AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

Course:

INTRODUCTION TO DATA SCIENCE

Fall 2025-2026
Section: D, Group: 3

Title:

COVID-19 Country-wise Data Analysis

Supervised By

Tohedul Islam

Submitted By

NAME	ID
Efty, Md. Emran Nazir	22-47802-2
Sudipta Niloy Sarker	22-46657-1
Khondaker Iffti Hasan Turjo	22-46146-1
Md. Jahidul Hoque	22-46754-1

Date of Submission:

December 02, 2025

Introduction:

This report details the analysis of COVID-19 country-wise, which contains country-specific COVID-19 attributes. The dataset includes 15 key features such as country, confirmed, deaths, recovered, active, new cases, new deaths, new recovered, Deaths 100 Cases, Confirmed last week, 1 week change, 1 week % increase, 1 week % decrease, and WHO Region.

Dataset Source:

Dataset Name: COVID-19 Dataset

Source: Kaggle(https://www.kaggle.com/datasets/imdevskp/corona-virus-report)

Dataset Creation& Description:

The COVID-19 Dataset, sourced from Kaggle contains detailed statistics on COVID-19 cases for countries and regions worldwide. It includes information such as confirmed cases, deaths, recovered cases, and active cases. Containing 187 rows and 10 columns, the dataset records COVID-19 data for individual countries and regions. This dataset provides valuable insights for analyzing, visualizing, and modeling the spread and impact of COVID-19 globally.

Description of Features:

Variable	Description
Country	Name of the country or region
Confirmed	Total number of confirmed COVID-19 cases
Deaths	Total number of deaths due to COVID-19
Recovered	Total number of patients who recovered
Active	Total number of active cases
New cases	New confirmed cases reported on the last day
New deaths	New deaths reported on the last day
New recovered	New recovered cases reported on the last day
Deaths / 100 Cases	Mortality rate = (Deaths ÷ Confirmed) × 100
Recovered / 100 Cases	Recovery rate = (Recovered ÷ Confirmed) × 100
Deaths / 100 Recovered	Deaths per 100 recovered cases
Confirmed last week	Number of confirmed cases reported in the last week
1 week change	Change in confirmed cases over the past week
1 week % increase	Percentage increase in confirmed cases over the past week
WHO Region	WHO-defined region of the country.

Dataset into R:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Country/R	Confirmed	Deaths	Recovered	Active	New cases	New death	New recov	Deaths / 1(	Recovered	Deaths / 1(	Confirmed	1 week cha	1 week % i	WHO Region		
2	Afghanista	36263	1269	25198	9796	106	10	18	3.5	69.49	5.04	35526	737	2.07	Eastern Mediterranean		
3	Albania	4880	144	2745	1991	117	6	63	2.95	56.25	5.25	4171	709	17	Europe		
4	Algeria	27973	1163	18837	7973	616	8	749	4.16	67.34	6.17	23691	4282	18.07	Africa		
5	Andorra	907	52	803	52	10	0	0	5.73	88.53	6.48	884	23	2.6	Europe		
6	Angola	950	41	242	667	18	1	0	4.32	25.47	16.94	749	201	26.84	Africa		
7	Antigua an	86	3	65	18	4	0	5	3.49	75.58	4.62	76	10	13.16	Americas		
8	Argentina	167416	3059	72575	91782	4890	120	2057	1.83	43.35	4.21	130774	36642	28.02	Americas		
9	Armenia	37390	711	26665	10014	73	6	187	1.9	71.32	2.67	34981	2409	6.89	Europe		
10	Australia	15303	167	9311	5825	368	6	137	1.09	60.84	1.79	12428	2875	23.13	Western Pacific		
11	Austria	20558	713	18246	1599	86	1	37	3.47	88.75	3.91	19743	815	4.13	Europe		
12	Azerbaijan	30446	423	23242	6781	396	6	558	1.39	76.34	1.82	27890	2556	9.16	Europe		
13	Bahamas	382	11	91	280	40	0	0	2.88	23.82	12.09	174	208	119.54	Americas		
14	Bahrain	39482	141	36110	3231	351	1	421	0.36	91.46	0.39	36936	2546	6.89	Eastern Mediterranean		
15	Banglades	226225	2965	125683	97577	2772	37	1801	1.31	55.56	2.36	207453	18772	9.05	South-East Asia		
16	Barbados	110	7	94	9	0	0	0	6.36	85.45	7.45	106	4	3.77	Americas		
17	Belarus	67251	538	60492	6221	119	4	67	0.8	89.95	0.89	66213	1038	1.57	Europe		
18	Belgium	66428	9822	17452	39154	402	1	14	14.79	26.27	56.28	64094	2334	3.64	Europe		
19	Belize	48	2	26	20	0	0	0	4.17	54.17	7.69	40	8	20	Americas		
20	Benin	1770	35	1036	699	0	0	0	1.98	58.53	3.38	1602	168	10.49	Africa		
21	Bhutan	99	0	86	13	4	0	1	0	86.87	0	90	9	10	South-East Asia		
22	Bolivia	71181	2647	21478	47056	1752	64	309	3.72	30.17	12.32	60991	10190	16.71	Americas		
23	Bosnia anc	10498	294	4930	5274	731	14	375	2.8	46.96	5.96	8479	2019	23.81	Europe		
24	Botswana	739	2	63	674	53	1	11	0.27	8.53	3.17	522	217	41.57	Africa		
25	Brazil	2442375	87618	1846641	508116	23284	614	33728	3.59	75.61	4.74	2118646	323729	15.28	Americas		
26	Brunei	141	3	138	0	0	0	0	2.13	97.87	2.17	141	0	0	Western Pacific		

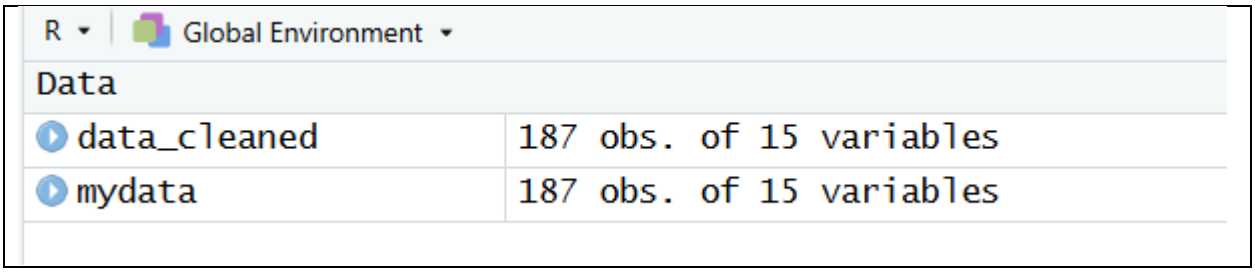
Data Cleaning Process:

