Video Game Sales

Rahul Bharadwaj Mysore Venkatesh

23/08/2020

- We first load the libraries required for our analysis.

```
#loading libraries
library(tidyverse)
library(visdat)
library(kableExtra)
library(ggpubr)
```

- We now read our data into R environment from a source file

```
#reading video games sales data from csv file
vgsales <- read.csv("vgsales.csv")
#displaying dimensions of data and sample observations
glimpse(vgsales)
```

```
## Rows: 16,598
## Columns: 11
## $ Rank         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,...
## $ Name         <chr> "Wii Sports", "Super Mario Bros.", "Mario Kart Wii", "...
## $ Platform     <chr> "Wii", "NES", "Wii", "Wii", "GB", "GB", "DS", "Wii", "...
## $ Year         <chr> "2006", "1985", "2008", "2009", "1996", "1989", "2006"...
## $ Genre        <chr> "Sports", "Platform", "Racing", "Sports", "Role-Playin...
## $ Publisher    <chr> "Nintendo", "Nintendo", "Nintendo", "Nintendo", "Ninte...
## $ NA_Sales     <dbl> 41.49, 29.08, 15.85, 15.75, 11.27, 23.20, 11.38, 14.03...
## $ EU_Sales     <dbl> 29.02, 3.58, 12.88, 11.01, 8.89, 2.26, 9.23, 9.20, 7.0...
## $ JP_Sales     <dbl> 3.77, 6.81, 3.79, 3.28, 10.22, 4.22, 6.50, 2.93, 4.70,...
## $ Other_Sales  <dbl> 8.46, 0.77, 3.31, 2.96, 1.00, 0.58, 2.90, 2.85, 2.26, ...
## $ Global_Sales <dbl> 82.74, 40.24, 35.82, 33.00, 31.37, 30.26, 30.01, 29.02...
```

Initial Data Analysis:

- Initial Data Analysis is a process which helps one get a feel of the data in question. This helps us have an overview of the data and gives insights about potential Exlporatory Data Analyis (EDA).
- Initial data analysis is the process of data inspection steps to be carried out after the research plan and data collection have been finished but before formal statistical analyses. The purpose is to minimize the risk of incorrect or misleading results. Link for more info
- IDA can be divided into 3 main steps:
  - Data cleaning is the identification of inconsistencies in the data and the resolution of any such issues.
  - Data screening is the description of the data properties.
  - Documentation and reporting preserve the information for the later statistical analysis and models.

visdat

1. The visdat package in R helps us get a visual overview of the data in the form of plots. The vis_dat() function helps us get a glimpse of the data types for all variables in our dataset.

```
#Initial Data Analysis to get a feel of the dataset
visdat::vis_dat(vgsales)
```

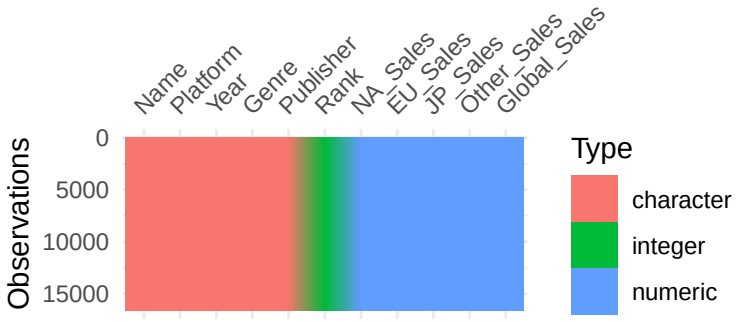

Figure 1: Visulaization of Data Types of the data

- We can observe that there are only three Types of data in our dataset viz, character, integer, and numeric. This makes it pretty straightforward and simple to conduct analysis.

visguess

2. The vis_guess() function tries to predict the kind of data in each cell of our dataset.

```
#cell data type guess
visdat::vis_guess(vgsales)
```

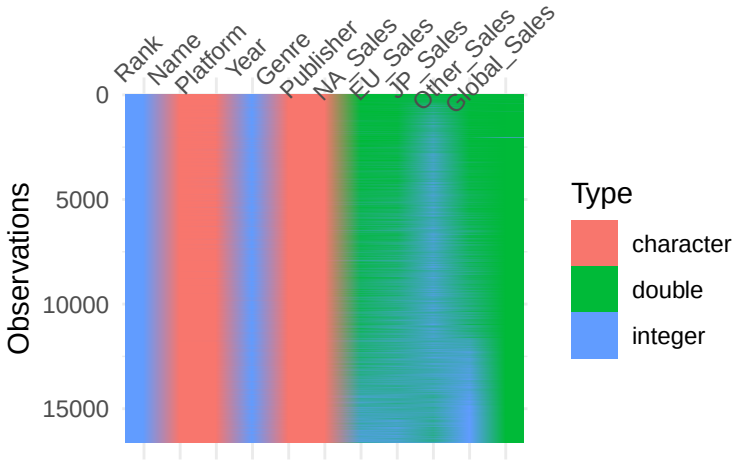

Figure 2: Data Type for each cell in dataset

- We can thus oobserve the following from the dataset.

