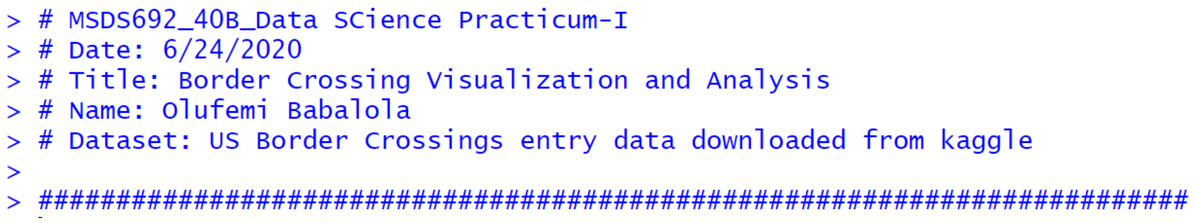
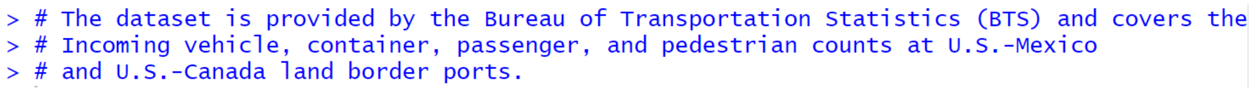
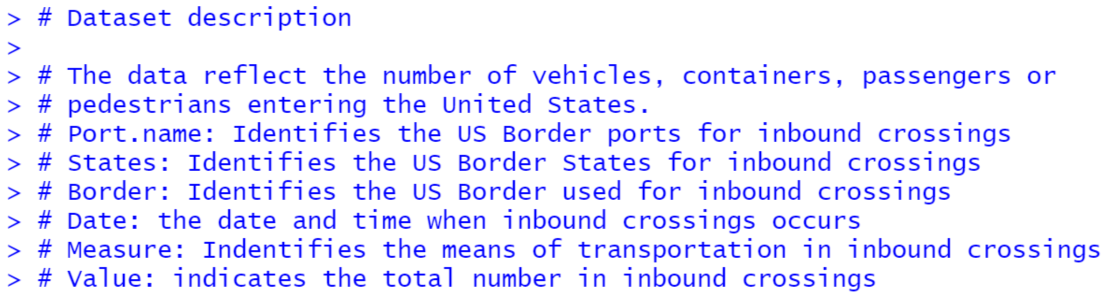
MSDS692_40 Data Science Practicum 1



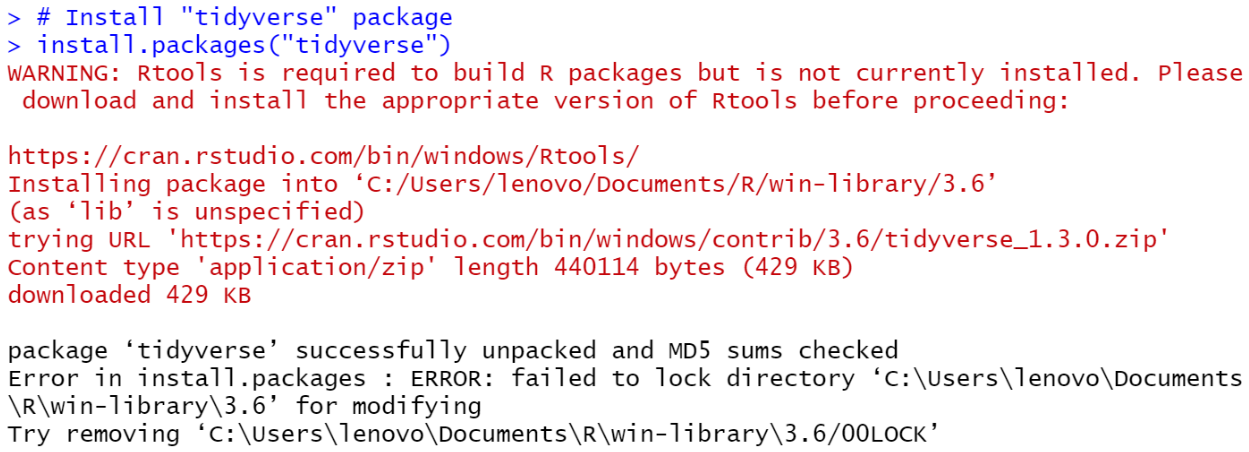
```
> # MSDS692_40B_Data SCience Practicum-I
> # Date: 6/24/2020
> # Title: Border Crossing Visualization and Analysis
> # Name: Olufemi Babalola
> # Dataset: US Border Crossings entry data downloaded from kaggle
>
> ##############################################################################
```



```
> # The dataset is provided by the Bureau of Transportation Statistics (BTS) and covers the
> # Incoming vehicle, container, passenger, and pedestrian counts at U.S.-Mexico
> # and U.S.-Canada land border ports.
```

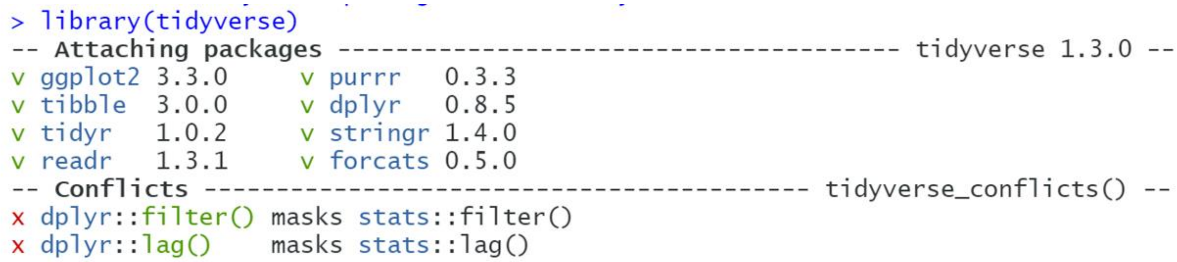


```
> # Dataset description
>
> # The data reflect the number of vehicles, containers, passengers or
> # pedestrians entering the United States.
> # Port.name: Identifies the US Border ports for inbound crossings
> # States: Identifies the US Border States for inbound crossings
> # Border: Identifies the US Border used for inbound crossings
> # Date: the date and time when inbound crossings occurs
> # Measure: Indentifies the means of transportation in inbound crossings
> # Value: indicates the total number in inbound crossings
```

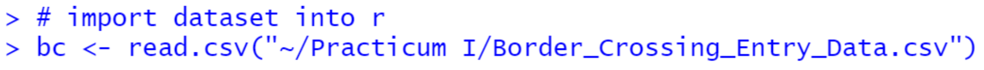


```
> # Install "tidyverse" package
> install.packages("tidyverse")
WARNING: Rtools is required to build R packages but is not currently installed. Please
 download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into ‘C:/Users/lenovo/Documents/R/win-library/3.6’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/tidyverse_1.3.0.zip'
Content type 'application/zip' length 440114 bytes (429 KB)
downloaded 429 KB

package ‘tidyverse’ successfully unpacked and MD5 sums checked
Error in install.packages : ERROR: failed to lock directory ‘C:\Users\lenovo\Documents
\R\win-library\3.6’ for modifying
Try removing ‘C:\Users\lenovo\Documents\R\win-library\3.6/00LOCK’
```

