R Lab 1: Incidence

STAT 244NF: Infectious Disease Modeling

Sample Solutions

12 September, 2021

In this lab, we are going to work with the *incidence* package, which was developed to so that users could easily compute, manipulate, visualize, and model incidences from data with dates. This lab was motivated by parts of the worked example on Ebola that is presented in the *incidence* package vignette, by Jombart and Kamvar (https://cran.r-project.org/web/packages/incidence/vignettes/overview.html).

Installing and loading packages

If you using RStudio in the same place that you were for Lab 0, then you have already installed the *incidence* package. You can check to see if it is installed by clicking on the **Packages** tab in the lower right window of your RStudio session and typing incidence into the search bar. If it is not installed, please install it now.

```r
# install.packages("incidence)

library(incidence)
```

Run the following code chunk to load the incidence library. We will also need to load ggplot2 and outbreaks. Please add two lines to the code chunk below after `library(incidence)` to load ggplot2 and outbreaks.

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

```r
library(outbreaks)
```

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

```r
library(ggplot2)
```

Examining the data: Simulated Ebola data

```r
data("ebola_sim")
```

We will be working with the `ebola_sim` data, available in the `outbreaks` package. Although it loads automatically with the package, run the following line of code to make `ebola_sim` show up under Values in the Environment tab in the upper right corner.

```r
class(ebola_sim)
```

The `ebola_sim` is a different kind of object than you may be used to. To find out more about it, we can use the functions `class` and `str` (structure) and pass the variable `ebola_sim` to them (e.g., run `class(ebola_sim)`). In the empty code chunk below, determine the class and structure of `ebola_sim`.

```
## [1] "list"
```

```r
str(ebola_sim)
```

```
## List of 2
##  $ linelist:'data.frame':    5888 obs. of  11 variables:
##   ..$ case_id                : chr [1:5888] "d1fafd" "53371b" "f5c3d8" "6c286a" ...
##   ..$ generation             : int [1:5888] 0 1 1 2 2 0 3 3 2 3 ...
##   ..$ date_of_infection      : Date[1:5888], format: NA "2014-04-09" ...
##   ..$ date_of_onset          : Date[1:5888], format: "2014-04-07" "2014-04-15" ...
##   ..$ date_of_hospitalisation: Date[1:5888], format: "2014-04-17" "2014-04-20" ...
##   ..$ date_of_outcome        : Date[1:5888], format: "2014-04-19" NA ...
##   ..$ outcome                : Factor w/ 2 levels "Death","Recover": NA NA 2 1 2 NA 2 1 2 1 ...
##   ..$ gender                 : Factor w/ 2 levels "f","m": 1 2 1 1 1 1 1 1 2 2 ...
##   ..$ hospital               : Factor w/ 11 levels "Connaught Hopital",..: 4 2 7 NA 7 NA 2 9 7 11 ..
##   ..$ lon                    : num [1:5888] -13.2 -13.2 -13.2 -13.2 -13.2 ...
##   ..$ lat                    : num [1:5888] 8.47 8.46 8.48 8.46 8.45 ...
##  $ contacts:'data.frame':    3800 obs. of  3 variables:
##   ..$ infector: chr [1:3800] "d1fafd" "cac51e" "f5c3d8" "0f58c4" ...
##   ..$ case_id : chr [1:3800] "53371b" "f5c3d8" "0f58c4" "881bd4" ...
##   ..$ source  : Factor w/ 2 levels "funeral","other": 2 1 2 2 2 1 2 2 2 2 ...
```

`ebola_sim` is a list of length 2, consisting of two data frames, `linelist` and `contacts`.

```r
ebola_linelist <- ebola_sim$linelist
str(ebola_linelist)
```

Based on the output from `str(ebola_sim)`, we can see that `ebola_sim` is a list of two data frames, which are called `linelist` and `contacts`. Let's get these data into a more familiar form, namely, two data frames. In the code chunk below, assign `ebola_sim$linelist` to `ebola_linelist` and assign `ebola_sim$contacts` to `ebola_sim_contacts`.

```
## 'data.frame':    5888 obs. of  11 variables:
##  $ case_id                : chr  "d1fafd" "53371b" "f5c3d8" "6c286a" ...
##  $ generation             : int  0 1 1 2 2 0 3 3 2 3 ...
##  $ date_of_infection      : Date, format: NA "2014-04-09" ...
##  $ date_of_onset          : Date, format: "2014-04-07" "2014-04-15" ...
##  $ date_of_hospitalisation: Date, format: "2014-04-17" "2014-04-20" ...
##  $ date_of_outcome        : Date, format: "2014-04-19" NA ...
##  $ outcome                : Factor w/ 2 levels "Death","Recover": NA NA 2 1 2 NA 2 1 2 1 ...
##  $ gender                    : Factor w/ 2 levels "f","m": 1 2 1 1 1 1 1 1 2 2 ...
##  $ hospital                  : Factor w/ 11 levels "Connaught Hopital",..: 4 2 7 NA 7 NA 2 9 7 11 ...
##  $ lon                       : num  -13.2 -13.2 -13.2 -13.2 -13.2 ...
##  $ lat                       : num  8.47 8.46 8.48 8.46 8.45 ...
```

```
ebola_sim_contacts <- ebola_sim$contacts
```

Examine your two data frames. What do you notice about them (focus on `case_id` if you are not sure where to start)? The `ebola_sim_contacts` data frame only has 3800 observations, whereas the `ebola_linelist` has 5888 observations. This suggests that we only have information on the source of infection for a subset of the observations.

