Exploratory Data Analysis of COVID-19 Data

Your Name

The Date

Section 1: Exploratory Data Analysis (EDA)

Supplementary Dataset EDA to Select Transmission Rate Proxy

```r
## This subsection is dedicated to performing an Exploratory Data Analysis (EDA)
#  on the supplementary COVID-19 dataset.
## The rationale for this analysis is twofold:

## 1. Comparative Data Analysis:
##    - The supplementary dataset provides a broader, global context to the
#       COVID-19 pandemic.
##    - By exploring this data, we can compare and contrast different
#       regions/countries in terms of COVID-19 impact and response.
##    - This comparative analysis is crucial to understand regional variations
#       in pandemic progression and policy effectiveness.

# 2. Data Compatibility and Integrity Check:
#    - Conducting an EDA on this dataset is essential for verifying its
#      compatibility with our primary data source.
#    - We need to ensure that the metrics and trends in this dataset align with
#      those in our primary dataset, providing a cohesive analytical base.
#    - This step is also necessary to check for data integrity issues such as
#      missing values, outliers, or inconsistencies, which could affect our
#      overall analysis.

# The insights derived from this EDA will help in framing our analysis, guiding
# subsequent data processing steps,
# and ensuring that our conclusions are based on a comprehensive understanding
# of both datasets.

# Read COVID-19 data into 'cd' dataframe
cd <- read_csv("owid-covid-data.csv")
```

```
Rows: 358803 Columns: 67
-- Column specification --------------------------------------------------------
Delimiter: ","
chr   (4): iso_code, continent, location, tests_units
dbl  (62): total_cases, new_cases, new_cases_smoothed, total_deaths, new_dea...
date  (1): date

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
glimpse(cd)
```

```
Rows: 358,803
Columns: 67
$ iso_code                                   <chr> "AFG", "AFG", "AFG", "AFG",~
$ continent                                  <chr> "Asia", "Asia", "Asia", "As~
$ location                                   <chr> "Afghanistan", "Afghanistan~
$ date                                       <date> 2020-01-03, 2020-01-04, 20~
$ total_cases                                <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ new_cases                                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ new_cases_smoothed                         <dbl> NA, NA, NA, NA, NA, 0, 0, 0~
$ total_deaths                               <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ new_deaths                                 <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ new_deaths_smoothed                        <dbl> NA, NA, NA, NA, NA, 0, 0, 0~
$ total_cases_per_million                    <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ new_cases_per_million                      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ new_cases_smoothed_per_million             <dbl> NA, NA, NA, NA, NA, 0, 0, 0~
$ total_deaths_per_million                   <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ new_deaths_per_million                     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ new_deaths_smoothed_per_million            <dbl> NA, NA, NA, NA, NA, 0, 0, 0~
$ reproduction_rate                          <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ icu_patients                               <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ icu_patients_per_million                   <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ hosp_patients                              <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ hosp_patients_per_million                  <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ weekly_icu_admissions                      <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ weekly_icu_admissions_per_million          <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ weekly_hosp_admissions                     <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ weekly_hosp_admissions_per_million         <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ total_tests                                <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ new_tests                                  <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ total_tests_per_thousand                   <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ new_tests_per_thousand                     <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ new_tests_smoothed                         <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ new_tests_smoothed_per_thousand            <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ positive_rate                              <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ tests_per_case                             <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ tests_units                                <chr> NA, NA, NA, NA, NA, NA, NA,~
$ total_vaccinations                         <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ people_vaccinated                          <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ people_fully_vaccinated                    <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ total_boosters                             <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ new_vaccinations                           <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ new_vaccinations_smoothed                  <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ total_vaccinations_per_hundred             <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ people_vaccinated_per_hundred              <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ people_fully_vaccinated_per_hundred        <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ total_boosters_per_hundred                 <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ new_vaccinations_smoothed_per_million      <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ new_people_vaccinated_smoothed             <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ new_people_vaccinated_smoothed_per_hundred <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ stringency_index                           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ population_density                         <dbl> 54.422, 54.422, 54.422, 54.~
$ median_age                                 <dbl> 18.6, 18.6, 18.6, 18.6, 18.~
$ aged_65_older                              <dbl> 2.581, 2.581, 2.581, 2.581,~
$ aged_70_older                              <dbl> 1.337, 1.337, 1.337, 1.337,~
$ gdp_per_capita                             <dbl> 1803.987, 1803.987, 1803.98~
$ extreme_poverty                            <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ cardiovasc_death_rate                      <dbl> 597.029, 597.029, 597.029, ~
$ diabetes_prevalence                        <dbl> 9.59, 9.59, 9.59, 9.59, 9.5~
$ female_smokers                             <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ male_smokers                               <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ handwashing_facilities                     <dbl> 37.746, 37.746, 37.746, 37.~
$ hospital_beds_per_thousand                 <dbl> 0.5, 0.5, 0.5, 0.5, 0.5, 0.~
$ life_expectancy                            <dbl> 64.83, 64.83, 64.83, 64.83,~
$ human_development_index                    <dbl> 0.511, 0.511, 0.511, 0.511,~
$ population                                 <dbl> 41128772, 41128772, 4112877~
$ excess_mortality_cumulative_absolute       <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ excess_mortality_cumulative                <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ excess_mortality                           <dbl> NA, NA, NA, NA, NA, NA, NA,~
$ excess_mortality_cumulative_per_million    <dbl> NA, NA, NA, NA, NA, NA, NA,~
```

```r
summary(cd)
```

```
   iso_code          continent           location              date           
 Length:358803      Length:358803      Length:358803      Min.   :2020-01-01  
 Class :character   Class :character   Class :character   1st Qu.:2020-12-25  
 Mode  :character   Mode  :character   Mode  :character   Median :2021-12-13  
                                                          Mean   :2021-12-13  
                                                          3rd Qu.:2022-12-01  
                                                          Max.   :2023-11-30  
                                                                              
  total_cases          new_cases       new_cases_smoothed  total_deaths    
 Min.   :        1   Min.   :      0   Min.   :      0    Min.   :      1  
 1st Qu.:     8355   1st Qu.:      0   1st Qu.:      0    1st Qu.:    129  
 Median :    72719   Median :      1   Median :     23    Median :   1349  
 Mean   :  6862419   Mean   :   9376   Mean   :   9409    Mean   :  86954  
 3rd Qu.:   784483   3rd Qu.:    244   3rd Qu.:    474    3rd Qu.:  12019  
 Max.   :772165753   Max.   :8401963   Max.   :6402036    Max.   :6981250  
 NA's   :38175       NA's   :9771      NA's   :11030      NA's   :59717    
   new_deaths       new_deaths_smoothed total_cases_per_million
 Min.   :    0.00   Min.   :    0.000   Min.   :     0         
 1st Qu.:    0.00   1st Qu.:    0.000   1st Qu.:  2693         
 Median :    0.00   Median :    0.143   Median : 29666         
 Mean   :   83.52   Mean   :   83.807   Mean   :104442         
 3rd Qu.:    2.00   3rd Qu.:    4.857   3rd Qu.:138729         
 Max.   :57889.00   Max.   :14822.000   Max.   :746008         
 NA's   :9714       NA's   :10944       NA's   :38175          
 new_cases_per_million new_cases_smoothed_per_million total_deaths_per_million
 Min.   :     0.00     Min.   :    0.00                Min.   :   0.00        
```

```
1st Qu.:     0.00     1st Qu.:    0.03               1st Qu.:  61.64
Median :     0.06     Median :    5.95               Median : 388.26
Mean   :   141.89     Mean   :  142.40               Mean   : 883.47
3rd Qu.:    32.89     3rd Qu.:   78.60               3rd Qu.:1382.95
Max.   :228872.02     Max.   :37241.78               Max.   :6511.88
NA's   :9771          NA's   :11030                  NA's   :59717
new_deaths_per_million new_deaths_smoothed_per_million reproduction_rate
Min.   : 0.000         Min.   : 0.000                  Min.   :-0.07
1st Qu.: 0.000         1st Qu.: 0.000                  1st Qu.: 0.72
Median : 0.000         Median : 0.008                  Median : 0.95
Mean   : 0.887         Mean   : 0.890                  Mean   : 0.91
3rd Qu.: 0.156         3rd Qu.: 0.541                  3rd Qu.: 1.14
Max.   :603.656        Max.   :148.641                 Max.   : 5.87
NA's   :9714           NA's   :10944                   NA's   :173986
 icu_patients     icu_patients_per_million hosp_patients
Min.   :    0.0   Min.   : 0.0             Min.   :     0
1st Qu.:   22.0   1st Qu.: 2.5             1st Qu.:   197
Median :   96.0   Median : 6.9             Median :   784
Mean   :  675.1   Mean   : 16.2            Mean   :  3936
3rd Qu.:  434.0   3rd Qu.: 19.9            3rd Qu.:  3056
Max.   :28891.0   Max.   :180.7            Max.   :154497
NA's   :320958    NA's   :320958           NA's   :319562
hosp_patients_per_million weekly_icu_admissions
Min.   :   0.0            Min.   :   0.0
1st Qu.:  33.8            1st Qu.:  20.2
Median :  76.9            Median : 105.0
Mean   : 129.4            Mean   : 336.6
3rd Qu.: 164.2            3rd Qu.: 408.0
Max.   :1526.8            Max.   :4838.0
NA's   :319562            NA's   :348489
weekly_icu_admissions_per_million weekly_hosp_admissions
Min.   : 0.0                      Min.   :     0
1st Qu.: 1.8                      1st Qu.:   241
Median : 5.1                      Median :   887
Mean   : 10.2                     Mean   :  4325
3rd Qu.: 13.4                     3rd Qu.:  3996
Max.   :225.0                     Max.   :153977
NA's   :348489                    NA's   :335296
weekly_hosp_admissions_per_million  total_tests          new_tests
Min.   : 0.0                       Min.   :0.000e+00   Min.   :       1
1st Qu.: 25.6                      1st Qu.:3.647e+05   1st Qu.:    2244
Median : 58.7                      Median :2.067e+06   Median :    8783
Mean   : 85.4                      Mean   :2.110e+07   Mean   :   67285
3rd Qu.:113.7                      3rd Qu.:1.025e+07   3rd Qu.:   37229
Max.   :712.1                      Max.   :9.214e+09   Max.   :35855632
NA's   :335296                     NA's   :279416      NA's   :283400
total_tests_per_thousand new_tests_per_thousand new_tests_smoothed
Min.   :    0.00         Min.   : 0.00          Min.   :       0
1st Qu.:   43.59         1st Qu.: 0.29          1st Qu.:    1486
Median :  234.14         Median : 0.97          Median :    6570
Mean   :  924.25         Mean   : 3.27          Mean   :  142178
3rd Qu.:  894.37         3rd Qu.: 2.91          3rd Qu.:   32205
Max.   :32925.83         Max.   :531.06         Max.   :14769984
NA's   :279416           NA's   :283400         NA's   :254838
```

```
new_tests_smoothed_per_thousand positive_rate    tests_per_case
Min.   :  0.00                  Min.   :0.00     Min.   :      1.0
1st Qu.:  0.20                  1st Qu.:0.02     1st Qu.:      7.1
Median :  0.85                  Median :0.06     Median :     17.5
Mean   :  2.83                  Mean   :0.10     Mean   :   2403.6
3rd Qu.:  2.58                  3rd Qu.:0.14     3rd Qu.:     54.6
Max.   :147.60                  Max.   :1.00     Max.   :1023631.9
NA's   :254838                  NA's   :262876   NA's   :264455
tests_units        total_vaccinations  people_vaccinated
Length:358803      Min.   :0.000e+00   Min.   :0.000e+00
Class :character   1st Qu.:1.706e+06   1st Qu.:9.055e+05
Mode  :character   Median :1.207e+07   Median :6.278e+06
                   Mean   :4.617e+08   Mean   :2.049e+08
                   3rd Qu.:9.857e+07   3rd Qu.:4.478e+07
                   Max.   :1.353e+10   Max.   :5.630e+09
                   NA's   :278655      NA's   :282092
people_fully_vaccinated total_boosters      new_vaccinations
Min.   :1.000e+00       Min.   :1.000e+00   Min.   :       0
1st Qu.:8.557e+05       1st Qu.:4.676e+05   1st Qu.:    2572
Median :5.747e+06       Median :4.674e+06   Median :   24383
Mean   :1.870e+08       Mean   :1.201e+08   Mean   :  792131
3rd Qu.:4.272e+07       3rd Qu.:3.530e+07   3rd Qu.:  204079
Max.   :5.178e+09       Max.   :2.802e+09   Max.   :49673299
NA's   :285397          NA's   :310461      NA's   :292694
new_vaccinations_smoothed total_vaccinations_per_hundred
Min.   :       0          Min.   :  0.00
1st Qu.:     311          1st Qu.: 39.66
Median :    4230          Median :120.74
Mean   :  299839          Mean   :119.07
3rd Qu.:   34106          3rd Qu.:191.56
Max.   :43691637          Max.   :406.90
NA's   :174761            NA's   :278655
people_vaccinated_per_hundred people_fully_vaccinated_per_hundred
Min.   :  0.00                Min.   :  0.00
1st Qu.: 25.61                1st Qu.: 19.26
Median : 61.44                Median : 55.35
Mean   : 52.21                Mean   : 47.26
3rd Qu.: 77.46                3rd Qu.: 73.32
Max.   :129.07                Max.   :126.89
NA's   :282092                NA's   :285397
total_boosters_per_hundred new_vaccinations_smoothed_per_million
Min.   :  0.00             Min.   :     0
1st Qu.:  4.76             1st Qu.:   129
Median : 33.74             Median :   681
Mean   : 34.55             Mean   :  1950
3rd Qu.: 56.64             3rd Qu.:  2565
Max.   :150.47             Max.   :117113
NA's   :310461             NA's   :174761
new_people_vaccinated_smoothed new_people_vaccinated_smoothed_per_hundred
Min.   :       0               Min.   : 0.00
1st Qu.:      54               1st Qu.: 0.00
Median :     936               Median : 0.02
Mean   :  110941               Mean   : 0.08
3rd Qu.:   10456               3rd Qu.: 0.08
```

```
Max.   :21071272               Max.   :11.71
NA's   :175013                 NA's   :175013
stringency_index population_density   median_age    aged_65_older
Min.   :  0.00   Min.   :    0.14   Min.   :15.1    Min.   : 1.14
1st Qu.: 22.22   1st Qu.:   37.73   1st Qu.:22.2    1st Qu.: 3.53
Median : 42.59   Median :   90.67   Median :29.7    Median : 6.38
Mean   : 42.71   Mean   :  401.25   Mean   :30.5    Mean   : 8.70
3rd Qu.: 62.04   3rd Qu.:  222.87   3rd Qu.:38.7    3rd Qu.:13.93
Max.   :100.00   Max.   :20546.77   Max.   :48.2    Max.   :27.05
NA's   :161152   NA's   :54145      NA's   :75516   NA's   :85385
aged_70_older   gdp_per_capita     extreme_poverty  cardiovasc_death_rate
Min.   : 0.53   Min.   :   661.2   Min.   : 0.10    Min.   : 79.37
1st Qu.: 2.08   1st Qu.:  3823.2   1st Qu.: 0.60    1st Qu.:175.70
Median : 3.87   Median : 12294.9   Median : 2.50    Median :245.46
Mean   : 5.50   Mean   : 18968.4   Mean   :13.84    Mean   :264.30
3rd Qu.: 8.64   3rd Qu.: 27216.4   3rd Qu.:21.40    3rd Qu.:333.44
Max.   :18.49   Max.   :116935.6   Max.   :77.60    Max.   :724.42
NA's   :78356   NA's   :81125      NA's   :179808   NA's   :80468
diabetes_prevalence female_smokers    male_smokers    handwashing_facilities
Min.   : 0.99       Min.   : 0.10     Min.   : 7.70   Min.   :  1.19
1st Qu.: 5.35       1st Qu.: 1.90     1st Qu.:22.60   1st Qu.: 20.86
Median : 7.20       Median : 6.30     Median :33.10   Median : 49.84
Mean   : 8.56       Mean   :10.79     Mean   :32.91   Mean   : 50.79
3rd Qu.:10.79       3rd Qu.:19.30     3rd Qu.:41.30   3rd Qu.: 82.50
Max.   :30.53       Max.   :44.00     Max.   :78.10   Max.   :100.00
NA's   :66300       NA's   :149976    NA's   :152816  NA's   :222460
hospital_beds_per_thousand life_expectancy human_development_index
Min.   : 0.1               Min.   :53.28   Min.   :0.39
1st Qu.: 1.3               1st Qu.:69.59   1st Qu.:0.60
Median : 2.5               Median :75.05   Median :0.74
Mean   : 3.1               Mean   :73.71   Mean   :0.72
3rd Qu.: 4.2               3rd Qu.:79.46   3rd Qu.:0.83
Max.   :13.8               Max.   :86.75   Max.   :0.96
NA's   :113056             NA's   :28656   NA's   :89027
  population        excess_mortality_cumulative_absolute
Min.   :4.700e+01   Min.   : -37726.1
1st Qu.:4.490e+05   1st Qu.:    121.6
Median :5.882e+06   Median :   5969.0
Mean   :1.286e+08   Mean   :  53121.7
3rd Qu.:2.830e+07   3rd Qu.:  37707.3
Max.   :7.975e+09   Max.   :1289776.5
                    NA's   :346592
excess_mortality_cumulative excess_mortality
Min.   :-44.2               Min.   :-95.9
1st Qu.:  1.4               1st Qu.: -1.6
Median :  8.1               Median :  5.7
Mean   :  9.8               Mean   : 11.3
3rd Qu.: 15.4               3rd Qu.: 16.3
Max.   : 76.6               Max.   :377.6
NA's   :346592              NA's   :346592
excess_mortality_cumulative_per_million
Min.   :-2752.9
1st Qu.:   73.8
Median : 1116.0
```

```
 Mean   : 1675.4
 3rd Qu.: 2746.7
 Max.   :10292.9
 NA's   :346592
```

```
# The 'date' column contains temporal data ranging from 01-01-2020 to
# 30-11-2023, which is crucial for time series analysis.
# This allows us to track the progression of COVID-19 metrics over time and to
# examine trends, seasonality, and the impact of interventions.
# To facilitate this analysis, it's essential to ensure that the 'date' column
# is in the proper date format.
# Converting the 'date' column to R's Date type will enable accurate
# chronological ordering and time-based operations.
cd$date <- as.Date(cd$date, format="%Y-%m-%d") # Convert 'date' column to Date format.
```

```
# Check the range of dates
date_range <- range(cd$date, na.rm = TRUE)
```

```
# Format and print the date range in a more readable format
formatted_date_range <- format(date_range, "%Y-%m-%d")
cat("The date range in the dataset is from", formatted_date_range[1], "to", formatted_date_range[2], "\
```

```
The date range in the dataset is from 2020-01-01 to 2023-11-30
```

```
# To check for any missing dates, we can use the `seq` function to generate a complete
# sequence of datesand then identify which ones are not in the 'date' column of our dataset
all_dates <- seq(from = min(cd$date, na.rm = TRUE), to = max(cd$date, na.rm = TRUE), by = "day")
missing_dates <- setdiff(all_dates, cd$date)
```

```
# Display missing dates, if any
if (length(missing_dates) > 0) {
  cat("There are", length(missing_dates), "missing dates in the dataset:\n")
  print(missing_dates)
} else {
  cat("No missing dates in the dataset.\n")
}
```

```
No missing dates in the dataset.
```

```
unique_regions <- unique(cd$location)
print(unique_regions)
```

```
  [1] "Afghanistan"                        "Africa"
  [3] "Albania"                            "Algeria"
  [5] "American Samoa"                     "Andorra"
  [7] "Angola"                             "Anguilla"
  [9] "Antigua and Barbuda"                "Argentina"
 [11] "Armenia"                            "Aruba"
 [13] "Asia"                               "Australia"
 [15] "Austria"                            "Azerbaijan"
 [17] "Bahamas"                            "Bahrain"
 [19] "Bangladesh"                         "Barbados"
 [21] "Belarus"                            "Belgium"
 [23] "Belize"                             "Benin"
 [25] "Bermuda"                            "Bhutan"
 [27] "Bolivia"                            "Bonaire Sint Eustatius and Saba"
 [29] "Bosnia and Herzegovina"             "Botswana"
 [31] "Brazil"                          "British Virgin Islands"
 [33] "Brunei"                          "Bulgaria"
 [35] "Burkina Faso"                    "Burundi"
 [37] "Cambodia"                        "Cameroon"
 [39] "Canada"                          "Cape Verde"
 [41] "Cayman Islands"                  "Central African Republic"
 [43] "Chad"                            "Chile"
 [45] "China"                           "Colombia"
 [47] "Comoros"                         "Congo"
 [49] "Cook Islands"                    "Costa Rica"
 [51] "Cote d'Ivoire"                   "Croatia"
 [53] "Cuba"                            "Curacao"
 [55] "Cyprus"                          "Czechia"
 [57] "Democratic Republic of Congo"    "Denmark"
 [59] "Djibouti"                        "Dominica"
 [61] "Dominican Republic"              "Ecuador"
 [63] "Egypt"                           "El Salvador"
 [65] "England"                         "Equatorial Guinea"
 [67] "Eritrea"                         "Estonia"
 [69] "Eswatini"                        "Ethiopia"
 [71] "Europe"                          "European Union"
 [73] "Faeroe Islands"                  "Falkland Islands"
 [75] "Fiji"                            "Finland"
 [77] "France"                          "French Guiana"
 [79] "French Polynesia"                "Gabon"
 [81] "Gambia"                          "Georgia"
 [83] "Germany"                         "Ghana"
 [85] "Gibraltar"                       "Greece"
 [87] "Greenland"                       "Grenada"
 [89] "Guadeloupe"                      "Guam"
 [91] "Guatemala"                       "Guernsey"
 [93] "Guinea"                          "Guinea-Bissau"
 [95] "Guyana"                          "Haiti"
 [97] "High income"                     "Honduras"
 [99] "Hong Kong"                       "Hungary"
[101] "Iceland"                         "India"
[103] "Indonesia"                       "Iran"
[105] "Iraq"                            "Ireland"
[107] "Isle of Man"                     "Israel"
[109] "Italy"                           "Jamaica"
[111] "Japan"                           "Jersey"
[113] "Jordan"                          "Kazakhstan"
[115] "Kenya"                           "Kiribati"
[117] "Kosovo"                          "Kuwait"
[119] "Kyrgyzstan"                      "Laos"
[121] "Latvia"                          "Lebanon"
[123] "Lesotho"                         "Liberia"
[125] "Libya"                           "Liechtenstein"
[127] "Lithuania"                       "Low income"
[129] "Lower middle income"             "Luxembourg"
[131] "Macao"                           "Madagascar"
[133] "Malawi"                          "Malaysia"
[135] "Maldives"                        "Mali"
[137] "Malta"                           "Marshall Islands"
```

```
[139] "Martinique"                       "Mauritania"
[141] "Mauritius"                        "Mayotte"
[143] "Mexico"                           "Micronesia (country)"
[145] "Moldova"                          "Monaco"
[147] "Mongolia"                         "Montenegro"
[149] "Montserrat"                       "Morocco"
[151] "Mozambique"                       "Myanmar"
[153] "Namibia"                          "Nauru"
[155] "Nepal"                            "Netherlands"
[157] "New Caledonia"                    "New Zealand"
[159] "Nicaragua"                        "Niger"
[161] "Nigeria"                          "Niue"
[163] "North America"                    "North Korea"
[165] "North Macedonia"                  "Northern Cyprus"
[167] "Northern Ireland"                 "Northern Mariana Islands"
[169] "Norway"                           "Oceania"
[171] "Oman"                             "Pakistan"
[173] "Palau"                            "Palestine"
[175] "Panama"                           "Papua New Guinea"
[177] "Paraguay"                         "Peru"
[179] "Philippines"                      "Pitcairn"
[181] "Poland"                           "Portugal"
[183] "Puerto Rico"                      "Qatar"
[185] "Reunion"                          "Romania"
[187] "Russia"                           "Rwanda"
[189] "Saint Barthelemy"                 "Saint Helena"
[191] "Saint Kitts and Nevis"            "Saint Lucia"
[193] "Saint Martin (French part)"       "Saint Pierre and Miquelon"
[195] "Saint Vincent and the Grenadines" "Samoa"
[197] "San Marino"                       "Sao Tome and Principe"
[199] "Saudi Arabia"                     "Scotland"
[201] "Senegal"                          "Serbia"
[203] "Seychelles"                       "Sierra Leone"
[205] "Singapore"                        "Sint Maarten (Dutch part)"
[207] "Slovakia"                         "Slovenia"
[209] "Solomon Islands"                  "Somalia"
[211] "South Africa"                     "South America"
[213] "South Korea"                      "South Sudan"
[215] "Spain"                            "Sri Lanka"
[217] "Sudan"                            "Suriname"
[219] "Sweden"                           "Switzerland"
[221] "Syria"                            "Taiwan"
[223] "Tajikistan"                       "Tanzania"
[225] "Thailand"                         "Timor"
[227] "Togo"                             "Tokelau"
[229] "Tonga"                            "Trinidad and Tobago"
[231] "Tunisia"                          "Turkey"
[233] "Turkmenistan"                     "Turks and Caicos Islands"
[235] "Tuvalu"                           "Uganda"
[237] "Ukraine"                          "United Arab Emirates"
[239] "United Kingdom"                   "United States"
[241] "United States Virgin Islands"     "Upper middle income"
[243] "Uruguay"                          "Uzbekistan"
[245] "Vanuatu"                          "Vatican"
```

```
[247] "Venezuela"                        "Vietnam"
[249] "Wales"                            "Wallis and Futuna"
[251] "Western Sahara"                   "World"
[253] "Yemen"                            "Zambia"
[255] "Zimbabwe"
```

```
names(cd)
```