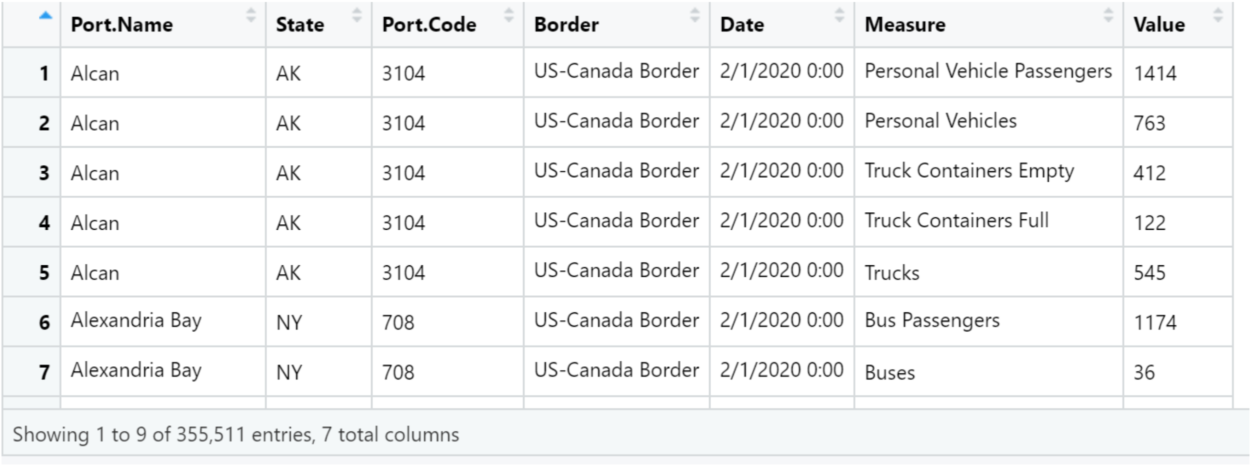
Load the installed tidyverse package into r



```
> library(tidyverse)
-- Attaching packages --------------------------------------- tidyverse 1.3.0 --
v ggplot2 3.3.0     v purrr   0.3.3
v tibble  3.0.0     v dplyr   0.8.5
v tidyr   1.0.2     v stringr 1.4.0
v readr   1.3.1     v forcats 0.5.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```



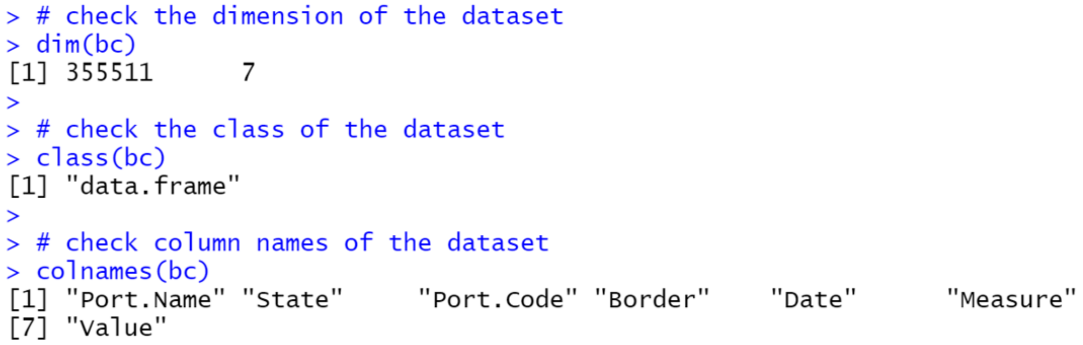
```
> # import dataset into r
> bc <- read.csv("~/Practicum I/Border_Crossing_Entry_Data.csv")
```



	Port.Name	State	Port.Code	Border	Date	Measure	Value
1	Alcan	AK	3104	US-Canada Border	2/1/2020 0:00	Personal Vehicle Passengers	1414
2	Alcan	AK	3104	US-Canada Border	2/1/2020 0:00	Personal Vehicles	763
3	Alcan	AK	3104	US-Canada Border	2/1/2020 0:00	Truck Containers Empty	412
4	Alcan	AK	3104	US-Canada Border	2/1/2020 0:00	Truck Containers Full	122
5	Alcan	AK	3104	US-Canada Border	2/1/2020 0:00	Trucks	545
6	Alexandria Bay	NY	708	US-Canada Border	2/1/2020 0:00	Bus Passengers	1174
7	Alexandria Bay	NY	708	US-Canada Border	2/1/2020 0:00	Buses	36

Showing 1 to 9 of 355,511 entries, 7 total columns

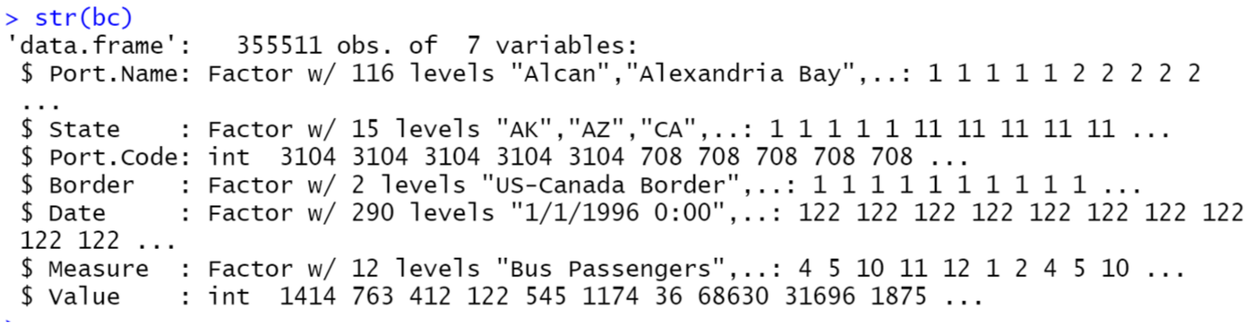
Exploring the content and structure of the dataset



```
> # check the dimension of the dataset
> dim(bc)
[1] 355511      7
>
> # check the class of the dataset
> class(bc)
[1] "data.frame"
>
> # check column names of the dataset
> colnames(bc)
[1] "Port.Name" "State"     "Port.Code" "Border"    "Date"      "Measure"  
[7] "Value"
```

Above, we see the dataset consist of 7 variables including Port.Name, State, Port.Code, Border, Date, Measure and Value.

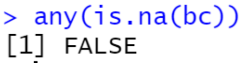
Next, let’s check the structure of the Border crossings dataset.



```
> str(bc)
'data.frame':	355511 obs. of  7 variables:
 $ Port.Name: Factor w/ 116 levels "Alcan","Alexandria Bay",..: 1 1 1 1 1 2 2 2 2 2 ...
 $ State    : Factor w/ 15 levels "AK","AZ","CA",..: 1 1 1 1 1 11 11 11 11 11 ...
 $ Port.Code: int  3104 3104 3104 3104 3104 708 708 708 708 708 ...
 $ Border   : Factor w/ 2 levels "US-Canada Border",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Date     : Factor w/ 290 levels "1/1/1996 0:00",..: 122 122 122 122 122 122 122 122 122 122 ...
 $ Measure  : Factor w/ 12 levels "Bus Passengers",..: 4 5 10 11 12 1 2 4 5 10 ...
 $ Value    : int  1414 763 412 122 545 1174 36 68630 31696 1875 ...
```

Looking at the structure, we observe that this is a data frame with 355511 observations and 7 variables.

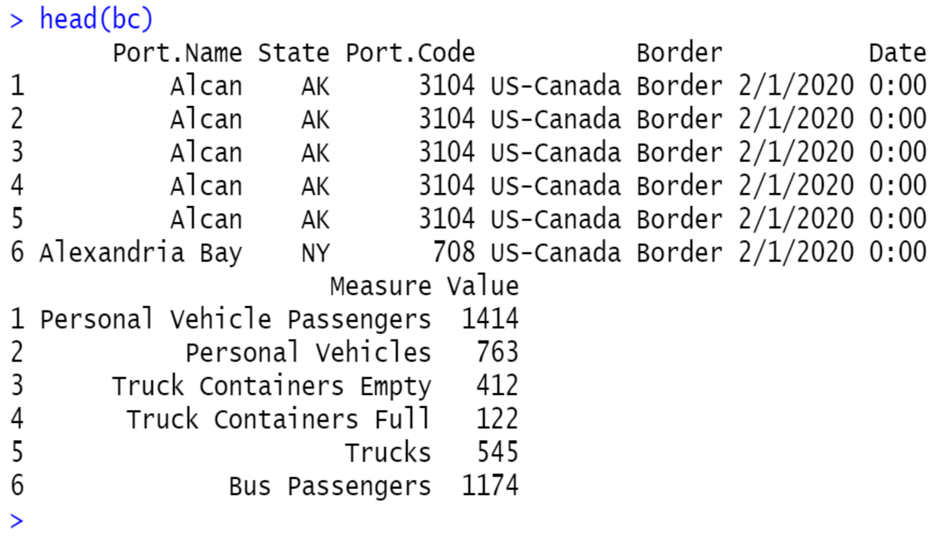
Next, check to see if there are missing values.