- Checked for missing values and handled them appropriately (replace, median, mean, mode)
- Remove rows with any missing values
- Duplicate Row Check and Removal

Empty string replace:

```
data_cleaned <-mydata
data_cleaned$Country.Region[data_cleaned$Country.Region == ""] <- NA
data_cleaned
```

```
> data_cleaned$Country.Region[data_cleaned$Country.Region == ""] <- NA
> data_cleaned
           Country.Region Confirmed Deaths Recovered Active New.cases New.deaths
1             Afghanistan     36263   1269     25198   9796       106         10
2                 Albania      4880    144      2745   1991       117         NA
3                 Algeria     27973   1163     18837   7973       616          8
4                 Andorra       907     52       803     52        10          0
5                  Angola       950     41       242    667        18          1
6      Antigua and Barbuda        86     NA        65     18         4          0
7               Argentina    167416   3059     72575  91782      4890        120
8                 Armenia     37390    711     26665  10014        73          6
9               Australia     15303    167      9311   5825       368          6
10                Austria     20558    713     18246   1599        86          1
11             Azerbaijan     30446    423     23242   6781       396          6
12                Bahamas       382     11        91    280        40          0
13                Bahrain     39482    141     36110   3231       351          1
14             Bangladesh    226225   2965    125683  97577      2772         37
```



Description:

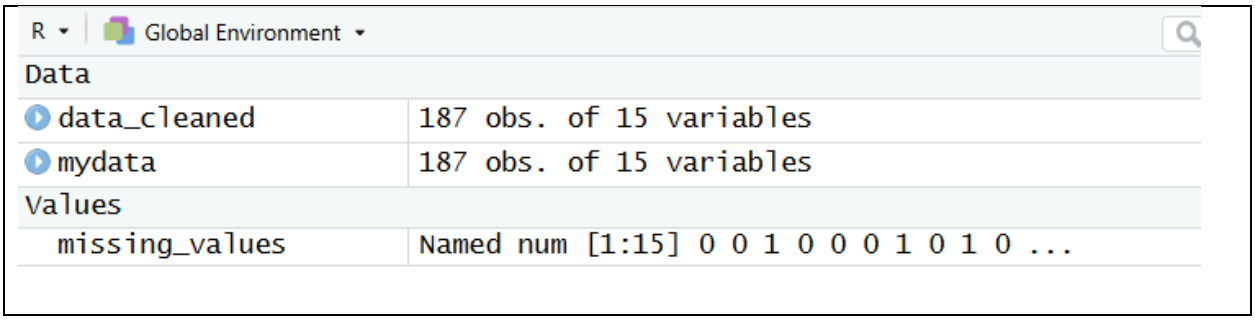
In the original dataset (mydata), some entries in the Country.Region column were empty (""). Empty strings can cause issues during data analysis, such as counting missing values, grouping, or visualizations. To clean the data, all empty strings in the Country.Region column were replaced with NA (missing value indicator) using the following code. data_cleaned now contains 187 observations (rows) and 15 variables (columns).

Any empty Country.Region entries are replaced with NA, which allows proper handling of missing values in later steps like median/mean/mode imputation, filtering, and visualization. The dataset is now cleaner and ready for further preprocessing.

Missing Value:

```
missing_values <- colSums(is.na(data_cleaned))
missing_values
if(any(missing_values > 0)) {
  print(missing_values[missing_values > 0])
} else {
  cat("No missing values found in the dataset.\n\n")
}
```

```
> missing_values
       Country.Region             Confirmed                Deaths             Recovered
                    0                     0                     1                     0
               Active             New.cases            New.deaths          New.recovered
                    0                     0                     1                     0
   Deaths...100.Cases Recovered...100.Cases Deaths...100.Recovered    Confirmed.last.week
                    1                     0                     0                     1
       X1.week.change     X1.week...increase            WHO.Region
                    0                     0                     0
> if(any(missing_values > 0)) {
+   print(missing_values[missing_values > 0])
+ } else {
+   cat("No missing values found in the dataset.\n\n")
+ }
            Deaths          New.deaths  Deaths...100.Cases Confirmed.last.week
                 1                   1                   1                   1
```



Description:

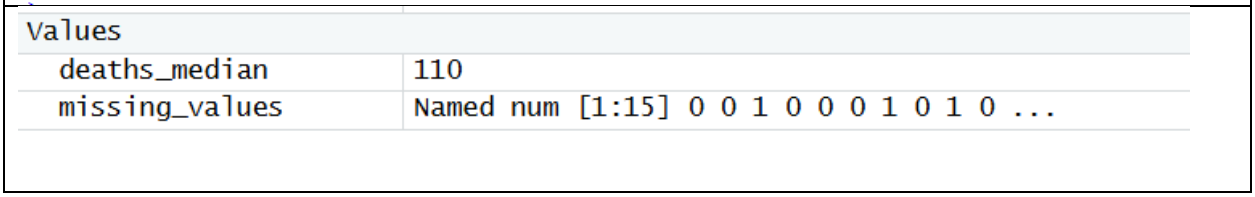
In the original dataset (mydata), most of the data was complete. After checking with colSums(is.na(data_cleaned)), only four columns showed missing values: Deaths, New deaths, Deaths / 100 Cases, and Confirmed last week. Each of these columns had exactly one missing entry. All other columns had no missing values across the 187 rows. This very small amount of missing data was noted so it could be properly handled in the next steps. The dataset still has 187 observations (rows) and 15 variables (columns).

