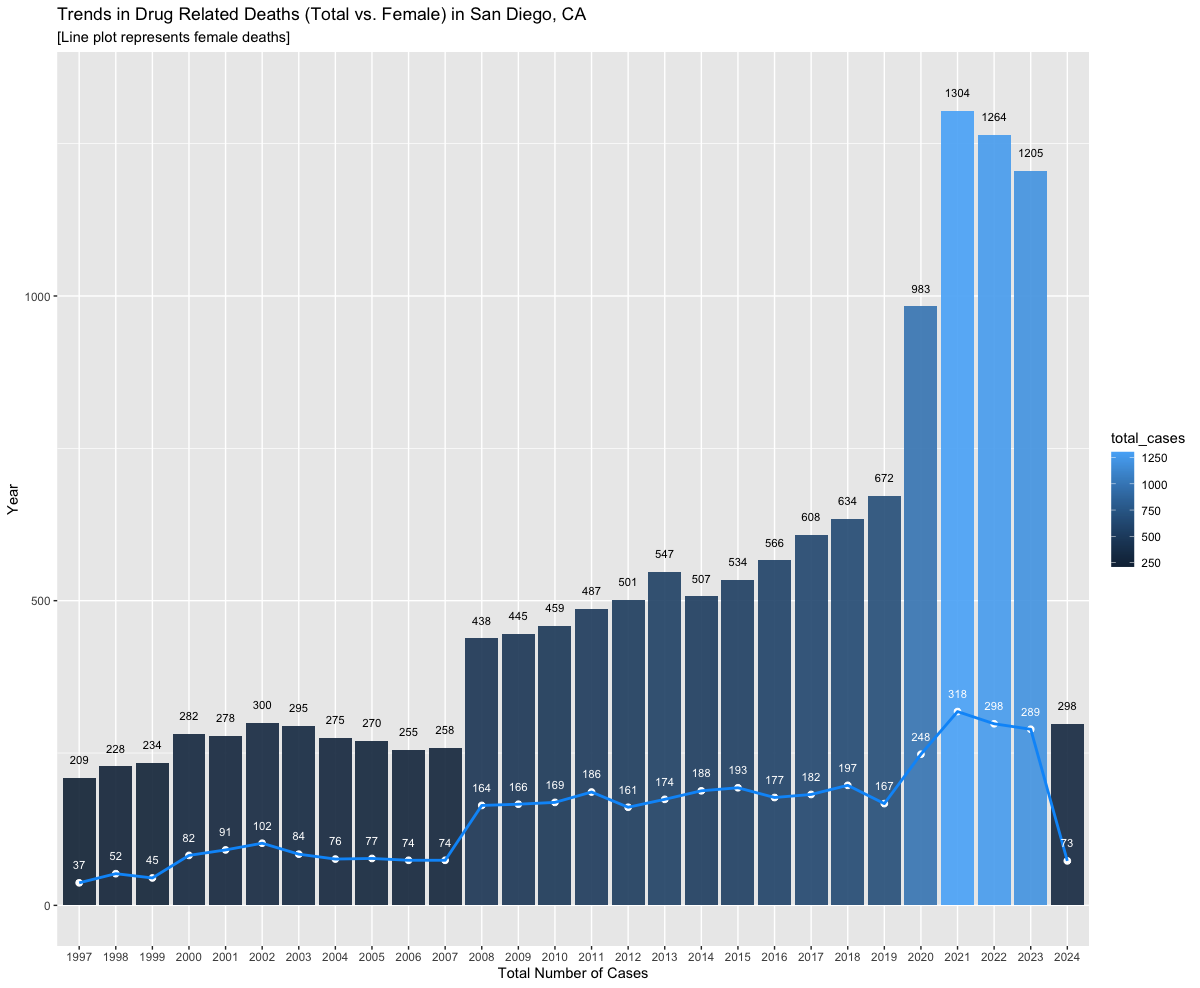
* 1. First graph is basic bar plot of the total number of deaths for each manner of deaths for FY2023. In addition to this, I have also plotted a simple pie chart as well for the same exact data. This shows how many deaths occurred due to each manner of death. I have attached these graphs to the following link – [Bar Plot](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/manner_of_death_bar.png) | [Pie Chart](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/manner_of_death_pie.png)

A graph with different colored bars

Description automatically generated A pie chart with numbers and a graph

Description automatically generated

* 1. For the second graph, I have bult a combined ***bar + line*** chart for showing trend in deaths caused due to drugs over years (1997-2024). This required a lot of exploring of how we can overlay line chart over bar chart. After reading lot of blogs, I finally found out the easiest way is to plot each required element one after other and use **+** to combine each of them. Here in this graph, we can see that the total number of deaths due to drugs are increased over years. However, the number of deaths is on decline since 2021 till 2023. As the data for 2024 is not complete we cannot say for sure if the deaths will decrease further. This plot is uploaded at – [combined bar + line chart](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/drug_trends_over_years.png)



