Martin Sikora 3/05/2025

data analysis 2025 - wrangle

ATTENTIVE ELEPHANT 3/5/25 9:08AM

```
slice_max(covid_data,
new_cases_per_million, n = 10)%>%
 select(country, new_cases_per_million)
```

♡ 0 ◯ 0

DEPENDABLE PORCUPINE 3/5/25 9:09AM

```
covid_data %>%
 slice_max(new_cases_per_million,
n=10) %>%
 select(country, new_cases_per_million) %>%
 view()
```

♡ 0 ◯ 0

	country	new_cases_per_million
1	Wallis and Futuna	230762.55
2	Falkland Islands	208819.34
3	Saint Helena	158130.00
4	Falkland Islands	137126.60
5	Nauru	104982.66
6	Saint Pierre and Miquelon	100173.61
7	Tuvalu	99680.38
8	Pitcairn	88888.89
9	Saint Helena	83628.23
10	Saint Martin (French part)	79417.26

CAROLINE 3/5/25 9:09AM

```
slice_max(covid_data,
new_cases_per_million, n = 10) %>%
 select(country, date, new_cases_per_million)
```

♡ 0 ◯ 0

```
# A tibble: 10 × 3
   country                  date       new_cases_per_million
   <chr>                    <date>                     <dbl>
 1 Wallis and Futuna        2022-10-14               230763.
 2 Falkland Islands         2022-05-13               208819.
 3 Saint Helena             2022-09-23               158130
 4 Falkland Islands         2022-05-24               137127.
 5 Nauru                    2022-06-25               104983.
 6 Saint Pierre and Miquelon 2022-03-22              100174.
 7 Tuvalu                   2022-11-15                99680.
 8 Pitcairn                 2022-07-20                88889.
 9 Saint Helena             2022-09-20                83628.
10 Saint Martin (French part) 2022-01-15              79417.
```

MORITZ 3/5/25 9:13AM

```
highest_case_country <-
slice_max(covid_data,
new_cases_per_million, n = 10) %>%
select(country, date, new_cases_per_million)
```

♡ 0 ◯ 0

	country	date	new_cases_per_million
1	Wallis and Futuna	2022-10-14	230762.55
2	Falkland Islands	2022-05-13	208819.34
3	Saint Helena	2022-09-23	158130.00
4	Falkland Islands	2022-05-24	137126.60
5	Nauru	2022-06-25	104982.66
6	Saint Pierre and Miquelon	2022-03-22	100173.61
7	Tuvalu	2022-11-15	99680.38
8	Pitcairn	2022-07-20	88888.89
9	Saint Helena	2022-09-20	83628.23
10	Saint Martin (French part)	2022-01-15	79417.26

BRAVE BARRACUDA 3/5/25 9:13AM

```
slice_max(covid_data,
new_cases_per_million, n = 10) %>%
```

select(country, new_cases_per_million)

♡ 0 ◯ 0

ATTENTIVE ELEPHANT 3/5/25 9:14AM

10 max

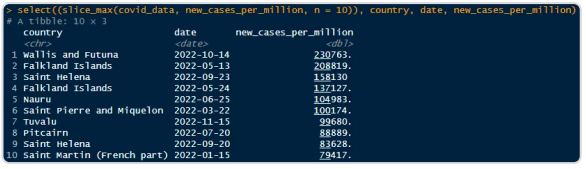
```
covid_data %>%
  select(country,new_cases_per_million)%>%
  slice_max(new_cases_per_million, n = 10)
```

♡ 0 ◯ 0

MYSTERIOUS OSTRICH 3/5/25 9:14AM

Filtering

```
select((slice_max(covid_data,
new_cases_per_million, n = 10)),
country, date, new_cases_per_million)
```