  - Rank is integer Type.
  - Name is character Type.
  - Platform is character Type with an exception of one cell value which might have an integer.
  - Year is integer Type with some excpetion that might look like character.
  - Genre and Publisher are character Type.
  - The rest of the sales variables are either integer or double.

- Note that this is a cell-wise interpretation and the actual data will have only one type for one column.

vismiss

3. The vis_miss() function give an overall visual of the missing data in our data set.

```
#NA or missing values viz
visdat::vis_miss(vgsales)
```

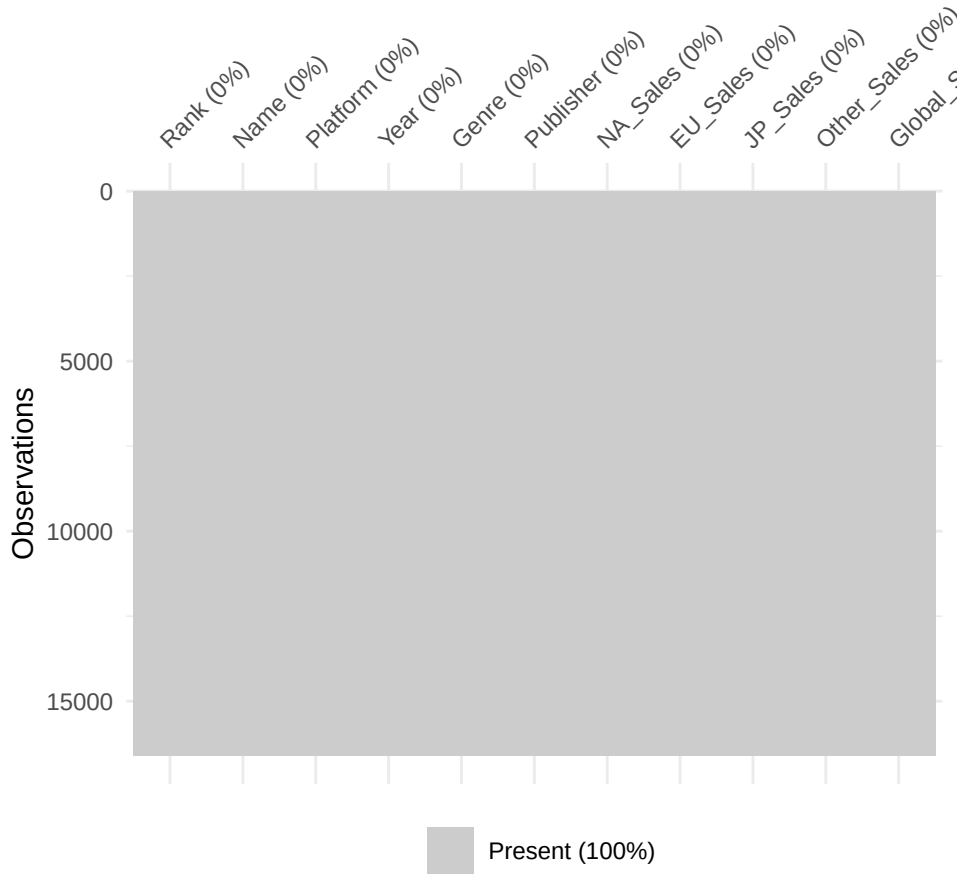

Figure 3: Missing Values Plot

- We can observe that there is no missing values in our data. Present 100% indicates the same. This means we do not have to deal with missing values.

viscor

4. The vis_cor() function gives us a visual plot of the correlation between variables in our dataset. An important thing to note here is that it takes only numeric variables. We have already established this, thanks to vis_dat(). So we select only the numeric columns for this function.

```
#visual correlation for numerical variables
visdat::vis_cor(vgsales[7:11])
```

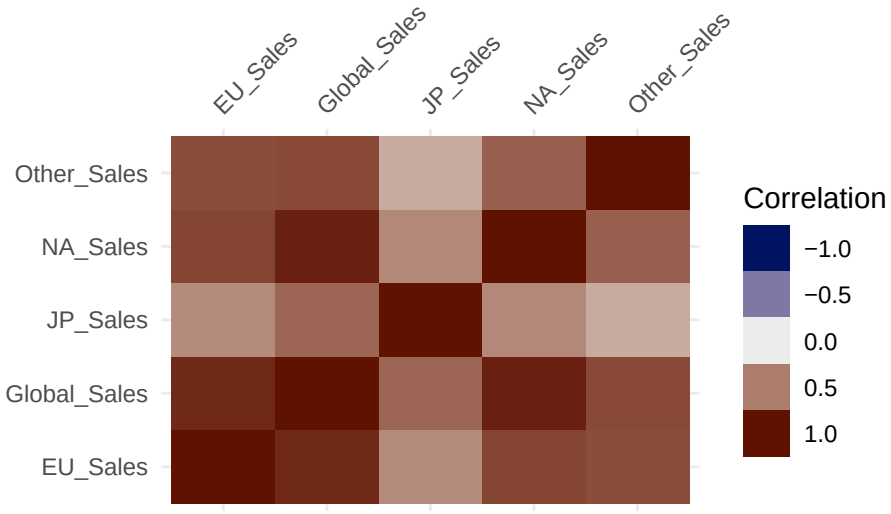

Figure 4: Correlation Plot for variables

- The above figure shows correlation as a range between +1.0 and -1.0 for the different variables.

