

Appendix

STA442 Homework 4

Jiaqi Bi

```
# election results
# https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/NH5S2I
# save file wi_2016.zip in ~/Downloads

xFiles = unzip('~Downloads/wi_2016.zip')
x = raster::shapefile(grep("shp$", xFiles, value=TRUE))

# county boundaries
cFileShp = Pmisc::downloadIfOld('https://www.sciencebase.gov/catalog/file/get/4f4e4a2ee4b07f02a
    file='../data/countyBoundaries.zip')
wisconsinC1 = raster::shapefile(grep('shp$', cFileShp, value=TRUE))

# subcounty boundaries
cSubFileShp = Pmisc::downloadIfOld('https://www2.census.gov/geo/tiger/TIGER2016/COUSUB/tl_2016
wisconsinCsub1 = raster::shapefile(grep('shp$', cSubFileShp, value=TRUE))

# census tract boundaries
ctFile = Pmisc::downloadIfOld('https://www2.census.gov/geo/pvs/tiger2010st/55_Wisconsin/55/tl_2
ct = raster::shapefile(grep("shp$", ctFile, value=TRUE))
raster::projection(ct) = mapmisc::crsLL

# reproject
wisconsinCsub2 = sp::spTransform(wisconsinCsub1,
    mapmisc::omerc(wisconsinCsub1))
ct2 = sp::spTransform(ct, wisconsinCsub2@proj4string)
wisconsinC = sp::spTransform(wisconsinC1[wisconsinC1$STFIPS==55,], wisconsinCsub2@proj4string)
x = sp::spTransform(x, wisconsinCsub2@proj4string)

# format subcounty data
wisconsinCsub3 = wisconsinCsub2[wisconsinCsub2$GEOID %in% x$MCD_FIPS,]
wisconsinCsub4 = rgeos::gUnaryUnion(wisconsinCsub3, wisconsinCsub3$COUSUBFP)
wisconsinCsub5 = rgeos::gSimplify(wisconsinCsub4, tol=50, TRUE)

cSubData = wisconsinCsub3@data[match(
    names(wisconsinCsub5),
    wisconsinCsub3$COUSUBFP),]
rownames(cSubData) = names(wisconsinCsub5)
wisconsinCsub = sp::SpatialPolygonsDataFrame(
```

```

    wisconsinCsub5, cSubData)

# format election results
x$Total = as.numeric(x$G16PRERTru) + as.numeric(x$G16PREDCli)
x$trump = as.numeric(x$G16PRERTru)
x$persons = as.numeric(as.character(x$PERSONS))
x$pop = pmax(x$persons, x$Total)
x$white = as.numeric(as.character(x$WHITE))
x$amindian = as.numeric(as.character(x$AMINDIAN))
x$propWhite = pmax(x$white / x$pop, 0, na.rm=TRUE)
x$propInd = pmax(x$amindian / x$pop, 0, na.rm=TRUE)
x$area = rgeos::gArea(x, TRUE)/10^6
x$pdens = x$pop/ x$area
x$logPdens = log(pmax(1, x$pdens))

# county-level election results
xAggCounty = aggregate(x@data[,c('white','persons','Total','trump','pop','amindian')], x@data[,
    wisconsinCm = sp::merge(wisconsinC, xAggCounty,
        by.x='CTFIPS', by.y='CNTY_FIPS')
wisconsinCm$propTrump = wisconsinCm$trump / wisconsinCm$Total
wisconsinCm$propWhite = wisconsinCm$white / wisconsinCm$persons
wisconsinCm$propInd = wisconsinCm$amindian / wisconsinCm$persons
wisconsinCm$area = rgeos::gArea(wisconsinCm, TRUE)/10^6
wisconsinCm$pdens = wisconsinCm$persons/ wisconsinCm$area
wisconsinCm$logPdens = log(wisconsinCm$pdens)

# subcounty election results
xAggSubCounty = aggregate(x@data[,
    c('white','persons','Total','trump','pop','amindian')],
    x@data[,c('COUSUBFP'),drop=FALSE], sum, na.rm=TRUE)
wisconsinCsubm = sp::merge(wisconsinCsub, xAggSubCounty,
    by.x='COUSUBFP', by.y='COUSUBFP')

wisconsinCsubm$propTrump = wisconsinCsubm$trump / wisconsinCsubm$Total
wisconsinCsubm$propWhite = wisconsinCsubm$white / wisconsinCsubm$persons
wisconsinCsubm$propInd = wisconsinCsubm$amindian / wisconsinCsubm$persons
wisconsinCsubm$area = rgeos::gArea(wisconsinCsubm, TRUE)/10^6
wisconsinCsubm$pdens = wisconsinCsubm$persons/ wisconsinCsubm$area
wisconsinCsubm$logPdens = log(wisconsinCsubm$pdens)

wisconsinTract = x
wisconsinTract$propTrump = wisconsinTract$trump / wisconsinTract$Total
save(wisconsinCsubm, wisconsinCm, wisconsinTract,
    file='../Downloads/wisconsin.RData')

(load('../Downloads/wisconsin.RData'))

