Mapping with ggplot2

Samuel Croker

April 27, 2012

Thematic Maps in R

Essential Elements

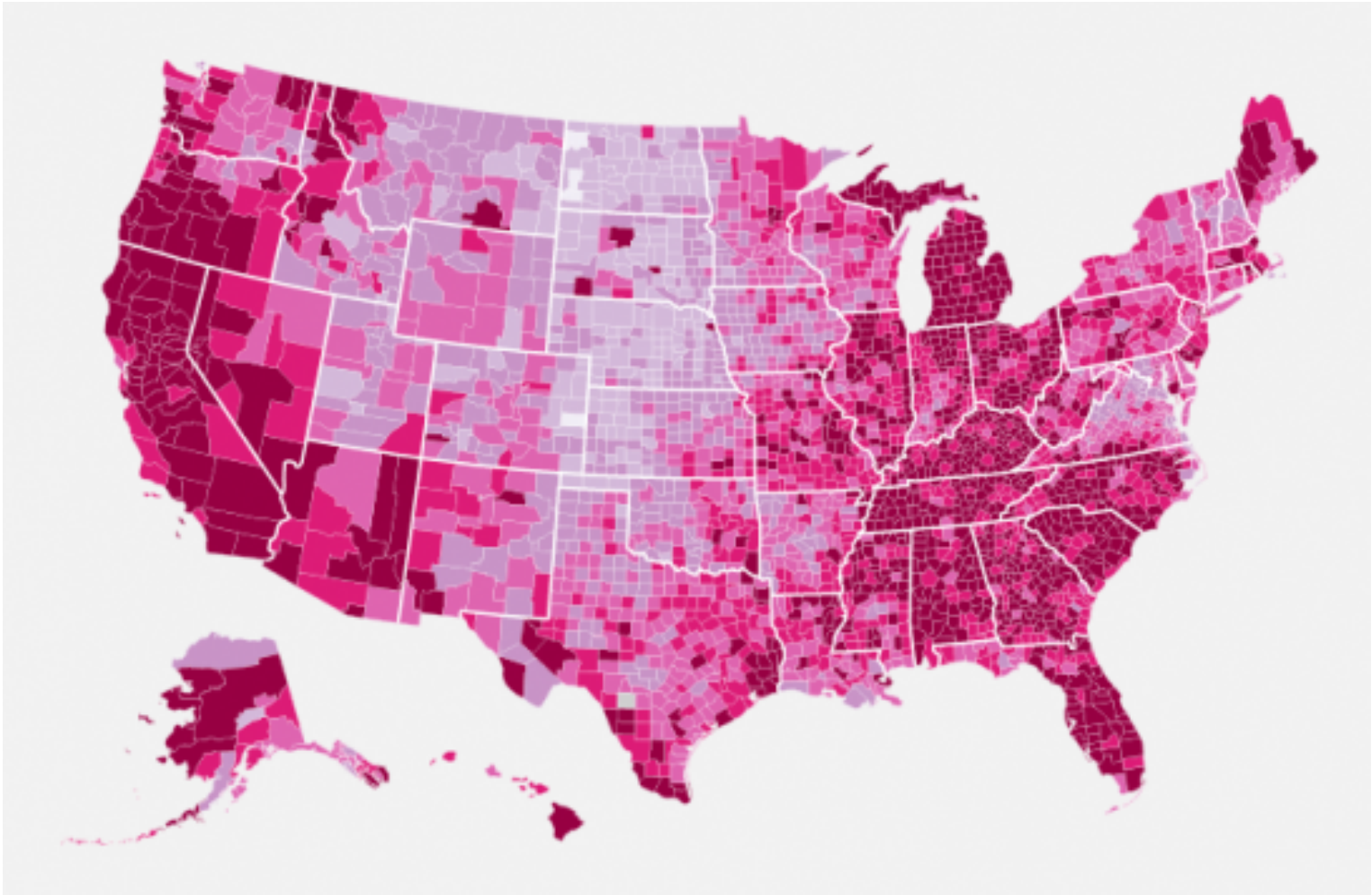
- Read geo-spatial data into R
- Associate the geo-spatial data with the thematic data
- Draw the map with colored attributes from the thematic data

Key Ideas

- Graph elements are lines or polygons with coordinates as vertices
- The grouping and ordering of polygons or lines is important
- Projections can be problematic
- More detail means more runtime

