Lab 9
Spatial Statistics

Learning Objectives:

- ✓ Deepen understanding of R for statistics
- ✓ Draw maps in R
- ✓ Display geographically weighted regression on an example data set

1 Displaying maps

A shape file is actually a set of files with extensions .shp, dbf, and shx.

Download file shapefile for eire from your Webcourses area.
Working directory here has been set to C://My-R-Dir. Please adjust this value down to your specific case.

```
   # Load the eire.shp file into R.
library(spdep)
library(maptools)
library(RColorBrewer)
library(classInt)

setwd("C://My-R-Dir")
eireMap <- readShapePoly("eire.shp"[1],ID="names", proj4string=CRS("+proj=utm +zone=30
+units=km"))
names(eireMap)
factor(eireMap$pale)
eireMap$names
#The question mark gets help on a topic
?eire

# Some colours for the counties
colors = c("#F1EEF6", "#D4B9DA", "#C994C7", "#DF65B0", "#DD1C77", "#980043","#F1EEF6", "#D4B9DA",
"#C994C7", "#DF65B0", "#DD1C77", "#980043","#F1EEF6", "#D4B9DA", "#C994C7", "#DF65B0", "#DD1C77",
"#980043","#F1EEF6", "#D4B9DA", "#C994C7", "#DF65B0", "#DD1C77", "#980043","#F1EEF6", "#D4B9DA")
plot(eireMap, col=colors[eireMap$names])

# colour in the Pale. The default 0,1 will give black and white
plot(eireMap)
color <- eireMap$pale+3
plot(eireMap, col=color)

# Get the neighbours of each county.
eire.nb <- poly2nb(eireMap)
# Examine contiguity
summary(eire.nb)
plot(eireMap)
plot(eire.nb, coordinates(eireMap), add=TRUE)
#Print county names
text(coordinates(eireMap), labels=as.character(eireMap$names), cex=0.4)


# Column A represents the percentage of sample with blood group A
# See http://en.wikipedia.org/wiki/Blood_type_distribution_by_country
You can investigate the data in eire with:
summary(eireMap$A)
res <- eireMap$A
# A five-number summary description about a set of observations.
brks <- round(fivenum(eireMap$A), digits=2)
```

Lab 9
Spatial Statistics

```
cols <- rev(heat.colors(4))
plot(eireMap, col=cols[findInterval(res, brks, all.inside=TRUE)])
title(main="Percentage with blood group A")
legend(x=c(-300, 70), y=c(6120, 6050), legend=leglabs(brks), fill=cols, bty="n")
text(coordinates(eireMap), labels=as.character(eireMap$names), cex=0.5)
```

Identifying neighbours.

```
# Get the neighbours of each county.
eire.nb <- poly2nb(eireMap)
plot(eireMap)
plot(eire.nb, coordinates(eireMap), add=TRUE)
```

Plot Counties in more detail.

The file county_region is available on webcourses

```
county <- readShapePoly("county_REGION.shp")
names(county)
county$NAME
plot(county)
text(coordinates(county), labels=as.character(county$NAME), cex=0.5)
```

Note that readShapePoly is from maptools package.

2 Pie Charts

We make a pie chart relating retail sales (RETSALE) to income (INCOME).

```
library(plotrix)
plot(eireMap)
floating.pie(coordinates(eireMap)[1],coordinates(eireMap)[1,2],c(eireMap$RETSALE[1],
eireMap$INCOME[1]),radius=10,  col=c("#ff0000","#80ff00","#00ffff","#44bbff","#8000ff"))
mypercent <- paste(round(100*(eireMap$RETSALE/eireMap$INCOME),1),"%")
text(coordinates(eireMap)+10, labels=mypercent, cex=0.5)


## Here is a loop from 1 to 26
for(i in 1:26){cat("iteration number", "->",i,"\n")}

## make a map with pie chart for every county
for(i in 1:26) {
floating.pie(coordinates(eireMap)[i],coordinates(eireMap)[i,2],c(eireMap$
RETSALE[i],eireMap$INCOME[i]),radius=12,
col=c("#ff0000","#80ff00","#00ffff","#44bbff","#8000ff"))
text(coordinates(eireMap)+10, labels=mypercent, cex=0.5)}
```

To see a reasonable size map use the zoom tool:

3 Display CSO data

Use the cso_eds from your Webcourses folder.
Plot new born male children.
Read in shape file

```
dub.eds <- readShapePoly("cso_eds_data")
```