gathercor

5. The gather_cor() function gives us the exact values for the same instead of a range.

```
#tabular correlation for numerical variables
visdat::gather_cor(vgsales[7:11]) %>% filter(row_1 == "Global_Sales") %>% head(5) %>%
  kable() %>% kable_styling(full_width = FALSE)
```

row_1	row_2	value
Global_Sales	NA_Sales	0.9410474
Global_Sales	EU_Sales	0.9028358
Global_Sales	JP_Sales	0.6118155
Global_Sales	Other_Sales	0.7483308
Global_Sales	Global_Sales	1.0000000

- The above table shows the exact correlations between the variables in our dataset. This sample table is filtered to show the correlation between Global Sales and Sales in each major locations of the world.

<

>

Exploratory Data Analysis:

- In statistics, exploratory data analysis is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task (Wikipedia)

Example Code: GTA <- vgsales %>% filter(str_detect(vgsales$Name, "Grand Theft Auto")) %>% ggplot(aes(x = reorder(Platform, -Global_Sales), y = Global_Sales, fill = Platform)) + geom_bar(stat = "identity", show.legend = FALSE) + xlab("Platform") + ylab("Global Sales") + ggtitle("Grand Theft Auto: All-Time Sales") + coord_flip()

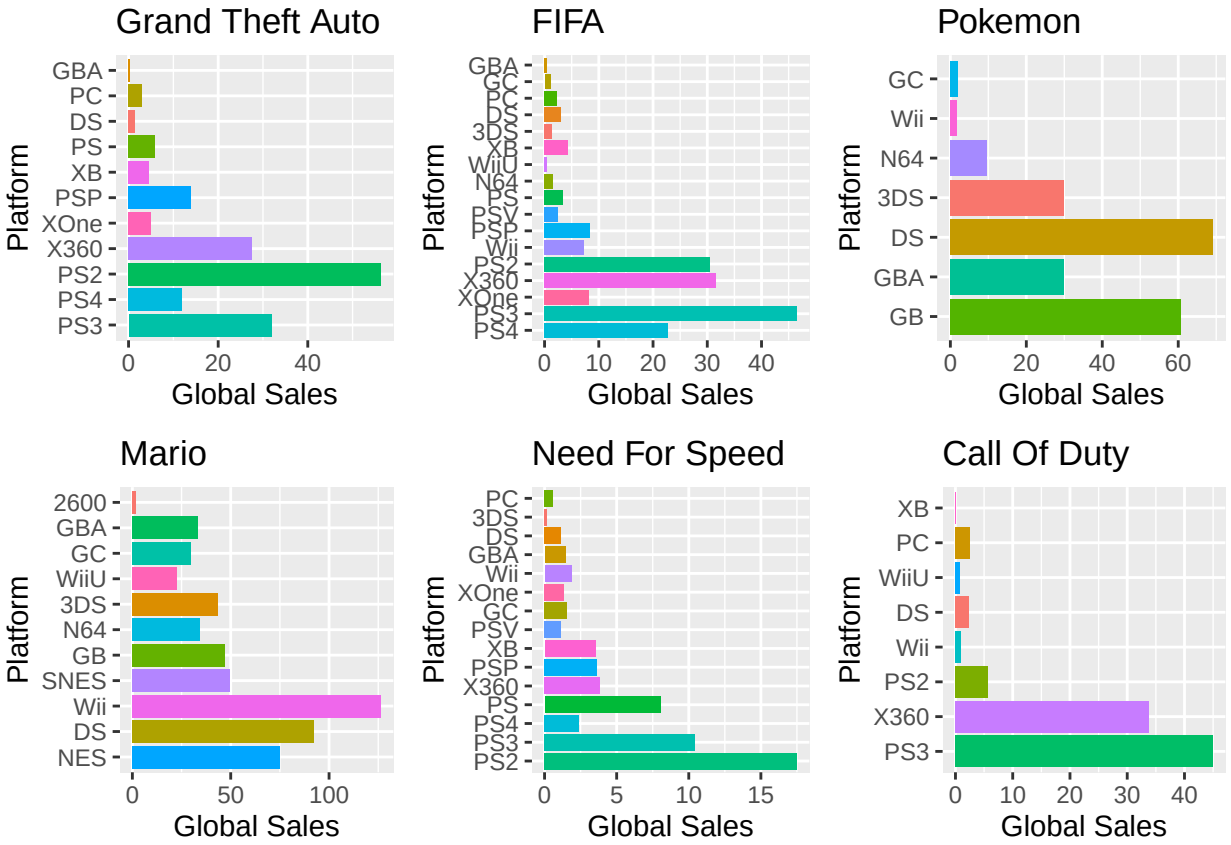

Figure 5: All Time Global Sales (millions)