Final_Project_Aditi

Aditi Jain

4/13/2021

OVERALL QUESTIONS: How much time passed between the breach date and the date the breach was found? How long does a breach normally last? How do the two relate to the States and the Individuals Affected?

Question 1: HOW MUCH TIME PASSES BETWEEN THE BREACH DATE AND THE DATE OF BREACH WAS FOUND?

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.0
## v tidyr   1.1.0     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()
```

```
library(dplyr)
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(hexbin)
```

```
## Warning: package 'hexbin' was built under R version 4.0.3
```

```
library(modelr)
library(ggpubr)
library(mapproj)
```

```
## Warning: package 'mapproj' was built under R version 4.0.5

## Loading required package: maps

##
## Attaching package: 'maps'

## The following object is masked from 'package:purrr':
##
##     map
```

```
cyberdata = read_csv("Cyber Security Breaches.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   Number = col_double(),
##   Name_of_Covered_Entity = col_character(),
##   State = col_character(),
##   Business_Associate_Involved = col_character(),
##   Individuals_Affected = col_double(),
##   Date_of_Breach = col_character(),
##   Type_of_Breach = col_character(),
##   Location_of_Breached_Information = col_character(),
##   Date_Posted_or_Updated = col_character(),
##   Summary = col_character(),
##   breach_start = col_character(),
##   breach_end = col_character(),
##   year = col_double()
## )
```

Variables being explored: Date_of_Breach and Date_Posted_or_Updated

Types of the variables

After checking the type of the variables, they were character variables. So, I converted them to the Date type to make it easier to look at patterns and explore their relationship.

Furthermore, since I will only be exploring the relationships between the dates in the cyberdata data frame, I will filter out all the other variables.

```r
str(cyberdata$Date_of_Breach)
```

```
##  chr [1:1055] "10/16/2009" "9/22/2009" "10/12/2009" "10/9/2009" "9/27/2009" ...
```

```r
str(cyberdata$Date_Posted_or_Updated)
```

```
##  chr [1:1055] "6/30/2014" "5/30/2014" "1/23/2014" "1/23/2014" "1/23/2014" ...
```

```r
str(cyberdata$breach_start)
```

```
##  chr [1:1055] "10/16/2009" "9/22/2009" "10/12/2009" "10/9/2009" "9/27/2009" ...
```

```r
str(cyberdata$breach_end)
```

```
##  chr [1:1055] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA ...
```

```r
#Change type of Data_of_Breach:
cyberdata_edited = data.frame(cyberdata)
cyberdata_edited <- cyberdata_edited %>%
mutate(Date_of_Breach = mdy(Date_of_Breach))
```

```
## Warning: 146 failed to parse.
```

```r
#Change type of Date_Posted_or_Updated
cyberdata_edited <- cyberdata_edited %>%
mutate(Date_Posted_or_Updated = mdy(Date_Posted_or_Updated))

#Change type of breach_start and breach_end
cyberdata_edited <- cyberdata_edited %>%
mutate(breach_start = mdy(breach_start))

cyberdata_edited <- cyberdata_edited %>%
mutate(breach_end = mdy(breach_end))

cyberdata_edited <- cyberdata_edited %>%
select(Date_of_Breach, Date_Posted_or_Updated, breach_start, breach_end, year)

str(cyberdata_edited$Date_of_Breach)
```

```
##  Date[1:1055], format: "2009-10-16" "2009-09-22" "2009-10-12" "2009-10-09" "2009-09-27" ...
```

```r
str(cyberdata_edited$Date_Posted_or_Updated)
```

```
##  Date[1:1055], format: "2014-06-30" "2014-05-30" "2014-01-23" "2014-01-23" "2014-01-23" ...
```

```r
str(cyberdata_edited$breach_start)
```

```
##  Date[1:1055], format: "2009-10-16" "2009-09-22" "2009-10-12" "2009-10-09" "2009-09-27" ...
```

```
str(cyberdata_edited$breach_end)
```

```
##  Date[1:1055], format: NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA ...
```

Describe why these variables are relevant and why others are not relevant?

These two variables of Date_of_Breach and Date_Posted_or_Updated are relevant because we want to explore the time it takes for a breach to be detected after it occurs. We will determine by this by first individually understanding the variables and then looking at their distribution of their difference to observe a trend.

Variable 1: Date_of_breach

Visualizing distribution

1. Which values are most common and why?

When filtering the data set to look at the number of values that are the most common, there are 11 date/date-ranges that appear 4 times. When looking at dates/data ranges that have values that repeat more than 4 times, I find that the date that is repeated the most is 2012-01-11 and 2011-06-24 with a count of 7.

```
cyberdata_edited %>% group_by(Date_of_Breach) %>% count() %>% filter(n==4)
```

```
## # A tibble: 11 x 2
## # Groups:   Date_of_Breach [11]
##    Date_of_Breach     n
##    <date>         <int>
##  1 2008-01-07         4
##  2 2009-09-23         4
##  3 2011-04-17         4
##  4 2011-05-09         4
##  5 2011-07-28         4
##  6 2011-09-06         4
##  7 2011-09-09         4
##  8 2013-01-01         4
##  9 2013-08-22         4
## 10 2013-10-16         4
## 11 2014-01-23         4
```

```
cyberdata_edited %>% group_by(Date_of_Breach) %>% count() %>% filter(n>4)
```

```
## # A tibble: 7 x 2
## # Groups:   Date_of_Breach [7]
##   Date_of_Breach     n
##   <date>         <int>
## 1 2009-09-27         6
## 2 2010-02-04         5
## 3 2011-03-10         6
## 4 2011-06-24         7
```

```
## 5 2012-01-11          7
## 6 2013-09-20          6
## 7 NA                146
```

2. Which values are rare? Why? Does that match your expectations?

The values that are rare are the ones that only have a count of 1. There are 488 dates that have these counts. This just means that this date occurs only once in the data set. The date of 1997-01-01 is the earliest date that occurs once and then the years 2002-05-06, 2003-03-29, 2004-05-01 are also rare occurences when considering the year. Additionally, the year 2008 also has only two occurences in the months of January and December.

```
cyberdata_edited %>% group_by(Date_of_Breach) %>% count() %>% filter(n==1)
```

```
## # A tibble: 488 x 2
## # Groups:   Date_of_Breach [488]
##    Date_of_Breach     n
##    <date>         <int>
##  1 1997-01-01         1
##  2 2002-05-06         1
##  3 2003-03-29         1
##  4 2004-05-01         1
##  5 2008-01-01         1
##  6 2008-12-01         1
##  7 2009-01-11         1
##  8 2009-02-19         1
##  9 2009-07-15         1
## 10 2009-09-30         1
## # ... with 478 more rows
```

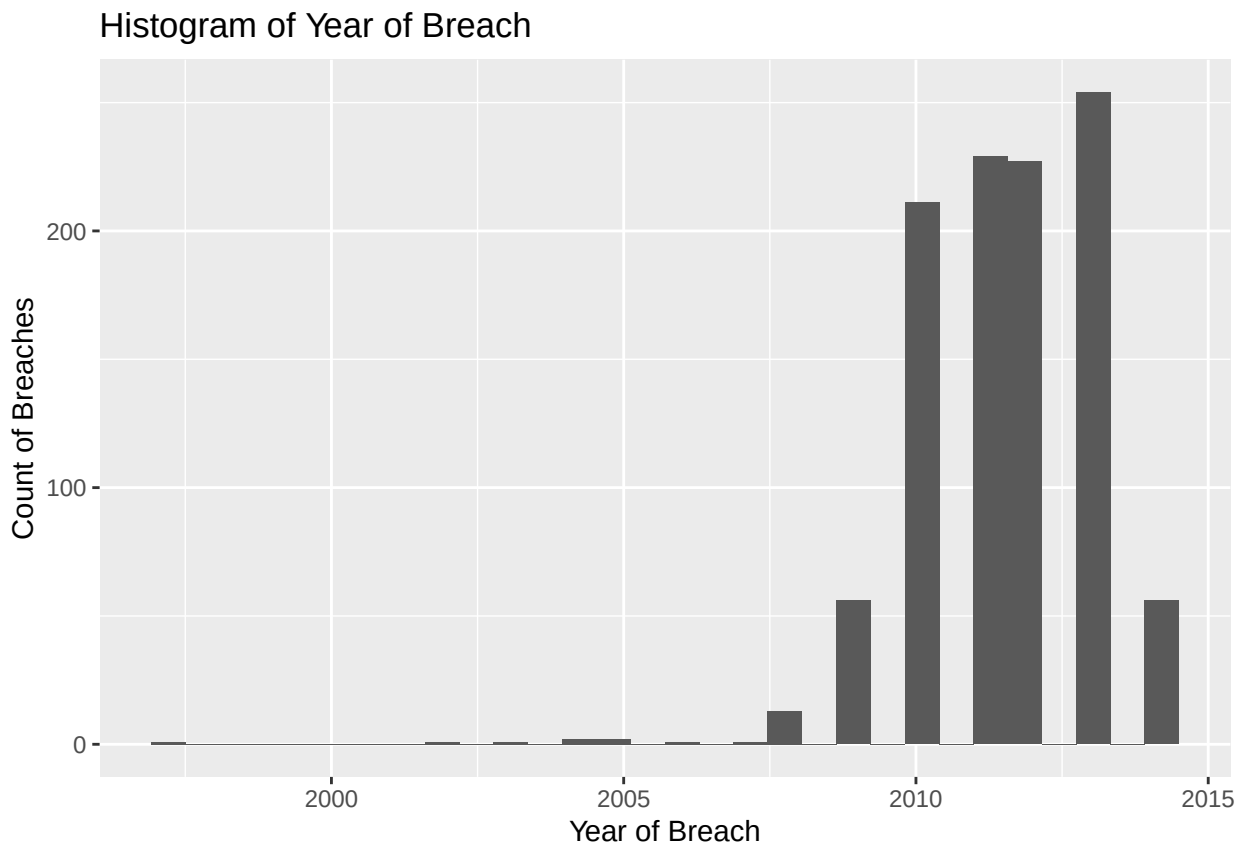
3. Can you see any unusual patterns? What might explain them?

When plotting the year variable, which represents the year of the Date of Breach, we can see that there is a left skewed distribution. This means that there is a lot of data towards the left, which consists of the years 2010-2013. The unusual pattern we can see is that not a lot of data points are available for the years before 2009.

Important

```
cyberdata_edited %>% group_by(year) %>%
ggplot(aes(year)) + geom_histogram() +
labs(x="Year of Breach", y="Count of Breaches", title="Histogram of Year of Breach")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

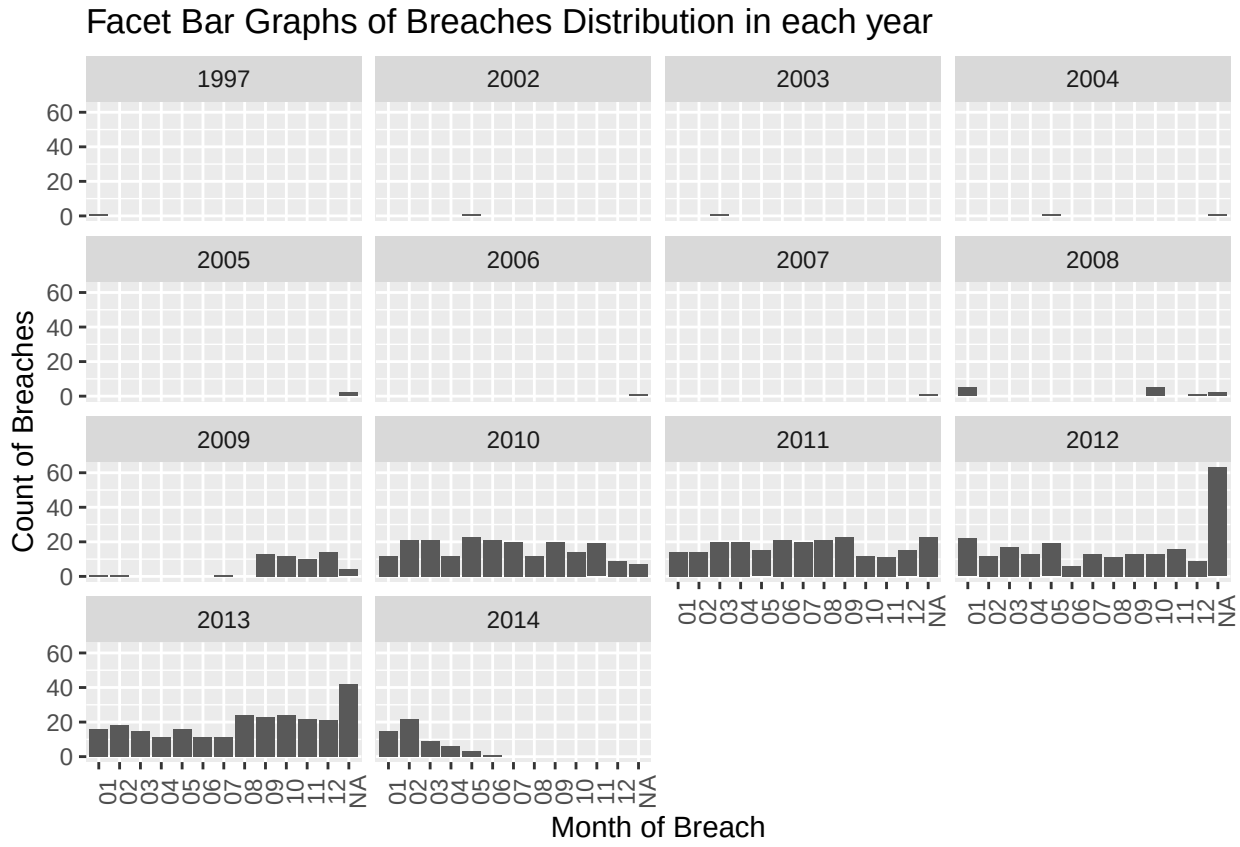


4. How are the observations within each cluster similar to or different from each other?

When observing the clusters of months within each of the years among the Date of Breach category, we can see that in 2009, a majority of the breaches occured towards the last months such as August to December. In the years 2010 through 2013, there seems to be breaches occuring in all the months. But in 2014, we can see that breaches occur in the first couple of months from January through May and then significantly decline.

Additionally, we must keep in the mind the NA values that are represented in the last bar after the month of December. There seems to be the most number of NA values in the year 2012, followed by 2013 and then 2011.

```
cyberdata_edited <- cyberdata_edited %>%
mutate(month_of_breach = strftime(cyberdata_edited$Date_of_Breach, "%m"))

ggplot(data = cyberdata_edited) +
geom_bar(mapping = aes(x = month_of_breach)) +
facet_wrap(~year) +
theme(axis.text.x = element_text(angle = 90)) +
labs(x="Month of Breach", y="Count of Breaches", title="Facet Bar Graphs of Breaches Distribution in ea
```



5. How can you explain or describe the clusters?

While there is no explanation explicitly for why these clusters occur, I believe its mostly because throughout the years from 2010-2013, there were probably more number of breaches. However, other than that, there is no clear way to account for the clusters.

Unusual values

1. Describe and demonstrate how you determine if there are unusual values in the data. E.g. too large, too small, negative, etc.

The largest or latest date in the data set is 2014-06-02 or 2nd of June, 2014. The smallest or earliest date in the data set is 1997-01-01 or 1st of January, 1997. Additionally, we find no negative values which is what we expect since they are dates.

Hence we determine there to be no unusual values.

```
max_dates <- summarize(cyberdata_edited, max(Date_of_Breach, na.rm = TRUE))
max_dates
```

```
##   max(Date_of_Breach, na.rm = TRUE)
## 1                        2014-06-02
```

```
min_dates <- summarize(cyberdata_edited, min(Date_of_Breach, na.rm = TRUE))
min_dates
```

```
##   min(Date_of_Breach, na.rm = TRUE)
## 1                        1997-01-01
```

```
filter(cyberdata_edited, Date_of_Breach<0)
```

```
## [1] Date_of_Breach         Date_Posted_or_Updated breach_start
## [4] breach_end             year                   month_of_breach
## <0 rows> (or 0-length row.names)
```

2. Describe and demonstrate how you determine if they are outliers.

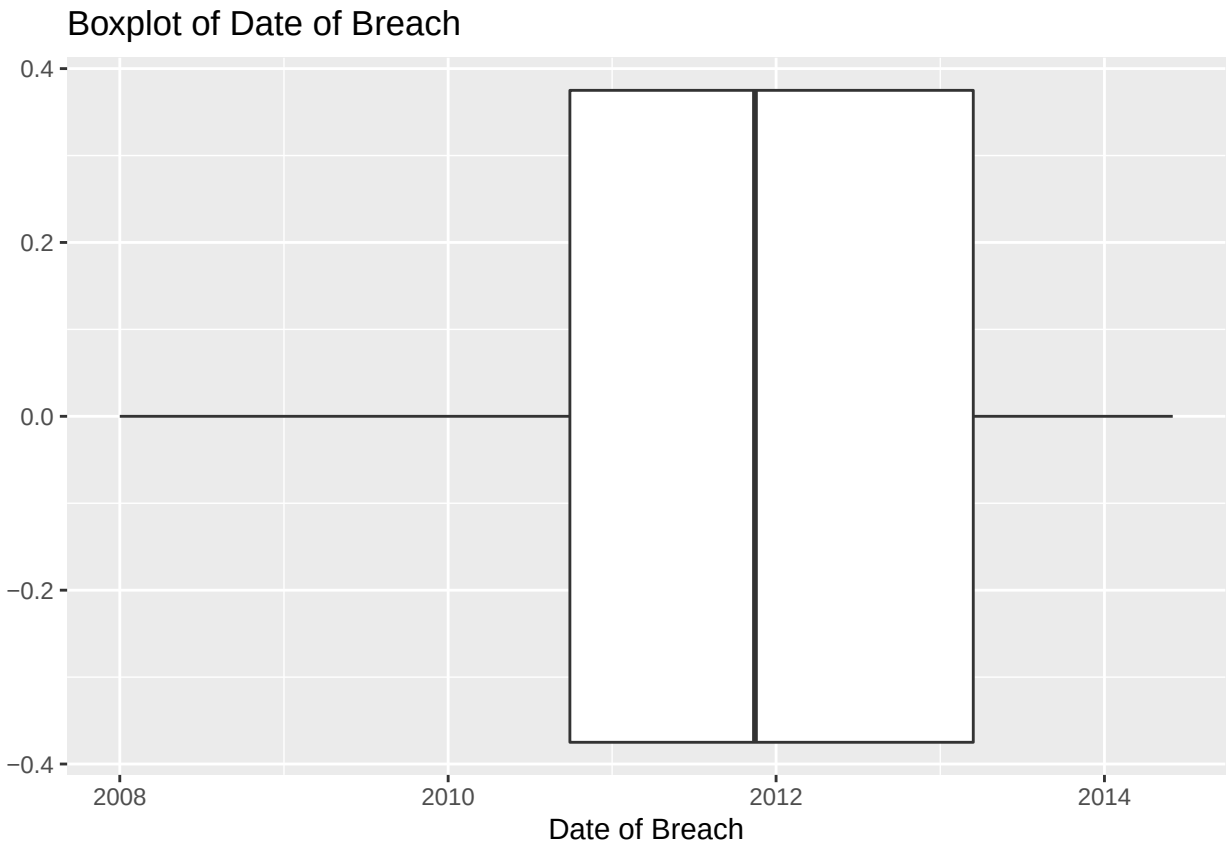
In order to determine if they are outliers, we will first check the count of the years that represent the Date of Breach to determine which dates are not in the range of 2009-2014 since this range contains a majority of the date values. From the counts below, we can tell that the years 1997, 2002, 2003, 2006 and 2007 occur only once. However, we cannot determine them to be outliers because in the boxplot, they don't appear separated from the tail.

```
cyberdata_edited %>% summarize(year) %>% count(year)
```

```
##    year   n
## 1  1997   1
## 2  2002   1
## 3  2003   1
## 4  2004   2
## 5  2005   2
## 6  2006   1
## 7  2007   1
## 8  2008  13
## 9  2009  56
## 10 2010 211
## 11 2011 229
## 12 2012 227
## 13 2013 254
## 14 2014  56
```

```
ggplot(data = cyberdata_edited, aes(x = Date_of_Breach)) +
geom_boxplot()+
labs(x="Date of Breach", title="Boxplot of Date of Breach")
```

```
## Warning: Removed 146 rows containing non-finite values (stat_boxplot).
```



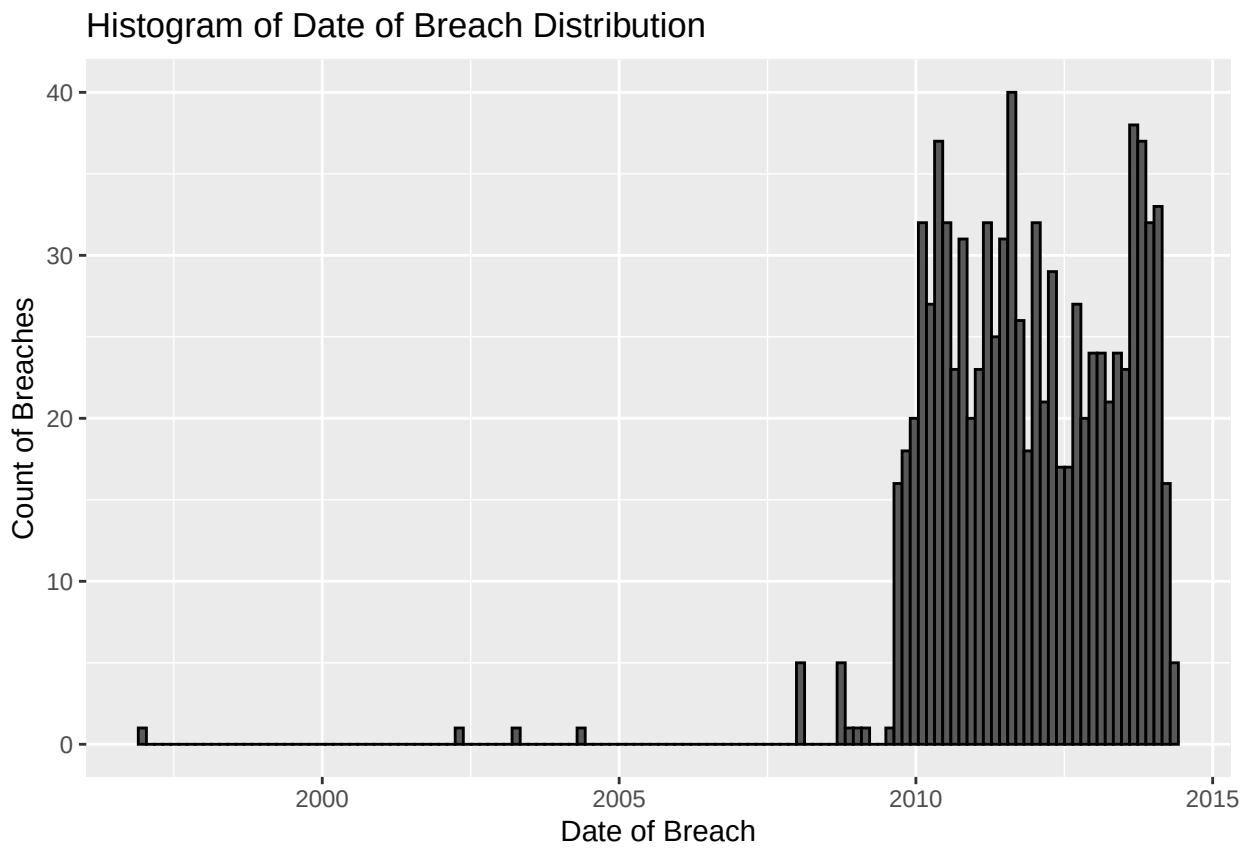
3. Show how do your distributions look like with and without the unusual values.

Since we determined there to be no outliers, we will only look at the original distribution of the Date_of_Breach variable that has no unusual values to begin with.

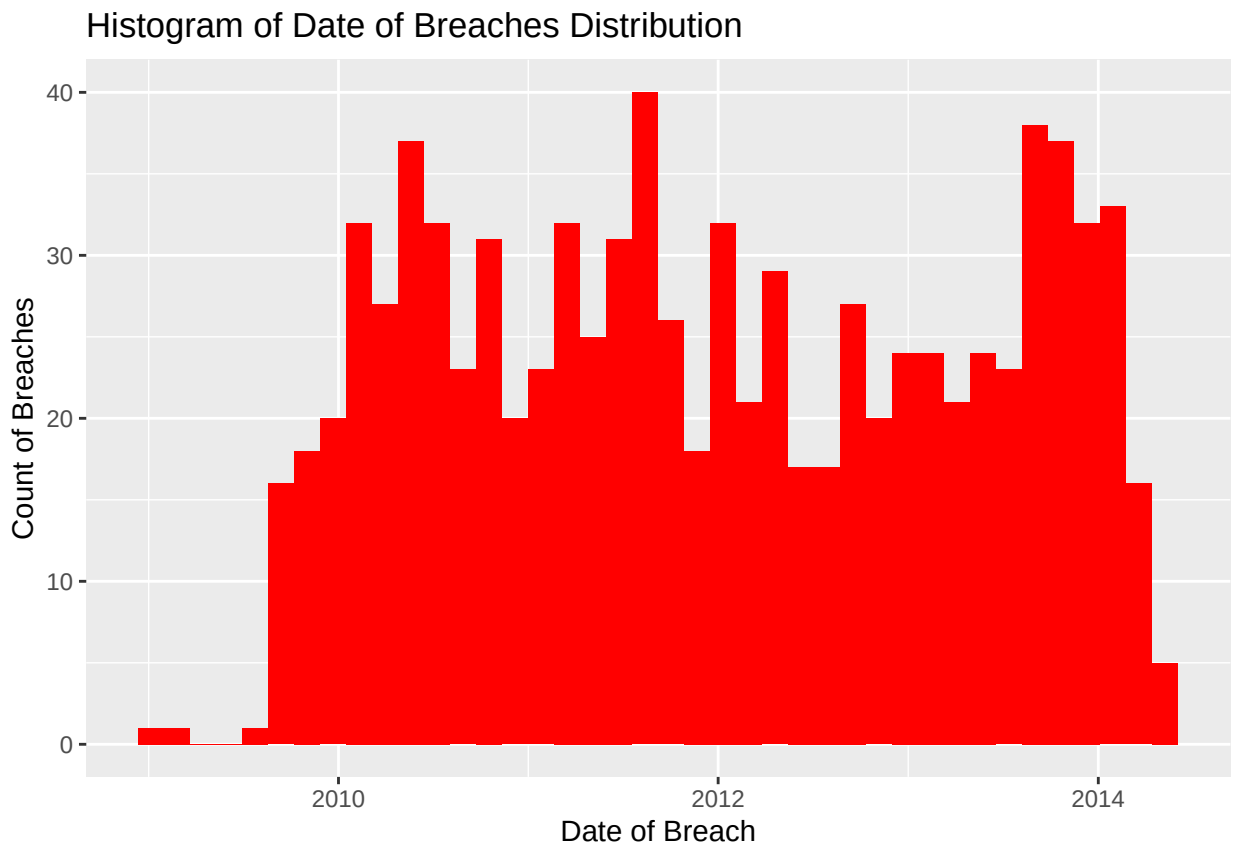
However, if we were to examine the distribution by disregarding the small counts of 2008 and below, we can do that as well.

```
ggplot(data = cyberdata_edited, mapping = aes(Date_of_Breach)) +  geom_histogram(binwidth = 50, color="
labs(x="Date of Breach", y="Count of Breaches", title="Histogram of Date of Breach Distribution")
```

```
## Warning: Removed 146 rows containing non-finite values (stat_bin).
```




```
cyberdata_DB_outliers <- cyberdata_edited %>% filter(year(Date_of_Breach) > 2008)

ggplot(data = cyberdata_DB_outliers, mapping = aes(Date_of_Breach)) + geom_histogram(binwidth = 50, fil
labs(x="Date of Breach", y="Count of Breaches", title="Histogram of Date of Breaches Distribution")
```



4. Discuss whether or not you need to remove unusual values and why.

I don't think we should remove the unusual values such as the dates that are below 2009 because they provide us with important information that there were not a lot of breaches during this time. Additionally, removing the unusual values might inhibit us from understanding when breaches were more versus less and thus removing them is not recommended.

