Data visualization

John M. Drake & Andrew W. Park

Introduction

This is the first in a series of five exercises that constitute *Training Module 1: Introduction to Scientific Programming*, taught through the IDEAS PhD program at the University of Georgia Odum School of Ecology in conjunction with the Center for the Ecology of Infectious Diseases. This module is based on the premise that computer coding is a basic scientific skill. This module introduces the principles and practice of scientific computing with special emphasis on analysis of infectious disease data. Programming will be done in R. Students will be taught how to create reproducible research documents using R and R Markdown and to use git/Github for collaborative and individual projects. An introduction to scientific programming will teach basic operations and classes of base R, installation and use of R packages, data import and transformation, flow control with loops, writing functions, calculating summary statistics, data visualization, and basic mapping.

Recommended reading for this module is the book *R for Data Science* by Hadley Wickham and Garret Grolemund (O'Reilly Media, 2017).

This exercise seeks to introduce the student to basic tasks and operations required to visualize data in R using `ggplot2`. *Data visualization* is a key component of *exploratory data analysis*. `ggplot2` implements Leland Wilkinson's idea that there is a "grammar of graphics" – that is an organizational scheme based on the semantic replationships among different graphical elements. Data visualization theory, practice, and exploratory data analysis are not covered systematically, but rather by example, with the expectation that students will develop further skills by extending the provided examples.

Case study

As a running example in this exercise, we will study a data set on the spread of Middle East Respiratory Syndrome Corona Virus (MERS-CoV) compiled and made available by Andrew Rambaut on his Github site (https://github.com/rambaut/MERS-Cases/blob/gh-pages/data/cases.csv). Github is a development platform used by developers to host a wide range of coding projects and is very popular with data scientists and others interested in open science. We will return to Github on the final day of the module. For now, we will just use it to access Rambaut's data. MERS-CoV is a positive-sense single-stranded Betacoronavirus. Its closest relatives are the SARS coronavirus, common-cold coronavirus, and other human betacoronaviruses. MERS-CoV first emerged in Suudi Arabia in 2012. It causes a severe respiratory illness Transmission to humans may be direct (person-to-person), particularly in hospitals, or from contact with infected animals. Exposure to camels is associated with many cases, although bats, particularly the Egyptian Tomb bat (*Taphozous perforatus*), are suspected to be the maintenance reservoir. The case fatality rate is around 40%.

Getting the data into R

To load the MERS data into an R session, do the following:

1. Create a working directory called `mers`
2. Download the file `cases.csv` and move it to `mers`
3. Open a session of R Studio
4. Set the working directory by typing `setwd('~/./mers)` where `.` is the file path to your working directory. (Alternatively, you can navigate by using the `Session` drop down menu and selecting `Set Working Directory`.)
5. Create an R *dataframe* by typing `data <- read.csv('cases.csv')` as shown below.

```r
mers <- read.csv('cases.csv')
```

Formatting some dates

We can inspect the data using the base R function `head`. We see that some variables, such as `onset` and `hospitalized` are dates, but formatted as a `factor`.

```r
head(mers)
```

```
##   number FT KSA_case code gender age country province  city district
## 1      1  2           25M      M  25  Jordan          Zarqa
## 2      2              30M      M  30  Jordan          Zarqa
## 3      3  1           40F      F  40  Jordan          Zarqa
## 4      4              60M      M  60  Jordan          Zarqa
## 5      5              29M      M  29  Jordan          Zarqa
## 6      6              33M      M  33  Jordan          Zarqa
##   prior_travel hospital exposure      onset hospitalized sampled reported
## 1                                2012-03-21   2012-04-04
## 2                                2012-03-30   2012-04-08
## 3                                2012-04-02   2012-04-09
## 4                                2012-04-02
## 5                                2012-04-11   2012-04-15
## 6                                2012-04-12   2012-04-14
##        death discharged comorbidity severity outcome    clinical
## 1 2012-04-25                           fatal   fatal       fatal
## 2                                        CCU            clinical
## 3 2012-04-19                           fatal   fatal       fatal
## 4                                                    subclinical
## 5                                        CCU            clinical
## 6                                        CCU            clinical
##   old_cluster cluster Cauchemez.cluster animal_contact camel_contact   HCW
## 1           A       A                4          FALSE               FALSE
## 2           A       A                4          FALSE                TRUE
## 3           A       A                4          FALSE                TRUE
## 4           A       A                4          FALSE                TRUE
## 5           A       A                4                               TRUE
## 6           A       A                4                               TRUE
##   contact_with            contact secondary suspected inferred    notes
## 1                                                           NA
## 2            1 health care worker      TRUE      TRUE       NA probable
## 3            1 health care worker      TRUE                 NA
## 4            1 health care worker      TRUE      TRUE       NA probable
## 5              health care worker      TRUE      TRUE       NA probable
## 6            1 health care worker      TRUE      TRUE       NA probable
##                                                                  citation
## 1 http://applications.emro.who.int/emhj/v19/Supp1/EMHJ_2013_19_Supp1_S12_S18.pdf
## 2 http://applications.emro.who.int/emhj/v19/Supp1/EMHJ_2013_19_Supp1_S12_S18.pdf
## 3 http://applications.emro.who.int/emhj/v19/Supp1/EMHJ_2013_19_Supp1_S12_S18.pdf
## 4 http://applications.emro.who.int/emhj/v19/Supp1/EMHJ_2013_19_Supp1_S12_S18.pdf
## 5 http://applications.emro.who.int/emhj/v19/Supp1/EMHJ_2013_19_Supp1_S12_S18.pdf
## 6 http://applications.emro.who.int/emhj/v19/Supp1/EMHJ_2013_19_Supp1_S12_S18.pdf
##   citation2 citation3 citation4 citation5       sequence accession patient
## 1                                                                        1
## 2                                                                            2
## 3                                          Jordan-N3_2012  KC776174          3
## 4                                                                            4
## 5                                                                            5
## 6                                                                            6
##   speculation  X                                          X.1
## 1            NA http://promedmail.org/direct.php?id=3587349
## 2            NA
## 3            NA
## 4            NA
## 5            NA
## 6            NA
```

```r
class(mers$onset)
```

```
## [1] "factor"
```