Handling Missing Values

Replaced by Median:

```
median_replace<-mydata
median_replace[is.na(median_replace$Deaths), ]
deaths_median <- median(median_replace$Deaths, na.rm = TRUE)
print(deaths_median)

median_replace$Deaths[is.na(median_replace$Deaths)] <- deaths_median
median_replace[is.na(median_replace$Deaths), ]
median_replace
```

```
       Country.Region Confirmed Deaths Recovered Active New.cases New.deaths New.recovered
6 Antigua and Barbuda        86     NA        65     18         4          0              5
  Deaths...100.Cases Recovered...100.Cases Deaths...100.Recovered Confirmed.last.week
6               3.49                 75.58                  4.62                  76
  X1.week.change X1.week...increase WHO.Region
6             10              13.16   Americas
> deaths_median <- median(median_replace$Deaths, na.rm = TRUE)
> print(deaths_median)
[1] 110
```



Description:

In the dataset, the Deaths column had one missing value. We calculated the median of all existing Deaths values, which was 110, and used this number to fill the empty spot. After this step, the Deaths column had no missing values anymore. Using the median is a simple and safe way because it is not affected by very high or low numbers. The dataset still has 187 observations (rows) and 15 variables (columns).

Replaced by Mean:

```
mean_add<-median_replace
mean_add[is.na(mean_add$Confirmed), ]

Confirmed_mean <- mean(mean_add$Confirmed, na.rm = TRUE)
print(Confirmed_mean)
mean_add$Confirmed[is.na(mean_add$Confirmed)] <- Confirmed_mean
mean_add[is.na(mean_add$Confirmed), ]
mean_add
```

```
> mean_add[is.na(mean_add$Confirmed), ]
 [1] Country.Region         Confirmed              Deaths
 [4] Recovered              Active                 New.cases
 [7] New.deaths             New.recovered          Deaths...100.Cases
[10] Recovered...100.Cases  Deaths...100.Recovered Confirmed.last.week
[13] X1.week.change         X1.week...increase     WHO.Region
<0 rows> (or 0-length row.names)
>
> Confirmed_mean <- mean(mean_add$Confirmed, na.rm = TRUE)
> print(Confirmed_mean)
[1] 88130.94
```

Values	
Confirmed_mean	88130.935828877
deaths_median	110
missing_values	Named num [1:15] 0 0 1 0 0 0 1 0 1 0 ...

Description:

In the dataset, the Confirmed column had one missing value. We first checked which row was affected and then calculated the average (mean) of all existing Confirmed cases, ignoring the missing entry. The result was 88,130.94. This mean value was used to fill the single empty spot in the Confirmed column. After replacement, we again checked and confirmed that there were no more missing values in this column. Using the mean is a simple and effective method when only a few values are missing, as it keeps the overall average of confirmed cases almost unchanged. The dataset continues to have 187 observations (rows) and 15 variables (columns).

Replaced by Mode:

```
mode_add<-mean_add
mode_add[is.na(mode_add$WHO.Region),]
freq_table <- table(mode_add$WHO.Region)
mode<-names(freq_table)[which.max(freq_table)]
mode_add$WHO.Region[is.na(mode_add$WHO.Region)] <- mode
mode_add[is.na(mydata$WHO.Region),]
mode_add
```

```
> mode_add[is.na(mydata$WHO.Region),]
 [1] Country.Region         Confirmed              Deaths
 [4] Recovered              Active                 New.cases
 [7] New.deaths             New.recovered          Deaths...100.Cases
[10] Recovered...100.Cases  Deaths...100.Recovered Confirmed.last.week
[13] X1.week.change         X1.week...increase     WHO.Region
<0 rows> (or 0-length row.names)
> mode_add
             Country.Region Confirmed Deaths Recovered Active New.cases New.deaths
1               Afghanistan     36263   1269     25198   9796       106         10
2                   Albania      4880    144      2745   1991       117         NA
3                   Algeria     27973   1163     18837   7973       616          8
4                   Andorra       907     52       803     52        10          0
5                    Angola       950     41       242    667        18          1
6       Antigua and Barbuda        86    110        65     18         4          0
7                 Argentina    167416   3059     72575  91782      4890        120
```

Values	
Confirmed_mean	88130.935828877
deaths_median	110
freq_table	'table' int [1:6(1d)] 48 35 22 56 10 16
missing_values	Named num [1:15] 0 0 1 0 0 0 1 0 1 0 ...
mode	"Europe"

Description:

The WHO.Region column had some missing entries. We first checked the missing rows and then created a frequency table of all existing regions. The most frequent region was “Europe”, appearing 56 times, so it was chosen as the mode. All missing values in the WHO.Region column were replaced with “Europe”. After this step, the WHO.Region column became fully complete with no missing values left. Using the mode is the best choice for categorical data because it keeps the most common category dominant. The dataset still has 187 observations (rows) and 15 variables (columns).

Remove rows with any missing values:

```
Remove_missing_row<-mode_add
Remove_missing_row<-na.omit(Remove_missing_row)
Remove_missing_row
```

```
> Remove_missing_row
              Country.Region Confirmed Deaths Recovered Active New.cases New.deaths
1                Afghanistan     36263   1269     25198   9796       106         10
3                    Algeria     27973   1163     18837   7973       616          8
4                    Andorra       907     52       803     52        10          0
5                     Angola       950     41       242    667        18          1
6        Antigua and Barbuda        86    110        65     18         4          0
7                  Argentina    167416   3059     72575  91782      4890        120
8                    Armenia     37390    711     26665  10014        73          6
9                  Australia     15303    167      9311   5825       368          6
10                   Austria     20558    713     18246   1599        86          1
11                Azerbaijan     30446    423     23242   6781       396          6
12                   Bahamas       382     11        91    280        40          0
```

Data	
data_cleaned	187 obs. of 15 variables
mean_add	187 obs. of 15 variables
median_replace	187 obs. of 15 variables
mode_add	187 obs. of 15 variables
mydata	187 obs. of 15 variables
Remove_missing_row	185 obs. of 15 variables

Description:

Another common technique to handle missing values is to simply remove any row that contains at least one missing entry. We applied the na.omit() function to the dataset after the previous imputation steps. This removed all incomplete rows in. As a result, the dataset went from 187 rows to 185 rows. This method is very quick and guarantees that the remaining data is 100% complete across all columns. The final cleaned dataset now contains 185 observations (rows) and 15 variables (columns) with no missing values.

