Q3.R

jing

Mon Feb 1 04:40:56 2016

```
# DataIncubator Challenge Q3
library(ggmap)
```

```
## Loading required package: ggplot2
```

```
library(XLConnect)
```

```
## Loading required package: XLConnectJars
## XLConnect 0.2-11 by Mirai Solutions GmbH [aut],
##   Martin Studer [cre],
##   The Apache Software Foundation [ctb, cph] (Apache POI, Apache Commons
##     Codec),
##   Stephen Colebourne [ctb, cph] (Joda-Time Java library)
## http://www.mirai-solutions.com ,
## http://miraisolutions.wordpress.com
```

```
library(ggplot2)
library(gridExtra)
library(nlme)
library(mgcv)
```

```
## This is mgcv 1.8-7. For overview type 'help("mgcv-package")'.
```

```
library(geoR)
```

```
## ----------------------------------------------------------
##  Analysis of Geostatistical Data
##  For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
##  geoR version 1.7-5.1 (built on 2015-04-15) is now loaded
## ----------------------------------------------------------
```

```
library(fields)
```

```
## Loading required package: spam
## Loading required package: grid
## Spam version 1.2-1 (2015-09-30) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
##
## Attaching package: 'spam'
##
```

```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
##
## Loading required package: maps
##
##  # ATTENTION: maps v3.0 has an updated 'world' map.        #
##  # Many country borders and names have changed since 1990. #
##  # Type '?world' or 'news(package="maps")'. See README_v3. #
##
##
##
## Attaching package: 'fields'
##
## The following object is masked from 'package:maps':
##
##     ozone
```

```
library(spdep)
```

```
## Loading required package: sp
## Loading required package: Matrix
```

```
library(maptools)
```

```
## Checking rgeos availability: TRUE
```

```
library(spgwr)
```

```
## NOTE: This package does not constitute approval of GWR
## as a method of spatial analysis; see example(gwr)
```

```
work_dir="/home/jing/Temp/Q3/"
setwd(work_dir)

# reading restaurants data downloaded from public website
restaurants=readWorksheetFromFile("Active_restaurant_heat_map.xlsx", sheet=1)
restaurants$zip=sapply(strsplit(restaurants$ZIP.CODE, "-"), "[[", 1)
restaurants=restaurants[restaurants$LOCATION!="" & !is.na(restaurants$LOCATION.START.DATE),]
restaurants$establish=as.numeric(sapply(strsplit(as.character(restaurants$LOCATION.START.DATE), "-"), "
restaurants$latitude=as.numeric(sapply(strsplit(restaurants$LOCATION, ",|\\(|\\)"), "[[", 2))
restaurants$longitude=as.numeric(sapply(strsplit(restaurants$LOCATION, ",|\\(|\\)"), "[[", 3))

# LA map from API
la=get_map(location="Los Angeles", zoom=10, color="bw", maptype="toner")
```

```
## maptype = "toner" is only available with source = "stamen".
## resetting to source = "stamen"...
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=Los+Angeles&zoom=10&size=640x640
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Los%20Angeles&sensor=
## Map from URL : http://tile.stamen.com/toner/10/174/407.png
```

```
## Map from URL : http://tile.stamen.com/toner/10/175/407.png
## Map from URL : http://tile.stamen.com/toner/10/176/407.png
## Map from URL : http://tile.stamen.com/toner/10/174/408.png
## Map from URL : http://tile.stamen.com/toner/10/175/408.png
## Map from URL : http://tile.stamen.com/toner/10/176/408.png
## Map from URL : http://tile.stamen.com/toner/10/174/409.png
## Map from URL : http://tile.stamen.com/toner/10/175/409.png
## Map from URL : http://tile.stamen.com/toner/10/176/409.png
## Map from URL : http://tile.stamen.com/toner/10/174/410.png
## Map from URL : http://tile.stamen.com/toner/10/175/410.png
## Map from URL : http://tile.stamen.com/toner/10/176/410.png
```

```
# distribution of all active restaurants in LA
active_now=ggmap(la)
active_now=active_now+geom_point(data=restaurants, aes(x=longitude, y=latitude), size=1, col="red")+ggt

# newly opened restaurants in LA since 2010
active_2015=ggmap(la)
active_2015=active_2015+geom_point(data=restaurants[restaurants$establish==2015,], aes(x=longitude, y=la

png("Established_restaurants_LA.png", width=1800, height=900)
grid.arrange(active_now, active_2015, ncol=2)
```

