12/9/2021

CA 1 – Data Exp. and Prep.

VG\_Games

Text

Description automatically generated

|  |  |
| --- | --- |
| **Module Title:** | Data Exploration and Preparation |
| **Assessment Title:** | Group Assignment |
| **Lecturer Name:** | Muhammad Iqbal |
| **Student Full Name:** | Daniel Araujo de Souza  Ricardo Alves de Souza  Carlos Santana Bobadilla |
| **Student Number:** | 2018364  2018499  2017227 |
| **Assessment Due Date:** | 11th December 2021 |
| **Date of Submission:** |  |

**Declaration**

|  |
| --- |
| By submitting this assessment, we confirm that we have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not our own or does not appropriately reference material taken from a third party or other source. We declare it to be our own work and that all material from third parties has been appropriately referenced. We further confirm that this work has not previously been submitted for assessment by ourselves or someone else in CCT College Dublin or any other higher education institution. |

Word Count: 3054 without References and tables of Contents 

Contents

[Introduction 3](#_Toc89985497)

[About the Dataset 3](#_Toc89985498)

[Fields include: 3](#_Toc89985499)

[Skimr 4](#_Toc89985500)

[a) Identify which variables are categorical, discrete and continuous in the chosen data set and show using some visualization or plot. Explore whether there are missing values for any of the variables. 4](#_Toc89985501)

[Missing Values and columns: 5](#_Toc89985502)

[b) Calculate the statistical parameters (mean, median, minimum, maximum, and standard deviation) for each of the numerical variables. 6](#_Toc89985503)

[SD 7](#_Toc89985504)

[c) Apply Min-Max Normalization, Z-score Standardization and Robust scalar on the numerical data variables. 7](#_Toc89985505)

[Min-Max 7](#_Toc89985506)

[Z-Score 8](#_Toc89985507)

[Scalar robust 8](#_Toc89985508)

[d) Scatter, Line and Heatmaps can be used to show the correlation between the features of the dataset. 9](#_Toc89985509)

[Scatter 11](#_Toc89985510)

[Line 12](#_Toc89985511)

[HeatMap 13](#_Toc89985512)

[HeatMap Plot 13](#_Toc89985513)

[e) Graphics and descriptive understanding should be provided along with Data Exploratory analysis (EDA). Identify sub-groups of features that can explore some interesting facts. 14](#_Toc89985514)

[Quantile 14](#_Toc89985515)

[Covariation between North America and Europe Sales: 15](#_Toc89985516)

[Log Transformation 17](#_Toc89985517)

[f) Apply dummy encoding to categorical variables (at least one variable use from the data set) and discuss the benefits of dummy encoding to understand the categorical data. Show the implementation of encoding scheme, such as one-hot. 18](#_Toc89985518)

[Dummy 18](#_Toc89985519)

[One – Hot Encoding 19](#_Toc89985520)

[g) Apply PCA with your chosen number of components. Write up a short profile of the first few components extracted based on your understanding. 20](#_Toc89985521)

[h) What is the purpose of dimensionality reduction? Explore the situations where you can gain the benefit of dimensionality reduction for data analysis. 25](#_Toc89985522)

[Refences 26](#_Toc89985523)

Figures

Figure 1 skimr output from vg\_dataset

Figure 2 Commands glimpse()( and str() to print relevant information about the variables

Figure 3 Inserting a new column

Figure 4 Comparison between Index and Rank Columns

Figure 5 Command to transform data 0 to NA

Figure 6 Command to copy NA data to a new dataset

Figure 7 Cleaning the main dataset from corrupt data

Figure 8 Histogram plot for variable Year, commonly used to identify outliers

Figure 9 Summary function to visualize Min, Max, Mean e Median of all variables altogether

Figure 10 Standard Deviation for NA\_Sales, EU\_Sales, JP\_Slares, Other\_Sales e Global\_Sales

Figure 11 Min - Max Normalization being applied on VG Dataset

Figure 12 Z-Score being applied on VG Dataset

Figure 13 Robust Scalar being applied on VG Dataset

Figure 14 Summary Function of VG\_Dataset normalized

Figure 15 Simple print of number row and columns

Figure 16 Created a parallel dataset to work with correlations.