This is nothing really new, see Further Reading

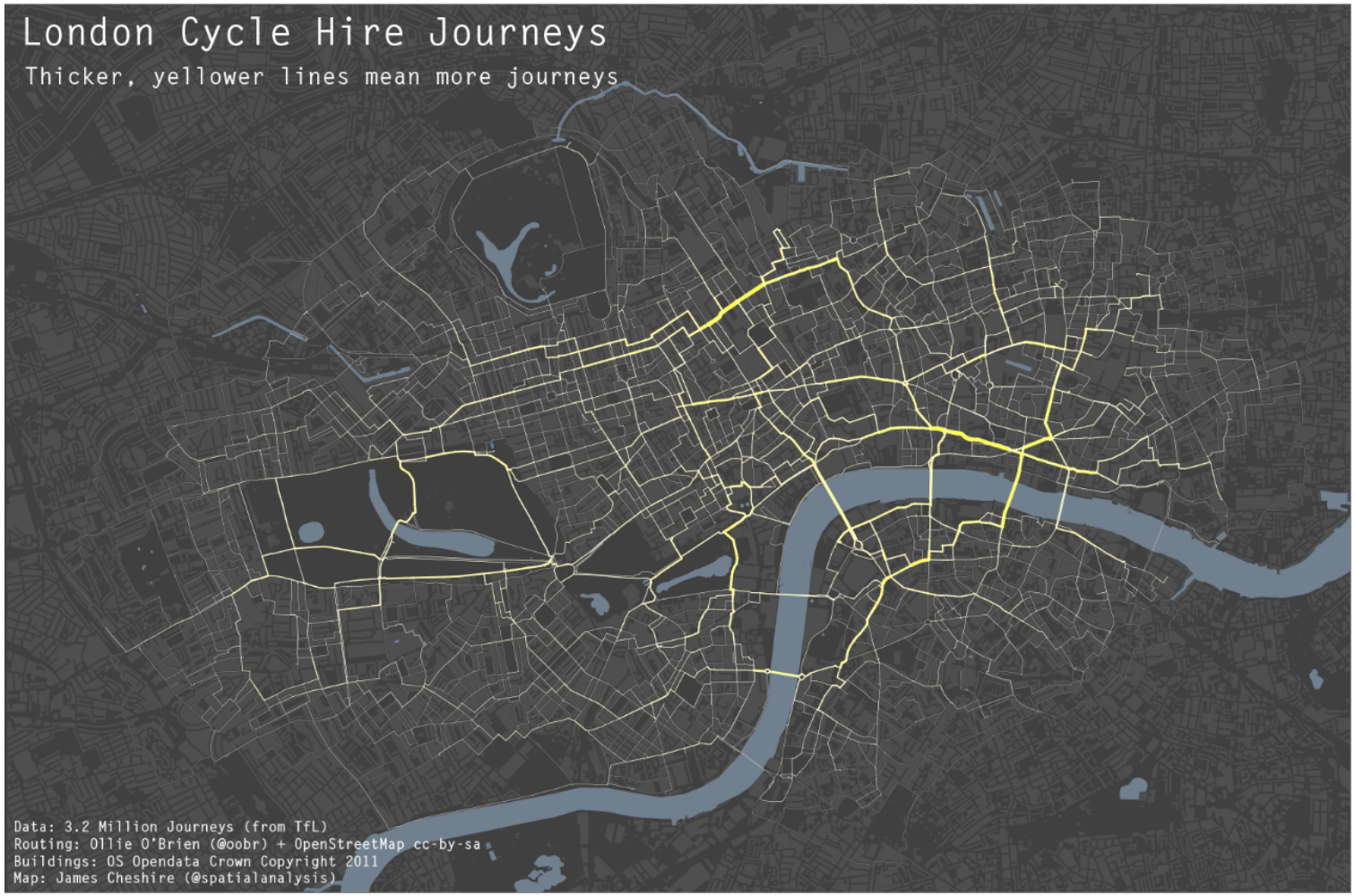
Inspiration



Source:

http://blog.revolutionanalytics.com/2009/11/choropleth-map-r-challenge.html

Inspiration



Source: http://spatialanalysis.co.uk/2012/02/great-maps-ggplot2/

“Simple” Example

Task: Create a map of Michigan with each county colored to show population

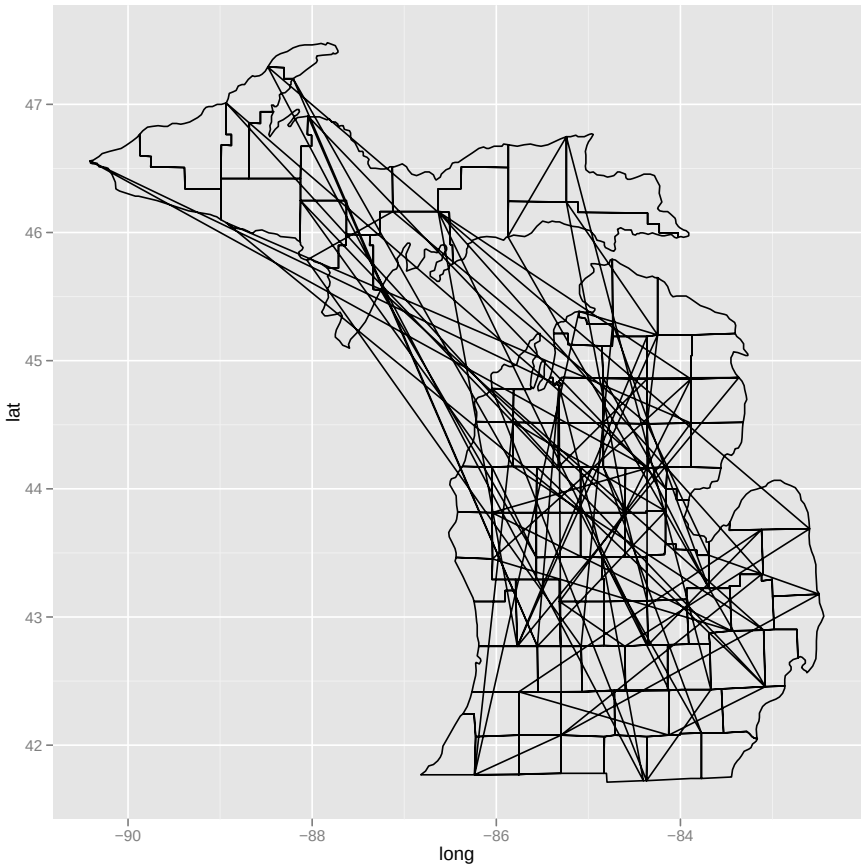
- Get a polygonal map dataset - US State and County data are in the `maps` package
- Get the population data for each county - Census.gov
- Associate the two sources and create choropleth

