COVID19data Vignette

Package - COVID19data

This R-package `COVID19data` contains 16 wrapper functions to get Covid-19 data of Canada. These functions process API https://opencovid.ca/api/#welcome to get data. Some functions provide daily data and others provide summary of data for all the provinces. This package contains four internal functions to request API which are not available to user. The list of 16 wrapper functions available to user is given below:

1. `active_cases(provinceName = 'Canada')`
2. `vaccine_as_per_province(provinceName = 'Canada')`
3. `cases_as_per_province(provinceName = 'Canada')`
4. `Yearlycases_as_per_province(provinceName = 'Canada',yearpassed='2020')`
5. `cumulativevaccine_as_per_province(provinceName = 'Canada')`
6. `vaccine_distribution(provinceName = 'Canada')`
7. `yearly_vaccine_distribution(provinceName = 'Canada',distyear='2020')`
8. `mortality(provinceName = 'Canada')`
9. `yearly_deaths(provinceName = 'Canada', dyear='2020')`
10. `recovered(provinceName = 'Canada')`
11. `yearly_recovered(provinceName = 'Canada', ryear='2020')`
12. `testing(provinceName = 'Canada')`
13. `summary_of_cases(provinceName = "Canada")`
14. `summary_of_peaktime(provinceName = "Canada", date = '01-09-2020')`
15. `summary_of_canada(date = '01-09-2020')`
16. `province_population(provinceName = 'Canada')`

The detailed description of these functions as given below.

To install package

First, user need to install `install.packages("devtools")` and then install the package from github using command `install_github('aishwaryasharma10/COVID19data', build_vignettes=TRUE)`. Make the optional argument `build_vignettes` true if you want to build vignettes for this package and use `browseVignettes("COVID19data")` to see the vignette . Then, the package can be loaded as below:

```
library(covid19data)
```

1. active_cases(provinceName = 'Canada')

This function `active_cases` returns a dataframe of daily active Covid cases of desired province or Canada. The function accepts the province names in its full form not abbreviated. If user does not pass any argument then Canada is used as an argument which returns the data of whole Canada as a whole. The returned data is a data frame and contains the columns including the date, province name, active cases, change in active cases, cumulative cases, cumulative deaths and cumulative recovered. The function can be called as below.

```
data <- active_cases(provinceName = 'British Columbia')
head(data)
#>         date         province active_cases active_cases_change cumulative_cases
#> 1 2020-01-25 British Columbia            0                   0                0
#> 2 2020-01-26 British Columbia            0                   0                0
#> 3 2020-01-27 British Columbia            0                   0                0
#> 4 2020-01-28 British Columbia            1                   1                1
#> 5 2020-01-29 British Columbia            1                   0                1
#> 6 2020-01-30 British Columbia            1                   0                1
#>   cumulative_deaths cumulative_recovered
#> 1                 0                    0
#> 2                 0                    0
#> 3                 0                    0
#> 4                 0                    0
#> 5                 0                    0
#> 6                 0                    0
```

2. vaccine__as__per__province(provinceName = ‘Canada’)

This function `vaccine_as_per_province` returns a dataframe of vaccines given as per the desired province or Canada. The function accepts the province names in its full form not abbreviated. If user does not pass any argument then Canada is used as an argument which returns the data of whole Canada as a whole. The returned data is a data frame and contains the columns including the date of the vaccination, province name, administered vaccine and cumulative vaccine. The function can be called as below.

```
vaccine_as_per_province(provinceName = 'British Columbia')
```

3. cases__as__per__province(provinceName = ‘Canada’)

This function `cases_as_per_province` returns a dataframe of Covid -19 cases as per the desired province or Canada. The function accepts the province names in its full form not abbreviated. If user does not pass any argument then Canada is used as an argument which returns the data of whole Canada as a whole. The returned data is a data frame and contains the columns including the date of the cases, province name, cases and cumulative cases. The function can be called as below.

```
cases_as_per_province(provinceName = 'British Columbia')
```

4. Yearlycases__as__per__province(provinceName = ‘Canada’,yearpassed=‘2020’)

This function `Yearlycases_as_per_province` returns a dataframe of Covid -19 cases as per the desired province or Canada. Also, it takes year as an argument along with the province name. It will return the Covid -19 cases as per the passed province and year. The function accepts the province names in its full form not abbreviated. If user does not pass any argument then for province it takes Canada and for year it takes 2020. This returns the data of full Canada as a whole for year 2020. The returned data is a data frame and contains the columns including the date of the cases, province name, cases and cumulative cases for the year specified. The function can be called as below.

```
Yearlycases_as_per_province(provinceName = 'British Columbia')
```

5. cumulativevaccine__as__per__province(provinceName = ‘Canada’)

This function `cumulativevaccine_as_per_province` returns a dataframe of Covid -19 cumulative vaccines as per the desired province or Canada. The function accepts the province names in its full form not abbreviated. If user does not pass any argument then Canada is used as an argument which returns the data of whole Canada as a whole. The returned data is a data frame and contains the columns including the date of the vaccination, province name,cumulative vaccine and cvaccine. The function can be called as below.

```
cumulativevaccine_as_per_province(provinceName = 'British Columbia')
```

6. vaccine__distribution(provinceName = ‘Canada’)

This function `vaccine_distribution` returns a dataframe of Covid -19 vaccine distribution as per the desired province or Canada. The function accepts the province names in its full form not abbreviated. If user does not pass any argument then Canada is used as an argument which returns the data of whole Canada as a whole. The returned data is a data frame and contains the columns including the date distribution, province name,cumulative vaccine and count of distribution of vaccines. The function can be called as below.

```
vaccine_distribution(provinceName = 'British Columbia')
```

