Project 2

simplymathematics

October 7, 2018

Instead of using three different databases, I wanted to build a reference database for the data I collected previously. The first step, as always, is to get your environment ready.

```
library(curl, quietly = TRUE)
library(XML, quietly= TRUE)
library(stringr, quietly= TRUE)
suppressPackageStartupMessages(library(tidyverse, quietly = TRUE))
```

Internet Exchange Points

I wanted an update-able dataset of Internet Exchange points around the world. Each NIC (regional IP/TCP benevolent overlords) has datasets of their own, but I'd have to parse each individually. Wikipedia seems to keep an accurate enough dataset.

Then, I used curl to import the data and the XML library to parse it as a tree

```
data.file <- curl_download("https://en.wikipedia.org/wiki/List_of_Internet_exchange_points_by_size", "I

raw.data <- readHTMLTable(data.file)
```

Then, I had set the first row of the data frame as the column names.

```
data <- data.frame(raw.data[])
colnames(data) <- as.character(unlist(data[1,]))
data = data[-1, ]
head(data)
```

```
##   Short name                                    Name
## 2     DE-CIX Deutsche Commercial Internet Exchange[1]
## 3      IX.br              Brazil Internet Exchange[4]
## 4     AMS-IX           Amsterdam Internet Exchange[7]
## 5       LINX             London Internet Exchange[18]
## 6     MSK-IX                                MSK-IX[22]
## 7      NL-ix            Neutral Internet Exchange[25]
##
## 2
## 3 Aracaju, Belém, Belo Horizonte, Brasília, Campina Grande, Campinas, Caxias do Sul, Cuiabá, Curitiba
## 4
## 5
## 6
## 7
##
## 2                                                                                          Germany
## 3
## 4
```

```
## 5
## 6
## 7  Austria,  Belgium,  Czech Republic,  Denmark,  France,  Germany,  Italy,  Luxembourg,  Netherlands
##   Established  Members Maximum throughput (Gb/s) Average throughput (Gb/s)
## 2        1995   735[2]                6,408[3]                 4,004[3]
## 3        2004 3,121[5]                5,710[6]                 3,740[6]
## 4    1997[15]  818[16]               5,513[17]                3,339[17]
## 5        1994  825[20]               4,340[21]                2,850[21]
## 6        1995  504[23]               2,821[24]                1,211[24]
## 7    2002[28]  600[29]               1,770[30]                  979[30]
##     Values updated
## 2    15 March 2018
## 3   20 August 2018
## 4 13 December 2017
## 5       2 May 2018
## 6 25 February 2017
## 7  19 October 2016
```

Finally, I wanted to spread the data so that each city had its own listing with corresponding provider information. First I had to separate the cities and treat them as independent variables.

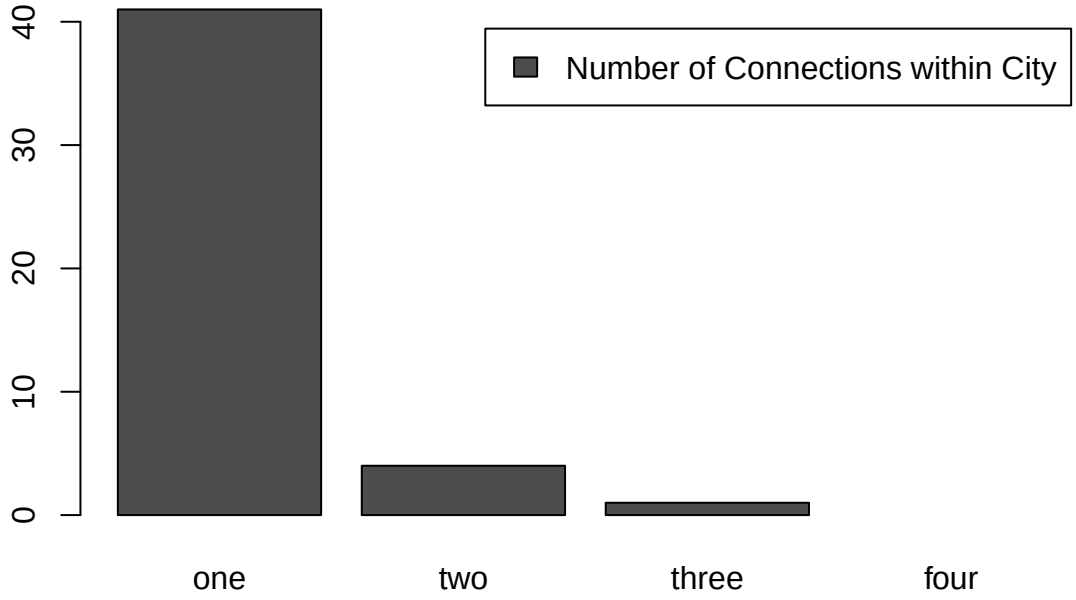
```
data <- mutate(data, City = strsplit(as.character(City), ",")) %>%
    unnest(City)
```