Visualizing incidence

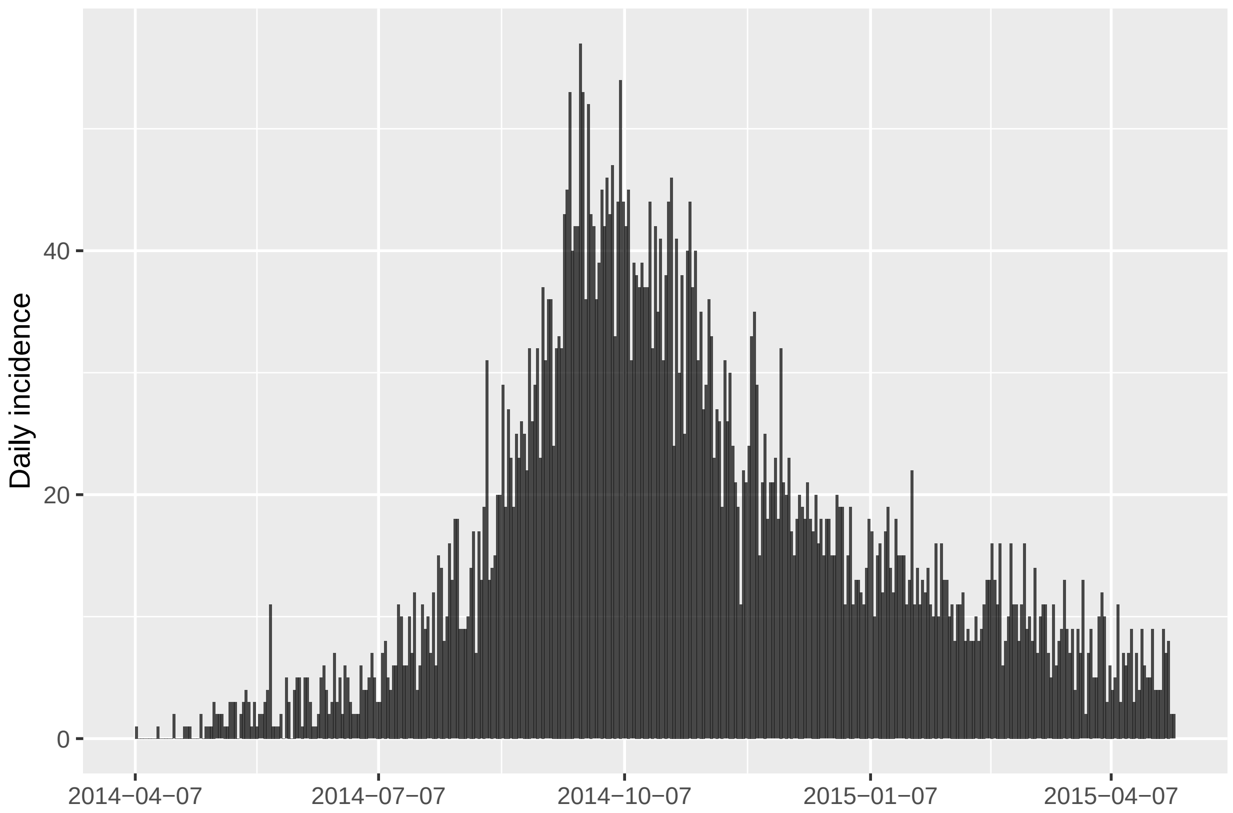
```
ebola_incidence <- incidence(dates = ebola_linelist$date_of_onset, interval = 1)
str(ebola_incidence)
```

The `ebola_linelist` data frame lists new cases of ebola and associated information about the cases, outcomes, etc. We can compute the daily incidence using the `incidence` function from the *incidence* package. Run the R code below to see how this works.

```
## List of 6
##  $ dates     : Date[1:389], format: "2014-04-07" "2014-04-08" ...
##  $ counts    : int [1:389, 1] 1 0 0 0 0 0 0 0 1 0 ...
##  $ timespan  : ’difftime’ num 389
##   ..- attr(*, "units")= chr "days"
##  $ interval  : int 1
##  $ n         : int 5888
##  $ cumulative: logi FALSE
##  - attr(*, "class")= chr "incidence"
```

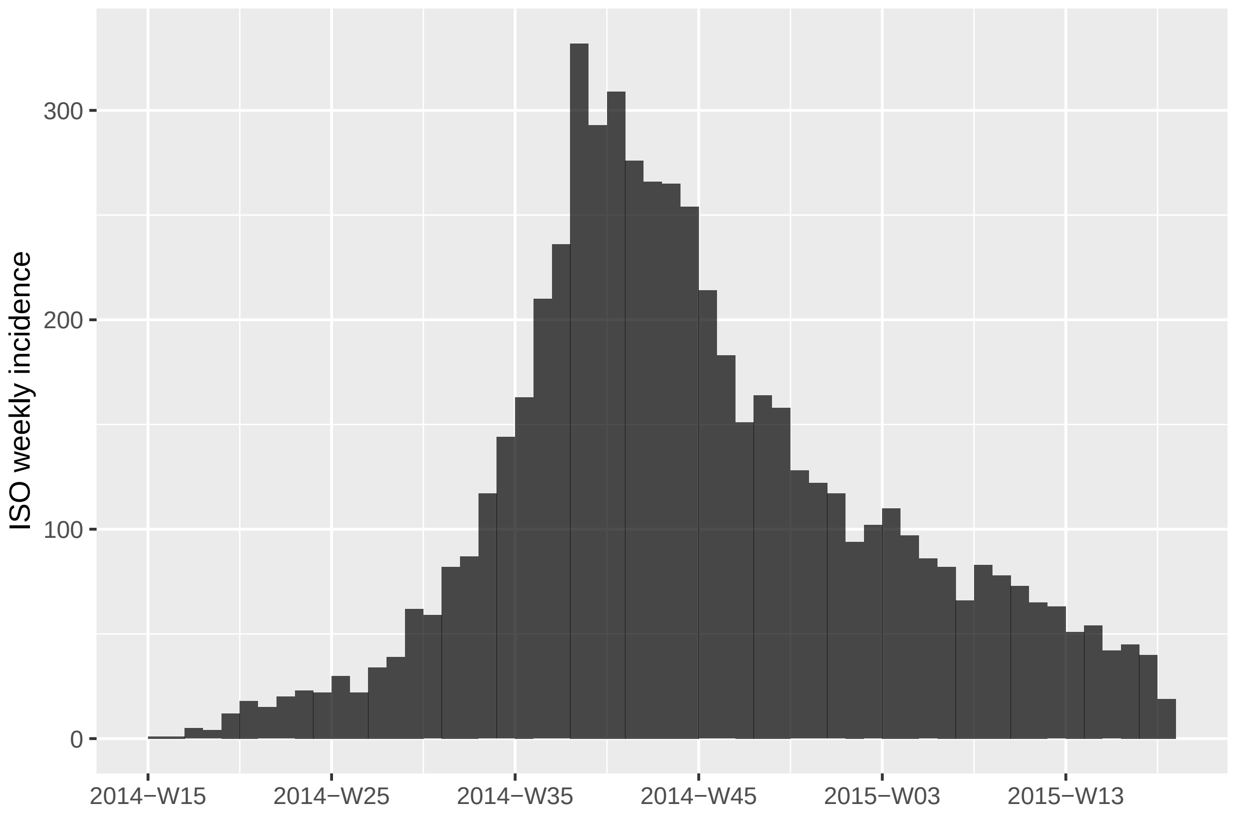
```
plot(ebola_incidence)
```

The *incidence* package includes a wrapper plot function. Run the code below to plot daily incidence of ebola from this simulated data set.