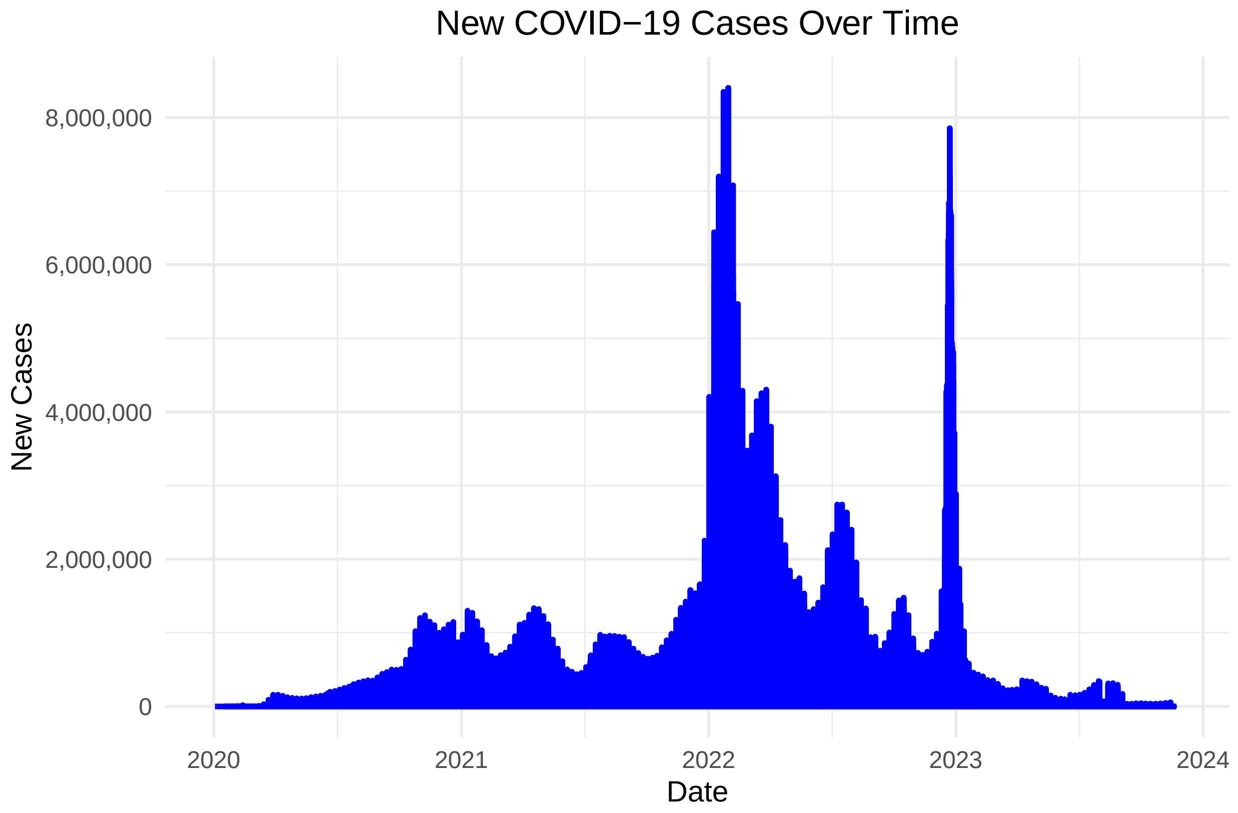
```
 [1] "iso_code"
 [2] "continent"
 [3] "location"
 [4] "date"
 [5] "total_cases"
 [6] "new_cases"
 [7] "new_cases_smoothed"
 [8] "total_deaths"
 [9] "new_deaths"
[10] "new_deaths_smoothed"
[11] "total_cases_per_million"
[12] "new_cases_per_million"
[13] "new_cases_smoothed_per_million"
[14] "total_deaths_per_million"
[15] "new_deaths_per_million"
[16] "new_deaths_smoothed_per_million"
[17] "reproduction_rate"
[18] "icu_patients"
[19] "icu_patients_per_million"
[20] "hosp_patients"
[21] "hosp_patients_per_million"
[22] "weekly_icu_admissions"
[23] "weekly_icu_admissions_per_million"
[24] "weekly_hosp_admissions"
[25] "weekly_hosp_admissions_per_million"
[26] "total_tests"
[27] "new_tests"
[28] "total_tests_per_thousand"
[29] "new_tests_per_thousand"
[30] "new_tests_smoothed"
[31] "new_tests_smoothed_per_thousand"
[32] "positive_rate"
[33] "tests_per_case"
[34] "tests_units"
[35] "total_vaccinations"
[36] "people_vaccinated"
[37] "people_fully_vaccinated"
[38] "total_boosters"
[39] "new_vaccinations"
[40] "new_vaccinations_smoothed"
[41] "total_vaccinations_per_hundred"
[42] "people_vaccinated_per_hundred"
[43] "people_fully_vaccinated_per_hundred"
[44] "total_boosters_per_hundred"
[45] "new_vaccinations_smoothed_per_million"
[46] "new_people_vaccinated_smoothed"
[47] "new_people_vaccinated_smoothed_per_hundred"
```

```
[48] "stringency_index"
[49] "population_density"
[50] "median_age"
[51] "aged_65_older"
[52] "aged_70_older"
[53] "gdp_per_capita"
[54] "extreme_poverty"
[55] "cardiovasc_death_rate"
[56] "diabetes_prevalence"
[57] "female_smokers"
[58] "male_smokers"
[59] "handwashing_facilities"
[60] "hospital_beds_per_thousand"
[61] "life_expectancy"
[62] "human_development_index"
[63] "population"
[64] "excess_mortality_cumulative_absolute"
[65] "excess_mortality_cumulative"
[66] "excess_mortality"
[67] "excess_mortality_cumulative_per_million"
```

```r
# Plotting new cases over time with formatted y-axis labels
ggplot(cd, aes(x = date, y = new_cases)) +
  geom_line(color = "blue", size = 1) +
  labs(title = "New COVID-19 Cases Over Time",
       x = "Date",
       y = "New Cases",
       caption = "Data Source: OWID") +
  scale_y_continuous(labels = scales::comma) + # Formats the y-axis labels with commas
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) # Centering the title
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
generated.
```

```
Warning: Removed 147 rows containing missing values (`geom_line()`).
```



Data Source: OWID

```
# 'new_cases' is considered a good proxy for transmission rates because it
# directly reflects the number of new COVID-19
# cases reported, offering insights into the current spread of the virus. This metric is timely
# and typically available across different regions, providing a near-real-time
# snapshot of the pandemic's progression.

# However, there are limitations to consider:
# - The number of new cases can be influenced by the rate and criteria of testing.
#   Increased testing may lead to more cases being detected.
# - Reporting delays and practices can vary, potentially leading to fluctuations
#   that don't necessarily represent actual changes in transmission.
# - Asymptomatic or undetected cases mean that the actual number of new infections
#   could be higher than reported.
# - Changes in testing protocols or public health policies can also impact the
#   number of cases detected over time.
```