I then had to clean up the cities and figure out how many connections they each had.

```
cities <- unique(trimws(data$City)) #cleaning
connections_per_city <- data.frame()
for (city in cities){ #counting
connections <- dim(subset(data, City == city))[1]
new.row <- cbind(city, connections)
connections_per_city <- rbind(new.row, connections_per_city)
}
one <- sum(connections_per_city$connections == 1) #binning
two <- sum(connections_per_city$connections == 2)
three <- sum(connections_per_city$connections == 3)
four <- sum(connections_per_city$connections > 4)
```

```
## Warning in Ops.factor(connections_per_city$connections, 4): '>' not
## meaningful for factors
```

```
distribution <- cbind(one, two, three, four)
barplot(distribution, legend.text = "Number of Connections within City")
```




```
write.csv(data, file = "IXPs.csv")
```

This data is stil messy, in particular because the city data isn't standardized. For example, the unique() fucntion does not understand the similarity between "NYC" and " New York City". Manual work will have to be done to collapse all of these cities. Additionally, I do not know whether Seacaucus NJ should count as NYC for the purposes of this project since it serves the same metropolitan area and fiber optic signals travel nearly the spped of light. However, despite this limited issue, we can see that the vast majority of cities in the world have only a single high-level connection to *the* internet, and that that privilege is only granted to a few cities.

Fiber to the Home

I also wanted a list of cities and municipalities that hand locally-controlled infrastructure. For that, I scraped the Muninetworks website.

```
data.file2 <- curl_download("https://muninetworks.org/content/municipal-ftth-networks", "FTTH/ftth.html
data.file2
```

```
## [1] "/home/nologs/Mesh-Data/data/Global/FTTH/ftth.html"
```

Next, I read the file in, line by line.

```
lines <- readLines(data.file2)
head(lines)
```

```
## [1] "<!DOCTYPE html>"
## [2] "<html  lang=\"en\" dir=\"ltr\" prefix=\"og: http://ogp.me/ns# article: http://ogp.me/ns/article#
## [3] "<head>"
## [4] "  <meta charset=\"utf-8\" />"
## [5] "<link rel=\"shortcut icon\" href=\"https://muninetworks.org/sites/www.muninetworks.org/files/fav
## [6] "<meta name=\"description\" content=\"This is a list of municipalities across the United States
```