These dates can be reformatted using the `lubridate` package. Here we create new variables using the `Date` class. But, first we correct a few errors.

```r
mers$hospitalized[890] <- c('2015-02-20')
mers <- mers[-471,]

library(lubridate)
mers$onset2 <- ymd(mers$onset)
mers$hospitalized2 <- ymd(mers$hospitalized)
class(mers$onset2)
```

```
## [1] "Date"
```

We may also find it useful to have a simple numerical value for the days elapsed since the start of the epidemic. We use the following code to search for the earliest onset date.

```r
day0 <- min(na.omit(mers$onset2))
```

Question. Why do we use the function `na.omit`? What happens if we neglect this command?

Now we can create a new, numeric value for the "epidemic day" for each case.

```r
mers$epi.day <- as.numeric(mers$onset2 - day0)
```

Question. What purpose does the command `as.numeric` serve?

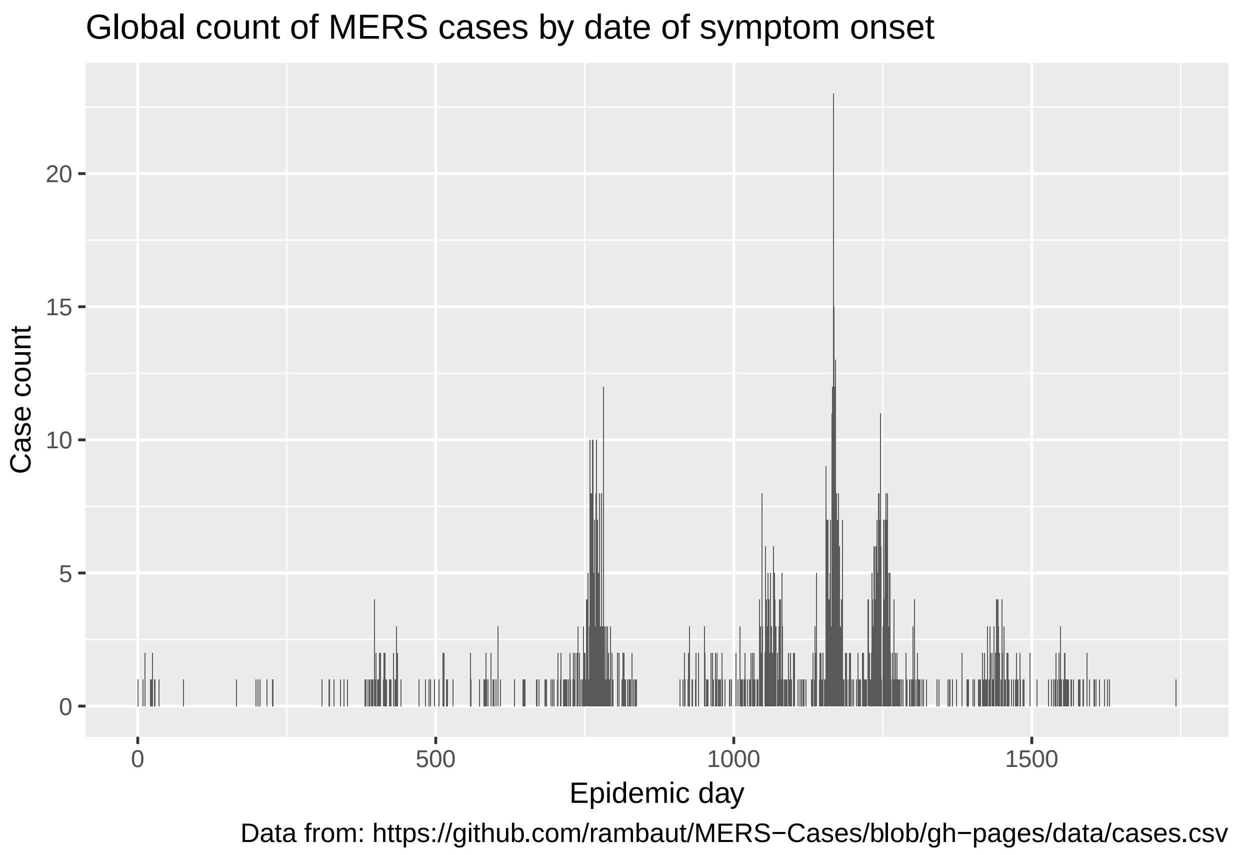
Making a plot

Next, we load `ggplot2`.

```r
library(ggplot2)
```

We can explore some of the MERS data using the function `ggplot`. One plot we might wish to produce is the *epidemic curve* which is basically a bar plot. An empty plot can be produced using the command `ggplot(data=mers)`. The epidemic curve is then produced by adding a bar plot using the geom function `geom_bar`. The last line of our code adds some labels.

