WILDSI Use Cases

Blaise Alako

2020-07-17

-

```
## Warning: `as.tibble()` is deprecated as of tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
## Warning in postgresqlExecStatement(conn, statement, ...): RS-DBI driver warning:
## (unrecognized PostgreSQL field type point (id:600) in column 1)
```

```
system.time(pmc_ena <- dbGetQuery(conn, pmc_ena))
```

```
##    user  system elapsed
##  31.289  16.224 539.709
```

```
column_name <- c('_id','accession','idpmc','source' ,'pubtype','issn','isopenaccess','secondary_pmid',
                   'first_pub_date','first_epub_date', 'author_orcid', 'language','grantid','grant_age
                 'ena_accession','primary_pmid','primary_doi', 'primary_pmcid',
                 'seq_origin','seq_country','submission_date', 'first_created', 'seq_lat_lon','organism
# Rename pmc_ena column header
colnames(pmc_ena) <- column_name
pmc_ena <- tbl_df(pmc_ena)
```

```
## Warning: `tbl_df()` is deprecated as of dplyr 1.0.0.
## Please use `tibble::as_tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

-

```
#load('seqref.Rdata')
```

Use Case #1:

For each country in the world, please collect the following data:

-

```
# A function that take a country name generates the above
self <- function (countryName='Cameroon'){
  pmc_ena %>% filter( author_country == countryName & seq_country == countryName) %>%
    select(author_country, seq_country, ena_accession, author, secondary_pmid) %>%
    unique() %>% group_by(author_country, seq_country) %>%
    mutate(nseq=length(unique(ena_accession)), nscientist=length(unique(author)),  npublication=length(u
    ungroup() %>% select (author_country, seq_country, nseq, nscientist, npublication) %>% unique()
}
# Loop through all country name
system.time(selfUse <- mclapply(countries$name, function(x) self(countryName=x)))
```

```
##    user  system elapsed
##   6.073   0.517   8.231
```

```
# Merge the results
selfUseOnly<- do.call(rbind, selfUse)

selfUseOnly <- selfUseOnly %>% mutate(seqByScientist=nseq/nscientist, pubByScientist=npublication/nscier
selfUseOnly
```

```
## # A tibble: 128 x 7
##    author_country seq_country  nseq nscientist npublication seqByScientist
##    <chr>          <chr>       <int>      <int>        <int>          <dbl>
##  1 United Arab E~ United Ara~    17         78           13          0.218
##  2 Albania        Albania         1          3            1          0.333
##  3 Armenia        Armenia         1          4            3          0.25
##  4 Argentina      Argentina     114        424           77          0.269
##  5 Austria        Austria       401        403          118          0.995
##  6 Australia      Australia     913       1745          423          0.523
##  7 Bosnia and He~ Bosnia and~     1          9            2          0.111
##  8 Bangladesh     Bangladesh     36        118           23          0.305
##  9 Belgium        Belgium       118        397           75          0.297
## 10 Burkina Faso   Burkina Fa~     4         22            3          0.182
## # ... with 118 more rows, and 1 more variable: pubByScientist <dbl>
```

-

```
#A function that take a country name generates the above
target <- function (countryName='Cameroon'){
  pmc_ena %>% filter(seq_country==countryName & !author_country == countryName ) %>%
    select(author_country, seq_country, ena_accession, author, secondary_pmid) %>%
    unique()  %>% group_by(author_country,seq_country) %>%
```

```
    mutate(nscientist=length(unique(author)), nseq=length(unique(ena_accession)), npublication=length(u
    ungroup() %>% select(seq_country, author_country, nscientist, nseq, npublication) %>% unique()
}

# Loop through all country name
system.time(targetByCountry <- mclapply(countries$name, function(x) target(countryName=x)))
```

```
##    user  system elapsed
##  10.130   1.115  11.930
```

```
# Merge the results
targetByCountry <- do.call(rbind, targetByCountry)
```

```
targetByCountry <- targetByCountry %>% mutate(seqByScientist=nseq/nscientist, pubByScientist=npublicati
targetByCountry
```

```
## # A tibble: 6,791 x 7
##    seq_country author_country nscientist  nseq npublication seqByScientist
##    <chr>       <chr>               <int> <int>        <int>          <dbl>
##  1 Andorra     Italy                  41     2            6         0.0488
##  2 Andorra     China                  21     1            3         0.0476
##  3 Andorra     Mongolia                7     1            1         0.143
##  4 Andorra     United Kingdom         11     1            1         0.0909
##  5 Andorra     Mali                    4     1            1         0.25
##  6 United Ara~ France                 39     7            6         0.179
##  7 United Ara~ United States         117    18           11         0.154
##  8 United Ara~ Japan                  24    14            3         0.583
##  9 United Ara~ Malaysia               10     1            1         0.1
## 10 United Ara~ China                 265    23           33         0.0868
## # ... with 6,781 more rows, and 1 more variable: pubByScientist <dbl>
```

-

```
worldUse <- function (countryName='Cameroon'){
  pmc_ena %>% filter(!seq_country==countryName & author_country == countryName ) %>%
    select(author_country, seq_country, ena_accession, author, secondary_pmid) %>%
    unique()  %>% group_by(author_country,seq_country) %>%
    mutate(nscientist=length(unique(author)), nseq=length(unique(ena_accession)), npublication=length(u
    ungroup() %>% select(seq_country, author_country, nscientist, nseq, npublication) %>% unique()
}

# Loop through all country name
system.time(worldUseByCountry <- mclapply(countries$name, function(x) worldUse(countryName=x)))
```

```
##    user  system elapsed
##  11.658   1.037  12.886
```

```r
# Merge the results
worldUseByCountry <- do.call(rbind, worldUseByCountry)
worldUseByCountry
```

```
## # A tibble: 6,791 x 5
##    seq_country    author_country       nscientist  nseq npublication
##    <chr>          <chr>                     <int> <int>        <int>
##  1 United States  United Arab Emirates         17    46            4
##  2 United Kingdom United Arab Emirates         20     5            2
##  3 Japan          United Arab Emirates         12     2            2
##  4 Italy          United Arab Emirates         25     6            4
##  5 Hungary        United Arab Emirates          3     7            1
##  6 India          United Arab Emirates         18     3            3
##  7 Netherlands    United Arab Emirates         11     3            2
##  8 Hong Kong      United Arab Emirates          3     4            1
##  9 China          United Arab Emirates         13     9            3
## 10 Spain          United Arab Emirates          3     6            1
## # ... with 6,781 more rows
```

-

```r
worldUseOnly <- worldUseByCountry %>% group_by(author_country) %>%
  mutate(nscientist=sum(nscientist), nseq=sum(nseq), npublication=sum(npublication)) %>%
  ungroup() %>% select(author_country, nscientist, nseq, npublication) %>% unique() %>% mutate(seqByScie
worldUseOnly
```

```
## # A tibble: 157 x 6
##    author_country    nscientist  nseq npublication seqByScientist pubByScientist
##    <chr>                  <int> <int>        <int>          <dbl>          <dbl>
##  1 United Arab Emir~        287   138           51         0.481          0.178
##  2 Afghanistan               21     2            2         0.0952         0.0952
##  3 Anguilla                   6     2            1         0.333          0.167
##  4 Albania                    5     1            1         0.2            0.2
##  5 Armenia                    6     1            1         0.167          0.167
##  6 Angola                    24     3            3         0.125          0.125
##  7 Argentina               1676   400          245         0.239          0.146
##  8 Austria                 4582  3189         1036         0.696          0.226
##  9 Australia              11865  2747         1743         0.232          0.147
## 10 Azerbaijan                 5     2            1         0.4            0.2
## # ... with 147 more rows
```

```r
# Retrieve world map and rename UK and USA accordingly

world_map <- map_data("world")
world_map <- world_map %>%
  mutate(region=ifelse (region=="UK", "United Kingdom", region)) %>%
  mutate(region=ifelse (region=="USA", "United States", region))
```

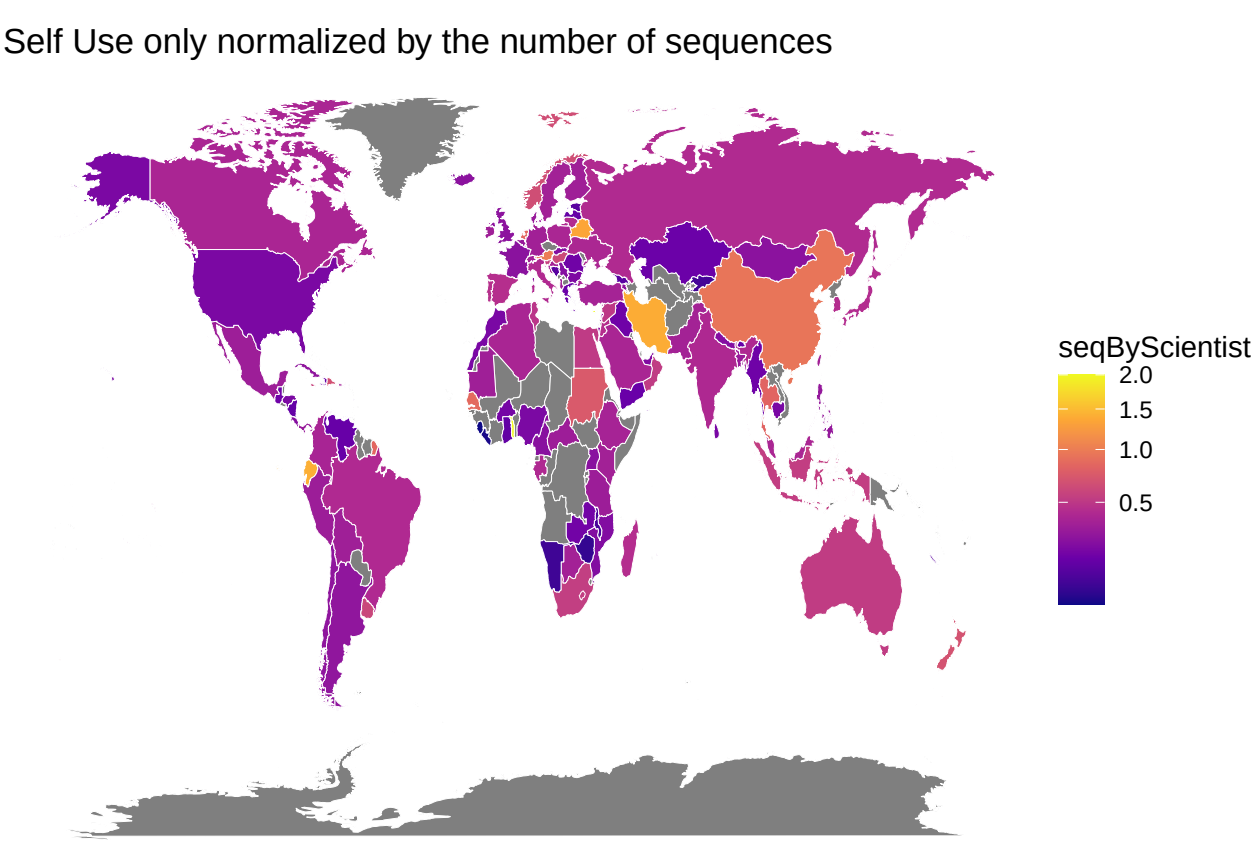
Please create a table for each country in the world with these 3 data types. Next, please display 5 world maps as follows:

-

```
# Join geolocation info with Self only use
selfUseOnly_map <- left_join(world_map, selfUseOnly,  by = c("region"="author_country"))
```

-

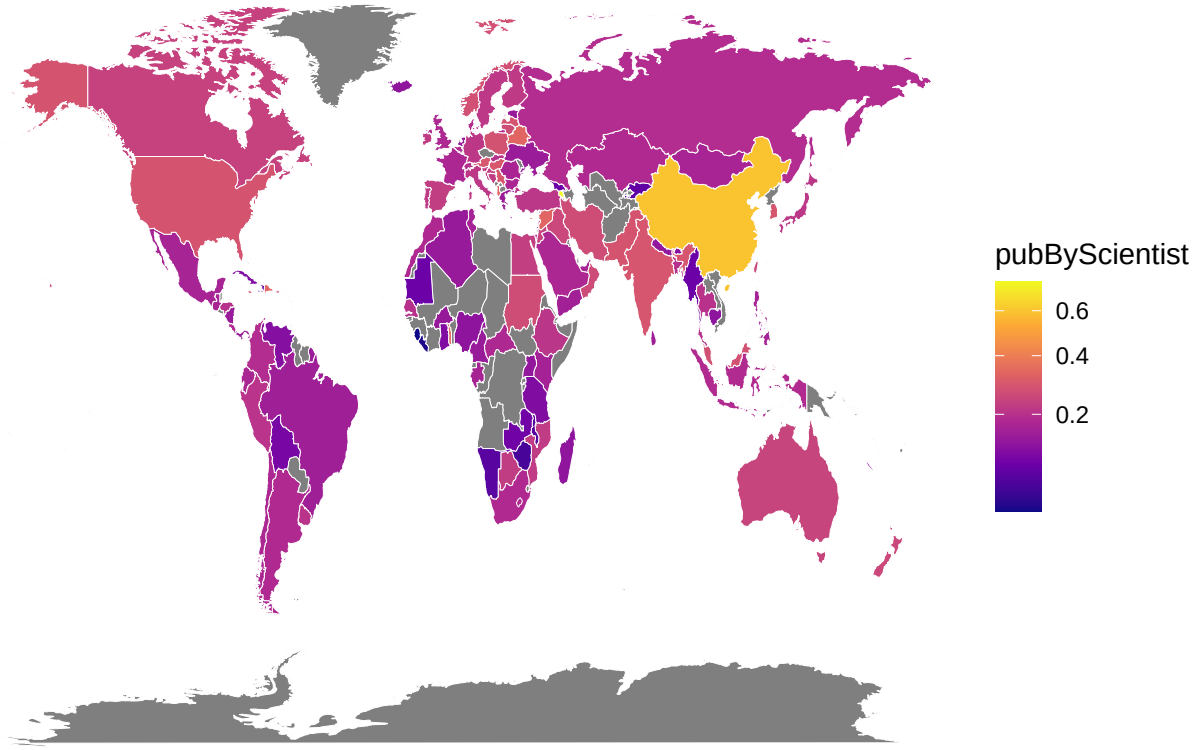
```
ggplot(selfUseOnly_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = seqByScientist), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') + ggtitle('Self Use only normalized by the number of
```



-

```
ggplot(selfUseOnly_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = pubByScientist), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') + ggtitle('Self Use only normalized by the number of
```

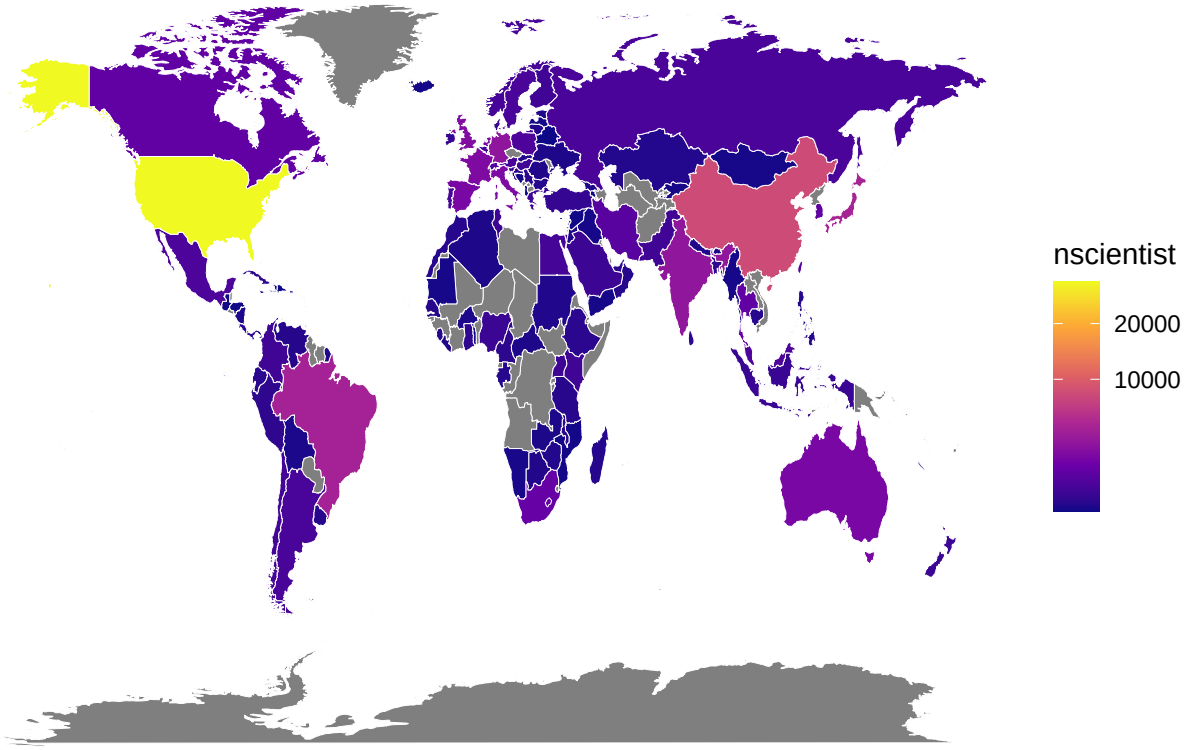
Self Use only normalized by the number of publication



-

```
ggplot(selfUseOnly_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = nscientist), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') + ggtitle('Self Use only Absolute scientists count')
```

Self Use only Absolute scientists count



-

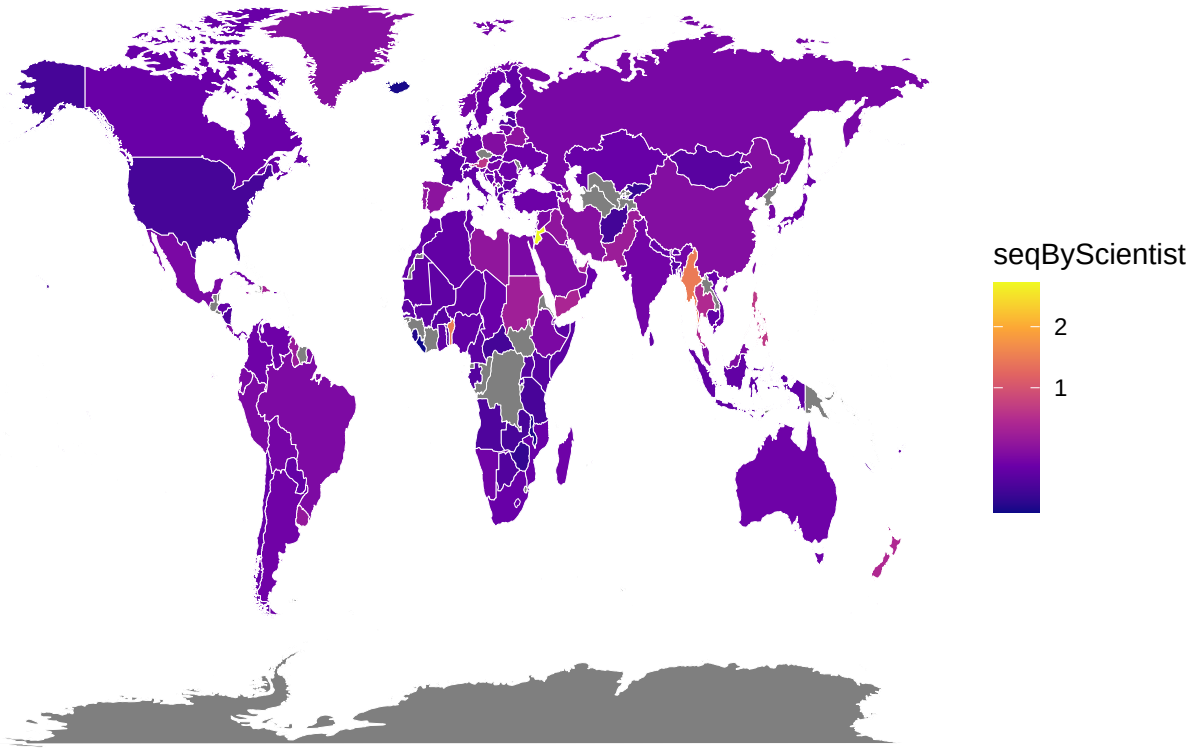
```
worldUseOnly_map <- tbl_df(left_join(world_map, worldUseOnly , by = c("region"="author_country")))
worldUseOnly_map
```

```
## # A tibble: 99,338 x 11
##     long   lat group order region subregion nscientist  nseq npublication
##    <dbl> <dbl> <dbl> <int> <chr>  <chr>          <int> <int>         <int>
##  1 -69.9  12.5     1     1 Aruba  <NA>              NA    NA            NA
##  2 -69.9  12.4     1     2 Aruba  <NA>              NA    NA            NA
##  3 -69.9  12.4     1     3 Aruba  <NA>              NA    NA            NA
##  4 -70.0  12.5     1     4 Aruba  <NA>              NA    NA            NA
##  5 -70.1  12.5     1     5 Aruba  <NA>              NA    NA            NA
##  6 -70.1  12.6     1     6 Aruba  <NA>              NA    NA            NA
##  7 -70.0  12.6     1     7 Aruba  <NA>              NA    NA            NA
##  8 -70.0  12.6     1     8 Aruba  <NA>              NA    NA            NA
##  9 -69.9  12.5     1     9 Aruba  <NA>              NA    NA            NA
## 10 -69.9  12.5     1    10 Aruba  <NA>              NA    NA            NA
## # ... with 99,328 more rows, and 2 more variables: seqByScientist <dbl>,
## #   pubByScientist <dbl>
```

-

```
ggplot(worldUseOnly_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = seqByScientist), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') + ggtitle('World Use only normalized by number of se
```

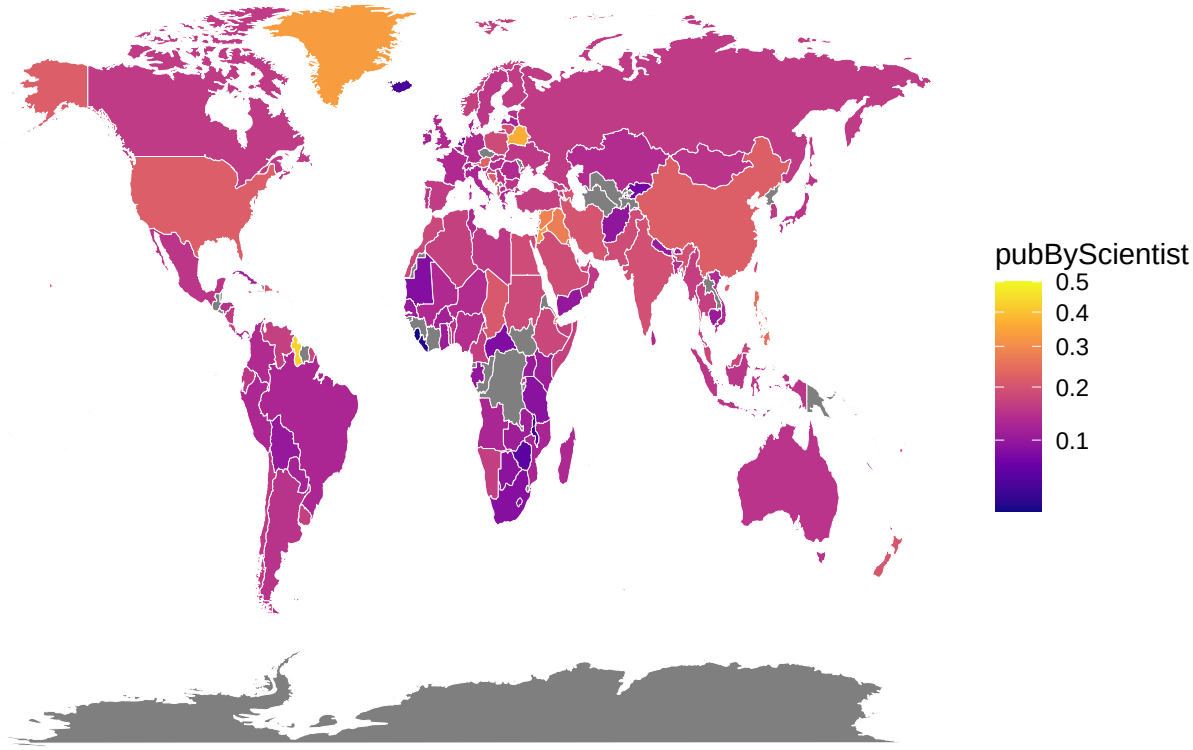
World Use only normalized by number of sequences