Get the Map Dataset


```
library(maps)
library(ggplot2)
mapcty <- map_data("county",region='michigan')
head(mapcty)
       long      lat group order   region subregion
1 -83.88675 44.85686     1     1 michigan    alcona
2 -83.36536 44.86832     1     2 michigan    alcona
3 -83.36536 44.86832     1     3 michigan    alcona
4 -83.33098 44.83968     1     4 michigan    alcona
5 -83.30806 44.80530     1     5 michigan    alcona
6 -83.30233 44.77665     1     6 michigan    alcona
```

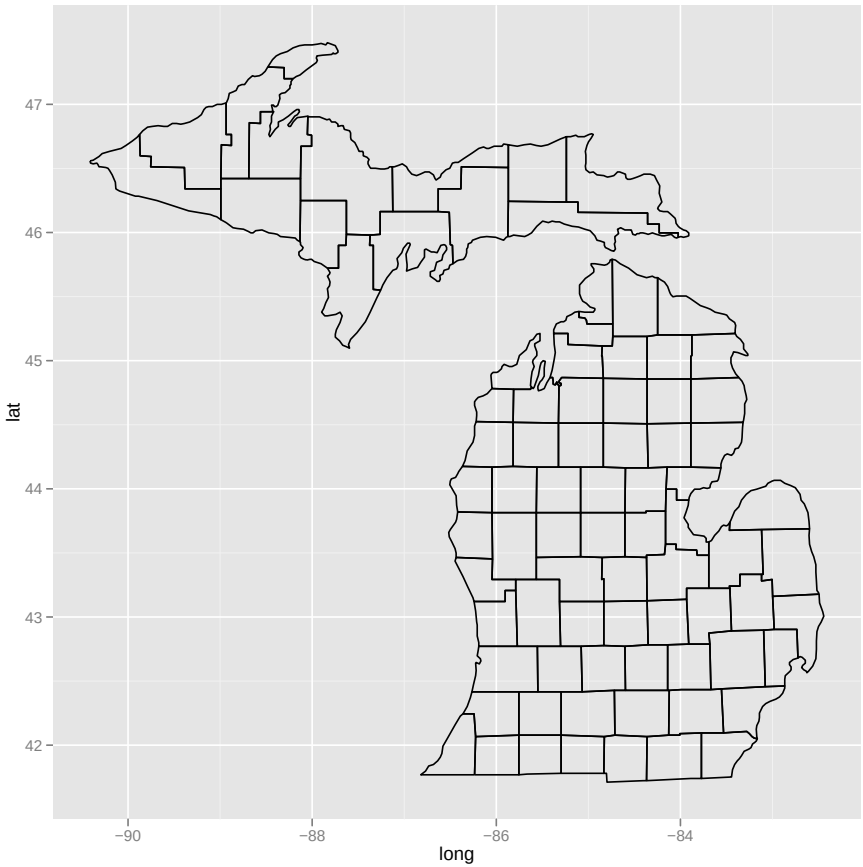
Use ggplot2 to Produce a Map


```
ggplot(data=mapcty,aes(long,lat)) + geom_path()
```



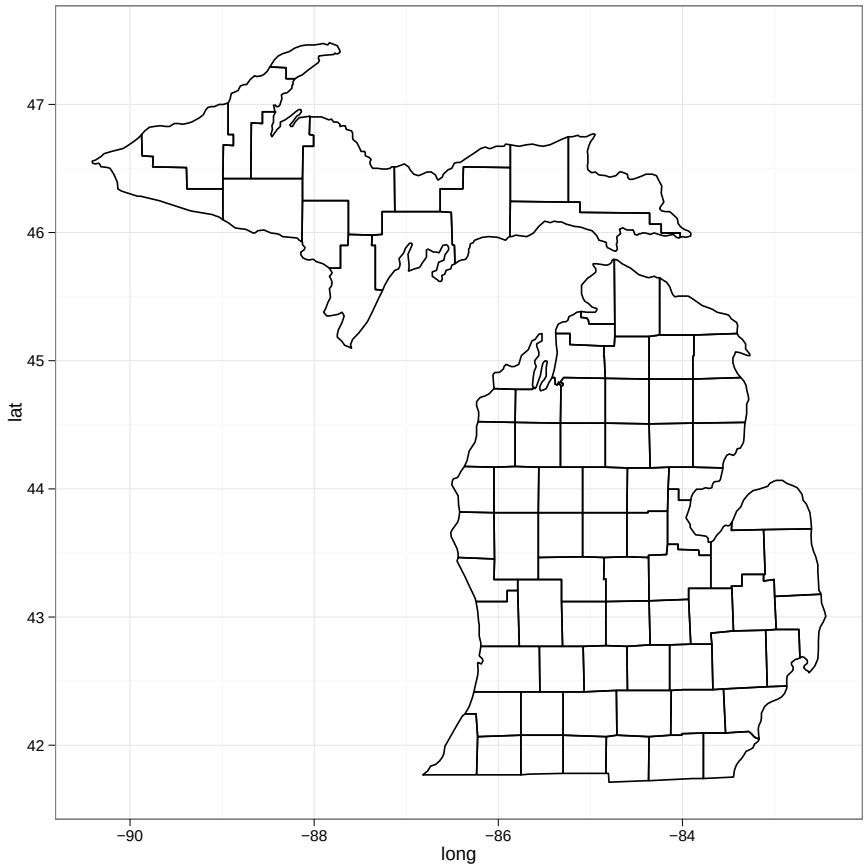
Use ggplot2 to Produce a Map


```
ggplot(data=mapcty,aes(long,lat,group=group)) + geom_path()
```