Duplicate Row Check and Removal :

```
duplicated_row <- Remove_missing_row
duplicate_rows <- sum(duplicated(duplicated_row))
if(duplicate_rows > 0) {
  cat("Found", duplicate_rows, "duplicate rows. Removing them.\n")
  Remove_missing_row <- Remove_missing_row[!duplicated(Remove_missing_row),
]
} else {
  cat("No duplicate rows found.\n\n")
}
nrow(duplicated_row)
```

```
No duplicate rows found.

> nrow(duplicated_row)
[1] 185
```

Data	
data_cleaned	187 obs. of 15 variables
duplicated_row	185 obs. of 15 variables

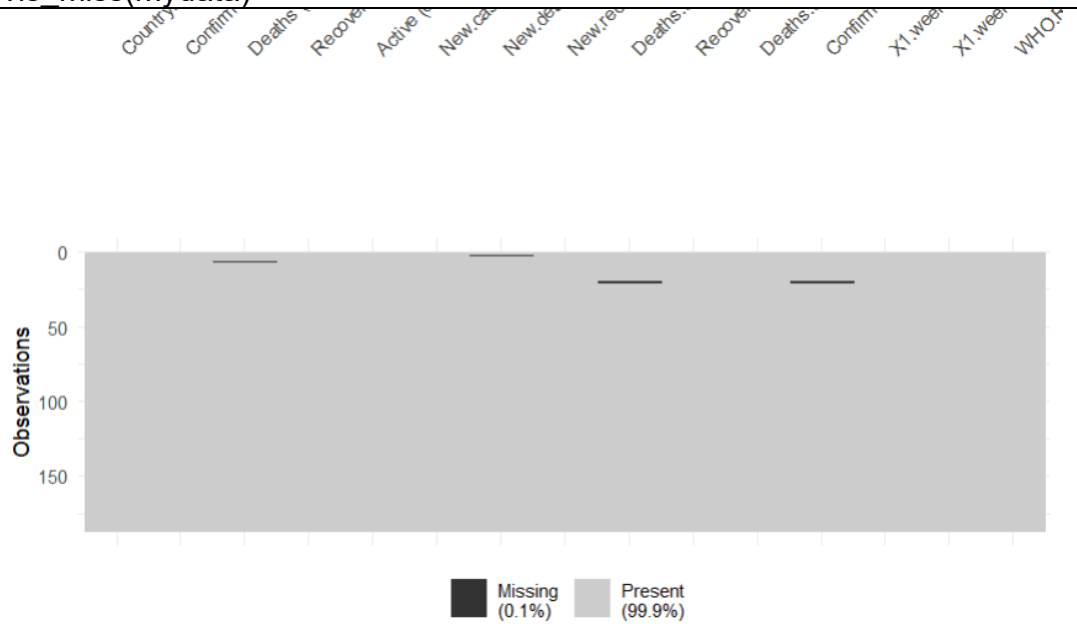
Values	
Confirmed_mean	88130.935828877
deaths_median	110
duplicate_rows	0L

Description:

We checked the dataset for duplicate rows using the duplicated() function. No duplicate rows were found (result: 0). Therefore, no rows were removed. The dataset remains unchanged with 185 observations (rows) and 15 variables (columns). All rows are unique and ready for further analysis.

Missing values on a graph

```
gg_miss_var(mydata)
vis_miss(mydata)
```



Description:

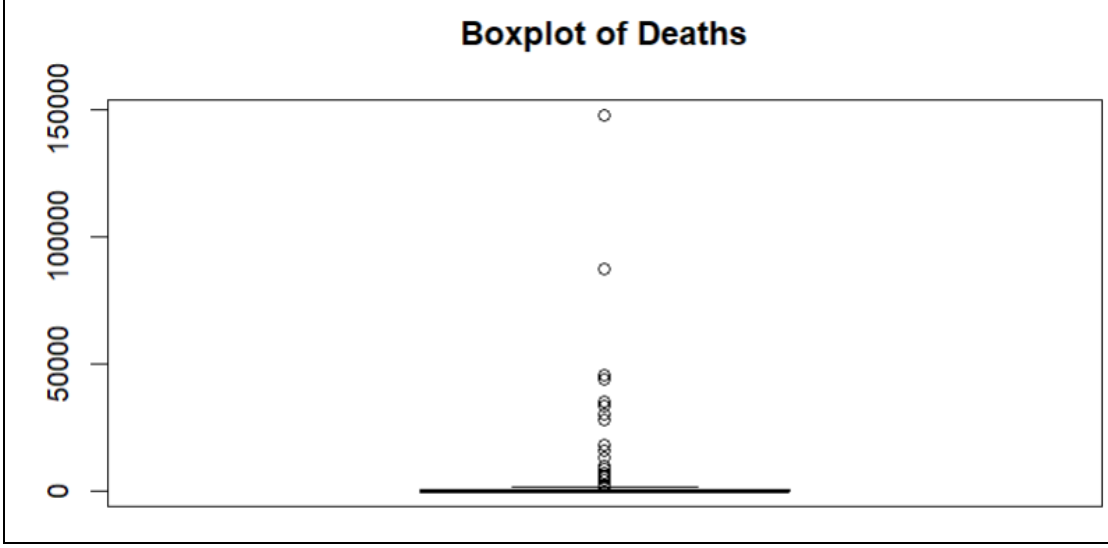
The gg_miss_var() and vis_miss() plots to see the missing data clearly. The graphs show that almost all the data is complete. Only a very small part (about 0.1%) is missing, and it is in just a few columns. More than 99.9% of the data is present.

Detect outlier

```
outlier_data <- duplicated_row
summary(outlier_data$Deaths)

boxplot(outlier_data$Deaths, main = "Boxplot of Deaths", col = "Black")
outlier_data$Deaths[outlier_data$Deaths < 0] <- 0
outlier_data$Deaths[is.na(outlier_data$Deaths)] <- median(outlier_data$Deaths,
na.rm = TRUE)

Q1_Deaths <- quantile(outlier_data$Deaths, 0.25)
Q3_Deaths <- quantile(outlier_data$Deaths, 0.75)
IQR_Deaths <- Q3_Deaths - Q1_Deaths

lower_bound_Deaths <- Q1_Deaths - 1.5 * IQR_Deaths
upper_bound_Deaths <- Q3_Deaths + 1.5 * IQR_Deaths

outlier_data$Deaths[outlier_data$Deaths < lower_bound_Deaths |
                    outlier_data$Deaths > upper_bound_Deaths] <-
median(outlier_data$Deaths)
summary(outlier_data$Deaths)
```

```
summary(outlier_data$Deaths)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.0    20.0   110.0   212.7   165.0  1764.0
```

Q1_Deaths	Named num 20
Q3_Deaths	Named num 748
upper_bound_Deaths	Named num 1840



Description:

A boxplot was created for the Deaths column. It showed several high outliers because a few countries had much higher death counts. The IQR method (1.5 × IQR rule) was applied to find extreme values. All values above the upper limit were replaced with the median (110), and negative values were set to 0. After this step, the Deaths column became clean with no outliers. The dataset still has 185 rows and 15 columns.