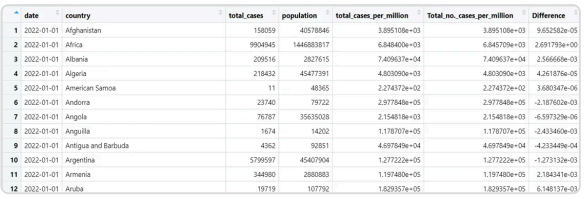


```
> select((slice_max(covid_data, new_cases_per_million, n = 10)), country, date, new_cases_per_million)
# A tibble: 10 × 3
   country                  date       new_cases_per_million
   <chr>                    <date>                     <dbl>
 1 Wallis and Futuna        2022-10-14               230763.
 2 Falkland Islands         2022-05-13               208819.
 3 Saint Helena             2022-09-23               158130
 4 Falkland Islands         2022-05-24               137127.
 5 Nauru                    2022-06-25               104983.
 6 Saint Pierre and Miquelon 2022-03-22              100174.
 7 Tuvalu                   2022-11-15                99680.
 8 Pitcairn                 2022-07-20                88889.
 9 Saint Helena             2022-09-20                83628.
10 Saint Martin (French part) 2022-01-15              79417.
```

♡ 0 ◯ 0

DEPENDABLE PORCUPINE 3/5/25 9:46AM

```
covid_data_small %>%
  mutate(Total_no._cases_per_million
= total_cases/(population/1000000),
         Difference = total_cases_per_million
- Total_no._cases_per_million) %>%
  view
```

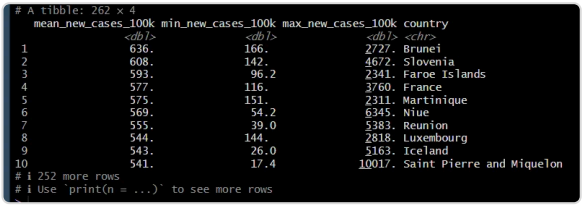


	date	country	total_cases	population	total_cases_per_million	Total_no._cases_per_million	Difference
1	2022-01-01	Afghanistan	158059	40578846	3.895108e+03	3.895108e+03	9.652582e-05
2	2022-01-01	Africa	9904945	1446883817	6.848400e+03	6.845709e+03	2.691793e+00
3	2022-01-01	Albania	209516	2827615	7.409637e+04	7.409637e+04	2.566668e-03
4	2022-01-01	Algeria	218432	45477391	4.803090e+03	4.803090e+03	4.261876e-05
5	2022-01-01	American Samoa	11	48365	2.274372e+02	2.274372e+02	3.680347e-06
6	2022-01-01	Andorra	23740	79722	2.977848e+05	2.977848e+05	-2.187602e-03
7	2022-01-01	Angola	76787	35635028	2.154818e+03	2.154818e+03	-6.597329e-06
8	2022-01-01	Anguilla	1674	14202	1.178707e+05	1.178707e+05	-2.433460e-03
9	2022-01-01	Antigua and Barbuda	4362	92851	4.697849e+04	4.697849e+04	-4.233449e-04
10	2022-01-01	Argentina	5799597	45407904	1.277222e+05	1.277222e+05	-1.273132e-03
11	2022-01-01	Armenia	344980	2880883	1.197480e+05	1.197480e+05	2.184341e-03
12	2022-01-01	Aruba	19719	107792	1.829357e+05	1.829357e+05	6.148137e-03

ATTENTIVE ELEPHANT 3/5/25 10:19AM

new cases 100k

```
covid_data %>%
  group_by(country)%>%
  slice_max(new_cases,n=100)%>%
  mutate(new_cases_100k =
new_cases_per_million/10)%>%
  reframe(min_new_cases_100k =
min(new_cases_100k),max_new_cases_100k
=
max(new_cases_100k),mean_new_cases_100
k = mean(new_cases_100k))%>%%

relocate(mean_new_cases_100k,min_new_ca
ses_100k,max_new_cases_100k,country)%>%
  arrange(desc(mean_new_cases_100k))
```