```
## Warning: Removed 45 rows containing missing values (geom_point).

## Warning: Removed 4 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
# counting number of restaurants by zip code
zip_act=aggregate(restaurants$BUSINESS.NAME, by=list(c(restaurants$zip)), length)
colnames(zip_act)=c("zipcode", "ActRest")
zip_new_2015=aggregate(BUSINESS.NAME~zip, data=subset(restaurants, establish==2015), length)
colnames(zip_new_2015)=c("zipcode", "NewRest2015")
ActRest=merge(zip_act, zip_new_2015, by="zipcode", all=TRUE, sort=FALSE)
ActRest[is.na(ActRest)]=0

# reading demographic data scrapped from LA city website
lazipdem=readShapeSpatial("./LAZipDem/LAZipDem.shp", ID="OBJECTID")
lazipdem=merge(lazipdem, ActRest, by.x="ZIP", by.y="zipcode", all.x=TRUE, sort=FALSE)
```

```
## Warning in .local(x, y, ...): 71 records in y cannot be matched to x
```

```
lazipdem$rest_pt=100*lazipdem$NewRest2015/lazipdem$ActRest
lazipdem$rest_pt=ifelse(is.na(lazipdem$rest_pt), 0, lazipdem$rest_pt)
lazipdem$NewRest2015=ifelse(is.na(lazipdem$NewRest2015), 0, lazipdem$NewRest2015)

# creating regressive correlation across adjacent regions
```

```
rook=poly2nb(lazipdem, queen=FALSE)
rook.w=nb2listw(rook)
# Moran's I test
moranRest.rook=moran.test(lazipdem$rest_pt, rook.w)
moranRest.rook
```

```
##
##  Moran's I test under randomisation
##
## data:  lazipdem$rest_pt
## weights: rook.w
##
## Moran I statistic standard deviate = -0.1592, p-value = 0.5632
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      -0.010986805      -0.004587156       0.001615876
```

```
png("Morans Scatterplot.png", width=1600, height=600)
par(mfrow=c(1,2))
mscat.rest_pt=moran.plot(lazipdem$rest_pt, rook.w, zero.policy=TRUE, pch=16, col="black", quiet=FALSE, l
```

```
## Potentially influential observations of
##   lm(formula = wx ~ x) :
##
##     dfb.1_ dfb.x dffit   cov.r   cook.d hat
## 67  -0.01  -0.09 -0.10    1.03_*  0.01   0.03
## 82   0.17  -0.06  0.17    0.96_*  0.01   0.01
## 86   0.07  -0.39 -0.40_*  1.33_*  0.08   0.24_*
## 90   0.18  -0.06  0.18    0.95_*  0.02   0.01
## 107  0.07  -0.39 -0.40_*  1.33_*  0.08   0.24_*
## 109  0.22  -0.07  0.22    0.93_*  0.02   0.01
## 110  0.22  -0.07  0.22    0.93_*  0.02   0.01
## 111  0.22  -0.07  0.22    0.93_*  0.02   0.01
## 113 -0.01   0.15  0.16    1.07_*  0.01   0.06_*
## 117  0.36  -0.12  0.36_*  0.81_*  0.06   0.01
## 196  0.07  -0.39 -0.40_*  1.33_*  0.08   0.24_*
## 214  0.25  -0.08  0.25    0.91_*  0.03   0.01
## 215  0.01  -0.13 -0.14    1.07_*  0.01   0.06_*
## 216  0.17  -0.06  0.17    0.96_*  0.01   0.01
## 217  0.36  -0.12  0.36_*  0.81_*  0.06   0.01
```