Missing values

1. Does this variable include missing values? Demonstrate how you determine that

Yes, the Date_of_Breach variable does consist of 146 NA or missing values.

```
filter(cyberdata_edited, is.na(Date_of_Breach))
```

```
##     Date_of_Breach Date_Posted_or_Updated breach_start breach_end year
## 1             <NA>              2014-04-23   2011-09-20 2011-10-28 2011
## 2             <NA>              2014-02-14   2011-10-14 2011-10-17 2011
## 3             <NA>              2014-06-02   2011-11-19 2011-12-01 2011
## 4             <NA>              2014-01-23   2011-12-22 2011-12-23 2011
## 5             <NA>              2014-01-23   2011-11-02 2011-11-16 2011
## 6             <NA>              2014-03-24   2010-12-01 2011-11-21 2010
## 7             <NA>              2014-02-14   2009-09-23 2011-12-02 2009
## 8             <NA>              2014-01-23   2011-12-01 2011-12-17 2011
```

```
## 9            <NA>           2014-01-23   2011-12-20 2012-01-04 2011
## 10           <NA>           2014-01-23   2012-01-17 2012-02-02 2012
## 11           <NA>           2014-04-23   2008-07-01 2011-11-30 2008
## 12           <NA>           2014-01-23   2012-04-20 2012-04-21 2012
## 13           <NA>           2014-01-23   2012-03-10 2012-04-02 2012
## 14           <NA>           2014-06-03   2011-11-15 2011-12-14 2011
## 15           <NA>           2014-01-23   2011-08-01 2012-02-12 2011
## 16           <NA>           2014-01-23   2012-02-07 2012-02-20 2012
## 17           <NA>           2014-01-23   2012-01-31 2012-04-02 2012
## 18           <NA>           2014-01-23   2011-03-15 2011-08-18 2011
## 19           <NA>           2014-01-23   2012-02-22 2012-02-23 2012
## 20           <NA>           2014-01-23   2012-02-06 2012-03-14 2012
## 21           <NA>           2014-01-23   2011-06-28 2011-12-12 2011
## 22           <NA>           2014-01-23   2012-04-21 2012-04-22 2012
## 23           <NA>           2014-01-23   2004-04-21 2012-02-16 2004
## 24           <NA>           2014-01-23   2010-10-01 2012-03-21 2010
## 25           <NA>           2014-01-23   2012-03-23 2012-03-26 2012
## 26           <NA>           2014-02-20   2010-07-26 2012-03-29 2010
## 27           <NA>           2014-01-23   2012-03-29 2012-03-30 2012
## 28           <NA>           2014-01-23   2012-03-16 2012-04-20 2012
## 29           <NA>           2014-01-23   2012-05-01 2012-05-02 2012
## 30           <NA>           2014-06-03   2009-08-13 2012-04-12 2009
## 31           <NA>           2014-01-23   2012-05-03 2012-05-04 2012
## 32           <NA>           2014-01-23   2012-03-16 2012-05-11 2012
## 33           <NA>           2014-01-23   2012-06-22 2012-06-25 2012
## 34           <NA>           2014-01-23   2012-06-01 2012-06-04 2012
## 35           <NA>           2014-01-23   2012-07-15 2012-07-16 2012
## 36           <NA>           2014-04-23   2008-04-14 2011-02-28 2008
## 37           <NA>           2014-01-23   2011-01-01 2012-07-05 2011
## 38           <NA>           2014-01-23   2011-08-19 2011-09-20 2011
## 39           <NA>           2014-01-23   2012-07-31 2012-08-02 2012
## 40           <NA>           2014-06-03   2011-05-01 2011-08-05 2011
## 41           <NA>           2014-01-23   2012-01-01 2012-09-12 2012
## 42           <NA>           2014-01-23   2011-11-04 2012-04-15 2011
## 43           <NA>           2014-01-23   2012-09-17 2012-09-20 2012
## 44           <NA>           2014-01-23   2012-09-07 2012-09-09 2012
## 45           <NA>           2014-01-23   2012-09-19 2012-09-26 2012
## 46           <NA>           2014-01-23   2012-06-15 2012-10-01 2012
## 47           <NA>           2014-01-23   2012-06-15 2012-10-01 2012
## 48           <NA>           2014-01-23   2012-06-15 2012-10-01 2012
## 49           <NA>           2014-01-23   2012-09-28 2012-09-30 2012
## 50           <NA>           2014-01-23   2012-06-15 2012-10-01 2012
## 51           <NA>           2014-01-23   2012-06-15 2012-10-01 2012
## 52           <NA>           2014-01-23   2012-06-15 2012-10-01 2012
## 53           <NA>           2014-01-23   2012-06-15 2012-10-01 2012
## 54           <NA>           2014-01-23   2012-03-11 2012-10-08 2012
## 55           <NA>           2014-01-23   2012-05-01 2012-09-21 2012
## 56           <NA>           2014-01-23   2012-06-15 2012-10-01 2012
## 57           <NA>           2014-01-23   2012-10-12 2012-10-15 2012
## 58           <NA>           2014-01-23   2012-10-13 2012-10-27 2012
## 59           <NA>           2014-01-23   2012-10-13 2012-10-27 2012
## 60           <NA>           2014-01-23   2012-10-13 2012-10-27 2012
## 61           <NA>           2014-01-23   2012-08-31 2012-09-21 2012
## 62           <NA>           2014-04-23   2012-10-20 2012-10-21 2012
```

```
## 63          <NA>          2014-01-23   2012-12-10 2012-12-18 2012
## 64          <NA>          2014-01-23   2012-10-18 2012-11-04 2012
## 65          <NA>          2014-02-19   2010-01-02 2012-11-15 2010
## 66          <NA>          2014-01-23   2012-10-18 2012-10-29 2012
## 67          <NA>          2014-01-23   2007-01-01 2012-11-15 2007
## 68          <NA>          2014-05-28   2012-11-06 2012-11-15 2012
## 69          <NA>          2014-03-24   2012-05-30 2012-08-31 2012
## 70          <NA>          2014-01-23   2012-09-15 2012-11-30 2012
## 71          <NA>          2014-03-24   2012-10-27 2012-12-13 2012
## 72          <NA>          2014-01-23   2011-05-26 2012-02-18 2011
## 73          <NA>          2014-01-23   2012-12-15 2012-12-17 2012
## 74          <NA>          2014-01-23   2012-11-01 2012-12-20 2012
## 75          <NA>          2014-01-23   2012-01-09 2012-04-17 2012
## 76          <NA>          2014-01-23   2012-11-01 2012-12-20 2012
## 77          <NA>          2014-01-23   2013-01-10 2013-01-11 2013
## 78          <NA>          2014-01-23   2010-06-07 2012-12-07 2010
## 79          <NA>          2014-01-23   2011-11-01 2012-10-01 2011
## 80          <NA>          2014-01-23   2011-11-01 2012-10-01 2011
## 81          <NA>          2014-01-23   2011-11-01 2012-10-01 2011
## 82          <NA>          2014-01-23   2012-10-01 2013-02-18 2012
## 83          <NA>          2014-01-23   2013-01-18 2013-01-23 2013
## 84          <NA>          2014-01-23   2012-11-01 2013-01-19 2012
## 85          <NA>          2014-01-23   2013-03-11 2013-03-12 2013
## 86          <NA>          2014-01-23   2012-05-02 2012-06-22 2012
## 87          <NA>          2014-01-23   2009-03-01 2012-10-25 2009
## 88          <NA>          2014-06-03   2012-11-02 2013-03-14 2012
## 89          <NA>          2014-01-23   2012-12-27 2013-02-22 2012
## 90          <NA>          2014-01-23   2013-03-18 2013-03-25 2013
## 91          <NA>          2014-01-23   2013-03-01 2013-03-13 2013
## 92          <NA>          2014-01-23   2013-04-14 2013-04-19 2013
## 93          <NA>          2014-01-23   2012-06-01 2013-03-07 2012
## 94          <NA>          2014-01-23   2013-03-14 2013-03-18 2013
## 95          <NA>          2014-01-23   2013-04-14 2013-04-15 2013
## 96          <NA>          2014-01-23   2012-02-01 2013-04-11 2012
## 97          <NA>          2014-02-12   2013-03-20 2013-03-26 2013
## 98          <NA>          2014-01-23   2013-03-08 2013-05-09 2013
## 99          <NA>          2014-01-23   2013-05-01 2013-05-03 2013
## 100         <NA>          2014-01-23   2013-05-01 2013-05-02 2013
## 101         <NA>          2014-01-23   2013-04-06 2013-05-21 2013
## 102         <NA>          2014-01-23   2013-04-01 2013-05-31 2013
## 103         <NA>          2014-01-23   2013-05-25 2013-05-30 2013
## 104         <NA>          2014-01-23   2013-03-28 2013-03-29 2013
## 105         <NA>          2014-01-23   2012-09-01 2013-07-01 2012
## 106         <NA>          2014-06-10   2013-05-21 2013-05-29 2013
## 107         <NA>          2014-01-23   2012-08-01 2013-07-08 2012
## 108         <NA>          2014-01-23   2005-08-15 2007-06-14 2005
## 109         <NA>          2014-01-23   2011-10-01       <NA> 2011
## 110         <NA>          2014-01-31   2011-01-01 2013-07-03 2011
## 111         <NA>          2014-02-12   2013-05-07 2013-06-06 2013
## 112         <NA>          2014-01-23   2013-05-05 2013-06-24 2013
## 113         <NA>          2014-01-23   2011-11-09 2013-06-17 2011
## 114         <NA>          2014-01-23   2011-10-16 2013-06-07 2011
## 115         <NA>          2014-01-23   2013-01-08 2013-01-10 2013
## 116         <NA>          2014-01-23   2013-07-22 2013-08-02 2013
```

```
## 117           <NA>            2014-01-23   2013-05-05 2013-06-24 2013
## 118           <NA>            2014-06-20   2013-03-05 2013-07-16 2013
## 119           <NA>            2014-02-12   2013-06-30 2013-08-15 2013
## 120           <NA>            2014-01-23   2013-06-30 2013-08-15 2013
## 121           <NA>            2014-01-23   2006-01-01 2012-01-12 2006
## 122           <NA>            2014-01-23   2013-07-01 2013-08-02 2013
## 123           <NA>            2014-02-12   2009-12-21 2013-06-07 2009
## 124           <NA>            2014-01-23   2013-09-14 2013-09-15 2013
## 125           <NA>            2014-01-23   2013-08-08 2013-08-09 2013
## 126           <NA>            2014-01-23   2013-02-01 2013-08-27 2013
## 127           <NA>            2014-01-23   2013-03-18 2013-05-13 2013
## 128           <NA>            2014-01-23   2013-06-28 2013-07-16 2013
## 129           <NA>            2014-01-23   2013-07-03 2013-07-11 2013
## 130           <NA>            2014-01-23   2012-10-01 2012-12-31 2012
## 131           <NA>            2014-01-23   2012-10-01 2013-07-11 2012
## 132           <NA>            2014-02-18   2010-11-26 2013-10-01 2010
## 133           <NA>            2014-01-23   2013-05-01 2013-07-26 2013
## 134           <NA>            2014-01-23   2012-09-17 2013-09-17 2012
## 135           <NA>            2014-01-23   2013-01-01 2013-10-04 2013
## 136           <NA>            2014-02-18   2012-11-05 2013-11-06 2012
## 137           <NA>            2014-01-23   2013-06-18 2013-10-07 2013
## 138           <NA>            2014-01-23   2013-09-13 2013-10-15 2013
## 139           <NA>            2014-01-23   2013-09-18 2013-10-04 2013
## 140           <NA>            2014-01-23   2005-09-01 2013-08-01 2005
## 141           <NA>            2014-01-23   2013-03-06 2013-09-09 2013
## 142           <NA>            2014-01-23   2013-09-17 2013-11-08 2013
## 143           <NA>            2014-01-23   2013-09-09 2013-10-03 2013
## 144           <NA>            2014-02-11   2010-01-01 2013-11-30 2010
## 145           <NA>            2014-05-30   2013-04-13 2013-10-31 2013
## 146           <NA>            2014-06-24   2013-03-09 2013-03-11 2013
##     month_of_breach
## 1              <NA>
## 2              <NA>
## 3              <NA>
## 4              <NA>
## 5              <NA>
## 6              <NA>
## 7              <NA>
## 8              <NA>
## 9              <NA>
## 10             <NA>
## 11             <NA>
## 12             <NA>
## 13             <NA>
## 14             <NA>
## 15             <NA>
## 16             <NA>
## 17             <NA>
## 18             <NA>
## 19             <NA>
## 20             <NA>
## 21             <NA>
## 22             <NA>
## 23             <NA>
```

```
## 24            <NA>
## 25            <NA>
## 26            <NA>
## 27            <NA>
## 28            <NA>
## 29            <NA>
## 30            <NA>
## 31            <NA>
## 32            <NA>
## 33            <NA>
## 34            <NA>
## 35            <NA>
## 36            <NA>
## 37            <NA>
## 38            <NA>
## 39            <NA>
## 40            <NA>
## 41            <NA>
## 42            <NA>
## 43            <NA>
## 44            <NA>
## 45            <NA>
## 46            <NA>
## 47            <NA>
## 48            <NA>
## 49            <NA>
## 50            <NA>
## 51            <NA>
## 52            <NA>
## 53            <NA>
## 54            <NA>
## 55            <NA>
## 56            <NA>
## 57            <NA>
## 58            <NA>
## 59            <NA>
## 60            <NA>
## 61            <NA>
## 62            <NA>
## 63            <NA>
## 64            <NA>
## 65            <NA>
## 66            <NA>
## 67            <NA>
## 68            <NA>
## 69            <NA>
## 70            <NA>
## 71            <NA>
## 72            <NA>
## 73            <NA>
## 74            <NA>
## 75            <NA>
## 76            <NA>
## 77            <NA>
```

```
## 78            <NA>
## 79            <NA>
## 80            <NA>
## 81            <NA>
## 82            <NA>
## 83            <NA>
## 84            <NA>
## 85            <NA>
## 86            <NA>
## 87            <NA>
## 88            <NA>
## 89            <NA>
## 90            <NA>
## 91            <NA>
## 92            <NA>
## 93            <NA>
## 94            <NA>
## 95            <NA>
## 96            <NA>
## 97            <NA>
## 98            <NA>
## 99            <NA>
## 100           <NA>
## 101           <NA>
## 102           <NA>
## 103           <NA>
## 104           <NA>
## 105           <NA>
## 106           <NA>
## 107           <NA>
## 108           <NA>
## 109           <NA>
## 110           <NA>
## 111           <NA>
## 112           <NA>
## 113           <NA>
## 114           <NA>
## 115           <NA>
## 116           <NA>
## 117           <NA>
## 118           <NA>
## 119           <NA>
## 120           <NA>
## 121           <NA>
## 122           <NA>
## 123           <NA>
## 124           <NA>
## 125           <NA>
## 126           <NA>
## 127           <NA>
## 128           <NA>
## 129           <NA>
## 130           <NA>
## 131           <NA>
```

```
## 132            <NA>
## 133            <NA>
## 134            <NA>
## 135            <NA>
## 136            <NA>
## 137            <NA>
## 138            <NA>
## 139            <NA>
## 140            <NA>
## 141            <NA>
## 142            <NA>
## 143            <NA>
## 144            <NA>
## 145            <NA>
## 146            <NA>
```

2. Demonstrate and discuss how you handle the missing values. E.g., removing, replacing with a constant value, or a value based on the distribution, etc.

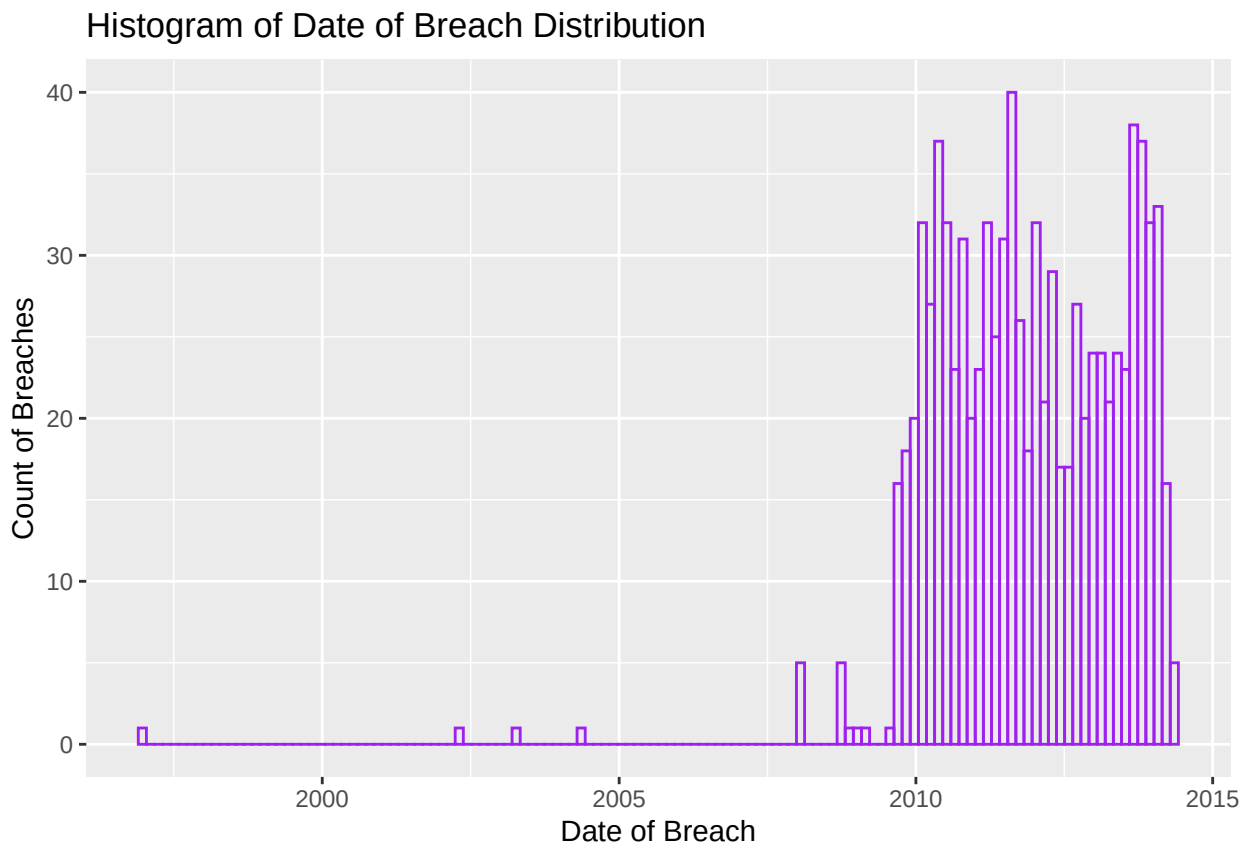
Due to converting the type of the Date_of_Breach variable to Date type, the rows that had a range of dates in this column obtained NA values. However, I didn't lose the data representing the range of dates because the breach_start and breach_end columns showcase the starting and ending date of the range respectively.

Hence, instead of removing the rows with NAs, I will keep them since they represent the ranges. I will later look at the breach length which represents the ranges that were supposed to exist in the Date_of_Breach column.

3. Show how your data looks in each case after handling missing values.Describe and discuss the distribution.

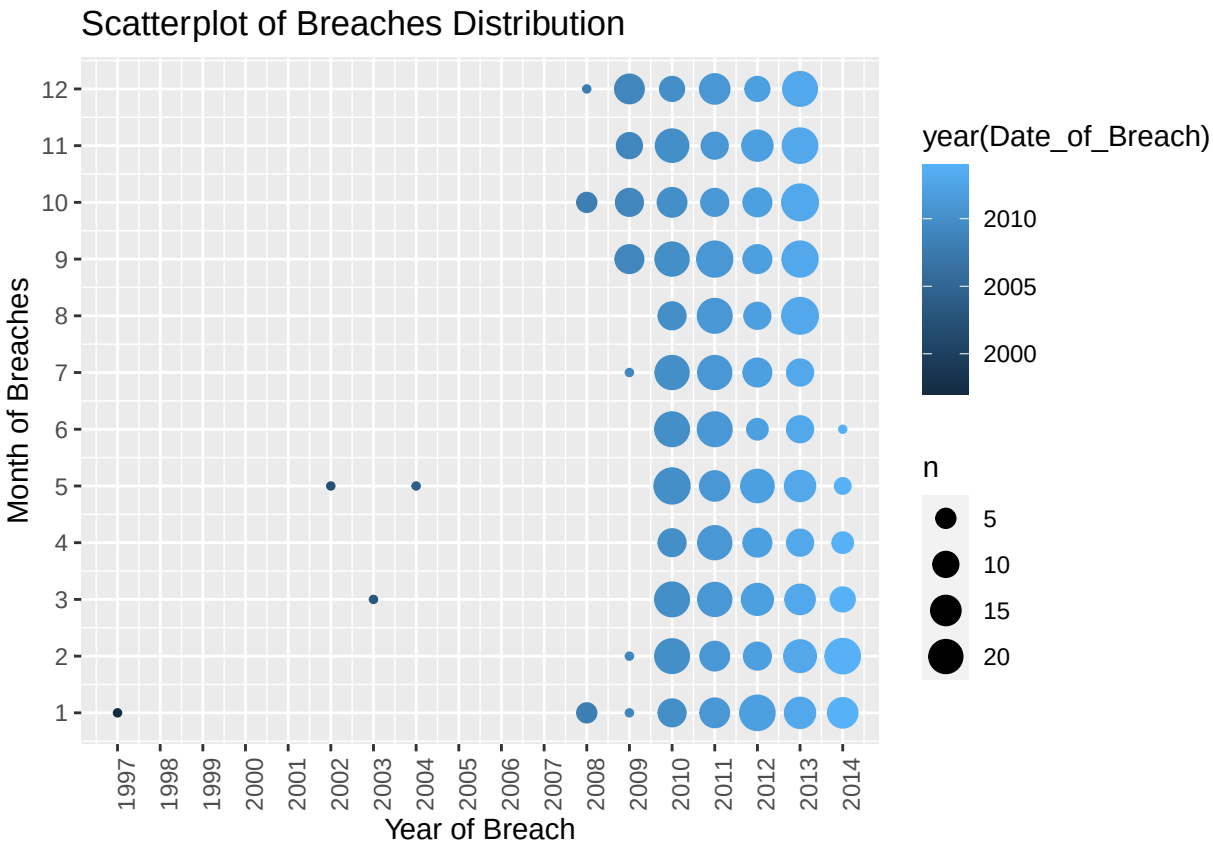
If we were to remove the NA values to analyze them better, we can look at the distribution below to see what it would look like. From the scatterplot distribution, we can clearly see the monthly distribution of the values in each year. From the years 2010 to 2013, all the 12 months have breaches in them. And there seem to be no breaches in the years 1998 through 2001 and 2005 through 2007.

```
cyberdata_noNA <- cyberdata_edited %>% filter(!is.na(Date_of_Breach))

ggplot(data = cyberdata_noNA, mapping = aes(Date_of_Breach)) +
  geom_histogram(binwidth = 50, fill=NA, color="purple") +
  labs(x="Date of Breach", y="Count of Breaches", title="Histogram of Date of Breach Distribution")
```



Important

```
ggplot(data = cyberdata_noNA) +
  geom_count(mapping = aes(x = year(Date_of_Breach), y = month(Date_of_Breach), color = year(Date_of_Br
scale_y_continuous(breaks=c(1:12)) +
scale_x_continuous(breaks=c(1997:2014)) +
theme(axis.text.x = element_text(angle = 90)) +
labs(x="Year of Breach", y="Month of Breaches", title="Scatterplot of Breaches Distribution")
```



Does converting the type of this variable help exploring the distribution of its values or identifying outliers or missing values?

1. What type can the variable be converted to?

Since originally, the type of the Date_of_Breach variable was character and we converted it to the Date type to conduct our analysis, there is no further conversion of types that we need to do.

2. How will the distribution look? Please demonstrate with appropriate plots.

What new variables do you need to create from this

1. List the variables
2. Describe and discuss why they are needed and how you plan to use them.

There are no new variables we need to create, however, we must analyze the breach_start and breach_end variables to further understand breaches when they occur over a period of time.

First however, we will look at how the Date_of_Breach variable relates to the Date_Posted_or_Updated variable. We will do this to understand the amount of time it takes for breaches to be recognized after they occur. Again, accounting for the NA values which represent the range in the Date_of_Breach variable, we will look at them later with the breach_end variable.

Variable 2: Date_Posted_or_Updated

1. Which values are most common and why? Which values are rare? Why? Does that match your expectations?

After segmenting the counts of dates into those that are less than 10 and those that are more than 10, the most common value is the date of 2014-01-23 with 691 counts, then followed by 2014-03-24 with 48 counts. These values with higher counts indicate that a majority of the breaches were found during this time.

As for values that are rare, we find many dates with counts of 1 such as 2014-01-31, 2014-02-25, 2014-02-26, 2014-03-05, 2014-03-21, 2014-03-31, 2014-04-22, 2014-05-06 and 2014-06-11. This means these very the rare dates when a breach was found.

```
cyberdata_edited %>% group_by(Date_Posted_or_Updated) %>% count() %>% filter(n<=10)
```

```
## # A tibble: 30 x 2
## # Groups:   Date_Posted_or_Updated [30]
##    Date_Posted_or_Updated     n
##    <date>                 <int>
##  1 2014-01-24                 2
##  2 2014-01-31                 1
##  3 2014-02-14                10
##  4 2014-02-18                 7
##  5 2014-02-19                 5
##  6 2014-02-20                 2
##  7 2014-02-21                 6
##  8 2014-02-24                 2
##  9 2014-02-25                 1
## 10 2014-02-26                 1
## # ... with 20 more rows
```

```
cyberdata_edited %>% group_by(Date_Posted_or_Updated) %>% count() %>% filter(n>10)
```

```
## # A tibble: 13 x 2
## # Groups:   Date_Posted_or_Updated [13]
##    Date_Posted_or_Updated     n
##    <date>                 <int>
##  1 2014-01-23               691
##  2 2014-02-11                13
##  3 2014-02-12                21
##  4 2014-03-13                15
##  5 2014-03-24                48
##  6 2014-04-21                28
##  7 2014-04-23                25
##  8 2014-05-27                11
##  9 2014-06-03                18
## 10 2014-06-18                12
## 11 2014-06-19                21
## 12 2014-06-20                21
## 13 2014-06-30                13
```

3. Can you see any unusual patterns? What might explain them?

When finding the count of the Date_Posted_or_Updated data set using its year, we find that the entire column is only of the year 2014. Hence, when we make our distribution, we see the distribution of breaches in each month of the year 2014. The month of January has the highest count of breaches that are posted followed by June. This is also confirmed from the count table.

But we don't find any unusual trends in the data.

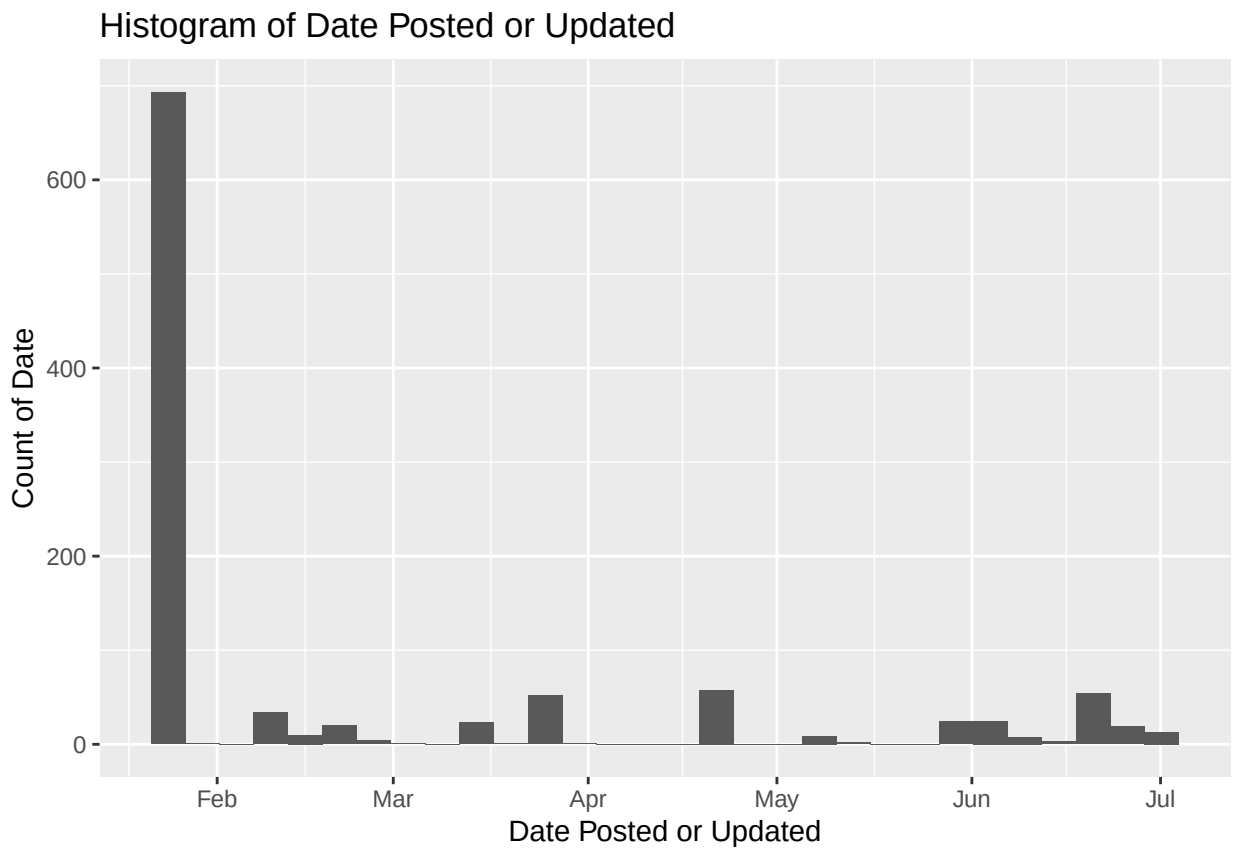
Important

```
cyberdata_edited %>% group_by(year(Date_Posted_or_Updated)) %>% count()
```

```
## # A tibble: 1 x 2
## # Groups:   year(Date_Posted_or_Updated) [1]
##   `year(Date_Posted_or_Updated)`     n
##                             <dbl> <int>
## 1                            2014  1055
```

```
cyberdata_edited %>% group_by(Date_Posted_or_Updated) %>% ggplot(aes(Date_Posted_or_Updated)) +
geom_histogram() +
labs(x="Date Posted or Updated", y="Count of Date", title="Histogram of Date Posted or Updated")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
cyberdata_edited %>% group_by(month(Date_Posted_or_Updated)) %>% count()
```

```
## # A tibble: 6 x 2
## # Groups:   month(Date_Posted_or_Updated) [6]
##   `month(Date_Posted_or_Updated)`     n
##                             <dbl> <int>
## 1                               1   694
## 2                               2    68
## 3                               3    78
## 4                               4    58
## 5                               5    35
## 6                               6   122
```

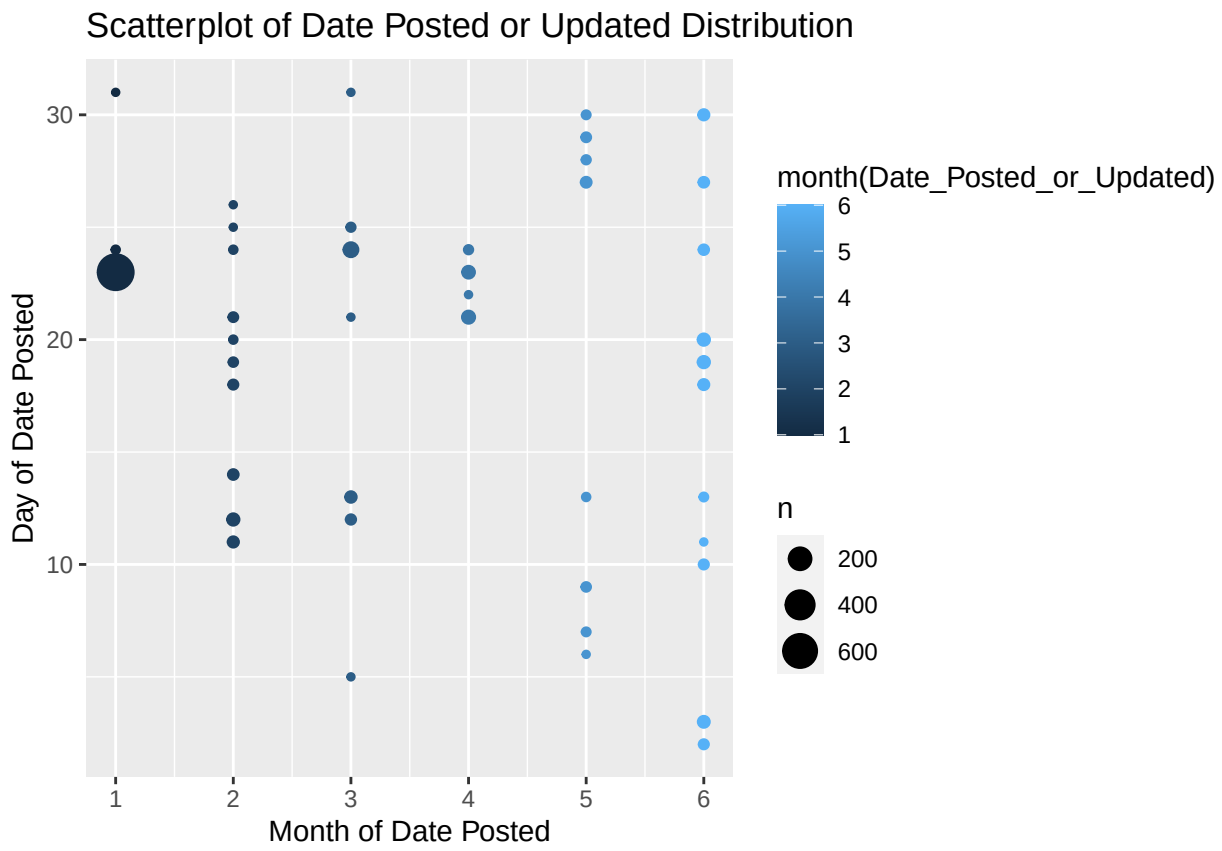
4. How are the observations within each cluster similar to or different from each other? How can you explain or describe the clusters?

When examining the dates in this variable, we see that the months of January and April have less spread out breach dates as compared to February, March, May and June.

While there is no explicit reasoning for why these clusters occur, the best explanation is because the months of January and April have the certain dates when they focus on detecting breaches and so not a lot of days are spent on detecting them. On the other hand, in the rest of the months, there are more days when they work on detecting the breaches and thus there are more spread out data points.

Important

```
ggplot(data = cyberdata_edited) +
geom_count(mapping = aes(x = month(Date_Posted_or_Updated), y = day(Date_Posted_or_Updated), color = mo
labs(x="Month of Date Posted", y="Day of Date Posted", title="Scatterplot of Date Posted or Updated Dis
```



Unusual values

1. Describe and demonstrate how you determine if there are unusual values in the data. E.g. too large, too small, negative, etc.