If not already loaded then load the colour library

Lab 9
Spatial Statistics

```
library(RColorBrewer)
```

Set some colours using 8 intervals

```
pop8 <- brewer.pal(8,'Set2')
```

To label each Electoral Divisions use the following command.
The next command places the ED name at the centre of the EDs polygon

```
list1 = list("sp.text", coordinates(dub.eds)+1, as.character(dub.eds$GEOGDESC)
,col="red",font=2,cex=0.5)
```

The next command makes a choropleth map using default colours

```
spplot(dub.eds, "T1_1AGE0M",  main='Male Children')
```

Or we can use the set of 8 colours defined in pop8.

```
spplot(dub.eds, "T1_1AGE0M", col.regions=pop8, main='Male Children')
```

You can zoom and/or save the map as a PNG file.

In general we can get the ranges for thematic colouring as follows:

```
lower = min(dub.eds@data$T1_1AGE0M)
upper = max(dub.eds@data$ T1_1AGE0M)
intrv = (lower+upper)/8
```

Here are some useful computations.

Sum first three male age columns (age 0 to 3 years).

```
male.zero2three <- paste("The sum of males age 0-3 for ",dub.eds$GEOGDESC," is ",dub.eds$T1_1AGE0M
+ dub.eds$T1_1AGE1M + dub.eds$T1_1AGE2M)
```

To see result type.:

```
male.zero2three
```

You can get the column numbers and names as follows:

```
names(dub.eds@data)
```

column number=40, column name =T1_1AGE0F (youngest females)
column number=73, column name =T1_1AGETF (total females)
column number=74, column name = T1_1AGEGE_.1 (oldest females)
Two ways of accessing columns, by number or by name

```
head(dub.eds@data[,40]) == head(dub.eds$T1_1AGE0F) #youngest
head(dub.eds@data[,74]) == head(dub.eds$T1_1AGETF) #total
```

Select first three columns, by number or by name

```
dub.eds@data[,1:3]
subset(dub.eds@data, select = SP_ID:GEOGTYPE)
```

Select the first six rows of the age columns for all females.

```
head(subset(dub.eds@data, select = T1_1AGE0F:T1_1AGEGE_.1))
```

Get the sum of children in one age group of all EDs (i.e. sum one column):

```
sum(dub.eds$T1_1AGE0M,na.rm = TRUE)  # na.rm = remove missing data
```

Get the sum of a range of column.
The age columns for females start at column 40 (T1_1AGE0F =youngest) up to 73 (T1_1AGEGE_.1=oldest) with the total in 74 (T1_1AGETF).

```
colSums(dub.eds@data[,c(40:73)])
colSums(subset(dub.eds@data, select = T1_1AGE0F:T1_1AGEGE_.1))
```

Lab 9
Spatial Statistics

Here we add all the females in all age groups and check that the calculated sum is equal to the total provided by the CSO:

```
rowSums(subset(dub.eds@data, select = T1_1AGE0F:T1_1AGEGE_.1)) == dub.eds$T1_1AGETF
```

OR just the first six.

```
head(rowSums(subset(dub.eds@data, select = T1_1AGE0F:T1_1AGEGE_.1))) ==head(dub.eds$T1_1AGETF)
```

The function head returns the first six elements.

For all 161 EDs does Total Males + Total Females = Population Total:

```
dub.eds$T1_1AGETM +  dub.eds$T1_1AGETF  == dub.eds$T1_1AGETT
```

4 Run Moran's I on Dublin EDs CSO data

```
dub.nb <- poly2nb(dub.eds,queen = TRUE)
dub.I <- moran.test(dub.eds$T1_1AGE0M, nb2listw(dub.nb))
moran.plot(dub.eds$T1_1AGE0M,labels=dub.eds$GEOGDESC,listw=nb2listw(dub.nb))
title(paste("Map 1: Moran's I =",as.character(round(dub.I$estimate[1],2))))
```