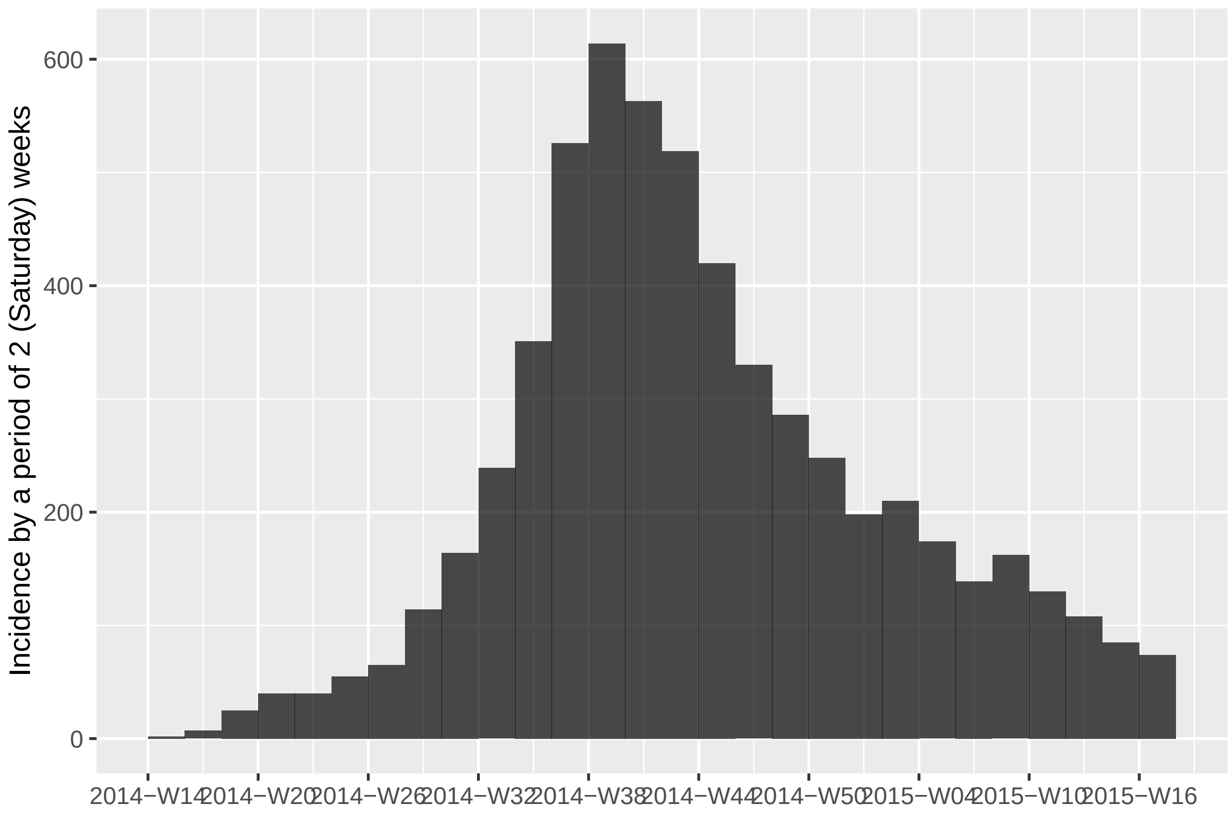
```
ebola_incidence_weekly <- incidence(dates = ebola_linelist$date_of_onset,
                                    interval = "1 week")
plot(ebola_incidence_weekly)
```

Daily incidence is quite noisy for this data set, so it can be advantageous to plot weekly incidence instead. The `incidence` function helps us make this change easily if we specify a different interval. Create a new object, `ebola_incidence_weekly` by modifying the code we used to create `ebola_incidence` - change `interval = 1` to `interval = "1 week"`. Then plot the weekly incidence. Do this in the code chunk below:



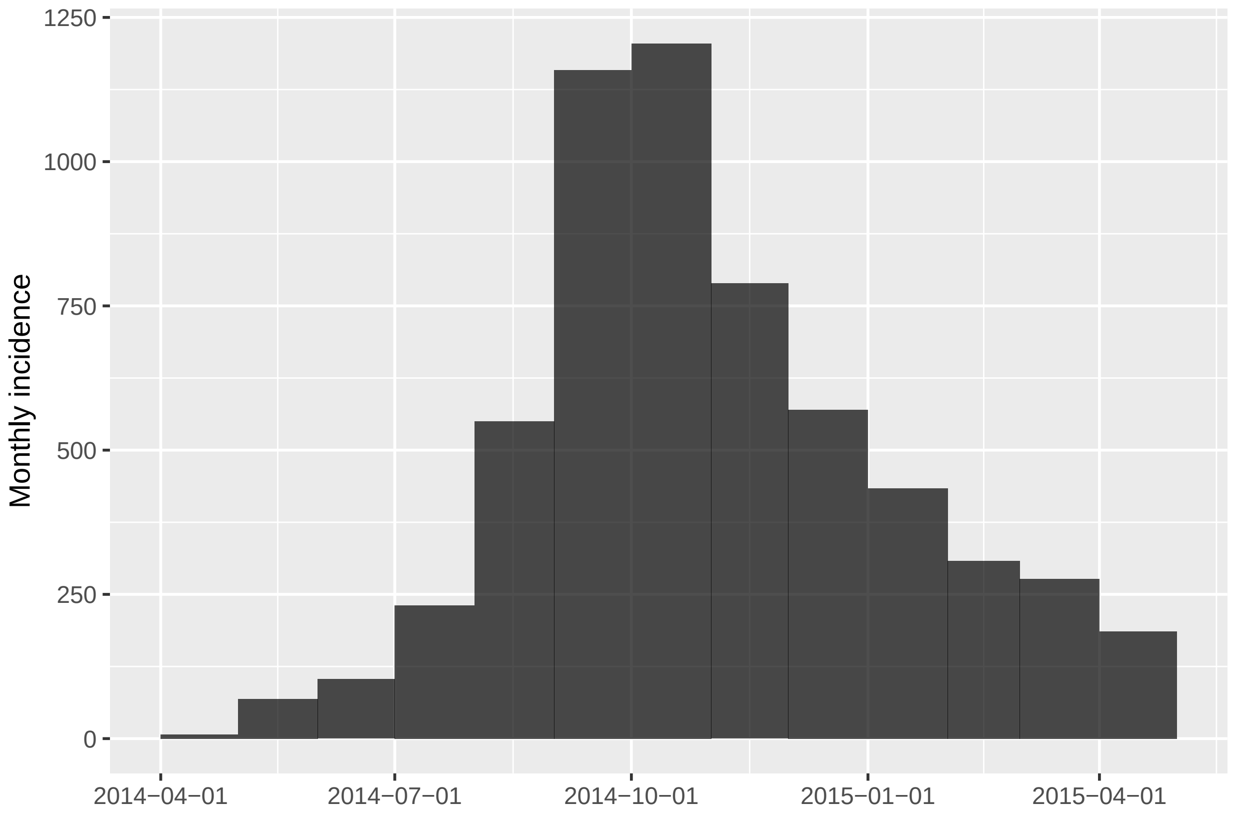
```
ebola_incidence_14 <- incidence(dates = ebola_linelist$date_of_onset,
                                interval = "2 saturday weeks")
plot(ebola_incidence_14)
```

