library(formatR) library(tidyverse) #Used to load tidyverse package into R. library(readr) #Used to load readr package in R to use to read CSV file.

#Loading the data set Video\_Games\_Sales <- read\_csv(“C:/Users/Matthew Afoakwah/OneDrive/Documents/Bport/Bport Spring 2023/CSC435 Data Analytics/Project/CSC435\_Video\_Games\_Sales\_Project/data/Video\_Games\_Sales\_as\_at\_22\_Dec\_2016.csv”)

#head() function to get a preview of the dataset. head(Video\_Games\_Sales)

#shows a glimpse() of the data set along with the attributes. glimpse(Video\_Games\_Sales)

#Shows the name of the columns in the Video\_Games\_Sales data set. colnames(Video\_Games\_Sales)

#dim() function to find the length of the data set. dim(Video\_Games\_Sales)

#sapply() function to apply the class function on each attribute of our data set. sapply(Video\_Games\_Sales, class)

#filter() funtion to filter out all years less then 2010 and greater then 2016. gamesales2010\_2016 <- filter(Video\_Games\_Sales, Year\_of\_Release >= 2010 ) gamesales2010\_2016 <- filter(gamesales2010\_2016, Year\_of\_Release <= 2016)

#The filter() function removes all values that are “N/A” in the Year\_of\_Release column gamesales2010\_2016 <- filter(gamesales2010\_2016, Year\_of\_Release != “N/A”)

#The duplicated() function finds all rows where there are repeated occurences. Then we use the ! to filter them out from the graph. gamesales2010\_2016 <- gamesales2010\_2016[!duplicated(gamesales2010\_2016), ]

#Create a new variable outliers, then use boxplot.stats() to assign outliers from Global sales to outliers. boxplot(gamesales2010\_2016$Global\_Sales, main = "Global Sales BoxPlot") outliers <- boxplot.stats(gamesales2010\_2016$Global\_Sales)$out

#Create a new variable outliersNA, then use boxplot.stats() to assign our outliers from NA sales to outliersNA. boxplot(gamesales2010\_2016$NA\_Sales, main = "NA Sales BoxPlot") outliersNA <- boxplot.stats(gamesales2010\_2016$NA\_Sales)$out

#unique function lists the unique characters in the platform column. unique(gamesales2010\_2016$Platform)

#We use recode() to change the values of certain attributes to new attributes such as Nintendo, PlayStation, and Xbox. gamesales2010\_2016Platform, “X360” = “Xbox”, “PS3” = “PlayStation”, “DS” = “Nintendo”, “PS4” = “PlayStation”, “3DS” = “Nintendo”, “Wii” = “Nintendo”, “XOne” = “Xbox”, “WiiU” = “Nintendo”, “PSP” = “PlayStation”, “PSV” = “PlayStation”, “PS2” = “PlayStation”)

#We use unique()to list the unique attributes in the platform column. unique(gamesales2010\_2016$Platform)

#We create a new data thar has the sum of NA sales for each platform. platform\_NA\_Sales <- gamesales2010\_2016 %>% #We use the group\_by() to group all the platforms in the platform\_NA\_Sales together.  
group\_by(Platform) %>% #We use the summarize() to summarize the sum of NA\_sales for each different platform. summarize(NA\_total = sum(NA\_Sales)) %>%  
#We arrange the data set by decreasing order according to the NA\_total. arrange(desc(NA\_total)) #We use the head() function to preview the new data set. head(platform\_NA\_Sales)

#We create a new data thar has the sum of NA and Global sales for each platform in each year. platform\_SalesbyYear <- gamesales2010\_2016 %>% #We use the group\_by() to group all the Platform and Year\_of\_Release together. group\_by(Platform, Year\_of\_Release) %>% #We use the summarize() to summarize the sum of NA\_sales for each different platform, and year. summarize(NA\_total = sum(NA\_Sales), glbl\_total = sum(Global\_Sales)) %>%  
#We arrange the data set by decreasing order according to the glbl\_total. arrange(desc(glbl\_total)) #We use the head() function to preview the new data set. head(platform\_SalesbyYear)