```
> any(is.na(bc))
[1] FALSE
```

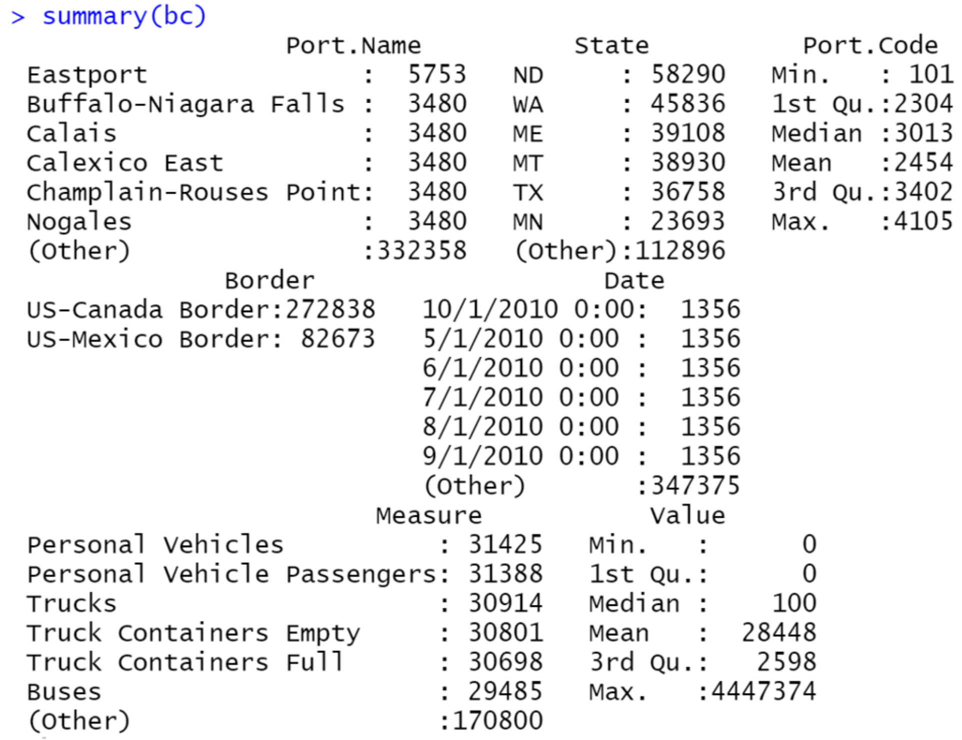
The result is false, showing that there are no missing values.

Let’s take a look at the top six records in the dataset using the head function.



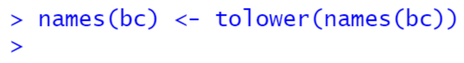
```
> head(bc)
       Port.Name State Port.Code           Border          Date
1          Alcan    AK      3104 US-Canada Border 2/1/2020 0:00
2          Alcan    AK      3104 US-Canada Border 2/1/2020 0:00
3          Alcan    AK      3104 US-Canada Border 2/1/2020 0:00
4          Alcan    AK      3104 US-Canada Border 2/1/2020 0:00
5          Alcan    AK      3104 US-Canada Border 2/1/2020 0:00
6 Alexandria Bay    NY       708 US-Canada Border 2/1/2020 0:00
                      Measure Value
1 Personal Vehicle Passengers  1414
2           Personal Vehicles   763
3      Truck Containers Empty   412
4       Truck Containers Full   122
5                      Trucks   545
6              Bus Passengers  1174
>
```

Next, let’s check the summary statistics for this dataset.



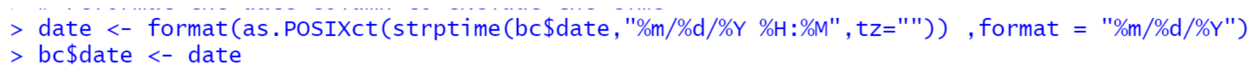
```
> summary(bc)
                  Port.Name          State          Port.Code
 Eastport              :  5753   ND     : 58290   Min.   : 101
 Buffalo-Niagara Falls :  3480   WA     : 45836   1st Qu.:2304
 Calais                :  3480   ME     : 39108   Median :3013
 Calexico East         :  3480   MT     : 38930   Mean   :2454
 Champlain-Rouses Point:  3480   TX     : 36758   3rd Qu.:3402
 Nogales               :  3480   MN     : 23693   Max.   :4105
 (Other)               :332358   (Other):112896
              Border                      Date
 US-Canada Border:272838   10/1/2010 0:00:   1356
 US-Mexico Border: 82673   5/1/2010 0:00 :   1356
                           6/1/2010 0:00 :   1356
                           7/1/2010 0:00 :   1356
                           8/1/2010 0:00 :   1356
                           9/1/2010 0:00 :   1356
                           (Other)       :347375
                        Measure           Value
 Personal Vehicles          : 31425   Min.   :      0
 Personal Vehicle Passengers: 31388   1st Qu.:      0
 Trucks                     : 30914   Median :    100
 Truck Containers Empty     : 30801   Mean   :  28448
 Truck Containers Full      : 30698   3rd Qu.:   2598
 Buses                      : 29485   Max.   :4447374
 (Other)                    :170800
```

Let’s change the column header to lowercase for uniformity.



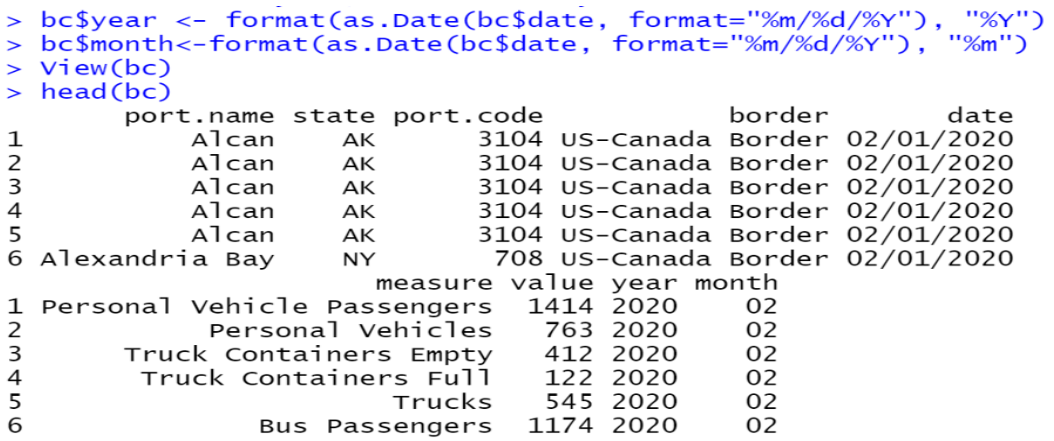
```
> names(bc) <- tolower(names(bc))
>
```

And let’s reformat the date variable to exclude the time factor.



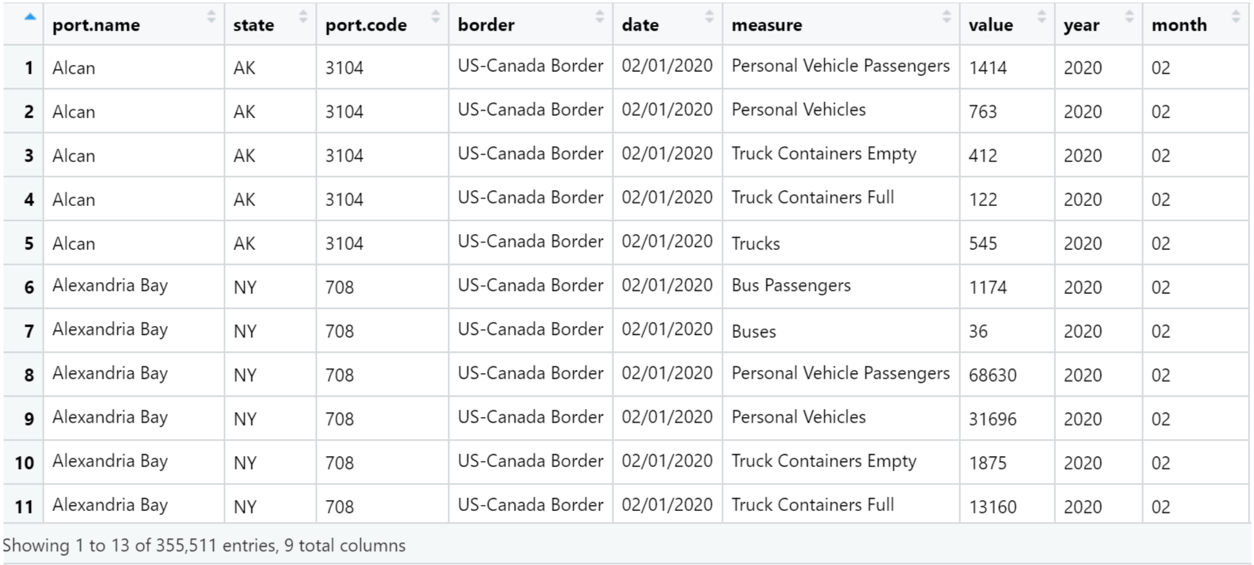
```
> date <- format(as.POSIXct(strptime(bc$date,"%m/%d/%Y %H:%M",tz="")) ,format = "%m/%d/%Y")
> bc$date <- date
```

Extract the year and month from the date column and view the dataset



```
> bc$year <- format(as.Date(bc$date, format="%m/%d/%Y"), "%Y")
> bc$month<-format(as.Date(bc$date, format="%m/%d/%Y"), "%m")
> View(bc)
> head(bc)
      port.name state port.code           border       date
1         Alcan    AK      3104 US-Canada Border 02/01/2020
2         Alcan    AK      3104 US-Canada Border 02/01/2020
3         Alcan    AK      3104 US-Canada Border 02/01/2020
4         Alcan    AK      3104 US-Canada Border 02/01/2020
5         Alcan    AK      3104 US-Canada Border 02/01/2020
6 Alexandria Bay    NY       708 US-Canada Border 02/01/2020
                measure value year month
1 Personal Vehicle Passengers  1414 2020    02
2           Personal Vehicles   763 2020    02
3       Truck Containers Empty   412 2020    02
4        Truck Containers Full   122 2020    02
5                     Trucks   545 2020    02
6              Bus Passengers  1174 2020    02
```

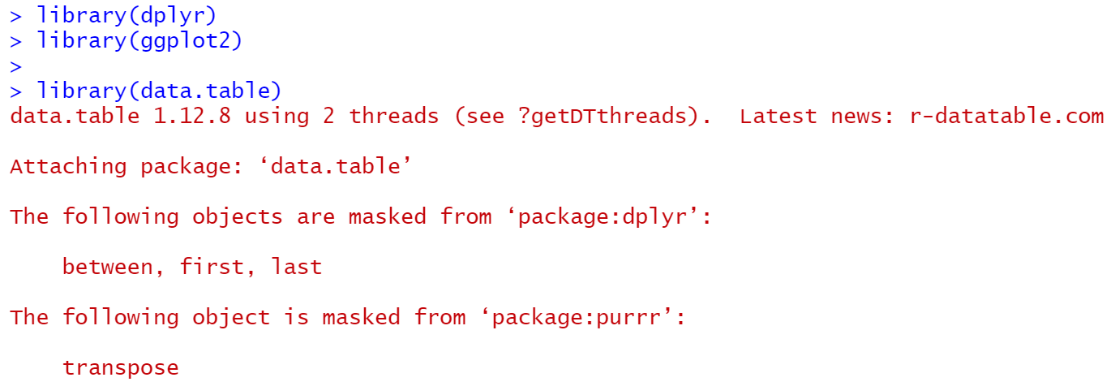
Below is the view of the reformatted dataset showing new column for year and month.



	port.name	state	port.code	border	date	measure	value	year	month
1	Alcan	AK	3104	US-Canada Border	02/01/2020	Personal Vehicle Passengers	1414	2020	02
2	Alcan	AK	3104	US-Canada Border	02/01/2020	Personal Vehicles	763	2020	02
3	Alcan	AK	3104	US-Canada Border	02/01/2020	Truck Containers Empty	412	2020	02
4	Alcan	AK	3104	US-Canada Border	02/01/2020	Truck Containers Full	122	2020	02
5	Alcan	AK	3104	US-Canada Border	02/01/2020	Trucks	545	2020	02
6	Alexandria Bay	NY	708	US-Canada Border	02/01/2020	Bus Passengers	1174	2020	02
7	Alexandria Bay	NY	708	US-Canada Border	02/01/2020	Buses	36	2020	02
8	Alexandria Bay	NY	708	US-Canada Border	02/01/2020	Personal Vehicle Passengers	68630	2020	02
9	Alexandria Bay	NY	708	US-Canada Border	02/01/2020	Personal Vehicles	31696	2020	02
10	Alexandria Bay	NY	708	US-Canada Border	02/01/2020	Truck Containers Empty	1875	2020	02
11	Alexandria Bay	NY	708	US-Canada Border	02/01/2020	Truck Containers Full	13160	2020	02

Showing 1 to 13 of 355,511 entries, 9 total columns

Let’s start by creating some visualization of the dataset. I will begin by loading dplyr, ggplot2 and data.table library.