Figure 17 cor(x) function do analyse correlation between variables.

Figure 18 car.test(x) function to apply the theory of Pearson and Kendall

Figure 19 corrplot being used to identify correlations

Figure 20 Scatterplot with basic line, visualizing North America Sales and Japan Sales, in Robust Scale.

Figure 21 Smooth Scatterplot visualizing North America Sales and Japan Sales, in Robust Scale.

Figure 22 Line plot analysing Europe and Japan Sales against Global Sales, in Z-score Scale.

Figure 23 Line plot analysing Europe and Japan Sales against Global Sales, in Robust Scalar

Figure 24 HeatMap plot analysing all correlations between the Sales (North America, Europe, Japan and Worldwide Sales correlating Global Sales, in Robust Scalar

Figure 25 var() function, describing all variables from the dataset

Figure 26 quantile() function, describing all numeric variables

Figure 27 BarPlot of Titles per Genre Sold

Figure 28 ggplot(), using geom\_count to analyse number of observations.

Figure 29 Bin Histogram

Figure 30 ggplot using geom\_freqpoly

Figure 31 BoxPlot of platform titles mostly bought in Japan

Figure 32 Stacked Bar Chart of Platform by Year

Figure 33 Log 10 Transformation

Figure 34 Dummy Coding performed on Genre variable

Figure 35 dcasting in reshape2 package, utilizing oneHotData

Figure 36 One-Hot Encoding, dcasting with reshape2

Figure 37 eigenvalues printed

Figure 38 Bar Plot the eigenvalues

Figure 39 Scree Plot also showing eigenvalues/variances

Figure 40 Graphic plot showing variance proportion of each variable

Figure 41 BiPlot PCA Graph of PCA variable results extracted with para a var.

Figure 42 Additional Plot to check the quality of the representation with COS2 and Graph

Figure 43 Plot for individuals

Figure 44 Control variable colours using their contributions

Figure 45 Multiplot with all numeric variables

Figure 46 Another customized plot, this time to view Individuals (ind)

Figure 47 Performs Principal Component Analysis (PCA) a Shiny application.

# 

# Introduction

For decades, video games have been around, from arcade games to home consoles and mobile devices. They are also at the forefront of computer technology.

When video games first became popular in the 1950s, they provided entertainment. People worldwide are buying new video game consoles every day, and the video game business is growing at a gigantic pace. Arcade games were addictive, and people played them for long periods.

Despite several industry failures in the 1980s, publishers managed to successfully manage the evolution of games. The industry is now one of the largest globally, managing billions of dollars annually.

Our report will address global sales of video game games for various platforms, ranging from handhelds to the most popular consoles. Our dataset powered by VGChartz\* provides us with data that spans history from 1980 to 2016.

The purpose of this project is to communicate through a report the exploration, analysis and maintenance of a dataset, performing data cleaning, noise reduction, dimensionality reduction, as well as the procedures to apply standards such as Data Exploratory Analysis (EDA) and Principal Component Analysis (PAC), among others. Moreover, to complete, we prepared a series of visual plots to discuss the scenario.

# About the Dataset

This dataset contains a list of video games with sales greater than 10,000 copies, provided by VGChartz (https://www.vgchartz.com/), and here we find information on video game sales ranging from 1980 to 2016.

## Fields include:

Rank - Ranking of overall sales

Name - The games name

Platform - Platform of the game's release (i.e. PC, PS4)

Year - Year of the game's release

Genre - Genre of the game

Publisher - Publisher of the game

NA\_Sales - Sales in North America (in millions)

EU\_Sales - Sales in Europe (in millions)

JP\_Sales - Sales in Japan (in millions)

Other\_Sales - Sales in the rest of the world (in millions)

Global\_Sales - Total worldwide sales (in millions).