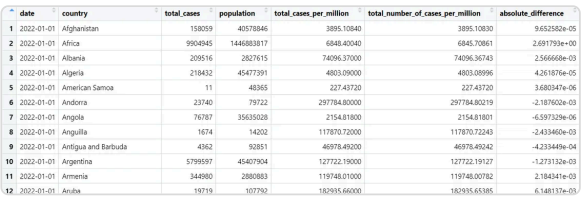


```
# A tibble: 262 × 4
   mean_new_cases_100k min_new_cases_100k max_new_cases_100k country
                 <dbl>              <dbl>              <dbl> <chr>
 1                636.              166.               2727. Brunei
 2                608.              142.               4672. Slovenia
 3                593.               96.2              2341. Faroe Islands
 4                577.              116.               3760. France
 5                575.              151.               2311. Martinique
 6                569.               54.2              6345. Niue
 7                555.               39.0              5383. Reunion
 8                544.              144.               2818. Luxembourg
 9                543.               26.0              5163. Iceland
10                541.               17.4             10017. Saint Pierre and Miquelon
# ℹ 252 more rows
# ℹ Use `print(n = ...)` to see more rows
```

♡ 0 ◯ 0

MORITZ 3/5/25 9:50AM

```
covid_data_small_new
<- covid_data_small %>%

mutate(total_number_of_cases_per_million
= total_cases/(population/1000000),
```



	date	country	total_cases	population	total_cases_per_million	total_number_of_cases_per_million	absolute_difference
1	2022-01-01	Afghanistan	158059	40578846	3895.10840	3895.10830	9.652582e-05
2	2022-01-01	Africa	9904945	1446883817	6848.40040	6845.70861	2.691793e+00
3	2022-01-01	Albania	209516	2827615	74096.37000	74096.36743	2.566668e-03
4	2022-01-01	Algeria	218432	45477391	4803.09000	4803.08996	4.261876e-05
5	2022-01-01	American Samoa	11	48365	227.43720	227.43720	3.680347e-06
6	2022-01-01	Andorra	23740	79722	297784.80000	297784.80219	-2.187602e-03
7	2022-01-01	Angola	76787	35635028	2154.81800	2154.81801	-6.597329e-06
8	2022-01-01	Anguilla	1674	14202	117870.72000	117870.72243	-2.433460e-03
9	2022-01-01	Antigua and Barbuda	4362	92851	46978.49200	46978.49242	-4.233449e-04
10	2022-01-01	Argentina	5799597	45407904	127722.19000	127722.19127	-1.273132e-03
11	2022-01-01	Armenia	344980	2880883	119748.01000	119748.00782	2.184341e-03
12	2022-01-01	Aruba	19719	107792	182935.66000	182935.65385	6.148137e-03

absolute_difference =
total_cases_per_million -
total_number_of_cases_per_million)

♡ 0 ◯ 0

SIMON 3/5/25 9:52AM

Create variables

```
transmute(covid_data_small,
    country = country,
    total_cases_1mio =
(total_cases/population) * 1000000,
    total_cases_diff = abs(total_cases_1mio
- total_cases_per_million)
)
```

♡ 0 ◯ 0

```
# A tibble: 258 × 3
   country              total_cases_1mio total_cases_diff
   <chr>                           <dbl>            <dbl>
 1 Afghanistan                     3895.       0.0000965
 2 Africa                          6846.       2.69
 3 Albania                        74096.       0.00257
 4 Algeria                         4803.       0.0000426
 5 American Samoa                   227.       0.00000368
 6 Andorra                        297785.       0.00219
 7 Angola                          2155.       0.00000660
 8 Anguilla                      117871.       0.00243
 9 Antigua and Barbuda            46978.       0.000423
10 Argentina                     127722.       0.00127
# ℹ 248 more rows
```

MYSTERIOUS OSTRICH 3/5/25 9:55AM

total_cases_per_million_calc + diff

```
mutate(covid_data_small,
       total_cases_per_million_calc
= (total_cases / population)*1000000,
       diff_total_cases_per_million
= total_cases_per_million -
total_cases_per_million_calc
)
```