The latest date or the largest date in the data set is 2014-06-30 or the 30th of June, 2014. The earliest or the smallest date in the data set is 2014-01-23 or the 23rd of January, 2014. There are no negative date values determined in the data set which is expected since these are dates.

```
max_dates_posted <- summarize(cyberdata_edited, max(Date_Posted_or_Updated, na.rm = TRUE))
max_dates_posted
```

```
##   max(Date_Posted_or_Updated, na.rm = TRUE)
## 1                                2014-06-30
```

```
min_dates_posted <- summarize(cyberdata_edited, min(Date_Posted_or_Updated, na.rm = TRUE))
min_dates_posted
```

```
##   min(Date_Posted_or_Updated, na.rm = TRUE)
## 1                                2014-01-23
```

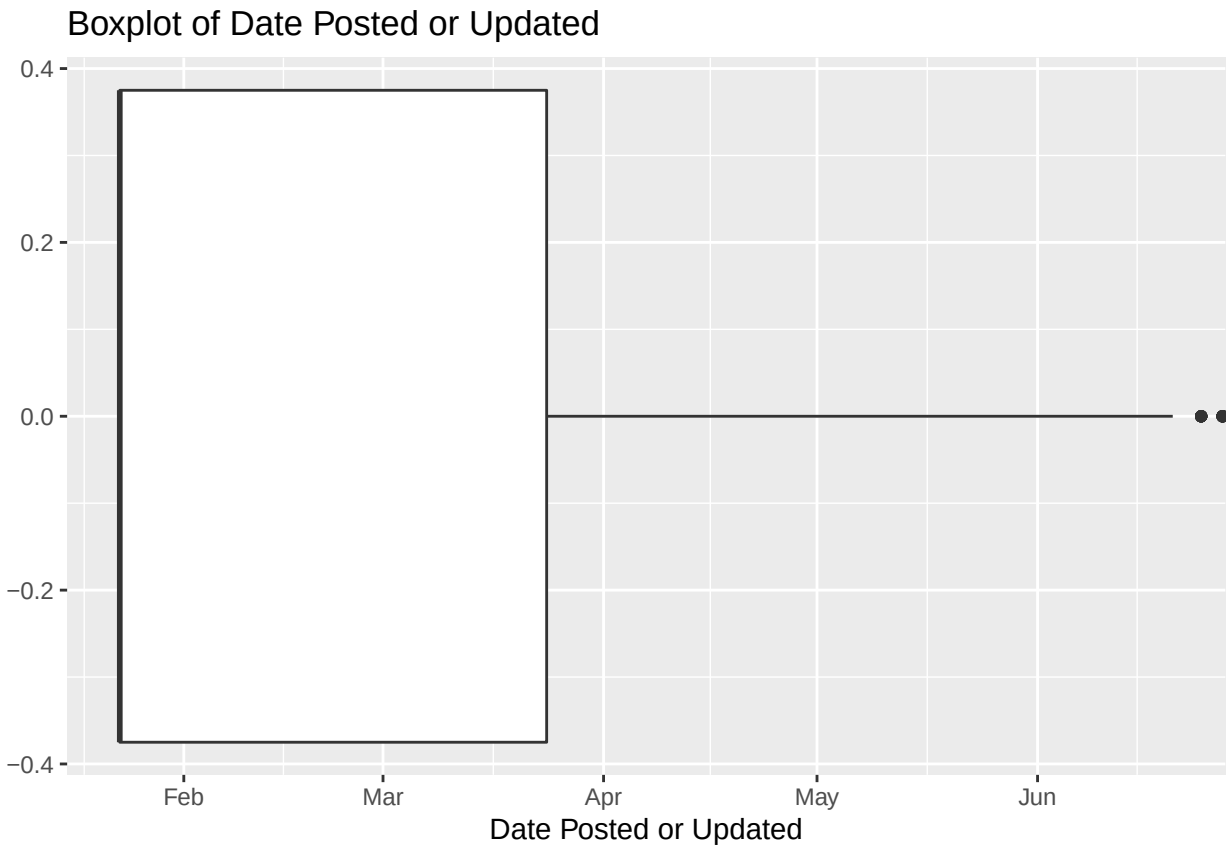
```
filter(cyberdata_edited, Date_Posted_or_Updated<0)
```

```
## [1] Date_of_Breach         Date_Posted_or_Updated breach_start
## [4] breach_end             year                   month_of_breach
## <0 rows> (or 0-length row.names)
```

2. Describe and demonstrate how you determine if they are outliers.

In order to determine if they are outliers, we will create a boxplot and look for points that are separated from the tail. We see that there are 2 points that act as outliers.

```
ggplot(data = cyberdata_edited, aes(x = Date_Posted_or_Updated)) + geom_boxplot()+
labs(x="Date Posted or Updated", title="Boxplot of Date Posted or Updated")
```

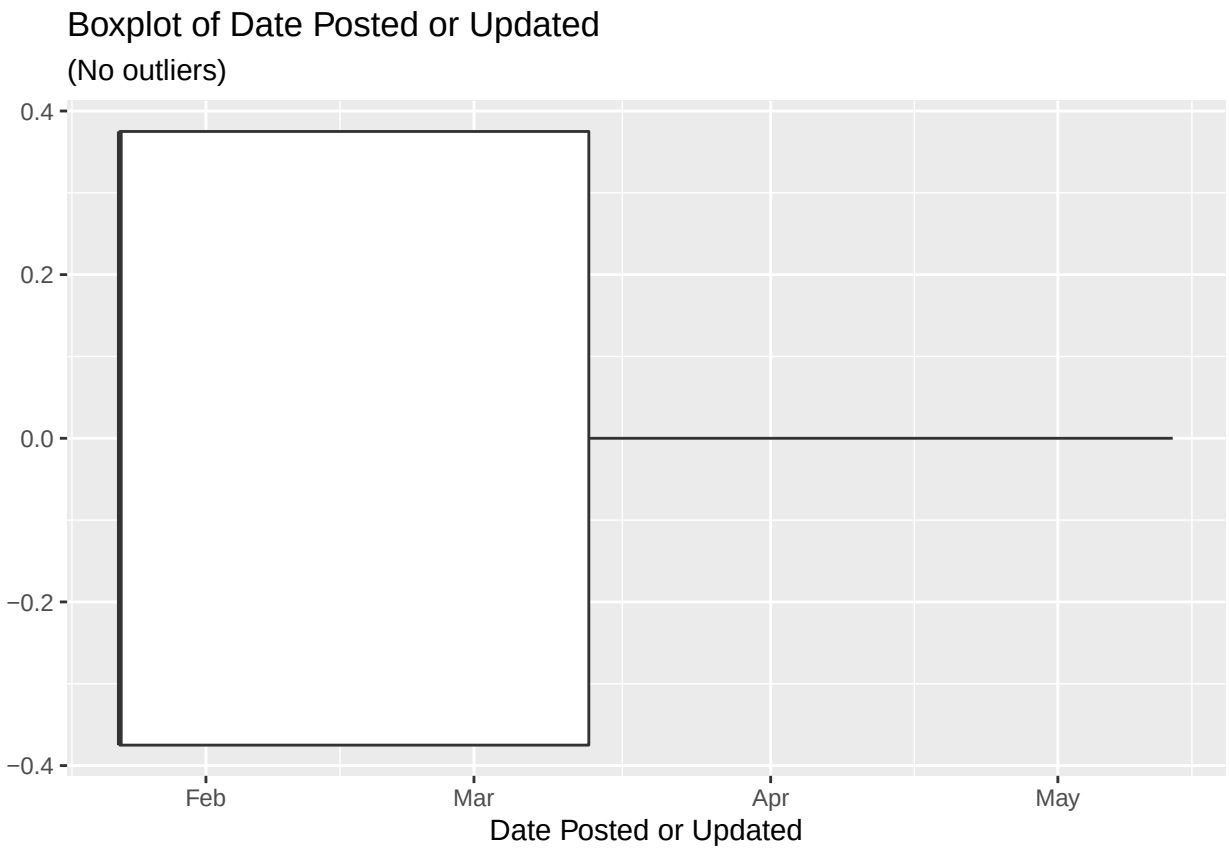


3. Show how do your distributions look like with and without the unusual values.

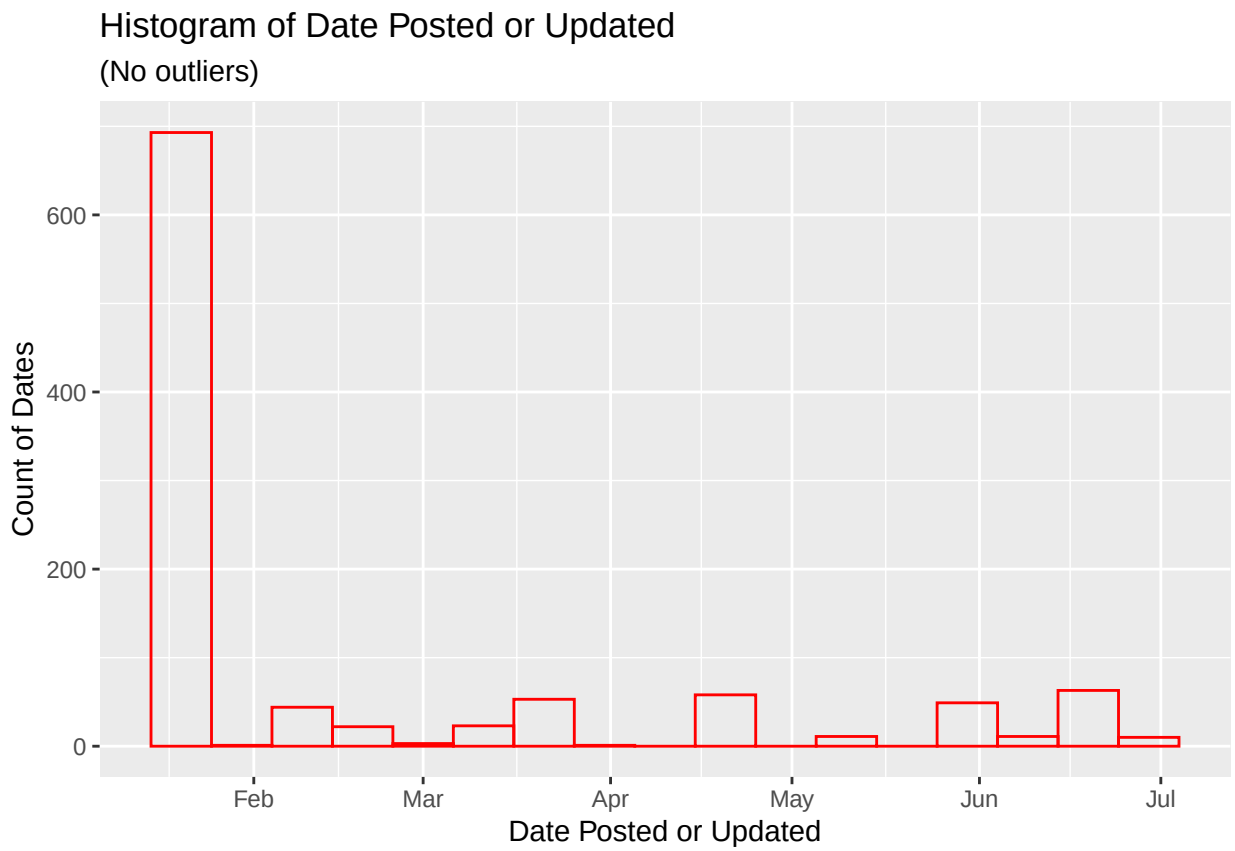
Since the only unusual values are the outliers, we will examine the distribution without them. After filtering out the values that have dates later than 2014-06-29, which consists of 13 values, we can recreate the boxplot and we see that it doesn't consist of the outliers.

Additionally, the new distribution in red shows a very minute difference only in the last bin which is shorted in height.

```r
cyberdata_DPoU_outliers <- cyberdata_edited %>%
group_by(Date_Posted_or_Updated) %>%
filter(!(month(Date_Posted_or_Updated) == 6 & day(Date_Posted_or_Updated) > 29))

ggplot(data = cyberdata_DPoU_outliers, aes(x = Date_Posted_or_Updated)) + geom_boxplot() +
labs(x="Date Posted or Updated", title="Boxplot of Date Posted or Updated", subtitle="(No outliers)")
```




```r
ggplot(data = cyberdata_DPoU_outliers, mapping = aes(Date_Posted_or_Updated)) + geom_histogram(binwidth
labs(x="Date Posted or Updated", y="Count of Dates", title="Histogram of Date Posted or Updated", subti
```



4. Discuss whether or not you need to remove unusual values and why.

No, we should not remove the missing values because they aren't that different from the other values in the data. Even though those data points are classified as outliers by the box plot distribution, they are not values that should be removed. These values are part of the distribution and are valuable in determining the length of time it takes for a breach to be detected.

Missing Values

1. Does this variable include missing values? Demonstrate how you determine that

No, this variable does not include any NA values.

```
filter(cyberdata_edited, is.na(Date_Posted_or_Updated))
```

```
## [1] Date_of_Breach         Date_Posted_or_Updated breach_start
## [4] breach_end             year                   month_of_breach
## <0 rows> (or 0-length row.names)
```

Analyzing length of detection of breach

In order to determine the time it takes for a breach to be detected, we will subtract the Date_of_Breach variable from the Date_Posted_or_Updated variable to give us the difference between the two. This variable has days as its unit of measurement.

```
cyberdata_edited <- cyberdata_edited %>%
mutate(detection_length = Date_Posted_or_Updated - Date_of_Breach)
```

Comparing detection length with Date posted

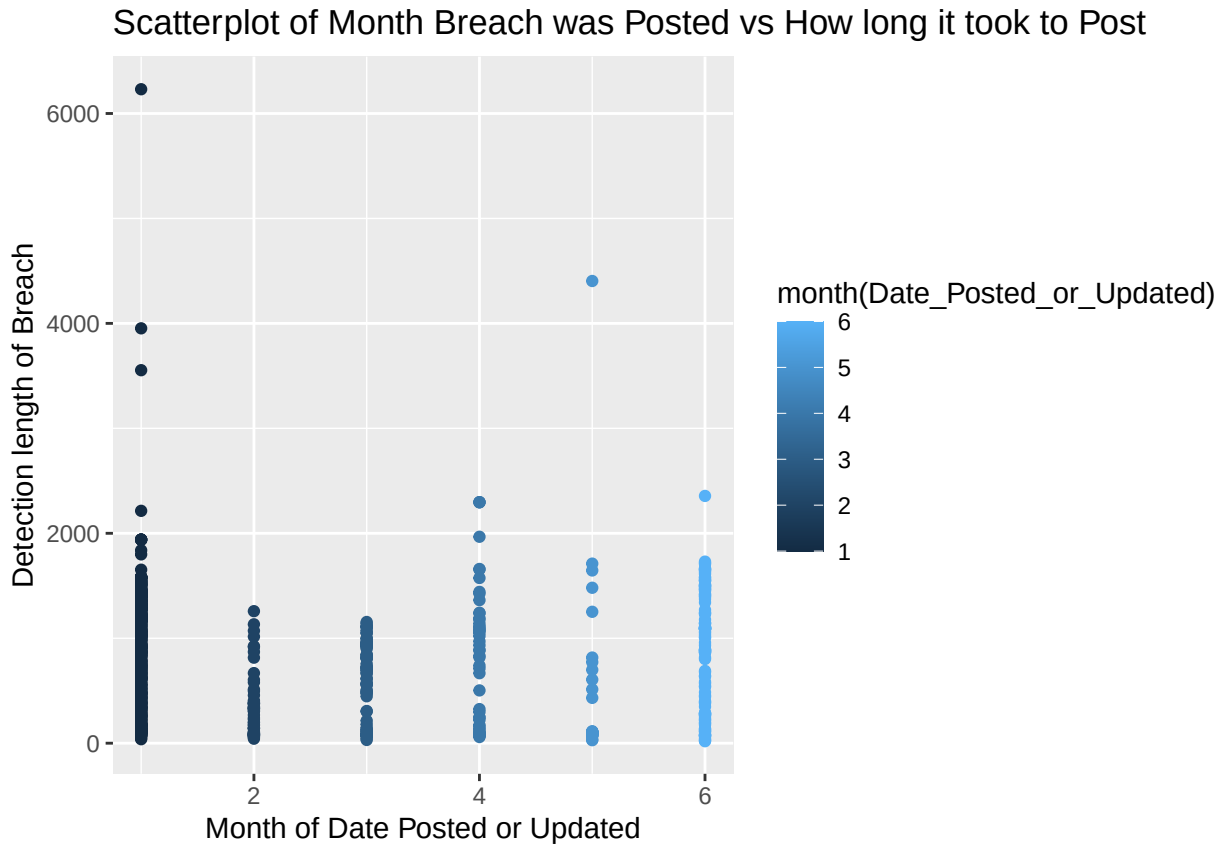
The scatterplot shows us that the month of January has higher detection_length values that are scattered and reach above 6000. The month of February and March have the smallest distribution in values, with values lesser than 1500. Similarly, April and June also consist of values that are within the range of 2500 detection_length values. For the month of May, the values are mostly less than 2000 except for one extreme value.

The same conclusions can be made from looking at the boxplot distribution. The month of January and May show the outliers. Outliers occur in the later half of January and towards the middle of the month for May.

```
ggplot(data = cyberdata_edited) +
geom_point(mapping = aes(x = month(Date_Posted_or_Updated), y =
detection_length, color = month(Date_Posted_or_Updated))) +
labs(x="Month of Date Posted or Updated", y="Detection length of Breach", title="Scatterplot of Month B
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 146 rows containing missing values (geom_point).
```

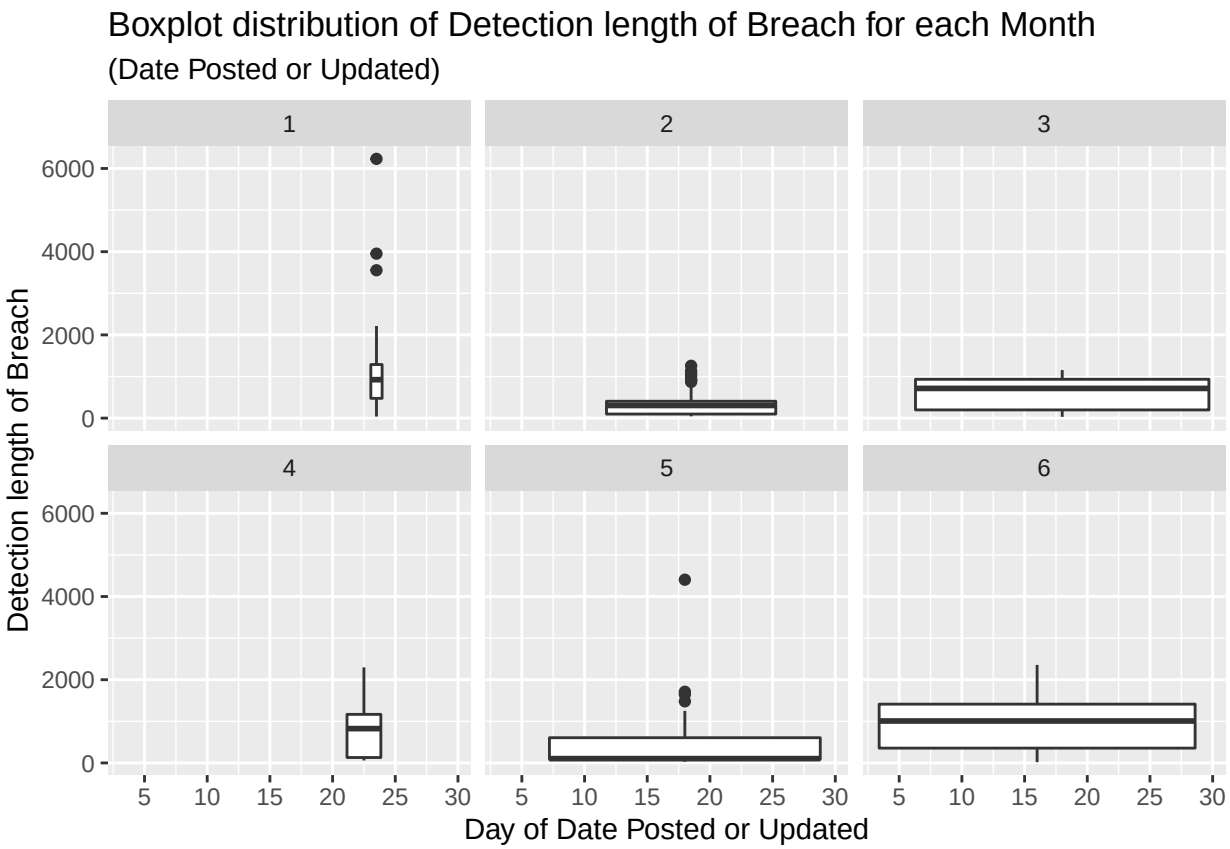


```
ggplot(data = cyberdata_edited, mapping = aes(x = day(Date_Posted_or_Updated), y = detection_length)) +
geom_boxplot() +
facet_wrap(~month(Date_Posted_or_Updated)) + scale_x_continuous(breaks=c(1,5,10,15,20,25,30)) +
labs(x="Day of Date Posted or Updated", y="Detection length of Breach", title="Boxplot distribution of
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```

```
## Warning: Removed 146 rows containing non-finite values (stat_boxplot).
```



Graph below doesn't show proper trend even though linear because detection length is mostly indicative of the year 2014 and so examining the yearly trend will most definitely show decreasing trend.

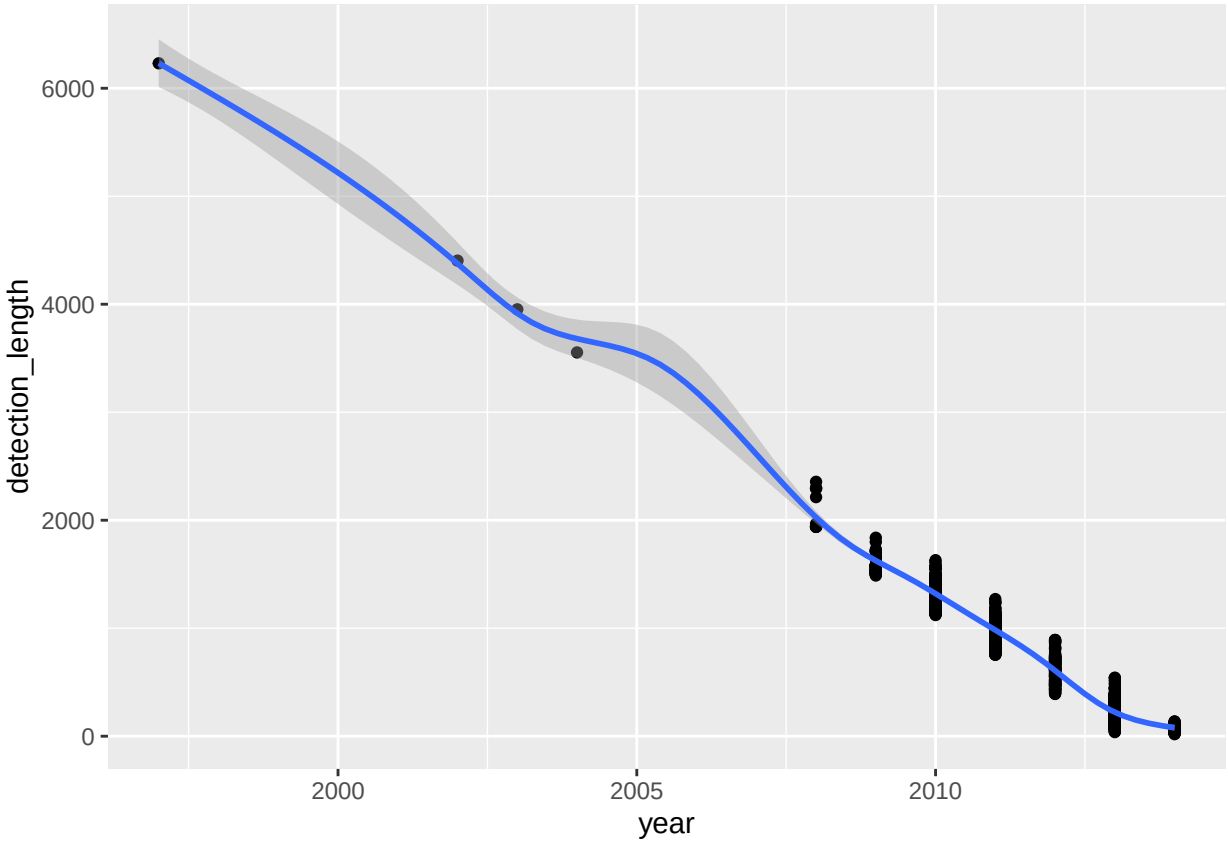
```
ggplot(data = cyberdata_edited, aes(x=year, y=detection_length)) +
geom_point() + geom_smooth()
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 146 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 146 rows containing missing values (geom_point).
```



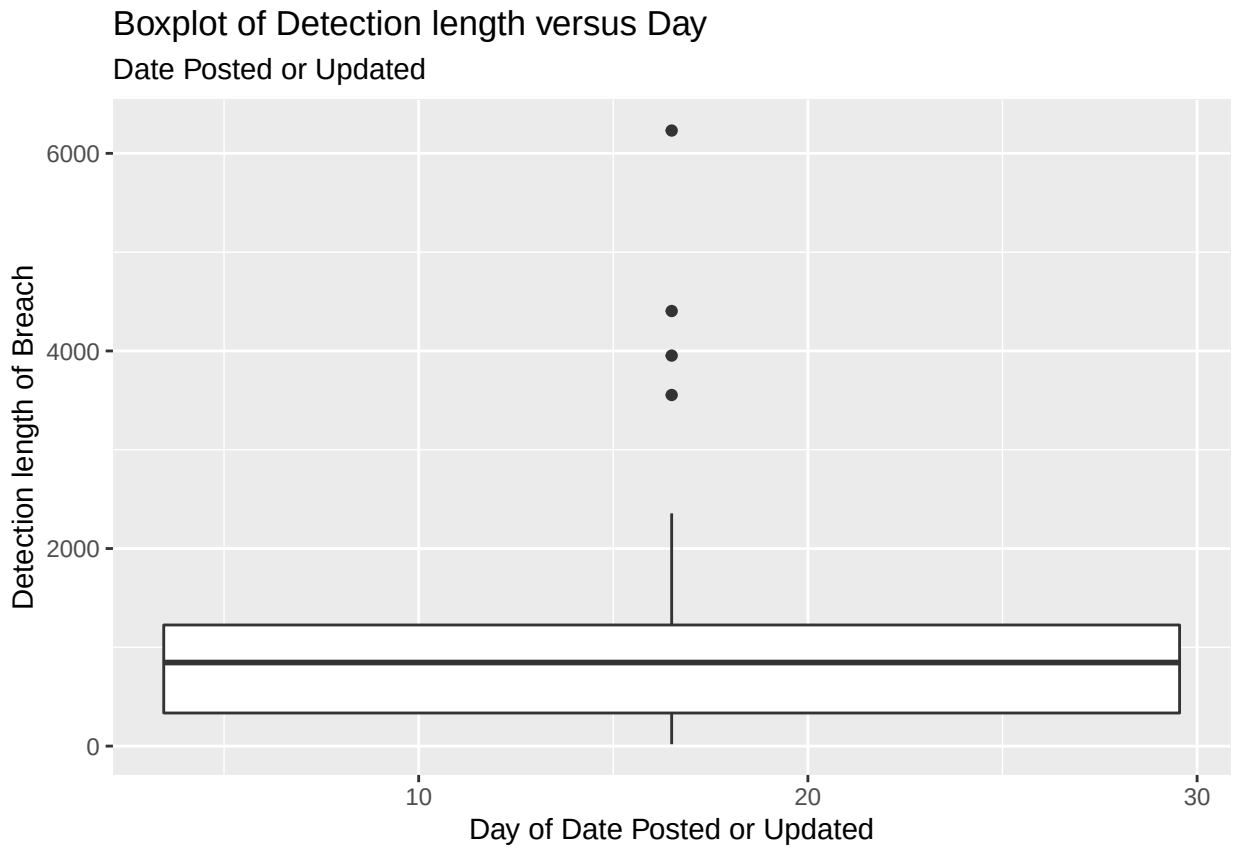
The graph below shows that a majority of the days that have outliers in the Date Posted or Updated variable is around day 17. The median of the boxplot is a little less than the 1000 Detectin length of Breach mark, meaning that most detection lengths are small.

```
ggplot(data = cyberdata_edited, mapping = aes(x = day(Date_Posted_or_Updated), y = detection_length)) +
geom_boxplot() +
labs(x="Day of Date Posted or Updated", y="Detection length of Breach", title="Boxplot of Detection leng
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```

```
## Warning: Removed 146 rows containing non-finite values (stat_boxplot).
```



This analysis and distribution tells us that the months of January and May have a higher range of distribution lengths, meaning that for some breaches, it takes longer for the breach to be detected. Hence, in the months of February, March, April and June, the detection lengths are not that widely distributed, meaning that the detection of the breach happens in fairly less time. The month of March, however, has the quickest detection time as indicated by the smallest distribution of detection lengths.

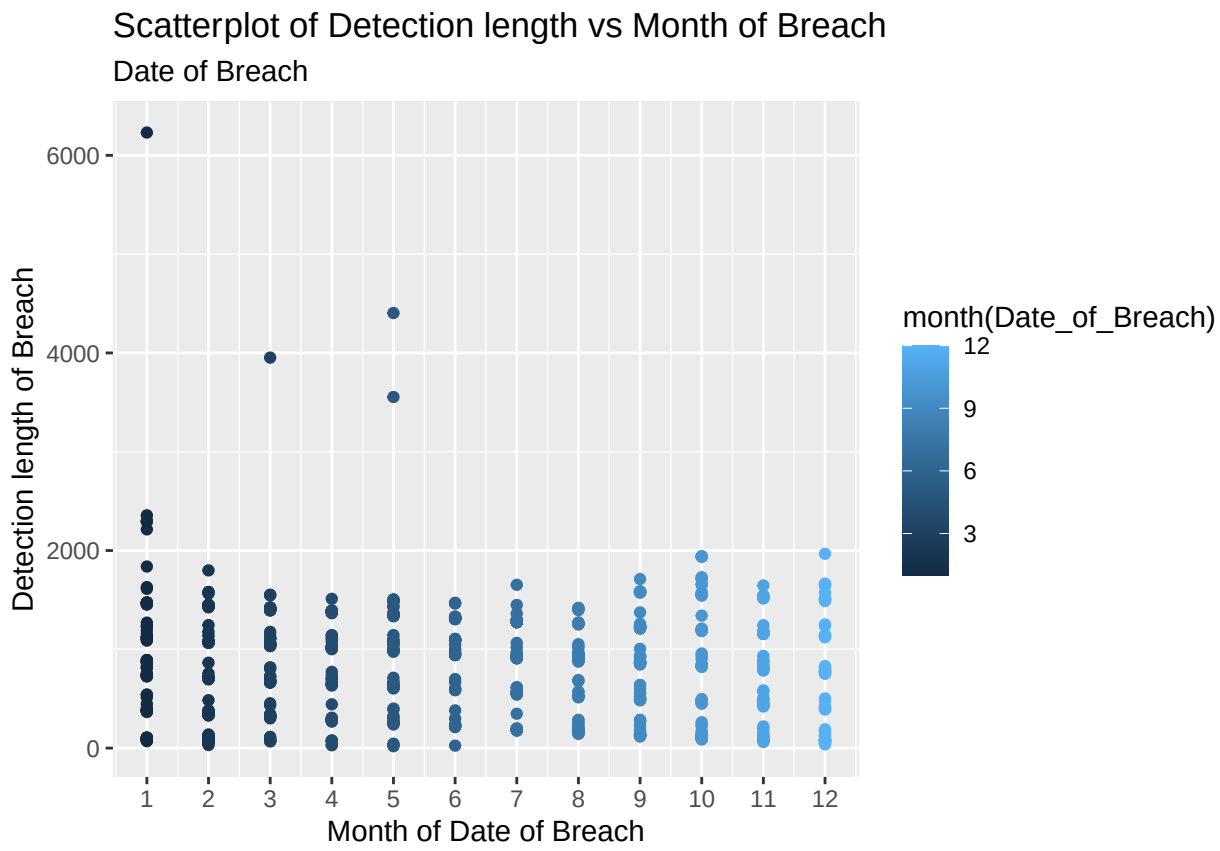
Comparing detection length with Date of Breach

I created three different graphs such as one with geom_point(), one with a boxplot for months, and one final boxplot for years (reordered). I did this to ensure that my trend and the pattern I noticed was noticeable throughout.

```
ggplot(data = cyberdata_edited) +
geom_point(mapping = aes(x = month(Date_of_Breach), y = detection_length, color = month(Date_of_Breach)
scale_x_continuous(breaks=c(1:12)) +
labs(x="Month of Date of Breach", y="Detection length of Breach", title="Scatterplot of Detection lengt
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 146 rows containing missing values (geom_point).
```

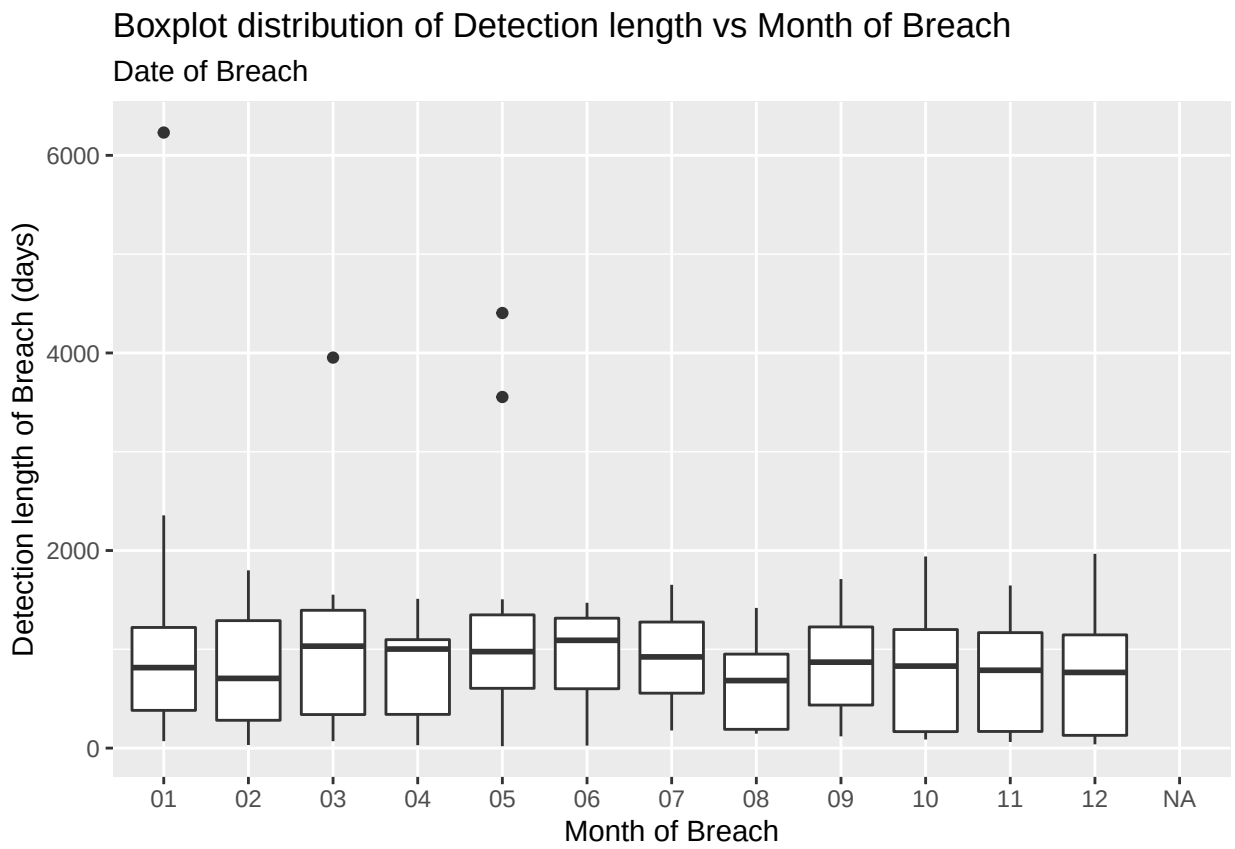


Important

```
ggplot(data = cyberdata_edited) +
geom_boxplot(mapping = aes(x = month_of_breach, y= detection_length)) + labs(x="Month of Breach", y="De
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 146 rows containing non-finite values (stat_boxplot).
```

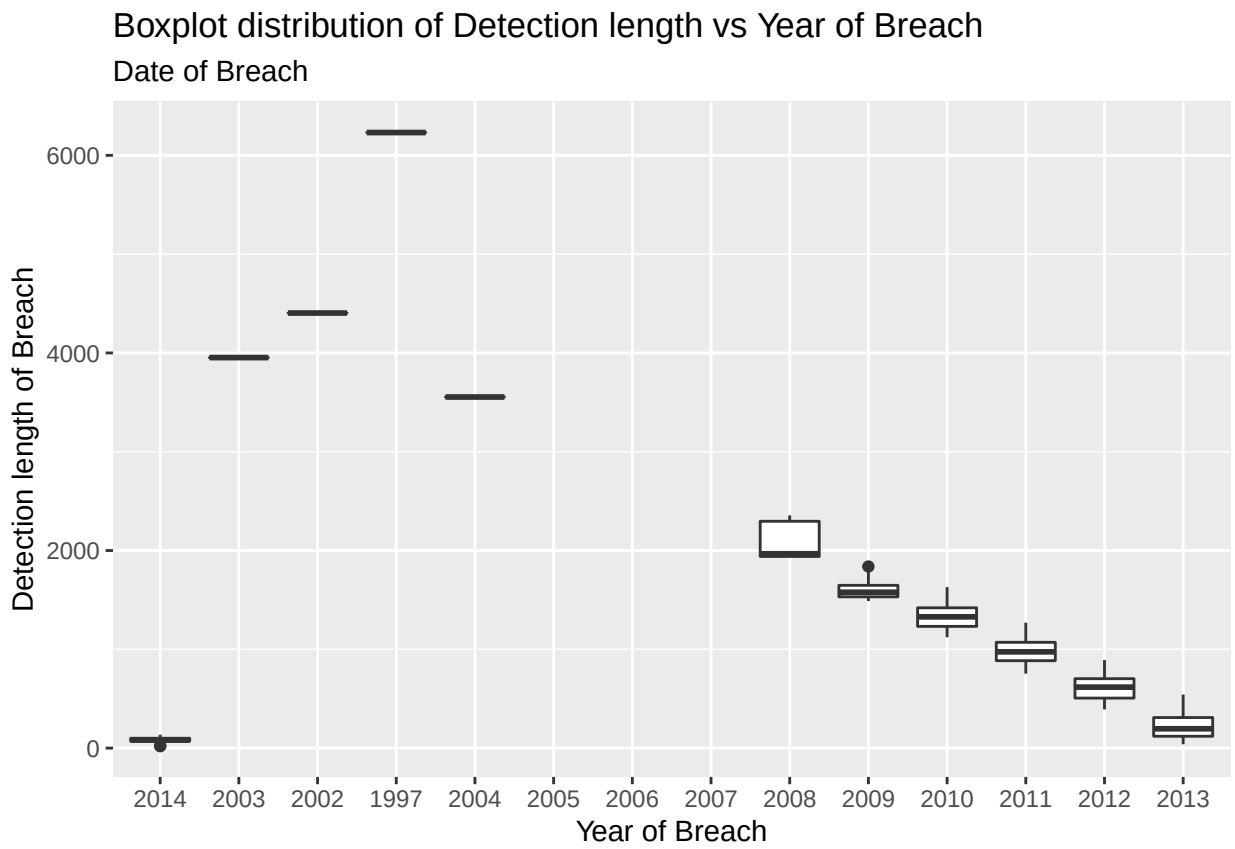


I reorderd the boxplot of Detection length of Breach versus the year of breach to notice how the median changed throughout the years.

```
ggplot(data = cyberdata_edited) +
geom_boxplot(mapping = aes(x = reorder(year,detection_length, FUN = median), y=detection_length)) +
labs(x="Year of Breach", y="Detection length of Breach", title="Boxplot distribution of Detection lengt
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 146 rows containing non-finite values (stat_boxplot).
```



From our analysis above, we see that creating a distribution of the detection length with the month of the Date of Breach showcases that the detection lengths have some outliers in the months of January, March and May. The trend here is that the detection of the breach seems to be quick from January till June, and then increases in July, decreases in August, increases from September to October and then decreases in November, with finally increasing in December. Even the boxplot distribution shows that the highest median is in the month of June.