## Skimr

Skimr is an excellent function, which we can see much information regarding dataset content. We can acknowledge both basic things such as data types, the number of rows and columns, Missing values, and complex information, for example, the values of p, sd, and mean, among others. See below the result of skimr on our dataset:

Text

Description automatically generated

Figure 1 skimr output from vg\_dataset.

# Identify which variables are categorical, discrete and continuous in the chosen data set and show using some visualization or plot. Explore whether there are missing values for any of the variables.

The chosen dataset is straightforward to understand, as it does not have very complex data. This table shows the types of variables found in the dataset:

|  |  |  |
| --- | --- | --- |
| Attribute | Variable | Type |
| index | Quantitative | Discrete |
| Rank | Quantitative | Discrete |
| Name | Qualitative | Nominal |
| Platform | Qualitative | Ordinal |
| Year | Qualitative | Ordinal |
| Genre | Qualitative | Ordinal |
| Publisher | Qualitative | Ordinal |
| NA\_Sales | Quantitative | Continuous |
| EU\_Sales | Quantitative | Continuous |
| JP\_Slares | Quantitative | Continuous |
| Other\_Sales | Quantitative | Continuous |
| Global\_Sales | Quantitative | Continuous |

Likewise, with the visualisation of these commands, we can see the type of variables found in the dataset:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 2 Commands glimpse()( and str() to print relevant information about the variables.

## Missing Values and columns:

At first, our dataset had 11 columns. The "Rank" column that was supposed to do the index's function, at the end not consistent enough with missing indexes and, consequently, after cleaning the dataset, we disposed of many observations in this column. Our idea was to add a new column called index which we would be able to have exact order of numbering among the most diverse items.



Figure 3 Inserting a new column.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 4 Comparison between Index and Rank Columns

At this point, our dataset was already reasonably cleaned, and it was only necessary to make some changes, for example, to years counted at 0, and migrate all the faulty observations to another Dataset (NAs\_vg\_dataset):



Figure 5 Command to transform data 0 to NA.



Figure 6 Command to copy NA data to a new dataset.



Figure 7 Cleaning the primary dataset from corrupt data.

Finishing with the visualisation of a Histogram to ensure that the Year variable is within the expected pattern (1980~2016). Using the command: > hist(vg\_dataset$Year).

As we can see below, our Year column is adequate.

Chart, histogram

Description automatically generated

Figure 8 Histogram plot for variable Year, commonly used to identify outliers

# Calculate the statistical parameters (mean, median, minimum, maximum, and standard deviation) for each of the numerical variables.

We use the summary() command to solve this question, which brings us information about Min, Max, Mean and Median of each numeric column. (These columns must be clean of incorrect data.)

A picture containing text, screenshot, battery

Description automatically generated

Figure 9 Summary function to visualize Min, Max, Mean and Median of all variables altogether.

## SD

o calculate Standard Deviation, we need the desirable columns to be numeric and use the sd() command. In our case, we will calculate for NA\_Sales, EU\_Sales, JP\_Slares, Other\_Sales and Global\_Sales:

Text

Description automatically generated with medium confidence

Figure 10 Standard Deviation for NA\_Sales, EU\_Sales, JP\_Slares, Other\_Sales e Global\_Sales.

# Apply Min-Max Normalization, Z-score Standardization and Robust scalar on the numerical data variables.

The application of normalisations and scaling standards is very straightforward and objective. We need to create some functions to calculate numerical data and scale both. We used the variables NA\_Sales, EU\_Sales, JP\_Slares, Other\_Sales, and Global\_Sales.

## Min-Max

Data normalisation is a process of transforming outliers using a scale ranging from 0.0 for the smallest to 1.0 for the most significant value. Below is the code for Min-Max Normalization:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 11 Min - Max Normalization being applied on VG Dataset.

