Introduction to Data Scientist 11372 (UG)

Final Assessment Part D – Documentation and Reporting

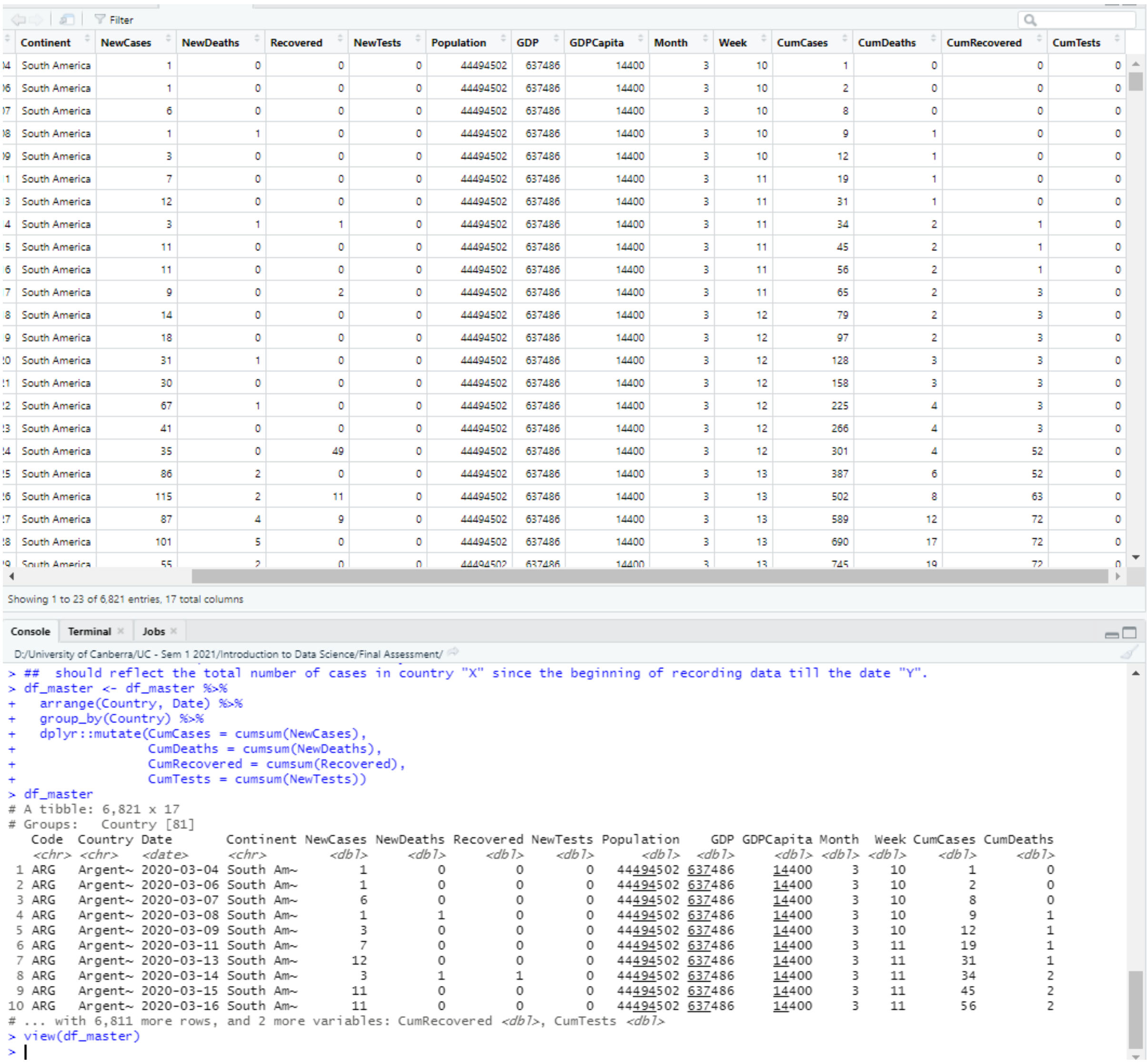
Tuan Anh (Vincent) Nguyen – u3196825

Table of Contents

Task 2 Exploratory Data Analysis

Q1:

```
> # PART 2: Exploratory Data Analysis
> ##1.Add 4 variables ("CumCases", "CumDeaths", "CumRecovered", "CumTests")
> ##  These variables should reflect the cumulative relevant data up to the date
> ##  of the observation, i.e. CumCases for country "X" at Date "Y"
> ##  should reflect the total number of cases in country "X" since the beginning of recording data till the date "Y".
> df_master <- df_master %>%
+   arrange(Country, Date) %>%
+   group_by(Country) %>%
+   dplyr::mutate(CumCases = cumsum(NewCases),
+                 CumDeaths = cumsum(NewDeaths),
+                 CumRecovered = cumsum(Recovered),
+                 CumTests = cumsum(NewTests))
>
```



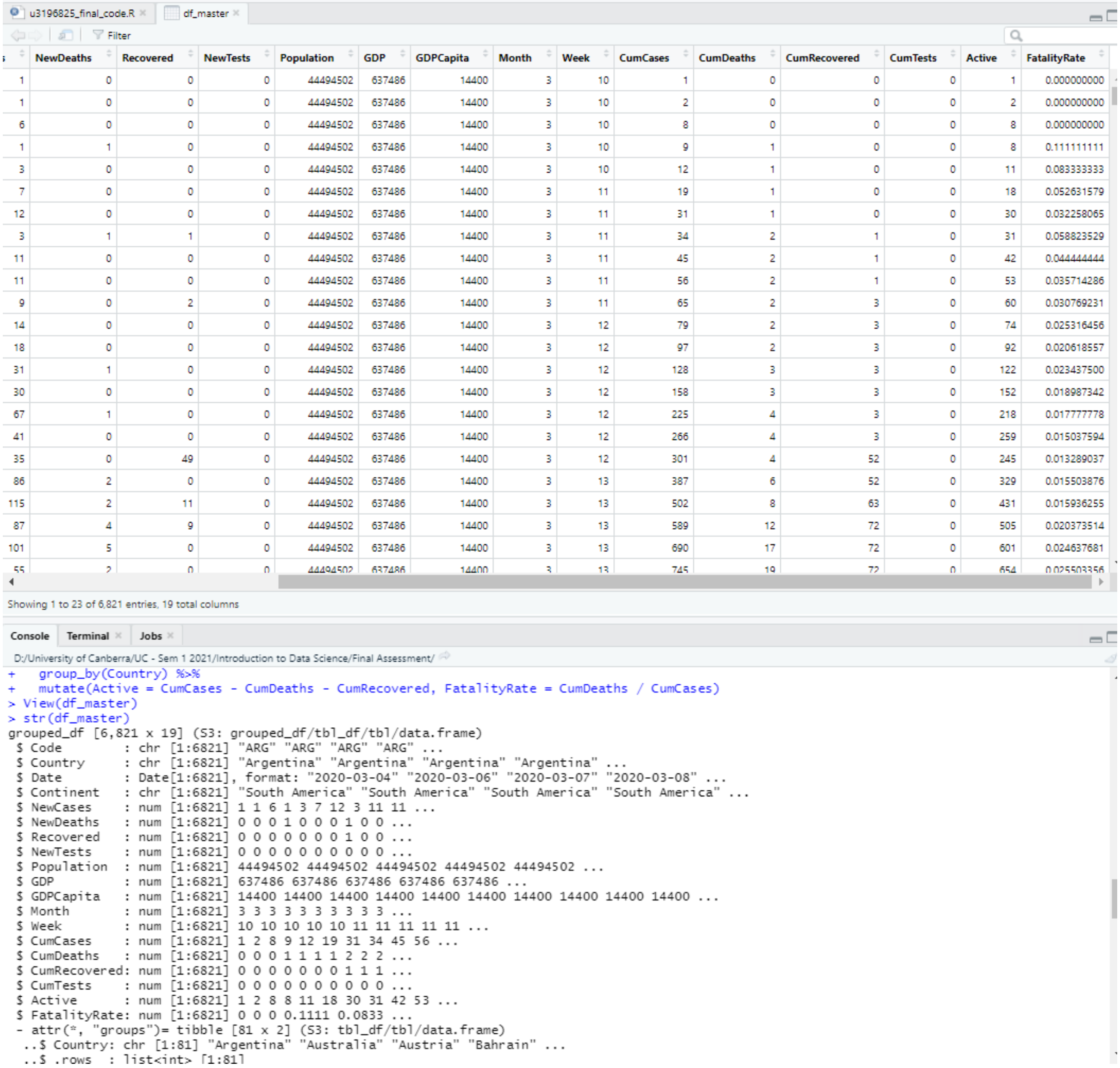
	Continent	NewCases	NewDeaths	Recovered	NewTests	Population	GDP	GDPCapita	Month	Week	CumCases	CumDeaths	CumRecovered	CumTests
4	South America	1	0	0	0	44494502	637486	14400	3	10	1	0	0	0
6	South America	1	0	0	0	44494502	637486	14400	3	10	2	0	0	0
7	South America	6	0	0	0	44494502	637486	14400	3	10	8	0	0	0
8	South America	1	1	0	0	44494502	637486	14400	3	10	9	1	0	0
9	South America	3	0	0	0	44494502	637486	14400	3	10	12	1	0	0
1	South America	7	0	0	0	44494502	637486	14400	3	11	19	1	0	0
3	South America	12	0	0	0	44494502	637486	14400	3	11	31	1	0	0
4	South America	3	1	1	0	44494502	637486	14400	3	11	34	2	1	0
5	South America	11	0	0	0	44494502	637486	14400	3	11	45	2	1	0
6	South America	11	0	0	0	44494502	637486	14400	3	11	56	2	1	0
7	South America	9	0	2	0	44494502	637486	14400	3	11	65	2	3	0
8	South America	14	0	0	0	44494502	637486	14400	3	12	79	2	3	0
9	South America	18	0	0	0	44494502	637486	14400	3	12	97	2	3	0
0	South America	31	1	0	0	44494502	637486	14400	3	12	128	3	3	0
1	South America	30	0	0	0	44494502	637486	14400	3	12	158	3	3	0
2	South America	67	1	0	0	44494502	637486	14400	3	12	225	4	3	0
3	South America	41	0	0	0	44494502	637486	14400	3	12	266	4	3	0
4	South America	35	0	49	0	44494502	637486	14400	3	12	301	4	52	0
5	South America	86	2	0	0	44494502	637486	14400	3	13	387	6	52	0
6	South America	115	2	11	0	44494502	637486	14400	3	13	502	8	63	0
7	South America	87	4	9	0	44494502	637486	14400	3	13	589	12	72	0
8	South America	101	5	0	0	44494502	637486	14400	3	13	690	17	72	0
9	South America	55	2	0	0	44494502	637486	14400	3	13	745	19	72	0

Showing 1 to 23 of 6,821 entries, 17 total columns

D:/University of Canberra/UC - Sem 1 2021/Introduction to Data Science/Final Assessment/

```
> ##  should reflect the total number of cases in country "X" since the beginning of recording data till the date "Y".
> df_master <- df_master %>%
+   arrange(Country, Date) %>%
+   group_by(Country) %>%
+   dplyr::mutate(CumCases = cumsum(NewCases),
+                 CumDeaths = cumsum(NewDeaths),
+                 CumRecovered = cumsum(Recovered),
+                 CumTests = cumsum(NewTests))
> df_master
# A tibble: 6,821 x 17
# Groups:   Country [81]
   Code  Country Date       Continent NewCases NewDeaths Recovered NewTests Population    GDP GDPCapita Month  Week CumCases CumDeaths
   <chr> <chr>   <date>     <chr>        <dbl>     <dbl>     <dbl>    <dbl>      <dbl>  <dbl>     <dbl> <dbl> <dbl>    <dbl>     <dbl>
 1 ARG   Argent~ 2020-03-04 South Am~        1         0         0        0   44494502 637486     14400     3    10        1         0
 2 ARG   Argent~ 2020-03-06 South Am~        1         0         0        0   44494502 637486     14400     3    10        2         0
 3 ARG   Argent~ 2020-03-07 South Am~        6         0         0        0   44494502 637486     14400     3    10        8         0
 4 ARG   Argent~ 2020-03-08 South Am~        1         1         0        0   44494502 637486     14400     3    10        9         1
 5 ARG   Argent~ 2020-03-09 South Am~        3         0         0        0   44494502 637486     14400     3    10       12         1
 6 ARG   Argent~ 2020-03-11 South Am~        7         0         0        0   44494502 637486     14400     3    11       19         1
 7 ARG   Argent~ 2020-03-13 South Am~       12         0         0        0   44494502 637486     14400     3    11       31         1
 8 ARG   Argent~ 2020-03-14 South Am~        3         1         1        0   44494502 637486     14400     3    11       34         2
 9 ARG   Argent~ 2020-03-15 South Am~       11         0         0        0   44494502 637486     14400     3    11       45         2
10 ARG   Argent~ 2020-03-16 South Am~       11         0         0        0   44494502 637486     14400     3    11       56         2
# ... with 6,811 more rows, and 2 more variables: CumRecovered <dbl>, CumTests <dbl>
> view(df_master)
> 
```

Q2:



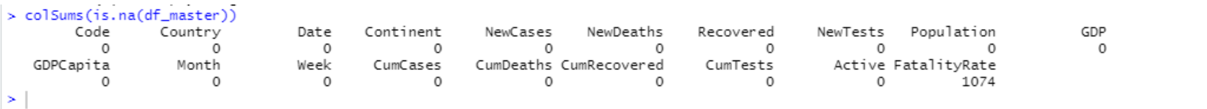
s	NewDeaths	Recovered	NewTests	Population	GDP	GDPCapita	Month	Week	CumCases	CumDeaths	CumRecovered	CumTests	Active	FatalityRate
1	0	0	0	44494502	637486	14400	3	10	1	0	0	0	1	0.000000000
1	0	0	0	44494502	637486	14400	3	10	2	0	0	0	2	0.000000000
6	0	0	0	44494502	637486	14400	3	10	8	0	0	0	8	0.000000000
1	1	0	0	44494502	637486	14400	3	10	9	1	0	0	8	0.111111111
3	0	0	0	44494502	637486	14400	3	10	12	1	0	0	11	0.083333333
7	0	0	0	44494502	637486	14400	3	11	19	1	0	0	18	0.052631579
12	0	0	0	44494502	637486	14400	3	11	31	1	0	0	30	0.032258065
3	1	1	0	44494502	637486	14400	3	11	34	2	1	0	31	0.058823529
11	0	0	0	44494502	637486	14400	3	11	45	2	1	0	42	0.044444444
11	0	0	0	44494502	637486	14400	3	11	56	2	1	0	53	0.035714286
9	0	2	0	44494502	637486	14400	3	11	65	2	3	0	60	0.030769231
14	0	0	0	44494502	637486	14400	3	12	79	2	3	0	74	0.025316456
18	0	0	0	44494502	637486	14400	3	12	97	2	3	0	92	0.020618557
31	1	0	0	44494502	637486	14400	3	12	128	3	3	0	122	0.023437500
30	0	0	0	44494502	637486	14400	3	12	158	3	3	0	152	0.018987342
67	1	0	0	44494502	637486	14400	3	12	225	4	3	0	218	0.017777778
41	0	0	0	44494502	637486	14400	3	12	266	4	3	0	259	0.015037594
35	0	49	0	44494502	637486	14400	3	12	301	4	52	0	245	0.013289037
86	2	0	0	44494502	637486	14400	3	13	387	6	52	0	329	0.015503876
115	2	11	0	44494502	637486	14400	3	13	502	8	63	0	431	0.015936255
87	4	9	0	44494502	637486	14400	3	13	589	12	72	0	505	0.020373514
101	5	0	0	44494502	637486	14400	3	13	690	17	72	0	601	0.024637681
55	2	0	0	44494502	637486	14400	3	13	745	19	72	0	654	0.025503356

Showing 1 to 23 of 6,821 entries, 19 total columns

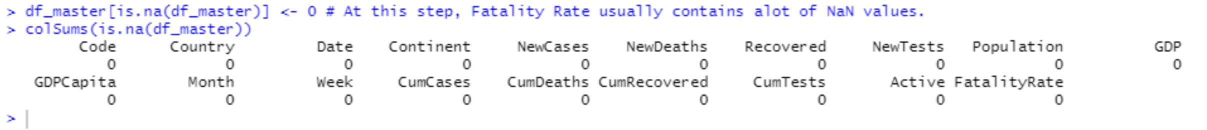
D:/University of Canberra/UC - Sem 1 2021/Introduction to Data Science/Final Assessment/

```
+   group_by(Country) %>%
+   mutate(Active = CumCases - CumDeaths - CumRecovered, FatalityRate = CumDeaths / CumCases)
> View(df_master)
> str(df_master)
grouped_df [6,821 x 19] (S3: grouped_df/tbl_df/tbl/data.frame)
 $ Code        : chr [1:6821] "ARG" "ARG" "ARG" "ARG" ...
 $ Country     : chr [1:6821] "Argentina" "Argentina" "Argentina" "Argentina" ...
 $ Date        : Date[1:6821], format: "2020-03-04" "2020-03-06" "2020-03-07" "2020-03-08" ...
 $ Continent   : chr [1:6821] "South America" "South America" "South America" "South America" ...
 $ NewCases    : num [1:6821] 1 1 6 1 3 7 12 3 11 11 ...
 $ NewDeaths   : num [1:6821] 0 0 0 1 0 0 0 1 0 0 ...
 $ Recovered   : num [1:6821] 0 0 0 0 0 0 0 1 0 0 ...
 $ NewTests    : num [1:6821] 0 0 0 0 0 0 0 0 0 0 ...
 $ Population  : num [1:6821] 44494502 44494502 44494502 44494502 44494502 ...
 $ GDP         : num [1:6821] 637486 637486 637486 637486 637486 ...
 $ GDPCapita   : num [1:6821] 14400 14400 14400 14400 14400 14400 14400 14400 14400 14400 ...
 $ Month       : num [1:6821] 3 3 3 3 3 3 3 3 3 3 ...
 $ Week        : num [1:6821] 10 10 10 10 10 11 11 11 11 11 ...
 $ CumCases    : num [1:6821] 1 2 8 9 12 19 31 34 45 56 ...
 $ CumDeaths   : num [1:6821] 0 0 0 1 1 1 1 2 2 2 ...
 $ CumRecovered: num [1:6821] 0 0 0 0 0 0 0 1 1 1 ...
 $ CumTests    : num [1:6821] 0 0 0 0 0 0 0 0 0 0 ...
 $ Active      : num [1:6821] 1 2 8 8 11 18 30 31 42 53 ...
 $ FatalityRate: num [1:6821] 0 0 0 0.1111 0.0833 ...
 - attr(*, "groups")= tibble [81 x 2] (S3: tbl_df/tbl/data.frame)
  ..$ Country: chr [1:81] "Argentina" "Australia" "Austria" "Bahrain" ...
  ..$ .rows  : list<int> [1:81]
```

NOTE: however, at this state, I discovered the there are number of NaN(s) in Fatality Rate (due to divide by 0 in some rows). So, I change them into 0 for easier calculation.

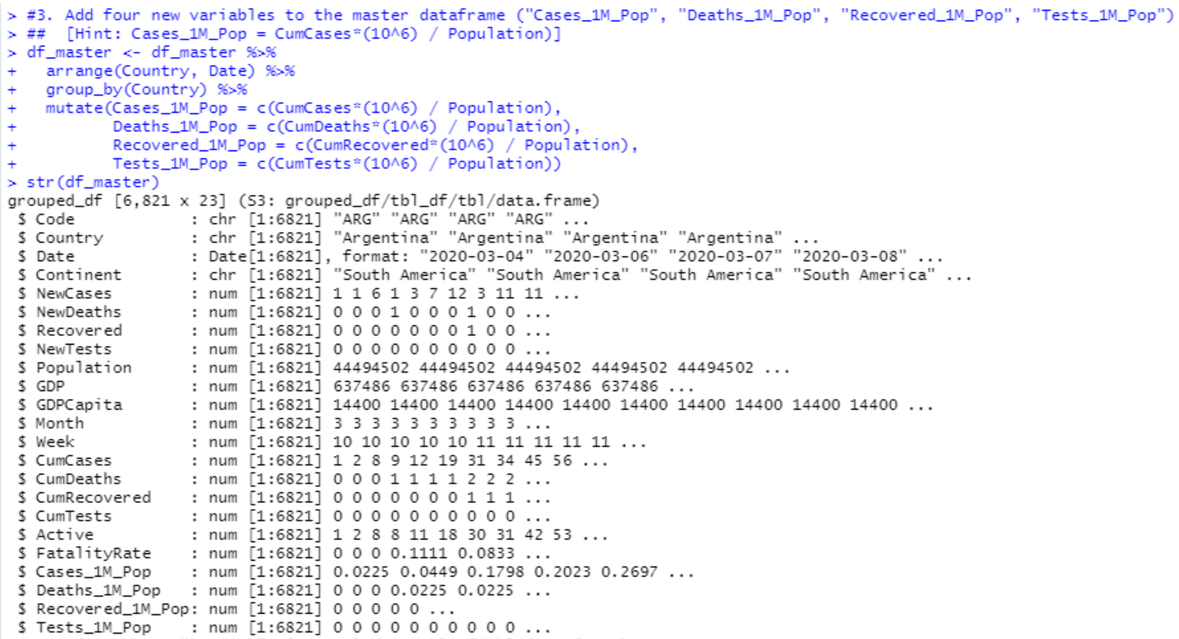


```
> colSums(is.na(df_master))
        Code      Country         Date    Continent     NewCases    NewDeaths    Recovered     NewTests   Population          GDP
           0            0            0            0            0            0            0            0            0            0
   GDPCapita        Month         Week     CumCases    CumDeaths CumRecovered     CumTests       Active FatalityRate
           0            0            0            0            0            0            0            0         1074
> |
```

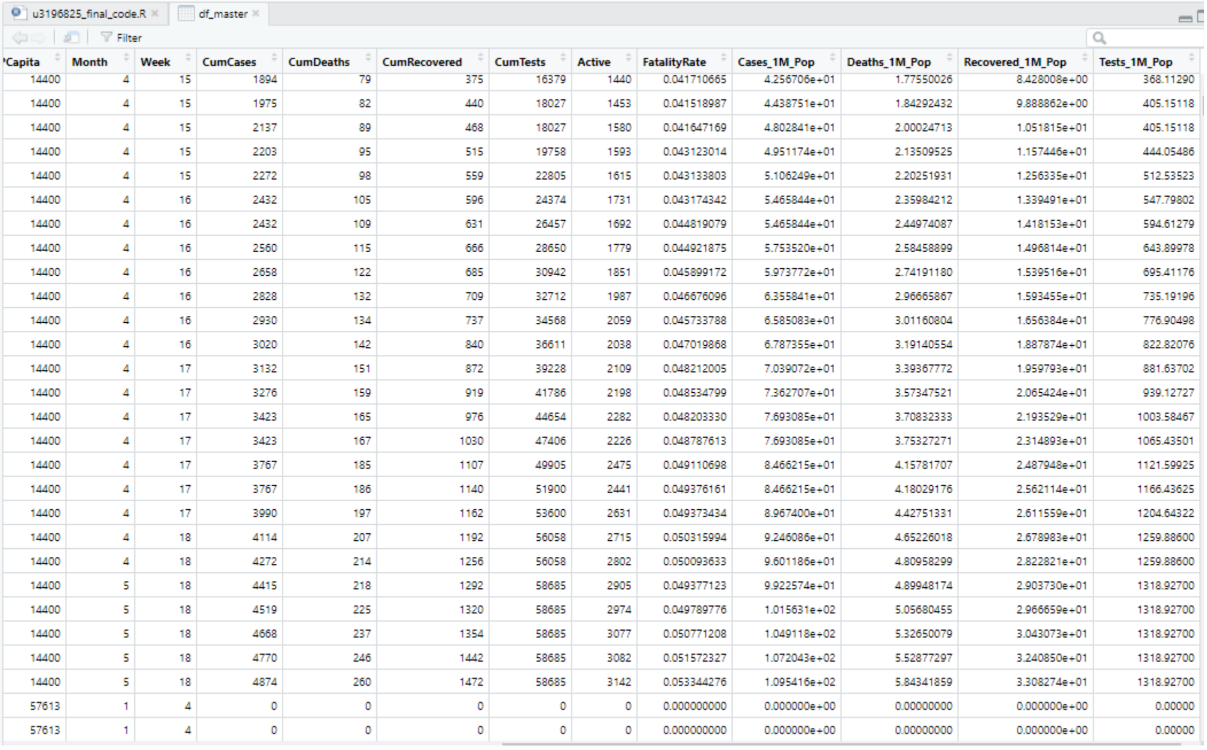


```
> df_master[is.na(df_master)] <- 0 # At this step, Fatality Rate usually contains alot of NaN values.
> colSums(is.na(df_master))
        Code      Country         Date    Continent     NewCases    NewDeaths    Recovered     NewTests   Population          GDP
           0            0            0            0            0            0            0            0            0            0
   GDPCapita        Month         Week     CumCases    CumDeaths CumRecovered     CumTests       Active FatalityRate
           0            0            0            0            0            0            0            0            0
> |
```

Q3:

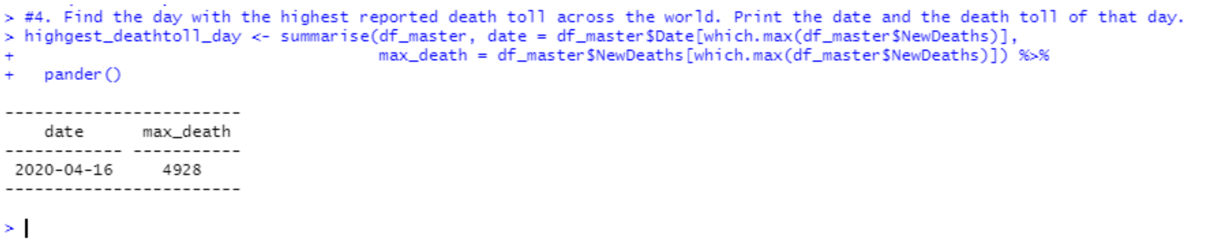


```
> #3. Add four new variables to the master dataframe ("Cases_1M_Pop", "Deaths_1M_Pop", "Recovered_1M_Pop", "Tests_1M_Pop")
> ##  [Hint: Cases_1M_Pop = CumCases*(10^6) / Population)]
> df_master <- df_master %>%
+   arrange(Country, Date) %>%
+   group_by(Country) %>%
+   mutate(Cases_1M_Pop = c(CumCases*(10^6) / Population),
+          Deaths_1M_Pop = c(CumDeaths*(10^6) / Population),
+          Recovered_1M_Pop = c(CumRecovered*(10^6) / Population),
+          Tests_1M_Pop = c(CumTests*(10^6) / Population))
> str(df_master)
grouped_df [6,821 x 23] (S3: grouped_df/tbl_df/tbl/data.frame)
 $ Code            : chr [1:6821] "ARG" "ARG" "ARG" "ARG" ...
 $ Country         : chr [1:6821] "Argentina" "Argentina" "Argentina" "Argentina" ...
 $ Date            : Date[1:6821], format: "2020-03-04" "2020-03-06" "2020-03-07" "2020-03-08" ...
 $ Continent       : chr [1:6821] "South America" "South America" "South America" "South America" ...
 $ NewCases        : num [1:6821] 1 1 6 1 3 7 12 3 11 11 ...
 $ NewDeaths       : num [1:6821] 0 0 0 1 0 0 0 1 0 0 ...
 $ Recovered       : num [1:6821] 0 0 0 0 0 0 0 1 0 0 ...
 $ NewTests        : num [1:6821] 0 0 0 0 0 0 0 0 0 0 ...
 $ Population      : num [1:6821] 44494502 44494502 44494502 44494502 44494502 ...
 $ GDP             : num [1:6821] 637486 637486 637486 637486 637486 ...
 $ GDPCapita       : num [1:6821] 14400 14400 14400 14400 14400 14400 14400 14400 14400 14400 ...
 $ Month           : num [1:6821] 3 3 3 3 3 3 3 3 3 3 ...
 $ Week            : num [1:6821] 10 10 10 10 10 11 11 11 11 11 ...
 $ CumCases        : num [1:6821] 1 2 8 9 12 19 31 34 45 56 ...
 $ CumDeaths       : num [1:6821] 0 0 0 1 1 1 1 2 2 2 ...
 $ CumRecovered    : num [1:6821] 0 0 0 0 0 0 0 1 1 1 ...
 $ CumTests        : num [1:6821] 0 0 0 0 0 0 0 0 0 0 ...
 $ Active          : num [1:6821] 1 2 8 8 11 18 30 31 42 53 ...
 $ FatalityRate    : num [1:6821] 0 0 0 0.1111 0.0833 ...
 $ Cases_1M_Pop    : num [1:6821] 0.0225 0.0449 0.1798 0.2023 0.2697 ...
 $ Deaths_1M_Pop   : num [1:6821] 0 0 0 0.0225 0.0225 ...
 $ Recovered_1M_Pop: num [1:6821] 0 0 0 0 0 ...
 $ Tests_1M_Pop    : num [1:6821] 0 0 0 0 0 0 0 0 0 0 ...
```