As for looking at the years boxplot, we can see that the years from 2008 to 2013 the median of the detection length decreases. This means that from 2008 to 2013, the detection length decreases meaning that the time between the breach occuring and the breach being posted, reduces, hence it is put into the system quicker. However, this graph isn't very useful because the Date Posted variable is all within the same year of 2014 and thus it is expected to see the decreasing trend over the years.

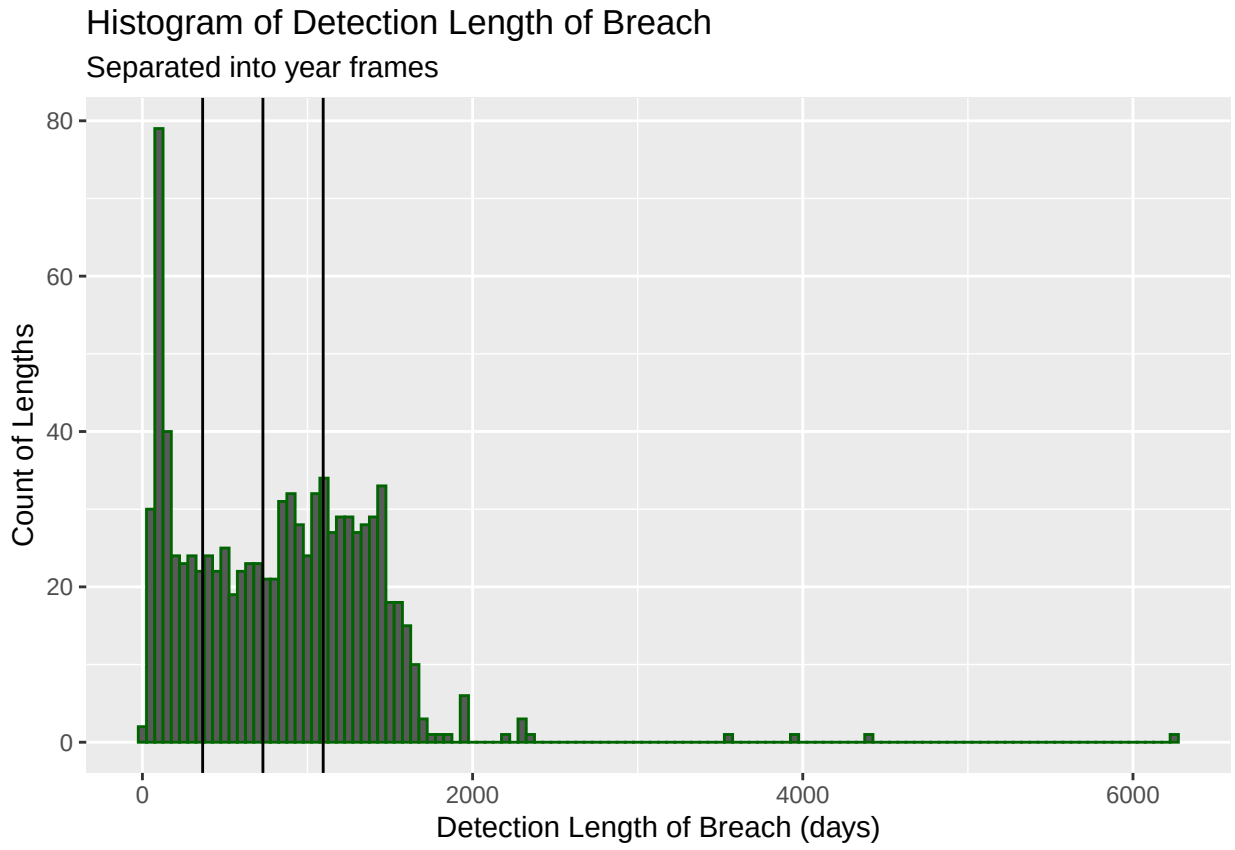
The detection length histogram created to understand how long detection takes by examining the yearly frames. The first line is the 1 year mark, second is for the 2nd year mark and so on. We can see the decreasing trend. The majority of the lengths are for 0 detection lengths, meaning that most breaches were detected fast.

Important

```
ggplot(data = cyberdata_edited, mapping = aes(detection_length)) + geom_histogram(binwidth = 50, color=
labs(x="Detection Length of Breach (days)", y="Count of Lengths", title="Histogram of Detection Length
geom_vline(xintercept = 365) +
geom_vline(xintercept = 730) +
geom_vline(xintercept = 1095)
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 146 rows containing non-finite values (stat_bin).
```



Comparing detection length with State

Since the cyberdata_edited data set only contains the date variables, we will create a new data frame and add the variables we need to compare the detection lengths against such as the state and the number of individuals affected.'

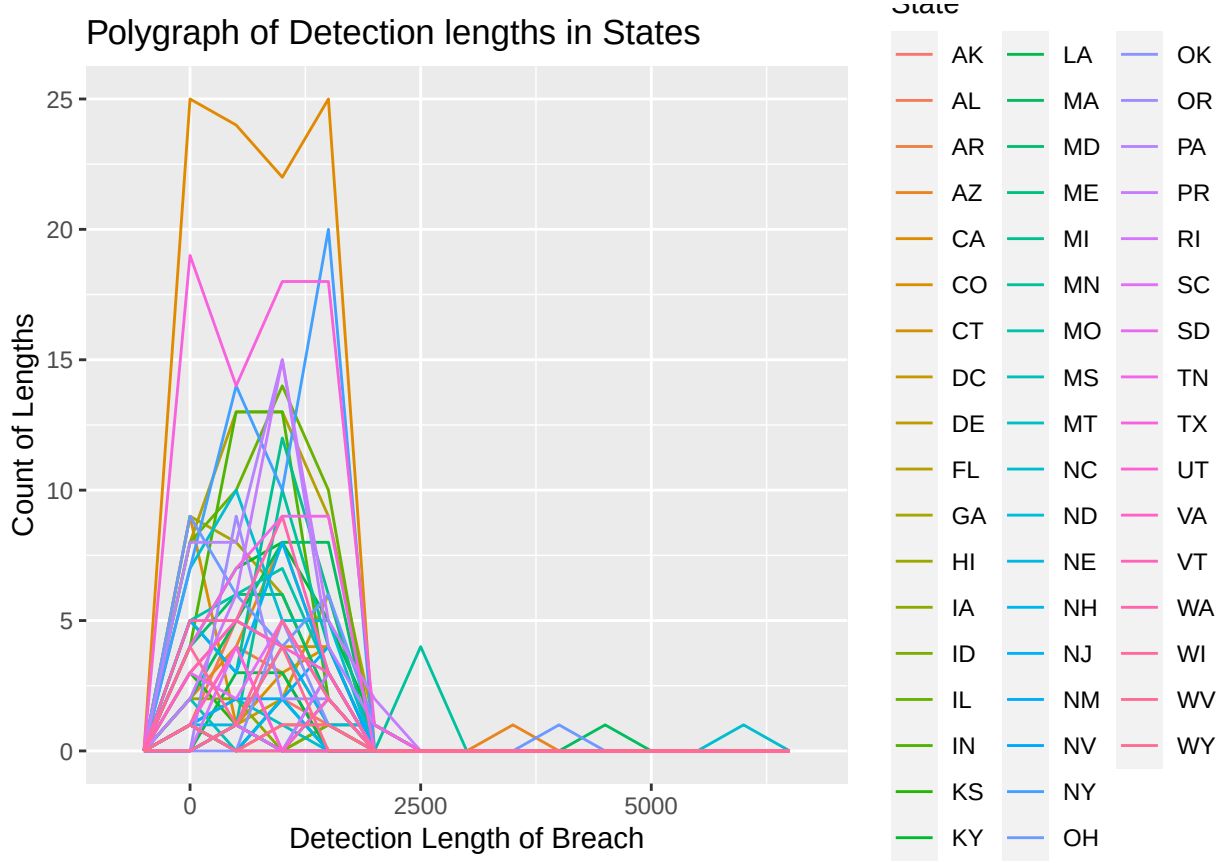
I created three graphs such as the frequency polygraph, jitter graph and a boxplot to understand how the trend can be looked at in different ways. I also wanted to make sure that the trend was consistent and created so many graphs to decide which graph does the best job of showing the distribution.

```
cyberdata_combined <- cbind(cyberdata_edited, cyberdata)
cyberdata_combined <- subset(cyberdata_combined, select=-c(8,9,10,12,14:21))
```

```
ggplot(data = cyberdata_combined, mapping = aes(x = detection_length)) +
geom_freqpoly(mapping = aes(colour = State), binwidth = 500) +
labs(x="Detection Length of Breach", y="Count of Lengths", title="Polygraph of Detection lengths in Sta
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 146 rows containing non-finite values (stat_bin).
```



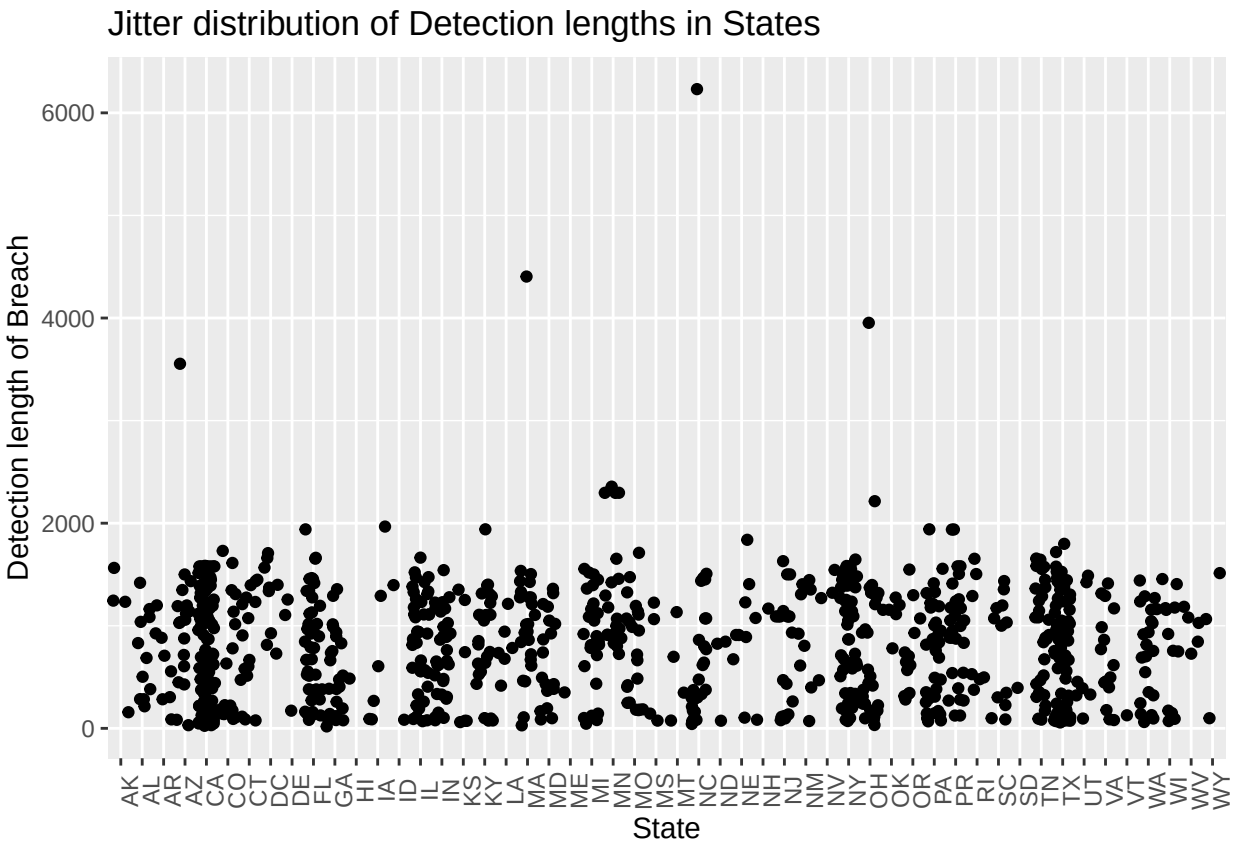

```
cyberdata_combined %>%
group_by(State) %>%
summarize(max_length = max(detection_length, na.rm = TRUE))
```

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

## # A tibble: 52 x 2
##    State max_length
##    <chr> <drtn>
##  1 AK    1564 days
##  2 AL    1419 days
##  3 AR    1199 days
##  4 AZ    3554 days
##  5 CA    1584 days
##  6 CO    1730 days
##  7 CT    1449 days
##  8 DC    1708 days
##  9 DE    1256 days
## 10 FL    1940 days
## # ... with 42 more rows
```

```
ggplot(data = cyberdata_combined, aes(x=State, y=detection_length)) + geom_jitter() +
theme(axis.text.x = element_text(angle = 90)) +
labs(x="State", y="Detection length of Breach", title="Jitter distribution of Detection lengths in State
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 146 rows containing missing values (geom_point).
```

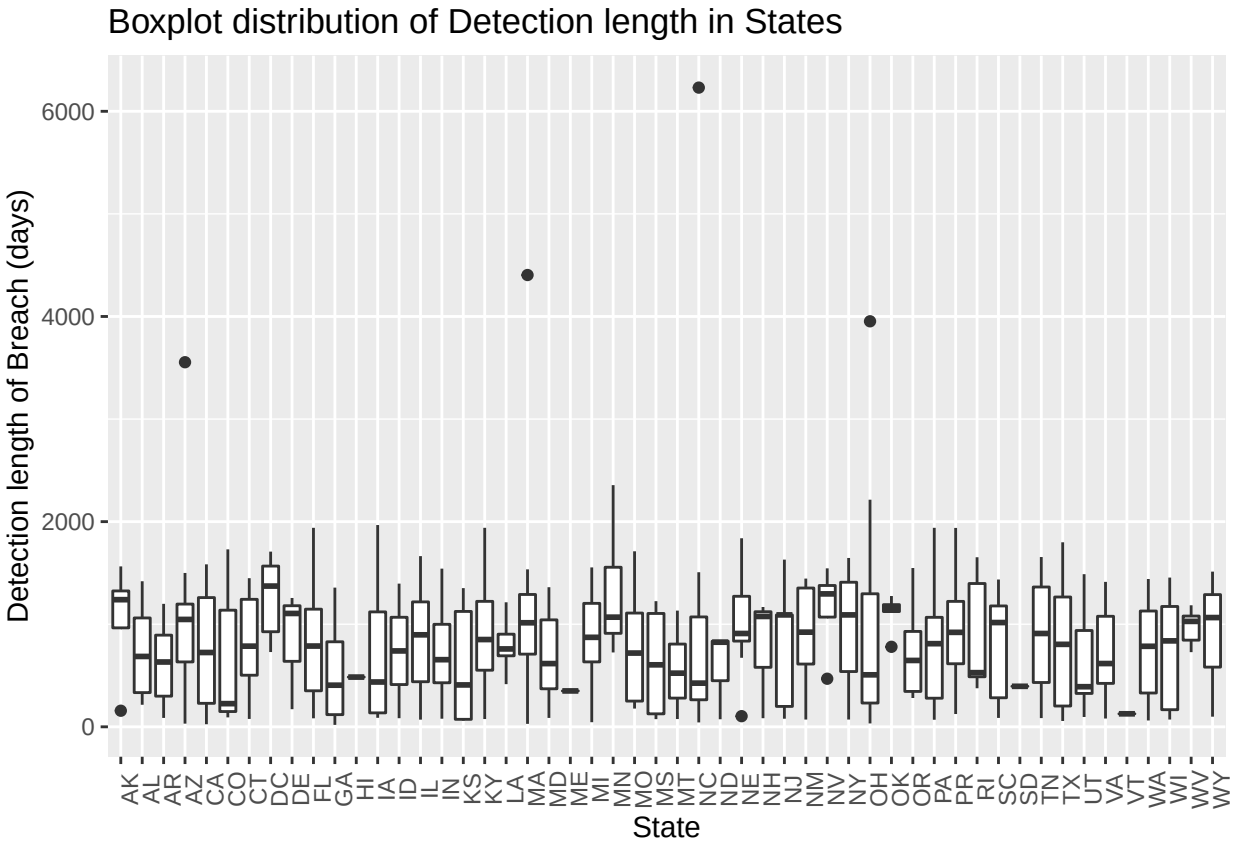


Important

```
ggplot(data = cyberdata_combined, aes(x=State, y=detection_length)) + geom_boxplot() +
theme(axis.text.x = element_text(angle = 90)) +
labs(x="State", y="Detection length of Breach (days)", title="Boxplot distribution of Detection length
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

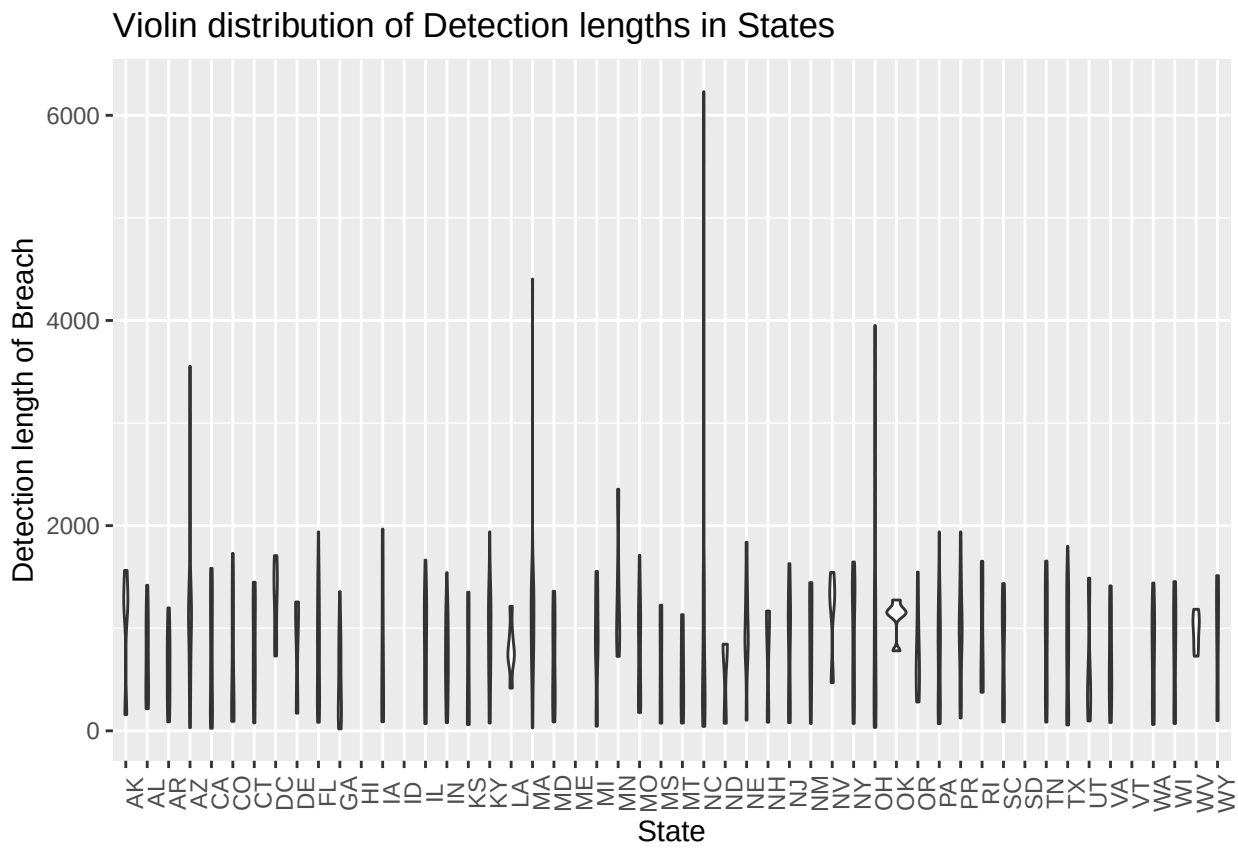
```
## Warning: Removed 146 rows containing non-finite values (stat_boxplot).
```




```
ggplot(data = cyberdata_combined, aes(x=State, y=detection_length)) + geom_violin() +
theme(axis.text.x = element_text(angle = 90)) +
labs(x="State", y="Detection length of Breach", title="Violin distribution of Detection lengths in Stat
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 146 rows containing non-finite values (stat_ydensity).
```



Examining the frequency polygraph, we can see that a majority of the states have shorter detection lengths and fewer states have larger detection lengths due to the lesser number of polygraphs over the 2500 mark. Arizona, Massachusetts, North Carolina, and Ohio are found to be the states that are above the 2500 mark, making these the states that take the longest to detect the breach after it has been found. This is proved by the jitter graph, boxplot and violin graphs since these 4 states have outliers and the largest distribution of values.

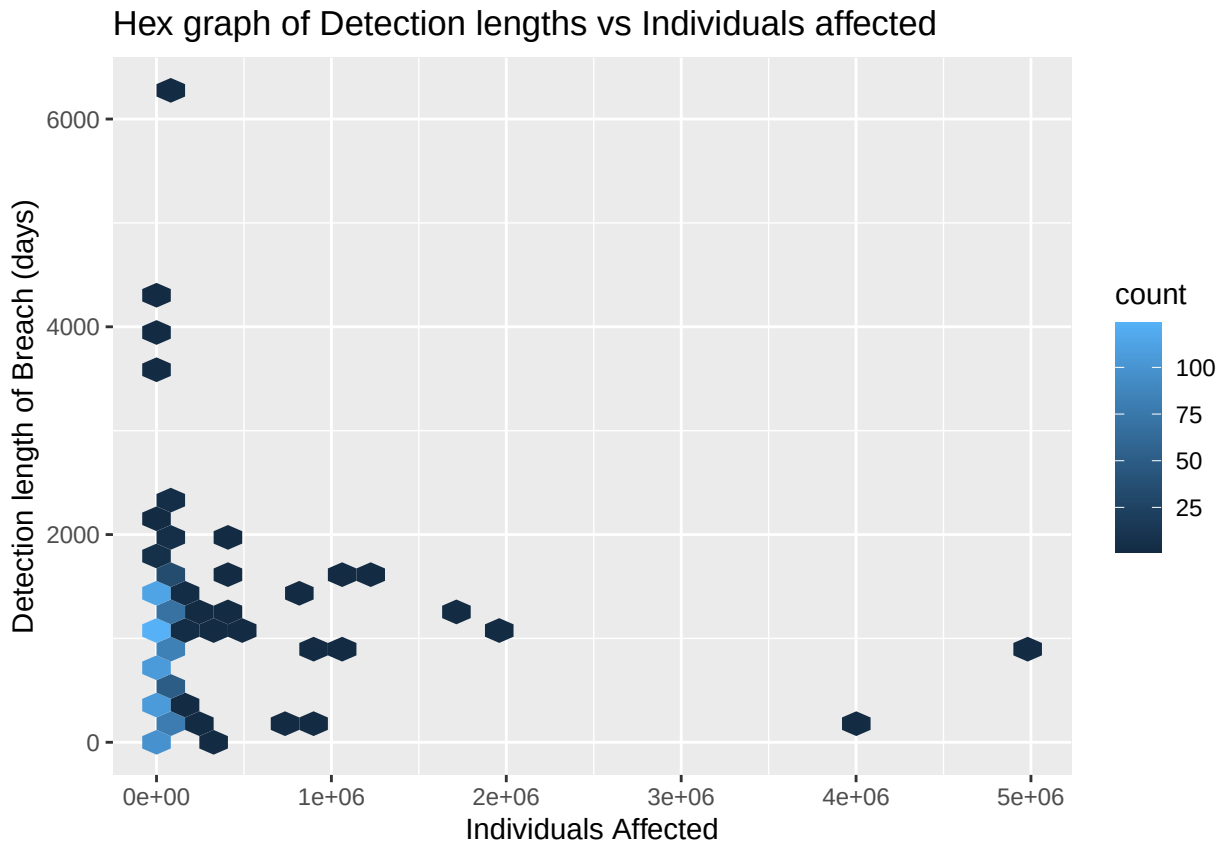
Comparing detection length with individuals affected

Important

```
ggplot(data = cyberdata_combined) +
geom_hex(mapping = aes(x = Individuals_Affected, y = detection_length)) + labs(x="Individuals Affected"
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 146 rows containing non-finite values (stat_binhex).
```



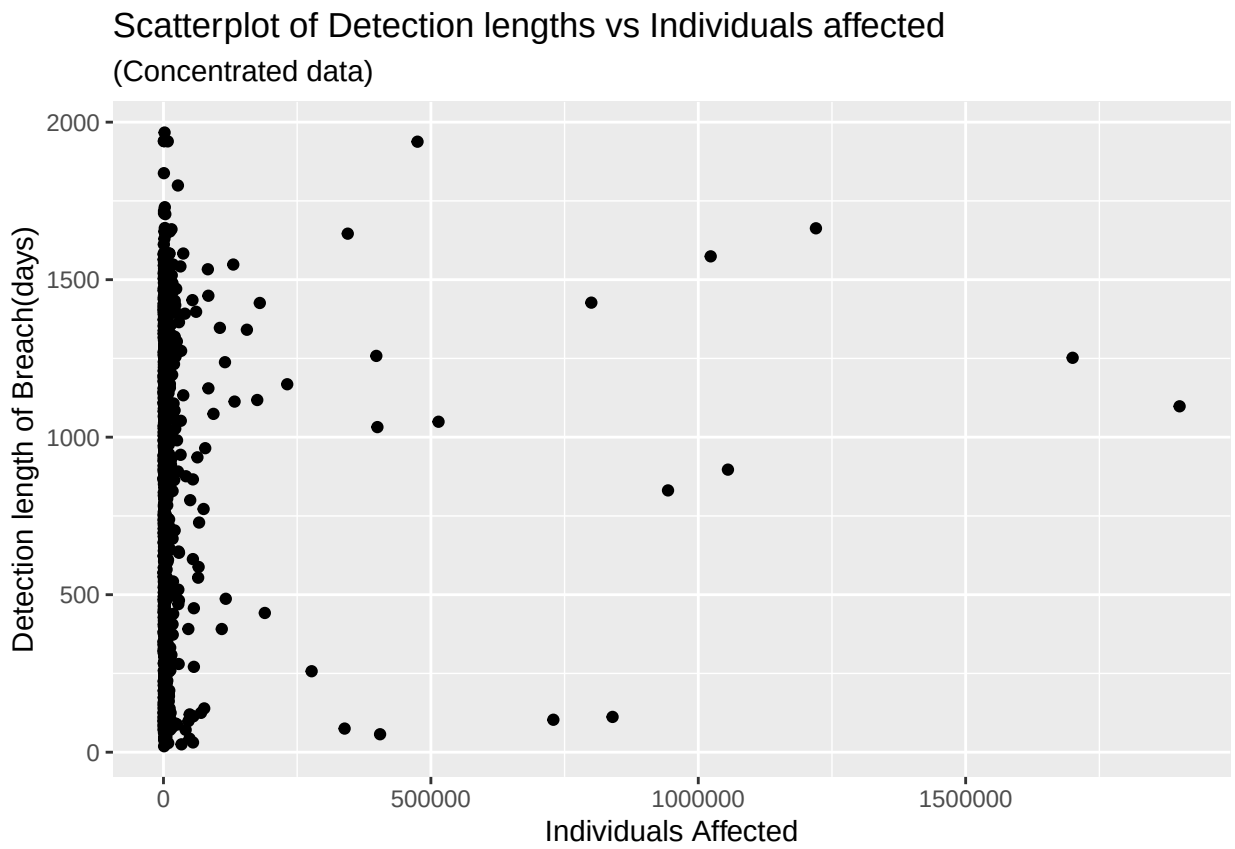
When comparing the individuals affected with the detection time for the breach, very few individuals are affected by increasing detection lengths. Even for detection lengths over 6000, there are extremely few individuals who are affected by the breach. The most number of individuals are affected by a detectin length that is less than the 1000 mark, but there a good amount of individuals in the 1000-2000 range of detection lengths as well.