```

```

resC = diseasemapping::bym(trump ~ logPdens + propWhite + propInd, data = wisconsinCm,
  prior = list(sd = c(log(2.5), 0.5), propSpatial = c(0.5, 0.5)),
  Ntrials = wisconsinCm$Total, family='binomial')

resSubC = diseasemapping::bym(trump ~ logPdens + propWhite + propInd, data = wisconsinCsubm,
  prior = list(sd = c(log(2.5), 0.5), propSpatial = c(0.5, 0.5)),
  Ntrials = wisconsinCsubm$Total, family='binomial')

x2 = x[which(x$pop > 1),]
resP = diseasemapping::bym(trump ~ logPdens + propWhite + propInd, data = x,
  prior = list(sd = c(log(2.5), 0.5), propSpatial = c(0.5, 0.5)),
  Ntrials = x$Total, family='binomial')

resP$parameters$summary[,paste0(c(0.5, 0.025, 0.975), 'quant')]
resSubC$parameters$summary[,paste0(c(0.5, 0.025, 0.975), 'quant')]
resC$parameters$summary[,paste0(c(0.5, 0.025, 0.975), 'quant')]

save(resP, resSubC, resC, file='resWisconsinAll.RData', compress='xz')

resTrump = diseasemapping::bym(trump ~ logPdens + propWhite + propInd,
  data = wisconsinCsubm,
  prior = list(sd = c(log(2.5), 0.5), propSpatial = c(0.5, 0.5)),
  Ntrials = wisconsinCsubm$Total, family='binomial')
save(resTrump, file='../data/resWisconsin.RData', compress='xz')

(load('../data/resWisconsin.RData'))

theColTrump = mapmisc::colourScale(wisconsinCsubm$propTrump, col='RdBu',
  breaks=sort(unique(setdiff(c(0,1, seq(0.2,0.8,by=0.1)), 0.5))),
  style='fixed', rev=TRUE)

theColPop = mapmisc::colourScale(wisconsinCsubm$pdens, col='Spectral',
  breaks=11,
  style='equal', transform='log', digits=1, rev=TRUE)

theColWhite = mapmisc::colourScale(wisconsinCsubm$propWhite,
  col='Spectral',
  breaks=c(0, 0.5, 0.8, 0.9, seq(0.9, 1, by=0.02)),
  style='fixed', rev=TRUE)

theColInd = mapmisc::colourScale(wisconsinCsubm$propInd,
  col='Spectral',
  breaks=seq(0, 1, by=0.1),
  style='fixed', rev=TRUE)

theBg = mapmisc::tonerToTrans(mapmisc::openmap(wisconsinCm,
  fact=2, path='stamen-toner'), col='grey30')