♡ 0 ◯ 0

```
mutate(covid_data_small,
       total_cases_per_million_calc = (total_cases / population)*1000000,
       diff_total_cases_per_million = total_cases_per_million - total_cases_per_million_calc
)
# A tibble: 258 × 7
   date       country        total_cases population total_cases_per_mill.* total_cases_per_mill.* diff_total_cases_per_.*
   <date>     <chr>                <dbl>      <dbl>                  <dbl>                  <dbl>                  <dbl>
 1 2022-01-01 Afghanistan         158059   40578846                  3895.                  3895.             0.0000965
 2 2022-01-01 Africa             9904945 1446883817                  6848.                  6846.             2.69
 3 2022-01-01 Albania             209516    2827615                 74096.                 74096.             0.00257
 4 2022-01-01 Algeria             218432   45477391                  4803.                  4803.             0.0000426
 5 2022-01-01 American Samoa          11      48365                   227.                   227.             0.00000368
 6 2022-01-01 Andorra              23740      79722                297785.                297785.            -0.00219
 7 2022-01-01 Angola               76787   35635028                  2155.                  2155.            -0.00000660
 8 2022-01-01 Anguilla              1674      14202                117871.                117871.            -0.00243
 9 2022-01-01 Antigua and Ba…       4362      92851                 46978.                 46978.            -0.000423
10 2022-01-01 Argentina           5799597   45407904                127722.                127722.            -0.00127
```

ATTENTIVE ELEPHANT 3/5/25 9:55AM

Delta total pr. million

```
covid_data_small |>

select(country,total_cases,population,total_cas
es_per_million)|>

mutate(calc_total_number_cases_pr_million
= total_cases/(population/10e6))|>
 mutate(delta_calc_existing =
abs(total_cases_per_million-
calc_total_number_cases_pr_million))|>
 filter(delta_calc_existing > 1)|>
 arrange(desc(delta_calc_existing))
```

♡ 0 ◯ 0

```
# A tibble: 235 × 6
   country        total_cases population total_cases_per_million calc_total_number_cases_pr_million delta_calc_existing
   <chr>                <dbl>      <dbl>                   <dbl>                               <dbl>               <dbl>
 1 Andorra              23740      79722                  297785.                             0.298             297785.
 2 Montenegro          158010     614650                  257073.                             0.257             257073.
 3 Georgia             936844    3794783                  246877.                             0.247             246877.
 4 San Marino            8417      34113                  246739.                             0.247             246739.
 5 Czechia            2522617   10673216                  236350.                             0.236             236350.
 6 Gibraltar             8701      37631                  231219.                             0.231             231219.
 7 Slovenia            456842    2115230                  215977.                             0.216             215977.
 8 Mongolia            693546    3386015                  204827.                             0.205             204826.
 9 United Kingdom    13678283   68179315                  200622.                             0.201             200622.
10 Seychelles           24991     125532                  199081.                             0.199             199081.
# ℹ 225 more rows
# ℹ Use `print(n = ...)` to see more rows
```

BRAVE BARRACUDA 3/5/25 9:57AM

Pipe

```
covid_data_small %>%
 mutate(total_no_cases_per_million
= total_cases/(population/1000000),
       absolute_diff =
abs(total_no_cases_per_million-
total_cases_per_million)) %>%
 filter(absolute_diff > 1) %>%
```

```
# A tibble: 8 × 7
  date       country        total_cases population total_cases_per_mill…¹ total_no_cases_per_m…² absolute_diff
  <date>     <chr>                <dbl>      <dbl>                  <dbl>                  <dbl>         <dbl>
1 2022-01-01 Cyprus              165347    1331376                 175831.                124193.       51638.
2 2022-01-01 France             8709926   66277412                 135505.                131416.        4089.
3 2022-01-01 High-income co…  135342623 1252308750                 110907.                108074.        2832.
4 2022-01-01 World            2868273371 8021407170                  35910.                 35764.         146.
5 2022-01-01 Asia             847664677 4746329726                  17979.                 17859.         120.
6 2022-01-01 European Union…   53864957  449113453                 120041.                119936.         105.
7 2022-01-01 Europe           86962704  748236158                 116274.                116214.          60.8
8 2022-01-01 Africa            9904945 1446883817                   6848.                  6846.          2.69
# ℹ abbreviated names: ¹total_cases_per_million, ²total_no_cases_per_million
```

arrange(desc(absolute_diff))