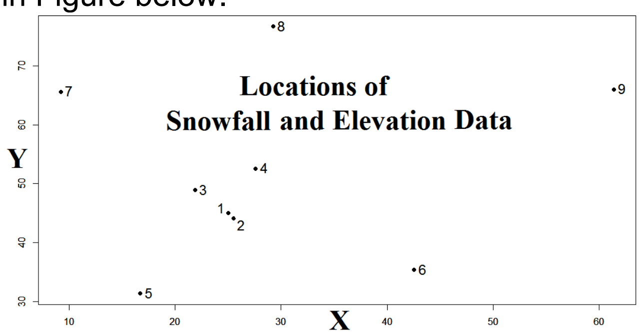
5 Geographically Weighted Regression

For the rest of the lab we will assume My-R-Dir ignoring the drive.

```
#0)A GWR example
#1)An example of coefficient of determination (r2)
#2)Plot and explore education and car ownership for Dublin
#3)Perform ordinary regression on the data
#4)Perform spatially weighted regression
```

```
#0)A GWR example
```

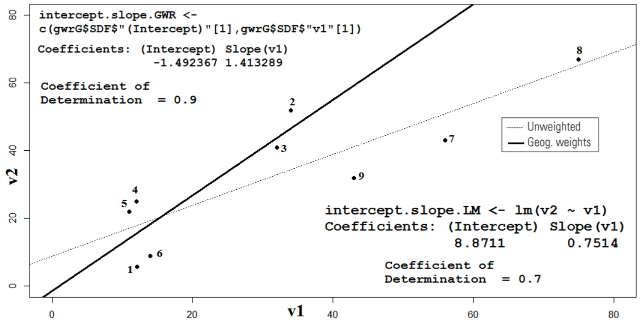
The amount of snowfall and the elevation was recorded at nine locations, shown in Figure below.



Below is the R code for the data and GWR

```
library(spgwr)
v1 <- c(12,34,32,12,11,14,56,75,43)
v2 <- c(6, 52, 41, 25, 22, 9,43, 67, 32)
id <- c(1,2,3,4,5,6,7,8,9)
x <- c(25.0, 25.51, 21.87, 27.6, 16.69, 42.52, 9.2,  29.23, 61.37)
y <- c(45.0, 44.14, 48.9, 52.57, 31.33, 35.35, 65.65,76.72, 66.01)
cds <- SpatialPoints(cbind(x,y))
ds <- data.frame(cbind(id,v2,v1))
sp <- SpatialPointsDataFrame(cds,ds)
gwrG <- gwr(v2 ~ v1, data = sp, bandwidth = 10)
plot(v1,v2); abline(lm(v2~v1), lty=3)
abline(gwrG$SDF$"(Intercept)"[1], gwrG$SDF$"v1"[1],lwd=3)
text(v1, v2, labels=sp$id, cex= 1.5, pos=3)
```

Lab 9
Spatial Statistics



The snowfall is represented by the vector v2, while the elevation is represented by the vector v1.

Here is an explanation of the code

Lines 2-6 data, ids, and locations

Line 7 makes spatial points from raw coordinates

Line 8 makes dataframe using the non-spatial data

Line 9 Combines the spatial and non-spatial data

Line 10 computes the GWR with v2 as the dependent variable, v1 as the independent variable

Line 11 the OLS line is computed and plotted

Line 12 the GWR for the first point (id=1) is plotted.

Line 12 the labels are plotted.

Interpretation: The coefficient of determination (R2) is nearer 1 for GWR which implies a better model (see below for an explanation of R2)

```
#1)The coefficient of determination R2 is used in the prediction of future outcomes on the basis
of other related information. R2 is the local R-squared of a statistical model. R2 shows how well,
locally, the regression model manages to predict the dependent variable
#It is the proportion of variability in a data set that is accounted for by the statistical model.
#It provides a measure of how well future outcomes are likely to be predicted by the model.
# Anything above .7 is good
# An example of coefficient of determination (r2)
v1 <- c(12,34,32,12,11,14,56,75,43)
# Snowfall z, response
v2 <- c(6, 52, 41, 25, 22, 9,43, 67, 32)

v1 <- c(12,34,32,12,11,14,56,75,43)
# Snowfall z, response
v2 <- c(6, 52, 41, 25, 22, 9,43, 67, 32)
snow.lm <- lm (v2 ~ v1)
summary(snow.lm)$r.squared
#[1] 0.7324664


#2)Plot education and car ownership for Dublin
#Load the required Libraries, you may need to install some packages (e.g.
install.packages('spgwr'))

library(sp)
library(maptools)
gpclibPermit()
library(spdep)
library(spgwr)
library(RColorBrewer)
```