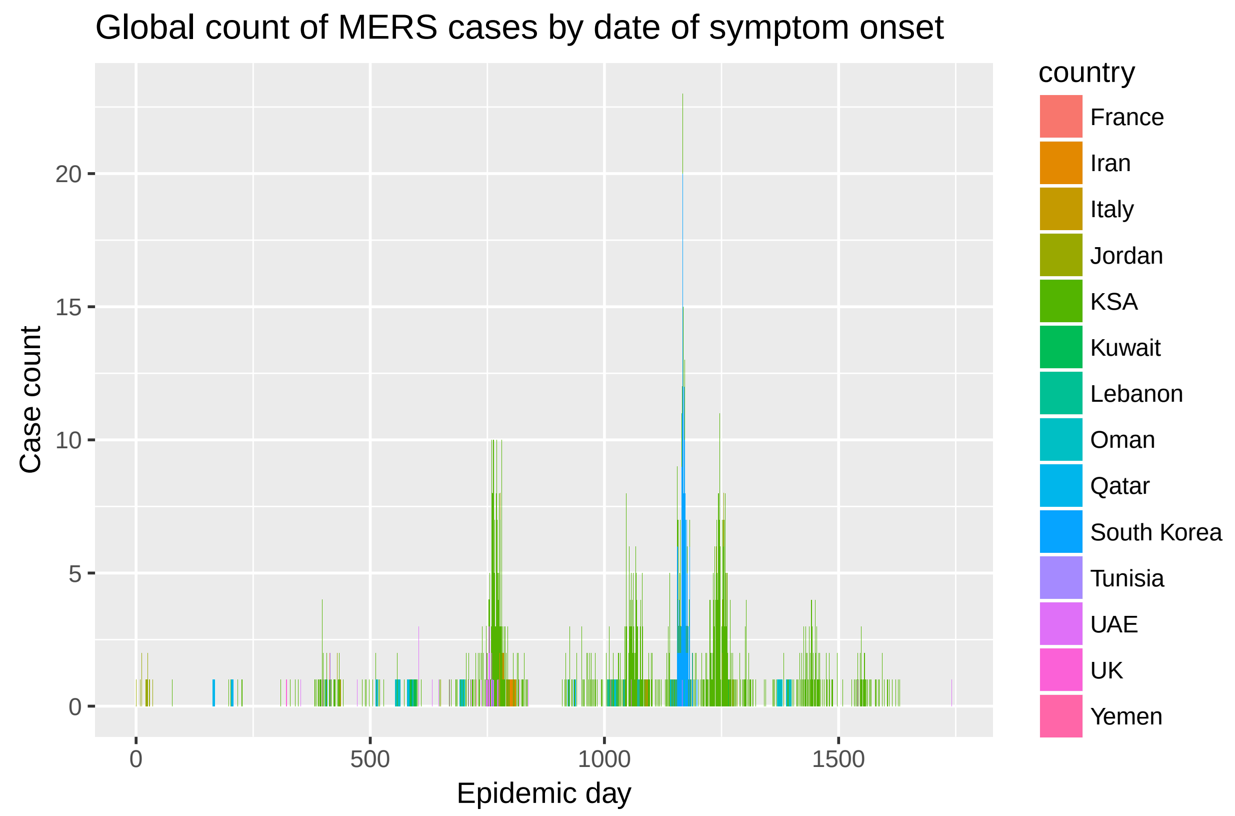
```r
ggplot(data=mers) +
  geom_bar(mapping=aes(x=epi.day)) +
  labs(x='Epidemic day', y='Case count', title='Global count of MERS cases by date of symptom onset',
       caption="Data from: https://github.com/rambaut/MERS-Cases/blob/gh-pages/data/cases.csv")
```



Exercise. To produce this plot, type all the commands up to this point *exactly* as they appear. Particularly, note that as we "build" the plot in the last code snippet, we end each line with the addition syhmbol "+". What happens if we don't use this convention?

Of course, all these cases are distributed among a number of different countries. We can modify the plot to show this using the using the aesthetic `fill`.

```r
ggplot(data=mers) +
  geom_bar(mapping=aes(x=epi.day, fill=country)) +
  labs(x='Epidemic day', y='Case count', title='Global count of MERS cases by date of symptom onset',
       caption="Data from: https://github.com/rambaut/MERS-Cases/blob/gh-pages/data/cases.csv")
```



Data from: https://github.com/rambaut/MERS−Cases/blob/gh−pages/data/cases.csv

In this example, we've shown how to make a basic bar plot with `ggplot` and the geom function `geom_bar`. Our mapping of data objects to plot objects was performed using `aes` and we used the `fill` aesthetic to examing the distribution of cases among countries. There are lots of variations on the bar plot that we can examine. For instance, we can modify the `position`, which is another argument to `geom_bar`.

Exercise. Modify the epidemic curve using the argument `position=fill`. What does this plot show?

Exercise. Another way to modify a bar plot is to change the coordinates. This can be done by "adding" `coord_flip()` and `coord_polar()` to the plot. Modify the epidemic curve using the argument `position=fill`. What does this plot show?