We can do the same thing, but change the time scale to semi-weekly, starting on Saturday. Create a new object, `ebola_incidence_14` by modifying the code we used to create `ebola_incidence` - change `interval = 1` to `interval = "2 saturday weeks"`. Then plot the semi-weekly incidence. Do this in the code chunk below:



```
ebola_incidence_monthly <- incidence(dates = ebola_linelist$date_of_onset,
                                     interval = "1 month")
plot(ebola_incidence_monthly)
```

We can do the same thing, but change the time scale to semi-weekly, starting on Saturday. Create a new object, `ebola_incidence_monthly` by modifying the code we used to create `ebola_incidence`. Use the help function for incidence (`?incidence`) to determine how to specify an interval of a month. Then plot the semi-weekly incidence. Do this in the code chunk below:



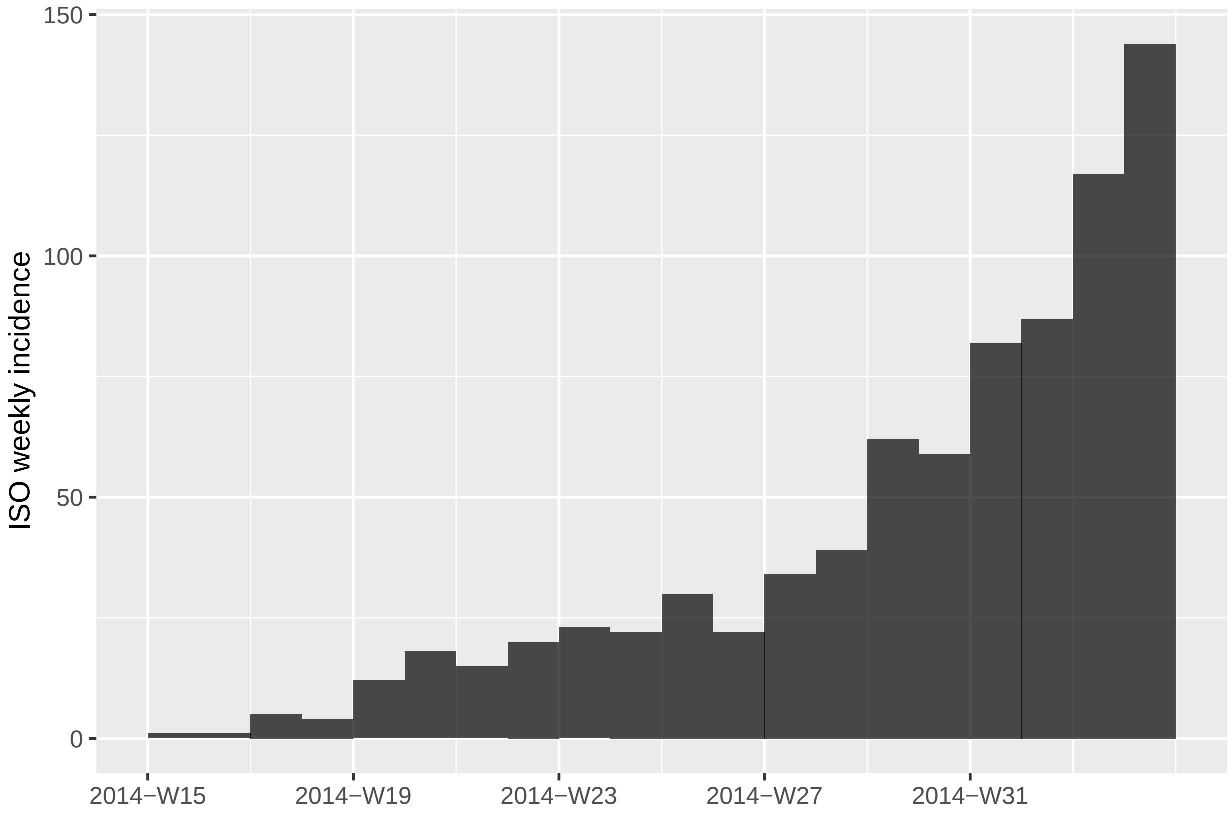
Modeling incidence via log-linear regression

Next week, we will spend some time formally discussing statistical models for incidence, but we will get an informal introduction here. Incidence data can be modeled using the following model:

$$\log(y) = b + r \times t$$

where y is the incidence (a count, here), r is the growth rate, and t is the number of days that have passed since a particular point in time. We can consider, informally, fitting a log-linear model to the beginning of this epidemic, when it is clearly growing - the first 20 weeks of the epidemic, and fitting a log linear model to these incidence counts:

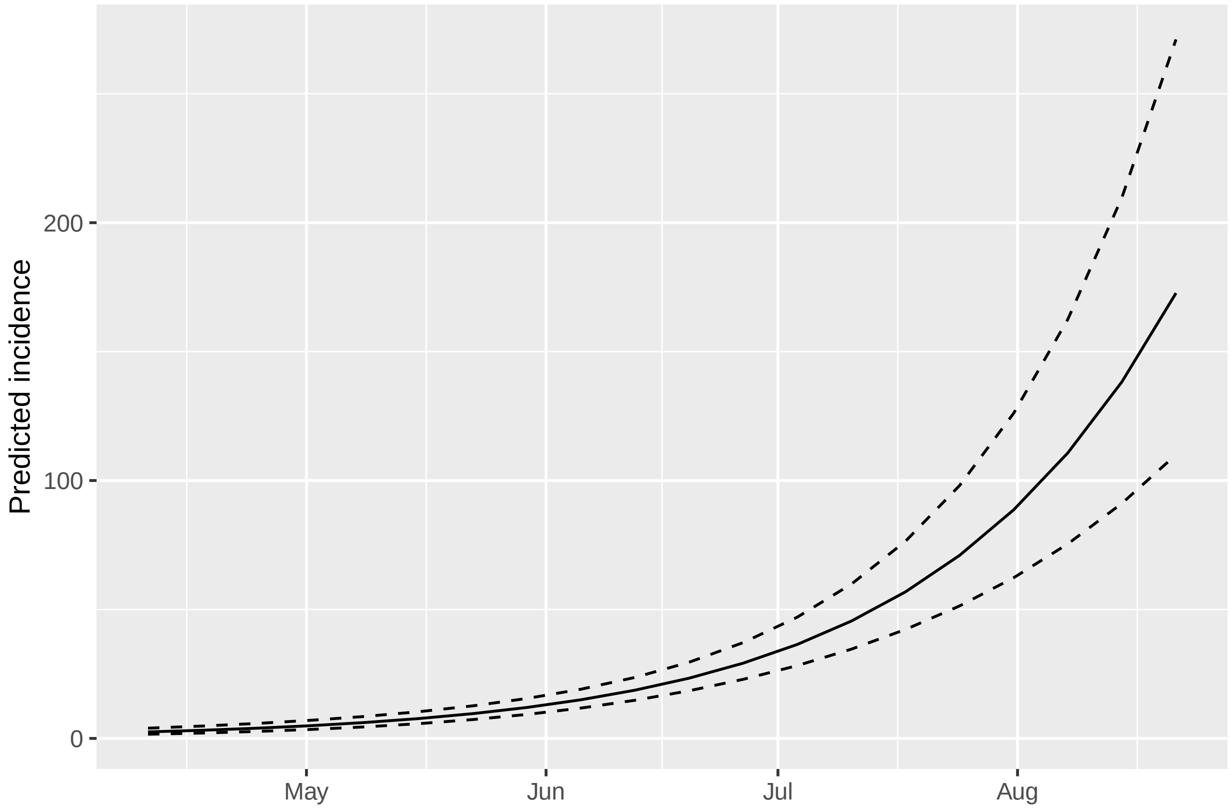
```
plot(ebola_incidence_weekly[1:20])
```