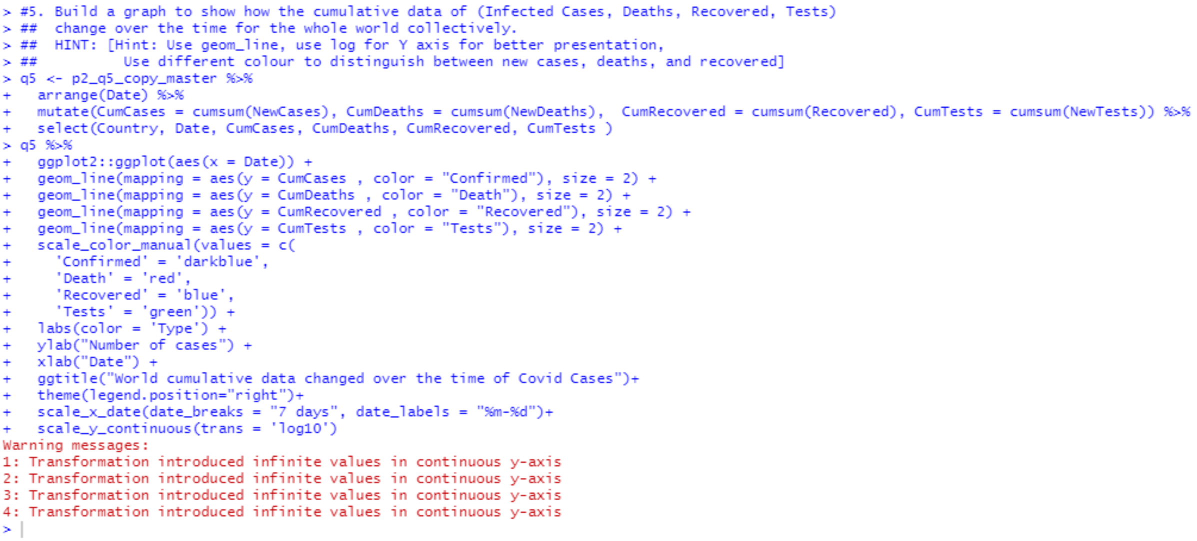
'Capita	Month	Week	CumCases	CumDeaths	CumRecovered	CumTests	Active	FatalityRate	Cases_1M_Pop	Deaths_1M_Pop	Recovered_1M_Pop	Tests_1M_Pop
14400	4	15	1894	79	375	16379	1440	0.041710665	4.256706e+01	1.77550026	8.428008e+00	368.11290
14400	4	15	1975	82	440	18027	1453	0.041518987	4.438751e+01	1.84292432	9.888862e+00	405.15118
14400	4	15	2137	89	468	18027	1580	0.041647169	4.802841e+01	2.00024713	1.051815e+01	405.15118
14400	4	15	2203	95	515	19758	1593	0.043123014	4.951174e+01	2.13509525	1.157446e+01	444.05486
14400	4	15	2272	98	559	22805	1615	0.043133803	5.106249e+01	2.20251931	1.256335e+01	512.53523
14400	4	16	2432	105	596	24374	1731	0.043174342	5.465844e+01	2.35984212	1.339491e+01	547.79802
14400	4	16	2432	109	631	26457	1692	0.044819079	5.465844e+01	2.44974087	1.418153e+01	594.61279
14400	4	16	2560	115	666	28650	1779	0.044921875	5.753520e+01	2.58458899	1.496814e+01	643.89978
14400	4	16	2658	122	685	30942	1851	0.045899172	5.973772e+01	2.74191180	1.539516e+01	695.41176
14400	4	16	2828	132	709	32712	1987	0.046676096	6.355841e+01	2.96665867	1.593455e+01	735.19196
14400	4	16	2930	134	737	34568	2059	0.045733788	6.585083e+01	3.01160804	1.656384e+01	776.90498
14400	4	16	3020	142	840	36611	2038	0.047019868	6.787355e+01	3.19140554	1.887874e+01	822.82076
14400	4	17	3132	151	872	39228	2109	0.048212005	7.039072e+01	3.39367772	1.959793e+01	881.63702
14400	4	17	3276	159	919	41786	2198	0.048534799	7.362707e+01	3.57347521	2.065424e+01	939.12727
14400	4	17	3423	165	976	44654	2282	0.048203330	7.693085e+01	3.70832333	2.193529e+01	1003.58467
14400	4	17	3423	167	1030	47406	2226	0.048787613	7.693085e+01	3.75327271	2.314893e+01	1065.43501
14400	4	17	3767	185	1107	49905	2475	0.049110698	8.466215e+01	4.15781707	2.487948e+01	1121.59925
14400	4	17	3767	186	1140	51900	2441	0.049376161	8.466215e+01	4.18029176	2.562114e+01	1166.43625
14400	4	17	3990	197	1162	53600	2631	0.049373434	8.967400e+01	4.42751331	2.611559e+01	1204.64322
14400	4	18	4114	207	1192	56058	2715	0.050315994	9.246086e+01	4.65226018	2.678983e+01	1259.88600
14400	4	18	4272	214	1256	56058	2802	0.050093633	9.601186e+01	4.80958299	2.822821e+01	1259.88600
14400	5	18	4415	218	1292	58685	2905	0.049377123	9.922574e+01	4.89948174	2.903730e+01	1318.92700
14400	5	18	4519	225	1320	58685	2974	0.049789776	1.015631e+02	5.05680455	2.966659e+01	1318.92700
14400	5	18	4668	237	1354	58685	3077	0.050771208	1.049118e+02	5.32650079	3.043073e+01	1318.92700
14400	5	18	4770	246	1442	58685	3082	0.051572327	1.072043e+02	5.52877297	3.240850e+01	1318.92700
14400	5	18	4874	260	1472	58685	3142	0.053344276	1.095416e+02	5.84341859	3.308274e+01	1318.92700
57613	1	4	0	0	0	0	0	0.000000000	0.000000e+00	0.00000000	0.000000e+00	0.00000
57613	1	4	0	0	0	0	0	0.000000000	0.000000e+00	0.00000000	0.000000e+00	0.00000

Q4:

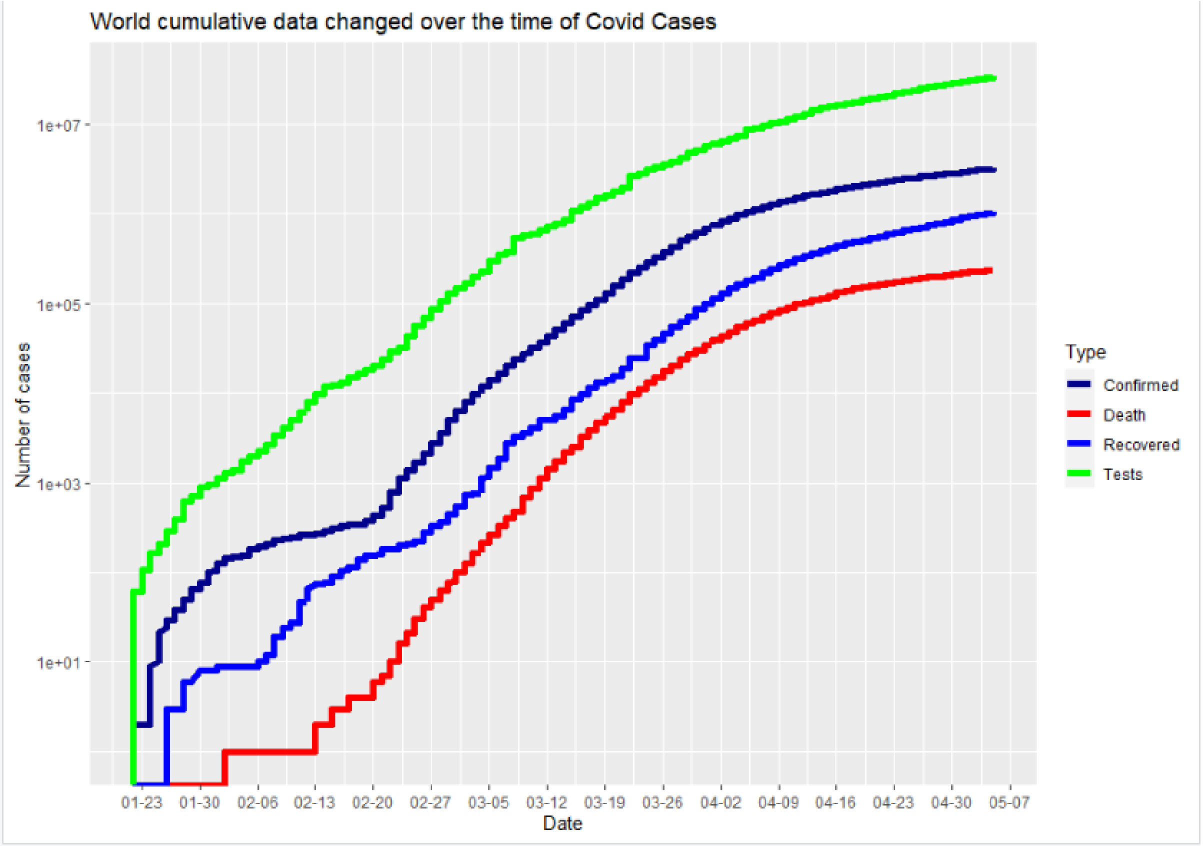


```
> #4. Find the day with the highest reported death toll across the world. Print the date and the death toll of that day.
> highest_deathtoll_day <- summarise(df_master, date = df_master$Date[which.max(df_master$NewDeaths)],
+                                    max_death = df_master$NewDeaths[which.max(df_master$NewDeaths)]) %>%
+   pander()

------------------------
    date      max_death
------------ -----------
 2020-04-16     4928
------------------------

> |
```