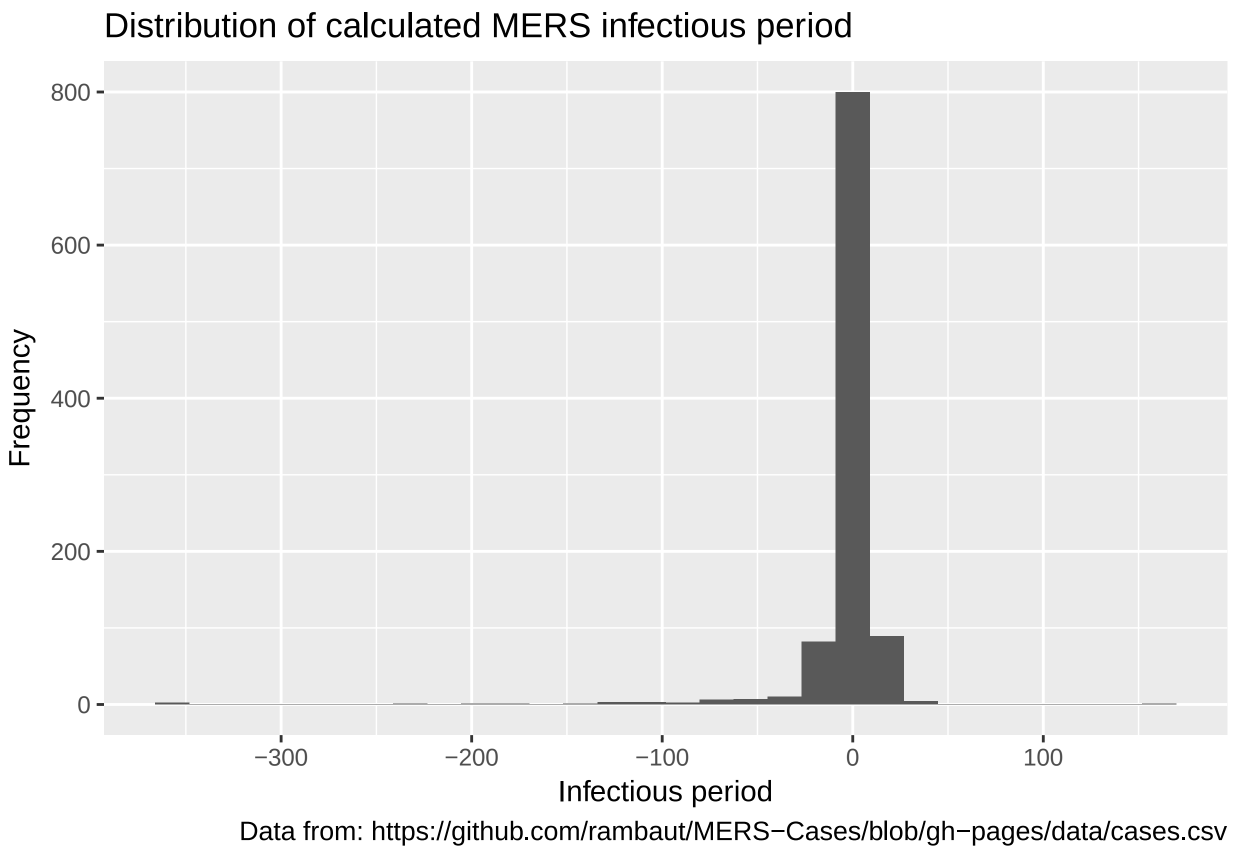
Univariate plots

Of course, there are lots of plot types other than bar plots. A quick reference for some of the more common plot types is the *ggplot2 Cheat Sheet*, available online at https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf. To explore some of these plot types, we will first construct a continuous quantity that is often of interest, the *infectious period*. From the standpoint of disease transmission, the infectious period is best defined as the duration of infectiousness for a patient. From an epidemiological point of view, this may often be approximated as the time between the onset of symptoms and the time of death, hospitalization, or isolation. Here we caculate the infectious period and plot a histogram.

```
mers$infectious.period <- mers$hospitalized2-mers$onset2    # calculate "raw" infectious period
class(mers$infectious.period)          # these data are class "difftime"
```

```
## [1] "difftime"
```