♡ 0 ◯ 0

DEPENDABLE PORCUPINE 3/5/25 9:59AM

Pipe

```
covid_data_small %>%
 mutate(Total_no._cases_per_million
= total_cases/(population/1000000),
       Difference = total_cases_per_million
- Total_no._cases_per_million) %>%
 filter(Difference >1) %>%
 arrange(desc(Difference)) %>%
 view
```

	date	country	total_cases	population	total_cases_per_million	Total_no._cases_per_million	Difference
1	2022-01-01	Cyprus	165347	1331376	175830.73	124192.565	51638.165315
2	2022-01-01	France	8709926	66277412	135505.23	131416.206	4089.024431
3	2022-01-01	High-income countries	135342623	1252308750	110906.81	108074.485	2832.325173
4	2022-01-01	World	286877371	8021407170	35909.68	35763.971	145.713300
5	2022-01-01	Asia	84766467	4746329726	17979.03	17859.372	119.663255
6	2022-01-01	European Union (27)	53864957	449113453	120040.70	119936.191	104.503732
7	2022-01-01	Europe	86962204	748296158	116274.38	116213.618	60.756808
8	2022-01-01	Africa	9904945	1446883817	6848.40	6845.709	2.691793

♡ 0 ◯ 0

SIMON 3/5/25 10:01AM

Pipe

```
covid_data_small %>%
  mutate(
    total_cases_1mio =
(total_cases/population) * 1000000,
    total_cases_diff = abs(total_cases_1mio
- total_cases_per_million)) %>%
 filter(total_cases_diff > 1) %>%
 arrange(desc(total_cases_diff)) %>%
 select(country, date,
total_cases_1mio, total_cases_diff)
```

```
# A tibble: 8 × 4
  country               date       total_cases_1mio total_cases_diff
  <chr>                 <date>                <dbl>            <dbl>
1 Cyprus                2022-01-01          124193.          51638.
2 France                2022-01-01          131416.           4089.
3 High-income countries 2022-01-01          108074.           2832.
4 World                 2022-01-01           35764.            146.
5 Asia                  2022-01-01           17859.            120.
6 European Union (27)   2022-01-01          119936.            105.
7 Europe                2022-01-01          116214.             60.8
8 Africa                2022-01-01            6846.              2.69
```

♡ 0 ◯ 0

SIMON 3/5/25 10:20AM

Group and summarize

```
covid_data %>%
 group_by(continent, country) %>%
 slice_max(new_cases, n=100) %>%
 summarise(
   min_new_cases =
min(new_cases_per_million),
   max_new_cases =
max(new_cases_per_million),
   mean_new_cases =
mean(new_cases_per_million)
 ) %>%
 arrange(desc(mean_new_cases))
```

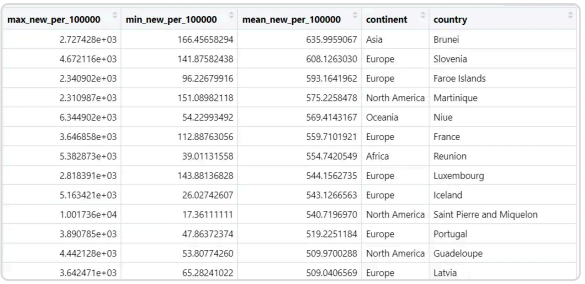
```
   continent     country                      min_new_cases max_new_cases mean_new_cases
   <chr>         <chr>                                <dbl>         <dbl>          <dbl>
 1 Asia          Brunei                               1665.         27274.         6360.
 2 Europe        Slovenia                             1419.         46721.         6081.
 3 Europe        Faroe Islands                         962.         23409.         5932.
 4 Europe        France                               1164.         37603.         5771.
 5 North America Martinique                           1511.         23110.         5752.
 6 Oceania       Niue                                  542.         63449.         5694.
 7 Africa        Reunion                               390.         53829.         5547.
 8 Europe        Luxembourg                           1439.         28184.         5442.
 9 Europe        Iceland                               260.         51634.         5431.
10 North America Saint Pierre and Miquelon             174.        100174.         5407.
# i 252 more rows
```