Convert attributes from numerical to categorical

```
NumToCata <- duplicated_row
NumToCata$Deaths...100.Cases <- ifelse(
  NumToCata$Deaths...100.Cases >= 0 & NumToCata$Deaths...100.Cases <= 4,
"Low",
  ifelse(
    NumToCata$Deaths...100.Cases >= 5 & NumToCata$Deaths...100.Cases <= 9,
"Medium",
    "High"
  )
)
NumToCata$Deaths...100.Cases
```

```
> NumToCata$Deaths...100.Cases
  [1] "Low"    "High"   "Medium" "High"   "Low"    "Low"    "Low"    "Low"    "Low"    "Low"
 [11] "Low"    "Low"    "Low"    "Medium" "Low"    "High"   "High"   "Low"    "Low"    "Low"
 [21] "Low"    "Low"    "Low"    "Low"    "High"   "Low"    "Low"    "Low"    "Low"    "Low"
 [31] "Medium" "Low"    "Medium" "Low"    "Medium" "Low"    "Low"    "Low"    "Low"    "Low"
 [41] "Low"    "Low"    "Low"    "Low"    "Low"    "High"   "Low"    "Low"    "Low"    "Medium"
 [51] "Medium" "Low"    "Low"    "Low"    "Low"    "Low"    "Low"    "Low"    "High"   "High"
 [61] "Low"    "Low"    "Low"    "High"   "Low"    "High"   "Low"    "Low"    "Low"    "Low"
 [71] "Low"    "Medium" "Low"    "Low"    "Low"    "High"   "Low"    "Low"    "High"   "Medium"
 [81] "Low"    "Medium" "Low"    "High"   "Low"    "Low"    "Low"    "Low"    "Low"    "Low"
 [91] "Low"    "Low"    "Low"    "Low"    "Low"    "Low"    "Medium" "Low"    "Low"    "Low"
[101] "Low"    "Low"    "Low"    "Low"    "Low"    "High"   "Low"    "Low"    "Low"    "High"
[111] "Low"    "Low"    "Low"    "Low"    "Low"    "Low"    "Low"    "Low"    "High"   "Low"
[121] "Low"    "Medium" "Low"    "High"   "Low"    "Low"    "Low"    "Low"    "Low"    "Low"
```

mode_add	187 obs. of 15 variables
mydata	187 obs. of 15 variables
NumToCata	185 obs. of 15 variables

Description:

The Deaths / 100 Cases column was originally numeric. It was converted into three simple categories using the ifelse() function. Values between 0 and 4 became "Low", values from 5 to 9 became "Medium", and values 10 or higher became "High". After this step, the column contains only the labels "Low", "Medium", and "High" instead of numbers. The dataset still has 185 rows and 15 columns.

Convert categorical to numerical

```
CataToNum <- NumToCata
CataToNum$Deaths_numeric <- ifelse(
  CataToNum$Deaths...100.Cases == "Low", 1,
  ifelse(
    CataToNum$Deaths...100.Cases == "Medium", 2,
    ifelse(
      CataToNum$Deaths...100.Cases == "High", 3,
      NA_real_
    )
  )
)
CataToNum$Deaths_numeric
```

```
> CataToNum$Deaths_numeric
  [1] 1 3 2 3 1 1 1 1 1 1 1 1 1 2 1 3 3 1 1 1 1 1 1 1 3 1 1 1 1 1 2 1 2 1 2 1 1 1 1 1 1 1 1 1
 [46] 3 1 1 1 2 2 1 1 1 1 1 1 1 3 3 1 1 1 3 1 3 1 1 1 1 1 2 1 1 1 3 1 1 3 2 1 2 1 3 1 1 1 1 1
 [91] 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 3 1 1 1 3 1 1 1 1 1 1 1 1 3 1 1 2 1 3 1 1 1 1 1 1 3 1 1 1 1
[136] 3 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 3 1 2 1 2 2 2 1 1 3 1 1 1 2 1 1 1 1 1 1 3 1 1 1 1
[181] 1 3 3 1 1
>
```

Data	
CataToNum	185 obs. of 16 variables
convert	185 obs. of 15 variables

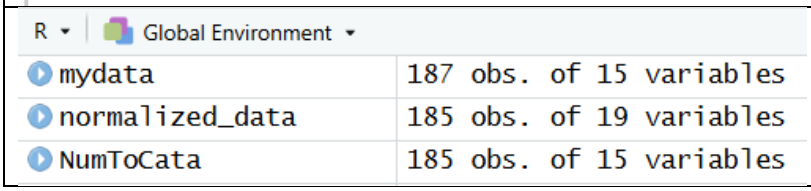
Description:

The Deaths / 100 Cases column, which had been changed into categories (Low, Medium, High), was converted back into numbers for easier use in calculations. A new column named Deaths_numeric was created using the ifelse() function. The category "Low" was assigned the number 1, "Medium" became 2, and "High" became 3. After this step, all rows now have a simple numeric value (1, 2, 3) in the new column instead of text.