Then, I used a text editor to find the first line in my dataset. From there, I reconstructed the html table using regex.

```
first.chunk <- which(grepl("<p><strong>", lines))
#lines[first.chunk]
networks <- c(str_extract(lines[first.chunk], "(?<=<strong>)(.*\n?)(?=</strong>)"))
communities <- c(str_extract(lines[first.chunk+1], "(?<=Served: )(.*)(?=</em>)" ))
years <- c(str_extract(lines[first.chunk+3], "(?<=Year: )(.*)(?=</li>)" ))
populations <- c(str_extract(lines[first.chunk+4], "(?<=Population: )(.*)(?=</li>)" ))
costs <- c(str_extract(lines[first.chunk+5], "(?<=Cost: )(.*)(?=</li>)"))
funding.methods <- c(str_extract(lines[first.chunk+6], "(?<=Method: )(.*)(?=</li>)" ))
governances <- c(str_extract(lines[first.chunk+7], "(?<=Governance: )(.*)(?=</li>)" ))
services <- c(str_extract(lines[first.chunk+8], "(?<=Services: )(.*)(?=</li>)" ))
speeds <- c(str_extract(lines[first.chunk+9], "(?<=Speed: )(.*)(?=</li>)" ))
costs
```

```
##  [1] "around $8 million"
##  [2] "<em>Unknown</em>"
##  [3] "$45.3 million"
##  [4] "About $13 million"
##  [5] "$33 million"
##  [6] NA
##  [7] "$19.3 million ($17 million, overbuild; $2.3 million, expansion)"
##  [8] "$1.5 million"
##  [9] "About $6.5 million"
## [10] "About $14 million"
## [11] "$9 million"
## [12] "$12 million"
## [13] "$11 million"
## [14] "$160 million"
## [15] "$4 million"
## [16] "$3.6 million"
## [17] "about $2 million"
## [18] "$10 million"
## [19] "$12.8 million"
## [20] "About $10 - $12 million"
## [21] "$10.5 million"
## [22] "<em>Unknown</em>"
## [23] "$33 million"
## [24] "$29 million"
## [25] "$40 million"
## [26] "$7.5 million"
## [27] "$7.5 million"
```

```
## [28] "About $27 million"
## [29] "$4 million"
## [30] NA
## [31] "$388.4 million"
## [32] "About $17 million"
## [33] "$54 million"
## [34] "About $1 million"
## [35] "$75 million"
## [36] "About $18 million"
## [37] "$8.5 million"
## [38] "$15 million"
```

Then I bound all the variables into a dataframe and separated the services.

```
data <- data.frame(cbind(networks, communities, years, populations, costs, funding.methods, governances
data <- separate(data, services, into = c("Service1", "Service2", "Service3"), sep = ',')
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 11 rows [1,
## 3, 4, 6, 15, 16, 17, 21, 22, 27, 34].
```

```
head(data,10)
```

```
##                                                networks
## 1                      Loma Linda Connected Community Program
## 2  Vernon Gas &amp;amp; Electric Department Fiber Optic Division
## 3                                                NextLight
## 4                                  Rio Blanco County Network
## 5                                                 OptiLink
## 6                                                     iVue
## 7                                      Cedar Falls Utilities
## 8                                    Lenox Municipal Utilities
## 9                                  Spencer Municipal Utilities
## 10                                           Connect Waverly
##                communities                                  years
## 1   Loma Linda, California                                   2005
## 2                     <NA>                                   2013
## 3       Longmont, Colorado                                   2014
## 4  Rangely, CO; Meeker, CO                                   2016
## 5          Dalton, Georgia                                   2003
## 6           Bellevue, Iowa            2009 (FTTH); 1992 (fiber)
## 7        Cedar Falls, Iowa            2011 (FTTH); 1995 (fiber)
## 8              Lenox, Iowa                                   2010
## 9            Spencer, Iowa 2013 (FTTH); 2000 (hybrid fiber-coax)
## 10             Waverly, IA                                   2016
##                         populations
## 1  23,000 (about 9,000 households)
## 2                                112
## 3                             90,000
## 4                              4,571
## 5                             33,500
## 6                              2,000
## 7                             34,000
```

```
## 8                            1,400
## 9                           11,000
## 10                          10,000
##                                                                  costs
## 1                                                  around $8 million
## 2                                                   <em>Unknown</em>
## 3                                                      $45.3 million
## 4                                                  About $13 million
## 5                                                        $33 million
## 6                                                               <NA>
## 7  $19.3 million ($17 million, overbuild; $2.3 million, expansion)
## 8                                                        $1.5 million
## 9                                                  About $6.5 million
## 10                                                  About $14 million
##
## 1                                                  in part, requirements for private developers to inclu
## 2
## 3
## 4   small matching grants from Colorado Department of Local Affairs (DOLA),  local Community Anchor
## 5
## 6
## 7
## 8
## 9
## 10                                                                  5 million in revenue bonds, 7 mil
##                                                            governances
## 1            Information Systems Department, under the City Manager
## 2                                                               <NA>
## 3         The electric utility, Longmont Power &amp;amp; Communications
## 4                                                  Rio Blanco County
## 5              The municipal utilities department, Dalton Utilities
## 6  The municipal utilities department, Bellevue Municipal Utilities
## 7         The municipal utilities department, Cedar Falls Utilities
## 8                                The municipal utilities department
## 9   The municipal utilities department, Spencer Municipal Utilities
## 10                          The electric utility, Waverly Utilities
##           Service1 Service2 Service3              speeds
## 1  Internet access     <NA>     <NA> 15 Mbps symmetrical
## 2              <NA>     <NA>     <NA>                <NA>
## 3  Internet access    voice     <NA>  1 Gbps symmetrical
## 4  Internet access     <NA>     <NA>  1 Gbps symmetrical
## 5  Internet access    voice    video    100 Mbps/10 Mbps
## 6  Internet access    video     <NA> 25 Mbps symmetrical
## 7  Internet access    voice    video     1 Gbps/500 Mbps
## 8  Internet access    voice   video  50 Mbps symmetrical
## 9  Internet access    voice    video  1 Gbps Symmetrical
## 10 Internet access    voice    video  1 Gbps symmetrical
```

