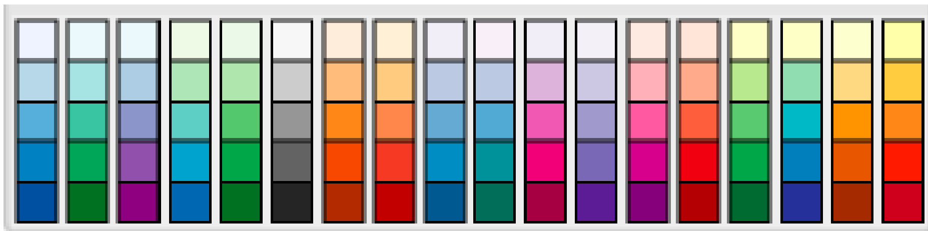
Final

1) Color Pallets

When using color in visualizations there are different kinds of pallets that can be used.

a. Pallet 1

Describe the pallet and explain when you would use such a pallet.



The palette above is an example of sequential color schemes using multi-hued colors.

Each scheme is sequential because it allows the highlighting of order in data through the use of shading. Within a given hue such as blue, there are multiple shades from light to dark. The lightness or darkness of the hue can be used to represent different levels of values. Typically, light hues represent low values and dark hues represent high values.

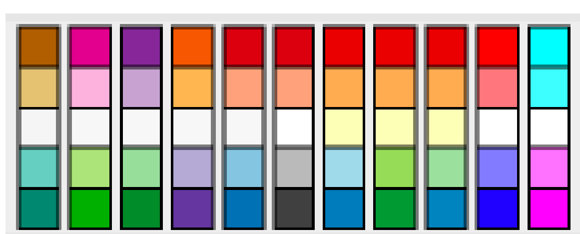
Each scheme is multi-hued meaning the colors used are not just shades of a single hue but instead, use multiple hues. When using multiple hues, the palettes still provide a pleasing aethetic transition from lighter shades to darker shades that preserves the implied sequential meaning of the collective colors.

Sequential colors schemes are suited to highlighted data that can be categorized into ordered categories. Examples could be age ranges, levels of experience, density ranges, etc.

One vizualiation that may beneit from this color scheme, is a tile plot. One of the main features of a heat map, is its use of color to distinguish ordinal categorical variables, in a spacial setting. With a sequential color scheme as such, I would apply this theme to a heat map, in order to provide a well read, distinguishable, vizualization.

b. Pallet 2

Describe the pallet and explain when you would use such a pallet.



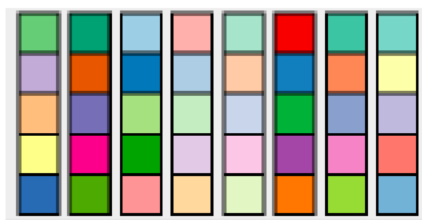
The palette above is an example of diverging color schemes. Each scheme is diverging because it allows the highlighting of both central and extreme values in underlying data. Lighter shades and hues are used as the central colors in each of these palettes. In the palettes above, each has five (5) colors with the third (3) color being the central, lightest color. Moving away from this color in either direction towards the first and last colors in the palette, the shades and hues get darker. The colors diverge away from a light, neutral color towards darker, more distinct colors. The colors at the ends of the palettes typically contrast highly from each other to help amplify the meaning of the divergence in the underlying data away from its central values.

Diverging colors schemes are suited to highlight the central and extreme value in data distributions. The light coloring of central values tends to indicate the typical values of data while the bold, contrasting coloring of extreme values tends to highlight these extreme values. Examples could be grade distributions, income level distributions, age ranges, etc.

Note that many datasets could be highlighted by either sequential or diverging schemes. For example, age ranges could be highlighted by either. However, the intent of the visualization would help dictate which to use. Consider a question posed such as "Comparing pre-teen, teen, adult, and eldery populations. . . ?" Now consider a second, similar question posed such as "What are the average ages. . . ?" The first question is being posed from a categorial perspective that implies a sequence tied to human lifecycles. There is implied interest in order so a sequential color scheme would be applicable. For the second question, there was not much emphasis on any difference between young or old but instead more interest in the distribution, the *average*. In this case, a diverging color scheme may be more suited to the vizualization to not only highlight the average (central valeus) but also highlight the extremes.

c. Pallet 3

Describe the pallet and explain when you would use such a pallet.



The palette above is an example of qualitative color schemes. Each scheme is designed with a set of color shades and hues that contrast from one another. Sequential and diverging color schemes do not try to contrast as much but instead try to show more relationship or transitioning of values between each color. Qualitative schemes try to show the contrast as much as possible attempting to highlighted the grouping and differences more than the similarities or nearness to other groups.

Qualitative color schemes are best used when tyring to depict different categories of data that are more distinct from each other than they are as similar or close to one another. Examples include demographic data such as racial identity, gender identity, political affiliation, religious affiliation, sports team fan affiliation, etc.

Some vizualization to which this color scheme could be applied, include pie charts, waffle plots, donut plots, and ring plots.