Behavioral Dataset Exploratory Data Analysis (EDA)

```
# This subsection is dedicated to performing an Exploratory Data Analysis (EDA)
# on the primary COVID-19 behaviors dataset.
```

```r
# Purpose:
# The following code conducts an initial assessment of the primary behaviors dataset ('bd').
# This EDA aims to uncover the dataset's basic structure, identify any immediate data quality issues,
# and prepare the data for more detailed analysis.

# Steps:
# 1. Preview the data to get a sense of the information contained in the first few rows.
# 2. Explore the structure of the dataset, including data types and the first few entries, to
#    understand how the data is organized.
# 3. Generate summary statistics for each column to capture central tendency, dispersion, and
#    the presence of NA values, which will be crucial for assessing data quality.
# 4. Check for missing values across the dataset to determine if any imputation or data cleaning
#    steps are necessary.
# 5. Identify and count duplicate rows to ensure the uniqueness of data points in the dataset.

# Code Execution:
# The results from these exploratory steps will inform how we handle data preprocessing and guide
# the analytical techniques applied in subsequent stages of the analysis.

# Read COVID-19 behaviors data into 'bd' dataframe
bd <- read_csv("covid_behaviors (1).csv")
```

```
Rows: 291 Columns: 32
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (1): Country
dbl (31): Days since outbreak, Counts.Household contacts, Counts.Total conta...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Preview the first few rows of the dataset
head(bd)
```

```
# A tibble: 6 x 32
  Country   `Days since outbreak` Counts.Household cont~1 Counts.Total contact~2
  <chr>                     <dbl>                   <dbl>                  <dbl>
1 Australia                    85                     2                      8.2
2 Australia                   115                     2.3                    8
3 Australia                   146                     2.5                   14.3
4 Australia                   176                     2.4                   13
5 Australia                   207                     2.5                   14.4
6 Australia                   238                     2.1                   15.5
# i abbreviated names: 1: `Counts.Household contacts`,
#   2: `Counts.Total contacts`
# i 28 more variables: `Counts.Times left home` <dbl>, Counts.Handwashes <dbl>,
#   `Scores.Isolate.Willingness if symptoms` <dbl>,
#   `Scores.Isolate.Willingness if advised` <dbl>,
#   Scores.Isolate.Difficulty <dbl>, `Scores.Masks.Outside home` <dbl>,
#   `Scores.Masks.Grocery store` <dbl>, ...
```

```r
# View the structure of the dataset: column names, data types, and the first
# few entries in each column
str(bd)
```

```
spc_tbl_ [291 x 32] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
```

```
$ Country                                  : chr [1:291] "Australia" "Australia" "Australia" "Australia
$ Days since outbreak                      : num [1:291] 85 115 146 176 207 238 268 299 329 360 ...
$ Counts.Household contacts                : num [1:291] 2 2.3 2.5 2.4 2.5 2.1 2.2 3.2 2.3 2.2 ...
$ Counts.Total contacts                    : num [1:291] 8.2 8 14.3 13 14.4 15.5 17.7 19.9 19.5 20.5 .
$ Counts.Times left home                   : num [1:291] 1 1.1 1.3 1.4 1.3 1.4 1.4 1.5 1.4 1.4 ...
$ Counts.Handwashes                        : num [1:291] 11.5 10.3 9.6 10.5 10.2 9.6 10 9 10.3 10.8 ..
$ Scores.Isolate.Willingness if symptoms   : num [1:291] 85.8 83 80.8 83.7 84.2 83.2 81 78.5 78 80.2 .
$ Scores.Isolate.Willingness if advised    : num [1:291] 14.3 13.6 13.9 14 13.8 15.7 14.7 16.3 16.6 16
$ Scores.Isolate.Difficulty                : num [1:291] 91.9 90.3 89.3 90.5 90.7 88.8 89.4 88.5 89.4 8
$ Scores.Masks.Outside home                : num [1:291] 23.9 24.1 22.2 24.9 49.5 48.1 43.2 45.2 41.7 5
$ Scores.Masks.Grocery store               : num [1:291] 0 0 16.9 21.7 46.9 46 41.2 43.1 41.6 55.8 ...
$ Scores.Masks.Clothing store              : num [1:291] 0 0 15.8 20.5 41.4 39.5 35.3 39.7 36.7 49.2 .
$ Scores.Masks.Work                        : num [1:291] 0 0 22 26.7 41.8 40.3 37.4 40.1 32.1 40.7 ...
$ Scores.Masks.Public transport            : num [1:291] 0 0 19.1 24.3 43.3 42.4 38.5 42.3 39.5 50.1 .
$ Scores.Avoidance.Symptomatic people      : num [1:291] 86.6 82.2 77.3 78.5 77 74.2 72.4 71.8 73.3 77
$ Scores.Avoidance.Going out                : num [1:291] 77.6 67.9 51.6 51.3 55.7 49.1 42.2 39.7 37.9 4
$ Scores.Avoidance.Healthcare settings     : num [1:291] 72.9 66.3 54.4 51.9 54.4 50.2 46.8 44.6 48.1 5
$ Scores.Avoidance.Public transport        : num [1:291] 83.3 80.2 70.4 69.9 70 65.6 61.8 60.1 61.6 63
$ Scores.Avoidance.Working outside home    : num [1:291] 59.4 54.6 43.3 41.2 42.1 40.9 30 40.9 35.5 40
$ Scores.Avoidance.Children going to school: num [1:291] 79.1 70 40.9 39.4 38.4 40.5 27.3 30.7 30.8 32
$ Scores.Avoidance.Having guests           : num [1:291] 87.2 80.6 61.8 60.2 65.9 59.2 53.4 50.5 47.5 5
$ Scores.Avoidance.Gatherings.Small        : num [1:291] 85.7 76.7 54.5 53 59.9 52.9 46.7 43.2 39.9 45
$ Scores.Avoidance.Gatherings.Medium       : num [1:291] 90.9 84.3 63.8 64.4 67.3 60.9 55.2 52.2 47.4 5
$ Scores.Avoidance.Gatherings.Large        : num [1:291] 92.7 89.2 76.1 76.4 76.7 72.6 66.9 63.3 58.7 6
$ Scores.Avoidance.Crowded areas           : num [1:291] 89.8 84.7 75.7 77.5 77.5 72.5 68.7 66.6 64.5 6
$ Scores.Avoidance.Shops                   : num [1:291] 60.2 53.8 40.4 40.5 44.1 37.8 31 28.6 28.6 33
$ Scores.Precautions.Cleaned surfaces      : num [1:291] 64.4 60.6 57.4 57.9 59.1 58.1 56.4 54.7 55 0
$ Scores.Precautions.Covered mouth sneeze  : num [1:291] 91.8 90.3 90.6 91.2 92.8 90.2 92.1 91.3 89.7 9
$ Scores.Precautions.Used hand sanitiser   : num [1:291] 72.9 75.1 77 81.2 80.9 78.8 79.1 80.3 78.8 81
$ Scores.Outlooks.Covid is dangerous       : num [1:291] 0 0 49.2 55.7 54 52.9 49.1 47.6 52.5 49.9 ...
$ Scores.Outlooks.Likely to get covid      : num [1:291] 0 0 18.9 24.6 20.7 19.6 18 19.8 19.4 17.8 ...
$ Scores.Outlooks.Life greatly impacted    : num [1:291] 0 0 46.1 49.2 52.2 51.1 43.8 45.7 41.3 43.7 .
- attr(*, "spec")=
 .. cols(
 ..   Country = col_character(),
 ..   `Days since outbreak` = col_double(),
 ..   `Counts.Household contacts` = col_double(),
 ..   `Counts.Total contacts` = col_double(),
 ..   `Counts.Times left home` = col_double(),
 ..   Counts.Handwashes = col_double(),
 ..   `Scores.Isolate.Willingness if symptoms` = col_double(),
 ..   `Scores.Isolate.Willingness if advised` = col_double(),
 ..   Scores.Isolate.Difficulty = col_double(),
 ..   `Scores.Masks.Outside home` = col_double(),
 ..   `Scores.Masks.Grocery store` = col_double(),
 ..   `Scores.Masks.Clothing store` = col_double(),
 ..   Scores.Masks.Work = col_double(),
 ..   `Scores.Masks.Public transport` = col_double(),
 ..   `Scores.Avoidance.Symptomatic people` = col_double(),
 ..   `Scores.Avoidance.Going out` = col_double(),
 ..   `Scores.Avoidance.Healthcare settings` = col_double(),
 ..   `Scores.Avoidance.Public transport` = col_double(),
 ..   `Scores.Avoidance.Working outside home` = col_double(),
 ..   `Scores.Avoidance.Children going to school` = col_double(),
 ..   `Scores.Avoidance.Having guests` = col_double(),
 ..   Scores.Avoidance.Gatherings.Small = col_double(),
 ..   Scores.Avoidance.Gatherings.Medium = col_double(),
 ..   Scores.Avoidance.Gatherings.Large = col_double(),
 ..   `Scores.Avoidance.Crowded areas` = col_double(),
 ..   Scores.Avoidance.Shops = col_double(),
 ..   `Scores.Precautions.Cleaned surfaces` = col_double(),
 ..   `Scores.Precautions.Covered mouth sneeze` = col_double(),
 ..   `Scores.Precautions.Used hand sanitiser` = col_double(),
 ..   `Scores.Outlooks.Covid is dangerous` = col_double(),
 ..   `Scores.Outlooks.Likely to get covid` = col_double(),
 ..   `Scores.Outlooks.Life greatly impacted` = col_double()
 .. )
- attr(*, "problems")=<externalptr>
```

```
# Generate summary statistics for each column
summary(bd)
```

```
   Country           Days since outbreak Counts.Household contacts
 Length:291          Min.   : 85.0       Min.   : 1.500           
 Class :character    1st Qu.:146.0       1st Qu.: 2.300           
 Mode  :character    Median :238.0       Median : 2.900           
                     Mean   :246.2       Mean   : 3.064           
                     3rd Qu.:329.0       3rd Qu.: 3.500           
                     Max.   :480.0       Max.   :10.000           
 Counts.Total contacts Counts.Times left home Counts.Handwashes
 Min.   : 4.00         Min.   :0.600          Min.   : 4.00    
 1st Qu.: 8.20         1st Qu.:1.400          1st Qu.: 8.70    
 Median :11.60         Median :1.700          Median :10.40    
 Mean   :13.27         Mean   :1.711          Mean   :10.38    
 3rd Qu.:16.90         3rd Qu.:2.000          3rd Qu.:11.95    
 Max.   :36.70         Max.   :3.500          Max.   :19.20    
 Scores.Isolate.Willingness if symptoms Scores.Isolate.Willingness if advised
 Min.   :22.00                          Min.   : 5.10                        
 1st Qu.:71.35                          1st Qu.:14.75                        
 Median :79.70                          Median :17.70                        
 Mean   :76.65                          Mean   :19.01                        
 3rd Qu.:85.45                          3rd Qu.:21.55                        
 Max.   :93.30                          Max.   :43.80                        
 Scores.Isolate.Difficulty Scores.Masks.Outside home Scores.Masks.Grocery store
 Min.   :65.30             Min.   : 4.20             Min.   : 0.00             
 1st Qu.:79.55             1st Qu.:66.40             1st Qu.: 7.30             
 Median :83.40             Median :87.80             Median :88.20             
 Mean   :82.73             Mean   :73.97             Mean   :61.27             
 3rd Qu.:87.20             3rd Qu.:94.25             3rd Qu.:94.60             
 Max.   :95.70             Max.   :98.20             Max.   :98.60             
 Scores.Masks.Clothing store Scores.Masks.Work Scores.Masks.Public transport
 Min.   : 0.00               Min.   : 0.00     Min.   : 0.00                
 1st Qu.: 6.40               1st Qu.:10.90     1st Qu.:12.70                
 Median :76.20               Median :62.90     Median :74.60                
 Mean   :57.77               Mean   :51.72     Mean   :58.83                
 3rd Qu.:92.50               3rd Qu.:86.50     3rd Qu.:92.70                
 Max.   :98.40               Max.   :95.40     Max.   :98.40                
 Scores.Avoidance.Symptomatic people Scores.Avoidance.Going out
 Min.   :58.70                       Min.   :14.70             
```

```
1st Qu.:76.55                        1st Qu.:40.35
Median :83.40                        Median :52.10
Mean   :81.95                        Mean   :53.44
3rd Qu.:88.05                        3rd Qu.:68.35
Max.   :95.10                        Max.   :90.70
Scores.Avoidance.Healthcare settings Scores.Avoidance.Public transport
Min.   :36.20                         Min.   :25.20
1st Qu.:56.65                         1st Qu.:62.60
Median :66.60                         Median :71.70
Mean   :65.71                         Mean   :69.53
3rd Qu.:75.05                         3rd Qu.:78.25
Max.   :93.20                         Max.   :95.60
Scores.Avoidance.Working outside home
Min.   :15.00
1st Qu.:32.85
Median :39.90
Mean   :42.24
3rd Qu.:49.55
Max.   :86.70
Scores.Avoidance.Children going to school Scores.Avoidance.Having guests
Min.   : 5.50                             Min.   :32.00
1st Qu.:21.75                             1st Qu.:54.75
Median :39.40                             Median :68.40
Mean   :44.17                             Mean   :67.15
3rd Qu.:67.15                             3rd Qu.:80.00
Max.   :93.40                             Max.   :96.70
Scores.Avoidance.Gatherings.Small Scores.Avoidance.Gatherings.Medium
Min.   :23.50                     Min.   :34.70
1st Qu.:48.80                     1st Qu.:63.05
Median :60.70                     Median :72.40
Mean   :61.38                     Mean   :71.27
3rd Qu.:75.80                     3rd Qu.:82.45
Max.   :92.80                     Max.   :96.20
Scores.Avoidance.Gatherings.Large Scores.Avoidance.Crowded areas
Min.   :49.50                     Min.   :53.50
1st Qu.:75.25                     1st Qu.:73.70
Median :82.70                     Median :80.80
Mean   :80.84                     Mean   :79.74
3rd Qu.:88.10                     3rd Qu.:87.05
Max.   :97.30                     Max.   :97.10
Scores.Avoidance.Shops Scores.Precautions.Cleaned surfaces
Min.   :15.20          Min.   : 0.00
1st Qu.:34.15          1st Qu.:28.25
Median :46.90          Median :53.30
Mean   :47.03          Mean   :44.70
3rd Qu.:59.25          3rd Qu.:66.70
Max.   :87.40          Max.   :85.00
Scores.Precautions.Covered mouth sneeze Scores.Precautions.Used hand sanitiser
Min.   :79.80                           Min.   :31.50
1st Qu.:87.60                           1st Qu.:69.00
Median :90.80                           Median :79.30
Mean   :90.31                           Mean   :76.42
3rd Qu.:93.20                           3rd Qu.:85.40
Max.   :97.80                           Max.   :94.40
```

```
 Scores.Outlooks.Covid is dangerous Scores.Outlooks.Likely to get covid
 Min.   : 0.00                      Min.   : 0.00
 1st Qu.:33.20                      1st Qu.:13.65
 Median :45.90                      Median :21.20
 Mean   :42.08                      Mean   :19.59
 3rd Qu.:58.50                      3rd Qu.:26.80
 Max.   :87.30                      Max.   :47.70
 Scores.Outlooks.Life greatly impacted
 Min.   : 0.00
 1st Qu.:31.00
 Median :51.60
 Mean   :42.54
 3rd Qu.:59.20
 Max.   :75.70
```

```r
# Identify missing values in the dataset - no missing values
sum(is.na(bd))
```

```
[1] 0
```

```r
# Check for duplicate rows - no duplicate rows
sum(duplicated(bd))
```

```
[1] 0
```

```r
# Get the column names of the behaviors dataset
column_names <- names(bd)
print(column_names)
```

```
 [1] "Country"
 [2] "Days since outbreak"
 [3] "Counts.Household contacts"
 [4] "Counts.Total contacts"
 [5] "Counts.Times left home"
 [6] "Counts.Handwashes"
 [7] "Scores.Isolate.Willingness if symptoms"
 [8] "Scores.Isolate.Willingness if advised"
 [9] "Scores.Isolate.Difficulty"
[10] "Scores.Masks.Outside home"
[11] "Scores.Masks.Grocery store"
[12] "Scores.Masks.Clothing store"
[13] "Scores.Masks.Work"
[14] "Scores.Masks.Public transport"
[15] "Scores.Avoidance.Symptomatic people"
[16] "Scores.Avoidance.Going out"
[17] "Scores.Avoidance.Healthcare settings"
[18] "Scores.Avoidance.Public transport"
[19] "Scores.Avoidance.Working outside home"
[20] "Scores.Avoidance.Children going to school"
[21] "Scores.Avoidance.Having guests"
[22] "Scores.Avoidance.Gatherings.Small"
[23] "Scores.Avoidance.Gatherings.Medium"
[24] "Scores.Avoidance.Gatherings.Large"
[25] "Scores.Avoidance.Crowded areas"
[26] "Scores.Avoidance.Shops"
[27] "Scores.Precautions.Cleaned surfaces"
```

```
[28] "Scores.Precautions.Covered mouth sneeze"
[29] "Scores.Precautions.Used hand sanitiser"
[30] "Scores.Outlooks.Covid is dangerous"
[31] "Scores.Outlooks.Likely to get covid"
[32] "Scores.Outlooks.Life greatly impacted"
```

```r
# Find common countries that are present in both datasets
common_countries <- intersect(unique(cd$location), unique(bd$Country))
```