Then I wrote it all to a csv

```
write.csv(data, file = "ftth.csv")
```

For my analysis, I wanted to see the cost/per person of building a fiber network. So I cleaned up the data by pulling out the dollar figures and removing the NAs.

```
communities <- data.frame(communities)
data <- data.frame(data)

#Funds (in millions)
funds <- str_extract(data$costs, "[0-9]{1,4}")
funds <- funds[!is.na(funds)]
funds = as.double(funds)
mean_f = mean(funds)
#Population #(in thousands)
pop <- str_extract(data$populations, "[0-9]{1,7}")
pop <- pop[!is.na(pop)]
pop <- as.double(pop)
mean_p <- mean(pop)

#Wrapping it Up
dollar.per.user = mean_f /mean_p
dollar.per.user
```

```
## [1] 1.049278
```

The average community network costs about 1,0049 per person! Below, we can find the maximum lifespan of one of these networks.

```
year <- c(str_extract(data$year, "[0-9]{4}"))
year <- year[!is.na(year)]
year <- as.integer(year)
print("Minimum:")
```

```
## [1] "Minimum:"
```

```
min(year)
```

```
## [1] 2002
```

```
print("Summary:")
```

```
## [1] "Summary:"
```

```
summary(year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2002    2006    2009    2009    2012    2016
```

```
data[22,]
```

```
##                       networks        communities years populations
## 22 Marshall Municipal Utilities Marshall, Missouri  2002      13,000
##               costs  funding.methods           governances        Service1
## 22 <em>Unknown</em> <em>Unknown</em> Board of Public Works Internet access
##    Service2 Service3          speeds
## 22     <NA>     <NA> 90 Mbps/45 Mbps
```

The oldest network on the list is the Marshall, Missouri Municipal Utilities Corporation. With a coaxial network to each home (reflected in the 90/45 transfer rate), this network not only surpasses many places in the world, but it is self-managed by the city. It is not as fast as a brand-new fiber network, but easily serves Netflix to a household at peak times. If we amortize these costs over 16 years, we can find our cost per person per year. Even if we look at the average lifespan, we see that the cost is only $116/user/year.

```
dollar.per.user.per.year = dollar.per.user/(2018 - 2002)
dollar.per.user.per.year * 1000 # converting 10^6 dollars/10^3 people
```

```
## [1] 65.57986
```

```
less.optimistic = dollar.per.user/(2018-2009) *1000
less.optimistic
```

```
## [1] 116.5864
```

With reasonable optimism, we see that municipal a network costs about $65/user/year. The only other concern would be the missing data from the the cost and population datasets. These may be outliers in actuality, which could dramatically shift this average.