* 1. The 3rd graph is the pie chart for showing total number of deaths across various age groups. This graph suggests that the deaths are highest at the age group 60-69 and least for age groups 0-9 (Not considering unknown age group). The link to the plot – [Pie chart](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/age_groups_deaths.png)

A graph with different colored circles

Description automatically generated with medium confidence

* 1. The final graph is just for fun, and I would like to learn more on using the contour plots more effectively. For now, this shows the density of deaths for each distinct age. Link to the density plot - [Link](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/death_density_plot.png)

A chart showing a diagram of a person's age

Description automatically generated with medium confidence

1. I have created 2 different wordclouds. Both these had a little difference while processing the raw data.
   1. For the 1st one, I have used five different files from Christopher Nolan’s 5 movie scripts (*Dunkirk, Inception, Interstellar, Tenet, The Dark Knight Rises*) – [Dataset Link](https://www.kaggle.com/datasets/diptaraj23/christopher-nolan-last-10-years-movies-subtitles/) | [GitHub Link](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/nolan_scripts.zip)
      * I have used following stopwords - *"the", "youre", "dont", "can", "will", "you", “and” "your", "that", "this", "what", "for", "not", "have", "was", "but"*
      * I have added the wordcloud image file to following link – [nolan\_scripts.png](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/nolan_scripts.png)
      * Additionally, added the code for creating the wordcloud – [nolan\_cloud.r](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/nolan_cloud.r)

A close up of words

Description automatically generated

* 1. For the 2nd one, I have used a relatively simpler data of various companies in USA which are hiring for Data Science positions. This datafile required me to preprocess and cleanup the data before loading to the R environment – [Dataset Link](https://www.kaggle.com/datasets/cormac42/indeed-data-science) | [GitHub Link to preprocessed file](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/job_companies.txt)
     + I removed few identifiers from company names like - *INC, COM, LLP, LTD, etc.,*
     + Unlike the first wordcloud example, I wanted to create wordcloud of whole company names instead of single words, so I changed the approach of the R – code. Instead of using TermDocumentMatrix function to create the tdm matrix, I used simple aggregate function to count number of positions available in each company.
     + Link to the wordcloud image - [top\_companies.png](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/top_companies.png)
     + Link to the code for creating the wordcloud - [companies\_cloud.r](https://github.com/pTidke/BDA594-ptidke/blob/master/Attachments/companies_cloud.r)

A close up of words

Description automatically generated

1. R-Shiny is an open-source R package used to create interactive web applications within the R environment. It provides a platform to build dynamic web interfaces for data analysis and visualizations without needing to know HTML, CSS, or JavaScript.