In order to better understand the spread of data, we will look at a small set of data values in the detection_length variable

```
cyberdata_concentrated <- data.frame(cyberdata_combined)
cyberdata_concentrated <- cyberdata_concentrated %>% filter(detection_length<2000 & Individuals_Affecte
```

```
ggplot(data = cyberdata_concentrated) +
geom_point(mapping = aes(x = Individuals_Affected, y = detection_length)) + labs(x="Individuals Affecte
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



After concentrating our data to only look at number of individuals that are affected to be less than 2,000,000 and the detection lengths to be less than 2000, we can see that the data is a little bit more spread out with no clear trend. However, we do see that when the individuals that are affected are larger, the detection lengths usually lie in the 750-1750 range.

QUESTION 2: HOW LONG DOES A BREACH NORMALLY LAST?

Describe why these variables are relevant and why others are not relevant?

The breach_end and breach_start variables are relevant because they represent the beginning and end of the breach respectively. Additionally, they are also Date variables making comparison and evaluation easily. The other variables don't show what the questions is asking for.

Variable 3: breach_start

1. Which values are the most common? Why? Which values are rare? Why? Does that match your expectations?

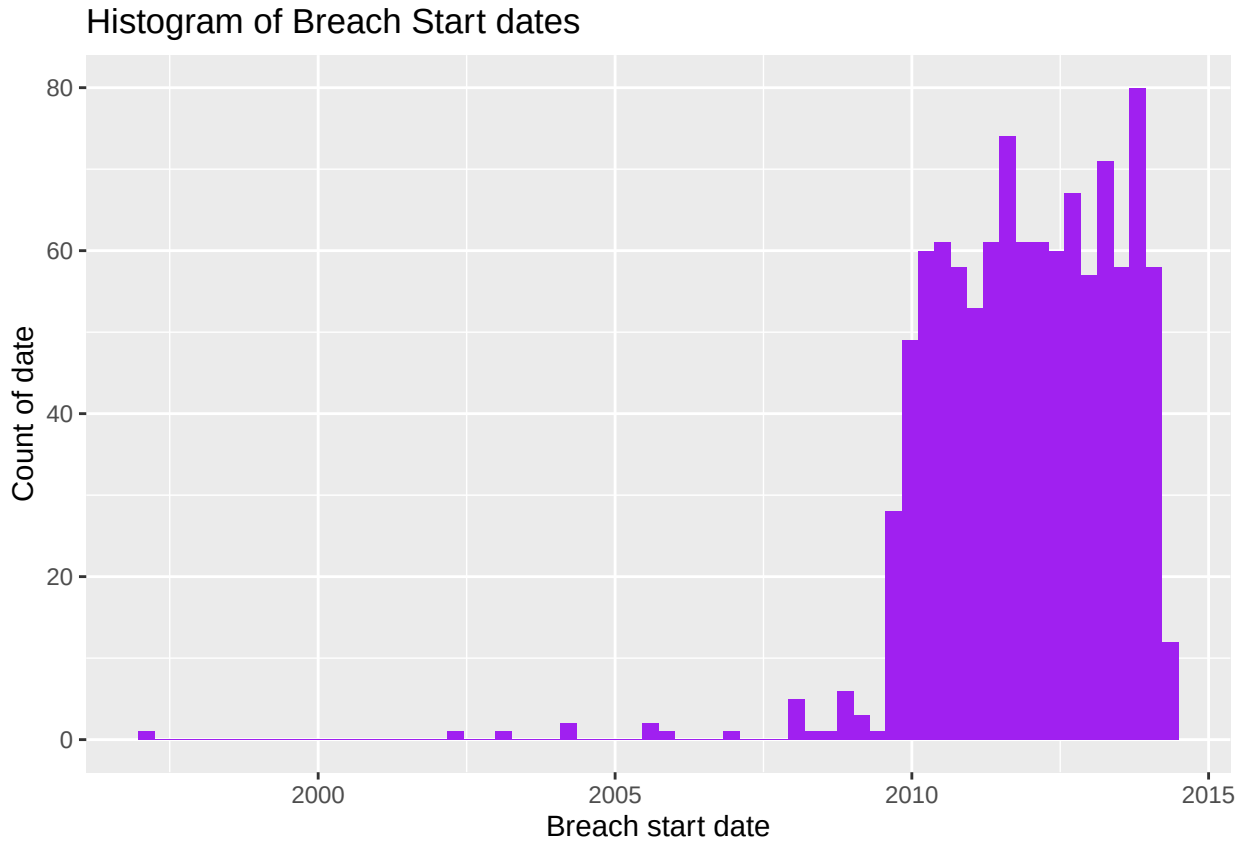
Since we determined this variable to be representative of the Date_of_Breach for single day breaches, a lot of our analysis will be similar to that of Date_of_Breach. However, this variable will have more values than that in the Date_of_Breach column because it is not affected by the multiple day breaches, since it still represents the initial day of the breach in the period.

From the histogram, we can tell that around 2012 is when the count is the highest for the breach_start variable. Hence, this data is the most common. From filtering the data set to examine the variable, we can confirm that the value that has the highest count of 8 occurences is 2012-06-15 or the 15th of June, 2012.

As for which values are rare, a majority of the dates in the initial years such as in the range from 1997 to 2008 seem to have rare occurences. We determine there to be 522 values with rare occurences or counts of 1.

Important

```
ggplot(cyberdata_edited, aes(breach_start)) +
geom_histogram(binwidth = 100, fill="purple") +
labs(x="Breach start date", y="Count of date", title="Histogram of Breach Start dates")
```




```
cyberdata_edited %>% group_by(breach_start) %>% count() %>% filter(n>5)
```

```
## # A tibble: 6 x 2
## # Groups:   breach_start [6]
##   breach_start     n
##   <date>       <int>
## 1 2009-09-27       6
## 2 2011-03-10       6
## 3 2011-06-24       7
## 4 2012-01-11       7
## 5 2012-06-15       8
## 6 2013-09-20       6
```

```
cyberdata_edited %>% group_by(breach_start) %>% count() %>% filter(n==1)
```

```
## # A tibble: 522 x 2
## # Groups:   breach_start [522]
##    breach_start     n
##    <date>       <int>
##  1 1997-01-01       1
##  2 2002-05-06       1
##  3 2003-03-29       1
##  4 2004-04-21       1
##  5 2004-05-01       1
##  6 2005-08-15       1
##  7 2005-09-01       1
##  8 2006-01-01       1
##  9 2007-01-01       1
## 10 2008-01-01       1
## # ... with 512 more rows
```

2. Can you see any unusual patterns? What might explain them?

After plotting the month, year and day of the breach_start variable, we can get a better idea of any unusual patterns. We can determine that a majority of the breaches started in the month of September, followed by March. The year that faced the most breach starts was the year 2013, followed by 2012/2011. As for days, we see that the beginning days had the highest count.

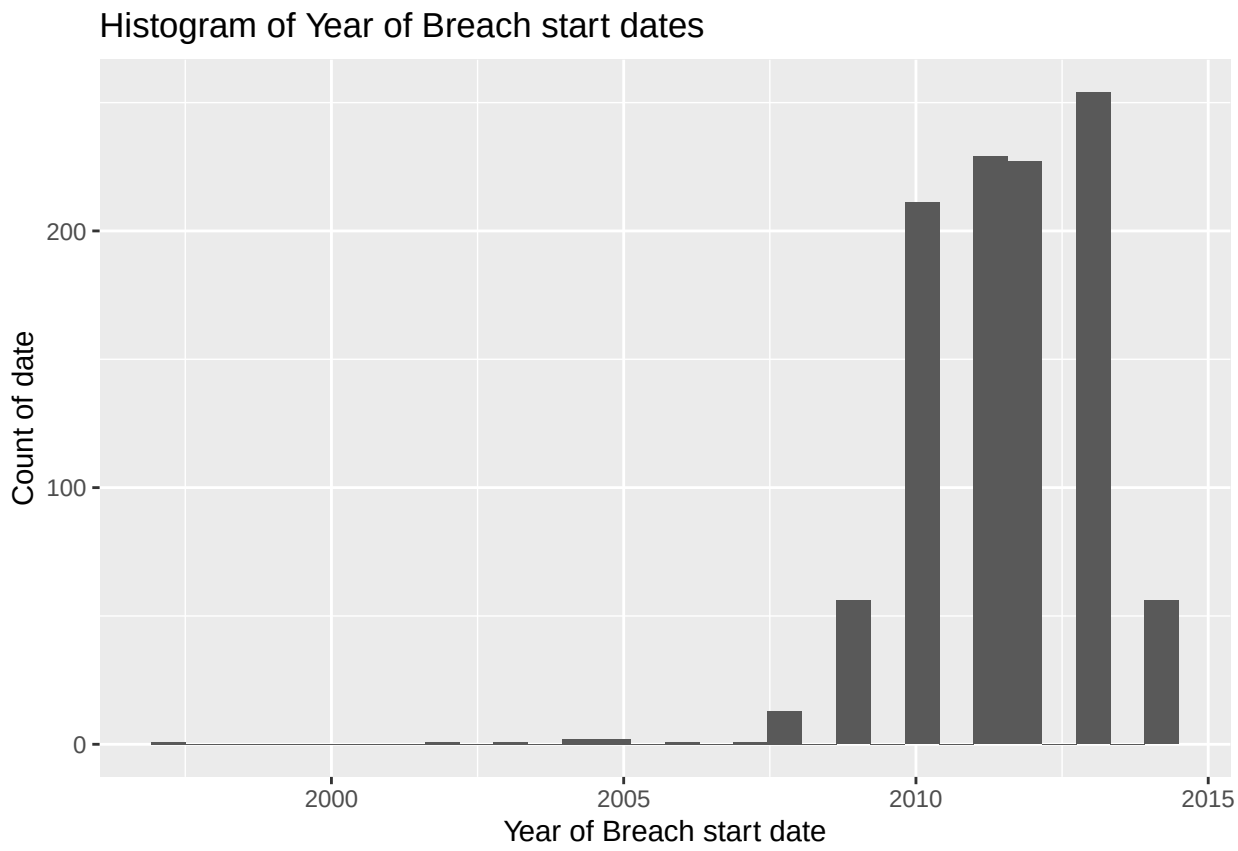
Hence, while there are no explicit unusual values, what we notice is that every month has almost a good count of breach starts but the latter half of the years seem to be the ones with higher breaches. As for days, the beginning and middle have high counts but in between those, the breach start counts are fairly low.

When thinking about what might explain these trends, we can attribute it to the fact that the restrictions or the data protection methods were weak in the years from 2011-2013 and the violators probably plotted on how to conduct the breach towards the end of the month so they can breach at the start of each month, and even planned in the first half so they can steal data in the middle of the month.

Important

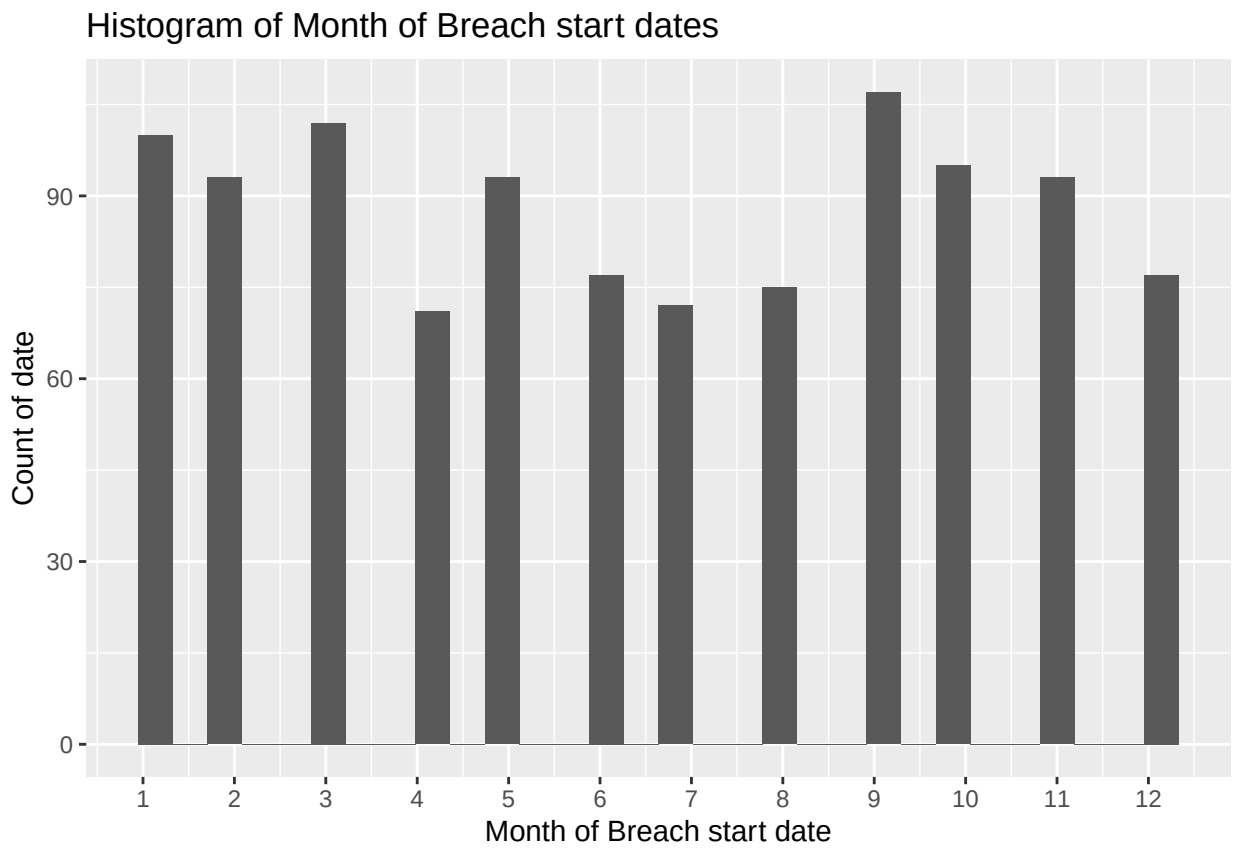
```
cyberdata_edited %>% ggplot(aes(year(breach_start))) +
geom_histogram() +
labs(x="Year of Breach start date", y="Count of date", title="Histogram of Year of Breach start dates")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



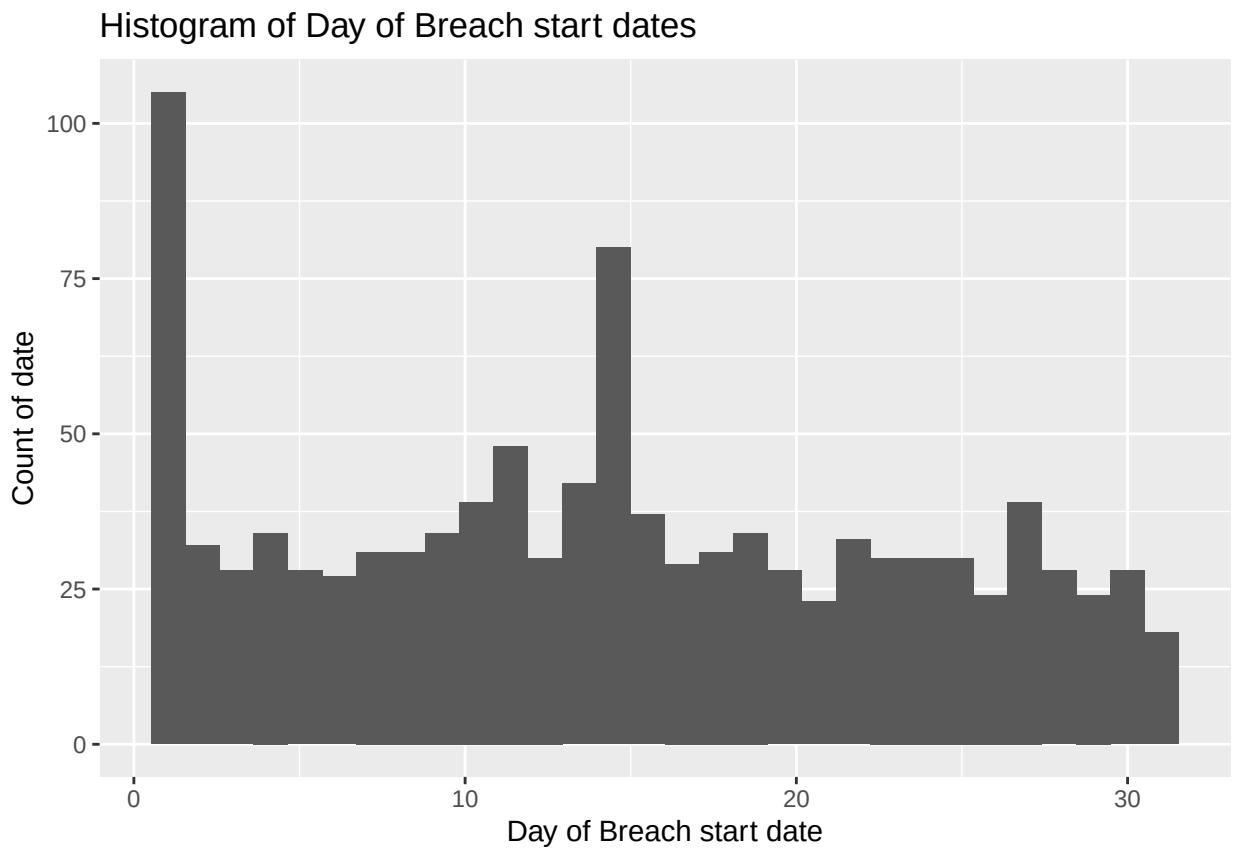

```
cyberdata_edited %>% ggplot(aes(month(breach_start))) +
geom_histogram() +
scale_x_continuous(breaks=c(1:12)) +
labs(x="Month of Breach start date", y="Count of date", title="Histogram of Month of Breach start dates
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
cyberdata_edited %>% ggplot(aes(day(breach_start))) +
geom_histogram() +
labs(x="Day of Breach start date", y="Count of date", title="Histogram of Day of Breach start dates")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



3. Are there clusters in the data? How are the observations within each cluster similar to or different from each other? How can you explain or describe the clusters?

For the month of the breach_start variable, there are no clusters since their counts seem to increase and decrease randomly. For the year of the breach_start, the cluster is formed by the years from 2010 to 2013 due to their high counts. For the day of the breach_start variable, the days in the first half(between the initial and the mid) and the days in the second half(after the mid) can also be clustered due to their low counts in the same range.

Unusual values

1. Describe and demonstrate how you determine if there are unusual values in the data. E.g. too large, too small, negative, etc.

The maximum date value is found to be 2014-06-02 or the 2nd of June in 2014. The minimum date value is found to be 1997-01-01 or the 1st of January in 1997. There are no negative values found in the breach_start column which is expected because it is a date.

```
max_breach_start <- summarize(cyberdata_edited, max(breach_start, na.rm = TRUE))
max_breach_start
```

```
##   max(breach_start, na.rm = TRUE)
## 1                      2014-06-02
```

```
min_breach_start <- summarize(cyberdata_edited, min(breach_start, na.rm = TRUE))
min_breach_start
```

```
##   min(breach_start, na.rm = TRUE)
## 1                      1997-01-01
```

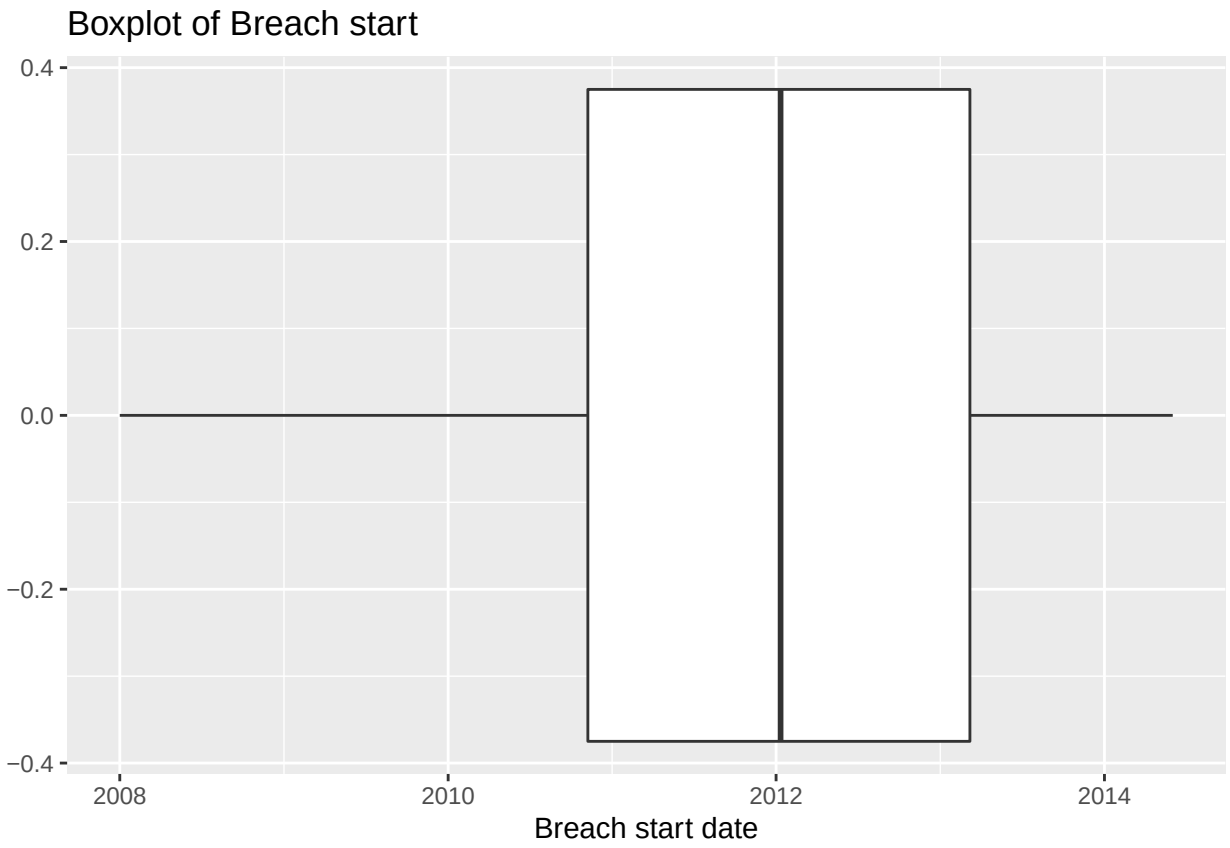
```
sum(cyberdata_edited$breach_start<0)
```

```
## [1] 0
```

2. Describe and demonstrate how you determine if they are outliers.

In creating the boxplot, we can see that there are no data points that are far away from the whisker. So there are no outliers in this variable. The median is found to be at 2012.

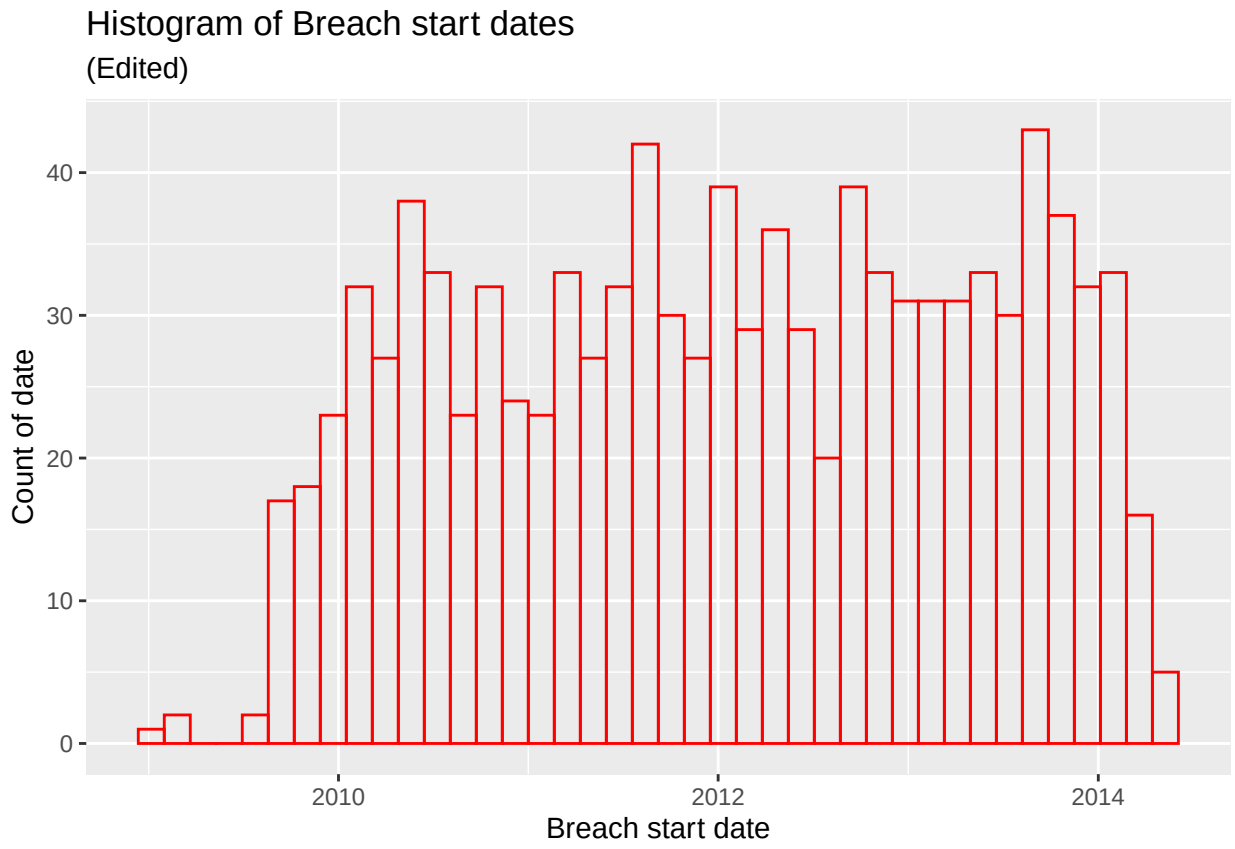
```
ggplot(data = cyberdata_edited, aes(x = breach_start)) +
geom_boxplot() +
labs(x="Breach start date", title="Boxplot of Breach start")
```



3. Show how do your distributions look like with and without the unusual values.

Since there are no unusual values, our distribution will look the same.But we can filter out the beginning values in the years below 2008 to look at the data in a more zoomed in sense.

```
cyberdata_BS_outliers <- cyberdata_edited %>% filter(year(breach_start) > 2008)

ggplot(data = cyberdata_BS_outliers, mapping = aes(breach_start)) + geom_histogram(binwidth = 50, fill=
labs(x="Breach start date", y="Count of date", title="Histogram of Breach start dates", subtitle="(Edit
```



4. Discuss whether or not you need to remove unusual values and why.

There are no unusual values or outliers in the data set and so no need to remove anything.

Missing Values

1. Does this variable include missing values? Demonstrate how you determine that.

This variable has no missing or NA values.

```
sum(is.na(cyberdata_edited$breach_start))
```

```
## [1] 0
```

Variable 4: breach_end

1. Which values are the most common? Why? Which values are rare? Why? Does that match your expectations?

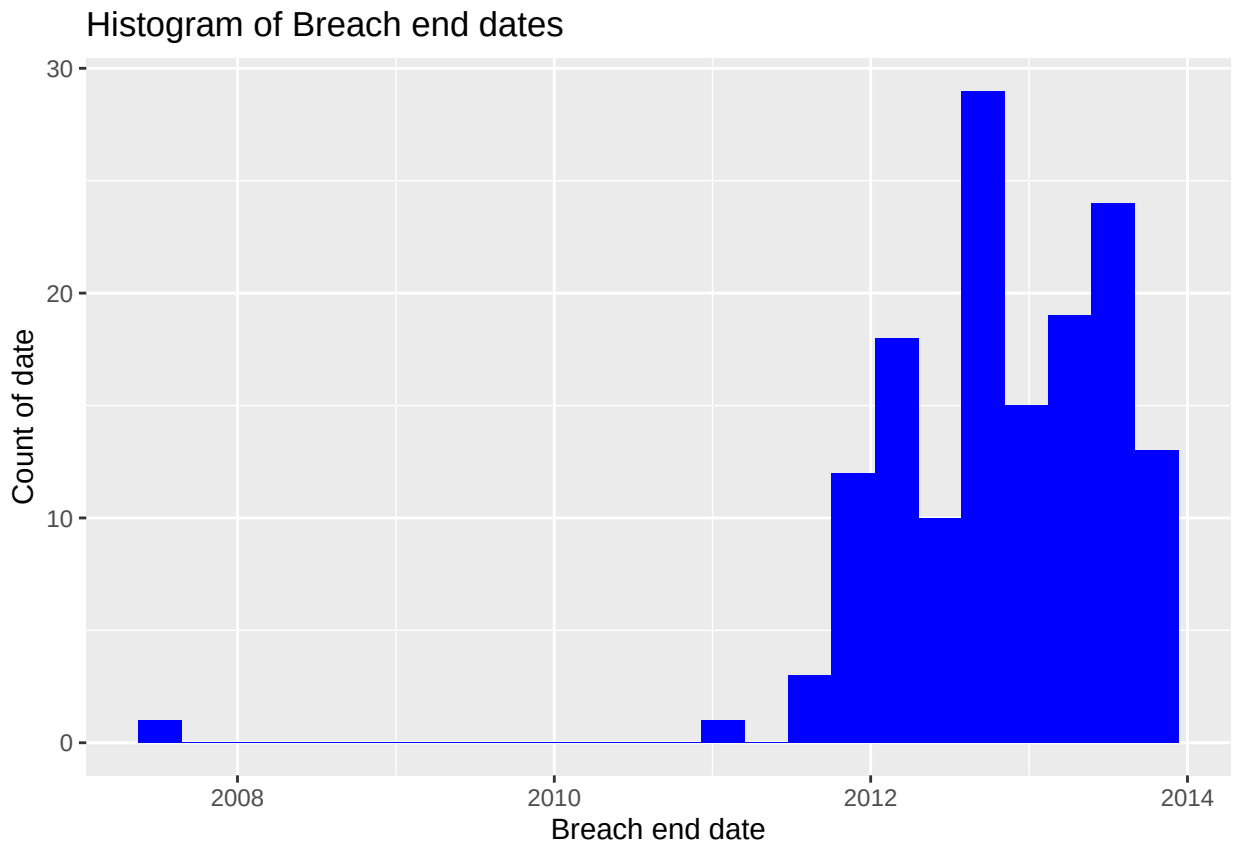
This variable has values when the Date_of_Breach was not a single day event but in fact took a period of time to execute.

The histogram shows that the bulk of breach end dates occur from 2011-2014. The most common value is the date 2012-10-01 or the 1st of October 2012 because it has 11 counts. There are many rare values, 108 to be precise, due to their count being 1.

Important

```
ggplot(cyberdata_edited, aes(breach_end)) +
geom_histogram(binwidth = 100, fill="blue") +
labs(x="Breach end date", y="Count of date", title="Histogram of Breach end dates")
```

```
## Warning: Removed 910 rows containing non-finite values (stat_bin).
```




```
cyberdata_edited %>% group_by(breach_end) %>% count() %>% filter(n==1)
```

```
## # A tibble: 108 x 2
## # Groups:   breach_end [108]
##    breach_end     n
##    <date>     <int>
##  1 2007-06-14     1
##  2 2011-02-28     1
```

```
##  3 2011-08-05     1
##  4 2011-08-18     1
##  5 2011-09-20     1
##  6 2011-10-17     1
##  7 2011-10-28     1
##  8 2011-11-16     1
##  9 2011-11-21     1
## 10 2011-11-30     1
## # ... with 98 more rows
```