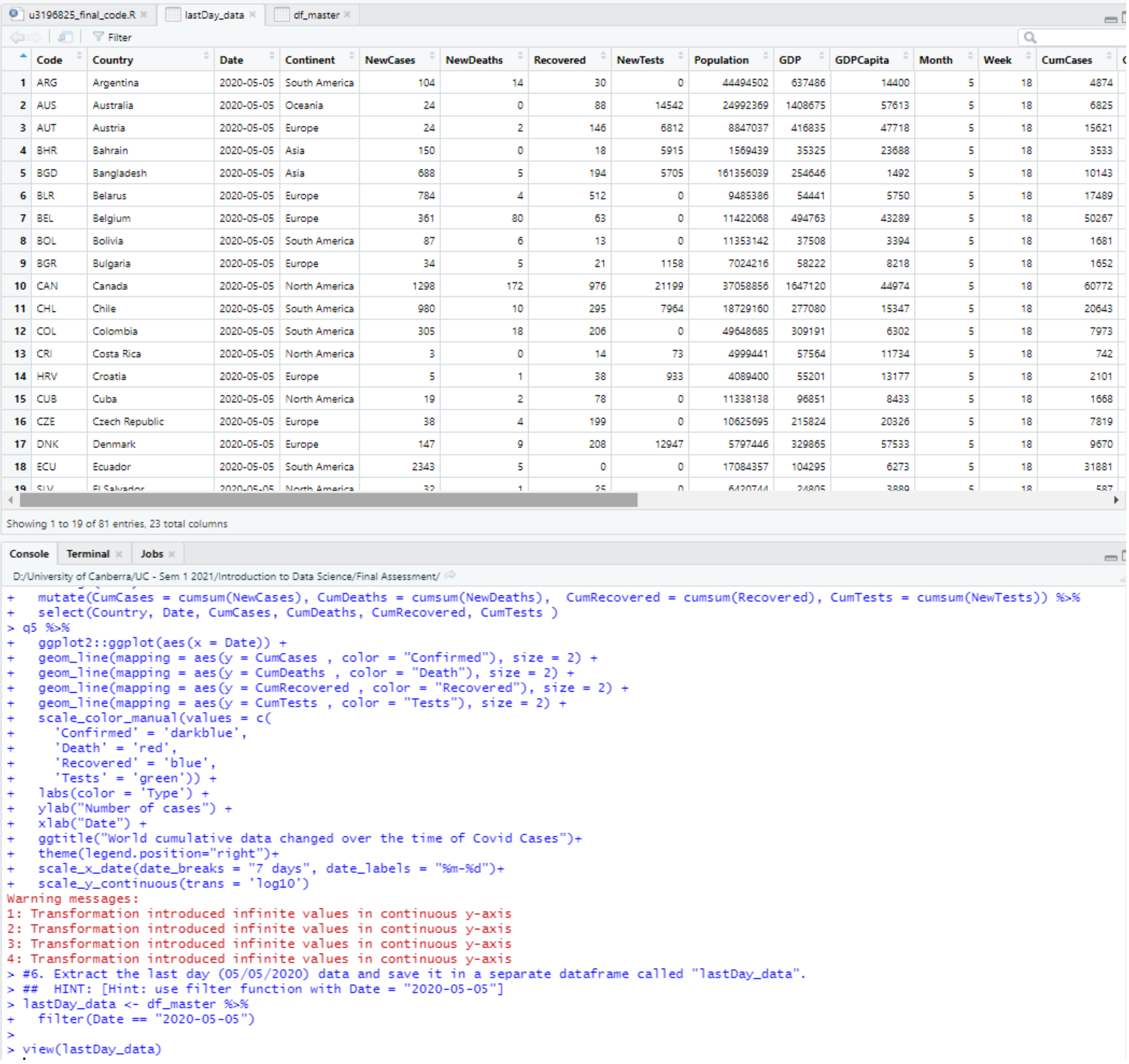
Q5:



```
> #5. Build a graph to show how the cumulative data of (Infected Cases, Deaths, Recovered, Tests)
> ##  change over the time for the whole world collectively.
> ##  HINT: [Hint: Use geom_line, use log for Y axis for better presentation,
> ##          Use different colour to distinguish between new cases, deaths, and recovered]
> q5 <- p2_q5_copy_master %>%
+   arrange(Date) %>%
+   mutate(CumCases = cumsum(NewCases), CumDeaths = cumsum(NewDeaths),  CumRecovered = cumsum(Recovered), CumTests = cumsum(NewTests)) %>%
+   select(Country, Date, CumCases, CumDeaths, CumRecovered, CumTests )
> q5 %>%
+   ggplot2::ggplot(aes(x = Date)) +
+   geom_line(mapping = aes(y = CumCases , color = "Confirmed"), size = 2) +
+   geom_line(mapping = aes(y = CumDeaths , color = "Death"), size = 2) +
+   geom_line(mapping = aes(y = CumRecovered , color = "Recovered"), size = 2) +
+   geom_line(mapping = aes(y = CumTests , color = "Tests"), size = 2) +
+   scale_color_manual(values = c(
+     'Confirmed' = 'darkblue',
+     'Death' = 'red',
+     'Recovered' = 'blue',
+     'Tests' = 'green')) +
+   labs(color = 'Type') +
+   ylab("Number of cases") +
+   xlab("Date") +
+   ggtitle("World cumulative data changed over the time of Covid Cases")+
+   theme(legend.position="right")+
+   scale_x_date(date_breaks = "7 days", date_labels = "%m-%d")+
+   scale_y_continuous(trans = 'log10')
Warning messages:
1: Transformation introduced infinite values in continuous y-axis
2: Transformation introduced infinite values in continuous y-axis
3: Transformation introduced infinite values in continuous y-axis
4: Transformation introduced infinite values in continuous y-axis
> |
```



Q6:



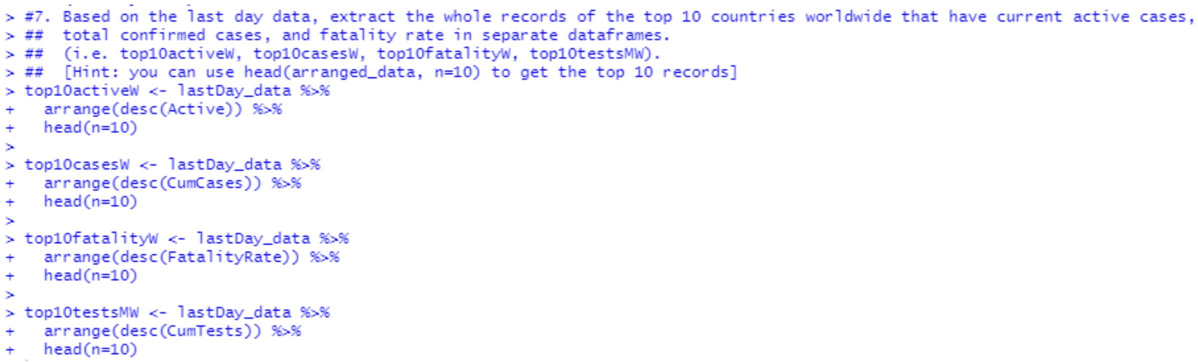
	Code	Country	Date	Continent	NewCases	NewDeaths	Recovered	NewTests	Population	GDP	GDPCapita	Month	Week	CumCases
1	ARG	Argentina	2020-05-05	South America	104	14	30	0	44494502	637486	14400	5	18	4874
2	AUS	Australia	2020-05-05	Oceania	24	0	88	14542	24992369	1408675	57613	5	18	6825
3	AUT	Austria	2020-05-05	Europe	24	2	146	6812	8847037	416835	47718	5	18	15621
4	BHR	Bahrain	2020-05-05	Asia	150	0	18	5915	1569439	35325	23688	5	18	3533
5	BGD	Bangladesh	2020-05-05	Asia	688	5	194	5705	161356039	254646	1492	5	18	10143
6	BLR	Belarus	2020-05-05	Europe	784	4	512	0	9485386	54441	5750	5	18	17489
7	BEL	Belgium	2020-05-05	Europe	361	80	63	0	11422068	494763	43289	5	18	50267
8	BOL	Bolivia	2020-05-05	South America	87	6	13	0	11353142	37508	3394	5	18	1681
9	BGR	Bulgaria	2020-05-05	Europe	34	5	21	1158	7024216	58222	8218	5	18	1652
10	CAN	Canada	2020-05-05	North America	1298	172	976	21199	37058856	1647120	44974	5	18	60772
11	CHL	Chile	2020-05-05	South America	980	10	295	7964	18729160	277080	15347	5	18	20643
12	COL	Colombia	2020-05-05	South America	305	18	206	0	49648685	309191	6302	5	18	7973
13	CRI	Costa Rica	2020-05-05	North America	3	0	14	73	4999441	57564	11734	5	18	742
14	HRV	Croatia	2020-05-05	Europe	5	1	38	933	4089400	55201	13177	5	18	2101
15	CUB	Cuba	2020-05-05	North America	19	2	78	0	11338138	96851	8433	5	18	1668
16	CZE	Czech Republic	2020-05-05	Europe	38	4	199	0	10625695	215824	20326	5	18	7819
17	DNK	Denmark	2020-05-05	Europe	147	9	208	12947	5797446	329865	57533	5	18	9670
18	ECU	Ecuador	2020-05-05	South America	2343	5	0	0	17084357	104295	6273	5	18	31881
19	SLV	El Salvador	2020-05-05	North America	32	1	25	0	6420744	24805	3889	5	18	587

Showing 1 to 19 of 81 entries, 23 total columns

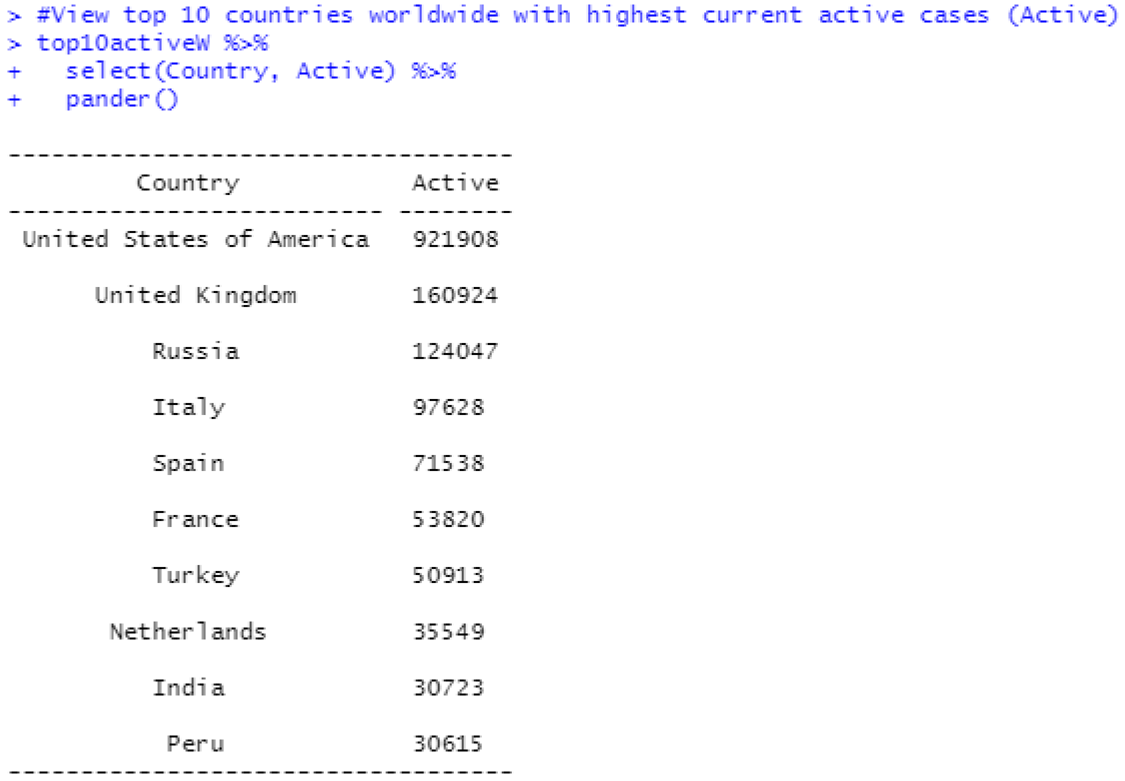
D:/University of Canberra/UC - Sem 1 2021/Introduction to Data Science/Final Assessment/

```
+   mutate(CumCases = cumsum(NewCases), CumDeaths = cumsum(NewDeaths),  CumRecovered = cumsum(Recovered), CumTests = cumsum(NewTests)) %>%
+   select(Country, Date, CumCases, CumDeaths, CumRecovered, CumTests )
> q5 %>%
+   ggplot2::ggplot(aes(x = Date)) +
+   geom_line(mapping = aes(y = CumCases , color = "Confirmed"), size = 2) +
+   geom_line(mapping = aes(y = CumDeaths , color = "Death"), size = 2) +
+   geom_line(mapping = aes(y = CumRecovered , color = "Recovered"), size = 2) +
+   geom_line(mapping = aes(y = CumTests , color = "Tests"), size = 2) +
+   scale_color_manual(values = c(
+     'Confirmed' = 'darkblue',
+     'Death' = 'red',
+     'Recovered' = 'blue',
+     'Tests' = 'green')) +
+   labs(color = 'Type') +
+   ylab("Number of cases") +
+   xlab("Date") +
+   ggtitle("World cumulative data changed over the time of Covid Cases")+
+   theme(legend.position="right")+
+   scale_x_date(date_breaks = "7 days", date_labels = "%m-%d")+
+   scale_y_continuous(trans = 'log10')
Warning messages:
1: Transformation introduced infinite values in continuous y-axis
2: Transformation introduced infinite values in continuous y-axis
3: Transformation introduced infinite values in continuous y-axis
4: Transformation introduced infinite values in continuous y-axis
> #6. Extract the last day (05/05/2020) data and save it in a separate dataframe called "lastDay_data".
> ##  HINT: [Hint: use filter function with Date = "2020-05-05"]
> lastDay_data <- df_master %>%
+   filter(Date == "2020-05-05")
> 
> view(lastDay_data)
```

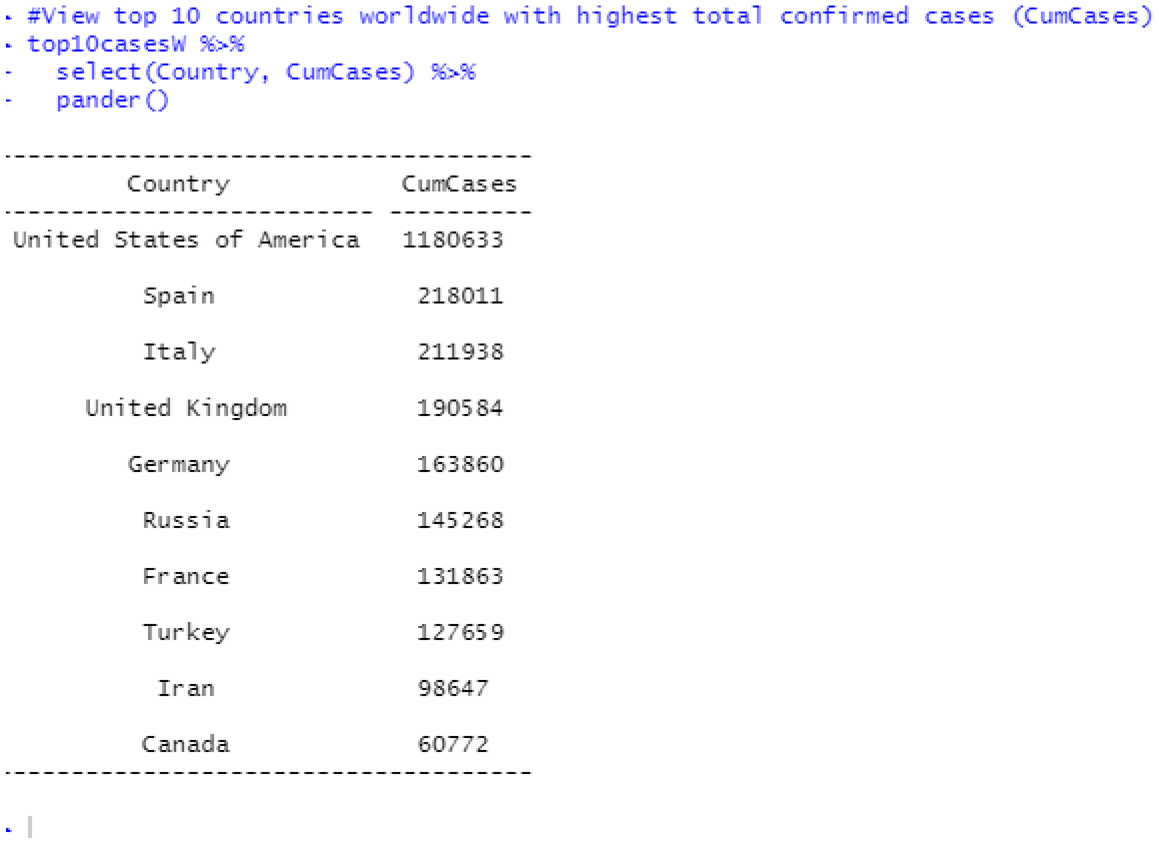
Q7:



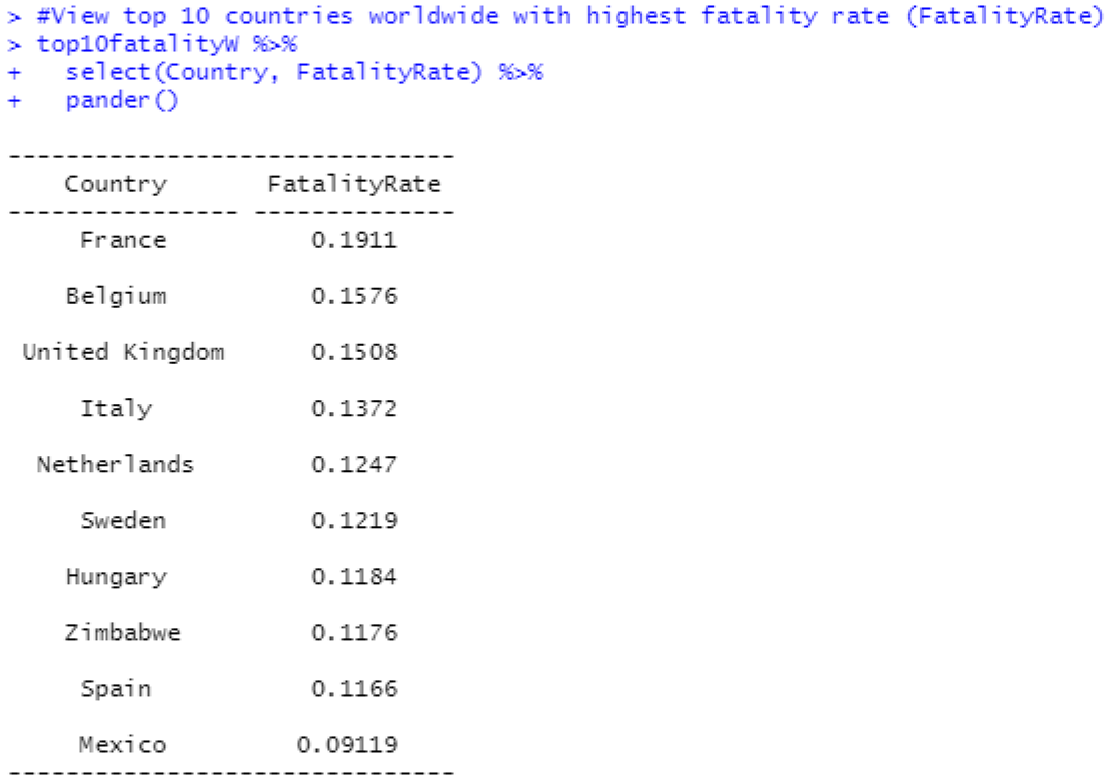
```
> #7. Based on the last day data, extract the whole records of the top 10 countries worldwide that have current active cases,
> ##  total confirmed cases, and fatality rate in separate dataframes.
> ##  (i.e. top10activeW, top10casesW, top10fatalityW, top10testsMW).
> ##  [Hint: you can use head(arranged_data, n=10) to get the top 10 records]
> top10activeW <- lastDay_data %>%
+   arrange(desc(Active)) %>%
+   head(n=10)
>
> top10casesW <- lastDay_data %>%
+   arrange(desc(CumCases)) %>%
+   head(n=10)
>
> top10fatalityW <- lastDay_data %>%
+   arrange(desc(FatalityRate)) %>%
+   head(n=10)
>
> top10testsMW <- lastDay_data %>%
+   arrange(desc(CumTests)) %>%
+   head(n=10)
```



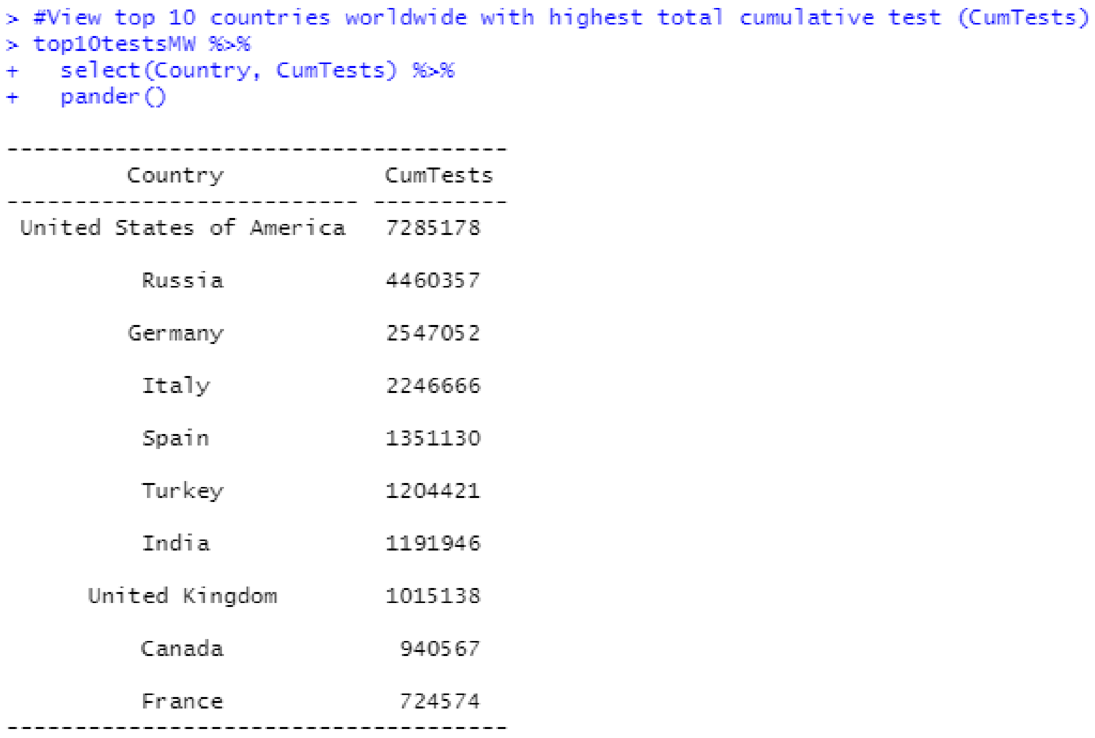
```
> #View top 10 countries worldwide with highest current active cases (Active)
> top10activeW %>%
+   select(Country, Active) %>%
+   pander()

-----------------------------------
          Country             Active
-------------------------- --------
 United States of America   921908

      United Kingdom        160924

          Russia            124047

          Italy             97628

          Spain             71538

          France            53820

          Turkey            50913

        Netherlands         35549

          India             30723

           Peru             30615
-----------------------------------
```



```
> #View top 10 countries worldwide with highest total confirmed cases (CumCases)
> top10casesW %>%
+   select(Country, CumCases) %>%
+   pander()

------------------------------------
        Country           CumCases
------------------------- ----------
United States of America   1180633

         Spain              218011

         Italy              211938

     United Kingdom         190584

        Germany             163860

         Russia             145268

         France             131863

         Turkey             127659

          Iran              98647

         Canada             60772
------------------------------------

> |
```

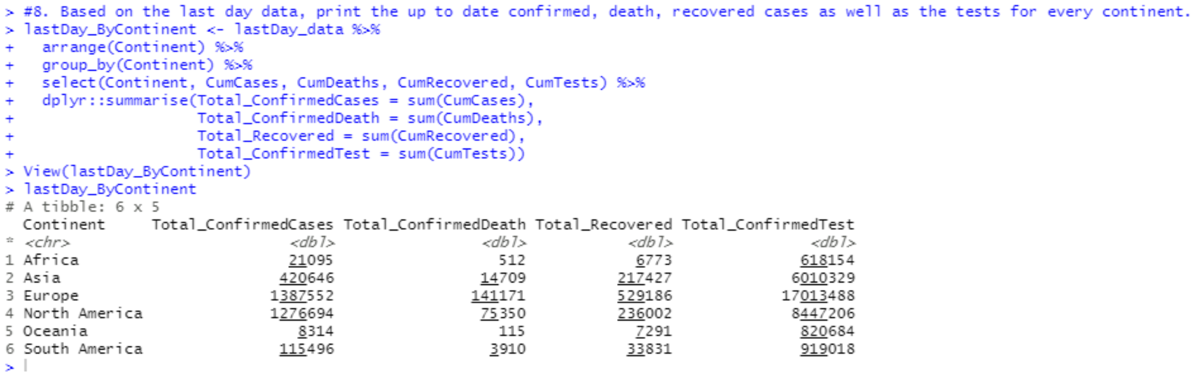


```
> #View top 10 countries worldwide with highest fatality rate (FatalityRate)
> top10fatalityW %>%
+   select(Country, FatalityRate) %>%
+   pander()

------------------------------
    Country       FatalityRate
---------------- -------------
     France          0.1911

    Belgium          0.1576

 United Kingdom      0.1508

     Italy           0.1372

  Netherlands        0.1247

     Sweden          0.1219

    Hungary          0.1184

    Zimbabwe         0.1176

     Spain           0.1166

     Mexico         0.09119
------------------------------
```



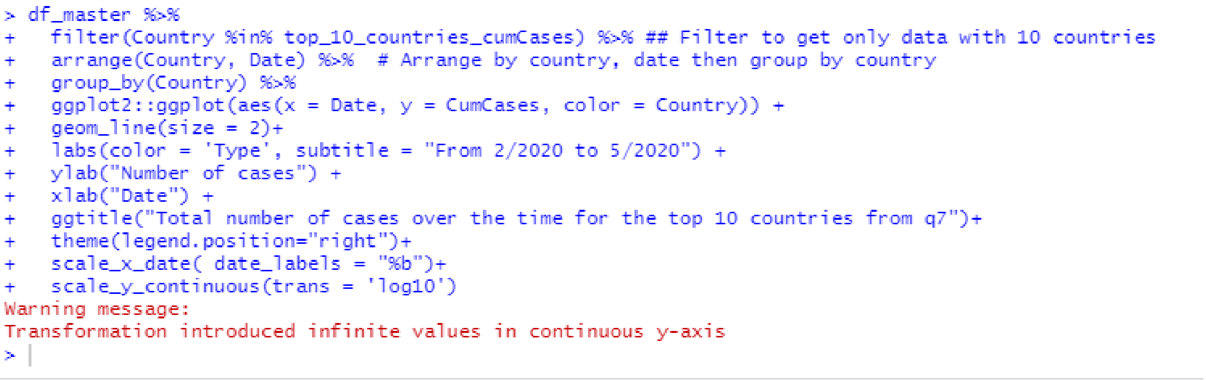
```
> #View top 10 countries worldwide with highest total cumulative test (CumTests)
> top10testsMW %>%
+   select(Country, CumTests) %>%
+   pander()

-------------------------------------
         Country            CumTests
-------------------------- ----------
 United States of America   7285178

          Russia            4460357

         Germany            2547052

          Italy             2246666

          Spain             1351130

          Turkey            1204421

          India             1191946

      United Kingdom        1015138

          Canada             940567

          France             724574
-------------------------------------
```

Q8:

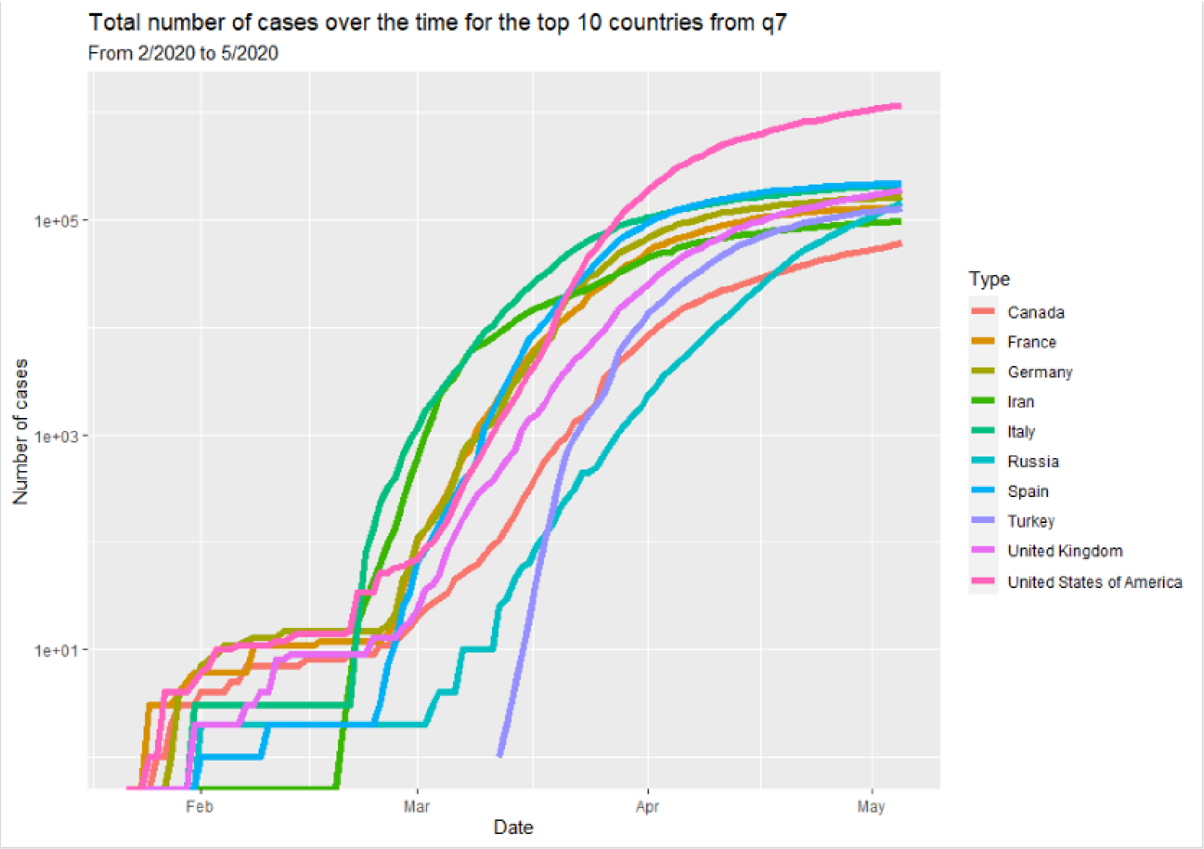


```
> #8. Based on the last day data, print the up to date confirmed, death, recovered cases as well as the tests for every continent.
> lastDay_ByContinent <- lastDay_data %>%
+   arrange(Continent) %>%
+   group_by(Continent) %>%
+   select(Continent, CumCases, CumDeaths, CumRecovered, CumTests) %>%
+   dplyr::summarise(Total_ConfirmedCases = sum(CumCases),
+                    Total_ConfirmedDeath = sum(CumDeaths),
+                    Total_Recovered = sum(CumRecovered),
+                    Total_ConfirmedTest = sum(CumTests))
> View(lastDay_ByContinent)
> lastDay_ByContinent
# A tibble: 6 x 5
  Continent     Total_ConfirmedCases Total_ConfirmedDeath Total_Recovered Total_ConfirmedTest
* <chr>                        <dbl>                <dbl>           <dbl>               <dbl>
1 Africa                       21095                  512            6773              618154
2 Asia                        420646                14709          217427             6010329
3 Europe                     1387552               141171          529186            17013488
4 North America              1276694                75350          236002             8447206
5 Oceania                       8314                  115            7291              820684
6 South America               115496                 3910           33831              919018
> 
```

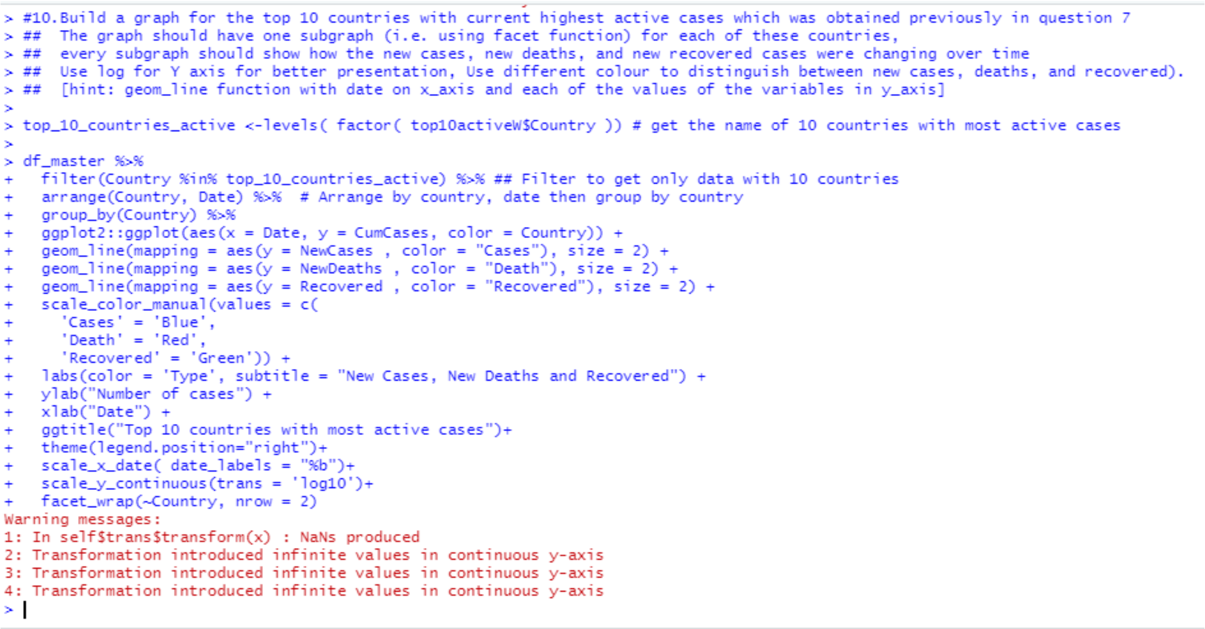
Q9:



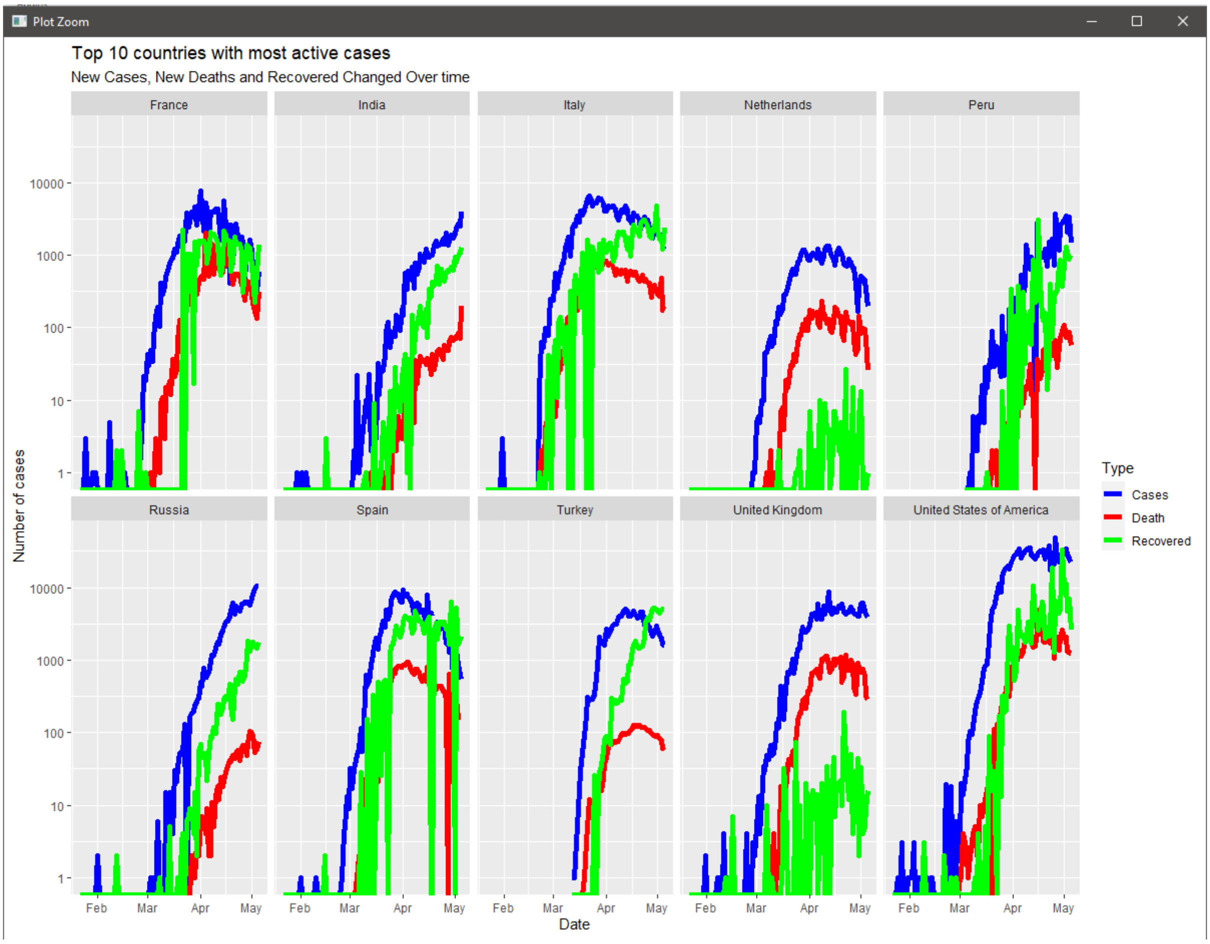
```
> df_master %>%
+   filter(Country %in% top_10_countries_cumCases) %>% ## Filter to get only data with 10 countries
+   arrange(Country, Date) %>%  # Arrange by country, date then group by country
+   group_by(Country) %>%
+   ggplot2::ggplot(aes(x = Date, y = CumCases, color = Country)) +
+   geom_line(size = 2)+
+   labs(color = 'Type', subtitle = "From 2/2020 to 5/2020") +
+   ylab("Number of cases") +
+   xlab("Date") +
+   ggtitle("Total number of cases over the time for the top 10 countries from q7")+
+   theme(legend.position="right")+
+   scale_x_date( date_labels = "%b")+
+   scale_y_continuous(trans = 'log10')
Warning message:
Transformation introduced infinite values in continuous y-axis
> 
```



Q10:

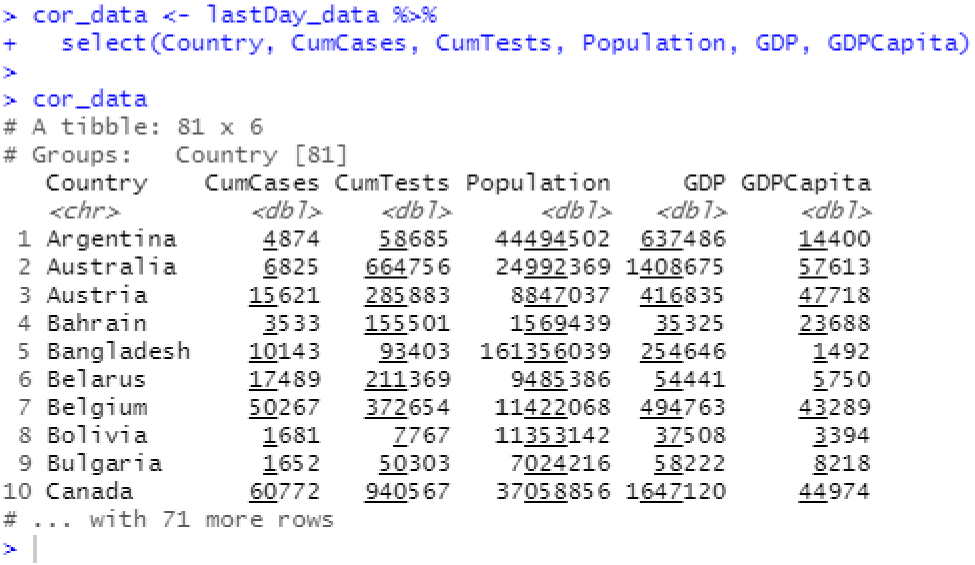


```
> #10.Build a graph for the top 10 countries with current highest active cases which was obtained previously in question 7
> ##  The graph should have one subgraph (i.e. using facet function) for each of these countries,
> ##  every subgraph should show how the new cases, new deaths, and new recovered cases were changing over time
> ##  Use log for Y axis for better presentation, Use different colour to distinguish between new cases, deaths, and recovered).
> ##  [hint: geom_line function with date on x_axis and each of the values of the variables in y_axis]
>
> top_10_countries_active <-levels( factor( top10activeW$Country )) # get the name of 10 countries with most active cases
>
> df_master %>%
+   filter(Country %in% top_10_countries_active) %>% ## Filter to get only data with 10 countries
+   arrange(Country, Date) %>%  # Arrange by country, date then group by country
+   group_by(Country) %>%
+   ggplot2::ggplot(aes(x = Date, y = CumCases, color = Country)) +
+   geom_line(mapping = aes(y = NewCases , color = "Cases"), size = 2) +
+   geom_line(mapping = aes(y = NewDeaths , color = "Death"), size = 2) +
+   geom_line(mapping = aes(y = Recovered , color = "Recovered"), size = 2) +
+   scale_color_manual(values = c(
+     'Cases' = 'Blue',
+     'Death' = 'Red',
+     'Recovered' = 'Green')) +
+   labs(color = 'Type', subtitle = "New Cases, New Deaths and Recovered") +
+   ylab("Number of cases") +
+   xlab("Date") +
+   ggtitle("Top 10 countries with most active cases")+
+   theme(legend.position="right")+
+   scale_x_date( date_labels = "%b")+
+   scale_y_continuous(trans = 'log10')+
+   facet_wrap(~Country, nrow = 2)
Warning messages:
1: In self$trans$transform(x) : NaNs produced
2: Transformation introduced infinite values in continuous y-axis
3: Transformation introduced infinite values in continuous y-axis
4: Transformation introduced infinite values in continuous y-axis
> |
```