2) Earthquakes

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.5.3

## -- Attaching packages ---------------------------------------------------------------------------

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.5.3

## Warning: package 'tibble' was built under R version 3.5.3

## Warning: package 'tidyr' was built under R version 3.5.3

## Warning: package 'readr' was built under R version 3.5.3

## Warning: package 'purrr' was built under R version 3.5.3

## Warning: package 'dplyr' was built under R version 3.5.3

## Warning: package 'stringr' was built under R version 3.5.3

## Warning: package 'forcats' was built under R version 3.5.3

## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(ggmap)
```

```
## Warning: package 'ggmap' was built under R version 3.5.3

## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.

## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```r
library(WHO)
```

```
## Warning: package 'WHO' was built under R version 3.5.3
```

```r
library(stringr)
library(waffle)
```

```
## Warning: package 'waffle' was built under R version 3.5.3
```

Here is the link to the USGS website where the worldwide earthquake data can be downloaded. Download all earthquake data for the past 30 days in .csv format. Using R, make a map of the world with points where the earthquakes occurred. Make a bubble map using the magnitude. Thoroughly discuss your visualizations.

World Map of Points

As the earthquake data of the past 30 days is updated every minute, it is important to specify that the earthquake data was downloaded at 1:08 p.m., on November 18, 2019.

To begin the map, I will read in the earthquake dataset under the variable name: "earthquake."

```r
earthquake <- read_csv("earthquake.csv", col_names = TRUE)
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   time = col_datetime(format = ""),
##   magType = col_character(),
##   net = col_character(),
##   id = col_character(),
##   updated = col_datetime(format = ""),
##   place = col_character(),
##   type = col_character(),
##   status = col_character(),
##   locationSource = col_character(),
##   magSource = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

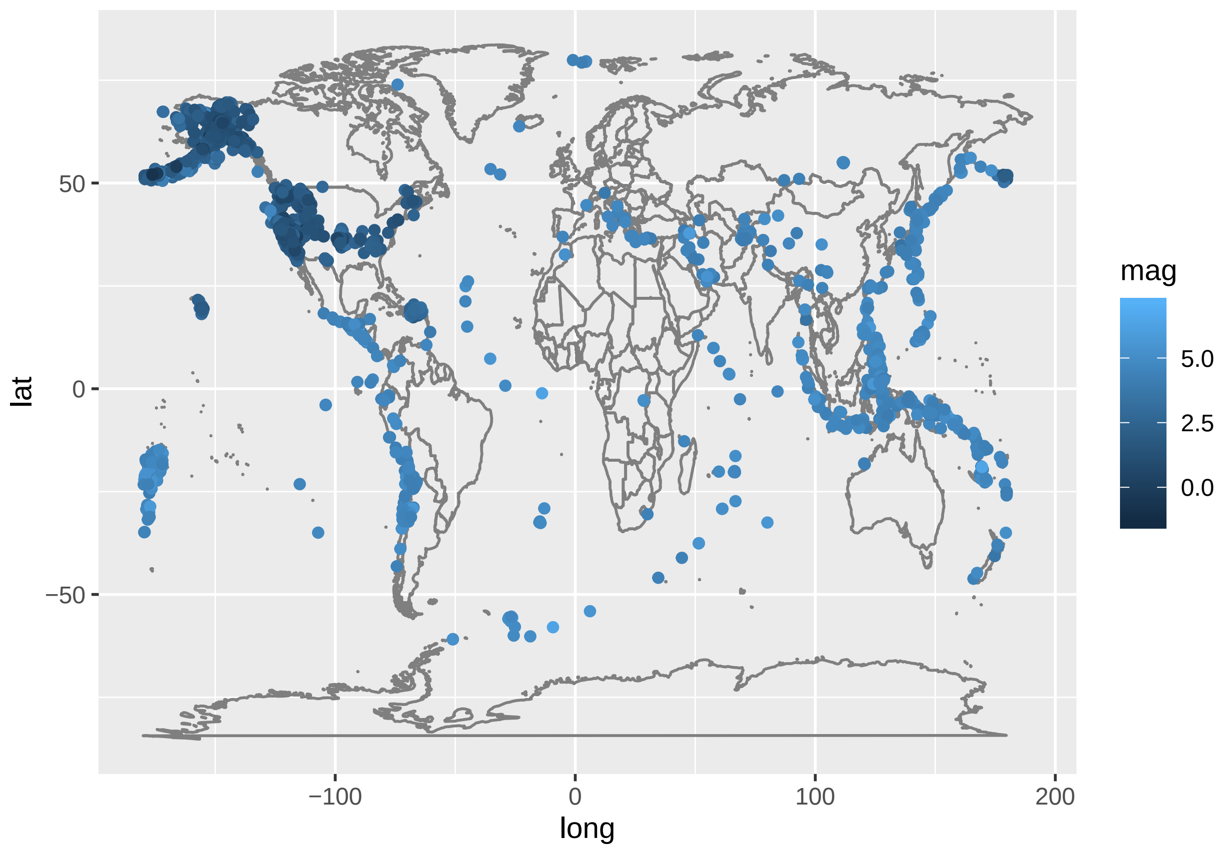
```r
head(earthquake)
```

```
## # A tibble: 6 x 22
##   time                latitude longitude  depth   mag magType   nst   gap
##   <dttm>                 <dbl>     <dbl>  <dbl> <dbl> <chr>   <dbl> <dbl>
## 1 2019-11-18 21:01:46     35.8     -119.   8.82  1.14 ml         14    88
## 2 2019-11-18 20:54:46     61.4     -150.  30.4   1.7  ml         NA    NA
## 3 2019-11-18 20:42:44     33.6     -117.  13.4   0.33 ml         17    70
## 4 2019-11-18 20:34:00     33.9     -117.  19.1   0.87 ml         25    67
## 5 2019-11-18 20:32:41     34.4     -118.  11.2   0.83 ml         12   105
## 6 2019-11-18 20:30:32     61.3     -148. 130.    1.6  ml         NA    NA
## # ... with 14 more variables: dmin <dbl>, rms <dbl>, net <chr>, id <chr>,
## #   updated <dttm>, place <chr>, type <chr>, horizontalError <dbl>,
## #   depthError <dbl>, magError <dbl>, magNst <dbl>, status <chr>,
## #   locationSource <chr>, magSource <chr>
```