## Z-Score

The z-score is the number of standard deviations from the mean of an information point. In addition, is a proportion of the number of standard deviations, below or above the score, which means a raw score. Our graph, however, allows us to have a cleaner and more scaled perception of comparison between sales variables.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 12 Z-Score being applied on VG Dataset.

## Scalar robust

It is similar to normalisation but uses the interquartile range to robustly extreme values. But it does not consider the median and focuses only on the bulk data parts.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 13 Robust Scalar being applied on VG Dataset.

After implementing all these columns our dataset got a little wider with 27 variables. See this summary of all of them:

A picture containing calendar

Description automatically generated

Figure 14 Summary Function of VG\_Dataset normalized.

After cleaning and applying the standards, our dataset has 27 columns and 16295.

Text

Description automatically generated with medium confidence

Figure 15 Simple print of number row and columns.

# Scatter, Line and Heatmaps can be used to show the correlation between the features of the dataset.

For this activity, we first separate only the data we want to work with, in this case, the variables NA\_Sales, EU\_Sales, JP\_Slares, Other\_Sales and Global\_Sales. We create a new dataset (corr\_data) and start some functions to analyse the correlation of the data before viewing the plots. See below:

Text

Description automatically generated

Figure 16 Created a parallel dataset to work with correlations.

Next, let's call the color(x) function to see how correlated our variables are.

Text, application, chat or text message

Description automatically generated

Figure 17 cor(x) function do analyse correlation between variables.

We go a little further and call a function that will apply the Pearson and Kendall-specific formula:

Text

Description automatically generated

Figure 18 car.test(x) function to apply the theory of Pearson and Kendall.

And to finish our test, we used corrplot to demonstrate in colours and numbers the correlation between these variables:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 19 corrplot being used to identify correlations.

## Scatter

A Scatter Plot with a Trend Line using functions (geom\_point() and geom\_abline()), great to identify covariations;

A screenshot of a computer

Description automatically generated with medium confidence

Figure 20 Scatterplot with basic line, visualizing North America Sales and Japan Sales, in Robust Scale.

A Scatter Plot with a Trend Line using the functions (geom\_point(), geom\_abline() and geom\_smooth());

A screenshot of a computer

Description automatically generated with medium confidence

Figure 21 Smooth Scatterplot visualizing North America Sales and Japan Sales, in Robust Scale.

## Line

Line Plot Scaled: We compare Europe and Japan against Global Sales using Z-score variables that we just created earlier.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 22 Line plot analysing Europe and Japan Sales against Global Sales, in Z-score Scale.

Line Plot Robust Scalar. As the Plot above, we are comparing the same sales using Robust Scalar variables.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 23 Line plot analysing Europe and Japan Sales against Global Sales, in Robust Scalar.

## HeatMap

## 

A two-dimensional data visualisation technique depicts a phenomenon's magnitude as colour. The colour fluctuation might be via hue or intensity, giving the reader clear visual indications about how the occurrence is clustered or evolves. See the Heatmap of our table:

A screenshot of a computer

Description automatically generated

Figure 24 HeatMap plot analysing all correlations between the Sales (North America, Europe, Japan and Worldwide Sales correlating Global Sales, in Robust Scalar.

# Graphics and descriptive understanding should be provided along with Data Exploratory analysis (EDA). Identify sub-groups of features that can explore some interesting facts.

Throughout the project, we mentioned several of the many explorations, visualisation and normalisation processes, such as Mean, Median values, variable types (str), and standard deviation. But to complete our EDA we still need some key factors for the process.

Var(vg\_dataset), this function brings us the variation between all the columns of the table:

A computer screen capture

Description automatically generated with medium confidence

Figure 25 var() function, describing all variables from the dataset.

## Quantile

Quantile(), we used to show information about the numeric columns (NA\_Sales, EU\_Sales, JP\_Slares, Other\_Sales e Global\_Sales)

Text

Description automatically generated

Figure 26 quantile() function, describing all numeric variables .