Add Other Themes


```
ggplot(data=mapcty,aes(long,lat,group=group))
    + geom_path()
    + theme_bw()
```



Create Thematic Layer

The next step is to add some data to the map so that a choropleth is produced.

```
mipop <- read.csv("~/Reports/R_Network_Choro/mipop.csv")
head(mipop)
   county    pop
1  alcona  11091
2   alger   9286
3 allegan 113449
4  alpena  29289
5  antrim  23834
6  arenac  16092
```

Join Data to Map


```
choro_map <- merge(x=mapcty, y=mipop
                   , by.x='subregion'
                   , by.y='county')
head(choro_map)
  subregion      long      lat group order   region   pop
1    alcona -83.88675 44.85686     1     1 michigan 11091
2    alcona -83.36536 44.86832     1     2 michigan 11091
3    alcona -83.36536 44.86832     1     3 michigan 11091
4    alcona -83.33098 44.83968     1     4 michigan 11091
5    alcona -83.30806 44.80530     1     5 michigan 11091
6    alcona -83.30233 44.77665     1     6 michigan 11091
```

Format Thematic Data

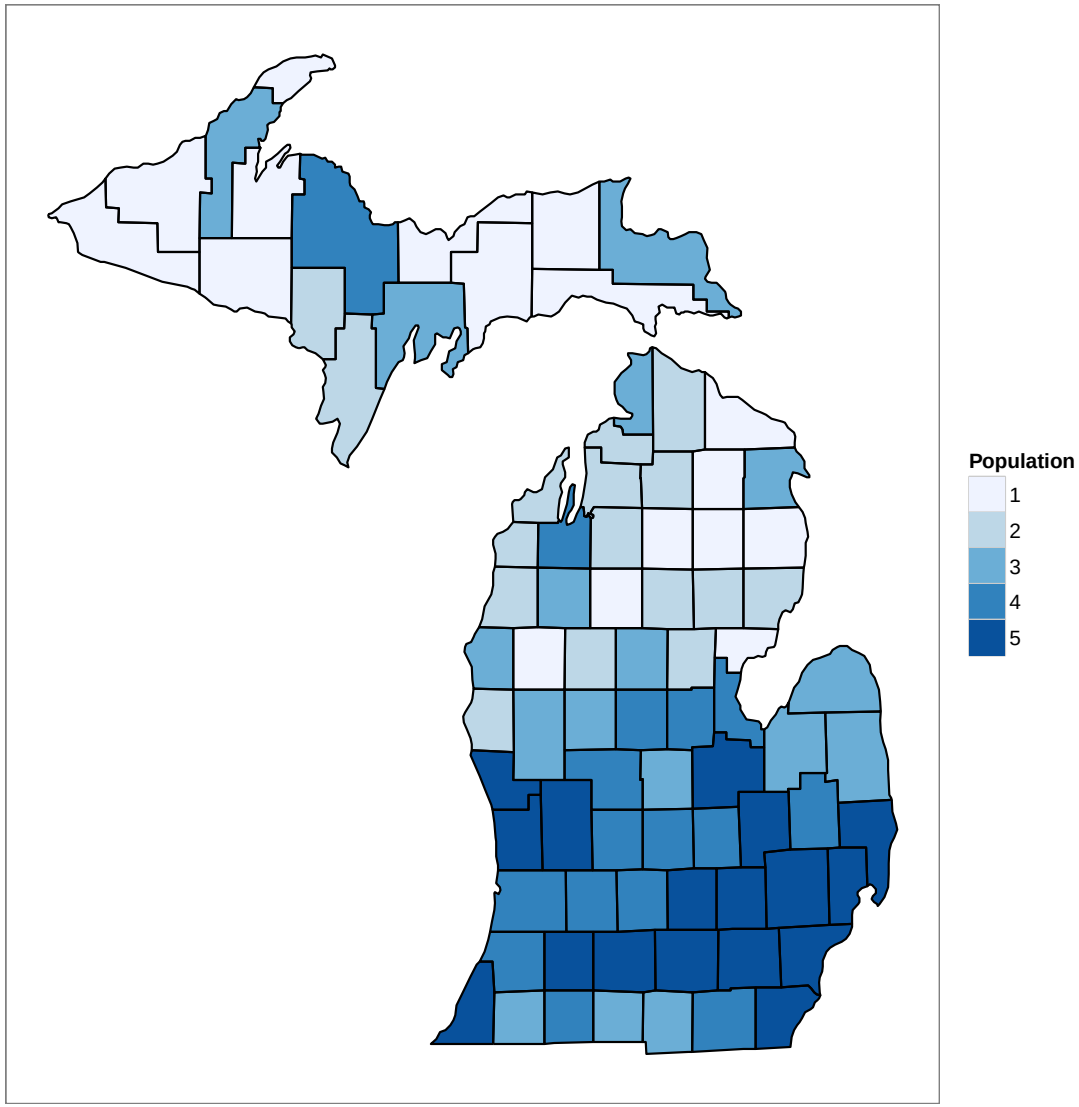

```
mipop <- data.frame(mipop,cut(mipop$pop
         ,breaks = quantile(mipop$pop
                            , probs = seq(0, 1, .2 ))
         ,include.lowest = TRUE
         ,labels=1:5))
choro_map <- merge(x=mapcty, y=mipop
              , by.x='subregion'
              , by.y='county')
colnames(choro_map)[8]<-'bucket'
```

```
head(choro_map)
  subregion      long      lat group order   region   pop bucket
1    alcona -83.88675 44.85686     1     1 michigan 11091      1
2    alcona -83.36536 44.86832     1     2 michigan 11091      1
3    alcona -83.36536 44.86832     1     3 michigan 11091      1
4    alcona -83.33098 44.83968     1     4 michigan 11091      1
5    alcona -83.30806 44.80530     1     5 michigan 11091      1
6    alcona -83.30233 44.77665     1     6 michigan 11091      1
```