Variable	Description
time	Time of Earthquake occurence
latitude	Latitude Location of Earthquake
longitude	Longitude location of Earthquake
depth	Depth of the Event
mag	Magnitude of Event
magType	Algorithm or Method Used to Evaluate the Method of the Earthquake
nst	Number of Seismic Stations used to evaluate Eathquake Location
gap	The Largest azimutahl gap between azimuthally adjacent stations (in degrees)
horizontalError	Uncertainty of Observed Event's Location (in KM)
dmin	Smallest observed Distance to event epicenter from the Closest Seismic Station
rms	Root Mean Square Calculations of Residuals in predictions of Event occurence.
net	ID of Data Contributer
id	Unique Identification of Eathquake

Variable	Description
updated	Time of Upload in Original Dataset
place	Nearby Named Geographical Region
horizontalError	Uncertainty of Earthquake Location (in KM)
depthError	Uncertainty of Earthquake Depth (in KM)
magNst	Total number of Seismic Stations used to Calculate Earthquake's Magnitude
Status	Indicates Whether Event has been viewed by a Person
locationSource	Network that Authored location of Event
magSource	Network that Authored Preffered Magnitude

```r
ggplot(data = earthquake) +
  borders("world") +
  geom_point(mapping = aes(x = longitude, y = latitude, color = mag))
```



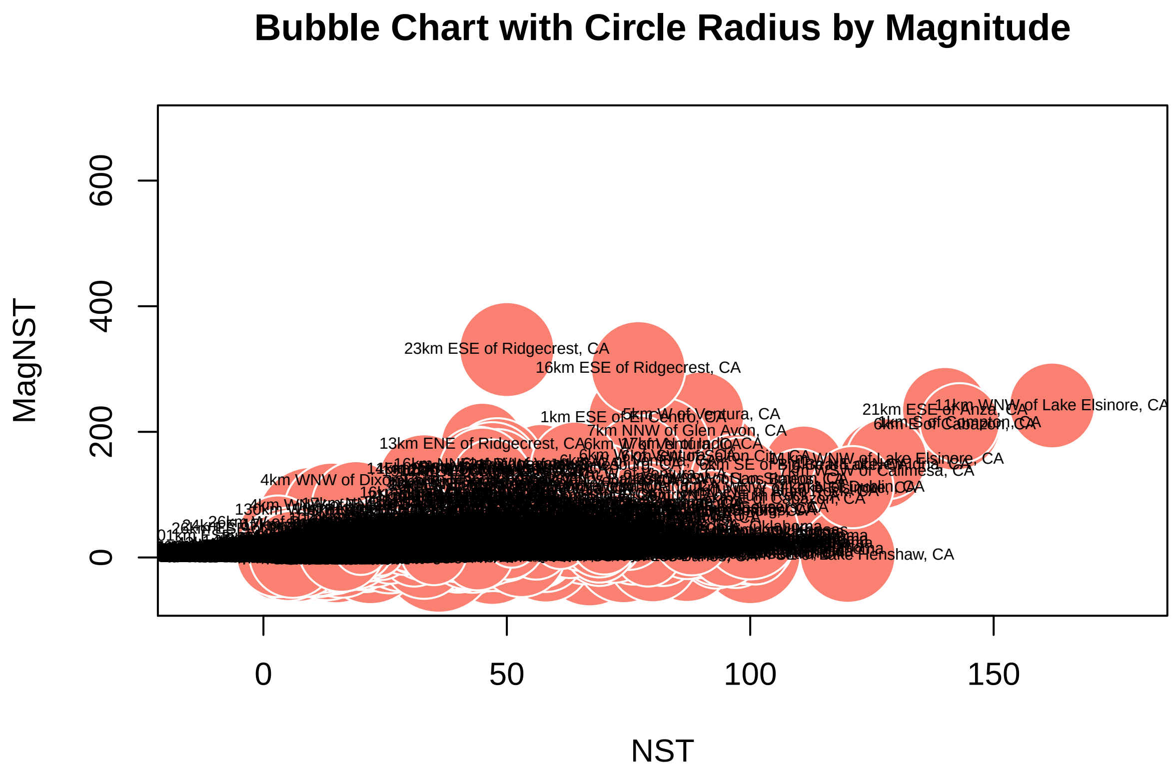
Analysis: As evident from the world map, it appears that the west coast of the United States has had the highest amount of recorded earthquakes in the past 30 days. Other frequent earthquakes sites include Japan, Malaysia, Alaska, and the Phillipines. While the West coast of the United States appears to have the biggest cluster of earthquakes, the continent and shoreline of Asia appears to have the most earthquakes with the highest magnitude.