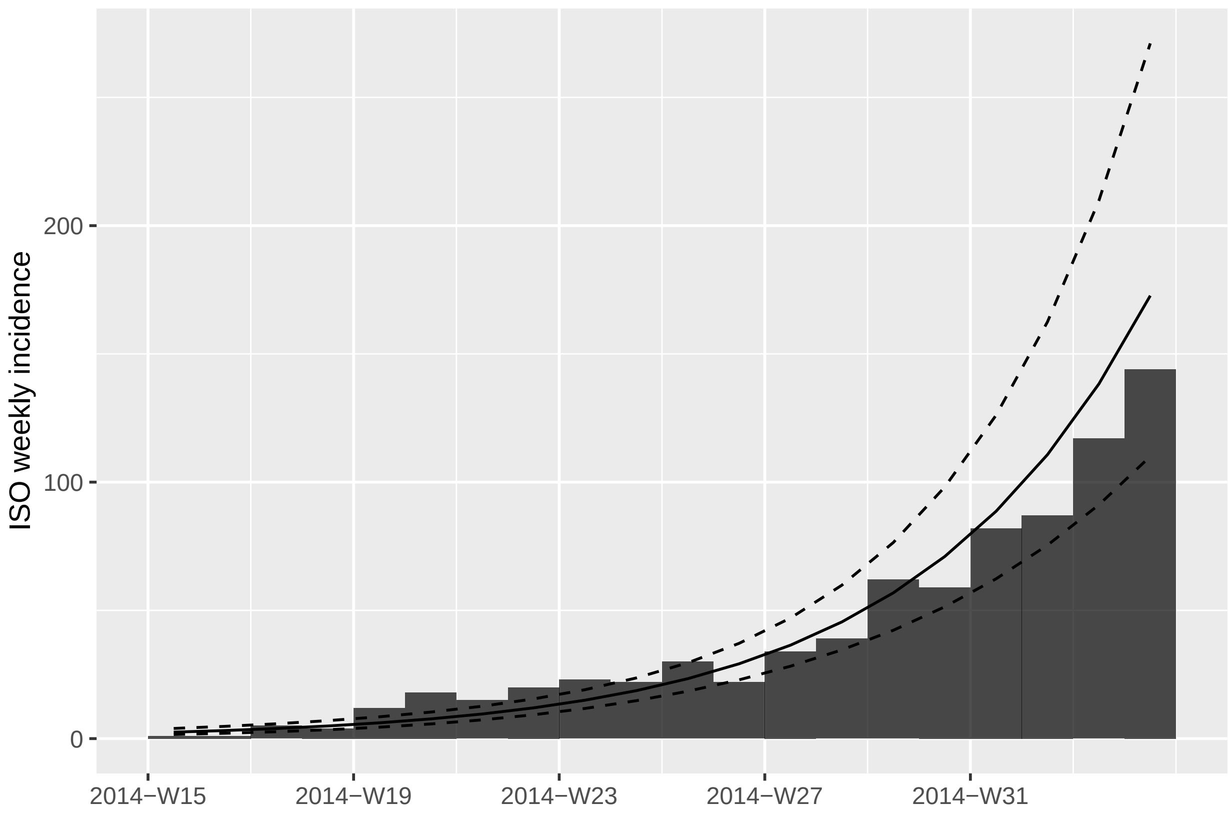
```
epidemic_first_20 <- fit(ebola_incidence_weekly[1:20])
epidemic_first_20
```

```
## <incidence_fit object>
##
## $model: regression of log-incidence over time
##
## $info: list containing the following items:
##   $r (daily growth rate):
## [1] 0.03175771
##
##   $r.conf (confidence interval):
##           2.5 %     97.5 %
## [1,] 0.02596229 0.03755314
##
##   $doubling (doubling time in days):
## [1] 21.8261
##
##   $doubling.conf (confidence interval):
##         2.5 %   97.5 %
## [1,] 18.45777 26.69823
##
##   $pred: data.frame of incidence predictions (20 rows, 5 columns)
```

```
plot(epidemic_first_20)
```