Lab 9
Spatial Statistics

```
# You can find a object in your R workspace with ls() and remove it with remove(objectname)
#Load in the dublin.eds data from lab 4 into R.
setwd("C:\\My-R-Dir\\")
dub.eds <- readShapePoly("cso_eds_data")


#The structure dub.eds is a SpatialPolygonsDataFrame.
#It should include all the CSO data. Explore dubl.eds:
is(dub.eds)
names(dub.eds)
is(dub.eds)
slotNames(dub.eds)
names(dub.eds@data)
## case sensitive
dub.eds$T1_1AGETM
dub.eds$T1_1AGETF
## Add male and female, it should equal grand total for each of the 162 EDs
(dub.eds$T1_1AGETM + dub.eds$T1_1AGETF) == dub.eds$T1_1AGETT

#If needed you can save your data when you make changes.
#It is a good idea to increment file version as follows:
# writePolyShape(dub.eds,"C:\\My-R-Dir\\"cso_eds_data_V1.shp")
#The above command saves dub.eds to file named cso_eds_data_V1.shp


#First we make some thematic maps containing the educational information.
#We will consider three categories: Primary,Degree, and  PhDs.
#Examine each of these maps
## MAP 1
plot(dub.eds)
text(coordinates(dub.eds),labels= dub.eds@data$T10_4_PM,cex=0.5)
title("Primary School")
## MAP 2
plot(dub.eds)
text(coordinates(dub.eds),labels= dub.eds@data$T10_4_HDPQ,cex=0.5)
title("Degree")
## MAP 3
plot(dub.eds)
text(coordinates(dub.eds),labels= dub.eds@data$T10_4_DM
 ,cex=0.5)
title("PhD")


# Coloured map for Primary Education
# Set some colours and
pop8 <- brewer.pal(8,'Set2')

#Get the ranges for thematic colouring for Primary Education
lower = min(dub.eds@data$"T10_4_PM")
upper = max(dub.eds@data$"T10_4_PM")
intrv = (lower+upper)/8

#Now plot the map with the above intervals:
spplot(dub.eds, "T10_4_PM",main='Dublin Primary Education')



#Check individual theme
dub.eds@data$"T10_4_PM"


#Check individual areas
dub.eds@data$GEOGDESC=="Arran Quay A"
ifelse(dub.eds@data$GEOGDESC=="Arran Quay A",dub.eds$T10_4_PM,"not found")
```