♡ 0 ◯ 0

DEPENDABLE PORCUPINE 3/5/25 10:20AM

Group and summarise

```
covid_data %>%
 group_by(continent, country) %>%
 slice_max(new_cases, n=100) %>%
 summarise(max_new_per_100000 =
max(new_cases/(population/100000)),
           min_new_per_100000 =
min(new_cases/(population/100000)),
           mean_new_per_100000 =
mean(new_cases/(population/100000))
           ) %>%
 relocate(1:2, .after=
mean_new_per_100000) %>%
 arrange(desc(mean_new_per_100000)) %>%
 view
```



max_new_per_100000	min_new_per_100000	mean_new_per_100000	continent	country
2.727428e+03	166.45658294	635.9959067	Asia	Brunei
4.672116e+03	141.87582438	608.1263030	Europe	Slovenia
2.340902e+03	96.22679916	593.1641962	Europe	Faroe Islands
2.310987e+03	151.08982118	575.2258478	North America	Martinique
6.344902e+03	54.22993492	569.4143167	Oceania	Niue
3.646858e+03	112.88763056	559.7101921	Europe	France
5.382873e+03	39.01131558	554.7420549	Africa	Reunion
2.818391e+03	143.88136828	544.1562735	Europe	Luxembourg
5.163421e+03	26.02742607	543.1266563	Europe	Iceland
1.001736e+04	17.36111111	540.7196970	North America	Saint Pierre and Miquelon
3.890785e+03	47.86372374	519.2251184	Europe	Portugal
4.442128e+03	53.80774260	509.9700288	North America	Guadeloupe
3.642471e+03	65.28241022	509.0406569	Europe	Latvia

♡ 0 ◯ 0

CYPRIEN 3/5/25 10:22AM

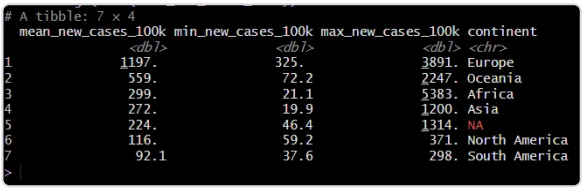
```
covid_data %>%
 group_by(continent, country) %>%
 slice_max(new_cases, n =
100, with_ties = FALSE) %>%
 mutate(new_cases_per_100 =
(new_cases / population) * 1e5) %>%
 summarise(
   min_new_cases_per_100 =
min(new_cases_per_100, na.rm = TRUE),
   max_new_cases_per_100 =
max(new_cases_per_100, na.rm = TRUE),
   mean_new_cases_per_100=
mean(new_cases_per_100,
na.rm = TRUE)) %>%
 arrange(desc(mean_new_cases_per_100))
```

♡ 0 ◯ 0

ATTENTIVE ELEPHANT 3/5/25 10:22AM

new cases by continent

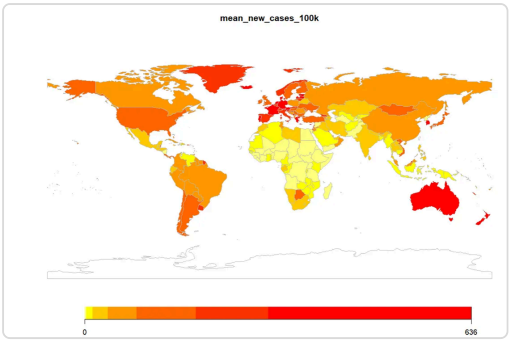
```
covid_data %>%
 group_by(continent)%>%
 slice_max(new_cases,n=100)%>%
 mutate(new_cases_100k =
new_cases_per_million/10)%>%
 reframe(min_new_cases_100k =
min(new_cases_100k),max_new_cases_100k
=
max(new_cases_100k),mean_new_cases_100
k = mean(new_cases_100k))%>%

relocate(mean_new_cases_100k,min_new_ca
ses_100k,max_new_cases_100k,continent)%>
%
 arrange(desc(mean_new_cases_100k))

mapped_data <-
joinCountryData2Map(map_data, joinCode
= "NAME", nameJoinColumn = "country")
```



```
# A tibble: 7 × 4
  mean_new_cases_100k min_new_cases_100k max_new_cases_100k continent
                <dbl>              <dbl>              <dbl> <chr>
1              1197.              325.               3891. Europe
2               559.               72.2              2247. Oceania
3               299.               21.1              5383. Africa
4               272.               19.9              1200. Asia
5               224.               46.4              1314. NA
6               116.               59.2               371. North America
7                92.1              37.6               298. South America
>
```

```
mapCountryData(mapped_data,
nameColumnToPlot =
"mean_new_cases_100k")
```

