Introduction

These are lecture/lab notes for Chapter 6 in Spatial Regression Models. You will find all necessary information, including R-code, to replicate the figures and maps found in this chapter. A few of the maps were obtained from public repositories (Wiki and others), but what you find here will allow replication of the bulk of chapter 6. This is intended to provide code as well as a didactic repository.

Code appears below as it is in the unadorned code files. Comments are shown with one poundsign; messages from the execution of this code is flagged with two poundsigns.

There are two goals of this material:

1. Provide replication code for the materials in Spatial Regression Models, version 2. This is hosted on https://srmbook.com
2. To elaborate a bit on what the code does to provide instruction on develop code in R that replications the material in the published volume.

This was developed on the following platform. Earlier (and later) platforms should work (as should windows and linux flavored platforms). However, some libraries may not be available for all platforms. Such is life.

If you have any feature requests or find bugs herein, please do not hesitate to pass them along to the authors (michael.don.ward at gmail.com and ksg at essex.ac.uk).

```r

# R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
# Copyright (C) 2017 The R Foundation for Statistical Computing
# Platform: x86_64-apple-darwin15.6.0 (64-bit)

rm(list = ls())
library(cshapes)
library(wbstats)
p5 <- read.csv("rawdata/p5.csv")
wmap <- cshp(date = as.Date("2015-01-01"),useGW=TRUE) # get map from CSHAPES
wmap$ISO1AL3 <- as.character(wmap$ISO1AL3)
wmap$ISO1AL3[wmap$CNTRY_NAME == "Kosovo"] <- c("RKS")
wmap$ISO1AL3 <- as.factor(wmap$ISO1AL3)
wmap$data$id <- as.character(rownames(wmap$data))
wmap.df <- wmap$data
wmap$data$democ <- p5$democ[match(wmap$data$ISO1AL3, p5$scode)]
wmap$data$autoc <- p5$autoc[match(wmap$data$ISO1AL3, p5$scode)]
wmap$data$polity <- wmap$data$democ - wmap$data$autoc
# for convenience assign 0 to NA
wmap$data$polity[is.na(wmap$data$polity)] <- 0

# now get GDP per capita data
library(wbstats)
gdp <- wb(indicator = c("NY.GDP.PCAP.CD"),
          startdate = 2014, enddate = 2014,country="countries_only")
```
```r
# join the GDP data to the map data.frame
wmap$data$gdp <- gdp$value[match(wmap.df$ISO1AL2, gdp$iso2c)]
wmap$data$gdp[53]<-6.36647
wmap$data$gdp[67]<-11.99541
wmap$data$gdp[134]<-6.920672
wmap$data$gdp[176]<-10.02398
wmap$data$gdp[181]<-9.123365
wmap$data$gdp[184]<-7.438384
wmap$data$gdp[188]<-8.119994
# Hacked from the interwebs
# 53 Eritrea = 582 6.36647
# 67 Monaco = 162009 11.99541
# 134 North Korea = 1013 6.920672
# 176 Taiwan = 22561 10.02398
# 181 Libya = 9167 9.123365
# 184 Syria = 1700 7.438384
# 188 Kosovo = 3661 8.119994

# now get distance matrix
wmap.dist <- distmatrix(date = as.Date("2015-01-01"))

# recode distance data to be binary, for distance < 400km
w.dist<-wmap.dist
w.dist[w.dist<=400]<-1
w.dist[w.dist>400]<-0
# row standardize
w.dist.rstd<-w.dist/rowSums(w.dist)
# 0 on diagonal
diag(w.dist.rstd)<-0

which(rowSums(w.dist.rstd)==0)

395 402 590 591 781 920 950 970 971 972 973 983 986 987 990 83 84 129 130 170 184 187 188 189 190 191 192 193 194 195

# create spatial lag of polity. round result
wmap$data$sp.lag.polity<- round((w.dist.rstd)%*%(wmap$data$polity))

# create spatial lag of polity. round result
wmap$data$sp.lag.gdp<- round((w.dist.rstd)%*%(wmap$data$gdp))
# for convenience assign 0 to NA
wmap$data$polity[is.na(wmap$data$polity)]<- 0
# order map dataframe on GWCODE
map.df<-wmap$data[order(wmap$data$GWCODE),]

# get rid of isolates
omits<-which(rowSums(w.dist.rstd)==0)

# get rid of isolates
map.df<-map.df[-omits,]
w.d.rstd<-w.dist.rstd[-omits,-omits]
w.dd.rstd<-w.d.rstd/rowSums(w.d.rstd)
rowSums(w.dd.rstd)
```

library(spdep)
listw.obj <- mat2listw(w.dd.rstd)
ols.fit <- glm(polity ~ gdp + sp.lag.gdp, data=map.df)
#summary(ols.fit)
#moran.test(resid(ols.fit),listw.obj)
sar.fit <- lagsarlm(polity ~ gdp, data=map.df, listw.obj,type="Durbin")
#summary(sar.fit)
#moran.test(resid(sar.fit),listw.obj)

library(stargazer)
stargazer(ols.fit, type = "latex", star.char = NULL, nobs=TRUE, digits=2, star.cutoffs = NA, notes.append = FALSE, notes = "OLS", dep.var.caption = "Polity Score [10 to 10]"
)

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu % Date and time: Mon, Jan 22, 2018 - 07:45:44

Table 1:

<table>
<thead>
<tr>
<th>Polity Score [10 to 10]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>polity</td>
<td></td>
</tr>
<tr>
<td>gdp</td>
<td>0.86 (0.32)</td>
</tr>
<tr>
<td>sp.lag.gdp</td>
<td>0.23 (0.21)</td>
</tr>
<tr>
<td>Constant</td>
<td>−5.69 (2.89)</td>
</tr>
</tbody>
</table>

Observations 180
Log Likelihood −587.86
Akaike Inf. Crit. 1,181.71

Note: OLS

stargazer(sar.fit, type = "latex", star.char = NULL, nobs=TRUE, digits=2, star.cutoffs = NA, notes.append = FALSE, notes = "Spatial Durbin", dep.var.caption = "Polity Score [10 to 10]"
)

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu % Date and time: Mon, Jan 22, 2018 - 07:45:48

This result is not identical to that in the book, owing to imputed values.
<table>
<thead>
<tr>
<th></th>
<th>Polity Score [10 to 10]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>polity</td>
</tr>
<tr>
<td>gdp</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>(0.47)</td>
</tr>
<tr>
<td>lag.gdp</td>
<td>−0.21</td>
</tr>
<tr>
<td></td>
<td>(0.61)</td>
</tr>
<tr>
<td>Constant</td>
<td>−2.12</td>
</tr>
<tr>
<td></td>
<td>(3.05)</td>
</tr>
<tr>
<td>Observations</td>
<td>180</td>
</tr>
<tr>
<td>Log Likelihood</td>
<td>−567.96</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>30.45</td>
</tr>
<tr>
<td>Akaike Inf. Crit.</td>
<td>1,145.93</td>
</tr>
<tr>
<td>Wald Test</td>
<td>42.95 (df = 1)</td>
</tr>
<tr>
<td>LR Test</td>
<td>38.56 (df = 1)</td>
</tr>
</tbody>
</table>

*Note:* Spatial Durbin