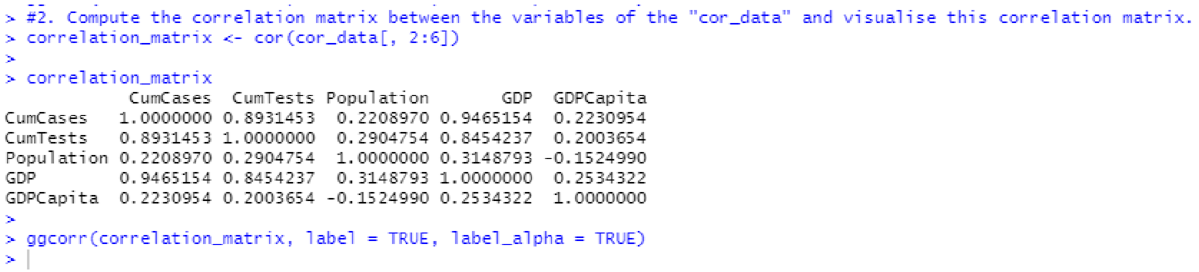
Task 3: Data-Driven Modelling:

Q1:

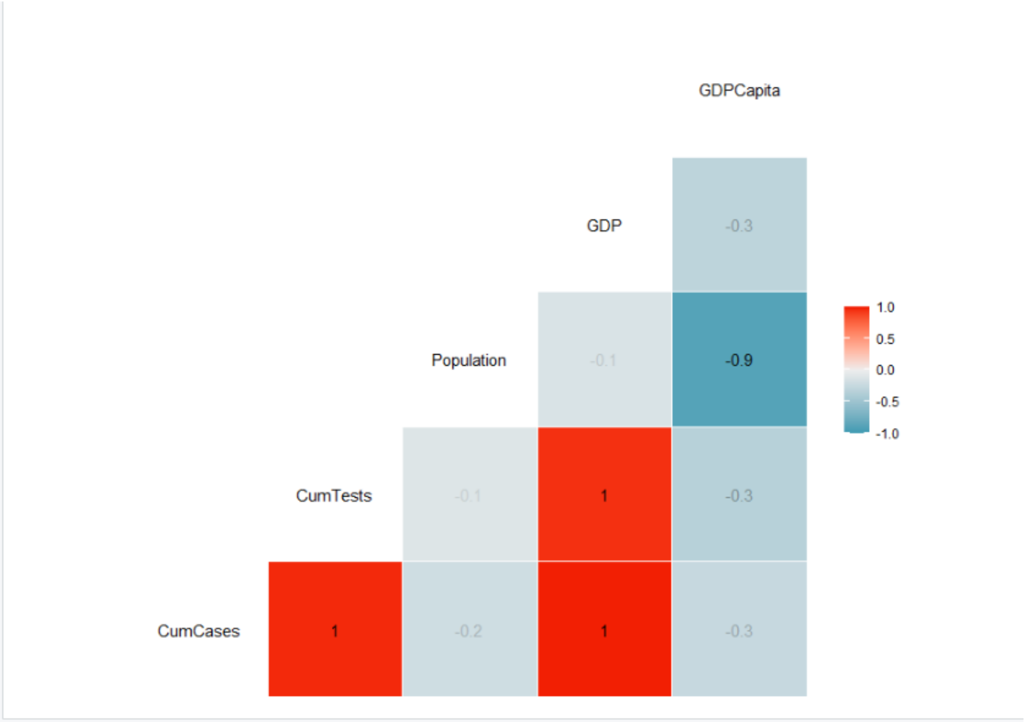


```
> cor_data <- lastDay_data %>%
+   select(Country, CumCases, CumTests, Population, GDP, GDPCapita)
>
> cor_data
# A tibble: 81 x 6
# Groups:   Country [81]
   Country    CumCases CumTests Population     GDP GDPCapita
   <chr>         <dbl>    <dbl>      <dbl>   <dbl>     <dbl>
 1 Argentina      4874    58685   44494502  637486     14400
 2 Australia      6825   664756   24992369 1408675     57613
 3 Austria       15621   285883    8847037  416835     47718
 4 Bahrain        3533   155501    1569439   35325     23688
 5 Bangladesh    10143    93403  161356039  254646      1492
 6 Belarus       17489   211369    9485386   54441      5750
 7 Belgium     50267   372654   11422068  494763     43289
 8 Bolivia        1681     7767   11353142   37508      3394
 9 Bulgaria       1652    50303    7024216   58222      8218
10 Canada        60772   940567   37058856 1647120     44974
# ... with 71 more rows
> |
```

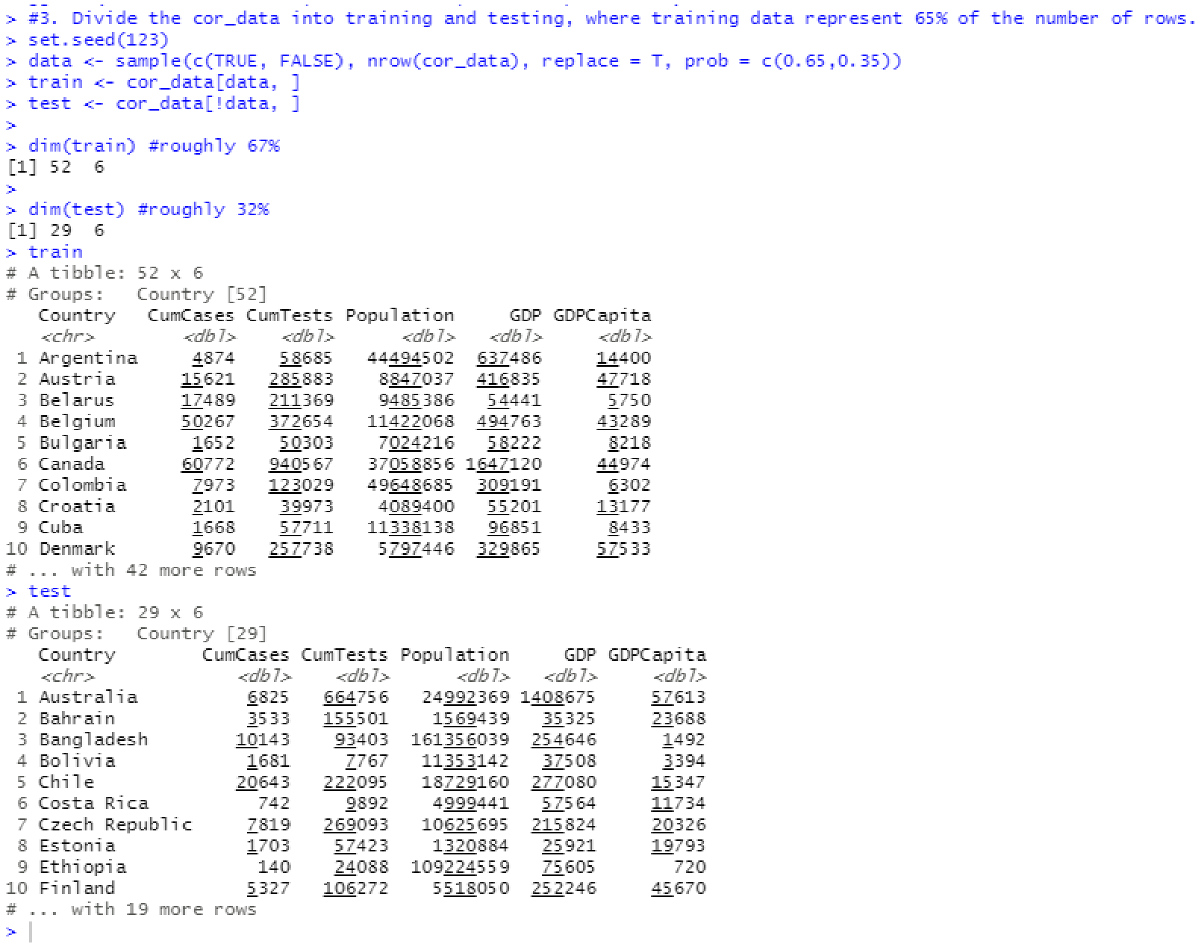
Q2:



```
> #2. Compute the correlation matrix between the variables of the "cor_data" and visualise this correlation matrix.
> correlation_matrix <- cor(cor_data[, 2:6])
>
> correlation_matrix
            CumCases  CumTests Population       GDP  GDPCapita
CumCases   1.0000000 0.8931453  0.2208970 0.9465154  0.2230954
CumTests   0.8931453 1.0000000  0.2904754 0.8454237  0.2003654
Population 0.2208970 0.2904754  1.0000000 0.3148793 -0.1524990
GDP        0.9465154 0.8454237  0.3148793 1.0000000  0.2534322
GDPCapita  0.2230954 0.2003654 -0.1524990 0.2534322  1.0000000
>
> ggcorr(correlation_matrix, label = TRUE, label_alpha = TRUE)
> |
```



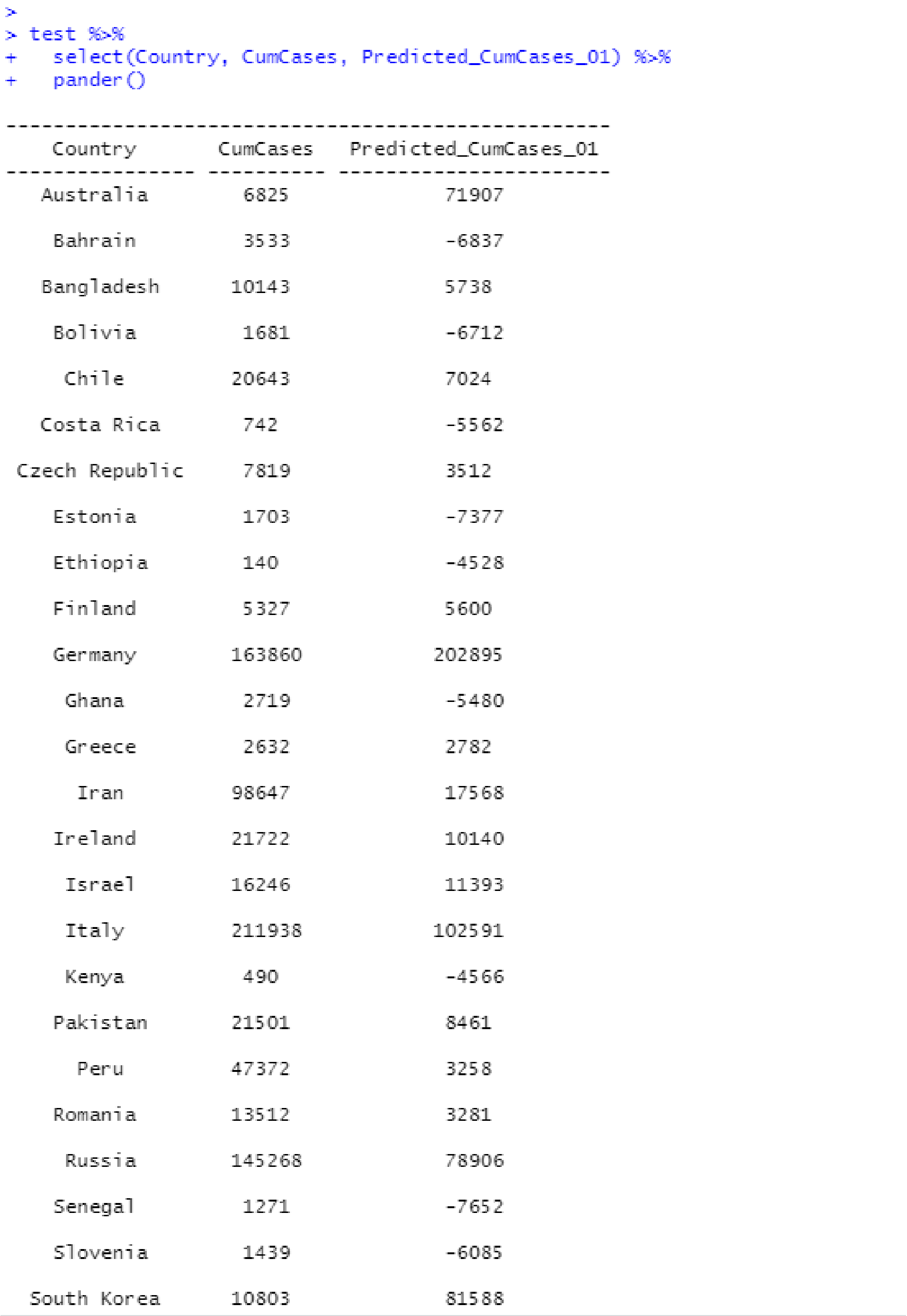
Q3:



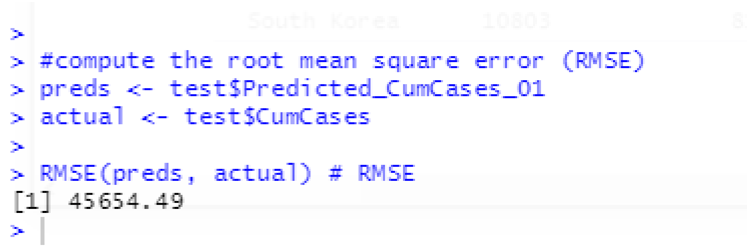
```
> #3. Divide the cor_data into training and testing, where training data represent 65% of the number of rows.
> set.seed(123)
> data <- sample(c(TRUE, FALSE), nrow(cor_data), replace = T, prob = c(0.65,0.35))
> train <- cor_data[data, ]
> test <- cor_data[!data, ]
> 
> dim(train) #roughly 67%
[1] 52  6
> 
> dim(test) #roughly 32%
[1] 29  6
> train
# A tibble: 52 x 6
# Groups:   Country [52]
   Country   CumCases CumTests Population     GDP GDPCapita
   <chr>        <dbl>    <dbl>      <dbl>   <dbl>     <dbl>
 1 Argentina     4874    58685   44494502  637486     14400
 2 Austria      15621   285883    8847037  416835     47718
 3 Belarus      17489   211369    9485386   54441      5750
 4 Belgium      50267   372654   11422068  494763     43289
 5 Bulgaria      1652    50303    7024216   58222      8218
 6 Canada       60772   940567   37058856 1647120     44974
 7 Colombia      7973   123029   49648685  309191      6302
 8 Croatia       2101    39973    4089400   55201     13177
 9 Cuba          1668    57711   11338138   96851      8433
10 Denmark       9670   257738    5797446  329865     57533
# ... with 42 more rows
> test
# A tibble: 29 x 6
# Groups:   Country [29]
   Country        CumCases CumTests Population     GDP GDPCapita
   <chr>             <dbl>    <dbl>      <dbl>   <dbl>     <dbl>
 1 Australia          6825   664756   24992369 1408675     57613
 2 Bahrain            3533   155501    1569439   35325     23688
 3 Bangladesh        10143    93403  161356039  254646      1492
 4 Bolivia            1681     7767   11353142   37508      3394
 5 Chile             20643   222095   18729160  277080     15347
 6 Costa Rica          742     9892    4999441   57564     11734
 7 Czech Republic     7819   269093   10625695  215824     20326
 8 Estonia            1703    57423    1320884   25921     19793
 9 Ethiopia            140    24088  109224559   75605       720
10 Finland            5327   106272    5518050  252246     45670
# ... with 19 more rows
> 
```