♡ 0 ◯ 1

> **Attentive Elephant** 3/5/25 10:32AM
>
> 
>

BRAVE BARRACUDA 3/5/25 10:22AM

Group and summaries

```
covid_data %>%
 group_by(continent,country) %>%
 slice_max(new_cases, n = 100) %>%
 summarise(
  Min_pr_1e5 =
min(new_cases/population *1e5),
  Max_pr_1e5 =
max(new_cases/population *1e5),
  Mean_pr_1e5 =
mean(new_cases/population *1e5)
 ) %>%

select(Min_pr_1e5,Max_pr_1e5,Mean_pr_1e5,
everything()) %>%
 arrange(desc(Mean_pr_1e5))
```

	Min_pr_1e5	Max_pr_1e5	Mean_pr_1e5	continent	country
	<dbl>	<dbl>	<dbl>	<chr>	<chr>
1	166.	2727.	636.	Asia	Brunei
2	142.	4672.	608.	Europe	Slovenia
3	96.2	2341.	593.	Europe	Faroe Islands
4	151.	2311.	575.	North America	Martinique
5	54.2	6345.	569.	Oceania	Niue
6	113.	3647.	560.	Europe	France
7	39.0	5383.	555.	Africa	Reunion
8	144.	2818.	544.	Europe	Luxembourg
9	26.0	5163.	543.	Europe	Iceland
10	17.4	10017.	541.	North America	Saint Pierre and Miquelon

♡ 0 ◯ 0

RASMUS 3/5/25 10:25AM

Group and summarise

```
covid_data %>%
 group_by(continent, country) %>%
 slice_max(new_cases, n = 100) %>%
 summarise(
  min_new_cases =
min(new_cases_per_million),
  max_new_cases =
max(new_cases_per_million),
  mean_new_cases =
mean(new_cases_per_million)
 ) %>%
 arrange(mean_new_cases)
```

♡ 0 ◯ 0

MORITZ 3/5/25 11:44AM

```
test_pos_rate <- covid_data %>%
 mutate(year = year(date),
        month = month(date)) %>%
 group_by(year, month, country) %>%
 summarize(avg_positive_rate =
mean(positive_rate, na.rm = TRUE)) %>%
 ungroup() %>%
 arrange(desc(avg_positive_rate))

test_pos_rate %>%
 filter(country == "Denmark") %>%
 arrange(desc(avg_positive_rate))

# -> February 2022

test_pos_rate_5_months
<- test_pos_rate %>%
 arrange(desc(avg_positive_rate)) %>%
 slice_head(n=5)
```

♡ 0 ◯ 0

	year	month	country	avg_positive_rate
1	2022	6	Puerto Rico	93.30830
2	2022	4	Laos	92.92929
3	2022	4	Maldives	88.11380
4	2022	3	Vietnam	87.25661
5	2020	8	United States Virgin Islands	87.22943

ATTENTIVE ELEPHANT 3/5/25 11:33AM

Last exercise: calculate test positivity rate, averaged per month for each year and country,

```
subset_average_positive_rate<-
covid_data |>
select(country, date, positive_rate)
|> separate(date, into= c("year", "month", "day"))
|>
group_by(country, year, month)%>% drop_na()
|> summarise(averaged_positivity_rate
= mean(positive_rate,na.rm=TRUE ))
```