Bubble Map of Earthquakes by Magnitude (With Labels)

```
radius <- sqrt(earthquake$mag/ pi )
```

```
## Warning in sqrt(earthquake$mag/pi): NaNs produced
```

```
symbols(earthquake$nst, earthquake$magNst, circles = radius, inches = .35, fg = "white", bg = "salmon",
text(earthquake$nst, earthquake$magNst, earthquake$place, cex = .5)
```

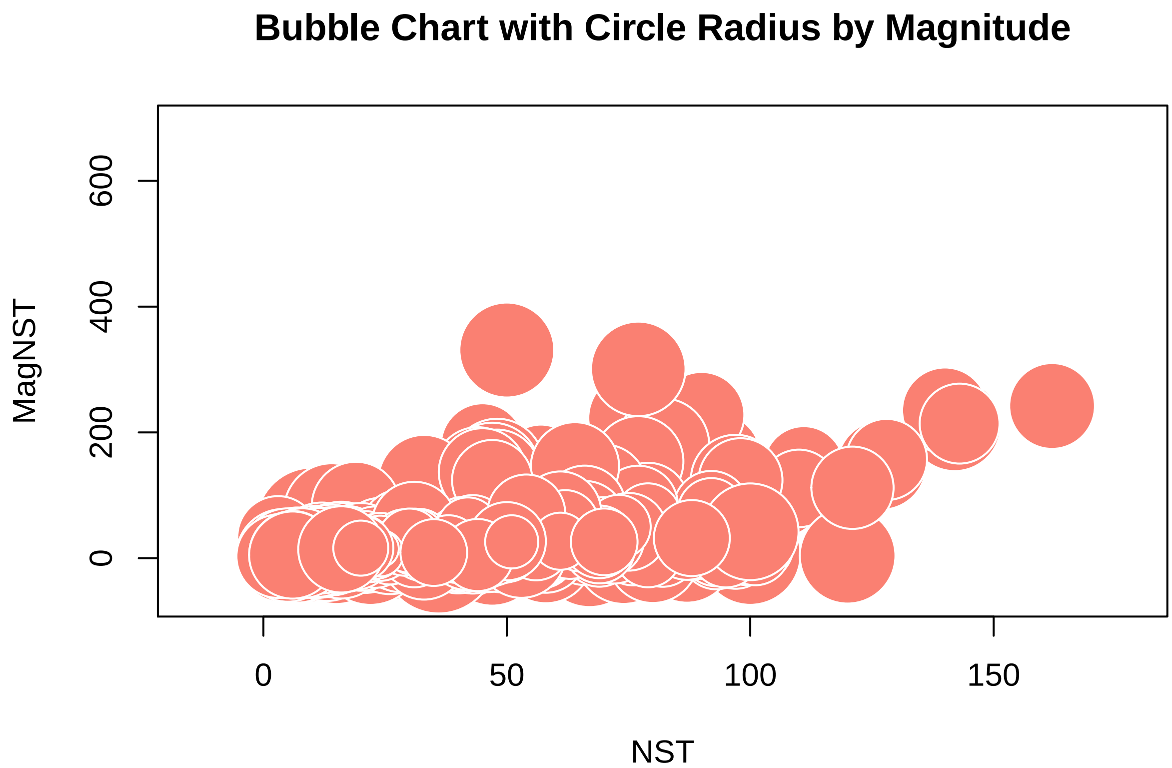


Analysis: For the Bubble Chart, we decided to utilize the variables NST, MAgNST, and the Magnitude. The x-axis represents the amount of seismic sensors used to detect the location, and the y-axis represents the number of seismic sensors used to calculate the magnitude. The diameter of each circle, is based off of the Magnitude for each indivudal observation. As evident from the bubble chart, it appeards that there are more Location seismic sensors used than sensors used to calculate magnitude. In addition, one can tell from the distribuion of the larger circles, that the number of seismic sensors used has no effect on evaluating the magnitude of a quake.

As the clustering of the circle's labels leads to a very distracting plot, the bubble plot has been plotted again, without the text labels.

Bubble Plot of Earthquakes by Magnitude (Without Labels)

```
symbols(earthquake$nst, earthquake$magNst, circles = radius, inches = .35, fg = "white", bg = "salmon",
```


Bubble Chart with Circle Radius by Magnitude
MagNST
0
200
400
600
0
50
100
150
NST

TO DO

3) Disease / Illness Story

See `Final.pdf` in this project's Files section for detailed instructions.

