The original editor in mapedit provided by leaflet.extras, is a  
wonderful tool but struggles with snapping and those pesky holes that we  
commonly face in geospatial tasks. Depending on the task, a user might  
prefer to continue using Leaflet.draw, so we will maintain full  
support for both editors. We’ll spend the rest of the post demonstrating  
where Leaflet.pm excels to help illustrate when you might want to  
choose editor = "leafpm".

**Install/Update**

At a minimum, to follow along with the rest of this post, please update  
mapedit and install the new standalone package leafpm. While we are  
it, we highly recommend updating your other geospatial dependencies.

install.packages(c("sf", "leaflet", "leafpm", "mapview", "mapedit"))

# lwgeom is optional but nice when working with holes in leaflet.pm

install.packages("lwgeom")

**Holes**

mapedit now supports holes. Let’s look at a quick example in which we  
add, edit, and delete holes.

library(sf)

library(leaflet)

library(mapview)

library(mapedit)

library(leafpm)

# make a contrived polygon with holes for testing

outer1 = matrix(c(0,0,10,0,10,10,0,10,0,0),ncol=2, byrow=TRUE)

hole1 = matrix(c(1,1,1,2,2,2,2,1,1,1),ncol=2, byrow=TRUE)

hole2 = matrix(c(5,5,5,6,6,6,6,5,5,5),ncol=2, byrow=TRUE)

outer2 = matrix(c(11,0,11,1,12,1,12,0,11,0),ncol=2, byrow=TRUE)

pts1 = list(outer1, hole1, hole2)

pts2 = list(outer2)

pl1 = st\_sf(geom = st\_sfc(st\_polygon(pts1)))

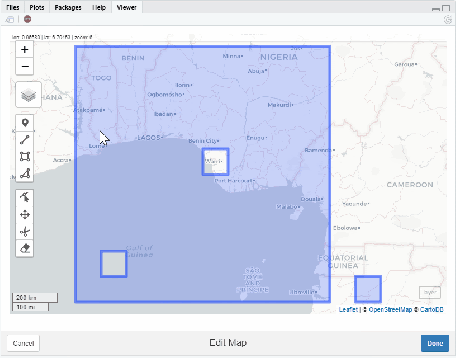
pl2 = st\_sf(geom = st\_sfc(st\_polygon(pts2)))

mpl = st\_sf(geom = st\_combine(rbind(pl1, pl2)), crs=4326)

tst = editFeatures(mpl, editor = "leafpm")

# look at our creation

mapview(tst)



Please note that right mouse click deletes vertexes. For a more real  
world application franconia[5,] from mapview has a hole. Try to edit  
it with the following code.

library(sf)

library(leaflet)

library(mapview)

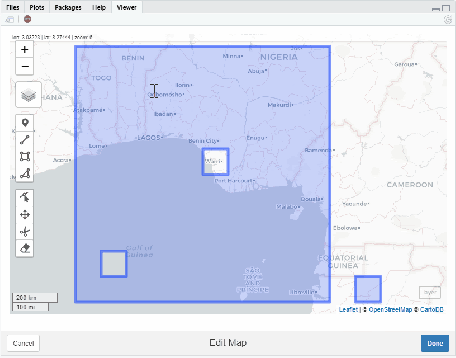
library(mapedit)

library(leafpm)

editFeatures(franconia[5,], editor="leafpm")

**Snapping**

Leaflet.pm gives us a very pleasant snapping experience, so if you  
want to snap, set editor = "leafpm" and snap away. Snapping is  
particular important when drawing/digitizing features from scratch. Here  
is how it looks with the example from above.



Snapping is enabled by default.

**Fixes For Lingering Issues**

**GeoJSON Precision**

I am happy to report that we have found a solution for the loss of  
precision. Please let us know if you discover any remaining problems.

**Mulitlinestring Editing**

Leaflet.js and multilinestrings don’t get along For complete support of sf, mapedit should work with  
multilinestring.

We backed into a solution with MULTILINESTRING since Leaflet.pm’s  
approach fits better with MULTI\* features. As an example, let’s edit  
one of the trails from mapview.

library(sf)

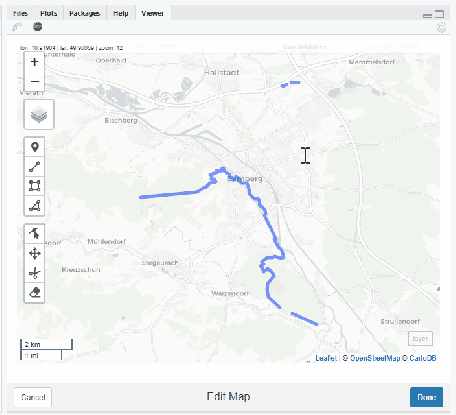
library(leaflet)

library(mapview)

library(mapedit)

library(leafpm)

editFeatures(trails[4,], editor="leafpm")



**Conclusion and Thanks**

Although the funding is over, we still expect to actively maintain and  
improve mapedit. One feature that we had hoped to implement as part of  
the mapedit toolset was editing of feature attributes. This turned out  
to be very ambitious, and unfortunately we were not able to implement a  
satisfactory solution for this feature during the funding period. We  
plan, however, to develop a solution. Your participation, ideas, and  
feedback are as vital as ever, so please continue to engage. Thanks to  
all those who have contributed so far and thanks to all open source  
contributors in the R and JavaScript communities.