-

```
ggplot(worldUseOnly_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = pubByScientist), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') + ggtitle('World Use only normalized by publication'
```

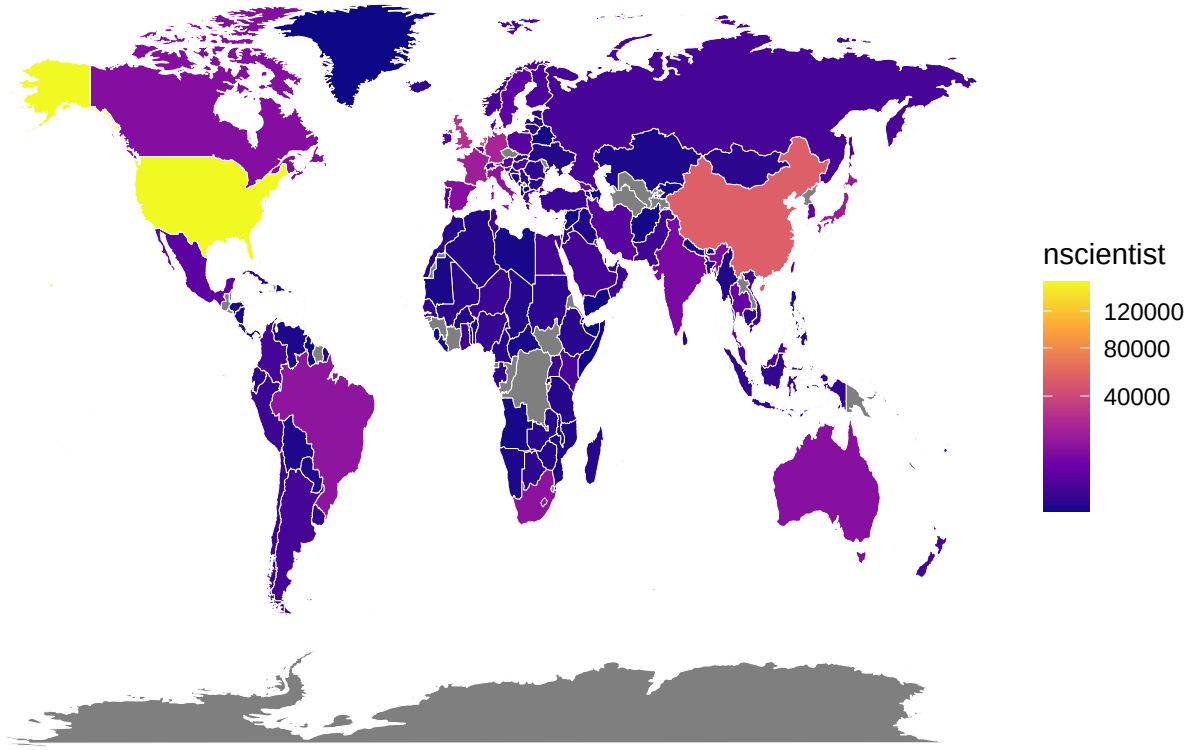
World Use only normalized by publication



-

```
ggplot(worldUseOnly_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = nscientist), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') + ggtitle('World Use only absolute scientists count')
```

World Use only absolute scientists count



-

```
targetUseOnly <- targetByCountry %>% group_by(seq_country) %>%
  mutate(nscientist=sum(nscientist), nseq=sum(nseq), npublication=sum(npublication)) %>%
  ungroup() %>% select(seq_country, nscientist, nseq, npublication) %>% unique() %>%
  mutate(seqByScientist=nseq/nscientist, pubByScientist=npublication/nscientist)
targetUseOnly
```

```
## # A tibble: 212 x 6
##    seq_country       nscientist  nseq npublication seqByScientist pubByScientist
##    <chr>                  <int> <int>        <int>          <dbl>          <dbl>
##  1 Andorra                   84     6           12         0.0714         0.143
##  2 United Arab Emir~       1066   146          129         0.137          0.121
##  3 Afghanistan              737    91          118         0.123          0.160
##  4 Antigua and Barb~         24     5            5         0.208          0.208
##  5 Albania                  149    19           22         0.128          0.148
##  6 Armenia                  147    23           25         0.156          0.170
##  7 Angola                   738    85           73         0.115          0.0989
##  8 Antarctica              2219   331          334         0.149          0.151
##  9 Argentina               5533   825          984         0.149          0.178
## 10 American Samoa           170     9            9         0.0529         0.0529
## # ... with 202 more rows
```

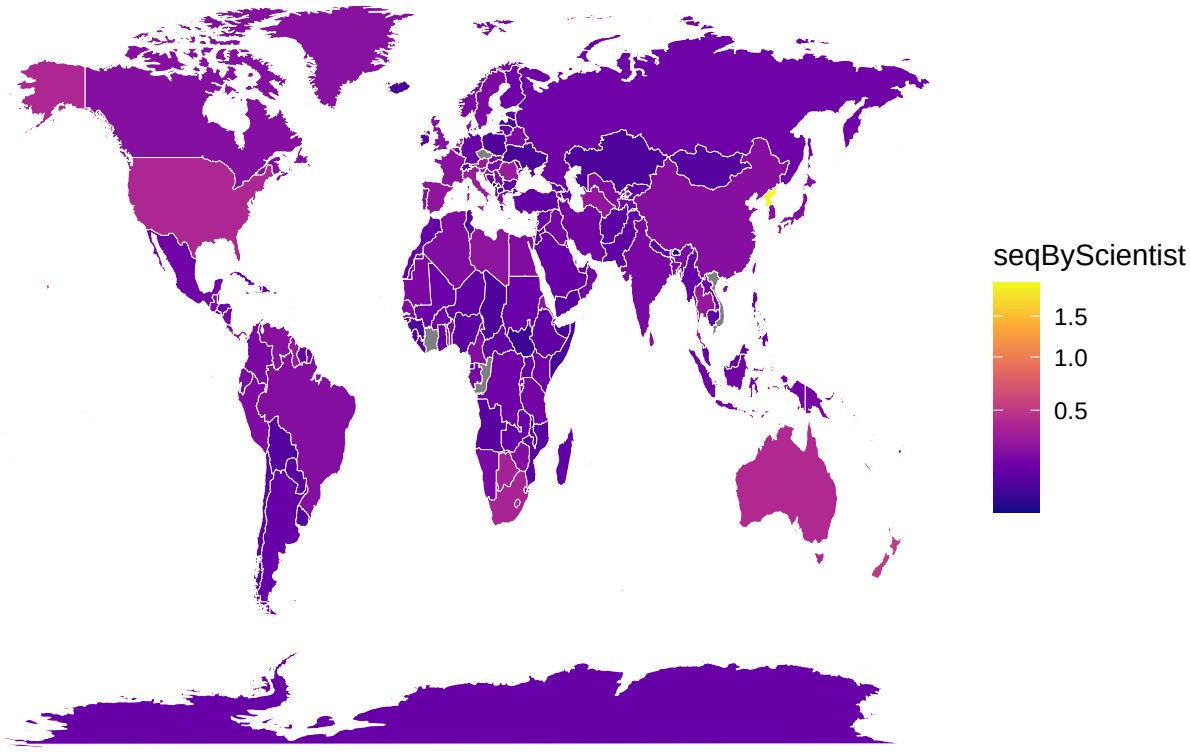
-

```
targetUseOnly_map <- tbl_df(left_join(world_map, targetUseOnly, by = c("region"="seq_country")))
```

-

```
ggplot(targetUseOnly_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = seqByScientist), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans = "sqrt") + ggtitle('Target Use normalized by number of sequ
```

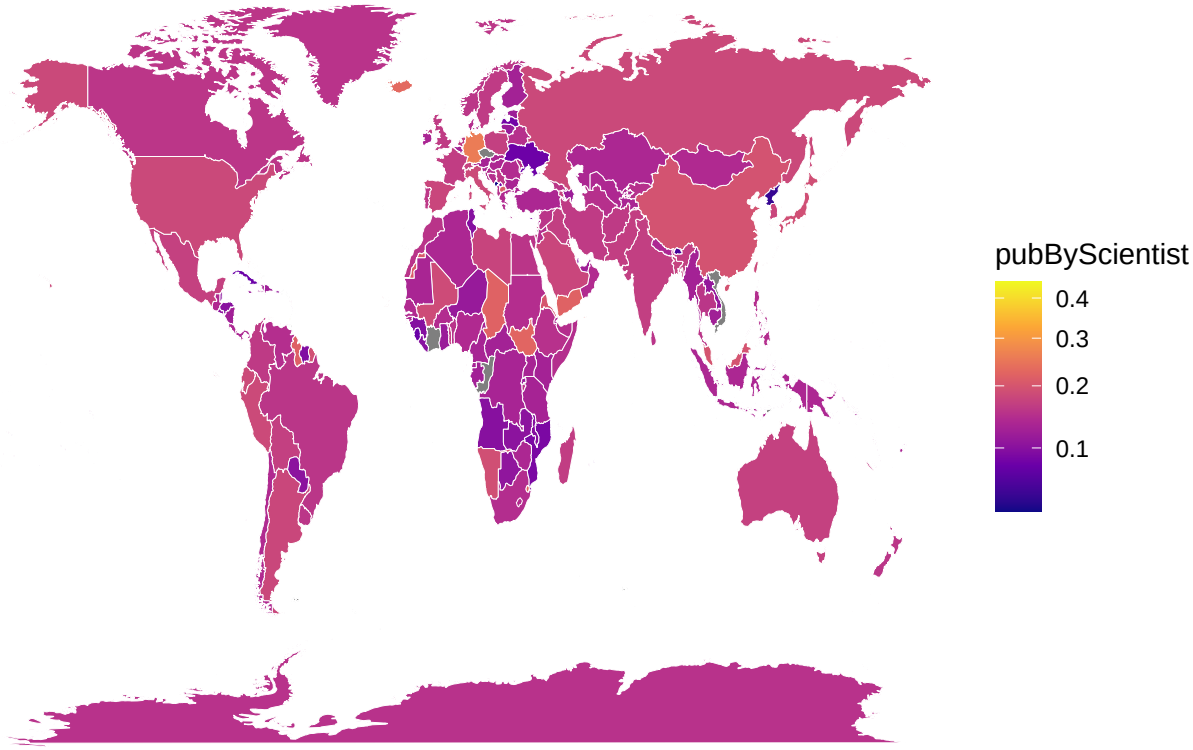
Target Use normalized by number of sequences



-

```
ggplot(targetUseOnly_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = pubByScientist), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans = "sqrt") + ggtitle('Target Use normalized by number of publ
```

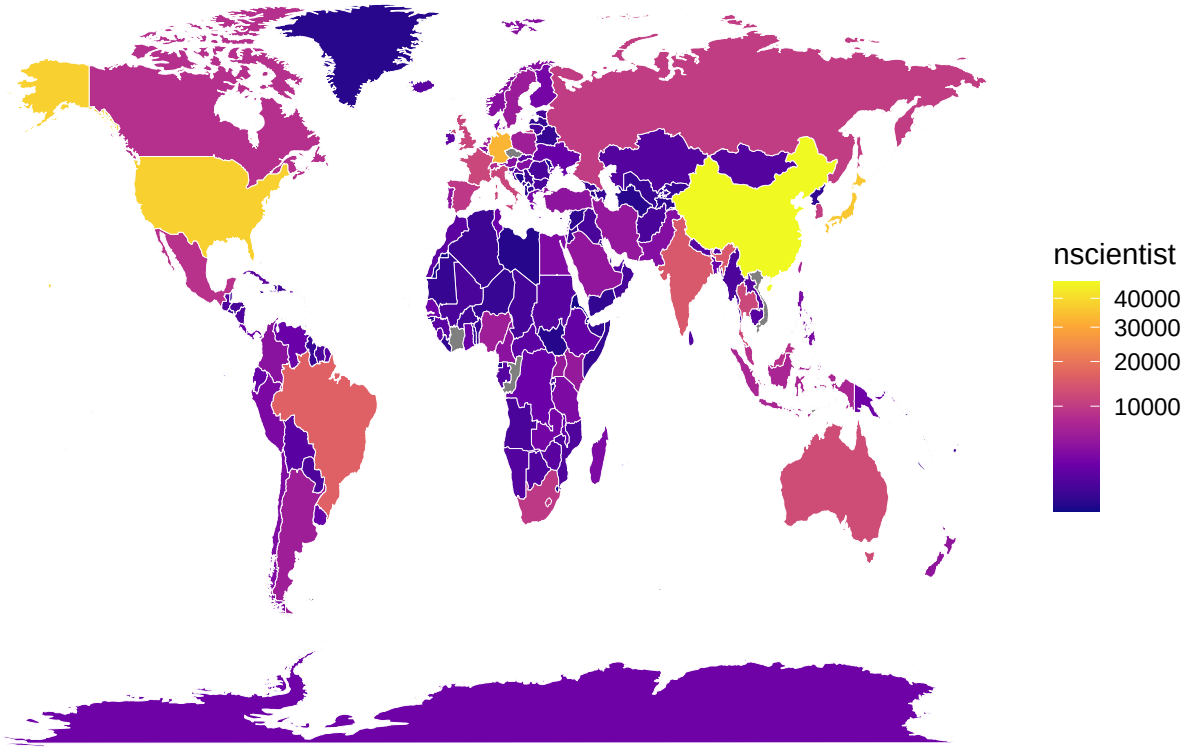
Target Use normalized by number of publication