First, I will read in the "CHOLERA" dataset from the "WHO" metadata.

```r
gho_vectorize <- function(v) { ### To create a clean, factored vector
  switch(class(v),
         "character" = {
           f <- as_factor(v)
           fct_explicit_na(fct_relevel(f,
                                       sort(levels(f))),
                           na_level = "NA")
         },
         "factor" = {
           fct_explicit_na(fct_relevel(v,
                                       sort(levels(v))),
                           na_level = "NA")
         },
         "numeric" = as_factor(v))
}




tb_cholera <- get_data("CHOLERA_0000000001")
tb_cholera$country <- gho_vectorize(tb_cholera$country)
tb_cholera$region <- gho_vectorize(tb_cholera$region)
tb_cholera$publishstate <- gho_vectorize(tb_cholera$publishstate)

tb_cholera <- tb_cholera %>%
  group_by(country) %>%
  arrange(country, year) %>%
  select(country, year, value, worldbankincomegroup, region) %>% rename("cases" = value)
tb_cholera
```

```
## # A tibble: 2,480 x 5
## # Groups:   country [162]
##    country      year cases worldbankincomegroup region
##    <fct>       <dbl> <dbl> <chr>                <fct>
##  1 Afghanistan  1960   887 <NA>                 Eastern Mediterranean
##  2 Afghanistan  1965   218 <NA>                 Eastern Mediterranean
##  3 Afghanistan  1993 37046 <NA>                 Eastern Mediterranean
##  4 Afghanistan  1994 38735 <NA>                 Eastern Mediterranean
##  5 Afghanistan  1995 19903 <NA>                 Eastern Mediterranean
##  6 Afghanistan  1997  4170 <NA>                 Eastern Mediterranean
##  7 Afghanistan  1998 10000 <NA>                 Eastern Mediterranean
##  8 Afghanistan  1999 24639 <NA>                 Eastern Mediterranean
##  9 Afghanistan  2000  4330 <NA>                 Eastern Mediterranean
## 10 Afghanistan  2001  4499 <NA>                 Eastern Mediterranean
## # ... with 2,470 more rows
```

Next, with the dataset successfully read in, I will now add to the "WSH_10" dataset. The "WSH_10" dataset, records the number of deaths related to improper water sources. My goal, is to knit these to datasets by their common variables, and gain an inference on how the cases of Cholera in a country effect the number of deaths related to contaminated water intake. However, as the WSH-10 dataset only contains values for, 2016, I will only filter out the observations from the year 2016.

```
d_table <- get_data("WSH_10")
viz_data <- tb_cholera %>% left_join(d_table) %>% filter( year == 2016) %>% rename('deaths' = value)
```

```
## Joining, by = c("country", "year", "region")
```

```
## Warning: Column `country` joining factor and character vector, coercing into
## character vector
```

```
## Warning: Column `region` joining factor and character vector, coercing into
## character vector
```

```
viz_data
```

```
## # A tibble: 152 x 11
## # Groups:   country [37]
##    country  year cases worldbankincome~ region sex   inadequacy agegroup gho
##    <chr>   <dbl> <dbl> <chr>            <chr>  <chr> <chr>      <chr>    <chr>
##  1 Afghan~  2016   677 <NA>             Easte~ Fema~ Inadequat~ All age~ Numb~
##  2 Afghan~  2016   677 <NA>             Easte~ Both~ Inadequat~ All age~ Numb~
##  3 Afghan~  2016   677 <NA>             Easte~ Male  Inadequat~ All age~ Numb~
##  4 Afghan~  2016   677 <NA>             Easte~ Both~ Inadequat~ < 5 yea~ Numb~
##  5 Angola   2016    78 <NA>             Africa Male  Inadequat~ All age~ Numb~
##  6 Angola   2016    78 <NA>             Africa Both~ Inadequat~ All age~ Numb~
##  7 Angola   2016    78 <NA>             Africa Fema~ Inadequat~ All age~ Numb~
##  8 Angola   2016    78 <NA>             Africa Both~ Inadequat~ < 5 yea~ Numb~
##  9 Austra~  2016     1 <NA>             Weste~ Both~ Inadequat~ All age~ Numb~
## 10 Austra~  2016     1 <NA>             Weste~ Male  Inadequat~ All age~ Numb~
## # ... with 142 more rows, and 2 more variables: publishstate <chr>,
## #   deaths <chr>
```