Section 2: Dataset Reconciliation and Country Clustering

```r
# Given the size of the datasets and the limited time for this assignment this analysis will focus
# on ten randomly selected countries. The hope is that by randomly selecting countries that appear
# in both datasets the results will be managable but the selection process will not introduce bias.

# Random sample from the list of common countries
set.seed(123)  # Setting a seed for reproducibility
random_common_countries <- sample(common_countries, 10)

# Check the selected countries
print(random_common_countries)
```

```
 [1] "Mexico"       "Saudi Arabia" "Malaysia"     "Canada"       "India"       
 [6] "Philippines"  "Spain"        "Indonesia"    "Denmark"      "Singapore"   
```

```r
# Sub-setting the dataframes by randomly selected countries.
# Subset the COVID data for the selected countries
cd_subset <- cd[cd$location %in% random_common_countries, ]

# Subset the behavior data for the selected countries
# Make sure to replace 'Country' with the actual column name for countries in the bd dataframe
bd_subset <- bd[bd$Country %in% random_common_countries, ]
```

Manual Selection Based on Outbreak Start Dates

```r
# Issue:
# The 'Days since outbreak' column in the behavior dataset (bd) presented a challenge for analysis
# due to inconsistent or non-standardized outbreak start dates across different countries. Directly
# comparing behavioral responses between countries became problematic because the relative timelines
# did not align, potentially skewing any comparative analysis.

# Resolution:
# To address this, a manual review of country-specific outbreak start dates was conducted. By
# consulting individual country data files hosted on the GitHub repository, accurate outbreak start
# dates were determined for each country. Countries were then selected for inclusion in the analysis
# based on whether their outbreak start dates fell within a similar timeframe (within one month of
# each other). This manual curation ensured a more accurate and meaningful comparison of behavioral
# responses during comparable stages of the pandemic response.

# The manual approach, while more time-consuming, provided a level of precision and customization
# in the selection process. It allowed for the identification of a subset of countries with
# closely aligned outbreak timelines, thereby facilitating a more robust and reliable comparative
# analysis of behavioral data.
```

Country Selection Based on Proximal Outbreak Start Dates

```
# After a manual review of country-specific start dates for the COVID-19 outbreak,
# countries were categorized into groups based on the similarity of their outbreak onset.
# This categorization ensures that behavioral responses are compared during equivalent
# stages of the pandemic, allowing for more accurate cross-country comparisons.

# The following groups were identified:
# - Group 1 (Start Dates: February to March): Canada, Spain, Mexico, Singapore
#   These countries experienced the start of their outbreaks within one month of each other,
#   providing a comparable time frame for early pandemic behaviors.

# - Group 2 (Start Dates: January to February): India, Canada, Spain
#   Although Canada and Spain appear in both Group 1 and Group 2, the inclusion criteria
#   for Group 2 is based on a slightly earlier phase, capturing the very onset of the pandemic.

# - Group 3 (Start Dates: August to September):
#   Indonesia, Saudi Arabia, Philippines, Denmark, Malaysia
#   This group represents countries where the outbreak was recognized later, which may reflect
#   different stages of public awareness and response.

# Analysis will proceed with these groups, examining behavioral data within each group to
# assess patterns and responses to the pandemic. This approach acknowledges the temporal
# context of behavioral data, ensuring that findings are not confounded by vastly different
# stages of pandemic progression.
```

```
# Check if there is a "World" or similar entry indicating global data
any(cd$location == "World")
```

```
[1] TRUE
```

```
# Extract global data
global_cases <- cd[cd$location == "World", ]

# Ensure that the 'date' and 'new_cases' columns are correctly named and formatted
global_cases$date <- as.Date(global_cases$date)
```

```r
# Convert the outbreak start date of each group to Date format
group1_start <- as.Date("2020-02-15")  # replace with actual group start date
group2_start <- as.Date("2020-01-10")
group3_start <- as.Date("2020-08-20")
```

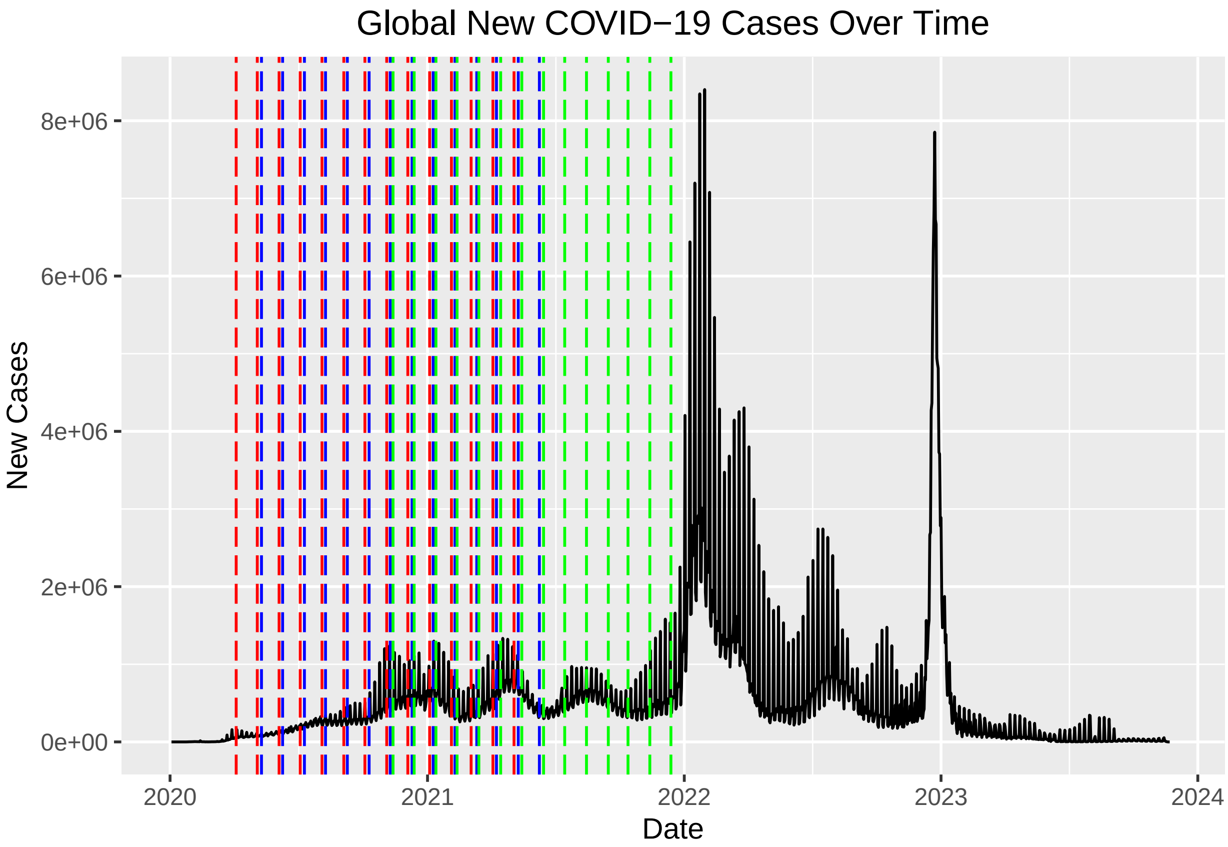
```r
# These are the 'Days since outbreak' for group 1 you've listed
group1_days_since_outbreak <- c(85, 115, 146, 176, 207, 238, 268, 299, 329, 360, 391, 419, 450, 480)
```

```r
# Convert these days to actual dates by adding them to the group's start date
group1_dates <- group1_start + group1_days_since_outbreak
group2_dates <- group2_start + group1_days_since_outbreak
group3_dates <- group3_start + group1_days_since_outbreak
```

```r
# Initialize the plot with global new cases
p <- ggplot(global_cases, aes(x = date, y = new_cases)) +
  geom_line() +
  labs(title = "Global New COVID-19 Cases Over Time", x = "Date", y = "New Cases") +
  theme(plot.title = element_text(hjust = 0.5))
```

```r
# Add vertical lines for group 1
for(i in group1_dates) {
  p <- p + geom_vline(xintercept = as.numeric(i), color = "blue", linetype = "longdash")
}

# Add vertical lines for group 2
for(i in group2_dates) {
  p <- p + geom_vline(xintercept = as.numeric(i), color = "red", linetype = "longdash")
}

# Add vertical lines for group 3
for(i in group3_dates) {
  p <- p + geom_vline(xintercept = as.numeric(i), color = "green", linetype = "longdash")
}

# Print the plot
print(p)
```

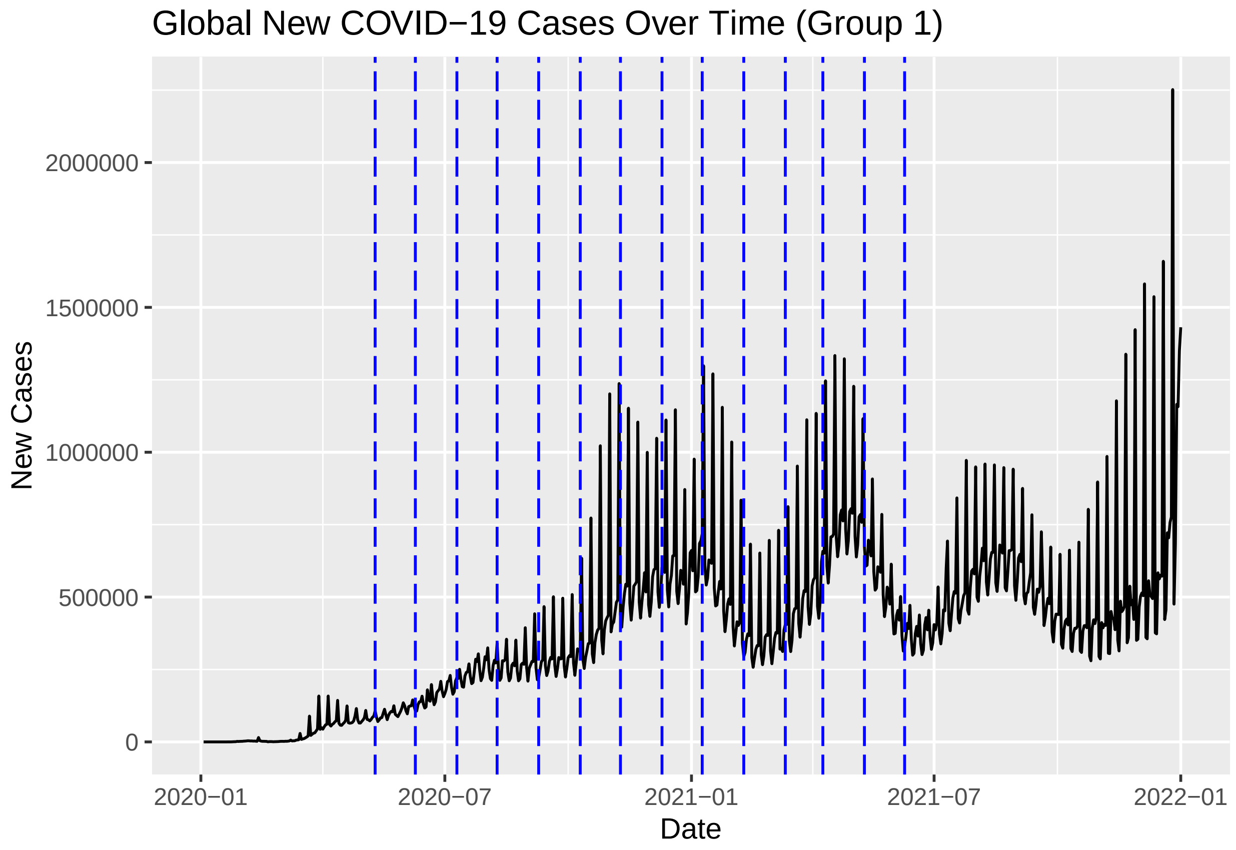
```
Warning: Removed 7 rows containing missing values (`geom_line()`).
```



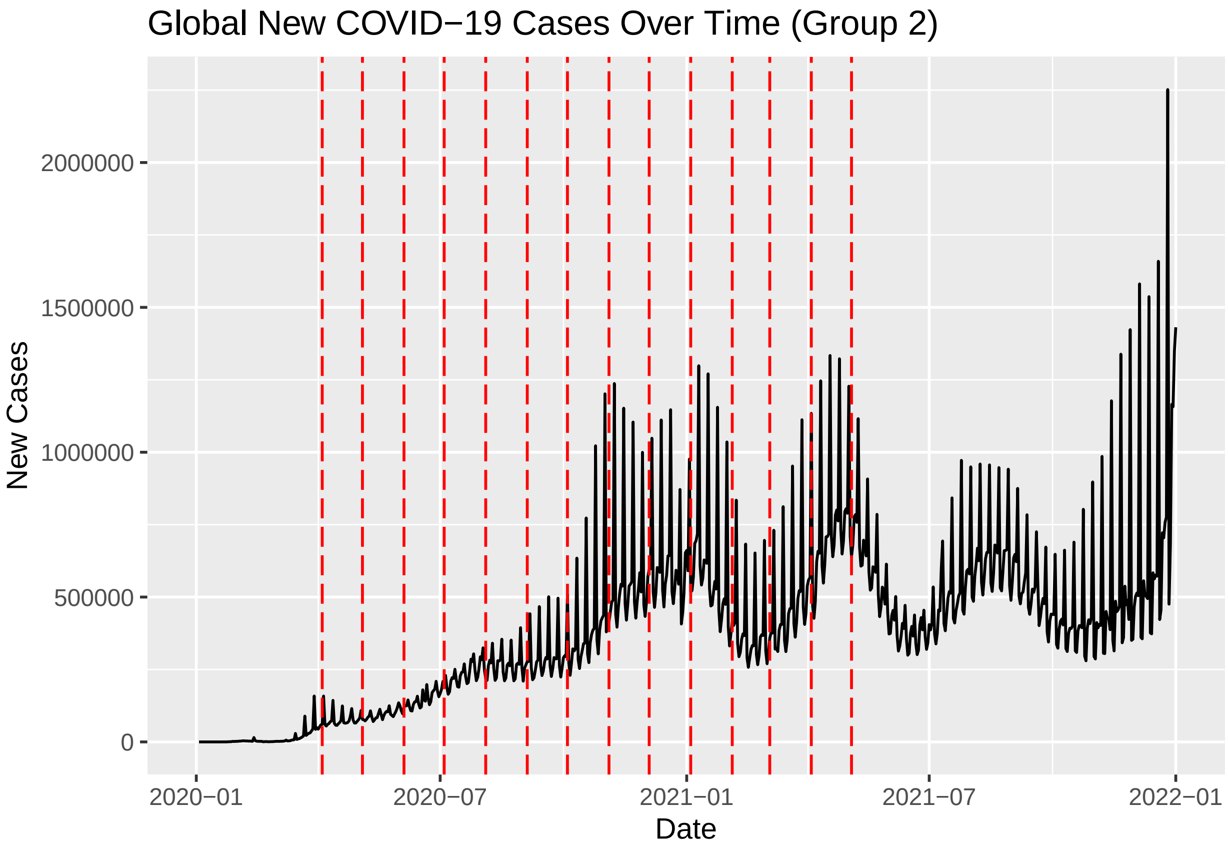
```
cd$date <- as.Date(cd$date)

# Create subsets for each group based on the dates we've calculated
cd_group1 <- cd[cd$date %in% group1_dates, ]
cd_group2 <- cd[cd$date %in% group2_dates, ]
cd_group3 <- cd[cd$date %in% group3_dates, ]
```

```
# Filter global_cases to include dates up to January 2022
global_cases_filtered <- global_cases %>%
  filter(date <= as.Date("2022-01-01"))
```