```
mscat.NewRest2015=moran.plot(lazipdem$NewRest2015, rook.w, zero.policy=TRUE, pch=16, col="black", quiet=
```

```
## Potentially influential observations of
##   lm(formula = wx ~ x) :
##
##     dfb.1_ dfb.x   dffit   cov.r   cook.d hat
## 5   -0.04   0.29    0.30_*  1.07_*  0.04   0.06_*
## 6    0.11  -0.50   -0.51_*  1.12_*  0.13   0.11_*
## 9    0.01   0.50    0.55_*  0.93_*  0.15   0.02
## 11   0.18  -0.83   -0.85_*  1.08_*  0.35   0.11_*
```

```
## 18   0.13   0.05    0.17    0.96_*  0.01   0.01
## 19  -0.01   0.07    0.07    1.09_*  0.00   0.07_*
## 20   0.00   0.00    0.00    1.03_*  0.00   0.02
## 24   0.04  -0.45   -0.48_*  1.00    0.11   0.04_*
## 26   0.02   0.26    0.30_*  0.99    0.04   0.02
## 27   0.31  -1.17_* -1.19_*  1.15_*  0.69   0.18_*
## 28   0.04   0.25    0.30_*  0.98    0.04   0.02
## 36   0.05   0.30    0.35_*  0.96_*  0.06   0.02
## 43   0.13  -0.83   -0.86_*  0.98    0.36   0.06_*
## 49   0.13   0.18    0.28    0.93_*  0.04   0.01
## 59   0.41  -0.17    0.41_*  0.78_*  0.07   0.01
## 61   0.25   0.02    0.28    0.87_*  0.04   0.00
## 64   0.18  -0.07    0.18    0.96_*  0.02   0.01
## 101  0.17  -0.07    0.17    0.97_*  0.01   0.01
## 180  0.02  -0.27   -0.29_*  1.02    0.04   0.04_*
```

```
dev.off()
```

```
## pdf
##   2
```

```
# population change
demographics=read.csv("demographics.csv", header=TRUE)
lazipdem=merge(lazipdem, demographics[c("ZIP", "income2000", "income2010")], by="ZIP", all.x=TRUE, sort=
lazipdem$popchange=100*(lazipdem$POP2010b-lazipdem$POP2000b)/lazipdem$POP2000b

lazipdem$popchange=ifelse(lazipdem$popchange=="NaN",0, lazipdem$popchange)
lazipdem$income2010=ifelse(is.na(lazipdem$income2010),0, lazipdem$income2010)
lazipdem$white_pt=100*lazipdem$WHITE/lazipdem$POP2010b
lazipdem$white_pt=ifelse(is.na(lazipdem$white_pt),0, lazipdem$white_pt)
lazipdem$hispanic_pt=100*lazipdem$HISPANIC/lazipdem$POP2010b
lazipdem$hispanic_pt=ifelse(is.na(lazipdem$hispanic_pt),0, lazipdem$hispanic_pt)
lazipdem$asian_pt=100*lazipdem$ASIAN/lazipdem$POP2010b
lazipdem$asian_pt=ifelse(is.na(lazipdem$asian_pt),0, lazipdem$asian_pt)

# auto regressive model
queen=poly2nb(lazipdem, queen=TRUE)
col.palette=colorRampPalette(c("white", "orange", "red"), space="rgb")
col.ramp=col.palette(length(seq(0,1,0.2)))