#We move the gamesales2010\_2016 data into a new data set then pipe the command. platform\_glbl\_sales <- gamesales2010\_2016 %>% #We use the group\_by() to group all the platforms together. group\_by(Platform) %>% #We use the summarize function to summarize the sum of Global\_sales for each different platform. summarize(glbl\_total = sum(Global\_Sales)) %>% #We arrange the data set by decreasing order according to the glbl\_total.  
arrange(desc(glbl\_total)) #We use the head() function to preview the table. head(platform\_glbl\_sales)

#We use the unique to search for any irregular values in the categories column. unique(gamesales2010\_2016$Genre)

#We move the gamesales2010\_2016 data into a new data set. genre\_na\_sales <- gamesales2010\_2016 %>% #We use the group\_by() to group all the Genres together. group\_by(Genre) %>%  
#We use summarize() to summarize the sum of na\_sales for each different genre. summarise(genre\_na\_total = sum(NA\_Sales)) %>%  
#We arrange the data set by decreasing order according to the genre\_na\_total. arrange(desc(genre\_na\_total)) #We use the head() function to preview the table. head(genre\_na\_sales)

#We move the gamesales2010\_2016 data into a new data set. genre\_SalesbyYear <- gamesales2010\_2016 %>% #We use the group\_by() to group all the Genre and Year\_of\_Release together. group\_by(Genre, Year\_of\_Release) %>% #We use the summarize() to summarize the sum of NA\_sales for each different Genre, and year.  
summarize(NA\_total = sum(NA\_Sales), glbl\_total = sum(Global\_Sales)) %>% #We arrange the data set by decreasing order according to the glbl\_total. arrange(desc(glbl\_total)) #We use the head() function to preview the new data set. head(genre\_SalesbyYear)

#We move the gamesales2010\_2016 data into a new data set then pipe the command. genre\_glbl\_sales <-gamesales2010\_2016 %>% #We use group\_by() to group all the Genres in the genre\_glbl\_sales together. group\_by(Genre) %>%  
#We use summarize() to summarize the sum of Global\_sales for each different genre. summarise(genre\_glbl\_total = sum(Global\_Sales)) %>%  
#We arrange the data set by decreasing order according to the genre\_na\_total. arrange(desc(genre\_glbl\_total)) #We use the head() function to preview the table. head(genre\_glbl\_sales)

#We use unique() to take all unique values in the developer column then sort them. unique\_dev <- sort(unique(gamesales2010\_2016$Developer))

#We use the recode() function from the dplyr package from our tidyverse package in order change our variables. gamesales2010\_2016Developer,“1C: Maddox Games” =“1C Company”, “1C:Ino-Co” = “1C Company”,“2K Australia” = “2K Games”, “2K Czech” = “2K Games”, “2K Sports” = “2K Games”, “2K Marin” = “2K Games”, “2K Play” = “2K Games”, “505 Games, Sarbakan Inc.” = “505 Games”, “Activision, Behaviour Interactive” = “Activision”, “Activision, FreeStyleGames” = “Activision”, “Ambrella, The Pokemon Company” = “Ambrella”, “Armature Studio, comcept” = “Armature Studio”, “Artificial Mind and Movement, EA Redwood Shores” = “Artificial Mind and Movement”, “Atari, Atari SA” = “Atari”, “Atari, Slightly Mad Studios, Atari SA” = “Atari”, “Atlus, Dingo Inc.” = “Atlus”, “Atomic Planet Entertainment” = “Atomic Games”, “Avalanche Software” = “Avalanche Studios”, “Bandai Namco Games, Artdink” = “Bandai Namco Games”, “Beenox, Other Ocean Interactive” = “Beenox”, “Big Blue Bubble Inc., Scholastic, Inc.” = “Big Blue Bubble Inc.”, “Big Blue Bubble Inc., Scholastic, Inc.” = “Big Blue Bubble Inc.”, “Blitz Games Studios” = “Blitz Games”, “Blue Byte, Related Designs” = “Blue Byte”, “Bungie Software, Bungie” = “Bungie”, “Capcom Vancouver” = “Capcom”, “Capcom, Pipeworks Software, Inc.” = “Capcom”)