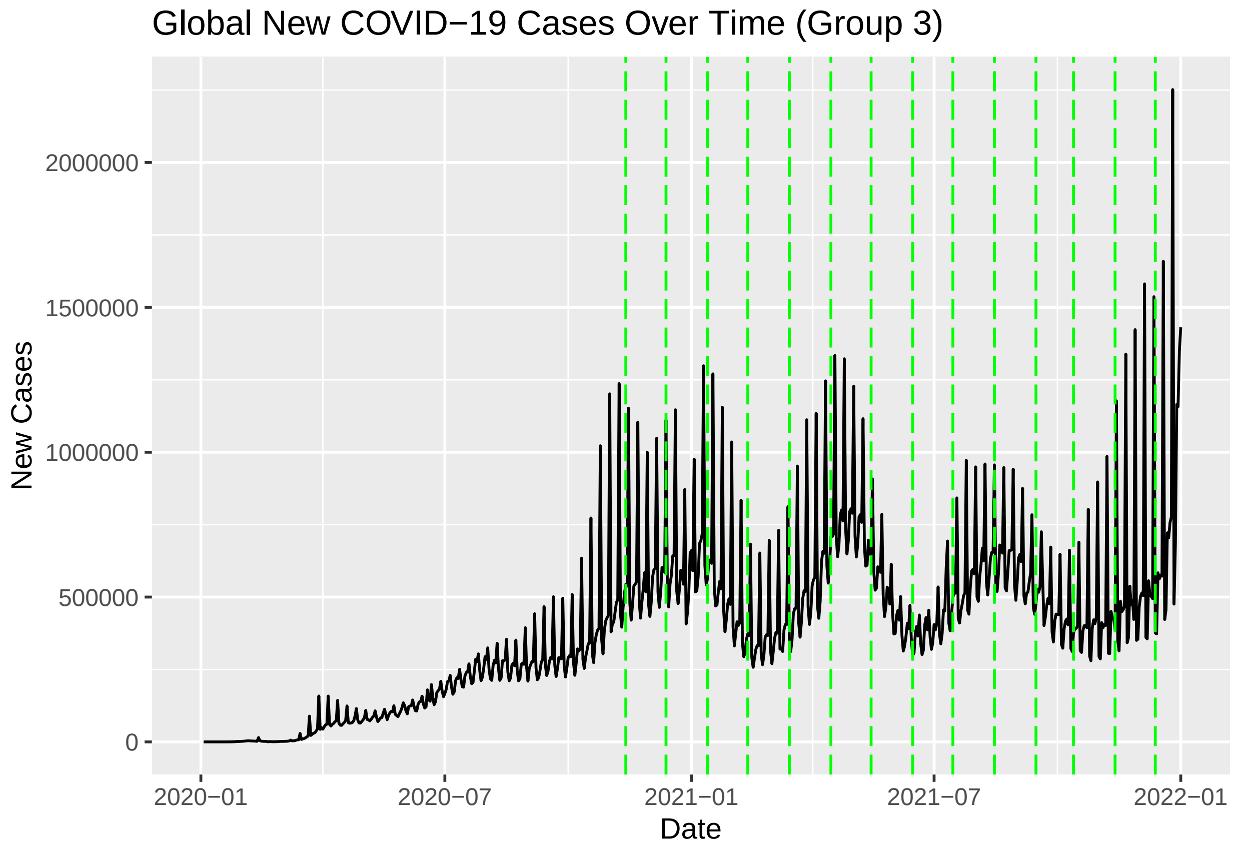
```
# Plot for Group 1 with dates up to January 2022
p_group1 <- ggplot(global_cases_filtered, aes(x = date, y = new_cases)) +
  geom_line() +
  geom_vline(xintercept = as.numeric(group1_dates), color = "blue", linetype = "longdash") +
  labs(title = "Global New COVID-19 Cases Over Time (Group 1)", x = "Date", y = "New Cases") +
  xlim(as.Date("2020-01-01"), as.Date("2022-01-01"))  # Set x-axis limits

print(p_group1)
```



```
# Plot for Group 2 with dates up to January 2022
p_group2 <- ggplot(global_cases_filtered, aes(x = date, y = new_cases)) +
  geom_line() +
  geom_vline(xintercept = as.numeric(group2_dates), color = "red", linetype = "longdash") +
  labs(title = "Global New COVID-19 Cases Over Time (Group 2)", x = "Date", y = "New Cases") +
  xlim(as.Date("2020-01-01"), as.Date("2022-01-01"))  # Set x-axis limits

print(p_group2)
```



```r
# Plot for Group 3 with dates up to January 2022
p_group3 <- ggplot(global_cases_filtered, aes(x = date, y = new_cases)) +
  geom_line() +
  geom_vline(xintercept = as.numeric(group3_dates), color = "green", linetype = "longdash") +
  labs(title = "Global New COVID-19 Cases Over Time (Group 3)", x = "Date", y = "New Cases") +
  xlim(as.Date("2020-01-01"), as.Date("2022-01-01"))  # Set x-axis limits

print(p_group3)
```



Section 3: Dataset Analysis

```
# This section presents the analysis of the COVID-19 behaviors dataset, focusing
# on two specific research questions.
# The analyses leverage a cluster-based approach, where each cluster represents
# a group of countries with similar outbreak start dates, allowing for meaningful
# comparisons of behavioral responses within these clusters.

# The research questions addressed are:
# 1. How did the willingness to self-isolate change throughout the pandemic in different countries?
#    - This question is explored by analyzing self-isolation willingness scores
#      over time in a selected cluster of countries.
#      The analysis focuses on both the willingness to self-isolate if symptoms
#      are present and if advised, providing insights into public sentiment
#      evolution during the pandemic.

# 2. Which countries reported the highest levels of compliance with mask-wearing guidelines?
#    - This question is addressed by comparing mask-wearing compliance levels in
#      ten selected countries.
#      The analysis involves transforming raw mask-wearing scores into a Likert scale
#      and creating visualizations for various contexts of mask-wearing
#      (outside home, grocery store, clothing store, work, public transport).

# The methodologies applied in these analyses aim to strike a balance between
# depth and practicality, considering the dataset size and project timeframe.
# The visualizations generated provide insights into how behavioral patterns
# in response to the pandemic evolved and varied across different countries and timeframes.

# Detailed comments are provided in each subsection to guide through the steps of data
# preparation, transformation, and visualization, ensuring clarity and reproducibility
# of the analysis.

# Willingness to Self-Isolate Analysis:
# - The analysis involves data preparation steps such as renaming columns,
#   selecting relevant data, and transforming 'Days since outbreak' into actual dates.
# - Time series plots are created for each group, illustrating changes in willingness
#   to self-isolate over time, accompanied by global transmission rates to provide context.

# Mask-Wearing Compliance Analysis:
# - The analysis begins with data preparation, including renaming columns and filtering
#   for selected countries.
# - Scores are converted to a Likert scale, and average mask-wearing scores for
#   different contexts are calculated.
# - Bar charts are created for each context of mask-wearing, displaying average
#   scores in a Likert scale format
#   across the selected countries.

# These analyses offer a comprehensive view of behavioral responses during the pandemic, highlighting
# the variations in public adherence to health guidelines and perceptions over time.
```

Q1 - Willingness to self-isolate

```
# Column renaming section using piping. The spaces in the original dataset column
# names were prevent analysis.
bd <- bd %>%
  rename(Days_since_outbreak = `Days since outbreak`)
bd <- bd %>%
  rename(Scores.Isolate.Willingness_if_symptoms = `Scores.Isolate.Willingness if symptoms`)
bd <- bd %>%
  rename(Scores.Isolate.Willingness_if_advised = `Scores.Isolate.Willingness if advised`)
```

```
# Data Preparation: Selecting relevant columns
isolation_data <- bd %>%
  select(Country, Days_since_outbreak, Scores.Isolate.Willingness_if_symptoms, Scores.Isolate.Willingnes
```

```
# Time Series Analysis: Calculating average scores over time for each country
average_isolation_willingness <- isolation_data %>%
  group_by(Country, Days_since_outbreak) %>%
  summarise(Average_Willingness_if_symptoms = mean(Scores.Isolate.Willingness_if_symptoms, na.rm = TRUE
            Average_Willingness_if_advised = mean(Scores.Isolate.Willingness_if_advised, na.rm = TRUE))
```

```
`summarise()` has grouped output by 'Country'. You can override using the
`.groups` argument.
```

```
# Check the structure of the summarised dataframe
str(average_isolation_willingness)
```

```
gropd_df [291 x 4] (S3: grouped_df/tbl_df/tbl/data.frame)
 $ Country                       : chr [1:291] "Australia" "Australia" "Australia" "Australia" ...
 $ Days_since_outbreak           : num [1:291] 85 115 146 176 207 238 268 299 329 360 ...
 $ Average_Willingness_if_symptoms: num [1:291] 85.8 83 80.8 83.7 84.2 83.2 81 78.5 78 80.2 ...
 $ Average_Willingness_if_advised : num [1:291] 14.3 13.6 13.9 14 13.8 15.7 14.7 16.3 16.6 16.1 ...
 - attr(*, "groups")= tibble [29 x 2] (S3: tbl_df/tbl/data.frame)
  ..$ Country: chr [1:29] "Australia" "Brazil" "Canada" "China" ...
  ..$ .rows  : list<int> [1:29]
  .. ..$ : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
  .. ..$ : int [1:6] 15 16 17 18 19 20
  .. ..$ : int [1:14] 21 22 23 24 25 26 27 28 29 30 ...
  .. ..$ : int [1:6] 35 36 37 38 39 40
  .. ..$ : int [1:14] 41 42 43 44 45 46 47 48 49 50 ...
  .. ..$ : int [1:10] 55 56 57 58 59 60 61 62 63 64
  .. ..$ : int [1:14] 65 66 67 68 69 70 71 72 73 74 ...
  .. ..$ : int [1:14] 79 80 81 82 83 84 85 86 87 88 ...
  .. ..$ : int [1:6] 93 94 95 96 97 98
  .. ..$ : int [1:6] 99 100 101 102 103 104
  .. ..$ : int [1:6] 105 106 107 108 109 110
  .. ..$ : int [1:14] 111 112 113 114 115 116 117 118 119 120 ...
  .. ..$ : int [1:14] 125 126 127 128 129 130 131 132 133 134 ...
  .. ..$ : int [1:6] 139 140 141 142 143 144
  .. ..$ : int [1:6] 145 146 147 148 149 150
  .. ..$ : int [1:11] 151 152 153 154 155 156 157 158 159 160 ...
  .. ..$ : int [1:14] 162 163 164 165 166 167 168 169 170 171 ...
  .. ..$ : int [1:6] 176 177 178 179 180 181
  .. ..$ : int [1:6] 182 183 184 185 186 187
  .. ..$ : int [1:14] 188 189 190 191 192 193 194 195 196 197 ...
  .. ..$ : int [1:14] 202 203 204 205 206 207 208 209 210 211 ...
  .. ..$ : int [1:14] 216 217 218 219 220 221 222 223 224 225 ...
  .. ..$ : int [1:14] 230 231 232 233 234 235 236 237 238 239 ...
  .. ..$ : int [1:6] 244 245 246 247 248 249
  .. ..$ : int [1:6] 250 251 252 253 254 255
  .. ..$ : int [1:6] 256 257 258 259 260 261
  .. ..$ : int [1:14] 262 263 264 265 266 267 268 269 270 271 ...
  .. ..$ : int [1:10] 276 277 278 279 280 281 282 283 284 285
  .. ..$ : int [1:6] 286 287 288 289 290 291
  .. ..@ ptype: int(0)
  ..- attr(*, ".drop")= logi TRUE
```

```
# Make sure that 'Days_since_outbreak' is in the expected date or numeric format
str(bd$Days_since_outbreak)
```

```
 num [1:291] 85 115 146 176 207 238 268 299 329 360 ...
```

```
# Make sure there are actual numeric values to plot and not just NA
summary(average_isolation_willingness$Average_Willingness_if_symptoms)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  22.00   71.35   79.70   76.65   85.45   93.30 
```

```r
summary(average_isolation_willingness$Average_Willingness_if_advised)
```

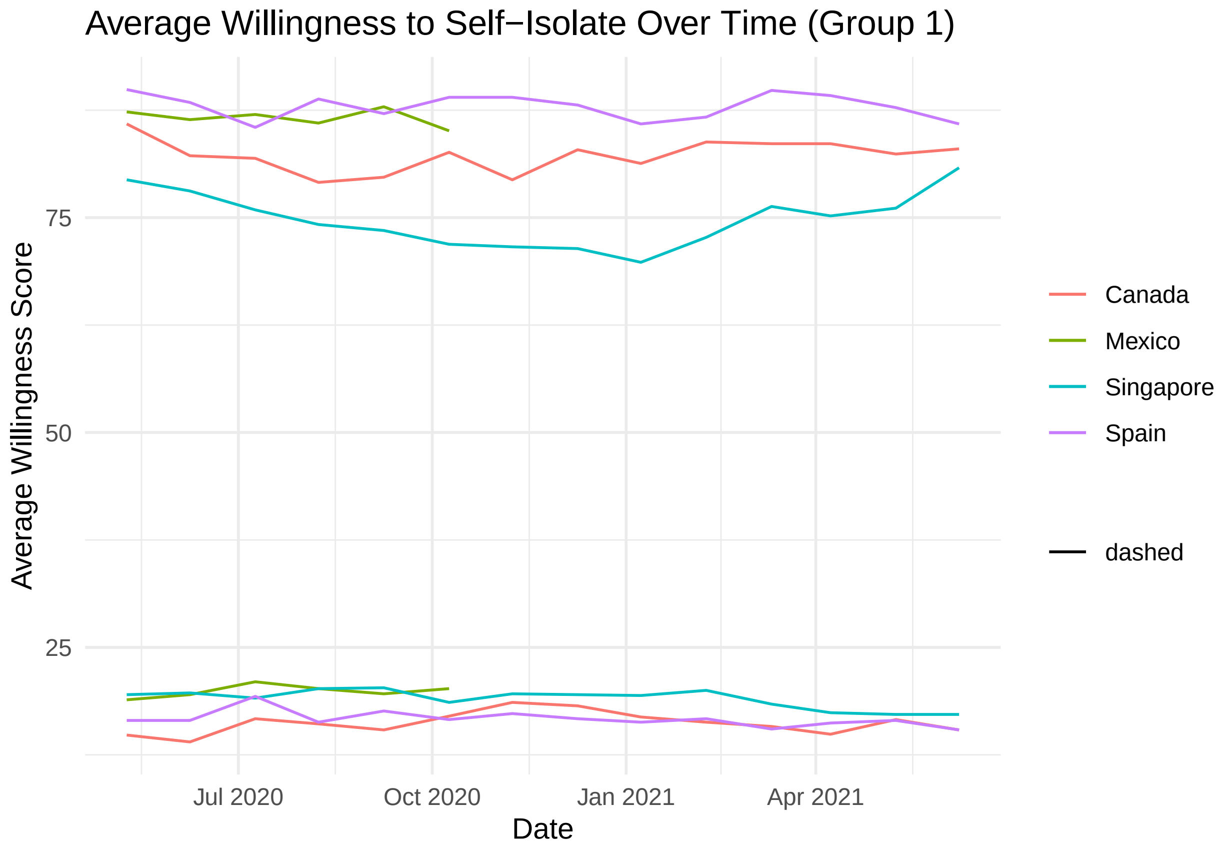
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   5.10   14.75   17.70   19.01   21.55   43.80 
```

```r
# Convert the 'Days_since_outbreak' to actual dates based on the group's start date
group1_start_date <- as.Date("2020-02-15")  # Assuming this is the correct start date for Group 1
bd$Actual_Date <- group1_start_date + bd$Days_since_outbreak - 1  # Correcting for the start date
```

```r
# Filtering for Group 1 countries and relevant dates
group1_isolation_data <- bd %>%
  filter(Country %in% c("Canada", "Spain", "Mexico", "Singapore"),
         Days_since_outbreak %in% c(85, 115, 146, 176, 207, 238, 268, 299,
                                    329, 360, 391, 419, 450, 480)) %>%
  select(Country, Actual_Date, Scores.Isolate.Willingness_if_symptoms,
         Scores.Isolate.Willingness_if_advised)
```

```r
# Plotting the data for Group 1 with individual lines per country
group1_isolation_plot <- ggplot(group1_isolation_data, aes(x = Actual_Date, group = Country)) +
  geom_line(aes(y = Scores.Isolate.Willingness_if_symptoms, color = Country)) +
  geom_line(aes(y = Scores.Isolate.Willingness_if_advised, color = Country, linetype = "dashed")) +
  labs(title = "Average Willingness to Self-Isolate Over Time (Group 1)",
       x = "Date", y = "Average Willingness Score") +
  theme_minimal() +
  theme(legend.title = element_blank())  # Remove the legend title
```

```r
# Print the plot
print(group1_isolation_plot)
```



```r
# Calculate the average transmission rate for each date for the global context
average_global_transmission <- global_cases %>%
  group_by(date) %>%
  summarise(Average_Transmission = mean(new_cases, na.rm = TRUE))
```

Country Cluster 1

```r
# Merge the transmission data with the group 1 behavior data
# Ensure that 'Actual_Date' in your behavior data is converted to Date format
# and aligns with 'date' in transmission data
group1_data_combined <- group1_isolation_data %>%
  left_join(average_global_transmission, by = c("Actual_Date" = "date"))
```