```
> library(dplyr)
> library(ggplot2)
>
> library(data.table)
data.table 1.12.8 using 2 threads (see ?getDTthreads).  Latest news: r-datatable.com

Attaching package: ‘data.table’

The following objects are masked from ‘package:dplyr’:

    between, first, last

The following object is masked from ‘package:purrr’:

    transpose
```

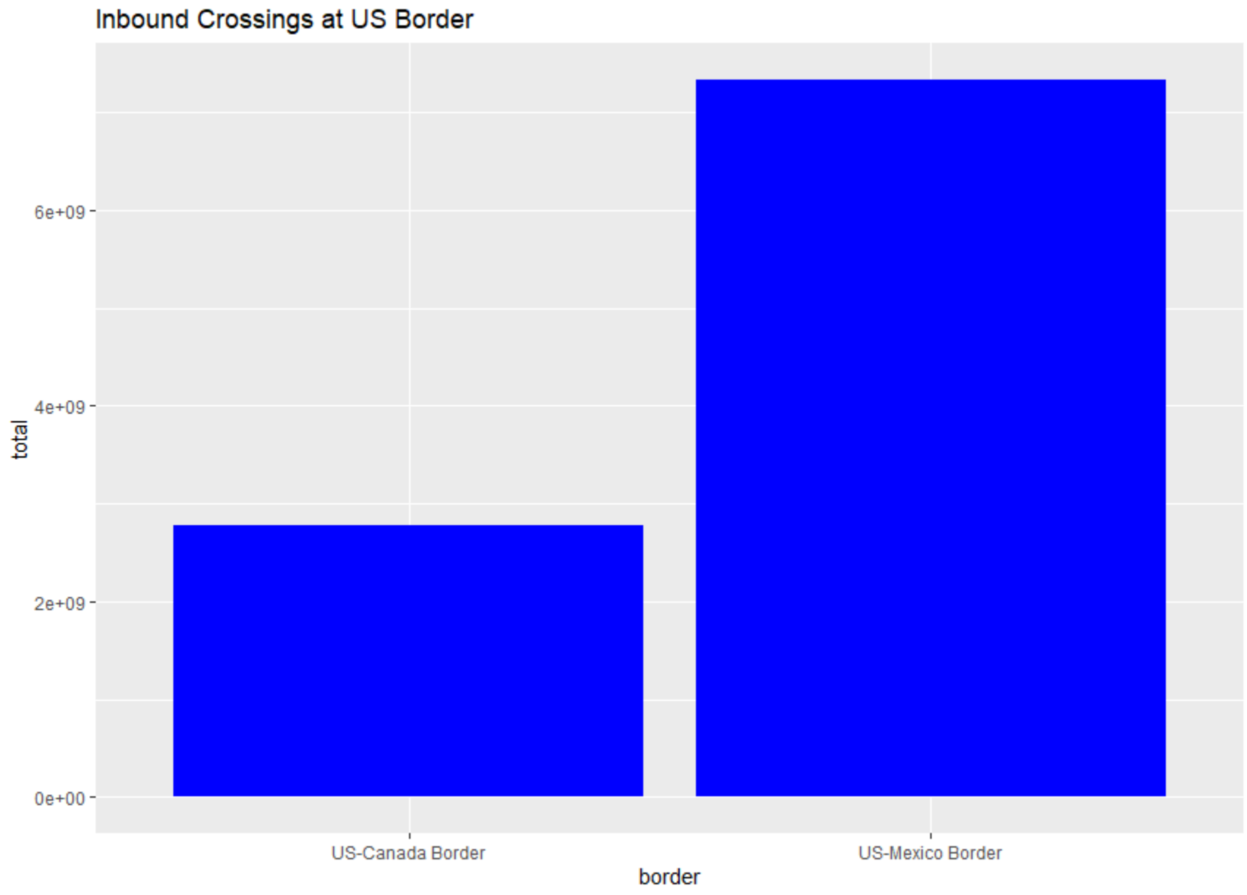
The first thing I would like to see is the inbound traffic at US Borders.

```
> summarized.border = bc[, list(total=sum(value)), by="border"]
Warning message:
In gsum(value) :
  The sum of an integer column for a group was more than type 'integer' can hold so the result
 has been coerced to 'numeric' automatically for convenience.
> summarized.border
             border      total
1: US-Canada Border 2776127401
2: US-Mexico Border 7337300710
```

From here, we noticed that 73.3 percent of total inbound crossing occurs at US-Mexico Border.

Next, let’s see the plot of inbound crossing at the two borders.

```
> ggplot(data = summarized.border,
+        mapping = aes(x = border,
+                      y = total)) +
+    geom_bar(stat = "identity", fill = "blue") +
+   ggtitle("Inbound Crossings at US Border")
>
```



Next, I would like to see the number of inbound crossing at the various ports at US Border.

```
> summarized.port = bc[, list(total=sum(value)), by="port.name"]
> summarized.port
           port.name    total
  1:           Alcan  4407101
  2:  Alexandria Bay 64210750
  3:         Algonac   121107
  4:         Ambrose   213484
  5:         Andrade 75204404
 ---
112: Toledo-Sandusky      607
113:        Portland   956834
114:       Whitetail   160092
115:      Bar Harbor   247988
116:           Noyes  1919393
```

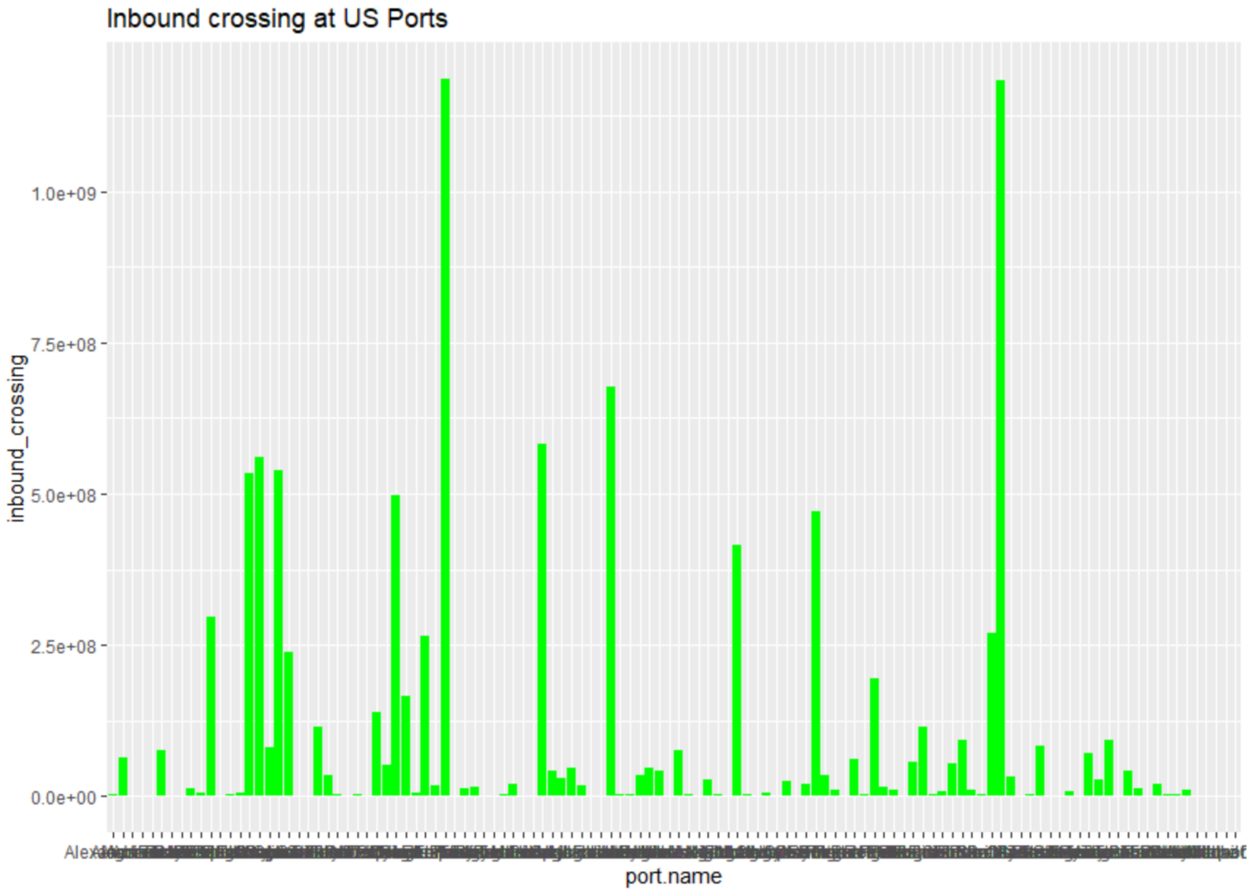
Here, we notice we have 116 border ports in our dataset.