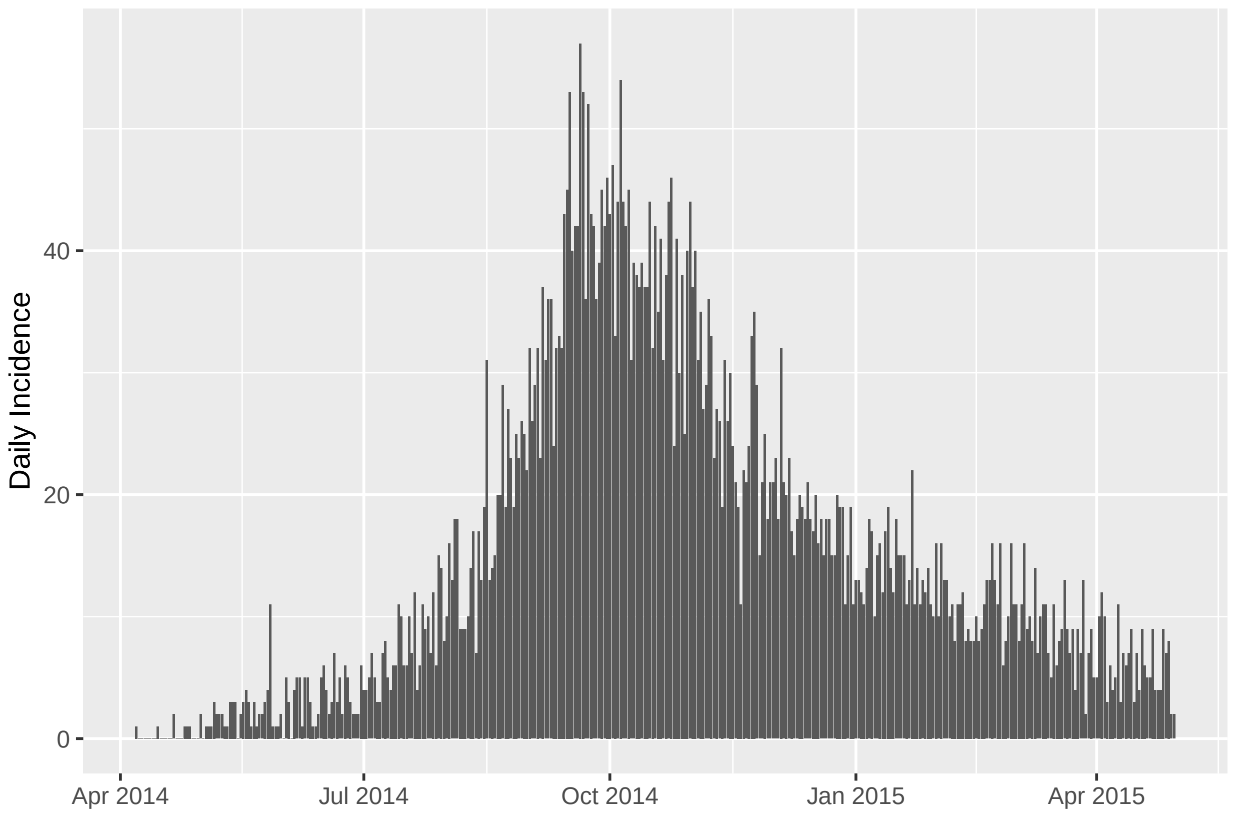
```
plot(ebola_incidence_weekly[1:20], fit=epidemic_first_20)
```



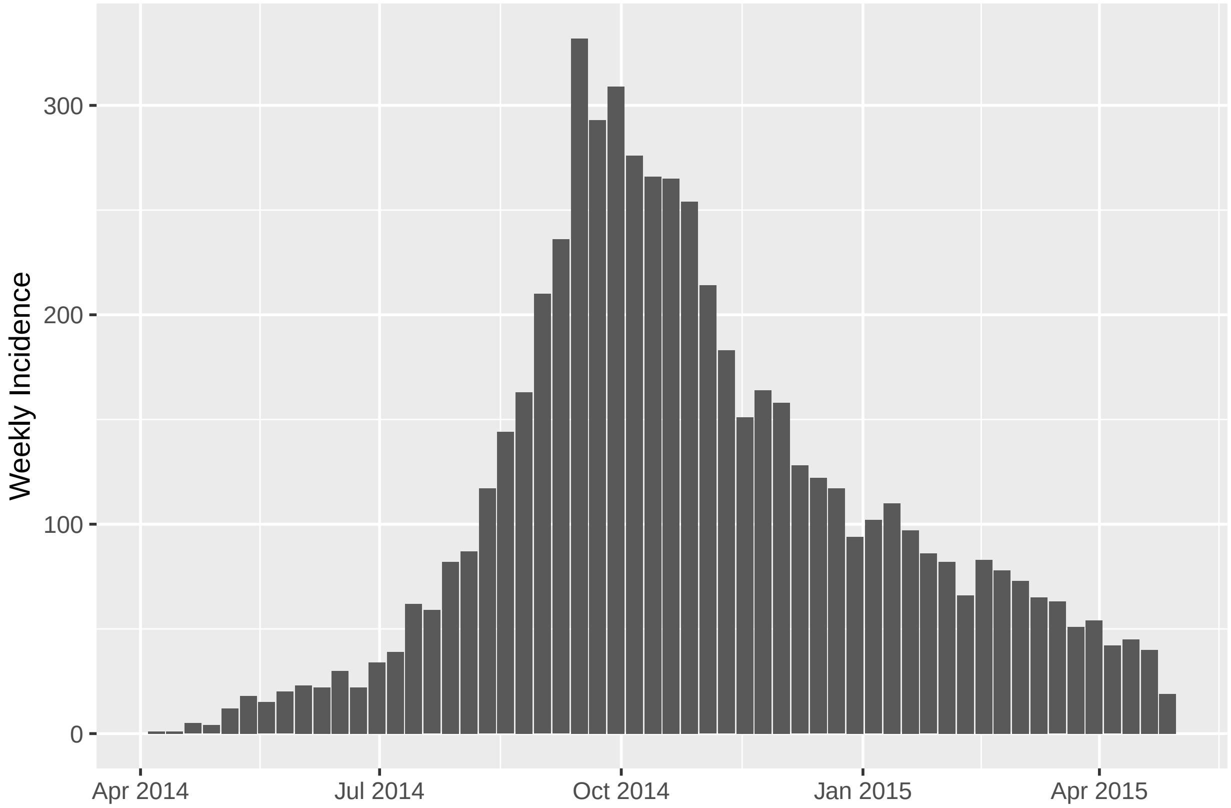
Challenge

All of the plots you made above can be created using ggplot2. If you have finished everything above, then try to make these plots using ggplot2, instead. There are a number of ways you can think about doing this.