BarPlot showing the number of Titles sold per Genre.

Chart, bar chart

Description automatically generated

Figure 27 BarPlot of Titles per Genre Sold.

## Covariation between North America and Europe Sales:

Using gem\_count(), The size of each circle in the plot displays how many observations occurred at each combination:

Graphical user interface

Description automatically generated

Figure 28 ggplot(), using geom\_count to analyse number of observations.

Bin Histogram of the variable Year:

Chart, histogram

Description automatically generated

Figure 29 Bin Histogram.

Covariation between Europe Sales and Genre:

Chart

Description automatically generated

Figure 30 ggplot using geom\_freqpoly.

BoxPlot illustrating a covariation of platform titles bought in Japan:

Chart

Description automatically generated

Figure 31 BoxPlot of platform titles mostly bought in Japan.

Stacked Bar Chart of Platform by Year, we can see the evolution of platform throughout the years:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 32 Stacked Bar Chart of Platform by Year.

## Log Transformation

Log 10 Transformation on the numeric variables.

Text

Description automatically generated

Figure 33 Log 10 Transformation.

# Apply dummy encoding to categorical variables (at least one variable use from the data set) and discuss the benefits of dummy encoding to understand the categorical data. Show the implementation of encoding scheme, such as one-hot.

## Dummy

Here we have the Dummy table’s variables created by the function dummy\_cols():

A screenshot of a computer

Description automatically generated

Figure 34 Dummy Coding performed on Genre variable.

A dummy variable is a variable that indicates whether an observation contains a specific feature or not. A dummy variable can only take 0 and 1, with 0 indicating the property's absence and 1 indicating its presence. The 0/1 values can be interpreted as no/yes or off/on. In modelling, using dummy variable will allow capturing the difference in the expected value between categories, that is, the coefficient (Beta) of the model will be the average value that a given category represents.

Now, One Hot Encoding has a lot to do with Dummy. Process by which each of the levels of a categorical variable is transformed into a column, filled with '0' and '1' depending on the presence or absence of it, likely dummy, we can see some differentiation between those methods:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 35 dcasting in reshape2 package, utilizing oneHotData.

The Data Script is available on the project’s folder, For both visualizations the code used was.

library(magrittr)

library(caret)

library(mltools)

library(data.table)

oneHotData <- subset(vg\_dataset,select = c(Index, Genre, Year))

newData <- one\_hot(as.data.table(oneHotData))

#dummyVars in caret package

dummy <- dummyVars(" ~ .", data=oneHotData)

newdata <- data.frame(predict(dummy, newData = oneHotData))

#dcast in reshape2 package

library(reshape2)

newdata <- dcast(data = oneHotData, Year ~ Genre, length)

## One – Hot Encoding

Encode utput for GenreStrategy variable (produced by DummyVars method):

A screenshot of a computer

Description automatically generated with medium confidence

Figure 36 One-Hot Encoding, dcasting with reshape2.

# Apply PCA with your chosen number of components. Write up a short profile of the first few components extracted based on your understanding.

To start with PCA, we first perform PCA Generation and following we can extract the proportion of variance from principal component values.

Text

Description automatically generated

Figure 37 eigenvalues printed.

In this first view, the barplot shows us the proportion of the eigenvalues, with these values we can use them to present our PCA graph:

Chart

Description automatically generated

Figure 38 Bar Plot the eigenvalues.

Scree Plot performing practically the same task as the barplot above, but this one we can see the actual percentage marked:

Graphical user interface

Description automatically generated

Figure 39 Scree Plot also showing eigenvalues/variances.

Graphic plot showing graph of PCA data, utilizing fviz() method:

A computer screen capture

Description automatically generated with medium confidence

Figure 40 Graphic plot showing variance proportion of each variable.