-

```
ggplot(targetUseOnly_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = nscientist), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans = "sqrt") + ggtitle('Target Use , Absolute scientists count')
```

Target Use , Absolute scientists count

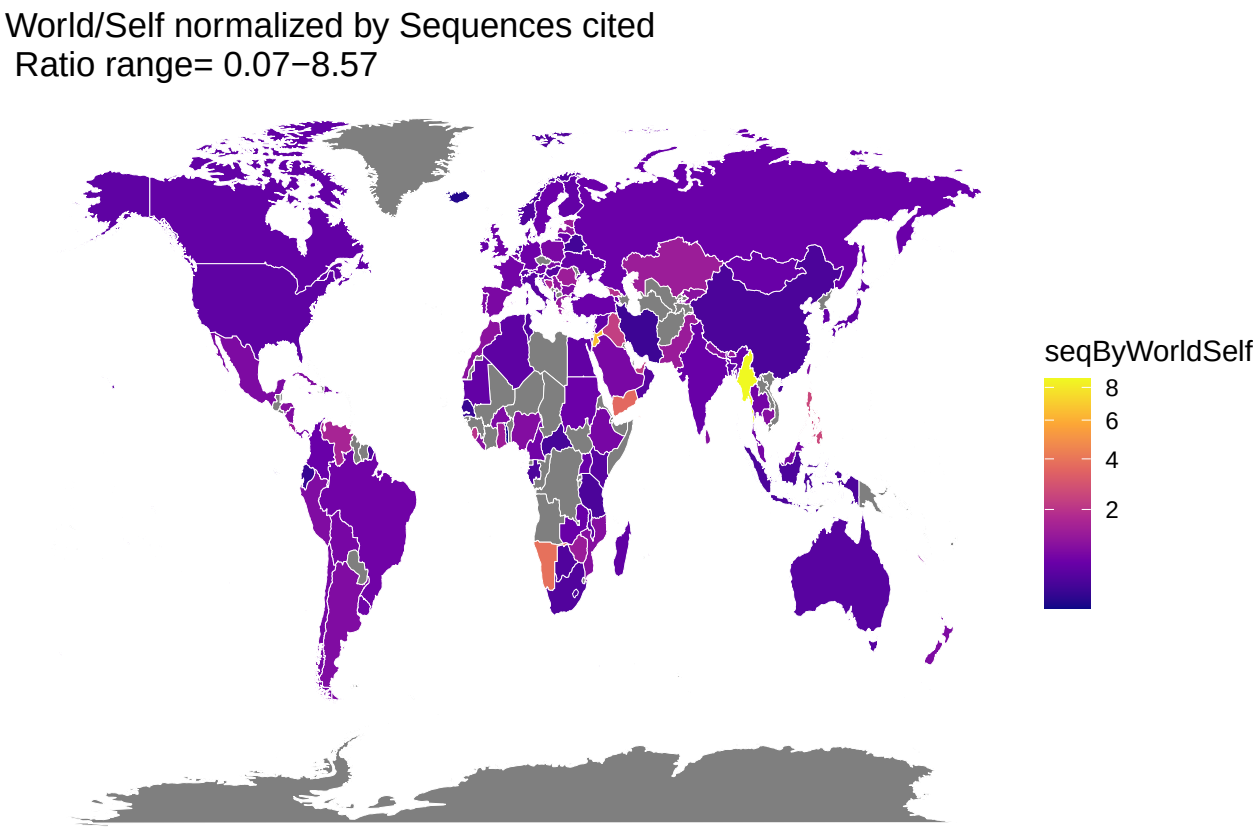


-

```
norm_world_self <- left_join(selfUseOnly,  worldUseOnly, by=c("author_country"="author_country")) %>%
  rename(selfScientist=nscientist.x, worldScientist=nscientist.y, selfNseq=nseq.x, worldNseq=nseq.y, sel
  mutate(seqByWorldSelf=(worldNseq/worldScientist)/(selfNseq/selfScientist), pubByWorldSelf=(worldNpub/
  select (author_country,worldNseq,worldNpub, worldScientist, selfNseq, selfNpub, selfScientist, seqByW
# Join geolocation info with world/Self
worldSelf_map <- tbl_df(left_join(world_map, norm_world_self, by = c("region"="author_country")))
```

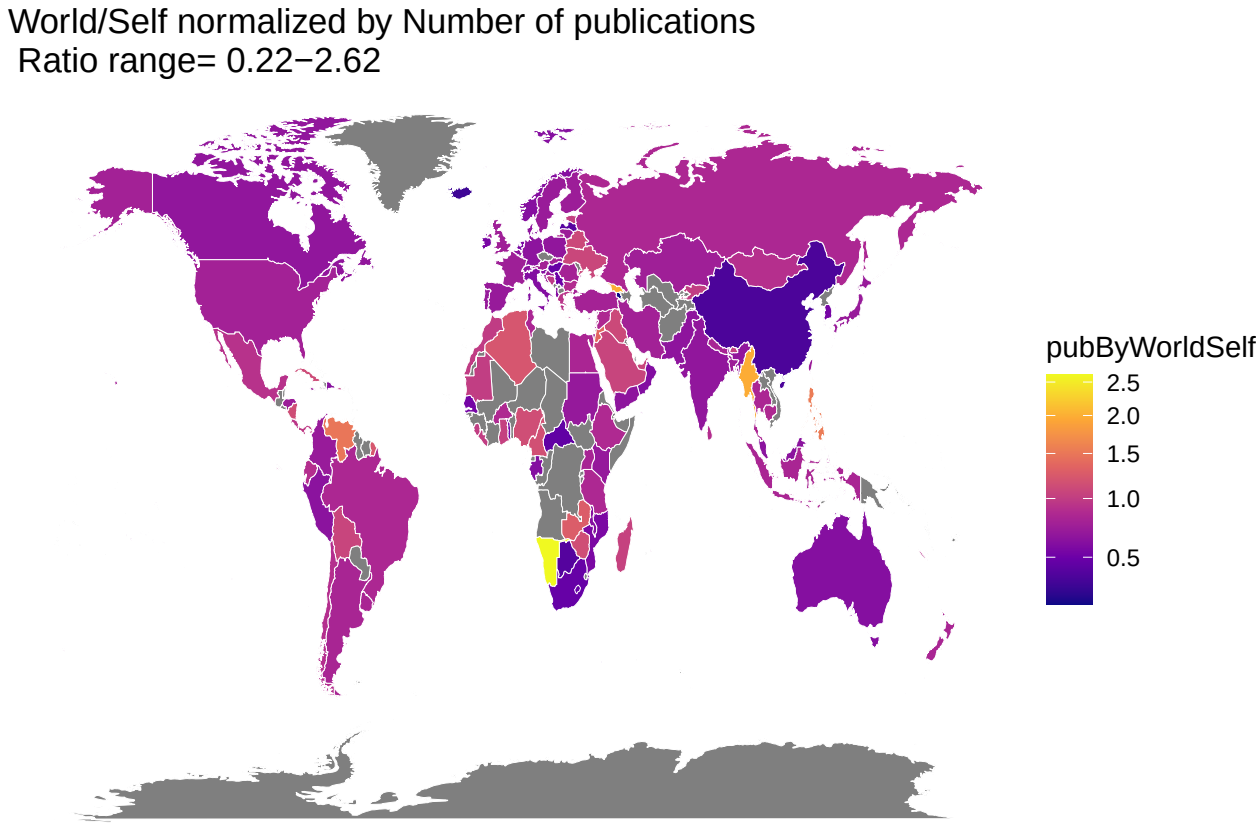
-

```
ggplot(worldSelf_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = seqByWorldSelf), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') +
  ggtitle(paste('World/Self normalized by Sequences cited \n Ratio range=', paste(round(range(na.omit(w
  theme_void()
```



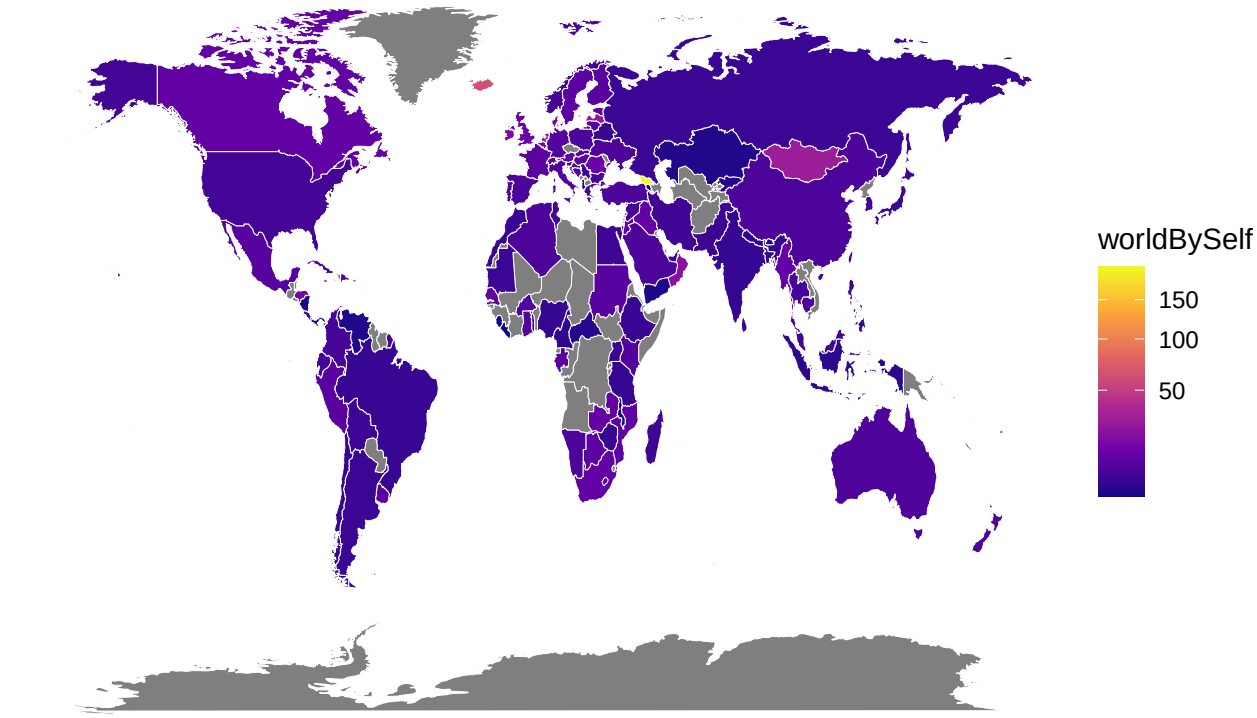
- #### Create the map Normalized by number of publications

```
ggplot(worldSelf_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = pubByWorldSelf), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') +
  ggtitle(paste('World/Self normalized by Number of publications \n Ratio range=', paste(round(range(na
  theme_void()
```



- #### Create the map absolute ratio