As evident from the code output, the data has successfuly been read in However, one column in particular is in need of tidying: deaths. unfortunately, this column was read in as a character vecotr. to accomplish this, I will first remove the extra, none-numerical values. Then, with the columns tidied up, I will now use the "dplyr" package in order to "parse" the columns into their proper forms.

```
viz_data$deaths <- viz_data$deaths%>% str_remove(pattern = "[:space:]\\[[:digit:]+-[:digit:]+\\]") %>%
viz_data$deaths
```

```
##   [1]    2489   4703   2214   4395   6976  13274   6298   7759     23      9
##  [11]      14      1   2697   5985   3289   2843   3048   6369   3321   1939
##  [21]     133     52     81      0   1844   1749   3593   1876    818   1050
##  [31]    1867    450  42621  17720  24901  23139      6     12     17      0
##  [41]     224    112    112    101    298    183    480      1   2277   2509
##  [51]    4786   2254   1392   2436   1045   1146 243551 108044 135507  61126
##  [61]    1109    697    412    883   1109    697    412    883     87    213
##  [71]     126      4  11826  12366  24192   3281   2401   2349   4750   1493
##  [81]    3772   3765   7537   3962   3203   6035   2833   1991   5711   3625
```

```
##  [91]   2087    665     16     25     41      0   7236   6393  13628   5656
## [101]  63737  56142 119879  50243   1744   2369   4113   2305     64     43
## [111]     21      1   2164   1039   1125    502   6151   5605  11756   7126
## [121]   3603   7510   3907   2434    946   1895    949     78   6042   6371
## [131]  12414   4604    130     83     47      1  10841  20043   9203   6132
## [141]    303    443    746     29   1194   2779   1585   2197   3472   1855
## [151]   1617   1636
```

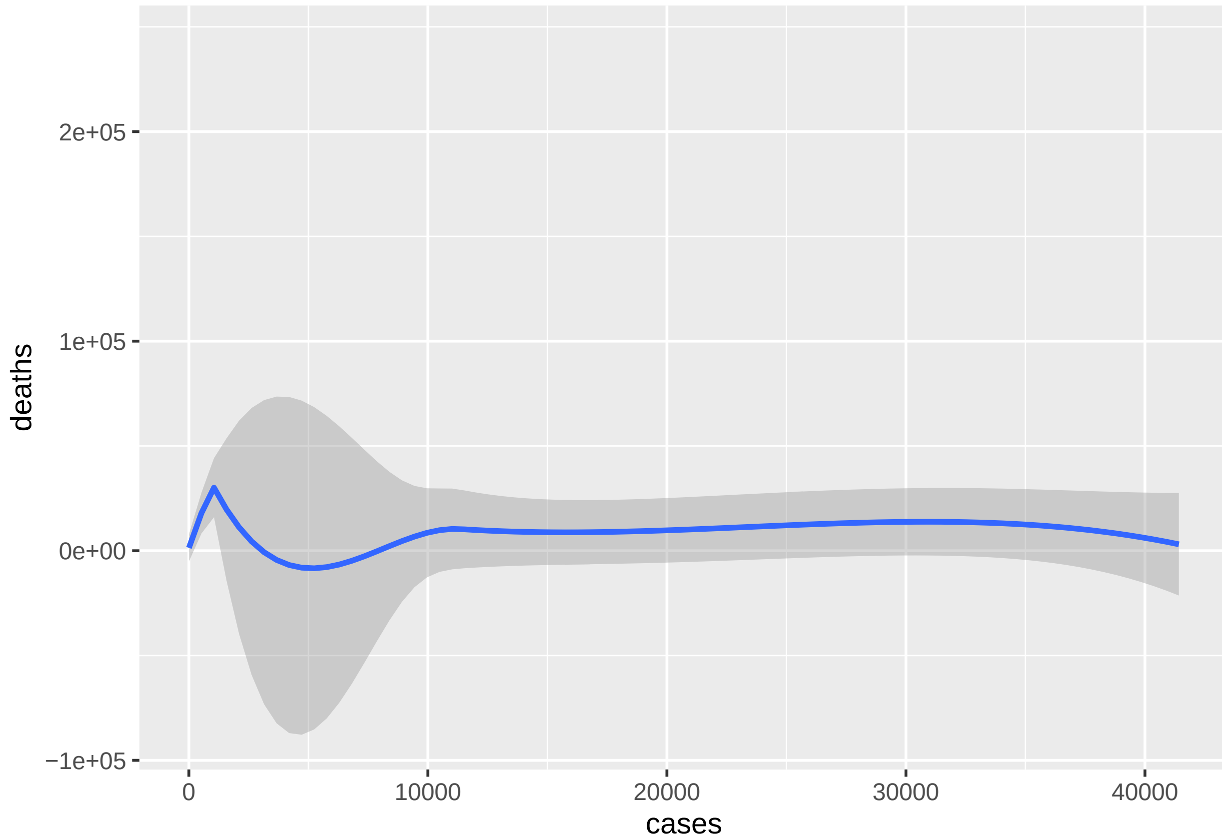
```
viz_data <- viz_data %>% select(-gho, -worldbankincomegroup)
viz_data
```

```
## # A tibble: 152 x 9
## # Groups:   country [37]
##    country   year cases region  sex   inadequacy   agegroup publishstate deaths
##    <chr>    <dbl> <dbl> <chr>   <chr> <chr>        <chr>    <chr>         <int>
##  1 Afghani~  2016   677 Easter~ Fema~ Inadequate ~ All age ~ Published     2489
##  2 Afghani~  2016   677 Easter~ Both~ Inadequate ~ All age ~ Published     4703
##  3 Afghani~  2016   677 Easter~ Male  Inadequate ~ All age ~ Published     2214
##  4 Afghani~  2016   677 Easter~ Both~ Inadequate ~ < 5 years Published     4395
##  5 Angola    2016    78 Africa  Male  Inadequate ~ All age ~ Published     6976
##  6 Angola    2016    78 Africa  Both~ Inadequate ~ All age ~ Published    13274
##  7 Angola    2016    78 Africa  Fema~ Inadequate ~ All age ~ Published     6298
##  8 Angola    2016    78 Africa  Both~ Inadequate ~ < 5 years Published     7759
##  9 Austral~  2016     1 Wester~ Both~ Inadequate ~ All age ~ Published       23
## 10 Austral~  2016     1 Wester~ Male  Inadequate ~ All age ~ Published        9
## # ... with 142 more rows
```