```
> incoming_crossing_port = bc %>%
+   group_by(port.name) %>%
+   summarise(inbound_crossing = sum(value))-> Port_crossings
> Port_crossings <- as.data.frame(Port_crossings)
> Port_crossings
                      port.name inbound_crossing
1                         Alcan          4407101
2                Alexandria Bay         64210750
3                       Algonac           121107
4                       Ambrose           213484
5                     Anacortes          1690849
6                       Andrade         75204404
7                        Antler           836811
8                    Bar Harbor           247988
9                      Baudette         13991091
10                Beecher Falls          6175430
11                       Blaine        295794708
12                     Boquillas            71870
13                     Boundary          3574372
14                  Bridgewater          5785245
15                  Brownsville        533360410
16         Buffalo-Niagara Falls        559736205
17                       Calais         80276044
18                      Calexico        538455020
19                 Calexico East        238071229
20                 Cape Vincent           342164
```

This list runs through 116 ports.

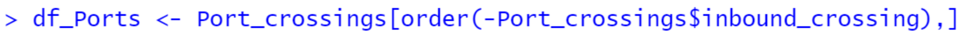
Next, let’s see the plots for inbound crossing at US Border plot.

```
> ggplot(data = Port_crossings,
+        mapping = aes(x = port.name,
+                      y = inbound_crossing)) +
+   geom_bar(stat = "identity", fill = "green") +
+   ggtitle("Inbound crossing at US Ports")
>
```

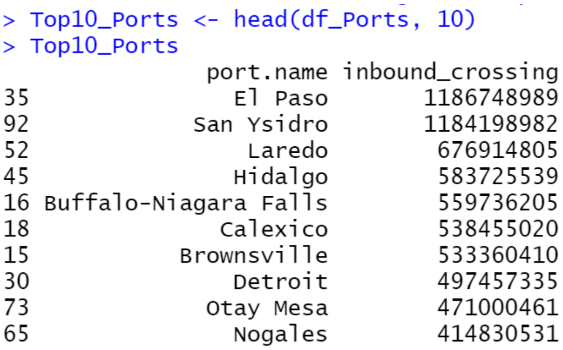


The x-axis is made of 116 port names hence they are fused together.

To rank the data in terms of the port with the most traffic, we sort 116 ports in descending order, and we check for the top ten ports.



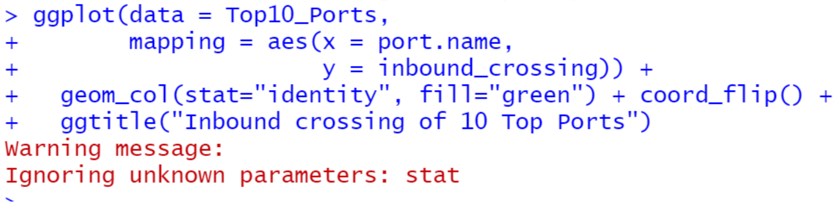
```
> df_Ports <- Port_crossings[order(-Port_crossings$inbound_crossing),]
```



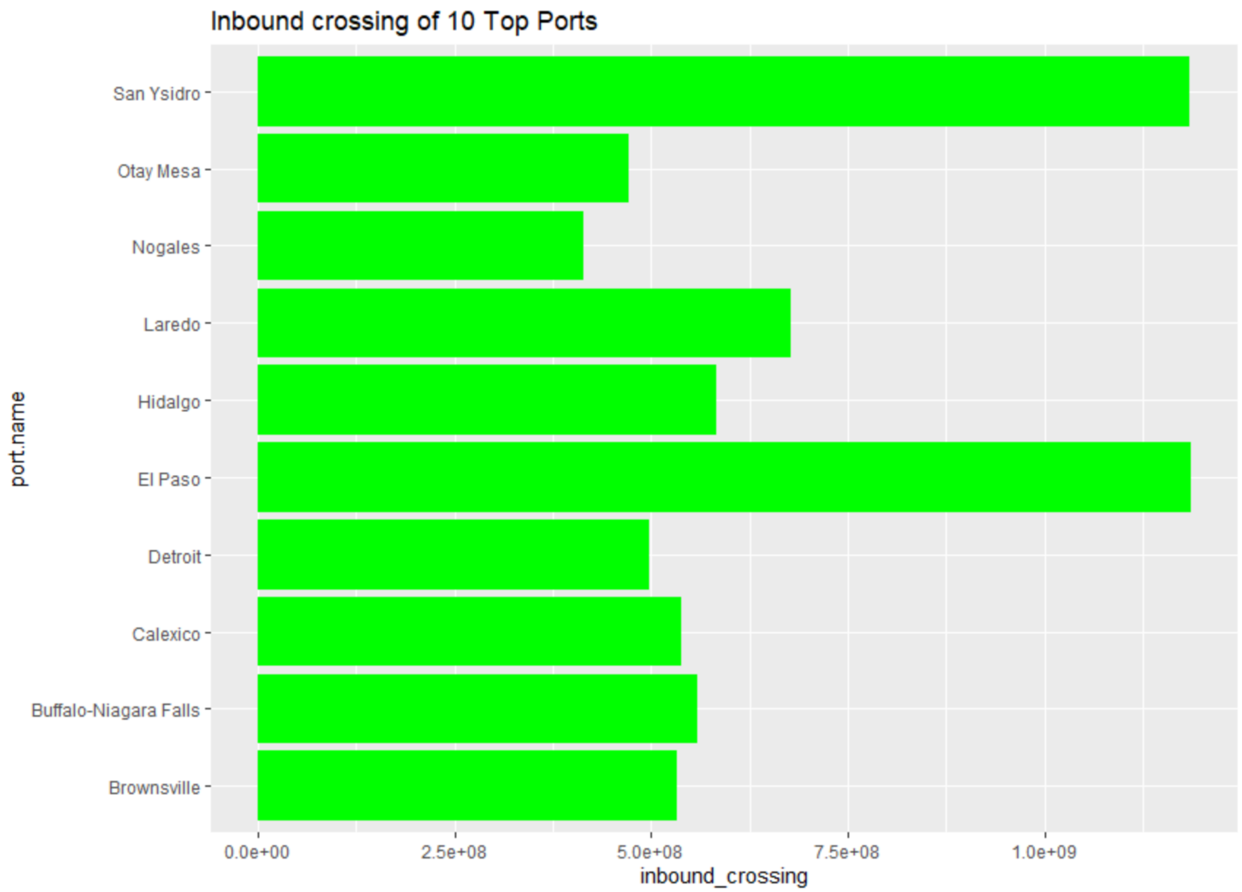
```
> Top10_Ports <- head(df_Ports, 10)
> Top10_Ports
               port.name inbound_crossing
35               El Paso       1186748989
92            San Ysidro       1184198982
52                Laredo        676914805
45               Hidalgo        583725539
16 Buffalo-Niagara Falls        559736205
18              Calexico        538455020
15           Brownsville        533360410
30               Detroit        497457335
73             Otay Mesa        471000461
65               Nogales        414830531
```

From the above numbers, we observe most of the crossing takes place at El Paso port followed by San Ysidro and Laredo. To my surprise, Detroit is number seven on the list.

Next, let’s see the plot of this ranking.