```r
# Check for the existence of 'Average_Transmission' in the combined dataset
# This step is to confirm that the data preparation steps were successful
# and that the Average_Transmission column is present for plotting
if("Average_Transmission" %in% names(group1_data_combined)) {
  print("Average_Transmission exists in the data frame.")
} else {
  print("Average_Transmission does not exist in the data frame. Check the data preparation steps.")
}
```

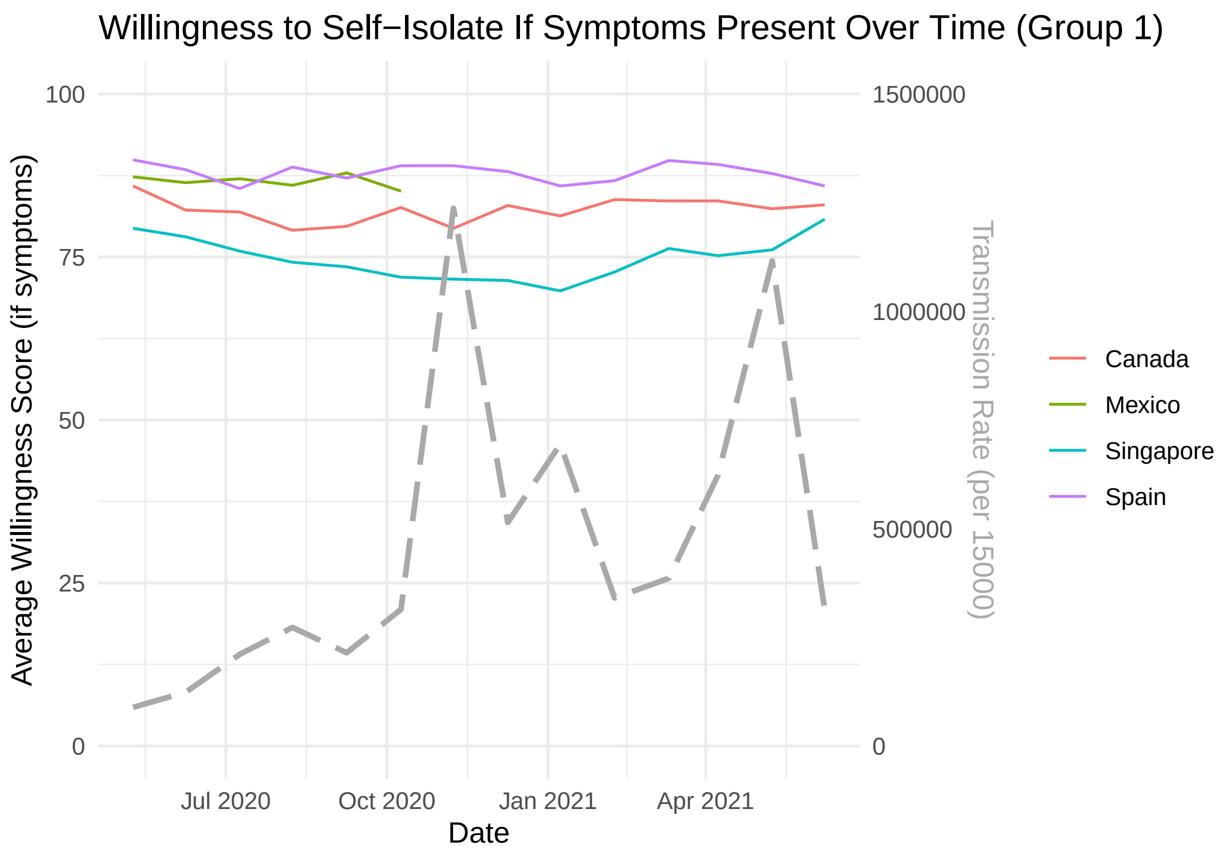
```
[1] "Average_Transmission exists in the data frame."
```

```r
# View the structure of the combined data to confirm column names and data types
# This function will give us an overview of the dataframe structure after the join
str(group1_data_combined)
```

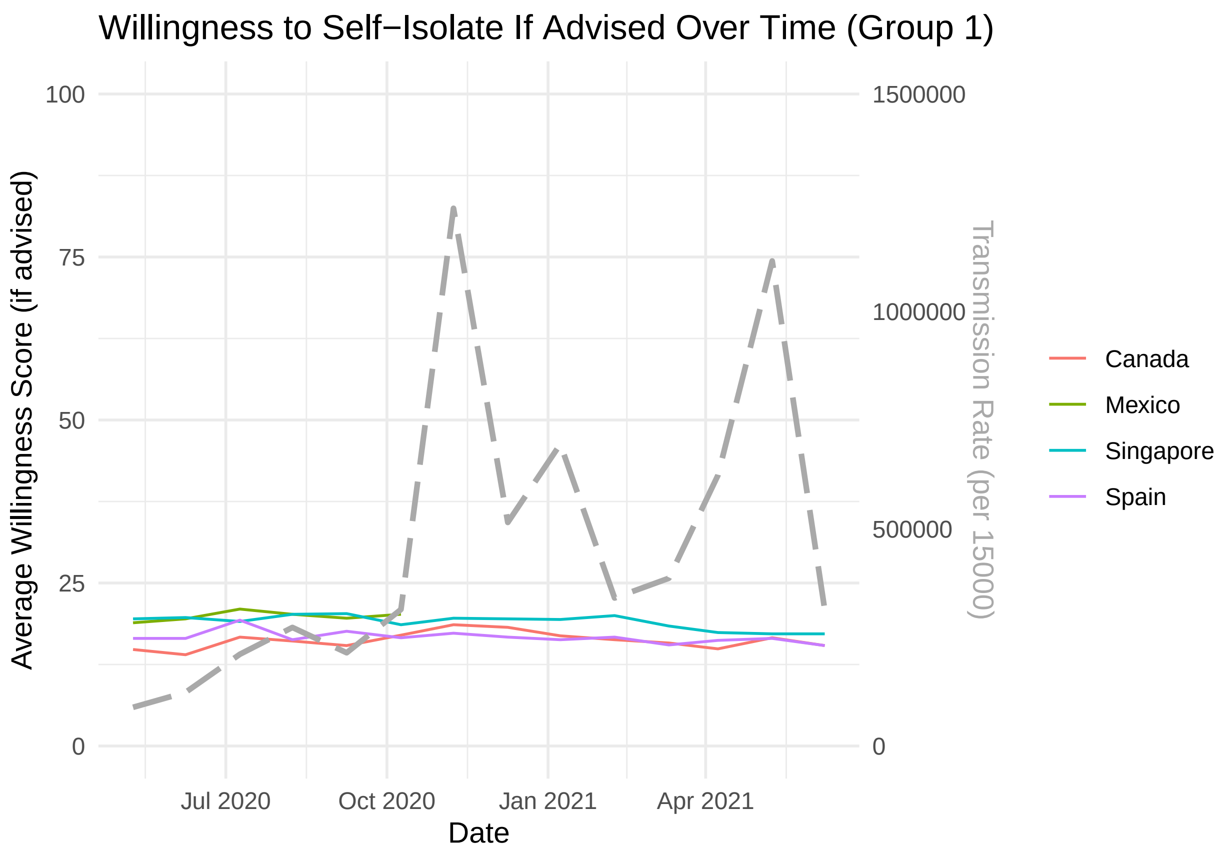
```
tibble [48 x 5] (S3: tbl_df/tbl/data.frame)
 $ Country                               : chr [1:48] "Canada" "Canada" "Canada" "Canada" ...
 $ Actual_Date                           : Date[1:48], format: "2020-05-09" "2020-06-08" ...
 $ Scores.Isolate.Willingness_if_symptoms: num [1:48] 85.9 82.2 81.9 79.1 79.7 82.6 79.4 82.9 81.3 83.8
 $ Scores.Isolate.Willingness_if_advised : num [1:48] 14.8 14 16.7 16.1 15.4 17 18.6 18.2 16.9 16.3 ...
 $ Average_Transmission                  : num [1:48] 89000 123431 211012 272733 214489 ...
```

```r
# Merge the average global transmission data with the group 1 behavior data
# This will append the transmission rate to the behavioral data, allowing for combined analysis
group1_data_combined <- group1_data_combined %>%
  left_join(average_global_transmission, by = c("Actual_Date" = "date"))
```

```r
# First plot: Willingness to self-isolate if symptoms are present
ggplot(group1_data_combined) +
  geom_line(aes(x = Actual_Date, y = Scores.Isolate.Willingness_if_symptoms, color = Country)) +
  geom_line(aes(x = Actual_Date, y = Average_Transmission.x/15000),
            linetype = "longdash", color = "darkgrey", size = 1) +
  scale_y_continuous(
    name = "Average Willingness Score (if symptoms)",
    limits = c(0, 100),
    sec.axis = sec_axis(~ . * 15000, name = "Transmission Rate (per 15000)")
  ) +
  labs(
    title = "Willingness to Self-Isolate If Symptoms Present Over Time (Group 1)",
    x = "Date"
  ) +
  theme_minimal() +
  theme(
    legend.title = element_blank(),
    axis.title.y.right = element_text(color = "darkgrey")
  )
```



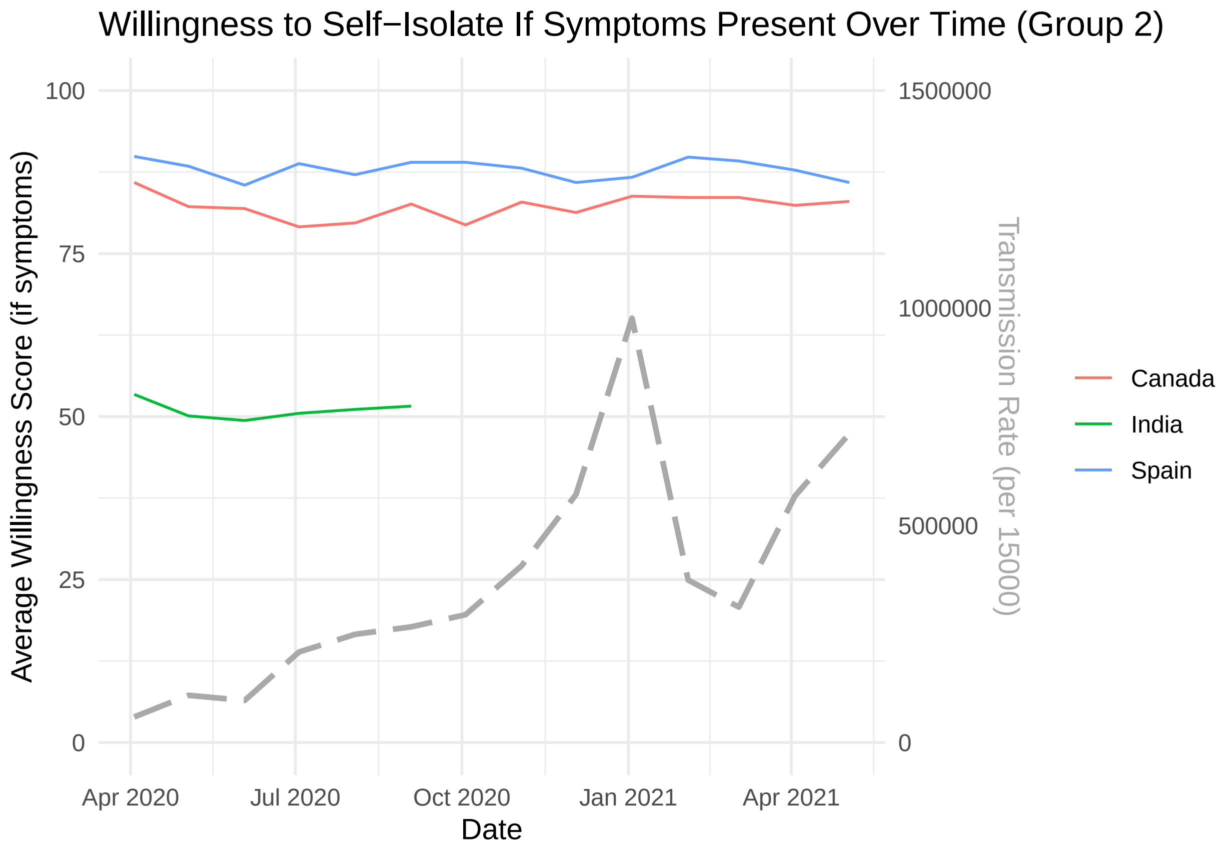
```
# Second plot: Willingness to self-isolate if advised
ggplot(group1_data_combined) +
  geom_line(aes(x = Actual_Date, y = Scores.Isolate.Willingness_if_advised, color = Country)) +
  geom_line(aes(x = Actual_Date, y = Average_Transmission.x/15000),
            linetype = "longdash", color = "darkgrey", size = 1) +
  scale_y_continuous(
    name = "Average Willingness Score (if advised)",
    limits = c(0, 100),
    sec.axis = sec_axis(~ . * 15000, name = "Transmission Rate (per 15000)")
  ) +
  labs(
    title = "Willingness to Self-Isolate If Advised Over Time (Group 1)",
    x = "Date"
  ) +
  theme_minimal() +
  theme(
    legend.title = element_blank(),
    axis.title.y.right = element_text(color = "darkgrey")
  )
```



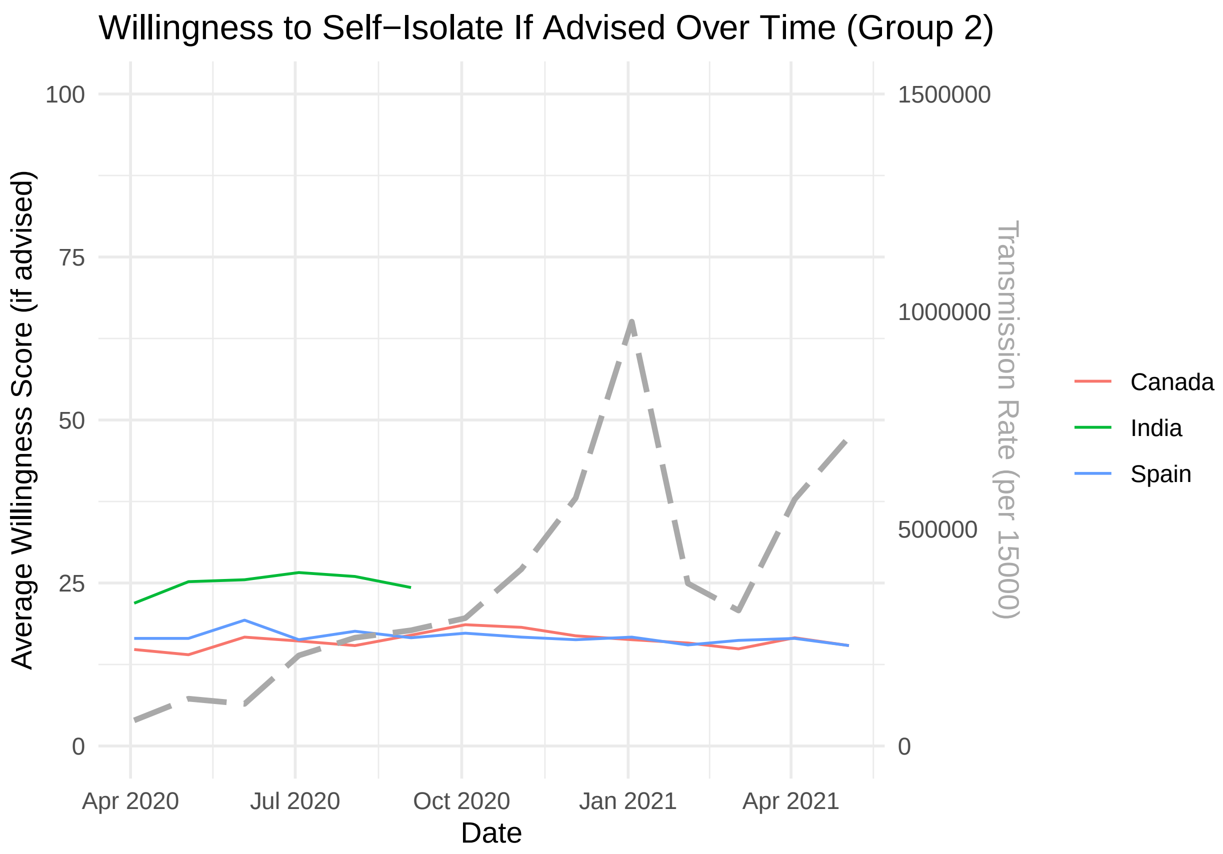
#

———— # Country Cluster 2 # ————

```
# Convert the 'Days_since_outbreak' to actual dates based on the group's start date
group2_start_date <- as.Date("2020-01-10")  # Assuming this is the correct start date for Group 1
bd$Actual_Date <- group2_start_date + bd$Days_since_outbreak - 1  # Correcting for the start date

group2_isolation_data <- bd %>%
  filter(Country %in% c("Canada", "Spain", "India"),
         Days_since_outbreak %in% c(85, 115, 146, 176, 207, 238, 268, 299, 329,
                                    360, 391, 419, 450, 480)) %>%
  select(Country, Actual_Date, Scores.Isolate.Willingness_if_symptoms,
         Scores.Isolate.Willingness_if_advised)

group2_data_combined <- group2_isolation_data %>%
  left_join(average_global_transmission, by = c("Actual_Date" = "date"))