```
ggplot(worldSelf_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = worldBySelf), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') +
  ggtitle(paste('World/Self absolute ratio \n Ratio range=', paste(round(range(na.omit(worldSelf_map$wo
  theme_void()
```

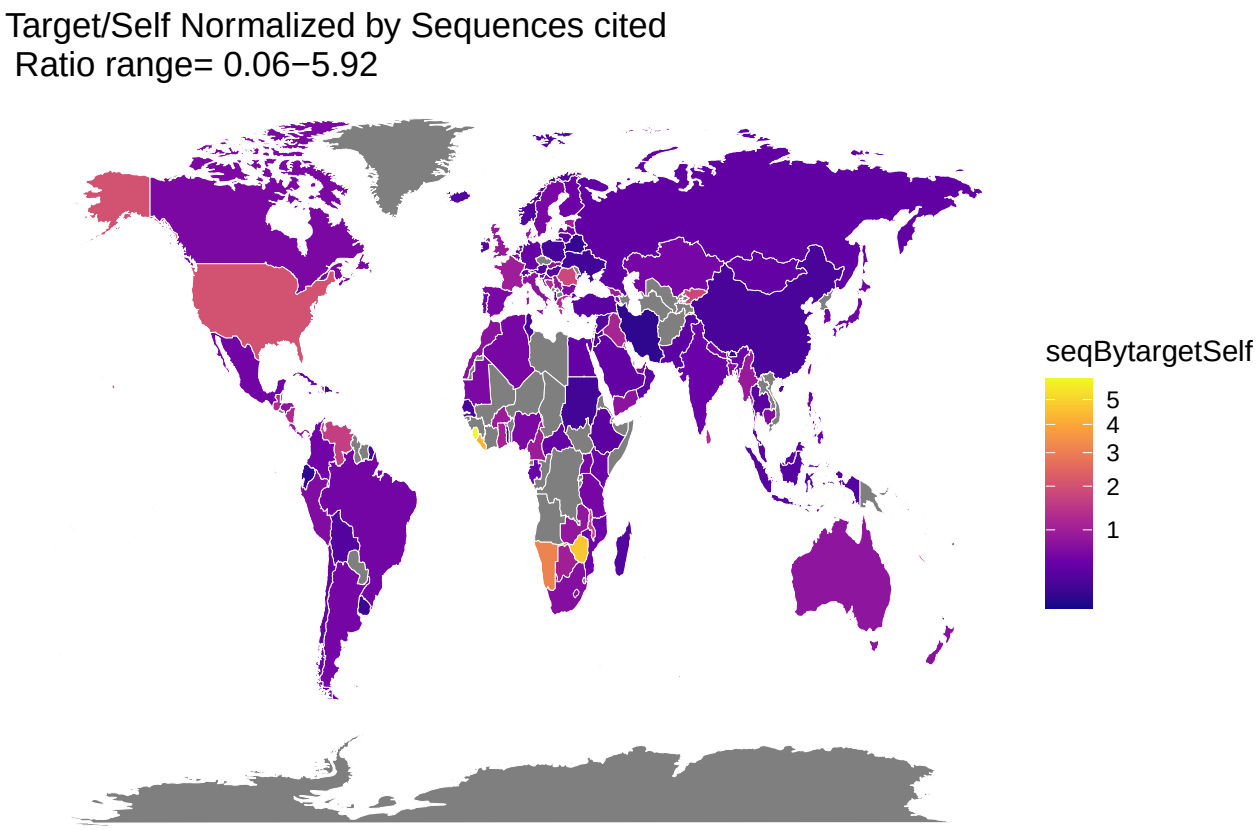


-

```
target_self <- left_join(selfUseOnly,  targetUseOnly, by=c("author_country"="seq_country")) %>% rename(s
  mutate(target_over_self=targetScientist/selfScientist) %>% select (author_country,targetScientist, sel

norm_target_self <- left_join(selfUseOnly,  targetUseOnly, by=c("author_country"="seq_country")) %>%
  rename(selfScientist=nscientist.x, targetScientist=nscientist.y, selfNseq=nseq.x, targetNseq=nseq.y, s
  mutate(seqBytargetSelf=(targetNseq/targetScientist)/(selfNseq/selfScientist), pubBytargetSelf=(targetN
  select (author_country,targetNseq,targetNpub, targetScientist, selfNseq, selfNpub, selfScientist, seqN

targetSelf_map <- left_join(world_map, norm_target_self, by = c("region"="author_country"))
```

-

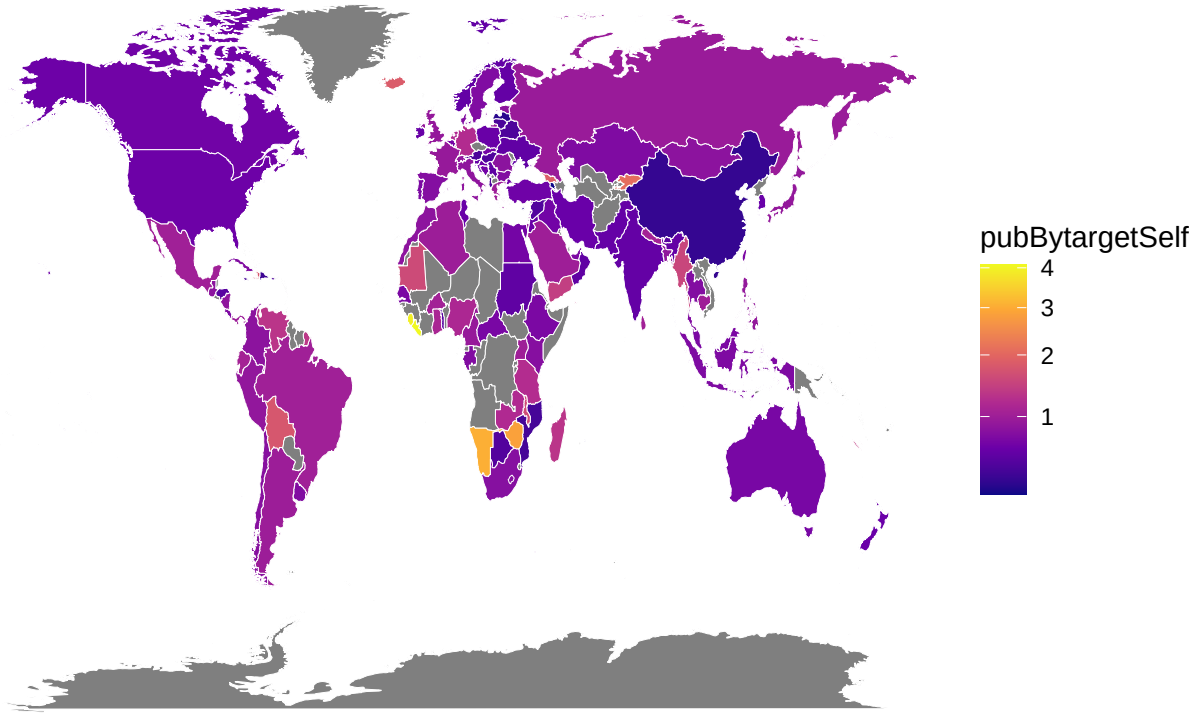
```
ggplot(targetSelf_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = seqBytargetSelf), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') +
  ggtitle(paste('Target/Self Normalized by Sequences cited \n Ratio range=', paste(round(range(na.omit(t
```



- #### Create the map / By Pub

```
ggplot(targetSelf_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = pubBytargetSelf), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') +
  ggtitle(paste('Target/Self Normalized by Number of publication \n Ratio range=', paste(round(range(na
```

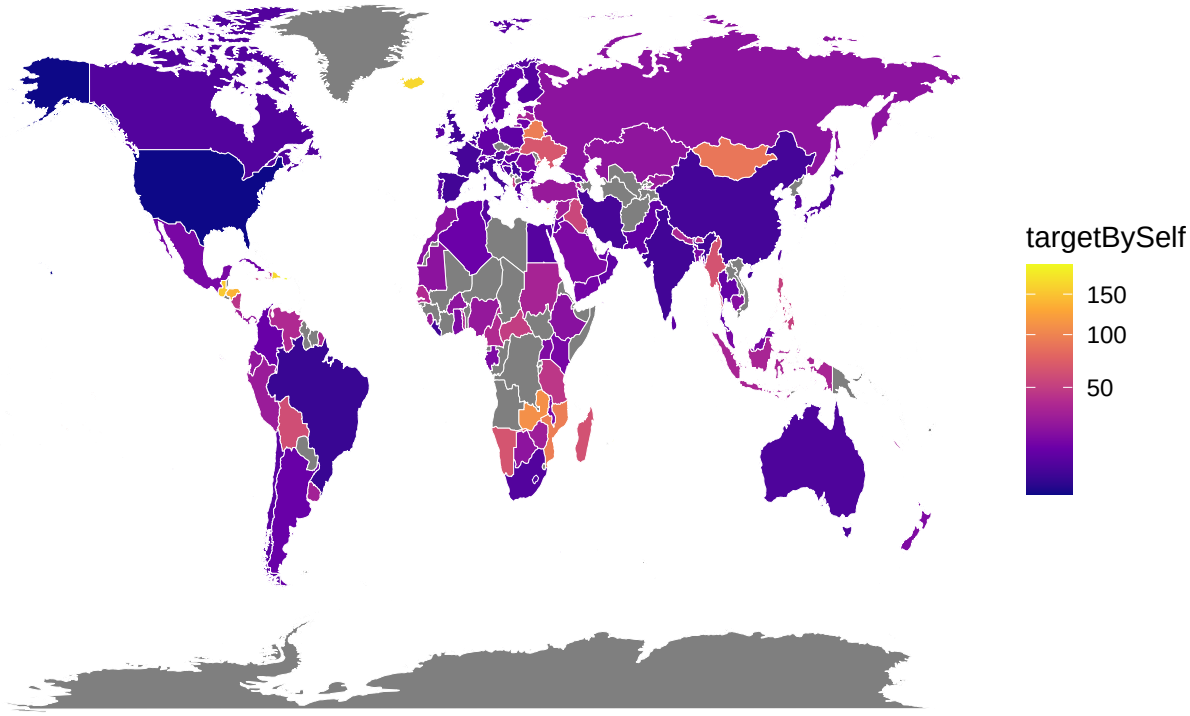
Target/Self Normalized by Number of publication
Ratio range= 0.23−4.08



-

```
ggplot(targetSelf_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = targetBySelf), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') +
  ggtitle(paste('Target/Self no normalization \n Ratio range=', paste(round(range(na.omit(targetSelf_ma
```

Target/Self no normalization
Ratio range= 1.27−192.5



Use case #2: General statistics across ALL countries

```
countries
```

```
## # A tibble: 252 x 6
##    isocode iso   iso3  name                  continent geonameid
##      <int> <chr> <chr> <chr>                 <chr>         <dbl>
##  1      20 AD    AND   Andorra               EU          3041565
##  2     784 AE    ARE   United Arab Emirates AS           290557
##  3       4 AF    AFG   Afghanistan           AS          1149361
##  4      28 AG    ATG   Antigua and Barbuda   NA          3576396
##  5     660 AI    AIA   Anguilla              NA          3573511
##  6       8 AL    ALB   Albania               EU           783754
##  7      51 AM    ARM   Armenia               AS           174982
##  8      24 AO    AGO   Angola                AF          3351879
##  9      10 AQ    ATA   Antarctica            AN          6697173
## 10      32 AR    ARG   Argentina             SA          3865483
## # ... with 242 more rows
```

```
pmc_ena
```

```
## # A tibble: 2,152,494 x 35
##     `_id` accession idpmc source pubtype issn  isopenaccess secondary_pmid
##     <int> <chr>     <chr> <chr>  <chr>   <chr> <chr>        <chr>
##  1 359119 AB075606  1450~ MED    Resear~ 0019~ N            14500536
##  2 359120 AB075606  1450~ MED    Resear~ 0019~ N            14500536
##  3 379508 AB115664  2841~ MED    resear~ 1949~ Y            28415720
##  4 379509 AB115664  2841~ MED    resear~ 1949~ Y            28415720
##  5 379510 AB115664  2841~ MED    resear~ 1949~ Y            28415720
##  6 379511 AB115664  2841~ MED    resear~ 1949~ Y            28415720
##  7 379512 AB115664  2841~ MED    resear~ 1949~ Y            28415720
##  8 379513 AB115664  2841~ MED    resear~ 1949~ Y            28415720
##  9 379514 AB115664  2841~ MED    resear~ 1949~ Y            28415720
## 10 379515 AB115664  2841~ MED    resear~ 1949~ Y            28415720
## # ... with 2,152,484 more rows, and 27 more variables: secondary_pmcid <chr>,
## #   secondary_doi <chr>, author <chr>, affiliation <chr>, author_country <chr>,
## #   first_pub_date <chr>, first_epub_date <chr>, author_orcid <chr>,
## #   language <chr>, grantid <chr>, grant_agency <chr>, grant_acronym <chr>,
## #   receipt_data <chr>, revision_date <chr>, ena_accession <chr>,
## #   primary_pmid <chr>, primary_doi <chr>, primary_pmcid <chr>,
## #   seq_origin <chr>, seq_country <chr>, submission_date <chr>,
## #   first_created <chr>, seq_lat_lon <chr>, organism <chr>, taxid <chr>,
## #   code <chr>, project_acc <chr>
```