gamesales2010\_2016Developer, “Capcom, QLOC” = “Capcom”, “Climax Entertainment” = “Climax Studios”, “Climax Group” = “Climax Studios”, “Climax Group, Climax Studios” = “Climax Studios”, “Codemasters Birmingham” = “Codemasters”, “Compile Heart, GCREST” = “Compile Heart”, “Crave, DTP Entertainment” = “Crave”, “Crystal Dynamics, Nixxes Software” = “Crystal Dynamics”, “Cyanide, Cyanide Studios” = “Cyanide”, “CyberConnect2, Racjin” = “CyberConnect2”, “CyberPlanet Interactive Public Co., Ltd., Maximum Family Games” = “CyberPlanet Interactive Public Co., Ltd.”, “Deep Silver Dambuster Studios” = “Deep Silver”, “Deep Silver, Keen Games” = “Deep Silver”, “Dimps Corporation, Dream Execution” = “Dimps Corporation”, “Dimps Corporation, SCE Japan Studio” = “Dimps Corporation”, “Disney Interactive Studios, Land Ho!” = “Disney Interactive Studios”, “EA Black Box” = “EA Games”, “EA Bright Light” = “EA Games”, “EA Canada” = “EA Games”, “EA Canada, EA Vancouver” = “EA Games” , “EA DICE” = “EA Games”, “EA DICE, Danger Close” = “EA Games”) #“EA Sports” = “EA Games”

gamesales2010\_2016Developer, “EA Sports” = “EA Games”, “EA Montreal” = “EA Games”, “EA Redwood Shores” = “EA Games”, “EA Sports, EA Canada” = “EA Games”, “EA Sports, EA Vancouver” = “EA Games”, “EA Tiburon” = “EA Games”, “Eidos Montreal, Nixxes Software” = “Eidos Montreal”, “Engine Software, Re-Logic” = “Engine Software”, “Epic Games, People Can Fly” = “Epic Games”, “Farsight Studios, Crave” = “Farsight Studios”, “Gaijin Entertainment” = “Gaijin Games”, “Gearbox Software, 3D Realms” = “Gearbox Software”, “Gearbox Software, WayForward” = “Gearbox Software”, “Guerilla Cambridge” = “Guerrilla Cambridge”, “Guerilla” = “Guerilla Cambridge”)

gamesales2010\_2016Developer,“Harmonix Music Systems, Demiurge” = “Harmonix Music Systems”, “Headup Games, Crenetic Studios” = “Headup Games”, “Konami Computer Entertainment Hawaii” = “Konami”, “Marvelous AQL” = “Marvelous Inc.”, “Marvelous Entertainment” = “Marvelous Inc.”, “Midway Studios - Austin” = “Midway”, “Monolith Soft” = “Monolith Productions”, “Monolith Soft, Banpresto” = “Monolith Productions”)

gamesales2010\_2016Developer,“Namco Bandai Games America, Namco Bandai Games” = “Namco Bandai Games”, “Namco Bandai Games, Bandai Namco Games” = “Namco Bandai Games”, “Namco Bandai Games, Cellius” = “Namco Bandai Games”, “Namco Bandai Games, Monkey Bar Games” = “Namco Bandai Games”, “NATSUME ATARI Inc.” = “Natsume”, “Nintendo EAD Tokyo” = “Nintendo”, “Nintendo, Camelot Software Planning” = “Nintendo”, “Nintendo, Headstrong Games” = “Nintendo”, “Nintendo, Intelligent Systems” = “Nintendo”, “Nintendo, Nd Cube” = “Nintendo”, “Nintendo, Nintendo Software Technology” = “Nintendo”,“Nintendo, Spike Chunsoft” = “Nintendo”, “Paradox Development Studio” = “Paradox Interactive”)