Render Map with ggplot2


```
ggplot(data=choro_map,aes(long,lat,group=group)) +
    geom_polygon(aes(fill=bucket)) +
    scale_fill_brewer(name='Population') +
    geom_path() +
    scale_x_continuous("", breaks='null') +
    scale_y_continuous("", breaks='null') +
    theme_bw()
```

Use ggplot2 to Produce a Map



Importing Maps Into R

What kinds of map data?

- Zipcode boundaries
- Hydrography
- Roads

Where to get GIS Data?

- US Census Bureau
- Georgia GIS Clearinghouse
- ESRI
- County and City Government

Importing GIS Data Into R

`rgdal` imports shapefiles into a R S4 geospacial vector. The `readOGR` function of

```
library(rgdal)
shapefilepath = "C:\\<path to directory containing shapefile>"
atltgr =  readOGR( dsn=shapefilepath
                ,layer="<shapefilename - no extension")
atltgr@data$id = rownames(atltgr@data)
atltgr.points = fortify(atltgr, region="id")
atltgr.df <- merge(atltgr.points, atltgr@data, by="id")
```

Unfortunately, `ggplot2` does not work with S4 objects directly.

Importing GIS Data Into R II

The `fortify` function in `ggplot2` converts S4 geospatial objects into a data frame.

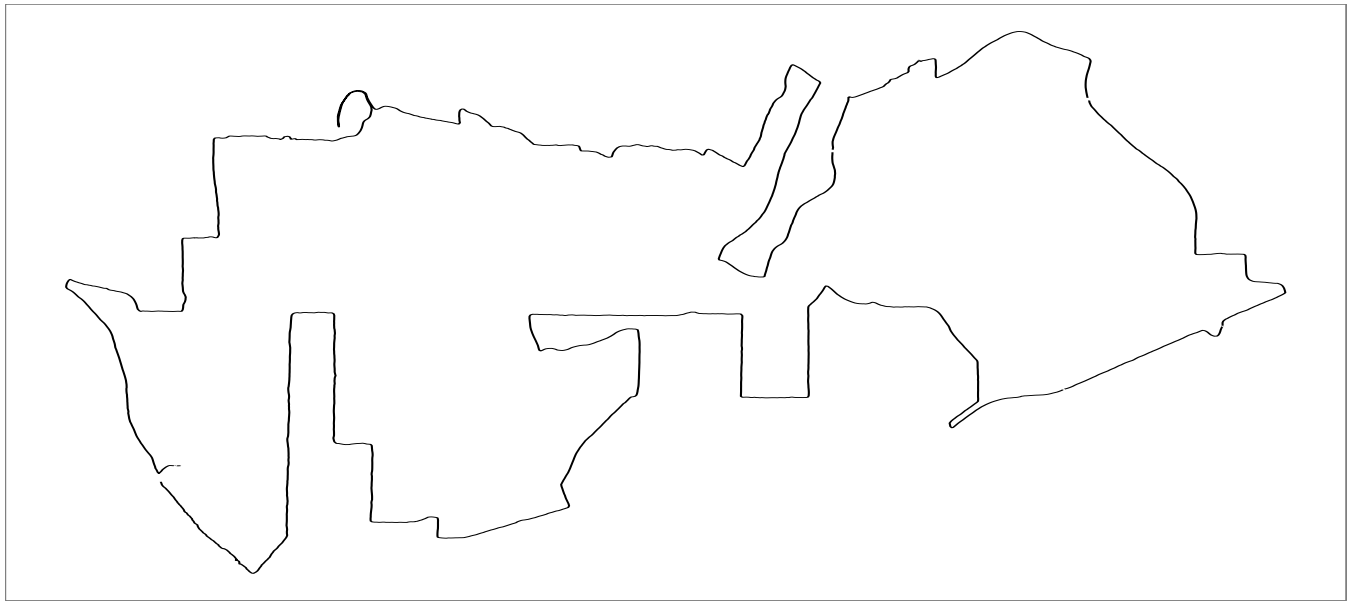
```
library(rgdal)
shapefilepath = "C:\\<path to directory containing shapefile>"
atltgr =  readOGR( dsn=shapefilepath
                ,layer="<shapefilename - no extension")
atltgr@data$id = rownames(atltgr@data)
atltgr.points = fortify(atltgr, region="id")
atltgr.df <- merge(atltgr.points, atltgr@data, by="id")
```