```
countries
```

```
## # A tibble: 252 x 6
##    isocode iso   iso3  name                continent geonameid
##      <int> <chr> <chr> <chr>               <chr>         <dbl>
##  1      20 AD    AND   Andorra             EU          3041565
##  2     784 AE    ARE   United Arab Emirates AS           290557
##  3       4 AF    AFG   Afghanistan         AS          1149361
##  4      28 AG    ATG   Antigua and Barbuda  NA          3576396
##  5     660 AI    AIA   Anguilla            NA          3573511
##  6       8 AL    ALB   Albania             EU           783754
##  7      51 AM    ARM   Armenia             AS           174982
##  8      24 AO    AGO   Angola              AF          3351879
##  9      10 AQ    ATA   Antarctica          AN          6697173
## 10      32 AR    ARG   Argentina           SA          3865483
## # ... with 242 more rows
```

```
# Append North South to pmc_ena
pmc_ena <- tbl_df(left_join(pmc_ena, countries, by=c("author_country"="name")))
pmc_ena_north_south <- pmc_ena %>% select(accession, secondary_pmid, author, affiliation, author_country
  group_by(secondary_pmid) %>%
  mutate(north=ifelse(continent %in% c('EU','NA'), TRUE, FALSE), south=ifelse(continent %in% c('AS','AF
```

-

```
# All authors are NORTH
north_north <- pmc_ena_north_south %>%
```

```
  filter(all(north)) %>% ungroup() %>% select(secondary_pmid, north)  %>% unique()
# Number of North-North Publication
dim(north_north )[1]
```

```
## [1] 47865
```

-

```
# Number of North-South Publication : The publication has a least one other from both North and south.
# Table with the corresponding pmid follows
north_south<- pmc_ena_north_south %>%  filter(any(north) & any(south)) %>%
  ungroup() %>% select (secondary_pmid) %>% unique()
# Number of North-South Publication
dim(north_south)[1]
```

```
## [1] 498
```

-

```
# Strictly South author
south_south <- pmc_ena_north_south %>%  filter(all(south)) %>%
  ungroup() %>% select (secondary_pmid) %>% unique()
# Number of North-South Publication
dim(south_south)[1]
```

```
## [1] 20126
```

-

```
# Table with the corresponding pmid follows
one_country_pub <- pmc_ena_north_south %>% select(secondary_pmid, author_country) %>%
  group_by(secondary_pmid) %>%
  mutate(country_count = length(unique(author_country))) %>%
  select(secondary_pmid, country_count) %>%
  filter(country_count==1) %>%
  ungroup() %>% unique()

# How many publications are 1 country only?
dim(one_country_pub)[1]
```

```
## [1] 68718
```

-

```
gmean <- function(x) exp(mean(log(x)))
pmc_ena %>% select(secondary_pmid, author_country) %>%
  unique() %>% group_by(secondary_pmid) %>%
  mutate(ncountry=length(unique(author_country))) %>%
  select(secondary_pmid, ncountry) %>%
  unique() %>% ungroup() %>% unique() %>%
  summarise(geometric_mean=gmean(ncountry))
```

```
## # A tibble: 1 x 1
##   geometric_mean
##            <dbl>
## 1           1.01
```

Use case #3: For each country in the world please make a map showing:

-

```
# Extract list of pmid with cameroonian author
# pmc_ena %>% select(secondary_pmid, author_country) %>% filter(author_country =='Cameroon') %>% select

collaboration_count <- function(countryName='Cameroon'){
  pmid <-  pmc_ena  %>% filter(author_country ==countryName) %>%
    select(secondary_pmid) %>%unique()
  if(length(pmid$secondary_pmid)>0){
    collab <- pmc_ena %>% filter(secondary_pmid %in% pmid$secondary_pmid) %>%
    select(secondary_pmid, author_country) %>% unique() %>%
    filter(!author_country==countryName) %>% select(author_country) %>% unique()
    if(dim(collab)[1] > 0){
      result <- data.frame(country=countryName, ncollab=dim(collab)[1], partner=collab$author_country)
      result
  }
  }
}

collabList <- mclapply(countries$name, function(x) collaboration_count (countryName=x))

collaborator<- tbl_df(do.call(rbind, collabList))
collaborator
```

```
## # A tibble: 2,206 x 3
##    country              ncollab partner
##    <chr>                  <int> <chr>
##  1 United Arab Emirates       1 France
##  2 Anguilla                   1 Italy
##  3 Armenia                    1 Colombia
##  4 Argentina                 41 China
```

```
##  5 Argentina                 41 United States
##  6 Argentina                 41 India
##  7 Argentina                 41 Ecuador
##  8 Argentina                 41 Paraguay
##  9 Argentina                 41 Georgia
## 10 Argentina                 41 Zambia
## # ... with 2,196 more rows
```

```
collaborator %>% select(country, ncollab) %>% unique()
```

```
## # A tibble: 120 x 2
##    country              ncollab
##    <chr>                  <int>
##  1 United Arab Emirates       1
##  2 Anguilla                   1
##  3 Armenia                    1
##  4 Argentina                 41
##  5 Austria                   45
##  6 Australia                 46
##  7 Bangladesh                 4
##  8 Belgium                   47
##  9 Bulgaria                  17
## 10 Benin                      1
## # ... with 110 more rows
```

-

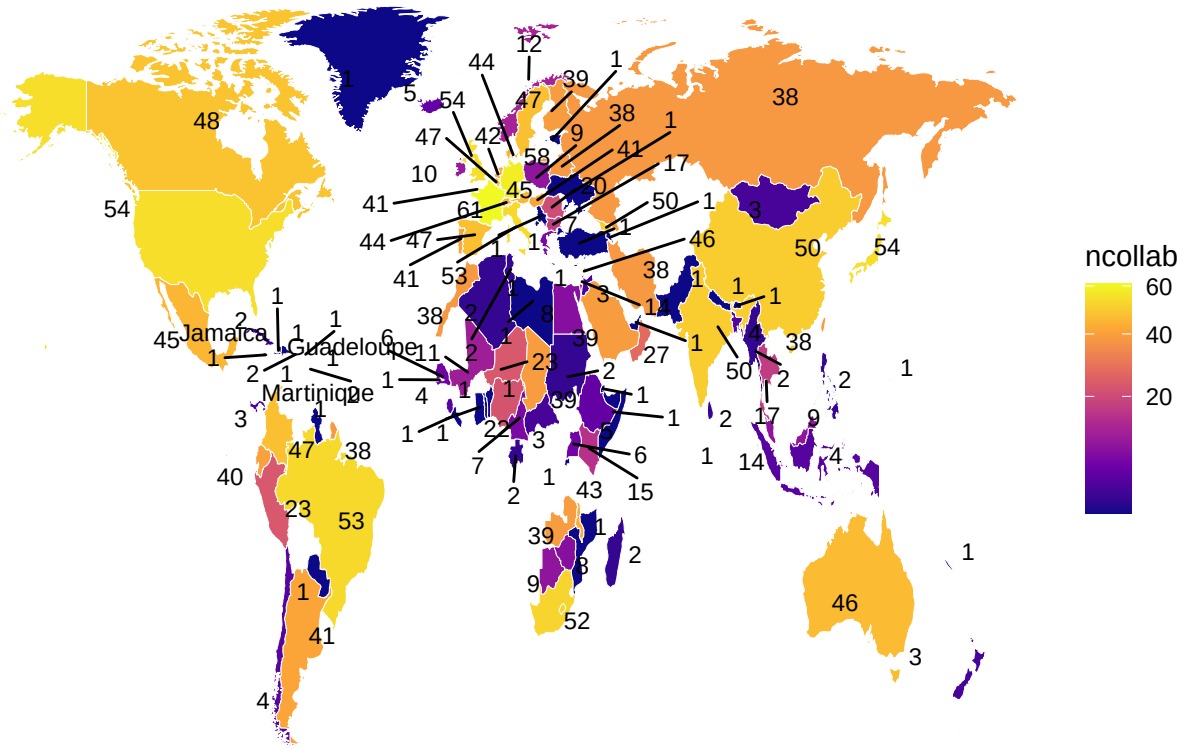
```
collab_map <-tbl_df(left_join(collaborator %>% select(country, ncollab) %>% unique(), world_map, by=c("
island <- c('Tonga','Solomon Islands','Samoa','Tonga','Marshall Islands','Guadeloupe','Martinique','Ber
label_data <- collab_map %>% group_by(country, ncollab) %>% summarise(long=mean(long), lat=mean(lat))
```

```
## `summarise()` regrouping output by 'country' (override with `.groups` argument)
```

```
ggplot(collab_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = ncollab), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') +
  geom_text_repel(data=label_data, aes(x=long, y=lat, label=ncollab),size=3) +
  geom_text_repel(data=label_data %>% filter(country %in% island), aes(x=long, y=lat, label=country),si
  ggtitle('Collaborators') +
  theme_void()
```

```
## Warning: Removed 3 rows containing missing values (geom_text_repel).
```

Collaborators



- #### Create a traffic map:

```
label_data2 <- tbl_df(left_join(label_data, collaborator, by=c('country'='country')))
country_centroid<- tbl_df(world_map) %>% group_by(region) %>% summarise(long=mean(long), lat=mean(lat))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
collabs <- tbl_df(left_join(label_data2, country_centroid, by=c('partner'='region')))

map_collaboration <- function(countryName='Cameroon'){
  map_result <- ggplot(collab_map, aes(long, lat, group = group))+
    geom_polygon(aes(fill = ncollab), color = "white", size=0.05)+
    scale_fill_viridis_c(option = "C", trans='sqrt') +
    geom_text_repel(data=label_data, aes(x=long, y=lat, label=ncollab),size=3) +
    geom_text_repel(data=label_data %>% filter(country %in% island), aes(x=long, y=lat, label=country),s
    geom_point(data = label_data, aes(x = long, y = lat), col = "#970027") +
    geom_curve(data=collabs %>% filter(country %in% countryName), aes(x = long.x, y = lat.x, xend = long
    ggtitle(paste(countryName,' Collaborators')) +
    theme_void()
  map_result
}

top_country <- lapply(c('United States', 'United Kingdom', 'Germany'), function(x) map_collaboration(cou
```