Mac Addresses

Finally, I wanted to be able to track the types of devices so that I can do more deep network intelligence. First, I have to load the dataset from the IEEE (available as a .txt).

```
data.file3 <- curl_download("http://standards-oui.ieee.org/oui.txt", "MACs/IEEE-MACs.txt")
data.file3
```

```
## [1] "/home/nologs/Mesh-Data/data/Global/MACs/IEEE-MACs.txt"
```

So, I parsed long data, converting it into wide data. Because the delimiter isn't constant, tidyr doesn't help much herre.

```
lines <- readLines(data.file3)
#head(lines)

first.chunk <- which(grepl("[0-9A-F]{2}-[0-9A-F]{2}-[0-9A-F]{2}", lines))
lines1 = lines[first.chunk]
lines2 = str_extract_all(lines[first.chunk+2], "(?<=\\t\\t\\t\\t)(.*)")
lines3 = str_extract_all(lines[first.chunk+3], "(?<=\\t\\t\\t\\t)(.*)")
lines4 = lines[first.chunk+4]
```

Then I had to parse each of these lines and extract their data points. I assigned each one of these to a vector corresponding to to row. Then, I bound all the data together and wrote it to a csv.

```
MACs <- c(str_extract(lines1, "[0-9A-F]{2}-[0-9A-F]{2}-[0-9A-F]{2}"))
Manufacturers <- c(str_extract(lines1, "(?<=\\t\\t)(.*)"))
Addresses <- c(lines2)
Zips <- c(str_extract(lines3, "[0-9]{5}"))
```

```
## Warning in stri_extract_first_regex(string, pattern, opts_regex =
## opts(pattern)): argument is not an atomic vector; coercing
```

```
Region <- c(str_extract(lines3, "([^[0-9]{5}]+)"))
```

```
## Warning in stri_extract_first_regex(string, pattern, opts_regex =
## opts(pattern)): argument is not an atomic vector; coercing
```

```
Country <- c(str_extract(lines4, "[:alpha:]{2}"))
data <- (cbind(MACs, Manufacturers, Addresses, Zips, Region, Country))
head(data)
```

```
##      MACs       Manufacturers
## [1,] "E0-43-DB" "Shenzhen ViewAt Technology Co.,Ltd. "
## [2,] "24-05-F5" "Integrated Device Technology (Malaysia) Sdn. Bhd."
## [3,] "3C-D9-2B" "Hewlett Packard"
## [4,] "9C-8E-99" "Hewlett Packard"
## [5,] "B4-99-BA" "Hewlett Packard"
## [6,] "1C-C1-DE" "Hewlett Packard"
##      Addresses
## [1,] "9A,Microprofit,6th Gaoxin South Road, High-Tech Industrial Park, Nanshan, Shenzhen, CHINA."
## [2,] "Phase 3, Bayan Lepas FIZ"
## [3,] "11445 Compaq Center Drive"
## [4,] "11445 Compaq Center Drive"
## [5,] "11445 Compaq Center Drive"
## [6,] "11445 Compaq Center Drive"
##      Zips    Region                  Country
## [1,] "51805" "shenzhen  guangdong  " "CN"
## [2,] "11900" "Bayan Lepas  Penang  " "MY"
## [3,] "77070" "Houston    "           "US"
## [4,] "77070" "Houston    "           "US"
## [5,] "77070" "Houston    "           "US"
## [6,] "77070" "Houston    "           "US"
```

```
write.csv(data, file = "IEEE-MACs.csv")
```