```
cyberdata_edited %>% group_by(breach_end) %>% count() %>% filter(n>1)
```

```
## # A tibble: 14 x 2
## # Groups:   breach_end [14]
##    breach_end     n
##    <date>     <int>
##  1 2012-04-02     2
##  2 2012-09-21     2
##  3 2012-10-01    11
##  4 2012-10-27     3
##  5 2012-11-15     3
##  6 2012-12-20     2
##  7 2013-06-07     2
##  8 2013-06-24     2
##  9 2013-07-11     2
## 10 2013-07-16     2
## 11 2013-08-02     2
## 12 2013-08-15     2
## 13 2013-10-04     2
## 14 NA           910
```

2. Can you see any unusual patterns? What might explain them?

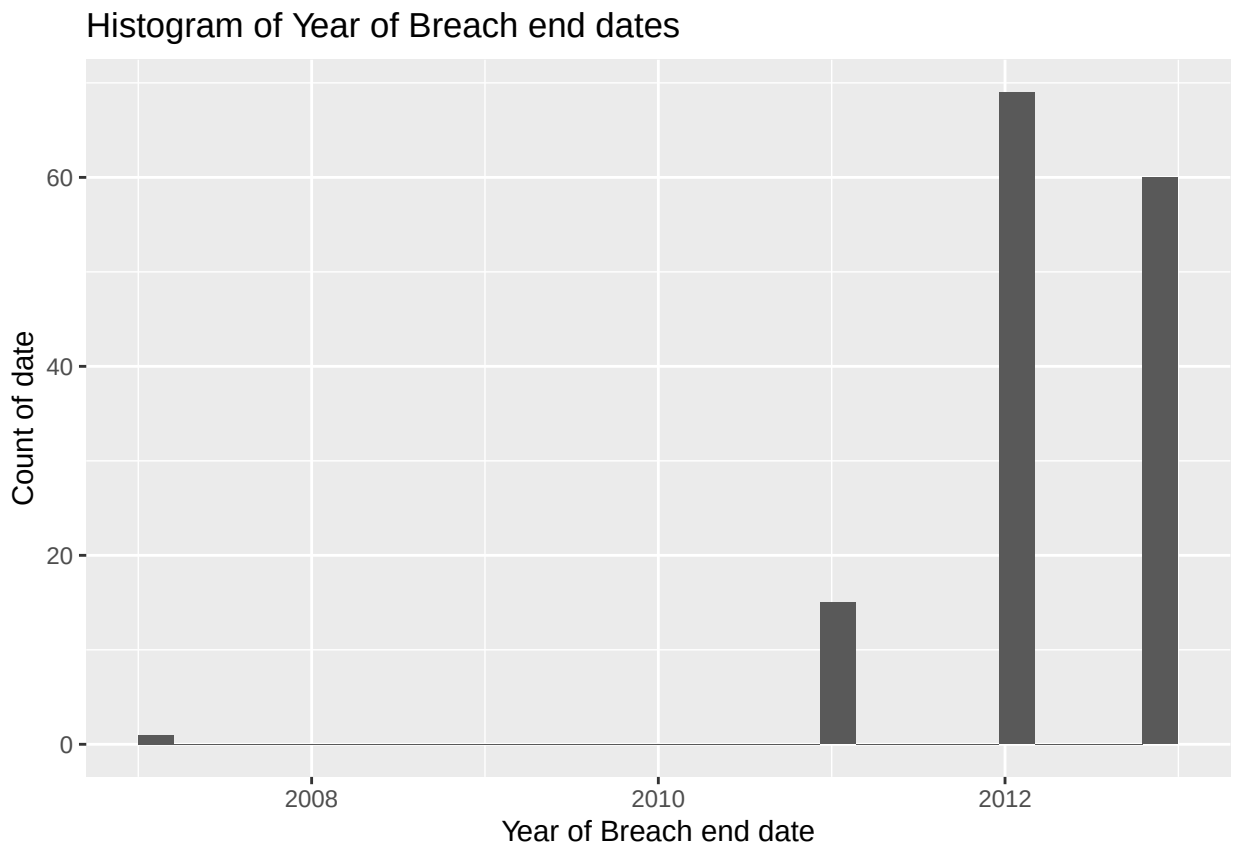
The unusual patterns we see is that in the histogram distribution, only the years 2007, 2011, 2012 and 2013 exist. This means that for the other years, the breach only took a single day to occur. But in these aforementioned years, breaches took place over a couple days. The month of October, followed by March seem to have the most counts, meaning these months had the most period breaches. As for the days, we notice the same trend as we saw in breach_start, with the very beginning and the mid of the month to have the highest count. Something that explains this is that performing the breach takes a while so they usually began and ended around the same time they began but after a couple days.

Important

```
cyberdata_edited %>% ggplot(aes(year(breach_end))) +
geom_histogram() +
labs(x="Year of Breach end date", y="Count of date", title="Histogram of Year of Breach end dates")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

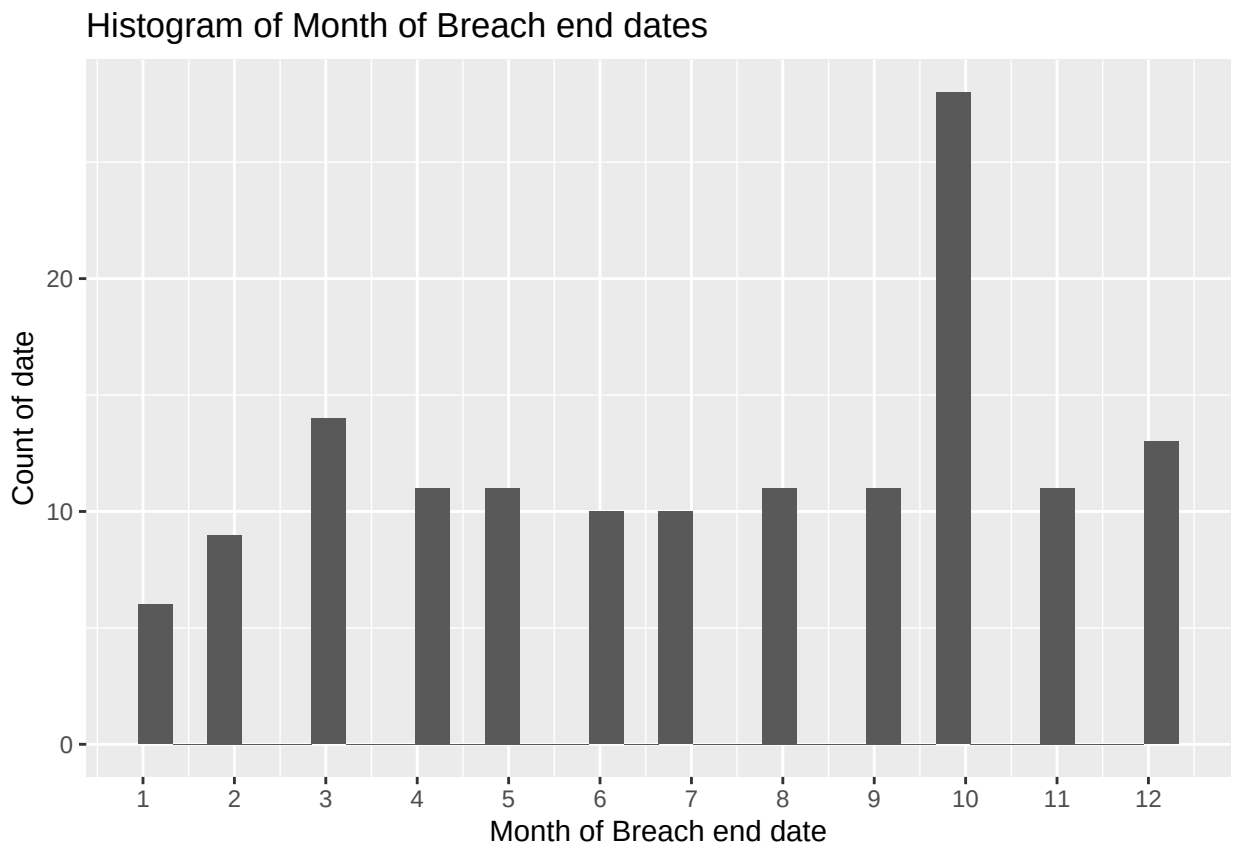
```
## Warning: Removed 910 rows containing non-finite values (stat_bin).
```




```
cyberdata_edited %>% ggplot(aes(month(breach_end))) +
geom_histogram() +
scale_x_continuous(breaks=c(1:12)) +
labs(x="Month of Breach end date", y="Count of date", title="Histogram of Month of Breach end dates")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

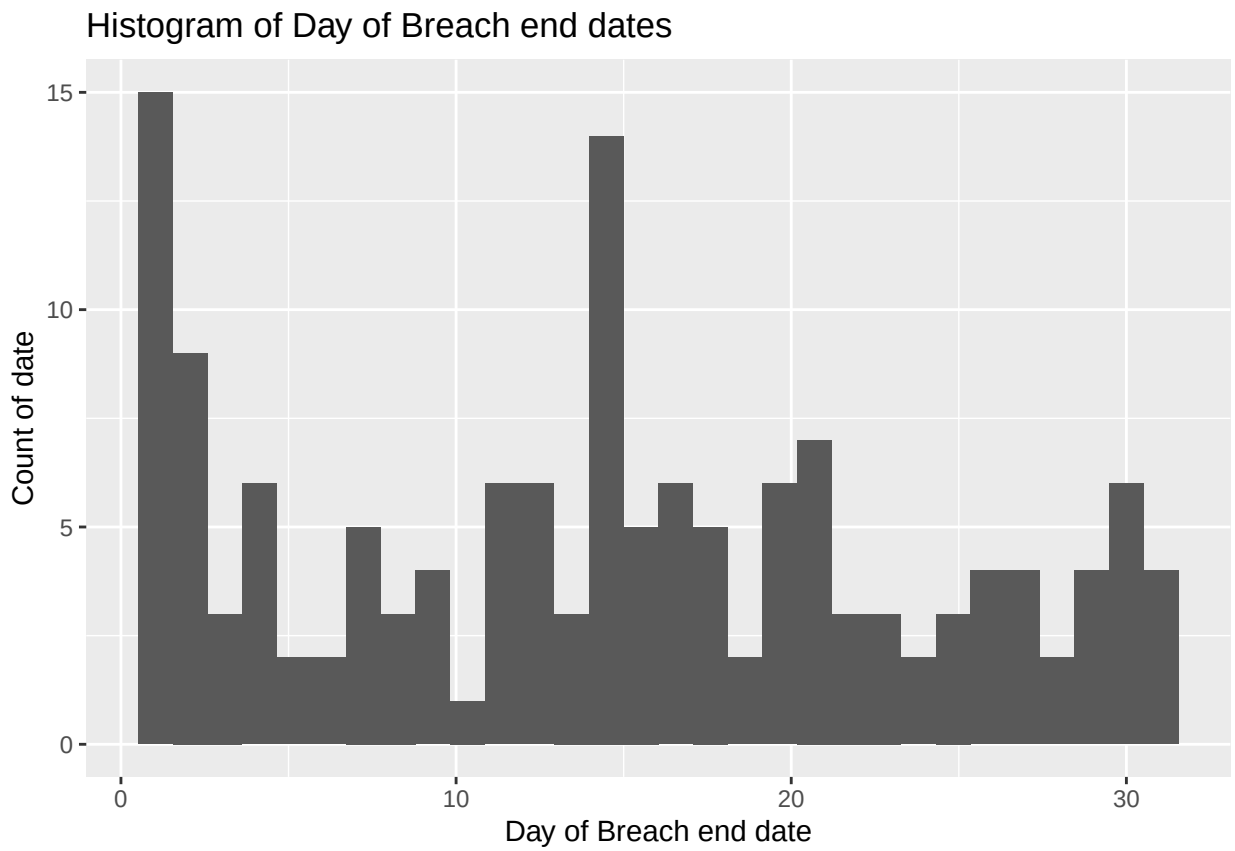
```
## Warning: Removed 910 rows containing non-finite values (stat_bin).
```




```
cyberdata_edited %>% ggplot(aes(day(breach_end))) +
geom_histogram() +
labs(x="Day of Breach end date", y="Count of date", title="Histogram of Day of Breach end dates")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 910 rows containing non-finite values (stat_bin).
```



3. Are there clusters in the data? How are the observations within each cluster similar to or different from each other? How can you explain or describe the clusters?

There are no clusters in the year of the breach_start. The months from April through September form a cluster since their count is all under 12. The days from the same clusters as we determined in the breach_start variable - between 1-12 and 15-31.

Unusual values

1. Describe and demonstrate how you determine if there are unusual values in the data. E.g. too large, too small, negative, etc.

The largest value or the latest date is 2013-11-30 or the 30th of November 2013. The smallest value or the earliest date is 2007-06-14 or the 14th of June 2014. There are no negative values which is expected because this is a date.

```
max_breach_end <- summarize(cyberdata_edited, max(breach_end, na.rm = TRUE))
max_breach_end
```

```
##   max(breach_end, na.rm = TRUE)
## 1                    2013-11-30
```

```r
min_breach_end <- summarize(cyberdata_edited, min(breach_end, na.rm = TRUE))
min_breach_end
```

```
##   min(breach_end, na.rm = TRUE)
## 1                    2007-06-14
```

```r
sum(cyberdata_edited$breach_end<0)
```

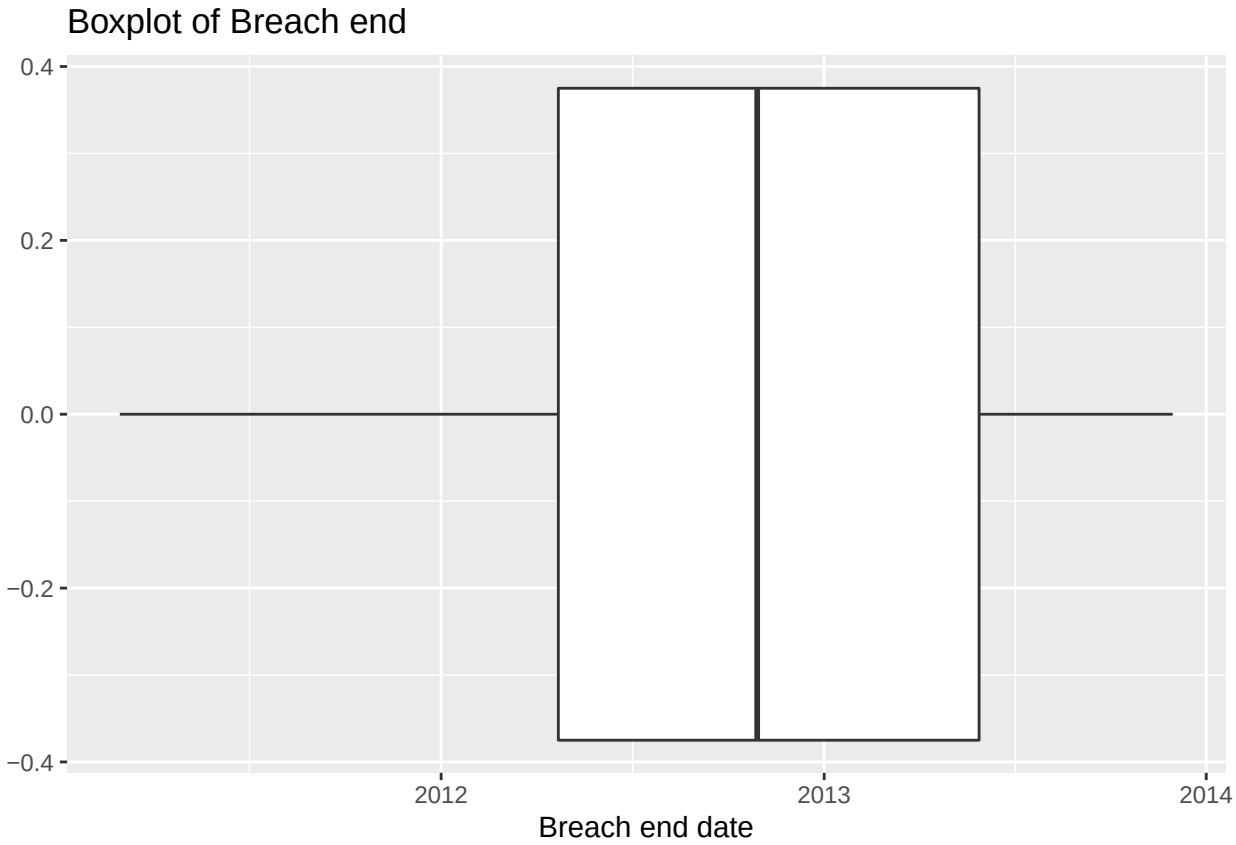
```
## [1] NA
```

2. Describe and demonstrate how you determine if they are outliers.

In creating the boxplot, we can see that there are no data points that are far away from the whisker. So there are no outliers in this variable. The median is found to be between 2012 and 2013 but closer to the latter.

```r
ggplot(data = cyberdata_edited, aes(x = breach_end)) + geom_boxplot() +
labs(x="Breach end date", title="Boxplot of Breach end")
```

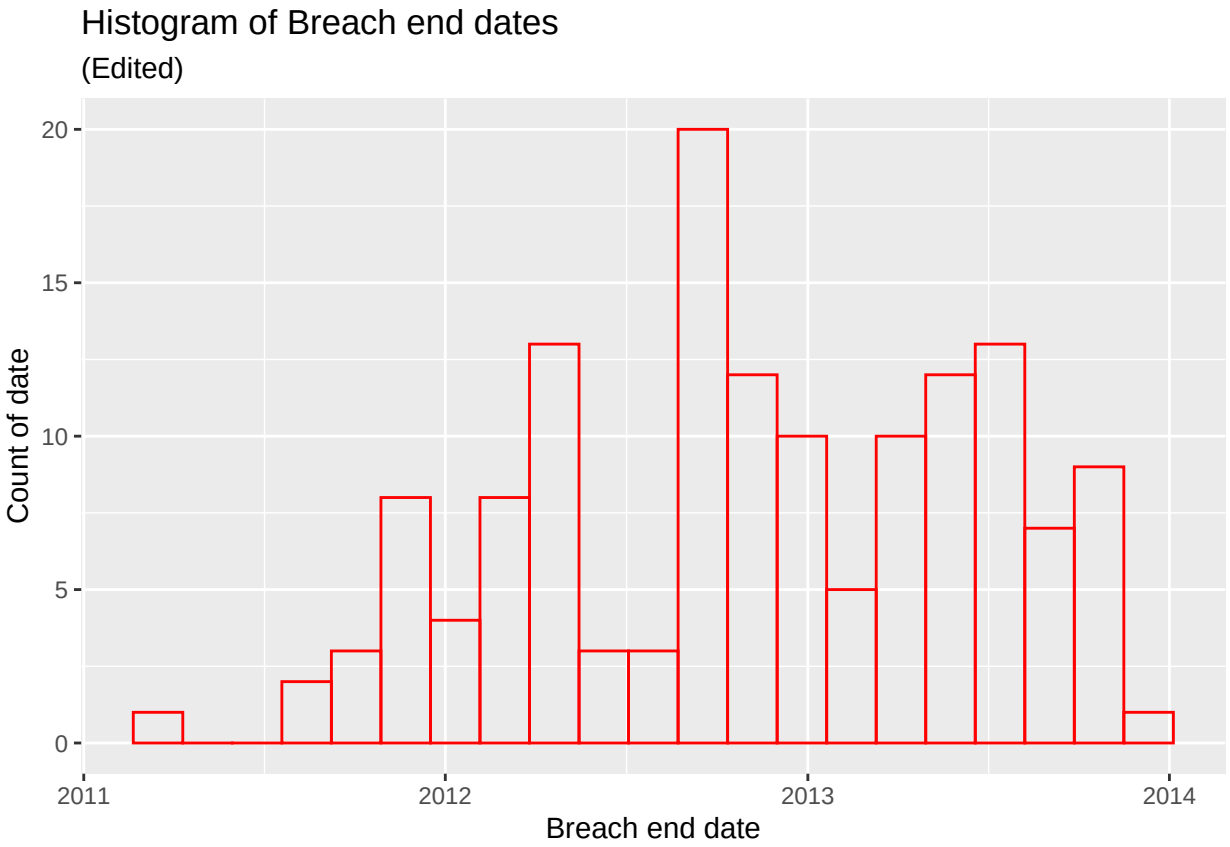
```
## Warning: Removed 910 rows containing non-finite values (stat_boxplot).
```



3. Show how do your distributions look like with and without the unusual values.

Since there are no unusual values, our distribution will look the same.But we can filter out the beginning values in the years below 2010 to look at the data in a more zoomed in sense.

```
cyberdata_BE_outliers <- cyberdata_edited %>% filter(breach_end > '2010-01-01')

ggplot(data = cyberdata_BE_outliers, mapping = aes(breach_end)) + geom_histogram(binwidth = 50, fill=NA
labs(x="Breach end date", y="Count of date", title="Histogram of Breach end dates", subtitle="(Edited)"
```



4. Discuss whether or not you need to remove unusual values and why.

There are no unusual values or outliers in the data set and so no need to remove anything.

Missing Values

1. Does this variable include missing values? Demonstrate how you determine that.

This variable has 910 missing or NA values. We determined this by using the is.na() function.

```
sum(is.na(cyberdata_edited$breach_end))
```

```
## [1] 910
```

2. Demonstrate and discuss how you handle the missing values. E.g., removing, replacing with a constant value, or a value based on the distribution, etc.

The missing values in this column exist to represent the breaches that only took a single day to occur and thus didn't have a breach_end since the breach_start was representative. We will not remove these rows, but we can replace them with the same value in the breach_start variable for future analysis. So when we subtract the breach_start from the breach_end, we will be able to obtain 0 values for the single day breaches and actual values for the multiple day breaches.

In our code below, we have replaced the values and have checked the type of the variable to ensure its a Date.

```
cyberdata_BE_replacedNA <- data.frame(cyberdata_edited)

invalid_dates <- is.na(cyberdata_BE_replacedNA$breach_end)
if(any(invalid_dates)) {
  cyberdata_BE_replacedNA$breach_end[invalid_dates] <- cyberdata_BE_replacedNA$breach_start[invalid_dat
}

str(cyberdata_BE_replacedNA$breach_end)
```

```
##  Date[1:1055], format: "2009-10-16" "2009-09-22" "2009-10-12" "2009-10-09" "2009-09-27" ...
```

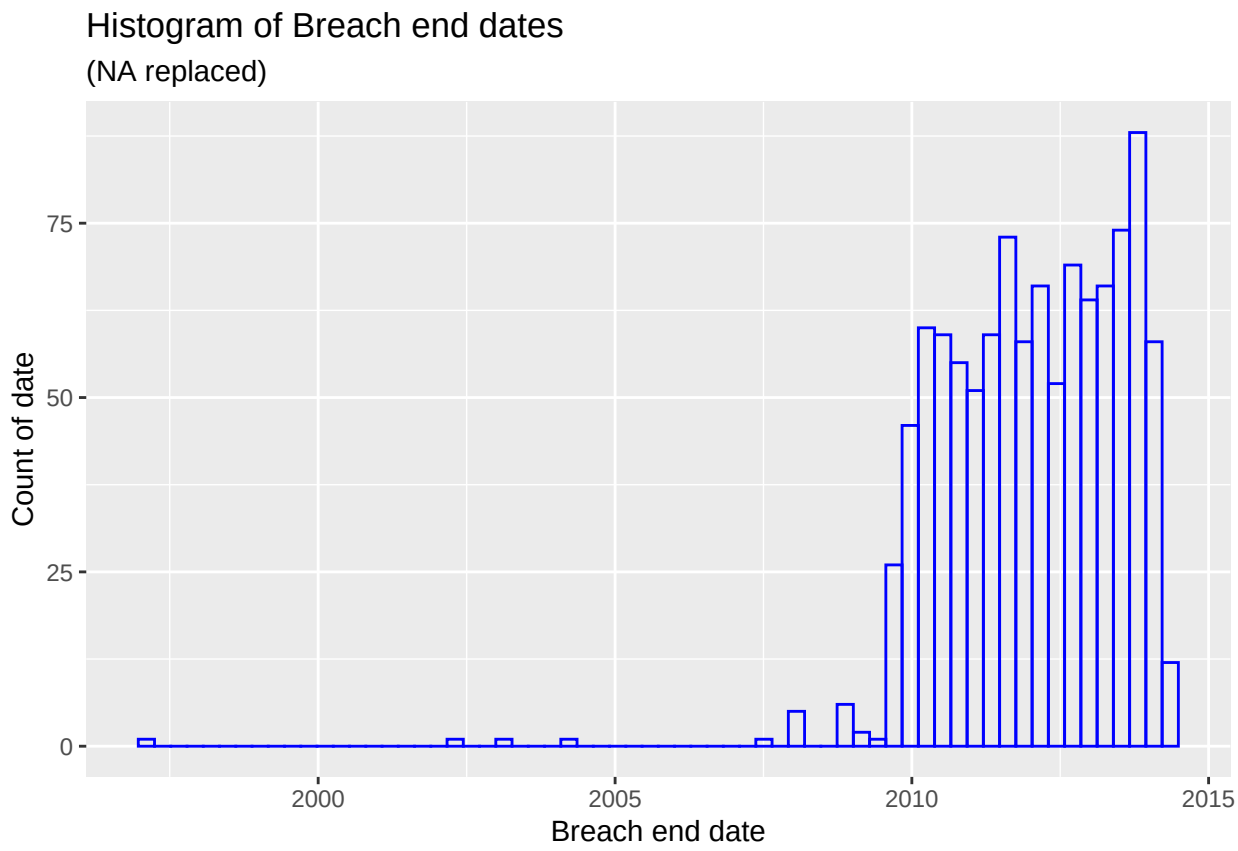
```
str(cyberdata_BE_replacedNA$breach_start)
```

```
##  Date[1:1055], format: "2009-10-16" "2009-09-22" "2009-10-12" "2009-10-09" "2009-09-27" ...
```

3. Show how your data looks in each case after handling missing values.Describe and discuss the distribution.

After handling the missing values by adding in the breach_start values in place of the NA values in the breach_end column, we can see that the distribution has a larger density of values starting from 2010. The distribution now also consists of values prior to 2008, such as including the 1997 year as well.

```
ggplot(cyberdata_BE_replacedNA, aes(breach_end)) +
geom_histogram(binwidth = 100, fill=NA, color="blue") +
labs(x="Breach end date", y="Count of date", title="Histogram of Breach end dates", subtitle="(NA repla
```



Analyzing breach time span for multiple day breaches

Since we determined that multiple day breaches usually have a breach_end value that different from the breach_start value or that is not NA, we will determine how much time it takes for breaches to happen.

The breach period variable consists of the breach start subtracted from the breach end variable. This tells us how long it takes for the breach to take place.

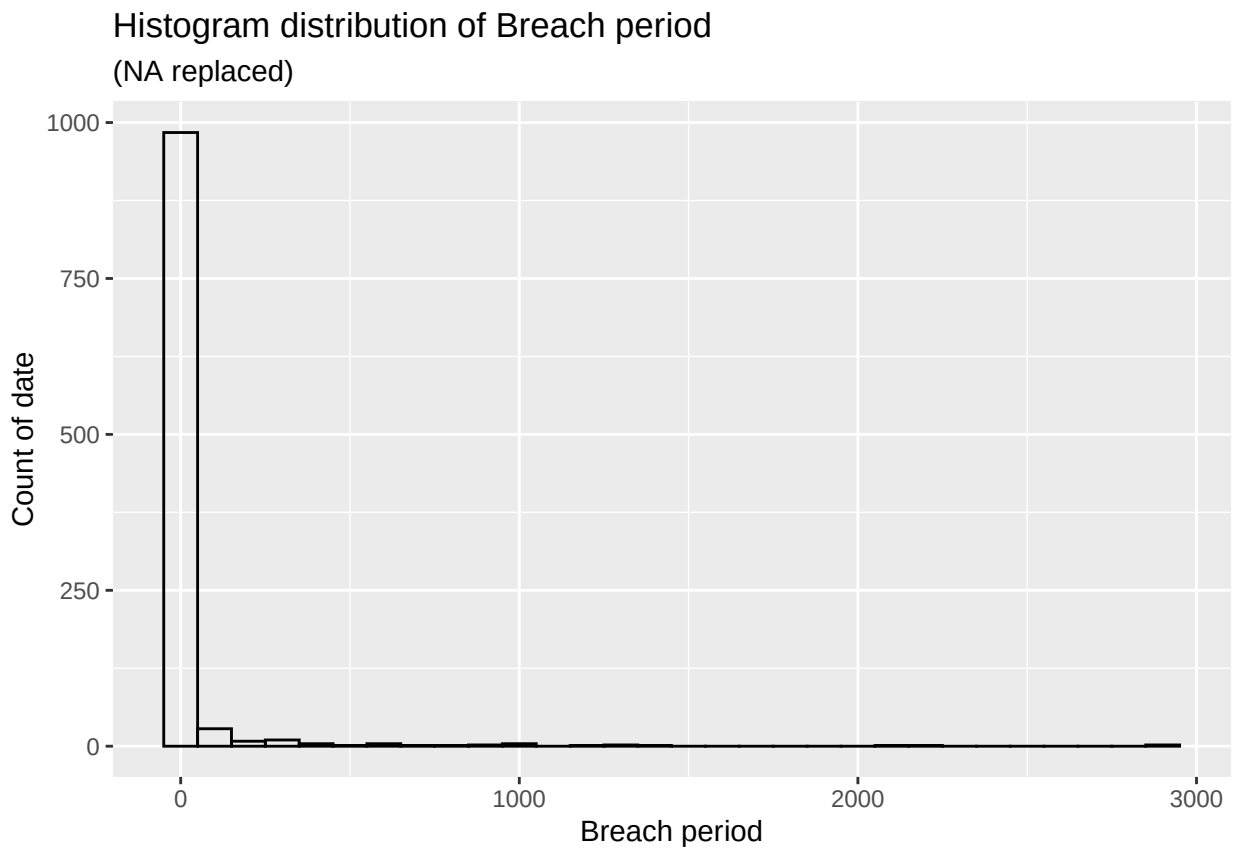
The unit of measure for this variable is days.

```
cyberdata_BE_replacedNA <- cyberdata_BE_replacedNA %>%
mutate(breach_period = breach_end - breach_start)
```

This distribution shows us that a majority of the values in the breach period variable are 0, this means that there are mostly single day breaches. There are very few multiple day breaches and to understand them better, we will take a closer look at the data.

```
ggplot(cyberdata_BE_replacedNA, aes(breach_period)) +
geom_histogram(binwidth = 100, fill=NA, color="black") +
labs(x="Breach period", y="Count of date", title="Histogram distribution of Breach period", subtitle="(
```

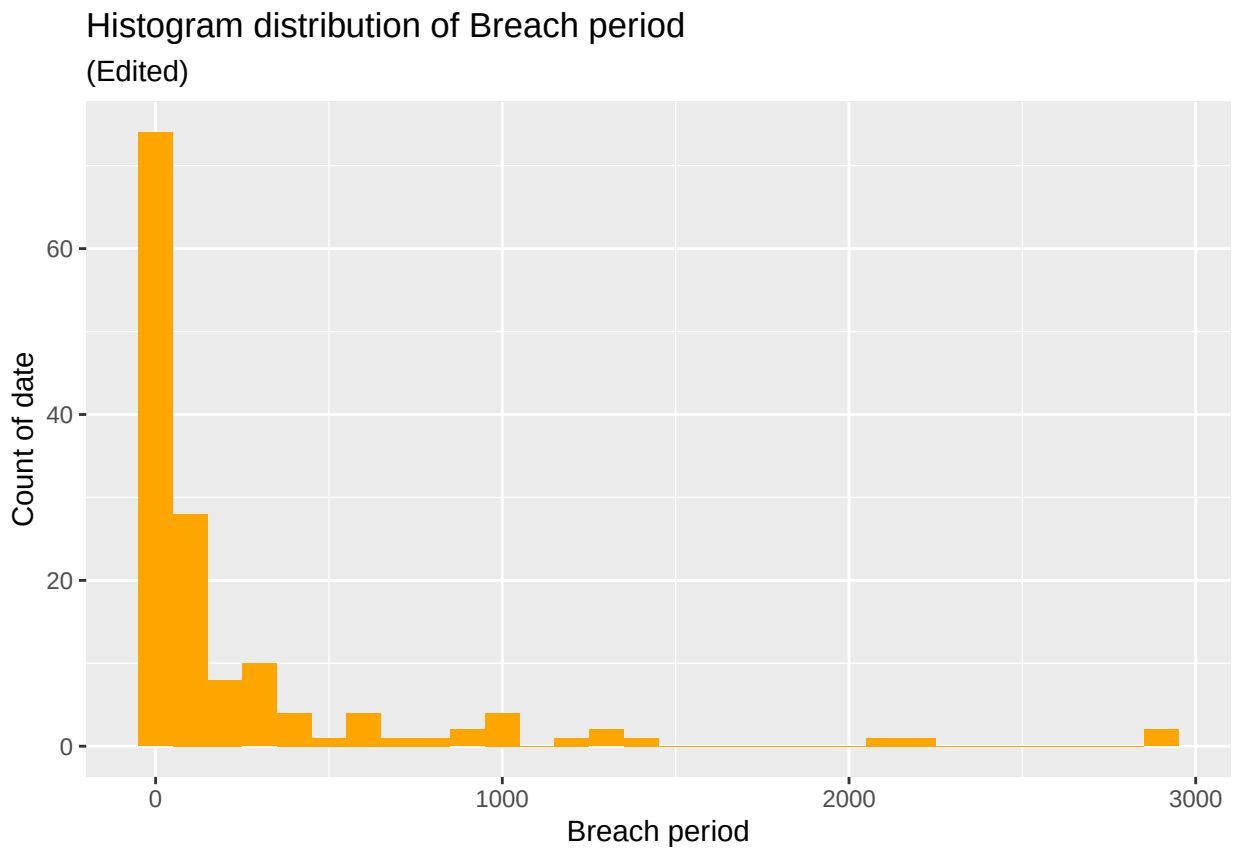
```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



This distribution below tells us that there are a few counts of dates that have extremely large breach periods. This indicates that for a couple breaches, it took a long time for the breach to happen.

```
cyberdata_BP_concentrated <- data.frame(cyberdata_BE_replacedNA)
cyberdata_BP_concentrated <- cyberdata_BE_replacedNA %>% filter(breach_period>0)

ggplot(cyberdata_BP_concentrated, aes(breach_period)) +
geom_histogram(binwidth = 100, fill="orange") +
labs(x="Breach period", y="Count of date", title="Histogram distribution of Breach period", subtitle="(
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



After filtering the data set to only look at values that have breach period counts greater than 1, we see that the most common breach period is 1 day followed by 108 days and then 2 days. The largest breach period that occurs is 2891 days.