gamesales2010\_2016Developer,“PLAYGROUND, Playground Games” = “Playground Games”, “Retro Studios, Entertainment Analysis & Development Division” = “Retro Studios”, “Rockstar Leeds” = “Rockstar Studios”, “Rockstar North” = “Rockstar Studios”, “Rockstar San Diego” = “Rockstar Studios”, “Sanzaru Games, Sanzaru Games, Inc.” = “Sanzaru Games”, “SCE Japan Studio, comcept” = “SCE Studio”, “SCE Santa Monica” = “SCE Studio”, “SCE Studio Cambridge” = “SCE Studio”, “SCE Japan Studio” = “SCE Studio”, “SCEA San Diego Studios” = “SCEA”,“SCEA, Zindagi Games” = “SCEA”, “SCEE London Studio” = “SCEE”)

gamesales2010\_2016Developer, “Sega Studios San Francisco” = “Sega”, “Sega Toys” = “Sega”, “Sega, Dimps Corporation” = “Sega”, “Sega, French-Bread” = “Sega”, “Sega, Sonic Team” = “Sega”, “Snapdragon” = “Snap Dragon Games”, “Sonic Team” = “Sega”, “Sony Bend” = “Sony Interactive Entertainment”, “Sony Online Entertainment” = “Sony Interactive Entertainment”, “Spike Chunsoft” = “Spike”, “Spike Chunsoft Co. Ltd., Spike Chunsoft” = “Spike”,“Tecmo” = “Tecmo Koei Games”, “Tecmo Koei Canada” = “Tecmo Koei Games”, “THQ Australia” = “THQ”, “THQ Digital Studio Phoenix” = “THQ”, “Ubisoft Casablanca” = “Ubisoft”, “Ubisoft Milan” = “Ubisoft”, “Ubisoft Montpellier” = “Ubisoft”, “Ubisoft Montreal” = “Ubisoft”, “Ubisoft Osaka” = “Ubisoft”, “Ubisoft Paris” = “Ubisoft”, “Ubisoft Paris, Ubisoft Montpellier” = “Ubisoft”, “Ubisoft Quebec” = “Ubisoft”, “Ubisoft Reflections” = “Ubisoft”, “Ubisoft Reflections, Ivory Tower” = “Ubisoft”, “Ubisoft Romania” = “Ubisoft”, “Ubisoft Sofia” = “Ubisoft”, “Ubisoft Toronto” = “Ubisoft”, “Ubisoft Vancouver” = “Ubisoft”, “Ubisoft, FunHouse” = “Ubisoft”, “Ubisoft, Ludia Inc.” = “Ubisoft”, “Ubisoft, Ubisoft Montreal” = “Ubisoft”)

dev\_global <- gamesales2010\_2016 %>% #We want to filter out any unknown Developers as there are so few NA developers that will affect our variables. filter(Developer != “N/A”) %>% #We will then group the columns according to the value in the developer columns. group\_by(Developer) %>% #We sum up the total global sales and NA sales of each developers games earned. summarise(global\_total = sum(Global\_Sales)) %>% #We then arrange each row in descending order based on the global\_total. arrange(desc(global\_total)) %>% #We use the slice\_head() function to take ONLY the top five most profitable developers. slice\_head(n = 5) #We preview the data set. head(dev\_global)

#We create a table which shows video games where the developers are the top five developers. developer\_genre\_sale <- gamesales2010\_2016 %>% filter(Developer %in% c(“EA Games”, “Ubisoft”, “Nintendo”, “Rockstar Studios”, “Treyarch”)) #We use this to preview the data set. head(developer\_genre\_sale)

#We use is.na() to determine if any values in Genre is N/A. #Then use sum() to count the exact number of N/A values. sum(is.na(developer\_genre\_sale$Genre))

#We store the developer\_genre\_sale data into df. df <- developer\_genre\_sale %>% #We then group by Developer and Genre. group\_by(Developer,Genre) %>%  
#We summarise the NA and Global total for each category based on developer. summarise(developer\_glbl\_total = sum(Global\_Sales), developer\_NA\_total = sum(NA\_Sales)) %>% #We arrange the order in alphabetical order using the Developer column. arrange(Developer) #We preview the data set. head(df)