Q4:

```
> #4. Train a linear regression model to predict cumulative cases from the GDP of the countries.
> ##  Then, evaluate this model on the test data and print the root mean square error value.
> lm_model_01 <- lm(CumCases ~ GDP, data = train) # Train model using "train" data.
> 
> print(lm_model_01)

Call:
lm(formula = CumCases ~ GDP, data = train)

Coefficients:
(Intercept)          GDP  
 -8.863e+03    5.734e-02  

> 
> summary(lm_model_01)

Call:
lm(formula = CumCases ~ GDP, data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-255278   -3046    5652    9081   87697 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -8.863e+03  6.456e+03  -1.373    0.176    
GDP          5.734e-02  2.237e-03  25.630   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 44330 on 50 degrees of freedom
Multiple R-squared:  0.9293,	Adjusted R-squared:  0.9279 
F-statistic: 656.9 on 1 and 50 DF,  p-value: < 2.2e-16

> 
> # predicting
> test$Predicted_CumCases_01 <- predict(lm_model_01, test)
```

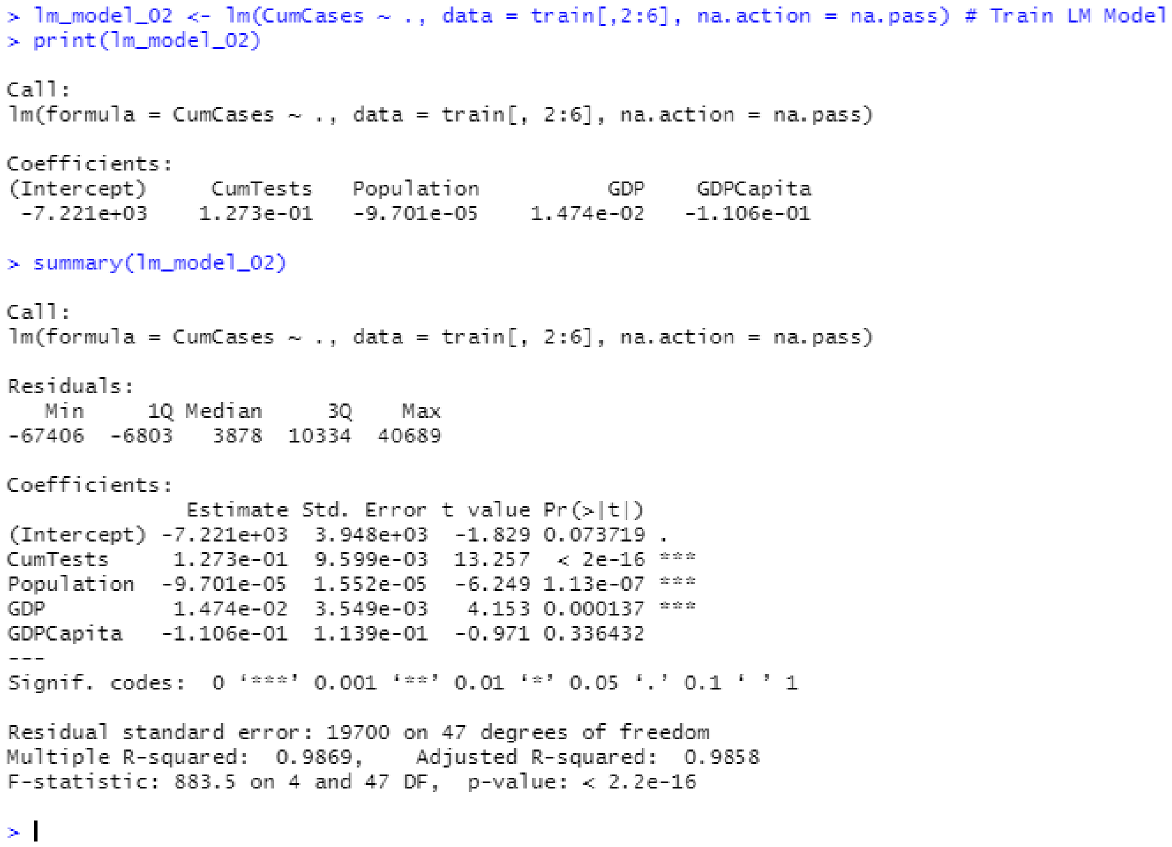


```
>
> test %>%
+   select(Country, CumCases, Predicted_CumCases_01) %>%
+   pander()

---------------------------------------------------
    Country       CumCases   Predicted_CumCases_01
---------------- ---------- -----------------------
   Australia        6825             71907

    Bahrain         3533             -6837

   Bangladesh      10143              5738

    Bolivia         1681             -6712

     Chile         20643              7024

   Costa Rica        742             -5562

 Czech Republic     7819              3512

    Estonia         1703             -7377

    Ethiopia         140             -4528

    Finland         5327              5600

    Germany        163860            202895

     Ghana          2719             -5480

     Greece         2632              2782

      Iran         98647             17568

    Ireland        21722             10140

     Israel        16246             11393

     Italy         211938            102591

     Kenya           490             -4566

    Pakistan       21501              8461

      Peru         47372              3258

    Romania        13512              3281

     Russia        145268             78906

    Senegal         1271             -7652

    Slovenia        1439             -6085

  South Korea      10803             81588
```

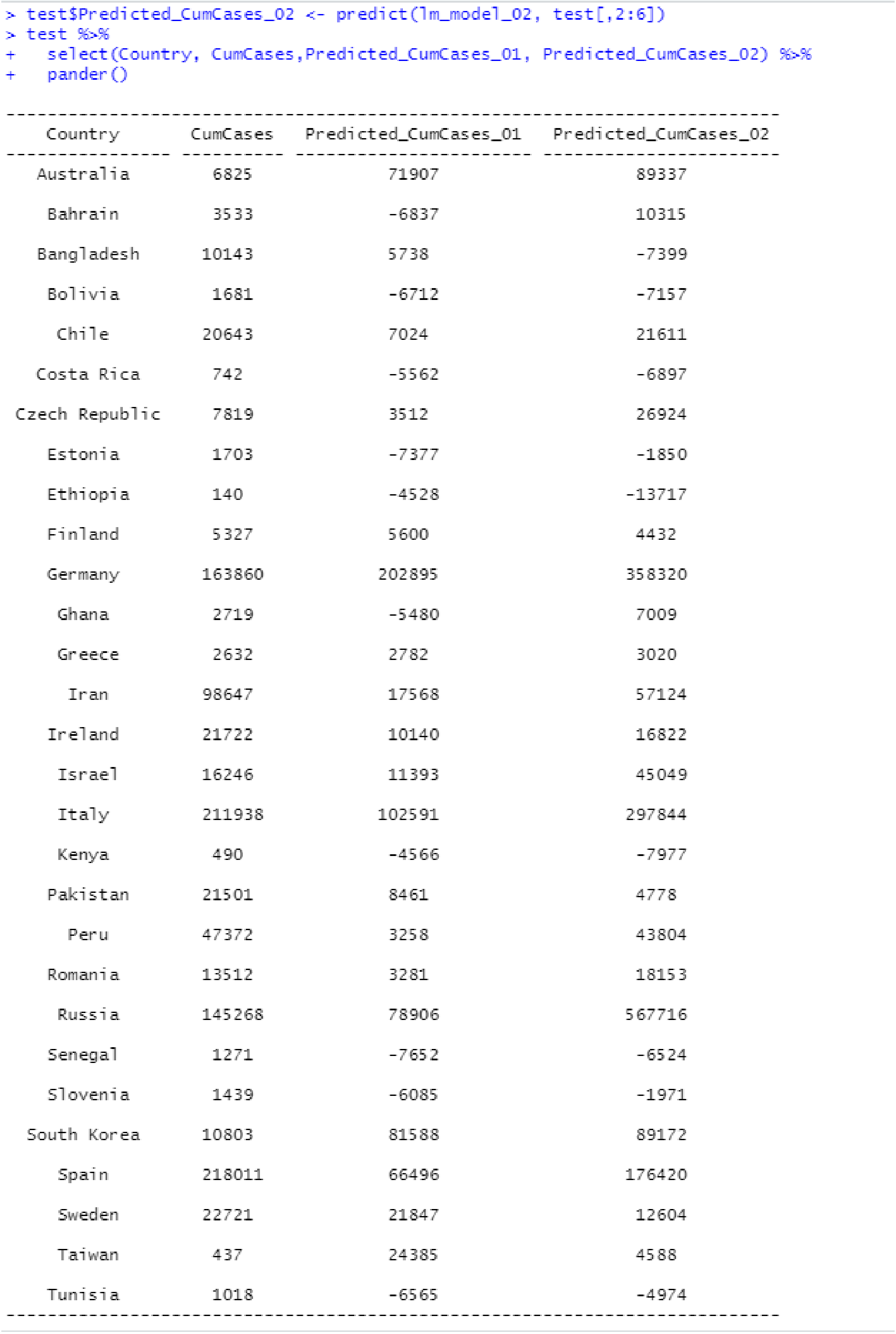


```
>
> #compute the root mean square error (RMSE)
> preds <- test$Predicted_CumCases_01
> actual <- test$CumCases
>
> RMSE(preds, actual) # RMSE
[1] 45654.49
>
```

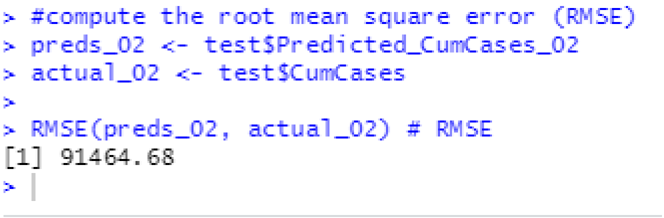
Q5:



```
> lm_model_02 <- lm(CumCases ~ ., data = train[,2:6], na.action = na.pass) # Train LM Model
> print(lm_model_02)

Call:
lm(formula = CumCases ~ ., data = train[, 2:6], na.action = na.pass)

Coefficients:
(Intercept)     CumTests   Population          GDP    GDPCapita  
 -7.221e+03    1.273e-01   -9.701e-05    1.474e-02   -1.106e-01  

> summary(lm_model_02)

Call:
lm(formula = CumCases ~ ., data = train[, 2:6], na.action = na.pass)

Residuals:
   Min     1Q Median     3Q    Max 
-67406  -6803   3878  10334  40689 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -7.221e+03  3.948e+03  -1.829 0.073719 .  
CumTests     1.273e-01  9.599e-03  13.257  < 2e-16 ***
Population  -9.701e-05  1.552e-05  -6.249 1.13e-07 ***
GDP          1.474e-02  3.549e-03   4.153 0.000137 ***
GDPCapita   -1.106e-01  1.139e-01  -0.971 0.336432    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19700 on 47 degrees of freedom
Multiple R-squared:  0.9869,	Adjusted R-squared:  0.9858 
F-statistic: 883.5 on 4 and 47 DF,  p-value: < 2.2e-16

> 
```



```
> test$Predicted_CumCases_02 <- predict(lm_model_02, test[,2:6])
> test %>%
+   select(Country, CumCases,Predicted_CumCases_01, Predicted_CumCases_02) %>%
+   pander()

------------------------------------------------------------------------
    Country      CumCases   Predicted_CumCases_01   Predicted_CumCases_02
--------------- ---------- ----------------------- -----------------------
   Australia       6825            71907                   89337

    Bahrain        3533            -6837                   10315

   Bangladesh     10143            5738                    -7399

    Bolivia        1681            -6712                   -7157

     Chile        20643            7024                    21611

   Costa Rica      742             -5562                   -6897

 Czech Republic    7819            3512                    26924

    Estonia        1703            -7377                   -1850

    Ethiopia       140             -4528                   -13717

    Finland        5327            5600                    4432

    Germany       163860          202895                  358320

     Ghana         2719            -5480                   7009

     Greece        2632            2782                    3020

      Iran        98647            17568                   57124

    Ireland       21722            10140                   16822

     Israel       16246            11393                   45049

     Italy        211938          102591                  297844

     Kenya         490             -4566                   -7977

    Pakistan      21501            8461                    4778

      Peru        47372            3258                    43804

    Romania       13512            3281                    18153

     Russia       145268           78906                  567716

    Senegal        1271            -7652                   -6524

    Slovenia       1439            -6085                   -1971

  South Korea     10803            81588                   89172

     Spain        218011           66496                  176420

     Sweden       22721            21847                   12604

     Taiwan        437             24385                   4588

    Tunisia        1018            -6565                   -4974
------------------------------------------------------------------------
```



```
> #compute the root mean square error (RMSE)
> preds_02 <- test$Predicted_CumCases_02
> actual_02 <- test$CumCases
>
> RMSE(preds_02, actual_02) # RMSE
[1] 91464.68
>
```