
advanced
refers
meaningful
complex
excel
amount
tools
amounts
sets
like
analysis
can
large
used
decisions
huge
data
issues
platforms
analyze
big
information
different
collection
think
datasets
make
insights
use
analyzed
human
handle
may
way
process
subject
processing
unstructured


unstructured
insights
human
analyzed
handle
meaningful
issues
tools
make
huge
information
analyze
large
like
collection
analysis
advanced
process
sets
can
different
decisions
amount
complex
used
way
amounts
datasets
use
subject
refers
excel
may
platforms
processing
think

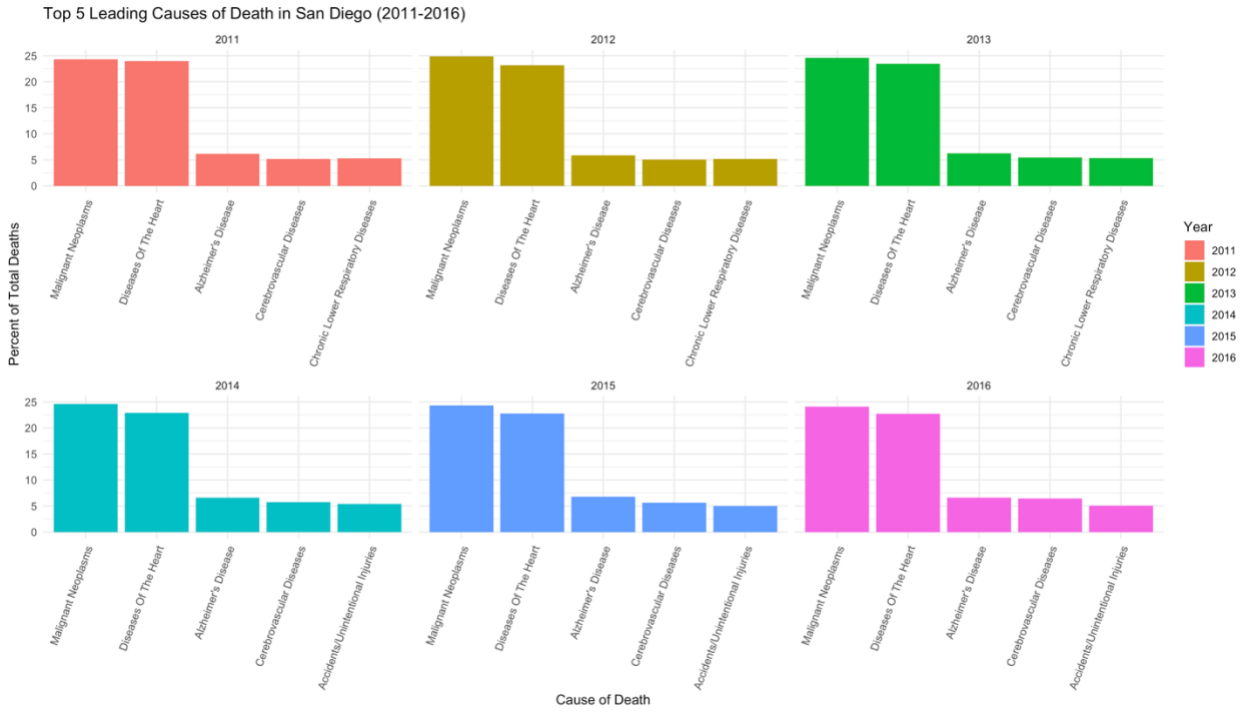
Nhan Le

BDA 594

September 8, 2025

Web Exercise 2

2. The following is a visual graphic illustrating the top 5 leading underlying cause of deaths in San Diego County from 2011-2016. The data is extracted from San Diego County's Data Catalogue, "ARCHIVED - Leading Causes of Death in San Diego County". The visual is done using R and R Studio. The two R packages being implemented in the code are:



- ggplot2

- dplyr

Code:

```
# Set working directory

 setwd("/Users/ren/Documents/webexercise-2")

# Load & install required libraries

# install.packages("dplyr")

library(ggplot2)

library(dplyr)
```

```
# Read csv data into R dataframe

cause_of_death_data <- read.csv("Leading_Causes_of_Death_in_SD_2011_2016.csv")

# List all the field names

names(cause_of_death_data)

> top5_causes <- cause_of_death_data %>%

        select(Year, Rank, `San.Diego.County.Underlying.Cause.of.Death`,
`San.Diego.County.Percent.Deaths`) %>%

        filter(Rank <= 5)

 # Rename columns

>colnames(top5_causes) <- c("Year", "Rank", "Cause", "Percent")

# With reference to Jay Yang's code, make a visual plot using the ggplot2 package, aes() and
geom_bar():

> ggplot(top5_causes, aes(x = reorder(Cause, -Percent), y = Percent, fill = factor(Year))) +
geom_bar(stat = "identity", position = "dodge") + facet_wrap(~Year, scales = "free_x") +
labs(title = "Top 5 Leading Causes of Death in San Diego (2011–2016)",  x = "Cause of Death",
y = "Percent of Total Deaths",  fill = "Year") + theme_minimal() +  theme(axis.text.x =
element_text(angle = 70, hjust = 1))

#show and save the plot to the working directory. I add the bg = "white" argument since the
export png was with light gray background which was hard to see.

> ggsave("Top5_Causes_By_Year_by_Nhan_Le_1.png", width = 14, height = 8, bg= "white")
```

3. Text source description:

For the text, I downloaded the book, "The Opium Monopoly" by Ellen N. La Motte, from Project Gutenberg. For the word cloud, I only use chapter "Great Britain's Opium Monopoly". For the selected "stopwords", aside from the common English stop words, I also select words such as: "British", "India", "America", "China", "Government".

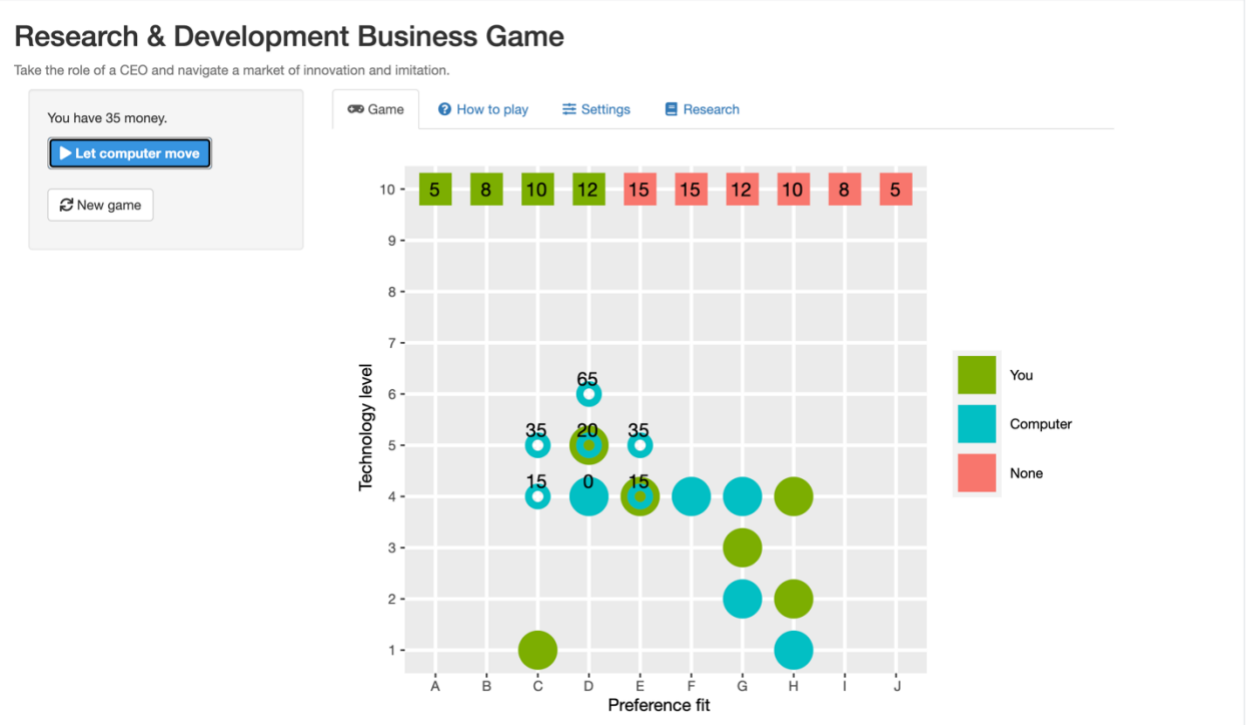
4. Shiny is a programming package or library of R that you can download and install from R and function in any R environment. The Shiny package allows programmer, data scientists, analysts, statisticians an easy way to build web applications that can be interacted by users without the need of using other front-end languages like HTML, CSS, and JavaScript. However, for more customization, you can add the front-end languages.

