

Epi Lab color code

Software/Packages/Add-ins
required

Software/Packages/Add-ins
recommended

Description text

R code to copy/paste into
console

R code to copy/paste into
console that needs adjustment to
your personal workspace

Websites where you can
download requirements

Lab #7 requirements

- R - <http://cran.r-project.org/bin/windows/base/>
- R Studio - www.rstudio.com/ide/download/desktop
- Internet connection
- epitools R package

Creating categorised epidemic curves

This month we re-visit epidemic curves - see the first back page lab at http://www.elsenburg.com/vetepi/BPEL/BPEL_2014_05_EpidemicCurve.pdf - in this case however we categorise aspects of the epidemic curve in order to include more information in one graph. Also, there are a few techniques in this lab that help with plotting information in R - its certainly not exhaustive but will be a start.

NOTE ON THE DATA: The data used are avian influenza outbreaks on ostrich farms classified by the Western Cape Department of Agriculture. The data has been altered slightly since the point of the lab is to get to the graph but not necessarily to evaluate it. Avian influenza outbreaks on ostrich farms are logged per farm and then classified per H and N type using PCR and/or HI testing. It does occur from time to time, especially on farms with low on-farm prevalence, that the H or N (or both) types are undefined and could not be further classified. In these cases the type is classified as UNDEFINED (a "U" in the data). This is also true for the pathogenicity of the viruses, with the difference here that if high path is not confirmed by PCR sequencing, or by epidemiological links to a high path farm, then the farm is considered low pathogenic. To re-iterate - the data are, firstly not available for further use outside this exercise and secondly the alteration thereof anyway renders it non-usable for purposes other than instruction.

The code

```
#load the epitools library if you've already installed it, if not then install
and load it
library(epitools)
#import the required dataset and call it aidata
aidata<-read.csv("http://www.jdata.co.za/backpagelabs/
backpagelabs_jdg_aidata.csv", header=T)
#view a summary of the data and check the class of the date field
aidata
summary(aidata)
class(aidata$posdate)
#the class for the date data (i.e. postdate is in a FACTOR form - a re-
minder for histograms using epitools : the class for the date should be a
DATE class and the format ideally should be "yyyy-mm-dd"
aidata$posdate
aidata$posdate<-format(as.Date(aidata$posdate), "%Y-%m-%d")
aidata$posdate
#to start - a quick and nasty epidemic curve
epicurve.months(aidata$posdate)
#This gives a very ugly boring curve, with x labels that are pointless, so this
lab is really about getting from this into to something worthwhile
#The goal is to have a epi curve for AI events separated into the H and N
subtypes - we'll start with just the H types
#Before we start run the following to add a new field for the full HN type
per event. For this we also need to concatenate the H and N type fields
aidata$subtype<-paste(aidata$hstype,aidata$ntype,sep="")
#In this case its important to add that the separation between the two
columns must be nothing (""), if you leave that out then the default separa-
tion is a SPACE
aidata$subtype
#first create a data list with all the variables needed for the epicurve graph
aicurve<-epicurve.months(aidata$posdate)
aicurve
summary(aicurve)
#you'll see that there are various variables created that essentially are the
backbone of the epicurve and these can be used in the final graph
#For the x-axis we'll need some month and year information - so the $cmon
or $cmonth variables, and the $cyear variable
```

```
#The $xvals variable indicates points on the x axis and run on for 50
months encapsulating the data
#The following commands we will run step for step - you could just run the
final one but it shows the process followed to get to the final outcome
#first sort out the y-axis - you can see on the plot that one bar (6 months
into the plot) extends past 12 events which is the current limit of the y-axis
#lets also remove the x-axis labels and y-axis, and also extend the y-axis
limit to 15
epicurve.months(aidata$posdate,
axisnames=FALSE,
yaxt="n",
ylim=c(0,15))
#add segments to view each outbreak on each farm as a separate block
epicurve.months(aidata$posdate,
axisnames=FALSE,
yaxt="n",
ylim=c(0,15),
segments="True")
#in order to view outbreak by H-type we stratify by that variable
epicurve.months(aidata$posdate,
axisnames=FALSE,
yaxt="n",
ylim=c(0,15),
segments="True",
strata=aidata$hstype)
#we want to add colours for every unique H type instead of the grey range
default
#We first need to know how many unique H types there are in the data set
length(unique(aidata$hstype))
#6 it is, so we need to add 6 colours (in this case rainbow is a nice func-
tion)
epicurve.months(aidata$posdate,
axisnames=FALSE,
yaxt="n",
ylim=c(0,15),
segments="True",
strata=aidata$hstype,
col=rainbow(6))
```

Continued on next page

#note that we could have also done that somewhat more automatically

```
epicurve.months(aidata$posdate,
  axisnames=FALSE,
  yaxt="n",
  ylim=c(0,15),
  segments="True",
  strata=aidata$hstype,
  col=rainbow(length(unique(aidata$hstype))))
```