Importing Garmin TCX Files


```
doc <- xmlParse(TCXfilepath)
namespaces <- c(ns=
 "http://www.garmin.com/xmlschemas/TrainingCenterDatabase/v2")
breakdown <- xmlToDataFrame(nodes <-
   getNodeSet( doc
              , "//ns:Trackpoint"
              , namespaces
  )
)
tcxfile  <- plyr::ldply(nodes[-1]
        , as.data.frame(xmlToList
                        ,stringsAsFactors=F
                        ,simplify=T))
tcxfile <- setNames(tcxfile
  ,c('time', 'lat', 'long', 'alt'
  ,'distance','bpm','speed','cadence'))
```

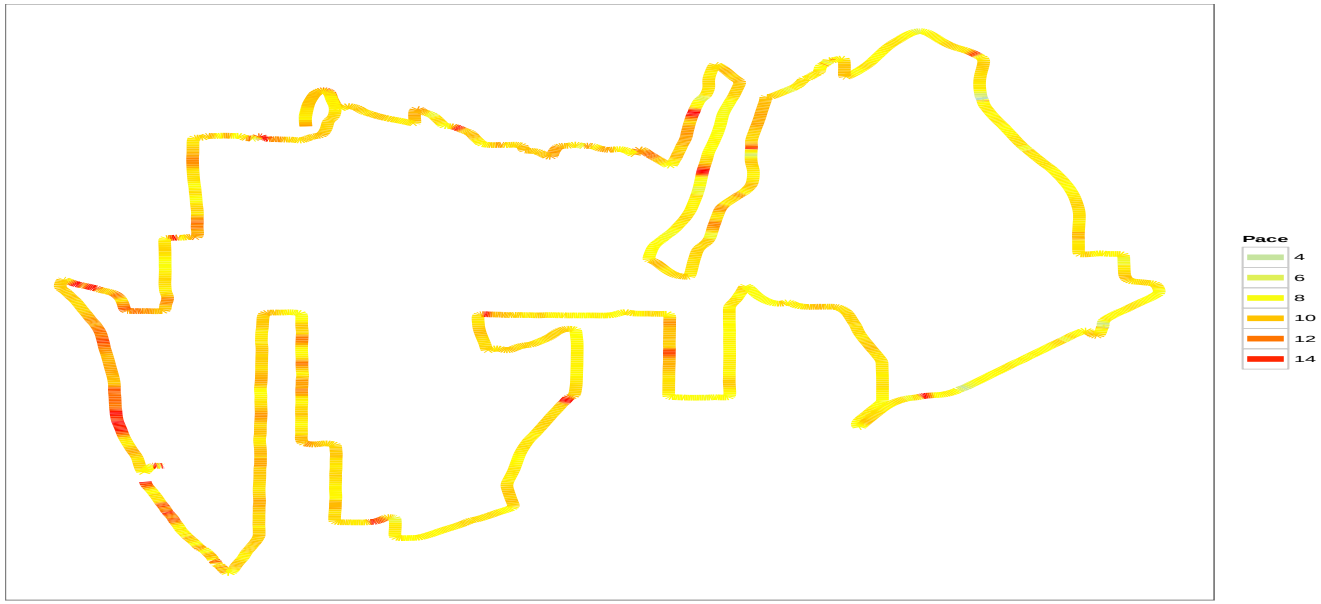
Plot of TCX Position Data


```
ggplot(mydf2, aes(x=long,y=lat)) +
     geom_path() +
     scale_x_continuous("", breaks='null') +
     scale_y_continuous("", breaks='null') +
     theme_bw()
```



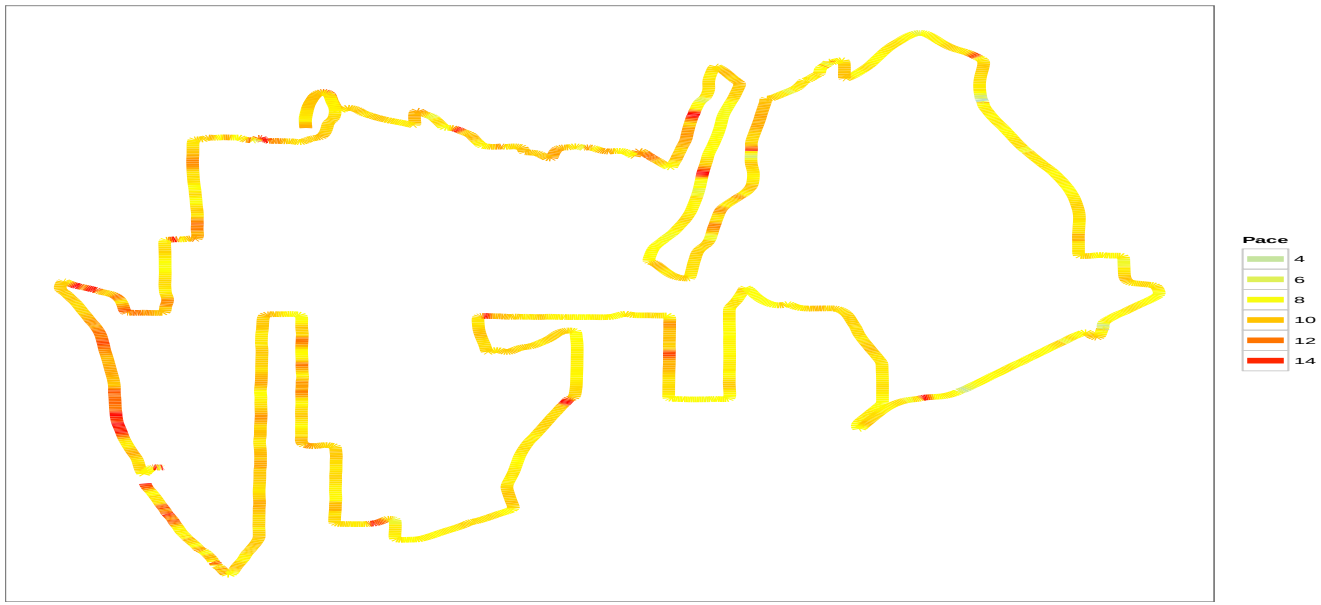
Adding Thematic Layer


```
ggplot(mydf2, aes(x=long,y=lat)) +
     geom_path(aes(colour=pacezone) +
     scale_colour_manual(values= pacepal,name='Pace') + ...
```