Normalization method

```
normalized_data <- outlier_data %>%
  mutate(
    Confirmed_norm = (Confirmed - min(Confirmed)) / (max(Confirmed) -
min(Confirmed)),
    Deaths_norm    = (Deaths    - min(Deaths))    / (max(Deaths)    - min(Deaths)),
    Recovered_norm = (Recovered - min(Recovered)) / (max(Recovered) -
min(Recovered)),
    Active_norm    = (Active    - min(Active))    / (max(Active)    - min(Active))
  )
head(normalized_data)
```

```
head(normalized_data)
     Country.Region Confirmed Deaths Recovered Active New.cases New.deaths New.recovered
        Afghanistan     36263   1269     25198   9796       106         10            18
            Algeria     27973   1163     18837   7973       616          8           749
            Andorra       907     52       803     52        10          0             0
             Angola       950     41       242    667        18          1             0
Antigua and Barbuda        86    110        65     18         4          0             5
          Argentina    167416    110     72575  91782      4890        120          2057
Deaths...100.Cases Recovered...100.Cases Deaths...100.Recovered Confirmed.last.week
              3.50                 69.49                   5.04               35526
              4.16                 67.34                   6.17               23691
              5.73                 88.53                   6.48                 884
              4.32                 25.47                  16.94                 749
              3.49                 75.58                   4.62                  76
              1.83                 43.35                   4.21              130774
           WHO.Region Confirmed_norm Deaths_norm Recovered_norm    Active_norm
Eastern Mediterranean  0.00845009229  0.71938776  0.01364531601 0.003478144781
               Africa  0.00651780351  0.65929705  0.01020068329 0.002830874677
               Europe  0.00020907877  0.02947846  0.00043484359 0.000018462998
               Africa  0.00021910150  0.02324263  0.00013104875 0.000236823455
             Americas  0.00001771459  0.06122449  0.00003519905 0.000006391038
             Americas  0.03902011282  0.06122449  0.03930108776 0.032587901623
```



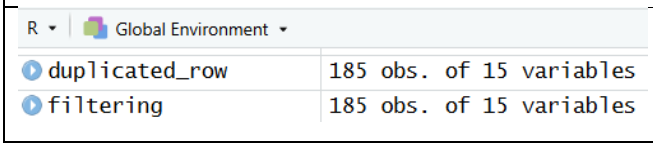
Description:

The numerical columns Confirmed, Deaths, Recovered, and Active were normalized using the min-max normalization method. Each value was transformed so that the smallest number in each column becomes 0 and the largest becomes 1, with all other values are between 0 and 1. This was done using the formula: (value – min) / (max – min). New columns were created: Confirmed_norm, Deaths_norm, Recovered_norm, and Active_norm. After normalization, all these columns now have values only between 0 and 1. The dataset now has 185 rows and 19 columns.

Filtering data

```
filtering <- duplicated_row
check <- filtering[filtering$Deaths > 0 | filtering$Deaths < 1000000, ]
print(check)
filtering$Deaths[filtering$Deaths < 0] <- 0
head(filtering)
```

	Country.Region	Confirmed	Deaths	Recovered	Active	New.cases	New.deaths
1	Afghanistan	36263	1269	25198	9796	106	10
3	Algeria	27973	1163	18837	7973	616	8
4	Andorra	907	52	803	52	10	0
5	Angola	950	41	242	667	18	1
6	Antigua and Barbuda	86	110	65	18	4	0
7	Argentina	167416	3059	72575	91782	4890	120
8	Armenia	37390	711	26665	10014	73	6
9	Australia	15303	167	9311	5825	368	6
10	Austria	20558	713	18246	1599	86	1



Description:

The dataset checks for invalid death values. Rows with negative or extremely high death counts were identified. Any negative values were replaced with 0 to ensure data accuracy. Negative values in the Deaths column were replaced with 0.

Invalid data

```
inv_data <- duplicated_row
valid_regions <- c("Americas", "Europe", "Africa",
                   "Western Pacific", "Eastern Mediterranean",
                   "South-East Asia")

invalid_categories <- inv_data$WHO.Region[!(inv_data$WHO.Region %in%
valid_regions)]

cat("\nInvalid categorical values in WHO.Region:\n")
print(invalid_categories)
```

```
> inv_data <- duplicated_row
> valid_regions <- c("Americas", "Europe", "Africa",
+                    "Western Pacific", "Eastern Mediterranean",
+                    "South-East Asia")
> invalid_categories <- inv_data$WHO.Region[!(inv_data$WHO.Region %in% valid_regions)]
> cat("\nInvalid categorical values in WHO.Region\n")

Invalid categorical values in WHO.Region
> print(invalid_categories)
character(0)
```

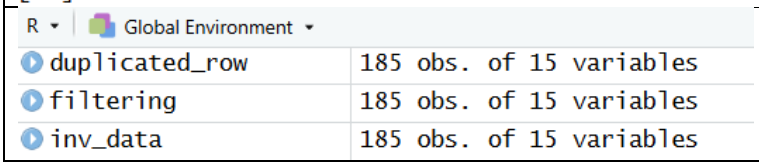
Description:

The WHO.Region column was checked for invalid entries. A list (vector) of six correct WHO regions was created: Americas, Europe, Africa, Western Pacific, Eastern Mediterranean, and South-East Asia. The code then found any values in WHO.Region that did not match these six official regions. No invalid region names were found.

Invalid data handle

```
freq_table_region <- table(inv_data$WHO.Region)
mode_region <- names(freq_table_region)[which.max(freq_table_region)]
inv_data$WHO.Region[!(inv_data$WHO.Region %in% valid_regions)] <-
mode_region
inv_data$WHO.Region
```

```
> inv_data$WHO.Region[!(inv_data$WHO.Region %in% valid_regions)] <- mode_regio
> inv_data$WHO.Region
 [1] "Eastern Mediterranean" "Africa"                "Europe"
 [4] "Africa"                "Americas"              "Americas"
 [7] "Europe"                "Western Pacific"       "Europe"
[10] "Europe"                "Americas"              "Eastern Mediterranean"
[13] "South-East Asia"       "Americas"              "Europe"
[16] "Europe"                "Americas"              "Africa"
[19] "Americas"              "Europe"                "Africa"
[22] "Americas"              "Western Pacific"       "Europe"
```



Description:

The WHO.Region column was checked for incorrect or invalid entries by comparing it with the six official WHO regions. The table() function was used to create a frequency count, and the most common region, "Europe", was selected as the mode. All invalid or misspelled region names were then replaced with "Europe" using the code: inv_data$WHO.Region[!(inv_data$WHO.Region %in% valid_regions)] <- mode_region. After this replacement, every value in the WHO.Region column became correct .The dataset still has 185 rows and 15 columns.