#now to add the legend

```
epicurve.months(
  aidata$posdate,
  axisnames=FALSE,
  yaxt="n",
  ylim=c(0,15),
  segments="True",
  strata=aidata$hstype,
  col=rainbow(length(unique(aidata$hstype))),
  legend=TRUE)
```

6

#your legend will be overlapping parts of your graph and it's not ideal

#first lets get the legend text slightly smaller (cex function)

#for this we add a argument into our string that influences the legend - namely "args.legend"

```
epicurve.months(
  aidata$posdate,
  axisnames=FALSE,
  yaxt="n",
  ylim=c(0,15),
  segments="True",
  strata=aidata$hstype,
  col=rainbow(length(unique(aidata$hstype))),
  legend=TRUE,
  args.legend=list(cex=0.5))
```

7

#for all legend options please see: <http://stat.ethz.ch/R-manual/R-patched/library/graphics/html/legend.html>

#for instance - lets take the box that's around the legend items out - using the bty function

```
epicurve.months(
  aidata$posdate,
  axisnames=FALSE,
  yaxt="n",
  ylim=c(0,15),
  segments="True",
  strata=aidata$hstype,
  col=rainbow(length(unique(aidata$hstype))),
  legend=TRUE,
  args.legend=list(cex=0.5, bty="n"))
```

8

#in my plot the legend is almost in a good position - I want to move it up slightly and to the left - your screen will be different

#in R there is a way of locating an X and a Y location on a plot that you have in you PLOT WINDOW

#type in the locator function and stipulate that you want to find one location

```
locator(n=1)
```

#now take your mouse and click where you'd prefer to see the your legend - you'll see that \$x and \$y results are returned, which you could add directly into your graph code

#in the code below we however incorporate the locator function directly into the plot. You'll need to click twice in your plot where you want the top right of the legend to be - first click for x location, second for y

#also add a Legend title

```
epicurve.months(aidata$posdate,
  axisnames=FALSE,
  yaxt="n",
  ylim=c(0,15),
  segments="True",
  strata=aidata$hstype,
  col=rainbow(length(unique(aidata$hstype))),
  legend=TRUE,
  args.legend=list(cex=0.5,
    bty="n",
    x=locator(1)$x,
    y=locator(1)$y,
    title="Legend"))
```

9

pre labels

#I put the locator function in directly, you can hard code it as follows:

```
.....args.legend=list(cex=0.5,
  bty="n",
  x=47,
  y=15.....)
```

#OK - so we have a good starting point with physically how our graph should look - now to fill in the axis labels and plot title

#we can now manipulate (and insert in the case of the Y axis) the axes of the graph

```
axis(
  side=1, # puts the axis at the bottom - i.e. x axis
  at=aicurve$xvals, # labels will be placed in the xvals, fitting is automatic
  labels=aicurve$cmmonth, # labels will be months in the aicurve data
  cex.axis=0.5, # changes the text size of the axis labels
  lwd=0, # width of the axis line is zero, makes invisible
  lwd.ticks=0, # width of the tick lines also zero, makes them invisible
  mgp=c(0,0.2,0)) #mgs controls where labels are put - we want it 0.2
units below the x axis
```

#now to put the years underneath the months in the x axis

#we don't want to put in more than one label per year, to find out how many unique years there are

```
length(unique(aicurve$year))
```

```
#so 5 years - 2010 through 2014
```

```
length(aicurve$xvals)
```

#there are 50 points along the x axis where labels can be placed -if we look at the year data in the aicurve dataset

```
aicurve$year
```

#we can see that there are two 2010 months, and then 12 months per year up to and including 2014

#a good way of labelling would then be to label maybe the 2nd 2010 month, and then every JUNE for the following years using the following functions to get the xvals location

```
aicurve$xvals[c(2,8,20,32,44)] #location of label
```

```
aicurve$year[c(2,8,20,32,44)] #label to put down - could also have been "unique(aicurve$year)"
```

#so now the axis

```
axis(
```

```
  side = 1, # puts the axis at the bottom - i.e. x axis
  at=aicurve$xvals[c(2,8,20,32,44)], # labels placed in the specific xvals
  labels=aicurve$year[c(2,8,20,32,44)], # labels for the years
  cex.axis=0.5, # changes the text size of the axis labels
  lwd=0, # width of the axis line is zero, makes invisible
  lwd.ticks=0, # width of the tick lines also zero, makes them invisible
  mgs=c(0,1,0)) # we want it 1 units below the x axis so its below the
month labels
```