```
top_country
```

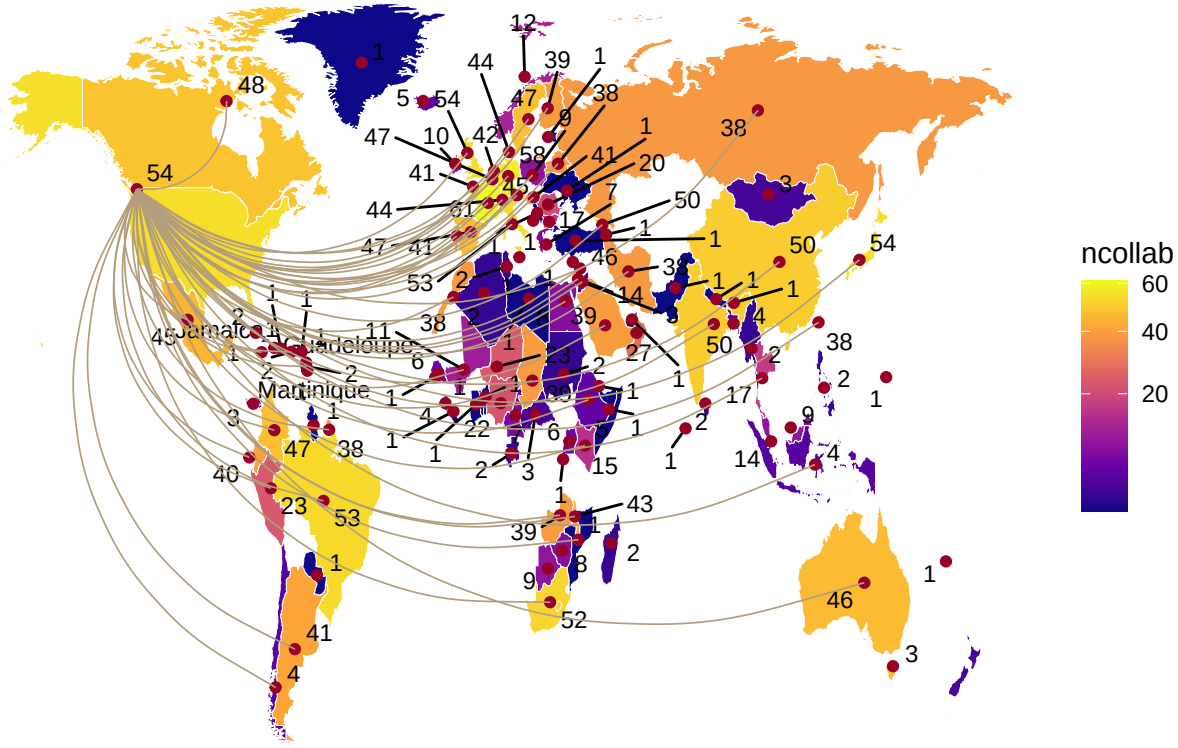
```
## [[1]]
```

```
## Warning: Removed 3 rows containing missing values (geom_text_repel).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 rows containing missing values (geom_curve).
```

United States Collaborators



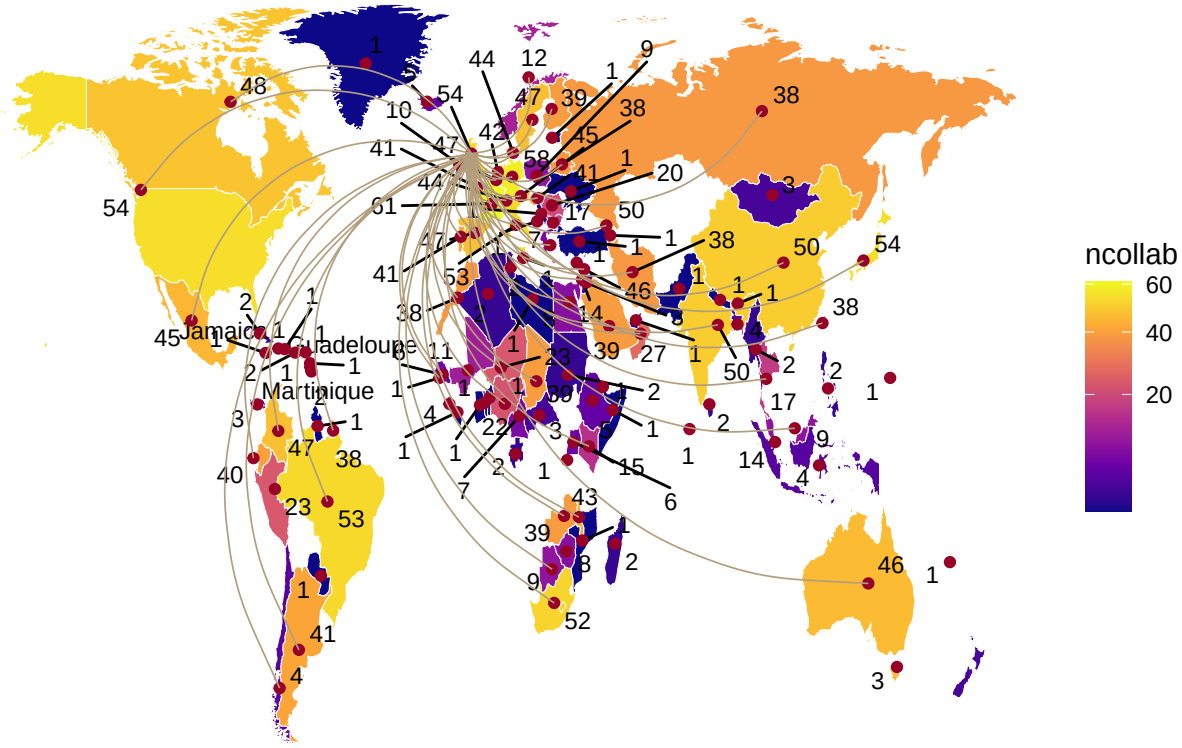

```
##
## [[2]]
```

```
## Warning: Removed 3 rows containing missing values (geom_text_repel).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

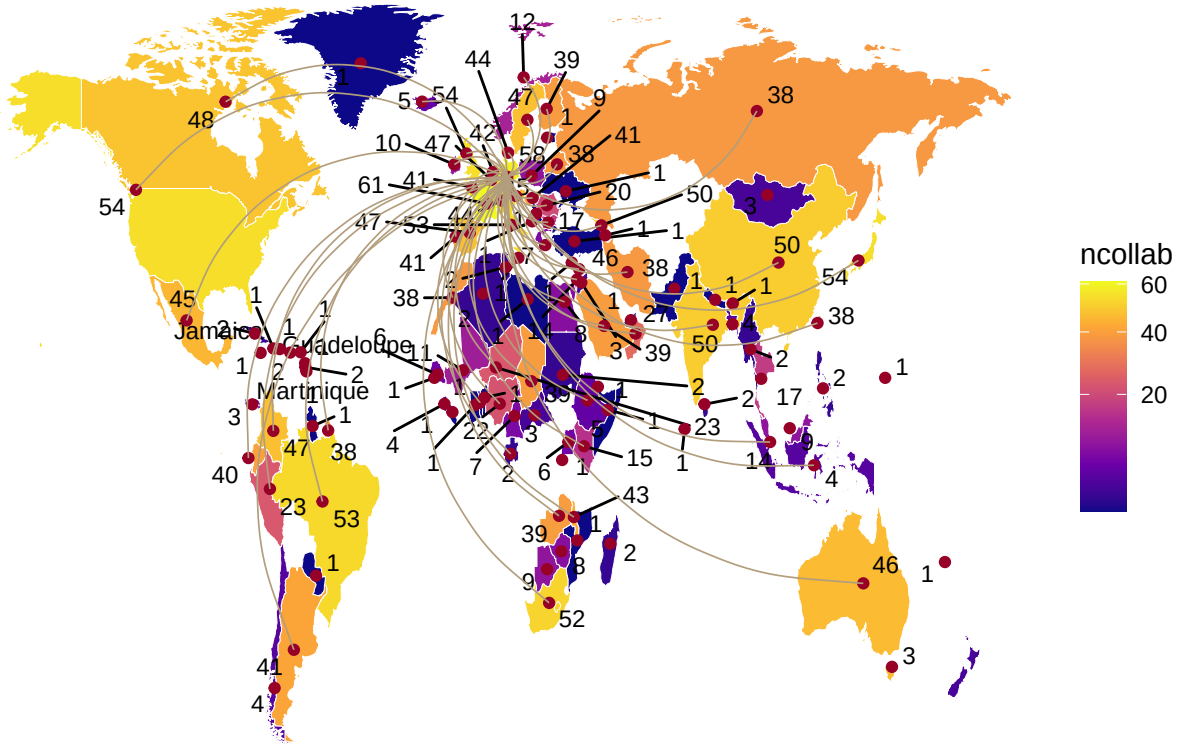
```
## Warning: Removed 1 rows containing missing values (geom_curve).
```

United Kingdom Collaborators

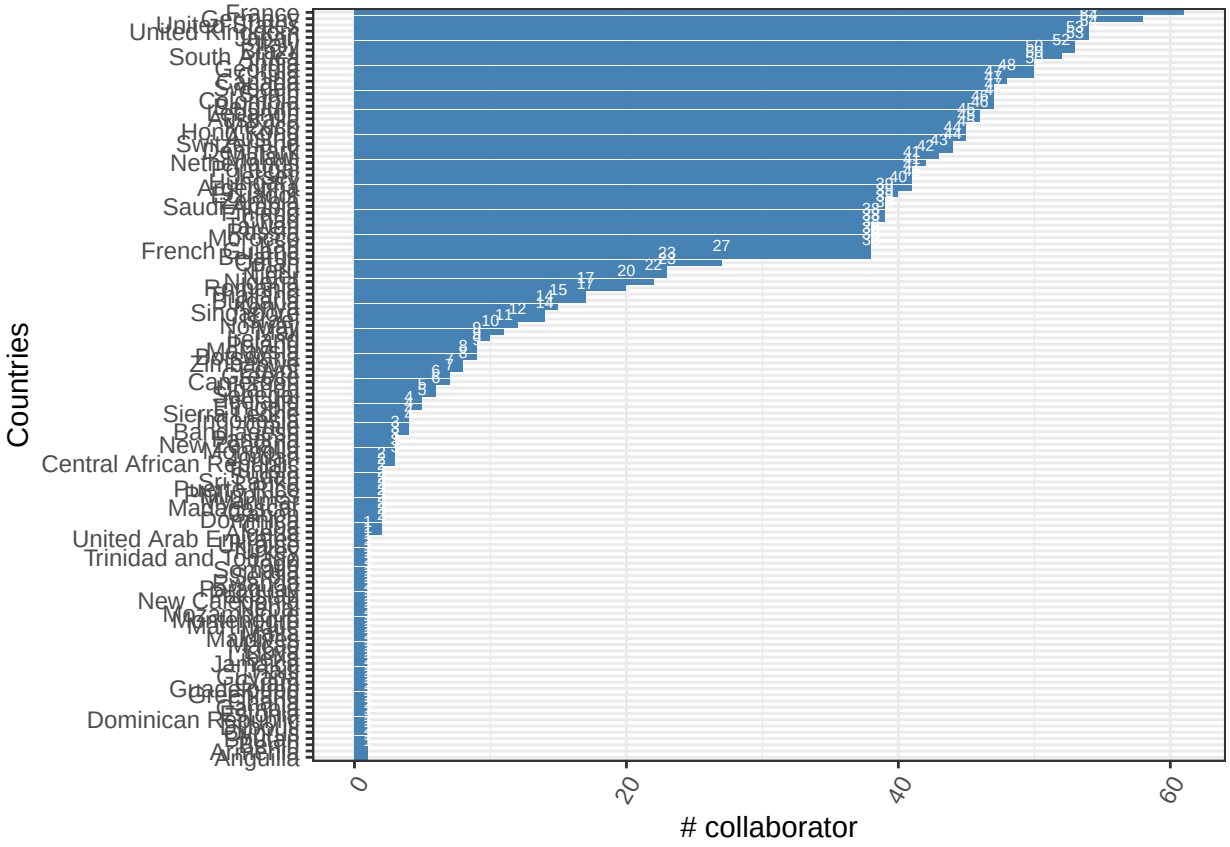



```
##
## [[3]]

## Warning: Removed 3 rows containing missing values (geom_text_repel).

## Warning: Removed 3 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_curve).
```

Germany Collaborators




```
ggplot(data=label_data %>% arrange(desc(ncollab)), aes(x=reorder(country, ncollab), y=ncollab)) +
  geom_bar(stat="identity", fill="steelblue")+
  geom_text(aes(label=ncollab), vjust=-1, size=2, col='white') + theme_bw() + ylab('# collaborator')+ xl
```



-

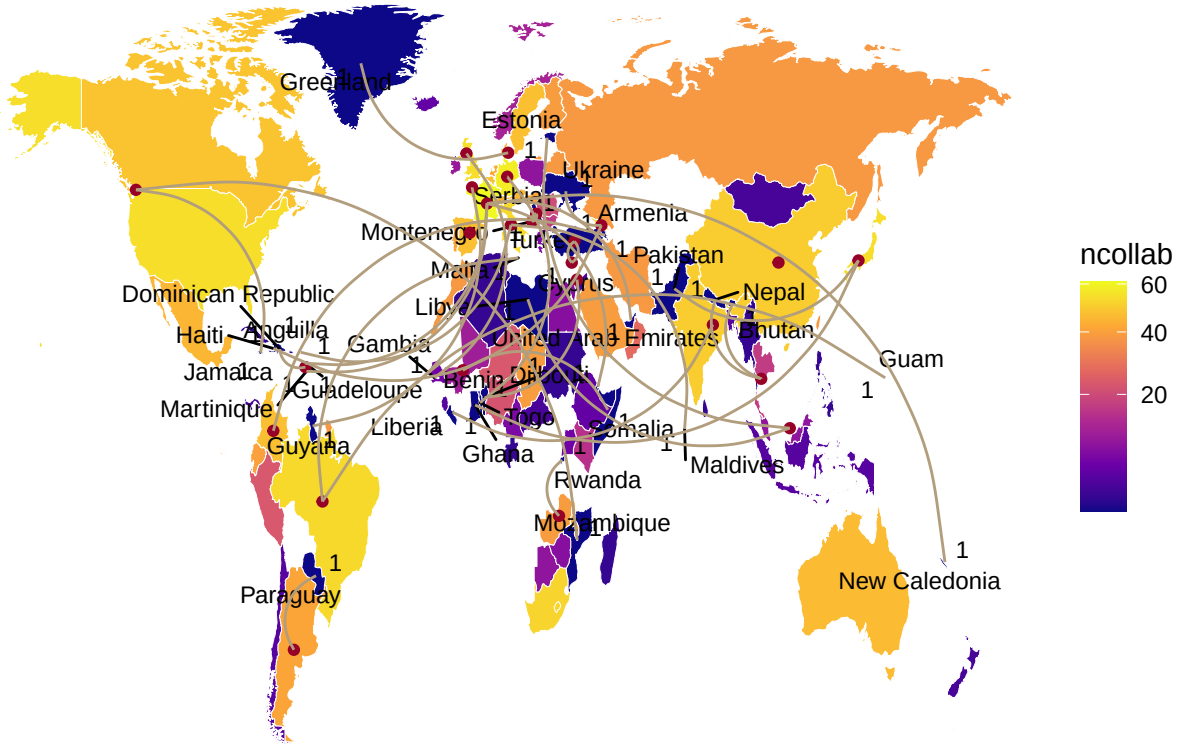
```
onecollab <- label_data %>% filter(ncollab==1) %>% select(country) %>% unique()
partner <- collaborator %>% filter(country %in% onecollab$country) %>% select(partner)
ggplot(collab_map , aes(long, lat, group = group))+
  geom_polygon(aes(fill = ncollab), color = "white", size=0.05)+
  scale_fill_viridis_c(option = "C", trans='sqrt') +
  geom_text_repel(data=label_data %>% filter(ncollab==1), aes(x=long, y=lat, label=ncollab),size=3) +
  geom_text_repel(data=label_data %>% filter(ncollab==1) %>% filter(country %in% onecollab$country) , ae
  geom_point(data = label_data %>% filter(country %in% partner$partner) , aes(x = long, y = lat), col =
  geom_curve(data=collabs %>% filter(ncollab.x==1) , aes(x = long.x, y = lat.x, xend = long.y, yend = la
  ggtitle(paste('How many single collaborations do they have?')) +
  theme_void()
```

```
## Warning: Removed 2 rows containing missing values (geom_text_repel).
```

```
## Warning: Removed 2 rows containing missing values (geom_text_repel).
```

```
## Warning: Removed 2 rows containing missing values (geom_curve).
```