# First plot: Willingness to self-isolate if symptoms are present
ggplot(group2_data_combined) +
  geom_line(aes(x = Actual_Date, y = Scores.Isolate.Willingness_if_symptoms, color = Country)) +
  geom_line(aes(x = Actual_Date, y = Average_Transmission/15000),
            linetype = "longdash", color = "darkgrey", size = 1) +
  scale_y_continuous(
    name = "Average Willingness Score (if symptoms)",
    limits = c(0, 100),
    sec.axis = sec_axis(~ . * 15000, name = "Transmission Rate (per 15000)")
  ) +
  labs(
    title = "Willingness to Self-Isolate If Symptoms Present Over Time (Group 2)",
    x = "Date"
  ) +
  theme_minimal() +
  theme(
    legend.title = element_blank(),
    axis.title.y.right = element_text(color = "darkgrey")
  )
```



```
# Second plot: Willingness to self-isolate if advised
ggplot(group2_data_combined) +
  geom_line(aes(x = Actual_Date, y = Scores.Isolate.Willingness_if_advised, color = Country)) +
  geom_line(aes(x = Actual_Date, y = Average_Transmission/15000),
            linetype = "longdash", color = "darkgrey", size = 1) +
  scale_y_continuous(
    name = "Average Willingness Score (if advised)",
    limits = c(0, 100),
    sec.axis = sec_axis(~ . * 15000, name = "Transmission Rate (per 15000)")
  ) +
  labs(
    title = "Willingness to Self-Isolate If Advised Over Time (Group 2)",
    x = "Date"
  ) +
  theme_minimal() +
  theme(
    legend.title = element_blank(),
    axis.title.y.right = element_text(color = "darkgrey")
  )
```

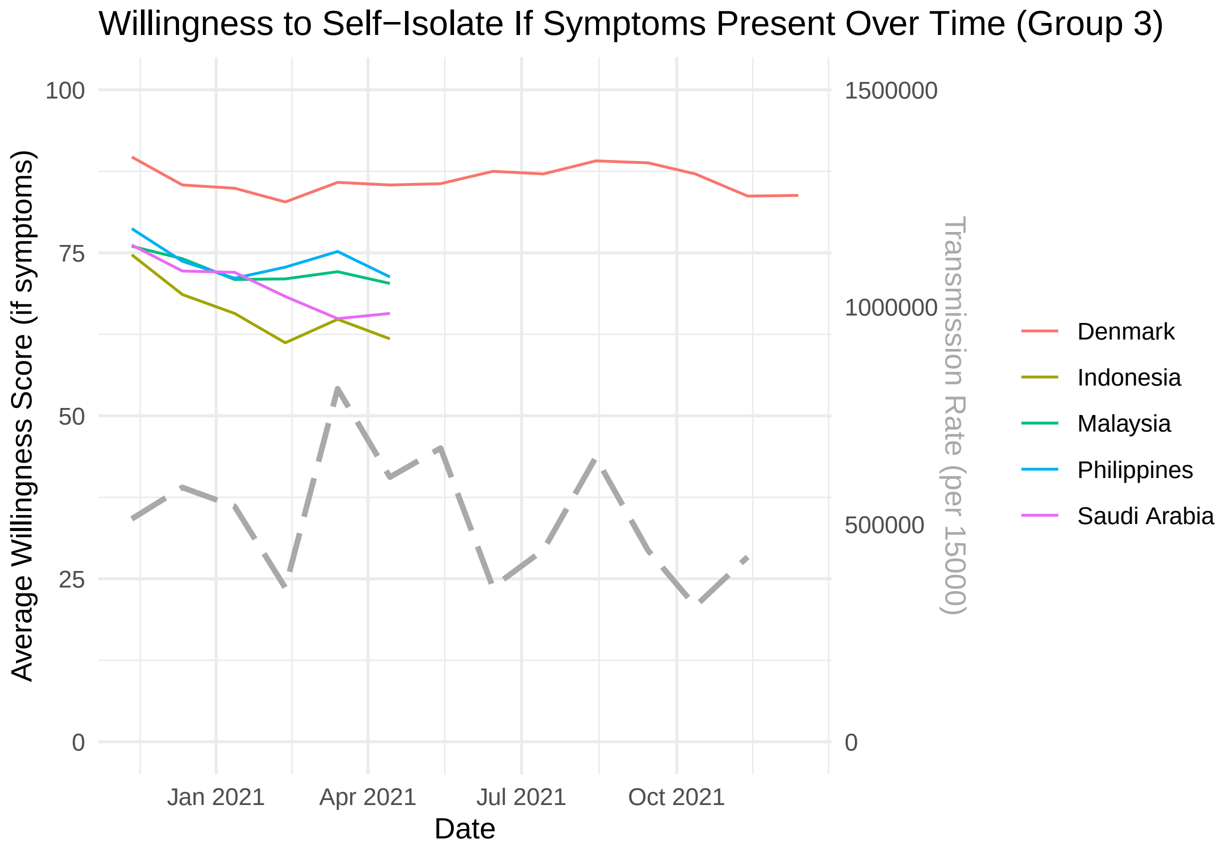


#

———— # Country Cluster 3 # ————

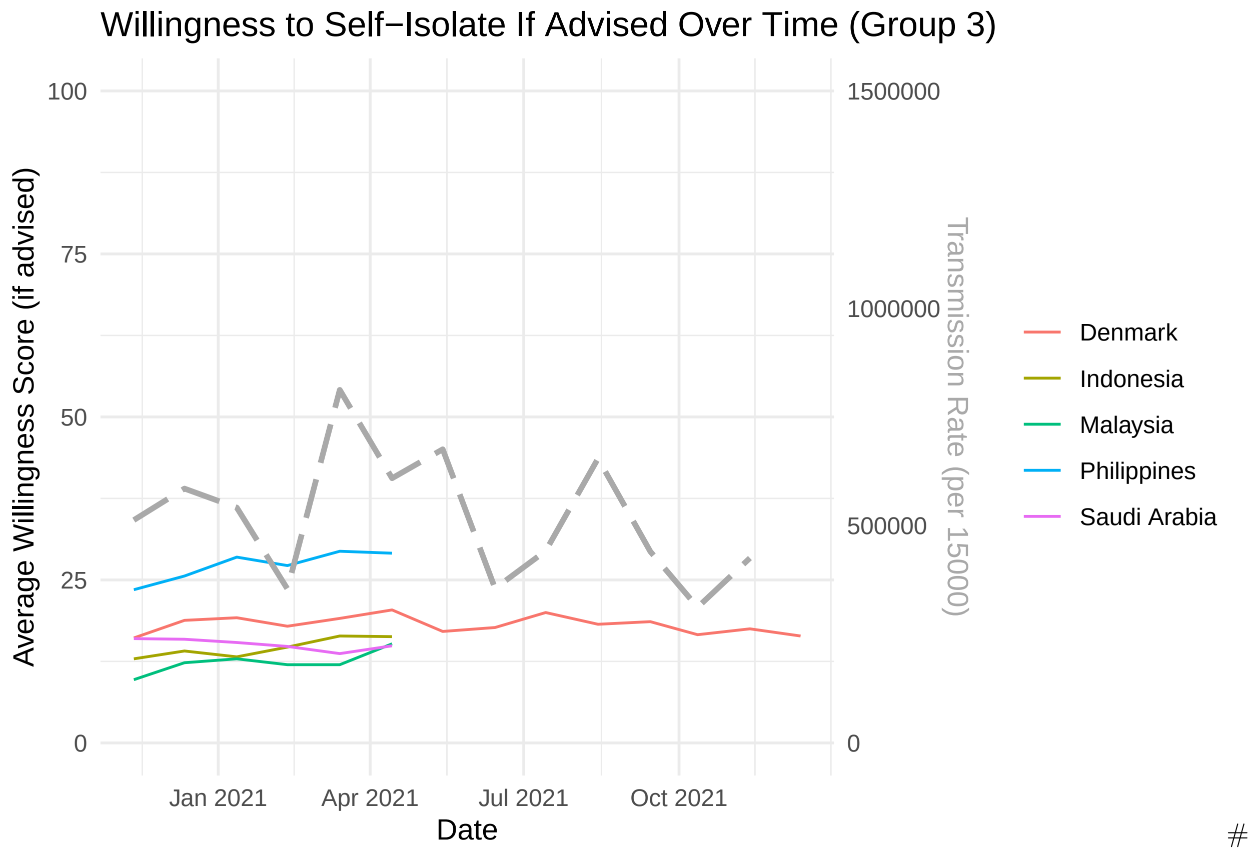
```
# Convert the 'Days_since_outbreak' to actual dates based on the group's start date
group3_start_date <- as.Date("2020-08-20")  # Assuming this is the correct start date for Group 1
bd$Actual_Date <- group3_start_date + bd$Days_since_outbreak - 1  # Correcting for the start date

group3_isolation_data <- bd %>%
  filter(Country %in% c("Indonesia", "Saudi Arabia", "Philippines", "Denmark", "Malaysia"),
         Days_since_outbreak %in% c(85, 115, 146, 176, 207, 238, 268, 299, 329,
                                    360, 391, 419, 450, 480)) %>%
  select(Country, Actual_Date, Scores.Isolate.Willingness_if_symptoms,
         Scores.Isolate.Willingness_if_advised)

group3_data_combined <- group3_isolation_data %>%
  left_join(average_global_transmission, by = c("Actual_Date" = "date"))

# First plot: Willingness to self-isolate if symptoms are present
ggplot(group3_data_combined) +
  geom_line(aes(x = Actual_Date, y = Scores.Isolate.Willingness_if_symptoms, color = Country)) +
  geom_line(aes(x = Actual_Date, y = Average_Transmission/15000),
            linetype = "longdash", color = "darkgrey", size = 1) +
  scale_y_continuous(
    name = "Average Willingness Score (if symptoms)",
    limits = c(0, 100),
    sec.axis = sec_axis(~ . * 15000, name = "Transmission Rate (per 15000)")
  ) +
  labs(
    title = "Willingness to Self-Isolate If Symptoms Present Over Time (Group 3)",
    x = "Date"
  ) +
  theme_minimal() +
  theme(
    legend.title = element_blank(),
    axis.title.y.right = element_text(color = "darkgrey")
  )
```

```
Warning: Removed 1 row containing missing values (`geom_line()`).
```



```
# Second plot: Willingness to self-isolate if advised
ggplot(group3_data_combined) +
  geom_line(aes(x = Actual_Date, y = Scores.Isolate.Willingness_if_advised, color = Country)) +
  geom_line(aes(x = Actual_Date, y = Average_Transmission/15000),
            linetype = "longdash", color = "darkgrey", size = 1) +
  scale_y_continuous(
    name = "Average Willingness Score (if advised)",
    limits = c(0, 100),
    sec.axis = sec_axis(~ . * 15000, name = "Transmission Rate (per 15000)")
  ) +
  labs(
    title = "Willingness to Self-Isolate If Advised Over Time (Group 3)",
    x = "Date"
  ) +
  theme_minimal() +
  theme(
    legend.title = element_blank(),
    axis.title.y.right = element_text(color = "darkgrey")
  )
```

```
Warning: Removed 1 row containing missing values (`geom_line()`).
```



#

# Q2 - Masking Compliance #

```
# Renaming columns for consistency and to facilitate analysis
# The original dataset contains spaces in the column names which can cause issues during analysis
bd <- bd %>%
  rename(Scores.Masks.Outside_home = `Scores.Masks.Outside home`) %>%
  rename(Scores.Masks.Grocery_store = `Scores.Masks.Grocery store`) %>%
  rename(Scores.Masks.Clothing_store = `Scores.Masks.Clothing store`) %>%
  rename(Scores.Masks.Public_transport = `Scores.Masks.Public transport`)
```

```
# Defining a function to convert raw scores to a Likert scale
# This function will be applied to mask-wearing scores to normalize them on a scale of 1 to 5
convert_to_likert <- function(score, min_score, max_score, likert_min, likert_max) {
  # Perform a linear transformation to scale the raw score to the Likert scale
  likert_score <- likert_min + (score - min_score) * (likert_max - likert_min) / (max_score - min_score
  return(round(likert_score))
}
```

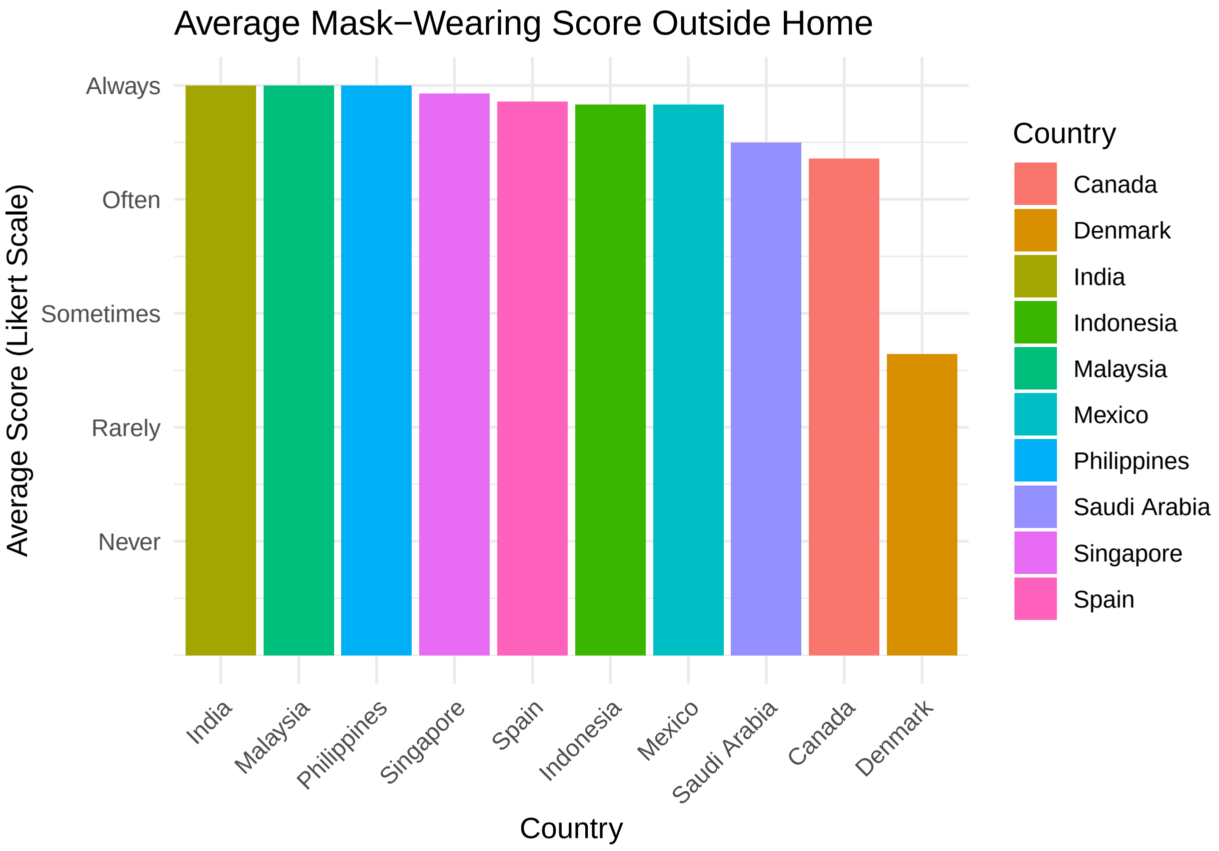
```
# Filtering the dataset for the ten countries selected for analysis
selected_countries <- c("Canada", "Spain", "Mexico", "Singapore", "India", "Indonesia",
                        "Saudi Arabia", "Philippines", "Denmark", "Malaysia")
bd_selected <- bd %>%
  filter(Country %in% selected_countries)
```