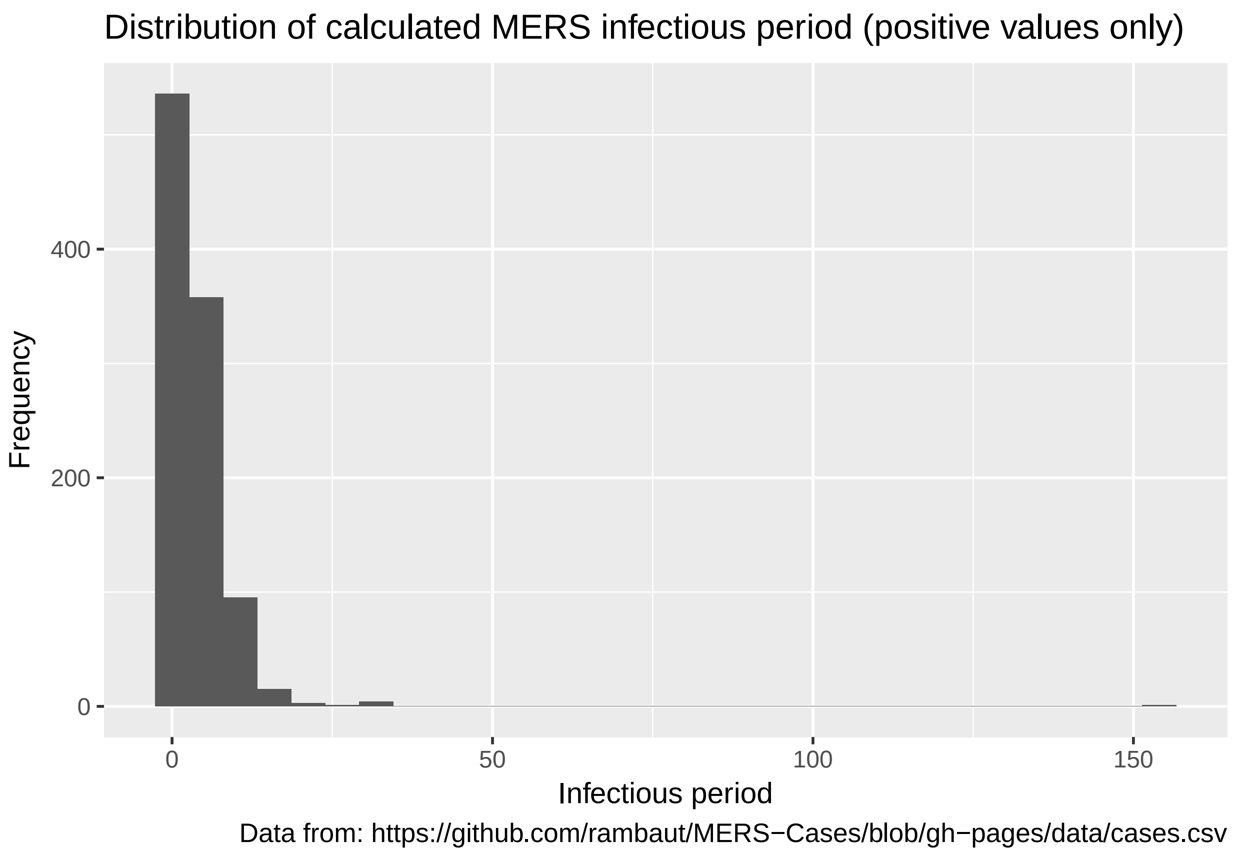
```r
mers$infectious.period <- as.numeric(mers$infectious.period, units = "days") # convert to days
ggplot(data=mers) +
  geom_histogram(aes(x=infectious.period)) +
  labs(x='Infectious period', y='Frequency', title='Distribution of calculated MERS infectious period',
       caption="Data from: https://github.com/rambaut/MERS-Cases/blob/gh-pages/data/cases.csv")
```



Wait a minute! What is a negative infectious period?

In the case of MERS, this epidemiological definition of infectious period is misleading, because in some cases the main source of transmission has been nosocomial (infections in a health care setting). This appears in our data as a negative time interval between onset and hospitalization. Perhaps we would wish to calculate a *new* value, which is the calculated infectious period in the case where it is positive and zero otherwise. To do this, we rely on the handy function `ifelse`.

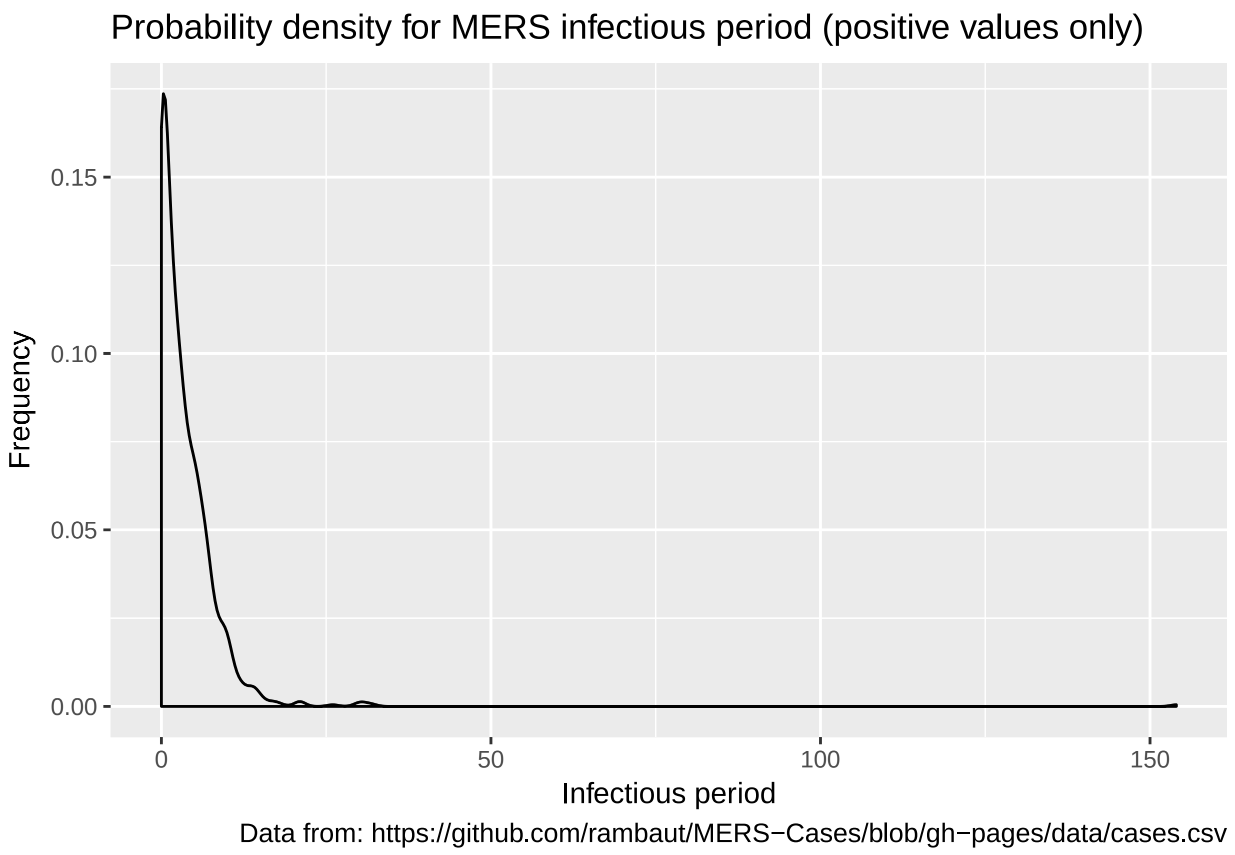
```r
mers$infectious.period2 <- ifelse(mers$infectious.period<0,0,mers$infectious.period)
ggplot(data=mers) +
  geom_histogram(aes(x=infectious.period2)) +
  labs(x='Infectious period', y='Frequency',
       title='Distribution of calculated MERS infectious period (positive values only)', caption="Data f
```