Biplot to graph the results of the PCA variables extracted in the previous step for the var:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 41 BiPlot PCA Graph of PCA variable results extracted with para a var.

Additional corrplot to validate COS2 representation quality using graph:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 42 Additional Plot to check the quality of the representation with COS2 and Graph.

Simple Plot do show Individual’s variable (PCA):

Graphical user interface, chart

Description automatically generated

Figure 43 Plot for individuals.

Control variable colours using their contributions, use red and blue colours to identify lower and high scale:

Graphical user interface

Description automatically generated

Figure 44 Control variable colours using their contributions.

Multiple plot function, to visualize in overall their relationship with PCA:

Graphical user interface

Description automatically generated

Figure 45 Multiplot with all numeric variables.

Fviz\_pca\_ind() Plot show us the relation of the variable (genre) in groups, based on the PCA individual:

A screenshot of a computer

Description automatically generated with medium confidence

Figure 46 Another customized plot, this time to view Individuals (ind).

An additional type of Plot, we bootstrap a simpler GUI interface to work with our PCA elements. Used the function PCAshiny(scaled\_vg), Package 'Factoshiny':

A screenshot of a computer

Description automatically generated with medium confidence

Figure 47 Performs Principal Component Analysis (PCA) a Shiny application.

# What is the purpose of dimensionality reduction? Explore the situations where you can gain the benefit of dimensionality reduction for data analysis.

With the horizontal increase of databases (dimensions/attributes), the increase in dimensionality (Course of Dimensionality) is a serious problem nowadays, that we have not only multicollinearity but heteroskedasticity and self-correction to use simple statistical examples. In computational terms, there is no doubt that the increase in attributes makes the Data Mining or Computational Intelligence algorithms must process a much larger volume of data (increased processing complexity = higher time cost).

There are many factors that make you apply dimensionality rules, but here we will briefly explain some of them:

* Quite so many missing values in a data column are unlikely to provide much relevant information. As a result, data columns having a higher number of missing values than a certain threshold can be eliminated. The more aggressive the reduction, the higher the threshold.
* The Low Variance Filter is a filter that reduces the amount of variation in a signal. Data columns with little changes in the data, like the preceding technique, carry little information. As a result, any data columns with variance below a certain threshold are deleted. A word of caution: because variance is range dependent, this approach must be normalized before use.
* Filter for high correlation. Data variables with related trends are more likely to contain similar data. In this scenario, only one of these will be enough to offer a good input for the machine learning model. The methods Pearson's chi-square value and the Pearson's Product Moment Coefficient are used to calculate the correlation coefficient between nominal and numerical columns. The number of columns in a pair with a correlation coefficient greater than a threshold is reduced to just one. An observation: correlation is scale sensitive. To have a plausible correlation comparison, it's necessary to have variables' normalisation.
* Ensemble Trees and Random Forests, in addition to being effective classifiers, Decision Tree Ensembles, often known as random forests, are beneficial for feature selection. One method for reducing dimensionality is to create a large, carefully designed set of trees against a target property, then utilize the usage statistics for each variable to discover the most informative subset of features. We can create a big number of extremely shallow trees (2 levels) with each tree trained on a small percentage (3) of the entire number of characteristics, resulting in many trees (2000). If an attribute is frequently chosen as the best split, it is most likely a useful trait to keep. A score based on the random forest's attribute usage data shows us which characteristics are the most predictive of the others.
* The statistical process Principal Component Analysis (PCA) quadratically transforms the original n coordinates of a data set into a new set of n coordinates called principal components. The first principal component has the highest possible variance because of the transformation; each subsequent component has the maximum possible variance under the constraint of being orthogonal to (i.e., uncorrelated with) the preceding components. By maintaining only, the first m n components, the data dimensionality is reduced while most of the data information, i.e. the variance in the data, is retained. It's worth noting that the PCA transformation is sensitive to the original variables' relative scale. Before using PCA, the data column ranges must be standardized. It's also worth noting that the new coordinates (PCs) are no longer true system-generated variables. When you use PCA to analyse your data, it loses its interpretability and If the results are significant, it could a problem.
* Backward Feature Elimination is a technique for removing features that have been used in the past. At each iteration, the specified classification algorithm is trained on n input features in this technique. Thus, one by one, we eliminate input features and train the same model on n-1 input features n times. We delete the input feature that caused the smallest increase in the error rate, leaving us with n-1 input features. Following that, the classification is repeated using n-2 features, and so on. Each iteration k yields a model with n-k features and an error rate of e. (k). We specify the fewest features required to achieve that classification performance with the chosen machine learning method by selecting the maximum allowable error rate.
* Feature Construction. The opposite of Feature Elimination. We begin with just one feature, gradually adding one feature at a time, i.e., the one that results in the greatest boost in performance. Feature Elimination and Construction are both time-consuming and computationally intensive. They're practically only useful for data sets with a small number of input columns, to begin with.