A Shiny app consists of a UI (User interface) which allows the users to view and interact with the visuals, a server that process the inputs from the users and return the outputs to the UI using R code, and a call, shinyApp(), that glues all the elements together and make the application run.

One of the strong points of Shiny is that whichever change you made to the data you input automatically get updated to the output. From the perspective of the programmer working without Shiny, you normally have to write code, run it to generate a plot, then rerun it again whenever you change the underlying data. The plot you produce would also be static. Whereases with Shiny, you only need to set up the UI and server, and the code runs in an interactive web app. You don't need to rerun a code to create new visual whenever a end-user changes an input. Your Shiny app can run locally on a viewer pane or another window on your laptop, but it can also be run on an external web browser, or on shinyapps.io.

Online Example: R&D Business Game

Description: This is a business strategy roleplaying game created by Paul Simmering. You are acting as the CEO of a tech company going against another opponent tech company. Your goal is to win the favor of all the consumer and maximize profit.



5. The three packages that I will be introduce are as follow:

- **dplyr** is an R package that allow you to manipulate your data more easily because of its simpler function. Some of its typical functions are “filter()”, “arrange()”, ”slice()”, etc. One of the most convenient things from the package is the pipe, %>%. It allows you to get an output from one command and immediately plug it to the next command without having to create additional variables . I can use an example line of code from task number 2:

```
> top5_causes <- cause_of_death_data %>%
        select(Year, Rank, `San.Diego.County.Underlying.Cause.of.Death`,
`San.Diego.County.Percent.Deaths`) %>%
        filter(Rank <= 5)
        ## from the cause_of_death_data select the required data, then filter and
only use the row with rank<=5, then save everything into top5_causes.
```

  If not, then I would have to write something like:

>selected_data <- cause_of_death_data[, c("Year", "Rank", "San.Diego.County.Underlying.Cause.of.Death", "San.Diego.County.Percent.Deaths")]

>top5_causes <- selected_data[selected_data$Rank <= 5,]

- **rvest** is another data-science oriented R package that allows data scientist to scrape data from websites, mostly in the format of HTML. Some of the data that rvest can scrape includes texts, tables, links, attributes and metadata. Some of the key functions are read_html("web link") –gather all the content from the webpage and allow you to work on it through R environment, read_text("web link") – extract text content, read_table("web link") – extract tables, html_nodes() – gather select elements, etc. For example, you want to gather lists from an article about GLP-1 by Havard Health Publishing:

  >library(rvest) ##run the library first

  >webpage <- read_html("https://www.health.harvard.edu/staying-healthy/glp-1-diabetes-and-weight-loss-drug-side-effects-ozempic-face-and-more") #read all html elements and save it to the webpage attribute in R environment, in my case, R Studio.

  >uls <- html_nodes(webpage, "ul") #extract all uls nodes. In order to see which type of data want to get, open the browser developer mode. The reason why you type "ul" instead of "li" is because "ul" refers to the whole list while "li" only refers to one item of the list.

  >length(uls) #this is to check how many lists you have

  >ul1_text<- html_text(uls[1]) #if you want to extract text only from the first list.

- **ggplot2** is an R package that allows data scientists and analysts to plot and draw static visualization in R environment. Some of its common functions are:

- ggplot(): this starts the plot by creating the coordination and inside the parentheses you can start adding arguments on which data you want to use and how you want to represent it. For example: ggplot(top5_causes, aes(x = reorder(Cause, -Percent), y = Percent, fill = factor(Year))) means creating a plot coordination with the data top5_causes, x-axis shows the cause arrange by the percentage, y-axis shows the percent, and fill different color by the year.

- aes(): the name is short for aesthetic. This is a function nested within the ggplot() function, all the arguments on how the data is represented on the plot should be inside the aes() function. For example: : **ggplot(**top5_causes, **aes(**x = reorder(Cause, -Percent), y = Percent, fill = factor(Year))).

-geom_bar(): as the geom, short for geomatric, indicates, this is a function used to draw geometric shapes. In this case, specifically, it is used to draw bar charts. Some more instances of the geom_ functions are geom_point(), geom_boxplot(),etc.

In short, ggplot2 is a package that provides a structure way for data users to visualized sophisticated data in R.