```
> ggplot(data = Top10_Ports,
+        mapping = aes(x = port.name,
+                      y = inbound_crossing)) +
+   geom_col(stat="identity", fill="green") + coord_flip() +
+   ggtitle("Inbound crossing of 10 Top Ports")
Warning message:
Ignoring unknown parameters: stat
```



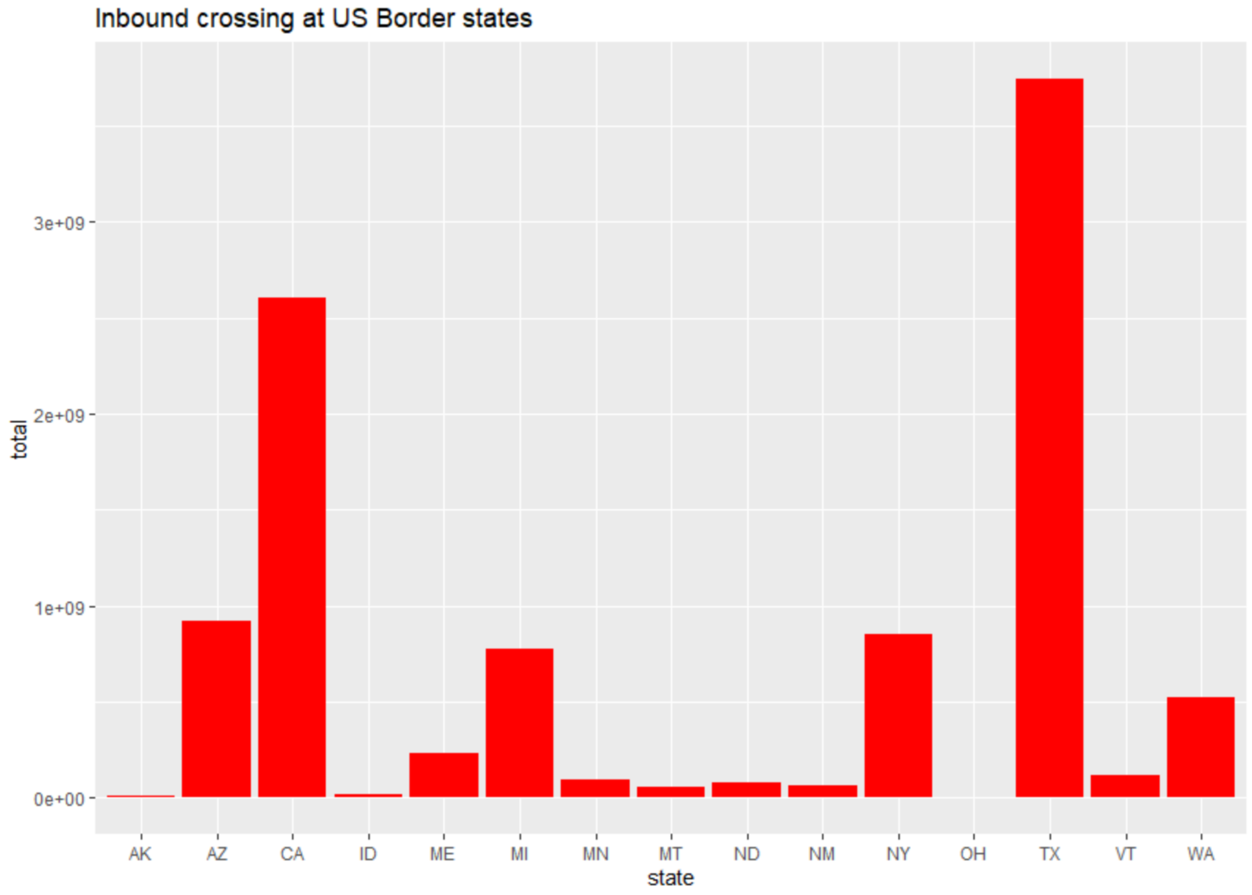
Next, I will check the inbound traffic at US Border States.

```
> summarized.state = bc[, list(total=sum(value)), by="state"]
Warning message:
In gsum(value) :
  The sum of an integer column for a group was more than type 'integer' can hold so the result
 has been coerced to 'numeric' automatically for convenience.
> summarized.state
    state      total
 1:    AK   14676856
 2:    NY  854042599
 3:    MI  775410321
 4:    ND   78600964
 5:    CA 2602572970
 6:    MN   99126985
 7:    VT  118116868
 8:    WA  521397911
 9:    TX 3747879529
10:    ME  236781464
11:    NM   68560332
12:    MT   56139436
13:    AZ  918287879
14:    ID   21833390
15:    OH        607
```

From the data drawn, we observe that Texas has the most inbound traffic followed by California.

Next, I plotted the figures above.

```
> library(ggplot2)
> ggplot(data = summarized.state,
+        mapping = aes(x = state,
+                      y = total)) +
+   geom_bar(stat = "identity", fill = "green") +
+   ggtitle("Inbound crossing at US Border states")
>
```



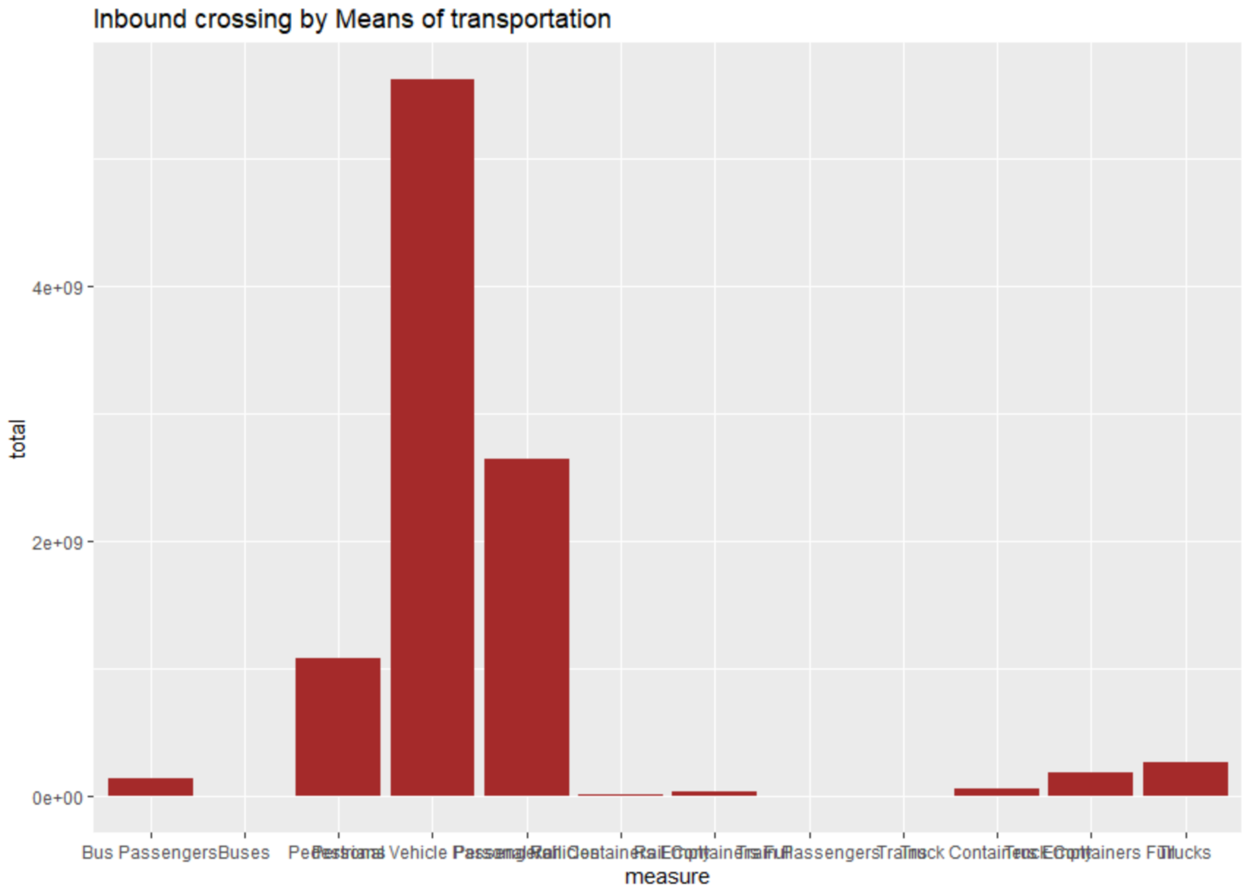
Next, I checked the inbound traffic by means of Transportation.

```
> summarized.measure = bc[, list(total=sum(value)), by="measure"]
Warning message:
In gsum(value) :
  The sum of an integer column for a group was more than type 'integer' can hold so the result
 has been coerced to 'numeric' automatically for convenience.
> summarized.measure
                    measure      total
 1: Personal Vehicle Passengers 5629526756
 2:           Personal Vehicles 2651535415
 3:        Truck Containers Empty   67036035
 4:         Truck Containers Full  185463194
 5:                      Trucks  264731943
 6:              Bus Passengers  146027374
 7:                       Buses    8754394
 8:                 Pedestrians 1090067964
 9:         Rail Containers Empty   22386399
10:          Rail Containers Full   40492650
11:            Train Passengers    6472717
12:                      Trains     933270
```

Here from the data retrieved, we see the most used mode of transportation were with personal vehicle passengers.

Next, I am plotting the inbound crossings by transportation methods.