```
cyberdata_BP_concentrated %>% group_by(breach_period) %>% count() %>% filter(n>1)
```

```
## # A tibble: 19 x 2
## # Groups:   breach_period [19]
##    breach_period     n
##    <drtn>        <int>
##  1   1 days         16
##  2   2 days          7
##  3   3 days          6
##  4   5 days          3
##  5   7 days          2
##  6   8 days          3
##  7  11 days          2
##  8  12 days          2
##  9  14 days          4
## 10  16 days          3
## 11  32 days          3
## 12  46 days          2
## 13  49 days          2
## 14  50 days          2
## 15  56 days          2
## 16  62 days          2
```

```
## 17 108 days          8
## 18 335 days          3
## 19 914 days          2
```

```
cyberdata_BP_concentrated %>%
summarize(max_breach_period = max(breach_period), min_breach_period = min(breach_period))
```

```
##   max_breach_period min_breach_period
## 1         2891 days            1 days
```

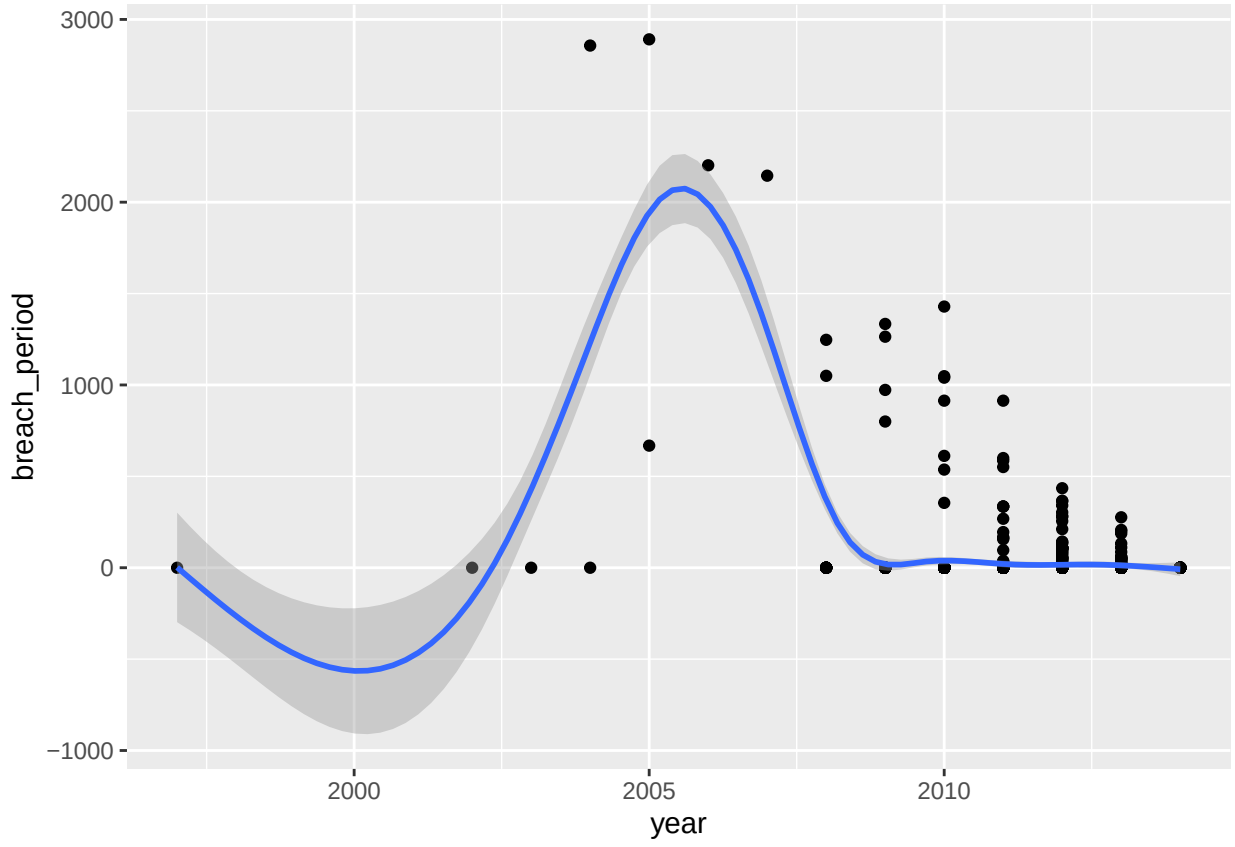
Created a new data frame by combining the State and Individuals Affected variables with the created detection length and breach period values.

Graph below shows no trend since values are scattered everywhere. There is no steady linear trend to be noticed between the year and breach period.

```
cyberdata_final_combined <- cbind(cyberdata_combined, cyberdata_BE_replacedNA)
cyberdata_final_combined <- subset(cyberdata_final_combined, select= -c(1:7))

ggplot(data = cyberdata_final_combined, aes(x = year, y = breach_period)) + geom_point() +
geom_smooth()
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



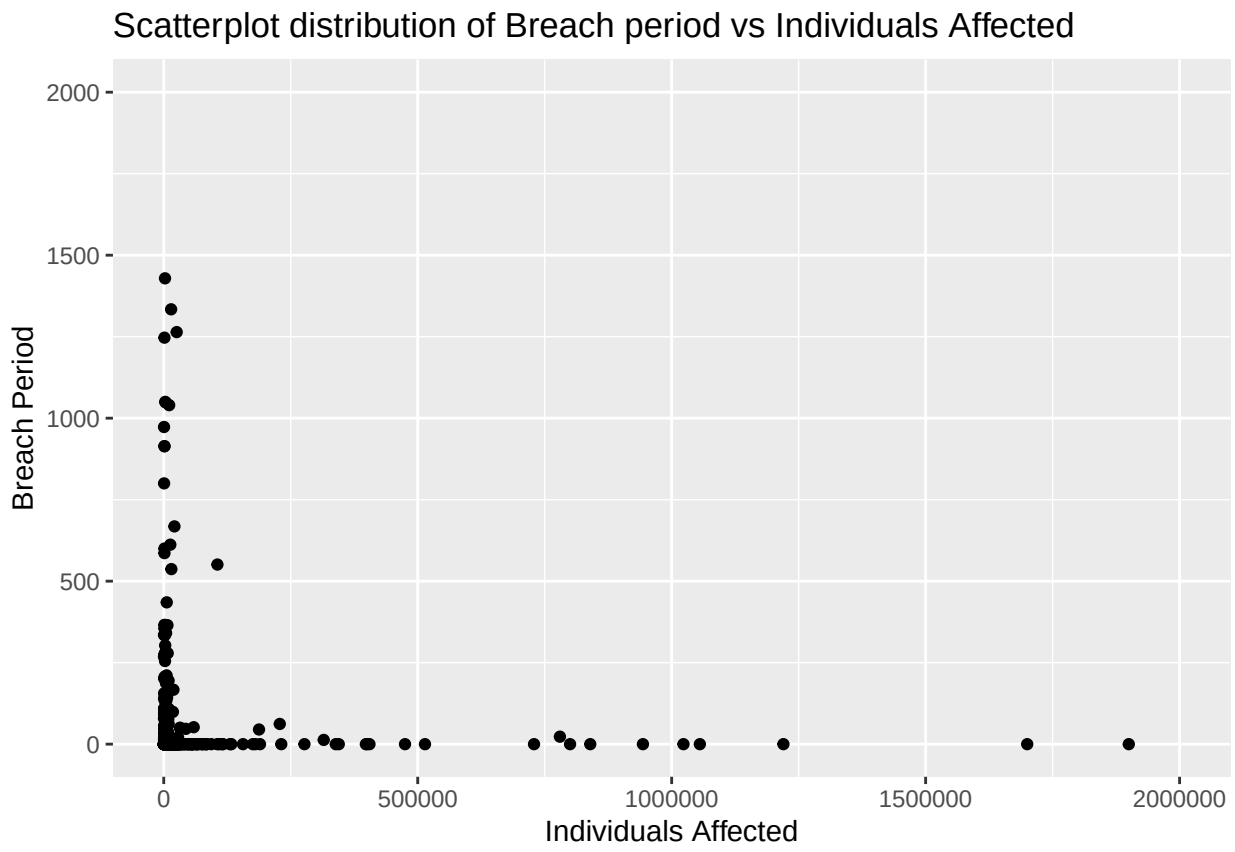
Analysing breach period against the Individuals Affected

This shows us that the number of people getting affected when the breach takes a long time is fairly low. Similarly, when more people get affected, the breach period is low.

The first graph is a geom_point() graph of a more zoomed in version such as the y limit being between 0 and 2000 and the x limit being between 0 and 2,000,000. The second graph is the full distribution.

```
ggplot(data = cyberdata_final_combined) +
geom_point(mapping = aes(x = Individuals_Affected, y = breach_period)) + ylim(0,2000) +
xlim(0,2000000) +
labs(x="Individuals Affected", y="Breach Period", title="Scatterplot distribution of Breach period vs In
```

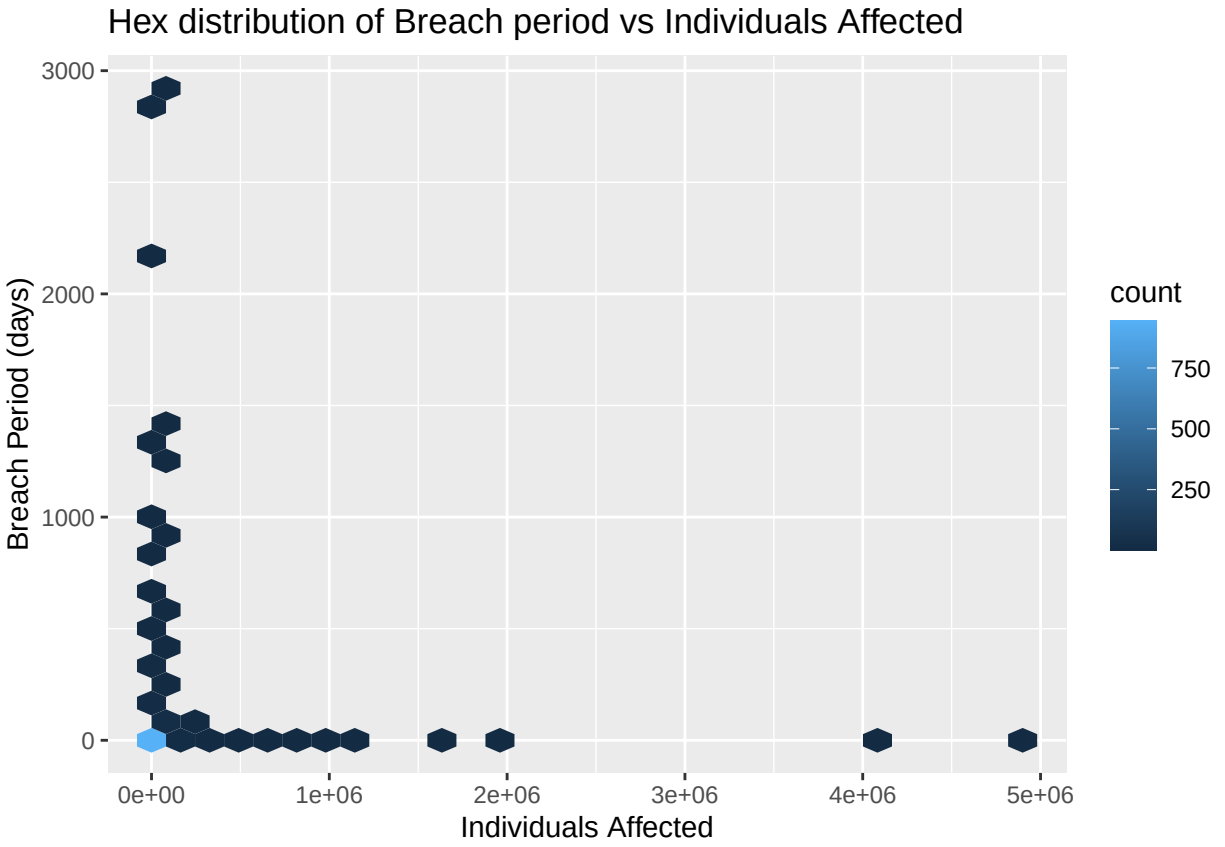
```
## Warning: Removed 6 rows containing missing values (geom_point).
```



Important

```
ggplot(data = cyberdata_final_combined, mapping = aes(x = Individuals_Affected, y = breach_period)) +
geom_hex() +
labs(x="Individuals Affected", y="Breach Period (days)", title="Hex distribution of Breach period vs In
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



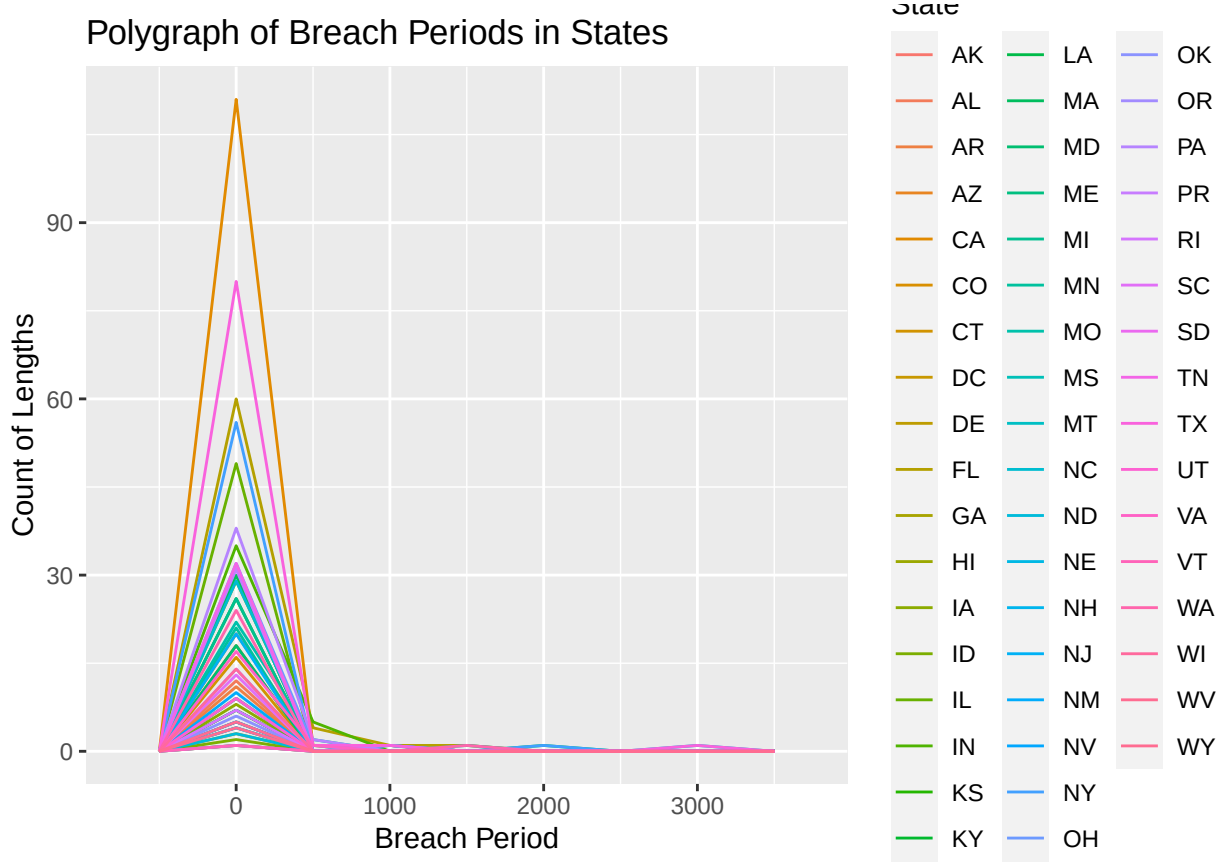
Analyzing breach period with States

The analysis below shows that the states usually have a fairly pyramid shaped distribution of the breach periods around 0. This means that a majority of the states have extremely small breach periods, or usually have single day breaches. There are only a couple states that have longer breach periods and they are determined to be Massachusetts(2145), North Carolina(2857), New York(2202) and Texas(2891). We can see this in the jitter graph distribution as well since these 4 states have the highest distribution.

I created different graphs so I can see the trend better and decide which graph shows it the best.

```
ggplot(data = cyberdata_final_combined, mapping = aes(x = breach_period)) +
geom_freqpoly(mapping = aes(colour = State), binwidth = 500) +
labs(x="Breach Period", y="Count of Lengths", title="Polygraph of Breach Periods in States")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

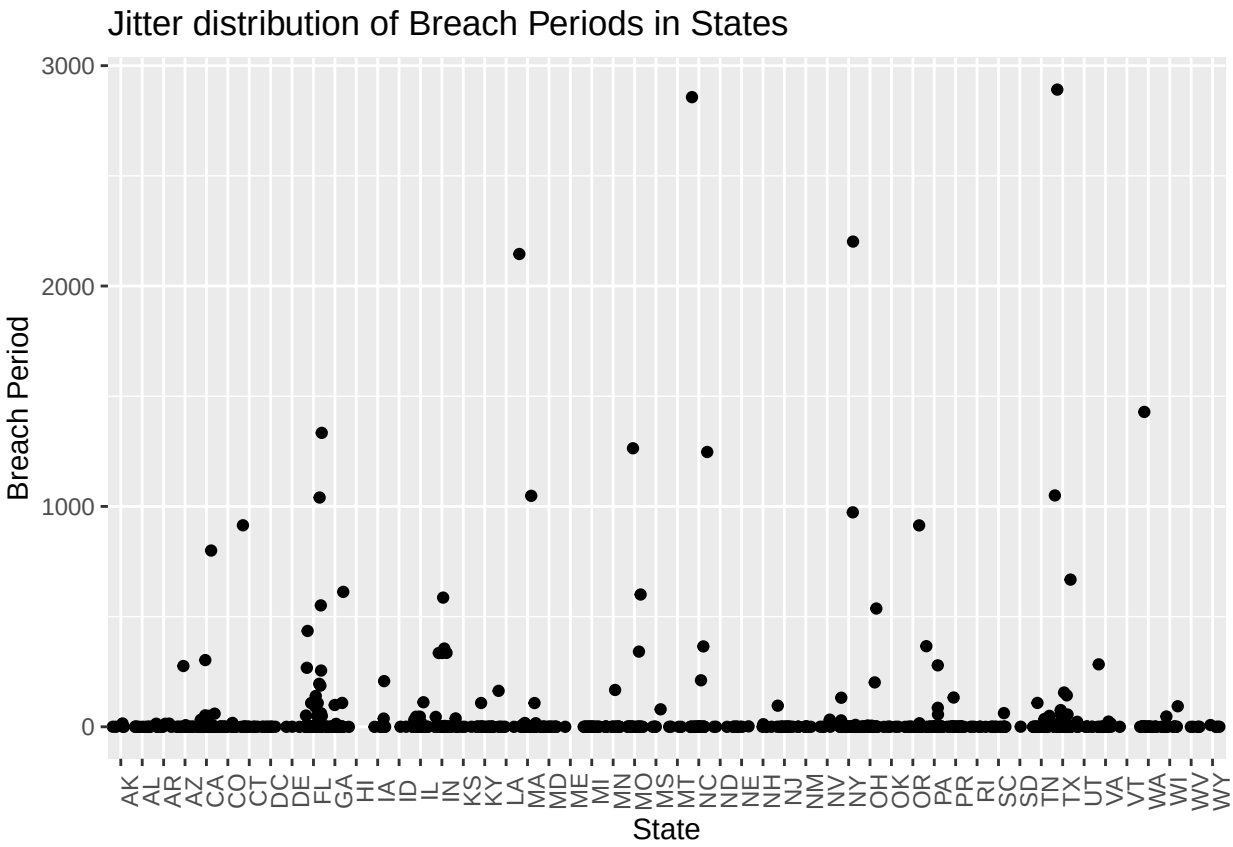



```
cyberdata_final_combined %>% group_by(State) %>%
summarize(max_period = max(breach_period, na.rm = TRUE))
```

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

## # A tibble: 52 x 2
##    State max_period
##    <chr> <drtn>
##  1 AK      15 days
##  2 AL       1 days
##  3 AR      14 days
##  4 AZ     276 days
##  5 CA     800 days
##  6 CO      18 days
##  7 CT     914 days
##  8 DC       0 days
##  9 DE       0 days
## 10 FL    1334 days
## # ... with 42 more rows
```

```
ggplot(data = cyberdata_final_combined, aes(x=State, y=breach_period)) + geom_jitter() +
theme(axis.text.x = element_text(angle = 90)) +
labs(x="State", y="Breach Period", title="Jitter distribution of Breach Periods in States")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

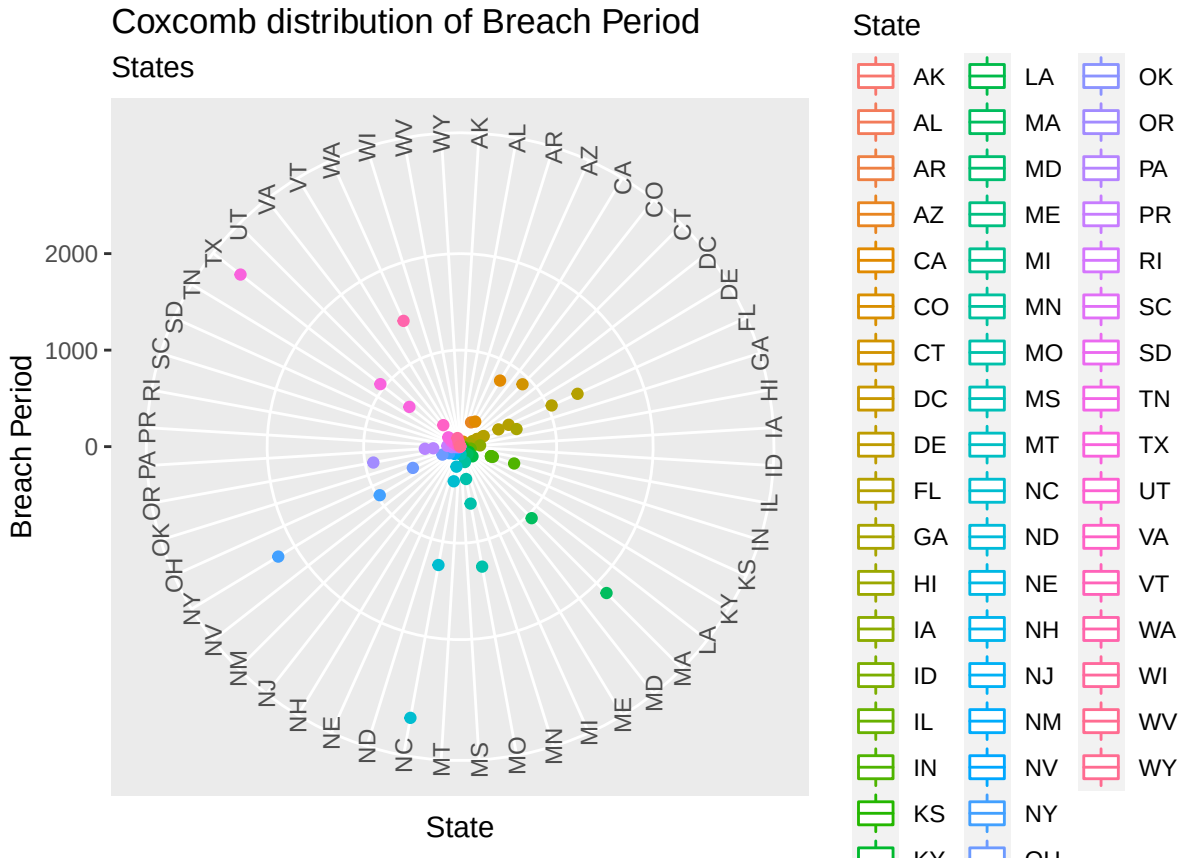


Important

```
ggplot(data = cyberdata_final_combined, aes(x=State, y=breach_period, color=State)) +
geom_boxplot() +
theme(axis.text.x = element_text(angle = 90)) +
labs(x="State", y="Breach Period", title="Coxcomb distribution of Breach Period", subtitle = "States") +
coord_flip() + coord_polar()
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```



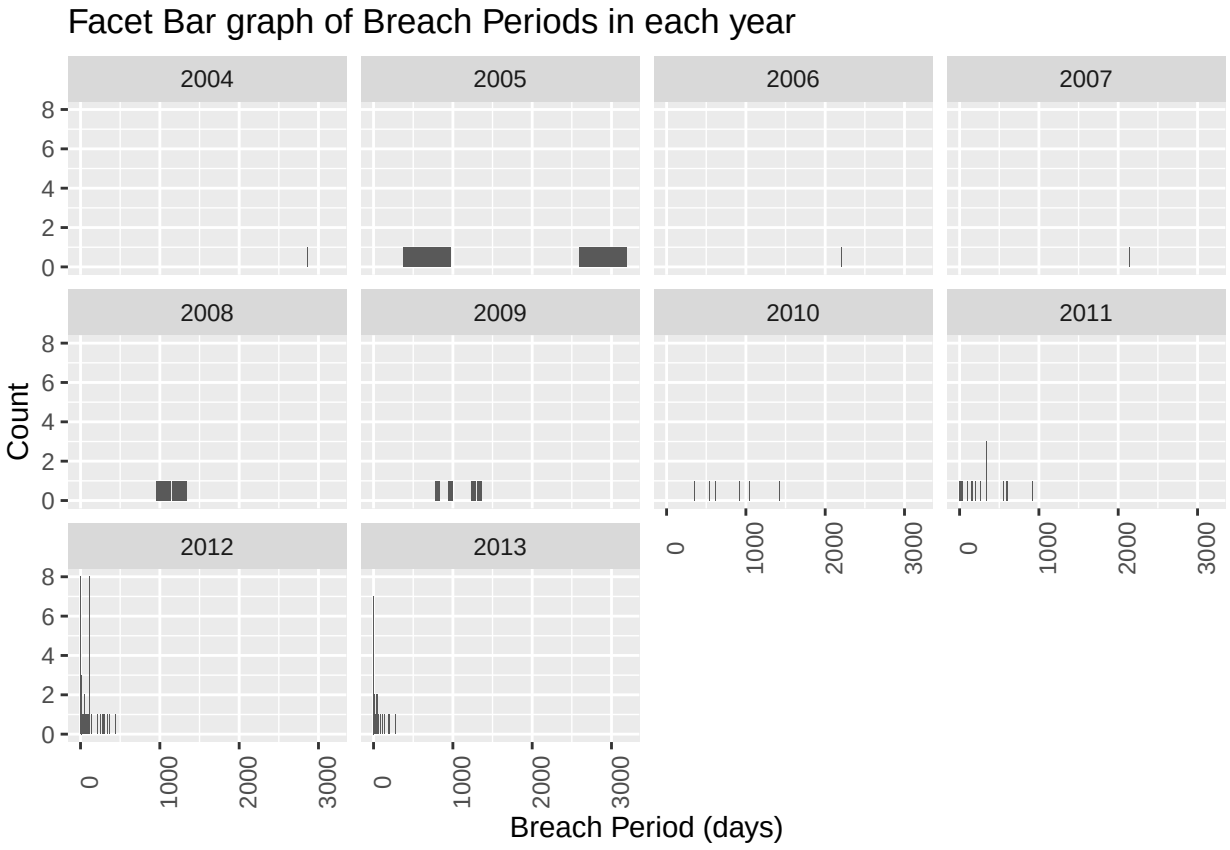
Analyzing Breach periods in the year of the Date of Breach

We can see that the year 2005 followed by 2008 has the longest breach period. This means that this year had a majority of the multiple day breaches.

Important

```
ggplot(data = filter(cyberdata_final_combined, breach_period>0)) +
geom_bar(mapping = aes(x=breach_period)) +
facet_wrap(~year) +
theme(axis.text.x = element_text(angle = 90)) +
labs(x="Breach Period (days)", y="Count", title="Facet Bar graph of Breach Periods in each year")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```




```
cyberdata_final_combined %>% group_by(year, breach_period) %>% count()
```

```
## # A tibble: 116 x 3
## # Groups:   year, breach_period [116]
##     year breach_period     n
##    <dbl> <drtn>        <int>
##  1  1997    0 days          1
##  2  2002    0 days          1
##  3  2003    0 days          1
##  4  2004    0 days          1
##  5  2004 2857 days          1
##  6  2005  668 days          1
##  7  2005 2891 days          1
##  8  2006 2202 days          1
##  9  2007 2145 days          1
## 10  2008    0 days         11
## # ... with 106 more rows
```

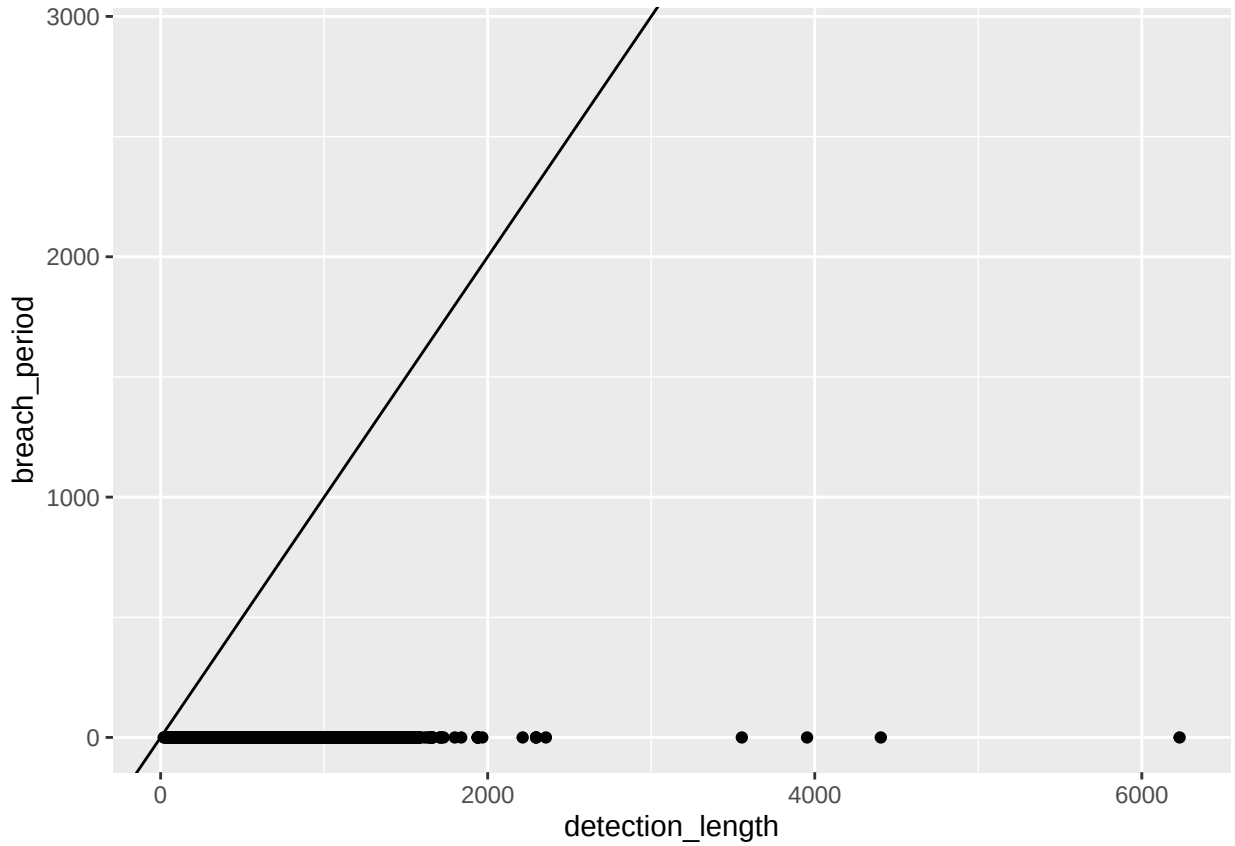
Analyzing detection length versus year and versus breach period

Abline is added that shows that there is no linear trend in the two graphs. For breach period versus detection length, due to there being so many 0 values, the scatterplot shows no trend. The abline added to it doesn't make sense. Hence, this distribution is not modeled.

For the detection length versus year graph, detection length decreases upto 2014 since it subtracts dates in the Date of Breach from Date posted which are all 2014. Similarly no linear trend with breach period. This distribution is also not modeled further.