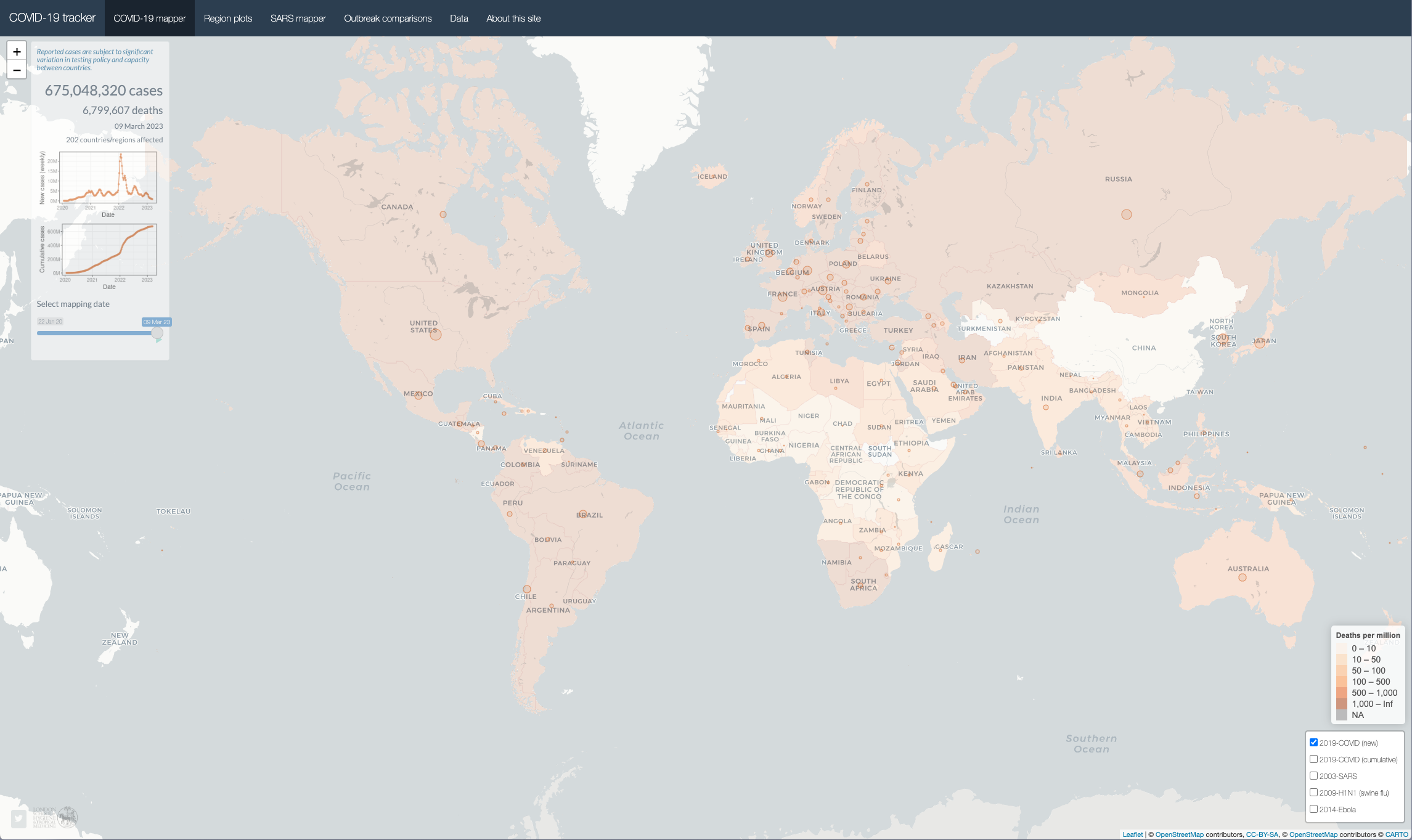
Its main function is to create interactive dashboards and applications. Users can change data input controls and see real-time results. Shiny only runs calculations related to the changed controls (lazy evaluation) and updates the associated charts, graphs, or visualizations.

Key features of R-Shiny include:

* 1. Web Application Development
  2. Reactive Programming (Lazy evaluations)
  3. UI and Server Components
  4. No HTML/CSS Expertise Required

Shiny now supports the Python language, making it useful for users who are more comfortable with Python instead of R. Leading organizations like Pfizer, StackOverflow, Johnson & Johnson, Harvard Medical School and University of Oxford use Shiny.

One very good use case example of R-Shiny is [Covid19 Tracker](https://vac-lshtm.shinyapps.io/ncov_tracker/) –



A screenshot of a computer

Description automatically generated

1. **Three packages in R that could be very useful for data science processing**
   1. **caret**

**caret** (short for Classification and Regression Training) is a comprehensive package in R designed to simplify the process of building machine learning models. Few of the important tools included in this package are -

* + - pre-processing
    - data splitting
    - feature selection
    - model tuning using resampling
    - variable importance estimation

All these tools can be very helpful for developers looking to build optimized end to end classification or regression ML pipelines.  Latest version of **caret** - <http://cran.r-project.org/web/packages/caret/>

* 1. **tidyr**

**tidyr** is an R package focused on simplifying data cleaning and transformation by organizing messy data into a tidy format. The concept of "tidy data" involves structuring data in a way that -

* + - each variable forms a column,
    - each observation forms a row, and
    - each type of observational unit forms a table.

**tidyr** provides easy-to-use functions like **gather(), spread(), separate(),** and **unite()** to restructure and reshape datasets for further analysis. It is particularly helpful in converting data into a consistent format, allowing for more efficient manipulation, visualization, and modeling.  **tidyr** is part of the larger **tidyverse** collection of R packages, which also includes **ggplot2** among others.

Latest version of **tidyr -** <https://tidyr.tidyverse.org/>

* 1. **rvest**

**rvest** is a package for web scraping and extracting data from HTML web pages. Although this is not primarily a machine learning package, it can be a valuable supporting tool in our toolkit for machine learning tasks. **rvest** simplifies the process of web data extraction by providing functions to navigate HTML documents, select nodes, and extract information. Functions like read\_html() and html\_nodes() make it easy to pull data from web pages for analysis. Rvest is particularly useful for collecting data from websites where APIs are not available, enabling users to gather and process web data efficiently. According to the original developer’s recommendation, it is best to use the **rvest** package along with the **polite** package. This approach helps ensure that websites are not overwhelmed with too many requests.

Latest version of **rvest -** <https://rvest.tidyverse.org/>