Exercise. Investigate the frequency of hospital-acquired infections of MERS

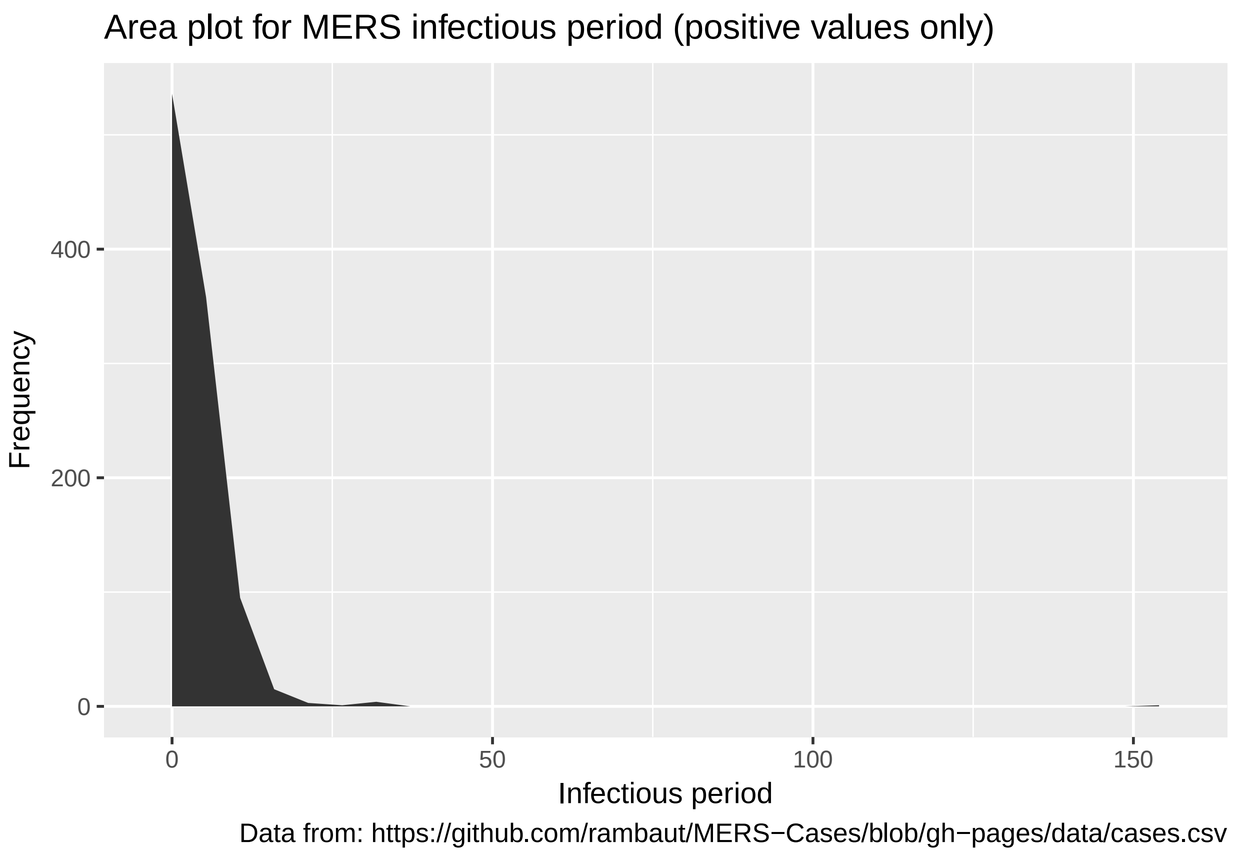
There are lots of different plot types that one can use to inspect continuously valued or integer-valued data like these. For instance, the density plot

```
ggplot(data=mers) +
  geom_density(mapping=aes(x=infectious.period2)) +
  labs(x='Infectious period', y='Frequency',
       title='Probability density for MERS infectious period (positive values only)', caption="Data fro
```



Or the area plot

```
ggplot(data=mers) +
  geom_area(stat='bin', mapping=aes(x=infectious.period2)) +
  labs(x='Infectious period', y='Frequency',
       title='Area plot for MERS infectious period (positive values only)', caption="Data from: https:/
```



Exercise. Use the infectious period data calculated in `mers$infectious.period2` to experiment with other univariate plot types like `geom_dotplot` and `geom_bar`.

Bivariate plots

The preceding plots have all concerned the distribution of one variable. Of course, the objective of data analysis is usually to determine the *relationships* among variables. In this section we study some plots for two variables.