# Refences

Johnson, D. (2021, October 7). Factor in R: Categorical Variable & Continuous Variables. Retrieved December 9, 2021, from <https://www.guru99.com/r-factor-categorical-continuous.html>

robk@statmethods.net, R. K.-. Date Values. Retrieved December 9, 2021, from <https://www.statmethods.net/input/dates.html>

Lathiya, K. (2021, November 26). Standard deviation in R: How to Use sd() in R. Retrieved December 9, 2021, from <https://r-lang.com/how-to-calculate-standard-deviation-in-r-using-sd/>

Kalpana, N. S. S. (2020, July 6). Data Normalisation With R. Retrieved December 9, 2021, from <https://medium.com/swlh/data-normalisation-with-r-6ef1d1947970>

How to Deal with Missing Values in R. Retrieved December 9, 2021, from <https://datascienceplus.com/missing-values-in-r/>

R is.na Function Example (remove, replace, count, if else, is not NA). (2020, October 1). Retrieved December 9, 2021, from <https://statisticsglobe.com/r-is-na-function/>

Kataruka, A. (2019, October 7). Adding manual legend to ggplot2. Retrieved December 9, 2021, from <https://community.rstudio.com/t/adding-manual-legend-to-ggplot2/41651/3>

Get year from date in R. (2019, October 20). Retrieved December 9, 2021, from <https://www.datasciencemadesimple.com/get-year-from-date-in-r-2/>

Outliers detection in R. Retrieved December 9, 2021, from <https://statsandr.com/blog/outliers-detection-in-r/>

Mean, Median and Mode. Retrieved December 9, 2021, from <https://www.tutorialspoint.com/r/r_mean_median_mode.htm>

Holtz, Y. Basic R barplot customization. Retrieved December 9, 2021, from <https://www.r-graph-gallery.com/209-the-options-of-barplot.html>

Holtz, Y. Building heatmap with R. Retrieved December 9, 2021, from <https://www.r-graph-gallery.com/215-the-heatmap-function.html>

Retrieved December 9, 2021, from <https://stat.ethz.ch/pipermail/r-help/2010-January/224663.html>

One-hot encoding in R: three simple methods. (2020, June 29). Retrieved December 9, 2021, from <https://datatricks.co.uk/one-hot-encoding-in-r-three-simple-methods>

Roberts, J. (2020, February 14). Reshape2- dcast problem. Retrieved December 9, 2021, from <https://community.rstudio.com/t/reshape2-dcast-problem/52920/10>

let&&chill, One Hot Encoding in R, 8 de ago. de 2021, from <https://www.youtube.com/watch?v=1WH0Q4EVJ7o&ab_channel=let%26%26chill>

Erik Marsja, 3, E. M. on S., 21, E. M. on O., & \*, N. (2021, April 15). How to Create Dummy Variables in R (with Examples). Retrieved December 9, 2021, from <https://www.marsja.se/create-dummy-variables-in-r/>