I'm repeating the same geographic analysis as above, but this time looking at countries where hardware is produced.

```
per_country = data.frame()
data <- data.frame(Country)
country.list <- unique(trimws(data$Country))
for (country in country.list){
 number <- dim(subset(data, Country == country))[[1]]
 new.row <- cbind(country, number)
 per_country <- rbind(per_country, new.row)
}
per_country <- data.frame(per_country)
arrange(per_country, number)
```

```
##     country number
```

```
## 1        CN   3690
## 2        MY    306
## 3        US   9254
## 4        DE   1195
## 5        TW   1824
## 6        SG    140
## 7        FR    493
## 8        DK    234
## 9        IT    297
## 10       FI    257
## 11       JP   1566
## 12       KR   2044
## 13       AU    202
## 14       DC      3
## 15       FE      3
## 16       RO      3
## 17       CY      3
## 18       IR      3
## 19       CE      3
## 20       RS      3
## 21       YU      3
## 22       CC      3
## 23       NL    179
## 24       GB    702
## 25       IN    132
## 26     <NA>      0
## 27       RU    135
## 28       SE    242
## 29       ba     59
## 30       HK    275
## 31       LV     13
## 32       NO     78
## 33       AR      9
## 34       LT      9
## 35       GR      9
## 36       TH     49
## 37       BR    111
## 38       BE    102
## 39       CH    225
## 40       CZ     44
## 41       NZ     44
## 42       MX     23
## 43       AT     98
## 44       BB      2
## 45       UA      2
## 46       EG      2
## 47       KW      2
## 48       DA      2
## 49       BF      2
## 50       EE      2
## 51       MT      2
## 52       GE      2
## 53       FA      2
## 54       MD      2
```

```
## 55       IS      2
## 56       BC      2
## 57       KZ      2
## 58       FC      2
## 59       CO      2
## 60       KP      2
## 61       CA    603
## 62       IL    260
## 63       QA      1
## 64       LB      1
## 65       LI      1
## 66       KM      1
## 67       MM      1
## 68       CB      1
## 69       MA      1
## 70       BN      1
## 71       BY      1
## 72       VU      1
## 73       EA      1
## 74       DZ      1
## 75       BS      1
## 76       VI      1
## 77       PE      1
## 78       MU      1
## 79       ED      1
## 80       VE      1
## 81       CF      1
## 82       BD      1
## 83       DB      1
## 84       CL      1
## 85       AA      1
## 86       BA      1
## 87       PT     15
## 88       ID     15
## 89       PL     50
## 90       SK     19
## 91       ES    130
## 92       he      4
## 93       LU      4
## 94       TJ      4
## 95       SI     22
## 96       TR     40
## 97       KY      5
## 98       HR      5
## 99       PH      5
## 100      BG     12
## 101      IE     30
## 102      ZA     47
## 103      HU     24
## 104      AE      7
## 105      VN      7
## 106      JO      6
## 107      VG      6
```