We can continue our study by focusing on the infectious period. Epidemiological theory holds that an outbreak will continue to grow in size as long as the *effective reproduction number* is greater than one. Roughly, the effective reproduction number is the ration of the number of transmission events per unit time to the infectious period. The number of transmission events, in turn, is given by the product of: (1) the per encounter probability of transmissibility, (2) the contact rate in the population, and (3) the fraction of the population that is susceptible to the disease.

This means outbreaks can end two ways. First, the outbreak might “burn out” by infecting such a large fraction of the susceptible population that the effective reproduction number falls below one. Second, outbreaks are often curtailed before this point by isolating infected individuals, effectively by reducing the infectious period. Thus, we might be interested in understanding how the infectious period changes over time (i.e., the effectiveness of isolation).

Exercise. Use our corrected infectious period variable (`infectious.period2`) to study the change in the inectious period over the course of the MERS epidemic.

Exercise. In data from many outbreaks it can be seen that there is a kind of *societal learning*. When the infection first emerges it is not quickly recognized, public health resources have not been mobilized, it is not known what symptoms are diagnostic, how to treat, etc. But, quickly,

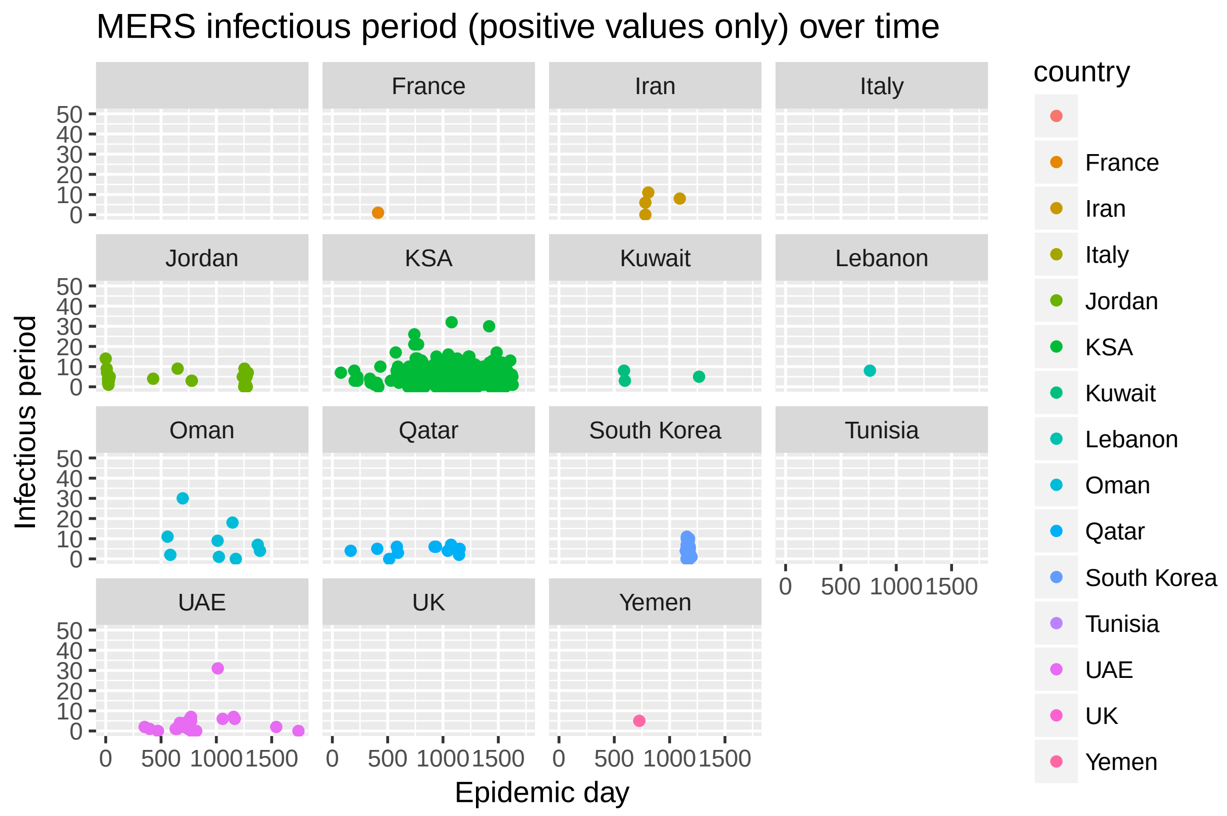
this information is collected and the utbreak is contained. Is there evidence of this kind of societal learning in the mers data. Add a curve fit using `geom_smooth` to explore this question. Hint: I solved using the `loess` method because the default smoother (gam) failed.

Exercise. Plot `infectious.period2` against time, as before, but this time add a separate smooth fit for each country.