7. yearly__vaccine__distribution(provinceName = ‘Canada’,distyear=‘2020’)

This function `yearly_vaccine_distribution` returns a dataframe of Covid -19 vaccine distribution as per the desired province or Canada. Also, it takes year as an argument along with the province name. It will return the Covid -19 cases as per the passed province and year. The function accepts the province names in its full form not abbreviated. If user does not pass any argument then for province it takes Canada and for year it takes 2020 as default. This returns the data of full Canada as a whole for year 2020. The returned data is a data frame and contains the columns including the date distribution, province name,cumulative vaccine and count of distribution of vaccines. The function can be called as below.

```
yearly_vaccine_distribution(provinceName = 'British Columbia', distyear = '2021')
```

8. mortality(provinceName = ‘Canada’)

This function `mortality` returns a dataframe of Covid -19 mortality rate as per the desired province or Canada. The function accepts the province names in its full form not abbreviated. If user does not pass any argument then Canada is used as an argument which returns the data of whole Canada as a whole. The returned data is a data frame and contains the columns including the date of the death, province name, cumulative deaths and deaths. The function can be called as below.

```
mortality(provinceName = 'British Columbia')
```

9. yearly__deaths(provinceName = ‘Canada’, dyear=‘2020’)

This function `yearly_deaths` returns a dataframe of number of deaths due to Covid -19 as per the desired province or Canada. Also, it takes year as an argument along with the province name. It will return the Covid -19 deaths as per the passed province and year. The function accepts the province names in its full form not abbreviated. If user does not pass any argument then for province it takes Canada and for year it takes 2020 as default. This returns the data of full Canada as a whole for year 2020. The returned data is a data frame and contains the columns including the date of the death, province name, cumulative deaths and deaths. The function can be called as below.

```
yearly_deaths(provinceName = 'British Columbia')
```

10. recovered(provinceName = ‘Canada’)

This function `recovered` returns a dataframe of Covid -19 recovery rates as per the desired province or Canada. The function accepts the province names in its full form not abbreviated. If user does not pass any argument then Canada is used as an argument which returns the data of whole Canada as a whole. The returned data is a data frame and contains the columns including the recovery date, province, cumulative recovered, and count of recovery. The function can be called as below.

```
recovered(provinceName = 'British Columbia')
```

11. yearly__recovered(provinceName = ‘Canada’, ryear=‘2020’)

This function `yearly_recovered` returns a dataframe of number of patients who have recovered after Covid -19 as per the desired province or Canada. Also, it takes year as an argument along with the province name.

It will return the Covid -19 recovery count as per the passed province and year. The function accepts the province names in its full form not abbreviated. If user does not pass any argument then for province it takes Canada and for year it takes 2020 as default. This returns the data of full Canada as a whole for year 2020. The returned data is a data frame and contains the columns including the recovery date, province, cumulative recovered, and count of recovery. The function can be called as below.

```
yearly_recovered(provinceName = 'British Columbia', ryear='2020')
```

12. testing(provinceName = 'Canada')

This function `testing` returns a dataframe of Covid -19 testing rates as per the desired province or Canada. The function accepts the province names in its full form not abbreviated. If user does not pass any argument then Canada is used as an argument which returns the data of whole Canada as a whole. The returned data is a data frame and contains the columns including the testing date, province, cumulative testing, and count of testing. The function can be called as below.

```
testing(provinceName = 'British Columbia')
```

13. summary_of_cases(provinceName = "Canada")

The function takes in the input string formatted which would either be the name of a particular province for which the user needs the Covid summary data or the country name – 'Canada'. The function after parsing the input from the user will return a data frame relating to the summary for the particular date, province name, active_cases, active_cases_change, avaccine, cases,cumulative_avaccine, cumulative_cases,cumulative_cvaccine, cumulative_deaths, cumulative_dvaccine, cumulative_recovered, cumulative_testing, cvaccine, deaths, dvaccine, recovered, testing and testing_info. The function can be called as below.

```
summary_of_cases(provinceName = 'British Columbia')
```

14. summary_of_peaktime(provinceName = "Canada", date = '01-09-2020')

The function returns the summary for the peak time for Canada which was September 2020. The input would be string either be the province or the country name – 'Canada'. The function returns a dataframe of the date, province name, active_cases, active_cases_change, avaccine, cases,cumulative_avaccine, cumulative_cases,cumulative_cvaccine, cumulative_deaths, cumulative_dvaccine, cumulative_recovered, cumulative_testing, cvaccine, deaths, dvaccine, recovered, testing and testing_info. The function can be called as below.

```
summary_of_peaktime(provinceName = 'British Columbia')
```

15. summary_of_canada(date = '01-09-2020')

The function returns the summary for the peak time for Canada which was September 2020. The input would be string the country name – 'Canada'. The function returns a dataframe of the date, province name, active_cases, active_cases_change, avaccine, cases,cumulative_avaccine, cumulative_cases,cumulative_cvaccine, cumulative_deaths, cumulative_dvaccine, cumulative_recovered, cumulative_testing, cvaccine, deaths, dvaccine, recovered, testing and testing_info. The function can be called as below.

```
summary_of_canada()
```

16. province_population(provinceName = 'Canada')

The function returns the total population count for every province individually. The input would be string either be the province or the country name – 'Canada'. The function returns a data frame which includes the province, and its population. The function can be called as below.

```r
data <- province_population()
head(data)
#>                    province     pop
#> 1                   Alberta 4371316
#> 2          British Columbia 5071336
#> 3                  Manitoba 1369465
#> 4             New Brunswick  776827
#> 5 Newfoundland and Labrador  521542
#> 6     Northwest Territories   44826
```