```

```
theInset = mapmisc::openmap(wisconsinCm, zoom=6, path='stamen-watercolor',
  crs=mapmisc::crsMerc, buffer=c(0,1500,100,700)*1000)
```

```
library('sp')
```

```
mapmisc::map.new(wisconsinCsubm, 0.85)
sp::plot(wisconsinCsubm, col = theColTrump$plot, add=TRUE,
  lwd=0.2)
raster::plot(theBg, add=TRUE, maxpixels=10^7)
mapmisc::insetMap(wisconsinCsubm, 'bottomright', theInset,
  outer=TRUE, width=0.35)
mapmisc::scaleBar(wisconsinCsubm, 'top', cex=0.8)
mapmisc::legendBreaks('topright', theColTrump, bty='n', inset=0)
```

```
mapmisc::map.new(wisconsinCsubm, 0.85)
plot(wisconsinCsubm, col = theColPop$plot, add=TRUE,
  lwd=0.2)
plot(theBg, add=TRUE, maxpixels=10^7)
mapmisc::legendBreaks('right', theColPop, bty='n', inset=0)
```

```
mapmisc::map.new(wisconsinCsubm, 0.85)
plot(wisconsinCsubm, col = theColInd$plot, add=TRUE,
  lwd=0.2)
plot(theBg, add=TRUE, maxpixels=10^7)
mapmisc::legendBreaks('right', theColInd, bty='n', inset=0)
```

```
mapmisc::map.new(wisconsinCsubm, 0.85)
plot(wisconsinCsubm, col = theColWhite$plot, add=TRUE,
  lwd=0.2)
plot(theBg, add=TRUE, maxpixels=10^7)
mapmisc::legendBreaks('right', theColWhite, bty='n', inset=0)
```

```
theColRandom = mapmisc::colourScale(
  resTrump$data$random.mean,
  col='Spectral', breaks = 11,
  style='quantile', rev=TRUE, dec=1)
theColFit= mapmisc::colourScale(
  resTrump$data$fitted.invlogit,
  col='RdBu', rev=TRUE,
  breaks=sort(unique(setdiff(c(0,1, seq(0.2,0.8,by=0.1)), 0.5))),
  style='fixed')
```

```
mapmisc::map.new(wisconsinCsubm, 0.85)
plot(resTrump$data, col = theColRandom$plot, add=TRUE,
  lwd=0.2)
plot(theBg, add=TRUE, maxpixels=10^7)
mapmisc::legendBreaks('topright', theColRandom)
```

```

mapmisc::map.new(wisconsinCsubm, 0.85)
plot(resTrump$data, col = theColFit$plot, add=TRUE,
      lwd=0.2)
plot(theBg, add=TRUE, maxpixels=10^7)
mapmisc::legendBreaks('topright', theColFit)

#https://www.huduser.gov/portal/qct/data_request.html
options(timeout=100)
incomeFile = '../data/ACSST5Y2015.S1701_data_with_overlays_2020-11-27T170117.csv'
income = read.table(incomeFile, header=TRUE, skip=1, sep=',')
income = income[,grep("Margin.of.Error", names(income), invert=TRUE)]
names(income) = gsub("Estimate[[:punct:]]+|..Population.for.whom.poverty.status.is.determined",
                    "", names(income))

#income = income[,c('id', 'Geographic.Area.Name', 'Percent.below.poverty.level')]
income[grep("55091", income$id), 1:6]
wisconsinCsubm@data[wisconsinCsubm$COUNTYFP=="091", 1:9]
income$GEOID = gsub("0600000US", "", income$id)
stuff = (wisconsinCsubm$GEOID %in% income$GEOID)
wisconsinCsubm@data[!stuff,]
income[grep("550597", income$GEOID), 1:5]

theColTrump = mapmisc::colourScale(wisconsinCsubm$propTrump,
  col = "RdBu", breaks = sort(unique(setdiff(c(0, 1, seq(0.2,
    0.8, by = 0.1)), 0.5))), style = "fixed", rev = TRUE)
theColPop = mapmisc::colourScale(wisconsinCsubm$pdens, col = "Spectral",
  breaks = 11, style = "equal", transform = "log", digits = 1,
  rev = TRUE)
theColWhite = mapmisc::colourScale(wisconsinCsubm$propWhite,
  col = "Spectral", breaks = c(0, 0.5, 0.8, 0.9, seq(0.9,
    1, by = 0.02)), style = "fixed", rev = TRUE)
theColInd = mapmisc::colourScale(wisconsinCsubm$propInd,
  col = "Spectral", breaks = seq(0, 1, by = 0.1), style = "fixed",
  rev = TRUE)
theBg = mapmisc::tonerToTrans(mapmisc::openmap(wisconsinCm,
  fact = 2, path = "stamen-toner"), col = "grey30")
theInset = mapmisc::openmap(wisconsinCm, zoom = 6, path = "stamen-watercolor",
  crs = mapmisc::crsMerc, buffer = c(0, 1500, 100, 700) *
1000)
library("sp")
mapmisc::map.new(wisconsinCsubm, 0.85)
sp::plot(wisconsinCsubm, col = theColTrump$plot, add = TRUE,
  lwd = 0.2)
raster::plot(theBg, add = TRUE, maxpixels = 10^7)
mapmisc::insetMap(wisconsinCsubm, "bottomright", theInset,
  outer = TRUE, width = 0.35)
mapmisc::scaleBar(wisconsinCsubm, "top", cex = 0.8)

```

```

mapmisc::legendBreaks("topright", theColTrump, bty = "n",
  inset = 0)
mapmisc::map.new(wisconsinCsubm, 0.85)
plot(wisconsinCsubm, col = theColPop$plot, add = TRUE, lwd = 0.2)
plot(theBg, add = TRUE, maxpixels = 10^7)
mapmisc::legendBreaks("right", theColPop, bty = "n", inset = 0)
mapmisc::map.new(wisconsinCsubm, 0.85)
plot(wisconsinCsubm, col = theColInd$plot, add = TRUE, lwd = 0.2)
plot(theBg, add = TRUE, maxpixels = 10^7)
mapmisc::legendBreaks("right", theColInd, bty = "n", inset = 0)
mapmisc::map.new(wisconsinCsubm, 0.85)
plot(wisconsinCsubm, col = theColWhite$plot, add = TRUE,
  lwd = 0.2)
plot(theBg, add = TRUE, maxpixels = 10^7)
mapmisc::legendBreaks("right", theColWhite, bty = "n", inset = 0)
theColRandom = mapmisc::colourScale(resTrump$data$random.mean,
  col = "Spectral", breaks = 11, style = "quantile", rev = TRUE,
  dec = 1)
theColFit = mapmisc::colourScale(resTrump$data$fitted.invlogit,
  col = "RdBu", rev = TRUE, breaks = sort(unique(setdiff(c(0,
    1, seq(0.2, 0.8, by = 0.1)), 0.5))), style = "fixed")
mapmisc::map.new(wisconsinCsubm, 0.85)
plot(resTrump$data, col = theColRandom$plot, add = TRUE,
  lwd = 0.2)
plot(theBg, add = TRUE, maxpixels = 10^7)
mapmisc::legendBreaks("topright", theColRandom)
mapmisc::map.new(wisconsinCsubm, 0.85)
plot(resTrump$data, col = theColFit$plot, add = TRUE, lwd = 0.2)
plot(theBg, add = TRUE, maxpixels = 10^7)
mapmisc::legendBreaks("topright", theColFit)

```