With the dataset properly transformed, I will now utilize the package "ggplot" in order to create vizualizations of the refined dataset.

Scatterplot With Refined Smoother of Cases of Cholera by Deaths from Improper Water

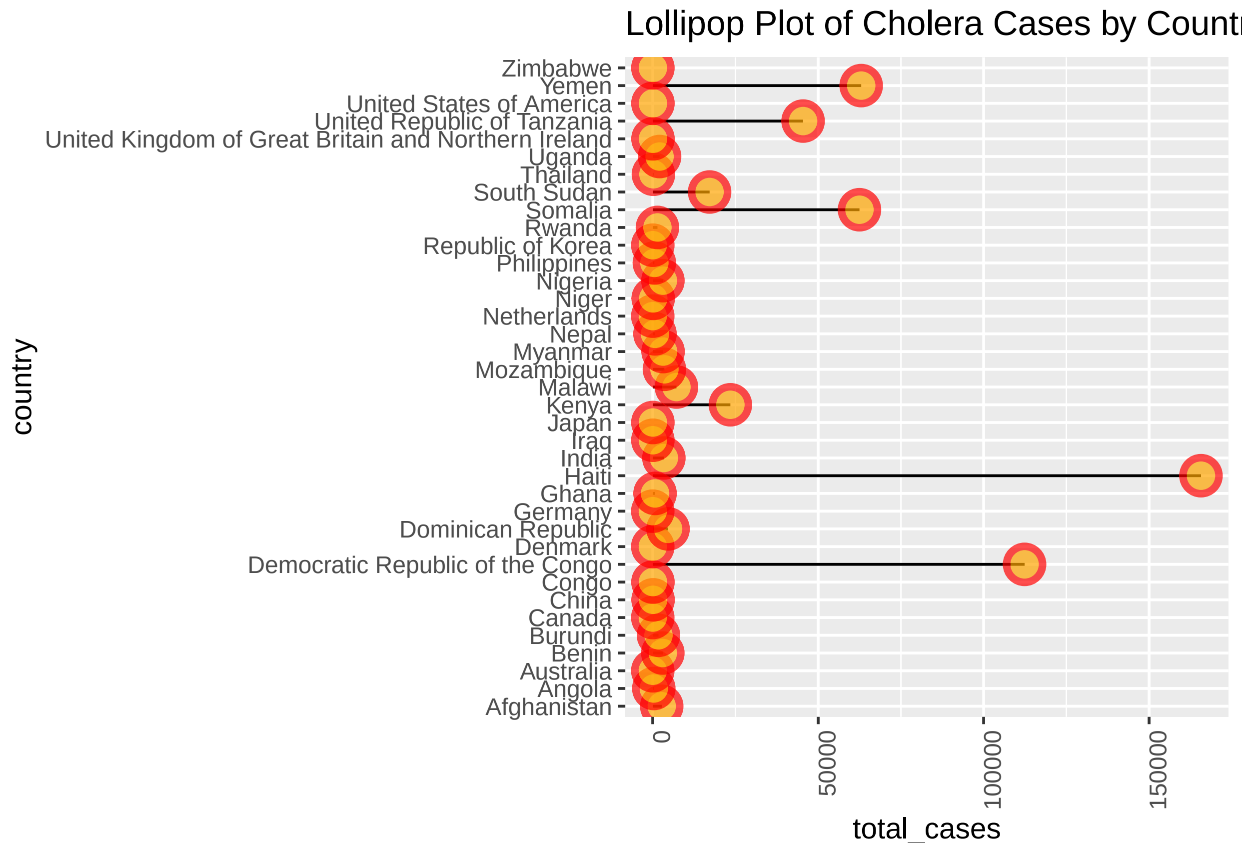
```
viz_data %>%
ggplot() +
  geom_point(mapping = aes(x = cases, y = deaths), color = 21) +
  geom_smooth(mapping = aes(x = cases, y = deaths), method = "loess")
```



Analysis: As evident from the smoothed scatterplot, it appears that as the number of cases of Cholera increases, the number of deaths from a basic water source remains relatively constant. To me, this is very odd, as I was expecting a very positive linear trend in the dataset, suggesting a strong correlation. However, it appears that the amount of Cholera cases is at a constant trend with the amount of deaths in a country.