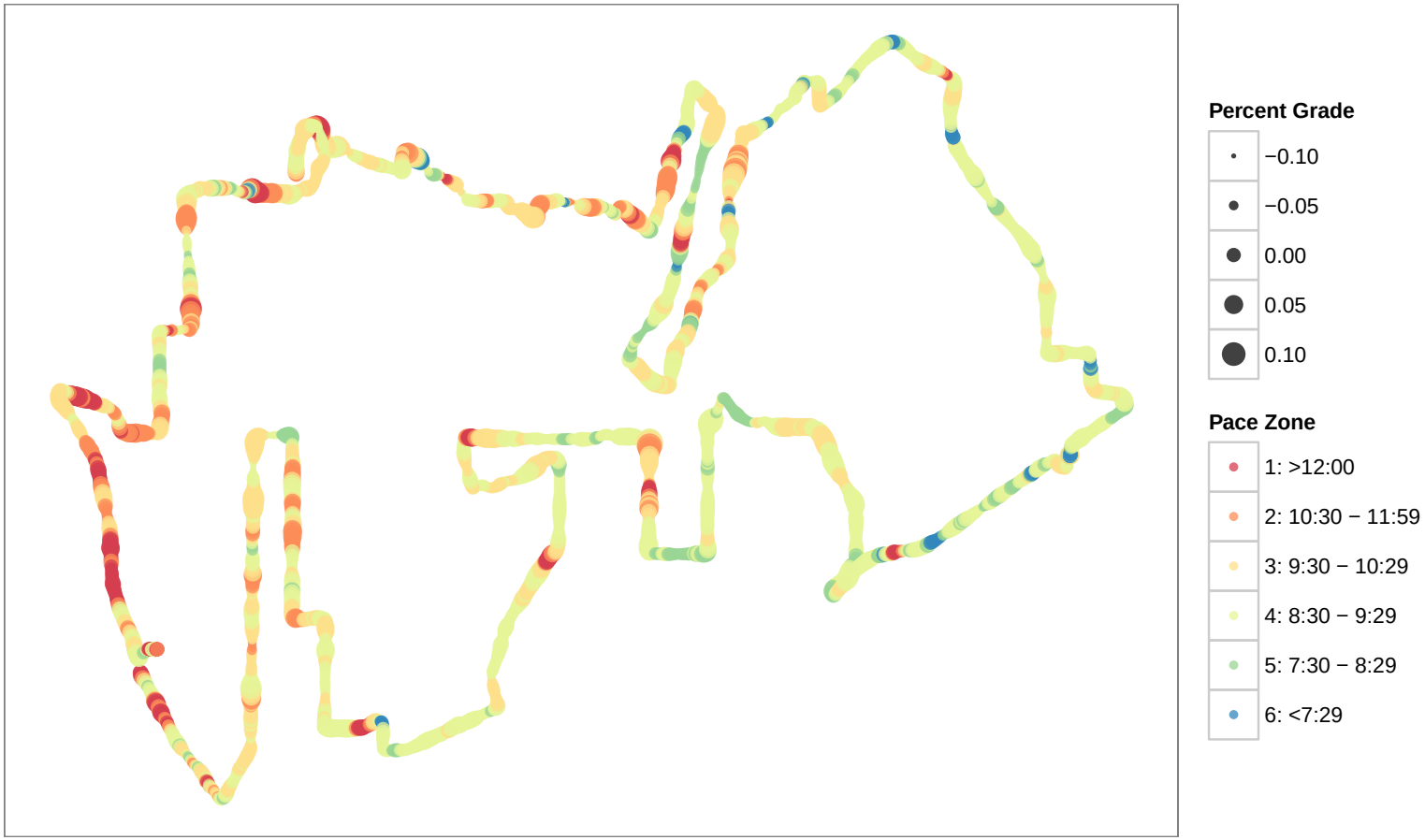


Two Thematic Layers


```
ggplot(mydf2, aes(x=long_s,y=lat_s))
  + geom_point(aes(x=long_s,y=lat_s
                   ,colour=pacezone
                   ,size=pct_grade_s)
                ,alpha=.75)
  + scale_colour_manual(values= pacepal,name='Pace Zone')
  + scale_size_continuous(name = 'Percent Grade') +...
```



Two Thematic Layers II



Import Street Data


```
atltgr =  readOGR(dsn="<<path>>", layer="tgr121rds")
atltgr@data$id = rownames(atltgr@data)
atltgr.points = fortify(atltgr, region="id")
atltgr.df <- merge(atltgr.points, atltgr@data, by="id")
```

The `minlong...` variables are calculated by finding the coordinate geographic bounding box of the geodata in the TCX file.