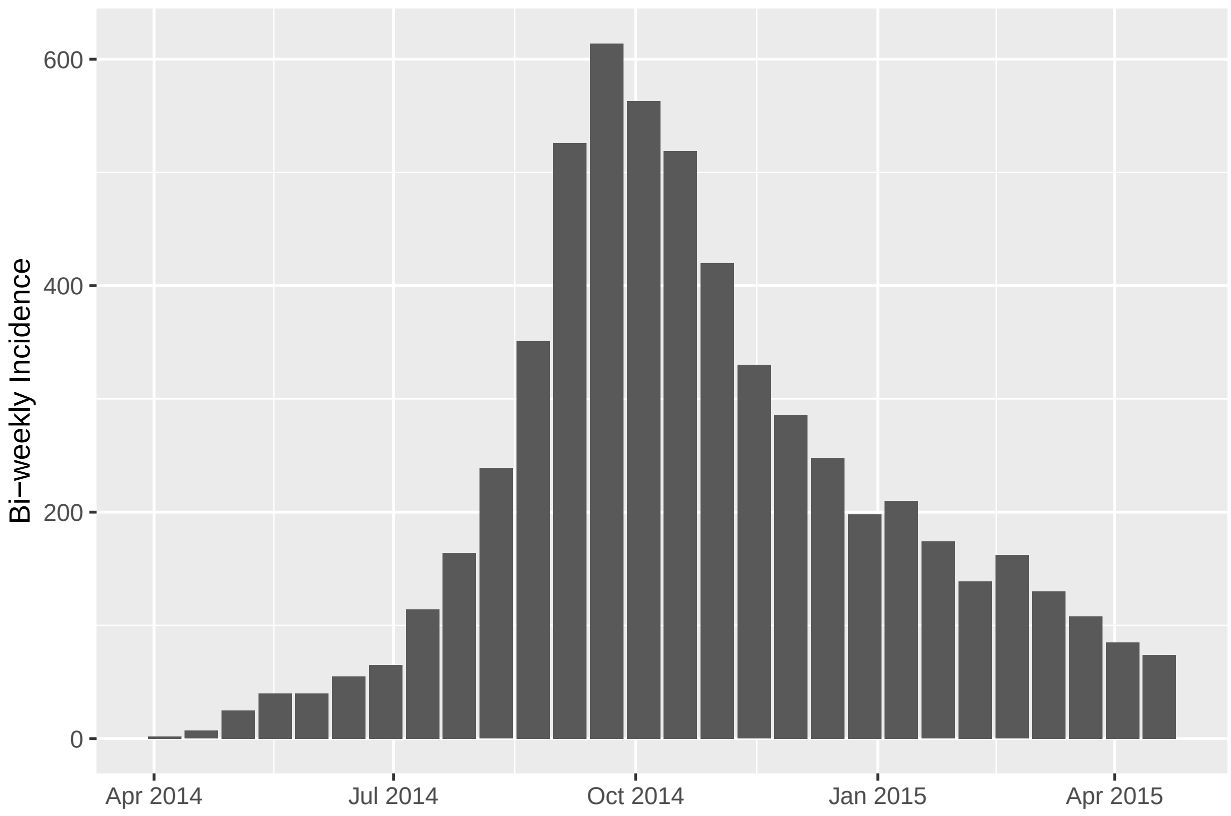
```
ggplot(data=as.data.frame(ebola_incidence), aes(x=dates, y=counts)) +
  geom_col() +
  xlab("") +
  ylab("Daily Incidence")
```



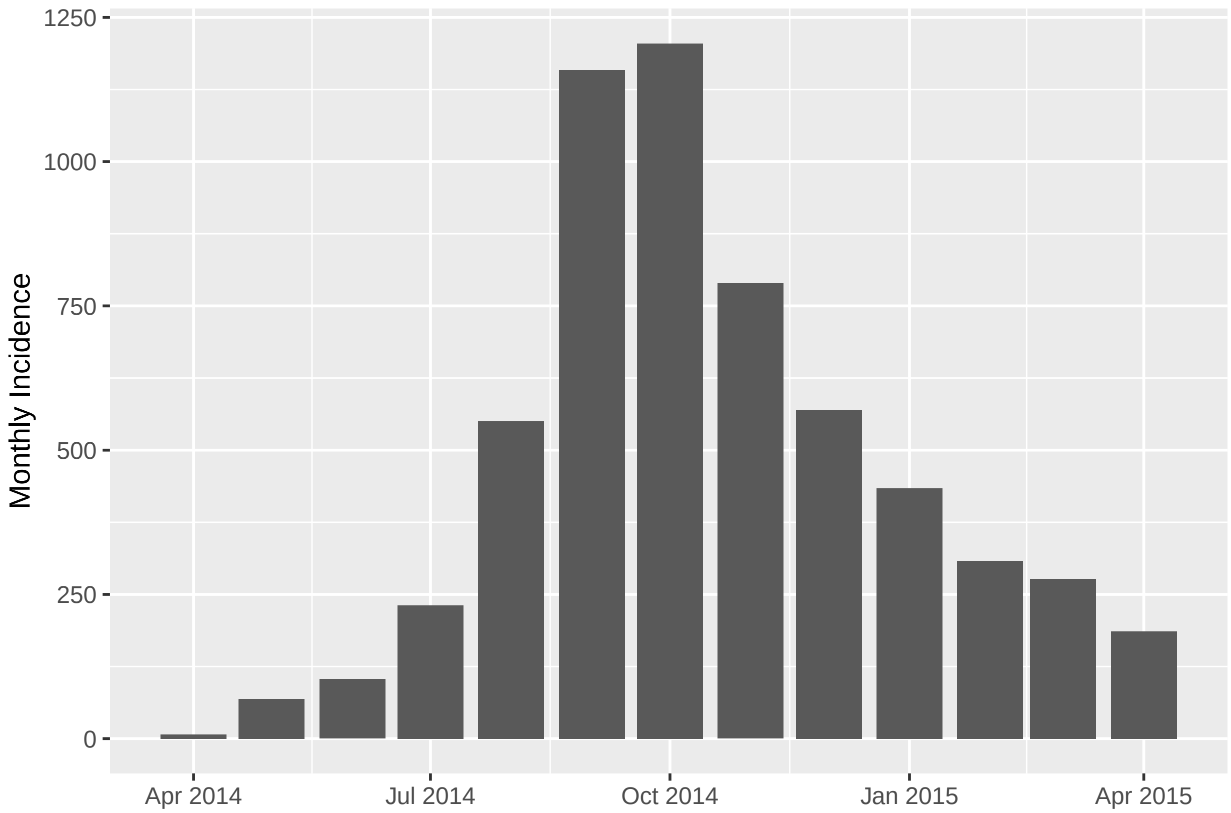
```
ggplot(data=as.data.frame(ebola_incidence_weekly), aes(x=dates, y=counts)) +
  geom_col() +
  xlab("") +
  ylab("Weekly Incidence")
```