We can see from this data that the US has 3 times as many networked devices manufacturers than China, despite other assumptions. Additionally, both Germany and Taiwan have significant investments in this field. Below are some dependencies for a map.

```
#install.packages("countrycode")
#install.packages("rworldmap")
suppressMessages(library(countrycode))
suppressMessages(library(rworldmap))
```

```
full.name <- countrycode(per_country$country, "iso2c", "country.name", nomatch = NULL )
```

```
## Warning in countrycode(per_country$country, "iso2c", "country.name", nomatch = NULL): The origin and
##                       class. Filling-in bad matches with NA instead.
```

```
## Warning in countrycode(per_country$country, "iso2c", "country.name", nomatch = NULL): Some values we
```

```
per_country <- cbind(per_country, full.name)
per_country
```

```
##     country number               full.name
## 1        CN   3690                   China
## 2        MY    306                Malaysia
## 3        US   9254           United States
## 4        DE   1195                 Germany
## 5        TW   1824                  Taiwan
## 6        SG    140               Singapore
## 7        FR    493                  France
## 8        DK    234                 Denmark
## 9        IT    297                   Italy
## 10       FI    257                 Finland
## 11       JP   1566                   Japan
## 12       KR   2044             South Korea
## 13       AU    202               Australia
## 14       DC      3                    <NA>
## 15       NL    179             Netherlands
## 16       GB    702          United Kingdom
## 17       IN    132                   India
## 18     <NA>      0                    <NA>
## 19       RU    135                  Russia
## 20       SE    242                  Sweden
## 21       ba     59                    <NA>
## 22       HK    275      Hong Kong SAR China
## 23       LV     13                  Latvia
## 24       NO     78                  Norway
## 25       AR      9               Argentina
## 26       TH     49                Thailand
## 27       FE      3                    <NA>
## 28       BR    111                  Brazil
## 29       BE    102                 Belgium
## 30       CH    225             Switzerland
## 31       CZ     44                 Czechia
## 32       MX     23                  Mexico
```

```
## 33       AT     98                 Austria
## 34       BB      2                Barbados
## 35       CA    603                  Canada
## 36       RO      3                 Romania
## 37       IL    260                  Israel
## 38       QA      1                   Qatar
## 39       PT     15                Portugal
## 40       PL     50                  Poland
## 41       LT      9               Lithuania
## 42       SK     19                Slovakia
## 43       ES    130                   Spain
## 44       he      4                    <NA>
## 45       SI     22                Slovenia
## 46       NZ     44             New Zealand
## 47       TR     40                  Turkey
## 48       ID     15               Indonesia
## 49       KY      5          Cayman Islands
## 50       BG     12                Bulgaria
## 51       IE     30                 Ireland
## 52       ZA     47            South Africa
## 53       HU     24                 Hungary
## 54       CY      3                  Cyprus
## 55       IR      3                    Iran
## 56       LB      1                 Lebanon
## 57       HR      5                 Croatia
## 58       UA      2                 Ukraine
## 59       AE      7    United Arab Emirates
## 60       EG      2                   Egypt
## 61       GR      9                  Greece
## 62       LI      1           Liechtenstein
## 63       KM      1                 Comoros
## 64       KW      2                  Kuwait
## 65       JO      6                  Jordan
## 66       VG      6  British Virgin Islands
## 67       MM      1          Myanmar (Burma)
## 68       CE      3                    <NA>
## 69       VN      7                 Vietnam
## 70       CB      1                    <NA>
## 71       DA      2                    <NA>
## 72       MA      1                 Morocco
## 73       PH      5             Philippines
## 74       RS      3                  Serbia
## 75       BF      2            Burkina Faso
## 76       BN      1                  Brunei
## 77       EE      2                 Estonia
## 78       YU      3                    <NA>
## 79       BY      1                 Belarus
## 80       MT      2                   Malta
## 81       LU      4              Luxembourg
## 82       TJ      4              Tajikistan
## 83       VU      1                 Vanuatu
## 84       GE      2                 Georgia
## 85       CC      3 Cocos (Keeling) Islands
## 86       FA      2                    <NA>
```

```
## 87       EA      1                    <NA>
## 88       MD      2                 Moldova
## 89       DZ      1                 Algeria
## 90       BS      1                 Bahamas
## 91       IS      2                 Iceland
## 92       VI      1      U.S. Virgin Islands
## 93       PE      1                    Peru
## 94       BC      2                    <NA>
## 95       KZ      2              Kazakhstan
## 96       FC      2                    <NA>
## 97       CO      2                Colombia
## 98       MU      1               Mauritius
## 99       KP      2             North Korea
## 100      ED      1                    <NA>
## 101      VE      1               Venezuela
## 102      CF      1 Central African Republic
## 103      BD      1              Bangladesh
## 104      DB      1                    <NA>
## 105      CL      1                   Chile
## 106      AA      1                    <NA>
## 107      BA      1     Bosnia & Herzegovina
```

```
ieee.manufacturers.per.country <- joinCountryData2Map(per_country, joinCode = "ISO2", nameJoinColumn =
```

```
## 90 codes from your data successfully matched countries in the map
## 17 codes from your data failed to match with a country code in the map
## 152 codes from the map weren't represented in your data
```

```
mapCountryData(ieee.manufacturers.per.country, nameColumnToPlot = "number", mapTitle ="Distinct IEEE Ma
```

```
## using catMethod='categorical' for non numeric data in mapCountryData
```

Distinct IEEE Manufacturers per Country