#GRAPHS

#This code produces a graph in which it displays the sum of all Platforms in the 2010 to 2016 range. ggplot(data = platform\_NA\_Sales, aes(x = reorder(Platform, NA\_total), y = NA\_total)) + geom\_bar(stat = “identity”) + labs(title = “NA sales by Platform”, x = “Platform”, y = “NA Sales”) + theme(axis.text.x = element\_text(angle = 45, hjust = 1))

#This code creates a bar graph that breaks up the total amount of NA sales for each of the different years of each different Platform. ggplot(platform\_SalesbyYear, aes(x = Year\_of\_Release, y = NA\_total, fill = Platform)) + geom\_bar(stat = “identity”) + labs(title = “Total NA sales by Video Game Platform (2010-2016)”, x = “Year”, y = “NA Sales(In millions)”)

#This code generates a lollipop graph in which it displays the total global sales of each platform in 2010 to 2016. ggplot(platform\_glbl\_sales , aes(x = Platform, y = glbl\_total, color = Platform)) + geom\_point(stat = “identity”) + geom\_segment(aes( x = Platform, xend = Platform, y = 0, yend = glbl\_total )) + labs(title = “Global Sales by Platform”, x = “Platform”, y = “Global Sales(in millions)”)

#This code generates pie charts which shows the difference in sales between each platform from 2010 to 2016. ggplot(data = platform\_SalesbyYear, aes(x = 0, y = glbl\_total, fill = Platform)) + geom\_col(position = “fill”) + facet\_wrap(~Year\_of\_Release) + coord\_polar(theta = “y”) + theme\_void() + labs(title = “Global Sales by Platform”)

#This code generates a line graph that shows the rate of increase/decrease of each Platform from 2010 to 2016. ggplot(data = platform\_SalesbyYear, aes(x = Year\_of\_Release, y = glbl\_total, color = Platform)) + geom\_point() + geom\_line(aes(group = Platform)) + labs(title = “Global Video Game Sales (2010-2016)”, x = “Year”, y = “Global Sales(in millions)”)

#This code creates a bar graph that displays the overall sum of sales of each genre between 2010 to 2016. ggplot(data = genre\_na\_sales, aes(x = reorder(Genre,genre\_na\_total), y = genre\_na\_total, fill = Genre)) + geom\_bar(stat = “identity”) + labs(title = “NA sales by Genre”, x = “Genre”, y = “NA Sales”) + theme(axis.text.x = element\_text(angle = 45, hjust = 1))

#We convert our numeric values in genre\_na\_sales into a percentage of the entire sum and store it into pie\_labels. pie\_labels <- paste0(round(100 \*genre\_na\_salesgenre\_na\_total), 2), “%”) #This code creates a pie graph, then displays the difference in NA sales each genre made out of 100% of the total sales made between 2010 to 2016 ggplot(genre\_na\_sales, aes(x = ““, y = genre\_na\_total, fill = Genre)) + geom\_col(color =”black”) + geom\_text(aes(label = pie\_labels), position = position\_stack(vjust = 0.5)) + coord\_polar(theta = “y”) + labs( y = “NA Sales”, title = “NA Sales by Genre”)

#This code creates a line graph that displays the change of NA sales based on each genre over the past 6 years. ggplot(data = genre\_SalesbyYear, aes(x = Year\_of\_Release, y = NA\_total, color = Genre)) + geom\_point() + geom\_line(aes(group = Genre)) + labs(title = “NA Video Game Sales (2010-2016)”, x = “Year”, y = “NA Sales(in millions)”)

#This code generates a bar graph that displays the total Global sales of each video game genre between 2010 to 2016. ggplot(data = genre\_glbl\_sales, aes(x = Genre, y = genre\_glbl\_total, fill = Genre)) + geom\_bar(stat = “identity”) + labs(title = “Global Sales by Genre”, x = “Genre”, y = “Global Sales(In millions)”) + theme(axis.text.x = element\_text(angle = 45, hjust = 1))