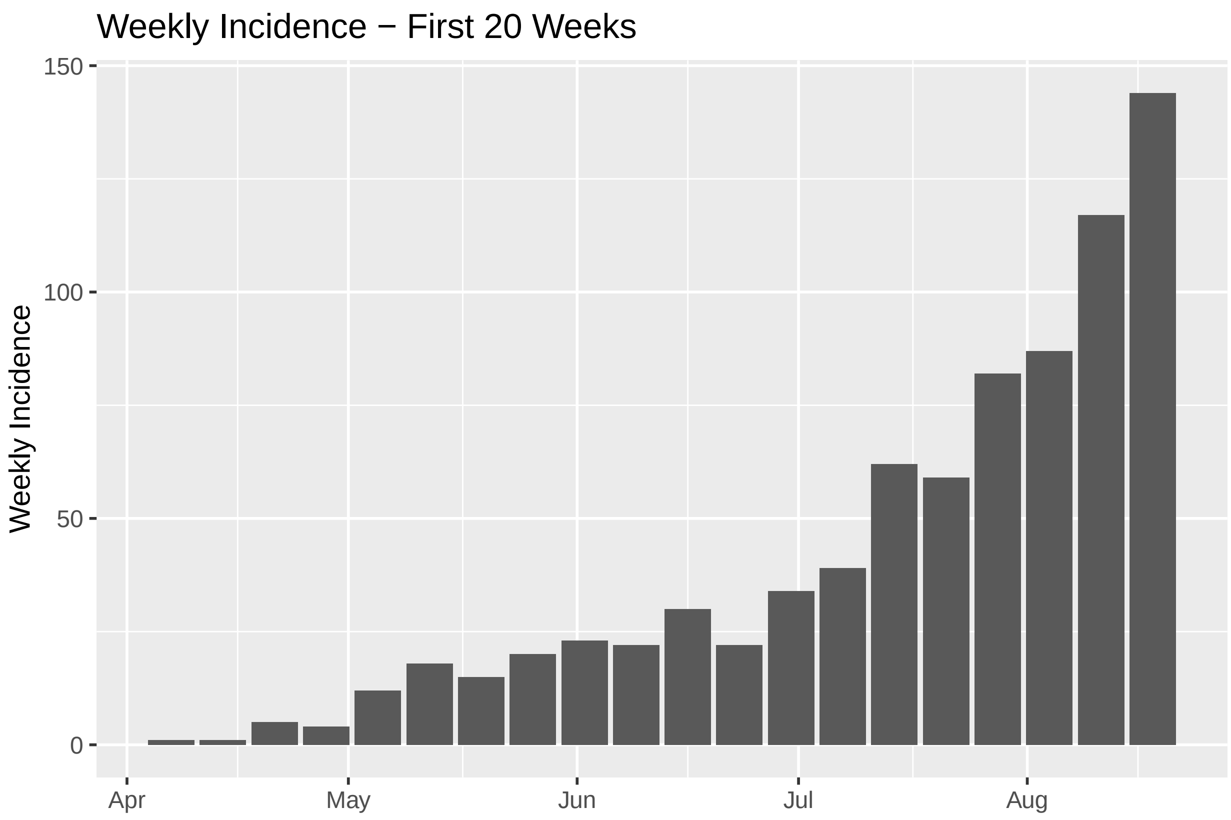
```
ggplot(data=as.data.frame(ebola_incidence_14), aes(x=dates, y=counts)) +
  geom_col() +
  xlab("") +
  ylab("Bi-weekly Incidence")
```



```r
ggplot(data=as.data.frame(ebola_incidence_monthly), aes(x=dates, y=counts)) +
  geom_col() +
  xlab("") +
  ylab("Monthly Incidence")
```

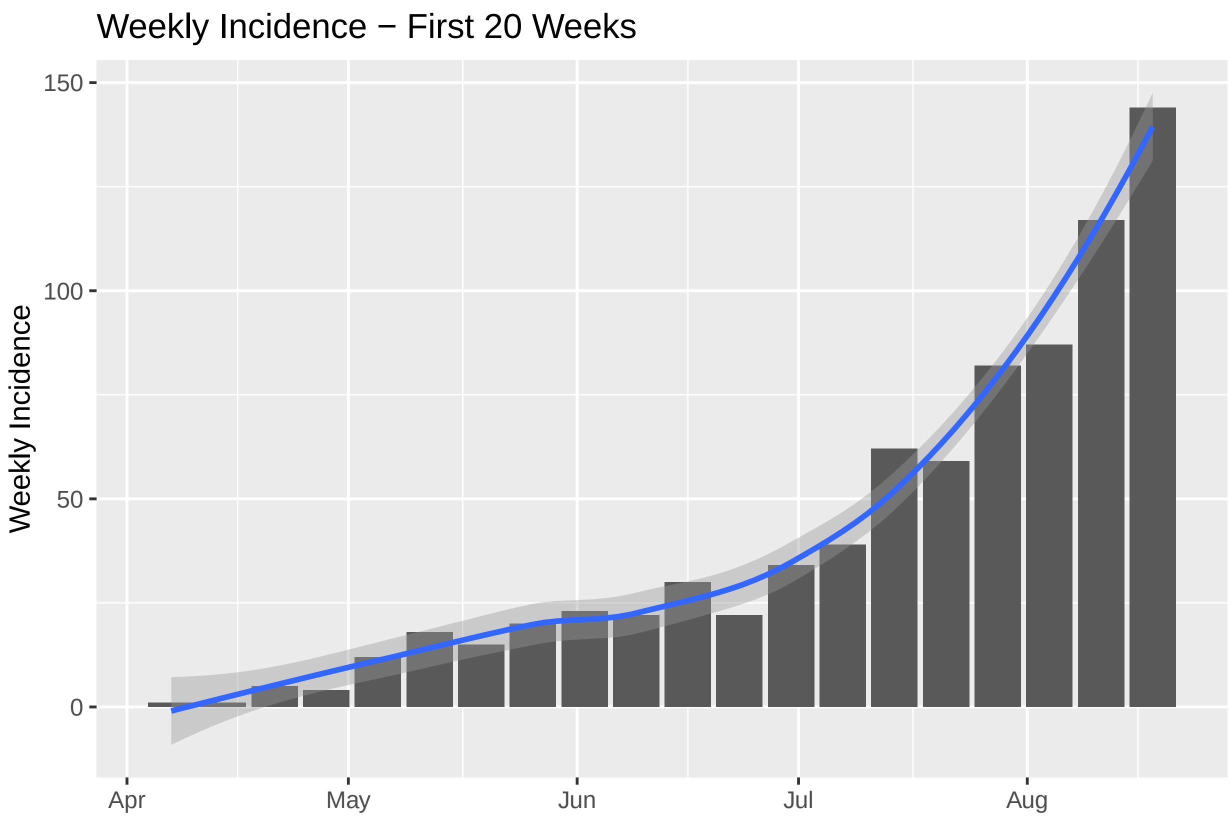


```
ggplot(data=as.data.frame(ebola_incidence_weekly[1:20]),
       aes(x=dates, y=counts)) +
  geom_col() +
  xlab("") +
  ylab("Weekly Incidence") +
  ggtitle("Weekly Incidence - First 20 Weeks")
```



```
ggplot(data=as.data.frame(ebola_incidence_weekly[1:20]),
       aes(x=dates, y=counts)) +
  geom_col() +
  xlab("") +
  ylab("Weekly Incidence") +
  ggtitle("Weekly Incidence - First 20 Weeks") +
  geom_smooth(method="loess")
```

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


Weekly Incidence − First 20 Weeks
Weekly Incidence
150
100
50
0
Apr
May
Jun
Jul
Aug