Faceting

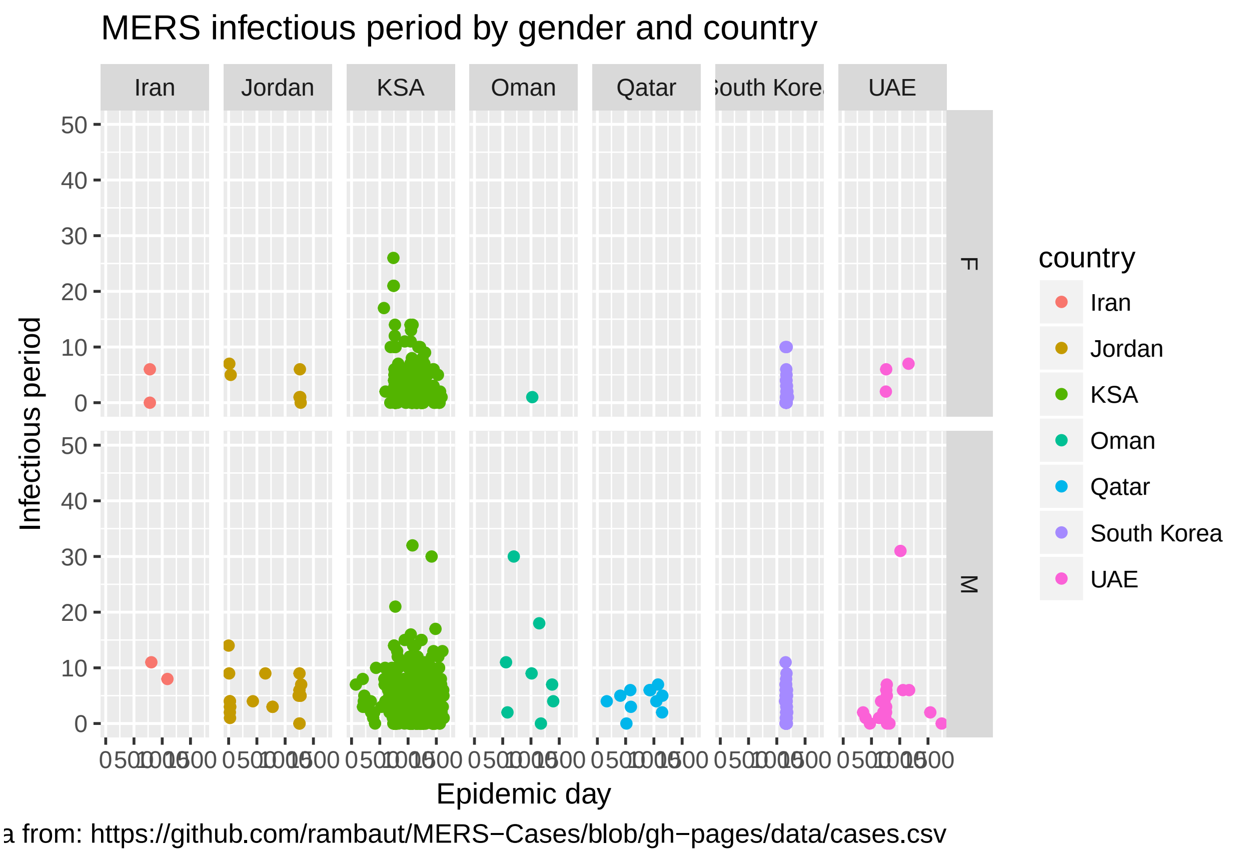
Ordinary plots are great when we want to compare two variables. Furthermore, we can study three or more variables by varying other features of the aesthetics (i.e., color). But, as we saw in the last exercise, when we begin adding information to our plot it can quickly get cluttered. There are numerous ways to add information from additional variables, for instance 3-d plots and contour plots. Another way is to create *multi-panel* plots. In ggplot2, this is called *faceting*. Faceting allows one to look at subsets of a data set simultaneously. In ggplot, this is accomplished using the functions `facet_wrap()` and `facet_grid`. The behavior of these functions is shown below. Notice how the second example uses `subset` to exclude countries that didn't report many cases and unusual codings for gender (e.g. `?M`).

```
ggplot(data=mers, mapping=aes(x=epi.day, y=infectious.period2)) +
  geom_point(mapping = aes(color=country)) +
  facet_wrap(~ country) +
  scale_y_continuous(limits = c(0, 50)) +
   labs(x='Epidemic day', y='Infectious period',
       title='MERS infectious period (positive values only) over time', caption="Data from: https://git
```



Data from: https://github.com/rambaut/MERS−Cases/blob/gh−pages/data/cases.csv

```
ggplot(data=subset(mers, gender %in% c('M', 'F') & country %in% c('KSA', 'Oman', 'Iran', 'Jordan', 'Qata
  geom_point(mapping = aes(color=country)) +
  facet_grid(gender ~ country) +
  scale_y_continuous(limits = c(0, 50)) +
   labs(x='Epidemic day', y='Infectious period',
       title='MERS infectious period by gender and country', caption="Data from: https://github.com/raml
```



Exercise. Study variation in the *case fatality rate* (the fraction of cases that end in death) over time and across countries.

More

The `ggplot2` package provides over 30 geoms and extension packages (e.g. `ggplot2-exts`) provide many others.

Exercise. Download and install the ggplot extension. Modify your plot with one or more new geoms.

Interactive graphics are graphics rendered in HTML that allow the viewer to extract more information or modify the information presented using selection, hovering, etc. The `plotly` package allows the graphics produced using the vast majority of `ggplot` functions to be used interactively using the following sequence:

1. Create a ggplot
2. Assign this plot to a variable
3. call `ggplotly` from te `plotly` package using the ggplot as the argument.

The following code demonstrates using the very first graph produced in this exercise, an epidemic curve created using the barplot geom.

```r
library(plotly)
epi.curve <- ggplot(data=mers) +
  geom_bar(mapping=aes(x=epi.day)) +
  labs(x='Epidemic day', y='Case count', title='Global count of MERS cases by date of symptom onset',
       caption="Data from: https://github.com/rambaut/MERS-Cases/blob/gh-pages/data/cases.csv")
ggplotly(epi.curve)
```

Exercise. Make one of your case fatality plots an interactive graphic.