```
> library(ggplot2)
> ggplot(data = summarized.measure,
+        mapping = aes(x = measure,
+                      y = total)) +
+   geom_bar(stat = "identity", fill = "brown") +
+   ggtitle("Inbound crossing by Means of transportation")
>
```

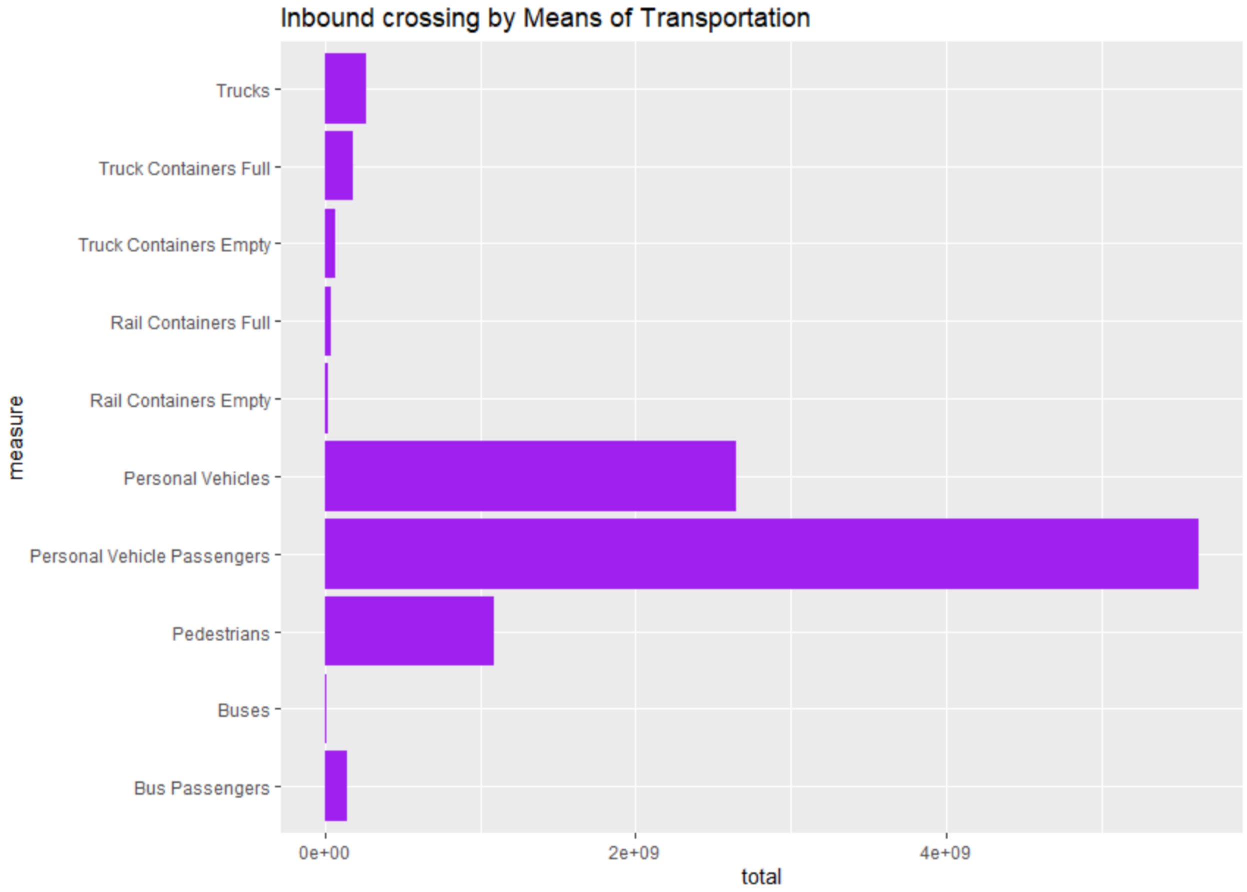


Next, I sorted the measure of transportation.

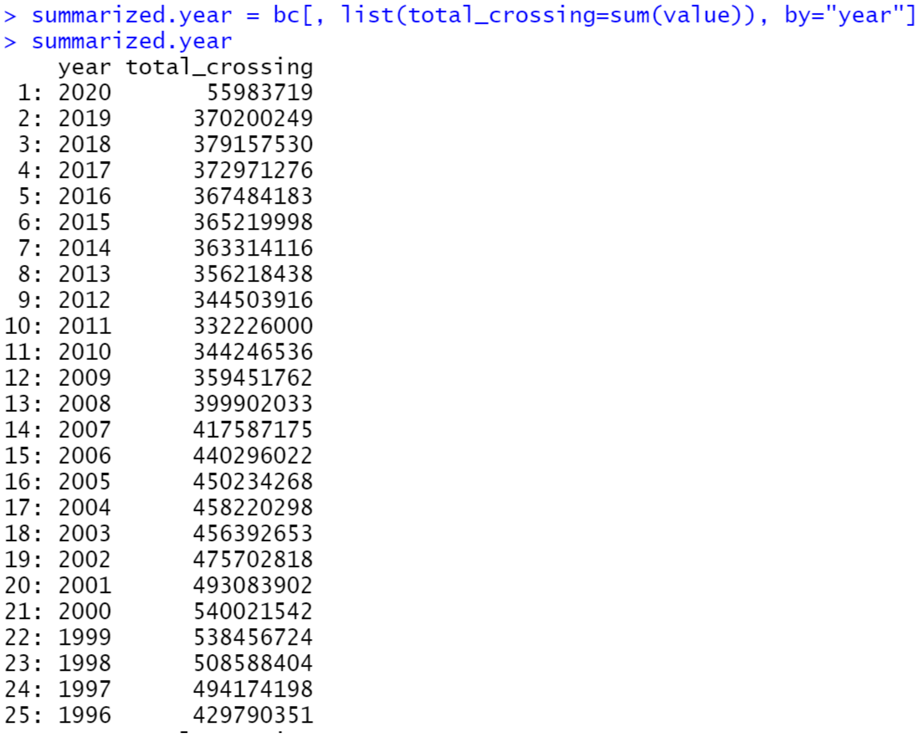
```
> # sort Measure in descending order
> Traffic <- summarized.measure[order(-summarized.measure$total),]
>
> # check the inbound crossings for top 10 measures
> # and assigned to new data frame "Top_10_Measure"
> Top_10_Measure <- head(Traffic, 10)
> Top_10_Measure
                        measure      total
 1: Personal Vehicle Passengers 5629526756
 2:           Personal Vehicles 2651535415
 3:                 Pedestrians 1090067964
 4:                      Trucks  264731943
 5:       Truck Containers Full  185463194
 6:              Bus Passengers  146027374
 7:      Truck Containers Empty   67036035
 8:        Rail Containers Full   40492650
 9:       Rail Containers Empty   22386399
10:                       Buses    8754394
```

In the following code, I plotted the top ten inbound crossing measure of transportation and we can see that personal vehicle passengers are the highest.

```
> ggplot(data = Top_10_Measure,
+        mapping = aes(x = measure,
+                      y = total)) +
+   geom_col(stat = "identity", fill = "purple") + coord_flip() +
+   ggtitle("Inbound crossing by Means of Transportation")
```

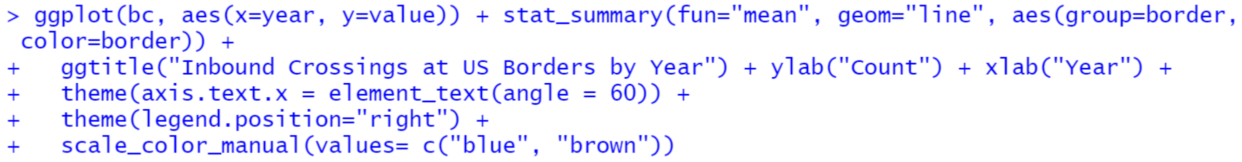


Next, let’s check the yearly inbound traffic.

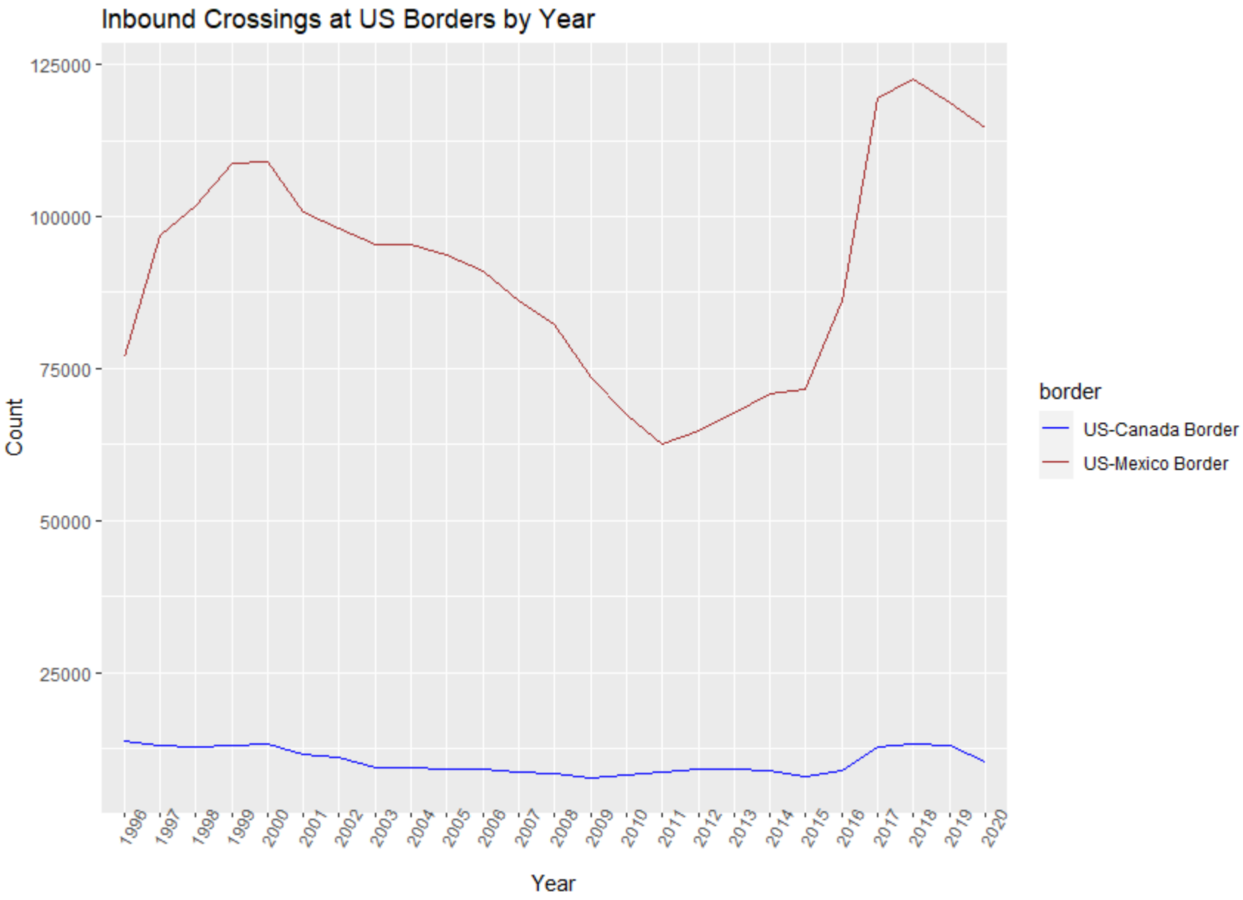


