Lab 8
Introduction to R

Learning Objectives:

- ✓ Download R and RStudio
- ✓ Packages, Libraries and Directories
- ✓ Regression: Maximum Likelihood
- ✓ Correlation
- ✓ Calculating kmeans

1. Download R and RStudio

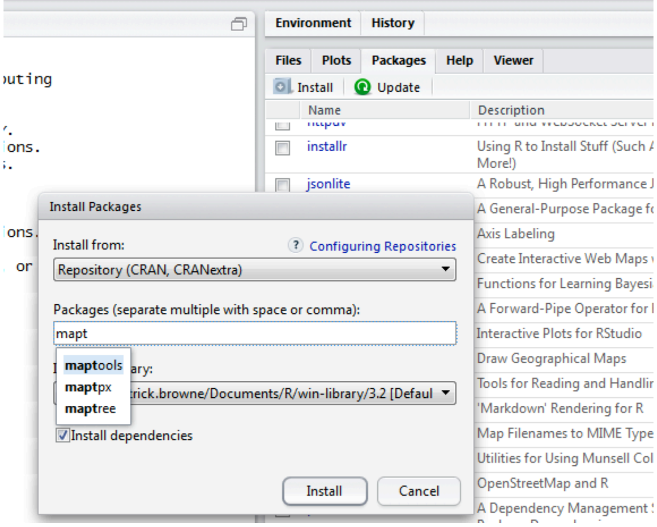
R is a stand-alone statistical package that can be used with many other systems e.g. PostgreSQL or Excel. If you wish to install RStudio on your own machine you must first install R itself. R can be downloaded from The Comprehensive R Archive. All download links necessary are in your Webcourses folder.

2. Packages, Libraries and Directories

```
# The terms directory is a synonym for folder.
# The R documentation normally refers to directories (Dir)
# The R language is extended with user-submitted packages
# On this course we are interested in packages for
#  1)basic statistics
#  2)spatial statistics

# To list available packages, type
(.packages())
# You can also check the package pane of Rstudio.
# A library is a directory where R can find installed packages.
# To see which packages are installed n your libraries, type
library()
# This command will open a new window which you can read and the close.
# To list the contents (functions and data) of a specific package, type
 ls(package:base)
# You will see a list of functions. Type the min function into R.
min(1,2)
# What does it do?

# R can be instructed to use a specific packages (i.e. load a package)
# This can be done in the Package Panel or the Console.
# For example the command library(sp) will load the spatial package.
# We will load packages later
# You can install packages from the Internet using the install tab.
```



Environment
History
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
installr
Using R to Install Stuff (Such More!)
jsonlite
A Robust, High Performance J
A General-Purpose Package fo
Axis Labeling
Create Interactive Web Maps
Functions for Learning Bayesi
A Forward-Pipe Operator for
Interactive Plots for RStudio
Draw Geographical Maps
Tools for Reading and Handlin
'Markdown' Rendering for R
Map Filenames to MIME Type
Utilities for Using Munsell Col
OpenStreetMap and R
A Dependency Management

Install Packages

Install from: Repository (CRAN, CRANextra)