Lollipop Plot:

```
viz_data %>% select(country, cases) %>% group_by(country) %>% summarise(total_cases = sum(cases, na.rm =
ggplot(mapping = aes(x = country, y = total_cases)) +
  geom_segment(mapping = aes(x = country, xend = country, y = 0, yend = total_cases)) +
  geom_point(size = 5, color = "red", fill = alpha("orange", .3), alpha = .7, shape = 21, stroke = 2) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  ggtitle("Lollipop Plot of Cholera Cases by Country") +
  coord_flip()
```

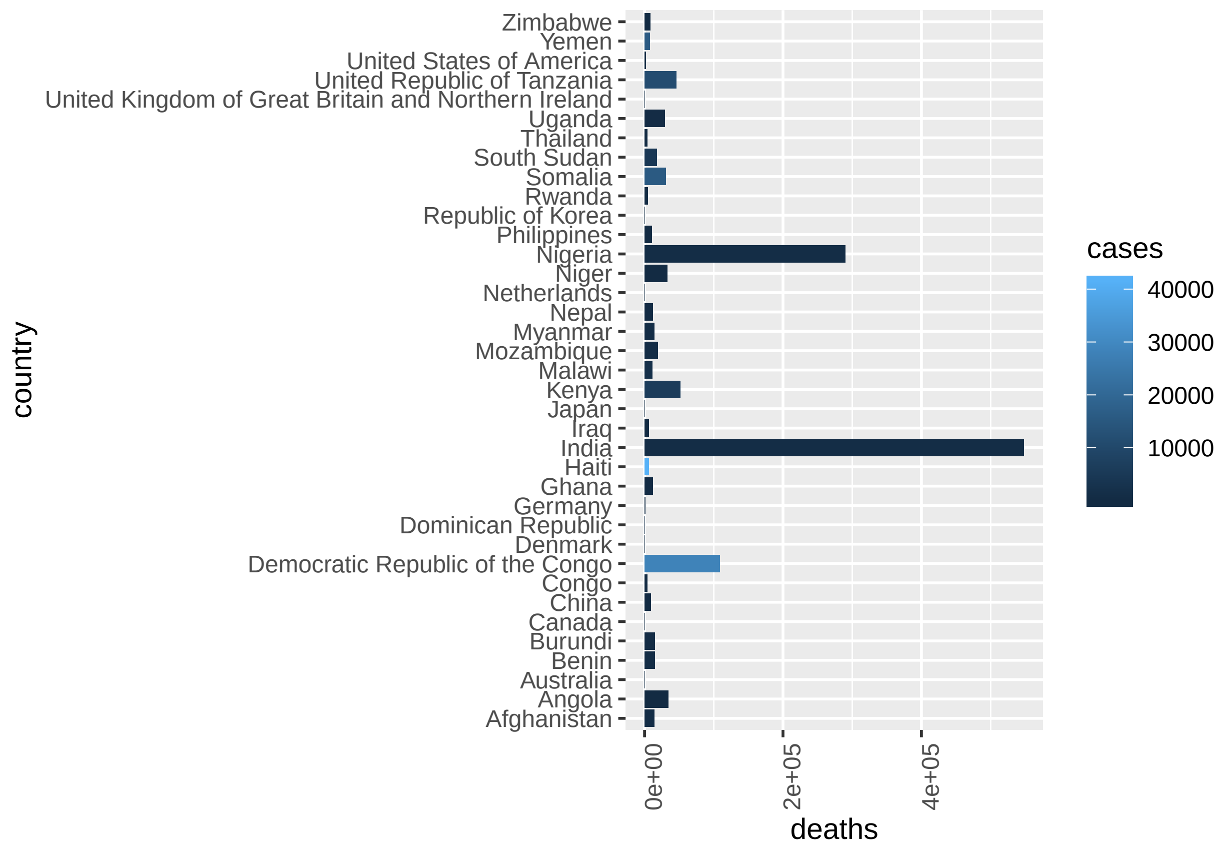


Analysis: As evident from the Lollipop chart, it appears that the country with the most observed cholera cases, is Haiti. Other countries with significant cholera populations, include the “Democratic Republic of the Congo,” “Somalia,” and “Kenya.” For the most part, it appears that Cholera is not very previlent amongs the other countries.

BarChart of Water Quality Related Deaths:

Next, in order to gain an inference on the spread of deaths related to poor water quality, I will create a histogram of the variables “deaths” and “counties.” However, I will also add a coloring aspect to the barplot, relating to how many observed cases of cholera were present in the listed countries.

```
ggplot(data = viz_data) +
  geom_bar(mapping = aes(x = country, y = deaths, fill = cases), stat = 'identity') +
  theme(axis.text.x = element_text(angle = 90)) +
  coord_flip()
```



Analysis: As evident from the Bar plot, it appears that the countries with the most water-quality related deaths, are India, Nigeria, and the Democratic Republic of the Congo. In addition, it appears that the countries with the most water quality related deaths do not differ from the countries with the least amount of water-quality related deaths, in their respective numbers of cholera cases. However, there is one exception: Kenya. From this observation, I believe that the number of water-quality related deaths may not be as correlated to the number of Cholera case, as expected, and may be dependent on the quality of treatment available for each respective country.