#We convert our numerical values into percentages, then store into pvgsg. PVGSG <- genre\_glbl\_sales %>% mutate(percentage=paste0(round(genre\_glbl\_total/sum(genre\_glbl\_total)\*100, 2), “%”)) PVGSG <- PVGSG %>% arrange(desc(genre\_glbl\_total)) #This code displays a stacked bar graph with a couple of functions at play. ggplot(PVGSG, aes(fill = Genre, y = genre\_glbl\_total, x = ““)) + geom\_bar(position =”fill”, stat = “identity”)

#We convert our numeric values in genre\_glbl\_sales into a percentage of the entire sum and store it into pie\_labels2. pie\_labels2 <- paste0(round(100\*genre\_glbl\_salesgenre\_glbl\_total), 2), “%”) #This code creates a pie chart in which it displays the sum of global sale of video games between 2010 to 2016 in a percentage of the total sales made by genre. ggplot(genre\_glbl\_sales, aes(x = ““, y = genre\_glbl\_total, fill = Genre)) + geom\_col(color =”black”) + geom\_text(aes(label = pie\_labels2), position = position\_stack(vjust = 0.5)) + coord\_polar(theta = “y”)

#This line creates a line chart that shows the decrease/increase of gloabl video game sales each year. ggplot(data = genre\_SalesbyYear, aes(x = Year\_of\_Release, y = glbl\_total, color = Genre)) + geom\_point() + geom\_line(aes(group = Genre)) + labs(title = “Global Video Game Sales (2010-2016)”, x = “Year”, y = “Global Sales(in millions)”)

#Overall, this code generates a bar chart that ranks the top 5 video game developers by their total global sales across all regions. ggplot(dev\_global, aes(x = reorder(Developer, global\_total), y = global\_total, fill = Developer)) + geom\_bar(stat = “identity”) + theme(axis.text.x = element\_text(angle = 45, hjust = 1)) + labs(title = “Top 5 Developer Global Sales”, x = “Developer”, y = “Global Sales(In millions)”)

#we convert our numeric values in genre\_glbl\_sales into a percentage of the entire sum and store it into pie\_labels2. pie\_labels3 <- paste0(round(100\*dev\_globalglobal\_total), 2), “%”) #Overall, this code generates a pie chart that ranks the top 5 video game developers by their total global sales across all regions. ggplot(dev\_global, aes(x = ““, y = global\_total, fill = Developer)) + geom\_col(stat =”identity”, width = 1, color = “black”) + coord\_polar(theta = “y”) + theme\_void() + geom\_text(aes(label = pie\_labels3), position = position\_stack(vjust = 0.5)) + labs(title = ” Top 5 Developer Sales”)

#Overall, this code generates a stacked bar chart that shows the contribution of each video game genre to the global sales of the top 5 video game developers. ggplot(df, aes( x = Developer, y = developer\_glbl\_total, fill = Genre)) + geom\_bar(stat = “identity”) + theme(axis.text.x = element\_text(angle = 45, hjust = 1)) + labs(title = “Top 5 Developer Sales by Genre”, x = “Developer”, y = “Global Game Sales(In millions)”)

#Overall, this code generates a polar bar chart that shows the relative contribution of each video game genre to the global sales of the top 5 video game developers. ggplot(data = df, aes(x = 0, y = developer\_glbl\_total, fill = Genre)) + geom\_col(position = “fill”) + facet\_wrap(~Developer) + coord\_polar(theta = “y”) + theme\_void() + labs(title = “Top 5 Developer sales by Genre”)

#Overall this code creates a scatter plot that shows how most categories are similar in sales. ggplot(data = df, aes(x = Genre, y = developer\_glbl\_total, colour = Developer)) + geom\_point() + labs(title = “Top 5 Developer sales by Genre”, x = “Genre”, y = “Global Game Sales(In millions)”) + coord\_flip()