Continued on next page

```
#now for adding the Y axis
axis(
  side=2, # puts the axis on the left
  line="-0.8", #indents the y axis slightly
  at=seq(0,14,by=2), # creates a vector of label locations starting at 0 to
14 with points every 2 labels
  labels=seq(0,14,by=2), # here we put in a vector of labels, this must be
equal in length to the "at" value above - e.g. 7 labels for 7 points
  las=1, # rotate labels to be horizontal
  cex.axis=0.5, # changes the text size of the axis labels
  lwd=1, # width of the axis line - 0 would make this invisible
  lwd.ticks=1, # tick marks are 1 wide
  tck=-0.02, # length of ticks, negative goes out from the plot
  mgp=c(0,0.35,0)) #location of labels

#Now to add axes titles - either text or mtext functions can be used
#mtext place text in the margins (hence the m)
#text places text in the plot area itself
#x-axis title
mtext(text="Year and Month of avian influenza outbreak",
  side = 1, # Bottom (or x as we've discussed above)
  adj=0.5, # Alignment parallel to margin
  line=1.5, # Alignment relative to margin
  cex=0.5) # text size

#classify the "U"
mtext(text="*U = undefined",
  side = 1, # Bottom (or x as we've discussed above)
  adj=0.02, # Alignment parallel to margin
  line=1.7, # Alignment relative to margin
  cex=0.3) # text size

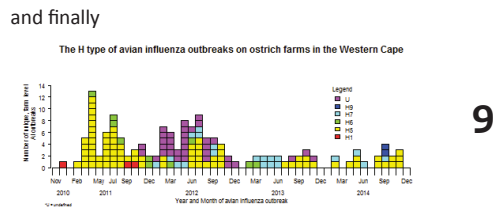
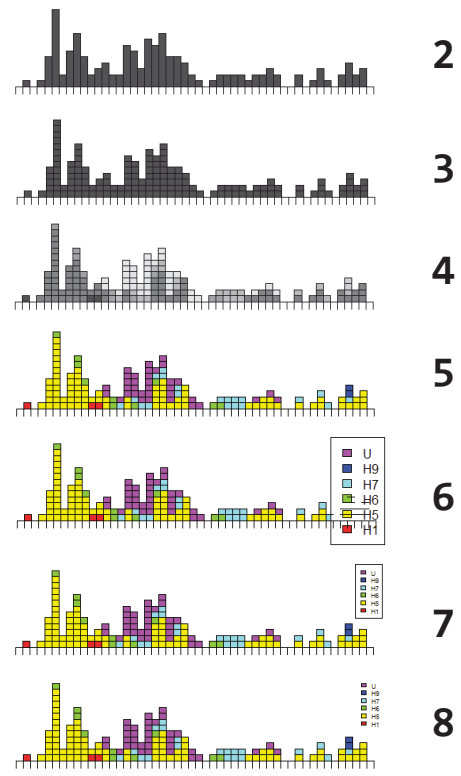
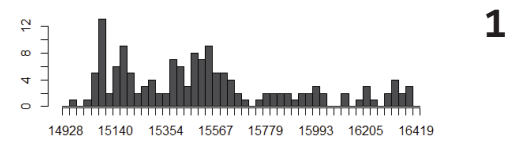
#yaxis title one
mtext(text="Number of unique, farm level",
  side = 2, # Left (or y as we've discussed above)
  adj=0.5, # Alignment parallel to margin
  line=0.5, # Alignment relative to margin
  cex=0.5) # text size

#yaxis title two
mtext(text="AI outbreaks",
  side = 2, # Left (or y as we've discussed above)
  adj=0.5, # Alignment parallel to margin
  line=0, # Alignment relative to margin
  cex=0.5) # text size

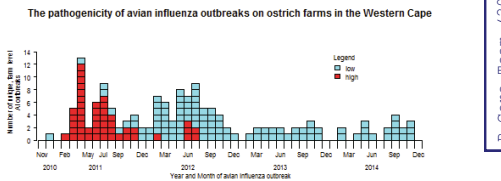
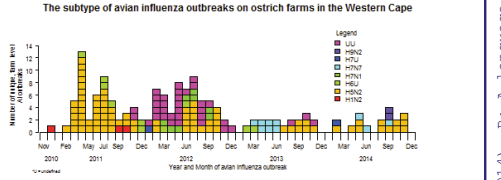
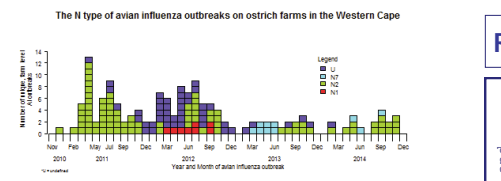
#now to add the main title
title(main=list("The H type of avian influenza outbreaks on ostrich farms in
the Western Cape", cex=0.75))
```

The result

The following series of graphs are those that are created piece meal as the code is run in the lab. Labels of the graphs are linked to the labels in the text



The n-type, HN type and pathogenicity types should look as follows when you alter the code to use these various subtypes - for these just change the stratify by variable and the text and mtext labels



References

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>

Tomas J. Aragon Developer (2012). epitools: Epidemiology Tools. R package version 0.5-7. <http://CRAN.R-project.org/package=epitools>