sar=spautolm(NewRest2015~POP2010b+popchange+income2010+white_pt+hispanic_pt+asian_pt, data=lazipdem, nb
summary(sar)
```

```
##
## Call:
## spautolm(formula = NewRest2015 ~ POP2010b + popchange + income2010 +
##     white_pt + hispanic_pt + asian_pt, data = lazipdem, listw = nb2listw(queen),
##     method = "eigen")
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -8.567323 -1.226935 -0.585728  0.008599 21.148487
##
```

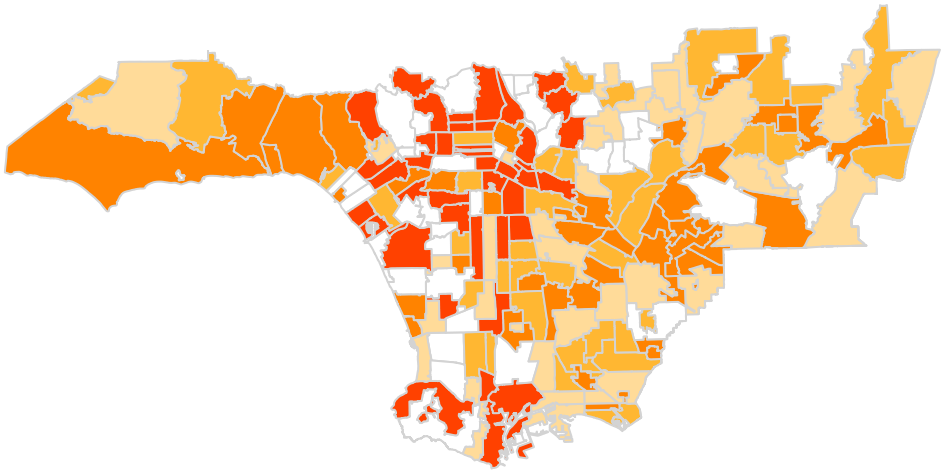
```
## Coefficients:
##                Estimate  Std. Error z value Pr(>|z|)
## (Intercept)  1.0218e+00  1.1362e+00  0.8993  0.36849
## POP2010b     1.6875e-05  1.1999e-05  1.4063  0.15962
## popchange   -4.4436e-03  2.0475e-03 -2.1702  0.02999
## income2010  -1.1581e-05  8.3800e-06 -1.3819  0.16699
## white_pt     1.7196e-02  1.6170e-02  1.0634  0.28760
## hispanic_pt -1.4005e-02  1.4860e-02 -0.9425  0.34595
## asian_pt     3.7265e-02  2.0916e-02  1.7816  0.07481
##
## Lambda: 0.605 LR test value: 61.264 p-value: 4.996e-15
## Numerical Hessian standard error of lambda: 0.062317
##
## Log likelihood: -574.8075
## ML residual variance (sigma squared): 10.234, (sigma: 3.1991)
## Number of observations: 219
## Number of parameters estimated: 9
## AIC: 1167.6
```

```
moran.test(sar$fit$residuals, listw=nb2listw(queen), zero.policy = T)
```

```
##
##  Moran's I test under randomisation
##
## data:  sar$fit$residuals
## weights: nb2listw(queen)
##
## Moran I statistic standard deviate = -1.565, p-value = 0.9412
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##      -0.070070208      -0.004587156       0.001750844
```

```
lazipdem$sar.res=sar$fit$residuals
sar.breaks=quantile(sar$fit$residuals, seq(0,1,0.2))
plot(lazipdem, border="lightgray", col=col.ramp[findInterval(lazipdem$sar.res, sar.breaks, all.inside=T
```

SAR Residuals



```
##
## calling INFO from Yelp API
# library(httr)
# library(httpuv)
# library(jsonlite)

# consumerKey = "####"
# consumerSecret = "####"
# token = "####"
# token_secret = "####"

# authoriztion
# myapp = oauth_app("YELP", key=consumerKey, secret=consumerSecret)
# sig=sign_oauth1.0(myapp, token=token, token_secret=token_secret)

# limit = 10

# yelpurl=paste("http://api.yelp.com/v2/search/?limit=",limit,"&location=Los+Angeles&term=restaurants&c

# locationdata=GET(yelpurl, sig)
# locationdataContent = content(locationdata)
# locationdataList=jsonlite::fromJSON(toJSON(locationdataContent))
# head(data.frame(locationdataList))
```