```
> library(ggplot2)
> library(dplyr)
> library(tidyr)
> library(stringr)
> library(forcats)
> library(lubridate)
> library(purrr)
> library(tibble)
> library(tidyverse)
> library(ggthemes)
> library(ggrepel)
> library(ggpubr)
> library(cowplot)
> library(scales)
> library(viridis)
> library(RColorBrewer)
> library(gridExtra)
> library(knitr)
> library(kableExtra)
> library(DT)
> library(plotly)
> library(ggmap)
> library(leaflet)
> library(shiny)
> library(shinydashboard)
> library(shinyWidgets)
> library(tmap)
> library(tmaptools)
> library(sf)
> library(rgdal)
> library(rgeos)
> library(maptools)
> library(spatstat)
> library(spdep)
> library(GWmodel)
> library(SpatialKDE)
> library(geoR)
> library(geosphere)
> library(rnaturalearth)
> library(rnaturalearthdata)
> library(rgeoboundaries)
> library(RgoogleMaps)
> library(ggsn)
> library(kriming)
> library(ggspatial)
> library(leaflet.extras)
> library(OpenStreetMap)
> library(spatialreg)
> library(mapproj)
> library(maps)
> library(ape)
> library(RCurl)
> library(jsonlite)
> library(purrrlyr)
> library(reshape2)
> library(magrittr)
> library(plyr)
> library(Matrix)
> library(foreach)
> library(glmnet)
> library(tidytext)
> library(gridtext)
> library(MASS)
> library(randomForest)
> library(stringi)
> library(xgboost)
> library(ranger)
> library(e1071)
> library(caret)
> library(SuperLearner)
> library(tsne)
> library(tseries)
> library(forecast)
> library(stringdist)
> library(LCMCR)
> library(recipes)
> library(prettyunits)
> library(hexbin)
> library(Matrix.utils)
> library(gganimate)
> library(latticeExtra)
> library(ggforce)
> library(GGally)
> library(ggridges)
> library(ggthemes)
> library(viridis)
> library(hrbrthemes)
> library(fivethirtyeight)
> library(tufte)
> library(plotdap)
> library(STRAMPR)
> library(dutchmasters)
> library(prettydoc)
> library(vembedr)
> library(urbanmapr)
> library(DataExplorer)
> library(greekmodels)
> library(scico)
> library(ggsci)
> library(ggtext)
> library(tidyverseRecipes)
> library(caretEnsemble)
> library(topicmodels)
> library(LDAvis)
> library(ldatuning)
> library(D3TableFilter)
> library(newsanchor)
> library(DT)
> library(diffobj)
> library(revtools)
> library(coronavirus)
> library(usethis)
```

Thus,

```
> devtools::install_github("RamiKrispin/coronavirus", force = TRUE)
```

and

```
> update_coronavirus()
```

will download the latest version of the coronavirus package. Let's load the package and look at the structure of the data:

```
> str(coronavirus)
'data.frame':	31338 obs. of  8 variables:
 $ Province.State: chr  NA NA NA NA NA NA NA NA NA NA ...
 $ Country.Region: chr  "Afghanistan" "Albania" "Algeria" "Andorra" ...
 $ Lat           : num  33.0 41.1 28.0 42.5 -11.2 17.1 -38.4 -34.6 40.2 -31.4 ...
 $ Long          : num  65.2 20.2 1.7 1.5 17.9 -61.8 -63.6 -58.4 44.5 -64.2 ...
 $ date          : Date, chr [1:31338] "2020-01-22" "2020-01-22" "2020-01-22" "2020-01-22" ...
 $ cases         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ type          : chr  "confirmed" "confirmed" "confirmed" "confirmed" ...
 $ uid           : int  4 8 12 20 24 28 32 36 51 52 ...
```

1. Dataset Analysis

```
> df <- load_csv("./data/coronavirus/")
Parsed with column specification:
cols(
  Province.State = col_character(),
  Country.Region = col_character(),
  Lat = col_double(),
  Long = col_double(),
  date = col_date(format = "%Y-%m-%d"),
  cases = col_double(),
  type = col_character(),
  uid = col_double(),
  iso3 = col_character(),
  code3 = col_double(),
  Combined_Key = col_character(),
  Population = col_double(),
  continent_name = col_character,
  continent_code = col_character,
  Date = col_date(format = "%Y-%m-%d"),
  Confirmed = col_double(),
  Deaths = col_double(),
  Recovered = col_double(),
  Active = col_double(),
  CFR = col_double(),
  incidence_rate = col_double(),
  case_fatality_ratio = col_double,
  Lat.1 = col_double(),
  Long_ = col_double(),
  Country_Region = col_character(),
  Province_State = col_character(),
  Cases = col_double(),
  Status = col_character(),
  Date.1 = col_date(format = "%Y-%m-%d"),
  Cases.1 = col_double(),
  Long.1 = col_double(),
  Lat.2 = col_double(),
  Country/Region = col_character(),
  Province/State = col_character(),
  Date.2 = col_date(format = "%Y-%m-%d"),
  Confirmed.1 = col_double(),
  Deaths.1 = col_double(),
  Recovered.1 = col_double(),
  Lat.3 = col_double(),
  Long.2 = col_double(),
  Country/Region.1 = col_character(),
  Province/State.1 = col_character(),
  Date.3 = col_date(format = "%Y-%m-%d"),
  Confirmed.2 = col_double(),
  Deaths.2 = col_double(),
  Recovered.2 = col_double(),
  Lat.4 = col_double(),
  Long.3 = col_double(),
  Country/Region.2 = col_character(),
  Province/State.2 = col_character(),
  Date.4 = col_date(format = "%Y-%m-%d"),
  Confirmed.3 = col_double(),
  Deaths.3 = col_double(),
  Recovered.3 = col_double,
  Lat.5 = col_double(),
  Long.4 = col_double(),
  Country/Region.3 = col_character(),
  Province/State.3 = col_character(),
  Date.5 = col_date(format = "%Y-%m-%d"),
  Confirmed.4 = col_double(),
  Deaths.4 = col_double(),
  Recovered.4 = col_double,
  Lat.6 = col_double(),
  Long.5 = col_double(),
  Country/Region.4 = col_character(),
  Province/State.4 = col_character(),
  Date.6 = col_date(format = "%Y-%m-%d"),
  Confirmed.5 = col_double(),
  Deaths.5 = col_double(),
  Recovered.5 = col_double,
```