Configuring Repositories

Packages (separate multiple with space or comma): mapt

maptools
maptpx
maptree

…ary: …rick.browne/Documents/R/win-library/3.2 [Defaul

☑ Install dependencies

Install   Cancel

3. Maxium likelihood

```
# A statistical model is a representation of a relationship between
# variables in the form of  mathematical equations.
# For the moment we will consider a variable as a numerical
# value that can differ from individual to individual (i.e. data).
# Variables will have an associated probability.
# A statistical variable is different from a program variable.

# Say we want a best fit (maximum likelihood) of a model to the data.
# We want a model to be an unbiased, variance minimizing estimator.
# Given the data and choice of model what values of the
# parameters that make that model most likely?
# A parameter is a numerical property of a population,
# such as its mean.

# Below is some data called x, y.
#   y is a response variable, x is an explanatory variable
# We will use a regression line as an example model.
# A regression line is a line drawn through the points
# on a scatter plot to summarise the relationship between the variables # When
it slopes down, this indicates a negative or inverse
# relationship between the variables;
# When it slopes up, a positive or direct relationship is indicated.
# The model can be written: y = a + b * x
# The model has two parameters
#    1) the intercept called a
#    2) the slope called b
# In this case we store the data in two R vectors name x and y.
# Our model
x <- c(1,3,4,6,8,9,12)
y <- c(5,8,6,10,9,13,12)
# You can view the Console.
x
```

```
View(x)
# The plot command produces a scatter plot with labelled axis.
plot(x,y,ylab="Response variable",xlab="Explanatory variable")
# You should see a scatter plot.
# You can view the Console.
View(x)
# R can calculate the maximum likelihood estimate of the intercept and slope
#  of a linear model which is:  y = 4.8 + (0.6 *x)
# We can plot the resulting best fit as a line on the existing plot.
abline(lm(y~x))
#The function lm is called Linear Model
#lm calculates the maximum likelihood of a and b
#  in the formula y = a + b * x
#The argument to the left of the tilde (~) is the response variable
#The argument to the right of the tilde (~) is the explanatory variable
#You can add a title.
title(main = "Lab 8")
# Note that when plot is called we lose previous plots.
# But title leaves the original graphic
```

4. Correlation

```
# Two continuous variables may related negatively,
# not at all or positively (-1,0,1).
# In this section will test relation with data from a file
# which can be found in thg course Google Drive.
data <- read.table("C:\\My-R-Dir\\twosample.txt",header=T)
# Attach allows components of frame can be extracted using
# frame$name. For example
data$x
# It is always useful to view the data in a scatter plot.
plot(data$x, data$y)
# Now calculate the correlation coefficient.
# Which is a measure of the strength of the
# linear relationship between two variables
cor(data$x,data$y)
cor(data$x,data$y) == cor(data$y,data$x)
# If we used just cor(x,y) we might get the x and y
# We can check the defined variables with the ls command:
 ls()
# and see what these commands do.
a<- 1:10
mean(a)
var(a)
sd(a)
# Getting help
help(sd)
?sd
??standard
help(package=sp)
?findInterval
x <- 2:18
v <- c(5, 10, 15)
```

```
# create two bins [5,10) and [10,15)
cbind(x, findInterval(x, v))
```

5. Calculating kmeans clusters

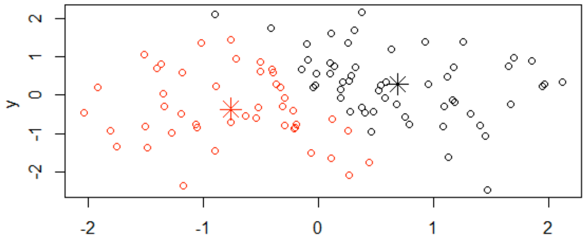
K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space which can form Voronoi cells. Although this is not a spatial algorithm, spatial data can be used by the algorithm. Given a point data set the x and y coordinates of the points could be supplied to the algorithm and the result would be spatial groupings of locations. The centers can be used to 'thin' data points

The R code below does the following:

- generates a set of 100 normally distributed points,
- generates two clusters using the R kmeans function,
- plots the clusters in two different colours,
- plots the centre of each cluster using a distinguishing symbol (i.e. different from the other points).

Experiment with different values.

```
x <- rbind(matrix(rnorm(100,mean=0, sd = 1), ncol = 2),
           matrix(rnorm(100, mean = 0, sd = 1), ncol = 2))
colnames(x) <- c("x", "y")
(cl <- kmeans(x, 2))
plot(x, col = cl$cluster)
points(cl$centers, col = 1:2, pch = 8, cex=2)
```


y
-2
-1
0
1
2
-2
-1
0
1
2