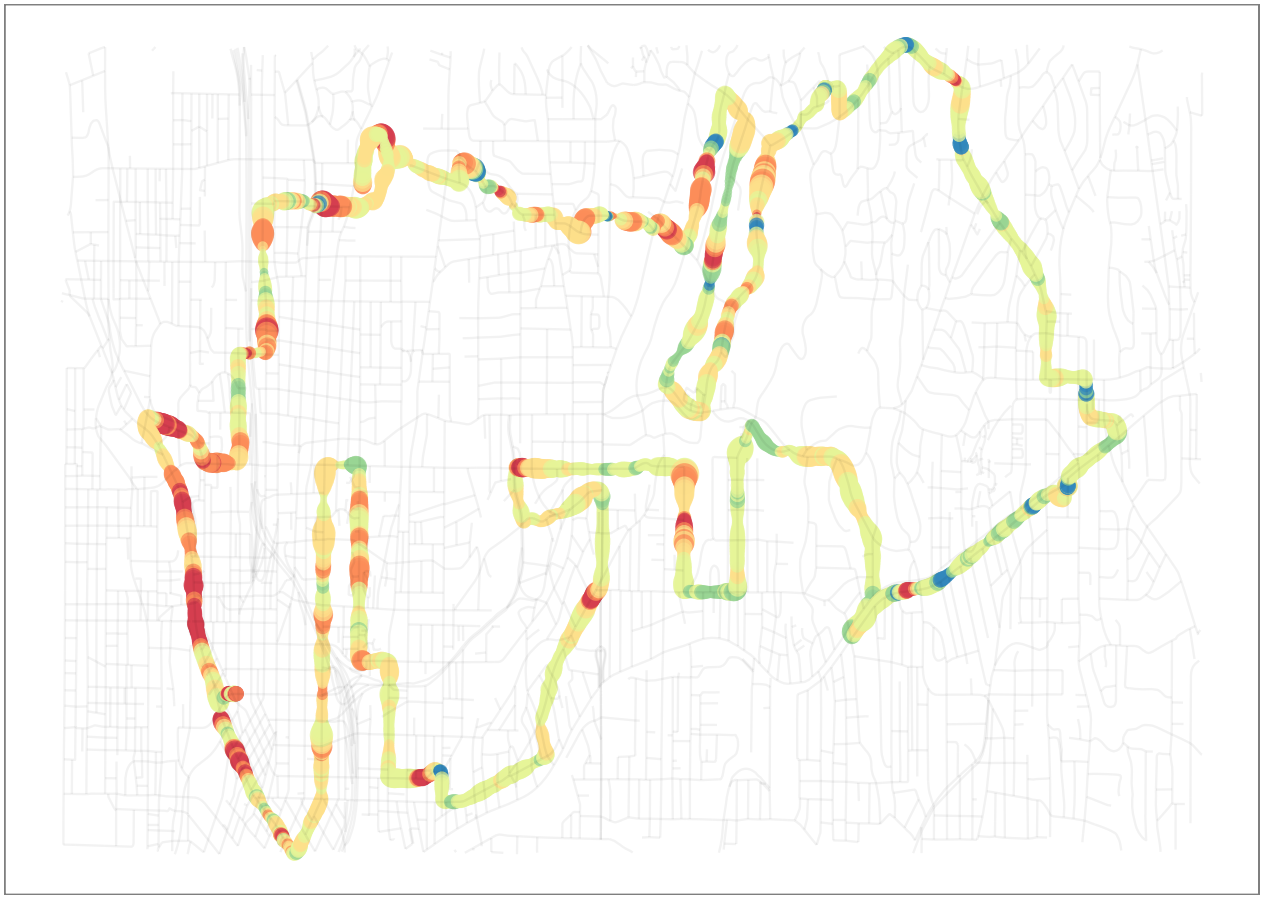
```
athj <- subset(atltgr.df,long <= maxlong
    & long >= minlong & lat <= maxlat & lat >= minlat)
```

This way, the resulting map is cropped to the extent of the TCX data, which speeds up processing and makes a better looking graph.

Create Final Map


```
ggplot(mydf2, aes(x=long_s,y=lat_s))
  + geom_point(aes(x=long_s,y=lat_s
     ,colour=pacezone,size=pct_grade_s),alpha=.75)
  + scale_colour_manual(values= pacepal,name='Pace Zone')
  + scale_size_continuous(name = 'Percent Grade')
  + geom_path(data=athj
     , aes(x=long,y=lat,group=group),size=.5,alpha=.05)
  + geom_path(data=dklbj
     , aes(x=long,y=lat,group=group),size=.5,alpha=.05)
  + quiet
  + theme_bw()
```

Two Thematic Layers II



Summary

- Watch out for...
  - Different projections
  - Large shapefiles

For Further Reading I

- http://blog.revolutionanalytics.com/2009/11/choropleth-challenge-result.html
- http://stackoverflow.com/questions/1260965/developing-geographic-thematic-maps-with-r
- http://spatialanalysis.co.uk/2012/02/great-maps-ggplot2/