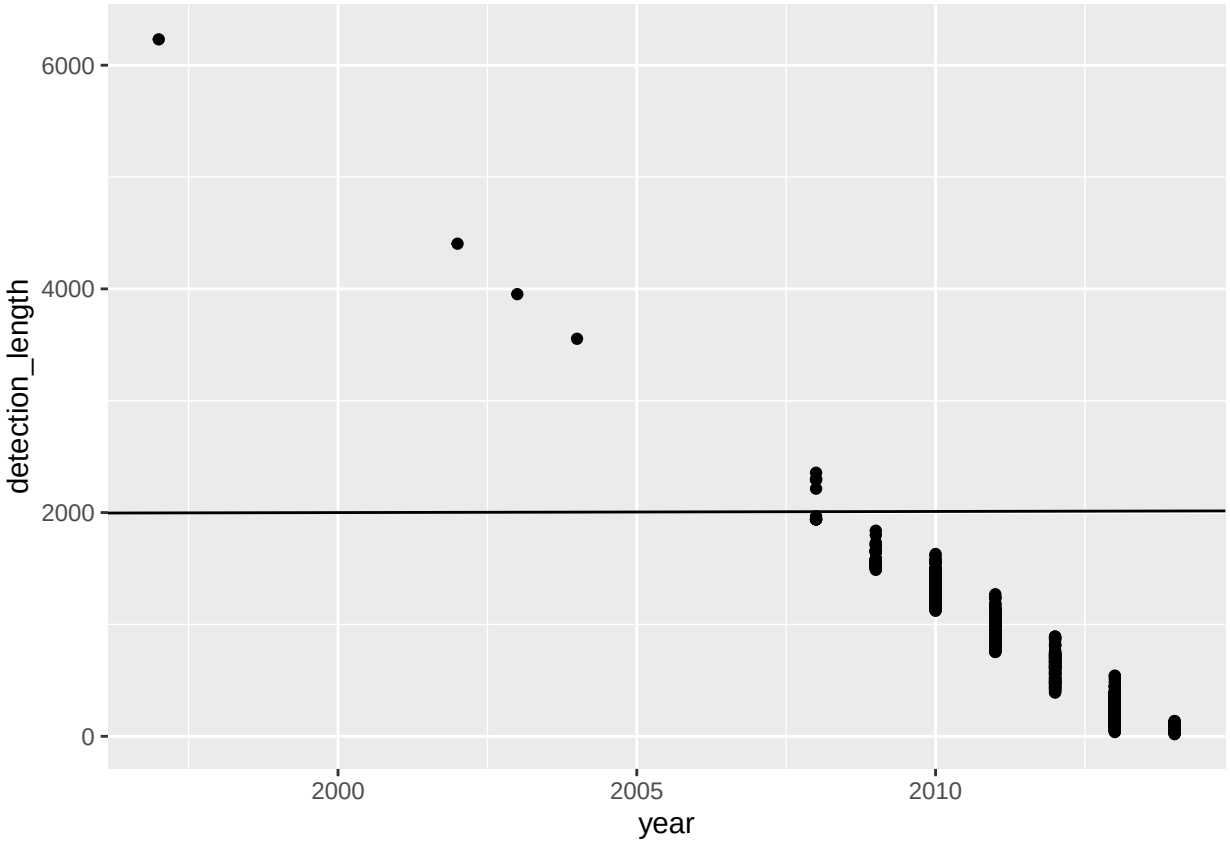
```
cyberdata_test_combined <- data.frame(cyberdata_final_combined)

cyberdata_test_combined <- cyberdata_test_combined %>%
mutate(detection_length = as.numeric(detection_length), breach_period = as.numeric(breach_period))

cyberdata_test_combined %>%
ggplot(aes(x = detection_length, y = breach_period)) +
geom_point() + geom_abline()
```

```
## Warning: Removed 146 rows containing missing values (geom_point).
```




```
cyberdata_test_combined %>%
ggplot(aes(x = year, y = detection_length)) +
geom_point() + geom_abline()
```

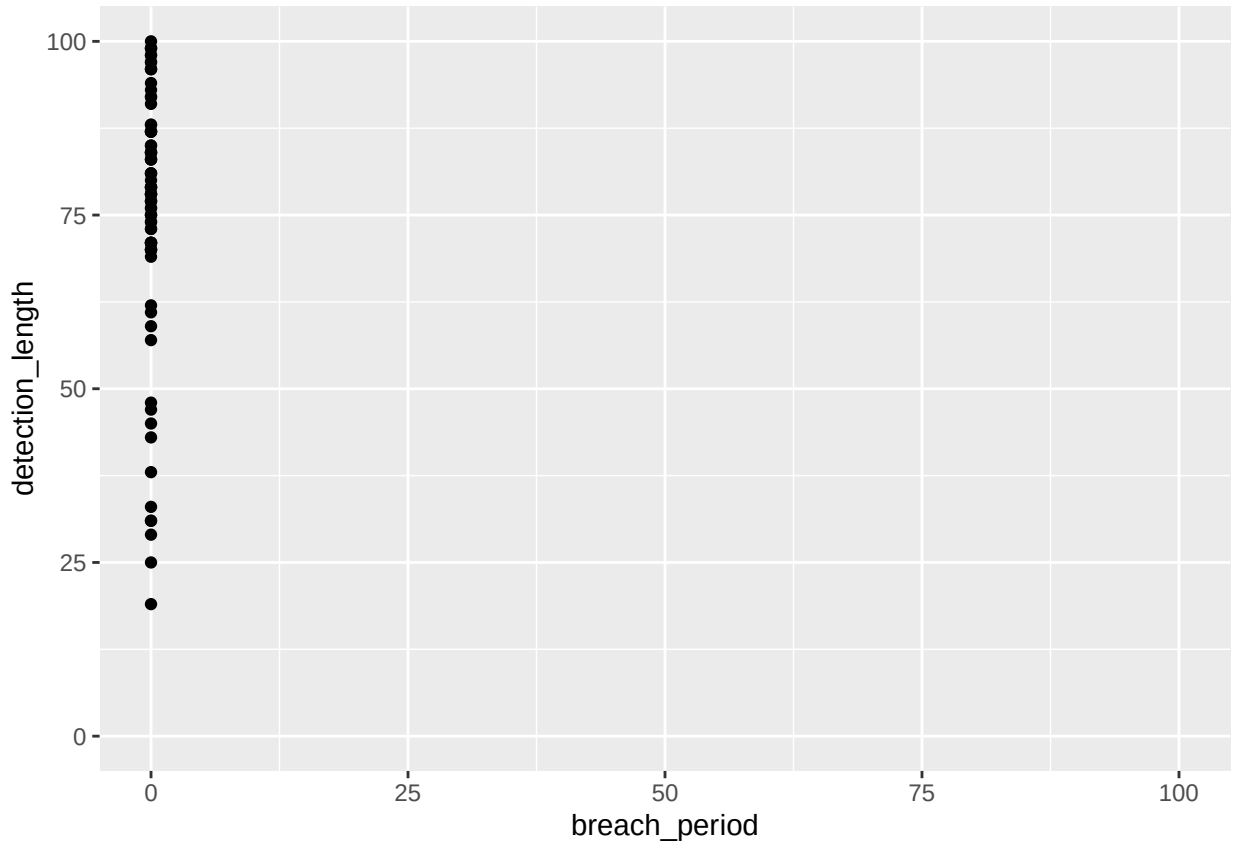
```
## Warning: Removed 146 rows containing missing values (geom_point).
```



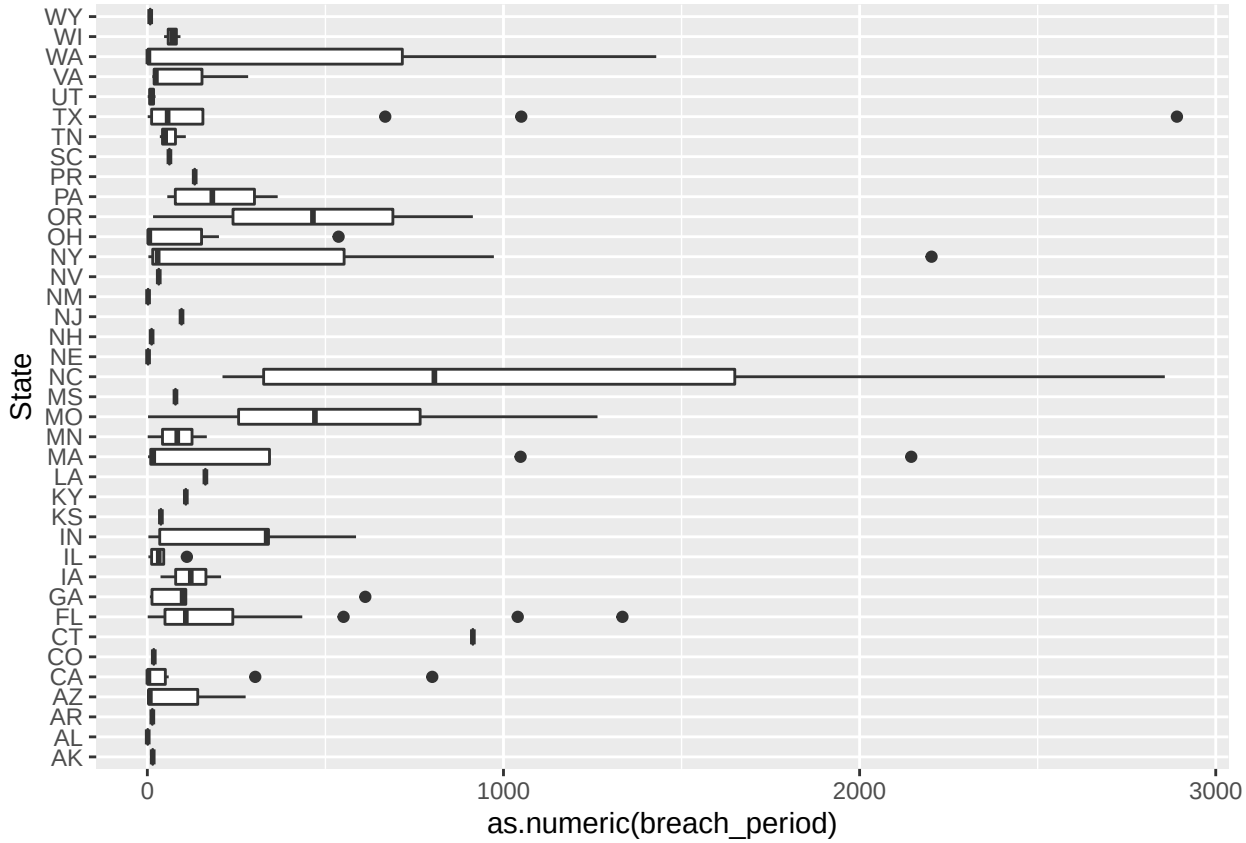
Graphs made of detection length versus breach period but no trend found - we just see that breach period stays at 0 as the detection length increases. Breach period is plotted against the states as boxplot resulting in the same findings as above. The bar graph distribution shows how the values are distributed apart from 0 in an L shaped.

```
cyber_test <- cyberdata_final_combined %>% filter(breach_period!=0)

ggplot(data = cyberdata_test_combined, aes(x=breach_period, y = detection_length)) +
geom_point() +
xlim(0,100) + ylim(0,100)
```

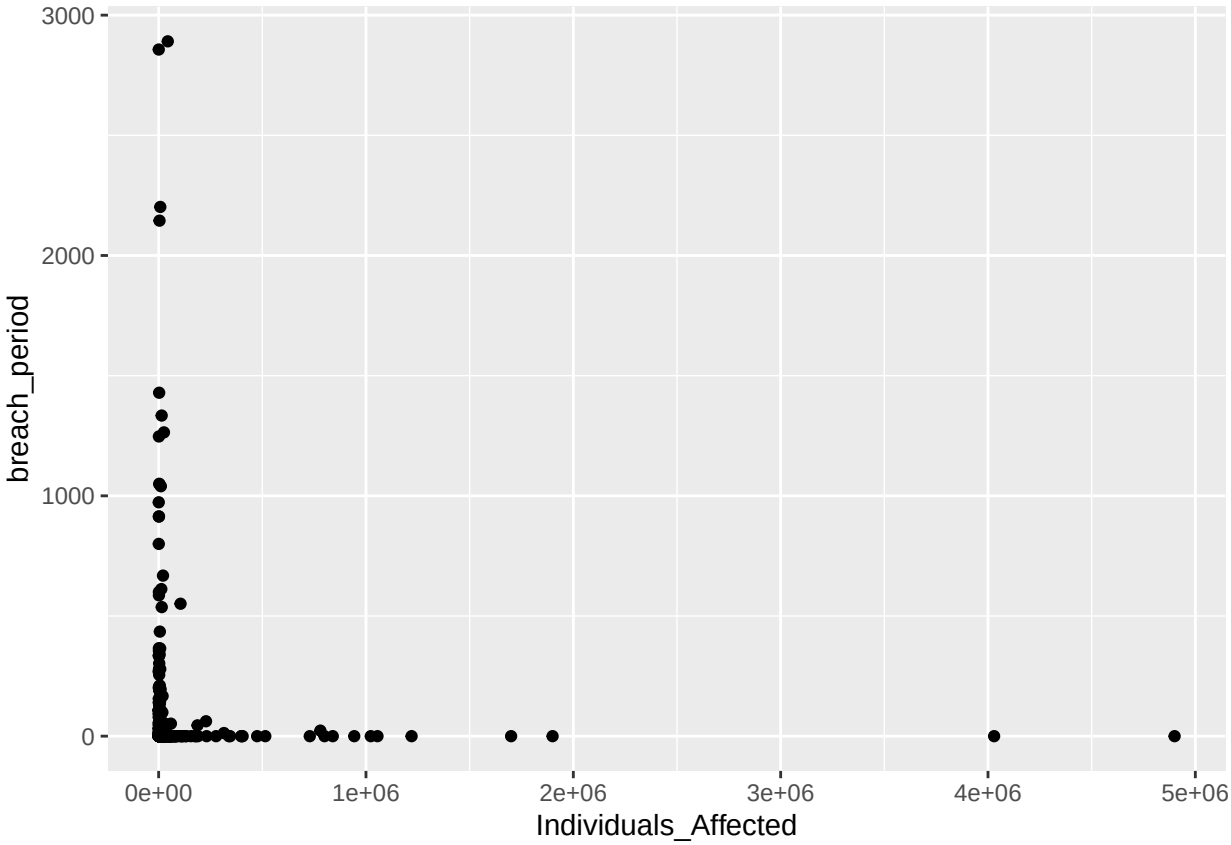
```
## Warning: Removed 975 rows containing missing values (geom_point).
```




```
ggplot(data = cyber_test, mapping = aes(x = State, y = as.numeric(breach_period))) +
geom_boxplot() + coord_flip()
```



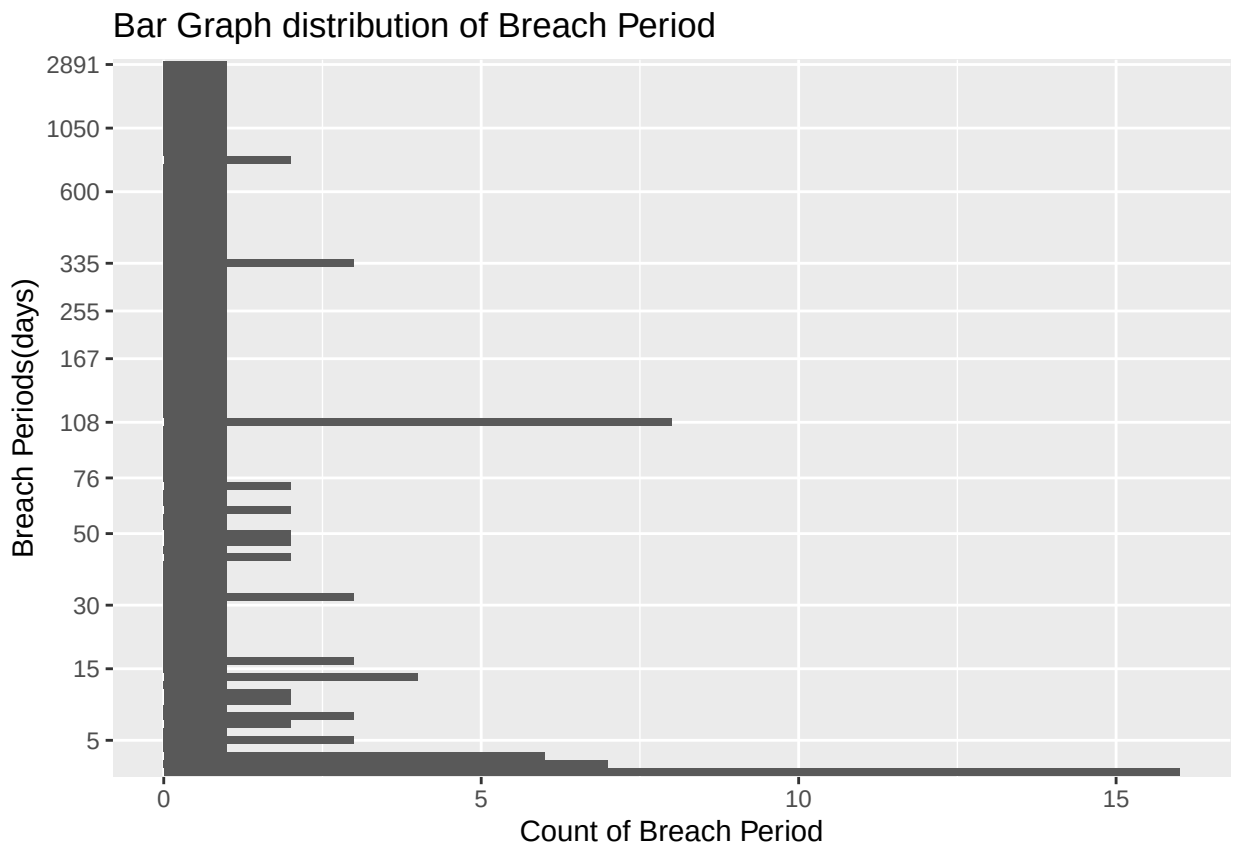

```
ggplot(data = cyberdata_test_combined) +
geom_point(mapping = aes(x = Individuals_Affected, y = breach_period))
```



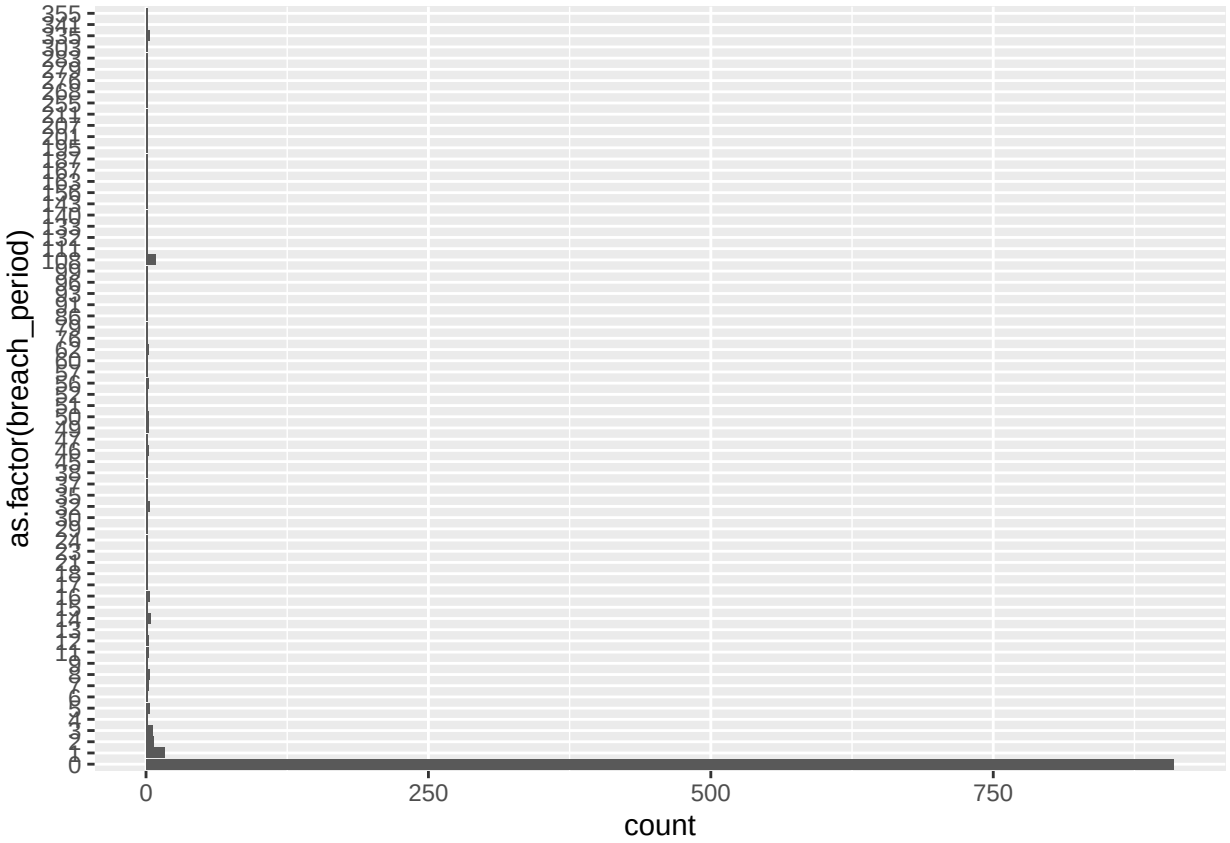
Important

The breach period bar graph distribution shows that a majority of the values are at 0 with an extremely high count. At around 108 breach periods, is the next highest count. The highest breach period is 2891 days with a count of 1.

```
cyberdata_final_combined %>%
filter(breach_period>0) %>%
ggplot(aes(x=as.factor(breach_period))) +
geom_bar() +coord_flip() +
labs(x="Breach Periods(days)", y="Count of Breach Period", title="Bar Graph distribution of Breach Peri
scale_x_discrete(breaks = c(5,15,30,50,76,108,167,255,335,600,1050,2891))
```




```
cyberdata_final_combined %>%
filter(breach_period<365) %>%
ggplot(aes(x=as.factor(breach_period))) +
geom_bar() +coord_flip()
```



Residuals and predictions for month of breach and detection length

No model can be made since the correlation coefficient is NA. This is because month is a categorical variable. The red dots represent the predicted values in the model, which are mostly in the center of the ranges for each month.

```
model <- lm(detection_length ~ month_of_breach, data = cyberdata_test_combined)
model
```

```
##
## Call:
## lm(formula = detection_length ~ month_of_breach, data = cyberdata_test_combined)
##
## Coefficients:
##       (Intercept)  month_of_breach02  month_of_breach03  month_of_breach04
##           919.779           -181.609            -37.333           -113.021
## month_of_breach05  month_of_breach06  month_of_breach07  month_of_breach08
##             4.888             19.188            -59.502           -244.426
## month_of_breach09  month_of_breach10  month_of_breach11  month_of_breach12
##           -88.073           -101.929           -194.484           -171.083
```

```
summary(model)
```

```
##
## Call:
## lm(formula = detection_length ~ month_of_breach, data = cyberdata_test_combined)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -914.0 -505.4   27.1  394.3 5311.2
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        919.779     60.864  15.112  < 2e-16 ***
## month_of_breach02 -181.609     85.584  -2.122  0.03411 *
## month_of_breach03  -37.333     86.849  -0.430  0.66740
## month_of_breach04 -113.021     94.037  -1.202  0.22973
## month_of_breach05    4.888     88.254   0.055  0.95585
## month_of_breach06   19.188     94.943   0.202  0.83989
## month_of_breach07  -59.502     92.767  -0.641  0.52142
## month_of_breach08 -244.426     91.594  -2.669  0.00775 **
## month_of_breach09  -88.073     84.660  -1.040  0.29848
## month_of_breach10 -101.929     87.674  -1.163  0.24530
## month_of_breach11 -194.484     88.254  -2.204  0.02780 *
## month_of_breach12 -171.083     91.223  -1.875  0.06106 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 564.4 on 897 degrees of freedom
##   (146 observations deleted due to missingness)
## Multiple R-squared:  0.0208, Adjusted R-squared:  0.008787
## F-statistic: 1.732 on 11 and 897 DF,  p-value: 0.0621
```

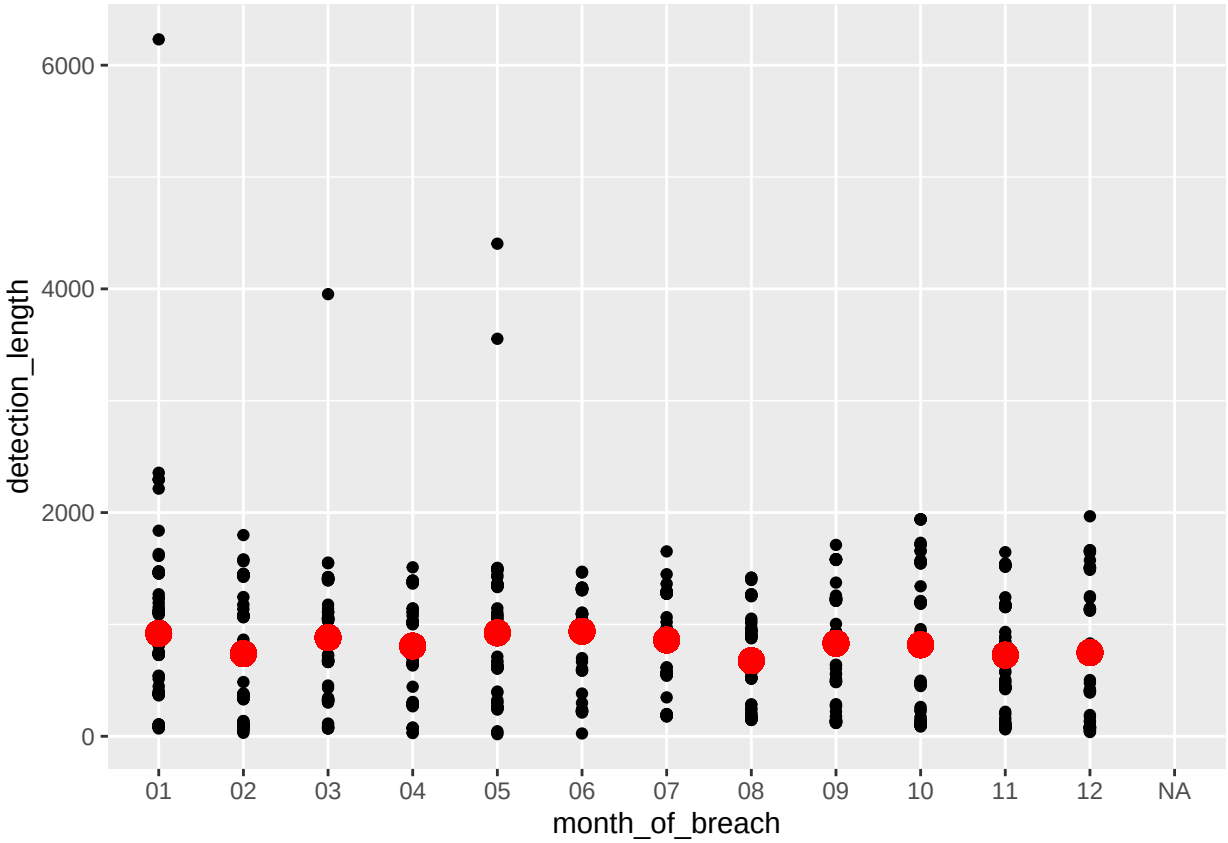
```
model_edited <- cyberdata_test_combined %>% add_predictions(model)
coef(model_edited)
```

```
## NULL
```

```
ggplot(cyberdata_test_combined, aes(x=month_of_breach)) +
geom_point(aes(y = detection_length)) +
geom_point(data = model_edited, aes(y = pred), colour = "red", size = 4)
```

```
## Warning: Removed 146 rows containing missing values (geom_point).
```

```
## Warning: Removed 146 rows containing missing values (geom_point).
```



Model Analysis for Individuals Affected and detection length

A slightly positive trend is seen as determined by the correlation of coefficient being 0.0081. Hence, while the trend isn't completely linear, it also isn't completely non-existent. The p-value is found to be 0.81 which is larger than 0.05, meaning we fail to reject the null hypothesis. This means that the true correlation is more equal to 0 than not. And the confidence interval contains the value 0, supporting this further.

Important

```
model1 <- lm(detection_length ~ Individuals_Affected, data = cyberdata_test_combined)
model1
```

```
##
## Call:
## lm(formula = detection_length ~ Individuals_Affected, data = cyberdata_test_combined)
##
## Coefficients:
##          (Intercept)  Individuals_Affected
##            8.221e+02             1.886e-05
```

```r
summary(model1)
```

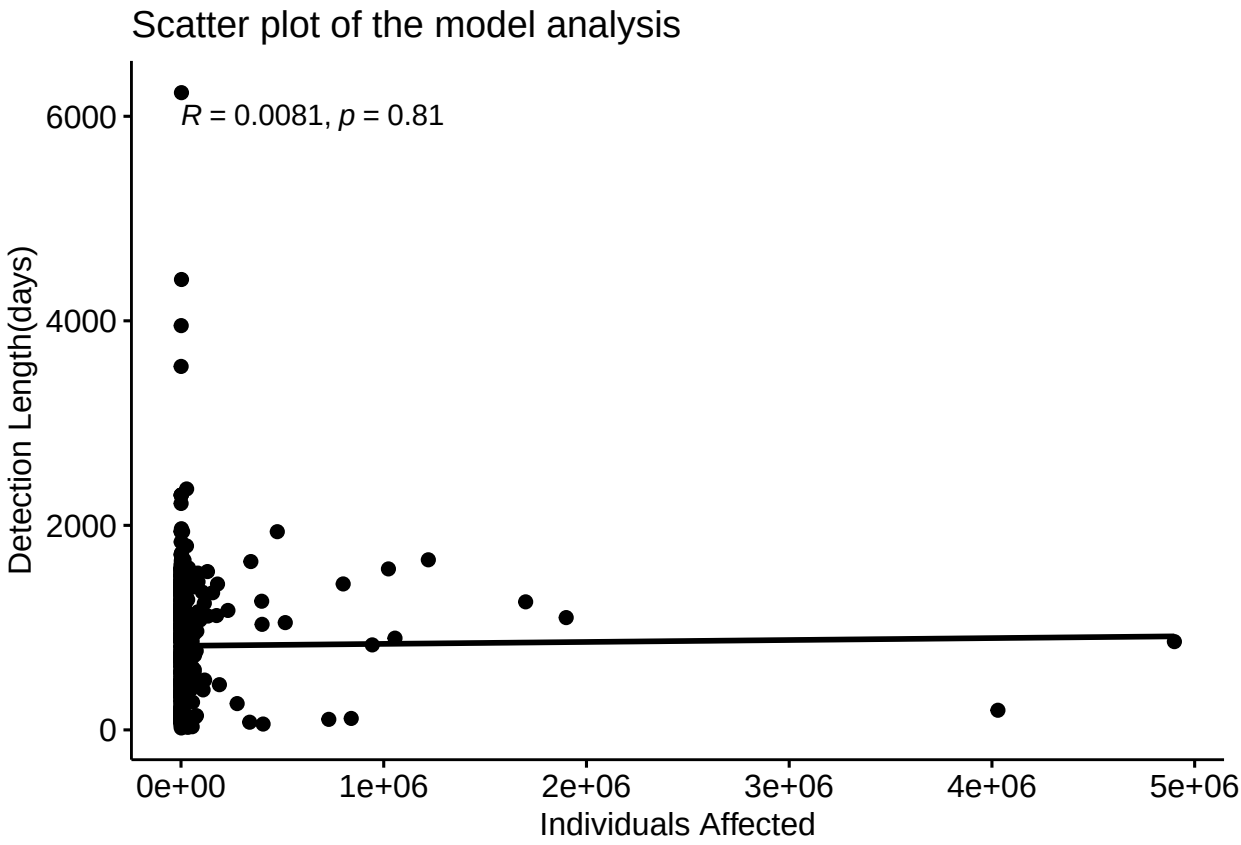
```
##
## Call:
## lm(formula = detection_length ~ Individuals_Affected, data = cyberdata_test_combined)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -803.1 -487.1   22.8  402.9 5408.9
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          8.221e+02  1.898e+01  43.313   <2e-16 ***
## Individuals_Affected 1.886e-05  7.725e-05   0.244    0.807
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 567.2 on 907 degrees of freedom
##   (146 observations deleted due to missingness)
## Multiple R-squared:  6.573e-05,  Adjusted R-squared:  -0.001037
## F-statistic: 0.05962 on 1 and 907 DF,  p-value: 0.8072
```

```r
res <- cor.test(cyberdata_test_combined$detection_length, cyberdata_test_combined$Individuals_Affected,
res
```

```
##
##  Pearson's product-moment correlation
##
## data:  cyberdata_test_combined$detection_length and cyberdata_test_combined$Individuals_Affected
## t = 0.24418, df = 907, p-value = 0.8072
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.05694615  0.07309247
## sample estimates:
##         cor
## 0.008107438
```

```r
ggscatter(cyberdata_test_combined, x="Individuals_Affected", y="detection_length", add = "reg.line", co
labs(x="Individuals Affected", y="Detection Length(days)", title="Scatter plot of the model analysis")
```

```
## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 146 rows containing non-finite values (stat_smooth).

## Warning: Removed 146 rows containing non-finite values (stat_cor).

## Warning: Removed 146 rows containing missing values (geom_point).
```



FINDINGS

After conducting the above analysis, my final findings show that overall, the amount of time that passes before a breach is found and the duration of breaches are both fairly low. This indicates that the company tends to detect breaches fairly quickly after the breach occurs. It also faces breaches that usually occur within one day and thus are maybe not very complex. However, when looking at the states, North Carolina and Massachusetts have the largest amount of detection lengths and breach periods. This means that these two states take the longest time to detect a breach and have breaches that span over many days. The analysis also determined that the number of individuals affected doesn't seem to increase with the longer breach durations, which is a trend I expected. One would think that larger breach durations correspond to more complex breaches, and thus more individuals being affected, but this trend was not shown. This might have happened because the company might be good at protecting the individuals from losing too much of their data during breaches. However, as for the number of individuals being affected with longer detection lengths, we can determine a slightly positive correlation. This slight linear trend is expected because the more time it takes for the company to detect the breach, more individuals will be harmed in the process since they might have to delay their work or take precautions.