Imbalanced data set into a balanced data set

```
set.seed(123)
min_n <- min(table(CataToNum$Deaths...100.Cases))
balanced_data <- CataToNum %>%
  group_by(Deaths...100.Cases) %>%
  sample_n(size = min_n) %>%
  ungroup()
table(balanced_data$Deaths...100.Cases)
prop.table(table(balanced_data$Deaths...100.Cases))
```

```
> table(balanced_data_under$Deaths...100.Cases)

  High    Low Medium
    19     19     19
> prop.table(table(balanced_data_under$Deaths...100.Cases))

     High       Low    Medium
0.3333333 0.3333333 0.3333333
```

Data	
balanced_data	57 obs. of 16 variables

Description:

The original dataset had an imbalanced Deaths / 100 Cases category (Low, Medium, High). To make it balanced, undersampling was done using set.seed(123) for consistent results. The code grouped the data by Deaths...100.Cases using group_by(), then used sample_n() to randomly select the same number of rows (19) from each category matching the size of the smallest group. Finally, ungroup() was applied. After running this code, the new dataset (balanced_data) has exactly 19 rows for Low, 19 for Medium, and 19 for High a total of 57 rows. Each category now makes up exactly 33.33% of the data. The final dataset has 57 rows and 16 columns.

Split the dataset

```
set.seed(123)
n <- nrow(duplicated_row)

train_index <- sample(1:n, size = 0.8 * n)
train_data <- duplicated_row[train_index, ]
test_data  <- duplicated_row[-train_index, ]

cat("Training rows:", nrow(train_data), "\n")
cat("Testing rows:", nrow(test_data), "\n")
```

```
> cat("Training rows:", nrow(train_data), "\n")
Training rows: 148
> cat("Testing rows:", nrow(test_data), "\n")
Testing rows: 37
```

R | Global Environment

Q1_Deaths	Named num 20
Q3_Deaths	Named num 748
train_index	int [1:148] 159 179 14 170 50 118 43 183 180 153 ...

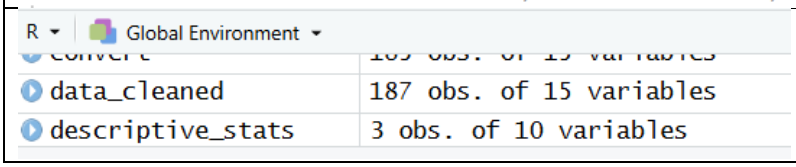
Description:

The cleaned dataset with 185 rows was split into training and testing sets. Using set.seed(123) for reproducibility, 80% of the rows were randomly selected for training and the remaining 20% for testing. The sample() function created the train_index, then the data was divided using indexing. After splitting, the training set has 148 rows and the testing set has 37 rows. This 80-20 split allows the model to be trained on most of the data while keeping a separate portion for fair evaluation. Both sets still have 15 columns.

Calculate descriptive statistics and interpret

```
numeric_vars <- c("Confirmed", "Deaths", "Recovered", "Active")
descriptive_stats <- balanced_data %>%
  group_by(Deaths...100.Cases) %>%
  summarise(
    Confirmed_mean = mean(Confirmed, na.rm = TRUE),
    Confirmed_sd   = sd(Confirmed, na.rm = TRUE),
    Deaths_mean    = mean(Deaths, na.rm = TRUE),
    Deaths_sd      = sd(Deaths, na.rm = TRUE),
    Recovered_mean = mean(Recovered, na.rm = TRUE),
    Recovered_sd   = sd(Recovered, na.rm = TRUE),
    Active_mean    = mean(Active, na.rm = TRUE),
    Active_sd      = sd(Active, na.rm = TRUE),
    Count = n()
  )
print(descriptive_stats)
```

```
Deaths...100.Cases Confirmed_mean Confirmed_sd Deaths_mean Deaths_sd Recovered_mean
<chr>                       <dbl>        <dbl>       <dbl>     <dbl>          <dbl>
High                      107541.      142953.      10652.    15793.         63436.
Low                        29947.       58194.        712.     2005.         18881.
Medium                     43680.       72453.       2650.     4175.         24738.
i 4 more variables: Recovered_sd <dbl>, Active_mean <dbl>, Active_sd <dbl>, Count <int>
```



Description:

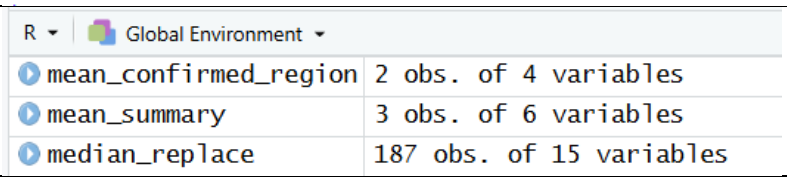
The balanced dataset was split into three groups: Low, Medium, and High based on Deaths / 100 Cases. For each group, the mean and standard deviation of Confirmed, Deaths, Recovered, and Active cases were calculated. The High group had the highest mean confirmed cases (107,541) and mean deaths (10,652) with the largest standard deviation. The Medium group showed medium values, and the Low group had the lowest means and smaller variation. This clearly shows that countries with higher death rates per 100 cases faced much bigger and more serious COVID-19 outbreaks. The analysis used 57 rows (19 per group).

Mean Comparison

```
mean_summary <- balanced_data %>%
  group_by(Deaths...100.Cases) %>%
  summarise(
    Confirmed_mean = mean(Confirmed, na.rm = TRUE),
    Deaths_mean    = mean(Deaths, na.rm = TRUE),
    Recovered_mean = mean(Recovered, na.rm = TRUE),
```

```
  Active_mean    = mean(Active, na.rm = TRUE),
  Count = n()
 )
print(mean_summary)
```

```
> print(mean_summary)
# A tibble: 3 × 6
  Deaths...100.Cases Confirmed_mean Deaths_mean Recovered_mean Active_mean Count
  <chr>                      <dbl>       <dbl>          <dbl>       <dbl> <int>
1 High                     107541.      10652.         63436.      33454.    19
2 Low                       29947.        712.         18881.      10353.    19
3 Medium                    43680.       2650.         24738.      16291.    19
```



Description:

This code calculates and compares the average values of key numerical variables across the different categories of Deaths...100.Cases. First, the dataset is grouped into three levels Low, Medium and High based on the death rate per 100 cases. For each of these categories, the code computes the mean number of confirmed cases, deaths, recovered cases and active cases. It also counts how many observations fall under each category. The purpose of this analysis is to see how these numerical values differ when the death-rate category changes, helping to understand whether higher death-rate groups are associated with higher or lower COVID-19 case numbers.