```r
# Applying the Likert scale conversion to the mask-wearing scores
# The mutate function adds new columns to the dataframe with the transformed Likert scale values
bd_selected <- bd_selected %>%
  mutate(
    Likert_Masks_Outside_Home = convert_to_likert(Scores.Masks.Outside_home, 0, 100, 1, 5),
    Likert_Masks_Grocery_Store = convert_to_likert(Scores.Masks.Grocery_store, 0, 100, 1, 5),
    Likert_Masks_Clothing_Store = convert_to_likert(Scores.Masks.Clothing_store, 0, 100, 1, 5),
    Likert_Masks_Work = convert_to_likert(Scores.Masks.Work, 0, 100, 1, 5),
    Likert_Masks_Public_Transport = convert_to_likert(Scores.Masks.Public_transport, 0, 100, 1, 5)
  )
```

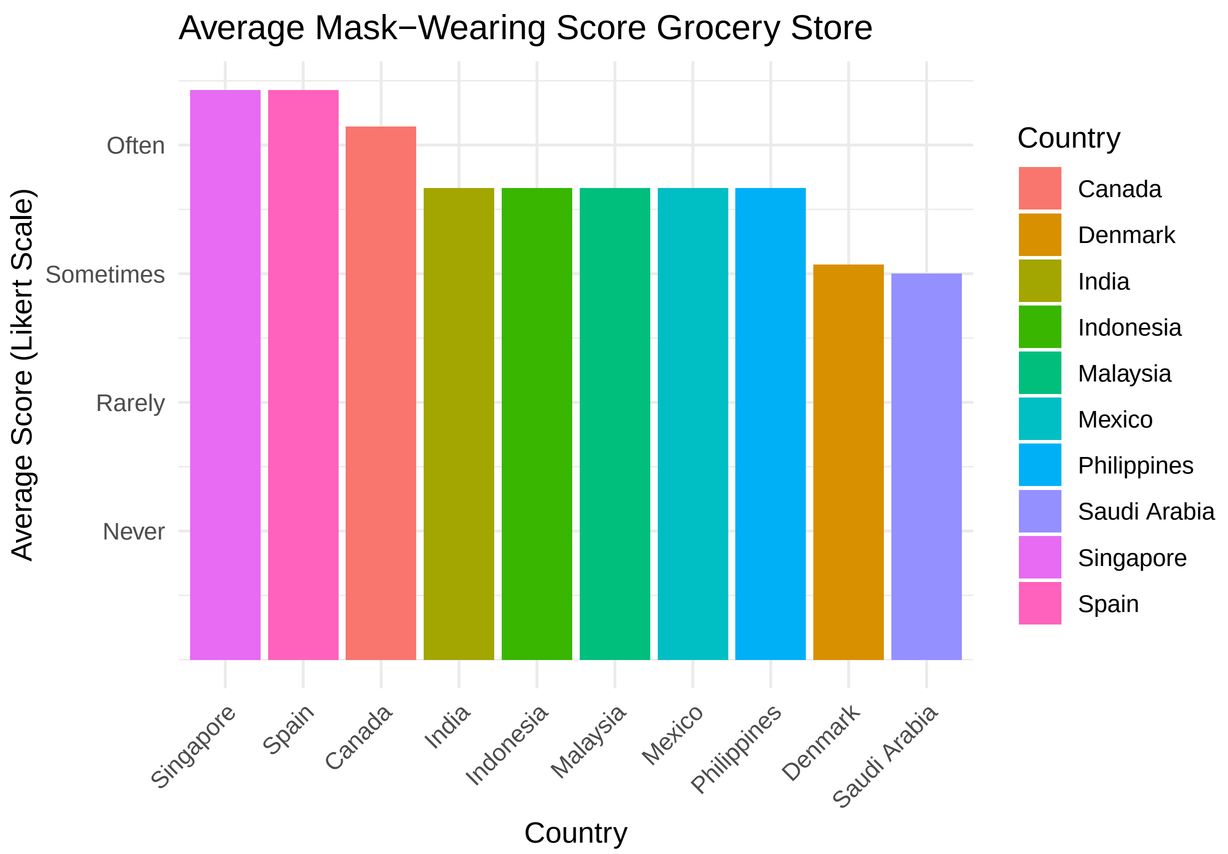
```r
# Calculating the average Likert scale scores for each mask-wearing context by country
# This summary will be used to compare countries based on their mask-wearing compliance
mask_wearing_scores <- bd_selected %>%
  group_by(Country) %>%
  summarise(
    Average_Outside_Home = mean(Likert_Masks_Outside_Home, na.rm = TRUE),
    Average_Grocery_Store = mean(Likert_Masks_Grocery_Store, na.rm = TRUE),
    Average_Clothing_Store = mean(Likert_Masks_Clothing_Store, na.rm = TRUE),
    Average_Work = mean(Likert_Masks_Work, na.rm = TRUE),
    Average_Public_Transport = mean(Likert_Masks_Public_Transport, na.rm = TRUE)
  )
```

```r
# Vector of labels for the Likert scale to be used on the y-axis of the bar charts
likert_labels <- c("Never", "Rarely", "Sometimes", "Often", "Always")
```

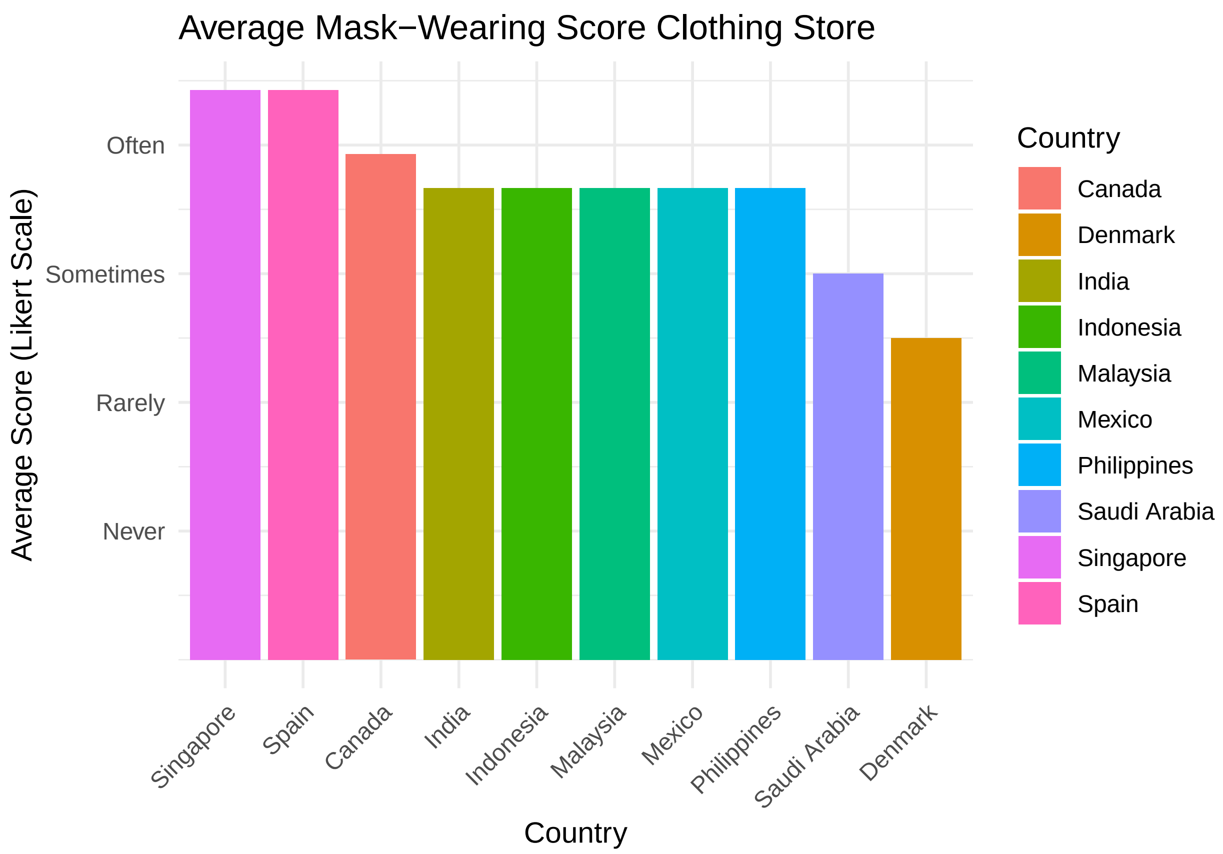
```r
# Creating a series of bar charts for each mask-wearing context
# Each plot shows the average Likert scale score for mask-wearing in different contexts by country
# The reorder function arranges the countries on the x-axis based on their average score
# This allows for easier visual comparison of mask-wearing compliance across countries

# Plot the average mask-wearing score outside home with Likert scale labels
ggplot(mask_wearing_scores, aes(x = reorder(Country, -Average_Outside_Home),
                                y = Average_Outside_Home, fill = Country)) +
  geom_bar(stat = "identity") +
  scale_y_continuous(breaks = 1:5, labels = likert_labels) + # Apply Likert scale labels
  labs(title = "Average Mask-Wearing Score Outside Home",
       x = "Country",
       y = "Average Score (Likert Scale)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



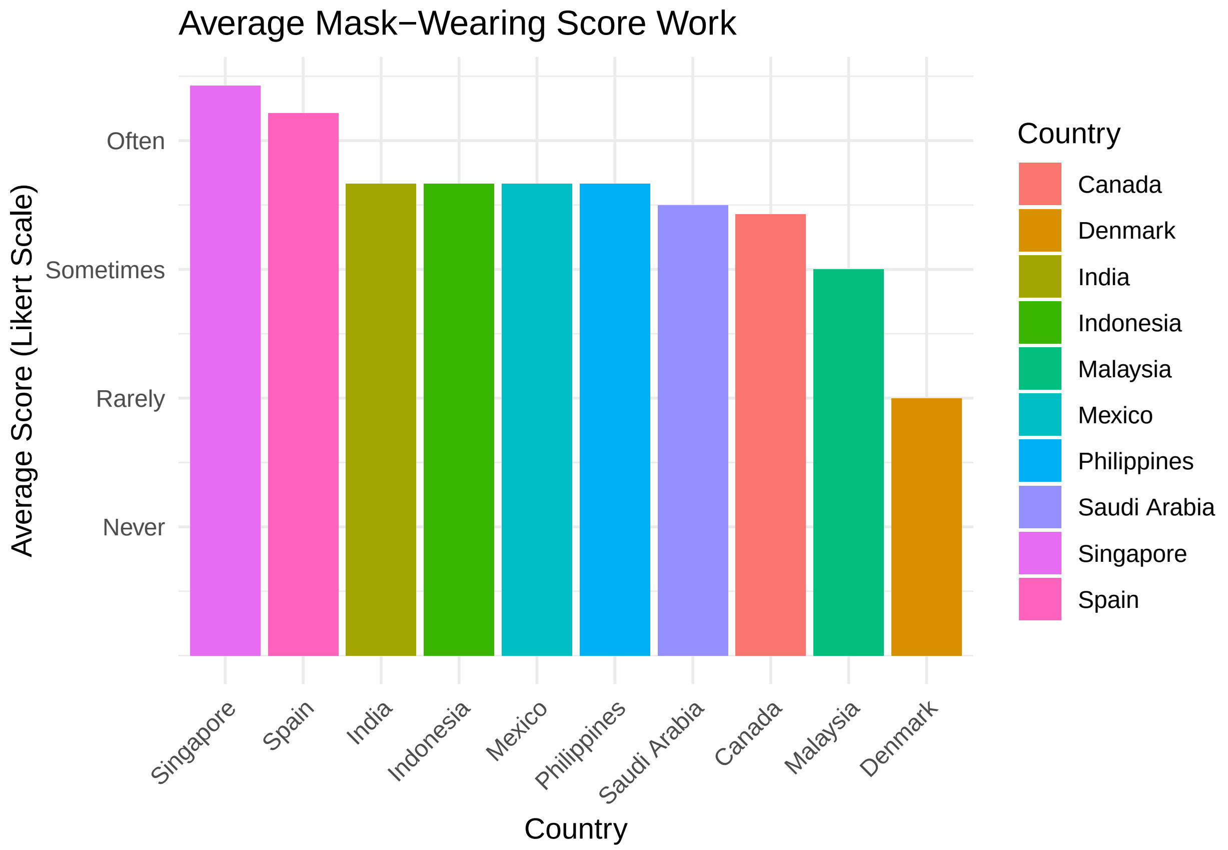
```r
# Plot the average mask-wearing score outside home with Likert scale labels
ggplot(mask_wearing_scores, aes(x = reorder(Country, -Average_Grocery_Store),
                                y = Average_Grocery_Store, fill = Country)) +
  geom_bar(stat = "identity") +
  scale_y_continuous(breaks = 1:5, labels = likert_labels) + # Apply Likert scale labels
  labs(title = "Average Mask-Wearing Score Grocery Store",
       x = "Country",
       y = "Average Score (Likert Scale)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



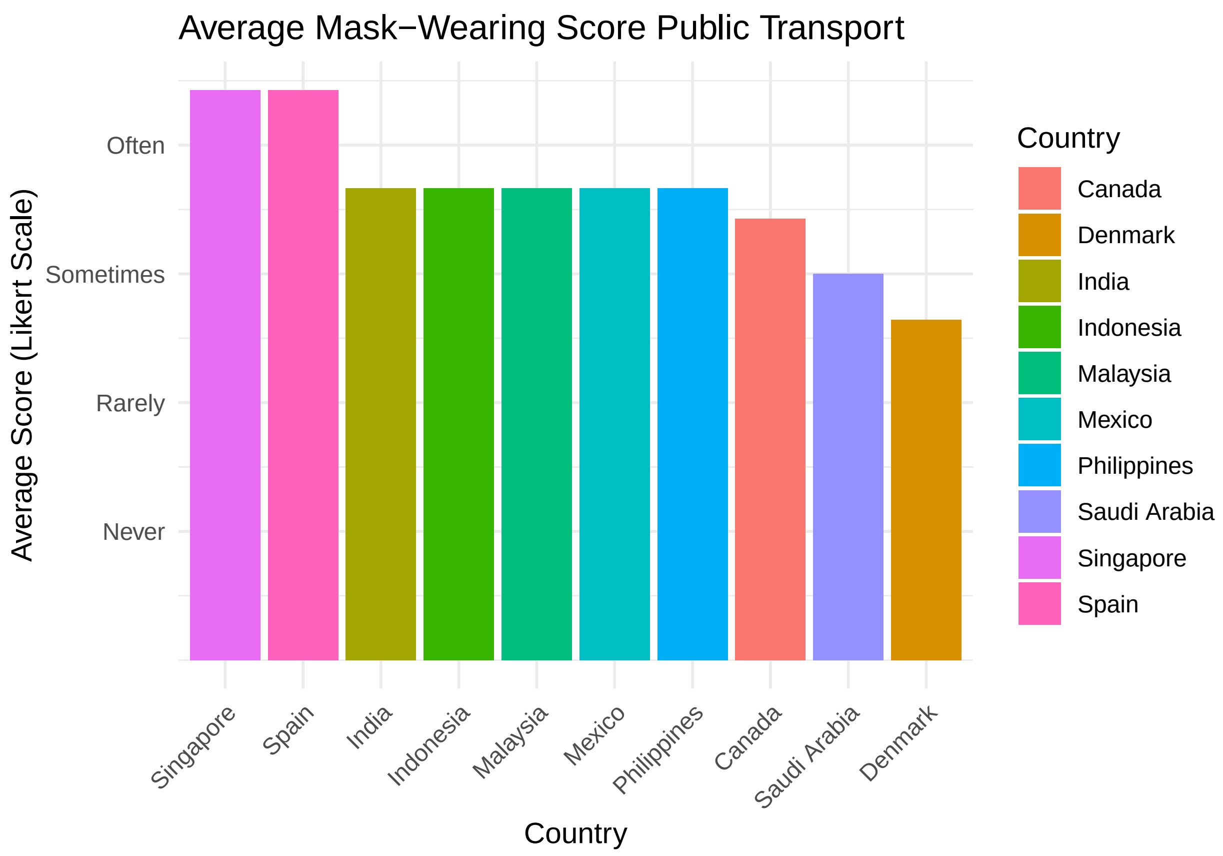
```
# Plot the average mask-wearing score outside home with Likert scale labels
ggplot(mask_wearing_scores, aes(x = reorder(Country, -Average_Clothing_Store),
                                y = Average_Clothing_Store, fill = Country)) +
  geom_bar(stat = "identity") +
  scale_y_continuous(breaks = 1:5, labels = likert_labels) + # Apply Likert scale labels
  labs(title = "Average Mask-Wearing Score Clothing Store",
       x = "Country",
       y = "Average Score (Likert Scale)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# Plot the average mask-wearing score outside home with Likert scale labels
ggplot(mask_wearing_scores, aes(x = reorder(Country, -Average_Work),
                                y = Average_Work, fill = Country)) +
  geom_bar(stat = "identity") +
  scale_y_continuous(breaks = 1:5, labels = likert_labels) + # Apply Likert scale labels
  labs(title = "Average Mask-Wearing Score Work",
       x = "Country",
       y = "Average Score (Likert Scale)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```r
# Plot the average mask-wearing score outside home with Likert scale labels
ggplot(mask_wearing_scores, aes(x = reorder(Country, -Average_Public_Transport),
                                y = Average_Public_Transport, fill = Country)) +
  geom_bar(stat = "identity") +
  scale_y_continuous(breaks = 1:5, labels = likert_labels) + # Apply Likert scale labels
  labs(title = "Average Mask-Wearing Score Public Transport",
       x = "Country",
       y = "Average Score (Likert Scale)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# These plots will be used to visually assess which countries had higher levels of compliance
# with mask-wearing guidelines throughout the pandemic. The use of Likert scales facilitates
# understanding the level of compliance in a standardized format.
```