How many single collaborations do they have?



-

```
dataUsage <- function(x='China'){
  result <- pmc_ena %>% filter(author_country==x) %>% select(author_country, seq_country) %>% unique() %
  result <- data.frame(country=x, country_count=result$country_count)
  result
}

odusage <- do.call(rbind, mclapply(countries$name, function(x) dataUsage(x)))
```

-

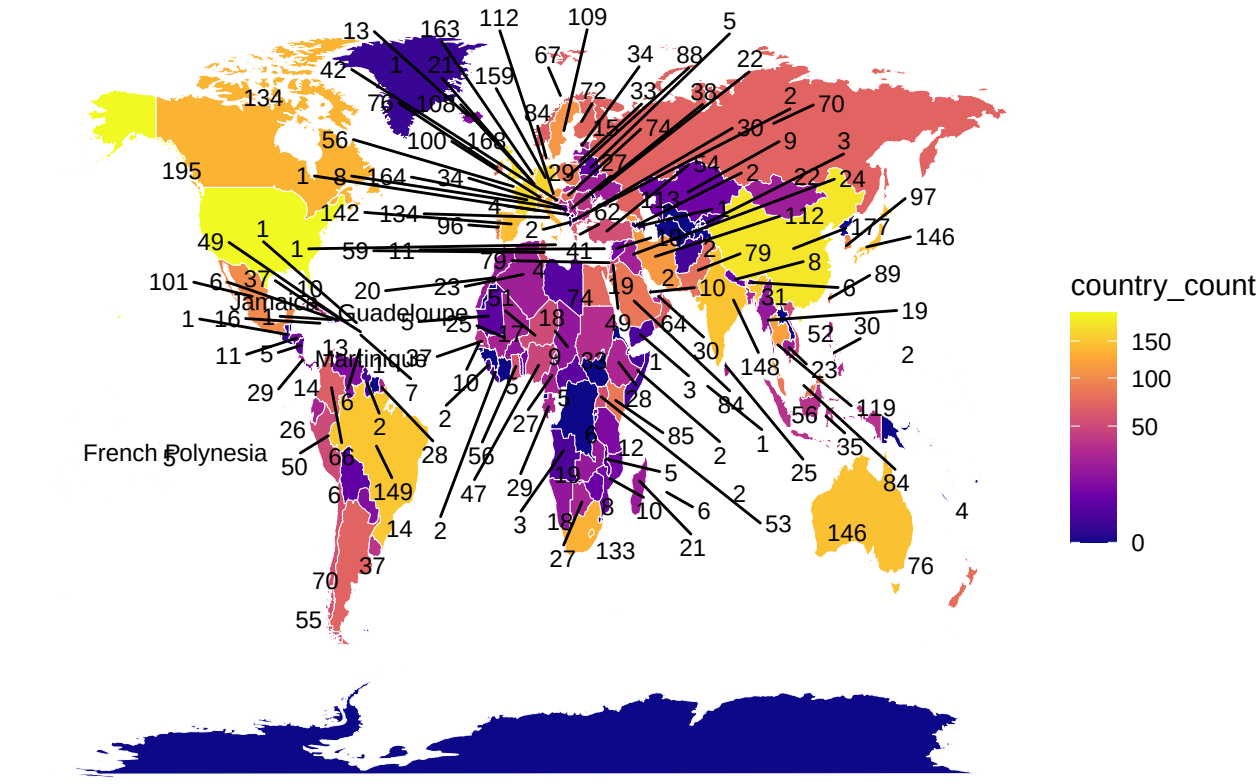
```
otherdata_map <-tbl_df(left_join(odusage, world_map, by=c("country"="region")))
island <- c('Tonga','Solomon Islands','Samoa','Tonga','Marshall Islands','Guadeloupe','Martinique','Ber
label_data <- otherdata_map %>% group_by(country, country_count) %>% summarise(long=mean(long), lat=mea
```

```
## `summarise()` regrouping output by 'country' (override with `.groups` argument)
```

```
ggplot(otherdata_map, aes(long, lat, group = group))+
  geom_polygon(aes(fill = country_count), color = "white", size=0.05)+
```

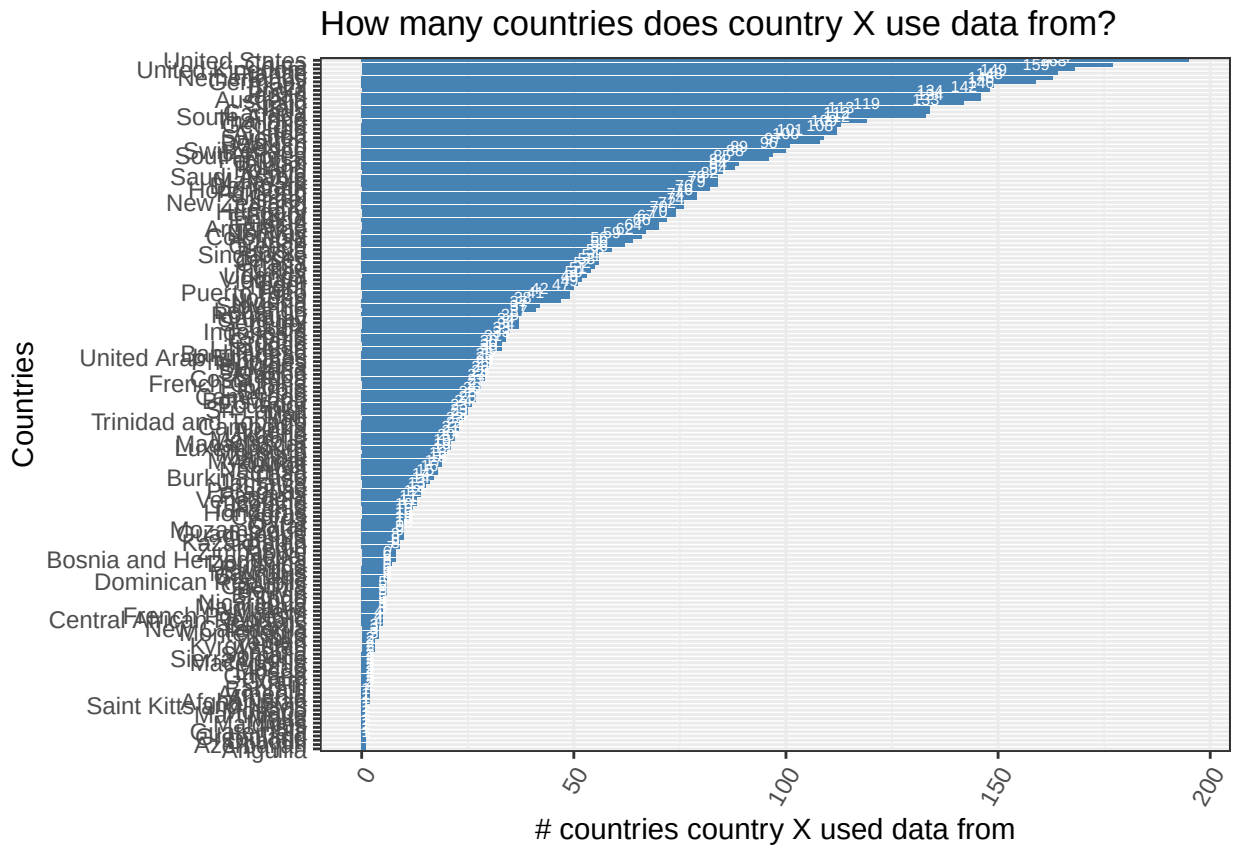
```
  scale_fill_viridis_c(option = "C", trans='sqrt') +
  geom_text_repel(data=label_data %>% filter(country_count>0), aes(x=long, y=lat, label=country_count),s
  geom_text_repel(data=label_data  %>% filter(country_count>0) %>% filter(country %in% island), aes(x=lo
  ggtitle('How many countries does country X use data from?') +
  theme_void()
```

```
## Warning: Removed 5 rows containing missing values (geom_text_repel).
```



-

```
ggplot(data=label_data %>% filter(country_count>0), aes(x=reorder(country, country_count), y=country_cou
  geom_bar(stat="identity", fill="steelblue")+
  geom_text(aes(label=country_count), vjust=-1, size=2, col='white') + theme_bw() +
  ylab('# countries country X used data from')+ xlab('Countries') +
  theme(axis.text.x = element_text(angle = 60, hjust = 1)) + coord_flip() + ggtitle('How many countries
```




```
#save(pmc_ena, countries, continent, file="seqref.Rdata")
sessionInfo()
```

```
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
##
## locale:
##  [1] LC_CTYPE=en_GB.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_GB.UTF-8        LC_COLLATE=en_GB.UTF-8
##  [5] LC_MONETARY=en_GB.UTF-8    LC_MESSAGES=en_GB.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
```

```
##  [1] ggrepel_0.8.2     viridis_0.5.1     viridisLite_0.3.0 maps_3.3.0
##  [5] forcats_0.5.0     stringr_1.4.0     dplyr_1.0.0       purrr_0.3.4
##  [9] readr_1.3.1       tidyr_1.1.0       tibble_3.0.3      ggplot2_3.3.2
## [13] tidyverse_1.3.0   RPostgreSQL_0.6-2 DBI_1.1.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.0 xfun_0.15        haven_2.3.0      lattice_0.20-41
##  [5] colorspace_1.4-1 vctrs_0.3.2      generics_0.0.2   htmltools_0.5.0
##  [9] yaml_2.2.1       utf8_1.1.4       blob_1.2.1       rlang_0.4.7
## [13] pillar_1.4.6     glue_1.4.1       withr_2.2.0      dbplyr_1.4.4
## [17] modelr_0.1.8     readxl_1.3.1     lifecycle_0.2.0  munsell_0.5.0
## [21] gtable_0.3.0     cellranger_1.1.0 rvest_0.3.5      evaluate_0.14
## [25] labeling_0.3     knitr_1.29       fansi_0.4.1      broom_0.5.6
## [29] Rcpp_1.0.5       scales_1.1.1     backports_1.1.8  jsonlite_1.7.0
## [33] farver_2.0.3     fs_1.4.2         gridExtra_2.3    hms_0.5.3
## [37] digest_0.6.25    stringi_1.4.6    grid_4.0.2       cli_2.0.2
## [41] tools_4.0.2      magrittr_1.5     crayon_1.3.4     pkgconfig_2.0.3
## [45] ellipsis_0.3.1   xml2_1.3.2       reprex_0.3.0     lubridate_1.7.8
## [49] assertthat_0.2.1 rmarkdown_2.2    httr_1.4.1       rstudioapi_0.11
## [53] R6_2.4.1         nlme_3.1-147     compiler_4.0.2
```

```
#rm(list=ls())
#ls()
```