Variability Comparison

```
variability_deaths <- balanced_data %>%
  group_by(Deaths...100.Cases) %>%
  summarise(
    Min_Deaths   = min(Deaths, na.rm = TRUE),
    Max_Deaths   = max(Deaths, na.rm = TRUE),
    Range_Deaths = Max_Deaths - Min_Deaths,
    IQR_Deaths   = IQR(Deaths, na.rm = TRUE),
    Var_Deaths   = var(Deaths, na.rm = TRUE),
    SD_Deaths    = sd(Deaths, na.rm = TRUE),
    Count        = n()
  )
print(variability_deaths)
```

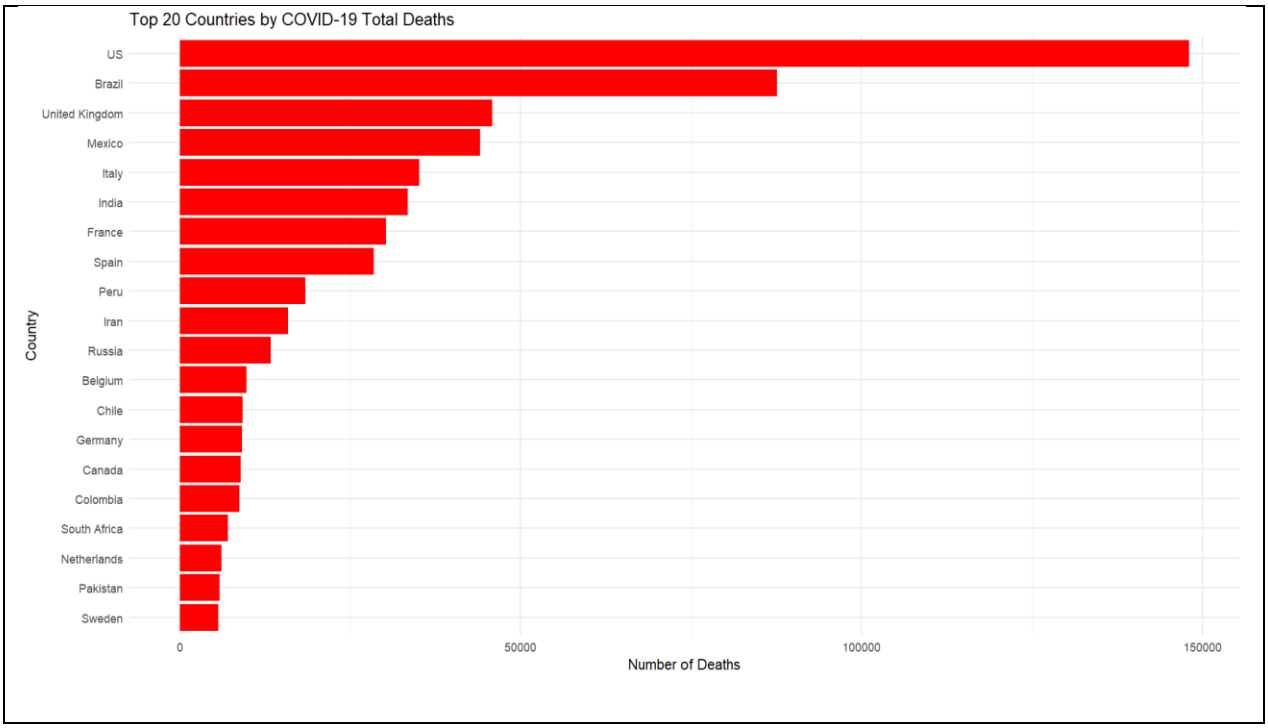
```
  Deaths...100.Cases Min_Deaths Max_Deaths Range_Deaths IQR_Deaths Var_Deaths SD_Deaths Count
  <chr>                   <dbl>      <dbl>        <dbl>      <dbl>      <dbl>     <dbl> <int>
1 High                        1      45844        45843      13957  249403503.    15793.    19
2 Low                         0       8777         8777        234.   4021314.     2005.    19
3 Medium                      7      15912        15905       4607   17427907.     4175.    19
```

Description:

This code measures how the number of deaths varies across the three categories of Deaths...100.Cases (Low, Medium, High). It groups the data by these categories and then calculates several variability measures for the Deaths variable. For each group, it finds the minimum and maximum number of deaths, the range (difference between max and min), the interquartile range (IQR), the variance and the standard deviation. These values show how spread out or consistent the death numbers are within each category. The code also counts how many data rows belong to each group. The final output helps to compare how much the death values change from one category to another.

Bar charts

```
data_plot <- duplicated_row
top20_deaths <- data_plot %>%
  arrange(desc(Deaths)) %>%
  slice(1:20)

ggplot(top20_deaths, aes(x = reorder(Country.Region, Deaths), y = Deaths)) +
  geom_col(fill = "red") +
  coord_flip() +
  labs(
    title = "Top 20 Countries by COVID-19 Total Deaths",
    x = "Country",
    y = "Number of Deaths"
  ) +
  theme_minimal()
```



Description:

A bar chart was created to show the 20 countries with the highest total deaths. The data was sorted in descending order by Deaths, and the top 20 rows were selected. The United States had the highest number of deaths, followed by Brazil, India, and Mexico. The bars are colored red and arranged horizontally for easy reading. This chart clearly shows that a small number of countries accounted for most of the COVID-19 deaths worldwide. The visualization was made using ggplot2 with coord_flip() for better country name display.

Conclusion:

This analysis of the country-wise COVID-19 dataset provided valuable insights into the global state of the pandemic. The findings indicate that the impact was highly concentrated, with the Americas region, particularly the United States and Brazil, bearing the heaviest burden in terms of confirmed cases and deaths.

The strong positive correlation between confirmed cases and deaths underscores the direct relationship between the spread of the virus and its fatal outcomes. Furthermore, the analysis highlighted significant regional variations in mortality rates, suggesting that factors beyond just case numbers, such as healthcare system capacity and demographic profiles, likely played a crucial role.