Lab 9
Spatial Statistics

```
#You do the same for other data in dublin.eds

#Do the same for the Car data
#Examine each of these maps
plot(dub.eds)
text(coordinates(dub.eds),labels= dub.eds$T15_1_1C,cex=0.5)
title("One Car")

plot(dub.eds)
text(coordinates(dub.eds),labels= dub.eds$T15_1_2C,cex=0.5)
title("Two Cars")

plot(dub.eds)
text(coordinates(dub.eds),labels= dub.eds$T15_1_3C,cex=0.5)
title("Three Cars")

#Examine each of these thematic maps
# First Primary Education
# Set some colours and
pop8 <- brewer.pal(8,'Set2')


#Now plot the map with the above intervals, spplot:
Lout = list("sp.text", coordinates(dub.eds), as.character(dub.eds$"T15_1_1C")
,col="red",font=2,cex=1)
spplot(dub.eds, "T15_1_1C", col.regions=pop8,main='One Car',sp.layout=Lout)

## Get max and min of one car ownership
paste(dub.eds$GEOGDESC, dub.eds$T15_1_1C)
ifelse(dub.eds$T15_1_1C == max(dub.eds$T15_1_1C), paste(dub.eds$GEOGDESC"),"no")
ifelse(dub.eds$T15_1_1C == min(dub.eds$T15_1_1C),paste(dub.eds$GEOGDESC),"no")



#3)Perform ordinary regression on the data

#Recall how regression was calculated in lab1.
#Experiment with the relationship between degree education and car ownership.
lm(dub.eds$T15_1_1C ~ dub.eds$T10_4_HDPQ)
plot(dub.eds$T15_1_1C, dub.eds$T10_4_HDPQ)
abline(lm(dub.eds$T10_4_HDPQ ~ dub.eds$T10_4_HDPQ))



#The argument to the left of the tilde (~) is the response variable(dependent)(y-axis)
#The argument to the right of the tilde (~) is the explanatory variable (independent)(x-axis)
# Get the number of people with one car, professionals, primary education
one.car <- dub.eds@data$T15_1_1C
prof <- dub.eds@data$T9_1_PWM
prim <- dub.eds$T10_4_HDPQ


#How is Arran Quay A predicted by normal regression?
dub.eds$GEOGDESC[1]
paste(dub.eds$GEOGDESC[1],"=one.car=",one.car[1])   # 230 people with one car
paste(dub.eds$GEOGDESC[1],"=primary education=",prim[1]) #60  people with primary education
paste(dub.eds$GEOGDESC[1],"=professionals=",prof[1]) #56 professionals


# What is
LM <- lm(one.car ~ prim)

# Given the number of primary educated, how may one car people would you expect?
# Example the linear model
#lm(formula = one.car ~ prim)
#
#Coefficients:
```

```
#(Intercept)        prim
#    342.507       1.477
# One car owner predicated by model: 342.6 + 60 * 1.477 = 432
# Actual number of one car owners in Arran Quay A = 230
#
# Find the predicated value of
names(LM)
##[1] "coefficients"  "residuals"     "effects"       "rank"          "fitted.values" "assign"
## [7] "qr"             "df.residual"   "xlevels"       "call"          "terms"         "model"

LM$coefficients

LMpredicted.ownership <- LM$coefficients[1] + LM$coefficients[2] * prim[1]
paste("LM predicated =",LMpredicted.ownership[1]," actual= ",one.car[1])

##Note the linear model is printed as global in GWR.


#4)Perform spatially weighted regression
#Professsionals
bwGP <- gwr.sel(one.car ~ prof, data = dub.eds, gweight = gwr.Gauss, verbose = FALSE)
gwrGP <- gwr(one.car ~ prof, data = dub.eds, bandwidth = bwGP )
#Primary ed
bwGC <- gwr.sel(one.car ~ prim, data = dub.eds, gweight = gwr.Gauss, verbose = FALSE)
gwrGC <- gwr(one.car ~ prim, data = dub.eds, bandwidth = bwGC )

#We now repeat that technique for comparing car ownership with education the we used in the linear
model.
names(gwrGC)
names(gwrGC$SDF)
View(gwrGC$SDF@data)
#will display the names in a scrollable fashion.  Also
head(gwrGC$SDF@data)
#will show just the first few if that's good enough.

#Get the values for primary education value for Arran Quay A
GWRpredicted.ownership  <- gwrGC$SDF$X.Intercept.[1] + (gwrGC$SDF$prim[1] * prim[1])
paste("GWR predicated =",GWRpredicted.ownership," actual= ",one.car[1])
Note the Global column in the GWR output is the same as the LM.
```

Same a Linear Model

```
gwr(formula = one.car ~ prim, data = dub.eds, bandwidth = bwGC)
Kernel function: gwr.Gauss
Fixed bandwidth: 939.971
Summary of GWR coefficient estimates at data points:
                  Min.  1st Qu.   Median  3rd Qu.     Max.   Global
X.Intercept.  20.8400 186.7000 293.1000 355.3000 566.6000 342.5075
prim           0.8598   1.4820   1.9710   2.5190   6.2050   1.4766
```

```
o
#Now look at all the GWRs and the R squared.
#Where SDF stands for a SpatialPointsDataFrame or SpatialPolygonsDataFrame object with fit.points,
weights, GWR
coefficient estimates, R-squared, and coefficient standard errors in its "data" slot.

gwrGC
gwrGC$SDF@data
gwrGC$SDF@data[1]
plot(dub.eds)
text(coordinates(dub.eds),labels= gwrGC$SDF@data$localR2 ,cex=0.5)
dub.eds$localR2 <- gwrGC$SDF@data$localR2

text(coordinates(dub.eds),labels= format.default(gwrGC$SDF@data$localR2, digits = 2, justify =
"left", trim = FALSE) ,cex=0.5)
```

```
text(coordinates(dub.eds),labels= format.default(dub.eds$localR2, digits = 1, justify = "left",
trim = FALSE) ,cex=0.5)


dub.eds$localR2 <- gwrGC$SDF@data$localR2
spplot(dub.eds, "localR2", col.regions=pop8,main='Residuals')


car.lm <- lm(one.car ~ prim)

#We extract the coefficient of determination from the r.squared attribute of its summary.
summary(car.lm)$r.squared
dub.eds$localR2[1] # R2 from GWR for Arran Quay A
```