```
> summarized.year = bc[, list(total_crossing=sum(value)), by="year"]
> summarized.year
    year total_crossing
 1: 2020       55983719
 2: 2019      370200249
 3: 2018      379157530
 4: 2017      372971276
 5: 2016      367484183
 6: 2015      365219998
 7: 2014      363314116
 8: 2013      356218438
 9: 2012      344503916
10: 2011      332226000
11: 2010      344246536
12: 2009      359451762
13: 2008      399902033
14: 2007      417587175
15: 2006      440296022
16: 2005      450234268
17: 2004      458220298
18: 2003      456392653
19: 2002      475702818
20: 2001      493083902
21: 2000      540021542
22: 1999      538456724
23: 1998      508588404
24: 1997      494174198
25: 1996      429790351
```

Here, I plotted the trend of annual inbound traffic at US Borders.

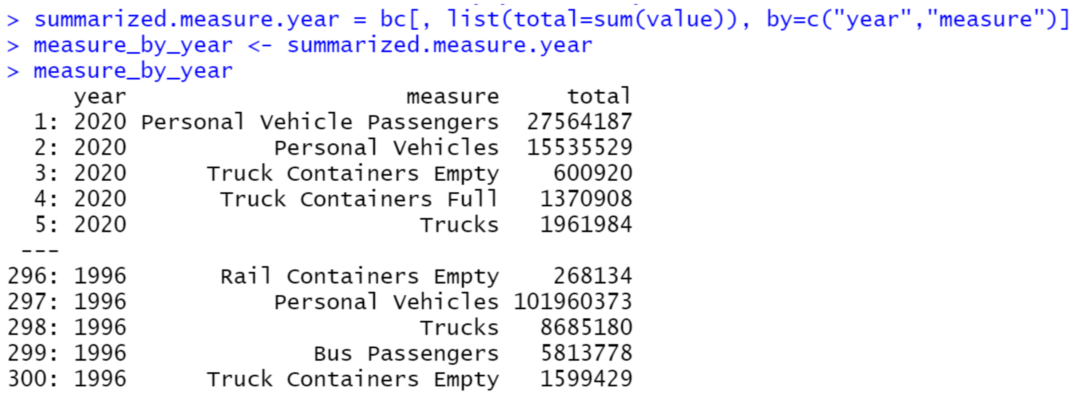


```
> ggplot(bc, aes(x=year, y=value)) + stat_summary(fun="mean", geom="line", aes(group=border,
 color=border)) +
+   ggtitle("Inbound Crossings at US Borders by Year") + ylab("Count") + xlab("Year") +
+   theme(axis.text.x = element_text(angle = 60)) +
+   theme(legend.position="right") +
+   scale_color_manual(values= c("blue", "brown"))
```



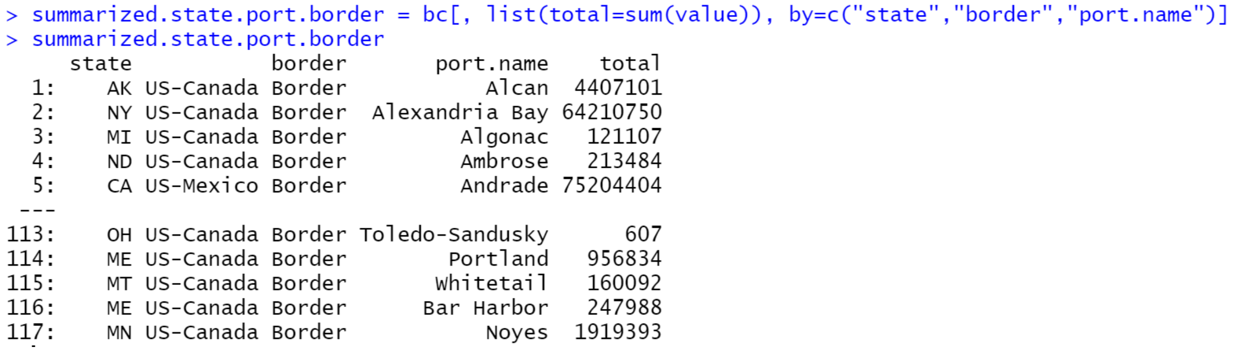
The numbers over the years at US-Canada Border portray to be steady whereas from the Mexico Border, we see a huge peak from 2016. When President Trump came into office and proposed the border wall, Hispanics started flooding into the country.

Next, I showcased the inbound traffic at US Borders by year and measure of transportation.



```
> summarized.measure.year = bc[, list(total=sum(value)), by=c("year","measure")]
> measure_by_year <- summarized.measure.year
> measure_by_year
     year                     measure     total
  1: 2020 Personal Vehicle Passengers  27564187
  2: 2020           Personal Vehicles  15535529
  3: 2020       Truck Containers Empty    600920
  4: 2020        Truck Containers Full   1370908
  5: 2020                      Trucks   1961984
 ---                                          
296: 1996        Rail Containers Empty    268134
297: 1996           Personal Vehicles 101960373
298: 1996                      Trucks   8685180
299: 1996               Bus Passengers   5813778
300: 1996       Truck Containers Empty   1599429
```

In this report, we can see the inbound traffic at US Borders by states, borders and ports.



```
> summarized.state.port.border = bc[, list(total=sum(value)), by=c("state","border","port.name")]
> summarized.state.port.border
     state          border        port.name    total
  1:    AK US-Canada Border            Alcan  4407101
  2:    NY US-Canada Border  Alexandria Bay 64210750
  3:    MI US-Canada Border          Algonac   121107
  4:    ND US-Canada Border          Ambrose   213484
  5:    CA US-Mexico Border          Andrade 75204404
 ---                                                 
113:    OH US-Canada Border Toledo-Sandusky      607
114:    ME US-Canada Border         Portland   956834
115:    MT US-Canada Border        Whitetail   160092
116:    ME US-Canada Border       Bar Harbor   247988
117:    MN US-Canada Border            Noyes  1919393
```

Conclusion:

The US Border crossing entry data found on Kaggle was explored, visualized and analyzed in this presentation. The results from these activities reveals some interesting trends in the inbound traffic across the US States associated with Mexico and Canada borders. Though the US and Canada Border are associated with 12 out of the 16 border States, we observed that most of the inbound crossings takes place at the southern US and Mexico Border. Also, noted is the fact that passenger vehicles are used for most of the incoming crossings into the US. This is followed by personal vehicles and thirdly by pedestrians who most likely crossed in by foot. There was no record of crossing into the US through underground tunnels. We also observed from the visualizations that most crossing occurred at Texas and specifically at El Paso, San Ysidro and Laredo ports. The Trend over the years from 1996 to February 2020 shows a sharp drop of inbound crossing between 2010 and 2015, thereafter, in 2016, there was a huge peak in the numbers flooding the country especially through the US and Mexico Border. These trends obviously reflect changes in the US Mexico immigration policy decisions which led to the building of wall across the southern border states with Mexico to control the flow of inbound traffic.

References

- Akhil. (2019, August 21). Border Crossing Entry Data. Retrieved May 20, 2020, from https://www.kaggle.com/akhilv11/border-crossing-entry-data

- Bischl, B., Lang, M., & Kotthoff, L. (n.d.). Learning Tasks. Retrieved June 3, 2020, from https://mlr.mlr-org.com/articles/tutorial/task.html

- Border Crossing/Entry Data. (n.d.). Retrieved May 24, 2020, from https://www.bts.gov/content/border-crossingentry-data

- Murray. (n.d.). Border Crossings Project. Retrieved June 15, 2020, from https://rstudio-pubs-static.s3.amazonaws.com/539991_356f6cb96e2f4062af09765937e4eea7.html