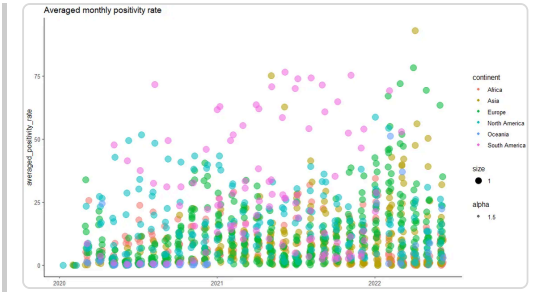
♡ 0 ◯ 3

```
> subset_average_positive_rate
# A tibble: 3,513 × 4
# Groups:   country, year [435]
   country     year  month averaged_positivity_rate
   <chr>       <chr> <chr>                    <dbl>
 1 Afghanistan 2022  02                        36.4
 2 Afghanistan 2022  03                        14.6
 3 Afghanistan 2022  04                         4.03
 4 Afghanistan 2022  05                         3.96
 5 Afghanistan 2022  06                        16.1
 6 Albania     2020  03                         7.72
 7 Albania     2020  04                        10.5
 8 Albania     2020  05                         5.32
 9 Albania     2020  06                        11.8
10 Albania     2020  07                        19.5
# i 3,503 more rows
# i Use `print(n = ...)` to see more rows
>
```

> **Attentive Elephant** 3/5/25 11:34AM
>
> ```
> # A tibble: 1 × 4
> # Groups:   country, year [1]
>   country year  month averaged_positivity_rate
>   <chr>   <chr> <chr>                    <dbl>
> 1 Denmark 2022  02                        30.3
> >
> ```
>
> # Which month had the highest positivity rate in Denmark?
> subset_average_positive_rate|>
> filter(country == "Denmark")|>
> arrange(desc(averaged_positivity_rate))%>%
> head(1)

> **Attentive Elephant** 3/5/25 11:54AM
>
> ```
> # A tibble: 5 × 4
> # Groups:   country, year [5]
>   country                     year  month averaged_positivity_rate
>   <chr>                       <chr> <chr>                    <dbl>
> 1 Puerto Rico                 2022  06                        93.3
> 2 Laos                        2022  04                        92.9
> 3 Maldives                    2022  04                        88.1
> 4 Vietnam                     2022  03                        87.3
> 5 United States Virgin Islands 2020 08                        87.2
> ```
>
> Which countries were included in the five months with the overall highest positivity rates?
> subset_average_positive_rate%>%
> arrange(desc(averaged_positivity_rate))%>%
> head(5)

> **Attentive Elephant** 3/5/25 11:54AM



DEPENDABLE PORCUPINE 3/5/25 11:36 AM

Exercise

```
pos_rate_overview = covid_data %>%
  select(country,date, positive_rate) %>%
  separate(date, sep ="-", into=
c("Year", "Month", "day")) %>%
  group_by(country, Year, Month) %>%
  summarise(mean_pos_rate_month
= mean(positive_rate)) %>%
  view
# Examining the resulting table,
answer the following questions:
# Which month had the highest
positivity rate in Denmark?
pos_rate_overview %>%
  filter(country=="Denmark")%>%
  arrange(desc(mean_pos_rate_month))
# February 2022
# Which countries were
included in the five months with
# the overall highest positivity rates?
pos_rate_overview %>%
  group_by(mean_pos_rate_month) %>%
  arrange(desc(mean_pos_rate_month)) %>%
  view()
#Netherlands, Mongolia, Puerto
Rico, (Netherlands), Germany
```

	country	Year	Month	mean_pos_rate_month
1	Netherlands	2022	04	78.32492
2	Mongolia	2021	10	78.00149
3	Puerto Rico	2022	05	73.38324
4	Netherlands	2022	03	72.03657
5	Germany	2022	04	71.46957

♡ 0 ◯ 0